Data Wrangling Assessment Task 2: Creating and pre-processing synthetic data

Uber data

Ammarah Naveed

Here are the packages required to produce the report:

```
library(magrittr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(outliers)
library(knitr)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:magrittr':
##
##     extract
```

```
library(openxlsx)
library(writexl)
library(deducorrect)
```

```
## Loading required package: editrules

## Loading required package: igraph

##
## Attaching package: 'igraph'
```

```
## The following object is masked from 'package:tidyr':
##
##     crossing

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union

##
## Attaching package: 'editrules'

## The following objects are masked from 'package:igraph':
##
##     blocks, normalize

## The following objects are masked from 'package:tidyr':
##
##     contains, separate

## The following object is masked from 'package:dplyr':
##
##     contains
```

```
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(igraph)
```

Uber dataset:

Getting a ride from an Uber driver is beautiful in its simplicity: simply open the app, set the pickup location, request a car, get picked up and pay with the tap of a button. But there's a great deal of data wrangling going on to make all of this happen in a (relatively) smooth process. Add to that the fact that sometimes there are things out of even Uber's control, like poor city transportation infrastructure, traffic jams, uncooperative drivers and much more.

Despite these setbacks, Uber has changed the way we move and is poised to change even more as the service comes to more and more cities. But what's actually happening in the background and what can we learn from it?

Uber uses your personal data in an anonymised and aggregated form to closely monitor which features of the Service are used most, to analyze usage patterns and to determine where we should offer or focus our Service. We may share this information with third parties for industry analysis and statistics. 2022, by Neil Patel Digital, LLC [id]:https://neilpatel.com/blog/how-uber-uses-data/

I am going to use uber data set here that might be helpful for uber to monitor hot spot areas in terms of where the traffic is flowing.I am mainly focusing on data within Sydney area, trip duartion, trip fare, trip departure,trip arrival and date as well.This data can also be helpful to show the profit and comission on each trip with kilometers.

I am going to create two data set as per the assignment requirement. 8 variables in data set one with 59 observations and second data set contains 6 variables with 59 observations as well and at the end i am going to mege these two synthetic dataset containing 12 variables and 118 observations.

- list of variables in mergerd data set:
  - trip_id
  - customer_name
  - customer_gender
  - address_departure
  - address_arrival
  - km's
  - total_fare
  - total_comission
  - travel_time
  - customer_id
  - trip_date

data set 1:

The 1st variable in dataset 1 is customer_names, I have generated it by using a website which has random name generator tool. [id]: https://www.fakenamegenerator.com/ First I exported it to excel sheet and created a formula of double quotes for each name and then paste it here under the variable of customer_names.

```
x <- 1:60
n <- 60
SEED <- 234


set.seed(SEED)
```

```
customer_names <- c("Sharon","Henry","Brandon","Catherine","Julian","Rita","Sara","Patricia","James","R

#n <- 59
#set.seed(SEED)
```

kms

Next is kilometers, we sometimes need to know the distance of each trip to analyse many things like cost of petrol, profit for Uber and profit for rider etc... according to the distance being covered.As this is not a real data I have tried to create the kilometer observations using the rnorm function.This function needs mean value and standard deviation and some seed value to create some random numbers for kilometers variable synthetically.

```
m <- 30
s <- 8
n <- 59
SEED <- 234
kms <- rnorm(n, mean = m, sd = s)
```

address_departure

address_departure varibale is helpful for uber company to figure out the hotspots and supply and demand issue.I wrote these addresses manually on excel sheet and paste it here to craete a variable.These variables are character vactor, we need double qoutes for them to writedown in fuction in r markdown.

```
address_departure <- c("street 7, baulkum hills",
"99 Bill Avenue, Marobarua",
"4, kellyville ridge",
"marickville road, kellyville",
"24,forest road liverpool",
"21, valley place circular quay",
"55, stephen road box hill",
"24,forest road liverpool",
"12 plaza,townsville sydney",
"4 born, macquarie university",
"55, stephen road box hill",
"2 woolsworths way bellavista",
"77 flynn street schofields",
"2 james street valcalause",
"21 craimer street ,kiribilli",
"22 sabreet street crownest",
"30 strew avenue castle hill",
"78 martin place",
"1 plum street Palm beach",
"41 raymond street bankstown",
"20 fairway drive kellyville",
"john palmer public school",
"1 Woolworths Way, Bella Vista",
"35 Noalimba Avenue, Earlwood",
"22 Kim Street, Bondi Beach",
"75 Marlin Avenue, Gunning",
"289 Robert Street, Crows Nest",
"21 craimer street kiribilli",
"4 born, macquarie university",
```

```
"293 Inglis Street, Parramatta",
"20 fairway drive kellyvill",
"2 felling street, Mcgrath Hills",
"55 Riverside, bella vista",
"49 mayfare avenue, kribilli",
"1 Wonderland place, liverpool",
"3 george street Sydney",
"18 Tay Street, Parramatta",
"11 Conrad Road, Kellyville Ridge",
"Unit 3, Raymond Street, Bansktown",
"133 Bellavenue, Baulkam Hills",
"88 Rouse Kee, Rouse Hill",
"90 Bill Street, Quakers Hill",
"18 Kim Street, Winsdor",
 "3 george street Sydney",
"1 Rek Road, Granville",
"38 Leme Street, Kirribilli",
"Main Road, Bondi Beach",
"33 troy street, Carlingford",
"6 felling avenue, Kenthurst",
"9 pedington street, CIrcular Quay",
"7 Napean street, The ponds",
"40 Quetta street, Riverstone",
"Westfield Burwood",
"96 Boundry Road Box Hills",
"109 Main Boulevard, The ponds",
"29 Robert Street, Schofields",
"11 Lahore street, Riverstone",
"22 Riverview street, The Ponds",
"9 Rockie street, Baulkham Hills")
```

address_arrival

This variable is helpful for the Uber company to know the distance traveled and area's in demand for uber ride. It is also useful for the customer for it's fare calculation and to figure out the time being traveled.This variable is also a character vector so same rule applies for this one.

```
address_arrival <- c("3 george street sydney",
"04 camero bayside, sydney",
"40 melinda street auburn, sydney",
"02 cathy street, lidcombe",
"40 rosville road, north sydney",
"88 relink,darling harbour",
"conrad road kellyville ridge",
"04 camero bayside, sydney",
"66 hey street,bondi beach",
"sydney airport",
"30 martin place",
"88 Brahman road box hill",
"41 raymond street bankstown",
"33, raymond street bellavista",
"24 medicine avenue crowsnest",
"2 pallmico street manly",
"99 fairy drive seven hills",
```

```
"22 meadow avenue north sydney",
"8 station street schofields",
"4 born, macquarie university",
"Westfield Parramatta",
"48 camero street box hill",
"29 Robert Street, Schofields",
"38 Kim Steet, Bondi Beach",
"Macqarie University, Macquarie",
"88 Bill Street, Kellyville",
"29 Kim Street, Crows Nest",
"24 medicine avenue crowsnest",
"38 fairway drive, kellyville",
"23 RIs Avenue, Harris Park",
"40 parramatta road, Auburn",
"21 craimer street, kiribilli",
"22 palace street, Seven Hills ",
"20 Westren road , Parramatta",
"Rashays restaurant, North Kellyville",
"1 Robert Place, Valclause",
"19 James Cook Aveneue, Bella Vista",
"33 Flynn Street, Schofields",
"1 Carlington Street, Campsie",
"23 Iowa Street, North Sydney",
"99 Bill Avenue, Marobarua",
"1 Woolworths way Bella Vista",
"Main Street, Wynyard",
"Main Road, Bondi Beach",
"George Street, Wynyard",
"22 Param Street, Blacktown",
"88 Carie Avenue, Box Hill",
"20 Window avenue, Richmond",
"22 swisse place, Penrith",
"11 Conrad Road, Kellyville Ridge",
"44 Riverbank Drive, Cherrybrook",
"58 Bellavenue, Baulkam Hills",
"29 Valley street, Seven Hills",
"12 Main street Mc Grath Hills",
"25 stelling street, Castle Hill",
"78 Martin place",
"4 Palmico street, Randwick",
"1 plum street Palm beach",
"2 Valley place, Wentworthville")
```

customer_gender This variable is useful to understand the audience of Uber rides, I have used set.seed to create the gender male and female randomly for 59 observations.It may be useful for the researchers on gender behavior, sometimes female rider want a female driver for her safety issues. This kind of analysis can be made by this gender data info. in Uber dataset.

```
n <- 59
gender <- c("Male", "Female")
set.seed(SEED)

customer_gender <- sample (gender, n, replace = TRUE)
```

total_fare

Total fare is very important variable as lots of analysis is based on this variable.I have create it manually because this data should look realistic so if i can make it with some random function, the distance and fare doesn't make sense for the real looking data set. companies can analyse it's profit,loss by total fare and drives can analyse their profit as well to figure out that which area is suitable for them to pick a ride.

```
total_fare <- c(
198,110,150,50,100,NA,45,160,130,185,140,37,87,109,134,220.05,150,78,78.2,88.0,196,67,161,179,67,48,75,
```

trip_id

This is unique trip id which usually generates by uber company which never changes. This trip_id is a unique identifier which always changes even if the customer is same.

I am now going to create a data frame for all of above variables by using data.frame function for more work on my report.

```
df <- data.frame(trip_id = 1:59,
                 customer_names, kms, address_departure, address_arrival,customer_gender, total_fare)

head(df)
```

```
##   trip_id customer_names      kms              address_departure
## 1       1         Sharon 35.28616        street 7, baulkum hills
## 2       2          Henry 13.57614      99 Bill Avenue, Marobarua
## 3       3        Brandon 18.00635            4, kellyville ridge
## 4       4      Catherine 41.76986   marickville road, kellyville
## 5       5         Julian 41.67311       24,forest road liverpool
## 6       6           Rita 31.12111 21, valley place circular quay
##                   address_arrival customer_gender total_fare
## 1           3 george street sydney            Male        198
## 2        04 camero bayside, sydney            Male        110
## 3 40 melinda street auburn, sydney          Female        150
## 4        02 cathy street, lidcombe          Female         50
## 5   40 rosville road, north sydney          Female        100
## 6        88 relink,darling harbour          Female         NA
```

export xlsx file

As per assignment requirement I am going to export my first data set file in xlsx file.

```
write_xlsx(df,"/Users/ammarahnaveed/Documents/first-dataset.xlsx")
```

** Now creating second data set for Uber**

travel_time This is a calculated time from departure to arrival.I create this variable synthetically to make it more realistic according to the distance being traveled. Travel_time is useful to figure out the estimates of time spent in ride's from different areas.it is also helpful for the driver to sort out that how many rides he/she can take in a day.

```
travel_time <- c("1:05:00",
"55:00","52:08","21:07","45;08:00","15:00","26:00","57:00","29:01","59:00","45:00","45:00","1:40:00","5
```

trip_date I have create this trip_date variable by using the sample function with the specified period of dates.Trip_date can be used to monitor that which month has most usage of uber in a year.

```
trip_date <-sample(seq(as.Date('2020/01/01'),
                       as.Date('2021/01/01'), by="day"), 59)
```

customer_id

It is usually unique customer id which uber has in it's records. customers don't know about this id and this is useful to get different data sets to analyse.I have used sample function to generate data for this variable with the random min and max values.

```
customers_id_range <- c(100:159)

customer_id <- sample( customers_id_range, n, replace = TRUE)
```

trip_id

In my second data set this variable is the same one as I have created in data set 1 as per assignment requirement.

```
trip_id <- c(1:59)
```

trip_comission Trip_comission shows the amount deducted per trip by Uber company.This variable can be used to analyse the amount that goes to uber company per trip.I have create it manually to show it more realistic for the dataset.

```
trip_comission <- c(35,20,25,10,20,7,10,32,NA,31,22,10,21,30,35,40,39,16,19,22,NA,13,29,34,18,13,66,NA,
```

payment_method

First things first, you need your account to be setup properly before you can order and pay for a ride. Assuming you've downloaded the app, you need to create an account or log-in, and then you can add your card and later on you will be asked to pay it through cash, card or paypal. I have created this payment_method variable to analyse the preference of payment method by the rider.

```
payment <- c("cash","card", "paypal")
n = 59
SEED = 234
set.seed(SEED)

payment_method <- sample( payment, n, replace = TRUE)
```

ratings

Ratings can be helpful to know the driver's history and it usually shows on the app when you book for the ride. It specifically shows for the individual driver.I have created this variable with a sample function specifying 1 to 5 number of stars for the individual trip. After created the variable from sample function i have transform this variable to ordered factor by using the ordered factor formula.

```
ratings <- c("1star","2star","3star","4star","5star")

n = 59
```

```
rating_trip <- sample(ratings, n, replace = TRUE)
rating_trip_factor_ordered <- factor(rating_trip, ordered = TRUE,
                                     levels = c("1star", "2star", "3star", "4star","5star"))
```

Here is the data frame for my second dataset.

```
df1 <- data.frame(trip_id, trip_comission, payment_method, travel_time,customer_id, trip_date,rating_tri
head(df1)
```

```
##   trip_id trip_comission payment_method travel_time customer_id  trip_date
## 1       1             35           cash     1:05:00         122 2020-07-01
## 2       2             20         paypal       55:00         159 2020-03-05
## 3       3             25           card       52:08         147 2020-11-21
## 4       4             10           card       21:07         155 2020-11-14
## 5       5             20           card    45;08:00         141 2020-12-14
## 6       6              7           card       15:00         133 2020-08-21
##   rating_trip_factor_ordered
## 1                      1star
## 2                      2star
## 3                      5star
## 4                      4star
## 5                      3star
## 6                      2star
```

Export xlsx file

Exporting xlsx file for data set 2:

```
write_xlsx(df1,"/Users/ammarahnaveed/Documents/second-dataset.xlsx")
```

Merged dataset As per assignment requirement I am going to merge my two of the uber data sets by using the following function. In assignment description i should have atleast one variable common in both the data sets. I have create common trip_id in both data sets and then merge these data sets by trip_id. Now in merged data set i have one trip_id variable.

```
df2 <- merge(df,df1,by="trip_id")
head(df2)
```

```
##   trip_id customer_names      kms              address_departure
## 1       1         Sharon 35.28616        street 7, baulkum hills
## 2       2          Henry 13.57614      99 Bill Avenue, Marobarua
## 3       3        Brandon 18.00635            4, kellyville ridge
## 4       4      Catherine 41.76986   marickville road, kellyville
## 5       5         Julian 41.67311       24,forest road liverpool
## 6       6           Rita 31.12111 21, valley place circular quay
##                   address_arrival customer_gender total_fare trip_comission
## 1           3 george street sydney            Male        198             35
## 2        04 camero bayside, sydney            Male        110             20
## 3 40 melinda street auburn, sydney          Female        150             25
## 4        02 cathy street, lidcombe          Female         50             10
## 5   40 rosville road, north sydney          Female        100             20
```

```
## 6        88 relink,darling harbour          Female         NA              7
##   payment_method travel_time customer_id  trip_date rating_trip_factor_ordered
## 1           cash     1:05:00         122 2020-07-01                      1star
## 2         paypal       55:00         159 2020-03-05                      2star
## 3           card       52:08         147 2020-11-21                      5star
## 4           card       21:07         155 2020-11-14                      4star
## 5           card    45;08:00         141 2020-12-14                      3star
## 6           card       15:00         133 2020-08-21                      2star
```

Now to understand merged data set i am going to use structure function to data set. This dataset have integer, numeric,character and date variables.These can be used to analyse rider's information, Uber financial decisions, trip timings and to moniter the supply and demand in different areas.

```
str(df2)
```

```
## 'data.frame':    59 obs. of  13 variables:
##  $ trip_id                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ customer_names           : chr  "Sharon" "Henry" "Brandon" "Catherine" ...
##  $ kms                      : num  35.3 13.6 18 41.8 41.7 ...
##  $ address_departure        : chr  "street 7, baulkum hills" "99 Bill Avenue, Marobarua" "4, kellyv
##  $ address_arrival          : chr  "3 george street sydney" "04 camero bayside, sydney" "40 melinda
##  $ customer_gender          : chr  "Male" "Male" "Female" "Female" ...
##  $ total_fare               : num  198 110 150 50 100 NA 45 160 130 185 ...
##  $ trip_comission           : num  35 20 25 10 20 7 10 32 NA 31 ...
##  $ payment_method           : chr  "cash" "paypal" "card" "card" ...
##  $ travel_time              : chr  "1:05:00" "55:00" "52:08" "21:07" ...
##  $ customer_id              : int  122 159 147 155 141 133 108 131 144 106 ...
##  $ trip_date                : Date, format: "2020-07-01" "2020-03-05" ...
##  $ rating_trip_factor_ordered: Ord.factor w/ 5 levels "1star"<"2star"<..: 1 2 5 4 3 2 5 1 4 1 ...
```

Now we need to find out the missing values in merged data set. colSums is the function for the identification of missing value in each column and it results 3 missing values in tatal_fare and 3 in total_comission. To fix these missing values i have tried the impute function to extract the mean of the variable's and it works for both of the variables to fix their missing values.

```
colSums(is.na(df2))
```

```
##                    trip_id              customer_names
##                          0                           0
##                        kms           address_departure
##                          0                           0
##             address_arrival             customer_gender
##                          0                           0
##                 total_fare              trip_comission
##                          3                           3
##              payment_method                 travel_time
##                          0                           0
##                 customer_id                   trip_date
##                          0                           0
## rating_trip_factor_ordered
##                          0
```

```
#imputed <- impute(df2$trip_comission, fun = mean)
#head(imputed$trip_comission)
df2$trip_comission[is.na(df2$trip_comission)] <- mean(df2$trip_comission, na.rm = TRUE)
df2$total_fare[is.na(df2$total_fare)] <- mean(df2$total_fare, na.rm = TRUE)

str(df2)
```

```
## 'data.frame':    59 obs. of  13 variables:
##  $ trip_id                   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ customer_names            : chr  "Sharon" "Henry" "Brandon" "Catherine" ...
##  $ kms                       : num  35.3 13.6 18 41.8 41.7 ...
##  $ address_departure         : chr  "street 7, baulkum hills" "99 Bill Avenue, Marobarua" "4, kellyv
##  $ address_arrival           : chr  "3 george street sydney" "04 camero bayside, sydney" "40 melinda
##  $ customer_gender           : chr  "Male" "Male" "Female" "Female" ...
##  $ total_fare                : num  198 110 150 50 100 ...
##  $ trip_comission            : num  35 20 25 10 20 ...
##  $ payment_method            : chr  "cash" "paypal" "card" "card" ...
##  $ travel_time               : chr  "1:05:00" "55:00" "52:08" "21:07" ...
##  $ customer_id               : int  122 159 147 155 141 133 108 131 144 106 ...
##  $ trip_date                 : Date, format: "2020-07-01" "2020-03-05" ...
##  $ rating_trip_factor_ordered: Ord.factor w/ 5 levels "1star"<"2star"<..: 1 2 5 4 3 2 5 1 4 1 ...
```

Manipulate data To make the merged data more useful I am going to create a new variable of total_profit for the driver,s analysis. This variable can be extracted bey using the data from total_fare and trip_comission.

```
df2 <- mutate(df2, trip_profit = total_fare - trip_comission)
head(df2)
```

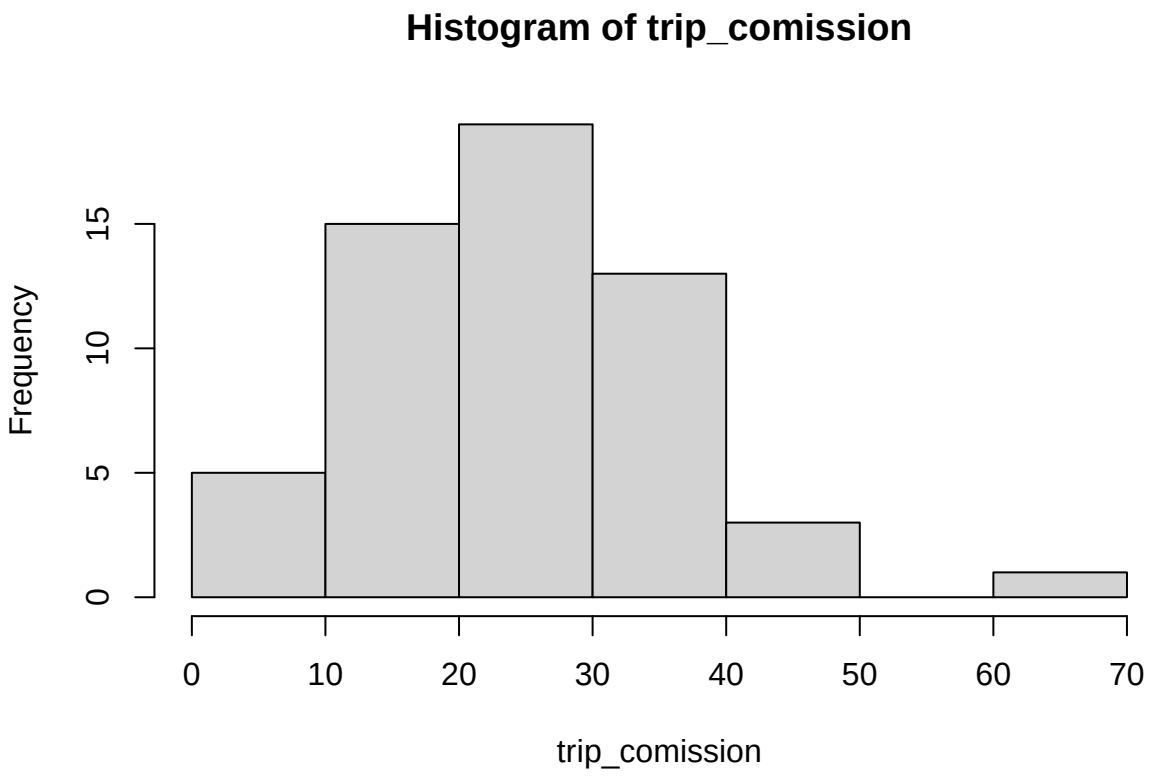
```
##   trip_id customer_names      kms              address_departure
## 1       1         Sharon 35.28616        street 7, baulkum hills
## 2       2          Henry 13.57614      99 Bill Avenue, Marobarua
## 3       3        Brandon 18.00635            4, kellyville ridge
## 4       4      Catherine 41.76986   marickville road, kellyville
## 5       5         Julian 41.67311       24,forest road liverpool
## 6       6           Rita 31.12111 21, valley place circular quay
##                    address_arrival customer_gender total_fare trip_comission
## 1           3 george street sydney            Male   198.0000             35
## 2        04 camero bayside, sydney            Male   110.0000             20
## 3 40 melinda street auburn, sydney          Female   150.0000             25
## 4        02 cathy street, lidcombe          Female    50.0000             10
## 5   40 rosville road, north sydney          Female   100.0000             20
## 6        88 relink,darling harbour          Female   111.5098              7
##   payment_method travel_time customer_id  trip_date rating_trip_factor_ordered
## 1           cash     1:05:00         122 2020-07-01                      1star
## 2         paypal       55:00         159 2020-03-05                      2star
## 3           card       52:08         147 2020-11-21                      5star
## 4           card       21:07         155 2020-11-14                      4star
## 5           card    45;08:00         141 2020-12-14                      3star
## 6           card       15:00         133 2020-08-21                      2star
##   trip_profit
## 1    163.0000
```

```
## 2     90.0000
## 3    125.0000
## 4     40.0000
## 5     80.0000
## 6    104.5098
```

There should be some outliers too, we can identify the outliers in each numeric variable. It can be done by using the box plot function as this function is used for parametric values. I can see 1 outlier in trip_comission function. I have picked the colour grey for box plot.

```
df2$trip_comission %>%
  boxplot(main = "Box Plot of trip_comission", ylab = "trip_comission", col = "grey")
```


```
hist(trip_comission)
```



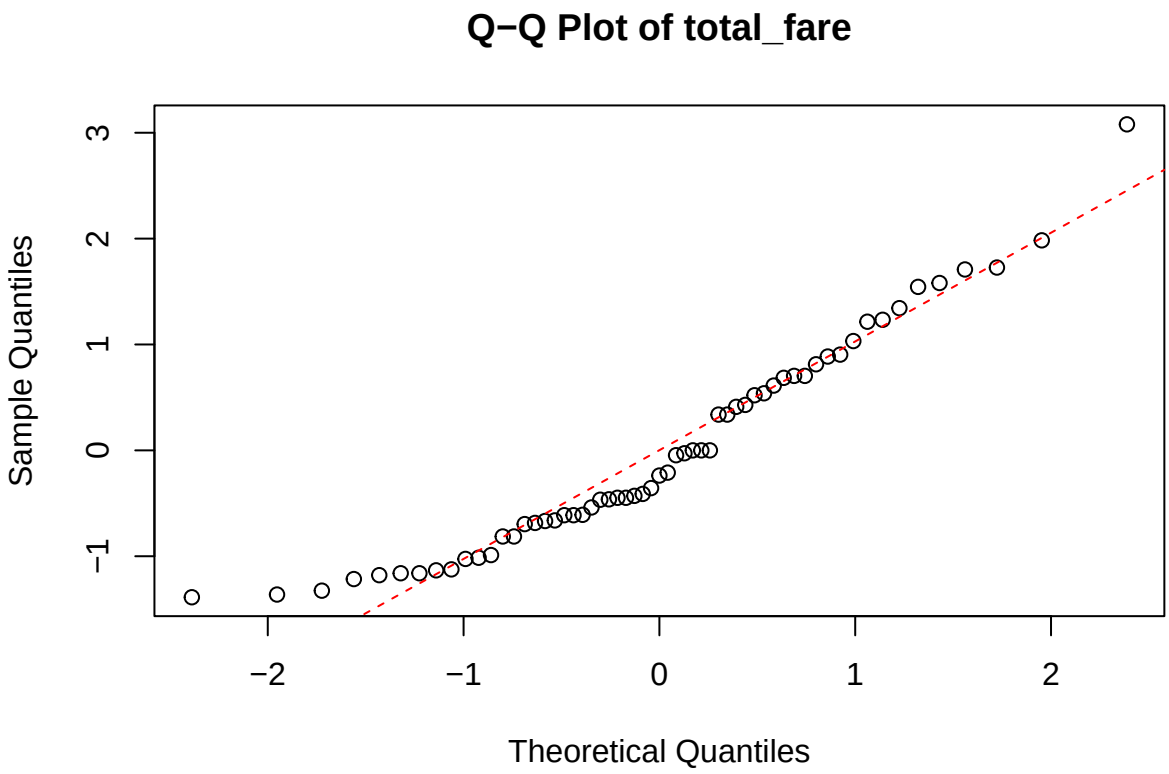
For total_fare z score function is used to find out outliers

```
z.scores <- df2$total_fare %>%
  outliers::scores(type = "z")
z.scores %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.3873 -0.6910 -0.2378  0.0000  0.6943  3.0792
```

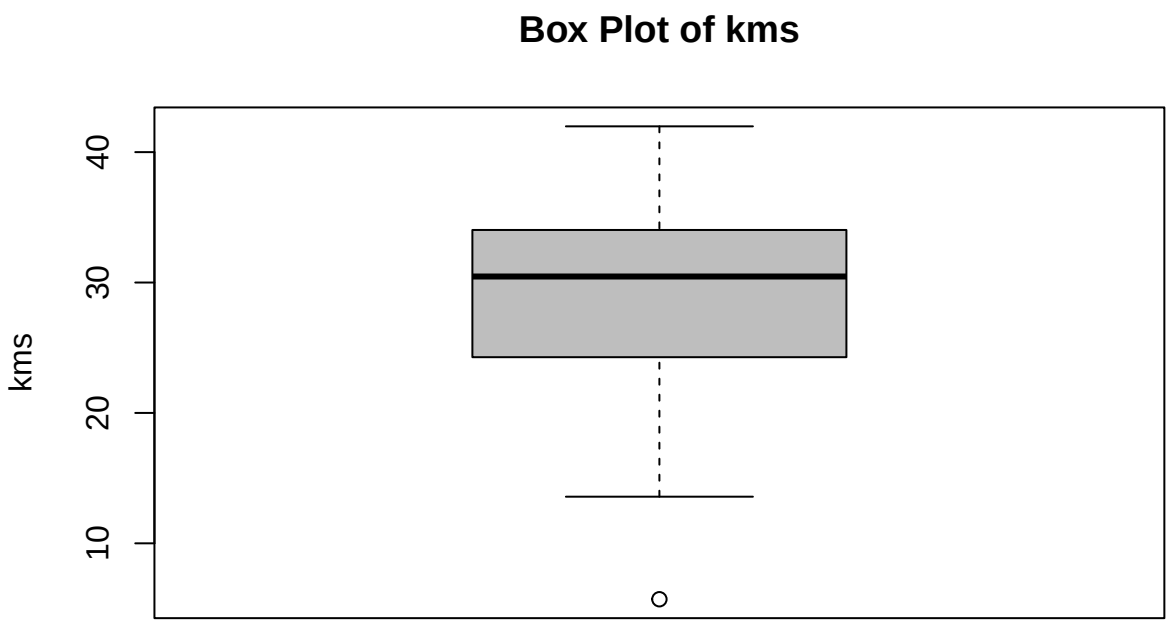
there is qqnorm and qqline function for total fare as well.

```
#df2$total_fare %>%
qqnorm(z.scores, main = "Q-Q Plot of total_fare")
qqline(z.scores, col = "red", lwd = 1, lty = 2)
```



In kms variable there is 1 outlier which shows in box plot funtion.

```
df2$kms %>%
  boxplot(main = "Box Plot of kms", ylab = "kms", col = "grey")
```



To find out the number of outliers, this is another function

```
iqr <- IQR(df2$kms)
q1 <- quantile(df2$kms, probs = 0.25)
q3 <- quantile(df2$kms, probs = 0.75)
lower_fence <- q1 - (1.5 * iqr) # Recall that the lower fence is Q1 minus the inter-quartile range
upper_fence <- q3 + (1.5 * iqr) # Recall that the upper fence is Q3 plus the inter-quartile range

up_outliers <- which(df2$kms > upper_fence)
low_outliers <- which(df2$kms < lower_fence)
```

There are no upper outlier for kilometers variable

```
length(up_outliers) # There are 0 outliers
```

```
## [1] 0
```

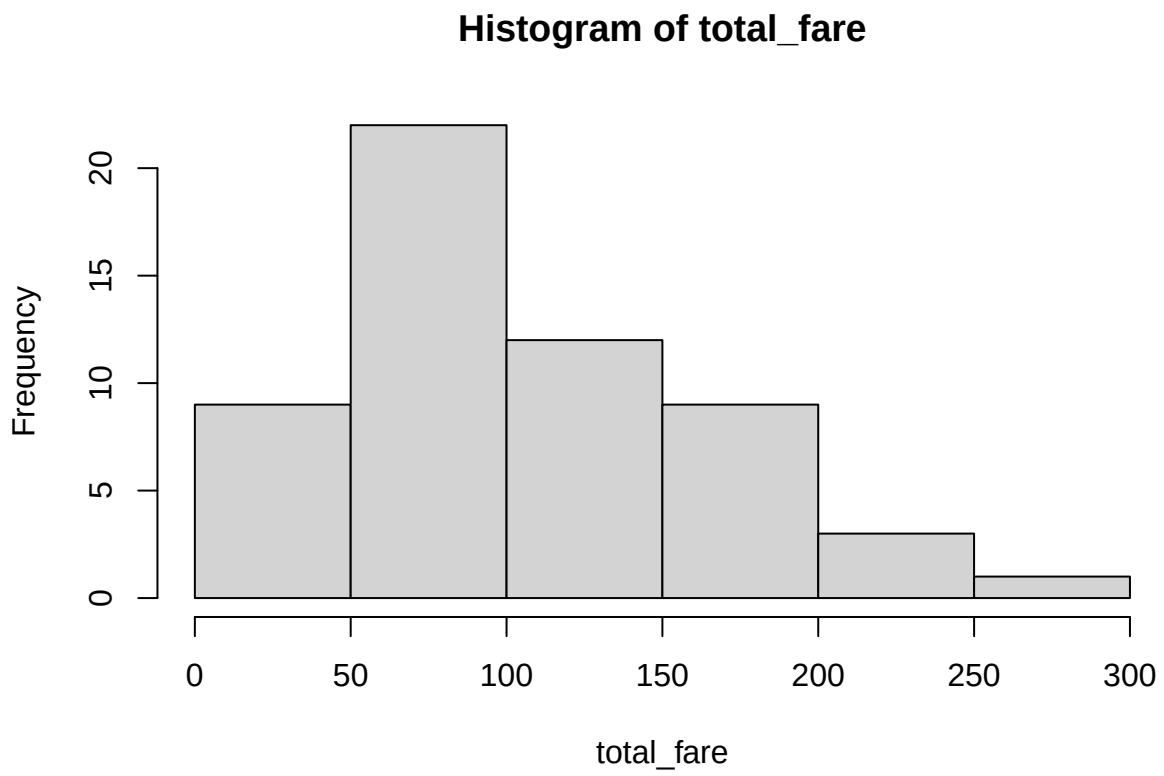
We have detected one lower outlier for kilometers variable

```
length(low_outliers) # There is 1 outlier
```

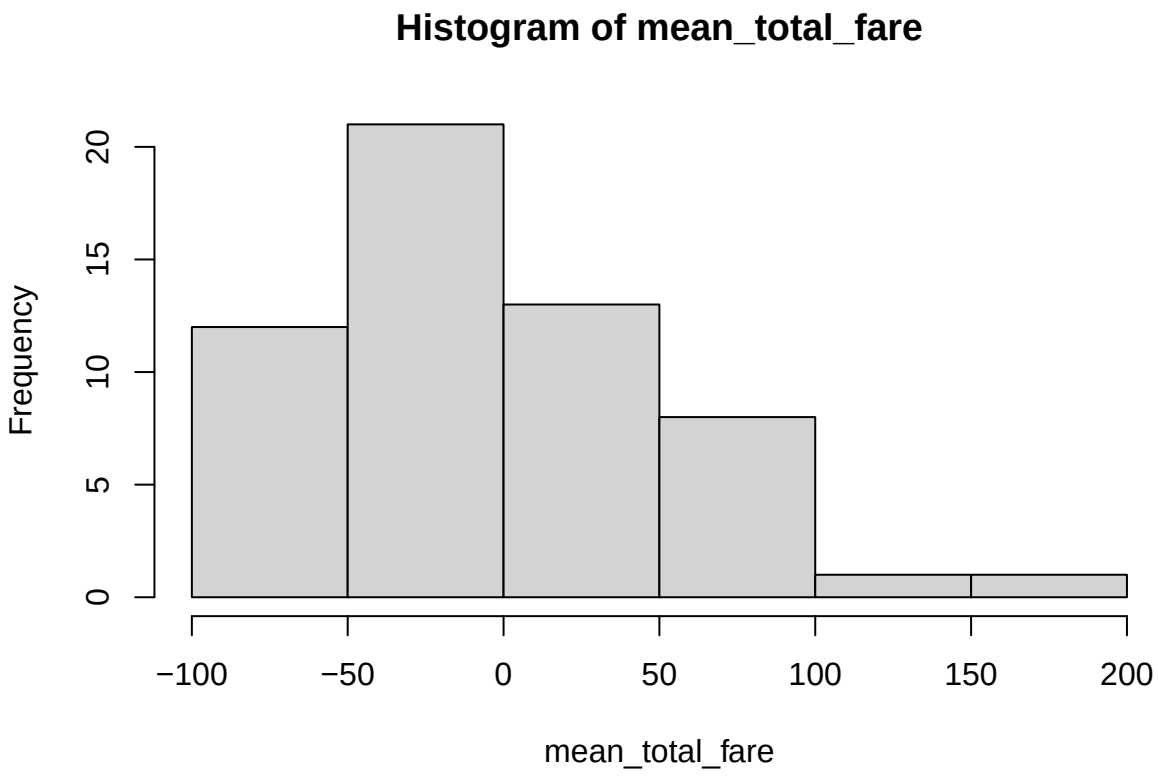
```
## [1] 1
```

I have transformed data by using total_fare varirablesuing following functions:

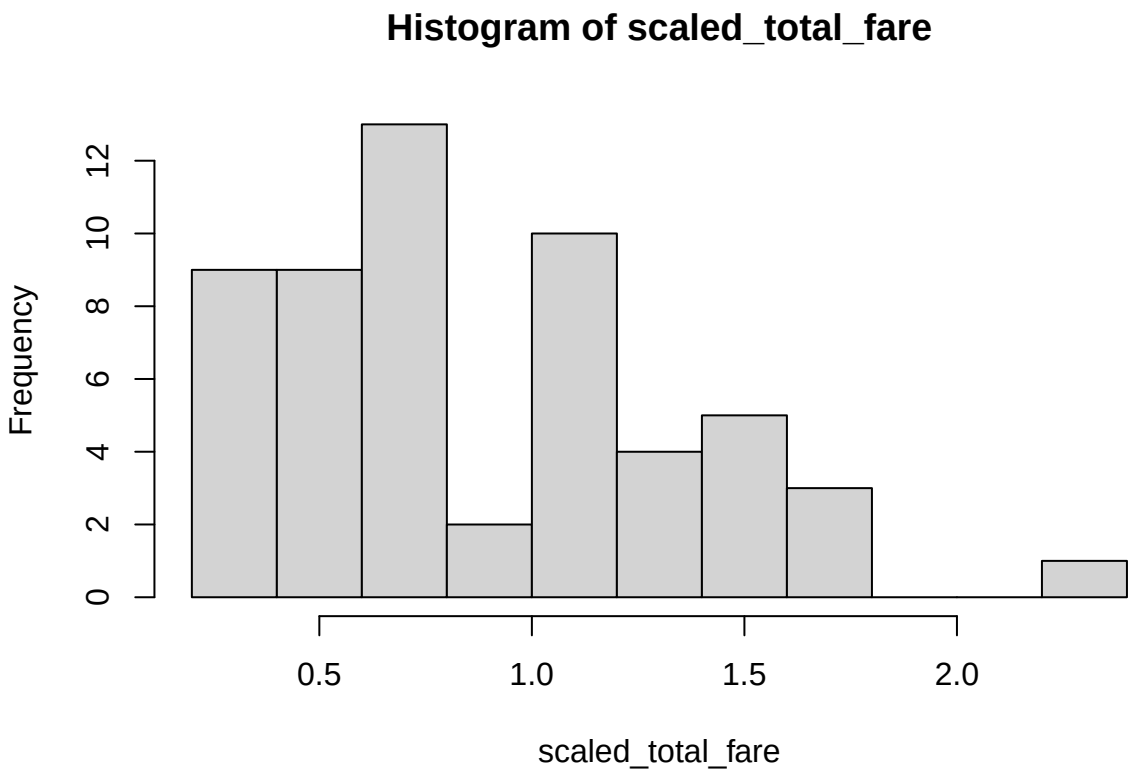
```r
min_max_norm <- function(x, na.rm = FALSE) {
  (x - min(x, na.rm = na.rm)) / (max(x, na.rm = na.rm) - min(x, na.rm = na.rm))
}

hist(total_fare)
```



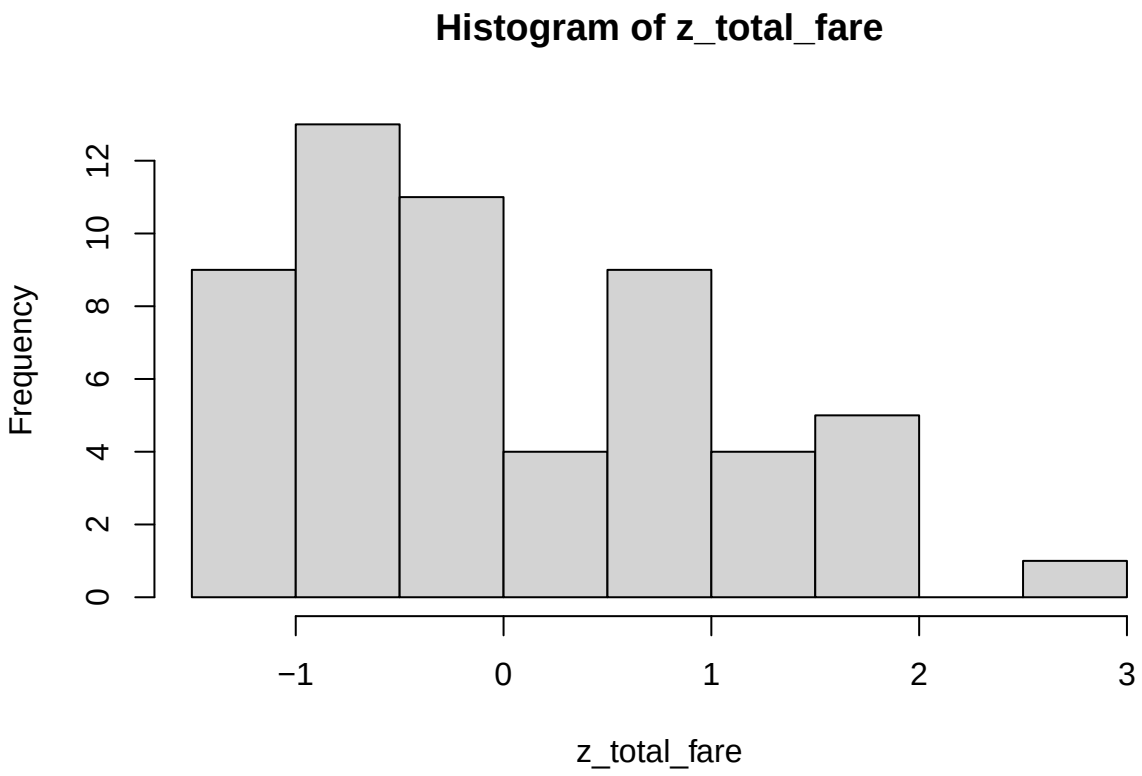

```r
mean_total_fare <- scale(total_fare, center = TRUE, scale = FALSE)
hist(mean_total_fare)
```



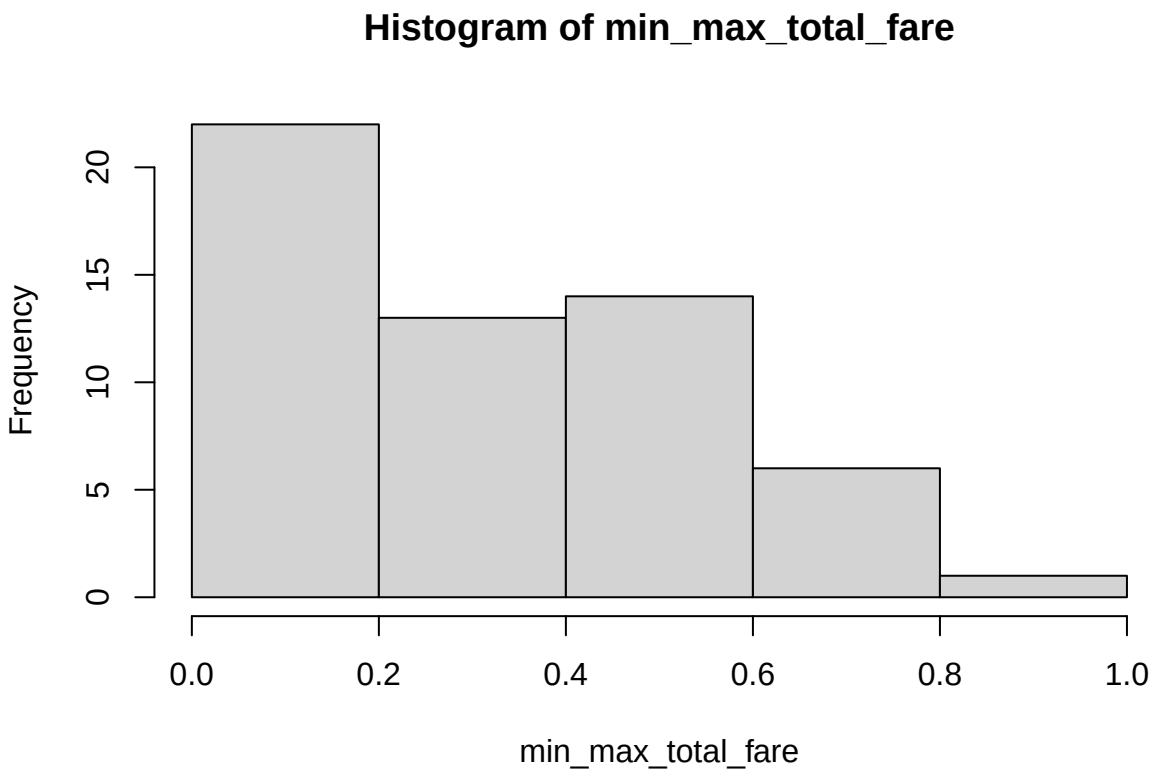

```
scaled_total_fare <- scale(total_fare, center = FALSE, scale = TRUE)
hist(scaled_total_fare)
```




```
z_total_fare <- scale(total_fare, center = TRUE, scale = TRUE)
hist(z_total_fare)
```




```
min_max_total_fare <- min_max_norm(total_fare, na.rm = TRUE)
hist(min_max_total_fare)
```



exporting the final data set

```
write_xlsx(df2,"/Users/ammarahnaveed/Documents/Merge-dataset.xlsx")
```

summary

Here is the summary statistics of kilometers grouped on payment method & total fare grouped on gender.

```
setNames(aggregate(df2$kms, by=list(df2$payment_method), FUN=mean),
         c("Payment Method","Average kms per payment method"))
```

```
##   Payment Method Average kms per payment method
## 1           card                       29.44678
## 2           cash                       28.69899
## 3         paypal                       29.15641
```

```
setNames(aggregate(df2$kms, by=list(df2$payment_method), FUN=median),
         c("Payment Method", "Median kms per payment method"))
```

```
##   Payment Method Median kms per payment method
## 1           card                       30.39751
## 2           cash                       30.49578
## 3         paypal                       30.46288
```

```
setNames(head(aggregate(df2$kms, by=list(df2$payment_method), FUN=min),10),
         c("Payment Method","Minimum kms per payment method"))
```

```
##   Payment Method Minimum kms per payment method
## 1           card                      18.006352
## 2           cash                       5.711281
## 3         paypal                      13.576136
```

```
max_data <- aggregate(df2$kms, by=list(df2$payment_method), FUN=max)
setNames(head(max_data[order(-max_data$x),],10),
         c("Payment Method","Maximum kms per payment method"))
```

```
##   Payment Method Maximum kms per payment method
## 3         paypal                       41.97718
## 2           cash                       41.82910
## 1           card                       41.76986
```

```
setNames(aggregate(df2$kms, by=list(df2$payment_method), FUN=sd),
         c("Payment Method","Standard deviation kms per payment method"))
```

```
##   Payment Method Standard deviation kms per payment method
## 1           card                                  6.920561
## 2           cash                                  8.829019
## 3         paypal                                  7.172900
```

```
# Summary of total_fare per different genders.

setNames(aggregate(df2$total_fare, by=list(df2$customer_gender), FUN=mean),
         c("Gender","Average fare per trip"))
```

```
##   Gender Average fare per trip
## 1 Female              111.1957
## 2   Male              111.8348
```

```
setNames(aggregate(df2$total_fare, by=list(df2$customer_gender), FUN=median),
         c("Gender", "Median fare per trip"))
```

```
##   Gender Median fare per trip
## 1 Female                96.0
## 2   Male                98.5
```

```
setNames(head(aggregate(df2$total_fare, by=list(df2$customer_gender), FUN=min),10),
         c("Gender","Minimum fare per trip"))
```

```
##   Gender Minimum fare per trip
## 1 Female                 35.6
## 2   Male                 37.0
```

```r
max_data_gender <- aggregate(df2$total_fare, by=list(df2$customer_gender), FUN=max)
setNames(head(max_data_gender[order(-max_data_gender$x),],10),
         c("Gender","Maximum rating per trip"))
```

```
##   Gender Maximum rating per trip
## 2   Male                 280.00
## 1 Female                 220.05
```

```r
setNames(aggregate(df2$total_fare, by=list(df2$customer_gender), FUN=sd),
         c("Gender","Standard deviation rating per trip"))
```

```
##   Gender Standard deviation rating per trip
## 1 Female                           49.80916
## 2   Male                           60.27017
```