lab-05

Justin Chan

2/16/2022

Lab 5

Loading required packages for the lab:

```
library(tidyverse)
library(stringr)
library(knitr)
library(skimr)
library(broom)
```

Read in dataset for Santa Cruz County Air BnBs:

```
airbnb <- read_csv("/Users/chanj4/Desktop/School/STAT108/lab-05/listings.csv")
```

```
## Rows: 1489 Columns: 18
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): name, host_name, neighbourhood, room_type
## dbl  (11): id, host_id, latitude, longitude, price, minimum_nights, number_o...
## lgl   (2): neighbourhood_group, license
## date  (1): last_review
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Exercises

Data Wrangling and EDA

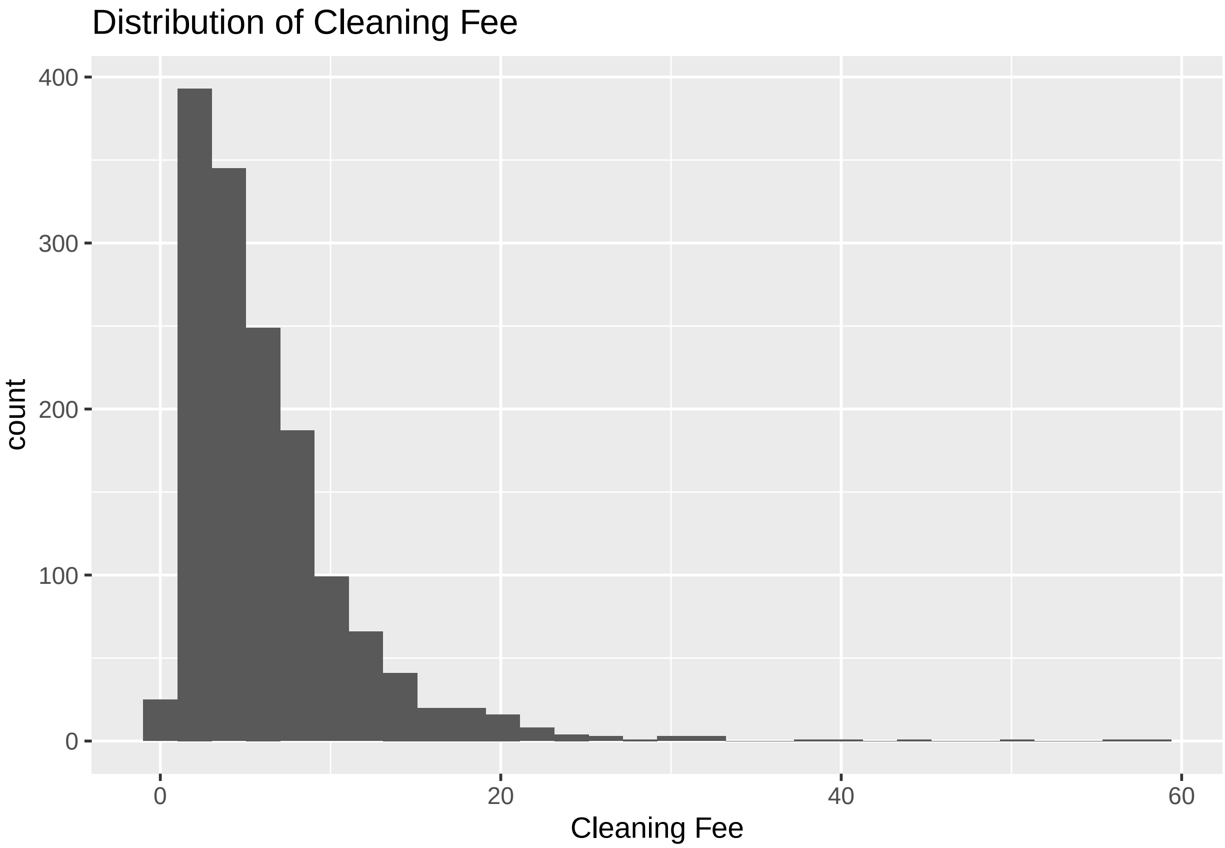
1. Some Airbnb rentals have cleaning fees, and we want to include the cleaning fee when we calculate the total rental cost. Create a variable call cleaning_fee calculated as the 2% of the price per night.

```
airbnb <- airbnb %>%
  mutate(cleaning_fee = price * .02)
```

2. Visualize the distribution of cleaning_fee and display the appropriate summary statistics. Use the graph and summary statistics to describe the distribution of cleaning_fee.

```
ggplot(data = airbnb, aes(x = cleaning_fee)) +
  geom_histogram() +
  labs(title = "Distribution of Cleaning Fee",
       x = "Cleaning Fee")
```

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



```
airbnb %>%
  summarise(min = min(cleaning_fee),
            q1 = quantile(cleaning_fee, probs = c(0.25)),
            median = median(cleaning_fee),
            q3 = quantile(cleaning_fee, probs = c(0.75)),
            max = max(cleaning_fee),
            iqr = IQR(cleaning_fee),
            mean = mean(cleaning_fee),
            std_dev = sd(cleaning_fee)
            )
```

```
## # A tibble: 1 x 8
##     min    q1 median    q3   max   iqr  mean std_dev
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1  0.62  2.88      5  8.06    59  5.18  6.38    5.39
```

Using the graph from this exercise, it seems that cleaning_price seems to have a unimodel and skewed to the right shape where there are a high concentration of cleaning fees per night a bit above zero which tend to decrease the higher the cleaning fee in terms of counts. There is also quite a few outliers which I would say above 25 dollars per night in terms for the cleaning fee. Moving on to the summary statistics, if we compare the center measures, median 5 to the mean which is 6.38, we can clearly also see that the graph is skewed to the right. We can also see the graph is skewed right based on the spacing of min, q1, median, q3, and max which there is 25 percent of data between each of these values. One can see that the spacing between each of these variables increases after the median which show the graph being skewed to the right due to outliers. Also, the summary statistics show the standard deviation of the cleaning_fee to be 5.39 and the IQR to be 5.18 which gives a bit more insight to the spread.

3. Next, let's examine the neighbourhood.

- How many different categories of neighbourhood are in the dataset? Show code and output to support your answer.

```
unique(airbnb$neighbourhood)
```

```
## [1] "Unincorporated Areas"  "City of Santa Cruz"    "City of Capitola"
## [4] "City of Scotts Valley" "City of Watsonville"
```

Based on the code and output, there are five different categories of neighbourhood in this dataset from AirBnBs.

- Which 3 neighborhoods are most common in the data? These 3 property types make up what percent of the observations in the data? Show code and output to support your answer.

```
airbnb %>%
  group_by(neighbourhood) %>%
  tally()
```

```
## # A tibble: 5 x 2
##   neighbourhood             n
##   <chr>                 <int>
## 1 City of Capitola        218
## 2 City of Santa Cruz      369
## 3 City of Scotts Valley    26
## 4 City of Watsonville      15
## 5 Unincorporated Areas    861
```

The three neighbourhoods that are most common in the data are Unincorporated Areas, City of Capitola, and City of Santa Cruz. The percentage that they make up in the dataset is 97.25 percent and can be calculated as:

$$Percentage_{ThreeCommonNeightbourhoods} = \frac{218 + 369 + 861}{218 + 369 + 861 + 26 + 15} = \frac{1448}{1489} = 97.25\%$$

4. Since an overwhelming majority of the observations in the data are one of the top 3 cities, we would like to create a simplified version of the neighbourhood variable that has 4 categories.

```r
airbnb <- airbnb %>%
  mutate(neighbourhood = fct_lump(neighbourhood, n = 3))

airbnb %>%
  count(neighbourhood)
```

```
## # A tibble: 4 x 2
##   neighbourhood            n
##   <fct>                <int>
## 1 City of Capitola       218
## 2 City of Santa Cruz     369
## 3 Unincorporated Areas   861
## 4 Other                   41
```

We combined the City of Watsonville and City of Scotts Valley group for neighbourhoods into one category called Other so that we have four categories in total: the three largest neighbourhoods + other). There is evidence that we combined the rest of the groups based on the counts.

Create a new variable called neigh_simp that has 4 categories: the three from the previous question and “Other” for all other places. Be sure to save the new variable in the data frame.

```r
airbnb <- read_csv("/Users/chanj4/Desktop/School/STAT108/lab-05/listings.csv")
```

```
## Rows: 1489 Columns: 18

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): name, host_name, neighbourhood, room_type
## dbl  (11): id, host_id, latitude, longitude, price, minimum_nights, number_o...
## lgl   (2): neighbourhood_group, license
## date  (1): last_review

## 
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
airbnb <- airbnb %>%
  mutate(cleaning_fee = price * .02)

airbnb <- airbnb %>%
  mutate(neigh_simp = fct_lump(neighbourhood, n = 3))

airbnb %>%
  count(neigh_simp)
```

```
## # A tibble: 4 x 2
##   neigh_simp               n
##   <fct>                <int>
## 1 City of Capitola       218
## 2 City of Santa Cruz     369
## 3 Unincorporated Areas   861
## 4 Other                   41
```

I recreated the dataset after modifying it in the previous exercise so instead of changing the neighbourhood field, I added the new fieldm neigh_simp.

5.What are the 4 most common values for the variable minimum_nights? Which value in the top 4 stands out? What is the likely intended purpose for Airbnb listings with this seemingly unusual value for minimum_nights? Show code and output to support your answer.

```
airbnb %>%
  count(minimum_nights)
```

```
## # A tibble: 21 x 2
##    minimum_nights     n
##             <dbl> <int>
##  1              1   420
##  2              2   571
##  3              3   223
##  4              4    56
##  5              5    32
##  6              6    10
##  7              7    30
##  8              8     1
##  9             10     3
## 10             14     7
## # ... with 11 more rows
```

The four most common variables for the minimum_nights for this AirBnb dataset in Santa Cruz is 1, 2, 3, and 30 nights. The value in the top 4 that stands out is 30 because it has the most counts for that many days which tend to have negative correlation where counts decrease as minimum nights increases. I think the intended purpose for at least staying thirty nights is at least a guarantee to more income which has people staying at least staying one month which makes cleaning eaiser. Furthermore, I think it could be targeted so people could have month stays which some people due for travel purposes..

Airbnb is most commonly used for travel purposes, i.e. as an alternative to traditional hotels, so we only want to include Airbnb listings in our regression analysis that are intended for travel purposes. Filter airbnb so that it only includes observations with minimum_nights <= 3.

```
airbnb_filtered <- airbnb %>%
  filter(minimum_nights >= 3)
```

6. For the response variable, we will use the total cost to stay at an Airbnb location for 3 nights. Create a new variable called price_3_nights that uses price and cleaning_fee to calculate the total cost to stay at the Airbnb property for 3 nights. Note that the cleaning fee is only applied one time per stay.

```
airbnb_filtered <- airbnb_filtered %>%
  mutate(price_3_nights = price * 3 + cleaning_fee)
```

Be sure price is in the correct format before calculating the new variable.

Regression

7.Fit a regression model with the response variable from the previous question and the following predictor variables: neigh_simp, number_of_reviews, and reviews_per_month. Display the model with the inferential statistics and confidence intervals for each coefficient.

```
modelfirst <- lm(price_3_nights ~ neigh_simp + number_of_reviews + reviews_per_month, data = airbnb_fil
tidy(modelfirst, conf.int=TRUE, conf.level = 0.95) %>%
  kable(format="markdown", digits=3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	1272.324	114.141	11.147	0.000	1047.993	1496.654
neigh_simpCity of Santa Cruz	-200.297	128.389	-1.560	0.119	-452.631	52.037
neigh_simpUnincorporated Areas	-17.609	122.278	-0.144	0.886	-257.932	222.714
neigh_simpOther	-735.947	276.679	-2.660	0.008	-1279.726	-192.168
number_of_reviews	-2.186	0.797	-2.744	0.006	-3.752	-0.621
reviews_per_month	14.306	44.803	0.319	0.750	-73.750	102.361

8. Interpret the coefficient of number_of_reviews and its 95% confidence interval in the context of the data.

The coefficient of number_of_reviews is simply the amount the change for our response variable, price_3_nights when increasing the number_of_reviews by 1. The coefficent can be interpreted that if a listing had one more review, it is predicted that on average that the total cost of a three night stay for the listing would be 2.19 dollars less. The 95% confidence interval for this coefficient is [-3.752, -0.621] which we believe with 95% confidence that the true coefficient for predicting price_3_nights from number_of_reviews given all other factors remain constant, is between this interval. From a statistical standpoint, we know that if we took a random sample multiple times, then we know 95% of our confidence intervals will contain the true coefficient for predicting price_3_nights from number_of_reviews.

9. Interpret the coefficient of neigh_simpCity of Santa Cruz and its 95% confidence interval in the context of the data.

The coefficient of neigh_simpCity of Santa Cruz tells us the estimate for price_3_nights from neigh_simpCity of Santa Cruz relative to neigh_simpCity of Capitola. The coefficient for neigh_simpCity of Santa Cruz is -200.297 where it can be said that price_3_nights decreases by 200.30 dollars on average when comparing AirBnB 3 night price from the City of Santa Cruz to the 3 night average prices from the City of Capitola. The 95% confidence interval of the coefficient of neigh_simpCity of Santa Cruz is [-452.631, 52.037] where it can be said that we are 95% confident that the true coefficent of neigh_simpCity of Santa Cruz lies between the confidence interval which is relative to the neigh_simpCity of Capitola value. Statistically, this can be explained that if a random sampling occurred, 95% of the confidence intervals generated for neigh_simpCity of Santa Cruz would contain the true coefficient relative to the coefficient of neigh_simpCity of Capitola.

10. Interpret the intercept in the context of the data. Does the intercept have a meaningful interpretation? Briefly explain why or why not.

The intercept, 1272.324 is the estimated value in dollars of the average price_3_night for an AirBnB in the City of Capitola where it has 0 reviews and 0 reviews left per month from the AirBnB dataset. This intercept does not have a meaningful interpretation in my opinion because it may be useful to know the average cost of 3 night stay in the City of Santa Cruz, it seems extremely unrealistic that a AirBnb listing would have 0 reviews and not have any reviews being added to the listing each month.

11. Suppose your family is planning to visit Santa Cruz over Spring Break, and you want to stay in an Airbnb. You find an Airbnb that is in Scotts Vallye, has 10 reviews, and 5.14 reviews per month. Use the model to predict the total cost to stay at this Airbnb for 3 nights. Include the appropriate 95% interval with your prediction.

```
predict(modelfirst, data.frame(neigh_simp = "Other", number_of_reviews = 10, reviews_per_month = 5.14),
```

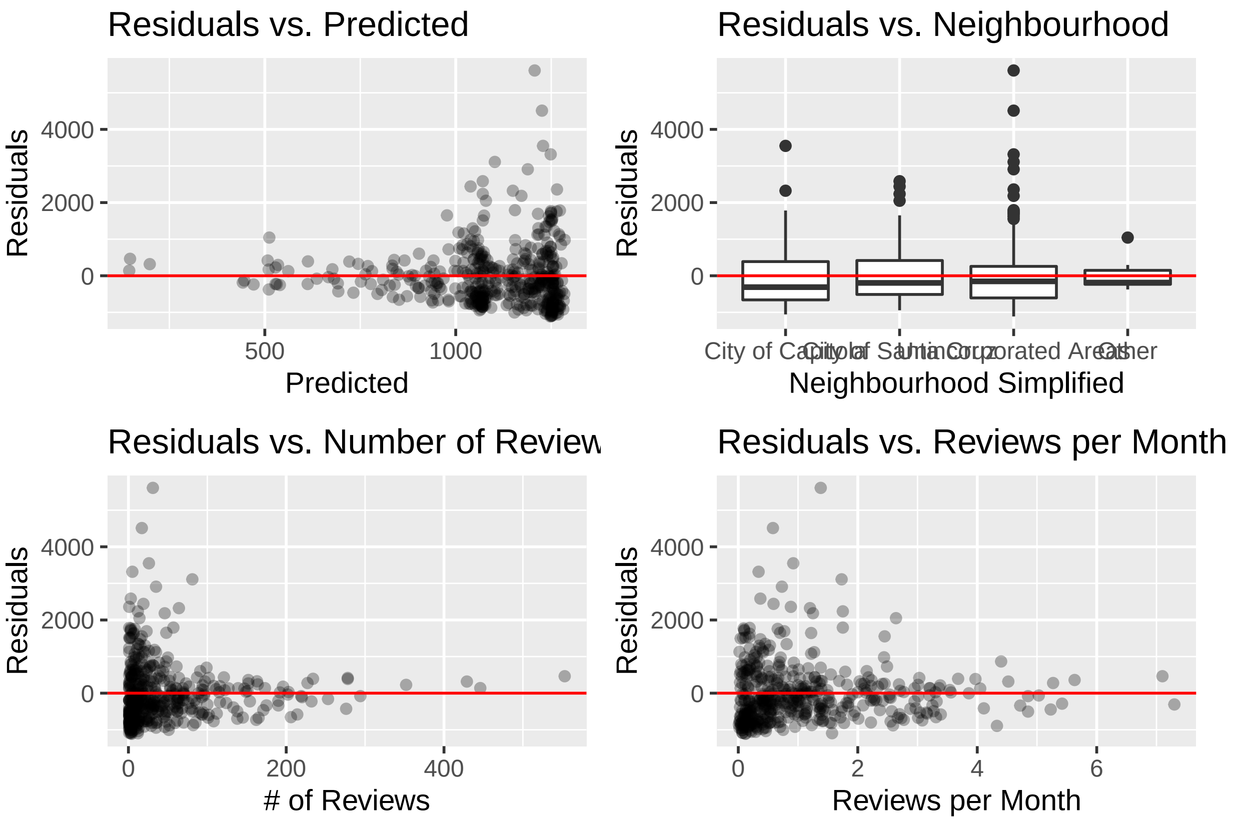
```
##       fit      lwr      upr
## 1 588.043 -1188.65 2364.736
```

The predicted cost of a three night stay at the City of Scotts Valley which was placed in the Other category for neighbourhood, has 10 reviews, and 5.14 reviews per month is 588.04 dollars. The 95 percent confidence interval for the cost is [-1199.65, 2364.74] where we can say that we are 95% confidence that the true cost of 3 night stay of AirBnB with these features is within this interval. The interval does not make too much sense though because it saying that we can get paid for our stay or pay negative amount of dollars for the stay. In other words, we know that if we took a random samples and generated a confidence interval for each of these samples, 95% of our confidence intervals would contain the true three night cost of AirBnB with these specifications.

12. Now check the assumptions for your regression model. Should you be confident on interpreting the inferential results of your model?

The assumptions that should be checked for the regression model is linearity, Normal, constant variance, and independence condition. To start off, we want to check the model for the linearity condition where we want plot the residuals versus the predicted values and the residuals versus each predictor variable to ensure there is no pattern. Since there is clear pattern when it comes to plotting, it can be said that this passes the linearity condition.

```
library(patchwork)

residual_airbnb_aug <- augment(modelfirst)

p1 = ggplot(data = residual_airbnb_aug, aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Predicted", y = "Residuals", title = "Residuals vs. Predicted")

p2 = ggplot(data = residual_airbnb_aug, aes(x = neigh_simp, y = .resid)) +
  geom_boxplot() +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Neighbourhood Simplified", y = "Residuals", title = "Residuals vs. Neighbourhood")

p3 = ggplot(data = residual_airbnb_aug, aes(x = number_of_reviews, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "# of Reviews", y = "Residuals", title = "Residuals vs. Number of Reviews")

p4 = ggplot(data = residual_airbnb_aug, aes(x = reviews_per_month, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Reviews per Month", y = "Residuals", title = "Residuals vs. Reviews per Month")

p1 + p2 + p3 + p4
```


Residuals vs. Predicted

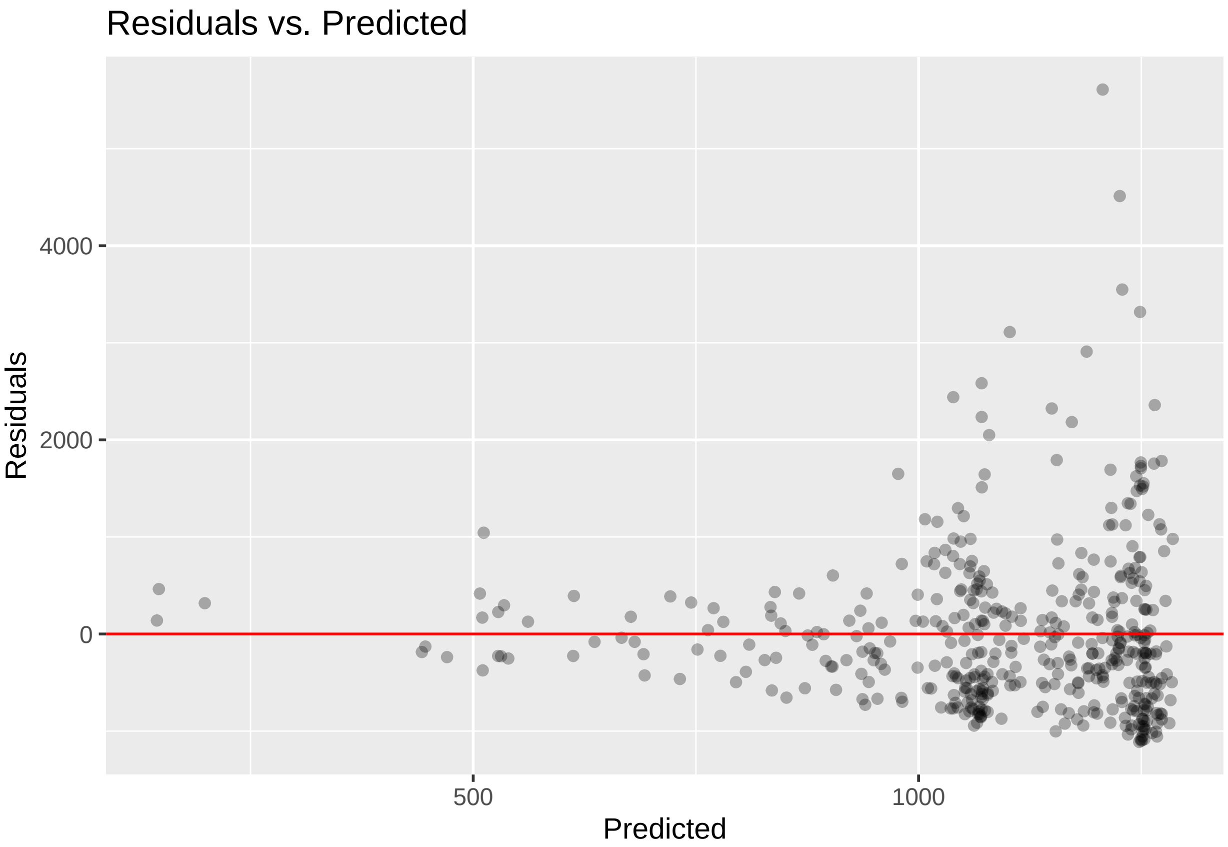
Residuals vs. Neighbourhood

Residuals vs. Number of Review

Residuals vs. Reviews per Month

The next condition we would like to look at is the constant variance condition. The constant variance condition is satisfied if the residuals values stay constant for across predicted values. We can plot residuals versus predicted values to see if residuals values stay constant across all the predicted values. Based on the plot below, we see a fan shape where the smaller predicted values for a three night stay for an airbnb in Santa Cruz have a much smaller variance than those with much larger predicted values. Therefore, we can say that the constant variance condition does not seem to be satisfied.

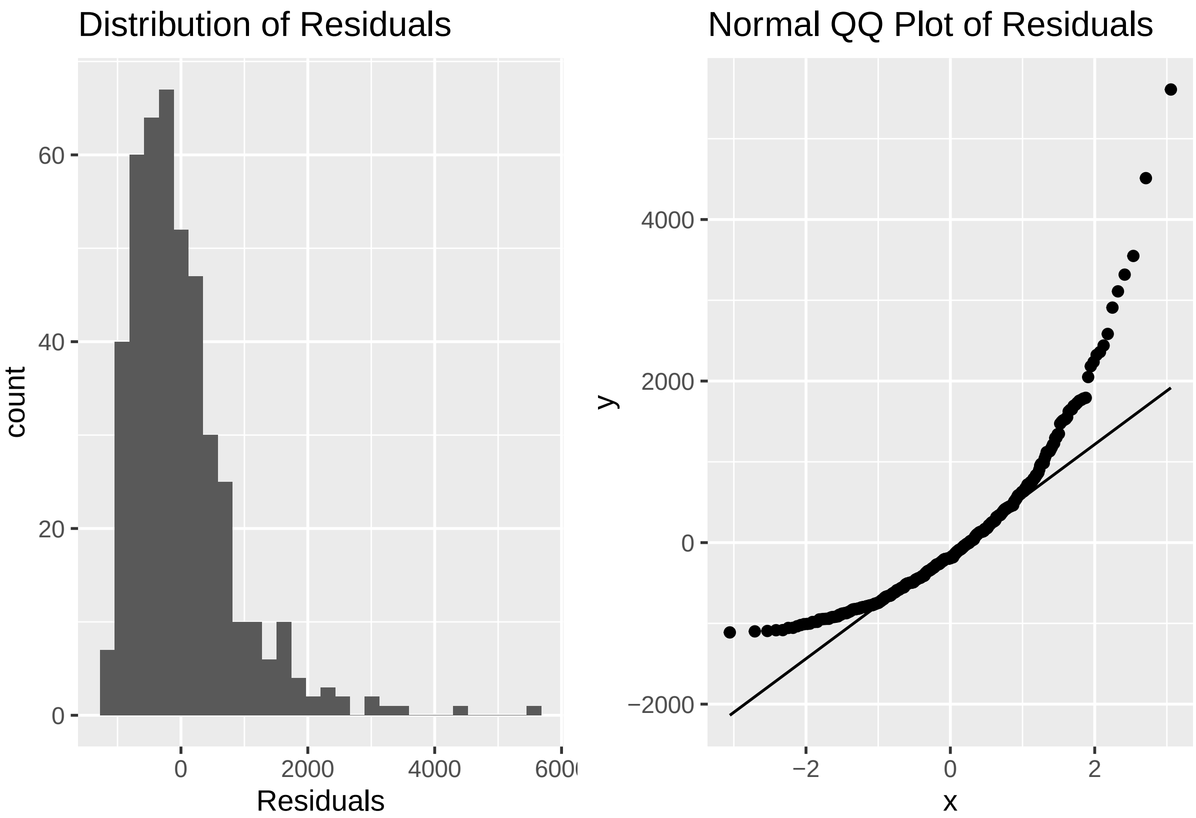
```
ggplot(data = residual_airbnb_aug, aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Predicted", y = "Residuals", title = "Residuals vs. Predicted")
```



After constant variance, we want to check the Normality condition which mentions that given our set of predictor variables, the response variable should follow a Normal distribution across its mean. In other words, we can make a histogram of the residuals and/or normal QQ-plot of the residuals to check the normal condition. The histogram should be approximately unimodal and symmetric and the Normal QQ-Plot should generally follow a straight diagonal line to satisfy the Normal condition. Based on the graph below, our histogram of residuals is unimodal but not symmetric, it is skewed to the right so we can say that this model does not satisfy the Normal condition on our linear model should not be used in this case. Furthermore, we can see the QQPlot does not have all the points on the y = x line where there is upward facing parabola where the ends are above the line.

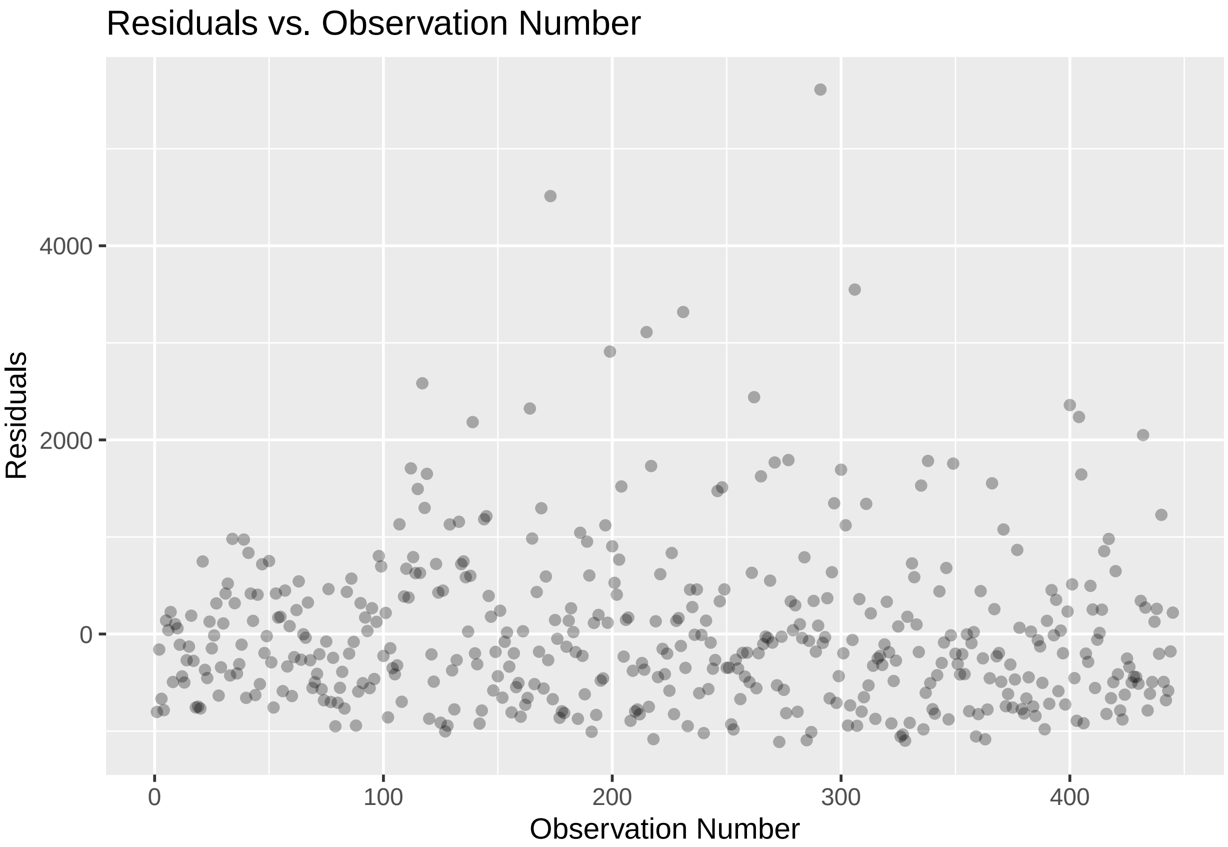
```
p1 = ggplot(data = residual_airbnb_aug, aes(x = .resid)) +
  geom_histogram() +
  labs(x = "Residuals", title = "Distribution of Residuals")

p2 = ggplot(data = residual_airbnb_aug, aes(sample = .resid)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Normal QQ Plot of Residuals")

p1 + p2
```

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



The last condition we want to check is the independence condition. The independence condition states that all observations are independent. Generally, we can state that the collection of data from AirBnB does not affect later collections of data points for the dataset. In other words, we can say that that the selection of one data point for this AirBnB dataset does not affect the next data point selection. However, we can plot the residuals against the observation number to check if this is true to see if there was no pattern in the order of collection. Based on this graph below, we can say that the independence condition is not satisfied because it does not seem that there is constant variance across all the observation order where the beginning observation points do not have much variance in residuals and later points have much larger variations closer to one another.

```r
residual_airbnb_aug <- residual_airbnb_aug %>%
  mutate(obs_num = 1:nrow(residual_airbnb_aug))
ggplot(data = residual_airbnb_aug, aes(x = obs_num, y = .resid)) +
  geom_point(alpha = 0.3) +
  labs(x = "Observation Number",
       y = "Residuals",
       title = "Residuals vs. Observation Number")
```



Since almost all conditions have not been satisfied except possibly the linearity condition, then we can say that our current model should not be used to estimate the price for a 3 night stay for an AirBnB in Santa Cruz.