StackExchange Info Security Trends Project

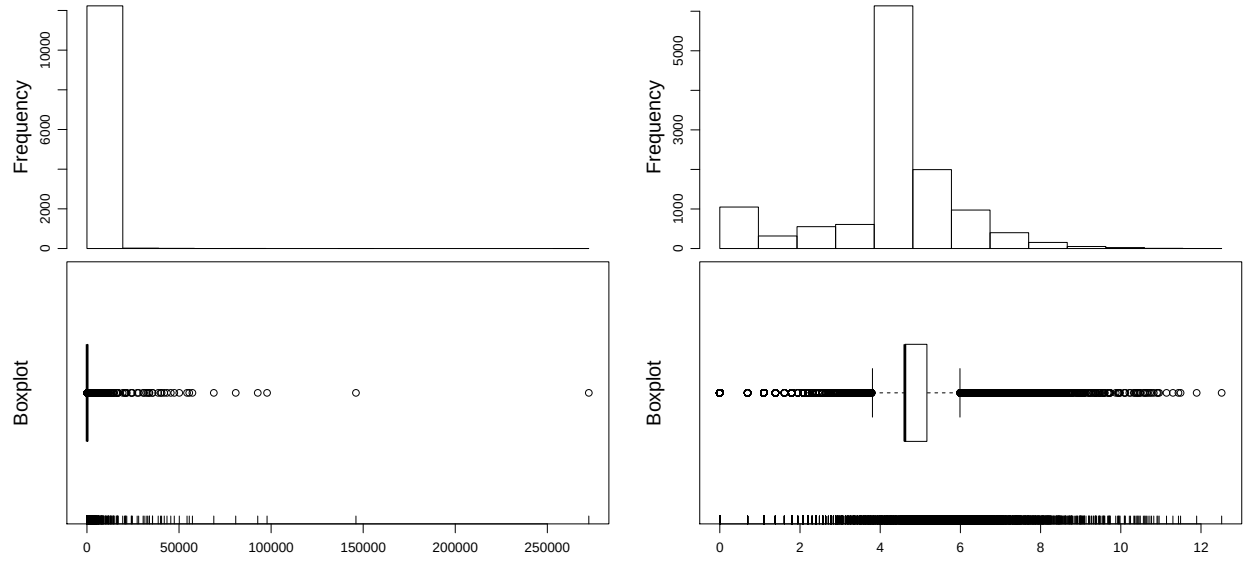
Egan McClave

February 5, 2019

All code is sorted at the end in an appendix on page 7.

Problem 1

Figure 1 contain a basic histogram and boxplot to visualize both the normal and log-transformed `Reputation` variable. In the standard distribution we can observe that `Reputation` is highly right skewed with many large positive values. In the log transformed distribution we can observe that `log(Reputation)` is still right skewed but not nearly as much. The values in Table 1 quantitatively support the visual observations of the distributions.



(a) User Reputation

(b) Log User Reputation

Figure 1: Graphical Analysis of Reputation over Time

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Sd.
User Reputation	1.00	101.00	101.00	448.76	175.00	272521.00	3766.19
Log User Reputation	0.00	4.62	4.62	4.46	5.16	12.52	1.83

Table 1: Reputation Summary Statistics

One exploration of the `Reputation` variable I would have liked to employ is interactive EDA with shiny apps. I feel that visualizing a skewed data, such as this variable, in an interactive environment could further assist with with understanding the distribution. Some examples of different features this application could utilize are interactive subsetting of ranges in the data and exploring the data with various transformations among other potential options.

Problem 2

A brief look at the reputation score for various users supports the idea that users with more questions and answers correlates to higher reputation scores. Additionally, users with more badges appear to have larger scores as well. There also appears to be some relationship between the combination of different types of badges, number of badges for each type and a users' reputation score. This basic intuition and personal observations will drive the direction of inquiry for this project.

$$\log(Reputation) = Bronze + Silver + DaysSinceCreation + DaysSinceLastAccess \tag{1}$$

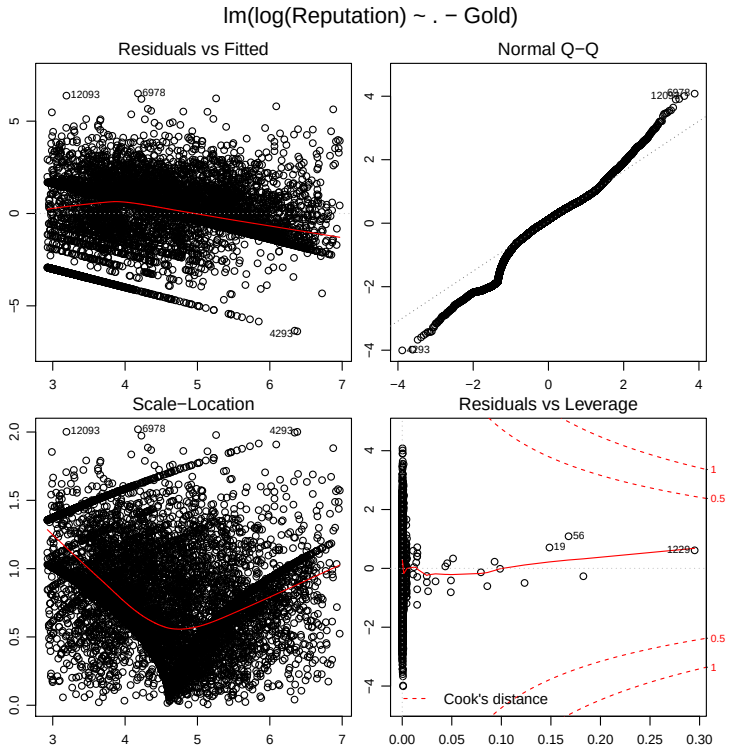


Figure 2: Linear Model Diagnostics

Figure 2 visualizes the diagnostic plots for the model in Equation 1. In general, the assumptions are mostly met. The residuals plot appears mostly symmetric around the line $y = 0$ with mostly constant variance. There is an exception of large values of `log(Reputation)`. For these range of values, there is some trend with the predictor. The assumptions for the QQ plot appear adequate as nothing deviates too much from the hypothetical distribution. The scale-location plot has nearly the same results as the residuals plot - a very small pattern within the random cloud of points. Lastly, the leverage plot reveals no outliers within the data. With these results, the model appears statistically sound based on the current data.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.6956	0.04	67.76	0.00
Silver	-0.0480	0.09	-0.51	0.61
Bronze	-0.0262	0.10	-0.27	0.79
SinceCreation	0.0015	0.00	51.14	0.00
SinceLastAccess	-0.0005	0.00	-17.18	0.00

Table 2: lm Coefficient Summary

Table 2 features the summary coefficients for the linear model. The log reputation for a user with no badges and recently created their account would have a 2.6955985. For a one badge increase in the number of silver badges, holding all else constant, there is a -4.8% unit change in reputation score. For a one badge increase in the number of bronze badges, holding all else constant, there is a -2.62% unit change in reputation score. For a one day increase since account creation, holding all else constant, there is a 0.15% unit change in reputation score. For a one day increase since their account was last accessed, holding all else constant, there is a 0.15% unit change in reputation score. The RMSE value for this model using a 80/20 split with training and testing is 1.6147035.

Due to the extreme skewness of the response variable, `log(Reputation)`, it is difficult to model the response with a linear model while still being clear and interpretable. The code appendix (page 7) contains all the models evaluated. In general, most of the models had similar diagnostic plots such as Figure 2. However, few other models had a high degree of interpretability. Because Equation 1 is so straightforward and clear this was selected as the final model. This is not to say that this model is the best way to predict `Reputation`; other methods might be more appropriate for this task depending on what other variables could be extracted for the data to be useful. However, that would entail more time on the analysis than available.

Problem 3

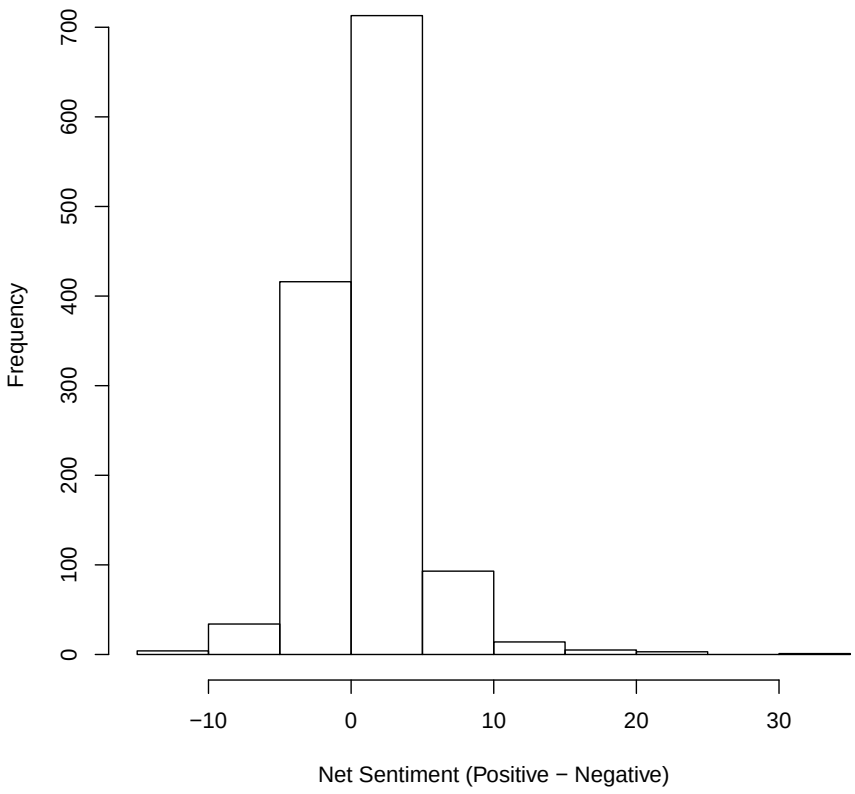


Figure 3: Histogram of Net Sentiment Across All Comments

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-14.00	0.00	1.00	1.50	3.00	31.00

Table 3: Net Sentiment Per Posting

Figure 3 illustrates the net sentiment for a given post using the `bing` lexicon. Overall, it appears there is a slight right shift in the distribution of net sentiment with a large peak centered around 0. Most users are usually neutral in their replies or comments to postings with the exception of a few comments. The values in Table 3 support the same conclusion. The first quartile of the data has a neutral sentiment with a minimum of -14 for "negative" sentiment and 31 for "positive" sentiment. I would consider taking the results of this analysis with a grain of salt because there are only so many words recorded in the `bing` lexicon. This makes the analysis exclude certain comments that do not have the sentiments of the words defined.

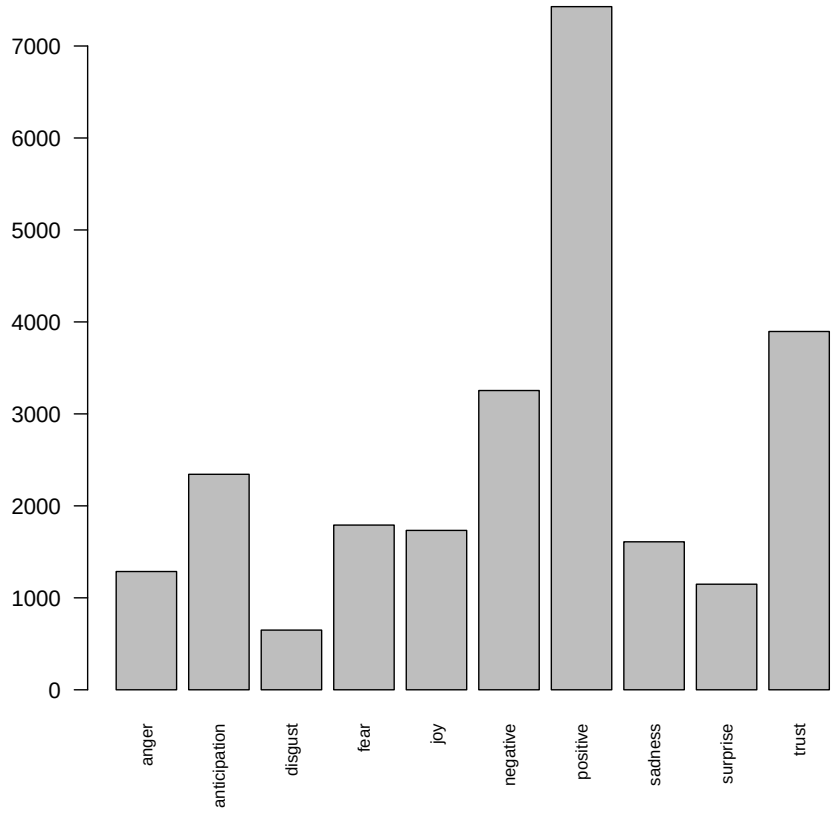


Figure 4: Histogram of Various Sentiment With All Words

anger	anticipation	disgust	fear	joy	negative	positive	sadness	surprise	trust
1286	2343	649	1791	1733	3254	7428	1609	1148	3896

Table 4: Count of Words Per Sentiment Label

Figure 4 illustrates the distribution of the sentiment across all words of comments using the `nrc` lexicon. Overall, there are an abundance of "positive" words used for comments. While there appears to be no remaining pattern to the distribution of sentiments across all words. The values in Table 4 support the same conclusion. There is an outstanding record of words with "positive" sentiment while most other sentiments are loosely around the same counts. I would consider taking the results of this analysis with a grain of salt because there are only so many words recorded in the `nrc` lexicon. This makes the analysis exclude certain comments that do not have the sentiments of the words defined which significantly limits the results.

Problem 4

Future analysis of post data might be useful to determining reputation. This would primarily relate to understanding how users with high reputation scores might be posing good questions to the community. This could be explored by using information in the 'Posts' dataframe. Utilizing the 'Score' variable for each

post and associating that with a user would be useful. Additionally, exploring the ‘AcceptedAnswersId‘ variable might provide insight on how users with high scores could have many accepted answers across the community.

Outside of the idea of predicting user reputation and characterizing comments, this dataset might provide useful information on understanding what people are most curious or interested about. Utilizing the ‘Posts‘ and ‘Votes‘ dataframes could be used Additionally, examining how questions are asked, again using the ‘Posts‘ dataframe, could elicit interesting content. This could be done with topic analysis among the different posts.

Code Appendix

Preliminary Setup

```
### Code chunk for preliminary set up purposes

# Load packages
pkgs <- c('xtable', 'knitr', 'tidyverse', 'tidytext', 'car')
invisible(lapply(pkgs, library, character.only=TRUE))

# Load data
if(!file.exists('./data.RData')) {
  all_files <- list.files('../data/')
  se_data <- lapply(all_files, function(file_name) {
    return(assign(tolower(strsplit(file_name, '.csv')),
                  read.csv(paste0('../data/', file_name))))
  })
  names(se_data) <- tolower(strsplit(all_files, '.csv'))

  # Save list of dataframes as RData object
  save(se_data, file = './data.RData')
} else {
  load('./data.RData')
}

# Custom color palette
cols <- c('#000000', '#999999', '#E69F00', '#56B4E9', '#009E73',
          '#FF0000', '#F0E442', '#0072B2', '#D55E00', '#CC79A7')

# Cache chunk options
knitr::opts_chunk$set(cache=TRUE, autodep=TRUE, cache.comments=TRUE)
```

Problem 1

```
# graphical analysis of reputation for each user
par(mfrow=c(2,1), oma=c(4,1,1,0.2) + 0.1, mar=c(0,3,0,0) + 0.1,
    mgp=c(2, 0.75, 0), cex.lab=1.3, cex.axis=0.9)

# User reputation
hist(se_data$users$Reputation, main='', xaxt='n', xlab='')
boxplot(se_data$users$Reputation, horizontal=T, ylab='Boxplot')
rug(se_data$users$Reputation)

# Log transformed user reputation
hist(log(se_data$users$Reputation), main='', xaxt='n', xlab='')
boxplot(log(se_data$users$Reputation), horizontal=T, ylab='Boxplot')
rug(log(se_data$users$Reputation))
```

```r
# statistical analysis of reputation for each user
stats <- with(se_data$users, rbind('User Reputation'=c(summary(Reputation), 'Sd.'=sd(Reputation)),
           'Log User Reputation'=c(summary(log(Reputation)), 'Sd.'=sd(log(Reputation)))))

print(xtable::xtable(stats, caption='Reputation Summary Statistics',
        label='tab:rep_stats'), table.placement='H')
```

Problem 2

```r
# Aggregate counts of badges across users
merge_data <- table(se_data$badges[,c('UserId', 'Class')]) %>%
  as.data.frame.matrix() %>%
  rownames_to_column(var='UserId') %>%
  dplyr::mutate(UserId=as.numeric(UserId)) %>%
  dplyr::rename('Gold'=`1`, 'Silver'=`2`, 'Bronze'=`3`) %>%

  # Merge aggregate badge counts with user information
  base::merge(se_data$users, by.x='UserId', by.y='AccountId', all=TRUE) %>%
  dplyr::filter(!is.na(Reputation)) %>%
  tidyr::replace_na(list(Gold=0, Silver=0, Bronze=0)) %>%

  # Convert dates to days since certain events
  dplyr::mutate(SinceCreation=Sys.Date() - as.Date(CreationDate),
                SinceLastAccess=Sys.Date() - as.Date(LastAccessDate)) %>%

  # Select appropriate variables for concise dataset
  dplyr::select(Reputation, everything(), -UserId, -Id, -DisplayName, -Location, -AboutMe, -Views,
                -UpVotes, -DownVotes, -WebsiteUrl, -ProfileImageUrl, -CreationDate, -LastAccessDate)

# Selecting testing and training set for model training and evaluating
i_train <- base::sample(nrow(merge_data), nrow(merge_data) * 0.8)
train <- merge_data[i_train,]
test <- merge_data[-i_train,]
```

```r
### Past models
# All basic variables
lm_1 <- lm(log(Reputation) ~ ., data=train)
par(mfrow=c(2,2)); plot(lm_1); par(mfrow=c(1,1))
alias(lm_1)

# Removal of Gold due to alias
lm_2 <- lm(log(Reputation) ~ . - Gold, data=train)
par(mfrow=c(2,2)); plot(lm_2); par(mfrow=c(1,1))
alias(lm_2)
vif(lm_2)

# Exploring interaction of Silver:Bronze
lm_3 <- lm(log(Reputation) ~ . - Gold + Silver:Bronze, data=train)
```

```
par(mfrow=c(2,2)); plot(lm_3); par(mfrow=c(1,1))
coef(summary(lm_3))
alias(lm_3)
vif(lm_3)

lm_4 <- lm(log(Reputation) ~ . - Gold - Silver - Bronze + Silver:Bronze, data=train)
par(mfrow=c(2,2)); plot(lm_4); par(mfrow=c(1,1))
coef(summary(lm_4))
alias(lm_4)
vif(lm_4)

# Exploring interaction between SinceCreation:SinceLastAccess
lm_5 <- lm(log(Reputation) ~ . - Gold + SinceCreation:SinceLastAccess, data=train)
par(mfrow=c(2,2)); plot(lm_5); par(mfrow=c(1,1))
coef(summary(lm_5))
alias(lm_5)
vif(lm_5)

lm_6 <- lm(log(Reputation) ~ . - Gold - Silver - Bronze + SinceCreation:SinceLastAccess, data=train)
par(mfrow=c(2,2)); plot(lm_6); par(mfrow=c(1,1))
coef(summary(lm_6))
alias(lm_6)
vif(lm_6)

lm_7 <- lm(log(Reputation) ~ . - Gold - SinceLastAccess + SinceCreation:SinceLastAccess, data=train)
par(mfrow=c(2,2)); plot(lm_7); par(mfrow=c(1,1))
coef(summary(lm_7))
alias(lm_7)
vif(lm_7)
```

```
# Creating final model
lm_final <- lm(log(Reputation) ~ . - Gold, data=train)

# Displaying diagnostic plots
par(mfrow=c(2,2), oma=c(1,1,1,1) + 0.1, mar=c(1,1,2,1) + 0.1,
    mgp=c(2, 0.75, 0), cex.lab=1.3, cex.axis=0.9)
plot(lm_final, main='')

# Calculating RMSE
rmse <- sqrt(mean((predict(lm_final, test) - log(test$Reputation))^2))
```

```
# Displaying summary of coefficients
coef_tab <- coef(summary(lm_final))
print(xtable::xtable(coef_tab, digits=c(0, 4, 2, 2, 2),
        caption='lm Coefficient Summary', label='tab:coef'),
      table.placement='H')
```

Problem 3

```
# tokenize the text into words
comments <- with(se_data$comments, data_frame(txt = as.character(Text), postId=PostId, score=Score)) %>%
  tidytext::unnest_tokens(word, txt)
```

```
### Using sentiment analysis to understand comments
# Positive vs negative sentiment
simple_sentiment <- suppressMessages(comments %>%
  inner_join(get_sentiments('bing'))) %>%
  count(postId, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(net_sentiment = positive - negative) %>%
  select(postId, negative, positive, net_sentiment)

hist(simple_sentiment$net_sentiment, xlab='Net Sentiment (Positive - Negative)',
     main='')

net_sentiment <- t(as.matrix(summary(simple_sentiment$net_sentiment)))
print(xtable::xtable(net_sentiment, caption='Net Sentiment Per Posting',
        label='tab:simple_sentiment'),
      table.placement='H', include.rownames=FALSE)
```

```
# Various emotional sentiment
robust_sentiment <- suppressMessages(comments %>%
  inner_join(get_sentiments('nrc'))) %>%
  count(postId, sentiment) %>%
  spread(sentiment, n, fill = 0)

various_sentiment <- robust_sentiment %>%
  select(-postId) %>%
  apply(2, sum) %>%
  as.table()

barplot(various_sentiment, cex.names=0.75, las=2)

print(xtable::xtable(t(as.matrix(various_sentiment)), digits=0,
        caption='Count of Words Per Sentiment Label', label='tab:various_sentiment'),
      table.placement='H', include.rownames=FALSE)
```