Data Preparation

Ali Thursland and Mary Helen Wood

12/10/2018

== Setup ==

```
#Setup
library("dplyr")
library("ggplot2")
library("broom")
library("knitr")
library("cowplot")
library("readr")
library("dslabs")
library("varhandle")
library("olsrr")
library("car")
```

== Introduction ==

In producing our analysis, our data required a **lot** of manipulation. Instead of including all of this code in our analysis' .Rmd file, we decided to instead make a separate .Rmd and save the resulting data as a new .csv file, which we then used in our analysis. We did this for simplicity and readability's sake; while not an integral part of our analysis, we think that the time we put into making our data workable was important.

== Reading in our Data ==

```
#Read file
airbnb <- read.csv("https://raw.githubusercontent.com/athursland/STA-210/master/airbnb.csv?fbclid=IwAR0
```

Because not every Airbnb listing in our dataset had reviews, we didn't want the regression model to be affected by listings that lacked reviews. We weren't sure if number_of_reviews was a strong predictor yet, so we chose to make the indicator variable has_reviews, which would be 1 if the listing had any reviews and 0 if the listing has no reviews.

```
#Make new variable has_reviews
airbnb <- airbnb %>%
  mutate(has_reviews = case_when(
    number_of_reviews == 0 ~ 0,
    number_of_reviews > 0 ~ 1)
  )
```

For the rest of our variables, we needed to do something to account for the high quantity of NAs present in our data. Instead of omitting all of these incomplete cases (which would greatly reduce the size of our data set) we chose to code NAs as 0's for all of our numeric variables.

```r
#Make NAs = 0 for security_deposit
airbnb <- airbnb %>%
  mutate(security_deposit = if_else(is.na(security_deposit),0,security_deposit))
```

```r
#Make NAs = 0 for cleaning_fee
airbnb <- airbnb %>%
  mutate(cleaning_fee = if_else(is.na(cleaning_fee),0,cleaning_fee))
```

```r
#Make NAs = 0 for review_scores_rating
airbnb <- airbnb %>%
  mutate(review_scores_rating = case_when(
    is.na(review_scores_rating) ~ 0,
    !is.na(review_scores_rating) ~ review_scores_rating)
    )
```

```r
#Make NAs = 0 for review_scores_accuracy
airbnb <- airbnb %>%
  mutate(review_scores_accuracy = case_when(
    is.na(review_scores_accuracy) ~ 0,
    !is.na(review_scores_accuracy) ~ review_scores_accuracy)
    )
```

```r
#Make NAs = 0 for review_scores_cleanliness
airbnb <- airbnb %>%
  mutate(review_scores_cleanliness = case_when(
    is.na(review_scores_cleanliness) ~ 0,
    !is.na(review_scores_cleanliness) ~ review_scores_cleanliness)
    )
```

```r
#Make NAs = 0 for review_scores_checkin
airbnb <- airbnb %>%
  mutate(review_scores_checkin = case_when(
    is.na(review_scores_checkin) ~ 0,
    !is.na(review_scores_checkin) ~ review_scores_checkin)
    )
```

```r
#Make NAs = 0 for review_scores_communcation
airbnb <- airbnb %>%
  mutate(review_scores_communication = case_when(
    is.na(review_scores_communication) ~ 0,
    !is.na(review_scores_communication) ~ review_scores_communication)
    )
```

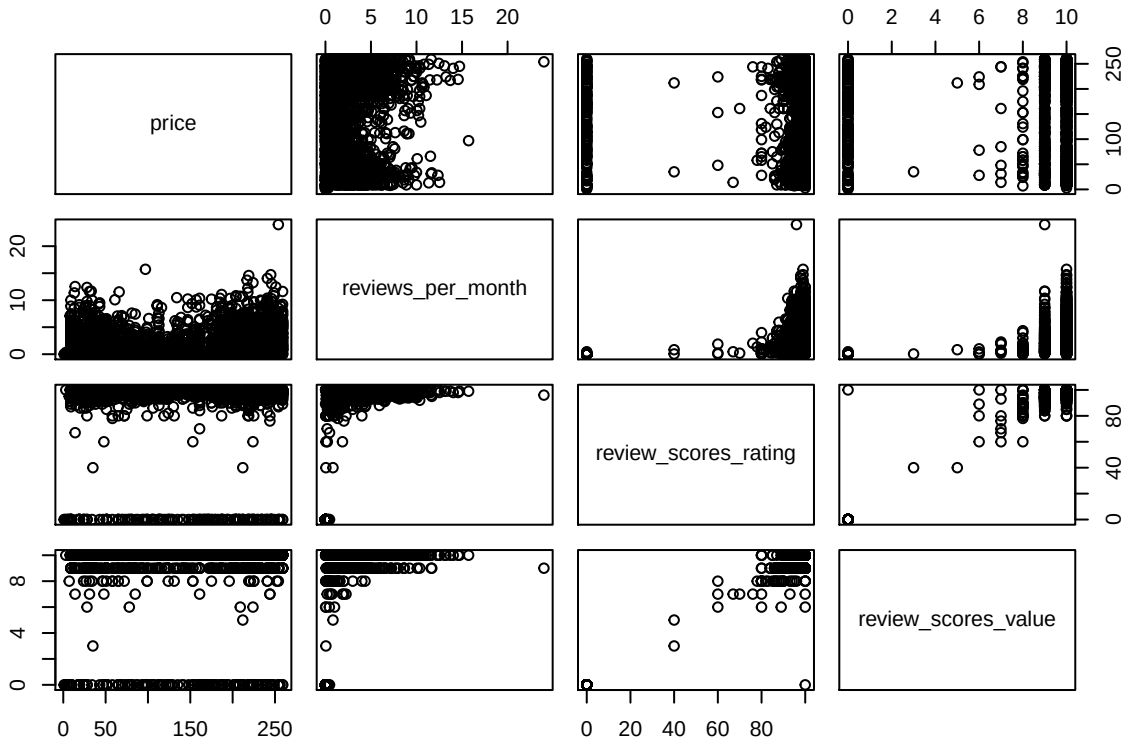
```r
#Make NAs = 0 for review_sores_location
airbnb <- airbnb %>%
  mutate(review_scores_location = case_when(
    is.na(review_scores_location) ~ 0,
    !is.na(review_scores_location) ~ review_scores_location)
    )
```

```r
#Make NAs = 0 for review_scores_value
airbnb <- airbnb %>%
  mutate(review_scores_value = case_when(
    is.na(review_scores_value) ~ 0,
    !is.na(review_scores_value) ~ review_scores_value)
    )
```

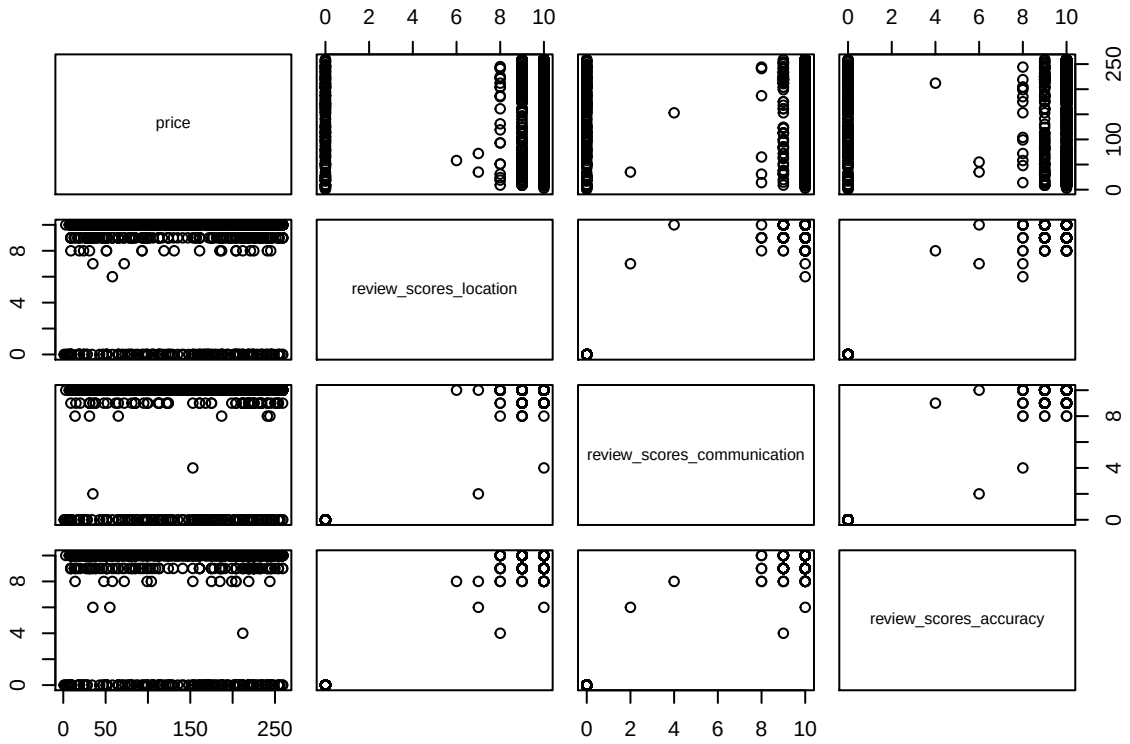
```r
#Make NAs = 0 for reviews_per_month
airbnb <- airbnb %>%
  mutate(reviews_per_month = case_when(
    is.na(reviews_per_month) ~ 0,
    !is.na(reviews_per_month) ~ reviews_per_month)
    )
```

After this, we took a look at a pairs plot of our all of our variables. We saw that there seemed to be a non-linear relationship between our various review_scores variables, number_of_reviews and reviews_per_month, so we decided to log transform each of these. The original and resultant pairs plots can be seen below. While the pairs plots for the log-transformed variables aren't perfect, they look better than the originals.

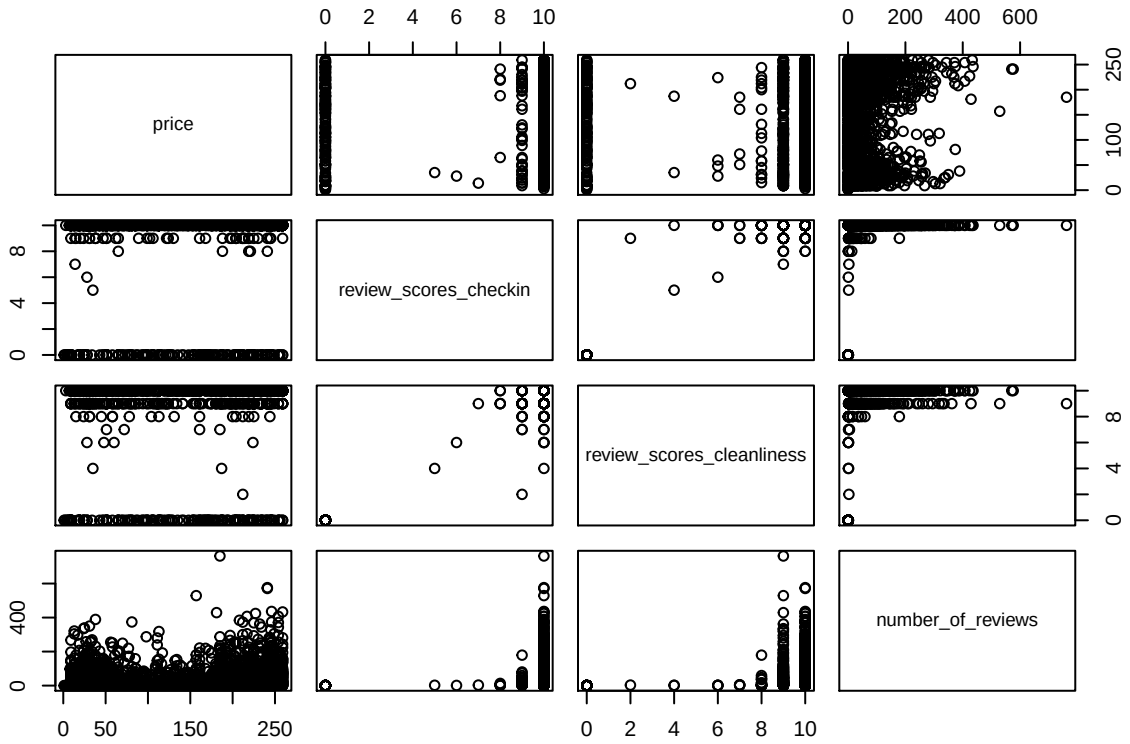
```r
#Original pairs plots before transforming
pairs(data=airbnb, price ~ reviews_per_month + review_scores_rating + review_scores_value)
```




```r
pairs(data=airbnb, price ~ review_scores_location + review_scores_communication + review_scores_accuracy
```



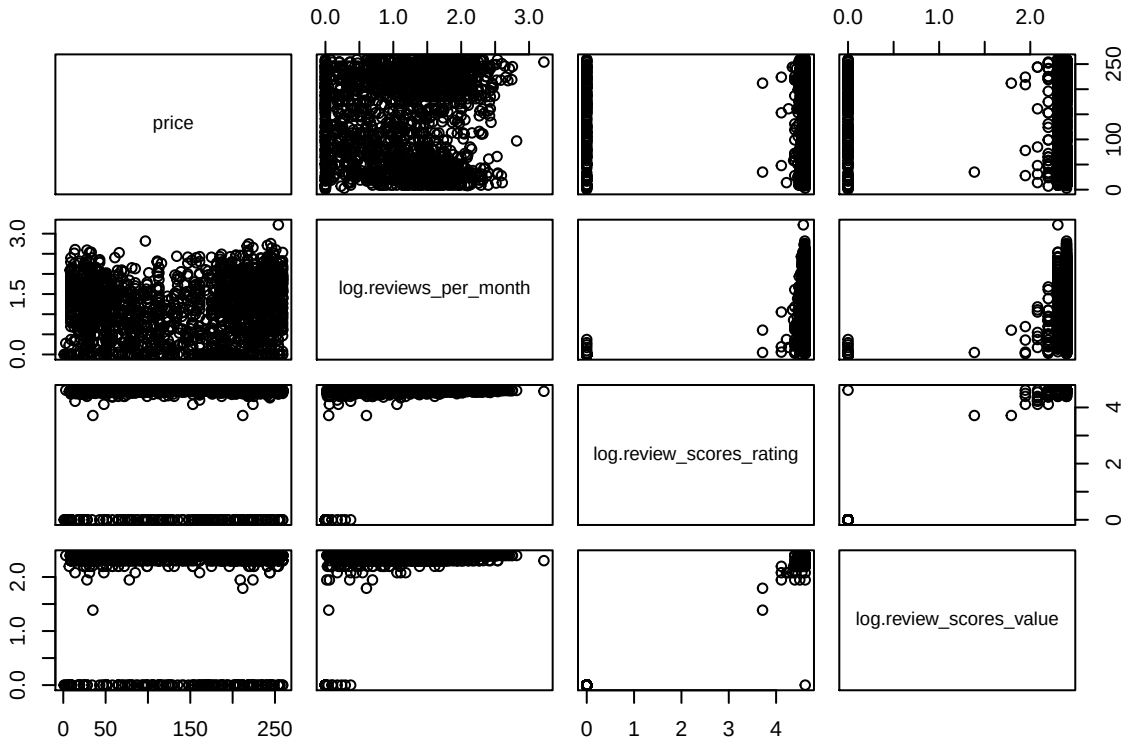

```
pairs(data=airbnb, price ~ review_scores_checkin + review_scores_cleanliness + number_of_reviews)
```



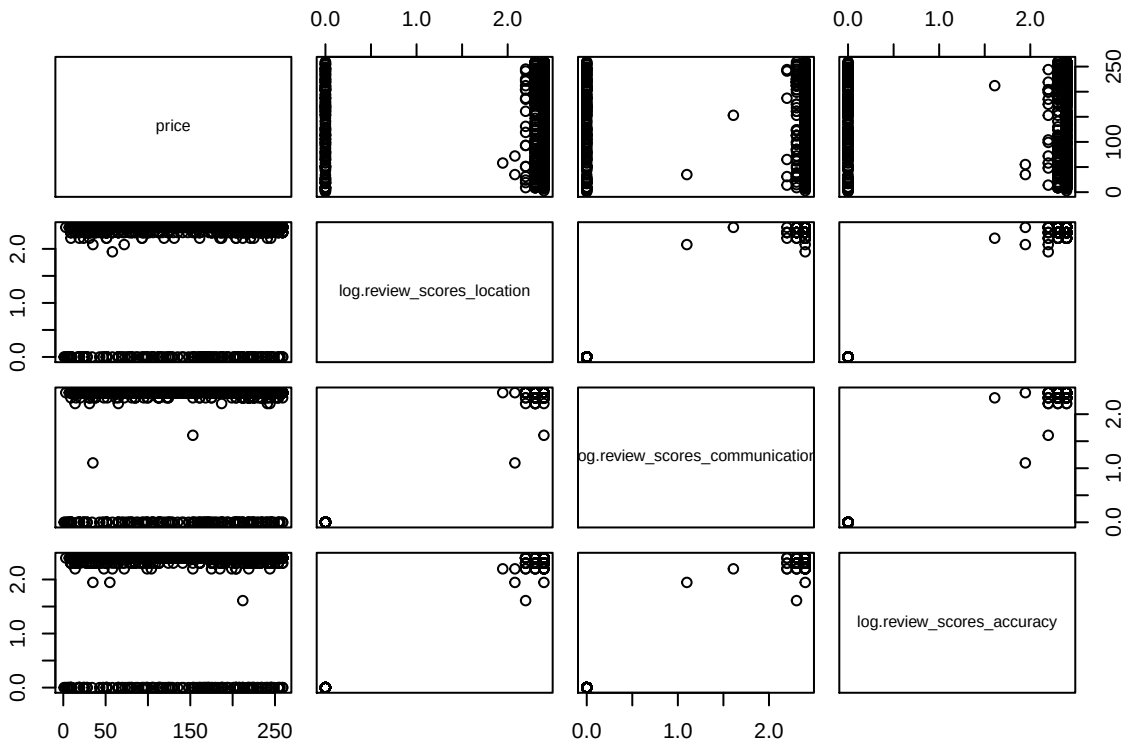

```
#Log transform reviews_per_month
airbnb <- airbnb %>%
  mutate(reviews_per_month.1 = reviews_per_month + 1,
  log.reviews_per_month = log(reviews_per_month.1))
```

```r
#Log transform review_scores_rating
airbnb <- airbnb %>%
  mutate(review_scores_rating.1 = review_scores_rating + 1,
  log.review_scores_rating = log(review_scores_rating.1))

#Log transform reviews_scores_value
airbnb <- airbnb %>%
  mutate(review_score_value.1 = review_scores_value + 1,
  log.review_scores_value = log(review_score_value.1))

#Log transform review_scores_location
airbnb <- airbnb %>%
  mutate(review_scores_location.1 = review_scores_location + 1,
  log.review_scores_location = log(review_scores_location.1))

#Log transform review_scores_communication
airbnb <- airbnb %>%
  mutate(review_scores_communication.1 = review_scores_communication + 1,
  log.review_scores_communication = log(review_scores_communication.1))

#Log transform review_scores_checkin
airbnb <- airbnb %>%
  mutate(review_scores_checkin.1 = review_scores_checkin + 1,
  log.review_scores_checkin = log(review_scores_checkin.1))

#Log transform review_scores_accuracy
airbnb <- airbnb %>%
  mutate(review_scores_accuracy.1 = review_scores_accuracy + 1,
  log.review_scores_accuracy = log(review_scores_accuracy.1))

#Log transform review_scores_cleanliness
airbnb <- airbnb %>%
  mutate(review_scores_cleanliness.1 = review_scores_cleanliness + 1,
  log.review_scores_cleanliness = log(review_scores_cleanliness.1))

#log transform number_of_reviews
airbnb <- airbnb %>%
  mutate(number_of_reviews.1 = number_of_reviews + 1,
  log.number_of_reviews = log(number_of_reviews.1))
```

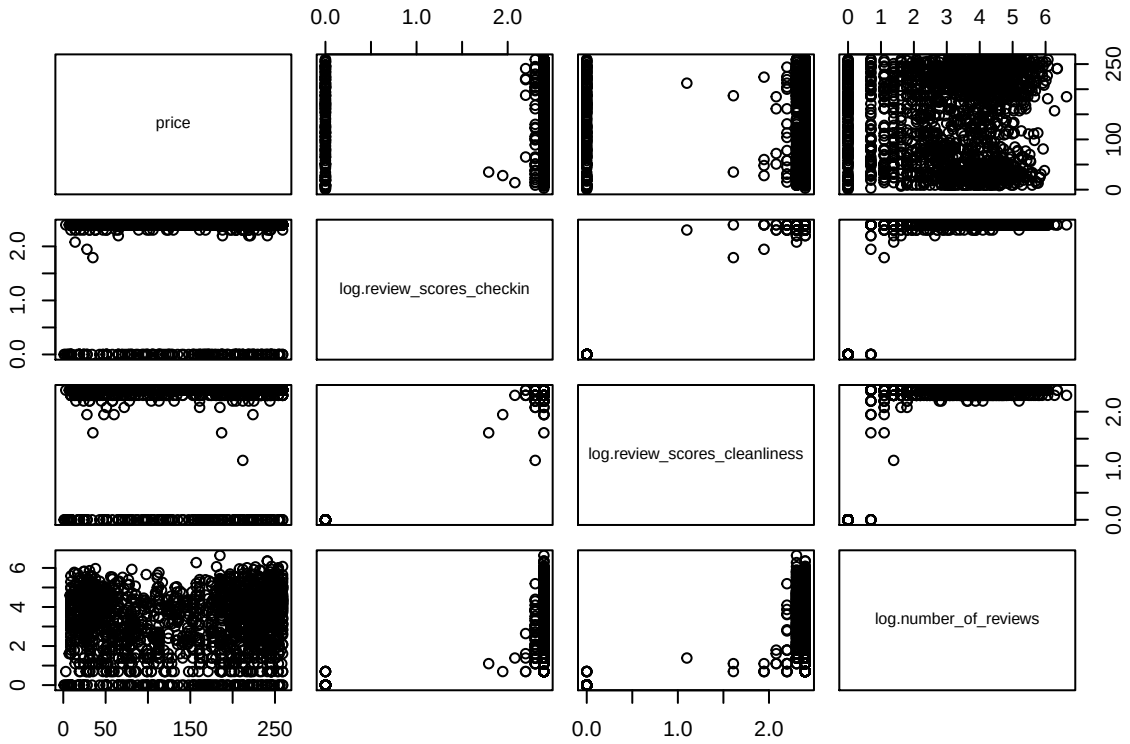
```r
#Pairs plots after transforming
pairs(data=airbnb, price ~ log.reviews_per_month + log.review_scores_rating + log.review_scores_value)
```




```
pairs(data=airbnb, price ~ log.review_scores_location + log.review_scores_communication + log.review_sc
```




```
pairs(data=airbnb, price ~ log.review_scores_checkin + log.review_scores_cleanliness + log.number_of_re
```



Price was originally not considered a numberic variable, so we needed to mutate it to be such.

```
#Make price as numeric
airbnb$price <- as.numeric(as.character(airbnb$price))
```

```
## Warning: NAs introduced by coercion
```

When we looked at the p-values for a full model, we noticed that of the 12 different levels of the variable zipcode, only two were significant - the rest had very high p-values. To deal with this, we decided to make two indicator variables to indicate whether or not the listing is in one of those significant zipcodes, 28801 and 28805. We made these variables factors and set the reference level as "no".

```
#Make zipcode a factor
airbnb$zipcode <- as.factor(airbnb$zipcode)

# Make zipcode indicator for 28801
airbnb <- airbnb %>%
  mutate(inzip28801 = case_when(
    zipcode == 28801 ~ "yes",
    zipcode == 28704 ~ "no",
    zipcode == 28715 ~ "no",
    zipcode == 28732 ~ "no",
    zipcode == 28748 ~ "no",
    zipcode == 28787 ~ "no",
    zipcode == 28803 ~ "no",
    zipcode == 28804 ~ "no",
    zipcode == 28805 ~ "no",
    zipcode == 28806 ~ "no",
    zipcode == 28815 ~ "no",
    zipcode == 29710 ~ "no")
    )
```

```r
#Make zipcode indicator for 28804
airbnb <- airbnb %>%
  mutate(inzip28804 = case_when(
    zipcode == 28801 ~ "no",
    zipcode == 28704 ~ "no",
    zipcode == 28715 ~ "no",
    zipcode == 28732 ~ "no",
    zipcode == 28748 ~ "no",
    zipcode == 28787 ~ "no",
    zipcode == 28803 ~ "no",
    zipcode == 28804 ~ "yes",
    zipcode == 28805 ~ "no",
    zipcode == 28806 ~ "no",
    zipcode == 28815 ~ "no",
    zipcode == 29710 ~ "no")
    )

#Make zipcode indicator for 28804
glimpse(airbnb)
```

```
## Observations: 1,935
## Variables: 47
## $ host_response_rate             <fct> 100%, 100%, N/A, N/A, 100%, N/...
## $ host_is_superhost              <fct> t, t, f, f, f, f, f, f, f, f, ...
## $ host_listings_count            <int> 1, 12, 1, 11, 2, 1, 7, 7, 7, 1...
## $ zipcode                        <fct> 28804, 28801, 28801, 28801, 28...
## $ room_type                      <fct> Private room, Entire home/apt,...
## $ bathrooms                      <dbl> 1.0, 2.0, 1.0, 1.5, 1.0, 2.5, ...
## $ accommodates                   <int> 2, 12, 4, 2, 2, 6, 2, 2, 1, 2,...
## $ beds                           <int> 2, 6, 2, 1, 1, 3, 1, 1, 1, 1, ...
## $ bed_type                       <fct> Real Bed, Real Bed, Real Bed, ...
## $ price                          <dbl> 55, 765, 225, 77, 75, 220, 107...
## $ security_deposit               <dbl> 150, 200, 250, 150, 0, 0, 0, 0...
## $ cleaning_fee                   <dbl> 0, 0, 75, 35, 0, 75, 15, 0, 0,...
## $ guests_included                <dbl> 1, 2, 2, 1, 2, 1, 2, 2, 1, 1, ...
## $ extra_people                   <dbl> 20, 25, 25, 20, 0, 15, 12, 12,...
## $ minimum_nights                 <int> 2, 1, 30, 1, 3, 30, 1, 1, 1, 3...
## $ availability_30                <int> 10, 19, 29, 26, 0, 30, 15, 17,...
## $ number_of_reviews              <int> 116, 90, 0, 48, 83, 3, 179, 46...
## $ review_scores_rating           <dbl> 96, 95, 0, 92, 90, 100, 88, 88...
## $ review_scores_accuracy         <dbl> 10, 9, 0, 9, 9, 9, 9, 9, 10, 1...
## $ review_scores_cleanliness      <dbl> 10, 10, 0, 9, 9, 10, 8, 9, 9, ...
## $ review_scores_checkin          <dbl> 10, 10, 0, 9, 10, 10, 9, 10, 1...
## $ review_scores_communication    <dbl> 10, 10, 0, 9, 10, 9, 9, 9, 10,...
## $ review_scores_location         <dbl> 9, 10, 0, 10, 10, 9, 10, 10, 9...
## $ review_scores_value            <dbl> 10, 9, 0, 9, 9, 10, 9, 9, 10, ...
## $ cancellation_policy            <fct> moderate, super_strict_60, str...
## $ reviews_per_month              <dbl> 1.20, 1.04, 0.00, 0.53, 0.96, ...
## $ has_reviews                    <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, ...
## $ reviews_per_month.1            <dbl> 2.20, 2.04, 1.00, 1.53, 1.96, ...
## $ log.reviews_per_month          <dbl> 0.78845736, 0.71294981, 0.0000...
## $ review_scores_rating.1         <dbl> 97, 96, 1, 93, 91, 101, 89, 89...
## $ log.review_scores_rating       <dbl> 4.574711, 4.564348, 0.000000, ...
## $ review_score_value.1           <dbl> 11, 10, 1, 10, 10, 11, 10, 10,...
```

```
## $ log.review_scores_value         <dbl> 2.397895, 2.302585, 0.000000, ...
## $ review_scores_location.1        <dbl> 10, 11, 1, 11, 11, 10, 11, 11,...
## $ log.review_scores_location      <dbl> 2.302585, 2.397895, 0.000000, ...
## $ review_scores_communication.1   <dbl> 11, 11, 1, 10, 11, 10, 10, 10,...
## $ log.review_scores_communication <dbl> 2.397895, 2.397895, 0.000000, ...
## $ review_scores_checkin.1         <dbl> 11, 11, 1, 10, 11, 11, 10, 11,...
## $ log.review_scores_checkin       <dbl> 2.397895, 2.397895, 0.000000, ...
## $ review_scores_accuracy.1        <dbl> 11, 10, 1, 10, 10, 10, 10, 10,...
## $ log.review_scores_accuracy      <dbl> 2.397895, 2.302585, 0.000000, ...
## $ review_scores_cleanliness.1     <dbl> 11, 11, 1, 10, 10, 11, 9, 10, ...
## $ log.review_scores_cleanliness   <dbl> 2.397895, 2.397895, 0.000000, ...
## $ number_of_reviews.1             <dbl> 117, 91, 1, 49, 84, 4, 180, 47...
## $ log.number_of_reviews           <dbl> 4.762174, 4.510860, 0.000000, ...
## $ inzip28801                      <chr> "no", "yes", "yes", "yes", "ye...
## $ inzip28804                      <chr> "yes", "no", "no", "no", "no",...
```

```
airbnb$inzip28801 <- as.factor(airbnb$inzip28801)
airbnb$inzip28804 <- as.factor(airbnb$inzip28804)

#Reference Levels
airbnb$inzip28801 <- relevel(airbnb$inzip28801, ref="no")
airbnb$inzip28804<- relevel(airbnb$inzip28804, ref="no")
```

Finally, we decided to omit any remaining incomplete cases we had left. At this point in our data preparation process, we only had 35 out of our original 1935 observations that were incomplete cases. After looking at the data, we concluded that any NAs we still had left were the result of random errors in the data collection process. We were comfortable in omitting these few observations for the sake of our model.

```
#Omit all observations with NAs
airbnb <- airbnb[complete.cases(airbnb),]
```

Finally, after building a model and checking the assumptions, we saw that the plot of our residuals vs. the predicted values had a strange shape. After consulting with Dr. Tackett, we thought that the best move would be to log transform our response variable, price, and use that in our final model.

```
#Log transform price
airbnb <- airbnb %>% mutate(logprice = log(price))
```

== Saving Data ==

To prevent cluttering our analysis .Rmd file, we decided to save our edited data set and use that moving forward. That way, we can knit our document much faster and don't have to worry about damaging our data set.

```
#Write CSV in R
write_csv(airbnb, "finalairbnb.csv")
```