Technical Report: Predicting Yelp Price Ranges

Josh Dey, Emmett Powers, & Giorlando Ramirez

12/10/2019

Abstract

In this project, we leverage a large Yelp data set comprised of about 200k observations in order to build a predictive model of restaurant prices. We go about answering this question deploying three supervised learning models: linear regression, ordinal regression, and classification trees. We find that the available variables have fairly limited predictive power across our models.

Preparing Libraries

```
library(ggplot2)
library(Amelia)
library(rlist)
library(dplyr)
library(tidyverse)
library(jsonlite)
library(MASS)
```

Introduction

In this project we want to use a large dataset provided by Yelp to predict restaurant prices. Economic intuition would have it that there are certain things restaurants might sell and/or provide that would make them more or less expensive, a prime example being the sale of alcohol. Thus, with this project we wish to put this intuition to the test. The Yelp dataset is comprised of many individual restaurant observations with variables that are likely determinants of the restaurants prices, such as whether they provide outdoor seating or the expected attire, so with this data we go about building a variety of models which might predict a given restaurant's price range.

The Data

The data used in this project comes from the Yelp Open Dataset. This is a massive dataset Yelp releases for educational purposes, which is comprised of three main sub datasets: reviews, business, and images. In this project we use the business dataset, a dataset which contains a variety of descriptive variables for 192,609 business location. However, in this project we're focused specifically on restaurants, so when we trim the data to only include restaurants we end up with 59,371 total observations. Each observation is a given restaurant with all of the associated variables, from name of the restaurant, star rating through whether the restaurant serves alcohol. There are a total of 53 variables.

In order to perform any meaningful analysis we needed to clean the data. Thus, as mentioned previously, we began this process by cutting the data to only include restaurants. This resulted in a total of 59,371 observations. We then filtered the data set to only include predictor variables we believed would have the

strongest predictive capability- such as: # of stars, whether the restaurant served alcohol, etc.- and proceeded to further cut the data to only include observations which were complete (included an observation for every variable). This resulted in a dataset with 40,584 observations, a loss of about 30% of original observations. We have no reason to believe there exist systematic reasons as to why restaurants would omit data on these variables, and thus chose this as our final dataset, which we use to perform our analysis and create our models.

Exploratory Data Analysis

Explore the structure of the data through graphics. Here you can utilize both traditional plots as well as methods from unsupervised learning. Understanding the distribution of your response is particular important, but also investigate bivariate and higher-order relationships that you expect to be particular interesting.

In this section we look over the data in order to identify trends, concerns, and other consideration we might need to take into account when creating our models.

Initial Data Prep

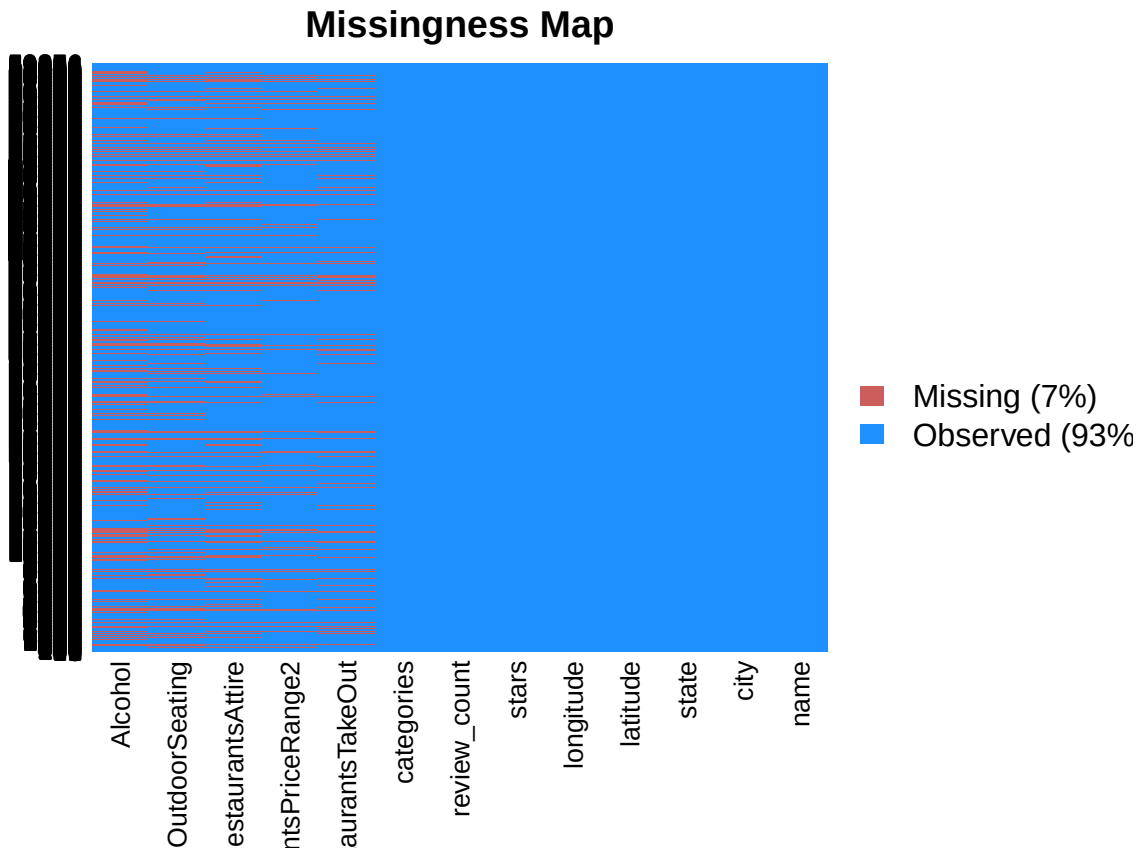
First, we load up the data before we begin our exploratory analysis. We also filter the data for the explanatory variables we wish to use.

```
business <- stream_in(file("business.json"))
clean.business <- business

## Filtering for restaurants
clean.business$Restaurant <- 0
new.business <- mutate(clean.business, Restaurant=grepl("Restaurants", clean.business$categories))
restaurants <- filter(new.business, Restaurant == "TRUE")

## Isolating the important variables
attribute <- restaurants$attributes
restaurant_data <- restaurants
restaurant_data$attributes <- NULL
restaurant_data$Restaurant <- NULL
restaurant_data <- cbind(restaurant_data, attribute)
voi <- c("name", "city", "state", "latitude", "longitude",
         "stars", "review_count", "RestaurantsTakeOut",
         "RestaurantsPriceRange2", "OutdoorSeating", "Alcohol",
         "categories", "RestaurantsAttire")
rdata <- subset(restaurant_data, select=voi)
```

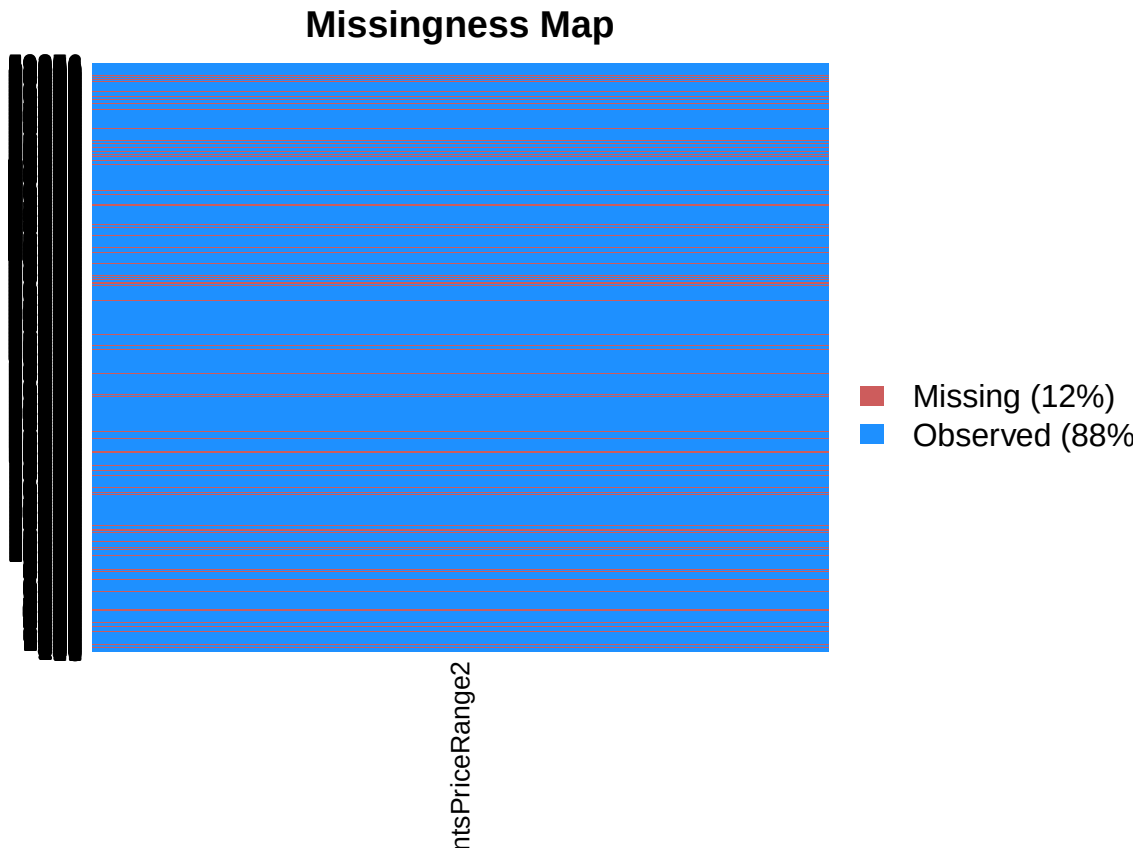
Missingness

Now we look at the missingness in the data, and we decouple it into response and some predictor variables.

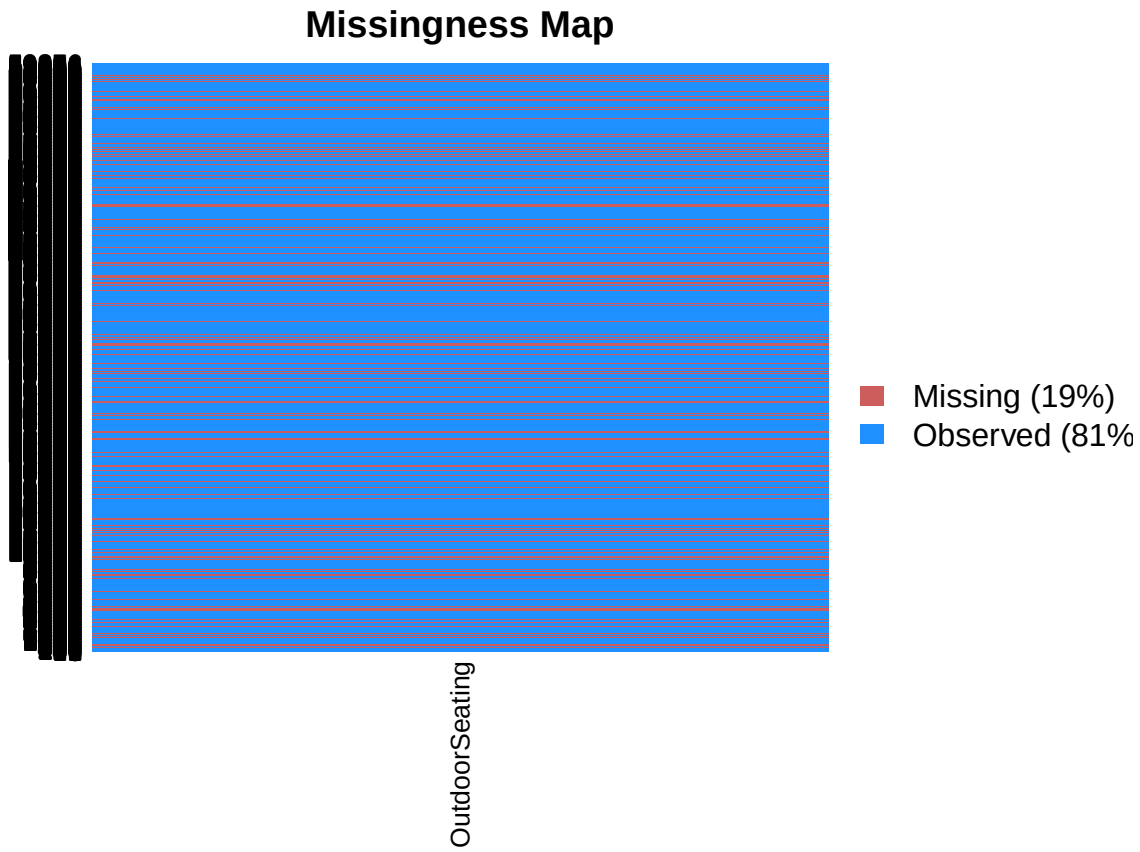
```
## Missingness when including all of the restaurant observations
missmap(rdata)
```



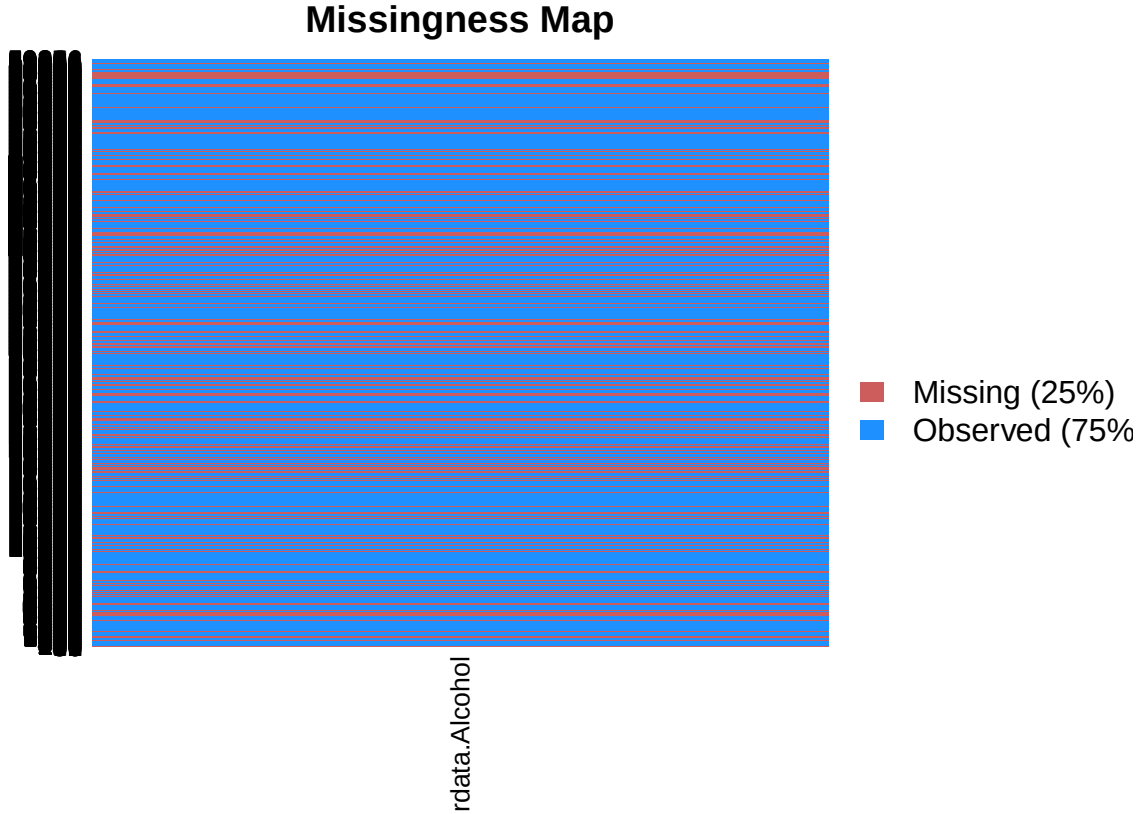

```
## Missingness in the response
prices <- data.frame(rdata$RestaurantsPriceRange2)
missmap(prices)
```



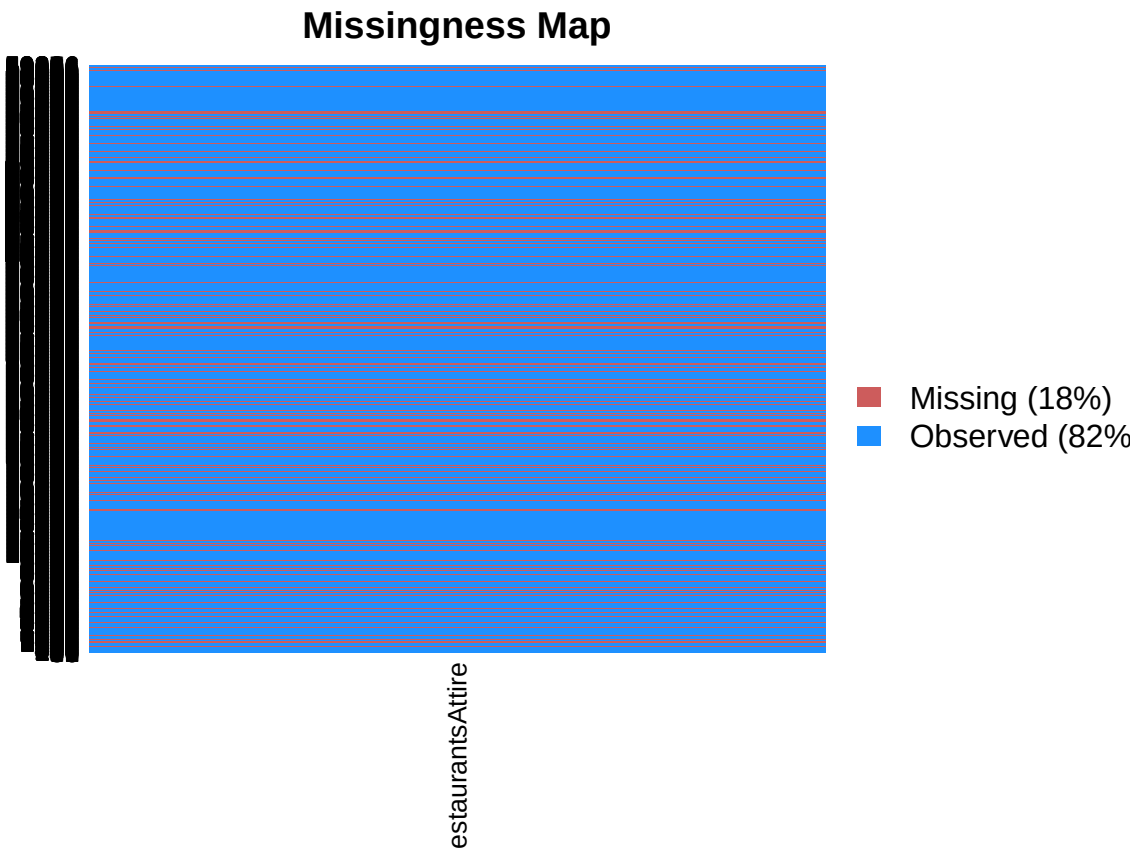

```
## Missingness in the explanatory variables
alcohol <- data.frame(rdata$Alcohol)
outdoorseating <- data.frame(rdata$OutdoorSeating)
dresscode <- data.frame(rdata$RestaurantsAttire)
missmap(outdoorseating)
```




```
missmap(alcohol)
```



```
missmap(dresscode)
```



The missingness maps reveal that though our overall missingness is minimal, there are a couple things worth noting. In terms of the response variable, it is only 12% which is not too concerning since we still have a value for the major of observations. Further, we notice that the missingness is concentrated around the more niche variables, which are the determinants of the price range. This would be an issue if we considered there to exist a systematic issue in the missingness occurring, however we do not think this is the case. We believe the missingness is randomly distributed, so we assume it is not problematic to simply get rid of observations which do not have values for all of our variables. Thus, the final data we use will contain only complete observations. Now that we have a cursory look at the overall data and know that we will get rid of incomplete observations, we turn to preparing it for a survey of the response and predictive variables.

Preparing Data

```
##Cleaning Data to only include complete observations
rdata.clean <- rdata[complete.cases(rdata),]

## Fixing Variables
rdata.clean$OutdoorSeating <- as.logical(rdata.clean$OutdoorSeating)
rdata.clean$OutdoorSeating <- as.numeric(rdata.clean$OutdoorSeating)

rdata.clean$RestaurantsTakeOut <- as.logical(rdata.clean$RestaurantsTakeOut)
rdata.clean$RestaurantsTakeOut <- as.numeric(rdata.clean$RestaurantsTakeOut)

#Assigning categorical levels 1, 2, or 3 for alcohol
```

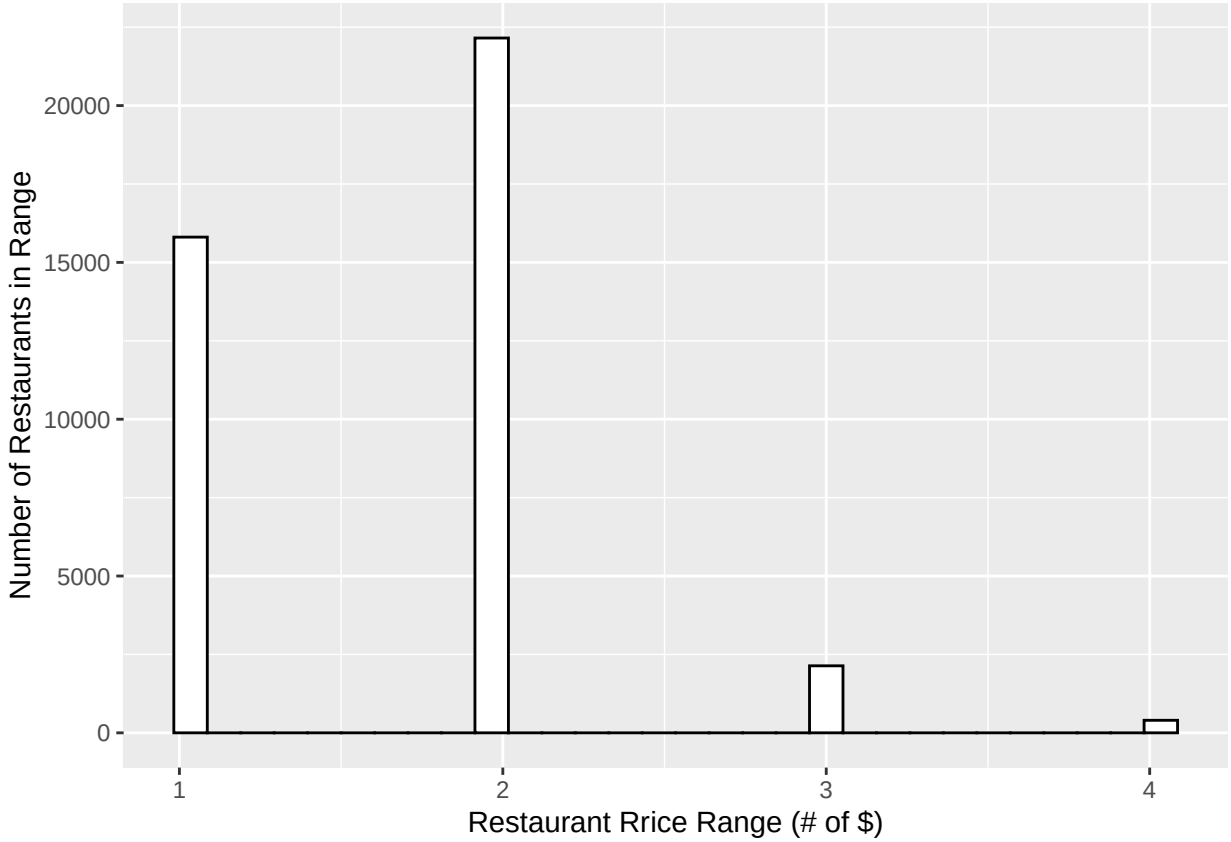
```
rdata.clean$Alc <- 0
lvlone <- c("'none'", "u'none'")
lvltwo <- c("'beer_and_wine'", "u'beer_and_wine'")
lvlthree <- c( "'full_bar'", "u'full_bar'")
rdata.clean$Alc <- ordered(rdata.clean$Alcohol, levels = c(lvlone, lvltwo, lvlthree))
rdata.clean$Alc <- as.numeric(rdata.clean$Alc)
rdata.clean$Alc[rdata.clean$Alc == 2] <- 1
rdata.clean$Alc[rdata.clean$Alc == 3] <- 2
rdata.clean$Alc[rdata.clean$Alc == 4] <- 2
rdata.clean$Alc[rdata.clean$Alc == 5] <- 3
rdata.clean$Alc[rdata.clean$Alc == 6] <- 3
# rdata.clean$Alc <- as.factor(rdata.clean$Alc)

# Doing the same for Attire; had to do it differently for some reason
rdata.clean$RestaurantsAttire[rdata.clean$RestaurantsAttire == "'casual'"] <- 1
rdata.clean$RestaurantsAttire[rdata.clean$RestaurantsAttire == "u'casual'"] <- 1
rdata.clean$RestaurantsAttire[rdata.clean$RestaurantsAttire == "'dressy'"] <- 2
rdata.clean$RestaurantsAttire[rdata.clean$RestaurantsAttire == "u'dressy'"] <- 2
rdata.clean$RestaurantsAttire[rdata.clean$RestaurantsAttire == "'formal'"] <- 3
rdata.clean$RestaurantsAttire[rdata.clean$RestaurantsAttire == "u'formal'"] <- 3
# rdata.clean$RestaurantsAttire <- as.factor(rdata.clean$RestaurantsAttire)

rdata.cleaner <- na.omit(rdata.clean)
d1 <- rdata.cleaner[c(-1 , -2, -3, -11, -12 )]
d1 <- d1 %>% rename(Takeout = RestaurantsTakeOut, PriceRange = RestaurantsPriceRange2, Attire = Restaura
d1$Attire[d1$Attire == "None"] <- 1
d1$PriceRange[d1$PriceRange == "None"] <- NA
d1 <- na.omit(d1)
```

Now we have d1, our final data set which we use to analyze the response & predictor variables, as well as for the construction of our models.

Response Variable Analysis

```
d1$PriceRange <- as.numeric(d1$PriceRange)
ggplot(d1, aes(x=PriceRange)) +
  geom_histogram(color="black", fill="white") +
  xlab("Restaurant Rrice Range (# of $)") +
  ylab("Number of Restaurants in Range")
```

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

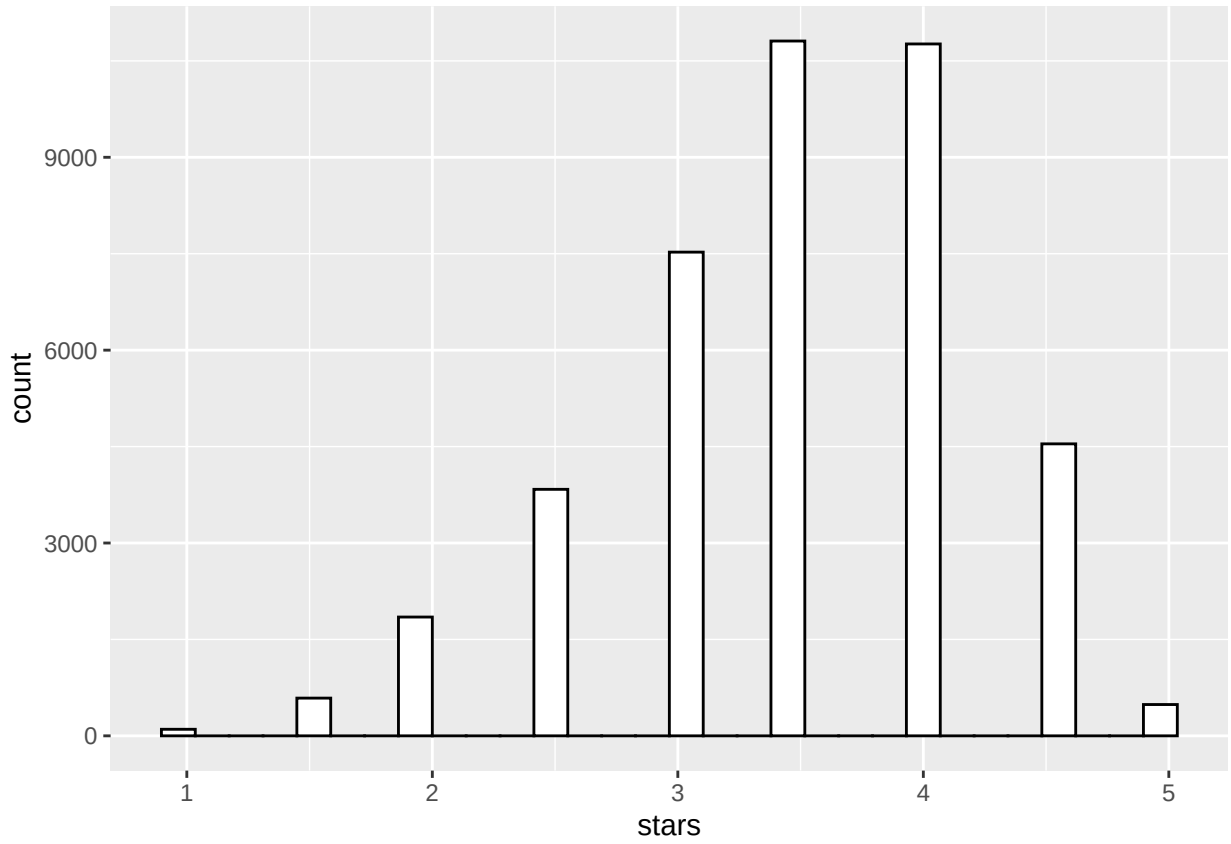


Here we see that our response variable (the price range) is heavily right skewed, with a large concentration of observations around 2 (about 53%). This is worth noting, as we might need to address it in our model construction.

Predictors

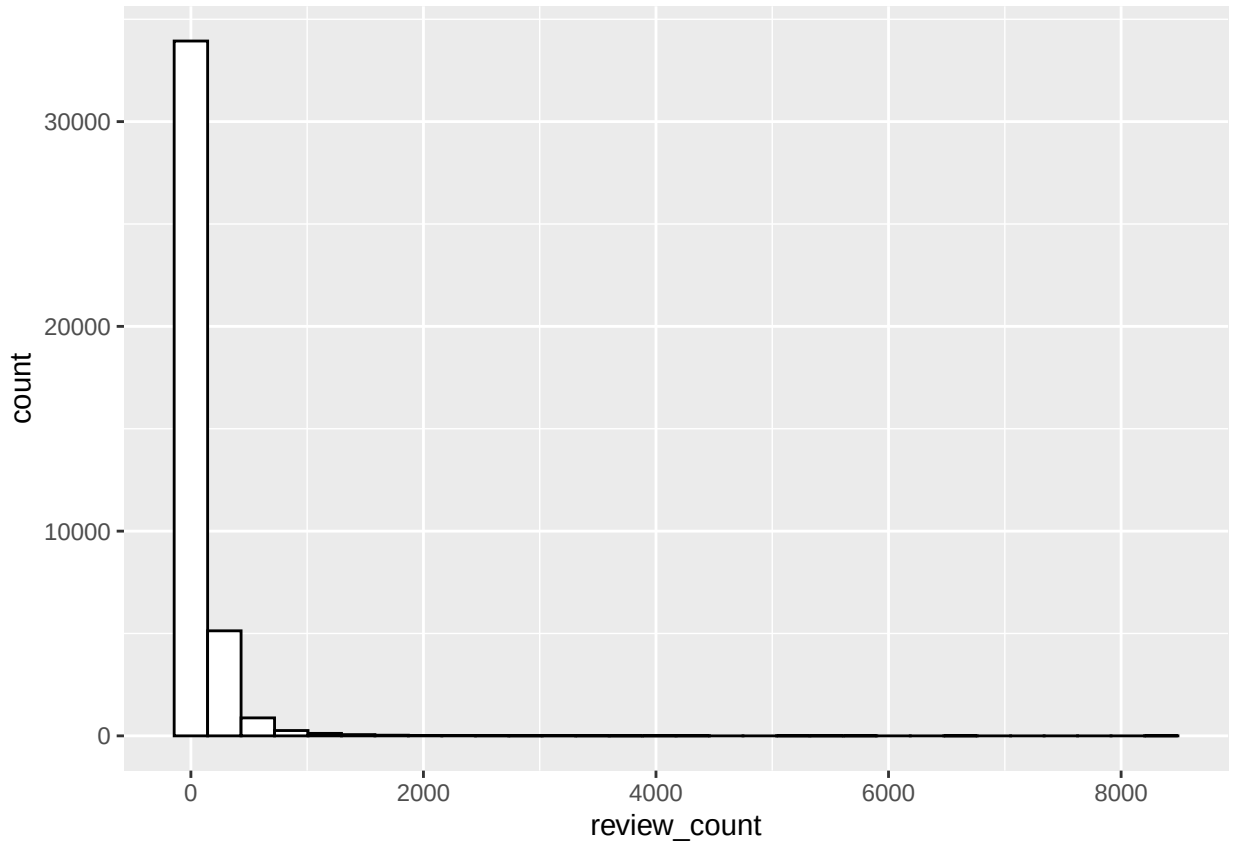
```
## Numerical
ggplot(d1, aes(x=stars)) +
  geom_histogram(color="black", fill="white")
```

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

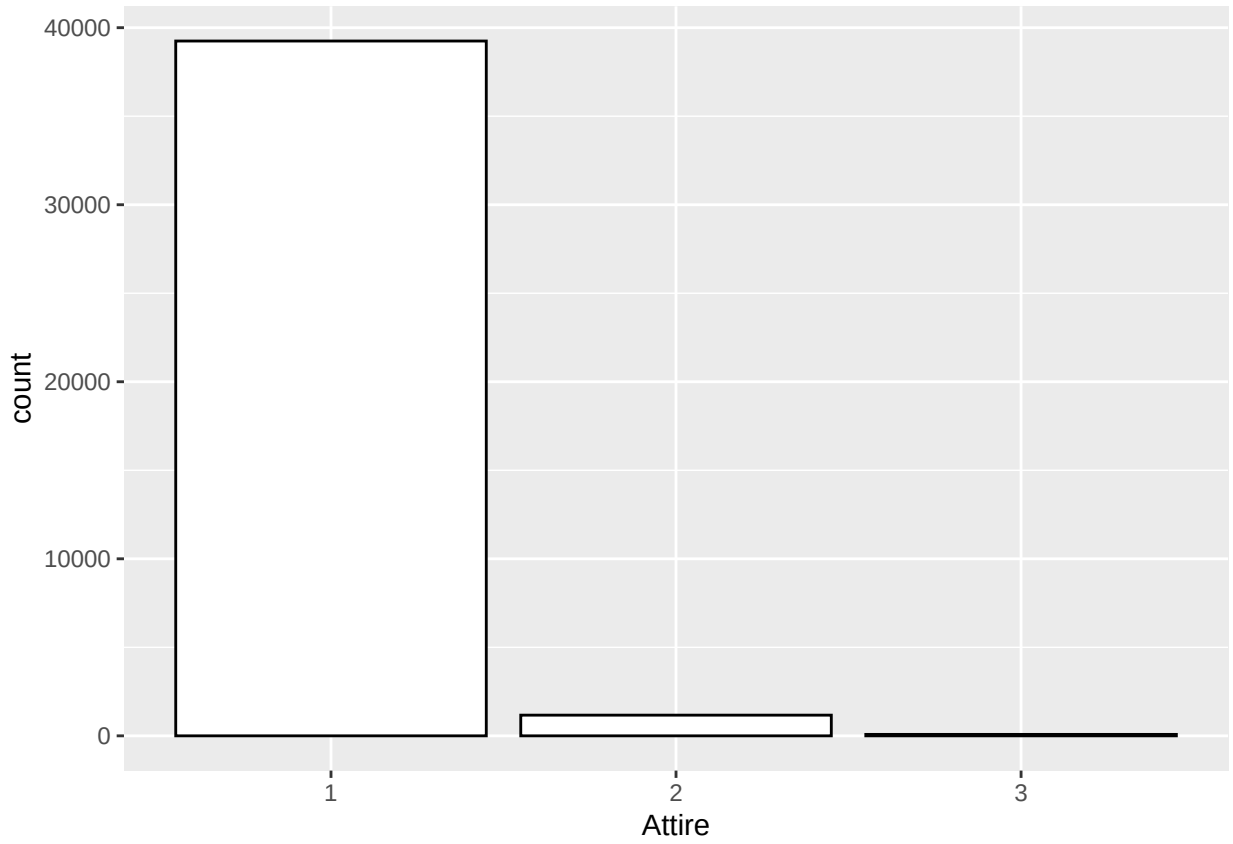



```
ggplot(d1, aes(x=review_count)) +
  geom_histogram(color="black", fill="white")
```

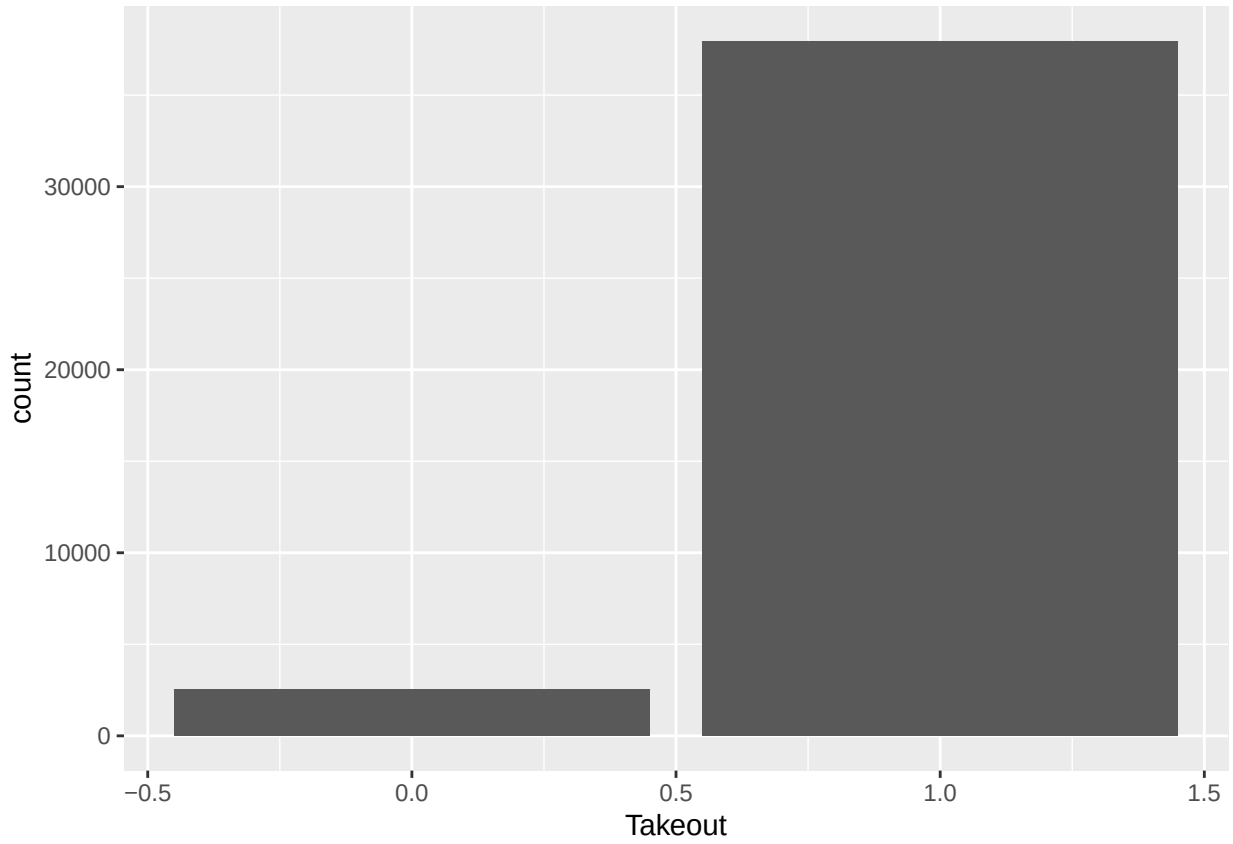
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



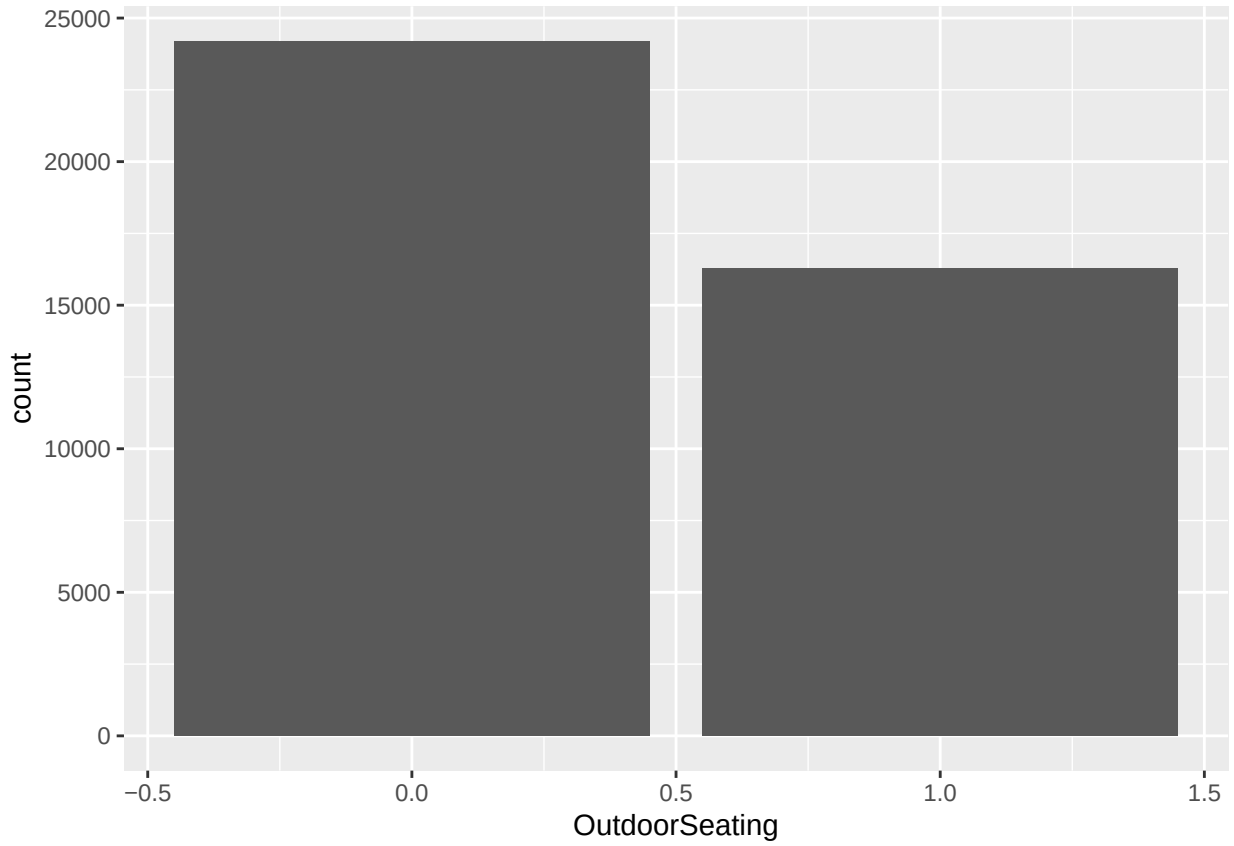

```
## Categorical
ggplot(d1, aes(x=Attire)) +
  geom_bar(color="black", fill="white")
```



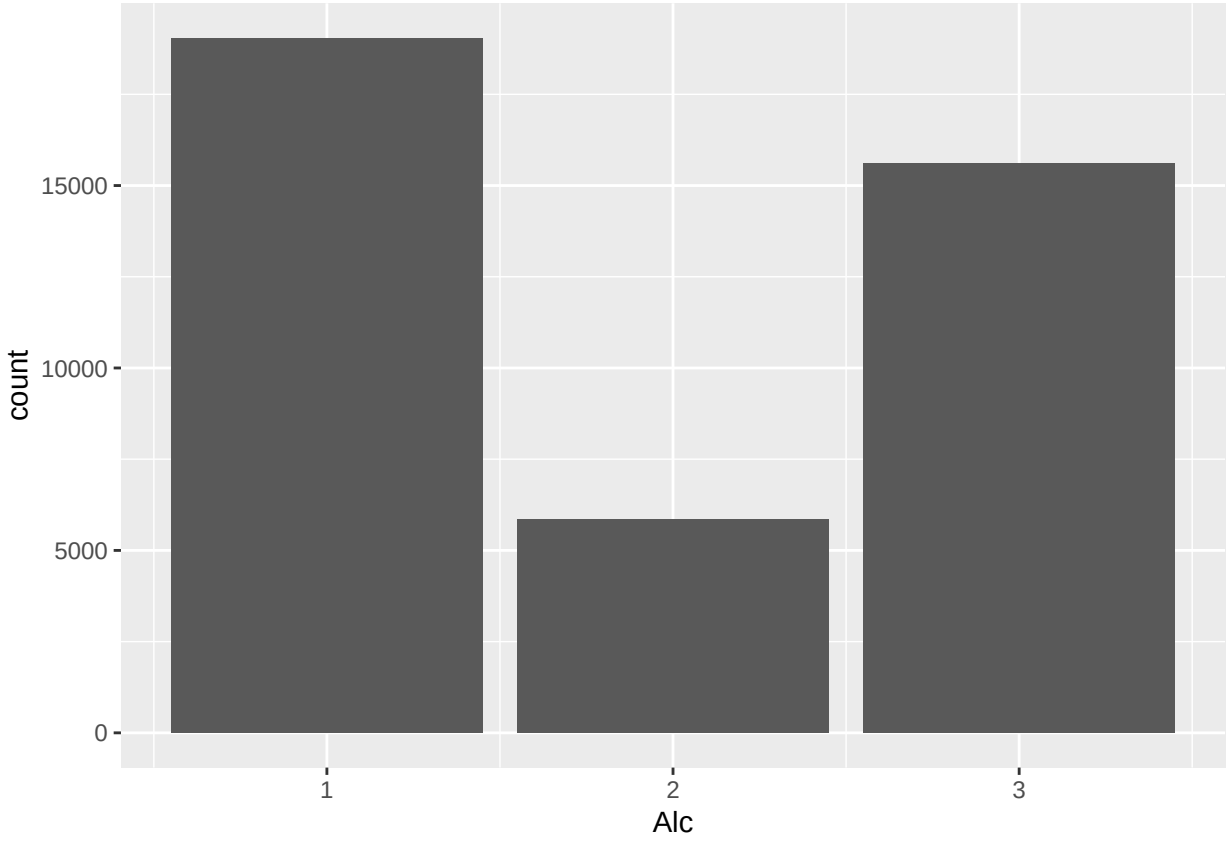

```
ggplot(d1, aes(x=Takeout)) +
  geom_bar()
```




```
ggplot(d1, aes(x=OutdoorSeating)) +
  geom_bar()
```




```
ggplot(d1, aes(x=Alc)) +
  geom_bar()
```



Numerical

There are a couple of interesting things about our predictive variables. The stars variable has a slightly left skewed distribution, with most restaurants falling under the 3.5 - 4 star rating. Review count appears to be centered around a couple of reviews, with very major outliers.

Categorical

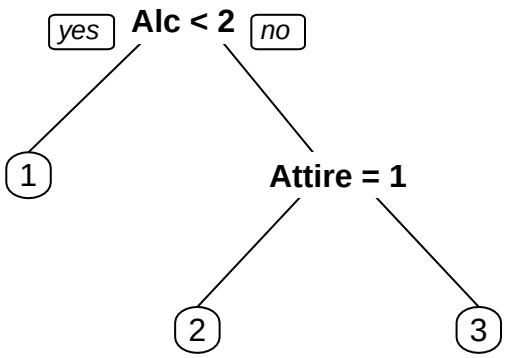
It is difficult to make any meaningful assertions from a cursory look at the categorical, however there are a few things worth noting. Attire is heavily concetrated around the first category, "casual", which is intuitive since most restaurants are casual walk-in locations. Take out is also very heavily concentrated around 1 (which means they do Take Out), which is again intuitive since we'd expect most places to do take out. Outdoor seating has a fairly even split, though there are a larger number of restaurants that do not have it available. And alcohol has a similar amount under the categories of no alcohol & full bar, with a fairly smaller number under "beer and wine".

Modeling

Descriptive Classification Tree

```
library(rpart)
library(rpart.plot)
d1$PriceRange <- as.factor(d1$PriceRange)
```

```
m1 <- rpart(PriceRange ~. , data = d1,
  control = rpart.control(minsplit = 2))
plot1<-prp(m1)
```



Preparing the Data: Training and Test

Before building our supervised learning models, we must first create our training and test sub datasets.

```
s.size <- floor(0.75 * nrow(d1))
set.seed(10)
train.data <- sample(seq_len(nrow(d1)), size = s.size)
train <- d1[train.data, ]
test <- d1[-train.data, ]
```

Linear Model

Though this is a classification exercise, the first model we developed was a linear model with rounding, as follows:

```
train$PriceRange <- as.numeric(train$PriceRange)
test$PriceRange <- as.numeric(test$PriceRange)
m2 <- lm(PriceRange ~.,  data = train)

## Predicting Price
predicted.price <- predict(m2, newdata = test)
predicted.price.rounded <- round(predicted.price, digits = 0)
yp <- predicted.price.rounded

## Misclassification rate:
yt <- test$PriceRange

mcr <- table(yp,yt)

1-sum(diag(mcr))/sum(mcr)
```

```
## [1] 0.2852346
```

A couple of things are worth noting about our process in building this model. First, we aimed to address the issue around the right skewedness of the data by performing some transformations. However, none of these changes significantly improved the predictive strength of the model, thus we chose to not keep any of the transformations. Regarding the explanatory variables we chose, we also tried a variety of models with different variables and opted to choose this one.

With this model we find a misclassification rate of 28.52%. While this modeling method is not necessarily made for this exercise, it is quite robust since it identifies just over 70% of restaurant price ranges correctly. However, we wished to further develop our understanding of the relationship between restaurant prices and the variables, so we continue our explorations with an Ordinal Regression, which should more aptly tackle the classification nature of this problem.

Ordinal Regression

```
library(MASS)
class <- ordered(d1$PriceRange, levels= c("1", "2", "3", "4"))
d1$class <- class
set.seed(20)
olr.train.data <- sample(seq_len(nrow(d1)), size = s.size)
olr.train <- d1[olr.train.data, ]
olr.test <- d1[-olr.train.data, ]
olr.train <- olr.train[c(-6)]
olr.test <- olr.test[c(-6)]
m3 <- polr(class ~. ,  data = olr.train,  Hess = TRUE)
m3
predicted.price3 <- predict(m3, newdata = olr.test)
predicted.price3
```

OLR Misclassification Rate:

```
yp2 <- predicted.price3
yt2 <- olr.test$class
mcr2 <- table(yp2,yt2)
1-sum(diag(mcr2))/sum(mcr2)
```

```
## [1] 0.2777284
```

Discussion

Review the results generated above and sythensize them in the context from which the data originated. What do the results tell your about your original research question? Are there any weaknesses that you see in your analysis? What additional questions would you explore next?

References

"Yelp Open Dataset." Yelp Dataset, 2019, www.yelp.com/dataset.