The Statistics of Cycling

Matthew Arnold

April 27, 2010

Contents

List of Figures

Abstract

We present an analysis of data from a small number of bicycle counters, located on cycle routes in the United Kingdom. We first apply methods of cluster analysis to the cleaned data. We then attempt to link the results of this clustering to explanatory variables, with our main focus on multinomial logit models, and we note some of the problems one faces with small datasets.

Declaration

This piece of work is a result of my own work except where it forms an assessment based on group project work. In the case of a group project, the work has been prepared in collaboration with other members of the group. Material from the work of others not involved in the project has been acknowledged and quotations and paraphrases suitably indicated.

Chapter 1

Introduction

In this project we intend to analyse cycle count data collected by Sustrans. We first introduce Sustrans, the cycle network and report on previous research in this area. Here we also summarise our communication and interaction with our client, Sustrans. Chapter 2 describes cleaning the supplied data as well as collecting information about certain parts of the data. Using the theory learnt in Chapter 3 we perform different cluster analyses upon the data in order to try and learn about any usage patterns. We display these results in Chapter 4. In Chapter 5 we use explanatory variables that describe cycle counter locations in order to gain an understanding of how changes in these variables lead to categorisation of counters. Chapter 6 is concerned with handling sparse contingency tables caused by small datasets.

1.1 Sustrans

Sustrans is a registered charity founded in 1977 that promotes sustainable transport within the United Kingdom. Significant financial support from the Millennium Commission allowed the construction of the first 5,000 miles of the National Cycle Network. Sustrans aims to maximise its effectiveness by creating new cycle routes and projects themselves as well as encouraging other bodies to jointly or independently fund similar schemes.

1.1.1 The National Cycle Network

The first route opened by Sustrans was the Bristol and Bath Railway Path, a traffic-free trail lasting for 17 miles along a disused railway. Sustrans worked with many local authorities, organisations and funders to complete over 12,000 miles of routes across the United Kingdom by December 2007. The network passes within one mile of half the population and the network continues to grow in popularity.

1.1.2 National Routes

The National Route numbering system works in a similar way to the original A road numbering system in Great Britain. The National Routes connect cycle routes from London to the regions (and separately in Scotland) and are numbered from 1 to 9. Towns that are not directly connected to a National Route are linked to the Network by Regional Routes. The UK has been divided into ten regions, and Regional Routes are labelled from 10 to 99 within each region. [1]

1.1.3 Research and Monitoring Unit

The Sustrans Research and Monitoring Unit, based in Newcastle, records and analyses data collected about the number of cyclists using the National Cycle Network every day. Route User Surveys are used to find out how cyclists and pedestrians use the Network and analyse the impact of Sustrans projects and interventions. Every year the unit produces an annual usage report using data from the counters and the Route User Surveys. [2]

1.1.4 Automatic Cycle Counters

The main source of data for the R&MU are the automatic cycle counters installed on routes throughout the UK. Sustrans uses data from the counters to evaluate changes in route usage by cyclists. The counters are able to distinguish between pedestrians and cyclists by evaluating the amount of metal that passes over them. The counters provide data on the number of cyclists that pass over it in 1, 15 or 60 minute intervals, and in up to 8 directions. A more detailed description of the counters is given in Section 1.3.3 and photographs of a counter are shown in Figure 1.1.

1.2 Previous Research

Research into the cycle counter data is undertaken by Sustrans, the Department for Transport and other interested bodies in the UK. A number of authors have published analyses of data from outside of the UK. In particular, Richardson [3] presents a summary of results from analysing the data available from the cycle network in Switzerland, operated by Veloland Schweiz. The count data are collected from manual counts at 16 locations across the network – in addition, cycle riders were sampled from and asked to complete a questionnaire about their cycle journey. The survey was carried out on dates from 1999 to 2002. Looking at the count data in particular, we see that in absolute terms the number of riders increased over these years. We do not see a breakdown of the counts into hourly counts, however. The rest of the report is mainly concerned with the results of the cycling questionnaires, which Sustrans do carry out on cyclists in the UK, but which are not the focus of this project. Some results for the UK surveys are summarised in Cope, et al. [4]. Richardson describes in further detail some results from the Veloland Schweiz study in [5]. Here, by combining the count data and the travel questionnaires with GPS and route data, he proposes a model for the number of journeys observed at a distance from a community. He then uses this to estimate usage across the entire Swiss network and thus the economic impact of the network in Switzerland.

Lumsdon et al. [6] give an analysis of automatic cycle count data on the North Sea cycle route in relation to the economic benefit of the route to local communities. The continuous count data was combined with intercept surveys and travel diaries of cyclists using the route in order to provide an appraisal of user spending along the route. With regards to the counter data, they found that counts were highest in urban centres. They found significant differences between mean daily flows during June/July and August, owing primarily to the holiday season, but no differences between counts in a single week. In particular, they note that at Woodhorn they saw a reduction in usage during the holiday period and attribute this to the counter primarily seeing commuter usage, whilst at the other counters they saw increased usage during the same period. They identified leisure usage at Low Hauxley and Leatham Shanks by seeing significant differences between average usage on weekdays compared to weekends.

Vélo Québec presents its analysis of cycling habits on routes in the province in its report for 2005 [7]. In particular, shapes representing average usage on urban cycleways are shown on page 9. We expect that analysing proportions of cyclists per hour will yield similar shapes with our data. The rest of the report summarises the success of the bicycle network there since the adoption of the 1995 cycling policy and construction beginning on the Route verte, as well as benefits to safety on the network and overall health of cyclists.

Every year the Department of Transport publishes the results of its National Travel Survey [8]. This contains data relating to cycle usage, particularly the number of trips and distances involved. The data used are collected using face-to-face interviews and travel diaries, similar to some of the methods analysed by Richardson and Cope, et al. The survey is concerned with all methods of travel within the UK and so the cycling statistics only form part of the results presented. Of particular interest in the 2008 report are Charts 8.2 and 8.4. Chart 8.2 divides daily trips across all modes of transport into 4 main groups – commuter, education, shopping and social/entertainment; and shows which group has more journeys on each day of the week. As we might expect, more trips are made for commuter

and education purposes compared with shopping and social on 'working week' days and the situation is reversed at the weekend. This is a pattern we might expect to see replicated in our cycling analysis. It is interesting to note that the Survey ignores leisure trips that 'are themselves a form of recreation'. It is unclear as to what extent this excludes for instance, circular cycle rides in the countryside. The counter data provided for our analysis will be counting these journeys. Chart 8.4 shows the average daily trips for each main mode of transport in each month of the year. Again we see an intuitive result for the cycling data; that is, lower usage in the winter months and higher usage in the summer. This is also something that might expect to be able to observe in our data.

1.3 Interactions with Sustrans

An important part of this project has been my interactions and collaboration with the client, Sustrans. The aim of the project has been to find something relevant or of use to Sustrans from their data. My liaison has been Dr. Lisa Muller from the Sustrans Research and Monitoring Unit in Newcastle.

1.3.1 Initial meeting

The initial meeting for this project took place in October 2009. Dr. Muller presented some examples of the sort of counter data files that Sustrans generates. Here, Dr. Muller noted some of the problems that Sustrans faces in data collection. Sometimes data is missing or corrupted because of a broken counter, or a counter has used up its power source etc. Sustrans often relies on local authorities to manage counters, collect data from them and then relay that data back. For reasons that are not always entirely clear some local authorities choose to obfuscate or remove portions of the data. Handling missing data is an important problem for Sustrans and one that I would have to consider. In addition, Dr. Muller commented on the sorts of problems that the RM&U handle, mainly to do with route usage. The Department of Transport has various formulae relating to the economic benefit of cycle routes in relation to usage so that Sustrans can evaluate the benefit of particular routes to a community. In addition, Sustrans performs interventions to encourage cycling in a particular area. These would benefit from knowing the characteristic usage along routes in the targeted area in order to maximise the effectiveness of interventions, and therefore increase usage. Usage increases would also have an economic benefit.

1.3.2 Data acquisition

The data that was seen at the initial meeting with Dr. Muller was emailed to me following that meeting. This contained the data for 5 counters so that I could begin some preliminary exploration of the data. More data for different counters, up to the total of 107, were emailed to me periodically during the first term. I concatenated these into a single file to work with. At the start of the second term, Dr. Muller emailed the Sustrans counter location classification data, which I used as the basis for the work on explanatory variables. Dr. Muller then provided some manual counts so that I could demonstrate prediction of counter classification from this data.

1.3.3 Trip to Prudhoe

At the invitation of the Sustrans unit in Newcastle, I visited an automatic cycle counter in operation on the Coast2Coast cycle route through Prudhoe in December 2009. The counter was set up to record cycles travelling in two directions along the path. The counter is connected to a loop of inductive metal that is laid in a roughly trapezoid shape underneath the cycle path surface. The photographs in Figure 1.1 were taken of the automatic cycle counter installed on the Coast2Coast cycle route on the south bank of the Tyne at Prudhoe. I have highlighted the shape of the loop of wire underneath the path. The loop is an inductive loop of coiled wire which generates a small, localised magnetic field. When a metal object passes through the magnetic field a waveform is detected by the counter and this is registered as a count. The size and shape of the loop enables it to detect more than one cycle travelling over it at once, as well as the direction of travel. Frequently, and as seen in the photograph,

Figure 1.1: Counter unit

this is achieved using a trapezoidal loop shape. We performed some experiments to see what types of bicycle the counter could detect. We found that the 'normal' bicycles were straightforwardly detected, including a fold up cycle brought along. However, a small child's bicycle was not detected at all in any direction. This of course has particular implications for the counts at school and leisure counters. We also observed that a carbon fibre bicycle (with only a small amount of ferrous material in the gear and brake assemblies) was counted incorrectly. The counter either did not detect the bicycle at all, or counted an erratic number of times always in the same direction, no matter which direction the cycle was ridden across the loop. Since the cost of carbon fibre cycles is very high, I suspect that the impact of these errors is small; however, it is still worth considering. I also spoke with one of the researchers at the unit who said that they had tested a particular automatic counter by performing a manual count of bicycles at the same location over the course of a day. They found that the record from the automatic counter was much higher than that for the manual count. This was somewhat concerning to me and to Sustrans. Dr. Muller explained later that this particular case is unusual and that as far as Sustrans was aware the counters were usually accurate. My work on daily and yearly counter usage profiles focuses upon proportion of counts per hour or month. Therefore the actual size of the counts is less crucial. The results proposed should therefore remain safe as long as we can assume that the counter produced errors in its counts in an independent and non-systematic way.

Returning to the RM&U unit in Newcastle, I took the opportunity to ask Dr. Muller some questions about my findings from the first half of the project, particularly the work on clustering. The four categories that I had found tallied with what Sustrans had previously found. It appears however that Sustrans had not previously found such a strong response from the schools group.

1.3.4 Further meeting

Returning to Durham after the winter break, we had a further meeting with Dr. Muller. Here we discussed what direction the project might take that would interest Sustrans. We agreed that trying to find out how the explanatory variables might predict the categorisation of a new counter would be useful. Certainly, if we are able to suggest the sort of usage we might see on a new route without having to send and pay people to conduct manual counts and surveys this would reduce the expenditure of new route planning. Equally, the advertising of the new route could be targeted directly to groups of people expected to use the route. In addition, where manual counts have been conducted in places where new routes are planned it would be useful to try and predict absolute numbers and types of users once the route was in place. Dr. Muller commented that once the route was completed and advertised Sustrans usually expected to see an approximately 30% increase in usage.

1.3.5 Presentation

In March 2010 I reported back to Sustrans with the near-final conclusions of this project. I gave an approximately 20 minute presentation, where I gave an overview of the work that I have completed and present in this project. I summarised the results of the cluster analysis and gave an example of the use of multinomial logit models in classifying counters based on usage. The presentation provoked some interesting discussion about the possible uses of what I propose in this project. The work on cluster analysis confirmed what had previously been believed about usage patterns at cycle counters and this was helpful to Sustrans as my results are data-driven. I did not present the work on using a discriminant analysis to predict counter classification from an existing cluster analysis, however, I expect this to be of use to Sustrans, as it will enable them to take manual counts from sites and predict the type of usage that might be seen if a counter was installed at that location. The work on multinomial logit models that I presented used the clustering as the basis for grouping the counters and then relating this grouping to the explanatory variables. I pointed out that these models do not necessarily depend on using clustering to get an initial grouping, but that Sustrans would be able to use them with any grouping of counters and variables that they think might explain the grouping. We also discussed the potential implications for predicting actual usage numbers on routes, and it would be interesting to see if the long term usage on routes and at counters could be explained and then predicted using the logit models shown.

Chapter 2

Data

We begin by looking at the type and nature of the data files supplied by Sustrans. We then consider some of the modifications and additions to the data format to make it usable for our purposes.

2.1 The data files

The data files from Sustrans came formatted using the Comma Separated Values scheme (CSV) with a year's worth of data from a counter in each file. Data was available for some counters as far back as 1999, but most have data starting in 2004. The data for most counters ends in 2009. The data files record the number of cyclists that passed over the counter in 2 directions for each hour in every day for the number of years specified above. Where the counter failed to record any data for any reason (e.g. battery failure), a -2 is recorded for both directions. Where data has been removed or obscured, a -1 is recorded. Sustrans often receives data from counters that are managed by local authorities. For reasons beyond the control of Sustrans, some data is obscured or missing. An excerpt from the data in CSV format is provided below.

```
01510,14/11/06,0800,-0002,-0002
01510,14/11/06,0900,-0002,-0002
01510,14/11/06,1000,-0002,-0002
01510,14/11/06,1100,0000,0001
01510,14/11/06,1200,0004,0001
01510,14/11/06,1300,0002,0001
01510,14/11/06,1400,0004,0004
```

The first column gives the counter number, the second the date, the third the hour. The fourth and fifth record the number of cyclists that passed over the counter in the given hour. The data files for 107 counters were concatenated into a single CSV file and this was loaded into R using `read.csv()`, and we refer to the created data frame as `biking` throughout. We also label the columns of `biking` with some user-friendly names, using

```
names(biking) <- c("counter","date","hour","upcount","downcount")
```

2.2 Data cleaning (and difficulties)

Unfortunately the data supplied has errors or invalid data in some places. After loading the data into R we perform a few basic sanity checks upon the data. We run `na.omit()` to remove incomplete rows from `biking`. We also check for rows that contain invalid data by checking that the date cell is formatted either DD/MM/YY or DD/MM/YYYY. Some of the data files contained invalid data when they were concatenated into 1 file, however it is far too difficult to identify these files singularly. Therefore we remove this erroneous data in R. This procedure is as follows

```
biking1 <- subset(biking, (nchar(as.character(biking$date))==8))
```

```
biking2 <- subset(biking, (nchar(as.character(biking$date))==10))
biking <- rbind(biking1,biking2)
```

2.2.1 Removing -1,-2

Entries in the data frame where the count for that hour has been recorded as `-1` or `-2` are easily removed. Again, these occur when the counter is unable to count or data has been removed or obscured.

```
biking <- subset(biking, !(upcount < 0))
biking <- subset(biking, !(downcount < 0))
```

2.2.2 Day of the week

It is reasonable to expect that we might see different usage patterns on the cycle network that depend on the day of the week. Commuter routes and particularly school routes should see different usage at the weekend compared to during the week. We therefore attach a further column to our `biking` data matrix that is the day of the week. We do this by defining a function that takes the date as formatted in our raw data files and outputs the day of the week, using the `chron` library [9].

```
library(chron)
##define these here for more speed
daynames <- c("Sunday", "Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")
daynums <- c(0:6)

getdayofweek <- function(mondayyear)  {
## extract day, month, year from the format in our data frame
date.day <- as.numeric(substr(mondayyear, 1,2))
date.month <- as.numeric(substr(mondayyear,4,5))
testyear <- as.numeric(substr(mondayyear,7,8))
## we need to tell the difference between the year expressed in 2 digits and 4 digits
if (testyear == 20) { testyear = as.numeric(substr(mondayyear,9,10)) }
if (testyear > 80) {
  date.year = testyear + 1900 } else
{
  date.year = testyear + 2000
}
## output the day of the week
return(daynames[day.of.week(date.month,date.day,date.year)+1])
}
```

We then use this function to attach another column to our data frame.

```
days <- character()
possibledays <- biking[,2]
days <- sapply(possibledays,getdayofweek)
biking <- cbind(biking, days)
```

2.2.3 Urban or rural?

One of the most general classifications for cycle routes given by Sustrans is whether a counter location is 'rural' or 'urban'. This is a subset of a larger classification system, but it is thought that this particular classification might be of most interest early on. We therefore create two further data frames which contain copies of the data from locations that are either rural (in `biking.rural`) or urban (`biking.urban`). This is done in the code below by removing locations that don't match the criteria from a copy of the original `biking` matrix.

```
ruralurban <- read.csv("ruralurban.csv", header=TRUE)
urbanID <- subset(ruralurban, (urban == 1))$counter
ruralID <- subset(ruralurban, (rural == 1))$counter

##chop the biking data into rural, urban
biking.rural <- biking
for (i in urbanID) {
biking.rural <- subset(biking.rural, !(counter==i))
}

biking.urban <- biking
for (i in ruralID) {
biking.urban <- subset(biking.urban, !(counter==i))
}
```

2.2.4 Counter location information

Sustrans provides information about the location of each counter. These data are primarily binary responses to classification criterion, such as whether the location is rural or urban (as in the previous paragraph); whether the route is surfaced; whether it is lit; and so on. In all, there are 30 binary responses to these criteria. A further explanation of this dataset and its application can be found in Section 5.1.

2.3 A single counter

We begin by examining the data for a single counter, numbered 1510. The internal Sustrans meta data lists the counter as recording data for the Normandy Way cycle route in Hinckley, Leicestershire. The route is on the National Cycle Network and is classified by Sustrans as urban. It is a cycleway laid upon a surfaced footpath and is traffic-free. The counter records in two directions the number of cycles that pass over it each hour - channel 1 is eastbound and channel 2 westbound. For each hour unit we shall consider the sum total of both channels, i.e. the number of bicycles to have passed over the counter that hour. We select the data for 45 months (Jan 2006 - September 2009) from the main data frame using

```
subset(biking, (counter == 1510))
```

2.3.1 An average day

An obvious place to start an examination of cycling counter data is the construction of the profile of an 'average' day graph. For each hour we calculate the mean response over the 45 months for which we have data. We then rescale the resulting vector by dividing by the total of the responses, and we obtain a 'percentage' of the daily total for each hour. The plot of this rescaled data is in Figure 2.1. Where the response for an hour was -2, this indicates that no usable data was returned and this response is disregarded. Alternatively we can look at the average day during the standard working week (Monday-Friday) or at the weekend. Already we are able to see that usage is different at the weekend compared to that during the week. During the week we see peaks (indicating high usage) at 9am and around 5-6pm. These are intuitively times when people are commuting to and from work. At the weekend we see that usage increases relatively consistently from 8am to a peak at 12pm and 1pm. The afternoon shows relatively high usage followed by a swift reduction in usage after 4pm. This is the usage we might intuitively expect to see on a route that is being used for leisure. In Figure 2.2 a further example of a similar graph for counter 18 in Bristol. The peaks of commuter type usage are very clear on the weekday graph, and by comparison the weekend usage is a much flatter, single peak curve depicting leisure usage.

Figure 2.1: Comparison of daily profiles at counter 1510

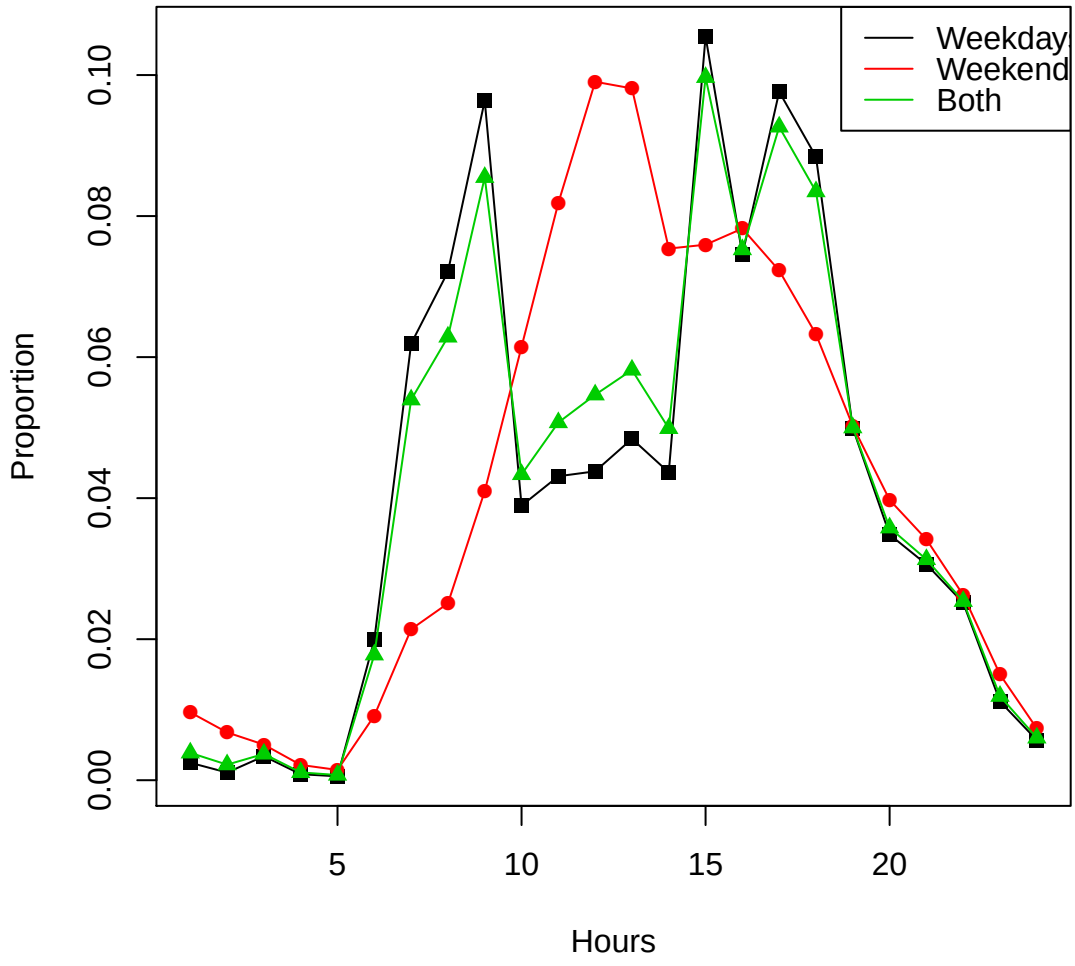
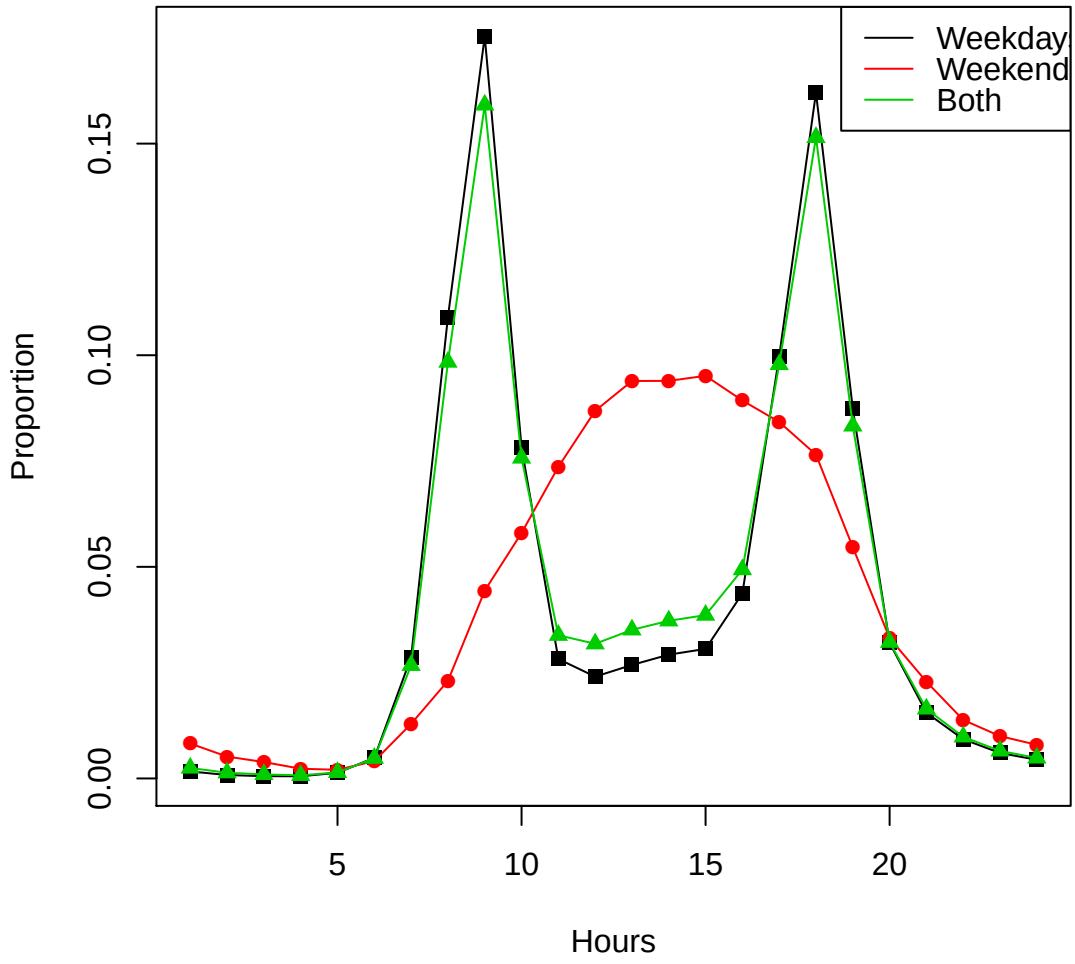


Figure 2.2: Comparison of daily profiles at counter 18



2.3.2 An average year

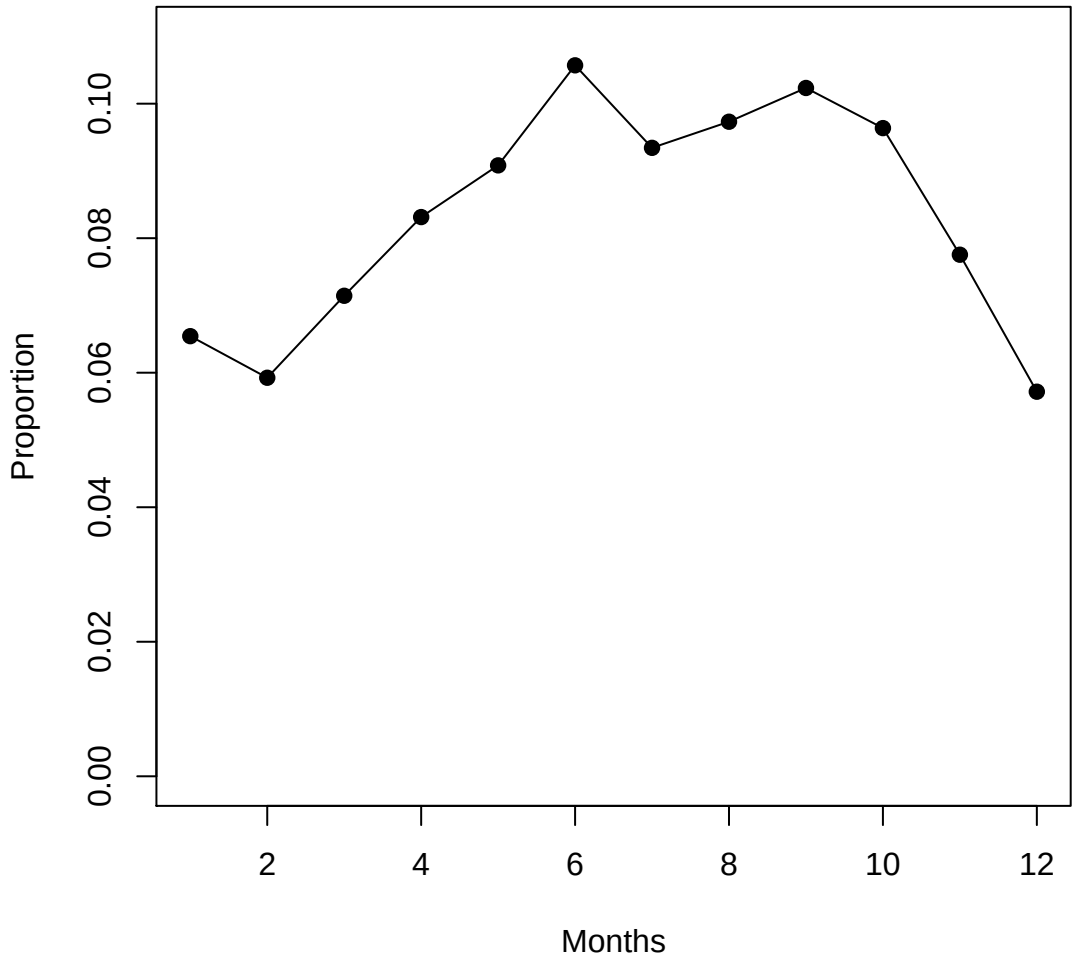
Another simple thing to look at is the 'average' year graph for counter 1510, where we consider the proportion of the year total count that occurs per month. Looking at the graph in Figure 2.3, we see results that we might intuitively expect, that is, much higher usage during the warmer spring and summer months compared to autumn and winter. The slight peak in January is likely to be random variation. We note that creating this graph has required a number of approximations. We first take the subset of the full data frame that relates to this counter. We then loop through each of the 12 months, and take a subset of the data for the location. From this subset, we sum the total number of cycles counted in this month across all available years. We then divide by the number of days sampled to find the average total for a day in that particular month. We then multiply by the usual number of days in this month. This allows us to somewhat guess monthly totals from counters where we see incomplete months in the data frame. We then create a vector of length 12 containing these totals for each month and rescale this to give us proportions. Obviously this method breaks down if we have a counter where we see no counts at all in a particular month due to missing data. The year profiles therefore allow us to get an impression of usage patterns across a year.

2.4 Sample of completed data frame

A short excerpt of the cleaned data frame is provided below. We note that although we have counts for both directions, in this project we will consider only the total traffic across the counter in an hour period, i.e. the sum of counts in both directions.

	counter	date	hour	upcount	downcount	days
5049	1047	30/07/03	800	0	0	Wednesday
5050	1047	30/07/03	900	0	0	Wednesday
5051	1047	30/07/03	1000	1	0	Wednesday
5052	1047	30/07/03	1100	3	0	Wednesday
5053	1047	30/07/03	1200	0	0	Wednesday
5054	1047	30/07/03	1300	0	0	Wednesday
5055	1047	30/07/03	1400	1	1	Wednesday
5056	1047	30/07/03	1500	3	1	Wednesday
5057	1047	30/07/03	1600	5	0	Wednesday
5058	1047	30/07/03	1700	3	5	Wednesday
5059	1047	30/07/03	1800	6	1	Wednesday

Figure 2.3: Year profile at counter 1510



Chapter 3

Clustering

We now consider a type of unsupervised learning about our data called cluster analysis. If we look at average usage profiles (as in Figures 2.1 and 2.3) we might try to find out what common shapes occur in the usage profiles from all counters. Cluster analysis allows us to look for these shapes. We expect counters that have usage profiles with similar shape to occur in the same clusters. We therefore present the theory associated with cluster analysis in this chapter. We follow this with a careful application of these techniques to our cycling data and present those results in Chapter 4.

3.1 Dissimilarity and distance measures

Before we can begin our cluster analysis we must find a way of identifying differences between the average day graphs at each counter. For this we use a dissimilarity measure. Dissimilarity measures that satisfy the metric property

$$d_{ij} + d_{ik} \leq d_{jk}$$

are called *distance* measures. Some possible dissimilarity measures are given in table 3.1, taken from [10, p.46].

$(\sum_{k=1}^{p}(x_{ik} - x_{jk})^2)^{\frac{1}{2}}$	Euclidean distance
$\sum_{k=1}^{p} \lvert x_{ik} - x_{jk} \rvert$	Manhattan distance
$\sum_{k=1}^{p} \frac{\lvert x_{ik} - x_{jk} \rvert}{x_{ik} + x_{jk}}$	Canberra metric
$\frac{\sum_{k=1}^{p} x_{ik} x_{jk}}{\left[\sum_{k=1}^{p} x_{ik}^2 \sum_{k=1}^{p} x_{ik}^2\right]^{\frac{1}{2}}}$	Angular separation

Table 3.1: Dissimilarity measures

We construct our matrix `biking.avgdays` so that each row is the average daily usage profile for a single counter. Reading across a row gives the proportion of counts observed in each hour, so there are 24 columns. Each row is therefore the equivalent numerical representation of the graph in Figure 2.1 for each separate counter. One can of course generalise this method to construct data matrices in order to analyse usage profiles for different time periods. We calculate the dissimilarity d_{ij} between pairs of rows using the Euclidean metric. This provides us with an idea of how similar the average day profiles are at counters i and j. This is easily done in R using the function `dist()` which produces a lower triangular distance matrix, the entries of which, (x_{ij}), correspond to the distance between counter i and counter j. The output `x` from `dist()` is a special R type called `dist` which is directly accepted by the R functions which we use for cluster analysis. If necessary, it is possible to output a distance matrix type `x` as a matrix by converting it using the function `as.matrix(x)`. Below is a sample distance matrix produced by `dist()` for 5 counters.

```
     1047 1063 1069 1073
1063 0.19
1069 0.35 0.27
1073 0.20 0.09 0.25
1078 0.19 0.14 0.20 0.11
```

From this, we see that counters 1073 and 1063 are most similar by the Euclidean metric; and counters 1069 and 1047 are the most dissimilar.

3.2 K-means clustering

The k-means algorithm for cluster analysis is partitional, i.e. the data set is split into k distinct subsets. Given a set of n d-dimensional observations x_i, the algorithm aims to partition the observations into k sets S_i with $k < n$ so that the within cluster sums of squares is minimized. Let $\boldsymbol{S} = S_i$ be the set of such partitions. Let $\|\cdot\|$ be the Euclidean distance metric, defined above. Then we aim to find the partition $\boldsymbol{S}$ that satisfies

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_j\|^2 \tag{3.1}$$

with μ_i the d-dimensional mean vector of S_i.

3.2.1 Algorithm

For general d and k, the k-means clustering problem is NP-hard. This remains even if we take $k = d = 2$. Thus a heuristic method must be used.
To initialise the algorithm, a set of k means $m_1 \dots m_k$ must be specified. In the R implementation `kmeans()` these are allocated randomly. They can however be heuristically determined, using for example the k-means++ algorithm proposed in [11]. The usual k-means algorithm is then executed by alternating between the following two steps:

Assignment: Assign each observation to the cluster with the closest mean according to the Euclidean distance metric.

$$S_i^{(t)} = \left\{ x_j : \|x_j - m_i^{(t)}\| \leq \|x_j - m_q^{(t)}\| \ \forall \ q = 1, \dots, k \right\} \tag{3.2}$$

Update: Update the means m_i to be the centroid of the observations in the cluster

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{3.3}$$

The algorithm ends when the assignments do not change upon further iterations. However, there is no guarantee that it will converge to the global optimum, since the results depend on the initial mean vectors.

3.2.2 Determining k

k is an important parameter, and the wrong choice can give poor results. However, determining the *correct* choice of k is tricky, particularly where the number of observations is quite constrained, as in the Sustrans data. There is no conclusive way to determine k. However, there are several methods that look at both the within-cluster and between-cluster sum of squared distances for varying k.

3.2.3 Formal methods of determining k

So called 'stopping rules' have been proposed as a way of identifying the value k at which the partitioning or amalgamation should stop. These stopping rules are either global or local. Global rules take into account within-cluster and between-cluster variability in order to identify k where a measure $G(k)$ is optimal. Local stopping rules are based only on a part of the data (except for the change from $k = 1$ and $k = 2$, obviously). Local rules only apply to nested or hierarchical clusterings (described 3.5), that is where an increase in k comes only from splitting apart existing clusters. Local rules require specifying a significance or threshold level, which may be difficult as it may depend on unknown properties of the given data. Some examples of global and local stopping rules are given in Gordon [12] and are reproduced below. These were the best performing methods identified by Milligan and Cooper [13]. Gordon identifies some methodological issues with the study, however he suggests that it may still be of use for identifying stopping rules that perform poorly.

Global rules

1. Calinski and Harabasz [14] index defined by

$$G(k) := \frac{B/(k-1)}{W/(n-k)}$$

   where B denotes the total between cluster sum of squared distances and W the total within cluster sum of squared distances. We look for the value of k which maximises this index.

2. Variants of Goodman and Kruskal's [15] γ. We compare all the within cluster dissimilarities and the between cluster dissimilarities. A comparison is called concordant if a within cluster dissimilarity is strictly less than a between cluster dissimilarity; and discordant if the comparison is strictly greater. If the comparison is an equality then it is disregarded. Denoting by S_+ and S_- respectively the number of concordant and discordant comparisons, the index is defined as

$$G(k) := \frac{S_+ - S_-}{S_+ + S_-}.$$

   We look for the value of k which maximises this index.

3. Let $D(k)$ be the sum of all within cluster dissimilarities in a k cluster partition. If the partition has a total of r such dissimilarities, D_{min} is defined to be the sum of the r smallest and D_{max} the largest. We then standardise $D(k)$ with respect to D_{min} and D_{max} by defining

$$D(k) := \frac{D(k) - D_{min}}{D_{max} - D_{min}}$$

   We look for the value of k which minimises this index.

With these indices we restrict ourselves to looking at small values of k.

Local rules

1. Duda and Hart [16]. Define W_1 as the within cluster sum of squared distances and W_2 as the within cluster sum of squared distances if the cluster is optimally divided into two. Comparing W_1 and W_2 gives a rule for identifying if a cluster should not be subdivided (homogeneous). If the cluster contains m objects explained by p variables then the hypothesis that the cluster is homogeneous is rejected if

$$\frac{W_2}{W_1} < 1 - \frac{2}{p\pi} - z\left[\frac{2\left(1 - \frac{8}{p\pi^2}\right)}{mp}\right]^{\frac{1}{2}}$$

   where z is the standard normal giving the significance level of the test.

2. Given W_1, W_2, m and p as above, Beale [17] similarly proposes a test for deciding if a cluster should be subdivided. Defining

$$F := \frac{\frac{W_1 - W_2}{W_2}}{\left(\frac{m-1}{m-2}\right) 2^{\frac{2}{p}} - 1}$$

we compare F with an $F_{p,(m-2)p}$ distribution and reject the hypothesis of a single cluster for significantly large values of F.

3.3 Partitioning around Medoids

The partitioning around medoids (PAM) algorithm uses actual data points as the centres of clusters (medoids), instead of choosing a point in space to which distance is minimised as in the k-means algorithm. This makes the PAM method more robust to noise and outliers compared to k-means, because it minimises sums of dissimilarities instead of the k-means approach of minimising sums of squared distances [18].

3.3.1 Algorithm

The PAM method for k clusters can be fairly simply described as follows

1. Initialisation: Randomly select k of the n data points to be the starting medoids.
2. Allocate each data point to the cluster centred on the closest medoid. Here closest is defined by using our dissimilarity matrix created earlier from our choice of distance metric.
3. For each medoid m and then for each non-medoid p, swap m and p and compute the total distance of this configuration.
4. Select the configuration with smallest distance.
5. Repeat 2 to 5 until there is no change in the set of m medoids.

Compare the initialisation step with that for k-means. Here the chance of picking starting centroids that are not representative of the data as a whole is reduced by restricting the choice of starting points to the n data points instead of, in theory, any point in the space that the data inhabits.

3.3.2 In R

In R, the function `pam()` is found in the package `cluster` [19]. `pam()` may either take a data matrix directly by specifying the distance metric to be used, or otherwise applies the PAM method to a dissimilarity matrix. The function also allows the user to specify directly the initial choice of medoids, if warranted. `pam()` outputs an object of class `"pam"` that represents the clustering.

3.4 Silhouette plots

Silhouette plots are a way to graphically display the results of a partitioning, with each cluster represented by a 'silhouette' which is based on a comparison of its tightness and separation. This method is first described by Rousseeuw in [20]. Firstly partition the data into k clusters via a clustering technique. Let a data point i have been assigned to a cluster A. Then denote by $a(i)$ the average dissimilarity between i and all other data points in A. Following this, compute, for any cluster C, the average dissimilarity between i and all other data points in C. We do this for all possible clusters C and take the minimum, and call this $b(i)$. Let B denote this mimimal cluster, and we call B the *neighbour* of i. B can be thought of as the 'next best alternative' to assign i into, if we could not assign i to A. We then define a number $s(i)$ as follows

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases} \tag{3.4}$$

Should A only contain a single data point, then we define $s(i) = 0$. It is also clear from the definition of $s(i)$ that

$$-1 \leq s(i) \leq 1.$$

We also require that our distance metric has a ratio scale, for example, that a dissimilarity of 8 is thought of as twice as large as a dissimilarity of 4. Usefully, our Euclidean metric satisfies this property.

3.4.1 Meaning of s

It is simplest to consider $s(i)$ to be a measure of how well the data point i has been clustered. By considering $s(i)$ for all possible i we get a feeling of how well defined our clusters are, in terms of how compact the individual clusters are as well as the distance between clusters. If $a(i) << b(i)$ then $s(i)$ is close to 1. The relative sizes of $a(i)$ and $b(i)$ suggest that the point i is appropriately clustered inside A and that the next best choice B is not nearly as good as A. If $a(i) \approx b(i)$ then $s(i) \approx 0$ which suggests that it is not particularly clear whether i should be assigned to A or B. If $s(i) = 0$ then i lies equally far away from A and B. If $a(i) >> b(i)$ then i is much closer to B than A, so it would have been most appropriate for i to have been assigned to B. Hence a value of $s(i)$ close to -1 is evidence that i has been misclassified.

3.4.2 Graphical representation

We then plot $s(i)$ as bars for each i in A ranked in decreasing order, an example of which is given in Figure 4.7. We stack the plots for each cluster on the same chart for comparison. Wide silhouettes indicate a tightly defined cluster, whereas narrow or negative bars suggest clusters that are not well-defined. The silhouette plot therefore gives an indication of which objects are assigned well to a cluster and those that are not. The average silhouette width for each cluster and the average silhouette of all data points gives an indication of whether k has been chosen correctly. The average silhouette across all data points should be high — we might choose k such that this value is maximised.

3.4.3 Considerations for obtaining the value of k

Silhouette plots only depend on the results of a clustering, rather than on the method used. Therefore silhouette plots can be used to validate a clustering or to attempt to heuristically determine k. By comparing silhouette plots of different clustering algorithms applied to the same data we can quickly compare results and identify poor performers. In our attempts to determine k we might compare silhouette plots for a range of 'reasonable' values of k and look for the value of k that gave the widest silhouettes. We can consider the average silhouette width within a cluster (that is, the average of the $s(i)$ over all i in A) as an indication of the quality of the clusters themselves, so that high values identify clear clusters. We might also consider the overall average silhouette width across the entire plot, which will be different as we vary k. We would reasonably then suggest to pick k such that the overall average silhouette width is maximised over our range of k.

3.4.4 In R

The function `silhouette` is found within the `cluster` library [19]. Given a clustering object `x`, we can call `silhouette(x)` which outputs a silhouette object for analysis, and we draw the silhouette plot using `plot(silhouette(x))` on our silhouette object.

3.5 Hierarchical Clustering Methods

It is useful to obtain a hierarchically-nested set of partitions so that we might observe better the relationships between clusters. Agglomerative algorithms start with each counter in a separate partition. At each step the 'most similar' partitions are joined together. The definition of 'most similar' depends on the clustering criterion being used. Amalgamation continues until all the counters are in a single partition. In some ways, this can be seen as the reverse of the clustering methods we have used previously, and we hope that we can identify relationships between different sets of counters. However, as pointed out in [12], the 'most similar' pair of partitions at a given step may not be unique and how ties are handled can affect the result.
The agglomerative algorithms are step-wise optimal, i.e. the partitions that are joined are the 'best' that could be done at that time. However, it cannot be said that the complete hierarchical classification satisfies any particular optimality criteria for a specified clustering criterion.

3.5.1 Commonly used clustering criteria

Agglomerative algorithms are usually described by a recurrence relation between partitions that have been newly merged, $C_i \cup C_j$, and any other partition C_k. This is defined by Gordon [12] as:

$$\begin{aligned} d(C_i \cup C_j, C_k) = \alpha_i d(C_i, C_k) + \alpha_j d(C_j, C_k) + \beta d(C_i, C_j) \\ + \gamma |d(C_i, C_k) - d(C_k, C_j)| + \delta_i h(C_i) \\ + \delta_j h(C_j) + \epsilon h(C_k) \end{aligned} \tag{3.5}$$

with $h(C_i)$ the height in the valued tree of partition C_i.
One choice of clustering criteria is the 'incremental sum of squares' criteria, or 'Ward's' method [21]. (Further choices of clustering criteria are available in Gordon [12, Table 4.2].) k-means clustering aimed to minimize sums of squares between clusters, so using this criteria will provide a new visualisation of the clustering. The 'Ward' criteria defines the following values needed for the recurrence relation:

$$\begin{aligned} \alpha_i &= \frac{n_i + n_k}{n_+} \\ \beta &= \frac{-n_k}{n_+} \\ \gamma &= 0 \\ \delta_i &= 0 \\ \epsilon &= 0 \end{aligned}$$

with n_i the number of objects in partition C_i and $n_+ := n_i + n_j + n_k$.
At the start, $C_i = i$, $C_j = j$ and $d(C_i, C_j) = d_{ij} (i, j = 1, \ldots, n)$. If the partitions to be joined at a particular step are C_r, C_s then the height in the valued tree of the union of partitions is

$$h(C_r \cup C_s) := d(C_r, C_s).$$

If both C_s and C_r only contain a single point then this is d_{rs} as per the distance matrix, else it is given by equation 3.5. The optimal amalgamation step is the one that leads to the minimum possible increase in the total sum of squared distances around the cluster centroids.

3.5.2 Determining the number of clusters

A method of determining k is described in Everitt [10, p.75], originating in Mojena [22]. The suggested procedure depends on the relative sizes of the 'fusion' levels (that is, the height on the dendrogram where clusters are joined together). In essence we select the number of groups that correspond to the first stage in the dendrogram satisfying

$$\alpha_{j+1} > \overline{\alpha} + cs_\alpha$$

where α_i are the fusion levels corresponding to the stage with $n - i$ clusters, for $i = 0 \ldots n - 1$. $\overline{\alpha}$ and s_α are the mean and unbiased standard deviation of the α values. c is a constant which Mojena suggests should be in the range 2.75 to 3.50 for the best results. However, Milligan and Cooper [13] suggest that 1.25 may be a better value.

3.5.3 Cophenetic correlation

This is a measure of how closely the dendrogram matches the pairwise distances between the original data points, described in [23]. By looking at the tree itself, one gets an idea of how closely related the data points are. Similar points will be grouped close to the bottom of the tree and dissimilar points nearer the top. We are then able to create an implied similarity matrix by looking at the height of the tree where data points are merged together. The cophenetic correlation is therefore the correlation between the values in the original similarity matrix and those in the similarity matrix implied by the tree. We note that original similarity matrix has $N(N-1)/2$ unique values, whereas the implied matrix only has $N - 1$, as a result of the $N - 1$ merger steps in each tree, where N is the number of data points. However, Aldenderfer and Blashfield [23] describes two main problems with the use of cophenetic correlation. Firstly the assumption of normal distributions for the values in both matrices when using the product-moment correlation is generally violated since the choice of clustering method determines the distribution of values in the implied similarity matrix. Hence the use of the correlation coefficient may not be justified. Also the very much smaller number of unique values in the implied similarity matrix leads to a much reduced amount of information in that matrix compared to the original matrix, making cophenetic correlation difficult to justify.

Chapter 4

Application of clustering techniques and results

We now use the theory discussed in the last chapter to obtain clusterings of usage profiles from the cycling counter data.

4.1 K-means

We create for each counter i a vector of length 24 containing the proportion of responses for each of the 24 hours in a day, which is outputted by our function `avgday()`. We put these into a 24×107 matrix `biking.avgdays`. By specifying an option in `avgday()` we can also create matrices of responses for weekdays, weekends as well as splitting the data matrices into rural and urban counters. We then perform k-means clustering using the function `kmeans(x,k)` where `x` is a data matrix and `k` is the choice of k clusters to generate. The k-means function works directly with the data matrix and uses the Euclidean distance metric. It outputs the clustering, as well as additional information on the cluster centres and the within-cluster sums of squares. A sample output from `kmeans(biking.avgdays,4)` is provided here, using a data matrix created from the weekdays data.

```
> kmeans(biking.avgdays,4)
K-means clustering with 4 clusters of sizes 43, 5, 27, 32

Cluster means:
      0     1     2     3     4     5     6     7     8     9    10    11
1 0.003 0.002 0.001 0.002 0.006 0.021 0.039 0.082 0.088 0.046 0.041 0.046
2 0.002 0.001 0.000 0.000 0.000 0.007 0.015 0.056 0.188 0.046 0.032 0.037
3 0.004 0.002 0.001 0.001 0.002 0.007 0.019 0.063 0.126 0.072 0.044 0.042
4 0.003 0.001 0.001 0.001 0.001 0.005 0.013 0.041 0.052 0.043 0.053 0.068
     12    13    14    15    16    17    18    19    20    21    22    23
1 0.053 0.058 0.064 0.077 0.094 0.101 0.063 0.042 0.029 0.021 0.015 0.007
2 0.038 0.046 0.050 0.158 0.087 0.083 0.053 0.043 0.027 0.018 0.010 0.005
3 0.049 0.052 0.052 0.063 0.088 0.123 0.079 0.044 0.027 0.018 0.012 0.008
4 0.077 0.081 0.087 0.090 0.092 0.095 0.075 0.059 0.035 0.015 0.007 0.003

Clustering vector:
1047 1063 1069 1073 1078 1096 1097 1098 1099 1112 1113 1127  113 1217
   1    4    2    1    1    4    3    1    3    1    4    3    4    4
1218   12 1221 1222 1223 1225 1245 1246 1247 1248 1249 1250 1251 1254
   4    4    1    1    1    4    1    1    1    4    1    1    1    1
1255 1256 1257 1258 1261 1262 1263 1264 1265 1267 1268 1269 1270 1289
   1    1    1    4    1    4    1    3    1    4    2    1    1    3
1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1304 1305
```

```
   3    3    1    4    4    3    3    3    3    3    3    3    1    4
  13  132   14 1469 1510 1513 1514 1515   15   16 1626 1686 1725 1726
   3    4    3    3    1    1    1    2    3    4    1    3    1    4
1727   18   19    2  256  260  320  438  439  441  469  470  478  527
   1    3    3    1    1    3    4    3    3    3    1    1    4    3
 528  597  636   66  691  692  693  694  695  729  765  827  828  829
   1    2    4    4    4    4    4    4    4    1    4    1    1    1
 830  831  833  834  838  855  912  973 1272
   1    4    3    4    1    4    4    2    1

Within cluster sum of squares by cluster:
[1] 0.14 0.025 0.072 0.11

Available components:
[1] "cluster"  "centers"  "withinss" "size"
```

4.1.1 Visualising the clustering

In order to better visualise the clustering, we project the data into 2-dimensions using principal component analysis. By considering the scree plot, this projection accounts for approximately 9% of the variation observed, so it is not very representative. We then colour the points according to cluster. From Figure 4.1 we see that the data do not form any distinct clusters. The k-means algorithm tries to create roughly spherical clusters — the data points exist in 24 dimensions, so it is not immediately clear from our 2-d plot whether there is any basis for the clustering at all!

4.2 Analysis of weekday clustering

We consider the average weekday profile for each cluster as a whole. We do this by averaging the day-profiles for each counter and drawing a graph of the results. This provides us with an idea of the sort of differences that the clustering algorithm has identified between groups of counters. Looking at Figure 4.2 we see the four different shapes. We use our intuition to label each graph. We suggest that the red graph represents counters that are near schools and thus the majority of users are people travelling to and from school. We see peaks at 9am and the highest peak at 4pm, which correspond well with the starting and ending times of most schools. (We do however note that private/independent schools often operate different hours, particularly in the evening, so our analysis would not necessarily identify a school counter in this case.) We suggest the green graph as being representative of commuter routes, with the same 9am peak as schools but the afternoon peak later on at 5pm and still seeing fairly high usage at 6pm. The black graph is characteristic of the usage patterns we might expect to see on leisure routes. There is a slight peak (compared to the surrounding hours) at 9am, suggesting that people do still use the leisure routes for commuting in some cases. We see increasing usage after 11am to a slight peak at 5pm. We suggest that leisure users would tend to take cycle rides during these hours. We conclude with the blue graph, which we propose as a hybrid usage pattern between commuter and leisure routes. The morning and evening rush hour peaks are less well defined than with commuter routes and we see slightly higher afternoon usage. Another possible explanation is that what we are seeing is people taking cycle routes that lead them to shops on their way to or from work or school, extending their journeys by perhaps an hour or so, leading to the flattened peaks that we see. However, as the counters themselves lie at fixed points on the cycle routes and we do not know which counters are in some way 'linked' to one another, it is difficult to make such assumptions. The results are roughly in line with what Sustrans would expect and have previously observed.

These results are further reinforced by looking at Figure 4.3. In this plot, we find the mean weekday usage profile averaged across all of the counters and then subtract this from each of the cluster usage profiles. The plot then shows us the differences from the mean that characterise each cluster profile. The plot is naturally centred around zero and we can see the times of the day when clusters have

Figure 4.1: 2 dimensional representation of k-means results for 4 clusters

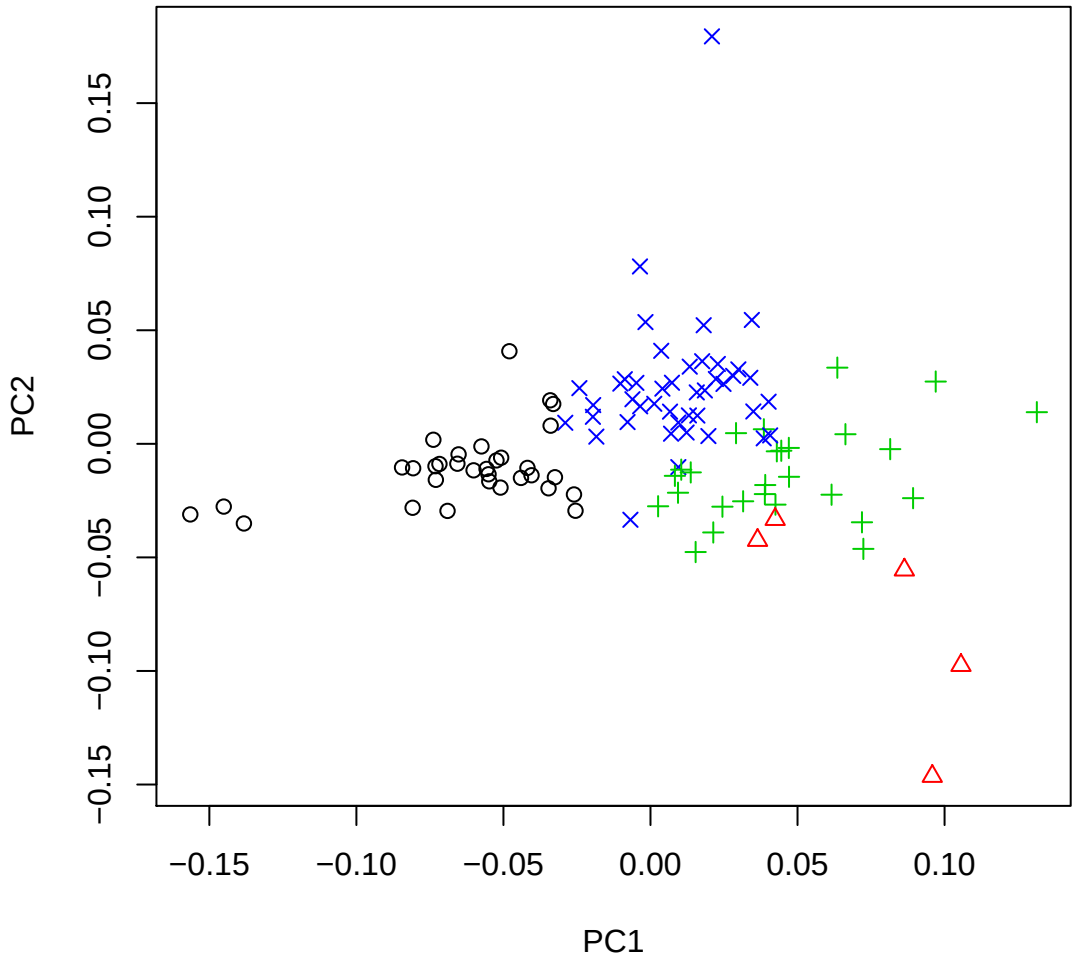


Figure 4.2: Average day plots of 4 clusters

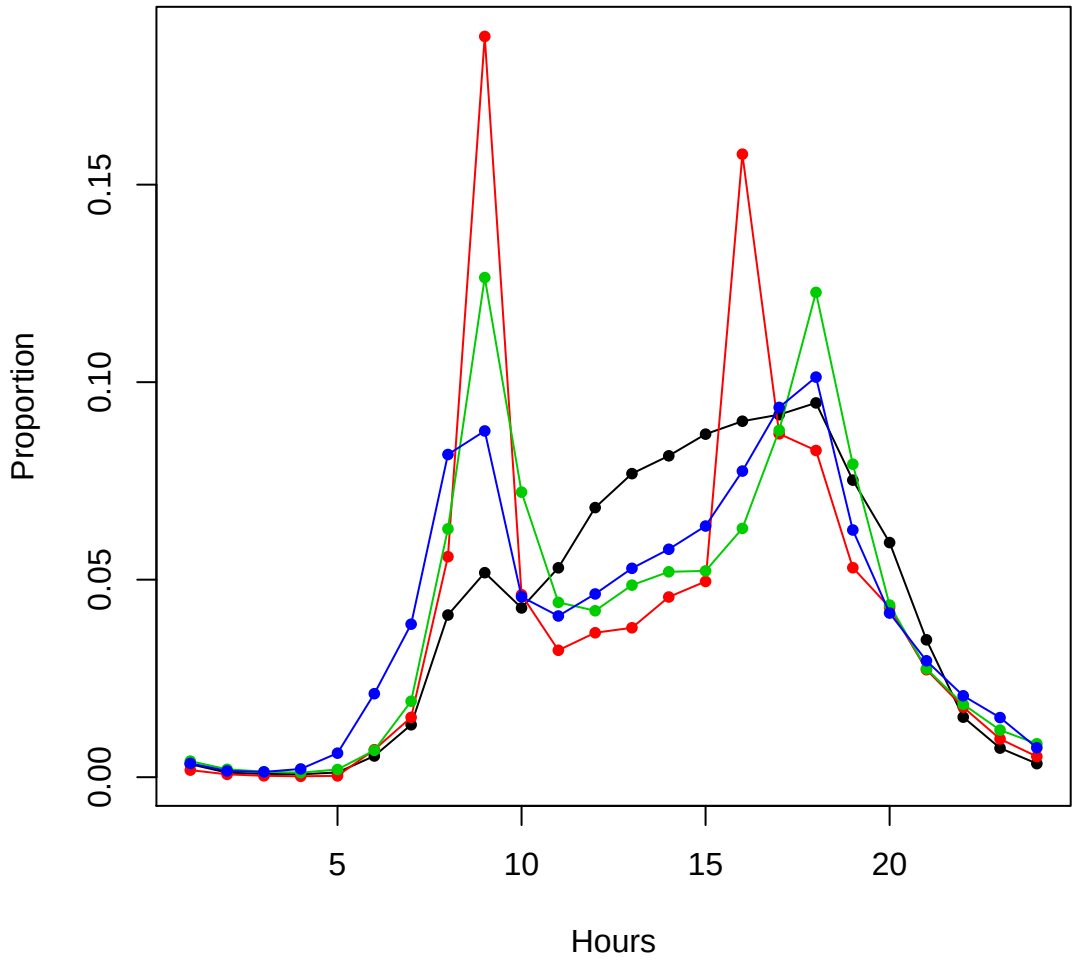
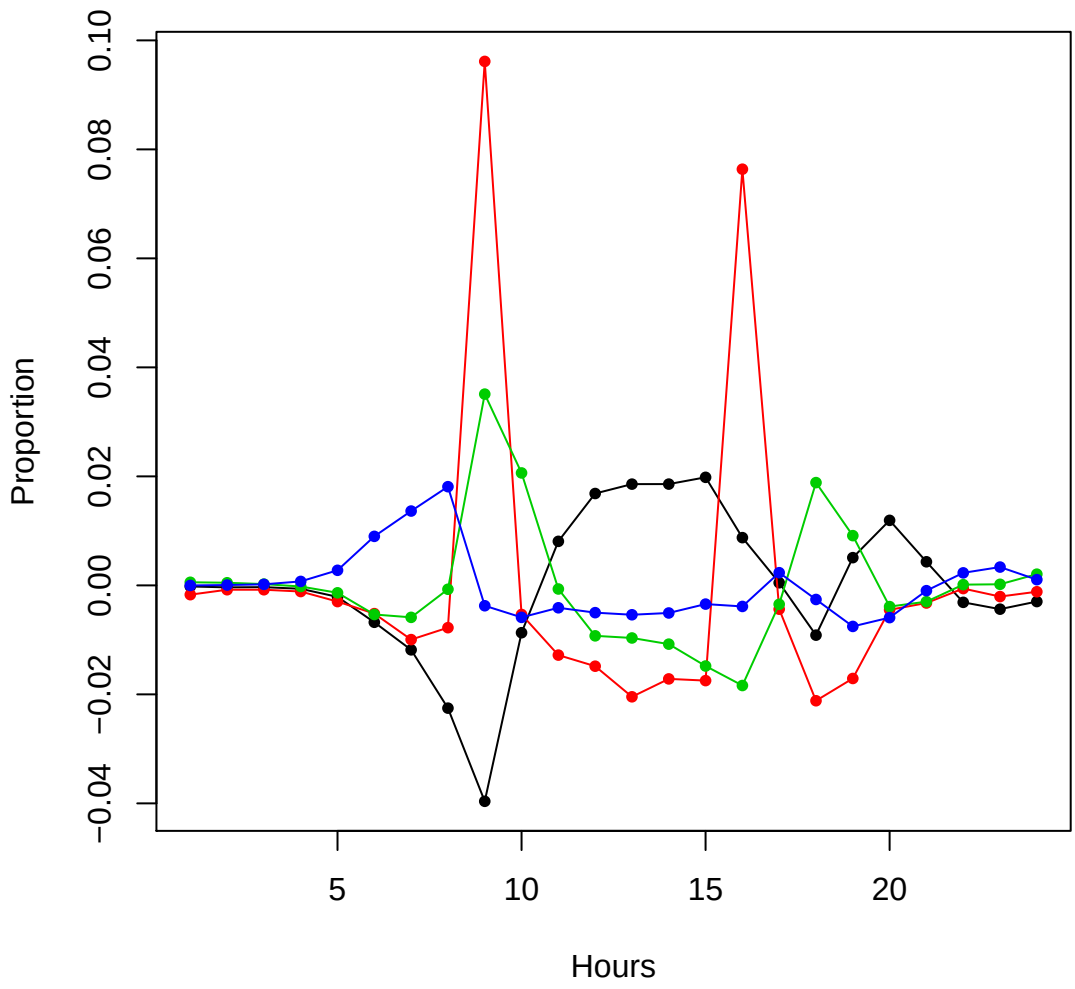


Figure 4.3: Difference from average plot for 4 clusters



above average or below average usage. We see that schools and commuter counters have much higher than average usage at their respective peak times of day. The leisure route is characterised by much lower usage in the morning 'rush' hour time, but then above average usage during the middle of the day and early evening. It is interesting to note that whilst the hybrid counters appear to represent the 'most average' usage patterns, hybrid counters also see above average usage between 5am and 9am.

4.3 Comparing rural and urban counters

Sustrans provides information about whether they consider a counter's location to be 'rural' or 'urban'. Intuitively, we might expect to see more 'leisure' type routes in rural locations compared and more commuter routes in urban locations. We use this information to split up our `biking.avgdays` matrix into two smaller matrices. There are 84 urban and 23 rural counters, so urban counters will have a strong influence upon the combined results. We then perform a cluster analysis as previously described on each part and compare the results. Using 4 clusters in each case, we create the average day profiles for each cluster, shown for both rural and urban counters in Figure 4.4.

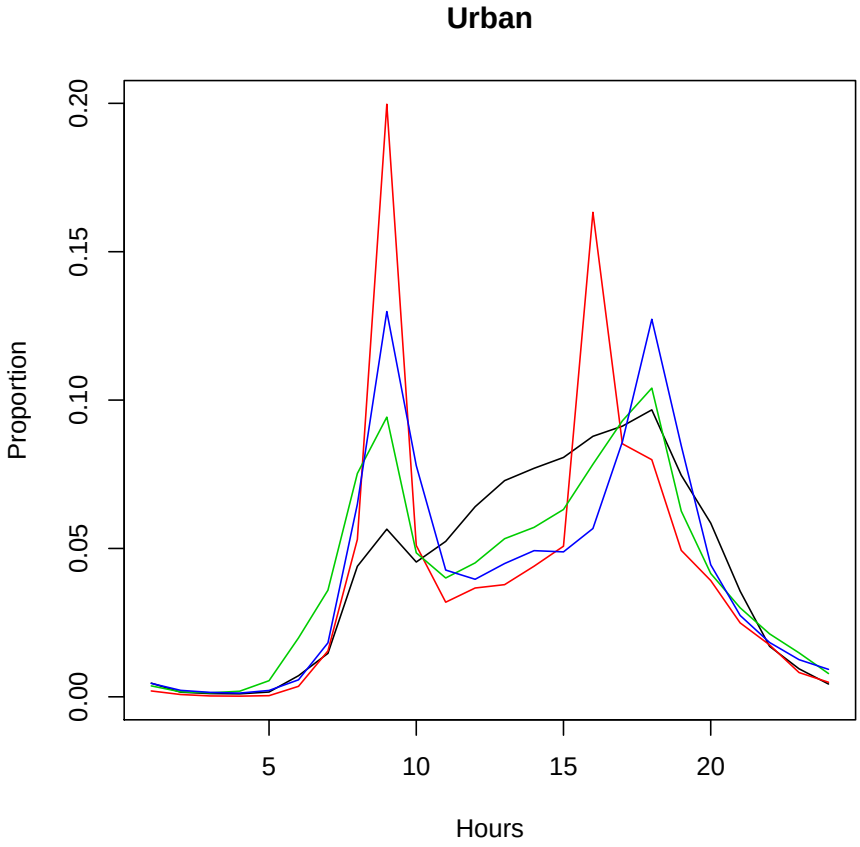
Looking at the urban clustering, we see roughly the same 4 shapes that we saw after applying the clustering to all the data. This is to some extent not unexpected given the way that urban data dominates the dataset. The leisure routes have something of a reduced 'hump' between 12pm and 4pm, but other than this the results are pretty much the same.

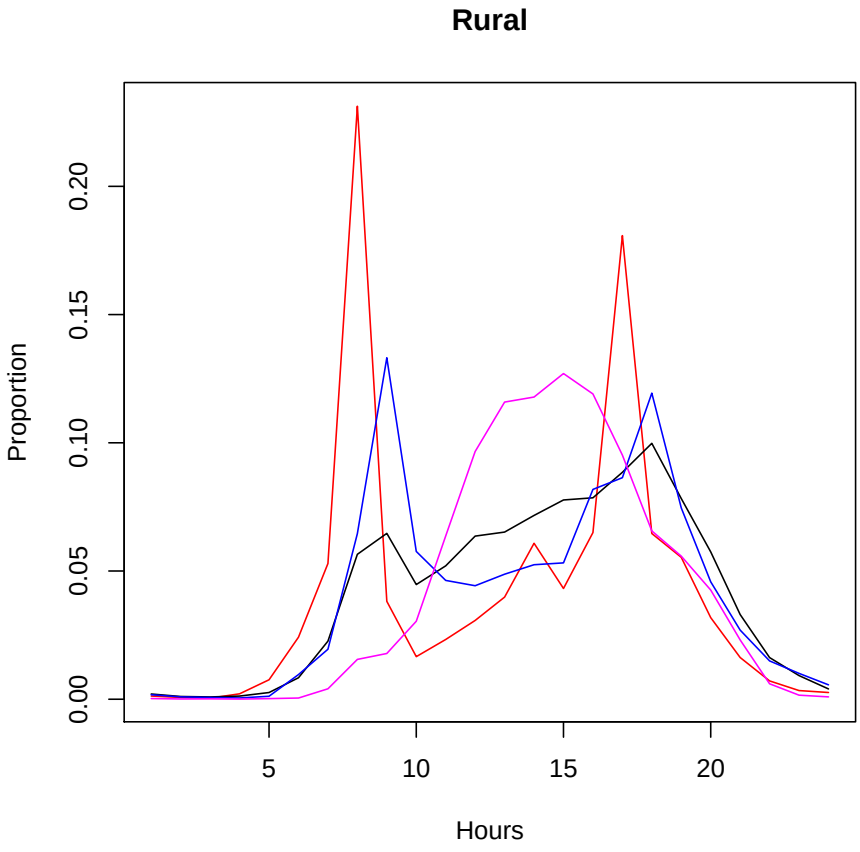
The rural results demonstrate some interesting features. The schools result has peaks further apart, at 8am and 5pm. However, a closer examination of the output of the algorithm reveals that this 'cluster' only contains a single point. The shape still shows the same dominant peaks as the schools result when looking at all the data combined, and indeed the slight peak around lunchtime might confirm that the counter is on a school route. The shape is considerably different from the others in the plot; but applying the clustering techniques to such a small dataset means that single element clusters are more likely and potentially more unsatisfactory. In the rural plot we still see the commuter type routes with the same flat usage characteristic of the hours in the middle of the day. We also see the same shape that we had described as 'leisure' when examining the data as a whole. However, there the fourth cluster created is that of a further type of leisure route, seeing most usage between 12pm and 4pm. This is an interesting finding — however, this cluster only contains 3 counters. When the data as a whole is considered, these 3 counters are in the leisure category. Some experimentation finds that we need to use 6 clusters in order to retrieve this shape when using the whole dataset and that no urban counters get added to this cluster. We therefore propose that we consider these 3 counters in the same leisure category as those previously identified. Noting this, we also propose that the rural and urban counters show the same types of counters, and that there is little to be gained from considering the rural and urban counters separately. It is therefore reasonable to work with both urban and rural counters in a single dataset.

4.4 Weekends

We apply the same sort of analysis used to compare rural and urban counters to try and understand the differences between usage on weekdays and weekends. By modifying `avgday()`, we allow that function to output average day profiles from weekdays, weekends or both. We similarly run a cluster analysis on both new data matrices and compare the results. The weekday clustering reveals only one type of shape, with the clustering only picking out slight changes in the width of the shape. This suggests that weekend usage at our counters is mainly leisure route type activity — which we would intuitively expect. Using the clustering generated for our counters from the weekday data we can plot the day profiles for weekend usage averaged across each cluster, to see if we can see any of the differences that are so apparent in the weekdays data. Indeed, as seen in Figure 4.5, we cannot. The colouring of the cluster graphs are the same as in Figure 4.2, i.e. black is leisure, blue is hybrid, red is schools and green is commuter. All categories of route show the same basic shape (particularly hybrid,

Figure 4.4: Comparison of rural and urban clustered into 4 clusters





commuter and schools) and the leisure cluster is the same shape but squeezed. This result confirms to some extent the intuitive results that we see for the weekdays plots as we might expect most weekend users to be cycling for leisure. The leisure clustering is slightly unusual however in that it is a narrow shape – this is somewhat intuitive, as we might still expect some commuter usage on the other types of routes at the weekend, widening the shapes, whereas usage on leisure routes should be even more apparent at weekends. However, it remains the same basic shape, and we conclude therefore that the weekend data is not particularly informative and this explains the focus upon weekday data in the rest of this chapter.

4.5 Choosing k

In addition to the formal methods of identifying k stated in the previous chapter, we can informally look at a graph of the sum of within-cluster sums of squares as we increase k over a range of values. We create a small function to output these values and we plot them in Figure 4.6.

```
sumsquares <- function(datamatrix) {
wss <- c()
dimension <- dim(as.matrix(datamatrix))[1]
wss <- (dimension-1)*sum(apply(as.matrix(datamatrix),2,var))
for (i in 2:10) {
W <- sum(kmeans(datamatrix,i)$withinss)
wss <- c(wss,W)
}
return(wss)
}
```

We look for an 'elbow' in the graph, which might suggest a good number of clusters to pick. In Figure 4.6, 4 clusters appears to be a reasonable choice. We also use R packages for some of the formal methods of determining k as described in Section 3.2.3. The Calinski Harabasz index can be found using `cascadeKM()` within the package *vegan* [24].

```
> cascadeKM(biking.avgdays,2,10,iter=25,criterion='calinski')$results
           2 groups   3 groups  4 groups   5 groups   6 groups   7 groups   8 groups
SSE       0.4994083  0.4163868  0.349085  0.3107621  0.2772478  0.2564095  0.2383023
calinski 40.2405087 34.2701624 33.608781 31.1847744 30.1312184 28.2356608 26.8551887
           9 groups  10 groups
SSE       0.2206769  0.2026934
calinski 26.0971835 25.9541559
```

We recall that the value of k for which the index is largest is the 'best' value of k. The output of course suggests that $k = 2$, however, from experimentation we see that we can pick out 4 different shapes. This level of detail is lost if we choose $k = 2$ and splits the counters roughly into commuter and leisure groups. It is quite possible that this is the case, but it is not particularly interesting.

4.6 Silhouette plots

As described in Section 3.4, we can create silhouette plots in order to analyse the weekdays clustering. The silhouette plot for a clustering of $k = 4$ is shown in Figure 4.7. We see that the clusters are not particularly compact, with an average silhouette width of 0.23. Cluster 2 is the 'schools' cluster, and here we note the only example of a counter with negative silhouette. This counter is one that should have been clustered elsewhere, thus demonstrating that the k-means algorithm only finds local optima, rather than the global optimum. Cluster 1 has the highest silhouette average (and indeed contains the counter with largest s_i) and is likely to be the clearest example of similar average day profiles.

Figure 4.5: Clustering from weekdays applied to weekend data

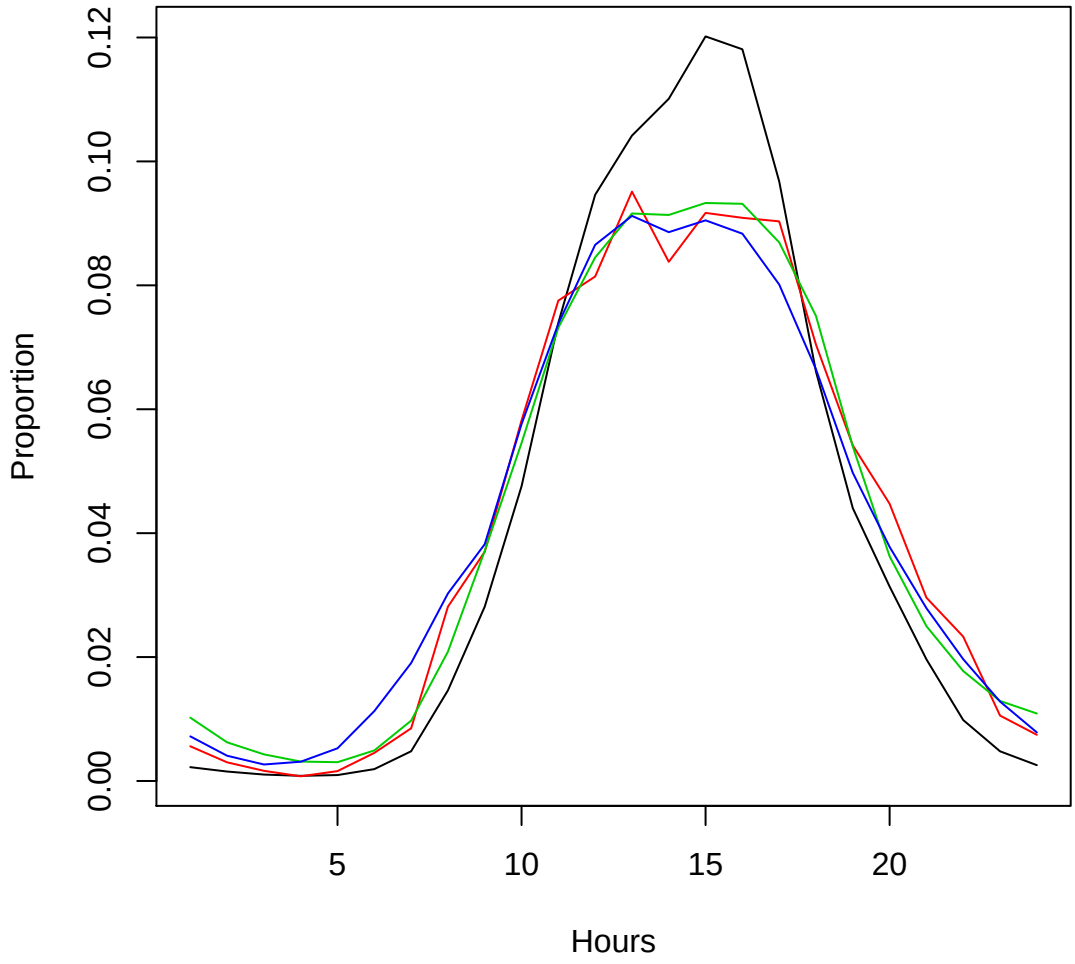


Figure 4.6: Sum of within-cluster sums of squares for varying k

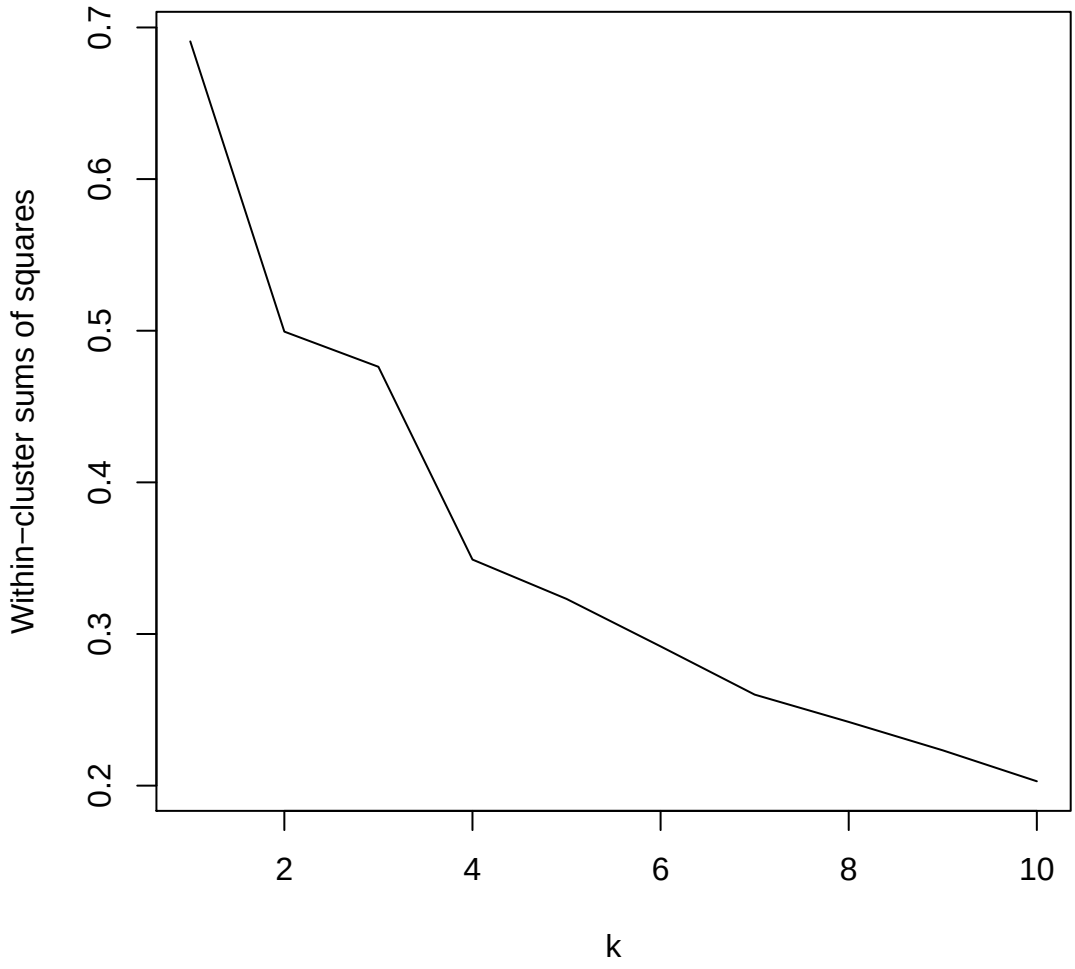
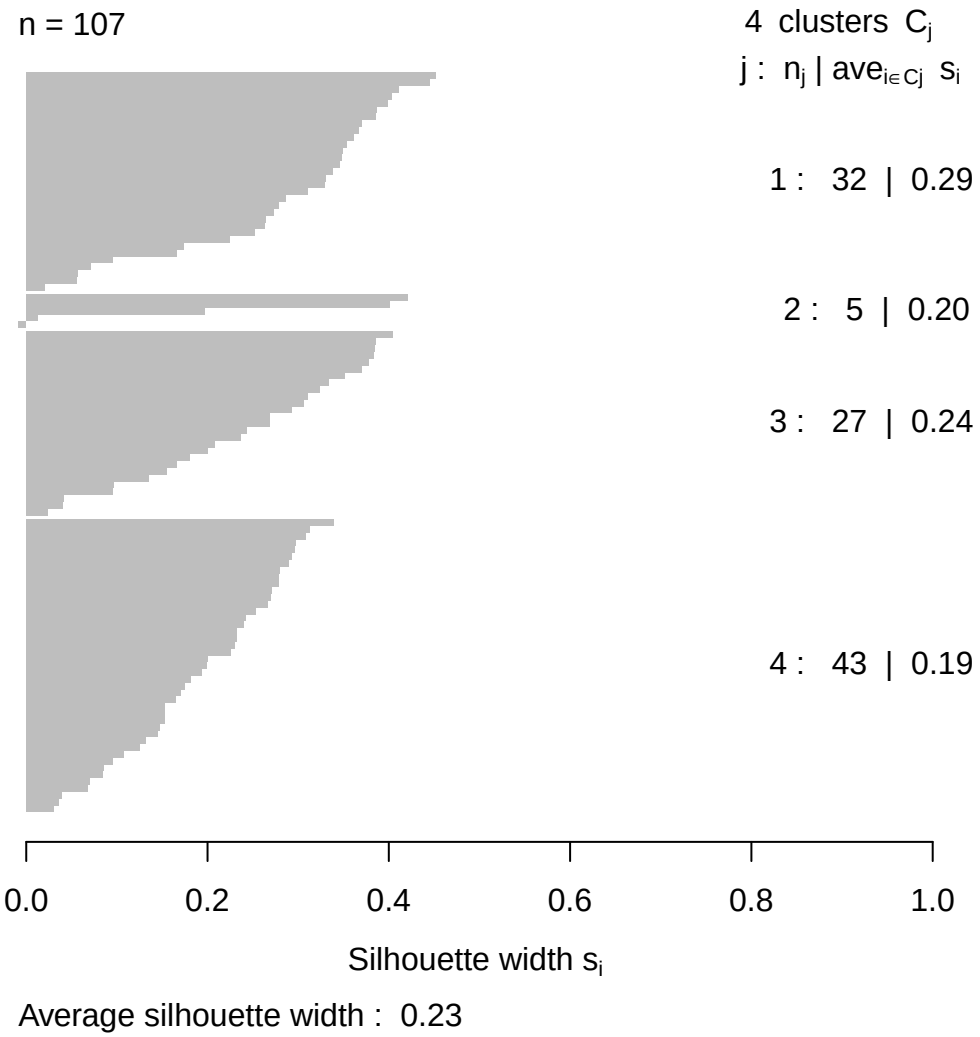


Figure 4.7: Silhouette plot for 4 clusters



We tabulate in Table 4.1 the average silhouette width across all data points for varying k in order to try and find the largest and therefore 'optimal' value of k. We see that there is some evidence for

k	Avg s_i
2	0.264
3	0.231
4	0.234
5	0.160
6	0.190
7	0.160
8	0.158
9	0.160
10	0.159
20	0.160

Table 4.1: Average silhouette widths for varying k

selecting $k = 2$ clusters, however, comparing the average day profiles of 2 clusters with those when using 4 clusters it is easy to see that there is additional, useful information to be gained by choosing to use 4 clusters.

4.7 PAM

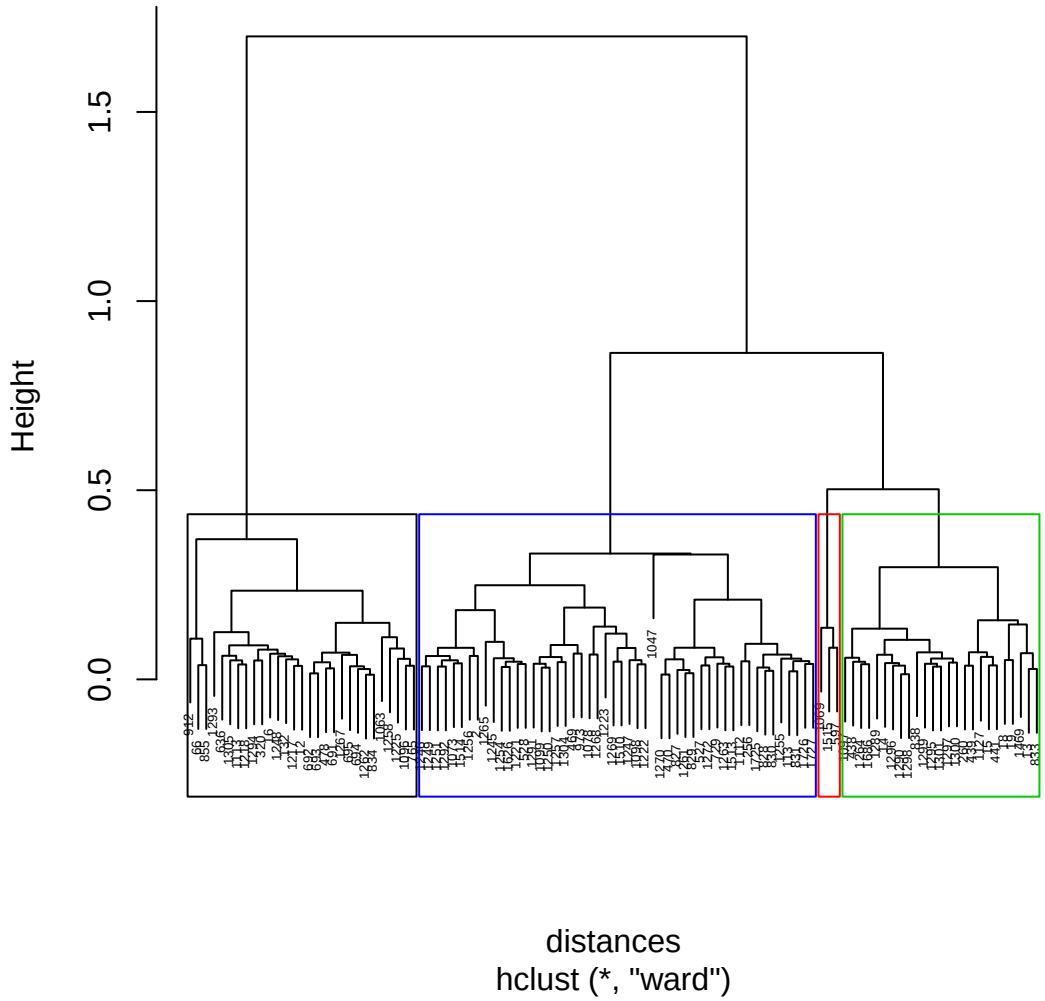
Unlike the `kmeans()` function, `pam()` requires a distance matrix to have been created first. This allows some flexibility in choosing a suitable distance metric, however, trials with both the Manhattan or Canberra metric provided no further insight compared to the standard Euclidean. The algorithm produces similar results to the k-means, producing roughly the same average day plots across each cluster (similar to that seen in Figure 4.2). However, upon closer examination of the silhouette plot, we notice that the number of counters with negative s_i is 8 for PAM compared to 1 for k-means. Negative s_i suggests inappropriate clustering and therefore we should conclude that the results produced by PAM are less sensible. This may be as a result of the small number of counters we are using: since the cluster centres must be actual data points, rather than the actual cluster means in space, we might expect a less effective clustering because the cluster centres only have a limited number of places they can move to. For datasets with more data points for each cluster we would expect more flexibility in the choice of the cluster centre and hence better clustering. It is still of interest to see the average day profile for the medoids for each cluster finalised by the algorithm. They are to some extent the 'most representative' counter of each cluster. Cluster 1 is counter 1261, cluster 2 counter 834, cluster 3 counter 1515 and cluster 4 counter 1290.

4.8 Hierarchical clustering

We perform a hierarchical clustering using the R function `hclust()` on a distance matrix outputted by `dist()`. We are then able to output a clustering from the tree for a specified number of clusters using `cutree(tree, k)` for our choice of k. We demonstrate this using Ward's method as this method is most similar to the way that the k-means and PAM algorithms work, giving us a different visualisation of the links between different clusters.

```
> distances <- dist(biking.avgdays)
> hclust.out <- hclust(distances, method="ward")
> plot(hclust.out)
> cutree(hclust.out,4)
1047 1063 1069 1073 1078 1096 1097 1098 1099 1112 1113 1127  113 1217 1218   12 1221
   1    2    3    1    1    2    4    1    1    1    2    4    1    2    2    2    1
...
```

Figure 4.8: Hierarchical clustering using Ward's method



Plotting average day graphs of the clustering given by Ward's method yields graphs of the same shapes as Figure 4.2. We might assume the hierarchical tree shown in Figure 4.8 allows us to see the links between different clusters that k-means generates. We note that the schools cluster contains only 3 counters, instead of the 5 found by k-means. One of those 5 found by k-means has negative silhouette, so there is some evidence that there are between 3 and 5 actual schools counters, and that the different clustering methods struggle to accurately to define this group.

4.9 Year profiles clustering

We can easily apply the techniques presented in this chapter to the year profiles for each counter. We use `cascadeKM` to generate a reasonable value of k, as we don't really have any expectation for how many different shapes there might be. We apply this function to our matrix of average year profiles `biking.yearavgs`, where each row is a different counter. As some counters do not have data for a complete year, we remove these rows using `na.omit`. We find that $k = 2$ is a reasonable choice. We can run `kmeans` in the usual fashion for two clusters and then plot the average graphs for each cluster, presented in Figure 4.9. (The functions `yearprofile` and `drawclusterplots` are available in Appendix A.) We see that we have two similar shapes, but with characteristics that allow us to intuitively understand them. The black graph cluster probably represents leisure users, since we see much higher usage during the warmer months, rising to peak during the holiday months of July and August, then falling away sharply. The red graph is likely to be representative of school and commuter routes, where we see much less of a peak during the warmer months. It is particularly interesting to note the dip in usage in August as this is exactly what we would expect on routes that are primarily used by commuters and those riding to school.

```
biking.yearavgs <- t(sapply(unique(biking$counter),yearprofile))
row.names(biking.yearavgs) <- unique(biking$counter)
biking.yearavgs <- as.data.frame(na.omit(biking.yearavgs))
names(biking.yearavgs) <- c(1:12)
drawclusterplots(kmeans(biking.yearavgs,2)$cluster, biking.yearavgs)
```

We can also plot the average year profiles for each of the daily usage clusters. Using the clustering that gave us Figure 4.2, we see in Figure 4.10 the year profile for each of those clusters, following the same colour scheme. As we might expect, we see increased usage during the warmer months in all the clusters. Most noticeably we see much lower usage for schools routes in August, which tallies with our intuition. Leisure routes in black show the effect of better weather on usage, peaking in August, when we also expect to see people taking time off of work and possibly cycling.

Figure 4.9: Average year profiles clustered into 2 groups

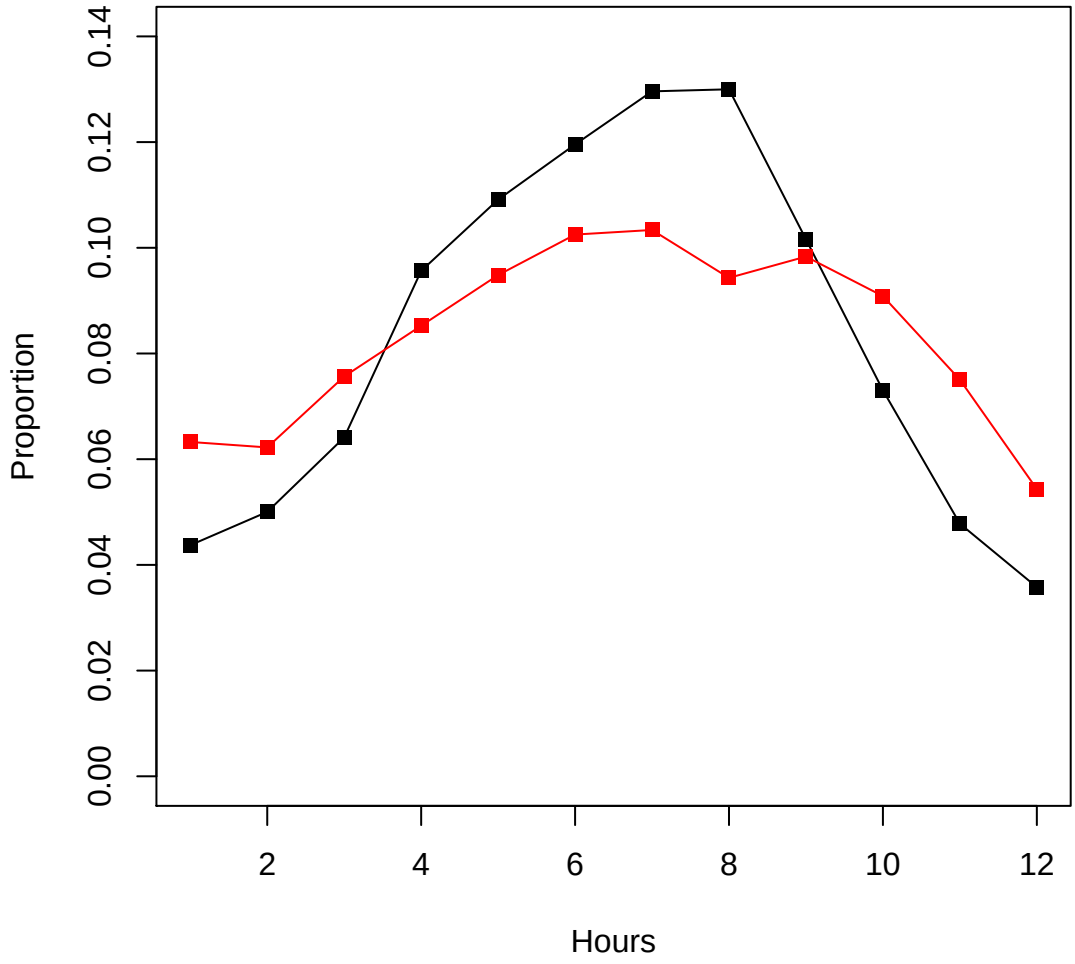
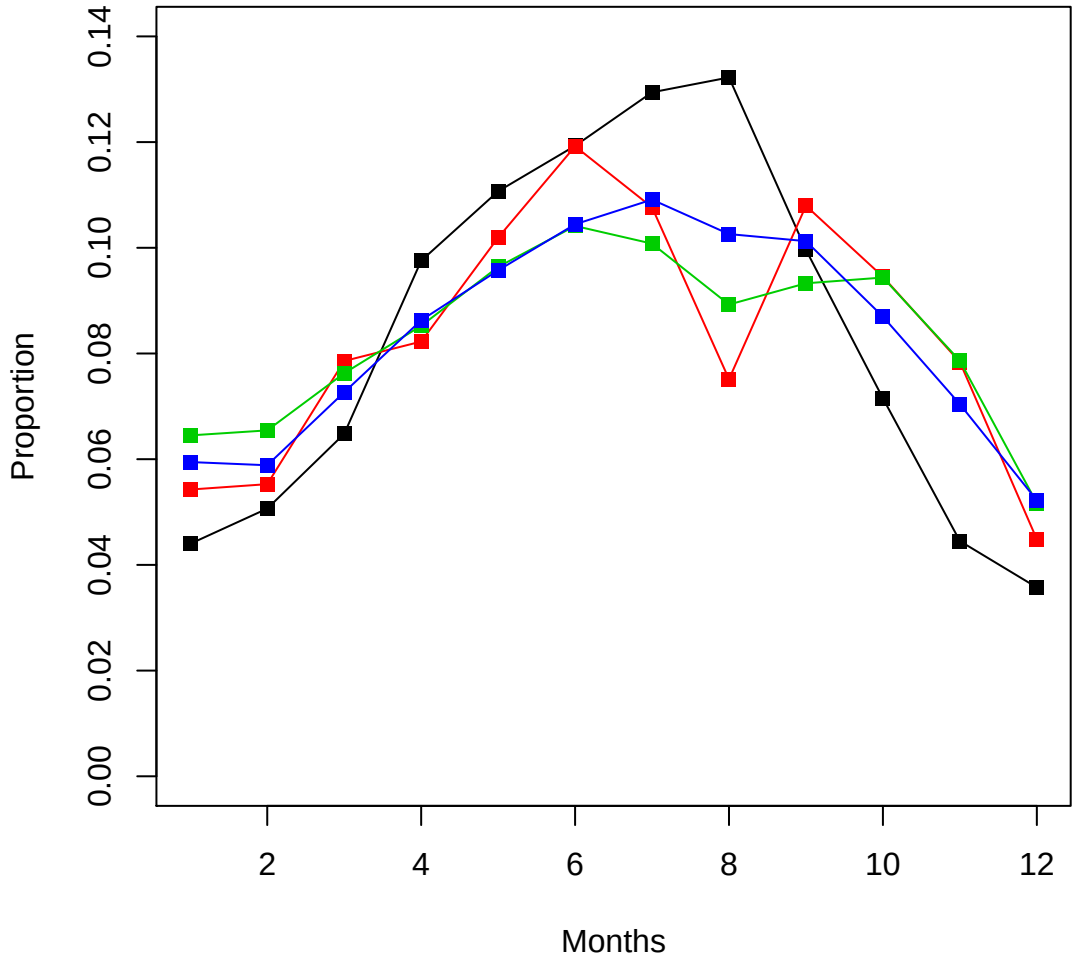


Figure 4.10: Year profiles for the 4 day profile clusters



Chapter 5

Prediction

We will consider as fixed the classifications of weekday data, as shown in Figure 4.2, and year data; as shown in Figure 4.10. We now use explanatory variables in order to try and predict what category a counter will be in for known values of these explanatory variables. We first analyse independence of explanatory variables and categorisation. We then use multinomial logit models to build a model for any dependence between the variables and the categorisation. We also use linear discriminant analysis techniques to predict counter classification for new counters from the results of our cluster analysis.

5.1 Available explanatory variables

We are provided with information from Sustrans about the location of each counter. For instance, we are given information about whether a counter is on a route that is surfaced, whether a location is traffic-free or not (and if not, what sort of road it is on), whether the counter is on a route that is lit at night, and so on. We will then attempt to find out whether the values of these explanatory variables affects the classification of a counter into one of the 4 categories suggested in the previous chapter. For instance, we might intuitively expect that leisure users would want to ride on traffic-free routes away from other road users and commuters using roads in towns and cities to reach their place of work. This chapter will attempt to find links between these explanatory variables and counter classification. The available explanatory variables are (using mainly Sustrans labels, and in no particular order):

```
Sharedcyclebuslane              NCN
Unmarked.cycle.route            Onroad
Designatedandmarkedcycleroute   Urban
Sharedcyclebuslane              Aroad
Routeadjacenttoroad             Broad
Sharedusepathkerbseparated      Croad
Cycleroutеonlykerbseparated     Footpath
CycleroutеonlyFenceseparated    Surfaced
Unclassifiedroad                Cycletrack
Trafficfreeroute                Othertrack
Railwaypath                     Canalriverside
Coastalpromenade                Bridleway
Cycletrackonfootpath            Lightingno
Lessthan3miles
```

We also have available a rudimentary regional classification of each counter into one of 'north', 'south' and 'midlands'. Whilst we could theoretically use this in our analysis, it would be difficult to describe this as an explanatory variable, as it is not something that Sustrans could easily manipulate in order to promote different types of usage. For instance, one could analyse the potential effect of changing the lighting on a particular route, but it would not make sense to look at the effect of a change in the region of a route. We also observe that the distribution of the different categories of counter from our

sample is not consistent across the regions.

```
           midlands north south
  commuter        1     8    18
  hybrid         24    16     3
  leisure         7    18     7
  schools         1     2     2
```

One would probably not expect to see such a high percentage of hybrid counters in the Midlands if we considered all possible counters, so it would be incorrect to draw any conclusions about the proportions of any of the categories of counters across the regions.

5.2 Fisher's exact test

Fisher's exact test is especially useful for contingency tables where the sample size is small. Fisher's test can be used where the observed values in cells in the contingency table are less than 5, unlike the χ^2 tests. The test is exact because one is able to compute the precise significance of the deviation from the null hypothesis instead of using approximations.

5.2.1 Basis

We consider the following 2x2 contingency table, with n observed responses in total.

```
       x1  x2  total
 y1    a   b    a+b
 y2    c   d    c+d
total a+c b+d    n
```

The probability of observing this set of values is given by the hypergeometric distribution.

$$p = \binom{a+b}{a}\binom{c+d}{c} / \binom{n}{a+c} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}$$

p defines the probability of observing this set of values given the marginal totals under the null hypothesis that both `x1` and `x2` are equally likely. We create a one-tailed p-value by summing the probabilities of observing data more extreme than those we have seen. A two-tailed p-value can also be obtained by adding in the probabilities of observing more extreme data, but in the other direction.

5.2.2 Application to our data

Fisher's test generalises to $M \times N$ contingency tables, although we only require $2 \times N$. We calculate the p-value for each contingency table that compares the classification of a counter to the binary value of an explanatory variable. Each observed counter can only appear in one cell in the table. An example contingency table is given below.

```
     Classification
Urban commuter hybrid leisure schools
    0        4      6      12       1
    1       23     37      20       4
```

This table gives us a p-value of 0.071. We easily create a function in R that evaluates the test for all explanatory variables.

```
> for (i in 1:28) {
+ pval <- signif(fisher.test(table(newClassFisher[,i],
+ newClassFisher$classification))$p.value,3)
+ cat(paste(sep="",names(newClassFisher)[i],":",pval,"\n"))
+ }
```

Variable	p-value
NCN	0.0905
Urban	0.0713
Onroad	0.429
Unmarked.cycle.route	1
Designatedandmarkedcycleroute	0.231
Sharedcyclebuslane	0.598
Routeadjacenttoroad	0.000125
Sharedusepathkerbseparated	0.000513
Cyclerouteonlykerbseparated	0.0164
CyclerouteonlyFenceseparated	0.299
Aroad	0.00264
Broad	0.416
Croad	0.553
Unclassifiedroad	0.557
Trafficfreeroute	0.002
Railwaypath	0.747
Canalriverside	0.237
Othertrack	0.299
Cycletrack	0.279
Footpath	0.0538
Coastalpromenade	0.00215
Cycletrackonfootpath	0.338
Bridleway	0.121
Surfaced	0.00114
Lightingno	0.00502
Lessthan3miles	0.0191

Table 5.1: p-values for Fisher's exact test comparing variable with counter classification

which leads to the table of p-values tabulated in Table 5.1. We compare the p-values indicated to some arbitrary value, say 0.05. A low p-value indicates an association between the value of the explanatory variable and the categorisation of a counter, although it does not indicate in which direction this association occurs. Looking at Table 5.1 we see therefore that the test suggests that whether a route is adjacent to a road has the most significant effect on counter classification. Similarly, whether a route is shared use and kerb separated from the road is highly significant; as is whether the route is surfaced, on a coastal promenade, Aroad or traffic free. Equally, we can see that variables such as a whether a route is on a B or C road, or on a railway path have a weak association to classification. We cannot, however, directly infer the direction of the change from independence in the cell counts – that is, for example, whether routes adjacent to roads have fewer leisure counters than expected. For this, we look at the standardised residuals, which compare the expected cell counts to those observed.

5.2.3 Standardised residuals

Although we are not able to perform a χ^2 test because of the small cell counts in the contingency tables (particularly as a result of the small number of counters classified as schools), we may still find it helpful to look at the standardised residuals resulting from such a test. These will give us an indication of the direction of the non-independence of the explanatory variables. We define a standardised residual $e_{i,j}$ for a cell (i, j) as follows

$$e_{i,j} = \frac{O_{i,j} - E_{i,j}}{\sqrt{E_{i,j}}}$$

The standardised residuals are related to the Pearson χ^2 statistic, since we have the relation

$$\sum_{i,j} e_{i,j}^2 = X^2.$$

The $e_{i,j}$ are asymptotically normal assuming our model of independence holds so we can compare them to standard normal 'reference points' e.g. ± 2. The tables of observed and expected counts are given below for the comparison between a counter's category and whether or not it is on an A road.

OBSERVED

Aroad	commuter	hybrid	leisure	schools
0	16	34	31	4
1	11	9	1	1

EXPECTED

Aroad	commuter	hybrid	leisure	schools
0	21.4	34.2	25.4	4
1	5.6	8.8	6.6	1

This leads to the following table of residuals.

Aroad	commuter	hybrid	leisure	schools
0	-1.18	-0.03	1.11	0.01
1	2.31	0.05	-2.18	-0.03

We can also visualise this using a mosaic plot, as in Figure 5.1. Here the size of the boxes represent the proportions observed in each counter category where counters are on A roads or not. The boxes are then shaded according to the residual value. Red suggests negative residuals, and blue positive. In this plot, we see that there are more commuter routes on A roads than expected, and fewer leisure routes on A roads. This tallies with our own intuition – we might expect people who ride for pleasure to want to do so away from cars and other traffic; whereas A roads provide a fast, direct route to work for people commuting into towns. Looking at mosaic plots for other variables provides similar results. Using the values in Table 5.1 we identify explanatory variables with significant association between variable state and counter classification and then examine the mosaic plot for that variable. Looking at the plot for whether or not a route is lit or not, we see a strong tendency towards unlit counter sites being classified as leisure. For whether or not a route is traffic free, we observed significantly fewer than expected leisure counters on routes with traffic, suggesting that leisure routes are usually on traffic free routes. Whilst these results appear obvious, it is important to note that these have come entirely from the data.

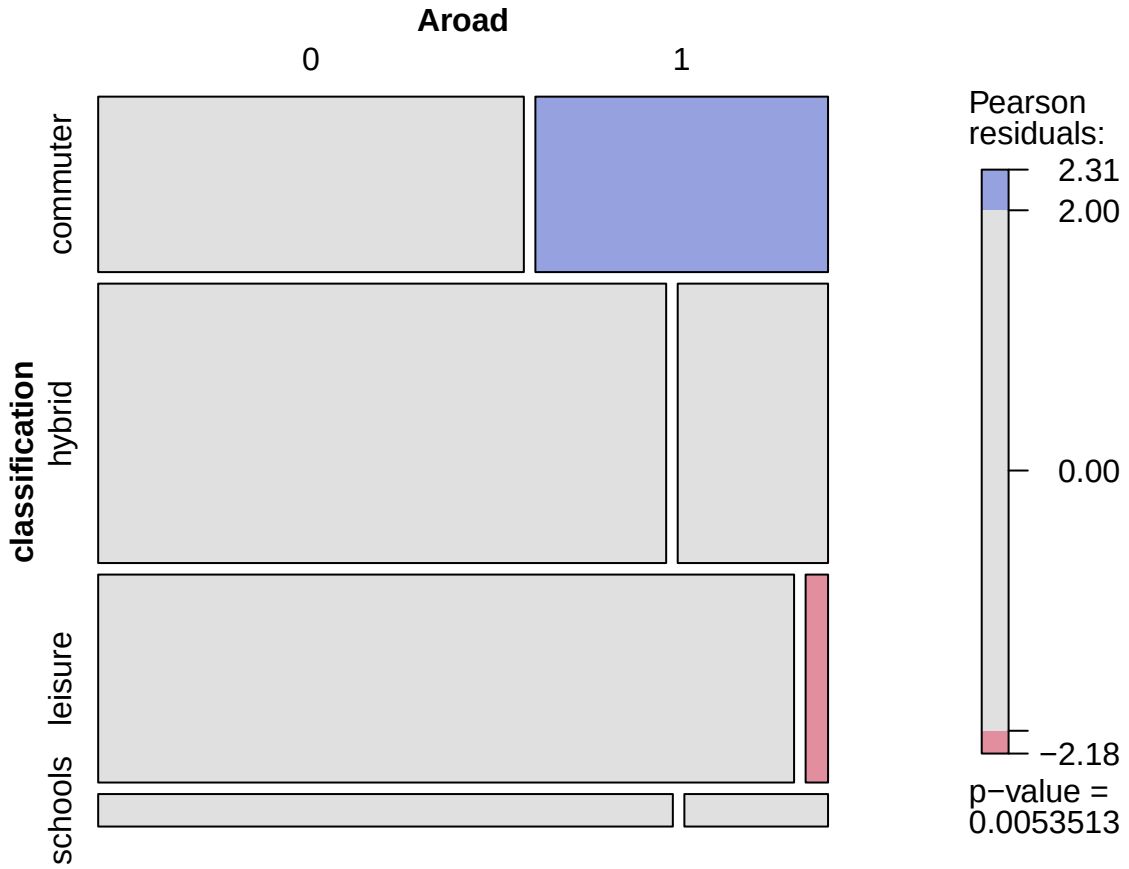
The analysis presented in this section will be useful to Sustrans. It provides a quick, visual way of identifying association between a counter classification and explanatory variables, as well as the potential direction of any association. Indeed, they need not use our counter classification from our clustering – as long as Sustrans can somehow classify counters into groups, they can then check the association of this classification to any explanatory variables that they believe to affect the classification. With high enough contingency table cell counts, they may of course use either Fisher's exact test or the standard χ^2-test for association, and then use mosaic plots to check the direction of any possible association.

5.3 Multinomial Response models

5.3.1 Baseline Category Logits

We primarily follow explanation in Agresti [25, Section 7.1]. Let Y be a categorical response with J categories. Multicategory logit models for nominal response variables describe the log odds for all $\binom{J}{2}$ pairs of categories. Given some choice of $J-1$ categories, the rest are redundant. In our case, we have 4 categories from our clustering. We let $\pi_j(\boldsymbol{x}) = Pr(Y = j|\boldsymbol{x})$ for some choice of values for the explanatory variables $\boldsymbol{x}$, and specify that $\sum_j \pi_j(\boldsymbol{x}) = 1$. For observations under the specification of $\boldsymbol{x}$ we treat the counts in the J categories of Y as a multinomial distribution with probabilities

Figure 5.1: Mosaic plot of A road status against counter classification



$\{\pi_1(\boldsymbol{x}), \ldots, \pi_J(\boldsymbol{x})\}$. Each response category is paired with a baseline category J and in effect we are seeking the 'change' required to transform responses from our baseline category into our specified response category. Our model is therefore

$$\log \frac{\pi_j(\boldsymbol{x})}{\pi_J(\boldsymbol{x})} = \alpha_j + \boldsymbol{\beta}^T \boldsymbol{x}, j = 1, \ldots, J-1. \tag{5.1}$$

The model simultaneously describes the effects of $\boldsymbol{x}$ on the $J-1$ logits. We also note that determining the logits by pairing with the baseline also determines the logits for other pairs of categories, since we have

$$\log \frac{\pi_a(\boldsymbol{x})}{\pi_b(\boldsymbol{x})} = \log \frac{\pi_a(\boldsymbol{x})}{\pi_J(\boldsymbol{x})} - \log \frac{\pi_b(\boldsymbol{x})}{\pi_J(\boldsymbol{x})}. \tag{5.2}$$

If the biking data were not sparse, we might also compute χ^2 and G^2 for model checking.

5.3.2 Generalised Linear models

Baseline category logit models are examples of generalised linear models. Given a univariate response variable y and some predictors $x = (x_1, \ldots, x_p)$, this is an extension of the standard linear model $E[y|x] = x^T\beta$ (with the assumption that $y|x \sim N(x^T\beta, \sigma^2)$). Linear models cannot adequately deal

with the distribution of responses y being non-normal. Generalised linear models are able to model responses from a range of distributions that are members of the exponential family. Following the lecture notes from Bayesian Statistics III/IV [26], a distribution is in the one-parameter exponential family if it is of the form

$$p(x|\theta) = f(x)g(\theta)\exp\{c\phi(\theta)h(x)\} \tag{5.3}$$

for the parameters θ of the distribution, some functions f, g, ϕ, h and a constant c. Then, considering $\mu_i = E(y_i|x_i)$, a generalised linear model is determined by the type of the exponential family which determines the distribution of $y_i|x_i$, the form of the linear predictor $\eta_i = x_i^T\beta$ and the link function g such that $g(\mu_i) = \eta_i = x_i^T\beta$. We can then extend this to a multivariate generalised linear model for distributions in the k-parameter exponential family

$$p(x|\theta) = f(x)g(\theta)\exp\{\sum_{j=1}^{k} c_j\phi_j(\theta)h_j(x)\}. \tag{5.4}$$

For a subject i let $\boldsymbol{y}_i = (y_{1i}, y_{2i}, \ldots)^T$ be a vector response, with $\boldsymbol{\mu}_i = E[\boldsymbol{Y}_i]$. Also let g be a vector of link functions. Then the multivariate generalised linear model has the form

$$\boldsymbol{g}(\boldsymbol{\mu}_i) = X_i\boldsymbol{\beta}. \tag{5.5}$$

Each row j of X for observation i contains the values of the explanatory variables for y_{ih}. For our baseline category model, as defined in Agresti [25, 7.1.5], we have $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,J-1})$. Then $\boldsymbol{\mu}_i = (\pi_1(x_i), \ldots, \pi_{J-1}(x_i))^T$ and

$$g_j(\mu_i) = \log\frac{\mu_{ij}}{1 - (\mu i1 + \ldots + \mu_{i,J-1})} \tag{5.6}$$

The design matrix $\boldsymbol{X}_i$ is

$$\boldsymbol{X}_i = \begin{pmatrix} 1 & \boldsymbol{x}_i^T & & & & \\ & & 1 & \boldsymbol{x}_i^T & & \\ & & & & \cdots & \\ & & & & 1 & \boldsymbol{x}_i^T \end{pmatrix}$$

5.3.3 Fitting baseline-category logit models

Following the description in Agresti [25], we set up the model as follows. Let $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iJ})$ be the multinomial trial for a counter i. We set $y_{ij} = 1$ if the response is in category j, else y_{ij}=0. Note that $\sum y_{ij} = 1$. Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ be the values that the p explanatory variables take for counter i as in section 5.3.1. Let $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jp})^T$ be the model parameters for the jth logit. The model is fitted by maximum likelihood estimation subject to $\pi_j(\boldsymbol{x})$ satisfying the $J-1$ equations that describe it [5.3.1]. We first note π_J is determined by $\pi_1, \ldots, \pi_{J-1}$, since $\pi_J = 1 - (\pi_1 + \ldots + \pi_{J-1})$. Similarly $y_{iJ} = 1 - (y_{i1} + \ldots + y_{i,J-1})$. The likelihood for a subject i is

$$\prod_{j=1}^{J} \pi_j(\boldsymbol{x}_i)^{y_{ij}}$$

therefore the contribution to the log likelihood by subject i is as follows

$$\begin{aligned} \log\left(\prod_{j=1}^{J}\pi_j(\boldsymbol{x}_i)^{y_{ij}}\right) &= \sum_{j=1}^{J-1} y_{ij}\log\pi_j(\boldsymbol{x}_i) + \left(1 - \sum_{j=1}^{J-1} y_{ij}\right)\log\left[1 - \sum_{j=1}^{J-1}\pi_j(\boldsymbol{x}_i)\right] \\ &= \sum_{j=1}^{J-1} y_{ij}\log\frac{\pi_j(\boldsymbol{x}_i)}{1 - \sum_{j=1}^{J-1}\pi_j(\boldsymbol{x}_i)} + \log\left[1 - \sum_{j=1}^{J-1}\pi_j(\boldsymbol{x}_i)\right]. \end{aligned}$$

We note that $\log \frac{\pi_j(\boldsymbol{x}_i)}{1-\sum_{j=1}^{J-1}\pi_j(\boldsymbol{x}_i)}$ is exactly that logit described in equation 5.1 and so logits look to be a natural parameter for the model. We now assume that we have n independent observations. We substitute $\alpha_j + \boldsymbol{\beta}_j^T \boldsymbol{x}_i$ for the logit in the first part of the above equation (as per our definition of the model) and use that $\pi_J(\boldsymbol{x}_i) = 1/(1+\sum_{j=1}^{J-1}\exp(\alpha_j + \boldsymbol{\beta}_j^T\boldsymbol{x}_i))$ to get the following form

$$\begin{aligned}
\log \prod_{i=1}^{n}\left(\prod_{j=1}^{J}\pi_j(\boldsymbol{x}_i)^{y_{ij}}\right) &= \sum_{i=1}^{n}\left[\sum_{j=1}^{J-1} y_{ij}(\alpha_j + \boldsymbol{\beta}_j^T\boldsymbol{x}_i) - \log\left(1+\sum_{j=1}^{J-1}\exp(\alpha_j+\boldsymbol{\beta}_j^T\boldsymbol{x}_i)\right)\right] \\
&= \sum_{j=1}^{J-1}\left[\alpha_j\left(\sum_{i=1}^{n} y_{ij}\right) + \sum_{k=1}^{p}\beta_{jk}\left(\sum_{i=1}^{n} x_{ik}y_{ij}\right)\right] \\
&\quad - \sum_{i=1}^{n}\log\left(1+\sum_{j=1}^{J-1}\exp(\alpha_j+\boldsymbol{\beta}_j^T\boldsymbol{x}_i)\right).
\end{aligned}$$

Thus the sufficient statistic for β_{jk} is $\sum_i x_{ik}y_{ij}$, for $j = 1,\ldots,J-1$ and $k = 1,\ldots,p$, where p is the number of explanatory variables used. Similarly the sufficient statistic for α_j is $\sum_i y_{ij}$.

5.3.4 Response probabilities

Once the estimates α and $\boldsymbol{\beta}$ have been computed for each category, we consider the *response probabilities* for given values of explanatory variables $\boldsymbol{x}$. For our baseline category J we set $\alpha_J = 0$ and $\boldsymbol{\beta}_J = 0$, then our response probabilities for category i are

$$\pi_i(\boldsymbol{x}) = \frac{\exp(\alpha_i + \boldsymbol{x}^T\boldsymbol{\beta}_i)}{1+\sum_{h=1}^{J-1}\exp(\alpha_h+\boldsymbol{x}^T\boldsymbol{\beta}_h)}. \tag{5.7}$$

This is the probability of assigning a counter to a particular category when we know the values of $\boldsymbol{x}$. This is an intuitive way of seeing the effect on categorisation of changes in explanatory variables.

5.3.5 In R

We primarily use the function `multinom()` within the nnet package [27], but one can also use `vglm()` from the VGAM [28] package. Although `multinom` is described as fitting log-linear models, Thompson demonstrates by example that in the baseline category logit model they give the same results. [29, p.118-121]. We can then calculate the response probabilities for given values of explanatory variables from the outputs of either functions using `predict()`.

5.3.6 Model Selection

We can consider the difference in residual deviance between two models. Consider two models M_0 and M_1. Let M_0 have deviance D_0 and degrees of freedom d_0, similarly for M_1. Then we can test the difference in deviance $|D_0 - D_1|$ against the χ^2 distribution with $|d_0 - d_1|$ degrees of freedom. We easily find a p-value for the difference in deviance to identify *better* models under this criterion.

We can also consider the Akaike's information criterion (AIC) as a measure of the goodness of fit of an estimated model. By adding or removing factors we look for the model which offers the lowest AIC value. This model is then the *best*. AIC is defined as follows (within the VGAM package) for a model with k parameters and maximised likelihood L:

$$\text{AIC} = 2k - 2\ln L$$

5.3.7 Single variable models

To start, we fit try fitting single parameter models. We note that within the framework of our model set up we have that $J = 4$. This case is equivalent to a two way contingency table and we have specified a saturated model. As shown by Xie and Powers [30], in this simple case we can estimate α_j and β_{ij} for all categories j. Simultaneous estimation, as performed by solving the maximum likelihood estimation as in R functions, will yield the same results. Simultaneous and separate estimations will usually give different results for unsaturated models. As an example, we fit the single model for the variable `NCN`, which takes the value 1 if the counter is on a route labelled as on the National Cycle Network, and 0 if not. An excerpt of the data frame giving these values for each counter is below.

```
> newClassNCN
    NCN clustering classification
1     1          4         hybrid
2     0          1        leisure
3     1          2        schools
4     1          4         hybrid
5     1          4         hybrid
6     1          1        leisure
7     0          3       commuter
...
```

Expressing the data frame as a contingency table we get the following table

```
> table(newClassNCN$NCN,newClassNCN$classification)

    commuter hybrid leisure schools
  0       10     22       8       3
  1       17     21      24       2
```

and we fit the model of the following form `classification ~ NCN`, using `multinom()`. The results are then summarised and displayed using

```
> summary(multinom(classification ~ NCN, data=newClassNCN,trace=F))
Call:
multinom(formula = classification ~ NCN, data = newClassNCN)

Coefficients:
        (Intercept)        NCN
hybrid    0.7884535 -0.5771422
leisure  -0.2231492  0.5679911
schools  -1.2039883 -0.9360823

Std. Errors:
        (Intercept)       NCN
hybrid    0.3813845 0.5018931
leisure   0.4743414 0.5705173
schools   0.6582832 0.9960738

Residual Deviance: 254.3616
AIC: 266.3616
```

We note that since this is a saturated model, we could easily use the definition of the model, as seen in equation 5.1, to calculate the value of α_j by setting $x = 0$, and then using this to calculate β_j. For

instance,

$$x = 0 \Rightarrow \alpha_{hybrid} = \log \frac{22}{10}$$
$$x = 1 \Rightarrow \beta_{hybrid} = \log \frac{21}{17} - \log \frac{22}{10}$$

Calculating the response probabilities for a given x are therefore exactly the proportions specified in the original contingency tables. This is a feature of saturated models.

5.3.8 Models with zeros in contingency table cells

We now consider a single variable model which has a 'sampling zero' in one cell of the contingency table. There are two types of zero that can occur in a contingency table, as Agresti describes [25, 9.8]. The first is the sampling zero, which can frequently occur if we have a large number of variables compared to the number of observations, leading to sparse contingency tables. The second is a 'structural zero', which occur less frequently in practice, where a cell is empty because a certain combination of explanatory variables are impossible. For instance, a counter location could not simultaneously be (road)-traffic free and an A road. The cells where both of these are true would contain structural zeros. The following 2 dimensional contingency table for `route` (i.e. is the counter on a route that is adjacent to a road?) and `classification` contains a sampling zero. We see no counters that are on leisure routes and next to a road.

```
> table(newClassRoute$route, newClassRoute$classification)

    commuter hybrid leisure schools
  0       17     28      32       4
  1       10     15       0       1
```

Following the same calculations as before, we see that we should find that

$$\beta_{leisure} = \log \frac{32}{17} - \log \frac{0}{10} = -\infty.$$

However, comparing this to the output from `multinom`, we see that this is not shown.

```
> summary(multinom(classification ~ route, data=newClassRoute))
Call:
multinom(formula = classification ~ route, data = newClassRoute)

Coefficients:
        (Intercept)        route
hybrid    0.4989866  -0.09345878
leisure   0.6324810 -10.71041396
schools  -1.4469457  -0.85564786

Std. Errors:
        (Intercept)      route
hybrid    0.3074673  0.5110844
leisure   0.3001218 48.8004645
schools   0.5557196  1.1869667

Residual Deviance: 238.1993
AIC: 250.1993
```

We see that the estimate for $\beta_{leisure}$ is negative but not $-\infty$ and that the standard error for that parameter is very large indeed. The software does ML fitting via an interative process, usually Newton-Raphson, that reports convergence. Due to the slight curve in log-likelihood the standard error

estimates are very large. They are also unstable, so small changes in the data lead to large changes in the parameter estimates and standard errors. Agresti notes that ML parameter estimates can be unaffected by empty cells in contingency tables of higher dimension and indeed demonstrates this in the alligator example in [25, Chapter 7]. Since the iterative method converges without properly identifying the 'correct' value of $\beta_{leisure}$, one must look closely at the summary output to identify problems of this kind, especially when working with higher dimensional models. We also note from the summary output that the nonsense parameter estimates have caused the model AIC and residual deviance values to be much lower than for a model with a complete contingency table. This means that merely running through all possible models looking for significant changes in AIC or deviance is no longer possible, as we must also check to see that zeros in the table are not affecting parameter estimates. This precludes somewhat the use of `step()` and `stepAIC()` for model selection. Variables that have 2-d contingency tables with sampling zeros when tabled against classification still cause problems when used in higher dimensional tables and models. We therefore create a function to run through the contingency tables for each available variable to establish a list of 'safe' variables. The list of 'safe' variables is:

```
NCN
Urban
Aroad
Trafficfreeroute
Canalriverside
Cycletrackonfootpath
Lightingno
Lessthan3miles
```

5.3.9 Two variable models

We now attempt to fit a 2 parameter model. (Because of some random reason) we choose to fit models featuring NCN and Aroad variables, which can again take the values 0 or 1. For model selection, we compare the deviances of various models to the deviance of the saturated model

```
classification ~ NCN*Aroad
```

We note that when the model contains a single variable these are exactly the same as the saturated models previously described. The 3-dimensional contingency table is as follows

```
               Aroad  0     1
               NCN    0  1  0  1
classification
commuter              7  9  3  8
hybrid               14 20  8  1
leisure               8 23  0  1
schools               2  2  1  0
```

We note that the contingency table does contain sampling zeros. However, collapsing the table down to either variable singularly does not lead to sampling zeros, so these will not prove problematic during parameter estimation. We now look at some model selection. We compare the deviances of smaller models to the full, saturated model. We note of course that changes in deviance are approximately χ^2 with degrees of freedom equal to the change in degrees of freedom between the models. We begin by defining each possible model, and then use a non-standard function `mdiff()` to compare the deviances of our model. `mdiff()` is defined in section A.4, in summary, it calculates the difference in deviance between two models fitted using `multinom()` in addition to the change in degrees of freedom; it then performs the usual χ^2 test on the change in deviance. It is easily adaptable to work with models fitted by `vglm()`.

```
> fitS <- multinom(classification ~ NCN*Aroad, data=newClassLoc,trace=F)
```

```
> fit1 <- multinom(classification ~ 1, data=newClassLoc,trace=F)
> fit2 <- multinom(classification ~ NCN, data=newClassLoc,trace=F)
> fit3 <- multinom(classification ~ Aroad, data=newClassLoc,trace=F)
> fit4 <- multinom(classification ~ NCN + Aroad, data=newClassLoc,trace=F)
> mdiff(fitS, fit1)
classification ~ NCN * Aroad
classification ~ 1
Deviance a:232.7, Deviance b:260.6, Difference:-27.9, Df:9, P:0.000977.
> mdiff(fitS, fit2)
classification ~ NCN * Aroad
classification ~ NCN
Deviance a:232.7, Deviance b:254.4, Difference:-21.7, Df:6, P:0.0014.
> mdiff(fitS, fit3)
classification ~ NCN * Aroad
classification ~ Aroad
Deviance a:232.7, Deviance b:246.4, Difference:-13.7, Df:6, P:0.0328.
> mdiff(fitS, fit4)
classification ~ NCN * Aroad
classification ~ NCN + Aroad
Deviance a:232.7, Deviance b:240.8, Difference:-8.1, Df:3, P:0.0436.
```

The model `classification ~ NCN + Aroad` is the 'best' of the models as we note that it has the least significant difference in deviance from the saturated model. However, examining the saturated model further we find that the parameter estimation has broken down. In a similar way to the single variable models, we see that the parameter estimates for the Aroad and NCN:Aroad terms for the leisure equation, as well as the NCN:Aroad for schools, are large in magnitude and the associated standard deviances very large. This is indicative of the software converging to a solution that is not correct.

```
Coefficients:
        (Intercept)        NCN       Aroad  NCN:Aroad
hybrid    0.6936519  0.1051208   0.2871505  -3.165985
leisure   0.1340086  0.8039931 -10.6631765   7.646255
schools  -1.2516582 -0.2515930   0.1525538 -14.418731

Std. Errors:
        (Intercept)       NCN       Aroad  NCN:Aroad
hybrid    0.4629689 0.6127319   0.8201542   1.399773
leisure   0.5176009 0.6500107 111.6395998 111.645328
schools   0.8016087 1.1195015   1.4058209 939.164114

Residual Deviance: 232.7087
AIC: 256.7087
```

We note that this directly affects the deviance estimate, so our comparisons cannot be trusted. It is clear that saturated models in particular struggle most with problematic zeros in the contingency tables.

5.3.10 Two variable model with structural zero

Certain combinations of variables lead to structural zeros in a contingency table. For instance, a model looking at the effects on classification of whether the route being on an A road and whether the route is vehicle traffic free obviously cannot have both states being true at once. Thus the contingency table shown below has a column of sampling zeros in it.

```
               Trafficfreeroute  0      1
               Aroad             0   1  0   1
classification
commuter                         3  11 13   0
hybrid                           7   9 27   0
leisure                          2   1 29   0
schools                          0   1  4   0
```

We can easily deal with this by combining these two variables into a single variable with 3 states. We could then fit the model `classification ~ TrafficAndAroad`.

5.3.11 Model selection leading to higher dimensional models

We now try model selection using model deviance as our criterion. We add a term if it reduces deviance significantly. We also look at each stage at whether removing a term reduces deviance. We begin with a model containing only an intercept.

Model	**Deviance**	**p value**
classification ~ 1	260.6	—
classification ~ NCN	254.4	0.0986
classification ~ Urban	254.0	0.0849
classification ~ Aroad	246.4	0.0026
classification ~ Trafficfreeroute	245.9	0.0021
classification ~ Canalriverside	257.4	0.351
classification ~ Cycletrackonfootpath	257.4	0.357
classification ~ Lightingno	248.1	0.00568
classification ~ Lessthan3miles	250.8	0.0198
+ NCN	240.7	0.159
+ Urban	239.7	0.101
+ Aroad	242.1	0.278
+ Canalriverside	239.3	0.083
+ Cycletrackonfootpath	242.9	0.393
+ Lightingno	239.8	0.103
+ Lessthan3miles	236.1	0.020
...		

Table 5.2: Table of deviances for model selection

The deviances and p-values presented in Table 5.2 are a useful indication of which variables explain 'most' about the classification of counters. Thus it allows Sustrans to identify variables that are most related to classification in general. For instance, adding the variable about whether the cycle track is on a footpath to a model is not helpful in explaining classification. Sustrans needs parsimonious models so that they can be easily used and understood. Models with large numbers of parameters and interactions will not be immediately helpful to understand what explanatory variables lead to changes in counter classification, so where possible we want the simplest model that explains the most variance. Indeed, when fitting models involving variables to explain different classifications, Sustrans would need to make sure that it is not specifying unnecessarily detailed models.

Proceeding with model selection as in Table 5.2 yields this final model:

```
classification ~ Trafficfreeroute + Lessthan3miles + Lightingno
```

The model parameters are summarised below. We perform the standard test to determine if the parameter estimates are significantly different from zero, that is (in this case)

$$\left| \frac{\hat{\beta}_{ij}}{SE[\hat{\beta}_{ij}]} \right| \sim t_{12}.$$

We calculate these and tabulate manually as the software does not do this itself. The commuter category is the baseline, and the region parameters represent dummy variables.

Parameter	**Value**	**Std.Error**	**t-value**	**p-value**
hybrid:Intercept	-0.147	0.390	0.377	0.356
hybrid:Trafficfreeroute	0.294	0.548	0.537	0.300
hybrid:Lessthan3miles	1.534	0.634	2.417	0.016
hybrid:Lightingno	0.072	0.939	0.076	0.470
leisure:Intercept	-1.575	0.641	2.457	0.015
leisure:Trafficfreeroute	1.874	0.752	2.492	0.014
leisure:Lessthan3miles	0.329	0.717	0.459	0.327
leisure:Lightingno	1.469	0.857	1.715	0.056
schools:Intercept	-3.052	1.110	2.751	0.009
schools:Trafficfreeroute	0.988	1.256	0.787	0.223
schools:Lessthan3miles	1.908	1.088	1.753	0.052
schools:Lightingno	0.417	1.414	0.295	0.386

Table 5.3: Table of parameter estimates and associated information

The values of the parameters given in Table 5.3 are of interest in their own right. The parameter estimates for the effect of each variable tell us about the relative change in proportions of counters compared to the baseline. Here we see that, for instance, relatively more leisure routes are traffic free than commuter routes, which tallies with our intuition. Also we see that more schools routes are less than 3 miles long, which is a reasonable conclusion, given that we might pupils who cycle to school to also live reasonably close to their school. These parameter values are useful; if Sustrans is able to fit larger models then they will be able to see the direction of change to particular counter classifications proportions for that variable, without having necessarily to fit hypothetical variable values. We can calculate response probabilities for a hypothetical counter on an unlit, traffic-free route that is less than 3 miles long.

```
  commuter     hybrid    leisure    schools
0.06165034 0.35577006 0.50244144 0.08013816
```

We find a high probability that this counter would be classified as leisure.

5.3.12 Schools

It is easy to see that the schools usage is not adequately explained by any of the explanatory variables so far. School type usage is as a result of a counter being located near to a school, something that is not one of the explanatory variables offered by Sustrans. One could then repeat the analysis performed in this chapter with the rows corresponding to schools counters removed from the data matrix used for model fitting. Another interesting analysis could be performed by using a GIS system to map the counters and look for nearby schools. Since Sustrans already has the location information for each counter, we could then create a further explanatory variable asking whether there is a school within, say, a mile of the counter location. Indeed, this would allow Sustrans to identify schools with high numbers of cyclists riding to and from school, as well as schools with low ridership. Schools with low numbers of cyclists could then be targeted for an intervention and any effect observed in the usage profiles and counter classification. Sustrans could also repeat this with commuter routes by identifying counters near to industrial or commercial estates.

5.3.13 Extending the model

As we have noted, the model fitting breaks down when we have insufficient data for non-sparse contingency tables. With more data one can envisage being able to fit higher dimensional models whilst also improving the accuracy of the models we have already established. We might also look for more explanatory variables. For instance, nearby geographical features might influence the categorisation of a route – leisure users might like to ride hilly terrain for fun, whereas we might expect that commuters would look for flatter alternatives to ride everyday. Also, it would be of interest to know if there was a school or large business within a certain distance of the counter, or whether a counter was on the only cycle link between residential and business areas, for instance.

5.4 Year profiles

It is of course be possible to repeat this sort of analysis using the year profile clustering given in Section 4.9. We first summarise the interesting results of the standardised residuals (as in Section 5.2.3) for our classification and explanatory variables in the form of mosaic plots displayed in Figure 5.2. We see that leisure routes are more frequently rural, but less likely to be on a shared use path. We also see that leisure routes are less likely to be next to roads, and more likely to be unlit. Similarly we see that commuter routes are lit more often than we might expect, and are predominantly in urban areas. This tallies with what we might intuitively expect.

We now apply the multinomial logit model techniques. In this case we would have a *binary* logit model, as described by Xie and Powers [30]. Naturally we have that, using the description of the logit model in Equation 5.1, $\pi_1(\boldsymbol{x}) = 1 - \pi_2(\boldsymbol{x})$ for explanatory variables $\boldsymbol{x}$, leading to the model

$$\log \frac{\pi_1(\boldsymbol{x})}{\pi_2(\boldsymbol{x})} = \log \frac{\pi_1(\boldsymbol{x})}{1 - \pi_1(\boldsymbol{x})} = \alpha_1 + \boldsymbol{\beta}^T \boldsymbol{x}$$

for some categories 1 and 2. We then repeat similar R code. We fit the following model using four explanatory variables

```
classification ~ Urban + Lessthan3miles + NCN + Routeadjacenttoroad
```

leading to the parameter estimates and standard errors in Table 5.4. The commuter group is the baseline category, and the model reflects the change in explanatory variable values required for a change in classification to leisure. As we might expect, we see that commuter routes occur more

Parameter	**Value**	**Std.Error**	**t-value**	**p-value**
Intercept	1.320	0.681	1.938	0.055
Urban	-2.021	0.626	3.228	0.011
Lessthan3miles	-0.681	0.527	1.292	0.126
NCN	0.737	0.499	1.477	0.099
Routeadjacenttoroad	-1.963	0.706	2.780	0.019

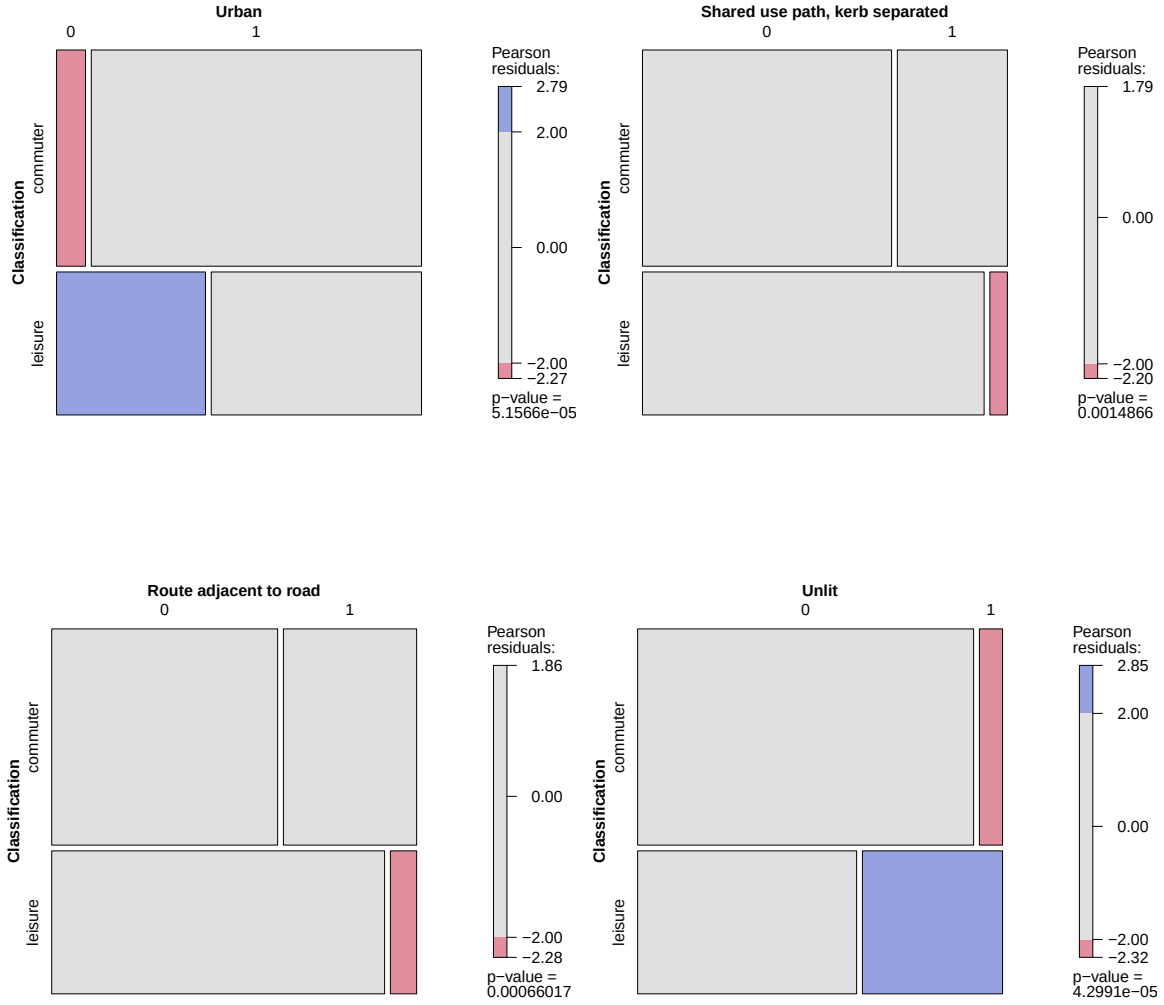
Table 5.4: Table of parameter estimates and associated information

frequently in urban areas (as we see in Figure 5.2) and that they are more often found next to the road. Leisure routes are more often NCN labelled, which makes sense, as we might expect leisure users to follow a particular cycle route through perhaps a scenic area, rather than travelling point-to-point as commuters would.

5.5 Categorising a new counter

We now attempt to categorise a new counter. Sustrans performs manual counts at locations where it thinks it might be beneficial to create a new route, or promote cycling to certain users in an area. Sustrans reports that when they advertise a new route then they usually see usage rise by about 30%

Figure 5.2: Mosaic plots



in absolute terms. Although we have not considered usage numbers in this project, it may still be beneficial to Sustrans to try and get an idea of the sort of usage patterns that they might expect to see in a particular location given easily obtainable information about that location. We are provided with 12-hour manual counts for 4 days at a proposed counter or route location. The days cover four different scenarios - a weekday and a Saturday during school term time; and a weekday and Saturday outside of term time, and the data comprises hourly cycle totals from 7am until 7pm. We perform a discriminant analysis to link the clustering results to the original proportion data.

5.5.1 Discriminant analysis

We first consider the classification of counters using a clustering algorithm as fixed. In particular, for continuity, we use the k-means partition that gave us our main result in Chapter 4, as displayed in Figure 4.2. So we have allocated each counter to a particular group (commuter, leisure, etc.) on the basis of the clustering algorithm. We now consider a *linear discriminant function* that allows us to summarise observations on a one dimensional scale and then allows us to *discriminate* between observations in each group. In effect this allows us to compute the function using the average day proportions from the existing counters as training data, and then use this function to classify new counters. We follow the explanation given in Morrison [31]. We assume that the observations in each of the k groups follows a multivariate normal distribution, with mean vectors $m_1, \ldots, m_k$ with common

covariance matrix $\mathbf{\Sigma}$. Since these are unknown, we replace them with their estimates as follows

$$\hat{m}_j = \bar{x}_j$$
$$\boldsymbol{S} = \frac{1}{N-k}\sum_{j=1}^{k} \boldsymbol{A}_j$$

where as usual $\bar{x}_j$ is the sample mean for group j, and $\boldsymbol{A}_j$ is the sums of squares and products matrix for group j. Denoting by x the new observation, then we calculate the linear discriminant score as (Morrison, [31, p.240])

$$W_{i,j} = x^T \boldsymbol{S}^{-1}(\bar{x}_i - \bar{x}_j) - \frac{1}{2}(\bar{x}_i - \bar{x}_j)^T \boldsymbol{S}^{-1}(\bar{x}_i - \bar{x}_j) \tag{5.8}$$

classify x by assigning it to group i if

$$W_{i,j} > 0 \ \ \forall\ j \neq i.$$

Tabachnick and Fidell [32, 11.4.2] simplify this somewhat by solving to find a single discriminant function for each group, so that instead of having to compare functions between pairs of groups, we look for the group that has the maximised discriminant function. Denoting by $\boldsymbol{C}_j$ the classification coefficients, then we have that

$$\boldsymbol{C}_j = \boldsymbol{S}^{-1} m_j$$

and that the constant in the discriminant function is found to be

$$c_{j0} = -\frac{1}{2}\boldsymbol{C}_j m_j.$$

This leads to the discriminant function of the form

$$C_j = c_{j0} + c_{j1}x_1 + \ldots + c_{jp}x_p.$$

When we substitute in the values of x we look for the discriminant function with the highest score, and classify the new data into that group.
In R we use the function `lda()` from the MASS library [33]. This function fits a linear discriminant function, and can allow us to specify prior probabilities for the groups. If we do not specify a prior then it automatically uses the observed proportions of the group allocations as the prior. We can then predict group classification for a new counter using `predict()`. We must first create a new data frame containing the data for the same hours that we have data in our new counter, i.e. 7am to 6pm. We then rescale so that we have the proportion in each hour of the 12 hour period, since this is what we have for the new counter. We also rename the columns for convenience.

```
> biking.reduced <- biking.avgdays[,8:19]

> for (i in 1:107) {
  biking.reduced[i,] <- biking.reduced[i,] / sum(biking.reduced[i,])
  }

> biking.reduced <- cbind(biking.reduced, as.vector(clustering))
> names(biking.reduced) <- c("7am","8am","9am","10am","11am",
     "12pm","1pm","2pm","3pm","4pm","5pm","6pm","clustering")
```

We then fit the model using the following command. Note that we allow the function to compute the prior group probabilities from the data.

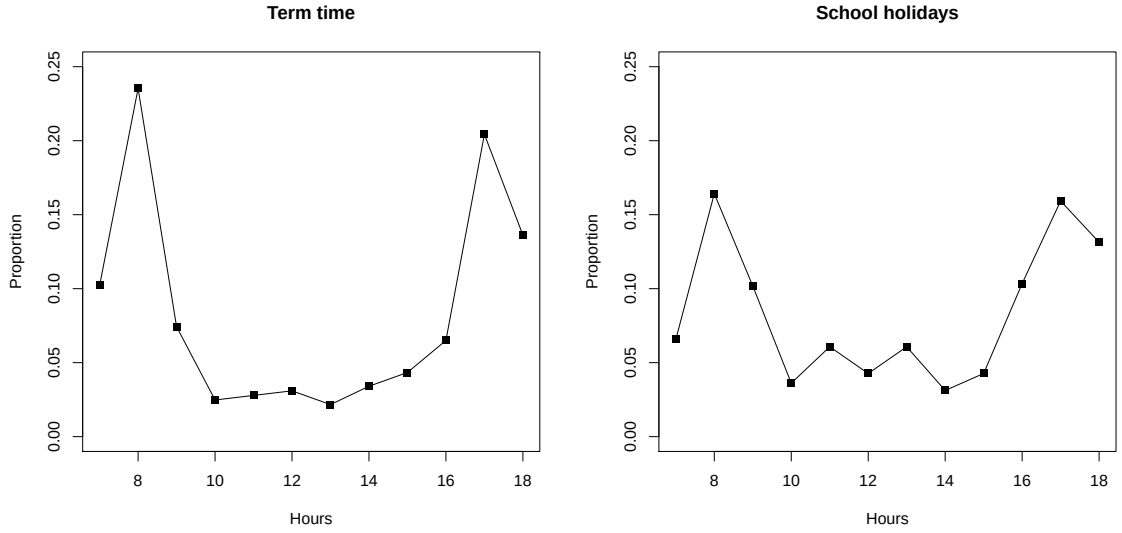
```
> z <- lda(clustering ~ .,data=biking.reduced)
```

Using this model, we are then able to predict the classification of a new counter, using the observed proportion of total counts across the 12 hour period. We can look at the predicted classification during school term time and outside of term time to look for any difference.

```
> predict(z, newsch)$class
[1] 3
Levels: 1 2 3 4
> predict(z, newsch)$posterior
             1            2         3            4
1 4.437973e-13 2.020593e-12 0.9999797 2.026688e-05

> predict(z, newhol)$class
[1] 3
Levels: 1 2 3 4
> predict(z, newhol)$posterior
             1            2         3            4
1 3.097997e-07 3.396413e-14 0.9999693 3.040437e-05
```

Figure 5.3: New counter usage profiles



In both cases we see very high probabilities of classifying the new counter as group 3, which is the commuter usage group. Plotting the usage profiles for those 12 hours shows this to be an acceptable prediction.

It is also of interest to look at the misclassification probabilities. We compare the actual classification from the cluster analysis with that given by our discriminant function for each of the 107 counters providing the training data. We summarise the results in a table.

	clustering			
discriminant	L	S	C	H
leisure	31	0	0	0
schools	0	5	0	0
commuter	0	0	25	0
hybrid	1	0	2	43

We see therefore that 104 of the counters were correctly classified by the discriminant function, suggesting an overall misclassification rate of 2.5%. We also see that within the groups, the predicted classifications matched the original classification for all of the counters in the hybrid group, as happened for the schools group. The leisure group had a misclassification rate of 3.1%, and the commuter had a rate of 7.4%. It is possible to estimate the misclassification probabilities. Here we use the discriminant function as defined by Morrison for pairs of groups, and note that the misclassification

probability of assigning observation x to group j (when it should be in group i), as shown in [31, 6.4], is therefore

$$\begin{aligned} P_i &= P[W_{i,j} \leq 0] \\ &= P\left[\frac{x^T a - \mu_i^T a}{(a^T \Sigma a)^{1/2}} \leq \frac{1/2(\bar{x}_i + \bar{x}_j)^T a - \mu_i^T a}{(a^T \Sigma a)^{1/2}}\right] \\ &= \Phi\left[\frac{1/2(\bar{x}_i + \bar{x}_j)^T a - \mu_i^T a}{(a^T \Sigma a)^{1/2}}\right] \end{aligned}$$

for groups i distributed as $N(\mu_i, \Sigma)$ and j distributed as $N(\mu_j, \Sigma)$, with $a = S^{-1}(\bar{x}_i - \bar{x}_j)$. Similarly the misclassification probability into group i is

$$P_j = \Phi\left[\frac{\mu_i^T a - 1/2(\bar{x}_i + \bar{x}_j)^T a}{(a^T \Sigma a)^{1/2}}\right]$$

These would allow us to estimate the misclassification rates for our discriminant functions before we evaluate them. Morrison also describes other methods of estimating the misclassification probabilities that we will not consider here.

5.5.2 Using the Multinomial Logit models

We would, in principle, be able to use the multinomial logit models presented in Section 5.3 in order to predict usage classification based upon knowledge of the explanatory variables at the location. However, the information about the nature of the location of the manual count is unavailable. In any case, the difficulties we have discussed as regards to the sparse contingency tables and small sample sizes only make it possible to fit relatively simple models, such as the three variable model given in Section 5.3.11. The discriminant functions shown in this section are therefore likely to be of most use to Sustrans by categorising a new counter location based on manual counts if they are available. If not, use multinomial logit models based on the explanatory variables at that location to categorise the new counter.

Chapter 6

Handling sparse contingency tables

Analysis of the contingency tables that we considered in the previous chapter is well known if the sample size is large relative to the number of cells. One can calculate maximum likelihood estimates for various methods of modelling the underlying independence, which have known asymptotic distributions; and goodness-of-fits based on the χ^2 distribution are available for model fitting. However, these sorts of models and analyses break down if the table is sparse. As we have seen in the previous chapter when using more explanatory variables our tables become progressively more and more sparse as a result of the relatively small number of counters used. Simonoff [34] provides an example of a sparse table for a cross-classification of MBA students about the importance of statistics and economics in business education. There are 55 responses to 49 possible cells. He demonstrates that the usual methods of analysing break down in the case of this table, particularly in terms of a possibly outlying cell in the table. He also notes that changing the count in this cell from a 1 to a 0 leads to a large change in perceived lack of fit. This sensitivity to change is somewhat problematic in terms of handling zeros in tables. Simonoff demonstrates that other methods suggested for dealing with sparseness as well as alternative model selection also break down due to the sensitivity to change in frequencies in particular cells.

6.1 Adding counts

One way of dealing with problematic zeros in cells is to add a constant to the count in every cell in the table and Goodman proposes $\frac{1}{2}$ [35]. Through their analysis of loglinear models which require non-zero counts, Grizzle, Starmer and Koch propose adding $1/K$ where K is the total number of cells. Both methods have significant use in practice, however, one can see the need for general purpose smoothing of tables instead of ad-hoc modifications. Indeed, the analysis by Simonoff demonstrates the problems that can arise when adding counts to tables. Software can also be problematic, as the R functions `multinom()` and `vglm()` appear to (at least in our case) deal with data-frames containing whole numbers of rows, rather than the contingency tables themselves, leading to problems when adding non-integer counts to cells in a table. One cannot create a data-frame containing 0.7 of a row, for instance. This problem will still be apparent after the proposals in the next sections.

6.2 Ordered category models

Dealing with sparse contingency tables is fairly well-documented when the categories are naturally ordered in some way. For instance, if we asked someone to categorise a film as 'Bad', 'Average', 'Good', 'Superb', then a natural ordering from these categories would be to go from bad through to superb, and we might believe that there is an underlying continuity across categories. In this case, we would hope to be able to take advantage of this natural ordering to in effect 'borrow' information from other cells in the table in order to compensate somewhat for the sparseness. We call this sort of method 'smoothing' of categorical data, and Simonoff [34] summarises various attempts.

6.3 Non-ordered categories

It is obvious that the categories proposed in our analysis of cycling data are not naturally ordered. Nevertheless, we are still able to use some results that were proposed prior to more complex smoothing applied to ordered categories. We begin by considering one-dimensional tables. Fienberg and Holland (1973) [36] use a Dirichlet prior to establish an estimate for a cell probability $\hat{p}_i$ of the form

$$\hat{p}_i = \frac{n_i + \alpha}{N + K\alpha}$$

for N the total number of observations, n_i the number of observations in a cell and K the total number of cells. We then choose α appropriately, with $\alpha \in [0, 1]$. Bishop, Fienberg and Holland (1975) [37, Chapter 12, 12.7-19,12.7-20] then generalise this result to two-way tables with

$$\hat{p}_{ij} = \frac{n_{ij} + Bn_{i.}n_{.j}/N^2}{N + B}$$

with

$$B = \frac{N^2 - \sum_{i,j} n_{ij}^2}{\sum_{i,j} (n_{ij} - n_{i.}n_{j.}/N)^2}$$

Applying this to an example from the earlier chapter, our initial 2-way table with a zero is as follows

```
> table(newClassRoute$route, newClassRoute$classification)

    commuter hybrid leisure schools
  0       17     28      32       4
  1       10     15       0       1
```

We have that $N = 107$, and for instance $n_{1.} = 81$, $n_{.1} = 27$ and so on. We calculate B as

$$B = \frac{107^2 - 17^2 - \ldots - 1^2}{[17 - 81 \times 10/107]^2 + \ldots + [1 - 5 \times 26/107]} = 48.414$$

leading to the $\hat{p}_{ij}$ as tabulated below.

```
    commuter hybrid leisure schools
  0   0.1689 0.2749  0.2764  0.0368
  1   0.0834 0.1269  0.0226  0.0099
```

which gives the following frequencies after multiplication by 107.

```
    commuter hybrid leisure schools
  0    18.07  29.41   29.57    3.93
  1     8.92  13.57    2.41    1.06
```

We see that the zero from the original table has gone. We note that the row and column totals are still the same as in the original table (aside from rounding errors). An interesting note is that now the cell which was a zero now has a larger count than the schools count in the same row. The leisure column has a lot of mass in the top row so in effect 'pulls in' frequency into the bottom row when smoothing, leading to the larger count.

Chapter 7

Summary

We have seen how data collected from the automatic cycle counters may be used to observe usage patterns at different sites across the United Kingdom. In Chapter 2 we presented an initial assessment of the data files. Here we demonstrated code to load the data into R, clean it and add useful columns to the main data frame. In Chapter 3 we presented the main theory of cluster analysis. These techniques have been popular of late in the fields of bioinformatics and medicine, but have also found use in a broad spectrum of other areas, including data-mining by internet search, social and shopping sites. We have presented a novel application of these methods to our cycling data and the results of this approach are analysed in Chapter 4. We were able to determine four clusters in the weekday data, with each cluster corresponding to a usage profile. Each counter location is allocated to a cluster and we saw the usage profiles averaged across each cluster displayed in Figure 4.2. We were able to intuitively define the type of usage in each cluster as one of schools, commuter, leisure and hybrid; and this was the main result from Chapter 4. We also considered the differences between rural and urban counters, but found they presented the same classification; looked at the differences between weekend and weekday usage profiles and saw that weekend usage was primarily leisure; and examined applying different algorithms such as k-medoids and hierarchical clustering. We also looked at cluster validation techniques and found that these suggested a smaller number of clusters, however, this entailed what appeared to be a significant loss of detail in the cluster profiles.

Considering the main clustering result in Figure 4.2 as fixed, Chapter 5 linked this clustering to the explanatory variables provided by Sustrans. First, we used Fisher's exact test to identify explanatory variables responsible for changes in counter classification, and looked at the standardised residuals from a χ^2 test to get an idea of the direction of any change. The main part of Chapter 5 concerned the theory and application of multinomial logit models. We considered first single variable models and then increased the complexity of the models by adding more variables. We saw some of the parameter approximation errors that software routines generate when we work with small cell counts in contingency tables. We then performed model selection by looking at the deviance reduction in adding and removing variables to our model and analysed a reasonable model in Section 5.3.11. We looked at categorising a new counter at the end of the chapter, in particular using a linear discriminant analysis. In Chapter 6 we briefly looked at ways of handling contingency table cells with zero observations, primarily by smoothing using Dirichlet prior. We saw that handling cell counts for ordered categories has been discussed more frequently as one can take advantage of the underlying order of categories – such a methodology would not work for the non-ordinal categorical data presented here.

The work undertaken in this project should be of use to Sustrans. We have used data driven methods to confirm the usage patterns that Sustrans believed to exist. The cluster analysis and methodology would benefit from further data to reinforce the schools cluster in particular. In addition, the linear discriminant functions presented at the end of Chapter 5 will be very useful for taking manual counts and predicting the 24-hour usage patterns that we might see. Not only this, Sustrans can run the cluster and discriminant analyses that consider longer time periods in order to look for patterns and then try to predict usage based on a small sample of data. Fisher's exact test, and the standardised

residuals from the χ^2 test will be useful for identifying potential influence on cluster group by explanatory variables and the direction of any change. This would be useful for identifying particular characteristics that lead, say, to commuter usage along a route. Sustrans could then observe the effect of replicating that characteristic at other counters on the network. The multinomial logit models will be effective in linking these clusters of similar usage patterns to explanatory variables – we have seen, for instance, that leisure routes are typically unlit. The models presented here would benefit from additional data to enable more interesting model parameters to be fitted, as well as refining those parameter estimates we have used. Sustrans need not necessarily use the clustering as the basis for grouping the counters before using the multinomial logit models – as long as they can group the counters in some appropriate way and have some variables that might explain that grouping, then the type of model we have shown will allow them to explore the effect of those variables on the grouping.

7.1 Further work

There are numerous unexplored avenues that could be followed in future work. Using the clustering tools presented, one could compare any number of different time periods and time steps from the data. For instance, it may be of interest to compare average summers to look for differences in usage, or average monthly usage. It would be interesting to use as the 'averages' matrix two day profiles from each counter, one from a weekday and one from a weekend day, and see if the weekend responses are always clustered with the leisure counters. An obvious direction for further work is to add more counters and see if the number or shape of clusters change. Certainly, with more schools counters we would expect to be able to perform more tests, such as χ^2 tests upon the 2-d tables to check for independence. We would also expect to be able to fit larger, more complex multinomial logit models without the issues of sparseness.

Of some interest to Sustrans, and that which has not really been considered in this project, is the usage on routes in absolute terms. Sustrans' success is defined in terms of the reduction of unsustainable transport, so an increase in journeys taken by cycle helps them to meet this goal. We would therefore use the data to try and predict usage on existing routes and we could link this to the work completed in this project. Certainly, things like the weather conditions and the local attitude towards cycling have a large effect on usage numbers and it would be interesting to combine this information with the models suggested in this project. We could link to each counter type a baseline usage total that we expect for each month or year and see if or how variables such as weather, advertising and interventions change the count from this baseline. We could also use the methods seen in Chapter 5 to see if there were any characteristics common to counters where we see high or low usage so that changes to new and existing routes could be targeted effectively to maximise usage increases.

Another interesting area would be to look at the effect of schools upon usage numbers. Looking at the usage shapes for commuter and school counters in Figure 4.2, we see that they follow a similar shape, but with the schools counters displaying much higher peak usage. Looking then at Figure 5.3, it is interesting to note the differences in usage patterns between term time and school holidays. Thinking in terms of the actual counts rather than proportions, we would be able to use the data from all the counters to get a baseline 'school usage' count by comparing term time and holiday usage. We could then consider this pattern as accounted for at a counter and consider the type of usage we see without school journeys included, perhaps as to whether the underlying usage pattern is commuter or leisure. Sustrans would also be able to use this to see if cycling interventions in schools lead to increases in the baseline school usage at counters in the area.

Acknowledgements

I am particularly grateful to my supervisor, David Wooff, for his advice and guidance throughout this project. In addition, special thanks are given to Dr. Lisa Muller at Sustrans for motivating the project, supplying the data and for her expert advice about Sustrans' work.

Bibliography

[1] Sustrans. http://www.sustrans.org.uk/what-we-do/national-cycle-network/route-numbering-system, 24th April 2010.

[2] Sustrans. http://www.sustrans.org.uk/resources/research-and-monitoring, 24th April 2010.

[3] A. J. Richardson. *The Survey of the Veloland Schweiz National Cycling Routes: a comparison of results from 1999 through 2002.* 2003.

[4] Andy Cope, Sally Cairns, Ken Fox, Debbie A Lawlor, Mary Lockie, Les Lumsdon, Chris Riddoch, and Paul Rosen. The UK national cycle network: an assessment of the benefits of a sustainable transport infrastructure. *World Transport Policy and Practice*, 9:6–17, 2003.

[5] A. J. Richardson. A survey method for cycle networks - a Swiss example. *Forum Papers, 23es Australasian Transport Research Forum, Perth*, 1:443–457, 1999.

[6] Les Lumsdon, Paul Downward, and Andy Cope. Monitoring of cycle tourism on long distance trails: the north sea cycle route. *Journal of Transport Geography*, 12:13–22, 2004.

[7] Velo Quebec. *Bicycling in Quebec in 2005.* 2005.

[8] National Statistics. *Transport Statistics Bulletin: National Travel Survey: 2008.* Transport Statistics, 2008.

[9] David James and Kurt Hornik. *chron: Chronological Objects which Can Handle Dates and Times*, 2009. R package version 2.3-33. S original by David James, R port by Kurt Hornik.

[10] Brian S. Everitt. *Cluster Analysis, Third Edition.* Edward Arnold, London–Melbourne–Auckland, 1993. ISBN 0-340-58479-3.

[11] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[12] A.D. Gordon. *Classification, 2nd Edition.* Chapman and Hall/CRC, 1993. ISBN 1-58488-013-9.

[13] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 45:325–342, 1985.

[14] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27, 1974.

[15] L. A. Goodman and W. H. Kruskal. Measures of association for cross-classifications. *Journal of the American Statistical Association*, 49:732–64, 1954.

[16] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973. ISBN 0-471-22361-1.

[17] E. M. L. Beale. Euclidean cluster analysis. *Bulletin of the International Statistical Institute*, 43:92–4, 1969.

[18] Leonard Kaufman and Peter J. Rousseeuw. *Finding groups in data.* Wiley, 1990. ISBN 0-471-87876-6.

[19] Martin Maechler, Peter Rousseeuw, Anja Struyf, and Mia Hubert. Cluster analysis basics and extensions. Rousseeuw et al provided the S original which has been ported to R by Kurt Hornik and has since been enhanced by Martin Maechler: speed improvements, silhouette() functionality, bug fixes, etc. See the 'Changelog' file (in the package source), 2005.

[20] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[21] J.H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–44, 1963.

[22] R. Mojena. Hierarchical grouping methods and stopping rules: an evaluation. *The Computer Journal*, 20:359–363, 1977.

[23] Mark S. Aldenderfer and Roger K. Blashfield. *Cluster Analysis.* SAGE, 1984.

[24] Jari Oksanen, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, R. G. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, and Helene Wagner. *vegan: Community Ecology Package*, 2010. R package version 1.17-0.

[25] Alan Agresti. *Categorical Data Analysis, 2nd Edition.* John Wiley and Sons, 2002. ISBN 0-471-36093-7.

[26] Peter Craig. *Bayesian Statistics III/IV, Lecture notes.* 2008.

[27] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S.* Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[28] Thomas W. Yee. *VGAM: Vector Generalized Linear and Additive Models*, 2010. R package version 0.7-10.

[29] Laura A. Thompson. *S-PLUS (and R) Manual to Accompany Agresti's Categorical Data Analysis (2002), 2nd Ed.* 2006.

[30] Daniel A. Powers and Yu Xie. *Statistical Methods for Categorical Data Analysis.* Academic Press, 2000. ISBN 0-12-563736-5.

[31] Donald F. Morrison. *Multivariate Statistical Methods, Second Ed.* McGraw-Hill, Inc, 1976.

[32] Barbara G. Tabachnick and Linda S. Fidell. *Using Multivariate Statistics, Fourth Ed.* Allyn and Bacon, 2001. ISBN 0-321-05677-9.

[33] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S.* Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[34] Jeffrey S. Simonoff. Smoothing categorical data. *Journal of Statistical Planning and Inference*, 47:41–69, 1995.

[35] Leo A. Goodman. The multivariate analysis of qualitative data: Interactions among multiple classifications. *Journal of the American Statistical Association*, 65(329):226–256, 1970.

[36] Stephen E. Fienberg and Paul W. Holland. Simultaneous estimation of multinomial cell probabilities. *Journal of the American Statistical Association*, 68(343):683–691, 1973.

[37] Y.M.M. Bishop, S.E. Fienberg, and P.W. Holland. *Discrete Multivariate Analysis: Theory and Practice.* MIT Press, Cambridge, MA, 1975.

Appendix A

Code

A.1 Creating an average day profile

This function outputs an average day profile for a given counter location, and in addition we are able to specify what years and what day of the week the averages are taken over.

```
avgday <- function(location, years="all", day="weekdays") {

## give a way of specifying to use all available valid data
if (years == "all") {
possibledates <- unique(subset(biking, (counter == location))$date)
biking.years <- biking } else {
## grab data only from specified years
biking.years <- data.frame()
for (x in years) {
   biking.years <- rbind(biking.years, subset(biking,
     as.numeric((substr(date,7,9))) == x))
}
}

## only grab our specified location
biking.loc <- subset(biking.years, (counter == location))

## select days etc, including all days
if (as.character(day[1]) == as.character("weekdays")) {
day <- c("Monday","Tuesday","Wednesday","Thursday","Friday") }
if (as.character(day[1]) == as.character("weekend")) {
day <- c("Saturday","Sunday") }
if (as.character(day[1]) == as.character("all")) {
biking.yearsdays <- biking.loc } else {
biking.yearsdays <- data.frame()
for (x in day) {
    biking.yearsdays <- rbind(biking.yearsdays, subset(biking.loc,
       (as.character(days) == as.character(x))))
}
}

## total how many journeys were made in the up and down directions
totalup <- sum(biking.new$upcount)
totaldown <- sum(biking.new$downcount)
totalboth <- totalup+totaldown
```

```
## vector of hour marks
hours <- c((0:23)*100)

## add up the number of up and down counts per hour
bothtotals <- c()
uptotals <- c()
downtotals <- c()
for (x in hours)
{
bothtotals <- c(bothtotals, sum(subset(biking.new, (hour == x))$upcount
              + subset(biking.new, (hour==x))$downcount))
uptotals <- c(uptotals, sum(subset(biking.new, (hour == x))$upcount))
downtotals <- c(downtotals, sum(subset(biking.new, (hour == x))$downcount))
}

weightedboth <- bothtotals / totalboth
weightedup <- uptotals / totalup
weighteddown <- downtotals / totaldown


## create a dataframe containing hour marks and totals
biking.avgday <- data.frame(cbind(hours,bothtotals,weightedboth,uptotals,
                 weightedup,downtotals,weighteddown))

return(biking.avgday)

}
```

A.2 Creating an average year profile

This function outputs the year profile for a given counter location.

```
yearprofile <- function(location) {

locationdata <- subset(biking, (counter == location))

data.vec <- rep(0,12)
days <- c(31,28,31,30,31,30,31,31,30,31,30,31)

for (i in 1:12) {
month.data <- subset(locationdata, (as.numeric(substr(date, 4,5)) == i))

daysmean <- (sum(month.data$upcount) + sum(month.data$downcount))
            / length(unique(month.data$date))

## we need to rescale data to take into account the
## number of years that we have data for that month for...
numyears <- length(unique(sapply(month.data$date,substr,7,8)))
##numyears could be zero, so set numyears to be 1
data.vec[i] <- daysmean * days[i]
}

data.vec <- data.vec / sum(data.vec)
## hand us back the data.vec
```

```
return(data.vec)

}
```

A.3 Drawing average cluster plots

This is a general purpose function that takes the clustering vector output and the 'averages' matrix that the clustering was generated from. This means it can be used for day or year profiles, or any types of averages, clustering or data matrix that the reader cares to use. In addition, one can specify whether to output the drawn graphs to a file.

```
drawclusterplots <- function(kcluster,bikingmatrix,filesave=F) {
##bikingmatrix is the biking.avgdays matrix

## define some useful stuff
numbers <- c()
sizebikingm <- dim(bikingmatrix)[2]
averages <- data.frame(row.names = c(1:sizebikingm))

## find out how many is in each cluster, and zero out the summing matrix
for (x in sort(unique(kcluster))) {
  numbers <- c(numbers,sum(subset(kcluster, (kcluster == x)))/x)
  averages <- cbind(averages, rep(0,sizebikingm))
}

## label stuff
names(averages) <- sort(unique(kcluster))
## sum up the totals from biking.avgdays for each cluster
for (i in 1:length(kcluster)) {
  averages[,as.numeric(kcluster[i])] <- as.numeric(averages[,kcluster[i]]
  + bikingmatrix[i,])
}

## divide by the number in each cluster
for (i in 1:length(numbers)) {
  averages[,i] <- averages[,i] / numbers[i]
}

## for some reason stuff gets renamed
names(averages) <- sort(unique(kcluster))

## find out the average day for all locations
n <- dim(bikingmatrix)[1]
allavg <- c()
for (i in 1:sizebikingm) {
allavg <- c(allavg, sum(bikingmatrix[,i])/n)
}

## draw the plot and (possibly) save it
matplot(averages, type="l", lty=1,xlab="Hours",ylab="Proportion")
if (filesave==T) {psf(as.character(runif(1)*100000000))}

## draw the plot - the average day and (possibly) save it
matplot(averages - allavg, type="l", lty=1, xlab="Hours",ylab="Proportion")
```

```
if (filesave==T) {psf(as.character(runif(1)*100000000))}

## give us back the averages
return(averages)
}
```

A.4 A function for multinomial logit model selection

This function, kindly provided by David Wooff, calculates the difference in deviance between two models fitted using `multinom()` and outputs the significance of this change.

```
mdiff<-function(f1,f2){
    #### Show differences in deviance between two models, and test
    #### assuming Chi sq distribution
    print(f1$call$formula)
    print(f2$call$formula)
    devd<-f1$deviance-f2$deviance
    dfd<-abs(f1$edf-f2$edf)
    cat(paste(sep="","Deviance a:",round(f1$deviance,1),", Deviance b:",
      round(f2$deviance,1),", Difference:",round(devd,1),", Df:",dfd,", P:",
      signif(pchisq(abs(devd),dfd,lower=F),3),".\n"))
}
```