Final Project

Alina

11/27/2020

My chosen data represents New York Airbnb data for 2019. This data was collected from Kaggle website. My goal in this project is to create a model predicting price for Airbnb. In order to do so, I will create a few models and perform variable selection. Also I will try to perform analysis to see where are the most expensive listings and what is their average price.
First of all, let's take a look at my data

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#library(knitr)
mydata <- read_csv("AB_NYC_2019.csv")
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   name = col_character(),
##   host_id = col_double(),
##   host_name = col_character(),
##   neighbourhood_group = col_character(),
##   neighbourhood = col_character(),
##   latitude = col_double(),
##   longitude = col_double(),
##   room_type = col_character(),
##   price = col_double(),
##   minimum_nights = col_double(),
##   number_of_reviews = col_double(),
##   last_review = col_date(format = ""),
##   reviews_per_month = col_double(),
##   calculated_host_listings_count = col_double(),
##   availability_365 = col_double()
## )
```

```r
#summary(mydata)
#head(mydata)
glimpse(mydata)
```

```
## Rows: 48,895
## Columns: 16
## $ id                              <dbl> 2539, 2595, 3647, 3831, 5022, 5099, ...
## $ name                            <chr> "Clean & quiet apt home by the park"...
## $ host_id                         <dbl> 2787, 2845, 4632, 4869, 7192, 7322, ...
## $ host_name                       <chr> "John", "Jennifer", "Elisabeth", "Li...
## $ neighbourhood_group             <chr> "Brooklyn", "Manhattan", "Manhattan"...
## $ neighbourhood                   <chr> "Kensington", "Midtown", "Harlem", "...
## $ latitude                        <dbl> 40.64749, 40.75362, 40.80902, 40.685...
## $ longitude                       <dbl> -73.97237, -73.98377, -73.94190, -73...
## $ room_type                       <chr> "Private room", "Entire home/apt", "...
## $ price                           <dbl> 149, 225, 150, 89, 80, 200, 60, 79, ...
## $ minimum_nights                  <dbl> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1, 5, 2...
## $ number_of_reviews               <dbl> 9, 45, 0, 270, 9, 74, 49, 430, 118, ...
## $ last_review                     <date> 2018-10-19, 2019-05-21, NA, 2019-07...
## $ reviews_per_month               <dbl> 0.21, 0.38, NA, 4.64, 0.10, 0.59, 0....
## $ calculated_host_listings_count <dbl> 6, 2, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, ...
## $ availability_365                <dbl> 365, 355, 365, 194, 0, 129, 0, 220, ...
```

There are 48,895 observations and 16 variables, such as name of the host, neighborhood, room type and reviews per month. I noticed that there are some missing values. Also price vary from 0 to 10.000, which seems unrealistic to me and will need to keep in mind it.

```r
summary(is.na(mydata))
```

```
##      id              name           host_id         host_name
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:48895     FALSE:48879     FALSE:48895     FALSE:48874
##                  TRUE :16                        TRUE :21
##  neighbourhood_group neighbourhood    latitude       longitude
##  Mode :logical       Mode :logical   Mode :logical   Mode :logical
##  FALSE:48895         FALSE:48895     FALSE:48895     FALSE:48895
##
##  room_type          price          minimum_nights  number_of_reviews
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:48895     FALSE:48895     FALSE:48895     FALSE:48895
##
##  last_review     reviews_per_month calculated_host_listings_count
##  Mode :logical   Mode :logical     Mode :logical
##  FALSE:38843     FALSE:38843       FALSE:48895
##  TRUE :10052     TRUE :10052
##  availability_365
##  Mode :logical
##  FALSE:48895
##
```

Precisely, there are 10052 NA values in reviews_per_month and last_review. Some people don't leave reviews, so I think it doesn't affect price and I will need to omit these missing values.

```
mydata$reviews_per_monthNA<-mydata$reviews_per_month
mydata$reviews_per_monthNA[mydata$reviews_per_month==0]<-NA
mydata$last_reviewNA<-mydata$last_review
mydata$last_reviewNA[mydata$last_review==0]<-NA
#summary(mydata$last_reviewNA)
summary(mydata$reviews_per_monthNA)
```

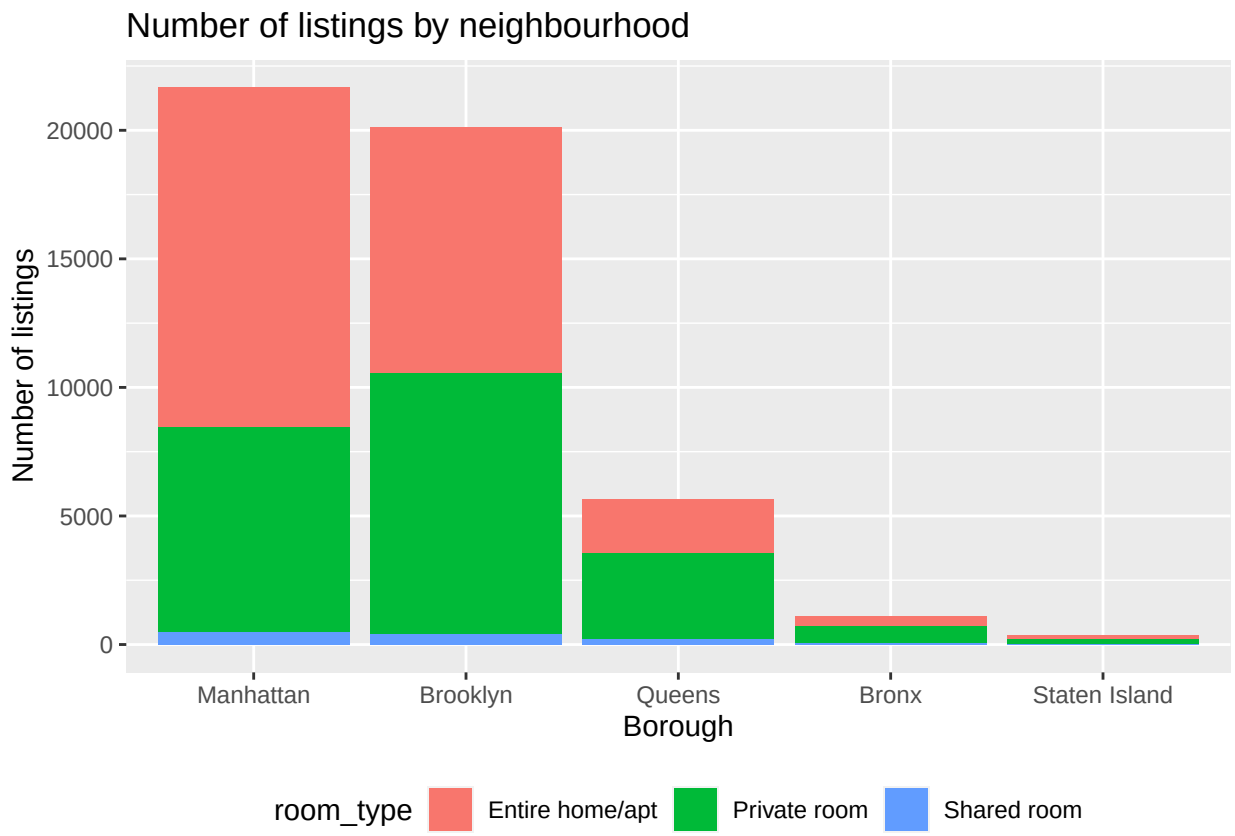
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.010   0.190   0.720   1.373   2.020  58.500   10052
```

So I now dealing with extreme values of price, I decided to remove 10% of the lowest and highest values in the price column.

```
filtered_mydata <- mydata %>%
  filter(price < quantile(mydata$price, 0.9) & price > quantile(mydata$price, 0.1))%>%drop_na()
```

Now after cleaning my data we see some trends.
I did analysis to find out the type of listing that are common to a particular neighborhood.

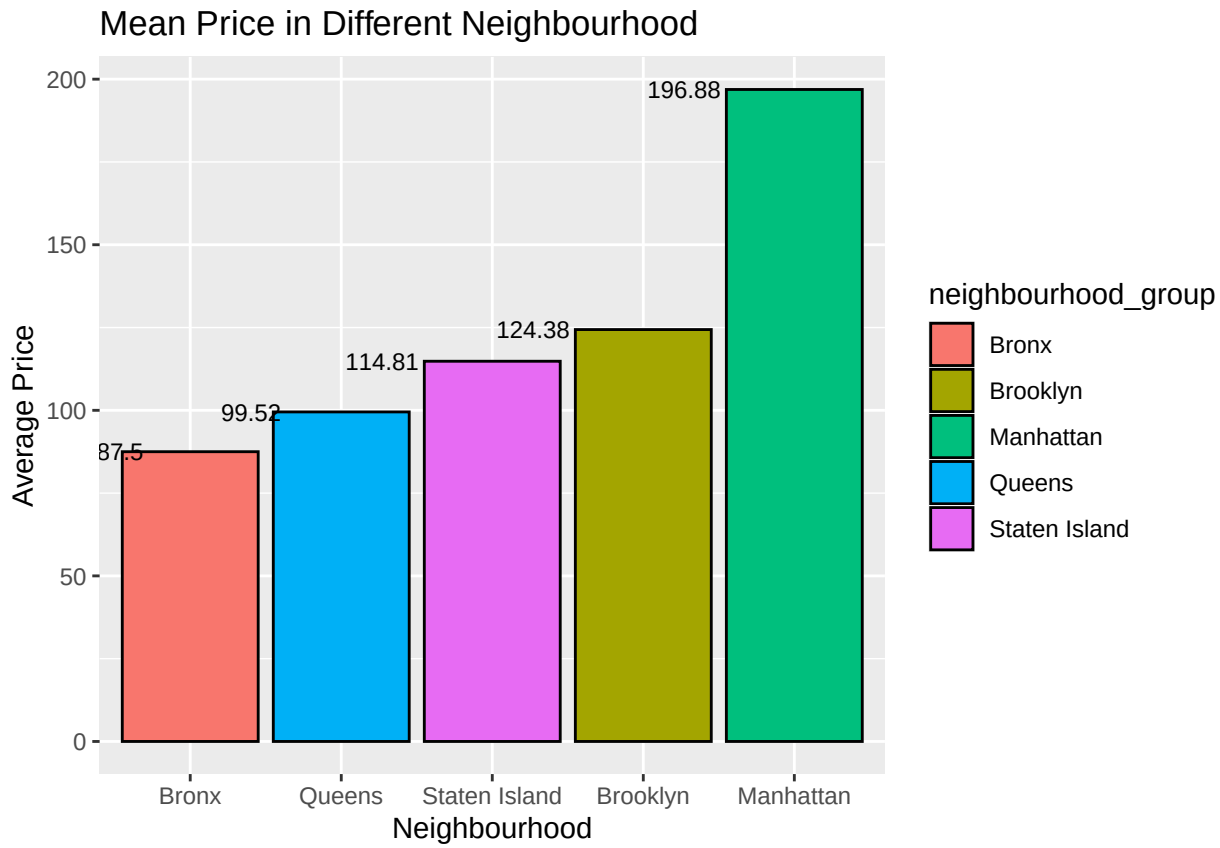
```
ggplot(mydata, aes(x = fct_infreq(neighbourhood_group), fill = room_type)) +
    geom_bar() +
    labs(title="Number of listings by neighbourhood",
         x="Borough",y="Number of listings") +
    theme(legend.position = "bottom")
```



We see that:
1. Manhattan has the most number of listings and entire home/apt dominates there.
2. In Brooklyn it's almost evenly split entire home/apt and private room.
3. Shared room is the least common in all neighborhoods.

```
mydata%>%
group_by(neighbourhood_group)%>%
  summarise(mean_price = mean(price))%>%
  ggplot(aes(x=reorder(neighbourhood_group,mean_price),y=mean_price, fill=neighbourhood_group))+geom_col
geom_text(aes(label = round(mean_price,digit = 2)), hjust = 2.0, color = "black", size = 3.0)+
  xlab("Neighbourhood")+ylab("Average Price")+
  ggtitle("Mean Price in Different Neighbourhood")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



We see that the most expensive listing is in Manhattan 196.88 USD, followed by Brooklyn 124.38 and Bronx has the cheapest listing with an average price of 87.50 USD.

```
mydata%>%
group_by(room_type)%>%
  summarise(mean_price = mean(price))%>%
  ggplot(aes(x=reorder(room_type,mean_price),y=mean_price, fill=room_type))+geom_col(color = "black") +
geom_text(aes(label = round(mean_price,digit = 2)), hjust = 2.0, color = "black", size = 4.0)+
  xlab("Room type")+ylab("Average Price")+
  ggtitle("Mean Price of Different Types of Listings")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

Not surprisingly an average price is the highest for Entire home/apt - 211.79 USD.

Now, I will try to build the linear model predicting price. I will omit such predictors as host name, ID and last review as I think it's not relevant to predicting price.
Thus, predictors that I will be using are: neighbourhood_group; latitude; longitude; room_type; minimum_nights; number_of_reviews; reviews_per_month; calculated_host_listings_count; availability_365.

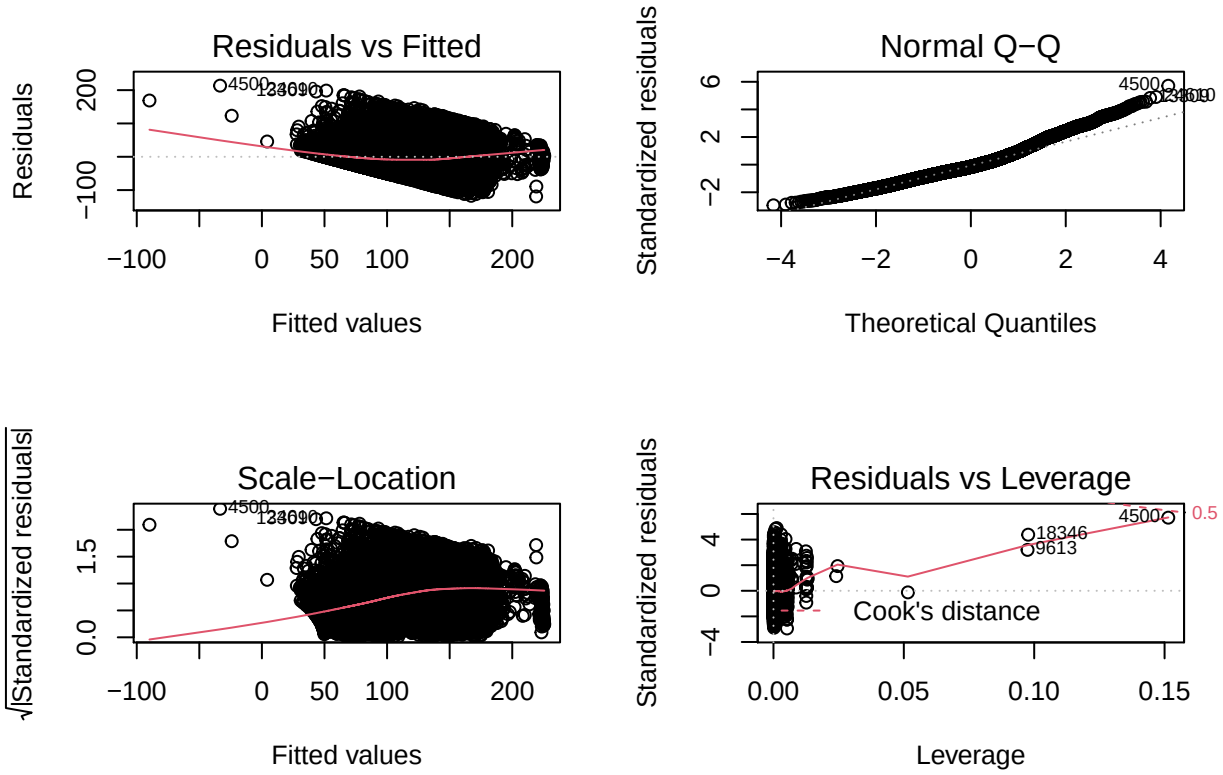
```
airbnb_model_1<- lm (price ~ neighbourhood_group + latitude + longitude + room_type + minimum_nights  +
                       availability_365, data = filtered_mydata)

summary(airbnb_model_1)
```

```
##
## Call:
## lm(formula = price ~ neighbourhood_group + latitude + longitude +
##     room_type + minimum_nights + number_of_reviews + reviews_per_month +
##     calculated_host_listings_count + availability_365, data = filtered_mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.084  -26.146   -6.409   21.037  213.405
##
## Coefficients:
##                                   Estimate Std. Error  t value Pr(>|t|)
```

```
## (Intercept)                     -1.527e+04  7.141e+02  -21.384  < 2e-16 ***
## neighbourhood_groupBrooklyn      -6.348e+00  2.064e+00   -3.075   0.0021 **
## neighbourhood_groupManhattan      1.683e+01  1.896e+00    8.875  < 2e-16 ***
## neighbourhood_groupQueens         4.753e+00  1.997e+00    2.380   0.0173 *
## neighbourhood_groupStaten Island -7.143e+01  3.806e+00  -18.767  < 2e-16 ***
## latitude                         -7.010e+01  6.933e+00  -10.111  < 2e-16 ***
## longitude                        -2.470e+02  8.026e+00  -30.774  < 2e-16 ***
## room_typePrivate room            -6.136e+01  4.706e-01 -130.379  < 2e-16 ***
## room_typeShared room             -7.426e+01  2.157e+00  -34.435  < 2e-16 ***
## minimum_nights                   -1.735e-01  1.277e-02  -13.585  < 2e-16 ***
## number_of_reviews                -2.249e-02  5.676e-03   -3.962 7.46e-05 ***
## reviews_per_month                -1.278e-01  1.676e-01   -0.763   0.4457
## calculated_host_listings_count    1.152e-01  9.424e-03   12.225  < 2e-16 ***
## availability_365                  4.127e-02  1.926e-03   21.425  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.56 on 31524 degrees of freedom
## Multiple R-squared:  0.4457, Adjusted R-squared:  0.4455
## F-statistic:  1950 on 13 and 31524 DF,  p-value: < 2.2e-16
```

Here we see that Adjusted R-squared(percentage of variance explained) is 45% which indicates that this model fits pretty well already.
Now I will use graphical diagnostics:

```
par(mfrow=c(2,2))
plot(airbnb_model_1)
```



The fist plot Residuals vs Fitted helps us to detect lack of fit. As smoother curve as more constant variance. Our plot shows some curvilinear trend suggesting some adjustments to the model. From QQ plot we can see if normality assumption is met. We see that two tails of points diverging from linearity suggesting a long-tailed error. Therefore, the model needs some changes.
I will use AIC to select needed variables.

```
airbnb_model_2<-step(airbnb_model_1)
```

```
## Start:  AIC=233574.8
## price ~ neighbourhood_group + latitude + longitude + room_type +
##     minimum_nights + number_of_reviews + reviews_per_month +
##     calculated_host_listings_count + availability_365
##
##                                   Df Sum of Sq      RSS    AIC
## - reviews_per_month                1       957 51868222 233573
## <none>                                         51867265 233575
## - number_of_reviews                1     25824 51893089 233588
## - latitude                         1    168194 52035460 233675
## - calculated_host_listings_count   1    245904 52113169 233722
## - minimum_nights                   1    303653 52170918 233757
## - availability_365                 1    755223 52622489 234029
## - longitude                        1   1558215 53425480 234506
## - neighbourhood_group              4   2658522 54525788 235143
## - room_type                        2  28719482 80586747 247468
##
## Step:  AIC=233573.4
```

```
## price ~ neighbourhood_group + latitude + longitude + room_type +
##     minimum_nights + number_of_reviews + calculated_host_listings_count +
##     availability_365
##
##                                 Df Sum of Sq      RSS    AIC
## <none>                                       51868222 233573
## - number_of_reviews              1     43826 51912049 233598
## - latitude                       1    167260 52035482 233673
## - calculated_host_listings_count  1    245301 52113523 233720
## - minimum_nights                 1    303144 52171366 233755
## - availability_365               1    755056 52623278 234027
## - longitude                      1   1585030 53453253 234521
## - neighbourhood_group            4   2657974 54526197 235141
## - room_type                      2  28758445 80626667 247481
```

Thus removing reviews_per_month would give us the lowest AIC of 233573. Removing any other predictors will only increase AIC.

```
anova(airbnb_model_1,airbnb_model_2,test='Chi')
```

```
## Analysis of Variance Table
##
## Model 1: price ~ neighbourhood_group + latitude + longitude + room_type +
##     minimum_nights + number_of_reviews + reviews_per_month +
##     calculated_host_listings_count + availability_365
## Model 2: price ~ neighbourhood_group + latitude + longitude + room_type +
##     minimum_nights + number_of_reviews + calculated_host_listings_count +
##     availability_365
##   Res.Df      RSS Df Sum of Sq Pr(>Chi)
## 1  31524 51867265
## 2  31525 51868222 -1   -957.03   0.4457
```

Finally, I will calculate test error using predict() function to see how my model performs on future data:

```
pi <- predict(object = airbnb_model_2, newdata = filtered_mydata)
# Mean Squared Error (MSE)
mean((pi - filtered_mydata$price)^2)
```

```
## [1] 1644.626
```

```
# Mean Absolute Error (MAE)
mean(abs(pi - filtered_mydata$price))
```

```
## [1] 31.13614
```

Therefore, selected model airbnb_model_2 predicts the variable with a good fit.
Doing this model I encounter some issues. For me was difficult to clean and prepare the data for analysis. I had to omit missing values and some extreme values in the price. Also I concluded that name of the host and id is not significant to my model so I can also exclude it.
It was interesting to see average price of listings. I think as time pass by and new data would be available for 2020 it would be exciting to see trends during pandemic.