MA684 Midterm Project: Airbnb Data Analysis

Shuyi Jiang

12/03/2017

Project overview and goal

As a beginner who is new to airbnb, I really have no idea how to find an ideal place to stay on this platform. I looked up the beginner guides online and figured out that what the majority of potential houseguests care the most are price, location and host. How people choose airbnb should be an interesting angle for marketing. The project will focus on how to find a nice airbnb at a good price, in other words, to figure out the relationship bewteen price and factors that houseguests care the most.

Boston: data cleaning

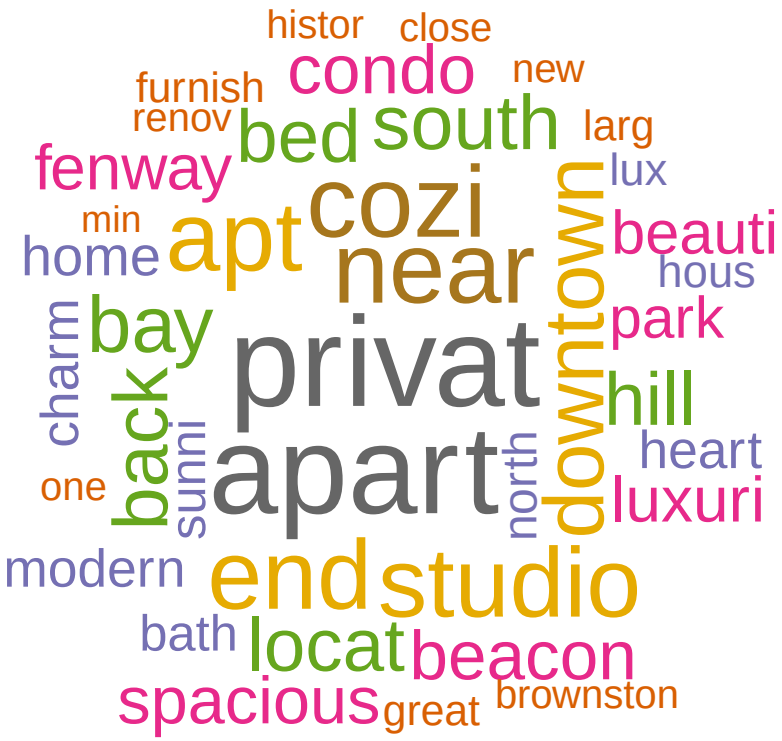
I choosed Boston, the city I am the most familiar with, for this project. The data includes listing information in Boston from January to November in 2016. The sample size is 36359 (1 rows was removed because of missing data).

```
boston1 <- read.csv("boston1.csv")
boston2 <- read.csv("boston2.csv")
boston3 <- read.csv("boston3.csv")
boston4 <- read.csv("boston4.csv")
boston5 <- read.csv("boston5.csv")
boston6 <- read.csv("boston6.csv")
boston7 <- read.csv("boston7.csv")
boston8 <- read.csv("boston8.csv")
boston9 <- read.csv("boston9.csv")
boston10 <- read.csv("boston10.csv")
boston11 <- read.csv("boston11.csv")

boston1$month <- 1
boston2$month <- 2
boston3$month <- 3
boston4$month <- 4
boston5$month <- 5
boston6$month <- 6
boston7$month <- 7
boston8$month <- 8
boston9$month <- 9
boston10$month <- 10
boston11$month <- 11

boston <- rbind(boston1,boston2,boston3,boston4,boston5,
                boston6,boston7,boston8,boston9,boston10,boston11)
boston <- boston[-c(4,14)]
boston <- filter(boston, !is.na(host_id))
```

Word cloud

Looking at the word clouds of top 40 words about listing in name (searching key words), summary and description, I figured out that houseguests pay the most attention of price, location and room type.

```
datatext<-read.csv("listings.csv",stringsAsFactors = FALSE)
jeopCorpus <- Corpus(VectorSource(datatext$name))
jeopCorpus <- tm_map(jeopCorpus, PlainTextDocument)
jeopCorpus <- tm_map(jeopCorpus, stripWhitespace)
jeopCorpus <- tm_map(jeopCorpus, tolower)
jeopCorpus <- tm_map(jeopCorpus, removeNumbers)
jeopCorpus <- tm_map(jeopCorpus, removePunctuation)
jeopCorpus <- tm_map(jeopCorpus, removeWords, stopwords('english'))
jeopCorpus <- tm_map(jeopCorpus, stemDocument)
jeopCorpus <- tm_map(jeopCorpus, removeWords, "bedroom")
jeopCorpus <- tm_map(jeopCorpus, removeWords, "room")
jeopCorpus <- tm_map(jeopCorpus, removeWords, "boston")
pal<-brewer.pal(10,"Dark2")
wordcloud(jeopCorpus, max.words = 40, random.order = FALSE,colors=pal)
```




```
jeopCorpus <- Corpus(VectorSource(datatext$summary))
jeopCorpus <- tm_map(jeopCorpus, PlainTextDocument)
jeopCorpus <- tm_map(jeopCorpus, stripWhitespace)
jeopCorpus <- tm_map(jeopCorpus, tolower)
jeopCorpus <- tm_map(jeopCorpus, removeNumbers)
jeopCorpus <- tm_map(jeopCorpus, removePunctuation)
jeopCorpus <- tm_map(jeopCorpus, removeWords, stopwords('english'))
jeopCorpus <- tm_map(jeopCorpus, stemDocument)
jeopCorpus <- tm_map(jeopCorpus, removeWords, "bedroom")
jeopCorpus <- tm_map(jeopCorpus, removeWords, "room")
jeopCorpus <- tm_map(jeopCorpus, removeWords, "boston")
```

```
pal<-brewer.pal(10,"Dark2")
wordcloud(jeopCorpus, max.words = 40, random.order = FALSE,colors=pal)
```


```
jeopCorpus <- Corpus(VectorSource(datatext$description))
jeopCorpus <- tm_map(jeopCorpus, PlainTextDocument)
jeopCorpus <- tm_map(jeopCorpus, stripWhitespace)
jeopCorpus <- tm_map(jeopCorpus, tolower)
jeopCorpus <- tm_map(jeopCorpus, removeNumbers)
jeopCorpus <- tm_map(jeopCorpus, removePunctuation)
jeopCorpus <- tm_map(jeopCorpus, removeWords, stopwords('english'))
jeopCorpus <- tm_map(jeopCorpus, stemDocument)
jeopCorpus <- tm_map(jeopCorpus, removeWords, "bedroom")
jeopCorpus <- tm_map(jeopCorpus, removeWords, "room")
jeopCorpus <- tm_map(jeopCorpus, removeWords, "boston")
pal<-brewer.pal(10,"Dark2")
wordcloud(jeopCorpus, max.words = 40, random.order = FALSE,colors=pal)
```

Airbnb daily price in Boston

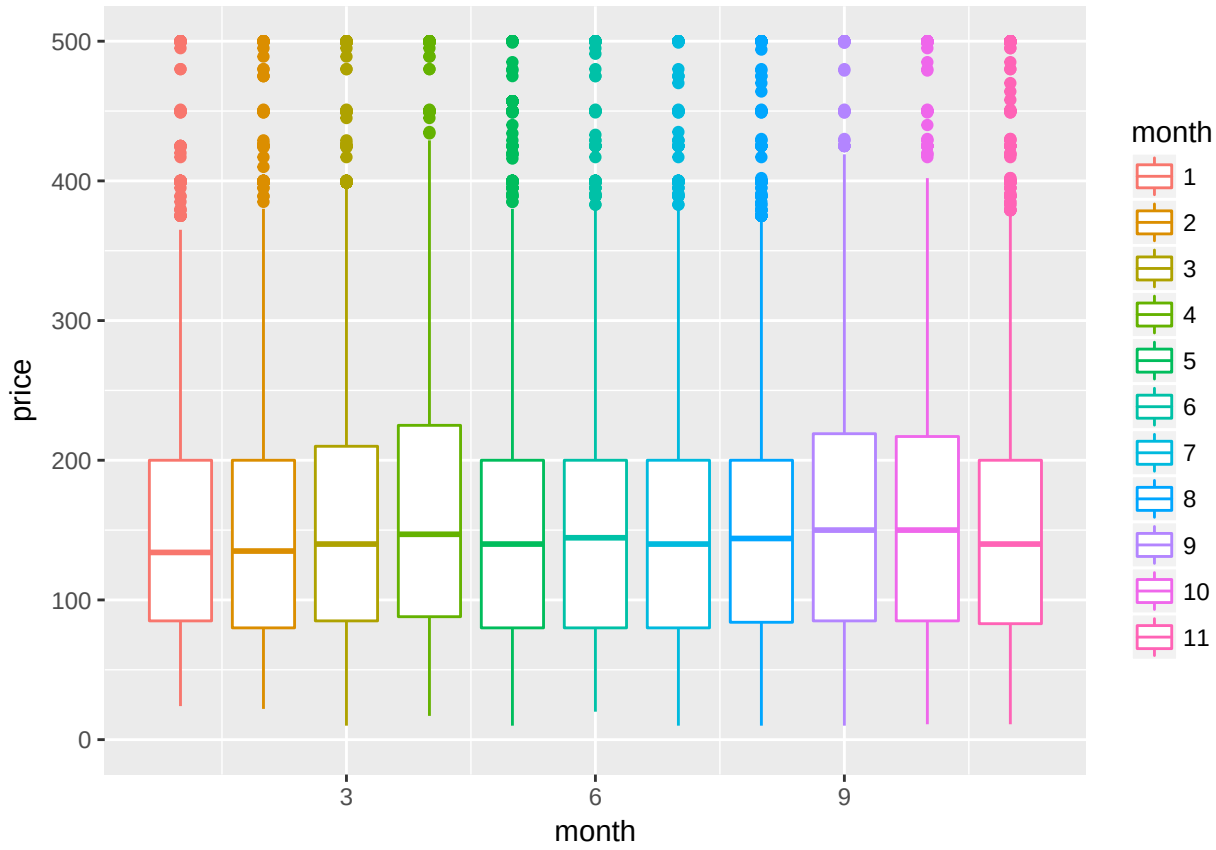
Based on what discussed about, I am interested in how location(neighborhood), room type and season affect price.

```
summary(boston$price)
```

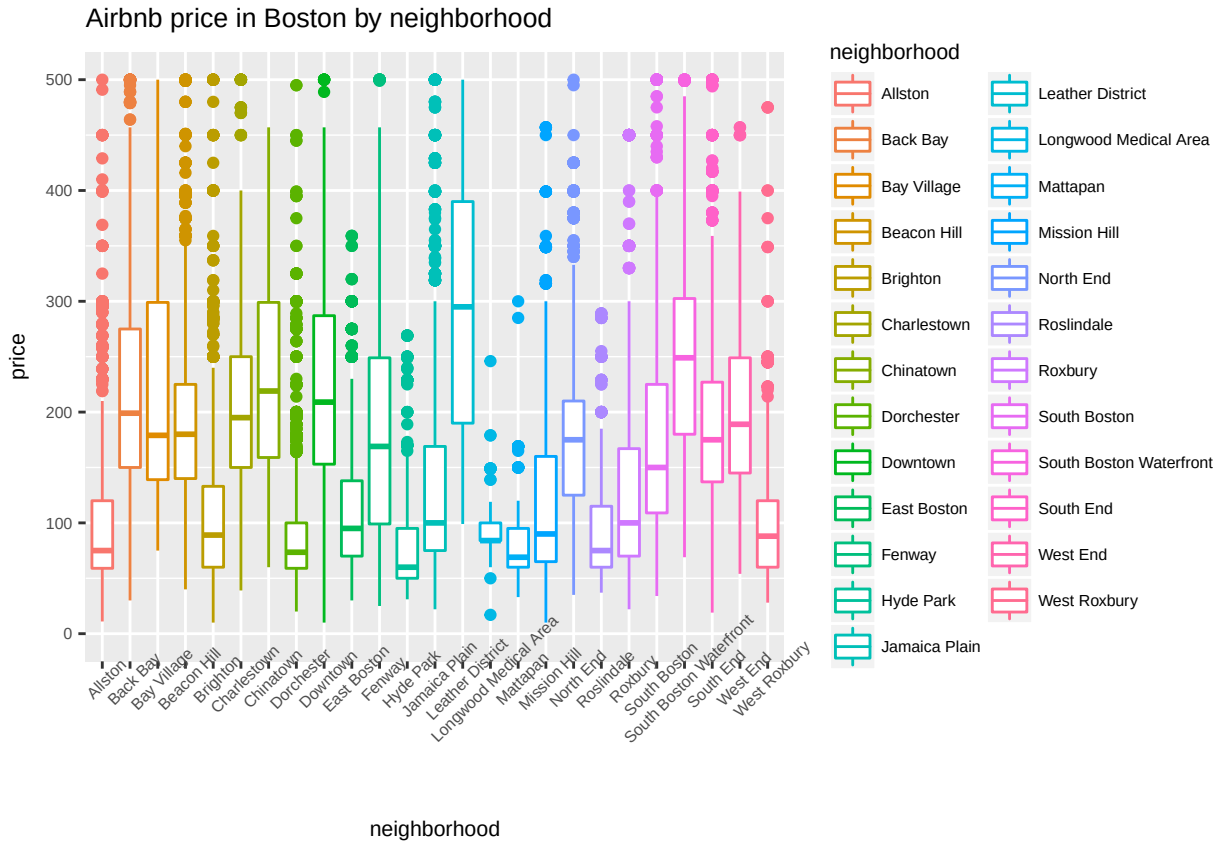
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    10.0    85.0   147.0   173.5   215.0 10000.0
```

Only 769 out of 36359 samples have price over 500. Thus the following plots will only show samples with price under 500. The boxplots indicates that season (month) has very limited influence on price, while neighborhood and room type make relatively great influence on price.

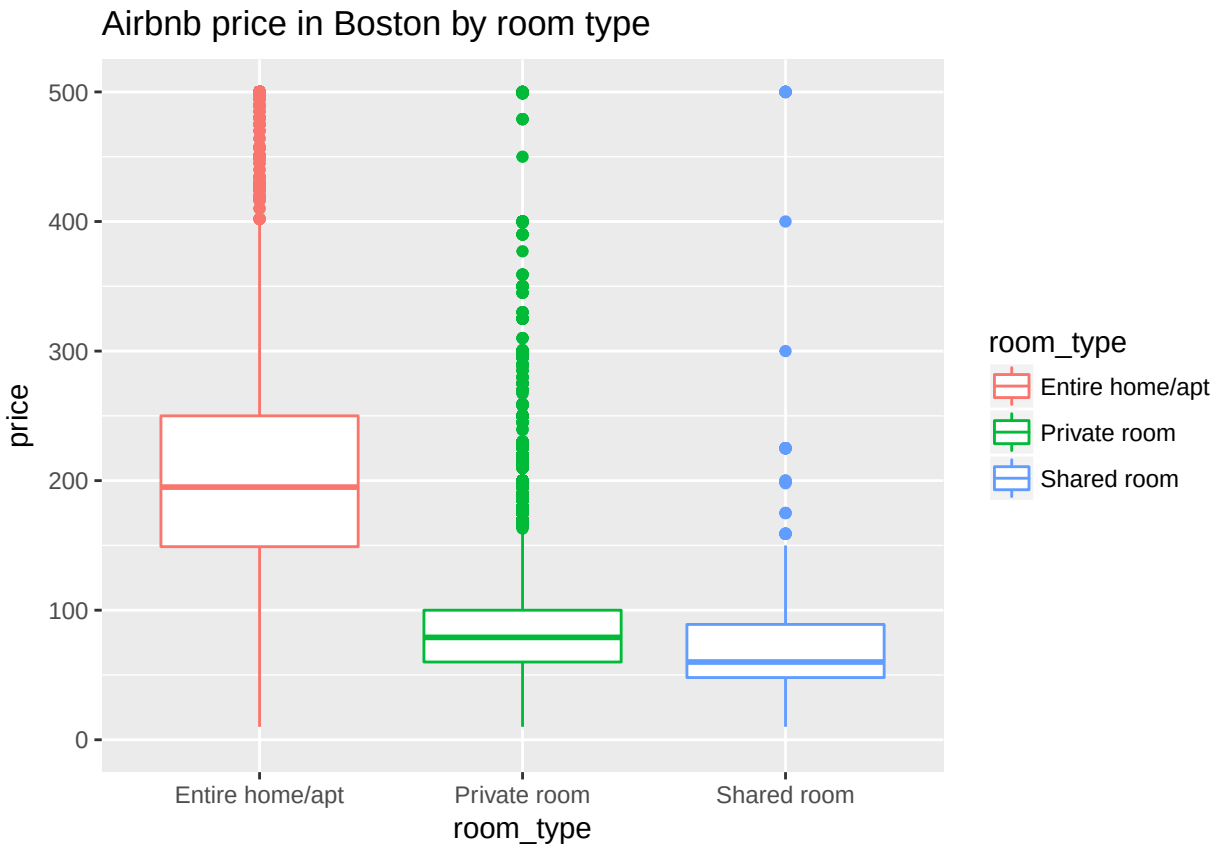
```
ggplot(boston, aes(x=month,y=price,group=month,colour=as.factor(month)))+
  geom_boxplot()+ylim(0,500)+labs(colour="month")
```




```
ggplot(boston, aes(x=neighborhood,y=price,group=neighborhood,colour=neighborhood))+
  geom_boxplot()+ylim(0,500)+theme(axis.text.x = element_text(angle = 45), text=element_text(size = 8))+
  labs(title="Airbnb price in Boston by neighborhood")
```




```
ggplot(boston, aes(x=room_type,y=price,group=room_type,colour=room_type))+
  geom_boxplot()+ylim(0,500)+labs(title="Airbnb price in Boston by room type")
```



Room type of airbnb in Boston

There are 3 room types: entire home/apt, private room and shared room. Entire home/apt is the most popular room type (more than half of 36359 listings) and the average price of this type of room is more than twice of the other 2. Share room is only about 2% of all the listing airbnb and it has the lowest average price.

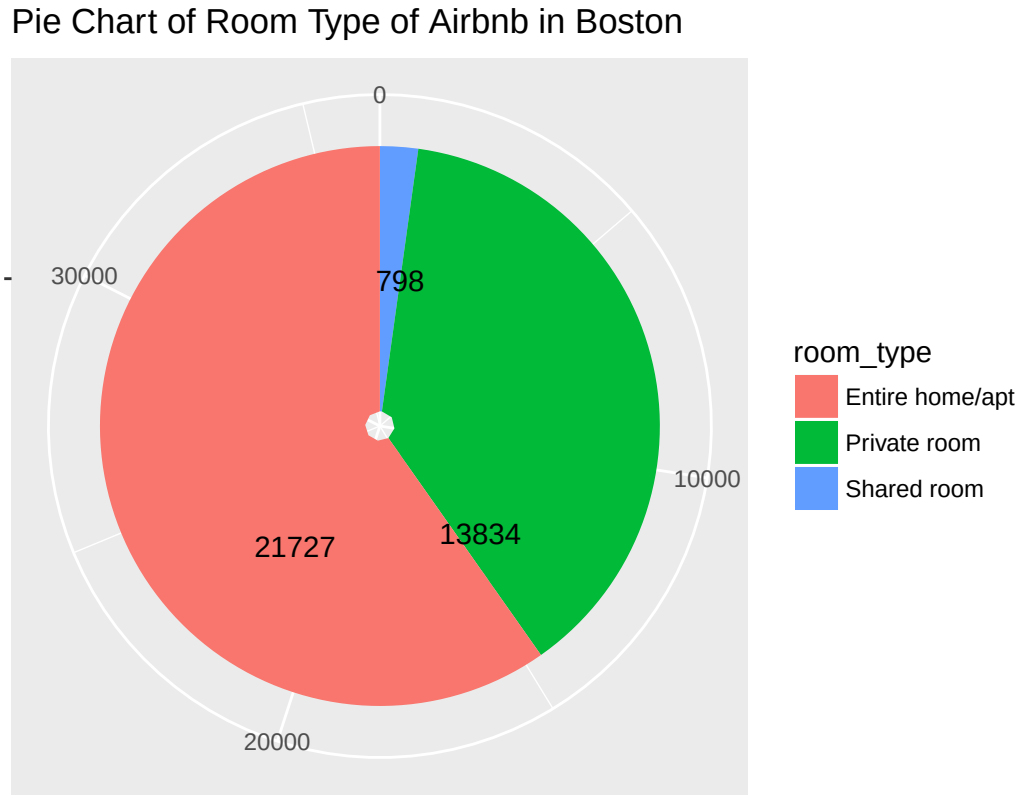
```
room_type <- count(boston, room_type)
room_type
```

```
## # A tibble: 3 x 2
##         room_type     n
##            <fctr> <int>
## 1 Entire home/apt 21727
## 2    Private room 13834
## 3     Shared room   798
```

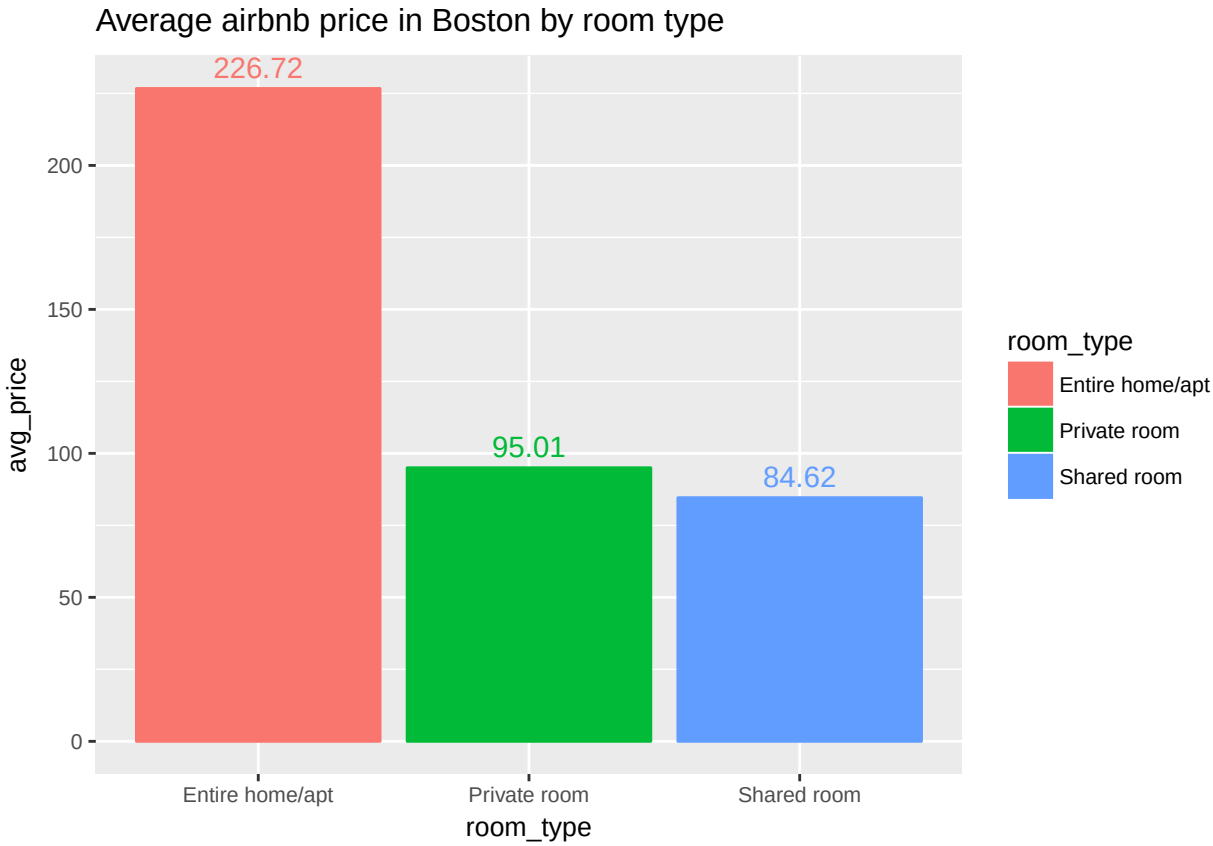
```
price_by_roomtype <- sqldf("SELECT room_type, avg(price) as avg_price
                           FROM boston
                           GROUP BY room_type")
price_by_roomtype
```

```
##         room_type avg_price
## 1 Entire home/apt 226.72154
## 2    Private room  95.00860
## 3     Shared room  84.61529
```

```
ggplot(room_type,aes(x="",y=n,fill=room_type))+geom_bar(stat = "identity")+
  coord_polar("y")+labs(x="",y="",title="Pie Chart of Room Type of Airbnb in Boston")+
  geom_text(aes(label=n))
```




```
ggplot(price_by_roomtype, aes(x=room_type,y=avg_price, fill=room_type,colour=room_type))+
  geom_bar(stat = "identity")+geom_text(aes(label=round(avg_price, digits = 2)),vjust=-0.5)+
  theme(text=element_text(size = 10))+labs(title="Average airbnb price in Boston by room type")
```



Neighborhood of airbnb in Boston

There are 25 different neighborhoods with airbnb listings in boston. The triangle area on the south riverbank of Charles River, including Allston, Fenway, Back Bay and South Boston is the most popular place for airbnb. This area does not have the lowest price but has relatively more convenient transportation.

```
neighborhood <- count(boston, neighborhood)
neighborhood
```

```
## # A tibble: 25 x 2
##    neighborhood     n
##          <fctr> <int>
##  1      Allston  2495
##  2     Back Bay  3035
##  3  Bay Village   222
##  4  Beacon Hill  2212
##  5     Brighton  2032
##  6  Charlestown   840
##  7    Chinatown   674
##  8   Dorchester  2670
##  9     Downtown  1869
## 10  East Boston  1482
## # ... with 15 more rows
```

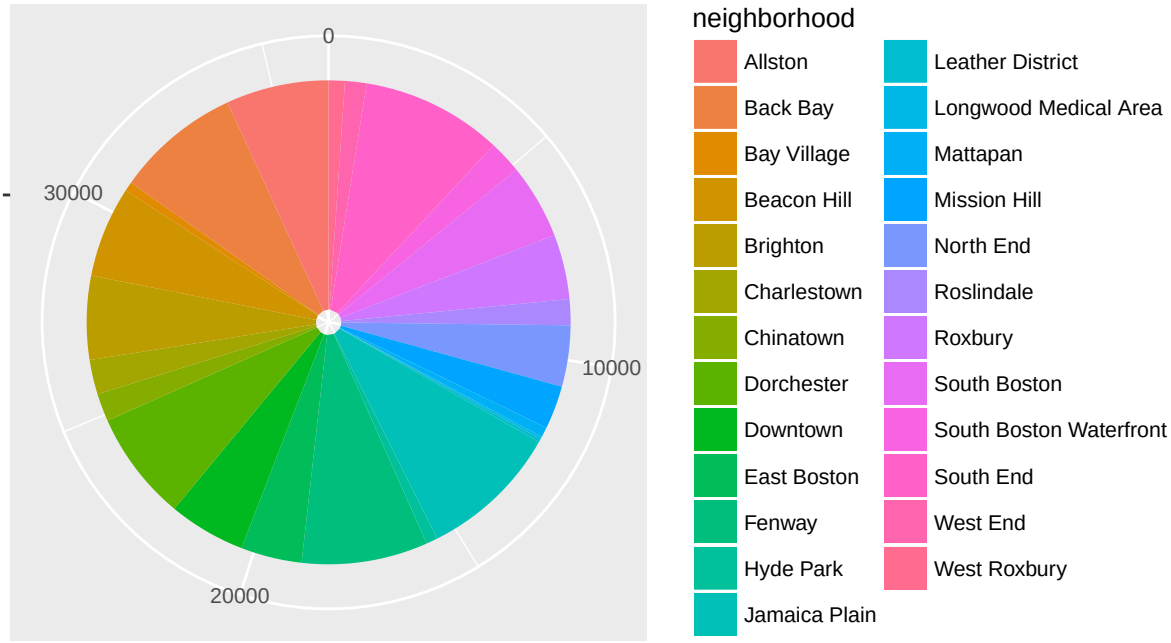
```
price_by_neighborhood <- sqldf("SELECT neighborhood, avg(price) as avg_price
                           FROM boston
```

```
                          GROUP BY neighborhood")
price_by_neighborhood
```

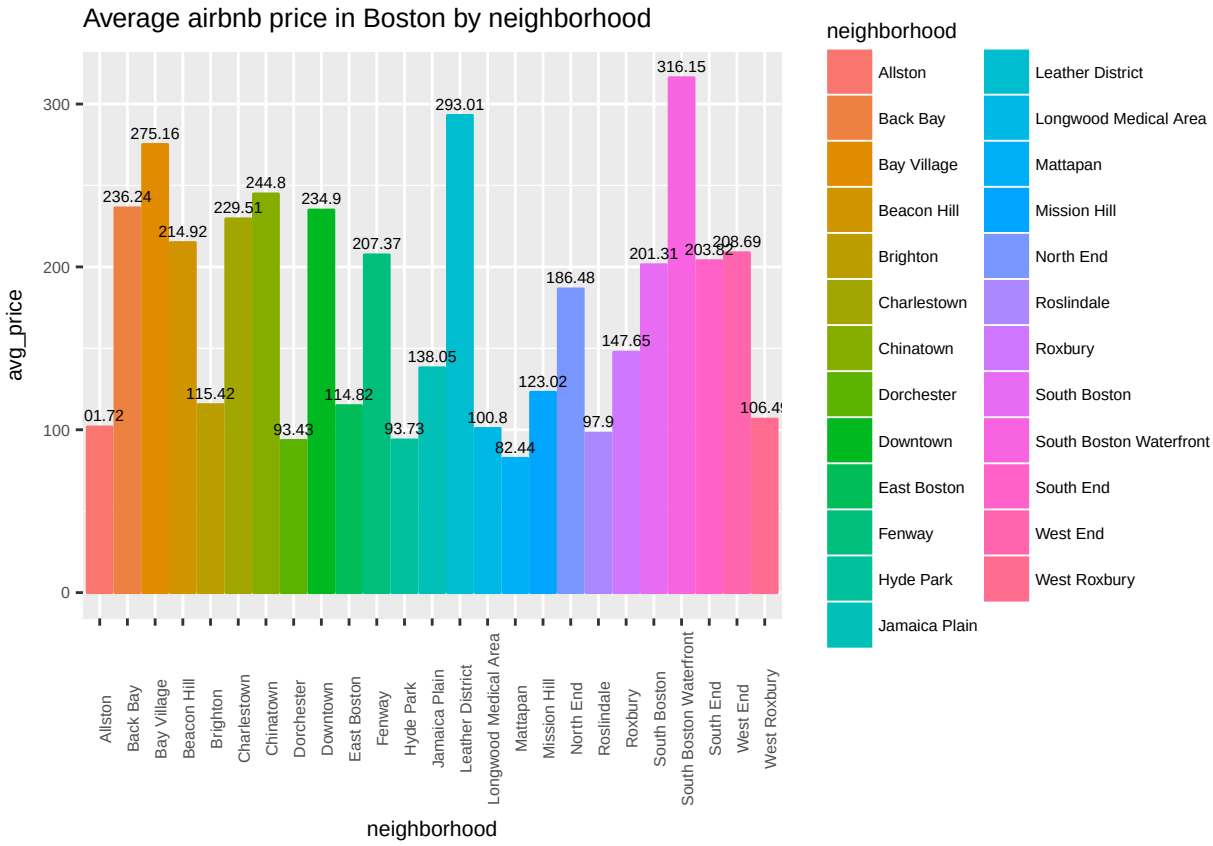
```
##                neighborhood avg_price
## 1                   Allston 101.71543
## 2                 Back Bay 236.24217
## 3               Bay Village 275.16216
## 4               Beacon Hill 214.91953
## 5                  Brighton 115.42470
## 6               Charlestown 229.51071
## 7                 Chinatown 244.79970
## 8                Dorchester  93.43408
## 9                  Downtown 234.89781
## 10              East Boston 114.81984
## 11                   Fenway 207.37304
## 12                Hyde Park  93.73118
## 13            Jamaica Plain 138.05409
## 14        Leather District 293.01235
## 15   Longwood Medical Area 100.80000
## 16                  Mattapan  82.44444
## 17             Mission Hill 123.02095
## 18                 North End 186.47613
## 19               Roslindale  97.89683
## 20                  Roxbury 147.64508
## 21             South Boston 201.30706
## 22 South Boston Waterfront 316.14713
## 23                 South End 203.82161
## 24                  West End 208.68893
## 25             West Roxbury 106.49490
```

```
ggplot(neighborhood,aes(x="",y=n,fill=neighborhood))+geom_bar(stat = "identity")+
  coord_polar("y")+labs(x="",y="",title="Pie Chart of Neighborhood of Airbnb in Boston")+
  theme(text=element_text(size = 10))
```

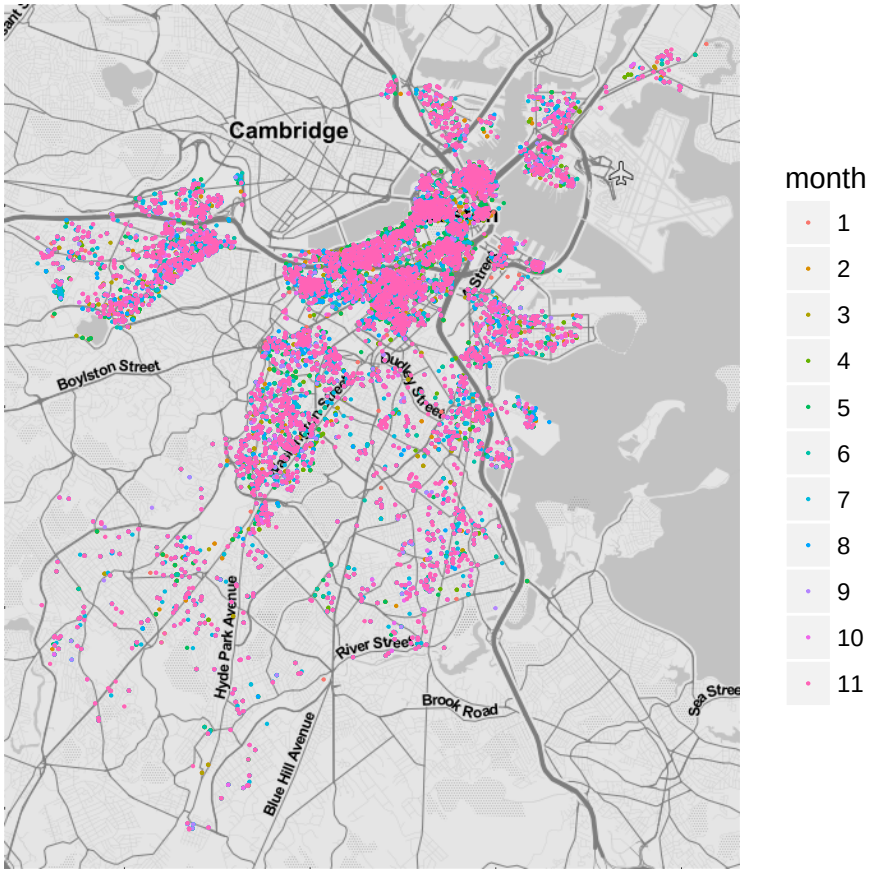
Pie Chart of Neighborhood of Airbnb in Boston




```
ggplot(price_by_neighborhood, aes(x=neighborhood,y=avg_price, fill=neighborhood,
                                  colour=neighborhood))+
  geom_bar(stat = "identity")+theme(axis.text.x = element_text(angle = 90),
                                    text=element_text(size = 8))+
  geom_text(aes(label=round(avg_price, digits = 2)),vjust=-0.5,size=2,colour="black")+
  labs(title="Average airbnb price in Boston by neighborhood")
```




```r
mapboston <- (data.frame(
  x = boston$latitude,
  y = boston$longitude,
  month = boston$month
))
qmplot(y, x, data = mapboston, colour = as.factor(month), size = I(0.1), darken = .1)+
  labs(colour="month")
```



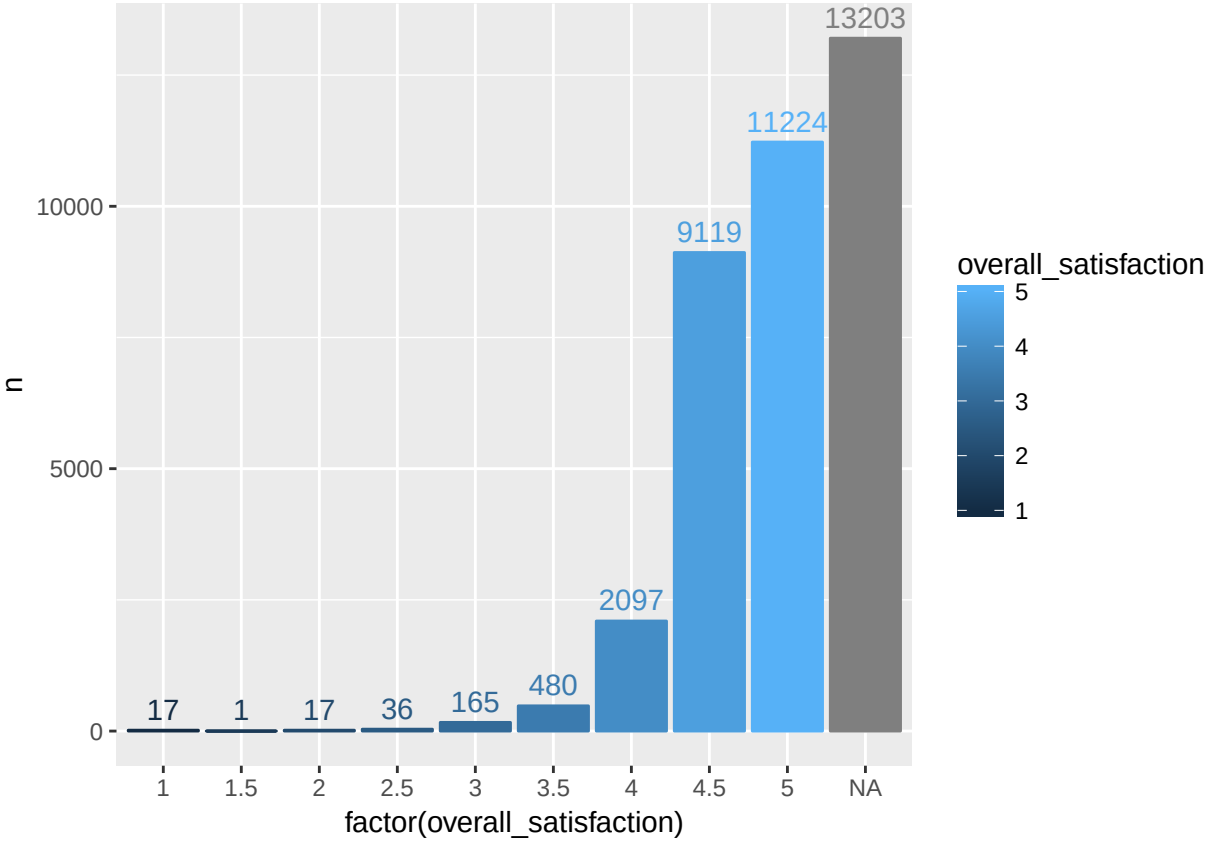
Overall satisfication of airbnb in Boston

The listed airbnb in this data set is rated in a scale of 1 to 5 with 0.5 interval. The majority (9119 + 11224 = 20343) of 36359 samples is rated as 4.5 or 5 and 13203 samples, which is about 36%, are missing rating data. Thus, the overall satisfaction will be disregarded in the analysis because it is obviously not a representitive indicator.

```
satisfaction <- count(boston, overall_satisfaction)
satisfaction
```

```
## # A tibble: 10 x 2
##    overall_satisfaction     n
##                   <dbl> <int>
##  1                  1.0    17
##  2                  1.5     1
##  3                  2.0    17
##  4                  2.5    36
##  5                  3.0   165
##  6                  3.5   480
##  7                  4.0  2097
##  8                  4.5  9119
##  9                  5.0 11224
## 10                   NA 13203
```

```
ggplot(satisfaction, aes(x=factor(overall_satisfaction),y=n,
                         fill=overall_satisfaction,colour=overall_satisfaction))+
  geom_bar(stat = "identity")+geom_text(aes(label=n),vjust=-0.5)
```



Model Testing

Multilevel linear model was choosen after several trials of different models. While fitting the multilevel, samples have NA in reviews, accommodates and bedrooms are removed.

```
boston.model <- filter(boston, !is.na(accommodates) & !is.na(bedrooms) & !is.na(room_type))
```

Model fit1, fit2, fit3 and fit4 are varing-intercept models with predictors. They are using different combination of 3 variables (reviews, accommodates and bedrooms) to see whether it is better to include any of the 3 variables in the model. It turns out that fit1 with all 3 variables is the best among these 4 models.

```
fit1 <- lmer(price ~ reviews + accommodates + bedrooms +
               (1|neighborhood) + (1|room_type), REML = FALSE, data=boston.model)
summary(fit1)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: price ~ reviews + accommodates + bedrooms + (1 | neighborhood) +
##     (1 | room_type)
##    Data: boston.model
##
##       AIC       BIC    logLik  deviance  df.resid
##  396470.1  396528.7 -198228.1  396456.1     31640
```

```
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.603 -0.347 -0.075  0.177 77.313
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  neighborhood (Intercept)  2606     51.05
##  room_type    (Intercept)  1133     33.66
##  Residual                 16080    126.81
## Number of obs: 31647, groups:  neighborhood, 25; room_type, 3
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  57.39900   22.08655    2.60
## reviews      -0.19555    0.02185   -8.95
## accommodates 10.65165    0.71465   14.90
## bedrooms     52.04004    1.47783   35.21
##
## Correlation of Fixed Effects:
##             (Intr) reviws accmmd
## reviews     -0.017
## accommodats -0.024 -0.083
## bedrooms    -0.024  0.094 -0.693
```

```
fit2 <- lmer(price ~ reviews + bedrooms +
               (1|neighborhood) + (1|room_type), REML = FALSE, data=boston.model)
summary(fit2)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: price ~ reviews + bedrooms + (1 | neighborhood) + (1 | room_type)
##    Data: boston.model
##
##       AIC       BIC    logLik  deviance  df.resid
##  396689.3  396739.4 -198338.6  396677.3     31641
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.428 -0.343 -0.075  0.177 77.028
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  neighborhood (Intercept)  2595     50.94
##  room_type    (Intercept)  1650     40.62
##  Residual                 16192    127.25
## Number of obs: 31647, groups:  neighborhood, 25; room_type, 3
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 65.37684   25.67660    2.55
## reviews     -0.16848    0.02185   -7.71
## bedrooms    67.29390    1.06935   62.93
##
## Correlation of Fixed Effects:
##          (Intr) reviws
```

```
## reviews  -0.016
## bedrooms -0.049  0.050
```

```
fit3 <- lmer(price ~ reviews + accommodates +
               (1|neighborhood) + (1|room_type), REML = FALSE, data=boston.model)
summary(fit3)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## price ~ reviews + accommodates + (1 | neighborhood) + (1 | room_type)
##    Data: boston.model
##
##       AIC       BIC    logLik  deviance  df.resid
##  397684.1  397734.3 -198836.0  397672.1     31641
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.129 -0.366 -0.087  0.184 75.729
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  neighborhood (Intercept)  2431.1   49.31
##  room_type    (Intercept)   751.7   27.42
##  Residual                  16711.5  129.27
## Number of obs: 31647, groups:  neighborhood, 25; room_type, 3
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  76.27360   18.80414    4.06
## reviews      -0.26782    0.02218  -12.08
## accommodates 28.09614    0.52523   53.49
##
## Correlation of Fixed Effects:
##             (Intr) reviws
## reviews     -0.018
## accommodats -0.069 -0.025
```

```
fit4 <- lmer(price ~ accommodates + bedrooms +
               (1|neighborhood) + (1|room_type), REML = FALSE, data=boston.model)
summary(fit4)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## price ~ accommodates + bedrooms + (1 | neighborhood) + (1 | room_type)
##    Data: boston.model
##
##       AIC       BIC    logLik  deviance  df.resid
##  396548.1  396598.3 -198268.1  396536.1     31641
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.586 -0.344 -0.073  0.168 77.241
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
```

```
##  neighborhood (Intercept)  2674     51.71
##  room_type    (Intercept)  1137     33.72
##  Residual                 16120    126.96
## Number of obs: 31647, groups:  neighborhood, 25; room_type, 3
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   54.0585    22.1761    2.44
## accommodates  10.1194     0.7131   14.19
## bedrooms      53.2819     1.4732   36.17
##
## Correlation of Fixed Effects:
##             (Intr) accmmd
## accommodats -0.026
## bedrooms    -0.023 -0.690
```

```
anova(fit1,fit2,fit3,fit4)
```

```
## Data: boston.model
## Models:
## fit2: price ~ reviews + bedrooms + (1 | neighborhood) + (1 | room_type)
## fit3: price ~ reviews + accommodates + (1 | neighborhood) + (1 | room_type)
## fit4: price ~ accommodates + bedrooms + (1 | neighborhood) + (1 | room_type)
## fit1: price ~ reviews + accommodates + bedrooms + (1 | neighborhood) +
## fit1:     (1 | room_type)
##      Df    AIC    BIC  logLik deviance    Chisq Chi Df Pr(>Chisq)
## fit2  6 396689 396739 -198339   396677
## fit3  6 397684 397734 -198836   397672    0.000      0          1
## fit4  6 396548 396598 -198268   396536 1135.979      0     <2e-16 ***
## fit1  7 396470 396529 -198228   396456   79.986      1     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model

The final model choosen is a multilevel model varying both intercepts and slopes.

$$y_i \sim \ N(\alpha_{j[i]} + \beta_{j[i]} x_i,\ \sigma_y^2),\ \ for\ i = 1, \ldots, n$$

```
fit5 <- lmer(price ~ reviews + accommodates + bedrooms +
               (1+reviews + accommodates + bedrooms|neighborhood) +
               (1+reviews + accommodates + bedrooms|room_type),
             REML = FALSE, data=boston.model)
summary(fit5)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## price ~ reviews + accommodates + bedrooms + (1 + reviews + accommodates +
##     bedrooms | neighborhood) + (1 + reviews + accommodates +
##     bedrooms | room_type)
##    Data: boston.model
##
##       AIC       BIC    logLik  deviance  df.resid
##  395770.1  395979.2 -197860.1  395720.1     31622
```

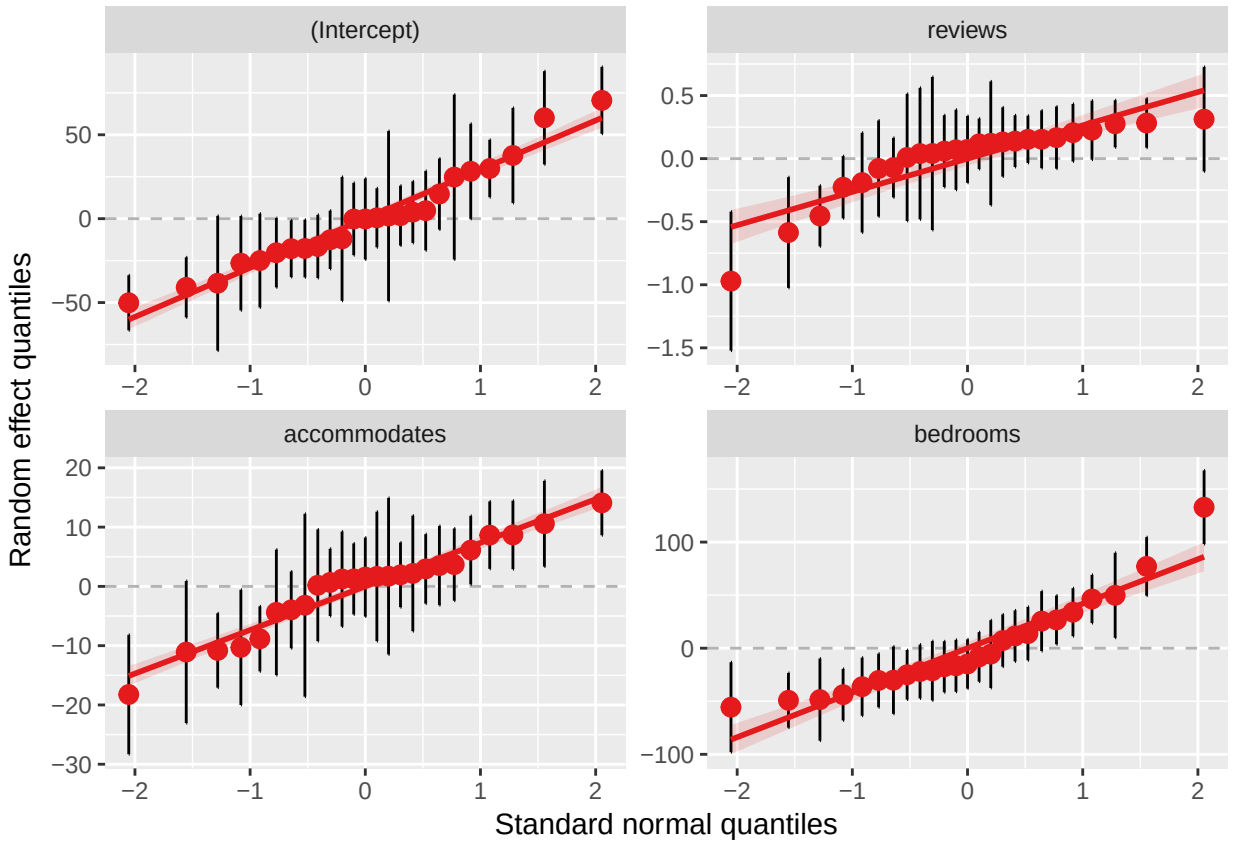
```
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.842 -0.323 -0.069  0.173 78.423
##
## Random effects:
##  Groups       Name         Variance  Std.Dev. Corr
##  neighborhood (Intercept)  1.039e+03  32.2264
##               reviews      1.111e-01   0.3334 -0.18
##               accommodates 7.434e+01   8.6221 -0.60  0.37
##               bedrooms     2.923e+03  54.0604  0.30 -0.28 -0.43
##  room_type    (Intercept)  9.019e+03  94.9677
##               reviews      9.308e+03  96.4788 -0.62
##               accommodates 1.342e+04 115.8441 -0.25  0.34
##               bedrooms     8.913e+03  94.4093 -0.12  0.23 -0.14
##  Residual                  1.559e+04 124.8679
## Number of obs: 31647, groups:  neighborhood, 25; room_type, 3
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   97.1534    72.0641   1.348
## reviews       -0.3488    55.7022  -0.006
## accommodates   4.2225    66.9410   0.063
## bedrooms      25.5634    72.3429   0.353
##
## Correlation of Fixed Effects:
##             (Intr) reviws accmmd
## reviews     -0.471
## accommodats -0.194  0.344
## bedrooms    -0.475  0.175 -0.112
## convergence code: 1
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 5 negative eigenvalues
```

As shown below, the coefficients of fitted model are quite significant, especially the number of rooms. Reviews (the number of total reviews of a specific room, which is used to predict the popularity of the this room) has much less influence than the other 2 variables. In general, with everything else remains the same, 1 increase in accommodates will lead to 4.2 higher daily price. One more room in airbnb will increase the daily price by 25.6 on average.

```
kable(as.data.frame(fixef(fit5)))
```

	fixef(fit5)
(Intercept)	97.1534111
reviews	-0.3487666
accommodates	4.2224770
bedrooms	25.5634352

```
sjp.lmer(fit5, type = "re.qq")
```

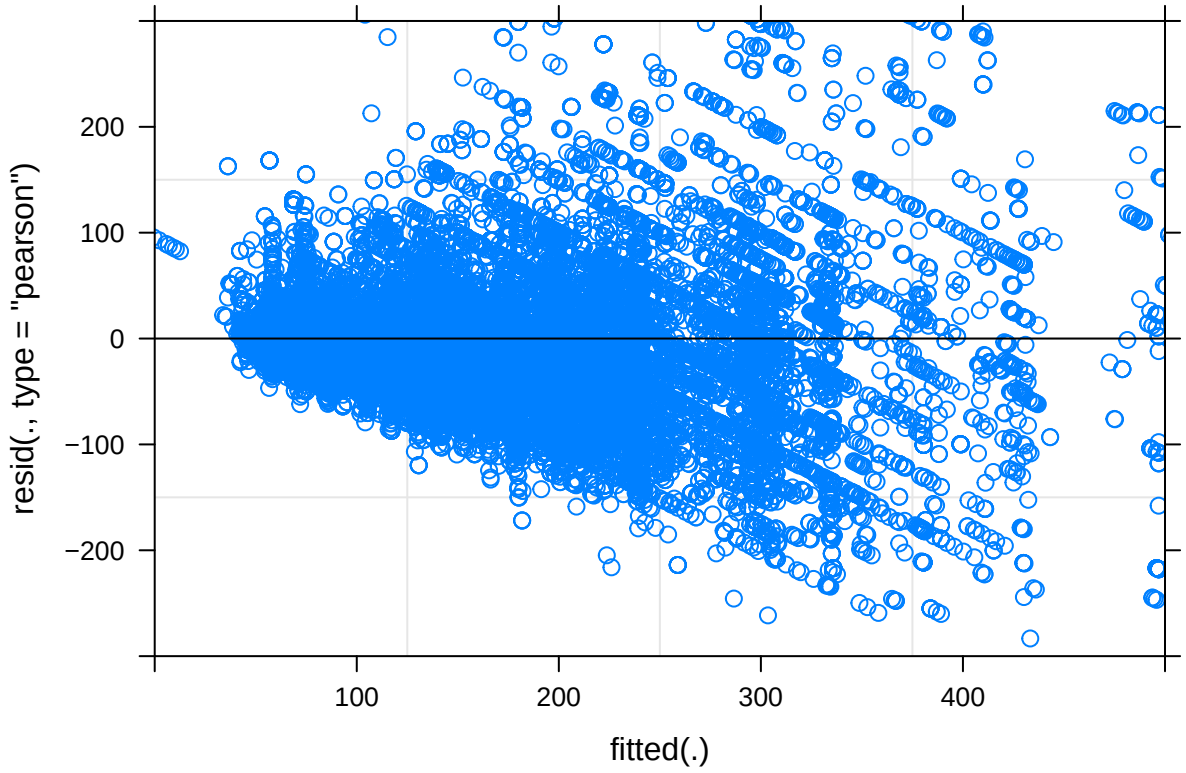



```
sjp.lmer(fit5, y.offset = .4)
```

The model has 2 multilevel predictors: neighborhood and room type. As shown in the graphs above, the

influence of variables varies by different levels of predictors. For example, bedrooms increases the price more with the same change in neighborhood with higher price, such as Back bay.

```
plot(fit5, ylim = c(-300,300), xlim = c(0,500))
```



Discussion and concerns

The original data has lots of limitation, such as missing guest_id, which is necessary to do analysis of houseguests' review of rooms they have stayed in. Although the final model is better than the other testing models, it has huge AIC. Digging deeper with help of data other than the original data is definitely what I need to do after this first step of research. More variables that will make the model better has to be determined.