Bike Prediction Demand

Code

Problem definition

Data: Dataset contains three tables which has London census data, bike stations data and bike journeys data for one year is available.

Problem: The problem is to predict the number of bikes demand on one station in one hour time. For this reason considering one station and one hour, finding the number of bikes.The station is Spatial granularity where as hour is temporal granularity.

Goal: The goal is to predict the number of bikes needed on one station in one hour time.

Research Questions

1. Is there any correlation between predictors and target variable?
2. Is there any Multicollinearity exist in predictors?
3. Which predictors have strong relationship with the target variable?

Preprocessing

Importing data

Hide

```
journeys=read.csv("D:\\Data Science\\Applied Data Analytics Tools\\CW Week 12\\Submission instru
ctions + data -- component 2-20191229\\data\\bike_journeys.csv")
```

Hide

```
stations=read.csv("D:\\Data Science\\Applied Data Analytics Tools\\CW Week 12\\Submission instru
ctions + data -- component 2-20191229\\data\\bike_s.csv")
census=read.csv("D:\\Data Science\\Applied Data Analytics Tools\\CW Week 12\\Submission instruct
ions + data -- component 2-20191229\\data\\census.csv")
```

Getting first 5 rows of data to understand what kind of data is there.

Hide

```
head(journeys)
```

	Journey_Duration <dbl>	Journey_ID <int>	End_D… <int>	End_Mo… <int>	End_Y… <int>	End_H… <int>	End_Minute <int>	End_Sta
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	

	Journey_Duration <dbl>	Journey_ID <int>	End_D… <int>	End_Mo… <int>	End_Y… <int>	End_H… <int>	End_Minute <int>	End_Sta
5	1200	14659	13	9	17	19	33	
6	1320	2351	14	9	17	14	53	

6 rows | 1-9 of 14 columns

Hide

```
head(stations)
```

	Station_ID <int>	Capacity <int>	Latitude <dbl>	Longitude <dbl>	Station_Name <fctr>
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

6 rows

Hide

```
head(census)
```

	WardCode <fctr>	WardName <fctr>	borough <fctr>	N… <fctr>	AreaS… <dbl>	lon <dbl>	lat <dbl>	Incom
1	E05000026	Abbey	Barking and Dagenham	East	1.3	0.077935	51.53971	
2	E05000027	Alibon	Barking and Dagenham	East	1.4	0.148270	51.54559	
3	E05000028	Becontree	Barking and Dagenham	East	1.3	0.118957	51.55453	
4	E05000029	Chadwell Heath	Barking and Dagenham	East	3.4	0.139985	51.58475	
5	E05000030	Eastbrook	Barking and Dagenham	East	3.5	0.173581	51.55365	
6	E05000031	Eastbury	Barking and Dagenham	East	1.4	0.105683	51.53590	

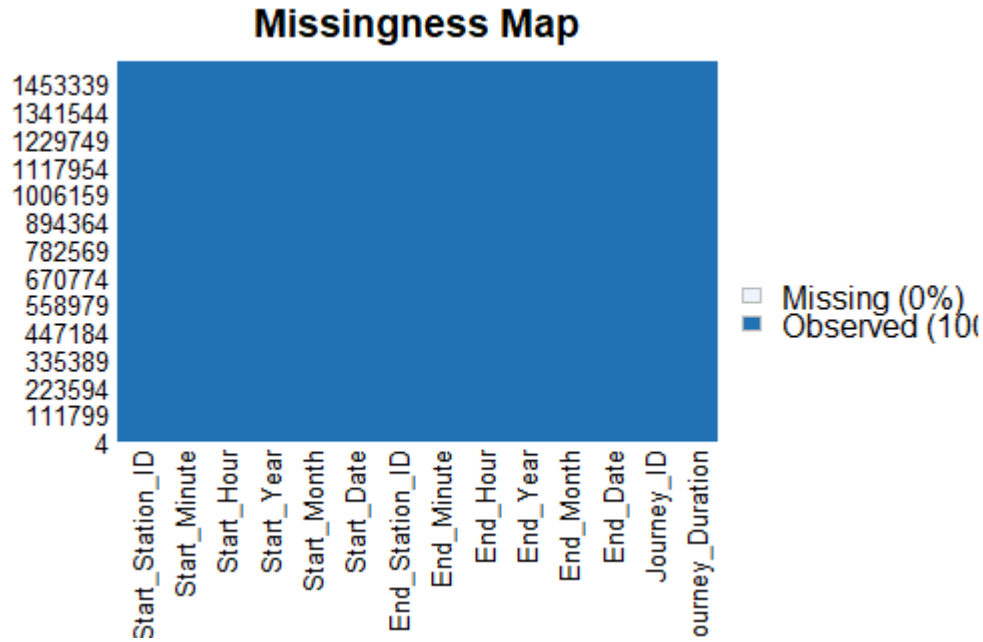
6 rows | 1-9 of 20 columns

Missing Values:

Before creating hypothesis, checking the data whether it has missing values or not using library Amelia. The plot below shows there is no missing values.

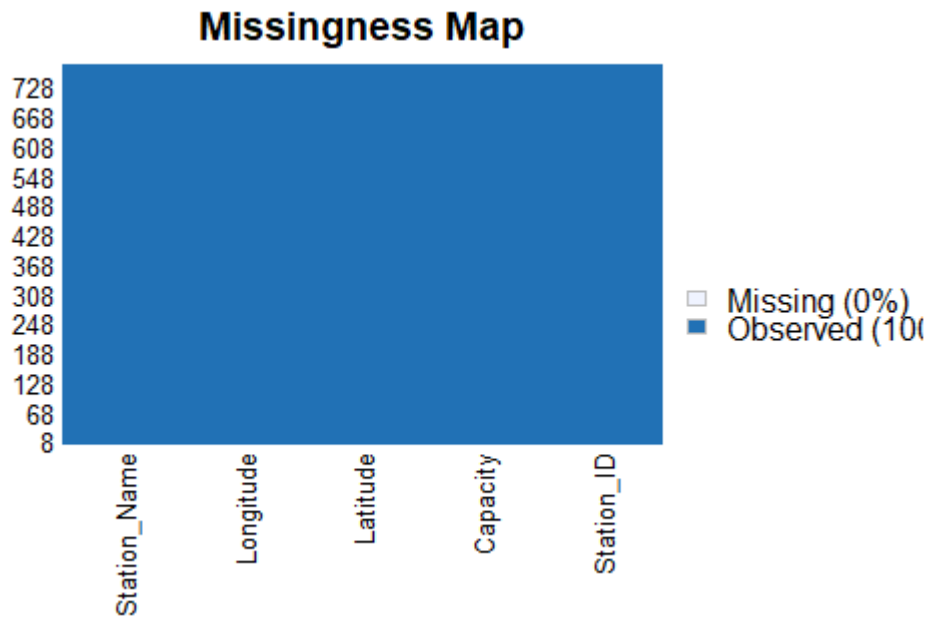
Hide

```
library(Amelia)
missmap(journeys)
```



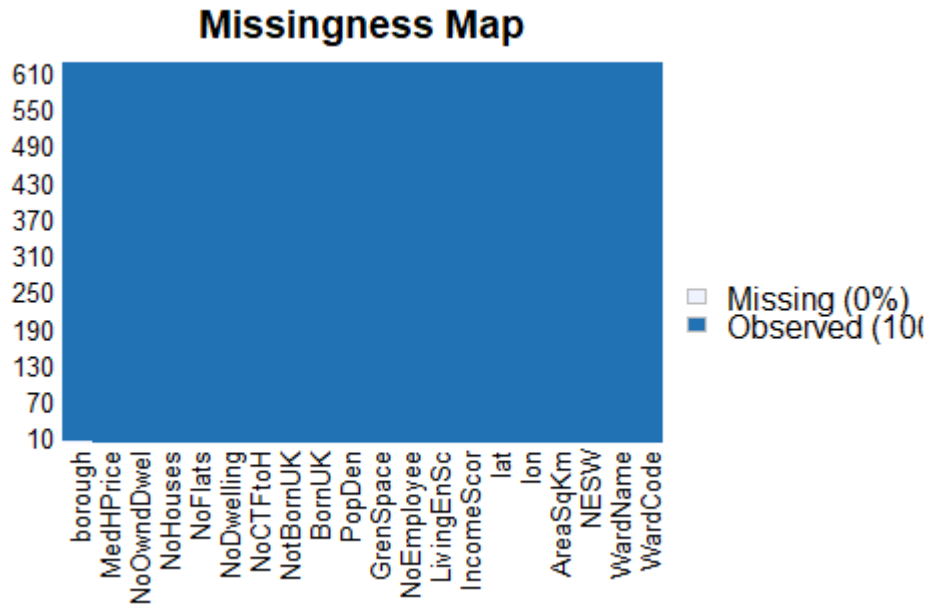
Hide

```
missmap(stations)
```



Hide

```
missmap(census)
```



Data Consistency

Checking the distinct values of the columns to check the consistency betwene all three data.

The London census data have 625 rows.

The Stations data have 773 rows.

In journeys data, 779 unqiue stations data is available.

Hide

```
length(unique(census$WardCode))
```

```
[1] 625
```

Hide

```
length(unique(stations$Station_ID))
```

```
[1] 773
```

Hide

```
length(unique(journeys$Start_Station_ID))
```

```
[1] 779
```

Intersect() function is giving common stations between journeys and stations data which are 771. We are losing 8 stations journey because stations data dont have that stations data.

Hide

```
length(unique(intersect(stations$Station_ID,journeys$Start_Station_ID)))
```

```
[1] 771
```

Calculating distance

To join three tables, there should be a unique column on which tables can be joined.However, stations and census tables do not have any common column. For this purpose, the wardcode will be assigned to stations in station table. To assign the **wardcode** to stations, the minimum distance of each **station** with the centre of ward in **census** data has been calculated based on two coordinates point (lat, lon) using **distm()** function and **geosphere** library.

After applying **distm()**, distance matrix is returned. A row in the matrix represents one station distance with all columns which represents wards. The index number of the **ward** has been extracted from each row with minimum distance. This index values has been used to get WardCode from census data and assigned to current station in stations data in a new created column **WardCode**.

Hide

```
library(geosphere)
x1=cbind((stations$Longitude), (stations$Latitude))
x2=cbind((census$lon), (census$lat))

# distance matrix
distance1=distm(x1, x2,fun=distHaversine)

# rowMins to find minimum distance and its index to a ward in a row.
minimum1=matrixStats::rowMins(distance1)
index1=(distance1==minimum1)%*% 1:ncol(distance1)

##Assiging ward code with index to current station
stations$WardCode=census$WardCode[index1]
head(stations)
```

	Station_ID <int>	Capacity <int>	Latitude <dbl>	Longitude <dbl>	Station_Name <fctr>	WardC <fctr>
1	1	19	51.52916	-0.109970	River Street , Clerkenwell	E05000
2	2	37	51.49961	-0.197574	Phillimore Gardens, Kensington	E05000
3	3	32	51.52128	-0.084605	Christopher Street, Liverpool Street	E05000
4	4	23	51.53006	-0.120973	St. Chad's Street, King's Cross	E05000
5	5	27	51.49313	-0.156876	Sedding Street, Sloane Square	E05000
6	6	18	51.51812	-0.144228	Broadcasting House, Marylebone	E05000

6 rows

Joning Three Datasets:

After getting **WardCode** column in stations table, stations and Census data has been joind on **WardCode** using **merge()** as left outer join to get all data from stations table.

Journeys have column **Start_Station_ID** which has been used to perform right outer join with previously merged data **stations_census** to get all rows from journey table.

Hide

```
#left outer join to get all stations data
station_census=merge(stations,census, by="WardCode", all.x = TRUE)
head(station_census)
```

	WardCode <fctr>	Station_ID <int>	Capacity <int>	Latitude <dbl>	Longitude <dbl>	Station_Name <fctr>	W <f
1	E05000129	65	17	51.52523	-0.135188	Gower Place , Euston	B
2	E05000129	244	18	51.51612	-0.128585	Earnshaw Street , Covent Garden	B
3	E05000129	287	23	51.52367	-0.128377	Bedford Way, Bloomsbury	B
4	E05000129	381	14	51.51953	-0.135777	Charlotte Street, Fitzrovia	B
5	E05000129	69	22	51.52624	-0.134407	Euston Road, Euston	B
6	E05000129	109	56	51.51563	-0.132328	Soho Square , Soho	B

6 rows | 1-8 of 25 columns

Hide

```
# Renamed Start_Station_ID to Station_ID in journey table
names(journeys)[names(journeys) == "Start_Station_ID"] <- "Station_ID"
head(journeys)
```

	Journey_Duration <dbl>	Journey_ID <int>	End_D… <int>	End_Mo… <int>	End_Y… <int>	End_H… <int>	End_Minute <int>	End_Sta
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	

6 rows | 1-9 of 14 columns

Hide

```
#right outer join to get all journeys data
s_j_census=merge(station_census,journeys, by="Station_ID",all.y =TRUE)
head(s_j_census)
```

	Station_ID <int>	WardCode <fctr>	Capacity <int>	Latitude <dbl>	Longitude <dbl>	Station_Name <fctr>	WardNa <fctr>
1	1	E05000370	19	51.52916	-0.10997	River Street , Clerkenwell	Clerkenw
2	1	E05000370	19	51.52916	-0.10997	River Street , Clerkenwell	Clerkenw
3	1	E05000370	19	51.52916	-0.10997	River Street , Clerkenwell	Clerkenw
4	1	E05000370	19	51.52916	-0.10997	River Street , Clerkenwell	Clerkenw
5	1	E05000370	19	51.52916	-0.10997	River Street , Clerkenwell	Clerkenw
6	1	E05000370	19	51.52916	-0.10997	River Street , Clerkenwell	Clerkenw

6 rows | 1-8 of 38 columns

Exploring the data

- The results below shows that bike journeys exists for all locations central, east, south and west except North.
- The journey data is only for two months August and september.
- Journey Dates shows more journeys are made in the start of month to till 20th of the month(first 20 days.)
- Hour of the journeys shows higher number of journeys made during peak hours (7am-9am) and (4pm-6pm)
- Income score does not show any pattern as higher or lower income score does not affect the number of journeys as shows below.

Hide

```
print("location")
```

```
[1] "location"
```

Hide

```
table(s_j_census$NESW)
```

```
Central    East   North   South    West
1209320  132436       0   11546  176938
```

Hide

```
print("Start Month")
```

```
[1] "Start Month"
```

Hide

```
table(s_j_census$Start_Month)
```

```
     8      9
982962 559882
```

Hide

```
print("Start date")
```

```
[1] "Start date"
```

Hide

```
table(s_j_census$Start_Date)
```

```
    1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16
17    18    19    20    21
74895 61442 55409 62455 58596 66934 66848 56189 39146 56136 60240 62243 66341 68115 68753 62292
56411 58092 62888 24986 27433
   22    23    24    25    26    27    28    29    30    31
33015 34134 34245 34779 31058 33186 34409 35014 22002 35158
```

Hide

```
print("Start Hour")
```

```
[1] "Start Hour"
```

Hide

```
table(s_j_census$Start_Hour)
```

```
     0      1      2      3      4      5      6      7      8      9     10     11     12     1
3     14     15     16     17
 15105   9825   6547   4461   3461   6071  27107  80160 150114  89088  61189  68245  81982  8528
9  84999  89450 108629 166065
    18     19     20     21     22     23
153276  96778  60171  39581  32280  22971
```

Hide

```
print("Income Score")
```

```
[1] "Income Score"
```

Hide

```
table(s_j_census$IncomeScor)
```

```
  0.01   0.03   0.05   0.06   0.07   0.08   0.09    0.1   0.11   0.12   0.14   0.15   0.16   0.1
7   0.18   0.19    0.2   0.21
 50555  22346  83821  97372  39989  20241  69174  31978 125933  60937  41525  14703  83744  6011
9  71881  41649  27976 119756
  0.22   0.23   0.24   0.25   0.27   0.29    0.3   0.31   0.32   0.33   0.34   0.35   0.36   0.3
7   0.39    0.4   0.42   0.44
 32610  27815  66783  37788  11140  23029   4275  15805  43843  45607  45838  40927  32600  1277
5  12429   3484   6037   3756
```

Hypothesis

The problem is to predict the number of bikes demand on a station during one hour slot.

H1: Higher bikes demand on stations which are in densely populated area.

H2: The stations in poor areas have higher demand of bikes.

H3: The number of bike demand is higher during peak hours on weekdays.

H4: The higher the ratio of employed people in an area has higher bike demand.

H5: The higher the uk born people in an area, have higher demand for bikes.

H6: There is higher demands of bikes on station in the start of month.

Metrics

The following metrice have been created to support and falsify the hypothesis.

1. **Greenspace (H1):** It is the percentage of area containing greenspace. Lower percentage means the area is more densely populated.
2. **IncomeScore (H2):** The proportion of area suffering low income. Higher value means the area is more poor.
3. **hour(H3):** The hour will be calculated after filtering the peakhours (7am-9am and 4pm-6pm) and weekdays(Mon,Tue, Wed, Thur, Fri) from start_hour, Start_month, Start_year from journeys data.
4. **Ratio employed(H4):** The ratio of people who are employed and have a profession. Ratioemployed=NoEmployee/PopDen*AreaSqKm. Higher value means more people are employed.
5. **ctf_h_ratio(H2):** NoCTFtoH/Nodwelling .The number of properties under council. To find out the poor area.

6. **No_owned_dwel_ratio(H2):** The ratio of properties owned by people in the area. NoOwnDwel/NoDwelling . To find the ratio of people not owning a property as poor areas.
7. **uk_born(H5):** The ratio of people born in uk is calculated as uk_born=uk_born/(uk_born+non_uk_born).
8. **Month_date(H6):** It will contain month start dates 1-20 date by filtering month column.

Hide

```
final=data.frame(s_ID=s_j_census$Station_ID,year=s_j_census$Start_Year,date=s_j_census$Start_Dat
e,month=s_j_census$Start_Month, hour=s_j_census$Start_Hour,Ratioemployed=s_j_census$NoEmployee/
(s_j_census$PopDen*s_j_census$AreaSqKm),g_space=s_j_census$GrenSpace,ctf_h_ratio=s_j_census$NoCT
FtoH/(s_j_census$NoDwelling),no_owned_dwel_ratio=s_j_census$NoOwndDwel/s_j_census$NoDwelling,uk_
b=(s_j_census$BornUK/(s_j_census$BornUK+s_j_census$NotBornUK)),i_score=s_j_census$IncomeScor,Jou
rneysid=s_j_census$Journey_ID)

head(final)
```

	s… <int>	y… <int>	d… <int>	mo… <int>	h… <int>	Ratioemployed <dbl>	g_spa… <dbl>	ctf_h_ratio <dbl>	no_owned_dwel_ratio <dbl>
1	1	17	21	8	9	3.817385	9.3	0.004238036	0.2763553
2	1	17	12	8	19	3.817385	9.3	0.004238036	0.2763553
3	1	17	17	8	10	3.817385	9.3	0.004238036	0.2763553
4	1	17	31	8	8	3.817385	9.3	0.004238036	0.2763553
5	1	17	18	8	18	3.817385	9.3	0.004238036	0.2763553
6	1	17	1	8	14	3.817385	9.3	0.004238036	0.2763553

6 rows | 1-10 of 12 columns

Hide

```
NA
```

Weekday calculation

- Changed the format of **year** as added year 2017 in the data to replace 17.
- Combined column year, month and date to get data in format such as yyyy-mm-dd so that weekdays can be calculated and extracted.
- **as.Date()** function has been used to create date format and added new column in the data.
- Then day has been calculated using **format()** and **as.date()** function and new column added with weekdays. It gave Mon, Tue, Wed,Thu, Friday, Sat, Sun

Hide

```
#changed year format from '17' to '2017'
final$new_year=final$year[final$year<17] <- 2017
final$year=NULL

# merging year,month and date column
final$date_mon_year <- as.Date(with(final, paste(new_year, month, date,sep="-")), "%Y-%m-%d")

# calculating days of the week from 'date'
final$weekday <- format(as.Date(final$date_mon_year), "%a")
head(final)
```

	s… <int>	d… <int>	mo… <int>	h… <int>	Ratioemployed <dbl>	g_sp… <dbl>	ctf_h_ratio <dbl>	no_owned_dwel_ratio <dbl>	uk_b <dbl>
1	1	21	8	9	3.817385	9.3	0.004238036	0.2763553	0.6145344
2	1	12	8	19	3.817385	9.3	0.004238036	0.2763553	0.6145344
3	1	17	8	10	3.817385	9.3	0.004238036	0.2763553	0.6145344
4	1	31	8	8	3.817385	9.3	0.004238036	0.2763553	0.6145344
5	1	18	8	18	3.817385	9.3	0.004238036	0.2763553	0.6145344
6	1	1	8	14	3.817385	9.3	0.004238036	0.2763553	0.6145344

6 rows | 1-10 of 14 columns

Hide

```
NA
```

Filtering days of month and weekdays

- Used **dplyr** library and **filter()** function to filter days of the month and week
- first 20 days of month has been filtered as there is more journeys made in the start of month
- Filterin weekdays
- Filtered for peak hours which are 7am-9am and 4pm-6pm

Hide

```
library(dplyr)
#filtering first 20 days of month
target <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
new=data.frame(filter(final, date %in% target))

#filtering weekdays
target <- c("Mon","Tue","Wed","Thu","Fri")
new1=data.frame(filter(new, weekday %in% target))

# filtering peakhours
target <- c(7,8,9,16,17,18)
new2=data.frame(filter(new1, hour %in% target))
head(new2)
```

	s… <int>	d… <int>	mo… <int>	h… <int>	Ratioemployed <dbl>	g_sp… <dbl>	ctf_h_ratio <dbl>	no_owned_dwel_ratio <dbl>	uk_b <dbl>
1	1	18	8	18	3.817385	9.3	0.004238036	0.2763553	0.6145344
2	1	2	8	8	3.817385	9.3	0.004238036	0.2763553	0.6145344
3	1	19	9	8	3.817385	9.3	0.004238036	0.2763553	0.6145344
4	1	4	8	16	3.817385	9.3	0.004238036	0.2763553	0.6145344
5	1	3	8	18	3.817385	9.3	0.004238036	0.2763553	0.6145344
6	1	15	8	18	3.817385	9.3	0.004238036	0.2763553	0.6145344

6 rows | 1-10 of 14 columns

Grouping data

Grouped the data based on all metrics and counts the number of bikes journeys made in one hour on a particular day using **groupby**, **summarize** and **count** funtion.

Hide

```
# grouping and counting the number of bikes journeys made
k=new2 %>%
group_by(date,hour,Ratioemployed,g_space,ctf_h_ratio,no_owned_dwel_ratio,uk_b,i_score,weekday) %
>%
summarise(journeysCount=n())
k
```

d… <int>	h… <int>	Ratioemployed <dbl>	g_sp… <dbl>	ctf_h_ratio <dbl>	no_owned_dwel_ratio <dbl>	uk_b <dbl>	i_score <dbl>	we <ch
1	7	0.1321321	22.6	2.868206e-05	0.1993403	0.5524905	0.32	Fri
1	7	0.1321321	22.6	2.868206e-05	0.1993403	0.5524905	0.32	Tue
1	7	0.1503759	20.9	7.764249e-04	0.2572790	0.6599652	0.31	Fri

d… <int>	h… <int>	Ratioemployed <dbl>	g_sp… <dbl>	ctf_h_ratio <dbl>	no_owned_dwel_ratio <dbl>	uk_b <dbl>	i_score <dbl>	we <ch
1	7	0.1503759	20.9	7.764249e-04	0.2572790	0.6599652	0.31	Tue
1	7	0.1558442	12.5	1.064916e-03	0.2978848	0.6034640	0.29	Fri
1	7	0.1558442	12.5	1.064916e-03	0.2978848	0.6034640	0.29	Tue
1	7	0.1590214	21.2	9.009009e-04	0.2474012	0.6017983	0.42	Fri
1	7	0.1590214	21.2	9.009009e-04	0.2474012	0.6017983	0.42	Tue
1	7	0.1616162	13.2	1.313321e-03	0.2650094	0.5509237	0.23	Fri
1	7	0.1616162	13.2	1.313321e-03	0.2650094	0.5509237	0.23	Tue

1-10 of 16,588 rows | 1-9 of 10 columns Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
NA
```

Hide

```
for(unique_value in unique(k$weekday)){
k[paste("weekday", unique_value, sep = ".")] <- ifelse(k$weekday == unique_value, 1, 0)
}
head(k)
```

d… <int>	h… <int>	Ratioemployed <dbl>	g_sp… <dbl>	ctf_h_ratio <dbl>	no_owned_dwel_ratio <dbl>	uk_b <dbl>	i_score <dbl>	we <ch
1	7	0.1321321	22.6	2.868206e-05	0.1993403	0.5524905	0.32	Fri
1	7	0.1321321	22.6	2.868206e-05	0.1993403	0.5524905	0.32	Tue
1	7	0.1503759	20.9	7.764249e-04	0.2572790	0.6599652	0.31	Fri
1	7	0.1503759	20.9	7.764249e-04	0.2572790	0.6599652	0.31	Tue
1	7	0.1558442	12.5	1.064916e-03	0.2978848	0.6034640	0.29	Fri
1	7	0.1558442	12.5	1.064916e-03	0.2978848	0.6034640	0.29	Tue

6 rows | 1-9 of 15 columns

Removed Unneccessary columns

Removed unnecessary columns which are not metrics and displaying final metrices as below.

Hide

```
#deleting unecessary columns
k$weekday=NULL
k$weekday.Mon=NULL
head(k)
```

d… <int>	h… <int>	Ratioemployed <dbl>	g_sp… <dbl>	ctf_h_ratio <dbl>	no_owned_dwel_ratio <dbl>	uk_b <dbl>	i_score <dbl>	jo
1	7	0.1321321	22.6	2.868206e-05	0.1993403	0.5524905	0.32	
1	7	0.1321321	22.6	2.868206e-05	0.1993403	0.5524905	0.32	
1	7	0.1503759	20.9	7.764249e-04	0.2572790	0.6599652	0.31	
1	7	0.1503759	20.9	7.764249e-04	0.2572790	0.6599652	0.31	
1	7	0.1558442	12.5	1.064916e-03	0.2978848	0.6034640	0.29	
1	7	0.1558442	12.5	1.064916e-03	0.2978848	0.6034640	0.29	

6 rows | 1-9 of 13 columns

Checking Missing values

Summary() gives idea of the metrices whether normalized or no. It also displays whether we have missing values or not.

Hide

```
summary(k)
```

```
      date             hour       Ratioemployed        g_space       ctf_h_ratio      no_owned_d
wel_ratio      uk_b
 Min.   : 1.000   Min.   : 7.00   Min.   : 0.1321   Min.   : 0.00   Min.   :0.00003   Min.   :0.
1380      Min.   :0.3543
 1st Qu.: 5.000   1st Qu.: 8.00   1st Qu.: 0.3333   1st Qu.: 8.90   1st Qu.:0.00145   1st Qu.:0.
2408      1st Qu.:0.4909
 Median :10.000   Median : 9.00   Median : 0.5333   Median :13.50   Median :0.00382   Median :0.
2859      Median :0.5621
 Mean   : 9.941   Mean   :12.44   Mean   : 1.9421   Mean   :18.08   Mean   :0.00513   Mean   :0.
3015      Mean   :0.5499
 3rd Qu.:15.000   3rd Qu.:17.00   3rd Qu.: 1.3617   3rd Qu.:26.90   3rd Qu.:0.00770   3rd Qu.:0.
3574      3rd Qu.:0.6096
 Max.   :19.000   Max.   :18.00   Max.   :50.5540   Max.   :69.10   Max.   :0.01794   Max.   :0.
5476      Max.   :0.7112
                                   NA's   :162       NA's   :162     NA's   :162       NA's   :16
2         NA's   :162
    i_score       journeysCount     weekday.Fri     weekday.Tue      weekday.Wed      weekday.Th
u
 Min.   :0.0100   Min.   :  1.00   Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0
000
 1st Qu.:0.1100   1st Qu.:  8.00   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0
000
 Median :0.1900   Median : 16.00   Median :0.000   Median :0.0000   Median :0.0000   Median :0.0
000
 Mean   :0.1968   Mean   : 28.69   Mean   :0.223   Mean   :0.2235   Mean   :0.1815   Mean   :0.1
863
 3rd Qu.:0.2900   3rd Qu.: 32.00   3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0
000
 Max.   :0.4400   Max.   :494.00   Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0
000
 NA's   :162
```

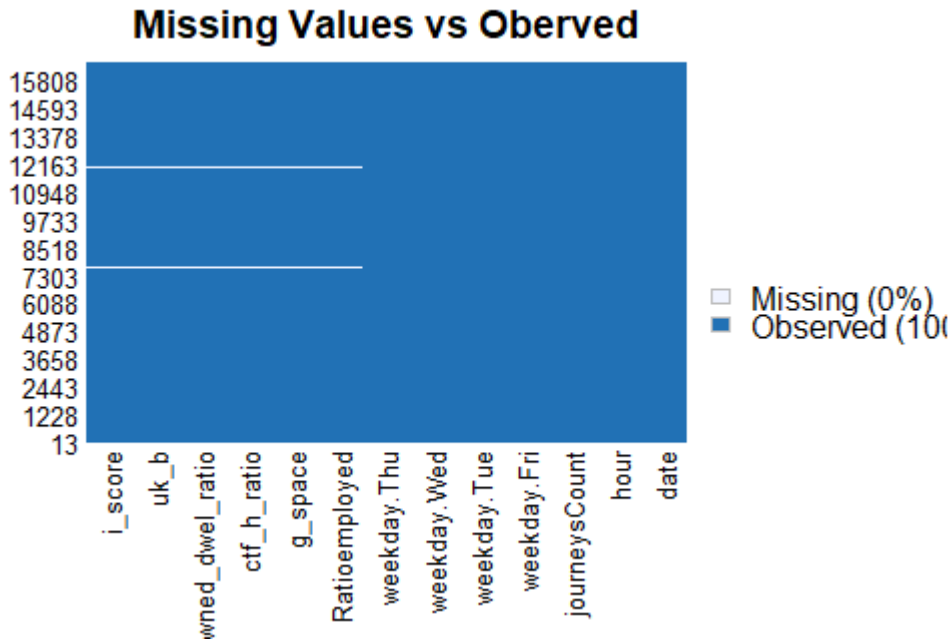
Removing missing values

Removing the NA's as we got after joing three tables.

Hide

```
library(Amelia)
missmap(k,main="Missing Values vs Oberved")
```

```
Unknown or uninitialised column: 'arguments'.Unknown or uninitialised column: 'arguments'.Unknow
n or uninitialised column: 'imputations'.
```



Hide

```
k=na.omit(k)
summary(k)
```

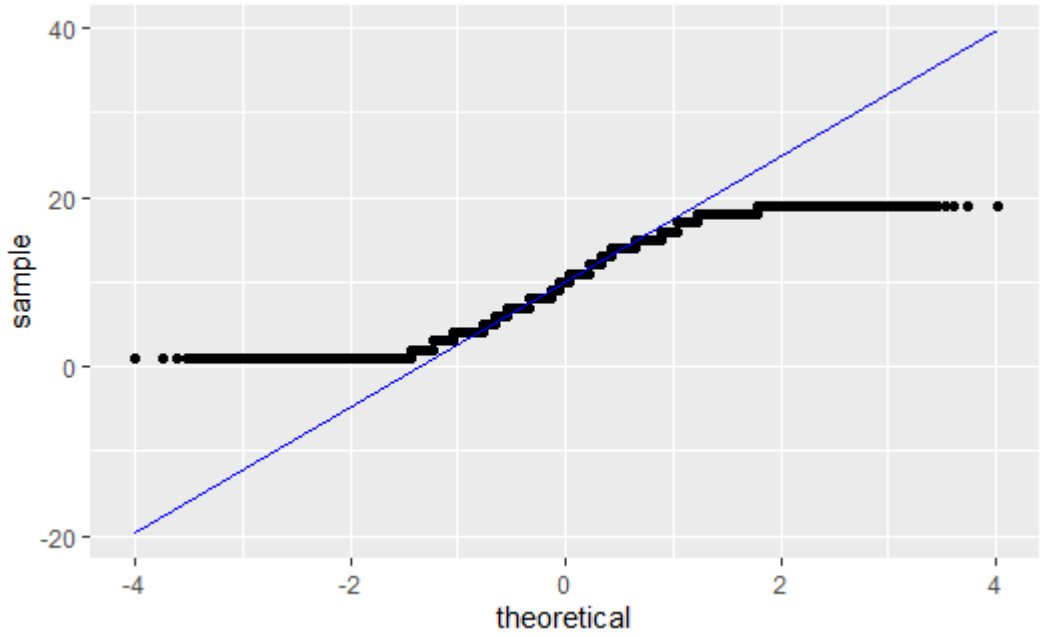
```
      date             hour       Ratioemployed        g_space       ctf_h_ratio        no_owned
_dwel_ratio      uk_b
 Min.   : 1.000   Min.   : 7.00   Min.   : 0.1321   Min.   : 0.00   Min.   :2.661e-05   Min.   :
0.1380      Min.   :0.3543
 1st Qu.: 5.000   1st Qu.: 8.00   1st Qu.: 0.3333   1st Qu.: 8.90   1st Qu.:1.450e-03   1st Qu.:
0.2408      1st Qu.:0.4909
 Median :10.000   Median : 9.00   Median : 0.5333   Median :13.50   Median :3.815e-03   Median :
0.2859      Median :0.5621
 Mean   : 9.941   Mean   :12.44   Mean   : 1.9421   Mean   :18.08   Mean   :5.131e-03   Mean   :
0.3015      Mean   :0.5499
 3rd Qu.:15.000   3rd Qu.:17.00   3rd Qu.: 1.3617   3rd Qu.:26.90   3rd Qu.:7.703e-03   3rd Qu.:
0.3574      3rd Qu.:0.6096
 Max.   :19.000   Max.   :18.00   Max.   :50.5540   Max.   :69.10   Max.   :1.794e-02   Max.   :
0.5476      Max.   :0.7112
    i_score       journeysCount    weekday.Fri     weekday.Tue      weekday.Wed      weekday.Thu
 Min.   :0.0100   Min.   :  1.0   Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00
00
 1st Qu.:0.1100   1st Qu.:  8.0   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00
00
 Median :0.1900   Median : 16.0   Median :0.000   Median :0.0000   Median :0.0000   Median :0.00
00
 Mean   :0.1968   Mean   : 28.8   Mean   :0.223   Mean   :0.2235   Mean   :0.1814   Mean   :0.18
64
 3rd Qu.:0.2900   3rd Qu.: 32.0   3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00
00
 Max.   :0.4400   Max.   :494.0   Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00
00
```

Normality Test

To check whether data is normally distributed or not. **ggplot2** library has been used to plot Q-Q plot and considered best case when distribution is with 45 degree line. As it can be clearly seen from below plots that most of the metrices points are not on line. There is need to perform transformation to make it normalized.

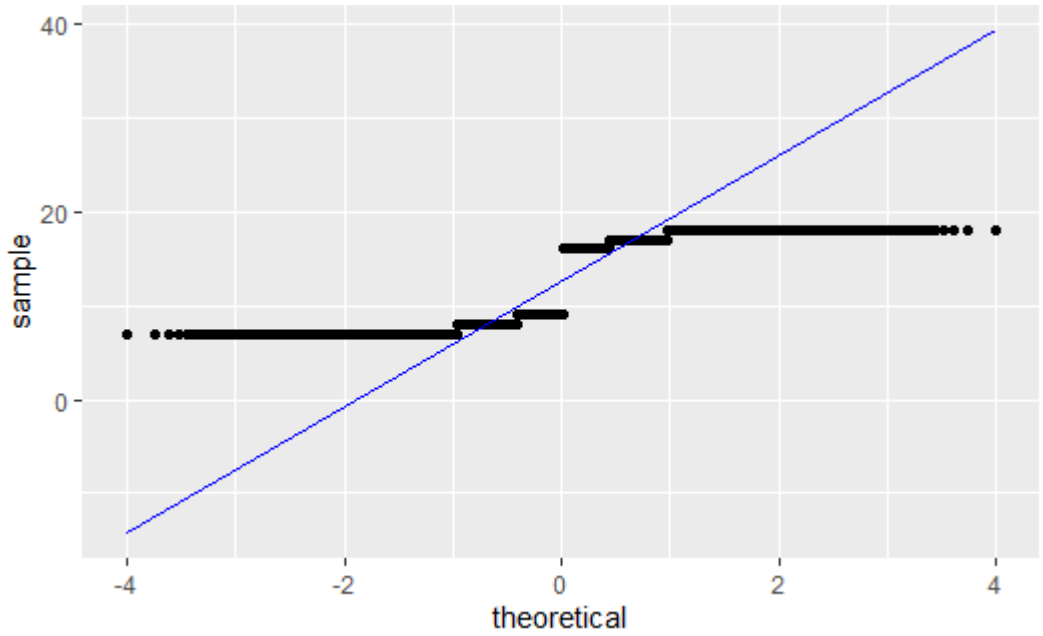
Hide

```
library(ggplot2)

ggplot(k, aes(sample=date)) + stat_qq() + stat_qq_line(col="blue")
```



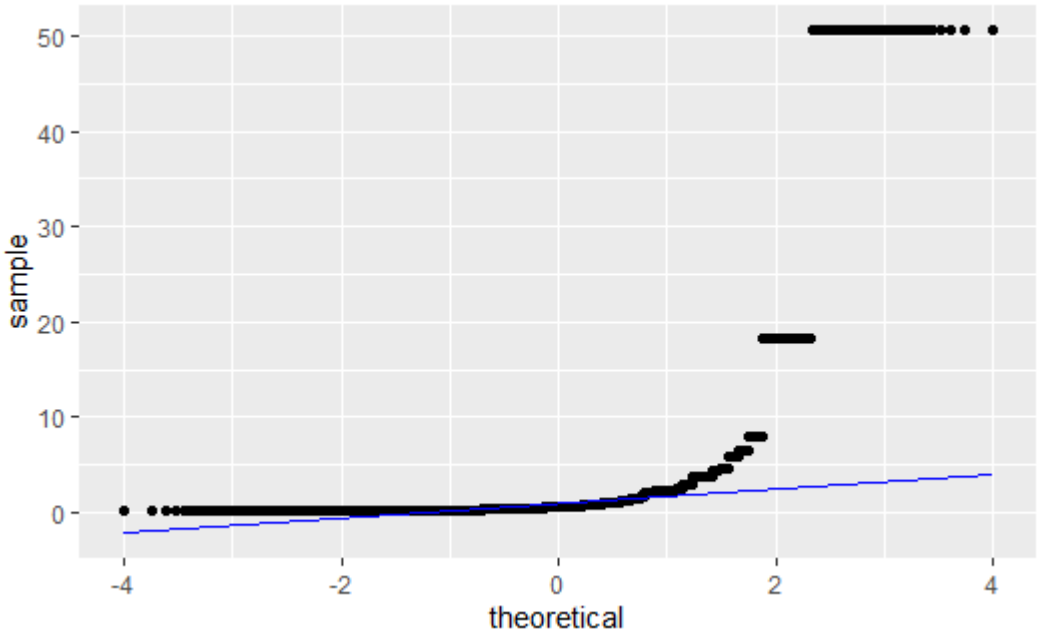
Hide

```
ggplot(k, aes(sample=hour)) + stat_qq() + stat_qq_line(col="blue")
```



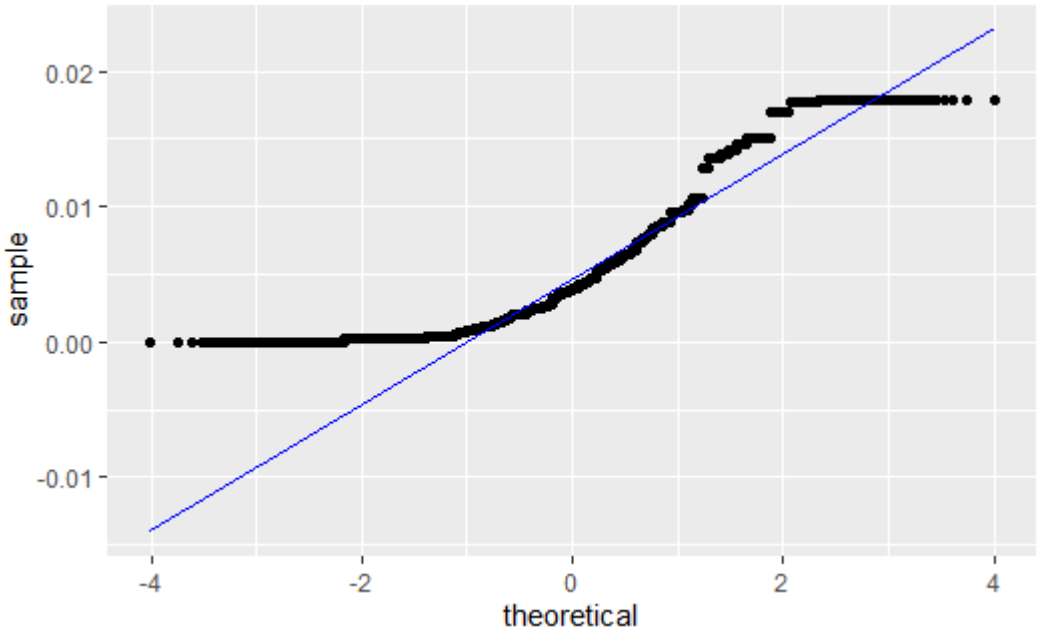
Hide

```
ggplot(k, aes(sample=Ratioemployed)) + stat_qq() + stat_qq_line(col="blue")
```



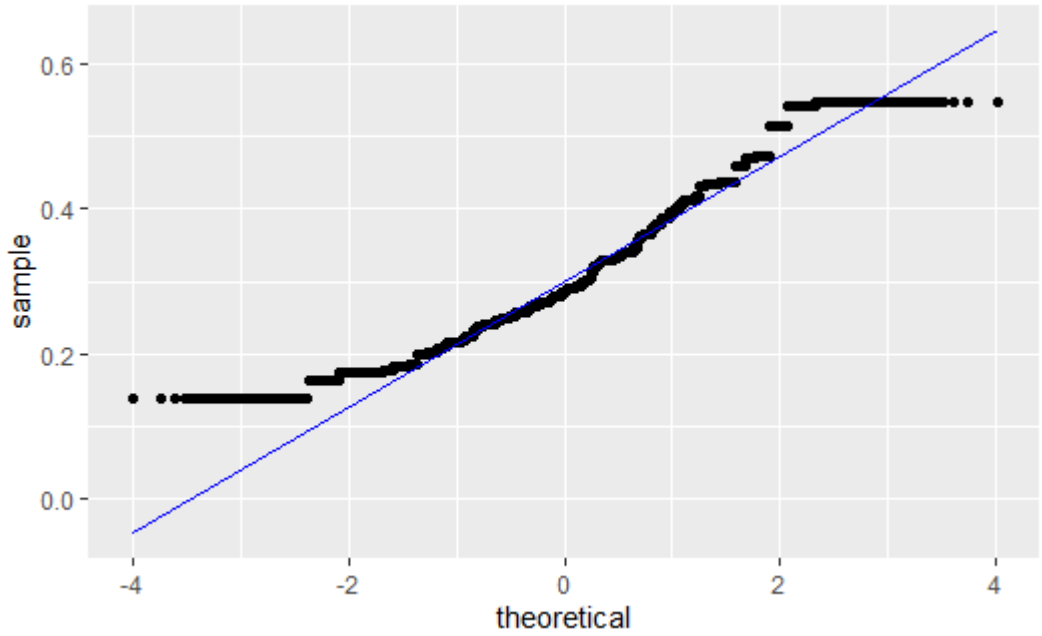
Hide

```
ggplot(k, aes(sample=ctf_h_ratio)) + stat_qq() + stat_qq_line(col="blue")
```



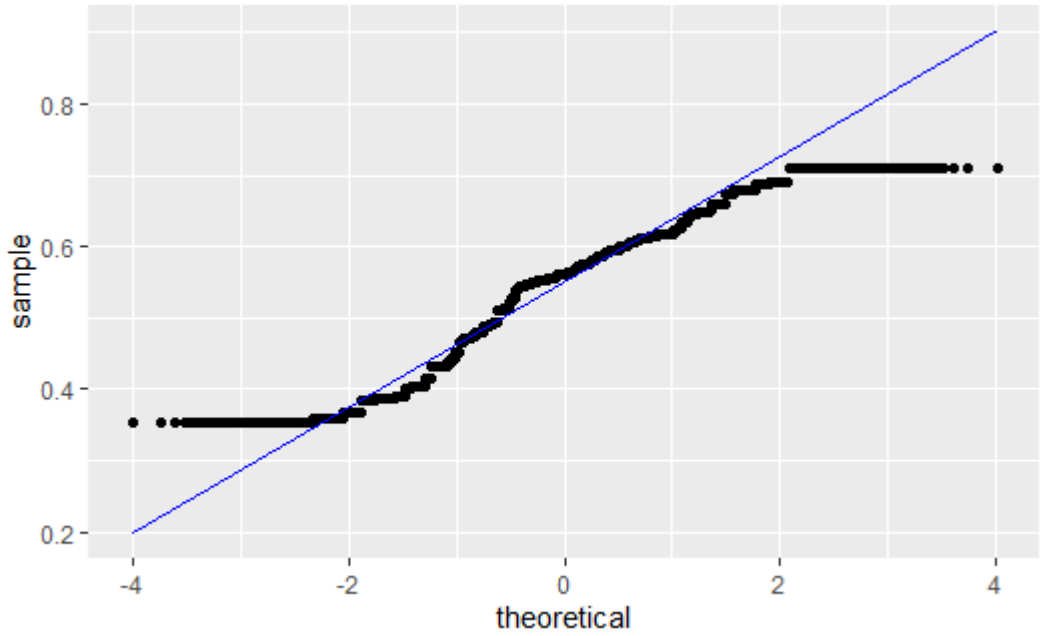
Hide

```
ggplot(k, aes(sample=no_owned_dwel_ratio)) + stat_qq() + stat_qq_line(col="blue")
```



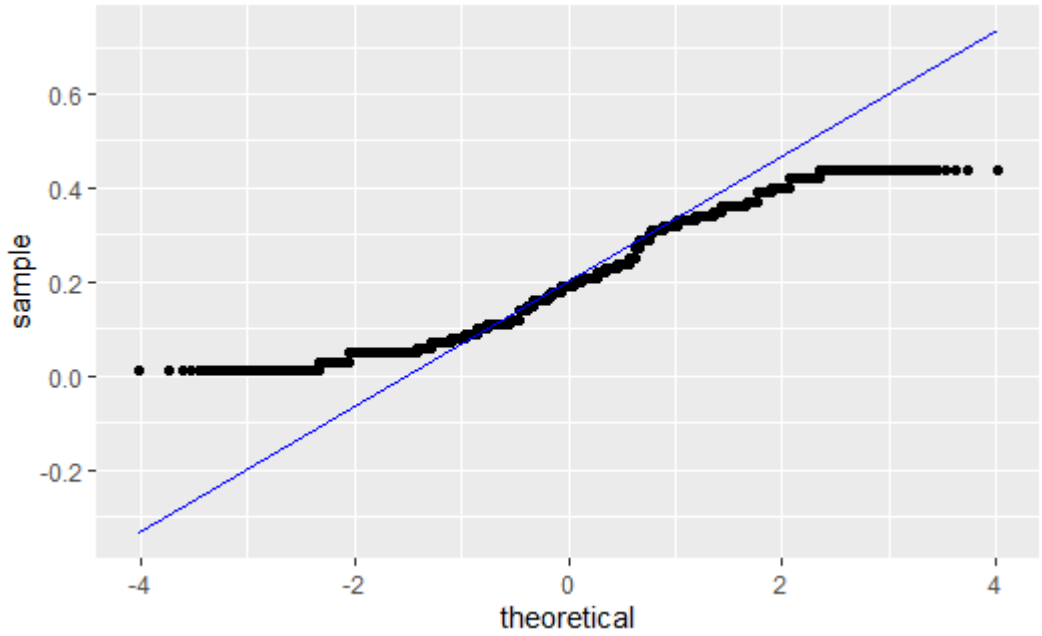
Hide

```
ggplot(k, aes(sample=uk_b)) + stat_qq() + stat_qq_line(col="blue")
```



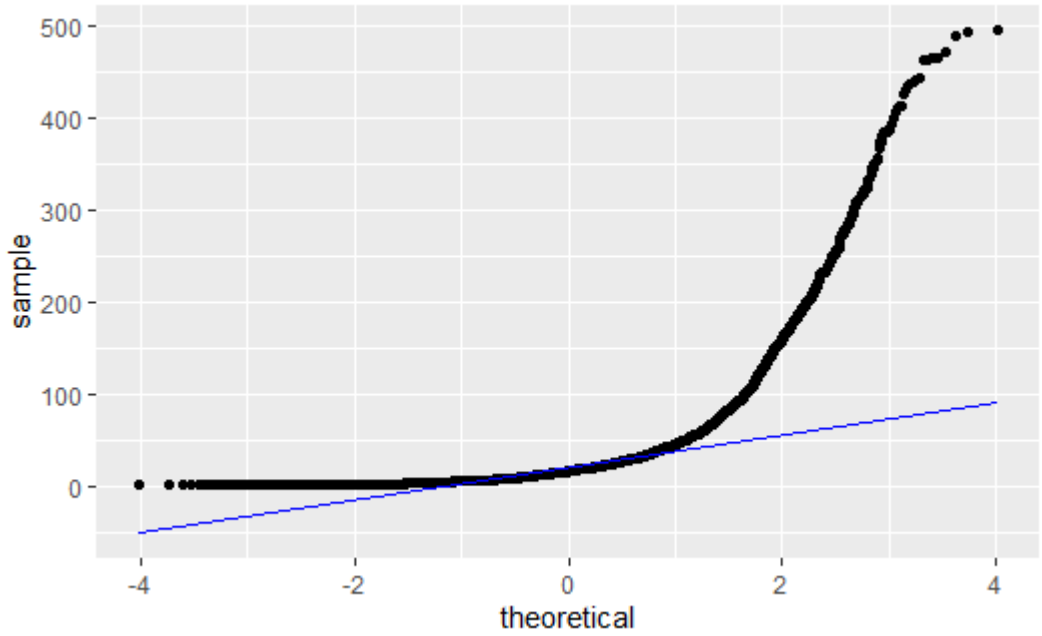
Hide

```
ggplot(k, aes(sample=i_score)) + stat_qq() + stat_qq_line(col="blue")
```



Hide

```
ggplot(k, aes(sample=journeysCount)) + stat_qq() + stat_qq_line(col="blue")
```

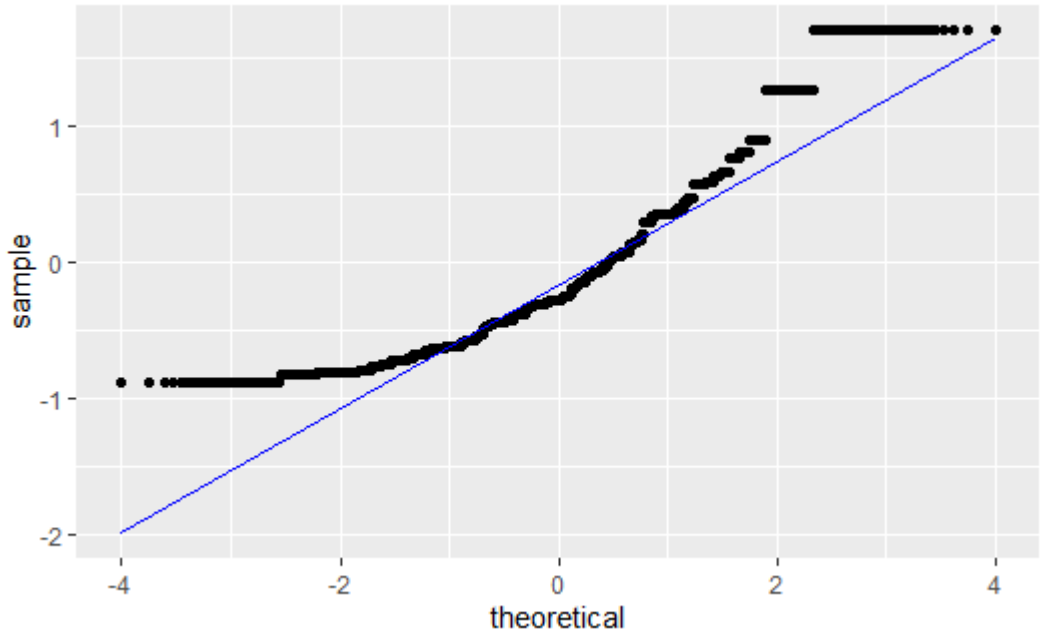


Log Transformation

After normality test, log transformation have been applied on metrics.It also display Q-Q plot after applying transformation

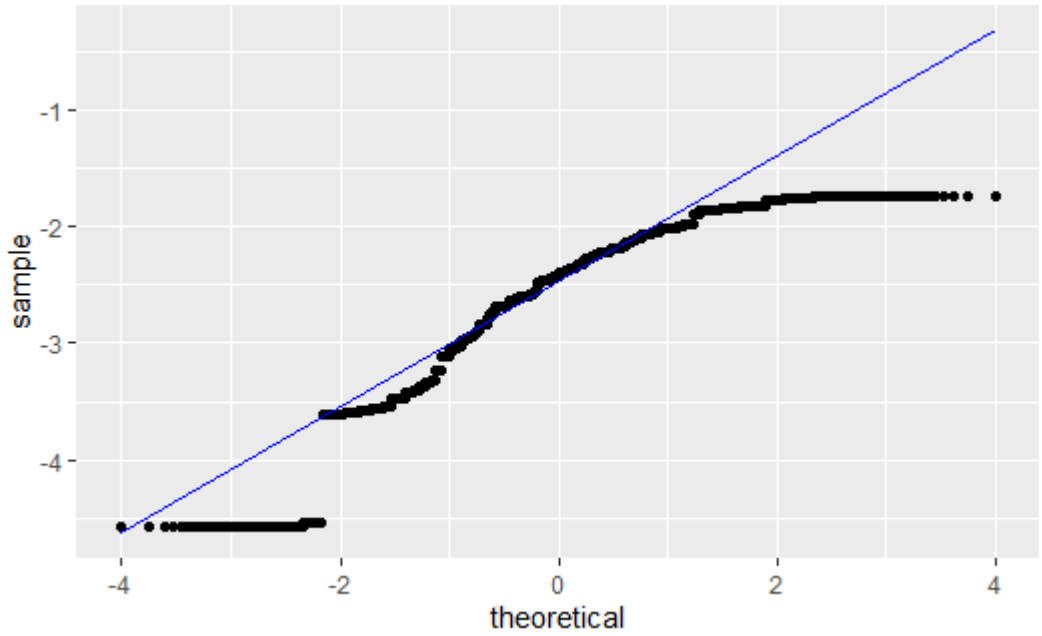
Hide

```
#log transformation
k$Ratioemployed = log10(k$Ratioemployed)
#plotting
ggplot(k, aes(sample=Ratioemployed)) + stat_qq() + stat_qq_line(col="blue")
```



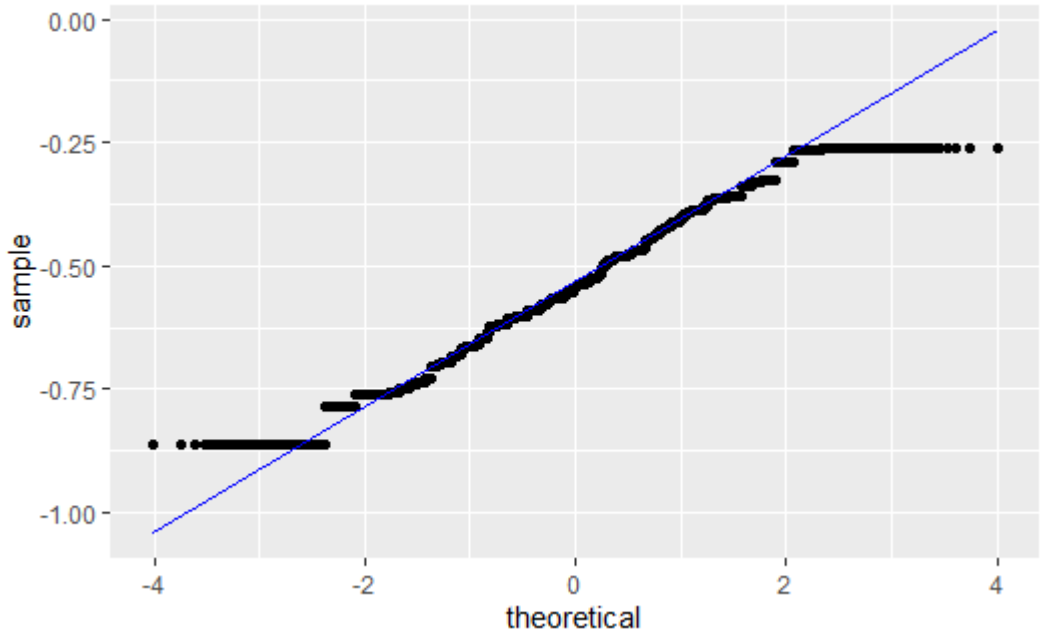
Hide

```
k$ctf_h_ratio = log10(k$ctf_h_ratio)
ggplot(k, aes(sample=ctf_h_ratio)) + stat_qq() + stat_qq_line(col="blue")
```



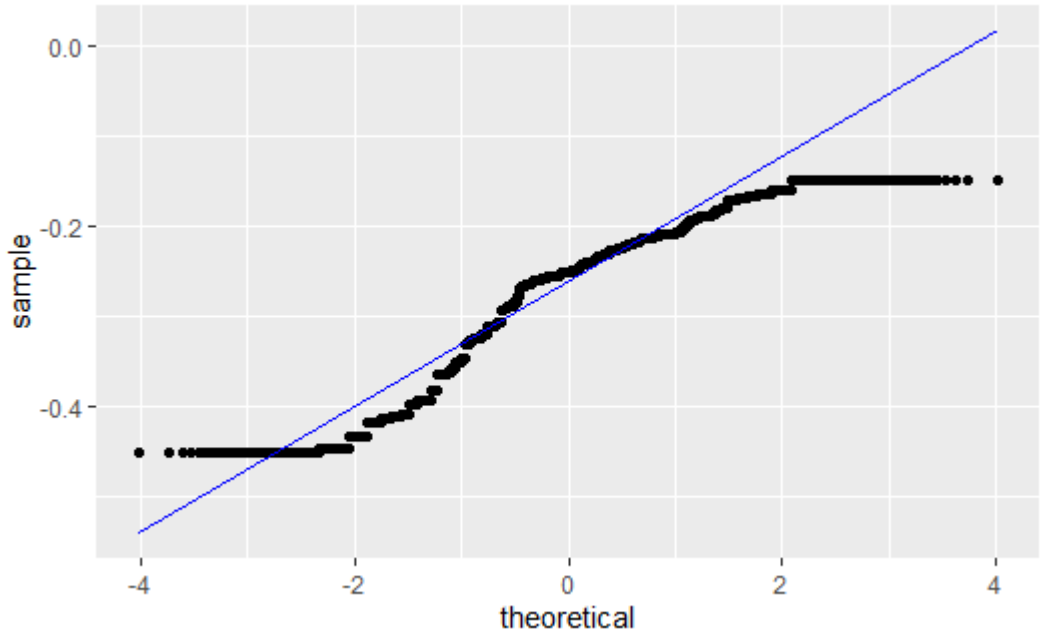
Hide

```
k$no_owned_dwel_ratio = log10(k$no_owned_dwel_ratio)
ggplot(k, aes(sample=no_owned_dwel_ratio)) + stat_qq() + stat_qq_line(col="blue")
```



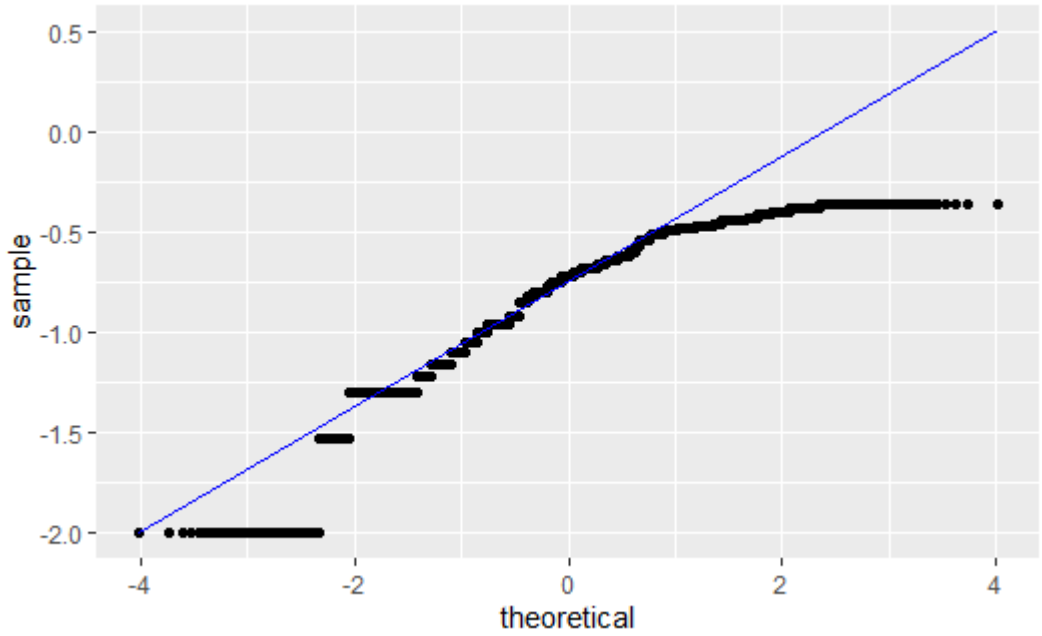
Hide

```
k$uk_b = log10(k$uk_b)
ggplot(k, aes(sample=uk_b)) + stat_qq() + stat_qq_line(col="blue")
```



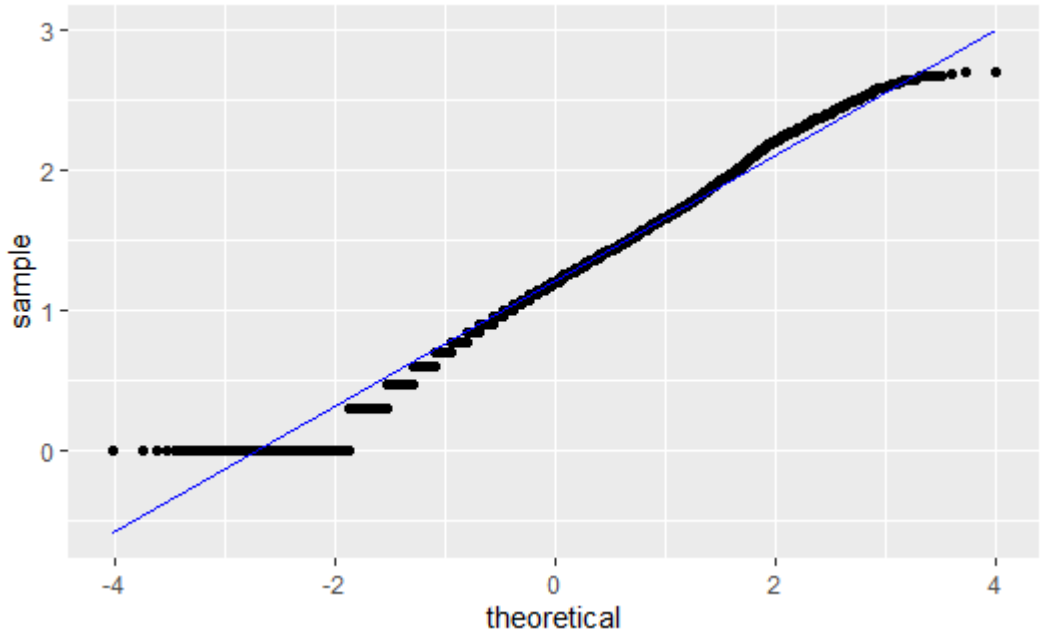
Hide

```
k$i_score = log10(k$i_score)
ggplot(k, aes(sample=i_score)) + stat_qq() + stat_qq_line(col="blue")
```



Hide

```
k$journeysCount = log10(k$journeysCount)
ggplot(k, aes(sample=journeysCount)) + stat_qq() + stat_qq_line(col="blue")
```

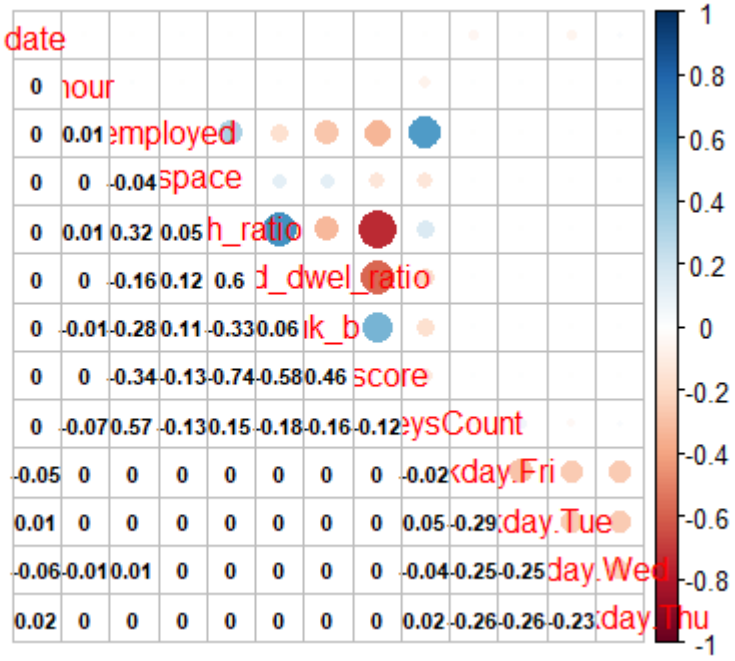


Correlation Matrix

- Finding Correlation matrix to check whether metrices are independet to each other or not.
- The plot below shows **ctf_h_ratio** have positive correlation with **no_owned_dwel_ratio** and **ratio_employed** because they both giving same information. So, will remove **ctf_h_ratio**.
- Secondly, **income_score** have correlation with **uk_born**. Will remove income score
- jouneysCount have correlation with **Ratio_employed**

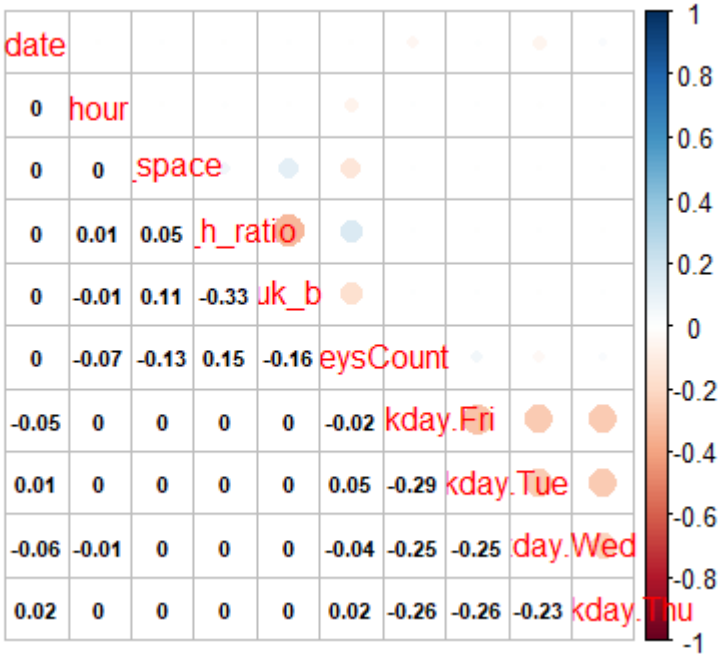
Hide

```
library(corrplot)
cor1 = cor(k)
corrplot.mixed(cor1, lower.col = "black", number.cex = .7)
```



Hide

```
k$Ratioemployed=NULL
k$i_score=NULL
k$no_owned_dwel_ratio=NULL #removed one day as applied one hot encoding
library(corrplot)
cor1 = cor(k)
corrplot.mixed(cor1, lower.col = "black", number.cex = .7)
```



Hide

k

d… <int>	h… <int>	g_sp… <dbl>	ctf_h_ratio <dbl>	uk_b <dbl>	journeysCount <dbl>	weekday.Fri <dbl>	weekday.Tue <dbl>	weekday
1	7	22.6	-4.542390	-0.2576752	0.0000000	1	0	
1	7	22.6	-4.542390	-0.2576752	0.4771213	0	1	
1	7	20.9	-3.109901	-0.1804789	0.7781513	1	0	
1	7	20.9	-3.109901	-0.1804789	0.8450980	0	1	
1	7	12.5	-2.972685	-0.2193486	1.3979400	1	0	
1	7	12.5	-2.972685	-0.2193486	1.5314789	0	1	
1	7	21.2	-3.045323	-0.2205491	1.1760913	1	0	
1	7	21.2	-3.045323	-0.2205491	1.3222193	0	1	
1	7	13.2	-2.881629	-0.2589085	1.3802112	1	0	
1	7	13.2	-2.881629	-0.2589085	1.4471580	0	1	

1-10 of 16,426 rows | 1-9 of 10 columns Previous **1** 2 3 4 5 6 … 100 Next

Algorithm

Training and applying linear regression

- In this step, data has been splitted into 75% training and 25% testing data.
- Then, data standardization have been performed on train and test data.
- Linear regression model has been applied to predict the number of bikes demand for an hour on a station.

Hide

```
sample_size<-floor(0.75*nrow(k))
set.seed(123)
#training and testing data splitting
trainIdx<-sample(seq_len(nrow(k)),size=sample_size)
train_s = k[trainIdx,]
test_s = k[-trainIdx,]

# data standardization
train= as.data.table( scale(train_s) )
test= as.data.table( scale(test_s) )

# Linear regression model with journeysCount as target variable
lr = lm(formula=journeysCount ~., data=train)
train_preds = predict(lr, train)
test_preds = predict(lr, test) #

print( paste("R-square on train:", cor(train_preds, train$journeysCount)^2))
```

```
[1] "R-square on train: 0.0631860262708537"
```

R2 for Test/Train data

- For train data R2 value is 0.063 which is very low which means the proposd model is very bad fit. For good fit, R2>0.70 .
- For test data, R2=0.062 which also show the proposed model is bad fit.

Hide

```
print( paste("R-square on test:", cor(test_preds, test$journeysCount)^2))
```

```
[1] "R-square on test: 0.0625492730336925"
```

Residual Summary

summary() gives the residual summary, The value of median should be zero. In the proposed model it is zero.The residuals is the difference between actual value and predicted value by the model.

1. **Significant Coefficient**

The summary also telling about each beta coefficient and which metric is significant or not. The sign *** next to metric show significancy of a metric. **hour**, **g_space**, **ctf_h_ratio**, **uk_b**, **tue** weekdays shows high significance as their p value is very low and <0.05.However, **date**,**month** and two weekdays (Wed, thur,Fri) are less significant as the value of p-value>0.05.

2. **Residual Standard Error(RSE)**

RSE value shows whether data fits the model perfectly or not. Closer to zero is best. But, in the proposed case value is near to 1 which means the data does not fit the model at all.

3. **Adjusted R-squared**

Multiple R-squared gives the proportion of variation explained by target variable in the model. It increases with increase in the number of predictors. So R-squared is adjusted to avoid this affect. However, the proposed model shows only around 6% variability have been explained by target variable in the model.

4. **Null Hypothesis(F-statistics)**

The low value of the F-statistic with low p-values <0.05 means null hypothesis is true. **Null Hypothesis** says predictors and target are independent of each other. However, the proposed model accepts null hypothesis.

5. **p-value: < 2.2e-16**

The p-value for almost all coefficients is <0.05 which means model is significant. The **Weekday.Wed**, **weekday.thu** and **Weekday.Fri** coefficients are less significant.

Hide

```
summary(lr)
```

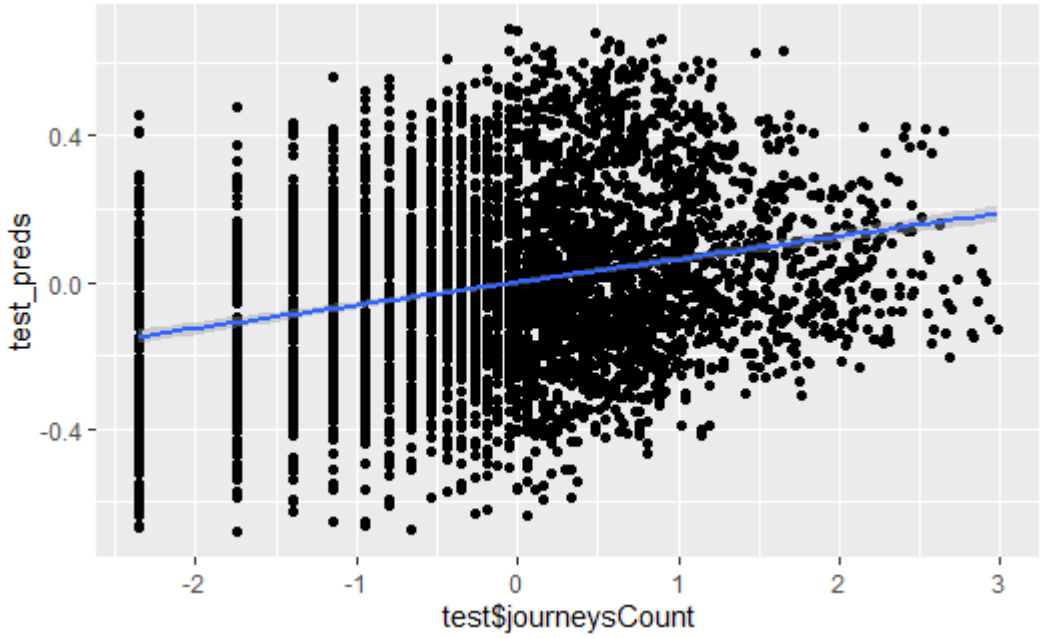
```

Call:
lm(formula = journeysCount ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.9500 -0.6260  0.0258  0.6287  3.1749 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.873e-17  8.724e-03   0.000   1.0000    
date        -8.551e-03  8.763e-03  -0.976   0.3292    
hour        -7.041e-02  8.725e-03  -8.070 7.69e-16 ***
g_space     -1.300e-01  8.822e-03 -14.738  < 2e-16 ***
ctf_h_ratio  1.312e-01  9.278e-03  14.138  < 2e-16 ***
uk_b        -1.017e-01  9.321e-03 -10.908  < 2e-16 ***
weekday.Fri -1.169e-02  1.145e-02  -1.021   0.3071    
weekday.Tue  4.697e-02  1.146e-02   4.100 4.16e-05 ***
weekday.Wed -2.587e-02  1.117e-02  -2.317   0.0205 *  
weekday.Thu  2.341e-02  1.118e-02   2.095   0.0362 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9682 on 12309 degrees of freedom
Multiple R-squared:  0.06319,   Adjusted R-squared:  0.0625 
F-statistic: 92.25 on 9 and 12309 DF,  p-value: < 2.2e-16
```

Testing prediction plot:

The plot below shows the difference between the actual and predicted values by model. All the points should lie around linear line. However, it shows random data there is not linear pattern.

Hide

```
library(ggplot2)
ggplot(test, aes(x=test$journeysCount, y=test_preds)) + geom_point() + geom_smooth(method = "lm"
)
```



Requirementsfor Linear Regression

1. **The Independece of Residuals:**

The first plot shows whether correlation exists between residuals or not. The Residuals points should be random. However,the proposed model shows there is some pattern in residuals.

2. **Normality of Residuals**

The second plot is q-Q plot which means whether normal distribution exists between residuals. The residuals are almost following the straight line which shows the normal distribution in the residuals.

3. **Homoscedasticity**

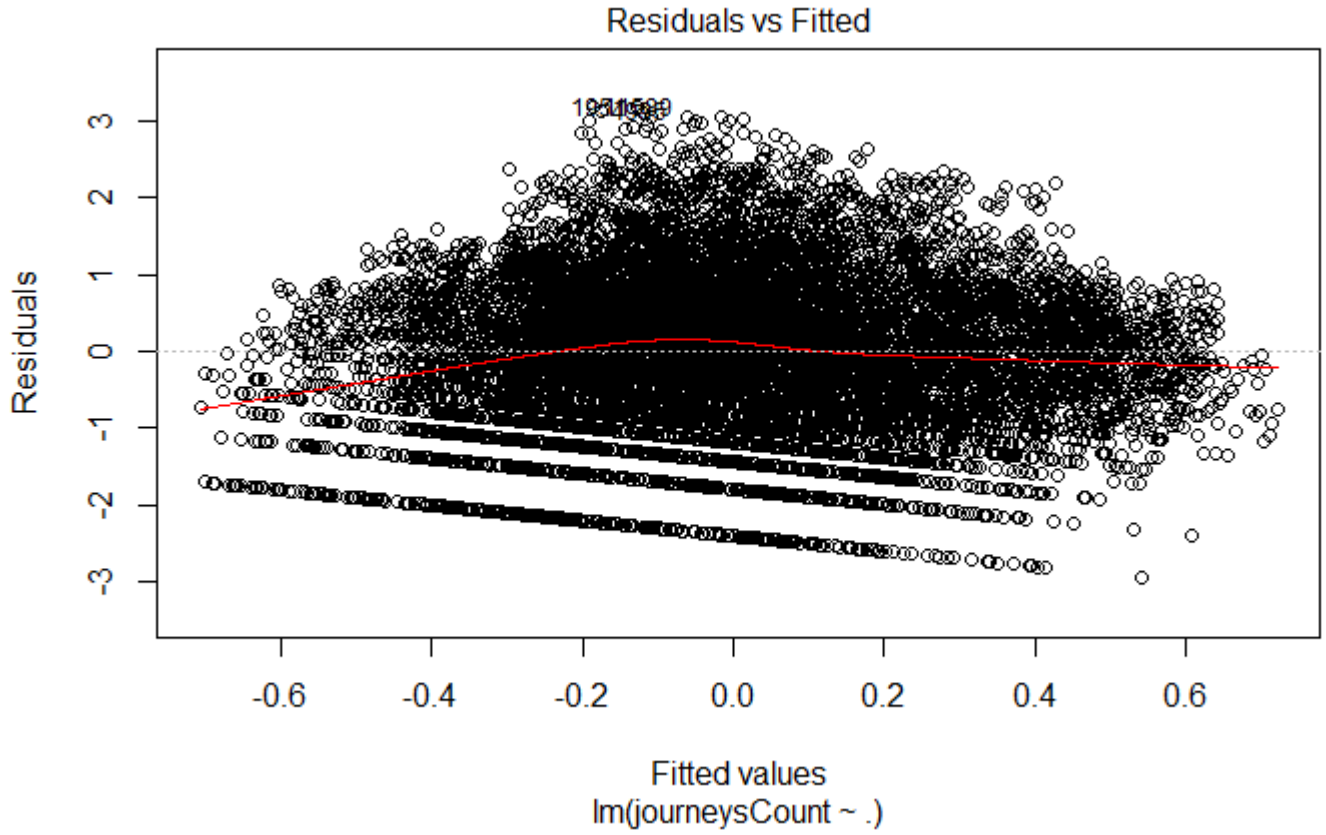
This third spread location plot is test Homoscedasticity which means wether there is any correlation in the varaince of independent variable. The plot should equally and randomly spread across horizontal line. In this case, there is so much data spread across the line but not equally spreaded.

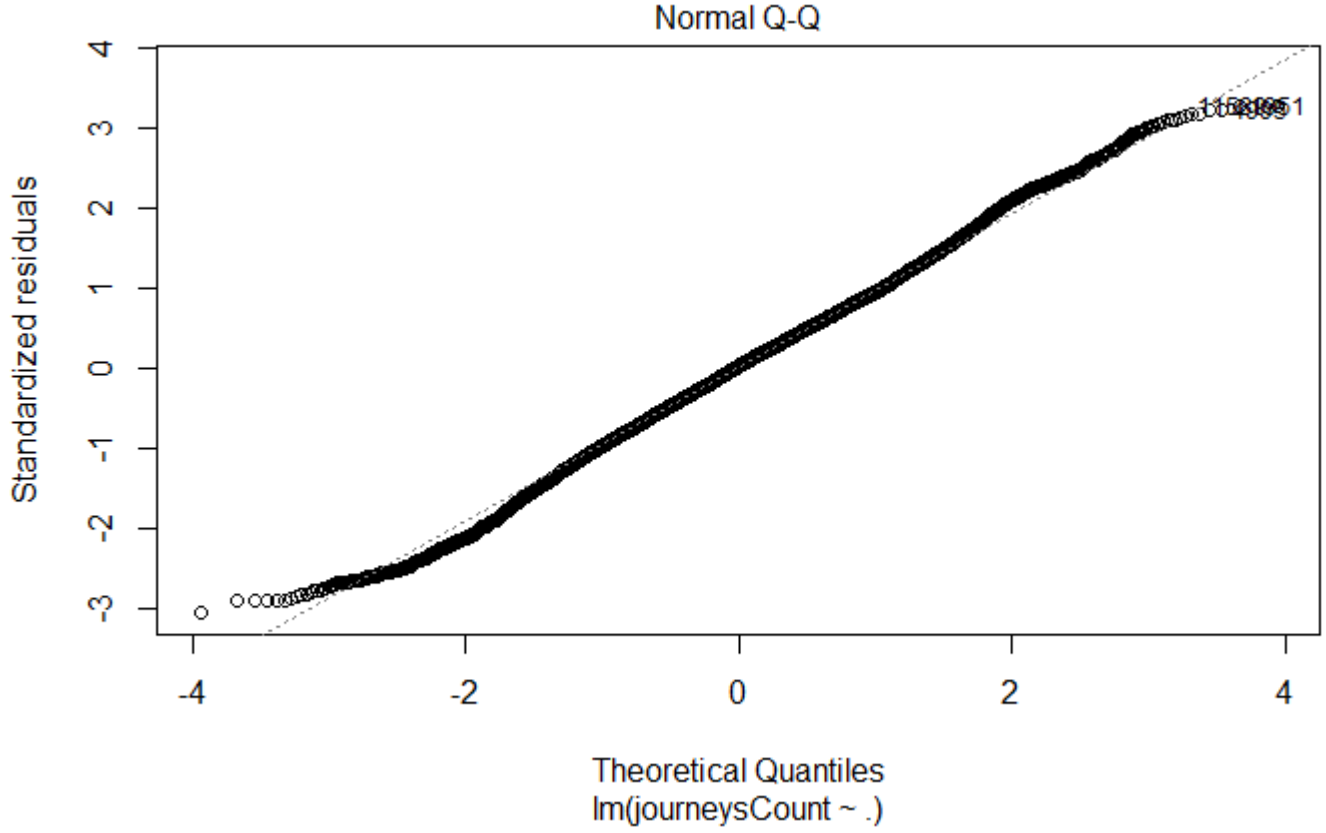
4. **Outlier Identification**

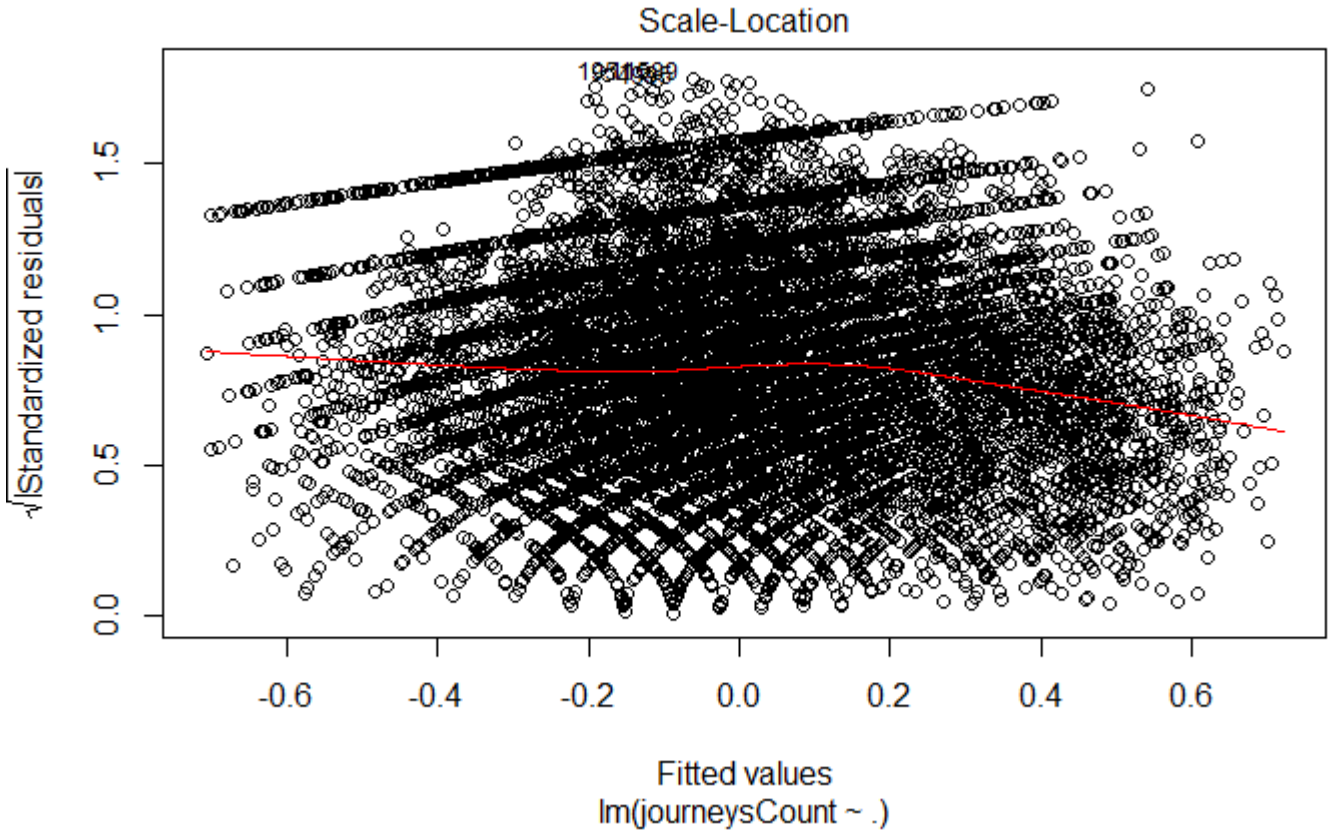
This fourth is Leverage plot which helps to identify the outliers exist and should be removed before building model. But, the plot does not show any outlier in the proposed model.

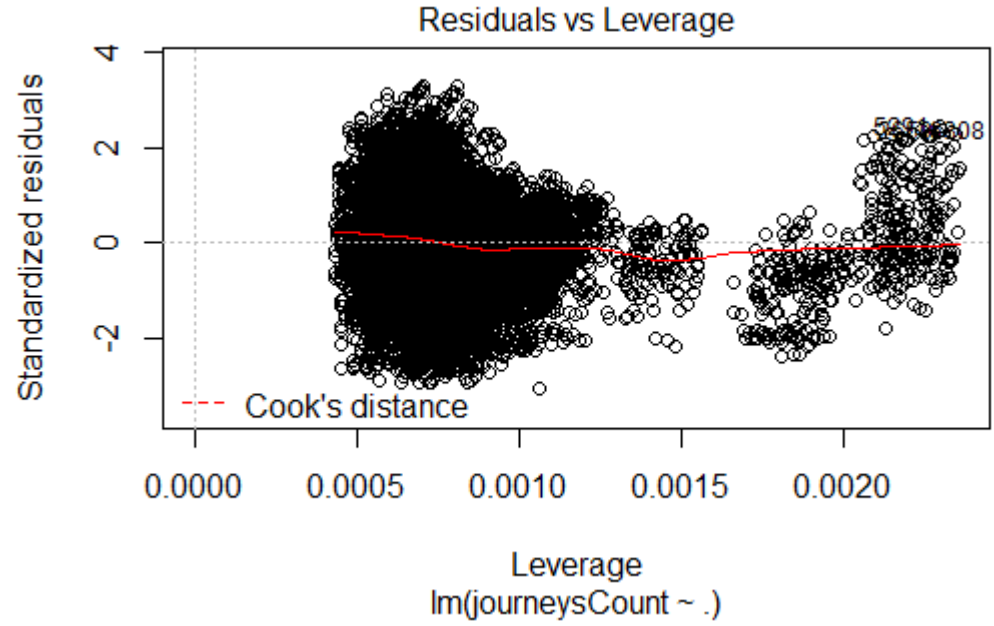
Hide

```
plot(lr)
```


Residuals vs Fitted
Residuals
Fitted values
lm(journeysCount ~ .)


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(journeysCount ~ .)





RQ2: Is there any Multicollinearity exist in predictors?

The below code uses Variance Inflation Factor (vif) to check multicolinearity between independet varaiables. If the value is higher than 4 or 5. Then it means variables are correlated. The one of correlated predictors can be removed. The code shows all the metrics values are less than 4. So there is no need to remove any predictors.

Hide

```
library(car)
vif(lr) %>%
    knitr::kable()
```

	x
date	1.008991
hour	1.000212
g_space	1.022710
ctf_h_ratio	1.130939
uk_b	1.141611
weekday.Fri	1.721593
weekday.Tue	1.724323
weekday.Wed	1.638417
weekday.Thu	1.641085

Optimizing Model

RQ3: Which predictors have strong relationship with the target variable?

Feature selection/significant features

To find the predictors having strong relationship with target variable,the feature selection can be performed using backward, forward and mixed feature selection.

Here, the following code selects features by **p-value** from the previous model and using **mixed** feature selection method. The features from both methods have been compared which gives same features.

Hide

```
##features selected from looking at p-value from previous model
lm_sig <- lm(formula=journeysCount~hour+g_space+ctf_h_ratio+uk_b+weekday.Tue+weekday.Wed+weekda
y.Thu,data = train)
##mixed feature selection using StepAIC function
lm_step <- MASS::stepAIC(lr, direction = "both", trace = FALSE)
print("feature selected on p-value")
```

```
[1] "feature selected on p-value"
```

Hide

```
lm_sig$call
```

```
lm(formula = journeysCount ~ hour + g_space + ctf_h_ratio + uk_b +
    weekday.Tue + weekday.Wed + weekday.Thu, data = train)
```

Hide

```
print("features selected using mixed method")
```

```
[1] "features selected using mixed method"
```

Hide

```
lm_step$call
```

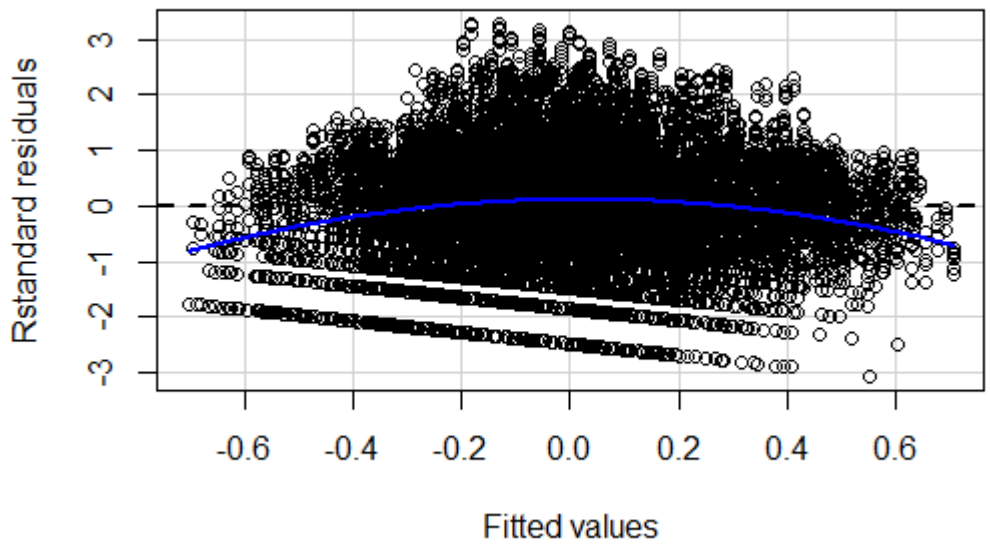
```
lm(formula = journeysCount ~ hour + g_space + ctf_h_ratio + uk_b +
    weekday.Tue + weekday.Wed + weekday.Thu, data = train)
```

Whether relationship is linear or not?

The residual plot below is not linear perfectly.So combinations of significant metrics will be added as interaction to final model.

Hide

```
library(car)
residualPlot(lm_sig, type = "rstandard")
```

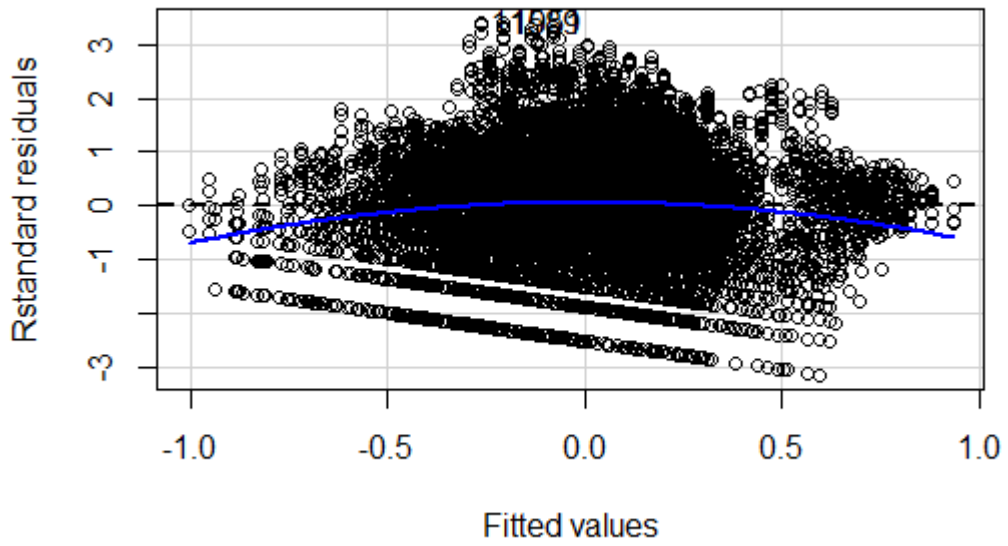


Interactions to fix non-linearity problem

Optimizing the model by adding interaction of **peak hour** with **g-space** and **uk_born**.Because the uk born people who are employed might use more bikes during peak hours.

Hide

```
lm_sig1 <- update(lm_sig, ~ .+hour*g_space+hour*uk_b+hour*ctf_h_ratio)
residualPlot(lm_sig1, type = "rstandard", id=TRUE)
```



Training/Testing of Final Model

After feature selection and adding interactions in the model, trainned and tested the model again. The results below shows R-square value has increased which is now 0.08 for train and 0.07 for test as the model is able to explain varaiation in the data.

Hide

```
train_preds = predict(lm_sig1, train)
test_preds = predict(lm_sig1, test) #
print( paste("R-square on train:", cor(train_preds, train$journeysCount)^2))
```

```
[1] "R-square on train: 0.0828008694495913"
```

Hide

```
print( paste("R-square on test:", cor(test_preds, test$journeysCount)^2))
```

```
[1] "R-square on test: 0.0772986513362683"
```

Summary

Hide

```
summary(lm_sig1)
```

```
Call:
lm(formula = journeysCount ~ hour + g_space + ctf_h_ratio + uk_b +
    weekday.Tue + weekday.Wed + weekday.Thu + hour:g_space +
    hour:uk_b + hour:ctf_h_ratio, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0012 -0.6025  0.0297  0.6119  3.2573

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.0006079  0.0086325  -0.070  0.94387
hour             -0.0704245  0.0086331  -8.157 3.75e-16 ***
g_space          -0.1306205  0.0087302 -14.962  < 2e-16 ***
ctf_h_ratio       0.1322201  0.0091816  14.401  < 2e-16 ***
uk_b             -0.1017805  0.0092237 -11.035  < 2e-16 ***
weekday.Tue       0.0541497  0.0094787   5.713 1.14e-08 ***
weekday.Wed      -0.0200824  0.0094007  -2.136  0.03268 *
weekday.Thu       0.0292640  0.0094115   3.109  0.00188 **
hour:g_space      0.0168858  0.0087229   1.936  0.05291 .
hour:uk_b        -0.1162368  0.0092044 -12.628  < 2e-16 ***
hour:ctf_h_ratio  0.0495965  0.0091502   5.420 6.06e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9581 on 12308 degrees of freedom
Multiple R-squared:  0.0828,    Adjusted R-squared:  0.08206
F-statistic: 111.1 on 10 and 12308 DF,  p-value: < 2.2e-16
```

Results Interpretation

low R-sqaure and low p-value(significant coefficients)

The R-square value indicates the varaiation exist in the target variable explained by the model. Lower p-value indicates independent variable are significant for target variable.This combination in the model explains that predictors correlate with the outcome variable however, they fail to explain the variation in the target varaiable. R-square is spread of data around the linear line. It means the given dataset have higher variation in the data which gives low R-square value. The higher variation in the dataset cause model to have larger prediction intervals making it difficult to predict correctly. The domains where predicting human behavior it is difficult to get high R-square value. R-square is mostly less than 50%.

Hypothesis

H1: Higher bikes demand on stations which are in densely populated area.(**False green space** shows negative value in the below bar plot.)

H2: The stations in poor areas have higher demand of bikes.(**True** because **ctf_h_ratio** had positive correlation with **no_owned_dwel_ratio** and **income score**.

H3: The number of bike demand is higher during peak hours on weekdays. (**Partially True** as the demand of bikes is high on **Tuesday** and **thursday** as compared to other days as can be seen in the plot. However, peak hours shows negative normalized value in the plot below)

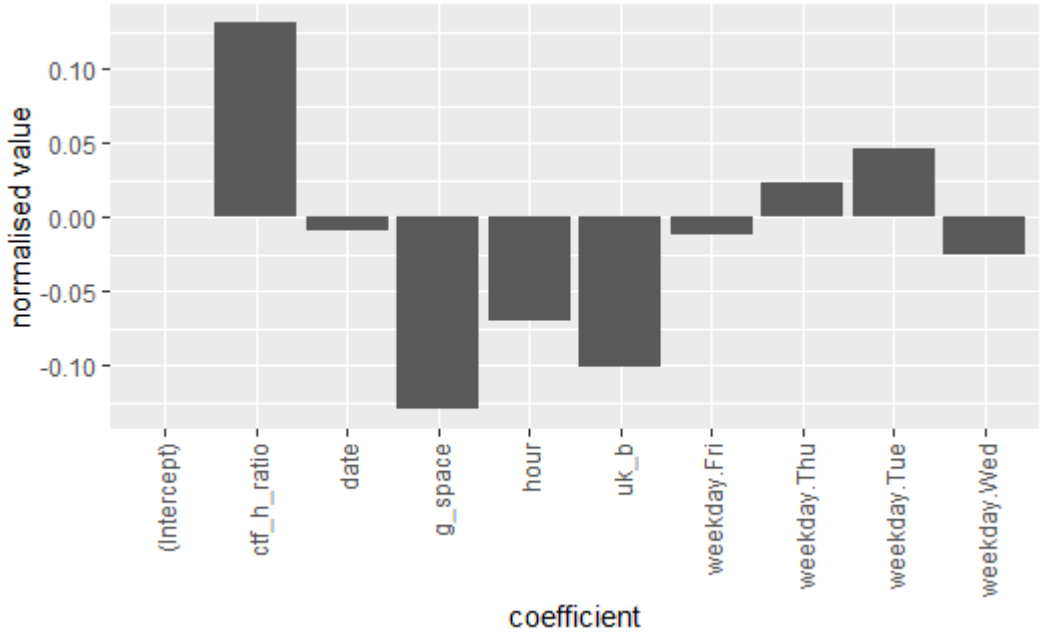
H4: The higher the ratio of employed people in an area has higher bike demand. (**True** as it had strong correlation with Journey count so removed before modelling)

H5: The higher the uk born people in an area, have higher demand for bikes.(**False** as the bar plot shows negative normalized value)

H6: The start of the month have higher demand(**False** because correlation matrix shows no correlation for **date** and negative value in the plot below)

Hide

```
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")
```



Limitations

1. The models shows mostly significant coefficients however, R-square value is low. R-square value depends on the the domain of data as here bike demand depends on human behavior which makes this models hard to predict bikes demand in practical.
2. Adding more combination of metrics resulted in higher accuracy which idicates more varaibles can be added to predict better.
3. There is need to change the parameters to see whether central London stations have more demand or not.
4. According to R-sqaure, the dataset contains higher variation making it harder to predict correctly.