R Notebook

#PROBLEM DEFINATION: Santander cycle is a public bicycle hire scheme, contracted by the transport of london or the tfl.However, with expansion of this scheme is important to have an equilibrium between the demand and the supply of these cycles for every particular place. *Problem:* The problem here is to find the total number of bikes rented in each bike station to allow every station to have quilibriumn quantity of cycles. *DataSets:* 1.*bike_journeys.csv*:This dataset contains the journey information about a bike.It contains the detail information about the bikes journey, that contains the station where the bike started and where it ends, the time of the journey along with its departure and arrival time. 2.*Bike_station.csv*: This dataset contains the infromation about the bike station, which contains the name and ID of the station, coordinates of its location and the number of bikes in the station. 3.*London_Census.csv*:This dataset contains all the information about the london census data, which contains information about the people living in that place and their living standards. *Temporarl Granularity:* Hour

#RESEARCH QUESTIONS: 1.Predctions of total number of bikes 2.Factors associated with the demand of the bikes in a station

```
library(Amelia)
```

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

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

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.3
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(caretEnsemble)
```

```
## Warning: package 'caretEnsemble' was built under R version 3.6.3
```

```
## 
## Attaching package: 'caretEnsemble'
## The following object is masked from 'package:ggplot2':
##
##     autoplot

library(corrplot)

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

## corrplot 0.84 loaded

library(data.table)

## Warning: package 'data.table' was built under R version 3.6.3

library(dbplyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:dbplyr':
##
##     ident, sql

## 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

library(e1071)

## Warning: package 'e1071' was built under R version 3.6.3

library(evaluate)
library(fuzzyjoin)

## Warning: package 'fuzzyjoin' was built under R version 3.6.2

library(geosphere)

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

library(ggplot2)
library(knitr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following object is masked from 'package:base':
##
##     date
```

```r
library(matrixStats)
```

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

##
## Attaching package: 'matrixStats'

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

```r
library(plyr)
```

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

## --------------------------------------------------------------------------
----

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first,
then dplyr:
## library(plyr); library(dplyr)

## --------------------------------------------------------------------------
----

##
## Attaching package: 'plyr'

## The following object is masked from 'package:matrixStats':
##
##     count

## The following object is masked from 'package:lubridate':
##
##     here

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(reshape2)

##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:data.table':
##
##     dcast, melt

library(tibble)
library(tidyr)

##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths

library(tidyselect)
library(timeDate)

## Warning: package 'timeDate' was built under R version 3.6.3

##
## Attaching package: 'timeDate'

## The following objects are masked from 'package:e1071':
##
##     kurtosis, skewness

library(tinytex)

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

#SOLUTION *Loading data*

```
x=read.csv("C:/Users/sarka/OneDrive/Desktop/submitted/4070/data/data
(9)/bike_journeys.csv")
y=read.csv("C:/Users/sarka/OneDrive/Desktop/submitted/4070/data/data
(9)/bike_stations.csv")
z=read.csv("C:/Users/sarka/OneDrive/Desktop/submitted/4070/data/data
(9)/London_census.csv")
```

Understanding the data

```
head(x)

##   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
```

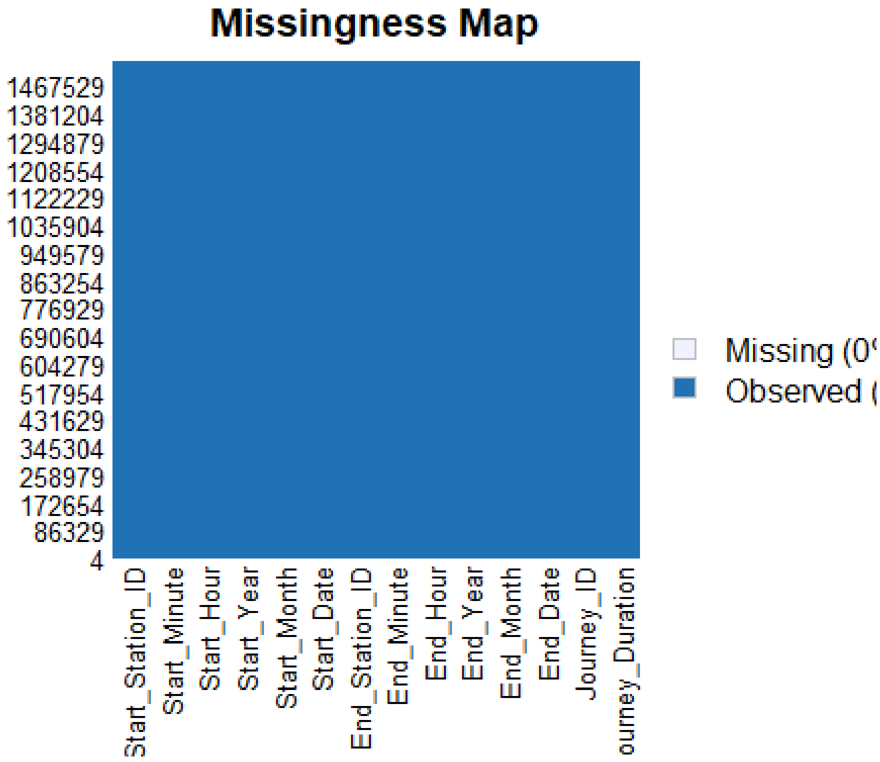
```
head(y)
```

```
##   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
```

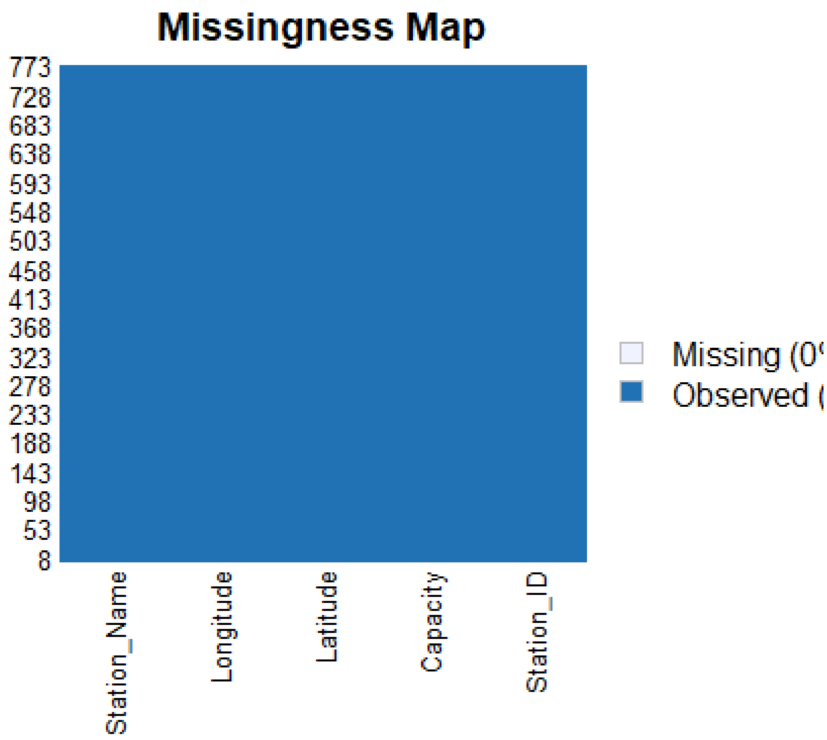
```
head(z)
```

```
##    WardCode       WardName              borough NESW AreaSqKm Longitude
## 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
##   Latitude IncomeScor LivingEnSc NoEmployee GrenSpace PopDen BornUK
```

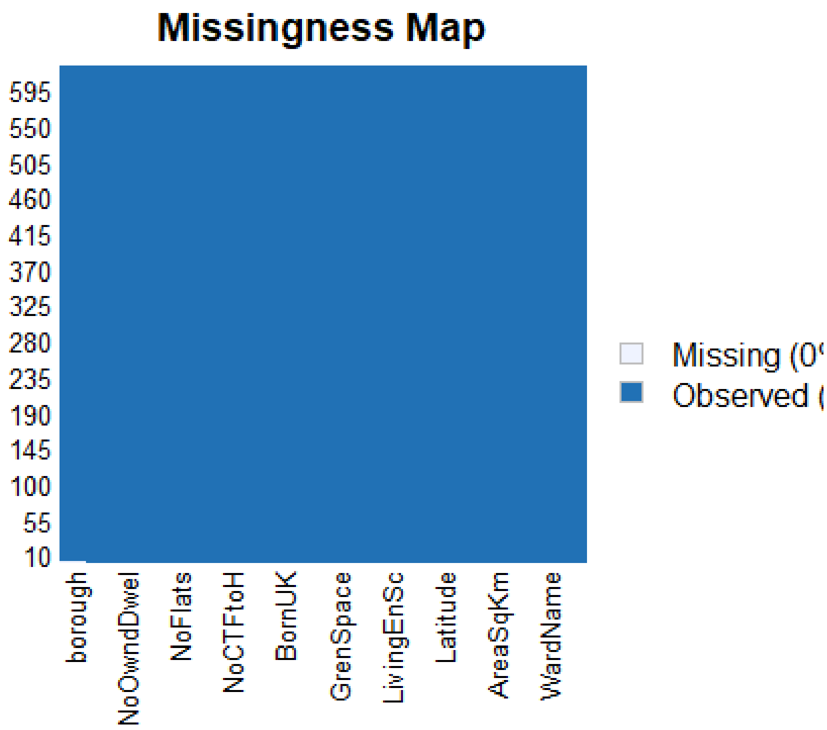
```
NotBornUK
## 1 51.53971       0.27      42.76       7900      19.6 9884.6   5459
7327
## 2 51.54559       0.28      27.96        800      22.4 7464.3   7824
2561
## 3 51.55453       0.25      31.59       1100       3.0 8923.1   8075
3470
## 4 51.58475       0.27      34.78       1700      56.4 2970.6   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

missmap(x)
```



```
missmap(y)
```


Missingness Map
773
728
683
638
593
548
503
458
413
368
323
278
233
188
143
98
53
8
Station_Name
Longitude
Latitude
Capacity
Station_ID
Missing (0
Observed (

```
missmap(z)
```


Missingness Map
595
550
505
460
415
370
325
280
235
190
145
100
55
10
borough
NoOwndDwel
NoFlats
NoCTFtoH
BornUK
GrenSpace
LivingEnSc
Latitude
AreaSqKm
WardName
Missing (0
Observed (

#HYPOTHESIS: *H1*: Higher quantity of cycles are demanded in *weekdays H2*: Higher cycles are demanded in place which high *ratio of Non owned Dwelling H3*: More Cycles are demanded in the places with high *ratio of UK born people H5*: places with higher *employment ratio* will have higher demand for cycles *H6*: Area with more *greenspace* will have high demand for cycles *H7*: places with higher *incomescore* will have higher demand for cycles *H8*: places with higher *Population ratio* will have higher demand for cycles All the hypothesis mentioned above can be falsefiable with the given data.

#METRICES The hypothesis can be worked out using the correct matrices *weekdays:* From the bike journey dataset the date is created from the given days, months and years from the bike start journey details, then the data is created into a boolean such that if the date is a weekday then 1 is used and if weakend then 0 is used.*H1 Ratio of Non Owned Dwelling:*This is obtained by calculating RNonOwnedDwel=((NoDwelling-NoOwndDwel)/NoDwelling)*H2 Ratio of UK born people:* This is calculated using (BornUK)/(BornUKNotBornUK)*H3 employment ratio:*It is obtained by NoEmployee/product of(PopDen and AreaSqKm) *H4 greenspace:* THis can be directly found in the data set it is the percentage of greenspace present in the Ward*H5 incomescor:* This represents the deprivation of an area. higher the score more deprived is the area.*H6* Population ratio:* The percentage population of that area*H7*

#PRE-PREPROCESSING *Date Transformation:*The journey timming of the dataset is given in days, month and year,It is converted in to Date, a boolean colummn is introduced for weekdays, is the day is a weekday then the value is 1 and if weekend the value is zero. This will later help us in building the matrices

```
x$Date<-as.Date(with(x,paste(Start_Date,Start_Month,Start_Year,sep="-")),"%d-
%m-%y")
x$Day<-(wday(x$Date,label = TRUE))
x$Date <- as.Date(x$Date)
x$weekend<-(as.numeric(as.logical(isWeekday(x$Date))))
head(x)
```

```
##   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       Date Day weekend
## 1              251 2017-09-19 Tue       1
## 2              550 2017-09-19 Tue       1
## 3              212 2017-09-15 Fri       1
## 4              163 2017-09-14 Thu       1
## 5               36 2017-09-13 Wed       1
## 6              589 2017-09-14 Thu       1
```

Merging Dataset The bike station and bike journey is merged along with the station ID of the bike station and Start_station_ID of the bike journey. Here start_station ID is used because the demand of the station is more likely to be associated with the starting place of the bike as it is the placce where the bikes are demanded.The merged dataset is then stored in total.

```
total<-merge(x,y,by.y= "Station_ID",by.x ="Start_Station_ID")
head(total)
```

```
##   Start_Station_ID Journey_Duration Journey_ID End_Date End_Month End_Year
## 1                1              240       5930       16         9       17
## 2                1             1080      11075       14         9       17
## 3                1              543      10283       12         8       17
## 4                1              720       1311        4         9       17
## 5                1              748      14820        8         8       17
## 6                1              767      14288        3         8       17
##   End_Hour End_Minute End_Station_ID Start_Date Start_Month Start_Year
## 1       11          6            264         16           9         17
## 2       19          1            641         14           9         17
## 3       12         19            275         12           8         17
## 4        6         53            136          4           9         17
## 5       11         38            232          8           8         17
## 6       20         55            536          3           8         17
##   Start_Hour Start_Minute       Date Day weekend Capacity Latitude
Longitude
## 1         11            2 2017-09-16 Sat       0       19 51.52916  -
0.10997
## 2         18           43 2017-09-14 Thu       1       19 51.52916  -
0.10997
## 3         12           10 2017-08-12 Sat       0       19 51.52916  -
0.10997
## 4          6           41 2017-09-04 Mon       1       19 51.52916  -
0.10997
## 5         11           26 2017-08-08 Tue       1       19 51.52916  -
0.10997
## 6         20           42 2017-08-03 Thu       1       19 51.52916  -
0.10997
##                  Station_Name
## 1 River Street , Clerkenwell
## 2 River Street , Clerkenwell
## 3 River Street , Clerkenwell
## 4 River Street , Clerkenwell
## 5 River Street , Clerkenwell
## 6 River Street , Clerkenwell
```

To merge London Census data with the merged dataset the distance between the the station location coordinates and the coordinates of the ward is cevaluated. This is done with the idea to associate the station with the nearest ward, and then each ward is associated with the nearnest station

```
d<-
distm(cbind((y$Longitude),(y$Latitude)),cbind((z$Longitude),(z$Latitude)),fun
= distHaversine)
#m2<-matrixStats::rowMins(d)
index<-(d==(apply(d,1,FUN = min)))%*% 1:ncol(d)
d1<-data.frame(Station_Name
=y$Station_Name,WardCode=z$WardCode[index],distance=(apply(d,1,FUN = min)))
head(d1)
```

```
##                            Station_Name  WardCode distance
## 1           River Street , Clerkenwell E05000370 317.6220
## 2        Phillimore Gardens, Kensington E05000382 356.3519
## 3 Christopher Street, Liverpool Street E05000367 802.1570
## 4        St. Chad's Street, King's Cross E05000141 325.1939
## 5         Sedding Street, Sloane Square E05000390 380.6109
## 6        Broadcasting House, Marylebone E05000641 456.4005
```

Then the datasets are joined allong with the station name and the ward.The final merged dataset is then stored as a datframe in K

```
r=merge.data.frame(z,d1,z.by=WardCode,d1.by=WardCode)
k=merge(r,total,by="Station_Name")
head(k)
```

```
##                         Station_Name  WardCode       WardName     borough
## 1 Abbey Orchard Street, Westminster E05000646 Vincent Square Westminster
## 2 Abbey Orchard Street, Westminster E05000646 Vincent Square Westminster
## 3 Abbey Orchard Street, Westminster E05000646 Vincent Square Westminster
## 4 Abbey Orchard Street, Westminster E05000646 Vincent Square Westminster
## 5 Abbey Orchard Street, Westminster E05000646 Vincent Square Westminster
## 6 Abbey Orchard Street, Westminster E05000646 Vincent Square Westminster
##      NESW AreaSqKm Longitude.x Latitude.x IncomeScor LivingEnSc NoEmployee
## 1 Central      0.7   -0.131793   51.49292       0.14      45.24      21800
## 2 Central      0.7   -0.131793   51.49292       0.14      45.24      21800
## 3 Central      0.7   -0.131793   51.49292       0.14      45.24      21800
## 4 Central      0.7   -0.131793   51.49292       0.14      45.24      21800
## 5 Central      0.7   -0.131793   51.49292       0.14      45.24      21800
## 6 Central      0.7   -0.131793   51.49292       0.14      45.24      21800
##   GrenSpace  PopDen BornUK NotBornUK NoCTFtoH NoDwelling NoFlats NoHouses
## 1        30 14285.7   5532      4456     41.7       5674    5566      148
## 2        30 14285.7   5532      4456     41.7       5674    5566      148
## 3        30 14285.7   5532      4456     41.7       5674    5566      148
## 4        30 14285.7   5532      4456     41.7       5674    5566      148
## 5        30 14285.7   5532      4456     41.7       5674    5566      148
## 6        30 14285.7   5532      4456     41.7       5674    5566      148
##   NoOwndDwel MedHPrice distance Start_Station_ID Journey_Duration
Journey_ID
## 1       1709    600000 579.7023              108              360
12870
## 2       1709    600000 579.7023              108              960
12676
## 3       1709    600000 579.7023              108             1768
6209
## 4       1709    600000 579.7023              108              240
12685
## 5       1709    600000 579.7023              108             1599
3032
## 6       1709    600000 579.7023              108             1307
4268
##   End_Date End_Month End_Year End_Hour End_Minute End_Station_ID
Start_Date
## 1       19         9       17       18         52            341
19
## 2       18         9       17       21         54            194
18
## 3       12         8       17       15         52            138
12
## 4       18         9       17       15         34            177
18
## 5       22         8       17       17         51            200
22
## 6       26         8       17        1         59            653
26
##   Start_Month Start_Year Start_Hour Start_Minute       Date Day weekend
## 1           9         17         18           46 2017-09-19 Tue       1
## 2           9         17         21           38 2017-09-18 Mon       1
## 3           8         17         15           23 2017-08-12 Sat       0
## 4           9         17         15           30 2017-09-18 Mon       1
## 5           8         17         17           24 2017-08-22 Tue       1
## 6           8         17          1           37 2017-08-26 Sat       0
##   Capacity Latitude.y Longitude.y
## 1       29   51.49813   -0.132102
## 2       29   51.49813   -0.132102
## 3       29   51.49813   -0.132102
## 4       29   51.49813   -0.132102
## 5       29   51.49813   -0.132102
## 6       29   51.49813   -0.132102
```

creating Matrices Here a table is created, containing, start stationID and weekend to obatian the number of occurance to obtain the cycles demanded in hour granularity.the table is then converted into a dataframe

```
k2<-table(Hour=x$Start_Hour,StationID=x$Start_Station_ID,Weekend=x$weekend)
d2<-data.frame(melt(data = k2,id=c(Hour)))
head(d2)
```

```
##   Hour StationID Weekend value
## 1    0         1       0     6
## 2    1         1       0     2
## 3    2         1       0     0
## 4    3         1       0     1
## 5    4         1       0     3
## 6    5         1       0     0
```

Then the remaning elements of the hypothesis are evalulated.Here uniquie is used to avoid repeation of the rows and a dataframe is used.

```
k3<-
unique(data.frame(StationID=k$Start_Station_ID,RNonOwnedDwel=(((k$NoDwelling)
-
(k$NoOwndDwel))/k$NoDwelling),RUKborn=(k$BornUK)/((k$BornUK)+(k$NotBornUK)),E
mploymentRatio=(k$NoEmployee/((k$PopDen)*(k$AreaSqKm))),population=k$PopDen,I
ncomeScore=k$IncomeScor,greenspace=k$GrenSpace))
head(k3)
```

```
##      StationID RNonOwnedDwel   RUKborn EmploymentRatio population
IncomeScore
## 1          108     0.6988016 0.5538646       2.1800022    14285.7
0.14
## 2298       559     0.5671267 0.4330910       0.3692308     9750.0
0.09
## 2954       394     0.6263359 0.4736199       0.2441315    17750.0
0.15
## 3983       554     0.7920755 0.6048186       0.1739135    12458.3
0.44
## 4327       583     0.6988016 0.5538646       2.1800022    14285.7
0.14
## 7565        38     0.6043207 0.4039748       0.6934687    16583.3
0.08
##      greenspace
## 1          30.0
## 2298       28.2
## 2954       15.5
## 3983       12.5
## 4327       30.0
## 7565        5.9
```

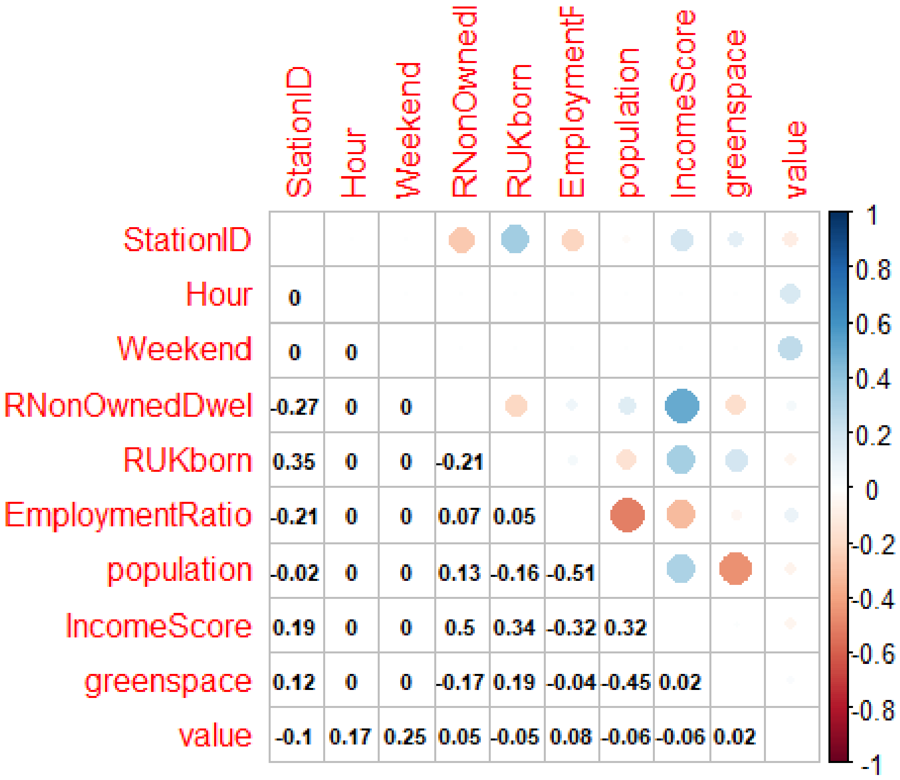
The tow data frame d3 and K3 are stored and the value column us moved to the extreme right as that will be our dependent variable

```
d0<-merge(d2,k3,d2.by=StationID)
dd<-d0[,c(1,2,3,5,6,7,8,9,10,4)]
head(dd)
```

```
##   StationID Hour Weekend RNonOwnedDwel   RUKborn EmploymentRatio
population
## 1         1   16       1     0.7236447 0.6145344        3.817385
12777.8
## 2         1   22       1     0.7236447 0.6145344        3.817385
12777.8
## 3         1   17       1     0.7236447 0.6145344        3.817385
12777.8
## 4         1    2       0     0.7236447 0.6145344        3.817385
12777.8
## 5         1   21       0     0.7236447 0.6145344        3.817385
12777.8
## 6         1   21       1     0.7236447 0.6145344        3.817385
12777.8
##   IncomeScore greenspace value
## 1        0.21        9.3    35
## 2        0.21        9.3     5
## 3        0.21        9.3    70
## 4        0.21        9.3     0
## 5        0.21        9.3     9
## 6        0.21        9.3    12
```

#DATA UNDERSTANDING AFTER PRE-PROCESSING, TO GET A PRESENT IDEA ABOUT THE MODIFIED DATA Correaltion between the elements are checked. multicolinearity is can have missleading effects.

```
cor1 = cor(dd)
corrplot.mixed(cor1, lower.col = "black", number.cex = .7, tl.pos = "lt")
```



Thus removing the elements that have multicollinearity and station ID.

```
dd$population= NULL
dd$RNonOwnedDwel= NULL
dd$EmploymentRatio= NULL
dd$StationID=NULL
```

The summary of the merged data is used to the normalisation of the data and if there exists any null value

```
summary(dd)

##       Hour          Weekend        RUKborn         IncomeScore    
##  Min.   : 0.00   Min.   :0.0   Min.   :0.3543   Min.   :0.0100  
##  1st Qu.: 5.75   1st Qu.:0.0   1st Qu.:0.4882   1st Qu.:0.1100  
##  Median :11.50   Median :0.5   Median :0.5565   Median :0.1800  
##  Mean   :11.50   Mean   :0.5   Mean   :0.5416   Mean   :0.1877  
##  3rd Qu.:17.25   3rd Qu.:1.0   3rd Qu.:0.5999   3rd Qu.:0.2500  
##  Max.   :23.00   Max.   :1.0   Max.   :0.7112   Max.   :0.4400  
##    greenspace        value        
##  Min.   : 0.00   Min.   :   0.00  
##  1st Qu.: 7.60   1st Qu.:   5.00  
##  Median :13.50   Median :  19.00  
##  Mean   :17.01   Mean   :  41.35  
##  3rd Qu.:25.00   3rd Qu.:  48.00  
##  Max.   :69.10   Max.   :4740.00
```

here it can be see that Value,RUKborn and greenspace needs to be standardised.so those values are standardised and all the null values and infinate values or nanas produced are then removed. Note here it is good to do the standardisation first and then the null values are to be rmoved this is so because, during the standardisation the null values can be produced.

```
dd$value=log10(dd$value + min(dd["value"!=0]$value))
dd$RUKborn=log10(dd$RUKborn + min(dd["RUKborn"!=0]$RUKborn))
dd$greenspace=log10(dd$greenspace +min(dd["greenspace"!=0]$greenspace))

is.na(dd)<-sapply(dd,is.infinite)
mydata <- (na.omit(dd))
summary(mydata)

##       Hour          Weekend          RUKborn          IncomeScore    
##  Min.   : 0.00   Min.   :0.0000   Min.   :-0.14960   Min.   :0.0100  
##  1st Qu.: 7.00   1st Qu.:0.0000   1st Qu.:-0.07445   1st Qu.:0.1100  
##  Median :12.00   Median :1.0000   Median :-0.04069   Median :0.1800  
##  Mean   :11.99   Mean   :0.5085   Mean   :-0.04978   Mean   :0.1887  
##  3rd Qu.:18.00   3rd Qu.:1.0000   3rd Qu.:-0.02036   3rd Qu.:0.2500  
##  Max.   :23.00   Max.   :1.0000   Max.   : 0.02757   Max.   :0.4400  
##    greenspace          value       
##  Min.   :-0.5229   Min.   :0.0000  
##  1st Qu.: 0.9294   1st Qu.:0.8451  
##  Median : 1.1303   Median :1.3424  
##  Mean   : 1.1025   Mean   :1.2666  
##  3rd Qu.: 1.3979   3rd Qu.:1.7076  
##  Max.   : 1.8395   Max.   :3.6758
```

#ALGORITHMS The data is split using K-fold cross validation, and then simple learner regression model is used. Then the model summary is print.

```
set.seed(180)
train.control<-trainControl(method="CV",number=500)
model<-caret::train(value ~., data = mydata, method = "lm",trControl =
train.control)
```

#UNDERSTANDING THE MATRICES.

```
print(model)

## Linear Regression 
## 
## 34310 samples
##     5 predictor
## 
## No pre-processing
## Resampling: Cross-Validated (500 fold) 
## Summary of sample sizes: 34242, 34242, 34242, 34242, 34241, 34241, ... 
## Resampling results:
## 
```

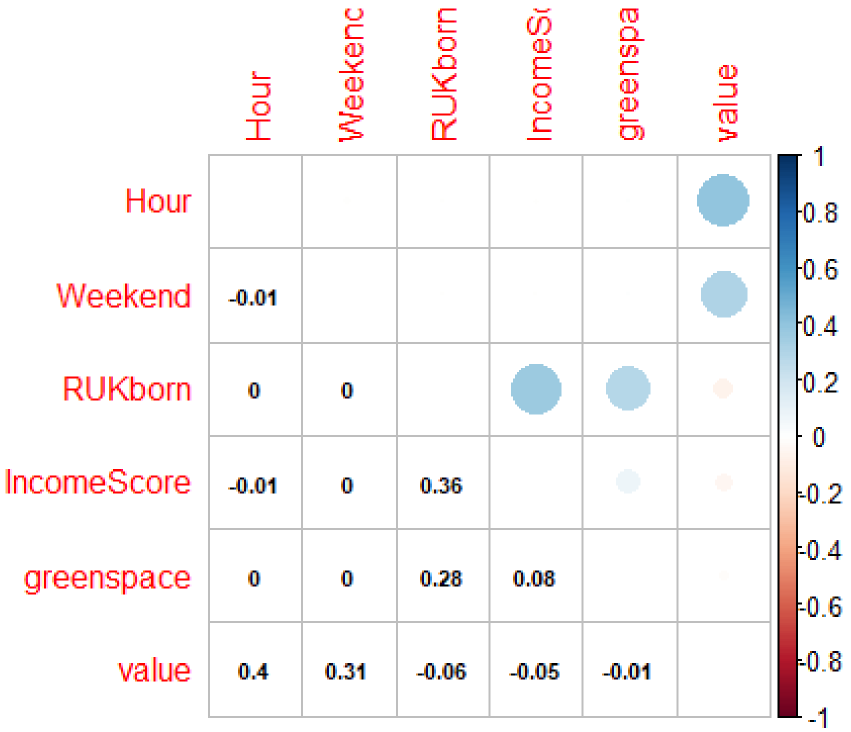
```
##   RMSE       Rsquared   MAE
##   0.5311094  0.2667597  0.4298348
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

MAE MAE also known as the mean absolute error is the Average of all the errors occurred, that is it is the average of the total difference between the actual value and the predicted value. Hence it should be minimum. for both the test and train dataset is fairly high and the values of both the dataset are nearly the same. **RMSE** Root mean square error is the root of Mean square error. The values as observed for both Training Dataset and Test dataset is significantly high. **R2** R square value represents the value that implies how close the predicted values are fitted to the regression line. Higher the value, better the model. However, The value of R2 is comparatively low for both train and test dataset, this implies that the model is not a well-fitted model. Though the value of R2 is low, it is slightly higher for test dataset than training dataset implying that there is a good bias-variance tradeoff in the model.

```
summary(model)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90507 -0.35910  0.04695  0.37938  2.34562
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6251372  0.0143858  43.455  < 2e-16 ***
## Hour         0.0360366  0.0004219  85.424  < 2e-16 ***
## Weekend      0.3832406  0.0057501  66.649  < 2e-16 ***
## RUKborn     -0.7521805  0.0778911  -9.657  < 2e-16 ***
## IncomeScore -0.1520402  0.0303158  -5.015 5.32e-07 ***
## greenspace   0.0052458  0.0080521   0.651    0.515
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5324 on 34304 degrees of freedom
## Multiple R-squared:  0.2566, Adjusted R-squared:  0.2565
## F-statistic:  2368 on 5 and 34304 DF,  p-value: < 2.2e-16
```

This is the summary of the five residuals of the model. Residuals are the difference between the actual value and the predicted value of the given model **The median:** the media in the residuals should be minimum,such that zero is the most favourable value of median. In this model, the median is approximately zero. Thus, the data is normally distributed. **The estimated value:**The estimated value is the beta coefficient of the regression analysis to the dependent variable, in our case that is value. A positive value of the beta coefficient implies That is if the value of the independent variable increase, the value of the dependent variable also increases. In the given model the value of weekend,hour and greenspace are positively related to value that is the dependent variable. **Standard error:** standard error is the deviation between the actual mean and the predicted mean. zero standard error is the most favourable condition, which implies that the model is a good fit. Here the value of standard error for every coeffecient is zero, which signifies that it is a good fitted model. **t-value:**t-value is coefficient divided by its standard error. The higher the value of t, the higher the coefficient is likely to be true. In the given model weekend has highest t-value of 85 followed by hour and then NoFlats etc. **p-values:** The P-value is the probability value of the null value, hence lower the p-value higher is the chance of falsified null hypothesis. In our model more the stars, less is the P-value. Here all of our coefficient has three stars That mean those coefficient have less p-value which is favourable condition. **Residual standard error:** Residual standard error is the observed value and their arithmetic mean.The value of residual error if zero, then it implies that the model fits the data perfectly. In this model, the residual standard error is 0.5324, which is nearly zero. This means that the model is fitted well to the data **Multiple R-squared:**It describes the proportion of variation in the outcome as informed by the model. The higher the value of the Multiple R-squared closers to 1, the better the model is able to demonstrate the reason for the variations occurred. The Multiple R squared value is significantly low. Thus not much information is gathered about the variation in the outcome. Adjusted R square is similar to R square, that has been adjusted to the number of predictors. The value of Adjusted R-square is low, that explains that the model has not improved with the significant value. **F-statistic:** F-statistic represents the variability among the means that exceeds the expected value due to chance. The value of F-statistic is high, that means the null hypothesis can be rejected.

```r
cor1 = cor(mydata)
corrplot.mixed(cor1, lower.col = "black", number.cex = .7, tl.pos = "lt")
```



#Results interpretation: *H1*: Higher quantity of cycles are demanded in weekdays: TRUE *H3*: More Cycles are demanded in the places with high ratio of UK born people:false *H6*: Area with more greenspace will have high demand for cycles:False *H7*: Area with more IncomeScore will have high demand for cycles: False (Here it is to be noted that Income score is positively correalted with Population and RNonOwnedDwel, that means those hypothesis too are false and the is negatively corelated to income ratio, thus its is positively correlated too demand of cycle)

#Limitation: 1.Multicoolenarity contraints the usage of all the matrices. 2.The data for the bike journey has entries which implies that the cycles are used for some minutes.Those entries create ambiguity about the demand of the cycles. 3.Not much information is gathered from the metrices about the variation of the demand 4.The creation of the wards are done on poltical and geographical grounds. hence It is very likely that though a station is near a ward, it may not belong to that ward, which can create significant variation in the whole model.