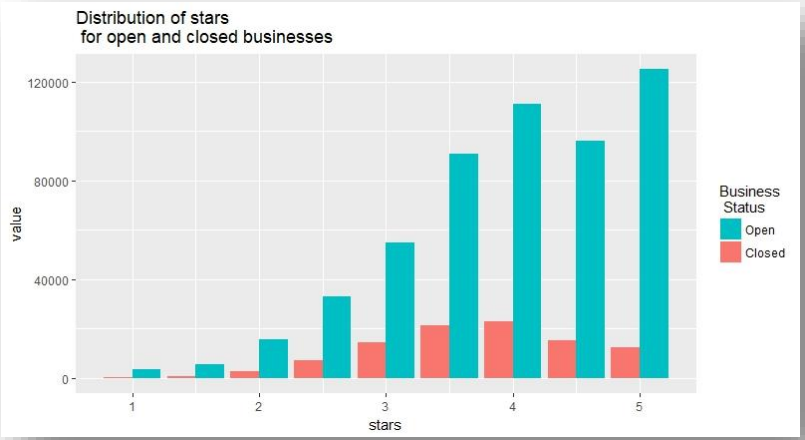
Week 13 Assignment – Yelp and R

Brian Mallari

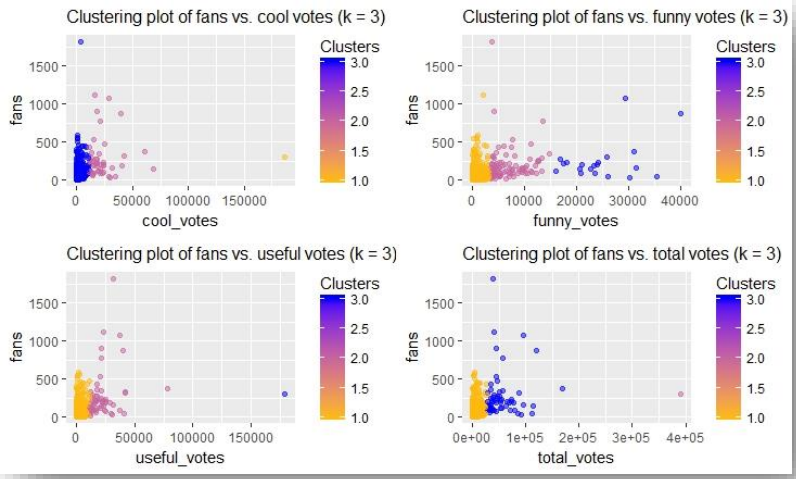
Part 1

Using R, create visualizations with different perspectives of the data we covered during the lecture, using Yelp dataset of 'business,' and 'users.' In other words, you are asked to try different variables for the visualization other than used in the example. [5 points]

'Business' Data



'User' Data



Note: Because there are over 1.3 million records in the 'user' data set, my computer could not reliably process all of that data. Therefore, I took a randomized sample of about 130,000 records (or 10% of the total number of records) and generated the clustered plots from that.

Part 2

Through this practice of data visualization with Yelp data set, did you find anything interesting that we didn't cover in the class? For instance, you may have created a plot that shows some interesting patterns about one variable or interesting relationships between different variables. These are the kinds of observations that generate new ideas and innovations. Report this using appropriate graphs (unless you had them for the previous question) and a brief description. [5 points]

'Business' Data

Looking at the data for 'business', I noticed that there are over 174,000 records in the data set. Moreover, there is a column titled 'open', as in some businesses are open and others are closed. This lead me to wonder how many businesses in the data set are open.

```
> # see how many business are open
> number_of_open_yelp_businesses <- sum(yelp_business_data$open)
> number_of_open_yelp_businesses
[1] 146702
```

```
> # see percentage of Yelp businesses still open
> percentage_of_open_yelp_business =
+   number_of_open_yelp_businesses/nrow(yelp_business_data)
> percentage_of_open_yelp_business
[1] 0.8403765
```

Now considering how businesses in this data set have a 'star' rating from 1 to 5, I figured I'd look at the distribution of stars for both the open and closed business. The bar graph of the distribution of 'star' ratings for both open and closed businesses shows that many of the scores for the open businesses are at the higher end of the scale (i.e. 3.5 – 5). This makes sense since a business is more likely to stay in business if it does a better job at satisfying the customer.

However, with respect to the closed businesses, appears to be a unimodal distribution with the peak at 4 and a tail extending into the lower scores. This detail about the closed businesses stands out to me because the distribution would show that at some point the now-closed businesses were actually satisfying their customer. Perhaps a timeline of the scores for each closed business along with a date for when the business finally closed could be beneficial to further analysis, as it is possible that the quality of the goods or services provided by the closed businesses started out quite good but degraded over time and people just stopped rating the business. Another possibility is that circumstances forced a prosperous business to suddenly close despite the positive reviews.

Also, the height of the bars for the opened businesses is taller than the height of the bars for the closed businesses. This makes sense since there are just more open businesses than closed businesses. (To be more precise, about 84% of the businesses in the Yelp 'business' data set are open.)

‘User’ Data

I figured that any sort of vote for a review (whether it be a cool vote, a funny vote, or a useful vote) would be an indicator of the quality of the reviewer themselves, so the count of such votes could somehow be related to the number of fans the reviewer would have.

Looking at the correlation matrix of just the numeric values (which includes the different kinds of votes and the number of fans), we can see that there are actually weak-to-moderate, positive correlations between each of the types of votes and the number of fans.

```
> # look for any correlations among the numerical values of the yelp user data
> yelp_user_data_numerics_corr <- cor(yelp_user_data_numerics)
> yelp_user_data_numerics_corr
             review_count average_stars  cool_votes funny_votes useful_votes       fans
review_count   1.00000000   0.013033868 0.254120531  0.22975916  0.291061751 0.57017469
average_stars  0.01303387   1.000000000 0.004150506  0.00285274  0.003682548 0.01062031
cool_votes     0.25412053   0.004150506 1.000000000  0.85081444  0.924656981 0.41741552
funny_votes    0.22975916   0.002852740 0.850814441  1.00000000  0.845040567 0.36483890
useful_votes   0.29106175   0.003682548 0.924656981  0.84504057  1.000000000 0.45355493
fans           0.57017469   0.010620311 0.417415517  0.36483890  0.453554935 1.00000000
```

Unlike the clustered plots, this correlation matrix was generated using all 1.3 million+ records in the ‘user’ data set.

I also noticed that there also are strong, positive correlations among each of the different types of votes. These correlations could make it difficult to detect the individual contributions of each variety of vote in a regression model between the types of votes and number of fans. Therefore, I went on to wonder if there is any relationship among the total number of votes received for any variety and the number of fans, since each of the three varieties of vote can be generalized as an indicator of any sort of positive sentiment towards the reviewer and their reviews.

```
> #add a column of total votes to the numerics data frame
> yelp_user_data_numerics$total_votes = yelp_user_data_numerics$cool_votes +
+   yelp_user_data_numerics$funny_votes +
+   yelp_user_data_numerics$useful_votes
>
> # look at correlation between total votes and fans
> cor(yelp_user_data_numerics$total_votes, yelp_user_data_numerics$fans)
[1] 0.4347465
```

Like in the previous correlation matrix, this correlation was generated using all 1.3 million+ records in the ‘user’ data set. Moreover, this value is comparable to the correlation coefficients between each variety of vote and the number of fans.

With respect to the clusters in the arranged grid of cluster plats above, the three clusters each exhibit similar properties despite the different distribution of points for each scatterplot:

1. One cluster towards the left of the x-axis with relatively few votes
2. One cluster towards the middle of the x-axis with a moderate number votes
3. One cluster towards the right of the x-axis with a high number of votes

The variation in color for each cluster is a function of the kmeans function being applied each time for each scatterplot.

From both the cluster plots and the weak-to-moderate, positive correlation coefficients, it can be seen that people with the highest number of votes don't necessarily have the highest number of fans. Nevertheless, people with the highest number of votes (either for a particular type of vote or for all of the votes combined) could still be influential among fans and non-fans alike on Yelp. If someone on Yelp would like to be gain some degree of influence, perhaps it may be advantageous for them to evaluate characteristics of high-vote users and try to mimic them. However, if a business would like to gain some exposure, perhaps they would like to appeal to a high-vote user in order to possibly receive a high-vote review which could in turn increase traffic to the business.

R Script

```
# R script for assignment on Yelp and R

# import the data
yelp_business_data <- read.csv(file = "business.json.csv", header = TRUE)
yelp_user_data <- read.csv(file = "user.json.csv", header = TRUE)

# exploratory data analysis - Yelp business data

# evalate the structure of the data frame for the Yelp business data
str(yelp_business_data)

# see how many business are open
number_of_open_yelp_businesses <- sum(yelp_business_data$open)
number_of_open_yelp_businesses

# see percentage of Yelp businesses still open
percentage_of_open_yelp_business =
  number_of_open_yelp_businesses/nrow(yelp_business_data)
percentage_of_open_yelp_business

# cut down the data to just "open" and "stars"
yelp_business_data_subset <- yelp_business_data[ ,
  c("open", "stars")
]

# get a summation of the different star counts
yelp_business_data_star_counts <- aggregate(yelp_business_data_subset,
  by = list(yelp_business_data_subset$open, yelp_business_data_subset$stars), sum
)

# rename the columns of the new data frame
colnames(yelp_business_data_star_counts) <- c("open", "stars", "open_count", "star_count")

# remove "open_count"
yelp_business_data_star_counts_reduced <-
  subset(yelp_business_data_star_counts, select = -open_count)

# reshape the reduced data frame
library(reshape2)
yelp_business_data_long <- melt(yelp_business_data_star_counts_reduced,
                          id.vars = c("open", "stars"),
                          measure.vars = "star_count"
)

# look at the combined disrubtion of stars for both open and closed businesses
library(ggplot2)
```

```
ggplot(yelp_business_data_long, aes(x = stars, y = value, fill = factor(open))) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_discrete(name = "Business\n Status",
                      breaks = c(1,0),
                      labels = c("Open", "Closed")
  ) +
  labs(title = "Distribution of stars\n for open and closed businesses")

# exploratory data analysis - Yelp user data

# look at the structure of the yelp user date file
str(yelp_user_data)

# extract from the yelp user data file only the columns with numeric values
library(dplyr)
yelp_user_data_numerics <- select_if(yelp_user_data, is.numeric)

# look for any correlations among the numerical values of the yelp user data
yelp_user_data_numerics_corr <- cor(yelp_user_data_numerics)
yelp_user_data_numerics_corr

#add a column of total votes to the numerics data frame
yelp_user_data_numerics$total_votes = yelp_user_data_numerics$cool_votes +
  yelp_user_data_numerics$funny_votes +
  yelp_user_data_numerics$useful_votes

# look at correlation between total votes and fans
cor(yelp_user_data_numerics$total_votes, yelp_user_data_numerics$fans)

# generate a randomly-selected subset of the 'user' data set
sample_size <- floor(0.10 * nrow(yelp_user_data_numerics))
set.seed(123)
sample_index <- sample(seq_len(nrow(yelp_user_data_numerics)),
                       size = sample_size
)
yelp_user_data_numerics_sample <- yelp_user_data_numerics[sample_index, ]

# look at clustering plots of the randomly-selected subset from above

# fans vs. cool votes
yelp_user_data_cluster_cool_fans = kmeans(yelp_user_data_numerics_sample[,
  c("cool_votes", "fans")],
3)
library(ggplot2)
cluster_fans_cool <- ggplot(yelp_user_data_numerics_sample, aes(x = cool_votes, y = fans,
  color = yelp_user_data_cluster_cool_fans$cluster)) +
  geom_point(alpha = 0.50) +
  scale_color_gradient(low = "darkgoldenrod1", high = "blue", name = "Clusters") +
```

```
  labs(title = "Clustering plot of fans vs. cool votes (k = 3)") +
  theme(plot.title = element_text(size = 12))

# fans vs. funny votes
yelp_user_data_cluster_funny_fans = kmeans(yelp_user_data_numerics_sample[,
  c("funny_votes", "fans")],
3)
library(ggplot2)
cluster_fans_funny <- ggplot(yelp_user_data_numerics_sample, aes(x = funny_votes, y = fans,
  color = yelp_user_data_cluster_funny_fans$cluster)) +
  geom_point(alpha = 0.50) +
  scale_color_gradient(low = "darkgoldenrod1", high = "blue", name = "Clusters") +
  labs(title = "Clustering plot of fans vs. funny votes (k = 3)") +
  theme(plot.title = element_text(size = 12))

# fans vs. useful votes
yelp_user_data_cluster_useful_fans = kmeans(yelp_user_data_numerics_sample[,
  c("useful_votes", "fans")],
3)
library(ggplot2)
cluster_fans_useful <- ggplot(yelp_user_data_numerics_sample, aes(x = useful_votes, y = fans,
  color = yelp_user_data_cluster_useful_fans$cluster)) +
  geom_point(alpha = 0.50) +
  scale_color_gradient(low = "darkgoldenrod1", high = "blue", name = "Clusters") +
  labs(title = "Clustering plot of fans vs. useful votes (k = 3)") +
  theme(plot.title = element_text(size = 12))

# fans vs. total votes
yelp_user_data_cluster_total_fans = kmeans(yelp_user_data_numerics_sample[,
  c("total_votes", "fans")],
3)
library(ggplot2)
cluster_fans_total <- ggplot(yelp_user_data_numerics_sample, aes(x = total_votes, y = fans,
  color = yelp_user_data_cluster_total_fans$cluster)) +
  geom_point(alpha = 0.50) +
  scale_color_gradient(low = "darkgoldenrod1", high = "blue", name = "Clusters") +
  labs(title = "Clustering plot of fans vs. total votes (k = 3)") +
  theme(plot.title = element_text(size = 12))

# look at all of the cluster plots between votes and fans
library(gridExtra)
grid.arrange(cluster_fans_cool, cluster_fans_funny,
          cluster_fans_useful, cluster_fans_total, nrow = 2)
```

References

1. Information on working with a TAR file was taken from the following webpage: https://www.lifewire.com/tar-file-2622386
2. Information on the .JSON file extension was taken from the following webpage: https://fileinfo.com/extension/json
3. Information on converting from .JSON to .CSV was taken from the following webpage: https://github.com/Yelp/dataset-examples/issues/21
4. Information on unpacking tar files was taken from the following webpage: https://wiki.haskell.org/How_to_unpack_a_tar_file_in_Windows
5. Tool for extracting the data from the tar file was taken from the following webpage: https://www.7-zip.org/
6. Information on resolving a 'charmap' codec error when running the data search scripts was taken from the following webpage: https://stackoverflow.com/questions/27092833/unicodeencodeerror-charmap-codec-cant-encode-characters
7. Information on running a Python script through the command prompt was taken from the video podcast, 13.1 - Using R to analyze Yelp data, by Dr. Chirag Shah
8. Information on MemoryError in Python was taken from the following webpage: https://stackoverflow.com/questions/11283220/memory-error-in-python
9. Information on installing 64-bit Python on Windows 10 was taken from the following webpage: https://www.youtube.com/watch?v=d8J6359gS_Q
10. Information on the 64-bit version of Anaconda was taken from the following webpage: https://courses.edx.org/asset-v1:MITx+6.008.1x+3T2016+type@asset+block/anaconda.html
11. Information on using command line options when running a script in Python was taken from the video podcast, 11.1 - Using Python to collect YouTube data, by Dr. Chirag Shah.
12. Code selection for analysis was based off notes for the video podcast, 13.1 - Using R to analyze Yelp data, by Dr. Chirag Shah.
13. Information on how to get the number of rows in data frame in R was taken from the following webpage: https://stat.ethz.ch/R-manual/R-devel/library/base/html/nrow.html
14. Information on generating a subset of a data frame in R was taken from the following webpage: https://www.statmethods.net/management/subset.html
15. Information on importing data from a csv file into R while preventing the strings from being converted to factors was taken from the following webpage: https://www.r-bloggers.com/5-ways-to-subset-a-data-frame-in-r/
16. Information on generating a plot of side-by-side bars in R was taken from the following webpage: https://stackoverflow.com/questions/22305023/how-to-get-a-barplot-with-several-variables-side-by-side-grouped-by-a-factor
17. Information on removing a column from a data frame in R was taken from the following webpage: https://stackoverflow.com/questions/5234117/how-to-drop-columns-by-name-in-a-data-frame
18. Information on the melt() function on R was taken from the following webpage: https://www.r-bloggers.com/melt/
19. Information on obtaining a summary for a regression model in R was taken from the following webpage: http://blog.yhat.com/posts/r-lm-summary.html

20. Information on selecting only numeric columns from a data frame in R was taken from the following webpage: https://stackoverflow.com/questions/5863097/selecting-only-numeric-columns-from-a-data-frame
21. Information on useful, funny, and cool votes for Yelp reviews was taken from the following webpage: https://www.yelp-support.com/article/How-do-I-vote-a-review-as-useful-funny-or-cool?l=en_US
22. Information on the alpha attribute in ggplot2 was taken from the following webpage: http://ggplot2.tidyverse.org/reference/aes_colour_fill_alpha.html
23. Information on adding a column to an existing data frame in R was taken from the following webpage: https://discuss.analyticsvidhya.com/t/how-to-add-a-column-to-a-data-frame-in-r/3278
24. Information on plotting millions of data points in R was taken from the following webpage: http://r.789695.n4.nabble.com/Plotting-15-million-points-td1569914.html
25. Information on randomly splitting a data set in R was taken from the following webpage: https://stackoverflow.com/questions/17200114/how-to-split-data-into-training-testing-sets-using-sample-function
26. Information on changing the scale gradient colors was taken from the following webpage: http://ggplot2.tidyverse.org/reference/scale_gradient.html
27. Information on named colors in R was taken from the following webpage: http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
28. Information on changing the name of a plot generated by ggplot2 in R was taken from the following webpage: https://stackoverflow.com/questions/14622421/how-to-change-legend-title-in-ggplot
29. Information on P-values in the context of regression models was taken from the following webpage: http://blog.minitab.com/blog/adventures-in-statistics-2/how-to-interpret-regression-analysis-results-p-values-and-coefficients
30. Information on interpreting correlation coefficients was taken from the following webpage: http://www.dummies.com/education/math/statistics/how-to-interpret-a-correlation-coefficient-r/
31. Information on the skewness of a distribution was taken from the following webpage: https://study.com/academy/lesson/skewed-distribution-examples-definition-quiz.html
32. Information on placing multiple plots on one page with ggplot2 was taken from the following webpage: https://cran.r-project.org/web/packages/egg/vignettes/Ecosystem.html
33. Information on changing the font size of a plot title was taken from the following webpage: http://www.sthda.com/english/wiki/ggplot2-title-main-axis-and-legend-titles
34. General information on Yelp was taken from the following webpage: https://en.wikipedia.org/wiki/Yelp
35. General information on timelines was taken from the following webpage: https://en.wikipedia.org/wiki/Timeline