AirbnbTest.R

AnujPC

2019-10-11

```
##Airbnb Datset EDA
##Author: Priyam Saxena

##Importing libraries
library(data.table)
library(ggplot2)  # tidyverse data visualization package
library(stringr)

library(tmap)    # for static and interactive maps
library(leaflet) # for interactive maps
library(mapview) # for interactive maps
library(shiny)   # for web applications

library(car)

## Loading required package: carData

#Importing csv file from my local computer
airbnbOriginalDF =read.csv("D:/Priyam/FirstSemester/MVA project/airbnb-host-analysis-for-newyork/Airbnb Host Data For Newyork City.csv")

##Converting data frame to data table
setDT(airbnbOriginalDF)

##Number of rows and columns in dataset
dim(airbnbOriginalDF)

## [1] 48895    16

##Gaining insight on data type of each column
str(airbnbOriginalDF)

## Classes 'data.table' and 'data.frame':   48895 obs. of  16 variables:
##  $ id                            : int  2539 2595 3647 3831 5022 5099 5121 5178 5203 5238 ...
##  $ name                          : Factor w/ 47897 levels "","'Fan'tastic",..: 12652 38163 45162 15693 19357 24992 8328 25039 15588 17673 ...
##  $ host_id                       : int  2787 2845 4632 4869 7192 7322 7356 8967 7490 7549 ...
##  $ host_name                     : Factor w/ 11453 levels "","'Cil","#NAME?",..: 5051 4846 2962 6264 5982 1970 3601 9699 6935 1264 ...
```

```
##  $ neighbourhood_group           : Factor w/ 5 levels
"Bronx","Brooklyn",..: 2 3 3 2 3 3 2 3 3 3 ...
##  $ neighbourhood                 : Factor w/ 221 levels "Allerton","Arden
Heights",..: 109 128 95 42 62 138 14 96 203 36 ...
##  $ latitude                      : num  40.6 40.8 40.8 40.7 40.8 ...
##  $ longitude                     : num  -74 -74 -73.9 -74 -73.9 ...
##  $ room_type                     : Factor w/ 3 levels "Entire
home/apt",..: 2 1 2 1 1 1 2 2 2 1 ...
##  $ price                         : int  149 225 150 89 80 200 60 79 79 150
...
##  $ minimum_nights                : int  1 1 3 1 10 3 45 2 2 1 ...
##  $ number_of_reviews             : int  9 45 0 270 9 74 49 430 118 160 ...
##  $ last_review                   : Factor w/ 1765 levels "","1/1/2013",..:
203 1059 1 1438 348 1234 277 1244 1383 1317 ...
##  $ reviews_per_month             : num  0.21 0.38 NA 4.64 0.1 0.59 0.4
3.47 0.99 1.33 ...
##  $ calculated_host_listings_count: int  6 2 1 1 1 1 1 1 1 4 ...
##  $ availability_365              : int  365 355 365 194 0 129 0 220 0 188
...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
##Gaining insight on complete data
summary(airbnbOriginalDF)
```

```
##        id                                                name
##  Min.   :     2539   Hillside Hotel                      :    18
##  1st Qu.: 9471945   Home away from home                 :    17
##  Median :19677284                                       :    16
##  Mean   :19017143   New york Multi-unit building        :    16
##  3rd Qu.:29152178   Brooklyn Apartment                  :    12
##  Max.   :36487245   Loft Suite @ The Box House Hotel:    11
##                      (Other)                             :48805
##     host_id                  host_name         neighbourhood_group
##  Min.   :       2438   Michael       :   417   Bronx          : 1091
##  1st Qu.:   7822033   David         :   403   Brooklyn       :20104
##  Median : 30793816   Sonder (NYC):   327   Manhattan      :21661
##  Mean   : 67620011   John          :   294   Queens         : 5666
##  3rd Qu.:107434423   Alex          :   279   Staten Island:   373
##  Max.   :274321313   Blueground    :   232
##                        (Other)       :46943
##              neighbourhood       latitude        longitude
##  Williamsburg       : 3920   Min.   :40.50   Min.   :-74.24
##  Bedford-Stuyvesant: 3714   1st Qu.:40.69   1st Qu.:-73.98
##  Harlem             : 2658   Median :40.72   Median :-73.96
##  Bushwick           : 2465   Mean   :40.73   Mean   :-73.95
##  Upper West Side    : 1971   3rd Qu.:40.76   3rd Qu.:-73.94
##  Hell's Kitchen     : 1958   Max.   :40.91   Max.   :-73.71
##  (Other)            :32209
##             room_type          price          minimum_nights
##  Entire home/apt:25409   Min.   :     0.0   Min.   :   1.00
```

```
##  Private room   :22326   1st Qu.:   69.0   1st Qu.:   1.00  
##  Shared room    : 1160   Median :  106.0   Median :   3.00  
##                          Mean   :  152.7   Mean   :   7.03  
##                          3rd Qu.:  175.0   3rd Qu.:   5.00  
##                          Max.   :10000.0   Max.   :1250.00  
##                                                              
##  number_of_reviews    last_review    reviews_per_month
##  Min.   :  0.00             :10052   Min.   : 0.010   
##  1st Qu.:  1.00    6/23/2019: 1413   1st Qu.: 0.190   
##  Median :  5.00    7/1/2019 : 1359   Median : 0.720   
##  Mean   : 23.27    6/30/2019: 1341   Mean   : 1.373   
##  3rd Qu.: 24.00    6/24/2019:  875   3rd Qu.: 2.020   
##  Max.   :629.00    7/7/2019 :  718   Max.   :58.500   
##                    (Other)  :33137   NA's   :10052    
##  calculated_host_listings_count availability_365
##  Min.   :  1.000                Min.   :  0.0   
##  1st Qu.:  1.000                1st Qu.:  0.0   
##  Median :  1.000                Median : 45.0   
##  Mean   :  7.144                Mean   :112.8   
##  3rd Qu.:  2.000                3rd Qu.:227.0   
##  Max.   :327.000                Max.   :365.0   
## 
```

```
##View first 5 rows to get insight of data
head(airbnbOriginalDF,5)
```

```
##      id                                             name host_id
## 1: 2539               Clean & quiet apt home by the park    2787
## 2: 2595                            Skylit Midtown Castle    2845
## 3: 3647              THE VILLAGE OF HARLEM....NEW YORK !    4632
## 4: 3831                  Cozy Entire Floor of Brownstone    4869
## 5: 5022 Entire Apt: Spacious Studio/Loft by central park    7192
##      host_name neighbourhood_group neighbourhood latitude longitude
## 1:        John            Brooklyn    Kensington 40.64749 -73.97237
## 2:    Jennifer           Manhattan       Midtown 40.75362 -73.98377
## 3:   Elisabeth           Manhattan        Harlem 40.80902 -73.94190
## 4: LisaRoxanne            Brooklyn  Clinton Hill 40.68514 -73.95976
## 5:       Laura           Manhattan   East Harlem 40.79851 -73.94399
##          room_type price minimum_nights number_of_reviews last_review
## 1:    Private room   149              1                 9  10/19/2018
## 2: Entire home/apt   225              1                45   5/21/2019
## 3:    Private room   150              3                 0            
## 4: Entire home/apt    89              1               270    7/5/2019
## 5: Entire home/apt    80             10                 9  11/19/2018
##    reviews_per_month calculated_host_listings_count availability_365
## 1:              0.21                              6              365
## 2:              0.38                              2              355
## 3:                NA                              1              365
## 4:              4.64                              1              194
## 5:              0.10                              1                0
```

```
   ######## DATA CLEANING #########

##Checking null/missing value in dataset
table(is.na(airbnbOriginalDF))

## 
##  FALSE   TRUE 
## 772268  10052

##Checking null values in review per month column
table(is.na(airbnbOriginalDF$reviews_per_month))

## 
## FALSE  TRUE 
## 38843 10052

#Removing values which are null and storing in new table.
airbnbNoNADT = airbnbOriginalDF[airbnbOriginalDF$reviews_per_month != 'NA']

## Rechecking, and can see no null values present now.
table(is.na(airbnbNoNADT))

## 
##  FALSE 
## 621488

table(is.na(airbnbNoNADT$reviews_per_month)) #airbnbNoNADT is datatable with 
not any null values

## 
## FALSE 
## 38843

#Converting datatype of last review date to DAte Format.
airbnbNoNADT[,last_review:=as.Date(last_review, '%m/%d/%Y')]
str(airbnbNoNADT)

## Classes 'data.table' and 'data.frame':   38843 obs. of  16 variables:
##  $ id                            : int  2539 2595 3831 5022 5099 5121 5178 
5203 5238 5295 ...
##  $ name                          : Factor w/ 47897 levels 
"","'Fan'tastic",..: 12652 38163 15693 19357 24992 8328 25039 15588 17673 
5645 ...
##  $ host_id                       : int  2787 2845 4869 7192 7322 7356 8967 
7490 7549 7702 ...
##  $ host_name                     : Factor w/ 11453 levels 
"","'Cil","#NAME?",..: 5051 4846 6264 5982 1970 3601 9699 6935 1264 6084 ...
##  $ neighbourhood_group           : Factor w/ 5 levels 
"Bronx","Brooklyn",..: 2 3 2 3 3 2 3 3 3 3 ...
##  $ neighbourhood                 : Factor w/ 221 levels "Allerton","Arden 
Heights",..: 109 128 42 62 138 14 96 203 36 203 ...
##  $ latitude                      : num  40.6 40.8 40.7 40.8 40.7 ...
```

```
##  $ longitude                     : num  -74 -74 -74 -73.9 -74 ...
##  $ room_type                     : Factor w/ 3 levels "Entire
home/apt",..: 2 1 1 1 1 2 2 2 1 1 ...
##  $ price                         : int  149 225 89 80 200 60 79 79 150 135
...
##  $ minimum_nights                : int  1 1 1 10 3 45 2 2 1 5 ...
##  $ number_of_reviews             : int  9 45 270 9 74 49 430 118 160 53
...
##  $ last_review                   : Date, format: "2018-10-19" "2019-05-21"
...
##  $ reviews_per_month             : num  0.21 0.38 4.64 0.1 0.59 0.4 3.47
0.99 1.33 0.43 ...
##  $ calculated_host_listings_count: int  6 2 1 1 1 1 1 1 4 1 ...
##  $ availability_365              : int  365 355 194 0 129 0 220 0 188 6
...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
#Lets try to further analyze our data by analysing data types.

##CONVERTING CATEGORICAL VALUES TO FACTORS

unique(airbnbNoNADT$neighbourhood_group)
```

```
## [1] Brooklyn      Manhattan     Queens        Staten Island Bronx
## Levels: Bronx Brooklyn Manhattan Queens Staten Island
```

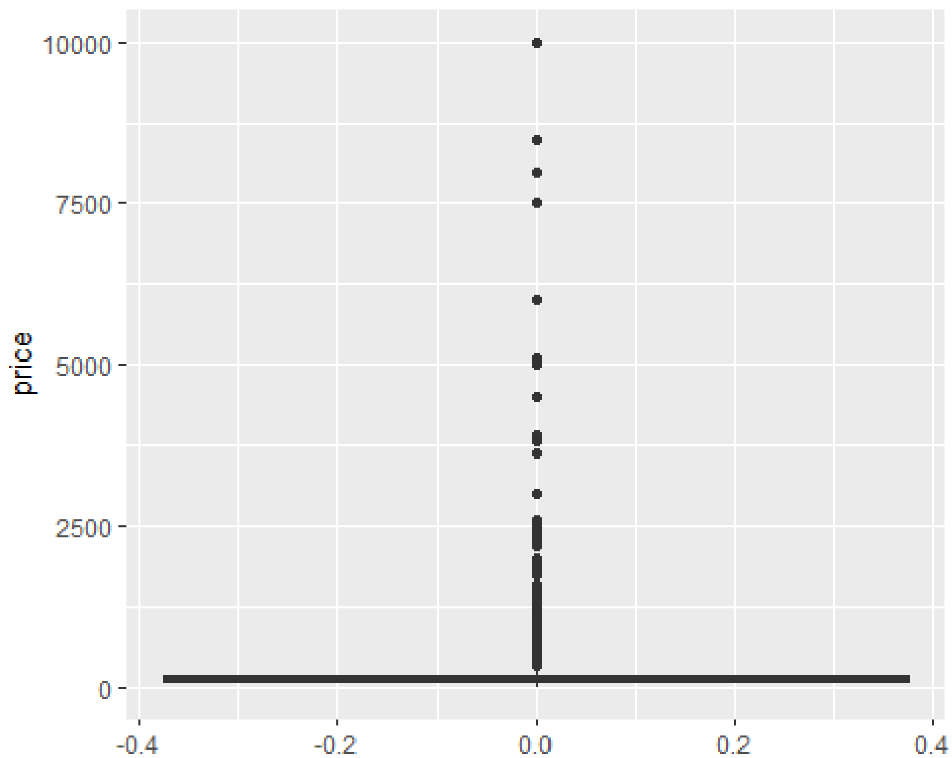
```
#As the neighbourhood_group column has 5 categorical values, we can factor
it, and convert our string data type.
airbnbNoNADT[,neighbourhood_group:= factor(neighbourhood_group)]


unique(airbnbNoNADT$neighbourhood)
```

```
##   [1] Kensington                 Midtown
##   [3] Clinton Hill               East Harlem
##   [5] Murray Hill                Bedford-Stuyvesant
##   [7] Hell's Kitchen             Upper West Side
##   [9] Chinatown                  South Slope
##  [11] West Village               Williamsburg
##  [13] Fort Greene                Chelsea
##  [15] Crown Heights              Park Slope
##  [17] Windsor Terrace            Inwood
##  [19] East Village               Harlem
##  [21] Greenpoint                 Bushwick
##  [23] Lower East Side            Prospect-Lefferts Gardens
##  [25] Long Island City           Kips Bay
##  [27] SoHo                       Upper East Side
##  [29] Prospect Heights           Washington Heights
##  [31] Woodside                   Flatbush
##  [33] Brooklyn Heights           Carroll Gardens
```

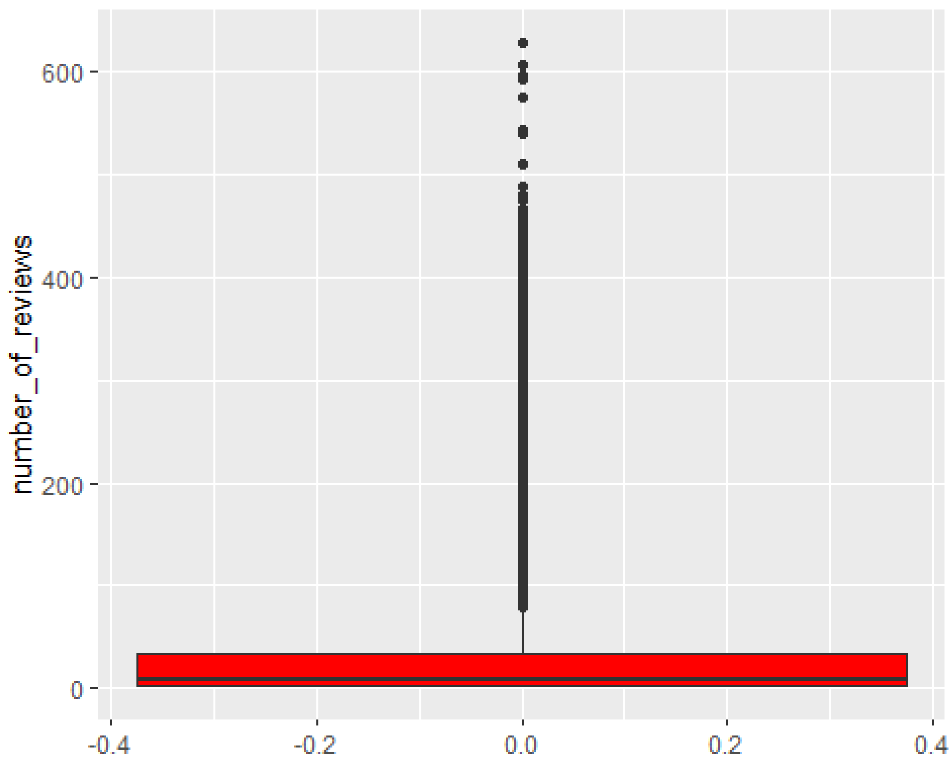
```
##  [35] Gowanus                    Flatlands
##  [37] Cobble Hill                Flushing
##  [39] Boerum Hill                Sunnyside
##  [41] DUMBO                      St. George
##  [43] Highbridge                 Financial District
##  [45] Ridgewood                  Morningside Heights
##  [47] Jamaica                    Middle Village
##  [49] NoHo                       Ditmars Steinway
##  [51] Flatiron District          Roosevelt Island
##  [53] Greenwich Village          Little Italy
##  [55] East Flatbush              Tompkinsville
##  [57] Astoria                    Eastchester
##  [59] Kingsbridge                Two Bridges
##  [61] Queens Village             Rockaway Beach
##  [63] Forest Hills               Nolita
##  [65] Woodlawn                   University Heights
##  [67] Gramercy                   Allerton
##  [69] East New York              Theater District
##  [71] Concourse Village          Sheepshead Bay
##  [73] Emerson Hill               Fort Hamilton
##  [75] Bensonhurst                Tribeca
##  [77] Shore Acres                Sunset Park
##  [79] Concourse                  Elmhurst
##  [81] Brighton Beach             Jackson Heights
##  [83] Cypress Hills              St. Albans
##  [85] Arrochar                   Rego Park
##  [87] Wakefield                  Clifton
##  [89] Bay Ridge                  Graniteville
##  [91] Spuyten Duyvil             Stapleton
##  [93] Briarwood                  Ozone Park
##  [95] Columbia St                Vinegar Hill
##  [97] Mott Haven                 Longwood
##  [99] Canarsie                   Battery Park City
## [101] Civic Center               East Elmhurst
## [103] New Springville            Morris Heights
## [105] Arverne                    Gravesend
## [107] Tottenville                Mariners Harbor
## [109] Concord                    Borough Park
## [111] Bayside                    Downtown Brooklyn
## [113] Port Morris                Fieldston
## [115] Kew Gardens                Midwood
## [117] College Point              Mount Eden
## [119] City Island                Glendale
## [121] Red Hook                   Richmond Hill
## [123] Maspeth                    Port Richmond
## [125] Williamsbridge             Soundview
## [127] Woodhaven                  Co-op City
## [129] Stuyvesant Town            Parkchester
## [131] North Riverdale            Dyker Heights
## [133] Bronxdale                  Sea Gate
```

```
## [135] Riverdale                   Kew Gardens Hills
## [137] Bay Terrace                 Norwood
## [139] Claremont Village           Whitestone
## [141] Fordham                     Bayswater
## [143] Navy Yard                   Brownsville
## [145] Eltingville                 Mount Hope
## [147] Clason Point                Lighthouse Hill
## [149] Springfield Gardens         Howard Beach
## [151] Belle Harbor                Jamaica Estates
## [153] Van Nest                    Bellerose
## [155] Fresh Meadows               Morris Park
## [157] West Brighton               Far Rockaway
## [159] South Ozone Park            Tremont
## [161] Corona                      Great Kills
## [163] Manhattan Beach             Marble Hill
## [165] Dongan Hills                East Morrisania
## [167] Hunts Point                 Neponsit
## [169] Pelham Bay                  Randall Manor
## [171] Throgs Neck                 Todt Hill
## [173] West Farms                  Silver Lake
## [175] Laurelton                   Grymes Hill
## [177] Holliswood                  Pelham Gardens
## [179] Rosedale                    Castleton Corners
## [181] Edgemere                    New Brighton
## [183] Baychester                  Melrose
## [185] Bergen Beach                Cambria Heights
## [187] Richmondtown                Howland Hook
## [189] Schuylerville               Coney Island
## [191] Prince's Bay                South Beach
## [193] Bath Beach                  Midland Beach
## [195] Jamaica Hills               Oakwood
## [197] Castle Hill                 Douglaston
## [199] Huguenot                    Edenwald
## [201] Belmont                     Grant City
## [203] Westerleigh                 Morrisania
## [205] Bay Terrace, Staten Island Westchester Square
## [207] Little Neck                 Rosebank
## [209] Unionport                   Mill Basin
## [211] Hollis                      Arden Heights
## [213] Bull's Head                 Olinville
## [215] Rossville                   Breezy Point
## [217] Willowbrook                 New Dorp Beach
## 221 Levels: Allerton Arden Heights Arrochar Arverne Astoria ... Woodside

#For neighbourhood, we get 217 unique values. Here to reduce storage we can
covert all similar type to lower case and also trim white spaces, so that
each anme is unique.

#Converting all same type name to lower cases
airbnbNoNADT[,neighbourhood:=tolower(neighbourhood)]
```

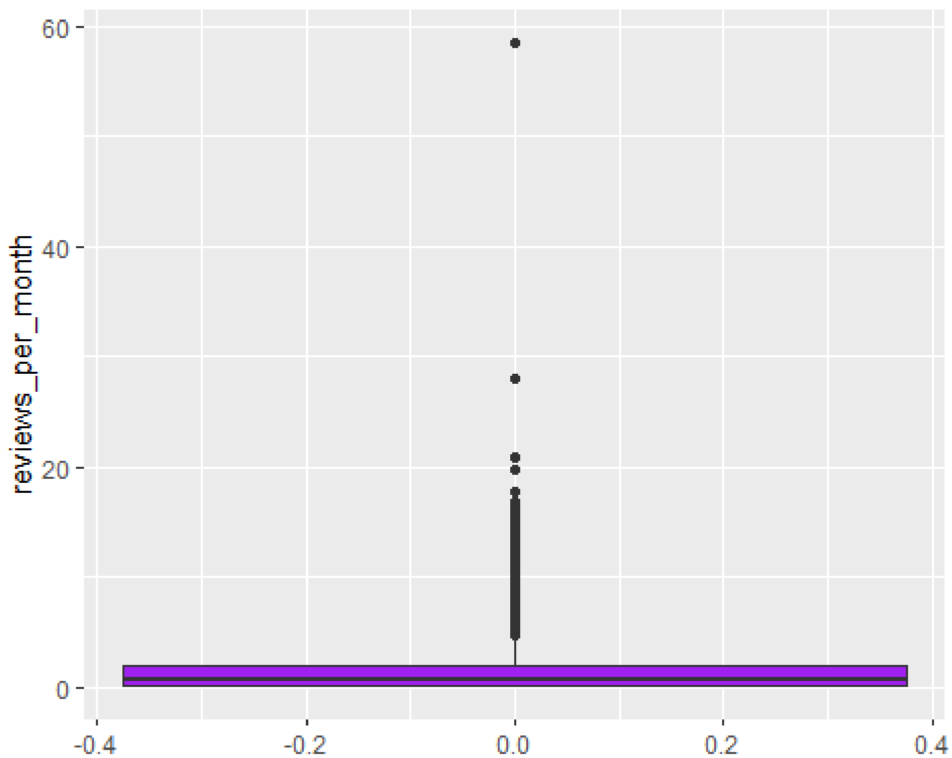
```r
#Removing all white spaces
airbnbNoNADT[,neighbourhood:=trimws(neighbourhood)]

#For room type, we get 3 unique categorical values. we can factor it, and
convert our string datatype.
unique(airbnbNoNADT$room_type)

## [1] Private room    Entire home/apt Shared room
## Levels: Entire home/apt Private room Shared room

airbnbNoNADT[,room_type:= factor(room_type)]

  ######  Exploratory Data Analysis #######

##We will further analyze our data to see if any outliers are there and also
find relations among useful variables.
#Analysing longitude data. The distribution is fair
summary(airbnbNoNADT$longitude)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -74.24  -73.98  -73.95  -73.95  -73.94  -73.71

#Analysing avialbility data. THe data is fair and no extreme values.
summary(airbnbNoNADT$availability_365)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0    55.0   114.9   229.0   365.0

#Analysing price data. Could see extremely large values. Lets draw a plot to
see the distribution.
summary(airbnbNoNADT$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    69.0   101.0   142.3   170.0 10000.0

ggplot(airbnbNoNADT,aes(y=price))+geom_boxplot(fill='yellow')
```




```
#In plot we can see some outliers. lets run below and see how many are such properties that have price greater than 2500.
nrow(airbnbNoNADT[price>2500])

## [1] 25

#By runing this, we find only 25 such properties. This can be dropped as we 38k plus data

#Analysing number of reviews data. Could see extremely large values. Lets draw a plot to see the distrinution.
summary(airbnbNoNADT$number_of_reviews)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     3.0     9.0    29.3    33.0   629.0

ggplot(airbnbNoNADT,aes(y=number_of_reviews))+geom_boxplot(fill ='red')
```

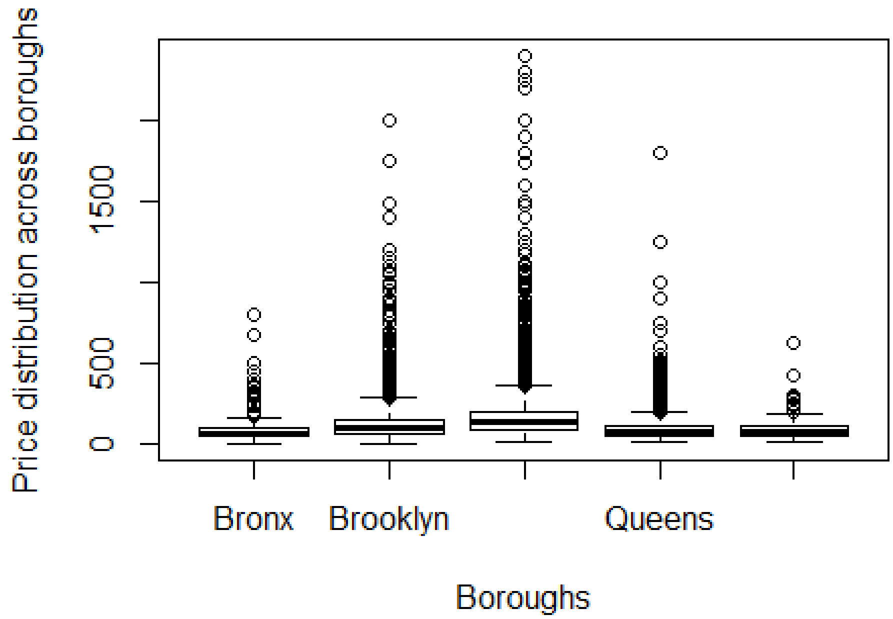



```
#In plot we can see some outliers. lets run below and see how many are such properties that have no of reviews greater than 400.
#Such a huge review for one or two property seems to be some spam or fake. We shall how many such rows are there in our data.
nrow(airbnbNoNADT[number_of_reviews>400])

## [1] 39

#We found 39 rows which have number of reviews greater than 400.
airbnbNoNADT[number_of_reviews>400,unique(neighbourhood_group)]

## [1] Manhattan Brooklyn  Queens
## Levels: Bronx Brooklyn Manhattan Queens Staten Island

#When we checked for which areas this spam review is , it shows Manhattan, Brooklyn and Queens. So there is no clear indication by this data, we will drop this to further clean our data and remove outliers.

#Analysingreviews per month Could see extremely large values. Lets draw a plot to see the distrinution.
summary(airbnbNoNADT$reviews_per_month)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.010   0.190   0.720   1.373   2.020  58.500

ggplot(airbnbNoNADT,aes(y=reviews_per_month))+geom_boxplot(fill='purple')
```

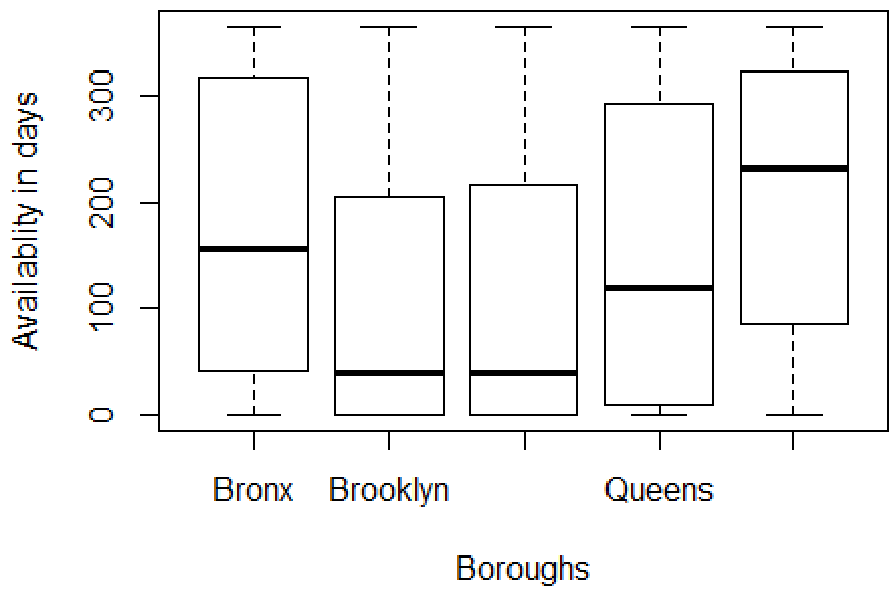



```
#In plot we can see some outliers. lets run below and see how many are such
properties that have reviews per month greater than 10.
#Most of the data is located below 5. We shall how many such rows rae there
in our data which have review per month greater than 10
nrow(airbnbNoNADT[reviews_per_month>10])

## [1] 81

airbnbNoNADT[reviews_per_month>10,unique(neighbourhood_group)]

## [1] Queens        Bronx         Brooklyn      Manhattan     Staten Island
## Levels: Bronx Brooklyn Manhattan Queens Staten Island

#When we tried checking if any particular locality has more reviews, it does
not give any indication. The result is spread out for all localities. We can
drop this rows, as it wont yield anything peculiar.

#With above summary and plot we found few ouliers, therefore that data  we
have dropped below, conforming it is not impact our main dataset.
airbnbCleaned = airbnbNoNADT[price<2500 & number_of_reviews<400 &
reviews_per_month<10]
#airbnbCleaned is our Final cleaned data


#Attach is used to access column directly without using data table name.
attach(airbnbCleaned)
```

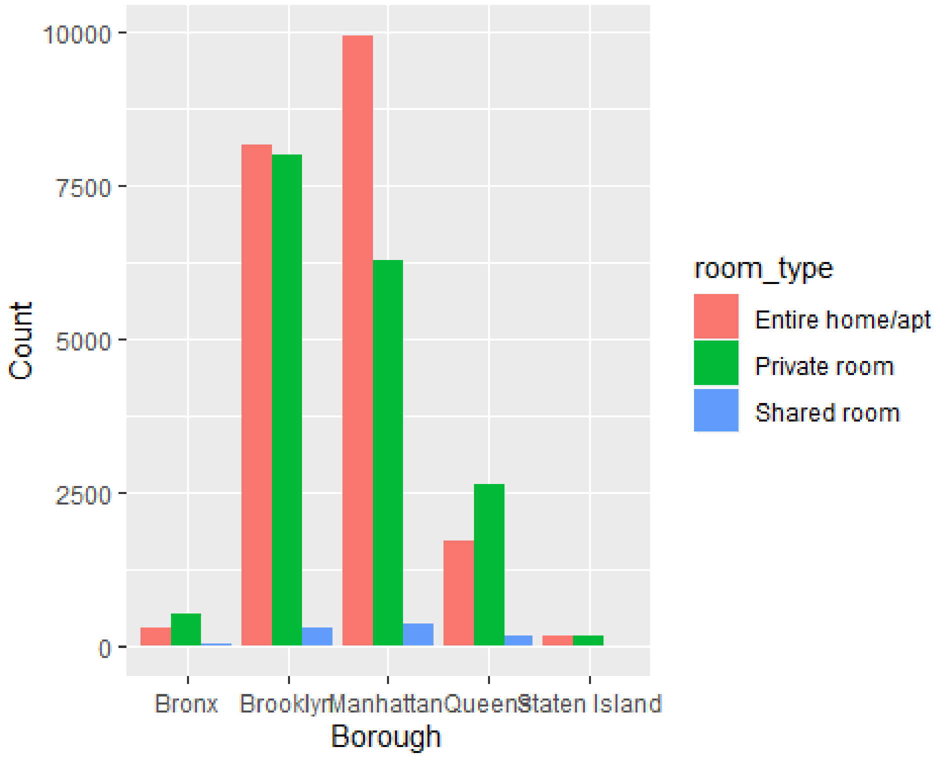
```r
#Analysing the price distribution based on location
plot(neighbourhood_group,price, xlab= 'Boroughs', ylab='Price distribution across boroughs')
```



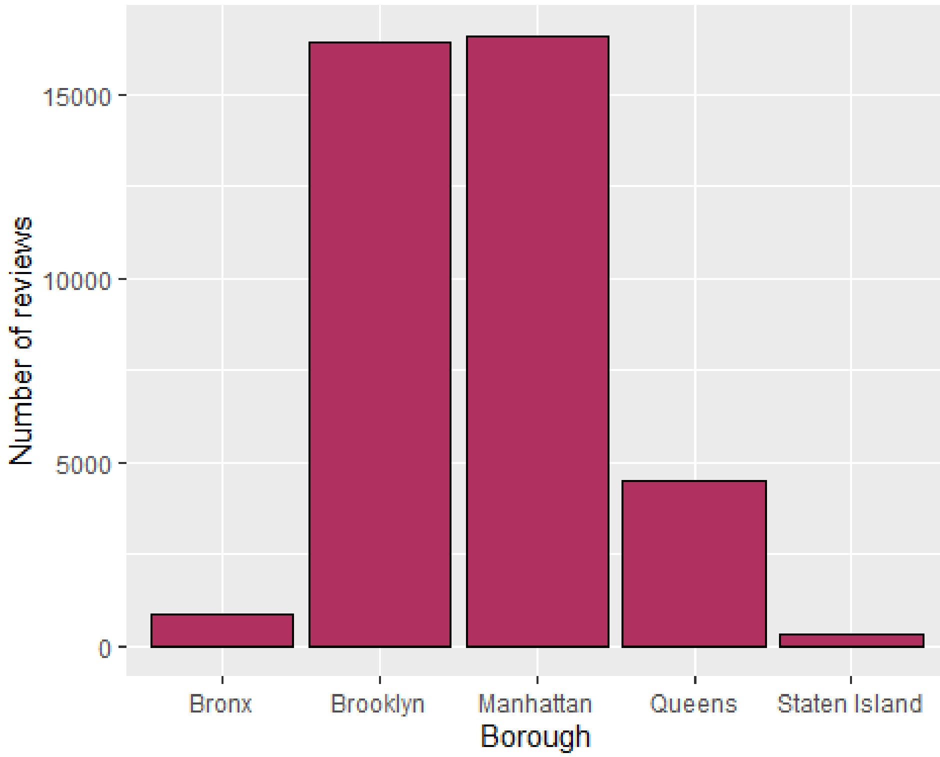

```r
#Analysing the availability across borouhgs
plot(airbnbCleaned$neighbourhood_group, airbnbCleaned$availability_365, xlab ='Boroughs', ylab= 'Availablity in days')
```



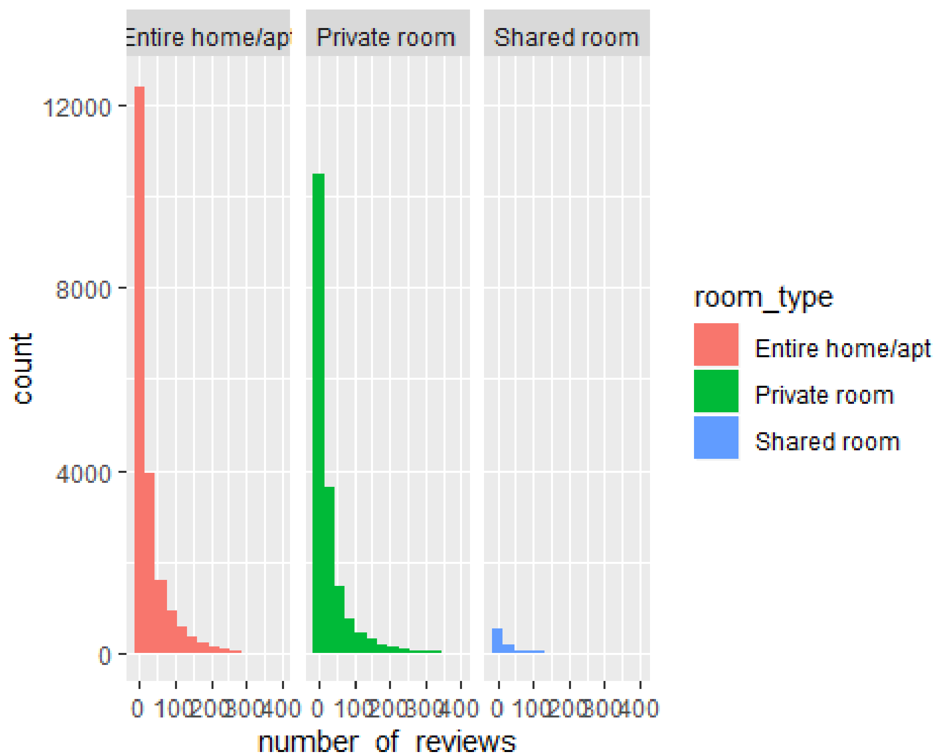

```
#Analysing the room types which are preferred and mostly listed across all
boroughs
ggplot(airbnbCleaned, aes(x=neighbourhood_group, fill =
room_type))+geom_bar(position = "dodge") + xlab("Borough") + ylab("Count")
```




```r
#Analysis:
#We can see that Entire home apartment listings are highest in number except
Queens and Bronx. Queens has more ‘Private’ style property than ‘Apartments’.
#The maximum apartment style listings are located in Manhattan, constituting
90% of all properties in that neighborhood. Next is Brooklyn with 75%
Apartment style listing.

#Analysing which borough property is mostly at top by ratings.
ggplot(airbnbCleaned, aes(x=neighbourhood_group, fill =
number_of_reviews))+geom_bar(color='black', fill='maroon') + xlab("Borough")
+ ylab("Number of reviews")
```

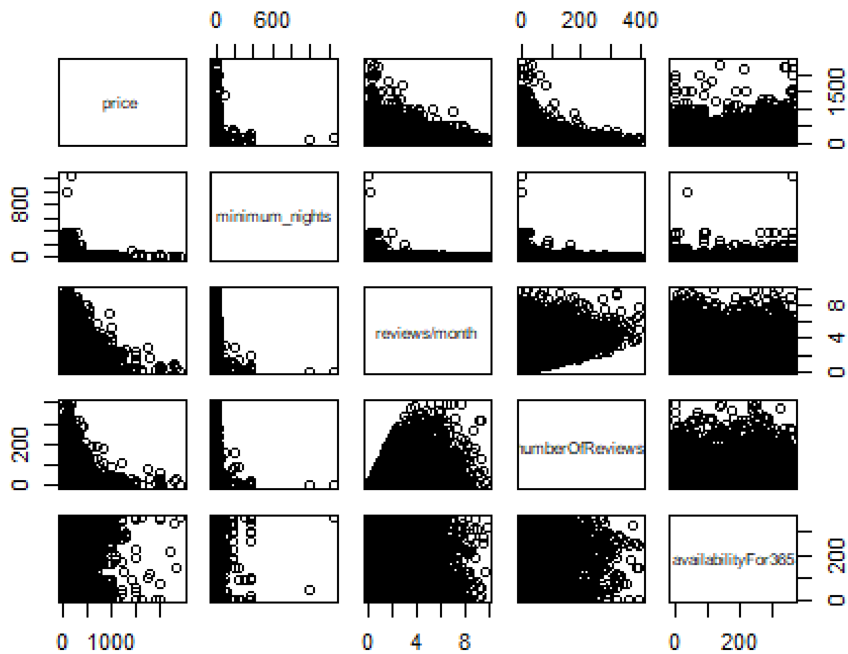



```
#Analysis:
#We can see that properties in Manhattan has recieved most of customer review
, followed by Brooklyn.

#Analyzing which kind of property is mostly preferred by people
ggplot(airbnbCleaned, aes(x= number_of_reviews, fill= room_type )) +
geom_histogram(binwidth = 30)+facet_wrap(room_type)
```

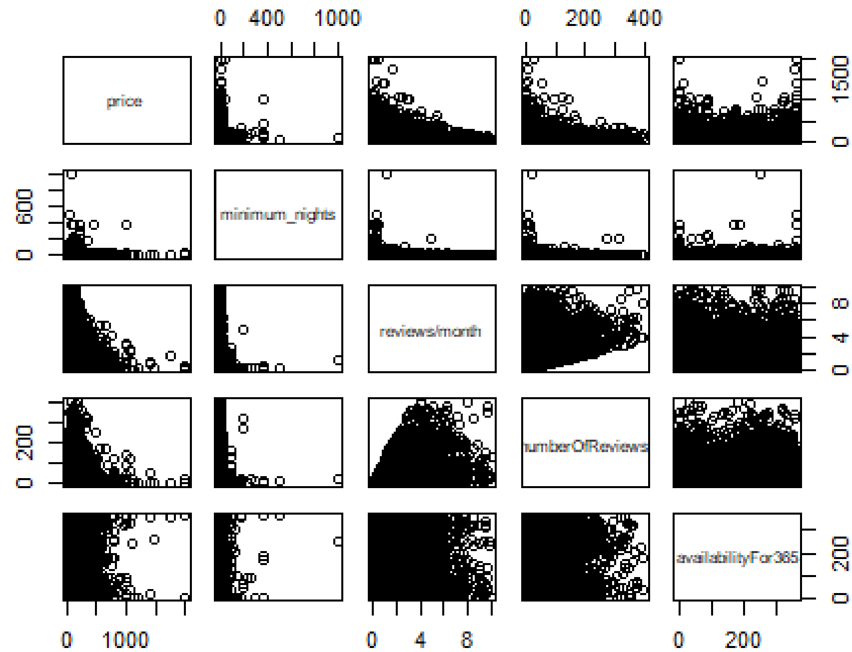



```r
#With above data, we can see that Apartment type properties are mostly
preferred, since they are the ones
#receiving maximum ratings. After which people prefer private rooms. Shared
rooms have received very few
#rating. This would be helpful for other business to avoid providing shared
rooms

##### FINDING CORRELATIONS #####

detach(airbnbCleaned)  ## Will unmask the columns

#Below we have stored the data for each boroughs in different table which
will help to analyze each borough individually as well if required
##Manhattan area dataset
airbnbManhattan = airbnbCleaned[neighbourhood_group=='Manhattan']
nrow(airbnbManhattan)

## [1] 16584

##Queens area dataset
airbnbQueens = airbnbCleaned[neighbourhood_group=='Queens']
nrow(airbnbQueens)

## [1] 4504
```

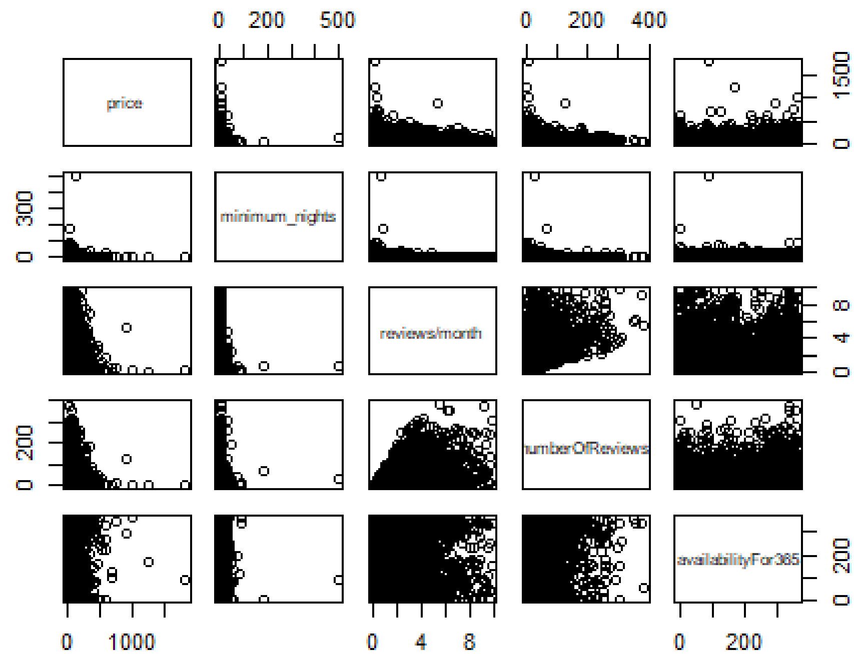
```r
##Brooklyn area dataset
airbnbBrooklyn = airbnbCleaned[neighbourhood_group=='Brooklyn']
nrow(airbnbBrooklyn)

## [1] 16421

##Bronx area dataset
airbnbBronx = airbnbCleaned[neighbourhood_group=='Bronx']
nrow(airbnbBronx)

## [1] 875

##Staten Island area dataset
airbnbStatenIsland = airbnbCleaned[neighbourhood_group=='Staten Island']
nrow(airbnbStatenIsland)

## [1] 313

#Creating corelation matrix for each  boroughs
diagnolcol = c("price","minimum_nights","reviews/month", "numberOfReviews",
"availabilityFor365")


##MANHATTAN
pairs(data.table(
  airbnbManhattan$price,
  airbnbManhattan$minimum_nights,
  airbnbManhattan$reviews_per_month,
  airbnbManhattan$number_of_reviews,
  airbnbManhattan$availability_365), labels = diagnolcol)
```



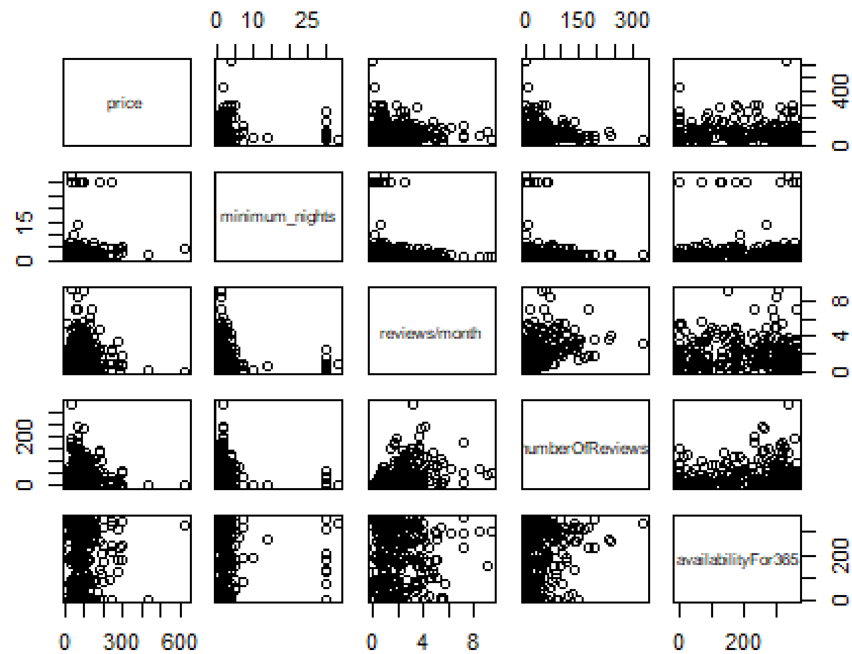

```r
##BROOKLYN
pairs(data.table(
  airbnbBrooklyn$price,
  airbnbBrooklyn$minimum_nights,
  airbnbBrooklyn$reviews_per_month,
  airbnbBrooklyn$number_of_reviews,
  airbnbBrooklyn$availability_365), labels = diagnolcol)
```



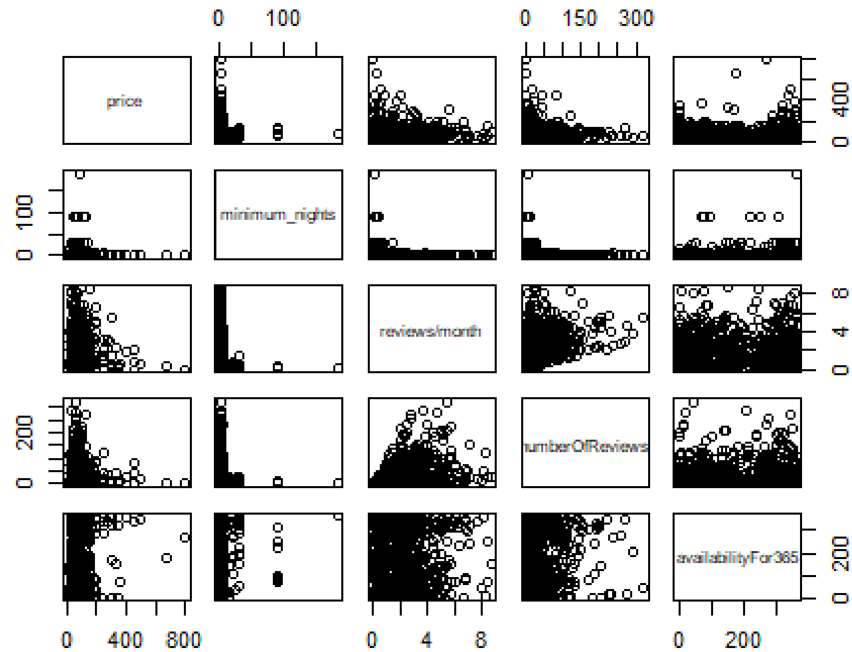

```
##QUEENS
pairs(data.table(
  airbnbQueens$price,
  airbnbQueens$minimum_nights,
  airbnbQueens$reviews_per_month,
  airbnbQueens$number_of_reviews,
  airbnbQueens$availability_365), labels = diagnolcol)
```




```
##Staten Island
pairs(data.table(
  airbnbStatenIsland$price,
  airbnbStatenIsland$minimum_nights,
  airbnbStatenIsland$reviews_per_month,
  airbnbStatenIsland$number_of_reviews,
  airbnbStatenIsland$availability_365), labels = diagnolcol)
```




```r
##BRONX
pairs(data.table(
  airbnbBronx$price,
  airbnbBronx$minimum_nights,
  airbnbBronx$reviews_per_month,
  airbnbBronx$number_of_reviews,
  airbnbBronx$availability_365), labels = diagnolcol)
```




```
pairs(data.table(airbnbBronx$price,
                 airbnbBronx$minimum_nights,
                 airbnbBronx$reviews_per_month,
                 airbnbBronx$number_of_reviews,
                 airbnbBronx$availability_365), labels = diagnolcol)

 ########3 ********** TESTS ********** ######
attach(airbnbCleaned)
#Tests
#T -test for price against different boroughs

with(data=airbnbCleaned,t.test(price[neighbourhood_group=="Manhattan"],price[
neighbourhood_group=="Brooklyn"],var.equal=TRUE))

##
##  Two Sample t-test
##
## data:  price[neighbourhood_group == "Manhattan"] and
price[neighbourhood_group == "Brooklyn"]
## t = 39.869, df = 33003, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  53.53904 59.07535
## sample estimates:
## mean of x mean of y
##  174.9481  118.6409
```

```r
##  P - value is small , it shows less correlation

with(data=airbnbCleaned,t.test(price[neighbourhood_group=="Queens"],price[nei
ghbourhood_group=="Bronx"],var.equal=TRUE))

##
##  Two Sample t-test
##
## data:  price[neighbourhood_group == "Queens"] and
price[neighbourhood_group == "Bronx"]
## t = 5.1808, df = 5377, p-value = 2.291e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   8.690078 19.270338
## sample estimates:
## mean of x mean of y
##  93.51621  79.53600

##  P - value is small , it shows less correlation

#Levene test for prices and neighbourhood_group

leveneTest(price ~ neighbourhood_group, data=airbnbCleaned)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     4  235.07 < 2.2e-16 ***
##       38692
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## the test shows homogeneity

detach(airbnbCleaned)
```