The Statistics of Dynamic Networks

Richard G. Clegg

A Thesis Submitted for the Degree of Ph.D.

University of York
Department of Mathematics

June 2004

Abstract

This thesis describes describes a small number of problems arising from the applied study of networks in various contexts. The work can be split into two main areas: telecommunications networks (particularly the Internet) and road networks.

In the area of telecom networks, this research focuses on current mathematical developments concerning long-range dependence (LRD). LRD is a statistical phenomenon describing correlations in time series. A large body of research has found LRD is present in measurements of data traffic on the Internet. A novel model for generating LRD is developed based upon Markov Modulated Processes. This technique has considerable advantages over a number of other methods currently used in the area.

In the area of road traffic, this work concerns the phenomenon known as driver route choice (how drivers pick their routes through a road network as day-follows-day). A survey is made of the current research in this area focussing on on-street studies and how theory (mainly equilibrium modelling) translates into practice. In analysing data related to driver route choice it became necessary to develop a technique for matching data across multiple survey sites. This novel mathematical technique uses set theory to investigate the "false match" problem in survey data. Finally, a large on-street survey is analysed statistically for insights into driver behaviour in response to a change in a network.

Contents

List of Figures

List of Tables

Acknowledgements

Thanks are due to several people who have contributed to this thesis. Maurice Dodson and Mike Smith are thanked for their patience, support and helpful suggestions. Richard Batley, Stephen Clark, Yann Golanski and David Watling all made contributions to the research contained in this thesis. Their contributions are acknowledged separately in chapter introductions.

I would like to thank Jason Levesley, Detta Dickinson, Simon Kristensen, David Arrowsmith, Arthur Clune and Simon Eveson for helpful discussions which clarified many of my ideas. Thanks are due to Sarah Mallet, Arthur Clune, Sara Marshall, Derek Muir and Doug Clow for their help with proof-reading. I would also like to thank my friends and family for their tolerance and support during the writing process.

Author's Declaration

The contents of this thesis are entirely my own work except where the contributions of other researchers are explicitly acknowledged in the text. The simulation modelling in Section 2.10 was performed with the help of Dr. Yann Golanski. Parts of Chapter 3 are based on a joint paper written with Dr. Richard Batley at the University of Leeds. His work (in a revised form) is the basis of Sections 3.7 and 3.8 and part of Section 3.9 and this work is included with his permission. Matching software written by Stephen Clark and Dr. David Watling at the University of Leeds was used with their permission in Chapter 5 to perform estimation of travel times. With these exceptions, the work within this thesis is my own.

Introduction

This thesis describes mathematical investigations of the statistical properties of dynamic networks. Two different types of networks are discussed: telecommunications networks (in this case, the Internet) and road networks. In the case of telecoms networks, the problem area studied is the statistical phenomenon known as long-range dependence (LRD). In the case of road networks a large data set is investigated for its effects on driver behaviour.

Chapter 1 discusses the theoretical background to LRD in Internet traffic. The topic is introduced with a brief summary of the various definitions of LRD which are in use. The Hurst parameter (a common measure of LRD) is introduced and related to these definitions. Recent research on LRD in the Internet is reviewed and techniques for measuring the Hurst parameter are discussed.

Chapter 2 introduces a new Markov model to generate LRD. This model is attractive in that it is extremely simple and can generate a binary time series with a known mean and Hurst parameter. In addition, some useful results are proved about the autocorrelation function of time series which take only two values. The Markov model is used as part of a simple simulation of Internet traffic. This model shows the effects of LRD on the performance of a simulated network.

Chapter 3 provides an introduction to route choice in road traffic networks. The chapter begins with a review of on-street evidence for driver route choice and departure time choice. This is placed in context with a description of the theoretical basis of route choice modelling. The review focusses on Wardrop equilibrium and the notion of traffic equilibria. The chapter concludes with

a discussion of how theoretical models are used in practice for scheme assessment. An important conclusion of this chapter is that there is a genuine lack of on-street evidence regarding the problem of driver route choice.

Chapter 4 describes a set-theoretic model which can be used to investigate matching data in multiple site surveys. This method was motivated by the need to investigate data from real-life traffic surveys. A mathematical framework is developed using set theory to describe types of match and this framework is used to create an algorithm to estimate the number of matches in survey data given certain assumptions.

Chapter 5 presents a rigorous statistical investigation of a large collection of survey data. This data set was collected in on-street surveys in the city of York. The aim of this data collection was to investigate hypotheses related to driver behaviour as described in Chapter 3. A number of standard statistical techniques are used in addition to the set theoretic method developed in Chapter 4. This data set provides an extremely useful insight into the behaviour of drivers on-street.

CHAPTER 1

Long-Range Dependence in Telecommunication Networks

This chapter provides an introduction to the topic of long-range dependence (LRD) in telecommunications (telecoms) networks. This is a large and expanding research area and this literature survey cannot be complete simply due to the huge number of papers published in the area. However, this chapter should serve as a useful summary of research on this subject.

1.1. Introduction to LRD

A good introduction to the topic of LRD is provided by [**15**]. LRD is a statistical phenomenon observed in some time series. A time series which has LRD appears stationary overall, remains at higher or lower values than its mean for relatively long periods of time and appears to exhibit cycles or trends but with no clear overall cycle emerging. LRD is also known as long memory or strong dependence and will be formally defined later. Mandelbrot also used the term *the Joseph effect* to describe the phenomenon (a reference to the biblical character who dreamed of seven fat years and seven lean years when making a prophecy for the Egyptian pharoah — and also to the fact that LRD was first observed by Hurst in analysis of flood levels of the Nile river [**85**].)

It should be noted that, throughout this chapter the asymptotic notation given in A.2 is used. Sometimes this is at odds with the literature in the area which is not always consistent and sometimes uses $\sim$ to mean *asymptotically proportional to* (which is written as $\asymp$ throughout this thesis).

1.1.1. Introductory Statistics and Time Series Analysis. This section provides a quick introduction to some basic concepts in statistics and time series analysis. For a slower paced introduction see [**80**] from which many of the definitions in this section are taken.

DEFINITION 1.1. A *sample space* is the set of all possible outcomes of an experiment. For example, consider tossing two coins. The possible outcomes are HH, HT, TH and TT. The sample space may be discrete (as in the previous example) or continuous (for example a measurement of a randomly chosen person's height in metres). Formally, a *discrete sample space* is one with a finite or countably infinite number of possible values. A *continuous sample space* is one which takes values in one or more intervals.

DEFINITION 1.2. An *event* is a subset of a sample space. For example, if the event is getting exactly one head in two coin tosses then it would be HT and TT. An example of an event on a continuous sample space is measuring a height which is between 1.5 and 2.0 metres.

DEFINITION 1.3. A *probability measure* $\mathbb{P}$ is a real-valued set function defined on a sample space S which satisfies:

(1) $0 \leq \mathbb{P}[A] \leq 1$ for every event $A \subseteq S$
(2) $\mathbb{P}[S] = 1$
(3) $\mathbb{P}[A_1 \cup A_2 \cup \dots] = \mathbb{P}[A_1] + \mathbb{P}[A_2] + \dots$ for every finite or infinite sequence of disjoint events $A_1, A_2, \dots$ where $A_i \subseteq S$.

DEFINITION 1.4. A *random variable* is a real-valued function defined on a sample space. For example, define X as the number of heads in two coin tosses or the height of a given measurement in metres. The domain of X is the sample space and its range is within the real numbers $\mathbb{R}$. A *discrete random variable* is a random variable defined on a discrete sample space and a *continuous random variable* is a random variable defined on a continuous sample space for which the probability is zero that it will assume any given value in an interval.

It should also be noted that it follows from these definitions that a real-valued function of a random variable (or a set of random variables) is itself a random variable.

DEFINITION 1.5. The *discrete density function* $f(x)$ for a discrete random variable X is given by the equation

$$f(x) = \mathbb{P}[X = x].$$

The sum of the density function up to x is known as the *distribution function* of a discrete variable. It is given by $F(x)$ in the equation

$$F(x) = \mathbb{P}[X \leq x].$$

DEFINITION 1.6. The *continuous density function* $f(x)$ for a continuous random variable X is uniquely determined by the following properties:

(1) $f(x) \geq 0$ for all $x \in \mathbb{R}$
(2) $\int_{-\infty}^{\infty} f(x)dx = 1$
(3) $\int_a^b f(x)dx = \mathbb{P}[a < x < b]$ for all $a, b \in \mathbb{R}$ where $a \leq b$.

The integral of $f(x)$ from $-\infty$ to x is known as the *distribution function* of a continuous variable. It is given by $F(x)$ in the equation

$$F(x) = \int_{-\infty}^{x} f(x).$$

Often it is useful to deal with more than one random variable at once. If two variables X and Y are considered then the system described above can easily be extended.

DEFINITION 1.7. The *joint density function* of two random variables X and Y is defined by $f(x, y)$. In the discrete case this is defined by the equation

$$f(x, y) = \mathbb{P}[X = x, Y = y].$$

In the continuous case it must possess the following properties:

(1) $f(x,y) \geq 0$
(2) $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)dxdy = 1$
(3) $\int_c^d \int_a^b f(x,y)dxdy = \mathbb{P}\left[a < X < b, c < Y < d\right]$, for all $a \leq b \in \mathbb{R}$ and $c \leq d \in \mathbb{R}$.

DEFINITION 1.8. The random variables X and Y with density functions $g(x)$ and $h(x)$ and the joint density function $f(x,y)$ are said to be *independent* if and only if

$$f(x,y) = g(x)h(y),$$

for all x and y.

Definitions 1.7 and 1.8 can be extended in the obvious way to more than two variables.

DEFINITION 1.9. The *expected value* or *expectation* of the function $g(X)$ on a discrete random variable X is given by

$$\mathrm{E}\left[g(X)\right] = \sum_{i=1}^{\infty} g(x_i)f(x_i),$$

where x_i are all the possible values of X (that is all the members of its sample space) and $f(x)$ is the density function for X.

For a continuous variable the sum in the above changes to an integral.

DEFINITION 1.10. The *expected value* or *expectation* of the function $g(X)$ on a continuous random variable X is given by

$$\mathrm{E}\left[g(X)\right] = \int_{-\infty}^{\infty} g(x)f(x)dx,$$

where $f(x)$ is the density function for X.

It should be noted that in Definitions 1.9 and 1.10 there is no guarantee that either the sum or the integral converge. If they diverge then the expectation is undefined.

The definition of expectation can easily be extended to a set of random variables $X_1, \ldots, X_n$.

DEFINITION 1.11. For random variables $X_1, \ldots, X_n$ with density function $f(x_1, \ldots, x_n)$ then the expectation value of a function $h(X_1, \ldots, X_n)$ is given by

$$\mathrm{E}[h] = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} h(x_1, \ldots, x_n) f(x_1, \ldots, x_n) dx_1 \ldots dx_n.$$

Expectation E is a linear operator. If g, g_1 and g_2 are three functions of a set of random variables then the following properties follow from the previous definitions:

- $\mathrm{E}[cg] = c\mathrm{E}[g]$ for any constant c.
- $\mathrm{E}[g_1 + g_2] = \mathrm{E}[g_1] + \mathrm{E}[g_2]$.
- $\mathrm{E}[g_1 g_2] = \mathrm{E}[g_1]\,\mathrm{E}[g_2]$, if g_1 and g_2 are independent.

The first two properties follow trivially from substituting $h = cg$ and $h = g_1 + g_2$ into Definition 1.11. The third property is derived as follows.

$$\mathrm{E}[g_1 g_2] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_1 g_2 f(g_1, g_2) dg_1 dg_2,$$

where $f(g_1, g_2)$ is the joint density function of g_1 and g_2. Since g_1 and g_2 are independent then from Definition 1.8:

$$\begin{aligned} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_1 g_2 f(g_1, g_2) dg_1 dg_2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_1 g_2 f_1(g_1) f_2(g_2) dg_1 dg_2 \\ &= \int_{-\infty}^{\infty} g_1 f_1(g_1) dg_1 \int_{-\infty}^{\infty} g_2 f_2(g_2) dg_2 \\ &= \mathrm{E}[g_1]\,\mathrm{E}[g_2], \end{aligned}$$

where $f_1(g_1)$ and $f_2(g_2)$ are the density functions of g_1 and g_2 respectively.

Using the Definitions 1.9 and 1.10 for expectation then *mean* μ and *variance* σ^2 of a random variable X can be defined.

DEFINITION 1.12. The *mean* μ of a random variable X (either discrete or continuous) is given by

$$\mu = \mathrm{E}[X].$$

DEFINITION 1.13. The *variance* σ^2 of a random variable X (either discrete or continuous) will be denoted by $\operatorname{var}(X)$ and is given by

$$\sigma^2 = \operatorname{var}(X) = \mathrm{E}\left[(X-\mu)^2\right].$$

The *standard deviation* σ is the square root of the variance.

As previously noted, the expectation is not guaranteed to converge and, for some random variables, μ and σ^2 do not exist.

Consider a time series or process $\{X_t : t \in \mathbb{N}\}$.

DEFINITION 1.14. The *autocovariance* is given by

$$\gamma(i,j) = \mathrm{E}\left[(X_i-\mu)(X_j-\mu)\right].$$

DEFINITION 1.15. The *autocorrelation* function (ACF) is given by

$$\rho(i,j) = \frac{\gamma(i,j)}{\sigma^2} = \frac{\mathrm{E}\left[(X_i-\mu)(X_j-\mu)\right]}{\sigma^2}.$$

It should be noted at this point that μ may be ∞ and σ^2 may be zero or ∞ and therefore the ACF is not defined for all processes.

DEFINITION 1.16. A process X_t is *weakly stationary* (also known as second-order stationary, wide-sense stationary or covariance stationary) if and only if:

(1) The mean exists and is finite. That is $E[X_i] = \mu$.
(2) The covariance $\gamma(i,j)$ depends only on the absolute value of the lag $k = i - j$. That is, for the lag $k = i - j$ then there is a $\gamma(k)$ such that

$$\gamma(k) = \gamma(i,j) = \gamma(j,i).$$

Throughout this thesis, unless explicitly stated, it is assumed that all processes are weakly stationary and when the term *stationary* is used without qualification it will refer to Definition 1.16. It should be noted in passing that by assuming that $\gamma(i, j)$ is defined and depends only on the lag this, in turn implies that σ^2 exists since $\sigma^2 = \gamma(i, i)$. (A strongly stationary process, by contrast, has all higher order moments constant.) If only weakly stationary processes are considered then the Definitions 1.14 and 1.15 simplify to the ones shown below.

DEFINITION 1.17. For a weakly stationary time series, the autocovariance as a function of lag k is given by

$$\gamma(k) = \mathrm{E}\left[(X_i - \mu)(X_{i+k} - \mu)\right].$$

DEFINITION 1.18. For a weakly stationary time series, the autocorrelation as a function of lag k is given by

$$\rho(k) = \frac{\gamma(k)}{\sigma^2} = \frac{\mathrm{E}\left[(X_i - \mu)(X_{i+k} - \mu)\right]}{\sigma^2}.$$

1.1.2. Definitions of LRD in Stationary Processes. A number of definitions of LRD are common in the literature — some of these are equivalent but, unfortunately, some are not. This section will list the commonly used ones and discuss which are (and which are not) equivalent. A list of selected references which cite each definition is given in order to provide some perspective as to how common each definition is. The definitions here are all given for stationary processes. The most common measure of LRD is the *Hurst parameter*, H which, for a process exhibiting LRD, is in the range $(1/2, 1)$. The asymptotic notation which is used throughout the remainder of this chapter is defined in appendix A.2.

DEFINITION 1.19. A stationary process X_t is said to be *long-range dependent* if its ACF $\rho(k)$ sums to infinity.

$$\sum_{k=-\infty}^{\infty} \rho(k) = \infty.$$

This definition is sometimes used in the literature, for example in [**123**]. (Note that the summation is usually given in the literature as being between $-\infty$ and ∞ although the assumption of weak stationarity means that $\rho(k) = \rho(-k)$.) Often a slightly less restrictive condition is used by including a modulus sign around the ACF.

DEFINITION 1.20. A stationary process X_t is said to be *long-range dependent* if the absolute value of its ACF $\rho(k)$ sums to infinity.

$$\sum_{k=-\infty}^{\infty} |\rho(k)| = \infty.$$

This definition (or the equivalent definition with autocovariance instead of ACF) is used by [**74**] [**106**] [**141**]. A more restrictive definition often used is given by putting conditions on how $\rho(k)$ decays as $k \to \infty$.

DEFINITION 1.21. A stationary process X_t is said to be *long-range dependent* if its ACF $\rho(k)$ has the asymptotic form

$$\rho(k) \sim c_\rho k^{-\alpha},$$

for some positive constant c_ρ and some real $\alpha \in (0, 1)$.

In this case, the parameter α is related to the Hurst parameter by $H = 1 - \alpha/2$. This definition is used in [**2**] [**15**] [**17**] [**41**] [**60**] [**77**] [**164**].

LRD can also be defined in terms of the spectral density of a process. This defines LRD in terms of a spectral density which has a pole at zero.

DEFINITION 1.22. The *spectral density* $f(\lambda)$ of a function with ACF $\rho(k)$ and variance σ^2 can be defined as

$$f(\lambda) = \frac{\sigma^2}{2\pi} \sum_{k=-\infty}^{\infty} \rho(k) e^{ik\lambda}.$$

Note that the spectral density is sometimes defined simply in terms of the square of the Fourier transform of the series. This equivalent definition is arrived at via the Wiener-Khinchine theorem [**160**].

DEFINITION 1.23. A stationary process X_t is said to be *long-range dependent* if its spectral density obeys

$$f(\lambda) \sim c_f |\lambda|^{-\beta},$$

as $\lambda \to 0$, for some positive constant c_f and some real $\beta \in (0, 1)$.

The parameter β is related to the Hurst parameter by $H = (1+\beta)/2$. This definition is found in [**2**] [**15**] [**41**] [**60**] [**77**]. A more general frequency domain definition is also occasionally used which allows the existence of LRD when the spectral density has a pole at a frequency $\lambda_0 \in [0, \pi]$. This definition was first used in [**64**] (cited in [**15**]) and when $\lambda_0 \neq 0$ this is sometimes known as seasonal long memory.

DEFINITION 1.24. A stationary process X_t is said to be *long-range dependent* with a pole at λ_0 if its spectral density follows

$$f(\lambda) \sim c_f (\cos\lambda - \cos\lambda_0)^{-\beta},$$

as $\lambda \to \lambda_0$, for some positive constant c_f, some frequency $\lambda_0 \in [0, \pi]$ and some real $\beta \in (0, 1)$.

This definition is referred to as *seasonal long memory* and will not be discussed further in this thesis. Details can be found in [**64**] and [**74**].

Sometimes Definitions 1.21 and 1.23 are slightly generalised by using a slowly varying function $L(x)$ (as defined in Appendix A.2) in place of c_ρ and c_f. With this replacement, the two definitions become

$$\rho(k) \sim L(k)k^{-\alpha},$$

and

$$f(\lambda) \sim L(\lambda)|\lambda|^{-\beta},$$

as $\lambda \to 0$, respectively. These definitions are used in [**17**] [**93**].

Of the definitions listed in this section, Definition 1.20 (the non summability of the modulus of the ACF) encompasses the widest class of processes and is implied by all the other definitions. It is obvious that Definition 1.19 (the non summability of the ACF) implies Definition 1.20. Further, since $\sum_{k=n}^{\infty} ck^{-\alpha}$ is infinite for all $n > 0$, all $c > 0$ and $\alpha \in (0, 1)$ then Definition 1.21 implies Definition 1.19 and, in turn, Definition 1.20.

The definitions in terms of ACF fall off and in terms of spectral density (Definitions 1.21 and 1.23) are equivalent. If Definition 1.21 holds then it can be shown (see [**167**, Chapter 5.2]) that

$$f(\lambda) \sim c_f(H)|\lambda|^{1-2H},$$

as $\lambda \to 0$, where

$$c_f = \sigma^2\pi^{-1}c_\rho\Gamma(2H-1)\sin(\pi - \pi H),$$

and Γ is Euler's Gamma function (see Section A.3). It can be seen, therefore, that Definition 1.23 holds.

Conversely, if Definition 1.23 holds then it can be shown that

$$\rho(k) \sim c_\rho(H),$$

where

$$c_\rho = \frac{2}{\sigma^2}c_f\Gamma(2-2H)\sin(\pi H - \pi/2). \tag{1.1}$$

As before, it can be seen that Definition 1.21 holds from this.

In [**64**] it is shown that seasonal long-range dependence (Definition 1.24 with $\lambda_0 \neq 0$) implies Definition 1.20 but not Definition 1.19 (that is $|\rho(k)|$ sums to infinity but $\rho(k)$ does not necessarily).

For the purposes of this thesis, Definition 1.21 will be used as the definition of LRD — although this is the strictest of the definitions encountered, almost all the processes which will be discussed will meet this definition (where this is not the case, it will be explicitly stated).

1.1.3. Some Basic Properties of LRD Series. This section attempts to list some basic properties of LRD. This section follows closely the discussion in [**15**, Chapter One].

1.1.3.1. *The Variance of the Sample Mean.* An often cited result in basic statistics is that the variance of the sample mean is the variance of the time series divided by the sample size. That is, for the n samples, the sample mean is

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}, \tag{1.2}$$

and the variance of the sample mean is

$$\operatorname{var}\left(\overline{X}\right) = \frac{\sigma^2}{n}. \tag{1.3}$$

It is often forgotten that this is only true under certain conditions. For example, [**80**, page 130] states "The property of $\overline{X}$ of possessing the mean μ and the variance σ^2/n is true not only for a normal variable but for any variable X that possesses a second moment."

The following conditions are required for equation (1.3) to hold:

(1) The population mean $\mu = \mathrm{E}[X_i]$ exists and is finite.
(2) The population variance $\sigma^2 = \operatorname{var}(X_i)$ exists and is finite.
(3) The observations $X_1, X_2, \ldots, X_n$ are uncorrelated. That is to say, $\rho(i, j) = 0$ for $i \neq j$.

By assuming that the process is stationary, the first two conditions are guaranteed. Therefore, in the following discussion the existence of μ and σ^2 are

assumed. Expanding var $\left(\overline{X}\right)$ gives:

$$\begin{aligned}
\text{var}\left(\overline{X}\right) &= \text{E}\left[\left((n^{-1}\sum_{i=1}^{n} X_i) - \mu\right)^2\right] \\
&= \text{E}\left[n^{-2}\left(\sum_{i=1}^{n} X_i\right)^2\right] - \mu^2 \\
&= \left(n^{-2}\sum_{i,j=1}^{n} \text{E}\left[X_i X_j\right]\right) - \mu^2 \\
&= n^{-2}\sum_{i,j=1}^{n} \text{E}\left[(X_i - \mu)(X_j - \mu)\right] \\
&= n^{-2}\sum_{i,j=1}^{n} \gamma(i,j),
\end{aligned}$$

and therefore,

$$\text{var}\left(\overline{X}\right) = n^{-2}\sigma^2 \sum_{i,j=1}^{n} \rho(i,j). \tag{1.4}$$

By its definition $\rho(i,i) = 1$. If $\sum_{i \neq j} \rho(i,j) = 0$, then clearly equation (1.3) holds. Otherwise, it is necessary to introduce a correction term $\delta_n(\rho)$ such that

$$\text{var}\left(\overline{X}\right) = \sigma^2[1 + \delta_n(\rho)]n^{-1}, \tag{1.5}$$

where,

$$\delta_n(\rho) = n^{-1}\sum_{i \neq j} \rho(i,j).$$

As previously discussed, for a stationary process $\rho(i,j) = \rho(i-j) = \rho(k)$ and therefore the above equation can be simplified to

$$\delta_n(\rho) = 2n^{-1}\sum_{k=1}^{n-1}(n-k)\rho(k). \tag{1.6}$$

For a process where samples are not strictly independent, equation (1.3) is adjusted by some correction factor $\delta_n(\rho)$ which depends on both the size of the sample and also on the correlation structure. It is instructive to ask how

the correlations affect the sample mean and whether equation (1.3) remains true asymptotically. Equivalently, does

$$\text{var}\left(\overline{X}\right) \sim C\sigma^2 n^{-1}, \tag{1.7}$$

where C is a finite positive constant, hold and if so under what conditions? Define

$$\delta(\rho) = \lim_{n\to\infty} \delta_n(\rho).$$

Equation (1.7) holds if the limit $\delta(\rho)$ exists. It can be readily seen from equation (1.6) that this is true if the condition of Definition 1.19 does not hold (the ACF does not sum to infinity).

To give a specific example, consider an AR(1) model as defined by equation (B.1). It is clear that for $k > 0$,

$$X_{i+k} = a_1^k X_i + a_1^{k-1}\varepsilon_{i+1} + a_1^{k-2}\varepsilon_{i+2} + \cdots + \varepsilon_{i+k},$$

where the ε_i are independent and identically distributed error terms.

Assuming that $j > i$, $k = j - i$ and $a_i \in (-1, 1)$ then $\mu = 0$. Therefore,

$$\rho(k) = \frac{\text{E}\left[X_i(a_1^k X_i + a_1^{k-1}\varepsilon_{i+1} + \cdots + \varepsilon_{i+k})\right]}{\text{E}\left[X_i^2\right]}.$$

Since all the ε_j are i.i.d. with a mean of 0 then $\text{E}\left[X_j\varepsilon_i\right] = 0$ for $i > j$. Hence,

$$\rho(k) = \frac{a_1^k \text{E}\left[X_i^2\right]}{\text{E}\left[X_i^2\right]} = a_1^k.$$

Repeating the same calculation for $j < i$ will get $\rho(k) = a_1^{-k}$ for $k = j - i$. In general, therefore,

$$\rho(k) = a_1^{|k|}.$$

It is clear that this process is not LRD since,

$$\sum_{k=-\infty}^{\infty} \rho(k) = 2\sum_{k=0}^{\infty} \rho(k) - 1 = 2/(1 - a_1) - 1,$$

which must be finite for $a_1 \in (-1, 1)$. (In fact, a simple extension of this will show that any stationary AR(p) process is not LRD for finite p). Now,

substituting this into equation (1.4) gives

$$\text{var}\left(\overline{X}\right) = n^{-2}\sigma^2\left[\sum_{i=1}^{n} 1 + \sum_{i\neq j} a_1^{|i-j|}\right] = n^{-2}\sigma^2\left[n + 2\sum_{k=1}^{n-1}(n-k)a_1^k\right].$$

Substituting from equation (1.6) gives

$$\text{var}\left(\overline{X}\right) = n^{-1}\sigma^2[1 + \delta_n(a_1)],$$

where $\delta_n(a_1) = 2n^{-1}\sum_{k=1}^{n-1}(n-k)a_1^k$. This can be rewritten as

$$\delta_n(a_1) = \frac{2a_1}{1-a_1}\left[1 - \frac{1}{n-na_1} + \frac{a_1^n}{n-na_1}\right].$$

Therefore, as $n \to \infty$ then

$$\delta(a_1) = \lim_{n\to\infty} \delta_n(a_1) = 2a_1/(1-a_1).$$

The constant $1 + \delta(a_1)$ shows how far from the ideal (independent) behaviour given by equation (1.3) the sample mean will be. It is easy to see that if a_1 is close to zero then equation (1.3) is nearly true and the sample mean will behave as expected. However, if a_1 is close to one then the sample mean could converge much more slowly than expected. However, if a_1 is near one then adjacent observations will be very similar and this will be noticed in the time series. For a typical time series which does not exhibit LRD either the sample mean will closely follow equation (1.3) or the short-range dependence will be obvious from observation. For a long-range dependent series, however, this is not the case. The correlations in the data are such that the asymptotic behaviour in equation (1.7) does not hold. More specifically, if LRD is present as specified in Definition 1.21 then as the sample size $n \to \infty$,

$$\text{var}\left(\overline{X}\right) \sim \frac{c_\rho n^{2H-2}}{\sigma^2 H(2H-1)}.$$

where H is the Hurst parameter and c_ρ is given by equation (1.1). This is proved in [**14**]. Note that, as expected, for short-range or independent data ($H = 1/2$) this will imply equation (1.7) as expected. The fact that the sample mean only converges slowly to the mean is a property which makes LRD extremely difficult to work with in real data.

1.1.3.2. *Variance and Confidence Interval Estimation for LRD.* In addition to the slower than expected convergence of the sample mean, LRD data sets have a number of other properties which make them difficult to work with.

DEFINITION 1.25. The sample variance, S^2 is given by

$$S^2 = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}{n-1},$$

It is well known that S^2 is an unbiased estimator for the variance σ^2. (See, for example, [**80**, page 221]). Again this assumes that the correlations are summable. So, to take the standard derivation:

$$\begin{aligned}
\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}^2\right)\right] &= \mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left((X_i-\mu)-(\overline{X}-\mu)\right)^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}\left[X_i-\mu\right]^2 - \mathrm{E}\left[\overline{X}-\mu\right]^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\sigma^2 - \mathrm{var}\left(\overline{X}\right) \\
&= \sigma^2 - \frac{\sigma^2}{n} \\
&= \frac{\sigma^2(n-1)}{n}.
\end{aligned}$$

The conclusion that S^2 is an unbiased estimator of σ^2 follows immediately by inspection. However, note the implicit assumption that $\mathrm{var}\left(\overline{X}\right) = n^{-1}\sigma^2$ which has already been shown not to hold for time series where correlations are important (or to put it another way, the implicit assumption was that the X_i were independent). Substituting the corrected expression from equation (1.5) then:

$$\begin{aligned}
\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}(X_i-\overline{X})^2\right] &= \sigma^2 - \frac{\sigma^2[1+\delta_n(\rho)]}{n} \\
&= \frac{\sigma^2[n-1-\delta_n(\rho)]}{n}.
\end{aligned}$$

The corrected sample variance estimate is

$$S_c^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n - 1 - \delta_n(\rho)}, \tag{1.8}$$

where $\delta_n(\rho)$ is given by equation (1.6). From the above, for the usual measure of sample variance S^2 then

$$\mathrm{E}\left[S^2\right] = \sigma^2 \left[1 - \frac{\delta_n(\rho)}{n-1}\right].$$

If H is near one then this the bias term $\delta_n(\rho)(n-1)^{-1}$ can converge very slowly to zero as n increases.

A final topic worth considering in a new light in its relation to LRD is that of confidence intervals. Student's t-statistic is given by

$$t(n) = \frac{\overline{X} - \mu}{S}\sqrt{n}, \tag{1.9}$$

where S is the square root of the sample variance S^2 [**80**, page 148]. More information about the t-statistic is given in section 5.2.1. The distribution of the t variable is near normal under a wide range of conditions. If this is the case then the t variable can be used to give confidence intervals. A $(1-a)$ confidence interval is given by

$$\overline{X} \pm z_{a/2}\frac{S}{\sqrt{n}}, \tag{1.10}$$

where $z_{a/2}$ is the upper $1 - a/2$ quantile of the standard normal distribution. However, It has already been established that, in the presence of LRD, the sample mean $\overline{X}$ converges slower than $1/\sqrt{n}$ and therefore the t-statistic diverges.

$$\lim_{n\to\infty} \mathbb{P}\left[|t(n)| > c\right] = 1,$$

for any constant c. The probability that the sample mean will lie within the bounds given by equation (1.10) will tend to zero even for a arbitrarily close to zero.

LRD data series are notoriously difficult to work with in practical situations. The sample mean converges slowly. The estimator S^2 is not an

unbiased estimator for the variance. Standard methods for estimating confidence intervals fail. These three problems make statistical tests on LRD series problematic.

1.1.4. LRD and Self-Similarity. A topic often associated with LRD is that of statistical self-similarity.

DEFINITION 1.26. Let Y_t be a stochastic process with continuous time parameter t. If the process is *self-similar* with self-similarity parameter H then for any positive constant c, the rescaled process $c^{-H}Y_{ct}$ is equal in distribution to the original process Y_t.

It should be noted that this H is the same as the Hurst parameter already encountered. A way of visualising this definition is that a process is self-similar if, when the x-axis (time axis) is stretched by a factor c and the y-axis is stretched by a factor c^{-H}, then the process looks the same statistically.

Consider a self-similar process Y_t with stationary increments and a self-similarity parameter $H \in (0,1)$. The increment process X_t is defined by: $X_i = Y_i - Y_{i-1}$ for $i \in \mathbb{N}$. It can be shown (see [**15**, page 51]) that this implies that for the process X_t the ACF is given by

$$\rho(k) \sim H(2H-1)k^{2H-2},$$

which implies that for $H \in (1/2,1)$ then $\rho(k) \asymp |k|^{\alpha}$ with $\alpha \in (0,1)$. In other words, the increment process of a self-similar process with stationary increments and $H \in (1/2,1)$ is, itself, an LRD process.

1.1.5. LRD and Heavy Tails. Heavy-tailed distributions (see [**3**]) are strongly related to LRD. The heavy-tailed distribution was called by Mandelbrot the *Noah Effect* (by analogy with the Joseph effect). A heavy-tailed distribution is one where the tail of the distribution function decreases to zero more slowly than exponentially. That is, for all $\varepsilon > 0$, a random variable X is heavy-tailed if it satisfies

$$\mathbb{P}[X > x]\, e^{\varepsilon x} \to \infty, \qquad x \to \infty.$$

It has been observed that many processes associated with computer networks follow a heavy tailed distribution — the lengths of files stored on computers and the amount of data which is transferred by a given connection to the Internet. It has been shown [**144**] that a superposition of ON/OFF sources (in Internet traffic this could be visualised as packet trains and inter-train pauses) will give rise to a time series exhibiting LRD if the lengths of the ON/OFF periods are heavy-tailed.

1.2. Modelling Techniques for LRD

There are a number of different methods which are standardly used in the modelling of long-range dependence. This section is a brief tour of these modelling techniques. In Chapter 2 a new process for generating LRD is discussed.

1.2.1. Fractional Brownian Motion and Fractional Gaussian Noise.

Brownian motion is a stochastic process $B(t)$ with the following properties:

- $B(t)$ is Gaussian,
- $B(0) = 0$ almost surely,
- $B(t)$ has independent increments,
- $E[B(t) - B(s)] = 0$,
- $\text{var}\,(B(t) - B(s)) = \sigma^2|t - s|$.

Brownian motion is the increment process of independent normal variables with zero mean and variance σ^2 (Gaussian Noise). Assuming that σ^2 above was normalised to one then this process can be defined as follows:

- $B(0) = 0$ almost surely,
- $B(t)$ is a continuous function of t,
- The distribution of $B(t)$ obeys

$$\mathbb{P}\left[B(t+k) - B(t) \leq x\right] = (2\pi k)^{-\frac{1}{2}} \int_{-\infty}^{x} \exp\left(\frac{-u^2}{2k}\right) du.$$

The process defined by $B(t+k) - B(t)$ is normally distributed with zero mean and variance k and is known as Gaussian White Noise. An obvious generalisation of this is to change the final condition to

$$\mathbb{P}\left[B_H(t+k) - B_H(t) \leq x\right] = (2\pi)^{-\frac{1}{2}} k^{-H} \int_{-\infty}^{x} \exp\left(\frac{-u^2}{2k^{2H}}\right) du. \tag{1.11}$$

where $H \in (1/2, 1)$ is the Hurst parameter.

The process B_H is known as fractional Brownian motion (FBM). It is the increment process of fractional Gaussian noise (FGN). FBM is a self-similar process with self-similarity parameter H. FGN is a stationary process which exhibits LRD with Hurst parameter H.

A number of authors have described computationally efficient methods for generating FGN and FBM — [**39**] (described in [**15**, page 216]) [**102**] and [**114**].

1.2.2. The Fractional Auto-Regressive Integrated Moving Average Model. The Fractional Auto-Regressive Integrated Moving Average (FARIMA) model is an obvious extension of the ARIMA model described in Appendix B. Equation (B.3) begs the obvious question, "what happens if the requirement $d \in \mathbb{Z}_+$ is relaxed to $d \in \mathbb{R}$?"

To make this generalisation the idea of fractionally differencing a time series is necessary. Consider the expression $(1-\mathbf{B})^d$ as found in equation (B.3). This can be expanded formally using the standard binomial series as

$$(1 - \mathbf{B})^d = \sum_{k=0}^{d} \binom{d}{k} (-1)^k \mathbf{B}^k. \tag{1.12}$$

where $\mathbf{B}$ is the backshift operator described in Section A.1.

Now, it is well known that,

$$\binom{d}{k} = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)},$$

where Γ is Euler's Gamma function defined in Appendix A.3. The replacement of the factorial with the Γ function means non-integer values for d can be used

by slightly altering equation (1.12) to

$$(1-\mathbf{B})^d = \sum_{k=0}^{\infty} \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}(-1)^k \mathbf{B}^k.$$

Note that, in fact, this only produces interesting processes for $d \in (-1/2, 1/2)$. So the FARIMA model is the ARIMA model with $d \in (-1/2, 1/2)$ instead of $d \in \mathbb{Z}_+$ and can be written as equation (B.3). It can also be written as

$$\left(1 - \sum_{j=1}^{p} a_j \mathbf{B}^j\right)\left(\sum_{k=0}^{\infty} \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}(-1)^k \mathbf{B}^k\right) X_i = (1 - \sum_{j=1}^{q} \theta_j \mathbf{B}^j)\varepsilon_i, \tag{1.13}$$

where $d \in (-1/2, 1/2)$. As might be expected the d parameter relates to the Hurst parameter. The relation is simply $H = d + 1/2$ — note that this only produces expected values for H when $d \in (0, 1/2)$. It can be seen from this definition that a value X_i depends on every previous X_j where $(j < i)$ — an obvious reason why the model has long memory.

FARIMA processes were proposed by [**63**] and a description in the context of LRD can be found in [**15**, pages 59–66].

1.2.3. Iterated Chaotic Maps. It has been known for some time that LRD can be generated using a family of chaotic maps. Take a map from the family given by

$$x_{n+1} = F(x_n; d, m_1, m_2) = \begin{cases} F_1(x_n) = x_n + \frac{1-d}{d^{m_1}} x_n^{m_1} & 0 < x_n < d, \\ F_2(x_n) = x_n - \frac{d}{(1-d)^{m_2}}(1-x_n)^{m_2} & d < x_n < 1, \end{cases} \tag{1.14}$$

where $d \in (0, 1)$ and $m_1, m_2 \in (3/2, 2)$. The map is shown in Figure 1.1. If $m_2 = 1$ this is the well known Manneville-Pomeau map. A problem with working with this map analytically is that there is no closed form for its invariant density. For this reason piecewise linear approximations to the map are often used.

Pioneering work in this area is [**154**] with early applications to telecoms being given by [**49**].

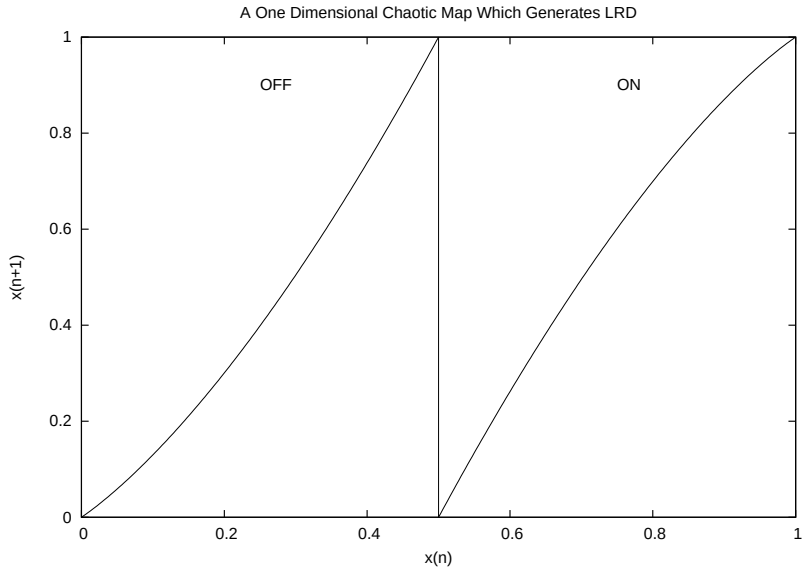

FIGURE 1.1. A one dimensional chaotic map for generating LRD.

The map is used to generate LRD by generating a binary series from the regions labelled *ON* and *OFF* in the diagram. The procedure used is described in Table 1.1.

(1) Pick a starting value for $x_0 \in (0, 1)$. Set $i = 0$.
(2) If $x_i \geq d$ then $y_i = 1$ otherwise $y_i = 0$.
(3) Calculate x_{i+1} using equation (1.14).
(4) Increment i and go to step two.

TABLE 1.1. A procedure for generating LRD using a one dimensional chaotic map.

The time series y_i generated by this procedure will have LRD. The mean will depend on the parameters d, m_1 and m_2. The Hurst parameter in this case is given by the largest value of m_1 and m_2. If $m = \max(m_1, m_2)$ then $H = (3m - 4)/(2m - 2)$. An explanation for the presence of LRD in this map is provided by examining the behaviour of the orbits at x_i near zero or one.

The escape from points near zero or one is extremely slow and this causes long sequences of zeros or ones in the generated y_i series.

1.2.4. Other Modelling Techniques. A technique gaining favour in modelling (and also in measuring) LRD is wavelet analysis. This allows the LRD hypothesis to be generalised to multifractals. While multifractal analysis is beyond the scope of this thesis, a passing mention is given here since wavelet based multifractal analysis is becoming important in the analysis of teletraffic. LRD (at least as described by Definition 1.21) defines a single scaling behaviour for the system (which applies in the tail of the ACF) — if this scaling behaviour was the same at any scale then the process defined would be a monofractal. However, if the scaling behaviour differs across scales then the process is a multifractal. There is some evidence (which will be discussed in Section 1.4) that Internet traffic exhibits different scaling behaviour at different timescales. A general description of multifractal processes and wavelets is found in [**123**] and a description of how wavelets can be used to create models with the same multifractal spectrum as a given data set can be found in [**124**].

Self-similar processes can be simulated (and, indeed, measured) using Embedded Branching Processes. Details can be found in [**90**] and [**91**]. An aggregation of ON-OFF sources with heavy-tails can be shown to generate a series with LRD. A modelling process based upon this is described in [**142**].

1.3. Measuring Techniques for LRD

A large number of techniques exist for measuring the presence of LRD in data series. There is no single perfect technique for measuring LRD and a variety are listed here. A good summary of a number of techniques and code for making estimations is to be found on Taqqu's website [**140**]. The descriptions in this section are essentially summaries of those found on this website.

All the techniques listed are estimators for the parameter H. The proofs of these techniques are beyond the scope of this thesis but, where practical, justifications for their usage are given.

1.3.1. The R/S Statistic. The R/S statistic (also known as rescaled adjusted range) is one of the oldest and best known techniques for estimating H. The R/S plot relies on the idea that in the presence of LRD more extreme events are more common. It is discussed in detail in [**103**] and also [**15**, pages 83–87].

For a time series $\{X_t : t = 1, 2, \ldots, N\}$ with partial sums given by $Y(n) = \sum_{i=1}^{n} X_i$ and the sample variance given by

$$S^2(n) = \frac{1}{n-1}\sum_{i=1}^{n} X_i^2 - \frac{1}{n(n-1)}Y(n)^2,$$

then the R/S statistic is given by

$$\frac{R}{S}(n) = \frac{1}{S(n)}\left[\max_{1\le t\le n}\left(Y(t) - \frac{t}{n}Y(n)\right) - \min_{1\le t\le n}\left(Y(t) - \frac{t}{n}Y(n)\right)\right]. \quad (1.15)$$

For FGN or FARIMA then:

$$\mathrm{E}\left[R/S(n)\right] \sim C_H n^H,$$

where C_H is a positive, finite constant independent of n.

The procedure to estimate H is therefore as follows: For a time series of length N subdivide the series into K blocks each of size N/K. For each lag n compute $R/S(n)$ for all the series which start at points $k_i = iN/K + 1$ for $i = 0, 1, \ldots, K-1$. (K should be chosen so that the blocks do not overlap). In this way, a number of estimates of $R/S(n)$ are obtained for each value of n.

By choosing logarithmically spaced values of n and plotting $\log[R/S(n)]$ versus $\log n$ then a straight line should be obtained. This plot is sometimes called the *pox plot* for the R/S statistic. The gradient of the line should be H.

There are several problems with this technique — most notably, there are more estimates of the statistic for low values of n where the statistic is affected most heavily by short range correlation behaviour. On the other hand, for

high values of n there are too few points for a reliable estimate. The values between these high and low cut off points should be used to estimate H but, in practice, often it is the case that widely differing values of H can be found by this method depending on the high and low cut off points chosen. Also it is worth noting that the convergence to a straight line is proven for FARIMA and FGN and not for LRD time series in general.

Lo modified the R/S statistic to use a weighted sum of autocovariances for normalisation instead of the sample variance. Details are found in [**95**].

1.3.2. Aggregated Variance. This method measures H by considering the scaling of the variance as the time series is aggregated. Given a time series $\{X_t : t = 1, 2, \ldots, N\}$ then divide this into blocks of length m and aggregate:

$$X^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X_i, \quad k = 1, 2, \ldots, N/m.$$

The sample variance is given by

$$\widehat{\operatorname{var}\,(X^{(m)})} = \frac{1}{(N/m) - 1} \sum_{k=1}^{N/m} \left(X^{(m)}(k) - \overline{X}\right)^2. \tag{1.16}$$

The sample variance should be asymptotically proportional to m^{2H-2} for large N/m and m. To use this method plot successive values of the aggregated variance as given by equation (1.16) against m on a log-log plot. The slope of the line of best fit should be $2H - 2$. As with the R/S statistic the low and high ends of the plot cannot be used (for the same reasons). A description of this method in slightly different terms can be found in [**15**, page 92].

Jumps in the mean and slowly decaying trends can severely affect this statistic. One technique to combat this is to difference the aggregated variance and work instead with

$$\widehat{\operatorname{var}\,(X^{(m+1)})} - \widehat{\operatorname{var}\,(X^{(m)})}).$$

1.3.3. Variance of Residuals. This method is described in more detail in [**117**]. Take the series $\{X_t : t = 1, 2, \ldots, N\}$ and divide it into blocks of length m. Within each block calculate partial sums: $Y(t) = \sum_{i=1}^{t} X_i$. For each block make a least squares fit to a line $a + bt$. Subtract this line from the samples in the block to obtain the residuals and then calculate their variance

$$V(m) = \frac{1}{m} \sum_{t=1}^{m} \left(Y(t) - a - bt\right)^2 .$$

The variance of residuals is proportional to m^{2H} and therefore a log-log plot of $\log(V(m))$ versus $\log(m)$ should be a line with a slope of $2H$. As with the previous time domain measures of H this method is strongly affected by the cut off points used for the sizes of m.

1.3.4. Periodogram. The periodogram is a frequency domain technique described in [**59**]. For a time series $\{X_t : t = 1, 2, \ldots, N\}$ it is defined by

$$I(\lambda) = \frac{1}{2\pi N} \left| \sum_{j=1}^{N} X_j e^{ij\lambda} \right|^2 ,$$

where λ is the frequency. If the variance of the series is finite then $I(\lambda)$ is an estimator of the spectral density of X_t. A series with long-range dependence will, by Definition 1.23, have a spectral density proportional to $|\lambda|^{1-2H}$ for frequencies close to $\lambda = 0$. (Note that this specifically rules out LRD of the type in Definition 1.24 where the frequency pole is not at zero.) Therefore, a log-log plot of the periodogram should have a slope of $1 - 2H$ close to the origin.

1.3.5. Whittle's Maximum Likelihood Estimator. The subject of Maximum Likelihood Estimators for LRD is a complex one and is covered in some detail in [**15**, pages 100–123]. While an exact MLE is available, its calculation is computationally demanding and an approximation is used for practical calculations. In brief, Whittle's MLE works by first calcuating $I(\lambda_j)$ for Fourier frequencies $\lambda_j = 2\pi j/N$ where N is the length of the time series.

The algorithm seeks to find a function $f^*(\lambda_j, \eta)$ which minimises, $Q^*(\eta)$ defined as

$$Q^*(\eta) = \sum_{j=1}^{(N-1)/2} \frac{I(\lambda_j)}{f^*(\lambda_j, \eta)},$$

where $f^*(\lambda_j, \eta)$ is chosen as a functional form related to the assumed functional form of the LRD series and η represents the parameters of this function. For example, if the functional form assumed is FARIMA$(0, d, 0)$ then η represents the d parameter. If the series is assumed to be FARIMA(p, d, q) then η also includes the coefficients in the AR and MA parts of the function. The estimate $\hat{H}$ converges to the true value H at a rate of $\sqrt{N}$ if the assumptions of the model are met. Further details can be found in [**56**]. The method known as Aggregated Whittle provides additional robustness by aggregating the data.

The Whittle estimator specifies the functional form of the power spectrum at all frequencies. A semi-parametric version known as Local Whittle is also available which assumes only the functional form chosen where λ is near zero. Details of the method are given in [**125**].

1.3.6. Other Estimation Methods and Comparison of Methods. Wavelet analysis has been used for the estimation of the Hurst parameter. In addition this has the benefit of providing an estimate of the multifractal spectrum of the data [**123**] [**124**]. Crossing trees (analysis of where a process crosses certain preset levels) can be used to estimate H in self-similar data sets [**90**] [**91**]. Higuchi's method estimates H by estimating fractal dimension of path lengths [**79**]. In addition, the techniques known as Absolute Moments and Ratio of Variance of Residuals are described on Taqqu's website [**140**]. A method known as the global log-periodogram estimator is described in [**108**] and is a frequency domain technique which uses the entire frequency spectrum to estimate H.

A number of authors have compared the different estimation techniques for H. Several techniques are compared empirically in [**145**] by testing methods against time series which are FARIMA $(0, d, 0)$ or FGN with known H.

The methods tested included R/S, Whittle, Aggregated Variance, Higuchi's Method and the Periodogram. Of these methods Whittle's was found to be clearly the best method with the lowest variance in its predictions and the least bias of those methods tested (each method was tested on fifty realisations of data sets for each value of H tried and for FARIMA and FGN data).

Whittle type techniques (Whittle, Aggregated Whittle and Local Whittle) are compared in [**142**]. Since the Whittle technique requires specification of the functional form of the data set, the paper takes also investigates what happens when the functional form is misspecified (for example, a FARIMA$(0, d, 0)$ model is fitted to data which is FARIMA$(1, d, 0)$). The conclusion is that if the series is known to be FGN or FARIMA of a given order then a correctly specified Whittle estimator gives the smallest biases and standard errors. On the other hand, an incorrectly fitted model can give poor performance, and if the form of the model is not known then, provided the time series is long enough (they cite $N = 10,000$), Aggregated Whittle or Local Whittle are to be preferred with Local Whittle performing slightly better in the tests quoted.

Semi-parametric techniques are investigated in [**6**] which compares a number of techniques for their ability to estimate FGN, FARIMA$(0, d, 0)$ and FARIMA$(1, d, 1)$ data sets. Wavelets, global log periodogram, Whittle and Local Whittle are amongst the techniques compared. The global log periodogram and Local Whittle techniques are considered to be the most effective.

1.4. LRD in the Internet

In 1993, Leland, Taqqu, Willinger and Wilson published their classic paper [**93**] which identified the presence of LRD in data sets captured on Ethernet Local Area Network (LAN) traffic. Since its publication, this paper has been cited more than five hundred times. The paper mainly discusses the subject in terms of self-similarity and concludes: "In the case of Ethernet LAN traffic, self-similarity is manifested in the absence of a natural length of a 'burst'; at every time scale ranging from a few milliseconds to minutes and hours, bursts

consist of bursty sub-periods separated by less burst sub-periods. We also show that the degree of self-similarity (defined via the Hurst parameter) typically depends on the utilisation level of the Ethernet and can be used to measure 'burstiness' of LAN traffic." The data sets measured in the paper are from 1989–1992. A bibliography of research in the area up to 1996 is found in [**165**] which references more than four hundred papers related to the subject area. A non-technical introduction and review of research up to 1999 is provided by [**128**]. An introduction to the difficulties of modelling and measuring Internet behaviour in general is provided by [**55**]. A more recent summary of work in the area is found in [**164**]. While this thesis does not actually make any measurements using telecoms data, a brief survey of the most relevant work in the area will provide context for the research undertaken.

1.4.1. Traffic Measurements. The paper [**93**] used R/S analysis, aggregated variance and Whittle's estimator to investigate a large number of Ethernet measurements made between 1989 and 1992. The paper examined busy times, "normal" traffic times and low traffic times and considered time series of packets per unit time. The conclusion was that LAN traffic is statistically self similar. The Hurst parameter H was shown to be a function of the usage of the Ethernet (higher usage meaning a higher Hurst parameter). Resolving the traffic into separate components (breaking it up by destination or protocol used) showed that the traffic shared a characteristic H value for all major components. In [**166**] some of these same traces are analysed to show that heavy tails are present in the data sources. That is, if the distribution of the lengths of data sent is plotted then its distribution is heavy tailed. As has been mentioned previously, aggregation of heavy-tailed sources leads to long-range dependent time series. The same paper uses measurements on a wide area network (WAN) collected in 1994 at Bellcore and demonstrates that these measurements show not only heavy tails but also long-range dependence.

Some studies show that it is not even necessary for there to be a network for traffic to exhibit LRD. For example, [**16**] makes measurements on video

traffic. The traffic is shown to be long-range dependent at source due to the encoding of the video stream.

In [**115**] a number of WAN traces collected from 1989 to 1994 are analysed. The general hypothesis of long-range dependence is confirmed. In addition to this, statistical models are given for how users connections to the network using various protocols. The protocol used is critical with some connections being Poisson and others presenting distributions of connection times which are completely at odds with Poisson modelling.

In [**107**] two hour traffic traces collected for the paper, each lasting twenty four hours and made on 100Mb links at Harvard University, were analysed. Using variance time plots of the bytes per unit time, the authors concluded that the traffic was long-range dependent.

In addition to the above studies, in a 1996 review of research in the area, [**165**] listed [**46**] [**50**] [**57**] [**82**] [**163**] as having "provided convincing evidence that actual traffic data from working packet networks are consistent with statistical *self-similarity* or *fractal* characteristics... measured packet traffic data are consistent with *long-range dependence*..."

A different view is put in [**24**] which reports measurements made on some high speed networks. The paper does not clearly describe when the measurements examined were taken but claims that in high speed networks then the merging of large numbers of data streams mean that the traffic tends to Poisson as the load increases and that in larger networks, the assumption that traffic is LRD is erroneous. Since the paper is both new and controversial, it is hard to say at this time whether the authors' claims will stand up to further analysis.

There is some controversy as to whether LRD is the best model for telecoms traffic with [**143**] claiming using analysis of a number of traces collected between from 1989 and 1994 that LRD (monofractal) modelling is sufficient and it is unnecessary to introduce the extra parameters required by multifractal modelling. However, other authors disagree. For example, the scaling

properties of teletraffic are discussed by [**81**] which suggests that LRD in teletraffic happens in two separate regimes: a scaling behaviour at time scales above one second and a less clear scaling behaviour at time scales below one second. (The data observed was recorded between 1998 and 2002. Each of these regimes is characterised by a different Hurst parameter.) One possible suggestion is that the nature of telecoms traffic has changed over time and the nature of the modelling required has also changed.

1.4.2. Engineering Implications. The reason for the considerable interest in the subject is the fact that the engineering implications of long-range dependence on queuing performance can be considerable. If Internet traffic is not modelled well by independent or short-range dependent models then much traditional queuing theory work based upon the assumption of Poisson processes is no longer appropriate. Traffic which is long-range dependent in nature can have a queuing performance which is significantly worse than Poisson traffic.

In general it has been found that a higher Hurst parameter often increases delays in a network, the probability of packet loss and affects a number of measures of engineering importance. In fact [**48**] claims that the Hurst parameter is "...a dominant characteristic for a number of packet traffic engineering problems...". Some of the effects on queuing performance are given by [**112**] [**128**]. However, [**111**] shows that while the Hurst parameter is important to queueing, the relationship is not a simple one — in some cases a high Hurst parameter may improve performance or have no effect. (A commonly given example is when the LRD arises from aggregation of heavy-tails in the OFF periods of the traffic sources this does not impact queuing performance.)

1.4.3. The Origins of LRD in Networks. In the literature, four possible origins for LRD in networks are commonly cited. These are as follows:

(1) LRD is inherent directly in the source of data.
(2) LRD is a result of the aggregation of heavy-tailed data streams.

(3) LRD is a result of feedback mechanisms in the TCP protocol.
(4) LRD arises from network topology.

These causes are explained in detail below. It is important to emphasise that these explanations are not contradictory. Each might make a contribution to the packet traffic behaviour of the network.

(1) The evidence that LRD arises directly in the source of data comes mainly from studies of video traffic (see [**16**], [**57**] and [**126**]). The claim in these papers is that variable-bit-rate (VBR) video traffic by its nature exhibits LRD at source. The LRD, in this case, arises from an encoding mechanism whereby video is encoded as a series of differences between frames with occasional full updates. In contrast to most literature in the field, [**126**] claims that this LRD has no "significant effect on Cell Loss Ratio". However, it should be emphasised that measurements of Internet traffic show that real time video traffic, indeed any Universal Datagram Protocol (UDP) traffic, is likely to be only a very small percentage of Internet traffic. It seems unlikely, therefore, that VBR video traffic could be the main component of LRD observed in aggregate traces (some from as far back as 1989). Of course, if VBR video traffic contains LRD at source then it is likely that other applications might have traffic distributions with unexpected statistical effects. For example, [**115**] shows that telnet packets are not well modelled by a Poisson distribution.
(2) The proposal that LRD in Internet traffic arises from the aggregation of heavy-tailed data streams is similar to the previously mentioned mechanism but has a slightly less direct causal mechanism. A causal connection between heavy-tailed sources and LRD was long suspected and is proved in [**142**]. There are good reasons to believe that source traffic on the Internet is heavy-tailed — file sizes and sizes of accessed web documents have been shown to have heavy tails (see [**35**]).

(3) Another potential cause of LRD is the feedback mechanisms in the Transmission Control Protocol (TCP). Markov chains were used in [**53**] to model TCP timeout and congestion window behaviour and the authors prove that these can cause what the authors refer to as "local long-range dependence" (that is, LRD up to a certain time scale).

(4) Finally, there remains the distinct possibility that LRD is an emergent property of the networks themselves. Measurements made in [**20**] show that, even when "packet inter-departure times are independent, arrival times at the destination show LRD". This, obviously indicates that round trip times in networks are LRD processes. This work is extended to multifractal measures by [**89**]. Recent work [**4**] shows that LRD can arise in a relatively simple simulation where Poisson sources randomly situated in a grid network are aggregated as they route via shortest paths to randomly situated sinks. Moreover, it shown that when the Poisson sources are changed to LRD, increased LRD occurs at both host and router sites.

Determining the origin of LRD in Internet networks remains an important research area and it is uncertain which (if any) of these four causes is really at the heart of the problem. The possibility remains that it is a mixture of some or all of them.

CHAPTER 2

Markov Modelling of Long-Range Dependence

2.1. Introduction

This chapter describes a Markov chain based model for producing time-series exhibiting LRD. The model is a Markov Modulated Process (MMP) which generates a time series $\{Y_t : t \in \mathbb{N}\}$ which exhibits LRD.

Section 2.2 provides a simple introduction to the topic of Markov chains and MMP. Much of this discussion is taken from [**92**]. Section 2.3 introduces the structure of the infinite Markov model for LRD. In Section 2.4 a finite approximation is given and it is shown that this converges to the infinite model. In Section 2.5 a simple proof is given about the auto-correlation function for two state processes. In Section 2.6 the parameters for the infinite Markov model are given and the asymptotic form of the ACF is proved to be that for LRD. In Section 2.7 an algorithm is given for implementing the finite chain computationally. In Section 2.8 an improved algorithm is given which implements the infinite chain computationally. In Section 2.9 the MMP is implemented and tested computationally. In Section 2.10 the simulated model is used as a source model for a simple simulation of a computer network with LRD sources showing the effect of the LRD on packet loss. I am grateful with Dr. Yann Golanski who was instrumental in setting up and running the simulation procedures used with his permission in this section. The work is discussed and placed in context in Section 2.11.

2.1.1. The Need for a New Modelling Method for LRD. Given the large (and not exhaustive) list of modelling techniques given in Section 1.2, it might be asked whether there was a need for another model. However, the

model here is specifically designed to be the simplest possible computational representation of LRD.

Fractional Gaussian Noise (Section 1.2.1) and FARIMA (Section 1.2.2) are relatively simple to analyse from a statistical point of view (though the model described here is arguably simpler). However, these processes cannot easily be calculated in an ongoing manner (that is, the entire time-series is usually generated "at once" and, having generated n points, the user must effectively start again to generate the $n+1$th point) [**90**].

Iterated chaotic maps (Section 1.2.3) are computationally parsimonious but are analytically problematic since no closed form for the invariant density of the map is known. Therefore, it is difficult to generate traffic with a given mean using the iterated map method and progress theoretically is difficult.

2.2. Markov Chains and Markov Modulated Processes

DEFINITION 2.1. The sequence of random variables $\{X_t : t \in \mathbb{N}\}$ is a *discrete-time Markov chain* if it takes values in some discrete sample space Ω ($X_t \in \Omega$ for all t) and, for all $t \in \mathbb{N}$ and for all $i_k \in \Omega$, then

$$\mathbb{P}\left[X_n = i_n | X_1 = i_1, X_2 = i_2, \ldots, X_{n-1} = i_{n-1}\right] = \mathbb{P}\left[X_n = i_n | X_{n-1} = i_{n-1}\right].$$

In other words, a discrete-time Markov chain is a discrete-valued time-series where the value at time t depends only on the value at time $t-1$.

The possible values of the Markov chain Ω are known as the *states* of the chain. It is usually assumed (and will be throughout this chapter) that the possible states of the chain are numbered using integers. That is $\Omega \subseteq \mathbb{Z}$.

DEFINITION 2.2. An *homogenous, discrete-time Markov chain* is a discrete-time Markov chain for which

$$\mathbb{P}\left[X_n = i_n | X_{n-1} = i_{n-1}\right],$$

is independent of n.

All the Markov chains which are discussed in this thesis are discrete-time, homogenous Markov chains with integer numbered states. If the phrase Markov chain is used unqualified within this chapter it will refer to a discrete-time, homogenous Markov chain with integer numbered states.

DEFINITION 2.3. The *transition probabilities* for a discrete-time, homogenous Markov chain are given by p_{ij} where

$$p_{ij} = \mathbb{P}\left[X_{n+1} = j | X_n = i\right].$$

This is the probability that the chain will move from state i to state j in one time step.

DEFINITION 2.4. The *m-step transition probabilities* for a discrete-time, homogenous Markov chain are given by $p_{ij}^{(m)}$ where

$$p_{ij}^{(m)} = \mathbb{P}\left[X_{n+m} = j | X_n = i\right].$$

DEFINITION 2.5. A *Markov Modulated Process* is a process or time-series $\{Y_t : t \in \mathbb{N}\}$ where Y_i is a function of an underlying Markov chain or, more generally, where the density function of Y_i is a function of an underlying Markov chain.

$$Y_t = g(X_t),$$

for some function $g(x)$ or, more generally Y_t might be generated by sampling from a probability distribution which depends upon the state of the underlying chain.

DEFINITION 2.6. A Markov chain is *irreducible* if, for all states $i, j \in \Omega$, then there exists some m such that $p_{ij}^{(m)} > 0$. That is any state j can be reached from any state i.

DEFINITION 2.7. A state $i \in \Omega$ of a Markov chain is *periodic* with period γ if, for some $k > 0$ and, for some $\gamma \in \mathbb{N} : \gamma \geq 2$, then,

$$p_{ii}^{(m)} \begin{cases} \geq 0 & m \in \{\gamma, 2\gamma, 3\gamma, \dots\} \\ = 0 & \text{otherwise.} \end{cases}$$

In other words, a state i is periodic with period γ if returns to the state i are only permitted after some multiple of the period γ.

DEFINITION 2.8. For a state $j \in \Omega$ then for $n \in \mathbb{N}$ the n-step *first return probability* $r_j^{(n)}$ is the probability that the first return to state j occurs after n steps or

$$r_j^{(n)} = \mathbb{P}\left[X_{t+n} = i | X_t = i, X_{t+1} \neq i, X_{t+2} \neq i, \ldots, X_{t+n-1} \neq i\right].$$

DEFINITION 2.9. The terms *recurrent* and *transient* are defined in terms of the probability r_j that the state of a chain ever returns to $j \in \Omega$, where

$$r_j = \sum_{n=1}^{\infty} r_j^{(n)}.$$

A state $j \in \Omega$ is *recurrent* if $r_j = 1$ and *transient* if $r_j < 1$.

DEFINITION 2.10. The *mean recurrence time* M_i of a state $i \in \Omega$ of a Markov chain is given by

$$M_i = \sum_{n=1}^{\infty} n r_i^{(n)}.$$

In other words, M_i is the expectation value of the first recurrence time. A recurrent state $i \in \Omega$ of a Markov chain is said to be *recurrent null* if $M_i = \infty$ and *recurrent nonnull* if $M_i < \infty$.

DEFINITION 2.11. The probability of finding the system in state $j \in \Omega$ at time n is given by

$$\pi_j^{(n)} = \mathbb{P}\left[X_n = j\right],$$

and the limiting probabilities (if they exist) are given by

$$\pi_j = \lim_{n \to \infty} \pi_j^{(n)}.$$

The terms π_j are known as the *equilibrium probabilities* for the states.

The following two theorems are taken from [**92**, page 29].

THEOREM 2.1. The states of an irreducible Markov chain are either all transient or all recurrent nonnull or all recurrent null. If periodic, then all states have the same period γ.

THEOREM 2.2. In an irreducible and aperiodic homogenous Markov chain, the limiting probabilities π_j always exist and are independent of the distribution of X_1 (the initial state of the chain). Moreover either

(1) all states are transient or all states are recurrent null and $\pi_j = 0$ for all $j \in \Omega$, or
(2) all states are recurrent nonnull and for all $j \in \Omega$, $\pi_j = 1/M_j > 0$.

If the second case occurs in Theorem 2.2 then the quantities π_j are uniquely determined from the equations

$$\sum_{j\in\Omega} \pi_j = 1, \tag{2.1}$$

and

$$\pi_j = \sum_{i\in\Omega} \pi_i p_{ij}. \tag{2.2}$$

DEFINITION 2.12. A state of a Markov chain is said to be *ergodic* if it is irreducible, aperiodic and recurrent nonnull. If all of the states are ergodic then the chain itself is said to be ergodic and π_j is known as the *equilibrium probability* of state j.

DEFINITION 2.13. The *transition probability matrix* $\mathbf{P}$ is the matrix of the elements p_{ij} for all $i, j \in \Omega$ and the equilibrium probability vector $\boldsymbol{\pi}$ is the vector of all the equilibrium probabilities π_j for all $j \in \Omega$.

For example, if $\Omega = \{1, 2, \ldots, n\}$, then the transition matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix},$$

and the equilibrium probability vector by

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)\,.$$

Equation (2.2) can be rewritten in terms of $\mathbf{P}$ and $\boldsymbol{\pi}$ as:

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}. \tag{2.3}$$

Note that, together with equation (2.1) this will always form $n+1$ equations for an n state chain and therefore $\boldsymbol{\pi}$ is always in principle determined fully by these two equations.

2.3. An Infinite Markov Model for LRD

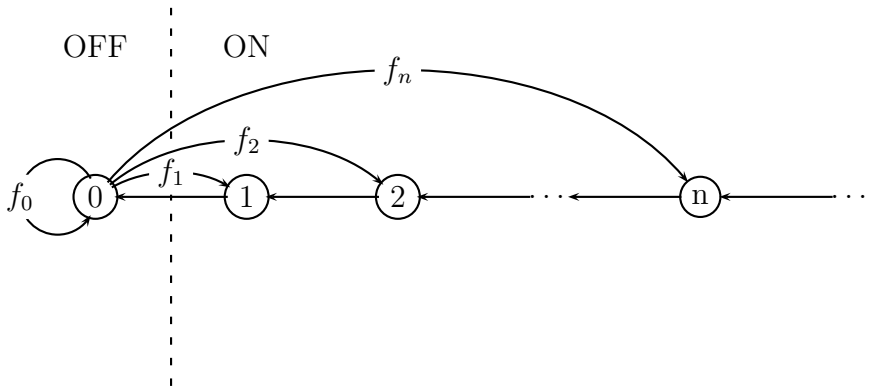

FIGURE 2.1. An infinite Markov chain which generates a time series exhibiting LRD.

The infinite Markov model developed in this research to capture the statistical properties of Internet data is shown in Figure 2.1. This chain with various transition parameters has been studied by a number of authors. The model is extremely simple. All states of the chain have a probability one transition to a lower state except for state zero which has a probability f_i of transition to a new state i.

The Markov chain will be used to derive two time series $\{X_t : t \in \mathbb{N}\}$ and $\{Y_t : t \in \mathbb{N}\}$. X_i is the state of the Markov chain at time step i.

DEFINITION 2.14. The traffic process generated by the chain is given by Y_i which is 0 if $X_i = 0$ and 1 otherwise. In other words, the system will emit at rate 1 if the underlying Markov chain is in a state other than zero.

It will be shown that, for a suitable choice of values for the f_i then the process Y_i will have long-range dependence.

The transition matrix $\mathbf{P}$ for the model shown is given by

$$\mathbf{P} = \begin{bmatrix} f_0 & f_1 & f_2 & \dots & f_n & \dots \\ 1 & 0 & 0 & \dots & 0 & \dots \\ 0 & 1 & 0 & \dots & 0 & \dots \\ 0 & 0 & 1 & \dots & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix} \tag{2.4}$$

It is an obvious property that the sum of all the transitions *from* any state to all other states must equal one. Therefore,

$$\sum_{i=0}^{\infty} f_i = 1. \tag{2.5}$$

PROPOSITION 2.1. If $f_0 > 0$ and for any i there exists $f_j > 0$ where $j \geq i$ then the chain given above is irreducible and aperiodic.

PROOF. State zero can be reached from any state. A given state $j > 0$ will reach state $j - 1$ after one step and therefore state zero after j steps. More generally, a state j will reach a state i after $j - i$ steps, if $j \geq i$. Since for any i there exists $f_j > 0$ by the condition given in the proposition, then state i can always be reached from state zero for any i. State zero can be reached from any state $i > 0$ and state i can be reached from state zero. Therefore the chain is irreducible.

To see that the chain is aperiodic, consider state zero. If it has an orbit of period γ then it can also have an orbit of period $\gamma + 1$ since $f_0 > 0$. Therefore, state zero is not periodic and by Theorem 2.1 the chain itself is aperiodic. □

PROPOSITION 2.2. Given choices of f_i such that the chain is aperiodic and irreducible then the chain is also ergodic if and only if $\sum_{i=0}^{\infty} i f_i < \infty$.

PROOF. Consider the state zero. The probability that the next state is i is given by f_i from the definition of the chain. The probability that the chain will be in state i after one step is given by f_i. If the chain is in state i then the total time to return to state zero is $i+1$ (i steps plus the one step already taken). By Definition 2.8 then $r_0^{(i+1)} = f_i$. Therefore, from Definition 2.10,

$$M_0 = \sum_{i=0}^{\infty}(i+1)f_i = \sum_{i=0}^{\infty} f_i + \sum_{i=0}^{\infty} i f_i.$$

By equation (2.5), the mean return time for state zero is

$$M_0 = 1 + \sum_{i=0}^{\infty} i f_i.$$

This expression is finite if and only if $\sum_{i=0}^{\infty} i f_i < \infty$ and therefore this is the condition for state zero to be recurrent nonnull. From Theorem 2.1 the chain itself is, therefore, recurrent nonnull. Since by hypothesis, the chain is irreducible and aperiodic then, from Theorem 2.2, the chain is ergodic. □

Proposition 2.2 gives a second condition on the chain. From this point on, it is assumed that the f_i variables will be chosen in such a way as to guarantee both of these conditions are met and the chain is ergodic.

2.4. A Finite Approximation to this Model

It is convenient to approximate this model with a finite Markov chain with $N+1$ elements numbered from 0 to N. This chain is constructed from the previous chain with transition probabilities g_i^N (for state i in the model with states from 0 to N). These transition probabilities are constructed from the rules

$$g_i^N = \begin{cases} f_i & 0 < i < N, \\ \frac{1}{N}\sum_{j=N}^{\infty} j f_j & i = N, \\ 1 - \sum_{j=1}^{N} g_j^N & i = 0. \end{cases} \tag{2.6}$$

Note that the condition on g_N^N is valid only if $\sum_{j=N}^{\infty} j f_j < N$ (otherwise $g_N^N > 1$). The condition on g_0^N ensures that the transition probabilities sum to 1. The chain is similar to the previous chain but with the states $N - \infty$

combined into a single state. The transition probabilities are the same as for the infinite chain except for states 0 and N. The same reasoning can be used to prove that this chain is also irreducible, aperiodic, recurrent nonnull and therefore ergodic. The only difference is that proposition 2.1 must be modified slightly.

PROPOSITION 2.3. If $g_0^N > 0$ and $g_N^N > 0$ the chain given above is ergodic.

PROOF. The conditions for irreducible and aperiodic follow from the same reasoning as for the infinite chain. The recurrent nonnull condition follows from the fact that the maximum possible time any state can take to get back to state 0 is N steps (from state N). State 0 is recurrent nonull and therefore all the other states must also be. □

Now it remains to be shown that, as $N \to \infty$, the $N+1$ state approximation of this chain tends to the same equilibrium probabilities as the infinite model. Denote the equilibrium probabilities of the ith state of the finite model with $N+1$ states (numbered 0 to N) as π_i^N.

THEOREM 2.3. The equilibrium probabilities of the finite model are given by

$$\pi_i^N = \pi_0^N \sum_{j=i}^{N} g_j^N,$$

for $i \geq 0$.

PROOF. From equation (2.2), for $0 \leq i < N$,

$$\begin{aligned} \pi_i^N &= g_i^N \pi_0^N + \pi_{i+1}^N \\ &= g_i^N \pi_0^N + g_{i+1}^N \pi_0^N + \pi_{i+2}^N \qquad \text{for } 0 \leq i < N-1 \\ &= g_i^N \pi_0^N + \cdots + g_{N-1}^N \pi_0^N + \pi_N^N, \end{aligned}$$

and

$$\pi_N = g_N^N \pi_0^N.$$

Hence,

$$\pi_i^N = \pi_0^N \sum_{j=i}^{N} g_j^N,$$

as required. □

Next it will be shown that the finite model converges to a limit as $N \to \infty$. It is necessary to show that both $g_i^N \to f_i$ and $\pi_i^N \to \pi_i$ for all values of $0 \leq i \leq N$. This will be shown in two parts.

PROPOSITION 2.4. In the limit as $N \to \infty$ then $g_i^N \to f_i$.

PROOF. Define $\delta_i^N = g_i^N - f_i$ for all $N > 0$. The proposition is equivalent to the claim that for any $\varepsilon > 0$ there exists an N_ε such that $|\delta_i^N| < \varepsilon$ for all $N > N_\varepsilon$ and for all i in the range $0 \leq i \leq N$.

For $0 < i < N$ the proposition is trivially true since $g_i^N = f_i$ and therefore $\delta_i^N = 0$. Consider the two remaining cases, for $i = 0$ and for $i = N$.

Firstly, for $i = N$,

$$g_N^N = \frac{1}{N} \sum_{i=N}^{\infty} i f_i.$$

Therefore,

$$\delta_N^N = \left(\frac{1}{N} \sum_{i=N}^{\infty} i f_i \right) - f_N = \frac{1}{N} \sum_{i=N+1}^{\infty} i f_i.$$

Since $\sum_{i=1}^{\infty} i f_i < \infty$, let $\sum_{i=1}^{\infty} i f_i = K$. It follows that $\sum_{i=N+1}^{\infty} i f_i \leq K$ for all $N > 0$. Therefore, there exists some N_ε such that $K/N_\varepsilon < \varepsilon$. Thus for all $N \geq N_\varepsilon$ it is the case that $|\delta_N^N| < \varepsilon$.

For $i = 0$,

$$f_0 = 1 - \sum_{i=1}^{\infty} f_i,$$

and also,

$$\begin{aligned} g_0^N =& 1 - \sum_{i=1}^{N} g_i^N, \\ =& 1 - \sum_{i=1}^{N-1} f_i - \frac{1}{N} \sum_{i=N}^{\infty} i f_i. \end{aligned}$$

Therefore,

$$\delta_0^N = \left| \sum_{i=N+1}^{\infty} \frac{1}{N}(Nf_i - if_i) \right| = \left| -\frac{1}{N} \sum_{i=N+1}^{\infty} (i-N)f_i \right|.$$

Now, since $i > N$ and $f_i \geq 0$ then,

$$\left| -\frac{1}{N} \sum_{i=N+1}^{\infty} (i-N)f_i \right| < \left| \frac{1}{N} \sum_{i=N+1}^{\infty} if_i \right|.$$

Therefore there exists some N_ε such that $|\delta_0^N| < \varepsilon$ for all $N > N_\varepsilon$ since $\sum_{i=N+1}^{\infty} if_i$ is convergent.

□

This is the first part of the proof that the finite model converges to a limit. It now remains to show that the equilibrium probabilities converge.

PROPOSITION 2.5. For every i then π_i^N is a Cauchy sequence and will therefore converge to a limit as $N \to \infty$.

PROOF. The proposition is equivalent to the claim that for any $\varepsilon > 0$ there exists N_K such that $|\pi_i^N - \pi_i^M| < \varepsilon$ for all $N, M > N_K$ and for all $i : 0 \leq i \leq N_K$.

Define $\gamma_i^N = \pi_i^N - \pi_i^{N+1}$ for all $N > 0$. Assume, without loss of generality, $M > N$.

$$\pi_i^N - \pi_i^M = \sum_{j=N}^{M-1} \gamma_i^j, \tag{2.7}$$

for all $M \geq N$.

Since $\pi_i^N = \pi_0^N \sum_{j=i}^N g_j^N$ from Theorem 2.3 and $\pi_0^N = 1 - \sum_{i=1}^N \pi_i^N$ from equation (2.1). Therefore,

$$\begin{aligned}
\pi_0^N &= 1 - \pi_0^N \sum_{i=1}^N \left(\sum_{j=i}^N g_j^N \right) \\
&= 1 - \pi_0^N \sum_{i=1}^N i g_i^N \\
&= 1 - \pi_0^N \sum_{i=1}^{N-1} i f_i - \pi_0^N N \frac{1}{N} \sum_{i=N}^{\infty} i f_i \\
&= 1 - \pi_0^N \sum_{i=1}^{\infty} i f_i.
\end{aligned}$$

Soliving for π_0^N gives

$$\pi_0^N = (1 + \sum_{i=1}^{\infty} i f_i)^{-1}. \tag{2.8}$$

Thus $\gamma_0^N = 0$ and therefore from equation (2.7) the proposition is true for $i = 0$. That is $\pi_0^N = \pi_0^M$ for all $M, N > 0$. In fact, this should be no surprise, since g_N^N was chosen to ensure this property.

Again, from Theorem 2.3, $\pi_i^N = \pi_0^N \sum_{j=i}^N g_j^N$ and hence from the definition of γ_i^N, for all $i > 0$,

$$\gamma_i^N = \pi_0 \left[\sum_{j=i}^{N-1} f_j + \frac{1}{N} \sum_{j=N}^{\infty} j f_j \right] - \pi_0 \left[\sum_{j=i}^{N} f_j + \frac{1}{N+1} \sum_{j=N+1}^{\infty} j f_j \right],$$

which simplifies to

$$\gamma_i^N = \pi_0 \left[\frac{1}{N} - \frac{1}{N+1} \right] \sum_{j=N+1}^{\infty} j f_j. \tag{2.9}$$

It is obvious that since all $f_j \geq 0$ and $N > 0$,

$$\sum_{j=N}^{\infty} j f_j \geq \sum_{j=N+1}^{\infty} j f_j.$$

Therefore, from equations (2.7) and (2.9),

$$\begin{aligned}\pi_i^N - \pi_i^M =& \sum_{k=N}^{M-1} \pi_0 \left[\left(\frac{1}{k} - \frac{1}{k+1} \right) \sum_{j=N+1}^{\infty} j f_j \right] \\ \leq& \pi_0 \left(\sum_{j=N+1}^{\infty} j f_j \right) \sum_{k=N}^{M-1} \left[\frac{1}{k} - \frac{1}{k+1} \right] \\ \leq& \pi_0 \left[\frac{1}{N} - \frac{1}{M} \right] \sum_{j=N+1}^{\infty} j f_j. \end{aligned} \tag{2.10}$$

Clearly, since $\sum_{j=N+1}^{\infty} j f_j < \infty$ there exists some N_ε such that for all $N > N_\varepsilon$,

$$\pi_0 \frac{1}{N} \sum_{j=N+1}^{\infty} j f_j < \varepsilon,$$

and since $M > N > 0$,

$$\left| \pi_0 \left(\frac{1}{N} - \frac{1}{M} \right) \sum_{j=N+1}^{\infty} j f_j \right| < \varepsilon,$$

for any choice of M and N such that $M > N > N_\varepsilon$.

Combining this with equation (2.10) gives

$$|\pi_i^N - \pi_i^M| \leq \pi_0 \left[\frac{1}{N} - \frac{1}{M} \right] \sum_{j=N+1}^{\infty} j f_j < \varepsilon,$$

for any choice of M and N such that $M > N > N_\varepsilon$ and for any $i > 0$. The case for $i = 0$ has already been covered and thus the proposition is proved. □

From Propositions 2.4 and 2.5 then Theorem 2.3 can be extended to the infinite case as follows:

$$\pi_i = \lim_{N \to \infty} \pi_i^N = \pi_0 \sum_{j=i}^{\infty} f_j \qquad \text{for } i > 0. \tag{2.11}$$

It is also useful to extend equation (2.8) to the infinite case.

$$\pi_0 = \lim_{N \to \infty} \pi_0^N = (1 + \sum_{i=1}^{\infty} i f_i)^{-1}. \tag{2.12}$$

2.5. The ACF for Two-state Processes

The ACF, $\rho(k)$, for a stationary time series $\{X_t : t \in \mathbb{N}\}$ was given in Definition 1.18 and the autocovariance, $\gamma(k)$, was given in Definition 1.17.

Consider a time series, $\{X_t : t \in \mathbb{N}\}$ where $X_t \in \{a, b\}$ for all t and where $a \neq b$.

The following shorthand notations will be used throughout this chapter

$$P_k(a) = \mathbb{P}\left[X_{t+k} = a | X_t = a\right]$$
$$P_k(b) = \mathbb{P}\left[X_{t+k} = b | X_t = b\right]$$
$$p = \mathbb{P}\left[X_t = a\right].$$

Note that clearly, if the time series is to have two values, these three quantities must be in the range $(0, 1)$.

The mean μ is given by

$$\mu = \mathrm{E}\left[X_t\right] = pa + (1-p)b. \tag{2.13}$$

The variance σ^2 is given by

$$\sigma^2 = \mathrm{E}\left[(X_t - \mu)^2\right] = \mathrm{E}\left[X_t^2\right] - \mu^2 = a^2 p + b^2(1-p) - (pa + (1-p)b)^2$$
$$= p(1-p)(a-b)^2.$$

Theorem 2.4. For a weakly-stationary time series $X_t : t \in \mathbb{N}$, which can only take two distinct values a and b, the autocorrelation function $\rho(k)$ is given by

$$\rho(k) = P_k(a) + P_k(b) - 1 = \frac{P_k(a) - p}{(1-p)} = \frac{P_k(b) - (1-p)}{p}.$$

Proof. Rearranging equation (2.13) then

$$a - \mu = (1-p)(a-b), \tag{2.14}$$

and,

$$b - \mu = p(b-a). \tag{2.15}$$

Since the series has only two values, it must be the case that

$$P_k(a) = 1 - \mathbb{P}\left[X_{t+k} = b | X_t = a\right]$$

and,

$$P_k(b) = 1 - \mathbb{P}\left[X_{t+k} = a | X_t = b\right].$$

The auto-covariance $\gamma(k)$ is given by

$$\begin{aligned}
\gamma(k) &= \mathrm{E}\left[(X_{t+k} - \mu)(X_t - \mu)\right] \\
&= \mathbb{P}\left[X_{t+k} = a, X_t = a\right](a-\mu)^2 \\
&\quad + \mathbb{P}\left[X_{t+k} = b, X_t = a\right](a-\mu)(b-\mu) \\
&\quad + \mathbb{P}\left[X_{t+k} = b, X_t = b\right](b-\mu)^2 \\
&\quad + \mathbb{P}\left[X_{t+k} = a, X_t = b\right](a-\mu)(b-\mu) \\
&= \mathbb{P}\left[X_{t+k} = a | X_t = a\right]p(a-\mu)^2 \\
&\quad + \mathbb{P}\left[X_{t+k} = b | X_t = a\right]p(a-\mu)(b-\mu) \\
&\quad + \mathbb{P}\left[X_{t+k} = b | X_t = b\right](1-p)(b-\mu)^2 \\
&\quad + \mathbb{P}\left[X_{t+k} = a | X_t = b\right](1-p)(a-\mu)(b-\mu) \\
&= P_k(a)p(a-\mu)^2 + (1-P_k(a))p(a-\mu)(b-\mu) \\
&\quad + (1-P_k(b))(1-p)(a-\mu)(b-\mu) + P_k(b)(1-p)(b-\mu)^2.
\end{aligned}$$

Substituting from equations (2.14) and (2.15),

$$\begin{aligned}
&= P_k(a)[p(1-p)^2(a-b)^2 + p^2(1-p)(a-b)^2] - p^2(1-p)(a-b)^2 \\
&\quad + P_k(b)[(1-p)p^2(a-b)^2 + p(1-p)^2(a-b)^2] - p(1-p)^2(a-b)^2 \\
&= P_k(a)p(1-p)(a-b)^2 + P_k(b)p(1-p)(a-b)^2 - p(1-p)(a-b)^2 \\
&= \sigma^2[P_k(a) + P_k(b) - 1].
\end{aligned}$$

Therefore, since $\rho(k) = \gamma(k)/\sigma^2$,

$$\rho(k) = P_k(a) + P_k(b) - 1, \tag{2.16}$$

which is the first part of the theorem.

Again, taking the autocovariance gives

$$\begin{aligned}\gamma(k) &= \mathrm{E}\left[(X_{t+k}-\mu)(X_t-\mu)\right] = \mathrm{E}\left[X_{t+k}X_t\right]-\mu^2 \\ &= P_k(a)pa^2 + (1-P_k(a))pab + (1-P_k(b))(1-p)ab \\ &\quad + P_k(b)(1-p)b^2 - \mu^2 \\ &= a(a-b)p\left[P_k(a)-p\right] - b(a-b)(1-p)\left[P_k(b)-(1-p)\right] \\ &= \frac{\sigma^2}{a-b}\left[\frac{a}{(1-p)}\left[P_k(a)-p\right] - \frac{b}{p}\left[P_k(b)-(1-p)\right]\right].\end{aligned}$$

This gives an ACF

$$\rho(k) = \frac{a\left[P_k(a)-p\right]}{(1-p)(a-b)} - \frac{b\left[P_k(b)-(1-p)\right]}{p(a-b)} \tag{2.17}$$

Setting this equal to equation (2.16) and rearranging gives

$$\begin{aligned}P_k(a)+P_k(b)-1 &= \frac{aP_k(a)-ap}{(1-p)(a-b)} - \frac{bP_k(b)-b(1-p)}{p(a-b)} \\ p(1-p)(a-b)P_k(b) + bP_k(b)(1-p) &= apP_k(a) - p(1-p)(a-b)P_k(a) \\ &\quad + p(1-p)(a-b) - ap^2 + b(1-p)^2 \\ (1-p)[P_k(b)-1] &= p[P_k(a)-1] \\ \frac{P_k(a)-p}{(1-p)} &= \frac{P_k(b)-(1-p)}{p}.\end{aligned}$$

Substituting this into equation (2.17) gives

$$\rho(k) = \frac{a\left[P_k(b)-(1-p)\right]}{p(a-b)} - \frac{b\left[P_k(b)-(1-p)\right]}{p(a-b)} = \frac{P_k(b)-(1-p)}{p},$$

which, in view of equation (2.16) completes the proof. □

Definition 2.15. Let $I(X_t)$ be an indicator variable which has the value 1 if $X_t = a$ and 0 otherwise.

Definition 2.16. Let A_n, where $n \in \mathbb{N}$, be the expected number of occurences of the value a in n samples from a weakly stationary, two-valued time-series. Let $A_n(t)$ be the number of occurences of a between X_{t+1} and

X_{t+n}. Then

$$A_n(t) = \sum_{i=1}^{n} I(X_{t+i}).$$

THEOREM 2.5. Given the conditions of the previous theorem then, for $k > 2$,

$$\rho(k) = \frac{\text{var}\,(A_{k+1}) - 2\text{var}\,(A_k) + \text{var}\,(A_{k-1})}{2p(1-p)}.$$

PROOF. Expanding the variance of $A_n(t)$ in terms of expectation values gives:

$$\begin{aligned}
\text{var}\,(A_n(t)) &= \text{E}\left[A_n(t)^2\right] - \text{E}\left[A_n(t)\right]^2 \\
&= \text{E}\left[\left(\sum_{i=1}^{n} I(X_{t+i})\right)^2\right] - \text{E}\left[\sum_{i=1}^{n} I(X_{t+i})\right]^2 \\
&= \text{E}\left[\left(\sum_{i=1}^{n} I(X_{t+i})\right)^2\right] - \left(\sum_{i=1}^{n} \text{E}\left[I(X_t + i)\right]\right)^2
\end{aligned}$$

Since the series is stationary, $\text{E}\left[X_t\right] = \text{E}\left[X_0\right]$ and also $\text{E}\left[X_t X_{t+k}\right] = \text{E}\left[X_0 X_k\right]$. Therefore $\text{E}\left[I(X_t)\right] = \text{E}\left[I(X_0)\right]$. Similarly $\text{var}\,(A_n) = \text{var}\,(A_n(t))$. Substituting and rearranging the first sum gives

$$\begin{aligned}
\text{var}\,(A_n) = \text{E}\left[2\left(\sum_{i=0}^{n-1}(n-i)I(X_0)I(X_i)\right) - \left(\sum_{i=1}^{n} I(X_0)I(X_0)\right)\right] \\
- \left(\sum_{i=1}^{n} \text{E}\left[I(X_0)\right]\right)^2.
\end{aligned}$$

Clearly, $\text{E}\left[I(X_0)\right] = p$ and $\text{E}\left[I(X_0)^2\right] = p$. Also:

$$\begin{aligned}
\text{E}\left[I(X_t)I(X_{t+k})\right] &= \text{E}\left[I(X_0)I(X_k)\right] \\
&= \mathbb{P}\left[X_k = a, X_0 = a\right] \\
&= \mathbb{P}\left[X_k = a | X_0 = a\right] \mathbb{P}\left[X_0 = a\right] \\
&= P_k(a)p.
\end{aligned}$$

Making these substitutions and rearranging the sums gives

$$\begin{aligned}\text{var}\,(A_n) &= \left(2\sum_{i=0}^{n-1}(n-i)\text{E}\left[I(X_0)I(X_i)\right]\right) - \sum_{i=1}^{n}\text{E}\left[I(X_0)I(X_0)\right] - \left(\sum_{i=0}^{n-1}p\right)^2 \\ &= 2\left(\sum_{i=1}^{n}(n-i)P_i(a)p\right) - np - n^2p^2.\end{aligned}$$

By the same process,

$$\text{var}\,(A_{n+1}) = 2\left(\sum_{i=0}^{n}(n+1-i)P_i(a)p\right) - (n+1)p - (n+1)^2p^2.$$

Taking the first difference gives

$$\text{var}\,(A_{n+1}) - \text{var}\,(A_n) = 2\left(\sum_{i=0}^{n}P_i(a)p\right) - p - 2np^2 - p^2.$$

Similarly,

$$\text{var}\,(A_n) - \text{var}\,(A_{n-1}) = 2\left(\sum_{i=0}^{n-1}P_i(a)p\right) - p - 2(n-1)p^2 - p^2,$$

where $n \geq 2$. The second difference is therefore,

$$\text{var}\,(A_{n+1}) - 2\text{var}\,(A_n) + \text{var}\,(A_{n-1}) = 2p(P_n - p).$$

Therefore, subsituting this into Theorem 2.4 gives, for $n \geq 2$,

$$\rho(n) = \frac{\text{var}\,(A_{n+1}) - 2\text{var}\,(A_n) + \text{var}\,(A_{n-1})}{2p(1-p)}.$$

□

2.6. Introducing Correlations Into the Markov Traffic Model

The original aim of the Markov model was to produce a time series Y_i (as given in Definition 2.14 which exhibits LRD. The next step is to choose f_i so as to induce correlation in the time series Y_t in order to meet the conditions for LRD. From Definition 1.21, a series is LRD with Hurst parameter H if the ACF $\rho(k)$ meets the condition,

$$\rho(k) \sim Ck^{-\alpha}, \tag{2.18}$$

where $\rho(k)$ is the ACF, C is some positive constant and $\alpha \in (0,1)$. The Hurst parameter is then given by $H = 1 - \frac{\alpha}{2}$.

An obvious way to introduce correlations over a lag of k is to include unbroken sequences of k or more 1s into a binary time series. In other words,

$$\mathbb{P}\left[Y_i = 1, Y_{i+1} = 1 \ldots Y_{i+k} = 1\right] \sim Ck^{-\alpha}.$$

This occurs if and only if $X_i \geq k$. The desired property is that $\mathbb{P}\left[X_i > k\right] \sim Ck^{-\alpha}$ which, in an ergodic chain, is equivalent to requiring that the sum of all states k or larger falls off with the form

$$\sum_{i=k}^{\infty} \pi_i \sim Ck^{-\alpha}.$$

To achieve this, an extremely strict condtion is introduced for $k > 0$,

$$\sum_{i=k}^{\infty} \pi_i = Ck^{-\alpha}, \tag{2.19}$$

where C is a constant. Note, that there is, as yet, no guarantee that this is a valid Markov chain — this will be discussed later.

The constant C can be quickly calculated by setting $k = 1$.

$$\sum_{i=1}^{\infty} \pi_i = 1 - \pi_0 = C1^{-\alpha} = C,$$

and therefore $C = 1 - \pi_0$. Therefore equation (2.19) becomes

$$\sum_{i=k}^{\infty} \pi_i = (1 - \pi_0)k^{-\alpha} \qquad k > 0.$$

From this equation for k and subtracting the same equation for $k + 1$

$$\pi_k = (1 - \pi_0)[k^{-\alpha} - (k+1)^{-\alpha}] \qquad k > 0.$$

Similarly, taking equation (2.11) for k and $k + 1$,

$$\pi_0 f_k = \pi_k - \pi_{k+1} \qquad k > 0,$$

and therefore for $k > 0$,

$$f_k = \frac{1 - \pi_0}{\pi_0}\left[k^{-\alpha} - 2(k+1)^{-\alpha} + (k+2)^{-\alpha}\right] \qquad k > 0. \tag{2.20}$$

Obviously, to satisfy the Markov property given in equation (2.5),

$$f_0 = 1 - \sum_{i=1}^{\infty} f_i,$$

which can be seen to be

$$f_0 = 1 - \frac{1-\pi_0}{\pi_0} \sum_{i=1}^{\infty} \left[i^{-\alpha} - 2(i+1)^{-\alpha} + (i+2)^{-\alpha} \right].$$

Expanding the sum and changing the limits gives

$$f_0 = 1 - \frac{1-\pi_0}{\pi_0} \left[\sum_{i=1}^{\infty} i^{-\alpha} - 2\sum_{i=2}^{\infty} i^{-\alpha} + \sum_{i=3}^{\infty} i^{-\alpha} \right].$$

Most of the terms of the sum cancel leaving

$$f_0 = 1 - \frac{1-\pi_0}{\pi_0} \left[1 - 2^{-\alpha} \right]. \tag{2.21}$$

2.6.1. A Brief Summary of the Infinite Chain Model. This infinite chain model is the main outcome of this chapter. For reference, the model is summarised here. The infinite chain model, is a Markov chain as shown in Figure 2.1. The chain generates a zero when in state zero and a one otherwise. The model has only two parameters, $\pi_0 \in (0,1)$ and $\alpha \in (0,1)$.

The first, π_0 is the equilibrium probability of the zero state and is $1-\mu$ where μ is the mean output of the model. The second, α is related to the Hurst parameter by the equation $H = 1 - \alpha/2$.

The transition probabilities f_k of the chain are given by equation (2.20) as

$$f_k = \frac{1-\pi_0}{\pi_0} \left[k^{-\alpha} - 2(k+1)^{-\alpha} + (k+2)^{-\alpha} \right] \qquad \text{for } k > 0,$$

and for $k = 0$,

$$f_0 = 1 - \frac{1-\pi_0}{\pi_0} \left[1 - 2^{-\alpha} \right].$$

The equilibrium probabilities of the chain are given for $k > 0$ by

$$\pi_k = (1-\pi_0)[k^{-\alpha} - (k+1)^{-\alpha}],$$

and, as has already been stated, π_0 is a parameter of the model.

2.6.2. Checking the Infinite Chain is Valid. It has been assumed that the chain is ergodic. Recall that by Proposition 2.2 the chain is ergodic if $\sum_{i=0}^{\infty} if_i < \infty$.

Substituting from equation (2.20) gives

$$\sum_{k=0}^{\infty} kf_k = \frac{1-\pi_0}{\pi_0} \sum_{k=1}^{\infty} \Big[k^{1-\alpha} - 2(k+1)^{1-\alpha} + 2(k+2)^{-\alpha} + (k+2)^{1-\alpha} \\ -2(k+2)^{-\alpha} \Big].$$

The series telescopes, therefore,

$$\sum_{k=0}^{\infty} kf_k = \frac{1-\pi_0}{\pi_0} \left[1 - 2^{1-\alpha} + 2.2^{-\alpha}\right] = \frac{1-\pi_0}{\pi_0}. \tag{2.22}$$

This is finite as required when $\alpha \in (0,1)$.

Finally, it should be noted that this equation is not valid for every possible combination of π_0 and α. In particular, for values of π_0 near zero then the term $(1-\pi_0)/\pi_0$ becomes extremely large and values of f_i from equation (2.20) will be negative but, since they are probabilities, they must remain in the range $(0,1)$. The fact that the model is invalid for some combinations of π_0 and α is not a great problem and the model can be confined to the valid region for experiments. Rearranging equation (2.21) shows that for $\alpha, \pi_0 \in (0,1)$ then $f_0 \in (0,1)$ if,

$$\pi_0 > \frac{2^{\alpha}-1}{2^{\alpha+1}-1}.$$

2.6.3. The ACF of the Infinite Chain. The ACF for the infinite chain can be approximated as $k \to \infty$ using a method due to Wang [**154**] which in turn derives, in part, from [**52**] and [**47**]. Here an original proof is used which relies only on [**52**] and gets tighter bounds on the performance.

The theory of recurrent events described in [**52**] describes the behaviour of systems where an event occurs periodically over a number of trials. If this event ε is associated with the event $X_t = 0$ the theory in [**52**] can be used. In [**52**] the event ε is "characterised by the property that, as far as ε is concerned the initial situation repeats itself every time when ε occurs: the trials following

the occurence of ε are a replica of the whole sequence." This is certainly the case for $X_t = 0$ due to the Markov property.

DEFINITION 2.17. The event ε is some event which may either occur at a given sample in a time series X_t. The number of samples of the series between occurrences of ε is an independent and identically distributed variable.

DEFINITION 2.18. The indicator variable $I(X_t)$ is defined as 1 if $X_t = 0$ and 0 otherwise.

DEFINITION 2.19. The expected number of occurrences of ε in n trials is M_n. The value of M_n where the zeroth trial is X_t is given by $M_n(t)$ where

$$M_n(t) = \sum_{i=1}^{\infty} I(X_{t+i}).$$

DEFINITION 2.20. Define N_n as M_n with the restriction that the event ε occurred on the trial before counting. Define $N_n(t)$ as N_n measured where the zeroth trial is at X_t which has the value $X_t = 0$. Note that this definition of N_n is taken from [**52**] and [**154**].

DEFINITION 2.21. Define T_j as one plus the number of trials between the $(j-1)$th and the jth occurrence of ε.

By the conditions of Definition 2.17, the T_j are mutually independent with a common probability distribution. In the case of the chain described,

$$\mathbb{P}[T_j = n] = f_{n-1},$$

for $n > 0$ since when the $(j-1)$th occurrence of ε occurs the chain must be in state zero and the next occurrence must be at its next return to the zero state. The distribution function $F(n)$ of T_j is therefore given by

$$F(n) = \sum_{i=1}^{n} f_{i-1}, \tag{2.23}$$

for the infinite chain.

The results from [**52**] and [**154**] both assume that the distribution function obeys

$$1 - F(n) \sim An^{\gamma}, \tag{2.24}$$

for some postive constant A and some γ. This will now be shown for the specified infinite chain.

From equation (2.23),

$$1 - F(n) = 1 - \sum_{i=1}^{n} f_{i-1} = \sum_{i=n+1}^{\infty} f_{i-1} = \sum_{i=n}^{\infty} f_i.$$

Substituting f_i from (2.20):

$$\begin{aligned} 1 - F(n) &= \left(\frac{1-\pi_0}{\pi_0}\right) \sum_{i=n}^{\infty} \left[i^{-\alpha} - 2(i+1)^{-\alpha} + (i+2)^{-\alpha}\right] \\ &= \left(\frac{1-\pi_0}{\pi_0}\right) \left[n^{-\alpha} - (n+1)^{-\alpha}\right] \\ &= \left(\frac{1-\pi_0}{\pi_0}\right) \frac{(n+1)^{\alpha} - n^{\alpha}}{(n+1)^{\alpha} n^{\alpha}} \\ &= \left(\frac{1-\pi_0}{\pi_0}\right) \frac{(1+1/n)^{\alpha} - 1}{n^{\alpha}(1+1/n)^{\alpha}}. \end{aligned}$$

Expanding $(1 + 1/n)^{\alpha}$ using the binomial theorem gives

$$(1 + 1/n)^{\alpha} = 1 + \alpha/n + O(n^{-2}).$$

Substituting this expression top and bottom gives:

$$\begin{aligned} 1 - F(n) &= \left(\frac{1-\pi_0}{\pi_0}\right) \frac{1 + \alpha/n + O(n^{-2}) - 1}{n^{\alpha}(1 + \alpha/n + O(n^{-2}))} \\ &= \left(\frac{1-\pi_0}{\pi_0}\right) \frac{n^{-\alpha}(\alpha/n + O(n^{-2}))}{(1 + \alpha/n + O(n^{-2}))} \\ &\sim \left(\frac{1-\pi_0}{\pi_0}\right) \alpha n^{-(1+\alpha)}. \end{aligned}$$

This is the form required by equation (2.24) with $\gamma = (1 + \alpha)$ and $A = \alpha(1 - \pi_0)/\pi_0$.

From [**154**, page 6651] [1] if the density function is of the form $F(x) = 1 - An^{-\gamma}$ with $1 < \gamma < 2$ then the autocorrelation function $\rho(n)$ is given by

$$\rho(n) \sim Cn^{-(\gamma-1)},$$

for some positive constant C therefore,

$$\rho(n) \sim Cn^{-\alpha}.$$

This is the form given by Definition 1.21 and therefore the time series Y_t generated by the infinite chain is long-range dependent with LRD parameter α. In the rest of this section, an independent proof will be developed which finds a value for C.

The proof in [**154**] is somewhat technical and relies on results from Fourier analysis in addition to the work in [**52**]. The proof that follows relies only on [**52**] but is not as general as that given in [**154**] since the proof given below works only when the Markov chain is ergodic (which occurs, as has been shown, when $\alpha \in (0, 1)$).

From [**52**, Theorem 10], given that the probability distribution satisifies $1 - F(x) \sim Ax^{-\gamma}$, where A is a positive constant and $1 < \gamma < 2$ then

$$\mathrm{E}[N_n] = \frac{n}{\mu} + \frac{A}{(\gamma-1)(2-\gamma)}\mu^2 n^{2-\gamma} + o(n^{2-\gamma}),$$

where μ is the mean recurrence time of ε, and

$$\mathrm{var}(N_n) \sim \frac{2A}{(2-\gamma)(3-\gamma)\mu^3} n^{3-\gamma}.$$

In the case of the chain under investigation $\gamma = 1 + \alpha$, and $A = \alpha(1 - \pi_0)/\pi_0$. Since the chain is ergodic, from Theorem 2.2, the mean recurrence time of state zero for the infinite chain is $1/\pi_0$. Therefore,

$$\mathrm{var}(N_n) \sim \frac{2\alpha\pi_0^2(1-\pi_0)}{(1-\alpha)(2-\alpha)} n^{2-\alpha}. \tag{2.25}$$

[1] The reference uses α where this thesis uses γ — the change has been made to avoid a clash with α in Definition 1.21.

THEOREM 2.6. As $n \to \infty$ if $\mathrm{E}[N_n] \to \infty$ and the underlying Markov chain is ergodic then

$$\mathrm{E}[N_n] \sim \mathrm{E}[M_n]$$

and

$$\mathrm{var}(N_n) \sim \mathrm{var}(M_n).$$

PROOF. Define T_k as one greater than the number of trials between the $(k-1)$th occurrence of ε and the kth occurrence. The T_k are clearly independent variables (from the definition of ε in Definition 2.17). Define S_k as

$$S_k = \sum_{i=1}^{k} T_i.$$

If k or more events ε occur in the n trials immediately following an event then $T_1 + \cdots + T_k$ must be less than or equal to n. This gives

$$\mathbb{P}[N_n \geq k] = \mathbb{P}[S_k \leq n].$$

Using a similar expression for $\mathbb{P}[N_n \geq k+1]$ gives

$$\mathbb{P}[N_n = k] = \mathbb{P}[S_k \leq n] - \mathbb{P}[S_{k+1} \leq n].$$

However, when considering M_n, there is no restriction that the n trials are immediately following an event. Therefore define S'_k as

$$S'_k = T'_1 + T_2 + T_3 + \cdots + T_k,$$

where $0 < T'_1 \leq T_1$ is the number of trials before the first event occurs.

$$\mathbb{P}[M_n \geq k] = \mathbb{P}[S'_k \leq n] \geq \mathbb{P}[S_k \leq n],$$

and therefore $\mathrm{E}[M_n] \geq \mathrm{E}[N_n]$. However, using the fact that the T_i are independent and identically distributed,

$$\mathbb{P}[N_n \geq k-1] = \mathbb{P}[S_{k-1} \leq n] = \mathbb{P}[S_k - T_1 \leq n] \geq \mathbb{P}[S_k - T_1 + T'_1 \leq n].$$

Therefore,

$$\mathbb{P}[N_n \geq k] \leq \mathbb{P}[M_n \geq k] \leq \mathbb{P}[N_n \geq k-1].$$

Taking expectations gives

$$\mathrm{E}\left[N_n\right] \leq \mathrm{E}\left[M_n\right] \leq \mathrm{E}\left[N_n\right] + 1,$$

which proves that $\mathrm{E}\left[N_n\right] \sim \mathrm{E}\left[M_n\right]$. The derivation for variance follows since, by similar reasoning, $\mathrm{E}\left[N_n^2\right] \sim \mathrm{E}\left[M_n^2\right]$ and $\mathrm{E}\left[N_n\right]^2 \sim \mathrm{E}\left[M_n\right]^2$. □

This theorem allows the substitution of $\mathrm{var}\left(N_n\right)$ from equation (2.25) into the result from Theorem 2.5 gives:

$$\begin{aligned}\rho(n) &\sim \frac{\mathrm{var}\left(N_{n+1}\right) - 2\mathrm{var}\left(N_n\right) + \mathrm{var}\left(N_{n-1}\right)}{2\pi_0(1-\pi_0)} \\ &\sim K(n+1)^{2-\alpha} - 2Kn^{2-\alpha} + K(n-1)^{2-\alpha}, \end{aligned} \tag{2.26}$$

where

$$K = \frac{\alpha\pi_0}{(1-\alpha)(2-\alpha)}.$$

By the binomial theorem,

$$\begin{aligned}&(n+1)^{2-\alpha} - 2n^{2-\alpha} + (n-1)^{2-\alpha} \\ =&n^{2-\alpha}\left[\left(1+\frac{1}{n}\right)^{2-\alpha} - 2 + \left(1-\frac{1}{n}\right)^{2-\alpha}\right] \\ =&n^{2-\alpha}\left[1 + \left(\frac{1}{n}\right)(2-\alpha) + \left(\frac{1}{n}\right)^2\frac{(2-\alpha)(1-\alpha)}{2} - 2 + \right. \\ &\left. 1 - \left(\frac{1}{n}\right)(2-\alpha) + \left(\frac{1}{n}\right)^2\frac{(2-\alpha)(1-\alpha)}{2} + O(n^{-3})\right] \\ =&n^{2-\alpha}\left[\frac{1}{n^2}(2-\alpha)(1-\alpha) + O(n^{-3})\right] \\ =&n^{-\alpha}(2-\alpha)(1-\alpha) + O(n^{-(1+\alpha)}),\end{aligned}$$

Substituting this result into equation (2.26) shows that the ACF of the chain has the form

$$\rho(n) \sim \alpha\pi_0 n^{-\alpha}, \tag{2.27}$$

which was exactly the fall off required to prove the existence of LRD in the series.

2.7. An Algorithm for the Finite Chain

For calculating which state of the Markov chain to move to next, it is useful to be able to calculate certain parameters directly. From equations (2.6) and (2.20) then

$$g_N^N = \frac{1}{N}\sum_{i=N}^{\infty}\frac{1-\pi_0}{\pi_0} i\left[i^{-\alpha} - 2(i+1)^{-\alpha} + (i+2)^{-\alpha}\right].$$

Rearranging this gives

$$g_N^N = \frac{1-\pi_0}{N\pi_0}\Bigg[\sum_{i=N}^{\infty} i(i^{-\alpha}) - 2\sum_{i=N}^{\infty}(i+1)(i+1)^{-\alpha} + 2\sum_{i=N}^{\infty}(i+1)^{-\alpha} + \sum_{i=N}^{\infty}(i+2)(i+2)^{-\alpha} - 2\sum_{i=N}^{\infty}(i+2)^{-\alpha}\Bigg].$$

Cancelling parts of the sums leads to,

$$g_N^N = \frac{1-\pi_0}{N\pi_0}\left[NN^{-\alpha} - (N+1)(N+1)^{-\alpha} + 2(N+1)^{-\alpha}\right],$$

which finally gives

$$g_N^N = \frac{1-\pi_0}{\pi_0}\left[N^{-\alpha} - \frac{(N-1)}{N}(N+1)^{-\alpha}\right]. \tag{2.28}$$

Now, to choose the state which follows the zero state, calculate the probability that the next state is in the range $[j,k]$ where $0 \le j \le k \le N$.

DEFINITION 2.22. Let $G_N(j,k)$ be the probability that if the N state chain is in state 0, then the next state picked is in the range $[j,k]$ where $0 \le j \le k \le N$.

In fact, this calculation is simple in the case $0 < j \le k < N$ since

$$G_N(j,k) = \sum_{i=j}^{k} g_i^N = \sum_{i=j}^{k} f_i = \frac{1-\pi_0}{\pi_0}\sum_{i=j}^{k}\left[i^{-\alpha} - 2(i+1)^{-\alpha} + (i+2)^{-\alpha}\right],$$

which becomes

$$G_N(j,k) = \frac{1-\pi_0}{\pi_0}\left[j^{-\alpha} - (j+1)^{-\alpha} - (k+1)^{-\alpha} + (k+2)^{-\alpha}\right]. \tag{2.29}$$

This, is valid for the range $0 < j \leq k < N$. To calculate $G_N(j, N)$ with $j > 0$ simply use

$$G_N(j, N) = G_N(j, N-1) + g_N^N.$$

Combining equations (2.29) and (2.28) gives

$$G_N(j, N) = \frac{1-\pi_0}{\pi_0}\Bigg[j^{-\alpha} - (j+1)^{-\alpha} - N^{-\alpha} + (N+1)^{-\alpha} + N^{-\alpha} - \frac{(N-1)}{N}(N+1)^{-\alpha}\Bigg].$$

This becomes

$$G_N(j, N) = \frac{1-\pi_0}{\pi_0}\left[j^{-\alpha} - (j+1)^{-\alpha} + \frac{1}{N}(N+1)^{-\alpha}\right].$$

To use the N state finite chain, follow the simple procedure in Table 2.1.

(1) If $X_n > 0$ then $X_{n+1} = X_n - 1$. Exit here.
(2) Choose a new random number R in the range $[0, 1]$.
(3) Set $j = 1$.
(4) If $R < G_N(j, N)$ then the new state is $X_{n+1} = j - 1$. Exit here.
(5) Increase j by 1. If $j > N$ the new state $X_{n+1} = N$. Exit here.
(6) Go to step 4.

TABLE 2.1. Procedure for finding X_{n+1} in the N state finite chain from X_n.

2.8. Calculating States in the Infinite Chain

The same calculations can be done for the infinite chain and an extension allows use of the infinite model in practical computation.

DEFINITION 2.23. Let $F(j, k)$ be the probability that, if the infinite chain is in state zero, then the next state picked is in the range $[j, k]$ where $0 \leq j \leq k$.

In other words, $F(j,k) = \sum_{i=j}^{k} f_i$.

For $j > 0$ and $k < \infty$ this can be shown (in a similar way to the expression for the finite chain) to be

$$F(j,k) = \frac{1-\pi_0}{\pi_0}\left[j^{-\alpha} - (j+1)^{-\alpha} - (k+1)^{-\alpha} + (k+2)^{-\alpha}\right]. \tag{2.30}$$

For $j > 0$ and $k = \infty$

$$F(j,\infty) = \frac{1-\pi_0}{\pi_0}\left[j^{-\alpha} - (j+1)^{-\alpha}\right]. \tag{2.31}$$

For $j = 0$ and $k < \infty$

$$F(0,k) = 1 - F(k+1,\infty),$$

where $F(k+1,\infty)$ can be calculated from the previous equation. The result for $j = 0$ and $k = \infty$ is therefore

$$F(0,\infty) = 1,$$

as would be expected.

To make the infinite chain useful in computation a few subsidiary results are needed. A computer can only generate a random number to a finite precision. Therefore, to simulate an infinite chain a method is needed to choose states using finite precision arithmetic. If X is the first state chosen following state zero then for $0 < k \le i \le j \le l$,

$$\mathbb{P}\left[X \in [i,j] | X \in [k,l]\right] = \frac{\mathbb{P}\left[X \in [i,j] \cap X \in [k,l]\right]}{\mathbb{P}\left[X \in [k,l]\right]},$$

and therefore, since $[i,j] \subseteq [k,l]$,

$$\mathbb{P}\left[X \in [i,j] | X \in [k,l]\right] = \frac{\mathbb{P}\left[X \in [i,j]\right]}{\mathbb{P}\left[X \in [k,l]\right]}.$$

Finally, if $0 < k \le i \le j \le l$ then from (2.30),

$$\mathbb{P}\left[X \in [i,j] | X \in [k,l]\right] = \frac{i^{-\alpha} - (i+1)^{-\alpha} - (j+1)^{-\alpha} + (j+2)^{-\alpha}}{k^{-\alpha} - (k+1)^{-\alpha} - (l+1)^{-\alpha} + (l+2)^{-\alpha}}. \tag{2.32}$$

If $l = \infty$ then the l terms simply vanish from the equation as can be seen in equation (2.31). This gives

$$\mathbb{P}\left[X \in [i,j] | X \in [k,\infty]\right] = \frac{i^{-\alpha} - (i+1)^{-\alpha} - (j+1)^{-\alpha} + (j+2)^{-\alpha}}{k^{-\alpha} - (k+1)^{-\alpha}}. \quad (2.33)$$

The procedure for finding X_{n+1} the first state after some $X_n = 0$ for the infinite chain can be given by Table 2.2.

(1) If $X_n > 0$ then $X_{n+1} = X_n - 1$. Exit here.
(2) Explicitly calculate $F(j, \infty)$ for values of $j \leq N$ where N is some small integer. Use the procedure for the finite state model to find a value for X_{n+1} if $X_{n+1} < N$.
(3) Generate a new random number R in the range $[0, 1]$.
(4) Calculate $\mathbb{P}\left[X_{n+1} \in [N, 2N-1] | X_{n+1} \in [N, \infty]\right]$ from equation (2.33). If R is less than or equal to this probability then X_{n+1} is in the required range. Otherwise go to step six.
(5) If X_{n+1} is in the required range then refine down by generating a new random number and seeing if X_{n+1} is in the range $[N, (3/2)N]$. Continue refining by a binary search (with a new random number each time) until X_{n+1} is found. Exit here.
(6) Increase the value of N to $2N$ and go to step 3.

TABLE 2.2. A procedure for finding X_{n+1} from X_n in the infinite chain.

2.9. Tests on Implementions of This Model

The model specified in Table 2.2 was run for various test scenarios to test repeatability and ability to model LRD with a given mean and Hurst parameter. Several simulation procedures are used, involving the finite and the infinite chain. In general, the procedure is to generate a large number of

values of $Y_i : i \in \{1, 2, \ldots, N\}$ (the binary time series generated with the rule $Y_i = 1$ if $X_i > 0$ and $Y_i = 0$ if $X_i = 0$). This series is aggregated over a scale m to form a series $Z_i : i \in \{1, 2, \ldots, N/m\}$ where $Z_i = \sum_{j=im}^{(i+1)m-1} Y_j$. All the series shown here are simulated with a mean of 0.5 (that is $\pi_0 = 0.5$). The tests shown in this section are aggregated over a scale $m = 100$.

Figures 2.2, 2.3 and 2.4 show sample traces generated from the infinite chain with differing Hurst parameters. As described, each point in this plot is the sum of one hundred binary samples generated from the infinite Markov model. Traces generated from the finite chain are indistinguishable to the naked eye. Figure 2.2 has the lowest Hurst parameter and, the majority of the time the trace appears to stay around the mean level of 50. For Figures 2.3 and 2.4 the trace seems to have more peaks and remain at the highest value (100) for longer. It should be noticed that because of the nature of the chain, long high periods are common, long low periods are absent from the plot (none of the plots reach 0). These plots should be viewed in conjunction with the sample mean calculations in Section 1.1.3.1. The sample mean converges more slowly as the Hurst parameter increases. This can be seen, as the Hurst parameter in the plots increases the plots cover more points on the y-axis showing that the variance on the sample mean estimate (represented by each point on the plot) has increased.

Table 2.3 shows the sample mean for several realisations of different parameters of the infinite chain with an actual mean of 50. As can be seen, and as expected, the smaller number of points and the higher Hurst values have a worse convergence to the actual mean.

The next three plots, Figures 2.5, 2.6 and 2.7 show the ACF for three realisations of ten thousand points (each of which is an aggregation of $m = 100$ points). Note that the plots are labelled $R(k)$ (where $R(k)$ is the ACF at lag k) versus k. The three figures show $\pi_0 = 0.5$ and Hurst parameters of $H = 0.625$, $H = 0.75$ and $H = 0.875$. The plots are on a log scale, therefore, from (2.27),

$$\log(\rho(k)) = \log(\alpha\pi_0) - \alpha \log(k) + o(1),$$

and therefore, a log-log plot of n versus $log(k)$ should be a straight line. Breaks in the logscale plots are where the ACF value was negative and therefore no log could be computed. It should be noted, however, that when k is small then the $o(1)$ term may dominate and as k increases, the estimate of $\rho(k)$ becomes less reliable even when the sample size is as high as one million points. This explains why the straight line deviates wildly in these realisations of ten thousand points. It also explains why the three lines do not lie directly on top of each other. As can be seen in the next three plots, Figures 2.8, 2.9 and 2.10 the autocorrelation function remains straighter for longer with a sample size of one million points. By comparison, Figures 2.11 and 2.12 show ten thousand and one million point samples respectively compared for three different Hurst parameters.

Figures 2.13, 2.14 and 2.15 show the fit against the theoretical line. It is clear from all three figures that the slope (which represents the rate of exponential decay) is at least approximately correct for all three Hurst parameters investigated. However, it also seems clear that for high Hurst parameters, while the prediction of the slope is correct the absolute value is wrong. It is not clear that the sample ACF is an unbiased estimator for a long-range dependent process described and it is clear that the ACF estimate does not converge quickly as the number of points in the series increases. This matter clearly merits further investigation since the discrepency between the theory and the experimental realisation is clear.

For contrast Figure 2.16 shows the results for a finite chain model with a variety of different numbers of states. As the number of states increases, the returns in accuracy diminish and the accuracy of real number storage in the machine becomes an issue. This kind of problem will beset a large number of LRD generating mechanisms in practical implementations. Breaks in the lines are particularly notable when the number of states is low.

Hurst parameter	Points	Run 1 Mean	Run 2 Mean	Run 3 Mean
0.625	10,000	49.9252	49.5322	49.7305
0.75	10,000	50.2001	50.3154	50.476
0.875	10,000	47.4101	46.9322	49.569
0.625	1,000,000	50.053464	50.00757	49.998927
0.75	1,000,000	49.847889	50.115351	49.835945
0.875	1,000,000	49.999512	48.590166	49.489746

TABLE 2.3. Means for several realisations of the infinite chain process

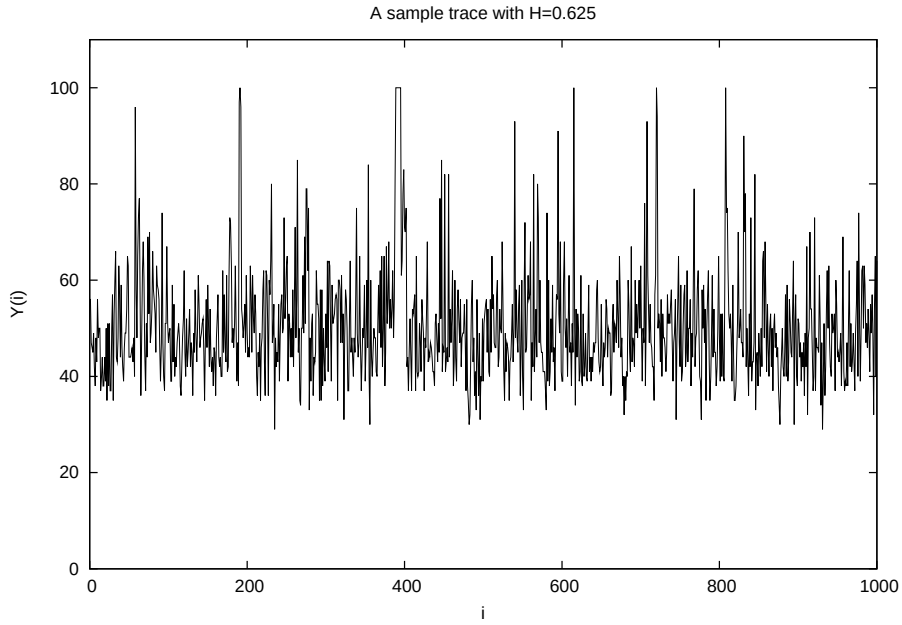

FIGURE 2.2. A sample path of 1000 points generated from the infinite chain with $H = 0.625$, $\pi_0 = 0.5$ and $m = 100$.

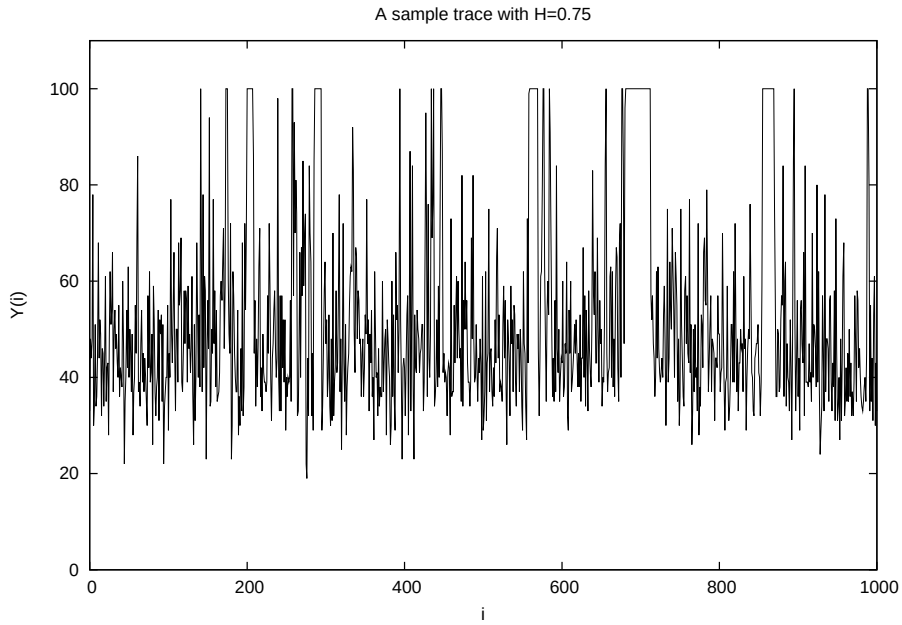

FIGURE 2.3. A sample path of 1000 points generated from the infinite chain with $H = 0.75$, $\pi_0 = 0.5$ and $m = 100$.

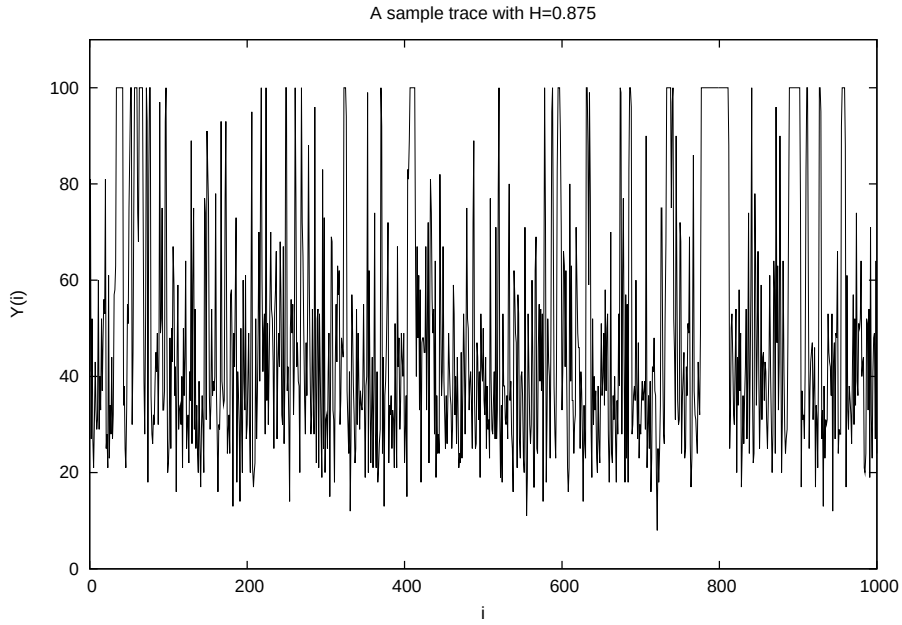

FIGURE 2.4. A sample path of 1000 points generated from the infinite chain with $H = 0.875$, $\pi_0 = 0.5$ and $m = 100$.

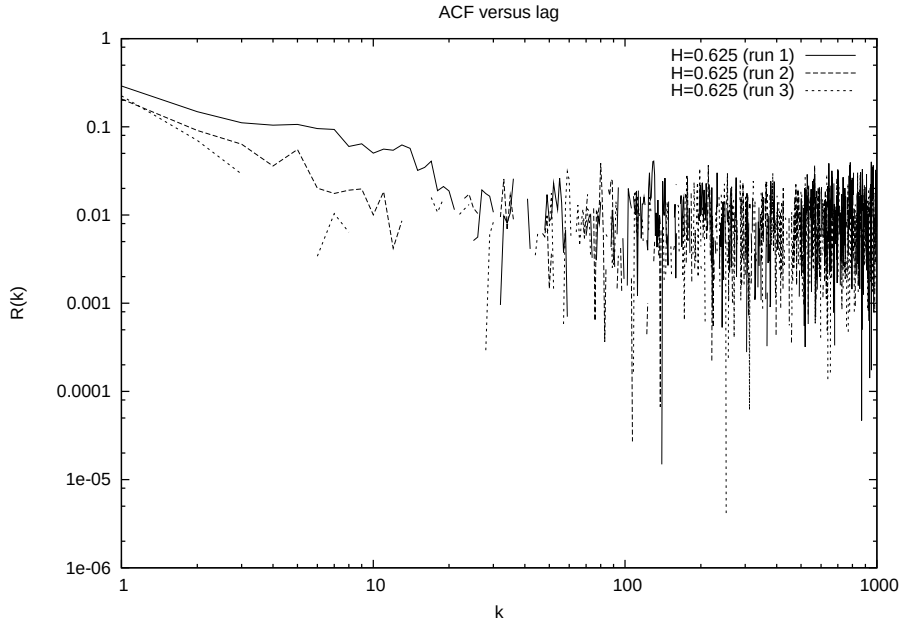

FIGURE 2.5. ACF of three runs of 10,000 points generated from the infinite chain with $H = 0.625$, $\pi_0 = 0.5$ and $m = 100$.

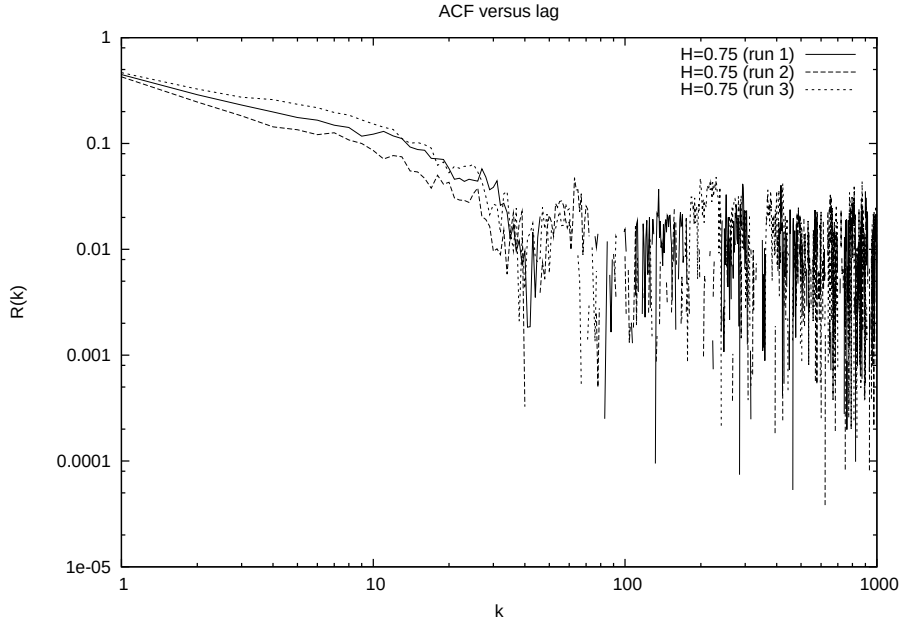

FIGURE 2.6. ACF of three runs of 10,000 points generated from the infinite chain with $H = 0.75$, $\pi_0 = 0.5$ and $m = 100$.

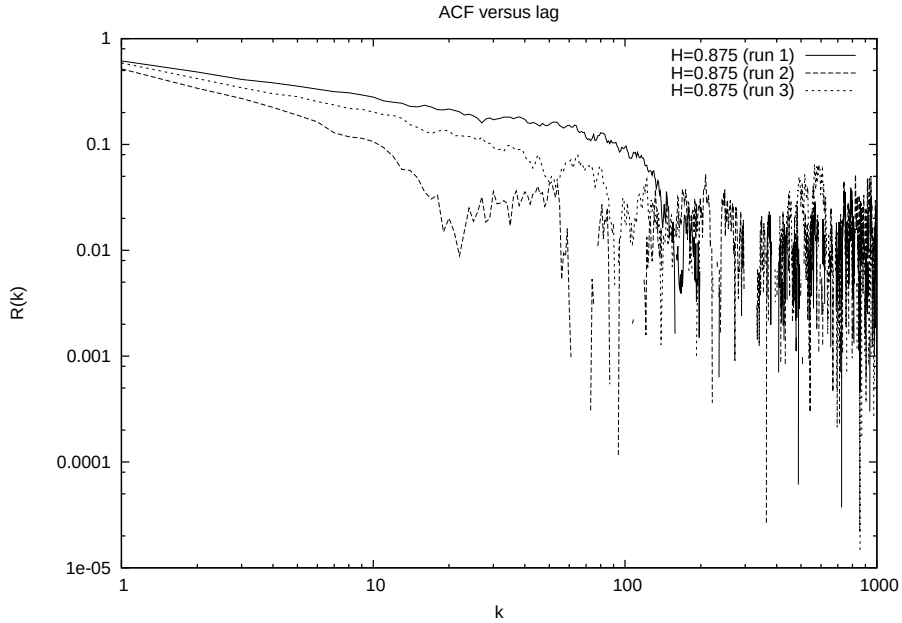

FIGURE 2.7. ACF of three runs of 10,000 points generated from the infinite chain with $H = 0.875$, $\pi_0 = 0.5$ and $m = 100$.

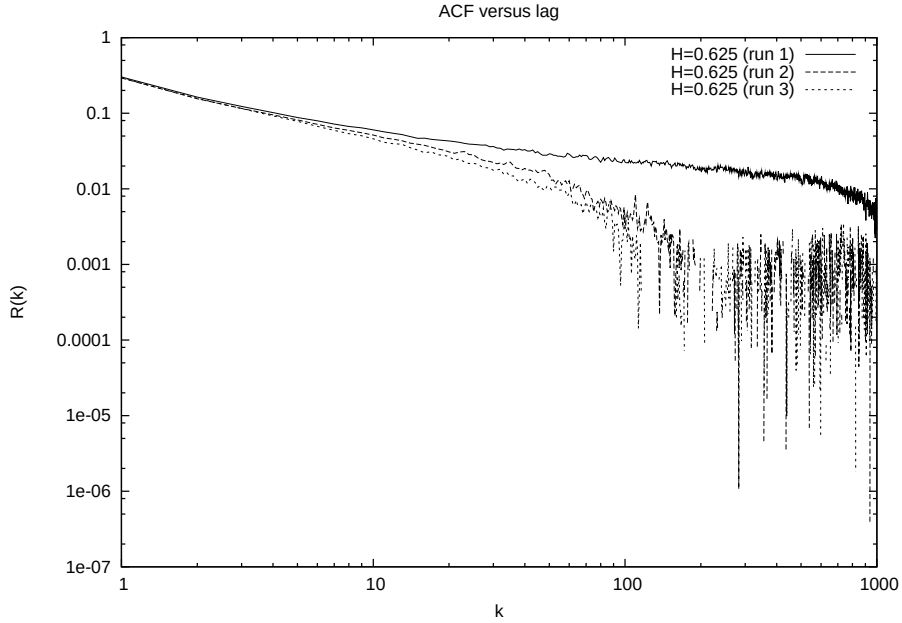

FIGURE 2.8. ACF of three runs of 1,000,000 points generated from the infinite chain with $H = 0.625$, $\pi_0 = 0.5$ and $m = 100$.

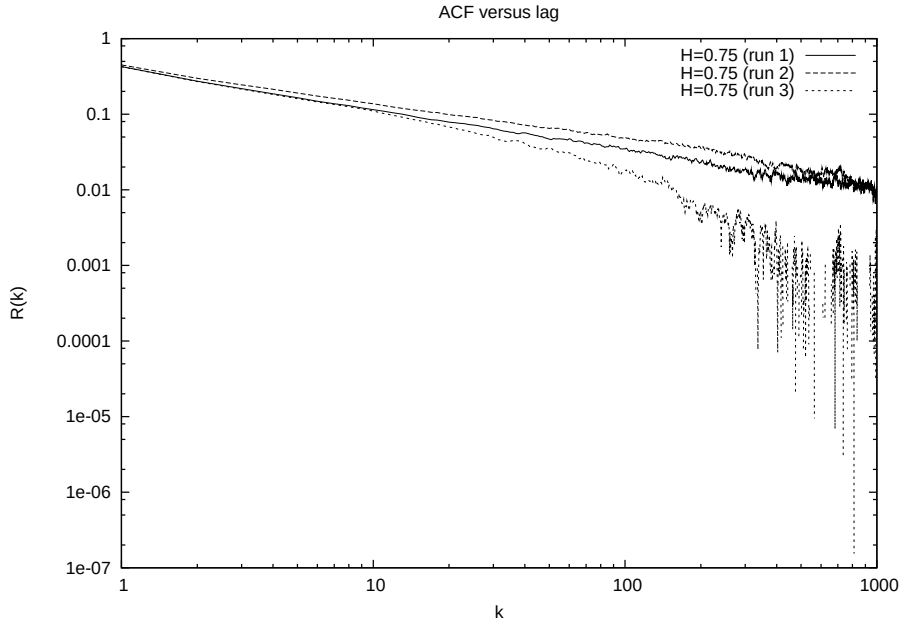

FIGURE 2.9. ACF of three runs of 1,000,000 points generated from the infinite chain with $H = 0.75$, $\pi_0 = 0.5$ and $m = 100$.

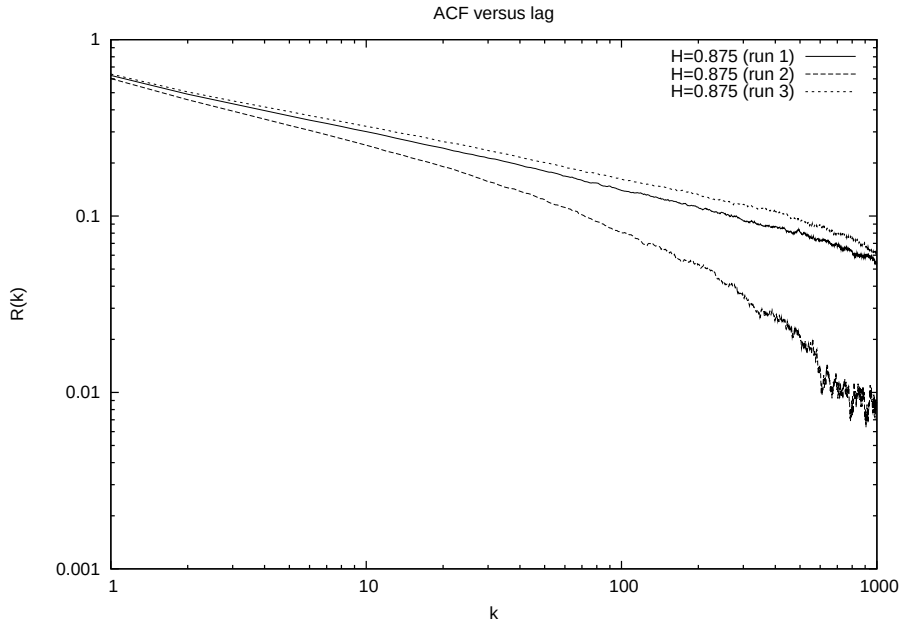

FIGURE 2.10. ACF of three runs of 1,000,000 points generated from the infinite chain with $H = 0.875$, $\pi_0 = 0.5$ and $m = 100$.

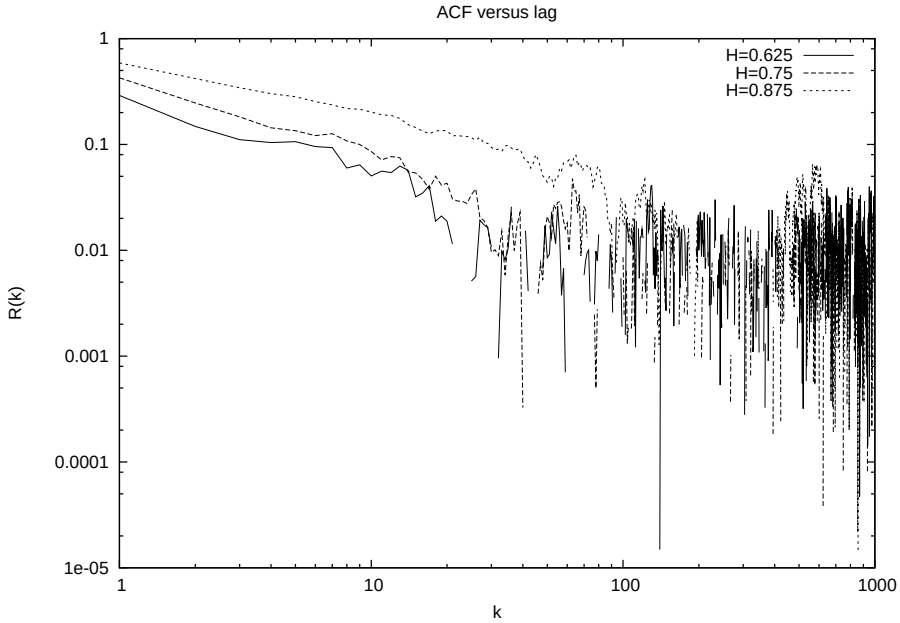

FIGURE 2.11. ACF of three runs of 1,000,000 points generated from the infinite chain with H values and $\pi_0 = 0.5$ and $m = 100$.

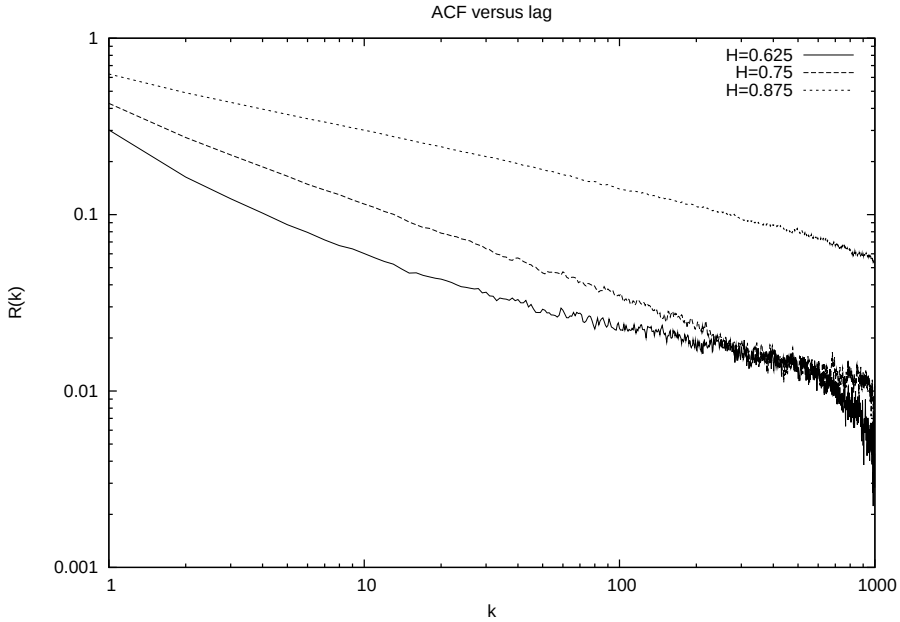

FIGURE 2.12. ACF of three runs of 1,000,000 points generated from the infinite chain with H values and $\pi_0 = 0.5$ and $m = 100$.

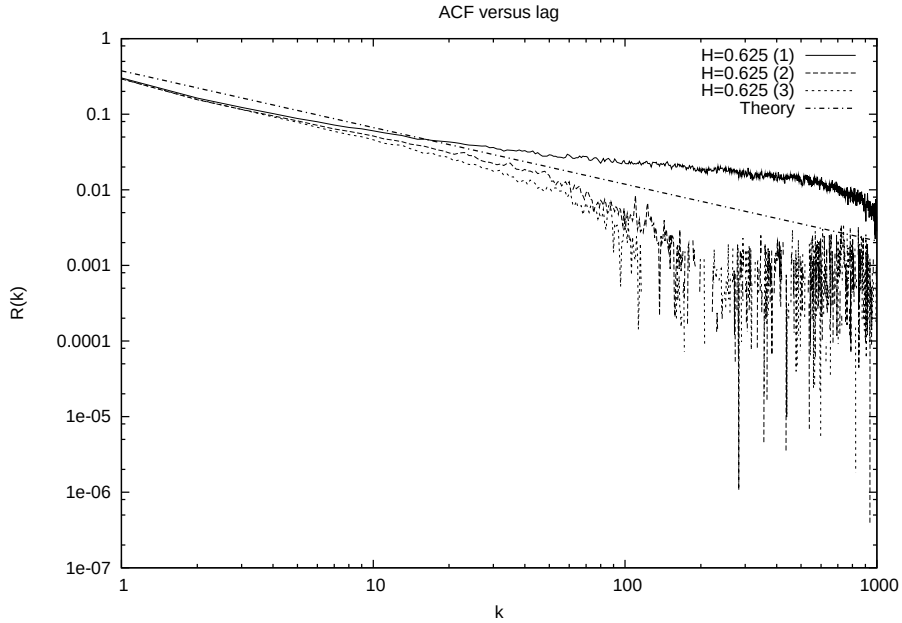

FIGURE 2.13. ACF of three runs of 1,000,000 points generated from the infinite chain with $H = 0.625$ and $\pi_0 = 0.5$ and $m = 100$ with theoretical line.

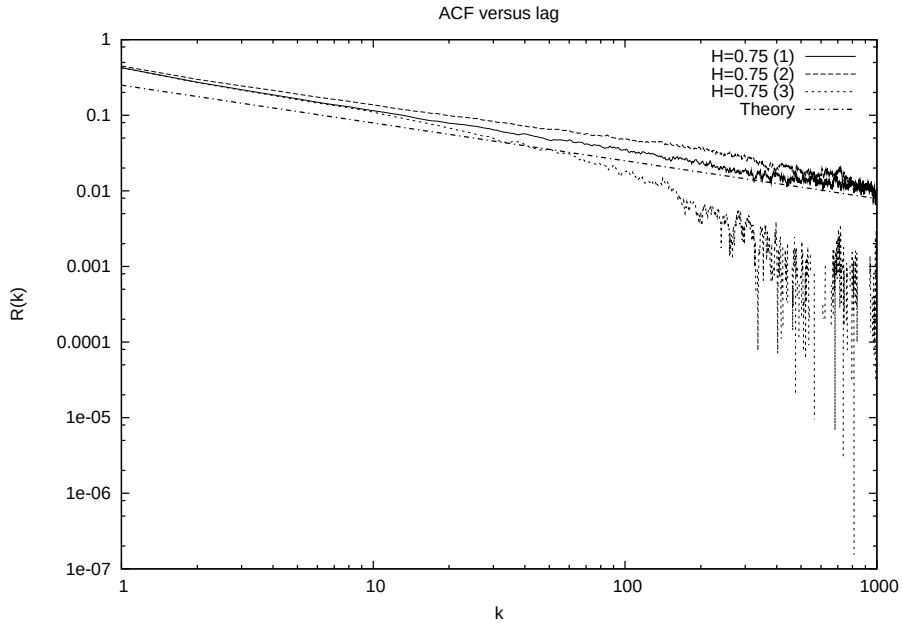

FIGURE 2.14. ACF of three runs of 1,000,000 points generated from the infinite chain with $H = 0.75$ and $\pi_0 = 0.5$ and $m = 100$ with theoretical line.

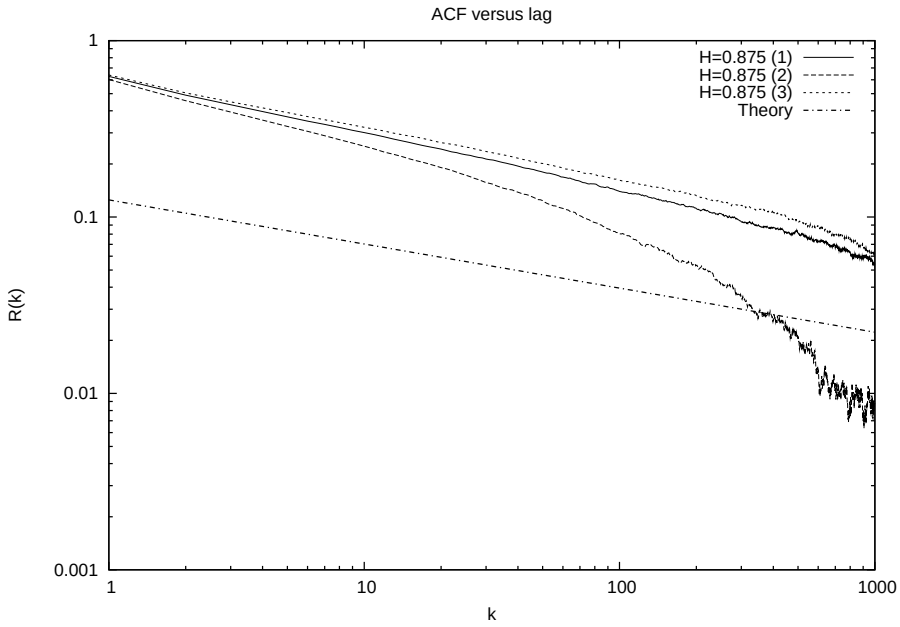

FIGURE 2.15. ACF of three runs of 1,000,000 points generated from the infinite chain with $H = 0.875$ and $\pi_0 = 0.5$ and $m = 100$ with theoretical line.

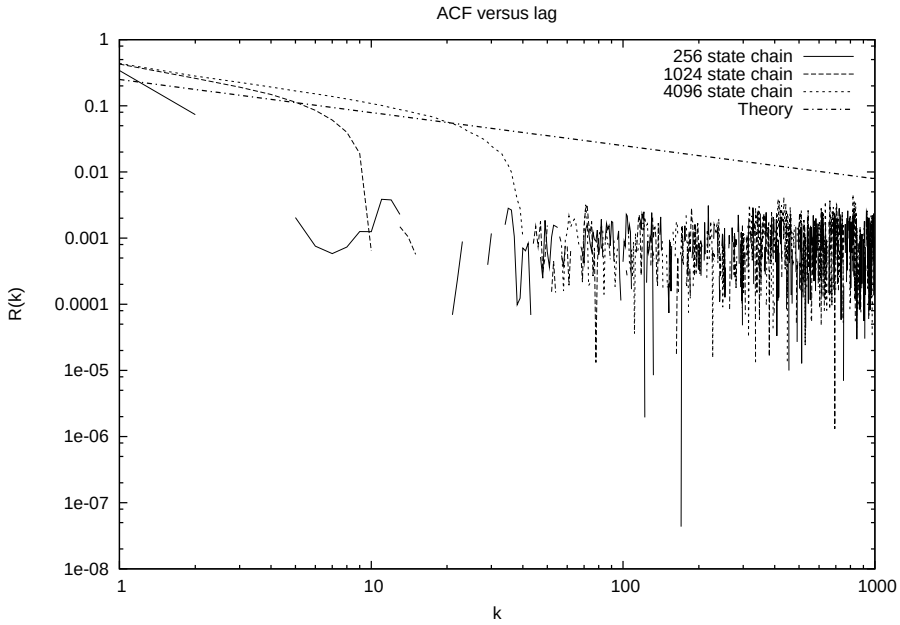

FIGURE 2.16. ACF from the finite chain with 256, 1024 and 4096 states. Three runs with 1,000,000 Points with $H = 0.75$, $\pi_0 = 0.5$ and $m = 100$ with theoretical line

2.9.1. Comparison With Other Models. In this section, Fractional Gaussian Noise (Section 1.2) and iterated chaotic maps (Section 1.2) are compared with the Markov method for generating traffic previously discussed. In each case, the traffic is generated with a known Hurst parameter and each generation method with each Hurst parameter is run three times.

In each case, a million sample points are generated. In the case of the Markov method and the iterated maps method each point is generated by aggregating a hundred points as discussed previously. The three methods were implemented in the C programming language. To generate one million points took approximately 55 seconds for the Markov method, 60 seconds for the iterated maps method and 6 seconds for the fractional Gaussian noise method. However, it is debatable whether this is a fair comparison since the first two methods could be considered to be generating a hundred million points and aggregating into groups of one hundred. No C code to generate FARIMA based data was available and the R code available took 188 seconds to generate only a hundred thousand points.

The Hurst parameter is estimated using various of the measuring techniques discussed in Section 1.3 to check the match between theory and practice. The estimators used are the R/S method and a modification of this which automatically selects the lag ranges to look at, the aggregated variance, the periodogram, local Whittle and wavelet based estimation.

Source	H	R/S	Mod. R/S	Agg. Var.	Period-ogram	Local Whit.	Wave-lets
FGN	0.625	0.637	0.624	0.623	0.626	0.639	0.635
FGN	0.625	0.632	0.624	0.622	0.624	0.638	0.635
FGN	0.625	0.645	0.633	0.620	0.622	0.638	0.635
FGN	0.75	0.728	0.738	0.741	0.747	0.774	0.767
FGN	0.75	0.741	0.736	0.749	0.755	0.776	0.769
FGN	0.75	0.694	0.719	0.741	0.754	0.774	0.768
FGN	0.875	0.784	0.837	0.858	0.877	0.908	0.897
FGN	0.875	0.750	0.823	0.850	0.876	0.908	0.897
FGN	0.875	0.747	0.835	0.860	0.876	0.908	0.898
It. map	0.625	0.635	0.590	0.604	0.630	0.719	0.706
It. map	0.625	0.608	0.595	0.604	0.627	0.716	0.703
It. map	0.625	0.637	0.594	0.610	0.637	0.718	0.707
It. map	0.75	0.828	0.666	0.717	0.746	0.813	0.800
It. map	0.75	0.725	0.650	0.712	0.739	0.813	0.801
It. map	0.75	0.678	0.694	0.765	0.768	0.814	0.803
It. map	0.875	0.703	0.779	0.851	0.876	0.925	0.910
It. map	0.875	0.779	0.802	0.854	0.877	0.924	0.910
It. map	0.875	0.846	0.817	0.861	0.874	0.925	0.912
Markov	0.625	0.526	0.597	0.611	0.621	0.703	0.691
Markov	0.625	0.593	0.645	0.700	0.684	0.710	0.702
Markov	0.625	0.632	0.603	0.646	0.650	0.707	0.698
Markov	0.75	0.663	0.684	0.744	0.760	0.793	0.784
Markov	0.75	0.670	0.667	0.751	0.759	0.793	0.783
Markov	0.75	0.671	0.671	0.724	0.736	0.786	0.776
Markov	0.875	0.724	0.732	0.816	0.848	0.884	0.873
Markov	0.875	0.757	0.754	0.830	0.859	0.885	0.874
Markov	0.875	0.656	0.781	0.852	0.866	0.885	0.875

TABLE 2.4. Hurst Parameter Estimates on Simulated Data.

Table 2.4 shows the result of various estimators for six estimators (grouped into three time-based and three frequency based [2]) applied to traffic from three different generating models. It would naturally be expected that the FGN model is the easiest to estimate and this shows in the results in Table 2.4. All the estimators were relatively close to correct with the possible exception of the R/S plot on traffic with a Hurst parameter of 0.875 where the underestimate of H was quite severe.

[2]Frequency based is arguable for wavelets which provide both time and frequency information.

Estimates on the iterated chaotic map traffic were not so successful. The raw R/S plot proved inconsistent and had a hard time estimating higher hurst parameters. It should be noted, for example, that for $H = 0.75$ estimates varied from 0.678 to 0.828. The performance for $H = 0.875$ was similarly bad. The modified R/S parameter was better in that it was more stable across runs but tended to overestimate. Local Whittle and wavelets tended to overestimate the Hurst parameter. It should also be noted that the true result was regularly outside the 95% confidence intervals for the wavelet estimator.

Estimates for the Markov based method were, in many ways, similar to the iterated map method. If anything, the results from the estimators are slightly closer to the theory and this is particularly notable for the wavelet and local Whittle case. The evidence provided by the estimators is hard to interpret. However, it can certainly be said that the results for the Markov method are as close as the results for the iterated map method.

Generally, considering the estimators themselves, the R/S method seemed unreliable (and this agrees with theory which shows it to be a biased estimator with poor convergence). The local whittle and wavelets methods which have better theoretical backing seem to have a better agreement with theory but it is worrying that the true Hurst parameter for the data lay outside 95% confidence for the wavelet estimator in many cases.

2.10. Simulation Results on a Simple Network

The simulation results in this section were obtained with the help of Dr. Yann Golanski. The software used was the ns-2 simulation [**113**] which models individual packets in a network using approximations of the protocols used in the Internet.

The topology chosen for testing in ns was to represent an aggregation of traffic from different sources being fed into a larger router. In this case, eight LRD sources generated from an infinite Markov chain feed into a single *shaper* router. The shaper then feeds into a drop tail router which sends the packets

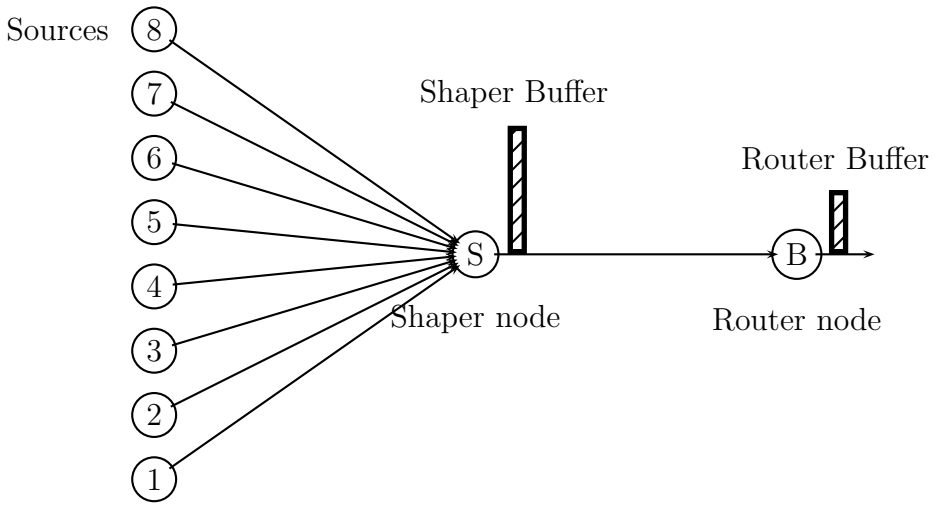

FIGURE 2.17. The simulation topology used.

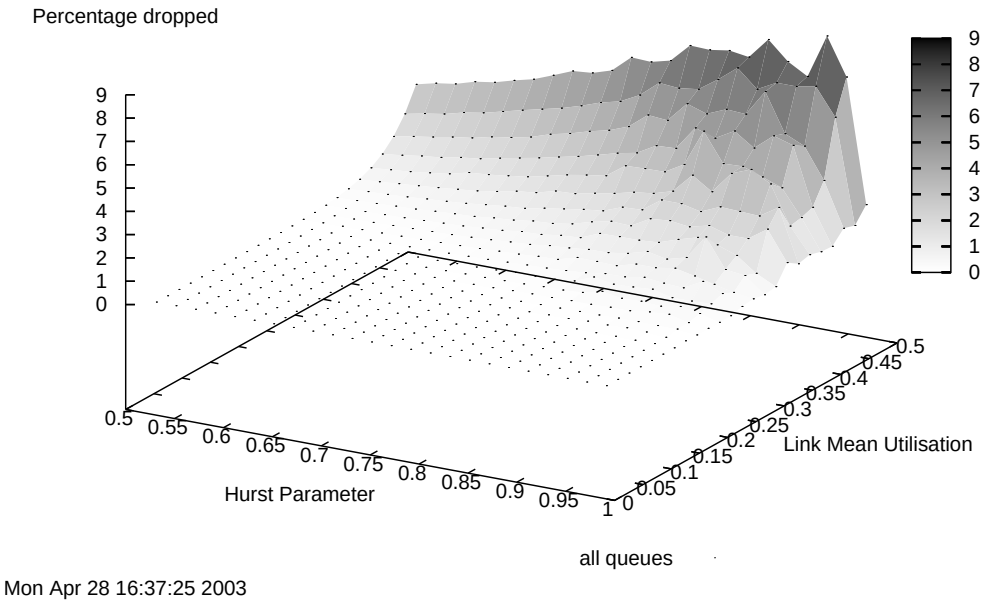

FIGURE 2.18. Drop tail results: percentage packet loss over all queues.

into the sink. The simulation is shown in Figure 2.17. The buffer at the shaper can hold twenty packets and the one at the router can hold only ten. The buffers all operate as drop-tail buffers, that is when they are full then newly arriving packets are dropped.

All links between the sources and the shaper have a capacity of 256kb/s, the link between the shaper and the router has a capcity of 2048kb/s and the router to the sink is half of that. The sources are all sending packets of size 256b at a maximum rate of 256kb/s. The rates were chosen such that if all the links send at exactly half their capacity then the router which carries traffic to the sink will be exactly full. In this setting, therefore, if the mean $(1 - \pi_0)$ is exactly 0.5 then the router will suffer a huge amount of packet loss unless the traffic arrives with a completely flat distribution. The system was chosen so that means between 0 and 0.5 could be tested in the model with 0.5 representing the case of an extremely overloaded system. The shaper maximum output capacity was chosen such that even if all links sent at maximum capacity, the shaper itself would never overflow (as is the case in the example discussed here).

Figure 2.18 shows a three dimensional plot of percentage of packet loss versus the mean utilisation and the Hurst parameter for the topology discussed. The z-axis is the percentage of packet loss over the entire network. Naturally, in this example, all the loss occurred at the router node and none at the shaper node (since the shaper node was simply outputting at its maximum rate and performing no shaping). The figure shows clearly that packet loss increases as the mean traffic level increases. Of course this is as expected. Similarly, the packet loss increases as the Hurst parameter increases, at least up to a point. It seems that, in these simulations at least, extremely large Hurst parameters actually reduce the amount of packet loss. The reason for this is unclear and merits further investiation.

2.11. Discussion

The Markov chain based approach has a number of advantages both theoretical and computational over other LRD generation mechanisms. Firstly, the method is extremely easy to implement and quick to run. A run of ten million iterations of the chain took only 2.4 seconds on a 2.2GHz PC running Free BSD. This makes it an attractive prospect for modelling. A number of mechanisms for generating traffic (fGn, fBm and FARIMA) require the user to specify in advance how many points are wanted and then the entire time series is generated at once. This can be a problem when a simulation does not know in advance how many points of data will be wanted. In addition these generation mechanisms are typically slower. The widely-used iterated map based approach [**4**] has issues related to double precision arithmetic. While generally, the precision is good enough for most purposes, the correlations in that model necessarily fall off eventually due to the finite precision of comptuer arithmetic. In the Markov model the limitation to accuracy is much less of a problem. Indeed the only limitation is that the model as described above is incapable of calculating series which contain bursts of ones of the order of `INT_MAX` (the largest integer which can be stored by the compiler used — approximately two billion in C++ on a typical modern compiler). Using a language with arbitrary precision integers would avoid this problem. However, this problem would only be expected to be important if the number of packets generated by a single stream was many orders of magnitude greater than two billion and is, therefore, vanishingly unlikely to occur in computational experiments.

In addition to computational advantages, the analytical advantages of the model may be considerable. Considerable work has already been done on the queuing performance of Markov moderated processes and it is hoped that existing theorems can be brought to bear to obtain queuing results without the need for simulation. This would greatly enhance the theoretical underpinning of the work as well as obviate the need for the many complexities involved with computational simulation of weakly convergent statistical processes.

CHAPTER 3

Driver Route and Departure Time Choice in Road Networks

This chapter has been developed, in part, from work presented at the Universities Transport Studies Group conference as a paper jointly prepared with Dr. Richard Batley of the Institute for Transport Studies, Leeds [**8**]. I am grateful to Dr. Batley for allowing me to adapt parts of that paper which were largely his work. In an adapted form, this gave the basis for Sections 3.7 and 3.8 and part of Section 3.9 in this chapter.

3.1. Introduction

Choice of route and departure time are considered by many researchers to be the two most important driver responses to a change in network conditions. According to an influential report: "...overall, the two responses — changing route and changing journey time — seem to be the most universal" [**23**, page 28]. This conclusion follows a review of evidence from ninety case studies where road capacity was reduced. This chapter reviews on-street evidence about route choice and departure time choice and then considers the models used to capture this phenomenon mathematically.

The behavioural evidence is further broken down into ambient variability and variability which occurs in response to a network change. It has often been noted that, even in situations where no changes occur in a network, drivers modify their behaviour and, as noted by [**45**], many drivers choose inefficient routes. When considering a network change, the time-scale of adjustment is another consideration. After an intervention in the network has occurred, over what time scale should choice effects be considered? Reference has been made in the literature to a "settling down" period.

Inevitably much of the evidence falls between camps and provides evidence on route and departure time choice aspects or on both ambient and responsive variability. This means that some reports will be mentioned in more than one section within this chapter. There is a considerable body of literature based upon laboratory and survey-based studies of route and departure time choice where no actual on-street measurements are taken. While this is a rich area of research with much published literature, it is less relevant to what follows in Chapter 5 than reviews of on-street evidence. While such studies are briefly mentioned in Section 3.9 no systematic attempt has been made to review them here.

In reviewing the modelling, the groundwork is laid with a short review of the theory of equilibrium modelling beginning with Wardrop's influential equilibrium condition [**155**]. Following this, deterministic and stochastic approaches to assignment are distinguished and attention is given to stochastic loading models. The assumption underlying many modelling approaches is that of "rational behaviour" on the part of drivers. Advancements in theoretical modelling have sometimes failed to become part of established practice in scheme assessment.

Including this introduction, this chapter is split into nine sections. Section 3.2 reports the on-street evidence on route choice, distinguishing between ambient variability in route choice and changes in route due to network changes. Section 3.3 reports the on-street evidence on departure time choice. Section 3.4 considers the time-scales of importance for choice effects. Section 3.5 summarises the modelling challenge when trying to capture these choice effects. Section 3.6 briefly considers the theoretical underpinnings of equilibrium modelling. Section 3.7 considers the modelling of route choice, distinguishing between deterministic and stochastic user equilibrium models. Section 3.8 reviews the modelling of departure time choice. Section 3.9 describes the practical difficulties inherent in these modelling approaches. This chapter is

complemented by Chapter 5 which analyses data from two surveys in York in considerable detail.

3.2. On-Street Evidence on Route Choice

In this section, evidence about driver route choice is reviewed, firstly with consideration to studies where network conditions were not subject to major change (that is to say the only changes to conditions on the network were due to weather, day of the week and the usual changes in demand from day to day). Following this, consideration is given to studies where network conditions were subject to major change (where, either due to accident or intervention, an identifiable major alteration to network conditions was made, for example a major re-timing of traffic signals, a road closure or a bridge closure).

3.2.1. Ambient Variability in Route Choice. It is hard to find good on-street studies of ambient variability in route choice. One reason for this is that it is an extremely difficult phenomenon to study. Even if observations show the same individual making a journey between the same origin and the same final destination by a different route it is hard to prove this is not due to some intermediate destination. Of the studies reported here, none were motivated by a desire to study the problem directly and most provide evidence which is tangential at best. While route choice is commonly cited as one of the two most common choice elements, it seems that ambient variability in this important choice dimension is rarely studied for its own sake.

A useful online review of this subject from the perspective of using Global Positioning System (GPS) data is provided by [**116**]. In this report, variability is split into *inter-personal* and *intra-personal* variability. The former arises due to socio-economic and behavioural differences between individuals and the latter is due to day of the week and other external effects not related to the drivers themselves. The author analyses data collected from seven small surveys each of a small number of individuals (each survey was of between sixteen and thirty-two individuals). The data was collected in Lexington, Kentucky.

One conclusion from analysis of the data is: "The percentage of individuals in each sample who exhibit the same characteristic across all days...is extremely small... [often] zero."

The Uppsala Household Travel Survey was a Swedish study of repetition in travel which was widely reported by Huff and Hanson in the 1980s ([**67**], [**68**], [**69**], [**70**], [**83**] and [**84**]). The study monitored all travel from home for 149 individuals over a thirty-five day period. An important conclusion of their work is: "observations taken for a single day in the travel history of an individual are not likely to be representative of the range of daily travel patterns exhibited by that person over a more extended time period, and we are led to reject the view that travel is highly routinized in the restricted sense that every weekday is assumed to look much like every other weekday" [**83**, page 108].

GPS data is a potentially valuable tool for the study of route choice. A report on GPS data from single vehicles from one hundred households (216 drivers) over a one-week period is given by [**87**]. They report: "the path chosen on a trip most often differs considerably from the shortest time path across the network" [**87**, page 1] and also that "travelers habitually follow the same path for the same trip" [**87**, page 12]. (The shortest path time accounted for errors associated with random delays at traffic signals and delays due to congestion). This suggests that the ambient variability in route choice may be low but also that the assumption that users are rationally choosing shortest paths may be a questionable one.

A variety of studies in Hertfordshire are examined in [**45**]. These studies looked at how drivers choose either a *rat-run* or a main route on a network (where the *rat-run* is defined as the usage of a minor road route as an alternative to a major road route — in some of the cases studied the rat-run was both shorter and quicker than the main route). In summary, they state that "travel time is the single most important criterion affecting driver route choice in networks where there is a viable alternative to the main route" [**45**, page 408].

Their observations also indicate that drivers are willing to travel an increased distance if it will reduce their travel time provided "the distance is not doubled or the alternative tortuous" [**45**, page 408]. The work was accompanied by questionnaire data about how drivers perceived factors affecting route choice. The authors give the following equation for the percentage of drivers using a particular rat-run route,

$$\mathrm{TRS} = 9.14 - 22.27(\mathrm{TTR}) + 30.98(\mathrm{DIR}) + 26.65(\mathrm{SPR}) - 0.089(\mathrm{TID}),$$

where TRS is the percentage of drivers using the rat-run route, TTR is the travel time ratio (rat-run / main road), DIR is the distance ratio (rat-run / main road), SPR is speed ratio (rat-run / main road) and TID is the travel time difference in seconds (main road − rat-run). Further details of the work are reported in [**44**]. It is hard to square the authors' statement that travel time was the most important factor with the coefficients given in this equation.

A report on large licence plate surveys undertaken in Leeds is described in [**19**]. One major conclusion of this report was that collecting licence plates in this way can be extremely unreliable. They report that they must "assume a 15% increase in the number of matches" [**19**, page 387] due to missed matches from incorrectly recorded data. Table 3.1 shows their data. From these data, route choice changes cannot be distinguished from decisions not to travel. Indeed, even when travel times are noted as being different, it is impossible to distinguish if this was genuinely due to a departure time choice decision or if this was due to congestion interfering with an unchanged departure choice. However, it is clear from this data that the day-to-day variability in the actual composition of the traffic is extremely large. On almost all days, even allowing for the author's suggested increase in matches due to misread data, the majority of travelling vehicles in the rush hour are not seen in the next rush hour.

To sum up this evidence, it would seem that a typical recurrence rate for traffic during the rush hour on weekdays is something between thirty and fifty

Day 1 Time period (beginning time)	% match in same period day 2	% match in same or adjacent period	% match in any period
7:15	23	36	35
7:30	24	26	30
7:45	15	23	28
8:00	19	28	32
8:15	24	38	45
8:30	19	27	38
8:45	11	21	23
9:00	9	15	22
9:15	6	10	8
9:30	5	8	0

TABLE 3.1. Match rates at different times within the peak from [**19**]. All figures should be increased by 10-20% to allow for misreading.

percent. While travel time is widely acknowledged to be the most important element in the route chosen, other elements such as distance and perceived *directness* of route are important. In general, it would seem that the variability in the typical morning peak, which is traditionally seen by modellers as the most stable part of the travelling day, is much greater than has been imagined.

3.2.2. Route Choice Responses to Network Changes. Data collected in Edmonton monitoring the closure of the Kinnaird Bridge is analysed in [**139**]. The Kinnaird Bridge was totally closed to traffic and rerouting was an inevitable driver response. However, it is clear from the study results that drivers who directly used the bridge were not the only ones to make a route choice change as a result. Drivers who were affected by congestion as a result of the closure made route choice responses to avoid the "knock-on" congestion effects.

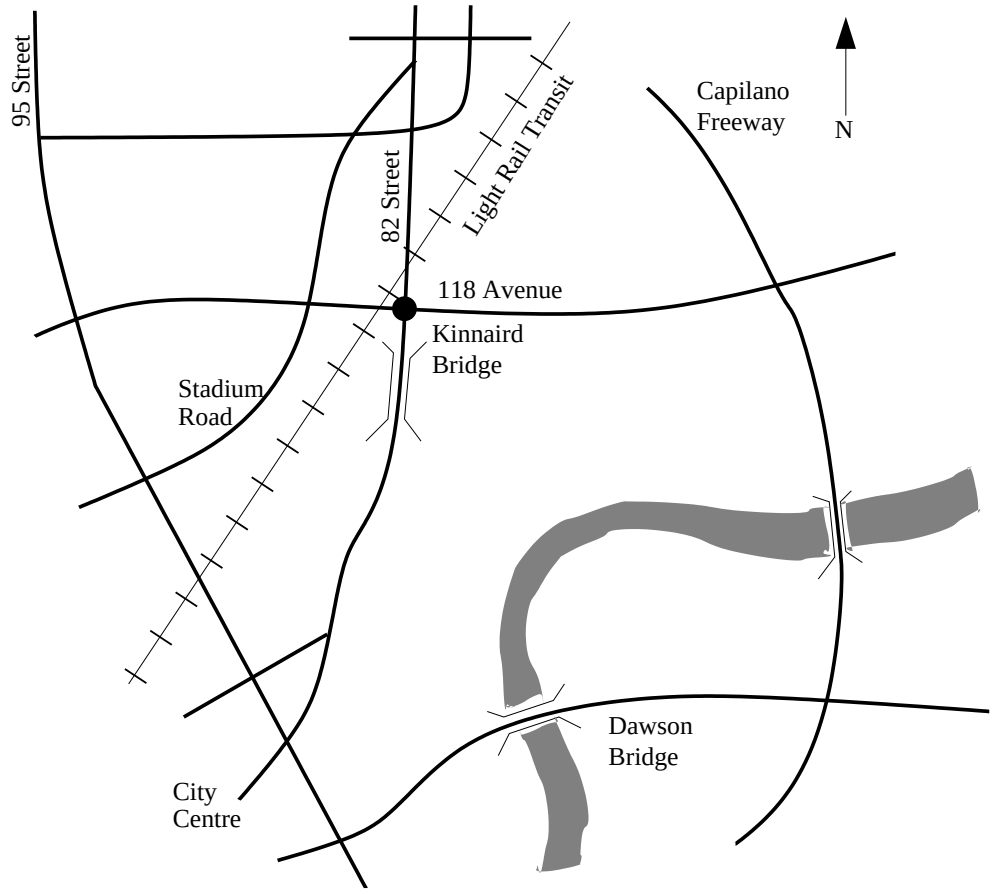

FIGURE 3.1. Kinnaird Bridge closure — area map adapted from [**139**].

Figures 3.1, 3.2 and 3.3 are schematic diagrams adapted from the figures in [**139**]. On these diagrams wider arrows indicate heavier flow (the indicated flows are all between two hundred and eight hundred vehicles per hour). The colours of the flows are consistent between the before and after figures and correspond to the rerouting of drivers in response to the closure. The flows on the diagrams are those from the morning rush hour. The route change can be clearly seen when comparing Figures 3.2 and 3.3. Particularly interesting is the rerouting of a traffic stream which was not actually using the closed bridge. In the before diagram, the black arrow showing traffic moving down Stadium Road is not using the closed bridge, but in the after-situation at least 25% of this traffic has rerouted to 95 Street. While this effect is not unexpected it is certainly good to have experimental confirmation of it.

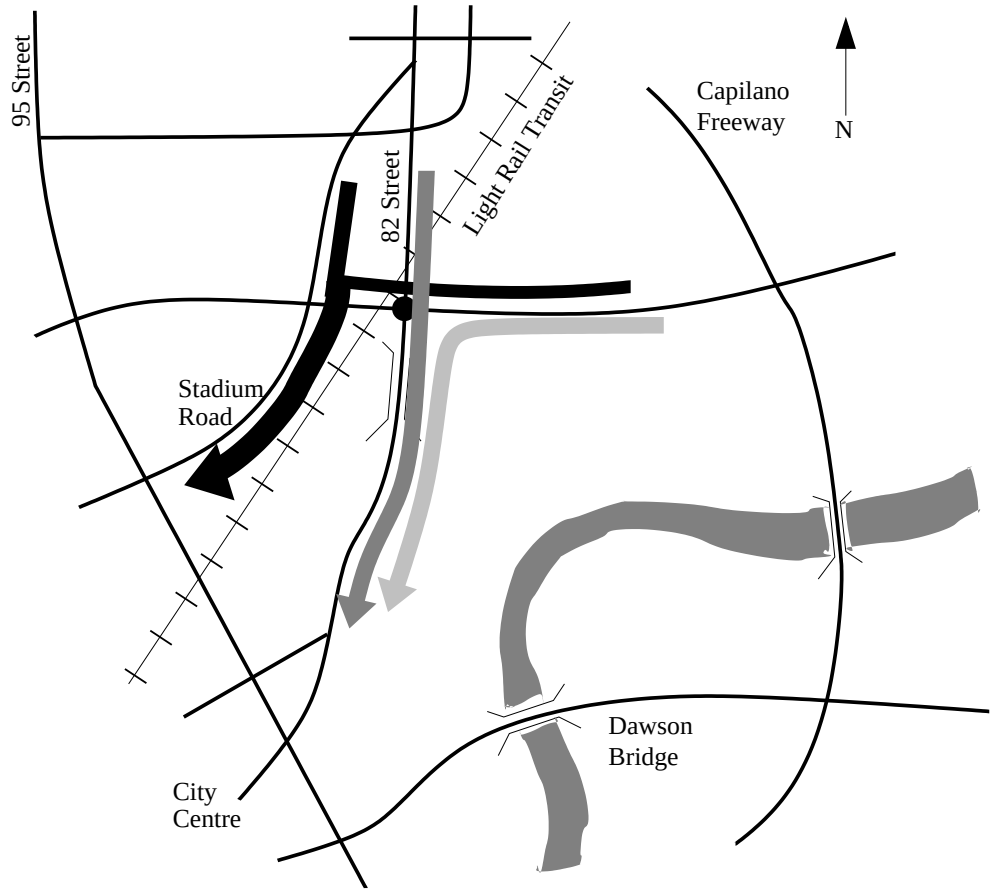

FIGURE 3.2. Kinnaird Bridge closure — before flows adapted from [**139**].

On-street evidence from a number of bus priority schemes implemented in the UK is reported in [**38**]. One conclusion is that "A feature of many schemes is that traffic tends to divert from the priority route if drivers perceive that their journey may be delayed along certain sections of the route. This is not a problem if traffic diverts to routes suitable and capable of absorbing the extra demand... however the diversion of traffic through residential areas or along other routes unsuitable for additional car traffic... should be discouraged on both environmental and safety grounds". A main conclusion of the report was that route choice adjustment, as a result of the capacity reallocation due to bus lanes, was a major driver response and further that scheme assessment should account for this.

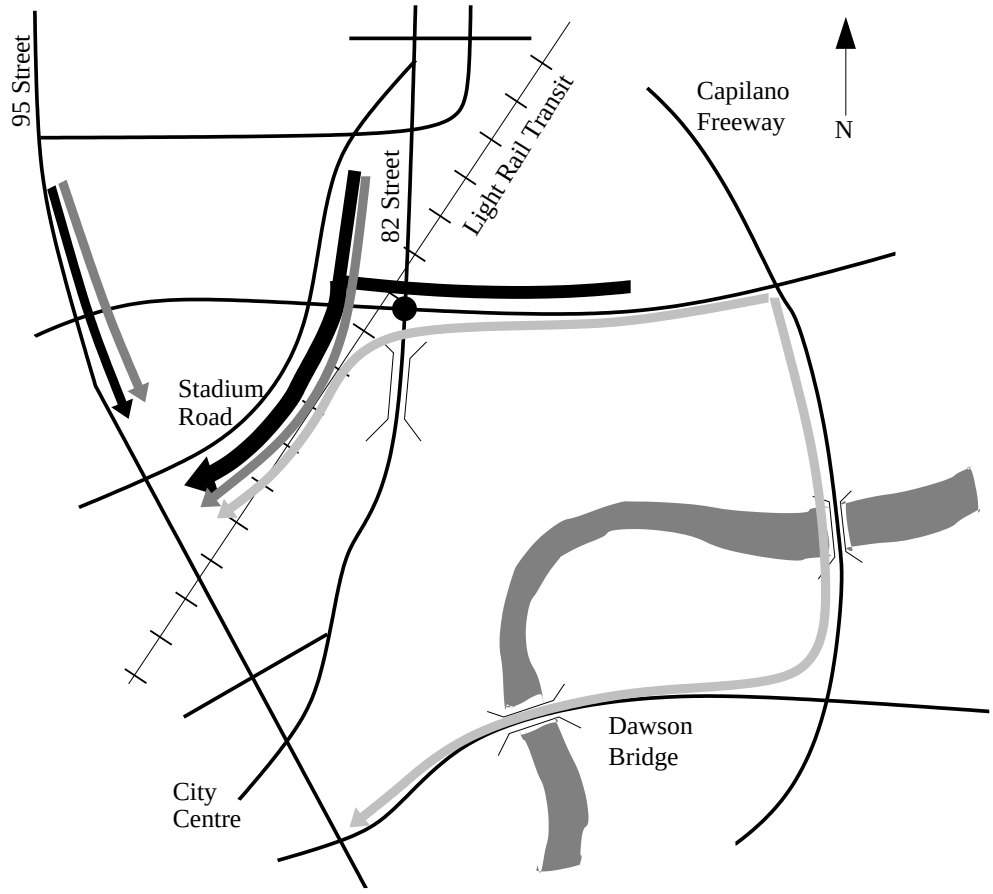

FIGURE 3.3. Kinnaird Bridge closure — after flows adapted from [**139**].

The MUSIC (Management of traffic USIng flow Control) project studied the effects of introducing new signal control policies in three European cities. The signal control policies chosen were designed specifically with route choice in mind. Computer simulation was performed with the aim of assessing the on-street results of the signal timing changes designed as part of the project. The project final report [**34**] states that at all of the three demonstration sites "models tended to overestimate the amount to which drivers would reroute". However, models based on the assumption that no driver rerouting would take place as a result of the signal timing changes were found to be less accurate than models based on the assumption that drivers would reroute completely to an equilibrium. Before and after studies measured the changes in vehicle flows

arising from changes to signal timings. While other causes for the flow changes cannot be ruled out, changes in flow levels of up to sixteen percent were found between the before and after cases. This would seem to indicate that some degree of driver rerouting is taking place (it is hard to imagine that a signal re-timing could cause such a large change in demand). More information on the MUSIC project can be found in [**32**], [**33**] and [**34**].

Again, finding good evidence of route choice in the literature was problematic. Though driver route choice due to network changes was often mentioned, it was hard to find concrete evidence on the subject. While some studies mentioned a belief that route choice had occurred as a result of a network change, few had studied it explicitly. It seems that this important choice aspect is not well-studied empirically.

3.3. On-Street Evidence on Departure Time Choice

It is perhaps useful to distinguish between two different types of departure time choice before discussing the subject in detail. Evidence on the subject often makes the distinction between small departure time shifts (of the order of five minutes to an hour) and larger departure time shifts that move the journey into an uncongested part of the day. These two effects, which are inevitably blurred (it is not clear what counts as a "small" departure time shift), can be the product of fundamentally different constraints upon the journey. For most commuters a decision to set off ten minutes earlier to avoid the traffic is very different to a decision to make their journey at mid-day instead of during the morning peak. Naturally, we would expect the former decision type to be the more common. In the literature this is often referred to as *micro time-shifting* to distinguish it from more radical changes in journey time.

Another important issue when discussing departure time choice is distinguishing a departure time shift from an involuntary change in time caused by delays elsewhere on the route. If a licence plate survey at a particular point

records that the same drivers are, on average, arriving at that point five minutes later, then this could be indicative of a departure time shift on the part of the drivers. Alternatively, it could indicate a five minute delay on an earlier part of the route. When it is considered that driver departure time shifts are often made in response to delays then the problem becomes a difficult one to resolve. Evidence is often found of the phenomenon known as *peak-spreading* — this, obviously, refers to the idea that the peak traffic period begins earlier, ends later or both. This could be one result of departure time choices by drivers.

Characteristic	% Same all all days	% Same all but one day	% Same all but two days	Total
Three weekday Sample ($N = 25$)				
Total trips	8.0	40.0	—	48.0
Non-work trips	12.0	40.0	—	52.0
Dep. time from home	44.0	40.0	—	84.0
Final arrival at home	72.0	24.0	—	96.0
Four weekday Sample ($N = 32$)				
Total trips	3.1	6.3	40.6	50.0
Non-work trips	3.1	18.8	43.8	63.6
Dep. time from home	34.4	37.5	9.4	81.3
Final arrival at home	59.4	40.6	0.0	100.0
Five weekday Sample ($N = 24$)				
Total trips	0.0	4.1	16.7	20.8
Non-work trips	0.0	4.1	16.7	20.8
Dep. time from home	8.3	41.6	33.3	83.2
Final arrival at home	50.0	37.5	8.3	95.8

TABLE 3.2. Selected Data showing ambient variability in weekday data from [**116**].

Table 3.2 shows selected data from a GPS study of user travel behaviour [**116**]. The study shows the percentage of surveyed individuals who exhibit the same behaviour across multiple surveyed days. Departure and arrival times are considered to be equal if they are within twenty percent of the median (where the value of travel times are expressed in minutes past midnight). This is a somewhat curious choice since it means that the final arrival time at home is considered to be equal within a much larger range (since, by this measure, the median final arrival time is larger than the departure time from home and therefore the permissible range for an arrival at home is much larger). This masks the behaviour which would perhaps be expected that departure times from home might be expected to be more consistent than arrival times back at home (it might be argued that more drivers are expected to be at work at a regular time that are expected to leave at a regular time given the reality of overtime and working late). Note also that the "same on all but two days" column is blank for the three day study (since the measure is meaningless on this study).

The range allowed on departure times is extremely generous (a driver departing from home at a median time of 8:00am would be counted as having left the house at the "same time" for departure times from 6:24am to 9:36am). This shows that departure times vary a great deal from day to day.

From the previously mentioned Kinnaird Bridge closure study [**139**], the authors conclude that when comparing two days from the before-period, "60% of drivers travelled at the same time (+/- 5 minutes) every day during uncongested conditions". However, when comparing one day from the before-period with one day from the after-period, only twenty percent of drivers kept the same travel time during the congested peak period. It is, however, unclear whether these results are caused by drivers making a decision to change their departure time or by drivers keeping the same departure time and their journey being delayed by the increased congestion. It should also be noted that it is not clear from the report whether the statement suggests that of all drivers

observed on one day, sixty percent of them were seen on the second day at a similar time or, of all drivers who are seen on both days sixty percent of them were seen at a similar time. The second interpretation is consistent with the time adjustment which can be inferred from Table 3.1.

The collapse of the Tasman Bridge, Hobart, Tasmania, is reported in [**96**]. The bridge was destroyed in an accident involving an ore carrier. Amongst the many effects observed by the authors was an effect on peak-spreading: "the morning peak in 1974 was 7–9am, but in 1975 and 1976, this had extended to 6:30–9am".

The closure of Lendal Bridge in York in 1978 is reported in [**40**]. The bridge was closed for six months to all traffic apart from buses, cyclists and pedestrians. In surveys, fifteen percent of drivers said that they had changed the time at which they made their journey by more than ten minutes. This is, perhaps, curious since elsewhere in the paper it is suggested that the average change in journey time was low except for in the morning peak and even then it was only 2.8 minutes. It is unclear if the high percentage of drivers changing their journey time had averted the worst effects of congestion, if the drivers were simply over-reacting to perceived congestion or if the drivers were over-exaggerating their time-shifting.

Many authors report peak-spreading as a response to increased congestion, but offer little in the way of evidence. Such studies are not included here since spreading may be merely a result of a change in travel time with the departure time remaining constant.

3.4. Time-Scales of Importance for Choice Effects

An important question about route and departure time choice arising from a change to a network is how long the effects of the change take to stabilise. According to [**23**] in the short term (defined as "say the second week"): "It is the common experience that, after an adjustment period, traffic alters to take account of the new conditions. Reference to a 'settling down' period has

been made... Following the Kinnaird Bridge closure, flows were estimated to stabilise in about three weeks".

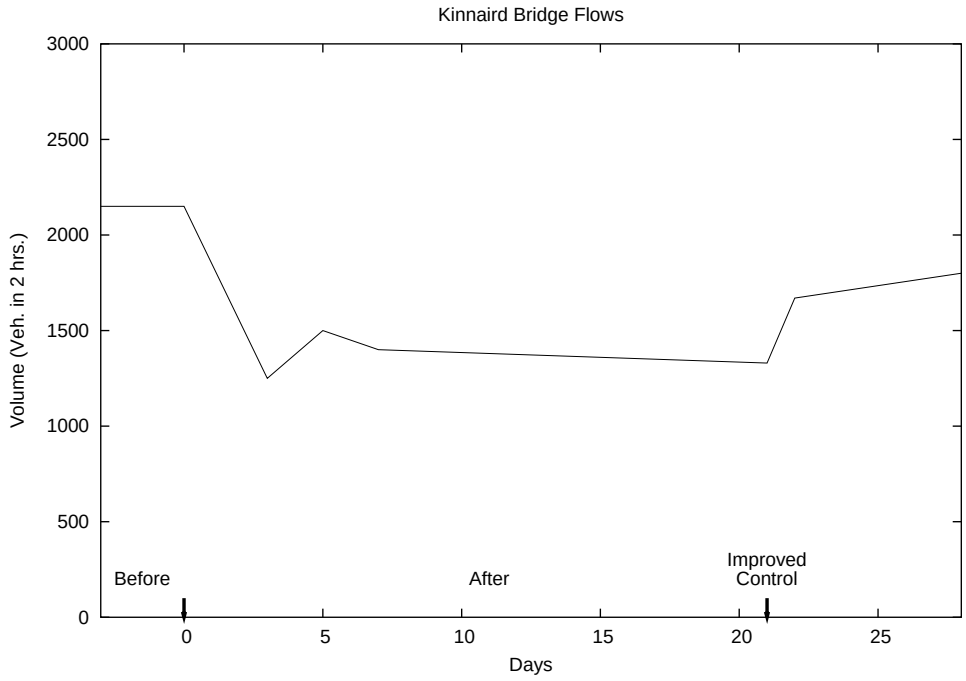

FIGURE 3.4. Development of volume equilibrium at the critical location near the Kinnaird Bridge closure (Recreated from [**139**]).

Figure 3.4 is of flows on the eastern approach to the intersection of 112 Avenue and 82 Street (see Figure 3.1). It appears to show that the most significant changes in flow occur in the first week after the closure although it appears that there is a small but steady downward trend in the graph for the following two weeks (after which time a second alteration to the network takes place). The authors state "Following the closure of Kinnaird Bridge, severe congestion developed in the immediate vicinity of the detour... Subsequent to the initial congestion in the network, however, drivers responded, over a period of two weeks, by altering their travel behaviour through the area" [**139**, page 378].

It should be noted that it is unclear from the original reference whether the days in Figure 3.4 include weekend days (since it would be expected that flows

would differ significantly on weekends). Also, while the authors claim that the response took place over a period of two weeks, close examination of Figure 3.4 above (reproduced from their paper) seems to show that the flow is still reducing slightly on day twenty-one and the introduction of improved control sets up another change which continues to the end of the survey period. The flows never seem to quite stabilise (though it should be noted that the flows shown are unusually similar from day to day). In fact, it is unclear from the paper exactly how many days have been surveyed.

The MUSIC project draws a slightly different conclusion. In the city of Thessaloniki, 128 traffic signal timings were changed in an attempt to reduce congestion and public transport queues in the city, at least partly by accounting for driver rerouting. The after studies took place six weeks after the final scheme was implemented. The project final report [**34**] notes that "A long time period after the implementation of the new traffic signal timing plans is necessary in order to allow rerouting and attainment of a new traffic equilibrium... it was considered that drivers had not fully settled into their new routes by the end of the study period". The study period mentioned was six weeks, compared to the three weeks estimated for flows to stabilise following the Kinnaird Bridge closure. Similar results were reported for a change of signal timings in the city of Porto. Perhaps the reason for this difference is that, in the case of the Kinnaird Bridge, the change was physical, easy to assess and located at a single point in the network whereas, in the case of Porto and Thessaloniki, the changes were harder for drivers to assess and located at a number of points in the transport network.

From the limited evidence available (very few reports could be found which gave evidence on the length of time taken to establish a new equilibrium) it would seem the agreement is that, as common sense would suggest, the most extreme effects of a network change are on the first day afterwards. The first week shows the major changes and then a more gradual settling down occurs

over the next few weeks but the duration of this phase is uncertain and is probably dependent on the exact nature of the change to the network.

3.5. The Modelling Challenge

The challenge faced in modelling route and departure time choice is considerable. Even in the simple situation where we assume that the origin (home) and the destination (work) are both fixed and the mode is fixed as private transport then it remains for the driver to choose a departure time and a path through the network. The route choice problem is a particularly problematic one — in reality the choice set of all physically feasible routes is large but it seems certain that the decision-maker will only consider a subset of such choices although it is hard to know by what criterion such a subset is chosen. Furthermore, it is uncertain what factors influence the decision maker when he or she is making the choice. In the departure time choice problem, it is clear that the problem must (in simulation approaches at least) be converted somehow from a continuous to a discrete problem (since it would be impossible to simulate all the points on an interval). The problem also involves finding an acceptable range of departure times for an individual. It is also clear that the two problems are somewhat inter-related (a route which is optimal in the peak may no longer be optimal in the off-peak) but it is not clear which decision (route or departure time choice) should be made first or if both should somehow be assessed simultaneously.

3.6. The Theory of Equilibrium Modelling

This section provides a brief description of research into equilibrium theory with some theoretical details given. The general formulation of traffic problems can be stated in many ways. This discussion largely follows the work of Smith [**135**] and describes a user-equilibrium formulation in a static network. For a good review of research in the area see [**157**]. This section will restrict the work discussed to that most relevant to route choice or *assignment modelling* (which models how drivers desiring travel from an origin to a destination are

assigned to a route on the network). There is a large literature on *demand modelling* (which models how desire for travel translates into a specific demand for a driver to travel between a particular origin and a particular destination). The problem of demand modelling is not covered here.

Consider the network as a directed graph $G = (N, A)$ where N are nodes (junctions) and A are arcs (roads). The arcs are ordered pairs (a, b) where $a, b \in N$. An arc (a, b) represents a road from node a to node b. The number of nodes is n and the number of arcs is m.

For $x, y \in N$ then the sequence of ordered pairs,

$$(x, x_1), (x_1, x_2), \ldots, (x_{k-2}, x_{k-1}), (x_{k-1}, y),$$

is a *route* or *chain* (to use the terms of graph theory) from the origin x to the destination y. All the pairs (a, b) in the above must be members of A.

Define $\mathcal{R}(x, y)$ as the set of all possible paths or *routes* from between the *origin-destination pair* (O-D pair) (x, y). This is the set of all chains as defined above which do not contain the same $(a, b) \in A$ twice. In a network with a finite set of nodes, then $\mathcal{R}(x, y)$ must clearly be a finite set.

The set of all routes between any O-D pair is given by

$$\mathcal{R} = \bigcup_{(x,y)\in N\times N} \mathcal{R}(x, y).$$

The number of members of $\mathcal{R}$ is M.

An arc in A will be represented by A_i and a route in $\mathcal{R}$ will be represented by R_r. The vector $\mathbf{f} = (f_1, f_2, \ldots, f_m)$ is a *link flow distribution* [1] vector or simply *link flow* vector. Each of the $f_i \geq 0$ represent the flow on the arc A_i. The vector $\mathbf{F} = (F_1, F_2, \ldots, F_M)$ where $F_r \geq 0$ is the *route flow distribution* vector or simply *route flow* vector. Each of the F_r represent the flow on the route $R_r \in \mathcal{R}$.

In a similar manner, the vector $\mathbf{c}$ is the *link cost distribution* vector and $\mathbf{C}$ is the *route cost distribution* vector. With these vectors c_i represents the cost

[1] This terminology is due to Smith in [**135**] and is not connected with distributions in the statistics sense — link flow distributions need not total to a given constant sum.

of travelling along the link A_i and C_r represents the cost of travelling along the route R_r. It should be noted that $c_i > 0$ and $C_r > 0$.

Using these definitions, the total cost on a network is given by

$$\sum_{r=1}^{M} C_r F_r = \mathbf{C} \cdot \mathbf{F}, \tag{3.1}$$

where the $\mathbf{C} \cdot \mathbf{F}$ is the scalar product. Alternatively, the sum over all links could be considered.

$$\sum_{i=1}^{m} c_i f_i = \mathbf{c} \cdot \mathbf{f}. \tag{3.2}$$

It should be noted that a given route flow vector $\mathbf{F}$ implies a unique link flow vector $\mathbf{f}$. However, a link flow vector $\mathbf{f}$ may arise from a number of distinct route flow vectors $\mathbf{F}$.

The *link-route* incidence matrix (a_{ir}) defines the connection between the links A and the routes $\mathcal{R}$.

$$a_{ir} = \begin{cases} 1 & \text{if link } A_i \text{ is part of route } R_r \\ 0 & \text{otherwise.} \end{cases}$$

This matrix can be used to formulate the correspondence between routes and links and to convert between $\mathbf{f}$ and $\mathbf{F}$ or $\mathbf{c}$ and $\mathbf{C}$.

$$f_i = \sum_{r=1}^{M} a_{ir} F_r$$
$$C_r = \sum_{i=1}^{m} a_{ir} c_i,$$

where the first equation can be stated as "the flow on a link is the sum of all the flows on routes which include that link" and the second equation can be stated as "the cost of traversing a route is the sum of all the links which are part of that route."

It is useful now to introduce the notion of a *cost-flow function* $\mathbf{c} : \mathbb{R}_+^m \mapsto \mathbb{R}_+^m$ which relates the flow on the network to the cost of traversing a given link.

$$\mathbf{c}(\mathbf{f}) = (c_1(\mathbf{f}), c_2(\mathbf{f}), \ldots, c_m(\mathbf{f})),$$

where $c_i(\mathbf{f})$ is the cost of traversing link A_i given the traffic distribution $\mathbf{f}$. The reason that $c_i(\mathbf{f})$ is a function of $\mathbf{f}$ rather than f_i is to include junction interactions. (For example, the cost of traversing a link may be greatly influenced if the link ends in a give-way junction and an opposing link at that junction has a high flow.)

The corresponding function for routes $\mathbf{C} : \mathbb{R}_+^M \mapsto \mathbb{R}_+^M$ is also useful.

$$\mathbf{C}(\mathbf{F}) = (C_1(\mathbf{F}), C_2(\mathbf{F}), \ldots, C_M(\mathbf{F})).$$

A famous early paper in road traffic research is [**155**] which discusses a wide range of subjects including optimising signals and speed-flow relations. The paper is, perhaps, most famous for introducing Wardrop's equilibrium condition. In fact, the paper suggests two possible equilibrium conditions. "(1) The journey times on all routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route. (2) The average journey time is a minimum... The first criterion is quite a likely one in practice." [**155**, page 345]. The equilibria described by these two equilibrium principles are now known as User Equilibrium (UE) and System Optimum Equilibrium or Wardrop's First and Second Equilibrium Principles. The UE principle is of most importance here.

Using the notation given above the first equilibrium principle can be expressed as follows. A route-flow vector $\mathbf{H}$ is in user equilibrium if it satisfies

$$(C_r(\mathbf{H}) > C_s(\mathbf{H})) \Rightarrow H_r = 0, \tag{3.3}$$

for all $(x, y) \in N \times N$ and all $R_r, R_s \in \mathcal{R}(x, y)$. This says, in effect, "for all O-D pairs (x, y) if the cost on route R_r is greater than the cost on route R_s then no flow will be on route R_r" which can be readily recognised as the Wardrop principle (1) above.

The system optimal equilibrium can be simply seen as minimising the total cost on the network given in equations (3.1) and (3.2).

Next, the idea of *demand* on a network needs to be introduced. Again, following [**135**], introduce the *origin-demand matrix* (O-D matrix) ρ where

$$\rho : N \times N \mapsto \mathbb{R}_+,$$

and $\rho(x, y)$ represents the number of drivers who wish to travel from an origin x to a destination y. Clearly not all $\mathbf{F}$ will satisfy this demand conditions. Therefore, it is necessary to introduce the set Δ of route-flow vectors which satisfy

$$\sum_{R_r \in \mathcal{R}(x,y)} F_r = \rho(x, y),$$

for all $x, y \in N$ where $F_r \geq 0$ for all r.

Formally $\rho(x, y)$ is what is known as a *fixed demand matrix* — that is, the number of drivers wishing to travel from x to y remains a constant whatever the cost of that travel. A cost-flow vector $\mathbf{F} \in \Delta$ is known as *demand feasible*. Similarly, define a set D of demand feasible link-flow vectors such that a link-flow vector $\mathbf{f} \in D$ is demand feasible. (Note that route-flow vectors uniquely determine link-flow vectors but link-flow vectors do not uniquely determine route-flow vectors.)

An early modelling approach to the problem is provided by [**51**] which combines a gravity model for demand with a Wardrop equilibrium formulation for route choice. An early review of models including models which vary the demand matrix is given by [**58**].

In [**135**] the author proves that, under certain quite general conditions, any network of the type described will have a unique, stable equilibrium. It should be noted that the formalism used here and the proofs given differ slightly from those given in [**135**] since the author restricts solutions to flow vectors within a *supply feasible set* S. This restriction is not made in this formulation since the same ends can be achieved to a very close approximation with a sufficiently steeply increasing cost function for areas outside S. The more general solution with the addition of a supply feasibility restriction is useful for problems related to control and pricing.

Consider a vector of demand feasible route flows $\mathbf{H} \in \Delta$ which satisfies equation (3.3). It follows from the definition of UE that no driver can lower his or her cost by swapping to another route *if those costs remain unchanged*. Therefore, it follows from equation (3.1)for total network cost that

$$\mathbf{C}(\mathbf{H}) \cdot \mathbf{F} \geq \mathbf{C}(\mathbf{H}) \cdot \mathbf{H} \text{ for all } \mathbf{F} \in \Delta. \tag{3.4}$$

This statement is equivalent to

$$[-\mathbf{C}(\mathbf{H})] \cdot (\mathbf{F} - \mathbf{H}) \leq 0 \text{ for all } \mathbf{F} \in \Delta, \tag{3.5}$$

or

$$-\mathbf{C}(\mathbf{H}) \text{ is normal, at } \mathbf{H}, \text{ to } \Delta. \tag{3.6}$$

Now, if equation (3.3) is not satisfied (the system is not in equilibrium) then there must be some $(x, y) \in N \times N$ and routes $R_r, R_s \in \mathcal{R}(x, y)$ such that

$$H_r > 0 \text{ and } C_r(\mathbf{H}) > C_s(\mathbf{H}),$$

which violates (3.3). Hence, the equivalent conditions given by equations (3.4), (3.5) and (3.6) are necessary and sufficient conditions for a Wardrop equilibrium. Reformulating these three conditions in terms of link flows, if $\mathbf{h}$ is the link flow vector corresponding to $\mathbf{H}$, then

$$\mathbf{C}(\mathbf{H}) \cdot \mathbf{F} = \mathbf{c}(\mathbf{h}) \cdot \mathbf{f},$$

which allows equations (3.4), (3.5) and (3.6) to be expressed as

$$\mathbf{c}(\mathbf{h}) \cdot \mathbf{f} \geq \mathbf{c}(\mathbf{h}) \cdot \mathbf{h} \text{ for all } \mathbf{f} \in D. \tag{3.7}$$

$$[-\mathbf{c}(\mathbf{h})] \cdot (\mathbf{f} - \mathbf{h}) \leq 0 \text{ for all } \mathbf{f} \in D. \tag{3.8}$$

$$-\mathbf{c}(\mathbf{h}) \text{ is normal, at } \mathbf{h}, \text{ to } D. \tag{3.9}$$

Given that D is a closed and convex set, for every point $\mathbf{g} \in \mathbb{R}^m$ there is a single point in D which is nearest to $\mathbf{g}$ (using the standard Euclidean distance). Define this point as $p(\mathbf{g})$. Define a map $T(\mathbf{f}) : D \mapsto D$ for every $\mathbf{f} \in D$ as

$$T(\mathbf{f}) = p(\mathbf{f} - \mathbf{c}(\mathbf{f})). \tag{3.10}$$

It can be shown that

$$\mathbf{h} \in D \text{ is a Wardrop equilibrium if and only if } T(\mathbf{h}) = \mathbf{h}. \tag{3.11}$$

Since $\mathbf{c}(\mathbf{f})$ is non-zero, the only way that $T(\mathbf{h}) = \mathbf{h}$ can occur is if $-\mathbf{c}(\mathbf{h})$ is normal at $\mathbf{h}$ to D which, by equation (3.9), is the condition for a Wardrop equilibrium.

THEOREM 3.1. If $\mathbf{c}(\mathbf{f})$ is a continuous function and D is a closed convex subset of $\mathbb{R}^m_+$ then there is a Wardrop equilibrium $\mathbf{h} \in D$.

PROOF. The map $T(\mathbf{f}) : D \mapsto D$ is a continuous map if $\mathbf{c}(\mathbf{f})$ is continuous. Therefore Brouwer's fixed point theorem [**21**] applies and the map has some fixed point $\mathbf{h}$. By equation (3.11) such a fixed point must be in Wardrop equilibrium. □

THEOREM 3.2. If $\mathbf{h} \in D$ is a Wardrop equilibrium then given the monotonicity condition

$$[\mathbf{c}(\mathbf{f}) - \mathbf{c}(\mathbf{g})] \cdot (\mathbf{g} - \mathbf{f}) < 0,$$

for any two distinct $\mathbf{f}, \mathbf{g} \in D$ then $\mathbf{h}$ is the only equilibrium in D.

PROOF. If $\mathbf{f}$ is any link flow vector in D distinct from $\mathbf{h} \in D$ (which is a Wardrop equilibrium) then

$$\mathbf{c}(\mathbf{f}) \cdot (\mathbf{h} - \mathbf{f}) = \mathbf{c}(\mathbf{h}) \cdot (\mathbf{h} - \mathbf{f}) + [\mathbf{c}(\mathbf{f}) - \mathbf{c}(\mathbf{h})] \cdot (\mathbf{h} - \mathbf{f}) < 0.$$

The first term must be zero or negative from equation (3.8) and the second term must be negative from the monotonicity condition in the hypothesis. But the equation with the terms reversed,

$$\mathbf{c}(\mathbf{h}) \cdot (\mathbf{f} - \mathbf{h}) < 0,$$

cannot be true since this would violate equation (3.8). Therefore $\mathbf{f} \in D$ cannot be an equilibrium position. □

It should be noted that this uniqueness is only in the sense of link flows. The route flows are not, in general, unique — a number of different route flow vectors $\mathbf{F} \in \Delta$ may be equivalent to a single $\mathbf{f} \in D$.

An ordered pair $(\mathbf{F}, \mathbf{H})$ of route flows in Δ is known as an *assignment process* if and only if

$$\mathbf{H} = \mathbf{F} \text{ or } \mathbf{C}(\mathbf{F}) \cdot \mathbf{H} < \mathbf{C}(\mathbf{F}) \cdot \mathbf{F}.$$

The pair $(\mathbf{F}, \mathbf{H})$ can be thought of as follows: if $\mathbf{F}$ represents the flow on the routes yesterday and $\mathbf{H}$ represents the flow on the routes today after drivers have made their route choice in response to the costs and flows experienced yesterday. This equation encodes the idea that drivers change their routes to reduce the expected costs that they would experience if the costs on routes remain the same on the next day.

A corresponding link-flow definition is that an ordered pair $(\mathbf{f}, \mathbf{h})$ is an assignment process if and only if $\mathbf{f}, \mathbf{h} \in D$ and

$$\mathbf{f} = \mathbf{h} \text{ or } \mathbf{c}(\mathbf{f}) \cdot \mathbf{h} < \mathbf{c}(\mathbf{f}).\mathbf{f}. \tag{3.12}$$

A Wardrop equilibrium is defined in [**135**] as *stable* if and only if $(\mathbf{f}, \mathbf{h})$ is an assignment process for any $\mathbf{f} \in D$.

THEOREM 3.3. Given this definition of stable, and given the monotonicity condition

$$[\mathbf{c}(\mathbf{f}) - \mathbf{c}(\mathbf{g})] \cdot (\mathbf{g} - \mathbf{f}) < 0,$$

for any two distinct link-flows $\mathbf{f}, \mathbf{g} \in D$ then a Wardrop equilibrium $\mathbf{h} \in D$ is stable.

PROOF. If $\mathbf{f} = \mathbf{h}$ then $(\mathbf{f}, \mathbf{h})$ is an assignment process. If $\mathbf{f} \neq \mathbf{h}$ then, from the proof of the previous theorem, $\mathbf{c}(\mathbf{f}) \cdot (\mathbf{h} - \mathbf{f}) < 0$ and hence, by equation (3.12) is an assignment process. □

The monotonicity condition in Theorems 3.2 and 3.3 is interesting. It was shown by [**36**] to be equivalent to requiring that the Jacobian matrix of link

costs with respect to link flows be positive definite. Without this condition it is easy to conceive networks where there are multiple equilibria or unstable equilibria. The condition can be thought of as very roughly stating, on the network as a whole, if flows increase on a link costs increase on that link and if flows decrease on a link costs decrease on that link. It is easy to conceive examples (particularly if we consider junction interactions) where this property does not hold. The monotonicity may also fail if responsive signal control or mixed travel modes are considered on the same network.

Stronger results can be achieved by assuming that link costs are *separable* (that is the cost on a link depends only on the flow on that link). The term *asymmetric* is often used to describe cost-flow relationships which are non-separable.

A number of extensions to this framework are possible. The most obvious extension is that the demand is not fixed. This case is addressed by a number of authors with the most obvious extension being the inclusion of an artificial link from the origin x to the destination y which represents the no travel decision. A model where the demand on the network is not fixed is known as an *elastic demand model*. This is a rich area of research but will not be covered in this chapter since the primary interest in this chapter is in route choice modelling. Other extensions, for example, using different user classes (for example, considering cars and goods vehicles as different demand matrices with different costs) are considered in [**1**] and [**137**].

Define Δ_{rs} where $r, s \in \{1, 2, \ldots, M\} : r \neq s$ as a vector of length M with -1 in the rth place, 1 in the sth place and 0 elsewhere. A route-flow vector $\mathbf{F} \in \Delta$ is *user-optimised* if, for all origins x and destinations y then

$$F_r > 0 \Rightarrow C_r(\mathbf{F}) \leq C_s(\mathbf{F} + \varepsilon\Delta_{rs}),$$

for all $0 < \varepsilon < F_r$ and for all $r, s \in \mathcal{R}(x, y) : r \neq s$.

This can be read as "flows are user-optimised if any driver who changes to an alternative route will experience a cost which is at least as great as the

old cost on his old route." [**78**]. Note the subtle difference between this and the claim that the user will experience greater costs if the route costs remain the same as on the previous day). It is clear that if this condition is met then the system is in a Wardrop equilibrium. In fact if $\varepsilon = 0$ this is just a rearrangement of equation (3.3). However, it has not been shown that a Wardrop equilibrium necessarily meets this condition. Indeed this condition could be seen as an alternate measure of the stability of a given UE assignment. For separable problems, the user-optimised condition is exactly equivalent to the Wardrop equilibrium condition [**136**]. However, for asymmetric problems, even monotone ones, counter-examples can be found.

It is proved in [**78**] that if the cost functions $\mathbf{C}(\mathbf{F})$ are differentiable and, for all origins x and destinations y,

$$\frac{\partial C_u}{\partial F_u} \geq \frac{\partial C_u}{\partial F_w} \text{ for all } u, w \in \mathcal{R}(x, y),$$

then a Wardrop equilibrium is user-optimised. This condition is true if no route passes through any intersection more than once and the dominant effect on a cost on a given link is due to flows on that link (as opposed to the dominant effect on the cost of the link being due to opposing links at an intersection for example).

The same paper makes the following definition.

Definition 3.1. A route-flow vector $\mathbf{F} \in \Delta$ is termed equilibrated if and only if for all origins x and destinations y,

$$(F_r > 0) \Rightarrow C_r(\mathbf{F} + \varepsilon\Delta_{rs}) \leq C_s(\mathbf{F} + \varepsilon\Delta_{rs}),$$

for all $0 < \varepsilon \leq F_r$ and for all $r \neq s \in \mathcal{R}(x, y)$.

This condition can be stated as: "any driver who changes to an alternative route will experience a cost which is at least as great as the *new* cost on his old route." It is a more rigorous condition than the previous one [**78**].

The extension of this type of problem to the dynamic case is problematic. It can be shown [**109**] that in the dynamic case the monotonicity condition of

Theorems 3.2 and 3.3 does not hold and thus the theorems do not translate naturally to the dynamic case.

3.7. Modelling Route Choice

This section describes the various theoretical models for route choice which have been developed stemming from the equilibrium theory given in the previous section. By necessity this review cannot be complete (and such a review ignores research on behavioural simulation models which are not based upon equilibrium assumptions). The section is split into discussions of Deterministic User Equilibrium (as described in the previous section), Stochastic User Equilibrium and Stochastic Loading Models.

3.7.1. Deterministic User Equilibrium Models. Deterministic user equilibrium (DUE) models are commonly used in practical assignment models. They are based on the assumption that drivers are rational, have complete and perfect information regarding network conditions and behave identically. (Some of these restrictions can be relaxed slightly, for example by splitting drivers into different "classes" of driver). Congestion is represented by means of capacity restraint and drivers choose a least cost route. The models seek a Wardrop type equilibrium [**155**].

Although DUE models are perhaps the most widely used models in practical assignment, it is recognised that they are characterised by limitations.

"Empirical studies of route choice demonstrate that the capacity restraint mechanism in such models is insufficient to explain the variety of routes chosen, especially in more lightly-loaded inter-urban networks" [**98**, page 174].

"Deterministic assignment is unrealistic since route choice decisions are based on perceived travel times or costs, which may vary across individuals. Further, some drivers do not know or judge incorrectly the shortest travel-time or least-cost path, or choose a path for reasons not captured by the time and cost functions" [**61**].

"...a deterministic (Wardrop) equilibrium is an unrealistic representation of the state of most urban networks. This is caused by variations in network conditions (e.g. the effect of weather and unexpected incidents on capacity) and variations in demand..." [**150**, page 42].

3.7.2. Stochastic User Equilibrium Models. Stochastic User Equilibrium (SUE) models were developed to account for the fact that not all users behave identically by assigning a variety of perceived link costs according to a distribution function. They form an equilibrium based around the idea of a random utility model which adds a random component to utilities (the benefit a user gets from traversing a link). They were originally described in [**22**] and [**42**]. At first stochastic models omitted capacity restraint considerations which limited their applicability to congested urban networks. This was rectified by models which combined UE with the SUE framework [**37**] [**54**] [**130**]. In such models, drivers' route choices are modelled as stochastic processes with capacity represented as link-based cost-flow relationships. The models can be thought of in terms of a *utility* (the costs/benefits of a certain route) and an *error term*.

The distribution function, also known as the error term (since it can be thought of as representing the "error" in the drivers' perceptions of the costs on the network), has proved extremely useful in justifying the applicability of such models. For example [**150**, page 41] states that the error term could represent three distinct effects: "influences on route choice which have been excluded from the generalised cost function; variations in route choice preferences between drivers, which are not explained by the route choice parameters used in assignment models; daily variation in network traffic conditions."

A general random utility model can be specified as,

$$U_{in} = V_{in} + \varepsilon_{in}, \tag{3.13}$$

were U_{in} is the utility that individual n associates with choice i, V_{in} is the deterministic part of that utility and ε_{in} is an error term for that choice and

that individual. The probability that individual n chooses alternative i is then given by

$$\mathbb{P}\left[i|C_n\right] = \mathbb{P}\left[U_{in} \geq U_{jn} \text{ for all } j \in C_n\right] = \mathbb{P}\left[U_{in} = \max_{j \in C_n} U_{jn}\right],$$

where C_n is the set of choices available to the individual. By varying the assumptions of this model a number of different models are available. In an SUE formulation, all paths are used (although excessively long paths will be used only by a vanishingly small amount of traffic).

Multinomial models, either multinomial logit(MNL) or probit(MNP), are used to implement SUE models. MNL is characterised by the following assumptions [**18**]:

(1) The utilities are independent and identically distributed (i.i.d.) with a Gumbel distribution.
(2) There is a response homogeneity across individuals.
(3) There is error variance-covariance homogeneity across individuals.

The Gumbel distribution is given by the distribution function,

$$F(\varepsilon) = \exp(-e^{-\mu(\varepsilon-\eta)}),$$

where $\mu > 0$ and η are the parameters of the distribution. The density function is

$$f(\varepsilon) = \mu e^{-\mu(\varepsilon-\eta)} \exp(-e^{-\mu(\varepsilon-\eta)}).$$

The mean of the distribution is $\eta + \gamma/\mu$, where,

$$\gamma = \lim_{k\to\infty} \sum_{i=1}^{k} \frac{1}{i} - \ln(k) \approx 0.5772,$$

also known as the Euler constant. The variance of the distribution is $\pi^2/6\mu^2$.

Solving this model, the probability that individual n chooses alternative i within the choice set C_n is given by

$$\mathbb{P}\left[i|C_n\right] = \frac{e^{\mu V_i n}}{\sum_{j \in C_n} e^{\mu V_j n}}.$$

The first assumption is particularly important when considering route and departure time choice. For route choice it implies that the costs of the routes are independent — this property is known as independence from irrelevant alternatives (i.i.a.) An illustration of the problems inherrent in the i.i.a. property is given by contemplating the distribution of users between, say, bus and private car. If the utility of each mode were identical for every user then half the population would use the bus and half would use the car. However, if we split the bus population into red and non-red buses then one third of the population would use the red buses, one third would use the non-red buses and one third would use the car. It is clear that red and non-red buses should not be considered as independent.

The i.i.a. property is widely held to be a difficulty for most route choice situations where two alternative routes may be largely (indeed almost entirely) identical. As an illustrative example consider two routes which are exactly the same except for at the end of the route where the driver has the choice of being in the left or right hand lane. It is clear that the costs of these routes are far from independent. Indeed they are almost wholly correlated. Similarly, for departure time choice, it is absolutely clear that the cost of travel when departing for work at 8:00am is, in no sense, independent from the cost of travel when departing at 8:05am.

MNP does not assume the i.i.d. property and therefore this criticism cannot be made of it. However, the formulation is much less tractable because so many more parameters must be estimated and therefore the application of MNP is restricted in real situations. The assumption of Probit is that the error terms in equation (3.13) are multivariate normal distributions with mean zero and a variance term which explicitly captures the interrelations of the choice set. The number of parameters to be estimated grows with the square of the size of the choice set. Since a given origin-destination pair may have an extremely large number of routes available, it can quickly be seen that the

tractability of the model will be limited for route choice problems in realistic networks.

3.7.3. Stochastic Loading Models. Following on from the MNL and MNP approaches previously described come stochastic loading models which attempt to address the gap between the i.i.d. problems of MNL and the tractability problems of MNP. A theoretical and empirical analysis of such models is provided by [**9**]. The models can be split into two main groups. The first group are derived from generalised extreme value (GEV) theory [**104**] and relax the assumption that the error components are independent. These models include nested logit [**10**] and [**104**], cross-nested logit [**153**], C-logit [**26**], paired combinatorial logit [**27**] and generalised nested logit [**161**].

The second group of models relaxes the assumptions of independence of error components and the assumption that they are identical. These models derived from the error components model [**25**]. Error components logit (ECL), also known as the logit kernel or mixed logit model, is a main model in this group and decomposes the error term into two components, one i.i.d. and one non-independent and non-identical. Recent work [**146**] further relaxes the MNL assumption of homogeneity across individuals. ECL is a relative newcomer to route choice modelling but provides interesting possibilities for the future of route choice modelling.

3.8. Modelling Departure Time Choice

A review of departure time choice up to 1996 is given by [**7**]. The report concludes: "Although much of the current research into dynamic assignment is also considering the simultaneous departure time choice problem, the additional computational complexity is likely to remain daunting for some time to come..." [**7**, page 86]. The author states: "It will be clear from this report that the topic is a major area of research, and that although much has been achieved, a satisfactory resolution of the outstanding problems remains some

way off." [**7**, page 87]. The report is downloadable online and is an excellent reference for the reader interested in more details of the topic than are available in the short review in this section.

The earliers work in departure time choice is [**152**] in which the author describes the use of tolls to spread departure time and hence reduce congestion. Three pioneering works in modelling departure time choice are [**75**], [**133**] and [**134**]. The first of these develops a UE based approach using queuing theory. The authors note that the approach is limited in that variability in travel times, work start times and users perception of costs may lead to their model over-predicting peak-spreading and under-predicting queue lengths. A stochastic approach and MNL formalism to analyse departure time choice is used in [**133**]. However, this approach is problematic as the author notes since the i.i.a. assumption is clearly violated. In [**134**] the author seeks to correct this by employing a generalisation of MNL.

More recent work has sought to incorporate time choice with route choice modelling in a dynamic framework. In [**99**] a framework is presented which describes the processes by which commuters' departure time decisions respond to the congestion they experience. An "indifference band" of tolerable delay which varies across individuals and shifts according to individual experienced is used to model how commuters adjust their departure time.

A dynamic model of peak-period congestion with a limited number of bottlenecks is developed in [**12**]. The model considers the effect of traffic conditions on mode, route and departure time choices. The temporal distribution of traffic volumes is predicted using an elastic demand model. The delays at bottlenecks are modelled using a deterministic queuing model, which determines waiting time as a function of queue length on arrival at the bottleneck. Day-to-day adjustments in the distribution of traffic are based on a Markovian model.

Congestion leads to dispersion of demand over a larger number of routes and simultaneously to shifts in departure time and peak-spreading [**148**]. The

paper suggests a model which implements route and departure time choices simultaneously, known as Dynamic User Optimium Departure time and Route choice (DUO-D&R). The model requires a dynamic O-D matrix with preferred arrival times.

Inspired by studies of peak-spreading [**120**] and demand responses for scheme appraisal [**65**] and [**127**] the problem of departure time choice has seen an upsurge of interest in the last five years. Recent work includes [**86**], [**118**] and [**151**].

3.9. Criticisms and Developments of Current Modelling Practice

Both DUE and SUE models make inherent assumptions about both rational behaviour and awareness of the network. Perhaps the best known paper criticising current modelling practices is [**62**]. This paper reviews errors and limitations of equilibrium modelling concluding: "It is the author's view that we were not in equilibrium when the data we use were collected, we are not in such a state now, there is no guarantee that the system is currently moving towards it, we will never arrive there, and even if we did we would not stay there for long." [**62**, page 124]. While this report is talking about all traveller choice dimensions (many of which are accepted to take place over a much longer time-scale than route and departure time choice), the criticisms are valid when applied to just the two choice dimensions in question here.

Another criticism of modelling practice comes from [**131**] which argues that rationality is bounded because of limits on the ability of drivers to assess the choices available. It is noted in [**110**] that most studies of transport networks assume equilibrium which, in turn, implies that drivers select routes rationally and from an unbiased perception of the state of the network. An alternative model without these assumptions is proposed where drivers choose routes in a heuristic manner with perceptions updated on the basis of user experience. The study finds that the system arising does not necessarily converge to a Wardrop type user equilibrium.

Driver information presents a specific challenge for equilibrium models. Advanced Traveller Information Systems (ATIS) such as roadside variable message signs and in-car systems for route guidance have impact only because they provide information to the driver. This information provision may significantly impact route and departure time choice [**11**]. Citing [**13**], [**100**] and [**119**], [**72**, page 110] argues that, "...the notion of a simple optimised decision-making rule is unrealistic for understanding fully the impact of ATIS on travel behavior." In [**88**] a Bayesian updating model is developed to analyse the mechanism by which drivers update their travel time perceptions from one day to the next, on the basis of ATIS and previous experience.

In [**101**] the effects of ATIS on route and departure time switching are analysed using experiments based upon a dynamic interactive travel simulator The data was applied to a behavioural model and the authors concluded that drivers' route choice decisions are based on the expectation of a travel time improvement exceeding a given threshold, which varies systematically with the remaining travel time to the destination, subject to a minimum absolute improvement.

As ATIS systems become more widespread, the assumption of rational behaviour in equilibrium models may become more reasonable. However, it is recognised that individuals may not comply with the provided information [**147**]. Route choice behaviour under real-time information is investigated by [**138**]. The model assumed behaviour is being based upon *compliance* (willingness to follow advice) and *inertia* (willingness to follow habitual behaviour). Simulator experiments support the simultaneous presence of both mechanisms in route choice behaviour.

While much attention has been given to the forecasting stage of SUE models, in practice, the parameters of such a model must be estimated from real-life measurements and this has received less attention. In practical scheme assessments, DUE models are more commonly used. Because of the previously discussed assumption of homogeneity of driver choice preferences and

constraints DUE models may under-predict the spread of drivers across routes in uncongested networks. The criticism about parameter estimation may also be made of departure time studies with little attention given to parameter estimation from on-street surveys or even driver surveys. Parameter values are usually extracted from historical studies such as [**76**] and [**133**] which are based upon surveys of user preferences.

Although substantial research effort has been devoted to the development of new methods for modelling route and departure time choices, it might be argued that more fundamental gaps in knowledge exist. One such gap is knowledge of the attribute-set relevant to route and departure time choices, and the appropriate representation of these attributes in choice utilities. In [**11**] it is suggested that travel time is perhaps the most important attribute influencing route choice, but recognised that difficulties exist in taking account of how individuals perceive travel time. Others variables to include might be path length, travel cost, traffic conditions, obstacles, road types and road condition. It seems clear that research in this area is an important priority.

3.10. Conclusions From Literature Survey

This chapter highlights several important weaknesses in current modelling practice and draws attention to certain research needs. Most modelling done in genuine scheme assessment makes the assumption of a fixed pool of drivers who travel every day or a larger pool who wish to travel every day but may not due to demand elasticity. It is far from clear if it is widely recognised that this is a crude approximation to the reality of a rush hour which appears to be composed of a variety of drivers, the majority of whom appear to travel only irregularly.

While SUE attempts to account for the fact that not all drivers have the same perception of a network, it still works within the framework of a fixed pool of rational drivers minimising their perceived costs within a network where the costs are generally assumed to be some linear combination of time and

distance. On-street evidence seems to show that the costs perceived by users are more complex than this.

Small shifts in departure time choice are widely acknowledged to be a major driver response to congestion. However, these have not been rigorously investigated by on-street studies and are only rarely modelled in practical scheme assessments even though they can absorb some of the worst impacts of increased congestion (or conversely cancel some of the benefits of reduced congestion).

It seems clear that route choice and departure time choice are, in some way, linked. However, there is little research investigating the nature of this linkage and practical evidence on the subject from on-street studies is scant. Further, while research is beginning on how ATIS influences choice, the modelling implications of this need to be examined, particularly with regard to assumptions about rationality and information availability in UE models.

In practical scheme assessment, little attention is given to the estimation of model parameters and there is a need to develop models which are theoretically-reasonable yet which have parameters which are efficient and robust to estimate in real-life studies.

CHAPTER 4

Set Theory for Matching Data

4.1. Introduction

This chapter describes a general framework for analysing problems in matching data across multiple data sets. The method developed is useful for situations where analysis is to be performed on several data sets containing information about unique individuals. The method answers questions of the type "How many unique individuals appear in three or more of the five data sets?" and is particularly useful for addressing situations where false matches are possible (that is, where two distinct individuals appear to be the same as a result of observational error).

The problem which gave rise to this work originally arose during roadside traffic surveys when attempting to track vehicles using their licence plates at multiple survey sites across a city. It should be emphasised, however, that the framework is sufficiently general that it could prove of use in any situation where it is important to track matches in data items across a small number of different data sets. In the real-life situation reported, the number of false matches could often be a significant fraction of the number of matches recorded

Using set theory, the problem has been placed in the context of lattices of the integer partition and a solution algorithm has been developed. The algorithm answers problems of the type "How many individuals are genuinely seen once each in every data set when the false matches have been excluded?" The algorithm has been implemented in the C++ programming language and tested on simulated data sets. The test results suggest that the method does indeed provide an unbiased estimator for the true number of matches in the data although the variance in the estimate can, unfortunately, be extremely

high in some cases. The method has been tested and found useful in removing false matches from real data but the high variance in the estimate can be a problem. The approach taken in this chapter is to begin by creating a framework for examining matches in multiple data sets in the most general manner and then to use this to specify the problem at hand and create an algorithm for its solution.

In Section 4.2 background to the problem within the context of transport engineering is described. In Section 4.3 an initial framework for discussing the problem is laid out. In Section 4.4 the concept of a *type of match* is defined using set theory and the concept of equivalence class. In Section 4.5 the set $\mathcal{M}_n$ of all types of match across n sites, is introduced. In Section 4.6 a partial ordering is defined for $\mathcal{M}_n$ and related to the problem of false matches. In Section 4.7 some functions for counting matches are introduced which, in Section 4.8, are used to create an algorithm for estimating the number of false matches in data. Finally, in Section 4.9 computational results are given for the performance of the matching algorithm on simulated data.

4.1.1. A Note About *Tuples*. Throughout this chapter the term *n-tuple* is used to describe an ordered set of n elements — somewhat akin to an *n-vector* but the n-tuples will not usually be elements within a vector space. The tuples are ordered sets of general elements. Sometimes tuples of sets are used. The notation of making an n-tuple bold will be used and its individual elements will be subscripted: $\mathbf{x} = (x_1, \ldots, x_n)$.

4.2. Background and Context of the Problem

The problem of tracking individual vehicles on a road network is a well-known and common problem in transport surveys. Several approaches are used for vehicle tracking. For example GPS location [**87**], [**121**] or cell-phones and vehicle tags [**43**]. One widely used method is the licence plate survey which may be either manual (using a roadside observer with a note pad, dictaphone or specialist recording equipment for the purpose) or automatic (using roadside

cameras [**162**]). In both manual and automatic surveys the problem of errors in the recordings must be considered. Some of the difficulties with such surveys are described in [**132**] and [**129**]. Manual surveys are commonly partial plate surveys (for reasons of time and convenience) and, in addition to the recording errors, the problem of accidental *false matches* between different vehicles which have the same partial plate is an important one.

A number of researchers have approached the false matching problem for licence plates. An early approach for removing false matches between observations at two sites is given by [**73**] which uses a simple correction based upon the probability of two plates being the same. Several methods for approaching the problem, including a method for making two point matches between pairs selected from a number of survey sites, is described in [**97**]. A graphical method which provides a good visualisation of the problem is described in [**158**] and this is used in the next section. A further refinement which uses journey time to assess the likelihood of a match is described in [**159**]. A maximum likelihood estimator for the true matches based upon assumptions about the statistical nature of the inbound traffic is provided in [**156**] and [**94**] extends this method to three sites. Many of these methods are used for matching between two sites in the next chapter. However, none of the authors tackles the general problem of removing false matches from matches across n sites.

More generally, a considerable amount of work has been done on "matching problems" in combinatorics — the usual approach being graph theoretic with an edge between two nodes indicating a match. However, in the case of matches across n data sets, the graph theoretic approach is inappropriate since the matches are not just pairwise.

The framework developed in the next section considers problems of the type *"How many individuals occur in three of the five data sets?"* or *"How many individuals are genuinely seen in all five data sets being investigated once all false matches are removed?"*. The framework places the problem in

the context of basic set theory [**66**] and shows how the problem maps onto the well-known mathematical topic, partitions of the natural numbers.

The motivating problem for this chapter arose when partial licence plate data was collected across a number of survey sites (the survey itself will be described in Chapter 5). In the survey undertaken, the researchers wished to know how many vehicles were seen on all of six survey days. Because only partial plate surveys were conducted, false matches occurred. In extreme cases, the number of matches attributed to false matching in data were estimated to exceed the number of genuine matches (and this was certainly the case in the simulated data). The problem is a surprisingly tricky one since false matches can occur in a huge number of ways. For example, the same partial plate observed on all five weekdays could represent: a single vehicle identified on all five days; five vehicles which by coincidence have the same partial plate, one observed on each day; one vehicle observed on Monday and a second vehicle observed on Tuesday through until Friday; one vehicle observed on Monday and Tuesday, a second vehicle observed on Wednesday and Friday and a third vehicle observed only on Thursday, and all three having the same partial place; or any of a multiplicity of other ways false matches could occur. Indeed, it is evident that merely enumerating the ways in which a false match can occur is a non-trivial problem.

4.2.1. Notes on Licence Plate Observation. Throughout this chapter, examples will be given using licence plates with a specific format. An example plate would be: `A134SDR`. This type of plate was used in the UK from 1983 up until mid 2001 [**5**]. The specific details of the type of plate used are completely irrelevant to the methods developed within this chapter. However, choosing parts of a plate to survey for partial plate surveys and estimating the probability of two unique plates matching is not straightforward due to correlations related to year and location identifiers on licence plates. The properties of UK licence plates are not of general interest and are not covered here.

While the exact details of licence plates themselves are not relevant, in general, when collecting partial plates, it is important to record sufficient information. Consider, for example, using the old format plates and collecting only digits. Add the simplifying assumption that the digits are flatly distributed over the thousand possible combinations from 0 to 999. Therefore, the chances of two distinct vehicles being a false match is one in one thousand. Now, a reasonable size of data set for a traffic survey is one thousand vehicles collected at a site — this would represent quite a significant flow but (as will be seen later) real roads have larger flows than this even in town centres. If two sites each have one thousand vehicles surveyed and the two sites have no genuine matches between them, the expected number of false matches is one thousand (from the fact that there are a million pairs of vehicles between the two sites, each with a one in a thousand chance of a match). When the number of false matches is obviously greater than the effect being measured (one thousand is the largest possible number of true matches) then the problems encountered are likely to be extreme. The problems worsen as the probability of a false match increases or as the number of vehicles seen at each site increases.

The specific details of the licence plates collected are not relevant to the method described here and it is not a problem for the work described if the fleet under study is composed of vehicles with different styles of licence plates. Indeed the method is extremely general and, it is hoped, can be used on studies which are of other observation types and are completely outside the sphere of road traffic engineering.

4.3. Setting for the Problem

Assume that there are n data sets (survey sites) and at each site i there exist a set of observations $\mathbf{S}_i$. Each observation is a sighting of one from a set of identifiable, unique individuals $\Omega = \{\omega_1, \ldots, \omega_N\}$ where N is the number of individuals. The n-tuple of all n sites is denoted by $\mathbf{S}$ where $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n)$.

It is assumed, initially, that enough information will be recorded in an observation to distinguish between any two members of Ω — this assumption will be relaxed later.

Note that there is no restriction on the time and location of these observations. In the context of traffic engineering they could represent the same site observed on different days or different sites observed on the same day or any mix of this (for example, three sites observed on one day and the same three sites observed again on a second day). There is no particular assumption about the ordering of the data sets and it is perfectly reasonable for an individual to be observed in data set one and three but not in data set two. In fact, the ordering of the data sets is totally arbitrary.

DEFINITION 4.1. The *observation function* is a function acting on the members of Ω such that

$$(i = j) \Leftrightarrow (f(\omega_i) = f(\omega_j)) .$$

In other words, the observation function is a function which uniquely identifies the objects observed. If the objects are different then the result of the observation function is different. The domain of $f(\omega)$ is Ω and its range is a property of each ω_i sufficient to uniquely distinguish it from other members of Ω.

In the case of the licence plate surveys discussed here, the domain of $f(\omega)$ is the fleet of vehicles operating in the UK and its range is the set of licence plates used by the vehicles in this fleet. In other words, the observation represented by the function is enough to uniquely determine the object observed and distinguish it from all other such objects.

The members of the sets $\mathbf{S}_i$ will be observations $f(\omega)$ with $\omega \in \Omega$. Therefore, for each site i,

$$\mathbf{S}_i = \{f(\omega_{(i,1)}), f(\omega_{(i,2)}), \ldots, f(\omega_{(i,N)})\}, \tag{4.1}$$

where N is the number of observations at site i and $\omega_{(i,j)} \in \Omega$ for all i, j. In the context of licence plate surveys, $\omega_{(i,j)}$ is the jth vehicle observed at site i and $f(\omega_{(i,j)})$ is the licence plate of this vehicle.

A technicality which should be noted in passing is the possibility that some ω_j is observed more than once in a set of observations $\mathbf{S}_i$ (in other words, an individual is observed twice at the same site). This would cause a problem since, formally, a set cannot contain members which are identical (or rather $\{x, x\} = \{x\}$). This would be the case if $\omega_{i,k} = \omega_{i,l}$ for any $k \neq l$ in equation (4.1). This problem will be made worse when the requirement that observations uniquely determine individuals is dropped. To prevent this problem, the observations could be, for example, tagged with a time of day or a suffix to denote the order in which the observation was made. This requirement is a pure technicality and will not affect anything which follows nor will it be mentioned again.

Definition 4.2. An *n-tuple of observations* can be formed by taking one observation from each of the n sites in order.

$$\mathbf{x} = (x_1, \ldots, x_n),$$

where $x_i \in \mathbf{S}_i$.

To make this more concrete, consider the following three sets of observations,

$$\mathbf{S}_1 = \{\texttt{A123XYZ}, \texttt{B256ABC}\}$$

$$\mathbf{S}_2 = \{\texttt{A123XYZ}, \texttt{C232SAD}, \texttt{B256ABC}\}$$

$$\mathbf{S}_3 = \{\texttt{C789ABC}, \texttt{A123XYZ}, \texttt{A543OPQ}\}.$$

Three possible n-tuples of observations are,

$$\mathbf{x} = (\texttt{A123XYZ}, \texttt{A123XYZ}, \texttt{C789ABC}) \tag{4.2}$$

$$\mathbf{y} = (\texttt{A123XYZ}, \texttt{A123XYZ}, \texttt{A123XYZ}) \tag{4.3}$$

$$\mathbf{z} = (\texttt{B256ABC}, \texttt{B256ABC}, \texttt{A123XYZ}). \tag{4.4}$$

DEFINITION 4.3. The set $\mathcal{S}$ is the set of all possible such n-tuples across the observations in the set of sites $\mathbf{S}$. This is given by the Cartesian product,

$$\mathcal{S} = \mathbf{S}_1 \times \mathbf{S}_2 \times \ldots \times \mathbf{S}_n = \prod_{i=1}^{n} \mathbf{S}_i.$$

It follows immediately that the number of possible n-tuples $\#\mathcal{S}$ is given by $\prod_{i=1}^{n} \#\mathbf{S}_i$.

4.4. Types of Match

Consider the tuples, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ as given by equations (4.2), (4.3) and (4.4). It is clear that in some sense that $\mathbf{x}$ and $\mathbf{z}$ are in some way the same type of tuple in that they both represent observations of the same vehicle at sites one and two and a different vehicle at site three. It is equally clear that $\mathbf{x}$ and $\mathbf{y}$ are in this sense a different type of tuple. This concept of type of match is formalised by an equivalence relation.

DEFINITION 4.4. Two n-tuples of observations $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ are the same *type of match* if and only if $\mathbf{x} \sim \mathbf{y}$ where $\sim$ is the equivalence relation

$$(\mathbf{x} \sim \mathbf{y}) \text{ if and only if } (x_i = x_j) \Leftrightarrow (y_i = y_j) \text{ for all } i, j \in 1, 2, \ldots, n.^{1}$$

[1]For simplicity the limits $i, j \in 1, 2, \ldots, n$ on indices will usually be omitted where, as in this case, they are obvious.

In other words, two n-tuples of observations are the same type of match if they match in the same places as each other and differ in the same places. For example,

$$(1, 2, 2, 4) \sim (5, 1, 1, 4),$$

and

$$(\text{pear}, \text{pear}, \text{apple}) \sim (\alpha, \alpha, \eta),$$

but

$$(\bigcirc, \bigcirc, \triangle, \triangle) \not\sim (1, 2, 1, 2).$$

It must now be shown that Definition 4.4 is, in fact, an equivalence relation (reflexive, symmetric and transitive).

Reflexive: $[\mathbf{x} \sim \mathbf{x}]$ follows immediately since clearly $(x_i = x_j) \Leftrightarrow (x_i = x_j)$.

Symmetric: $[(\mathbf{x} \sim \mathbf{y}) \Rightarrow (\mathbf{y} \sim \mathbf{x})]$ follows by assuming the converse. If $\mathbf{x} \sim \mathbf{y}$ and $\mathbf{y} \not\sim \mathbf{x}$ then there exists some i and j where $y_i = y_j$ but $x_i \neq x_j$, a contradiction if $\mathbf{x} \sim \mathbf{y}$.

Transitive: [$\mathbf{x} \sim \mathbf{y}$ and $\mathbf{y} \sim \mathbf{z}$ together imply $\mathbf{x} \sim \mathbf{z}$] follows because if $\mathbf{x} \sim \mathbf{y}$ and $\mathbf{y} \sim \mathbf{z}$ for all i and j then $x_i = x_j$ implies $y_i = y_j$ which in turn implies $z_i = z_j$. The same chain of reasoning means that $z_i = z_j$ implies $x_i = x_j$ and therefore the relationship is transitive.

4.5. The Set of All Types of Match, $\mathcal{M}_n$

An obvious next question to ask is "For n sites, how many *types of match* exist?" To answer this question, consider the equivalence relation given by Definition 4.4 as a partition of the set of all possible n-tuples. A *transversal* is a set containing one and only one representative for each partition. This *transversal* will be referred to as $\mathcal{M}_n$ and by definition has the properties that no distinct members of $\mathcal{M}_n$ are equivalent under Definition 4.4 but any n-tuple is equivalent to some member of $\mathcal{M}_n$. The notation $\mathbf{x}_n^{\mathcal{M}}$ will be used to designate n-tuples which are members of $\mathcal{M}_n$.

DEFINITION 4.5. An n-tuple $\mathbf{x}_n^{\mathcal{M}} = (x_1, \ldots, x_n) \in \mathcal{M}_n$ if and only if $x_i \in \mathbb{N}$ and

$$x_i = \begin{cases} 1 & i = 1 \\ x_j \text{ for some } j < i & i > 1 \quad \text{or} \\ 1 + \max_{j<i}(x_j) & i > 1. \end{cases}$$

This can be thought of as labelling the first element of the n-tuple 1 and every subsequent element with either the same label as an appropriate earlier element (if it matches some earlier element) or the next available integer (if it matches no earlier element).

THEOREM 4.1. The set $\mathcal{M}_n$ of all possible $\mathbf{x}_n^{\mathcal{M}}$ meeting the conditions of Definition 4.5 is a transversal of the set of all possible n-tuples partitioned by the equivalence relation in Definition 4.4.

PROOF. It is necessary to establish two things:

(1) For any n-tuple $\mathbf{x}$ there exists some $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{x} \sim \mathbf{y}_n^{\mathcal{M}}$.
(2) No two distinct elements of $\mathcal{M}_n$ are equivalent.

To prove the first part define a procedure to calculate $\mathbf{y}_n^{\mathcal{M}}$ from $\mathbf{x} = (x_1, \ldots, \mathbf{x}_n)$ such that $\mathbf{y}_n^{\mathcal{M}} \sim \mathbf{x}$. Such a procedure is defined in Table 4.1.

(1) Set $y_1 = 1$.
(2) Set $r = 2$.
(3) If $x_r = x_i$ where $(i < r)$ then $y_r = y_i$
(4) Otherwise $y_r = \max_{i<r}(y_i) + 1$
(5) If $r < n$ then increment r and go back to step 3.

TABLE 4.1. Procedure for forming $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{x} \sim \mathbf{y}_n^{\mathcal{M}}$.

This procedure will create some n-tuple $\mathbf{y}_n^{\mathcal{M}}$ given an n-tuple $\mathbf{x}$. It remains to prove that $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ and $\mathbf{y}_n^{\mathcal{M}} \sim \mathbf{x}$. Since $y_1 = 1$ and either $y_i = y_j$

for some $(j < i)$ or $y_i = \max_{j<i}(y_j) + 1$ then, clearly $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$. It is also clear that if the above procedure is followed $\mathbf{x} \sim \mathbf{y}_n^{\mathcal{M}}$. From step three in the procedure it must always be true that $(x_i = x_j) \Rightarrow (y_i = y_j)$ and from step four then $(x_i \neq x_j) \Rightarrow (y_i \neq y_j)$. Therefore $(x_i = x_j) \Leftrightarrow (y_i = y_j)$ and so, from Definition 4.4, $\mathbf{x} \sim \mathbf{y}_n^{\mathcal{M}}$.

For the second part of the proof, it must be shown that no two distinct elements of $\mathcal{M}_n$ are equivalent. Or alternatively, that if two elements of $\mathcal{M}_n$ are equivalent then they must also be equal. That is, for all $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ then $(\mathbf{x}_n^{\mathcal{M}} \sim \mathbf{y}_n^{\mathcal{M}}) \Rightarrow (\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}})$.

If $\mathbf{x}_n^{\mathcal{M}} \neq \mathbf{y}_n^{\mathcal{M}}$ then there must be some earliest element r of the n-tuples at which they differ. Therefore, define r as the earliest element of $\mathbf{x}_n^{\mathcal{M}}$ such that $x_r \neq y_r$. Assume without loss of generality that $x_r < y_r$. By Definition 4.5, either $y_r = y_i$ for some $i < r$ or $y_r = \max_{i<r}(y_i) + 1$.

In the first case, $y_r = y_i$, however, $x_r \neq y_r$ (by the definition of r) and therefore, since $y_i = x_i$ and $x_r \neq x_i$ by Definition 4.4, $\mathbf{x}_n^{\mathcal{M}} \not\sim \mathbf{y}_n^{\mathcal{M}}$.

In the second case, $y_r = \max(y_i) + 1$. Since $x_r \neq y_r$, it is clear that there is some element x_i with $(i < r)$ such that $x_r = x_i$ but $y_r \neq y_i$ and therefore $\mathbf{x}_n^{\mathcal{M}} \not\sim \mathbf{y}_n^{\mathcal{M}}$.

Therefore it has been proved that, if element r exists, the two classes are not equivalent. If there is no such element r then obviously $x_i = y_i$ for all i and $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$. □

The procedure defined by Table 4.1 can be thought of as a map from the set of all possible n-tuples to the set $\mathcal{M}_n$. An example of this map in use is,

$$(\bigcirc, \square, \bigcirc, \triangle, \triangle) \mapsto (1, 2, 1, 3, 3).$$

Thus it has been shown that $\mathcal{M}_n$ in Definition 4.5 is a transversal of the equivalence classes in Definition 4.4 for all n-tuples. Table 4.1 defines a procedure which will convert any n-tuple of observations $\mathbf{x}$ into $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n : \mathbf{x} \sim \mathbf{y}_n^{\mathcal{M}}$.

DEFINITION 4.6. The *matching class* of an n-tuple $\mathbf{x}$ is the member of $\mathcal{M}_n$ to which it is equivalent. That is, the matching class of an n-tuple $\mathbf{x}$ is $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n : \mathbf{x} \sim \mathbf{y}_n^{\mathcal{M}}$.

Table 4.1 gives a procedure to find the *matching class* of any n-tuple.

DEFINITION 4.7. The height $H(\mathbf{x}_n^{\mathcal{M}})$ of an n-tuple $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ is the value of its maximal element.

$$H(\mathbf{x}_n^{\mathcal{M}}) = \max(x_i).$$

DEFINITION 4.8. A *true match* $\mathcal{M}_n(\mathcal{T})$ is the member of $\mathcal{M}_n$ with height 1. That is, $\mathcal{M}_n(\mathcal{T}) = (1, 1, \ldots, 1)$. This represents an observation of the same individual at every one of n sites. A *false match* $\mathcal{M}_n(\mathcal{F})$ is the member of $\mathcal{M}_n$ with height n. That is, $\mathcal{M}_n(\mathcal{F}) = (1, 2, \ldots, n)$. This represents an observation of n different individuals, one each at every one of n sites.

Note, that most matching classes are neither a *true match* nor a *false match* by this definition but instead are somewhere in between. The matching classes $\mathcal{M}_1$ and $\mathcal{M}_2$ are special cases. In $\mathcal{M}_1 = \{(1)\}$ then the true and false matches are identical. That is $\mathcal{M}_1(\mathcal{T}) = \mathcal{M}_2(\mathcal{F}) = (1)$. It should be noted that $\mathcal{M}_1$ is something of a special case since it is hard to define quite what a match means when only one object is being matched. In $\mathcal{M}_2 = \{(1, 1), (1, 2)\}$ then $\mathcal{M}_2(\mathcal{T}) = (1, 1)$ and $\mathcal{M}_2(\mathcal{F}) = (1, 2)$ and there are no other elements.

4.5.1. Mapping $\mathcal{M}_n$ to the Set of Partitions of the First n Integers. A partition of the first n integers is a set $\mathcal{P}$ of non-empty sets Y_i (that is $\mathcal{P} = \{Y_1, \ldots, Y_m\}$) where each of the first n integers is a member of one and only one of the sets Y_i. Call the set of all possible such partitions of the first n integers $\mathcal{P}_n$.

THEOREM 4.2. The set $\mathcal{M}_n$ has the same number of elements as the set P_n, the set of all possible partitions of the first n integers.

To complete this proof some subsidiary propositions must first be proved.

DEFINITION 4.9. An *injection* from a set A to a set B is a function such that:

(1) $f(a) \in B$ for all $a \in A$.
(2) $(f(a) = f(a')) \Rightarrow (a = a')$ for all $a, a' \in A$.

DEFINITION 4.10. A *bijection* is an injection where, for every $b \in B$, there exists $a \in A$ such that $f(a) = b$. A bijection is also sometimes referred to as a *one-to-one correspondance*.

If a function can be found which is a bijection from A to B then the two sets must have the same number of members ($\#A = \#B$).

Take an n-tuple $\mathbf{x}_n^{\mathcal{M}} = (x_1, \ldots, x_n) \in \mathcal{M}_n$.

(1) Set $j = 1$.
(2) Define $Y_j = \{y_1, \ldots, y_m\}$ where the y_i are all the integers from 1 to n such that $x_{y_i} = j$. (That is, Y_j is the set of indices of elements in $\mathbf{x}_n^{\mathcal{M}}$ for which the elements with those indices have the value j.)
(3) Increment j and, if $j \leq H(\mathbf{x}_n^{\mathcal{M}})$, then go to step 2.

The set $\mathcal{P}$, given by

$$\mathcal{P} = \{Y_1, Y_2, \ldots, Y_{H(\mathbf{x}_n^{\mathcal{M}})}\},$$

is a partition of the first n integers.

TABLE 4.2. Procedure for mapping from $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ to $\mathcal{P} \in \mathcal{P}_n$.

PROPOSITION 4.1. Table 4.2 defines a injective map $\mathcal{M}_n \mapsto \mathcal{P}_n$.

PROOF. It must be shown that if $\mathbf{x}_n^{\mathcal{M}} = (x_1, \ldots, x_n) \in \mathcal{M}_n$ and $\mathbf{x}_n^{\mathcal{M}} \mapsto \mathcal{P}$ using the above procedure then $\mathcal{P} \in \mathcal{P}_n$. This follows immediately from the fact that during step 2, every integer from 1 to n must be placed in one and

only one Y_j above since every x_i must take a value in the range 1 to $H(\mathbf{x}_n^{\mathcal{M}})$ and j takes values from 1 to $H(\mathbf{x}_n^{\mathcal{M}})$. Thus Table 4.2 is a map from $\mathcal{M}_n \mapsto \mathcal{P}_n$.

Secondly it must be shown that if $\mathbf{x}_n^{\mathcal{M}} \mapsto \mathcal{P}$ and $\mathbf{y}_n^{\mathcal{M}} \mapsto \mathcal{P}$ (where $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$) then $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$. Assume to the contrary, that $\mathbf{x}_n^{\mathcal{M}} \neq \mathbf{y}_n^{\mathcal{M}}$ and define r as the smallest integer such that $x_r \neq y_r$. Clearly $r > 1$ since by Definition 4.5, then $x_1 = y_1 = 1$. Now, assume without loss of generality that $x_r > y_r$. Therefore, there must be some previous element y_i of $\mathbf{y}_n^{\mathcal{M}}$ such that $y_i = y_r$ where $(i < r)$. If this were not the case then $y_r = \max_{j<r}(y_j)+1$ and x_r can be no bigger than this since the largest value x_r can have is $x_r = \max_{j<r}(x_j) + 1$ which is equal to $\max_{j<r}(y_j) + 1$ since, by the definition of r, $x_j = y_j$ for all $j < r$. Since $y_i = y_r$ and $y_r \neq x_r$ then $x_i \neq x_r$. However, by step 2, if $y_i = y_r$ then they must be in the same set $Y \in \mathcal{P}$ but if $x_i \neq x_r$ then they cannot be in the same set $Y \in P$. This is a contradiction and thus the assumption must be false and $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$. Therefore, the map is an injection. □

An example of the map from Table 4.2 in use is

$$(x, x, y, x) \mapsto \{\{1, 2, 4\}, \{3\}\}.$$

Begin with a set $\mathcal{P} \in \mathcal{P}_n$.

(1) Create the set $\mathcal{P}' = \{Y_1, \ldots, Y_m\}$ which is a copy of $\mathcal{P}$. Create an n-tuple $\mathbf{x}$.
(2) Define k = 1.
(3) Let X be the set Y_i in $\mathcal{P}'$ which contains the smallest integer. Remove the set Y_i from $\mathcal{P}'$.
(4) For all $x_i : i \in X$ set $x_i = k$.
(5) Add one to k and if $\mathcal{P}' \neq \emptyset$ then go to step 3.

When this procedure is finished, $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{M}_n$.

TABLE 4.3. Procedure for mapping from $\mathcal{P} \in \mathcal{P}_n$ to $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$.

PROPOSITION 4.2. Table 4.3 defines a injective map $\mathcal{P}_n \mapsto \mathcal{M}_n$.

PROOF. Begin with the assumption that some set $\mathcal{P} = \{Y_1, \ldots, Y_m\} \in \mathcal{P}_n$ (where m is the number of sets into which the integers have been partitioned) is mapped via this function to some n-tuple $\mathbf{x}$.

First it is necessary to show that for all $\mathcal{P} \in \mathcal{P}_n$ then the resultant $\mathbf{x}$ is a member of $\mathcal{M}_n$. Since all integers from 1 to n are part of some $Y_l \in \mathcal{P}$ then each element x_i must be set at step 4 on some iteration of the procedure. By the same token, no element x_i will be set twice in step 4 since each number can only be in one of the sets Y_l. Consider the definition of the members of $\mathcal{M}_n$ given in Definition 4.5. It is clear that $x_1 = 1$ since, at step 4, when $k = 1$ then 1 is the lowest number from 1 to n in any of the Y_l and so $x_1 = 1$. It must be proved that for $i > 1$ either $x_i = x_j$ for some $j < i$ or $x_i = \max_{j<i}(x_j) + 1$. Assume the contrary for some x_i and further assume that i is the smallest such i for which this is the case.

It is clear that $x_i \geq 1$ since k begins at 1 and counts upwards and it has already been shown that all the x_i were set equal to some $k > 0$ at stage 4. Therefore $x_i > \max_{j<i}(x_j) + 1$. This implies that there exists no $x_j < x_i$ such that $j < i$ and $x_j = x_i - 1$. When stage 4 of the procedure was reached with $k = x_i - 1$ then there must have been some non-empty set Y_l remaining in $\mathcal{P}'$. Furthermore, Y_l was chosen to be the set which has the smallest integer in it. It follows that in step 4 some x_j (where $j < i$) must have been set to $x_i - 1$ which is a contradiction. Therefore it has been shown that $\mathcal{P} \mapsto \mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$.

It remains to be shown that for $\mathcal{P}^A \in \mathcal{P}_n$ and $\mathcal{P}^B \in \mathcal{P}_n$, if $\mathcal{P}^A \mapsto \mathbf{x}_n^{\mathcal{M}}$ and $\mathcal{P}^B \mapsto \mathbf{x}_n^{\mathcal{M}}$ then $\mathcal{P}^A = \mathcal{P}^B$. This follows almost immediately from noting that if two numbers $i, j \in (1, \ldots, n)$ are members of the same set Y_i in $\mathcal{P}^A$ then they are the indices of equal elements $x_i = x_j$ in $\mathbf{x}_n^{\mathcal{M}}$ and must therefore be in the same set Y_k in $\mathcal{P}^B$. If two numbers $i, j \in (1, \ldots, n)$ are part of different sets Y_k and Y_l in $\mathcal{P}^A$ then they are the indices of unequal elements in $\mathbf{x}_n^{\mathcal{M}}$ (that is $x_i \neq x_j$) and therefore must be part of different sets in $\mathcal{P}^B$. It therefore follows that $\mathcal{P}^A = \mathcal{P}^B$ since any two numbers are in the same set in $\mathcal{P}^A$ and

also in the same set in $\mathcal{P}^B$ and any two numbers in different sets in $\mathcal{P}^A$ are also in different sets in $\mathcal{P}^B$. □

Since there is an injective map $\mathcal{P}_n \mapsto \mathcal{M}_n$ then $\#\mathcal{P}_n \leq \#\mathcal{M}_n$. Similarly, since there is an injective map $\mathcal{M}_n \mapsto \mathcal{P}_n$ then $\#\mathcal{M}_n \leq \#\mathcal{P}_n$. Therefore there are as many members in $\mathcal{M}_n$ as there are in $\mathcal{P}_n$ ($\#Mn = \#Pn$) and the maps defined in Tables 4.2 and 4.3 are both bijections and Theorem 4.2 is proved.

4.5.2. Enumerating $\mathcal{M}_n$. It is well-known (see [**149**, pages 119–128]) that the number of members of $\mathcal{P}_n$ can be counted using Bell Numbers and Stirling numbers of the second kind.

DEFINITION 4.11. Stirling numbers of the second kind $S(n,k)$ are defined by the recursive relationship

$$S(n,k) = \begin{cases} kS(n-1,k) + S(n-1,k-1) & n > 0 \text{ and } 0 < k \leq n \\ 1 & n = k = 0 \\ 0 & \text{otherwise.} \end{cases}$$

DEFINITION 4.12. The Bell number $B(n)$ is given by

$$B(n) = \sum_{k=1}^{n} S(n,k) \text{ for all } n > 0.$$

THEOREM 4.3. Given the definitions of $S(n,k)$ and $B(n)$ above:

(1) The total number of members of $\mathcal{P}_n$ which are partitions into k sets is given by $S(n,k)$.
(2) The total number of members of $\mathcal{M}_n$ with height k is also given by $S(n,k)$.
(3) The total number of members of $\mathcal{P}_n$ (and therefore $\mathcal{M}_n$) is given by the Bell number $B(n)$.

PROOF. The first part is proved in [**149**, page 125]. The second part follows immediately from the fact that a partition of the integers into k sets is mapped bijectively to a member of $\mathcal{M}_n$ with height k using the map defined in Table 4.2. The third part follows from the fact that the Bell numbers are the sum over all possible Stirling numbers for a given n and the already established fact that $\#\mathcal{M}_n = \#\mathcal{P}_n$. □

4.5.3. Constructing $\mathcal{M}_n$ Computationally. Clearly $\mathcal{M}_1 = \{(1)\}$. To construct $\mathcal{M}_{n+1}$ from $\mathcal{M}_n$ use the procedure in Table 4.4.

(1) Define $M = \emptyset$ and $\mathbf{X}$ to be the set $\mathcal{M}_n$.
(2) Set $\mathbf{x}_n^{\mathcal{M}}$ to be some element from $\mathbf{X}$ and remove that element from $\mathbf{X}$.
(3) From the n-tuple $\mathbf{x}_n^{\mathcal{M}}$ construct an $(n+1)$-tuple by adding the integers from 1 to $H(\mathbf{x}_n^{\mathcal{M}})+1$ as the $n+1$th element of the $(n+1)$-tuple. Add these tuples to the set M.
(4) If any elements remain in $\mathbf{X}$ then go to step 2.
$\mathcal{M}_{n+1}$ is the set M after this procedure completes.

TABLE 4.4. Constructing $\mathcal{M}_{n+1}$ from $\mathcal{M}_n$.

This is process is illustrated in Figure 4.1.

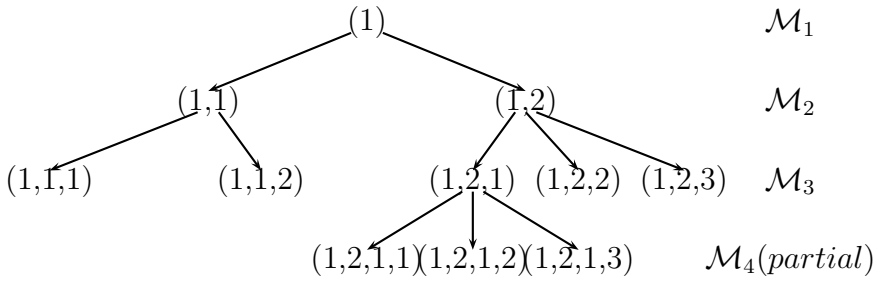

FIGURE 4.1. Construction of $\mathcal{M}_{n+1}$ from $\mathcal{M}_n$.

4.6. A Partial Ordering on the Set $\mathcal{M}_n$

A useful *partial ordering* can be defined on the set $\mathcal{M}_n$.

DEFINITION 4.13. For two n-tuples $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ a partial ordering relation $\succsim$ can be defined by

$$\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}} \text{ if and only if } (x_i = x_j) \Rightarrow (y_i = y_j).$$

To be a partial ordering, the relation must be reflexive, anti-symmetric and transitive. Again, these properties are easily proved.

Refexive [$\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{x}_n^{\mathcal{M}}$ for all $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$]: This is trivially true since $(x_i = x_j) \Rightarrow (x_i = x_j)$.

Anti-Symmetric [$\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}}$ and $\mathbf{y}_n^{\mathcal{M}} \succsim \mathbf{x}_n^{\mathcal{M}}$ together imply $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$ for all $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$]: This trivially follows since if both conditions together apply then $(x_i = x_j) \Leftrightarrow (y_i = y_j)$ and hence $\mathbf{x}_n^{\mathcal{M}} \sim \mathbf{y}_n^{\mathcal{M}}$ from Definition 4.4. It has already been shown that this implies $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$.

Transitive [$\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}}$ and $\mathbf{y}_n^{\mathcal{M}} \succsim \mathbf{z}_n^{\mathcal{M}}$ together imply $\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{z}_n^{\mathcal{M}}$ for all $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}}, \mathbf{z}_n^{\mathcal{M}} \in \mathcal{M}_n$]: This follows since, if $x_i = x_j$ implies $y_i = y_j$ and $y_i = y_j$ implies $z_i = z_j$ then clearly $x_i = x_j$ implies $z_i = z_j$.

Note that this definition is identical to the original equivalence relation in Definition 4.4 except that the implication is only in one direction. Note also that this partial ordering applies only to members of the set $\mathcal{M}_n$ not to general n-tuples. This is because the property of anti-symmetry would not hold for general n-tuples for example $(1,2) \succsim (\alpha, \beta)$ and $(\alpha, \beta) \succsim (1,2)$ but $(1,2) \neq (\alpha, \beta)$.

DEFINITION 4.14. The symbol $\succ$ will be used to mean *strictly succeeds*. That is $\mathbf{x} \succ \mathbf{y}$ means $\mathbf{x} \succsim \mathbf{y}$ and $\mathbf{x} \not\sim \mathbf{y}$. The symbol $\succ\!\!\succ$ will be used to mean *immediate successor* that is, if $\mathbf{x} \succ\!\!\succ \mathbf{z}$ then $\mathbf{x} \succ \mathbf{z}$ but there is no $\mathbf{y}$ such that $\mathbf{x} \succ \mathbf{y} \succ \mathbf{z}$. The symbols $\prec$, $\precsim$ and $\prec\!\!\prec$ will have their obvious meanings. The symbol $\mathbf{x}||\mathbf{y}$ will be used to mean *non-comparable* under the relation defined by $\succsim$, neither $\mathbf{x} \succsim \mathbf{y}$ nor $\mathbf{y} \succsim \mathbf{x}$ applies.

LEMMA 4.1. For all $m : 1 \leq m \leq n$, if $\mathbf{x}_n^{\mathcal{M}} = (x_1, \ldots, x_n) \in \mathcal{M}_n$ then $\mathbf{x}_m^{\mathcal{M}} = (x_1, \ldots, x_m)$ is a member of $\mathcal{M}_m$.

PROOF. If Definition 4.5 holds for x_i with $1 \leq i \leq n$ then clearly it holds for x_i with $1 \leq i \leq m$ if $m \leq n$. Therefore, $\mathbf{x}_m^{\mathcal{M}} \in \mathcal{M}_m$. □

This lemma states that the m-tuple obtained by choosing only the first m elements of a matching class is itself a matching class. (Note that this is not the case if the last m members are chosen. For example the last member of $(1, 2) \in \mathcal{M}_2$ is (2) which is not a member of $\mathcal{M}_1$).

LEMMA 4.2. For all $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$, if $\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}}$ then $\mathbf{x}_r^{\mathcal{M}} \succsim \mathbf{y}_r^{\mathcal{M}}$ for all $r \leq n$.

PROOF. By Definition 4.13 then since $(x_i = x_j) \Rightarrow (y_i = y_j)$ for all $i, j < n$ this is also true for all $i, j < r$ if $r \leq n$ by the same reasoning as for the previous lemma. □

4.6.1. A Consistent Enumeration for the Partial Ordering.

DEFINITION 4.15. A *consistent enumeration* of a partially ordered set S is a real valued function $f(\mathbf{x})$ where $\mathbf{x} \in S$ with the property that, for all $\mathbf{x}, \mathbf{y} \in S$ then $\mathbf{x} \succ \mathbf{y}$ implies $f(\mathbf{x}) > f(\mathbf{y})$.

THEOREM 4.4. The function $H(\mathbf{x}_n^{\mathcal{M}})$ provides a consistent enumeration of $\mathcal{M}_n$ uder the partial ordering given by Definition 4.13.

PROOF. It is necessary to show that for all $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$, then $\mathbf{x}_n^{\mathcal{M}} \succ \mathbf{y}_n^{\mathcal{M}}$ implies $H(\mathbf{x}_n^{\mathcal{M}}) > H(\mathbf{y}_n^{\mathcal{M}})$. The theorem is proved if, for all $\mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$, then $\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}}$ implies either $H(\mathbf{x}_n^{\mathcal{M}}) > H(\mathbf{y}_n^{\mathcal{M}})$ or $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$.

Consider constructing $\mathbf{x}_n^{\mathcal{M}}$ and $\mathbf{y}_n^{\mathcal{M}}$ by stages — that is, starting with a 1-tuple and adding an element onto the end to construct a 2-tuple and so on until an n-tuple is completed. Call the rth stage of construction $\mathbf{x}_r^{\mathcal{M}}$ and $\mathbf{y}_r^{\mathcal{M}}$ respectively. At the first stage of construction: $\mathbf{x}_1^{\mathcal{M}} = \mathbf{y}_1^{\mathcal{M}} = (1)$ and the heights of both are one. The proof proceeds by induction considering the

construction of stage r of $\mathbf{x}_r^{\mathcal{M}}$ by adding x_r to the end of $\mathbf{x}_{r-1}^{\mathcal{M}}$. There are three possibilities.

Case 1: $x_r = x_i$ for some $i < r$. By Definition 4.13, $x_r = x_i \Rightarrow y_r = y_i$ and, if $\mathbf{x}_{r-1}^{\mathcal{M}} = \mathbf{y}_{r-1}^{\mathcal{M}}$ then $\mathbf{x}_r^{\mathcal{M}} = \mathbf{y}_r^{\mathcal{M}}$. If $H(\mathbf{x}_{r-1}^{\mathcal{M}}) \geq H(\mathbf{y}_{r-1}^{\mathcal{M}})$ then $H(\mathbf{x}_r^{\mathcal{M}}) \geq H(\mathbf{y}_r^{\mathcal{M}})$ (since the height of neither change) and if $H(\mathbf{x}_{r-1}^{\mathcal{M}}) > H(\mathbf{y}_{r-1}^{\mathcal{M}})$ then $H(\mathbf{x}_r^{\mathcal{M}}) > H(\mathbf{y}_r^{\mathcal{M}})$.

Case 2: $x_r = \max_{i<r} x_i + 1$ and $y_r = \max_{i<r} y_i + 1$. In this case, trivially, if $\mathbf{x}_{r-1}^{\mathcal{M}} = \mathbf{y}_{r-1}^{\mathcal{M}}$ then $\mathbf{x}_r^{\mathcal{M}} = \mathbf{y}_r^{\mathcal{M}}$. If $H(\mathbf{x}_{r-1}^{\mathcal{M}}) \geq H(\mathbf{y}_{r-1}^{\mathcal{M}})$ then $H(\mathbf{x}_r^{\mathcal{M}}) \geq H(\mathbf{y}_r^{\mathcal{M}})$ (since the heights of both increase by one) and if $H(\mathbf{x}_{r-1}^{\mathcal{M}}) > H(\mathbf{y}_{r-1}^{\mathcal{M}})$ then $H(\mathbf{x}_r^{\mathcal{M}}) > H(\mathbf{y}_r^{\mathcal{M}})$.

Case 3: $x_r = \max_{i<r} x_i + 1$ and $y_r = y_j$ for some $j < r$. In this case, if $H(\mathbf{x}_{r-1}^{\mathcal{M}}) \geq H(\mathbf{y}_{r-1}^{\mathcal{M}})$ then $H(\mathbf{x}_r^{\mathcal{M}}) > H(\mathbf{y}_r^{\mathcal{M}})$.

Consider constucting $\mathbf{x}_n^{\mathcal{M}}$ and $\mathbf{y}_n^{\mathcal{M}}$ by stages. If only cases 1 and 2 occur then $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$. If case 3 occurs at any stage of construction then $H(\mathbf{x}_n^{\mathcal{M}}) > H(\mathbf{y}_n^{\mathcal{M}})$. □

COROLLARY 4.1. If $H(\mathbf{x}_n^{\mathcal{M}}) = H(\mathbf{y}_n^{\mathcal{M}})$ then either $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$ or $\mathbf{x}_n^{\mathcal{M}} || \mathbf{y}_n^{\mathcal{M}}$.

PROOF. The proof follows immediately from the construction in the proof of the previous theorem. If $H(\mathbf{x}_n^{\mathcal{M}}) = H(\mathbf{y}_n^{\mathcal{M}})$ and $\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}}$ then $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$. By the same reasoning, if $H(\mathbf{x}_n^{\mathcal{M}}) = H(\mathbf{y}_n^{\mathcal{M}})$ and $\mathbf{y}_n^{\mathcal{M}} \succsim \mathbf{x}_n^{\mathcal{M}}$ then $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$. Therefore, if $H(\mathbf{x}_n^{\mathcal{M}}) = H(\mathbf{y}_n^{\mathcal{M}})$ then either $\mathbf{x}_n^{\mathcal{M}} || \mathbf{y}_n^{\mathcal{M}}$ or $\mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$. □

4.6.2. The Hasse Diagram. A Hasse diagram is a way of visualising a partially ordered set. A Hasse diagram is constructed by plotting a partially ordered set S graphically in such a way that for all $\mathbf{x}, \mathbf{y} \in S$ if $\mathbf{x} \prec \mathbf{y}$ then $\mathbf{x}$ is further to the bottom of the diagram than $\mathbf{y}$. Further, if $\mathbf{x} \succ\!\succ \mathbf{y}$ then an arrow is drawn from $\mathbf{x}$ to $\mathbf{y}$.

Every Hasse diagram for $\mathcal{M}_n$ will have discrete levels defined by $H(\mathbf{x}_n^{\mathcal{M}})$ (since this has been shown to provide a consistent enumeration) and will have a singleton as the upper and lower levels defined, respectively, by $\mathcal{M}_n(\mathcal{F})$ (the only possible n-tuple in $\mathcal{M}_n$ with height n) and $\mathcal{M}_n(\mathcal{T})$ (the only possible

n-tuple in $\mathcal{M}_n$ with height 1). As an example, the Hasse diagram for $\mathcal{M}_4$ is shown in Figure 4.2.

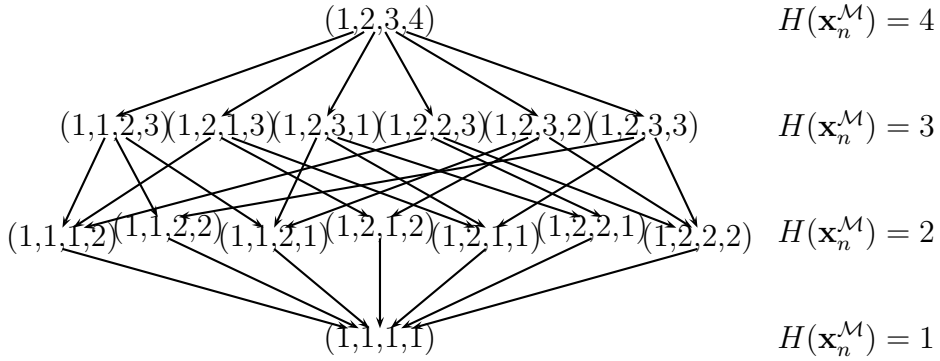

FIGURE 4.2. Hasse diagram for $\mathcal{M}_4$.

4.6.3. Partial (or Censored) Observations Related to Partial Ordering.

DEFINITION 4.16. The *censored observation function*, $C(\mathbf{x})$ is a function which acts on an n-tuple $\mathbf{x} = (x_1, \ldots, x_n)$ (this may be an n-tuple of observations or an n-tuple $\in \mathcal{M}_n$) to produce an n-tuple $\mathbf{y} = (y_1, \ldots, y_n)$ in such a way that if $\mathbf{y} = C(\mathbf{x})$ then

$$(x_i = x_j) \Rightarrow (y_i = y_j),$$

for all i and j. The domain of $C(\mathbf{x})$ is Ω^n and its range is the space of n-tuples of censored observations. For example, in the case of licence plate observations, the domain is the space of n-tuples of all possible licence plate observations and the range is the space of n-tuples of all possible partial licence plate observations.

The censored observation function is equivalent to the common sense notion of two or more observations of separate individuals which may be confused and appear to be the same individual. An example of a censored observation function would be correctly recording only part of a licence plate. By observing only part of the licence plate the same vehicle can never be recorded differently but different vehicles may be recorded as being the same.

THEOREM 4.5. The matching class of an n-tuple of censored observations $\precsim$ the n-tuple of the original observations. That is, for an n-tuple of observations $\mathbf{z}$ then

$$(\mathbf{x}_n^{\mathcal{M}} \sim \mathbf{z} \text{ and } \mathbf{y}_n^{\mathcal{M}} \sim C(\mathbf{z})) \Rightarrow (\mathbf{y}_n^{\mathcal{M}} \precsim \mathbf{x}_n^{\mathcal{M}}).$$

PROOF. This follows immediately from the fact that by Definition 4.4 then $(z_i = z_j) \Leftrightarrow (x_i = x_j)$. By Definitions 4.4 and 4.16 $(z_i = z_j) \Rightarrow (y_i = y_j)$. Therefore $(x_i = x_j) \Rightarrow (y_i = y_j)$ which is exactly the condition for the relationship $\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}}$ from Definition 4.13. □

4.7. The Exact and Relaxed Matching Functions

In this section the exact and relaxed matching functions are introduced and this is used to create an algebra of matching.

DEFINITION 4.17. The *exact matching function* $X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x})$, where $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ and $\mathbf{x}$ is an n-tuple of observations is defined as

$$X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}) = \begin{cases} 1 & \text{if and only if } \quad \mathbf{x} \sim \mathbf{y}_n^{\mathcal{M}} \\ 0 & \text{otherwise.} \end{cases}$$

This definition can be thought of as an indicator as to whether an observation is a equivalent to a particular matching class. The definition naturally extends from a single n-tuple of observations to a set of n-tuples as shown.

DEFINITION 4.18. The *exact matching function* $X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{Z})$, where $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$, $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ and all $\mathbf{z}_i$ are n-tuples of observations is defined as

$$X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{Z}) = \sum_{\mathbf{z} \in \mathbf{Z}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{z}).$$

In other words, the function counts the number of matches of type $\mathbf{y}_n^{\mathcal{M}}$ in the set of n-tuples $\mathbf{Z}$.

When used on a set of n-tuples, the exact matching functions simply counts the number of matches in a set of observations which belong to the given

matching class. The relaxed matching function allows the observations to belong to a matching class or any predecessor of that class.

DEFINITION 4.19. The *relaxed matching function* $R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x})$, where $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ ans $\mathbf{x}$ is an n-tuple of observations is defined as

$$R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}) = \begin{cases} 1 & \text{if and only if} \quad \mathbf{y}_n^{\mathcal{M}} \succeq \mathbf{x}_n^{\mathcal{M}} \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{x}_n^{\mathcal{M}} \sim \mathbf{x}$.

As previously this definition can be extended to a set of n-tuples as shown below.

DEFINITION 4.20. The *relaxed matching function* $R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{Z})$, where $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ and all $\mathbf{z}_i$ are n-tuples of observations is defined as

$$R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{Z}) = \sum_{\mathbf{z} \in \mathbf{Z}} R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{z}).$$

In other words, the relaxed matching function counts the number of n-tuples equivalent to a class $\mathbf{y}_n^{\mathcal{M}}$ or any successor class.

4.7.1. Some Proofs Relating to Exact and Relaxed Matches. It should be clear that the aim of the original problem (to find the number of genuine matches in a data set) is the problem of evaluating $X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$ where $\mathcal{S}$ is the set of all possible n-tuples of observations from Defintion 4.3. The problem is complicated by the fact that the observations $\mathcal{S}$ are not available and only the censored observations $C(\mathcal{S})$ are available to work with.

LEMMA 4.3. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be an n-tuple of observations and $\mathbf{y}_n^{\mathcal{M}} = (y_1, \ldots, y_n) \in \mathcal{M}_n$ be a matching class. If both are reordered in the same manner then the values of the exact and relaxed matching functions are unchanged. Swapping the elements i and j in both, giving the n-tuple $\mathbf{x}' = (x'_1, \ldots, x'_n)$ and the matching class $\mathbf{y}_n'^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{y}_n'^{\mathcal{M}} \sim (y'_1, \ldots, y'_n)$ (where $x'_i = x_j$, $x'_j = x_i$ and $x'_k = x_k$ for all $k \neq i, j$ and, in addition, $y'_i = y_j$, $y'_j = y_i$

and $y'_k = y_k$ for all $k \neq i, j$) [2] does not change the value of the exact or relaxed matching functions. In other words, $X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}) = X(\mathbf{y}_n'^{\mathcal{M}}, \mathbf{x}')$ and $R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}) = R(\mathbf{y}_n'^{\mathcal{M}}, \mathbf{x}')$.

PROOF. Consider first the exact matching function. It is equal to 1 if and only if $\mathbf{y}_n^{\mathcal{M}} \sim \mathbf{x}$ which implies in turn $(y_k = y_l) \Leftrightarrow (x_k = x_l)$. It can be easily seen that if y_i and y_j are swapped and simultaneously x_i and x_j are swapped then the truth (or otherwise) of this condition remains unchanged. Thus the lemma is proved for the exact matching function. The argument for the relaxed matching function is exactly the same but the implication is in one direction only since the relaxed matching function is 1 if and only if $(y_k = y_l) \Rightarrow (x_k = x_l)$. □

It should be noted that this lemma also applies to the relaxed and exact matching functions operating on sets of n-tuples when each n-tuple in the set is reordered in the manner described in the lemma.

LEMMA 4.4. Given a set of n-tuples of observations $\mathbf{Z} = \{\mathbf{z}_1, \ldots \mathbf{z}_m\}$ and a matching class $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ then

$$X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{Z}) = R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{Z}) - \sum_{\mathbf{y}_n^{\mathcal{M}}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{Z}),$$

where the sum is over those elements $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{y}_n^{\mathcal{M}} \prec \mathbf{x}_n^{\mathcal{M}}$.

PROOF. Rewrite the equation as

$$\sum_{\mathbf{z} \in \mathbf{Z}} \sum_{\mathbf{y}_n^{\mathcal{M}}} X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{z}) = \sum_{\mathbf{z} \in \mathbf{Z}} R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{z}),$$

where, again, the inner sum on the left-hand side is over all elements $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{y}_n^{\mathcal{M}} \precsim \mathbf{x}_n^{\mathcal{M}}$.

Therefore it is sufficient to prove that for all possible n-tuples $\mathbf{z}$,

$$R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{z}) = \sum_{\mathbf{y}_n^{\mathcal{M}}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{z}), \tag{4.5}$$

[2] Note that the n-tuple $(y'_1, \ldots, y'_n)$ is not necessarily a member of $\mathcal{M}_n$ hence the $\sim$ sign not the $=$ sign.

where $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{y}_n^{\mathcal{M}} \precsim \mathbf{x}_n^{\mathcal{M}}$.

Two cases are possible — either $R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{z})$ equals zero or it equals one.

Case 1: $R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{z}) = 0$.

In this case by Definitions 4.17 and 4.19 $X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{z}) = 0$ for all $\mathbf{y}_n^{\mathcal{M}} \precsim \mathbf{x}_n^{\mathcal{M}}$. Thus $\sum_{\mathbf{y}_n^{\mathcal{M}}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{z})$ must equal zero and equation (4.5) is true.

Case 2: $R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{z}) = 1$.

By Definition 4.19 there exists $\mathbf{y}_n^{\mathcal{M}} \sim \mathbf{z}$ such that $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ and $\mathbf{y}_n^{\mathcal{M}} \precsim \mathbf{x}_n^{\mathcal{M}}$ (note that by the definition of $\mathcal{M}_n$ there can be only one such $\mathbf{y}_n^{\mathcal{M}} \sim \mathbf{z}$). Thus $\sum_{\mathbf{y}_n^{\mathcal{M}}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{z}) = 1$ where the sum is over all $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{y}_n^{\mathcal{M}} \precsim \mathbf{x}_n^{\mathcal{M}}$ and therefore equation (4.5) holds and the lemma is proved. □

COROLLARY 4.2. For all n-tuples $\mathbf{x}$,

$$R(\mathcal{M}_n(\mathcal{T}), \mathbf{x}) = X(\mathcal{M}_n(\mathcal{T}), \mathbf{x})$$

PROOF. This follows since there are no $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{y}_n^{\mathcal{M}} \prec \mathcal{M}_n(\mathcal{T})$. Therefore, in the previous lemma, the term $\sum_{\mathbf{y}_n^{\mathcal{M}}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{z}) = 0$. □

To proceed with the theory two definitions are necessary which will be used in the next lemma. The definitions and lemmas which follow deal with the idea of breaking tuples (both tuples of observations and matching classes) into sub-tuples.

DEFINITION 4.21. Define $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, i)$ as the tuple of indices within $\mathbf{x}_n^{\mathcal{M}}$ which have the value i. The elements of $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, i)$ are ordered in increasing value.

$$\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, i) = (s(1), s(2), \ldots, s(m))$$

where the $s(j)$ are those elements of $\mathbf{x}_n^{\mathcal{M}}$ such that $x_{s(j)} = i$ and, obviously, m is the number of such elements. The $s(j)$ are ordered such that $s(j) < s(k)$ if $j < k$.

An example of this definition in use may help. If $\mathbf{x}_n^{\mathcal{M}} = (1, 2, 1, 1, 3, 2)$ then $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, 1) = (1, 3, 4)$ (since the first, third and fourth elements of $\mathbf{x}_n^{\mathcal{M}}$ are equal to one). Similarly, $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, 2) = (2, 6)$ and $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, 3) = (5)$.

DEFINITION 4.22. Define $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ as a set of tuples of observations, derived from $\mathcal{S}$ (the set of all possible n-tuples of observations as given in Definition 4.3). The set $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ is given by the Cartesian product,

$$\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i) = \prod_{j \in \mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, i)} \mathbf{S}_j,$$

where $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, i) = (s(1), s(2), \dots)$ is as given in Definition 4.21 and the product is over the elements $s(i)$ of the tuple.

In other words, the tuple $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, i)$ picks out a selection of those sites which, for a given matching class $\mathbf{x}_n^{\mathcal{M}} = (x1, \dots, x_n)$, correspond to an element $x_j = i$. The set of observations $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ is the set of all possible tuples of observations made at those sites only. For example, if $\mathbf{x}_n^{\mathcal{M}} = (1, 2, 1, 2)$ then $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, 1)$ is the set of all pairs of observations made at sites one and three and $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, 2)$ is the set of all pairs of observations made at sites two and four.

LEMMA 4.5. Given a set $\mathcal{S}$ of all possible n-tuples from a set of observations made over n sites (as defined in Definition 4.3), the number of relaxed matches of class $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ in $\mathcal{S}$ is given by

$$R(\mathbf{x}_n^{\mathcal{M}}, \mathcal{S}) = \prod_{i=1}^{h} X(\mathcal{M}_{m(i)}(\mathcal{T}), \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)),$$

where $h = H(\mathbf{x}_n^{\mathcal{M}})$, $m(i) = \#\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ and $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ is given by Definition 4.22.

PROOF. Firstly, from Lemma 4.3, the value of $R(\mathbf{x}_n^{\mathcal{M}}, \mathcal{S})$ is unchanged if $\mathbf{x}_n^{\mathcal{M}}$ and $\mathcal{S}$ are reordered in the same way. Therefore, assume, without loss of generality that $\mathbf{x}_n^{\mathcal{M}}$ is ordered such that $x_i \leq x_j$ for all $i < j$ and $\mathcal{S}$ has been reordered in the same way. By this property, the tuples in $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ must be made from elements which were adjacent in the original n-tuples in $\mathcal{S}$. Define

$\mathbf{S}(i)$ such that $\mathbf{S}(i) \subseteq \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ and $\mathbf{S}(i)$ consists of those tuples $\mathbf{x} \in \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ such that $\mathbf{x} \sim \mathcal{M}_{m(i)}(\mathcal{T})$. Clearly, because of the way these tuples were chosen then

$$X(\mathcal{M}_{m(i)}(\mathcal{T}), \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)) = \#\mathbf{S}(i).$$

In other words, $\mathbf{S}(i)$ is a set of all those tuples which appear to be observations of the same individual at all sites picked out by $\mathbf{X}(\mathbf{x}_n^{\mathcal{M}}, i)$. Consider the set of n-tuples $\mathbf{Y}$ formed by the Cartesian product of $\mathbf{S}(i)$.

$$\mathbf{Y} = \prod_{i=1}^{h} \mathbf{S}(i).$$

An n-tuple $\mathbf{y} \in \mathbf{Y}$ is an n-tuple of observations with one observation from each of the original n sites. It must now be shown that

$$\mathbf{Y} = \{y \in \mathcal{S} : R(\mathbf{x}_n^{\mathcal{M}}, y) = 1\},$$

or, by the definition of $R(\mathbf{x}_n^{\mathcal{M}}, y)$ (Definition 4.19), $\mathbf{Y}$ is the set of all those n-tuples in $\mathcal{S}$ for which $\mathbf{y} \precsim \mathbf{x}_n^{\mathcal{M}}$. This is equivalent to the claim that for all $\mathbf{y} \in \mathbf{Y}$ then $\mathbf{y} \precsim \mathbf{x}_n^{\mathcal{M}}$ and for all $\mathbf{z} \in \mathcal{S}/\mathbf{Y}$ then $\mathbf{z} \not\precsim \mathbf{x}_n^{\mathcal{M}}$. The set $\mathbf{S}(i)$ was constructed from sites corresponding to the set of all elements in $\mathbf{x}_n^{\mathcal{M}}$ for which $x_j = x_k = i$. Further, $\mathbf{S}(i)$ consists of all elements of observations where $y_j = y_k$. Therefore, for any given i then $(x_j = x_k = i) \Rightarrow (y_j = y_k)$ for all $\mathbf{y} \in \mathbf{S}(i)$. Similarly, if $x_j = x_k = i$ then all tuples such that $y_j = y_k$ are in $\mathbf{S}(i)$. Since $\mathbf{Y}$ is the Cartesian product of all such $\mathbf{S}(i)$ then all tuples in $\mathcal{S}$ such that $\mathbf{y} \precsim \mathbf{x}_n^{\mathcal{M}}$ must be in $\mathbf{Y}$ and all $\mathbf{y} \in \mathbf{Y}$ must be such that $\mathbf{y} \precsim \mathbf{x}_n^{\mathcal{M}}$.

Therefore,

$$R(\mathbf{x}_n^{\mathcal{M}}, \mathcal{S}) = \#\mathbf{Y} = \#\prod_{i=1}^{h} \mathbf{S}(i) = \prod_{i=1}^{h} \#\mathbf{S}(i) = \prod_{i=1}^{h} X(\mathcal{M}_{m(i)}(\mathcal{T}), \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)),$$

and thus the lemma is proved. □

The lemma can best be understood by example. Consider $\mathbf{x}_n^{\mathcal{M}} = (1, 2, 1, 1, 2)$. Any observation $\mathbf{y} \precsim \mathbf{x}_n^{\mathcal{M}}$ must have $y_1 = y_3 = y_4$ and $y_2 = y_5$. The set $\mathbf{S}(1)$ is the set of all triples of observations from sites 1, 3 and 4 meeting the first condition. The set $\mathbf{S}(2)$ is the set of all pairs of observations from sites 2 and

5 meeting the second condition. Therefore, the cartesian product $\mathbf{S}(1) \times \mathbf{S}(2)$ reordered must contain all $\mathbf{y} \precsim \mathbf{x}_n^{\mathcal{M}}$ and no $\mathbf{y} \not\precsim \mathbf{x}_n^{\mathcal{M}}$.

COROLLARY 4.3.

$$R(\mathcal{M}_n(\mathcal{F}), \mathcal{S}) = \prod_{i=1}^{n} \#\mathbf{S}_i,$$

or, in other words, the number of relaxed matches against the false matching class in a set of observations is simply the number of observations.

PROOF. This follows from the fact that, for $\mathcal{M}_n(\mathcal{F})$ then $\mathbf{X}(\mathcal{M}_n(\mathcal{F}), i)$ is the set of the single elements $\mathbf{X}(\mathcal{M}_n(\mathcal{F}), 1) = (1)$, $\mathbf{X}(\mathcal{M}_n(\mathcal{F}), 2) = (2)$ and so on. Obviously, for just a single site, then $\mathcal{M}_1 = \{(1)\}$ and all observations at site i must be a member of $\mathbf{S}(\mathcal{M}_n(\mathcal{F}), i)$. So $X(\mathcal{M}_{m(i)}(\mathcal{T}), \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)) = \#\mathbf{S}_i$ and the corollary then follows from the lemma. □

DEFINITION 4.23. For a given censoring function $C(\mathbf{x})$ the probability $p(n)$ is defined for $n \geq 1$ as

$$p(n) = \mathbb{P}\left[C(\mathbf{x}) \sim \mathcal{M}_n(\mathcal{T}) | \mathbf{x} \sim \mathcal{M}_n(\mathcal{F})\right],$$

for n-tuples of observations $\mathbf{x} = (x_1, \ldots, x_n)$. Note that $\mathbf{x}$ is chosen in such a way that $x_i = f(\omega_k)$ where the $f(\omega_k)$ is the observation function from Definition 4.1 and the $\omega_k \in \Omega$ are chosen from the same distribution as the genuine observations in the real data $\mathbf{S}$.

Note that implicit in this is the assumption that the probability $p(n)$ does not depend on the particular sites chosen from a subset of sites. This is a reasonable assumption for the particular problem chosen (that of vehicle licence plates). However, it might be criticised on a number of grounds. For example, in the UK, a wealthy neighbourhood might have a preponderance of newer vehicles with similar year letters and a site might be near such a neighbourhood. Also it is conceivable that military vehicles (which can have different plates) might distort this assumption. Note also that by this definition then $p(1) = 1$ — this follows from the fact that $\mathcal{M}_1(\mathcal{T}) = \mathcal{M}_1(\mathcal{F})$.

LEMMA 4.6. Given the censoring function $C(\mathbf{x})$ and some n-tuple of observations $\mathbf{x}$ then

$$\mathbb{P}\left[C(\mathbf{x}) \sim \mathcal{M}_n(\mathcal{T})\right] = p(h),$$

where $h = H(\mathbf{y}_n^{\mathcal{M}})$ and $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{y}_n^{\mathcal{M}} \sim \mathbf{x}$ and $\mathbf{x}$ is randomly chosen in the same manner as in Definition 4.23. In other words, the probability that, after censoring, a set of observations appears to be a true match is $p(h)$.

PROOF. Since $\mathbf{x}$ has h distinct elements then some h-tuple $\mathbf{z}$ can be formed by choosing elements from $\mathbf{x}$ such that $\mathbf{z} \sim \mathcal{M}_h(\mathcal{F})$. From Definition 4.23 then $\mathbb{P}\left[C(\mathbf{z}) \sim \mathcal{M}_h(\mathcal{T})\right] = p(h)$. If $C(\mathbf{z}) \sim \mathcal{M}_h(\mathcal{T})$ then $C(\mathbf{x}) \sim \mathcal{M}_n(\mathcal{T})$ and therefore $\mathbb{P}\left[C(\mathbf{x}) \sim \mathcal{M}_n(\mathcal{T})\right] = \mathbb{P}\left[C(\mathbf{z}) \sim \mathcal{M}_h(\mathcal{T})\right]$ and thus the lemma is proved. □

LEMMA 4.7. For a set of n-tuples of observations $\mathbf{Z}$ with a censoring function $C(\mathbf{Z})$ then an unbiased estimator for the number of true matches in the set of observations $t = X(\mathcal{M}_n(\mathcal{T}), \mathbf{Z})$ is given by

$$\hat{t} = X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{Z})) - \sum_{\mathbf{x}_n^{\mathcal{M}}} X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{Z}) p(h),$$

where $h = H(\mathbf{x}_n^{\mathcal{M}})$ and the sum is over $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{x}_n^{\mathcal{M}} \neq \mathcal{M}_n(\mathcal{T})$.

PROOF. Firstly define $\mathbf{Y}(\mathbf{x}_n^{\mathcal{M}})$ as the set of all those $\mathbf{z} \in \mathbf{Z}$ such that $C(\mathbf{z}) \sim \mathcal{M}_n(\mathcal{T})$ and $\mathbf{z} \sim \mathbf{x}_n^{\mathcal{M}}$. For a given $\mathbf{z} \sim \mathbf{x}_n^{\mathcal{M}}$ then $\mathbb{P}\left[\mathbf{z} \in \mathbf{Y}(\mathbf{x}_n^{\mathcal{M}})\right] = p(h)$ by Lemma 4.6. The number of elements $\mathbf{z} \sim \mathbf{x}_n^{\mathcal{M}}$ is given by $X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{Z})$. It therefore follows that

$$\hat{y} = X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{Z}) p(h). \tag{4.6}$$

is an unbiased estimator for $\#(\mathbf{Y}(\mathbf{x}_n^{\mathcal{M}}))$ since it is a sum of probabilities and expectation is a linear operator.

$$X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{Z})) = \sum_{\mathbf{x}_n^{\mathcal{M}}} \#\mathbf{Y}(\mathbf{x}_n^{\mathcal{M}}),$$

where the sum is over all $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$.

$$\begin{aligned} X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{Z})) &= \#\mathbf{Y}(\mathcal{M}_n(\mathcal{T})) + \sum_{\mathbf{x}_n^{\mathcal{M}}} \#\mathbf{Y}(\mathbf{x}_n^{\mathcal{M}}) \\ &= X(\mathcal{M}_n(\mathcal{T}), \mathbf{Z}) + \sum_{\mathbf{x}_n^{\mathcal{M}}} \#\mathbf{Y}(\mathbf{x}_n^{\mathcal{M}}) \\ X(\mathcal{M}_n(\mathcal{T}), \mathbf{Z}) &= X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{Z})) - \sum_{\mathbf{x}_n^{\mathcal{M}}} \#\mathbf{Y}(\mathbf{x}_n^{\mathcal{M}}) \end{aligned}$$

where the sum is over $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{x}_n^{\mathcal{M}} \neq \mathcal{M}_n(\mathcal{T})$. The estimator (4.6) substituted into this equation completes the lemma. Because the estimator is a sum of unbiased estimators then the estimator for $X(\mathcal{M}_n(\mathcal{T}), \mathbf{Z})$ is itself unbiased. □

4.7.2. Estimating $p(n)$ in Real Data. It may be thought that estimating $p(n)$ from Definition 4.23 is a major problem. It seems to require knowledge of the distribution of the uncensored observations. A number of strategies for estimating $p(n)$ are possible. The best strategy will depend on the particular nature of the problem under study. For the case of partial licence plate observations two sensible strategies are available.

The first strategy is to estimate the value of $p(n)$ from knowledge of the distribution of the licence plates themselves. Studies of UK plates in the form `ABC123X` where `X` is a year letter show that the digits have an approximately flat distribution, the year letter exhibits a complex distribution which depends on vehicle sales in that year and the rate at which old vehicles retire from service and the initial letters exhibit a distribution which depends upon where the car was purchased.

A common method for reading partial plates in this type of plate is to take the year letter and the three digits. The probability that two randomly chosen different vehicles have the same digits in their plate is approximately $\frac{1}{1000}$. The probability of two randomly chosen vehicles having the same year letter can be estimated as $\sum_{x \in A} f(x)^2$ where A is the set of all year letters and $f(x)$ is the fraction of surveyed vehicles with the given year letter. Therefore

the probability of two partial plates matching can be estimated as

$$\widehat{p(2)} = \frac{1}{1000} \sum_{x \in A} f(x)^2.$$

Values of $p(n)$ for $n > 2$ can be estimated with similar assumptions. Of course, this method depends on the particular details of how plates are allocated and is not of general interest.

The second strategy is based upon analysing the data recorded. Often there are two survey sites where it is known that all the vehicles must be different — for example, two sites that are fifty minutes drive apart in a half hour survey. Any matches between these two sites must be false matches. From this then an estimator for $p(2)$ is given by

$$\widehat{p(2)} = \frac{T}{N_1 N_2}.$$

where T is the total number of matches and N_1, N_2 are the number of observations at the first and second survey sites. Values of $p(n)$ for $n > 2$ can be estimated with similar assumptions if there are n widely separated survey sites. Alternatively, the values of $p(2)$ and $p(3)$ could be used to estimate the higher order probabilities.

It is important to note that a good estimate for $p(n)$ particularly for $n = 2$ and $n = 3$ is extremely important to the estimates made by this method. Section 5.8.1 discusses the estimation problem in a real data set.

4.8. An Algorithm for Estimating False Matches

It is not immediately obvious, but from the above Lemmas 4.4, 4.5 and 4.7 a procedure can be created to estimate $X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$ — the number of true matches in a set of observations over the set of sites $\mathbf{S}$. This was the original aim of the false match problem in licence plate data. The idea is to take the problem and reduce it to a number of sub-problems of finding false matches in a lesser number of data sets. Eventually, the problem will "bottom out" when there is only one data set since $\mathcal{M}_1(\mathcal{T}) = \{(1)\}$ and $X(\mathcal{M}_1(\mathcal{T}), \mathcal{S})$ is simply the number of members in the single data set $\#\mathcal{S}$.

Lemma 4.5 allows estimation of $X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$ from $X(\mathcal{M}_n(\mathcal{T}), C(\mathcal{S}))$ and $X(\mathbf{x}_n^{\mathcal{M}}, \mathcal{S})$ in classes other than $\mathcal{M}_n(\mathcal{T})$. The quantity $X(\mathcal{M}_n(\mathcal{T}, C(\mathcal{S}))$ can be directly measured since it is the number of tuples of observations which are the same in the censored data. Thus, from this data, the number of true matches can be estimated from the number of exact matches in all other matching classes.

From Lemma 4.4 these matches can be calculated exactly if the number of relaxed matches $R(\mathbf{x}_n^{\mathcal{M}}, \mathcal{S})$ is known and also the number of exact matches in all successor matching classes is known.

From Lemma 4.5 the number of relaxed matches of a particular type can be calculated if the number of exact true matches in a subset of sites is known. The value of $R(\mathcal{M}_n(\mathcal{F}), \mathcal{S})$ is given by Corollary 4.3. From Corollary 4.2, then $R(\mathcal{M}_n(\mathcal{T}), \mathcal{S}) = X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$, which is the quantity desired. For all other values of $\mathbf{x}_n^{\mathcal{M}}$, Lemma 4.5 allows the calculation of $R(\mathbf{x}_n^{\mathcal{M}}, \mathcal{S})$ in terms of $X(\mathcal{M}_m(\mathcal{T}), \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i))$ where $m < n$ and $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ is, from Defintion 4.22, a set of tuples defined over some subset of the original sites.

These lemmas must be used in conjunction with computer algebra to provide a solution. In the interests of clarity, a brief example is given in the next subsection.

Therefore, if $p(n)$ can be estimated, the problem of estimating $X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$ is solved by the procedure defined in Table 4.5.

4.8.1. Computer Algebra Example. In this section only a short hand notation will be used in order to prevent the expressions used becoming unwieldy. Consider the problem with only three sites. Let t_{123} be the true number of matches which occur between all sites and by extension t_{13} be the true number of matches which occur between sites one and three only. Similarly, let $\mathbf{Z}_{123}$ be the set of all n-tuples of observations over all sites and let $\mathbf{Z}_2$ be the set of all observations (strictly, 1-tuples of observations) at site two only. Let s_{123} be the observed number of matches across all three sites.

(1) Calculate from the data, $X(\mathcal{M}_n(\mathcal{T}), C(\mathcal{S}))$ for all n sites — this is simply a matter of counting the number *true matches* observed in the censored data.
(2) Begin with Lemma 4.7 and use computer algebra (see example) to expand this using Lemmas 4.4, and 4.5 to give an expression in terms of $X(\mathcal{M}_n(\mathcal{T}), C(\mathcal{S}))$, $X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$, $X(\mathbf{x}_n^{\mathcal{M}}, \mathcal{S})$ and $p(n)$.
(3) Again using computer algebra, gather all the terms which are $X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$ (the quantity to be found) on the left hand side — these terms will all be functions of $p(k)$ where $1 < k \leq n$.
(4) Steps 1 to 3 produce an equation for $X(\mathcal{M}_n(\mathcal{T}), \mathcal{S})$ in terms of $p(k)$, $R(\mathcal{M}_n(\mathcal{F}), \mathcal{S})$ (given by Corollary 4.3) and $X(\mathcal{M}_m(\mathcal{T}), \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i))$ where $m < n$ and $\mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$ is the set of tuples of observations over some subset of sites.
(5) For each of the terms $X(\mathcal{M}_m(\mathcal{T}), \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i))$ then if $m = 1$ the answer is trivial. If $m > 1$ then use this whole procedure from step 1 with $n = m$ and $\mathcal{S} = \mathcal{S}(\mathbf{x}_n^{\mathcal{M}}, i)$. In other words, the problem has become a sub-problem with a reduced number of sites.

TABLE 4.5. Algorithm for correcting false matches.

To relate this to the previous notation:

$$\mathbf{Z}_{123} = \mathcal{S}$$
$$t_{123} = X(\mathcal{M}_3(\mathcal{T}), \mathcal{S})$$
$$s_{123} = X(\mathcal{M}_3(\mathcal{T}), C(\mathcal{S}))$$

Now, in this new notation, beginning from Lemma 4.7 then

$$\hat{t}_{123} = s_{123} - \sum_{\mathbf{x}_n^{\mathcal{M}}} X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{Z}_{123}) p(h),$$

where $h = H(\mathbf{x}_n^{\mathcal{M}})$ and the sum is over $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ such that $\mathbf{x}_n^{\mathcal{M}} \neq \mathcal{M}_n(\mathcal{T})$. The sum in full is, therefore,

$$\begin{aligned}\hat{t}_{123} = s_{123} &- X((1,1,2), \mathbf{Z}_{123}) p(2) \\ &- X((1,2,1), \mathbf{Z}_{123}) p(2) \\ &- X((1,2,2), \mathbf{Z}_{123}) p(2) \\ &- X((1,2,3), \mathbf{Z}_{123}) p(3).\end{aligned}$$

This can be thought of in terms of "The estimated number of true matchs is the number of observed matches minus matches which appear to be true matches in any one of these four ways."

Next, the terms $X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{Z}_{123})$ can be expanded using Lemma 4.4. For example,

$$\begin{aligned}X((1,2,1), \mathbf{Z}_{123}) &= R((1,2,1), \mathbf{Z}_{123}) - X((1,1,1), \mathbf{Z}_{123}) \\ &= R((1,2,1), \mathbf{Z}_{123}) - t_{123}.\end{aligned}$$

Similarly,

$$\begin{aligned}X((1,2,3), \mathbf{Z}_{123}) = \; & R((1,2,3), \mathbf{Z}_{123}) - X((1,2,2), \mathbf{Z}_{123}) - X((1,1,2), \mathbf{Z}_{123}) \\ & - X((1,2,1), \mathbf{Z}_{123}) - X((1,1,1), \mathbf{Z}_{123}) \\ = \; & \#\mathbf{Z}_{123} - R((1,2,2), \mathbf{Z}_{123}) - R((1,1,2), \mathbf{Z}_{123}) \\ & - R((1,2,1), \mathbf{Z}_{123}) + 3t_{123},\end{aligned}$$

where Corrollary 4.3 has been used to get the substitution for $\#\mathbf{Z}_{123}$ in the second line.

Now, terms like $R((1,2,1), \mathbf{Z}_{123})$ can be expanded using Lemma 4.5 so that, for example,

$$R((1,2,1), \mathbf{Z}_{123}) = t_{13} \# \mathbf{Z}_2.$$

After completing all these expansions then,

$$\begin{aligned}\hat{t}_{123} = s_{123} &- t_{12}\#\mathbf{Z}_3 p(2) - t_{13}\#\mathbf{Z}_2 p(2) - t_{23}\#\mathbf{Z}_2 p(2) \\ &+ 3t_{123}p(2) - \#\mathbf{Z}_{123}p(3) + t_{12}\#\mathbf{Z}_3 p(3) \\ &+ t_{13}\#\mathbf{Z}_2 p(3) \qquad\qquad +t_{23}\#\mathbf{Z}_2 p(3) - 3t_{123}p(3).\end{aligned}$$

Rearranging the terms in t_{123} gives

$$\begin{aligned}\hat{t}_{123}(1 + 3p(3) - 3p(2)) = s_{123} &- \#\mathbf{Z}_{123}p(3) - t_{12}\#\mathbf{Z}_3 p(2) \\ &- t_{13}\#\mathbf{Z}_2 p(2) - t_{23}\#\mathbf{Z}_2 p(2) + t_{12}\#\mathbf{Z}_3 p(3) \\ &+ t_{13}\#\mathbf{Z}_2 p(3) + t_{23}\#\mathbf{Z}_1 p(3)\end{aligned}$$

The only unknown terms here are t_{12}, t_{23} and t_{13}. These can be found by repeating the same procedure on the problem with just two sites. The result obtained is that

$$\hat{t}_{12}(1 + p(2)) = s_{12} - \#\mathbf{Z}_{12}p(2),$$

and, naturally, similar equations can be found for t_{23} and t_{13}. With just three sites, the problem can be solved explicitly without computer algebra. With six sites this becomes sufficiently difficult that a computer is required to make the subsitutions above.

4.9. Simulation Results

The procedure developed in the previous section has been implemented in C++ and tried both on real data (from roadside surveys) and on simulated data. The simulated data is also presented as if it were a roadside survey. Results on the real data are not presented here since it is impossible to know the correct answer for this data.

No. Veh.	1–2	1–3	1–4	1–5	1–6	Av.Raw Matches	σ Raw Matches	Av.Cor. Matches	σ Cor. Matches
1000	10					111.4	8.5	11.4	8.5
2000	10					411.8	19.5	11.8	19.5
1000	100					199.2	12.0	99.2	12.0
1000	200					302.3	7.7	202.3	7.7
1000	500					596.6	12.3	496.7	12.3
1000	0	10				21.9	4.6	9.3	3.3
1000	500	10				73.8	7.5	10.2	6.2
1000	100	100				152.1	8.5	101.9	7.5
1000	500	250				388.3	22.7	253.2	20.1
1000	0	500				667.2	24.9	506.0	22.3
1000	0	0	100			154.6	26.6	104.0	22.6
1000	100	100	100			164.4	11.4	97.7	9.3
500	100	100	100			140.7	19.3	105.8	17.4
1000	500	250	100			207.8	29.7	106.1	23.7
500	10	10	10	10		14.2	2.2	10.5	1.8
1000	10	10	10	10		17.4	4.1	9.4	2.8
500	50	50	50	50		71.3	14.3	47.8	12.3
500	100	100	100	100		151.9	26.9	92.0	22.3
1000	0	0	0	100		177.6	29.9	103.4	22.6
1000	100	100	100	100		222.2	61.5	111.0	46.7
1000	0	0	0	0	10	21.2	13.4	12.3	9.9
500	0	0	0	0	100	152.6	45.5	92.2	37.3
1000	0	0	0	0	100	214.6	58.0	103.5	40.2
1000	100	100	100	100	100	289.8	88.4	101.3	55.0

TABLE 4.6. Simulation results — all performed over twenty runs with 10,000 distinct vehicle types.

Table 4.6 shows simulation results for between two and six observation sites. The table is to be interpreted as follows. Num. Veh. refers to the total number of observations at each of the sites (in these simulations, there are the same number of vehicles in each data set). The five columns of the form $1-n$ refer to the number of vehicles which genuinely went from site one to site n visiting all sites in between. If this column is blank it means that there was no site n. For example, if $1-2=100$, $1-3=200$ and $1-4$ is blank. This means that 100 vehicles travelled between site one and site two, 200 vehicles travelled between sites one, two and three and there were only three sites. Note that these are cumulative so that if $1-2=20$ and $1-3=10$ this means that 30 vehicles in total went from site one to site two and 10 of them continued to site three. Thus the first experiment is two sites, 1000 vehicles at each for which there were ten vehicles which were genuinely seen at both sites. Note that in every experiment, the number of different vehicle types was set at $10,000$ with a flat distribution (equal numbers of vehicles seen at each site). It should be clear that the desired answer from the correction process is the rightmost figure in these columns.

Each experiment is repeated twenty times with simulated data being generated anew each time. The correction process has no random element and will always give the same result for the same data. The mean raw number of matches is given — this is the total number of n-tuples which were seen to have the same value for each observation at every site (averaged over the twenty simulation runs). Note that, because of the combinatorial nature of the procedure, this could, in principle, be much larger than the number of vehicles in any of the data sets. The sample standard deviation (σ) is given for the raw matches. The mean estimated correct number of matches is then given (again averaged over the twenty simulations). The sample standard deviation σ is then given for the twenty corrected matches. It is clear that the most important test is that the mean corrected number of matches is as near to correct as possible. However, it should also be kept in mind that in reality, a researcher

could only run the matching procedure once on any given set of data — so it is also important that σ is as low as possible. A significant improvement to the method would be to estimate the variance as well as producing an estimate. If this were achieved, then the researcher could have some idea as to the likely accuracy of the corrected results. It should also be noted that in every experiment, the chances of any given two vehicles being a false match is 1 in $10,000$ with a flat distribution (so the chance of three distinct vehicles having the same partial plate is the square of this). In fact this is an extremely pessimistic assumption since four digits of a licence plate would be the least that a partial plate survey was likely to capture. A significant weakness of the method is that it requires a good estimate for $p(n)$. (In fact, it is mainly significant for lower values of n with $p(2)$ being the most important).

The first five rows are all results on just two test sites. This procedure is not the ideal one to use for estimates on matches between just two sites and the work of other authors in the field should be used in such a circumstance (especially if extra information such as travel time is available). However, these results are included here for completeness. In the two site case, the number of corrected matches is simply obtained by subtracting $n^2/10,000$ [3] from the raw matches (where n is the number of vehicles at each site).

To take an example, in the first experiment, the average number of raw matches over the ten runs is 111.4 and $n = 1000$. The average number of corrected matches is 11.4 (obtained by subtracting $100 = n^2/10,000$ from 111.4). This is close to the correct answer of 10. However, it should be noticed that the σ is high in comparison to the actual answer. In this case, the σ is 8.5 which is of the same order of magnitude as the answer. This is to be expected since we are looking for only 10 true matches in over 110 observed matches. If we increase the number of vehicles to 2000 then, as would be expected, the

[3] Strictly, this is not exact. There is also a correction factor of $1/(1+p(2))$ but this can generally be ignored since $p(2) \ll 1$ in real applications.

number of false matches goes up (to approximately 400) and the σ also rises (to almost 20).

The next five rows of results are all over three sites. In the first of these, 10 vehicles travel between all three and all other matches are coincidence. 1000 vehicles are observed at all sites. The mean corrected match across all sites 9.3 is close to the actual answer of 10 and the σ is lower than in the two site case. However, when the same experiment is run with 500 vehicles travelling from sites one to two in addition to 10 vehicles travelling from sites two to three, the σ increases markedly (it almost doubles). In all cases with three sites, the mean is a good estimate and the σ is generally low enough that a good estimate can be expected.

The next four rows of results are for experiments made over four sites. The first experiment has 100 vehicles which visit all four. The mean corrected match is 104 (very close) and the σ is only 22. It is hard to explain why this σ actually falls in the next experiment when more vehicles are genuinely seen in common between the other sites. This fall in σ is puzzling. In all cases the mean of the predictions is approximately correct (the worst performance being in the case of the fourth experiment when the mean was 106 not 100).

The next six rows of results are experiments made over five sites. Again, the mean corrected results are approximately correct. However, in the worst case, the mean is 11 too high and the σ in the result is 46.7 which is comparable to the level of the effect being observed. In this case approximately 120 false matches are being removed each time. However, previous experiments have been able to correct for a greater proportion of false matches with less σ in the result.

The final four rows of results are experiments over six sites. This was the largest number of sites for which it was practical to do runs of twenty or more simulations with the computer power available. Again, the mean corrected estimate of matches was nearly correct in all cases. The worst performance was an estimate of 92.2 (correct result 100). The σ was, however, relatively

high. This was a surprise in some cases — particularly the first row of results where the mean number of false matches was only 21.2. In many senses, the worst result was the final one where a σ of 55.0 was given on a corrected prediction of only 101.3.

The time taken to do one run over six sites with one thousand pieces of data on each site was thirty seconds on a Celeron 366 computer running Debian Linux. It is practical (if time consuming) to do experiments on seven sites, even using such comparatively obsolete equipment. However, eight sites or more is probably too computationally expensive for the moment and this is a limitation of the method outlined.

To test the method more fully, four very extreme tests were given. Each of these tests involved six sites at each of which one thousand vehicles were observed. Interacting flows were chosen to cause a large number of false matches in a diversity of ways. Because these experiments were chosen to cause a large number of false matches then one thousand runs of each experiment were performed. The averaged results are shown in Table 4.7.

Experiment Number	Expected Answer	Av.Raw Matches	σ Raw Matches	Av.Cor. Matches	σ Cor. Matches
1	0	739	305	11.9	196
2	0	110	45.5	-0.950	27.1
3	250	836	287	249	205
4	500	1920	531	496	356

TABLE 4.7. Simulation results — all performed over one thousand runs with 10,000 distinct vehicle types.

In experiment one, five hundred vehicles travelled from one to five and five hundred from two to six. The remaining five hundred vehicles at sites one and six were appeared nowhere else. No vehicles made the complete journey. As can be seen, on average over seven hundred false matches were seen and the standard deviation between runs was extremely large. However, the mean was within twelve of the correct answer (zero) although the standard deviation was

large. In such extreme circumstances, a single experiment would be next to useless but it is good evidence that the method was unbiased.

In experiment two, five hundred vehicles travelled from one to three. Five hundred vehicles travelled from four to six. Five hundred vehicles visited only odd numbered sites and five hundred vehicles visited only even numbered sites. In this experiment the corrected mean result was almost exact (within one) and the standard deviation was much lower.

In experiment three, two hundred and fifty vehicles travelled to all sites. Five hundred vehicles went from site one to three and five hundred from four to six. The remaining two hundred and fifty vehicles at each site visited only that single site. As can be seen, the corrected result is almost exactly correct although, again, the standard deviation is so high that a single reading would be worthless.

In experiment four, five hundred vehicles visited every site. Two hundred and fifty vehicles went from sites one to three. Two hundred and fifty vehicles went from sites four to six. Two hundred and fifty vehicles visited only sites one and two, two hundred and fifty vehicles visited only sites three and four and two hundred and two hundred and fifty vehicles visited only sites five and six. Again, the mean of all results is very close (within four vehicles) but the standard deviation is the highest yest seen. This is not surprising. The mean number of raw 6-tuples of matches averaged nearly 2000 — twice the number of vehicles at each site.

These four tests provide a convincing demonstration that the method is, indeed, unbiased as was shown by theory.

4.9.1. Summary of Results. The results given here are certainly consistent with the idea that the method gives an unbiased estimator for the true number of matches. In some experiments, there were problems with the standard deviation being higher than would be desirable in real cases. It is important to bear in mind that these were relatively extreme tests of the method since $p(2)$ and $p(3)$ were relatively low and the number of samples given were

quite high. Often the method was attempting to predict only ten true matches in a number of observed matches which might be several hundred. In the most extreme case given, the method was able to remove 1400 false matches and find the correct answer to within six. However, this was averaged over one thousand simulation runs. In reality only a single experiment run can be done and, if the standard deviation were so high in a real situation then the answer given would be useless. A critical need for this research is a method to assess the standard deviation.

CHAPTER 5

Statistical Analysis of Route Choice Data

This chapter describes and analyses data which were collected as part of an EPSRC funded project which was held jointly at the Universities of York and Leeds. The aim of the project was to collect and analyse data to study driver route choice. Some of the material in this chapter has previously been presented at the Universities Transport Studies Group [**31**]. More information can be found at the web site:
`http://gridlock.york.ac.uk/route/`.
Additional information and data connected with the project can also be found at this site. An initial report on data matching as performed in this chapter is given in [**28**].

Code to perform matching of licence plates based upon a Maximum Likelihood Estimator approach was written by Stephen Clark (Leeds City Council) and David Watling (Institute for Transport Studies, University of Leeds). This code was used in this chapter with their permission. Their good work and many useful discussions were extremely valuable in the work described here.

5.1. Introduction

Two large data collection exercises took place as part of this project. Both attempted to gather a data set suitable for the investigation of problems related to driver route choice. Both studies concerned capacity reducing network interventions and both centred on large licence plate surveys conducted over a number of weeks. The first study investigated the closure of Lendal Bridge, part of York's inner ring road and one of only three river crossing points in the city centre. This major capacity reduction had significant effects on the traffic in the city. However, complicating factors made the data hard to analyse.

A second study investigated the partial closure of Fishergate in one direction only. Again, this is part of York's inner ring road and again, it was anticipated that the effect on the traffic system would be significant. The collected data was analysed to determine the most used routes through the city and how drivers swapped between them. Given the difficulties inherent in the Lendal Bridge data set, the analysis here will concentrate more on the Fishergate data than on the Lendal Bridge data although some investigation will be performed on both data sets.

The data from these studies is analysed using the methods of the previous chapter in addition to techniques from the literature about licence plate matching and the standard statistical techniques of t-tests and modelling using General Linear Models (GLM). The aim of the modelling is to rigorously investigate hypotheses related to driver route choice.

Section 5.2 describes the statistical techniques which will be used in the analysis performed in this chapter. Section 5.3 describes the methodology used to carry out the surveys. Section 5.4 describes the initial data analysis and provides an overview of the data collected. Section 5.5 analyses the data captured using a simple graphical technique to show the changes in the network over time. Section 5.6 analyses the flow data in all the sites and Section 5.7 considers the data disaggregated by site. Section 5.8 matches data between pairs of sites in terms of flows and travel times. Section 5.9 considers matches between more than two sites using the methods of Chapter 4. Finally, Section 5.10 concludes the chapter with a summary of the main results of the analysis.

5.2. Statistical Techniques

A number of traditional statistics techniques are used in this chapter as well as some useful graphical visualisation techniques (for example the time plots in Section 5.5). A quick introduction to the statistical techniques is provided here. Proofs are not included here but references to proofs in a standard text are given.

5.2.1. Confidence Intervals and the t-Distribution. The discussion in this section follows [**80**, page 146].

DEFINITION 5.1. The *normal distribution* is given by the density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu)^2/\sigma^2},$$

where μ is the mean and σ^2 is the variance. A standard normal distribution has a zero mean and unit variance. In this case, the density function simplifies to

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{1}{2}x^2}.$$

DEFINITION 5.2. The *chi-square distribution* (χ^2) is given by the density function

$$f(x) = \frac{x^{(\nu/2)-1}e^{-(x/2)}}{2^{(\nu/2)}\Gamma(\nu/2)}, \qquad x > 0$$

$$f(x) = 0, \qquad x \leq 0$$

where ν is a parameter of the distribution, known as the *degrees of freedom* possessed by the distribution and Γ is the Euler gamma function defined in Section A.3.

THEOREM 5.1. The *central limit theorem* states that if X is a random variable with a mean μ and a variance σ^2 and $X_1, \ldots, X_n$ is a random sample of X, then $Z = (\overline{X} - \mu)\sqrt{n}/\sigma$ (where $\overline{X}$ is the sample mean of the first n samples), has a distribution which approaches the standard normal distribution as $n \to \infty$.

PROOF. For an outline proof see [**80**, page 383]. □

THEOREM 5.2. If the variable X is normally distributed with mean μ and variance σ^2 and $X_1, \ldots, X_n$ is a random sample of X then the random variable

$$U = \sum_{i=1}^{n} (X_i - \mu)^2/\sigma^2,$$

will possess a chi-square distribution with n degrees of freedom.

PROOF. For a proof see [**80**, page 136]. □

Since the true mean of a population is generally not known, a similar theorem for the sample mean is useful.

THEOREM 5.3. Under the conditions of Theorem 5.2, the random variable

$$V^2 = \sum_{i=1}^{n} \frac{(X_i - \overline{X})^2}{\sigma^2}, \tag{5.1}$$

has a chi-square distribution with $n - 1$ degrees of freedom.

PROOF. For a proof see [**80**, page 279]. □

This theorem is useful to calculate confidence intervals. Given a variable X then a $(1 - \alpha)$ confidence interval $[x1, x2]$ is given by two numbers x_1 and x_2 such that

$$\mathbb{P}[x_1 < X < x_2] = 1 - \alpha.$$

In this chapter, confidence intervals will usually be given as a percentage confidence. For example a 95% confidence interval is equivalent to $\alpha = 0.05$.

DEFINITION 5.3. *Students' t-distribution* is given by

$$T = \frac{Z}{V}\sqrt{\nu}, \tag{5.2}$$

where Z is a variable with a standard normal distribution and V^2 is an independent variable with a chi-square distribution which has ν degrees of freedom. A *t-statistic* is a variable which has a t-distribution.

The t-statistic is useful in a number of circumstances for estimation. For example, given a normally distributed variable X then a standard normal variable Z can be found by subtracting the mean and dividing by the standard deviation. This gives

$$Z = \frac{\overline{X} - \mu}{\sigma_{\overline{X}}} = \frac{\overline{X} - \mu}{\sigma}\sqrt{n}, \tag{5.3}$$

where n is the number of samples and the substitution in the final part of the equation is from equation (1.3) ($\text{var}\left(\overline{X}\right) = \sigma^2/n$).

It be shown [**80**, page394] that V^2 from equation (5.1) and Z from equation (5.3) are independent. Therefore, the two can be substituted in equation (5.2) to get

$$T = \frac{(\overline{X} - \mu)\sqrt{n}}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}}\sqrt{n-1},$$

which will possess a t-distribution. This can be expressed in terms of the sample variance S^2 as

$$T = \frac{\overline{X} - \mu}{S}\sqrt{n}.$$

Comparing this with (5.3) it can be seen that the difference is the use of the sample variance rather than the variance itself. As would be expected, this distribution quickly converges to a standard normal as $n \to \infty$.

The t-distribution can be used to calculate a confidence interval for the probability that the true mean μ lies within a certain range around the sample mean $\overline{X}$. For a given t-distribution and a given α then the procedure in Table 5.1 shows how to find a confidence interval for the mean.

This procedure can be simply adapted to find a confidence interval for the difference between two means μ_X and μ_Y of two independent normally distributed variables X and Y with sample sizes n_X and n_Y and the same variance σ^2. Follow the same procedure as before but with $\overline{X} - \overline{Y}$ instead of $\overline{X}$ [**80**, page 148]. The required t-distribution is given by

$$T = \frac{(\overline{X} - \overline{Y}) - (\mu_X - \mu_Y)}{\sqrt{(n_X - 1)S_X^2 + (n_Y - 1)S_Y^2}}\sqrt{\frac{n_X n_Y (n_X + n_Y - 2)}{n_X + n_Y}},$$

where T has $\nu = n_X - n_Y - 2$ degrees of freedom.

Following this, therefore, if t is chosen so that $\mathbb{P}[T > t] = \alpha/2$ then a $(1 - \alpha)$ confidence interval is given by

$$\overline{X} - \overline{Y} \pm t\sqrt{\frac{(n_X - 1)S_X^2 + (n_Y - 1)S_Y^2}{n_X n_Y (n_X + n_Y - 2)/(n_X + n_Y)}}.$$

(1) Using tables or a computer, find t_1 and t_2 such that $\mathbb{P}[T < t_1] = \alpha/2$ and $\mathbb{P}[T > t_2] = \alpha/2$. Since the t-distribution is symmetrical about the origin then $t_1 = -t_2$. Define $t = -t_1 = t_2$. If a variable has a t-distribution then there is a probability $(1 - \alpha)$ that it will fall in the range $(-t, t)$.
(2) With probability $(1 - \alpha)$ then $-t < \sqrt{n}(\overline{X} - \mu)/S < t$.
(3) Rearranging this inequality gives $\overline{X} - tS/\sqrt{n} < \mu < \overline{X} + tS/\sqrt{n}$.
(4) Therefore, a $(1 - \alpha)$ confidence interval for μ is given by $[\overline{X} - tS/\sqrt{n}, \overline{X} + tS/\sqrt{n}]$. Such an interval will usually be given in the form

$$\overline{X} \pm \frac{tS}{\sqrt{n}}.$$

TABLE 5.1. A procedure for finding confidence intervals for a mean.

This formulation is extremely useful for testing hypotheses of the form

$$H_0 : \mu_X = \mu_Y$$
$$H_1 : \mu_X \neq \mu_Y.$$

Note that this H_1 defines a *two-tailed test* — that is, the possibility that $\mu_X < \mu_Y$ or $\mu_X > \mu_Y$. Sometimes, particular problems make one of these options impossible and a one-tailed test is used.

If the two variables to be tested do not share a common variance then the test requires further modification [**105**, page 455] to produce a t-distribution. This is given by

$$T = \frac{(\overline{X} - \overline{Y}) - (\mu_X - \mu_Y)}{\sqrt{S_X^2/n_X - S_Y^2/n_Y}},$$

where ν is the degrees of freedom, where

$$\nu = \frac{(S_X^2/n_X - S_Y^2/n_Y)^2}{(S_X^2/n_X)^2/(n_X-1) + (S_Y^2/n_Y)^2/(n_Y-1)}.$$

Note that, in this case, ν is not generally an integer.

Using this method, given a value of t such that $\mathbb{P}[T > t] = \alpha/2$ then a $(1-\alpha)$ confidence interval for $\mu_X - \mu_Y$ is given by

$$\overline{X} - \overline{Y} \pm t\left(\frac{(S_X^2/n_X)^2}{n_X - 1} + \frac{(S_Y^2/n_Y)^2}{n_Y - 1}\right).$$

DEFINITION 5.4. The *p-value* for a test statistic is a measure of the confidence level with which the null hypothesis H_0 can be rejected. Let X_m be the measured value of the test statistic on the real data. Let X_t be the value of the test statistic measured on a sample of some hypothetical data set for which H_0 holds. The p-value is the probability that X_t has a value at least as contradictory to H_0 as X_m. This implies that the p-value is the confidence with which H_0 should be accepted. If the p-value is near zero then H_0 should be rejected and if it is near one then H_0 should be accepted. A result is said to be *significant* at a given level if the p-value is less than this. For example, if a result is said to be significant at the 1% level this means that the p-value is less than 0.01.

For a t-test as described then the p-value is the value of α at the borderline between accepting and rejecting H_0 — that is the smallest confidence level for which the confidence interval for the difference between the means includes zero.

5.2.2. General Linear Models. *Multiple regression analysis* attempts to explain the relationship between a modelled variable and several explanatory variables. *General Linear Models* (GLMs) are one technique for this — a fuller description can be found in [**105**] and also [**80**]. The discussion here follows that in [**105**] and proofs are omitted.

If it is believed that a variable y depends on x_1, x_1^2 and x_2 then write

$$\mathrm{E}[y] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2, \tag{5.4}$$

where the β_i are the model parameters to be estimated. (The models are known as linear models because they depend linearly on these parameters.)

Qualitative data can be represented by indicator variables. For example, to represent the situations of a road being fully closed, partially closed and working normally then *indicator variables* can be used as follows,

$$x_1 = \begin{cases} 1 & \text{if the road is fully closed} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if the road is partially closed} \\ 0 & \text{otherwise} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if the road is open} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, a potential model for the volume of traffic for this road is

$$\mathrm{E}[y] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3,$$

where β_i are the parameters to be estimated and y is the output volume of the road in question.

The model assumes a random error ε and therefore

$$y = \mathrm{E}[y] + \varepsilon.$$

The model itself can be written in a standard form as

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \varepsilon. \tag{5.5}$$

where k is the number of variables in the model. The x_i could be functions of some other x_j in the model. For example, the model in Equation (5.4) could be specified in the standard form with $n = 3$ and $x_3 = x_1^2$.

The assumptions behind the model are as follows [**105**, page 603],

(1) The mean of ε is zero.

(2) The variance of ε is independent of the values of x_i.

(3) The ε are normally distributed.

(4) The ε are independent.

Such models are fitted by minimising the sum of the squared errors (the technique of least squares). This can be done using matrix algebra although this can be computationally expensive in large data sets. The GLM modelling in this thesis is fitted using the computer package R [**122**]. The language R is a statistical programming language whihc has no connection to the R^2 statistic.

There are several measures which are used to assess the goodness of fit of a GLM. The *multiple coefficient of determination* R^2 is given by

$$R^2 = 1 - \frac{SSE}{SS_{yy}} = \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2}, \tag{5.6}$$

where SSE is the sum of the squares of the errors, SS_{yy} is the variance of y, y_i is a measured value, $\hat{y}_i$ is the estimate produced by the model for y_i and $\overline{y}$ is the sample mean for y. If $R^2 = 1$ then the model is a perfect fit with the chosen model passing exactly through every data point. If $R^2 = 0$ then the model does not explain any of the variance in the data. Of course, it should be noted that any model can be fitted with the R^2 value arbitrarily close to one by adding more parameters. For this reason, an alternative statistic, the adjusted R^2 value R_a^2 is often used.

$$R_a^2 = 1 = \frac{n-1}{n-(k+1)}(1 - R^2), \tag{5.7}$$

where n is the sample size and k is the number of parameters in the model. The R_a^2 value will always be lower than the R^2 value.

The R^2 value can be used to test statistical hypotheses. Take the model

$$\mathrm{E}[y] = \beta_0 + \beta_1 x_1 + \ldots \beta_k x_k,$$

and the two hypotheses

$$H_0 : \beta_1 = \cdots = \beta_k = 0$$

$$H_a : \text{there exists some } i > 0 \text{ such that } \beta_i \neq 0,$$

where H_0 is the null hypothesis.

The F *statistic* is given by

$$F = \frac{R^2/k}{(1-R^2)[n-(k+1)]}.$$

Under the previously listed assumptions for the GLM then the hypothesis H_0 is rejected with confidence $(1-\alpha)$ if

$$F > F_\alpha,$$

where F_α is a function of α (the required confidence level), $\nu_1 = k$ and $\nu_2 = [n-(k+1)]$ and can be looked up in tables or calculated by computer.

The *p-value* or *observed significance level* can be calculated as in the previous section. In the case of the F statistic, if the measured value from the sample is F_m the p-value is the α where $F_\alpha = F_m$. In the case of GLM it is usually desirable that the p-value is near zero since the null hypothesis is that the parameters of the model are all zero. If H_0 were accepted it would mean that the model were poorly specified. It should be noted that even if the p-value is near zero this does not mean that all the parameters of the model are necessary. It could well be that a more parsimonious model would fit the data equally well. In general, a model with fewer parameters should be preferred if one is available.

5.3. Survey Methodology

This section describes the data collection for the two surveys and the nature of the data collected. The survey methodology was informed by a pilot study on Park Row in Leeds. The pilot study data is also available for download from the web site as is a report on the pilot study [**29**]. It was decided that for cost effectiveness, the most appropriate survey type was a manual survey using a tape recorder to read licence plate data and later transcription of licence plate data. From the pilot survey, it was further decided that partial licence plates only would be recorded in order to deal with the traffic volumes that would be encountered in the main study. The surveys were also informed by modelling using the SATURN modelling package at the University of Leeds.

This software package was used to identify possible alternative routes used by drivers.

5.3.1. General Notes on Survey Methodology. While the two surveys were separately conducted, the same general methodology was employed in both. The number of sites and days surveyed was constrained by the budget available for the survey. Obviously, there was a trade-off between surveying a large number of sites and surveying a large number of days.

Survey sites were, in both cases, chosen with several considerations in mind.

(1) Both surveys were centred on a major network intervention and it was most important to concentrate on the effects of this intervention.
(2) Since the study was about route choice, it was considered important to locate the same vehicles at multiple points on a route. Ideally, vehicles would be spotted at three points: going towards the site of the intervention; at the site (or some obvious alternative site); and going away from the site.
(3) There was a trade-off between putting survey sites near to the intervention site in order to be sure to record relevant vehicles and putting survey sites further away and potentially gaining more information about travel behaviour (but risking getting a smaller number of matches).

Survey days were chosen with a number of considerations in mind:

(1) It was considered important to get a good estimate of the ambient variability between days in the city and also the trends in traffic patterns between weeks.
(2) It was considered important to monitor the transient response in the days immediately following the closure.
(3) It was thought of as desirable to get some estimate of the longer-term response to the closure (whether the traffic had time to form a new equilibrium and how long this took to establish).

(4) It was not considered appropriate to monitor weekends. While it was recognised that traffic differs considerably between weekdays, it was considered that this variation was extremely minor compared with the variation in travel patterns between a typical weekday and a Saturday or a Sunday.

The surveys themselves took place during the morning rush hour. This was chosen since the morning rush hour in York is more congested than the evening rush hour and is also held to be generally more consistent between days. In the Lendal study, the sites were monitored from 8:00 to 9:00. In the Fishergate study, the traffic was monitored at most sites from 7:45 to 9:15. This was in order to catch all of the rush hour traffic and a quarter of an hour window either side. However, at selected sites, this window was adjusted to monitor from 8:00 to 9:30. This happened at those sites that would be reached last on a journey (for example, in Figure 5.2 site J would always be reached after site A). This was decided since the travel time between some pairs of sites was of the order of half an hour. Without such an offset some of the survey time would otherwise be wasted since the earliest (or latest) parts of the data could not be expected to match with data at any other site. The sites which were surveyed from 8:00 to 9:30 were sites A, I and J.

Timing on the surveys was performed by asking the surveyors to record the time at approximately five minute intervals. Surveyors were supplied with synchronised watches at the beginning of the surveys. The times for data between each time stamp are interpolated so, for example, if there are ten plates between a time stamp at 8:10 and one at 8:19 they will be split so that one plate is seen in each minute. Because of this interpolation and possible rounding of the time, the times recorded can only be assumed to be accurate to within five minutes, however, it is hoped that it is accurate to a much greater resolution than this.

5.3.2. Lendal Bridge Study Methodology. The first study took place between June and October 2000. The aim of the study was to monitor the

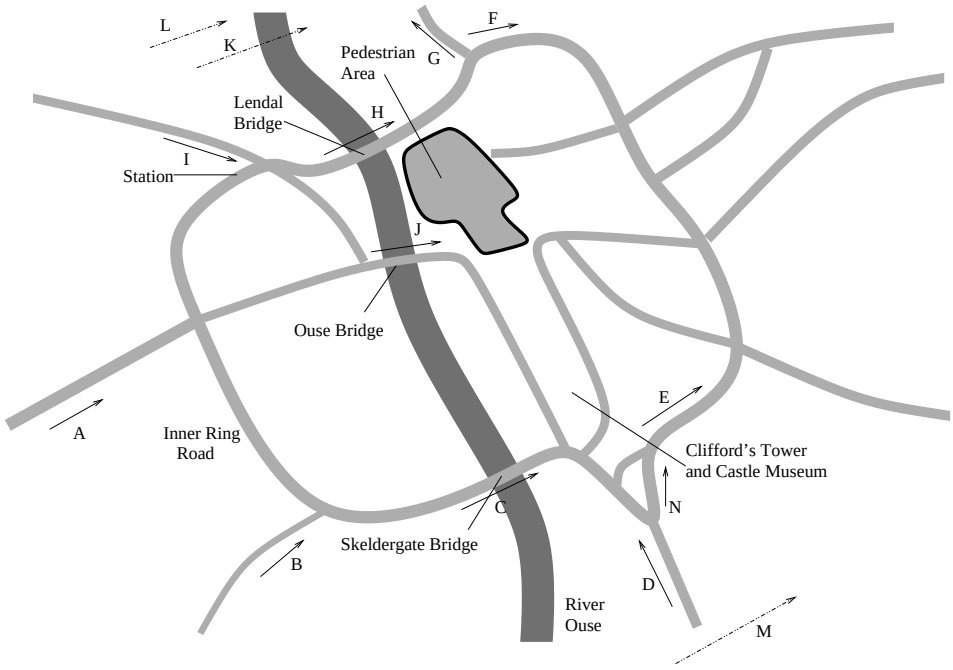

FIGURE 5.1. The Lendal Bridge study survey sites. Sites K, L and M are off the map given.

Key	Street Name	Notes
A	Blossom Street	
B	Bishopthorpe Road	
C	Skeldergate Bridge	
D	Fishergate Road	
E	Paragon Street	
F	Gillygate	
G	Bootham	
H	Lendal Bridge	Site of Closure. Not surveyed when closed.
I	Leeman Road	
J	Ouse Bridge	Potential river crossing.
K	Clifton Bridge	Potential river crossing. Actual location is off map.
L	A1237	Outer ring road. Potential river crossing. Actual location is off map.
M	A64	Outer ring road. Potential river crossing. Actual location is off map.
N	Barbican Road	Only surveyed when bridge is closed.

TABLE 5.2. A list of survey sites in the Lendal Bridge survey.

closure of Lendal Bridge, part of York's inner ring road, for scheduled maintenance. A diagram of the area of the study is shown in Figure 5.1. The sites on the figure are described in Table 5.2. The closure took place on the 11th of September 2000 and was planned to last for over a month. The bridge was closed to all vehicles but left open for public transport and cycles.

As can be seen from the diagram, not only is the bridge part of the inner ring road but it is also one of only three river crossings available to vehicles in the centre of York (three other bridge crossings are also available further out from the city centre, two of them on the outer ring road). In order to maximise the number of vehicles observed at several points on their route, it was decided to attempt to capture those vehicles making a journey from the south or west part of the city to the north or east. These vehicles obviously have to cross a bridge somewhere. The sites were chosen with the idea of capturing vehicles at three points on their journey: once on the southwest side of the river; once as they cross; and once on the northeast side of the river. Survey days were chosen with regard to the considerations mentioned above. In addition, an "early" before study was taken for two days a few months before the main survey. This was to allow researchers to react and change any problems with the survey methods used.

Table 5.3 shows the days that were surveyed for the Lendal Bridge study. Unfortunately, the day that the bridge closed coincided with the beginning of the UK Fuel Crisis. Because of a blockade of petrol depots by a group protesting against high fuel tax, petrol supplies in the UK were extremely limited for the entire week. This had a major effect on the flows throughout the city for the first week of closure. Surveys were rescheduled to try and avoid these effects and to concentrate on monitoring the bridge reopening. The bridge reopening coincided with the flooding of the city of York in October 2000. The Lendal Bridge study was abandoned at this point. The analysis of the fuel crisis data is of interest in itself and a report on this is available for download from the project web site [**30**]. Note that, because the bridge was

Day	Survey	Comment
27th June 2000	Before survey (not site N)	Early before survey to establish the repeatability of traffic patterns and the expected change between months.
28th June 2000	Before survey (not site N)	Early before survey to establish the repeatability of traffic patterns and the expected change between months.
6th Sept 2000	Before survey (not site N)	
7th Sept 2000	Before survey (not site N)	
8th Sept 2000	Before survey (not site N)	Final weekday before closure.
11th Sept 2000	During survey (not site H)	First day after closure. This also marks the first day of the UK fuel crisis.
13th Sept 2000	During survey (not site H)	Third day after closure. This was the third day of the UK fuel crisis (and marked a low point for flows).
27th Sept 2000	During survey (not site H)	
18th Oct 2000	During Survey (not site H)	Last day before planned reopening.

TABLE 5.3. The Lendal Bridge survey summary.

fully closed, this freed up an extra surveyor who was used at site N during the closure. This was identified as a possible rerouting for the closure using SATURN modelling.

5.3.3. Fishergate Study Methodology. The Fishergate Study is shown in diagrammatic form in Figure 5.2. Information about the survey sites is shown in Table 5.4. This survey was based around works to repair a collapsed sewer at site A. The repair site was slightly after the survey site (approximately ten metres further down the road) and no turn offs were available between site A and the site of the closure. The repair work involved a partial closure, essentially one lane being removed from the road. The closure was originally

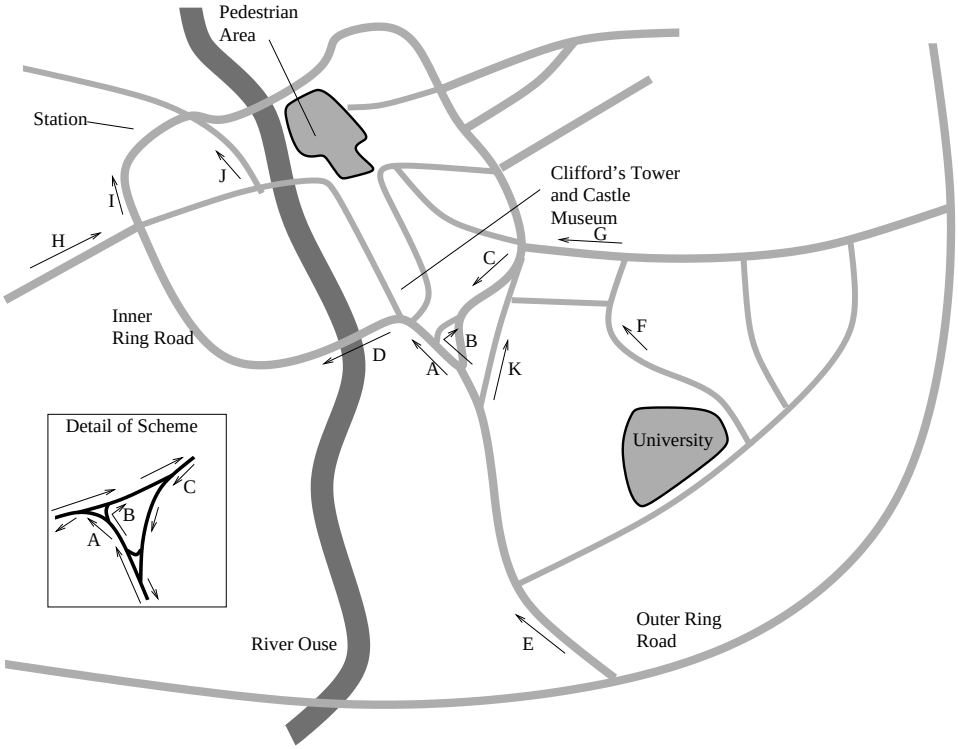

FIGURE 5.2. The Fishergate study survey sites.

Key	Street Name	Notes
A	Fishergate	Left turners only (one of the two lanes at this site was closed).
B	Fishergate	Right turners only (very low flow).
C	Paragon Street	
D	Skeldergate Bridge	
E	Fulford Road	
F	University Road	
G	Lawrence Street	
H	Blossom Street	
I	Queen Street	
J	Rougier Street	
K	Cemetery Road	

TABLE 5.4. A list of survey sites in Fishergate survey.

scheduled to last only two weeks and therefore the plan was to survey for one week before, one week during and one week after the closure. However, the closure was extended to four weeks and therefore no true after survey data

Day	Survey	Comment
25th June 2001	Before survey	
26th June 2001	Before survey	
27th June 2001	Before survey	
28th June 2001	Before survey	
29th June 2001	Before survey	
2nd July 2001	Before survey	Partial closure occurred at 9:15 and should not affect the data collected.
3rd July 2001	During survey	First day of partial closure.
4th July 2001	During survey	
5th July 2001	During survey	
6th July 2001	During survey	
11th July 2001	During survey	
12th July 2001	During survey	
13th July 2001	After survey	Road works removed to ease traffic for a race meeting in York. This can be considered to be an after survey day.
16th July 2001	During Survey	Road works put back in place.

TABLE 5.5. The Fishergate survey summary.

is available with the possible exception of the 13th of July when the closure was temporarily suspended to allow for the increase in traffic due to a major horse-racing event that weekend. The extra traffic due to the race-goers is thought not to have had a great effect on traffic during the morning peak.

5.3.4. Hypotheses Tested. Various hypotheses are tested on the data in this chapter. The main hypotheses are discussed here to provide a framework for the formal analysis later in the chapter. The hypotheses are investigated both informally (using graphical techniques) and formally using statistical models.

Several hypotheses relate to flow levels at individual sites. The most obvious hypothesis is that on average sites will either increase or decrease in flow as a result of an intervention in the network. Flow data is examined graphically and using a statistical model in Section 5.6. Following this, the

hypothesis is made the flows will be affected differently according to site depending on whether a survey site is a potential rerouting (where flow might be expected to increase) or is on the route affected (where flow might be expected to decrease). This is investigated in Section 5.7.

The next group of hypotheses relate to flows and travel times between site pairs. Section 5.8 investigates the hypothesis that at each site, travel time and flow is affected by the intervention on the network and the level of the effect produced changes with the number of days since the intervention. This is essentially, a trial of the hypothesis that the network is in a transitory state when the intervention first occurs and the effects of the intervention will lessen as time goes on. This could be thought of as the system finding a new equilibrium, to use the terms of Chapter 3.

The hypothesis relates to how vehicles change their behaviour as day follows day. Section 5.8.3 investigates the factors affecting a driver's decision to travel on a subsequent day and hypothesises that this depends on whether the days in question are on same day of the week, whether the days in question are in different weeks and how far apart in time the days in question are. Finally, in Section 5.9 the hypothesis that individual drivers in the data can be seen changing from one route to another as a result of the intervention is investigated.

5.4. Initial Data Analysis

A number of issues are worth mentioning related to analysis of the data. Firstly there is the problem of false matches that was extensively discussed in the previous chapter. Secondly there is the problem of missed journey ends. This is the problem of those vehicles that are seen at one site but are not seen at a second site because the survey at the second site ended before the vehicles were seen. It is difficult to avoid this problem totally but it can be mitigated by removing the end part of the data at the first site to such an extent that all vehicles seen at that site would complete their journey. Thirdly there is

the problem of errors in data recording. Great efforts were made to avoid the most common sources of error. Preliminary surveys found that a primary source of errors was the mistaken transcribing of letters that sound alike (N and M for example). This was reduced by encouraging surveyors to use a phonetic alphabet and also by minimising the reliance on letters by primarily recording digits.

Obviously, because the recordings were made by fallible human surveyors, not all vehicle plates were correctly recorded. Plate recordings which were recorded with question marks where the surveyor missed a vehicle are not matched. Similarly, any plate which had less than three digits or letters recorded was not matched since these shortened plates would be more susceptible to false matching. The flows presented in this section are also presented as adjusted flows with these poorly recorded plates removed. Tables 5.6 and 5.7 show the flows at the sites in the Lendal Bridge survey with the adjusted flows also recorded. Tables 5.8, 5.9 and 5.10 show the Fishergate flows and adjusted flows. Note that surveys which are marked with a † or ⋆ (a whole lane or other large amount of missing data) will be omitted from subsequent analysis but surveys marked with a ‡ (a small amount of missing data) will be included. The latter category only includes a small number of days in the Lendal Bridge survey.

5.5. Time Plots

Perhaps the simplest method for visualising matches between two sites is using graphical techniques as described in [**158**] — while the techniques used in this section do not provide quantitative results, they are extremely effective for demonstrating the nature of matches and, indeed, provide more of an insight into the data than some more sophisticated techniques. When the same plate is found at two sites, plot a point on a graph with the x value as the time at site one and the y value as the time at site two. The plots are collected together in Appendix C.

Site	27/6/2000 Flow		28/6/2000 Flow		6/9/2000 Flow		7/9/2000 Flow		8/9/2000 Flow	
	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.
A	954	934	942	933	830	810	728	712	699	683
B	390	378	—	—	410	407	441	434	487	473
C	906	884	876	857	—	—	837 ‡	813 ‡	718 ‡	695 ‡
D	1490	1412	1451	1395	1312 ‡	1281 ‡	1399	1359	1430	1384
E	572 †	560 †	631 †	621 †	1259	1234	1206	1181	1238	1209
F	413	399	399	390	434	419	429	420	438	424
G	419	413	381	374	364	345	395	381	341	323
H	519	502	458	444	472	459	488	474	446	431
I	344 ‡	331 ‡	449	437	491	481	479	473	533	522
J	462	453	526	517	567	558	592	579	569	553
K	864	845	871	849	879	865	861	840	860	841
L	1006	978	1068	1041	840	817	836	814	822	794
M	726 †	717 †	1729	1710	2079	2029	2040	2002	1994	1957

TABLE 5.6. Lendal survey before flows. † indicates data only available in one lane for this survey. ‡ indicates small amounts of missing data in this survey.

Site	11/9/2000 Flow		13/9/2000 Flow		27/9/2000 Flow		18/10/2000 Flow	
	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.
A	710	693	659	652	633	620	185 ⋆	182 ⋆
B	394	386	446	440	502	490	475	464
C	906 ‡	885 ‡	846	835	927	901	—	—
D	1494	1452	1222	1199	1465	1427	1438	1403
E	1295	1264	1175	1158	1323	1295	1080 ‡	1060 ‡
F	290	279	278	273	298	283	351	334
G	399	382	345	341	393	376	432	415
H	—	—	—	—	—	—	—	—
I	461	446	361	355	416	408	420	412
J	621	602	508	497	596	584	557	541
K	873	850	761	750	799	782	863	839
L	851	838	755	738	855	834	868	838
M	1999	1954	—	—	—	—	2145	2099
N	488	471	—	—	417	404	443	430

TABLE 5.7. Lendal survey during flows. ⋆ indicates only partial data available on this survey. ‡ indicates small amounts of missing data in this survey.

Figure C.1 shows two data sets which should be uncorrelated — the sites chosen are on opposite sides of the city (sites L and M on Figure 5.1) and the data is from the same day. A vehicle could not easily drive between these sites at rush hour in the survey time and there are no reasonable routes

Site	25/6/2001 Flow		26/6/2001 Flow		27/6/2001 Flow		28/6/2001 Flow		29/6/2001 Flow	
	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.
A	2040	1994	1953	1916	1970	1936	2073	2039	2007	1979
B	—	—	87	87	103	103	82	81	92	92
C	996	972	991	969	995	966	1008	980	1014	987
D	1187	1169	1078	1052	1182	1146	1265	1235	858 ⋆	845 ⋆
E	1421	1372	1387	1349	1306	1272	1469	1429	1448	1410
F	502	494	547	533	543	522	587	560	575	561
G	1148	1127	1135	1116	1116	1095	1102	1078	1120	1098
H	903	894	753 ⋆	745 ⋆	878	863	856	843	920	901
I	810	793	877	856	909	889	902	881	855	837
J	498	477	509	495	473 ⋆	451 ⋆	491	466	447	430
K	546	532	565	553	508	498	589	575	646	630

TABLE 5.8. Fishergate survey week one. ⋆ indicates only partial data on this day.

Site	2/7/2001 Flow		3/7/2001 Flow		4/7/2001 Flow		5/7/2001 Flow		6/7/2001 Flow	
	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.
A	2022	1993	1727	1673	1752	1696	1691	1637	1734	1667
B	100	100	93	91	79	73	76	73	95	89
C	1032	1003	898	873	972	946	903	877	959	926
D	1200	1172	1054	1039	1064	1042	1024	1003	1065	1045
E	1527	1488	1448	1413	1494	1445	1522	1485	1518	1465
F	583	564	595	572	619	591	573	552	589	568
G	1050	1032	1042	1021	1163	1138	1214	1186	—	—
H	868	854	963	951	924	908	939	922	906	895
I	842	828	859	834	885	861	882	862	858	843
J	492	475	438	422	444	423	428	408	484	461
K	606	589	691	676	693	677	750	733	749	724

TABLE 5.9. Fishergate survey week two.

including both sites. Any matches on this graph should be false matches and any perceived shape in the picture is due to the distribution of rush hour traffic or due to coincidence rather than being due to genuine matches. This picture of a set of false matches should be borne in mind when considering subsequent pictures which represent true matches corrupted by false matches.

By contrast, in Figures C.2 and C.3 some correlations in the data should be expected. Figure C.2 shows the matches between two days of plate data on the A64 at rush hour (site M on Figure 5.1). The diagonal represents drivers who are travelling at approximately the same time of day on both occasions.

Site	11/7/2001 Flow		12/7/2001 Flow		13/7/2001 Flow		16/7/2001 Flow	
	Raw	Adj.	Raw	Adj.	Raw	Adj.	Raw	Adj.
A	1675	1639	1649	1606	1910	1889	1779	1732
B	84	81	69	68	98	98	65	62
C	827	806	856	835	918	890	979	961
D	509 ⋆	503 ⋆	946	934	1133	1116	1108	1096
E	1481	1439	1512	1468	1603	1553	1578	1538
F	557	545	548	527	643	624	603	582
G	1165	1136	1109	1084	1300	1270	1226	1210
H	948	929	862	848	843	829	1355	1307
I	898	872	881	870	837	825	886	871
J	494	476	506	488	477	451	460	443
K	675 ⋆	659 ⋆	723	708	680	665	727	716

TABLE 5.10. Fishergate survey final weeks. ⋆ indicates only partial data on this day.

It can be seen with reference to Tables 5.6 and 5.7 that Figure C.2 should have a slightly higher number of false matches (due to the higher number of vehicles in the pair of files). However, it should also be clear that there is a significant number of genuine matches between the two days. In Figure C.3 matches are made at the same site but on two days which are four months apart. As can be seen, the correlation in the data is much lower. In both cases there appears to be some effect from drivers travelling at the same time of day each day but this effect is lessened as the days surveyed are further apart.

Figures C.4, C.5 and C.6 show journeys between Leeman Road and Ouse Bridge (sites I and J on Figure 5.1). These sites were considered by surveyors to be a common route through the city and should show a significant correlation. The absolute level of traffic is lower than in the previous figures. As can be seen in Figure C.4 there is a strong offset diagonal representing the vehicles which moved directly between the two sites. Other points represent either false matches or vehicles which (for whatever reason) travelled between the sites taking an unusual time. This figure can be made clearer by plotting the time seen at site one versus the travel time difference between the two. This type of plot is shown in Figure C.5 and with the y axis magnified in Figure C.6. These can be thought of as being like the previous figures but with a simple

transform. The false matches in this type of plot form a parallelogram since vehicles observed early at site one are more likely to falsely match with vehicles seen after them at site two and the converse is true for vehicles observed in the later parts of the data at site one (false matches will tend to show a negative travel time to get to site two). From Figure C.6 it seems that the typical vehicle takes three to five minutes to get between the sites.

Figures C.7 and C.8 show the travel times from these sites on the last day before the Lendal bridge closure and the first day after. It is hard to say whether a significant increase in travel time has taken place between these sites as a result of the Lendal bridge closure. As mentioned, the fuel crisis was a complicating factor. One possible explanation for Figure C.8 is that initial early congestion was relieved by commuters deciding not to travel as the news of the fuel crisis emerged that morning. However, it should be stressed that this is extremely speculative given the limited nature of the evidence.

Figures C.9 to C.14 show the main inbound route from Fulford Road to Fishergate (sites E–A on Figure 5.2) on the before survey days. From these plots, it can be seen that the typical pattern of traffic between the sites is a travel time of five minutes, rising to around twelve minutes in the rush hour and then declining back to five minutes.

Figures C.15 to C.18 show a selection of plots from the surveys taken during the closure. On the first day of the closure (3/7/01) it can be seen that the travel time begins at five minutes but continues to rise throughout the rush hour and at the end of the rush hour is up to approximately twenty five minutes (but may be beginning to decline). In subsequent days, a similar but less dramatic rise occurs. These plots on their own may be considered as evidence that the first day impact of congestion is much stronger. For whatever reason, it appears that on subsequent days the impact of the change is mitigated (perhaps by some driver behavioural mechanism).

5.6. Analysis of Flow Data

In this section, flow data is examined using standard statistical modelling techniques. Tables 5.11 and 5.12 summarise the flow data. In the case of the Lendal Bridge data the data is split according to whether the data is before, during the fuel crisis (and bridge closure) or during the bridge closure. In the case of the Fishergate date, the data is split according to whether the day is a before day, a during day or an after day (the 13/7/01 is considered an after day since the closure was not in place although it was reinstated on 16/7/01).

Site	All Days		Before		Fuel Crisis		During	
	Mean	σ^2	Mean	σ^2	Mean	σ^2	Mean	σ^2
A	769.4	15504.6	830.6	13870.8	684.5	1300.5	633	—
B	443.1	1821.8	432	1784.7	420	1352	488.5	364.5
C	859.4	4977.3	834.2	6804.2	876	1800	927	—
D	1411.2	8050.2	1416.4	4500.3	1358	36992	1451.5	364.5
E	1225.1	6619.1	1234.3	712.3	1235	7200	1201.5	29524.5
F	370	4412.5	422.6	264.3	284	72	324.5	1404.5
G	385.4	964.5	380	881	372	1458	412.5	760.5
H	476.6	807.8	476.6	807.8	—	—	—	—
I	439.3	3715.2	459.2	5056.2	411	5000	418	8
J	555.3	2431	543.2	2625.7	564.5	6384.5	576.5	760.5
K	847.9	1609.9	867	63.5	817	6272	831	2048
L	877.9	9414.4	914.4	13050.8	803	4608	861.5	84.5
M	1997.7	20462.3	1960.5	25025.7	1999	—	2145	—
N	449.3	1290.3	—	—	488	—	430	338

TABLE 5.11. Lendal survey flow data.

The analysis in this section is performed on the flow data shown in Tables 5.6, 5.7, 5.8, 5.9 and 5.10. The flows used are the total flows (rather than the adjusted flows) each representing ninety minutes of data in the case of the Fishergate surveys and one hour of data in the case of the Lendal surveys. All survey days which are marked as partial in the tables are completely omitted. In addition, the data from the Fishergate survey site H (Blossom Street) on 16/6/01 was omitted — this value was much larger than usual for that survey site. If the observations here were correct then it seems clear that some external effect caused a large increase in traffic at that site on that day. This site was

Site	All Days		Before		During		After
	Mean	σ^2	Mean	σ^2	Mean	σ^2	Mean
A	1855.9	23671.4	1970.3	13153.2	1735.2	8769.4	1779
B	86.4	141.1	92.8	76.7	80.1	128.8	98
C	953.4	3934.6	1006	238	913.4	3487	918
D	1108.8	7849.8	1182.4	4513.3	1043.5	3007.9	1133
E	1479.6	5841.8	1426.3	5695.9	1507.6	1622.6	1603
F	576	1256	556.2	1043.4	583.4	646	643
G	1145.4	5035.3	1111.8	1169.8	1153.2	4710.2	1300
H	900.8	1540.3	885	682	923.7	1297.9	843
I	870.1	775.9	865.8	1419.8	878.4	216.3	837
J	474.5	765.1	487.4	561.3	464.9	907.8	477
K	651.8	6610.5	576.7	2322.3	722.2	668.2	680

TABLE 5.12. Fishergate survey flow data.

omitted from analysis since, whatever this effect was, it is extremely unlikely to be caused by the closure at site A.

One clear observation about the flow is that it is a function of the site at which it was observed. One starting point for a GLM of flow would be a GLM with an explanatory variable for the particular site at which the observation was made. In fact, there would have to be one less variable than the number of sites since the remaining variable is the intercept β_0. If there are n sites then this model can be specified as

$$\mathrm{E}[f] = \beta_0 + \sum_{i=1}^{n-1} \beta_i I_i,$$

where f is the flow at a given site, I_i is an indicator variable which is one if the site of an observation is site i and zero otherwise.

For the Lendal Bridge survey, fitting this model in R gives the results shown in the table below.

Parameter	β_0	β_A	β_B	β_C	β_D	β_E	β_F
Estimate	1997	-1228	-1554	-1138	-586	-772	-1627
Parameter	β_G	β_H	β_I	β_J	β_K	β_L	β_N
Estimate	-1612	-1521	-1558	-1442	-1149	-1119	-1548
Statistic		R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate		0.9737	0.97	265.1	13	93	$< 2.2 \times 10^{-16}$

The β values are labelled according to the site for which they are an estimator. As can be seen, the R^2 and R_a^2 values for the model are high indicating that the parameters explain almost all the variance in the model. The extremely low p-value shows that H_0 can be rejected with a very high probability. All of the β values are significant at the 0.1% level. There is no parameter for site M and the flows at this site are represented by the β_0.

For the Fishergate survey, fitting the model gives the results shown in the table below.

Parameter	β_0	β_A	β_B	β_C	β_D	β_E
Estimate	900	955	-814	52	208	578
Parameter	β_F	β_G	β_I	β_J	β_K	
Estimate	-324	244	-30	-426	-249	
Statistic	R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate	0.9781	0.9765	603.9	10	135	$< 2.2 \times 10^{-16}$

The β values are labelled according to site as before. Again, the R^2 and R_a^2 values are high and the p-value is extremely low. In this case, the flows at site H are represented by the β_0 parameter. All of the β_i are significant at the 0.1% level apart from β_C (significant at the 10% level) and β_I (not significant).

It might be assumed that this is an extremely good model since almost without exception, the parameters are high when desired and low when desired. However, the model is practically useless. In fact, the only information to be found in this model is the mean flow at each site. The intercept β_0 is the mean flow at site H in the Fishergate model and at site M in the Lendal Bridge model. For the other sites, $\beta_0 + \beta_i$ is the mean flow at site i. (In the Fishergate model, the parameters which are not significant are those representing sites which have a mean close to the mean of site H.) The main point to be gleaned from this model is the obvious one that the main source of variance in observed flows (indeed, considering the F values, almost the only source) is the site at which the flow was observed. Therefore, any GLM which to explain flow must somehow avoid this problem. Two solutions suggest themselves, either

consider only the flows from a single site or work with the proportional flow at that site. That is, for each observation use

$$p_{i,s} = \frac{f_{i,s}}{\overline{f_s}}, \tag{5.8}$$

where $f_{i,s}$ is the observed flow on day i at side s and $\overline{f_s}$ is the mean flow from the observations at site s over all survey days.

To illustrate the idea of working with flows as a percentage of mean, take the hypothesis that flows were reduced during the fuel crisis. It would be surprising if this were not the case and inspection of the raw data on flows seems to confirm this. Take X_n as the sample flows on normal days (no fuel crisis) and X_c as the sample flows on fuel crisis days (11/9/2000 and 13/9/2000 were the only days of the crisis). A t-test can be performed with the hypotheses:

$$H_0 : \overline{X_n} = \overline{X_c}$$
$$H_1 : \overline{X_n} \neq \overline{X_c}.$$

Performing a t-test as described in Section 5.2.1 (without making the assumption that $\sigma_{X_n} = \sigma_{X_c}$) gives the statistics listed in the table below.

Statistic	X_n	X_c	t	ν	p-value
Estimate	823.9	767.3	0.541	38.01	0.591

A 95% confidence interval is given by $\mu_n - \mu_c = 56.5 \pm 211.3$. The most important things to note here are the p-value near 0.5 and the wide range of the 95% confidence interval. The model is insufficient to distinguish between H_0 and H_1 with confidence. This is certainly unsatisfactory since it is almost certain that the flows observed were reduced by the fuel crisis but this cannot be distinguished by this statistical model. The reason for the problem is obvious. The majority of the variance is a result of the site at which the observation was made (as seen in the previous model) rather than a result of the fuel or lack of fuel on a particular day. Worse than this, the model could

be extremely misleading due to missing data at a particular site, particularly if that site had extremely high or low flow.

Repeating this model using the proportional flow defined by equation (5.8) gives more satisfactory results. The range $\mu_n - \mu_c = 0.069 \pm 0.047$ is a 95% confidence interval. This indicates a reduction in flow over the fuel crisis of between 2.2% and 11.6%. The fact that this confidence interval is entirely positive indicates that (if the conditions of the model are met) then, with 95% confidence $\mu_n > \mu_c$ — the flows were reduced during the fuel crisis. The other parameters of the model are shown in the table below. As can be seen, the low p-value indicates that the hypothesis that the means are equal should be rejected with high confidence (in fact there was a reduction in flow at the 99% confidence level).

Statistic	X_n	X_c	t	ν	p-value
Estimate	1.016	0.947	2.96	36.289	0.00543

As an aside, if just the single worst day of the fuel crisis is compared with just the before data for the two survey days immediately before the fuel crisis then a flow reduction of between 6.8% and 17.7% is indicated with 95% confidence (the p-value of the t-test is 0.00018).

It is tempting to extend this model by asking if there is a measurable difference in the proportional flow caused by the first two survey days (which were well in advance of the other survey days) or if there was a measurable difference in the proportional flow on the two final survey days where the bridge was closed but there was no fuel crisis. One way to represent this is a GLM specified as follows

$$\mathrm{E}\,[p] = \beta_0 + \beta_1 I_b + \beta_2 I_f + \beta_3 I_c, \tag{5.9}$$

where p is the proportional flow on the route as defined by (5.8), β_i are the parameters of the model and I_b, I_f and I_c are indicator variables representing, respectively, "long before" survey days (the first two survey days), fuel crisis days and days where the bridge closure was in force but there was no fuel crisis. The GLM produces the parameters shown below.

Parameter	β_0		β_1	β_2	β_3	
Estimate	1.014		0.016	-0.062	-0.018	
Std. Error	0.016		0.026	0.025	0.026	
Significance	0.1%		low	5%	low	
Statistic	R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate	0.079	0.052	2.96	3	103	0.036

As can be seen, the p-value means that the null hypothesis can be rejected at the 5% significance level. However, this is not a surprise since the null hypothesis was that all the parameters of the model apart from β_0, the intercept, were zero. It has already been shown by a t-test that $\beta_2 \neq 0$. None of the other parameters were statistically significant and it would seem that this model is not well specified and the parameters chosen (with the exception of β_1) do not really represent the flows in the model. One important thing to notice is the low R^2 and R_a^2 parameters which indicate that this model does not capture the majority of the variance in the flow data.

Separate t-tests confirm that there is no significant difference between the proportional flows in the "long before" days and the before and also there is no significant difference between the proportional flows in the before and in the days where the bridge was closed but there was no significant changes to the flows in the network — or rather that that change was not in one consistent direction.

In the Fishergate data, a t-test can be performed to assess the effect of the bridge closure on flows. If X_o is the series of observations of the proportional flow when Fishergate was open and X_c is the series of observations when it was closed then a model of this type can be formed. Performing a t-test on this model gives a 90% confidence range of $\mu_o - \mu_c = 0.0042 \pm 0.0083$. Note that this confidence interval includes zero and therefore includes the possibility that $\mu_o < \mu_c$ and $\mu_c < \mu_o$. In other words, the model cannot say whether the proportional flows before or after are higher. The other statistics produced by the t-test are shown below.

Statistic	X_c	X_o	t	ν	p-value
Estimate	1.0060	0.9976	0.48	36.19	0.63

The middling p-value indicates that the model cannot distinguish between H_0 and H_1 with confidence. The means may or may not differ and this cannot be decided using this model on the data. This should not be a surprise because the effect of the closure on the flow is not so clear cut — while it would, obviously, be expected that the flow would be reduced at particular sites, it might also be expected that the flow would increase at diversion sites and the claim might be made (if the demand on the network were considered inelastic) that the total flow on the network as a whole would remain unaltered.

5.7. Flow Models Disaggregated by Site

Figures 5.3, 5.4 and 5.5 show the raw flow figures on the Lendal Bridge sites at each day. These flows are the flows for a single hour with incomplete surveys omitted. It should be noted that the x axis is not to any scale — days one and two are within a day of each other. Day three is more than two months later. The main thing to be noticed is that there seems to be no real pattern to be found. Days six and seven were the fuel crisis days but there does not seem to be a great reduction in flows on those days. Day seven shows a reduction on the majority of sites (this was widely held to be the most significant day of the fuel crisis). The bridge closure effects are even harder to see. Site F shows a considerable reduction on all days of closure. This might well be expected given that it is directly after site H (the closed bridge). However, the same thing could be said for site G which does not show a similar reduction. Site I also seems to show a reduction during all the closure days which might be expected as many drivers entering the network on site I might be normally continue on through to site H which was closed. This might also be said for site A although site A seems to show a consistent reduction throughout the surveyed period.

The Fishergate surveys appear to give much clearer results than the Lendal surveys when considering the raw flow data. The flows on these surveys are shown in Figures 5.6, 5.7 and 5.8. The graphs indicate the partial closure occurring on day six. While this was the case, the closure should only affect days from day seven onward. A noticeable drop in flow at site A occurs on days seven through fourteen with the exception of day thirteen (where the closure has been temporarily removed). This pattern is seen to a lesser extent in site D. Both A and D are the ones which would be most affected by the closure. Of the other sites, no particular reduction is seen but site G (identified as a potential rerouting) appears to have an increase in flow as does the obvious rerouting K. Site C feeds into site A and seems to show a slight reduction in the surveyed period. Site B does not appear to show a significant increase or decrease but this site has extremely low traffic.

In the previous section, it was shown that tests could not find a statistically significant change in the flow data except in the case of the fuel crisis days for the Lendal Bridge survey. From the graphs previously referred to, it can be seen that this seems to be not because there was no significant change but rather because such a change was observed in different directions at different sites.

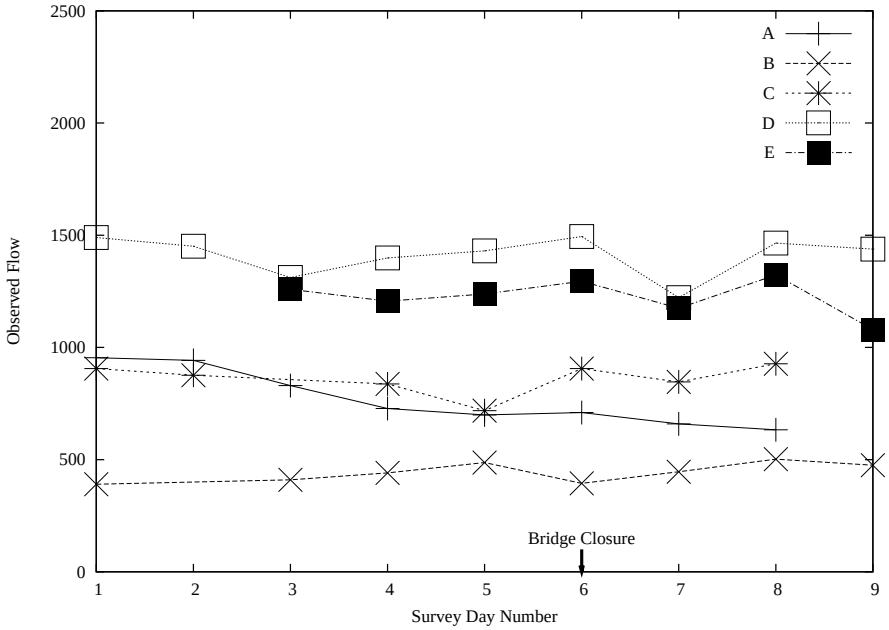

FIGURE 5.3. Lendal Bridge survey flows on sites A–E.

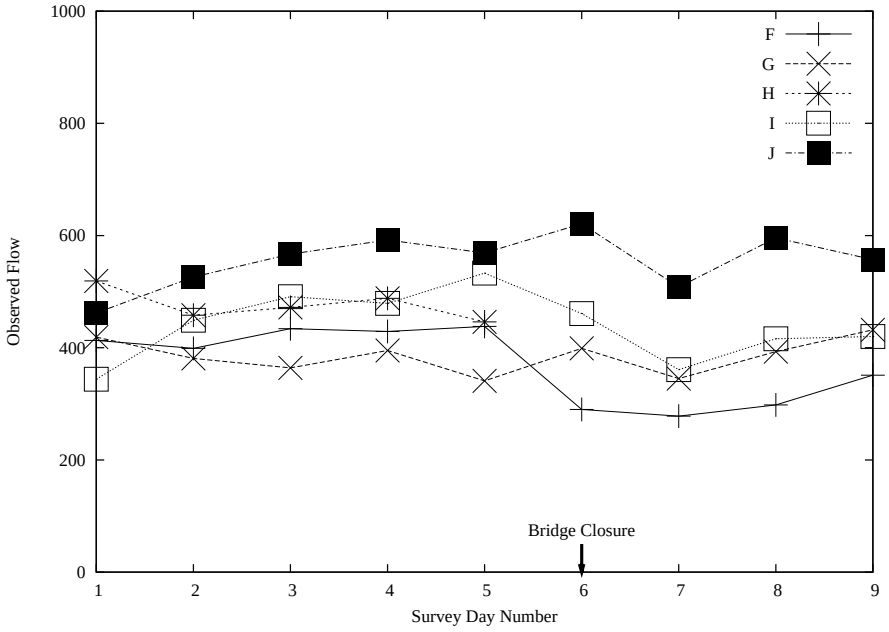

FIGURE 5.4. Lendal Bridge survey flows on sites F–J.

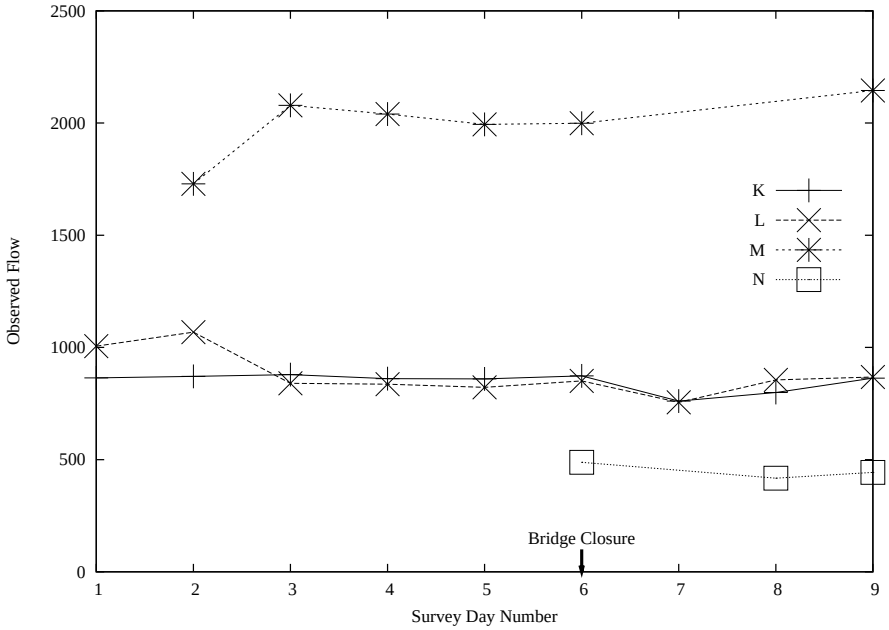

FIGURE 5.5. Lendal Bridge survey flows on sites K–N.

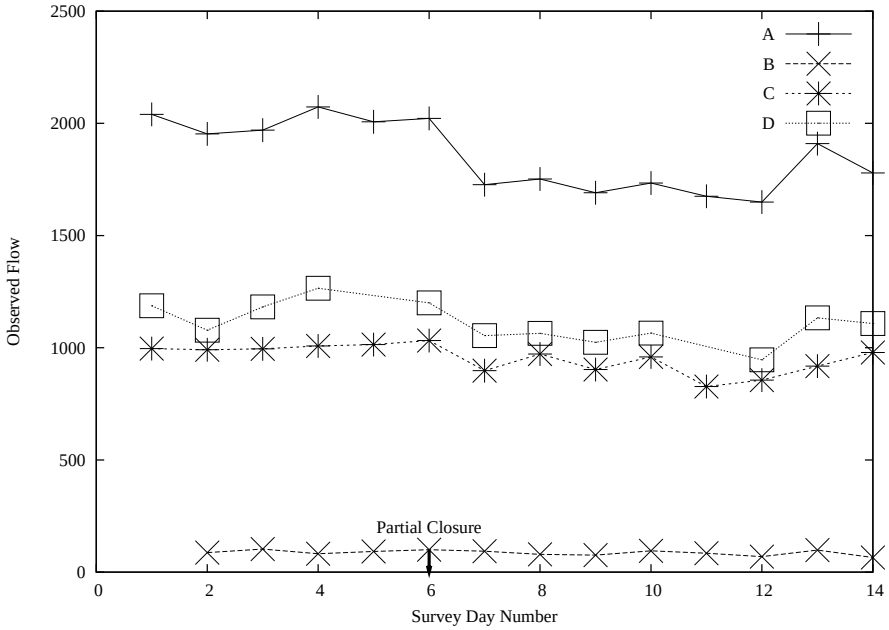

FIGURE 5.6. Fishergate survey flows on sites A–D.

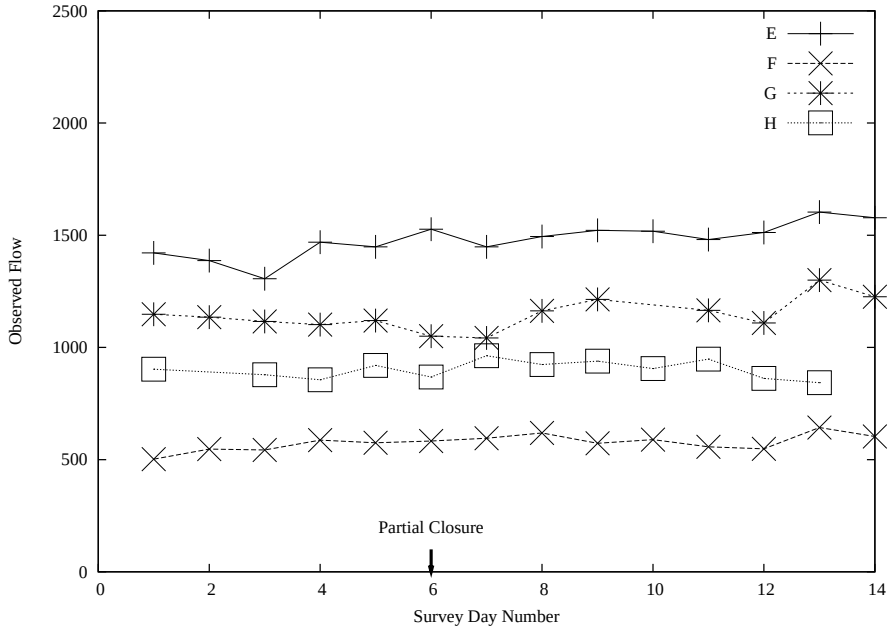

FIGURE 5.7. Fishergate survey flows on sites E–H.

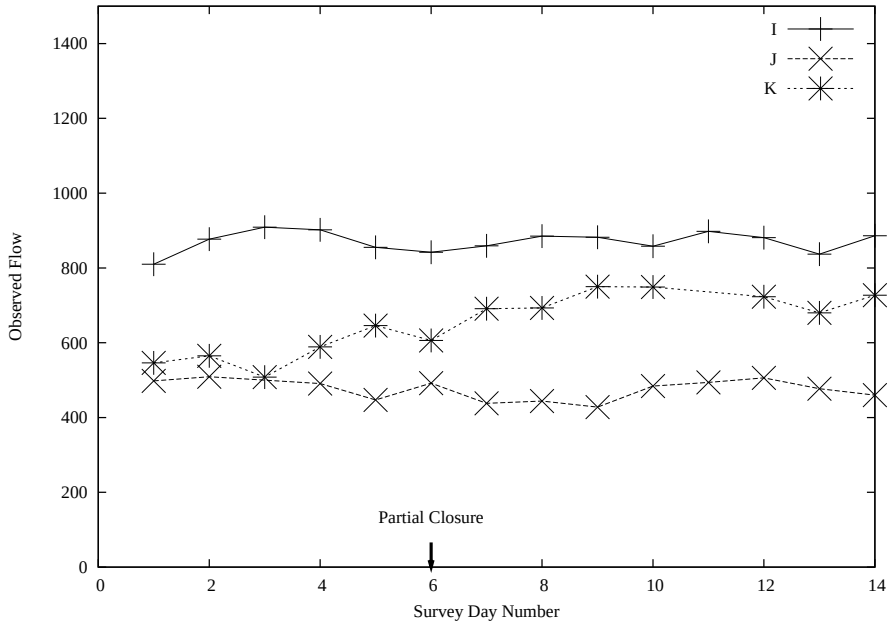

FIGURE 5.8. Fishergate survey flows on sites I–K.

A potential model is to separate the sites into those which are most likely to suffer a reduction in flow due to the closure and those which are likely to be potential diversions. The Lendal survey is complicated by the presence of the fuel crisis so this model is only applied to the Fishergate survey. Sites A, C and D are those which are closest to the incident (site B is ignored because of its low flow) and therefore most have their flows constrained. Sites F, G and K were identified as the most likely potential reroutings. The model applied is therefore

$$\mathrm{E}[p] = \beta_0 + \beta_1 I_c + \beta_2 I_r,$$

where p is the proportional flow, β_i are the model parameters, I_c (closure effects) is an indicator variable which is one if the survey in question comes from site A, C or D and the closure is in place on that day and I_r (rerouting effects) is one if the survey in question comes from site F, G or K and the closure is in effect.

Parameter	β_0	β_1	β_2			
Estimate	1.004	-0.063	0.037			
Std. Error	0.006	0.017	0.018			
Significance	0.1%	0.1%	5%			
Statistic	R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate	0.123	0.111	10.04	2	143	8.3×10^{-5}

This model seems quite successful. The p-value is low indicating that, if the assumptions of the GLM are met, then either β_1 or β_2 or both are non zero. Both beta parameters are significant, one at the 5% level and one at the 0.1% level. It seems likely, therefore, that there was a reduction on flow on the routes affected directly (estimated here at 6.3% at sites A, C and D together) and there was an increase in flow on the most obvious rerouting sites (estimated here at 3.7% at sites F, G and K together). It should be noted that the R^2 and R_a^2 values show that the majority of the variance in the flows is not explained by this model. This is only to be expected. It is probable that the fit of the model could be improved by choosing sites according to how they

maximise the fit — however, doing so would, effectively, introduce a hidden parameter into the model (the set of sites in I_c and I_r) and, hence, invalidate the p-value and the R_a^2 value calculated.

5.7.1. Flow Histogram Data by Site. The distribution of flow throughout the surveyed time can be visualised by plotting a histogram. To do so, those plates which were not removed for the reasons specified in Section 5.4 were put into five minute bins and a histogram is plotted of number of vehicles in each bin. It is should be kept in mind that, due to previously mentioned possible inaccuracies in timing, some vehicles may be misclassified by a single bin. It should also be noted that the data in the first and last survey bins may be partial if the survey began late or ended early by a few minutes. The histograms are gathered in Appendix D.

Figures D.1 to D.18 show histograms of the arrival times for selected site surveys from the Lendal Bridge Study. The times are given in minutes past midnight so they run from 480 (8:00am) to 540 (9:00am). No particularly obvious pattern emerges from the flow histograms. A large number of such figures could be generated (one for every flow reading in the cells of Tables 5.6 and 5.7). Figures D.1 to D.13 show the flows on 8/9/00 (a normal before day) at all sites. It is notable that site C (Figure D.3) shows two severe drops in flow at particular times but these are due to missing data in recording as indicated by the ‡ in Table 5.6. It is not clear from these plots whether there is a shape to the peak of the rush hour. The flow appears to be largely constant across the rush hour. This could indicate that the morning peak in York lasts longer than an hour. It could indicate either that the network is saturated for the full hour or, alternatively, that the demand on the network is extremely flat in the period specified.

Figures D.14 to D.18 (in addition to Figure D.6 already mentioned) show the histograms for a selection of other surveyed days at site F (Gillygate) which would be one site where the flow was greatly affected by the bridge closure (obviously site H would be affected more since it was completely closed apart

from buses and taxis). Figures D.6 and D.14 show before days — both of these show around thirty vehicles in the typical five minute period. The fuel crisis survey days (Figures D.15 and D.16) show a large drop in flow but still no noticeable peak structure. Figures D.17 and D.18 show the closure data with no fuel crisis and again seem to show a drop in flow compared with the before situation but no discernible structure to the peak itself.

Figures D.19 to D.29 show histograms of arrival times from selected site surveys for the Fishergate study on 2/7/01 (the last day unaffected by the partial closure) again with the flows in five minute bins. Most of the sites are surveyed from 7:45 to 9:15 (465–555 minutes past midnight) but sites H, I and J are surveyed from 8:00 to 9:30 (480–570 minutes past midnight). Sites H, I and J, Figures D.26, D.27 and D.28 seem to show a fall off in flow after 9:15. Site B (Figure D.20) is particularly unclear, presumably since the flow is so low. There is no clear evidence but, the graph of site E (Figure D.23) seems to show that the end of the rush hour is reached before the end of the survey period (this could be explained by the fact that site E was the furthest survey site from the centre of town).

Figures D.30 to D.34 in addition to Figure D.19 show the histogram for a variety of surveyed days at site A which was the site where the closure occurred. Figures D.32 and D.33 are the first two closure days and D.34 is the final closure day. There is certainly evidence that the flows are reduced after closure and this can be seen clearly from the graphs. Inspection of similar graphs (not shown) for site E shows no clear reduction in flow or change in flow pattern at this site. Site E also seems to keep the same slight decline in flow through the surveyed period as show in in Figure D.23.

Figures D.35 to D.39 show similar survey days for site D (the exception being that 27/6/01 is shown instead of 29/6/01 which was only partial data as can be seen in Table 5.8). The histograms seem to hint at a slight peaking phenomenon with lower flow at the left and right hand sides of the graphs but it is not totally clear from the data shown. A reduction in flow on the

closure days at side D is certainly present as would be expected since this is downstream from the closure point and there are few reasonable routes on the network which reach site D without going via site A.

5.8. Matching Between Pairs of Sites

In this section, matching of sites by pairs is analysed. In this case, the advanced multiple-site matching techniques discussed in Chapter 4 are not necessary. In the two site case, the method simplifies to subtracting a constant multiple of the number of pairs. However, this simple method is considered sufficient to produce a matrix of all pairs of sites to evaluate which are widely enough used for intensive study. This will be referred to as the *probabilistic correction* in the rest of the chapter.

For calculating matches between two sites on the same day, the technique used is the Maximum Likelihood Estimation (MLE) technique described in [**156**]. This model attempts to fit data to pairwise matches and assign travel times. The assumptions of the model are that vehicles travel between a number of origins and destinations with travel times that have a normal distribution for a given OD pair. The probability of a pair of plates being mistaken for a genuine match is also based upon the distribution of year letters. This method has been shown to produce robust estimates of the number of matches, the journey time and the standard deviation of the journey times when its assumptions are varied and when the distribution of travel times is not a normal distribution (for example a log-normal distribution). It should be noted that the actual mean travel time has already been shown to vary throughout the period studied and is likely to be non-normal. This method will be referred to as the *MLE correction* in the rest of this chapter. When considering matches between days, obviously the assumption of normality in travel times is more severely violated and the travel time between the two survey sites cannot be used to eliminate spurious matches.

5.8.1. Estimating $p(2)$ and $p(3)$. The parameter $p(n)$ is specified by Defnition 4.23. The value of $p(2)$ can be estimated by looking at pairs of sites where there is little or no chance that the same vehicle could actually be seen at both. This is the case for the pairs L and M and M and J in the Lendal survey and the pair E and H in the Fishergate survey. Assuming that there were no genuine matches in the surveys (a reasonable assumption given the locations of the sites) then the number of observed matches between each pair can be used to estimate $p(2)$.

Sites	No. Samples	$p(2)$	s.d.$(p(2))$
L–M Lendal	6	8.49×10^{-5}	3.81×10^{-6}
J–M Lendal	6	7.69×10^{-5}	4.16×10^{-6}
E–H Fish.	12	7.90×10^{-5}	8.97×10^{-6}
Total	24	8.00×10^{-5}	7.40×10^{-6}

It is concerning that the site pairs produce different estimates of $p(2)$. For example, t-tests show that the estimates of $p(2)$ obtained from L–M and M–J differ with a p-value of 0.0058. While it seems certain that the estimates differ it is uncertain as to why this should be – a possible explanation is that different surveyors making different errors will sometimes confuse plates and hence create more false matches (for example, a surveyor who always reports M for M and N will clearly increase the number of false matches). The difference between the lowest estimate (7.69×10^{-5}) and the highest estimate (8.49×10^{-5}) is considerable and tracking down the reasons for these differences is important further work since a good estimate of $p(n)$ for low n is critical to the matching process. The value $p(2) = 8 \times 10^{-5}$ will be used for the matching in the remainder of this chapter.

It is worth investigating the effects of separating the letters in the plate and the digits in the plate. The table below shows a column for the $p(2)$ estimate obtained from considering the partial plate estimate (as discussed above). The next column is the $p(2)$ estimate if partial plates were collected from the year letter only. The next shows the $p(2)$ estimate if partial plates were collected

using digits only. The final column shows the previous two columns multiplied together. This would be the $p(2)$ estimate from the partial plate (letters and numbers) if the year letter and digits were independent.

Sites	No. Samples	$p(2)$ Whole	$p(2)$ Letters	$p(2)$ Digits	$p(2)$ Letters $\times$ Digits
L–M Lendal	6	8.49×10^{-5}	0.0715	0.00104	7.80×10^{-5}
J–M Lendal	6	7.69×10^{-5}	0.0754	0.00103	7.43×10^{-5}
E–H Fish.	12	7.90×10^{-5}	0.0658	0.00105	6.90×10^{-5}
Total	24	8.00×10^{-5}	0.0696	0.00104	7.26×10^{-5}

From the table it is clear that the discrepencies in $p(2)$ are largely due to the year letters. One possibility is that different surveyors are likely to confuse letters in different ways. It can also be noted from this table that it seems unlikely that year letter and digits are uncorrelated which is somewhat surprising.

The value of $p(3)$ is estimated in a similar way. In the Lendal data, the sites K, L and M should have no traffic in common between any pair. In the Fishergate data, the sites, F, H and K should have no traffic in common between any pair. (Note, however, that the possibility of a small number of journeys between these match pairs cannot be eliminated.) For interest, the number of possible 3-tuples (different samples with one plate from each site) for sites K, L and M is between 1.3×10^9 and 1.6×10^9 depending on survey day. For the less heavily trafficked F, K and H the number of possible 3-tuples is between 2.2×10^8 and 5.5×10^8.

Sites	No. Samples	$\hat{p(3)}$	s.d.$(p(3))$
K, L, M	6	1.52×10^{-8}	3.23×10^{-9}
F, H, K	12	9.60×10^{-9}	3.56×10^{-9}
Total	18	1.33×10^{-8}	4.21×10^{-9}

Again, a t-test shows that the two means differ (with 5% significance) and this should be investigated in further work. A value of $p(3)$ of 1.33×10^{-8} will be assumed for the rest of this work.

Values of $p(4)$ and above are calculated by considering the distribution of year letters and assuming that the digits are uniformly distributed. Note, however, that this will almost certainly produce an underestimate. $p(2)$ calculated in this way is 0.0000644 not 0.00008 as used and $p(3)$ is 4.83×10^{-9} not 1.33×10^{-8}. The values of $p(n)$ are less critical as n increases. However, the estimation of $p(n)$ still remains a weakness in this method and needs further work.

5.8.2. Within Day Matches. Tables 5.13 to 5.18 show the matches between all the pairs of sites on the Fishergate survey. The matches are for three days chosen to include two before days and one during day for which complete data is available: 28/6/01, 2/7/01 and 3/7/01. The first two should establish the repeatability of the results and the second one should show the change caused by the intervention (if any).

In Tables 5.13, 5.15 and 5.17 the raw matches (number of pairs of plates matched between the two sites) and corrected matches using the probabilistic correction technique. The corrected matches are shown in brackets. The tables should be read by picking two sites (a row and a column) and cross referencing. So, for example, there were 29 raw matches between site A and site B and this was corrected down to 15.8. The diagonal shows the match from a site to itself. Naturally, this is usually large (of the order of the flow at that site) since every vehicle will be seen at least once in the file (it will always match with itself). The occasional small negative predicted flow is not unexpected. This is a result of the correction method overestimating the number of false matches. Naturally, these tables are symmetric about their diagonal and the entry for B–A is the same as that for A–B.

Comparing the two before days (Tables 5.13 and 5.15) the two appear largely consistent. The most noticeable difference is at site I which seems to have more matches on 28/6/01. For example, A–I, D–I, G–I and H–I show significantly lower matches on the second day surveyed. This cannot be

explained by reduced flow since the flows can be seen from Tables 5.8 and 5.9 to be almost unchanged at site I on these two days.

Comparing the before days with the first after day (Table 5.17) shows a much more significant change. This is to be expected, of course. However, a cursory inspection shows a reduction in matches which appears to be much greater than the reduction in flow (this will be confirmed statistically later in this section). The exception is at site I where the matches are mainly larger than on 2/7/01. It seems that, for whatever reason, vehicles found it hard to get to or from site I on 2/7/01.

Tables 5.14, 5.16 and 5.18 show the raw and corrected matches as percentages of the total flow on the first site. The corrected matches are in brackets. That is, the figure in row B, column E this is the percentage of vehicles at site B which are seen also at site E (34.6% uncorrected, 23.1% corrected). Conversely, row E, column B is the percentage of vehicles at site E which are seen also at site B (2.0% uncorrected 1.3% corrected). The reason for the large discrepancy in the case given is that the flow at B is extremely small (and a large percentage of it comes from site E which has a large flow). The corrected figures on the diagonals would ideally be 100% (since all vehicles are seen once in the two surveys) and, indeed, it can be seen that the figures are all around this number but some are slightly over. Indeed the estimate is as much as 111.2% at site B in Table 5.16 and 107.0% in Table 5.15. The site B result could be explained by the low flows at site B. At least some of the other overestimates could be the results of some vehicles (perhaps taxis) being seen twice at the same site during the hour and a half survey although this is speculation.

From these tables, the most significant matches in terms of percentages can be identified. From this list, a number of pairs to study can be chosen. The alternatives E–A, E–B, E–F and E–K seem to identify four possible routes to take from E and it might be expected that these include the most obvious reroutings if travel along E–A were slowed. A–D and A–J seem to be two

possible routes from A (D–J is unused — the one way system makes that route unlikely). G–C and C–A are obviously important pairs are D–I and H–I. Finally, F–G and F–A seem to carry significant traffic. This leaves out a number of pairs which have significant traffic. For example C–D seem to have a large amount of traffic but via an intermediate site A. The site pairs C–A and A–D are both studied.

Table 5.19 shows the journey times in minutes and estimated flows as calculated using the MLE correction method described at the beginning of Section 5.8. The standard deviation of the journey time is also shown in brackets after the time. The flows in this method are estimated in vehicles per hour. A number of observations can be made from this data. Firstly, on those pairs leading towards site A (E–A and C–A) an increased journey time is noted and a decreased flow. It also seems that the most severe effects are on the first day. Sites F–A seem to have the same pattern although this is less clear.

No particularly obvious effect is spotted at the possible diversion (E–F) although it is possible that the flow is increased. This increase in flow also seems to have occurred at diversion (E–K). Sites leading away from the intervention site (A–I and A–D) seem to show decreased flow (although this is not wholly clear with site D–I). Pairs further along from the diversion (D–I) and (H–I) seem to have a marginally reduced travel time which might be expected since the network would be slightly less congested. H–I also seems to be seeing a marginally increased flow which could indicate rerouting or that more flow was possible since competing flows were reduced.

In order to extract more information, a statistical model of the flows and journey times was constructed. In these models the journey times and flows from Table 5.19 were normalised to be zero mean and unit variance and then a model was constructed to attempt to explain the normalised flows and times.

The model was as follows:

$$\mathrm{E}[f] = \beta_0 + \beta_1 I_c + \beta_2 D,$$

where $\mathrm{E}[f]$ is the expectation of the normalised flow (or travel time), the β_i are the parameters of the model, I_c is an indicator variable which is one if the closure is in place and zero otherwise and D is the number of days since the closure occurred (not counting weekends) or zero if the closure is not in place. The modelling was performed separately for the data for each site pair since the responses from different site pairs might be in different directions.

Tables 5.20 and 5.21 show the parameter fitting for this model for the various sites looking at flow and travel time. Parameter estimates are given with percentage significance in brackets (or "low" if the parameter was not significant at the 10% level). For the majority of site pairs the model is a poor fit. This might be expected at some pairs. For example, sites (H–I) and (F–G) might be expected to be only weakly affected (if at all) by the intervention and the low fit to the model is to be expected.

Site pair C–A is interesting since it shows a considerable effect on travel time (the model is an extremely good fit here). The travel time is increased when the closure is in place but decreases as time goes on. The same seems to be true of site pair E–A and also F–A. However, for none of the site pairs (E–A, F–A or C–A) was the flow model a significantly good fit to the data. The times were affected but the flows were not.

Conversely, at site pair E–K, the flow was increased by the closure but no particular effect on travel time was observed. At sites A–D and A–J the flows decreased after the intervention (unsurprising since both of these pairs were directly after the intervention where the flow restriction was in place). However, no statistically significant increase in travel times on these sites was shown.

The only other site pair where a flow effect was observed was at site pair G–C which also shows a significant effect on the base travel time (the base

travel time being lower). The results of t-tests on the means show that G–C has a greater travel time and lower flow when the closure is in place with a 1% significance level. This could simply be a knock on effect from congestion at C–A although it is curious that C–A did not show a reduced flow effect with statistical significance.

The main striking feature of the results in Tables 5.20 and 5.21 is that they are completely inconsistent with the usual assumption of traffic modelling, that of the cost-flow curve. It is normally assumed that travel time between two sites is an increasing function of the flow between the two sites. This is simply not seen in this data. Indeed, almost the opposite would be hypothesised. For those sites where flow was affected, time was usually unaffected. A similar effect was noted when the histograms appeared to show a flat distribution of flow throughout the rush hour but the travel time seemed to increase and then decrease again throughout the hour. Of course for site A the saturation flow has changed and normal cost-flow relations would not apply at site pairs involving site A.

One possible explanation is that the model used all the data surveyed and there was an effect caused by journeys with missing ends. Some flow is always "missed" because the start of the journey is seen but the vehicle is not seen at the end of the survey. As travel time increases, more journeys will be "missed" like this and the flow will appear to decrease. To compensate for this effect, the data was sampled to remove some of these missing vehicles. The first site of the surveyed pair was trimmed so that the last half hour of data was removed. In this way, all journeys apart from those with unusual delays would be seen at the second site. The data from Table 5.19 is recalculated with this trimming effect and this is shown in Table 5.22. The GLM modelling is repeated and is shown in Tables 5.23 and 5.24.

In general, this trimming seems to have produced results which are broadly similar to the untrimmed results. The main differences in the travel time model is that the travel time now shows a significant difference in the closed case for

F–A and F–G. In the F–A case the closure increases the travel time in the F–G case it reduces it (this is hard behaviour to explain but this result is only significant at the 10% level – running so many models some would be expected to be significant at this level by chance). Most sites show the same trends in flow and significant parameters in general remain signficant.

The aim of the trimming procedure was to improve the flow modelling and, indeed, the model more often shows signficance in the flow modelling. This is particularly notable for site pairs C–A and H–I which previously did not have significant results in the flow model. C–A now shows an increase in flow throughout the period of the closure which could be interpreted as a return to base from an initial drop (although with only a 10% significance). H–I now shows an increase in flow during the closure which might be expected either if it were a rerouting or if traffic were able to flow more smoothly due to the reduction in congestion around H–I.

Again, what is interesting is the lack of clear relationship between flow and travel time. On some site pairs, when travel time decreases, flow also decreases (A–D, A–J, D–I). At H–I the travel time decreases but the flow increases. At E–K the flow has increased without significant effect on travel time. At C–A, a travel time increase has not been caused by a significant change in flow. Naturally, for sites directly leading to or from A then it might be expected that the cost-flow relation would change and therefore a clear relationship between flow and travel time would not be expected. However, the other sites (for example G–C) do not appear to be showing the cost-flow effects in the direction expected.

In all cases, when β_2 is significant then it is of the opposite sign to β_1 which supports the idea of an initial response to an intervention which dies down as time goes on.

	A	B	C	D
A	2515 (2182.4)	29 (15.8)	992 (832.1)	1346 (1144.5)
B	29 (15.8)	83 (82.5)	21 (14.6)	15 (7.0)
C	992 (832.1)	21 (14.6)	1042 (965.2)	635 (538.2)
D	1346 (1144.5)	15 (7.0)	635 (538.2)	1335 (1213.0)
E	581 (347.9)	28 (18.7)	114 (2.0)	244 (102.8)
F	214 (122.7)	3 (-0.6)	41 (-2.9)	120 (64.7)
G	623 (447.2)	11 (4.0)	532 (447.5)	369 (262.5)
H	185 (47.5)	6 (0.5)	70 (3.9)	120 (36.7)
I	362 (218.3)	6 (0.3)	131 (61.9)	266 (179.0)
J	318 (242.0)	11 (8.0)	121 (84.5)	71 (25.0)
K	122 (28.2)	5 (1.3)	51 (5.9)	69 (12.2)
	E	F	G	H
A	581 (347.9)	214 (122.7)	623 (447.2)	185 (47.5)
B	28 (18.7)	3 (-0.6)	11 (4.0)	6 (0.5)
C	114 (2.0)	41 (-2.9)	532 (447.5)	70 (3.9)
D	244 (102.8)	120 (64.7)	369 (262.5)	120 (36.7)
E	1631 (1467.6)	115 (51.0)	116 (-7.2)	91 (-5.4)
F	115 (51.0)	586 (560.9)	85 (36.7)	42 (4.2)
G	116 (-7.2)	85 (36.7)	1160 (1067.0)	84 (11.3)
H	91 (-5.4)	42 (4.2)	84 (11.3)	951 (894.1)
I	148 (47.3)	49 (9.5)	150 (74.0)	450 (390.6)
J	107 (53.7)	41 (20.1)	109 (68.8)	83 (51.6)
K	364 (298.3)	31 (5.2)	58 (8.4)	31 (-7.8)
	I	J	K	
A	362 (218.3)	318 (242.0)	122 (28.2)	
B	6 (0.3)	11 (8.0)	5 (1.3)	
C	131 (61.9)	121 (84.5)	51 (5.9)	
D	266 (179.0)	71 (25.0)	69 (12.2)	
E	148 (47.3)	107 (53.7)	364 (298.3)	
F	49 (9.5)	41 (20.1)	31 (5.2)	
G	150 (74.0)	109 (68.8)	58 (8.4)	
H	450 (390.6)	83 (51.6)	31 (-7.8)	
I	1017 (954.9)	82 (49.2)	30 (-10.5)	
J	82 (49.2)	510 (492.6)	26 (4.6)	
K	30 (-10.5)	26 (4.6)	609 (582.5)	

TABLE 5.13. Fishergate Survey 28/6/2001. Raw matches and corrected matches between each pair of sites.

	A	B	C	D
A	123.3 (107.0)	1.4 (0.8)	48.7 (40.8)	66.0 (56.1)
B	35.8 (19.5)	102.5 (101.8)	25.9 (18.1)	18.5 (8.6)
C	101.2 (84.9)	2.1 (1.5)	106.3 (98.5)	64.8 (54.9)
D	109.0 (92.7)	1.2 (0.6)	51.4 (43.6)	108.1 (98.2)
E	40.7 (24.3)	2.0 (1.3)	8.0 (0.1)	17.1 (7.2)
F	38.2 (21.9)	0.5 (-0.1)	7.3 (-0.5)	21.4 (11.5)
G	57.8 (41.5)	1.0 (0.4)	49.4 (41.5)	34.2 (24.4)
H	21.9 (5.6)	0.7 (0.1)	8.3 (0.5)	14.2 (4.4)
I	41.1 (24.8)	0.7 (0.0)	14.9 (7.0)	30.2 (20.3)
J	68.2 (51.9)	2.4 (1.7)	26.0 (18.1)	15.2 (5.4)
K	21.2 (4.9)	0.9 (0.2)	8.9 (1.0)	12.0 (2.1)
	E	F	G	H
A	28.5 (17.1)	10.5 (6.0)	30.6 (21.9)	9.1 (2.3)
B	34.6 (23.1)	3.7 (-0.8)	13.6 (5.0)	7.4 (0.7)
C	11.6 (0.2)	4.2 (-0.3)	54.3 (45.7)	7.1 (0.4)
D	19.8 (8.3)	9.7 (5.2)	29.9 (21.3)	9.7 (3.0)
E	114.1 (102.7)	8.0 (3.6)	8.1 (-0.5)	6.4 (-0.4)
F	20.5 (9.1)	104.6 (100.2)	15.2 (6.6)	7.5 (0.8)
G	10.8 (-0.7)	7.9 (3.4)	107.6 (99.0)	7.8 (1.0)
H	10.8 (-0.6)	5.0 (0.5)	10.0 (1.3)	112.8 (106.1)
I	16.8 (5.4)	5.6 (1.1)	17.0 (8.4)	51.1 (44.3)
J	23.0 (11.5)	8.8 (4.3)	23.4 (14.8)	17.8 (11.1)
K	63.3 (51.9)	5.4 (0.9)	10.1 (1.5)	5.4 (-1.4)
	I	J	K	
A	17.8 (10.7)	15.6 (11.9)	6.0 (1.4)	
B	7.4 (0.4)	13.6 (9.9)	6.2 (1.6)	
C	13.4 (6.3)	12.3 (8.6)	5.2 (0.6)	
D	21.5 (14.5)	5.7 (2.0)	5.6 (1.0)	
E	10.4 (3.3)	7.5 (3.8)	25.5 (20.9)	
F	8.8 (1.7)	7.3 (3.6)	5.5 (0.9)	
G	13.9 (6.9)	10.1 (6.4)	5.4 (0.8)	
H	53.4 (46.3)	9.8 (6.1)	3.7 (-0.9)	
I	115.4 (108.4)	9.3 (5.6)	3.4 (-1.2)	
J	17.6 (10.5)	109.4 (105.7)	5.6 (1.0)	
K	5.2 (-1.8)	4.5 (0.8)	105.9 (101.3)	

TABLE 5.14. Fishergate Survey 28/6/2001. Raw and corrected matches as percentage of flow.

	A	B	C	D
A	2441 (2123.2)	30 (14.1)	1018 (858.1)	1301 (1114.1)
B	30 (14.1)	112 (111.2)	24 (16.0)	15 (5.6)
C	1018 (858.1)	24 (16.0)	1075 (994.5)	659 (565.0)
D	1301 (1114.1)	15 (5.6)	659 (565.0)	1278 (1168.1)
E	574 (336.8)	32 (20.1)	109 (-10.4)	242 (102.5)
F	197 (107.1)	13 (8.5)	47 (1.7)	110 (57.1)
G	603 (438.5)	17 (8.7)	554 (471.2)	367 (270.2)
H	185 (48.8)	17 (10.2)	89 (20.5)	115 (34.9)
I	277 (145.0)	13 (6.4)	121 (54.6)	204 (126.4)
J	303 (227.3)	7 (3.2)	113 (74.9)	42 (-2.5)
K	120 (26.1)	8 (3.3)	73 (25.7)	72 (16.8)
	E	F	G	H
A	574 (336.8)	197 (107.1)	603 (438.5)	185 (48.8)
B	32 (20.1)	13 (8.5)	17 (8.7)	17 (10.2)
C	109 (-10.4)	47 (1.7)	554 (471.2)	89 (20.5)
D	242 (102.5)	110 (57.1)	367 (270.2)	115 (34.9)
E	1676 (1498.9)	125 (57.9)	96 (-26.8)	120 (18.3)
F	125 (57.9)	590 (564.6)	90 (43.4)	46 (7.5)
G	96 (-26.8)	90 (43.4)	1142 (1056.8)	72 (1.5)
H	120 (18.3)	46 (7.5)	72 (1.5)	930 (871.7)
I	127 (28.4)	52 (14.6)	101 (32.6)	373 (316.4)
J	120 (63.5)	35 (13.6)	113 (73.8)	85 (52.5)
K	381 (310.9)	38 (11.4)	75 (26.4)	48 (7.8)
	I	J	K	
A	277 (145.0)	303 (227.3)	120 (26.1)	
B	13 (6.4)	7 (3.2)	8 (3.3)	
C	121 (54.6)	113 (74.9)	73 (25.7)	
D	204 (126.4)	42 (-2.5)	72 (16.8)	
E	127 (28.4)	120 (63.5)	381 (310.9)	
F	52 (14.6)	35 (13.6)	38 (11.4)	
G	101 (32.6)	113 (73.8)	75 (26.4)	
H	373 (316.4)	85 (52.5)	48 (7.8)	
I	900 (845.2)	50 (18.5)	42 (3.0)	
J	50 (18.5)	511 (492.9)	30 (7.6)	
K	42 (3.0)	30 (7.6)	631 (603.2)	

TABLE 5.15. Fishergate Survey 2/7/2001. Raw matches and corrected matches between each pair of sites.

	A	B	C	D
A	122.5 (106.5)	1.5 (0.7)	51.1 (43.1)	65.3 (55.9)
B	30.0 (14.1)	112.0 (111.2)	24.0 (16.0)	15.0 (5.6)
C	101.5 (85.6)	2.4 (1.6)	107.2 (99.2)	65.7 (56.3)
D	111.0 (95.1)	1.3 (0.5)	56.2 (48.2)	109.0 (99.7)
E	38.6 (22.6)	2.2 (1.4)	7.3 (-0.7)	16.3 (6.9)
F	34.9 (19.0)	2.3 (1.5)	8.3 (0.3)	19.5 (10.1)
G	58.4 (42.5)	1.6 (0.8)	53.7 (45.7)	35.6 (26.2)
H	21.7 (5.7)	2.0 (1.2)	10.4 (2.4)	13.5 (4.1)
I	33.5 (17.5)	1.6 (0.8)	14.6 (6.6)	24.6 (15.3)
J	63.8 (47.8)	1.5 (0.7)	23.8 (15.8)	8.8 (-0.5)
K	20.4 (4.4)	1.4 (0.6)	12.4 (4.4)	12.2 (2.8)
	E	F	G	H
A	28.8 (16.9)	9.9 (5.4)	30.3 (22.0)	9.3 (2.5)
B	32.0 (20.1)	13.0 (8.5)	17.0 (8.7)	17.0 (10.2)
C	10.9 (-1.0)	4.7 (0.2)	55.2 (47.0)	8.9 (2.0)
D	20.6 (8.7)	9.4 (4.9)	31.3 (23.1)	9.8 (3.0)
E	112.6 (100.7)	8.4 (3.9)	6.5 (-1.8)	8.1 (1.2)
F	22.2 (10.3)	104.6 (100.1)	16.0 (7.7)	8.2 (1.3)
G	9.3 (-2.6)	8.7 (4.2)	110.7 (102.4)	7.0 (0.1)
H	14.1 (2.1)	5.4 (0.9)	8.4 (0.2)	108.9 (102.1)
I	15.3 (3.4)	6.3 (1.8)	12.2 (3.9)	45.0 (38.2)
J	25.3 (13.4)	7.4 (2.9)	23.8 (15.5)	17.9 (11.1)
K	64.7 (52.8)	6.5 (1.9)	12.7 (4.5)	8.1 (1.3)
	I	J	K	
A	13.9 (7.3)	15.2 (11.4)	6.0 (1.3)	
B	13.0 (6.4)	7.0 (3.2)	8.0 (3.3)	
C	12.1 (5.4)	11.3 (7.5)	7.3 (2.6)	
D	17.4 (10.8)	3.6 (-0.2)	6.1 (1.4)	
E	8.5 (1.9)	8.1 (4.3)	25.6 (20.9)	
F	9.2 (2.6)	6.2 (2.4)	6.7 (2.0)	
G	9.8 (3.2)	10.9 (7.1)	7.3 (2.6)	
H	43.7 (37.1)	10.0 (6.2)	5.6 (0.9)	
I	108.7 (102.1)	6.0 (2.2)	5.1 (0.4)	
J	10.5 (3.9)	107.6 (103.8)	6.3 (1.6)	
K	7.1 (0.5)	5.1 (1.3)	107.1 (102.4)	

TABLE 5.16. Fishergate Survey 2/7/2001. Raw and corrected matches as percentage of flow.

	A	B	C	D
A	1881 (1657.1)	16 (3.8)	859 (742.2)	1102 (962.9)
B	16 (3.8)	91 (90.3)	17 (10.6)	11 (3.4)
C	859 (742.2)	17 (10.6)	939 (878.0)	577 (504.4)
D	1102 (962.9)	11 (3.4)	577 (504.4)	1121 (1034.6)
E	393 (203.9)	19 (8.7)	83 (-15.7)	175 (57.6)
F	198 (121.4)	11 (6.8)	44 (4.1)	114 (66.5)
G	540 (403.3)	10 (2.6)	507 (435.7)	338 (253.1)
H	143 (15.7)	10 (3.1)	67 (0.6)	102 (23.0)
I	211 (99.4)	5 (-1.1)	94 (35.8)	168 (98.7)
J	215 (158.5)	4 (0.9)	87 (57.5)	26 (-9.1)
K	120 (29.5)	5 (0.1)	58 (10.8)	75 (18.8)
	E	F	G	H
A	393 (203.9)	198 (121.4)	540 (403.3)	143 (15.7)
B	19 (8.7)	11 (6.8)	10 (2.6)	10 (3.1)
C	83 (-15.7)	44 (4.1)	507 (435.7)	67 (0.6)
D	175 (57.6)	114 (66.5)	338 (253.1)	102 (23.0)
E	1593 (1433.3)	146 (81.3)	101 (-14.4)	105 (-2.5)
F	146 (81.3)	618 (591.8)	104 (57.3)	51 (7.5)
G	101 (-14.4)	104 (57.3)	1105 (1021.6)	96 (18.3)
H	105 (-2.5)	51 (7.5)	96 (18.3)	1041 (968.6)
I	118 (23.7)	51 (12.8)	116 (47.9)	453 (389.5)
J	80 (32.3)	46 (26.7)	111 (76.5)	83 (50.9)
K	414 (337.6)	34 (3.1)	53 (-2.2)	51 (-0.4)
	I	J	K	
A	211 (99.4)	215 (158.5)	120 (29.5)	
B	5 (-1.1)	4 (0.9)	5 (0.1)	
C	94 (35.8)	87 (57.5)	58 (10.8)	
D	168 (98.7)	26 (-9.1)	75 (18.8)	
E	118 (23.7)	80 (32.3)	414 (337.6)	
F	51 (12.8)	46 (26.7)	34 (3.1)	
G	116 (47.9)	111 (76.5)	53 (-2.2)	
H	453 (389.5)	83 (50.9)	51 (-0.4)	
I	918 (862.4)	51 (22.8)	60 (14.9)	
J	51 (22.8)	462 (447.8)	23 (0.2)	
K	60 (14.9)	23 (0.2)	718 (681.4)	

TABLE 5.17. Fishergate Survey 3/7/2001. Raw matches and corrected matches between each pair of sites.

	A	B	C	D
A	112.4 (99.0)	1.0 (0.2)	51.3 (44.4)	65.9 (57.6)
B	17.6 (4.2)	100.0 (99.3)	18.7 (11.7)	12.1 (3.8)
C	98.4 (85.0)	1.9 (1.2)	107.6 (100.6)	66.1 (57.8)
D	106.1 (92.7)	1.1 (0.3)	55.5 (48.6)	107.9 (99.6)
E	27.8 (14.4)	1.3 (0.6)	5.9 (-1.1)	12.4 (4.1)
F	34.6 (21.2)	1.9 (1.2)	7.7 (0.7)	19.9 (11.6)
G	52.9 (39.5)	1.0 (0.3)	49.7 (42.7)	33.1 (24.8)
H	15.0 (1.7)	1.1 (0.3)	7.0 (0.1)	10.7 (2.4)
I	25.3 (11.9)	0.6 (-0.1)	11.3 (4.3)	20.1 (11.8)
J	50.9 (37.6)	0.9 (0.2)	20.6 (13.6)	6.2 (-2.2)
K	17.8 (4.4)	0.7 (0.0)	8.6 (1.6)	11.1 (2.8)
	E	F	G	H
A	23.5 (12.2)	11.8 (7.3)	32.3 (24.1)	8.5 (0.9)
B	20.9 (9.6)	12.1 (7.5)	11.0 (2.8)	11.0 (3.4)
C	9.5 (-1.8)	5.0 (0.5)	58.1 (49.9)	7.7 (0.1)
D	16.8 (5.5)	11.0 (6.4)	32.5 (24.4)	9.8 (2.2)
E	112.7 (101.4)	10.3 (5.8)	7.1 (-1.0)	7.4 (-0.2)
F	25.5 (14.2)	108.0 (103.5)	18.2 (10.0)	8.9 (1.3)
G	9.9 (-1.4)	10.2 (5.6)	108.2 (100.1)	9.4 (1.8)
H	11.0 (-0.3)	5.4 (0.8)	10.1 (1.9)	109.5 (101.9)
I	14.1 (2.8)	6.1 (1.5)	13.9 (5.7)	54.3 (46.7)
J	19.0 (7.7)	10.9 (6.3)	26.3 (18.1)	19.7 (12.1)
K	61.2 (49.9)	5.0 (0.5)	7.8 (-0.3)	7.5 (-0.1)
	I	J	K	
A	12.6 (5.9)	12.9 (9.5)	7.2 (1.8)	
B	5.5 (-1.2)	4.4 (1.0)	5.5 (0.1)	
C	10.8 (4.1)	10.0 (6.6)	6.6 (1.2)	
D	16.2 (9.5)	2.5 (-0.9)	7.2 (1.8)	
E	8.4 (1.7)	5.7 (2.3)	29.3 (23.9)	
F	8.9 (2.2)	8.0 (4.7)	5.9 (0.5)	
G	11.4 (4.7)	10.9 (7.5)	5.2 (-0.2)	
H	47.6 (41.0)	8.7 (5.4)	5.4 (0.0)	
I	110.1 (103.4)	6.1 (2.7)	7.2 (1.8)	
J	12.1 (5.4)	109.5 (106.1)	5.5 (0.0)	
K	8.9 (2.2)	3.4 (0.0)	106.2 (100.8)	

TABLE 5.18. Fishergate Survey 3/7/2001. Raw and corrected matches as percentage of flow.

Date	t (s.d.)	Flow	t (s.d.)	Flow	t (s.d.)	Flow	t (s.d.)	Flow
	E–A		E–B		E–F		E–K	
25/6/01	7.52(2.23)	279	—	—	6.50(0.94)	38	7.24(2.25)	241
26/6/01	9.15(1.75)	244	10.73(2.11)	15	7.48(1.22)	50	8.59(1.32)	205
27/6/01	9.34(2.1421)	236	9.60(1.67)	15	8.02(2.43)	49	9.04(2.66)	215
28/6/01	8.34(2.97)	258	8.54(1.60)	13	7.35(1.81)	51	5.36(1.84)	251
29/6/01	5.64(1.09)	258	6.84(2.28)	19	7.31(1.41)	59	5.24(0.85)	235
2/7/01	5.97(1.61)	258	5.58(1.53)	19	7.18(1.15)	51	5.47(1.07)	259
3/7/01	12.63(8.23)	237	9.50(4.73)	12	6.81(1.04)	52	6.46(3.20)	303
4/7/01	11.93(5.48)	236	10.31(3.17)	13	7.36(1.75)	56	5.90(1.95)	282
5/7/01	10.89(4.88)	233	10.38(7.83)	13	7.13(1.96)	68	4.85(0.94)	295
6/7/01	8.19(4.09)	253	6.15(2.21)	13	6.97(1.31)	59	4.73(1.03)	294
11/7/01	8.41(4.01)	232	11.18(5.39)	11	7.04(1.27)	57	—	—
12/7/01	9.75(4.60)	244	11.15(8.04)	13	6.94(0.77)	52	5.03(1.35)	277
13/7/01	5.20(1.07)	229	6.79(3.59)	14	7.70(1.80)	96	4.62(0.98)	273
16/7/01	9.35(5.29)	271	9.24(5.00)	17	6.41(0.81)	59	4.99(1.27)	289
Date	A–D		A–J		G–C		C–A	
25/6/01	0.61(0.76)	776	5.32(0.89)	151	1.66(0.66)	393	0.63(0.69)	652
26/6/01	0.21(0.76)	808	4.17(0.73)	176	1.48(0.88)	401	1.33(0.70)	680
27/6/01	0.39(0.63)	847	—	—	2.34(1.15)	379	0.90(0.60)	662
28/6/01	0.32(0.66)	889	4.01(0.86)	160	1.32(0.70)	381	1.30(0.85)	654
29/6/01	—	—	4.90(1.08)	155	1.00(0.62)	371	1.22(0.64)	657
2/7/01	0.27(0.72)	857	3.93(0.89)	164	2.07(0.69)	387	1.17(0.64)	673
3/7/01	0.25(0.59)	768	4.11(0.95)	128	2.46(1.14)	366	3.29(1.35)	610
4/7/01	0.59(0.52)	776	3.75(0.73)	131	2.16(0.75)	361	2.66(1.23)	678
5/7/01	0.46(0.67)	741	3.74(0.87)	111	2.87(1.24)	361	3.27(1.59)	625
6/7/01	0.52(0.60)	771	3.70(0.73)	142	—	—	2.13(0.98)	630
11/7/01	—	—	3.79(0.83)	147	3.02(1.56)	344	2.26(1.32)	561
12/7/01	0.72(0.58)	705	3.80(0.78)	143	3.55(1.50)	340	2.33(1.27)	585
13/7/01	0.92(0.53)	836	3.68(1.10)	148	2.23(0.82)	416	1.17(0.71)	590
16/7/01	0.94(0.80)	767	3.74(0.77)	134	2.51(0.67)	397	1.62(1.20)	647
Date	D–I		H–I		F–G		F–A	
25/6/01	4.01(0.73)	74	1.29(0.82)	221	2.62(0.76)	29	4.19(1.52)	91
26/6/01	5.07(1.64)	85	—	—	2.74(0.87)	35	3.95(0.83)	87
27/6/01	5.07(1.42)	96	1.25(0.70)	224	3.39(1.67)	33	3.82(0.83)	88
28/6/01	4.68(1.26)	100	1.08(0.62)	221	2.88(1.01)	33	4.03(1.56)	110
29/6/01	—	—	1.30(0.71)	228	3.03(0.83)	39	3.57(0.97)	99
2/7/01	7.09(2.15)	85	1.02(0.67)	192	1.29(0.61)	24	3.29(0.99)	101
3/7/01	3.46(1.15)	63	0.97(0.61)	233	1.50(0.74)	34	5.58(1.92)	104
4/7/01	4.35(1.74)	72	1.42(0.80)	220	1.61(0.72)	36	6.48(2.50)	105
5/7/01	3.99(1.05)	70	1.13(0.70)	236	2.02(1.42)	52	4.59(1.26)	93
6/7/01	4.24(1.56)	101	1.23(0.82)	215	—	—	3.79(0.89)	95
11/7/01	—	—	1.29(0.64)	234	2.77(1.45)	30	6.36(2.72)	72
12/7/01	4.05(1.45)	82	1.61(0.71)	213	1.00(0.77)	27	4.04(1.61)	67
13/7/01	5.48(1.86)	106	1.37(0.61)	180	2.17(1.28)	36	3.62(1.05)	88
16/7/01	5.95(1.58)	100	—	—	1.64(1.21)	36	4.36(2.01)	106

TABLE 5.19. Journey times and flows for Fishergate survey.

Pair	β_0 (sig)	β_1 (sig)	β_2 (sig)	R^2	R_a^2	p-value
A – D	-0.25 (low)	-0.62 (low)	0.24 (10%)	0.40	0.27	0.096
A – J	0.57 (low)	-0.88 (low)	-0.03 (low)	0.31	0.17	0.16
C – A	-0.82 (0.1%)	2.5 (0.1%)	-0.18 (1%)	0.86	0.87	6.5×10^6
D – I	0.44 (low)	-1.7 (10%)	0.18 (low)	0.39	0.26	0.10
E – A	-0.64 (10%)	2.1 (1%)	-0.15 (low)	0.57	0.49	0.0095
E – F	0.47 (low)	-0.34 (low)	-0.12(low)	0.32	0.20	0.12
E – K	0.37 (low)	-0.44 (low)	-0.08(low)	0.21	0.05	0.31
F – A	-0.62 (10%)	1.80 (5%)	-0.12 (low)	0.48	0.39	0.027
F – G	0.51 (low)	-1.10 (low)	0.0023 (low)	0.33	0.19	0.14
G – C	-0.66 (5%)	0.96 (low)	0.09 (low)	0.61	0.53	0.0095
H – I	-0.16 (low)	-0.97 (low)	0.31 (10%)	0.37	0.22	0.13

TABLE 5.20. GLM modelling results for travel times at various site pairs for the Fishergate survey.

Pair	β_0 (sig)	β_1 (sig)	β_2 (sig)	R^2	R_a^2	p-value
A – D	0.76 (5%)	-1.25 (5%)	-0.058 (low)	0.65	0.57	0.0089
A – J	0.80 (5%)	-2.0 (1%)	0.11 (low)	0.66	0.59	0.0045
C – A	0.44 (low)	-0.41 (low)	-0.09 (low)	0.26	0.12	0.19
D – I	0.34 (low)	-1.70 (5%)	0.22 (low)	0.40	0.27	0.098
E – A	0.26 (low)	-1.36 (low)	0.17 (low)	0.22	0.08	0.26
E – F	-0.049 (low)	0.13 (low)	-0.0067 (low)	0.0028	-0.18	0.98
E – K	-0.73 (5%)	1.79 (1%)	-0.04 (low)	0.69	0.62	0.0031
F – A	0.13 (low)	0.55 (low)	-0.16 (low)	0.15	-0.0013	0.40
F – G	-0.21 (low)	1.17 (low)	-0.14 (low)	0.16	-0.0039	0.41
G – C	0.58 (10%)	-1.47 (10%)	0.042 (low)	0.44	0.33	0.056
H – I	-0.42 (low)	1.14 (low)	-0.071 (low)	0.21	0.035	0.34

TABLE 5.21. GLM modelling results for flows at various site pairs for the Fishergate survey.

Date	t (s.d.)	Flow	t (s.d.)	Flow	t (s.d.)	Flow	t (s.d.)	Flow
	E–A		E–B		E–F		E–K	
25/6/01	7.28(2.27)	242	—	—	6.52(0.97)	29	7.00(2.20)	220
26/6/01	9.02(1.81)	215	10.85(2.25)	13	7.51(1.30)	43	8.51(1.31)	191
27/6/01	9.12(2.16)	209	9.29(1.67)	14	8.09(2.63)	43	8.70(2.50)	193
28/6/01	8.44(3.12)	220	10.08(3.52)	13	7.40(1.86)	48	5.30(1.78)	228
29/6/01	5.47(1.04)	214	6.76(2.39)	17	7.48(1.46)	48	5.27(0.81)	215
2/7/01	5.79(1.59)	224	5.35(1.45)	17	7.12(1.11)	48	5.41(1.08)	237
3/7/01	12.12(7.98)	224	9.50(4.73)	12	6.76(1.03)	50	6.12(2.91)	287
4/7/01	10.64(4.43)	207	10.31(3.17)	13	7.43(1.92)	54	5.46(1.48)	253
5/7/01	10.03(4.62)	199	10.38(7.83)	13	6.66(1.29)	62	4.70(0.82)	265
6/7/01	7.36(3.41)	224	6.15(2.21)	13	7.04(1.35)	53	4.69(1.03)	259
11/7/01	7.83(3.68)	206	9.33(3.74)	9	7.26(1.52)	50	—	—
12/7/01	9.16(4.53)	216	11.00(7.76)	14	6.90(0.78)	39	4.62(0.89)	230
13/7/01	5.13(1.09)	202	5.70(1.00)	10	7.55(1.72)	88	4.48(0.89)	239
16/7/01	8.81(5.09)	253	8.82(4.53)	17	6.36(0.74)	56	4.70(1.08)	251
Date	A–D		A–J		G–C		C–A	
25/6/01	0.60(0.78)	688	5.33(0.93)	135	1.73(0.58)	358	0.54(0.65)	586
26/6/01	0.23(0.79)	711	4.20(0.73)	158	1.45(0.88)	371	1.32(0.72)	594
27/6/01	0.38(0.64)	727	—	—	2.19(1.05)	349	0.89(0.61)	579
28/6/01	0.29(0.67)	783	3.96(0.78)	136	1.29(0.72)	333	1.36(0.84)	569
29/6/01	—	—	4.81(1.08)	134	1.04(0.62)	330	1.17(0.61)	588
2/7/01	0.24(0.73)	767	3.81(0.88)	137	2.06(0.66)	332	1.12(0.64)	582
3/7/01	0.25(0.60)	672	4.19(0.94)	113	2.41(1.16)	337	3.13(1.31)	542
4/7/01	0.61(0.51)	682	3.78(0.72)	117	2.05(0.68)	324	2.57(1.26)	578
5/7/01	0.40(0.68)	663	3.72(0.90)	103	2.94(1.28)	326	3.04(1.52)	548
6/7/01	0.52(0.61)	683	3.56(0.64)	118	—	—	2.08(1.01)	556
11/7/01	—	—	3.81(0.85)	136	2.73(1.38)	308	2.09(1.24)	501
12/7/01	0.72(0.58)	627	3.88(0.75)	127	3.49(1.53)	312	2.10(1.11)	522
13/7/01	0.93(0.53)	747	3.58(1.06)	133	2.23(0.85)	378	1.17(0.72)	528
16/7/01	0.79(1.09)	694	3.77(0.77)	112	2.51(0.67)	370	1.37(0.98)	576
Date	D–I		H–I		F–G		F–A	
25/6/01	3.90(0.72)	61	1.29(0.86)	187	2.62(0.75)	24	4.17(1.54)	82
26/6/01	4.87(1.56)	78	—	—	2.97(0.90)	31	3.96(0.80)	82
27/6/01	4.94(1.48)	83	1.19(0.70)	187	3.20(1.88)	25	3.86(0.86)	77
28/6/01	4.53(1.22)	87	1.04(0.62)	189	2.87(1.02)	30	4.02(1.60)	96
29/6/01	—	—	1.30(0.73)	195	2.94(0.84)	34	3.47(0.94)	85
2/7/01	7.04(2.20)	78	0.98(0.69)	172	1.32(0.65)	19	3.29(1.03)	92
3/7/01	3.48(1.24)	54	0.92(0.59)	202	1.32(0.60)	28	5.20(1.48)	92
4/7/01	4.60(1.70)	63	1.22(0.59)	188	1.66(0.69)	32	6.33(2.57)	94
5/7/01	3.87(1.03)	63	1.11(0.69)	198	1.89(1.30)	44	4.45(1.16)	87
6/7/01	4.42(1.19)	83	1.31(0.81)	186	—	—	3.78(0.94)	82
11/7/01	—	—	1.34(0.60)	204	2.78(1.52)	27	5.85(2.57)	60
12/7/01	4.03(1.50)	74	1.48(0.59)	182	0.91(0.65)	23	3.76(1.29)	62
13/7/01	5.65(1.95)	95	1.31(0.58)	150	2.31(1.72)	35	3.66(1.06)	83
16/7/01	5.91(1.68)	78	—	—	1.68(1.28)	31	4.02(1.90)	91

TABLE 5.22. Journey times and flows for Fishergate survey — adjusted by trimming.

Pair	β_0 (sig)	β_1 (sig)	β_2 (sig)	R^2	R_a^2	p-value
A – D	-0.58 (5%)	-0.32 (low)	0.24 (0.1%)	0.74	0.68	0.002
A – J	0.65 (low)	-0.54 (low)	-0.10 (low)	0.35	0.22	0.12
C – A	-0.67 (1%)	1.9 (0.1%)	-0.093 (10%)	0.76	0.71	0.00040
D – I	0.21 (low)	-1.11 (10%)	0.11 (low)	0.31	0.15	0.19
E – A	-0.37 (low)	1.65 (1%)	-0.14 (5%)	0.55	0.47	0.012
E – F	0.51 (low)	-0.92 (low)	-0.02 (low)	0.26	0.12	0.19
E – K	0.59 (low)	-0.49 (low)	-0.13 (low)	0.41	0.30	0.069
F – A	-0.44 (low)	1.41 (5%)	-0.08 (low)	0.40	0.30	0.058
F – G	0.56 (low)	-1.11 (10%)	-0.016 (low)	0.38	0.25	0.09
G – C	-0.78 (5%)	1.06 (5%)	-0.094 (low)	0.65	0.58	0.0054
H – I	-0.53 (low)	-0.25 (low)	0.20 (5%)	0.49	0.39	0.034

TABLE 5.23. GLM modelling results for travel times at various site pairs for the Fishergate survey — adjusted by trimming.

Pair	β_0 (sig)	β_1 (sig)	β_2 (sig)	R^2	R_a^2	p-value
A – D	0.73 (5%)	-1.50 (1%)	0.0062 (low)	0.60	0.51	0.017
A – J	0.74 (5%)	-1.49 (1%)	0.017 (low)	0.55	0.47	0.018
C – A	0.68 (10%)	-0.51 (low)	0.067 (10%)	0.47	0.38	0.030
D – I	0.21 (low)	-1.45 (5%)	0.17 (5%)	0.57	0.47	0.023
E – A	-0.041 (low)	-0.047 (low)	0.020 (low)	0.0046	-0.18	0.97
E – F	-0.30 (low)	-0.43 (low)	0.17 (10%)	0.30	0.17	0.14
E – K	-0.65 (5%)	1.48 (1%)	-0.012(low)	0.56	0.48	0.016
F – A	0.33 (low)	0.03 (low)	-0.11(low)	0.17	0.017	0.36
F – G	-0.20 (low)	0.35 (low)	0.013 (low)	0.045	-0.15	0.79
G – C	0.27 (low)	-1.35 (5%)	0.11 (low)	0.37	0.24	0.10
H – I	-0.16 (low)	1.44 (1%)	-0.19 (5%)	0.60	0.51	0.02

TABLE 5.24. GLM modelling results for flows at various site pairs for the Fishergate survey — adjusted by trimming.

5.8.3. Between Day Matches. Tables 5.25 to 5.29 show matches between different days at the same site. The term *recurrence rate* refers to the percentage of traffic on day A which are seen again on day B. These tables show recurrence rates for the two surveys. In each cell of the tables, the number represents the percentage of vehicles which are seen on the day represented by a given row are seen again on the day represented by a given column. It should be noted that in all tables except Table 5.27 the recurrence rate is calculated on data between 8:00am and 9:00am on all days. Note that the adjusted figure on the diagonal should always be close to 100%.

Table 5.25 shows matching at site L in the Lendal Bridge survey. This site was picked because it should have been least affected by the bridge closure and hence, apart from on the two fuel crisis days, the recurrence rates should be unaffected by interventions on the network. The most obvious thing is that the recurrence rate falls off rapidly with the separation between the two days matched. For surveys within one day of each other, the adjusted recurrence rate is between 35% and 40%. At two days this appears to have fallen to between 30% and 35% (although the 13/9/00 data may be affected by the fuel crisis). By the time the days are more than two months apart (27 and 28/6/00 versus 6,7 and 8/9/00) the recurrence rate has fallen to between 15% and 18%. The recurrence rate between the 27/9/00 and 18/10/00 is unexpectedly high (32%) considering they are three weeks apart — this could be due to the fact that both days are a Wednesday. Indeed this hypothesis seems to be confirmed by the Fishergate data and will be further confirmed by a GLM later in this section.

Table 5.26 shows the recurrence rates for the data at Fishergate survey site A. This was the site where the intervention itself took place. Again, the rapid fall off of recurrence is notable. Adjacent days have a recurrence rate between 35% and 42%. After three weeks, this has fallen to approximately 25%. There is, however, an exception to this. A clear effect is noted by day of the week. Recurrence rates are significantly higher when the two surveyed days are the

same days of the week. For example, the recurrence rate between 16/7/01 and 25/6/01 is 32% despite the fact that these are four weeks apart. The day of the week effect is an interesting one. It shows that there are a pool of drivers who consistently drive in rush hour only on certain days of the week. Note also that it appears weeks are a significant unit in recurrence. Recurrence rates between days in the same week are usually higher than recurrence rates between days in different weeks. This pattern is visible in all the Fishergate data. No significant reduction in recurrence rates can be seen due to the intervention in the network.

Table 5.27 shows the effect of changing how recurrence rates are calculated. In this case, traffic seen between 8:20am and 8:40am is matched against traffic seen at any point in the survey (at this site, this is between 7:45am and 9:15am). As can be seen, in almost all cases, this increases the recurrence rate greatly as might be expected (the recurrence rates also fluctuate more, probably because the sample size has been reduced). However, the recurrence rates still remain below 50% in all but one case and below 40% in the majority of cases. More than half the vehicles in the rush hour will not travel during the rush hour on any given follow up survey day at that site.

Table 5.28 shows the recurrence rates at site E. Site E is a radial entry to the city. Interestingly, the recurrence rates seem to be higher at this site. This is possible due to the fact that site E is harder to reroute around. Table 5.29 shows the recurrence rates at site K. The recurrence rates here are in between those of site E and site A.

The data suggests the following GLM

$$\mathrm{E}\left[R\right] = \beta_0 + \beta_1|d| + \beta_2 I_w + \beta_3 I_d,$$

where R is the percentage recurrence rate, β_i are the parameters of the model, d is the difference in days between the two survey days, I_w is an indicator variable which is one if the two days are in a different week and I_d is an indicator variable which is one if the two days are on the same day of the

week. Note that the variable d omits weekends so a Monday is assumed to be only one day away from the adjacent Friday. The model was run with both assumptions and the former was found to produce a better fit in all cases tried. Same day ($d = 0$) samples are removed as meaningless. To prevent double counting, only pairs of sites in the upper diagonal of the table are counted — that is, if the site pair (i, j) was included, the site pair (j, i) was omitted. The modelling was only carried out for sites A and E since site K had incomplete data.

The table below shows the parameters of the model fitted at site A. All parameters have the expected signs and are significant at the the 1% level or better.

Parameter	β_0	β_1	β_2	β_3		
Estimate	36.71	-0.54	-2.92	4.73		
Std. Error	0.63	0.10	0.89	0.89		
Significance	0.l%	0.1%	1%	0.1%		
Statistic	R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate	0.551	0.536	35.6	3	87	$< 4.1 \times 10^{-15}$

The table below shows the parameters of the model fitted at site E. All parameters have the expected signs and are significant at the the 0.1% level.

Parameter	β_0	β_1	β_2	β_3		
Estimate	43.08	-0.62	-4.94	3.78		
Std. Error	0.53	0.08	0.75	0.75		
Significance	0.1%	0.1%	0.1%	0.1%		
Statistic	R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate	0.735	0.726	80.52	3	87	$< 2.2 \times 10^{-16}$

The low p-value, and relatively high R^2 and R_a^2 values suggest that this is a good model for recurrence rates. The model shows that, for these two sites, the recurrence rates decay by approximately 0.5% every day and by approximately 4% if the other surveyed day is in a different week. However, surveys which are on the same day of the week have a recurrence rate approximately 4% higher.

A number of alterations to this model were considered but none produced a sufficiently significant increase in the R_a^2 value to warrant inclusion of extra parameters. The assumption that the recurrence rate falls off linearly with $|d|$ is clearly false in the longer term. However, other variants such as $1/|d|$ and $e^{-|d|}$ produced a worse fit to the model. Whether the days surveyed were the same type of day as regards the closure or otherwise of Fishergate made no significant difference to the model and adding this to the model made no difference to the results.

The amount of data collected is not sufficient to model days of the week separately (that is, to try to separate the effect that both days were a Monday, both were a Tuesday and so on). Only three surveys were made for each day of the week except Tuesday which only had two surveys. However, it is interesting to report that this model had a similar R_a^2 value (0.567 at site A and 0.72 at site E) although not all of its parameters were statistically significant (this is unsurprising given the low number of samples). It is tempting to pool the data for all sites to increase the number of samples but this would be problematic since the recurrence rate between two days at site A is clearly not independent from the recurrence rate between the same two days at site B. It is important to remember that the data was collected to investigate an intervention therefore these results may not quite be typical.

	27/6/00	28/6/00	6/9/00	7/9/00	8/9/00
27/6/00	107.6 (99.7)	47.3 (39.0)	24.3 (17.8)	24.6 (18.1)	25.2 (18.8)
28/6/00	44.5 (36.7)	109.8 (101.5)	25.3 (18.7)	25.1 (18.6)	22.0 (15.6)
6/9/00	29.1 (21.3)	32.2 (23.9)	105.1 (98.6)	45.4 (38.9)	39.9 (33.6)
7/9/00	29.6 (21.8)	32.1 (23.7)	45.6 (39.0)	107.9 (101.4)	43.2 (36.9)
8/9/00	31.0 (23.2)	28.8 (20.5)	41.1 (34.5)	44.3 (37.8)	105.5 (99.2)
11/9/00	29.7 (21.9)	29.7 (21.4)	37.4 (30.8)	37.9 (31.4)	37.2 (30.9)
13/9/00	26.8 (19.0)	30.6 (22.3)	34.0 (27.5)	33.6 (27.1)	34.3 (27.9)
27/9/00	31.4 (23.6)	32.7 (24.4)	34.8 (28.2)	32.5 (26.0)	32.4 (26.0)
18/10/00	25.9 (18.1)	30.7 (22.3)	29.1 (22.6)	30.9 (24.4)	29.4 (23.0)
	11/9/00	13/9/00	27/9/00	18/10/00	
27/6/00	25.5 (18.8)	20.2 (14.3)	26.8 (20.1)	22.2 (15.5)	
28/6/00	23.9 (17.2)	21.7 (15.8)	26.2 (19.6)	24.7 (18.0)	
6/9/00	38.3 (31.6)	30.7 (24.8)	35.5 (28.8)	29.9 (23.2)	
7/9/00	39.1 (32.4)	30.5 (24.6)	33.3 (26.6)	31.8 (25.1)	
8/9/00	39.3 (32.6)	31.9 (26.0)	34.0 (27.3)	31.0 (24.3)	
11/9/00	108.4 (101.6)	36.0 (30.1)	37.5 (30.8)	30.3 (23.6)	
13/9/00	40.9 (34.2)	110.0 (104.1)	34.6 (27.9)	32.8 (26.1)	
27/9/00	37.6 (30.9)	30.6 (24.7)	106.7 (100.0)	39.2 (32.5)	
18/10/00	30.3 (23.6)	28.9 (23.0)	39.0 (32.3)	104.1 (97.4)	

TABLE 5.25. Matches between days for site L in the Lendal Bridge survey (8:00 – 9:00).

	25/6/01	26/6/01	27/6/01	28/6/01	29/6/01
25/6/01	116.8 (105.5)	46.1 (35.4)	45.5 (34.5)	46.1 (34.7)	43.4 (32.3)
26/6/01	48.4 (37.1)	114.9 (104.2)	52.1 (41.1)	53.4 (42.0)	44.3 (33.2)
27/6/01	46.7 (35.5)	51.1 (40.3)	118.6 (107.6)	52.8 (41.4)	45.7 (34.6)
28/6/01	45.5 (34.2)	50.3 (39.5)	50.8 (39.8)	118.3 (106.9)	47.1 (35.9)
29/6/01	43.8 (32.5)	42.7 (31.9)	44.9 (33.9)	48.0 (36.6)	115.9 (104.8)
2/7/01	48.0 (36.7)	44.0 (33.3)	44.0 (33.0)	45.9 (34.6)	43.3 (32.2)
3/7/01	44.1 (32.9)	49.9 (39.1)	46.6 (35.6)	49.1 (37.7)	43.3 (32.2)
4/7/01	40.8 (29.6)	42.5 (31.8)	49.7 (38.7)	46.5 (35.1)	43.3 (32.1)
5/7/01	42.7 (31.4)	43.1 (32.4)	44.0 (33.0)	51.1 (39.7)	40.1 (29.0)
6/7/01	38.5 (27.2)	42.1 (31.3)	41.0 (30.1)	43.6 (32.2)	44.5 (33.4)
11/7/01	39.1 (27.9)	40.6 (29.8)	46.9 (35.9)	43.4 (32.0)	37.7 (26.5)
12/7/01	37.5 (26.3)	41.1 (30.4)	41.3 (30.4)	44.7 (33.3)	38.0 (26.8)
13/7/01	35.2 (24.0)	35.3 (24.6)	37.0 (26.0)	36.7 (25.4)	40.9 (29.7)
16/7/01	43.2 (32.0)	38.8 (28.1)	41.7 (30.8)	41.0 (29.6)	37.4 (26.3)
	2/7/01	3/7/01	4/7/01	5/7/01	6/7/01
25/6/01	46.7 (35.7)	35.5 (26.4)	34.3 (24.8)	34.7 (25.5)	32.1 (22.7)
26/6/01	44.9 (34.0)	42.0 (33.0)	37.4 (28.0)	36.7 (27.6)	36.7 (27.4)
27/6/01	44.0 (33.0)	38.5 (29.4)	42.8 (33.4)	36.7 (27.6)	35.1 (25.8)
28/6/01	44.2 (33.2)	39.0 (29.9)	38.6 (29.1)	41.1 (31.9)	35.9 (26.5)
29/6/01	42.5 (31.6)	35.1 (26.1)	36.6 (27.2)	32.9 (23.8)	37.4 (28.0)
2/7/01	112.6 (101.6)	44.5 (35.4)	42.6 (33.1)	39.7 (30.5)	37.7 (28.3)
3/7/01	53.8 (42.9)	109.9 (100.9)	49.3 (39.9)	43.5 (34.3)	42.3 (32.9)
4/7/01	49.4 (38.5)	47.3 (38.2)	109.5 (100.1)	47.9 (38.7)	43.1 (33.7)
5/7/01	47.5 (36.5)	43.0 (34.0)	49.4 (39.9)	108.9 (99.8)	44.3 (34.9)
6/7/01	44.0 (33.1)	40.8 (31.7)	43.3 (33.9)	43.3 (34.1)	108.9 (99.5)
11/7/01	42.3 (31.3)	39.4 (30.3)	43.8 (34.4)	40.5 (31.3)	39.6 (30.2)
12/7/01	40.9 (29.9)	38.8 (29.8)	40.4 (31.0)	42.3 (33.1)	38.2 (28.8)
13/7/01	38.4 (27.5)	33.7 (24.7)	33.7 (24.3)	34.0 (24.8)	37.9 (28.5)
16/7/01	45.5 (34.6)	37.2 (28.1)	37.5 (28.1)	37.1 (28.0)	37.2 (27.8)
	11/7/01	12/7/01	13/7/01	16/7/01	
25/6/01	30.8 (21.9)	28.9 (20.3)	33.3 (22.6)	37.2 (27.5)	
26/6/01	33.5 (24.6)	33.2 (24.6)	34.9 (24.3)	35.0 (25.3)	
27/6/01	37.9 (29.1)	32.7 (24.1)	35.9 (25.2)	36.9 (27.2)	
28/6/01	33.7 (24.9)	34.1 (25.4)	34.3 (23.6)	34.8 (25.2)	
29/6/01	29.9 (21.0)	29.5 (20.9)	38.9 (28.3)	32.5 (22.8)	
2/7/01	34.2 (25.3)	32.4 (23.7)	37.3 (26.6)	40.2 (30.6)	
3/7/01	38.5 (29.7)	37.2 (28.6)	39.6 (29.0)	39.8 (30.1)	
4/7/01	41.1 (32.2)	37.1 (28.4)	38.0 (27.3)	38.5 (28.8)	
5/7/01	39.2 (30.3)	40.0 (31.4)	39.4 (28.8)	39.2 (29.6)	
6/7/01	37.4 (28.5)	35.3 (26.7)	42.9 (32.3)	38.4 (28.7)	
11/7/01	109.0 (100.2)	47.2 (38.6)	43.7 (33.1)	43.6 (34.0)	
12/7/01	48.2 (39.4)	108.7 (100.0)	46.7 (36.1)	45.7 (36.0)	
13/7/01	36.4 (27.6)	38.1 (29.4)	115.4 (104.7)	39.3 (29.6)	
16/7/01	39.9 (31.1)	40.9 (32.2)	43.1 (32.5)	110.1 (100.4)	

TABLE 5.26. Matches between days for site A in the Fishergate survey (8:00 – 9:00).

	25/6/01	26/6/01	27/6/01	28/6/01	29/6/01
25/6/01	123.7 (107.7)	57.5 (42.2)	55.8 (40.3)	55.4 (39.1)	52.0 (36.2)
26/6/01	54.7 (38.8)	116.0 (100.7)	61.0 (45.5)	62.6 (46.3)	51.4 (35.5)
27/6/01	54.9 (38.9)	55.3 (40.0)	116.6 (101.1)	64.2 (47.9)	54.2 (38.4)
28/6/01	56.3 (40.3)	59.2 (43.8)	61.0 (45.5)	130.5 (114.2)	52.3 (36.4)
29/6/01	54.1 (38.1)	48.6 (33.2)	53.9 (38.4)	57.8 (41.5)	115.2 (99.4)
2/7/01	56.2 (40.2)	55.1 (39.8)	54.0 (38.5)	57.7 (41.4)	52.7 (36.9)
3/7/01	50.6 (34.7)	61.0 (45.7)	55.9 (40.5)	59.0 (42.7)	51.9 (36.1)
4/7/01	50.0 (34.0)	54.7 (39.4)	56.2 (40.7)	58.9 (42.6)	48.5 (32.7)
5/7/01	50.4 (34.4)	49.6 (34.3)	50.4 (34.9)	61.0 (44.7)	46.8 (30.9)
6/7/01	44.5 (28.5)	51.4 (36.1)	50.9 (35.4)	52.4 (36.1)	52.7 (36.9)
11/7/01	44.1 (28.2)	49.7 (34.4)	54.5 (39.0)	56.9 (40.6)	46.0 (30.2)
12/7/01	43.9 (27.9)	46.4 (31.0)	48.6 (33.1)	52.8 (36.5)	42.7 (26.9)
13/7/01	42.7 (26.8)	39.6 (24.3)	46.3 (30.8)	46.7 (30.4)	45.4 (29.5)
16/7/01	50.4 (34.4)	44.4 (29.1)	46.9 (31.4)	48.9 (32.6)	41.4 (25.6)
	2/7/01	3/7/01	4/7/01	5/7/01	6/7/01
25/6/01	58.1 (42.2)	46.9 (33.6)	44.4 (30.8)	46.9 (33.8)	45.5 (32.1)
26/6/01	54.7 (38.8)	51.4 (38.0)	44.8 (31.3)	44.1 (31.0)	43.5 (30.1)
27/6/01	54.2 (38.3)	48.3 (34.9)	49.7 (36.1)	45.6 (32.5)	43.8 (30.4)
28/6/01	51.5 (35.6)	49.5 (36.2)	47.4 (33.8)	51.7 (38.6)	44.3 (30.9)
29/6/01	51.4 (35.5)	45.3 (31.9)	44.2 (30.6)	40.2 (27.1)	45.5 (32.1)
2/7/01	117.6 (101.6)	56.4 (43.0)	52.9 (39.4)	45.3 (32.2)	47.3 (34.0)
3/7/01	67.6 (51.7)	114.4 (101.0)	59.7 (46.2)	52.4 (39.3)	51.6 (38.3)
4/7/01	57.1 (41.2)	56.2 (42.8)	112.3 (98.7)	55.9 (42.8)	52.2 (38.9)
5/7/01	58.7 (42.8)	55.3 (41.9)	62.3 (48.8)	113.8 (100.7)	52.2 (38.9)
6/7/01	51.9 (36.0)	54.5 (41.1)	50.4 (36.8)	55.8 (42.7)	113.4 (100.0)
11/7/01	47.3 (31.4)	45.5 (32.1)	55.3 (41.8)	48.9 (35.8)	47.6 (34.3)
12/7/01	47.2 (31.3)	47.5 (34.1)	51.1 (37.5)	51.7 (38.6)	46.9 (33.6)
13/7/01	44.5 (28.5)	41.6 (28.2)	39.0 (25.4)	43.6 (30.5)	46.9 (33.6)
16/7/01	53.9 (37.9)	44.1 (30.8)	43.4 (29.8)	43.6 (30.5)	42.4 (29.1)
	11/7/01	12/7/01	13/7/01	16/7/01	
25/6/01	38.1 (24.9)	39.5 (26.7)	43.6 (28.4)	48.2 (34.3)	
26/6/01	43.0 (29.9)	37.6 (24.8)	39.2 (24.1)	45.3 (31.4)	
27/6/01	48.1 (35.0)	41.7 (28.9)	43.1 (28.0)	45.4 (31.5)	
28/6/01	44.8 (31.7)	46.6 (33.8)	43.6 (28.4)	44.8 (31.0)	
29/6/01	39.3 (26.2)	39.7 (26.9)	46.8 (31.7)	42.8 (29.0)	
2/7/01	44.7 (31.6)	38.6 (25.8)	46.6 (31.5)	51.2 (37.3)	
3/7/01	48.9 (35.7)	44.6 (31.7)	46.3 (31.2)	47.8 (34.0)	
4/7/01	48.5 (35.4)	44.6 (31.7)	43.8 (28.7)	44.3 (30.5)	
5/7/01	48.1 (34.9)	50.4 (37.5)	45.2 (30.1)	51.2 (37.3)	
6/7/01	46.5 (33.4)	46.3 (33.4)	51.7 (36.6)	48.6 (34.7)	
11/7/01	116.8 (103.6)	59.8 (47.0)	52.1 (37.0)	53.5 (39.6)	
12/7/01	57.3 (44.2)	110.6 (97.8)	52.2 (37.1)	53.6 (39.8)	
13/7/01	47.8 (34.7)	45.2 (32.3)	117.4 (102.3)	45.8 (32.0)	
16/7/01	46.9 (33.8)	47.4 (34.5)	51.1 (36.0)	112.5 (98.6)	

TABLE 5.27. Matches between days for site A in the Fishergate survey (8:20 – 8:40).

	25/6/01	26/6/01	27/6/01	28/6/01	29/6/01
25/6/01	107.7 (100.6)	53.7 (46.7)	49.0 (42.4)	49.8 (42.4)	50.1 (42.4)
26/6/01	54.6 (47.5)	105.7 (98.7)	51.5 (44.9)	53.2 (45.9)	48.9 (41.2)
27/6/01	53.2 (46.1)	55.0 (48.0)	107.8 (101.3)	54.3 (46.9)	51.2 (43.6)
28/6/01	48.2 (41.1)	50.7 (43.7)	48.4 (41.9)	108.9 (101.6)	52.1 (44.5)
29/6/01	46.5 (39.3)	44.6 (37.6)	43.7 (37.2)	49.9 (42.6)	107.3 (99.6)
2/7/01	47.0 (39.9)	44.3 (37.3)	41.9 (35.4)	44.7 (37.3)	43.0 (35.3)
3/7/01	45.1 (38.0)	46.5 (39.6)	42.5 (36.0)	47.3 (40.0)	44.4 (36.7)
4/7/01	42.1 (35.0)	40.3 (33.3)	43.3 (36.8)	43.8 (36.5)	41.5 (33.9)
5/7/01	42.2 (35.1)	41.1 (34.1)	37.8 (31.2)	44.0 (36.6)	42.9 (35.2)
6/7/01	38.3 (31.2)	38.7 (31.7)	37.0 (30.5)	38.0 (30.6)	44.8 (37.1)
11/7/01	38.9 (31.8)	39.0 (32.0)	41.1 (34.5)	40.7 (33.3)	38.1 (30.5)
12/7/01	38.2 (31.1)	39.8 (32.8)	35.2 (28.6)	41.6 (34.3)	38.0 (30.3)
13/7/01	37.2 (30.1)	37.2 (30.2)	34.7 (28.2)	37.3 (29.9)	42.3 (34.6)
16/7/01	42.3 (35.2)	38.0 (31.0)	34.4 (27.9)	37.7 (30.3)	38.7 (31.0)
	2/7/01	3/7/01	4/7/01	5/7/01	6/7/01
25/6/01	54.2 (46.0)	47.1 (39.7)	47.5 (39.5)	47.6 (39.6)	44.3 (36.0)
26/6/01	51.8 (43.6)	49.3 (41.9)	46.2 (38.2)	47.1 (39.1)	45.4 (37.2)
27/6/01	52.4 (44.3)	48.2 (40.8)	53.1 (45.0)	46.3 (38.3)	46.5 (38.2)
28/6/01	49.8 (41.7)	47.8 (40.4)	47.9 (39.9)	48.1 (40.1)	42.5 (34.3)
29/6/01	45.9 (37.7)	42.9 (35.5)	43.4 (35.4)	44.9 (36.9)	48.0 (39.8)
2/7/01	107.6 (99.4)	53.4 (46.0)	51.1 (43.1)	48.1 (40.1)	46.3 (38.1)
3/7/01	59.0 (50.8)	109.9 (102.5)	56.0 (48.0)	51.8 (43.8)	49.6 (41.4)
4/7/01	52.2 (44.0)	51.8 (44.4)	108.2 (100.2)	52.0 (44.0)	47.8 (39.6)
5/7/01	49.1 (40.9)	47.9 (40.4)	51.9 (43.9)	107.0 (99.0)	50.3 (42.1)
6/7/01	46.2 (38.0)	44.7 (37.3)	46.6 (38.6)	49.2 (41.1)	106.4 (98.2)
11/7/01	46.1 (38.0)	44.1 (36.7)	51.3 (43.3)	44.6 (36.5)	43.9 (35.7)
12/7/01	46.8 (38.7)	43.4 (36.0)	43.5 (35.5)	46.7 (38.7)	43.1 (34.9)
13/7/01	43.1 (35.0)	39.4 (32.0)	42.1 (34.1)	42.9 (34.8)	44.9 (36.7)
16/7/01	46.7 (38.6)	39.5 (32.0)	40.7 (32.7)	41.3 (33.3)	41.2 (32.9)
	11/7/01	12/7/01	13/7/01	16/7/01	
25/6/01	41.4 (33.9)	42.8 (34.8)	43.4 (35.1)	50.3 (41.9)	
26/6/01	42.2 (34.6)	45.3 (37.3)	44.1 (35.8)	46.0 (37.5)	
27/6/01	47.6 (40.0)	42.8 (34.8)	44.0 (35.7)	44.5 (36.0)	
28/6/01	42.0 (34.4)	45.1 (37.2)	42.1 (33.8)	43.4 (34.9)	
29/6/01	37.7 (30.1)	39.5 (31.5)	45.7 (37.4)	42.7 (34.2)	
2/7/01	42.7 (35.1)	45.6 (37.6)	43.7 (35.4)	48.3 (39.8)	
3/7/01	45.1 (37.6)	46.7 (38.7)	44.1 (35.8)	45.0 (36.6)	
4/7/01	48.5 (40.9)	43.2 (35.3)	43.5 (35.2)	42.9 (34.5)	
5/7/01	42.1 (34.5)	46.4 (38.4)	44.3 (36.0)	43.6 (35.1)	
6/7/01	40.5 (32.9)	41.8 (33.8)	45.3 (37.0)	42.4 (33.9)	
11/7/01	107.0 (99.4)	51.6 (43.7)	49.5 (41.2)	48.0 (39.6)	
12/7/01	49.1 (41.6)	109.0 (101.1)	49.7 (41.5)	47.6 (39.2)	
13/7/01	45.3 (37.7)	47.8 (39.8)	107.5 (99.2)	46.1 (37.7)	
16/7/01	43.0 (35.5)	44.8 (36.9)	45.2 (36.9)	109.5 (101.0)	

TABLE 5.28. Matches between days for site E in the Fishergate survey (8:00 – 9:00).

	25/6/01	26/6/01	27/6/01	28/6/01	29/6/01
25/6/01	103.5 (100.7)	42.3 (39.4)	35.4 (32.8)	35.9 (32.9)	38.0 (34.6)
26/6/01	39.5 (36.7)	103.8 (100.8)	41.4 (38.8)	42.4 (39.4)	37.6 (34.2)
27/6/01	37.8 (35.0)	47.4 (44.4)	104.3 (101.8)	40.9 (37.8)	44.9 (41.5)
28/6/01	32.3 (29.5)	40.9 (37.9)	34.4 (31.8)	105.2 (102.1)	38.5 (35.2)
29/6/01	31.3 (28.6)	33.3 (30.3)	34.7 (32.1)	35.4 (32.3)	103.3 (100.0)
2/7/01	34.8 (32.0)	36.5 (33.5)	30.8 (28.2)	35.2 (32.2)	39.2 (35.9)
3/7/01	30.4 (27.6)	35.5 (32.5)	27.6 (25.0)	31.2 (28.1)	33.8 (30.4)
4/7/01	28.0 (25.2)	31.8 (28.9)	32.5 (29.9)	31.2 (28.1)	34.0 (30.6)
5/7/01	28.8 (26.0)	30.2 (27.2)	25.2 (22.6)	30.8 (27.7)	30.6 (27.2)
6/7/01	27.8 (25.0)	30.3 (27.3)	26.3 (23.8)	27.6 (24.5)	34.9 (31.5)
11/7/01	—	—	—	—	—
12/7/01	22.4 (19.6)	27.0 (24.0)	21.7 (19.2)	26.3 (23.2)	30.2 (26.9)
13/7/01	25.8 (23.1)	27.2 (24.2)	22.9 (20.4)	27.4 (24.3)	34.7 (31.4)
16/7/01	27.4 (24.6)	27.4 (24.4)	21.9 (19.3)	26.7 (23.7)	28.6 (25.3)
	2/7/01	3/7/01	4/7/01	5/7/01	6/7/01
25/6/01	40.3 (37.1)	41.4 (37.7)	38.0 (34.2)	41.4 (37.5)	38.8 (35.0)
26/6/01	39.5 (36.3)	45.1 (41.4)	40.3 (36.5)	40.5 (36.6)	39.5 (35.6)
27/6/01	38.1 (34.9)	40.2 (36.5)	47.1 (43.3)	38.7 (34.7)	39.3 (35.5)
28/6/01	36.7 (33.5)	38.3 (34.5)	38.0 (34.3)	39.8 (35.9)	34.6 (30.8)
29/6/01	37.6 (34.4)	38.0 (34.3)	38.0 (34.3)	36.4 (32.4)	40.2 (36.3)
2/7/01	105.0 (101.8)	43.2 (39.5)	43.5 (39.8)	39.2 (35.3)	37.8 (33.9)
3/7/01	36.7 (33.5)	103.4 (99.6)	43.5 (39.8)	44.6 (40.6)	38.4 (34.6)
4/7/01	37.2 (34.0)	43.8 (40.0)	105.6 (101.8)	45.1 (41.1)	38.0 (34.2)
5/7/01	31.6 (28.4)	42.3 (38.5)	42.5 (38.7)	106.8 (102.9)	43.5 (39.6)
6/7/01	31.3 (28.1)	37.6 (33.8)	36.9 (33.2)	44.8 (40.8)	104.6 (100.7)
11/7/01	—	—	—	—	—
12/7/01	28.5 (25.3)	34.1 (30.4)	34.6 (30.8)	39.6 (35.6)	33.5 (29.6)
13/7/01	29.0 (25.8)	30.3 (26.5)	31.6 (27.9)	33.9 (29.9)	35.0 (31.1)
16/7/01	31.6 (28.4)	32.2 (28.4)	31.4 (27.6)	30.3 (26.3)	30.7 (26.9)
	11/7/01	12/7/01	13/7/01	16/7/01	
25/6/01	—	29.9 (26.2)	33.6 (30.0)	37.7 (33.9)	
26/6/01	—	33.5 (29.8)	33.0 (29.4)	35.1 (31.3)	
27/6/01	—	31.0 (27.3)	31.9 (28.3)	32.2 (28.4)	
28/6/01	—	31.5 (27.8)	32.0 (28.4)	33.1 (29.3)	
29/6/01	—	33.3 (29.6)	37.3 (33.7)	32.5 (28.7)	
2/7/01	—	32.8 (29.1)	32.5 (28.9)	37.5 (33.7)	
3/7/01	—	33.3 (29.7)	28.9 (25.3)	32.5 (28.7)	
4/7/01	—	34.0 (30.3)	30.3 (26.7)	31.8 (28.0)	
5/7/01	—	36.6 (32.9)	30.6 (27.0)	29.0 (25.2)	
6/7/01	—	32.0 (28.3)	32.6 (29.0)	30.3 (26.5)	
11/7/01	—	—	—	—	
12/7/01	—	105.7 (102.0)	34.6 (31.0)	33.7 (29.9)	
13/7/01	—	35.4 (31.7)	104.9 (101.3)	33.2 (29.4)	
16/7/01	—	32.6 (29.0)	31.4 (27.8)	104.6 (100.8)	

TABLE 5.29. Matches between days for site K in the Fishergate survey (8:00 – 9:00).

5.9. Multiple Site Matching

Using the methods of the previous chapter, matches across more than two sites can be examined. However, the difficulties of investigating the data in this context are large. Consider, the issue of finding drivers who swap routes. The number of drivers changing route to any given new route is unlikely to be large. For example, in the Fishergate data, if 20% of the drivers travelling from E–A rerouted to E–K and E–F this would only be approximately fifty drivers split between those two routes. For perspective, data for the triple match E–A–K was examined. While E–A and E–K are likely pairs, E–A–K is an unlikely triple, it would only be taken by a driver who was lost, took a wrong turning or had a specific reason to make the diversion. The expected number of drivers seen at these three routes would be zero or in single figures. However, the mean number of matches seen across all three sites in the thirteen days where data was available was 82.8 with a standard deviation of 23.4. In short, the noise in the experiment is almost certainly as large or larger than the effect which is to be measured. All results on multiple site matching should be viewed with this considerable caveat in mind.

The original hypothesis was that there was a rerouting of vehicles from E–A to E–K or E–F. The analysis performed, therefore, is to look at data from the four point match E–A on day one versus E–K or E–F on day two (where day one is not equal to day two). The corrected matches for these two experiments are presented in Tables 5.30 and 5.31. The tables should be interpreted as follows: the figure in the column 26/6/01 and the row 25/6/01 represents an estimate of the number of drivers who were seen using E–A on day one and E–F on day two. Any negatives should be considered as over correction of false matches. Note that the table includes also results where day one is after day two (those results below and to the left of the diagonal).

The data from the tables is hard to interpret directly. It is immediately clear that the error in the correction process is extremely large. This is unfortunate but inevitable. To mitigate this problem, an attempt was made to

fit a GLM to the data. Three explanatory variables suggest themselves immediately. Whether the first day considered is a "before" day, whether the second day considered is an after day and whether both of these conditions apply simultaneously. The latter condition is the most interesting since that is the effect being searched for (drivers who are on route E–A in the before but switch to one of the alternatives when the closure is in place).

The indicator variables appropriate can be designated by the following system: I_{bb} (day one and day two are both before), I_{dd} (day one and day two are both during), $I_{d\cdot}$ (day one is before day two is unspecified), $I_{\cdot d}$ (day one is unspecified, day two is during) and any one of the five other variants.

For the data from E–A switching to E–K then two models seem to have reasonable predictive power. The first model is

$$\mathrm{E}[f] = \beta_0 + \beta_1 I_{bb} + \beta_2 I_{bd},$$

where $\mathrm{E}[f]$ is the expected value of the switching flow and β_i are the parameters of the model. The parameters of the fitted model are given by the following table:

Parameter	β_0	β_1	β_2			
Estimate	-1.74	13.33	12.14			
Significance	low	1%	0.1%			
Statistic	R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate	0.173	0.152	8.156	2	78	0.00061

While this model might seem successful it is not as good as the simpler model given by

$$\mathrm{E}[f] = \beta_0 + \beta_1 I_{b\cdot},$$

which when fitted gives the results:

Parameter	β_0	β_1				
Estimate	-5.17	16.05				
Significance	10%	0.1%				
Statistic	R^2	R_a^2	F	ν_1	ν_2	p-value
Estimate	0.254	0.244	26.9	1	79	1.61×10^{-6}

This model says that approximately 16 more drivers switch route from E–A to E–K if the first day is a before day than would be the case otherwise. No other models were found which were a closer fit to the data than this. This result is somewhat curious since it implies that more switching took place when the first day was in the before scenario regardless of when the second day was. For the data from E–A to E–F the situation was even worse. No models of this type with significant parameters were found. The conclusion, therefore is that this data cannot answer even simple questions which directly concern rerouting. Table 5.32 shows the number of vehicles which are estimated to be present at every day of weeks one and two at all the sites of the Fishergate survey. As can be seem the number of spurious matches in the five point data is extremely high (an extreme case being at site A where there are more than one thousand estimated false matches). This is due to the discussed combinatorial nature of the problem. It is hard to know to what extent these data can be trusted. They are presented here without comment except to say that an important priority with the work presented in the previous chapter is to find a way to assign confidence limits to the estimates it gives.

	25/6/01	26/6/01	27/6/01	28/6/01	29/6/01	2/7/01	3/7/01
25/6/01	—	-19.2	-15.8	-9.2	-27.7	-26.5	-4.2
26/6/01	-15.0	—	-0.1	6.3	0.8	-15.6	-8.6
27/6/01	-2.0	-8.7	—	-7.5	-3.3	1.1	9.8
28/6/01	7.3	-3.5	2.9	—	7.9	-4.0	13.8
29/6/01	-5.8	-8.9	8.7	5.7	—	17.9	2.0
2/7/01	-17.9	-17.8	-6.3	-3.6	-13.6	—	6.7
3/7/01	-0.6	-9.6	-0.5	-8.8	-6.1	-7.1	—
4/7/01	-11.7	2.8	5.1	-7.3	-7.3	-13.2	1.7
5/7/01	5.3	-5.9	0.7	-10.4	-12.5	-6.0	15.0
6/7/01	-7.7	-13.3	-5.6	-8.0	-11.5	-7.7	-0.5
11/7/01	-14.1	-13.7	-12.1	-8.9	-15.5	-7.0	-11.3
12/7/01	-12.5	-9.0	-4.1	-12.7	-17.7	-7.0	2.1
13/7/01	-5.5	-0.8	-9.1	-18.6	-19.2	6.3	-2.2
16/7/01	-14.7	-5.0	-0.3	-8.1	-7.4	-14.0	-7.4
	4/7/01	5/7/01	6/7/01	11/7/01	12/7/01	13/7/01	16/7/01
25/6/01	-12.8	-5.5	-8.1	—	15.5	-5.2	7.0
26/6/01	9.2	1.4	-10.6	—	6.8	9.7	-4.1
27/6/01	3.3	4.1	1.6	—	3.0	10.4	13.1
28/6/01	26.1	56.8	-2.9	—	29.0	14.5	26.0
29/6/01	17.3	25.7	9.3	—	12.5	5.9	33.0
2/7/01	-5.8	11.3	-5.6	—	23.3	0.8	-5.0
3/7/01	7.7	-0.4	-0.1	—	25.5	11.6	4.4
4/7/01	—	2.3	-0.5	—	13.9	16.2	6.4
5/7/01	23.6	—	4.3	—	22.8	10.3	1.4
6/7/01	-18.1	-4.5	—	—	13.1	2.6	16.5
11/7/01	3.1	-2.9	-4.4	—	-7.8	-5.8	16.3
12/7/01	16.3	19.2	3.7	—	—	17.7	28.9
13/7/01	-4.2	12.5	0.2	—	16.4	—	53.6
16/7/01	8.7	38.9	7.7	—	16.5	1.4	—

TABLE 5.30. Matches for vehicles switching from sites E–A to E–K across days for the Fishergate survey.

	25/6/01	26/6/01	27/6/01	28/6/01	29/6/01	2/7/01	3/7/01
25/6/01	—	-5.2	0.1	0.4	-3.2	-11.2	13.8
26/6/01	-8.8	—	-4.3	1.5	0.6	-6.6	19.3
27/6/01	0.4	-7.2	—	-5.6	-9.9	-7.0	-4.6
28/6/01	-9.4	0.3	-9.3	—	0.9	-5.9	9.6
29/6/01	-8.8	-12.0	-10.7	-3.8	—	-15.4	-8.1
2/7/01	-6.3	3.5	-2.3	2.2	-10.7	—	2.9
3/7/01	2.1	4.5	2.2	4.3	-1.7	-5.4	—
4/7/01	-3.4	1.5	-3.1	2.1	0.6	-11.8	15.5
5/7/01	18.8	1.1	9.5	0.8	9.0	-3.9	12.5
6/7/01	7.8	2.7	2.2	0.7	1.1	-4.7	14.0
11/7/01	1.7	-4.9	0.7	-4.8	-1.5	-7.1	5.2
12/7/01	2.7	-0.9	-4.5	-2.8	0.6	-6.7	3.8
13/7/01	5.1	-6.9	-1.6	-0.9	1.5	-8.1	6.6
16/7/01	-0.6	-5.6	0.6	-4.9	-1.5	3.8	5.2

	4/7/01	5/7/01	6/7/01	11/7/01	12/7/01	13/7/01	16/7/01
25/6/01	0.8	5.3	4.0	-2.7	-4.2	15.0	-4.7
26/6/01	5.0	-5.7	0.7	-10.9	-3.5	0.8	0.7
27/6/01	8.5	-5.8	-0.4	6.4	-4.8	1.5	0.9
28/6/01	13.0	31.7	1.5	-10.7	-1.5	4.8	10.5
29/6/01	3.6	10.1	-4.4	-12.6	-2.3	13.8	12.3
2/7/01	-9.9	-5.5	0.3	-6.2	6.8	-6.1	0.6
3/7/01	-0.7	-1.8	1.9	7.4	8.0	1.7	4.8
4/7/01	—	-3.8	10.5	2.7	6.2	0.2	0.4
5/7/01	11.9	—	1.1	-7.4	7.4	14.3	6.1
6/7/01	-1.3	0.5	—	-3.3	5.8	8.4	0.4
11/7/01	-3.9	-7.5	-2.4	—	-3.5	1.6	-2.0
12/7/01	-0.8	-3.5	-2.3	-2.0	—	10.7	9.1
13/7/01	-0.3	4.5	3.5	-4.3	6.2	—	17.7
16/7/01	6.0	-0.1	2.9	-9.2	-3.4	2.7	—

TABLE 5.31. Matches for vehicles switching from sites E–A to E–F across days for the Fishergate survey.

Site	Week One	Week Two
A	1302.0 (260.7)	921.0 (213.8)
B	—	5.0 (4.6)
C	277.0 (94.1)	211.0 (68.1)
D	—	251.0 (83.6)
E	856.0 (244.8)	1134.0 (403.5)
F	44.0 (22.1)	54.0 (21.5)
G	437.0 (166.7)	—
H	—	60.0 (18.4)
I	131.0 (60.3)	148.0 (63.7)
J	—	74.0 (51.2)
K	131.0 (75.8)	179.0 (90.6)

TABLE 5.32. Vehicles seen on all surveyed days in given weeks for the Fishergate studies — corrected estimate in brackets.

5.10. Discussion of Results

In this chapter a considerable amount of analysis has been performed on the data collected in the two surveys. The most important results are summarised in this section. It is clear that this data set, while problematic to analyse, is a rich source of information and could be potentially extremely useful to anyone interested in investigating the traffic effects of interventions.

Time plots revealed evidence about travel time during York's rush hour (Figures C.1 to C.18). Analysis of the Fishergate data showed that the travel time between sites E–A tended to rise as the rush hour went on and then fall at the end of the rush hour. This is interesting as histograms of flows reveal only a slight reduction in flow throughout the duration of the rush hour (Figures D.19 to D.39) — this seems, perhaps, slightly at odds with the fall off in travel time seen in the time plots. On the first day of closure the travel time could be clearly seen to rise continually throughout the rush hour (Figure C.15). As

the survey continued the effects of the closure appeared to lessen. Statistical tests showed no clear effects on flow over all the survey sites in either survey except in the case of the fuel crisis days for Lendal Bridge survey where a reduction in flow levels was shown as might be expected.

In the case of the Fishergate data, it was hypothesised that sites A, C and D should show reductions in flows when the closure was in place and sites F, G and K were potentially diversions and might be expected to show an increase in flow during the closure. This model proved successful in that all the parameters were significant although the R_a^2 value was low indicating that there was considerable variance in the model which was not explained by the parameters included. The reduction in flow was estimated at 6.3% on average over sites A, C and D and the increase in flow at sites F, G and K was estimated at 3.7%.

Work to estimate $p(2)$ and $p(3)$ revealed the problem that estimates from different sets of sites provided parameter estimates which differed with statistical significance. This is extremely important to the matching and work to understand why this should be so is vital to improving the performance of matching correction.

For the Fishergate survey, matches between pairs of sites were investigated to determine which pairs had the most significant flows. This was used as a factor to determine which pairs to investigate using MLE estimation techniques. Table 5.19 shows the estimated journey times for these pairs. A GLM model was fitted to these predicted times and flows. Statistical models were used at each site to estimate the effects of the closure on times of flows. The results were revealing. Site pairs leading to the closure site showed increased journey times but no observable effects on flows. Site pairs leading away from the closure site showed no significant effect on journey time but reduced flows. At the site pair showing the best fitting model of journey times (pair C–A) the alteration in journey time was shown to be returning to its base level throughout the duration of the survey with a statistical significance of 1%

on the parameter. It is interesting to note that, as with the histogram data, standard assumptions about cost-flow relations did not seem to be followed by this data. No clear relationship between cost and flow could be observed.

Several sites showed a "return to normal" type effect in the data and it seems that this data confirms the idea that initial transient responses to an intervention are damped (both for route flow and for journey times) on subsequent days. It was definitely of interest that the cost-flow relationship did not appear to work as expected in this data. Indeed no clear relationship between observed flows and travel times could be seen in the data set.

Matches between different days at the same site were performed to establish the recurrence rate of the traffic as defined in Section 5.8.3. No significant effect on the recurrence rate was shown due to the intervention on the network. However, the recurrence rate was shown to be effected by the days elapsed between the surveyed days, whether the surveyed days were in the same week and whether the surveyed days were the same day of the week. The latter effect was found to be particularly significant with the recurrence rate raised by an estimated 4% for surveys which occurred on the same day of the week. Recurrence rates were usually 50% or lower even given the most generous measure and this fell off sharply with the passage of time. After two months the recurrence rates on the Lendal Bridge survey were less than 20%. In the short term (first three weeks) a decay in the recurrence rate of 0.5% per week day was shown to be a good fit to the data.

Matching the data between multiple sites proved less successful. The high variance in the estimates produced coupled with the small effects being sought meant that the rerouting effects of the intervention could not be unequivocally established. Further work is needed either to reduce the variance in the estimates or to find some way to estimate it.

Overall, the data analysis revealed much of interest. No clear conclusions could be drawn about rerouting but insights about the transient effects of a

network intervention and about recurrence rates in surveys can be gained from the data. It is clear that more remains to be discovered in this rich data set.

CHAPTER 6

Conclusions and Further Research

In five chapters this thesis has covered various problems in internet and transport research with the unifying theme of statistical analysis of dynamic networks. The first major research area studied was the study of long-range dependence (LRD). A new model for generating streams of data exhibiting LRD was introduced and proved theoretically to generate a time series exhibiting LRD with a given mean and Hurst parameter. The model has is significant in its simplicity both computationally and analytically. When compared with other models, it is computationally simple and has only two parameters. The measurements on the model showed good agreement with theory, however, the intercept on the auto-correlation plot was incorrect. It could be that this is due to a known bias in the standard estimator of ACF and this merits further investigation.

It is hoped that the model will be a useful tool in studying the queuing properties of systems since the tractability may enable progress to be made in this area. Further research could continue in a number of directions. Firstly, it would be interesting to study a two-sided version of this model which would allow both ON and OFF periods to exhibit heavy tails. Initial investigation of this has begun. Secondly, the model assumed that ON periods are heavy tailed whereas the OFF periods are Poisson. The opposite assumption will certainly have different effects on queuing. While the Hurst parameter and mean of the traffic would remain unchanged it could well be that the queuing performance would be totally different.

In chapters three, four and five, topics related to road networks were studied. In particular, the work centered around the equilibrium concepts reviewed in chapter three. A data collection exercise was undertaken which is reported

on in chapter five. In order to fully investigate this data set, the matching framework in chapter four was developed. The matching framework in chapter four has been shown (both theoretically and experimentally) to provide an unbiased estimate of the true number of matches between a number of data sets if certain probabilities are known correctly. Several improvements to this method would be useful. While the estimator is unbiased, it can have high variance. A lower variance estimator would help and failing this an estimate of the variance would be useful. Research is actively continuing in this area. Further, it has been emphasised throughout that the matching method here is extremely general and not confined to licence plate data. Work is underway to find new data sets where this method can be applied.

Finally, in chapter five, standard statistical analysis techniques were applied to a large body of licence plate data. While the analysis showed that the uncertainties were too large to directly infer information about driver route choice, a number of interesting results arose. There was direct statistical evidence for an "effect" followed by a settling down period as a result of network intervention. Furthermore, there was also evidence of the extremely fast fall off of driver recurrence rates over a period of just a few weeks. Certainly, more analysis could be done with this data and it is likely that more will be discovered about this rich data set as more time is spent working with it.

APPENDIX A

Symbols, Functions and Notation Used in This Thesis

This chapter lists symbols and notation which are used throughout this thesis. Occasionally, symbols are used differently in different contexts (for example $\sim$ is used to mean *asymptotic to* in the context of functions but also to denote an equivalence relation in the context of sets).

A.1. General Notation Used

The following notation is used throughout the thesis. If definitions are given in the body of the thesis they are referred to here

.

- i — a positive unit imaginary number ($i^2 = -1$).
- $\#(\mathbf{X})$ — the number of elements in the set or tuple $\mathbf{X}$.
- μ — mean. See Definition 1.12.
- Ω — a sample space. See Definition 1.4.
- $\sim$ — equivalent to. See Definition 4.4. Note that this symbol is also used in a different context (see the next section).
- $\precsim$, $\succsim$, $\prec$, $\succ$, $||$, $\prec\prec$, $\succ\succ$ — preceeds, succeeds, strictly preceeds, strictly succeeds, non-comparable, immediately preceeds, immediately succeeds. These symbols are all defined for partial ordering in Definition 4.14.
- σ^2 — variance. See Definition 1.13.
- σ — standard deviation. See Definition 1.13.
- $\mathbf{B}$ — the backshift operator. See Definition B.1.
- $\mathrm{E}[X]$ — the expectation value of a random variable X. See Definition 1.9.

- i.i.d. — independent and identically distributed. Independent see Defintion 1.8. Distribution function see Definitions 1.5 and 1.6.
- M_j — the mean return time of state j in a Markov chain. See Definition 2.10.
- $\mathbb{P}[X = x]$ — the probability that a random variable X has the value x.
- π_i — the equilibrium probability of a state i in a Markov chain. See Defintion 2.11.
- S^2 — the sample variance for a data set. See Definition 1.25
- $\mathrm{var}(X)$ — variance of a variable X. (The symbol σ^2 is also used depending on context). See Definition 1.13.
- $\overline{X}$ — the sample mean of a variable X. See equation (1.2).

A.2. Asymptotic Notation

The following definitions are those used in [**71**, page 7]. Suppose there exists an integral variable n which tends to infinity and a real variable x which tends to infinity, zero or some other limiting value (unless otherwise stated these terms will be used with the assumption that $x \to \infty$). Given either $\phi(n)$ or $\phi(x)$ which is a positive function of n or x and a corresponding $f(n)$ or $f(x)$ which is any other real-valued function of n or x then:

- $f = O(\phi)$ means that $|f| < A\phi$ for some positive constant A and for all values of n or x,
- $f = o(\phi)$ means that $f/\phi \to 0$,
- $f \sim \phi$ means that $f/\phi \to 1$,
- $f \asymp \phi$ means that $A\phi < f < B\phi$ for some positive constants A and B for all values of n or x.

In addition, the notion of a *slowly varying function* will sometimes be used. A slowly varying function $L(x)$ is one where, for any $t \in \mathbb{R}$, then $L(tx) \sim L(x)$ as $x \to \infty$ (or, depending on circumstances, as $x \to 0$ — this will be made clear in context as the function is used).

A.3. Euler's Gamma Function

Euler's Gamma function $\Gamma(x)$ is a generalisation of the well-known factorial $n!$ from the domain of the natural numbers to the domain of the reals.

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt \tag{A.1}$$

It is clear that $\Gamma(1) = -e^{-\infty} + e^0 = 1$ which, in turn, is equal to $1!$. Integrating $\Gamma(x+1)$ for $x > 0$ by parts gives

$$\Gamma(x+1) = \left[-e^{-t} t^x\right]_0^\infty + x \int_0^\infty e^{-t} t^{x-1} dt = x\Gamma(x)$$

Thus, given that $\Gamma(1) = 1$, for any $n \in \mathbb{N}$ then $\Gamma(n+1) = n!$.

APPENDIX B

Basic Time Series Analysis

This appendix provides a quick introduction to a few basic time-series models: Auto-Regressive (AR), Moving Average (MA), Auto-Regressive Moving Average (ARMA), Auto-Regressive Integrated Moving Average (ARIMA) and some simple variants. These models all have in common the idea that they are processes for generating the nth point in a time series given all previous points.

The AR model is the simplest to state. An AR(1) model is given by

$$X_i = a_1 X_{i-1} + \epsilon_i, \tag{B.1}$$

where $a_1 \in (-1, 1)$ and the sequence of ϵ_i are normally distributed independent increments with zero mean and a constant variance σ_ϵ^2. An AR(2) model is the obvious extension of this.

$$X_i = a_1 X_{i-1} + a_2 X_{i-2} + \epsilon_i,$$

with $a_1, a_2 \in (-1, 1)$ and ϵ_i as before. The backshift operator is an important notational convenience here.

Definition B.1. The *backshift operator* $\mathbf{B}$ operates on an element of a time series and returns the previous element. For example $\mathbf{B}(X_n) = X_{n-1}$ and $\mathbf{B}^k(X_n) = X_{n-k}$.

Using that notation gives

$$(1 - a_1\mathbf{B} - a_2\mathbf{B}^2)X_i = \epsilon_i.$$

From this, the obvious generalisation is the AR(p) model.

$$(1 - \sum_{j=1}^{p} a_j \mathbf{B}^j)X_i = \epsilon_i, \tag{B.2}$$

with $a_j \in (-1, 1)$.

An MA(1) model, by contrast is

$$X_i = \epsilon_i - \theta_1 \epsilon_{i-1},$$

where ϵ_i is as before and θ_1 is a parameter of the model. The generalised MA(q) model is, therefore, given by

$$X_i = (1 - \sum_{j=1}^{q} \theta_j \mathbf{B}^j)\epsilon_i,$$

where the θ_j terms are parameters of the model.

The AR and MA models can be combined to form an ARMA(p, q) model.

$$(1 - \sum_{j=1}^{p} a_j \mathbf{B}^j)X_i = (1 - \sum_{j=1}^{q} \theta_j \mathbf{B}^j)\epsilon_i.$$

The ARMA model can be differenced d times to form an ARIMA(p, d, q) model.

$$(1 - \sum_{j=1}^{p} a_j \mathbf{B}^j)(1 - \mathbf{B})^d X_i = (1 - \sum_{j=1}^{q} \theta_j \mathbf{B}^j)\epsilon_i, \tag{B.3}$$

where $d \in \mathbb{Z}_+$.

APPENDIX C

Plots of Licence Plate Matches Between Sites

This appendix contains plots of matches between plates at sites. The method and the plots are described in Section 5.5.

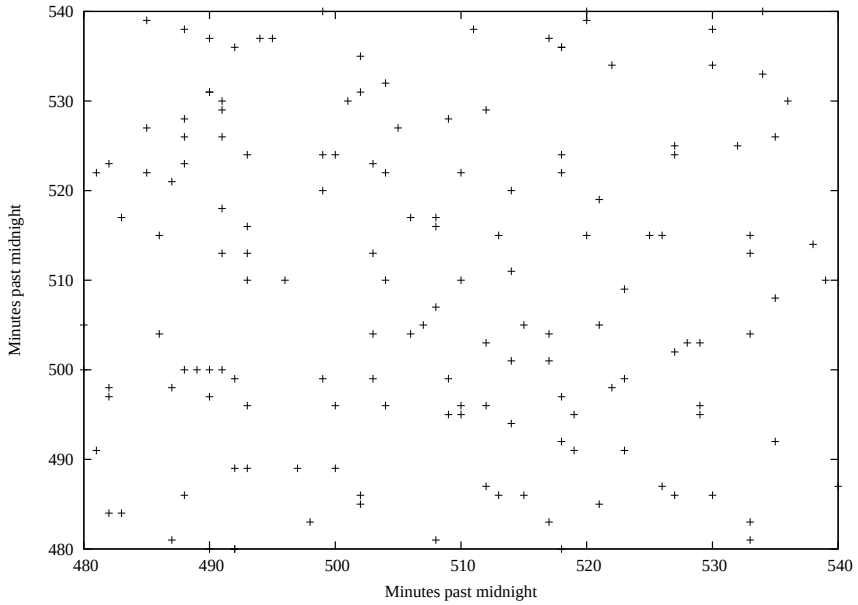

FIGURE C.1. Matches between vehicles observed at Lendal Bridge sites L and M on 28/6/00.

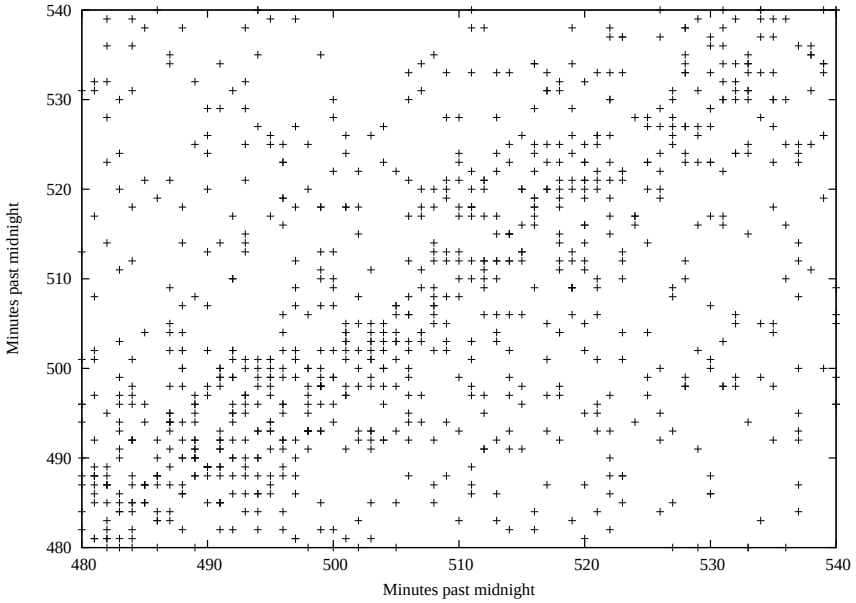

FIGURE C.2. Matches between vehicles at Lendal Bridge site M observed on 6/9/00 and 7/9/00.

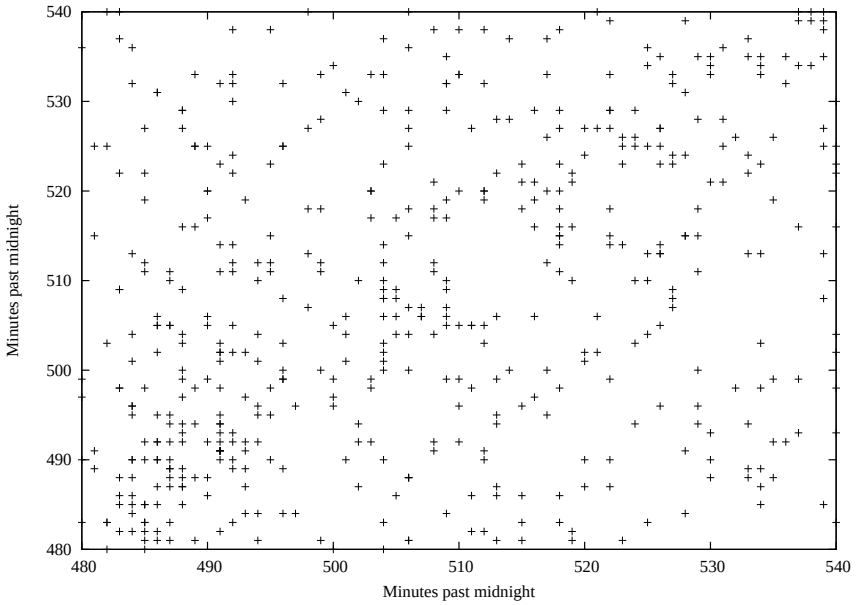

FIGURE C.3. Matches between vehicles at Lendal Bridge site M observed on 28/6/00 and 18/10/00.

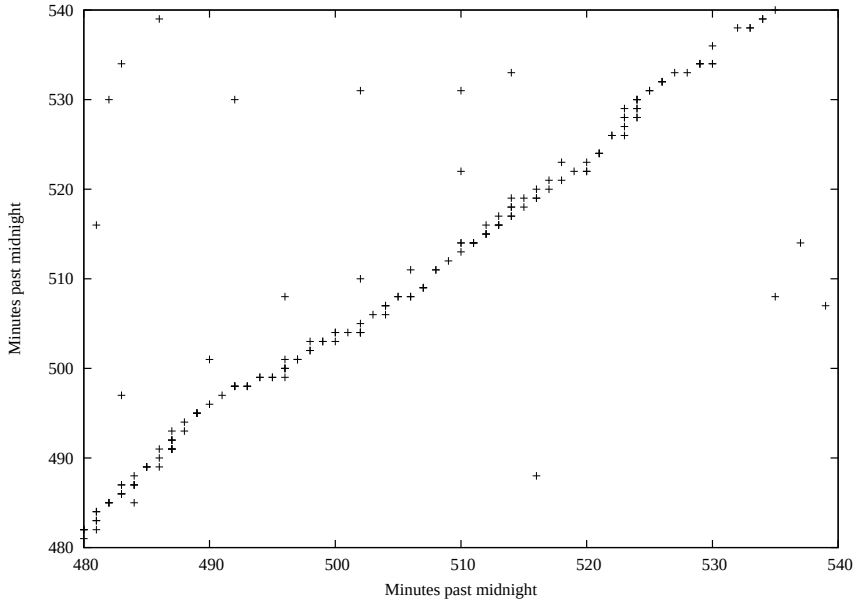

FIGURE C.4. Matches between vehicles at Lendal Bridge sites I and J on 28/6/00.

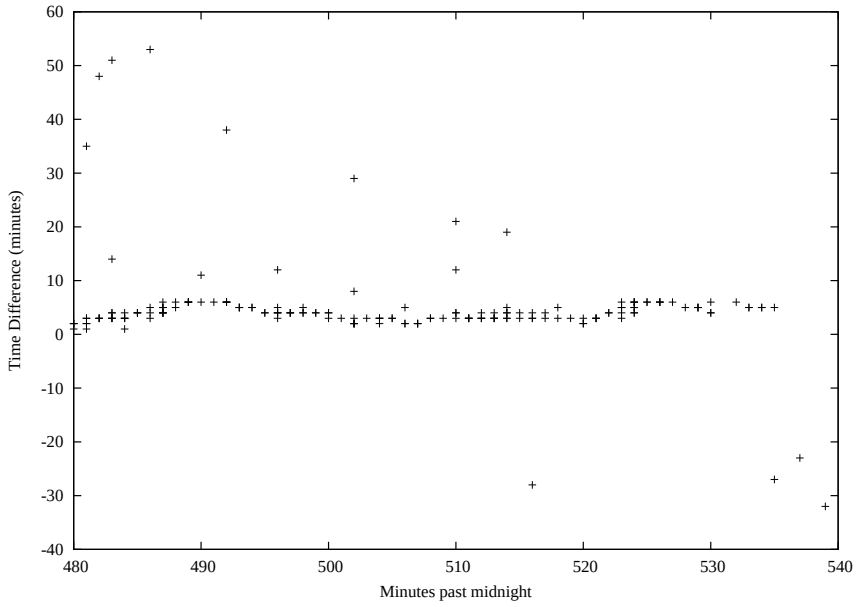

FIGURE C.5. Matches between vehicles at Lendal Bridge sites I and J on 28/6/00 showing time difference.

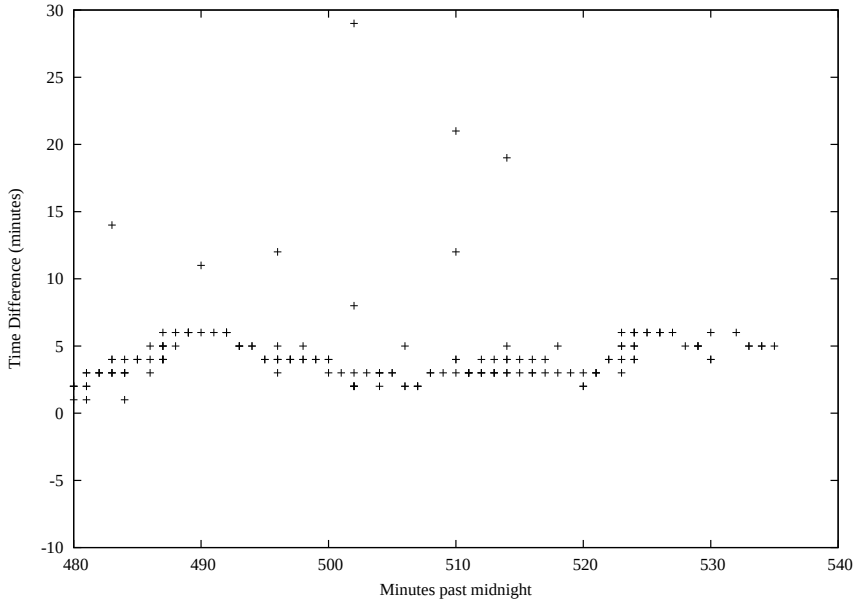

FIGURE C.6. Matches between vehicles at Lendal Bridge sites I and J on 28/6/00 showing time difference. (Detail of previous figure).

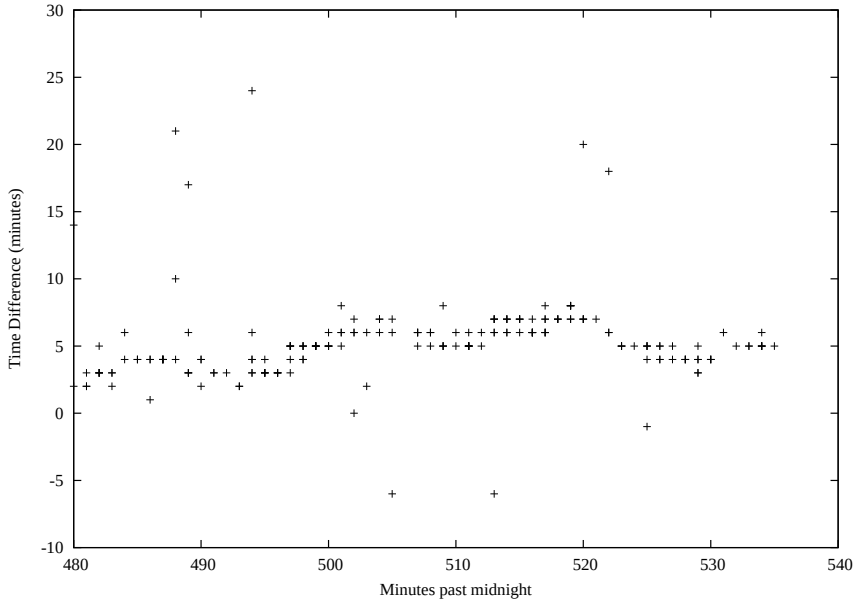

FIGURE C.7. Matches between vehicles at Lendal Bridge sites I and J on 8/9/00 showing time difference. (Last day before bridge closure).

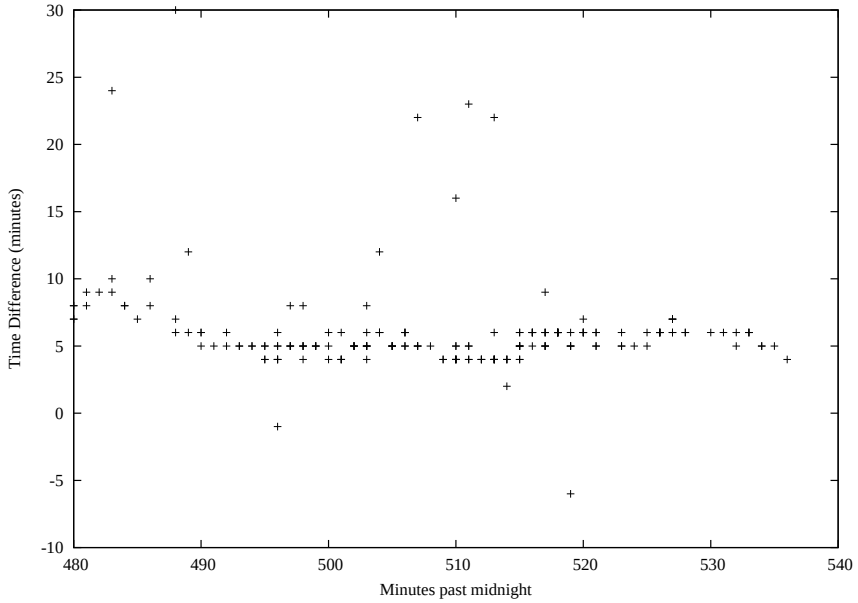

FIGURE C.8. Matches between vehicles at Lendal Bridge sites I and J on 11/9/00 showing time difference. (First day after bridge closure).

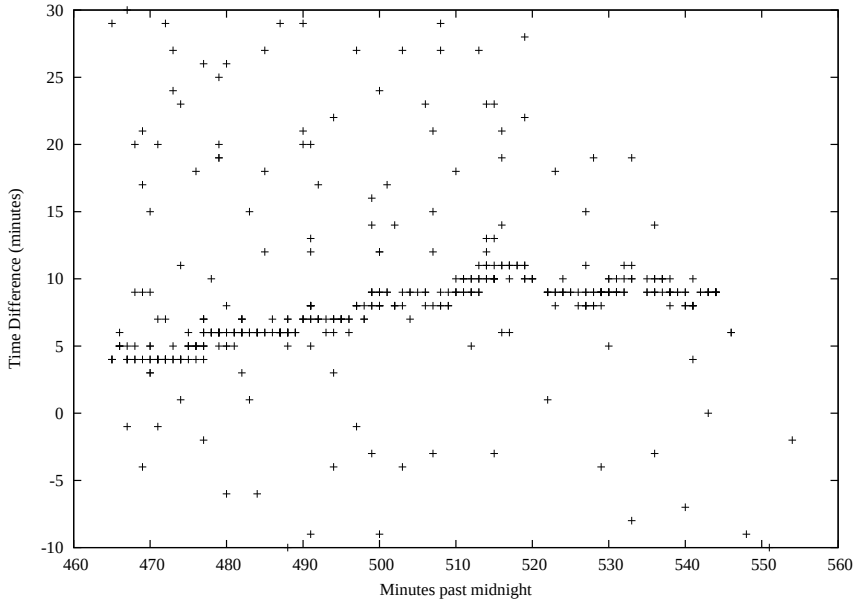

FIGURE C.9. Matches between vehicles at Fishergate sites E and A on 25/6/01.

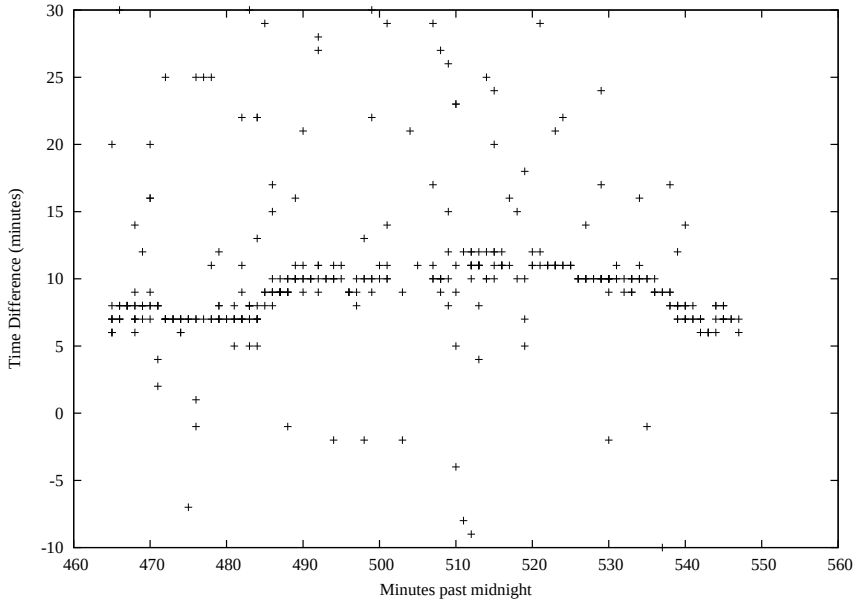

FIGURE C.10. Matches between vehicles at Fishergate sites E and A on 26/6/01.

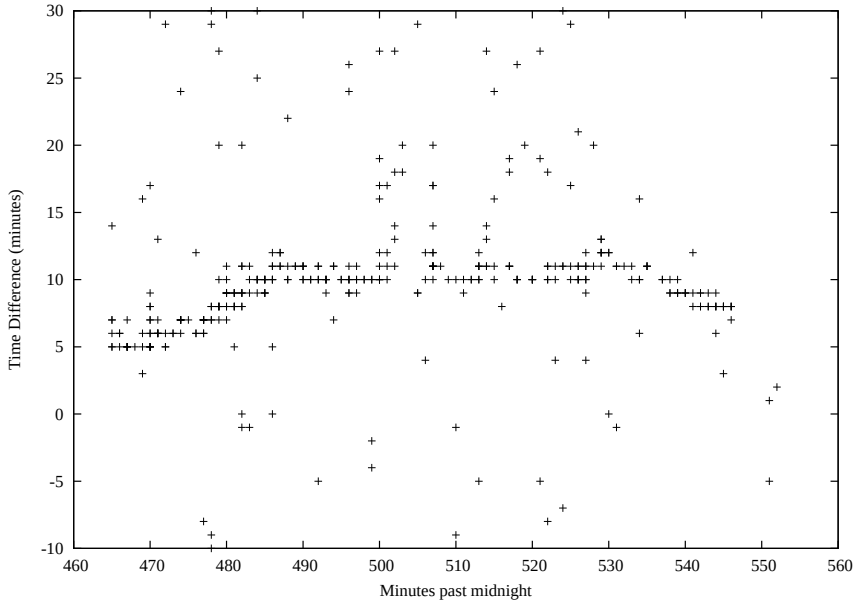

FIGURE C.11. Matches between vehicles at Fishergate sites E and A on 27/6/01.

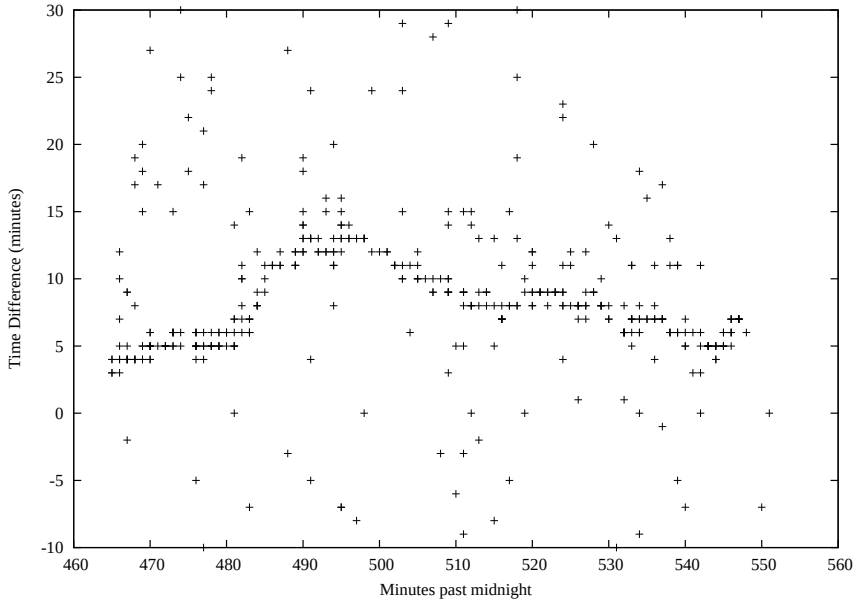

FIGURE C.12. Matches between vehicles at Fishergate sites E and A on 28/6/01.

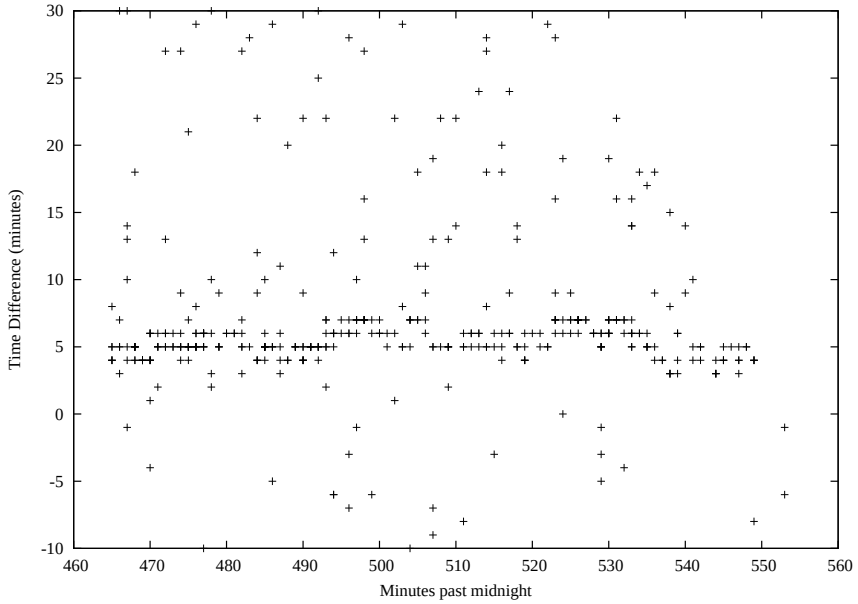

FIGURE C.13. Matches between vehicles at Fishergate sites E and A on 29/6/01.

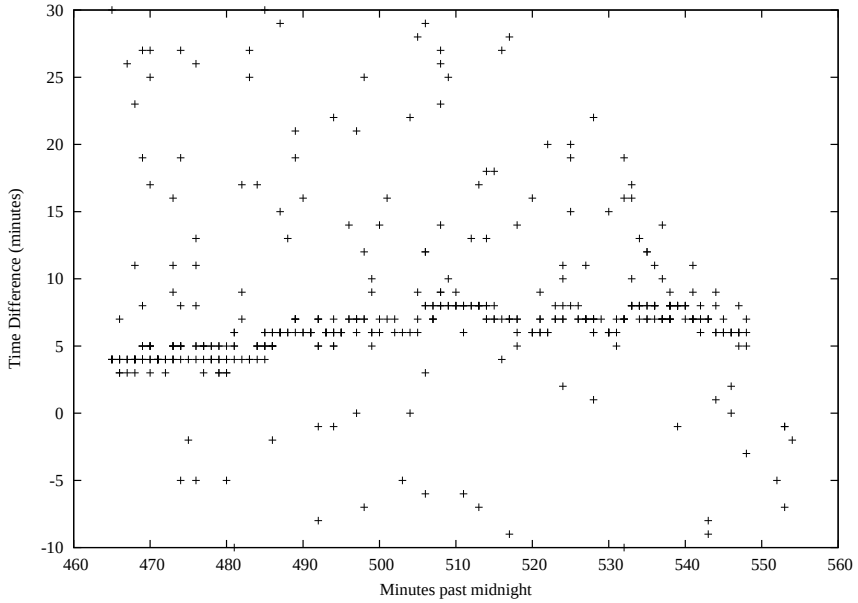

FIGURE C.14. Matches between vehicles at Fishergate sites E and A on 2/7/01.

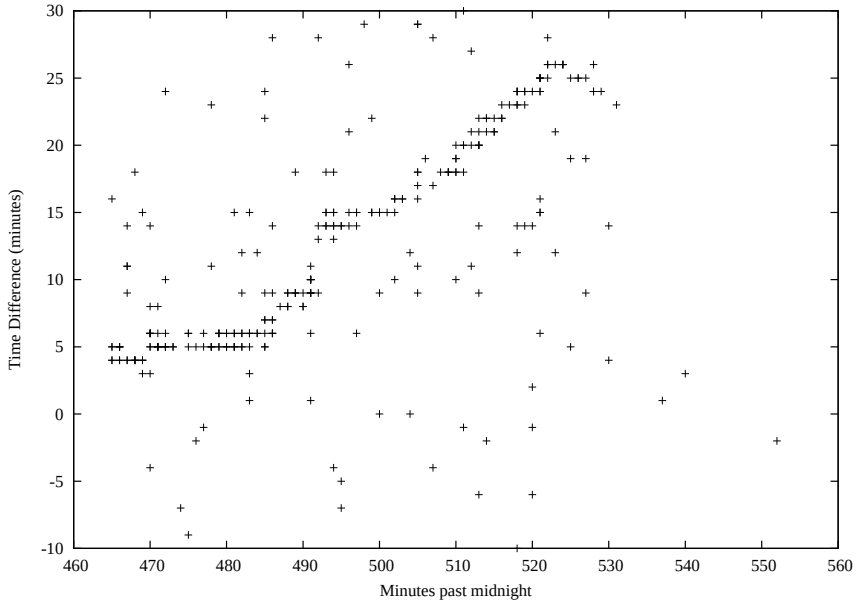

FIGURE C.15. Matches between vehicles at Fishergate sites E and A on 3/7/01. (First day of partial closure.)

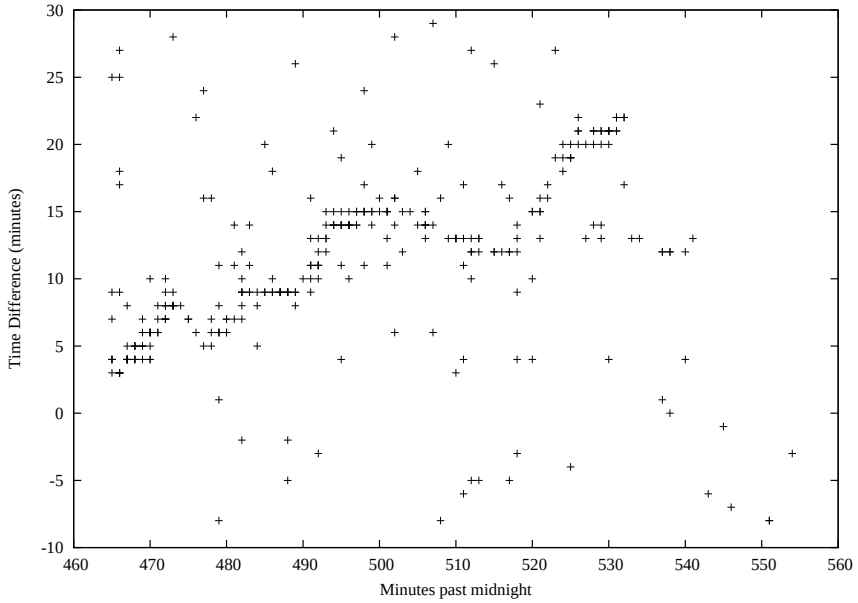

FIGURE C.16. Matches between vehicles at Fishergate sites E and A on 4/7/01.

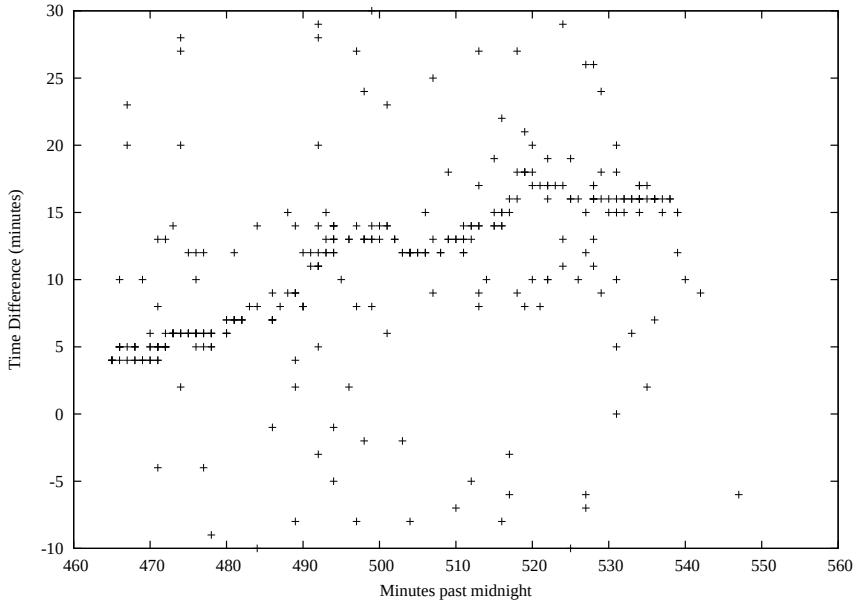

FIGURE C.17. Matches between vehicles at Fishergate sites E and A on 5/7/01.

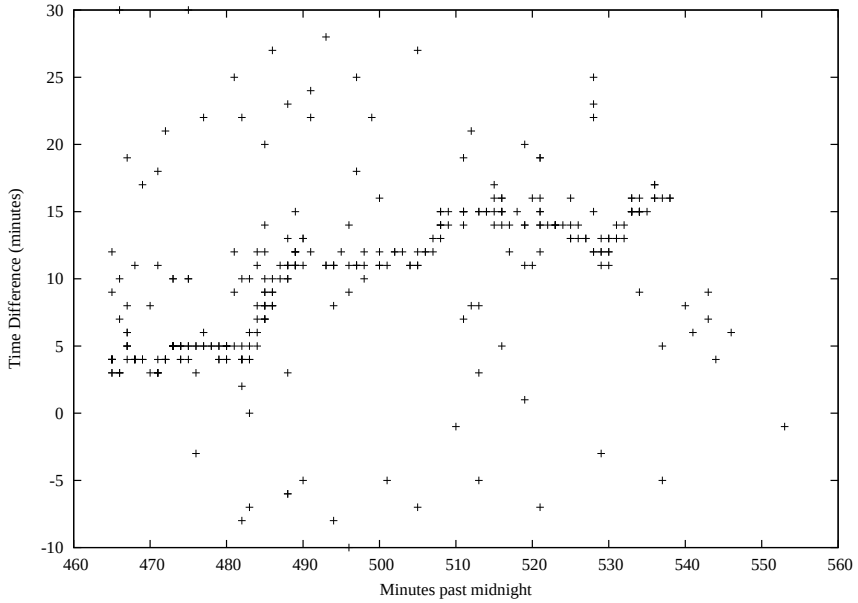

FIGURE C.18. Matches between vehicles at Fishergate sites E and A on 12/7/01.

APPENDIX D

Histograms of Travel Times

This chapter contains historgrams of travel time data for the Lendal Bridge and Fishergate surveys. Refer to Section 5.7.1 for details and analysis.

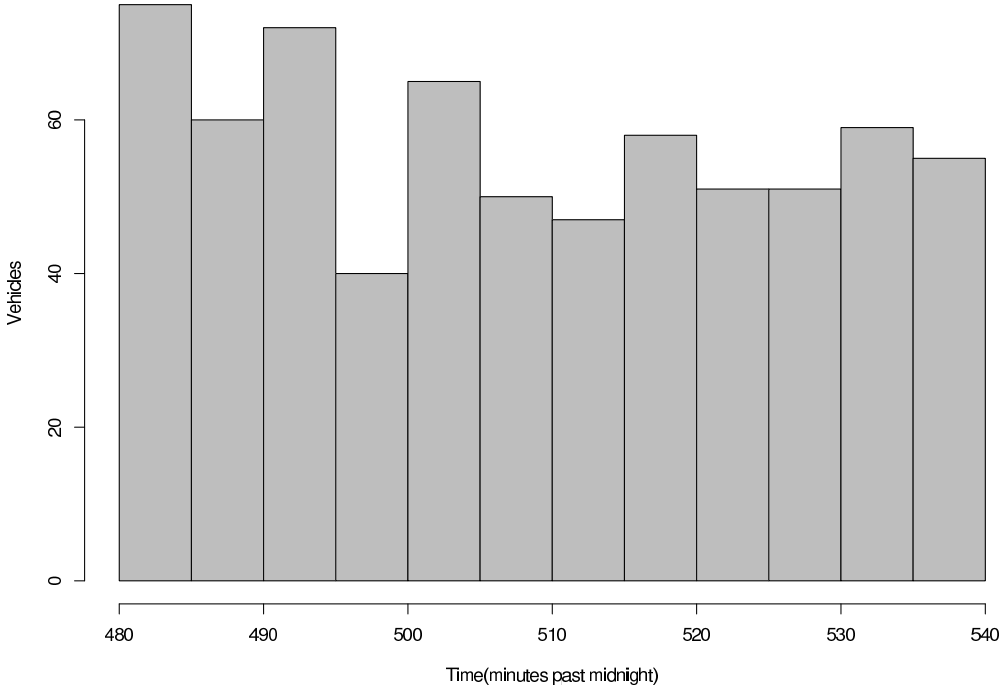

FIGURE D.1. Lendal Bridge survey arrival times at site A 8/9/00

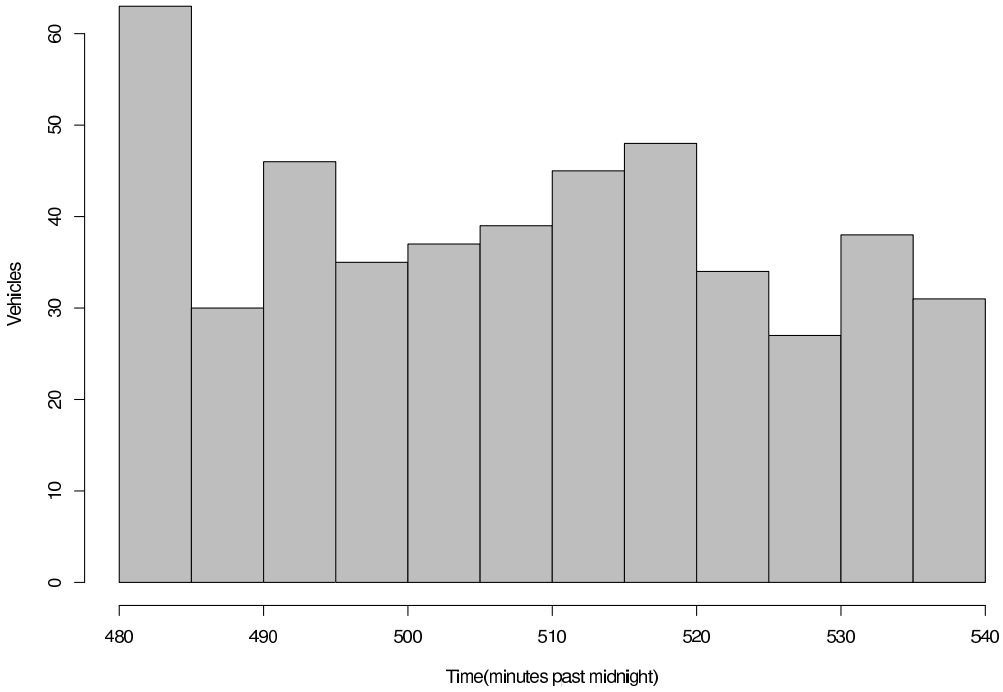

FIGURE D.2. Lendal Bridge survey arrival times at site B 8/9/00

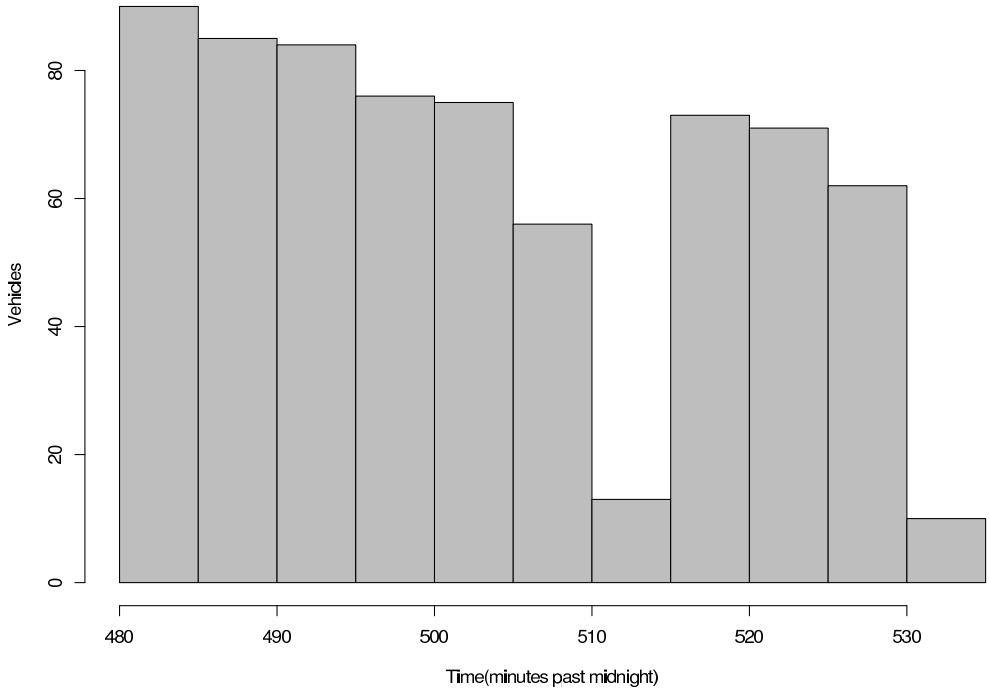

FIGURE D.3. Lendal Bridge survey arrival times at site C 8/9/00

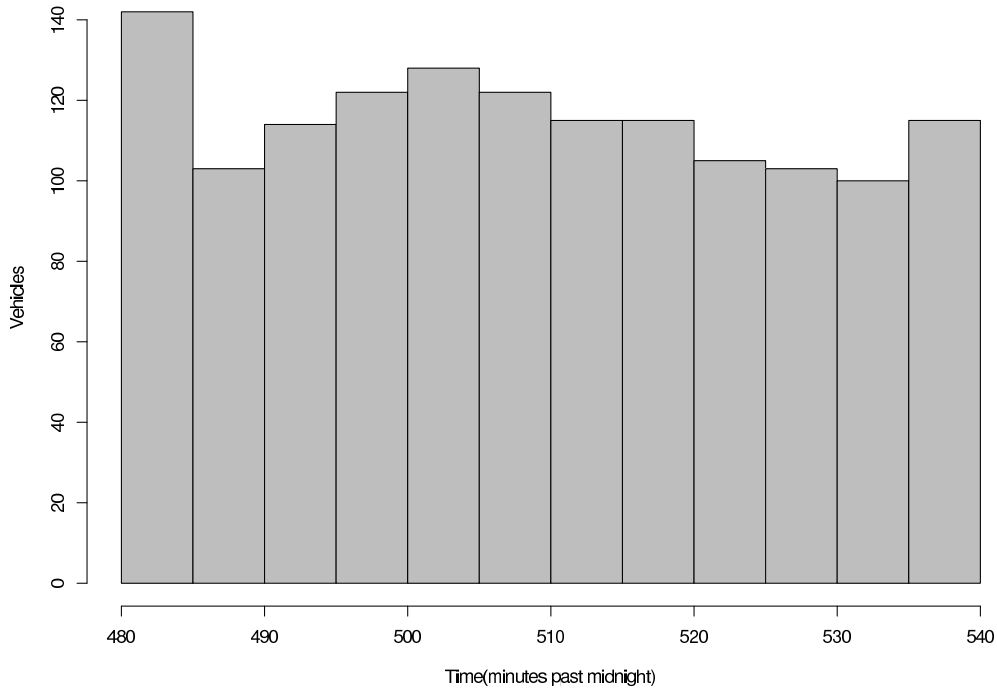

FIGURE D.4. Lendal Bridge survey arrival times at site D 8/9/00

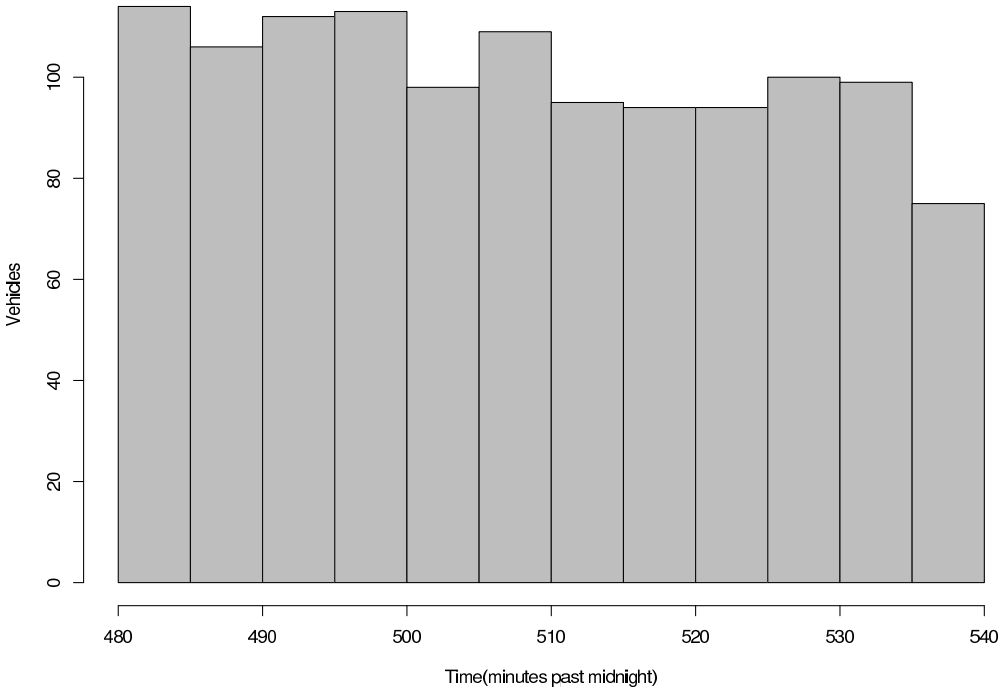

FIGURE D.5. Lendal Bridge survey arrival times at site E 8/9/00

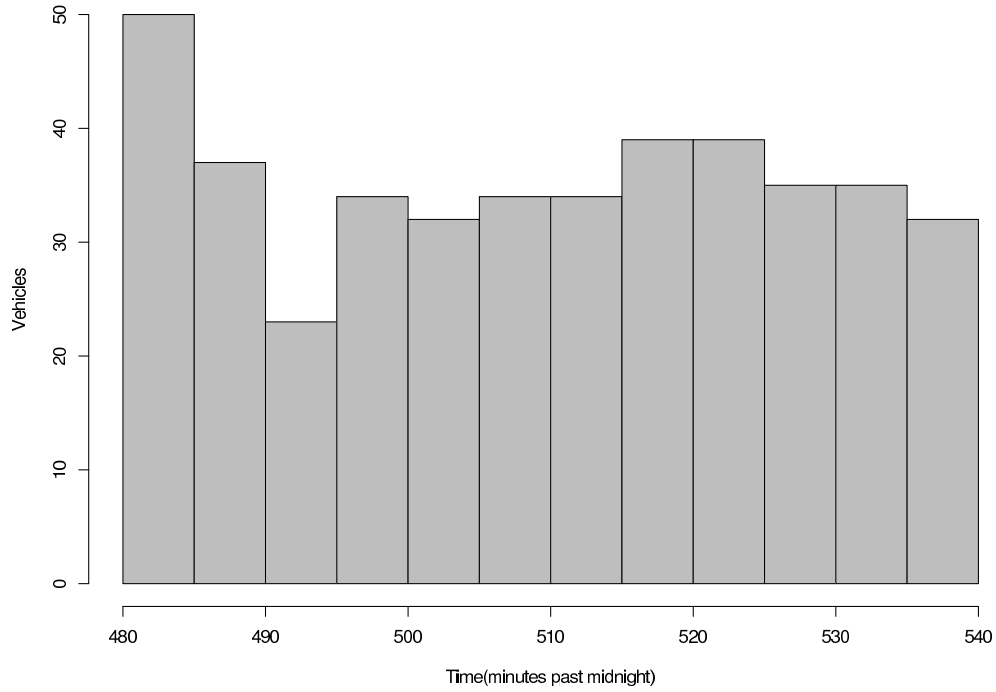

FIGURE D.6. Lendal Bridge survey arrival times at site F 8/9/00

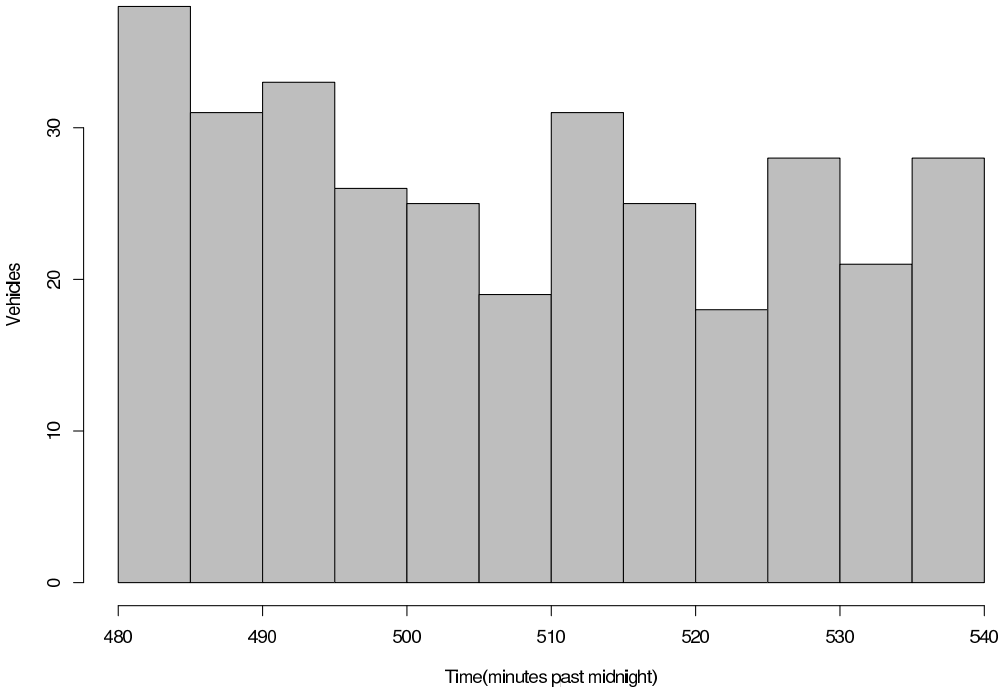

FIGURE D.7. Lendal Bridge survey arrival times at site G 8/9/00

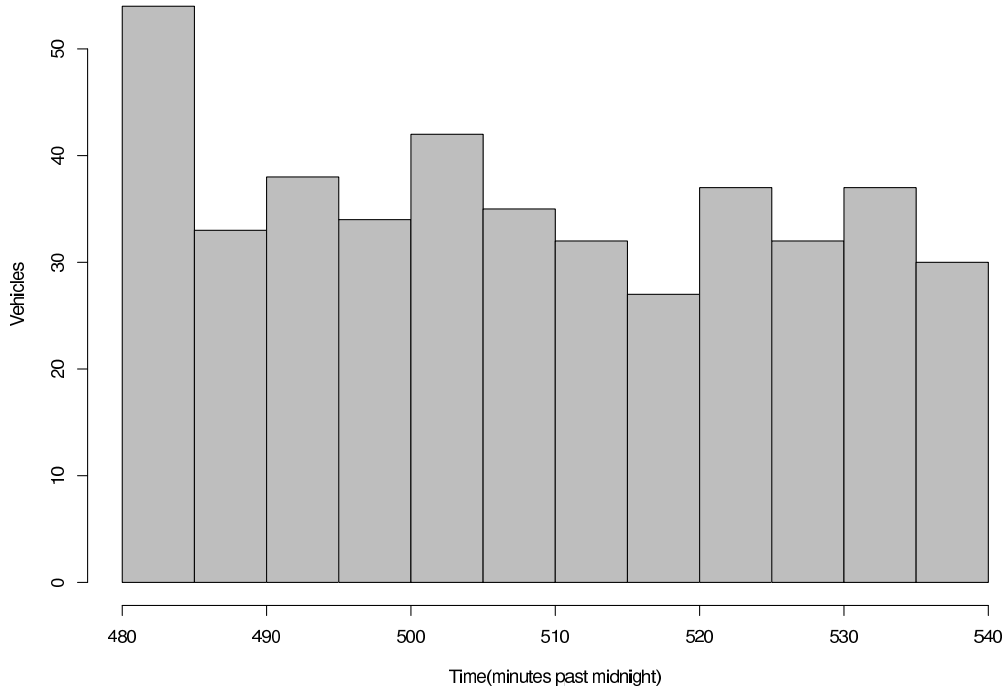

FIGURE D.8. Lendal Bridge survey arrival times at site H 8/9/00

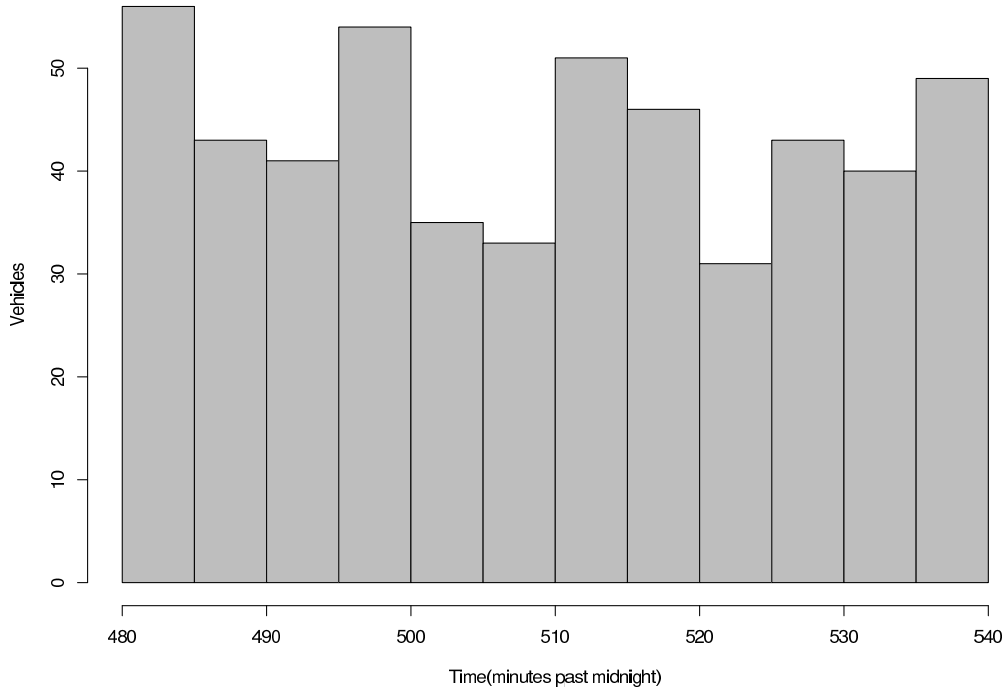

FIGURE D.9. Lendal Bridge survey arrival times at site I 8/9/00

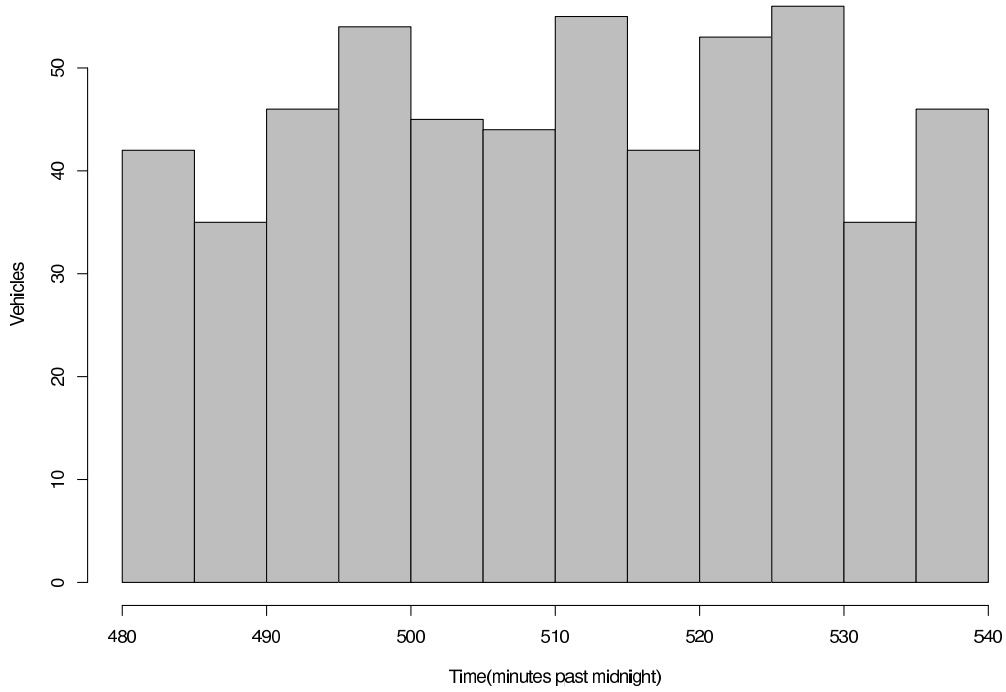

FIGURE D.10. Lendal Bridge survey arrival times at site J 8/9/00

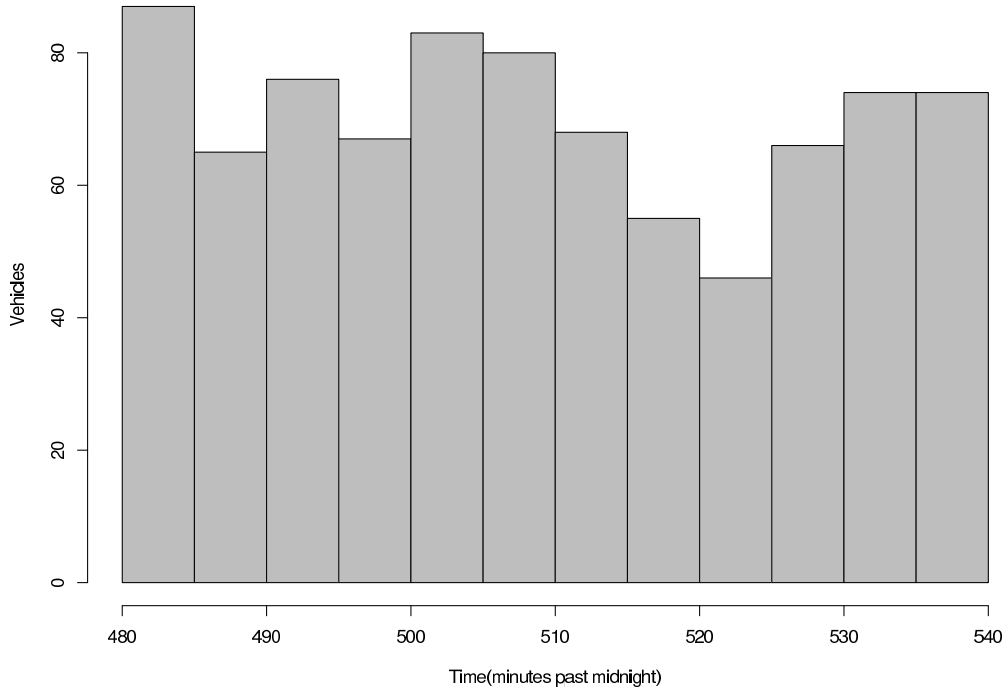

FIGURE D.11. Lendal Bridge survey arrival times at site K 8/9/00

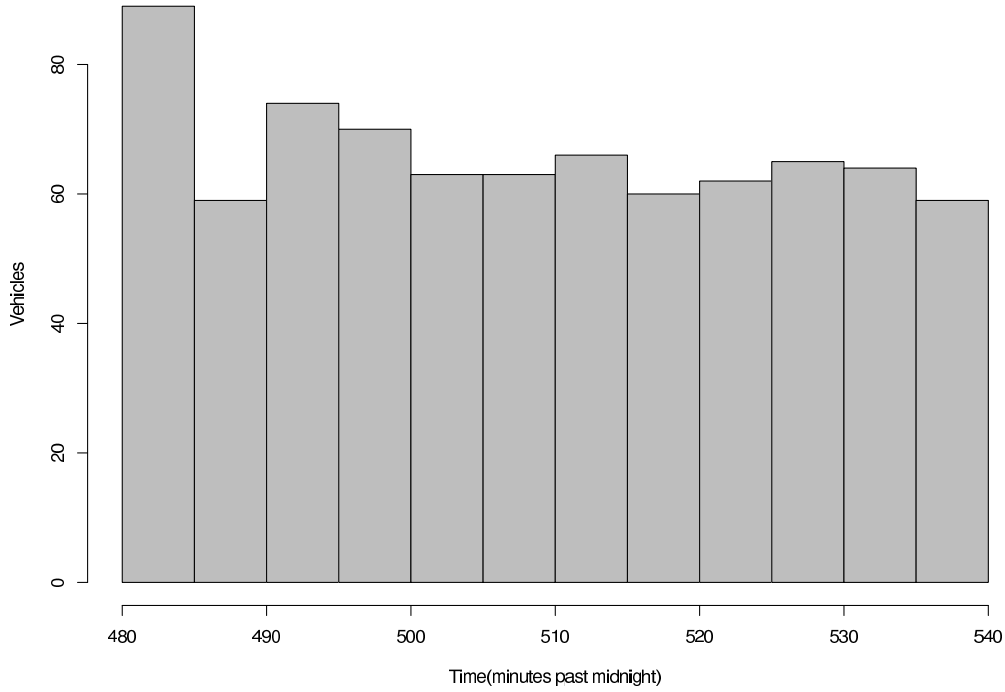

FIGURE D.12. Lendal Bridge survey arrival times at site L 8/9/00

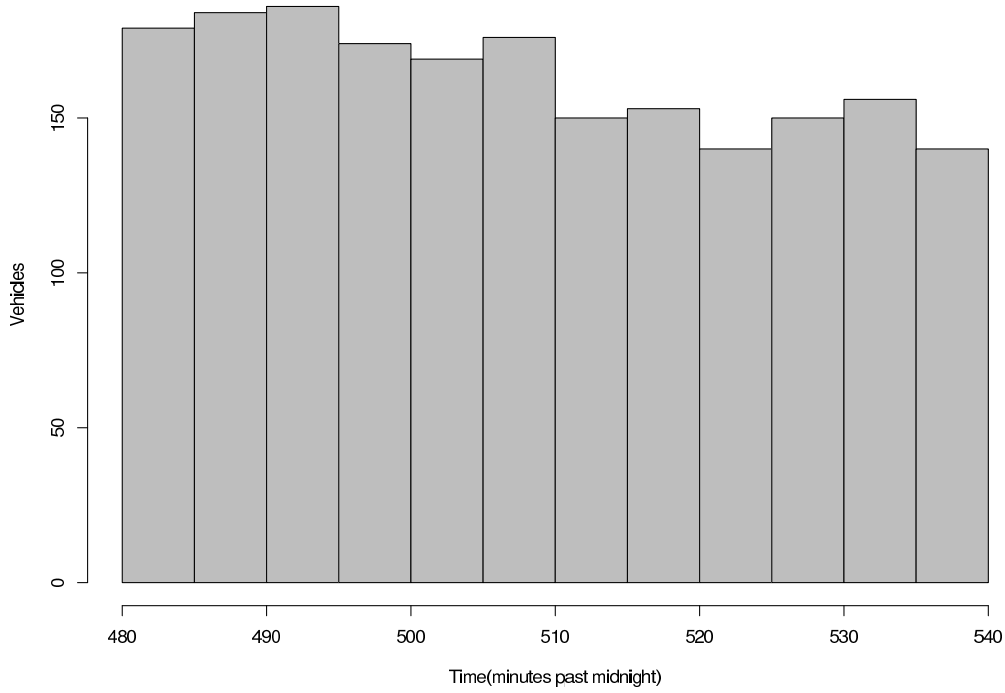

FIGURE D.13. Lendal Bridge survey arrival times at site M 8/9/00

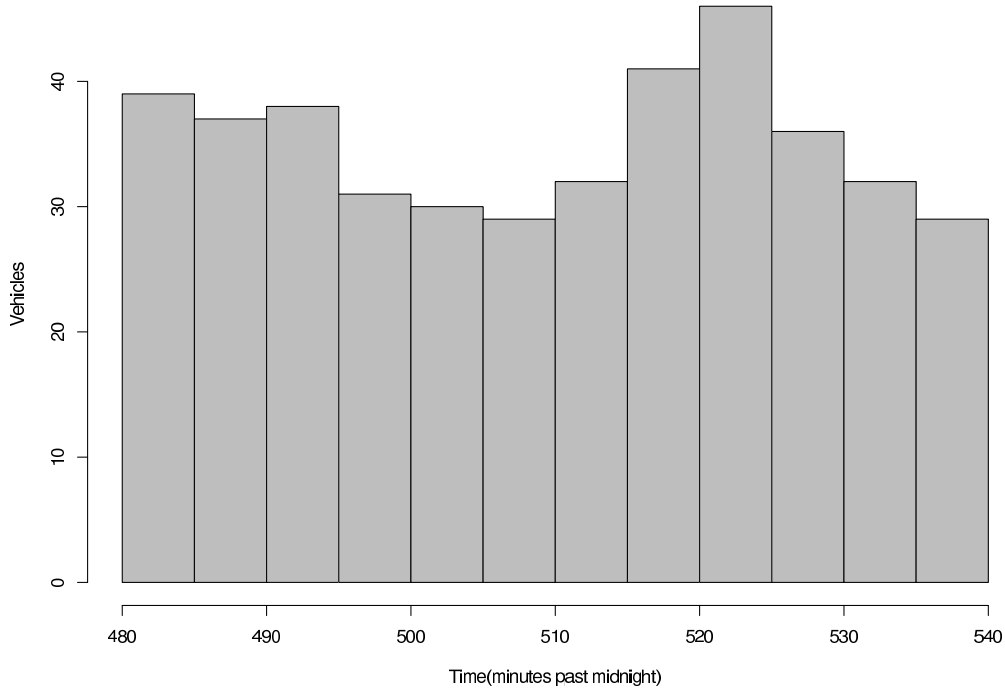

FIGURE D.14. Lendal Bridge survey arrival times at site F 7/9/00

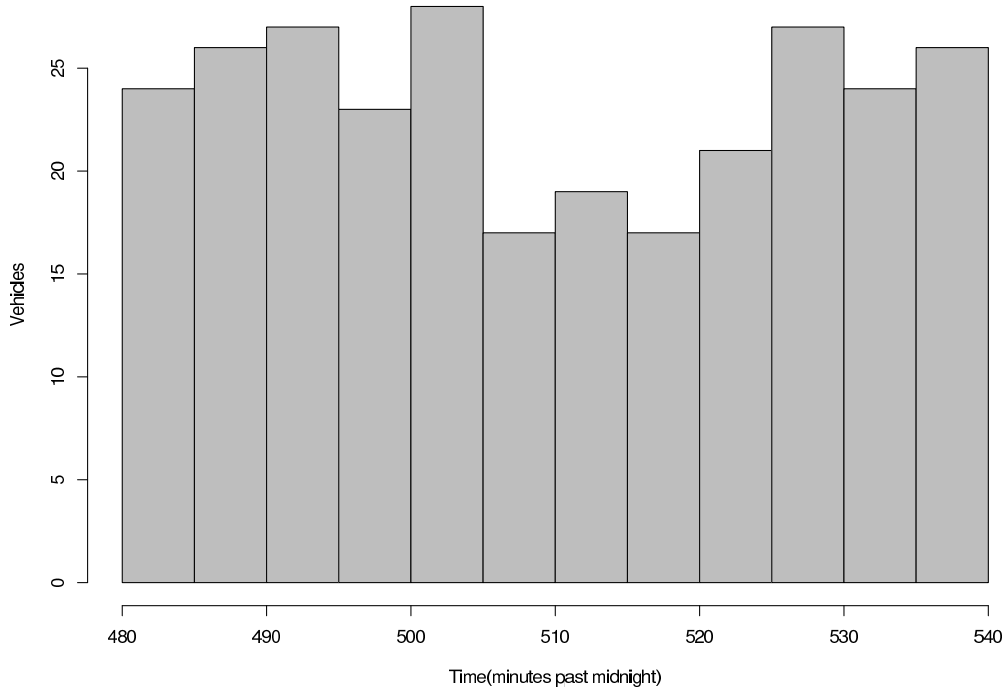

FIGURE D.15. Lendal Bridge survey arrival times at site F 11/9/00

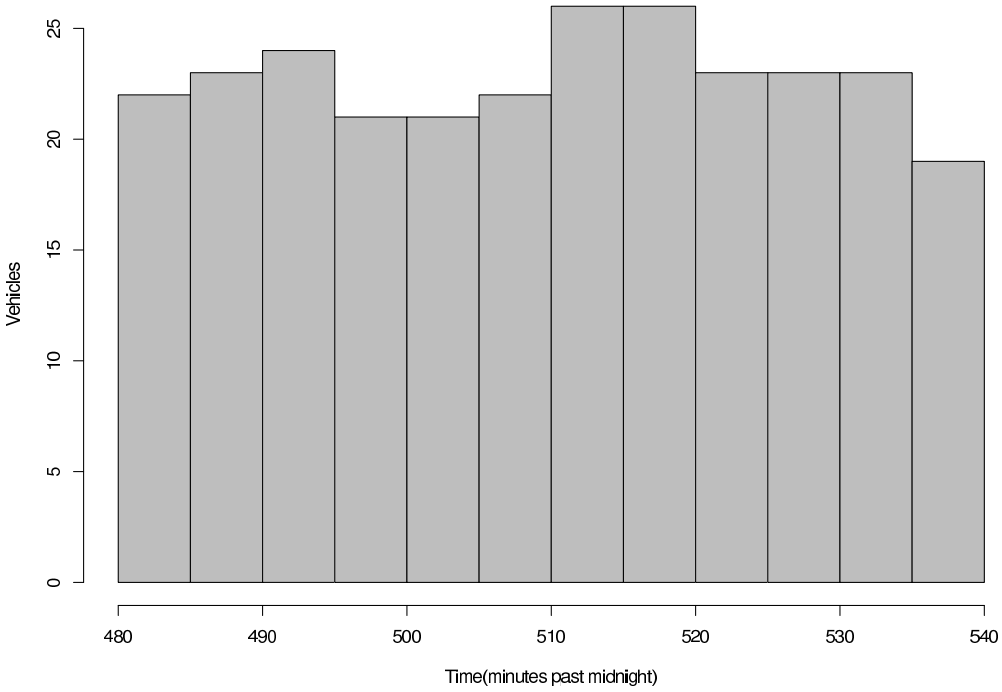

FIGURE D.16. Lendal Bridge survey arrival times at site F 13/9/00

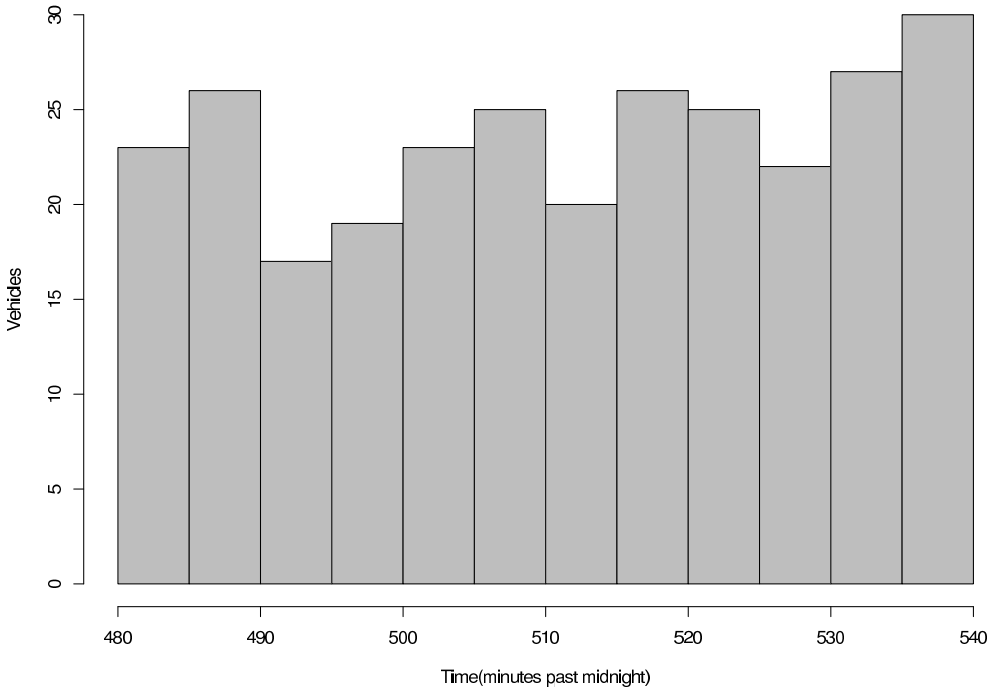

FIGURE D.17. Lendal Bridge survey arrival times at site F 27/9/00

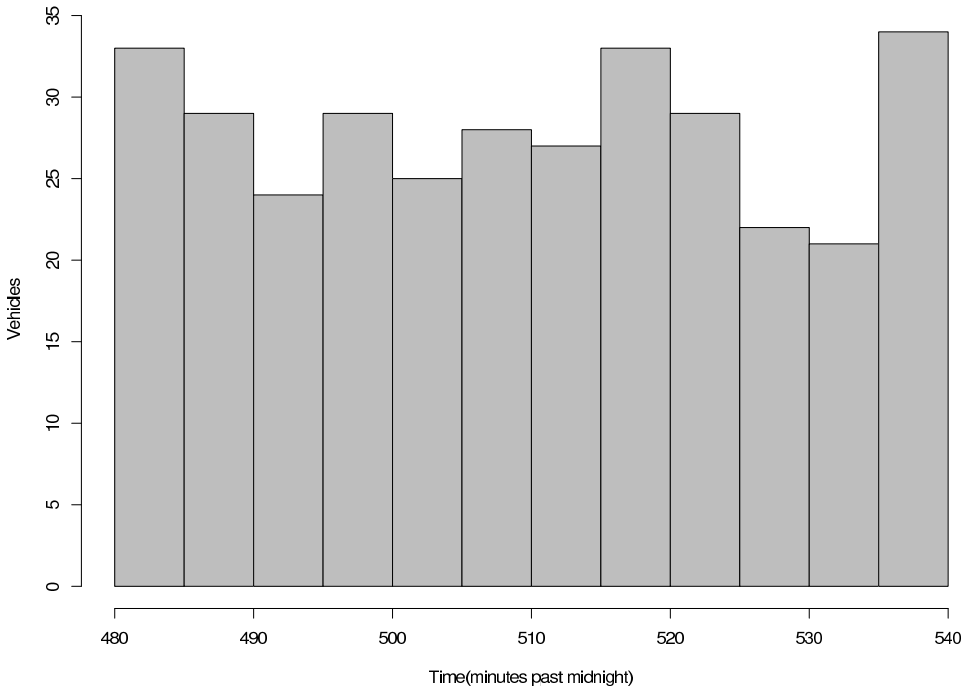

FIGURE D.18. Lendal Bridge survey arrival times at site F 18/10/00

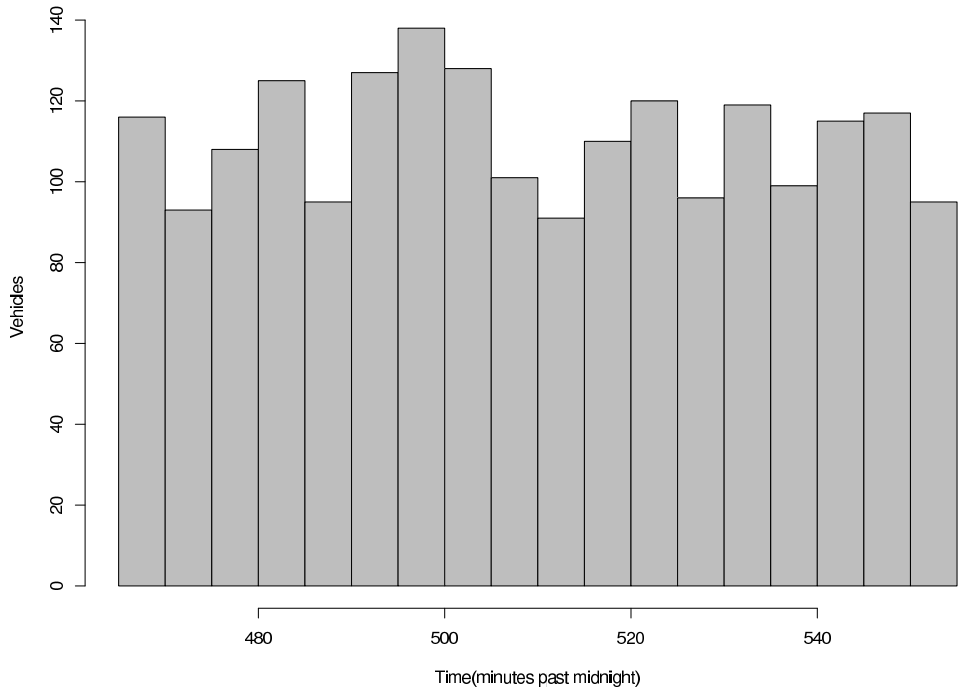

FIGURE D.19. Fishergate survey arrival times at site A 2/7/01.

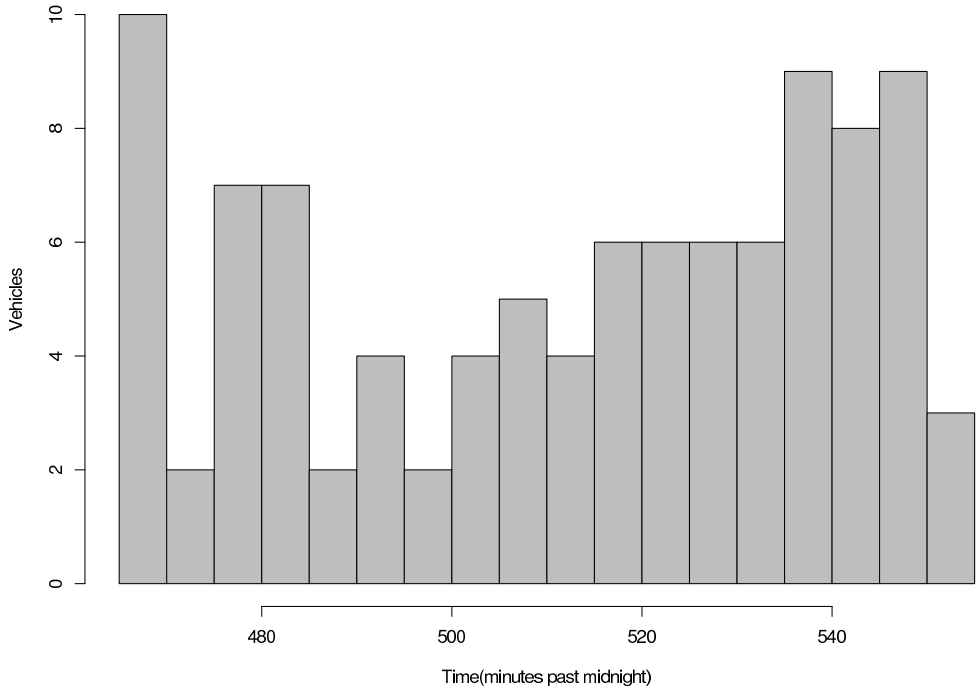

FIGURE D.20. Fishergate survey arrival times at site B 2/7/01.

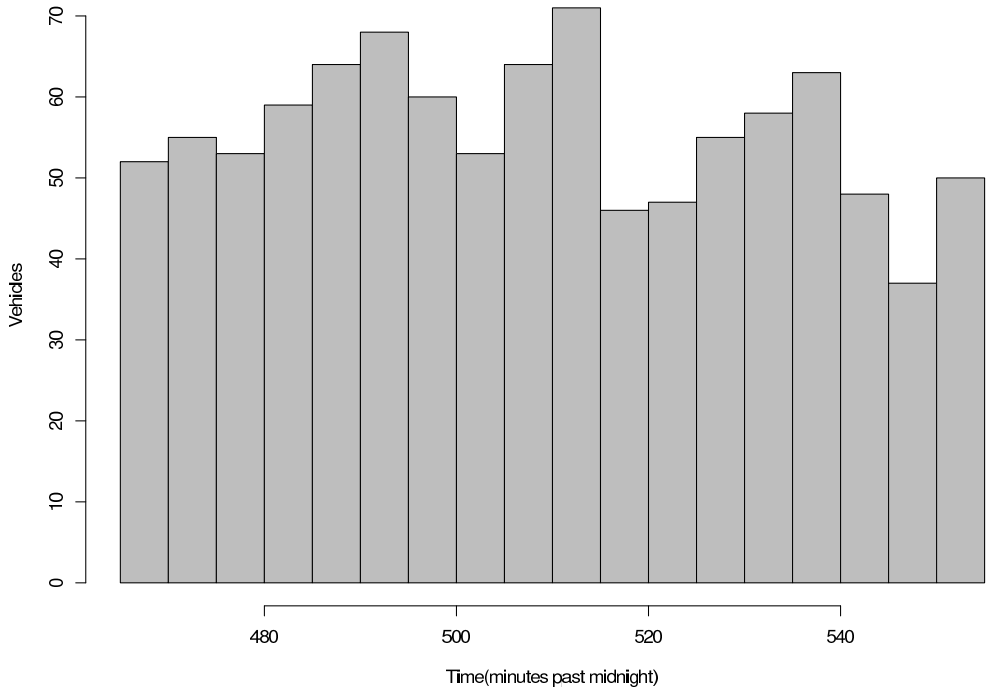

FIGURE D.21. Fishergate survey arrival times at site C 2/7/01.

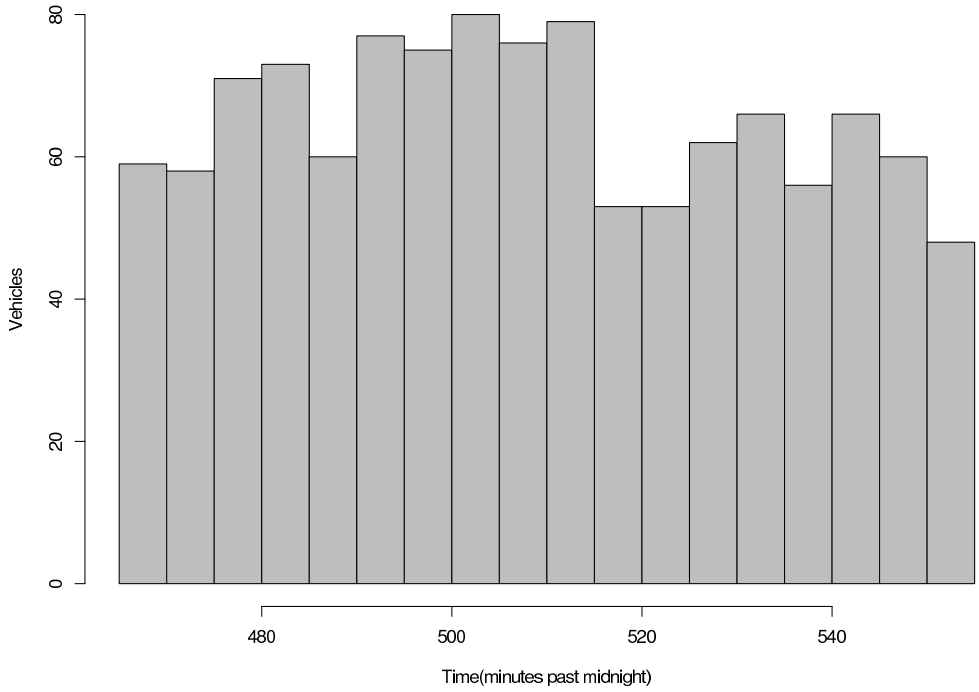

FIGURE D.22. Fishergate survey arrival times at site D 2/7/01.

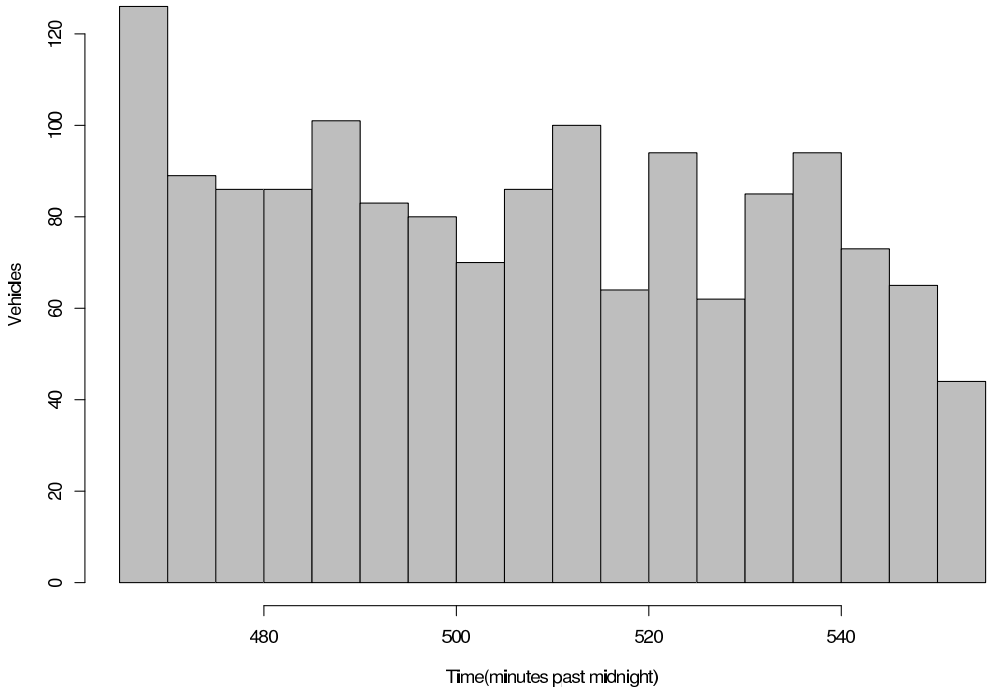

FIGURE D.23. Fishergate survey arrival times at site E 2/7/01.

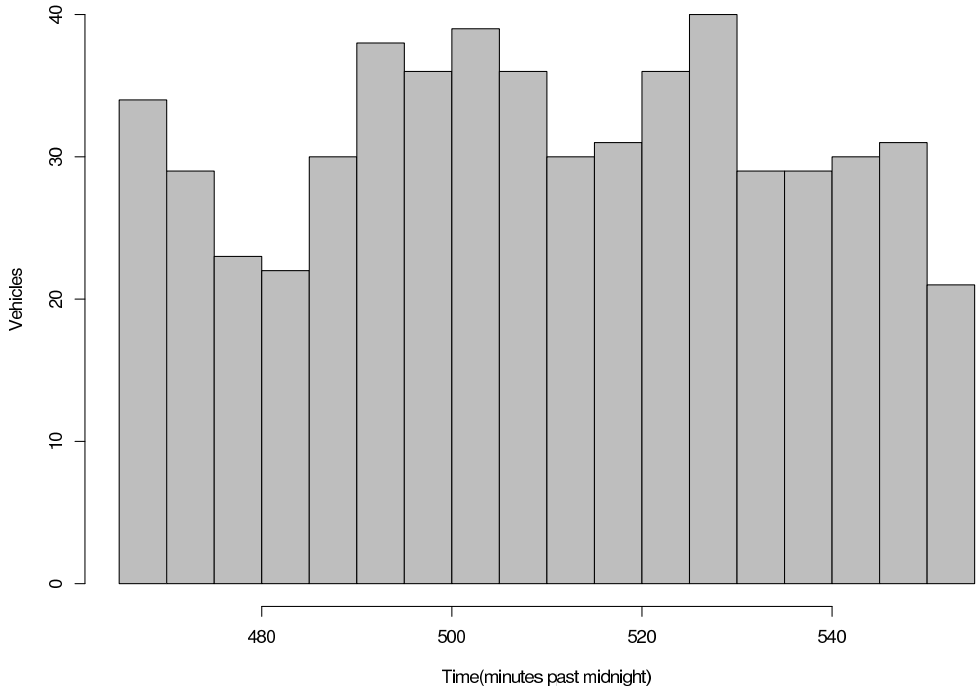

FIGURE D.24. Fishergate survey arrival times at site F 2/7/01.

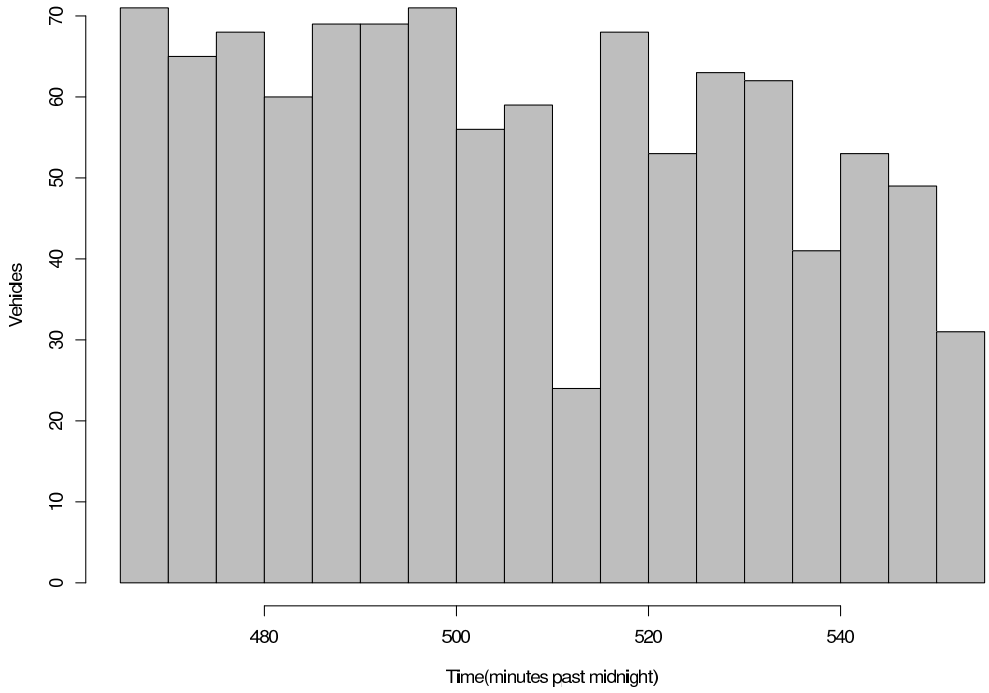

FIGURE D.25. Fishergate survey arrival times at site G 2/7/01.

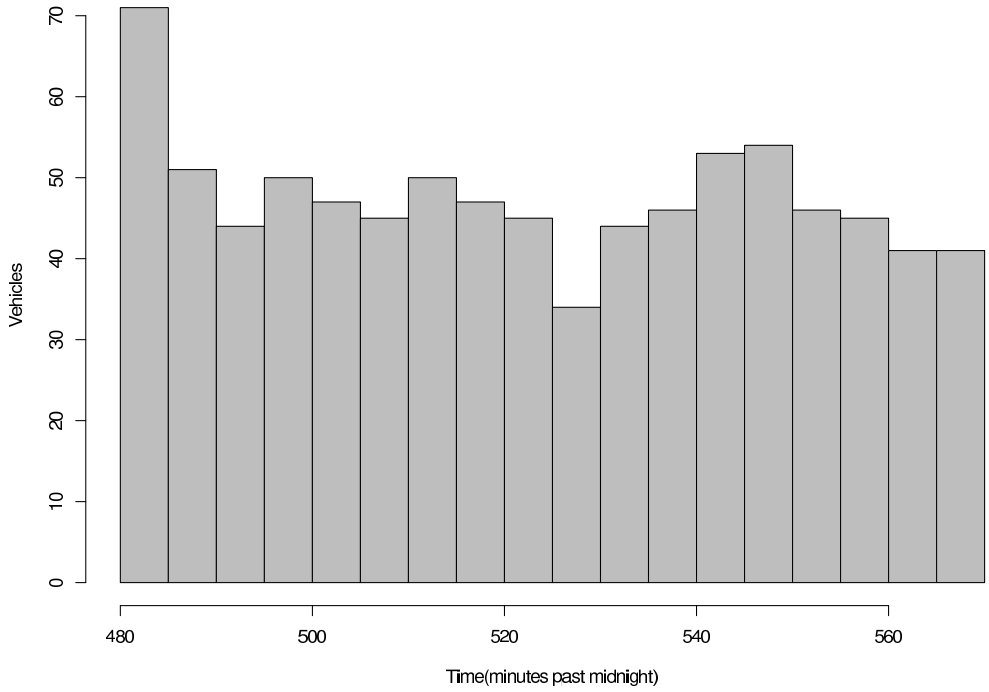

FIGURE D.26. Fishergate survey arrival times at site H 2/7/01.

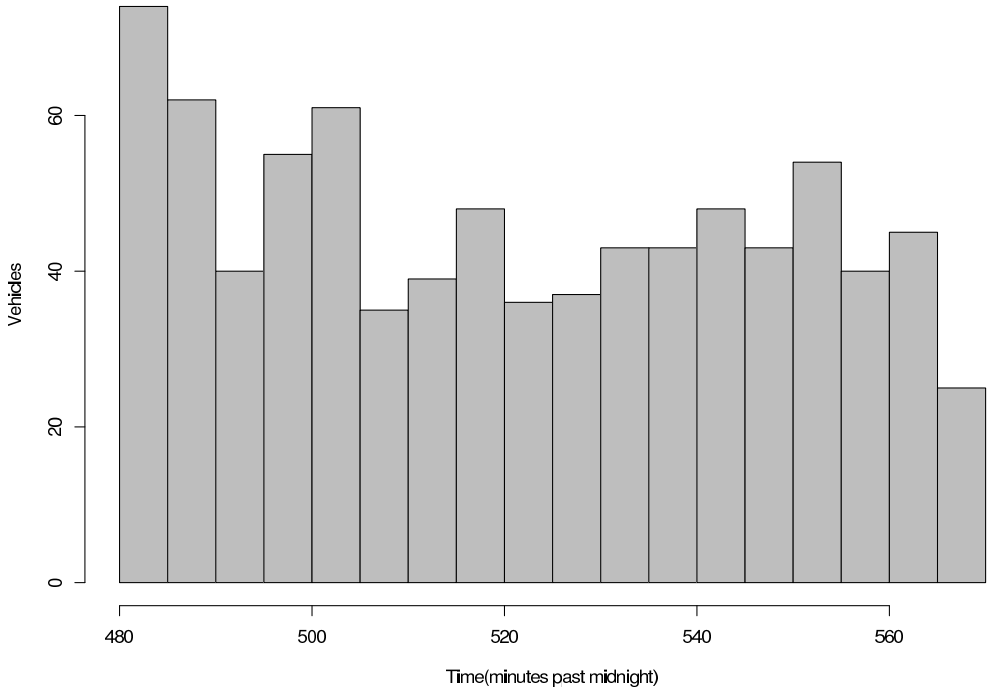

FIGURE D.27. Fishergate survey arrival times at site I 2/7/01.

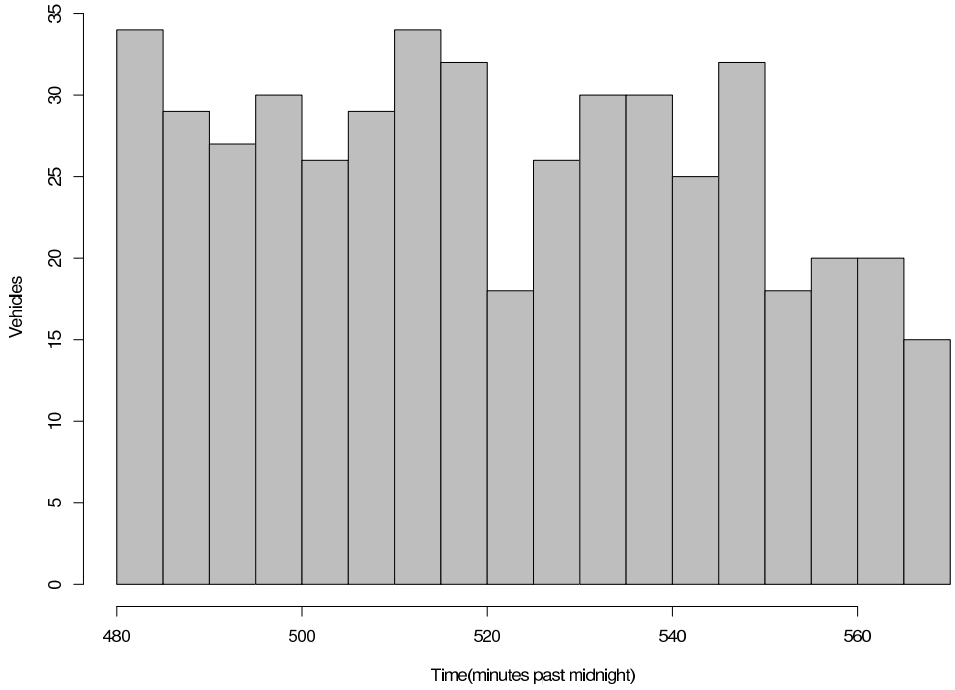

FIGURE D.28. Fishergate survey arrival times at site J 2/7/01.

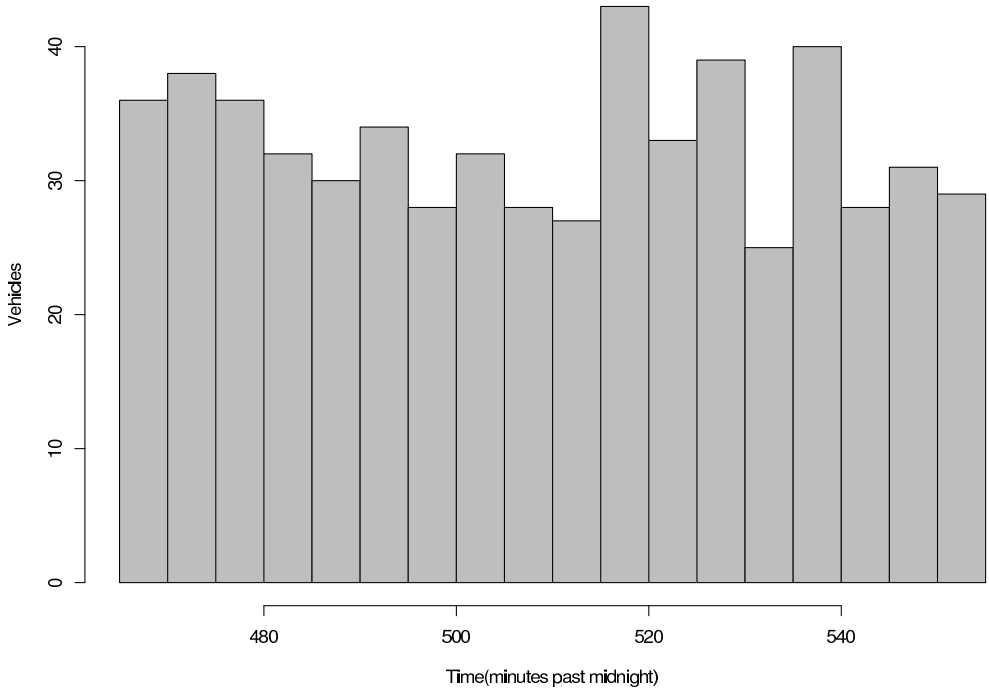

FIGURE D.29. Fishergate survey arrival times at site K 2/7/01.

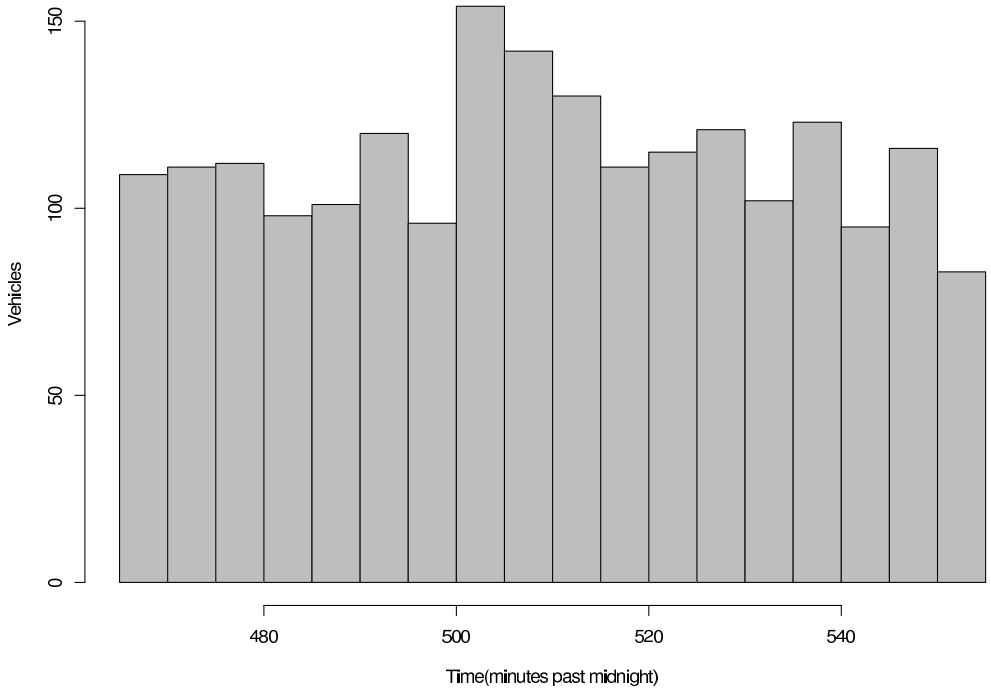

FIGURE D.30. Fishergate survey arrival times at site A 28/6/01.

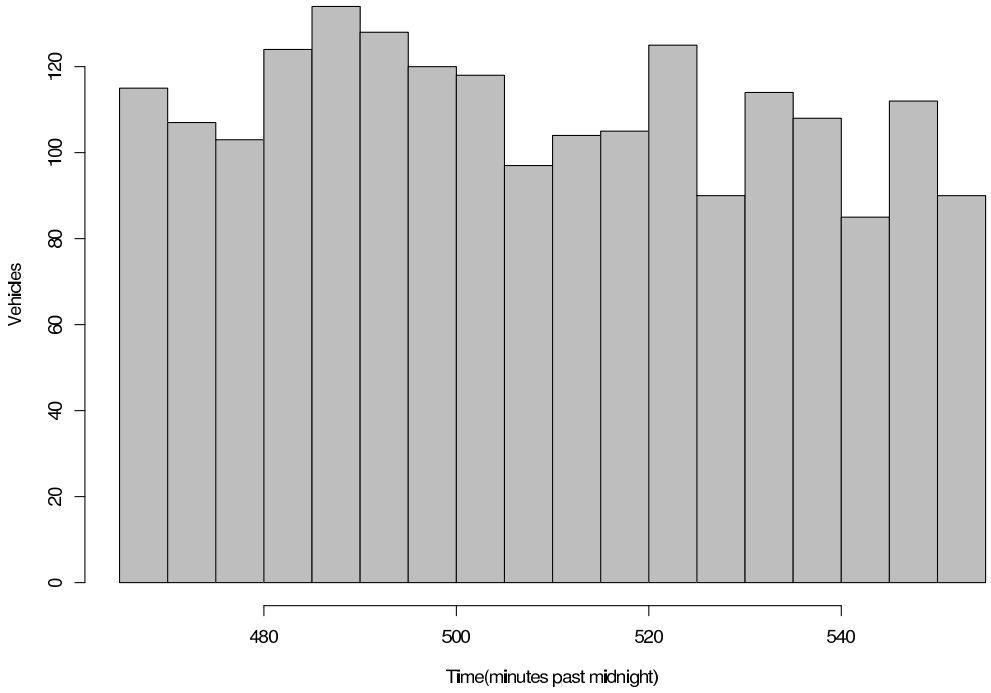

FIGURE D.31. Fishergate survey arrival times at site A 29/6/01.

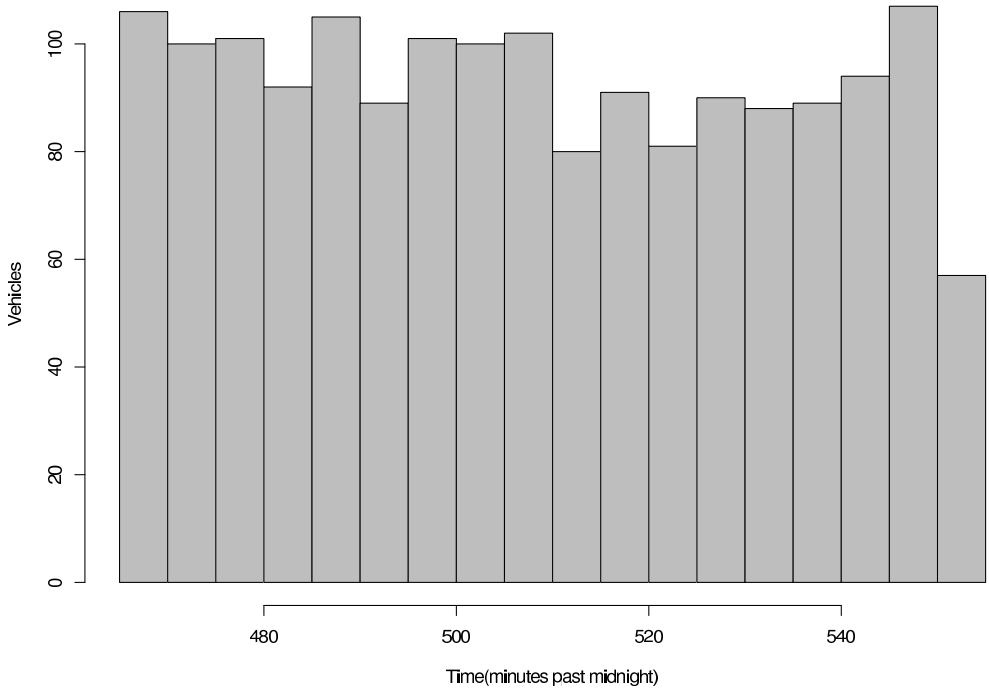

FIGURE D.32. Fishergate survey arrival times at site A 3/7/01.

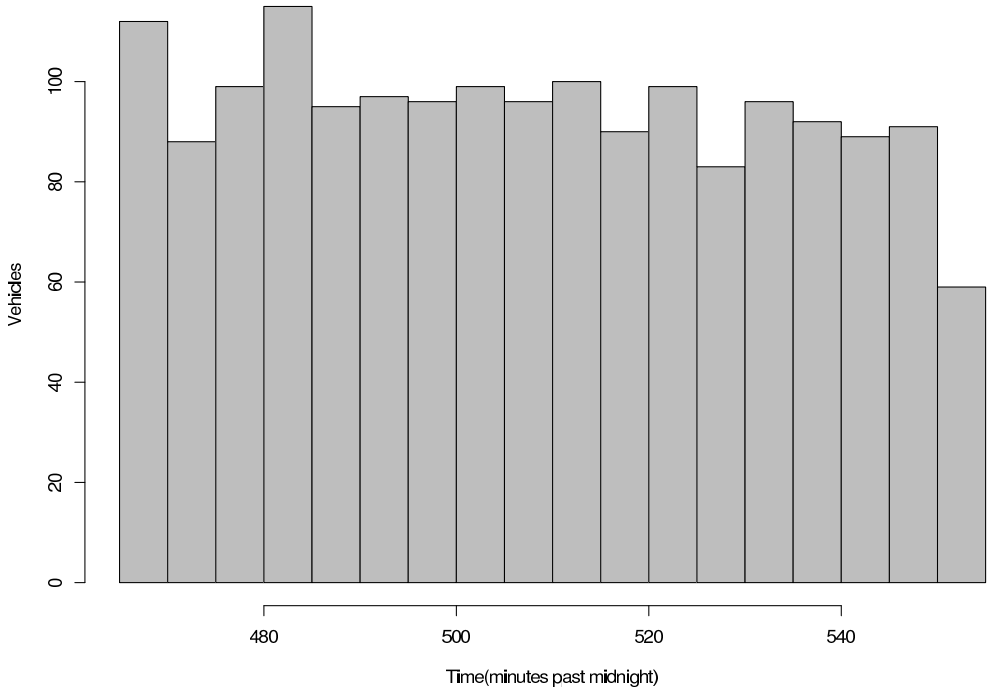

FIGURE D.33. Fishergate survey arrival times at site A 4/7/01.

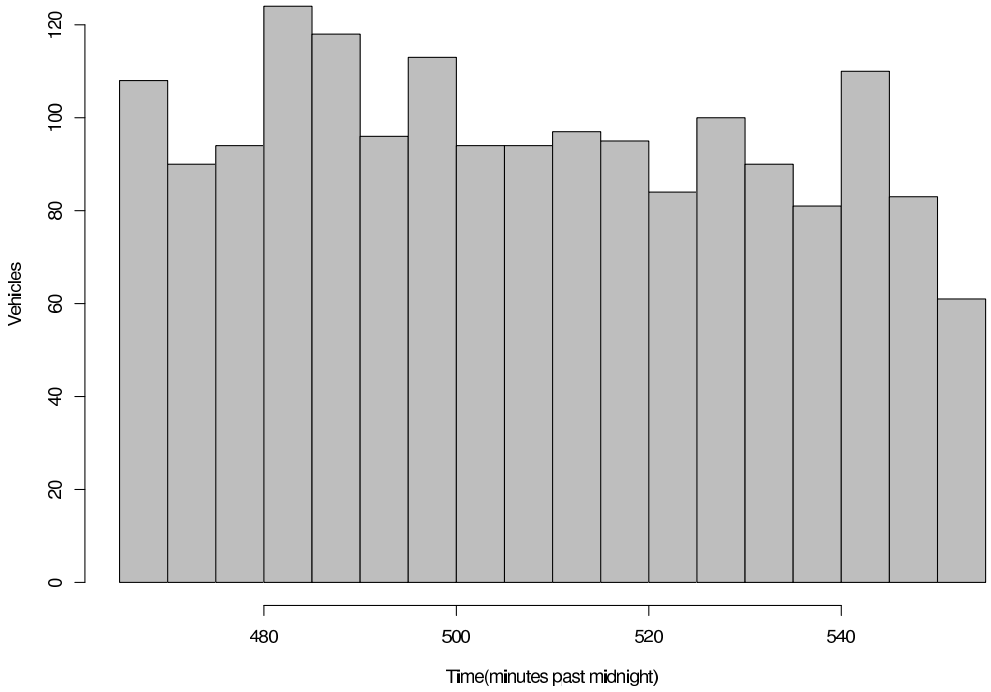

FIGURE D.34. Fishergate survey arrival times at site A 16/7/01.

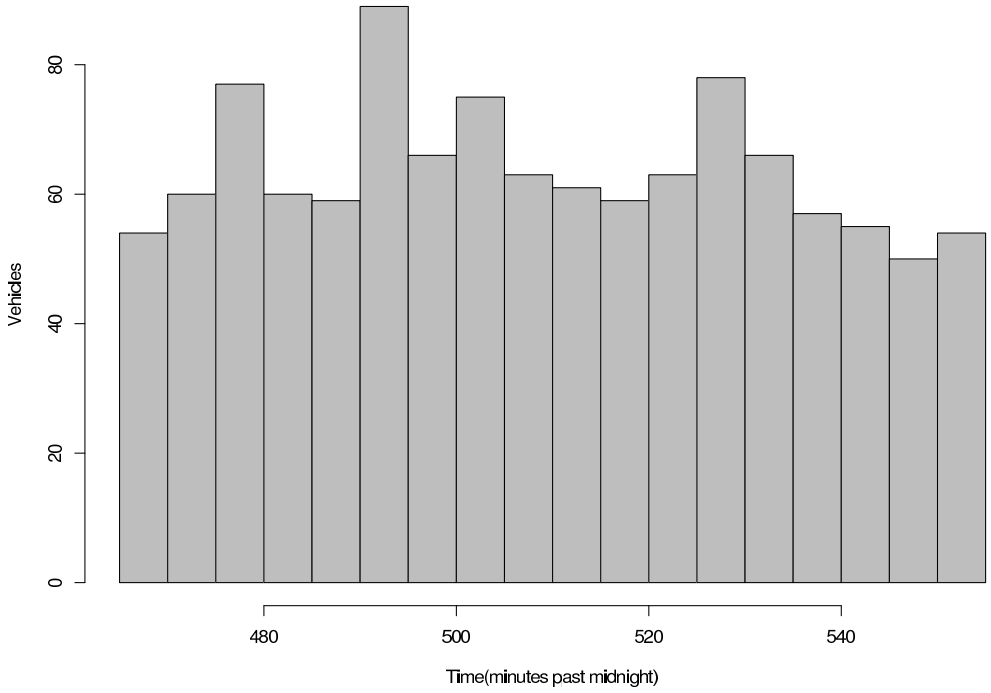

FIGURE D.35. Fishergate survey arrival times at site D 27/6/01.

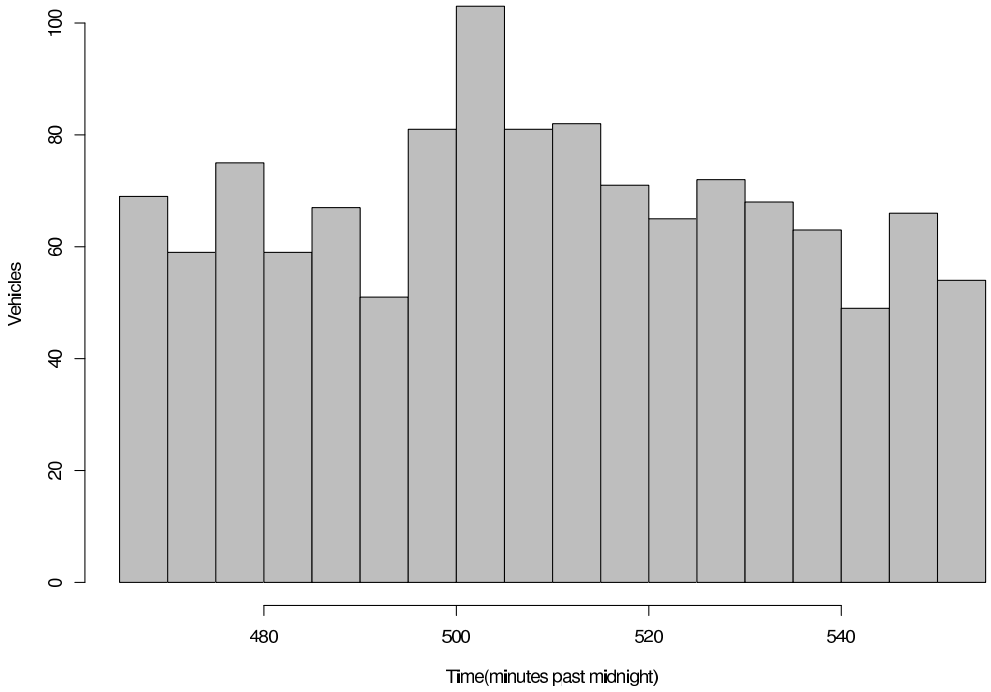

FIGURE D.36. Fishergate survey arrival times at site D 28/6/01.

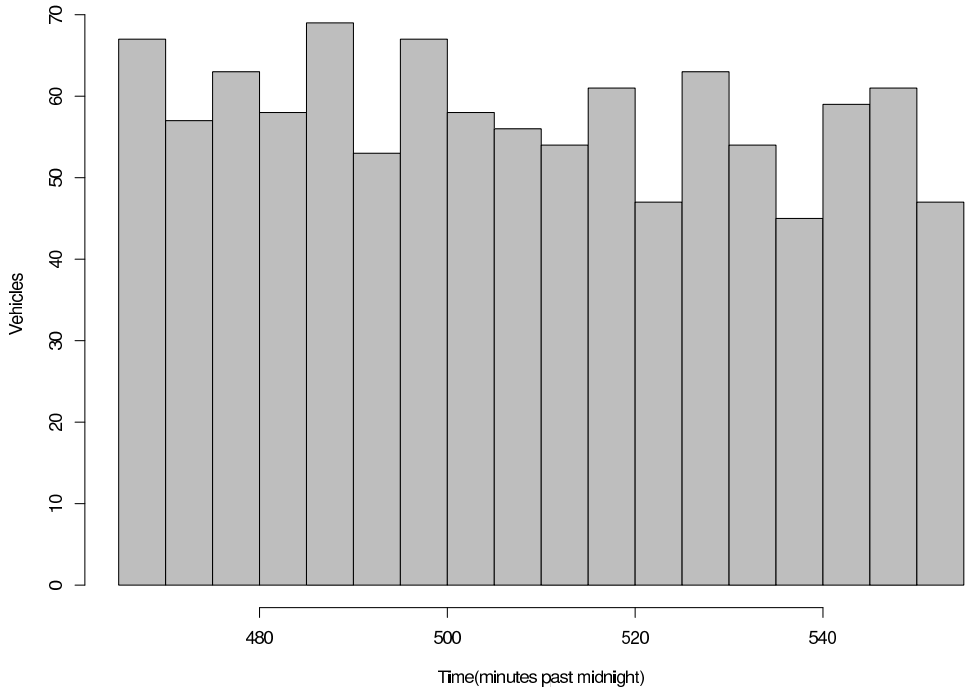

FIGURE D.37. Fishergate survey arrival times at site D 3/7/01.

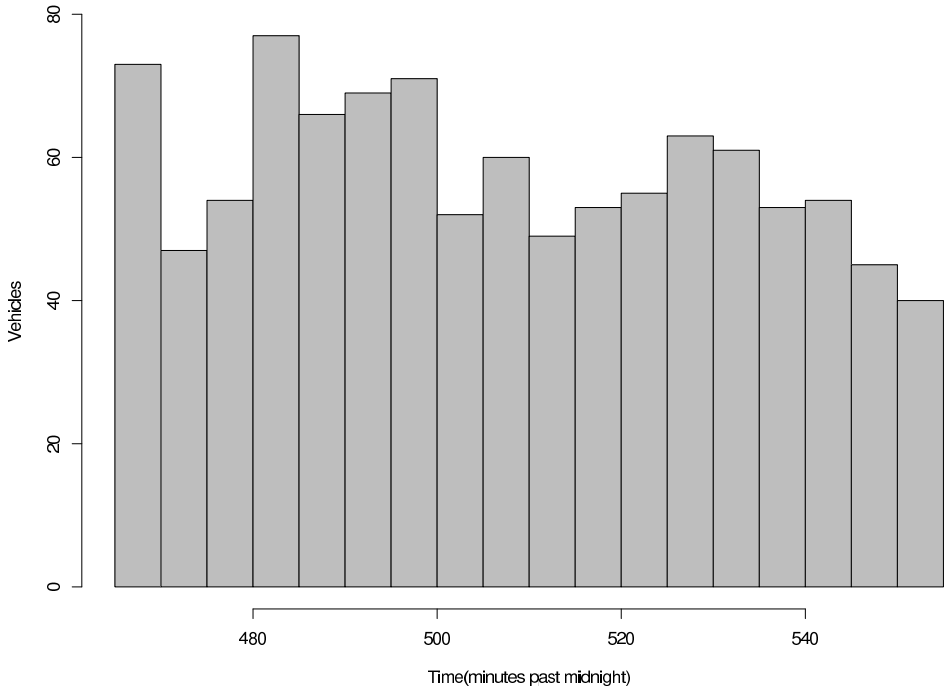

FIGURE D.38. Fishergate survey arrival times at site D 4/7/01.

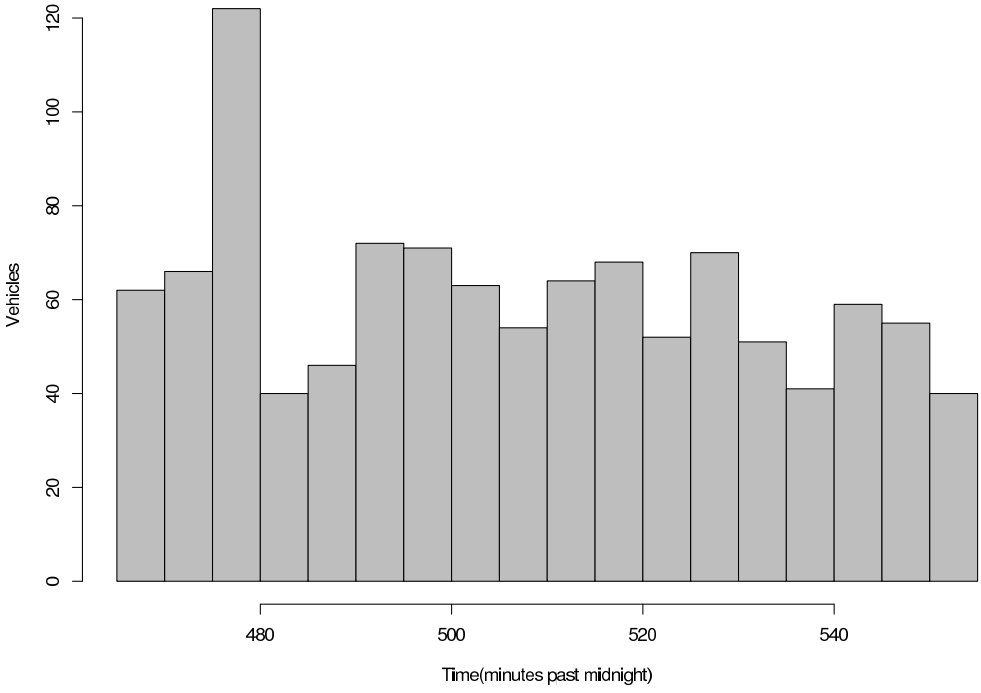

FIGURE D.39. Fishergate survey arrival times at site D 16/7/01.

APPENDIX E

Source Code For Licence Plate Matching

The following source code is used to execute the matching algorithm described in Chapter 4.

Header files:

- `match.h`
- `combine.h`
- `evaluate.h`
- `hoursmins.h`
- `matchdraw.h`
- `parsestring.h`
- `poly.h`
- `readplates.h`

Source code:

- `match.cpp`
- `combine.cpp`
- `evaluate.cpp`
- `hoursmins.cpp`
- `matchdraw.cpp`
- `matchimpl.cpp`
- `parsestring.cpp`
- `poly.cpp`
- `readplates.cpp`

E.1. match.h

```
#ifndef MATCH_H
#define MATCH_H
#include <vector>
#include <iostream>
using namespace std;

// Class for types of match and transversal of all types

// Definitions for matchClass and matchTrans
// Implementations are in matchimpl.cpp
// Template based implementations are in this header (necessary
// as of gcc 2.95 and earlier)

class matchClass
{                               //  See papers on the subject for full description
    //  Class represents a type of match
  public:
    matchClass ():n (0), x (0), height (0)
    {
    }
    // Default constructor - empty - no sites

    matchClass (int width):n (width), x (width, 1), height (1)
    {
    }
    // Constructor of (1, 1, .... 1) True match for "n" sites
    matchClass (const matchClass & match, int add);
    // Add one onto existing class
    bool isValid (void);        // Check if this is a valid match class
    bool lexLT (const matchClass &) const;      // Lexicographical less than.
    const int getWidth () const
    {
        return n;
    }
    const int getHeight () const
    {
        return height;
    }
    const int getElement (const int j) const
    {
        return x[j];
    }

  private:

    int n;                      // Number of elements in this match
    vector < int >x;            // Parts of our match
    int height;                 // Height of match (no of distinct elements)
    int calcHeight (void);      // Calculate height of matching class
    friend ostream & operator << (ostream & os,
                                  const matchClass & right);
```

```
    friend bool operator == (const matchClass & left,
                             const matchClass & right);
    friend bool operator != (const matchClass & left,
                             const matchClass & right)
    {
        return !(left == right);
    }
    friend bool operator > (const matchClass & left,
                            const matchClass & right);
    friend bool operator < (const matchClass & left,
                            const matchClass & right);
};

template < class T > class matchTrans
{
    // Transversal of all possible matches for n sites
  public:
    matchTrans (int n);
    // Construct the n'th transversal
    matchTrans (int n, int classes, vector < T > xnew):sites (n),
        noClasses (classes), x (xnew)
    {
    };
    T & getElement (int i)      // Return i'th element
    {
        return (x[i]);
    }
    static int Stirling (int n, int k); // Stirling no S(n,k)
    static int calcClasses (int n);     // No of classes in n'th transversal
    int getSites () const
    {
        return sites;
    }
    int getNoClasses () const
    {
        return noClasses;
    }
    matchTrans nextMatch ();    // Return the next transveral up
    int countHeight (int h);    // Return the number of transversal
    // Elements with height h
    void drawTrans (bool swapAxes = false, int xWid = 140, int yWid = 180);     // Draw a
    // Including arrows - using psfig.  Only defined for
    // matchDraw type transversals.

  private:
    int sites;                  // Number of sites for this transversal
    int noClasses;              // Number of matching classes in transversal
    vector < T > x;             // Classes in transversal
    template < class C > friend ostream & operator <<
        (ostream & os, matchTrans < C > right);
};
```

```
template < class T > matchTrans < T >::matchTrans (int n)
{
    if (n < 1)
      {
          noClasses = 0;
          sites = 0;
          x = vector < T > (0);
          return;
      }
    if (n == 1)
      {
          noClasses = 1;
          sites = 1;
          x = vector < T > (1, 1);      // Set up a vector with 1 match
          return;               // The (1) Class
      }
    matchTrans Mprev = matchTrans < T > (n - 1);

    // This duplicates code in nextMatch - horrible (but seemingly
    // unavoidable in C++)

    sites = n;
    noClasses = calcClasses (n);
    x = vector < T > (noClasses);

    int k = 0;
    for (int i = 0; i < Mprev.getNoClasses (); i++)
      {
          T z = Mprev.getElement (i);
          for (int j = 1; j <= z.getHeight () + 1; j++)
            {
                x[k] = T (z, j);
                k++;
            }
      }
}

template < class T > matchTrans < T > matchTrans < T >::nextMatch ()
{
    int newnoSites = getSites () + 1;
    int newnoClasses = calcClasses (newnoSites);
    vector < T > newx (newnoClasses, 0);
    int k = 0;
    for (int i = 0; i < getNoClasses (); i++)
      {
          T z = getElement (i);
          for (int j = 1; j <= z.getHeight () + 1; j++)
            {
                newx[k] = T (z, j);
                k++;
            }
```

```
      }
    return matchTrans < T > (newnoSites, newnoClasses, newx);
}

template < class T > int matchTrans < T >::calcClasses (int n)
// Count the number of classes in a particular type of match
// Currently done with Stirling nos - replace when I find a
// more efficient method
{
    int classCount = 0;
    for (int k = 1; k <= n; k++)
      {
          classCount += Stirling (n, k);
      }
    return classCount;

}

template < class T > int matchTrans < T >::Stirling (int n, int k)
{
    if (k <= 1 || k >= n)
        return 1;
    if (n <= 1)
        return 1;
    return (Stirling (n - 1, k - 1) + k * Stirling (n - 1, k));
}

template < class T > ostream & operator << (ostream & os,
                                            matchTrans < T > right)
// Output for match class
{

    for (int i = 0; i < right.noClasses; i++)
      {
          cout << right.x[i] << endl;
      }
    return os;
}

#endif
```

E.2. combine.h

```
#ifndef _COMBINE_H
#define _COMBINE_H
#include "readplates.h"
#include "match.h"
#include <vector>
#include <string>
#include <map>
// Class for manipulating combinations of platelists

// implementations are in combine.cpp

namespace licencePlates
{

    class allMatches
    {                           // Get all matches for a particular plate in a list
      public:
        void addMatch (const int &j)
        {
            matches.push_back (j);
        }
        const int noMatches () const
        {
            return matches.size ();
        }
        const int getElement (int i) const
        {
            return matches[i];
        }
      private:
          vector < int >matches;
    };

    class matchList
    {                           // Represents matches between two lists
      public:
        matchList ():noMatches (0), matches (0)
        {
        }
        // Construct a list of matches from two plate lists
        matchList (const plateList & list1, const plateList & list2);
        void addMatch (const int &i, const int &j)
        {
            matches[i].addMatch (j);
            noMatches++;
        }
        const int getNoMatchesAt (const int &i) const
        {
            return (matches[i].noMatches ());
        }
        const int getMatchAt (const int &i, const int &j) const
```

```
    {
        return (matches[i].getElement (j));
    }
    const allMatches & getElement (const int &i) const
    {
        return matches[i];
    }
    const int getNoMatches () const
    {
        return noMatches;
    }
    const int getSize () const
    {
        return matches.size ();
    }
  private:
    int noMatches;
    vector < allMatches > matches;
};


class lexLT                 // Lexicographical less than functor for matchClass
{
  public:
    bool operator () (const matchClass & lhs,
                      const matchClass & rhs) const
    {
        return lhs.lexLT (rhs);
    };
};

class combinePlates
{                           // Represents all possible matches in a class
  public:
    void addList (const plateList & newList)
    {
        lists.push_back (newList);
    }
    void addFile (string fileName);
    plateList & getElement (int i)
    {
        return lists[i];
    }
    const matchList & getComboElement (const int i, const int j) const
    {
        return combos[which_match (i, j)];
    }
    const matchList & getMatches (int i, int j) const
    {
        return combos[which_match (i, j)];
    }
    // Number of Observations at site n
```

```
        int noObs (int n)
        {
            return lists[n].size ();
        }
        int noLists ()
        {
            return lists.size ();
        }
        double cartesianProd (vector < int >&whichSites) const;
        void makeMatches ();    // Make all matches for this list
        int countMatchAll ();   // Count matches across all lists
        int countMatch (vector < int >&sites);
        // Count matches across sites on this list of sites.
      private:
        vector < plateList > lists;
        vector < matchList > combos;
        map < matchClass, int, lexLT > noRelaxedMatches;
        map < vector < int >, int >noSetMatches;
        const int which_match (int i, int j) const
        {
            return lists.size () * i + j;
        }
    };



}                               // end of namespace licencePlates

#endif
```

E.3. evaluate.h

```
#ifndef _EVALUATE_H
#define _EVALUATE_H
#include "match.h"
#include "poly.h"
#include "combine.h"
#include <vector>
#include <map>

double evaluate (licencePlates::combinePlates & obsList);
double evaluateSites (vector < polynomial * >&plist,
                      licencePlates::combinePlates & obslist,
                      vector < int >&whichSites);
double matchProb (int n);
#endif
```

E.4. hoursmins.h

```
#ifndef _HOURS_MINS_H
#define _HOURS_MINS_H

#include <iostream>
#include <string>
#include <sstream>
#include "parsestring.h"

// Class to represent times in hours and minutes

namespace hoursMins
{

    class timeError
    {
      public:
        timeError (const string & error):err (error)
        {
        }
        const string & getError ()
        {
            return err;
        }
      private:
          string err;
    };

    class basicTime
    {
      public:
        basicTime (int initHrs = 0, int initMins =
                   0):hrs (initHrs), mins (initMins)
        {
        }
        basicTime (const string & str);
        int timeDiff (const basicTime & t2);
        const int getHrs () const
        {
            return hrs;
        }
        const int getMins () const
        {
            return mins;
        }
        const string getTimeStr () const;
      private:
        int hrs;
        int mins;
        friend ostream & operator << (ostream & os,
                                      const basicTime & tm);
    };
```

```
}                                   // end of namespace hoursMins

#endif
```

E.5. matchdraw.h

```
#ifndef MATCHDRAW_H
#define MATCHDRAW_H
#include "match.h"
#include <vector>

// Class for matching classes which are to be drawn
// Implementation of functions is in matchdraw.cpp

class matchDraw
{

  public:
    matchDraw ():x (0), y (0), match ()
    {
    };
    matchDraw (int n):x (0), y (0), match (n), nodeId (nodeIdCount++)
    {
    }
    matchDraw (matchDraw & currMatch, int add):x (currMatch.getX ()),
        y (currMatch.getY ()), match (currMatch.getMatch (), add),
        nodeId (nodeIdCount++)
    {
    }
    void setXY (int x1, int y1)
    {
        x = x1;
        y = y1;
    }
    bool swapedAxes () const
    {
        return swapAxes;
    }
    int getX () const
    {
        return x;
    }
    int getY () const
    {
        return y;
    }
    int getId () const
    {
        return nodeId;
    }
    const int getWidth () const
    {
        return match.getWidth ();
    }
    const int getHeight () const
    {
        return match.getHeight ();
```

```
    }
    const int getElement (int i) const
    {
        return match.getElement (i);
    }
    const matchClass & getMatch () const
    {
        return match;
    }
  private:
    int x, y;                   // Co-ords when drawn
    matchClass match;           // Details of match type
    static bool swapAxes;       // Swap axes when printing
    int nodeId;                 // Unique node identifier
    static int nodeIdCount;     // Node Identifier counter

    friend ostream & operator << (ostream & os, matchDraw & right);
    friend void setAxisRotation (bool swap)
    {
        swapAxes = swap;
    }
    friend bool operator == (const matchDraw & left,
                             const matchDraw & right);
    friend bool operator > (const matchDraw & left,
                            const matchDraw & right);
    friend bool operator < (const matchDraw & left,
                            const matchDraw & right);
};

#endif
```

E.6. parsestring.h

```
#ifndef _PARSESTRING_H
#define _PARSESTRING_H
#include <string>
#include <vector>

using namespace std;

namespace parseString
{

    class stringTokeniser
    {
// Sort of like the Java version - give it a string and a split thingy
      public:
        stringTokeniser (const string & input, const string & split =
                         " \t\n\r");
        const int getNoTokens () const
        {
            return tokens.size ();
        }
        const string & getElement (int i) const
        {
            return tokens[i];
        }
        const int getPos (int i) const
        {
            return strpos[i];
        }
      private:
          vector < string > tokens;
          vector < int >strpos;
    };

}                               // End of namespace parseString

#endif
```

E.7. poly.h

```
#ifndef _POLY_H
#define _POLY_H
#include <vector>
#include "match.h"
#include "combine.h"

class polynomial;               //forward declaration - class declaration later in file

class polyElement
{
  public:
    virtual ~ polyElement ()
    {
    };
    virtual void putTo (ostream & os) const;
    virtual bool isExpansible () = 0;
    virtual vector < polyElement * >getExpansion (matchTrans <
                                                  matchClass > &Mn,
                                                  vector <
                                                  polyElement * >elems)
    {
        vector < polyElement * >pv (0);
        return pv;
    }
    int getFactor () const
    {
        return mult;
    }
    void addFactor (int f)
    {
        mult += f;
    }
    void setFactor (int f)
    {
        mult = f;
    }
    int getNoSites () const
    {
        return noSites;
    }
    int getProb () const
    {
        return prob;
    }
    virtual bool equals (polyElement * pe)
    {
        return false;
    }
    virtual double lhsEvaluate (vector < polynomial * >&plist,
                                licencePlates::combinePlates & obslist,
                                vector < int >&whichSites) const = 0;
```

```
    virtual double rhsEvaluate (vector < polynomial * >&plist,
                                licencePlates::combinePlates & obslist,
                                vector < int >&whichSites) const = 0;
  protected:
    int prob;                   // Probability multiplier
    int noSites;                // Number of sites in match
    int mult;                   // Multiplier
    bool polyAddTo (vector < polyElement * >elems, polyElement * match);
  private:


    friend ostream & operator << (ostream & os, const polyElement & pe);
};

//  This match is a true match on censored data - observable
//  X(M(T),C(S))
class matchTrue:public polyElement
{
  public:
    matchTrue (int n, int p = 1, int fact = 1)
    {
        noSites = n;
        prob = p;
        mult = fact;
    }
     ~matchTrue ()
    {
    }
    bool isExpansible ()
    {
        return false;
    }
    void putTo (ostream & os) const;
    bool equals (polyElement * pe);

    double lhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const;
    double rhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const;
  private:

    friend ostream & operator << (ostream & os, const matchTrue & pe);

};

// This match is an exact match of a particular type
// It expands into real matches of all lower types
class exactMatch:public polyElement
{
  public:
```

```
    exactMatch (int n, matchClass & match, int p = 1, int fact =
                1):mc (match)
    {
        noSites = n;
        prob = p;
        mult = fact;
    }
     ~exactMatch ()
    {
    }
    bool isExpansible ()
    {
        return true;
    }
    vector < polyElement * >getExpansion (matchTrans < matchClass > &Mn,
                                          vector <
                                          polyElement * >elems);
    void putTo (ostream & os) const;
    bool equals (polyElement * pe);
    matchClass & getMatchClass ()
    {
        return mc;
    }
    double lhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const
    {
        cout << "ERROR!  This should never be called!\n";
        return 0;
    }
    double rhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const
    {
        cout << "ERROR!  This should never be called!\n";
        return 0;
    }
  private:
      matchClass mc;


    friend ostream & operator << (ostream & os, const exactMatch & pe);
};

class matchByParts:public polyElement
{
  public:
    matchByParts (int n, matchClass & match, int p = 1,
                  int fact = 1):mc (match)
    {
        noSites = n;
        prob = p;
```

```
        mult = fact;
    }
     ~matchByParts ()
    {
    }
    bool isExpansible ()
    {
        return false;
    }
    void putTo (ostream & os) const;
    bool equals (polyElement * pe);
    matchClass & getMatchClass ()
    {
        return mc;
    }
    double lhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const;
    double rhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const;
  private:

    matchClass mc;
    friend ostream & operator << (ostream & os,
                                  const matchByParts & pe);

};


class polynomial
{

  public:
    polynomial ():elements (0), noSites (0), Mn (1)
    {
    };
    polynomial (int n);
    ~polynomial ();
    void putTo (ostream & os) const;
    int getNoSites () const
    {
        return noSites;
    }
    int length () const
    {
        return elements.size ();
    }
    int noTrans () const
    {
        return Mn.getNoClasses ();
    }
```

```
    const polyElement *getElement (int i) const
    {
        return elements[i];
    }
  private:
      vector < polyElement * >elements;
    int noSites;
    matchTrans < matchClass > Mn;

    polynomial (const polynomial &);
    const polynomial & operator= (const polynomial &);

    void expandPoly ();         // Expand terms of polynomial
    void gatherPoly ();         // Gather like terms
    void deleteElement (polyElement * e);
    void gatherElement (polyElement * e);
    friend ostream & operator << (ostream & os, const polynomial & p);
};



#endif
```

E.8. readplates.h

```
#ifndef _READPLATES_H
#define _READPLATES_H
#include <iostream>
#include <string>
#include <vector>
#include <ctime>
//#include <strstream>
#include <sstream>
#include "hoursmins.h"

// Implementations of functions are found in readplates.cpp


namespace licencePlates
{

    class plateReadError
    {
      public:
        plateReadError (const string & err, int line =
                        noLineNo):errName (err), lineNo (line)
        {
        }
        const string & getError () const
        {
            return errName;
        }
        const int getLineNo () const
        {
            return lineNo;
        }
        const bool validLineNo () const
        {
            return (lineNo != noLineNo);
        }
      private:
        static const int noLineNo = -1;
        string errName;
        int lineNo;
        friend ostream & operator << (ostream & os,
                                      const plateReadError & err);
    };

    class plate
    {                           // Class for a single licence plate observation
      public:
        //plate(): reg(0), myTime(0), comment(0) {}
        plate (string const &raw);      //  Construct a plate from a string
        const string & getReg () const
        {
            return reg;
```

```
    }
    const string & getComment () const
    {
        return comment;
    }
    const int getHrs () const
    {
        return myTime.getHrs ();
    }
    const int getMins () const
    {
        return myTime.getMins ();
    }
    const string getTimeStr () const
    {
        return myTime.getTimeStr ();
    }
  private:
      string reg;           // Registration part
      hoursMins::basicTime myTime;
    string comment;         // Extra notes on this plate (if any)

    friend ostream & operator << (ostream & os,
                                  const plate & thisPlate);
    friend bool operator == (const plate & left,
                             const plate & right);
};

class plateList
{                           // List of information from licence plates
  public:
    plateList ():noVehicles (0)
    {
    }
    plateList (string const &fname);        // Construct a list from a file
    static const int maxLineLen = 1000;
    static const int lineTypeInfo = 1;
    static const int lineTypeComment = 2;
    static const int lineTypeData = 3;
    const string & getName () const
    {
        return name;
    }
    const int getNoVehicles () const
    {
        return noVehicles;
    }
    const int size () const
    {
        return list.size ();
    }
    plate & getElement (int &i)
```

```
        {
            return list[i];
        }
        const plate & getElement (int i) const
        {
            return list[i];
        }
        void clearList ()
        {
            list.clear ();
        }
      private:                  // Data first
        vector < plate > list;  // All vehicle plates
        int noVehicles;         // No. of vehicles (from #INFO line)
        string name;            // Name of list used
        // Private functions
        void tidyup ();         // Deal with a failed constructor
        static int parseInfoLine (string);      // Parse a #INFO line
        static int typeOfLine (string); // Parses a data line to determine type
        friend ostream & operator << (ostream & os,
                                      const plateList & plates);
    };

}                               //End of licenceplate namespace

#endif
```

E.9. match.cpp

```
#include "match.h"
#include "matchdraw.h"
#include "readplates.h"
#include "hoursmins.h"
#include "combine.h"
#include "poly.h"
#include "evaluate.h"
#include <iostream>

using namespace licencePlates;

int
main (int argc, char **argv)
{
    combinePlates combos;
    if (argc < 2)
      {
          cout << "Usage match [files]" << endl;
          return -1;
      }
    try
    {
        for (int i = 1; i < argc; i++)
          {
              combos.addFile (argv[i]);
          }
    }
    catch (plateReadError e1)
    {
        cerr << e1;
        return -1;
    }


    //cerr << "Making matches" << endl;
    combos.makeMatches ();
    cout << combos.countMatchAll () << " ";
    //cout << "Counted " << combos.countMatchAll() << " Matches " << endl;
    //cout << combos.getElement(2);
    //cout << list << endl;
    //polynomial poly(5);
    //cout << poly << endl;
    cout << evaluate (combos) << endl;
    return 0;
}
```

E.10. combine.cpp

```
#include "combine.h"

// Implementations from class combine
namespace licencePlates
{

    void combinePlates::addFile (string fileName)
    {
        plateList list;
          try
        {
            plateList tmpList (fileName);
              list = tmpList;
        }
        catch (plateReadError e1)
        {
            throw (e1);
        }
        addList (list);
    }

    double combinePlates::cartesianProd (vector < int >&whichSites) const
    {
        double product = 1.0;
        //cout << "Cart prod" << endl;
        for (unsigned int i = 0; i < whichSites.size (); i++)
          {
              product *= lists[whichSites[i]].size ();
              //cout << lists[whichSites[i]].size() << endl;
          }
        //cout << product << endl;
        return product;
    }

    void combinePlates::makeMatches ()
// make all matches for a set of lists
    {
        int noLists = lists.size ();
        combos = vector < matchList > (noLists * noLists);
        for (int i = 0; i < noLists; i++)
          {
              for (int j = i + 1; j < noLists; j++)
                {
                    //cout << "Matching " << i << " and " << j << endl;

                    matchList newList (lists[i], lists[j]);
                    combos[which_match (i, j)] = newList;
                    //cout << "Match " << i << " " << j << " " <<
                    //  combos[which_match(i,j)].getNoMatches() << endl;
                }
          }
```

```
    }

    int combinePlates::countMatchAll ()
// Count the matches across all lists in the set
    {
        vector < int >list (noLists ());
        for (int i = 0; i < noLists (); i++)
            list[i] = i;
        return countMatch (list);
    }

    int combinePlates::countMatch (vector < int >&sites)
// Count the number of matches across the sites listed in the vector
    {
        map < vector < int >, int >::iterator findPlate;
        findPlate = noSetMatches.find (sites);
        if (findPlate != noSetMatches.end ())
          {
              //cout << "Matches for ";
              //for (unsigned int i= 0; i < sites.size(); i++) {
              //    cout << sites[i] << "  ";
              //}
              //cout << noSetMatches[sites] << endl;
              return noSetMatches[sites];
          }
        int count = 0;
        int noSites = sites.size ();
        if (noSites < 2)
          {
              cerr << "Problem in countMatch" << endl;
              return 0;
          }
        vector < const matchList *>matches (noSites - 1);
        for (int i = 0; i < noSites - 1; i++)
          {
              matches[i] = &getComboElement (sites[i], sites[i + 1]);
          }                     //matches[i] is the list of all matches between the i'th
        // i+1 th site

        vector < int >level (noSites - 1, 0);
        vector < int >part (noSites - 1, 0);
        int depth = 0;          // Depth at which we are travesing the match tree

        //Insanely complex traverse of web of matches --- efficient
        // More so than nicer recursive code would be
        for (int i = 0; i < matches[0]->getSize (); i++)
          {
              if (matches[0]->getNoMatchesAt (i) == 0)
                  continue;
              level[0] = i;
              part[0] = 0;
              depth = 0;
```

```
                while (1)
                  {

                      if (depth == noSites - 2)
                        {
                            count +=
                                matches[depth]->
                                getNoMatchesAt (level[depth]);
                            level[depth] = 0;
                            part[depth] = 0;
                            depth--;
                            if (depth < 0)
                                break;
                            continue;
                        }
                      if (matches[depth]->getNoMatchesAt (level[depth])
                          > part[depth])
                        {
                            level[depth + 1] = matches[depth]->getMatchAt
                                (level[depth], part[depth]);
                            part[depth + 1] = 0;
                            part[depth]++;
                            depth++;
                            continue;
                        }
                      part[depth] = 0;
                      level[depth] = 0;
                      depth--;
                      if (depth < 0)
                          break;
                  }
            }
        noSetMatches[sites] = count;
        return count;
    }

    matchList::matchList (const plateList & list1,
                          const plateList & list2)
// Find all the matches between list one and list two
    {
        noMatches = 0;
        //cout << "Matching" << endl;
        matches = vector < allMatches > (list1.size ());
        for (int i = 0; i < list1.size (); i++)
          {
              for (int j = 0; j < list2.size (); j++)
                {
                    if (list1.getElement (i) == list2.getElement (j))
                      {
                          addMatch (i, j);
                          //cout << "Plate " << list1.getElement(i)
                          //    << " matches " << list2.getElement(j) << endl;
```

```
                    }
                }
            }
        //cout << list1.getName() << " + " << list2.getName () << " " <<
        //  noMatches << endl;
    }


}                               // end of namespace
```

E.11. evaluate.cpp

```
#include "evaluate.h"
#include <math.h>

using namespace licencePlates;

double
evaluate (combinePlates & obslist)
// Evaluate matches on this list
{
    vector < polynomial * >plist (obslist.noLists ());
    vector < int >whichSites (obslist.noLists ());

    //cerr << "Generating equations" << endl;
    for (int i = 0; i < obslist.noLists (); i++)
      {
          plist[i] = new polynomial (i + 1);

          whichSites[i] = i;
      }
    //cout << (*plist[obslist.noLists()-1]) << endl;
    //cerr << "Generating matches" << endl;
    double count = evaluateSites (plist, obslist, whichSites);

    for (int i = 0; i < obslist.noLists (); i++)
      {
          delete (plist[i]);
      }
    return count;
}

double
evaluateSites (vector < polynomial * >&plist,
               combinePlates & obslist, vector < int >&whichSites)
{
    double lhsCount = 1;
    double rhsCount = 0;
    int noSites = whichSites.size ();

    if (noSites < 1)
      {
          cerr << "Problem in evaluateSites" << endl;
          return 0;
      }
    if (noSites == 1)
      {
          double noObs = obslist.noObs (whichSites[0]);
          return noObs;
      }

    for (int i = 0; i < plist[noSites - 1]->length (); i++)
      {
```

```
            const polyElement *pe;
            pe = plist[noSites - 1]->getElement (i);
            //pe->putTo(cout);
            //cout << endl;
            double lhsAdd = pe->lhsEvaluate (plist, obslist, whichSites);
            lhsCount -= lhsAdd;
            //cout << "lhs count " << lhsAdd << endl;
            double rhsAdd = pe->rhsEvaluate (plist, obslist, whichSites);
            rhsCount += rhsAdd;
            //cout << "rhs count " << rhsAdd << endl;
        }
    //cout << "rhs " << rhsCount << " lhs " << lhsCount << endl;
    //cout << "Sites: ";
    // for (int i= 0; i < noSites; i++) {
    //cout << whichSites[i] << " ";
    //}
    //cout << rhsCount/lhsCount << endl;
    return rhsCount / lhsCount;
}

double
matchProb (int n)
// Function to calculate p(n)
{
    if (n <= 1)
        return 1;
    return pow (0.0001, (n - 1));
}
```

E.12. hoursmins.cpp

```
#include "hoursmins.h"
#include "parsestring.h"

using namespace std;

namespace hoursMins
{

    basicTime::basicTime (const string & str)
    {
        parseString::stringTokeniser strtok (str, ": \n\r\t");
        if (strtok.getNoTokens () != 2)
            throw timeError ("Unable to parse time " + str);
        string hourstr = strtok.getElement (0);
        string minstr = strtok.getElement (1);
          hrs = atoi (hourstr.c_str ());
          mins = atoi (minstr.c_str ());
    }

    const string basicTime::getTimeStr () const
// Return time as a string
    {
        std::ostringstream ost;
        ost.width (2);
        ost.fill ('0');
        ost << hrs << ":";
        ost.width (2);
        ost.fill ('0');
        ost << mins << std::ends;
        string timestr = ost.str ();
          return timestr;
    }

    ostream & operator << (ostream & os, const basicTime & tm)
    {
        string out = tm.getTimeStr ();
        os << out;
        return os;
    }

}                               // end of namespace hoursMins
```

E.13. matchdraw.cpp

```
#include <iostream>
#include "matchdraw.h"

bool
    matchDraw::swapAxes =
    false;
int
    matchDraw::nodeIdCount =
    1;

ostream & operator << (ostream & os, matchDraw & right)
// Output for match class
{
    os << "\\rput";
    if (right.swapedAxes ())
        os << "{90}";
    os << "(" << right.getX () << "," << right.getY () << ")";
    os << "{\\Rnode{N" << right.getId () << "}{(";

    for (int i = 0; i < right.getWidth (); i++)
      {
          os << right.getElement (i);
          if (i != right.getWidth () - 1)
              os << ",";
      }
    os << ")}}";

    return os;
}

bool
operator == (const matchDraw & left, const matchDraw & right)
{
    return (left.getMatch () == right.getMatch ());
}

bool
operator < (const matchDraw & left, const matchDraw & right)
{
    return (left.getMatch () < right.getMatch ());
}

bool
operator > (const matchDraw & left, const matchDraw & right)
{
    return (left.getMatch () > right.getMatch ());
}
```

E.14. matchimpl.cpp

```
#include <iostream>
#include "match.h"
#include "matchdraw.h"

bool
matchClass::isValid (void)
// Checks if this meets criterion for matching classes
{
    int h = 1;
    if (n == 0)
        return true;
    if (x[0] != 1)
        return false;
    for (int i = 1; i < n; i++)
      {
          if (x[i] < 1 || x[i] > h + 1)
              return false;
          if (x[i] > h)
              h++;
      }
    return true;
}

bool
matchClass::lexLT (const matchClass & rhs) const
{
    int rWidth = rhs.getWidth ();
    int lWidth = getWidth ();

    for (int i = 0; i < rWidth; i++)
      {
          if (i == lWidth)
              return true;
          if (getElement (i) < rhs.getElement (i))
              return true;
          if (getElement (i) > rhs.getElement (i))
              return false;
      }
    return false;

}

matchClass::matchClass (const matchClass & match, int add)
// Add one element on the end of a match class vector
{
    n = match.getWidth () + 1;
    x = vector < int >(n, 1);
    for (int i = 0; i < n - 1; i++)
        x[i] = match.getElement (i);
    x[n - 1] = add;
    height = calcHeight ();
```

```
}


int
matchClass::calcHeight (void)
{
    int h = 1;
    for (int i = 0; i < n; i++)
      {
          if (x[i] > h)
              h = x[i];
      }
    return h;
}

ostream & operator << (ostream & os, const matchClass & right)
// Output for match class
{
    cout << "(";

    for (int i = 0; i < right.n; i++)
      {
          cout << right.x[i];
          if (i != right.n - 1)
              cout << ",";
      }
    cout << ")";
    return os;
}

bool
operator == (const matchClass & left, const matchClass & right)
{
    int w;
    if ((w = left.getWidth ()) != right.getWidth ())
      {
          //cout << left << "!=" << right << endl;
          return false;
      }
    for (int i = 0; i < w; i++)
      {
          if (left.getElement (i) != right.getElement (i))
            {
                //cout << left << "!=" << right << endl;
                return false;
            }
      }
    // cout << left << "==" << right << endl;
    return true;
}

bool
```

```
operator > (const matchClass & left, const matchClass & right)
{
    int w;
    if ((w = left.getWidth ()) != right.getWidth ())
        return false;
    if (left == right)
        return false;
    for (int i = 0; i < w; i++)
      {
          for (int j = 0; j < w; j++)
            {
                if (i == j)
                    continue;
                if (left.getElement (i) == left.getElement (j) &&
                    right.getElement (i) != right.getElement (j))
                    return false;

            }
      }
    return true;
}

bool
operator < (const matchClass & left, const matchClass & right)
{

    return (right > left);
}

template <> void matchTrans < matchDraw >::drawTrans
    (bool swap, int xWid, int yWid)
// Draw out the Transversal including arrows.
{
    double redval = 0;
    double blueval = 0;
    double greenval = 0;

    setAxisRotation (swap);
    cout << "\\documentclass{article}" << endl;
    cout << "\\usepackage{epsfig}" << endl;
    cout << "\\usepackage{pstricks}" << endl;
    cout << "\\usepackage{pst-node}" << endl;
    cout << "\\begin{document}" << endl;
    cout << "%Latex Figure created by matching program" << endl;
    cout << "\\begin{figure}" << endl;
    cout << "\\begin{center}" << endl;
    cout << "\\unitlength=1mm" << endl;
    cout << "\\psset{unit=1mm}" << endl;
    cout << "\\psset{linewidth=0.5pt}" << endl;
    if (swap)
      {
          cout << "\\psset{swapaxes=true}" << endl;
```

```
      cout << "\\begin{picture}(" << yWid << "," << xWid << ")" <<
          endl;
  }
else
  {
      cout << "\\begin{picture}(" << xWid << "," << yWid << ")" <<
          endl;
  }
for (int i = sites; i > 0; i--)
  {
      int no = countHeight (i);
      int xMult = 1;
      for (int j = 0; j < noClasses; j++)
        {
            if (x[j].getHeight () != i)
                continue;
            x[j].setXY ((xMult) * xWid / (no + 1),
                        (sites - i) * yWid / (sites - 1));
            xMult++;
        }
  }
for (int i = 0; i < noClasses; i++)
  {
      cout << x[i] << endl;
  }
cout.precision (2);
cout.setf (ios::fixed, ios::floatfield);
for (int i = 0; i < noClasses; i++)
  {
      redval += 1.0;
      if (redval > 1.0)
        {
            redval = 0.0;
            greenval += 1.0;
            if (greenval > 1.0)
              {
                  greenval = 0.0;
                  blueval += 1.0;
                  if (blueval > 1.0)
                    {
                        blueval = 0.0;
                    }
              }
        }
      if (blueval == 1.0 && redval == 1.0 && greenval == 1.0)
        {
            blueval = 0.0;
            redval = 0.0;
            greenval = 0.0;
        }
      //cerr << "RGB: " << redval << " " << greenval << " " << blueval
      //  << endl;
```

```
            for (int j = 0; j < noClasses; j++)
              {
                  if (j == i)
                      continue;
                  if (x[i].getHeight () == x[j].getHeight () + 1
                      && x[i] > x[j])
                    {
                        cout << "\\newrgbcolor{tmpcolor}{" <<
                            redval << " " << greenval << " " <<
                            blueval << "}" << endl;


                        cout << "\\psset{linecolor=tmpcolor}" << endl;
                        cout << "\\ncdiag[arm= 5pt, angleA=";
                        if (swap == true)
                            cout << "90";
                        else
                            cout << "0";

                        cout << ", angleB=";
                        if (swap == true)
                            cout << "270";
                        else
                            cout << "180";


                        cout << "]{->}{N" << x[i].getId ()
                            << "}{N" << x[j].getId () << "}" << endl;
                    }

              }
      }
    cout << "\\psset{linecolor=black}" << endl;
    cout << "\\end{picture}" << endl;
    cout << "\\end{center}" << endl;
    cout << "\\end{figure}" << endl;
    cout << "\\end{document}" << endl;
}

template < class T > int matchTrans < T >::countHeight (int h)
{
    int no = 0;
    for (int i = 0; i < noClasses; i++)
      {
          if (x[i].getHeight () == h)
              no++;
      }
    return no;
}
```

E.15. parsestring.cpp

```
#include "parsestring.h"

namespace parseString
{

    stringTokeniser::stringTokeniser (const string & input,
                                      const string & split)
    {
        string tmp = "";
        bool intoken = false;
        int startpos = 0;
        for (unsigned int i = 0; i < input.size (); i++)
          {
              unsigned int j;
              for (j = 0; j < split.size (); j++)
                {
                    if (input[i] == split[j])
                      {
                          if (intoken == true)
                            {
                                intoken = false;
                                if (tmp.size () > 0)
                                  {
                                      tokens.push_back (tmp);
                                      strpos.push_back (startpos);
                                  }
                                tmp = "";
                            }
                          break;
                      }
                }
              if (j == split.size ())
                {
                    if (intoken == false)
                        startpos = i;
                    intoken = true;
                    tmp += input[i];
                }
          }
        if (tmp.size () > 0)
          {
              tokens.push_back (tmp);
              strpos.push_back (startpos);
          }
    }

}                               // End of namespace parseString
```

E.16. poly.cpp

```
#include "poly.h"
#include "evaluate.h"
#include <iostream>

ostream & operator << (ostream & os, const polyElement & p)
// printout operator for polyElement class;
{
    p.putTo (os);
    return os;
}


void
polyElement::putTo (ostream & os) const
{
    os << "Illegal Call to Base Class element " << endl;
}

bool
polyElement::polyAddTo (vector < polyElement * >elems,
                        polyElement * match)
{
    for (vector < polyElement * >::iterator i = elems.begin ();
         i != elems.end (); i++)
      {
          if (match->equals (*i))
            {
                (*i)->addFactor (match->getFactor ());
                return true;
            }
      }
    return false;
}

void
matchTrue::putTo (ostream & os) const
{
    if (mult == -1)
      {
          os << "-";
      }
    else if (mult != 1)
      {
          os << mult << ".";
      }
    if (prob != 1)
        os << "p(" << prob << ") ";
    os << "X(M" << getNoSites () << "(T),C(S" << getNoSites () << "))";
}

bool
```

```
matchTrue::equals (polyElement * mt)
{
    matchTrue *mtr = dynamic_cast < matchTrue * >(mt);
    if (mtr == NULL)
      {
          return false;
      }
    if (getProb () != mtr->getProb ())
        return false;
    return true;
}

double
matchTrue::lhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const
{

    return 0;
}

double
matchTrue::rhsEvaluate (vector < polynomial * >&plist,
                        licencePlates::combinePlates & obslist,
                        vector < int >&whichSites) const
{
    double noMatches = obslist.countMatch (whichSites);
    return noMatches * matchProb (getProb ()) * getFactor ();
}

void
exactMatch::putTo (ostream & os) const
{
    if (mult == -1)
      {
          os << "-";
      }
    else if (mult != 1)
      {
          os << mult << ".";
      }
    if (prob != 1)
        os << "p(" << prob << ") ";
    os << "X(" << static_cast < const matchClass > (mc) <<
        ",S" << getNoSites () << ")";
}

bool
exactMatch::equals (polyElement * mt)
{
    exactMatch *em = dynamic_cast < exactMatch * >(mt);
    if (em == NULL)
```

```
      {
          return false;
      }
    if (getProb () != em->getProb ())
        return false;
    if (getMatchClass () != em->getMatchClass ())
        return false;
    return true;
}

void
matchByParts::putTo (ostream & os) const
{
    if (mult == -1)
      {
          os << "-";
      }
    else if (mult != 1)
      {
          os << mult << ".";
      }
    if (prob != 1)
        os << "p(" << prob << ") ";
    os << "R(" << mc << ",S" << getNoSites () << ")";
    //os << "match class is " << mc;
}

bool
matchByParts::equals (polyElement * mt)
{
    matchByParts *mbp = dynamic_cast < matchByParts * >(mt);
    if (mbp == NULL)
      {
          return false;
      }
    if (getProb () != mbp->getProb ())
        return false;
    if (getMatchClass () != mbp->getMatchClass ())
        return false;
    return true;
}

double
matchByParts::lhsEvaluate (vector < polynomial * >&plist,
                           licencePlates::combinePlates & obslist,
                           vector < int >&whichSites) const
{
    if (mc.getHeight () == 1)
      {                         // height 1 is true match
          return matchProb (getProb ()) * getFactor ();
      }
    return 0;
```

```
}

double
matchByParts::rhsEvaluate (vector < polynomial * >&plist,
                           licencePlates::combinePlates & obslist,
                           vector < int >&whichSites) const
{
    //cout << "MBP height " << mc.getHeight() << " sites size " <<
    //  whichSites.size() << endl;
    if (mc.getHeight () == 1)
      {
          return 0;
      }
    int noSites = whichSites.size ();
    if (noSites == mc.getHeight ())
      {
          return matchProb (getProb ()) * getFactor () *
              obslist.cartesianProd (whichSites);
      }
    double mbp = 1;
    //putTo(cout);
    //cout << endl;
    for (int i = 1; i <= mc.getHeight (); i++)
      {
          vector < int >siteList;
          //cout << " at level " << i << " matches elements ";
          for (int j = 0; j < mc.getWidth (); j++)
            {

                if (mc.getElement (j) == i)
                  {
                      //cout << j << " ";
                      siteList.push_back (whichSites[j]);
                  }
            }

          double mult = evaluateSites (plist, obslist, siteList);
          mbp *= mult;
          //cout << " and has " << mult << " estimated matches" << endl;
      }
    return matchProb (getProb ()) * getFactor () * mbp;
}

vector < polyElement * >exactMatch::getExpansion (matchTrans <
                                                  matchClass > &Mn,
                                                  vector <
                                                  polyElement * >elems)
// exact match of a particular type expands into a relaxed match
// of a particular type
{
    vector < polyElement * >pv;
    matchByParts *
```

```
        mbp =
        new
        matchByParts (getNoSites (), getMatchClass (),
                      getProb (), getFactor ());
    pv.push_back (mbp);

    int
        nc =
        Mn.
        getNoClasses ();

    for (int i = 0; i < nc; i++)
      {
          //cout << "n = " << i << " out of " << nc << endl;
          matchClass & mClass = Mn.getElement (i);
          if (!(mClass < mc))
              continue;
          //cout << "Adding match of class " << mClass << endl;
          exactMatch *
              xmat =
              new
              exactMatch (getNoSites (), mClass, getProb (),
                          -getFactor ());
          if (polyAddTo (elems, xmat))
            {
                delete (xmat);
            }
          else
            {
                pv.push_back (xmat);
            }
          //cout << "Added match: ";
          //xmat->putTo(cout);
          //cout << endl;
      }
    return pv;
}

polynomial::polynomial (int n):
noSites (n), Mn (n)
//  Construct the polynomial for match transversal M_n
{
    matchTrue *mt = new matchTrue (n);
    elements.push_back (mt);
    int nc = Mn.getNoClasses ();
    exactMatch *me;
    matchClass *mc;
    for (int i = 0; i < nc; i++)
      {
          mc = &(Mn.getElement (i));
          if (mc->getHeight () == 1)    // Don't add the "true match" class
              continue;
```

```
          me = new exactMatch (n, (*mc), mc->getHeight (), -1);
          elements.push_back (me);
      }
    //putTo(cout);
    //cout << endl;
    expandPoly ();
}

polynomial::~polynomial ()
{
    int nel = length ();
    for (int i = 0; i < nel; i++)
      {
          //cout << "Deleted element " << i << endl;
          delete (elements[i]);
      }
}

void
polynomial::expandPoly ()
// Expand terms of polynomial
{
    //cout << "EXPANDING ONCE" << endl;
    //putTo(cout);
    //cout << endl;
    vector < polyElement * >addvect;
    for (vector < polyElement * >::iterator i = elements.begin ();
         i != elements.end (); i++)
      {
          if ((*i)->isExpansible ())
            {
                //cout << "Expanding:  ";
                //(*i)->putTo(cout);
                //cout << endl;
                addvect = (*i)->getExpansion (Mn, elements);    // Add expansion onto the
                delete (*i);    // Delete the memory saved for the vector
                elements.erase (i);     // And remove it from the vector
                for (size_t j = 0; j < addvect.size (); j++)
                  {
                      elements.push_back (addvect[j]);
                  }
                expandPoly ();  // Restart the expansion and leave this
                return;         // function
            }
      }
    //cout << "FINISHED EXPANSION" << endl;
}

void
polynomial::putTo (ostream & os) const
{
    os << "M" << getNoSites () << "(T,S" << getNoSites () << ") = ";
```

```
    for (int i = 0; i < length (); i++)
      {
          const polyElement *pe = getElement (i);
          if (i != 0)
            {
                if (pe->getFactor () > 0)
                    os << " +";
                else
                    os << " ";
            }
          os << (*pe);
      }


}

ostream & operator << (ostream & os, const polynomial & p)
//
{
    p.putTo (os);
    return os;
}
```

E.17. readplates.cpp

```
#include "readplates.h"
#include "parsestring.h"
#include <fstream>
#include <stdlib.h>

using namespace parseString;

namespace licencePlates
{

    ostream & operator << (ostream & os, const plateReadError & err)
// Output operator for plate errors - print error and line no.
    {
        os << err.getError ();
        if (err.validLineNo ())
          {
              os << " at line " << err.getLineNo ();
          }
        return os;
    }

    plate::plate (string const &raw)
// Construct licence plate information from raw string
    {
        stringTokeniser strtok (raw);
        if (strtok.getNoTokens () < 2)
            throw plateReadError ("Unable to read plate line " + raw);
        reg = strtok.getElement (0);
        //cout << "Read plate " << reg << endl;
        if (strtok.getNoTokens () >= 3)
          {
              //cout << "Comment starts at " << strtok.getPos(2) << endl;
              comment = raw.substr (strtok.getPos (2));
              //cout << "Read comment " << comment << endl;
          }
        else
          {
              comment = "";
          }
        string timestr = strtok.getElement (1);
        try
        {
            hoursMins::basicTime tm (timestr);
            myTime = tm;
        }
        catch (hoursMins::timeError e1)
        {
            throw plateReadError (e1.getError ());
        }
        // cout << "Read Time " << myTime.getTimeStr() << endl;
```

```
    }

    ostream & operator << (ostream & os, const plate & thisPlate)
    {
        string plateTime = thisPlate.getTimeStr ();
        os << thisPlate.getReg () << " " << plateTime << " " <<
            thisPlate.getComment ();
        return os;
    }


    ostream & operator << (ostream & os, const plateList & plates)
    {
        for (int i = 0; i < plates.size (); i++)
          {
              os << plates.getElement (i) << endl;
          }
        return os;
    }

    bool operator == (const plate & left, const plate & right)
    {
        if (left.getReg () == right.getReg ())
            return true;
        return false;
    }

    plateList::plateList (string const &fname)
// Construct a list of plates from a file
    {
        name = fname;
        ifstream readFile (fname.c_str ());
        if (!readFile)
          {
              throw plateReadError ("Unable to open file " + fname);
          }
        string readLine;
        bool infoSet = false;
        int lineNo = 1;
        while (getline (readFile, readLine))
          {
              //cout << lineNo << endl;
              //cout << typeOfLine(readLine) << endl;
              switch (typeOfLine (readLine))
                {
                case (plateList::lineTypeInfo):
                    if (infoSet == true)
                      {
                          tidyup ();
                          throw
                              plateReadError
                              ("Second #INFO line found in file " +
```

```
                                fname, lineNo);
                      }
                    infoSet = true;
                    noVehicles = parseInfoLine (readLine);
                    if (noVehicles <= 0)
                      {
                          tidyup ();
                          throw
                              plateReadError
                              ("Incorrect #INFO line found in file " +
                               fname, lineNo);
                      }
                    break;
                case (plateList::lineTypeComment):
                    // Ignore comments
                    break;
                case (plateList::lineTypeData):
                    try
                    {
                        plate newplate (readLine);
                        list.push_back (newplate);
                    }
                    catch (plateReadError e1)
                    {
                        tidyup ();
                        throw plateReadError (e1.getError () +
                                              " in file " + fname,
                                              lineNo);
                    }
                    break;
                default:
                    cout << "Error" << endl;
                    tidyup ();
                    throw plateReadError ("Unrecognised line in file " +
                                          fname, lineNo);

                }
              lineNo++;
          }
        if (infoSet == false)
          {
              tidyup ();
              throw plateReadError ("No #INFO line in file " + fname);
          }
    }

    void plateList::tidyup ()
// Clear anything necessary after constructor fails
    {
        clearList ();
    }
```

```
    int plateList::parseInfoLine (string input)
// Given that we have an input line, return the no of vehicles.
    {
        string::size_type i1 = input.find_first_of ("0123456789");
        if (i1 == string::npos) // Return 0 if there are no digits
            return 0;
        string num = input.substr (i1);
        int infonum = atoi (num.c_str ());
        return infonum;
    }

    int plateList::typeOfLine (string input)
// Returns lineTypeInfo, lineTypeComment or lineTypeData
    {
        if (input.size () == 0)
            return plateList::lineTypeComment;
        if ("#INFO" == input.substr (0, 5))
            return plateList::lineTypeInfo;
        if (input[0] == '#')
            return plateList::lineTypeComment;
        return plateList::lineTypeData;
    }



}                               // End of namespace licencePlates
```

Bibliography

[1] H. Z. Aashtiani and T. L. Magnanti. Equilibria on a congested transportation network. *SIAM Jour. Alg. Disc. Meth.*, 2(3), 1981.

[2] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch. Self-similarity and long-range dependence through the wavelet lens. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory and Applications of Long-Range Dependence*, pages 526–556. Birkhäuser, 2003.

[3] R. J. Adler, R. E. Feldman, and M. S. Taqqu, editors. *A Practical Guide to Heavy Tails*. Birkhäuser, 1998.

[4] D. K. Arrowsmith, R. J. Mondragón, and J. M. Pitts. Chaotic maps for traffic modelling and queueing performance analysis. *Performance Analysis*, 42:223–240, 2001.

[5] Automobile Association. Automobile Association guide to licence plate formats: `www.theaa.com/allaboutcars/drive_plates_history.html`, 2003.

[6] J.-M. Bardet, G. Lang, G. Oppenheim, A. Phillipe, S. Stoev, and M. S. Taqqu. Semi-parametric estimation of the long-range dependence parameter: A survey. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory and Applications of Long-Range Dependence*, pages 557–577. Birkhäuser, 2003.

[7] J. J. Bates. Time period choice modelling: a preliminary review. Report for UK Deparment for Transport. Available online at: `www.dft.gov.uk/stellent/groups/dft_transstrat/documents/page/dft_transstrat_504847-11.hcsp`, 1996.

[8] R. Batley and R. G. Clegg. Driver route choice and departure time choice: The models and the evidence. Paper presented at Universities' Transport Studies Group, University of Oxford. Available online at: `gridlock.york.ac.uk/route/docs/utsg2001.doc`, January 2001.

[9] R. Batley, T. Fowkes, D. Watling, G. Whelan, A. Daly, and E. Hato. Models for analysing route choice. Paper presented at Universities' Transport Studies Group, University of Oxford, January 2001.

[10] M. Ben-Akiva. *Structure of passenger travel demand models*. PhD thesis, Department of Civil Engineering, MIT, Cambridge, Mass., USA, 1973.

[11] M. Ben-Akiva and M. Bierlaire. Discrete choice methods and their applications to short-term travel decisions. In R. W. Hall, editor, *Handbook of Transportation Science*. Kluwer Academic Publishers, 1999.

[12] M. Ben-Akiva, A. De Palma, and P. Kanaroglou. Dynamic model of peak period traffic congestion with elastic arrival rates. *Transpn. Sci.*, 20(2):164–181, 1986.

[13] M. Ben-Akiva, A. De Palma, and I. Kaysi. Dynamic network models and driver information systems. *Transpn. Res. A*, 25(5):251–266, 1991.

[14] J. Beran. A test of location for data with slowly decaying serial correlations. *Biometrika*, 76:261–269, 1989.

[15] J. Beran. *Statistics For Long-Memory Processes*. Chapman and Hall, 1994.

[16] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Trans. on Communications*, 43:1566–1579, 1995.

[17] R. J. Bhansali and P. S. Kokoszka. Predictions of long-memory time series. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 355–367. Birkhäuser, 2003.

[18] C. Bhat. Recent methodological advances relevant to activity and travel behavior analysis. Resource paper prepared for the IATBR Conference, Austin, Texas, September 1997.

[19] P. Bonsall, F. Montgomery, and C. Jones. Deriving the constancy of traffic flow composition from vehicle registration data. *Traf. Eng. & Cont.*, 25(6/7):386–391, 1984.

[20] M. Borella and G. Brewster. Measurement and analysis of long-range dependent behavior of internet packet delay. In *Proc. IEEE INFOCOM*, pages 497–504, 1998.

[21] L. E. J. Brouwer. Uber eineindeutige, stetige transformationen von flächen in sich. *Math. Ann.*, 67:176–180, 1910.

[22] J. E. Burrell. Multiple route assignment and its application to capacity restraint. In *Fourth International Symposium on the Theory of Traffic Flow, Karlsruhe*, 1968.

[23] S. Cairns, H-K. Carmen, and P. Goodwin. *Traffic Impact of Highway Capacity Reductions: Assessment of the Evidence.* Landor Publishing, 1998.

[24] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun. Internet traffic tends toward Poisson and independent as the load increases. In C. Holmes, D. Denison, M. Hansen, B. Yu, and B. Mallick, editors, *Nonlinear Estimation and Classification*. Springer, 2002.

[25] N. S. Cardell and F. C. Dunbar. Measuring the societal impact of automobile downsizing. *Transpn. Res. A*, 14(5/6):432–434, 1980.

[26] E. Cascetta, A. Nuzzolo, F. Russo, and A. Vitetta. A modified logit route choice model overcoming path overlapping problems. specification and some calibration results for

interurban networks. In J. B. Lesort, editor, *Proceedings of the 13th International Symposium on Transportation and Traffic Theory, Lyon, July 1996*, pages 697–711. Pergamon, 1996.

[27] C. Chu. *Structural issues and sources of bias in residential location and travel mode choice models*. PhD thesis, Department of Civil Engineering, Northwestern University, USA, 1981.

[28] S. D. Clark. An exercise in matching number plates: Using a maximum likelihood approach. Available online at:
`gridlock.york.ac.uk/route/docs/maxlike.doc`, 2001.

[29] S. D. Clark. Report on park row before surveys. Available online at:
`gridlock.york.ac.uk/route/docs/parkrow.pdf`, 2001.

[30] S. D. Clark and R. G. Clegg. Assessing the impact of United Kingdom fuel protest actions in the city of York, United Kingdom. Available online at:
`gridlock.york.ac.uk/route/docs/fuel.pdf`, 2001.

[31] R. G. Clegg. A freely available data set for modelling day-to-day route choice. Presented at the 2003 Universities Transport Studies Groups. Available online at:
`gridlock.york.ac.uk/route/docs/utsg2003.doc`, 2003.

[32] R. G. Clegg, A. J. Clune, and M. J. Smith. Traffic signal settings for diverse policy goals. In *Proc. of PTRC Annual Meeting*, volume Seminar K(Modelling), pages 93–103, 2000.

[33] R. G. Clegg, A. J. Clune, and M. J. Smith. When the music's over: the final results of music, an eu project to design and implement traffic signal settings which meet a variety of transport goals. In *11th Mini Euro Conference On Artificial Intelligence In Transport Systems*, 2000.

[34] MUSIC Consortium. MUSIC project final report to the European Union. Available online at:
`gridlock.york.ac.uk/music/docs/Finalv2c.doc`, 1997.

[35] M. Crovella, M. Taqqu, and A. Bestavros. Heavy-tailed probability distributions in the world wide web. In R. E. Feldman R. J. Adler and M. S. Taqqu, editors, *A Practical Guide to Heavy Tails*. Birkhäuser, 1998.

[36] S. Dafermos. Traffic equilibrium and variational inequalities. *Transpn. Sci.*, 14(1):42–54, 1980.

[37] C. F. Daganzo and Y. Sheffi. On stochastic models of traffic assignment. *Transpn. Sci.*, 11(3):253–274, 1977.

[38] G. G. Daugherty, R. J. Balcombe, and A. J. Astrop. A comparative assessment of major bus priority schemes in Great Britain. Transport Research Laboratory Report: TRL409, 1999.

[39] R. B. Davies and D. S. Harte. Tests for Hurst effect. *Biometrika*, 74:95–102, 1987.

[40] J. A. L. Dawson. Comprehensive traffic management in York — the monitoring and the modelling. *Traf. Eng. & Cont.*, 20(11):510–515, 1979.

[41] R. S. Deo and C. M. Hurvich. Estimation of long memory in volatility. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 313–324. Birkhäuser, 2003.

[42] R. B. Dial. A probabilistic multi-path traffic assignment algorithm which obviates path enumeration. *Transpn. Res.*, 5:83–111, 1971.

[43] M. P. Dixon and L. R. Rilett. Real-time OD estimation using automatic vehicle identification and traffic count data. *Journal of Computer-Aided Civil and Infrastructure Engineering*, 17(1):7–21, 2002.

[44] J. R. Duffell and P. J. Carden. Car driver route choice: a perception study of the 'rat running' phenomenon at St. Albans, Hertfordshire. *Traf. Eng. & Cont.*, 24(11):520–527, 1983.

[45] J. R. Duffell and A. Kalombaris. Empirical studies of car driver route choice in Hertfordshire. *Traf. Eng. & Cont.*, 29(7/8):398–408, 1988.

[46] D. E. Duffy, A. A. Mcintosh, M. Rosenstein, and W. Willinger. Statistical analysis of CCSN/SS7 traffic data from working CCS subnetworks. *IEEE Journal on Selected Areas in Communications*, 12:544–551, 1994.

[47] P. Erdös, H. Pollard, and W. Feller. A property of power series with positive coefficients. *Bull. of Amer. Math. Soc.*, 55:201–204, 1949.

[48] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. on Networking*, 4(2):209–223, 1996.

[49] A. Erramilli, R. P. Singh, and P. Pruthi. An application of deterministic chaotic maps to model packet traffic. *Queueing Systems*, 20:171–206, 1995.

[50] A. Erramilli and W. Willinger. Fractal properties in packet traffic measurements. In *Proceedings of the St. Petersburg Regional ITC Seminar*, pages 144–159, 1993.

[51] S. Evans. Derivation and analysis of some models for combining trip distribution and assignment. *Transpn. Res.*, 10:37–57, 1975.

[52] W. Feller. Fluctuation theory of recurrent events. *Trans. of the Amer. Math. Soc.*, 67(1):94–119, 1949.

[53] D. R. Figueiredo, B. Liu, V. Misra, and D. F. Towsley. On the autocorrelation structure of TCP traffic. *Computer Networks*, 40(3):339–361, 2002.

[54] C. Fisk. Some developments in equilibrium traffic assignment. *Transpn. Res. B*, 14(3):243–255, 1980.

[55] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Trans. on Networking*, 9(4):392–403, 2001.

[56] R. Fox and M. S. Taqqu. Large-sample properties of parameter estimates for strongly dependent stationary Gaussian time series. *The Annals of Statistics*, 14:517–532, 1986.

[57] M. W. Garrett and W. Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *Proc. ACM SIGCOMM*, pages 269–280, 1994.

[58] N. H. Gartner. Optimal traffic assignment with elastic demands: A review part II. algorithmic approaches. *Transpn. Sci.*, 14(2):192–208, 1980.

[59] J. Geweke and S. Porter-Hudak. The estimation and application of long memory time series models. *J. Time Ser. Anal.*, 4:221–238, 1983.

[60] L. Giraitis and P. M. Robinson. Parametric estimation under long-range dependence. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 229–249. Birkhäuser, 2003.

[61] J. P. Gliebe, F. S. Koppelman, and A. Ziliaskopoulos. Route choice using a paired combinatorial logit model. Paper presented at the 78th meeting of the Transportation Research Board, Washington DC, January 1999.

[62] P. Goodwin. The end of equilibrium. In T. Garling, T. Laitila T, and K. Western, editors, *Theoretical Foundations of Travel Choice Modelling*, 1998.

[63] C. W. J. Granger and R. Joyeux. An introduction to long-range time series models and fractional differencing. *J. Time Ser. Anal.*, 1:15 – 30, 1980.

[64] H. L. Gray, N. Zhang, and W. A. Woodward. On generalized fractional processes. *J. Time Ser. Anal.*, 10:233–257, 1989.

[65] M. D. Hall, T. Fashole-Luke, D. Van Vliet, and D. P. Watling. Demand responsive assignment in SATURN. In *Proc. of PTRC Annual Meeting*, volume Seminar E, 1992.

[66] P. R. Halmos. *Naive Set Theory*. Springer-Verlag, 1970.

[67] S. Hanson and J. O. Huff. Assessing day-to-day variability in complex travel patterns. *Transpn. Res. Rec.*, 891:18–24, 1982.

[68] S. Hanson and J. O. Huff. Classification issues in the analysis of complex travel behavior. *Transpn.*, 13:271–293, 1986.

[69] S. Hanson and J. O. Huff. Repetition and day-to-day variability in individual travel patterns: Implications for classification. In *Behavioral Modelling in Geography and Planning*, 1988.

[70] S. Hanson and J. O. Huff. Systematic variability in repetitious travel. *Transpn.*, 15:111–135, 1988.

[71] G. H. Hardy and E. M. Wright. *An Introduction To The Theory Of Numbers (Fifth Edition)*. Oxford Science Publications, 1979.

[72] E. Hato, M. Taniguchi, Y. Sugie, M. Kuwahara, and H. Morita. Incorporating an information acquisition process into a route choice model with multiple information sources. *Transpn. Res. C*, 15(1):109–129, 1999.

[73] E. Hauer. Correction of licence plate surveys for spurious matches. *Transpn. Res. A*, 13A:71–78, 1979.

[74] M. O. Haye and M-C Viano. Limit theorems under seasonal long-memory. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 102–110. Birkhäuser, 2003.

[75] C. Hendrickson and G. Kocur. Schedule delay and departure time decisions in a deterministic model. *Transpn. Sci.*, 15(1):62–77, 1981.

[76] C. Hendrickson and E. Planck. The flexibility of departure times for work trips. *Transpn. Res. A*, 18:25–36, 1984.

[77] M. Henry and P. Zaffaroni. The long-range dependence paradigm for macroeconomics and finance. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 419–438. Birkhäuser, 2003.

[78] B. G. Heydecker. On the definition of traffic equilibrium. *Transpn. Res. B*, 20(6):435–440, 1986.

[79] T. Higuchi. Approach to an irregular time series on the basis of the fractal theory. *Physica D*, 31:277–283, 1988.

[80] P. G. Hoel. *Introduction To Mathematical Statistics (Fifth Edition)*. John Wiley & Sons, 1984.

[81] N. Hohn, D. Veitch, and P. Abry. Does fractal scaling at the IP level depend on TCP flow arrival processes? In *Proceedings of the Second Internet Measurement Workshop*, pages 63–68, 2002.

[82] C. Huang, M. Devetsikiotis, I. Lambadaris, and A. R. Kaye. Modeling and simulation of self-similar variable bit rate compressed video: A unified approach. *Computer Comm. Rev.*, 25:114–125, 1995.

[83] J. O. Huff and S. Hanson. Repetition and variability in urban travel. *Geographical Analysis*, 18(2):97–114, 1986.

[84] J. O. Huff and S. Hanson. Measurement of habitual behavior: Examining systematic variability in repetitive travel. In P. Jones, editor, *Developments in Dynamic and Activity-Based Approaches to Travel Analysis*, pages 229–249. Gower Publishing Co., Aldershot, England, 1990.

[85] H. E. Hurst. Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, pages 770–808, 1951.

[86] G. Hyman. The development of operational models for time period choice. Technical report, Department of Environment, Transport and the Regions, Highways Economics and Traffic Appraisal Division, 1998.

[87] O. Jan, A. J. Horowitz, and Z. R. Peng. Using global positioning system data to understand variations in path choice. *Transpn. Res. Rec.*, 1725:37–44, 2000.

[88] M. Jha, S. Madanat, and S. Peeta. Perception updating and day-to-day travel choice dynamics in traffic networks with information provision. *Transpn. Res. C*, pages 189–212, 1998.

[89] M. Jolly and K. M. Briggs. Analysis and simulation of internet round trip times. Available from :
`www.btexact.com/people/briggsk2/Report.pdf`, 2001.

[90] O. D. Jones. Fast, efficient on-line simulation of self-similar processes. Online preprint available from:
`www.maths.soton.ac.uk/staff/ODJones/papers/EBP_SimB.pdf`, 2003.

[91] O. D. Jones and Y. Shen. Analyzing self-similarity in network traffic via the crossing tree. Online preprint available from:
`www.maths.soton.ac.uk/staff/ODJones/papers/EBP_ICCCN03B.pdf`, 2003.

[92] L. Kleinrock. *Queueing Systems: Volume 1.* John Wiley & Sons, 1975.

[93] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic. In D. P. Sidhu, editor, *Proc. ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[94] R. Liu. Analysis of the uncertainties in day-to-day dynamic models from observed choice responses, 2002. Presented at 13th Mini-EURO Conference and 9th Meeting of the EURO Working Group on Transportation, Bari, Italy.
Available online at: `gridlock.york.ac.uk/route/docs/rliubari.doc`.

[95] A. W. Lo. Long-term memory in stock market prices. *Econometrica*, 59:1279–1313, 1991.

[96] J. B. Lock and M. J. Gelling. The tasman bridge disaster. *Australian Road Research*, 6(2):9–16, 1976.

[97] M. Maher. The analysis of partial registration-plate data. *Traf. Eng. & Cont.*, 26(10):495–497, 1985.

[98] M. J. Maher and P. C. Hughes. Recent developments in stochastic assignment modelling. *Traf. Eng. & Cont.*, 39(3):174–179, 1998.

[99] H. S. Mahmassani and G. L. Chang. Dynamic aspects of departure-time choice behavior in a commuting system: theoretical framework and experimental analysis. *Transpn. Res. Rec.*, 1037:88–101, 1985.

[100] H. S. Mahmassani and R. Jayakrishnan. System performance and user response under real-time information in a congested traffic corridor. *Transpn. Res. A*, 25(5):293–307, 1991.

[101] H. S. Mahmassani and Y. Liu. Dynamics of commuting decision behaviour under advanced traveller information systems. *Transpn. Res. B*, 7:91–107, 1999.

[102] B. B. Mandelbrot. A fast fractional gaussian noise generator. *Water Resources Research*, 7(3):543–553, 1971.

[103] B. B. Mandelbrot and J. R. Wallis. Computer experiments with fractional gaussian noises. *Water Resources Research*, 5:228–267, 1969.

[104] D. McFadden. Modelling the choice of residential location. In A. Karlqvist, L. Lundqvist, F. Snickars, and J. Weibull, editors, *Spatial Interaction Theory and Residential Location*, pages 75–96. North-Holland (Amsterdam), 1978.

[105] W. Mendenhall and T. Sincich. *Statistics for Engineering and the Sciences*. Prentice-Hall (Englewood Cliffs, N.J.), 1995.

[106] T. Mikosch and C. Stărică. Long-range dependence effects and ARCH modelling. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 439–459. Birkhäuser, 2003.

[107] R. Morris and D. Lin. Variance of aggregated web traffic. In *Proc. IEEE INFOCOM*, pages 360–366, 2000.

[108] E. Moulines and P. Soulier. Broadband log-periodogram regression of time series with long-range dependence. *J. Time Ser. Anal.*, 27(4):1415–1439, 1998.

[109] R. Mounce. Non monotonicity in dynamic traffic assigment networks. Paper presented at Universities' Transport Studies Group, University of Oxford, 2001.

[110] S. Nakayama, R. Kitamura, and S. Fujii. Drivers route choice heuristics and network behavior: A simulation study using genetic algorithms. In *The proceedings of*

8th International Association of Travel Behavior Research Conference, Gold Coast, Queensland, Australia, July 2000.

[111] A. L. Neidhardt and J. L. Wang. The concept of relevant time scales and its application to queuing analysis of self-similar traffic (or is Hurst naughty or nice?). In *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and Modeling of Computer Systems*, pages 222–232, 1998.

[112] I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.

[113] ns-2 web site. Web site for the ns-2 simulation. URL: `www.isi.edu/nsnam/ns`.

[114] V. Paxson. Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic. *Computer Comm. Rev.*, 27:5–18, 1997.

[115] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ ACM Trans. on Networking*, 3(3):226–244, 1995.

[116] R. M. Pendyala. Measuring day-to-day variability in travel behavior using GPS data. Report for Federal Highway Administration. Available online at: `www.fhwa.dot.gov/ohim/gps/index.html`, 2003.

[117] C. K. Peng, S. V. Buldyrev, H. E. Stanley, and A. L. Goldberger. Mosaic organization of DNA nucleotides. *Phys. Rev. E*, 49:1685 – 1689, 1994.

[118] J. W. Polak. An analysis of travellers' preferred arrival times. Paper presented at the 1st International Symposium on Travel Demand Management, University of Newcastle-upon-Tyne, July 2000.

[119] J. W. Polak and P. Jones. The acquisition of pre-trip information: a stated preference approach. *Transpn.*, 20(2):179–198, 1993.

[120] S. Porter, M. Field, and T. van Vuren. Evidence of peak spreading in the UK. In *Proc. of PTRC Annual Meeting*, volume P393, 1995.

[121] C. Quiroga, R. Henk, and M. Jacobson. Innovative data collection techniques for roadside origin-destination surveys. *Transpn. Res. Rec.*, 1719:140–146, 2000.

[122] R language web site. Web site for the r language. URL: `www.r-project.org/`.

[123] R. H. Riedi. Multifractal processes. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 625–716. Birkhäuser, 2003.

[124] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk. A multifractal wavelet model with application to network traffic. *IEEE Special Issue On Information Theory*, 45(April):992–1018, 1999.

[125] P. M. Robinson. Gaussian semiparametric estimation of long-range dependence. *The Annals of Statistics*, 23:1630–1661, 1995.

[126] B. Ryu and A. Elwalid. The importance of long-range dependence of VBR traffic in ATM traffic engineering: Myths and realities. In *Proc. ACM SIGCOMM*, pages 3–14, 1996.

[127] SACTRA. *Trunk Roads and the generation of Traffic*. HMSO, 1994.

[128] Z. Sahinoglu and S. Tekinay. On multimedia networks: Self similar traffic and network performance. *IEEE Communications*, pages 48–52, 1999.

[129] M. C. Schaefer. License plate matching surveys: Practical issues and statistical considerations. *Inst. of Trans. Engineers Journal*, July:37–42, 1988.

[130] Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall (Englewood Cliffs, N.J.), 1985.

[131] H. A. Simon. Invariants of human behaviour. *Anuual Review of Psychology*, 41:1–19, 1990.

[132] M. M. Slavik. Errors in origin-destination surveys done by number-plate techniques. *Transpn. Res. Rec.*, 1050:46–52, 1985.

[133] K. A. Small. The scheduling of consumer activities: work trips. *American Economic Review*, 72(3):467–479, 1982.

[134] K. A. Small. A discrete choice model for ordered alternatives. *American Economic Review*, 55(2):409–424, 1987.

[135] M. J. Smith. The existence, uniqueness and stability of traffic equilibria. *Transpn. Res. B*, 13:295–304, 1979.

[136] M. J. Smith. Two alternative definitions of traffic equilibrium. *Transpn. Res. B*, 18(1):63–65, 1984.

[137] M. J. Smith. Traffic control and traffic assignment in a signal controlled network with queueing. Paper presetnted aat the Tenth International Symposium on Transportation and Traffic Theory, Boston, Massachussetts, 1987.

[138] K. K. Srinivasan and H. S. Mahmassani. Modeling inertia and compliance mechanisms in route choice behavior under real-time information. Paper presented at the 79th Annual Meeting of the Transportation Research Board, Washington, January 2000.

[139] B. Stephenson and S. Tepley. The verification of CONTRAM in Edmonton. *Traf. Eng. & Cont.*, 25(6/7):376–385, 1984.

[140] M. S. Taqqu. Murad S. Taqqu's Homepage: `math.bu.edu/individual/murad/home.html`.

[141] M. S. Taqqu. Fractional brownian motion and long-range dependence. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 5–38. Birkhäuser, 2003.

[142] M. S. Taqqu and V. Teverovsky. Robustness of Whittle type estimators for time series with long-range dependence. *Stochastic Models*, 13:723–757, 1997.

[143] M. S. Taqqu, V. Teverovsky, and W. Willinger). Is network traffic self-similar or multifractal? *Fractals*, 5:63–73, 1997.

[144] M. S. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modelling. *Computer Comm. Rev.*, 27:5–23, 1997.

[145] M.S. Taqqu, V. Teverovsky, and W. Willinger. Estimators for long-range dependence: an empirical study. *Fractals*, 3(4):785–788, 1995.

[146] K. Train. Recreation demand models with taste variation over people. *Land Economics*, 74(2):230–239, 1998.

[147] E. van Berkum and P. van der Mede. Driver information and the (de)formation of habit in route choice. In R. Emmerink and P. Nijkamp, editors, *Behavioural and Network Impacts of Driver Information Systems*. Ashgate (Aldershot), 1999.

[148] N. J. van der Zijpp and C. D. R. Lindveld. Estimation of O-D demand for dynamic assignment with simultaneous route and departure time choice. In *Proc. of PTRC Annual Meeting*, January 2000.

[149] J. H. van Lint and R. M. Wilson. *A Course in Combinatorics: Second Edition*. Cambridge University Press, 2001.

[150] T. van Vuren. The trouble with SUE – stochastic assignment problems in practice. In *Proc. of PTRC Annual Meeting*, September 1994.

[151] T. van Vuren, A. J. Daly, and G. Hyman. Modelling departure time choice. Paper presented at Colloquium Vervoersplanologisch Speurwerk, Amsterdam, November 1998.

[152] W. Vickrey. Congestion theory and transport investment. *The American Economic Review*, 59(2 (Papers and Proceedings of the Eighty-first Annual Meeting of the American Economic Association)):251–260, May 1969.

[153] P. Vovsha. Cross-nested logit model: an application to mode choice in the tel-aviv metropolitan area. Paper presented at the 76th Annual Meeting of the Transportation Research Board, Washington DC, January 1997.

[154] X. J. Wang. Statistical physics of temporal intermittency. *Phys. Rev. A*, 40(11):6647–6661, 1989.

[155] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proc. of the Inst. of Civil Engineers II*, 1:325–378, 1952.

[156] D. P. Watling. Maximum likelihood estimation of an origin-destination matrix from a partial registration plate survey. *Transpn. Res. B*, 28B(4):289–314, 1994.

[157] D. P. Watling. Asymmetric problems and stochastic process models of traffic assignment. *Transpn. Res. B*, 30(5):339–257, 1996.

[158] D. P. Watling and M. J. Maher. A graphical procedure for analysing partial registration-plate data. *Transpn. Res. B*, October, 1988.

[159] D. P. Watling and M. J. Maher. A statistical procedure for estimating a mean origin-destination matrix from a partial registration plate survey. *Transpn. Res. B*, 26B(3):171–193, 1992.

[160] N. Weiner. Generalized harmonic analysis. *Acta. Math.*, 55:178–258, 1930.

[161] C. Wen and F. S. Koppleman. The generalised nested logit model. Paper presented at the 79th Annual Meeting of the Transportation Research Board, Washington DC, January 2000.

[162] P. G. Williams, H. R. Kirby, F. O. Montgomery, and R. D. Boyle. Evaluation of video-recognition equipment for number-plate matching. In *Proc. of PTRC Annual Meeting*, volume P306, pages 229–239, 1988.

[163] W. Willinger. Traffic modeling for high-speed networks: Theory versus practice. In *Stochastic Networks*, pages 395–409. Springer-Verlag, 1995.

[164] W. Willinger, V. Paxson, R. H. Riedi, and M. S. Taqqu. Long-range dependence and data network traffic. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory And Applications Of Long-Range Dependence*, pages 373–407. Birkhäuser, 2003.

[165] W. Willinger, M. Taqqu, and A. Erramilli. A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks. In *Stochastic Networks*, pages 339–366. Oxford University Press, 1996.

[166] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ ACM Trans. on Networking*, 5(1):71–86, 1997.

[167] A. Zygmund. *Trigonometric Series (Third Edition)*. Cambridge University Press, 2003.