SWA Assessment Write-Up

Luke Dickson

2024-10-15

```
## Importing Packages
library(RedditExtractoR)
library(dplyr)
library(stringr)
library(RPostgres)
library(DBI)
library(ggplot2)
library(patchwork)


## Connecting to database
user = "luke"
pw = {
  "V92Gs0AsLosXGCcxXfdL"
  }
gabUser = "gabriel"
gabPw = {
  "CgYkoLFAYsNvStdh"
}

con = DBI::dbConnect(RPostgres::Postgres(), dbname = "redditdata",
                     host = "188.245.90.113", port = 5432,
                     user = gabUser, password = gabPw)

## Remove Passwords
rm(pw)
rm(gabPw)

## Loading Tables
subreddits = dbGetQuery(con,
      paste('SELECT * FROM "US_Election_Subreddits_260924"', sep = ""))

elec_posts_by_sub = dbGetQuery(con,
      paste('SELECT * FROM "US_Election_Posts_By_Subreddits_Year_260924"', sep = ""))

elec_posts_year = dbGetQuery(con,
      paste('SELECT * FROM "US_Election_Posts_Year_260924"', sep = ""))
```

Does Online Activity In Election-Related Sub-Reddits Increase as the Election Approaches?

— Hypothesis Testing ——————————————————————

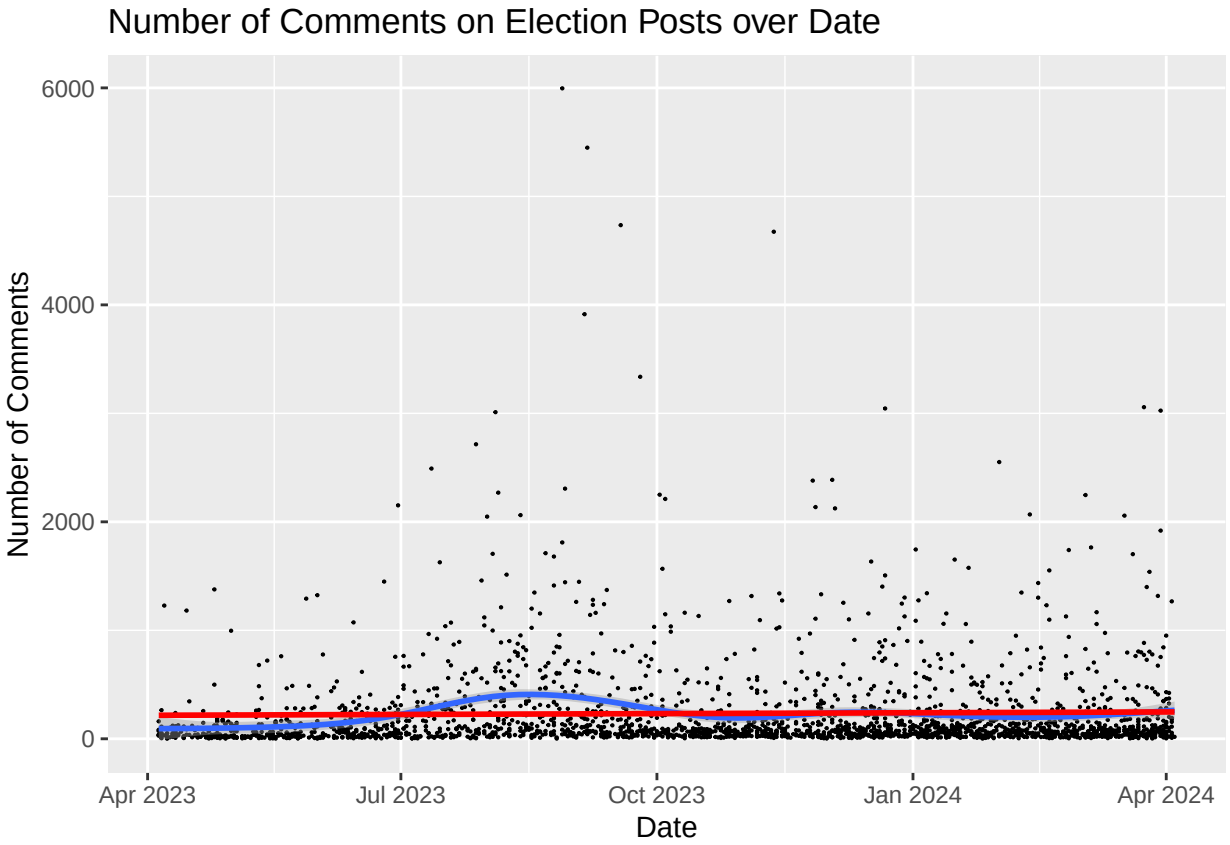
The data that was scraped from Reddit has a vast amount of information that can be used to find many relationships. These relationships include how users, communities, and even believes are linked. However, it can also provide information about user habits. Using this data, the online presence of users can be tracked to find relationships between the significant events and their activity. An example of this is to test and see whether the activity in online communities about the US election increase in the lead up to the election itself, as well as around significant milestones in the build up. To test this, the following Hypotheses will be tested:

H_0 : There is no relationship between the amount online activity in these communities and electoral events

H_a : The presence in election-related online communities will increase with time as the election approaches

Online Presence Plots

```
## Converting date_utc into class Date
elec_posts_by_sub$date_utc = as.Date(elec_posts_by_sub$date_utc)

## Use ggplot to create a plot for comments over date
ggplot(elec_posts_by_sub, aes(x = date_utc, y = comments)) +
  geom_point(size = 0.1) +
  geom_smooth() +
  geom_smooth(method = 'lm', col = "red", se = FALSE)+
  labs(x = "Date", y = "Number of Comments") +
  ggtitle("Number of Comments on Election Posts over Date")
```



The above plot shows the the number of comments on a US election related post, based on the date that the post was created. However, it is quite obvious that the data is very scattered - making it hard to observe any trends. In an attempt to overcome this issue, normalisation and scaling techniques were applied to the data using the following formulae:

Normalisation Formula:

$$\text{Norm(Comments)} = \frac{\text{Comments on Post}}{\text{Subscribers of Subreddit}}$$

Min-Max Scaling Formula:

$$\text{Scaled(X)} = \frac{\text{Norm(Comment)} - \text{min(Normalised Comments)}}{\text{max(Normalised Comments) - min(Normalised Comments)}}$$

The following functions were created to apply each of the formulae:

```
## Function to normalise comments using above formula
normalise = function(table){
  for(row in 1:nrow(table)){
    comments = table$comments[row]
    if(!(table$subreddit[row] %in% subreddits$subreddit)){
      table$normalised[row] = NA
    } else {
      subreddit = which(subreddits$subreddit == table$subreddit[row])
      subscribers = subreddits$subscribers[subreddit]
      table$normalised[row] = comments/subscribers
    }
```

```
  }
  return(table)
}

## Apply Normalise Function to elec_posts_by_sub
elec_posts_by_sub = normalise(elec_posts_by_sub)
elec_posts_by_sub[1:5,c(1,8)]
```

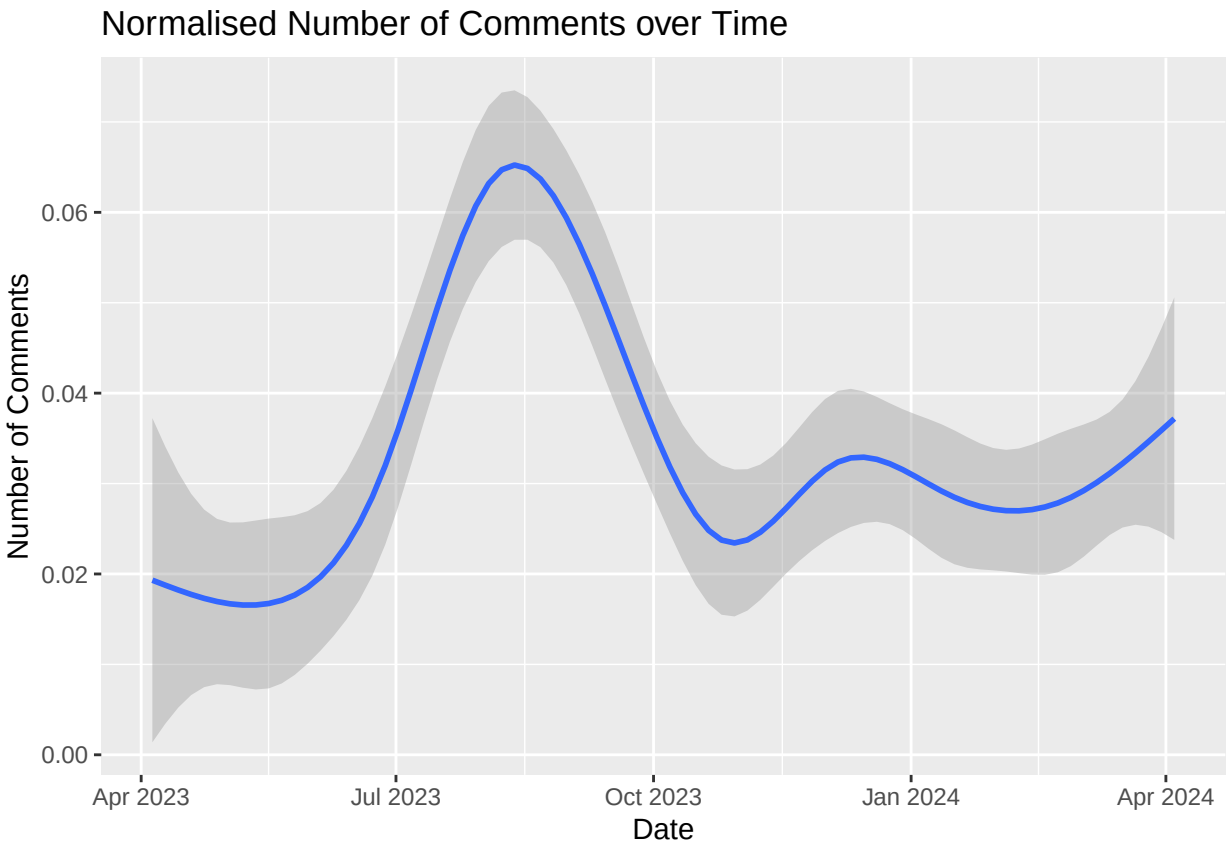
```
##     date_utc normalised
## 1 2023-08-28 0.02575181
## 2 2023-09-06 0.02340254
## 3 2023-09-18 0.02033603
## 4 2023-11-12 0.02007404
## 5 2023-09-05 0.01680997
```

```
## Function to scale comments using above formula
scale = function(table){
  min_norm = min(table$normalised)
  max_norm = max(table$normalised)
  for(row in 1:nrow(table)){
    norm = table$normalised[row]
    table$scaled_comments[row] =
        (norm - min_norm)/(max_norm - min_norm)
  }
  return(table)
}
## Apply scale Function to elec_posts_by_sub
elec_posts_by_sub = scale(elec_posts_by_sub)
elec_posts_by_sub[1:5,c(1,9)]
```

```
##     date_utc scaled_comments
## 1 2023-08-28       1.0000000
## 2 2023-09-06       0.9087530
## 3 2023-09-18       0.7896482
## 4 2023-11-12       0.7794726
## 5 2023-09-05       0.6526944
```

After normalising and scaling the number of comments, the trends in the data become much more apparent, as shown in the following plot:

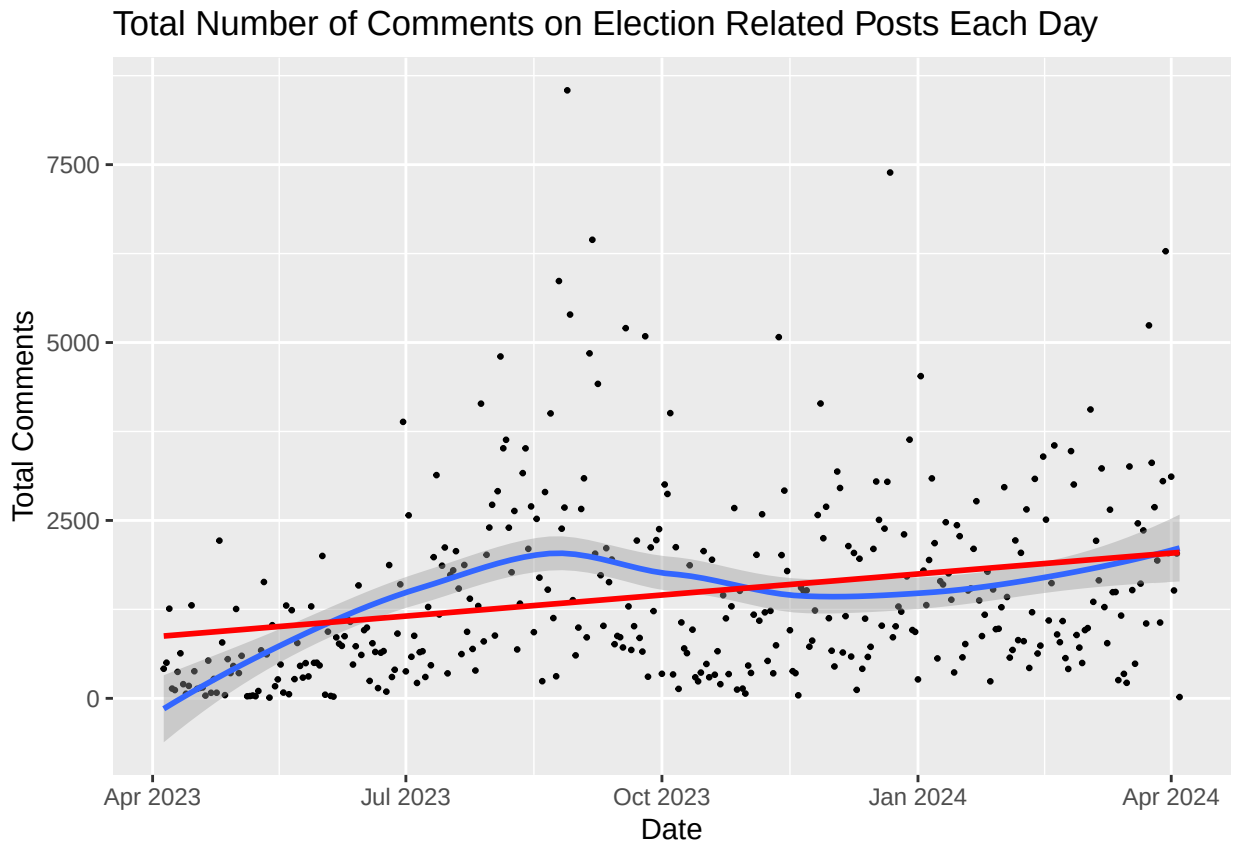
```
ggplot(elec_posts_by_sub, aes(x = date_utc, y = scaled_comments)) +
  labs(x = "Date", y = "Number of Comments") +
  geom_smooth() +
  ggtitle("Normalised Number of Comments over Time")
```



In the plot above, there is a gradual increase of comments over time, with a clear peak in comments around September 2023. After conducting background research on the timing of events, it was found that this peak aligns with the period where the convictions against Donald Trump were announced. This event is very likely to have contributed to a very significant increase in online presence in communities/subreddits about Donald Trump, hence creating the peak. Otherwise, as time progress and the election approaches, the number of comments in these communities (and therefore the activity/online presence) increases gradually.

Another way to measure the online presence is to count the comments per day on election related posts, rather than the comments individual posts. By summing the total comments on each date, the following plot can be created:

```
## Sum the total comments per date
elec_posts_summary = elec_posts_by_sub %>% group_by(date_utc) %>%
  summarise(comment_count = sum(comments, na.rm = TRUE))

## Plot the above sum over the date
ggplot(elec_posts_summary, aes(x = date_utc, y = comment_count)) +
  labs(x = "Date", y = "Total Comments") +
  geom_point(size = 0.5)+
  geom_smooth() +
  geom_smooth(method = 'lm', col = "red", se = FALSE) +
  ggtitle("Total Number of Comments on Election Related Posts Each Day")
```



This plot has a similar shape to the above plot with scaled counts of comments on individual posts. It has the same peak around the announcement of Trump's convictions, as well as a similar gradual increase over time as the election approaches. By applying a linear model to this plot, there is a noticeable increase in gradient - implying there is a positive linear relationship between the two variables.

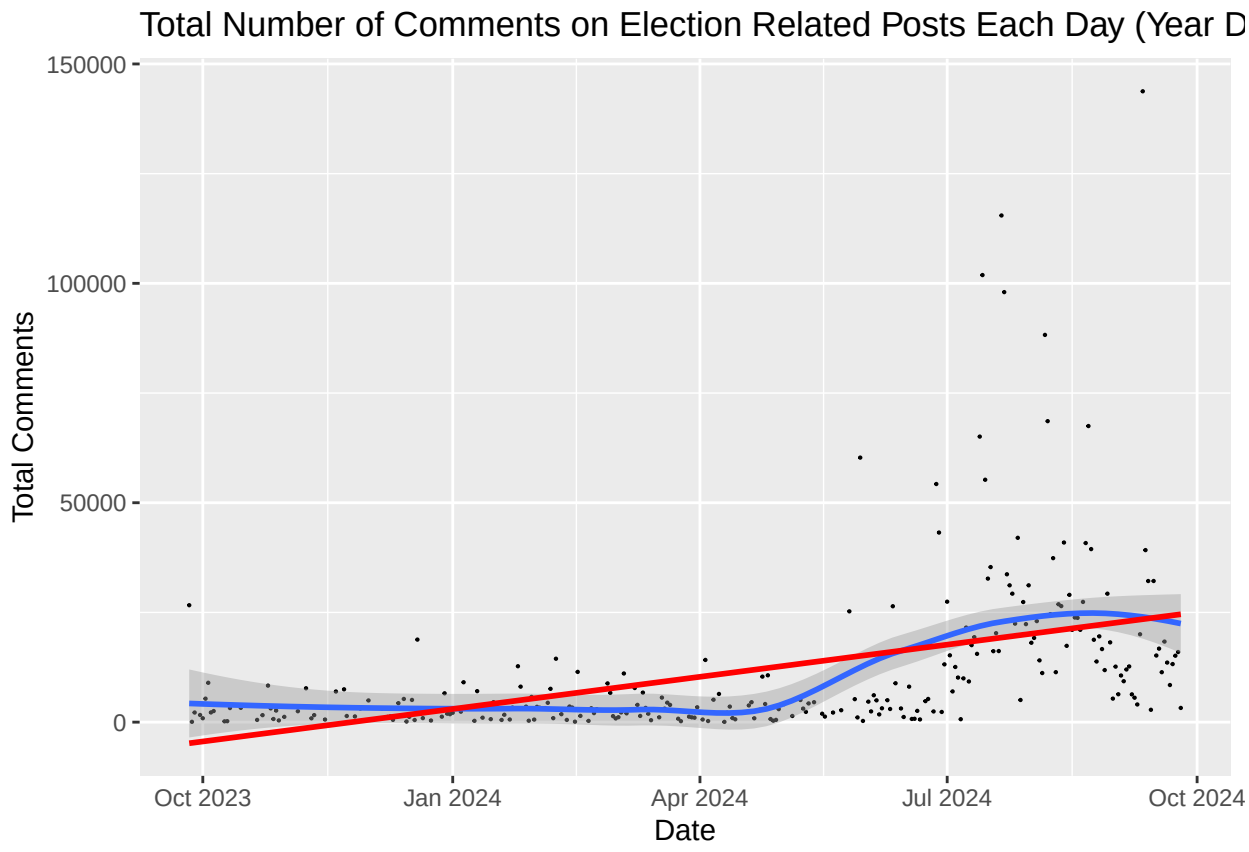
In order to see if this relationship is continued, a new dataset is introduced - with information about posts from a year long period, starting in October 2023. This gives us more of an idea of how the activity changes closer to the election, however cannot be normalised/scaled as there is no information about the subreddit which these posts were from. Below is a plot of the total number of comments on each day using this new dataset:

```
## Sum the total comments per date
elec_posts_year_summary = elec_posts_year %>% group_by(date_utc) %>%
  summarise(comment_count = sum(comments, na.rm = TRUE))

elec_posts_year_summary$comment_count[1:10]
```

```
##  [1] 26647    71  2191  1625   885  5353  8964  2211  2524   181
```

```
elec_posts_year_summary$date_utc = as.Date(elec_posts_year_summary$date_utc)

## Plot the above sum over the date
ggplot(elec_posts_year_summary, aes(x = date_utc, y = comment_count)) +
  labs(x = "Date", y = "Total Comments") +
  geom_point(size = 0.1) +
  geom_smooth() +
  geom_smooth(method = 'lm', col = "red", se = FALSE) +
  ggtitle("Total Number of Comments on Election Related Posts Each Day (Year Data)")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



This plot shows a steady incline in the number of comments per day from around May 2024 - however appears to begin to drop off in the later weeks of the dataset. There also appears to be many more days that have a large number of comments, that stray greatly from the trend line, once that steady incline begins.

Online Presence Stats

While the plots that were created above show that there may be some relationship between the date of posts and the number of comments they receive, there is not enough evidence to confirm it. However, statistical analysis can help provide more evidence to either confirm or deny the null hypothesis H_0.

```
## Count the total number of comments per day as comment_count
elec_posts_summary = elec_posts_by_sub %>% group_by(date_utc) %>%
  summarise(comment_count = sum(comments, na.rm = TRUE))

## Create a linear model of daily comment count and date
comment_count_model = lm(comment_count ~ date_utc, data = elec_posts_summary)
summary(comment_count_model)
```

```
##
## Call:
## lm(formula = comment_count ~ date_utc, data = elec_posts_summary)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2032.3  -839.8  -377.0   572.2  7201.5
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.171e+04  1.221e+04  -5.052 6.93e-07 ***
## date_utc     3.217e+00  6.221e-01   5.172 3.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1254 on 363 degrees of freedom
## Multiple R-squared:  0.06864,    Adjusted R-squared:  0.06607
## F-statistic: 26.75 on 1 and 363 DF,  p-value: 3.834e-07
```

This linear model using the daily total count of comments (rather than the number of comments per post) returns a much lower p-value of '3.834×10^{-7} - indicating that the relationship is much stronger. This implies that, while the number of comments on individual posts does not necessarily increase as the election approaches, the total number of people commenting on election related posts increases.

In order to confirm this relationship, the same modelling method was applied to the second dataset:

```
## Count the total number of comments per day as comment_count
elec_posts_year_summary = elec_posts_year %>% group_by(date_utc) %>%
  summarise(comment_count = sum(comments, na.rm = TRUE))

elec_posts_year_summary$date_utc = as.Date(elec_posts_year_summary$date_utc)

## Create a linear model of daily comment count and date
comment_count_year_model = lm(comment_count ~ date_utc, data = elec_posts_year_summary)
summary(comment_count_year_model)
```

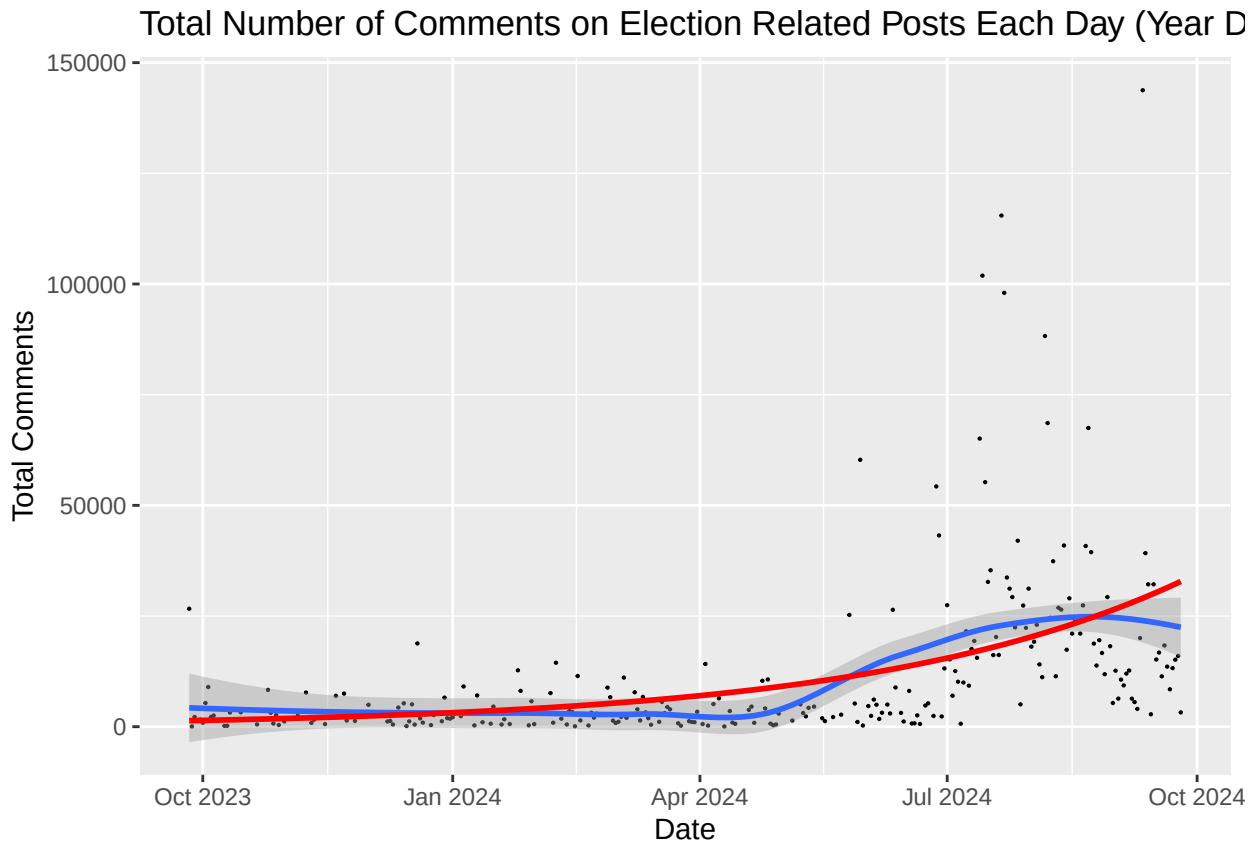
```
##
## Call:
## lm(formula = comment_count ~ date_utc, data = elec_posts_year_summary)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21348  -9003  -2889   3330 120316
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.585e+06  1.925e+05  -8.237 8.08e-15 ***
## date_utc     8.054e+01  9.707e+00   8.297 5.39e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16660 on 266 degrees of freedom
## Multiple R-squared:  0.2056, Adjusted R-squared:  0.2026
## F-statistic: 68.84 on 1 and 266 DF,  p-value: 5.39e-15
```

The summary of this linear model shows the statistics and can help determine the significance of it. A p-value of 5.39×10^{-15} is an extremely low p-value - thus indicating there is a strong linear relationship between these two variables. The calculated gradient has a value of 8.054 which is also significantly high - meaning that this model suggests that the date has a very strong effect on the number of comments per day on Reddit.

After further research, it was found that a Poisson Regression model is a better fit for counts of data, rather than a linear model, so the models were refitted as Poisson regression models below:

```r
## Plot the poisson model for year data
ggplot(elec_posts_year_summary, aes(x = date_utc, y = comment_count)) +
  labs(x = "Date", y = "Total Comments") +
  geom_point(size = 0.1) +
  geom_smooth() +
  geom_smooth(method = 'glm', method.args = list(family = "poisson"), col = "red", se = FALSE) +
  ggtitle("Total Number of Comments on Election Related Posts Each Day (Year Data)")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```r
## Fitting Subs dataset to poisson model
poisson_subs = glm(comment_count ~ date_utc, family = "poisson", data = elec_posts_summary)
summary(poisson_subs)
```

```
##
## Call:
## glm(formula = comment_count ~ date_utc, family = "poisson", data = elec_posts_summary)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -63.74  -26.97  -10.54   13.79  132.58
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.636e+01  2.592e-01  -140.3   <2e-16 ***
## date_utc     2.222e-03  1.318e-05   168.5   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 370894  on 364  degrees of freedom
## Residual deviance: 342017  on 363  degrees of freedom
## AIC: 345183
##
## Number of Fisher Scoring iterations: 5
```

After fitting a Poisson model to the dataset, the p-value is not nearly as low as it previously was, however is still low enough to be very significant. Returning a value of 2.222×10^{-3}, this p-value indicates that the model strongly fits the dataset. When adding the model onto the plot, it can be seen that there is a significant rise in the gradient beginning around July 2024, and increasing until the end of the plot - indicating it would continue to rise as time progresses.

```
## Fitting year dataset to poisson model
poisson_year = glm(comment_count ~ date_utc, family = "poisson", data = elec_posts_year_summary)
summary(poisson_year)
```

```
##
## Call:
## glm(formula = comment_count ~ date_utc, family = "poisson", data = elec_posts_year_summary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -210.13   -78.51   -35.50    19.06   480.10
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.638e+02  1.455e-01   -1126   <2e-16 ***
## date_utc     8.715e-03  7.307e-06    1193   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4870680  on 267  degrees of freedom
## Residual deviance: 3010925  on 266  degrees of freedom
## AIC: 3013677
##
## Number of Fisher Scoring iterations: 5
```

Likewise with the above model, this model also has a decrease in the p-value, but is still small enough to be very significant. A returned value of 8.175×10^{-3} also implies that the model is strong fit for the dataset and indicates that there is a strong relationship present between the two variables. To further test this, the Mean Squared Error (MSE) of both models are computed below:

```
## Predictor returns log values
predicted_log_values = predict(poisson_year, elec_posts_year_summary)
##Get actual predicted values
predicted_values = exp(predicted_log_values)

actual_values = elec_posts_year_summary$comment_count
```

```r
## Calculate MSE
year_mse = mean((actual_values - predicted_values)^2)


## Predictor returns log values
predicted_log_values = predict(poisson_year, elec_posts_summary)
##Get actual predicted values
predicted_values = exp(predicted_log_values)

actual_values = elec_posts_summary$comment_count

## Calculate MSE
subs_mse = mean((actual_values - predicted_values)^2)

cat("Subs Dataset MSE = ", subs_mse, "\nYears Dataset MSE = ", year_mse)
```

```
## Subs Dataset MSE =  4707943
## Years Dataset MSE =  268900450
```

The result of the MSE returned extremely high results for both models. This is an indicator that there is a high level of variance in the models, and that the variables do not necessarily account for all of randomness/variation in the data. For example, this result means that any specific day will have more comments than the previous day just because it is closer to the election date. Therefore, this shows evidence that there are some other factors involved in the relationship.

— Limitations & Conclusions ————————————————————

Limitations

While quite successful in terms of results, this project did have a few limitations. Due to the pressure of having to complete the project before the due date, time was limited and prevented the analysis from being more in-depth than what it was. Given more time, more tests may have been conducted to gather stronger evidence or find other conclusions. Another limitation was the inability to access certain data. If there was an accessible API for software such as Reddit or X, a much larger amount of data and information would have been readily available and may have contributed to other branches of analysis (Mastodon was considered for its open-source API, however there was too little activity to conduct an analysis such as this).

Conclusions:

From the testing that was conducted on this data, there is evidence that suggests a relationship that is present between the tested variables and, therefore, reject the Null Hypothesis. However, there is a certain factor of randomness to it that contributes to a high level of variance. These factors could be many things, including worldwide events (political or non-political), or even external factors, such as certain times when online presence is generally higher. Factors such as these acting on the data is present in our dataset - with a peak present at the time of his convictions. From this project, future testing that may be conduced may include a comparison to general online presence to see if there is an increase in all online presence, rather than just those in the electoral communities. Furthermore, more in-depth testing could be performed on these posts - such as frequency of posts per day, posts per subscriber in each subreddit, how the percentage of inactive subscribers changes over time in electoral subreddits, etc. Testing topics such as these can lead to a deeper analysis of this data and can help solidify these findings/confirm these relationships, or discover completely new ones.