CST4070 - Applied Data Analytics - Tools, Practical Big Data Handling, Cloud Distribution Summative assessment − Component 2 Individual report - Dragomir Nedev M00724882

##Problem deffinition Three datasets are available: bike_journeys, bike_stations and LondonCensus. Spatial granilarity: each bike station. Temporal granularity: one hour time slot. Goal: predicting the total number of bikes rented in each bike station with the temporal granularity of one hour time slot.

##Preprocecssing Importing the datasets

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.6.2
```

```
bike_journeys = fread('bike_journeys.csv')
bike_stations = fread('bike_stations.csv')
census = fread('London_census.csv')
```

Exploring the datasets

```
head(bike_journeys)
```

```
##    Journey_Duration Journey_ID End_Date End_Month End_Year End_Hour End_Minute
## 1:             2040        953       19         9       17       18          0
## 2:             1800      12581       19         9       17       15         21
## 3:             1140       1159       15         9       17       17          1
## 4:              420       2375       14         9       17       12         16
## 5:             1200      14659       13         9       17       19         33
## 6:             1320       2351       14         9       17       14         53
##    End_Station_ID Start_Date Start_Month Start_Year Start_Hour Start_Minute
## 1:            478         19           9         17         17           26
## 2:            122         19           9         17         14           51
## 3:            639         15           9         17         16           42
## 4:            755         14           9         17         12            9
## 5:            605         13           9         17         19           13
## 6:            514         14           9         17         14           31
##    Start_Station_ID
## 1:              251
## 2:              550
## 3:              212
## 4:              163
## 5:               36
## 6:              589
```

```
head(bike_stations)
```

```
##    Station_ID Capacity Latitude Longitude                         Station_Name
## 1:          1       19 51.52916 -0.109970           River Street , Clerkenwell
## 2:          2       37 51.49961 -0.197574       Phillimore Gardens, Kensington
## 3:          3       32 51.52128 -0.084605 Christopher Street, Liverpool Street
## 4:          4       23 51.53006 -0.120973      St. Chad's Street, King's Cross
## 5:          5       27 51.49313 -0.156876        Sedding Street, Sloane Square
## 6:          6       18 51.51812 -0.144228       Broadcasting House, Marylebone
```

```
head(census)
```

```
##     WardCode       WardName              borough NESW AreaSqKm      lon
## 1: E05000026          Abbey Barking and Dagenham East      1.3 0.077935
## 2: E05000027         Alibon Barking and Dagenham East      1.4 0.148270
## 3: E05000028      Becontree Barking and Dagenham East      1.3 0.118957
## 4: E05000029 Chadwell Heath Barking and Dagenham East      3.4 0.139985
## 5: E05000030      Eastbrook Barking and Dagenham East      3.5 0.173581
## 6: E05000031       Eastbury Barking and Dagenham East      1.4 0.105683
##         lat IncomeScor LivingEnSc NoEmployee GrenSpace PopDen BornUK NotBornUK
## 1: 51.53971       0.27      42.76       7900      19.6 9884.6   5459      7327
## 2: 51.54559       0.28      27.96        800      22.4 7464.3   7824      2561
## 3: 51.55453       0.25      31.59       1100       3.0 8923.1   8075      3470
## 4: 51.58475       0.27      34.78       1700      56.4 2970.6   7539      2482
## 5: 51.55365       0.19      21.25       4000      51.1 3014.3   8514      1992
## 6: 51.53590       0.27      31.16       1000      18.1 8357.1   7880      3744
##    NoCTFtoH NoDwelling NoFlats NoHouses NoOwndDwel MedHPrice
## 1:      0.1       4733    3153     1600       1545    177000
## 2:      0.1       4045     574     3471       1849    160000
## 3:      0.1       4378     837     3541       2093    170000
## 4:      0.4       4050    1400     2662       2148    195000
## 5:      0.5       3976     742     3235       2646    191750
## 6:      0.0       4321     933     3388       1913    167250
```

Importing libraries which will check and plot heatmap of missing values

```
library(Rcpp)
```

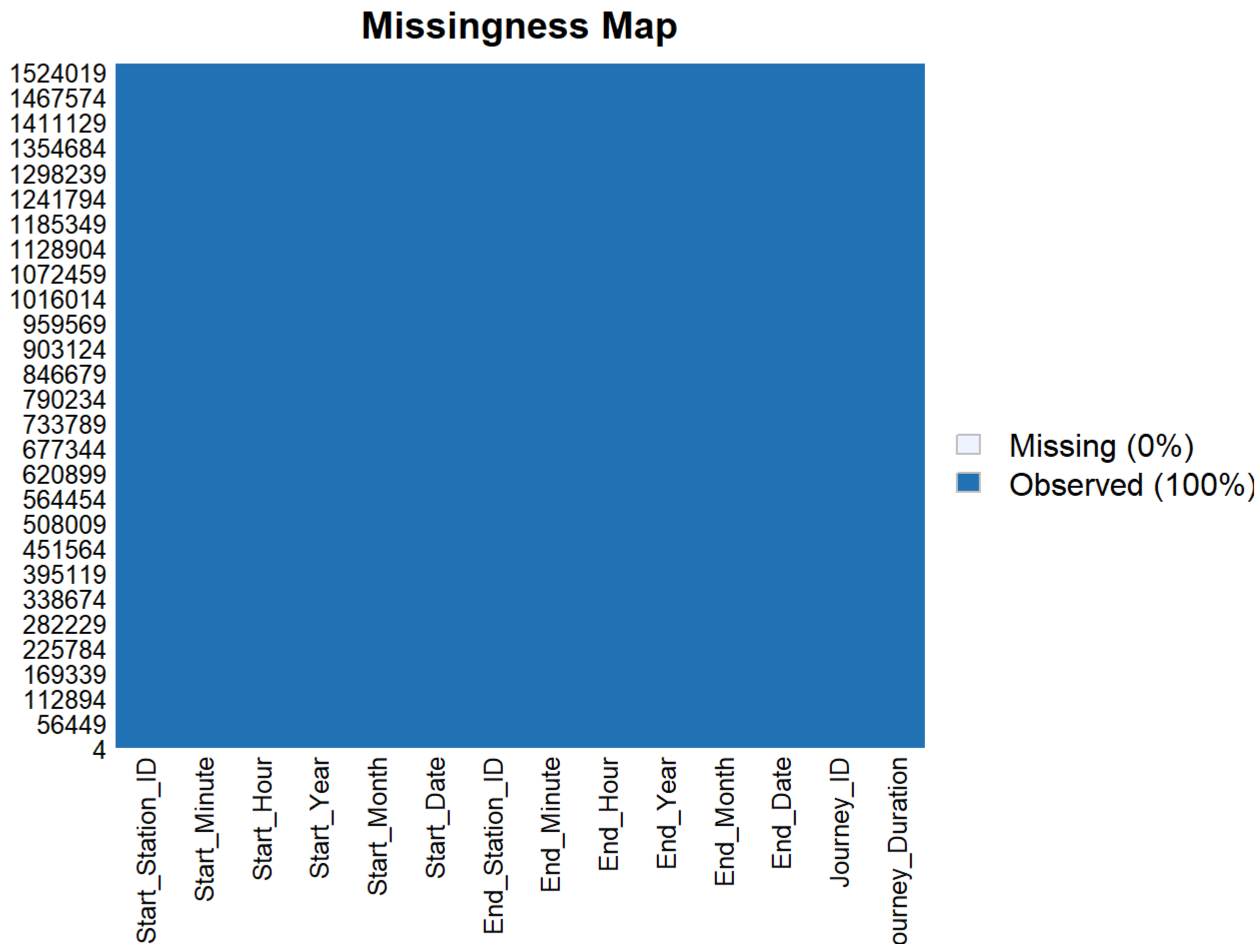
```
## Warning: package 'Rcpp' was built under R version 3.6.2
```

```
library(Amelia)
```

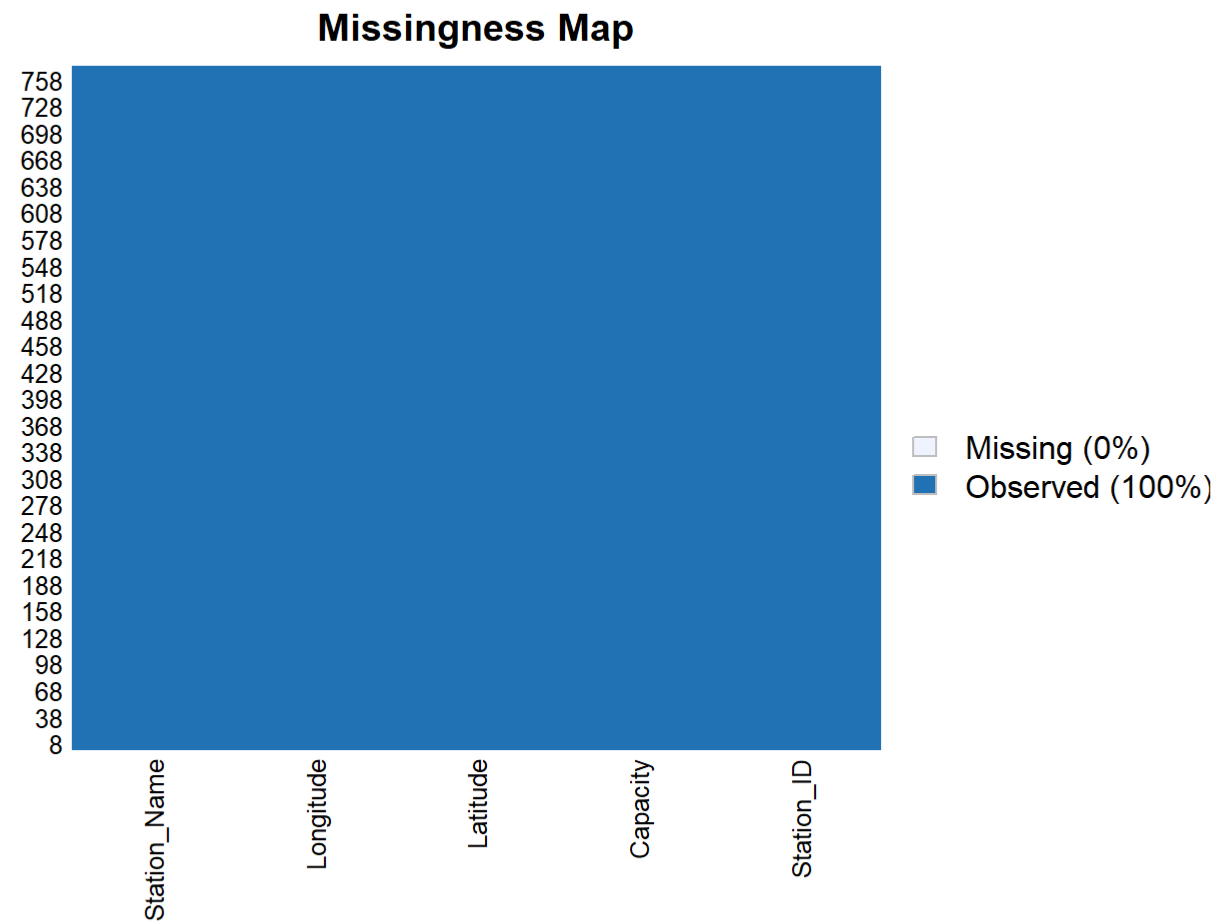
```
## Warning: package 'Amelia' was built under R version 3.6.2
```

```
## ##
## ## 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
## ##
```

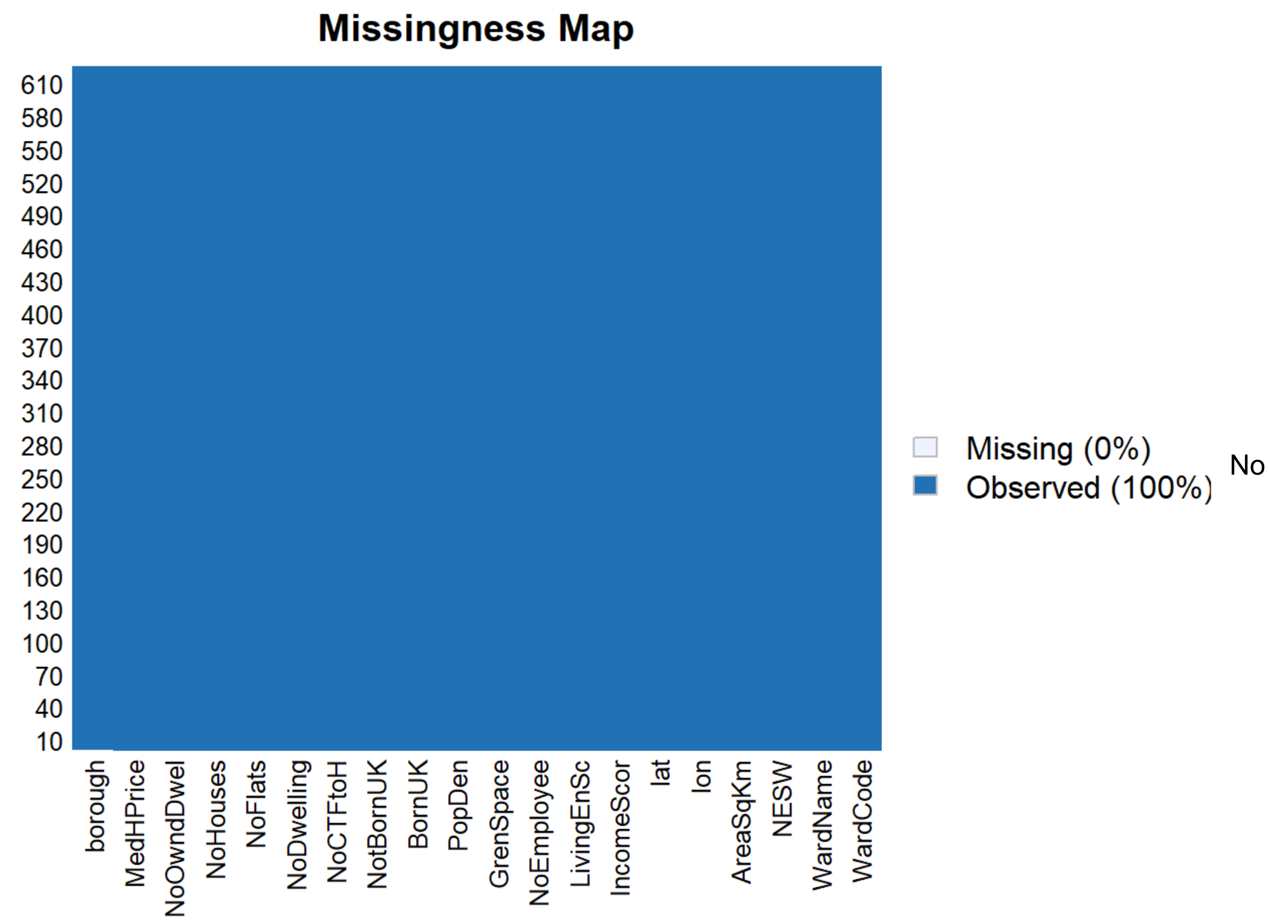
```
missmap(bike_journeys)
```




```
missmap(bike_stations)
```




```
missmap(census)
```




missing data, in the datasets, means there would not be NaN values.

Checking consistency between bike_journeys and bike_stations. We have to join this datasets based on Start_Station_ID and StationID so we need to check whether they contain the same values.

```
length(unique(bike_journeys$Start_Station_ID))
```

```
## [1] 779
```

```
length(unique(bike_journeys$End_Station_ID))
```

```
## [1] 779
```

```
length(unique(bike_stations$Station_ID))
```

```
## [1] 773
```

```
length(unique(intersect(bike_stations$Station_ID, bike_journeys$Start_Station_ID)))
```

```
## [1] 771
```

Bike_journeys dataset contains 779 unique stations (the same number for end stations and start stations). Bike_stations dataset contains 773 unique stations. Both datasets have 771 matching unique stations which means that we will exclude data for 8 stations.

##Hypotheses

H1. Bikes demand is higher durin peak hours. H2. Bikes demand have a daily trend. H3. Higher demand of bikes rented at stations which are close to central London. H4. Higher demand of bikes rented where is high employment rate. H5. Higher demand of bikes rented where is high population density. H6. Higher demand of bikes rented where is high percentage of green space. H7. Higher demand of bikes rented in deprived areas. H8. Higher demand of bikes rented in poor areas. H9. Higher demand of bikes rented where is high immigration rate. H10. Higher demand of bikes rented where is high flats rate. H11. Higher demand of bikes rented where is low number of owned properties rate.

##Metrics

- bike_rides. Number of rides would be our depandant variable that we need to predict
- Start_hour. Indicate the hour when the journey started. Linked to H1.
- Start_Day. Indicate the day when the journey started. Linked to H2.
- finalRatioEmployee. Ratio of people who are employed. NoEmployee over PopDen times AreaSqKm. Linked to H4.
- PopDen. Population divided by the ward area. Linked to H5.
- GrenSpace. Percentage of green space associated with the ward. Linked to H6.
- LivingEnSc. Quality of the local environment. The more deprived is an area, the higher the score. Linked to H7.
- IncomeScor. Proportion of the population experiencing deprivation relating to low income. Higher score means lower income and poorer areas. Linked to H8.
- MedHPrice. Median house price. The lower median means the poorer areas. Linked to H8.
- RatioCTFtoH. Ratio of properties in council tax band F-H (the highest median house price). The lower score means the poorer areas. Linked H8.
- RatioBornUK. Ratio of people who were born in the UK. It is defined as NotBornUK over BornUK plus NotBornUK. Linked to H9.
- FlatsRate. Ratio of flats. It is defined as NoFlats over NoHouses. Linked to H10.
- RatioOwndDwel. Ratio of owned properties in each ward. It is defined as NoOwndDwel over NoDwelling. Linked to H11.

##Data processing

Due to the fact that the cencus data holds the record of longitute and latitude of the ward and the bike_station dataset, contains the coordinates of the bike stations, we need to calculate the nearest distance.Importing library “geosphere” will help us calculate the distance between the locations from the two datasets

```
library(geosphere)
```

```
## Warning: package 'geosphere' was built under R version 3.6.2
```

```
distance <- distm(bike_stations[, 4:3], census[, 6:7])
```

```
distance_calc <- cbind(bike_stations, census[apply(distance, 1, which.min),])
View(distance_calc)
```

Renaming the column Start_Station_ID to match Station_ID, so we could berge the data

```
colnames(bike_journeys)[colnames(bike_journeys) == "Start_Station_ID"] <- "Station_ID"
```

After we are done the the transformations of the location we can merge the datasets

```
total <- merge(bike_journeys,distance_calc,by = "Station_ID")
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

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

Combining the different data fields into one

```
total$Journey_date <- as.Date(with(total, paste(Start_Year, Start_Month, Start_Date ,sep="-")),
"%y-%m-%d")
```

```
total2 <- total %>% group_by(Start_Hour, Station_ID, Journey_date) %>% summarise(bike_rides = n
())
```

```
View(total2)
```

```
total2 <- left_join(total, total2, by=c("Station_ID","Start_Hour", "Journey_date")) %>% rowwise
()
```

The data frame needs to be transformed into a datatable, before extracting the final dataset

```
setDT(total2)
```

The data needs to be transformed from the format:

<Journey_Duration, Journey_ID, End_Date, End_Month, End_Year, End_Hour, End_Minute, End_Station_ID, Start_Date, Start_Month, Start_Year, Start_Hour, Start_Minute, Start_Station_ID> <Station_ID, Capacity, Latitude, Longitude, Station_Name> <WardCode, WardName, Borough, NESW, AreaSqKm, lon, lat, IncomeScor, LivingEnSc, NoEmployee, GrenSpace, PopDen, BornUK, NotBornUK, NoCTFtoH, NoDwelling, NoFlats, NoHouses, NoWndDwel, MedHPrice>

Into the format:

<bike_rides, Station_ID,Start_Date, Start_Hour, MedHPrice, finalRatioEmployee, IncomeScor, LivingEnSc, GrenSpace, RatioBornUK, RatioCTFtoH, RatioOwndDwel, FlatsRate>

```
final = total2[, .(bike_rides, Station_ID, Start_Date,
                   Start_Hour,  MedHPrice,
                   finalRatioEmployee=NoEmployee/(PopDen*AreaSqKm), IncomeScor, LivingEnSc,
                   GrenSpace, RatioBornUK=BornUK/(BornUK+NotBornUK), RatioCTFtoH=NoCTFtoH/(NoDwe
lling),
                   RatioOwndDwel=NoOwndDwel/NoDwelling, FlatsRate=NoFlats/(NoFlats+NoHouses))]
str(final)
```

```
## Classes 'data.table' and 'data.frame':   1530240 obs. of  13 variables:
##  $ bike_rides        : int  2 4 1 1 4 7 8 10 1 3 ...
##  $ Station_ID        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Start_Date        : int  17 14 18 13 19 15 15 19 17 13 ...
##  $ Start_Hour        : int  12 7 6 6 6 9 8 8 19 19 ...
##  $ MedHPrice         : int  455000 455000 455000 455000 455000 455000 455000 455000 455000 45
5000 ...
##  $ finalRatioEmployee: num  3.82 3.82 3.82 3.82 3.82 ...
##  $ IncomeScor        : 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 ...
##  $ GrenSpace         : num  9.3 9.3 9.3 9.3 9.3 9.3 9.3 9.3 9.3 9.3 ...
##  $ RatioBornUK       : num  0.615 0.615 0.615 0.615 0.615 ...
##  $ RatioCTFtoH       : num  0.00424 0.00424 0.00424 0.00424 0.00424 ...
##  $ RatioOwndDwel     : num  0.276 0.276 0.276 0.276 0.276 ...
##  $ FlatsRate         : num  0.905 0.905 0.905 0.905 0.905 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Summerising the information from the final dataset

```
summary(final)
```

```
##    bike_rides       Station_ID      Start_Date     Start_Hour
##  Min.   :  1.000   Min.   :  1.0   Min.   : 1.0   Min.   : 0.00
##  1st Qu.:  3.000   1st Qu.:163.0   1st Qu.: 7.0   1st Qu.: 9.00
##  Median :  5.000   Median :333.0   Median :13.0   Median :14.00
##  Mean   :  8.576   Mean   :366.8   Mean   :13.8   Mean   :13.76
##  3rd Qu.:  9.000   3rd Qu.:570.0   3rd Qu.:19.0   3rd Qu.:18.00
##  Max.   :182.000   Max.   :826.0   Max.   :31.0   Max.   :23.00
##    MedHPrice       finalRatioEmployee   IncomeScor       LivingEnSc
##  Min.   : 188000   Min.   : 0.1321    Min.   :0.0100   Min.   :22.05
##  1st Qu.: 362500   1st Qu.: 0.5446    1st Qu.:0.0900   1st Qu.:43.29
##  Median : 455000   Median : 1.4114    Median :0.1700   Median :48.34
##  Mean   : 559274   Mean   : 5.4905    Mean   :0.1766   Mean   :48.36
##  3rd Qu.: 652500   3rd Qu.: 3.8660    3rd Qu.:0.2400   3rd Qu.:53.64
##  Max.   :1750000   Max.   :50.5540    Max.   :0.4400   Max.   :68.06
##    GrenSpace      RatioBornUK      RatioCTFtoH        RatioOwndDwel
##  Min.   : 0.00   Min.   :0.3543   Min.   :2.661e-05   Min.   :0.1380
##  1st Qu.: 7.50   1st Qu.:0.4785   1st Qu.:2.491e-03   1st Qu.:0.2167
##  Median :13.50   Median :0.5521   Median :4.613e-03   Median :0.2707
##  Mean   :17.61   Mean   :0.5333   Mean   :5.683e-03   Mean   :0.2803
##  3rd Qu.:25.00   3rd Qu.:0.5955   3rd Qu.:8.481e-03   3rd Qu.:0.3352
##  Max.   :69.10   Max.   :0.7112   Max.   :1.794e-02   Max.   :0.5476
##    FlatsRate
##  Min.   :0.5423
##  1st Qu.:0.8397
##  Median :0.8928
##  Mean   :0.8733
##  3rd Qu.:0.9480
##  Max.   :0.9794
```

In a few of the vairables it could be seen that they are not normally distributed, which indicates that they have to be transformed in to log value.

```
final$bike_rides = log10(final$bike_rides + min(final[bike_rides!=0]$bike_rides))
final$RatioBornUK = log10(final$RatioBornUK + min(final[RatioBornUK!=0]$RatioBornUK))
final$RatioCTFtoH = log10(final$RatioCTFtoH + min(final[RatioCTFtoH!=0]$RatioCTFtoH))
```

Standardising the data

```
mydata_std = as.data.table(scale(final) )
summary(mydata_std)
```

```
##    bike_rides        Station_ID       Start_Date        Start_Hour     
##  Min.   :-1.46429   Min.   :-1.5385   Min.   :-1.51647   Min.   :-2.80363  
##  1st Qu.:-0.59281   1st Qu.:-0.8571   1st Qu.:-0.80589   1st Qu.:-0.96936  
##  Median :-0.08303   Median :-0.1421   Median :-0.09531   Median : 0.04969  
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000  
##  3rd Qu.: 0.55922   3rd Qu.: 0.8548   3rd Qu.: 0.61528   3rd Qu.: 0.86492  
##  Max.   : 4.21399   Max.   : 1.9315   Max.   : 2.03645   Max.   : 1.88396  
##    MedHPrice       finalRatioEmployee   IncomeScor          LivingEnSc      
##  Min.   :-1.1803   Min.   :-0.4830    Min.   :-1.64916   Min.   :-3.24909  
##  1st Qu.:-0.6255   1st Qu.:-0.4458    1st Qu.:-0.85726   1st Qu.:-0.62599  
##  Median :-0.3315   Median :-0.3677    Median :-0.06537   Median :-0.00232  
##  Mean   : 0.0000   Mean   : 0.0000    Mean   : 0.00000   Mean   : 0.00000  
##  3rd Qu.: 0.2964   3rd Qu.:-0.1464    3rd Qu.: 0.62754   3rd Qu.: 0.65222  
##  Max.   : 3.7852   Max.   : 4.0623    Max.   : 2.60727   Max.   : 2.43307  
##    GrenSpace         RatioBornUK       RatioCTFtoH       RatioOwndDwel    
##  Min.   :-1.2211   Min.   :-2.2525   Min.   :-3.9194   Min.   :-1.9282  
##  1st Qu.:-0.7011   1st Qu.:-0.6027   1st Qu.:-0.3728   1st Qu.:-0.8612  
##  Median :-0.2852   Median : 0.2625   Median : 0.1893   Median :-0.1297  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 0.5121   3rd Qu.: 0.7397   3rd Qu.: 0.7469   3rd Qu.: 0.7441  
##  Max.   : 3.5694   Max.   : 1.9145   Max.   : 1.4342   Max.   : 3.6237  
##    FlatsRate      
##  Min.   :-3.5828  
##  1st Qu.:-0.3632  
##  Median : 0.2117  
##  Mean   : 0.0000  
##  3rd Qu.: 0.8088  
##  Max.   : 1.1487
```

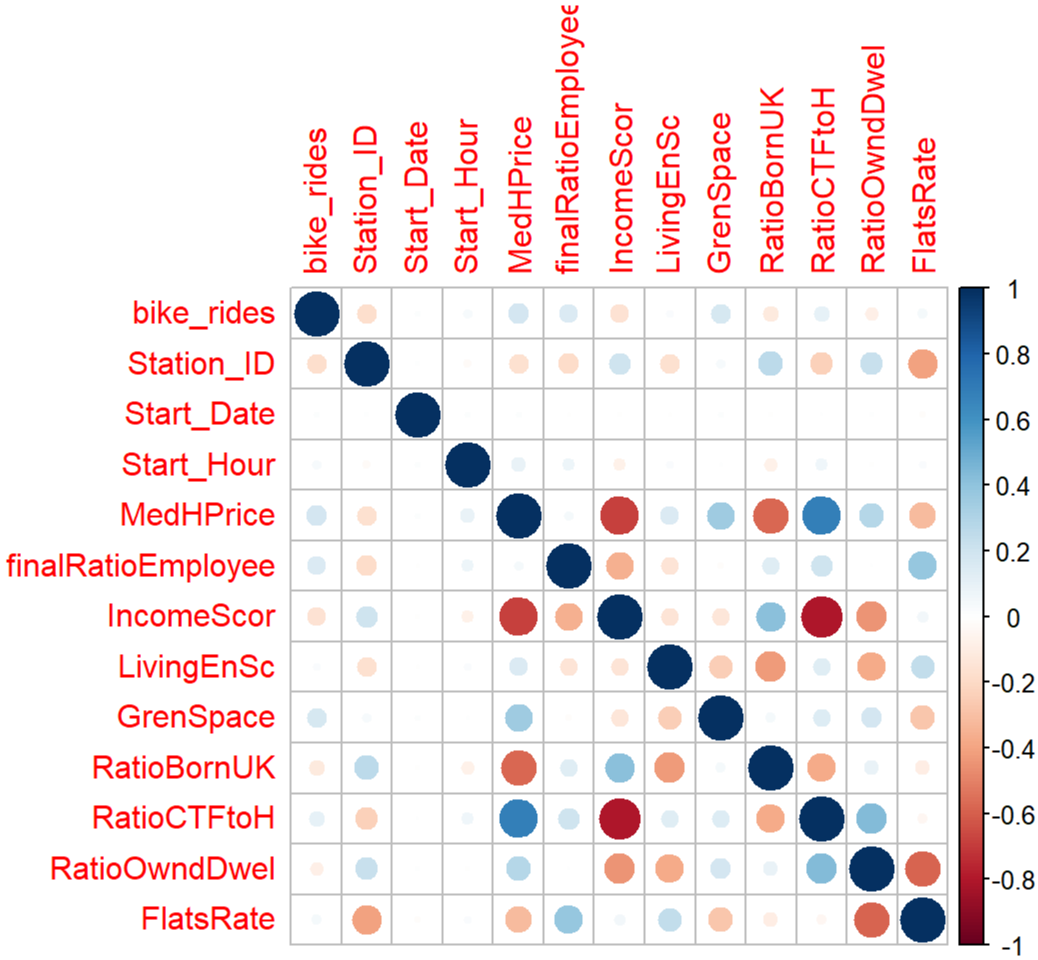
Checking for multicollinearity.

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.6.2
```

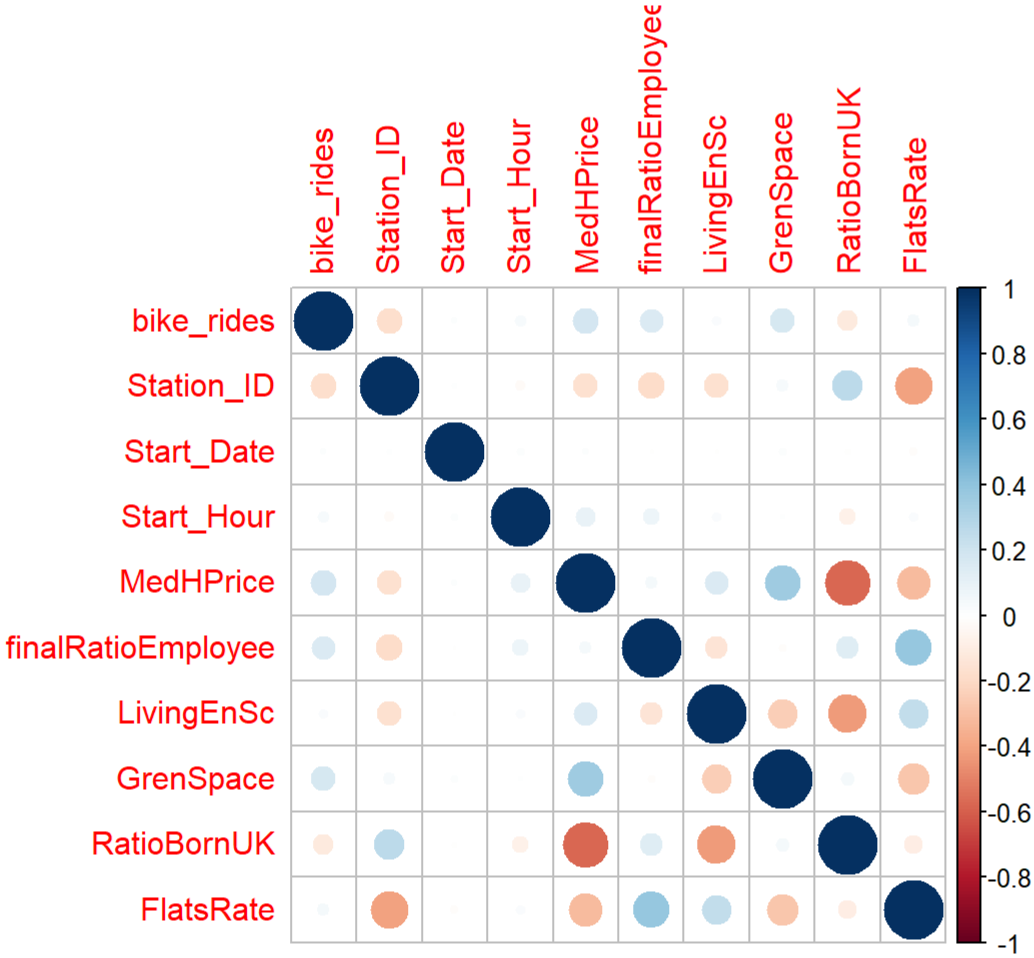
```
## corrplot 0.84 loaded
```

```
corrplot(cor(mydata_std))
```




There is high correlation between RatioCTFtoH, RatioOwndDwel and IncomeScor so they will be romoved from the model. Again checking multicollinearity.

```
mydata_std$RatioCTFtoH = NULL
mydata_std$RatioOwndDwel = NULL
mydata_std$IncomeScor = NULL
corrplot(cor(mydata_std))
```




##Algorithms Linear regression model needs to implemented as part of the final goal

```
set.seed(0)
trainIdx = sample(1:nrow(mydata_std), 0.75*nrow(mydata_std))
train = mydata_std[trainIdx]
test = mydata_std[-trainIdx]
lr = lm(bike_rides ~ ., data=train)
train_preds = predict(lr, train)
test_preds = predict(lr, test)
```

Printing the R2 scores

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

```
## [1] "R2 on train: 0.0925138096168877"
```

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

```
## [1] "R2 on test: 0.0933658930141483"
```

##Data undestanding Plotting the beta coefficients ot understand the model.

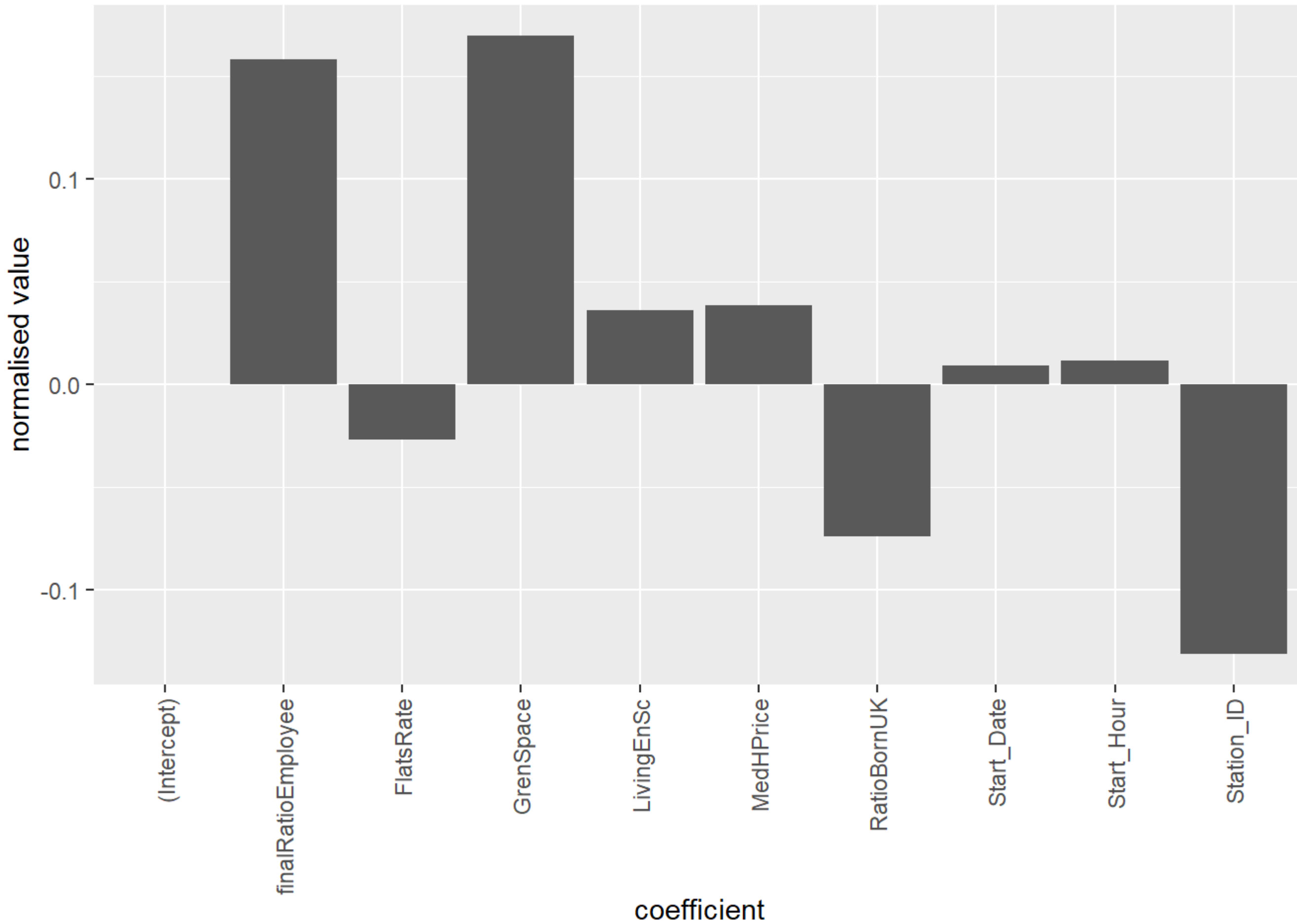
```
lr = lm(bike_rides ~ ., data=mydata_std)
summary(lr)
```

```
##
## Call:
## lm(formula = bike_rides ~ ., data = mydata_std)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4997 -0.6818 -0.0653  0.5922  4.1429
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.825e-13  7.700e-04    0.00        1
## Station_ID         -1.316e-01  8.970e-04 -146.68   <2e-16 ***
## Start_Date          9.129e-03  7.703e-04   11.85   <2e-16 ***
## Start_Hour          1.142e-02  7.768e-04   14.70   <2e-16 ***
## MedHPrice           3.821e-02  1.328e-03   28.77   <2e-16 ***
## finalRatioEmployee  1.579e-01  9.661e-04  163.42   <2e-16 ***
## LivingEnSc          3.590e-02  9.199e-04   39.02   <2e-16 ***
## GrenSpace           1.694e-01  9.063e-04  186.91   <2e-16 ***
## RatioBornUK        -7.413e-02  1.185e-03  -62.54   <2e-16 ***
## FlatsRate          -2.711e-02  1.170e-03  -23.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9525 on 1530230 degrees of freedom
## Multiple R-squared:  0.09273,    Adjusted R-squared:  0.09272
## F-statistic: 1.738e+04 on 9 and 1530230 DF,  p-value: < 2.2e-16
```

```
library(ggplot2)
```

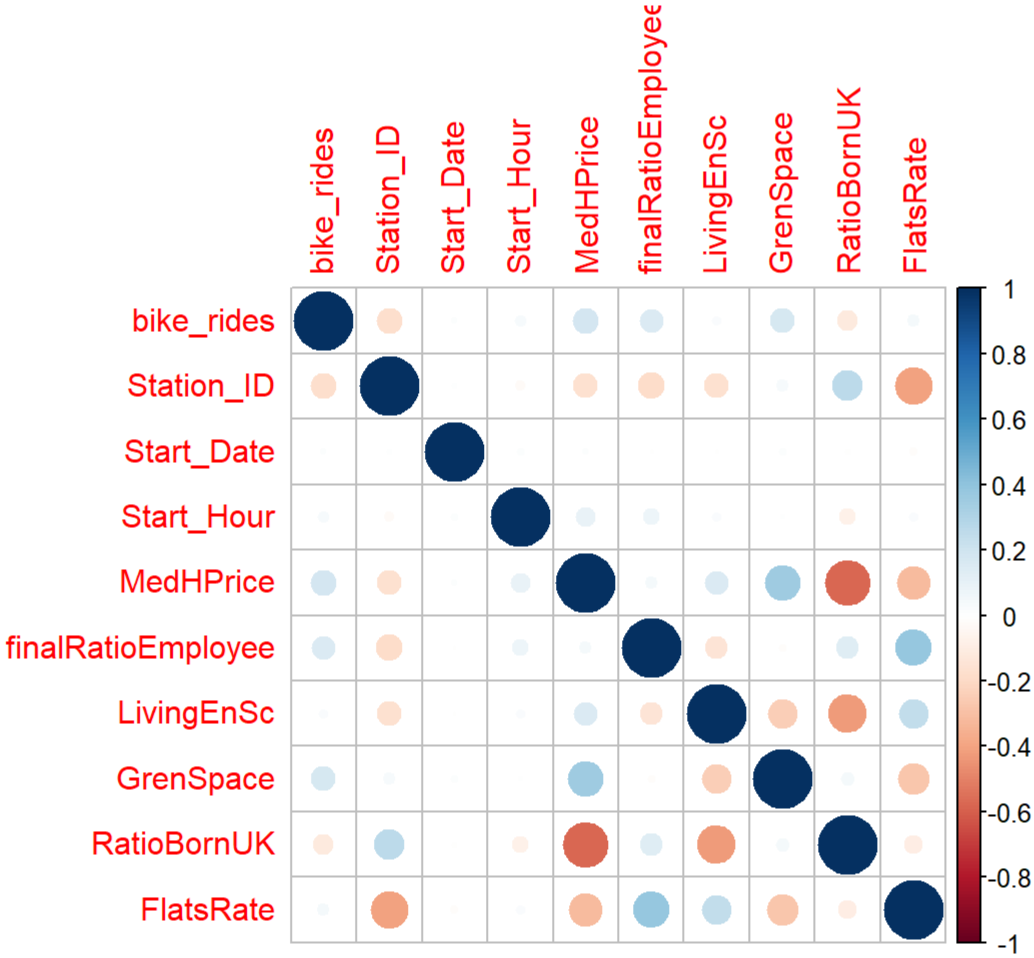
```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
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")
```



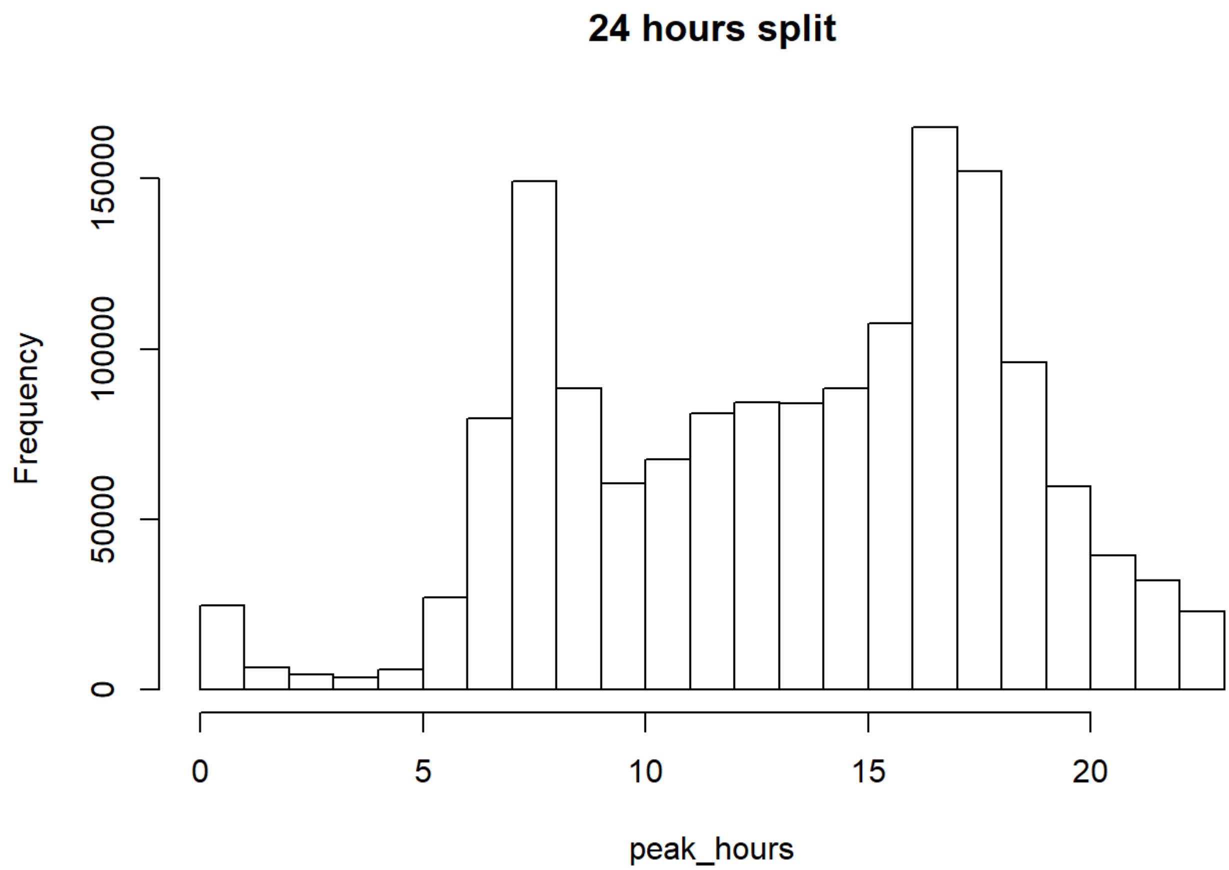

Checking the multicollinearity of the data

```
corrplot(cor(mydata_std))
```

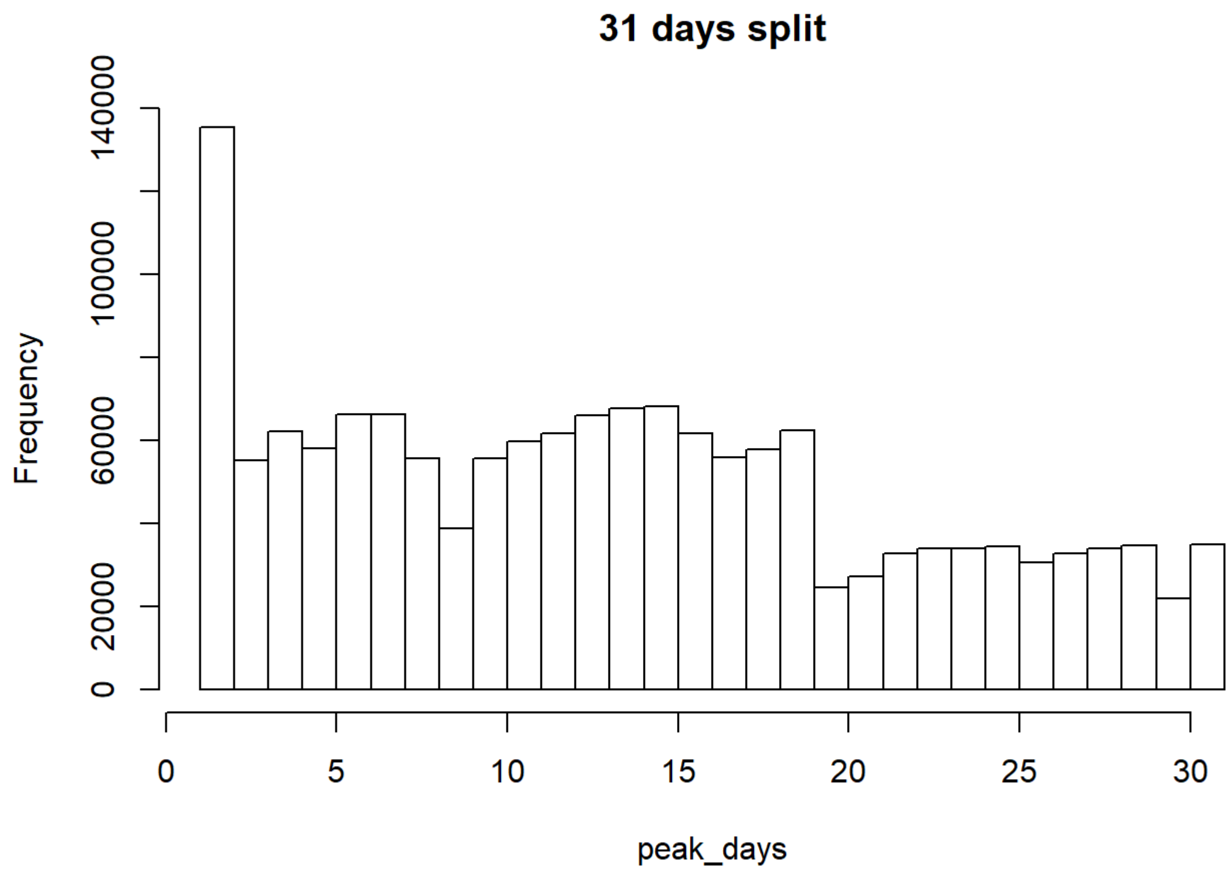



Plotting histograms to prove H1 and H2

```
peak_hours <- total2$Start_Hour
hist(peak_hours, breaks = 24, main = "24 hours split")
```




```
peak_days <- total2$Start_Date
hist(peak_days, breaks = 31, main = "31 days split")
```




##Main findings H1. Bikes demand is higher durin peak hours. TRUE. As seen from vis 24 hour split, where we can see there are clear high deman during peak hours (07-09:00 and 16-18:00), which prooves our hypothesis.

H2. Bikes demand have a daily trend. TRUE. As seen in vis 31 days split, there is higher demand in the first half of the month, than the second half.

H3. Higher demand of bikes rented at stations which are close to central London. Cannot be falsified due to the the fact that the data needs to be standardised and the values for the locations is not numeric

H4. Higher demand of bikes rented where is high employment rate. TRUE.

H5. Higher demand of bikes rented where is high population density. Cannot be falsified due to multicollinearity.

H6. Higher demand of bikes rented where is high percentage of green space. TRUE. We can see that the bike_rides are fairly high correlated to the zones with high concentration of green spaces.

H7. Higher demand of bikes rented in deprived areas. FALSE

H8. Higher demand of bikes rented in poor areas. TRUE. Lower demand in wealthier zones.

H9. Higher demand of bikes rented where is high immigration rate. TRUE. We can see that the bike_rides are fairly high correlated to the zones where there are people who are predominantly born in UK

H10. Higher demand of bikes rented where is high flats rate. FALSE.

H11. Higher demand of bikes rented where is low number of owned properties rate. Cannot be falsified due to multicollinearity of OwndDwelRate

##Limitaions

1. The short period of time, reviewed in the dataset, does not allow us to do perfected model. More months would give us better predictions
2. Multicollinearity of some of the features reduces the accuracy of the model
3. Introducing weather data would further improve our model as we would be able to take external factors.