Assignment 2 Report

Eamonn Mc Gonigle
L00093050
Emerging Technologies

15th December 2014

Contents

List of Figures

1 Introduction

The following report details investigations into the application of Big Data analytics and visualisation techniques on public transport systems. Specifically the investigations involved utilising the statistical programming language R to process and analyse publicly available data sets relating to the Dublin Bus system. The graphical visualisation of the Dublin Bus system with R was alas also investigated.

The following section will define the hypotheses to be tested. Section 3 will give an overview of the technologies that were used in testing the hypotheses. Section 4 will describe the methods used in testing the hypotheses. The results of the tests will then be evaluated in section 5. Finally conclusions will be drawn and ways in which the work carried out here could be expanded upon in the future will be discussed (section6).

2 Hypotheses

The investigation tested following hypotheses:

Hypothesis 1 Big data can be used to identify weaknesses in public transportation systems.

Hypothesis 2 Geospatial data can be graphically represented with the with the R Programming Language and Software Environment for Statistical Computing.

Before detailing how the hypotheses where tested and evaluating the results, an overview of the data and technologies involved is warranted.

3 Data & Technology Overview

This section will provide both an overview of the data used in the investigation, and of the R statistical environment used to process the data.

3.1 R Programming Language and Software Environment for Statistical Computing

The R website describes R a “a language and environment for statistical computing and graphics”(Team, 2014). It has gained popularity in recent years. This is due to the ease with which data analysts and researchers with little programming experience can carry out statistical analysis with it(Vance, 2009). It is also gaining popularity in the computing industry as data analytics becomes more of an integral part of almost every software company(Ramel, 2014).

The language is relatively simple, albeit unconventional. It has been suggested that a better way to think of R is as an “interactive environment for doing statistics” that has a scripting language built in(Cook, 2014).

3.2 Data Overview

The investigation utilised two datasets of relating to the Dublin Bus network.The first dataset contains a sample AVL(Automatic Vehicle Location) data. The second dataset contains GTFS(General Transit Feed Specification) data for the Dublin Bus Network. Both datasets can be obtained freely from the Dublinked website(Dublinked, 2013a)(Dublinked, 2013b). The two datasets will be explained further in the following subsections.

3.2.1 AVL dataset

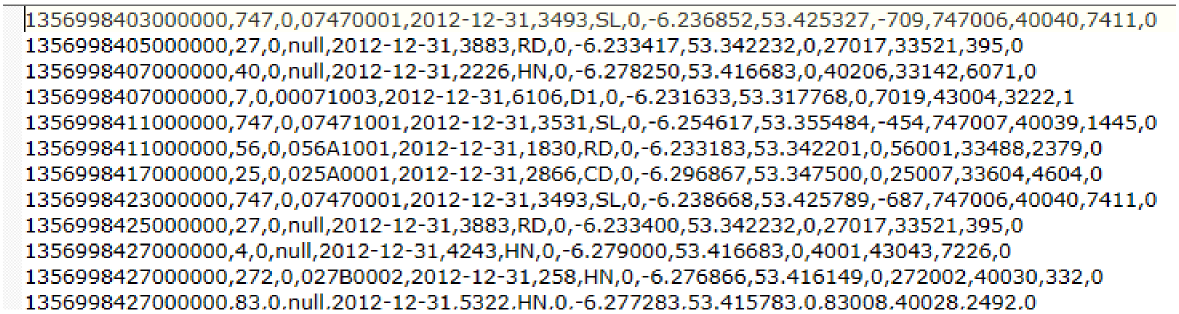
Automatic vehicle locating (AVL) gives bus operators the ability to track buses in real time by means of on-board transmitters that communicate each buses GPS coordinates(Gartner, n.da). However, the data can also be archived and analysed historically as was done in this investigation.

The AVL dataset obtained for this investigation contains a record of every GPS probe transmitted by every bus in the Dublin Bus network over the period of one month. The dataset is divided into 31 CSV files - a file for every day. For the purposes of this investigation, two days worth of records was deemed sufficient.

As evident in figure 3.1, the raw CSV files have no column headers and are very difficult for a human to make sense of. Even with the header names and brief descriptions provided by Dublinked web page(figure 3.2), the data appears essentially meaningless in its current structure.

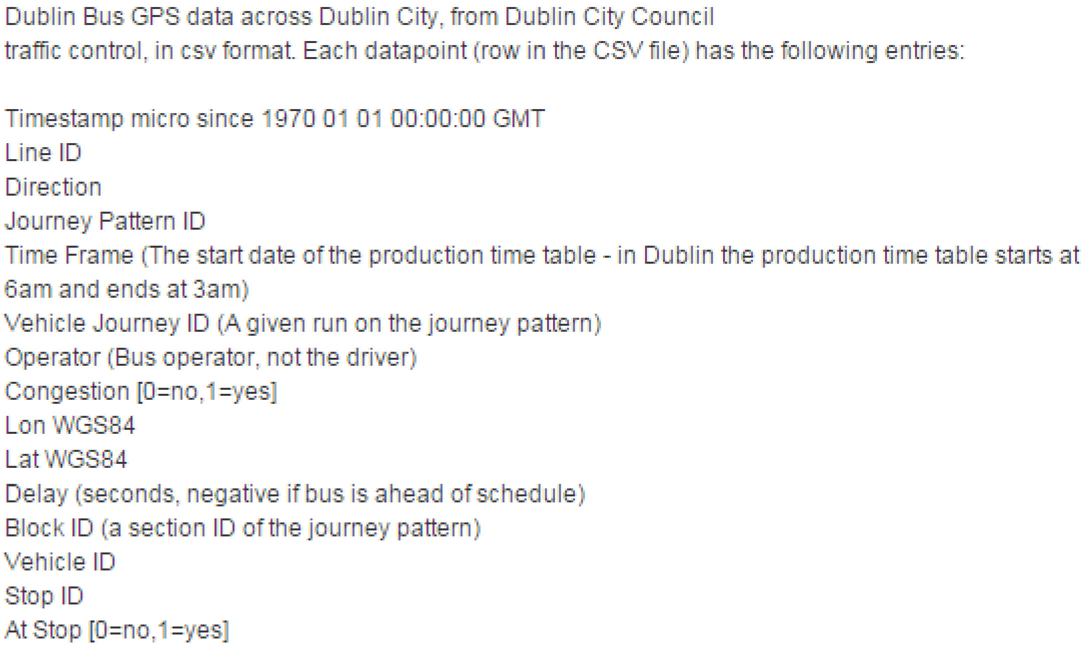
Closer inspection reveals that the data is ordered by the time stamp the AVL probe was recorded. The fact that the dataset contains 2,464,051 unique probes from hundreds of different buses, ensures the data must be cleaned up and sub setted before useful analysis can be done.

This makes the data a characteristic of many real world big data analytic scenarios. Gartner(n.db) defines big data as “...high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making." This AVL dataset is certainly of high volume and of high variety(given the many different data types across the columns). Although being analysed historically it could also be described as high velocity given the number of records recorded over a short period of time.



```
1356998403000000,747,0,07470001,2012-12-31,3493,SL,0,-6.236852,53.425327,-709,747006,40040,7411,0
1356998405000000,27,0,null,2012-12-31,3883,RD,0,-6.233417,53.342232,0,27017,33521,395,0
1356998407000000,40,0,null,2012-12-31,2226,HN,0,-6.278250,53.416683,0,40206,33142,6071,0
1356998407000000,7,0,00071003,2012-12-31,6106,D1,0,-6.231633,53.317768,0,7019,43004,3222,1
1356998411000000,747,0,07471001,2012-12-31,3531,SL,0,-6.254617,53.355484,-454,747007,40039,1445,0
1356998411000000,56,0,056A1001,2012-12-31,1830,RD,0,-6.233183,53.342201,0,56001,33488,2379,0
1356998417000000,25,0,025A0001,2012-12-31,2866,CD,0,-6.296867,53.347500,0,25007,33604,4604,0
1356998423000000,747,0,07470001,2012-12-31,3493,SL,0,-6.238668,53.425789,-687,747006,40040,7411,0
1356998425000000,27,0,null,2012-12-31,3883,RD,0,-6.233400,53.342232,0,27017,33521,395,0
1356998427000000,4,0,null,2012-12-31,4243,HN,0,-6.279000,53.416683,0,4001,43043,7226,0
1356998427000000,272,0,027B0002,2012-12-31,258,HN,0,-6.276866,53.416149,0,272002,40030,332,0
1356998427000000.83.0.null.2012-12-31.5322.HN.0.-6.277283.53.415783.0.83008.40028.2492.0
```

Figure 3.1: Raw CSV of AVL dataset



Dublin Bus GPS data across Dublin City, from Dublin City Council
traffic control, in csv format. Each datapoint (row in the CSV file) has the following entries:

Timestamp micro since 1970 01 01 00:00:00 GMT
Line ID
Direction
Journey Pattern ID
Time Frame (The start date of the production time table - in Dublin the production time table starts at 6am and ends at 3am)
Vehicle Journey ID (A given run on the journey pattern)
Operator (Bus operator, not the driver)
Congestion [0=no,1=yes]
Lon WGS84
Lat WGS84
Delay (seconds, negative if bus is ahead of schedule)
Block ID (a section ID of the journey pattern)
Vehicle ID
Stop ID
At Stop [0=no,1=yes]

Figure 3.2: AVL dataset columns as described by Dublinked(Dublinked, 2013a)

3.2.2 GTFS Dataset

The General Transit Feed Specification (GTFS) was created Google with the aim of providing public transportation operators the means of publishing their schedules and geographical information in a standardised format so that developers can utilise the data in creating useful applications (Google, 2012).

The GTFS data for the Dublin bus system was obtained for this investigation. The GTFS data comes in a number of CSV files relating to both bus timetable information and the geographical information for stops. Only the shapes.txt file is used in this investigation. The shapes.txt data set contains the GPS coordinates outlining all the routes travelled by Dublin buses

The GTFS data set was primarily utilised to investigate the graphical capabilities of R in relation to geospatial data.

4 Testing the Hypotheses

The following section will describe the artefact created to test the hypotheses stated in section 2. The artefact takes the form of an R script that reads in, processes and visualises particular data sets. Both Hypothesis are tested by means of this same artefact as they are inherently tied together and rely on R. Hypothesis 1 is tested by reading in and processing the data, the data is then visualised in such a way that tests hypothesis 2. With this in mind, subsection 4.1 will walk through the implementation of the testing of hypothesis 1 while subsection 4.2 will elaborate on the methodology behind the testing of hypotheses 2.

4.1 Hypothesis 1

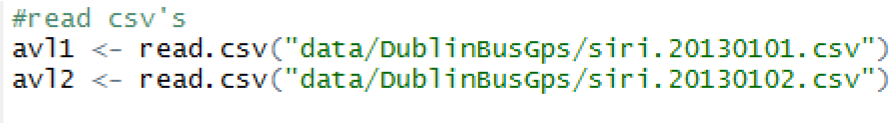
> "Big data can be used to identify weaknesses in public transportation systems."

Hypothesis one was tested using R and the AVL data set in order to find the average time a single bus spends stopped at a given bus stop. How long a bus spends stopped at a bus stop loading and unloading bus stop or dwell time is a well-documented concern amongst public transport architects and is a major factor in the smooth and timely operation of bus systems (Dueker et al., 2004). Deriving the average time a bus spends stopped at each of its stops could identify which stops are busiest, or initiate further investigation into what might be causing the problem. By finding how long a bus spends stopped at a particular stop using data characteristic of big data (i.e. the AVL dataset) and the R statistical language it can be deduce that indeed, big data can be used to identify weaknesses (i.e. long dwell time) in public transport systems (i.e. Dublin bus system).

The execution of this test mostly consisted of cleaning and subletting the AVL dataset in order to isolate the required data. This was done using R. The method was as follows.

4.1.1 Method

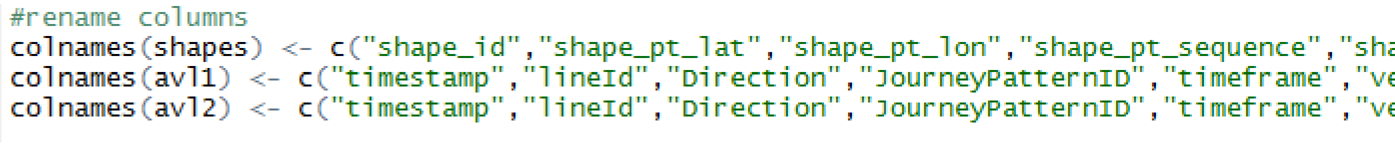
The AVL data set used initially consisted of two separate CSV files. The files were read into two separate data frame objects in R using r read.csv method (figure 4.1).



```
#read csv's
avl1 <- read.csv("data/DublinBusGps/siri.20130101.csv")
avl2 <- read.csv("data/DublinBusGps/siri.20130102.csv")
```

Figure 4.1: Read in AVL CSV files

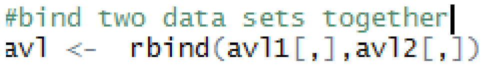
The column headers (as provided by Dublinked) where then added to the data frames(figure 4.2)..



```
#rename columns
colnames(shapes) <- c("shape_id","shape_pt_lat","shape_pt_lon","shape_pt_sequence","sha
colnames(avl1) <- c("timestamp","lineId","Direction","JourneyPatternID","timeframe","ve
colnames(avl2) <- c("timestamp","lineId","Direction","JourneyPatternID","timeframe","ve
```

Figure 4.2: Rename column headers

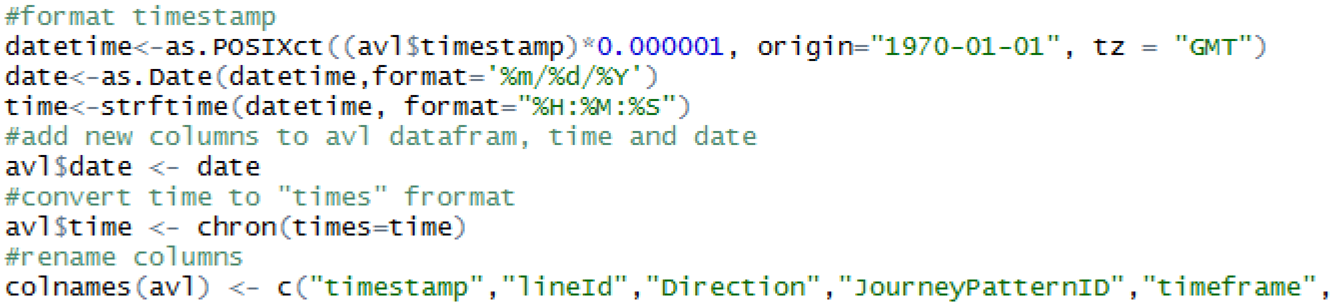
The two data frame objects where then binded together into one data frame object4.3.



```
#bind two data sets together
avl <-  rbind(avl1[,],avl2[,])
```

Figure 4.3: Bind AVL data frames

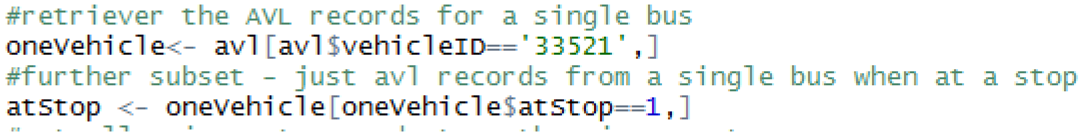
Next the date and time was appended to the data set for every record in human readable format. This was done by converting the UNIX time stamp column into a date objects and storing these new date time objects in a vector object.. This vector was then converted into two more vector objects holding just the time and the date respectively. The time and date vectors where then added to the greater AVL data frame, resulting in every record having a readable time and date value. The column names were then updated to reflect this (figure 4.4).



```
#format timestamp
datetime<-as.POSIXct((avl$timestamp)*0.000001, origin="1970-01-01", tz = "GMT")
date<-as.Date(datetime,format='%m/%d/%Y')
time<-strftime(datetime, format="%H:%M:%S")
#add new columns to avl datafram, time and date
avl$date <- date
#convert time to "times" format
avl$time <- chron(times=time)
#rename columns
colnames(avl) <- c("timestamp","lineId","Direction","JourneyPatternID","timeframe",
```

Figure 4.4: Format Time

Next a subset data frame of the AVL data frame containing only records relating to one bus was created. From this subset a further subset was made containing only those records transmitted when the bus was at a stop.(figure 4.5).

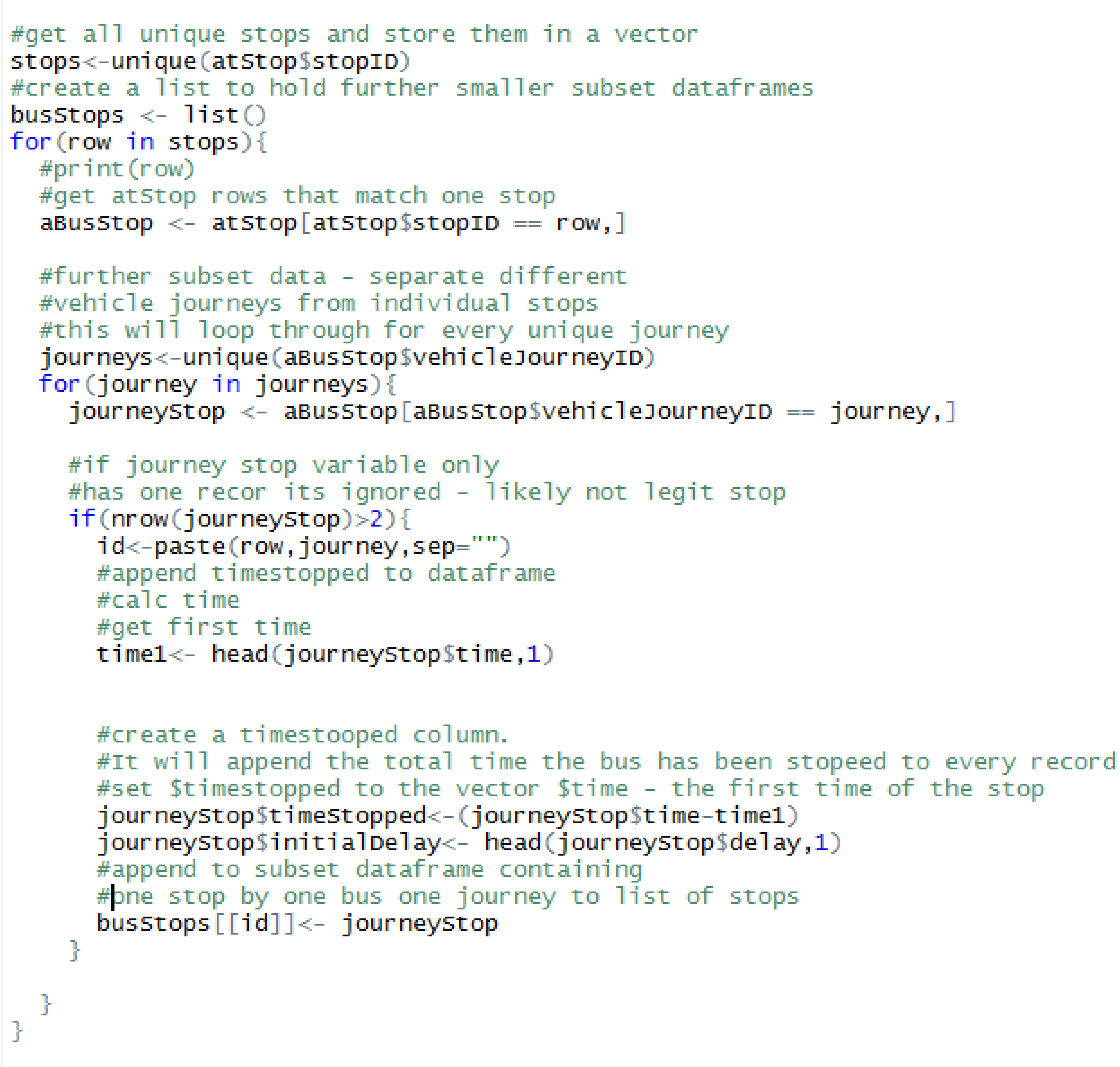


```
#retriever the AVL records for a single bus
oneVehicle<- avl[avl$vehicleID=='33521',]
#further subset - just avl records from a single bus when at a stop
atStop <- oneVehicle[oneVehicle$atStop==1,]
```

Figure 4.5: Subset of records from only one particular vehicle at a stop .

Next the more difficult task of isolating individual stops by a single bus on a single journey (one can stop twice at the same stop throughout the day). Each record meeting this criteria is then to be stored in a data frame which is then stored in a list. In other words list will then be a list of stops by a bus.

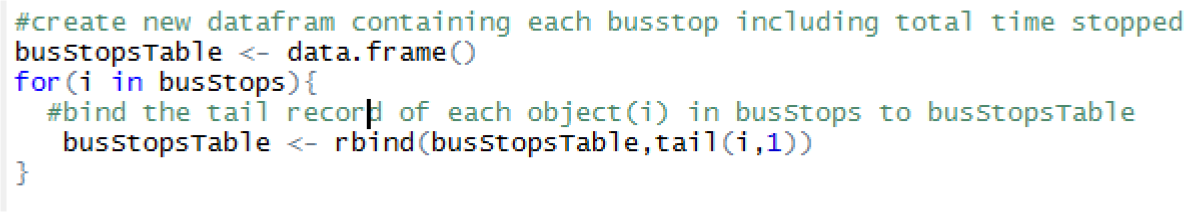
Figure 4.6 displays the code that carried this out. As well as isolating each individual stop the code also ignores stops that have less than one record. These are probably not legitimate stops as each record represents a 20 second timespan. It also append a new time stopped column to each record in the new bus stops subset.



```
#get all unique stops and store them in a vector
stops<-unique(atStop$stopID)
#create a list to hold further smaller subset dataframes
busStops <- list()
for(row in stops){
  #print(row)
  #get atStop rows that match one stop
  aBusStop <- atStop[atStop$stopID == row,]

  #further subset data - separate different
  #vehicle journeys from individual stops
  #this will loop through for every unique journey
  journeys<-unique(aBusStop$vehicleJourneyID)
  for(journey in journeys){
    journeyStop <- aBusStop[aBusStop$vehicleJourneyID == journey,]

    #if journey stop variable only
    #has one recor its ignored - likely not legit stop
    if(nrow(journeyStop)>2){
      id<-paste(row,journey,sep="")
      #append timestopped to dataframe
      #calc time
      #get first time
      time1<- head(journeyStop$time,1)


      #create a timestooped column.
      #It will append the total time the bus has been stopeed to every record
      #set $timestopped to the vector $time - the first time of the stop
      journeyStop$timeStopped<-(journeyStop$time-time1)
      journeyStop$initialDelay<- head(journeyStop$delay,1)
      #append to subset dataframe containing
      #one stop by one bus one journey to list of stops
      busStops[[id]]<- journeyStop
    }

  }
}
```

Figure 4.6: Isolate individual stops by a bus

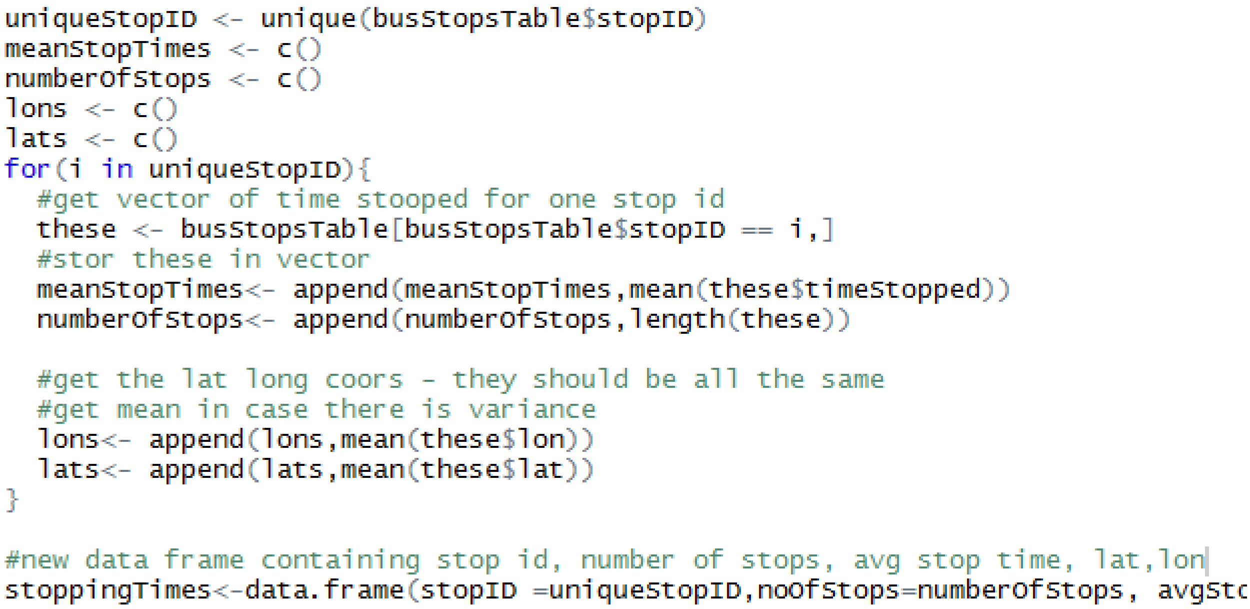
Now that each unique stop has been isolated and its length calculated, some more data sub setting and manipulation must be carried out to calculate the average stopping time for every stop at a particular bus stop. To do this first a new data frame was created and populated with the tail (or last) record of every data frame in the list of stops (figure 4.7).



```r
#create new datafram containing each busstop including total time stopped
busStopsTable <- data.frame()
for(i in busStops){
  #bind the tail record of each object(i) in busStops to busStopsTable
   busStopsTable <- rbind(busStopsTable,tail(i,1))
}
```

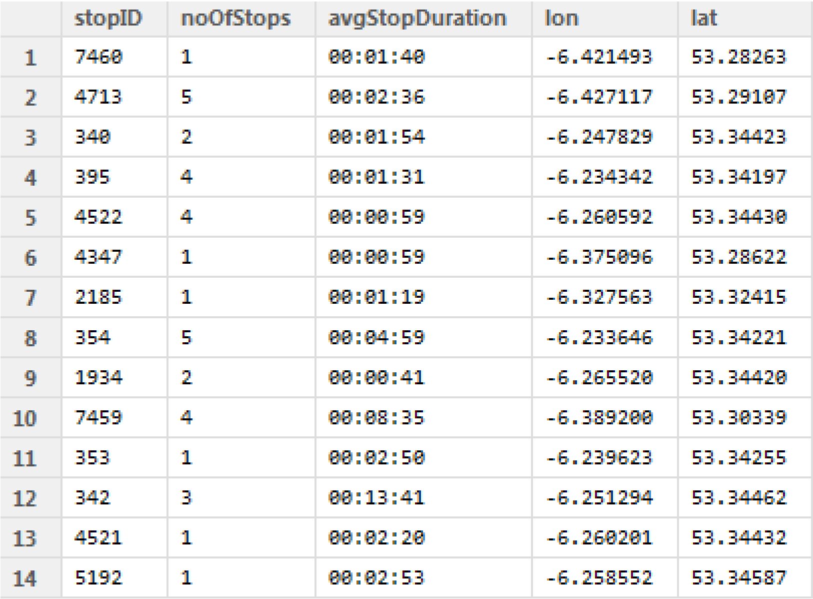
Figure 4.7: New data frame with one record per stop

From this busStopsTable data frame a new table could be create containing only the data we originally set out to achieve. The code to do this (figure) iterates through each unique stop in the busStopsTable and calculates the average time of all stops at that particular stop. This average is then stored in a new data frame along with the number of stops the average is calculated from, the stop id and the latitude and longitude GPS coordinates. The final table can be seen in figure 4.9.



```r
uniqueStopID <- unique(busStopsTable$stopID)
meanStopTimes <- c()
numberOfStops <- c()
lons <- c()
lats <- c()
for(i in uniqueStopID){
  #get vector of time stooped for one stop id
  these <- busStopsTable[busStopsTable$stopID == i,]
  #stor these in vector
  meanStopTimes<- append(meanStopTimes,mean(these$timeStopped))
  numberOfStops<- append(numberOfStops,length(these))

  #get the lat long coors - they should be all the same
  #get mean in case there is variance
  lons<- append(lons,mean(these$lon))
  lats<- append(lats,mean(these$lat))
}

#new data frame containing stop id, number of stops, avg stop time, lat,lon
stoppingTimes<-data.frame(stopID =uniqueStopID,noOfStops=numberOfStops, avgStc
```

Figure 4.8: Calculate average stop time for each stop



	stopID	noOfStops	avgStopDuration	lon	lat
1	7460	1	00:01:40	-6.421493	53.28263
2	4713	5	00:02:36	-6.427117	53.29107
3	340	2	00:01:54	-6.247829	53.34423
4	395	4	00:01:31	-6.234342	53.34197
5	4522	4	00:00:59	-6.260592	53.34430
6	4347	1	00:00:59	-6.375096	53.28622
7	2185	1	00:01:19	-6.327563	53.32415
8	354	5	00:04:59	-6.233646	53.34221
9	1934	2	00:00:41	-6.265520	53.34420
10	7459	4	00:08:35	-6.389200	53.30339
11	353	1	00:02:50	-6.239623	53.34255
12	342	3	00:13:41	-6.251294	53.34462
13	4521	1	00:02:20	-6.260201	53.34432
14	5192	1	00:02:53	-6.258552	53.34587

Figure 4.9: Average time bus id 33521 stops at each of its stops over two day period

4.2 Hypothesis 2

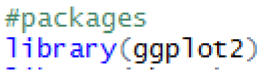
> "Geospatial data can be graphically represented with the with the R Programming Language and Software Environment for Statistical Computing."

Hypothesis 2 was tested by graphically representing the entire Dublin bus route network, as described by the longitude and latitude coordinates in the GTFS shapes.txt dataset. The bus route from the single bus as found in hypothesis 1 was then also visualised, superimposed over the GTFS map. The stops that this bus stopped at was added to the graph, coloured according to the average time they bus spent stopped at each stop. All of this visualisation was performed using the popular R graphing package GGPlot2(Wickham, 2009).

The method to carry out this test was as follows.

4.2.1 Method

To create the graph, the package GGPlot2 was imported into the workspace(figure 4.10).

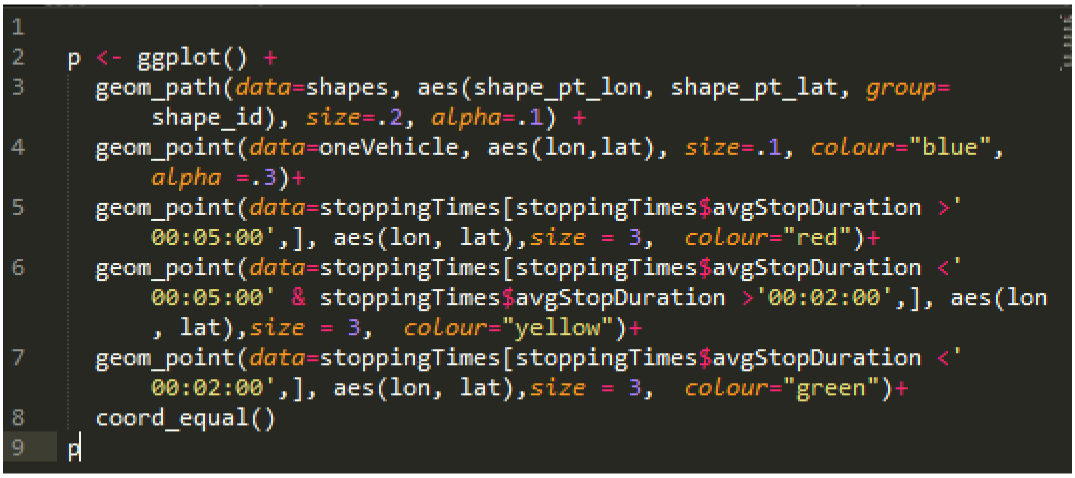


```
#packages
library(ggplot2)
```

Figure 4.10: Import the GGPlot2 Package

Now the plot can be created and displayed. The code in figure 4.11 does this. Credit must be attributed to Joshua Kunst(2014), from whose tutorial exploring visualising the GTFS data of the Santiago public transportation system with GGPlot2, inspired this graph.

First the the shapes.txt GPS data is drawn (line 3) in black. Then the GPS coordinated for the one bus being analysed are superimposed over this(line 4) in blue. Final the stops from the at are then superimposed over this. Stops that average a stopping time of under 2 minutes are portrayed in green, yellow denotes stops of between 2 an 5 minutes, while stops over 5 minutes are coloured red.



```
1
2   p <- ggplot() +
3     geom_path(data=shapes, aes(shape_pt_lon, shape_pt_lat, group=
          shape_id), size=.2, alpha=.1) +
4     geom_point(data=oneVehicle, aes(lon,lat), size=.1, colour="blue",
          alpha =.3)+
5     geom_point(data=stoppingTimes[stoppingTimes$avgStopDuration >'
          00:05:00',], aes(lon, lat),size = 3,  colour="red")+
6     geom_point(data=stoppingTimes[stoppingTimes$avgStopDuration <'
          00:05:00' & stoppingTimes$avgStopDuration >'00:02:00',], aes(lon
          , lat),size = 3,  colour="yellow")+
7     geom_point(data=stoppingTimes[stoppingTimes$avgStopDuration <'
          00:02:00',], aes(lon, lat),size = 3,  colour="green")+
8     coord_equal()
9   p
```

Figure 4.11: Define GGplot2 graph.

The final graph can be seen in figure 4.12.

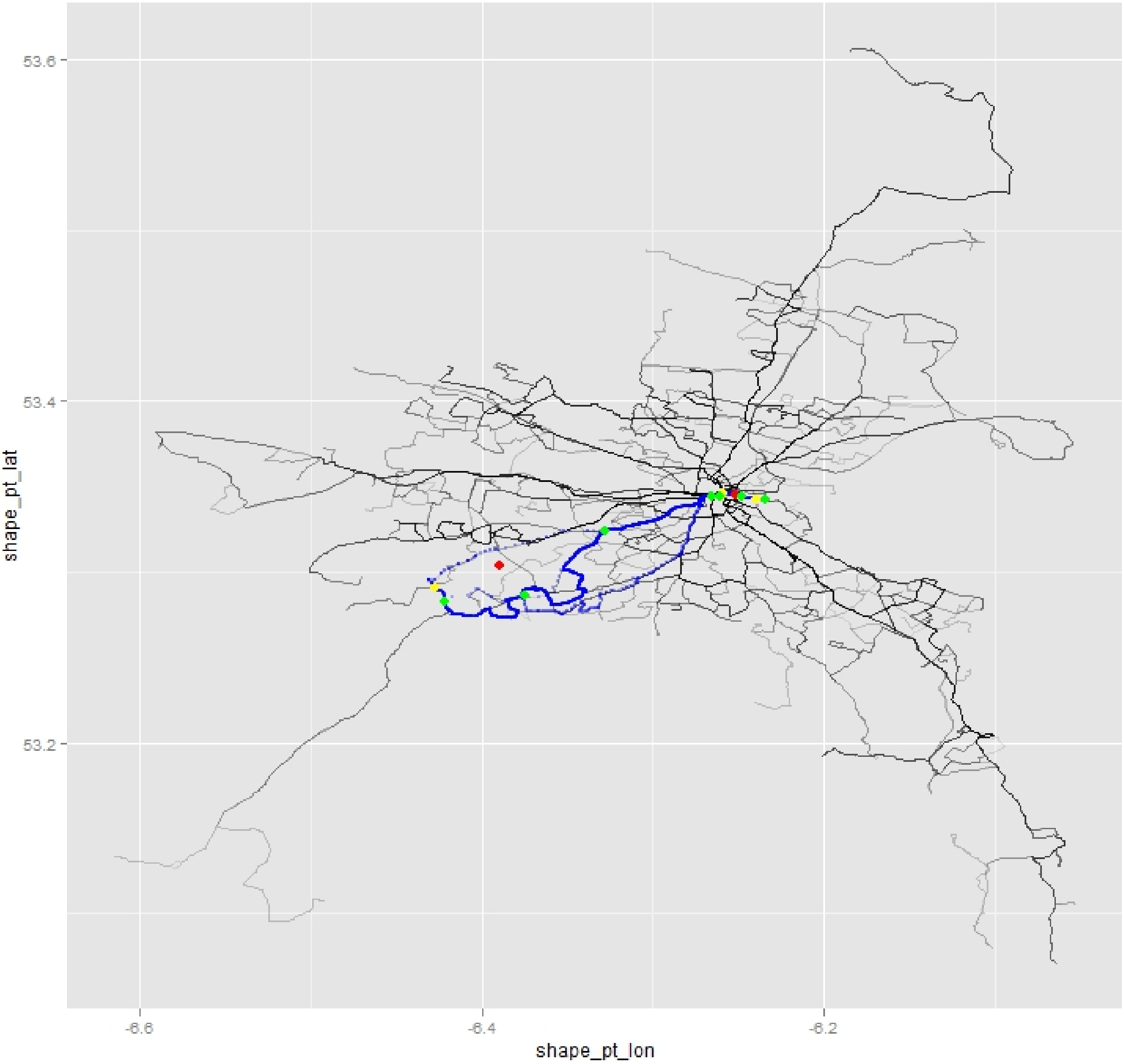


Figure 4.12: Graph of Dublin Bus Network created using R and GGplot2.

5 Results/Evaluation

This section will examine whether the work carried out in the previous section did indeed prove of disprove the initially stated hypotheses. Limitations will also be addressed. Both hypotheses will be discussed in turn.

5.1 Hypotheses 1

> "Big data can be used to identify weaknesses in public transportation systems."

The work described in the previous section, confirms that big data can indeed be uses to identify weaknesses in public transport systems.

Using only a relatively unstructured AVL data set, the average time a single bus spends at each stop on its route was calculated by means of extensive data cleaning and processing. The findings where then visualised, highlighting particular bus stops that have consistently long stop times.

It must be conceded that the usefulness and accuracy the data produced in the artefact (and presented in figure 4.9) may be questionable. However these limitations are a product of the authors restricted access to comprehensive data and unfamiliarity with the internal workings of the Dublin bus (the column descriptions from figure 3.2 where the only source of context available).

The actual results are less important that the methods used to achieve them. Methods which the author believes could be modified and scaled to identify real weaknesses in public transportation systems. The artefact built for this investigation analysed only 2 days worth of AVL data for one particular vehicle. This could hardly be characterised as big data(see (Gartner, n.db) definition). However, scale the data set up to a few months worth of AVL records, processed across a cluster of processors, and much more accurate picture of the average dwell time at a stop could be calculated, truly identifying weaknesses in public transport systems with big data.

The artefact succeeds in demonstrating this potential in spite of the limitations described.

5.2 Hypotheses 2

> "Geospatial data can be graphically represented with the with the R Programming Language and Software Environment for Statistical Computing."

The artefact successfully demonstrates that R is capable of graphically representing geospatial data.

With aid of the GGplot2 plotting package R was able to visualise the entire map of the Dublin bus network, using the GPS coordinates from the shapes.txt (part of the GTFS data set). The data extrapolated while testing hypotheses 1 could then be easily superimposed over this, as demonstrated in figure 4.12.

All of this was done with relative ease in few lines of code (see figure 4.11). Customisation options were plentiful, with an extensive documentation found on the official website of GGplot2(2009), demonstrating the powerful graphing potential of R when working with and data, including geospatial data.

6 Conclusions and Further Work

In carrying out this investigation the potential benefits big data can lend to public transport systems are clear to see. The investigation also proved the R language and environment to be an able tool for both big data analytics and geospatial visualisation, both of relevance to the public transport industry.

The task of calculating the average time a given bus spends at each of its stops from a data set containing only AVL records highlighted the power of R as a data mining and analytics tool. The initial data set was unstructured and difficult to interoperate. R powerful sub setting functions and vector based data types facilitated the isolation of relevant and useful data. In big data, where data is coming from numerous different sources in multiple formats the ability to prepare the data before analysis is particularly important (de Jonge and van der Loo, 2013). As the artefact demonstrates, R is capable of this task.

As well as verifying the utility of the R programming language, the task also provided insight into the greater potential for big data analytics in public transport. In the artefact, the average stop times at different stops for a single bus was calculated. However, this is only one potential application of AVL and other data sources to optimise public transport systems. In the future it would be interesting to investigate whether things like the average speed of a bus, or the average time it takes a bus to complete a certain route. The system could be scaled to analyse all routes and all buses, flagging the routes that are consistently slow.

The investigation also showed that GTFS and AVL data can be visualised with relative ease with R. A worthwhile future project might be to try to map the AVL records of from the individual buses to the routes defined by the GTFS data. This would require the implementation of map matching algorithms due to the 40m error range of GPS (Gerstle, 2012). By linking the two data a much more comprehensive dataset could be analysed and more practical insights into the Dublin bus system (or in fact any bus system that provides both GTFS and AVL datasets freely).

References

Cook, J., 2014. R language for programmers. [accessed 14/12/2014].
URL http://www.johndcook.com/blog/r_language_for_programmers/

de Jonge, E., van der Loo, M., 2013. An introduction to data cleaning with r. Tech. rep.

Dublinked, 2013a. Dublin bus gps sample data from dublin city council (insight project). [accessed 14/12/2014].
URL http://dublinked.com/datastore/datasets/dataset-304.php

Dublinked, 2013b. Dublin bus gtfs data. [accessed 14/12/2014].
URL http://dublinked.com/datastore/datasets/dataset-254.php

Dueker, K. J., Kimpel, T. J., Strathman, J. G., Callas, S., 2004. Determinants of bus dwell time. Journal of Public Transportation 7 (1), 21–40.

Gartner, n.da. Automated vehicle locating (avl). [accessed 14/12/2014].
URL http://www.gartner.com/it-glossary/automated-vehicle-locating-avl/

Gartner, n.db. Big data. [accessed 14/12/2014].
URL http://www.gartner.com/it-glossary/big-data/

Gerstle, D. G., 2012. Understanding bus travel time variation using avl data. Ph.D. thesis, Massachusetts Institute of Technology.

Google, 2012. What is gtfs? [accessed 14/12/2014].
URL https://developers.google.com/transit/gtfs/

Kunst, J., 2014. Plotting gtfs data with r. [accessed 14/12/2014].
URL https://rpubs.com/jbkunst/Plotting-Gtfs-Data-With-R

Ramel, D., 2014. R programming language rides crest of big data wave. [accessed 14/12/2014].
URL http://adtmag.com/articles/2014/11/12/r-gains-popularity.aspx

Team, R. C., 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org/

Vance, A., 2009. Data analysts captivated by rs power. [accessed 14/12/2014].
URL http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html?pagewanted=all&_r=0

Wickham, H., 2009. ggplot2: elegant graphics for data analysis. Springer New York.
URL http://had.co.nz/ggplot2/book