

STAT 27410 Final Project Proposal

Jake Wei

1. Introduction

Airbnb has become one of the most popular choices for people traveling and seeking housing. However, one problem exists. It is difficult for owners to come up with prices given the location and amenities. It is also hard to predict prices given seasons. Different features such as host ratings, season, location, and number of bedrooms make these two questions immensely complex. In this project, we try to provide insight into predicting prices given different neighborhoods in New York. The data set we picked comes from Inside Airbnb, which is an organization that periodically scrapes data from Airbnb listings. There are 12 columns in the data set. Four are descriptions of the host and 10 can serve as explanatory variables. The explanatory variables are: neighborhood, room_type, accommodation, bed, price, minimum nights, and number of reviews. The response variable is the rent price. Firstly, We will'start with exploratory analysis and data cleaning. Secondly, we'll use the frequentist approach to fit the data and analyze the models. Lastly, we'll describe how we are going to fit the model with respect to a Bayesian approach.

We used R-Markdown to prepare this document.

2. Exploratory Data Cleaning and Data Analysis

To start, we load our New York data from Inside Airbnb, and obtain the summary statistics on the numerical features. From table 1, there are signs of outliers which we will do further analysis to confirm. The two most notable ones are 42 beds and 1250 minimum amounts of nights.

```
data <- read.csv('nydata.csv')

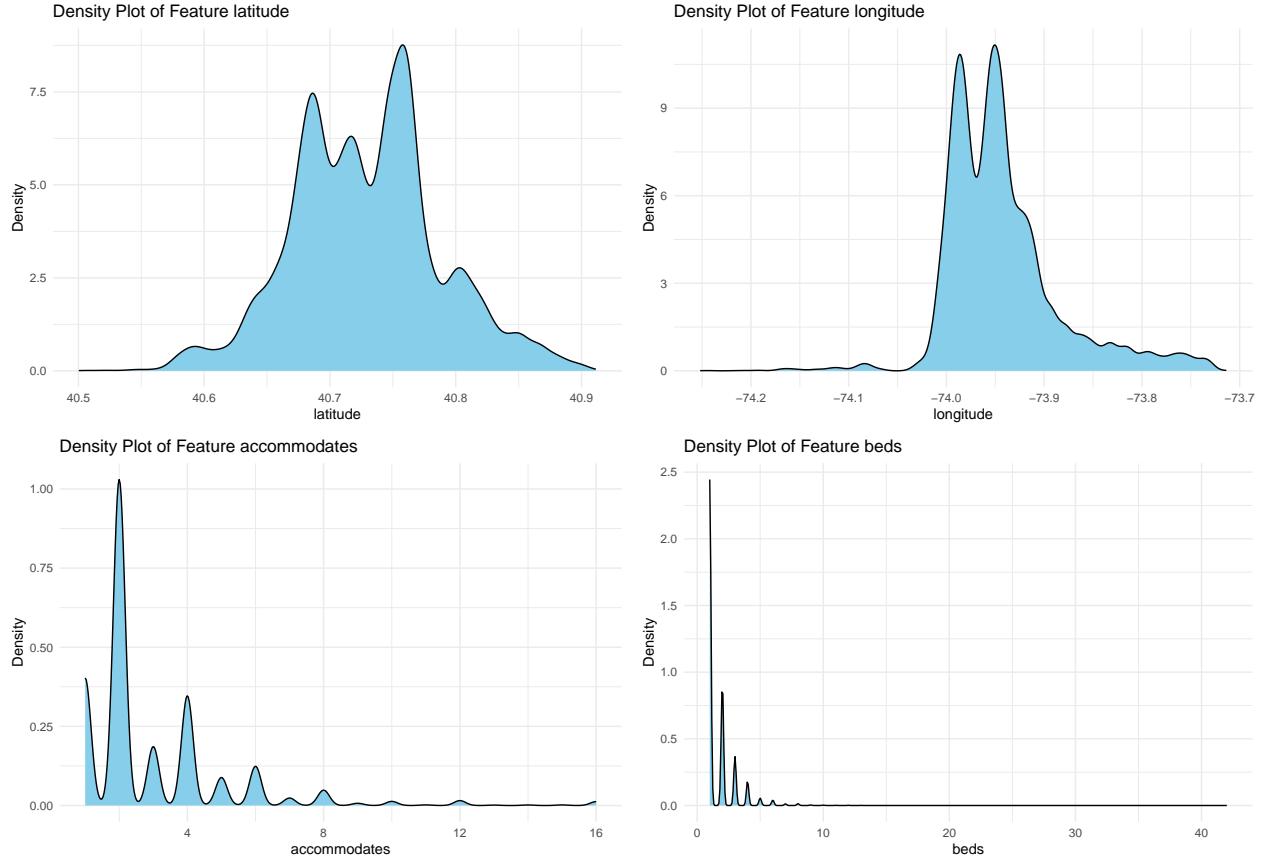
numerical_features <- data[3:9]
categorical_features <- data[10:11]
price = summary(as.data.frame(data$price))
num_and_price <- data[c(3:9,12)]
```

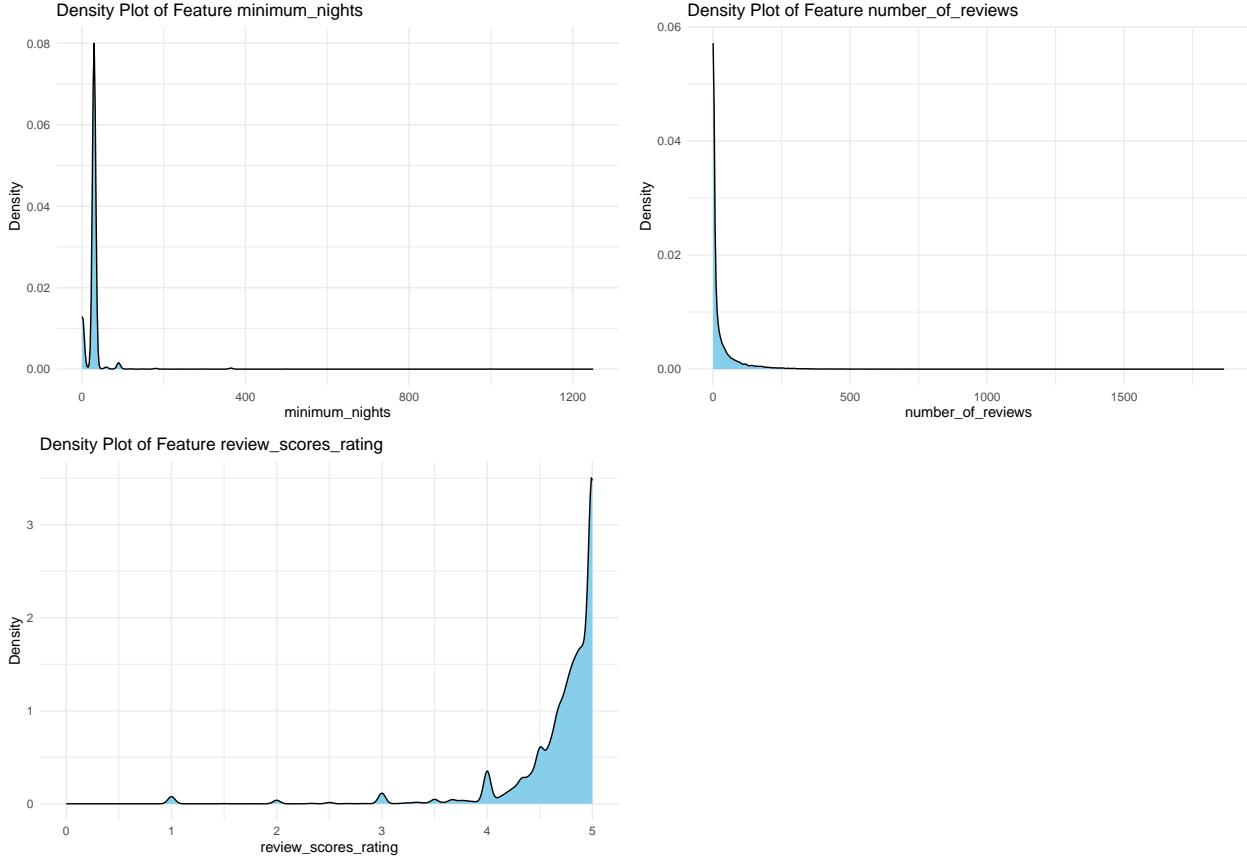
Next, we plot the density for each of the feature to get a sense of the shape. From the plots of each feature, there are signs of skewness from each of the features. The features are also

Table 1: Summary for Numerical Features

Statistic	N	Mean	St. Dev.	Min	Max
latitude	29,091	40.729	0.058	40.500	40.911
longitude	29,091	-73.943	0.059	-74.252	-73.714
accommodates	29,091	2.953	2.182	1	16
beds	28,363	1.710	1.237	1	42
minimum_nights	29,091	29.578	33.708	1	1,250
number_of_reviews	29,091	30.428	65.061	0	1,865
review_scores_rating	20,768	4.696	0.494	0.000	5.000
price	29,091	212.543	946.727	10	100,000

poorly distributed. This also confirms our claim that there are outliers. The main outlier exist with minimum nights stay, price, and amount of beds. To remedy this, we will remove the outliers according to the summary statistics in Table 1.

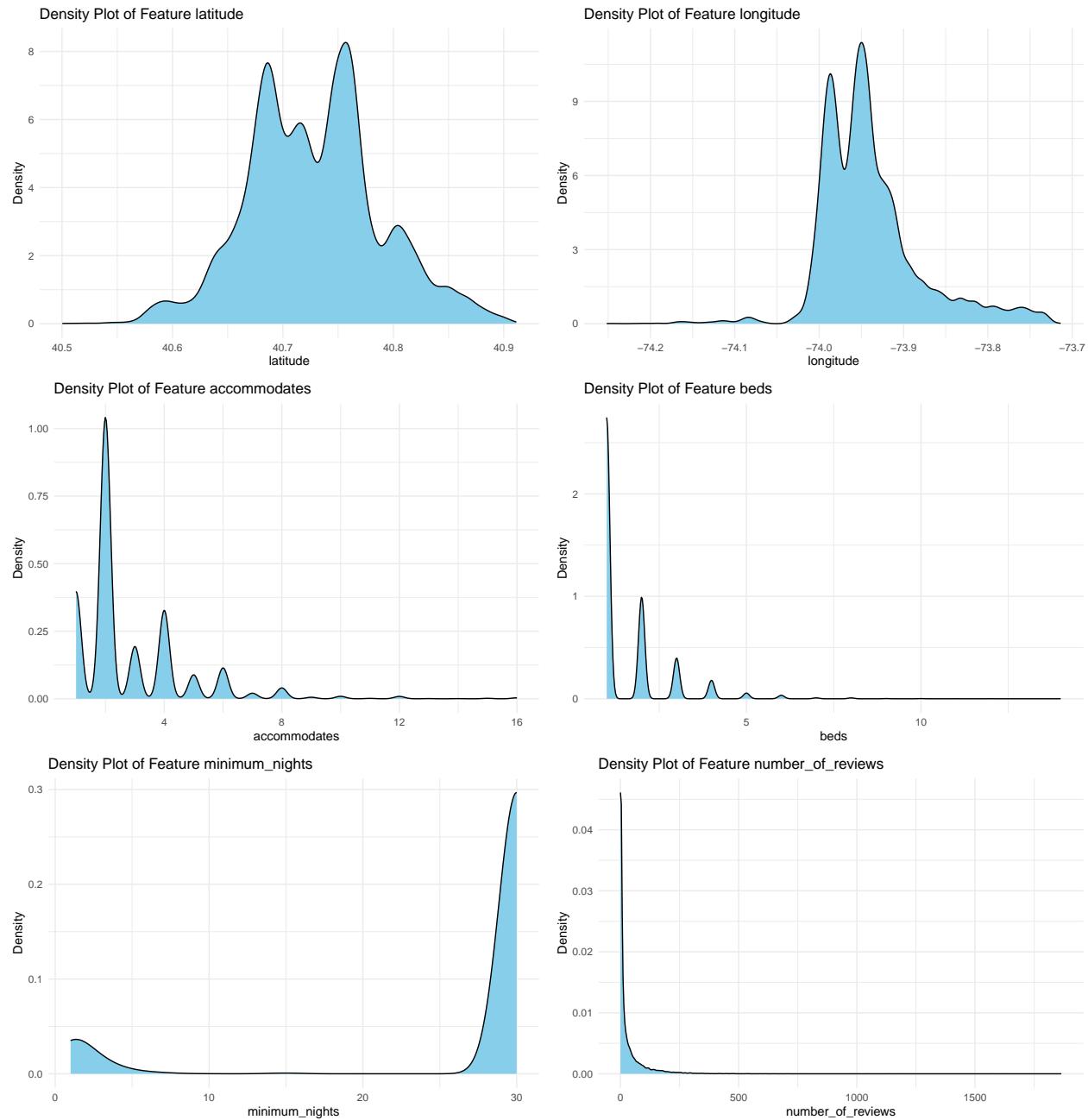




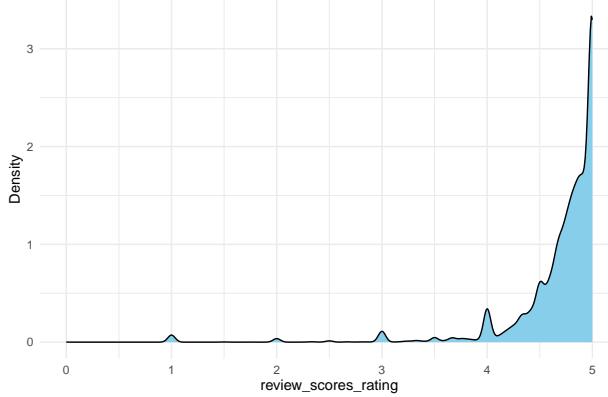
We clean the data by using two filters. The first one is to only look at listing that is under 500 dollars. The second one is that we only look at listings that have a minimum nights of under 30. We choose these two because of the distribution plots above. Most data are clustered around price ≤ 500 and minimum nights ≤ 30 . Looking at table 2, which is the summary stat for the cleaned data, and the density plots for the cleaned data, the skewness problem seems to be alleviated.

Table 2: Figure 2: Summary Stat for Cleaned Numerical Features

Statistic	N	Mean	St. Dev.	Min	Max
latitude	25,604	40.729	0.060	40.500	40.911
longitude	25,604	-73.940	0.060	-74.252	-73.714
accommodates	25,604	2.807	1.882	1	16
beds	24,947	1.648	1.079	1	14
minimum_nights	25,604	26.055	9.750	1	30
number_of_reviews	25,604	33.002	67.832	0	1,865
review_scores_rating	19,234	4.694	0.488	0.000	5.000
price_clean	25,604	146.213	96.199	10	500



Density Plot of Feature review_scores_rating



Next, we check for correlation between the predictors. Because predictors such as number of reviews on a host might be correlated with other predictors that can reflect quality of housing , i.e number of beds and amenities, we want to make sure that our predictors does not show strong correlation. If our predictors are correlated, it can affect our model fit. To check, we created a correlation heat map of predictors. From figure 1 and 2, we see light correlation between some of the predictors. However, since no two predictor exhibit strong correlation, we will fit our model and analyze the model to see if our model is good fit.

Figure 1: Pairs Plot

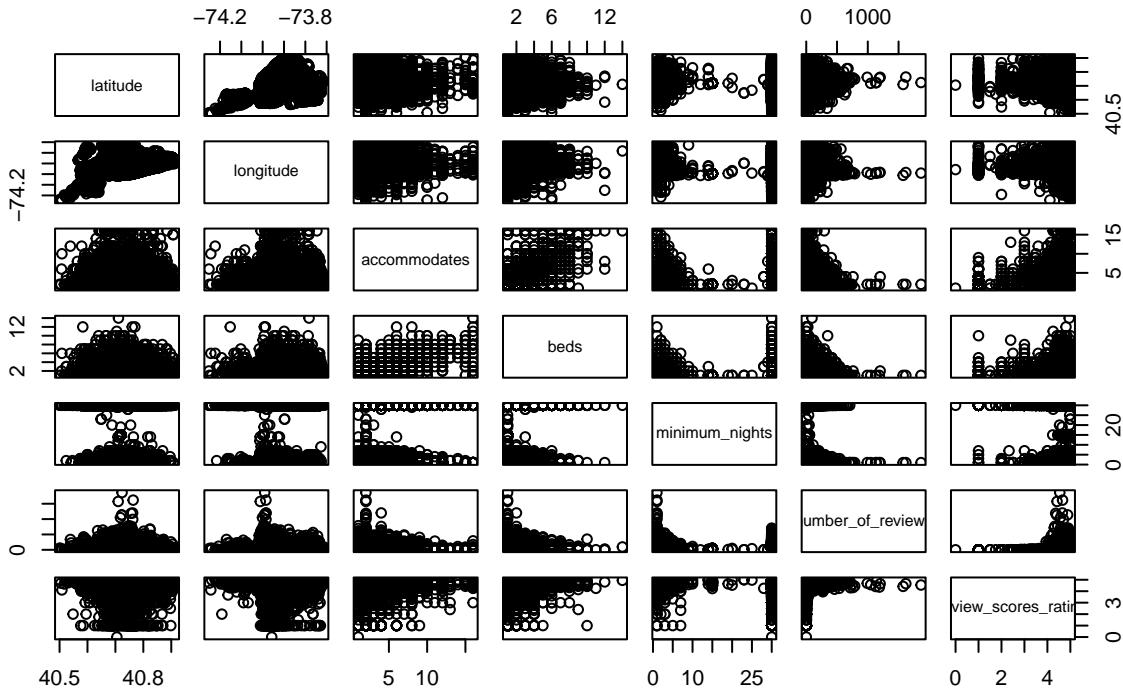
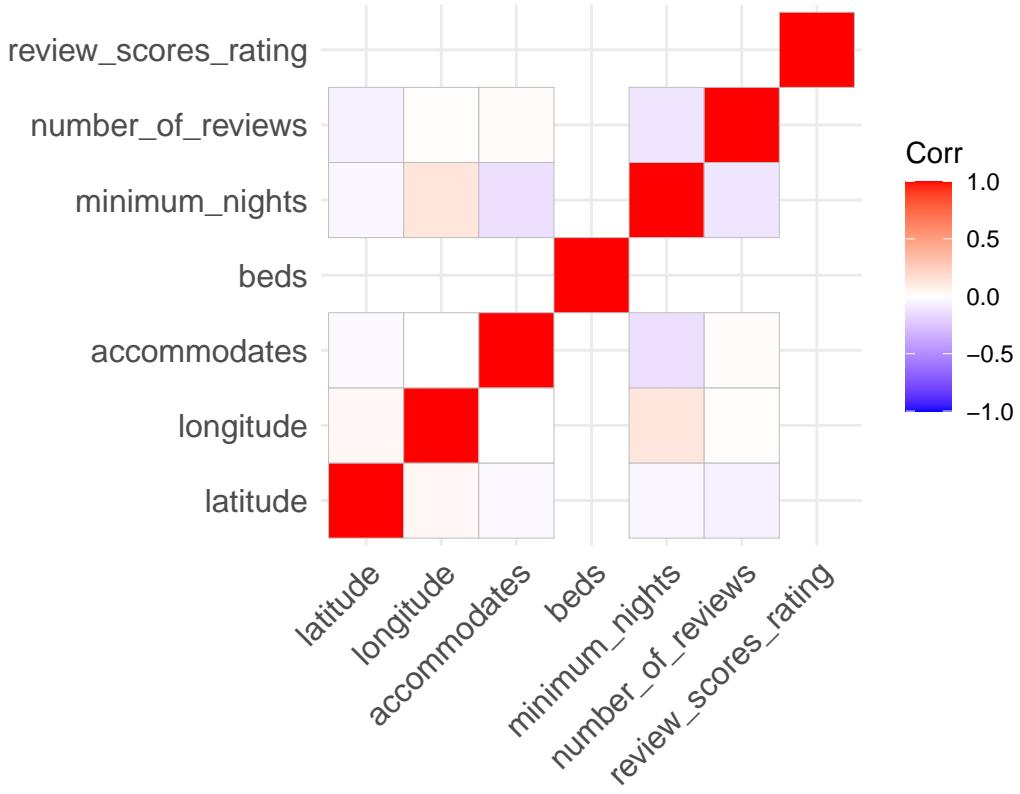


Figure 2: Correlation Heat Map



To conclude our exploratory data analysis, we plot the type of housing in figure 3 and the relationship between neighborhood and price in table 3. From table 3, we break down the prices by the neighborhood. Note that the graph only shows 30 neighborhoods because there are around 200 total neighborhoods in New York. We will insert the total in the appendix.

Figure 4: Type of Housing Count Plot

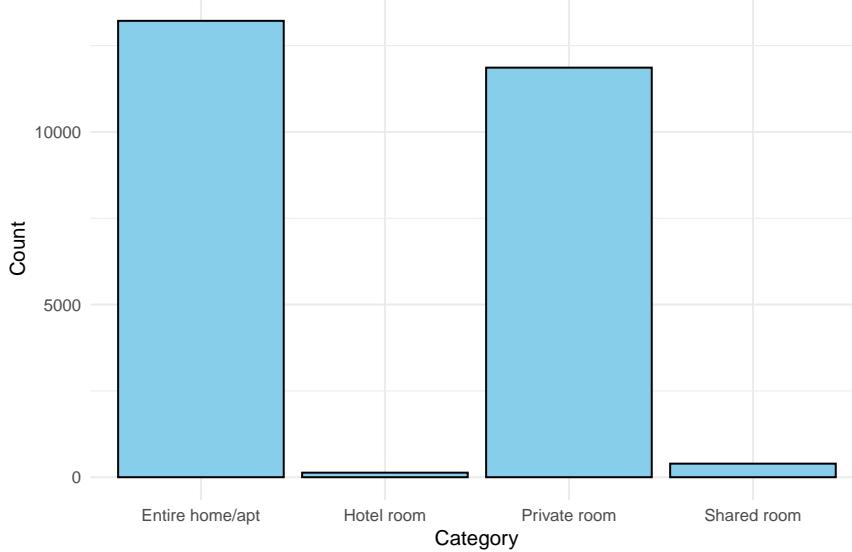


Table 3: Summary Stat by Neighborhood

	neighbourhood_cleansed	mean_price	median_price	min_price	max_price	total_list
1	Allerton	117.825	94	31	500	40
2	Arden Heights	133.750	134.500	100	166	4
3	Arrochar	137.667	97.500	75	350	12
4	Arverne	134.603	108	33	375	63
5	Astoria	111.634	94.500	25	500	418
6	Bath Beach	147.667	110	40	399	24
7	Battery Park City	185.650	180.500	10	360	40
8	Bay Ridge	106.707	85	40	350	99
9	Bay Terrace	149.500	124.500	99	250	4
10	Bay Terrace, Staten Island	175	175	175	175	1
11	Baychester	107.424	98	40	300	33
12	Bayside	135.231	108	39	415	39
13	Bayswater	95.389	80	38	225	18
14	Bedford-Stuyvesant	123.060	100	10	500	1,943
15	Belle Harbor	221.286	225	92	350	7
16	Bellerose	136.200	107	50	300	15
17	Belmont	112.552	90	30	300	29
18	Bensonhurst	114.489	100	24	351	45
19	Bergen Beach	204.882	187	50	500	17
20	Boerum Hill	199.642	195	44	415	53
21	Borough Park	94.500	72.500	35	330	46
22	Breezy Point	150	150	150	150	4
23	Briarwood	140.111	82	37	499	36
24	Brighton Beach	134.375	119.500	50	385	48
25	Bronxdale	74.938	58.500	40	159	16
26	Brooklyn Heights	194.667	179	70	410	45
27	Brownsville	128.079	95	31	500	101
28	Bull's Head	124.833	101	60	200	6
29	Bushwick	109.358	85	18	500	1,008
30	Cambria Heights	124.159	106.500	43	350	44

3. Frequentist Analysis

3.1 Proposed Frequentist Model(s)

We will be using Multiple Linear Regression to analyze our dataset.

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ij} + \epsilon_i$$

- Y is the predicted rent price
- β_0 is the intercept
- β_1, \dots, β_p are the coefficients for the respective X_n independent variables
- ϵ_i is the error term

The variables that we included in our model were:

- *area*, the New York neighborhood of the Airbnb listing
- *accommodates*, the number of individuals the Airbnb can accommodate
- *beds*, the number of beds in the Airbnb
- *bathrooms*, the number bathrooms in the Airbnb
- *min nights*, the minimum number of nights required to book the Airbnb
- *room type*, whether the Airbnb listing is a private room, shared room, hotel room, or the entire home/apartment
- *rating*, the rating score out of 5

I decided to only use categorical predictor, area, instead of longitude and latitude because it can give the same information on the data and makes analysis easier than numerical longitude and latitude.

3.2 Fitting the Frequentist Model(s)

We start by fitting the following 2 models without any interaction terms.

The formula for the 2 models are as following:

- $Price = \beta_0 + \beta_1 \times area + \beta_2 \times accommodates + \beta_3 \times bathrooms + \beta_4 \times nights + \beta_5 \times room\ type + \beta_6 \times beds + \beta_7 \times rating$

- $\log(Price) = \beta_0 + \beta_1 \times area + \beta_2 \times accommodates + \beta_3 \times bathrooms + \beta_4 \times nights + \beta_5 \times room\ type + \beta_6 \times beds + \beta_7 \times rating$

We fit the model by first testing if the model needed a log transformation. Thus, we graphed area vs. Price and area vs. $\log(\text{Price})$. We can see from figure 4 and 5 that after performing a log transformation on Price, the data becomes more linear and variance becomes more constant. This is further demonstrated if we graph the fitted values against the residuals in figure 5 and 6.

```
lm24 = lm(price ~ as.factor(area) + accommodates + bathrooms + min_nights +
           as.factor(room_type) + beds + rating, data=airbnb24)
log24 = lm(log(price) ~ as.factor(area) + accommodates + bathrooms + min_nights +
           as.factor(room_type) + beds, data=airbnb24)
```

Figure 4: Area vs. Price

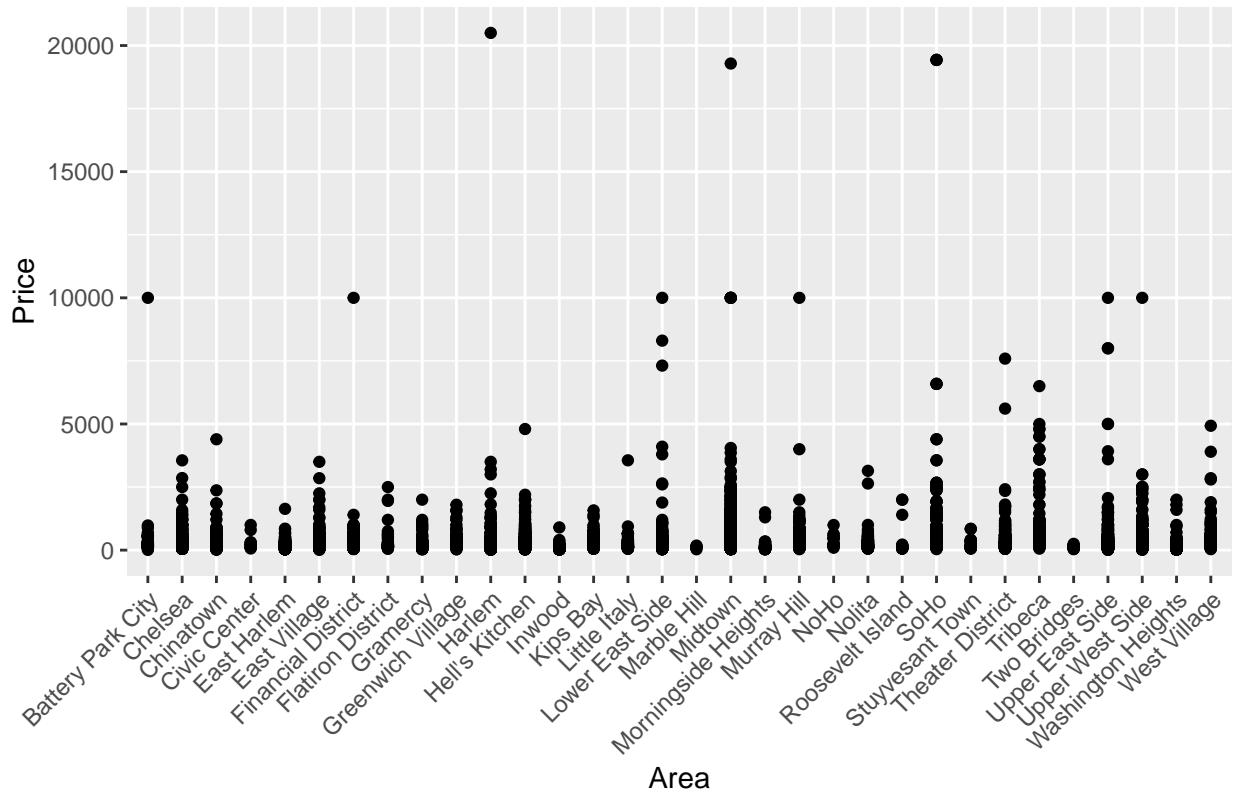


Figure 5: Area vs. Log(Price)

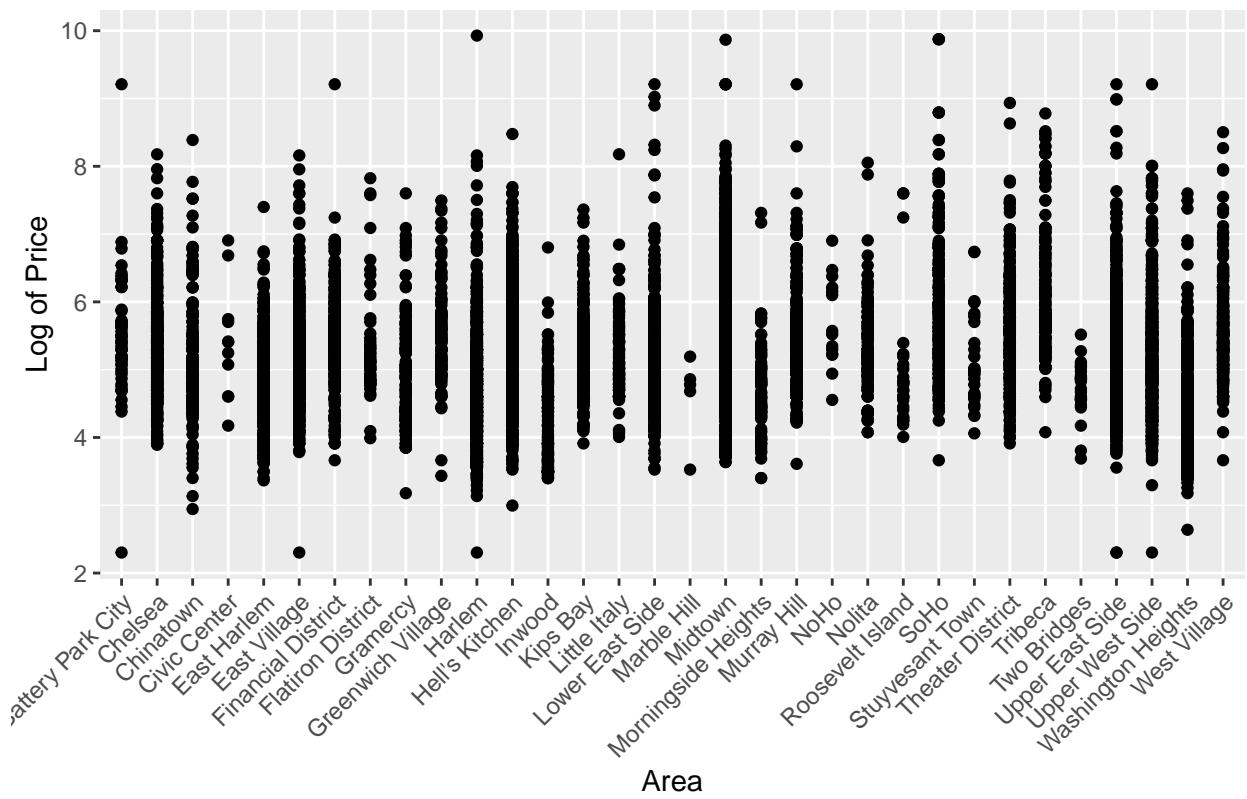


Figure 5: Without Log Transformation

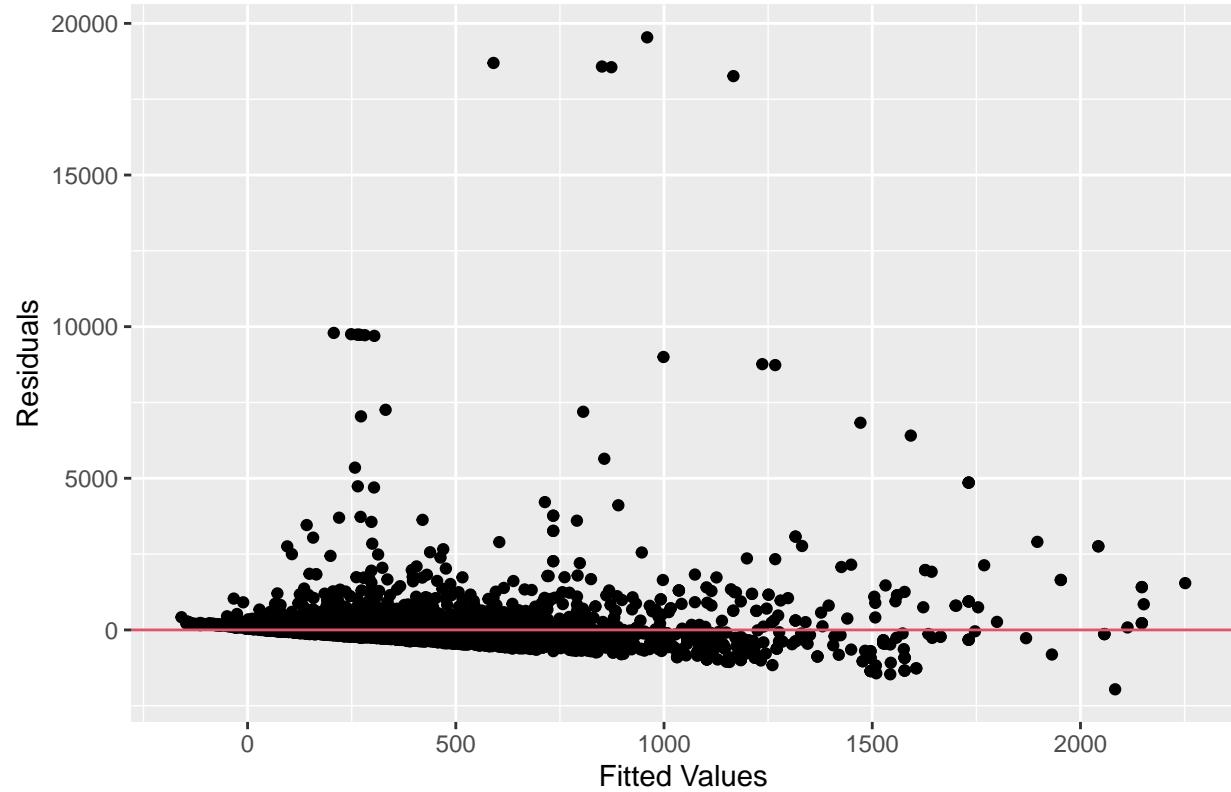
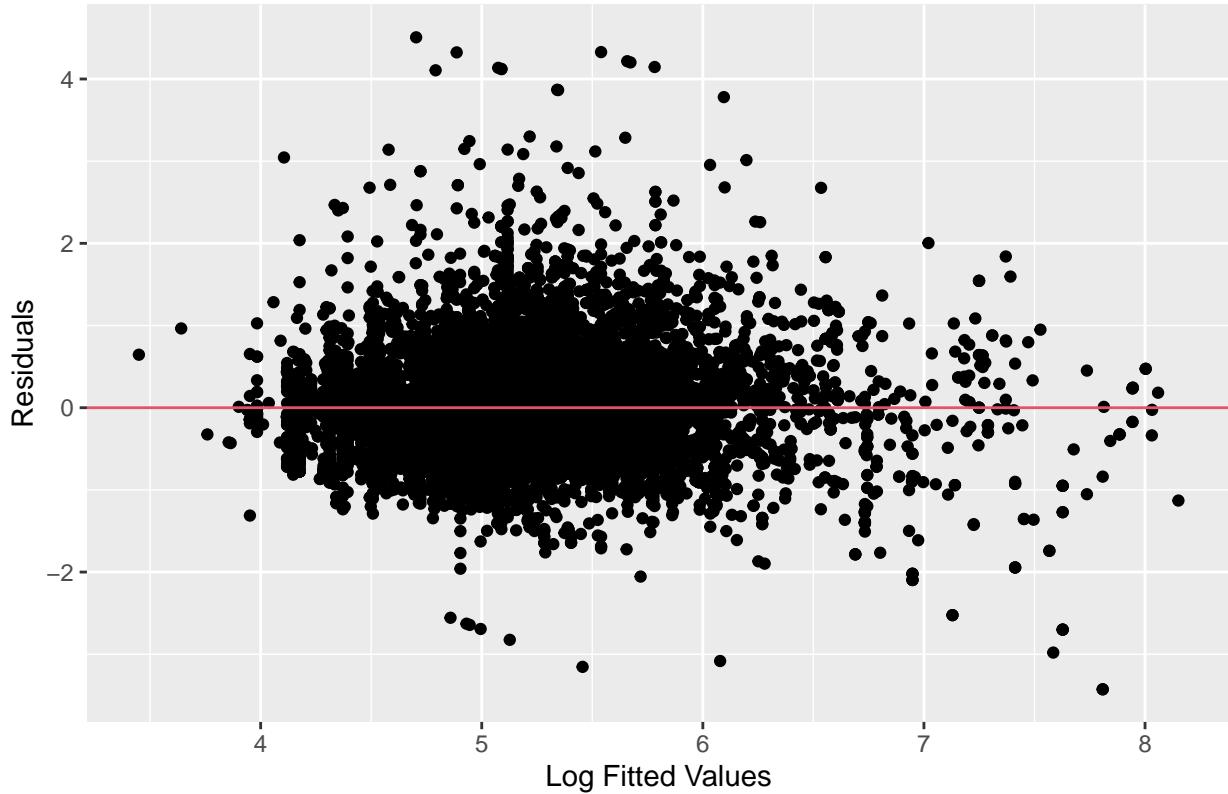


Figure 6: With Log Transformation



Thus, we decided to go with $\log(\text{Price})$ in fitting our final model.

Because there are two categorical variables in our dataset, we want to control for the interaction terms. To determine the best model, I manually use greedy backwards algorithm to find the most significant predictors between all order two interaction terms.

The final model formula with most significant number of predictors is the following:

$$\log(\text{Price}) = \beta_0 + \beta_1 \times \text{accommodates} + \beta_2 \times \text{bathrooms} + \beta_3 \times \text{min_nights} + \beta_4 \times \text{as.room_type} + \beta_4 \times \text{rating} + \beta_5 \times \text{as.factor(room_type)} * \text{bathrooms}$$

From figure 4 and 5 which contains the Anova table, and AIC/BIC values, we can see a general comparison between the models without a log transformation (lm24), the model with a log transformation (log24), and the final model we arrived at with a log transformation and interaction terms (lm1). From both figures and the adjusted r^2 value of 0.438 , the model that includes interaction terms and log transform performs the best.

```
lm1 = lm(log(price) ~ (accommodates) + (min_nights) + (rating) + (bathrooms)
+ rating + as.factor(room_type) + as.factor(area)*(bathrooms)
+ as.factor(room_type)*(accommodates), data=airbnb24)
```

Given that there are 32 neighborhoods in our sample, we will not display the final equation of the fitted model as there are 288 coefficients, but we will briefly explain our results below.

Table 4: Anova for 2 models

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	12, 209	4, 513.760				
2	12, 242	4, 723.330	-33	-209.569	17.177	0

Table 5: Figure 12: AIC/BIC for 3 Models

	Name	AIC	BIC
1	Log Interaction Model	22, 705.600	23, 247.000
2	Log Linear Model	23, 197.000	23, 493.600
3	Linear Model	191, 358.000	191, 662.000

In our final fitted model, the intercept $\beta_0 = 3.857386$, which means if the Airbnb could accommodate 0 people, had 0 beds and bathrooms, required 0 minimum nights to book, was the entire home or apartment, and was located in Battery Park, the estimated rent price of the Airbnb is $\sim \$50$. We also found that the rent price of the Airbnb increases when it can accommodate more people, has more bathrooms, or is a private room and decreases the more minimum nights are required to book, is a hotel room or shared room, or has more beds.

As for the neighborhoods, we found that, compared to Battery Park, the neighborhoods that increased the rent price of an Airbnb the greatest were Civic Center, Inwood, Morningside Heights, and Roosevelt Island, and the neighborhoods that decreased the rent price the greatest were Flatiron District, Gramercy, and Stuyvesant Town.

The mean predicated log price is around 5.2 with a standard deviation of 0.539074. In table 6, I also plot the first 30 neighborhoods' mean and standard deviation to get an idea of how spread out the price is.

4. Bayesian Analysis

4.1 Proposed Bayesian Model(s)

In the Bayesian context, we can use our Airbnb data in two ways. We can use a non-informative prior such as $1/\sigma^2$ that does follows $IG(0, 0)$. Alternatively, we can use this dataset as the prior and download the more recent 2024 as our data. For this proposal, we will use the reference prior, $1/\sigma^2$ due to the new data have not been published as the proposal due date.

We now formally define the notion for this section:

Table 6: Mean and SD by neighborhoods

	area	Log predicted_mean	Log predicted_sd
1	Battery Park City	5.502	0.510
2	Chelsea	5.332	0.372
3	Chinatown	5.101	0.466
4	Civic Center	5.416	0.736
5	East Harlem	4.786	0.342
6	East Village	5.181	0.382
7	Financial District	5.424	0.249
8	Flatiron District	5.355	0.621
9	Gramercy	4.936	0.340
10	Greenwich Village	5.523	0.474
11	Harlem	4.725	0.402
12	Hell's Kitchen	5.261	0.399
13	Inwood	4.245	0.400
14	Kips Bay	5.194	0.350
15	Little Italy	5.228	0.329
16	Lower East Side	5.159	0.420
17	Marble Hill	4.608	0.110
18	Midtown	5.502	0.379
19	Morningside Heights	4.699	0.287
20	Murray Hill	5.358	0.493
21	NoHo	5.707	0.508
22	Nolita	5.320	0.285
23	Roosevelt Island	4.971	0.241
24	SoHo	6.069	0.798
25	Stuyvesant Town	5.111	0.432
26	Theater District	5.575	0.429
27	Tribeca	6.173	0.604
28	Two Bridges	4.724	0.272
29	Upper East Side	5.070	0.429
30	Upper West Side	5.164	0.401

$$Y_{n \times 1} = X_{n \times p} \beta_{p \times 1} + \epsilon_{n \times 1}$$

- $X_{n \times p}$ = data matrix
- $Y_{n \times 1}$ = price response vector
- $\beta_{p \times 1}$ = predictor vector
- $\epsilon_{n \times 1} \sim N(0, \sigma^2 I_{p \times p})$ = noise vector

4.2 Fitting the Bayesian model(s)

Following the correlation heat map from exploratory data analysis, we see that some predictors are correlated. However, since the correlation are not over 0.5, we can assume that it will not have a significant impact and each predictors are independent.

To start the Bayesian regression analysis, we will center data first by subtracting the mean from each column.

```
means = colMeans(numerical_features_clean, na.rm = TRUE)
numerical_features_clean_subtracted <- numerical_features_clean
for (i in seq_along(df)) {
  numerical_features_clean_subtracted[, i] <- ifelse(is.na(numerical_features_clean[, i]),
```



```
stargazer(numerical_features_clean_subtracted, type = "latex", title = "Summary of Centered Data")
```

Table 7: Summary of Centered Data

Statistic	N	Mean	St. Dev.	Min	Max
latitude	25,604	0.000	0.060	-0.228	0.183
longitude	25,604	-73.940	0.060	-74.252	-73.714
accommodates	25,604	2.807	1.882	1	16
beds	24,947	1.648	1.079	1	14
minimum_nights	25,604	26.055	9.750	1	30
number_of_reviews	25,604	33.002	67.832	0	1,865
review_scores_rating	19,234	4.694	0.488	0.000	5.000

Since this a multiple linear regression. We can define the distribution as the following:

- $\pi(\beta | \sigma^2, X, Y) \sim N(\hat{\beta}, \sigma^2(X'X)^{-1})$ where $\hat{\beta} = (X'X)^{-1}X'Y$
- $\pi(\sigma^2 | X, Y) \sim IG((n - p)/2, SSE/2)$ where $SSE = (Y - X\hat{\beta})'(Y - X\hat{\beta})$

Alternatively, we can fit a linear hierarchical model to better compensate for neighborhood data. To do this, we would use the brms library to do the following.

Stage 2

$$\alpha | \beta, \Sigma_\alpha \sim \text{MVN}_2(\beta, \Sigma_\alpha)$$

Stage 3

- $\beta | \mu, \Sigma_0 \sim \text{MVN}_2(\mu, \Sigma_0)$
- $\Sigma_\alpha \sim \text{Inv-Wishart}(R, \rho)$
- $\sigma_e^2 \sim \text{IG}(a_e, b_e)$

We'll be able to get a better understanding of to do this after 12/14/24 lectures and learning how to use the brms after the lab06.

To perform sensitivity test, we will make sure the following conditions are satisfied.

- $n > p$
- Making sure the predictors are independent
- $X'X$ is convertible.
- Checking collinearity
- $SSE > 0$

Another way that we can check for sensitivity is using data from prior years to do cross validation to make sure our model is correct.

For ensuring MCMC convergence, we will plot and monitor the following showing signs of convergence:

- trace plots
- histograms
- Gelman-Rubin convergence diagnosis
- Effective sample size

4.3 Prediction

After generating the posterior distribution, we can generate the predictive posterior distribution, and compute the predictive mean and credible intervals.

5. Discussion

We can improve our model by narrowing down the neighborhoods where we do not have enough information in order to make reliable predictions and remove it from our model. While this means that we will have to exclude predictions for that area, that could potentially help in obtaining a better fitted model. Another way we can improve our model is to use a informative prior derived from previous years' dataset. However, this might be difficult because of other variable impacting the Airbnb market such as the economy.

6. Contributions

For the first proposal, Bridgette and Jake split evenly. Bridgette did sections 3,4, and 5. Jake did selection 1 and 2. For the final proposal, Jake did all the revision. We have spent 15+ hours for this proposal

References

Appendix

From Section 2:

```
summary_price <- as.data.frame(summary_price)

data <- read.csv('nydata.csv')

numerical_features <- data[3:9]
categorical_features <- data[10:11]
price <- data$price
summary(price)
summary(numerical_features)

features <- c("latitude", "longitude", "accommodates", "beds", "minimum_nights", "number

for (i in 1:ncol(numerical_features)) {
  p <- ggplot(numerical_features, aes(x = numerical_features[[i]])) +
    geom_density(fill = "skyblue", color = "black") +
    labs(title = paste("Density Plot of Feature", features[i]), x = features[i], y = "De
    theme_minimal()

  plot(p) # Print each plot
}
```

```

#clean data
data_clean = subset(data, price <= 500 & minimum_nights <= 30)
numerical_features_clean <- data_clean[3:9]
categorical_features_clean <- data_clean[10:11]
price_clean <- data_clean$price

summary(price_clean)
summary(numerical_features_clean)

#graph again
for (i in 1:ncol(numerical_features_clean)) {
  p <- ggplot(numerical_features_clean, aes(x = numerical_features_clean[[i]])) +
    geom_density(fill = "skyblue", color = "black") +
    labs(title = paste("Density Plot of Feature", features[i]), x = features[i], y = "De
    theme_minimal()

  print(p) # Print each plot
}

plot(price_clean)

pairs(numerical_features_clean)

cor_matrix <- cor(numerical_features_clean)
cor_matrix
ggcorrplot(cor_matrix)

#graphs with the categorical features
ggplot(categorical_features_clean, aes(x = categorical_features_clean$room_type)) +
  geom_bar(stat = "count", fill = "skyblue", color = "black") +
  labs(title = "Count Plot", x = "Category", y = "Count") +
  theme_minimal()

neighborhood_Price <- data_clean[11:12]
summary_price <- neighborhood_Price %>%
  group_by(neighbourhood_cleansed) %>%
  summarise(mean_price = mean(price),
            median_price = median(price),
            min_price = min(price),
            max_price = max(price),
            total_listings = n())

summary_price

```

From Section 3.2

```
airbnb24 <- read.csv('ny_ny.24.csv')
airbnb24$logP <- log(airbnb24$price)

lm24 = lm(price ~ area + accommodates + bathrooms + min_nights + room_type + beds, data=airbnb24)
log24 = lm(logP ~ area + accommodates + bathrooms + min_nights + room_type + beds, data=airbnb24)

lm1 = lm(logP ~ area*log(accommodates) + area*bathrooms + area*log(min_nights) + area*room_type)

## Residuals vs. Y
ggplot(airbnb24, aes(x=area, y=price)) + geom_point() +
  xlab("Area") + ylab("Price") + labs(title="Area vs. Price")

ggplot(airbnb24, aes(x=area, y=logP)) + geom_point() +
  xlab("Area") + ylab("Log of Price") + labs(title="Area vs. Log(Price)")

## Residuals vs. fitted values
ggplot(airbnb24, aes(x=lm24$fit, y=lm24$res)) + geom_point() +
  xlab("Fitted Values") + ylab("Residuals") + labs(title = "Without Log Transformation") +
  geom_hline(yintercept = 0, col=2)

ggplot(airbnb24, aes(x=log24$fit, y=log24$res)) + geom_point() +
  xlab("Log Fitted Values") + ylab("Residuals") + labs(title = "With Log Transformation") +
  geom_hline(yintercept = 0, col=2)

summary(lm1)$r.squared
summary(log24)$r.squared
summary(lm24)$r.squared

coef(lm1)[1] #log(price)
tidy(lm1)
tidy(lmUES)
```