# Homework 02 Data Handling, Graphics, More R

Due by 11:59pm, Friday, 1.31.25

S&DS 230/530/ENV 757

**(1) Obama Tweets: Retweets vs. Favorites** A `.CSV` file containing Tweets from former President Barack Obama can be downloaded HERE. The data is sorted by date, most recent at the top.

The variables (columns) are:

- `text`: the body of the tweet
- `date`: when the tweet was sent, original format
- `date2`: when the tweet was sent, JUST the day (not the time of day)
- `retweet_count`: how many people retweeted this tweet
- `favorite_count`: how many people favorited this tweet
- `is_retweet`: whether or not this tweet is a retweet of someone else's tweet
- `source`: device used to send the tweet
- `is_quote`: is the tweet a quote of someone else
- `is_reply`: is the tweet a reply to someone else
- `possibly_sensitive`: does the tweet possibly contain sensitive material

You can read more about retweets vs. replies HERE.

There are two ways in which other Twitter users can indicate support for a tweet: *favoriting* and *retweeting*. For example, if a tweet has `favorite_count` = 5 and `retweet_count` = 10, then this suggests that 5 people favorited the tweet (saved it) and 10 people retweeted it (broadcasted it to their followers).

(1.1) Insert an R code chunk right below this that imports the data into a dataframe called `recent`. Note that the data is sorted in reverse time order. Get the header names of `recent` to confirm that the data imported correctly. Look at the first few rows of the data and the final few rows of the data. Also get the dimension of `recent`. What is the date range of the tweets? How many tweets does this dataset include?

```
recent <- read.csv('http://reuningscherer.net/S&DS230/data/ObamaTweetsNEW.csv')
recent$X <- NULL
dim(recent)
```

```
## [1] 2000   10
```

```
names(recent)
```

```
##  [1] "text"               "date"               "source"
##  [4] "is_quote"           "is_retweet"         "is_reply"
##  [7] "favorite_count"     "retweet_count"      "possibly_sensitive"
## [10] "date2"
```

```
head(recent, 5)
```

```
##
## 1
## 2
## 3 This week, Illinois joined states across the country in passing a historic gun violence prevention
## 4
## 5
##                   date             source is_quote is_retweet is_reply
## 1 2023-01-13 13:30:43 Twitter for iPhone    FALSE      FALSE     TRUE
## 2 2023-01-13 13:30:43 Twitter for iPhone    FALSE      FALSE    FALSE
## 3 2023-01-12 08:30:25 Twitter for iPhone    FALSE      FALSE    FALSE
## 4 2023-01-11 10:45:56 Twitter for iPhone    FALSE      FALSE    FALSE
## 5 2023-01-11 09:31:33 Twitter for iPhone    FALSE      FALSE    FALSE
##   favorite_count retweet_count possibly_sensitive      date2
## 1           4045           847              FALSE 2023-01-13
## 2          15256          1563              FALSE 2023-01-13
## 3          28154          3760                 NA 2023-01-12
## 4              0           347                 NA 2023-01-11
## 5              0          3145                 NA 2023-01-11
```

```
tail(recent, 5)
```

```
##
## 1996                                    Retweet if you believe it's time for the United States to #L
## 1997                                         Speak up for a fair hearing for Judge Merrick Garland:
## 1998                                                                                  This is unprecedented.
## 1999 Add a comment if you agree: American workers shouldn't have to choose between their health and
## 2000                Working families in America should have the basic security of paid sick leave. #
##                      date             source is_quote is_retweet is_reply
## 1996 2016-04-11 08:34:06 Twitter Web Client    FALSE      FALSE    FALSE
## 1997 2016-04-08 14:23:02 Twitter Web Client    FALSE      FALSE    FALSE
## 1998 2016-04-08 11:52:17 Twitter Web Client    FALSE      FALSE    FALSE
## 1999 2016-04-08 10:04:33 Twitter Web Client    FALSE      FALSE    FALSE
## 2000 2016-04-08 08:45:49 Twitter Web Client    FALSE      FALSE    FALSE
##      favorite_count retweet_count possibly_sensitive      date2
## 1996           6015          3184              FALSE 2016-04-11
## 1997           2271           762              FALSE 2016-04-08
## 1998           4388          1890              FALSE 2016-04-08
## 1999           3141           724              FALSE 2016-04-08
## 2000           7082          1732              FALSE 2016-04-08
```

*Since the dataframe has 2000 columns, this dataset contains 2000 tweets. The first column is a tweet from January 13th, 2023, and the last column is a tweet from April 8th, 2016, so that is the date range.*

(1.2) Create a table that shows how many of Tweets came from each source and call this object `table1`. Show the results of `table1`. Write a single line that calculates the proportion of Tweets that were from Twitter Web Client, rounds this value to two decimal places, multipies the results by 100, and pastes on a "%" symbol. There should be no space between the number and the '%' symbol.

```
table1 <- table(recent$source)
table1
```

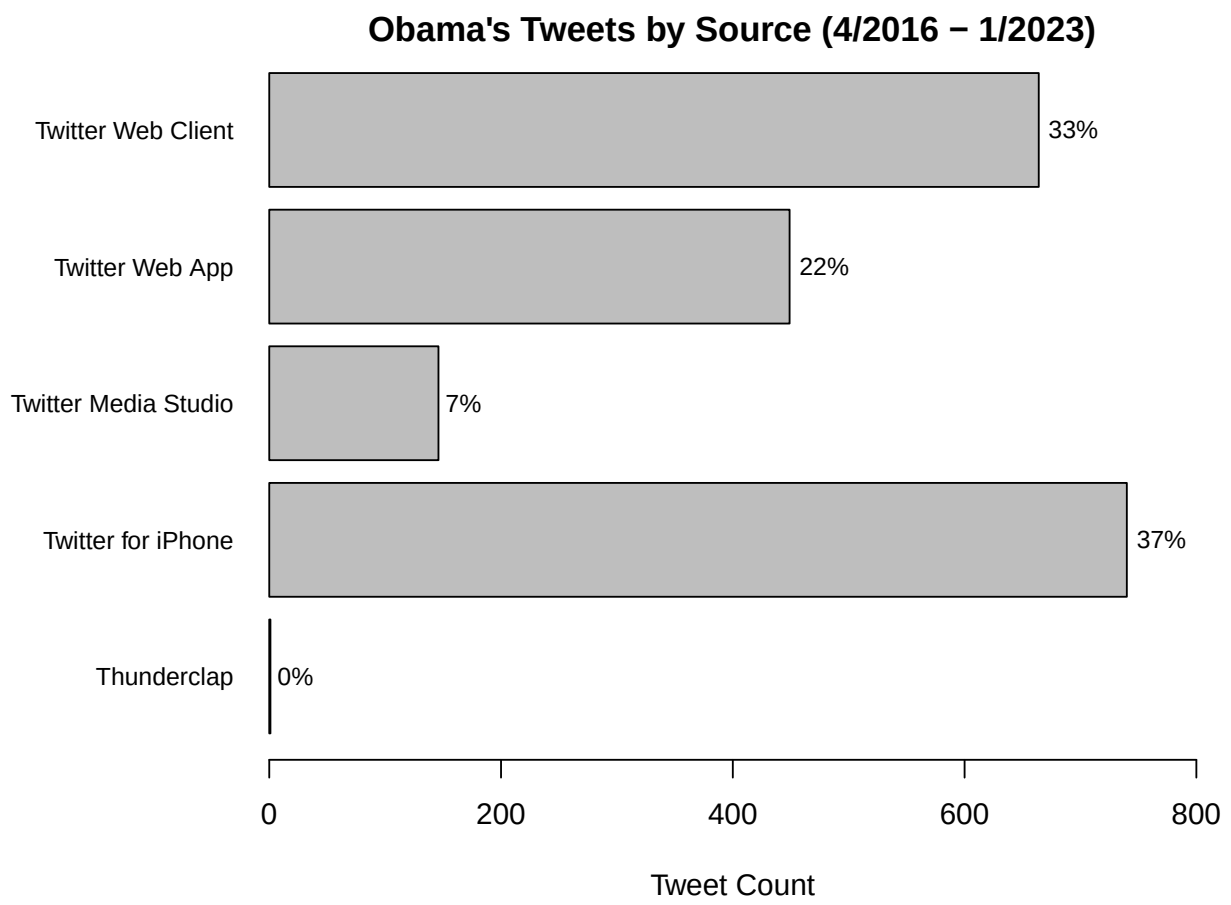
```
##
##          Thunderclap   Twitter for iPhone Twitter Media Studio
##                    1                  740                  146
##      Twitter Web App   Twitter Web Client
##                  449                  664
```

```
paste(round(nrow(recent[recent$source == "Twitter Web Client",])
            / nrow(recent), digits=2) * 100, "%", sep="")
```

```
## [1] "33%"
```

(1.3) Create a barplot that shows the number of tweets from each source. The labels of the barplot should also contain the whole number percentages for each tweet source (i.e. Thunderclap (14%) as an example (this isn't the correct percentage). Take the time to format your graph, and make sure the bars are horizontal. You'll want to include the commented line of code below AND you'll want to use the barplot option `cex.names = .6`. Write a comment in your code that explaines what this option does.

```
par(mar=c(4,8,1,1)) # change the margins to fit the horizontal labels
barplot1 <- barplot(table1,
                    horiz=TRUE, # make bars horizontal
                    cex.names=.8, # change the font size/proportions
                    las=1, # make labels horizontal
                    main="Obama's Tweets by Source (4/2016 - 1/2023)",
                    xlim=c(0, 800),
                    xlab="Tweet Count")
text(x=table1,
     y=barplot1 + 0,
     adj=-0.2,
     cex=.8,
     labels=paste0(round(proportions(table1), digits=2) * 100, "%"))
```




(1.4) Get summary statistics for both `favorite_count` and `retweet_count`. Make histograms for each of these two variables as well. Put a title on each histogram, label the horizontal axis, and make the bars red. How would you describe the shape of these distributions (use words like 'symmetric' or 'skewed', or perhaps the name of some distribution that has a similar shape . . .)?

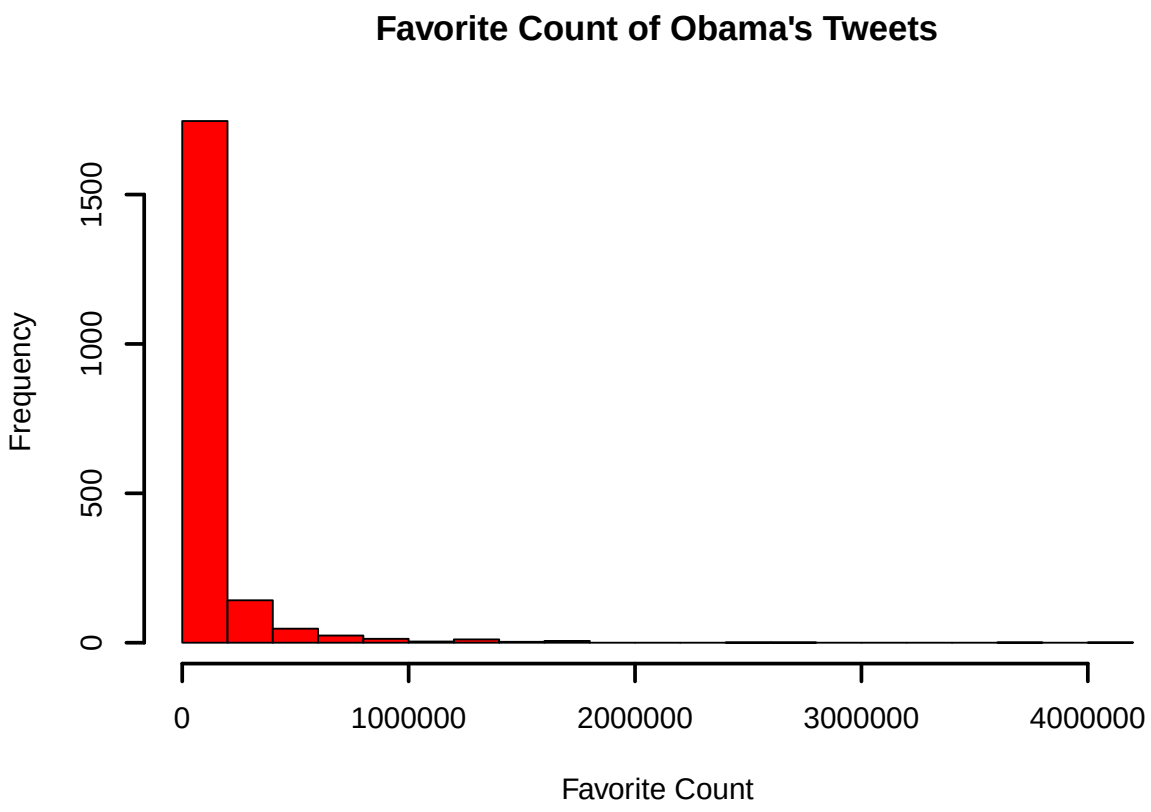
```
summary(recent$favorite_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    3969   15358   96327   78010 4010967
```

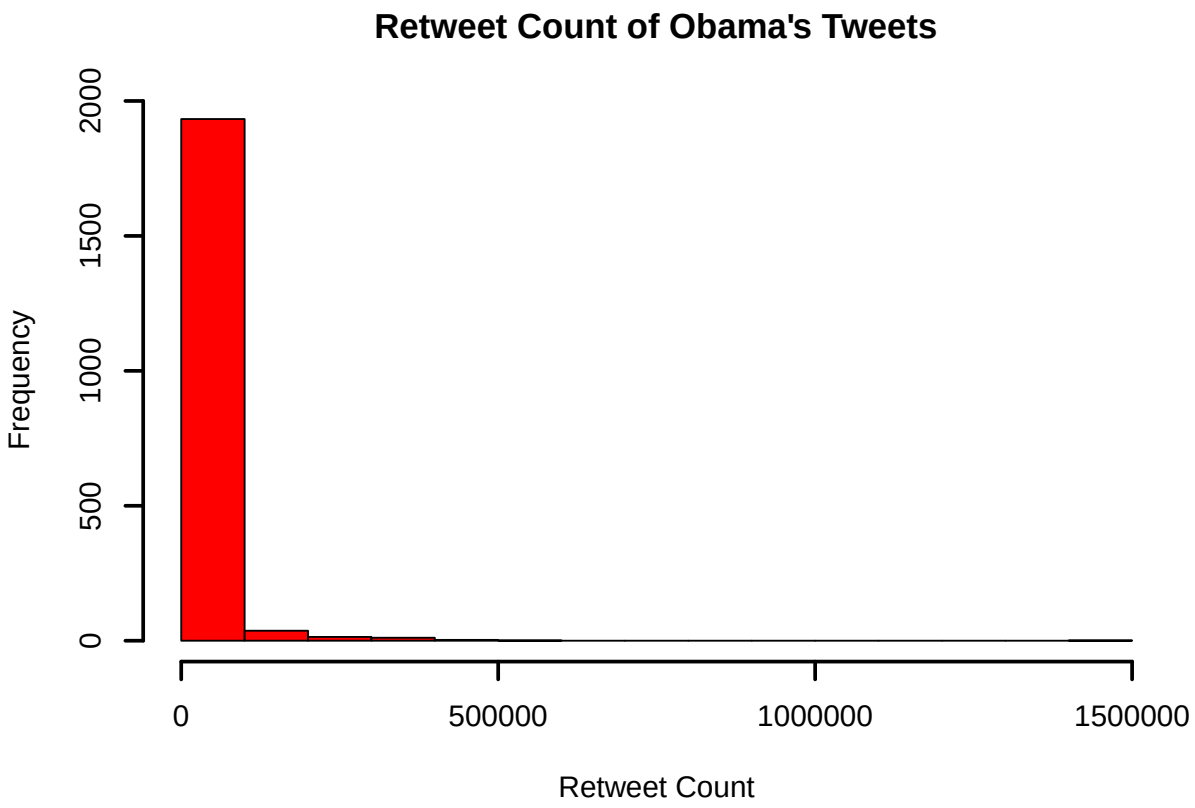
```
summary(recent$retweet_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     198    1164    3071   16841   12153 1435375
```

```
options(scipen=5)
hist(recent$favorite_count,
     col='red',
     lwd=2,
     xlab='Favorite Count',
     main="Favorite Count of Obama's Tweets",
     breaks=20)
```




```
hist(recent$retweet_count,
     col='red',
     lwd=2,
     xlab='Retweet Count',
     main="Retweet Count of Obama's Tweets",
     breaks=20)
```




*Both the distributions of the favorite counts and the retweet counts of Obama's tweets are right skewed and appear to follow a geometric distribution.*

(1.5) Get summary statistics for `retweet_count` FIRST for the observations for which `is_quote` is `TRUE`, then for the observations for which 'is_quote`isFALSE`. Compare the medians of these two distributions - what do you observe?

```
summary(recent[recent["is_quote"] == TRUE,]$retweet_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     500    2053    6721   19566   25733  208778
```

```
summary(recent[recent["is_quote"] == FALSE,]$retweet_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     198    1107    2868   16566   10869 1435375
```

*The median retweet count for Obama's tweets that are quotes is higher than the median retweet count for his tweets that are not quotes.*

(1.6) Create a new dataframe called `recent_NoQuote` that contains all data from `recent` for which `is_quote` is `FALSE` (essentially, we're removing quotes and only looking at strictly original texts). USE THIS NEW DATAFRAME for the remainder of this problem set. Get the dimension of this dataframe. Compare this to a table of `is_quote` for the entire dataset to make sure the remaining number of rows (and columns) is correct.

Finally, make two new variables as a part of `recent_NoQuote` which will be the log transformations of `favorite_count` and `retweet_count`. Call these variables `logfavCnt` and `logreCnt`, respectively. The function you want to take log is called `log()`.

```
recent_NoQuote <- recent[recent["is_quote"] == FALSE,]
dim(recent_NoQuote)
```

```
## [1] 1817   10
```

```
table(recent$is_quote)
```

```
##
## FALSE  TRUE
##  1817   183
```

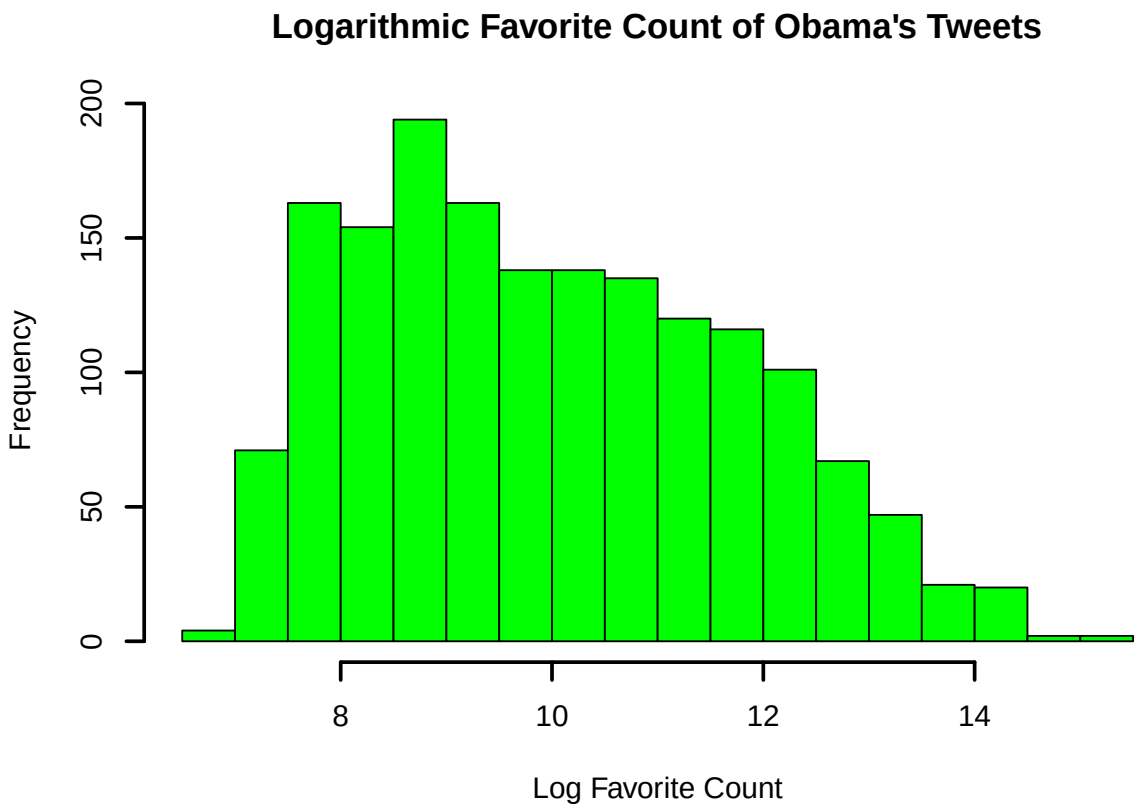
```
recent_NoQuote$logfavCnt <- log(recent_NoQuote$favorite_count)
recent_NoQuote$logreCnt <- log(recent_NoQuote$retweet_count)
head(recent_NoQuote, 10)
```

```
##
## 1
## 2
## 3                                This week, Illinois joined states across the country in passing a
## 4
## 5
## 6            If you haven't already, I hope you'll take some time to watch Descendant on @Netflix. I
## 7              Last fall, visual artist Adam Davis captured tintype photos of the descendants of Afri
## 8
## 9                                                                                 The entire world has a
## 10 15 years ago today, our campaign won the Iowa caucuses. I'll always be grateful to the people who
##                    date               source is_quote is_retweet is_reply
## 1   2023-01-13 13:30:43   Twitter for iPhone    FALSE      FALSE     TRUE
## 2   2023-01-13 13:30:43   Twitter for iPhone    FALSE      FALSE    FALSE
## 3   2023-01-12 08:30:25   Twitter for iPhone    FALSE      FALSE    FALSE
## 4   2023-01-11 10:45:56   Twitter for iPhone    FALSE      FALSE    FALSE
## 5   2023-01-11 09:31:33   Twitter for iPhone    FALSE      FALSE    FALSE
## 6   2023-01-10 14:37:04      Twitter Web App    FALSE      FALSE     TRUE
## 7   2023-01-10 14:37:03      Twitter Web App    FALSE      FALSE    FALSE
## 8   2023-01-10 14:23:22   Twitter for iPhone    FALSE      FALSE    FALSE
## 9   2023-01-09 16:45:16   Twitter for iPhone    FALSE      FALSE    FALSE
## 10 2023-01-03 07:30:22 Twitter Media Studio    FALSE      FALSE    FALSE
##    favorite_count retweet_count possibly_sensitive      date2 logfavCnt
## 1            4045          847              FALSE 2023-01-13  8.305237
## 2           15256         1563              FALSE 2023-01-13  9.632728
## 3           28154         3760                 NA 2023-01-12 10.245445
## 4               0          347                 NA 2023-01-11      -Inf
## 5               0         3145                 NA 2023-01-11      -Inf
## 6            8404         1310              FALSE 2023-01-10  9.036463
## 7           17416         1620              FALSE 2023-01-10  9.765145
## 8               0         2021                 NA 2023-01-10      -Inf
## 9          182676        18580                 NA 2023-01-09 12.115469
## 10          20812         2275              FALSE 2023-01-03  9.943285
```

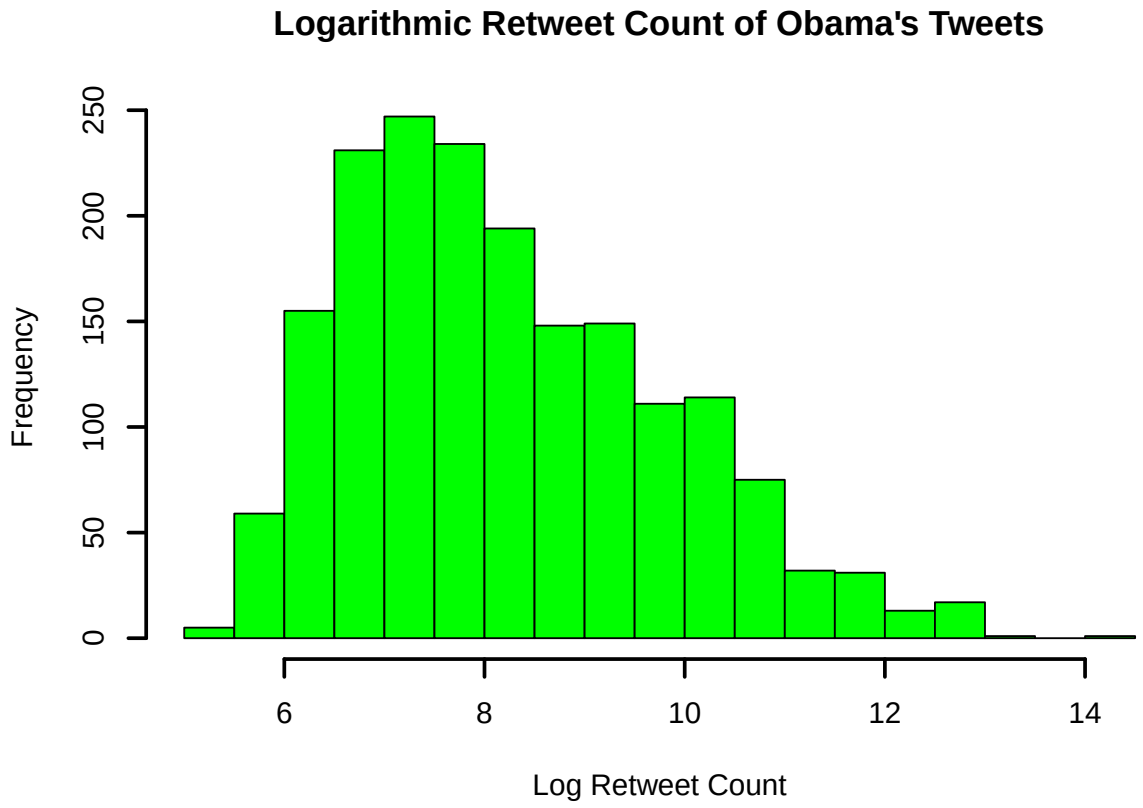
```
##    logreCnt
## 1  6.741701
## 2  7.354362
## 3  8.232174
## 4  5.849325
## 5  8.053569
## 6  7.177782
## 7  7.390181
## 8  7.611348
## 9  9.829841
## 10 7.729735
```

(1.7) Make histograms of these two new log-scale variables. Put a title on each histogram, label the horizontal axis, and make the bars green How would you describe the shape of these transformed distributions (use words like 'symmetric' or 'skewed')?

```
hist(recent_NoQuote$logfavCnt,
     col='green',
     lwd=2,
     xlab='Log Favorite Count',
     main="Logarithmic Favorite Count of Obama's Tweets",
     breaks=20)
```



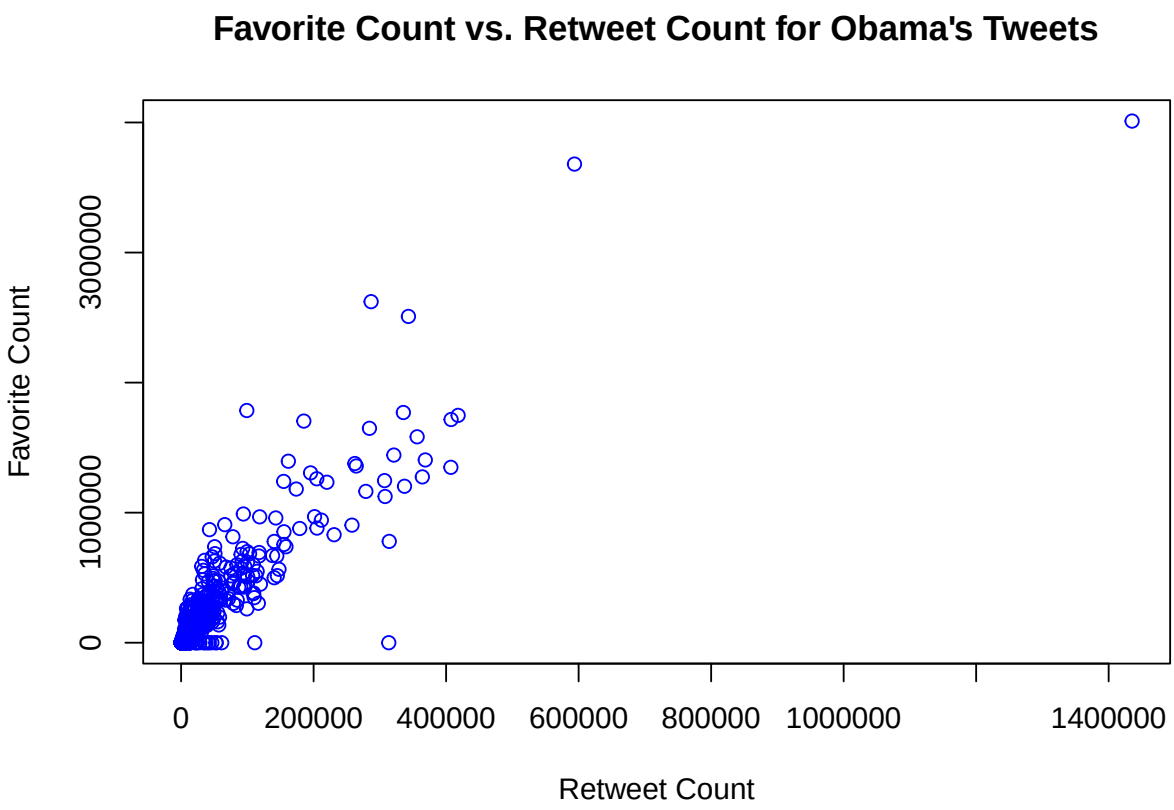
```r
hist(recent_NoQuote$logreCnt,
     col='green',
     lwd=2,
     xlab='Log Retweet Count',
     main="Logarithmic Retweet Count of Obama's Tweets",
     breaks=20)
```




*Both of the logarithmic distributions are unimodal and right skewed.*

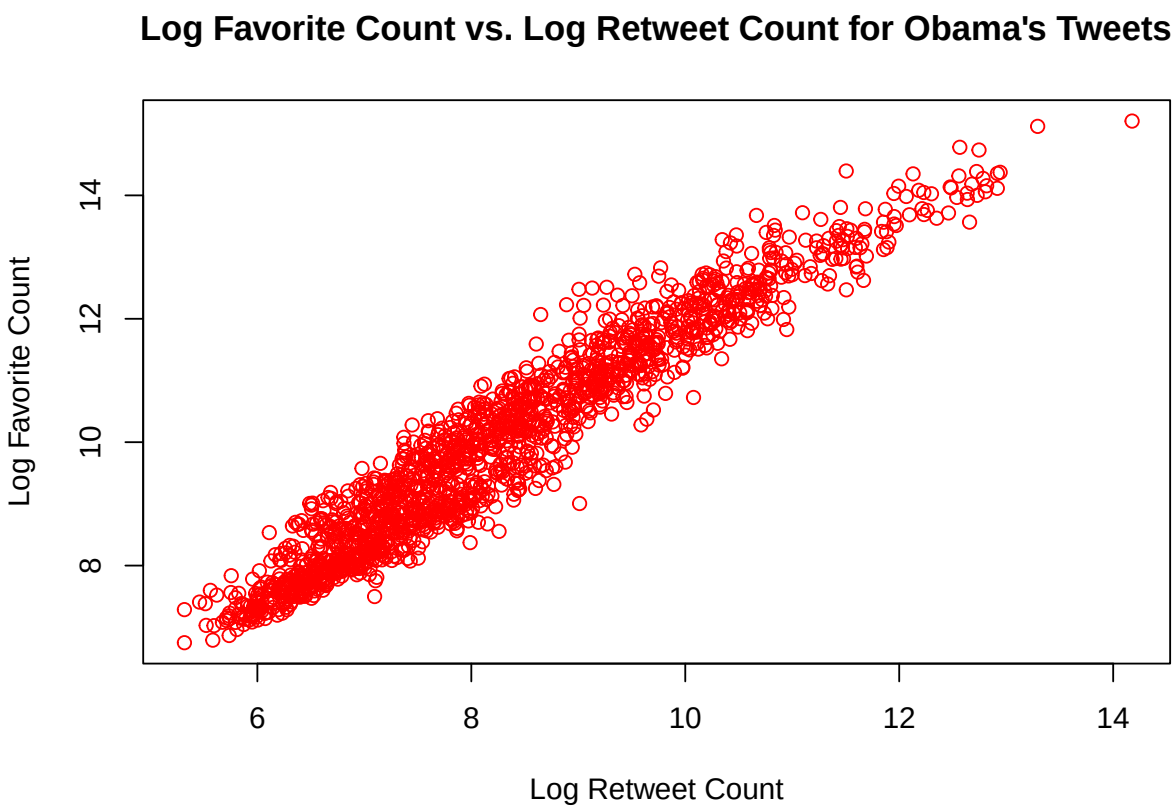
(1.8) Make a plot of the number of times that each tweet was favorited vs. the number of times a tweet was retweeted. Put `favorite_count` on the y-axis and `retweet_count` on the x-axis. Label your axes, put on a main title, and make the plot characters blue.

```r
plot(recent_NoQuote$retweet_count,
     recent_NoQuote$favorite_count,
     xlab="Retweet Count",
     ylab="Favorite Count",
     main="Favorite Count vs. Retweet Count for Obama's Tweets",
     col="Blue")
```




(1.9) Repeat part (1.8) but use the log-transformed variables. Label your axes, put on a main title, and make the plot characters red. How does the scatterplot on the log-scale compare to the scatterplot on the raw scale? Which one do you prefer?

```
plot(recent_NoQuote$logreCnt,
     recent_NoQuote$logfavCnt,
     xlab="Log Retweet Count",
     ylab="Log Favorite Count",
     main="Log Favorite Count vs. Log Retweet Count for Obama's Tweets",
     col="red")
```




*The data points are more evenly distributed across the entire scatterplot, which makes it easier to see an underlying trend compared to the previous scatterplot where the majority of data points were amassed in one area.*

(1.10) Create two new variables on the `recent_NoQuote` dataframe called `year` and `month` that will contain respectively the year and month the tweet was created. You'll need to look up how to use the function `substr()`. You'll also need to use the `as.numeric()` function to make sure that both new variables are numbers. Show the first 20 observations for each resulting variable.

```
recent_NoQuote$year <- as.numeric(substr(recent_NoQuote$date2, 1, 4))
recent_NoQuote$month <- as.numeric(substr(recent_NoQuote$date2, 6, 7))
head(recent_NoQuote$year, 20)
```

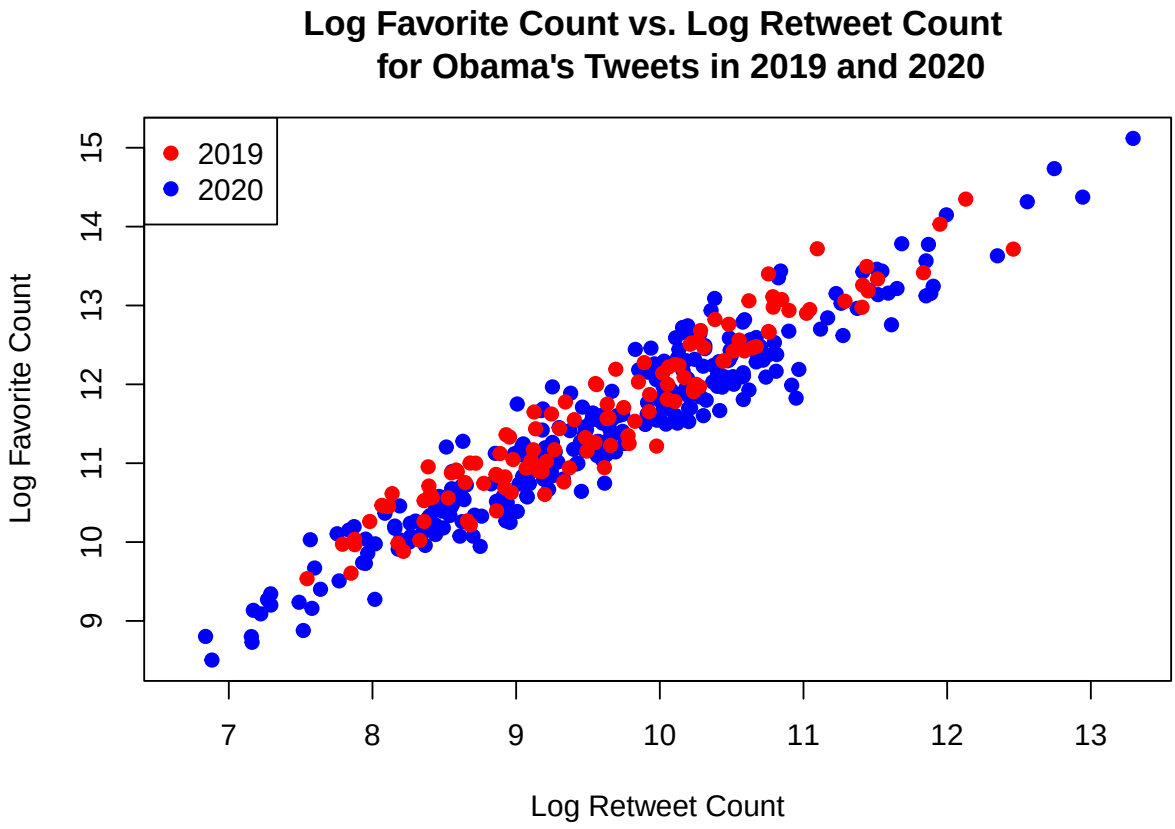
```
##  [1] 2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 2022 2022 2022 2022
## [16] 2022 2022 2022 2022 2022
```

```
head(recent_NoQuote$month, 20)
```

```
##  [1]  1  1  1  1  1  1  1  1  1  1  1 12 12 12 12 12 12 12 12 12
```

(1.11) Repeat part (1.9) BUT only for 2019 and 2020. First, create a dataframe called `recent_3` that only has observations from the specified years. You might want to use the `%in%` operator on your newly created variable year. Use this new dataframe to make your plot. Use the graphics option `pch = 19` to get solid round points, and make sure you have different colors for each of the two years. Finally, make sure your plot has a legend.

```r
recent_3 <- recent_NoQuote[recent_NoQuote$year %in% c(2019, 2020),]
plot(recent_3$logreCnt,
     recent_3$logfavCnt,
     xlab="Log Retweet Count",
     ylab="Log Favorite Count",
     main="Log Favorite Count vs. Log Retweet Count
     for Obama's Tweets in 2019 and 2020",
     pch=19,
     col=ifelse(recent_3$year == 2019, "red", "blue"))
legend(x = "topleft",
       legend = c(2019, 2020),
       col = c("red","blue"),
       pch = 19)
```




(1.12) Write no more than three sentences that describe what you see. Does the pattern appear any different between 2019 and 2020?

*For the most part, the relationship between log retweet count and log favorite count seems to be the same between 2019 and 2020. 2020 seems to have more variation with the outliers being farther out than in 2019.*