The effect of road network structure on speeding using GPS data

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Toshihiro Yokoo

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

David Levinson

May, 2016
Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor David Levinson, for the continuous support of my Master study and related research and for his patience, motivation, and immense knowledge. His guidance helped me throughout the research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master study.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor Davis and Professor Mc Cullough, for their insightful comments and encouragement. I thank my colleagues for the help and support of research. I am also thankful for Jim Tilus and staff of the Center for Writing at the University of Minnesota. Without their editing assistance, I would not have finished this thesis.

I appreciate the financial support from West Nippon Expressway Company Ltd. (NEXCO-West) that funded my tuition to attend the University of Minnesota.

I am also grateful for my friends, especially my Brazilian friends, for happy times we spent together during my stay at the University of Minnesota. I deeply appreciate their friendship. Finally, I would like to thank my family: my parents and my sister for supporting me spiritually throughout writing this thesis and my life in general.
Abstract

This paper analyzes the relationship between road network structure and speeding using GPS data collected from 152 individuals over a 7 day period. To investigate the relationship, we develop an algorithm and process to match the GPS data and GIS data accurately. Comparing actual travel speed from GPS data with posted speed limits, we measure where and when speeding occurs, and by whom. We posit that road network structure shapes the decision to speed. Speeding is large in both high speed limit zones (e.g. 60 mph (97 km/h)) and low speed limit zones (less than 25 mph (40 km/h)); in contrast, speeding is much lower in the 30 - 35 mph (48-56 km/h) zones. The results suggest driving patterns depend on the road type. We also find that if there are many intersections on the road, the average link speed (and speeding) drops. Long links are conducive to speeding.
Contents

Acknowledgements i

Abstract ii

List of Tables vi

List of Figures vii

1 Introduction 1

2 Literature review 3
   2.1 Speed limit ........................................... 3
   2.2 Reasons for speeding .................................... 4
   2.3 Transportation network structure variables .............. 4
   2.4 Use of GPS data in safety analysis ................. 5
   2.5 Map matching ......................................... 6

3 Data and Processing 8
   3.1 Data .................................................. 8
   3.2 Data processing and map-matching .................. 9
      3.2.1 Trip generation ................................ 12
      3.2.2 Remove outside Minneapolis - St. Paul region (7 counties) .... 14
      3.2.3 Remove speed that is less than or equal to 5 km/h .......... 15
      3.2.4 Mode classification ................................ 17
      3.2.5 Convert coordinate system ...................... 20
A.4.1  Trip Generation .......................................................... 89
A.4.2  Remove outside Minneapolis - St. Paul region (7 counties) .... 93
A.4.3  Remove speed that is less than 1 km/h .............................. 95
A.4.4  Mode classification ...................................................... 97
A.4.5  Remove speed that is less than or equal to 5 km/h ............... 113
A.4.6  Convert coordinate system ............................................ 115
A.4.7  Matching of person ID ................................................ 117
A.4.8  Remove matching error and Map-matching .......................... 119
A.4.9  Remove data (age is less than 16) .................................. 144
A.4.10 Remove speed limit that is 0 mph .................................. 146
A.4.11 Create variables for statistical analysis ......................... 148
A.4.12 Compile data .............................................................. 155
A.4.13 t-test: time ............................................................... 160
A.4.14 t-test: gender ........................................................... 165
A.4.15 t-test: age ............................................................... 167
A.4.16 t-test: education ....................................................... 169
A.4.17 Speeding difference depending on driver ......................... 171
A.4.18 Fixed effects estimation with integrated GPS points ........... 176
A.4.19 Computing and counting number in \( R \) .......................... 243
A.4.20 Box plot in \( R \) .......................................................... 249
A.4.21 Graph in \( R \) ............................................................ 250
A.4.22 Regression analysis in \( R \) ......................................... 269
A.4.23 Dummy variable regression in \( R \) ................................. 277
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Example of GPS data</td>
<td>9</td>
</tr>
<tr>
<td>3.2</td>
<td>Number of travel mode</td>
<td>19</td>
</tr>
<tr>
<td>3.3</td>
<td>Accuracy of mode identification</td>
<td>20</td>
</tr>
<tr>
<td>3.4</td>
<td>Accuracy of matching data (6 trips)</td>
<td>22</td>
</tr>
<tr>
<td>3.5</td>
<td>Accuracy of matching data (20 trips)</td>
<td>35</td>
</tr>
<tr>
<td>3.6</td>
<td>Example of matching data (csv-file)</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Total data in each speed limit zone</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Regression Results</td>
<td>45</td>
</tr>
<tr>
<td>4.3</td>
<td>List of dependent and independent variables</td>
<td>57</td>
</tr>
<tr>
<td>4.4</td>
<td>Output of statistical analysis in ( R )</td>
<td>58</td>
</tr>
<tr>
<td>4.5</td>
<td>Output of the dummy variable regression in ( R )</td>
<td>60</td>
</tr>
<tr>
<td>A.1</td>
<td>Independent variable: Speed limit zones</td>
<td>82</td>
</tr>
<tr>
<td>A.2</td>
<td>Independent variable: Link length</td>
<td>83</td>
</tr>
<tr>
<td>A.3</td>
<td>Independent variable: Age group</td>
<td>83</td>
</tr>
<tr>
<td>A.4</td>
<td>Independent variable: Gender</td>
<td>84</td>
</tr>
<tr>
<td>A.5</td>
<td>Independent variable: Education</td>
<td>84</td>
</tr>
<tr>
<td>A.6</td>
<td>Independent variable: Time</td>
<td>84</td>
</tr>
<tr>
<td>A.7</td>
<td>Mean value: Degree of speeding across time</td>
<td>85</td>
</tr>
<tr>
<td>A.8</td>
<td>Mean value: Degree of speeding across gender</td>
<td>85</td>
</tr>
<tr>
<td>A.9</td>
<td>Mean value: Degree of speeding across age group</td>
<td>86</td>
</tr>
<tr>
<td>A.10</td>
<td>Mean value: Degree of speeding across education level</td>
<td>87</td>
</tr>
</tbody>
</table>
List of Figures

3.1 Flow chart of data processing and map-matching ................. 10
3.2 Image of initial GPS data ............................................. 11
3.3 Example of initial GPS data (7 days) ............................. 12
3.4 Example of one trip data ............................................. 13
3.5 Remove outside Minneapolis - St. Paul region (7 counties) (Left: before, Right: after) .................................................. 14
3.6 Image of 0 km/h speed data (Number means speed) .............. 15
3.7 Remove speed that is less than 1 km/h (Left: before, Right: after) .................................................. 16
3.8 Method of calculating nearest GIS link ............................ 21
3.9 Curve Error ................................................................. 22
3.10 Intersection Error ....................................................... 23
3.11 Missing Error ......................................................... 24
3.12 Flow chart of Remove matching error and Map-matching ...... 25
3.13 Result of narrowing down the number of GIS links ............. 27
3.14 Problem of measurement error (GPS) and error at intersection area (GIS) .................................................. 29
3.15 Error at concentrated road (GIS) Part.1 .......................... 30
3.16 Error at concentrated road (GIS) Part.2 .......................... 30
3.17 Image of algorithm Part.1 (Left:grid links, Right:diagonal links) .................................................. 32
3.18 Image of algorithm Part.2 ............................................. 32
3.19 Result of removing error at intersection area and missing links .................................................. 34
3.20 Flow chart of data processing ....................................... 38
4.1 Percentage of speeding across speed limit zones .................. 42
4.2 Degree of speeding across speed limit zones ........................ 43
4.3 Relationship between link length and speeding .................... 44
4.4 Speeding by link length depending on speed limit zones . . . . . . . . . . 47
4.5 Percentage of speeding by time of day . . . . . . . . . . . . . . . . . . . 48
4.6 Degree of speeding across time . . . . . . . . . . . . . . . . . . . . . . . 49
4.7 Percentage of speeding by gender . . . . . . . . . . . . . . . . . . . . . . 50
4.8 Degree of speeding by gender . . . . . . . . . . . . . . . . . . . . . . . . 51
4.9 Percentage of speeding by age . . . . . . . . . . . . . . . . . . . . . . . . 52
4.10 Degree of speeding across age . . . . . . . . . . . . . . . . . . . . . . . . 53
4.11 Speeding by education level . . . . . . . . . . . . . . . . . . . . . . . . . . 54
4.12 Degree of speeding across education . . . . . . . . . . . . . . . . . . . . 54
4.13 Percentage of speeding per individuals . . . . . . . . . . . . . . . . . . . 55
4.14 Estimate of coefficients per individuals . . . . . . . . . . . . . . . . . . . 61
5.1 Image of complementing missing link . . . . . . . . . . . . . . . . . . . . . 64
A.1 GIS map . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
A.2 Buffer process 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
A.3 Buffer process 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
A.4 GIS links . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
A.5 GIS buffer(s) (10m) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
A.6 Add Delimited Text Layer . . . . . . . . . . . . . . . . . . . . . . . . . . 76
A.7 Create a Layer from a Delimited Text File . . . . . . . . . . . . . . . . . . 77
A.8 Coordinate Reference System Selector . . . . . . . . . . . . . . . . . . . . 78
A.9 Result of output . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
A.10 Select by location 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
A.11 Select by location 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
A.12 Relationship between link length and speeding . . . . . . . . . . . . . . 81
A.13 Significant code: Degree of speeding across time . . . . . . . . . . . . . 86
A.14 Significant code: Degree of speeding across age group . . . . . . . . . . 87
A.15 Significant code: Degree of speeding across education level . . . . . . . 88
Chapter 1

Introduction

Speeding behavior, driving faster than the speed limit, is affected by the road environment, the vehicle, and the driver. For example on interurban roads, young, educated, male drivers with cars are likely to exceed the speed limit [1, 2]. Considering this relationship, it is important to further understand specific influences on speeding behavior. This is especially true as studies show traffic crashes are often due to driver “manipulation error” (e.g., speeding and illegal overtaking) and “perception error” (e.g., wrong assessment of speed and distance) [1], and as speed increases, the percentage of both errors increases.

We posit that road network structures shape the decision to speed. Others have found that the percentage of speeding is large in both high speed limit zones (70 mph) and low speed limit zones (20, 30 mph); on the other hand, speeding is not remarkable in 60 mph zones [3]. This result suggests that drivers change their driving patterns depending on the road type. If there are many intersections on the road network, the average speed on the road linking those intersections would reduce due to external factors (e.g. traffic light, traffic condition, and approaching vehicle from other direction).

In this thesis, we analyze the relationship between the road network structure variables and the percentage of speeding from the GPS data. To investigate the relationship, we develop an algorithm and process to match the GPS data and GIS data accurately.

- Chapter 2 presents a literature review
- Chapter 3 explains the material and methodology

1
• Chapter 4 describes results of analysis

• Chapter 5 discusses the analyses
Chapter 2

Literature review

2.1 Speed limit

Posted speed limits in Minnesota are regulated by statute\(^1\) and the statute mentions that speed limit differs depending on the road type. (e.g., 30 mph on streets in urban districts, 65 mph on expressways, and 70 mph on rural interstate highways). Based on the statutes, the Minnesota Department of Transportation (MnDOT) establishes the posted speed limit on each road considering several factors (e.g., road type and condition, existing traffic control devices, and crash history) \(^4\). Moreover, MnDOT utilizes “the 85th percentile” as a guideline of speed limit, and therefore the posted speed limit would be close to the “the speed at or below which 85 percent of vehicles travel” \(^5\). This level is based on a study by David Solomon in 1964. He analyzed the relationship between accident and speed, and the results showed that the accident rate increases upward at the 85 to 90 percentile speed. This finding became common and went mainstream for traffic engineers \(^6\). However, there is doubt about the 85th percentile rule because this definition ignores the risk for both pedestrians and cyclists. Solomon’s result was analyzed in highways, but now the 85th percentile is used not only in highways but also local residential roads. This is risky for both pedestrians and cyclists \(^7\).

\(^1\) 2014 Minnesota Statute: 169.14 SPEED LIMITS, ZONES; RADAR.
2.2 Reasons for speeding

Although speed limit is defined by engineering study and practice, some drivers don’t follow the speed limit. It is widely understood that vehicle speed is influenced by three factors: drivers, vehicles and the road environment [8]. Greek data show that elderly, female and uneducated drivers who don’t often drive on interurban roads are likely to follow the speed limit [1]. The traffic violator thinks the posted speed limit is not trustworthy, and they exceed the speed limit because “they are in a hurry”, there is an “absence of police”, and “speed limits are not reliable” [1]. A Survey of 207 drivers asked to answer the element among 14 factors that is related to the speed choice on the curve section found that “sight distance” has the most impact for the speed choice, followed by “pavement condition”, “sharp curvature”, and “additional warning signing”, while “existence of free roadside space” and “speed limit signing” had little effect on speed choice to the driver [8]. Logically non-violators are more likely to comply with speed signage than violators.

Speed limit compliance in work zones is influenced by the vehicle gaps, traffic volume, time of day, leader vehicle, and vehicle size [2]. For example, light vehicles are likely to ignore the speed limit more than medium and heavy vehicles. When leading vehicle’s speed is higher, the following vehicle also tends to exceed the speed limit. When the distance between vehicles is large, the driver is likely to exceed the posted speed limit. Also, high traffic volume tends to generate the non-compliance of speed limit. As the vehicle weight increases, the mean and variance of vehicle speed becomes small [9], clarifying that the level of speed limit compliance depends on surrounding traffic, road conditions, type of vehicle, and drivers.

2.3 Transportation network structure variables

Although some researchers analyze reasons for speeding regarding drivers, traffic condition, and vehicles, it is assumed that the road network structure also has an effect on speeding. Considering the relationship between transportation network structure variables (e.g., connectivity, hierarchy, circuity, treeness, entropy, and accessibility) and transportation performance (e.g., journey-to-work time and automobile mode share)
across 50 U. S. cities, it was found that large network connectivity decreases the commute time, large accessibility reduces auto mode share, and large city size increases the delay [10].

This analysis might be applied to speed limit compliance. For example, it is posited that a small number of intersections in the city (long link length) would be correlated to speeding. In order to analyze the relationship between network structure variables and speeding, we need to collect enormous amount of traffic data over a broad area. In general, a traffic detector is used to analyze the traffic data. However, the amount of data on local roads is small. Therefore, in our thesis we use GPS data because GPS data contains several information; time, position (latitude, longitude, altitude), distance, speed, the number of satellites [11]. If GPS data covers the whole network, it is more effective to analyze the relationship between network structures and speeding than loop detectors on selected facilities.

2.4 Use of GPS data in safety analysis

GPS data contains strengths and weaknesses. In terms of strengths, this data is likely to reflect a daily driving pattern. Compared to the driving experiment on a test route, participants are likely to drive on routes they prefer. Drivers live in Minnesota, so they would generally be familiar with their driving route. Moreover, compared with loop detector data, we can evaluate time series data of each driver and analyze extensive coverage not only for trunk roads but also residential roads [12]. GPS data is more practical and trustworthy for collecting traffic data than inductive loop detectors because inductive loop detectors are expensive to install and maintain and they are prone to failure [12]. However, GPS data is limited because route and date are differ by driver. Each driver drove under different conditions. It is assumed that some driving patterns are affected by weather, congestion and road type, and it is also difficult to define external factors of driving (e.g. a traffic light and a sudden deceleration due to the preceding car). Furthermore, GPS requires map matching [13].
2.5 Map matching

GPS data contains speed information while GIS maps possess the posted speed limit data, therefore when we match the GPS data with GIS maps, we can analyze the relationships between driving speed data and road networks. However, there are some problems about this process regarding the accuracy of GPS data and calculation time.

GPS devices provide highly accurate, but not perfect data because of the effect of atmospheric conditions, neighborhood structure, and selective availability [11]. That research shows that 86.5% of GPS data was accurate within 1.5m and 99.89% accurate within 2.0m. The accuracy of speed data depends on the device and road alignment (straight or curve). For example, non-differential GPS device [11] and the Wide-Angle Augmentation System (WAAS) [14] were used to analyze the accuracy of GPS data. In a straight line, 97.9% of non-differential GPS device speed data was within 0.2 m/s; on the other hand, 67% of WAAS GPS speed data was within 0.2 m/s. On a curved area, 86.7% of non-differential GPS device speed data was within 0.2 m/s. On the contrary, 65% of WAAS GPS speed data was within 0.2 m/s. This result shows that the accuracy differs by GPS device, and straight line data indicates higher accuracy than curve area.

Accuracy of speed data depends on the method of calculating speed; “Doppler shift” and “differences in GPS position over time”. The result in straight course showed that 90.8% of “Doppler shift” data was accurate within 0.1 m/s and 66.5% of “differences in GPS position over time” data was accurate within 0.1 m/s [11].

This inaccuracy of both GIS and GPS data lead to the difficulty of finding the actual route when we match GPS data and GIS map; however, if we match every GPS data manually, it would take a huge amount of time. Therefore, it is desirable to find a proper map-matching method and create an algorithm for analyzing the GPS data. In the ST-matching algorithm, each GPS point chooses the candidate of actual road network point and calculates the probability [15]. The probability is decided by a zero-mean normal distribution with a standard deviation of 20 meters. While the algorithm chooses the highest probability point for the map-matching, it also considers the following two key factors to increase the accuracy of map-matching:

- “True paths tend to be direct, rather than roundabout” [15] – Because of the measurement errors, some GPS points are likely to be located at different roads
that are close to the actual road. Therefore, if previous and future GPS data are located in the same road, one point that is located along another road will be modified to the same road.

- “True paths tend to follow the speed constraints of the road” [15] – In some areas, highways and local roads are close to each other, and therefore it is difficult to define the actual road network link from the nearest distance between the GPS data and GIS map. The author considers the average vehicle speed, and if the average speed is high enough, the driver is likely to use highways more than local roads.

Then, the author tests the ST-Matching in Beijing road network and this network contains 58,624 nodes and 130,714 links. The results mention that the accuracy and calculation time is related to the number of candidate points that algorithm chooses. As the number of candidates increases, the accuracy of matching data increases; on the other hand, the running time increases. For example, when the number of candidates is one point, the accuracy is about 80%. On the contrary, when the number of candidates is three points, the accuracy is approximately 95%.

One of the reasons for the map-matching failure is a missing link in the map [16]. Comparison of the accuracy of map-matching between high-resolution and low-resolution networks show that low-resolution networks cannot find actual trip links, and therefore, detailed GIS network maps are important for the map-matching and it is desirable to use the most accurate GIS map that covers majority of routes in the area [17]
Chapter 3

Data and Processing

3.1 Data

This paper uses GPS data from the 2010 Twin Cities (Minneapolis - St. Paul area) Travel Behavior Inventory (TBI2010) administered by the Metropolitan Council between 2010 and 2012 (most data was collected in 2011). Each subject in the GPS component of the TBI carried a GPS pendant for 7 days. The raw GPS data contains the following trip information: Speed (km/h), Longitude, Latitude, Altitude (meters), Date (year/month/day), Time (hour/minute/second), Distance (meters), Course (degree), Number of satellites, HDOP. An example of GPS data is Table 3.1. Seven days of movement for each person is recorded in the data, and the trajectories were recorded every second as Table 3.1 shows.

In order to check the speed limit compliance, we will need to identify the road that participants drove, and then compare the driving speed with the posted speed limit. Therefore, we mainly use location data (Longitude and Latitude) and speed data (Speed (km/h)) in our analysis.

However, this GPS data by itself does not contain personal information (e.g., gender, age). Therefore, we use the associated records from the TBI2010 Household Interview Survey as complementary data (e.g., Household data, Personal information, Trip data). These data can be matched on TripID. Among 274 GPS subjects (drawn from 250 households), 152 travelers have trip IDs matching the survey and no other data problems, allowing us to analyze 152 participants. (Although the total with trip ID
Table 3.1: Example of GPS data

<table>
<thead>
<tr>
<th>Longitude</th>
<th>Latitude</th>
<th>Speed (kilometer)</th>
<th>Course (degrees)</th>
<th>Number Of Satellites</th>
</tr>
</thead>
<tbody>
<tr>
<td>-86.283816</td>
<td>39.712155</td>
<td>20</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>-86.28378</td>
<td>39.712171</td>
<td>19</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>-86.283718</td>
<td>39.712211</td>
<td>19</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>-86.28367</td>
<td>39.712255</td>
<td>19</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>-86.283631</td>
<td>39.712276</td>
<td>17</td>
<td>46</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HDOP</th>
<th>Altitude (meters)</th>
<th>DD/MM/YY</th>
<th>HH:MM:SS</th>
<th>Distance (meters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.29</td>
<td>223</td>
<td>23/6/2011</td>
<td>15:45:34</td>
<td>0</td>
</tr>
<tr>
<td>6.3</td>
<td>223</td>
<td>23/6/2011</td>
<td>15:45:35</td>
<td>3</td>
</tr>
<tr>
<td>5.4</td>
<td>223</td>
<td>23/6/2011</td>
<td>15:45:36</td>
<td>6</td>
</tr>
<tr>
<td>6.29</td>
<td>223</td>
<td>23/6/2011</td>
<td>15:45:37</td>
<td>6</td>
</tr>
<tr>
<td>6.29</td>
<td>223</td>
<td>23/6/2011</td>
<td>15:45:38</td>
<td>4</td>
</tr>
</tbody>
</table>

is 160, the age of 8 travelers is under 16.) We append the GPS data with personal information describing each subject’s gender, age, and education.

With the purpose of understanding actual speed limit data, we use a GIS map maintained by the Metropolitan Council and The Lawrence Group (TLG) that covers the majority of routes in the Twin Cities seven county metropolitan area. This GIS street map has the most accuracy at the present time. The map contains 290,231 links, and each link has several attributes (e.g. speed limit, length of link, street name, one-way).

To conduct the analysis, we use QGIS [20], an open source geographic information system.

### 3.2 Data processing and map-matching

In order to match GPS data and GIS data, we use 278 GPS subjects that are recorded in a CSV file; however, each file contains an enormous amount of data (total size is 1.89GB). Therefore we need to create an algorithm to improve the efficiency and speed of calculation and data processing while keeping the desirable accuracy.

Before the map matching, we need to extract the GPS data where the travel mode is car and it has personal information. The process of extracting appropriate driving
data is shown in Figure 3.1 that has 6 steps from “1: Trip generation” to “6: Matching of trip ID”.

![Flow chart of data processing and map-matching](image)

**Figure 3.1: Flow chart of data processing and map-matching**

Figure 3.2 is an example of initial GPS data. First, we divide the initial GPS data into each trip data (From Trip 1 to Trip 6) because time of initial GPS data is disconnected. Then, based on speed and location data we assign a travel mode from five types: Walk, Train, Bus, Bicycle, and Car. After mode identification, we extract driving trips (Trips 1 and 6) and match GPS data and GIS data. The algorithm is implemented in the Python language because QGIS supports Python.

In terms of the algorithm, we use an algorithm and methodology developed previously [21] as a reference. Some algorithms (“1: Trip generation” A.4.1, “3: Remove speed less than or equal to 5 km/h” A.4.3 & A.4.5, “4: Mode classification” A.4.4) are reused [21], but converted to execute on Mac rather than Windows. We explain the details of each step in the next section.
Figure 3.2: Image of initial GPS data
3.2.1 Trip generation

First, we extract the driving trips from GPS data of TBI2010. Each GPS data contains 7 days travel data and as Figure 3.3 shows, GPS travel records initially contains data from multiple trips. Since the initial GPS data doesn’t record each trip and the time that the GPS data is disconnected, we define the trip based on the time difference. The algorithm divides the initial GPS data into the trip data when a time difference of the GPS data is more than 300 seconds. The code of this algorithm is mentioned in A.4.1.

As a result, the initial 278 GPS subjects made 16,902 trips. It means that each GPS subject made about 60 trip data on average. Figure 3.4 illustrates the GPS data after being decomposed into individual trips.

Figure 3.3: Example of initial GPS data (7 days)
Figure 3.4: Example of one trip data
3.2.2 Remove outside Minneapolis - St. Paul region (7 counties)

Although participants live in the Twin Cities metro area, some people traveled outside of the Minneapolis area. The TLG map, known as the GIS map, covers the seven county metropolitan area in Minnesota; however, some GPS data are not located on this map. To remove the GPS data that are plotted outside Minneapolis - St.Paul region (7 counties), we use the following boundary to define the study area:

\[-94.0123 \leq \text{Longitude} \leq -92.7397\]

\[44.4714 \leq \text{Latitude} \leq 45.4139\]

Data outside these boundary coordinates are removed. Based on this algorithm, some trip data are modified and 791 trip data are removed. Then, the total number of trip decreases to 16,111 trip data. An example of the result is in Figure 3.5. Figure 3.5 shows GPS data that are plotted outside of the map are removed. The code of this algorithm is in A.4.2.

Figure 3.5: Remove outside Minneapolis - St. Paul region (7 counties) (Left: before, Right: after)
3.2.3 Remove speed that is less than or equal to 5 km/h

Before the mode classification, we remove the speed data that is less than 1 km/h.

As Figure 3.6 shows, enormous GPS data are gathered at one place and the speed of this area is almost 0 km/h. It is considered that participants stop moving because of shopping or preparation to go out. These data are included in the trip data especially at the beginning or ending. When these speed data (0 km/h) account for the large percentage of the trip data, they affect the result of the mode classification because the average speed decreases.

Figure 3.6: Image of 0 km/h speed data (Number means speed)
Figure 3.7: Remove speed that is less than 1 km/h (Left: before, Right: after)

An example of the result is in Figure 3.7. As Figure 3.7 shows, some GPS points are concentrated at the center area of the map and these speeds are 0 km/h. It seems that the vehicle was stopped at an intersection due to the traffic light. After the algorithm, these points are removed while other speed data remain. The code of this algorithm is mentioned in A.4.3. Due to this algorithm, 3,543 trips are removed and the total number of trips decreases to 12,568.

After the mode classification, we remove speed data that is less than or equal to 5 km/h from the driving trip data in order to remove the effect of traffic lights and traffic jams.

We want a driving condition that is not affected by external factors for analyzing speed limit compliance. If we include the effect of traffic lights and traffic jams, the result of speed limit compliance would be overestimated. The code of this algorithm is mentioned in A.4.5. In this algorithm, no trip data changes while some data are deleted.
3.2.4 Mode classification

GPS data contains not only driving data but also walking, biking, and transit data (typically lower speed data). The purpose of our research is to focus on driving speeding behavior, so the algorithm extracts the driving data. The definition of mode classification is from Wenyun [21]. The condition to classify the data is as follows; (When the data doesn’t apply to any of travel mode conditions, it becomes an error mode)

1. Walk:
   - Maximum speed of all points \( \leq 20\text{km/h} \);
   - 85th percentile of speed of all points \( \leq 10\text{km/h} \);
   - Average speed of all points \( \leq 6\text{km/h} \);
   - Duration \( \geq 60\text{s} \).

2. Rail:
   - Distance from first point of speed accelerates to 10km/h to the nearest rail station \( \leq 150\text{m} \);
   - Distance from last point that speed is greater than 10km/h to the nearest rail station \( \leq 150\text{m} \);
   - Average speed of all points \( \geq 10\text{km/h} \);
   - Duration \( \geq 60\text{s} \).

3. Bus:
   - Distance from first point of speed accelerates to 10km/h to the nearest bus stop \( \leq 50\text{m} \);
   - Distance from last point that speed is greater than 10km/h to the nearest bus stop \( \leq 50\text{m} \);
   - Average speed of all points \( \geq 10\text{km/h} \);
   - Duration \( \geq 60\text{s} \).

4. Bicycle:
• 85th percentile of speed of all point >10km/h and <20km/h;
• Max speed of all points ≤ 30km/h;
• Duration ≥ 60s.

5. Car:
• The remaining trip segments with average speed of all points ≥ 10km/h;
• Duration ≥ 60s.

Both bus stop and train station data are obtained from http://datafinder.org [22] in the form of a shapefile, and then we save the attributes as a CSV file in QGIS. The bus stop data is from “Transit Stops” [23] and the train station data is from “Transitway Stations” [24].

After that we revise bus stop and train station data because the date of these data are different from GPS data. The bus stop data was updated in 2015 and the train station data was recorded in 2014 [22]; on the other hand, the GPS data was recorded in 2011. Therefore, we revise the bus stop (train station) data into the version in 2011 when possible.

1. Train station data

We revise the train station data based on the attribute of transitway name (Blue line, Green line, Red line, and Northstar Line). Northstar Commuter Rail Line connects to downtown Minneapolis and Big Lake, and the service began in 2009 [25, 26]. The METRO Blue and Green Lines are light-rail services and the Blue Line connects downtown Minneapolis and Mall of America, and started service in 2004 [27]. The Green Line connects to downtown Minneapolis and downtown St.Paul, and opened in 2014 [28]. Therefore, we remove the Green Line data from the train station data.

The original data also contains the METRO Red Line, but the METRO Red Line is a freeway bus rapid transit service [29]. After checking the trip data, we find that no GPS trips uses the METRO Red line.
2. Bus stop data

The bus service is operated by Metro Transit and this database has 19,285 active and inactive bus stops in the Twin Cities seven county metropolitan area \[23\]. From the attribute ‘ada_comm’ of database, we remove the data of the bus stop before 2011. After this process, the total number of bus stops reduces to 16,545.

As the definition of mode classification indicates, we calculate the distance from the GPS data to the bus stop (train station). Before calculating, we change the coordinate system of the bus stop data (train station) because the GPS data is geographic latitude and longitude; while the bus stop data (train station) uses the Universal Transverse Mercator (UTM). The detail of converting the coordinate system is described in \(3.2.5\). The code of this algorithm is mentioned in \(A.4.4\).

<table>
<thead>
<tr>
<th>Mode</th>
<th>Number of trip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>4,499</td>
</tr>
<tr>
<td>Bus</td>
<td>92</td>
</tr>
<tr>
<td>Train</td>
<td>3</td>
</tr>
<tr>
<td>Bike</td>
<td>115</td>
</tr>
<tr>
<td>Drive</td>
<td>4,560</td>
</tr>
<tr>
<td>Error</td>
<td>3,299</td>
</tr>
<tr>
<td>Total</td>
<td>12,568</td>
</tr>
</tbody>
</table>

The result of mode classification is in Table 3.2. After calculating the algorithm, 4,560 data are recorded as drive. Besides drive, many trips are classified as walk mode or error mode. It suggests that a lot of trips have small average speed (<10km/h) or small duration time (<60 sec).

We also check the accuracy of mode classification. In the TBI2010 Household Interview Survey, participants recorded their trip for a selected day. While each individual could report up to 3 modes, most of participants recorded only one. TBI data has 160 households’ data. Among them, the date of 140 modes match the date of GPS data. Note that there are multiple GPS trips for each selected day. When one of them matches the mode of TBI data, it is recorded as accurate. As Table 3.3 illustrates, 4% of non-auto trips show up as auto. Therefore, our methods ensure few false positives.
### Table 3.3: Accuracy of mode identification

Identified as

<table>
<thead>
<tr>
<th>TBI data</th>
<th>Walk</th>
<th>Rail</th>
<th>Bus</th>
<th>Bike</th>
<th>Car</th>
<th>Error</th>
<th>Total</th>
<th>Success rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>88%</td>
</tr>
<tr>
<td>Rail</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>#DIV/0!</td>
</tr>
<tr>
<td>Bus</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>Bike</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>33%</td>
</tr>
<tr>
<td>Car</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>106</td>
<td>5</td>
<td>119</td>
<td>89%</td>
</tr>
<tr>
<td>Total</td>
<td>21</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>110</td>
<td>7</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>Success rate (%)</td>
<td>67%</td>
<td>#DIV/0!</td>
<td>0%</td>
<td>100%</td>
<td>96%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 3.2.5 Convert coordinate system

In order to match the GPS data and GIS data, the coordinate system of these two data should be the same; however, the GPS data is geographic latitude and longitude; while the TLG Map uses the Universal Transverse Mercator (UTM). We convert longitude and latitude data into UTM data following standard algorithms [30, 31]. The formula is mentioned in A.1 and the code of this algorithm is mentioned in A.4.6.

#### 3.2.6 Matching person ID

And then, we match person ID of GPS data with the TBI2010 Household Interview Survey data in order to add personal information (gender, age, and education) to the GPS data for the statistical analysis. It is noted that file name of initial GPS data is person ID. If person ID doesn’t match, the GPS trips removed. The code of this algorithm is mentioned in A.4.7.

After this process, 2,710 trip data (160 individuals) match the TBI data. However, about 40.6 % of data are removed in this process. Thanks to this process, we can extract driving trip data that has personal information.

#### 3.2.7 Map-matching process

This algorithm is a main part of this study. Because of measurement error and map resolution, GPS location is not the exact location of the GIS link. Therefore, several
map-matching methods have been developed to identify the correct GIS road link \[32, 33\]. First we assume that the closest length between GPS and GIS data indicates the accurate map-matching. To begin, we match the GPS data and GIS data when the length between GPS point and GIS link is smallest. In order to estimate the length, the algorithm calculates the area of parallelogram (D) by vector outer product operations, then calculate the height of parallelogram (H) by dividing the area (D) by the length of the base (L), as shown in Figure 3.8.

When the height is smallest, the algorithm extracts the road information (e.g., link length, street name, roadtype, and speed limit) from the GIS data, and combines this data into the GPS data.

![Figure 3.8: Method of calculating nearest GIS link](image)

**Problem of accuracy**

Based on the above matching process, we analyze the accuracy of matching data by checking the street name manually. When we check 6 trips of data (shown in Table 3.4), the result shows that 76.0 % of data are accurate on average; however, this percentage of accuracy is not enough to analyze the relationship between road network variables and the degree of speeding. In order to improve the accuracy we need to find the reason of inaccuracy on the map, and we find out that inaccuracy is significant in three types of area.
Table 3.4: Accuracy of matching data (6 trips)

<table>
<thead>
<tr>
<th>Trip</th>
<th># of correct data</th>
<th># of total data</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>trip1</td>
<td>396</td>
<td>607</td>
<td>65.2%</td>
</tr>
<tr>
<td>trip2</td>
<td>307</td>
<td>374</td>
<td>82.1%</td>
</tr>
<tr>
<td>trip3</td>
<td>577</td>
<td>802</td>
<td>71.9%</td>
</tr>
<tr>
<td>trip4</td>
<td>524</td>
<td>642</td>
<td>81.6%</td>
</tr>
<tr>
<td>trip5</td>
<td>852</td>
<td>949</td>
<td>89.8%</td>
</tr>
<tr>
<td>trip6</td>
<td>768</td>
<td>1,134</td>
<td>67.7%</td>
</tr>
<tr>
<td>total</td>
<td>3,424</td>
<td>4,508</td>
<td>76.0%</td>
</tr>
</tbody>
</table>

Curve section

The GIS link between two points is not necessarily a straight line like Figure 3.8; some GPS links are curved as seen Figure 3.9. When the algorithm calculates the shortest length between the GPS data and the curved GIS link, the algorithm is more likely to select the wrong GIS link. This failure significantly occurs in ramp section of interstate or freeway.

Figure 3.9: Curve Error
(Red line: used GIS link, Purple line: GIS link of algorithm, Green dot: GPS data)
Intersection

When the GPS point is near an intersection area, a matching failure is likely to occur. As Figure 3.10 shows, the vehicle moves on the vertical road (NICOLLET AVE S); however, the algorithm selects coordinates on the horizontal road (86TH ST W).

Missing link

Although the TLG network data is the most accurate digital map in the Metropolitan area, some road information is missing. When vehicles drive on these missing roads, it is difficult to match the data correctly (Figure 3.11).
Figure 3.11: Missing Error
(Line: GIS links, Dot: GPS data)
3.2.8 Remove matching error and Map-matching

Based on the result of the map-matching algorithm, simply matching to the closest point between GPS and GIS data is far from perfect, and many errors are found at intersections and due to missing links. To control for that, when GPS data is near an intersection area or far from a mapped link, this GPS point is removed. This process is useful for removing the error, but wastes data. The process is in Figure 3.12. The code of this algorithm is mentioned in A.4.8.

Figure 3.12: Flow chart of Remove matching error and Map-matching
Narrowing down the number of GIS links
(From “Create Buffer” to “Select by location(1)”)

When one GPS point is used to calculate the length to all GIS links, it takes a lot of calculation time because the number of GIS points is enormous (≈290,231). Therefore, we narrow down the area of the object to reduce the running time of the algorithm by the function of QGIS.

First we create a buffer for the GIS links (A.2). The initial GIS links are line like in Figure A.4 and it describes the centerline of the road. Then, we create the road area by using the “buffer” function in QGIS. It is true that road width is different depending on the road; however, we set “10m” as the buffer width for all roads. The image of the 10m buffer zone is shown in Figure A.5. It is noted that a 10m buffer shows accurate map-matching results more than other buffer sizes. Moreover, the 10m buffer retains a large amount of data.

And then, we extract specific GIS links that intersect with GPS points by the function “Select by location” (A.2.2). With the process, we can extract GIS links that are close to GPS points, and the extracted GIS link is similar to the driving route. (Figure 3.13)
Figure 3.13: Result of narrowing down the number of GIS links 
the yellow area (links) is selected by the function “Select by location”
Problem of measurement error (GPS) and error at intersection area (GIS)

Although this function can reduce the number of GIS links, it also has a problem. When GPS data is located far from the GIS link because of measurement error, some actual driving routes are removed. Figure 3.14 shows that participants drove on “NICOLLET AVE S”; however, this road is not chosen as a candidate driving route.

Moreover, unused routes remain like “90TH ST W”, “BLAISDELL AVE S”, and “PILLSBURY AVE S”. It is noted that these streets are unused driving routes; however, these streets are selected because they intersect with actual driving routes.

Error at concentrated road (GIS)

If roads are far from each other, the process of “Select by location” chooses only the actual driving route. But if some roads are close to each other, an unused driving route may be chosen. As Figure 3.15 shows, many ramp section of an interstate (such as “WB 1494 TO SB HWY77” and “NB HWY77 TO WB 1494”) remain as candidates of driving routes although these links are not actual driving routes. Moreover, as Figure 3.16 shows when local roads are close to interstates, the local roads are also chosen as a candidate drive routes like “78TH ST E”.

Especially, when we increase the buffer width, unused driving routes are likely to be chosen. On the other hand, when the buffer width is very narrow, some actual driving routes are removed because many GPS points cannot be plotted within the buffer area. Based on this outcome, we use “10m” as the buffer width.
Figure 3.14: Problem of measurement error (GPS) and error at intersection area (GIS)
Figure 3.15: Error at concentrated road (GIS) Part.1

Figure 3.16: Error at concentrated road (GIS) Part.2
Remove error at intersections and missing links
(From “Remove GPS data” to “Select by location(2)”)

Figure 3.17 is the image of the algorithm that removes error data points when the GPS data is in the intersection of two buffers and when they are not within a buffer. Each link is used to create a buffer containing all points within 10m of the GIS link. As Figure 3.17 shows, the intersected GIS link area has more than two layers. The algorithm counts the number when GPS data is inside the range area. When the GPS point is near an intersection, the number of layers becomes greater than two. Also, when GPS data is outside of the range of GIS data, the number of layers is zero. The algorithm only saves the GPS data with one layer.

This process can remove the effect of intersection, but there are some problems with this process. As Figure 3.17 shows, when the streets are laid out in a grid running north-south and east-west, the covering area of removed GPS data is small. However, when the streets are laid out diagonally or adjacent streets are close to each other (Figure 3.18), many areas have greater than two layers and a lot of GPS data are removed because of this process.
Figure 3.17: Image of algorithm Part.1 (Left: grid links, Right: diagonal links)

Figure 3.18: Image of algorithm Part.2
After removing GPS data at intersections and on missing links, we extract GIS links again by using the function “Select by location” that is the same process as A.2.2. Figure 3.19 shows that this method can remove some unused links and save actual driving route links. The problem of this process is that most of the GPS data is removed when several road links are close to each other. For example, parallel roads and ramps are near the mainline of the interstate as shown in Figure 3.15. Even if drivers use the interstate, these GPS data are likely to be removed because of the process as we mentioned in Figure 3.18.

Moreover, we can see other errors that unused routes (links) are chosen at the bottom-right part of the map (Figure 3.19). As Figure 3.14 shows, GPS data is more than 10m away from the actual driving route (“NICOLLET AVE S”). In this case, actual links are removed and unused links remain.
Figure 3.19: Result of removing error at intersection area and missing links
Yellow area (links) is selected by the function “Select by location”
Map-matching

After this process, we match the extracted GPS data and GIS link by calculating the smallest length between the GPS data and GIS link (Figure 3.8). Also, we analyze the accuracy from the result of map-matching manually by visual inspection in QGIS. Visual inspection shows that 97.4% of matched points on 20 trips are accurate (Table 3.5), and therefore this result is satisfactory. The problem is that about 51.0% of the GPS data are removed and some driving information during the trip is missing due to the process. Therefore, we cannot analyze continuous trip data. In order to ensure few false positives, there are many false negatives. After this algorithm, the total number of trips is 2,710 (160 drivers).

<table>
<thead>
<tr>
<th>Trip</th>
<th># of accuracy</th>
<th># of total data</th>
<th>Accuracy (%)</th>
<th># of initial trip data</th>
<th>% of removal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trip1</td>
<td>7,103</td>
<td>7,283</td>
<td>97.5%</td>
<td>12,144</td>
<td>40.0%</td>
</tr>
<tr>
<td>Trip2</td>
<td>3,502</td>
<td>3,544</td>
<td>98.8%</td>
<td>5,786</td>
<td>38.7%</td>
</tr>
<tr>
<td>Trip3</td>
<td>2,945</td>
<td>3,048</td>
<td>96.6%</td>
<td>16,152</td>
<td>81.1%</td>
</tr>
<tr>
<td>Trip4</td>
<td>2,980</td>
<td>3,063</td>
<td>97.3%</td>
<td>6,056</td>
<td>49.4%</td>
</tr>
<tr>
<td>Trip5</td>
<td>2,645</td>
<td>2,655</td>
<td>99.6%</td>
<td>3,857</td>
<td>31.2%</td>
</tr>
<tr>
<td>Trip6</td>
<td>2,502</td>
<td>2,540</td>
<td>98.5%</td>
<td>5,814</td>
<td>56.3%</td>
</tr>
<tr>
<td>Trip7</td>
<td>2,496</td>
<td>2,536</td>
<td>98.4%</td>
<td>4,688</td>
<td>45.9%</td>
</tr>
<tr>
<td>Trip8</td>
<td>2,442</td>
<td>2,473</td>
<td>98.7%</td>
<td>4,666</td>
<td>47.0%</td>
</tr>
<tr>
<td>Trip9</td>
<td>2,337</td>
<td>2,439</td>
<td>95.8%</td>
<td>5,152</td>
<td>52.7%</td>
</tr>
<tr>
<td>Trip10</td>
<td>2,300</td>
<td>2,351</td>
<td>97.8%</td>
<td>4,783</td>
<td>50.8%</td>
</tr>
<tr>
<td>Trip11</td>
<td>2,280</td>
<td>2,390</td>
<td>95.4%</td>
<td>3,739</td>
<td>36.1%</td>
</tr>
<tr>
<td>Trip12</td>
<td>2,325</td>
<td>2,344</td>
<td>99.2%</td>
<td>5,637</td>
<td>58.4%</td>
</tr>
<tr>
<td>Trip13</td>
<td>2,176</td>
<td>2,311</td>
<td>94.2%</td>
<td>4,189</td>
<td>44.8%</td>
</tr>
<tr>
<td>Trip14</td>
<td>2,003</td>
<td>2,236</td>
<td>89.6%</td>
<td>3,722</td>
<td>39.9%</td>
</tr>
<tr>
<td>Trip15</td>
<td>2,138</td>
<td>2,171</td>
<td>98.5%</td>
<td>3,885</td>
<td>44.1%</td>
</tr>
<tr>
<td>Trip16</td>
<td>2,103</td>
<td>2,160</td>
<td>97.4%</td>
<td>3,586</td>
<td>39.8%</td>
</tr>
<tr>
<td>Trip17</td>
<td>2,063</td>
<td>2,117</td>
<td>97.4%</td>
<td>4,032</td>
<td>47.5%</td>
</tr>
<tr>
<td>Trip18</td>
<td>2,096</td>
<td>2,106</td>
<td>99.5%</td>
<td>5,468</td>
<td>61.5%</td>
</tr>
<tr>
<td>Trip19</td>
<td>2,044</td>
<td>2,100</td>
<td>97.3%</td>
<td>3,051</td>
<td>31.2%</td>
</tr>
<tr>
<td>Trip20</td>
<td>2,115</td>
<td>2,136</td>
<td>99.0%</td>
<td>3,730</td>
<td>42.7%</td>
</tr>
<tr>
<td>Total</td>
<td>52,595</td>
<td>54,003</td>
<td>97.4%</td>
<td>110,137</td>
<td>51.0%</td>
</tr>
</tbody>
</table>
As Table 3.6 shows, the results of map-matching are recorded as CSV file. CSV file has the following attributes:

1. Location data [Longitude, Latitude]
2. Driving speed [Speed(kilometer)]
3. Time of date [DD/MM/YY, HH:MM:SS]
4. Location data (UTM) [GX, GY]
5. Length between GPS data and GIS link [Min_height]
6. Street name [Streetname]
7. Street length (Link length, Unit: meter) [Streetlength]
8. Road type [Roadtype]
9. Speed limit of road link [Speedlimit(mph)]

Attributes 1,2,3 are from original GPS data, 4 is the result of coordinate conversion, 5 is the calculation result (Figure 3.8), 6,7,8,9 is from original GIS map data.

3.3 Data processing for Statistical Analysis

With the purpose of analyzing the hypotheses described in the next chapter, we need to calculate the speeding from the driving speed of GPS data and the speed limit of GIS
Table 3.6: Example of matching data (csv-file)

<table>
<thead>
<tr>
<th>Longitude</th>
<th>Latitude</th>
<th>Speed(kilometer)</th>
<th>DD/MM/YY</th>
<th>HH:MM:SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-93.245091</td>
<td>44.875816</td>
<td>59</td>
<td>24/6/2011</td>
<td>10:34:18</td>
</tr>
<tr>
<td>-93.245296</td>
<td>44.875791</td>
<td>56</td>
<td>24/6/2011</td>
<td>10:34:19</td>
</tr>
<tr>
<td>-93.245486</td>
<td>44.875768</td>
<td>50</td>
<td>24/6/2011</td>
<td>10:34:20</td>
</tr>
<tr>
<td>-93.245661</td>
<td>44.875733</td>
<td>48</td>
<td>24/6/2011</td>
<td>10:34:21</td>
</tr>
<tr>
<td>-93.245816</td>
<td>44.875711</td>
<td>41</td>
<td>24/6/2011</td>
<td>10:34:22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GX</th>
<th>GY</th>
<th>Min_hight</th>
<th>Streetname</th>
<th>Streetlength</th>
</tr>
</thead>
<tbody>
<tr>
<td>480641.3983</td>
<td>4969184.497</td>
<td>29.69856906</td>
<td>CARGO RD</td>
<td>2444.999086</td>
</tr>
<tr>
<td>480625.1979</td>
<td>4969181.768</td>
<td>24.76099366</td>
<td>CARGO RD</td>
<td>2444.999086</td>
</tr>
<tr>
<td>480610.183</td>
<td>4969179.259</td>
<td>20.16780883</td>
<td>CARGO RD</td>
<td>2444.999086</td>
</tr>
<tr>
<td>480596.3487</td>
<td>4969175.413</td>
<td>17.30193749</td>
<td>CARGO RD</td>
<td>2444.999086</td>
</tr>
<tr>
<td>480584.0986</td>
<td>4969173.006</td>
<td>13.87454209</td>
<td>CARGO RD</td>
<td>2444.999086</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Roadtype</th>
<th>Speedlimit(mph)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>35</td>
</tr>
<tr>
<td>RD</td>
<td>35</td>
</tr>
<tr>
<td>RD</td>
<td>35</td>
</tr>
<tr>
<td>RD</td>
<td>35</td>
</tr>
</tbody>
</table>

map. Moreover, we also need to add the personal information to GPS data in order to analyze the effect of the speeding statistically.

Figure 3.20 is a flow of data processing. First we remove unnecessary data (young driver and speed limit = 0 mph), and then we change the format that can be analyzed in statistical package \( R \).

3.3.1 Remove data (age is less than 16)

When we check the driver’s data which is TBI 2010 Household Interview Survey Data, some drivers are under 16. Therefore, we remove the trip data when the driver is under 16 years of age. The code of this algorithm is mentioned in Appendix 4.9.

The algorithm removed 128 trip data (8 individuals) for whom the driver’s age is under 16. Subsequently, the number of trip data decreases to 2,582 (152 individuals).
3.3.2 Remove speed limit 0 mph

After the map-matching process, we find out that some speed limits are 0 mph. In the analysis part, we will calculate how much the driver exceeds the speed limit (= driving speed / speed limit). However, the calculation doesn’t work when the speed limit is 0 (mph). Therefore, we remove the GPS data where the speed limit is 0 mph. The code of this algorithm is mentioned in [A.4.10]

No trip data are removed in this process. However, we remove 22 trip data apart from this process because no GPS data are recorded due to the map-matching process.

3.3.3 Create variables for statistical analysis

We create the format of dependent variable and independent variables for the statistical analysis. The code of this algorithm is mentioned in [A.4.11]

- Convert the unit of speed limit

  When we compare the driving speed with the speed limit, the unit should be matched. However, driving speed is [km/h] while speed limit is [mph], so we convert the unit of speed limit from [mph] to [km/h].
• Change the hour from UTC to MN time
  Time zone in GPS data is Coordinated Universal Time (UTC). That is why we change the timezone from UTC to MN time. It is noted that the time difference between UTC and Minnesota depends on the date because of daylight saving time [34].
  “-6h” during standard time (01/01/2011 - 03/12/2011), (11/07/2011 - 12/31/2011)
  “-5h” during daylight Saving time (03/13/2011 - 11/06/2011)

• Calculate the speeding
  We calculate the speeding behavior from two perspectives:
  1: Whether speeding or not (Percentage of speeding)
  When the driving speed is over the speed limit, the value becomes 1. If not, the value becomes 0.
  2: How much the driver exceeds the speed limit (Degree of speeding)
  The result of percentage of speeding is critical because “driving at 60 mph in 40 mph zone” and “driving at 41 mph in 40 mph zone” are identical outcomes. Therefore, we also analyze another speeding behavior by calculating how much each GPS data exceeds the speed limit. DegreeOfSpeeding is defined in Equation 3.1.

\[
DegreeOfSpeeding = \frac{DrivingSpeed}{SpeedLimit} \tag{3.1}
\]

When the value is 1, driving speed equals the speed limit, and when the value is more than 1, the data shows speeding.

• Create dummy variables and add personal information
  In order to compute the multi-variate regression analysis in R, we create dummy variables for speed limit zone and time. Moreover, we add personal information (age, gender, education) that is recorded in Household Survey data to GPS data by matching trip ID. These personal data are also changed to dummy variables. The detail of these variables is mentioned in Table 4.3.

Finally, the remaining 2,560 trips (from 152 individuals) are compiled to small sample data that can be analyzed in R. The code of this algorithm is mentioned in A.4.12.
Chapter 4

Analysis

4.1 Hypotheses

We hypothesize that speeding is affected by road type (hierarchy of the network) and road characteristics.

1. Hierarchy (Road type) – *Hypothesis: Position near the top of the hierarchy of roads is correlated with speeding*

   We analyze the effect of hierarchy in the network. Compared with local roads, the driver is less affected by external factors on freeways. GIS data contains speed limit, street name and road type, and therefore we can calculate speeding depending on each road type.

2. Link length – *Hypothesis: Long link length is correlated with speeding*

   We investigate the effect of link length on speeding. When the length is small, there are many intersections in the network. The GIS network has link length data, so it is possible to calculate the relationship between link length and speeding behavior.
Table 4.1: Total data in each speed limit zone

<table>
<thead>
<tr>
<th>Speed limit (mph)</th>
<th>GPS data</th>
<th>GIS link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of data</td>
<td>percentage</td>
</tr>
<tr>
<td>5</td>
<td>759</td>
<td>0.1%</td>
</tr>
<tr>
<td>10</td>
<td>2,364</td>
<td>0.2%</td>
</tr>
<tr>
<td>20</td>
<td>80</td>
<td>0.0%</td>
</tr>
<tr>
<td>25</td>
<td>20,867</td>
<td>1.8%</td>
</tr>
<tr>
<td>30</td>
<td>133,339</td>
<td>11.4%</td>
</tr>
<tr>
<td>35</td>
<td>377,960</td>
<td>32.2%</td>
</tr>
<tr>
<td>40</td>
<td>346,876</td>
<td>29.6%</td>
</tr>
<tr>
<td>45</td>
<td>22,624</td>
<td>1.9%</td>
</tr>
<tr>
<td>50</td>
<td>20,001</td>
<td>1.7%</td>
</tr>
<tr>
<td>55</td>
<td>118,160</td>
<td>10.1%</td>
</tr>
<tr>
<td>60</td>
<td>91,164</td>
<td>7.8%</td>
</tr>
<tr>
<td>65</td>
<td>28,416</td>
<td>2.4%</td>
</tr>
<tr>
<td>70</td>
<td>9,493</td>
<td>0.8%</td>
</tr>
<tr>
<td>Total</td>
<td>1,172,103</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

4.2 Analysis Result

For 2,560 trips, we analyze the relationship between road network variables and speeding. The total amount of data are 1,172,103 GPS points. As Table 4.1 illustrates, most of the data are concentrated on 30, 35, 40 and 55 mph zone. Moreover, relatively few points are in low speed limit zones (less than 20 mph) and high speed limit zones (70 mph). (Note some travel that is retained in the analysis is on private roads or in parking lots, this explains travel on roads with 5 mph speed limits).

4.2.1 Road network structure

Hierarchy (Road type)

The bar chart shows the percentage of speeding on each speed limit zone (Figure 4.1). Overall, 23.3 percent of GPS observations exceed the speed limit. It can also be seen that speeding behavior is significant in low speed limit zones (e.g. from 5 to 25 mph) and high speed limit zones (e.g. from 55 to 65 mph). The number of individuals (N = *) suggests that most participants encountered speed limits between 25 and 65 mph while driving.
The 20 mph zone stand at 88.8%, and it is followed by 5 and 25 mph zone at 80.9%, 73.3% respectively. However, the amount of data in low speed limit zones is relatively small, therefore more data are required in order to improve the reliability of this result. In addition to low speed limit zones, speeding percentage in high speed limit zones (from 45 to 70 mph) is also more than average. On the other hand, speed limits from 30 to 40 mph, have a lower speeding percentage than lower and higher speed limit zones. Especially, 30 mph is the lowest number at 10.1%. The code of computing and counting number in \textit{R} is mentioned in A.4.19.

\section*{Figure 4.1: Percentage of speeding across speed limit zones}

(N = *) indicates number of individuals persons encountering that particular speed limit.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{speeding_distribution.png}
\caption{Percentage of speeding across speed limit zones}
\end{figure}

The provided box plot graph illustrates how much the driver exceeds the speed limit across each speed limit zone (Figure 4.2). When the speed limit is low (5, 20, and 25 mph zone), the driver exceeds the speed limit greatly more than other speed limit zones. The value of first quartile is about more than 1.0, therefore about 75% of data indicate the speeding in this speed limit zone. On the other hand, in intermediate and high speed limit zones (From 30 mph to 70 mph), speeding is not significant because many data are close to or less than 1.0. The code of creating boxplot in \textit{R} is mentioned in A.4.20.

When we check the median in box plot, it looks like a “V-shape” across speed limit zones. From 5 mph to 30 mph, the value of median are likely to increase as the
speed limit becomes small. In contrast, from 30 mph to 70 mph the median slightly rises as the speed limit increases. This result suggests that many driver exceed the speed limit significantly in low speed limit zones; on the other hand, they tend to drive proportionally nearer the speed limit in high speed limit zones. For intermediate speed limit zones, they are likely to drive below the speed limit.

Compared the result between the percentage of speeding (Figure 4.1) and the degree of speeding (Figure 4.2), we can see a similar trend although there are small differences.

![Figure 4.2: Degree of speeding across speed limit zones](image)

43
Link length

Figure 4.3 illustrates the relationship between link length and speeding behavior. Y axis is Degree of Speeding defined in Equation 3.1.

We get the information of link length from TLG network, and generally link length indicates the length between intersections. However, some links are divided into several parts between intersections especially in interstate and freeway zones. Exact link length on highways, which means the length between interchange, is much longer; however, highway link data is divided into several sections between interchanges. Therefore, we should be careful about the result of link length in high speed limit zones.

Most link lengths are less than 1,000m and some data in long link length might be too far away from the fitted regression line because of an outlier problem (Figure A.12).
To reduce data issues, we analyze the data for link length ≤ 1,000m.

Figure 4.3 and Table 4.2 shows that when link length is longer, the driver is more likely to exceed the speed limit. According to the regression results (Equation 1: Table 4.2), t-value is large and the p-value of overall F-test is very small. Therefore it appears that this model fits the data well; however, the value of $R^2$ is relatively small (0.103). Next, we compute the polynomial model (Equation 4.2) which fits the data better.

In the polynomial model, the value of $E(speeding|streetlength)$ reaches a maximum when street length is 761.2m. It implies that speeding is likely to decrease as the link length becomes longer than 761.2m. The code of creating graph and regression result in R is mentioned in A.4.21.

**Equation 1**

$$E(speeding|streetlength) = \beta_0 + \beta_1 streetlength$$  \hspace{1cm} (4.1)

**Equation 2**

$$E(speeding|streetlength) = \beta_0 + \beta_1 streetlength + \beta_2 streetlength^2$$  \hspace{1cm} (4.2)

<table>
<thead>
<tr>
<th>Coefficients:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent variable</strong> : Speeding (link length is less than 1,000m)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Equation 1</th>
<th></th>
<th>Equation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate</td>
<td>t-value</td>
<td>Estimate</td>
<td>t-value</td>
</tr>
<tr>
<td>Intercept</td>
<td>6.078E-01</td>
<td>1197.5</td>
<td>***</td>
</tr>
<tr>
<td>***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street length</td>
<td>4.881E-04</td>
<td>358.2</td>
<td>***</td>
</tr>
<tr>
<td>***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street length$^2$</td>
<td>-</td>
<td>-</td>
<td>-7.501E-07</td>
</tr>
</tbody>
</table>

signif. Codes: 0 '****' 0.001, '**' 0.01, '*' 0.05, '.' 0.1 ' ' 1

<table>
<thead>
<tr>
<th></th>
<th>Equation 1</th>
<th></th>
<th>Equation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple R-squared</td>
<td>0.103</td>
<td></td>
<td>0.118</td>
</tr>
<tr>
<td>F-statistic</td>
<td>1.28e+05</td>
<td></td>
<td>7.41e+04</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;2.2e-16</td>
<td></td>
<td>&lt;2.2e-16</td>
</tr>
</tbody>
</table>

The result of Figure 4.3 is from all roads, but driving condition depends on road type, therefore, we analyze this relationship according to each speed limit zone. Figure 4.4

45
illustrates that the relationship between link length and speeding on the speed limit zone. An interesting finding is that the graph shows there is little correlation between link length and speeding at 30 mph zones. On the other hand, there is positive correlation at other speed limit zones.
Figure 4.4: Speeding by link length depending on speed limit zones

(Top left: 25mph, Top right: 30mph, Bottom left: 35mph, Bottom right: 40mph)
4.2.2 Time of day

The bar chart illustrates the proportion exceeding the speed limit across the time of day (Figure 4.5). Overall, speeding behavior varies by time of day and the percentage of speeding during daytime driving is lower than nighttime driving, with a peak at 4am of 64.7%, and it is followed by 5am and 11pm (hour 23) at 40.1% and 39.8% respectively. On the other hand, 8am and 9am are smaller than other morning hours, it appears that this result is related to the effect of rush hour. It is assumed that drivers cannot drive faster because of traffic conditions. Although the percentage at 3am is considerably lower than other time zones (3.7%), the number of participants are very small (N=3) and it appears that this result might be significantly influenced by personal factors.

![Figure 4.5: Percentage of speeding by time of day](image)

The provided box plot graph illustrates how much the driver exceeds the speed limit across each time (Figure 4.6). Midnight shows a high degree of speeding compared with other times; however, it seems that there is little difference. In order to check this difference statistically, we calculate the t-test for the means of two time.

The Null hypothesis is as follows:

\[ \mu_i = \mu_j \ (i \neq j) \] (4.3)

Firstly we measure whether the mean value of 4am differs from 7am (\(\mu_4 = \mu_7\)). The
average value at 4am is 0.947 while 7am is 0.784 (Table A.7). According to the t-test, t-value: 25.300 and p-value: 0. Therefore, we reject the null hypothesis and it indicates that there is a significant difference. Secondly, we compute t-values at 12pm and 2pm ($\mu_{12} = \mu_{14}$). Average value at 12pm is 0.754 while 14am is 0.756. According to the t-test, t-value: -0.808, p-value: 0.419. Therefore, we cannot reject the null hypothesis. The result of significant code between all time is shown in Figure A.13. Although the difference of mean value is not significant between some time, mostly there is a significant difference between each time. The code of this algorithm is in A.4.13.

![Figure 4.6: Degree of speeding across time](image)

### 4.2.3 Personal information

Several researchers mention that speeding behavior depends on the driver [1][3]. Unlike loop detector data, GPS data is linked to personal information. We choose gender, age, and education as personal data to analyze the data of 152 participants.
Gender

Figure 4.7 shows speeding is undifferentiated by gender. While the number of participants is large (male drivers: 65, female drivers: 87), the percentage of speeding for male drivers is same as female drivers (Male, Female: 23.3%).

The given box plot compares the degree of speeding by gender (Figure 4.8). It appears that there is little difference between gender because average value is almost similar; Male: 0.754, Female: 0.775 (Table A.8). For checking this hypothesis, we measure whether the average value differs significantly across samples. The null hypothesis is as follows:

$$\mu_{\text{male}} = \mu_{\text{female}}$$  \hspace{1cm} (4.4)

According to the t-test, t-value: 33.660, p-value: 0. Therefore, we reject the null hypothesis and there is a difference between genders from statistical results. The code of this algorithm is mentioned in A.4.14.
Figure 4.8: Degree of speeding by gender
Age

Figure 4.9 presents the result of the percentage of speeding across age. The age group from 25 to 34 illustrates the highest speeding behavior at 34.9%. Although the number of participants are small, elderly drivers (85 and over) are also likely to exceed the speed limit (33.2%). It is followed by 55 to 64 and 35 to 44 at 25.0% and 24.0%, respectively. On the other hand, the age group who are 75 to 84 shows the lowest speeding at 13.5%.

![Figure 4.9: Percentage of speeding by age](image)

Figure 4.10 illustrates the difference of degree of speeding across age groups. It appears that the degree of speeding is not different depending on the age group; however, all t-value is large according to the t-test (Equation 4.3). Therefore, we reject the null hypothesis and there is a significant difference between each age group (Figure A.14). The code of this algorithm is in A.4.15.
Figure 4.10: Degree of speeding across age
(5: 18 to 24, 6: 25 to 34, 7: 35 to 44, 8: 45 to 54, 9: 55 to 64, 10: 65 to 74, 11: 75 to 84, 12: 85 and over)
Education

Figure 4.11 illustrates the result of education, and education level is divided into 3 groups; education_0 (High school graduate and Vocational/Technical training), education_1 (Some college and Associates degree) and education_2 (Bachelors degree and Graduate/Post-graduate degree). ‘education_0’ means lower education level while ‘education_2’ means higher education level. According to Figure 4.11 we cannot see a significant difference between education levels because education_0 (23.1%), education_1 (24.0%), and education_2 (23.1%).

Figure 4.11: Speeding by education level

Figure 4.12: Degree of speeding across education
Figure 4.12 illustrates the difference of degree of speeding across education level. It seems that the mean value of each education status is similar; however, all t-value are large according to the t-test (Equation 4.3). Therefore, we reject the null hypothesis and there is a significant difference between each education level (Figure A.15). The code of this algorithm is in A.4.16.

**Speeding difference depends on driver**

Figure 4.13 illustrates the difference of speeding across drivers, and the result shows that there are significant differences depending on the driver. Average is 22.5% and maximum number is 66.5%; on the other hand, minimum number is 0.0%. This graph looks like an inverse square law. The code of this algorithm is in A.4.17.

![Figure 4.13: Percentage of speeding per individuals](image)
4.2.4 Statistical result

Next, we analyze the relationship between several variables (road network structure and personal information) and the speeding statistically. Table 4.3 shows the list of dependent variable and independent variables.

**Dependent variable**
Degree of speeding [Equation 3.1]

**Independent variables**
We change the data into binary (0/1) variable except for “Street length”. Moreover, in order to simplify the result, we categorize time into 4 parts; 1. morning (06:00 to 12:00), 2. afternoon (12:00 to 18:00), 3. night (18:00 to 24:00), 4. midnight (24:00 to 06:00). Education is also categorized into 3 parts; 1. education_0 (High school graduate and Vocational/Technical training), 2. education_1 (Some college and Associates degree) and 3. education_2 (Bachelors degree and Graduate/Post-graduate degree).

In order to avoid the dummy variable trap, we drop one category from the model.

- Speed limit: If all indicator is 0, speed limit zone is 70
- Age: If all indicator is 0, age is 18 to 24
- Education: If all indicator is 0, education is High school graduate or Vocational/Technical training [education_0]
- Time: If all indicator is 0, time is 24:00 to 6:00 [midnight]

Table 4.4 showing regression results is calculated using the statistical package R. The code of regression analysis in R is in A.4.22 While t-value of [age_75_to_84] and [morning] are small, t-value of other variables are large, therefore they are statistically significant. Overall F-test shows that F value is 1.496e+04 and p-value is less than 2.2e-16. Therefore, this model shows high significance. The findings from earlier are corroborated in this multivariate analysis.

Coefficient value indicates that when the street length is long, speeding is more likely to occur. Also, 30 and 35 mph zones are negatively associated with speeding. Persons who are young (18 to 24), male, drive in the afternoon, and/or are educated speed less. Overall the $R^2$ is 0.23.
Table 4.3: List of dependent and independent variables

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th>Speeding</th>
<th>Speed data / speed limit zone</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Independent variables</strong></td>
<td><strong>Variables</strong></td>
<td><strong>Definition</strong></td>
</tr>
<tr>
<td>Street length</td>
<td>Link length</td>
<td></td>
</tr>
<tr>
<td>speed_25</td>
<td>Indicator, 1= speed limit zone is less than 25 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_30</td>
<td>Indicator, 1= speed limit zone is 30 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_35</td>
<td>Indicator, 1= speed limit zone is 35 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_40</td>
<td>Indicator, 1= speed limit zone is 40 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_45</td>
<td>Indicator, 1= speed limit zone is 45 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_50</td>
<td>Indicator, 1= speed limit zone is 50 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_55</td>
<td>Indicator, 1= speed limit zone is 55 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_60</td>
<td>Indicator, 1= speed limit zone is 60 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>speed_65</td>
<td>Indicator, 1= speed limit zone is 65 mph, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>age_25_to_34</td>
<td>Indicator, 1= age is 25 to 34, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>age_35_to_44</td>
<td>Indicator, 1= age is 35 to 44, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>age_45_to_54</td>
<td>Indicator, 1= age is 45 to 54, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>age_55_to_64</td>
<td>Indicator, 1= age is 55 to 64, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>age_65_to_74</td>
<td>Indicator, 1= age is 65 to 74, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>age_75_to_84</td>
<td>Indicator, 1= age is 75 to 84, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>age_85_and_over</td>
<td>Indicator, 1= age is more than 85, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>male</td>
<td>Indicator, 1= gender is male, 0 = female</td>
<td></td>
</tr>
<tr>
<td>education_1</td>
<td>Indicator, 1= education level is Some college or Associates degree, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>education_2</td>
<td>Indicator, 1= education level is Bachelors degree or Graduate/Post-graduate degree, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>morning</td>
<td>Indicator, 1= time is 06:00 to 12:00, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>afternoon</td>
<td>Indicator, 1= time is 12:00 to 18:00, 0 = otherwise</td>
<td></td>
</tr>
<tr>
<td>evening</td>
<td>Indicator, 1= time is 18:00 to 24:00, 0 = otherwise</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4.4: Output of statistical analysis in R

**Coefficients:**

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Estimate</th>
<th>t-value</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>6.878e-01</td>
<td>155.171</td>
<td>***</td>
</tr>
<tr>
<td>streetlength</td>
<td>1.650e-04</td>
<td>173.801</td>
<td>***</td>
</tr>
<tr>
<td>speed_25</td>
<td>6.365e-01</td>
<td>173.934</td>
<td>***</td>
</tr>
<tr>
<td>speed_30</td>
<td>-9.581e-02</td>
<td>-29.015</td>
<td>***</td>
</tr>
<tr>
<td>speed_35</td>
<td>-1.035e-01</td>
<td>-32.139</td>
<td>***</td>
</tr>
<tr>
<td>speed_40</td>
<td>3.614e-02</td>
<td>17.665</td>
<td>***</td>
</tr>
<tr>
<td>speed_45</td>
<td>6.506e-02</td>
<td>17.665</td>
<td>***</td>
</tr>
<tr>
<td>speed_50</td>
<td>4.953e-02</td>
<td>13.265</td>
<td>***</td>
</tr>
<tr>
<td>speed_55</td>
<td>1.226e-01</td>
<td>38.317</td>
<td>***</td>
</tr>
<tr>
<td>speed_60</td>
<td>1.166e-01</td>
<td>36.527</td>
<td>***</td>
</tr>
<tr>
<td>speed_65</td>
<td>7.844e-02</td>
<td>22.555</td>
<td>***</td>
</tr>
<tr>
<td>age_25_to_34</td>
<td>5.604e-02</td>
<td>23.292</td>
<td>***</td>
</tr>
<tr>
<td>age_35_to_44</td>
<td>1.221e-02</td>
<td>5.462</td>
<td>***</td>
</tr>
<tr>
<td>age_45_to_54</td>
<td>3.321e-02</td>
<td>14.976</td>
<td>***</td>
</tr>
<tr>
<td>age_55_to_64</td>
<td>4.196e-02</td>
<td>19.246</td>
<td>***</td>
</tr>
<tr>
<td>age_65_to_74</td>
<td>3.868e-02</td>
<td>17.403</td>
<td>***</td>
</tr>
<tr>
<td>age_75_to_84</td>
<td>3.389e-03</td>
<td>1.420</td>
<td></td>
</tr>
<tr>
<td>age_85_and_over</td>
<td>8.744e-02</td>
<td>30.881</td>
<td>***</td>
</tr>
<tr>
<td>male</td>
<td>-1.341e-02</td>
<td>-23.704</td>
<td>***</td>
</tr>
<tr>
<td>education_1</td>
<td>-4.808e-03</td>
<td>-4.906</td>
<td>***</td>
</tr>
<tr>
<td>education_2</td>
<td>-1.705e-02</td>
<td>-21.314</td>
<td>***</td>
</tr>
<tr>
<td>morning</td>
<td>4.021e-03</td>
<td>1.796</td>
<td></td>
</tr>
<tr>
<td>afternoon</td>
<td>-1.322e-02</td>
<td>-5.934</td>
<td>***</td>
</tr>
<tr>
<td>night</td>
<td>1.595e-02</td>
<td>6.972</td>
<td>***</td>
</tr>
</tbody>
</table>

Signif. Codes: 0 ‘***’ 0.001, ‘**’ 0.01 ‘*’ 0.05 , ’ 1

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple R-squared</td>
<td>0.2269</td>
</tr>
<tr>
<td>F-statistic</td>
<td>1.496e+04</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;2.2e-16</td>
</tr>
</tbody>
</table>

58
4.2.5 Fixed effects estimation with integrated GPS points

While analytical results of this paper are from all GPS points, we also analyze the results from integrated GPS points. GPS points were recorded every second; however, it is considered that speeding patterns of drivers in the same link are similar. Therefore, we integrate multiple GPS points from one link into one analysis record; however, we separate the link data depending on drivers and trips because driving environments and driving patterns are different. The total amount of GPS points is 115,536 (152 drivers and 24,262 link). The code of this algorithm is in A.4.18.

Moreover, in order to eliminate the fixed effect prior to estimation, we use the dummy variable regression. We put in a dummy variable for each individual along with the independent variables. While the t-value of [speed.40], [speed.45], [speed.50], [age.55_to.64], [age.85_and_over] and [male] are small, the t-value of other variables are large. Overall F-test shows that F value is 280.5 and p-value is less than 2.2e-16. Therefore, this model shows high significance. Overall the $R^2$ is 0.29.

A positive coefficient value indicates that when the street length is long, speeding is likely to occur. Also, 30 and 35 mph zones are negatively associated with speeding. Persons who are young, drive at midnight (24:00 to 6:00), and/or are educated speed more (Table 4.5). Compared with the result of Table 4.4, the result of road network structure is similar, but the result of personal information is different. The code of regression result in $R$ is mentioned in A.4.23.

Figure 4.14 illustrates the estimate of coefficients across drivers, and the result shows that there are significant differences depending on the driver. 62.4% of individuals are negative and some individuals’ estimates are significantly large.
Table 4.5: Output of the dummy variable regression in R

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>t-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>8.386e-01</td>
<td>35.082</td>
</tr>
<tr>
<td>streetlength</td>
<td>1.280e-04</td>
<td>34.736</td>
</tr>
<tr>
<td>speed_25</td>
<td>6.587e-01</td>
<td>53.206</td>
</tr>
<tr>
<td>speed_30</td>
<td>-3.703e-02</td>
<td>-3.125</td>
</tr>
<tr>
<td>speed_35</td>
<td>-1.177e-01</td>
<td>-10.162</td>
</tr>
<tr>
<td>speed_40</td>
<td>-8.413e-03</td>
<td>-0.727</td>
</tr>
<tr>
<td>speed_45</td>
<td>-3.285e-03</td>
<td>-0.255</td>
</tr>
<tr>
<td>speed_50</td>
<td>8.432e-03</td>
<td>0.636</td>
</tr>
<tr>
<td>speed_55</td>
<td>1.056e-01</td>
<td>9.164</td>
</tr>
<tr>
<td>speed_60</td>
<td>9.919e-02</td>
<td>8.538</td>
</tr>
<tr>
<td>speed_65</td>
<td>3.506e-02</td>
<td>2.823</td>
</tr>
<tr>
<td>age_25_to_34</td>
<td>7.488e-02</td>
<td>4.963</td>
</tr>
<tr>
<td>age_35_to_44</td>
<td>-1.055e-01</td>
<td>-3.933</td>
</tr>
<tr>
<td>age_45_to_54</td>
<td>-2.184e-01</td>
<td>-8.490</td>
</tr>
<tr>
<td>age_55_to_64</td>
<td>-3.575e-02</td>
<td>-1.513</td>
</tr>
<tr>
<td>age_65_to_74</td>
<td>-1.572e-01</td>
<td>-7.345</td>
</tr>
<tr>
<td>age_75_to_84</td>
<td>-5.696e-02</td>
<td>-3.927</td>
</tr>
<tr>
<td>age_85_and_over</td>
<td>-3.847e-02</td>
<td>-1.474</td>
</tr>
<tr>
<td>male</td>
<td>-1.026e-02</td>
<td>-0.580</td>
</tr>
<tr>
<td>education_1</td>
<td>1.520e-01</td>
<td>9.020</td>
</tr>
<tr>
<td>education_2</td>
<td>1.347e-01</td>
<td>4.676</td>
</tr>
<tr>
<td>morning</td>
<td>-3.225e-02</td>
<td>-4.726</td>
</tr>
<tr>
<td>afternoon</td>
<td>-3.566e-02</td>
<td>-5.254</td>
</tr>
<tr>
<td>night</td>
<td>-2.295e-02</td>
<td>-3.285</td>
</tr>
<tr>
<td>person1</td>
<td>8.224e-02</td>
<td>5.936</td>
</tr>
<tr>
<td>person2</td>
<td>-3.585e-01</td>
<td>-8.735</td>
</tr>
<tr>
<td>— person 151</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Signif. codes: .0 *** 0.001 ** 0.01 * 0.05 . 0.1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple R-squared</td>
<td>0.2851</td>
</tr>
<tr>
<td>F-statistic</td>
<td>280.5</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;2.2e-16</td>
</tr>
</tbody>
</table>
Figure 4.14: Estimate of coefficients per individuals
Chapter 5

Conclusion and Discussion

5.1 Conclusion

This thesis investigates GPS-based speed data from 152 participants in Minnesota to examine the relationship between road network structure and speeding. The most pertinent findings from the results are that speeding behavior is significant in both low and high speed limit zones, and long link length is correlated with speeding. In terms of degree of speeding, the t-test indicates that there are significant differences across gender, age, and education level. Moreover, our algorithm and process shows sufficient accuracy of map-matching with very few false positives, although half of GPS data are removed due to the process. We analyzed 1,172,103 GPS points.

Unlike the speed data of a loop detector, GPS-based analysis enables us to investigate the speed information over a broad area. Multi-variate regression analysis finds persons who are young (18 to 24), male, drive in the afternoon, and/or are educated speed less than other age groups, female drivers, other times of the day, and/or less educated drivers. Future research will aim to reduce the amount of excluded data (false negatives).

5.2 Discussion

- Comparison with other studies
  Speeding behavior across speed limit zones has been previously analyzed [3], and according to both that result and our result, speeding illustrates a ‘V-shape’ across
all speed limit zones. In the previous study, the 60 mph zone had the lowest
speeding behavior \cite{3} while our result shows the 30 mph zone is the lowest. It is
assumed that the difference in road characteristics might affect the result.

Other studies mention that elderly, female and uneducated drivers who don’t often
drive on interurban roads are likely to follow the speed limit \cite{1, 3}. In our study,
young (18 to 24), male, drive in the afternoon, and/or are educated drivers showed
less speeding behavior than comparison drivers. To confirm our finding, we need
to see this difference in detail in further studies.

- Over-fitting problem
  We perform statistical analysis from several independent variables. But it is con-
sidered the model might be affected by over-fitting problems \cite{Table 4.4}. When
the independent variables are only speed limit zones \cite{Table A.1}, the \( R^2 \) is 0.20
and it is close to the value of \cite{Table 4.4} (the \( R^2 \) is 0.23). On the other hand, \( R^2 \)
is small when other factors are independent variables \cite{Table A.2 to Table A.6}.
Therefore, the speed limit zone has a more significant impact on speeding com-
pared with other factors.

- Data
  Because of the limited data, we cannot analyze the relationship between speeding
behavior and some road characteristics like road shape (curvature) and lane width.
Also we cannot analyze the effect of vehicle size.

If we add this information to GPS data, we would analyze more interesting effects
of speeding. For example, if we add crash data into GPS data, we can analyze
whether speeding is related to crashes.

We analyze 152 drivers; however, the amount of data differs depending on the
driver. As \cite{Figure 4.13} shows, speed limit compliance depends on the driver.

- Algorithm and process
  We investigate GPS data of each participants for 7 days; however, 51.0\% of GPS
data are removed in our algorithm in order to secure the accuracy of the map-
matching. Many GPS points near intersections and interstates are deleted due to
our algorithm.
If drivers show significant speeding behavior in the excluded GPS data, analysis results are likely to be biased. In order to avoid this bias, we need to decrease the amount of excluded GPS data. For example, an algorithm that increases the accuracy of map-matching near intersections and interstates has been proposed. This algorithm considers neighboring GPS points for improving the accuracy of map-matching near the intersection [15]. The algorithm should consider the average speed of GPS data by distinguishing between nearby parallel routes.

Some actual driving links are also deleted due to the process and it makes it difficult to find the whole trip data. A solution is that if the street name and speed limit in front and behind the missing link is identical, then data of missing link is also identical to adjacent link (Figure 5.1). Further use of upstream and downstream points to identify the most likely link for a point in overlapping layers is also a promising direction.

![Figure 5.1: Image of complementing missing link](image)

If we keep both whole driving GPS data and satisfactory accuracy, we would be able to analyze the effect of other road network structures:

1. Discontinuity of speed limit zones – Hypothesis: Large discontinuity is correlated with speeding

   This network variable means the effect of discontinuity (speed limit distribution) during the trip. When the speed limit is uniform during the trip, it is predicted that the amount of acceleration and deceleration is small. This situation would create stable driving during the trip.
2. Geographical feature (Curve or straight?) – *Hypothesis: Straight roads are correlated with speeding*

The driver is significantly affected by geographical features (network shape). When the driver drives on a curved section and changes direction, it would affect speeding. GPS data has course (degrees) data, so we can calculate the amount of degree change.

- Differences of speeding across speed limit zones

  Figure 4.1 and Figure 4.2 illustrate that speeding behavior is significant in low speed limit areas more than high speed limit areas. However, traffic enforcement against speeding isn’t usually conducted in low speed limit zones. Speed traps or speed cameras are generally installed in high speed limit zones (to be clear, speed cameras are not used in Minnesota). Speeding behavior on local roads is obviously dangerous for the pedestrian and it risks vehicle-to-pedestrian collisions. On the other hand, speeding behavior is relatively small near 30 and 35mph zones. These 35mph zones are most prevalent in Minneapolis and the road geometry and traffic condition in this zone fits the driver’s preferred speed for the physical conditions by the road.
References


Appendix A

Glossary and Acronyms

A.1 Formula

A.1.1 Convert coordinate system

Convert the unit of Longitude and Latitude from degree to radian

\[
\lambda = \text{Longitude}[\text{radian}] = \text{Longitude}[\text{degree}] \times \pi/180 \quad (A.1)
\]

\[
\varphi = \text{Latitude}[\text{radian}] = \text{Latitude}[\text{degree}] \times \pi/180 \quad (A.2)
\]

f means flattening factor of the earth in WGS84 [35]

\[
f = \frac{1}{298.257223563} \quad (A.3)
\]

n means third flattening

\[
n = \frac{f}{2-f} \quad (A.4)
\]

a means semi-major axis (Unit: m) [35]

\[
a = 6,378,137 \quad (A.5)
\]

Following formula is from [30]

\[
A = \frac{a}{1 + n} (1 + \frac{n^2}{4} + \frac{n^4}{64}) \quad (A.6)
\]
\[ \alpha_1 = \frac{1}{2}n - \frac{2}{3}n^2 + \frac{5}{16}n^3 \]  
(A.7)

\[ \alpha_2 = \frac{13}{48}n^2 - \frac{3}{5}n^3 \]  
(A.8)

\[ \alpha_3 = \frac{61}{240}n^3 \]  
(A.9)

Following formula is from [31]

\[ \tan \chi = \sinh(\tanh^{-1}\sin \varphi - \frac{2\sqrt{n}}{1 + n} \tanh^{-1}\left(\frac{2\sqrt{n}}{1 + n} \sin \varphi\right)) \]  
(A.10)

UTM zone in Minnesota is 15 [36]

\[ \lambda'[degree] = 15*6 - 183 \]  
(A.11)

Convert the unit of Longitude and Latitude from degree to radian

\[ \lambda[radian] = \lambda'[radian] \times \pi/180 \]  
(A.12)

Following formula is from [31]

\[ \xi = \tanh^{-1}\left(\frac{\tan \chi}{\cos(\lambda - \lambda_0)}\right) \]  
(A.13)

\[ \eta = \tanh^{-1}\left(\frac{\sin(\lambda - \lambda_0)}{\sqrt{1 + \tan \chi^2}}\right) \]  
(A.14)

\( E_0 \) means false easting (Unit: m) [36]

\[ E_0 = 500,000 \]  
(A.15)

\( m_0 \) means scale factor [36]

\[ m_0 = 0.9996 \]  
(A.16)

Plane rectangular coordinates easting \( X \) and northing \( Y \) (Coordinate system: UTM) [31]

\[ X = X_0 + m_0A(\eta + \sum_{k=1}^{3} \alpha_k \cos(2k\xi) \sinh(2k\eta)) \]  
(A.17)

\[ Y = m_0A(\xi + \sum_{k=1}^{3} \alpha_k \sin(2k\xi) \cosh(2k\eta)) \]  
(A.18)
A.2 Process

A.2.1 Create buffer of GIS links

It is noted that we don’t have the algorithm about this process.

1-1. Put GIS map on QGIS
   - Drag .shp file to QGIS screen (Figure A.1)

1-2. Create a buffer

- Choose a “Buffer(s)” function
- Create 10m buffer(s) (Figure A.2)
- Input “10” as ‘Buffer distance’. It is noted that the unit of buffer distance is meters.
- We choose “Input vector layer” as GIS map data. (Figure A.3)
- Result of output (10m buffer)

As Figure A.5 shows, GIS links change from line to polygon.
Figure A.2: Buffer process 1
Figure A.3: Buffer process 2
Figure A.4: GIS links

Figure A.5: GIS buffer(s) (10m)
A.2.2 Extract GIS links that are close to GPS data

2.1. Load a GPS Layer from a Delimited Text File

Layer -> Add Delimited Text Layer [Figure A.6]

Input the GPS file and set X field and Y field as the location data of UTM coordinates. [Figure A.7]

File format: CSV file

Geometry definition: Point coordinates

X field: X [Equation A.17], Y field: Y [Equation A.18] [X and Y are UTM coordinates]

2.2. Change coordinate system

Choose a GPS layer and click properties button

General -> Coordinate reference system

Press “Specify” button

Coordinate Reference System: “NAD83 / UTM zone 15N”, Authority ID: “ESPG: 26915” [Figure A.8]

- Result of output

[Figure A.9] is the result of the process
Figure A.7: Create a Layer from a Delimited Text File
Figure A.8: Coordinate Reference System Selector
Figure A.9: Result of output
(Blue lines: 10m GIS buffer, Green dots: GPS data)
2.3. Extract GIS links
- Choose a “Select by location” function
Vector -> Research tools -> Select by location  
Select “Select features in:” as GIS buffer data and “that intersect features in:” as GPS data.  

![Figure A.10: Select by location 1](image1)

![Figure A.11: Select by location 2](image2)
A.3 Result

A.3.1 Output of speeding analysis in R

![Figure A.12: Relationship between link length and speeding](image)

Figure A.12: Relationship between link length and speeding
### A.3.2 Output of statistical analysis in $R$

Table A.1: Independent variable: Speed limit zones

<table>
<thead>
<tr>
<th>Coefficients:</th>
<th>Estimate</th>
<th>t-value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.939</td>
<td>310.909</td>
<td>***</td>
</tr>
<tr>
<td>speed_25</td>
<td>0.460</td>
<td>128.836</td>
<td>***</td>
</tr>
<tr>
<td>speed_30</td>
<td>-0.307</td>
<td>-98.108</td>
<td>***</td>
</tr>
<tr>
<td>speed_35</td>
<td>-0.302</td>
<td>-98.792</td>
<td>***</td>
</tr>
<tr>
<td>speed_40</td>
<td>-0.154</td>
<td>-50.316</td>
<td>***</td>
</tr>
<tr>
<td>speed_45</td>
<td>-0.109</td>
<td>-30.307</td>
<td>***</td>
</tr>
<tr>
<td>speed_50</td>
<td>-0.104</td>
<td>-28.253</td>
<td>***</td>
</tr>
<tr>
<td>speed_55</td>
<td>-0.013</td>
<td>-4.137</td>
<td>***</td>
</tr>
<tr>
<td>speed_60</td>
<td>0.008</td>
<td>2.552</td>
<td>*</td>
</tr>
<tr>
<td>speed_65</td>
<td>-0.007</td>
<td>-1.912</td>
<td>.</td>
</tr>
</tbody>
</table>

---

signif. Codes: 0 '****' 0.001 '**', 0.01 '*' 0.05 ',' 0.1 ' ' 1

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple R-squared</td>
<td></td>
<td>0.2021</td>
<td></td>
</tr>
<tr>
<td>F-statistic</td>
<td></td>
<td>3.299e+04</td>
<td></td>
</tr>
<tr>
<td>p-value</td>
<td></td>
<td>&lt;2.2e-16</td>
<td></td>
</tr>
</tbody>
</table>
Table A.2: Independent variable: Link length

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>t-value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.6679e-01</td>
<td>1604.8</td>
<td>***</td>
</tr>
<tr>
<td>streetlength</td>
<td>2.716e-04</td>
<td>330.7</td>
<td>***</td>
</tr>
</tbody>
</table>

signif. Codes: 0 ‘***’ 0.001, ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared | 0.0854
F-statistic        | 1.094e+05
p-value            | <2.2e-16

Table A.3: Independent variable: Age group

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>t-value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.703</td>
<td>308.161</td>
<td>***</td>
</tr>
<tr>
<td>age_25_to_34</td>
<td>0.117</td>
<td>45.662</td>
<td>***</td>
</tr>
<tr>
<td>age_35_to_44</td>
<td>0.039</td>
<td>16.219</td>
<td>***</td>
</tr>
<tr>
<td>age_45_to_54</td>
<td>0.050</td>
<td>20.862</td>
<td>***</td>
</tr>
<tr>
<td>age_55_to_64</td>
<td>0.083</td>
<td>35.254</td>
<td>***</td>
</tr>
<tr>
<td>age_65_to_74</td>
<td>0.061</td>
<td>25.509</td>
<td>***</td>
</tr>
<tr>
<td>age_75_to_84</td>
<td>0.013</td>
<td>5.093</td>
<td>***</td>
</tr>
<tr>
<td>age_85_and_over</td>
<td>0.139</td>
<td>44.868</td>
<td>***</td>
</tr>
</tbody>
</table>

signif. Codes: 0 ‘***’ 0.001, ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared | 0.00717
F-statistic        | 1210
p-value            | <2.2e-16
### Table A.4: Independent variable: Gender

**Coefficients:**

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>t-value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.775</td>
<td>1921.23</td>
<td>***</td>
</tr>
<tr>
<td>male</td>
<td>-0.021</td>
<td>-33.843</td>
<td>***</td>
</tr>
</tbody>
</table>

signif. Codes: 0,’***’ 0.001,’**’,0.01 ’*’, 0.05 ’.’ 0.1 ’ ’ 1

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple R-squared</td>
<td>0.00098</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-statistic</td>
<td>1145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;2.2e-16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table A.5: Independent variable: Education

**Coefficients:**

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>t-value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.777</td>
<td>1010.81</td>
<td>***</td>
</tr>
<tr>
<td>education_1</td>
<td>-0.010</td>
<td>-9.861</td>
<td>***</td>
</tr>
<tr>
<td>education_2</td>
<td>-0.013</td>
<td>-15.479</td>
<td>***</td>
</tr>
</tbody>
</table>

signif. Codes: 0,’***’ 0.001,’**’,0.01 ’*’, 0.05 ’.’ 0.1 ’ ’ 1

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple R-squared</td>
<td>0.00020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-statistic</td>
<td>119.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;2.2e-16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table A.6: Independent variable: Time

**Coefficients:**

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>t-value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.849</td>
<td>343.761</td>
<td>***</td>
</tr>
<tr>
<td>morning</td>
<td>-0.079</td>
<td>-31.217</td>
<td>***</td>
</tr>
<tr>
<td>afternoon</td>
<td>-0.098</td>
<td>-39.011</td>
<td>***</td>
</tr>
<tr>
<td>night</td>
<td>-0.057</td>
<td>-22.125</td>
<td>***</td>
</tr>
</tbody>
</table>

signif. Codes: 0,’***’ 0.001,’**’,0.01 ’*’, 0.05 ’.’ 0.1 ’ ’ 1

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple R-squared</td>
<td>0.0030</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-statistic</td>
<td>1186</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;2.2e-16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A.7: Mean value: Degree of speeding across time

<table>
<thead>
<tr>
<th>Hour</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.709</td>
</tr>
<tr>
<td>1</td>
<td>0.816</td>
</tr>
<tr>
<td>2</td>
<td>0.913</td>
</tr>
<tr>
<td>3</td>
<td>0.831</td>
</tr>
<tr>
<td>4</td>
<td>0.947</td>
</tr>
<tr>
<td>5</td>
<td>0.870</td>
</tr>
<tr>
<td>6</td>
<td>0.870</td>
</tr>
<tr>
<td>7</td>
<td>0.784</td>
</tr>
<tr>
<td>8</td>
<td>0.727</td>
</tr>
<tr>
<td>9</td>
<td>0.744</td>
</tr>
<tr>
<td>10</td>
<td>0.789</td>
</tr>
<tr>
<td>11</td>
<td>0.766</td>
</tr>
<tr>
<td>12</td>
<td>0.754</td>
</tr>
<tr>
<td>13</td>
<td>0.766</td>
</tr>
<tr>
<td>14</td>
<td>0.756</td>
</tr>
<tr>
<td>15</td>
<td>0.736</td>
</tr>
<tr>
<td>16</td>
<td>0.751</td>
</tr>
<tr>
<td>17</td>
<td>0.747</td>
</tr>
<tr>
<td>18</td>
<td>0.777</td>
</tr>
<tr>
<td>19</td>
<td>0.770</td>
</tr>
<tr>
<td>20</td>
<td>0.789</td>
</tr>
<tr>
<td>21</td>
<td>0.824</td>
</tr>
<tr>
<td>22</td>
<td>0.810</td>
</tr>
<tr>
<td>23</td>
<td>0.925</td>
</tr>
</tbody>
</table>

Table A.8: Mean value: Degree of speeding across gender

<table>
<thead>
<tr>
<th>Gender</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>female</td>
<td>0.775</td>
</tr>
<tr>
<td>male</td>
<td>0.754</td>
</tr>
</tbody>
</table>
Figure A.13: Significant code: Degree of speeding across time

<table>
<thead>
<tr>
<th>Age</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>5: 18 to 24</td>
<td>0.703</td>
</tr>
<tr>
<td>6: 25 to 34</td>
<td>0.820</td>
</tr>
<tr>
<td>7: 35 to 44</td>
<td>0.742</td>
</tr>
<tr>
<td>8: 45 to 54</td>
<td>0.753</td>
</tr>
<tr>
<td>9: 55 to 64</td>
<td>0.786</td>
</tr>
<tr>
<td>10: 65 to 74</td>
<td>0.764</td>
</tr>
<tr>
<td>11: 75 to 84</td>
<td>0.716</td>
</tr>
<tr>
<td>12: 85 and over</td>
<td>0.842</td>
</tr>
</tbody>
</table>
### Table A.10: Mean value: Degree of speeding across education level

<table>
<thead>
<tr>
<th>Education</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associates degree</td>
<td>0.789</td>
</tr>
<tr>
<td>Bachelors degree</td>
<td>0.774</td>
</tr>
<tr>
<td>Graduate/Post-graduate degree</td>
<td>0.750</td>
</tr>
<tr>
<td>High school graduate</td>
<td>0.768</td>
</tr>
<tr>
<td>Some college</td>
<td>0.760</td>
</tr>
<tr>
<td>Vocational/Technical training</td>
<td>0.806</td>
</tr>
<tr>
<td></td>
<td>Associates degree</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Associates degree</td>
<td>-</td>
</tr>
<tr>
<td>Some college</td>
<td>***</td>
</tr>
<tr>
<td>High school graduate</td>
<td>***</td>
</tr>
<tr>
<td>Vocational/Technical training</td>
<td>***</td>
</tr>
<tr>
<td>Bachelors degree</td>
<td>***</td>
</tr>
<tr>
<td>Graduate/Post-graduate degree</td>
<td>***</td>
</tr>
</tbody>
</table>

Figure A.15: Significant code: Degree of speeding across education level
A.4 Algorithm

A.4.1 Trip Generation

# Trip generation. Divide the GPS data into trip data.
# When the gap time is more than 300 seconds, divide the data.

import csv, dateutil, os
from itertools import islice
from dateutil import parser

tripEndTimeList = []
title = ['Longitude', 'Latitude', 'Speed (kilometer)', 'Course (degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude (meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance (meters)']

def getTimeElapsed(time1, time2):
    t1 = parser.parse(time1)
    t2 = parser.parse(time2)
    return (t2 - t1).seconds

def countTrips():
    count = 0
    line = 0
    root_path = "'/Users/toshiyokoo/Desktop/result1'"
    dir_path = os.path.join(root_path, filename[0:12])
    #os.mkdir(dir_path)
    inputFile = open(path + '/' + filename, 'rU')
    reader = csv.reader(inputFile)
    fileCount = len(inputFile.readlines())
    inputFile.seek(0)
    oldTime = None
oldDate = None
for row in islice(reader, 1, fileCount-1):
    line += 1
    if oldDate == None:
        oldDate = row[7]
        oldTime = row[8]
        tripFile = open(os.path.join(root_path, filename[0:12] + '_'+ str(count+1) +'.csv', 'a')
        print >>tripFile, ', '.join(title)
        print >>tripFile, ', '.join(row)
        tripFile.close()
        continue
    else:
        newDate = row[7]
        newTime = row[8]
        oldLine = line
        if line < fileCount - 2:
            if newDate != oldDate:
                tripEndTimeList.append(oldTime)
                oldDate = newDate
                oldTime = newTime
                count += 1
                tripFile = open(os.path.join(root_path, filename[0:12] + '_'+ str(count+1) +'.csv', 'a')
                print >>tripFile, ', '.join(title)
                join(title)

90
print >>tripFile, ','.join(row)
tripFile.close()
continue
if newDate == oldDate:
    if getTimeElapsed(
        oldTime, newTime
    ) >= 300:
        tripEndTimeList.append(oldTime)
oldDate = newDate
oldTime = newTime
count += 1
tripFile = open
    (os.path.join(
        root_path, filename
    )[0:12] + '_' +
     str(count+1) +'.csv', 'a'
)
print >>
    tripFile
    , ','.join(title)
elif line == fileCount - 2:
    tripFile = open(os.path.join(root_path, filename[0:12] + '_' + str(count+1) + '.csv', 'a'))
    print >> tripFile, ','.join(row)
    tripFile.close
    continue

else:
    oldDate = newDate
    oldTime = newTime
    tripFile = open(os.path.join(root_path, filename[0:12] + '_' + str(count+1) + '.csv', 'a'))
    print >> tripFile, ','.join(row)
    tripFile.close
    continue
tripFile.close
count +=1
tripEndTimeList.append(row[8])

print filename[0:12], count

path = "/Users/toshiyokoo/Desktop/aaac"
for filename in os.listdir(path):
    global filename
    countTrips()

A.4.2 Remove outside Minneapolis - St. Paul region (7 counties)

# Remove GPS points which are outside Minneapolis.
# Longitude: -94.0123 <= Minneapolis <= -92.7397
# Latitude: 44.4714 <= Minneapolis <= 45.4139

import csv, os
from itertools import islice
title = ['Longitude', 'Latitude', 'Speed(kilometer)', 'Course(degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude(meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance(meters)']
def insideMap():
    inputFile = open(path + '/' + filename, 'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    root_path = "/Users/toshiyokoo/Desktop/02_result"
    #dir_path = os.path.join(root_path, directory)
    #if not os.path.exists(dir_path):
    #    os.mkdir(dir_path)
    count = 0
    all_count = 0
#writer = csv.writer(outputFile, lineterminator='\n')
for row in islice(reader, 1, None):
    if (-94.0123 <= float(row[0]) <= -92.7397 and  
         44.4714 <= float(row[1]) <= 45.4139):
        count += 1
        all_count += 1
    else:
        all_count += 1
inputFile.seek(0)
if count == all_count:
    print filename[-4:] + ' is full'
    outputFile = open(os.path.join(root_path,  
                            filename[-4]+'.csv'), 'a')
    print >> outputFile, ',', join(title)
    for row in islice(reader, 1, None):
        print >> outputFile, ',', join(row)
        #writer.writerow(row)
    outputFile.close()
elif 0 < count < all_count:
    print filename[-4:] + ' is modified'
    outputFile = open(os.path.join(root_path,  
                            filename[-4]+'.csv'), 'a')
    print >> outputFile, ',', join(title)
    for row in islice(reader, 1, None):
        if (-94.0123 <= float(row[0]) <=  
             -92.7397) and (44.4714 <= float(row[1]) <= 45.4139):
            print >> outputFile, ',', join(row)
            #writer.writerow(row)
    outputFile.close()
elif count == 0:
    print filename[-4] + 'is empty'
inputFile.close()

path = "/Users/toshiyokoo/Desktop/aaaun"
for filename in os.listdir(path):
    global filename
    insideMap()

A.4.3 Remove speed that is less than 1 km/h

# Remove data whose speed is less than 1km/h.
import csv, os
from itertools import islice

title = ['Longitude', 'Latitude', 'Speed(kilometer)', 'Course(degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude(meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance(meters)']
def smallspeed():
    inputFile = open(path + '/' + filename, 'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    root_path = "/Users/toshiyokoo/Desktop/03_result"
    #dir_path = os.path.join(root_path, directory)
    #if not os.path.exists(dir_path):
    #    os.mkdir(dir_path)

    count = 0
    all_count = 0
    for row in islice(reader, 1, None):
        if int(row[2]) >= 1:
            count += 1
            all_count += 1

95
else:
    all_count += 1

inputFile.seek(0)
if count == all_count:
    print(filename[:−4] + ' is full')
    outputFile = open(os.path.join(root_path, filename[:−4]+'.csv'), 'a')
    writer = csv.writer(outputFile)
    print >> outputFile, ', '.join(title)
    for row in islice(reader, 1, None):
        print >> outputFile, ', '.join(row)
    outputFile.close()
elif 0 < count < all_count :
    print(filename[:−4] + ' is modified')
    outputFile = open(os.path.join(root_path, filename[:−4]+'.csv'), 'a')
    writer = csv.writer(outputFile)
    print >> outputFile, ', '.join(title)
    for row in islice(reader, 1, None):
        if int(row[2]) >= 1:
            print >> outputFile, ', '.join(row)
    outputFile.close()
elif count == 0:
    print filename[:−4] + 'is empty'
inputFile.close()

path = '/Users/toshiyokoo/Desktop/aaadw'
for filename in os.listdir(path):
    global filename
    smallspeed()
A.4.4 Mode classification

# This program is mode classification. Five kinds of modes are found in the trips.
# Walk, off-network public transport mode, bus, bicycle and car.

import csv, dateutil, os, math
from scipy import stats
from itertools import islice
from dateutil import parser

def distanceBus():
    busStopFile = open("/Users/toshiyokoo/Desktop/5133/bus_stop.csv", 'rU')
    busStop = csv.reader(busStopFile)
    inputFile.seek(0)
    GPSLonS = 0
    GPSLatS = 0
    startBusstop = 0
    for row in islice(reader, 1, None):
        if int(row[2]) >= 10: # speed is larger than 10km/h
            changeLonS = float(row[0])
            changeLatS = float(row[1])
            Longi_rad = changeLonS * math.pi / 180
            Lati_rad = changeLatS * math.pi / 180
            f = 1 / 298.257223563
            flattening

97
\[
\begin{align*}
n &= f / (2 - f) \\
a &= 6378137 \\
#\text{equatorial radius (meter)} \\
A &= (a / (1 + n)) * (1 + (n ** 2) / 4 + (n ** 4) / 64) \\
alphal &= 0.5 * n - 0.666666667 * n ** 2 \\
&\quad + 0.3125 * 1.0 * n ** 3 \#1/2, 2/3, 5/16 \\
alphal2 &= 0.270833333 * n ** 2 - 0.6 * n ** 3 \#13/48, 3/5 \\
alphal3 &= 0.254166667 * n ** 3 \#61/240 \\
t &= \text{math.sinh(math.atanh(math.sin(Lati_rad))} - ((2 * n ** 0.5) / (1 + n)) * \text{math.atanh(((2 * n ** 0.5) / (1 + n)) * math.sin(Lati_rad))) \\
\text{Initial_deg} &= 15 * 6 - 183 \#\text{zone15} \\
\text{Initial_rad} &= \text{Initial_deg} * \text{math.pi} / 180 \\
\xi &= \text{math.atan}(t / \text{math.cos(Longi_rad - Initial_rad)}) \\
\eta &= \text{math.atanh(math.sin(Longi_rad - Initial_rad)) / ((1 + t ** 2) ** 0.5)} \\
E0 &= 500000 \\
#\text{initial (meter)} \\
k0 &= 0.9996
\end{align*}
\]
\[ E = E_0 + k_0 \times A \times (\eta + \alpha_1 \times \cos(2 \times \xi) \times \sinh(2 \times \eta) + \alpha_2 \times \cos(4 \times \xi) \times \sinh(4 \times \eta) + \alpha_3 \times \cos(6 \times \xi) \times \sinh(6 \times \eta)) \]

\[ N = k_0 \times A \times (\xi + \alpha_1 \times \sin(2 \times \xi) \times \cosh(2 \times \eta) + \alpha_2 \times \sin(4 \times \xi) \times \cosh(4 \times \eta) + \alpha_3 \times \sin(6 \times \xi) \times \cosh(6 \times \eta)) \]

\[
\text{GPSLonS} = \text{float}(E) \\
\text{GPSLatS} = \text{float}(N)
\]

# print GPSLonS, GPSLatS
break

else:

\[
\text{GPSLonS} = 0 \\
\text{GPSLatS} = 0
\]

startBusstop = 0

i = 1

\[
\text{GPSLonE} = 0 \\
\text{GPSLatE} = 0
\]

endBusstop = 0

for row in reversed(list(reader)):

    if i < fileCount - 2:
        if int(row[2]) > 10: # speed is greater than 10km/h
            changeLonE = float(row[0])
            changeLatE = float(row[1])
            Longi_rad = changeLonE \times \text{math.} \pi / 180 # Longitude radians
\[
\text{Lati\_rad} = \text{changeLatE} \times \text{math.pi} / 180 \quad \# \quad \text{Latitude\_radians}
\]
\[
f = 1 / 298.257223563
\]
\[
\#\text{inverse flattening}
\]
\[
n = f / (2 - f)
\]
\[
a = 6378137
\]
\[
\# \quad \text{equatorial radius (meter)}
\]
\[
A = (a / (1 + n)) \times (1 + (n ** 2) / 4 + (n ** 4) / 64)
\]
\[
\alpha_1 = 0.5 \times n - 0.666666667 \times n ** 2 + 0.3125 \times 1.0 \times n ** 3 \quad \#1/2, 2/3, 5/16
\]
\[
\alpha_2 = 0.270833333 \times n ** 2 - 0.6 \times n ** 3 \quad \#13/48, 3/5
\]
\[
\alpha_3 = 0.254166667 \times n ** 3 \quad \#61/240
\]
\[
t = \text{math.sinh}((\text{math.atanh}(\text{math.sin(Lati\_rad)}) - ((2 \times n ** 0.5) / (1 + n)) \times \text{math.atanh}(((2 \times n ** 0.5) / (1 + n)) \times \text{math.sin(Lati\_rad))))
\]
\[
\text{Initial\_deg} = 15 \times 6 - 183
\]
\[
\#\text{zone15}
\]
\[
\text{Initial\_rad} = \text{Initial\_deg} \times \text{math.pi} / 180
\]
\[
\xi = \text{math.atan}(t / \text{math.cos}(\text{Longi\_rad} - \text{Initial\_rad}))
\]
eta = math.atan(math.sin(Longi_rad - Initial_rad) / ((1 + t ** 2) ** 0.5))

E0 = 500000

# initial (meter)
k0 = 0.9996

E = E0 + k0 * A * (eta + alpha1 * math.cos(2 * xi) * math.sinh(2 * eta) + alpha2 * math.cos(4 * xi) * math.sinh(4 * eta) + alpha3 * math.cos(6 * xi) * math.sinh(6 * eta))

N = k0 * A * (xi + alpha1 * math.sin(2 * xi) * math.cosh(2 * eta) + alpha2 * math.sin(4 * xi) * math.cosh(4 * eta) + alpha3 * math.sin(6 * xi) * math.cosh(6 * eta))

GPSLonE = float(E)
GPSLatE = float(N)

# print GPSLonE, GPSLatE
break

else:
    GPSLonE = 0
    GPSLatE = 0
endBusstop = 0

else:
    GPSLonE = 0
    GPSLatE = 0
endBusstop = 0
i += 1

if (GPSLonS != 0 and GPSLatS != 0 and GPSLonE != 0 and
    GPSLatE != 0 and GPSLonS != GPSLonE and GPSLatS !=
    GPSLatE):
    for row in slice(busStop, 1, None):
        busLonS = float(row[0])
        busLatS = float(row[1])
        dS = ((GPSLonS - busLonS) ** 2 +
          (GPSLatS - busLatS) ** 2) ** 0.5
        #print 'busS=', dS
        if dS <= 50:  # distance less than 50 meters
            startBusstop = 1
            #print startBusstop
            break
    else:
        startBusstop = 0

busStopFile.seek(0)
for row in slice(busStop, 1, None):
    busLonE = float(row[0])
    busLatE = float(row[1])
    dE = ((GPSLonE - busLonE) ** 2 +
      (GPSLatE - busLatE) ** 2) ** 0.5
    #print 'busE=', dE
    if dE <= 50:  # distance less than 50 meters
        endBusstop = 1
        break
    else:
endBusstop = 0
else:
    startBusstop = 0
    endBusstop = 0

if (startBusstop == 1 and endBusstop == 1):
    distanceBus = 1
else:
    distanceBus = 0

return distanceBus

def distanceTrain():
    trainStopFile = open("/Users/toshiyokoo/Desktop/5133/
    train_station.csv",'rU')
    trainStop = csv.reader(trainStopFile)
    inputFile.seek(0)
    GPSLonS = 0
    GPSLatS = 0
    startTrainstop = 0
    for row in islice(reader,1,None):
        if int(row[2])>= 10: #speed is larger than 10km /h
            TchangeLonS = float(row[0])
            TchangeLatS = float(row[1])
            Longi_rad = TchangeLonS * math.pi / 180
            #Longitude_radians
            Lati_rad = TchangeLatS * math.pi / 180
            #Latitude_radians
            f = 1 / 298.257223563
            #inverse flattening
\( n = \frac{f}{(2 - f)} \)
\( a = 6378137 \)

#equatorial radius (meter)
\[
A = \left( \frac{a}{1 + n} \right) \times \left( 1 + \left( n \times n \right)^2 / 4 + \left( n \times n \right)^4 / 64 \right)
\]
\[
\text{alpha1} = 0.5 \times n - 0.666666667 \times n \times n^2 + 0.3125 \times 1.0 \times n \times n^3 \quad \#1/2, 2/3, 5/16
\]
\[
\text{alpha2} = 0.270833333 \times n \times n^2 - 0.6 \times n \times n^3 \quad \#13/48, 3/5
\]
\[
\text{alpha3} = 0.254166667 \times n \times n^3 \quad \#61/240
\]
\[
t = \text{math.sinh}(\text{math.atanh}(\text{math.sin}(\text{Lati.rad})) - ((2 \times n \times 0.5) / (1 + n)) \times \text{math.atanh}((2 \times n \times 0.5) / (1 + n)) \times \text{math.sin}(\text{Lati.rad}))
\]
\[
\text{Initial.deg} = 15 \times 6 - 183 \quad \#\text{zone15}
\]
\[
\text{Initial.rad} = \text{Initial.deg} \times \text{math.pi} / 180
\]
\[
\text{xi} = \text{math.atan}(t / \text{math.cos}(\text{Longi.rad} - \text{Initial.rad}))
\]
\[
\text{eta} = \text{math.atanh}(\text{math.sin}(\text{Longi.rad} - \text{Initial.rad}) / ((1 + t \times t) \times 0.5))
\]
\[
E0 = 500000
\]

#initial (meter)
\[
k0 = 0.9996
\]
\[ E = E_0 + k_0 \times A \times (\eta + \alpha_1 \times \text{math.}\cos(2 \times \xi) \times \text{math.}\sinh(2 \times \eta) + \alpha_2 \times \text{math.}\cos(4 \times \xi) \times \text{math.}\sinh(4 \times \eta) + \alpha_3 \times \text{math.}\cos(6 \times \xi) \times \text{math.}\sinh(6 \times \eta)) \]

\[ N = k_0 \times A \times (\xi + \alpha_1 \times \text{math.}\sin(2 \times \xi) \times \text{math.}\cosh(2 \times \eta) + \alpha_2 \times \text{math.}\sin(4 \times \xi) \times \text{math.}\cosh(4 \times \eta) + \alpha_3 \times \text{math.}\sin(6 \times \xi) \times \text{math.}\cosh(6 \times \eta)) \]

\( \text{GPSLonS} = \text{float}(E) \)
\( \text{GPSLatS} = \text{float}(N) \)
\text{break}

\( \text{else:} \)
\( \text{GPSLonS} = 0 \)
\( \text{GPSLatS} = 0 \)
\text{startTrainstop} = 0

\( i = 1 \)
\( \text{GPSLonE} = 0 \)
\( \text{GPSLatE} = 0 \)
\( \text{endTrainstop} = 0 \)
\text{for} \text{row} \text{in} \text{reversed(list(reader))}:
\quad \text{if} \ i < \text{fileCount} - 2:
\quad \text{if} \ \text{int(row[2])}>10: \# \text{speed is greater than 10km/h}
\quad \text{TchangeLonE} = \text{float(row[0])} \)
\quad \text{TchangeLatE} = \text{float(row[1])} \)
\quad \text{Longi_rad} = \text{TchangeLonE} \times \text{math.}\pi / 180 \# \text{Longitude radians} \)

105
Lati_rad = TchangeLatE * math.pi / 180  # Latitude radians
f = 1 / 298.257223563

#inverse flattening
n = f / (2 - f)
a = 6378137

# equatorial radius (meter)
A = (a / (1 + n)) * (1 + (n ** 2) / 4 + (n ** 4) / 64)
alpha1 = 0.5 * n - 0.666666667 * n ** 2 + 0.3125 * 1.0 * n ** 3  #1/2, 2/3, 5/16
alpha2 = 0.270833333 * n ** 2 - 0.6 * n ** 3  #13/48, 3/5
alpha3 = 0.254166667 * n ** 3  #61/240

t = math.sinh(math.atanh(math.sin(Lati_rad)) - ((2 * n ** 0.5) / (1 + n)) * math.atanh(((2 * n ** 0.5) / (1 + n)) * math.sin(Lati_rad)))
Initial_deg = 15 * 6 - 183

#zone15
Initial_rad = Initial_deg * math.pi / 180
xi = math.atan(t / math.cos(Longi_rad - Initial_rad))
eta = math.atan(math.sin(
    Longi_rad - Initial_rad) /
    ((1 + t ** 2) ** 0.5))
E0 = 500000

# initial (meter)
k0 = 0.9996
E = E0 + k0 * A * (eta + alpha1 * math.cos(2 * xi) * math.sinh(2 * eta) + alpha2 * math.cos(4 * xi) * math.sinh(4 * eta) + alpha3 * math.cos(6 * xi) * math.sinh(6 * eta))
N = k0 * A * (xi + alpha1 * math.sin(2 * xi) * math.cosh(2 * eta) + alpha2 * math.sin(4 * xi) * math.cosh(4 * eta) + alpha3 * math.sin(6 * xi) * math.cosh(6 * eta))
GPSLonE = float(E)
GPSLatE = float(N)
break
else:
    GPSLonE = 0
    GPSLatE = 0
    endTrainstop = 0
else:
    GPSLonE = 0
    GPSLatE = 0
    endTrainstop = 0
i +=1

if (GPSLonS != 0 and GPSLatS != 0 and GPSLonE != 0 and
    GPSLatE != 0 and GPSLonS != GPSLonE and GPSLatS !=
    GPSLatE):

    for row in islice(trainStop, 1, None):
        trainLonS = float(row[0])
        trainLatS = float(row[1])
        dS = ((GPSLonS - trainLonS) ** 2 + (GPSLatS - trainLatS) ** 2) ** 0.5
        #print 'trainS=', dS
        if dS <=150: #distance less than 150 meters
            startTrainstop = 1
            break
        else:
            startTrainstop = 0

trainStopFile.seek(0)
for row in islice(trainStop, 1, None):
    trainLonE = float(row[0])
    trainLatE = float(row[1])
    dE = ((GPSLonE - trainLonE) ** 2 + (GPSLatE - trainLatE) ** 2) ** 0.5
    #print 'trainE=', dE
    if dE <=150: #distance less than 150 meters
        endTrainstop = 1
        break
    else:
        endTrainstop = 0
else:
startTrainstop = 0
endTrainstop = 0

if (startTrainstop == 1 and endTrainstop == 1):
    distanceTrain = 1
else:
    distanceTrain = 0

return distanceTrain

title = ['Longitude', 'Latitude', 'Speed(kilometer)', 'Course(degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude(meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance(meters) ']
def checkMode():
    
    fileNumber = 1
    numberWalk = 0
    numberBike = 0
    numberTrain = 0
    numberBus = 0
    numberDrive = 0
    numberError = 0
    inputfile = open(path + '/ ' + filename , 'rU')
    reader = csv.reader(inputfile)
    fileCount = len(inputfile.readlines())
    global inputFile , reader , fileCount
    inputFile .seek(0)
    count = 0
    speed = []
    for row in islice(reader, 1, None):
        count += 1
        speed.append(int(row[2]))
        
        if (count == 1):
#

```python
startSpeed = int(row[2])
startTime = parser.parse(row[8])

if (count == fileCount - 1):
    endTime = parser.parse(row[8])

Cspeed = (sum([speed]) / count)

print filename[:-4], Cspeed, stats.scoreatpercentile(speed, 85)

inputFile.seek(0)

if (endTime - startTime).seconds >= 60:
    if ((sum([speed]) / count) <= 6 and stats.
        scoreatpercentile(speed, 85) <= 10 and max(speed) <= 20):
        root_walk = '/Users/tosiyokoo/Desktop
        /04_result_walk'
        outputFile_walk = open(os.path.join(
            root_walk, filename[:-4] + '.csv'),
            'a')
        print >> outputFile_walk, ',', 'join('
            title)
        for row in islice(reader, 1, None):
            print >> outputFile_walk, ',', 'join(row)
        outputFile_walk.close()
        numberWalk += 1

elif (distanceTrain() == 1 and (sum([speed]) / count) >= 10):
    root_train = '/Users/tosiyokoo/Desktop
    /04_result_train'
    outputFile_train = open(os.path.join(
        root_train, filename[:-4] + '.csv'),
        'a')
```

110
print >> outputFile_train, ', '.join(title)
inputFile.seek(0)
for row in islice(reader, 1, None):
    print >> outputFile_train, ', '.join(row)
outputFile_train.close()
numberTrain += 1

elif (distanceBus() == 1 and (sum((speed)) / count) >= 10):
    root_bus = '/Users/toshibo/Desktop
    /04_result_bus'
    outputFile_bus = open(os.path.join(root_bus, filename[:−4] + '.csv'), 'a')
    print >> outputFile_bus, ', '.join(title)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile_bus, ', '.join(row)
    outputFile_bus.close()
    numberBus += 1

elif (stats.scoreatpercentile(speed, 85) > 10 and stats.scoreatpercentile(speed, 85) < 20 and max(speed) <= 30):
    root_bike = '/Users/toshibo/Desktop
    /04_result_bike'
    outputFile_bike = open(os.path.join(root_bike, filename[:−4] + '.csv'), 'a')
print >> outputFile_bike, ','.join(title)
inputFile.seek(0)
for row in islice(reader,1,None):
    print >> outputFile_bike, ','.join(row)
outputFile_bike.close()
numberBike +=1

elif ((sum((speed)) / count) >= 10):
    root_drive = '/Users/toshiyokoo/Desktop/04_result_drive'
    outputFile_drive = open(os.path.join(
        root_drive, filename[:-4] + '.csv'), 'a')
    print >> outputFile_drive, ','.join(title)
inputFile.seek(0)
for row in islice(reader,1,None):
    print >> outputFile_drive, ','.join(row)
outputFile_drive.close()
numberDrive +=1
else:
    root_error = '/Users/toshiyokoo/Desktop/04_result_error'
    outputFile_error = open(os.path.join(
        root_error, filename[:-4] + '.csv'), 'a')
    print >> outputFile_error, ','.join(title)
inputFile.seek(0)
for row in islice(reader,1,None):
print >> outputFile_error, ',',
     join(row)
outputFile_error.close()
numberError +=1
else:
    root_error = '/Users/toshiyokoo/Desktop/04
    _result_error'
outputFile_error = open(os.path.join(root_error
     , filename[:-4] + '.csv'), 'a')
print >> outputFile_error, ',',join(title)
inputFile.seek(0)
for row in islice(reader,1,None):
    print >> outputFile_error, ',',join(row
     )
outputFile_error.close()
numberError +=1

path = "'/Users/toshiyokoo/Desktop/aaadx"
for filename in os.listdir(path):
    global filename
    checkMode()

A.4.5  Remove speed that is less than or equal to 5 km/h

# Remove data whose speed is less than 5km/h.
import csv,os
from itertools import islice

title = ['Longitude','Latitude','Speed(kilometer)','Course(
degrees)','NumberOfSatellites','HDOP','Altitude (meters)'
     ,'DD
/MM/YY','HH:MM:SS','Distance (meters) ']
def smallspeed():
    inputFile = open(path + '/' +filename,'rU')
reader = csv.reader(inputFile)
inputFile.seek(0)
root_path = "/Users/toshiyokoo/Desktop/05_result"
#dir_path = os.path.join(root_path, directory)
#if not os.path.exists(dir_path):
#    os.mkdir(dir_path)

count = 0
all_count = 0
for row in islice(reader, 1, None):
    if int(row[2]) > 5:
        count += 1
        all_count += 1
    else:
        all_count += 1

inputFile.seek(0)
if count == all_count:
    print filename[:−4] + 'is full'
    outputFile = open(os.path.join(root_path, filename[:−4]+'.csv'), 'a')
    writer = csv.writer(outputFile)
    print >> outputFile, ', '.join(title)
    for row in islice(reader, 1, None):
        print >> outputFile, ', '.join(row)
    outputFile.close()
elif 0 < count < all_count:
    print filename[:−4] + 'is modified'
    outputFile = open(os.path.join(root_path, filename[:−4]+'.csv'), 'a')
    writer = csv.writer(outputFile)
    print >> outputFile, ', '.join(title)
for row in islice(reader, 1, None):
    if int(row[2]) > 5:
        print >> outputFile, ','.join(row)
    outputFile.close()

elif count == 0:
    print filename[:-4] + ' is empty'
inputFile.close()

path = ""/Users/toshiyokoo/Desktop/aaady"
for filename in os.listdir(path):
    global filename
    smallspeed()

A.4.6 Convert coordinate system

# convert latitude & longitude to UTM data

import csv, os
from itertools import islice
import math
title = ['Longitude', 'Latitude', 'Speed(kilometer)', 'Course(degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude(meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance(meters)', 'GX', 'GY']
def convert():
    inputFile = open(path + '/' + filename, 'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    root_path = ""/Users/toshiyokoo/Desktop/06_result"
    #dir_path = os.path.join(root_path, directory)
    #if not os.path.exists(dir_path):
    #    os.mkdir(dir_path)
outputFile = open(os.path.join(root_path, filename[:-4] + '.csv'), 'a')
writer = csv.writer(outputFile)
print >> outputFile, ',', '.join(title)

for row in islice(reader, 1, None):
    Longi_deg = float(row[0])
        #Longitude_degree
    Lati_deg = float(row[1])
        #Latitude_degree
    Longi_rad = Longi_deg * math.pi / 180
        #Longitude_radians
    Lati_rad = Lati_deg * math.pi / 180
        #Latitude_radians
    f = 1 / 298.257223563
        #inverse flattening
    n = f / (2 - f)
    a = 6378137
        #equatorial
    
    radius (meter)
    A = (a / (1 + n)) * (1 + (n ** 2) / 4 + (n ** 4) / 64)
    alpha1 = 0.5 * n - 0.666666667 * n ** 2 +
        0.3125 * 1.0 * n ** 3 #1/2, 2/3, 5/16
    alpha2 = 0.270833333 * n ** 2 - 0.6 * n ** 3
        #13/48, 3/5
    alpha3 = 0.254166667 * n ** 3 #61/240
    t = math.sinh(math.atanh(math.sin(Lati_rad)) -
        ((2 * n ** 0.5) / (1 + n)) * math.atanh(((2 * n ** 0.5) / (1 + n)) * math.sin(Lati_rad)))
    Initial_deg = 15 * 6 - 183
        #zone15
Initial_rad = Initial_deg * math.pi / 180

xi = math.atan(t / math.cos(Longi_rad - Initial_rad))

eta = math.atanh(math.sin(Longi_rad - Initial_rad) / ((1 + t ** 2) ** 0.5))

E0 = 500000

# initial (meter)
k0 = 0.9996

E = E0 + k0 * A * (eta + alpha1 * math.cos(2 * xi) * math.sinh(2 * eta) + alpha2 * math.cos(4 * xi) * math.sinh(4 * eta) + alpha3 *  
math.cos(6 * xi) * math.sinh(6 * eta))

N = k0 * A * (xi + alpha1 * math.sin(2 * xi) * 
math.cosh(2 * eta) + alpha2 * math.sin(4 * xi) * math.cosh(4 * eta) + alpha3 * math.sin(6 * xi) * math.cosh(6 * eta))

East = str(E)

North = str(N)

row.extend([East, North])

print >> outputFile, '"' + ','.join(row)

print filename[:−4]

inputFile.close()

path = "/Users/toshiyokoo/Desktop/aaaea"

for filename in os.listdir(path):
    global filename
    convert()

A.4.7 Matching of person ID

# extract file that match tripid of TBI data
import csv, os
from itertools import islice
import math
import shutil
title = ['Longitude', 'Latitude', 'Speed (kilometer)', 'Course (degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude (meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance (meters)', 'GX', 'GY']
def matching()
    inputFile = open(path + '/ ' + filename, 'rU')
    reader = csv.reader(inputFile)
    TBIFile = open("/Users/toshibyokoo/Desktop/5133/TBIdata.csv", 'rU')
idlist = csv.reader(TBIFile)
TBIFile.seek(0)
id = 0
for row in islice(idlist, 1, None):
    if int(row[3]) == int(tripid):
        root_path = '/Users/toshibyokoo/Desktop/07_result'
        outputFile = open(os.path.join(
            root_path, filename[:-4] + '.csv'), 'a')
        print >> outputFile, ','.join(title)
        inputFile.seek(0)
        for row in islice(reader, 1, None):
            print >> outputFile, ',', ','.join(row)
        outputFile.close()
        print filename[:-4], "match"
id = 1

if id == 0:
    root_error = '/Users/toshiyokoo/Desktop/07_result_error'
    outputFile_error = open(os.path.join(root_error, filename[:-4] + '.csv'), 'a')
    print >> outputFile_error, ',', join(title)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile_error, ',', join(row)
    outputFile_error.close()
    print filename[:-4], "mismatch"
    inputFile.close()

path = "'/Users/toshiyokoo/Desktop/aaaee"
for filename in os.listdir(path):
    global filename
    matching()

A.4.8 Remove matching error and Map-matching

#load GPS
#— 1 Set pathname here
InDrPth1='Users/toshiyokoo/Desktop/aaadm/'

#— 2 Set pathname here
InDrPth2='Users/toshiyokoo/Desktop/aaadn/'

#execute algorithm_sub0
execfile ('Users/toshiyokoo/Desktop/Matching/sub0_matching.py')

import csv, os, numpy, math

119
from itertools import islice
import processing
import time

def matching():
    # load GIS map
    #—— 1 Load map layer
    InGIS = "/Users/toshiyokoo/Desktop/Buffer0318/10m_buffer.shp"
    vlayer = QgsVectorLayer(InGIS, "10m_buffer", "ogr")

    #—— 2 Confirm something is loaded and valid
    vlayer.isValid()

    #—— 3 Display the layer into QGIS
    QgsMapLayerRegistry.instance().addMapLayer(vlayer)

    # Part1
    # load GPS data
    #—— 2 Set file name here
    InFlnm1=filename
    #—— 3 Build file name an path for uri
    InFlPth1="file:///"+InDrPth1+InFlnm1
    #gps_File = open(InDrPth1+InFlnm1,'rU')
    #gps_File.seek(0)
    #—— 4 Set import Sting here note only need to set x and y other come for free
    uril = InFlPth1+"?delimiter=%s&xField=%s&yField=%s" % ("","","GX","GY")
    # "Longitude" , "Latitude" -> "GX" , "GY"
    #—— 5 Load point layer
    bh1 = QgsVectorLayer(uril, InFlnm1, "delimitedtext")
# 6 Confirm something is loaded and valid
bh1.isValid()

# 7 Set CRS
bh1.setCrs(QgsCoordinateReferenceSystem(26915,
          QgsCoordinateReferenceSystem.EpsgCrsId))
#4326: WGS84, 26915: NAD83/UTM zone 15N

# 8 Display the layer into QGIS
QgsMapLayerRegistry.instance().addMapLayer(bh1)

# select by location
processing.runalg("qgis:selectbylocation", vlayer, uri1, 0)

# save vector layer as csv (True: selected, False: all)
cLayer = qgis.utils.iface.mapCanvas().layers()[1]
gi1 = 'GIS1.'+InFlnm1
gis1 = 'Users/toshiyokoo/Desktop/5536/'+gi1
QgsVectorFileWriter.writeAsVectorFormat(cLayer, gis1, "CP1250", None, "CSV", True)
global InDrPth2, gis1, InFlnm1

# execute algorithm_sub1
# remove intersection & missing link GPS data
execfile('Users/toshiyokoo/Desktop/Matching/
         sub1_remove_intersect.py')

# Part 2
# load GPS data

# 1 Set file name here
InFlnm2=InFlnm1

# 2 Set pathname here
#InDrPth2='Users/toshiyokoo/Desktop/5880/'

# 3 Build file name an path for uri
InFlPth2="file:///"+InDrPth2+InFlnm2
#— 4 Set import Sting here note only need to set x and y other come for free
uri2 = InFlPth2+"?delimiter=%s\&xField=%s\&yField=%s" %
(",", "GX", "GY")
#"Longitude", "Latitude" -> "GX", "GY"
#— 5 Load point layer
bh2 = QgsVectorLayer(uri2, InFlnm2, "delimitedtext")
#— 6 Confirm something is loaded and valid
bh2.isValid()
#— 7 Set CRS
bh2.setCrs(QgsCoordinateReferenceSystem(26915,
    QgsCoordinateReferenceSystem.EpsgCrsId))
#4326: WGS84, 26915: NAD83/UTM zone 15N
#— 8 Display the layer into QGIS
QgsMapLayerRegistry.instance().addMapLayer(bh2)
#— select by location
processing.runalg("qgis:selectbylocation", vlayer, uri2, 0)
#—save vector layer as csv (True: selected, False: all)
cLayer = qgis.utils.iface.mapCanvas().layers()[2]
gi2 = 'GIS2_' + filename
gis2 = 'Users/toshibiyokoo/Desktop/5536/' + gi2
global gis2, InFlnm2
QgsVectorFileWriter.writeAsVectorFormat(cLayer, gis2, 
    CP1250" , None,"CSV", True)
#execute algorithm_sub2
# find nearest link between GPS point and GIS map
# calculate minimum height
# link nearest (Streetname and Speedlimit)
execfile('Users/toshibiyokoo/Desktop/Matching/
sub2_matching.py')
#remove GPS layer
QgsMapLayerRegistry.instance().removeMapLayer(bh1.id())
QgsMapLayerRegistry.instance().removeMapLayer(bh2.id())
QgsMapLayerRegistry.instance().removeMapLayer(vlayer.id())

#reset selected layer
#cLayer.setSelectedFeatures([0])

−−−
1 Set file name here
InFlnm3=InFlnm2
InDrPth3='/Users/toshiyokoo/Desktop/box/'+InFlnm3

#move file
os.rename(gis1, '/Users/toshiyokoo/Desktop/comp1/'+gi1)
os.rename(gis2, '/Users/toshiyokoo/Desktop/comp1/'+gi2)
os.rename(InDrPth3, '/Users/toshiyokoo/Desktop/comp1
*/'+InFlnm3)

#gps_File.close()

for filename in os.listdir(InDrPth1):
    #global filename, InFlnm2, gi1, gi2, gis1, gis2
    matching()

# remove intersection & missing link GPS data

import csv, os, numpy, math
from itertools import islice
import time

#start = time.time()

global title1
title1 = ['Longitude', 'Latitude', 'Speed (kilometer)', 'Course (degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude (meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance (meters)', 'GX', 'GY']

def intersect():
    gps_inputFile1 = open(InDrPth1 + InFlnm1, 'rU')
    gps_reader1 = csv.reader(gps_inputFile1)
    gps_inputFile1.seek(0)
    root_path1 = InDrPth2
    outputFile1 = open(os.path.join(root_path1, InFlnm1[4] + '.csv'), 'a')
    writer1 = csv.writer(outputFile1, lineterminator='
')
    print >> outputFile1, ',', join(title1)
    gis_inputFile1 = open(gis1, 'rU')
    gis_reader1 = csv.reader(gis_inputFile1)
    min_height1 = 99  # dummy
    speedlimit1 = 99   # dummy
    for row1 in islice(gps_reader1, 1, None):
        list1 = []
        x_gps1 = float(row1[10])
        y_gps1 = float(row1[11])
        gis_inputFile1.seek(0)
        count1 = 0
        for j in islice(gis_reader1, 1, None):
            direction1 = float(j[42])
            ox_gis1 = float(j[43])  # OX
            oy_gis1 = float(j[44])  # OY
            dx_gis1 = float(j[45])  # DX
            dy_gis1 = float(j[46])  # DY
            if direction1 == 1:
                if (ox_gis1 - 10 <= x_gps1 <= dx_gis1 + 10):
if (oy_gis1 - 10) <= 
y_gps1 <= (dy_gis1 + 10):
    count1 += 1
elif (dy_gis1 - 10) <= 
y_gps1 <= (oy_gis1 + 10):
    count1 += 1
elif (dx_gis1 - 10) <= x_gps1 <= ox_gis1 + 10):
    if (oy_gis1 - 10) <= 
y_gps1 <= (dy_gis1 + 10):
        count1 += 1
    elif (dy_gis1 - 10) <= 
        y_gps1 <= (oy_gis1 + 10):
        count1 += 1

if count1 == 1:
    print >> outputFile1, ','.join(row1)

gis_inputFile1.close()
gps_inputFile1.close()
print InFlnm1
#elapsed_time = time.time() - start
#print("elapsed_time:{0}".format(elapsed_time))

#global filename, InDrPth2, InFlnm1, InFlnm2, gi1, gi2, gis1, gis2
intersect()
# find nearest link between GPS point and GIS map
# calculate minimum height
# link nearest (Streetname and Speedlimit)

import csv, os, numpy, math
from itertools import islice
import time

# start = time.time()

global title2
title2 = ['Longitude', 'Latitude', 'Speed(kilometer)', 'DD/MM/YY', 'HH:MM:SS', 'GX', 'GY', 'Min_hight', 'Streetname', 'Streetlength', 'Roadtype', 'Speedlimit(mph)']

def nearlink():
    gps_inputFile = open(InDrPath2 + InFlnm2, 'rU')
gps_reader = csv.reader(gps_inputFile)
gps_inputFile.seek(0)
root_path = '/Users/toshiyokoo/Desktop/box'
outputFile = open(os.path.join(root_path, InFlnm2[:-4] + '.csv'), 'a')
writer = csv.writer(outputFile, lineterminator='
')
print >> outputFile, ','.join(title2)
gis_inputFile = open(gis2, 'rU')
gis_reader = csv.reader(gis_inputFile)
min_height = 99 #dummy
speedlimit = 99 #dummy
for row in islice(gps_reader, 1, None):
    list = []
x_gps = float(row[10])
y_gps = float(row[11])
126
longitude = str(row[0])
latitude = str(row[1])
speed = str(row[2])
date = str(row[7])
hms = str(row[8])
GX = str(row[10])
GY = str(row[11])
gis_inputFile.seek(0)
for i in islice(gis_reader, 1, None):
    direction = float(i[42])
ox_gis = float(i[43]) #OX
oy_gis = float(i[44]) #OY
dx_gis = float(i[45]) #DX
dy_gis = float(i[46]) #DY
if direction == 1:
    if (ox_gis <= x_gps <= dx_gis):
        if (oy_gis - 30) <=
            y_gps <= (oy_gis +
                       30):
            length = ((
                        ox_gis -
                       dx_gis) ** 2
                + (oy_gis -
                       dy_gis) **
                2) ** 0.5
            gis_o = numpy.
                array([[ox_gis,
                       oy_gis]])
gis_d = numpy.array([dx_gis, dy_gis])
gps = numpy.array([x_gps, y_gps])
v1 = gps - gis_o
v2 = gis_d - gis_o
Area = numpy.cross(v1, v2)
D = abs(Area)
height = D / length
if min_height > height:
    min_height = height
streetlength = str(i[0])
streetname = str(i[5])
roadtype = str(i[27])
speedlimit = str(i[31])

# print '1',
min_height

eif (oy_gis <= y_gps <= dy_gis):  #
road x-y & curve
length = ((
ox_gis -
dx_gis)**2 + (oy_gis -
dy_gis)**2)**0.5
gis_o = numpy.
    array([[ox_gis,
            oy_gis]])
gis_d = numpy.
    array([[dx_gis,
            dy_gis]])
gps = numpy.
    array([[x_gps,
            y_gps]])
v1 = gps -
gis_o
v2 = gis_d -
gis_o
Area = numpy.
    cross(v1, v2)
D = abs(Area)
height = D /
    length
if
    min_height >
    height:
        min_height =
    height

130
streetlength = str( i [0])
streetname = str( i [5])
roadtype = str( i [27])
speedlimit = str( i [31])

elif (dy_gis <= y_gps <= oy_gis): #
road x−y & curve
length = ((ox_gis − dx_gis) ** 2 + (oy_gis − dy_gis) ** 2) ** 0.5
gis_o = numpy.array([[ox_gis, oy_gis]])
gis_d = numpy.array([[dx_gis, dy_gis]])
gps = numpy.array([[x_gps, y_gps]])
v1 = gps - gis_o
v2 = gis_d - gis_o
Area = numpy.cross(v1, v2)
D = abs(Area)
height = D / length
if min_height > height:
    min_height = height
streetlength = str(i[0])
streetname = str(i[5])
roadtype = str(i[27])
speedlimit = str(i[31])

elif (dx_gis <= x_gps <= ox_gis):
    if (oy_gis - 30) <= y_gps <= (oy_gis + 30):
length = ((
    ox_gis -
    dx_gis) ** 2
+ (oy_gis -
    dy_gis) ** 
2) ** 0.5

gis_o = numpy.array([ox_gis,
    oy_gis])
gis_d = numpy.array([[dx_gis,
    dy_gis]])
gps = numpy.array([[x_gps,
    y_gps]])

v1 = gps -
    gis_o

v2 = gis_d -
    gis_o

Area = numpy.cross(v1, v2)

D = abs(Area)

height = D / length

if
    min_height >
    height:
min_height = height

streetlength = str(i[0])
streetname = str(i[5])
roadtype = str(i[27])
speedlimit = str(i[31])

# print '1',

min_height
elif (oy_gis <= y_gps <= dy_gis):
    # road x-y & curve
    length = ((
        ox_gis -
        dx_gis) ** 2
    + (oy_gis -
        dy_gis) **
        2) ** 0.5
gis_o = numpy.array([
    ox_gis,
    oy_gis])
gis_d = numpy.array([
    dx_gis,
    dy_gis])
gps = numpy.array([
    x_gps,
    y_gps])
v1 = gps -
    gis_o
v2 = gis_d -
    gis_o
Area = numpy.
cross(v1, v2)
D = abs(Area)
height = D /
    length
if
    min_height >
    height:
        min_height
        =
        height

streetlength
        =
        str(
            i
            [0])
streetname
        =
        str(
            i
            [5])
roadtype
        =
        str(
            i
            [27])
speedlimit
        =
        str(
            i
            [31])

elif (dy_gis <= y_gps
     <= oy_gis):  #
    road x–y & curve
length = ((
    ox_gis -
    dx_gis)**2
+ (oy_gis -
    dy_gis)**2
) ** 0.5

gis_o = numpy.array([
    ox_gis,
    oy_gis])

gis_d = numpy.array([
    dx_gis,
    dy_gis])

gps = numpy.array([[
    x_gps,
    y_gps]])

v1 = gps -
    gis_o
v2 = gis_d -
    gis_o

Area = numpy.cross(v1, v2)

D = abs(Area)

height = D / length

if
    min_height >
    height:

138
min_height = height

streetlength = str(i[0])
streetname = str(i[5])
roadtype = str(i[27])
speedlimit = str(i[31])

elif (oy_gis <= y_gps <= dy_gis):

if (ox_gis - 30) <= x_gps <= (ox_gis + 30):  # road y-axis, interstate range 50m
    length = (((ox_gis - dx_gis) ** 2 + (oy_gis - dy_gis) ** 2) ** 0.5)
gis_o = numpy.array([ox_gis, oy_gis])
gis_d = numpy.array([dx_gis, dy_gis])
gps = numpy.array([x_gps, y_gps])

v1 = gps - gis_o
v2 = gis_d - gis_o
Area = numpy.cross(v1, v2)

D = abs(Area)

height = D / length
```python
if
    min_height >
height:
    min_height
    =
    height

streetlength
    =
    str(i
    [0])
streetname
    =
    str(i
    [5])
roadtype
    =
    str(i
    [27])
speedlimit
    =
    str(i
    [31])

elif (dy_gis <= y_gps <= oy_gis
    ):
if (ox_gis - 30) <=
    x_gps <= (ox_gis +
    30):  # road y-
    axis, interstate
    range 50m
    length = ((
        ox_gis -
        dx_gis) ** 2
        + (oy_gis -
        dy_gis) **
        2) ** 0.5
gis_o = numpy.
    array([[ox_gis,
        oy_gis]])
gis_d = numpy.
    array([[dx_gis,
        dy_gis]])
gps = numpy.
    array([[x_gps
        , y_gps]])
v1 = gps -
    gis_o
v2 = gis_d -
    gis_o
Area = numpy.
    cross(v1, v2)
D = abs(Area)
height = D /
    length
if 
    min_height > height:
    min_height
    =
    height

streetlength
    =
    str(
    i
    [0])
streetname
    =
    str(
    i
    [5])
roadtype
    =
    str(
    i
    [27])
speedlimit
    =
    str(
    i
    [31])

else:
    pass

143
if min_height < 30:
    #streetlength = 99
    #streetname = 99  #dummy
    #roadtype = 99  #dummy
    #speedlimit = 99  #dummy
    Min_height = str(min_height)
    Streetname = str(streetname)
    Streetlength = str(streetlength)
    Roadtype = str(roadtype)
    Speedlimit = str(speedlimit)
    list.extend([longitude, latitude, speed
                  , date, hms, GX, GY, Min_height,
                  Streetname, Streetlength, Roadtype,
                  Speedlimit])
    writer.writerow(list)
    #print >> outputFile, ', '.join(row)
    min_height = 99  #dummy
    speedlimit = 99  #dummy

gis_inputFile.close()
gps_inputFile.close()

print InFlnm2

#elapsed_time = time.time() - start
#print("elapsed_time:{0}".format(elapsed_time))

#global filename, InDrPth2, InFlnm2, gi1, gi2, gis1, gis2
nearlink()

A.4.9 Remove data (age is less than 16)

import csv, os
from itertools import islice
import math

title = ['Longitude', 'Latitude', 'Speed (kilometer)', 'DD/MM/YY', 'HH:MM:SS', 'GX', 'GY', 'Min_hight', 'Streetname', 'Streetlength', 'Roadtype', 'Speedlimit (mph)']
def remove():
    inputFile = open(path + '/' + filename, 'rU')
    reader = csv.reader(inputFile)
    TBIFile = open("/Users/toshiyokoo/Desktop/5133/
                      TBIdata_age.csv", 'rU')
    idlist = csv.reader(TBIFile)
    TBIFile.seek(0)
    id = 0
    for row in islice(idlist, 1, None):
        if int(row[3]) == int(tripid):
            root_error ='/Users/toshiyokoo/Desktop
                       /09_result_error'
            outputFile_error = open(os.path.join(              
                root_error, filename[−4] + '.csv'),              
                'a')
            print >> outputFile_error, ',', row.join(title)
        inputFile.seek(0)
        for row in islice(reader, 1, None):
            print >> outputFile_error, ',', row.join(title)
        outputFile_error.close()
        print filename[−4], "age_young"
    id = 1

145
if id == 0:
    root_path = '/Users/toshiyokoo/Desktop/09_result'
    outputFile = open(os.path.join(root_path, filename[:-4] + '.csv'), 'a')
    print >> outputFile, ','.join(title)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile, ','.join(row)
    outputFile.close()
    print filename[:-4], "no_error"
    inputFile.close()

path = "'/Users/toshiyokoo/Desktop/aaafn"
for filename in os.listdir(path):
    global filename
    remove()

A.4.10  Remove speed limit that is 0 mph

# Remove data whose speed limit is 0 km/h.
import csv, os
from itertools import islice

title = ['Longitude', 'Latitude', 'Speed(kilometer)', 'DD/MM/YY', '
    HH:MM:SS', 'GX', 'GY', 'Min_hight', 'Streetname', 'Streetlength 
    ', 'Roadtype', 'Speedlimit(mph)']
def removespeed():
    inputFile = open(path + '/' + filename, 'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    root1_path = "'/Users/toshiyokoo/Desktop/10_result"
root2_path = "/Users/toshiyokoo/Desktop/10_result_error"
count = 0
all_count = 0
for row in islice(reader, 1, None):
    if int(row[11]) > 0:
        count += 1
        all_count += 1
    else:
        all_count += 1

inputFile.seek(0)
if count == 0:
    print filename[−4] + 'is empty'
    outputFile = open(os.path.join(root2_path, filename[−4]+'.csv'), 'a')
    writer = csv.writer(outputFile)
    print >> outputFile, ','.join(title)
    for row in islice(reader, 1, None):
        print >> outputFile, ','.join(row)
    outputFile.close()
elif count == all_count:
    print filename[−4] + 'is full'
    outputFile = open(os.path.join(root1_path, filename[−4]+'.csv'), 'a')
    writer = csv.writer(outputFile)
    print >> outputFile, ','.join(title)
    for row in islice(reader, 1, None):
        print >> outputFile, ','.join(row)
    outputFile.close()
elif 0 < count < all_count:
    print filename[−4] + 'is modified'
outputFile = open(os.path.join(root1_path, filename[4] + '.csv'), 'a')
writer = csv.writer(outputFile)
print >> outputFile, ',', .join(title)
for row in islice(reader, 1, None):
    if int(row[11]) > 0:
        print >> outputFile, ',', .join(row)
outputFile.close()
inputFile.close()

path = "~/Users/tosiyokoo/Desktop/aaafw"
for filename in os.listdir(path):
    global filename
    removespeed()

A.4.11 Create variables for statistical analysis

import csv, os, numpy, math
from itertools import islice
import datetime

title = ['filename', 'tripid', 'date', 'time', 'speed(kilometer)
', 'streetlength', 'speedlimit(mph)', 'speedlimit(km)', 'speed_limit_compliance', 'speed2', 'hour', 'speed_25', 'speed_30', 'speed_35', 'speed_40', 'speed_45', 'speed_50', 'speed_55', 'speed_60', 'speed_65', 'morning', 'afternoon', 'night', 'o_gender', 'o_edu', 'o_age', 'gender', 'education_1', 'education_2', 'age_25_to_34', 'age_35_to_44', 'age_45_to_54', 'age_55_to_64', 'age_65_to_74', 'age_75_to_84', 'age_85_and_over']

def summarize():

148
inputFile = open(path + '/ ' + filename, 'rU')
reader = csv.reader(inputFile)
inputFile.seek(0)
root_path = '/Users/toshiyokoo/Desktop/11_result'
outputFile = open(os.path.join(root_path, filename[:-4] + '.csv'), 'a')
writer = csv.writer(outputFile, lineterminator='
')
print >> outputFile, ',', '.join(title)
for row in islice(reader, 1, None):
    list = []
    Filename = filename[:-4]
    speed = str(row[2])
    date = str(row[3])
    hms = str(row[4])
    streetlength = str(row[9])
    speedlimit = str(row[11])
    drive_speed = float(row[2])
    if date[-4:] == '2011':
        d = datetime.datetime.strptime(date, '%d/%m/%Y')
    else:
        d = datetime.datetime.strptime(date, '%d/%m/%y')

    #convert the unit of speedlimit from mile to kilometer
    mile = float(row[11])
    conv_fac = 1.609344
    kilometer = mile * conv_fac
# calculate speed limit compliance
speed_limit_compliance = drive_speed / kilometer

# whether the speeding or not
if (drive_speed - kilometer) > 0:
speed2 = 1
else:
speed2 = 0

# change the hour from UTC to MN (Time in GPS is Greenwich Mean Time (GMT))
f = datetime.datetime.strptime(hms, '%H:%M:%S')
hour = f.hour
date1 = datetime.datetime(2011, 3, 13)
date2 = datetime.datetime(2011, 11, 5)
# Daylight Saving Time
if date1 <= d <= date2:
    cal_hour = hour - 5
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour = cal_hour + 24
# Standard Time
else:
    cal_hour = hour - 6
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour = cal_hour + 24

# Dummy variable (Speed limit)
num_speedlimit = int(row[11])
speed_25 = 0
speed_30 = 0
speed_35 = 0
speed_40 = 0
speed_45 = 0
speed_50 = 0
speed_55 = 0
speed_60 = 0
speed_65 = 0
if num_speedlimit <= 25:
    speed_25 = 1
elif num_speedlimit == 30:
    speed_30 = 1
elif num_speedlimit == 35:
    speed_35 = 1
elif num_speedlimit == 40:
    speed_40 = 1
elif num_speedlimit == 45:
    speed_45 = 1
elif num_speedlimit == 50:
    speed_50 = 1
elif num_speedlimit == 55:
    speed_55 = 1
elif num_speedlimit == 60:
    speed_60 = 1
elif num_speedlimit == 65:
    speed_65 = 1
elif num_speedlimit == 70:
    dummy = 0
else:
    speed_25 = 99
    speed_30 = 99
    speed_35 = 99
speed_40 = 99  
speed_45 = 99  
speed_50 = 99  
speed_55 = 99  
speed_60 = 99  
speed_65 = 99  

# Dummy variable (morning, afternoon, night)  
morning = 0  
afternoon = 0  
night = 0  
dummy_hour = int(fix_hour)  
if 6 <= dummy_hour < 12:  
    morning = 1  
elif 12 <= dummy_hour < 18:  
    afternoon = 1  
elif 18 <= dummy_hour < 24:  
    night = 1  
elif 0 <= dummy_hour < 6:  
    dummy = 0  
else:  
    morning = 99  
    afternoon = 99  
    night = 99

# Input Household survey data  
# Match Tripid between GPS file and Household survey data  
TBIFile = open("/Users/toshiyokoo/Desktop/5133/TBIdata.csv",'rU')  
idlist = csv.reader(TBIFile)  
TBIFile.seek(0)
gender = 0
age_25_to_34 = 0
age_35_to_44 = 0
age_45_to_54 = 0
age_55_to_64 = 0
age_65_to_74 = 0
age_75_to_84 = 0
age_85_and_over = 0
education_1 = 0
education_2 = 0

for dem in islice(idlist, 1, None):
    if int(dem[3]) == int(tripid):
        # Dummy variable (Gender)
        o_gender = dem[4]
        if dem[4] == 'Male':
            gender = 1
        elif dem[4] == 'Female':
            gender = 0
        else:
            gender = 99

        # Dummy variable (Education)
        o_education = dem[5]
        if dem[5] == 'Daycare / Pre-school':
            dummy = 0
        elif dem[5] == 'Less than high school':
            dummy = 0
        elif dem[5] == 'High school graduate':
            dummy = 0
        elif dem[5] == 'Some college':
            dummy = 0
education_1 = 1
elif dem[5] == 'Vocational/
    Technical training':
    dummy = 0
degree':
    education_1 = 1
elif dem[5] == 'Bachelors
degree':
    education_2 = 1
elif dem[5] == 'Graduate/Post-
    graduate degree':
    education_2 = 1
else:
    education_1 = 99
    education_2 = 99

#Dummy variable (Age)
o_age = dem[6]
if int(dem[6]) == 6:
age_25_to_34 = 1
elif int(dem[6]) == 7:
age_35_to_44 = 1
elif int(dem[6]) == 8:
age_45_to_54 = 1
elif int(dem[6]) == 9:
age_55_to_64 = 1
elif int(dem[6]) == 10:
age_65_to_74 = 1
elif int(dem[6]) == 11:
age_75_to_84 = 1
elif int(dem[6]) == 12:
age_85_and_over = 1
elif int(dem[6]) == 5:
dummy = 0
else:
age_25_to_34 = 99
age_35_to_44 = 99
age_45_to_54 = 99
age_55_to_64 = 99
age_65_to_74 = 99
age_75_to_84 = 99
age_85_and_over = 99

list.extend([Filename, tripid, date, hms, speed, streetlength, speedlimit, kilometer, speed_limit_compliance, speed2, fix_hour, speed_25, speed_30, speed_35, speed_40, speed_45, speed_50, speed_55, speed_60, speed_65, morning, afternoon, night, o_gender, o_education, o_age, gender, education_1, education_2, age_25_to_34, age_35_to_44, age_45_to_54, age_55_to_64, age_65_to_74, age_75_to_84, age_85_and_over])
write.writerow(list)
inputFile.close()
print(filename)

path = "/Users/toshiyokoo/Desktop/aaaif"
for filename in os.listdir(path):
global filename
summarize()

A.4.12 Compile data
Remove data whose speed limit is 0 km/h.

```python
import csv, os
from itertools import islice
title = ['filename', 'tripid', 'date', 'time', 'speed(kilometer)', 'streetlength', 'speedlimit(mph)', 'speedlimit(km)', 'speed_limit_compliance', 'speed2', 'hour', 'speed_25', 'speed_30', 'speed_35', 'speed_40', 'speed_45', 'speed_50', 'speed_55', 'speed_60', 'speed_65', 'morning', 'afternoon', 'night', 'o_gender', 'o_edu', 'o_age', 'gender', 'education_1', 'education_2', 'age_25_to_34', 'age_35_to_44', 'age_45_to_54', 'age_55_to_64', 'age_65_to_74', 'age_75_to_84', 'age_85_and_over']
def compile():
    inputFile = open(path + '/ ' + filename, 'rU')
top = filename[0:2]
top2 = int(top)
    if top2 == 10:
        reader = csv.reader(inputFile)
        inputFile.seek(0)
        for row in islice(reader, 1, None):
            print >> outputFile1, ',', output.join(row)
        inputFile.close()
    elif top2 == 11:
        reader = csv.reader(inputFile)
        inputFile.seek(0)
        for row in islice(reader, 1, None):
            print >> outputFile2, ',', output.join(row)
        inputFile.close()
    elif top2 == 12:
        reader = csv.reader(inputFile)
        inputFile.seek(0)
```

156
for row in islice(reader, 1, None):
    print >> outputFile3, ','.join(row)
inputFile.close()

elif top2 == 13:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile4, ','.join(row)
    inputFile.close()

elif top2 == 14:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile5, ','.join(row)
    inputFile.close()

elif top2 == 15:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile6, ','.join(row)
    inputFile.close()

elif top2 == 16:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile7, ','.join(row)
    inputFile.close()

elif top2 == 17:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile8, ','.join(row)
inputFile.close()

e elif top2 == 18:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader,1,None):
        print >> outputFile9, ',', join(row)
    inputFile.close()

e elif top2 == 19:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader,1,None):
        print >> outputFile10, ',', join(row)
    inputFile.close()

e elif top2 == 20:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader,1,None):
        print >> outputFile11, ',', join(row)
    inputFile.close()

print filename

path = "'/Users/toshiyokoo/Desktop/aaaig"
root_path = "'/Users/toshiyokoo/Desktop/12_result"
outputFile1 = open(os.path.join(root_path,"summary_10")+'.csv', 'a')
outputFile2 = open(os.path.join(root_path,"summary_11")+'.csv', 'a')
outputFile3 = open(os.path.join(root_path,"summary_12")+'.csv', 'a')
outputFile4 = open(os.path.join(root_path,"summary_13")+'.csv', 'a')
outputFile5 = open(os.path.join(root_path, "summary_14") + '.csv', 'a')
outputFile6 = open(os.path.join(root_path, "summary_15") + '.csv', 'a')
outputFile7 = open(os.path.join(root_path, "summary_16") + '.csv', 'a')
outputFile8 = open(os.path.join(root_path, "summary_17") + '.csv', 'a')
outputFile9 = open(os.path.join(root_path, "summary_18") + '.csv', 'a')
outputFile10 = open(os.path.join(root_path, "summary_19") + '.csv', 'a')
outputFile11 = open(os.path.join(root_path, "summary_20") + '.csv', 'a')
writer = csv.writer(outputFile1)
writer = csv.writer(outputFile2)
writer = csv.writer(outputFile3)
writer = csv.writer(outputFile4)
writer = csv.writer(outputFile5)
writer = csv.writer(outputFile6)
writer = csv.writer(outputFile7)
writer = csv.writer(outputFile8)
writer = csv.writer(outputFile9)
writer = csv.writer(outputFile10)
writer = csv.writer(outputFile11)
print >> outputFile1, ','.join(title)
print >> outputFile2, ','.join(title)
print >> outputFile3, ','.join(title)
print >> outputFile4, ','.join(title)
print >> outputFile5, ','.join(title)
print >> outputFile6, ','.join(title)
print >> outputFile7, ','.join(title)
print >> outputFile8, ','.join(title)
print >> outputFile9, ','.join(title)
print >> outputFile10, ','.join(title)
print >> outputFile11, ','.join(title)
for filename in os.listdir(path):
    global filename
    compile()
outputFile1.close()
outputFile2.close()
outputFile3.close()
outputFile4.close()
outputFile5.close()
outputFile6.close()
outputFile7.close()
outputFile8.close()
outputFile9.close()
outputFile10.close()
outputFile11.close()

A.4.13  t-test: time

# compute t-test and average

import csv, os, pandas
import pandas as pd
from scipy import stats
from itertools import islice
import math
import numpy

path = "~/Users/toshyokoo/Desktop/aaals"
count = 1
for filename in os.listdir(path):
if count == 1:
    data = pandas.read_csv(path + '/ ' + filename)
    am0 = data[data['hour'] == 0][
        'speed_limit_compliance'
    ]
    am1 = data[data['hour'] == 1][
        'speed_limit_compliance'
    ]
    am2 = data[data['hour'] == 2][
        'speed_limit_compliance'
    ]
    am3 = data[data['hour'] == 3][
        'speed_limit_compliance'
    ]
    am4 = data[data['hour'] == 4][
        'speed_limit_compliance'
    ]
    am5 = data[data['hour'] == 5][
        'speed_limit_compliance'
    ]
    am6 = data[data['hour'] == 6][
        'speed_limit_compliance'
    ]
    am7 = data[data['hour'] == 7][
        'speed_limit_compliance'
    ]
    am8 = data[data['hour'] == 8][
        'speed_limit_compliance'
    ]
    am9 = data[data['hour'] == 9][
        'speed_limit_compliance'
    ]
    am10 = data[data['hour'] == 10][
        'speed_limit_compliance'
    ]
    am11 = data[data['hour'] == 11][
        'speed_limit_compliance'
    ]
    pm12 = data[data['hour'] == 12][
        'speed_limit_compliance'
    ]
    pm1 = data[data['hour'] == 13][
        'speed_limit_compliance'
    ]
    pm2 = data[data['hour'] == 14][
        'speed_limit_compliance'
    ]
pm3 = data[data['hour'] == 15]['speed_limit_compliance']

pm4 = data[data['hour'] == 16]['speed_limit_compliance']

pm5 = data[data['hour'] == 17]['speed_limit_compliance']

pm6 = data[data['hour'] == 18]['speed_limit_compliance']

pm7 = data[data['hour'] == 19]['speed_limit_compliance']

pm8 = data[data['hour'] == 20]['speed_limit_compliance']

pm9 = data[data['hour'] == 21]['speed_limit_compliance']

pm10 = data[data['hour'] == 22]['speed_limit_compliance']

pm11 = data[data['hour'] == 23]['speed_limit_compliance']

print filename + "1"

else:

    data = pandas.read_csv(path + '/ ' + filename)

    add0 = data[data['hour'] == 0]['speed_limit_compliance']

    add1 = data[data['hour'] == 1]['speed_limit_compliance']

    add2 = data[data['hour'] == 2]['speed_limit_compliance']

    add3 = data[data['hour'] == 3]['speed_limit_compliance']

    add4 = data[data['hour'] == 4]['speed_limit_compliance']
add5 = data[data['hour'] == 5]['speed_limit_compliance']
add6 = data[data['hour'] == 6]['speed_limit_compliance']
add7 = data[data['hour'] == 7]['speed_limit_compliance']
add8 = data[data['hour'] == 8]['speed_limit_compliance']
add9 = data[data['hour'] == 9]['speed_limit_compliance']
add10 = data[data['hour'] == 10]['speed_limit_compliance']
add11 = data[data['hour'] == 11]['speed_limit_compliance']
pdd12 = data[data['hour'] == 12]['speed_limit_compliance']
pdd1 = data[data['hour'] == 13]['speed_limit_compliance']
pdd2 = data[data['hour'] == 14]['speed_limit_compliance']
pdd3 = data[data['hour'] == 15]['speed_limit_compliance']
pdd4 = data[data['hour'] == 16]['speed_limit_compliance']
pdd5 = data[data['hour'] == 17]['speed_limit_compliance']
pdd6 = data[data['hour'] == 18]['speed_limit_compliance']
pdd7 = data[data['hour'] == 19]['speed_limit_compliance']
pdd8 = data[data['hour'] == 20]['speed_limit_compliance']
pdd9 = data [data ['hour'] == 21]['speed_limit_compliance']
pdd10 = data [data ['hour'] == 22]['speed_limit_compliance']
pdd11 = data [data ['hour'] == 23]['speed_limit_compliance']
am0 = pd.concat([am0, add0])
am1 = pd.concat([am1, add1])
am2 = pd.concat([am2, add2])
am3 = pd.concat([am3, add3])
am4 = pd.concat([am4, add4])
am5 = pd.concat([am5, add5])
am6 = pd.concat([am6, add6])
am7 = pd.concat([am7, add7])
am8 = pd.concat([am8, add8])
am9 = pd.concat([am9, add9])
am10 = pd.concat([am10, add10])
am11 = pd.concat([am11, add11])
pm12 = pd.concat([pm12, pdd12])
pm1 = pd.concat([pm1, pdd1])
pm2 = pd.concat([pm2, pdd2])
pm3 = pd.concat([pm3, pdd3])
pm4 = pd.concat([pm4, pdd4])
pm5 = pd.concat([pm5, pdd5])
pm6 = pd.concat([pm6, pdd6])
pm7 = pd.concat([pm7, pdd7])
pm8 = pd.concat([pm8, pdd8])
pm9 = pd.concat([pm9, pdd9])
pm10 = pd.concat([pm10, pdd10])

if count == 11:
    print filename + "2"
print "am4, am7", stats.ttest_ind(am4, am7, equal_var = False)
print "pm12, pm2", stats.ttest_ind(pm12, pm2, equal_var = False)
print numpy.mean(am0)
print numpy.mean(am1)
print numpy.mean(am2)
print numpy.mean(am3)
print numpy.mean(am4)
print numpy.mean(am5)
print numpy.mean(am6)
print numpy.mean(am7)
print numpy.mean(am8)
print numpy.mean(am9)
print numpy.mean(am10)
print numpy.mean(am11)
print numpy.mean(pm12)
print numpy.mean(pm1)
print numpy.mean(pm2)
print numpy.mean(pm3)
print numpy.mean(pm4)
print numpy.mean(pm5)
print numpy.mean(pm6)
print numpy.mean(pm7)
print numpy.mean(pm8)
print numpy.mean(pm9)
print numpy.mean(pm10)
print numpy.mean(pm11)
count += 1

A.4.14 t-test: gender

# compute t-test and average
import csv, os, pandas
import pandas as pd
from scipy import stats
from itertools import islice
import math
import numpy

path = "~/Users/toshiyokoo/Desktop/aaals"
count = 1
for filename in os.listdir(path):
    if count == 1:
        data = pandas.read_csv(path + '/' +filename)
        female = data[data['gender'] == 0]['speed_limit_compliance']
        male = data[data['gender'] == 1]['speed_limit_compliance']
        print filename + "1"
    else:
        data = pandas.read_csv(path + '/' +filename)
        add1 = data[data['gender'] == 0]['speed_limit_compliance']
        add2 = data[data['gender'] == 1]['speed_limit_compliance']
        female = pd.concat([female, add1])
        male = pd.concat([male, add2])
        print filename + "2"

    if count == 11:
        print stats.ttest_ind(female, male, equal_var = False)
        print numpy.mean(female)
A.4.15 t-test: age

# extract file that match tripid of TBI data

```python
import csv, os, pandas
import pandas as pd
from scipy import stats
from itertools import islice
import math
import numpy

path = "Users/tosiyokoo/Desktop/aaals"
count = 1
for filename in os.listdir(path):
    if count == 1:
        data = pandas.read_csv(path + '/' +filename)
        age5 = data[data['o_age'] == 5]['speed_limit_compliance']
        age6 = data[data['o_age'] == 6]['speed_limit_compliance']
        age7 = data[data['o_age'] == 7]['speed_limit_compliance']
        age8 = data[data['o_age'] == 8]['speed_limit_compliance']
        age9 = data[data['o_age'] == 9]['speed_limit_compliance']
        age10 = data[data['o_age'] == 10]['speed_limit_compliance']
        age11 = data[data['o_age'] == 11]['speed_limit_compliance']
```

age12 = data[data['o_age'] == 12][
    'speed_limit_compliance']
print filename + "1"

else:
    data = pandas.read_csv(path + '/' + filename)
    add5 = data[data['o_age'] == 5]['
        speed_limit_compliance']
    add6 = data[data['o_age'] == 6]['
        speed_limit_compliance']
    add7 = data[data['o_age'] == 7]['
        speed_limit_compliance']
    add8 = data[data['o_age'] == 8]['
        speed_limit_compliance']
    add9 = data[data['o_age'] == 9]['
        speed_limit_compliance']
    add10 = data[data['o_age'] == 10]['
        speed_limit_compliance']
    add11 = data[data['o_age'] == 11]['
        speed_limit_compliance']
    add12 = data[data['o_age'] == 12]['
        speed_limit_compliance']
    age5 = pd.concat([age5, add5])
    age6 = pd.concat([age6, add6])
    age7 = pd.concat([age7, add7])
    age8 = pd.concat([age8, add8])
    age9 = pd.concat([age9, add9])
    age10 = pd.concat([age10, add10])
    age11 = pd.concat([age11, add11])
    age12 = pd.concat([age12, add12])
    print filename + "2"

if count == 11:
print "age6, age7", stats.ttest_ind(age6, age7, equal_var = False)
print "age7, age8", stats.ttest_ind(age7, age8, equal_var = False)
print numpy.mean(age5)
print numpy.mean(age6)
print numpy.mean(age7)
print numpy.mean(age8)
print numpy.mean(age9)
print numpy.mean(age10)
print numpy.mean(age11)
print numpy.mean(age12)

count += 1

A.4.16 t-test: education

# extract file that match tripid of TBI data

import csv, os, pandas
import pandas as pd
from scipy import stats
from itertools import islice
import math
import numpy

path = "/Users/toshiyokoo/Desktop/aaals"
count = 1
for filename in os.listdir(path):
    if count == 1:
        data = pandas.read_csv(path + '/' + filename)
        asso = data[data["o_edu"] == 'Associates degree ']["speed_limit_compliance "]
some = data[data['o_edu'] == 'Some college'][
    'speed_limit_compliance']
high = data[data['o_edu'] == 'High school graduate'][
    'speed_limit_compliance']
voca = data[data['o_edu'] == 'Vocational/
    Technical training'][
    'speed_limit_compliance']
bach = data[data['o_edu'] == 'Bachelors degree'][
    'speed_limit_compliance']
grad = data[data['o_edu'] == 'Graduate/Post-
    graduate degree'][
    'speed_limit_compliance']
print filename + "1"
else:
    data = pandas.read_csv(path + '/ ' + filename)
add0 = data[data['o_edu'] == 'Associates degree'][
    'speed_limit_compliance']
add1 = data[data['o_edu'] == 'Some college'][
    'speed_limit_compliance']
add2 = data[data['o_edu'] == 'High school graduate'][
    'speed_limit_compliance']
add3 = data[data['o_edu'] == 'Vocational/
    Technical training'][
    'speed_limit_compliance']
add4 = data[data['o_edu'] == 'Bachelors degree'][
    'speed_limit_compliance']
add5 = data[data['o_edu'] == 'Graduate/Post-
    graduate degree'][
    'speed_limit_compliance']
asso = pd.concat([asso, add0])
some = pd.concat([some, add1])
high = pd.concat([high, add2])
voca = pd.concat([voca, add3])
bach = pd.concat([bach, add4])
170
grad = pd.concat([grad, add5])

print filename + "2"

if count == 11:
    print "high, some", stats.ttest_ind(high, some, equal_var = False)
    print "grad, some", stats.ttest_ind(grad, some, equal_var = False)
    print numpy.mean(asso)
    print numpy.mean(bach)
    print numpy.mean(grad)
    print numpy.mean(high)
    print numpy.mean(some)
    print numpy.mean(voca)

    count += 1

A.4.17 Speeding difference depending on driver

#inputfile is from csv file after the algorithm 10_removeSL0km

import csv, os, numpy, math
from itertools import islice

title = ['filename', 'tripid', 'date', 'time', 'speed(kilometer )', 'streetlength', 'speedlimit(mph)', 'speedlimit(km)', 'speed_limit_compliance', 'speed2']

path = "\Users\toshiyokoo\Desktop\aaahu"
newname = 99

for filename in os.listdir(path):
    inputFile = open(path + '/' + filename, 'rU')
    root_path = '\Users\toshiyokoo\Desktop\14_result'
    outputFile = open(os.path.join(root_path, filename[0:12] + '.csv'), 'a')
writer = csv.writer(outputFile, lineterminator = '\n')
Filename = filename[: -4]
tripid = int(trip)
list = []
if tripid == newname:
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        speed_km = str(row[2])
date = str(row[3])
time = str(row[4])
street_length = str(row[9])
speedlimit_mph = str(row[11])
drive_speed = float(row[2])
#convert the unit of speedlimit from mile to kilometer
mile = float(row[11])
conv_fac = 1.609344
speedlimit_km = mile * conv_fac

#calculate speed limit compliance
speed_limit_compliance = drive_speed / speedlimit_km

#whether the speeding or not
if (drive_speed - speedlimit_km) > 0:
    speed2 = 1
else:
    speed2 = 0
```python
list.extend([Filename, trip, date, time, speed_km, street_length, speedlimit_mph, speedlimit_km, speed_limit_compliance, speed2])
writer.writerow(list)
list = []

newname = tripid
inputFile.close()
outputFile.close()
print filename, "1"

else:
    print >> outputFile, ','.join(title)
reader = csv.reader(inputFile)
inputFile.seek(0)
for row in islice(reader, 1, None):
    speed_km = str(row[2])
    date = str(row[3])
    time = str(row[4])
    street_length = str(row[9])
    speedlimit_mph = str(row[11])
    drive_speed = float(row[2])
    #convert the unit of speed limit from mile to kilometer
    mile = float(row[11])
    conv_fac = 1.609344
    speedlimit_km = mile * conv_fac
    #calculate speed limit compliance
    speed_limit_compliance = drive_speed / speedlimit_km
    
```
#whether the speeding or not
if (drive_speed - speedlimit_km) > 0:
    speed2 = 1
else:
    speed2 = 0

list.extend([Filename, trip, date, time, speed_km, street_length, speedlimit_mph, speedlimit_km, speed_limit_compliance, speed2])
writer.writerow(list)
list = []

newname = tripid
inputFile.close()
outputFile.close()
print filename, "2"

import csv, os
from itertools import islice
title = ['tripid', 'total_time', 'total_speeding', 'percentage', 'gender', 'education', 'age']
def howmuch():
    inputFile = open(path + '/' + filename, 'rU')
    reader = csv.reader(inputFile)
    count = 0
    num_speed = 0
    list = []
    for row in islice(reader, 1, None):
        for
count += 1
tripid = int(row[1])
speeding = int(row[9])
if speeding == 1:
    num_speed += 1

if count == 1:
    #Input Household survey data
    #Match Tripid between GPS file and
    #Household survey data
    TBIFile = open("/Users/toshiyokoo/
    Desktop/5133/TBIdata.csv",'rU")
    idlist = csv.reader(TBIFile)
    TBIFile.seek(0)
    for dem in islice(idlist, 1, None):
        if int(dem[3]) == int(tripid):
            o_gender = dem[4]
            o_education = dem[5]
            o_age = dem[6]

    time = count
    speed = num_speed
    total_time = float(time)
    total_speeding = float(speed)

    #percentage of speeding depends on individuals
    percentage = float(total_speeding / total_time)

    list.extend([tripid, total_time, total_speeding,
                 percentage, o_gender, o_education, o_age])
    writer.writerow(list)
    list = []
inputFile.close()
print filename

path = "/Users/toshiyokoo/Desktop/aaaie"
root_path = '/Users/toshiyokoo/Desktop/15_result'
outputFile = open(os.path.join(root_path,"summary_percentage")
+'.csv', 'a')
writer = csv.writer(outputFile, lineterminator='
')
print >> outputFile, ',', '.join(title)
for filename in os.listdir(path):
    global filename
    howmuch()
outputFile.close()

A.4.18 Fixed effects estimation with integrated GPS points

#load GPS
#— 1 Set pathname here
InDrPth1='Users/toshiyokoo/Desktop/aaadq/

#— 2 Set pathname here
InDrPth2='Users/toshiyokoo/Desktop/aaadr/

#execute algorithm_sub0
execfile('Users/toshiyokoo/Desktop/Matching/sub0_matching.py')

import csv, os, numpy, math
from itertools import islice
import processing
import time

def matching():
    #load GIS map
#— 1 Load map layer
InGIS = "/Users/toshiyokoo/Desktop/Buffer0318/10
    m_buffer.shp"
vlayer = QgsVectorLayer(InGIS, "10_m_buffer", "ogr")

#— 2 Confirm something is loaded and valid
vlayer.isValid()

#— 3 Display the layer into QGIS
QgsMapLayerRegistry.instance().addMapLayer(vlayer)

#Part1
#load GPS data

#— 2 Set file name here
InFlnm1=filename

#— 3 Build file name an path for uri
InFlPth1="file:///"+InDrPth1+InFlnm1
#gps_File = open(InDrPth1+InFlnm1,'rU')
#gps_File.seek(0)

#— 4 Set import Sting here note only need to set x
    and y other come for free
uri1 = InFlPth1+"?delimiter=%s&xField=%s&yField=%s" %
    ("","GX","GY")
# "Longitude", "Latitude" → "GX", "GY"

#— 5 Load point layer
bh1 = QgsVectorLayer(uri1, InFlnm1, "delimitedtext")
#— 6 Confirm something is loaded and valid
bh1.isValid()

#— 7 Set CRS
bh1.setCrs(QgsCoordinateReferenceSystem(26915,
        QgsCoordinateReferenceSystem.EpsgCrsId))
#4326: WGS84, 26915: NAD83/UTM zone 15N
#— 8 Display the layer into QGIS
QgsMapLayerRegistry.instance().addMapLayer(bh1)
#— select by location
processing.runalg("qgis:selectbylocation", vlayer, uri1, 0)
#— save vector layer as csv (True: selected, False: all)
cLayer = qgis.utils.iface.mapCanvas().layers()[1]
gi1 = 'GIS1.'+InFlnm1
gis1 = 'Users/toshiyokoo/Desktop/5536/'+gi1
QgsVectorFileWriter.writeAsVectorFormat(cLayer, gis1, "CP1250", None, "CSV", True)
global InDrPth2, gis1, InFlnm1
#execute algorithm_sub1
#remove intersection & missing link GPS data
execfile('Users/toshiyokoo/Desktop/Matching/sub1_remove_intersect.py')
#Part2
#load GPS data
#— 1 Set file name here
InFlnm2=InFlnm1
#— 2 Set pathname here
#InDrPth2='Users/toshiyokoo/Desktop/5880/'
#— 3 Build file name an path for uri
InFlPth2="file:///"+InDrPth2+InFlnm2
#— 4 Set import Sting here note only need to set x and y other come for free
uri2 = InFlPth2+"?delimiter=%s&xField=%s&yField=%s" % (",", "GX", "GY")
# "Longitude", "Latitude" -> "GX", "GY"
#— 5 Load point layer
bh2 = QgsVectorLayer(uri2, InFlnm2, "delimitedtext")
6 Confirm something is loaded and valid
bh2.isValid()

7 Set CRS
bh2.setCrs(QgsCoordinateReferenceSystem(26915,
QgsCoordinateReferenceSystem.EpsgCrsId))
#4326: WGS84, 26915: NAD83/UTM zone 15N

8 Display the layer into QGIS
QgsMapLayerRegistry.instance().addMapLayer(bh2)

select by location
processing.runalg("qgis:selectbylocation", vlayer, uri2, 0)

save vector layer as csv (True: selected, False: all)
cLayer = qgis.utils.iface.mapCanvas().layers()[2]

gi2 = 'GIS2.'+filename
gis2 = 'Users/toshiyokoo/Desktop/5536/'+gi2

global gis2, InFlnm2
QgsVectorFileWriter.writeAsVectorFormat(cLayer, gis2, "CP1250", None, "CSV", True)

execute algorithm_sub2
# find nearestlink between GPS point and GIS map
# calculate minimum height
# link nearest (Streetname and Speedlimit)
execfile('Users/toshiyokoo/Desktop/Matching/
sub2_matching.py')

remove GPS layer
QgsMapLayerRegistry.instance().removeMapLayer(bh1.id())
QgsMapLayerRegistry.instance().removeMapLayer(bh2.id())
QgsMapLayerRegistry.instance().removeMapLayer(vlayer.id())
#reset selected layer
#cLayer.setSelectedFeatures([0])
#—— 1 Set file name here
InFlnm3=InFlnm2
InDrPth3='/Users/toshiyokoo/Desktop/box/'+InFlnm3
#move file
os.rename(gis1, '/Users/toshiyokoo/Desktop/comp1/'+gi1)
os.rename(gis2, '/Users/toshiyokoo/Desktop/comp1/'+gi2)
os.rename(InDrPth3, '/Users/toshiyokoo/Desktop/comp1
"/"+InFlnm3)
#gps_File.close()

for filename in os.listdir(InDrPth1):
    #global filename, InFlnm2, gi1, gi2, gis1, gis2
    matching()

# remove intersection & missing link GPS data

import csv, os, numpy, math
from itertools import islice
import time

#start = time.time()

global title1
title1 = ['Longitude', 'Latitude', 'Speed(kilometer)', 'Course (degrees)', 'NumberOfSatellites', 'HDOP', 'Altitude (meters)', 'DD/MM/YY', 'HH:MM:SS', 'Distance (meters)', 'GX', 'GY']

def intersect():
    gps_inputFile1 = open(InDrPth1 +InFlnm1, 'rU')
gps_reader1 = csv.reader(gps_inputFile1)
gps_inputFile1.seek(0)
root_path1 = InDrPth2
outputFile1 = open(os.path.join(root_path1, InFlnm1
[-4]+'.csv'), 'a')
writer1 = csv.writer(outputFile1, lineterminator='\n')
print >> outputFile1, ',', join(title1)
gis_inputFile1 = open(gis1, 'rU')
gis_reader1 = csv.reader(gis_inputFile1)
min_height1 = 99 #dummy
speedlimit1 = 99 #dummy
for row1 in islice(gis_reader1, 1, None):
    list1 = []
x_gps1 = float(row1[10])
y_gps1 = float(row1[11])
gis_inputFile1.seek(0)
count1 = 0
for j in islice(gis_reader1, 1, None):
    direction1 = float(j[42])
ox_gis1 = float(j[43]) #OX
oy_gis1 = float(j[44]) #OY
dx_gis1 = float(j[45]) #DX
dy_gis1 = float(j[46]) #DY
if direction1 == 1:
    if (ox_gis1 - 10 <= x_gps1 <=
dx_gis1 + 10):
        if (oy_gis1 - 10) <=
y_gps1 <= (dy_gis1 +
10):
            count1 += 1
        elif (dy_gis1 - 10) <=
y_gps1 <= (oy_gis1 +
10):
        else:
            pass
181
count1 += 1
elif (dx_gis1 - 10 <= x_gps1 <= ox_gis1 + 10):
    if (oy_gis1 - 10) <= y_gps1 <= (dy_gis1 + 10):
        count1 += 1
elif (dy_gis1 - 10) <= y_gps1 <= (oy_gis1 + 10):
    count1 += 1
if count1 == 1:
    print >> outputFile1, ','.join(row1)

gis_inputFile1.close()
gps_inputFile1.close()
print InFlnm1
#elapsed_time = time.time() - start
#print("elapsed_time:{0}".format(elapsed_time))

#global filename, InDrPth2, InFlnm1, InFlnm2, gi1, gi2, gis1, gis2
intersect()

# find nearestlink between GPS point and GIS map
# calculate minimum height
# link nearest(Streetname and Speedlimit)

import csv, os, numpy, math
from itertools import islice
import time

182
#start = time.time()

global title2
title2 = ['Longitude', 'Latitude', 'Speed(kilometer)', 'DD/MM/YY', 'HH:MM:SS', 'GX', 'GY', 'Min_height', 'Streetname', 'Streetlength', 'Roadtype', 'Speedlimit(mph)', 'LinkID']

def nearlink():
    gps_inputFile = open(InDrPth2 +InFlnm2, 'rU')
gps_reader = csv.reader(gps_inputFile)
gps_inputFile.seek(0)
root_path = '/Users/toshiyokoo/Desktop/box'
outputFile = open(os.path.join(root_path, InFlnm2[:4]+'.csv'), 'a')
writer = csv.writer(outputFile, lineterminator='
')
print >> outputFile, ','.join(title2)
gis_inputFile = open(gis2, 'rU')
gis_reader = csv.reader(gis_inputFile)
min_height = 99 #dummy
speedlimit = 99 #dummy
for row in islice(gps_reader, 1, None):
    list = []
x_gps = float(row[10])
y_gps = float(row[11])
longitude = str(row[0])
latitude = str(row[1])
speed = str(row[2])
date = str(row[7])
hms = str(row[8])
GX = str(row[10])
GY = str(row[11])
gis_inputFile.seek(0)
for i in islice(gis_reader, 1, None):
    direction = float(i[42])
    ox_gis = float(i[43]) #OX
    oy_gis = float(i[44]) #OY
    dx_gis = float(i[45]) #DX
    dy_gis = float(i[46]) #DY
    if direction == 1:
        if (ox_gis <= x_gps <= dx_gis):
            if (oy_gis - 30) <=
                y_gps <= (oy_gis +
                          30):
                length = ((
                    ox_gis -
                    dx_gis) ** 2
                + (oy_gis -
                    dy_gis) **
                2) ** 0.5
        gis_o = numpy.
        array([[ox_gis ,
                oy_gis]])
        gis_d = numpy.
        array([[dx_gis ,
                dy_gis]])
        gps = numpy.
        array([[x_gps
                , y_gps]])
        v1 = gps -
        gis_o
\[ v2 = \text{gis}\_d - \text{gis}\_o \]
Area = numpy.
cross(v1, v2)
D = abs(Area)
height = D /
length
if
min_height >
height:
min_height =
height

streetlength =
str(i
[0])
streetname =
str(i
[5])
roadtype =
str(i
[27])
speedlimit
= str(
    i
    [31])

linkID
= str(
    i
    [41])

# print
'1',

min_height

e if (oy_gis <= y_gps
<= dy_gis):      #
road x–y & curve
    length = ((
        ox_gis -
        dx_gis) ** 2
    + (oy_gis -
        dy_gis) **
    2) ** 0.5
gis_o = numpy.
array([[
    ox_gis,
    oy_gis]])
gis_d = numpy.array([dx_gis, dy_gis])
gps = numpy.array([x_gps, y_gps])

v1 = gps - gis_o
v2 = gis_d - gis_o

Area = numpy.cross(v1, v2)
D = abs(Area)
height = D / length

if min_height > height:
    min_height = height

streetlength = str(i[0])
if (dy_gis <= y_gps <= oy_gis):
    # road x–y & curve length
    length = (((ox_gis – dx_gis) ** 2
               + (oy_gis – dy_gis) **
               2) ** 0.5
gis_o = numpy.array([[ox_gis, oy_gis]])
gis_d = numpy.array([[dx_gis, dy_gis]])
gps = numpy.array([[x_gps, y_gps]])
v1 = gps - gis_o
v2 = gis_d - gis_o
Area = numpy.cross(v1, v2)
D = abs(Area)
height = D / length
if min_height > height:
    min_height = height
streetlength
    =
    str(i
    [0])
streetname
    =
    str(i
    [5])
roadtype
    =
    str(i
    [27])
speedlimit
    =
    str(i
    [31])
linkID
    =
    str(i
    [41])

    elif (dx_gis <= x_gps <= ox_gis

    ):
if (oy_gis - 30) <=
y_gps <= (oy_gis +
30):
    length = ((
ox_gis -
dx_gis) ** 2
+ (oy_gis -
dy_gis) **
2) ** 0.5
gis_o = numpy.
    array([[
ox_gis ,
    oy_gis]])
gis_d = numpy.
    array([[ dx_gis ,
dy_gis]])
gps = numpy.
    array([[x_gps
    ,y_gps]])
v1 = gps -
gis_o
ev2 = gis_d -
gis_o
Area = numpy.
    cross(v1, v2 )
D = abs(Area)
height = D /
length
if
    min_height >
    height:
        min_height
        =
        height
    streetlength
        =
        str(i
           [0])
    streetname
        =
        str(i
           [5])
    roadtype
        =
        str(i
           [27])
    speedlimit
        =
        str(i
           [31])
linkID
    =
    str(
        i
        [41])

    #print
    '1',

    min_height

    elif (oy_gis <= y_gps
            <= dy_gis):
        #
        road x-y & curve
        length = ((
            ox_gis -
            dx_gis) ** 2
            + (oy_gis -
            dy_gis) **
            2) ** 0.5
        gis_o = numpy.
        array([[ox_gis ,
                oy_gis ]])
        gis_d = numpy.
        array([[dx_gis ,
                dy_gis ]])
        gps = numpy.
        array([[x_gps ,
                y_gps ]])
v1 = gps - gis_o
v2 = gis_d - gis_o
Area = numpy. cross(v1, v2)
D = abs(Area)
height = D / length
if min_height > height:
    min_height = height

streetlength = str(i[0])
streetname = str(i[5])
roadtype = str(i[27])

speedlimit = str(i[31])

linkID = str(i[41])

elif (dy_gis <= y_gps <= oy_gis):
    # road x-y & curve
    length = (((ox_gis - dx_gis) ** 2
                + (oy_gis - dy_gis) **
                2) ** 0.5

    gis_o = numpy.array([ox_gis,
                        oy_gis])
gis_d = numpy.array([[dx_gis, dy_gis]])
gps = numpy.array([x_gps, y_gps])
v1 = gps - gis_o
v2 = gis_d - gis_o
Area = numpy.cross(v1, v2)
D = abs(Area)
height = D / length
if min_height > height:
    min_height = height
streetlength = str(i[0])
streetname
    =
    str(i[5])
roadtype
    =
    str(i[27])
speedlimit
    =
    str(i[31])
linkID
    =
    str(i[41])

elif (oy_gis <= y_gps <= dy_gis):
    if (ox_gis - 30) <=
        x_gps <= (ox_gis +
            30):
        #road y-axis, interstate
        range 50m


```python
length = ((
    ox_gis -
    dx_gis) ** 2
+ (oy_gis -
    dy_gis) ** 2) ** 0.5

gis_o = numpy.
    array([[ox_gis,
    oy_gis]])
gis_d = numpy.
    array([[dx_gis,
    dy_gis]])
gps = numpy.
    array([[x_gps,
    y_gps]])
v1 = gps -
    gis_o
v2 = gis_d -
    gis_o
Area = numpy.
    cross(v1, v2)
D = abs(Area)
height = D /
    length
if
    min_height >
    height:
```
min_height
    =
    height

streetlength
    =
    str(i
    [0])
streetname
    =
    str(i
    [5])
roadtype
    =
    str(i
    [27])
speedlimit
    =
    str(i
    [31])
linkID
    =
    str(i
    [41])
elif (dy_gis <= y_gps <= oy_gis):
    if (ox_gis - 30) <=
        x_gps <= (ox_gis +
            30):  # road y-
        axis, interstate
        range 50m
            length = ((
                ox_gis -
                dx_gis) ** 2
            + (oy_gis -
                dy_gis) **
            2) ** 0.5
    gis_o = numpy.
        array([[
                ox_gis ,
                oy_gis]])
    gis_d = numpy.
        array([[
                dx_gis ,
                dy_gis]])
    gps = numpy.
        array([[x_gps
                , y_gps]])
    v1 = gps -
        gis_o
    v2 = gis_d -
        gis_o
    Area = numpy.
        cross(v1, v2
    )
    D = abs(Area)
height = D / length
if
   min_height > height:
      min_height = height

streetlength = str(i[0])
streetname = str(i[5])
roadtype = str(i[27])
speedlimit = str(i[31])
linkID = str(
    i
[41])

else:
    pass

if min_height < 30:
    #streetlength = 99
    #streetname = 99  # dummy
    #roadtype = 99    # dummy
    #speedlimit = 99  # dummy
    Min_height = str(min_height)
    Streetname = str(streetname)
    Streetlength = str(streetlength)
    Roadtype = str(roadtype)
    Speedlimit = str(speedlimit)
    LinkID = str(linkID)
    list.extend([longitude, latitude, speed
                  , date, hms, GX, GY, Min_height,
                  Streetname, Streetlength, Roadtype,
                  Speedlimit, LinkID])
    writer.writerow(list)
    #print >> outputFile, ','.join(row)
    min_height = 99  # dummy
    speedlimit = 99  # dummy

gis_inputFile.close()
gps_inputFile.close()

print InFlnm2
#elapsed_time = time.time() - start
# print("elapsed_time:{0}".format(elapsed_time))

global filename, InDrPth2, InFlnm2, gi1, gi2, gis1, gis2

def nearlink():
    # extract file that match tripid of TBI data

    import csv, os
    from itertools import islice
    import math

    title = ['Longitude', 'Latitude', 'Speed(kilometer)', 'DD/MM/YY', '
             HH:MM:SS', 'GX', 'GY', 'Min_hight', 'Streetname', 'Streetlength'
             , 'Roadtype', 'Speedlimit(mph)', 'LinkID']

    def remove():
        inputFile = open(path + '/'+filename,'rU')
        reader = csv.reader(inputFile)
        TBIFile = open("/Users/toshiyokoo/Desktop/5133/
                         TBIdata_age.csv","rU")
        idlist = csv.reader(TBIFile)
        TBIFile.seek(0)
        id = 0
        for row in islice(idlist, 1, None):
            if int(row[3]) == int(tripid):
                root_error = '/Users/toshiyokoo/Desktop
                       /09_result_error'}
outputFile_error = open(os.path.join(root_error, filename[:-4] + '.csv'), 'a')
print >> outputFile_error, ','.join(title)
inputFile.seek(0)
for row in islice(reader, 1, None):
    print >> outputFile_error, ','.join(row)
outputFile_error.close()
print filename[:-4], "age_young"

id = 1

if id == 0:
    root_path = '/Users/toshiyokoo/Desktop/09_result'
    outputFile = open(os.path.join(root_path, filename[:-4] + '.csv'), 'a')
    print >> outputFile, ','.join(title)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile, ','.join(row)
    outputFile.close()
    print filename[:-4], "no_error"
    inputFile.close()

path = "'/Users/toshiyokoo/Desktop/aaaef"
for filename in os.listdir(path):
    global filename
    remove()

# Remove data whose speed limit is 0 km/h.
import csv, os
from itertools import islice

title = ['Longitude', 'Latitude', 'Speed (kilometer)', 'DD/MM/YY', 'HH:MM:SS', 'GX', 'GY', 'Min_hight', 'Streetname', 'Streetlength', 'Roadtype', 'Speedlimit (mph)', 'LinkID']
def removespeed():
    inputFile = open(path + '/'+filename, 'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    root1_path = r'/Users/toshiyokoo/Desktop/10_result'
    root2_path = r'/Users/toshiyokoo/Desktop/10_result_error'
    count = 0
    all_count = 0
    for row in islice(reader, 1, None):
        if int(row[11]) > 0:
            count += 1
            all_count += 1
        else:
            all_count += 1
    inputFile.seek(0)
    if count == 0:
        print filename[-4] + ' is empty'
        outputFile = open(os.path.join(root2_path,
                                        filename[-4]+'.csv'), 'a')
        writer = csv.writer(outputFile)
        print >> outputFile, ','.join(title)
        for row in islice(reader, 1, None):
            print >> outputFile, ','.join(row)
        outputFile.close()
elif count == all_count:
    print filename[−4] + ' is full'
outputFile = open(os.path.join(root1_path,
    filename[−4]+'.csv'),'a')
writer = csv.writer(outputFile)
print >> outputFile, ','.join(title)
for row in islice(reader,1,None):
    print >> outputFile, ','.join(row)
outputFile.close()

elif 0 < count < all_count:
    print filename[−4] + ' is modified'
outputFile = open(os.path.join(root1_path,
    filename[−4]+'.csv'),'a')
writer = csv.writer(outputFile)
print >> outputFile, ','.join(title)
for row in islice(reader,1,None):
    if int(row[11]) > 0:
        print >> outputFile, ','.join(row)
outputFile.close()

inputFile.close()

path ='/Users/toshiyokoo/Desktop/aaaeg'
for filename in os.listdir(path):
    global filename
    removespeed()

import csv, os, numpy, math
from itertools import islice
import datetime

206
title = ['PersonID', 'TripID', 'LinkID', 'Longitude', 'Latitude', 'Speed (kilometer)', 'DD/MM/YY', 'HH:MM:SS', 'GX', 'GY', 'Streetname', 'Streetlength', 'Speedlimit (mph)', 'Speedlimit (km/h)', 'speeding']

def addinfo():
    inputFile = open(path + '/ ' + filename, 'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    root_path = '/Users/toshiyokoo/Desktop/11_result'
    outputFile = open(os.path.join(root_path, filename[:−4] + '.csv'), 'a')
    writer = csv.writer(outputFile, lineterminator='\n')
    print >> outputFile, ', '.join(title)
    for row in islice(reader, 1, None):
        list = []
        Filename = filename[−4]
        longitude = str(row[0])
        latitude = str(row[1])
        speed = str(row[2])
        date = str(row[3])
        hms = str(row[4])
        GX = str(row[5])
        GY = str(row[6])
        streetname = str(row[8])
        streetlength = str(row[9])
        speedlimit = str(row[11])
        linkID = str(row[12])
        drive_speed = float(row[2])
#convert the unit of speedlimit from mile to kilometer
mile = float(row[11])
conv_fac = 1.609344
kilometer = mile * conv_fac

#whether the speeding or not
if (drive_speed - kilometer) > 0:
speed2 = 1
else:
speed2 = 0

list.extend([personid, Filename, linkID, longitude, latitude, speed, date, hms, GX, GY, streetname, streetlength, speedlimit, kilometer, speed2])
writer.writerow(list)
inputFile.close()
print filename

path = "/Users/toshiyokoo/Desktop/aaaei"
for filename in os.listdir(path):
global filename
addinfo()

#integrate multiple GPS points of one link into one GPS point

import csv,os, numpy, math
from itertools import islice
import datetime
title = ['PersonID', 'Tripid', 'LinkID', 'streetname', 'hour', 'streetlength', 'Speedlimit', 'Sum_drivespeed(kilometer)', 'sum_speedlimit(kilometer)', 'sum_speed', 'sum_time', 'degree', 'percentage']

def calculate():
    link_new = 999999 #Errornumber
    count = 0
    all_count = 0
    time = 0
    drive = 0
    speedlimit = 0
    speeding = 0
    inputFile = open(path + '/' +filename,'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    for row in islice(reader, 1, None):
        all_count += 1
        reader = csv.reader(inputFile)
        inputFile.seek(0)
        for row in islice(reader, 1, None):
            root_path = '/Users/toshiyokoo/Desktop/12_result'
            outputFile = open(os.path.join(root_path,
                filename[:-4] + '.csv'), 'a')  
            writer = csv.writer(outputFile, lineterminator ='
')
            Tripname = filename[:-4]
            date = str(row[6])
            209
hms = str(row[7])
streetname = str(row[10])
streetlength = str(row[11])
linkID = str(row[2])
speedmph = str(row[12])
list = []

if link_new == 999999:
    print >> outputFile, ', '.join(title)
    time += 1
    drive += float(row[5])  # drive_speed
    speedlimit += float(row[13])  # speed_limit
    speeding += float(row[14])  # whether speeding or not
    link_new = linkID
    old_date = date
    old_hms = hms
    old_streetname = streetname
    old_streetlength = streetlength
    old_linkID = linkID
    old_speedlimit = speedmph
    count += 1
    continue

elif link_new == linkID:
    time += 1
    drive += float(row[5])
    speedlimit += float(row[13])
    speeding += float(row[14])
    # old_streetname = streetname
# old_streelength = streetlength
# old_linkID = linkID
count += 1
if count == all_count:
    sum_drive = drive
    sum_speedlimit = speedlimit
    sum_speed = speeding
    sum_time = time
    degree = sum_drive / sum_speedlimit
    percentage = sum_speed / sum_time
if old_date[-4:] == '2011':
    d = datetime.datetime.strptime(old_date, '%d/%m/%Y')
else:
    d = datetime.datetime.strptime(old_date, '%d/%m/%y')

# change the hour from UTC to MN
# (Time in GPS is Greenwich Mean Time (GMT))
f = datetime.datetime.strptime(old_hms, '%H:%M:%S')
hour = f.hour
date1 = datetime.datetime(2011, 3, 13)
date2 = datetime.datetime(2011, 11, 5)
# Daylight Saving Time
if date1 <= d <= date2:
    cal_hour = hour - 5
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour =
        cal_hour +
        24
#Standard Time
else:
    cal_hour = hour - 6
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour =
        cal_hour +
        24

list.extend([personid, Tripname,
             old_linkID, old_streetname,
             fix_hour, old_streetlength,
             old_speedlimit, sum_drive,
             sum_speedlimit, sum_speed,
             sum_time, degree, percentage])
writer.writerow(list)
b
continue

elif link_new != linkID:
    sum_drive = drive
    sum_speedlimit = speedlimit
    sum_speed = speeding
    sum_time = time

212
degree = sum_drive / sum_speedlimit
percentage = sum_speed / sum_time
if old_date[-4:] == '2011':
    d = datetime.datetime.strptime(
        old_date, '%d/%m/%Y')
else:
    d = datetime.datetime.strptime(
        old_date, '%d/%m/%Y')

#change the hour from UTC to MN (Time
in GPS is Greenwich Mean Time (GMT))
f = datetime.datetime.strptime(old_hms, '
%H:%M:%S'
) hour = f.hour
date1 = datetime.datetime(2011, 3, 13)
date2 = datetime.datetime(2011, 11, 5)
#Daylight Saving Time
if date1 <= d <= date2:
    cal_hour = hour - 5
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour = cal_hour + 24

#Standard Time
else:
    cal_hour = hour - 6
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour = cal_hour + 24

213
list.extend([personid, Tripname, 
    old_linkID, old_streetname, fix_hour, 
    old_streetlength, old_speedlimit, 
    sum_drive, sum_speedlimit, sum_speed, 
    sum_time, degree, percentage])

writer.writerow(list)
list = []
time = 1
drive = float(row[5])
speedlimit = float(row[13])
speeding = float(row[14])
link_new = linkID
count += 1
old_date = date
old_hms = hms
old_streetname = streetname
old_streetlength = streetlength
old_linkID = linkID
old_speedlimit = speedmph
if count == all_count:
    sum_drive = drive
    sum_speedlimit = speedlimit
    sum_speed = speeding
    sum_time = time
    degree = sum_drive / 
        sum_speedlimit
    percentage = sum_speed / 
        sum_time
if old_date[-4:] == '2011':
    d = datetime.datetime.strptime(old_date, 
        '%d/%m/%Y')
else:

d = datetime.datetime.strptime(old_date, '
%d/%m/%Y')

# change the hour from UTC to MN
# (Time in GPS is Greenwich Mean Time (GMT))
f = datetime.datetime.strptime(old_hms, '%H:%M:%S')
hour = f.hour
date1 = datetime.datetime(2011, 3, 13)
date2 = datetime.datetime(2011, 11, 5)

# Daylight Saving Time
if date1 <= d <= date2:
    cal_hour = hour - 5
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour =
        cal_hour +
        24

# Standard Time
else:
    cal_hour = hour - 6
    fix_hour = cal_hour
    if cal_hour < 0:
        fix_hour =
        cal_hour +
        24
list.extend([personid, Tripname, old_linkID, old_streetname, fix_hour, old_streetlength, old_speedlimit, sum_drive, sum_speedlimit, sum_speed, sum_time, degree, percentage])
writer.writerow(list)
break

print filename
inputFile.close()
outputFile.close()

path = "/Users/toshiyokoo/Desktop/aaafh"
for filename in os.listdir(path):
global filename
calculate()

import csv, os
from itertools import islice
title = ['PersonID', 'Tripid', 'LinkID', 'streetname', 'hour', 'streetlength', 'Speedlimit', 'Sum_drivespeed(kilometer)', 'sum_speedlimit(kilometer)', 'sum_speed', 'sum_time', 'degree', 'percentage']
def compile():
    input_file = open(path + '/' + filename, 'rU')
    reader = csv.reader(input_file)
    input_file.seek(0)
    for row in islice(reader, 1, None):
        print >> outputFile, ','.join(row)

216
inputFile.close()
print filename

path = "/Users/toshiyokoo/Desktop/aaafi"
root_path = "/Users/toshiyokoo/Desktop/13_result"
outputFile = open(os.path.join(root_path,"summary_13")+.csv', 'a')
writer = csv.writer(outputFile)
print >> outputFile, ',', ''.join(title)
for filename in os.listdir(path):
    global filename
    compile()
outputFile.close()

import csv, os, numpy, math
from itertools import islice
import datetime

title = ['PersonID', 'TripID', 'LinkID', 'Hour', 'streetlength', 'Speedlimit', 'degree', 'percentage', 'speed_25', 'speed_30', 'speed_35', 'speed_40', 'speed_45', 'speed_50', 'speed_55', 'speed_60', 'speed_65', 'morning', 'afternoon', 'night', 'o_gender', 'o_edu', 'o_age', 'gender', 'education_1', 'education_2', 'age_25_to_34', 'age_35_to_44', 'age_45_to_54', 'age_55_to_64', 'age_65_to_74', 'age_75_to_84', 'age_85_and_over']

def summarize():
    inputFile = open(path + '/'+filename,'rU')
    reader = csv.reader(inputFile)
    inputFile.seek(0)
    root_path = '/Users/toshiyokoo/Desktop/14_result'
outputFile = open(os.path.join(root_path, "summary_14")
+ '.csv', 'a')
writer = csv.writer(outputFile, lineterminator = '\n')
print >> outputFile, ',', .join(title)
for row in islice(reader, 1, None):
  list = []
  personid = str(row[0])
  tripid = str(row[1])
  linkid = str(row[2])
  hour = str(row[4])
  streetlength = str(row[5])
  speedlimit = str(row[6])
  degree = str(row[11])
  percentage = str(row[12])

  # Dummy variable (Speed limit)
  num_speedlimit = int(row[6])
  speed_25 = 0
  speed_30 = 0
  speed_35 = 0
  speed_40 = 0
  speed_45 = 0
  speed_50 = 0
  speed_55 = 0
  speed_60 = 0
  speed_65 = 0
  if num_speedlimit <= 25:
    speed_25 = 1
  elif num_speedlimit == 30:
    speed_30 = 1
  elif num_speedlimit == 35:
    speed_35 = 1
elif num_speedlimit == 40:
    speed_40 = 1
elif num_speedlimit == 45:
    speed_45 = 1
elif num_speedlimit == 50:
    speed_50 = 1
elif num_speedlimit == 55:
    speed_55 = 1
elif num_speedlimit == 60:
    speed_60 = 1
elif num_speedlimit == 65:
    speed_65 = 1
elif num_speedlimit == 70:
    dummy = 0
else:
    speed_25 = 99
    speed_30 = 99
    speed_35 = 99
    speed_40 = 99
    speed_45 = 99
    speed_50 = 99
    speed_55 = 99
    speed_60 = 99
    speed_65 = 99

# Dummy variable (morning, afternoon, night)
morning = 0
afternoon = 0
night = 0
dummy_hour = int(hour)
if 6 <= dummy_hour < 12:
    morning = 1
elif 12 <= dummy_hour < 18:
    afternoon = 1
elif 18 <= dummy_hour < 24:
    night = 1
elif 0 <= dummy_hour < 6:
    dummy = 0
else:
    morning = 99
    afternoon = 99
    night = 99

# Input Household survey data
# Match Tripid between GPS file and Household survey data
TBIFile = open("/Users/toshiyokoo/Desktop/5133/TBIdata.csv", 'rU')
idlist = csv.reader(TBIFile)
TBIFile.seek(0)
gender = 0
age_25_to_34 = 0
age_35_to_44 = 0
age_45_to_54 = 0
age_55_to_64 = 0
age_65_to_74 = 0
age_75_to_84 = 0
age_85_and_over = 0
education_1 = 0
education_2 = 0
for dem in islice(idlist, 1, None):
    if int(dem[3]) == int(personid):
        # Dummy variable (Gender)
        o_gender = dem[4]
if dem[4] == 'Male':
    gender = 1
elif dem[4] == 'Female':
    gender = 0
else:
    gender = 99

# Dummy variable (Education)
o_education = dem[5]
if dem[5] == 'Daycare / Pre-school':
    dummy = 0
elif dem[5] == 'Less than high school':
    dummy = 0
elif dem[5] == 'High school graduate':
    dummy = 0
elif dem[5] == 'Some college':
    education_1 = 1
elif dem[5] == 'Vocational/Technical training':
    dummy = 0
    education_1 = 1
elif dem[5] == 'Bachelors degree':
    education_2 = 1
elif dem[5] == 'Graduate/Post-graduate degree':
    education_2 = 1
else:

221
education_1 = 99
education_2 = 99

# Dummy variable (Age)
o_age = dem[6]
if int(dem[6]) == 6:
age_25_to_34 = 1
elif int(dem[6]) == 7:
age_35_to_44 = 1
elif int(dem[6]) == 8:
age_45_to_54 = 1
elif int(dem[6]) == 9:
age_55_to_64 = 1
elif int(dem[6]) == 10:
age_65_to_74 = 1
elif int(dem[6]) == 11:
age_75_to_84 = 1
elif int(dem[6]) == 12:
age_85_and_over = 1
elif int(dem[6]) == 5:
dummy = 0
else:
age_25_to_34 = 99
age_35_to_44 = 99
age_45_to_54 = 99
age_55_to_64 = 99
age_65_to_74 = 99
age_75_to_84 = 99
age_85_and_over = 99
list.extend([personid, tripid, linkid, hour, 
streetlength, speedlimit, degree, percentage 
speed_25, speed_30, speed_35, speed_40, 
speed_45, speed_50, speed_55, speed_60, 
speed_65, morning, afternoon, night, 
o_gender, o_education, o_age, gender, 
education_1, education_2, age_25_to_34, 
age_35_to_44, age_45_to_54, age_55_to_64, 
age_65_to_74, age_75_to_84, age_85_and_over 
])

writer.writerow(list)

inputFile.close()

print filename

path = "/Users/toshiyokoo/Desktop/aaafj"
for filename in os.listdir(path):
    global filename
    summarize()

import csv, os, numpy, math
from itertools import islice
import datetime
def summarize():
    input_file = open(path + '/filename', 'rU')
    reader = csv.reader(input_file)
    input_file.seek(0)
    root_path = '/Users/toshiyokoo/Desktop/15_result'
    output_file = open(os.path.join(root_path, "summary_15") + '.csv', 'a')
    writer = csv.writer(output_file, lineterminator='\n')
    print >> output_file, ', '.join(title)
    for row in islice(reader, 1, None):
        list = []
        personid = str(row[0])
        person = int(row[0])
        tripid = str(row[1])
        linkid = str(row[2])
        hour = str(row[3])
        streetlength = str(row[4])
        speedlimit = str(row[5])
        degree = str(row[6])
        percentage = str(row[7])
        speed_25 = str(row[8])
        speed_30 = str(row[9])
        speed_35 = str(row[10])
        speed_40 = str(row[11])
        speed_45 = str(row[12])
        speed_50 = str(row[13])
        speed_55 = str(row[14])
        speed_60 = str(row[15])
        speed_65 = str(row[16])
        morning = str(row[17])
        afternoon = str(row[18])
night = str(row[19])
gender = str(row[20])
education = str(row[21])
age = str(row[22])

#person
person1 = 0
person2 = 0
person3 = 0
person4 = 0
person5 = 0
person6 = 0
person7 = 0
person8 = 0
person9 = 0
person10 = 0
person11 = 0
person12 = 0
person13 = 0
person14 = 0
person15 = 0
person16 = 0
person17 = 0
person18 = 0
person19 = 0
person20 = 0
person21 = 0
person22 = 0
person23 = 0
person24 = 0
person25 = 0
person26 = 0
person27 = 0
person28 = 0
person29 = 0
person30 = 0
person31 = 0
person32 = 0
person33 = 0
person34 = 0
person35 = 0
person36 = 0
person37 = 0
person38 = 0
person39 = 0
person40 = 0
person41 = 0
person42 = 0
person43 = 0
person44 = 0
person45 = 0
person46 = 0
person47 = 0
person48 = 0
person49 = 0
person50 = 0
person51 = 0
person52 = 0
person53 = 0
person54 = 0
person55 = 0
person56 = 0
person57 = 0
person58 = 0
person59 = 0
person60 = 0
person61 = 0
person62 = 0
person63 = 0
person64 = 0
person65 = 0
person66 = 0
person67 = 0
person68 = 0
person69 = 0
person70 = 0
person71 = 0
person72 = 0
person73 = 0
person74 = 0
person75 = 0
person76 = 0
person77 = 0
person78 = 0
person79 = 0
person80 = 0
person113 = 0
person114 = 0
person115 = 0
person116 = 0
person117 = 0
person118 = 0
person119 = 0
person120 = 0
person121 = 0
person122 = 0
person123 = 0
person124 = 0
person125 = 0
person126 = 0
person127 = 0
person128 = 0
person129 = 0
person130 = 0
person131 = 0
person132 = 0
person133 = 0
person134 = 0
person135 = 0
person136 = 0
person137 = 0
person138 = 0
person139 = 0
person140 = 0
person141 = 0
person142 = 0
person143 = 0
person144 = 0


```python
person145 = 0
person146 = 0
person147 = 0
person148 = 0
person149 = 0
person150 = 0
person151 = 0
#person152 = 0

#if person == 1013360102:
    # person1 = 1
#elif person == 1022380102:
    # person2 = 1

#Input personID data
PersonFile = open("/Users/toshiyokoo/Desktop/5133/Persondata.csv",'rU')
idlist = csv.reader(PersonFile)
PersonFile.seek(0)
for dem in islice(idlist, 0, None):
    if person == int(dem[0]):
        person1 = 1
    elif person == int(dem[1]):
        person2 = 1
    elif person == int(dem[2]):
        person3 = 1
    elif person == int(dem[3]):
        person4 = 1
    elif person == int(dem[4]):
        person5 = 1
    elif person == int(dem[5]):
        person6 = 1
```
elif person == int(dem[6]):
    person7 = 1
elif person == int(dem[7]):
    person8 = 1
elif person == int(dem[8]):
    person9 = 1
elif person == int(dem[9]):
    person10 = 1
elif person == int(dem[10]):
    person11 = 1
elif person == int(dem[11]):
    person12 = 1
elif person == int(dem[12]):
    person13 = 1
elif person == int(dem[13]):
    person14 = 1
elif person == int(dem[14]):
    person15 = 1
elif person == int(dem[15]):
    person16 = 1
elif person == int(dem[16]):
    person17 = 1
elif person == int(dem[17]):
    person18 = 1
elif person == int(dem[18]):
    person19 = 1
elif person == int(dem[19]):
    person20 = 1
elif person == int(dem[20]):
    person21 = 1
elif person == int(dem[21]):
    person22 = 1
elif person == int(dem[22]):
    person23 = 1
elif person == int(dem[23]):
    person24 = 1
elif person == int(dem[24]):
    person25 = 1
elif person == int(dem[25]):
    person26 = 1
elif person == int(dem[26]):
    person27 = 1
elif person == int(dem[27]):
    person28 = 1
elif person == int(dem[28]):
    person29 = 1
elif person == int(dem[29]):
    person30 = 1
elif person == int(dem[30]):
    person31 = 1
elif person == int(dem[31]):
    person32 = 1
elif person == int(dem[32]):
    person33 = 1
elif person == int(dem[33]):
    person34 = 1
elif person == int(dem[34]):
    person35 = 1
elif person == int(dem[35]):
    person36 = 1
elif person == int(dem[36]):
    person37 = 1
elif person == int(dem[37]):
    person38 = 1
elif person == int(dem[38]):
    person39 = 1
elif person == int(dem[39]):
    person40 = 1
elif person == int(dem[40]):
    person41 = 1
elif person == int(dem[41]):
    person42 = 1
elif person == int(dem[42]):
    person43 = 1
elif person == int(dem[43]):
    person44 = 1
elif person == int(dem[44]):
    person45 = 1
elif person == int(dem[45]):
    person46 = 1
elif person == int(dem[46]):
    person47 = 1
elif person == int(dem[47]):
    person48 = 1
elif person == int(dem[48]):
    person49 = 1
elif person == int(dem[49]):
    person50 = 1
elif person == int(dem[50]):
    person51 = 1
elif person == int(dem[51]):
    person52 = 1
elif person == int(dem[52]):
    person53 = 1
elif person == int(dem[53]):
    person54 = 1
elif person == int(dem[54]):
    person55 = 1
elif person == int(dem[55]):
    person56 = 1
elif person == int(dem[56]):
    person57 = 1
elif person == int(dem[57]):
    person58 = 1
elif person == int(dem[58]):
    person59 = 1
elif person == int(dem[59]):
    person60 = 1
elif person == int(dem[60]):
    person61 = 1
elif person == int(dem[61]):
    person62 = 1
elif person == int(dem[62]):
    person63 = 1
elif person == int(dem[63]):
    person64 = 1
elif person == int(dem[64]):
    person65 = 1
elif person == int(dem[65]):
    person66 = 1
elif person == int(dem[66]):
    person67 = 1
elif person == int(dem[67]):
    person68 = 1
elif person == int(dem[68]):
    person69 = 1
elif person == int(dem[69]):
    person70 = 1
elif person == int(dem[70]):
    person71 = 1
elif person == int(dem[71]):
    person72 = 1
elif person == int(dem[72]):
    person73 = 1
elif person == int(dem[73]):
    person74 = 1
elif person == int(dem[74]):
    person75 = 1
elif person == int(dem[75]):
    person76 = 1
elif person == int(dem[76]):
    person77 = 1
elif person == int(dem[77]):
    person78 = 1
elif person == int(dem[78]):
    person79 = 1
elif person == int(dem[79]):
    person80 = 1
elif person == int(dem[80]):
    person81 = 1
elif person == int(dem[81]):
    person82 = 1
elif person == int(dem[82]):
    person83 = 1
elif person == int(dem[83]):
    person84 = 1
elif person == int(dem[84]):
    person85 = 1
elif person == int(dem[85]):
    person86 = 1
elif person == int(dem[86]):
    person87 = 1
elif person == int(dem[87]):
    person88 = 1
elif person == int(dem[88]):
    person89 = 1
elif person == int(dem[89]):
    person90 = 1
elif person == int(dem[90]):
    person91 = 1
elif person == int(dem[91]):
    person92 = 1
elif person == int(dem[92]):
    person93 = 1
elif person == int(dem[93]):
    person94 = 1
elif person == int(dem[94]):
    person95 = 1
elif person == int(dem[95]):
    person96 = 1
elif person == int(dem[96]):
    person97 = 1
elif person == int(dem[97]):
    person98 = 1
elif person == int(dem[98]):
    person99 = 1
elif person == int(dem[99]):
    person100 = 1
elif person == int(dem[100]):
    person101 = 1
elif person == int(dem[101]):
    person102 = 1
elif person == int(dem[102]):
    person103 = 1
elif person == int(dem[103]):
    person104 = 1
elif person == int(dem[104]):
    person105 = 1
elif person == int(dem[105]):
    person106 = 1
elif person == int(dem[106]):
    person107 = 1
elif person == int(dem[107]):
    person108 = 1
elif person == int(dem[108]):
    person109 = 1
elif person == int(dem[109]):
    person110 = 1
elif person == int(dem[110]):
    person111 = 1
elif person == int(dem[111]):
    person112 = 1
elif person == int(dem[112]):
    person113 = 1
elif person == int(dem[113]):
    person114 = 1
elif person == int(dem[114]):
    person115 = 1
elif person == int(dem[115]):
    person116 = 1
elif person == int(dem[116]):
    person117 = 1
elif person == int(dem[117]):
    person118 = 1
elif person == int(dem[118]):
    person119 = 1
elif person == int(dem[119]):
    person120 = 1
elif person == int(dem[120]):
    person121 = 1
elif person == int(dem[121]):
    person122 = 1
elif person == int(dem[122]):
    person123 = 1
elif person == int(dem[123]):
    person124 = 1
elif person == int(dem[124]):
    person125 = 1
elif person == int(dem[125]):
    person126 = 1
elif person == int(dem[126]):
    person127 = 1
elif person == int(dem[127]):
    person128 = 1
elif person == int(dem[128]):
    person129 = 1
elif person == int(dem[129]):
    person130 = 1
elif person == int(dem[130]):
    person131 = 1
elif person == int(dem[131]):
    person132 = 1
elif person == int(dem[132]):
    person133 = 1
elif person == int(dem[133]):
    person134 = 1
elif person == int(dem[134]):
    person135 = 1
elif person == int(dem[135]):
    person136 = 1
elif person == int(dem[136]):
    person137 = 1
elif person == int(dem[137]):
    person138 = 1
elif person == int(dem[138]):
    person139 = 1
elif person == int(dem[139]):
    person140 = 1
elif person == int(dem[140]):
    person141 = 1
elif person == int(dem[141]):
    person142 = 1
elif person == int(dem[142]):
    person143 = 1
elif person == int(dem[143]):
    person144 = 1
elif person == int(dem[144]):
    person145 = 1
elif person == int(dem[145]):
    person146 = 1
elif person == int(dem[146]):
    person147 = 1
elif person == int(dem[147]):
    person148 = 1
elif person == int(dem[148]):
    person149 = 1
elif person == int(dem[149]):
    person150 = 1
elif person == int(dem[150]):
    person151 = 1

# elif person == int(dem[151]):
#     person152 = 1
list.extend([personid, tripid, linkid, hour, streetlength, speedlimit, degree, percentage, speed_25, speed_30, speed_35, speed_40, speed_45, speed_50, speed_55, speed_60, speed_65, morning, afternoon, night, o_gender, o_education, o_age, gender, education_1, education_2, age_25_to_34, age_35_to_44, age_45_to_54, age_55_to_64, age_65_to_74, age_75_to_84, age_85_and_over, person1, person2, person3, person4, person5, person6, person7, person8, person9, person10, person11, person12, person13, person14, person15, person16, person17, person18, person19, person20, person21, person22, person23, person24, person25, person26, person27, person28, person29, person30, person31, person32, person33, person34, person35, person36, person37, person38, person39, person40, person41, person42, person43, person44, person45, person46, person47, person48, person49, person50, person51, person52, person53, person54, person55, person56, person57, person58, person59, person60, person61, person62, person63, person64, person65, person66, person67, person68, person69, person70, person71, person72, person73, person74, person75, person76, person77, person78, person79, person80, person81, person82, person83, person84, person85, person86, person87, person88, person89, person90, person91, person92, person93, person94, person95, person96, person97, person98, person99, person100, person101, person102, person103, person104, person105, person106, person107, person108, person109, person110, person111, person112, person113, person114, person115, person116, person117,
writer.writerow(list)
inputFile.close()
print filename

path = "/Users/toshiyokoo/Desktop/aaafm"
for filename in os.listdir(path):
    global filename
    summarize()

A.4.19 Computing and counting number in R

R version 3.1.2 (2014-10-31) — "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.65 (6833) x86_64-apple-darwin10.8.0]
[Workspace restored from /Users/toshiyokoo/.RData]
> fnames <- dir(pattern="*.csv")
> csvlist <- lapply(fnames, read.csv)
> names(csvlist) <- fnames
> n <- length(csvlist)
> n
[1] 11
> temp <- csvlist[[1]]
> for (i in 2:n) {temp <- merge(temp, csvlist[[i]], all=T)}
> res <- temp
> summary(res)

   filename        tripid      date         time     speed kilometer.
187010_02_02.1  7283 Min. :1.013e+09 11/10/2011:
   27531 21:42:58:  54 Min. : 6.00
183019_02_02.21 3544 1st Qu.:1.229e+09 10/10/2011:
   26916 21:43:22:  53 1st Qu.: 31.00
186319_02_02.44 3063 Median :1.505e+09 28/9/2011:
   23472 21:43:21:  52 Median : 48.00
184208_02_02.44 3048 Mean :1.489e+09 22/7/2011:
121926_02_02.25 2536 3rd Qu.:1.766e+09 12/10/2011:
   19845 21:44:20:  51 3rd Qu.: 70.00
152077_01_02.112 2480 Max. :2.091e+09 29/9/2011:
   18262 21:43:0 :  50 Max. : 391.00
(Other) :1150149 (Other) :1171792
streetlength speedlimit.mph. speedlimit.km.
speed_limit_compliance speed2 hour
   Min. : 16.74 Min. : 5.00 Min. : 8.047 Min.
   :0.05736 Min. :0.0000 Min. : 0.00
<table>
<thead>
<tr>
<th></th>
<th>speed_25</th>
<th>speed_30</th>
<th>speed_35</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Me.</td>
<td>0.02054</td>
<td>0.1138</td>
<td>0.3225</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Median</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>speed_40</th>
<th>speed_45</th>
<th>speed_50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Me.</td>
<td>0.0193</td>
<td>0.01706</td>
<td>0.0193</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Median</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>1.0000</td>
<td>3.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>speed_55</th>
<th>speed_60</th>
<th>speed_65</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Me.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Median</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Max.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Median : 0.0000  Median : 0.0000  Median : 0.0000  Median : 0.0000
Mean : 0.1008  Mean : 0.0777  Mean : 0.0242  Mean : 0.3494
3rd Qu. : 0.0000  3rd Qu. : 0.0000  3rd Qu. : 0.0000  3rd Qu. : 1.0000
Max. : 1.0000  Max. : 1.0000  Max. : 1.0000  Max. : 1.0000

o_gender
goed
age
gender
education_1
Female : 666621  Associates degree : 48153  Min. : 5.000  Min. : 0.0000  Min. : 0.0000
Male : 505482  Bachelors degree : 428046  1st Qu. : 7.000  1st Qu. : 0.0000  1st Qu. : 0.0000
Graduate/Post-graduate degree : 343870  Median : 9.000  Median : 0.0000  Median : 0.0000
High school graduate : 140378  Mean : 8.576  Mean : 0.4313  Mean : 0.1845
Some college : 168149  3rd Qu. : 10.000  3rd Qu. : 1.0000  3rd Qu. : 0.0000
Vocational/Technical training : 43507  Max. : 12.000  Max. : 1.0000  Max. : 1.0000

education_2
age_25_to_34
age_35_to_44
age_45_to_54
age_55_to_64
age_65_to_74
Min. : 0.0000  Min. : 0.0000  Min. : 0.0000  Min. : 0.0000  Min. : 0.0000
1st Qu. : 0.0000  1st Qu. : 0.0000  1st Qu. : 0.0000  1st Qu. : 0.0000
Median : 1.0000  Median : 0.0000  Median : 0.0000  Median : 0.0000

246
Mean : 0.6586  Mean : 0.06539  Mean : 0.1697  Mean : 0.1787  Mean : 0.3082  Mean : 0.1808
3rd Qu.: 1.0000  3rd Qu.: 0.00000  3rd Qu.: 0.0000  3rd Qu.: 0.0000  3rd Qu.: 1.0000  3rd Qu.: 0.0000
Max. : 1.0000  Max. : 1.00000  Max. : 1.0000  Max. : 1.0000  Max. : 1.0000  Max. : 1.0000

age_75_to_84   age_85_and_over
Min. : 0.00000  Min. : 0.0000
1st Qu.: 0.00000  1st Qu.: 0.0000
Median : 0.00000  Median : 0.0000
Mean : 0.05869  Mean : 0.0209
3rd Qu.: 0.00000  3rd Qu.: 0.0000
Max. : 1.00000  Max. : 1.0000

> #Table
> #Percentage of speeding across speed limit zone
> table(res$speedlimit.mph., res$speed2)

                  0    1
5   145 614
10 1468 896
20   9 71
25  5581 15286
30 119937 13402
35  339354 38606
40  272477 74399
45  15470 7154
50  15303 4698
55  60023 58137
60  44765 46399
65  17589 10827
> x <- table(res$speedlimit.mph, res$speed2)
> write.csv(x, "x.csv")
>
> #count the number of individuals who used specific speed limit
> r <- table(res$tripid, res$speedlimit.mph)
> write.csv(r, "r.csv")
>
> #Percentage of speeding by time of day
> y <- table(res$hour, res$speed2)
> write.csv(y, "y.csv")
>
> #count the number of individuals who drove specific hour
> s <- table(res$tripid, res$hour)
> write.csv(s, "s.csv")
>
> #Speeding by gender
> z <- table(res$gender, res$speed2)
> write.csv(z, "z.csv")
>
> #count the number of driver depends on gender
> t <- table(res$tripid, res$gender)
> write.csv(t, "t.csv")
>
> #Speeding by age
> e <- table(res$o_age, res$speed2)
> write.csv(e, "e.csv")
>
> #count the number of driver depends on age
> u <- table(res$tripid, res$o_age)
> write.csv(u, "u.csv")
> # Speeding by education level
> f <- table(res$o$edu, res$s$speed2)
> write.csv(f, "f.csv.csv")
>
> # Count the number of driver depends on education level
> v <- table(res$t$tripid, res$s$education_1)
> write.csv(v, "v.csv.csv")
> w <- table(res$t$tripid, res$s$education_2)
> write.csv(w, "w.csv.csv")
>
**A.4.20 Box plot in R**

> fnames <- dir(pattern=".csv")
> csvlist <- lapply(fnames, read.csv)
> names(csvlist) <- fnames
> n <- length(csvlist)
> n
[1] 11
> temp <- csvlist[[1]]
> for (i in 2:n) {temp <- merge(temp, csvlist[[i]], all=T)}
> res <- temp
>
> # Speed limit
> boxplot(speed_limit_compliance ~ speedlimit.mph., data=res,
> ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Speed limit (mph)", ylab="driving speed/speed limit")
>
> # Time
> boxplot(speed_limit_compliance ~ hour, data=res, ylim=c(0, 8)
> , yaxp=c(0, 8, 8), xlab="Time", ylab="driving speed/speed limit")
> #Education
> boxplot(speed_limit_compliance ~ o_edu, data=res, ylim=c(0, 8), yaxp=c(0, 8), xlab="Education", ylab="driving speed/speed limit")
>
> #Age
> boxplot(speed_limit_compliance ~ o_age, data=res, ylim=c(0, 8), yaxp=c(0, 8), xlab="Age", ylab="driving speed/speed limit")
>
> #Gender
> boxplot(speed_limit_compliance ~ o_gender, data=res, ylim=c(0, 8), yaxp=c(0, 8), xlab="Gender", ylab="driving speed/speed limit")
>
A.4.21 Graph in R

> fnames <- dir(pattern="*.csv")
> csvlist <- lapply(fnames, read.csv)
> names(csvlist) <- fnames
> n <- length(csvlist)
> n
> [1] 11
> temp <- csvlist[[1]]
> for (i in 2:n) {temp <- merge(temp, csvlist[[i]], all=T)}
> res <- temp
> #Plot between street length and speeding
> street_length <- res$streetlength
> speeding <- res$speed_limit_compliance
> plot(street_length, speeding, pch=".", ylim=c(0, 8), xlab="Link length", ylab="driving speed/speed limit")
> lm.obj< lm(speeding ~ street_length)
> abline(lm.obj, col=2)
> summary(lm.obj)

Call:
  lm(formula = speeding ~ street_length)

Residuals:
   Min     1Q Median     3Q    Max
-1.4096 -0.2039  0.0400  0.1957  6.8803

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.679e-01  4.162e-04 1604.8  <2e-16 ***
street_length 2.716e-04  8.212e-07  330.7  <2e-16 ***

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3152 on 1172101 degrees of freedom
Multiple R-squared:  0.08536,  Adjusted R-squared:  0.08536
F-statistic: 1.094e+05 on 1 and 1172101 DF,  p-value: < 2.2e-16

> mtext("N=1,172,103, R^2=0.0854, y=0.6679+0.0003x", side=3)
> 
> #Link length is less than 1,000m
> length1000 <- subset(res, res$streetlength <= 1000)
> street_length_1 <- length1000$streetlength
> speeding_1 <- length1000$speed_limit_compliance
```r
> lm1000.obj <- lm(speeding_1 ~ street_length_1)
> summary(lm1000.obj)

Call:
  lm(formula = speeding_1 ~ street_length_1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.02700 -0.19760  0.03170  0.19140  6.91440

Coefficients:         Estimate Std. Error t value Pr(>|t|)
(Intercept)           6.078e-01  5.075e-04   1197.5 <2e-16 ***
street_length_1      4.881e-04  1.363e-06    358.2 <2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.3124 on 1112317 degrees of freedom
Multiple R-squared:  0.1034 , Adjusted R-squared:  0.1034
F-statistic: 1.283e+05 on 1 and 1112317 DF, p-value: < 2.2e-16

> plot(street_length_1, speeding_1, pch=".", ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Link length", ylab="driving speed/speed limit")
> abline(lm1000.obj, col=2)
> mtext("N=1,112,319, R^2=0.103, y=0.6078+0.0005x", side=3)
> # polynomial model
> m2 <- lm(speeding ~ street_length + I(street_length^2))
> summary(m2)
```

252
Call:
\( \text{lm(formula = speeding} \sim \text{street.length + I(street.length}^2) \)\)

Residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1.0322</td>
<td>-0.1914</td>
<td>0.0327</td>
<td>0.1890</td>
<td>6.9124</td>
</tr>
</tbody>
</table>

Coefficients:

|                     | Estimate | Std. Error | t value | Pr(>|t|) |
|---------------------|----------|------------|---------|----------|
| (Intercept)         | 6.025e-01| 5.226e-04  | 1152.8  | <2e-16   *** |
| street.length       | 5.679e-04| 1.680e-06  | 338.1   | <2e-16   *** |
| I(street.length^2)  | -1.632e-07| 8.113e-10  | -201.2  | <2e-16   *** |

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.3099 on 1172100 degrees of freedom
Multiple R-squared: 0.1159, Adjusted R-squared: 0.1159
F-statistic: 7.681e+04 on 2 and 1172100 DF, p-value: < 2.2e-16

> # polynomial model (Link length is less than 1,000m)
> m4 <- lm(speeding_1 ~ street.length_1 + I(street.length_1^2))
> summary(m4)

Call:
\( \text{lm(formula = speeding_1} \sim \text{street.length_1 + I(street.length_1}^2) \)\)

Residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.8900</td>
<td>-0.1926</td>
<td>0.0351</td>
<td>0.1863</td>
<td>6.9403</td>
</tr>
</tbody>
</table>
Coefficients:

|                     | Estimate | Std. Error | t value | Pr(>|t|) |
|---------------------|----------|------------|---------|----------|
| (Intercept)         | 5.142e-01| 8.627e-04  | 596.0   | <2e-16 ***|
| street_length_1     | 1.142e-03| 5.075e-06  | 225.0   | <2e-16 ***|
| I(street_length_1^2)| -7.501e-07| 5.614e-09  | -133.6  | <2e-16 ***|

---

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 0.1 1

Residual standard error: 0.31 on 1112316 degrees of freedom
Multiple R-squared: 0.1176, Adjusted R-squared: 0.1176
F-statistic: 7.41e+04 on 2 and 1112316 DF, p-value: < 2.2e-16

> #plot (speed limit: 25 mph)
> speed25 <- subset(res, res$speedlimit.mph=="25")
> summary(speed25)

<table>
<thead>
<tr>
<th>filename</th>
<th>tripid</th>
<th>date</th>
<th>time</th>
<th>speed.kilometer</th>
</tr>
</thead>
<tbody>
<tr>
<td>126871_01_02_57:</td>
<td>253</td>
<td>11/10/2011:</td>
<td>597</td>
<td>1.013e+09</td>
</tr>
<tr>
<td>115164_01_02_3:</td>
<td>178</td>
<td>10/8/2011 :</td>
<td>481</td>
<td>1.219e+09</td>
</tr>
<tr>
<td>125425_01_02_65:</td>
<td>126</td>
<td>27/9/2011:</td>
<td>467</td>
<td>1.373e+09</td>
</tr>
<tr>
<td>121926_01_02_21:</td>
<td>114</td>
<td>11/8/2011:</td>
<td>448</td>
<td>1.428e+09</td>
</tr>
<tr>
<td>121926_01_02_18:</td>
<td>112</td>
<td>9/8/2011 :</td>
<td>401</td>
<td>1.576e+09</td>
</tr>
<tr>
<td>176075_02_02_2:</td>
<td>111</td>
<td>24/8/2011:</td>
<td>393</td>
<td>2.091e+09</td>
</tr>
</tbody>
</table>

254
<table>
<thead>
<tr>
<th></th>
<th>speed2</th>
<th>hour</th>
<th>speed 25</th>
<th>speed 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>25</td>
<td>Min.</td>
<td>40.23</td>
<td>Min.</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>Min.</td>
<td>0.00</td>
<td>Min.</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>25</td>
<td>1st Qu.</td>
<td>40.23</td>
<td>1st Qu.</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>1st Qu.</td>
<td>9.00</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>Median</td>
<td>25</td>
<td>Median</td>
<td>40.23</td>
<td>Median</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>Median</td>
<td>13.00</td>
<td>Median</td>
</tr>
<tr>
<td>Mean</td>
<td>25</td>
<td>Mean</td>
<td>40.23</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>0.7325</td>
<td>Mean</td>
<td>12.98</td>
<td>Mean</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>25</td>
<td>3rd Qu.</td>
<td>40.23</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>3rd Qu.</td>
<td>17.00</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td>Max.</td>
<td>25</td>
<td>Max.</td>
<td>40.23</td>
<td>Max.</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>Max.</td>
<td>23.00</td>
<td>Max.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>speed 35</th>
<th>speed 40</th>
<th>speed 45</th>
<th>speed 50</th>
<th>speed 55</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0</td>
<td>Min.</td>
<td>0</td>
<td>Min.</td>
<td>0</td>
</tr>
<tr>
<td>Min.</td>
<td>0</td>
<td>Min.</td>
<td>0</td>
<td>Min.</td>
<td>0</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0</td>
<td>1st Qu.</td>
<td>0</td>
<td>1st Qu.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1st Qu.</td>
<td>0</td>
<td>1st Qu.</td>
<td>0</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>Median</td>
<td>0</td>
<td>Median</td>
<td>0</td>
<td>Median</td>
<td>0</td>
</tr>
<tr>
<td>Median</td>
<td>0</td>
<td>Median</td>
<td>0</td>
<td>Median</td>
<td>0.0000</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
<td>Mean</td>
<td>0</td>
<td>Mean</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
<td>Mean</td>
<td>0</td>
<td>Mean</td>
<td>0.3736</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0</td>
<td>3rd Qu.</td>
<td>0</td>
<td>3rd Qu.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3rd Qu.</td>
<td>0</td>
<td>3rd Qu.</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Max.</td>
<td>0</td>
<td>Max.</td>
<td>0</td>
<td>Max.</td>
<td>0</td>
</tr>
<tr>
<td>Max.</td>
<td>0</td>
<td>Max.</td>
<td>0</td>
<td>Max.</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

255
<table>
<thead>
<tr>
<th></th>
<th>afternoon</th>
<th>night</th>
<th>o_gender</th>
<th>o_edu</th>
<th>o_age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>Female</td>
<td>11322</td>
<td>Associates</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Male</td>
<td>9545</td>
<td>Bachelors</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>575</td>
<td>8330</td>
<td>7.000</td>
</tr>
<tr>
<td>Median</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>8.000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.4195</td>
<td>0.1756</td>
<td>2297</td>
<td>8,342</td>
<td>8.342</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>1.0000</td>
<td>0.0000</td>
<td>2138</td>
<td>9.000</td>
<td></td>
</tr>
<tr>
<td>Max.</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1202</td>
<td>12.000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>gender</th>
<th>education_1</th>
<th>education_2</th>
<th>age_25_to_34</th>
<th>age_35_to_44</th>
<th>age_45_to_54</th>
<th>age_55_to_64</th>
<th>age_65_to_74</th>
<th>age_75_to_84</th>
<th>age_85_and_over</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Median</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.4574</td>
<td>0.1300</td>
<td>0.7023</td>
<td>0.2100</td>
<td>0.1800</td>
<td>0.1760</td>
<td>0.1800</td>
<td>0.1800</td>
<td>0.1800</td>
<td>0.1800</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

256
1st Qu.: 0.0000 1st Qu.: 0.00000 1st Qu.: 0.00000
Median : 0.0000  Median : 0.00000  Median : 0.00000
Mean : 0.1555   Mean : 0.04179  Mean : 0.02037
3rd Qu.: 0.0000 3rd Qu.: 0.00000 3rd Qu.: 0.00000
Max. : 1.0000  Max. : 1.00000  Max. : 1.00000

> plot(speed25$streetlength, speed25$speed_limit_compliance, pch = ".", ylim=c(0, 3), yaxp=c(0, 3, 6), xlab="Link length (Speed limit: 25mph)", ylab="driving speed/speed limit")
> lm_25.obj<-lm(speed25$speed_limit_compliance ~ speed25$streetlength)
> summary(lm_25.obj)

Call:
  lm(formula = speed25$speed_limit_compliance ~ speed25$streetlength)

Residuals:
     Min      1Q  Median     3Q   Max
-1.89516 -0.44544  0.03034  0.45883  1.56718

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.074e+00  1.186e-02  90.52  <2e-16 ***
speed25$streetlength  9.142e-04  2.943e-05  31.06  <2e-16 ***

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6208 on 20865 degrees of freedom
Multiple R-squared:  0.04419,   Adjusted R-squared:  0.04415
F-statistic: 964.8 on 1 and 20865 DF,  p-value: < 2.2e-16
```r
> abline(lm.25.obj, col=2)
> mtext("N=20,867, R^2=0.0442, y=1.0740+0.0009x", side=3)
>
> #plot (speed limit: 30 mph)
> speed30 <- subset(res, res$speedlimit.mph."=="30")
> summary(speed30)

<table>
<thead>
<tr>
<th>filename</th>
<th>tripid</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>152077_01_02_112:</td>
<td>1914</td>
<td>10/7/2011 :</td>
</tr>
<tr>
<td>4995</td>
<td>12</td>
<td>Min. : 6.00</td>
</tr>
<tr>
<td>125650_01_02_47:</td>
<td>1520</td>
<td>24/10/2011:</td>
</tr>
<tr>
<td>4123</td>
<td>12</td>
<td>1st Qu. : 19.00</td>
</tr>
<tr>
<td>125650_02_02_63:</td>
<td>1419</td>
<td>6/9/2011 :</td>
</tr>
<tr>
<td>3458</td>
<td>11</td>
<td>Median : 30.00</td>
</tr>
<tr>
<td>184208_02_02_36:</td>
<td>1287</td>
<td>23/7/2011 :</td>
</tr>
<tr>
<td>3406</td>
<td>11</td>
<td>Mean : 30.54</td>
</tr>
<tr>
<td>184208_02_02_26:</td>
<td>1192</td>
<td>21/7/2011 :</td>
</tr>
<tr>
<td>3338</td>
<td>11</td>
<td>3rd Qu. : 43.00</td>
</tr>
<tr>
<td>184208_02_02_44:</td>
<td>1190</td>
<td>22/7/2011 :</td>
</tr>
<tr>
<td>3190</td>
<td>11</td>
<td>Max. : 113.00</td>
</tr>
</tbody>
</table>

(Other) :124817
(Other) :110829
(Other) :133271

streetlength speedlimit.mph. speedlimit.km. speed_limit_compliance speed2 hour
Min. : 21.4 Min. :30 Min. :48.28 Min. :0.1243 Min. :0.0000 Min. : 0.00
1st Qu.: 102.1 1st Qu.:30 1st Qu.:48.28 1st Qu.:0.3935 1st Qu.:0.0000 1st Qu.:10.00
Median : 135.5 Median :30 Median :48.28 Median :0.6214 Median :0.0000 Median :14.00

258
<table>
<thead>
<tr>
<th>speed_25</th>
<th>speed_30</th>
<th>speed_35</th>
<th>speed_40</th>
<th>speed_45</th>
<th>speed_50</th>
<th>speed_55</th>
<th>speed_60</th>
<th>speed_65</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. : 0</td>
<td>Min. : 1</td>
<td>Min. : 0</td>
<td>Min. : 0</td>
<td>Min. : 0</td>
<td>Min. : 0</td>
<td>Min. : 0</td>
<td>Min. : 0</td>
<td>Min. : 0</td>
</tr>
<tr>
<td>1st Qu. : 0</td>
<td>1st Qu. : 1</td>
<td>1st Qu. : 0</td>
<td>1st Qu. : 0</td>
<td>1st Qu. : 0</td>
<td>1st Qu. : 0</td>
<td>1st Qu. : 0</td>
<td>1st Qu. : 0</td>
<td>1st Qu. : 0</td>
</tr>
<tr>
<td>Median : 0</td>
<td>Median : 1</td>
<td>Median : 0</td>
<td>Median : 0</td>
<td>Median : 0</td>
<td>Median : 0</td>
<td>Median : 0</td>
<td>Median : 0</td>
<td>Median : 0</td>
</tr>
<tr>
<td>Mean : 0</td>
<td>Mean : 1</td>
<td>Mean : 0</td>
<td>Mean : 0</td>
<td>Mean : 0</td>
<td>Mean : 0</td>
<td>Mean : 0</td>
<td>Mean : 0</td>
<td>Mean : 0</td>
</tr>
<tr>
<td>3rd Qu. : 0</td>
<td>3rd Qu. : 1</td>
<td>3rd Qu. : 0</td>
<td>3rd Qu. : 0</td>
<td>3rd Qu. : 0</td>
<td>3rd Qu. : 0</td>
<td>3rd Qu. : 0</td>
<td>3rd Qu. : 0</td>
<td>3rd Qu. : 0</td>
</tr>
<tr>
<td>Max. : 0</td>
<td>Max. : 1</td>
<td>Max. : 0</td>
<td>Max. : 0</td>
<td>Max. : 0</td>
<td>Max. : 0</td>
<td>Max. : 0</td>
<td>Max. : 0</td>
<td>Max. : 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>morning</th>
<th>afternoon</th>
<th>night</th>
<th>o_gender</th>
<th>o_edu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. : 0.0000</td>
<td>Min. : 0.0000</td>
<td>Min. : 0.0000</td>
<td>Female</td>
<td>72062</td>
</tr>
<tr>
<td>1st Qu. : 0.0000</td>
<td>1st Qu. : 0.0000</td>
<td>1st Qu. : 0.0000</td>
<td>Male</td>
<td>61277</td>
</tr>
<tr>
<td>Median : 0.0000</td>
<td>Median : 0.0000</td>
<td>Median : 0.0000</td>
<td>Graduate/Post-graduate degree : 42775</td>
<td></td>
</tr>
<tr>
<td>Mean : 0.3407</td>
<td>Mean : 0.4798</td>
<td>Mean : 0.1664</td>
<td>High school graduate : 8674</td>
<td></td>
</tr>
<tr>
<td>Age Group</td>
<td>Gender</td>
<td>Education 1</td>
<td>Education 2</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>25 to 34</td>
<td>Min.</td>
<td>:0.0000</td>
<td>Min. :0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max.</td>
<td>:1.0000</td>
<td>Max. :1.0000</td>
<td></td>
</tr>
<tr>
<td>35 to 44</td>
<td>Median</td>
<td>:0.0000</td>
<td>Median :0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>:8.46</td>
<td>Mean :0.4596</td>
<td></td>
</tr>
<tr>
<td>45 to 54</td>
<td>Min.</td>
<td>:0.0000</td>
<td>Min. :0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max.</td>
<td>:1.0000</td>
<td>Max. :1.0000</td>
<td></td>
</tr>
<tr>
<td>55 to 64</td>
<td>Median</td>
<td>:0.0000</td>
<td>Median :0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>:0.1686</td>
<td>Mean :0.3166</td>
<td></td>
</tr>
<tr>
<td>65 to 74</td>
<td>Min.</td>
<td>:0.0000</td>
<td>Min. :0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max.</td>
<td>:1.0000</td>
<td>Max. :1.0000</td>
<td></td>
</tr>
<tr>
<td>75 to 84</td>
<td>Median</td>
<td>:0.0000</td>
<td>Median :0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>:0.08826</td>
<td>Mean :0.01127</td>
<td></td>
</tr>
<tr>
<td>85 and over</td>
<td>Min.</td>
<td>:0.0000</td>
<td>Min. :0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max.</td>
<td>:1.0000</td>
<td>Max. :1.0000</td>
<td></td>
</tr>
</tbody>
</table>

Some college: 30281
Vocational/Technical training: 3003
> plot(speed30$streetlength, speed30$speed_limit_compliance, pch = ".", ylim=c(0, 3), yaxp=c(0, 3, 6), xlab="Link length (Speed limit: 30mph)", ylab="driving speed/speed limit")
> lm_30.obj<-
> summary(lm_30.obj)

Call:
lm(formula = speed30$speed_limit_compliance ~ speed30$streetlength)

Residuals:

Min 1Q Median 3Q Max
-0.50857 -0.23909 -0.01125 0.25799 1.70788

Coefficients:

  Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.326e-01 1.430e-03 442.303 <2e-16 ***
speed30$streetlength 1.965e-07 6.224e-06  0.032   0.975

Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.3025 on 133337 degrees of freedom
Multiple R-squared:  7.476e-09,   Adjusted R-squared:  -7.492e-06
F-statistic: 0.0009968 on 1 and 133337 DF,  p-value: 0.9748

> abline(lm_30.obj, col=2)
> mtext("N=133,339, R^2=7.5e-09, y=0.6326+1.97e-07x", side=3)
```r
> #plot (speed limit: 35 mph)
> speed35 <- subset(res, res$speedlimit.mph.=="35")
> summary(speed35)

    filename tripid date            time speed.kilometer.
187010_02_02_1 : 2364 Min. :1.013e+09 11/10/2011: 12223 20:38:12: 27 Min. : 6.00
127709_02_02_48: 1690 1st Qu.:1.219e+09 10/10/2011: 9588 20:38:13: 26 1st Qu.: 22.00
137171_01_02_8 : 1590 Mean :1.457e+09 28/9/2011 : 7894 20:38:14: 23 Mean : 35.89
193857_01_02_69: 1510 3rd Qu.:1.721e+09 8/10/2011 : 6923 20:38:11: 22 3rd Qu.: 48.00
184208_02_02_31: 1508 Max. :2.091e+09 12/10/2011: 6910 21:25:43: 22 Max. :150.00
(Other) :367692 (Other) :325981 (Other) :377817

streetlength speedlimit.mph. speedlimit.km.
speed_limit_compliance speed2 hour
Min. : 18.05 Min. :35 Min. :56.33 Min.:0.1065 Min. :0.0000 Min. : 0.00
1st Qu.:111.10 1st Qu.:35 1st Qu.:56.33 1st Qu.:0.3906 1st Qu.:0.0000 1st Qu.:9.00
Median :197.51 Median :35 Median :56.33 Median :0.6569 Median :0.0000 Median :13.00
Mean :243.35 Mean :35 Mean :56.33 Mean :0.6372 Mean :0.1021 Mean :13.18
3rd Qu.:291.75 3rd Qu.:35 3rd Qu.:56.33 3rd Qu.:0.8522 3rd Qu.:0.0000 3rd Qu.:16.00
```
<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed_25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_55</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>speed_65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>morning</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>afternoon</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>night</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>o_gender</td>
<td>Female:200978</td>
<td>Associates degree: 13270</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Male:176982</td>
<td>Bachelors degree: 126144</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o_edu</td>
<td>High school graduate: 58226</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Some college: 50968</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vocational/Technical training: 13254</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>o_age</td>
<td>gender</td>
<td>education_1</td>
<td>education_2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_25_to_34</td>
<td>age_35_to_44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min.</td>
<td>5.000</td>
<td>Min.</td>
<td>0.0000</td>
<td>Min.</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>1st Qu.</td>
<td>7.000</td>
<td>1st Qu.</td>
<td>0.0000</td>
<td>1st Qu.</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>9.000</td>
<td>Median</td>
<td>0.0000</td>
<td>Median</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>8.499</td>
<td>Mean</td>
<td>0.4683</td>
<td>Mean</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>10.000</td>
<td>3rd Qu.</td>
<td>1.0000</td>
<td>3rd Qu.</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Max.</td>
<td>12.000</td>
<td>Max.</td>
<td>1.0000</td>
<td>Max.</td>
<td>1.0000</td>
<td></td>
</tr>
</tbody>
</table>

| age_45_to_54 | age_55_to_64 | age_65_to_74 |
| age_75_to_84 | age_85_and_over |
| Min. | 0.0000 | Min. | 0.0000 | Min. | 0.0000 | Min. | 0.0000 |
| 1st Qu. | 0.0000 | 1st Qu. | 0.0000 | 1st Qu. | 0.0000 | 1st Qu. | 0.0000 |
| Median | 0.0000 | Median | 0.0000 | Median | 0.0000 | Median | 0.0000 |
| Mean | 0.2094 | Mean | 0.2694 | Mean | 0.1947 | Mean | 0.05336 |
| 3rd Qu. | 0.0000 | 3rd Qu. | 1.0000 | 3rd Qu. | 0.0000 | 3rd Qu. | 0.0000 |
| Max. | 1.0000 | Max. | 1.0000 | Max. | 1.0000 | Max. | 1.0000 |
```r
> plot(speed35$streetlength, speed35$speed_limit_compliance, pch = ".", ylim=c(0, 3), yaxp=c(0, 3, 6), xlab="Link length (Speed limit: 35mph)", ylab="driving speed/speed limit")
> lm_35.obj<-lm(speed35$speed_limit_compliance ~ speed35$streetlength)
> summary(lm_35.obj)

Call:
lm(formula = speed35$speed_limit_compliance ~ speed35$streetlength)

Residuals:
   Min     1Q  Median     3Q    Max
-1.06627 -0.21815  0.01676  0.21111  2.05986

Coefficients:                     Estimate Std. Error t value Pr(>|t|)
(Intercept)             5.780e-01  7.266e-04  795.5 <2e-16 ***
speed35$streetlength  2.433e-04  2.260e-06  107.6 <2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.292 on 377958 degrees of freedom
Multiple R-squared: 0.02974, Adjusted R-squared: 0.02974
F-statistic: 1.159e+04 on 1 and 377958 DF, p-value: < 2.2e-16

> abline(lm_35.obj, col=2)
> mtext("N=377,960, R^2=0.0297, y=0.5780+0.0002x", side=3)
> #plot (speed limit: 40 mph)
> speed40 <- subset(res, res$speedlimit.mph."40")
```

265
> summary(speed40)

<table>
<thead>
<tr>
<th>filename</th>
<th>tripid</th>
<th>date</th>
<th>time</th>
<th>speed</th>
<th>kilometer</th>
</tr>
</thead>
<tbody>
<tr>
<td>187010_02_02_1</td>
<td>2233</td>
<td>10/10/2011</td>
<td>21:12:31</td>
<td>Min. :1.013e+09</td>
<td>9230</td>
</tr>
<tr>
<td>183019_02_02_21</td>
<td>2167</td>
<td>28/9/2011</td>
<td>22:42:22</td>
<td>1st Qu.:1.254e+09</td>
<td>8522</td>
</tr>
<tr>
<td>183019_01_02_23</td>
<td>1737</td>
<td>11/10/2011</td>
<td>21:12:29</td>
<td>Median :1.520e+09</td>
<td>8194</td>
</tr>
<tr>
<td>157551_01_02_12</td>
<td>1242</td>
<td>29/9/2011</td>
<td>21:12:30</td>
<td>Mean :1.510e+09</td>
<td>8160</td>
</tr>
<tr>
<td>157551_01_02_30</td>
<td>1239</td>
<td>22/7/2011</td>
<td>21:12:28</td>
<td>3rd Qu.:1.830e+09</td>
<td>5891</td>
</tr>
<tr>
<td>194902_02_02_40</td>
<td>1087</td>
<td>12/10/2011</td>
<td>21:12:32</td>
<td>Max. :2.091e+09</td>
<td>5700</td>
</tr>
<tr>
<td>(Other) :337171</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Other) :346744</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

streetlength speedlimit.mph. speedlimit.km. speed_limit_compliance speed2 hour

<table>
<thead>
<tr>
<th>Min.</th>
<th>Min.</th>
<th>Min.</th>
<th>Min.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09321</td>
<td>0.0000</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1st Qu.:</td>
<td>1st Qu.:</td>
<td>1st Qu.:</td>
<td>1st Qu.:</td>
</tr>
<tr>
<td>135.87</td>
<td>140.00</td>
<td>143.71</td>
<td>147.00</td>
</tr>
<tr>
<td>Median</td>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>203.00</td>
<td>214.50</td>
<td>227.37</td>
<td>240.00</td>
</tr>
<tr>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>284.99</td>
<td>297.45</td>
<td>310.00</td>
<td>322.50</td>
</tr>
<tr>
<td>3rd Qu.:</td>
<td>3rd Qu.:</td>
<td>3rd Qu.:</td>
<td>3rd Qu.:</td>
</tr>
<tr>
<td>344.86</td>
<td>357.44</td>
<td>370.00</td>
<td>382.50</td>
</tr>
<tr>
<td>Max.</td>
<td>Max.</td>
<td>Max.</td>
<td>Max.</td>
</tr>
<tr>
<td>3074.42</td>
<td>3207.00</td>
<td>3339.57</td>
<td>3472.10</td>
</tr>
<tr>
<td></td>
<td>speed_25</td>
<td>speed_30</td>
<td>speed_35</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Min.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Min.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Median</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>morning</th>
<th>afternoon</th>
<th>night</th>
<th>o_gender</th>
<th>o_edu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>Female</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>Male</td>
</tr>
<tr>
<td>Median</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.3391</td>
<td>0.4863</td>
<td>0.1673</td>
<td></td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Max.</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td></td>
</tr>
</tbody>
</table>

- Associates degree: 12172
- Bachelors degree: 131846
- Graduate/Post-graduate degree: 108089
- High school graduate: 33423
- Some college: 52847
- Vocational/Technical training: 8499
Min. : 5.000  Min. :0.0000  Min. :0.0000  Min. :0.0000
1st Qu.: 8.000  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000
Median : 9.000  Median :0.0000  Median :0.0000  Median :1.0000
Mean : 8.766  Mean :0.3806  Mean :0.1874  Mean :0.6917
3rd Qu.:10.000  3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:0.0000
Max. :12.000  Max. :1.0000  Max. :1.0000  Max. :1.0000

age_45_to_54  age_55_to_64  age_65_to_74
age_75_to_84  age_85_and_over
Min. :0.0000  Min. :0.0000  Min. :0.0000  Min. :0.0000
1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000
Median :0.0000  Median :0.0000  Median :0.0000  Median :0.0000
Mean :0.1502  Mean :0.3385  Mean :0.2117  Mean :0.06489
3rd Qu.:0.0000  3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:0.0000
Max. :1.0000  Max. :1.0000  Max. :1.0000  Max. :1.0000

> plot(speed40$streetlength, speed40$speed_limit_compliance, pch = ".", ylim=c(0, 3), yaxp=c(0, 3, 6), xlab="Link length (Speed limit: 40mph)", ylab="driving speed/speed limit")
```r
> lm_40.obj <- lm(speed40$speed_limit_compliance ~ speed40$streetlength)
> abline(lm_40.obj, col=2)
> summary(lm_40.obj)

Call:
  lm(formula = speed40$speed_limit_compliance ~ speed40$streetlength)

Residuals:
   Min     1Q   Median     3Q    Max
-1.8500 -0.1463  0.0350  0.1724  5.3431

Coefficients:
                          Estimate Std. Error   t value     Pr(>|t|)
  (Intercept)              6.671e-01  7.175e-04     929.7 <2e-16 ***
speed40$streetlength     4.151e-04  1.913e-06     217.0 <2e-16 ***

---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05  . 0.1 1

Residual standard error: 0.2748 on 346874 degrees of freedom
Multiple R-squared:  0.1195 ,  Adjusted R-squared:  0.1195
F-statistic: 4.708e+04 on 1 and 346874 DF,  p-value: < 2.2e-16

> mtext("N=346,876, R^2=0.1195, y=0.6671+0.0004x", side=3)
>
A.4.22 Regression analysis in R

> fnames <- dir(pattern=".csv")
> csvlist <- lapply(fnames, read.csv)
> names(csvlist) <- fnames
```

269
> n <- length(csvlist)
> n
[1] 11
> temp <- csvlist[[1]]
> for (i in 2:n) {temp <- merge(temp, csvlist[[i]], all=T)}
> res <- temp
> #Multi Linear Regression
> regression = lm(speed_limit_compliance ~ streetlength + speed_25 + speed_30 + speed_35 + speed_40 + speed_45 + speed_50 + speed_55 + speed_60 + speed_65 + age_25_to_34 + age_35_to_44 + age_45_to_54 + age_55_to_64 + age_65_to_74 + age_75_to_84 + age_85_and_over + gender + education_1 + education_2 + morning + afternoon + night, res)
> summary(regression)

Call:
  lm(formula = speed_limit_compliance ~ streetlength + speed_25 + speed_30 + speed_35 + speed_40 + speed_45 + speed_50 + speed_55 + speed_60 + speed_65 + age_25_to_34 + age_35_to_44 + age_45_to_54 + age_55_to_64 + age_65_to_74 + age_75_to_84 + age_85_and_over + gender + education_1 + education_2 + morning + afternoon + night, data = res)

Residuals:
     Min      1Q  Median       3Q      Max
-1.3491 -0.1723  0.0337  0.1780  6.2159

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.878e-01  4.433e-03  155.171  < 2e-16 ***
street length 1.650e-04 9.496e-07 173.801 < 2e-16 ***
speed_25 6.365e-01 3.659e-03 173.934 < 2e-16 ***
speed_30 -9.581e-02 3.302e-03 -29.015 < 2e-16 ***
speed_35 -1.035e-01 3.219e-03 -32.139 < 2e-16 ***
speed_40 3.614e-02 3.683e-03 -29.015 < 2e-16 ***
speed_45 6.506e-02 3.683e-03 17.665 < 2e-16 ***
speed_50 4.953e-02 3.734e-03 13.265 < 2e-16 ***
speed_55 1.226e-01 3.199e-03 38.317 < 2e-16 ***
speed_60 1.166e-01 3.191e-03 36.527 < 2e-16 ***

age 25 to 34 5.604e-02 2.406e-03 23.292 < 2e-16 ***
age 35 to 44 1.221e-02 2.235e-03 5.462 4.70e-08 ***
age 45 to 54 3.321e-02 2.180e-03 19.246 < 2e-16 ***
age 55 to 64 3.896e-02 2.222e-03 14.976 < 2e-16 ***
age 65 to 74 7.844e-02 3.478e-03 22.555 < 2e-16 ***
age 75 to 84 3.389e-03 2.387e-03 1.420 0.1557

age 85 and over 8.744e-02 2.832e-03 30.881 < 2e-16 ***
gender -1.341e-02 5.658e-04 -23.704 < 2e-16 ***
education_1 -4.808e-03 9.801e-04 -4.906 9.30e-07 ***
education_2 -1.705e-02 8.001e-04 -21.314 < 2e-16 ***
morning 4.021e-03 2.238e-03 1.796 0.0724
afternoon -1.322e-02 2.228e-03 -5.934 2.95e-09 ***
night 1.595e-02 2.288e-03 6.972 3.12e-12 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.2898 on 1172079 degrees of freedom
Multiple R-squared: 0.2269, Adjusted R-squared: 0.2269
F-statistic: 1.496e+04 on 23 and 1172079 DF, p-value: < 2.2e-16
> write.csv (summary(regression2)$coef, "regression2.csv")
>
> #Regression by street length
> regression2 = lm(speed_limit_compliance ~ streetlength , res)
> summary(regression2)

Call:
lm(formula = speed_limit_compliance ~ streetlength , data = res)

Residuals:
   Min     1Q Median     3Q    Max
-1.4096 -0.2039  0.0400  0.1957  6.8803

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept)  6.679e-01  4.162e-04 1604.8   <2e-16 ***
streetlength 2.716e-04  8.212e-07   330.7   <2e-16 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05
               ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3152 on 1172101 degrees of freedom
Multiple R-squared:  0.08536 ,  Adjusted R-squared:  0.08536
F-statistic: 1.094e+05 on 1 and 1172101 DF,  p-value: < 2.2e-16

> write.csv (summary(regression22)$coef, "regression22.csv")
>
> #Regression by speed limit
> regression22 = lm(speed_limit_compliance ~ speed_25+speed_30+
>                  speed_35+speed_40+speed_45+speed_50+speed_55+speed_60+
>                  speed_65 , res)
> summary(regression22)
Call:

\[
\text{lm(formula = speed\_limit\_compliance ~ speed\_25 + speed\_30 + speed\_35 + speed\_40 + speed\_45 + speed\_50 + speed\_55 + speed\_60 + speed\_65, data = res)}
\]

Residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1.2500</td>
<td>-0.1756</td>
<td>0.0281</td>
<td>0.1794</td>
<td>6.1816</td>
</tr>
</tbody>
</table>

Coefficients:

|                  | Estimate | Std. Error | t value | Pr(>|t|) |
|------------------|----------|------------|---------|-------|
| (Intercept)      | 0.939460 | 0.003022   | 310.909 | < 2e-16 *** |
| speed\_25        | 0.459701 | 0.003568   | 128.836 | < 2e-16 *** |
| speed\_30        | -0.306821 | 0.003127   | -98.108 | < 2e-16 *** |
| speed\_35        | -0.302239 | 0.003059   | -98.792 | < 2e-16 *** |
| speed\_40        | -0.154104 | 0.003063   | -50.316 | < 2e-16 *** |
| speed\_45        | -0.109113 | 0.003600   | -30.307 | < 2e-16 *** |
| speed\_50        | -0.103670 | 0.003669   | -28.253 | < 2e-16 *** |
| speed\_55        | -0.012994 | 0.003141   | -4.137  | 3.52e-05 *** |
| speed\_60        | 0.008102  | 0.003175   | 2.552   | 0.0107 * |
| speed\_65        | -0.006673 | 0.003490   | -1.912  | 0.0559 . |

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.2944 on 1172093 degrees of freedom
Multiple R-squared: 0.2021 , Adjusted R-squared: 0.2021
F-statistic: 3.299e+04 on 9 and 1172093 DF, p-value: < 2.2e-16
> write.csv(summary(regression2)$coef, "regression2.csv")
>
> # Regression by age
> regression3=lm(speed_limit_compliance~age.25.to.34+
>                  age.35.to.44+age.45.to.54+age.55.to.64+age.65.to.74+
>                  age.75.to.84+age.85.and.over, res)
> summary(regression3)

Call:
lm(formula = speed_limit_compliance ~ age.25.to.34 +
           age.35.to.44 +
           age.45.to.54 + age.55.to.64 + age.65.to.74 + age.75.to.84 +
           age.85.and.over, data = res)

Residuals:
   Min     1Q  Median     3Q    Max
-0.7798 -0.2253  0.0430  0.2206  6.8169

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)  0.702842   0.002281 308.1611 < 2e-16 ***
age.25.to.34 0.117389   0.002571  45.6620 < 2e-16 ***
age.35.to.44 0.038872   0.002397  16.2191 < 2e-16 ***
age.45.to.54 0.049881   0.002391  20.8616 < 2e-16 ***
age.55.to.64 0.082682   0.002345  35.2545 < 2e-16 ***
age.65.to.74 0.060960   0.002390  25.5092 < 2e-16 ***
age.75.to.84 0.013250   0.002602   5.0933 3.53e-07 ***
age.85.and.over 0.139055  0.003099  44.8688 < 2e-16 ***

Signif. codes:  0 ***  0.001 **  0.01 *  0.05

. 0.1 1
Residual standard error: 0.3284 on 1172095 degrees of freedom
Multiple R-squared: 0.007174, Adjusted R-squared: 0.007168
F-statistic: 1210 on 7 and 1172095 DF, p-value: < 2.2e-16

> write.csv(summary(regression3)$coef, ”regression3.csv”)
>
> #Regression by gender
> reg=lm(speed_limit_compliance~gender, res)
> summary(reg)

Call:
  lm(formula = speed_limit_compliance ~ gender, data = res)

Residuals:
   Min     1Q Median     3Q    Max
-0.7178 -0.2248  0.0327  0.2191  6.8055

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.7751797  0.0004035 1921.23  <2e-16 ***
gender      -0.0207935  0.0006144  -33.84  <2e-16 ***

---
Signif. codes:  0 ***  0.001 **  0.01 *  0.05 .  0.1  1

Residual standard error: 0.3294 on 1172101 degrees of freedom
Multiple R-squared: 0.0009762, Adjusted R-squared: 0.0009754
F-statistic: 1145 on 1 and 1172101 DF, p-value: < 2.2e-16

> write.csv(summary(reg)$coef, ”regression4.csv”)
>
> #Regression by education
> reg2=lm(speed_limit_compliance~education_1+education_2,res)
> summary(reg2)

Call:
  lm(formula = speed_limit_compliance ~ education_1 + education_2,
     data = res)

Residuals:
     Min      1Q  Median      3Q     Max
-0.7195 -0.2265  0.0442  0.2193  6.8142

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.776832   0.000768  1010.81  <2e-16 ***
education_1             -0.010309   0.001045  -9.861  <2e-16 ***
education_2             -0.013237   0.000855 -15.479  <2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.3296 on 1172100 degrees of freedom
Multiple R-squared:  0.0002046, Adjusted R-squared:  0.0002029
F-statistic: 119.9 on 2 and 1172100 DF,  p-value: < 2.2e-16

> write.csv(summary(reg2)$coeff, "regression5.csv")
>
> #Regression by time
> reg3=lm(speed_limit_compliance~morning + afternoon + night, res)
> summary(reg3)
Call:
```
lm(formula = speed_limit_compliance ~ morning + afternoon +
    night, 
    data = res)
```

Residuals:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>-0.7744</td>
<td></td>
<td>-0.2234</td>
<td></td>
<td>0.0375</td>
</tr>
<tr>
<td>1Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.2234</td>
<td></td>
<td>0.0375</td>
<td></td>
<td>0.2229</td>
</tr>
<tr>
<td>Median</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0375</td>
<td></td>
<td>0.2229</td>
<td></td>
<td>6.8296</td>
</tr>
<tr>
<td>Max</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | 0.848985 | 0.002470 | 343.76 | <2e-16 *** |
| morning | -0.078751 | 0.002523 | -31.22 | <2e-16 *** |
| afternoon | -0.097900 | 0.002510 | -39.01 | <2e-16 *** |
| night | -0.057023 | 0.002577 | -22.12 | <2e-16 *** |

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.3291 on 1172099 degrees of freedom
Multiple R-squared: 0.003026 , Adjusted R-squared: 0.003024
F-statistic: 1186 on 3 and 1172099 DF, p-value: < 2.2e-16

```text
write.csv(summary(reg3)$coef, "regression6.csv")
```

A.4.23 Dummy variable regression in R

R version 3.1.2 (2014-10-31) --- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.65 (6833) x86_64-apple-darwin10.8.0]

[Workspace restored from /Users/toshiyokoo/.RData]
[History restored from /Users/toshiyokoo/.Rapp.history]

> fnames <- dir(pattern=".csv")
> csvlist <- lapply(fnames, read.csv)
> names(csvlist) <- fnames
> n <- length(csvlist)
> n

[1] 1
> temp <- csvlist[[1]]
> res <- temp
> summary(res)

    PersonID    TripID   LinkID           Hour    streetlength   Speedlimit
degree 278
<table>
<thead>
<tr>
<th></th>
<th>187010_02_02_1</th>
<th>441</th>
<th>186319_02_02_44</th>
<th>270</th>
<th>144314_01_02_40</th>
<th>255</th>
<th>279</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>1.013e+09</td>
<td></td>
<td>0.0000000000</td>
<td></td>
<td>0.0000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min.</td>
<td>0.0000000000</td>
<td></td>
<td>0.0000000000</td>
<td></td>
<td>0.0000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st Qu.</td>
<td>1.219e+09</td>
<td>336</td>
<td>101.03</td>
<td>50947</td>
<td>101.03</td>
<td>50947</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>10.00</td>
<td></td>
<td>1st Qu.</td>
<td>10.00</td>
<td>1st Qu.</td>
<td>10.00</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.05736</td>
<td></td>
<td>1st Qu.</td>
<td>0.05736</td>
<td>1st Qu.</td>
<td>0.05736</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>Median</td>
<td>1.477e+09</td>
<td>284</td>
<td>Median</td>
<td>165.53</td>
<td>Median</td>
<td>40.00</td>
<td>Median</td>
</tr>
<tr>
<td>Median</td>
<td>13.00</td>
<td></td>
<td>Median</td>
<td>13.25</td>
<td>Mean</td>
<td>13.25</td>
<td>Mean</td>
</tr>
<tr>
<td>Mean</td>
<td>1.481e+09</td>
<td></td>
<td>Mean</td>
<td>254.53</td>
<td>Max.</td>
<td>23.00</td>
<td>Max.</td>
</tr>
<tr>
<td>Max.</td>
<td>2.091e+09</td>
<td></td>
<td>Max.</td>
<td>4841.32</td>
<td>Max.</td>
<td>70.00</td>
<td>Max.</td>
</tr>
<tr>
<td>(Other)</td>
<td>113689</td>
<td></td>
<td>(Other)</td>
<td>113689</td>
<td>(Other)</td>
<td>113689</td>
<td>(Other)</td>
</tr>
<tr>
<td>speed_25</td>
<td>0.0000000000</td>
<td></td>
<td>speed_25</td>
<td>0.0000000000</td>
<td>speed_25</td>
<td>0.0000000000</td>
<td>speed_25</td>
</tr>
<tr>
<td>speed_30</td>
<td>0.0000000000</td>
<td></td>
<td>speed_30</td>
<td>0.0000000000</td>
<td>speed_30</td>
<td>0.0000000000</td>
<td>speed_30</td>
</tr>
<tr>
<td>speed_35</td>
<td>0.0000000000</td>
<td></td>
<td>speed_35</td>
<td>0.0000000000</td>
<td>speed_35</td>
<td>0.0000000000</td>
<td>speed_35</td>
</tr>
<tr>
<td>speed_40</td>
<td>0.0000000000</td>
<td></td>
<td>speed_40</td>
<td>0.0000000000</td>
<td>speed_40</td>
<td>0.0000000000</td>
<td>speed_40</td>
</tr>
<tr>
<td>speed_45</td>
<td>0.0000000000</td>
<td></td>
<td>speed_45</td>
<td>0.0000000000</td>
<td>speed_45</td>
<td>0.0000000000</td>
<td>speed_45</td>
</tr>
<tr>
<td>speed_50</td>
<td>0.0000000000</td>
<td></td>
<td>speed_50</td>
<td>0.0000000000</td>
<td>speed_50</td>
<td>0.0000000000</td>
<td>speed_50</td>
</tr>
</tbody>
</table>

279
<table>
<thead>
<tr>
<th>Variable</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed_55</td>
<td>0.0000</td>
<td>0.0775</td>
<td>0.10</td>
<td>0.3425</td>
<td>1.00</td>
</tr>
<tr>
<td>speed_60</td>
<td>0.0000</td>
<td>0.0775</td>
<td>0.10</td>
<td>0.3425</td>
<td>1.00</td>
</tr>
<tr>
<td>speed_65</td>
<td>0.0000</td>
<td>0.0775</td>
<td>0.10</td>
<td>0.3425</td>
<td>1.00</td>
</tr>
<tr>
<td>morning</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.00</td>
</tr>
<tr>
<td>afternoon</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.00</td>
</tr>
<tr>
<td>night</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.00</td>
</tr>
<tr>
<td>age</td>
<td>0.0000</td>
<td>0.0975</td>
<td>0.10</td>
<td>0.3525</td>
<td>1.00</td>
</tr>
<tr>
<td>gender</td>
<td>0.0000</td>
<td>0.0975</td>
<td>0.10</td>
<td>0.3525</td>
<td>1.00</td>
</tr>
<tr>
<td>education</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.00</td>
</tr>
<tr>
<td>education_2</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.00</td>
</tr>
<tr>
<td>Associates degree</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.00</td>
</tr>
<tr>
<td>Education Level</td>
<td>Count</td>
<td>Mean</td>
<td>Median</td>
<td>1st Quartile</td>
<td>3rd Quartile</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>--------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Bachelors degree</td>
<td>40086</td>
<td>8.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Graduate/Post-graduate degree</td>
<td>35579</td>
<td>9.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>High school graduate</td>
<td>14006</td>
<td>8.639</td>
<td>0.427</td>
<td>0.1848</td>
<td>0.6549</td>
</tr>
<tr>
<td>Some college</td>
<td>17112</td>
<td>8.000</td>
<td>1.000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Vocational/Technical training</td>
<td>4515</td>
<td>12.000</td>
<td>1.000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age Groups</th>
<th>Count</th>
<th>Mean</th>
<th>Median</th>
<th>1st Quartile</th>
<th>3rd Quartile</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>age_35_to_44</td>
<td></td>
<td>0.1562</td>
<td>0.1857</td>
<td>0.3105</td>
<td>0.1869</td>
<td>0.01064</td>
</tr>
<tr>
<td>age_45_to_54</td>
<td></td>
<td>0.06209</td>
<td>0.02317</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_55_to_64</td>
<td></td>
<td>0.1000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.00000</td>
</tr>
<tr>
<td>age_65_to_74</td>
<td></td>
<td>0.0900</td>
<td>0.0900</td>
<td>0.0900</td>
<td>0.0900</td>
<td>0.09000</td>
</tr>
<tr>
<td>age_75_to_84</td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00000</td>
</tr>
<tr>
<td>age_85_and_over</td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

281
<table>
<thead>
<tr>
<th>person2</th>
<th>person3</th>
<th>person4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>Min.</td>
<td>Min.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>1st Qu.</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>0.007573</td>
<td>0.002423</td>
<td>0.004639</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>Max.</td>
<td>Max.</td>
</tr>
<tr>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person5</th>
<th>person6</th>
<th>person7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>Min.</td>
<td>Min.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>1st Qu.</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>0.008932</td>
<td>0.005202</td>
<td>0.005868</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>Max.</td>
<td>Max.</td>
</tr>
<tr>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person8</th>
<th>person9</th>
<th>person10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>Min.</td>
<td>Min.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>1st Qu.</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>0.004683</td>
<td>0.003116</td>
<td>0.01235</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>Max.</td>
<td>Max.</td>
</tr>
<tr>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person11</th>
<th>person12</th>
<th>person13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>Min.</td>
<td>Min.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>1st Qu.</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>0.000666</td>
<td>0.004085</td>
<td>0.005236</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>Max.</td>
<td>Max.</td>
</tr>
<tr>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

282
<table>
<thead>
<tr>
<th>person14</th>
<th>person15</th>
<th>person16</th>
</tr>
</thead>
<tbody>
<tr>
<td>person17</td>
<td>person18</td>
<td>person19</td>
</tr>
<tr>
<td>person20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Min.</th>
<th>Min.</th>
<th>Min.</th>
<th>Min.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st Qu.</th>
<th>1st Qu.</th>
<th>1st Qu.</th>
<th>1st Qu.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Median</th>
<th>Median</th>
<th>Median</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mean</th>
<th>Mean</th>
<th>Mean</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01033</td>
<td>0.00431</td>
<td>0.009876</td>
<td>0.005401</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3rd Qu.</th>
<th>3rd Qu.</th>
<th>3rd Qu.</th>
<th>3rd Qu.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max.</th>
<th>Max.</th>
<th>Max.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person21</th>
<th>person22</th>
<th>person23</th>
</tr>
</thead>
<tbody>
<tr>
<td>person24</td>
<td>person25</td>
<td>person26</td>
</tr>
<tr>
<td>person27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Min.</th>
<th>Min.</th>
<th>Min.</th>
<th>Min.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st Qu.</th>
<th>1st Qu.</th>
<th>1st Qu.</th>
<th>1st Qu.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st Qu.</th>
<th>1st Qu.</th>
<th>1st Qu.</th>
<th>1st Qu.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>person28</td>
<td>person29</td>
<td>person30</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.010217</td>
<td>0.011467</td>
<td>0.015741</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person31</th>
<th>person32</th>
<th>person33</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.005202</td>
<td>0.003436</td>
<td>0.002848</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.004406</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>person35</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>Min.</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.008664</td>
</tr>
<tr>
<td></td>
<td>0.008543</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
</tr>
<tr>
<td></td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person38</th>
<th>person39</th>
<th>person40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.008205</td>
<td>0.005756</td>
<td>0.008127</td>
</tr>
<tr>
<td></td>
<td>0.003523</td>
<td>0.007617</td>
<td>0.001471</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td></td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person41</th>
<th>person42</th>
<th>person43</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.008205</td>
<td>0.005756</td>
<td>0.008127</td>
</tr>
<tr>
<td></td>
<td>0.003523</td>
<td>0.007617</td>
<td>0.001471</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td></td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>Max</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person47</td>
<td>person48</td>
<td>person49</td>
<td>person50</td>
</tr>
<tr>
<td>Min</td>
<td>Min</td>
<td>Min</td>
<td>Min</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>1st Qu.</td>
<td>1st Qu.</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>0.01146</td>
<td>0.02476</td>
<td>0.006065</td>
<td>0.00882</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
<td>3rd Qu.</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person54</td>
<td>person55</td>
<td>person56</td>
<td>person57</td>
</tr>
<tr>
<td>Min</td>
<td>Min</td>
<td>Min</td>
<td>Min</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>person58</td>
<td>person59</td>
<td>person60</td>
<td>Min</td>
</tr>
<tr>
<td>Min</td>
<td>Min</td>
<td>Min</td>
<td>Min</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

286
1st Qu.: 0.0000 1st Qu.: 0.000000 1st Qu.: 0.000000 1st Qu.
.: 0.000000 1st Qu.: 0.000000 1st Qu.: 0.000000 1st Qu.
.: 0.000000
Median : 0.0000  Median : 0.000000  Median : 0.000000  Median
.: 0.000000  Median : 0.000000  Median : 0.000000  Median
.: 0.000000
Mean : 0.0074 Mean : 0.007227 Mean : 0.008231 Mean
.: 0.007816 Mean : 0.01136 Mean : 0.005609 Mean
.: 0.002363
3rd Qu.: 0.0000  3rd Qu.: 0.000000  3rd Qu.: 0.000000  3rd Qu.
.: 0.000000  3rd Qu.: 0.000000  3rd Qu.: 0.000000  3rd Qu.
.: 0.000000
Max. : 1.0000 Max. : 1.000000 Max. : 1.000000 Max.
.: 1.000000 Max. : 1.000000 Max. : 1.000000 Max.
.: 1.000000

person61 person62 person63
person64 person65 person66
person67
Min. : 0.0000 Min. : 0.0000 Min. : 0.000000 Min.
.: 0.000000 Min. : 0.000000 Min. : 0.000000 Min.
.: 0.000000
1st Qu.: 0.0000 1st Qu.: 0.0000 1st Qu.: 0.000000 1st Qu.
.: 0.000000 1st Qu.: 0.000000 1st Qu.: 0.000000 1st Qu.
.: 0.000000
Median : 0.0000 Median : 0.0000 Median : 0.000000 Median
.: 0.000000 Median : 0.000000 Median : 0.000000 Median
.: 0.000000
Mean : 0.01465 Mean : 0.01073 Mean : 0.006976 Mean
.: 0.001082 Mean : 0.002882 Mean : 0.002155 Mean
.: 0.002311

287
<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>person68</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.001965</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person69</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.009590</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person70</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.008413</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person71</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.009010</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person72</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.002804</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person73</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.005184</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person74</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.008223</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person75</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.004250</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person76</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.006509</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person77</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.004345</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person78</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.008526</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person79</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.004302</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

288
<table>
<thead>
<tr>
<th></th>
<th>person80</th>
<th>person81</th>
<th>person82</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0003938</td>
<td>0.005427</td>
<td>0.013</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person83</th>
<th>person84</th>
<th>person85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0008231</td>
<td>0.01206</td>
<td>0.00927</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person86</th>
<th>person87</th>
<th>person88</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0006665</td>
<td>0.01206</td>
<td>0.00927</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person89</th>
<th>person90</th>
<th>person91</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.000927</td>
<td>0.01206</td>
<td>0.00927</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

289
<table>
<thead>
<tr>
<th></th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>person93</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.00429</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person94</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.01441</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person95</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.003029</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person96</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.004181</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person97</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.006154</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person98</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.005366</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person99</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.001441</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person100</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.003102</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person101</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.007539</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person102</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.004864</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person103</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.005176</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person104</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.004864</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>person105</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.004864</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td></td>
<td>person106</td>
<td>person107</td>
<td>person108</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st Qu.:</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Median:</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean:</td>
<td>0.002112</td>
<td>0.00167</td>
<td>0.01035</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.00335</td>
<td>0.008422</td>
<td>0.004293</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.002475</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person109</th>
<th>person110</th>
<th>person111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.:</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median:</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean:</td>
<td>0.007011</td>
<td>0.005089</td>
<td>0.004163</td>
</tr>
<tr>
<td></td>
<td>0.003852</td>
<td>0.00438</td>
<td>0.001567</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person112</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.:</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>Median:</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean:</td>
<td>0.001567</td>
</tr>
<tr>
<td>Min.</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

Min. : 0.000000  Min. : 0.000000  Min. : 0.000000  Min. : 0.000000  Min. : 0.000000  Min. : 0.000000  Min. : 0.000000  Min. : 0.000000

3rd Qu. : 0.000000  3rd Qu. : 0.000000  3rd Qu. : 0.000000  3rd Qu. : 0.000000  3rd Qu. : 0.000000  3rd Qu. : 0.000000  3rd Qu. : 0.000000  3rd Qu. : 0.000000


person113 person114 person115
person116 person117 person118
person119

person120 person121 person122
person123 person124 person125
person126
Min. : 0.00000  Min. : 0.00000  Min. : 0.00000  Min. : 0.00000
Min. : 0.00000  Min. : 0.00000  Min. : 0.00000  Min. : 0.00000
1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000
1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000
Median : 0.00000  Median : 0.00000  Median : 0.00000  Median : 0.00000
Median : 0.00000  Median : 0.00000  Median : 0.00000  Median : 0.00000
Mean : 0.01017  Mean : 0.004354  Mean : 0.009902  Mean : 0.01721
Mean : 0.02052  Mean : 0.005133  Mean : 0.003817
3rd Qu.: 0.00000  3rd Qu.: 0.00000  3rd Qu.: 0.00000  3rd Qu.: 0.00000
3rd Qu.: 0.00000  3rd Qu.: 0.00000  3rd Qu.: 0.00000  3rd Qu.: 0.00000
Max. : 1.00000  Max. : 1.00000  Max. : 1.00000  Max. : 1.00000
Max. : 1.00000  Max. : 1.00000  Max. : 1.00000  Max. : 1.00000

column127  column128  column129

column130  column131  column132

column133
Min. : 0.00000  Min. : 0.00000  Min. : 0.00000  Min. : 0.00000
Min. : 0.00000  Min. : 0.00000  Min. : 0.00000  Min. : 0.00000
1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000
1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000  1st Qu.: 0.00000
Median : 0.00000  Median : 0.00000  Median : 0.00000  Median : 0.00000
Median : 0.00000  Median : 0.00000  Median : 0.00000  Median : 0.00000

293
<table>
<thead>
<tr>
<th></th>
<th>person134</th>
<th>person135</th>
<th>person136</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.005548</td>
<td>0.0004501</td>
<td>0.007824</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.01168</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person137</th>
<th>person138</th>
<th>person139</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.005505</td>
<td>0.002397</td>
<td>0.01166</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person140</th>
<th>person141</th>
<th>person142</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0054451</td>
<td>0.0004501</td>
<td>0.0076824</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.01168</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person143</th>
<th>person144</th>
<th>person145</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.005892</td>
<td>0.0004501</td>
<td>0.01166</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person146</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.005871</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person147</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.005871</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person148</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.000000</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median</td>
<td>0.000000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.005871</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
</tr>
<tr>
<td></td>
<td>Median</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.004371</td>
</tr>
<tr>
<td></td>
<td>0.004778</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
</tr>
<tr>
<td></td>
<td>1.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>Mean</th>
<th>Median</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.005479</td>
<td>0.000000</td>
<td>0.008482</td>
<td>0.000000</td>
<td>0.00631</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.004622</td>
<td>0.000000</td>
<td>0.003453</td>
<td>0.000000</td>
<td>0.004631</td>
<td>0.000000</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Max.</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

> #Multi Linear Regression

295
> regression = lm(degree ~ streetlength + speed_25 + speed_30 + speed_35 + speed_40 + speed_45 + speed_50 + speed_55 + speed_60 + speed_65 + age_25_to_34 + age_35_to_44 + age_45_to_54 + age_55_to_64 + age_65_to_74 + age_75_to_84 + age_85_and_over + gender + education_1 + education_2 + morning + afternoon + night + person1 + person2 + person3 + person4 + person5 + person6 + person7 + person8 + person9 + person10 + person11 + person12 + person13 + person14 + person15 + person16 + person17 + person18 + person19 + person20 + person21 + person22 + person23 + person24 + person25 + person26 + person27 + person28 + person29 + person30 + person31 + person32 + person33 + person34 + person35 + person36 + person37 + person38 + person39 + person40 + person41 + person42 + person43 + person44 + person45 + person46 + person47 + person48 + person49 + person50 + person51 + person52 + person53 + person54 + person55 + person56 + person57 + person58 + person59 + person60 + person61 + person62 + person63 + person64 + person65 + person66 + person67 + person68 + person69 + person70 + person71 + person72 + person73 + person74 + person75 + person76 + person77 + person78 + person79 + person80 + person81 + person82 + person83 + person84 + person85 + person86 + person87 + person88 + person89 + person90 + person91 + person92 + person93 + person94 + person95 + person96 + person97 + person98 + person99 + person100 + person101 + person102 + person103 + person104 + person105 + person106 + person107 + person108 + person109 + person110 + person111 + person112 + person113 + person114 + person115 + person116 + person117 + person118 + person119 + person120 + person121 + person122 + person123 + person124 + person125 + person126 + person127 + person128 + person129 + person130 + person131 + person132 + person133 + person134 + person135 + person136 + person137 + person138 + person139 + person140 + person141 + person142 + person143 + person144 + person145 + person146 + person147 + person148 + person149 + person150 + person151, res)
> summary(regression)
Call:

\[
\text{lm(formula = degree ~ streetlength + speed.25 + speed.30 + speed.35 + speed.40 + speed.45 + speed.50 + speed.55 + speed.60 + speed.65 + age.25_to.34 + age.35_to.44 + age.45_to.54 + age.55_to.64 + age.65_to.74 + age.75_to.84 + age.85_and_over + gender + education.1 + education.2 + morning + afternoon + night + person1 + person2 + person3 + person4 + person5 + person6 + person7 + person8 + person9 + person10 + person11 + person12 + person13 + person14 + person15 + person16 + person17 + person18 + person19 + person20 + person21 + person22 + person23 + person24 + person25 + person26 + person27 + person28 + person29 + person30 + person31 + person32 + person33 + person34 + person35 + person36 + person37 + person38 + person39 + person40 + person41 + person42 + person43 + person44 + person45 + person46 + person47 + person48 + person49 + person50 + person51 + person52 + person53 + person54 + person55 + person56 + person57 + person58 + person59 + person60 + person61 + person62 + person63 + person64 + person65 + person66 + person67 + person68 + person69 + person70 + person71 + person72 +}
\]
person73 + person74 + person75 + person76 + person77 +
person78 +
person79 + person80 + person81 + person82 + person83 +
person84 +
person85 + person86 + person87 + person88 + person89 +
person90 +
person91 + person92 + person93 + person94 + person95 +
person96 +
person97 + person98 + person99 + person100 + person101 +
person102 + person103 + person104 + person105 + person106 +
person107 + person108 + person109 + person110 + person111 +
person112 + person113 + person114 + person115 + person116 +
person117 + person118 + person119 + person120 + person121 +
person122 + person123 + person124 + person125 + person126 +
person127 + person128 + person129 + person130 + person131 +
person132 + person133 + person134 + person135 + person136 +
person137 + person138 + person139 + person140 + person141 +
person142 + person143 + person144 + person145 + person146 +
person147 + person148 + person149 + person150 + person151,
data = res)

Residuals:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>1Q</td>
<td>Median</td>
<td>3Q</td>
<td>Max</td>
</tr>
<tr>
<td>−1.3912</td>
<td>−0.1497</td>
<td>0.0225</td>
<td>0.1530</td>
<td>6.1722</td>
</tr>
</tbody>
</table>

Coefficients: (10 not defined because of singularities)

|                          | Estimate | Std. Error | t value | Pr(>|t|) |
|--------------------------|----------|------------|---------|---------|
| (Intercept)              | 8.386e−01| 2.390e−02  | 35.082  | < 2e−16 *** |
| streetlength             | 1.280e−04| 3.685e−06  | 34.736  | < 2e−16 *** |
| speed_25                 | 6.587e−01| 1.238e−02  | 53.206  | < 2e−16 *** |
| speed_30                 | −3.703e−02| 1.185e−02  | −3.125  | 0.001781 ** |
| speed_35                 | −1.177e−01| 1.158e−02  | −10.162 | < 2e−16 *** |
speed_40  -8.413e-03  1.157e-02  -0.727  0.467026
speed_45  -3.285e-03  1.286e-02  -0.255  0.798432
speed_50  8.432e-03  1.326e-02   0.636  0.524808
speed_55  1.056e-01  1.152e-02   9.164 < 2e−16 ***
speed_60  9.919e-02  1.162e-02   8.538 < 2e−16 ***
speed_65  3.506e-02  1.242e-02   2.823  0.004758 **
age_25_to_34 7.488e-02  1.509e-02   4.963  6.94e−07 ***
age_35_to_44 -1.055e-01  2.683e-02  -3.933  8.38e−05 ***
age_45_to_54 -2.184e-01  2.572e-02  -8.490 < 2e−16 ***
age_55_to_64 -3.575e-02  2.363e-02  -1.513  0.130242
age_65_to_74 -1.572e-01  2.140e-02   7.345  2.06e−13 ***
age_75_to_84 -5.696e-02  1.451e-02  -3.927  8.60e−05 ***
age_85_and_over -3.847e-02  2.611e-02  -1.474  0.140547
gender -1.026e-02  1.770e-02  -0.580  0.562177
education_1  1.520e-01  1.685e-02   9.020 < 2e−16 ***
education_2  1.347e-01  2.882e-02   4.676  2.93e−06 ***
morning -3.225e-02  6.823e-03  -4.726  2.29e−06 ***
afternoon -3.566e-02  6.786e-03  -5.254  1.49e−07 ***
night -2.295e-02  6.986e-03  -3.285  0.001019 **
person1  8.224e-02  1.385e-02   5.936  2.93e−09 ***
person2 -3.585e-01  4.104e-02  -8.735 < 2e−16 ***
person3 -2.376e-01  3.023e-02  -7.859  3.90e−15 ***
person4 -1.397e-01  2.816e-02  -4.961  7.03e−07 ***
person5 -8.730e-02  2.141e-02  -4.077  4.57e−05 ***
person6  8.727e-02  2.382e-02   3.663  0.000249 ***
person7 -5.351e-02  2.767e-02  -1.934  0.053160
person8  1.706e-02  1.635e-02   1.043  0.296725
person9 -2.118e-01  2.885e-02  -7.342  2.12e−13 ***
person10 -1.365e-01  2.645e-02  -5.162  2.45e−07 ***
person11 -3.694e-01  5.026e-02  -7.349  2.01e−13 ***
person12 -1.909e-01  1.673e-02  -11.414 < 2e−16 ***
person13 -1.762e-01  1.570e-02  -11.223 < 2e−16 ***
<table>
<thead>
<tr>
<th>Person</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>person14</td>
<td>3.327e-02</td>
<td>2.254e-02</td>
<td>1.476</td>
<td>0.139913</td>
<td></td>
</tr>
<tr>
<td>person15</td>
<td>4.261e-02</td>
<td>1.668e-02</td>
<td>2.554</td>
<td>0.010644 *</td>
<td></td>
</tr>
<tr>
<td>person16</td>
<td>-1.367e-01</td>
<td>4.077e-02</td>
<td>-3.354</td>
<td>0.000797 ***</td>
<td></td>
</tr>
<tr>
<td>person17</td>
<td>5.759e-02</td>
<td>2.231e-02</td>
<td>2.581</td>
<td>0.009850 **</td>
<td></td>
</tr>
<tr>
<td>person18</td>
<td>-1.470e-01</td>
<td>3.445e-02</td>
<td>-4.267</td>
<td>1.98e-05 ***</td>
<td></td>
</tr>
<tr>
<td>person19</td>
<td>-2.957e-01</td>
<td>2.274e-02</td>
<td>-13.006</td>
<td>&lt; 2e-16 ***</td>
<td></td>
</tr>
<tr>
<td>person20</td>
<td>-6.261e-02</td>
<td>2.196e-02</td>
<td>-2.851</td>
<td>0.004361 **</td>
<td></td>
</tr>
<tr>
<td>person21</td>
<td>-6.608e-02</td>
<td>2.861e-02</td>
<td>-2.310</td>
<td>0.020911 *</td>
<td></td>
</tr>
<tr>
<td>person22</td>
<td>-1.003e-01</td>
<td>1.636e-02</td>
<td>-6.134</td>
<td>8.59e-10 ***</td>
<td></td>
</tr>
<tr>
<td>person23</td>
<td>-1.187e-01</td>
<td>1.716e-02</td>
<td>-6.919</td>
<td>4.56e-12 ***</td>
<td></td>
</tr>
<tr>
<td>person24</td>
<td>-4.141e-02</td>
<td>2.727e-02</td>
<td>-1.518</td>
<td>0.128919</td>
<td></td>
</tr>
<tr>
<td>person25</td>
<td>-1.422e-01</td>
<td>2.766e-02</td>
<td>-5.142</td>
<td>2.73e-07 ***</td>
<td></td>
</tr>
<tr>
<td>person26</td>
<td>-1.782e-01</td>
<td>4.047e-02</td>
<td>-4.405</td>
<td>1.06e-05 ***</td>
<td></td>
</tr>
<tr>
<td>person27</td>
<td>9.587e-02</td>
<td>2.346e-02</td>
<td>4.086</td>
<td>4.39e-05 ***</td>
<td></td>
</tr>
<tr>
<td>person28</td>
<td>-5.970e-02</td>
<td>1.395e-02</td>
<td>-4.280</td>
<td>1.87e-05 ***</td>
<td></td>
</tr>
<tr>
<td>person29</td>
<td>-3.586e-02</td>
<td>1.350e-02</td>
<td>-2.656</td>
<td>0.007915 **</td>
<td></td>
</tr>
<tr>
<td>person30</td>
<td>6.698e-02</td>
<td>2.057e-02</td>
<td>3.256</td>
<td>0.001129 **</td>
<td></td>
</tr>
<tr>
<td>person31</td>
<td>1.175e-01</td>
<td>2.247e-02</td>
<td>5.232</td>
<td>1.68e-07 ***</td>
<td></td>
</tr>
<tr>
<td>person32</td>
<td>-1.328e-01</td>
<td>4.219e-02</td>
<td>-3.147</td>
<td>0.001649 **</td>
<td></td>
</tr>
<tr>
<td>person33</td>
<td>-1.934e-01</td>
<td>2.543e-02</td>
<td>-7.604</td>
<td>2.88e-14 ***</td>
<td></td>
</tr>
<tr>
<td>person34</td>
<td>-1.760e-01</td>
<td>1.603e-02</td>
<td>-10.983</td>
<td>&lt; 2e-16 ***</td>
<td></td>
</tr>
<tr>
<td>person35</td>
<td>-4.660e-02</td>
<td>1.302e-02</td>
<td>-3.580</td>
<td>0.000343 ***</td>
<td></td>
</tr>
<tr>
<td>person36</td>
<td>-1.517e-02</td>
<td>1.858e-02</td>
<td>-0.817</td>
<td>0.414151</td>
<td></td>
</tr>
<tr>
<td>person37</td>
<td>6.025e-02</td>
<td>2.699e-02</td>
<td>2.258</td>
<td>0.023970 *</td>
<td></td>
</tr>
<tr>
<td>person38</td>
<td>-1.463e-01</td>
<td>3.257e-02</td>
<td>-4.491</td>
<td>7.11e-06 ***</td>
<td></td>
</tr>
<tr>
<td>person39</td>
<td>-1.074e-01</td>
<td>2.747e-02</td>
<td>-3.911</td>
<td>9.21e-05 ***</td>
<td></td>
</tr>
<tr>
<td>person40</td>
<td>-2.098e-01</td>
<td>1.309e-02</td>
<td>-16.033</td>
<td>&lt; 2e-16 ***</td>
<td></td>
</tr>
<tr>
<td>person41</td>
<td>4.365e-02</td>
<td>1.208e-02</td>
<td>3.612</td>
<td>0.000304 ***</td>
<td></td>
</tr>
<tr>
<td>person42</td>
<td>5.804e-02</td>
<td>2.217e-02</td>
<td>2.618</td>
<td>0.008833 **</td>
<td></td>
</tr>
<tr>
<td>person43</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>person44</td>
<td>-8.501e-02</td>
<td>1.572e-02</td>
<td>-5.409</td>
<td>6.33e-08 ***</td>
<td></td>
</tr>
<tr>
<td>person45</td>
<td>-6.978e-02</td>
<td>2.156e-02</td>
<td>-3.237</td>
<td>0.001209 **</td>
<td></td>
</tr>
<tr>
<td>Person</td>
<td>$t_i$</td>
<td>$r_i$</td>
<td>$\bar{t_i}$</td>
<td>$\bar{r_i}$</td>
<td>$t_{\text{null}}$</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>person46</td>
<td>$2.890 \times 10^{-1}$</td>
<td>$3.176 \times 10^{-2}$</td>
<td>$9.099$</td>
<td>$&lt; 2 \times 10^{-16}$</td>
<td>***</td>
</tr>
<tr>
<td>person47</td>
<td>$6.476 \times 10^{-2}$</td>
<td>$2.697 \times 10^{-2}$</td>
<td>$2.401$</td>
<td>$0.016347$</td>
<td>*</td>
</tr>
<tr>
<td>person48</td>
<td>$1.977 \times 10^{-1}$</td>
<td>$2.470 \times 10^{-2}$</td>
<td>$8.003$</td>
<td>$1.22 \times 10^{-15}$</td>
<td>***</td>
</tr>
<tr>
<td>person49</td>
<td>$-1.747 \times 10^{-2}$</td>
<td>$2.336 \times 10^{-2}$</td>
<td>$-0.748$</td>
<td>$0.454528$</td>
<td>*</td>
</tr>
<tr>
<td>person50</td>
<td>$-4.392 \times 10^{-2}$</td>
<td>$1.430 \times 10^{-2}$</td>
<td>$-3.071$</td>
<td>$0.002133$</td>
<td>**</td>
</tr>
<tr>
<td>person51</td>
<td>$3.185 \times 10^{-2}$</td>
<td>$1.533 \times 10^{-2}$</td>
<td>$2.078$</td>
<td>$0.037705$</td>
<td>*</td>
</tr>
<tr>
<td>person52</td>
<td>$-1.023 \times 10^{-1}$</td>
<td>$2.728 \times 10^{-2}$</td>
<td>$-3.750$</td>
<td>$0.000177$</td>
<td>***</td>
</tr>
<tr>
<td>person53</td>
<td>$7.841 \times 10^{-3}$</td>
<td>$4.337 \times 10^{-2}$</td>
<td>$0.181$</td>
<td>$0.856517$</td>
<td></td>
</tr>
<tr>
<td>person54</td>
<td>$1.065 \times 10^{-2}$</td>
<td>$1.244 \times 10^{-2}$</td>
<td>$8.561$</td>
<td>$&lt; 2 \times 10^{-16}$</td>
<td>***</td>
</tr>
<tr>
<td>person55</td>
<td>$8.024 \times 10^{-2}$</td>
<td>$2.169 \times 10^{-2}$</td>
<td>$3.699$</td>
<td>$0.000217$</td>
<td>***</td>
</tr>
<tr>
<td>person56</td>
<td>$-1.941 \times 10^{-1}$</td>
<td>$2.274 \times 10^{-2}$</td>
<td>$-8.537$</td>
<td>$&lt; 2 \times 10^{-16}$</td>
<td>***</td>
</tr>
<tr>
<td>person57</td>
<td>$-1.035 \times 10^{-1}$</td>
<td>$1.441 \times 10^{-2}$</td>
<td>$-7.183$</td>
<td>$6.86 \times 10^{-13}$</td>
<td>***</td>
</tr>
<tr>
<td>person58</td>
<td>$-1.395 \times 10^{-2}$</td>
<td>$1.300 \times 10^{-2}$</td>
<td>$-10.733$</td>
<td>$&lt; 2 \times 10^{-16}$</td>
<td>***</td>
</tr>
<tr>
<td>person59</td>
<td>$-3.652 \times 10^{-2}$</td>
<td>$1.562 \times 10^{-2}$</td>
<td>$-2.339$</td>
<td>$0.019353$</td>
<td>*</td>
</tr>
<tr>
<td>person60</td>
<td>$5.077 \times 10^{-2}$</td>
<td>$2.000 \times 10^{-2}$</td>
<td>$2.539$</td>
<td>$0.011115$</td>
<td>*</td>
</tr>
<tr>
<td>person61</td>
<td>$-2.185 \times 10^{-1}$</td>
<td>$2.817 \times 10^{-2}$</td>
<td>$-7.756$</td>
<td>$8.84 \times 10^{-15}$</td>
<td>***</td>
</tr>
<tr>
<td>person62</td>
<td>$4.141 \times 10^{-2}$</td>
<td>$1.510 \times 10^{-2}$</td>
<td>$2.742$</td>
<td>$0.006112$</td>
<td>**</td>
</tr>
<tr>
<td>person63</td>
<td>$-6.742 \times 10^{-2}$</td>
<td>$1.491 \times 10^{-2}$</td>
<td>$-4.522$</td>
<td>$6.14 \times 10^{-6}$</td>
<td>***</td>
</tr>
<tr>
<td>person64</td>
<td>$8.739 \times 10^{-2}$</td>
<td>$3.194 \times 10^{-2}$</td>
<td>$2.736$</td>
<td>$0.006213$</td>
<td>**</td>
</tr>
<tr>
<td>person65</td>
<td>$9.984 \times 10^{-2}$</td>
<td>$1.691 \times 10^{-2}$</td>
<td>$5.906$</td>
<td>$3.52 \times 10^{-9}$</td>
<td>***</td>
</tr>
<tr>
<td>person66</td>
<td>$-6.332 \times 10^{-2}$</td>
<td>$2.641 \times 10^{-2}$</td>
<td>$-2.398$</td>
<td>$0.016508$</td>
<td>*</td>
</tr>
<tr>
<td>person67</td>
<td>$-3.954 \times 10^{-2}$</td>
<td>$2.115 \times 10^{-2}$</td>
<td>$-1.869$</td>
<td>$0.061576$</td>
<td>.</td>
</tr>
<tr>
<td>person68</td>
<td>$-1.039 \times 10^{-1}$</td>
<td>$2.121 \times 10^{-2}$</td>
<td>$-0.490$</td>
<td>$0.624268$</td>
<td></td>
</tr>
<tr>
<td>person69</td>
<td>$-2.376 \times 10^{-1}$</td>
<td>$2.676 \times 10^{-2}$</td>
<td>$-8.878$</td>
<td>$&lt; 2 \times 10^{-16}$</td>
<td>***</td>
</tr>
<tr>
<td>person70</td>
<td>$1.313 \times 10^{-1}$</td>
<td>$2.141 \times 10^{-2}$</td>
<td>$6.131$</td>
<td>$8.76 \times 10^{-10}$</td>
<td>***</td>
</tr>
<tr>
<td>person71</td>
<td>$4.016 \times 10^{-2}$</td>
<td>$1.182 \times 10^{-2}$</td>
<td>$3.398$</td>
<td>$0.000678$</td>
<td>***</td>
</tr>
<tr>
<td>person72</td>
<td>$-3.664 \times 10^{-1}$</td>
<td>$3.123 \times 10^{-2}$</td>
<td>$-11.732$</td>
<td>$&lt; 2 \times 10^{-16}$</td>
<td>***</td>
</tr>
<tr>
<td>person73</td>
<td>$-4.029 \times 10^{-1}$</td>
<td>$2.367 \times 10^{-2}$</td>
<td>$-17.022$</td>
<td>$&lt; 2 \times 10^{-16}$</td>
<td>***</td>
</tr>
<tr>
<td>person74</td>
<td>$-9.615 \times 10^{-2}$</td>
<td>$3.697 \times 10^{-2}$</td>
<td>$-2.601$</td>
<td>$0.009303$</td>
<td>**</td>
</tr>
<tr>
<td>person75</td>
<td>$-2.107 \times 10^{-1}$</td>
<td>$3.367 \times 10^{-2}$</td>
<td>$-6.258$</td>
<td>$3.92 \times 10^{-10}$</td>
<td>***</td>
</tr>
<tr>
<td>person76</td>
<td>$-3.576 \times 10^{-2}$</td>
<td>$1.497 \times 10^{-2}$</td>
<td>$-2.390$</td>
<td>$0.016869$</td>
<td>*</td>
</tr>
<tr>
<td>person77</td>
<td>$-1.457 \times 10^{-1}$</td>
<td>$2.462 \times 10^{-2}$</td>
<td>$-5.917$</td>
<td>$3.29 \times 10^{-9}$</td>
<td>***</td>
</tr>
<tr>
<td>Person</td>
<td>Value1</td>
<td>Value2</td>
<td>Value3</td>
<td>Value4</td>
<td>Value5</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>person78</td>
<td>4.996e-02</td>
<td>1.193e-02</td>
<td>4.188</td>
<td>2.82e-05</td>
<td>***</td>
</tr>
<tr>
<td>person79</td>
<td>7.623e-03</td>
<td>2.296e-02</td>
<td>0.332</td>
<td>0.739919</td>
<td></td>
</tr>
<tr>
<td>person80</td>
<td>-1.377e-01</td>
<td>3.726e-02</td>
<td>-3.697</td>
<td>0.000218</td>
<td>***</td>
</tr>
<tr>
<td>person81</td>
<td>-1.418e-01</td>
<td>2.766e-02</td>
<td>-5.127</td>
<td>2.94e-07</td>
<td>***</td>
</tr>
<tr>
<td>person82</td>
<td>-7.100e-02</td>
<td>2.211e-02</td>
<td>-3.211</td>
<td>0.001323</td>
<td>**</td>
</tr>
<tr>
<td>person83</td>
<td>-6.847e-02</td>
<td>1.427e-02</td>
<td>-4.799</td>
<td>1.60e-06</td>
<td>***</td>
</tr>
<tr>
<td>person84</td>
<td>-3.327e-02</td>
<td>4.062e-02</td>
<td>-0.819</td>
<td>0.412738</td>
<td></td>
</tr>
<tr>
<td>person85</td>
<td>-1.575e-02</td>
<td>2.743e-02</td>
<td>-0.574</td>
<td>0.565958</td>
<td></td>
</tr>
<tr>
<td>person86</td>
<td>-2.053e-02</td>
<td>2.479e-02</td>
<td>-0.828</td>
<td>0.407570</td>
<td></td>
</tr>
<tr>
<td>person87</td>
<td>1.361e-01</td>
<td>2.304e-02</td>
<td>5.907</td>
<td>3.50e-09</td>
<td>***</td>
</tr>
<tr>
<td>person88</td>
<td>4.039e-02</td>
<td>2.234e-02</td>
<td>1.807</td>
<td>0.070698</td>
<td>.</td>
</tr>
<tr>
<td>person89</td>
<td>4.039e-02</td>
<td>1.148e-02</td>
<td>1.807</td>
<td>0.070698</td>
<td>.</td>
</tr>
<tr>
<td>person90</td>
<td>4.039e-02</td>
<td>1.148e-02</td>
<td>1.807</td>
<td>0.070698</td>
<td>.</td>
</tr>
<tr>
<td>person91</td>
<td>4.039e-02</td>
<td>1.148e-02</td>
<td>1.807</td>
<td>0.070698</td>
<td>.</td>
</tr>
<tr>
<td>person92</td>
<td>4.039e-02</td>
<td>1.148e-02</td>
<td>1.807</td>
<td>0.070698</td>
<td>.</td>
</tr>
<tr>
<td>person93</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>person94</td>
<td>-9.194e-02</td>
<td>2.607e-02</td>
<td>-3.527</td>
<td>0.000420</td>
<td>***</td>
</tr>
<tr>
<td>person95</td>
<td>-9.194e-02</td>
<td>2.607e-02</td>
<td>-3.527</td>
<td>0.000420</td>
<td>***</td>
</tr>
<tr>
<td>person96</td>
<td>-9.194e-02</td>
<td>2.607e-02</td>
<td>-3.527</td>
<td>0.000420</td>
<td>***</td>
</tr>
<tr>
<td>person97</td>
<td>-9.194e-02</td>
<td>2.607e-02</td>
<td>-3.527</td>
<td>0.000420</td>
<td>***</td>
</tr>
<tr>
<td>person98</td>
<td>-9.194e-02</td>
<td>2.607e-02</td>
<td>-3.527</td>
<td>0.000420</td>
<td>***</td>
</tr>
<tr>
<td>person99</td>
<td>-9.194e-02</td>
<td>2.607e-02</td>
<td>-3.527</td>
<td>0.000420</td>
<td>***</td>
</tr>
<tr>
<td>person100</td>
<td>3.813e-03</td>
<td>2.387e-02</td>
<td>0.160</td>
<td>0.873055</td>
<td>.</td>
</tr>
<tr>
<td>person101</td>
<td>-7.372e-02</td>
<td>4.181e-02</td>
<td>-1.763</td>
<td>0.077881</td>
<td>.</td>
</tr>
<tr>
<td>person102</td>
<td>-5.340e-02</td>
<td>2.790e-02</td>
<td>-1.914</td>
<td>0.055633</td>
<td>.</td>
</tr>
<tr>
<td>person103</td>
<td>7.016e-02</td>
<td>1.522e-02</td>
<td>4.611</td>
<td>4.02e-06</td>
<td>***</td>
</tr>
<tr>
<td>person104</td>
<td>4.638e-02</td>
<td>2.352e-02</td>
<td>1.972</td>
<td>0.048571</td>
<td>*</td>
</tr>
<tr>
<td>person105</td>
<td>9.160e-02</td>
<td>2.641e-02</td>
<td>3.468</td>
<td>0.000524</td>
<td>***</td>
</tr>
<tr>
<td>person106</td>
<td>-3.594e-02</td>
<td>3.483e-02</td>
<td>-1.032</td>
<td>0.302070</td>
<td>.</td>
</tr>
<tr>
<td>person107</td>
<td>-4.507e-01</td>
<td>3.144e-02</td>
<td>-14.335</td>
<td>&lt; 2e-16</td>
<td>***</td>
</tr>
<tr>
<td>person108</td>
<td>1.246e-01</td>
<td>2.107e-02</td>
<td>5.915</td>
<td>3.33e-09</td>
<td>***</td>
</tr>
<tr>
<td>person109</td>
<td>-8.831e-02</td>
<td>3.765e-02</td>
<td>-2.346</td>
<td>0.018985</td>
<td>*</td>
</tr>
<tr>
<td>Person</td>
<td>Value1</td>
<td>Value2</td>
<td>Value3</td>
<td>Value4</td>
<td>Value5</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>person110</td>
<td>9.840e-02</td>
<td>2.772e-02</td>
<td>3.550</td>
<td>0.000386</td>
<td>***</td>
</tr>
<tr>
<td>person111</td>
<td>-2.075e-01</td>
<td>2.470e-02</td>
<td>-8.400</td>
<td>&lt; 2e-16</td>
<td>***</td>
</tr>
<tr>
<td>person112</td>
<td>-1.366e-01</td>
<td>3.017e-02</td>
<td>-4.529</td>
<td>5.95e-06</td>
<td>***</td>
</tr>
<tr>
<td>person113</td>
<td>-6.209e-02</td>
<td>4.119e-02</td>
<td>-1.507</td>
<td>0.131701</td>
<td></td>
</tr>
<tr>
<td>person114</td>
<td>-1.552e-02</td>
<td>3.097e-02</td>
<td>-0.501</td>
<td>0.616191</td>
<td></td>
</tr>
<tr>
<td>person115</td>
<td>-1.417e-02</td>
<td>2.805e-02</td>
<td>-5.053</td>
<td>4.35e-07</td>
<td>***</td>
</tr>
<tr>
<td>person116</td>
<td>7.728e-02</td>
<td>2.270e-02</td>
<td>3.404</td>
<td>0.000663</td>
<td>***</td>
</tr>
<tr>
<td>person117</td>
<td>5.878e-02</td>
<td>2.729e-02</td>
<td>2.154</td>
<td>0.031259</td>
<td>*</td>
</tr>
<tr>
<td>person118</td>
<td>6.930e-02</td>
<td>3.155e-02</td>
<td>4.058</td>
<td>4.95e-05</td>
<td>***</td>
</tr>
<tr>
<td>person119</td>
<td>-1.189e-01</td>
<td>2.671e-02</td>
<td>-4.451</td>
<td>8.54e-06</td>
<td>***</td>
</tr>
<tr>
<td>person120</td>
<td>2.940e-02</td>
<td>2.257e-02</td>
<td>1.303</td>
<td>0.192656</td>
<td></td>
</tr>
<tr>
<td>person121</td>
<td>-3.562e-01</td>
<td>1.649e-02</td>
<td>-7.560</td>
<td>4.07e-14</td>
<td>***</td>
</tr>
<tr>
<td>person122</td>
<td>-3.562e-01</td>
<td>2.247e-02</td>
<td>-15.848</td>
<td>&lt; 2e-16</td>
<td>***</td>
</tr>
<tr>
<td>person123</td>
<td>1.281e-01</td>
<td>3.155e-02</td>
<td>4.058</td>
<td>4.95e-05</td>
<td>***</td>
</tr>
<tr>
<td>person124</td>
<td>9.131e-02</td>
<td>1.822e-02</td>
<td>5.012</td>
<td>5.40e-07</td>
<td>***</td>
</tr>
<tr>
<td>person125</td>
<td>-9.923e-03</td>
<td>2.820e-02</td>
<td>-0.352</td>
<td>0.724897</td>
<td></td>
</tr>
<tr>
<td>person126</td>
<td>3.512e-02</td>
<td>2.456e-02</td>
<td>1.430</td>
<td>0.152618</td>
<td></td>
</tr>
<tr>
<td>person127</td>
<td>2.766e-01</td>
<td>3.146e-02</td>
<td>8.792</td>
<td>&lt; 2e-16</td>
<td>***</td>
</tr>
<tr>
<td>person128</td>
<td>-7.630e-01</td>
<td>2.680e-01</td>
<td>-2.847</td>
<td>0.004419</td>
<td>**</td>
</tr>
<tr>
<td>person129</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>person130</td>
<td>2.102e-02</td>
<td>3.722e-02</td>
<td>0.565</td>
<td>0.572222</td>
<td></td>
</tr>
<tr>
<td>person131</td>
<td>7.813e-02</td>
<td>2.336e-02</td>
<td>3.344</td>
<td>0.000825</td>
<td>***</td>
</tr>
<tr>
<td>person132</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>person133</td>
<td>-1.086e-01</td>
<td>1.352e-02</td>
<td>-8.032</td>
<td>9.71e-16</td>
<td>***</td>
</tr>
<tr>
<td>person134</td>
<td>3.639e-03</td>
<td>1.447e-02</td>
<td>0.251</td>
<td>0.801465</td>
<td></td>
</tr>
<tr>
<td>person135</td>
<td>-9.218e-02</td>
<td>4.189e-02</td>
<td>-2.200</td>
<td>0.027783</td>
<td>*</td>
</tr>
<tr>
<td>person136</td>
<td>-1.304e-01</td>
<td>2.717e-02</td>
<td>-4.799</td>
<td>1.60e-06</td>
<td>***</td>
</tr>
<tr>
<td>person137</td>
<td>-2.016e-01</td>
<td>1.516e-02</td>
<td>-13.297</td>
<td>&lt; 2e-16</td>
<td>***</td>
</tr>
<tr>
<td>person138</td>
<td>-8.496e-02</td>
<td>3.014e-02</td>
<td>-2.819</td>
<td>0.004818</td>
<td>**</td>
</tr>
<tr>
<td>person139</td>
<td>-1.771e-01</td>
<td>2.668e-02</td>
<td>-6.638</td>
<td>3.19e-11</td>
<td>***</td>
</tr>
<tr>
<td>person140</td>
<td>9.608e-02</td>
<td>2.292e-02</td>
<td>4.192</td>
<td>2.77e-05</td>
<td>***</td>
</tr>
<tr>
<td>person141</td>
<td>-2.037e-01</td>
<td>2.711e-02</td>
<td>-7.512</td>
<td>5.85e-14</td>
<td>***</td>
</tr>
</tbody>
</table>
```r
person142  -9.676e-02  1.584e-02  -6.108 1.01e-09 ***
person143  -1.203e-01  4.158e-02  -2.894 0.003801 **
person144  NA   NA   NA   NA
person145  1.800e-01  2.807e-02  6.411 1.45e-10 ***
person146  NA   NA   NA   NA
person147  -4.584e-02  2.363e-02  -1.940 0.052383 .
person148  NA   NA   NA   NA
person149  NA   NA   NA   NA
person150  NA   NA   NA   NA
person151  NA   NA   NA   NA

---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.2671 on 115371 degrees of freedom
Multiple R-squared: 0.2851, Adjusted R-squared: 0.2841
F-statistic: 280.5 on 164 and 115371 DF, p-value: < 2.2e-16

> write.csv(summary(regression)$coef, "regression.csv")
>
> #Boxplot
> #Speedlimit
> boxplot(degree ~ Speedlimit, data=res, ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Speed limit (mph)", ylab="driving speed/speed limit")
> #Time
> boxplot(degree ~ Hour, data=res, ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Time", ylab="driving speed/speed limit")
> #Education
> boxplot(degree ~ o_edu, data=res, ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Education", ylab="driving speed/speed limit")
> #Age
```

304
> boxplot(degree ~ o_age, data=res, ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Age", ylab="driving speed/speed limit")
> #Gender
> boxplot(degree ~ o_gender, data=res, ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Gender", ylab="driving speed/speed limit")
>
> #Plot between street length and speeding
> street_length <- res$streetlength
> speeding <- res$degree
> plot(street_length, speeding, pch=".", ylim=c(0, 8), yaxp=c(0, 8, 8), xlab="Link length", ylab="driving speed/speed limit")
> lm.obj<-lm(speeding~street_length)
> abline(lm.obj,col=2)
> summary(lm.obj)

Call:
lm(formula = speeding ~ street_length)

Residuals:
     Min      1Q  Median       3Q      Max
-1.3322 -0.1889  0.0193  0.1696  6.8196

Coefficients:  Estimate  Std. Error t value Pr(>|t|)
(Intercept)  7.246e-01   1.249e-03  580.0  <2e-16 ***
street_length  3.043e-04   3.412e-06   89.2  <2e-16 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3053 on 115534 degrees of freedom
Multiple R-squared: 0.06443, Adjusted R-squared: 0.06442

F-statistic: 7957 on 1 and 115534 DF, p-value: < 2.2e-16

> mtext("N=115,536, R^2=0.0644, y=0.7246+0.0003x", side=3)
> #Link length is less than 1,000m
> length1000 <- subset(res, res$streetlength <= 1000)
> street_length_1 <- length1000$streetlength
> speeding_1 <- length1000$degree
> lm1000.obj<-lm(speeding_1~street_length_1)
> summary(lm1000.obj)

Call:
  lm(formula = speeding_1 ~ street_length_1)

Residuals:
       Min        1Q   Median       3Q      Max
-1.0305 -0.1855   0.0137   0.1677   6.8350

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.883e-01  1.425e-03 482.86  <2e-16 ***
street_length_1 4.787e-04  4.812e-06   99.47  <2e-16 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ 1

Residual standard error: 0.3033 on 113124 degrees of freedom
Multiple R-squared: 0.08043, Adjusted R-squared: 0.08042
F-statistic: 9894 on 1 and 113124 DF, p-value: < 2.2e-16
```r
> plot(street_length, speeding, pch=".", ylim=c(0, 8), yaxp=c(0, 8, 8),
      xlab="Link length", ylab="driving speed/speed limit")
> abline(lm1000.obj, col=2)
> mtext("N=115,536, R^2=0.0804, y=0.6883+0.0005x", side=3)
> 
```