DS 710 Final Project

Patrick Christ

12/16/2020

Load in Libraries and Data from csvs

```
library(ggformula)
```

```
## Loading required package: ggplot2

## Loading required package: ggstance

##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

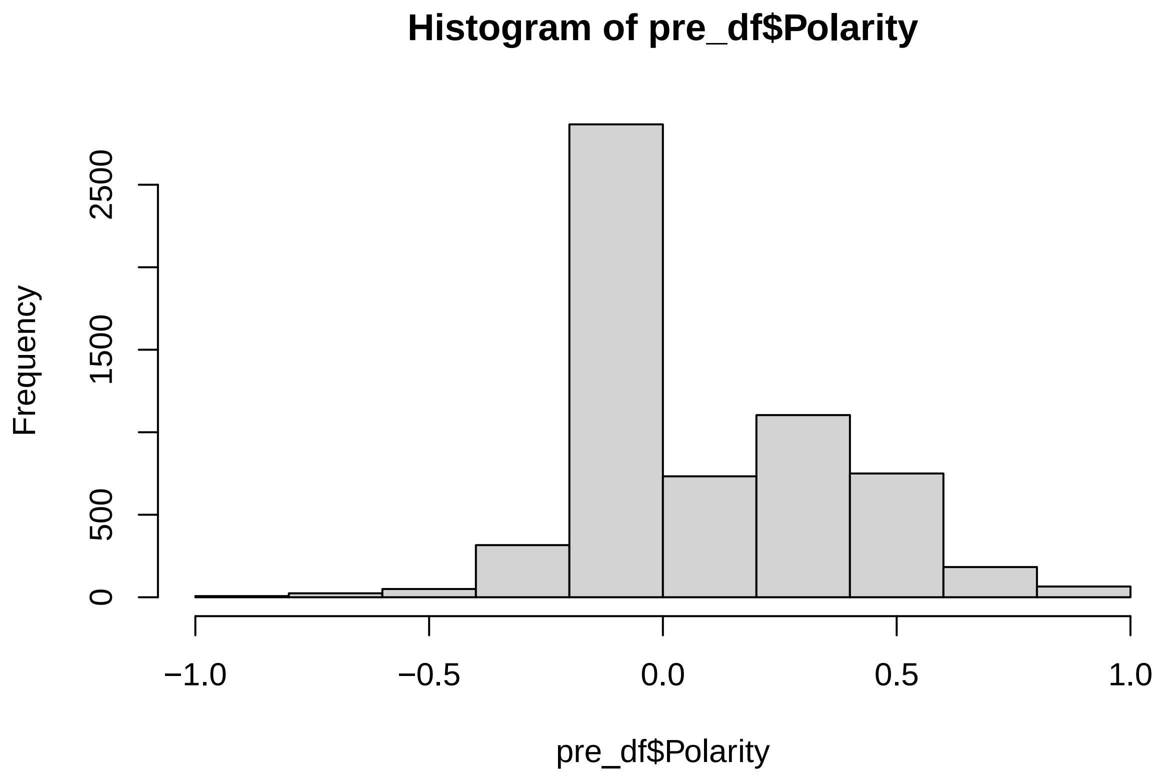
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

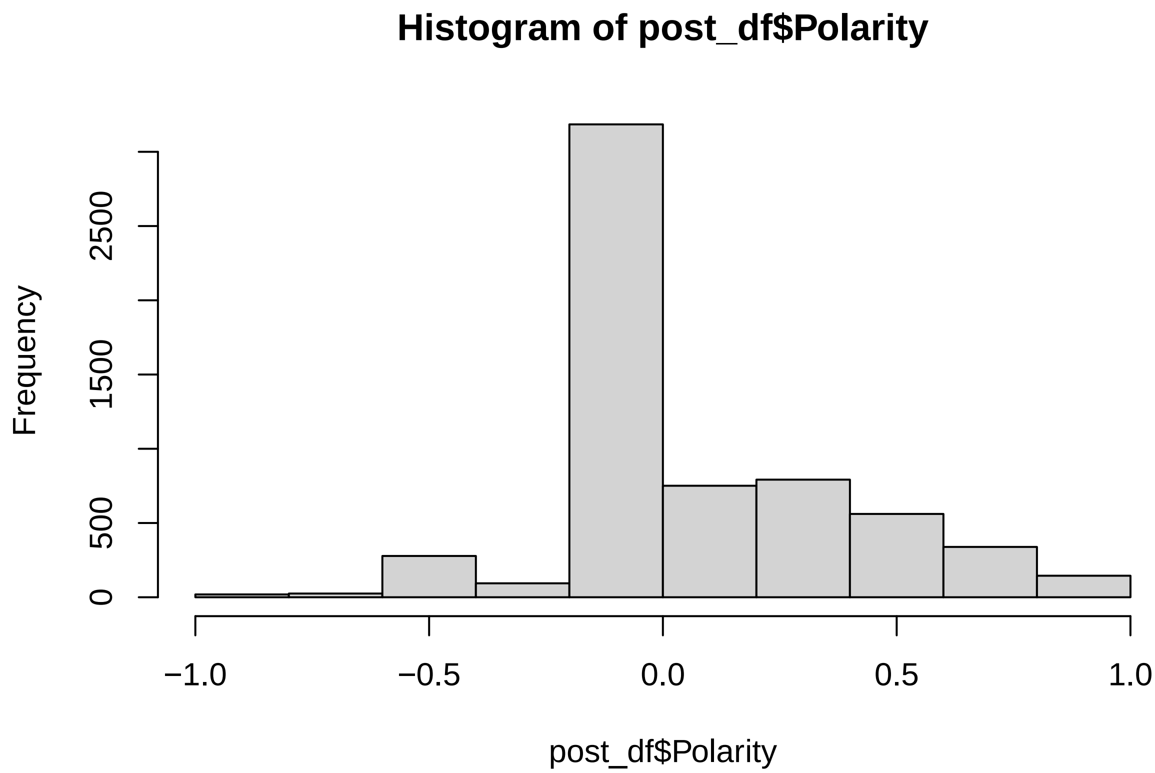
```
#Load in our data to a few different data frames
pre_df = read.csv("PrePatchTweets.csv")
post_df = read.csv("PostPatchTweets.csv")
ffxiv_tweets_df = read.csv("FFXIVTweets.csv")
```

Preliminary Look at Data

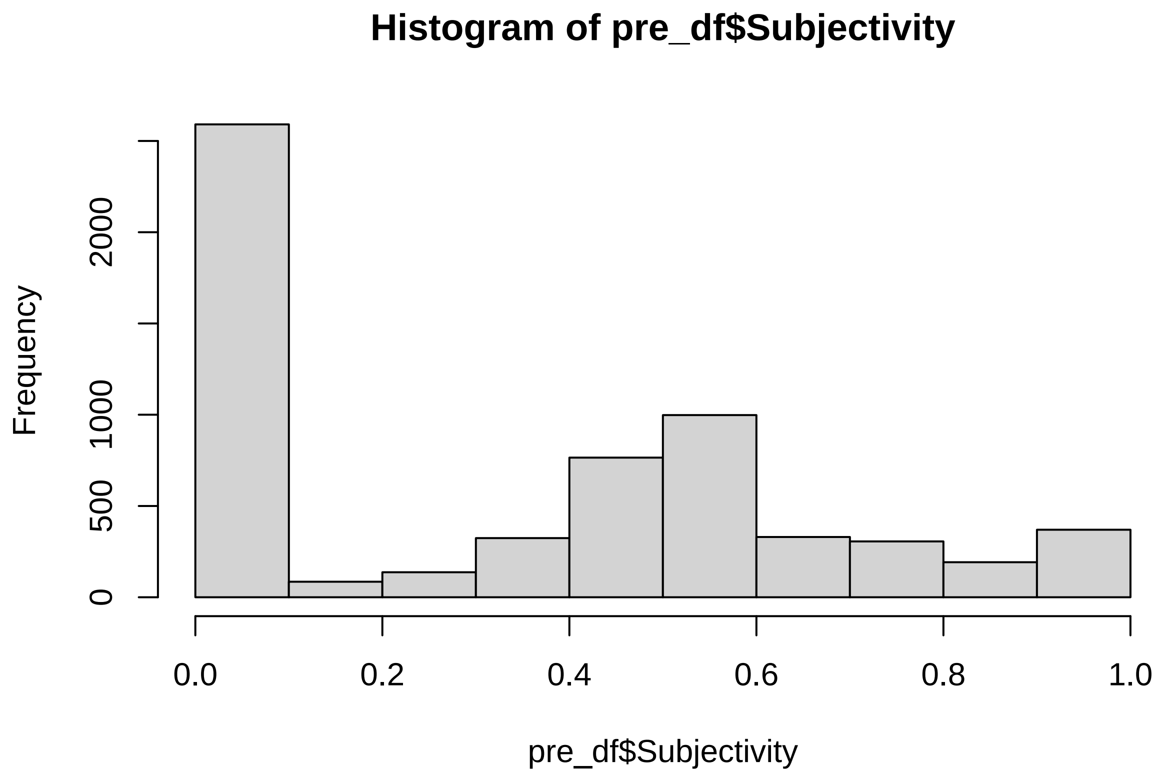
```
#Look at the distribution of my data
hist(pre_df$Polarity)
```



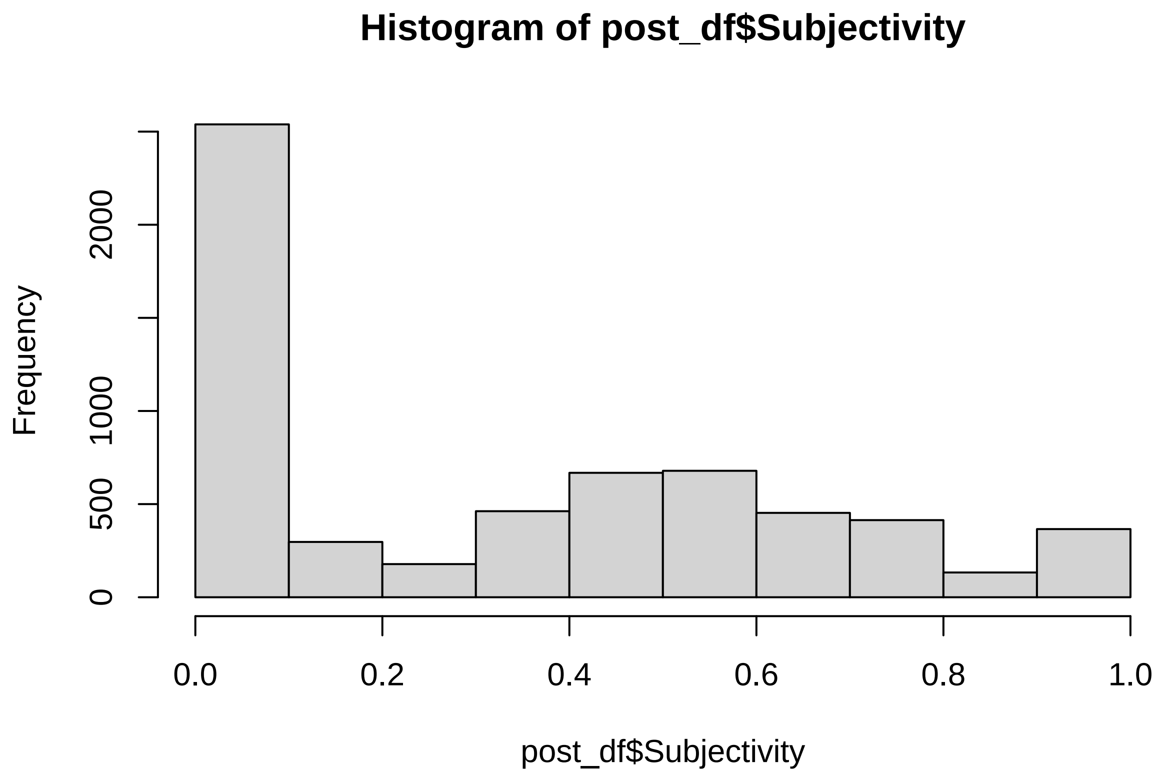

```r
hist(post_df$Polarity)
```




```r
hist(pre_df$Subjectivity)
```

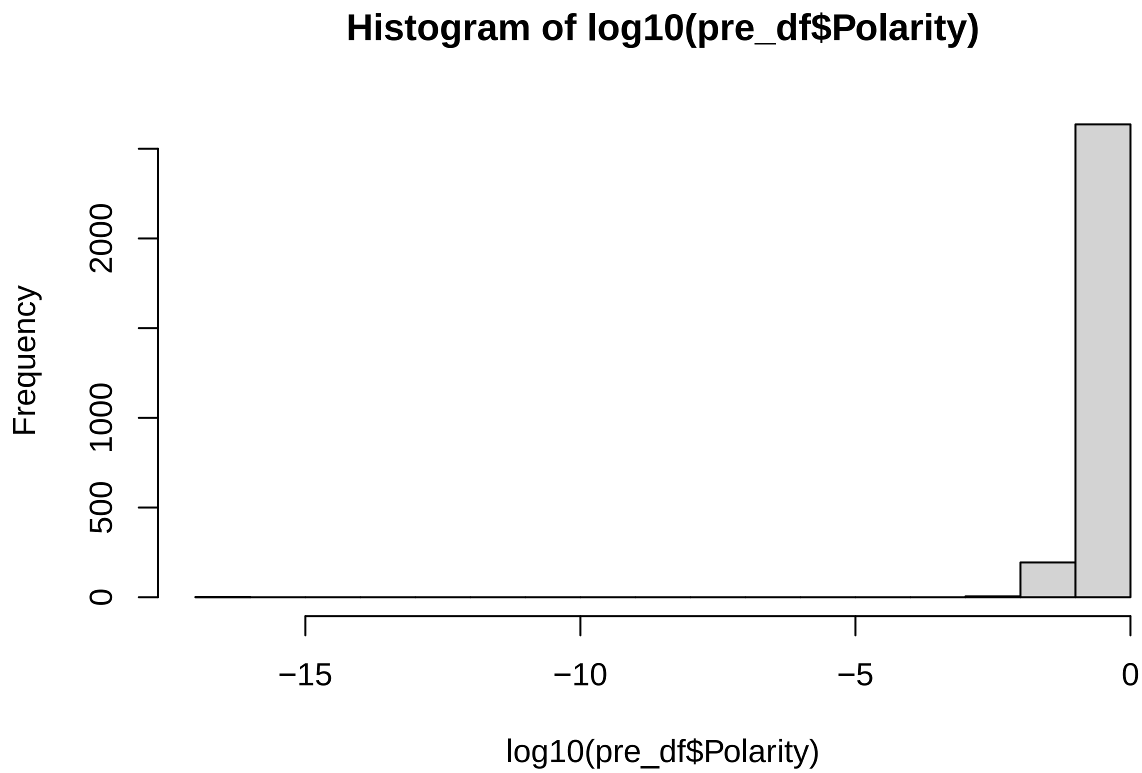



```r
hist(post_df$Subjectivity)
```



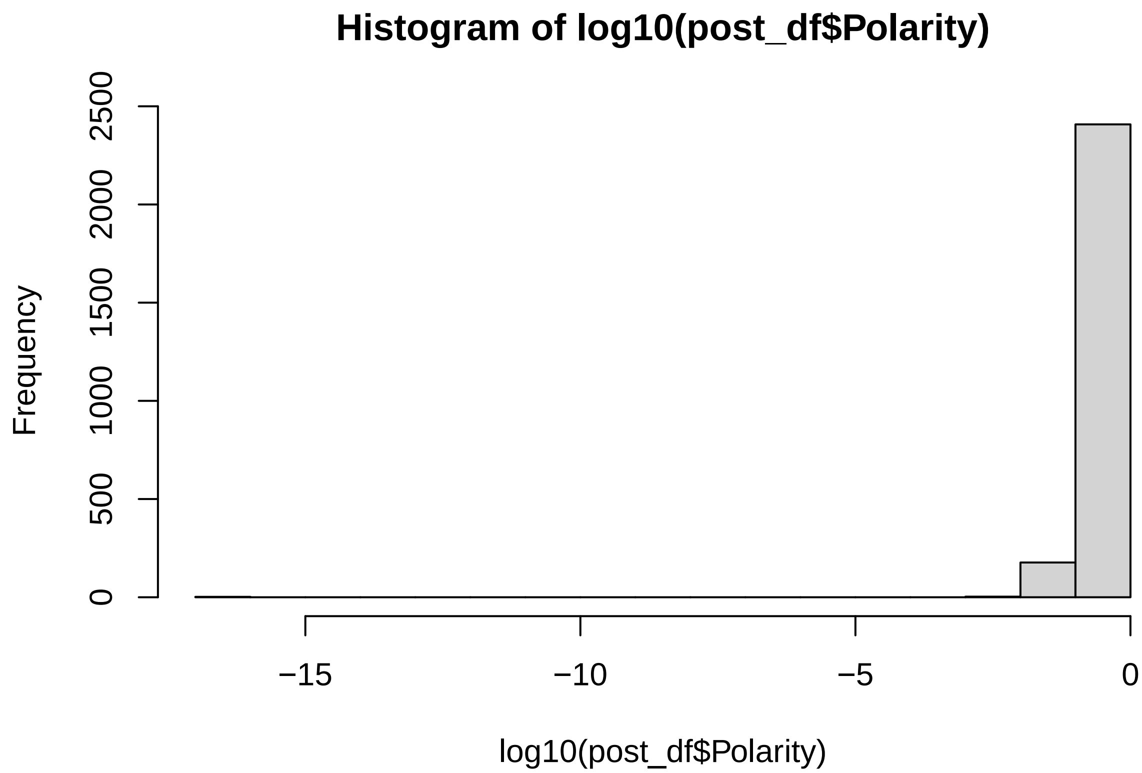

```r
#It does not look normally distributed, so I want to look at a transformation.
hist(log10(pre_df$Polarity))
```

```
## Warning in hist(log10(pre_df$Polarity)): NaNs produced
```

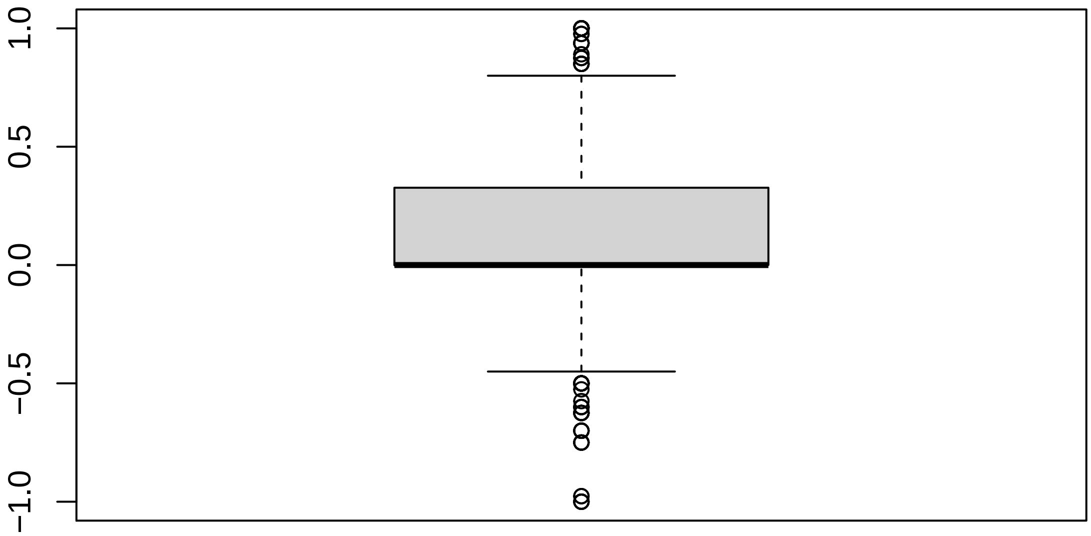



```
hist(log10(post_df$Polarity))
```

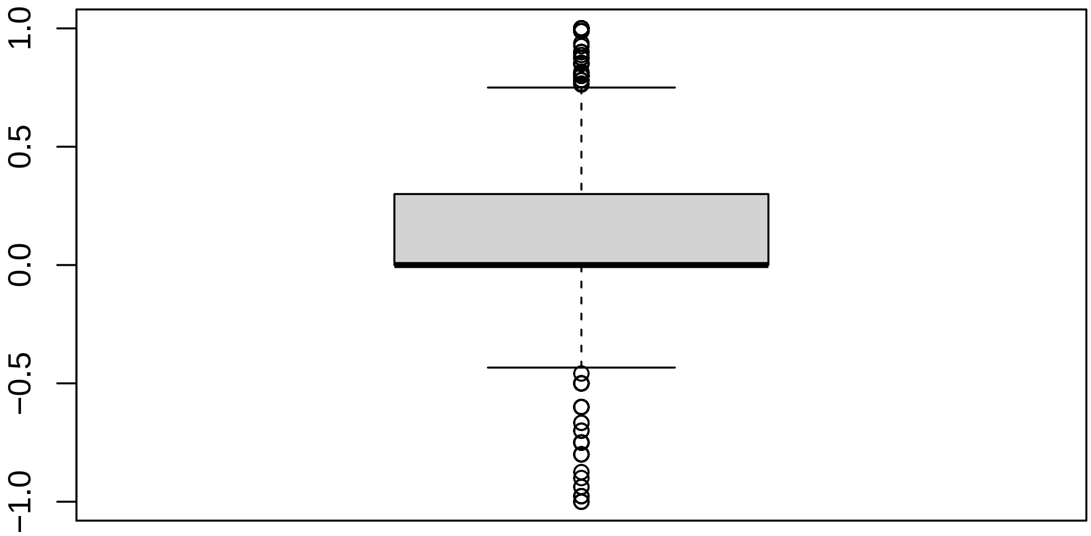
```
## Warning in hist(log10(post_df$Polarity)): NaNs produced
```




```
#Also check boxplots
boxplot(pre_df$Polarity)
```




```r
boxplot(post_df$Polarity)
```




```r
#Let's just try a test real quick to see if my initial hypothesis is correct
#without any manipulation

#no normality so we need to use wilcox
wilcox.test(post_df$Polarity[1:6098],
            pre_df$Polarity[1:6098],
            alternative = "greater",
            paired = TRUE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  post_df$Polarity[1:6098] and pre_df$Polarity[1:6098]
## V = 5566281, p-value = 0.9922
## alternative hypothesis: true location shift is greater than 0
```

```r
# Wilcoxon signed rank test with continuity correction
#
#data:  post_df$Polarity[1:6098] and pre_df$Polarity[1:6098]
#V = 5565868, p-value = 0.9923
#alternative hypothesis: true location shift is greater than 0
```

```r
#Interesting, so my alternative hypothesis is not true and in fact
#might be the opposite...let's check
wilcox.test(post_df$Polarity[1:6098],
            pre_df$Polarity[1:6098],
            alternative = "less",
```

```r
            paired = TRUE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  post_df$Polarity[1:6098] and pre_df$Polarity[1:6098]
## V = 5566281, p-value = 0.007785
## alternative hypothesis: true location shift is less than 0
```

```r
# Wilcoxon signed rank test with continuity correction
#
#data:  post_df$Polarity[1:6098] and pre_df$Polarity[1:6098]
#V = 5566281, p-value = 0.007785
#alternative hypothesis: true location shift is less than 0
```

Data Manipulation

```r
#It looks like a polarity of 0 is skewing the data. I'm not sure if that is
#real data or bad data caused by TextBlob (or some mix of the two).

ffxiv_tweets_df %>%
  group_by(Patch_Timing) %>%
  filter(Polarity != 0) %>%
  summarise(
    n = n()
  )
```

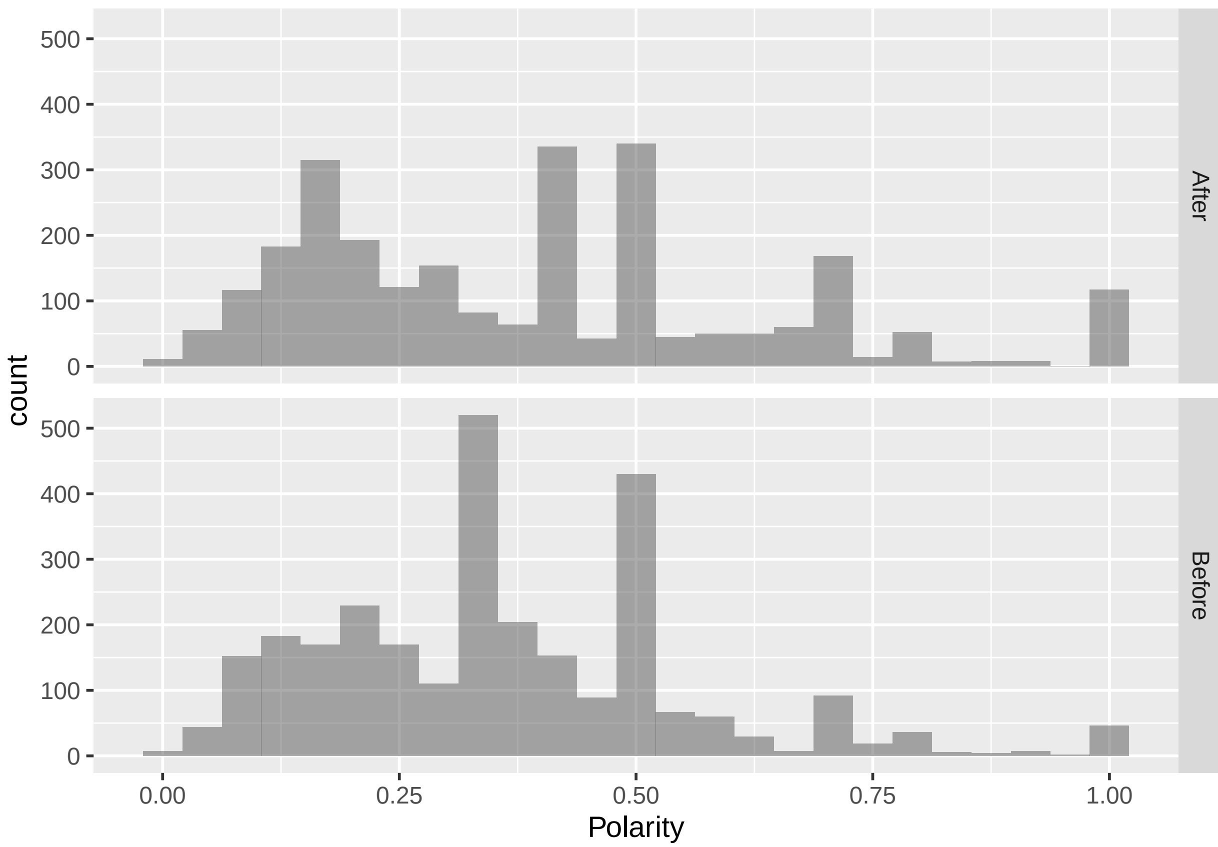
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   Patch_Timing     n
##   <chr>        <int>
## 1 After         3221
## 2 Before        3397
## 3 Downtime      1223
```

```r
#It looks like we have a lot of tweets where there is no polarity. I find
#this odd, so I also want to look at tweets above 0 and below 0 in both
#timeframes
no_downtime <- ffxiv_tweets_df %>%
  filter(Patch_Timing == "Before" | Patch_Timing == "After")

no_downtime_pos <- no_downtime %>%
  filter(Polarity > 0)

no_downtime_neg <- no_downtime %>%
  filter(Polarity < 0)

no_downtime_neg_before <- no_downtime_neg %>%
  filter(Patch_Timing == "Before")

no_downtime_neg_after <- no_downtime_neg %>%
  filter(Patch_Timing == "After")

no_downtime_pos_before <- no_downtime_pos %>%
```

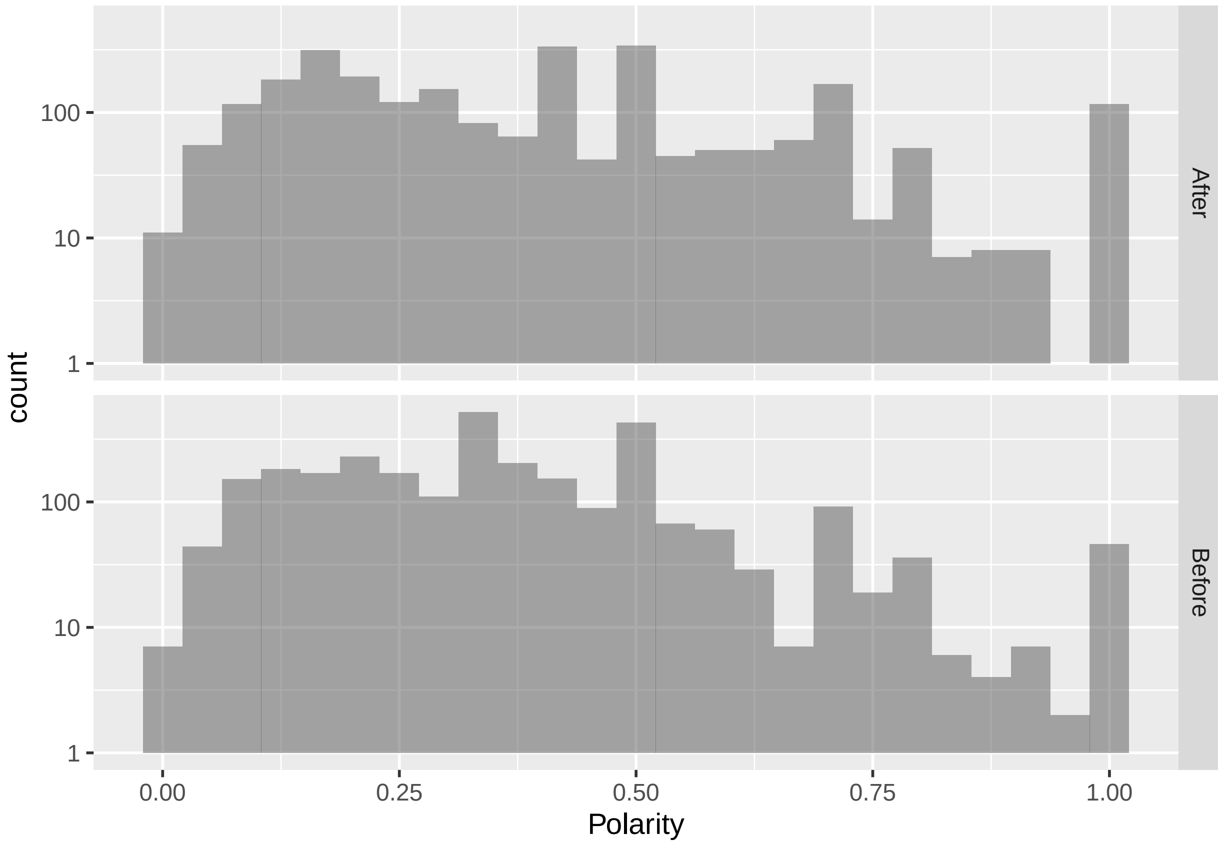
```r
  filter(Patch_Timing == "Before")

no_downtime_pos_after <- no_downtime_pos %>%
  filter(Patch_Timing == "After")
```

```r
#Checking to see if a transformation would help my data
no_downtime_pos %>%
  gf_histogram(~Polarity) %>%
  gf_facet_grid(Patch_Timing ~ .)
```

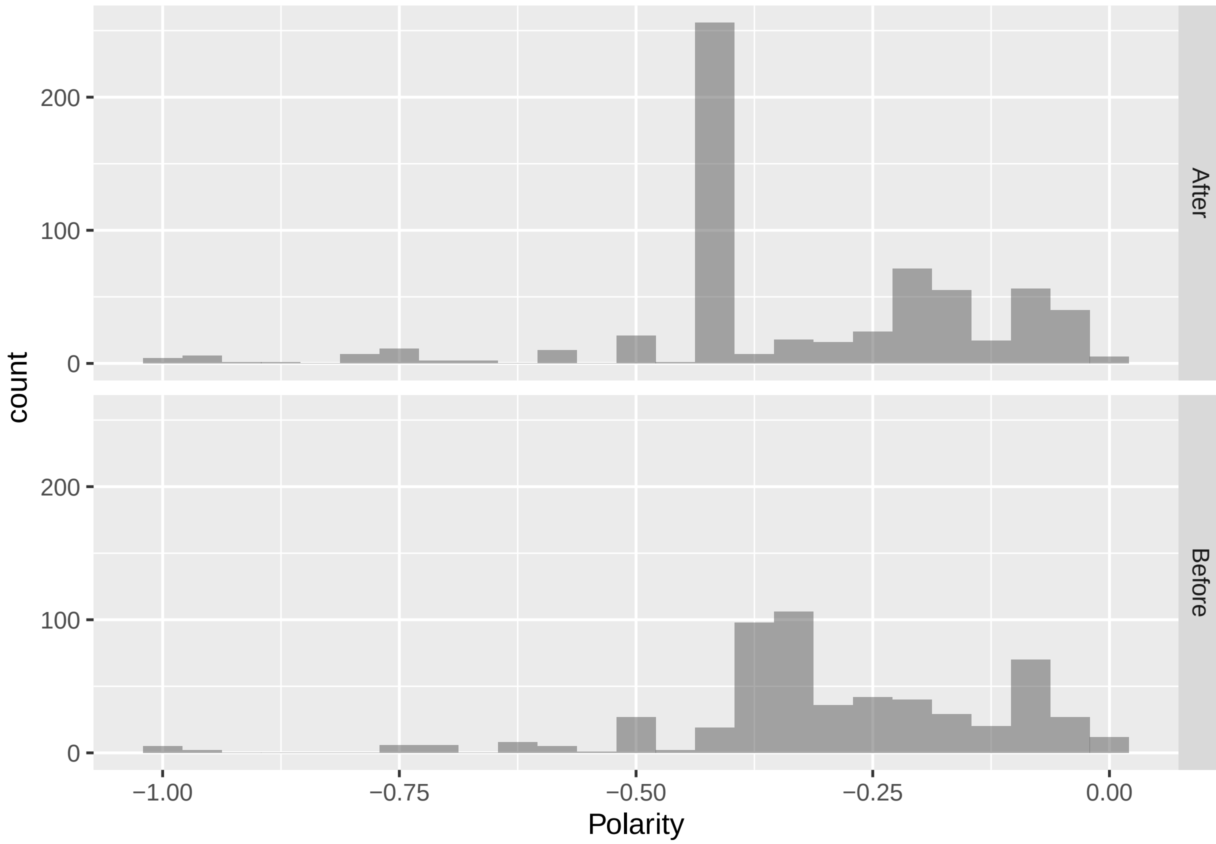



```r
no_downtime_pos %>%
  gf_histogram(~Polarity) %>%
  gf_facet_grid(Patch_Timing ~ .) %>%
  gf_refine(scale_y_log10())
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 rows containing missing values (geom_bar).
```

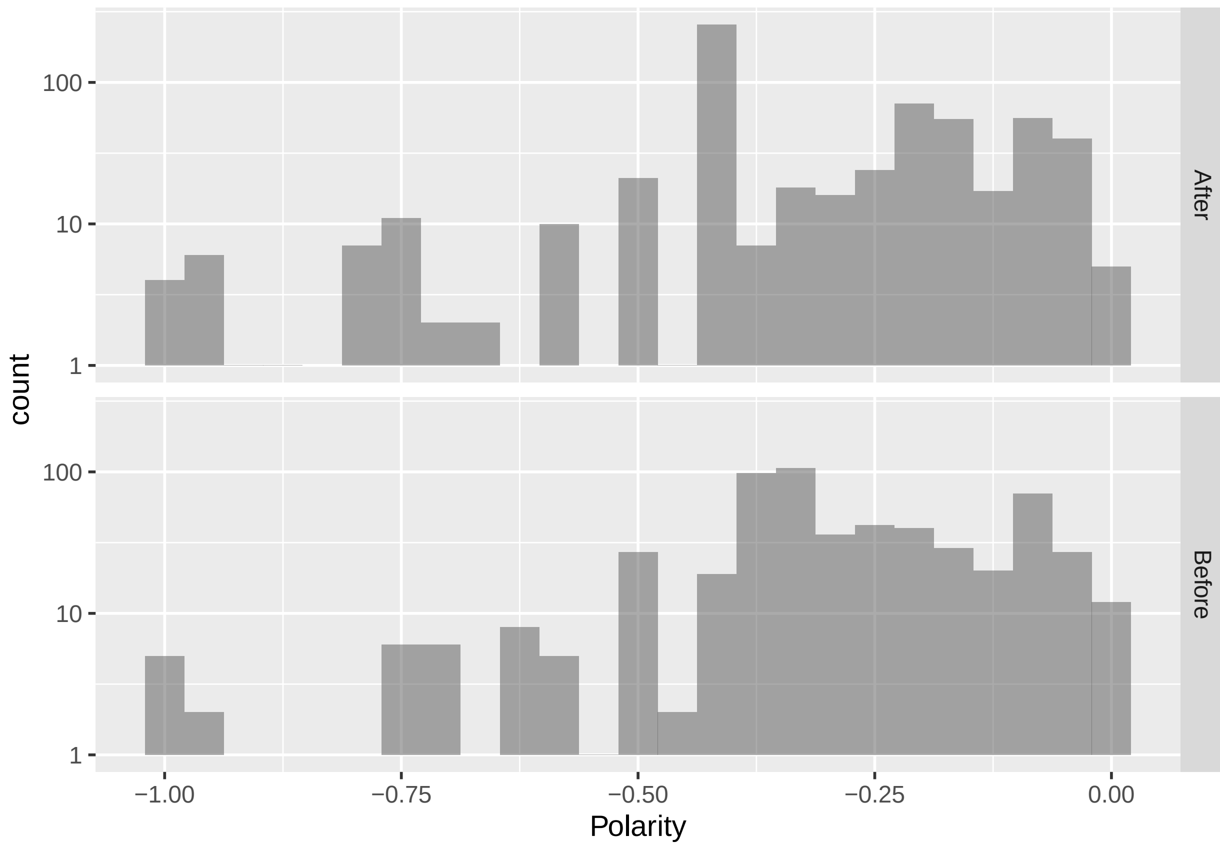



```
no_downtime_neg %>%
  gf_histogram(~Polarity) %>%
  gf_facet_grid(Patch_Timing ~ .)
```




```
no_downtime_neg %>%
  gf_histogram(~Polarity) %>%
  gf_facet_grid(Patch_Timing ~ .) %>%
  gf_refine(scale_y_log10())
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 8 rows containing missing values (geom_bar).
```




```r
#That data doesn't look like it follows a normal distribution either, which
#means we'll use another wilcox test

#x is to the right of y = no
#"Before the patch, negative reviews were closer to 0"
wilcox.test(no_downtime_neg_before$Polarity[1:561],
            no_downtime_neg_after$Polarity[1:561],
            alternative = "greater",
            paired = TRUE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  no_downtime_neg_before$Polarity[1:561] and no_downtime_neg_after$Polarity[1:561]
## V = 82113, p-value = 0.0521
## alternative hypothesis: true location shift is greater than 0
```

```r
##Wilcoxon signed rank test with continuity correction
#
#data:  no_downtime_neg_before$Polarity[1:561] and no_downtime_neg_after$Polarity[1:561]
#V = 82113, p-value = 0.0521
#alternative hypothesis: true location shift is greater than 0
```

```r
# x is to the left of y = yes
#"Before the patch, positive reviews were closer to 0"
wilcox.test(no_downtime_pos_before$Polarity[1:2590],
            no_downtime_pos_after$Polarity[1:2590],
```

```r
            alternative = "less",
            paired = TRUE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  no_downtime_pos_before$Polarity[1:2590] and no_downtime_pos_after$Polarity[1:2590]
## V = 1452972, p-value = 3.27e-05
## alternative hypothesis: true location shift is less than 0
```

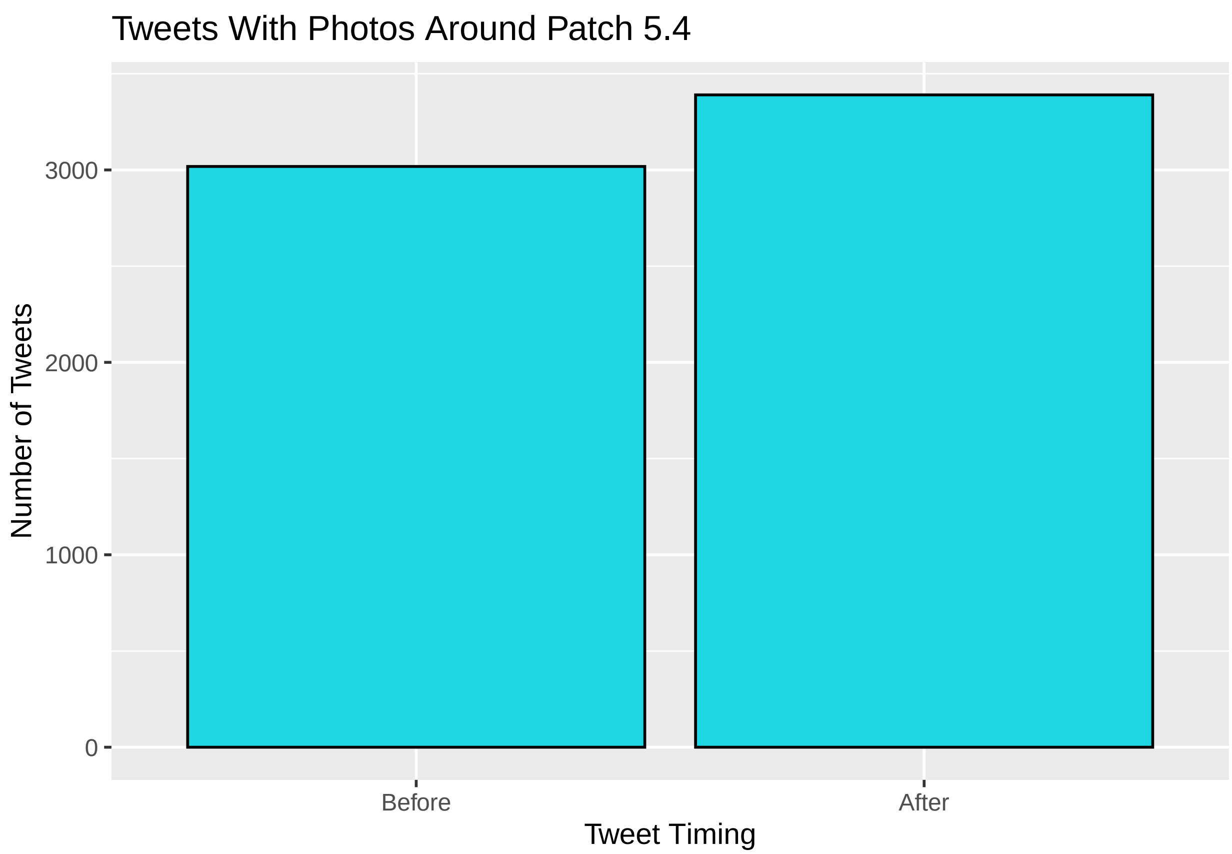
```r
##Wilcoxon signed rank test with continuity correction
#
#data:  no_downtime_pos_before$Polarity[1:2590] and no_downtime_pos_after$Polarity[1:2590]
#V = 1452972, p-value = 3.27e-05
#alternative hypothesis: true location shift is less than 0
```

Looking at some other variables for fun

```r
#Find a count of tweets with Photos in them by group around the patch
media_tweets <- no_downtime %>%
  group_by(Patch_Timing) %>%
  count(Media)
```

Graphs!

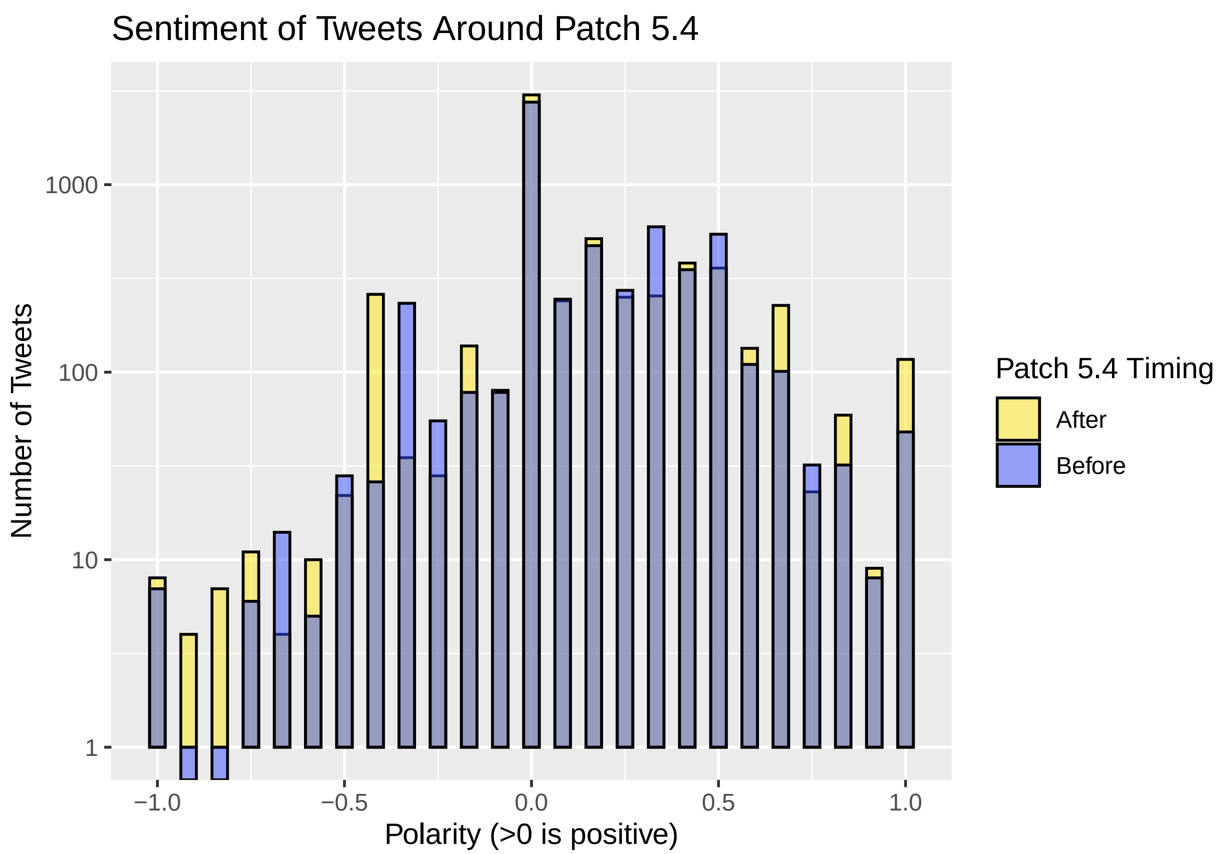
```r
#Graph Tweets with photos in ascending order
media_tweets %>%
  ungroup(Patch_Timing) %>%
  filter(Media == "photo") %>%
  mutate(Patch_Timing = reorder(Patch_Timing,n)) %>%
  gf_col(n~Patch_Timing, color = "black", fill = "#1fd6e3") %>%
  gf_labs(title = "Tweets With Photos Around Patch 5.4",
          y = "Number of Tweets",
          x = "Tweet Timing")
```



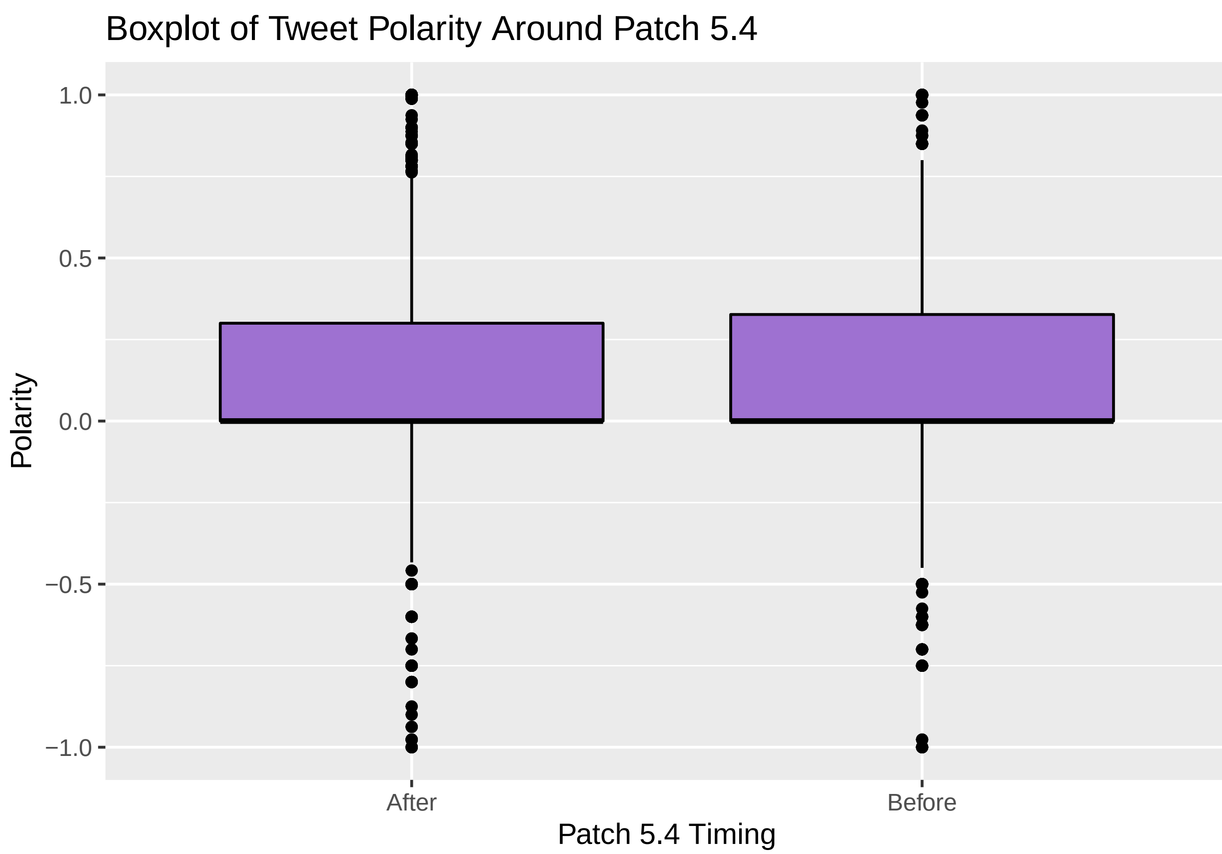

```
#This comparison might have made a little more sense to use percentages rather
#than a strict count, since the tweet count was slightly different, but
#this wasn't really a variable of interest to my question so I have not
#omitted that for this project
```

```
#Graph a spread of polarity
#I use width = 0 to on position_dodge to make comparison simpler,
#and to make the graph less cluttered
no_downtime %>%
  gf_histogram(~Polarity,
               color = "black",
               fill =~ Patch_Timing,
               position=position_dodge(width=0)) %>%
  gf_refine(scale_y_log10(),
            scale_fill_manual(values = c("#ffe818","#384fff"))) %>%
  gf_labs(title = "Sentiment of Tweets Around Patch 5.4",
          fill = "Patch 5.4 Timing",
          y = "Number of Tweets",
          x = "Polarity (>0 is positive)")
```

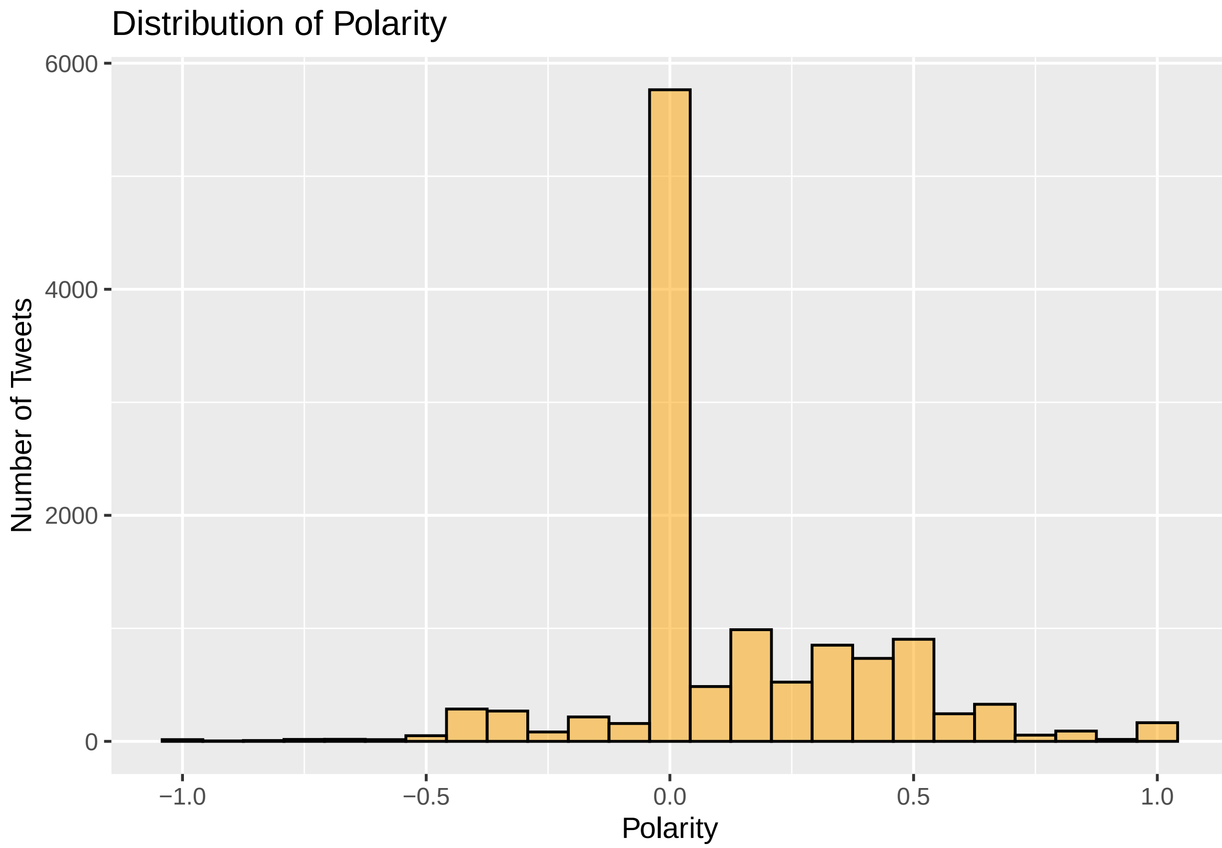
```
## Warning: Transformation introduced infinite values in continuous y-axis
```




```
#Boxplot of polarities
no_downtime %>%
  gf_boxplot(Polarity~Patch_Timing,
             color = "black",
             fill = "#9e71d1") %>%
  gf_labs(title = "Boxplot of Tweet Polarity Around Patch 5.4",
          x= "Patch 5.4 Timing")
```




```
#So many 0's plot
no_downtime %>%
  gf_histogram(~Polarity, color = "black", fill = "orange") %>%
  gf_labs(title = "Distribution of Polarity",
          y = "Number of Tweets")
```


Distribution of Polarity
6000
4000
2000
0
Number of Tweets
−1.0
−0.5
0.0
0.5
1.0
Polarity