EDAV Final Project Report

Sabrina Li (zl2728) & Joey Gu (zg2319)

12/8/2018

Introduction

Yelp has become a great helper in our daily life, as people are searching for restaurants everyday using this application. By choosing a level of stars and leaving a review, users can evaluate their experiences in different restaurants and share it with others. Therefore, it makes our lives more convenient when looking for great restaurants. Our group is interested in the comparisons of restaurants in terms of different regions and level of stars, and the relationship between review and quality of restaurants.

Our group members, Sabrina Li and Joey Gu worked on this project together. As our dataset contains 5 distinct files, we first separated them into two and analyzed them individually. Then we combined and compared our findings and finished this report together.

Description of Data

The dataset we used is from Yelp Open Dataset https://www.yelp.com/dataset. The official Yelp dataset contains information of businesses, reviews and users. The structure of this dataset is as follows:

1. business.json: a dataset that includes the name, location, stars, number of reviews, attributes, and open hours of a business.

2. review.json: a text dataset that contains ratings, time, content, and votes of a review for a business.

3. user.json: a user dataset including user id, name, review count, friends, and average rating of each user.

The dataset has three other files that contains the information about check-in time, tips, and photos for each business, which were not introduced in this report since they have not been used in our analysis. As the original dataset is in JSON format, we preprocessed the data using Python for textual analysis. The code for data cleaning is on our GitHub page: https://github.com/sabrinali18/EDAV_Yelp.git.

Analysis of Data Quality

Read in the data

```
yelp_business0 <- stream_in(file("yelp_academic_dataset_business.json"), verbose = FALSE)
yelp_business <- flatten(yelp_business0)
```

The dataset above contains the information about businesses in Yelp, in which we find these features probably have more patterns to plot: state, categories, star, and review_count. Thus the following analysis of this dataset is based on these 4 features.

Data cleaning

```
yelp_business <- as_data_frame(yelp_business)
yelp_business <- yelp_business %>%
  select(-starts_with("hours"), -starts_with("attribute")) %>%
  filter(str_detect(categories, "Restaurant"))
yelp_business$categories <- strsplit(yelp_business$categories, split = ",")
yelp_unn <- unnest(yelp_business, categories)
```

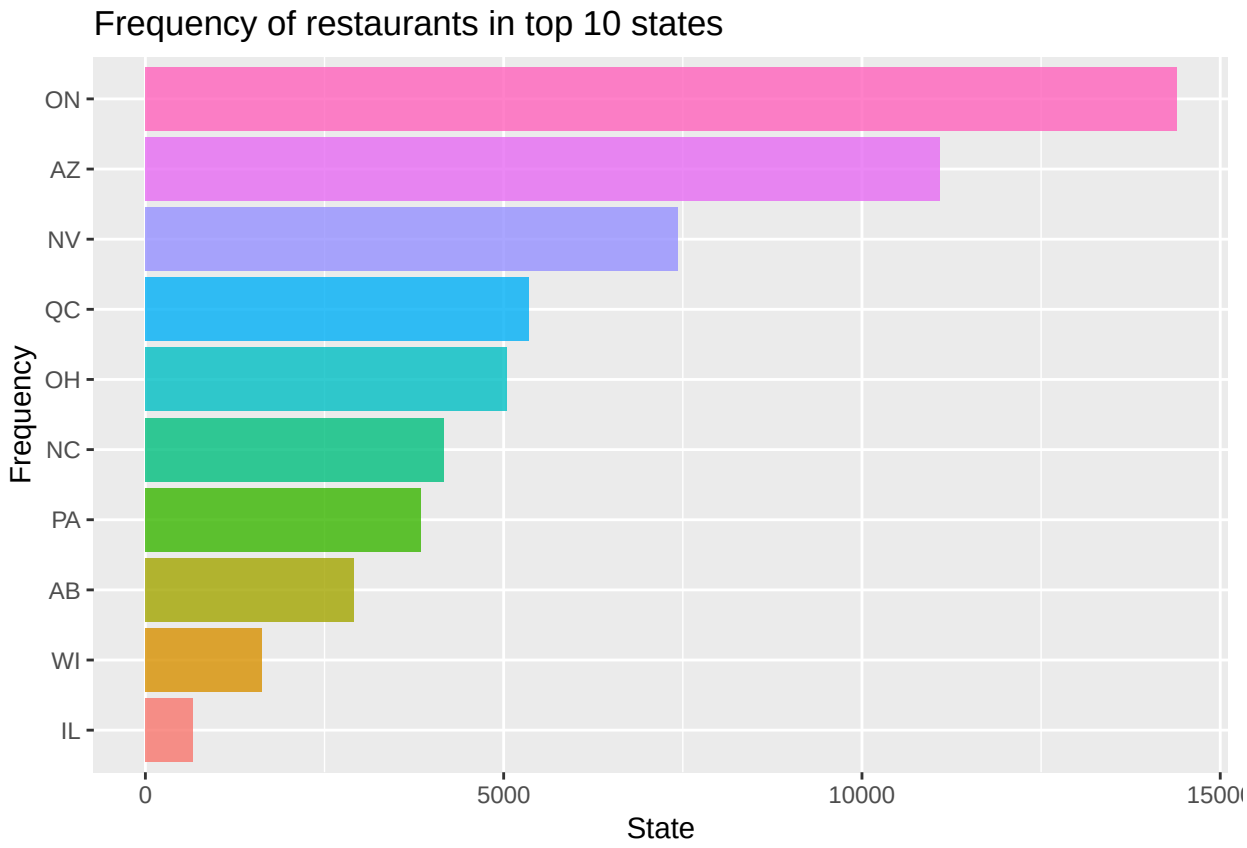
During the cleaning process, we got rid of features of little use such as attributes and hours, which makes the original dataset clearer. And we unnested categories and stored the new dataset as yelp_unn: as one restaurant belongs to different categories, it's necessary to do so in order to further analyze categories.

Checking Data Quality

Firstly, we want to have a general look at 4 critical factors in our analysis: state, category, star, and review count.

State

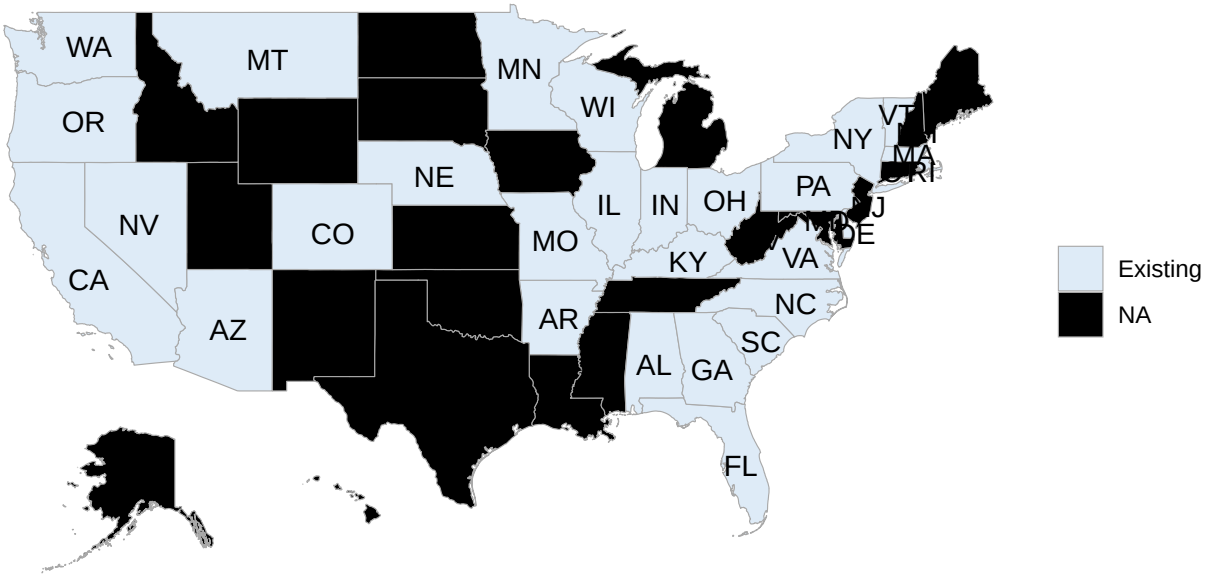
```
yelp_business$state <- as.factor(yelp_business$state)
res_state <- yelp_business %>%
  group_by(state) %>%
  summarize(freq = n()) %>%
  arrange(desc(freq))
res_state$state <- as.character(res_state$state)
res_state <- res_state[is.na(as.numeric(res_state$state)),]
res_state <- res_state[nchar(res_state$state) == 2,]
res_state_top <- res_state[c(1:10),]
res_state_top$state <- factor(res_state_top$state,
                              levels = res_state_top$state[order(res_state_top$freq)])
res_state_top_fig <- ggplot(res_state_top,
                            aes(x = state, y = freq, fill = cut(freq, 100))) +
  geom_histogram(stat = "identity", show.legend = FALSE, alpha = 0.8) +
  coord_flip() +
  xlab("Frequency") + ylab("State") +
  ggtitle("Frequency of restaurants in top 10 states")
res_state_top_fig
```



As the dataset we obtained is a subset of the whole dataset on Yelp, there are missing information about businesses in some of the states. For this reason, we only show the 10 of the states that are the most representative in the list. The figure above shows the rank in a decreasing order. To further verify whether state is complete or not, we generated the following figure:

```
state_map <- yelp_business0 %>% group_by(state) %>%
  summarise(freq = n()) %>% mutate(is_existing = "Existing")
state_map <- state_map[,c("state", "is_existing")]
colnames(state_map) <- c("region", "value")
state_map$region <- tolower(abbr2state(as.character(state_map$region)))
state_map <- state_map[!is.na(state_map$region),]
state_choropleth(state_map, title = "Data existing or not in each area")
```

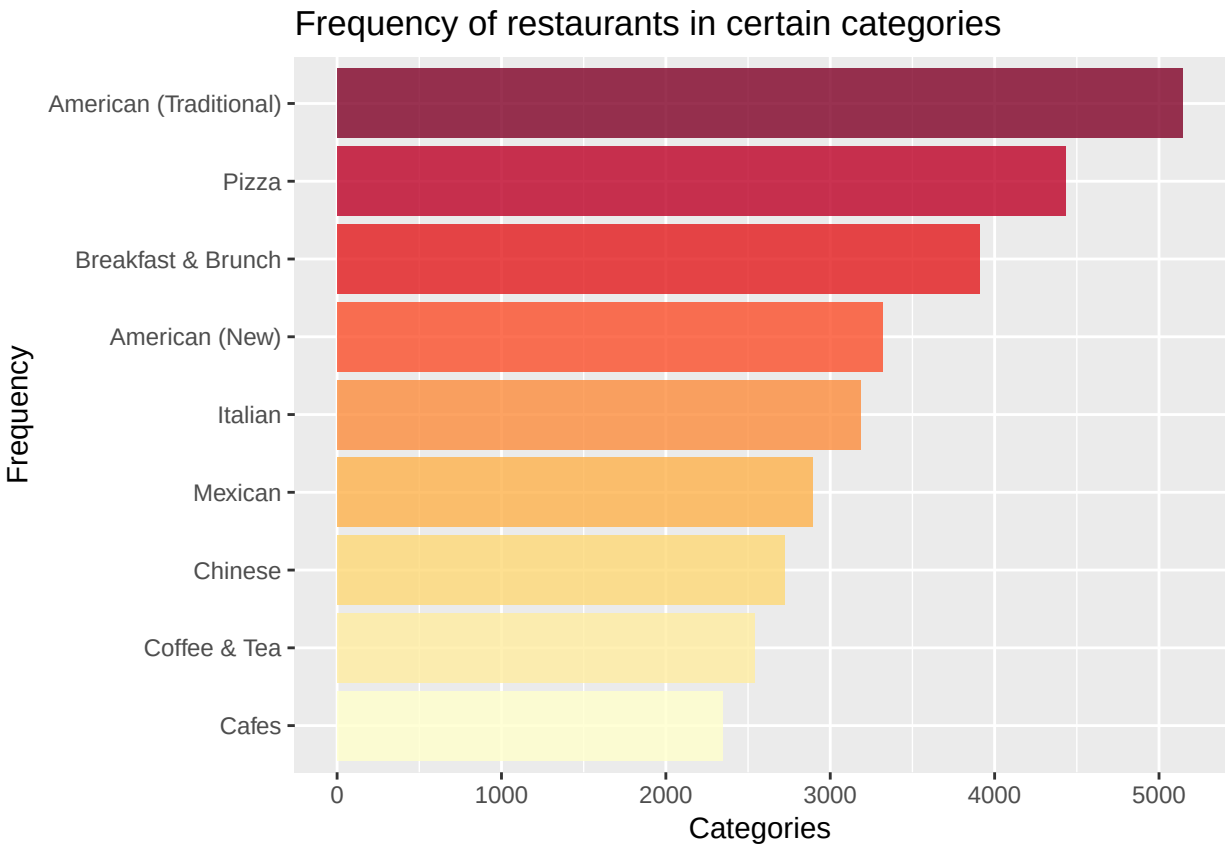
Data existing or not in each area



This map shows that there are almost half of the states in the United States missing in the dataset, which indicates that our analysis would be more convincing if we explore comparisons among specific states.

Category

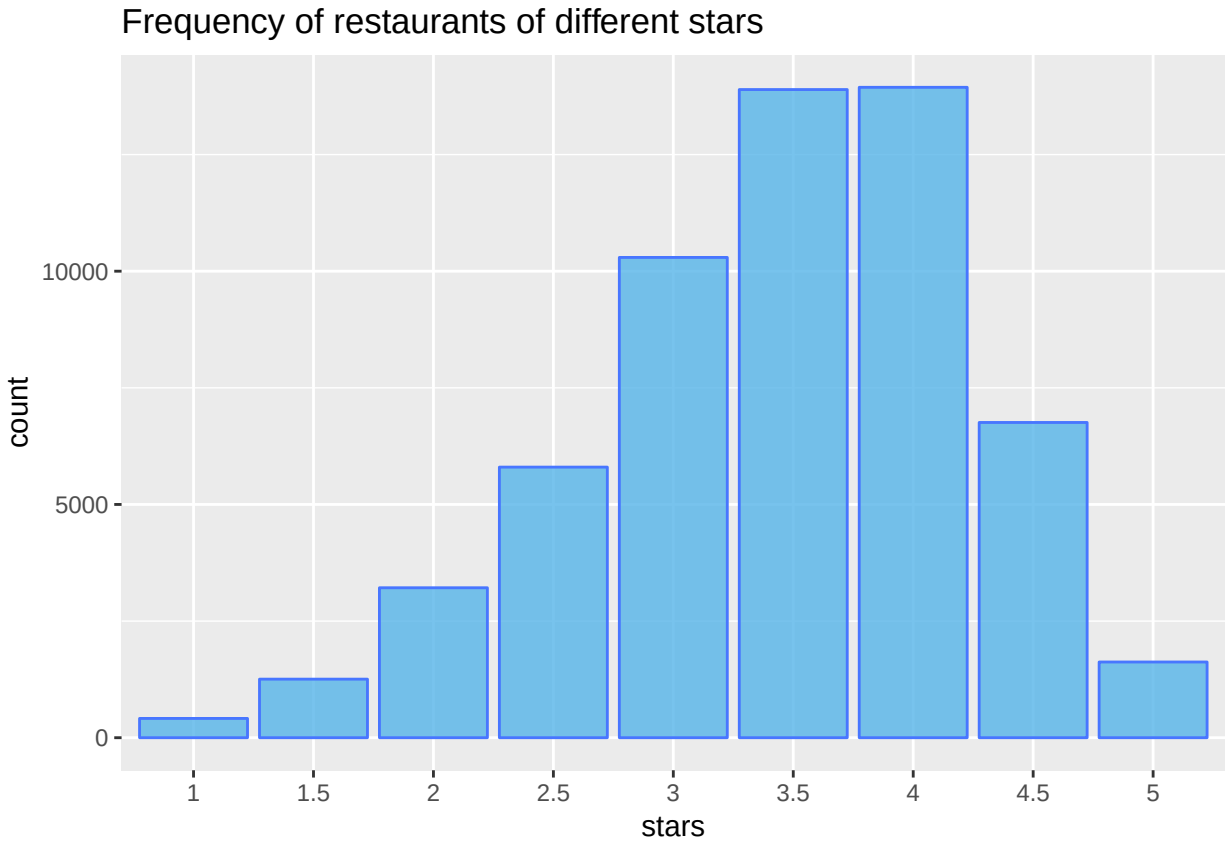
```
res_cate <- yelp_unn %>%
 count(categories) %>%
 arrange(desc(n))
res_cate_top <- res_cate[c(7, 9, 10, 12:17),]
res_cate_top$categories <- factor(res_cate_top$categories,
                                  levels = res_cate_top$categories[order(res_cate_top$n)])
res_categ_top_fig <- ggplot(res_cate_top,
                            aes(x = categories, y = n, fill = categories)) +
  geom_histogram(stat = "identity", show.legend = FALSE, alpha = 0.8) +
  coord_flip() +
  scale_fill_brewer(palette="YlOrRd") +
  xlab("Frequency") + ylab("Categories") +
  ggtitle("Frequency of restaurants in certain categories")
res_categ_top_fig
```



For the category factor, we found that the distribution of categories is pretty dispersed: there are general categories such as Restaurant, Food, Nightlife with rather high frequency; while different restaurants may have their own unique tags, such as Cigar Bars, Landscape Architects, Signmaking, and so on, which are with very low frequency. This happens because the categories themselves are not exclusive to each other and has a pattern of hierarchy. Hence, we chose categories that people frequenty use when describing a restaurant to have a visual sense of the distribution of different categories.

Stars

```
yelp_business$stars <- as.factor(yelp_business$stars)
ggplot(yelp_business, aes(stars)) +
  geom_bar(fill = "#56B4E9", color = "royalblue1", alpha = 0.8) +
  ggtitle("Frequency of restaurants of different stars")
```

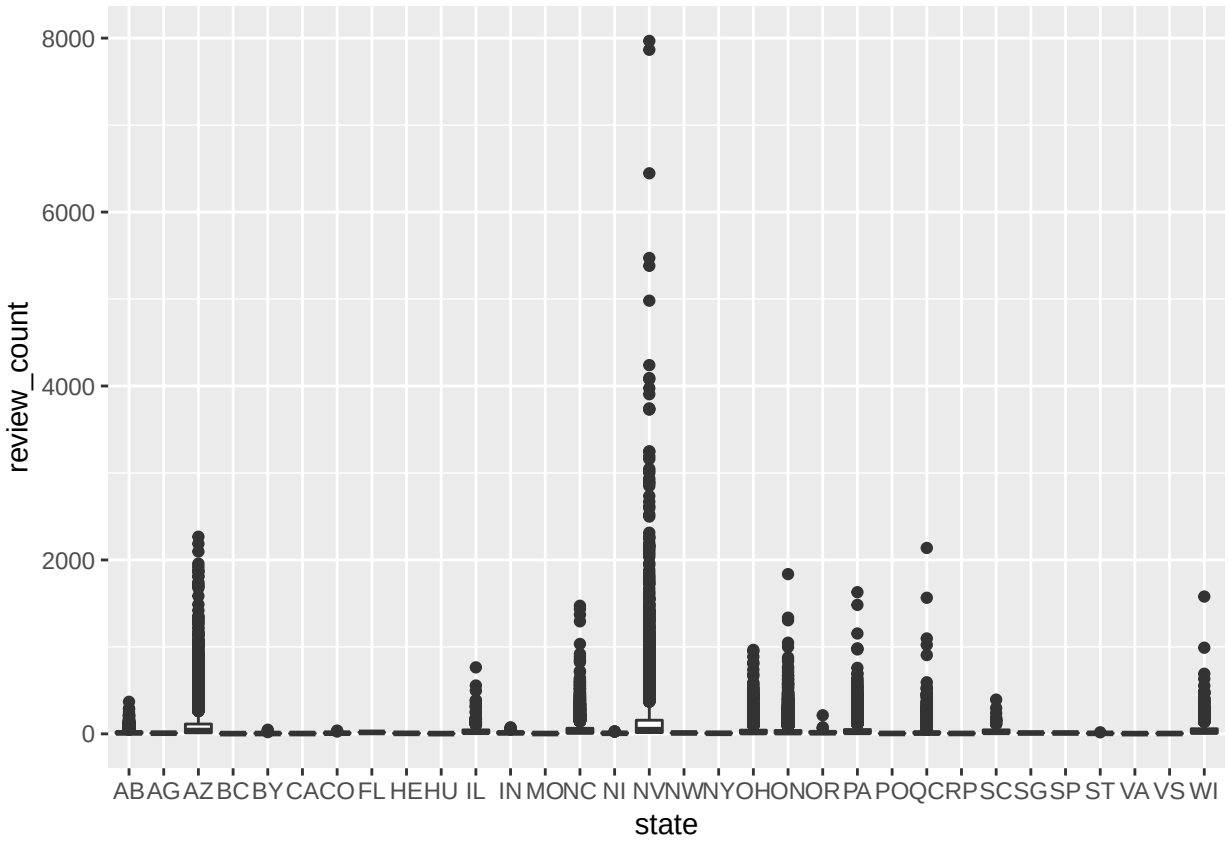


The figure above shows the distribution of stars. It indicates that most retaurants has a rating of stars between 3 stars to 4.5 stars, which agrees with our intuition.

Review count

Here we give a plot of distribution of review count according to states.

```
yelp_business$state <- as.character(yelp_business$state)
yelp_business1 <- yelp_business[is.na(as.numeric(yelp_business$state)),]
yelp_business1 <- yelp_business1[nchar(yelp_business1$state) == 2,]
ggplot(yelp_business1, aes(x = state, y = review_count)) +
  geom_boxplot()
```



Here we can see that for most states, there're very few review count outlier, which is probably because of their small data size. To make the distribution clearer, we will give some further analysis on certain states with more information.

Main Analysis (Exploratory Data Analysis)

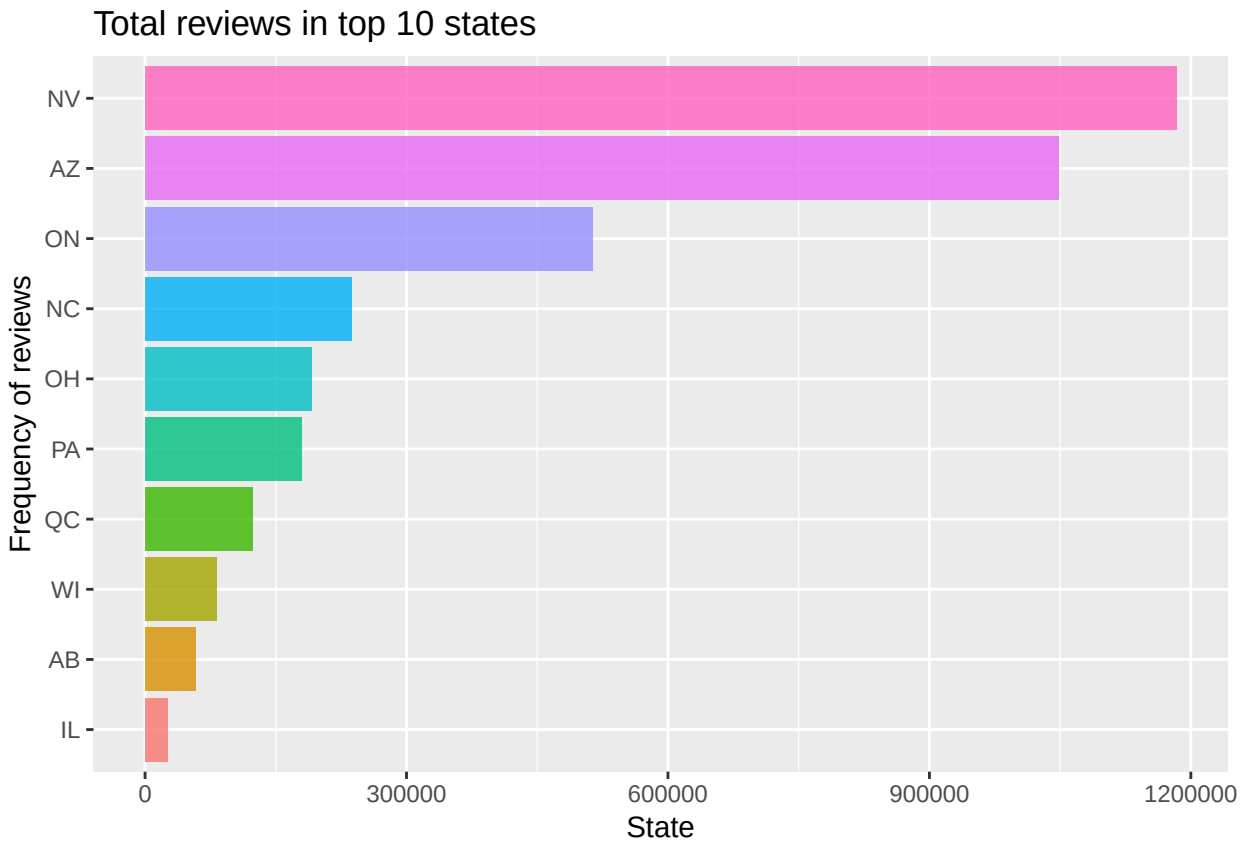
Review count (total)

This section introduces the relationship between total review count and 3 factors: state, category, and star.

State

```
rev_total_state <- yelp_business %>%
  filter(str_detect(categories, "Restaurant")) %>%
  group_by(state) %>%
  summarize(freq = n(), total = sum(review_count)) %>%
  mutate(average = round(total/freq, 4)) %>%
  arrange(desc(total))
rev_total_state_top <- rev_total_state[c(1:10),]
rev_total_state_top$state <- factor(rev_total_state_top$state,
                                    levels = rev_total_state_top$state[order(rev_total_state_top$total)]
rev_total_state_top_fig <- ggplot(rev_total_state_top,
                                  aes(x = state, y = total,
                                      fill = cut(total, 100))) +
  geom_histogram(stat = "identity", show.legend = FALSE, alpha = 0.8) + coord_flip() +
  xlab("Frequency of reviews") + ylab("State") +
```

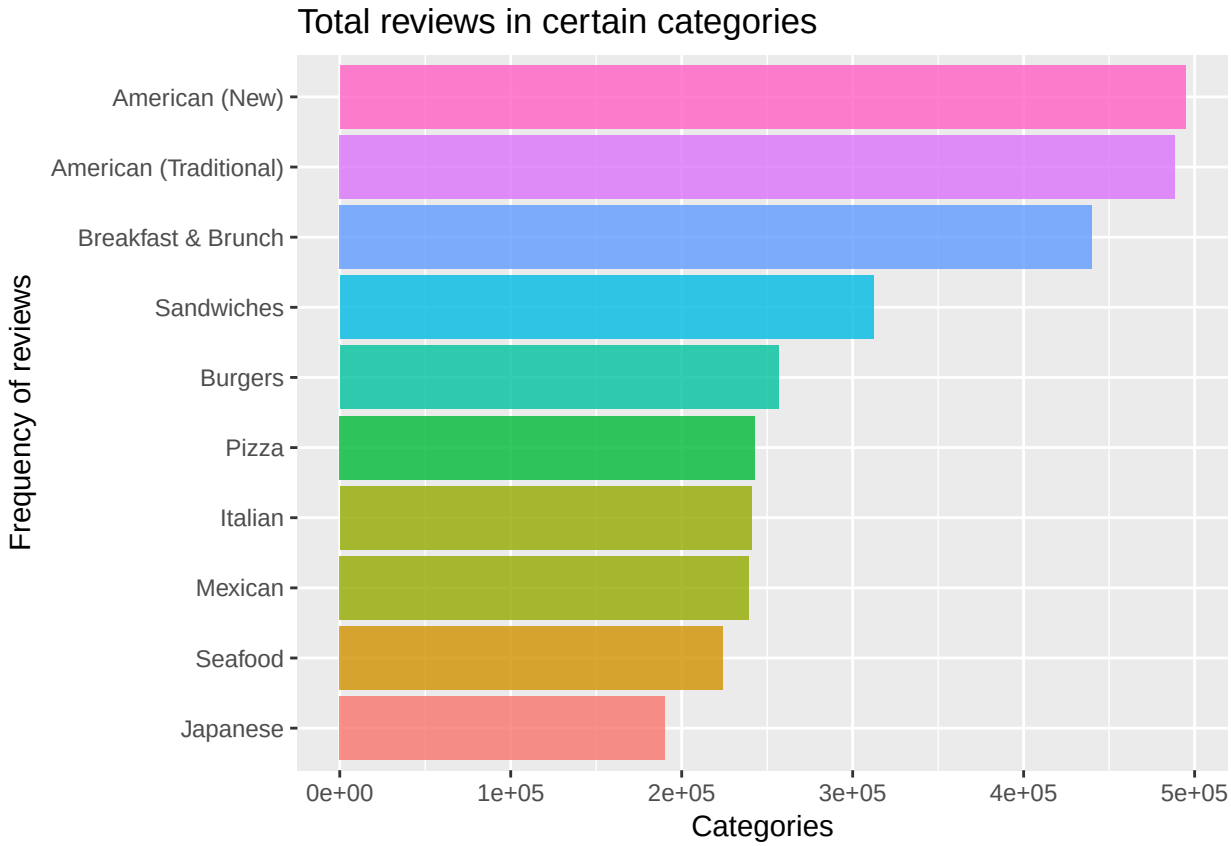
```
  ggtitle("Total reviews in top 10 states")
rev_total_state_top_fig
```



For the same reason we have mentioned in our data quality section, we only plot states with top 10 frequency. We can see that NV has the largest number of reviews, about 1.2 millions; and AZ is the second, about 1.05 millions.

Category

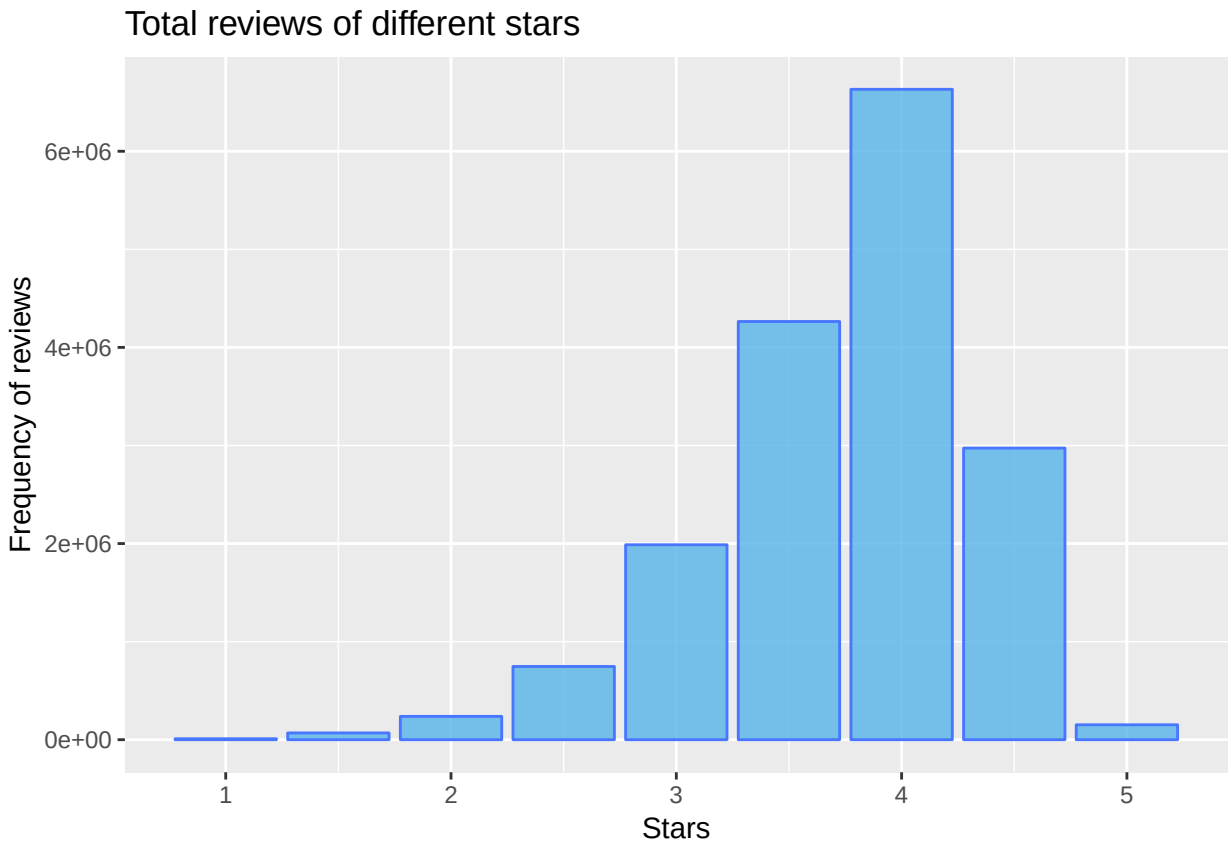
```
rev_total_cate <- yelp_unn %>%
  group_by(categories) %>%
  summarize(freq = n(), total = sum(review_count)) %>%
  mutate(average = round(total/freq, 4)) %>%
  arrange(desc(total))
rev_total_cate_top <- rev_total_cate[c(6:15),]
rev_total_cate_top$categories <- factor(rev_total_cate_top$categories,
                        levels = rev_total_cate_top$categories[order(rev_total_cate_top$total)])
rev_total_cate_top_fig <- ggplot(rev_total_cate_top,
                        aes(x = categories, y = total, fill = cut(total, 100))) +
  geom_histogram(stat = "identity", show.legend = FALSE, alpha = 0.8) +
  coord_flip() +
  xlab("Frequency of reviews") + ylab("Categories") +
  ggtitle("Total reviews in certain categories")
rev_total_cate_top_fig
```



For this plot, we ignore categories like Restaurants, Nightlife, Foods which are too general, and then plot categories with top 10 review counts. It's unsurprising to see American foods take the most part, and the only Asian foods in the list is Japanese.

Stars

```
rev_total_star <- yelp_unn %>%
  group_by(stars) %>%
  summarize(freq = n(), total = sum(review_count)) %>%
  mutate(average = round(total/freq, 4)) %>%
  arrange(desc(total))
rev_total_star_fig <- ggplot(rev_total_star,
                             aes(x = stars, y = total)) +
  geom_histogram(fill = "#56B4E9", color = "royalblue1",
                 stat = "identity", show.legend = FALSE,
                 alpha = 0.8) +
  ylab("Frequency of reviews") + xlab("Stars") +
  ggtitle("Total reviews of different stars")
rev_total_star_fig
```



The histogram above shows that 4 stars restaurants receive far more reviews than other levels, over 6 millions. And restaurants under 2 stars and at 5 stars receive fewer reviews, probably because of the less amount of restaurants than other levels. In addition, the gap between different levels are larger than that of restaurants frequency, which means amount of restaurants is not the only reason for more reviews.

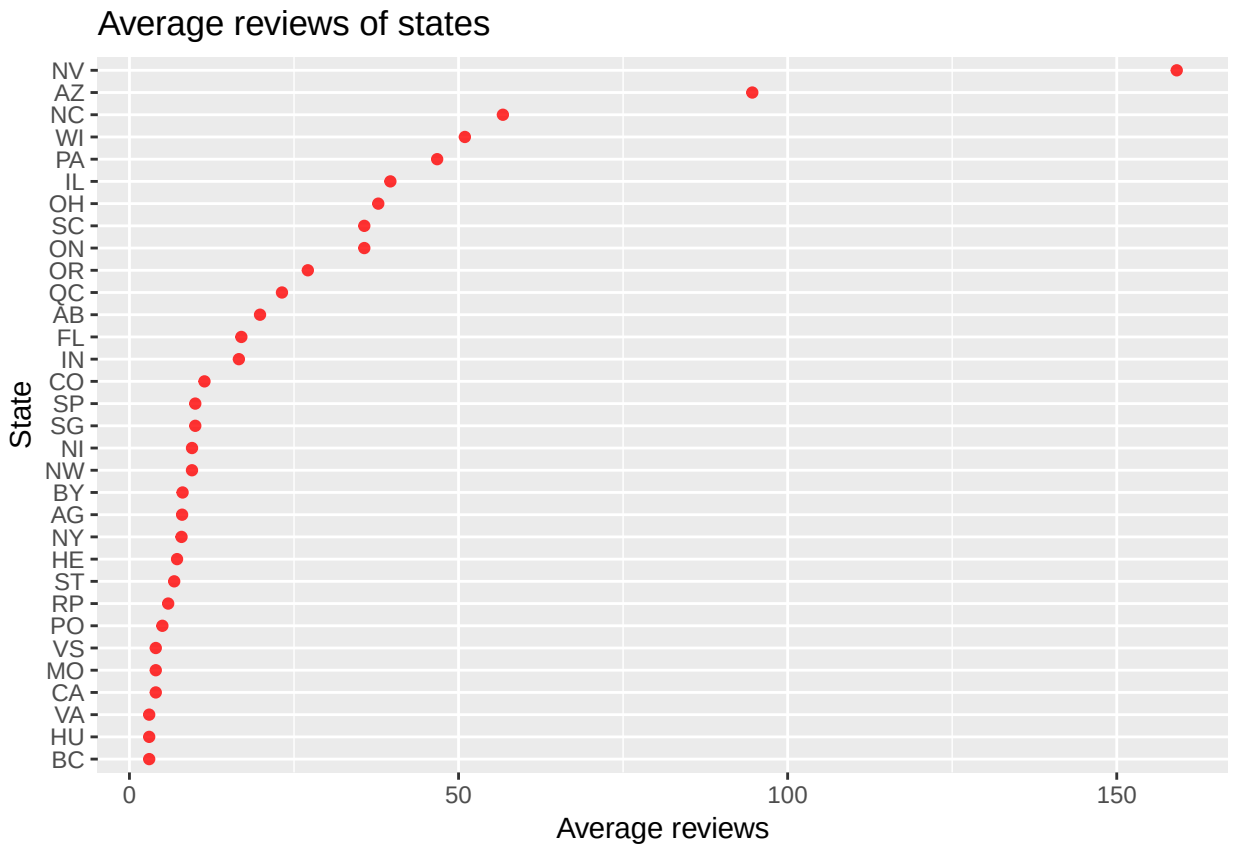
Review count (average)

In this section, we introduce the relationships between average review count and 3 factors: state, category and star. And we will find a different pattern from total review. And we can regard average review as popularity of this level: people are more willing to give evaluations to them.

State

```
rev_ave_state <- yelp_business %>%
  filter(str_detect(categories, "Restaurant")) %>%
  group_by(state) %>%
  summarize(freq = n(), total = sum(review_count)) %>%
  mutate(average = round(total/freq, 4)) %>%
  arrange(desc(average))
rev_ave_state$state <- as.character(rev_ave_state$state)
rev_ave_state <- rev_ave_state[is.na(as.numeric(rev_ave_state$state)),]
rev_ave_state <- rev_ave_state[nchar(rev_ave_state$state) == 2,]
rev_ave_state$state <- factor(rev_ave_state$state,
                      levels = rev_ave_state$state[order(rev_ave_state$average)])
rev_ave_state_fig <- ggplot(rev_ave_state,
                      aes(x = average, y = state)) +
```

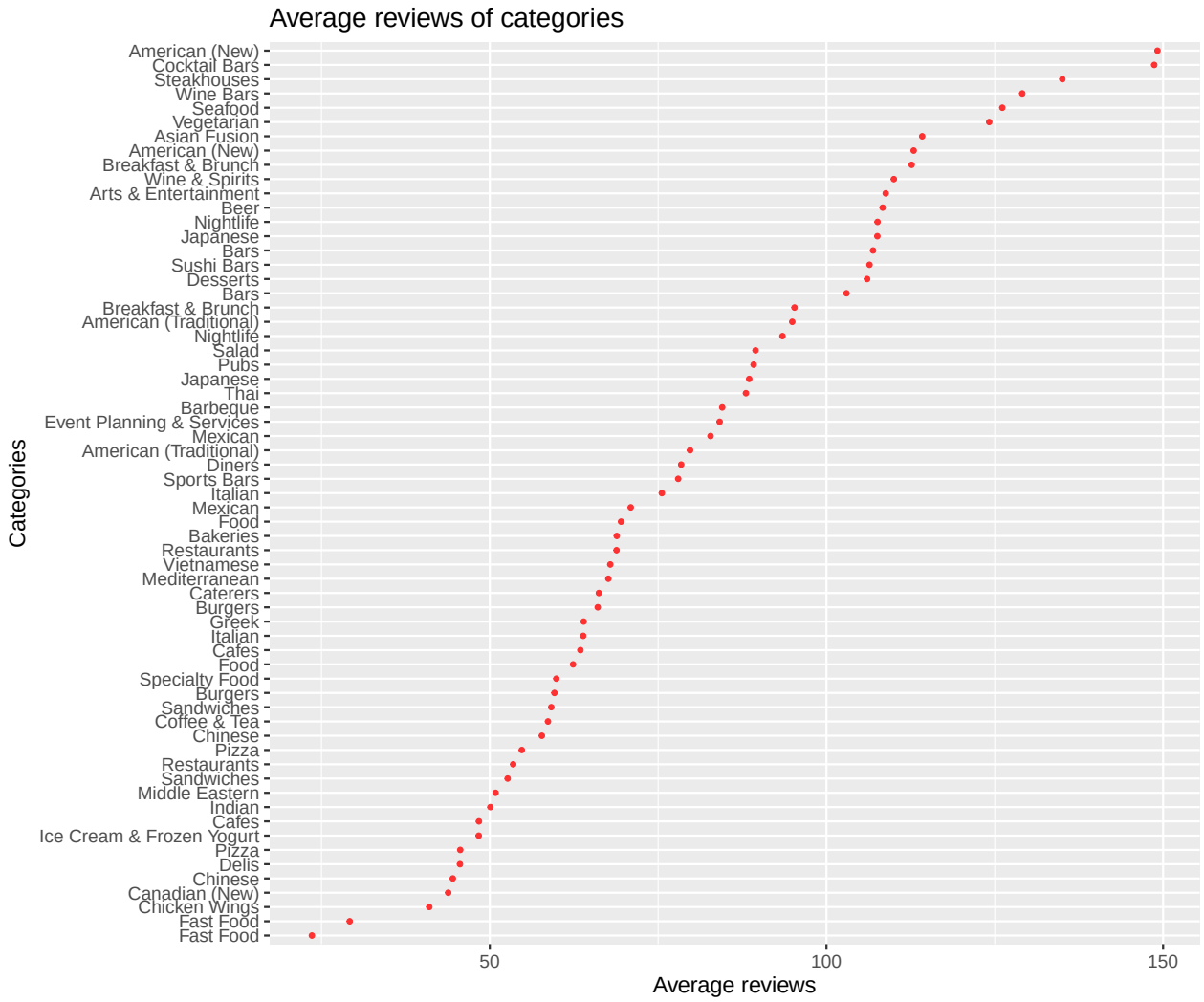
```
  geom_point(color = "red", alpha = 0.8) +
  xlab("Average reviews") + ylab("State") +
  ggtitle("Average reviews of states")
rev_ave_state_fig
```



From the Cleveland plot above, we can still see NV has more average reviews than other states. Because of the bias of dataset, we can only conclude that the information about NV is more complete than other states, rather than people in NV are more likely to give reviews.

Category

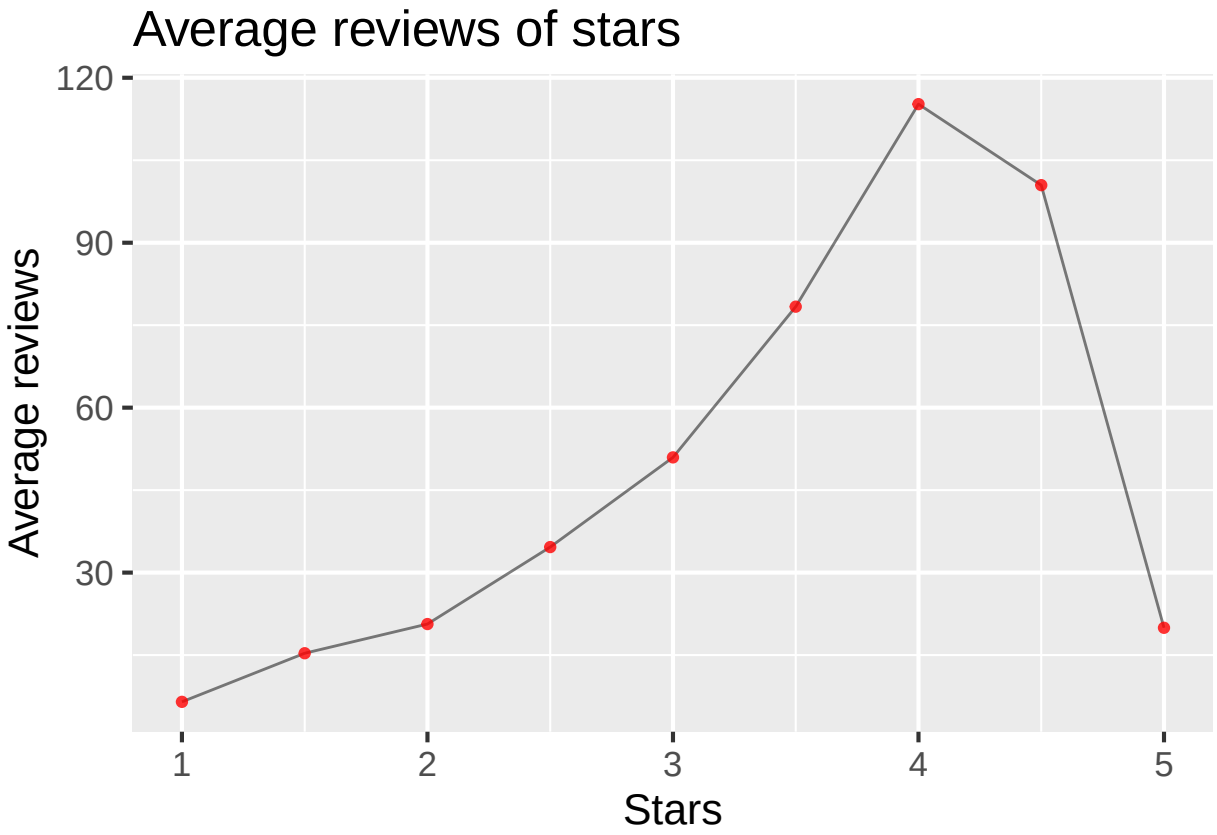
```
rev_ave_cate <- yelp_unn %>%
  group_by(categories) %>%
  summarize(freq = n(), total = sum(review_count)) %>%
  mutate(average = round(total/freq, 4)) %>%
  filter(freq > 700) %>%
  arrange(desc(average))
rev_ave_cate$categories <- factor(rev_ave_cate$categories,
                    levels = rev_ave_cate$categories[order(rev_ave_cate$average)])
rev_ave_cate_fig <- ggplot(rev_ave_cate, aes(x = average, y = categories)) +
  geom_point(color = "red", alpha = 0.8) +
  theme_grey(16) + xlab("Average reviews") +
  ylab("Categories") +
  ggtitle("Average reviews of categories")
rev_ave_cate_fig
```



In this figure, to only take a look of larger categories, we set retaurants frequency to be larger than 700, which can help to filter categories that are too specific. The plot indicates that American (New) and Cocktail Bar are more popular than other categories.

Stars

```
rev_ave_star <- yelp_unn %>%
  group_by(stars) %>%
  summarize(freq = n(), total = sum(review_count)) %>%
  mutate(average = round(total/freq, 4))
rev_ave_star_fig <- ggplot(rev_ave_star,
                      aes(x = stars, y = average)) +
  geom_line(alpha = 0.5) + geom_point(color = "red", alpha = 0.8) +
  theme_grey(16) + ylab("Average reviews") + xlab("Stars") +
  ggtitle("Average reviews of stars")
rev_ave_star_fig
```



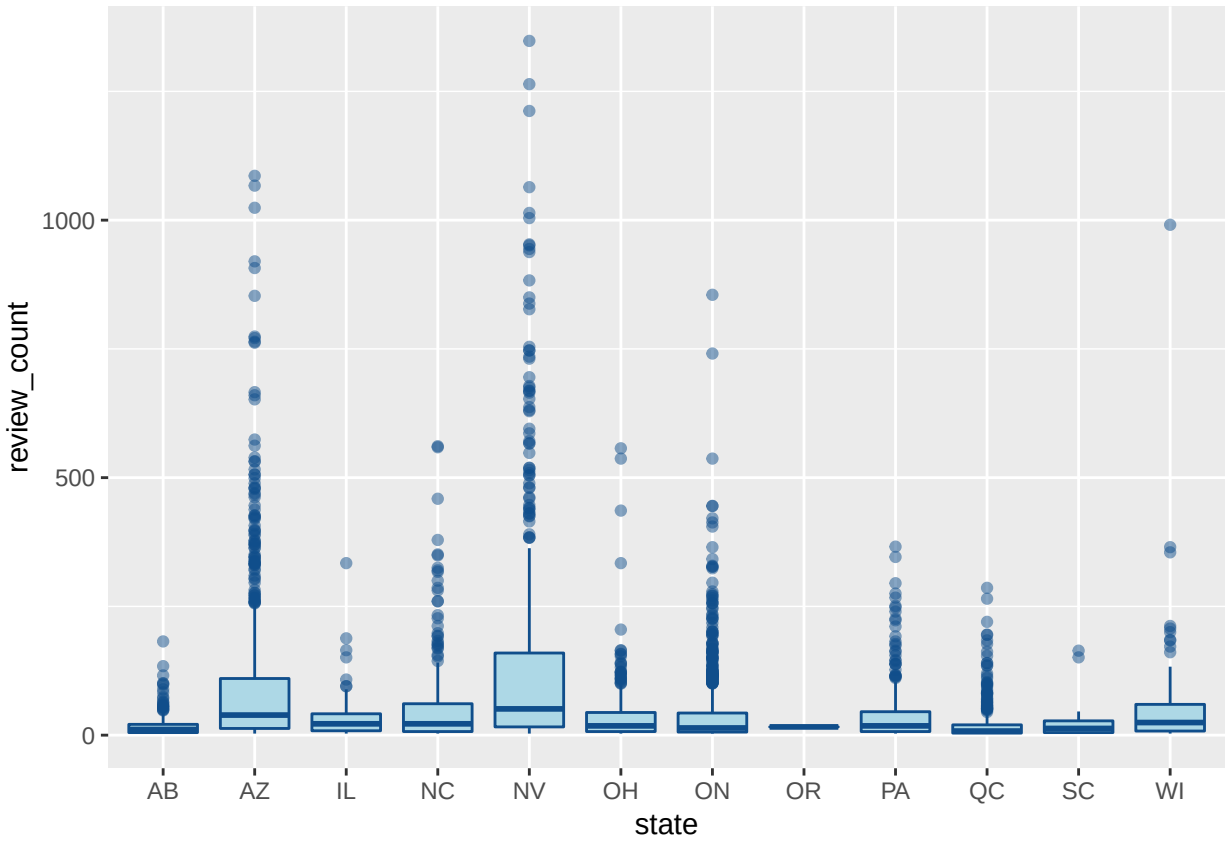
As we've discussed in the previous section, the number of the restaurants is not the only factor that leads to the variation of number of reviews. We can see from the dot plot that 4 stars and 4.5 stars are more likely to receive reviews than other levels. Also, in general, there's an increasing trend of receiving reviews from star 1 restaurants and star 4 restaurants, and decreasing trend afterwards. It makes sense because, according to our intuition, many 4 stars restaurants are "popular and good" restaurants; while higher stars, say 5, are likely to be very young restaurants: it's hardly possible for a restaurant to maintain a totally perfect feedback as long as it gets enough number of customers.

Review distribution

State

As we've mentioned in data quality section, we will only pick states with more information to see their distribution clearly. And for the reason of too much large outlier, we will get rid of outliers larger than 1500 in order to analyse the normal pattern of different states.

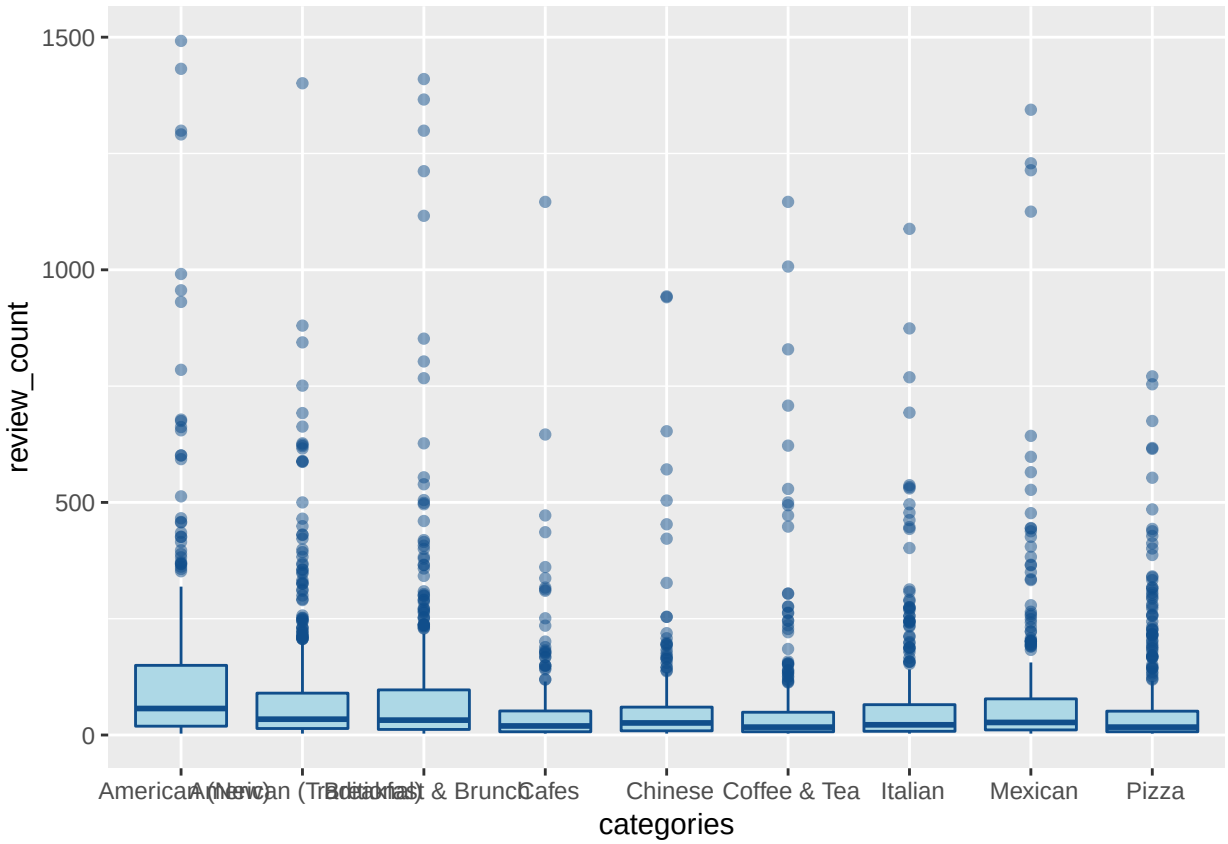
```
yelp_business2 <- yelp_business1 %>%
  filter(state == c("AB", "AZ", "IL", "NC",
              "NV", "OH", "ON", "OR", "PA", "QC", "SC", "WI")) %>%
  filter(review_count <= 1500)
ggplot(yelp_business2, aes(x = state, y = review_count)) +
  geom_boxplot(fill = "lightblue",
               color = "dodgerblue4", outlier.alpha = 0.5)
```



Category

Here we will only analyzed categories that have been chosen in the data quality section, which are more representative.

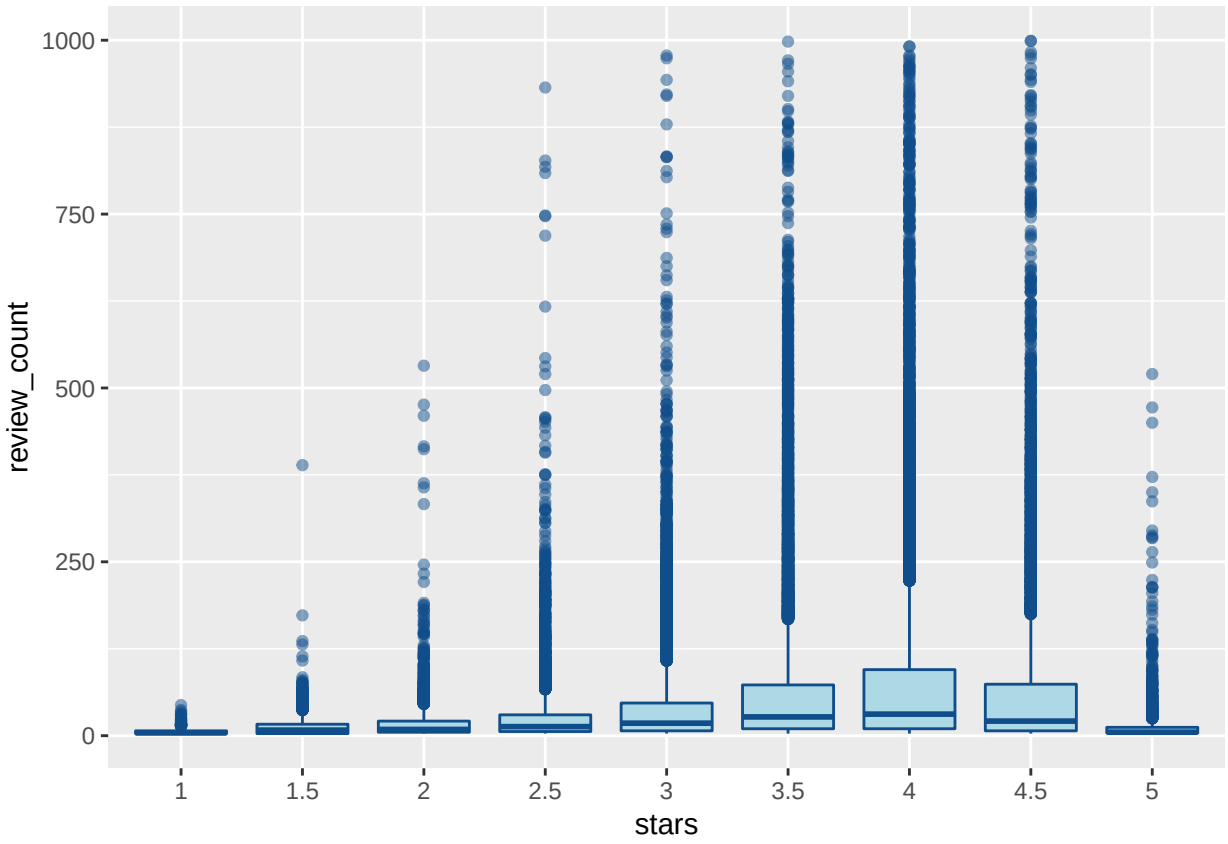
```
yelp_unn1 <- yelp_unn %>%
  filter(categories ==res_cate$categories[c(7, 9, 10, 12:17)]) %>%
  filter(review_count <= 1500)
ggplot(yelp_unn1,
       aes(x = categories, y = review_count)) +
  geom_boxplot(fill = "lightblue",
               color = "dodgerblue4",
               outlier.alpha = 0.5)
```



Clearly American food has a larger variation comparing to other kinds of food, while the median of these categories are quite similar to each other.

Star

```
yelp_business3 <- yelp_business %>%
  filter(review_count <= 1000)
yelp_business3$stars <- as.factor(yelp_business3$stars)
ggplot(yelp_business3,
       aes(x = stars, y = review_count)) +
  geom_boxplot(fill = "lightblue",
               color = "dodgerblue4", outlier.alpha = 0.5)
```



This boxplot shows that the trend of medians and Q3 of stars are same as that of average reviews.

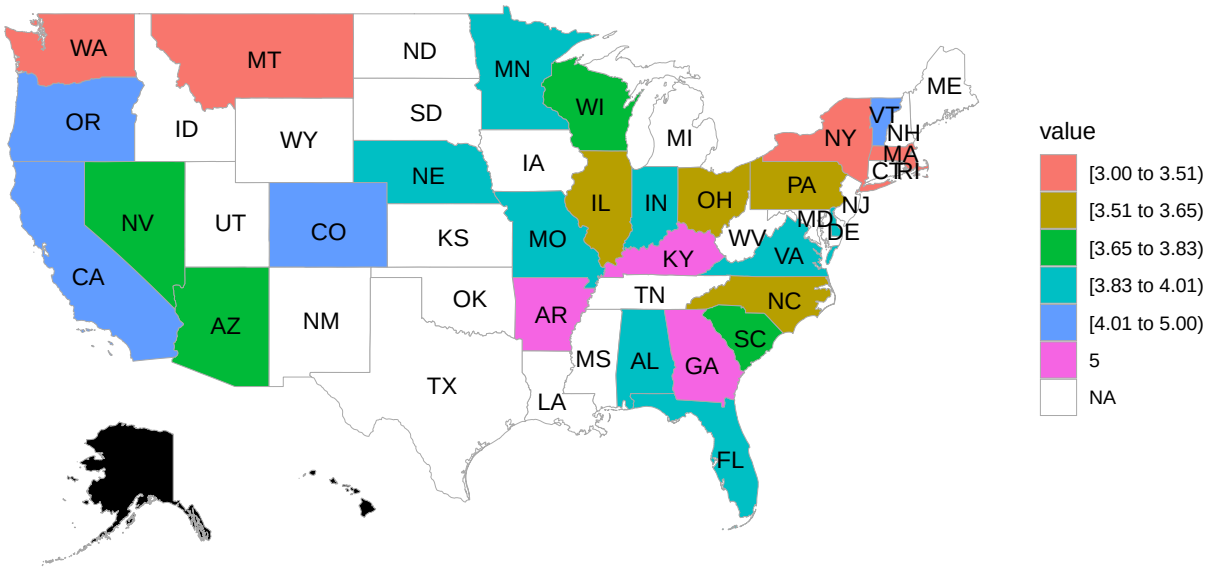
Average star

State

To explore the average stars in each state in our dataset, we generated a choropleth map and separated the states into six levels based on the ratings.

```
stars_map <- group_by(yelp_business0, state) %>%
  mutate(ave_stars = sum(stars)/length(stars))
stars_map <- stars_map[,c('state', 'ave_stars')]
stars_zip <- unique(stars_map)
colnames(stars_zip) <- c("region", "value")
stars_zip$region <- tolower(abbr2state(as.character(stars_zip$region)))
stars_zip$value <- as.numeric(stars_zip$value)
stars_zip <- stars_zip[!is.na(stars_zip$region),]
state_choropleth(stars_zip,
                 title = "Average stars for the restaurants in each area") +
  scale_fill_discrete(na.value="white")
```

Average stars for the restaurants in each area

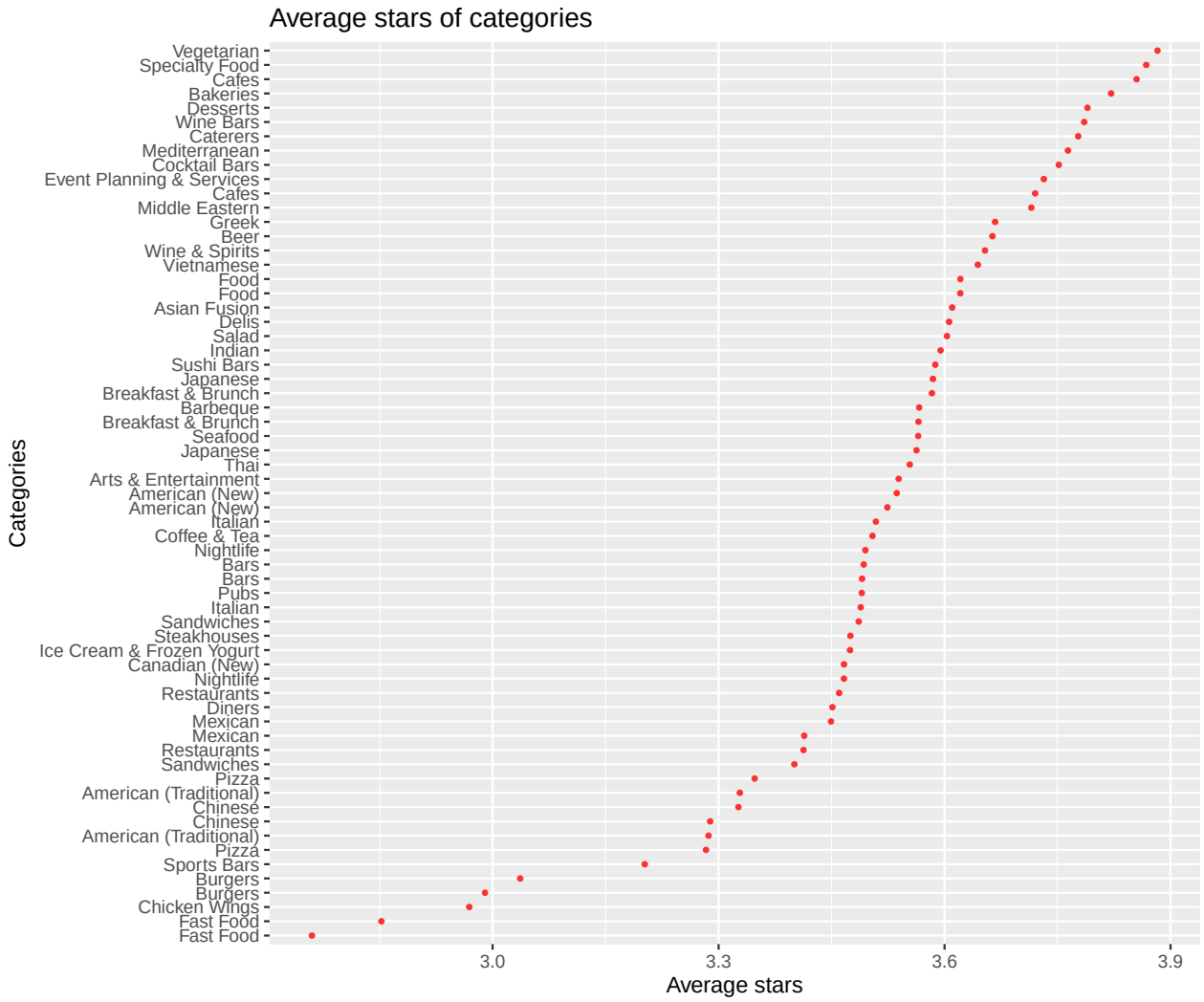


From the map, we can see NY, MT and WA have lower average stars than other states, at 3.00 to 3.51; and KY, AR and GA have average stars of 5, which probably because of small data size.

Category

We are also interested in what are the average stars for different categories. Therefore, we plot a Cleveland plot to rank the ratings in terms of categories.

```
star_cate <- yelp_unn %>%
  group_by(categories) %>%
  summarize(freq = n(), total = sum(stars)) %>%
  mutate(average = round(total/freq, 4)) %>%
  filter(freq > 700) %>%
  arrange(desc(average))
star_cate$categories <- factor(star_cate$categories,
                           levels = star_cate$categories[order(star_cate$average)])
star_cate_fig <- ggplot(star_cate, aes(x = average, y = categories)) +
  geom_point(color = "red", alpha = 0.8) +
  theme_grey(16) + xlab("Average stars") +
  ylab("Categories") +
  ggtitle("Average stars of categories")
star_cate_fig
```

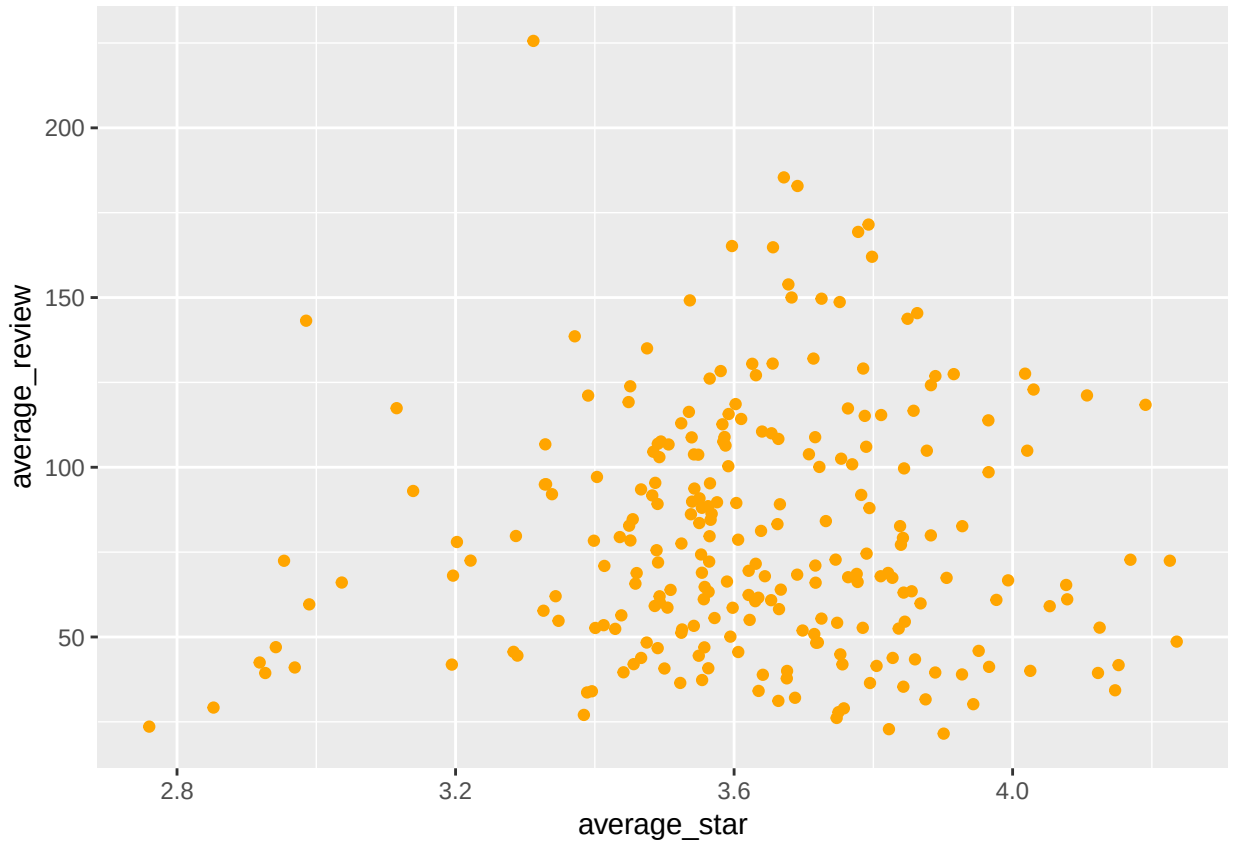


Interestingly, the categories average stars pattern and average review pattern are not very much alike. American (New) and Cocktail Bars, which are on the top of the average reviews, both fall behind, especially American (New). This indicates that although they have high discussion, they do not really receive higher evaluations than other categories.

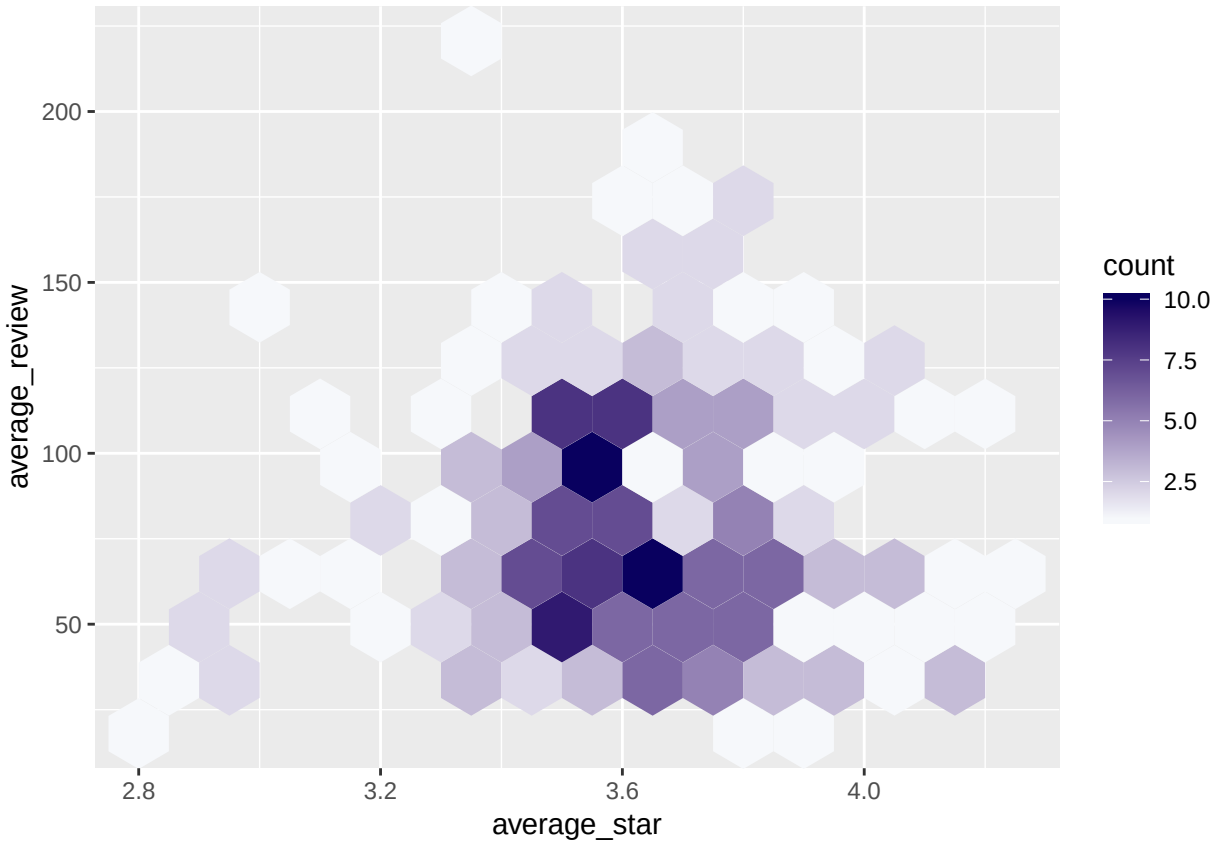
Next, we would like to explore the relationship between stars and review counts that a restaurant receives by plotting the following figures:

Relationship between average review counts and average stars

```
star_cate1 <- yelp_unn %>%
  group_by(categories) %>%
  summarize(freq = n(), total_review = sum(review_count),
            total_star = sum(stars)) %>%
  mutate(average_review = round(total_review/freq, 4),
         average_star = round(total_star/freq, 4)) %>%
  filter(freq > 50) %>%
  arrange(desc(average_star))
ggplot(star_cate1, aes(x = average_star, y = average_review)) +
  geom_point(color = "orange")
```




```
ggplot(star_cate1, aes(x = average_star, y = average_review)) +
  geom_hex(binwidth = c(0.1,18)) +
  scale_fill_gradient(low = "#F6F8FB", high = "#09005F")
```

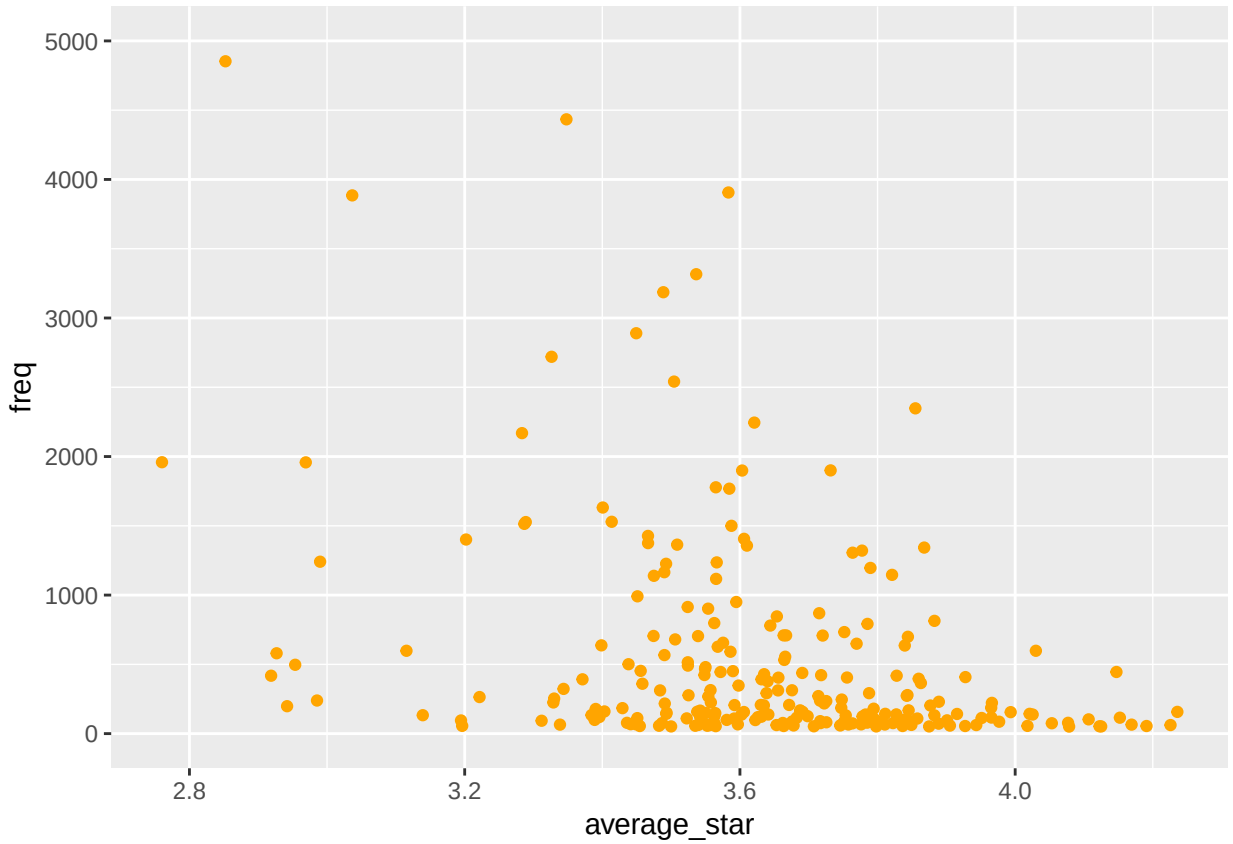


We first plot a scatterplot that shows the relationship between average review counts and average stars. Then we found that it would be easier to read if we draw a heatmap. The heatmap shows that the restaurants with lower stars are less likely to have more reviews.

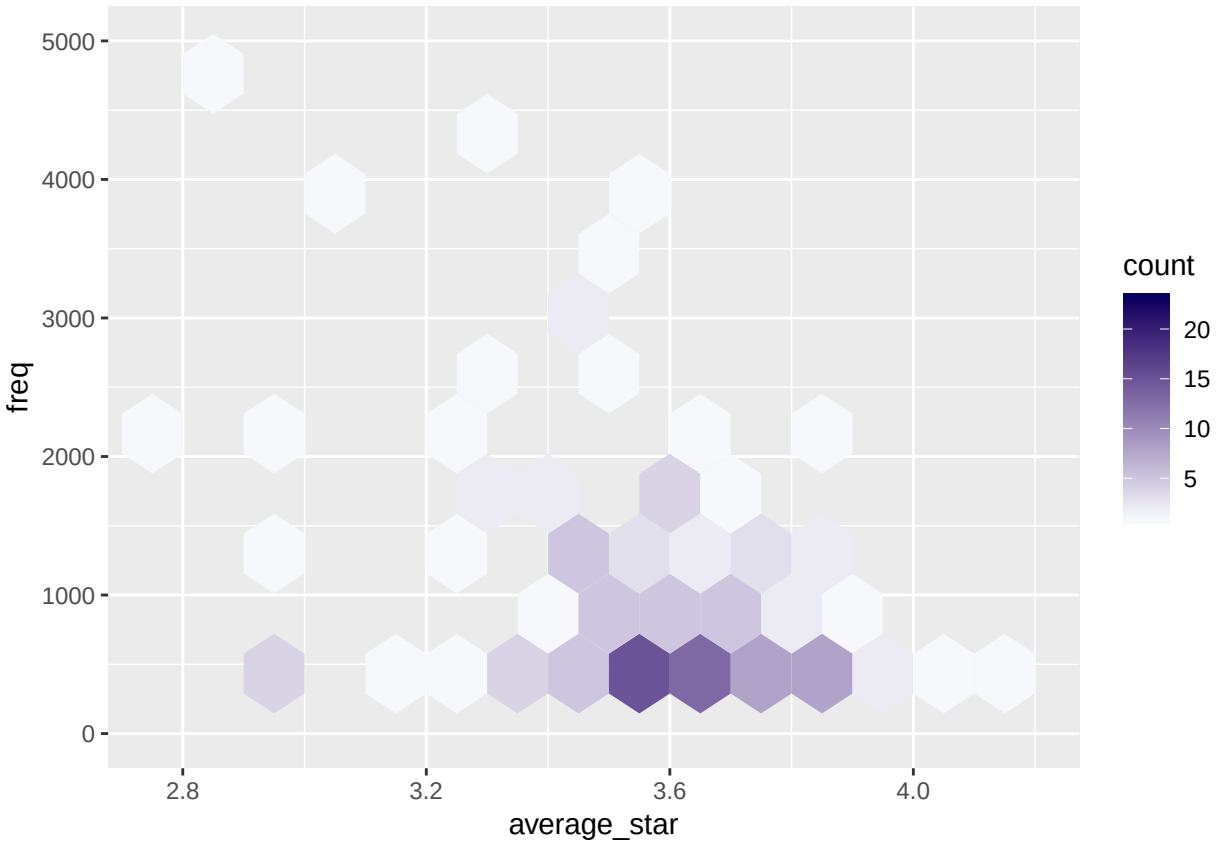
Relationship between category count and stars

Here we want to explore if there is any relationship between average stars and frequency of categories. To get rid of categories with very small frequency (we want to analyse in a general point of view), the plot omits the categories with frequency no more that 50.

```
ggplot(star_cate1, aes(x = average_star, y = freq)) +
  geom_point(color = "orange") + ylim(0, 5000)
```




```
ggplot(star_cate1, aes(x = average_star, y = freq)) +
  geom_hex(binwidth = c(0.1,500)) + ylim(0, 5000) +
  scale_fill_gradient(low = "#F6F8FB", high = "#09005F")
```



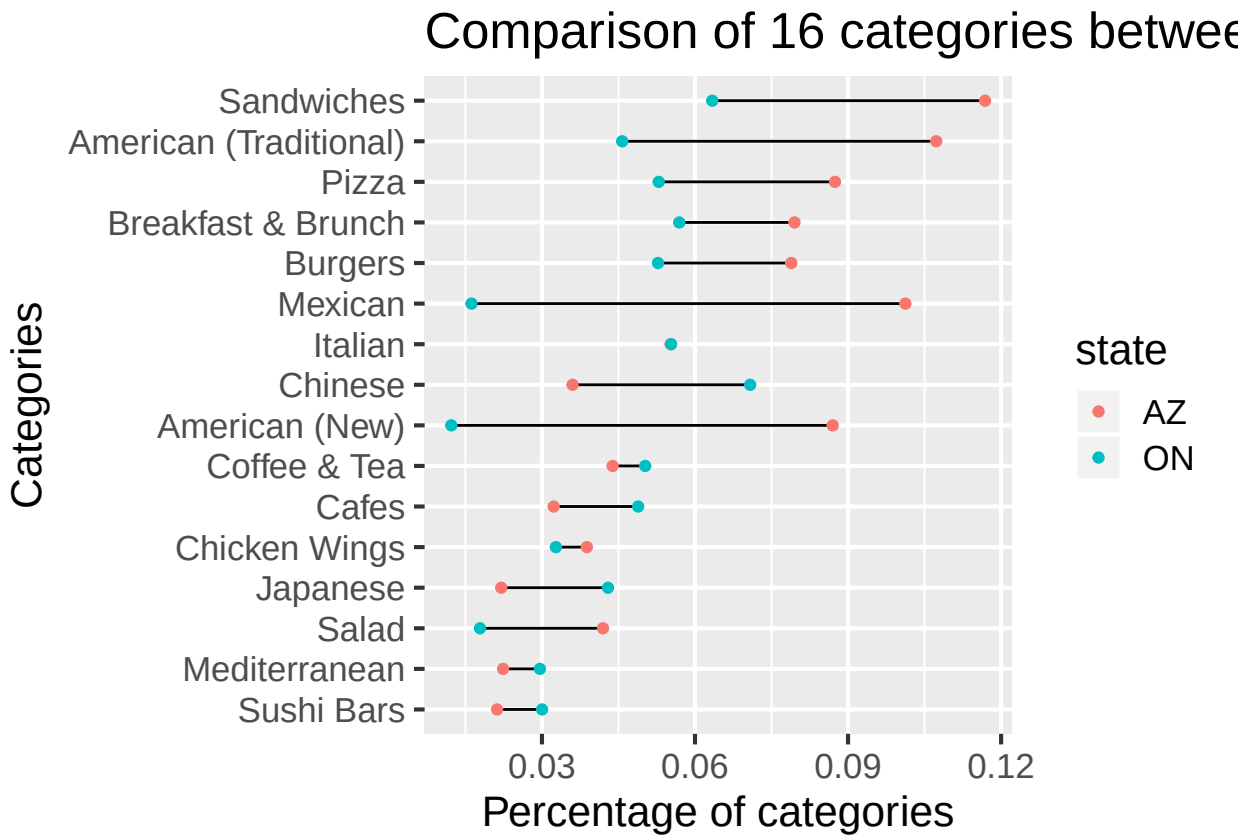
For category count and stars, both of the plots do not work well as the frequency varies from different restaruants with great variation.

ON versus AZ

From the plot that we have generated for the ranking of restaurant count, we chose Ontario, a state from Canada, and Arizona, a state from the US, to compare the difference in terms of their user preference.

```
onaz <- yelp_unn %>%
  group_by(state, categories) %>%
  summarize(freq = n()) %>%
  arrange(desc(freq))
onaz <- onaz %>%
  filter(state == "ON" | state == "AZ")
onaz <- onaz %>%
  filter(categories != "Restaurants" &
           categories != "Food" &
           categories != "Nightlife" &
           categories != "Fast Food")
onaz <- cbind(onaz, index = c(1:1304))
onaz <- onaz[-c(1:9,22),]
onaz_top <- onaz[onaz$freq > 100,]
onaz_top <- cbind(onaz_top[,c(1:3)], index = c(1:147))
onaz_top <- onaz_top[c(1,7,2,6,17,98,3,73,4,33,5,13,8,11,
                       9,14,12,19,15,22,16,40,18,66,24,27,
                       25,60,26,72,28,64),]
onaz_top <- onaz_top[order(onaz_top$state),]
```

```
onaz_top <- cbind(onaz_top[,c(1:3)],
                  total = c(rep(11082, 16), rep(14390, 16)))
onaz_top <- cbind(onaz_top,
                  perc = onaz_top$freq/onaz_top$total)
#ggplot(data = onaz, aes(x = categories, y = freq, fill = state)) + geom_bar(stat="identity", alpha = 0
onaz_fig <- ggplot(data = onaz_top,
                   aes(x = perc, y = reorder(categories, perc))) +
  geom_line(aes(group = categories)) +
  geom_point(aes(color = state)) +
  theme_grey(16) + xlab("Percentage of categories") +
  ylab("Categories") +
  ggtitle("Comparison of 16 categories between ON and AZ")
onaz_fig
```



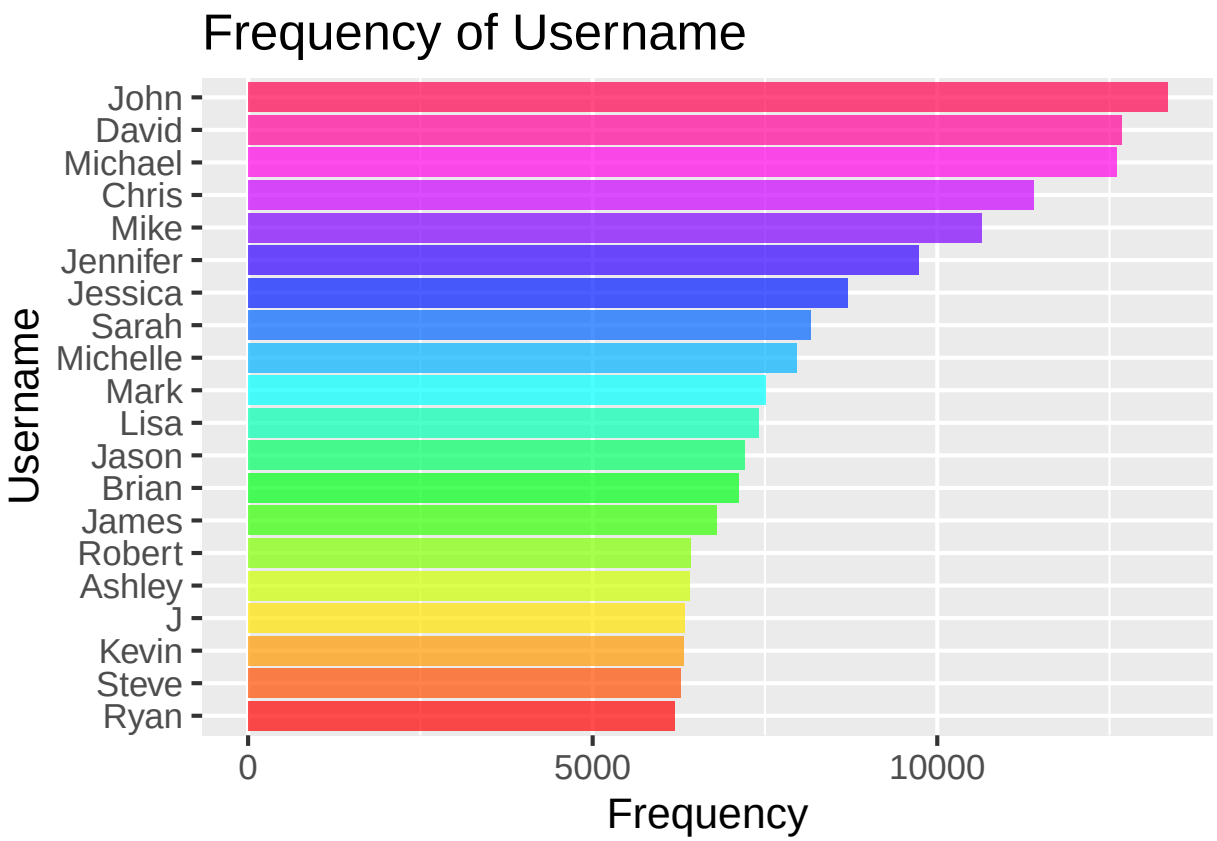
From this Cleveland plot, it's quite obvious that there preferences are different: people in AZ prefer American style food; while the ratio of Asian foods in On is larger than that of AZ, which meets our expectation since the ratio of Asian population in Canada is larger than that of United States.

User

Name frequency

```
user <- read.csv("yelp_user.csv")
username_freq <- data.frame(Name = names(summary(user$name)[1:20]),
                    Freq = summary(user$name)[1:20])
username_freq$Name <- factor(username_freq$Name,
            levels = username_freq$Name[order(-username_freq$Freq)])
```

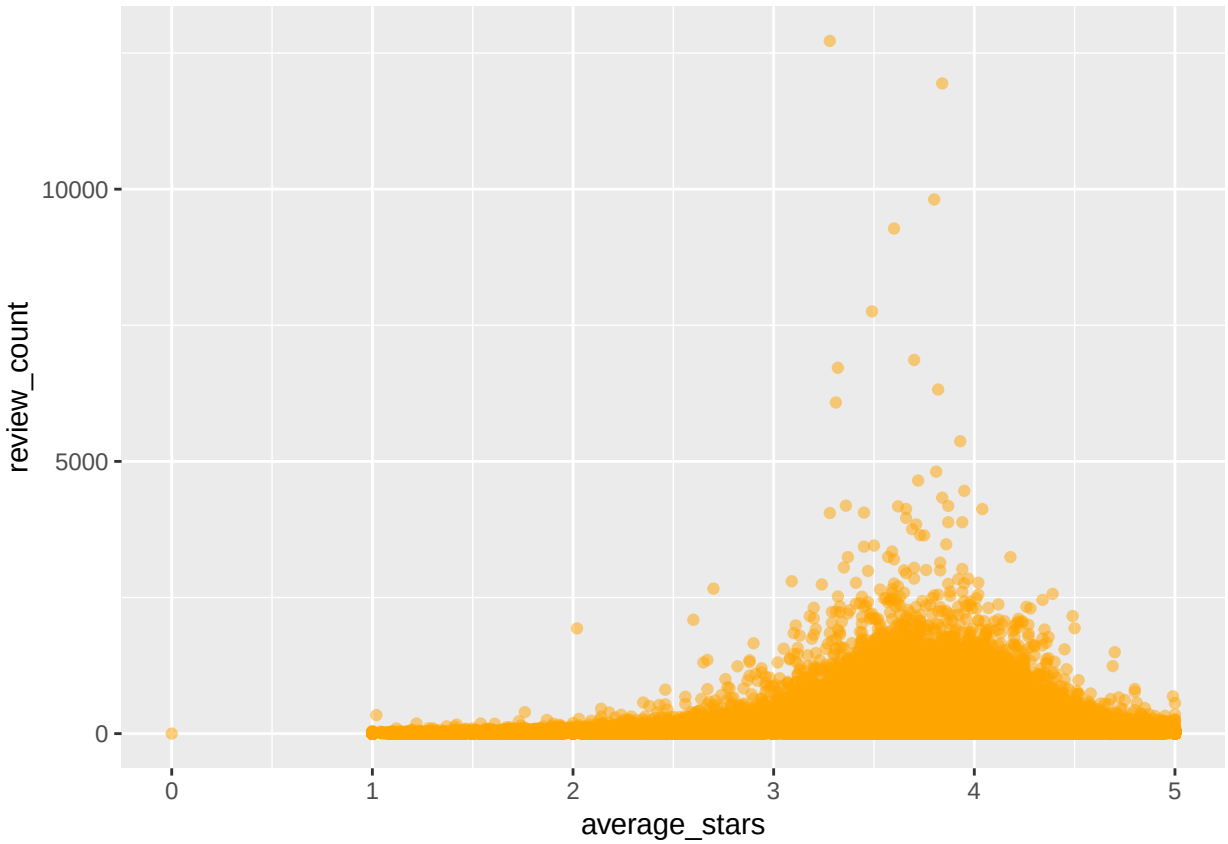
```
user_freq_p <- ggplot(username_freq,
                      aes(x = fct_rev(Name),
                          y = Freq,
                          fill=fct_rev(Freq))) +
  geom_bar(stat="identity",fill=rainbow(20, alpha = 0.7))+
  theme_minimal() + coord_flip() +
  ylab("Frequency") + xlab("Username") +
  ggtitle("Frequency of Username") + theme_grey(16)
user_freq_p
```



Above is an interesting finding when we are exploring the user data. The usernames with higher frequency tend to be similar with names we are familiar in our daily life.

Relationship between average stars and review count of users

```
ggplot(user, aes(x = average_stars, y = review_count)) +
  geom_point(color = "orange", alpha = 0.5)
```



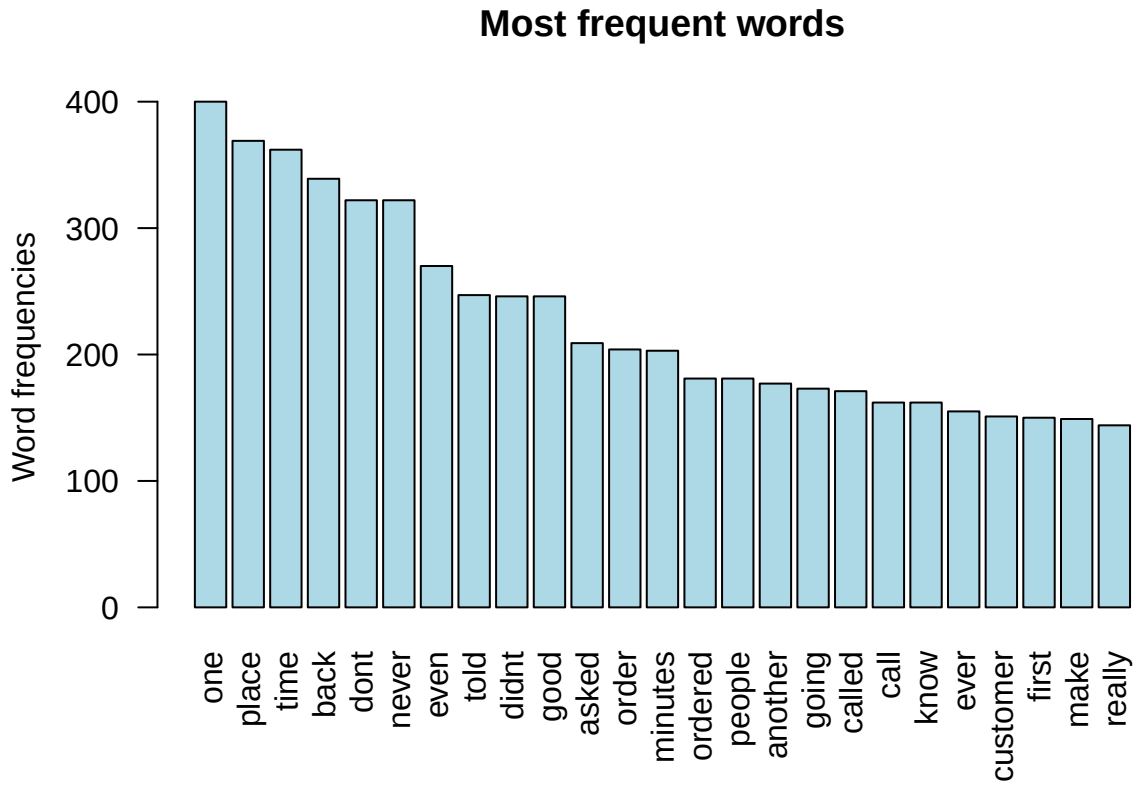
We plot the scatterplot of average stars and review count for each user, which does not provide much information of the relationship between the two variables.

Word Cloud

To further look at the review data, we would like to check if the words that show up more frequently in the reviews varies from different level of restaurants. Therefore, we generated two word clouds for restaurants with 1 star and 5 stars to see the difference.

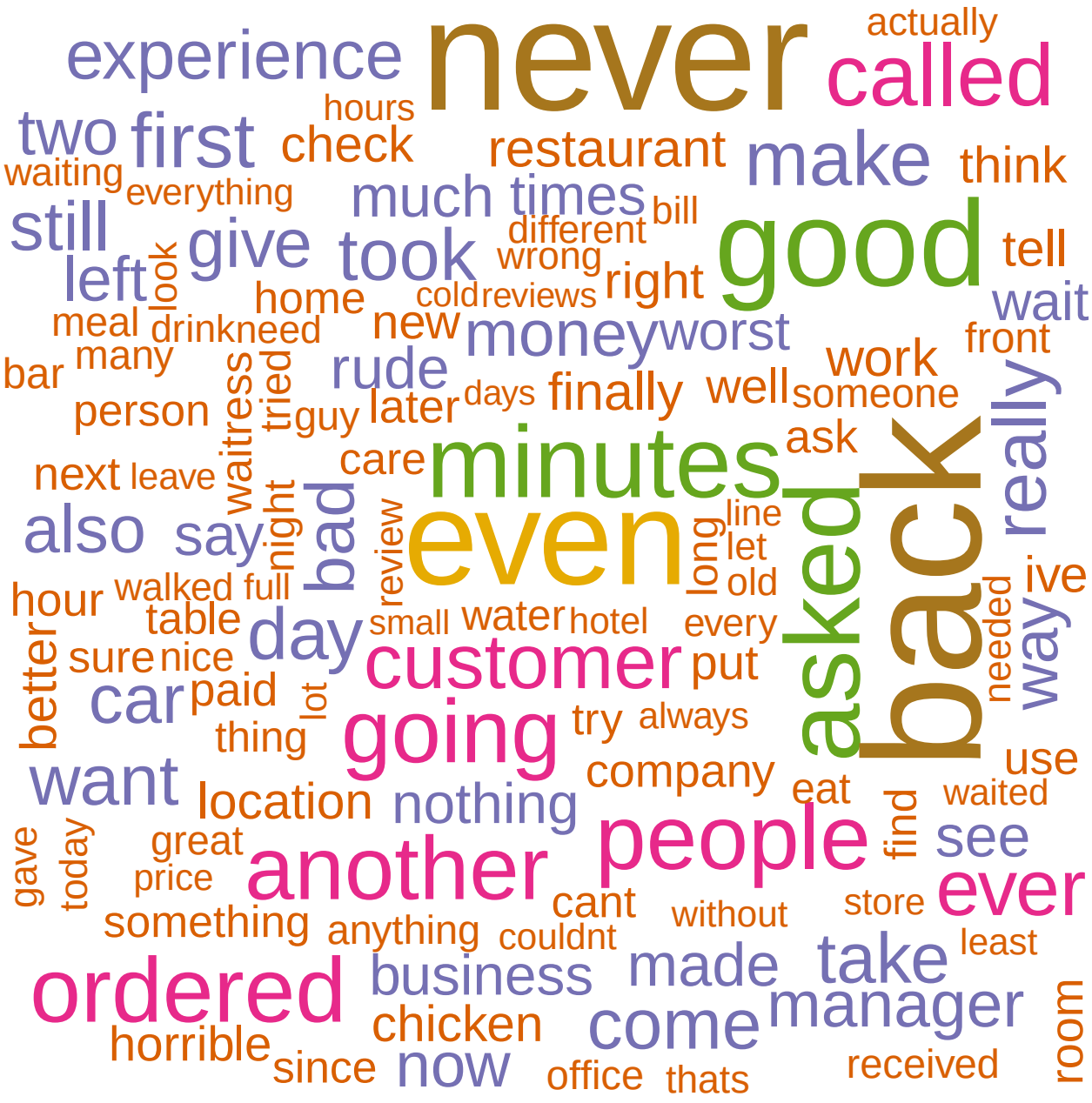
```
#save the reviews for different stars in txt files
text <- readLines("star1_review_s.txt",
                  encoding="UTF-8")

myCorpus = Corpus(VectorSource(text))
myCorpus = tm_map(myCorpus, content_transformer(tolower))
myCorpus = tm_map(myCorpus, removePunctuation)
myCorpus = tm_map(myCorpus, removeNumbers)
myCorpus <- tm_map(myCorpus, removeWords,
                   c("get", "just", "like", "said", "will",
                     "got", "went", "came" ,"can", "food", "service"))
myCorpus <- tm_map(myCorpus, removeWords, stopwords("english"))
dtm1 = TermDocumentMatrix(myCorpus,
                          control = list(minWordLength = 1))

m = as.matrix(dtm1)

freq1 <- sort(rowSums(m), decreasing = TRUE)
```

```
star1_df <- data.frame(word=names(freq1), freq=freq1)

barplot(star1_df[1:25,]$freq, las = 2,
        names.arg = star1_df[1:25,]$word,
        col ="lightblue",
        main ="Most frequent words",
        ylab = "Word frequencies")
```



We first plot this bar plot to show to frequency of words in reviews, then plot the word cloud for a better view of the words.

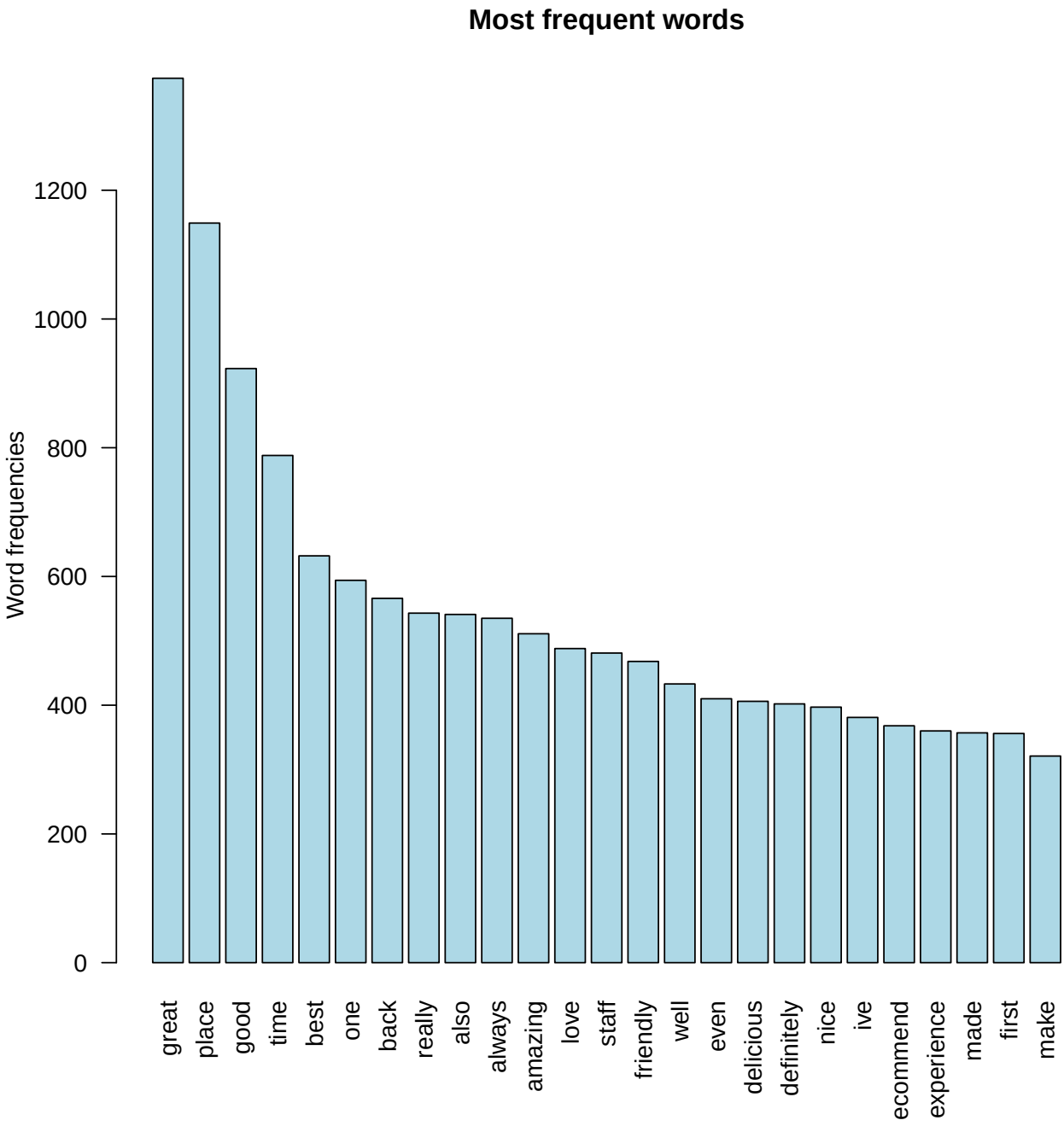
```
set.seed(5702)
wordcloud(star1_df$word,star1_df$freq,
          scale=c(8,.3),
          min.freq=2,max.words=150,
          random.order=T, rot.per=.15,
          colors=brewer.pal(8, "Dark2"))
```



From the word cloud, we clearly see that there are more nagetive words in 1-star restaurants, such as `dont`, `never`, and `horrible`.

```
#save the reviews for different stars in txt files
text <- readLines("star5_review_s.txt",
                  encoding="UTF-8")

myCorpus = Corpus(VectorSource(text))
myCorpus = tm_map(myCorpus, content_transformer(tolower))
myCorpus = tm_map(myCorpus, removePunctuation)
myCorpus = tm_map(myCorpus, removeNumbers)
myCorpus <- tm_map(myCorpus, removeWords,
                   c("get", "just", "like", "said", "will",
                     "got", "went", "came" ,"can", "food", "service"))
```

```
myCorpus <- tm_map(myCorpus, removeWords, stopwords("english"))
dtm5 = TermDocumentMatrix(myCorpus,
                          control = list(minWordLength = 1))

m = as.matrix(dtm5)

freq5 <- sort(rowSums(m), decreasing = TRUE)

star5_df <- data.frame(word=names(freq5), freq=freq5)

barplot(star5_df[1:25,]$freq, las = 2,
        names.arg = star5_df[1:25,]$word,
col ="lightblue", main ="Most frequent words",
ylab = "Word frequencies")
```




```
set.seed(5702)
wordcloud(star5_df$word,star5_df$freq, scale=c(8,.3),
          min.freq=2,max.words=150, random.order=T,
          rot.per=.15, colors=brewer.pal(8, "Dark2"))
```

After we did the exact same thing to 5-star restaruants, we found that words such as `great`, `best`, `amazing` have a much larger frequency in the reviews. This signifigant difference shows that the reviews can actually reflect the ratings from the users.

Executive Summary

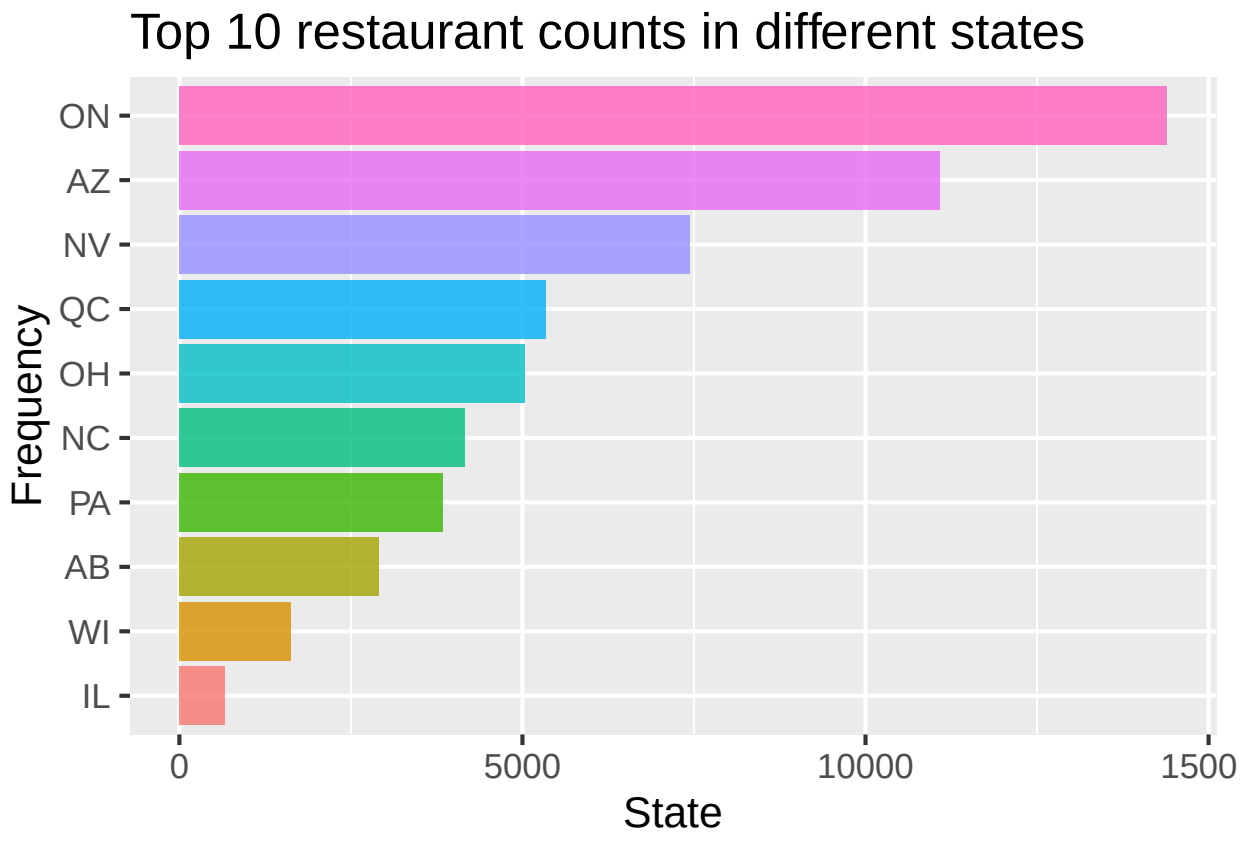
In the whole analysis, we had 2 main findings: different patterns of restaurants ratio in ON and AZ, and the different patterns of reviews for different stars. In this section, we will give the summary of each finding.

ON vs AZ

Firstly, to have a general look at the whole dataset, we plotted the restaurant counts in different states, and had the following plot:

```
yelp_business$state <- as.factor(yelp_business$state)
res_state <- yelp_business %>%
```

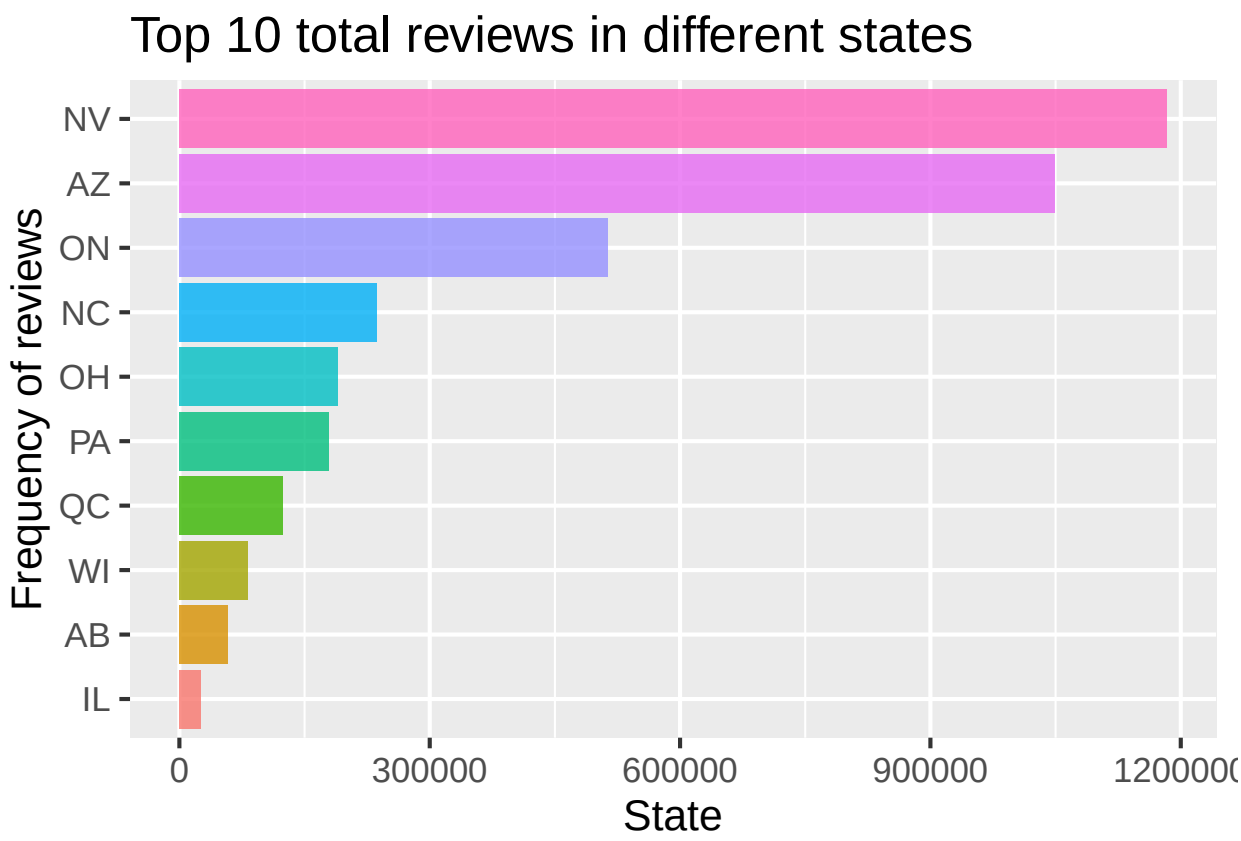
```
  group_by(state) %>%
  summarize(freq = n()) %>%
  arrange(desc(freq))
res_state$state <- as.character(res_state$state)
res_state <- res_state[is.na(as.numeric(res_state$state)),]
res_state <- res_state[nchar(res_state$state) == 2,]
res_state_top <- res_state[c(1:10),]
res_state_top$state <- factor(res_state_top$state,
                              levels = res_state_top$state[order(res_state_top$freq)])
res_state_top_fig <- ggplot(res_state_top,
                            aes(x = state, y = freq,
                                fill = cut(freq, 100))) +
  geom_histogram(stat = "identity", show.legend = FALSE, alpha = 0.8) +
  coord_flip() + xlab("Frequency") + ylab("State") +
  ggtitle("Top 10 restaurant counts in different states") + theme_grey(16)
res_state_top_fig
```



In this plot, we found that the dataset was probably not a full dataset: there are very few restaurants in some states, and even no restaurants, which is impossible in real life. The reason for this problem is probably that there are many lacking information - the dataset is not a full dataset of yelp. For this reason, we only showed the representative 10 states on the top of the list. And from the figure we can see an obvious decreasing trend of restaurant count, and the last state, IL, has around 500 restaurants, while ON, on the top of the list, has about 14000 restaurants.

```
rev_total_state <- yelp_business %>%
  filter(str_detect(categories, "Restaurant")) %>%
  group_by(state) %>%
  summarize(freq = n(), total = sum(review_count)) %>%
```

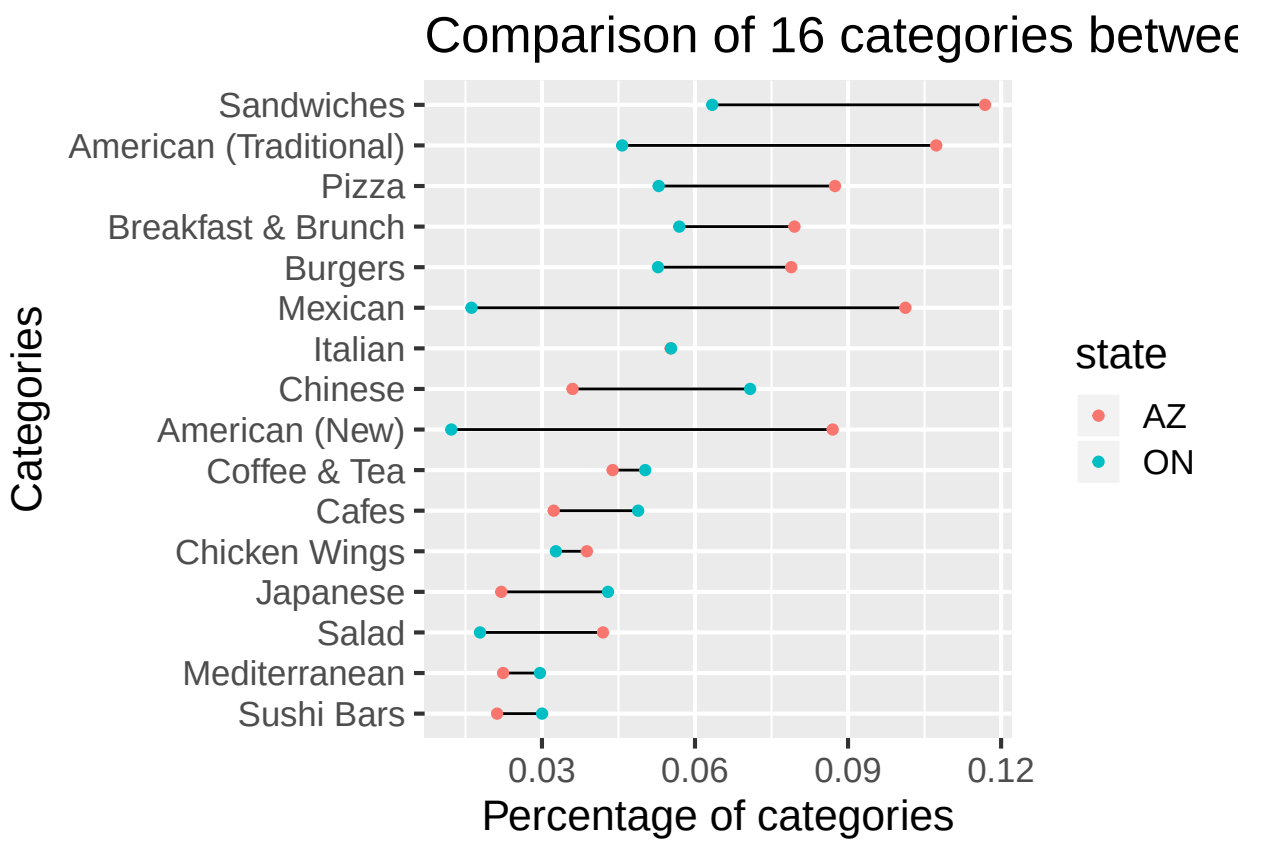
```
  mutate(average = round(total/freq, 4)) %>%
  arrange(desc(total))
rev_total_state_top <- rev_total_state[c(1:10),]
rev_total_state_top$state <- factor(rev_total_state_top$state,
                                    levels = rev_total_state_top$state[order(rev_total_state_top$total)]
rev_total_state_top_fig <- ggplot(rev_total_state_top,
                                  aes(x = state, y = total,
                                      fill = cut(total, 100))) +
  geom_histogram(stat = "identity", show.legend = FALSE, alpha = 0.8) +
  coord_flip() +
  xlab("Frequency of reviews") + ylab("State") +
  ggtitle("Top 10 total reviews in different states") +
  theme_grey(16)
rev_total_state_top_fig
```



For the same reason of the previous plot, we only plotted states with top 10 total review counts. Same as restaurant counts, there is an obvious decreasing trend of total review counts, and ON and AZ are on the top of the list. It means that there are states like ON and AZ that have more restaurant information than others. Thus we chose ON and AZ as 2 different states to analyze their category ratio difference.

```
onaz <- yelp_unn %>%
  group_by(state, categories) %>%
  summarize(freq = n()) %>%
  arrange(desc(freq))
onaz <- onaz %>%
  filter(state == "ON" | state == "AZ")
onaz <- onaz %>%
  filter(categories != "Restaurants" &
```

```
            categories != "Food" &
            categories != "Nightlife" &
            categories != "Fast Food")
onaz <- cbind(onaz, index = c(1:1304))
onaz <- onaz[-c(1:9,22),]
onaz_top <- onaz[onaz$freq > 100,]
onaz_top <- cbind(onaz_top[,c(1:3)], index = c(1:147))
onaz_top <- onaz_top[c(1,7,2,6,17,98,3,73,4,33,5,13,8,
                       11,9,14,12,19,15,22,16,40,18,66,
                       24,27,25,60,26,72,28,64),]
onaz_top <- onaz_top[order(onaz_top$state),]
onaz_top <- cbind(onaz_top[,c(1:3)],
                  total = c(rep(11082, 16), rep(14390, 16)))
onaz_top <- cbind(onaz_top, perc = onaz_top$freq/onaz_top$total)
#ggplot(data = onaz, aes(x = categories, y = freq, fill = state)) + geom_bar(stat="identity", alpha = 0
onaz_fig <- ggplot(data = onaz_top,
                   aes(x = perc, y = reorder(categories, perc))) +
  geom_line(aes(group = categories)) +
  geom_point(aes(color = state)) +
  xlab("Percentage of categories") +
  ylab("Categories") +
  ggtitle("Comparison of 16 categories between ON and AZ") +
  theme_grey(16)
onaz_fig
```



From this Cleveland plot, it's quite obvious that there preferences are different: people in AZ prefer American style food; while the ratio of Asian foods in On is larger than that of AZ, which meets our expectation since

the ratio of Asian population in Canada is larger than that of United States.

Word cloud

In this section, we would like to analyze how stars are related to contents of review. Does some words tend to appear more in 5 stars than that in 1 star? To find out the patterns, we plotted word clouds and tried to figure out the difference.

```
set.seed(5702)
wordcloud(star1_df$word,star1_df$freq, scale=c(8,.3),
          min.freq=10,max.words=150,
          random.order=T, rot.per=.15,
          colors=brewer.pal(8, "Dark2"))
```

```
wordcloud(star5_df$word,star5_df$freq, scale=c(8,.3),
          min.freq=10,max.words=150,
          random.order=T, rot.per=.15,
          colors=brewer.pal(8, "Dark2"))
```

The difference in these two word clouds shows that reviews do reflect the quality of restaurants. For example, words such as `dont`, `never`, and `horrible` clearly show up more frequently for 1-star restaurants, whereas 5-star restaurants have more reviews that include words such as `great`, `best`, and `amazing`.

Interactive Component

The word clouds that we have generated succesfully describe the difference between words that show up more frequently in terms of different ratings of restaurants. To further customize the results by controlling the number of words we would like to involve and the minimum frequency of words, we developed a Shiny app that allows users to interact with and find the patterns of words for restaurants in different levels. Here is the link to our app: https://sabrinali18.shinyapps.io/WordCloud/. Users can first choose a level of stars of restaurants, and then play around with the word cloud to see what it looks like when the minimum frequency

and the maximum number of words are changed. The word cloud ranks the word frequency by the size and color of words, and the changes through different choices are clear to identify.

In addition to the word frequency that we have discovered, we can also calculate the association between two words using our results of textual analysis. For example, the we can generate the words that are highly associated with the word "service" for restaurants with different ratings. Future work can involve another interactive component that allows users to type in a word and find out the rank of the words that it is associated with.

Conclusion

In conclusion, we visualized the Yelp dataset with different plots to explore the patterns and relationships within and across variables that we were interested in. Throughout this project, we have gained experience of choosing certain plots for better visualizing the patterns of data. In addition, we learned the process of exploratory data analysis that starts from scratch, and how to deliver our findings to audience. More work can be done if we had a full dataset with information for every state in the US. Also, for the word cloud, we noticed that there are still some neutral words in it after we removed certain stopwords. This can be improved by taking out words that show up frequently in every level of restaurants. More analysis on text and user preference can be done in the future as well.