DSC20 - Final Project

Sathish Manthani

11/12/2019

## Data import and cleansing

### Source dataset

[New York City Airbnb Open Data](https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data)

### Packages

Import the required packages/libraries to anlayze the dataset

library(ggplot2) # plots for visualization  
library(dplyr) # for data manipulation  
library(corrplot) # for display of correlation matrix  
library(GGally) # for gallary display of charts  
library(gridExtra) # to view plots in grid  
library(purrr) # for map\_dbl function

### Read the data file

airbnb.df <- read.csv("AB\_NYC\_2019.csv", stringsAsFactors = FALSE)

### Understanding the data

# view its class  
class(airbnb.df)

## [1] "data.frame"

# dimensions of the data  
dim(airbnb.df)

## [1] 48895 16

# Structure of the dataset  
str(airbnb.df)

## 'data.frame': 48895 obs. of 16 variables:  
## $ id : int 2539 2595 3647 3831 5022 5099 5121 5178 5203 5238 ...  
## $ name : chr "Clean & quiet apt home by the park" "Skylit Midtown Castle" "THE VILLAGE OF HARLEM....NEW YORK !" "Cozy Entire Floor of Brownstone" ...  
## $ host\_id : int 2787 2845 4632 4869 7192 7322 7356 8967 7490 7549 ...  
## $ host\_name : chr "John" "Jennifer" "Elisabeth" "LisaRoxanne" ...  
## $ neighbourhood\_group : chr "Brooklyn" "Manhattan" "Manhattan" "Brooklyn" ...  
## $ neighbourhood : chr "Kensington" "Midtown" "Harlem" "Clinton Hill" ...  
## $ latitude : num 40.6 40.8 40.8 40.7 40.8 ...  
## $ longitude : num -74 -74 -73.9 -74 -73.9 ...  
## $ room\_type : chr "Private room" "Entire home/apt" "Private room" "Entire home/apt" ...  
## $ 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 : chr "2018-10-19" "2019-05-21" "" "2019-07-05" ...  
## $ 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 ...

There are no special characters or spaces in column names, renaming is not required.  
However, some of the categorical variables can be converted to factors. Before conversion, let’s check how many unique values are there for each of these variables in the dataset.

### Unique values of variables

#following function gets unique values for each variable in the dataset   
sapply(airbnb.df, function(x) length(unique(x)))

## id name   
## 48895 47906   
## host\_id host\_name   
## 37457 11453   
## neighbourhood\_group neighbourhood   
## 5 221   
## latitude longitude   
## 19048 14718   
## room\_type price   
## 3 674   
## minimum\_nights number\_of\_reviews   
## 109 394   
## last\_review reviews\_per\_month   
## 1765 938   
## calculated\_host\_listings\_count availability\_365   
## 47 366

From the above output, I can see neighbourhood\_group, room\_type, neighbourhood variables can be factors

### Data type conversion

# convert neighbourhood\_group, room\_type, neighbourhood to factors  
airbnb.df['neighbourhood\_group'] <- as.factor(airbnb.df$neighbourhood\_group)  
airbnb.df['room\_type'] <- as.factor(airbnb.df$room\_type)  
airbnb.df['neighbourhood'] <- as.factor(airbnb.df$neighbourhood)  
  
#Convert last\_review to date datatype  
airbnb.df['last\_review'] <- as.Date(airbnb.df$'last\_review')  
  
#Look at the new structure  
str(airbnb.df)

## 'data.frame': 48895 obs. of 16 variables:  
## $ id : int 2539 2595 3647 3831 5022 5099 5121 5178 5203 5238 ...  
## $ name : chr "Clean & quiet apt home by the park" "Skylit Midtown Castle" "THE VILLAGE OF HARLEM....NEW YORK !" "Cozy Entire Floor of Brownstone" ...  
## $ host\_id : int 2787 2845 4632 4869 7192 7322 7356 8967 7490 7549 ...  
## $ host\_name : chr "John" "Jennifer" "Elisabeth" "LisaRoxanne" ...  
## $ 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 : Date, format: "2018-10-19" "2019-05-21" ...  
## $ 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 ...

### Looking at the raw data

head(airbnb.df)

## 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  
## 6 5099 Large Cozy 1 BR Apartment In Midtown East 7322  
## 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  
## 6 Chris Manhattan Murray Hill 40.74767 -73.97500  
## room\_type price minimum\_nights number\_of\_reviews last\_review  
## 1 Private room 149 1 9 2018-10-19  
## 2 Entire home/apt 225 1 45 2019-05-21  
## 3 Private room 150 3 0 <NA>  
## 4 Entire home/apt 89 1 270 2019-07-05  
## 5 Entire home/apt 80 10 9 2018-11-19  
## 6 Entire home/apt 200 3 74 2019-06-22  
## 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  
## 6 0.59 1 129

### Drop unwanted variables

As we can see there are a couple of id columns namely id, host\_id which dont carry much information as far as our analysis is concerned. So I’m removing them from the dataset.

drops <- c("id","host\_id")  
airbnb.df <- airbnb.df[ , !(names(airbnb.df) %in% drops)]

Check if there are any **duplicate rows** from the dataset. Duplicate rows can affect the data analysis.

### Duplicate check

#Number of observations in the file  
rows <- nrow(airbnb.df)  
  
#Number of unique rows   
unique\_rows <- nrow(unique(airbnb.df))  
  
rows == unique\_rows

## [1] TRUE

Since the total rows is equal to unique rows, we can see that there are no duplicates in the dataset.

### Missing values

Check for missing values in the dataset. Missing values can lead to unwanted results, so its good to cleanse such rows.

sapply(airbnb.df, # apply to each column of the dataframe   
 function(x) # this function:  
 sum(is.na(x)) # count the NAs  
) / nrow(airbnb.df) \* 100 # then divide the result by the number of rows in the airbnb dataframe

## name host\_name   
## 0.00000 0.00000   
## neighbourhood\_group neighbourhood   
## 0.00000 0.00000   
## latitude longitude   
## 0.00000 0.00000   
## room\_type price   
## 0.00000 0.00000   
## minimum\_nights number\_of\_reviews   
## 0.00000 0.00000   
## last\_review reviews\_per\_month   
## 20.55834 20.55834   
## calculated\_host\_listings\_count availability\_365   
## 0.00000 0.00000

#summary(airbnb.df)  
airbnb.df[is.na(airbnb.df$price)]

## data frame with 0 columns and 48895 rows

From the above output, we can see reviews\_per\_month variable has 20% NA values.  
I’ll omit the missing value rows if I find reviews\_per\_month has significant effect on the price of the hotel.  
For now, I think there’s no need to manually treat the data.

## Data Exploration

### Understand categorical variables

#Neighbourhood Groups  
levels(airbnb.df$neighbourhood\_group)

## [1] "Bronx" "Brooklyn" "Manhattan" "Queens"   
## [5] "Staten Island"

#Available room types  
levels(airbnb.df$room\_type)

## [1] "Entire home/apt" "Private room" "Shared room"

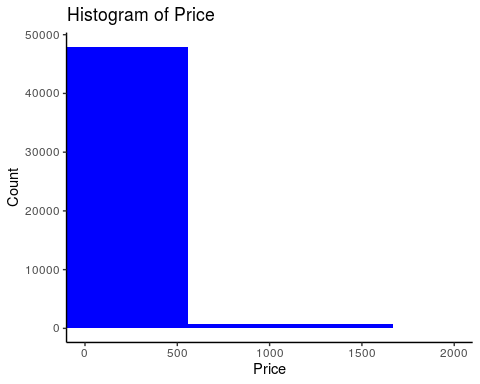
#Glimpse of Airbnb dataset  
glimpse(airbnb.df)

## Observations: 48,895  
## Variables: 14  
## $ name <chr> "Clean & quiet apt home by the pa…  
## $ host\_name <chr> "John", "Jennifer", "Elisabeth", …  
## $ neighbourhood\_group <fct> Brooklyn, Manhattan, Manhattan, B…  
## $ neighbourhood <fct> Kensington, Midtown, Harlem, Clin…  
## $ latitude <dbl> 40.64749, 40.75362, 40.80902, 40.…  
## $ longitude <dbl> -73.97237, -73.98377, -73.94190, …  
## $ room\_type <fct> Private room, Entire home/apt, Pr…  
## $ price <int> 149, 225, 150, 89, 80, 200, 60, 7…  
## $ minimum\_nights <int> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1, 5…  
## $ number\_of\_reviews <int> 9, 45, 0, 270, 9, 74, 49, 430, 11…  
## $ last\_review <date> 2018-10-19, 2019-05-21, NA, 2019…  
## $ reviews\_per\_month <dbl> 0.21, 0.38, NA, 4.64, 0.10, 0.59,…  
## $ calculated\_host\_listings\_count <int> 6, 2, 1, 1, 1, 1, 1, 1, 1, 4, 1, …  
## $ availability\_365 <int> 365, 355, 365, 194, 0, 129, 0, 22…

### Distributions

Let’s take a look at the distribution of our target variable “Price” -

#Distribution of Price variable  
ggplot(airbnb.df, aes(x=price)) + geom\_histogram(bins = 10, fill = 'blue') + coord\_cartesian(xlim=c(0,2000)) + theme\_classic() + labs(title = 'Histogram of Price', x = 'Price', y = 'Count')



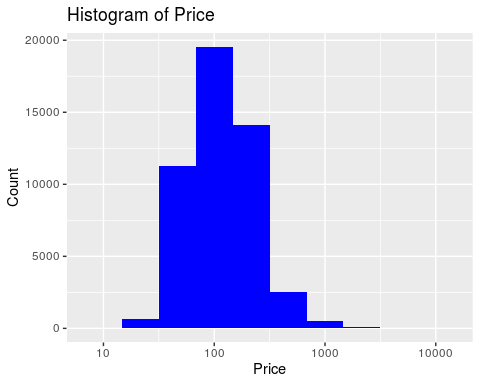
It doesn’t look like a Normal distribution. It is heavily skewed to the right.

### Histogram of Price with LOG transformation

#Distribution of Price variable  
ggplot(airbnb.df, aes(x=price) ) + geom\_histogram(bins = 10, fill = 'blue') + scale\_x\_log10() + labs(title = 'Histogram of Price', x = 'Price', y = 'Count')

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Removed 11 rows containing non-finite values (stat\_bin).

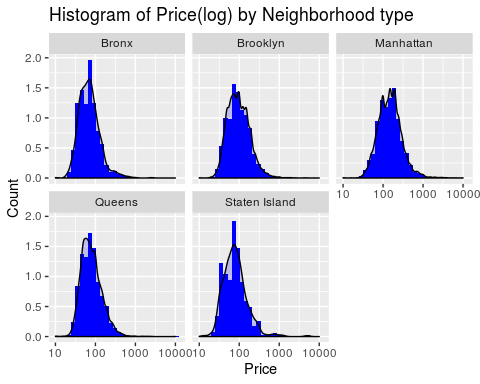


ggplot(airbnb.df, aes(x=price) ) + geom\_histogram(bins = 30, aes(y = ..density..), fill = "blue") + geom\_density(alpha = 0.2, fill = "blue") + scale\_x\_log10() + labs(title = 'Histogram of Price(log) by Neighborhood type ', x = 'Price', y = 'Count')+ facet\_wrap(~neighbourhood\_group)

## Warning: Transformation introduced infinite values in continuous x-axis  
  
## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Removed 11 rows containing non-finite values (stat\_bin).

## Warning: Removed 11 rows containing non-finite values (stat\_density).



### Correlation

I want to check the correlation between Price and Number of reviews, Minimum nights and Availability\_365 variables

# Reading numberic variables and checking for correlation  
df <- airbnb.df %>% select("price","number\_of\_reviews","minimum\_nights", "availability\_365")  
cor(df)

## price number\_of\_reviews minimum\_nights  
## price 1.00000000 -0.04795423 0.04279933  
## number\_of\_reviews -0.04795423 1.00000000 -0.08011607  
## minimum\_nights 0.04279933 -0.08011607 1.00000000  
## availability\_365 0.08182883 0.17202758 0.14430306  
## availability\_365  
## price 0.08182883  
## number\_of\_reviews 0.17202758  
## minimum\_nights 0.14430306  
## availability\_365 1.00000000

These variables are not highly linearly correlated(values < 0.1) to Price variable.

#Quantiles of price  
summary(airbnb.df$price)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 69.0 106.0 152.7 175.0 10000.0

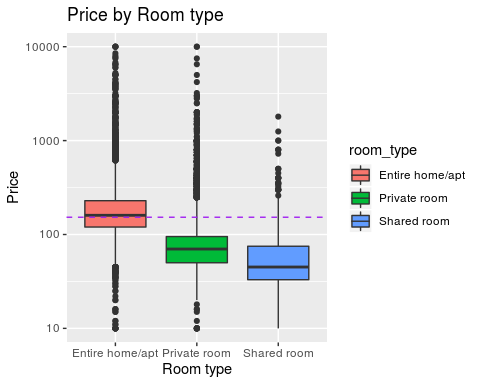
Median Airbnb rental price is $106 in NYC and Max price being $10k.

### Plots for Price Vs other variables

#Room type vs Price plot  
ggplot(airbnb.df, aes(x = room\_type, y = price)) +  
 geom\_boxplot(aes(fill = room\_type)) + scale\_y\_log10() +  
 labs(title = 'Price by Room type', x = 'Room type', y = 'Price') + geom\_hline(yintercept = mean(airbnb.df$price), color = "purple", linetype = 2)

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).

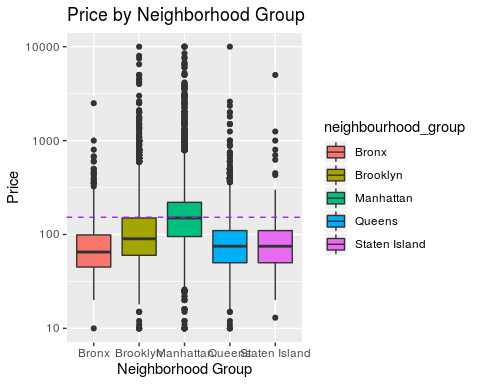


From the above plot, we can see Home/Apt’s mean price is higher than other room types.  
So, room type is definitely related to Price variable.

# Neighborhood group vs Price plot  
ggplot(airbnb.df, aes(x = neighbourhood\_group, y = price)) +  
 geom\_boxplot(aes(fill = neighbourhood\_group)) + scale\_y\_log10() +  
 labs(title = 'Price by Neighborhood Group', x = 'Neighborhood Group', y = 'Price') + geom\_hline(yintercept = mean(airbnb.df$price), color = "purple", linetype = 2)

## Warning: Transformation introduced infinite values in continuous y-axis

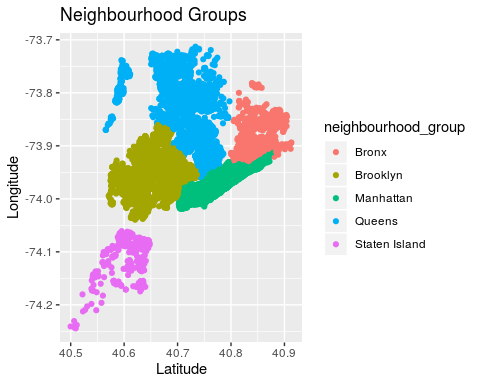
## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).



From the above plot, we can see Airbnb hotels are expensive in Manhattan and Bronx being the cheapest!  
So, Neighborhood Group has relationship with Price.

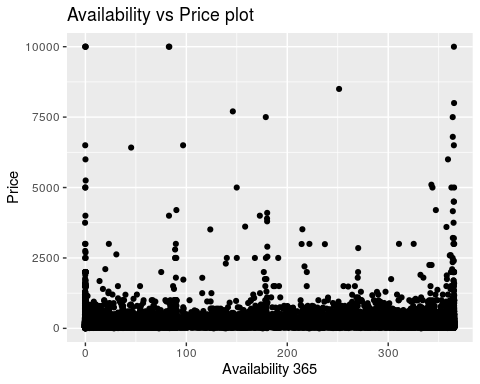
### Scatter plots

#Neighbourhood group with latitude and logitude  
ggplot(data = airbnb.df, aes(latitude, longitude, color = neighbourhood\_group)) + geom\_point() + labs(title = "Neighbourhood Groups", x = "Latitude", y = "Longitude")



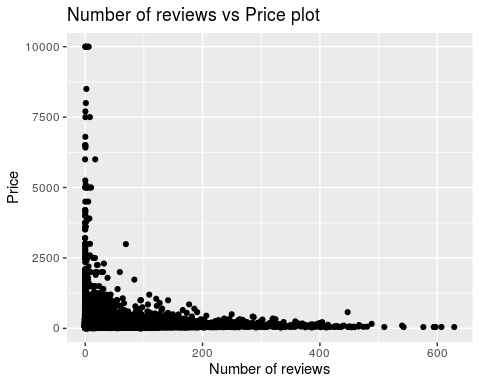
The above map shows the neighbourhood group map and appropriately colors it.

#Availability vs Price plot  
ggplot(airbnb.df, aes(availability\_365, price)) + geom\_jitter() + labs(title = 'Availability vs Price plot', x = 'Availability 365', y = 'Price')



There is no particular linear correlation between Price vs Availability variable alone.

#Number of reviews vs Price plot  
ggplot(airbnb.df, aes(number\_of\_reviews, price)) + geom\_jitter() + labs(title = 'Number of reviews vs Price plot', x = 'Number of reviews', y = 'Price')



### Top 10 Prices

##Price values are most occuring in the dataset  
top\_10 <- airbnb.df %>% select(price) %>% group\_by(price) %>% summarise(count = n()) %>%  
arrange(desc(count)) %>% top\_n(10)

## Selecting by count

top\_10

## # A tibble: 10 x 2  
## price count  
## <int> <int>  
## 1 100 2051  
## 2 150 2047  
## 3 50 1534  
## 4 60 1458  
## 5 200 1401  
## 6 75 1370  
## 7 80 1272  
## 8 65 1190  
## 9 70 1170  
## 10 120 1130

From the above aggregate counts, the most used Price value in the dataset is 100. So, no wonder the median is 106.

## Linear regression

Let’s build model for target variable “Price” with other independent variables

### Model 1

# linear regression model for Price  
price\_model1 = lm(price ~ latitude + longitude + room\_type + minimum\_nights + availability\_365 + neighbourhood\_group, data = airbnb.df)   
summary(price\_model1) #Review the results

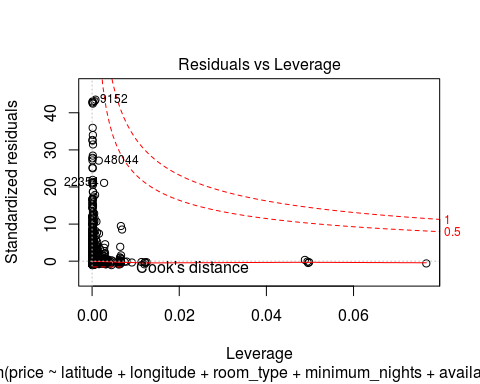
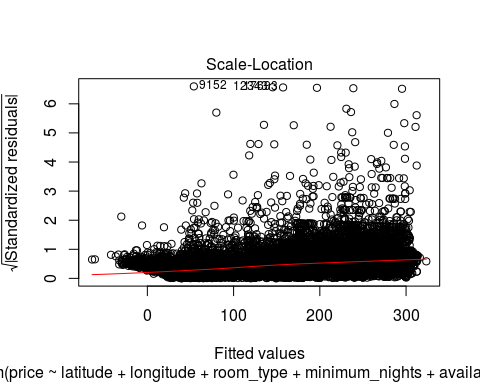
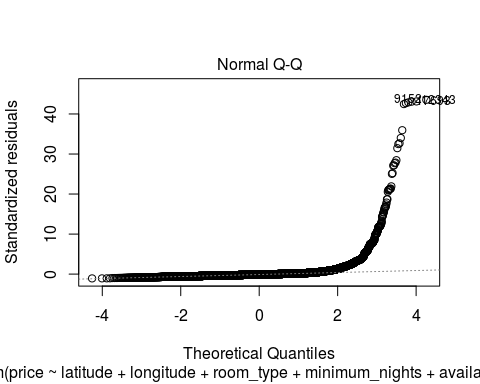
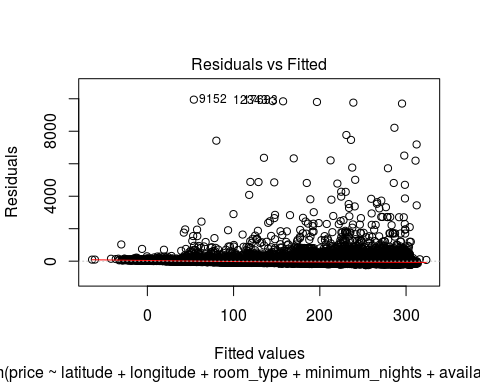
##   
## Call:  
## lm(formula = price ~ latitude + longitude + room\_type + minimum\_nights +   
## availability\_365 + neighbourhood\_group, data = airbnb.df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -244.3 -62.2 -24.7 14.8 9946.1   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -2.915e+04 3.210e+03 -9.083 < 2e-16  
## latitude -1.979e+02 3.132e+01 -6.320 2.64e-10  
## longitude -5.058e+02 3.597e+01 -14.062 < 2e-16  
## room\_typePrivate room -1.052e+02 2.160e+00 -48.707 < 2e-16  
## room\_typeShared room -1.380e+02 6.886e+00 -20.040 < 2e-16  
## minimum\_nights 1.210e-02 5.123e-02 0.236 0.813216  
## availability\_365 1.653e-01 8.078e-03 20.460 < 2e-16  
## neighbourhood\_groupBrooklyn -3.293e+01 8.773e+00 -3.753 0.000175  
## neighbourhood\_groupManhattan 2.756e+01 7.967e+00 3.459 0.000542  
## neighbourhood\_groupQueens -4.651e+00 8.459e+00 -0.550 0.582398  
## neighbourhood\_groupStaten Island -1.514e+02 1.668e+01 -9.077 < 2e-16  
##   
## (Intercept) \*\*\*  
## latitude \*\*\*  
## longitude \*\*\*  
## room\_typePrivate room \*\*\*  
## room\_typeShared room \*\*\*  
## minimum\_nights   
## availability\_365 \*\*\*  
## neighbourhood\_groupBrooklyn \*\*\*  
## neighbourhood\_groupManhattan \*\*\*  
## neighbourhood\_groupQueens   
## neighbourhood\_groupStaten Island \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 228.5 on 48884 degrees of freedom  
## Multiple R-squared: 0.09488, Adjusted R-squared: 0.09469   
## F-statistic: 512.4 on 10 and 48884 DF, p-value: < 2.2e-16

Linear regression for target variable Price with independent variables latitude, longitude, room\_type, minimum\_nights , availability\_365 and neighbourhood\_group.

Interpretion:  
1. Except minimum\_nights, all other variables are statistically significant. Because p-value is less than 1%.  
2. R-square is small (9.4%), that means the model doesn’t explain the variation to a great extent. 3. Max Residuals tells us there are outliers in the dataset.

The model can be viewed visually below.

plot(price\_model1)



Normal Q-Q plot tells us that the model is not a great fit.  
Outliers causing leverage to be high on certain points.

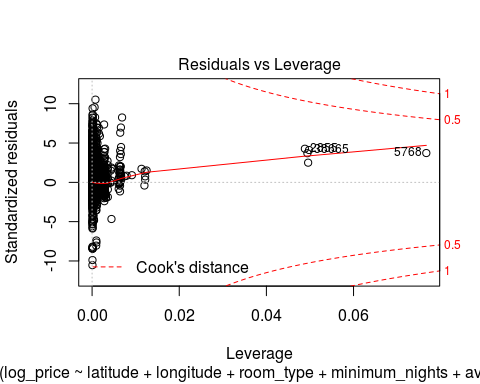
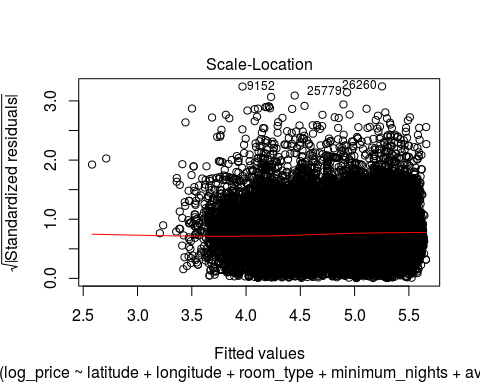
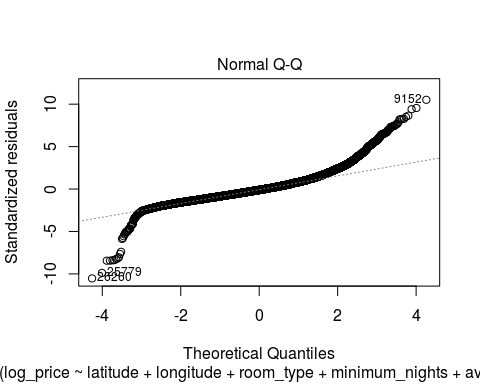
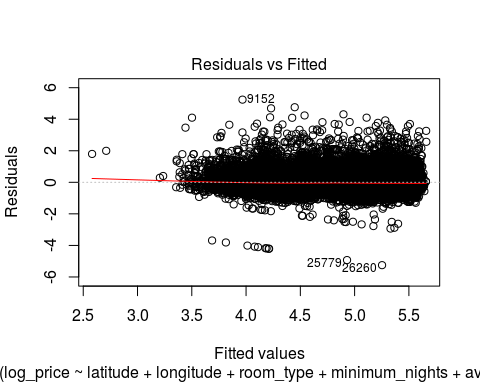
So, I want to test the regression with log transformation on the price, since log gave us better correlation earlier.

### Model 2 - with Log transformation

airbnb.df$log\_price <- log(airbnb.df$price + min(airbnb.df$price) + 1)  
#training.df$log\_price <- log(training.df$price + min(training.df$price) + 1)  
#testing.df$log\_price <- log(testing.df$price + min(testing.df$price) + 1)  
  
  
#all(is.na(log(airbnb.df$price)))  
# linear regression model for Price  
price\_model2 = lm(log\_price ~ latitude + longitude + room\_type + minimum\_nights + availability\_365 + neighbourhood\_group, data = airbnb.df, na.action=na.exclude)   
summary(price\_model2) #Review the results

##   
## Call:  
## lm(formula = log\_price ~ latitude + longitude + room\_type + minimum\_nights +   
## availability\_365 + neighbourhood\_group, data = airbnb.df,   
## na.action = na.exclude)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.2528 -0.3115 -0.0537 0.2347 5.2433   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -1.935e+02 7.005e+00 -27.625 < 2e-16  
## latitude -6.368e-01 6.836e-02 -9.315 < 2e-16  
## longitude -3.034e+00 7.850e-02 -38.654 < 2e-16  
## room\_typePrivate room -7.467e-01 4.714e-03 -158.406 < 2e-16  
## room\_typeShared room -1.143e+00 1.503e-02 -76.084 < 2e-16  
## minimum\_nights -1.742e-03 1.118e-04 -15.576 < 2e-16  
## availability\_365 6.964e-04 1.763e-05 39.502 < 2e-16  
## neighbourhood\_groupBrooklyn -4.707e-02 1.915e-02 -2.459 0.014  
## neighbourhood\_groupManhattan 2.604e-01 1.739e-02 14.974 < 2e-16  
## neighbourhood\_groupQueens 8.166e-02 1.846e-02 4.423 9.75e-06  
## neighbourhood\_groupStaten Island -8.208e-01 3.641e-02 -22.545 < 2e-16  
##   
## (Intercept) \*\*\*  
## latitude \*\*\*  
## longitude \*\*\*  
## room\_typePrivate room \*\*\*  
## room\_typeShared room \*\*\*  
## minimum\_nights \*\*\*  
## availability\_365 \*\*\*  
## neighbourhood\_groupBrooklyn \*   
## neighbourhood\_groupManhattan \*\*\*  
## neighbourhood\_groupQueens \*\*\*  
## neighbourhood\_groupStaten Island \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4987 on 48884 degrees of freedom  
## Multiple R-squared: 0.4857, Adjusted R-squared: 0.4856   
## F-statistic: 4617 on 10 and 48884 DF, p-value: < 2.2e-16

plot(price\_model2)



Normal Q-Q plot tells us this model is a better fit than previous model.  
And there is a great improvement in R-squared value (48% from 8% in the previous model).  
So, I conclude log(price) model is a great fit to the dataset. All variables in the model are statistically significant. Leverage plot tells us there are outliers in the dataset.

### End of assignment