AIRBNB PROPERTY PRICE PREDICTOR

GROUP 9

BY: Akshita Chawdhary, Bhumika Khandelwal, Prithvi Shetty, Ryan Han, Aakash Agrawal

INFX 573

1. ABSTRACT

Unlike established hospitality industry where dedicated management team controls how a property is maintained, Airbnb provides a listing of rental properties posted by individual users without any formal control. Since there are no formal controls presented in the Airbnb listed properties, it is not easy to compare the quality of listings by looking at the user ratings alone. Description and geo-location information of the listings can give us a lot of information about the price of a listing.

2. INTRODUCTION

Airbnb is a community marketplace where guests can book living accommodations from a list of verified hosts. Membership to the site is completely free and there is no cost to post a listing. Using a targeted user interface designed to narrow down traveling preferences, Airbnb offers an attractive, cost-saving alternative to traditional hotel bookings and vacation home rentals.

Upon finding a desired listing, guests are prompted to sign up for membership, which provides access to contact the host directly as well as provide payment information for a request. Only once the host accepts the transaction and the guest checks in is the credit card charged, along with a 6-12% transaction fee from Airbnb.

The process is similarly simple for hosts, who get a notification once a guest indicates interest in a listing and have the option to approve or deny the transaction. Once the listing is booked, the host receives the payment and Airbnb takes a 3% transaction fee.

However, unlike the established hospitality industry where dedicated management team controls how a property is maintained, Airbnb only provides a listing of rental properties posted by individual users without any formal control. Current establishment of hospitality industry relies on the five-star rating system that provides certain measurement of qualities of property that determines pricing point. This allows individual users to gauge the overall quality of a property by looking at the associated star rating, thus users can make an informed decision.

Since there are no formal controls presented in the Airbnb listed properties, it is extremely hard to predict the quality of a rental when an individual user books for a stay.
Because of this quality uncertainty, our team decided to find a better predictor for the end-users looking for a rental property. Once determined, this predictor can also possibly help listing individuals how to make their properties more appealing by utilizing such predictor.

Our research questions were:

1) Is the price of a rental influenced by the frequency of words used in the amenities field of the listing?
2) Will the price of a rental go higher if the average of user ratings gets higher?
3) Will the price of a rental go higher if the average of host response time improves?
4) If a rental property is closer to a popular destination, is it likely to be priced higher than properties further away?

1.1) Datasets

We selected our dataset from public.opendatasoft.com, particularly for the city of Melbourne and Los Angeles. The cities have a similar population of 3.9 million and are popular tourist destinations. Hence we decided to analyze these cities. Although there are a lot of variables (89) in the dataset, we used some of them for our analysis:

Summary: Summary of the listing given by the host

Description: Description of the listing

City: Details of the city the listing is in

Latitude and Longitude: Coordinates of the property

Accommodates: The number of people the listing accommodates

Amenities: List of the amenities available at the listing

Price: Price of the listing

Review Score Rating: Review score rating given to the listing

Geolocation: Geolocation of the listing

Response rate: The rate of how quickly the host responds to the customers

Derived Fields

USGeo: Universal Studios GPS coordinates (Quantitative)

downtownGeo: Downtown LA GPS coordinates (Quantitative)

SMGeo: Santa Monica GPS coordinates (Quantitative)

USDist: Distance to Universal Studios in mile (Quantitative)

DTDist: Distance to Universal Studios in mile (Quantitative)

SMDist: Distance to Santa Monica in mile (Quantitative)

Walk: Within walking distance (Qualitative)

2) METHODS

After the examination of some of listing datasets from the Airbnb.com website, it became apparent that most listing utilized description field of the property to entice users. Some properties focused on the location aspect while others focused on the aesthetic aspect. With this information, we decided to look at the free form text descriptor to find the relationship between the rating and types of words used. By categorizing different descriptors into several groups such as positive adjectives focusing on aesthetic, emphasis on the geolocation, and amenities offered, we hoped to find which category is a better indicator of the overall rating.

Our team also used the geo-location information to see if there is any difference in the rental pricing, taking distance to a well-known tourist destination, as a factor. We also checked if there is a positive correlation between how far a property is to a well-known tourist destination. This was done with an objective that it could help property owners come up with a better strategy to make their property more appealing, so they can compete better with properties closer to such destinations.

2.1) Data Cleaning

```
#Find per person price (total price / max number of people)
LAabnb$PersonPrice<- with(LAabnb, Price / Accommodates)

#Find per room price (total price / number of room)
price<-LAabnb$Price
bedrooms<-LAabnb$Bedrooms
bedrooms[bedrooms<0.1] <- 1
nprice<-price/bedrooms
LAabnb$RoomPrice<-nprice
```

```
#Create a dataframe USLST that only contains listings of rentals that
```

```
is within 5 miles to Universla Studios.
US <- LAabnb[which(LAabnb$USDist < 5),]

#Create a dataframe DTLST that only
contains listings of rentals that is within 5
miles to downtown LA.
DT <- LAabnb[which(LAabnb$DTDist < 5),]

#Create a dataframe DTLST that only
contains listings of rentals that is within 5
miles to Santa Monica.
SM <- LAabnb[which(LAabnb$SMDist <
5),]
```

Some listing lacked pricing information. This introduced 'na' in columns from 90 to 94. Of listing datasets, only listings with pricing information were added to each destination dataset.

```
#Remove Null from USLST, DTLST,
SMLST in colume 90 ~ 94
USLST<-US[complete.cases(US[, 90:94]),
]
SMLST<-SM[complete.cases(SM[, 90:94]),
]
DTLST<-DT[complete.cases(DT[, 90:94]),
]
```

2.2) Procedure

Procedures that we followed:

1. Exploratory Analysis
2. Text Analytics
3. Linear Regression
4. Logistic Regression

We started with exploring the relationship between the ratings and the price of the listing in general

We used text analytics to see which word association is prominent

We used linear regression to see the relationship between the features ratings vs price

We used linear regression to see the relationship between the features response rate vs price

We used linear regression to see the relationship between the features price vs distance

Log-Odds - Logistic Regression
We applied logistic regression on the feature walkable distance. For this, we created a column 'walk' and assigned the value 1 if the walking distance of the listing from the popular destinations was within 1.5 miles and assigned the value 0 if the distance was further than 1.5 miles against the three different pricing scheme.
i.e. Price per Listing, Price per Room and Price per Person.

3) ANALYSIS AND RESULTS

1. Text Analytics
2. Amenities vs Price
3. Review rates vs Price
4. Walking Distance vs Price

3.1) TEXT ANALYTICS

We performed the free-form text analysis on the listings on the "Description" feature of both Melbourne and Los Angeles dataset. From this exploratory analysis, we tried to find some patterns in the descriptions of the listings of both the cities.

We first converted the description value of all the listings (2097 listings) into a large corpus and then cleaned and transformed the text for processing. This process reduced the size of the corpus and removed the

unwanted/not useful words from our analysis. After doing the basic text cleaning and transformations we found some unusual patterns of words which were created during the text stemming and white space removal process. We mined these patterns and transformed them to the nearest meaningful root.

Next, we created a document term matrix from the corpus where each row of the matrix was a document vector, with one column for every term in the entire corpus. This document term matrix was converted to a matrix of words with their respective frequencies. We sorted the matrix in descending order of the frequencies of the words and created a data frame of the same for further analysis.

	words	freq
bed	bed	4275
locat	locat	2250
citi	citi	1834
free	free	1833
tram	tram	1667
walk	walk	1578
room	room	1460
cbd	cbd	1382
station	station	1361
kitchen	kitchen	1294

Figure 1: Frequencies of the words from Description column in Melbourne data set

We wrote the new data frame created into a CSV file named “frequency.csv” for Melbourne and “frequencyLa.csv” for Los Angeles, which helped us in creating the customized categories for Aesthetics and Amenities(Figure 1 and 2).

	words	freq
bed	bed	28160
room	room	19623
walk	walk	16377
locat	locat	15377
hollywood	hollywood	15372
park	park	14627
bath	bath	14549
kitchen	kitchen	12797
privat	privat	12747
hous	hous	11337

Figure 2: Frequencies of the words from Description column in LA data set

Figure 3. and Figure 4. gives us a good information about the listing descriptors and helps us in picking words for our customized categories. It also highlights some words which could further be removed from our analysis such as Melbourne, apart, etc.

Figure 3.1: Word Cloud for Melbourne data set

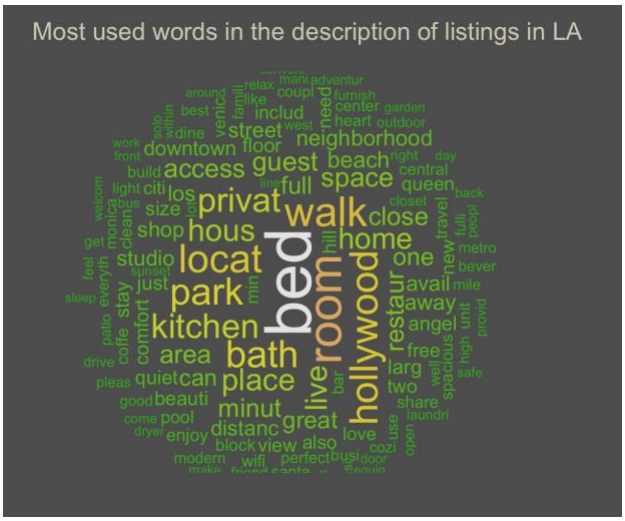


Figure 3.2: Word Cloud for LA data set

Further, to find interesting patterns among the words, we thought of checking the correlations among the frequently occurring words. For that, we ran associations on some of the high frequency words. We categorized the most prominent words like bed, room into “Amenities” and words like train and walk into “Distance”. Figures 5 & 6 show the correlations of words of these categories.

```
$breakfast
   friday    dinner    recept    sunday   tuesday
     0.60      0.57      0.57      0.56      0.56
discoveri therapist      vend   massag    refund
     0.53      0.53      0.53      0.52      0.52
   cinema      note  backpack     price      rate
     0.48      0.47      0.46      0.45      0.45
     book continent      larg     small  crossley
     0.33      0.32      0.31      0.31      0.30
```

Figure 5: Correlation for Amenities category

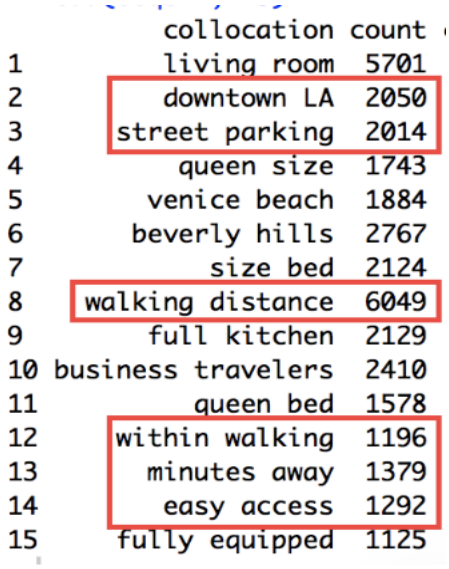
```
$parking
          315 fantastically     commercial
         0.53          0.41           0.41
          car           lot         behind
         0.35          0.33           0.32
```

Figure 6: Correlation for Distance category

To check how different set of words appeared in the two listings we created bigrams and trigrams of the words from the description. Figures 7 & 8

	collocation	count
1	free tram	468
2	melbourne CBD	481
3	cross station	610
4	living room	333
5	melbourne central	435
6	fully equipped	370
7	living area	291
8	fully furnished	253
9	queen size	266
10	victoria market	342
11	queen victoria	288
12	bedroom apartment	418
13	business travellers	267
14	sofa bed	363
15	size bed	257

Figure 7: Bigram for LA data set



	collocation	count
1	living room	5701
2	downtown LA	2050
3	street parking	2014
4	queen size	1743
5	venice beach	1884
6	beverly hills	2767
7	size bed	2124
8	walking distance	6049
9	full kitchen	2129
10	business travelers	2410
11	queen bed	1578
12	within walking	1196
13	minutes away	1379
14	easy access	1292
15	fully equipped	1125

Figure 8: Bigram for Melbourne data set

Results from bigrams:

Bigram for LA: Words like downtown, street parking, walking distance, etc (Grouped into the category - Distance)

Bigram for Melbourne: Words like living area, queen size, sofa size, etc. (Grouped into the category - Amenities)

3.2) AMENITIES VS PRICE

The word cloud and the bigram gives the picture of different things which owners and customers give preference to. Therefore, in order to analyze, how these words can be better related to the data set, the next approach was to understand how frequently these words appear in the for both the cities by focusing majorly on the words under two categories:

1. Amenities (prominent in Melbourne)
2. Distance (prominent in LA)

The first step for this was to find how frequently the different amenities are provided by the owner, in order to analyze if the presence of any amenities affects the price of the listings.

We created a data frame with the frequencies of all the amenities that every listing provides.

Table 1. below lists the different amenities in order of how frequently they are being provided by the owners of different listings.

Frequency	Melbourne	LA
Highly	Kitchen Internet Washer	Internet Kitchen Heating
Moderately	Gym Hair Dryer Shampoo Breakfast	Parking Gym Hair Dryer Shampoo
Low	Bed	Patio

Table 1: Frequencies of provided amenities

From the above table, we can say that in LA most of the listings provide amenities such as wireless internet, kitchen, heating, etc and in Melbourne most of the listings provide kitchen, wireless internet, washer, etc. To analyze the effects of amenities on the price of the room, we decided to pick the top keywords from the moderately listed amenities.

From the list, breakfast, gym and free parking were selected as the focus of analyses and compared the prices individually with the mean price for both the cities.

Mean price of listings in Melbourne: $136.19
Mean price of listings in LA: $138.56

a) Effect on Price by providing Breakfast and Gym

Table 2. shows the percentage of the listings providing these two amenities:

	Melbourne	LA
No. of listings	2097	19426
Breakfast	14%	12%
Gym	52%	15%

Table 2: Frequencies of provided amenities

Steps of analyses:

1. Created a new column in the data set and substituted value 1 against the listings providing breakfast, 2 against the listings providing gym facility and 0 against the ones who do not provide both of these
2. Plotted a box-plot for the frequencies of these amenities against the price of the listings

Box plot for Melbourne listing (Figure 9):

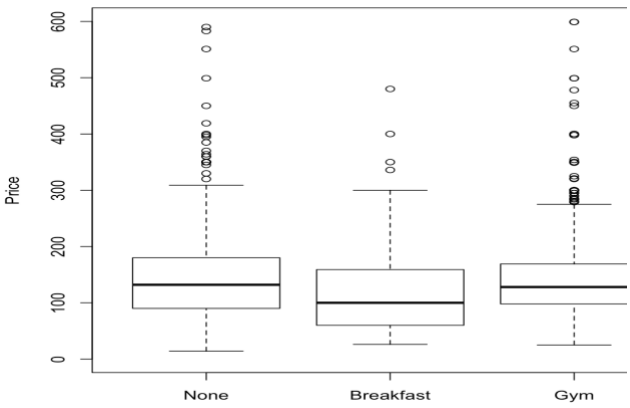


Figure 9: Box plot for Amenities Breakfast vs Gym for Melbourne data set

Anal

Breakfast

1. Minimum Price - Lower than the mean price as well lower than the price of the amenities not providing any of these amenities.
2. Highest Price - Lower than the one providing Gym as one of the facility as well as lower than the ones not providing any of these
3. Mean Price - Lower than both the other factors

Deduction: Not a significant factor affecting the high price

Gym

1. Minimum Price - Higher than the mean price of the listings providing Breakfast and also higher than the lowest price of the listings not providing any of these
2. Highest Price - Lower than the ones not providing both these amenities, but higher than the ones providing Breakfast
3. Mean Price - Higher than the mean price for both the other factors

Deduction: Gym can be considered as a factor affecting the high price of the listings.

Box plot for LA listings (Figure 10):

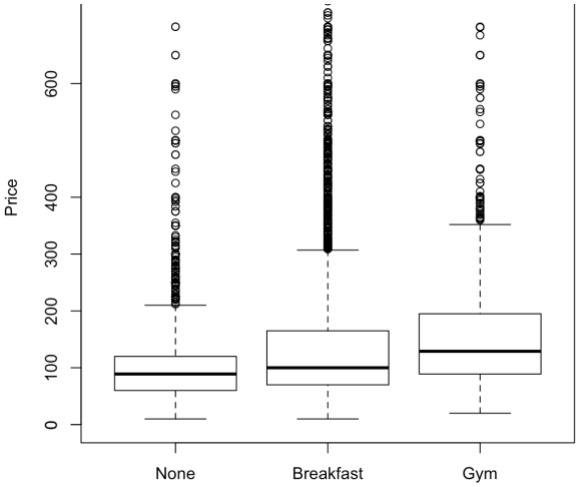


Figure 10: Box plot for Amenities Breakfast vs Gym for LA data set

A

Breakfast

1. Lowest Price - Higher than the listings not providing any of the two amenities but lower than the listings providing Gym as the facility
2. Highest Price - Higher than the listings not providing any of the two amenities but lower than the listings providing Gym as the facility
3. Mean Price - Higher than the listings not providing any of the two amenities but lower than the listings providing Gym as the facility

Deduction: Can be considered as the factor increasing the pricing

Gym

1. Lowest Price - Higher than both the other factors
2. Highest Price - Highest amongst the two factors
3. Mean Price - Highest amongst the two factors

Deduction: Can be considered as the factor increasing the pricing

In order to have a better understanding of the relation between providing different amenities and the price, the next idea was to add one more amenity.

b) Effect on price by providing Free Parking

Table 3, below shows the percentage of listings providing the three amenities: Breakfast, Gym and Free parking

	Melbourne	LA
No. of listings	2097	19426
Breakfast	14%	12%
Gym	52%	15%
Free Parking	60%	19%

Table 3: Frequencies of provided amenities

Steps of analyses:

1. Substituted value 3 in the column against all the listings providing free parking as one of the amenities
2. Plotted a box-plot for the frequencies of these amenities against the price of the listings

Box plot for Melbourne listings (Figure 11):

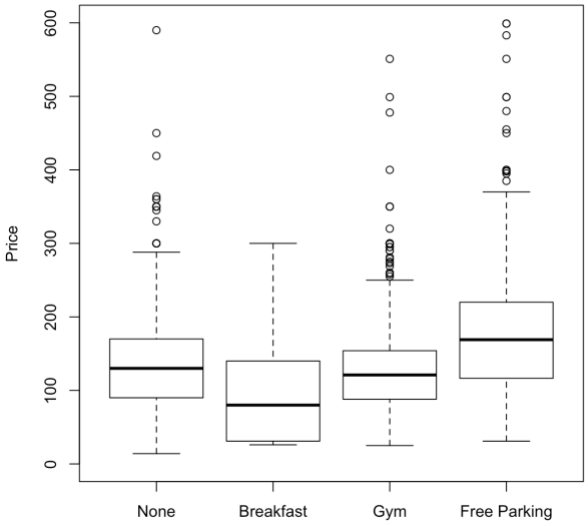


Figure 11: Box plot for Amenities Breakfast vs Gym vs Free Parking for Melbourne data set

Analyses:

Free Parking

1. Lowest Price - Marginally higher than listings providing gym facility but have a significant difference from the listings providing breakfast as the amenities
2. Highest Price - Highest amongst the listings providing breakfast and gym along with the listings not providing any of the three amenities
3. Mean Price - Significantly higher than the other three factors

Deduction: Free parking can be one of the significant reasons for the high price of the listing

Box plot for LA listings (Figure 12):

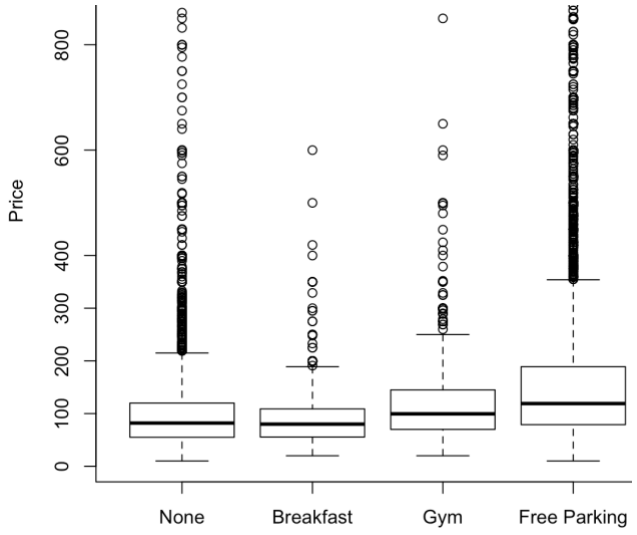


Figure 12: Box plot for Amenities Breakfast vs Gym vs Free Parking for LA data set

Free Parking

1. Lowest Price - Higher than the listings providing breakfast and not providing any of the amenities, but marginally higher than the listings providing gym
2. Highest Price - Highest amongst the listings providing breakfast and gym along with the listings not providing any of the three amenities
3. Mean Price - Marginally higher than the listings providing gym

Deduction: Free parking can be one of the significant reasons for the high price of the listing.

For reference below are the tables (Tables 4 & 5) listing the pricing under three categories for both the cities:

Price	Breakfast	Gym	Free Parking
Lowest	$30	$75	$115
Mean	$80	$120	$160
Highest	$135	$140	$225

Table 4: Approximate prices for Melbourne Listings

Price	Breakfast	Gym	Free Parking
Lowest	$ 50	$90	$93
Mean	$82	$102	$125
Highest	$115	$145	$210

Table 5: Approximate prices for LA Listings

3.3) REVIEW RATINGS VS PRICE

3.3.1) REVIEW RATINGS VS PRICE OF THE ROOM

We started with exploring the relationship between the ratings and the price of the listing in Los Angeles, in general. We saw that the review ratings of most of the listings vary between $80-100 and the price of these listing vary between $0-250.

This tells us that most people prefer a listing that is moderately priced and the listings receive a high rating as well. The ratings scale is set from a score of 0 to 100.

The price of a room can be different for different sizes of rooms. We cannot compare the prices of different types of rooms with the ratings as it can introduce a bias due to price difference. We decided to eliminate this bias by normalizing the price of the room. To do this, we brought the prices of rooms to the same range by scaling them down accordingly to the number of bedrooms.

Code for normalizing the price of the room:

```
bedrooms=LA$Bedrooms

if (bedrooms>0) {

  nprice=price/bedrooms

} else {

  nprice=price}
```

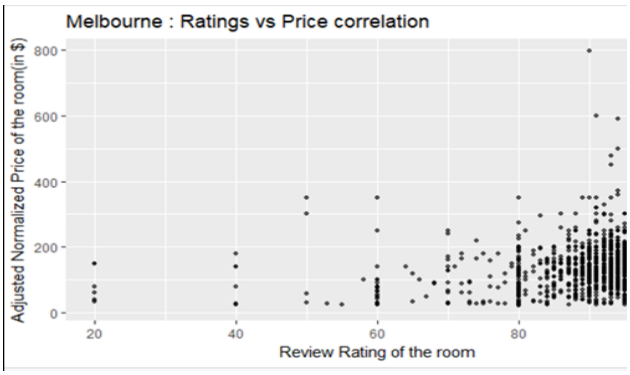


Figure 13: Review ratings vs Price of the room correlation in Melbourne data set

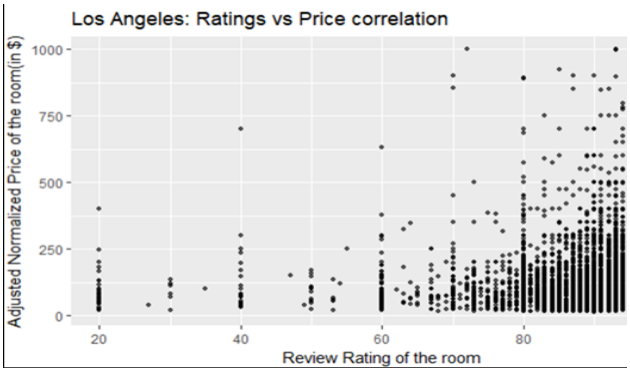


Figure 14: Review ratings vs Price of the room correlation in LA data set

Comparison between the results obtained in Los Angeles and Melbourne datasets (Figures 13 & 14):

Los Angeles	Melbourne
The slope between price and review rating is 1.1857.	The slope between price and review rating is 1.1714.
Increase in the value of review ratings by 1 unit leads to an increase in price by 1.1857 units. This indicates a medium positive correlation between price and the review ratings	Increase in the value of review ratings by 1 unit leads to an increase in price by 1.1714 units. This indicates a medium positive correlation between price and the review ratings.
The y intercept is 21.8156 When the value of review rating is 0, the price is 21.8156 units which does not make any sense.	The y intercept is 30.5249 When the value of review rating is 0, the price is 30.5249 units which does not make any sense.
Even though the Pearson correlation is good, the R squared value is quite less which signifies that a linear model cannot be fit.	Even though the Pearson correlation is good, the R squared value is quite less which signifies that a linear model cannot be fit.

We then compared the review rating of the room with the normalized price of the room. The slope between review ratings and price is 1.1857 which indicates a positive correlation and is evident from the plot.

Deduction: The price of the room is seen to increase as the review rating of that room increases.

3.3.2) HOST RESPONSE RATING VS PRICE OF THE ROOM

Host response rating is the ratio of total number of queries answered by the host to the total number of queries addressed to the host. We performed an exploratory data analysis by plotting this host response rate against the normalized prices of the rooms.

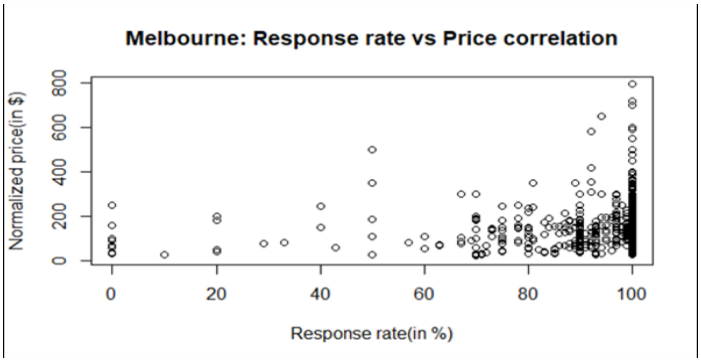


Figure 15: Host response vs Price of the room correlation in Melbourne data set

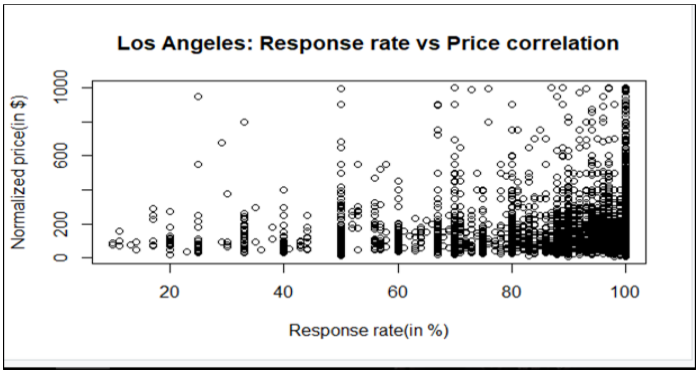


Figure 16: Host response vs Price of the room correlation in LA data set

Comparison between the results obtained in Los Angeles and Melbourne datasets (Figures 15 & 16):

Los Angeles	Melbourne
The slope between host response rate and review ratings is 0.436 Increase in the value of response rate by 1 unit leads to an increase in price by 0.436 units. This indicates a low positive correlation between price and the host response rate.	The slope between price and review rating is 0.6426. Increase in the value of response ratings by 1 unit leads to an increase in price by 0.6426 units. This indicates a low positive correlation between price and the host response rate.
The y intercept is 141.8156 When the value of review rating is 0, the price is 141.8156 units which does not make any sense.	The y intercept is 77.5249 When the value of review rating is 0, the price is 77.5249 units which does not make any sense.
Even though the Pearson correlation(0.2) is not good, the R squared value is quite less (0.04) which signifies that a linear model cannot be fit.	Even though the Pearson correlation(0.1) is not good, the R squared value(0.01) is quite less which signifies that a linear model cannot be fit.

Deduction: There is no sufficient evidence to show that host response rating affects the price of the room.

3.3) DISTANCE VS PRICE

From the linear regression model as mentioned below between Price, USDist, SMDist, and DTDist variables using LA dataset:

Model: lm(Price ~ USDist+SMDist+DTDist, data=LAabnb)

The following results were observed(Tables 6 & 7):

Residuals	
Min	-146.55
1Q	-66.68
Median	-33.38
3Q	23.51

Table 6: Residuals for Price vs Distance model

Coefficients				
	Estimate	Std.Error	t value	Pr(>|t|)
Intercept	172.4686	3.3842	50.963	< 2e-16 ***
USDist	-1.945	0.2667	7.292	3.18e-13 ***
SMDist	-3.433	0.2089	-16.435	< 2e-16 ***
DTDist	1.6218	0.2042	7.943	2.08e-15 ***

Table 7: Coefficients for Price vs Distance model

With adjusted R-squared value of 0.023, it is highly unlikely that this model explains our data well enough. The second regression test was performed. This time, our team used PersonPrice instead of Price (Tables 8 & 9).

Model: lm(PersonPrice ~ USDist+SMDist+DTDist, data=LAabnb)

Residuals	
Min	-49.32
1Q	-17.06
Median	-6.31
3Q	8.11
Max	794.08

Table 8: Residuals for PersonPrice vs Distance model

Coefficients				
	Estimate	Std.Error	t value	Pr(>|t|)
Intercept	57.01544	0.91297	62.451	< 2e-16 ***
USDist	-0.4263	0.07196	-5.924	3.19e-09 ***
SMDist	-1.00348	0.05635	-17.808	< 2e-16 ***
DTDist	0.2202	0.05508	3.998	6.42e-05 ***

Table 9: Coefficients for PersonPrice vs Distance model

From the result, we observed similarly low adjusted R-squared value. With both total price and normalized price per person fail to explain data using linear model, our team performed the last regression test using per room pricing (Tables 10 & 11).

Model: lm(RoomPrice ~ USDist+SMDist+DTDist, data=LAabnb)

Residuals	
Min	-97.16
1Q	-34.04
Median	-10.32
3Q	19.05
Max	908.3

Table 10: Residuals for RoomPrice vs Distance model

Coefficients				
	Estimate	Std.Error	t value	Pr(>|t|)
Intercept	125.4718	1.6979	73.9	< 2e-16 ***
USDist	-0.5829	0.1339	-4.355	1.34e-05 ***
SMDist	-2.0147	0.1048	-19.221	< 2e-16 ***
DTDist	-0.3143	0.1024	-3.068	0.00216 **

Table 11: Coefficients for RoomPrice vs Distance model

Looking at the result, it became apparent that simple linear regression between any types of pricing scheme and distance to popular destinations using all data will not produce good model that can explain observed data well.

Our exploratory data analysis didn't produce meaningful result when we ran linear regression. After the discussion, our team identified a possible bias that may have influenced the result. Because LA contains many popular destinations, the effect of distance on the price could be significantly

diminished if the distance was over certain threshold.
Any customer wanting to go to Santa Monica will not rent a property that is 40 miles away. Conversely, anyone visiting Universal Studio will not likely to be booking a rental property outside of 10 mi radius. Due to this realization, new subsets were created. Our team decided to focus on any property that is within 5 miles to minimize the distance bias (Figures 17, 18 & 19).

Model: lm(Price~USDist, data=USLST)

After creating new datasets focused on each destination, our team ran linear regression to see if there is any difference. When obtaining linear regression model, our team only focused on the pricing scheme with the relevant distance to the focused destination. Since our new datasets were specifically created for each destination only, distance information to other destinations will not explain pricing model for the focused area. From the linear regression result using each destination datasets against different pricing scheme, we still obtained very low R-squared value of 0.01 ~ 0.04.

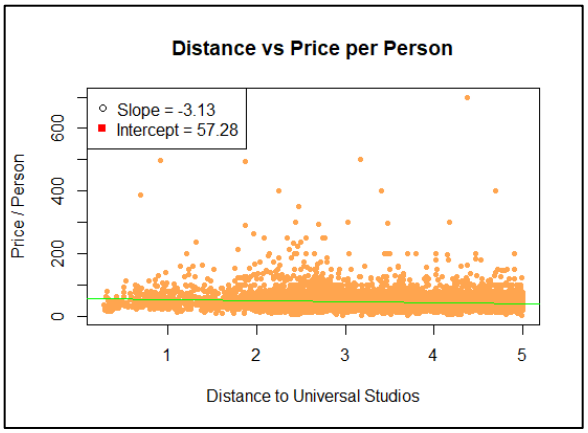


Figure 17: Distance vs Price per Person for Universal Studios

Distance vs. Price / Person

Distance = 0 → It will cost $57.28 per person
Distance increases by 1 mile → Cost decreases by $3

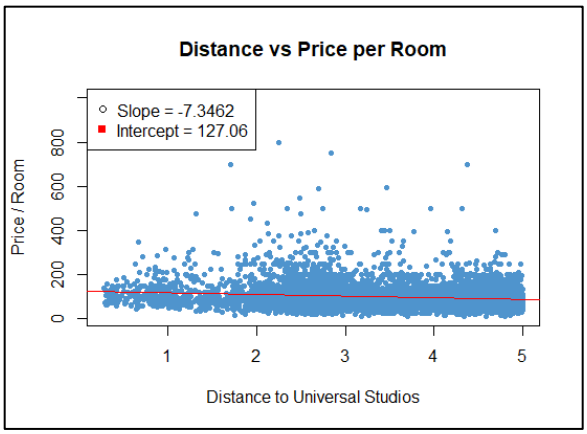


Figure 18: Distance vs Price per Room for Universal Studios

Distance vs. Price / Room

Distance = 0 → It will cost $127.06 per room
Distance increases by 1 mile → Cost decreases by $7

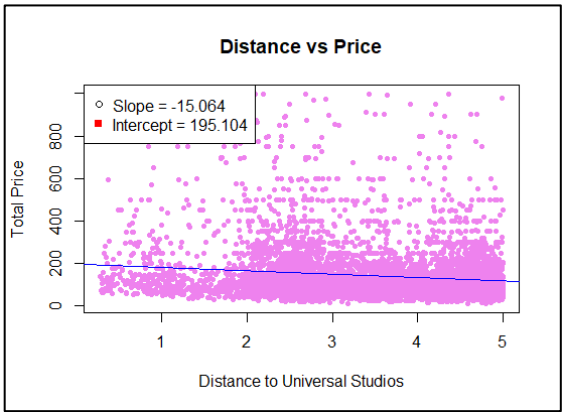


Figure 19: Distance vs Price for Universal Studios

Distance vs. Total Price

Distance = 0 → It will cost $195.10 per rental
Distance increases by 1 mile → Cost decreases by $15

3.4) LOGISTIC REGRESSION

After using linear regression model and getting inconsistent result, we reviewed our data sets. Maybe we were asking the wrong question. Perhaps, it is not the distance that predicts the pricing, maybe it is the other way around. To test this, we decided to create a logistic model that test the following:

When the price of a rental increases, it is likely to be within walking distance of the closest popular destination?

We randomly selected 1.5 mile as our walkable distance. We updated our dataset with 'walk' column with value 0 or 1 indicating within walkable distance.

We created a logistic model using walk, RoomPrice, PersonPrice, and Price. Afterward, fitted value was calculated and plotted to see what it looks like. However, each plot produced points only showing on the lower than 0.5.

With all data points less than 0.5, it indicated that the distance of 1.5 mi was too low to establish the relationship between walkable distance and the pricing.

Using our regression model, we predicted distance using the price. Also, the receiver operating characteristic curve was plotted and the AUC was calculated (Figures 20, 21 & 22).

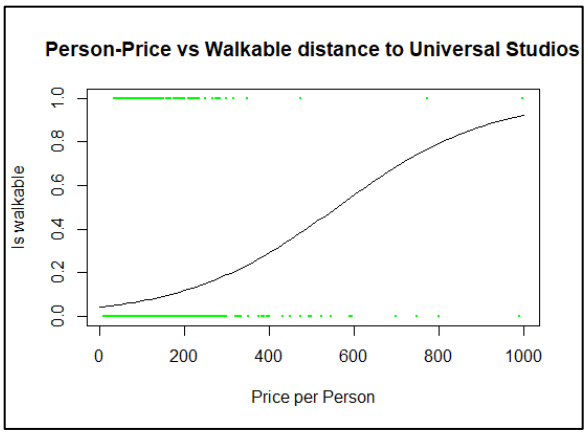


Figure 20: Person Price vs Walkable distance

auc 0.5850832

Auc is low. Error rate is 8%

Price / Person Prediction

Price/Person → $0 ~ $1000

glm(walk ~ PersonPrice, data=USLST, family=binomial)

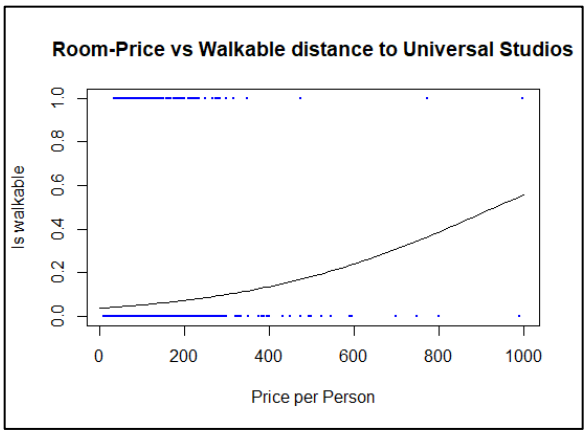


Figure 21: Room Price vs Walkable distance

auc 0.5614795
Error rate 0.087 with 0.5 cutoff'

Price / Room Prediction

Price/Room → $0 ~ $1000
glm(walk ~ RoomPrice, data=USLST, family=binomial)

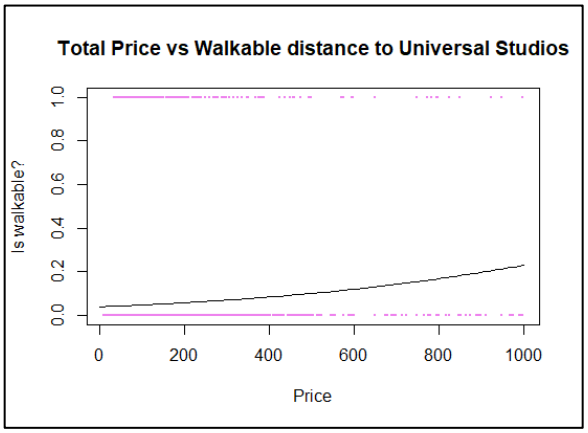


Figure 22: Price vs Walkable distance

Auc 0.5641625
Error rate with 0.5 cutoff - No value due to no true value found

Total Price Prediction

Price → $0 ~ $1000
glm(walk ~ Price, data=USLST, family=binomial)

4) DISCUSSION

Our exploratory data analysis didn't produce meaningful result when we ran linear regression. After the discussion, our team identified a possible bias that may have influenced the result. Because LA contains many popular destinations, the effect of distance on the price could be significantly diminished if the distance was over certain threshold.
Any customer wanting to go to Santa Monica will not rent a property that is 40 miles away. Inversely, anyone visiting Universal Studio will not likely to be booking a rental property outside of 10 mi radius. Due to this realization, new subsets were created. Our team decided to focus on any property that is within 5 miles to minimize the distance bias.

After running the model on each dataset with three different pricing scheme, our team observed similar pattern of low R-squared value with negative slope. Even though our expectation of decreasing price as distance increases, our model only explain very low percentage of data. Therefore, our team concluded linear model was not sufficient to establish linear relationship between the distance and the price.

From our logistic regression, we could not find any evidence that price increase will predict walkable distance to the destination.
Upon reviewing our analysis, we can see that there may be other more popular destination within 5 miles of three destination we chose. This may have influenced bias that skewed our result.
To eliminate or minimize such bias, it will be important to account for other popular destinations and run our regression with more covariates. Or it could be easier to look at a city with only one or two dominant destinations.
Also, it is important to filter out real-estimate volatility. With current trend of increasing

housing market, it may be important to account for such bias by either choosing a city with relatively stable housing price or account for adjusted pricing by incorporating mean and median rental pricing of the neighborhood and adjusting nearby rental pricing accordingly.

4.1) LIMITATIONS

Our datasets contain 89 fields. On top of that, we added additional 6 derived columns to include distance, walkable conditions and normalized pricing. Because of the high degree of freedom in our datasets, choosing one or two variables to run analysis may have contributed in our R-squared value being low. To improve our model, we need to incorporate additional variables. Also, we could've done better data clean up by removing listing that had less than 5 rentals. Rental properties that have short history could have influenced our result. Also, LA was a city that includes many popular destination. Therefore, distance to multiple destination may have been more important. Running regression against multiple destination or choosing a city with one or two dominant destination would've addressed this concern. We have found multiple factors that could have influence on pricing. By addressing them, we believe we will have more meaningful result that explains our data better.

5) CONCLUSIONS

In LA and Melbourne, listings that provide Free Parking have high prices. One of the reasons for this could be that parking is expensive and hence the owners include the price for this in the price of the listings that they provide.
Our analysis produced results that show the relationship between price vs certain amenities, distance, ratings, response times. However, low R-squared that we obtained doesn't allow us to value our model and analysis higher. With high deviance and low R-squared, we looked at what we can do to improve our analysis in the future.

6) FUTURE SCOPE

1. Incorporate additional variables in the analysis to improve the prediction of pricing

2. Clean up data more, remove listing that didn't have review, remove outliers

3. Add more number of datasets to remove the geographical bias encountered in the analysis

4. Choose a city with few popular destinations to analyze walk vs price relationship

5. Find a way to filter out property value - High property value could mean high property rental price

6. Doing sentiment analysis on the description by creating custom categories for aesthetics
Finding the relationship between the sentiment analysis and price of the listing

Exploratory Data Analysis:

1. Refine other methods of regression analysis and find the best possible algorithm for fitting the features to the data.
2. Frame a set of recommendations from the available findings that we have garnered from the exploratory data analysis.

3. Explore relationship between high and low rating with distance and price.

Text Analytics:

1. Creating customized categories for Aesthetics and Amenities
2. Mapping each description with the newly created categories, i.e, counting the frequencies of words for each description (from the new categories)
3. Running a regression model showing the relationship between the ratings and the occurrence of certain words based on the new categories.

APPENDICES

Appendix A : R Code Used

Exploratory Data Analysis:

1. **Text Analytics:**

Melbourne:

I. Data Loading, Transformation and Cleaning

```
library(tm)
library(SnowballC)
lib rary(wordcloud)
library(ggplot2)
library(syuzhet)
library(quanteda)

# Loading data
data <- read.csv(file = "Melbourne.csv",
encoding = "latin1")
data_des <- paste(data$Description,
collapse=" // ")


# Use TM library to process text
# Create a corpus of the entire
description field
docs <-
Corpus(VectorSource(data_des))

# Transformation and cleaning
# convert the text to lower case
docs <- tm_map(docs,
content_transformer(tolower))

# Remove punctuations
# Create the toSpace content
transformer
toSpace <-
content_transformer(function(x, pattern)
{return (gsub(pattern, " ", x))})

docs <- tm_map(docs, toSpace, "-")
docs <- tm_map(docs, toSpace, ":")
docs <- tm_map(docs,
removePunctuation)

# Remove numbers
docs <- tm_map(docs, removeNumbers)
# Remove english common stopwords
docs <- tm_map(docs, removeWords,
stopwords("english"))

# Eliminate extra whitespace
docs <- tm_map(docs, stripWhitespace)

# Text stemming
docs <- tm_map(docs, stemDocument)
# Display the processed corpus
writeLines(as.character(docs[[1]]))

# Remove additional stopwords
docs <- tm_map(docs, removeWords,
c("th", "ll", "m", "s"))
docs <- tm_map(docs, removeWords,
c("apart", "melbourn", "itÃ-", "s"))

# Transform unsual patterns created
due to stemming and removal of white
spaces to nearest meningful root
docs <- tm_map(docs,
content_transformer(gsub), pattern =
"\\b(bedroom|bedsid|twobedroom|onebe
droom|bedr|
        bedder|bedlinen|bedroombathr
oom|bedand|bedcan|bedmedium|bedroo
mson|bedroomstudi|bedsheet|
        livingbedroom|threebedroom|¬
¥bedroom|¬¥bedsid|bedadv|bedbed|bed
couch|beddingwar|beder|bedlamp|
        bedpillow|bedrm|bedrmmelbou
rn|bedro|bedroomapart|bedroomliv|bedr
oomtwo|bedroom√¶||bedroom√¶and|
```

```
        bedroom√≠|bedroon|bedsitt|bedspread|daybedcouch|fourbedroom|roombedroom|twobedroomtwobathroom|
        wifibillsbed)\\b", replacement = "bed")
docs <- tm_map(docs, content_transformer(gsub), pattern = "\\b(bathroom|bath|bathtub|bathr|bdrbathr|
        bedroombathroom|showerbathroom|bathroo|bathshow|kitchenbathroomliv|showerbath|¬¥bathroom|
        batheroom|bathmat|bathro|bathrob|bathroomcentr|bathroomkitchensof|bathroomlaundri|
        bathroompleas|bathrooms√≠|bathsuit|bdrbather|kitchenbathroomlaundrygym|roomkitchenbathroom|
        toiletbathroom|twobedroomtwobathroom|¬¥bath)\\b", replacement = "bath")
```

II. Highlighting most frequently used words in the Description of the listings

```
# Create Document Term Matrix
dtm_mel <- TermDocumentMatrix(docs)
dtm_mel
mat_dtm <- as.matrix(dtm_mel)    # creating a matrix of words with their frequencies
sorted_matdtm <- sort(rowSums(mat_dtm), decreasing =  TRUE )
df_words_freq <- data.frame(words = names(sorted_matdtm), freq = sorted_matdtm)
write_data1 <- write.csv(df_words_freq,file = "frequency_mel.csv")
head(df_words_freq)

# Generate word cloud
par(bg = "grey30")
png(file="Wordcloud_Mel_3.png", width = 600, height = 600, bg = "grey30")
# quartz()
# Reduce the number of words from the wordcloud
wordcloud(df_words_freq$word, df_words_freq$freq, min.freq = 250, col = terrain.colors(length
        (df_words_freq$word), alpha = 0.9), random.order = FALSE, rot.per = 0.3 )
title(main = "Most used words in the description of listings in Melbourne", font.main = 1,
     col.main = "cornsilk3", cex.main = 1.5)
dev.off()

# Plotting Histogram of words
# quartz()
p <- ggplot(subset(df_words_freq, sorted_matdtm>950), aes(words, freq))
p <- p + geom_bar(stat="identity")
p <- p + theme(axis.text.x=element_text(angle=45, hjust=1))
P
```

III. Associations and collocations

```
# Find Associations
findAssocs(dtm_mel, "station", 0.3)
findAssocs(dtm_mel, "gym", 0.3)
findAssocs(dtm_mel1, "breakfast", 0.3)
findAssocs(dtm_mel1, "parking", 0.3)

# Generating bigrams
toks2 <- tokens(data_des)
toks2 <- tokens_remove(toks2, stopwords("english"), padding = TRUE)
seqs <- textstat_collocations(toks2, size =2)
head(seqs, 15)
```

Los Angeles:

I. Data Loading, Transformation and Cleaning

```
# Load data
dataLA <- read.csv(file = "Los Angeles.csv", encoding = "latin1")
dataLA_df <- data.frame(dataLA)

# Use TM library to process text
# Create a corpus of the entire description field
LA <- Corpus(VectorSource(dataLA_df$Description))

# Transformation and cleaning
# Convert the text to lower case
LA <- tm_map(LA, content_transformer(tolower))

# Remove punctuations
# Create the toSpace content transformer
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})

LA <- tm_map(LA, toSpace, "-")
LA <- tm_map(LA, toSpace, ":")
LA <- tm_map(LA, removePunctuation)

# Remove numbers
LA <- tm_map(LA, removeNumbers)
# Remove english common stopwords
LA <- tm_map(LA, removeWords, stopwords("english"))

# Remove additional stopwords
LA <- tm_map(LA, removeWords, c("th", "ll", "m", "s"))
LA <- tm_map(LA, removeWords, c("apart", "itÃ-", "s"))

# Eliminate extra whitespace
LA <- tm_map(LA, stripWhitespace)


# Text stemming
library(SnowballC)
LA <- tm_map(LA, stemDocument)
# Remove additional stopwords
LA <- tm_map(LA, removeWords, c("th", "ll", "m", "s"))
LA <- tm_map(LA, removeWords, c("apart", "itÃ-", "s"))
LA <- tm_map(LA, removeWords, c("will"))

# Transform unsual patterns created due to stemming and removal of white spaces to nearest
# meaningful root
LA <- tm_map(LA, content_transformer(gsub), pattern = "\\b(bedroom|bedsid|twobedroom|
        onebedroom|bedr|bedder|bedlinen|bedroombathroom|bedand|bedcan|bedmedium|
        bedroomson|bedroomstudi|bedsheet|livingbedroom|threebedroom|¬¥bedroom|
        ¬¥bedsid|bedadv|bedbed|bedcouch|beddingwar|beder|bedlamp|bedpillow|
        bedrm|bedrmmelbourn|bedro|bedroomapart|bedroomliv|bedroomtwo|bedroom√¶|
        bedroom√¶and|bedroom√≠|bedroon|bedsitt|bedspread|daybedcouch|fourbedroom|
        roombedroom|twobedroomtwobathroom|wifibillsbed)\\b", replacement = "bed")
LA <- tm_map(LA, content_transformer(gsub), pattern = "\\b(bathroom|bath|bathtub|bathr|
        bdrbathr|bedroombathroom|showerbathroom|bathroo|bathshow|
        kitchenbathroomliv|showerbath|¬¥bathroom|batheroom|bathmat|bathro|
        bathrob|bathroomcentr|bathroomkitchensof|bathroomlaundri|bathroompleas|
        bathrooms√≠|bathsuit|bdrbather|kitchenbathroomlaundrygym|roomkitchenbathroom|
        toiletbathroom|twobedroomtwobathroom|¬¥bath)\\b", replacement = "bath")
LA <- tm_map(LA, content_transformer(gsub), pattern = "walkabl", replacement = "walk")
```

II. Highlighting most frequently used words in the Description of the listings

```
# Create Document Term Matrix
dtm_LA <- TermDocumentMatrix(LA)
dtm_LA
# Create a matrix of words with their
frequencies
mat_dtmLA <- as.matrix(dtm_LA)
sorted_matdtmLA <-
sort(rowSums(mat_dtmLA), decreasing
=  TRUE )
df_words_freqLA <- data.frame(words =
names(sorted_matdtmLA), freq =
sorted_matdtmLA)
write_dataLA <-
write.csv(df_words_freqLA,file =
"frequencyLA.csv")
head(df_words_freqLA)


# Generate word cloud
par(bg = "grey30")
png(file="Wordcloud_LA.png", width = 600,
height = 600, bg = "grey30")
# quartz()
#par(bg = "grey30")
# Reduce the number of words from the
wordcloud
wordcloud(df_words_freqLA$word,
df_words_freqLA$freq, min.freq = 2300,
        col =
terrain.colors(length(df_words_freqLA$word
), alpha = 0.9),
        random.order = FALSE, rot.per = 0.3
)

title(main = "Most used words in the
description of listings in LA",
     font.main = 1, col.main = "cornsilk3",
cex.main = 1.5)
dev.off()

pLA <- ggplot(subset(df_words_freqLA,
sorted_matdtmLA>950), aes(words, freq))
pLA <- p + geom_bar(stat="identity")
pLA <- p +
theme(axis.text.x=element_text(angle=45,
hjust=1))
pLA
```

III. Associations and Collocations

```
# Find Associations

findAssocs(dtm_LA,"bed",0.4)
findAssocs(dtm_LA, "free", 0.3)
findAssocs(dtm_LA, "walk", 0.3)

# Generate bigrams
LA <- read.csv(file = "Los Angeles.csv",
encoding = "latin1", stringsAsFactors =
FALSE)
#data_desLA <- paste(dataLA$Description,
collapse=" // ")
toksLA <- tokens(LA$Description)
toksLA <- tokens_remove(toksLA,
stopwords("english"), padding = TRUE)
seqsLA <- textstat_collocations(toksLA, size
=2)
head(seqsLA, 15)
```

2. Comparing the range of price for the listings with “breakfast” and “washer”

```
price <-
data.frame(mel_data$Amenities,mel_data$
Price) # create a data frame with price and
amenities columns
price$bfvswash <- rep(0, nrow(price)) # add
a column
price$bfvswash[grep("Washer",mel_data$A
menities)] <- 1 # replace 0 with 1 where the
word washer exists in the amenities
boxplot(mel_data.Price~bfvswash,
data=price,
ylab="Price",names=c("Breakfast","Washer"
))# comparing the range of price for the
listings with “breakfast” and “washer”
```

3. **Frequencies for Amenities**

```
freq_price <-
paste(price$mel_data.Amenities,
collapse=",")
library(tm)
docs <- Corpus(VectorSource(freq_price))
docs <- tm_map(docs, stripWhitespace)
head(docs)
docs <- tm_map(docs, stemDocument)
docs <- strsplit(as.character(docs), split=",")
words.fre <-table(unlist(docs))
```

```
fre <- as.data.frame(words.fre)
sorted_fre <- fre[order(fre[,2],decreasing =
TRUE),]
names(price) <- c("Amenities","Price","Gym
vs Breakfast")
```

From the list we can see that usually most of the listings provide: Kitchen, Wireless Internet,Washer, etc.

```
price$`Gym vs
Breakfast`[grep("Breakfast",price$Amenities
)] <-1
price$D[grep("Gym",price$A)] <- 2
boxplot(price$B~price$D,ylab="Price",
ylim=c(0,600),
names=c("None","Breakfast","Gym"))
```

4. Impact of ratings and response rate of hosts on the price of the room

```
LA = read.csv('LA.csv')
View(x=LA)
price=LA$Price
rating=LA$Review.Scores.Rating
accuracy=LA$Review.Scores.Accuracy
rate=lm(price~rating)
response=LA$Host.Response.Rate
summary(rate)
resp=lm(price~response)
summary(resp)

LA2=subset(LA,
LA$Host.Response.Rate!=0)
nrate=lm(LA2$Price~LA2$Host.Response.R
ate)
plot(nrate)
View(LA2)
summary(nrate)

cor.test(cleaned, price,
method=c("pearson"))
cor(cleaned,price,  method = "pearson", use
= "complete.obs")

bedrooms=LA$Bedrooms
if (bedrooms>0) {
  nprice=price/bedrooms
} else {
  nprice=price
}
#Normalized prices

cleaned=LA$Review.Scores.Cleanliness
cleanrate=lm(price~cleaned)
summary(cleanrate)


install.packages("UsingR")
install.packages('colorspace')
library(UsingR)

ggplot(LA, aes(y=nprice,
x=Review.Scores.Cleanliness)) +
geom_point(size=1, alpha=0.6,
color="blue") + ylab("Price of the room(in
$)") + xlab("Cleanliness Review Rating of
the room") + ggtitle("Los Angeles:
Cleanliness rating vs Price correlation")

ggplot(LA, aes(y=nprice,
x=Review.Scores.Rating)) +
geom_point(size=1, alpha=0.6) +
ylab("Adjusted Normalized Price of the
room(in $)") + xlab("Review Rating of the
room") + ggtitle("Los Angeles: Ratings vs
Price correlation")

plot(y=LA2$Price,
x=LA2$Host.Response.Rate,ylab="Normali
zed price(in $)", xlab="Response rate(in
%)")
title('Los Angeles: Response rate vs Price
correlation')
```

5. *Pricing vs Distance Analysis*

```
####################################
##################
# Data
Prep
                    #
####################################
##################

getwd()
```

```
setwd("G:/Team Drives/INFX 573/Ryan")
#install.packages("sp")
#install.packages("ResourceSelection")
#install.packages("UsingR")
#install.packages('colorspace')
#install.packages("geosphere")

library(ResourceSelection)
library(sp)
library(ISLR)
library(AUC)
library(geosphere)

library(UsingR)
LAabnb <- read.csv('LA.csv') #Loading LA.csv

USGeo<-rbind(c(as.numeric(-118.356051), as.numeric(34.136518))) #Universal Studios Hollywood GPS location
downtownGeo<-rbind(c(as.numeric(-118.24677), as.numeric(34.04071))) #LA downtown GPS location
SMGeo<-rbind(c(as.numeric(-118.49763), as.numeric(34.00918))) #LA downtown GPS location

#Calculate distance from each listing to Universal Studios
LAabnb$USDist <- c(distm(LAabnb[,c('Longitude', 'Latitude')], USGeo, fun = distVincentyEllipsoid))/1609.34

#Calculate distance from each listing to downtown LA
LAabnb$DTDist <- c(distm(LAabnb[,c('Longitude', 'Latitude')], downtownGeo, fun = distVincentyEllipsoid))/1609.34


#Calculate distance from each listing to Santa Monica beach
LAabnb$SMDist <- c(distm(LAabnb[,c('Longitude', 'Latitude')], SMGeo, fun = distVincentyEllipsoid))/1609.34

#Find per person price (total price / max number of people)
LAabnb$PersonPrice<- with(LAabnb, Price / Accommodates)

#Find per room price (total price / number of room)
price<-LAabnb$Price
bedrooms<-LAabnb$Bedrooms
bedrooms[bedrooms<0.1] <- 1
nprice<-price/bedrooms
LAabnb$RoomPrice<-nprice

#####################################
#**Exploratory Data Analysis**      #
#####################################

names(LAabnb)
#Linear model of distance to all destination vs price using all data
price.linmodel <- lm(Price ~ USDist+SMDist+DTDist, data=LAabnb)
summary(price.linmodel)
#R-squared is 0.023. Very low and doesn't really explain.

#Linear model of distance to all destination vs Person price using all data
person.price.linmodel <- lm(PersonPrice ~ USDist+SMDist+DTDist, data=LAabnb)
summary(person.price.linmodel)
#R-squared is 0.021. Very low and doesn't really explain.

#Linear model of downtown distance to price using all data
room.price.linmodel <- lm(RoomPrice ~ USDist+SMDist+DTDist, data=LAabnb)
summary(room.price.linmodel)
#R-squared is 0.019. Very low and doesn't really explain.

#####################################
#**Create Data subsets based on distance**  #
```

```
#####################################
##################

#Create a dataframe USLST that only
contains listings of rentals that is within 5
miles to Universla Studios.
US <- LAabnb[which(LAabnb$USDist < 5),]

#Create a dataframe DTLST that only
contains listings of rentals that is within 5
miles to downtown LA.
DT <- LAabnb[which(LAabnb$DTDist < 5),]

#Create a dataframe DTLST that only
contains listings of rentals that is within 5
miles to Santa Monica.
SM <- LAabnb[which(LAabnb$SMDist < 5),]

#Remove Null from USLST, DTLST,
SMLST in colume 90 ~ 94
USLST<-US[complete.cases(US[, 90:94]), ]
SMLST<-SM[complete.cases(SM[, 90:94]), ]
DTLST<-DT[complete.cases(DT[, 90:94]), ]

#####################################
##################
#Simple Analysis using
LM                    #
#####################################
##################

#Linear model of US distance to price using
rentals within 10 miles to Universal Studios.
us.price.limited <- lm(Price ~ USDist,
data=USLST)

#Linear model of US distance to price using
rentals within 10 miles to Downtown.
dt.price.limited <- lm(Price ~ DTDist,
data=DTLST)

#Linear model of US distance to price using
rentals within 10 miles to Santa Monica.
sm.price.limited <- lm(Price ~ SMDist,
data=SMLST)
summary(us.price.limited)
#R-squred is 0.015. Doesn't really explain
summary(dt.price.limited)
#R-squred is 0.016. Doesn't really explain
summary(sm.price.limited)
#R-squred is 0.022. Doesn't really explain

#Linear model of US distance to per person
price using rentals within 10 miles to
Universal Studios.
us.person.price.limited <- lm(PersonPrice ~
USDist, data=USLST)

#Linear model of downtown distance to per
person price using rentals within 10 miles to
downtown LA.
dt.person.price.limited <- lm(PersonPrice ~
DTDist, data=DTLST)

#Linear model of SM distance to per person
price using rentals within 10 miles to Santa
Monica.
sm.person.price.limited <- lm(PersonPrice ~
SMDist, data=SMLST)
#
summary(us.person.price.limited)
#R-squred is 0.010. Doesn't really explain
summary(dt.person.price.limited)
#R-squred is 0.006. Doesn't really explain
summary(sm.person.price.limited)
#R-squred is 0.017. Doesn't really explain
#Santa Monica has the highest rate of
decrease in price when distance to SM
increases when comparing per person
pricing

#Linear model of US distance to per person
price using rentals within 10 miles to
Universal Studios.
us.room.price.limited <- lm(RoomPrice ~
USDist, data=USLST)

#Linear model of downtown distance to per
person price using rentals within 10 miles to
downtown LA.
dt.room.price.limited <- lm(RoomPrice ~
DTDist, data=DTLST)

#Linear model of SM distance to per person
price using rentals within 10 miles to Santa
Monica.
sm.room.price.limited <- lm(RoomPrice ~
SMDist, data=SMLST)

summary(dt.room.price.limited)
```

```
summary(sm.room.price.limited)
summary(us.room.price.limited)

#####################################
##################
#Universal Studios LM
#####################################
##################

x.us.dist<-USLST$USDist
y.us.rm<-USLST$RoomPrice
plot(x.us.dist, y.us.rm, pch=20, cex=1,
col="steelblue3", main="Distance vs Price
per Room", xlab="Distance to Universal
Studios", ylab="Price / Room")
abline(us.room.price.limited, lty=1, col="red"
)
legend("topleft", legend=c("Slope = -
7.3462", "Intercept = 127.06"),
col=c("black", "red"), pch=c(1, 15))

y.us.per<-USLST$PersonPrice
plot(x.us.dist, y.us.per, pch=20, cex=1,
col="tan1", main="Distance vs Price per
Person", xlab="Distance to Universal
Studios", ylab="Price / Person")
abline(us.person.price.limited, lty=1,
col="green" )
legend("topleft", legend=c("Slope = -3.13",
"Intercept = 57.28"), col=c("black", "red"),
pch=c(1, 15))
y.us<-USLST$Price
plot(x.us.dist, y.us, pch=20, cex=1,
col="violet", main="Distance vs Price",
xlab="Distance to Universal Studios",
ylab="Total Price")
abline(us.price.limited, lty=1, col="blue" )
legend("topleft", legend=c("Slope = -
15.064", "Intercept = 195.104"),
col=c("black", "red"), pch=c(1, 15))

#####################################
##################
#Santa Monica LM                    #
#####################################
##################

x.sm.dist<-SMLST$SMDist
y.sm.rm<-SMLST$RoomPrice
plot(x.sm.dist, y.sm.rm, pch=20, cex=1,
col="steelblue3", main="Distance vs Price
per Room", xlab="Distance to Santa
Monica", ylab="Price / Room")
abline(sm.room.price.limited, lty=1,
col="red" )
legend("topleft", legend=c("Slope = -11.27",
"Intercept = 149.40"), col=c("black", "red"),
pch=c(1, 15))

y.sm.per<-SMLST$PersonPrice
plot(x.sm.dist, y.sm.per, pch=20, cex=1,
col="tan1", main="Distance vs Price per
Person", xlab="Distance to Santa Monica",
ylab="Price / Person")
abline(sm.person.price.limited, lty=1,
col="green" )
legend("topleft", legend=c("Slope = -4.25",
"Intercept = 67.42"), col=c("black", "red"),
pch=c(1, 15))

y.sm<-SMLST$Price
plot(x.sm.dist, y.sm, pch=20, cex=1,
col="violet", main="Distance vs Price",
xlab="Distance to Santa Monica",
ylab="Total Price")
abline(sm.price.limited, lty=1, col="blue" )
legend("topleft", legend=c("Slope = -18.84",
"Intercept = 225.01"), col=c("black", "red"),
pch=c(1, 15))

#####################################
##################
#Downtown LM                        #
#####################################
##################

x.dt.dist<-DTLST$DTDist
y.dt.rm<-DTLST$RoomPrice
plot(x.dt.dist, y.dt.rm, pch=20, cex=1,
col="steelblue3", main="Distance vs Price
per Room", xlab="Distance to Downtown",
ylab="Price / Room")
abline(dt.room.price.limited, lty=1, col="red"
)
legend("topleft", legend=c("Slope = -10.75",
"Intercept = 123.77"), col=c("black", "red"),
pch=c(1, 15))

y.dt.per<-DTLST$PersonPrice
```

```
plot(x.dt.dist, y.dt.per, pch=20, cex=1,
col="tan1", main="Distance vs Price per
Person", xlab="Distance to Downtown",
ylab="Price / Person")
abline(dt.person.price.limited, lty=1,
col="green" )
legend("topleft", legend=c("Slope = -1.99",
"Intercept = 46.87"), col=c("black", "red"),
pch=c(1, 15))

y.dt<-DTLST$Price
plot(x.dt.dist, y.dt, pch=20, cex=1,
col="violet", main="Distance vs Price",
xlab="Distance to Downtown", ylab="Total
Price")
abline(dt.price.limited, lty=1, col="blue" )
legend("topleft", legend=c("Slope = -8.84",
"Intercept = 141.14"), col=c("black", "red"),
pch=c(1, 15))

####################################
##################
#Logistic Regression                #
####################################
##################

#Analysis of Walking distance of a property
to the nearest popular destination vs price

#Create a column named walk. 0 mean
non-walking distance, 1 means walking
distance.
USLST$walk<-rep(0, nrow(USLST))
#Update walk column with 1 if miles is less
than 1.5 miles
USLST$walk[USLST$USDist<=1.5]<-1

#Repeat the same process to create a data
set for Santa Monica
SMLST$walk<-rep(0, nrow(SMLST))
SMLST$walk[SMLST$SMDist<=1.5]<-1

#Repeat the same process to create a data
set for Downtown
DTLST$walk<-rep(0, nrow(DTLST))
DTLST$walk[DTLST$DTDist<=1.5]<-1

#Logistic model of walking distance vs
RoomPrice, PersonPrice, TotalPrice in
Universal Studios datasets

us.price.glimited <- glm(walk ~
RoomPrice+PersonPrice+Price,
data=USLST, family=binomial)
summary(us.price.glimited)

x.ps<-USLST$PersonPrice
x.rm<-USLST$RoomPrice
x.pr<-USLST$Price
#Created fit.us from the logistic model
fit.us<-fitted(us.price.glimited)

#Plot total pice vs Walking distance
plot(x.pr, USLST$walk, col="blue")
points(x.pr, fit.us, pch=19, cex=0.2,
col="black", main="Walking Distance 1.5")

#Plot room pice vs Walking distance
plot(x.rm, USLST$walk, col="violet")
points(x.rm, fit.us, pch=19, cex=0.2,
col="red", main="Walking Distance 1.5")

#Plot person pice vs Walking distance
plot(x.ps, USLST$walk, col="blue3")
points(x.ps, fit.us, pch=19, cex=0.2,
col="green", main="Walking Distance 1.5")

tab.us.rm<-table(USLST$walk, fit.us>=0.5)
#(tab.us.rm[1,2]+tab.us.rm[2,1])/sum(tab.us.
rm)

#Increasing walk to 2.5 miles and repeating
to see if our data works better

#Create a column named walk. 0 mean
non-walking distance, 1 means walking
distance.
USLST$walk<-rep(0, nrow(USLST))

#Update walk column with 1 if miles is less
than 2.5 miles
USLST$walk[USLST$USDist<=2.5]<-1

#Repeat the same process to create a data
set for Santa Monica
SMLST$walk<-rep(0, nrow(SMLST))
SMLST$walk[SMLST$SMDist<=2.5]<-1

#Repeat the same process to create a data
set for Downtown
DTLST$walk<-rep(0, nrow(DTLST))
```

```
DTLST$walk[DTLST$DTDist<=2.5]<-1

#Logistic model of walking distance (2.5 Miles) vs RoomPrice, PersonPrice, TotalPrice in Universal Studios datasets
#Plot fitted value again to see if data points is spread better
us.price.glimited <- glm(walk ~ RoomPrice+PersonPrice+Price, data=USLST, family=binomial)
summary(us.price.glimited)

x.ps<-USLST$PersonPrice
x.rm<-USLST$RoomPrice
x.pr<-USLST$Price
fit.us<-fitted(us.price.glimited)

plot(x.pr, USLST$walk, col="blue", main="Walking Distance 2.5")
points(x.pr, fit.us, pch=19, cex=0.2, col="black")

plot(x.rm, USLST$walk, col="violet", main="Walking Distance 2.5")
points(x.rm, fit.us, pch=19, cex=0.2, col="red")

plot(x.ps, USLST$walk, col="blue3")
points(x.ps, fit.us, pch=19, cex=0.2, col="green", main="Walking Distance 2.5")

tab.us.rm<-table(USLST$walk, fit.us>=0.5)
(tab.us.rm[1,2]+tab.us.rm[2,1])/sum(tab.us.rm)

#Decreasing it to 2 miles and repeating to see if our data works better

#Create a column named walk. 0 mean non-walking distance, 1 means walking distance.
USLST$walk<-rep(0, nrow(USLST))
#Update walk column with 1 if miles is less than 2 miles
USLST$walk[USLST$USDist<=2]<-1

#Repeat the same process to create a data set for Santa Monica
SMLST$walk<-rep(0, nrow(SMLST))
SMLST$walk[SMLST$SMDist<=2]<-1

#Repeat the same process to create a data set for Downtown
DTLST$walk<-rep(0, nrow(DTLST))
DTLST$walk[DTLST$DTDist<=2]<-1

#Logistic model of walking distance (2 Miles) vs RoomPrice, PersonPrice, TotalPrice in Universal Studios datasets
#Plot fitted value again to see if data points is spread better
us.price.glimited <- glm(walk ~ RoomPrice+PersonPrice+Price, data=USLST, family=binomial)
summary(us.price.glimited)

x.ps<-USLST$PersonPrice
x.rm<-USLST$RoomPrice
x.pr<-USLST$Price
fit.us<-fitted(us.price.glimited)
plot(x.pr, USLST$walk, col="blue", main="Walking Distance 2")
points(x.pr, fit.us, pch=19, cex=0.2, col="black")

plot(x.rm, USLST$walk, col="violet")
points(x.rm, fit.us, pch=19, cex=0.2, col="red", main="Walking Distance 2.5")

plot(x.ps, USLST$walk, col="blue3")
points(x.ps, fit.us, pch=19, cex=0.2, col="green", main="Walking Distance 2.5")

tab.us.rm<-table(USLST$walk, fit.us>=0.5)
(tab.us.rm[1,2]+tab.us.rm[2,1])/sum(tab.us.rm)

#Logistic regression of walking distance of 2 miles vs Person Price
us.person.price.glimited <- glm(walk ~ PersonPrice, data=USLST, family=binomial)

summary(us.person.price.glimited)
#Create x value from $0 to $1000 with $10 increment
xprice<-seq(0, 1000, 10)
#Using us.person.price.glimited model to predict walking distance using xprice
```

```
y.us.per.dist<-
predict(us.person.price.glimited,
list(PersonPrice = xprice), type='response')
#Plot RoomPirce vs walk
plot(USLST$RoomPrice, USLST$walk, pch
= 16, cex=0.3, col="green", main="Person-
Price vs Walkable distance to Universal
Studios", xlab = "Price per Person", ylab =
"Is walkable")
#Draw a line using the prediction
lines(xprice, y.us.per.dist)
y.us<-factor(USLST$walk)
fits.us.per<-fitted(us.person.price.glimited)
rr.us.per <-roc(fits.us.per, y.us)
#Calculate AUC
auc(rr.us.per)

#Plot ROC for ROC for Walk vs Price per
Person - Universal Studios
plot(rr.us.per, main="ROC for Walk vs Price
per Person - Universal Studios")

#False true + true false / total = error
percent
#Error rate of 0.5 cutoff
tab.us.per.50<-table(USLST$walk,
fits.us.per>=0.50)
(tab.us.per.50[1,2]+tab.us.per.50[2,1])/sum(t
ab.us.per.50)

#Repeat above analysis using room price
instead of person price
us.room.price.glimited <- glm(walk ~
RoomPrice, data=USLST, family=binomial)
summary(us.room.price.glimited)
y.us.rm.dist<-
predict(us.room.price.glimited,
list(RoomPrice = xprice), type='response')
plot(USLST$RoomPrice, USLST$walk, pch
= 16, cex=0.3, col="blue", main="Room-
Price vs Walkable distance to Universal
Studios", xlab = "Price per Person", ylab =
"Is walkable")
lines(xprice, y.us.rm.dist)
fits.us.rm<-fitted(us.room.price.glimited)
rr.us.rm <-roc(fits.us.rm, y.us)
auc(rr.us.rm)
plot(rr.us.rm, main="ROC for Walk vs Price
per Room - Universal Studios")
tab.us.rm.50<-table(USLST$walk,
fits.us.rm>=0.50)
(tab.us.rm.50[1,2]+tab.us.rm.50[2,1])/sum(ta
b.us.rm.50)

#Repeat above analysis using total price
instead of person price
us.price.glimited <- glm(walk ~ Price,
data=USLST, family=binomial)
summary(us.price.glimited)
y.us.dist<- predict(us.price.glimited,
list(Price = xprice), type='response')
plot(USLST$Price, USLST$walk, pch = 16,
cex=0.3, col="violet", main="Total Price vs
Walkable distance to Universal Studios",
xlab = "Price", ylab = "Is walkable?")
lines(xprice, y.us.dist)
fits.us<-fitted(us.price.glimited)
rr.us <-roc(fits.us, y.us)
auc(rr.us)
plot(rr.us, main="ROC for Walk vs Price -
Universal Studios")
tab.us.50<-table(USLST$walk,
fits.us>=0.50)
#(tab.us.50[1,2]+tab.us.50[2,1])/sum(tab.us.
50)
#Error rate cannot be calculated since

#Repeat above analysis using total price
instead of room price and using Santa
Monica dataset
sm.room.price.glimited <- glm(walk ~
RoomPrice, data=SMLST, family=binomial)
summary(sm.room.price.glimited)
y.sm.rm.dist<-
predict(sm.room.price.glimited,
list(RoomPrice = xprice), type='response')
plot(SMLST$RoomPrice, SMLST$walk, pch
= 16, main="Room-Price vs Walkable
distance to Santa Monica", xlab = "Price per
Room", ylab = "Is walkable")
lines(xprice, y.sm.rm.dist)
y.sm<-factor(SMLST$walk)
fits.sm.rt<-fitted(sm.room.price.glimited)
rr.sm.rt <-roc(fits.sm.rt, y.sm)
auc(rr.sm.rt)
plot(rr.sm.rt, main="ROC for Walk vs Price
per Room - Santa Monica")
tab.sm.rt.50<-table(SMLST$walk,
fits.sm.rt>=0.50)
```

```
(tab.sm.rt.50[1,2]+tab.sm.rt.50[2,1])/sum(tab
.sm.rt.50)

#Repeat above analysis using total price
instead of person price and using Santa
Monica dataset
sm.person.price.limited <- glm(walk ~
PersonPrice, data=SMLST,
family=binomial)
summary(sm.person.price.limited)
y.sm.per.dist<-
predict(sm.person.price.limited,
list(PersonPrice = xprice), type='response')
plot(SMLST$RoomPrice, SMLST$walk, pch
= 16, main="Person-Price vs Walkable
distance to Santa Monica", xlab = "Price per
Person", ylab = "Is walkable")
lines(xprice, y.sm.per.dist)
fits.sm.per<-fitted(sm.person.price.limited)
rr.sm.per <-roc(fits.sm.per, y.sm)
auc(rr.sm.per)
plot(rr.sm.per, main="ROC for Walk vs Price
per Person - Santa Monica")
tab.sm.per.50<-table(SMLST$walk,
fits.sm.per>=0.50)
(tab.sm.per.50[1,2]+tab.sm.per.50[2,1])/sum
(tab.sm.per.50)

#Repeat above analysis using total price
instead of room price and using Santa
Monica dataset
sm.price.limited <- glm(walk ~ Price,
data=SMLST, family=binomial)
summary(sm.price.limited)
xprice<-seq(0, 1000, 10)
y.sm.dist<- predict(sm.price.limited,
list(Price = xprice), type='response')
plot(SMLST$Price, SMLST$walk, pch = 16,
main="Total Price vs Walkable distance to
Santa Monica", xlab = "Price", ylab = "Is
walkable")
lines(xprice, y.sm.dist)
fits.sm<-fitted(sm.price.limited)
rr.sm <-roc(fits.sm, y.sm)
auc(rr.sm)
plot(rr.sm, main="ROC for Walk vs Price -
Santa Monica")
tab.sm.50<-table(SMLST$walk,
fits.sm>=0.5)
#(tab.sm.50[1,2]+tab.sm.50[2,1])/sum(tab.s
m.50)
#> tab.sm.50

#FALSE
#0  3027
#1   451
#There is no true value. This is very bad.
#Logistic regression between total price and
walkable distance does not contain any true
true or falut true.

#Repeat above analysis using total price
instead of room price and using downtown
dataset
dt.room.price.limited <- glm(walk ~
RoomPrice, data=DTLST, family=binomial)
summary(dt.room.price.limited)
y.dt.rm.dist<- predict(dt.room.price.limited,
list(RoomPrice = xprice), type='response')
plot(DTLST$RoomPrice, DTLST$walk, pch
= 16, main="Room-Price vs Walkable
distance to Downtown LA", xlab = "Price per
Room", ylab = "Is walkable")
lines(xprice, y.dt.rm.dist)
y.dt<-factor(DTLST$walk)
fits.dt.rm<-fitted(dt.room.price.limited)
rr.dt.rm <-roc(fits.dt.rm, y.dt)
auc(rr.dt.rm)
plot(rr.dt.rm, main="ROC for Walk vs Price
per Room - Downtown LA")
tab.dt.rm.50<-table(DTLST$walk,
fits.dt.rm>=0.50)
(tab.dt.rm.50[1,2]+tab.dt.rm.50[2,1])/sum(ta
b.dt.rm.50)

#Repeat above analysis using total price
instead of person price and using downtown
dataset
dt.person.price.limited <- glm(walk ~
PersonPrice, data=DTLST, family=binomial)
summary(dt.person.price.limited)
y.dt.per.dist<-
predict(dt.person.price.limited,
list(PersonPrice = xprice), type='response')
plot(DTLST$RoomPrice, DTLST$walk, pch
= 16, main="Person-Price vs Walkable
distance to Downtown LA", xlab = "Price per
Person", ylab = "Is walkable")
lines(xprice, y.dt.per.dist)
fits.dt.per<-fitted(dt.person.price.limited)
rr.dt.per <-roc(fits.dt.per, y.dt)
```

```
auc(rr.dt.per)
plot(rr.dt.per, main="ROC for Walk vs Price
per Person - Downtown LA")
tab.dt.per.50<-table(DTLST$walk,
fits.dt.per>=0.50)
(tab.dt.per.50[1,2]+tab.dt.per.50[2,1])/sum(t
ab.dt.per.50)

#Repeat above analysis using total price
instead of total price and using downtown
dataset
dt.price.glimited <- glm(walk ~ Price,
data=DTLST, family=binomial)
summary(us.price.glimited)
y.dt.dist<- predict(dt.price.glimited, list(Price
= xprice), type='response')
plot(DTLST$Price, DTLST$walk, pch = 16,
main="Total Price vs Walkable distance to
Downtown LA", xlab = "Price", ylab = "Is
walkable")
lines(xprice, y.dt.dist)
fits.dt<-fitted(dt.price.glimited)
rr.dt <-roc(fits.dt, y.dt)
auc(rr.dt)
plot(rr.dt, main="ROC for Walk vs Price -
Downtown LA")
tab.dt.50<-table(DTLST$walk, fits.dt>=0.50)
(tab.dt.50[1,2]+tab.dt.50[2,1])/sum(tab.dt.50
)
```