MA 615 Final Project - Boston Airbnb

Shiyu Zhang

November, 2018

A. Abstract

In this project, I chose the Airbnb dataset from Boston Area. The methodology of this final project is to test the reivews and price variables to see if they follow the Benford Law and explore further details of the output. In addition, this project also includes Modelings, EDA, Data Visualizations, Leaflet Mapping and Shiny Dashboard Building.

B. Project Background

I have been using Airbnb for over three years and it has become a popular way of travelling. I have witness Airbnb develop from an unknown website to the most popular travelling website during the past several years. Many people choose Airbnb instead of hotels not only for its lower price and convenient location, but also for its humanness – travelers are able to make connections with people from all around the world. What's more, travelers are provided with more unique options compare to hotels - houses, condos, apartments, castles, houseboats, tree houses, barns, mansions, even caves! Therefore, these unique properties of Airbnb inspired me to explore more about it. For example, what the factors may have an impact on the ratings, or, what is the relationship between the occupancy rate and the neighborhood of an Airbnb apartment, etc.

C. Dataset Source

http://tomslee.net/airbnb- data-collection-get-the-data

D. Data Cleaning

```
library(dplyr)
library(esquisse)
library(ggplot2)
library(sqldf)
library(tidyr)
library(data.table)
library(arm)
library(knitr)
library(plyr)
library(leaflet)
library(webshot)
#import data
Boston.airbnb<-read.csv("tomslee_airbnb_boston_0649_2016-11-21.csv")
# replace all N/A with 0
Boston.airbnb[is.na(Boston.airbnb)] <- 0
# Remove unrelevant columns
Boston.data<-Boston.airbnb[, c(-4,-9)]
```

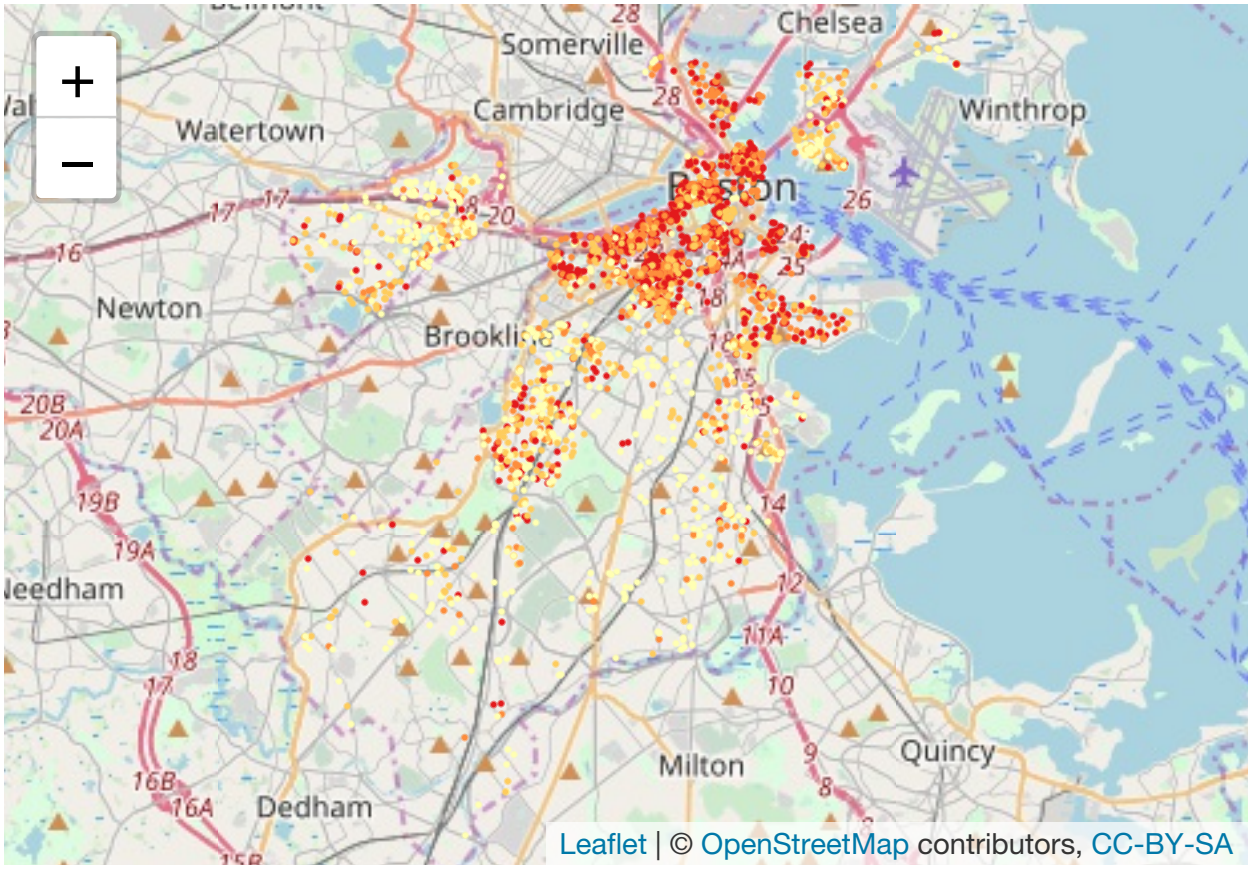
```r
#remove 0 review properties
Boston.data<-filter(Boston.data, reviews >0)
Boston.data<-filter(Boston.data, overall_satisfaction >0)
```

E. Dataset Structure & Overview

After the data cleaning process, the new dataset's structure is as follows:

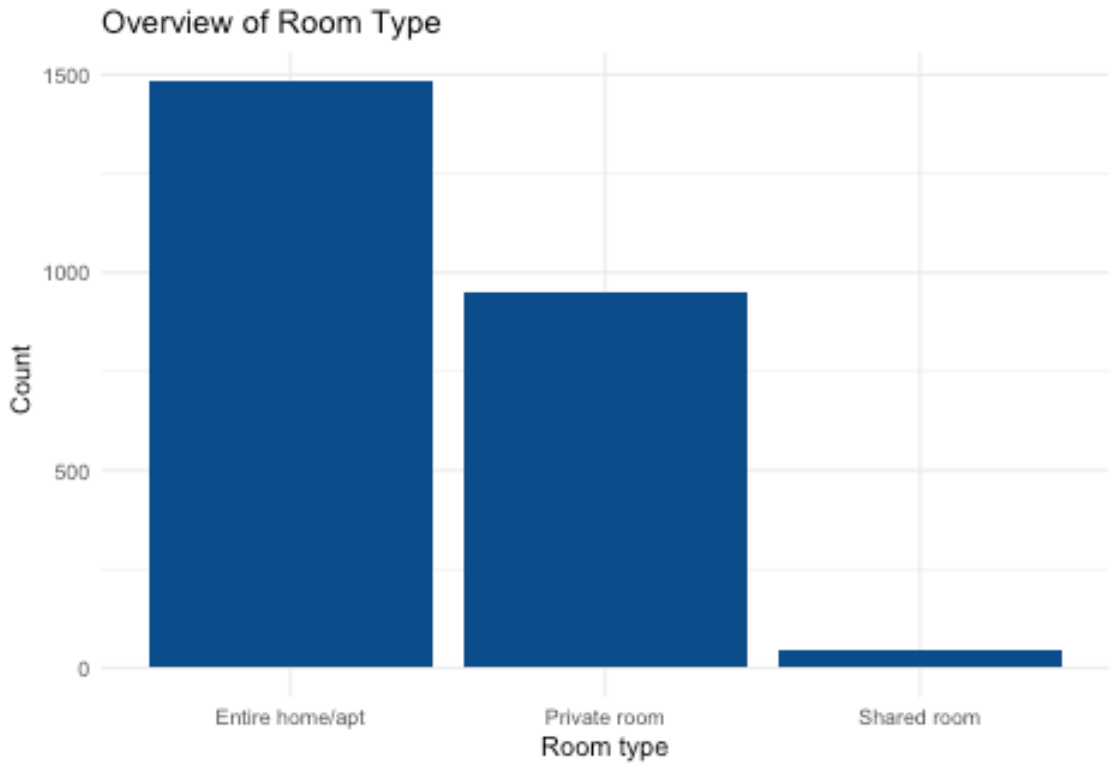
Properties Location Overview

```r
knitr::opts_chunk$set(echo = TRUE,out.width="0.9\\linewidth",dev="png",fig.align  = 'center')

pal <- colorQuantile(
  palette = "YlOrRd",
  domain = Boston.data$price
)
leaflet(Boston.data) %>% addTiles() %>%
  addCircles(lng = ~longitude, lat = ~latitude, weight = 1,
    popup = ~price, radius = 50,
    color = ~pal(price), fillOpacity = 1)
```



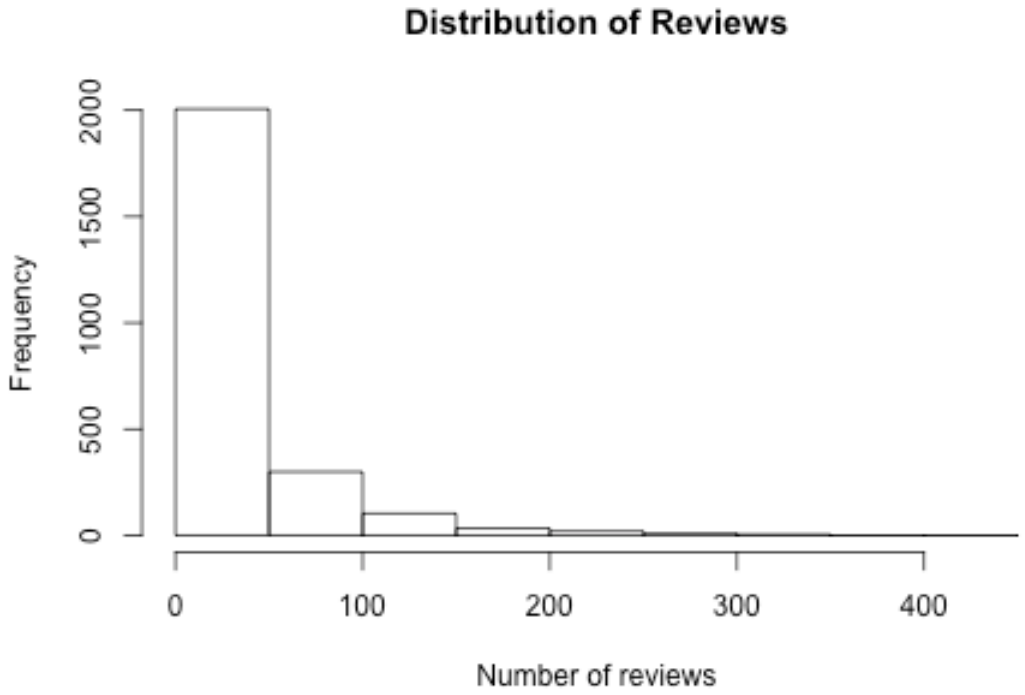
Room type overview

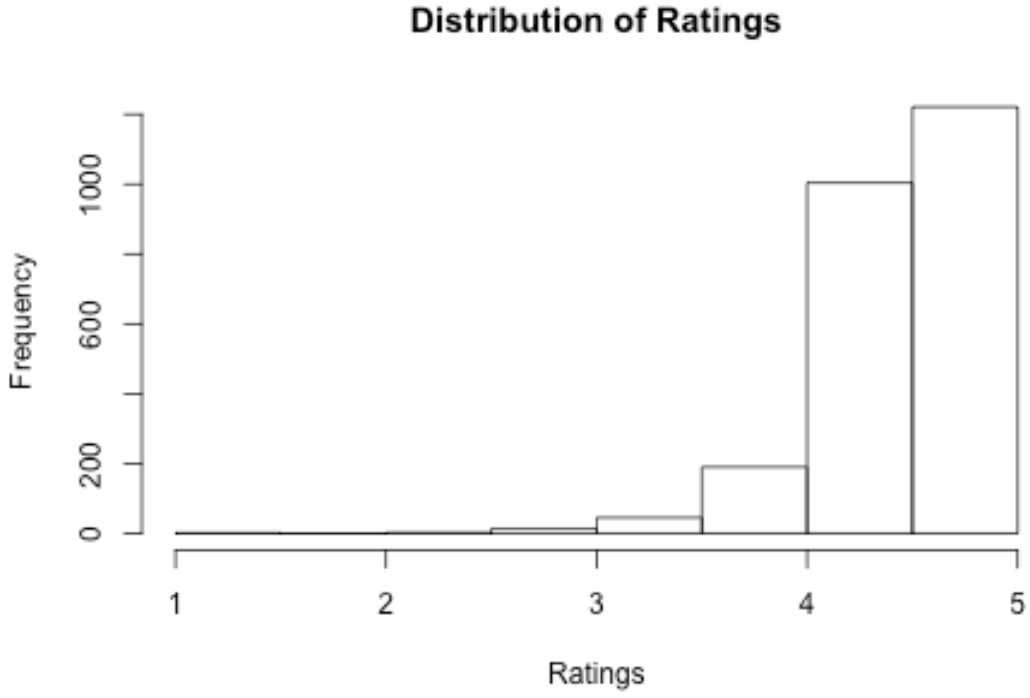
```
##
## Entire home/apt    Private room     Shared room
##            1483             947              46
```



From the output, we can see that in Boston area, entire home/apt is the most common type of properties for rent on the website, then is the private room. Shared room is the least common way on the website.

Check ratings and number of reviews





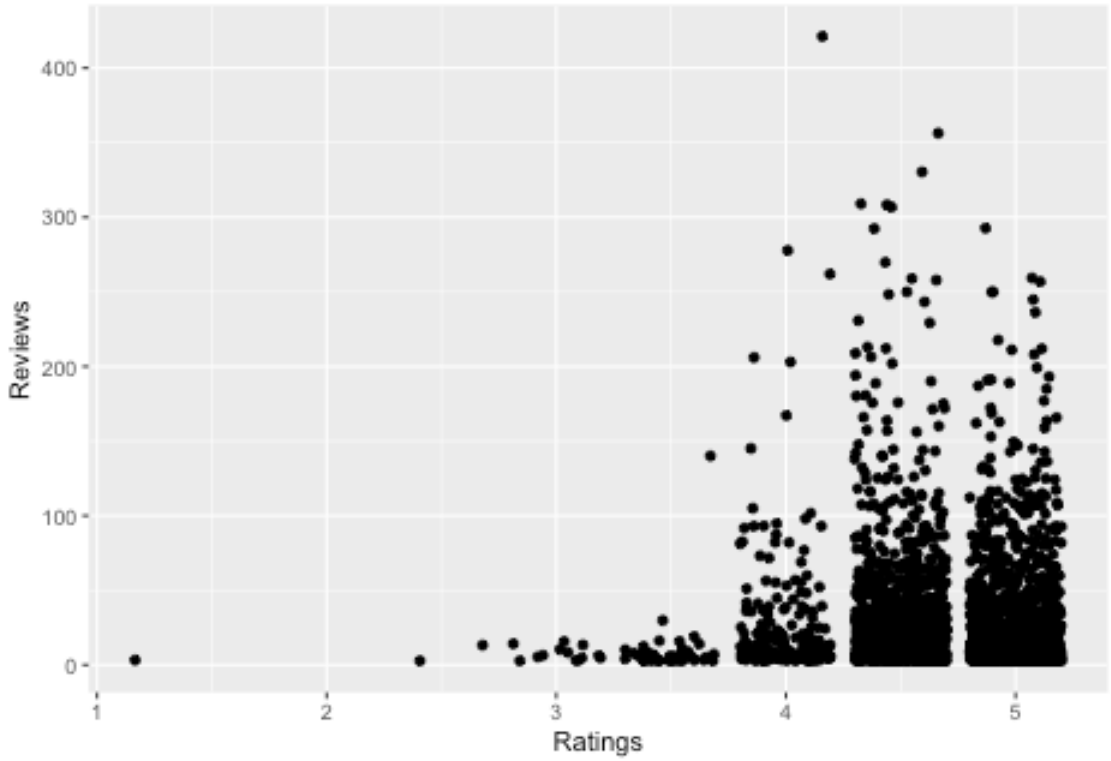

```
## [1] 0.8101777
```

```
## [1] 421
```

From the histogram for the distribution of ratings (overall_satisfaction), we can see that most of the ratings for Airbnb properties in Boston are above 4.0, the data shows right skewness. The distribution of reviews shows a left skewness. From the frequency table we can see that the majority number of reviews are less than 50 in Boston area. There are 3127 rooms in our data after cleaning, from the output, 84.9% of total Airbnb rooms have less than 50 reviews, while the maximum reviews for a room is 421.

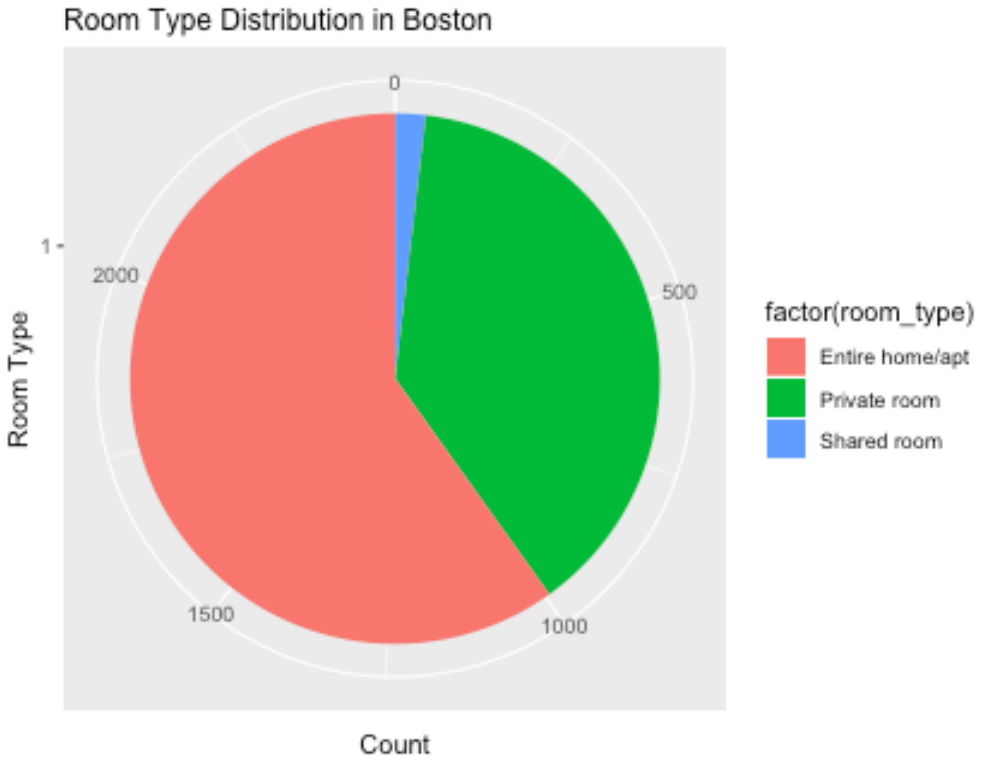
F. Data Visualization

Visualize relationship between overall satisfactions and number of reviews



Based on these output, we can tell that in general, higher ratings tend to have more reviews.

Distribution of room type

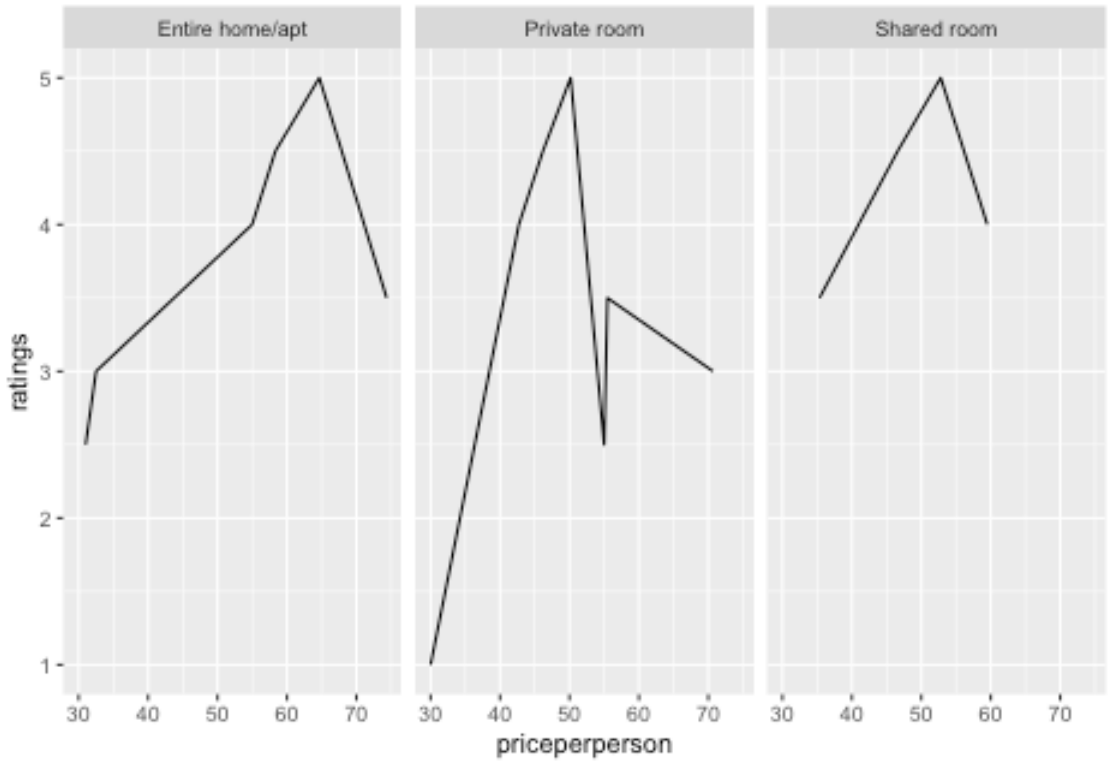


As we can tell from the pie chart, the entire home/apt has the majority proportion of the whole room types. Private room comes the next, and shared room has the least proportion among all the room types.

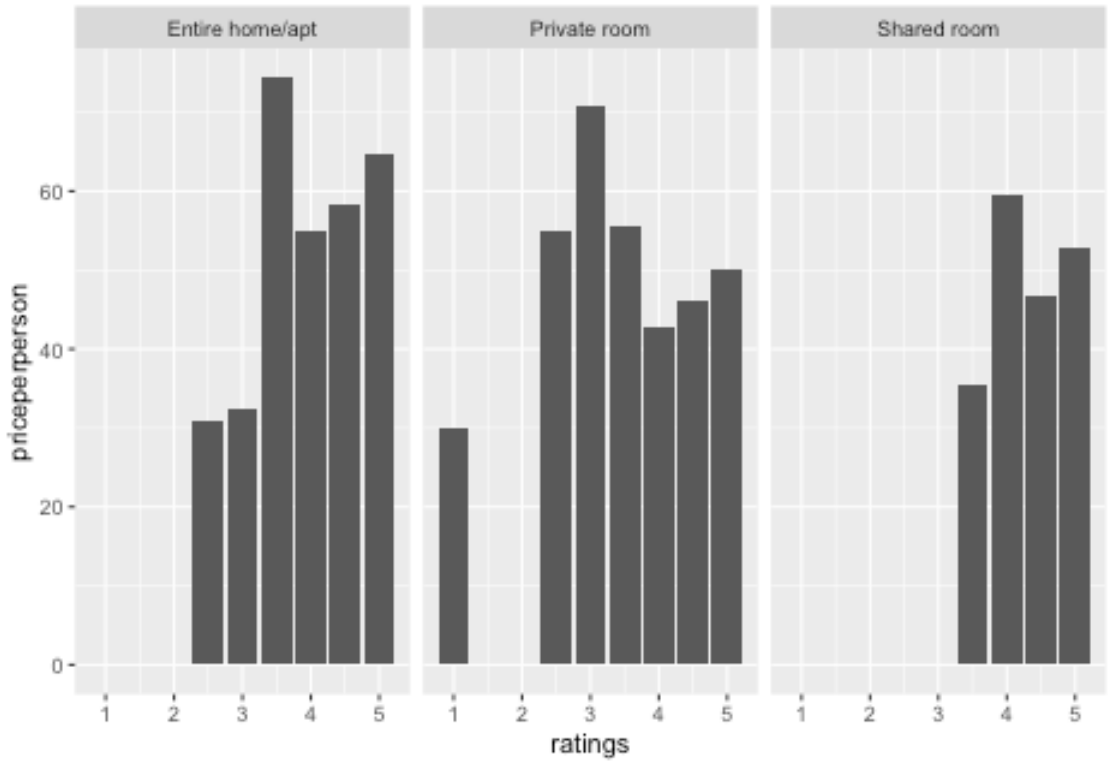
Explore Relationship Between Price per Person and Ratings & Room type

Calculate Price per Person

```
Boston.data$priceperperson <- (Boston.data$price)/(Boston.data$accommodates)
a1 <- aggregate( priceperperson ~ room_type+overall_satisfaction, Boston.data, mean )
a1 <- as.data.frame(a1)
names(a1) <- c("room_type","ratings","priceperperson")
ggplot(data=a1,aes(x=priceperperson,y=ratings))+geom_line()+facet_wrap(.~room_type)
```




```
ggplot(data=a1,aes(x=ratings,y=priceperperson))+geom_bar(stat="identity")+facet_wrap(.~room_type)
```



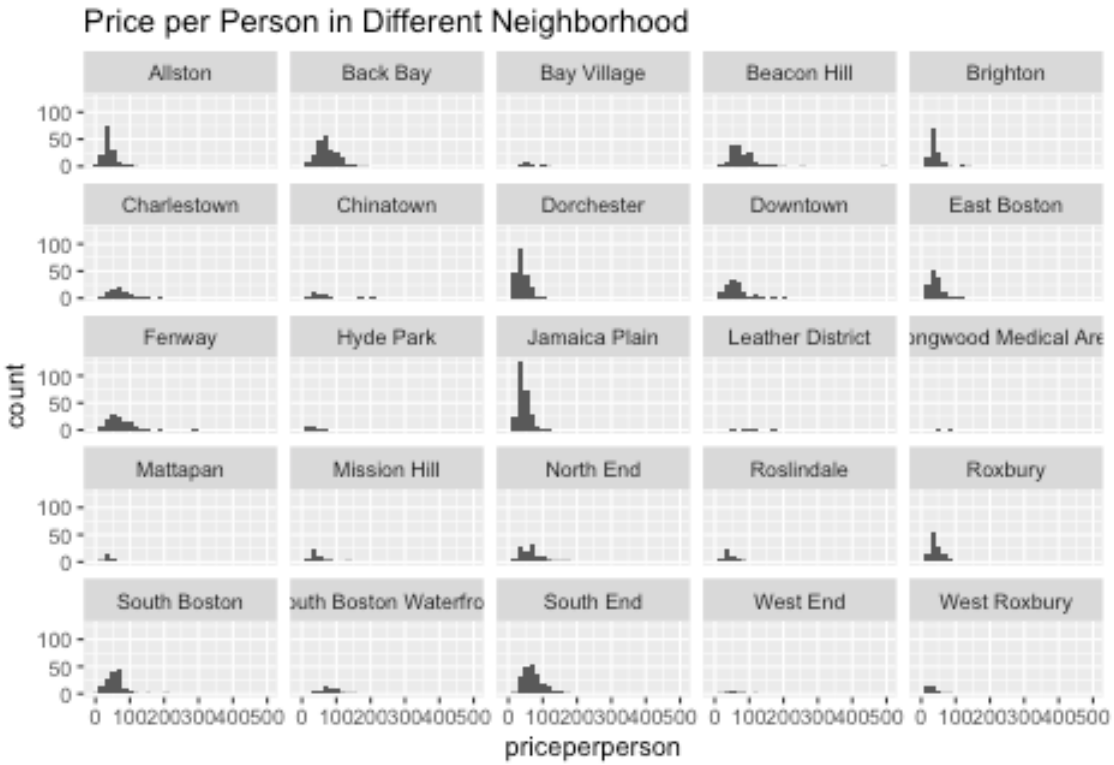

```
kable(a1, caption = "Average Price by Room Type")
```

Table 1: Average Price by Room Type

room_type	ratings	priceperperson
Private room	1.0	30.00000
Entire home/apt	2.5	31.00000
Private room	2.5	55.00000
Entire home/apt	3.0	32.50000
Private room	3.0	70.68750
Entire home/apt	3.5	74.35167
Private room	3.5	55.46875
Shared room	3.5	35.33333
Entire home/apt	4.0	54.97981
Private room	4.0	42.71354
Shared room	4.0	59.50000
Entire home/apt	4.5	58.32589
Private room	4.5	46.17653
Shared room	4.5	46.58333
Entire home/apt	5.0	64.65567
Private room	5.0	50.17040
Shared room	5.0	52.78947

From the two output, we can tell that for different room type, for example, for entire home/apt, the most ratings are 3.5. For private room, ratings 3 is the most common one and ro the shared room, ratings 4 is more common. So we can tell from the graph that ratings are somehow related with the room type, i will explore further later in the model part.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



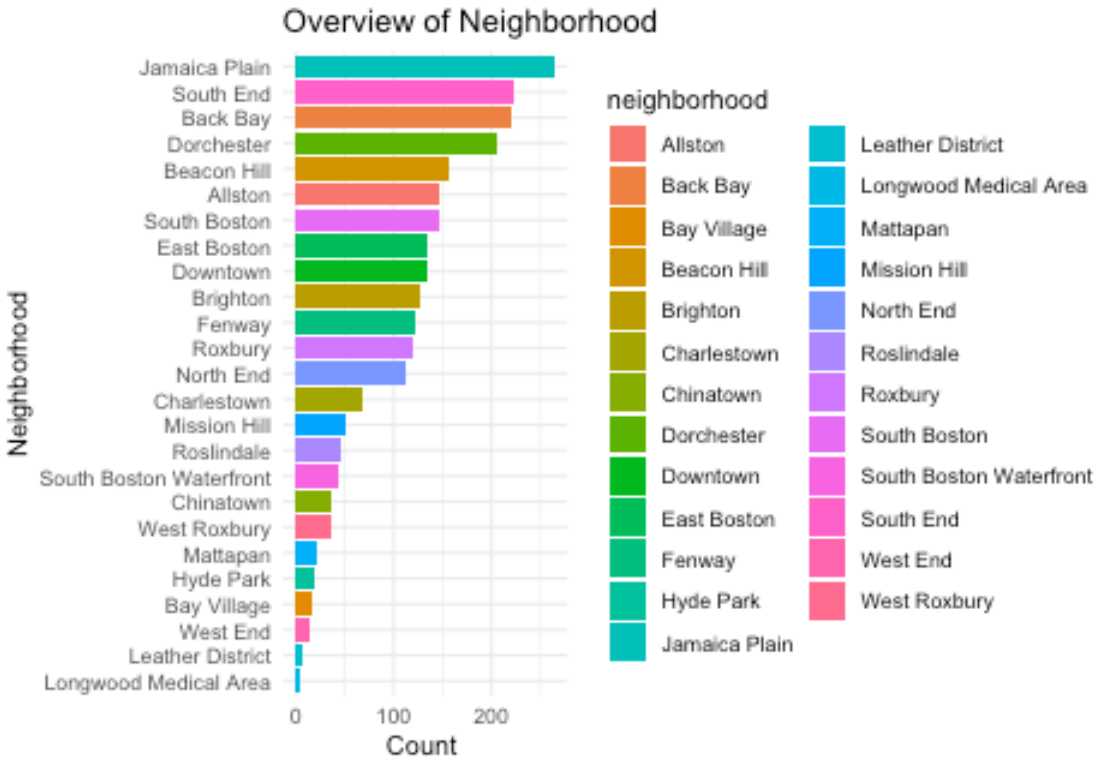
From the graph, we can see that the price in majority neighborhood ranges from 50 to 100 dollars per person,

some preperties in Back Bay could be a little bit more expensive. Most of the properties in Jamaica Plain and Dorchester are 80 dollars per person.

Now we want to explore the relationship between the ratings and properties' location.

Neighborhood overview

```
## List of 1
##  $ axis.text.x:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : num 60
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```



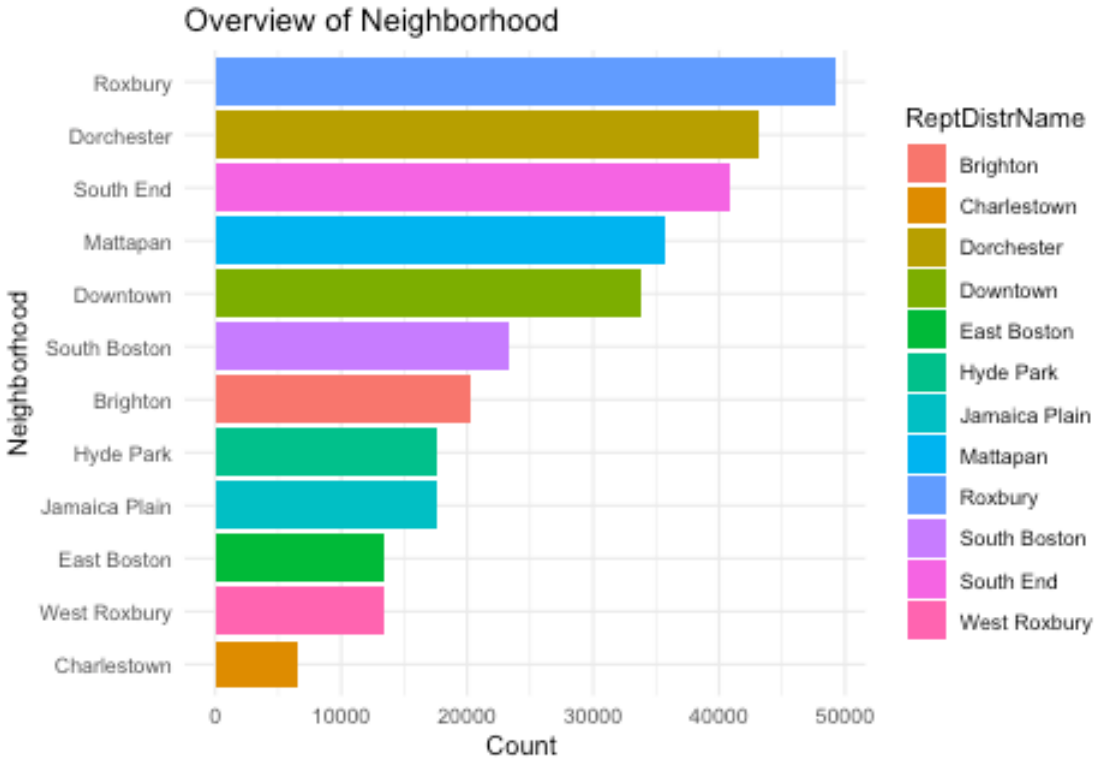
From this bar plot and r output, we can see that the top 5 neighborhood for rent on Airbnb in Boston area are Jamaica Plain, Back Bay, Allston, Dorchester and Beacon Hill. On contrast, Longwood, Leather District, Bay Village, West End and Roxbury are the least popular neighborhood for Airbnb in Boston area. We can see that the ratings are highly related to the neighborhood of the properties, in order to explore

further about reasons behind it, i introduced another dataset - Boston crime incident data. dataset source : https://data.boston.gov/dataset/ crime-incident-reports-august-2015-to-date-source-new-system.

```
crime<-read.csv("crime_incident_reports.csv")

# replace district code with names
distrName = c(
A1 = 'Downtown',
A15= 'Charlestown',
A7= 'East Boston',
B2= 'Roxbury',
B3= 'Mattapan',
C6= 'South Boston',
C11= 'Dorchester',
D4= 'South End',
D14= 'Brighton',
E5= 'West Roxbury',
E13= 'Jamaica Plain',
E18= 'Hyde Park',
HTU= 'Human Traffic Unit'
)
crime$ReptDistrName = as.factor(distrName[as.character(crime$DISTRICT)])
crime$DISTRICT = NULL

data.crime=na.omit(crime)
dff2= count(data.crime, 'ReptDistrName')

pic2 <- ggplot(dff2, aes(x = reorder(ReptDistrName, freq), y = freq, fill = ReptDistrName)) +
         geom_bar(stat = "identity") +
  labs(title = 'Overview of Neighborhood',
    x = 'Neighborhood',
    y = 'Count ') +
  theme_minimal()+coord_flip()
 theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

```
## List of 1
##  $ axis.text.x:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : num 60
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

```
print(pic2)
```



Compared to the two graphs above, we can see that ratings is highly related to the neighborhood of the property, more specific, in the neighborhood where the crime rates are lower, the ratings tends to be higher."

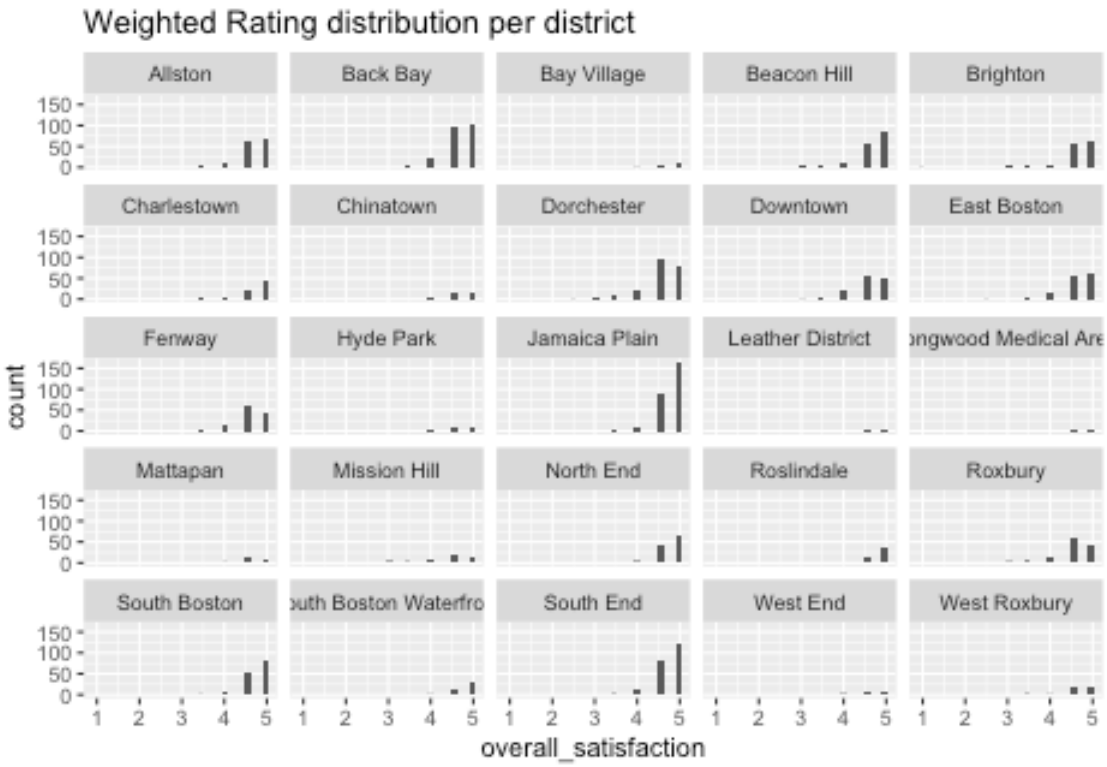
To sum up, we can say that ratings is highly related to number of reviews, price , accomodates, minimum stays and the location of the property. It has a moderate relationship with the room type."

G. EDA

Prior to the application for multilevel model, I think doing some initial EDA is helpful for a better understanding of relationship between independent variables and dependents variables.

1.Distribution of ratings in different districts

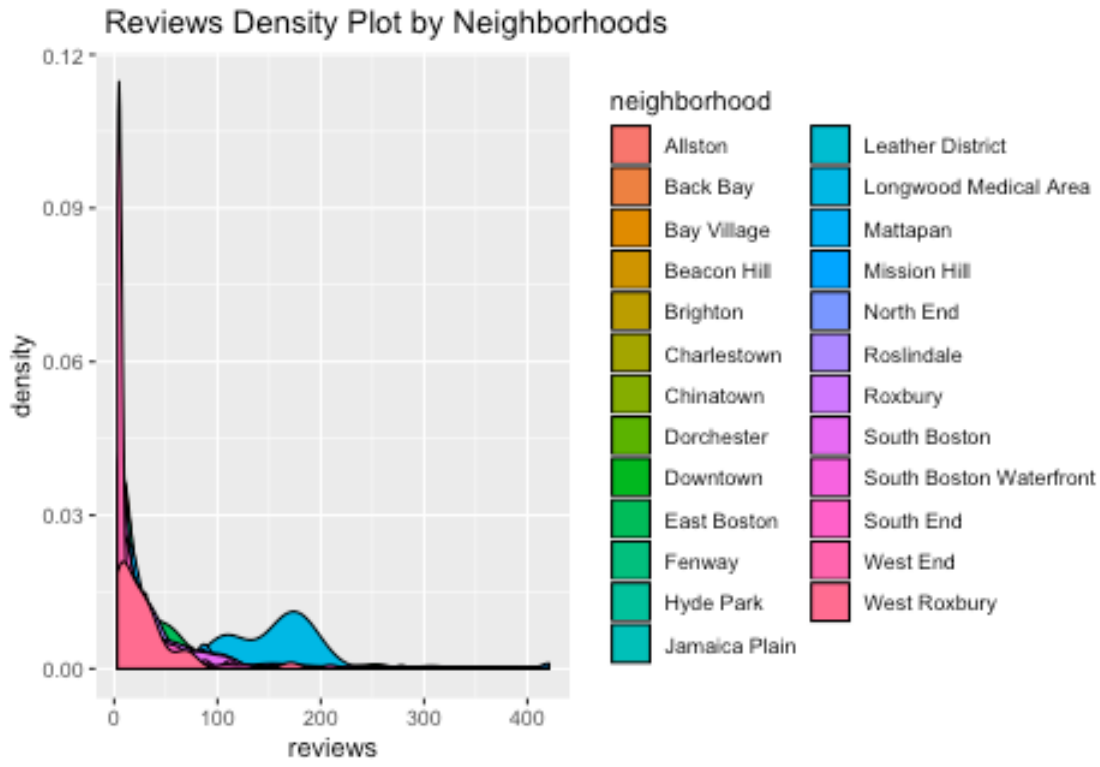
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



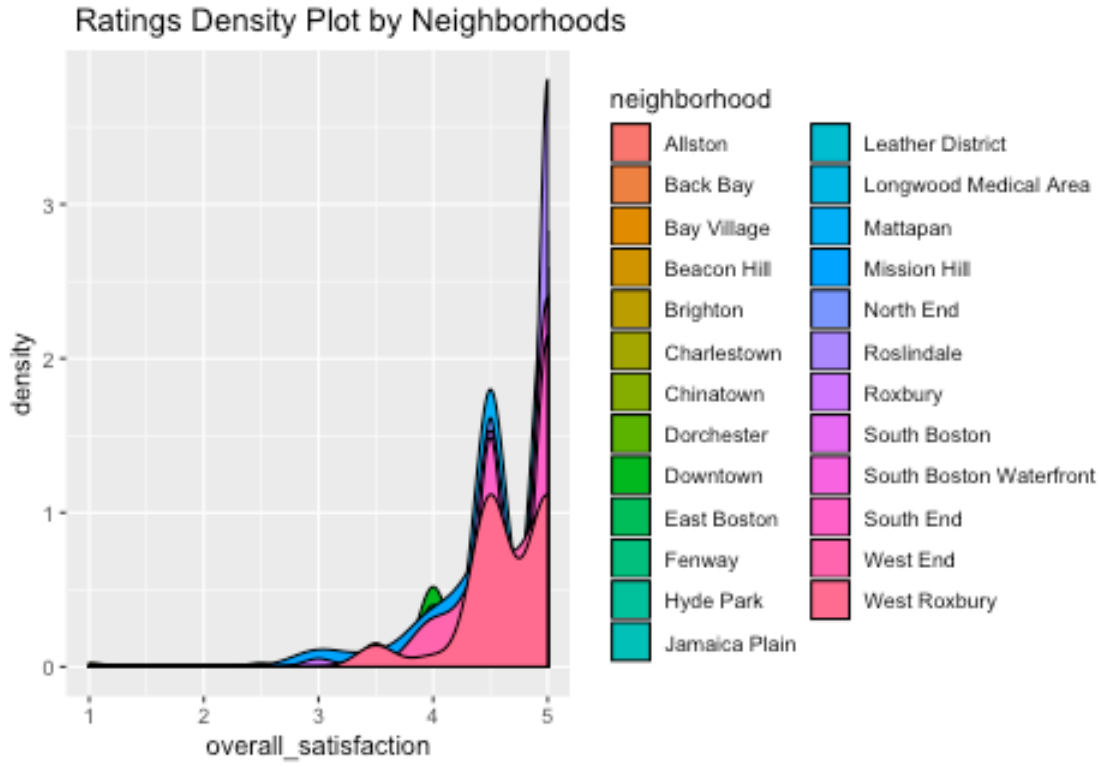
As we can see from the output, it is more clear that Bay Village,Leather District, Longwood Medical area, west end and west roxbury have very few reviews or ratings compared to the crime plot i showed in the previous part. we can say that the area where has more crime tend to have few ratings/ reviews. On the other hand, Allston, Back Bay, Jamacia Plain and South End have most ratings/ reviews.

2. Density Plot

```
ggplot(data=Boston.data, aes(x=reviews, fill=neighborhood))+geom_density()+ ggtitle(" Reviews Density P
```

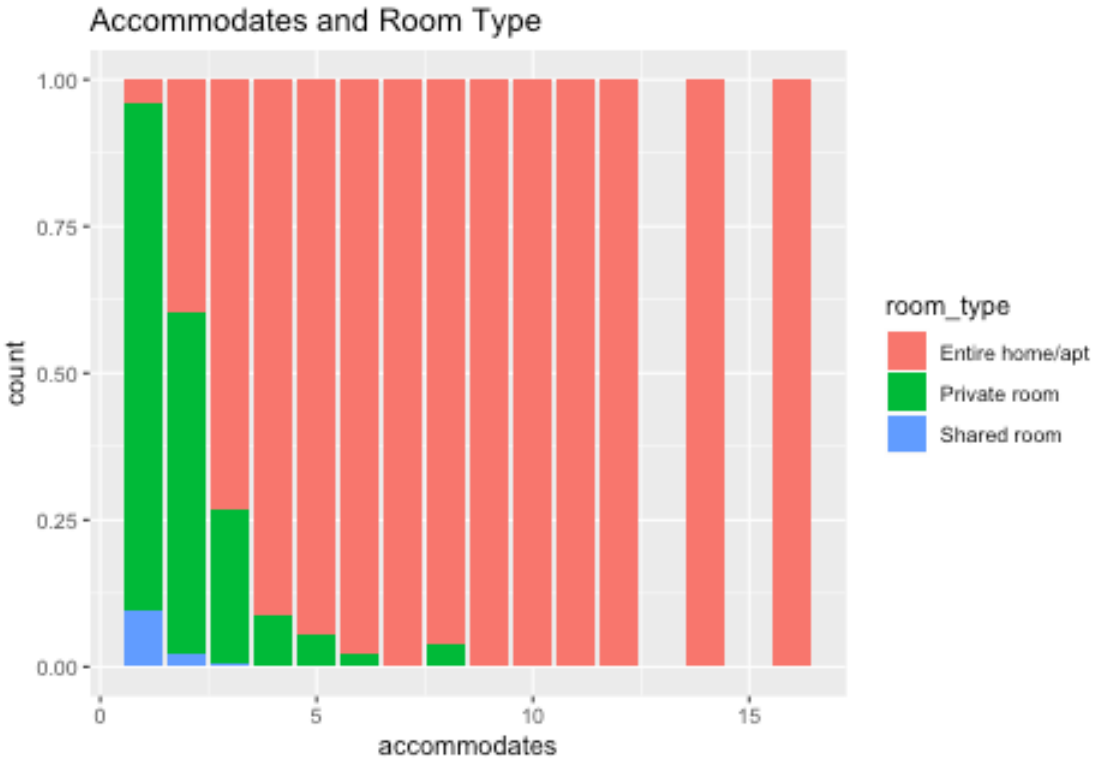



```
ggplot(data=Boston.data, aes(x=overall_satisfaction, fill=neighborhood))+geom_density()+ ggtitle(" Rati
```

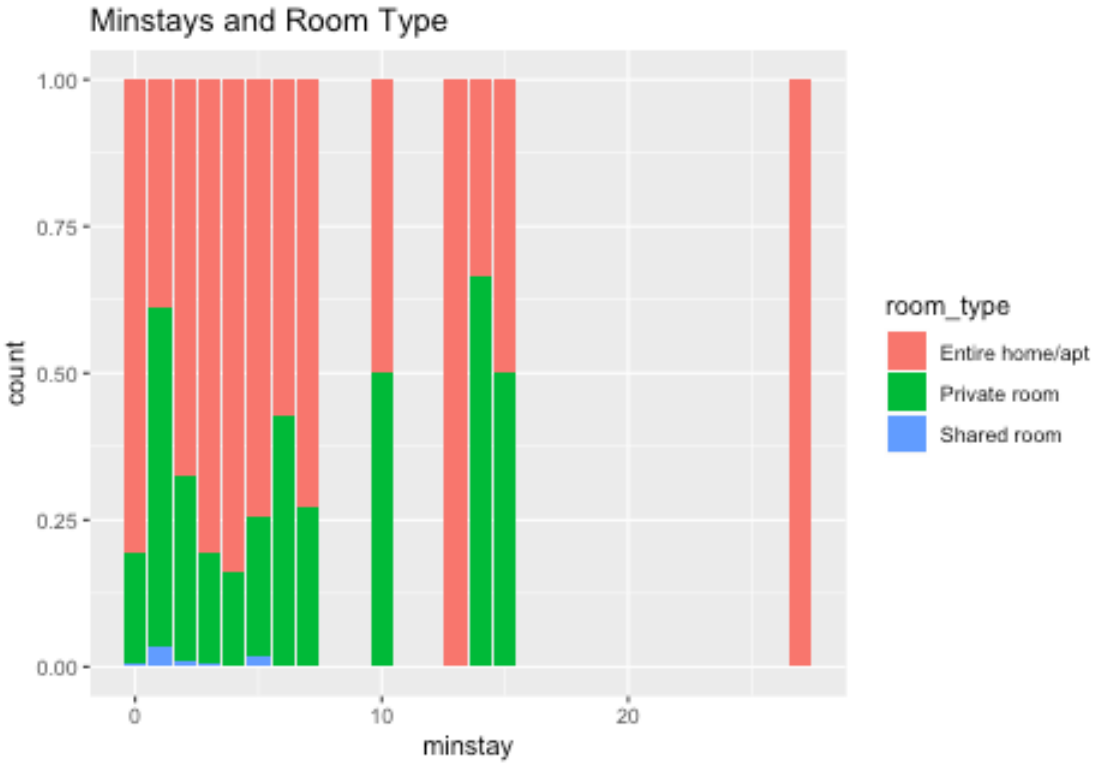


we can see that the distribution of ratings and number of reviews are different between neighborhoods.

3. Relationship between accomodates and room type

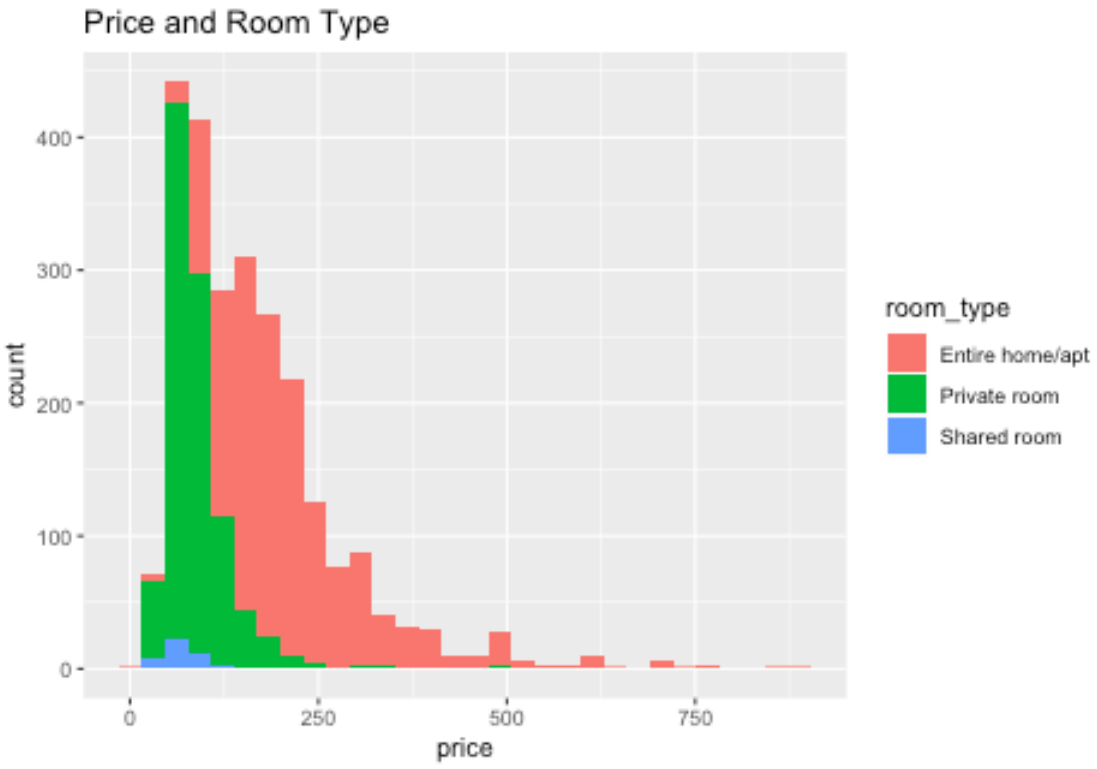


4. Relationship between mininum stay dates and room type



5. relationship between prices per night versus room type

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Benford Analysis for Price and Review Variables

```
library(benford.analysis)
library(BenfordTests)
```

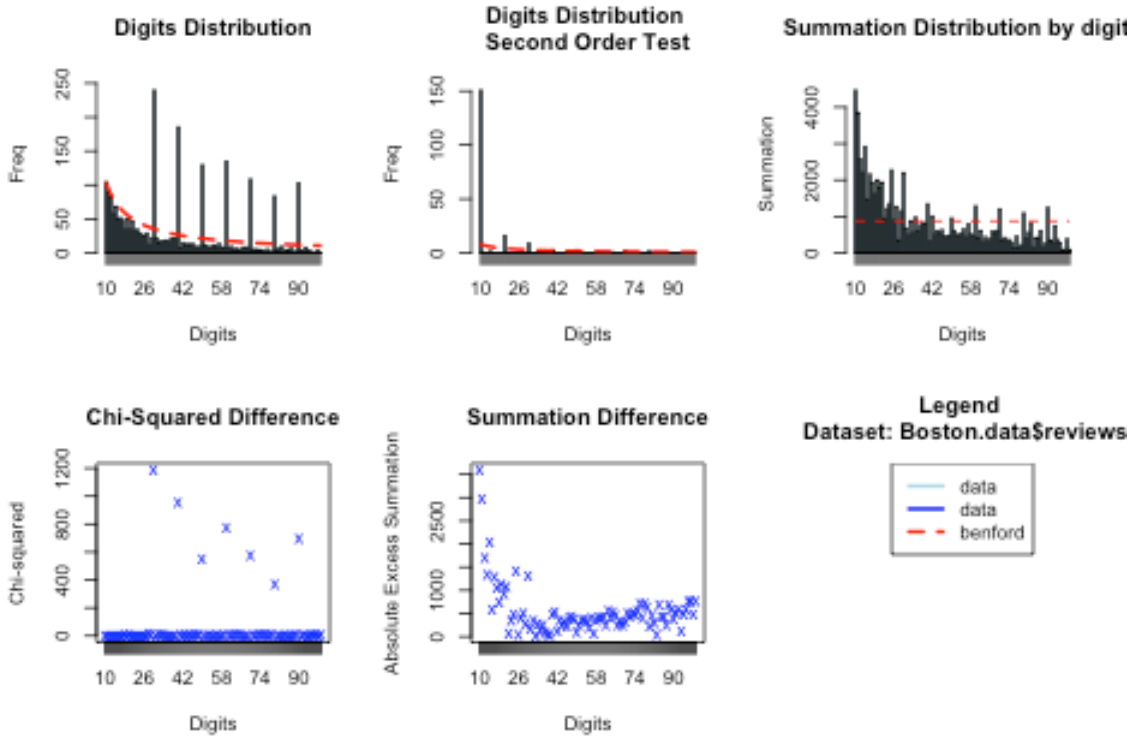
1. Benford Analysis for Reviews

```
#Get the the statistics of the first TWO digits of a benford object
ben_review1 <- benford(Boston.data$reviews, number.of.digits=2)
ben_review1
```

```
##
## Benford object:
##
## Data: Boston.data$reviews
## Number of observations used = 2476
## Number of obs. for second order = 192
## First digits analysed = 2
##
## Mantissa:
##
##    Statistic Value
##         Mean  0.51
##          Var  0.08
##  Ex.Kurtosis -1.07
```

```
##     Skewness -0.20
##
##
## The 5 largest deviations:
##
##   digits absolute.diff
## 1     30        204.74
## 2     40        159.45
## 3     60        117.23
## 4     50        108.71
## 5     70         93.75
##
## Stats:
##
##  Pearson's Chi-squared test
##
## data:  Boston.data$reviews
## X-squared = 5551.3, df = 89, p-value < 2.2e-16
##
##
##  Mantissa Arc Test
##
## data:  Boston.data$reviews
## L2 = 0.011057, df = 2, p-value = 1.288e-12
##
## Mean Absolute Deviation: 0.007612088
## Distortion Factor: -19.64199
##
## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!
```

```
plot(ben_review1)
```



The graph shows that the majority of the data for reviews match the expected trend expect several parts. For the example, that the lead digit of 3 occurred more than 4 times as often as is expected, this is the situation for leading digits of 4,5,6,7 and 8. I think that data spikes do not necessarily signal underlying problems such as fraud, but they do alert to the possibility of such problems. I have check the reviews policy for Airbnb on the website, and I found that a good number of reviews can help the host's ratings improve and thus, land more bookings. So it is critical factor in the Airbnb owner valuation system, the potential reason why the leading digits such as 4,5,6,7 and 8 are pretty high is that Airbnb is very cautious about the views, and they had a full control of the number of views from going too much. I will explore the reason behind more in details in later analysis.

```
# Get MAD and Distortion Factor
MAD(ben_review1)
```

```
## [1] 0.007612088
```

```
dfactor(ben_review1)
```

```
## [1] -19.64199
```

```
#Get main stats from the object
mantissa(ben_review1)
```

```
##                statistic      values
## 1:         Mean Mantissa  0.51323749
## 2:          Var Mantissa  0.08041346
## 3: Ex. Kurtosis Mantissa -1.07470388
## 4:     Skewness Mantissa -0.19605117
```

```
# Get the Mantissa Arc test of the object
marc(ben_review1)
```

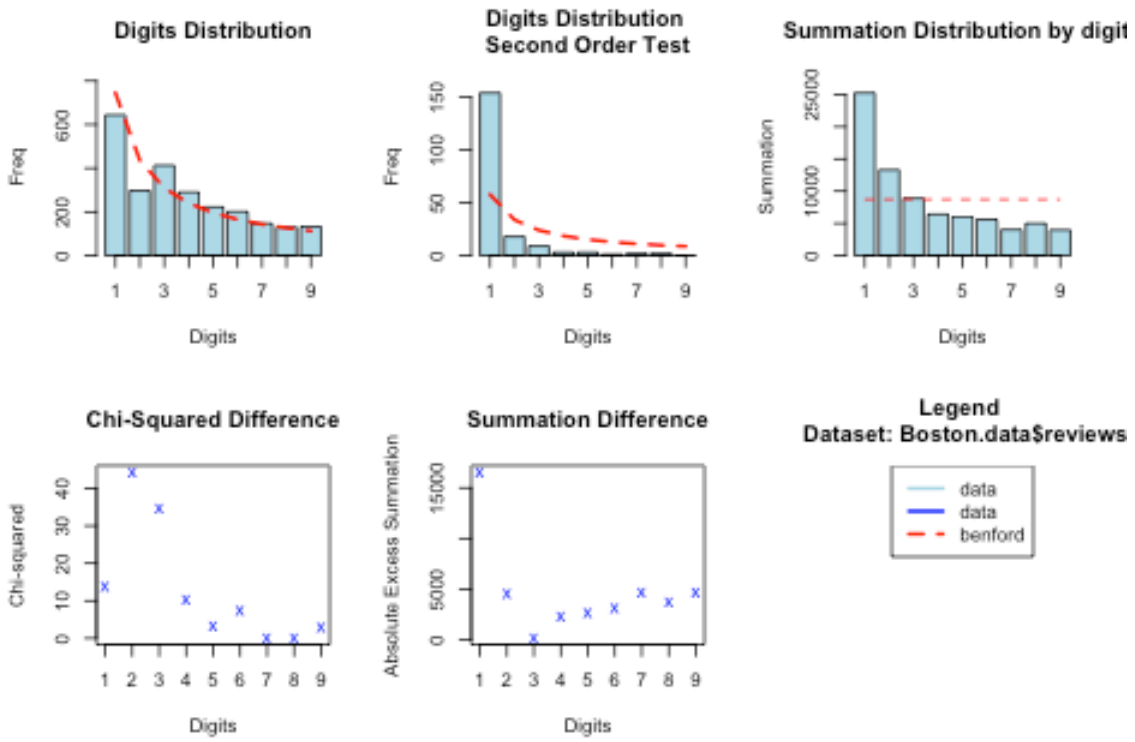
```
##
```

```
##  Mantissa Arc Test
##
## data:  Boston.data$reviews
## L2 = 0.011057, df = 2, p-value = 1.288e-12
```

```
#Gets the the statistics of the first ONE digits of a benford object

ben_review2 <- benford(Boston.data$reviews, number.of.digits=1)
ben_review2
```

```
##
## Benford object:
##
## Data: Boston.data$reviews
## Number of observations used = 2476
## Number of obs. for second order = 192
## First digits analysed = 1
##
## Mantissa:
##
##    Statistic Value
##         Mean  0.51
##          Var  0.08
##  Ex.Kurtosis -1.07
##     Skewness -0.20
##
##
## The 5 largest deviations:
##
##   digits absolute.diff
## 1      2        139.00
## 2      3        103.65
## 3      1        101.35
## 4      4         50.05
## 5      6         35.24
##
## Stats:
##
##  Pearson's Chi-squared test
##
## data:  Boston.data$reviews
## X-squared = 117.45, df = 8, p-value < 2.2e-16
##
##
##  Mantissa Arc Test
##
## data:  Boston.data$reviews
## L2 = 0.011057, df = 2, p-value = 1.288e-12
##
## Mean Absolute Deviation: 0.02157173
## Distortion Factor: -19.64199
##
## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!
```

```
plot(ben_review2)
```



From the output for the one digits test, we can see that the lead digit of 2 occurred only about 2/3 of as often as is expected, while the lead digit of 3 and 4 appeared much more often. I think it is common that most of the Airbnb properties have 30s or 40s reviews because unless the pretty popular ones.

```
#Probability of a digit sequence
prob.dig <- c()
for (i in 1:20){
  prob.dig[i] <- p.these.digits(i)}
prob.dig <- as.vector(prob.dig)

#Probability of a digit appear on the n-th position
p.this.digit.at.n(2,1) #probability of 2 appear on the first position
```

```
## [1] 0.1760913
```

```
#Find out suspect observations
suspects <- getSuspects(ben_review1, Boston.data)

#Creates a data frame with the first digits and the differences from Benford's Law in decreasing order
suspectsTable(ben_review1, by="absolute.diff")
```

```
##     digits absolute.diff
##  1:     30    204.740673
##  2:     40    159.447709
##  3:     60    117.225824
##  4:     50    108.705975
##  5:     70     93.747076
##  6:     90     91.117966
```

```
##  7:     80     70.641901
##  8:     28     24.734157
##  9:     18     23.139193
## 10:     13     20.689276
## 11:     15     18.399120
## 12:     41     17.912414
## 13:     20     17.464704
## 14:     32     17.089169
## 15:     25     15.174548
## 16:     24     14.896427
## 17:     26     14.582670
## 18:     29     14.454784
## 19:     34     14.170679
## 20:     33     14.101283
## 21:     22     13.799564
## 22:     16     13.190452
## 23:     48     13.172190
## 24:     43     12.720931
## 25:     47     12.639007
## 26:     17     12.463193
## 27:     36     12.462477
## 28:     31     12.139792
## 29:     76     12.056581
## 30:     63     11.934415
## 31:     49     11.724237
## 32:     78     11.698442
## 33:     45     11.634207
## 34:     65     11.417313
## 35:     35     11.292514
## 36:     53     11.099896
## 37:     56     11.032588
## 38:     23     10.764912
## 39:     84     10.725748
## 40:     79     10.526130
## 41:     61     10.485152
## 42:     52     10.482774
## 43:     42     10.302653
## 44:     75     10.242766
## 45:     66     10.170427
## 46:     44     10.165357
## 47:     88     10.150560
## 48:     96     10.143241
## 49:     27     10.106606
## 50:     59     10.072915
## 51:     71     10.039590
## 52:     97     10.028949
## 53:     99      9.807258
## 54:     54      9.731070
## 55:     64      9.671855
## 56:     73      9.630336
## 57:     74      9.433950
## 58:     58      9.381869
## 59:     46      9.125905
## 60:     89      9.014797
```

```
## 61:     51      8.880523
## 62:     72      8.832140
## 63:     92      8.625120
## 64:     95      8.259926
## 65:     81      8.194192
## 66:     77      7.875201
## 67:     91      7.752173
## 68:     68      7.698281
## 69:     37      7.676716
## 70:     85      7.576905
## 71:     86      7.431504
## 72:     94      7.379081
## 73:     55      7.375536
## 74:     62      7.205377
## 75:     67      6.930856
## 76:     98      6.916978
## 77:     69      6.472398
## 78:     39      6.224572
## 79:     82      6.034258
## 80:     38      5.931782
## 81:     87      5.289427
## 82:     14      5.188941
## 83:     57      4.701569
## 84:     19      4.156353
## 85:     11      3.564477
## 86:     93      3.500785
## 87:     12      3.070975
## 88:     21      3.023584
## 89:     83      2.878156
## 90:     10      2.511712
##     digits absolute.diff
```

From the suspects table output, we can see that the digits 2,3,1 have greater deviation from the expected distribution.

```
#Chi-sqaure test
chisq(ben_review1)
```

```
##
##  Pearson's Chi-squared test
##
## data:  Boston.data$reviews
## X-squared = 5551.3, df = 89, p-value < 2.2e-16
```

The chi-square test is a statistical test that measures how well the data distribution from a sample matches a hypothetical distribution dictated by theory. Based on the chisq output, the p-value is small so we reject null hypothesis, which means that the distances between data points and benford points are large.

BenfordTests Package for Reviews

```
#JP Sqaure test
jpsq.benftest(x=Boston.data$reviews,digits = 2, pvalmethod = "simulate", pvalsims = 10000)
```

```
##
##  JP-Square Correlation Statistic Test for Benford Distribution
```

```
##
## data:  Boston.data$reviews
## J_stat_squ = 0.17469, p-value < 2.2e-16
```

```
#Joenssen's JP-square Test for Benford's Law
#The result signifys that the square correlation between reviews and pbenf(2) is not zero.
# Euclidean Distance Test for Benford's Law
edist.benftest(Boston.data$reviews)
```

```
##
##  Euclidean Distance Test for Benford Distribution
##
## data:  Boston.data$reviews
## d_star = 4.2693, p-value < 2.2e-16
```

```
# The p-value is smaller than 0.05 so that we reject the null hypothesis. Therefore, the goodness-of-fi

#rbenf
rbenf(10) #10 observations
```

```
##  [1] 4.477359 2.846969 9.688559 1.950038 6.205361 2.698998 6.181275
##  [8] 2.230652 1.663415 5.921319
```

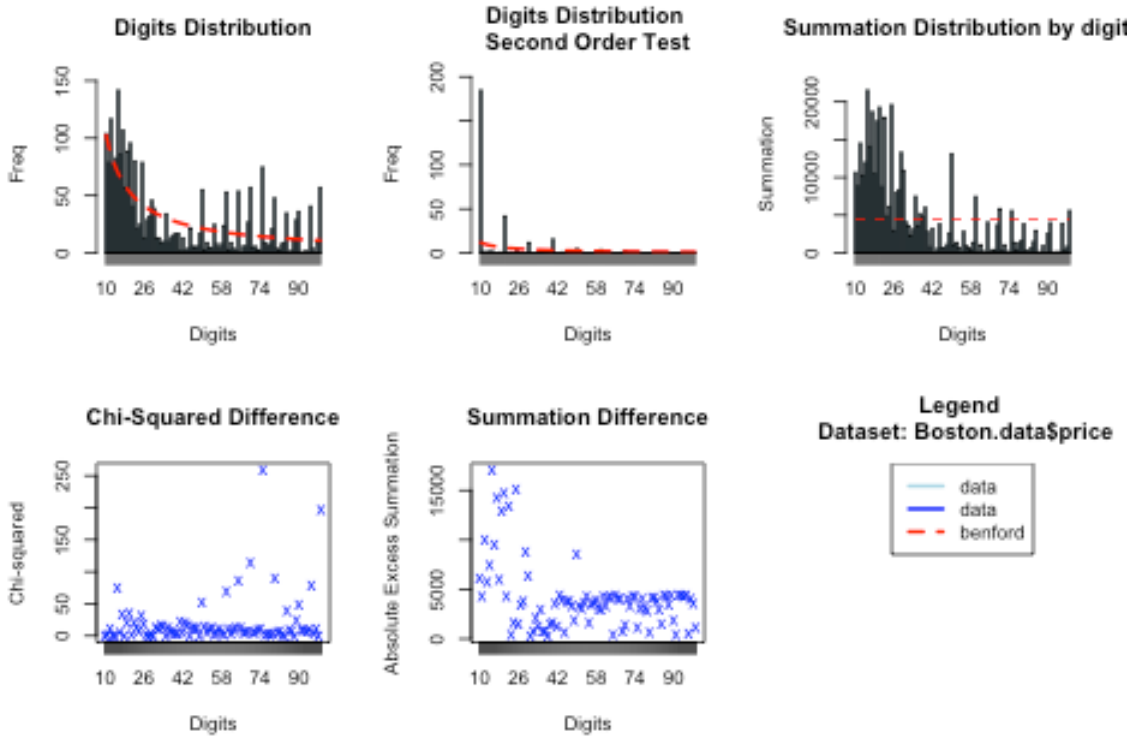
2. Benford Analysis for Price

```
#Get the the statistics of the first TWO digits of a benford object
ben_price1 <- benford(Boston.data$price, number.of.digits=2)
ben_price1
```

```
##
## Benford object:
##
## Data: Boston.data$price
## Number of observations used = 2476
## Number of obs. for second order = 282
## First digits analysed = 2
##
## Mantissa:
##
##    Statistic  Value
##         Mean  0.474
##          Var  0.095
##  Ex.Kurtosis -1.340
##     Skewness  0.248
##
##
## The 5 largest deviations:
##
##   digits absolute.diff
## 1     15         72.60
## 2     75         60.76
## 3     99         46.19
## 4     17         45.54
## 5     20         43.54
##
```

```
## Stats:
##
##  Pearson's Chi-squared test
##
## data:  Boston.data$price
## X-squared = 1826.5, df = 89, p-value < 2.2e-16
##
##
##  Mantissa Arc Test
##
## data:  Boston.data$price
## L2 = 0.039805, df = 2, p-value < 2.2e-16
##
## Mean Absolute Deviation: 0.006796536
## Distortion Factor: -1.957824
##
## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!
```

```
plot(ben_price1)
```



The graph shows that the majority of the data for price doesn't perfectly match the expected trends. For the example, that the lead digit of 10 or 23 occurred more than twice as often as is expected, however the digits 36 was only a half of the expected values. I think that data spikes do not necessarily signal underlying problems such as fraud, but they do alert to the possibility of such problems. I think that the avg price of one stay in Boston area is around 100 dollars to 200 dollars, that is the reason why spikes occurs on the digits 10 and 23.

```
# Get MAD and Distortion Factor
MAD(ben_price1)
```

```
## [1] 0.006796536
```

```
dfactor(ben_price1)
```

```
## [1] -1.957824
```

```
#Get main stats from the object
mantissa(ben_price1)
```

```
##                statistic      values
## 1:         Mean Mantissa  0.47370998
## 2:          Var Mantissa  0.09549829
## 3: Ex. Kurtosis Mantissa -1.34011392
## 4:     Skewness Mantissa  0.24795730
```

```
# Get the Mantissa Arc test of the object
marc(ben_price1)
```

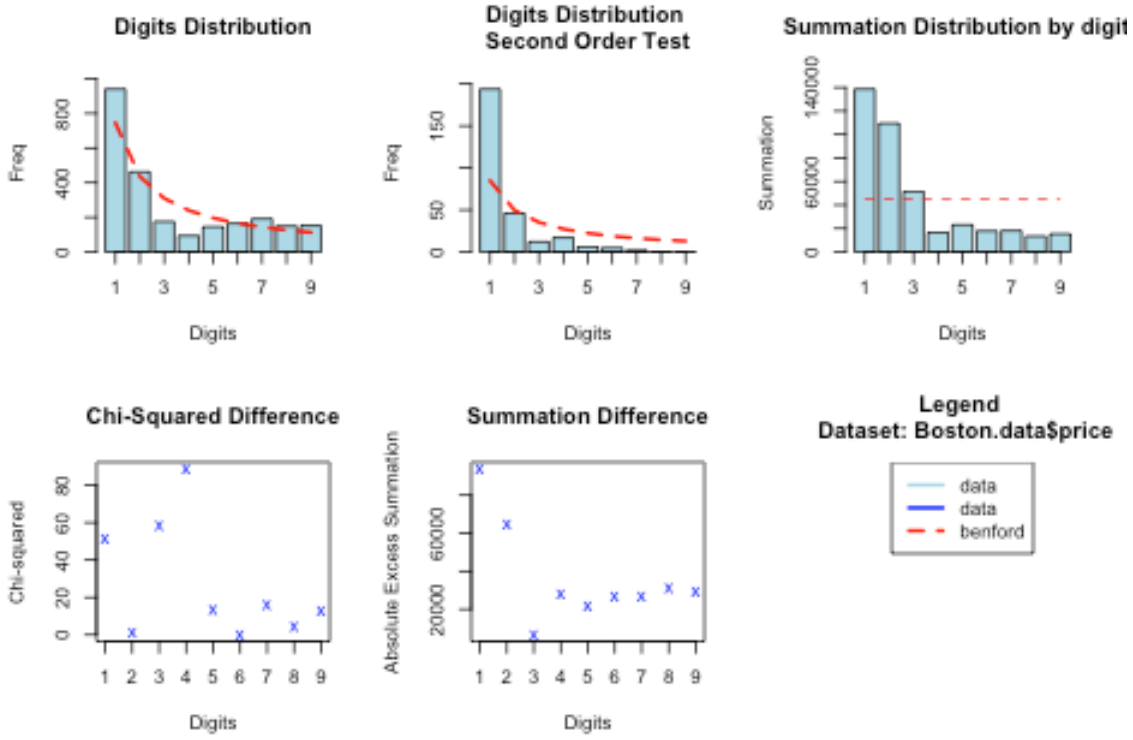
```
##
##  Mantissa Arc Test
##
## data:  Boston.data$price
## L2 = 0.039805, df = 2, p-value < 2.2e-16
```

```
#Gets the the statistics of the first ONE digits of a benford object

ben_price2 <- benford(Boston.data$price, number.of.digits=1)
ben_price2
```

```
##
## Benford object:
##
## Data: Boston.data$price
## Number of observations used = 2476
## Number of obs. for second order = 282
## First digits analysed = 1
##
## Mantissa:
##
##    Statistic  Value
##         Mean  0.474
##          Var  0.095
##  Ex.Kurtosis -1.340
##     Skewness  0.248
##
##
## The 5 largest deviations:
##
##   digits absolute.diff
## 1      1        195.65
## 2      4        145.95
## 3      3        134.35
## 4      5         51.05
## 5      7         48.41
##
## Stats:
##
```

```
##  Pearson's Chi-squared test
##
## data:  Boston.data$price
## X-squared = 247.17, df = 8, p-value < 2.2e-16
##
##
##  Mantissa Arc Test
##
## data:  Boston.data$price
## L2 = 0.039805, df = 2, p-value < 2.2e-16
##
## Mean Absolute Deviation: 0.02980708
## Distortion Factor: -1.957824
##
## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!
```

```
plot(ben_price2)
```



From the output for the one digits test, we can see that the lead digit of 3 and 4 occurred only about half of as often as is expected. I think it is becasue both cheaper ones and luxury ones are popular, the 300 to 400 dollars are kind of in the middle, the overal quality maybe not be as good as the expensive ones but the price is more expensive than the cheaper ones.

```
#Probability of a digit sequence
prob.dig <- c()
for (i in 1:20){
  prob.dig[i] <- p.these.digits(i)}
prob.dig <- as.vector(prob.dig)

#Probability of a digit appear on the n-th position
```

```r
p.this.digit.at.n(1,1) #probability of 1 appear on the first position
```

```
## [1] 0.30103
```

```r
#Find out suspect observations
suspects <- getSuspects(ben_price1, Boston.data)

#Creates a data frame with the first digits and the differences from Benford's Law in decreasing order
suspectsTable(ben_price1, by="absolute.diff")
```

```
##     digits absolute.diff
##  1:     15     72.600880
##  2:     75     60.757234
##  3:     99     46.192742
##  4:     17     45.536807
##  5:     20     43.535296
##  6:     70     41.747076
##  7:     65     37.582687
##  8:     25     36.825452
##  9:     60     35.225824
## 10:     80     34.641901
## 11:     19     33.843647
## 12:     50     33.705975
## 13:     22     32.200436
## 14:     12     30.929025
## 15:     95     29.740074
## 16:     26     27.582670
## 17:     23     24.764912
## 18:     90     24.117966
## 19:     41     23.912414
## 20:     33     23.101283
## 21:     43     22.720931
## 22:     85     22.423095
## 23:     34     21.170679
## 24:     31     21.139792
## 25:     16     20.809548
## 26:     44     20.165357
## 27:     32     20.089169
## 28:     36     19.462477
## 29:     24     18.896427
## 30:     51     17.880523
## 31:     47     17.639007
## 32:     46     17.125905
## 33:     89     16.985203
## 34:     48     16.172190
## 35:     54     15.731070
## 36:     66     15.170427
## 37:     53     15.099896
## 38:     57     14.701569
## 39:     37     14.676716
## 40:     64     14.671855
## 41:     11     14.564477
## 42:     61     14.485152
## 43:     63     12.934415
```

```
## 44:     74     12.433950
## 45:     42     12.302653
## 46:     38     11.931782
## 47:     73     11.630336
## 48:     69     11.527602
## 49:     52     11.482774
## 50:     86     11.431504
## 51:     56     11.032588
## 52:     97     11.028949
## 53:     91     10.752173
## 54:     92     10.625120
## 55:     58     10.381869
## 56:     94     10.379081
## 57:     87     10.289427
## 58:     39     10.224572
## 59:     27     10.106606
## 60:     71     10.039590
## 61:     82     10.034258
## 62:     67      9.930856
## 63:     68      9.698281
## 64:     40      9.552291
## 65:     29      9.545216
## 66:     93      9.500785
## 67:     81      9.194192
## 68:     21      9.023584
## 69:     72      8.832140
## 70:     14      8.811059
## 71:     62      8.205377
## 72:     77      7.875201
## 73:     79      7.473870
## 74:     96      7.143241
## 75:     59      5.927085
## 76:     28      5.734157
## 77:     78      5.698442
## 78:     55      5.624464
## 79:     76      5.056581
## 80:     83      4.878156
## 81:     49      4.724237
## 82:     30      3.740673
## 83:     84      3.725748
## 84:     35      3.707486
## 85:     13      2.689276
## 86:     98      1.916978
## 87:     45      1.634207
## 88:     10      1.511712
## 89:     88      1.150560
## 90:     18      1.139193
##     digits absolute.diff
```

```
#Chi-sqaure test
chisq(ben_price1)
```

```
##
##  Pearson's Chi-squared test
##
```

```
## data:  Boston.data$price
## X-squared = 1826.5, df = 89, p-value < 2.2e-16
```

```
# Based on the chisq output, the p-value is small so we reject null hypothesis, which means that the di
```

BenfordTests Package for Price

```
#JP Sqaure test
jpsq.benftest(x=Boston.data$price,digits = 2, pvalmethod = "simulate", pvalsims = 10000)
```

```
##
##  JP-Square Correlation Statistic Test for Benford Distribution
##
## data:  Boston.data$price
## J_stat_squ = 0.56771, p-value < 2.2e-16
```

```
#Joenssen's JP-square Test for Benford's Law
#The result signifys that the square correlation between signifd(data$Amount,2) and pbenf(2) is not zer
# Euclidean Distance Test for Benford's Law
edist.benftest(Boston.data$price)
```

```
##
##  Euclidean Distance Test for Benford Distribution
##
## data:  Boston.data$price
## d_star = 5.8694, p-value < 2.2e-16
```

```
# The p-value is smaller than 0.05 so that we reject the null hypothesis. Therefore, the goodness-of-fi

#rbenf
rbenf(10) #10 observations
```

```
##  [1] 4.652867 1.025490 1.283088 1.769185 5.047204 2.863717 5.811458
##  [8] 9.047630 2.622184 1.219592
```

Conclusion

In conclusion, we can say that the price and reviews in Boston Airbnb doesn't perfectly follow the Benford Law. However, it follows the trend in general. Accomodates and ratings have positive effect on the price, other variables have negative effect. Among all the variables, room type has the most significant effect on the price on Airbnb because it would cost a lot more when it comes to entire house/ apt than shared room or private room. Price, accomodates and neighborhood are the major factors that may influence the ratings on Airbnb. These all make sense since the neighborhood is related to crime rate in the area, properties in a nice place tend to have more high ratings. I think these are all useful points for Airbnb users or owners to know about.