MATH2349 Semester 1, 2020

Code

Assignment 2

Subbiah Soundarapandian - s3825012

Required packages

Provide the packages required to reproduce the report. Make sure you fulfilled the minimum requirement #10.

Hide

```
# This is the R chunk for the required packages
library(readr)
library(tidyr)
```

```
Registered S3 method overwritten by 'dplyr':
  method           from
  print.rowwise_df
```

Hide

```
library(dplyr)
```

```

Attaching package: 勉牶dplyr勉牷

The following objects are masked from 勉牶package:stats勉牷:

    filter, lag

The following objects are masked from 勉牶package:base勉牷:

    intersect, setdiff, setequal, union
```

Hide

```
library(Hmisc)
```

```
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2
Registered S3 method overwritten by 'htmlwidgets':
  method           from
  print.htmlwidget tools:rstudio
Registered S3 method overwritten by 'data.table':
  method           from
  print.data.table

Attaching package: 勄犈Hmisc勄犴

The following objects are masked from 勄犈package:dplyr勄犴:

    src, summarize

The following objects are masked from 勄犈package:base勄犴:

    format.pval, units
```

Hide

```
library(lubridate)
```

```

Attaching package: 勄犈lubridate勄犴

The following objects are masked from 勄犈package:dplyr勄犴:

    intersect, setdiff, union

The following objects are masked from 勄犈package:base勄犴:

    date, intersect, setdiff, union
```

Hide

```
library(outliers)
library(zoo)
```

```

Attaching package: 勄犈zoo勄犴

The following objects are masked from 勄犈package:base勄犴:

    as.Date, as.Date.numeric
```

Hide

```
library(editrules)
```

```
Loading required package: igraph

Attaching package: 勉牠igraph勉牷

The following objects are masked from 勉牠package:lubridate勉牷:

    %--%, union

The following objects are masked from 勉牠package:dplyr勉牷:

    as_data_frame, groups, union

The following object is masked from 勉牠package:tidyr勉牷:

    crossing

The following objects are masked from 勉牠package:stats勉牷:

    decompose, spectrum

The following object is masked from 勉牠package:base勉牷:

    union


Attaching package: 勉牠editrules勉牷

The following objects are masked from 勉牠package:igraph勉牷:

    blocks, normalize

The following object is masked from 勉牠package:dplyr勉牷:

    contains

The following objects are masked from 勉牠package:tidyr勉牷:

    contains, separate
```

Hide

```
library(deducorrect)
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

Executive Summary

- The dataset is imported to R by the read_csv function of the readr package and appropriate type conversions were made(character to factor with labels, character to date).
- The dataset is filtered with the variable we want for our analysis and then converted into tidy dataset using the tidyr package and mutated with an additional variable using the mutate function in dplyr package.

- Structure of the data and attributes have been checked using the appropriate functions.
- Then, the numeric variables have been checked for missing values, special values and obvious inconsistencies. Appropriate methods have been implemented to deal with the same. Edit rules package and deducorrect package played a important role in this step.
- All the numeric variables have been checked for outliers using methods like tukey's method and z-score method. Then those outliers are capped with 5th and 95 th percentile values using capping/winsoring method.
- Then the mutated variable is transformed to a normal variable since it was right skewed. This was done using BoxCox function in the forecast library.

Data

- The datasets for this assignment have been taken from the DATA VIC website.
- Dataset 1 -> https://discover.data.vic.gov.au/dataset/bluetooth-links (https://discover.data.vic.gov.au/dataset/bluetooth-links) Dataset 2 -> https://discover.data.vic.gov.au/dataset/bluetooth-travel-time-updates-every-2-minutes (https://discover.data.vic.gov.au/dataset/bluetooth-travel-time-updates-every-2-minutes).
- The datasets provide information on the travel time taken for the bluetooth links recorded every 2 minutes on a particular day.
- The two datasets are imported using read_csv function in the readr package.
- The two datasets were merged using left_join() function in dplyr package, using the variables 'link_id'(dataset1) and 'LINK_ID'(dataset2) and keeping the necessary variables for our analysis.
- It is filtered with those whose description is not 'No Name Set' since it hasn't got any information for our analysis.
- The Merged dataset 'dataset_final' has 16 variables.

Variables:

- link_id - This variable is an unique id. It represents all the Bluetooth links in the state of Victoria. Even though it's a numeric variable, it's a qualitative variable.
- description - This variable provides the link name of the road connecting two areas and the section name along with it separated by comma.
- origin - This is an unique id. Each Origin ID represents the location, from where the bluetooth link starts.
- destination - This is an unique id as well. Each destination ID represents the location where the bluetooth link ends.
- direction - This is a categorical variable and represents the direction of each bluetooth link.
- link_length - It's a direct length measurement of the bluetooth link between the origin and destination.
- SHAPE__Length - It's a length between Origin and Destination, that varies from the link_length due to the geometry of the links.
- DELAY - Delay parameter represents the difference between the fixed baseline time of travel and the actual time of travel of the bluetooth signal between origin and the destination.
- EXCESS_DELAY - Excess delay variable represents the difference between the expected travel time and actual travel time. Excess delays greater than zero indicates the link is operating worse than expected for this time of day.Negative values represent that the link is travelling better than expected.
- TRAVEL_TIME - This variable represents the time taken by the bluetooth link to travel between the origin site and destination site.
- SPEED - This variable is the speed with which the link travels between origin and destination,measured in metres per second. *TIME_STAMP - This variable is in the form of ymd_hms. This records the time when the bluetooth links parameters are noted.
- ETL_TIMESTAMP - This variable is a temporary variable of timestamps, created in the ETL(Extract,Transform,Load) tool, for processing the incremental data.

Hide

```
# This is the R chunk for the Data Section
setwd("F:/Subbu/RMIT/sem 1/data wrangling/assign 2/datasets")
```

```
The working directory was changed to F:/Subbu/RMIT/sem 1/data wrangling/assign 2/datasets ins
ide a notebook chunk. The working directory will be reset when the chunk is finished running.
Use the knitr root.dir option in the setup chunk to change the working directory for notebook
chunks.
```

Hide

```
dataset1 <- read_csv("Bluetooth_Travel_Time__updates_every_2_minutes_.csv")
```

```
Parsed with column specification:
cols(
  .default = col_double(),
  LINK_NAME = □[31mcol_character()□[39m,
  ROAD_NAME = □[31mcol_character()□[39m,
  SECTION_DESCRIPTION = □[31mcol_character()□[39m,
  DIRECTION = □[31mcol_character()□[39m,
  IS_FREEWAY = □[31mcol_character()□[39m,
  ENABLED = □[31mcol_character()□[39m,
  CLOSED = □[31mcol_character()□[39m,
  ENOUGH_DATA = □[31mcol_character()□[39m,
  IGNORED = □[31mcol_character()□[39m,
  TIMESTAMP = □[31mcol_character()□[39m,
  ETL_TIMESTAMP = □[31mcol_character()□[39m
)
See spec(...) for full column specifications.
```

Hide

```
head(dataset1)
```

OBJE… <dbl>	LINK… <dbl>	LINK_NAME <chr>	ROAD_NAME <chr>
1	3	Bulleen Rd, Eastern Fwy to Manningham Rd	Bulleen Rd
2	5	Greensborough Hwy, M80 to Grimshaw St	Greensborough Hv
3	6	Greensborough Hwy, Grimshaw St to M80	Greensborough Hv
4	7	Greensborough Hwy, Grimshaw St to Watsonia Rd	Greensborough Hv
5	8	Greensborough Hwy, Watsonia Rd to Grimshaw St	Greensborough Hv
6	9	Greensborough Hwy, Watsonia Rd to Lwr Plenty Rd	Greensborough Hv

6 rows | 1-4 of 25 columns

Hide

```
dataset2 <- read_csv("Bluetooth_Links.csv")
```

```
Parsed with column specification:
cols(
  link_id = □[32mcol_double()□[39m,
  description = □[31mcol_character()□[39m,
  origin = □[32mcol_double()□[39m,
  destination = □[32mcol_double()□[39m,
  direction = □[31mcol_character()□[39m,
  link_length = □[32mcol_double()□[39m,
  OBJECTID = □[32mcol_double()□[39m,
  SHAPE__Length = □[32mcol_double()□[39m
)
```

Hide

```
head(dataset2)
```

link_id <dbl>	description <chr>	origin <dbl>	destination <dbl>	direc <chr>
3	Bulleen Rd, Eastern Fwy to Manningham Rd	2827	686	NB
5	Greensborough Hwy, M80 to Grimshaw St	3357	4187	SB
6	Greensborough Hwy, Grimshaw St to M80	4187	3357	NB
7	Greensborough Hwy, Grimshaw St to Watsonia Rd	4187	3341	SB
8	Greensborough Hwy, Watsonia Rd to Grimshaw St	3341	4187	NB
9	Greensborough Hwy, Watsonia Rd to Lwr Plenty Rd	3341	3333	SB

6 rows | 1-6 of 8 columns

Hide

```
#Merging dataset using left join
dataset_final <- left_join(dataset2,dataset1,c("link_id" = "LINK_ID")) %>% select(-(OBJECTID.
y:DESTINATION_ID),-CONGESTION, -(SCORE:AVERAGE_DENSITY), -(IS_FREEWAY:IGNORED))
dataset_final <- filter(dataset_final, description != "No Name Set")
head(dataset_final)
```

link_id <dbl>	description <chr>	origin <dbl>	destination <dbl>	direc <chr>
3	Bulleen Rd, Eastern Fwy to Manningham Rd	2827	686	NB
5	Greensborough Hwy, M80 to Grimshaw St	3357	4187	SB
6	Greensborough Hwy, Grimshaw St to M80	4187	3357	NB
7	Greensborough Hwy, Grimshaw St to Watsonia Rd	4187	3341	SB
8	Greensborough Hwy, Watsonia Rd to Grimshaw St	3341	4187	NB
9	Greensborough Hwy, Watsonia Rd to Lwr Plenty Rd	3341	3333	SB

6 rows | 1-6 of 14 columns

Understand

- Summary of the types of variables and data structures is found using str() function.
- The summary shows that the dataset_final used for this analysis contains multiple datatypes like numeric, character etc.
- ‘direction’ variable of the dataset is converted into factor using factor() function and the labels of the factors are given accordingly.
- ‘ETL_TIMESTAMP’ is converted into a date format using ymd_hms() function in lubridate library.

Hide

```
# This is the R chunk for the Understand Section
str(dataset_final)
```

```
Classes ‘spec_tbl_df’, ‘tbl_df’, ‘tbl’ and 'data.frame':    1553 obs. of  14 variables:
 $ link_id      : num  3 5 6 7 8 9 10 11 12 13 ...
 $ description  : chr  "Bulleen Rd, Eastern Fwy to Manningham Rd" "Greensborough Hwy, M80 to
Grimshaw St" "Greensborough Hwy, Grimshaw St to M80" "Greensborough Hwy, Grimshaw St to Watso
nia Rd" ...
 $ origin       : num  2827 3357 4187 4187 3341 ...
 $ destination  : num  686 4187 3357 3341 4187 ...
 $ direction    : chr  "NB" "SB" "NB" "SB" ...
 $ link_length  : num  2032 1059 1101 1416 1406 ...
 $ OBJECTID.x   : num  177 178 179 180 181 182 183 184 185 186 ...
 $ SHAPE__Length: num  2578 1342 1395 1794 1781 ...
 $ DELAY        : num  -2 0 10 -8 31 -3 -3 17 18 3 ...
 $ EXCESS_DELAY : num  -10 -12 -4 -16 25 -10 -12 -16 -2 -12 ...
 $ TRAVEL_TIME  : num  136 60 54 51 122 133 135 114 147 60 ...
 $ SPEED        : num  52 66 73 98 41 62 61 56 44 60 ...
 $ TIMESTAMP    : chr  "2020/03/06 23:52:30+11" "2020/03/06 23:52:30+11" "2020/03/06 23:52:30
+11" "2020/03/06 23:52:30+11" ...
 $ ETL_TIMESTAMP: chr  "2020/03/06 23:53:18" "2020/03/06 23:53:18" "2020/03/06 23:53:18" "202
0/03/06 23:53:18" ...
```

Hide

```
attributes(dataset_final)$names
```

```
 [1] "link_id"       "description"   "origin"        "destination"
 [5] "direction"     "link_length"   "OBJECTID.x"    "SHAPE__Length"
 [9] "DELAY"         "EXCESS_DELAY"  "TRAVEL_TIME"   "SPEED"
[13] "TIMESTAMP"     "ETL_TIMESTAMP"
```

Hide

```
dataset_final$direction <- factor(dataset_final$direction, labels = c("Numeric/NA","East","In
wards", "North", "North East", "North East", "North West","North West","Outwards","South", "S
outh East","South East", "South West", "South West","West"))
levels(dataset_final$direction)
```

```
 [1] "Numeric/NA" "East"       "Inwards"    "North"      "North East" "North West"
 [7] "Outwards"   "South"      "South East" "South West" "West"
```

Hide

```
#Character to Date (POSIXct - Calendar time)
str(dataset_final$ETL_TIMESTAMP)
```

```
 chr [1:1553] "2020/03/06 23:53:18" "2020/03/06 23:53:18" ...
```

Hide

```
dataset_final$ETL_TIMESTAMP <- ymd_hms(dataset_final$ETL_TIMESTAMP,tz = "Australia/Melbourne"
)
str(dataset_final$ETL_TIMESTAMP)
```

```
 POSIXct[1:1553], format: "2020-03-06 23:53:18" "2020-03-06 23:53:18" "2020-03-06 23:53:18"
...
```

Hide

```
head(dataset_final)
```

link_id <dbl>	description <chr>	origin <dbl>	destination <dbl>	direc <fctr>
3	Bulleen Rd, Eastern Fwy to Manningham Rd	2827	686	North
5	Greensborough Hwy, M80 to Grimshaw St	3357	4187	South
6	Greensborough Hwy, Grimshaw St to M80	4187	3357	North
7	Greensborough Hwy, Grimshaw St to Watsonia Rd	4187	3341	South
8	Greensborough Hwy, Watsonia Rd to Grimshaw St	3341	4187	North
9	Greensborough Hwy, Watsonia Rd to Lwr Plenty Rd	3341	3333	South

6 rows | 1-6 of 14 columns

Tidy & Manipulate Data I

- The ‘dataset_final’ doesn’t abide by the tidy data principles because the variable ‘description’ has two values in the same column.Also, the ‘TIME_STAMP’ variable has the lapsed time in the same variable.
- So, the ‘description’ variable is split using the separate function in tidyr package. The variable ‘description’ is split into ‘ROAD_NAME’and ’SECTION_NAME’. The ‘extra’ parameter has been passed in case if there are any multiple separators in the same value.
- The ‘TIME_STAMP’ variable is split into ‘TIME_STAMP’ and ‘LAPSE(in secs)’ variables using the separate function.(‘\\’ has been used in the ‘sep’ parameter to detect the special symbol ‘+’)
- As a next step, ‘LAPSE(in secs)’ variable is converted into numeric class.
- The ‘TIME_STAMP’ variable is fitted to Australian time(AEDT) by passing the ‘tz’ parameter as “Australia/Melbourne”, in the ymd_hms function.
- ymd_hms() function comes from the lubridate package.

Hide

```
# This is the R chunk for the Tidy & Manipulate Data I
dataset_final <- dataset_final %>% separate(description, into = c("ROAD_NAME", "SECTION_DESCR
IPTION"),sep = "([,-])", extra = "merge")

dataset_final <- dataset_final %>% separate(TIMESTAMP, into = c("TIME_STAMP", "LAPSE(in sec
s)"),sep = "\\+")

dataset_final$`LAPSE(in secs)` <- as.numeric(dataset_final$`LAPSE(in secs)`)
dataset_final$TIME_STAMP <- ymd_hms(dataset_final$TIME_STAMP,tz = "Australia/Melbourne")
head(dataset_final)
```

link_id <dbl>	ROAD_NAME <chr>	SECTION_DESCRIPTION <chr>	origin <dbl>	destination <dbl>	directio <fctr>
3	Bulleen Rd	Eastern Fwy to Manningham Rd	2827	686	North
5	Greensborough Hwy	M80 to Grimshaw St	3357	4187	South
6	Greensborough Hwy	Grimshaw St to M80	4187	3357	North
7	Greensborough Hwy	Grimshaw St to Watsonia Rd	4187	3341	South
8	Greensborough Hwy	Watsonia Rd to Grimshaw St	3341	4187	North
9	Greensborough Hwy	Watsonia Rd to Lwr Plenty Rd	3341	3333	South

6 rows | 1-7 of 16 columns

Tidy & Manipulate Data II

- In this step, ‘dataset_final’ is mutated with an additional variable ‘difference in length’ which is the difference between link_length of the sector and the shape length of the sector that varies due to the geometry.
- Mutation is done with the help of mutate() function in the dplyr library.

Hide

```
# This is the R chunk for the Tidy & Manipulate Data II
dataset_final <- dataset_final %>% mutate(difference_in_length = SHAPE__Length - link_length)
head(dataset_final$difference_in_length)
```

```
[1] 545.4998 282.9590 294.1097 377.8092 375.1060 615.1394
```

Scan I

- The dataset is checked for missing/NA values across all variables using the colSums() funtion.
- After checking for NA values, all the NA values are imputed with mean values of the respective columns. This is done using impute() function in the Hmisc package.
- However, timestamps can not be done using impute() function. And hence it is done using ‘as.POSIXct’ function nested with na.approx() function in the zoo package, so that the common values can be approximately fitted to the missing timestamps.(Taken from Stack Overflow) [1]
- Finally it’s checked for special values using infinite() function combined with sapply() function.
- Edit rules have been set and the whole dataset is corrected using the same set of rules.

Hide

```
# This is the R chunk for the Scan I
colSums(is.na(dataset_final))
```

```
            link_id            ROAD_NAME  SECTION_DESCRIPTION
                  0                    0                    0
             origin          destination            direction
                  0                    0                    1
        link_length           OBJECTID.x        SHAPE__Length
                  0                    0                    1
              DELAY         EXCESS_DELAY          TRAVEL_TIME
                378                  378                  378
              SPEED           TIME_STAMP       LAPSE(in secs)
                378                  378                  378
      ETL_TIMESTAMP difference_in_length
                378                    1
```

Hide

```
dataset_final$SHAPE__Length <- impute(dataset_final$SHAPE__Length, fun = mean)
dataset_final$TRAVEL_TIME <- impute(dataset_final$TRAVEL_TIME, fun = mean)
dataset_final$DELAY <- impute(dataset_final$DELAY, fun = mean)
dataset_final$EXCESS_DELAY <- impute(dataset_final$EXCESS_DELAY, fun = mean)
dataset_final$SPEED <- impute(dataset_final$SPEED, fun = mean)
dataset_final$`LAPSE(in secs)` <- impute(dataset_final$`LAPSE(in secs)`, fun = mean)
dataset_final$direction <- impute(dataset_final$direction, fun = mean)
dataset_final$difference_in_length <- impute(dataset_final$difference_in_length, fun = mean)
# IMPUTING TIME STAMPS
dataset_final$TIME_STAMP <- as.POSIXct(na.approx(dataset_final$TIME_STAMP), origin = "1970-1-
1")
dataset_final$ETL_TIMESTAMP <- as.POSIXct(na.approx(dataset_final$ETL_TIMESTAMP), origin = "1
970-1-1")
colSums(is.na(dataset_final))
```

```
            link_id            ROAD_NAME  SECTION_DESCRIPTION
                  0                    0                    0
             origin          destination            direction
                  0                    0                    0
        link_length           OBJECTID.x        SHAPE__Length
                  0                    0                    0
              DELAY         EXCESS_DELAY          TRAVEL_TIME
                  0                    0                    0
              SPEED           TIME_STAMP       LAPSE(in secs)
                  0                    0                    0
      ETL_TIMESTAMP difference_in_length
                  0                    0
```

Hide

```
#Special values check
spl_values_chk <- sapply(dataset_final, is.infinite)
head(spl_values_chk, n=50)
```

```
      link_id ROAD_NAME SECTION_DESCRIPTION origin destination direction
 [1,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [2,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [3,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [4,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [5,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [6,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [7,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [8,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
 [9,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[10,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[11,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[12,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[13,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[14,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[15,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[16,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[17,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[18,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[19,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[20,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[21,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[22,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[23,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[24,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[25,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[26,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[27,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[28,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[29,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[30,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[31,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[32,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[33,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[34,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[35,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[36,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[37,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[38,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[39,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[40,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[41,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[42,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[43,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[44,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[45,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[46,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[47,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[48,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[49,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
[50,]   FALSE     FALSE               FALSE  FALSE       FALSE     FALSE
      link_length OBJECTID.x SHAPE__Length DELAY EXCESS_DELAY TRAVEL_TIME SPEED
 [1,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
 [2,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
 [3,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
 [4,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
 [5,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
```

```
 [6,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
 [7,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
 [8,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
 [9,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[10,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[11,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[12,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[13,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[14,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[15,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[16,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[17,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[18,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[19,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[20,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[21,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[22,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[23,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[24,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[25,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[26,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[27,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[28,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[29,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[30,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[31,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[32,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[33,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[34,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[35,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[36,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[37,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[38,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[39,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[40,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[41,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[42,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[43,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[44,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[45,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[46,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[47,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[48,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[49,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
[50,]       FALSE      FALSE         FALSE FALSE        FALSE       FALSE FALSE
      TIME_STAMP LAPSE(in secs) ETL_TIMESTAMP difference_in_length
 [1,]      FALSE          FALSE         FALSE                FALSE
 [2,]      FALSE          FALSE         FALSE                FALSE
 [3,]      FALSE          FALSE         FALSE                FALSE
 [4,]      FALSE          FALSE         FALSE                FALSE
 [5,]      FALSE          FALSE         FALSE                FALSE
 [6,]      FALSE          FALSE         FALSE                FALSE
 [7,]      FALSE          FALSE         FALSE                FALSE
 [8,]      FALSE          FALSE         FALSE                FALSE
 [9,]      FALSE          FALSE         FALSE                FALSE
[10,]      FALSE          FALSE         FALSE                FALSE
[11,]      FALSE          FALSE         FALSE                FALSE
[12,]      FALSE          FALSE         FALSE                FALSE
```

```
[13,]      FALSE          FALSE          FALSE                FALSE
[14,]      FALSE          FALSE          FALSE                FALSE
[15,]      FALSE          FALSE          FALSE                FALSE
[16,]      FALSE          FALSE          FALSE                FALSE
[17,]      FALSE          FALSE          FALSE                FALSE
[18,]      FALSE          FALSE          FALSE                FALSE
[19,]      FALSE          FALSE          FALSE                FALSE
[20,]      FALSE          FALSE          FALSE                FALSE
[21,]      FALSE          FALSE          FALSE                FALSE
[22,]      FALSE          FALSE          FALSE                FALSE
[23,]      FALSE          FALSE          FALSE                FALSE
[24,]      FALSE          FALSE          FALSE                FALSE
[25,]      FALSE          FALSE          FALSE                FALSE
[26,]      FALSE          FALSE          FALSE                FALSE
[27,]      FALSE          FALSE          FALSE                FALSE
[28,]      FALSE          FALSE          FALSE                FALSE
[29,]      FALSE          FALSE          FALSE                FALSE
[30,]      FALSE          FALSE          FALSE                FALSE
[31,]      FALSE          FALSE          FALSE                FALSE
[32,]      FALSE          FALSE          FALSE                FALSE
[33,]      FALSE          FALSE          FALSE                FALSE
[34,]      FALSE          FALSE          FALSE                FALSE
[35,]      FALSE          FALSE          FALSE                FALSE
[36,]      FALSE          FALSE          FALSE                FALSE
[37,]      FALSE          FALSE          FALSE                FALSE
[38,]      FALSE          FALSE          FALSE                FALSE
[39,]      FALSE          FALSE          FALSE                FALSE
[40,]      FALSE          FALSE          FALSE                FALSE
[41,]      FALSE          FALSE          FALSE                FALSE
[42,]      FALSE          FALSE          FALSE                FALSE
[43,]      FALSE          FALSE          FALSE                FALSE
[44,]      FALSE          FALSE          FALSE                FALSE
[45,]      FALSE          FALSE          FALSE                FALSE
[46,]      FALSE          FALSE          FALSE                FALSE
[47,]      FALSE          FALSE          FALSE                FALSE
[48,]      FALSE          FALSE          FALSE                FALSE
[49,]      FALSE          FALSE          FALSE                FALSE
[50,]      FALSE          FALSE          FALSE                FALSE
```

Hide

```
Rules <- editfile("EditRules.txt", type = "all")
s1 <- violatedEdits(Rules,dataset_final)
summary(s1)
```

```
Edit violations, 1553 observations, 0 completely missing (0%):
```

	editname <fctr>	freq <dbl>	rel <fctr>
num2	num2	1	0.1%
dat1	dat1	1	0.1%

2 rows

```
Edit violations per record:
```

	errors <fctr>	freq <int>	rel <fctr>
0	0	1551	99.9%
1	1	2	0.1%

2 rows

Hide

```
Rules2 <- correctionRules("CorrectionRules.txt")
cor <- correctWithRules(Rules2, dataset_final)
cor$corrected
```

link_id <dbl>	ROAD_NAME <chr>	SECTION_DESCRIPTION <chr>
3	Bulleen Rd	Eastern Fwy to Manningham Rd
5	Greensborough Hwy	M80 to Grimshaw St
6	Greensborough Hwy	Grimshaw St to M80
7	Greensborough Hwy	Grimshaw St to Watsonia Rd
8	Greensborough Hwy	Watsonia Rd to Grimshaw St
9	Greensborough Hwy	Watsonia Rd to Lwr Plenty Rd
10	Greensborough Hwy	Lwr Plenty Rd to Watsonia Rd
11	Grimshaw St	Greensborough Hwy to The Concord
12	Grimshaw St	The Concord to Greensborough Hwy
13	Grimshaw St	Greensborough Hwy to The Circuit

1-10 of 1,553 rows | 1-4 of 17 columns Previous **1** 2 3 4 5 6 … 100 Next

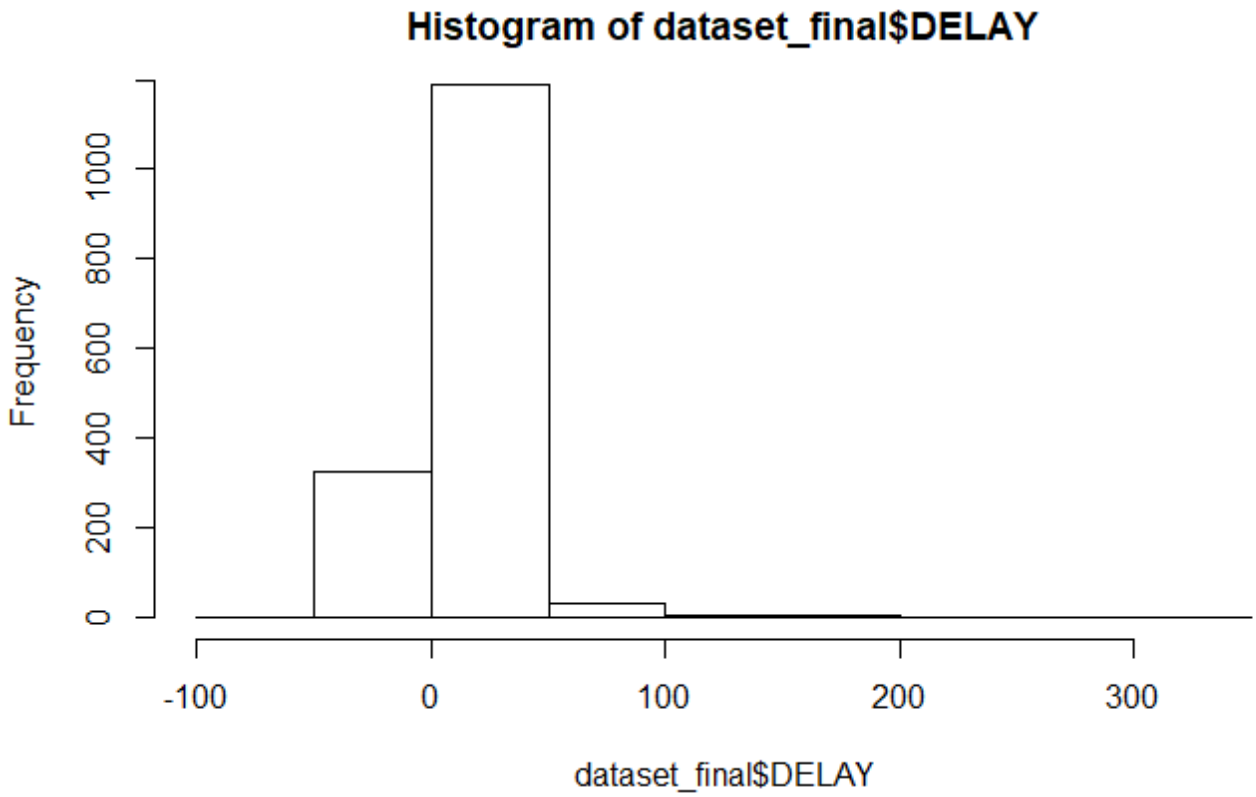
Scan II

- To scan for the outliers of numeric data, first I checked if any of the variables are approximately normally distributed, so that I can proceed with z-score method of outliers detection.
- So histogram of the variables have been checked and it is found that 'DELAY','EXCESS_DELAY','SPEED' & 'TRAVEL_TIME' variables are approximately normally distributed. Hence z-score method was used for these variables to detect the outliers.
- For the other two variables, tukey's method of outlier detection was used.
- Outliers have been handled with Capping/Windsoring method.
- For that, a function has been created to replace the values falling behind 5th and 95th percentile with 5th and 95th percentile values.
- I have chosen this method of handling outliers, because it does not involve any elimination.
- Lapply() function has been used as the 'capping' function needs to be done across many variables.

- The difference in variable values(removal of outliers) can be seen in ‘before cap’ and ‘after cap’ objects.

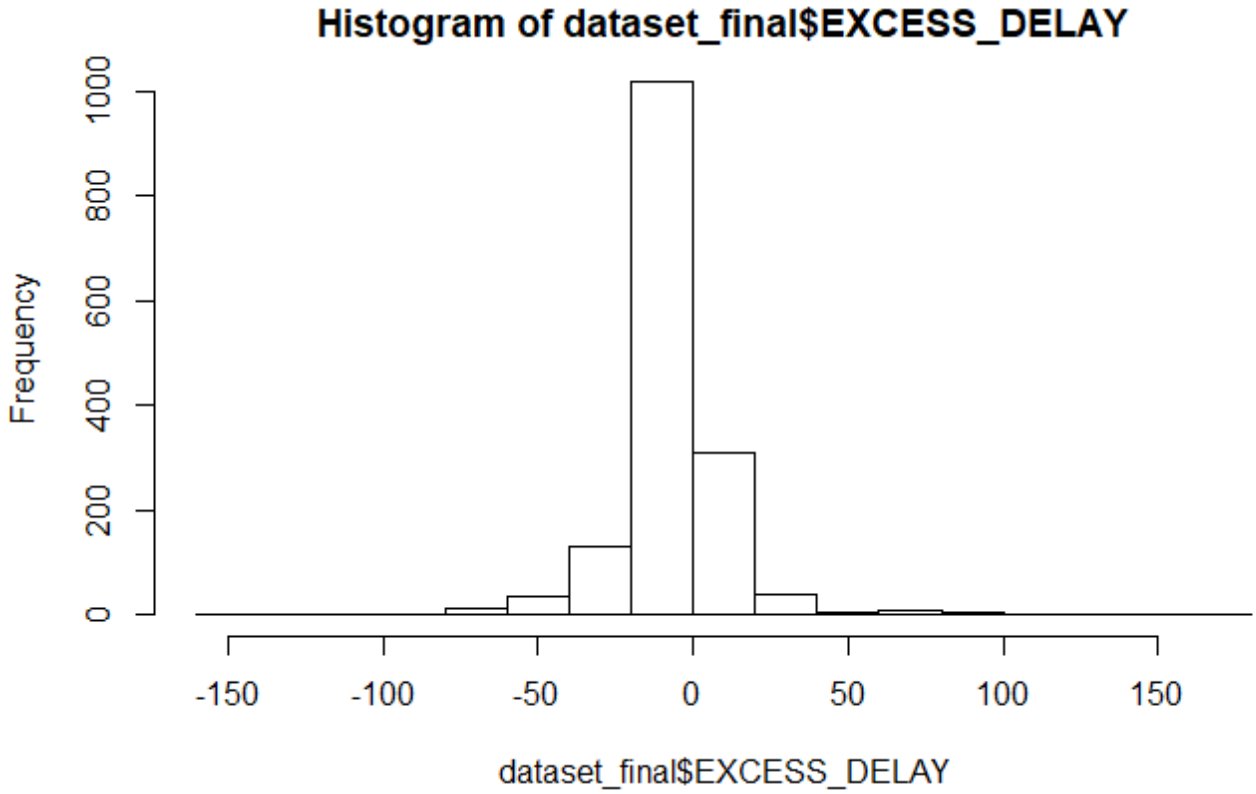
Hide

```
# This is the R chunk for the Scan II
delay <- hist(dataset_final$DELAY)
```



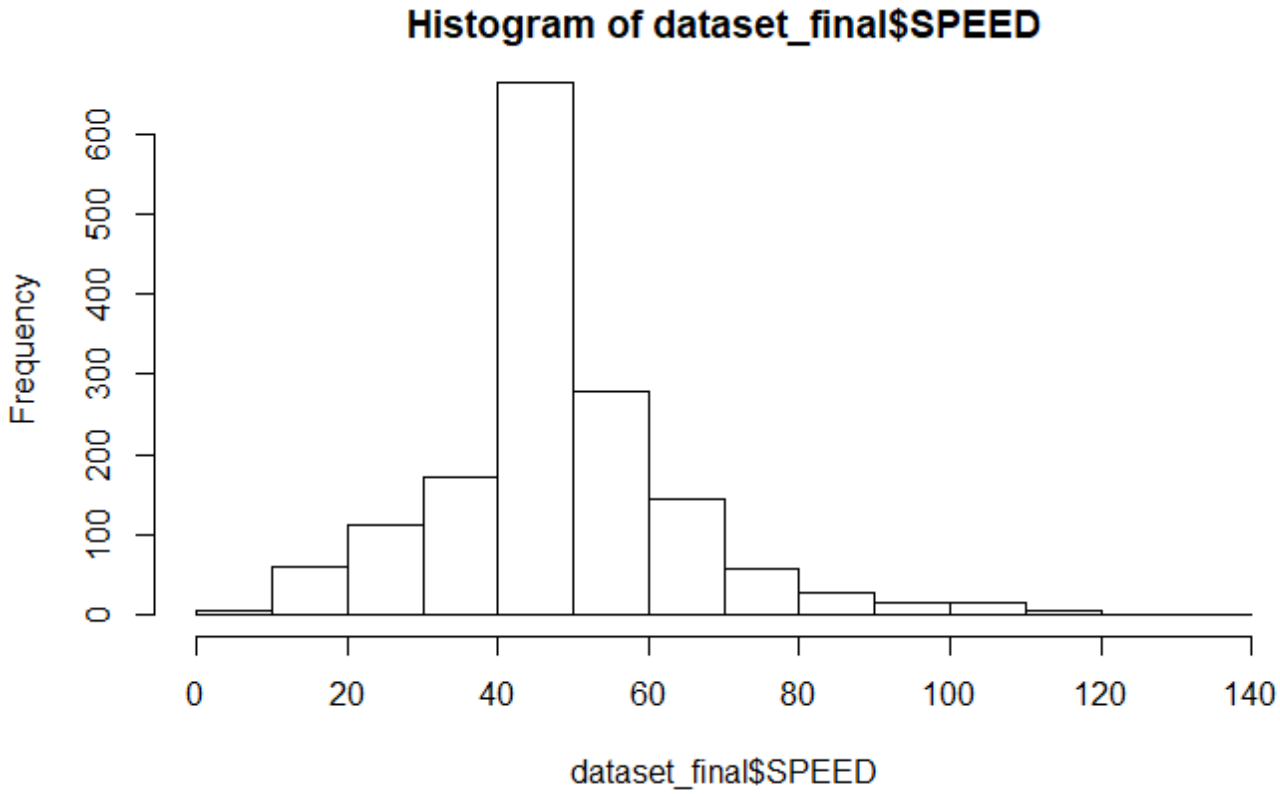
Hide

```
excess_delay <- hist(dataset_final$EXCESS_DELAY)
```



Hide

```
speed <- hist(dataset_final$SPEED)
```



Hide

```
#OUTLIERS DETECTION
delay_zs <- dataset_final$DELAY %>% scores(type = "z")
excdelay_zs <- dataset_final$EXCESS_DELAY %>% scores(type = "z")
speed_zs <- dataset_final$SPEED %>% scores(type = "z")
delay_outliers <- which(abs(delay_zs) > 3)
delay_outliers
```

```
 [1]  180  196  258  259  371  519  594  630  687  760  838  924  925  960  991
[16]  996 1118 1177 1178 1334 1398 1456 1457 1461 1462
```

Hide

```
excdelay_outliers <- which(abs(excdelay_zs) > 3)
excdelay_outliers
```

```
 [1]   27   78  180  196  209  259  305  371  687  760  767  829  833  924  960
[16]  991 1081 1107 1112 1118 1178 1180 1334 1398 1456 1458 1461 1463 1510
```

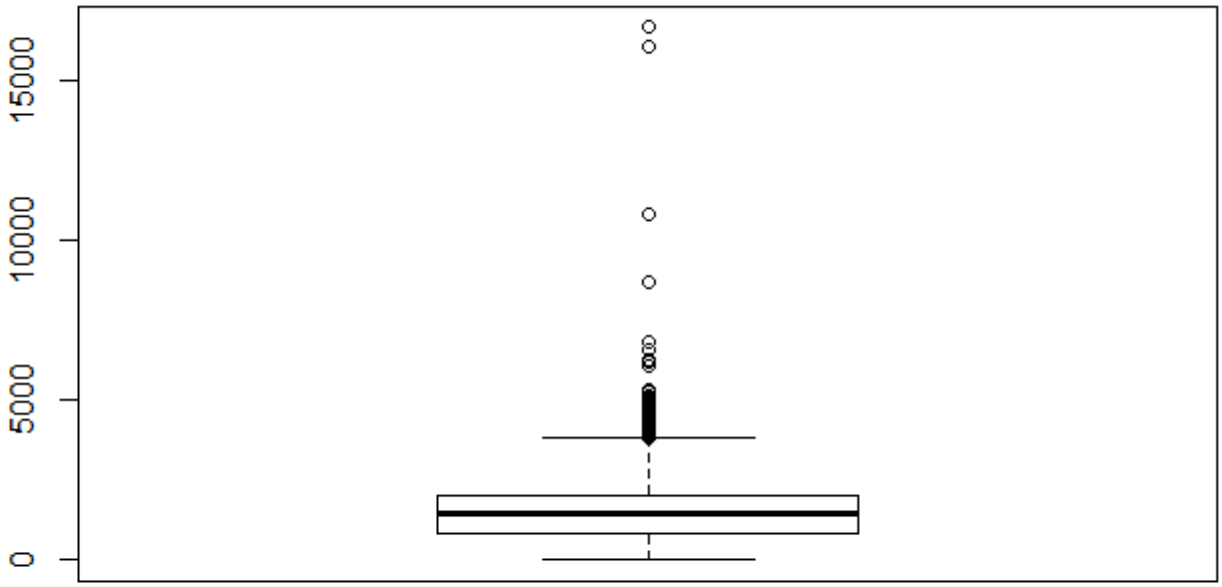
Hide

```
speed_outliers <- which(abs(speed_zs) > 3)
speed_outliers
```

```
 [1]    4   29   47  133  164  223  379  501  503  943 1035 1037 1038 1388 1390
[16] 1392 1394 1397 1403 1422 1423 1429 1480
```

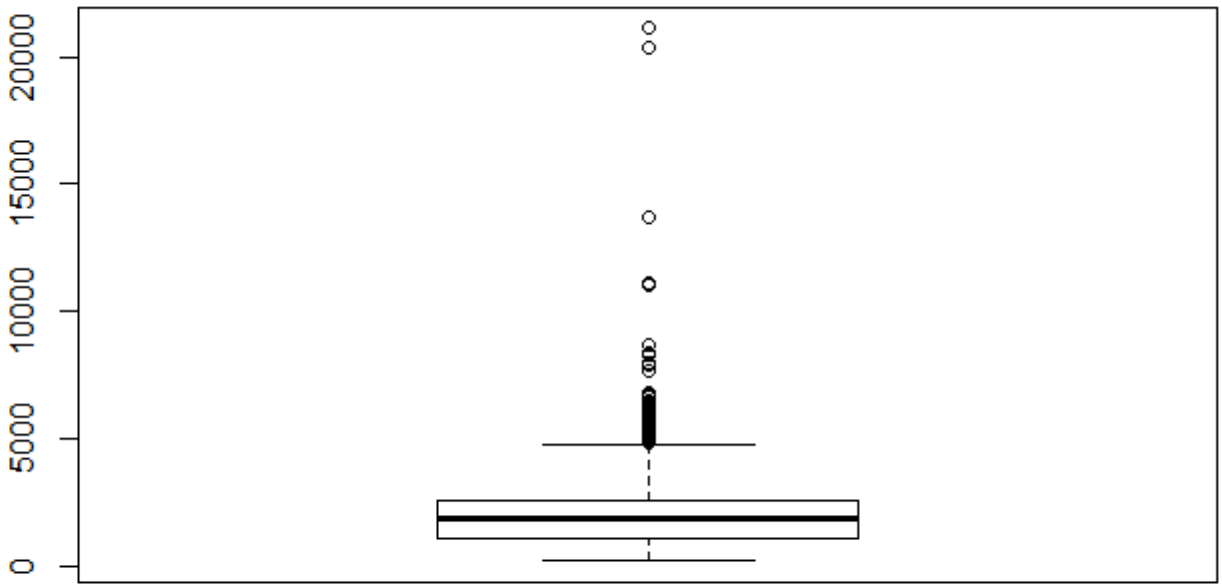
Hide

```
linklengthoutlier <- boxplot(dataset_final$link_length)
```



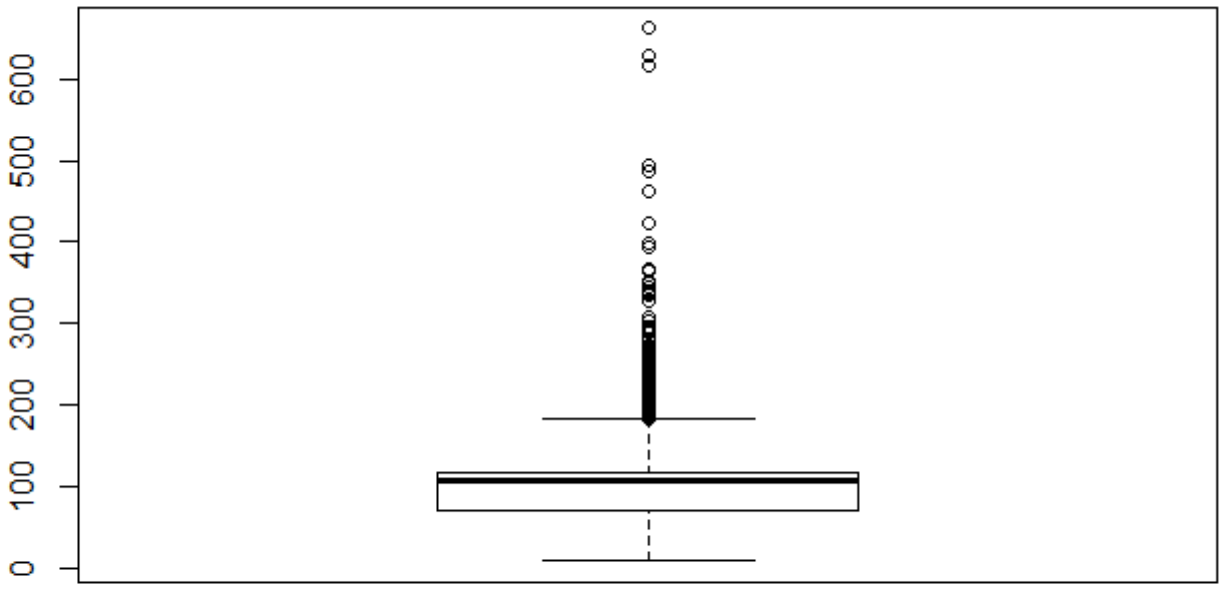
Hide

```
#converting impute class to numeric
dataset_final$SHAPE__Length <- as.numeric(dataset_final$SHAPE__Length)
dataset_final$TRAVEL_TIME <- as.numeric(dataset_final$TRAVEL_TIME)
shapelengthoutlier <- boxplot(dataset_final$SHAPE__Length)
```



Hide

```
Traveltimeoutlier <- boxplot(dataset_final$TRAVEL_TIME)
```



Hide

```
#DEALING WITH OUTLIERS
capping <- function(ds){
  quantiles <- quantile( ds, c(.05, 0.25, 0.75, .95 ) )
  ds[ ds < quantiles[2] - 1.5*IQR(ds) ] <- quantiles[1]
  ds[ ds > quantiles[3] + 1.5*IQR(ds) ] <- quantiles[4]
  ds
}
before_cap <- summary(dataset_final)
```

```
 1 values imputed to West


 378 values imputed to 10.36596


 378 values imputed to -5.610213


 378 values imputed to 49.34298


 378 values imputed to 11


 1 values imputed to 437.9565
```

Hide

```
before_cap
```

```
   link_id        ROAD_NAME         SECTION_DESCRIPTION     origin
 Min.   :   3.0   Length:1553        Length:1553         Min.   :  115
 1st Qu.: 540.0   Class :character   Class :character    1st Qu.:  990
 Median : 944.0   Mode  :character   Mode  :character    Median : 3115
 Mean   : 940.5                                         Mean   : 3616
 3rd Qu.:1361.0                                         3rd Qu.: 4273
 Max.   :1781.0                                         Max.   :31200

  destination          direction    link_length        OBJECTID.x
 Min.   :  115   West      :383   Min.   :    0.0   Min.   :   1
 1st Qu.: 1012   East      :374   1st Qu.:  818.3   1st Qu.: 390
 Median : 3132   North     :355   Median : 1429.3   Median : 778
 Mean   : 3618   South     :353   Mean   : 1633.2   Mean   : 778
 3rd Qu.: 4273   South East: 24   3rd Qu.: 2023.1   3rd Qu.:1166
 Max.   :31200   North West: 22   Max.   :16683.5   Max.   :1561
                 (Other)   : 42
 SHAPE__Length         DELAY         EXCESS_DELAY      TRAVEL_TIME
 Min.   :  213.4   Min.   :-52.00   Min.   :-150.00   Min.   :  9.0
 1st Qu.: 1037.6   1st Qu.:  2.00   1st Qu.: -11.00   1st Qu.: 71.0
 Median : 1817.8   Median : 10.37   Median :  -5.61   Median :107.2
 Mean   : 2072.2   Mean   : 10.37   Mean   :  -5.61   Mean   :107.2
 3rd Qu.: 2567.1   3rd Qu.: 12.00   3rd Qu.:   0.00   3rd Qu.:116.0
 Max.   :21099.7   Max.   :306.00   Max.   : 178.00   Max.   :662.0

     SPEED          TIME_STAMP                   LAPSE(in secs)
 Min.   :  4.00   Min.   :2020-03-06 23:52:30   Min.   :11
 1st Qu.: 42.00   1st Qu.:2020-03-06 23:52:30   1st Qu.:11
 Median : 49.34   Median :2020-03-06 23:52:30   Median :11
 Mean   : 49.34   Mean   :2020-03-06 23:52:30   Mean   :11
 3rd Qu.: 56.00   3rd Qu.:2020-03-06 23:52:30   3rd Qu.:11
 Max.   :136.00   Max.   :2020-03-06 23:52:30   Max.   :11

 ETL_TIMESTAMP                 difference_in_length
 Min.   :2020-03-06 23:53:18   Min.   :  44.48
 1st Qu.:2020-03-06 23:53:18   1st Qu.: 219.38
 Median :2020-03-06 23:53:19   Median : 384.09
 Mean   :2020-03-06 23:53:19   Mean   : 437.96
 3rd Qu.:2020-03-06 23:53:20   3rd Qu.: 541.73
 Max.   :2020-03-06 23:53:21   Max.   :4416.24
```

Hide

```
dataset_final[c("DELAY","EXCESS_DELAY","SPEED","TRAVEL_TIME","link_length","SHAPE__Length")]
 <- sapply(dataset_final[c("DELAY","EXCESS_DELAY","SPEED","TRAVEL_TIME","link_length","SHAPE_
_Length")], capping)
after_cap <- summary(dataset_final)
```

```
 1 values imputed to West


 378 values imputed to 11


 1 values imputed to 437.9565
```

Hide

```
after_cap
```

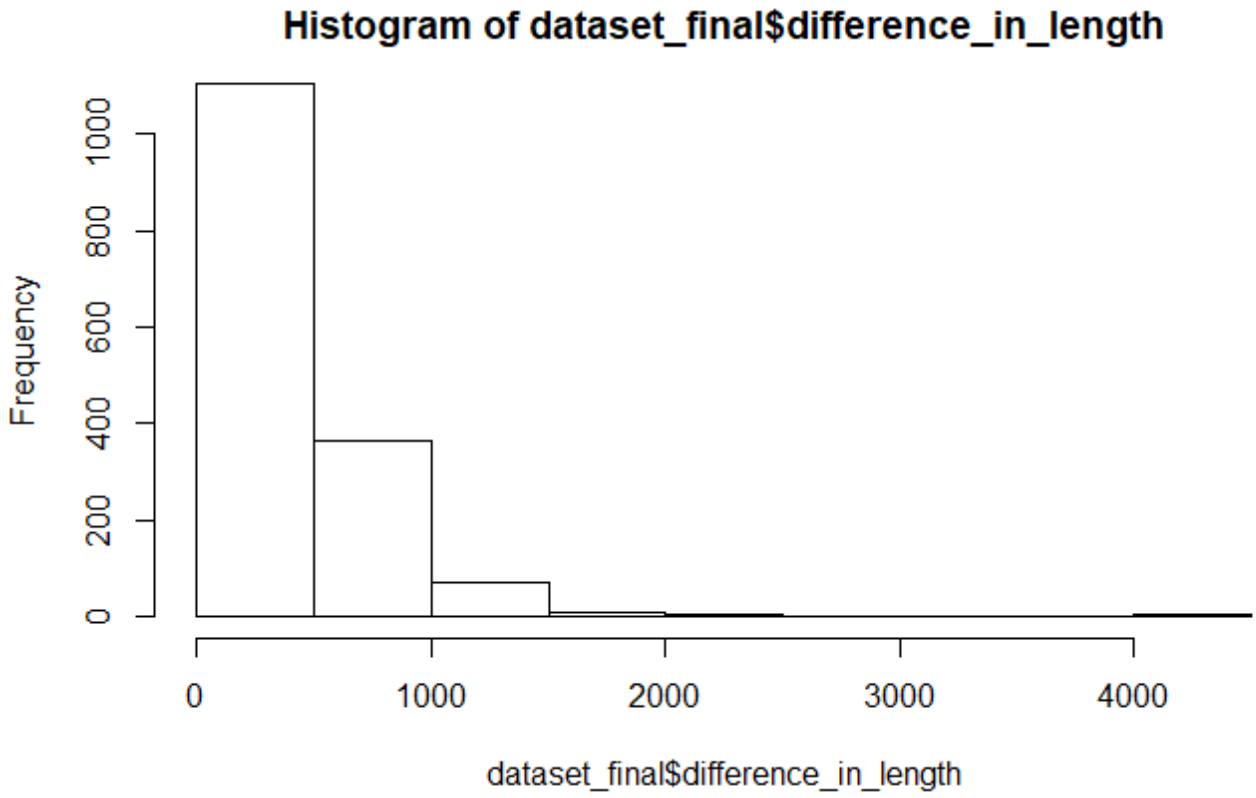
```
   link_id        ROAD_NAME         SECTION_DESCRIPTION     origin
Min.   :   3.0   Length:1553        Length:1553         Min.   :  115
1st Qu.: 540.0   Class :character   Class :character    1st Qu.:  990
Median : 944.0   Mode  :character   Mode  :character    Median : 3115
Mean   : 940.5                                         Mean   : 3616
3rd Qu.:1361.0                                         3rd Qu.: 4273
Max.   :1781.0                                         Max.   :31200

 destination         direction    link_length       OBJECTID.x   SHAPE__Length
Min.   :  115   West      :383   Min.   :   0.0   Min.   :   1   Min.   : 213.4
1st Qu.: 1012   East      :374   1st Qu.: 818.3   1st Qu.: 390   1st Qu.:1037.6
Median : 3132   North     :355   Median :1429.3   Median : 778   Median :1817.8
Mean   : 3618   South     :353   Mean   :1567.1   Mean   : 778   Mean   :1989.6
3rd Qu.: 4273   South East: 24   3rd Qu.:2023.1   3rd Qu.:1166   3rd Qu.:2567.1
Max.   :31200   North West: 22   Max.   :3865.6   Max.   :1561   Max.   :4927.6
                (Other)   : 42
    DELAY          EXCESS_DELAY      TRAVEL_TIME        SPEED
Min.   :-13.000   Min.   :-31.400   Min.   :  9.0   Min.   :21.00
1st Qu.:  2.000   1st Qu.:-11.000   1st Qu.: 71.0   1st Qu.:42.00
Median : 10.366   Median : -5.610   Median :107.2   Median :49.34
Mean   :  9.276   Mean   : -5.813   Mean   :103.1   Mean   :48.85
3rd Qu.: 12.000   3rd Qu.:  0.000   3rd Qu.:116.0   3rd Qu.:56.00
Max.   : 36.000   Max.   : 16.000   Max.   :209.0   Max.   :77.00

  TIME_STAMP                  LAPSE(in secs) ETL_TIMESTAMP
Min.   :2020-03-06 23:52:30   Min.   :11     Min.   :2020-03-06 23:53:18
1st Qu.:2020-03-06 23:52:30   1st Qu.:11     1st Qu.:2020-03-06 23:53:18
Median :2020-03-06 23:52:30   Median :11     Median :2020-03-06 23:53:19
Mean   :2020-03-06 23:52:30   Mean   :11     Mean   :2020-03-06 23:53:19
3rd Qu.:2020-03-06 23:52:30   3rd Qu.:11     3rd Qu.:2020-03-06 23:53:20
Max.   :2020-03-06 23:52:30   Max.   :11     Max.   :2020-03-06 23:53:21

difference_in_length
Min.   :  44.48
1st Qu.: 219.38
Median : 384.09
Mean   : 437.96
3rd Qu.: 541.73
Max.   :4416.24
```

Transform

- I have chosen the ‘difference_in_length’ variable to do transformation since this variable is not in normal form.
- First, histogram of the variable is checked and is found that the distribution of that variable is right skewed.
- After checking for different transformation methods, BoxCox function in the ‘forecast’ package is used to reduce the skewness and increase the normality.
- The variable is passed into BoxCox function with lamba parameter set to auto.
- After BoxCox, the variable seems to be normally distributed.

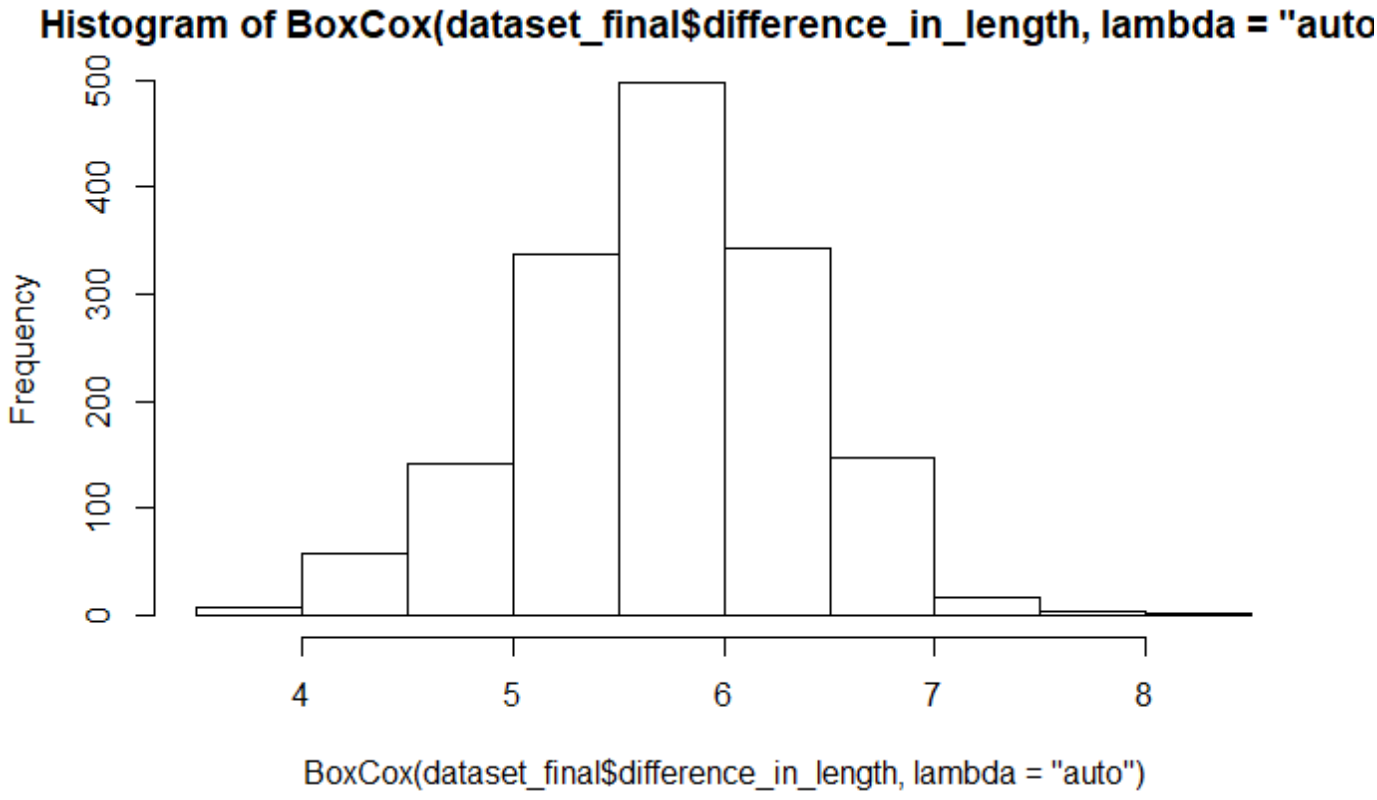
Hide

```
# This is the R chunk for the Transform Section
hist_DIL <- hist(dataset_final$difference_in_length)
```



Hide

```
DIL_Boxcox <- hist(BoxCox(dataset_final$difference_in_length,lambda = "auto"))
```



Reference

[1] MKR,24 Mar'18, "Populating missing Date and Time in time-series data in R, with zoo package", https://stackoverflow.com/questions/49460958/populating-missing-date-and-time-in-time-series-data-in-r-with-zoo-package (https://stackoverflow.com/questions/49460958/populating-missing-date-and-time-in-time-series-data-in-r-with-zoo-package)

[2] Dr. Anil Dolgun, 2018, "MATH2349 Data Wrangling", http://rare-phoenix-161610.appspot.com/secured/index.html (http://rare-phoenix-161610.appspot.com/secured/index.html)