CST4070 - Submmative CW2 - Bike Prediction By Tan Le Ping (M00724895)

Problem definition

Goal: To balance the bike sharing suppy and demand by forecast the number of bikes rented with temporal granularity of time slot one hour in each bike station via exploration of data and create a linear Regression Model to predict bike sharing demand. Three datasets are available: Bike journeys, bike stations and London census. Bike journeys are group by station Id and time. Spatial granularity: District level. Temporal granularity: Time and Date.

Pre-processing

Import all the data and create appropriate environment.

```
#read data
library(data.table)
journey = fread("bike_journeys.csv")
station = fread("bike_stations.csv")
census = fread("London_census.csv")
```

Firstly, have a view of all the data with basic exploration.

```
head(journey)
```

Journey_Duration <dbl>	Journey_ID <int>	End_D… <int>	End_Mo… <int>	End_Y… <int>	End_H… <int>	End_Minute <int>	End_S
2040	953	19	9	17	18	0	
1800	12581	19	9	17	15	21	
1140	1159	15	9	17	17	1	
420	2375	14	9	17	12	16	
1200	14659	13	9	17	19	33	
1320	2351	14	9	17	14	53	

6 rows | 1-8 of 14 columns

```
head(station)
```

Station_ID <int>	Capacity <int>	Latitude <dbl>	Longitude <dbl>	Station_Name <chr>
1	19	51.52916	-0.109970	River Street , Clerkenwell
2	37	51.49961	-0.197574	Phillimore Gardens, Kensington

Station_ID <int>	Capacity <int>	Latitude <dbl>	Longitude <dbl>	Station_Name <chr>
3	32	51.52128	-0.084605	Christopher Street, Liverpool Street
4	23	51.53006	-0.120973	St. Chad's Street, King's Cross
5	27	51.49313	-0.156876	Sedding Street, Sloane Square
6	18	51.51812	-0.144228	Broadcasting House, Marylebone

6 rows

```
head(census)
```

WardCode <chr>	WardName <chr>	borough <chr>	N... <chr>	AreaS... <dbl>	lon <dbl>	lat <dbl>	Inco
E05000026	Abbey	Barking and Dagenham	East	1.3	0.077935	51.53971	
E05000027	Alibon	Barking and Dagenham	East	1.4	0.148270	51.54559	
E05000028	Becontree	Barking and Dagenham	East	1.3	0.118957	51.55453	
E05000029	Chadwell Heath	Barking and Dagenham	East	3.4	0.139985	51.58475	
E05000030	Eastbrook	Barking and Dagenham	East	3.5	0.173581	51.55365	
E05000031	Eastbury	Barking and Dagenham	East	1.4	0.105683	51.53590	

6 rows | 1-8 of 20 columns

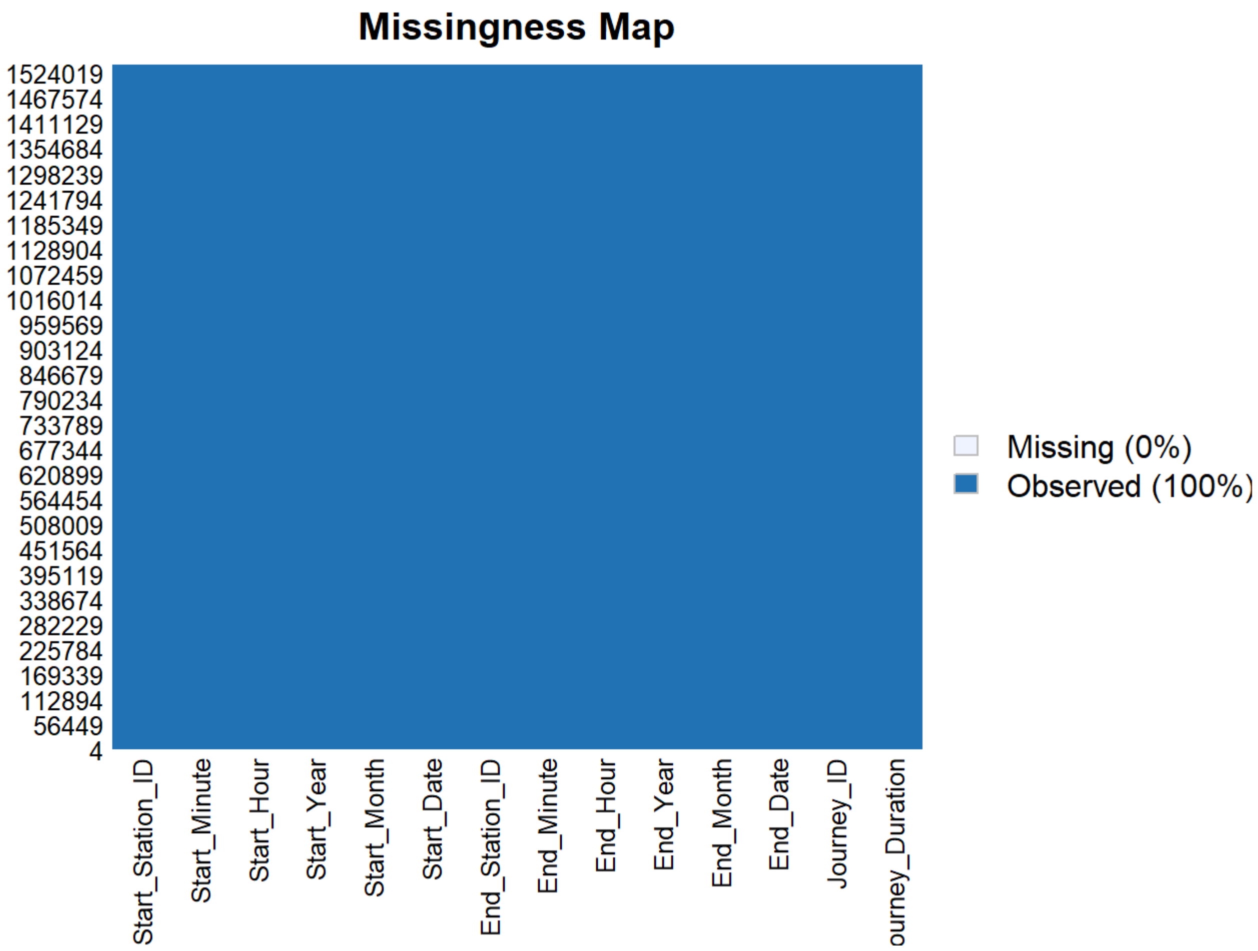
Secondly, to identify the missing values by inspecting if the data contains null values and outliers have to be carried out. The @Amelia package is used to investigate the missing value in data.

```
library(Amelia)
```

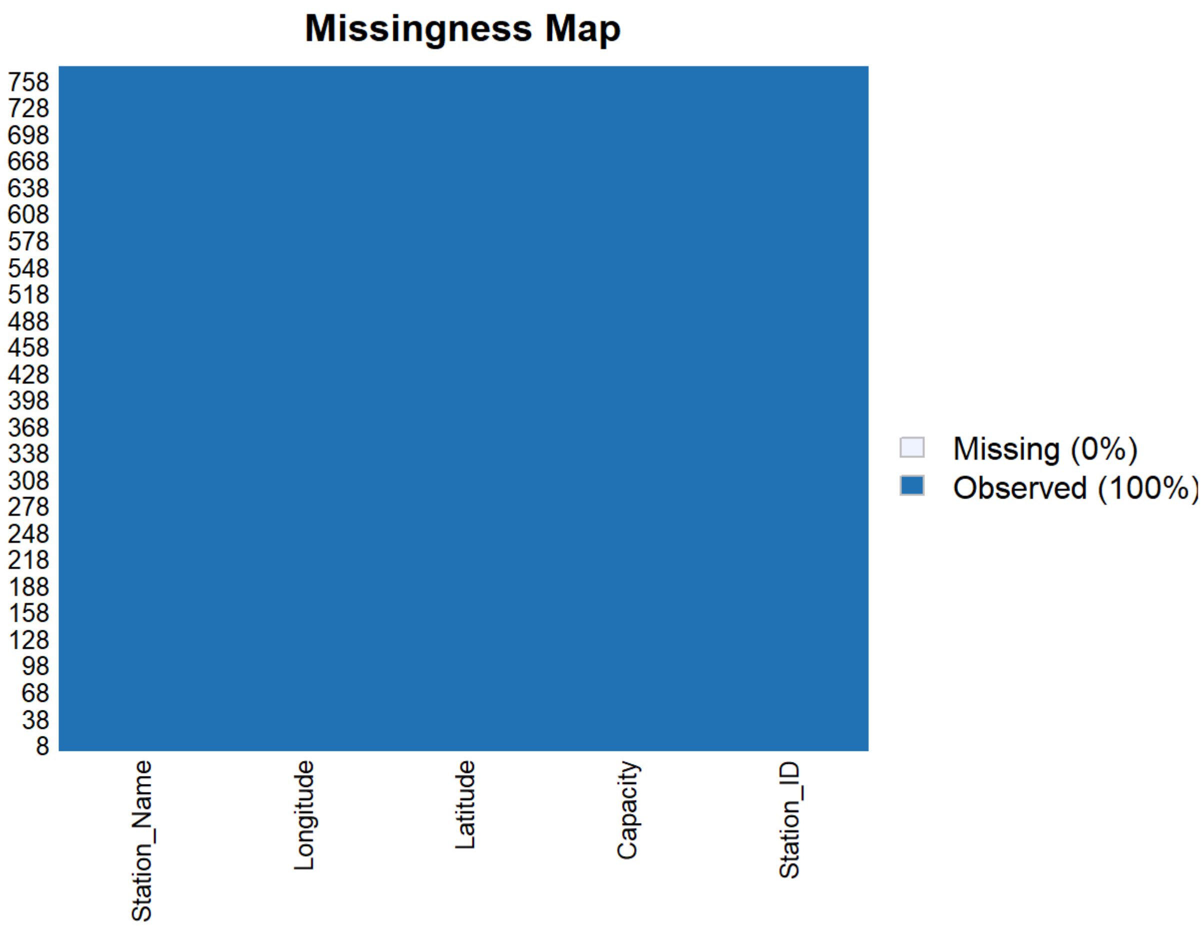
```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

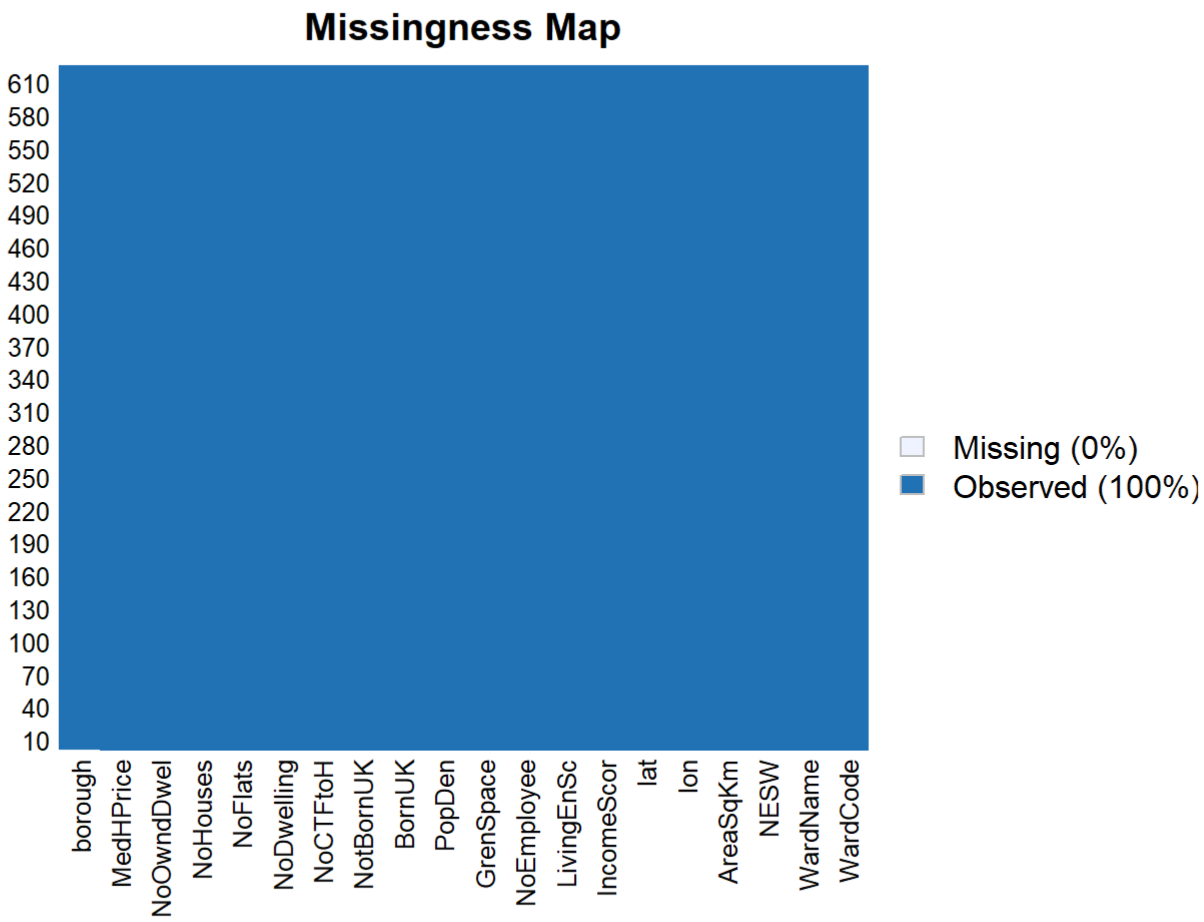
```
missmap(journey)
```




```
missmap(station)
```



```
missmap(census)
```



Observation: There were no misisng value found in the three datasets.

Hypothesis

Below the Hypotheses formulated:

- **H1** - Higher bike usage in the weekday compared to the weekend.
- **H2** - Higher bike usage during peak hour at 8 , 17 and 18 hours.
- **H3** - Higher bike usage in richer area.
- **H4** - Higher bike usage in area with higher employment rate.
- **H5** - Higher bike usage in densely green space.
- **H6** - Higher bike usage for longer journey.
- **H7** - Higher bike usage from station with more bike capacity.

All the above-mentioned hypotheses are falsifiable by validation of the following metrics via our data.

Metrics

Below are the metrics used to validate the hypotheses:

- **IsWeekend** - Lower demand for in weekend will link to **H1**.
- **Peak hour** - Start_Hour as the time metric which will represent the peak hour at 8 , 17 and 18 link to **H2**.
- **IncomeScor** - Income score from census data which is inversely proportionate wealthy link to **H3**.
- **LivingEnSc** - The local environment quality which is inversely proportionate to wealthy link to **H3**.
- **RatioCTFtoH** - Ratio of properties in council tax band F-H which define as *RatioCTFtoH* = $\frac{NoCTFtoH}{NoDwelling}$ link to **H3**.

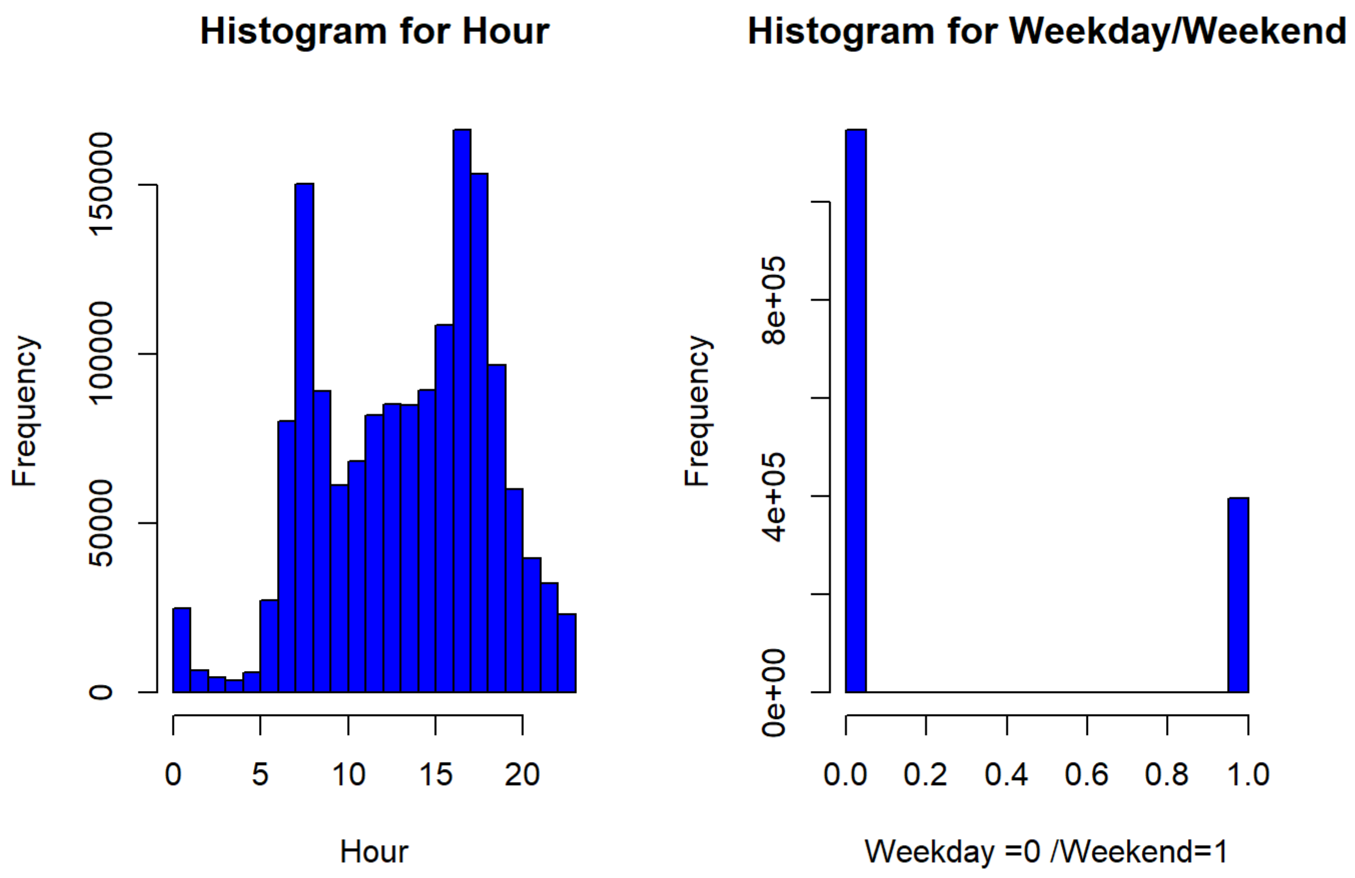
- **RatioEmployee** - Rate of people have work by define as *RatioEmployee* = $\frac{NoEmployee}{PopDen.AreaSqKm}$ is link to **H4**.
- **GrenSpace** - Higher percentage of green space with green zone link to **H5**.
- **JourneyMean** - Higher demand for longer journey link to **H6**.
- **CapacityDemand** - Higher demand for station with more bike link to **H7**.

Pre-processing of all the datasets by add and remove columns to adjust the structures of datasets in order to facilititate the join to to build metrics.

Firstly, to explore and understand the pattern of original datasets by following the stage of processing 'journey' dataset.Journey dataset is preprocess with created new column that date is completed in yy/mm/dd format and another column of boolean variable if its weekend. Later this two newly created variables are visualize in histogram.

```
##Add digits to column year in journey data.
journey$Start_Year <- as.numeric(sub("", "20", journey$Start_Year))

##Paste column together to creating new data with complete date in yy/mm/dd
journey[, Date := paste(Start_Year, Start_Month, Start_Date, sep = "-")]

##Create new column weekday to identify weekday = 1
library(chron)
journey[, isWeekend := ifelse(is.weekend(Date), 1,0)]
```

```
par(mfrow=c(1,2))  #fill by rows: Row, Cols

hist(journey$Start_Hour, main = "Histogram for Hour", xlab="Hour", col = "blue")

hist(journey$isWeekend, main = "Histogram for Weekday/Weekend", xlab="Weekday =0 /Weekend=1", col = "blue")
```



Observation: 1) By hourly higher demand of bike usage in hour 8, 17 and 18 2) By days higher demand in weekday.

Then to process the data of hour to identify peak hour and create two column for count of bike rental by Station ID, Hour and day and variables of journey mean. A new dataset named journey2 created with selected features. Library package of plyr is used for thesee data manipulation step.

```
##Create new column to identify peak hour in jouney dataset
journey[, Peak_hour := ifelse(Start_Hour %in% c(8,17,18), 1,0)]

## to count the bike rent by hour and day
library(plyr)
journey[, BikeRate:= .(.N), by = .(Start_Station_ID, Date, Start_Hour)]
journey[, JourneyMean:= sum(Journey_Duration)/(.N), by = .(Start_Station_ID)]
##Define metrics for journey dataset.
journey2 = journey[,.(Start_Station_ID, Start_Hour, isWeekend, Peak_hour, BikeRate, JourneyMe
an)]
str(journey2)
```

```
## Classes 'data.table' and 'data.frame':   1542844 obs. of  6 variables:
##  $ Start_Station_ID: int  251 550 212 163 36 589 478 478 153 396 ...
##  $ Start_Hour      : int  17 14 16 12 19 14 17 17 13 15 ...
##  $ isWeekend       : num  0 0 0 0 0 0 1 1 0 0 ...
##  $ Peak_hour       : num  1 0 0 0 0 0 1 1 0 0 ...
##  $ BikeRate        : int  31 1 2 4 2 3 5 5 8 4 ...
##  $ JourneyMean     : num  1082 841 1097 1124 1498 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Now to process ‘census’ dataset for new metrics of Ratio of Employee and Ratio CTF to H.

```
#create metric for census data for No of employee ratio and CTFtoH
census[, NoEmployee_Ratio := NoEmployee/(AreaSqKm*PopDen)]
census[, NoCTFtoH_Ratio := NoCTFtoH/NoDwelling]
str(census)
```

```
## Classes 'data.table' and 'data.frame':   625 obs. of  22 variables:
##  $ WardCode        : chr  "E05000026" "E05000027" "E05000028" "E05000029" ...
##  $ WardName        : chr  "Abbey" "Alibon" "Becontree" "Chadwell Heath" ...
##  $ borough         : chr  "Barking and Dagenham" "Barking and Dagenham" "Barking and Dagen
ham" "Barking and Dagenham" ...
##  $ NESW            : chr  "East" "East" "East" "East" ...
##  $ AreaSqKm        : num  1.3 1.4 1.3 3.4 3.5 1.4 1.1 1.3 2 1.6 ...
##  $ lon             : num  0.0779 0.1483 0.119 0.14 0.1736 ...
##  $ lat             : num  51.5 51.5 51.6 51.6 51.6 ...
##  $ IncomeScor      : num  0.27 0.28 0.25 0.27 0.19 0.27 0.36 0.27 0.31 0.17 ...
##  $ LivingEnSc      : num  42.8 28 31.6 34.8 21.2 ...
##  $ NoEmployee      : int  7900 800 1100 1700 4000 1000 2800 1300 2500 1600 ...
##  $ GrenSpace       : num  19.6 22.4 3 56.4 51.1 18.1 20.3 17.1 38.4 30.3 ...
##  $ PopDen          : num  9885 7464 8923 2971 3014 ...
##  $ BornUK          : int  5459 7824 8075 7539 8514 7880 6447 8244 8183 7660 ...
##  $ NotBornUK       : int  7327 2561 3470 2482 1992 3744 6005 3023 2603 3818 ...
##  $ NoCTFtoH        : num  0.1 0.1 0.1 0.4 0.5 0 0.1 0.1 0 7.7 ...
##  $ NoDwelling      : int  4733 4045 4378 4050 3976 4321 4662 4293 4409 3787 ...
##  $ NoFlats         : int  3153 574 837 1400 742 933 3368 657 1606 852 ...
##  $ NoHouses        : int  1600 3471 3541 2662 3235 3388 1343 3639 2812 2936 ...
##  $ NoOwndDwel      : int  1545 1849 2093 2148 2646 1913 1233 1938 1832 2618 ...
##  $ MedHPrice       : int  177000 160000 170000 195000 191750 167250 145000 155000 155000 2
50000 ...
##  $ NoEmployee_Ratio: num  0.6148 0.0766 0.0948 0.1683 0.3791 ...
##  $ NoCTFtoH_Ratio  : num  2.11e-05 2.47e-05 2.28e-05 9.88e-05 1.26e-04 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

To merge the station dataset to the census dataset with common field of spatial coordinates longitude and latitude by defining the closest distance and creating a new dataset name MergedStationandCensus_data with selected features.Then follow by a quick exploration of new created dataset.

Library sp package is used as this package provide classes and methods to create points, lines and grids by standardise the spatial data and better interoperability between the to datasets' point of cooordinates.

```r
library(sp)
##input list to spatial points dataframe
coordinates(census) <- c("lon", "lat")
coordinates(station)<- c("Longitude", "Latitude")

closestSiteCoordinates <- vector(mode = "numeric", length =nrow(station))
minDistCoordinates <- vector(mode = "numeric", length = nrow(station))

#Define these vectors and used in loop
for (i in 1 : nrow(station))
  {
        distCoordinates <- spDistsN1(census, station[i,], longlat = TRUE)
        minDistCoordinates[i] <- min(distCoordinates)
        closestSiteCoordinates[i] <- which.min(distCoordinates)
}

IncomeScor <- as.numeric(census[closestSiteCoordinates, ]$IncomeScor)
LivingEnSc <- as.numeric(census[closestSiteCoordinates, ]$LivingEnSc)
GrenSpace <- as.numeric(census[closestSiteCoordinates, ]$GrenSpace)
PopDen <- as.numeric(census[closestSiteCoordinates, ]$PopDen)
NoEmployee_ratio <- as.numeric(census[closestSiteCoordinates,]$NoEmployee_Ratio)
NoCTFtoH_ratio <- as.numeric(census[closestSiteCoordinates,]$NoCTFtoH_Ratio)

MergedStationandCensus_data <- data.frame(station$Station_ID, station$Capacity,
                                          IncomeScor,LivingEnSc,GrenSpace,PopDen,NoEmployee_r
atio,NoCTFtoH_ratio)

names(MergedStationandCensus_data) <- c("Start_Station_ID", "Capacity", "Income_score", "Livi
ngEnSc", "Gren_space", "Pop_Den", "No_Employee_ratio", "No_CTFtoH_ratio")

head(MergedStationandCensus_data)
```

	Start_Station_ID <int>	Capacity <int>	Income_score <dbl>	LivingEnSc <dbl>	Gren_spa… <dbl>	Pop_… <dbl>	No_Employe
1	1	19	0.21	50.95	9.3	12777.8	3.8
2	2	37	0.08	50.92	5.9	16583.3	0.6
3	3	32	0.24	44.07	13.5	13272.7	4.3
4	4	23	0.18	53.14	13.5	19583.3	1.3
5	5	27	0.07	43.77	8.5	14583.3	0.8
6	6	18	0.05	53.64	6.1	10350.0	5.8

6 rows | 1-8 of 9 columns

```r
str(MergedStationandCensus_data)
```

```
## 'data.frame':    773 obs. of  8 variables:
##  $ Start_Station_ID : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Capacity         : int  19 37 32 23 27 18 16 18 19 18 ...
##  $ Income_score     : num  0.21 0.08 0.24 0.18 0.07 0.05 0.1 0.36 0.16 0.16 ...
##  $ LivingEnSc       : num  51 50.9 44.1 53.1 43.8 ...
##  $ Gren_space       : num  9.3 5.9 13.5 13.5 8.5 6.1 62.2 7.6 15.3 15.3 ...
##  $ Pop_Den          : num  12778 16583 13273 19583 14583 ...
##  $ No_Employee_ratio: num  3.817 0.693 4.363 1.362 0.834 ...
##  $ No_CTFtoH_ratio  : num  0.00424 0.01283 0.00277 0.00265 0.0096 ...
```

Finally to join the 2 datasets(journey2 and MergedStationandCensus_data) via common field of Start_Station_ID to create the final dataset named bikedata.

```
#To merge the datasets
bikedata = merge(journey2, MergedStationandCensus_data, by="Start_Station_ID")
#create another variable in bikedate
bikedata[, Capacitydemand:= (.N)/Capacity, by=.(Start_Station_ID)]
str(bikedata)
```

```
## Classes 'data.table' and 'data.frame':   1530240 obs. of  14 variables:
##  $ Start_Station_ID : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Start_Hour       : int  12 7 6 6 6 9 8 8 19 19 ...
##  $ isWeekend        : num  1 0 0 0 0 0 0 0 1 0 ...
##  $ Peak_hour        : num  0 0 0 0 0 0 1 1 0 0 ...
##  $ BikeRate         : int  2 4 1 1 4 7 8 10 1 3 ...
##  $ JourneyMean      : num  959 959 959 959 959 ...
##  $ Capacity         : int  19 19 19 19 19 19 19 19 19 19 ...
##  $ Income_score     : num  0.21 0.21 0.21 0.21 0.21 0.21 0.21 0.21 0.21 0.21 ...
##  $ LivingEnSc       : num  51 51 51 51 51 ...
##  $ Gren_space       : num  9.3 9.3 9.3 9.3 9.3 9.3 9.3 9.3 9.3 9.3 ...
##  $ Pop_Den          : num  12778 12778 12778 12778 12778 ...
##  $ No_Employee_ratio: num  3.82 3.82 3.82 3.82 3.82 ...
##  $ No_CTFtoH_ratio  : num  0.00424 0.00424 0.00424 0.00424 0.00424 ...
##  $ Capacitydemand   : num  66.8 66.8 66.8 66.8 66.8 ...
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "sorted")= chr "Start_Station_ID"
```

Observation: Independent variable : Bike Rate.Count of bike rented in hourly of a day. Dependent variables: The remaining 13 variables except Bike rate as shown as output of Bikedata above.

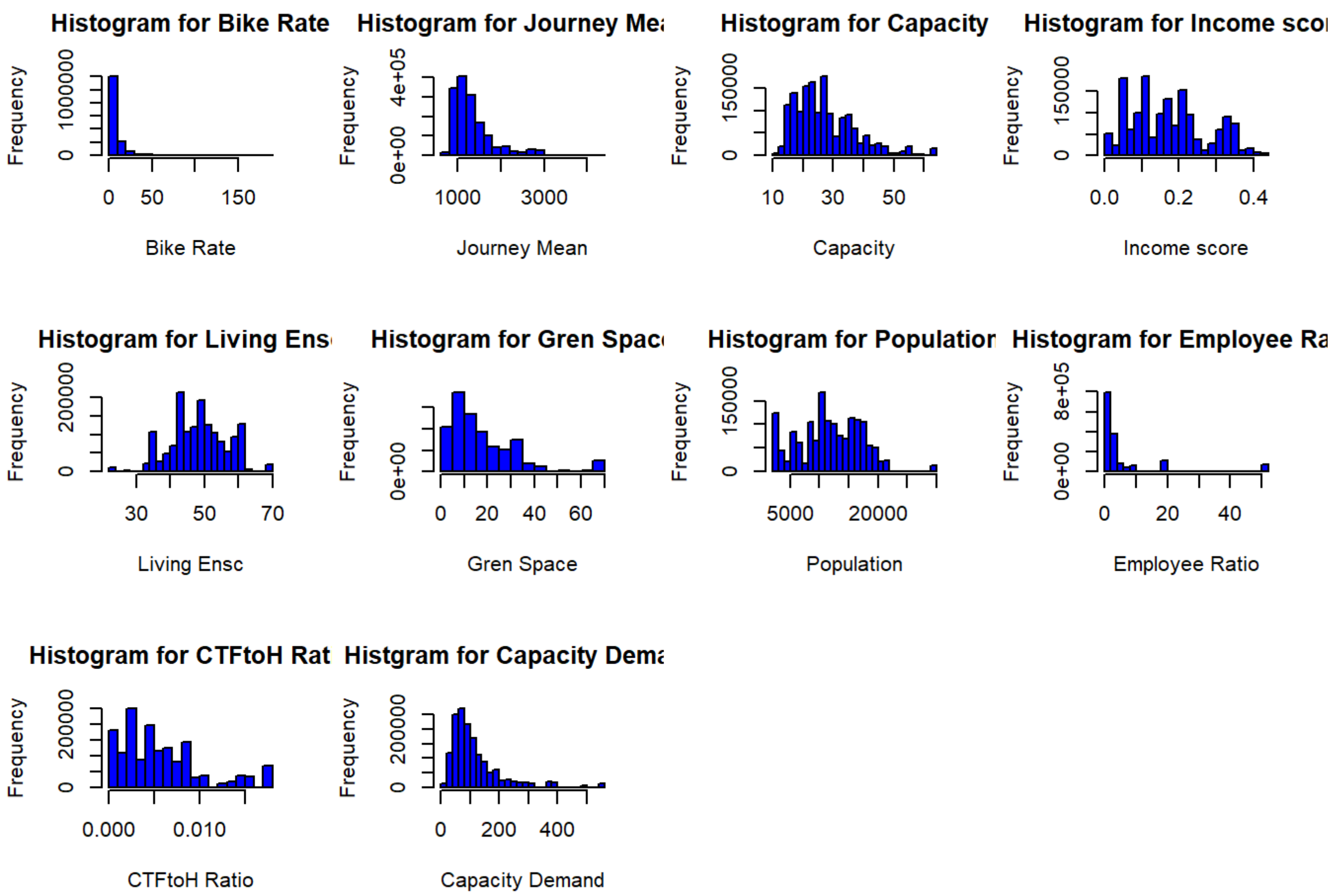
To view the summary of final data “bikedata”.

```
summary(bikedata)
```

```
##  Start_Station_ID   Start_Hour      isWeekend        Peak_hour
##  Min.   :  1.0    Min.   : 0.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:163.0    1st Qu.: 9.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :333.0    Median :14.00   Median :0.0000   Median :0.0000
##  Mean   :366.8    Mean   :13.76   Mean   :0.2562   Mean   :0.3047
##  3rd Qu.:570.0    3rd Qu.:18.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :826.0    Max.   :23.00   Max.   :1.0000   Max.   :1.0000
##     BikeRate        JourneyMean        Capacity      Income_score
##  Min.   :  1.000   Min.   : 620.8   Min.   :10.00   Min.   :0.0100
##  1st Qu.:  3.000   1st Qu.:1012.3   1st Qu.:21.00   1st Qu.:0.0900
##  Median :  5.000   Median :1198.3   Median :26.00   Median :0.1700
##  Mean   :  8.576   Mean   :1324.8   Mean   :27.88   Mean   :0.1766
##  3rd Qu.:  9.000   3rd Qu.:1471.9   3rd Qu.:34.00   3rd Qu.:0.2400
##  Max.   :182.000   Max.   :4310.6   Max.   :64.00   Max.   :0.4400
##    LivingEnSc      Gren_space       Pop_Den      No_Employee_ratio
##  Min.   :22.05   Min.   : 0.00   Min.   : 2312   Min.   : 0.1321
##  1st Qu.:43.29   1st Qu.: 7.50   1st Qu.: 8306   1st Qu.: 0.5446
##  Median :48.34   Median :13.50   Median :11958   Median : 1.4114
##  Mean   :48.36   Mean   :17.61   Mean   :11762   Mean   : 5.4905
##  3rd Qu.:53.64   3rd Qu.:25.00   3rd Qu.:16000   3rd Qu.: 3.8660
##  Max.   :68.06   Max.   :69.10   Max.   :29375   Max.   :50.5540
##  No_CTFtoH_ratio     Capacitydemand
##  Min.   :2.661e-05   Min.   :  4.513
##  1st Qu.:2.491e-03   1st Qu.: 59.765
##  Median :4.613e-03   Median : 90.488
##  Mean   :5.683e-03   Mean   :115.675
##  3rd Qu.:8.481e-03   3rd Qu.:138.125
##  Max.   :1.794e-02   Max.   :544.286
```

To understand the pattern of bikedata , hitogram is plotted.

```
par(mfrow=c(3,4))  #fill by rows: Row, Cols

hist(bikedata$BikeRate, main = "Histogram for Bike Rate", xlab="Bike Rate", col = "blue")
hist(bikedata$JourneyMean, main = "Histogram for Journey Mean", xlab="Journey Mean", col = "b
lue")
hist(bikedata$Capacity, main = "Histogram for Capacity", xlab="Capacity", col = "blue")
hist(bikedata$Income_score, main = "Histogram for Income score", xlab="Income score", col =
"blue")
hist(bikedata$LivingEnSc, main = "Histogram for Living Ensc", xlab="Living Ensc", col = "blu
e")
hist(bikedata$Gren_space, main = "Histogram for Gren Space", xlab="Gren Space", col = "blue")
hist(bikedata$Pop_Den, main = "Histogram for Population", xlab="Population", col = "blue")
hist(bikedata$No_Employee_ratio, main = "Histogram for Employee Ratio", xlab="Employee Ratio"
, col = "blue")
hist(bikedata$No_CTFtoH_ratio, main = "Histogram for CTFtoH Ratio", xlab="CTFtoH Ratio", col
 = "blue")
hist(bikedata$Capacitydemand, main = "Histgram for Capacity Demand", xlab = "Capacity Demand"
, col = "blue")
```



From the summary of histogram, there are variables not normally distributed like Bike rate,Journey Mean, GrenSpace,Employee ratio, CTFtoH ratio and capacity demand.

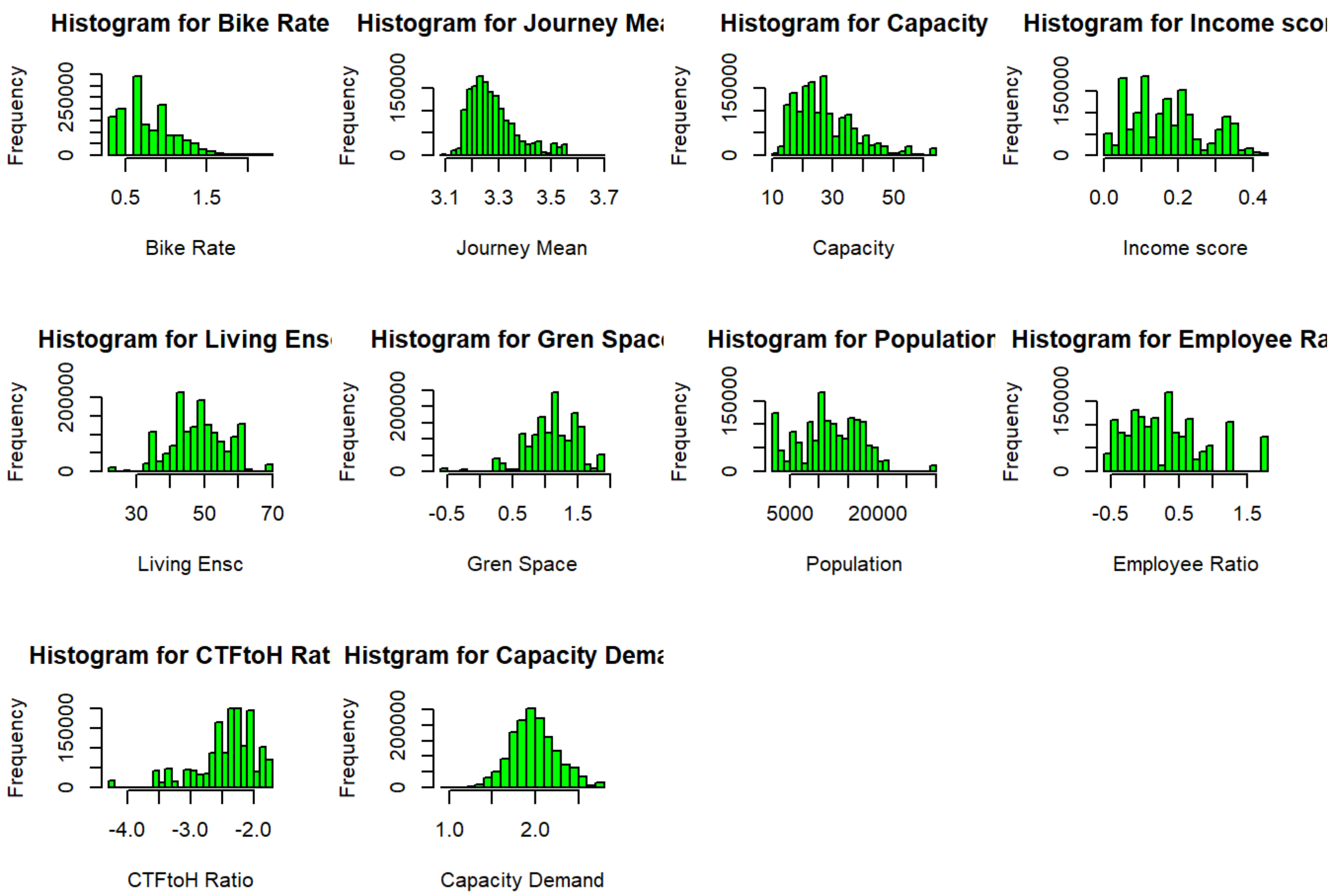
Thus, log transformation will be carry out on these data as logarathmic transformations is generally a better way to log transform data with values that range over several orders of magnitude and due to modeling techniques normally have difficult time with wide range of data.

```
bikedata$BikeRate = log10(bikedata$BikeRate + min(bikedata[BikeRate!=0]$BikeRate))
bikedata$JourneyMean = log10(bikedata$JourneyMean + min(bikedata[JourneyMean!=0]$JourneyMea
n))
bikedata$Gren_space = log10(bikedata$Gren_space + min(bikedata[Gren_space!=0]$Gren_space))
bikedata$No_Employee_ratio = log10(bikedata$No_Employee_ratio + min(bikedata[No_Employee_rati
o!=0]$No_Employee_ratio))
bikedata$No_CTFtoH_ratio = log10(bikedata$No_CTFtoH_ratio + min(bikedata[No_CTFtoH_ratio!=0]
$No_CTFtoH_ratio))
bikedata$Capacitydemand = log10(bikedata$Capacitydemand + min(bikedata[Capacitydemand!=0]$Cap
acitydemand))
summary(bikedata)
```

```
##  Start_Station_ID   Start_Hour      isWeekend        Peak_hour
##  Min.   :  1.0    Min.   : 0.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:163.0    1st Qu.: 9.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :333.0    Median :14.00   Median :0.0000   Median :0.0000
##  Mean   :366.8    Mean   :13.76   Mean   :0.2562   Mean   :0.3047
##  3rd Qu.:570.0    3rd Qu.:18.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :826.0    Max.   :23.00   Max.   :1.0000   Max.   :1.0000
##     BikeRate       JourneyMean      Capacity      Income_score
##  Min.   :0.3010   Min.   :3.094   Min.   :10.00   Min.   :0.0100
##  1st Qu.:0.6021   1st Qu.:3.213   1st Qu.:21.00   1st Qu.:0.0900
##  Median :0.7782   Median :3.260   Median :26.00   Median :0.1700
##  Mean   :0.8068   Mean   :3.279   Mean   :27.88   Mean   :0.1766
##  3rd Qu.:1.0000   3rd Qu.:3.321   3rd Qu.:34.00   3rd Qu.:0.2400
##  Max.   :2.2625   Max.   :3.693   Max.   :64.00   Max.   :0.4400
##    LivingEnSc      Gren_space         Pop_Den      No_Employee_ratio
##  Min.   :22.05   Min.   :-0.5229   Min.   : 2312   Min.   :-0.5780
##  1st Qu.:43.29   1st Qu.: 0.8921   1st Qu.: 8306   1st Qu.:-0.1696
##  Median :48.34   Median : 1.1399   Median :11958   Median : 0.1885
##  Mean   :48.36   Mean   : 1.1117   Mean   :11762   Mean   : 0.3017
##  3rd Qu.:53.64   3rd Qu.: 1.4031   3rd Qu.:16000   3rd Qu.: 0.6019
##  Max.   :68.06   Max.   : 1.8414   Max.   :29375   Max.   : 1.7049
##  No_CTFtoH_ratio  Capacitydemand
##  Min.   :-4.274   Min.   :0.9555
##  1st Qu.:-2.599   1st Qu.:1.8081
##  Median :-2.334   Median :1.9777
##  Mean   :-2.423   Mean   :1.9908
##  3rd Qu.:-2.070   3rd Qu.:2.1542
##  Max.   :-1.746   Max.   :2.7394
```

Review of the histograms upon log transformation

```
par(mfrow=c(3,4))  #fill by rows: Row, Cols

hist(bikedata$BikeRate, main = "Histogram for Bike Rate", xlab="Bike Rate", col = "green")
hist(bikedata$JourneyMean, main = "Histogram for Journey Mean", xlab="Journey Mean", col = "g
reen")
hist(bikedata$Capacity, main = "Histogram for Capacity", xlab="Capacity", col = "green")
hist(bikedata$Income_score, main = "Histogram for Income score", xlab="Income score", col =
"green")
hist(bikedata$LivingEnSc, main = "Histogram for Living Ensc", xlab="Living Ensc", col = "gree
n")
hist(bikedata$Gren_space, main = "Histogram for Gren Space", xlab="Gren Space", col = "green"
)
hist(bikedata$Pop_Den, main = "Histogram for Population", xlab="Population", col = "green")
hist(bikedata$No_Employee_ratio, main = "Histogram for Employee Ratio", xlab="Employee Ratio"
, col = "green")
hist(bikedata$No_CTFtoH_ratio, main = "Histogram for CTFtoH Ratio", xlab="CTFtoH Ratio", col
 = "green")
hist(bikedata$Capacitydemand, main = "Histgram for Capacity Demand", xlab = "Capacity Demand"
, col = "green")
```



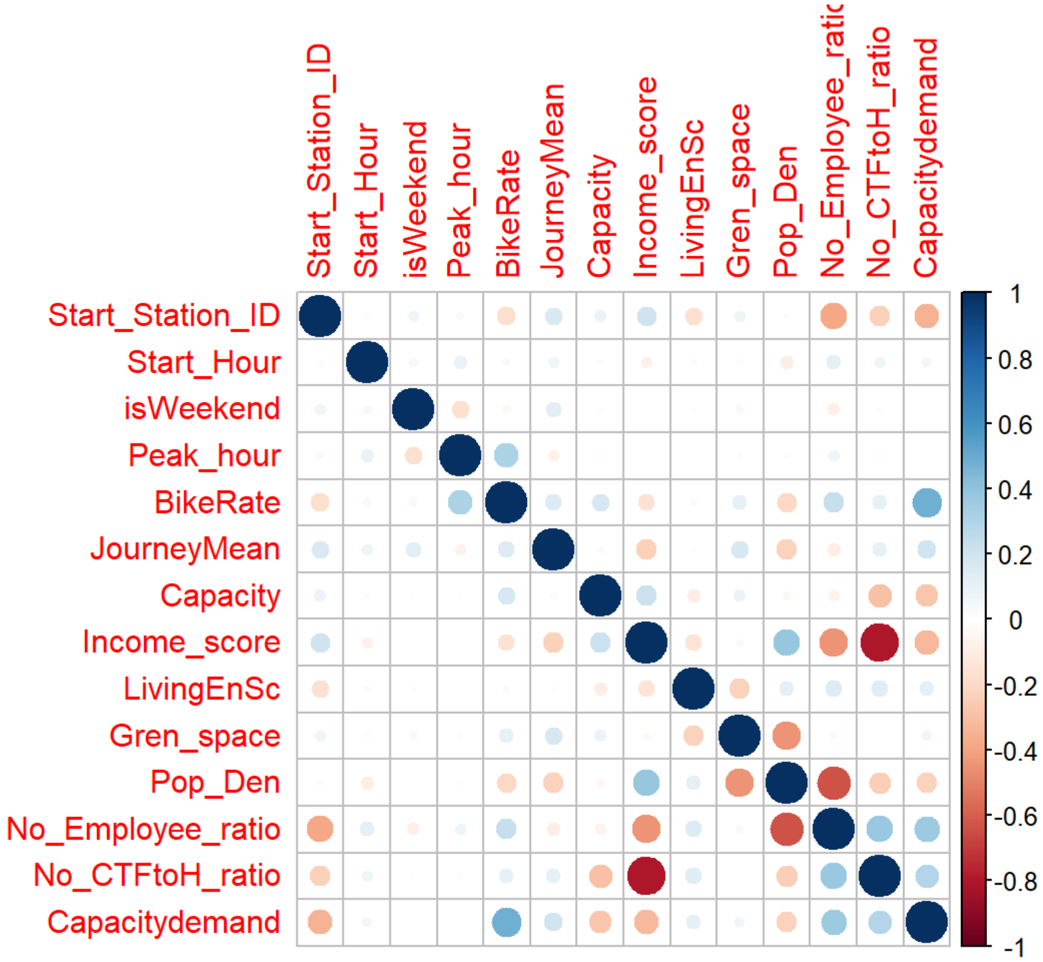
Observation: All the variables are better in distributions. 1) Capacity shown higher frequency in smaller capacity station. 2) Income score shown higer frequency in lower income score area. 3) Living Ensc shown higher frequency in higer score of living ensc. 4) Gren space shown higher frequency in greener area. 5) Population shown higher frequency in higer populated area. 6) Employee ratio shown higher frequency in lower employment rate area. 7) CTF to H ratio shown higer frequency in higher ratio. 8) Capacity demand shown higer frequency in higher capacity station.

To check the collinearity of all the variables as to validate the hypotheses. Corrplot package is used to graphically display the correlation matrix of the confidence interval of the variables.

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(cor(bikedata))
```

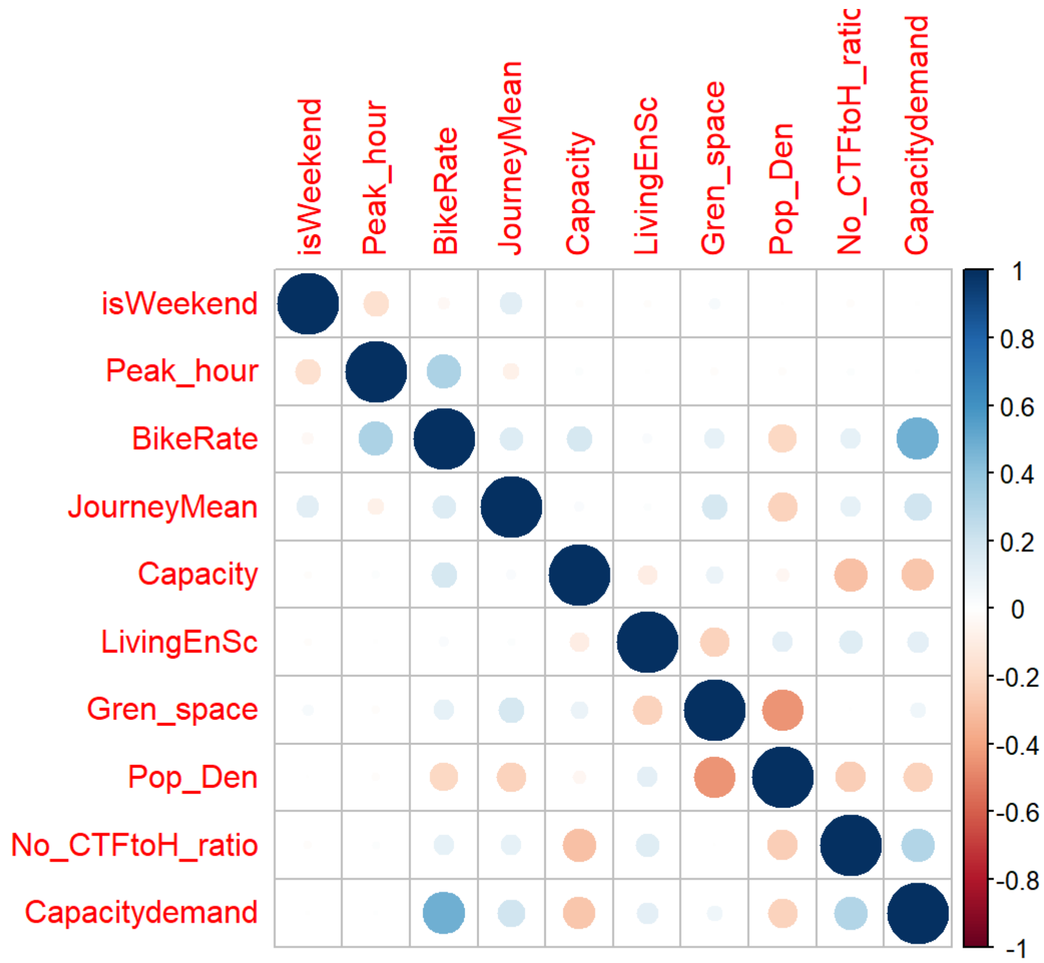


The variables of Start_Station_ID and Income_Score will be removed as they are the mostly correlated with the others. Then to recheck the multicollinearity.

```
bikedata$Start_Station_ID = NULL
bikedata$Start_Hour = NULL
bikedata$Income_score = NULL
bikedata$BornUK_ratio = NULL
```

```
## Warning in set(x, j = name, value = value): Column 'BornUK_ratio' does not exist
## to remove
```

```
bikedata$No_Employee_ratio = NULL
corrplot(cor(bikedata))
```



Data will be standardise before further validation and model training.

```
bikedata_std = as.data.table(scale(bikedata))
summary(bikedata_std)
```

```
##    isWeekend         Peak_hour         BikeRate          JourneyMean
##  Min.   :-0.5869   Min.   :-0.662   Min.   :-1.46429   Min.   :-2.0456
##  1st Qu.:-0.5869   1st Qu.:-0.662   1st Qu.:-0.59281   1st Qu.:-0.7287
##  Median :-0.5869   Median :-0.662   Median :-0.08303   Median :-0.2104
##  Mean   : 0.0000   Mean   : 0.000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 1.7040   3rd Qu.: 1.511   3rd Qu.: 0.55922   3rd Qu.: 0.4630
##  Max.   : 1.7040   Max.   : 1.511   Max.   : 4.21399   Max.   : 4.5822
##     Capacity         LivingEnSc          Gren_space          Pop_Den
##  Min.   :-1.7874   Min.   :-3.24909   Min.   :-4.27181   Min.   :-1.75342
##  1st Qu.:-0.6875   1st Qu.:-0.62599   1st Qu.:-0.57397   1st Qu.:-0.64134
##  Median :-0.1876   Median :-0.00232   Median : 0.07358   Median : 0.03646
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.6123   3rd Qu.: 0.65222   3rd Qu.: 0.76153   3rd Qu.: 0.78644
##  Max.   : 3.6120   Max.   : 2.43307   Max.   : 1.90681   Max.   : 3.26832
##  No_CTFtoH_ratio   Capacitydemand
##  Min.   :-3.9194   Min.   :-3.81325
##  1st Qu.:-0.3728   1st Qu.:-0.67298
##  Median : 0.1893   Median :-0.04806
##  Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.7469   3rd Qu.: 0.60207
##  Max.   : 1.4342   Max.   : 2.75743
```

Algorithms

Train-test split will be use to train linear regression model to predict the bike count hourly and in day.In order to obtain the best combination of parameters of regression model, the standardised bike data is further divided into training 75% and 25% of test dataset with is randomly selected. The 75% of train dataset is used to train the linear regression model whereas the remaining 25% will later being used to validate the trained model obtained.

```
set.seed(0)
trainIdx = sample(1:nrow(bikedata_std), 0.75*nrow(bikedata_std))
train = bikedata_std[trainIdx]
test = bikedata_std[-trainIdx]

lr = lm(BikeRate ~ ., data=train)
train_preds = predict(lr, train)
test_preds = predict(lr, test)

print( paste("R2 on train:", cor(train_preds, train$BikeRate)^2))
```

```
## [1] "R2 on train: 0.437132224783462"
```

```
print( paste("R2 on test:", cor(test_preds, test$BikeRate)^2))
```

```
## [1] "R2 on test: 0.436596869508046"
```

Observation: Above results shown model is stable as the two value are similar R-square value of 0.437 and 0.437 and do not overfitting.

Data Understanding

THe model that attempts to predict count based off the following features. Below beta coefficients will allow us better understand the model.

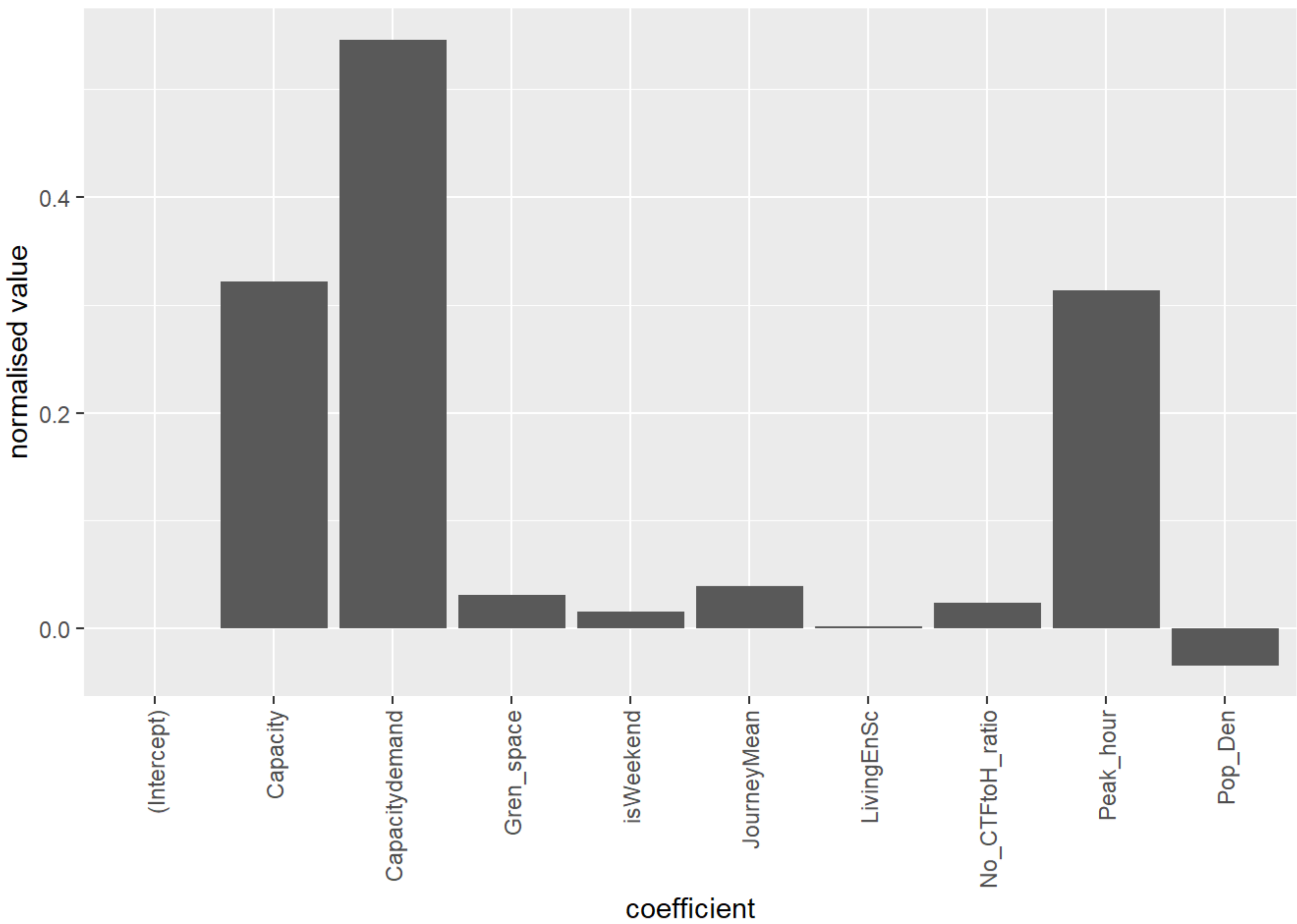
```
lr =lm(BikeRate ~ ., data = bikedata_std)
summary(lr)
```

```
##
## Call:
## lm(formula = BikeRate ~ ., data = bikedata_std)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7090 -0.5051  0.0086  0.5018  3.7221
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.492e-13  6.066e-04   0.000   1.0000
## isWeekend        1.543e-02  6.205e-04  24.868   <2e-16 ***
## Peak_hour        3.140e-01  6.167e-04 509.152   <2e-16 ***
## JourneyMean      3.951e-02  6.408e-04  61.655   <2e-16 ***
## Capacity         3.219e-01  6.595e-04 488.005   <2e-16 ***
## LivingEnSc       1.571e-03  6.344e-04   2.477   0.0133 *
## Gren_space       3.055e-02  6.985e-04  43.739   <2e-16 ***
## Pop_Den         -3.431e-02  7.259e-04 -47.271   <2e-16 ***
## No_CTFtoH_ratio  2.391e-02  6.783e-04  35.255   <2e-16 ***
## Capacitydemand   5.460e-01  6.732e-04 811.125   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7503 on 1530230 degrees of freedom
## Multiple R-squared:  0.437,  Adjusted R-squared:  0.437
## F-statistic: 1.32e+05 on 9 and 1530230 DF,  p-value: < 2.2e-16
```

Model interpretation: The linear model build’s R square value 0.437 is significant as p-value less than 5%.

To visualise the linear model built. ggplot2 package is used to map the variables graphically.

```
library(ggplot2)

ggplot(, aes(x = names(lr$coefficients), y=lr$coefficients)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  xlab("coefficient") +
  ylab("normalised value")
```



Findings

From the above visualisation may conclude and validate that:

- **H1** - Higher bike usage in the weekday.-True
- **H2** - Higher bike usage during peak hour.-True
- **H3** - Higher bike usage in richer area.-True
- **H4** - Higher bike usage in area with higher employment rate.-False, no correlation
- **H5** - Higher bike usage in densely green space.-True
- **H6** - Higher bike usage for longer journey. - True
- **H7** - Higher bike usage from station with more bike capacity. - True

From the R-square value, the linear regression model shown a decent prediction power which explained 44% of the variation of the outcome variable.Moreover all the p-value of the variables are significant.

Limitations

- Multicollinearity does not allow us to use all the metrics in the model.
- Almost 60% of the variation is not explained by this model indicate that must have some other factor has not been considered.
- Due to limited data as model built will be more accurate if humidity, wind, temperature and seasonal data are availabe.