

MSD 2019 Final Project

A replication and extension of Chilling Effects: Online Surveillance and Wikipedia Use by
Jonathon W. Penney, Berkeley Technology Law Journal

Thanaspakorn Niyomkarn, Alex Li Kong, and Sang Won Lee (tn2381, alk2225, and sl4447)

2019-05-12 15:11:13

Contents

1. Introduction	1
2. Reproducing the Original Study	1
2.1 Data	2
2.2 Methodology	2
2.3 Criticism	3
2.3.1 Defining and removing outliers	3
2.3.2 Privacy ratings: data collection and calculation	3
2.3.3 Result interpretation	4
2.4 Replication Results	4
2.4.1 Total views of terrorism-related keywords before and after the incident	4
2.4.2 Linear model with interactions: Analysis and Plots	5
3. Extended Analysis	9
3.1 Longer Trend Analysis	9
3.2 Per-keyword Analysis	21
3.2.1 Visualizing sample keywords	21
3.2.2 Quantifying difference between models	25
3.3 Time-series Analysis	27
4. Summary	27

1. Introduction

This Rmd file attempts to replicate and extend the results in Chilling Effects: Online Surveillance and Wikipedia Use by Jonathon W. Penney in Berkeley Technology Law Journal. The author is a research fellow at University of Toronto. This single author paper has H5-index of 21. This paper is about the NSA/PRISM surveillance 2007, where United States National Security Agency (NSA) started collecting Internet communications from various US Internet companies. This information was made public in 2013 by Edward Snowden revelations. This paper deals with the NSA paranoia where the paper studies traffic to Wikipedia articles on topics that raise privacy concerns for Wikipedia users before and after the Edward Snowden revelations. The Wikipedia traffic was chosen because over 50% of Internet users use Wikipedia as a source of information. Over 1/3 of Americans annually access Wikipedia as a source of information and is in top 10 of most popular sites on the internet.

2. Reproducing the Original Study

Our group decide to reproduce the main analysis of the study which is the study about the discontinuity of the trend of views on Wikipedia articles. Although the data is time series, linear regression is a good way to

Table 8: Topic Keyword—48 Article Group

Topic Keyword	Wikipedia Articles	Govern	Browser	Privacy	Avoid-
		-ment Trouble	Delete	Sensi- tive	ance
Al Qaeda	http://en.wikipedia.org/wiki/Al-Qaeda	2.20	2.11	2.21	2.84
Terrorism	http://en.wikipedia.org/wiki/terrorism	2.19	2.05	2.16	2.79
Terror	http://en.wikipedia.org/wiki/terror	1.98	1.96	2.01	2.64
Attack	http://en.wikipedia.org/wiki/attack	1.92	1.91	1.92	2.56

Figure 1: Sample keywords and their privacy score

capture the trend since our main goal is not predicting the exact number of views. The linear model used in the study is:

$$Y_t = \beta_0 + \beta_1 time + \beta_2 intervention + \beta_3 postslope$$

The model can be interpret as an ordinary regression for data at the time before the revelation of the surveillance in June 2013. For the data after the incident, an interaction is added to both intercept and slope, 0 if data is before and 1 for after. ‘Intervention’ or change in level is the binary value multiply with a weight which indicates the changing intercept after the event. ‘Postslope’ or change in slope is the binary value multiply with time indicating change in trend.

2.1 Data

The study uses a list of keywords the U.S. Department of Homeland Security uses to track and monitor social media. Keyword selection and ranking are done using a survey on Amazon’s Mechanical Turk (MTurk) asking their opinions about topics on ‘Government trouble’, ‘Browser delete’, ‘Privacy sensitive’, ‘Avoidance’. Then all the scores are averaged to a single value called ‘Combined privacy rating’.

The paper uses data from stats.grok.se which has stopped being updated as of January 2016 and the server is down at the moment. Our group chose to use an alternative data source from Wikipediatrend package in R (<https://github.com/petermeissner/wikipediatrend>) which allows user to specify page names, languages, start and end date of data and the library will return daily views for the articles.

2.2 Methodology

The analysis will be done of different set of keywords such as terrorism-related article which is expected to change and popular articles which is used as a baseline. The author concludes that there exists a change in trend if coefficients of the interaction terms are significant. Here is the example of regression analysis for terrorism-related articles.

The sample plots will be shown in the Replication Results section to compare with our results.

Table 2: Second Results, 47 Terrorism-related Articles (Hamas Excluded)

Independent Variable	Coefficients	Standard Error	P-value
Coefficient (β_0) Expected Total Views at Beginning of Study	2289153**	109751.5	0.000
Secular trend in data (β_1) Change in Views (Monthly) Before 6/2013	41420.51**	10710.65	0.001
Change in level (β_2) Change in Views Immediately After 6/2013	-693616.9 **	154640.9	0.000
Change in slope (β_3) Change in Views (Monthly) After 6/2013	-67513.1**	16789.25	0.000

* $p < 0.05$, ** $p < 0.01$

Figure 2: Original regression summary for terrorism-related keywords

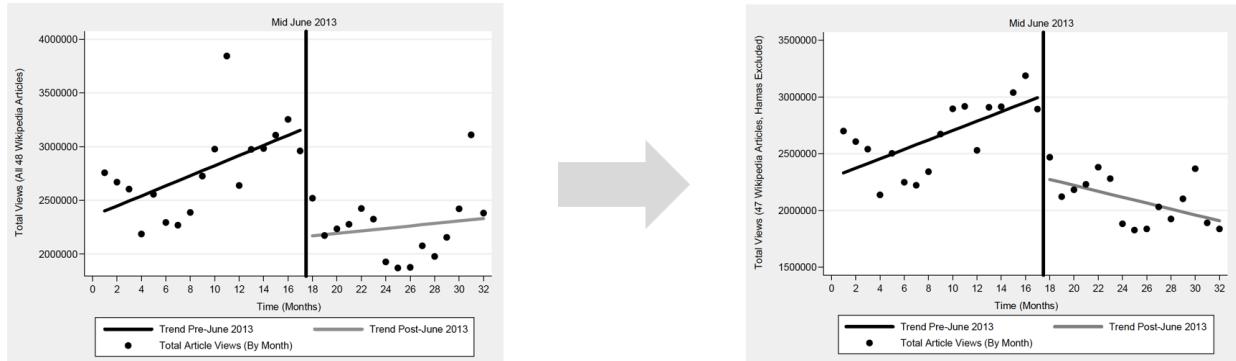


Figure 3: Difference in regression results before and after removing outliers

2.3 Criticism

2.3.1 Defining and removing outliers

Outliers are treated before performing further analysis in this study. There are two mains type of data that are considered to be outliers. The first reason is unusual events, for example the media coverage about dispute between Hamas and Israel. The other method is removing outliers by considering z-score. Both are reasonable way to deal with outliers. However, removing out might cause another problems such as missing data. Moreover, there is no clear rule to identify the events like news and other exposures for all the keywords.

2.3.2 Privacy ratings: data collection and calculation

The first problem about data collection is the representation of the population (MTurk, Wikipedia, US Internet users). This issue is acknowledged by the author that the sample of people who did the survey have slightly lower incomes, slightly more male than female and use “websites and other online resources” for information more generally than the overall US population. The use of multiple proxies: Wikipedia as a

representation of the internet usage and the opinion of MTurk users as public opinions might make the the result prone to more error.

The second problem about data is how the topics are presented to the subjects. There exists some neutral keywords like ‘recruitment’ and ‘terror’ which their contents in Wikipedia does not related with terrorism. The opinion or privacy rating will be less credible if only the keywords are presented to the subjects not the actual webpage.

Finally, the combined privacy rating is calculate by averaging ‘Government trouble’, ‘Browser delete’, ‘Privacy sensitive’, ‘Avoidance’ assumes that all the factors are equally importance which is difficult to prove the validity.

2.3.3 Result interpretation

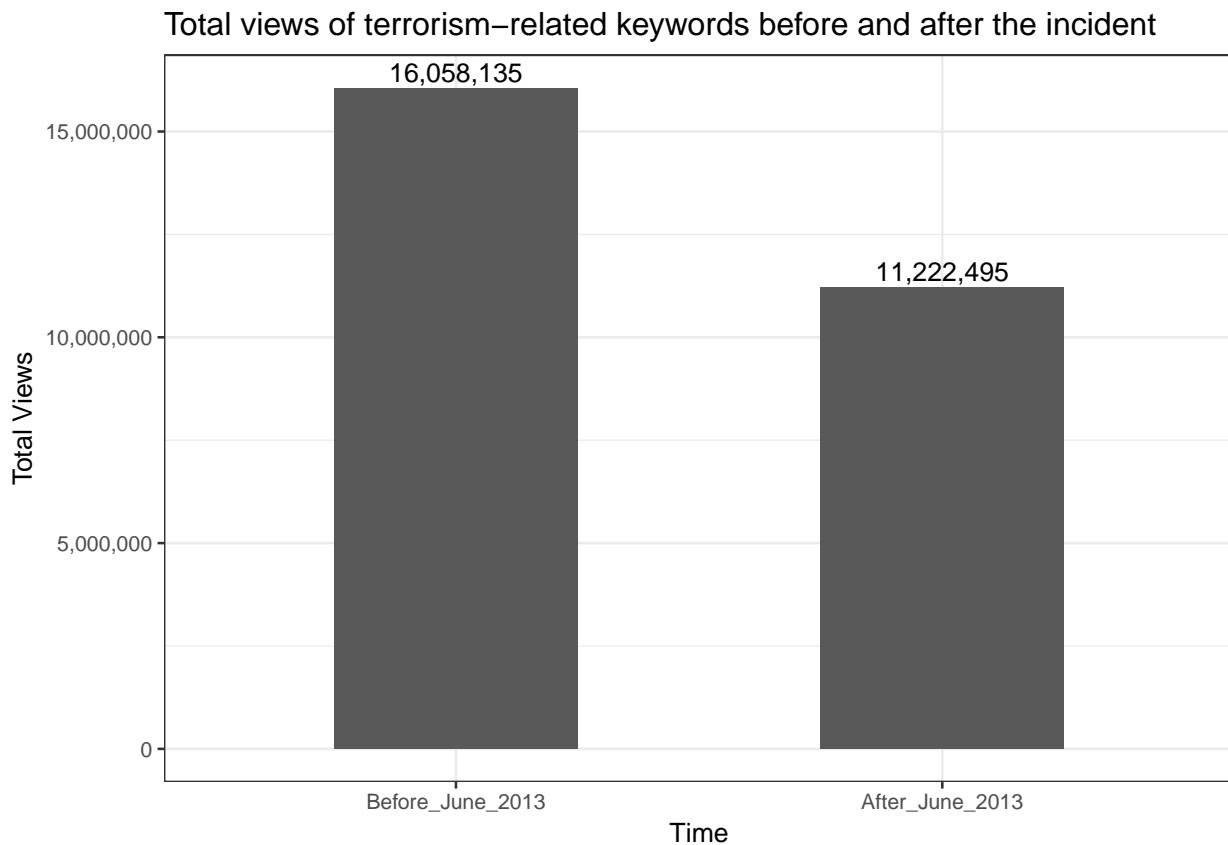
The interpretation of results focus on the significance of coefficients, however it does not provide the sense of magnitude or direction of the change. For example a trend might change from increasing to highly increasing or decreasing and both would give significant results. Moreover, the overall trend might be dominated by few keywords. Our group comes up with the solution to this problem and performs an alternative analysis in the Per-keyword Analysis section.

2.4 Replication Results

```
load("data/terrorism_data.RData")
load("data/infra_data.RData")
load("data/popular_data.RData")
load("data/terrorism_data_long.RData")
```

2.4.1 Total views of terrorism-related keywords before and after the incident

```
terrorism_data %>%
  mutate(before_after = ifelse(date < '2013-06-01', "Before_June_2013", "After_June_2013")) %>%
  group_by(before_after) %>%
  summarise(total_views = sum(views)) %>%
  ggplot(aes(x= factor(before_after, level = c("Before_June_2013", "After_June_2013")), y=total_views,
  scale_y_continuous(name="Total Views", labels = comma) +
  xlab("Time") +
  geom_text(aes(label=comma(total_views)), vjust=-0.3, color="black", size=3.5) +
  theme_bw(base_size = 10) +
  geom_bar(stat="identity") +
  ggtitle("Total views of terrorism-related keywords before and after the incident")
```



2.4.2 Linear model with interactions: Analysis and Plots

```

lm_plot_topic <- function(input_df, gg_title){

df <- data.frame(input_df)
df <- df %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize/views= sum/views)
df$surveillance <- 'before'
df$surveillance[df$month >= '2013-06-01'] <- 'after'

model <- lm(vIEWS ~ month + surveillance + month*surveillance, data = df)
print(summary(model))

df$prediction <- predict(model, df)
df$se <- predict(model, df,
                  se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
df$LoCI <- df$prediction - z.val * df$se
df$HiCI <- df$prediction + z.val * df$se

df$month <- ymd(df$month)

ggplot(df,

```

A. Terrorism Articles Study Group vs. Domestic Security Comparator Group

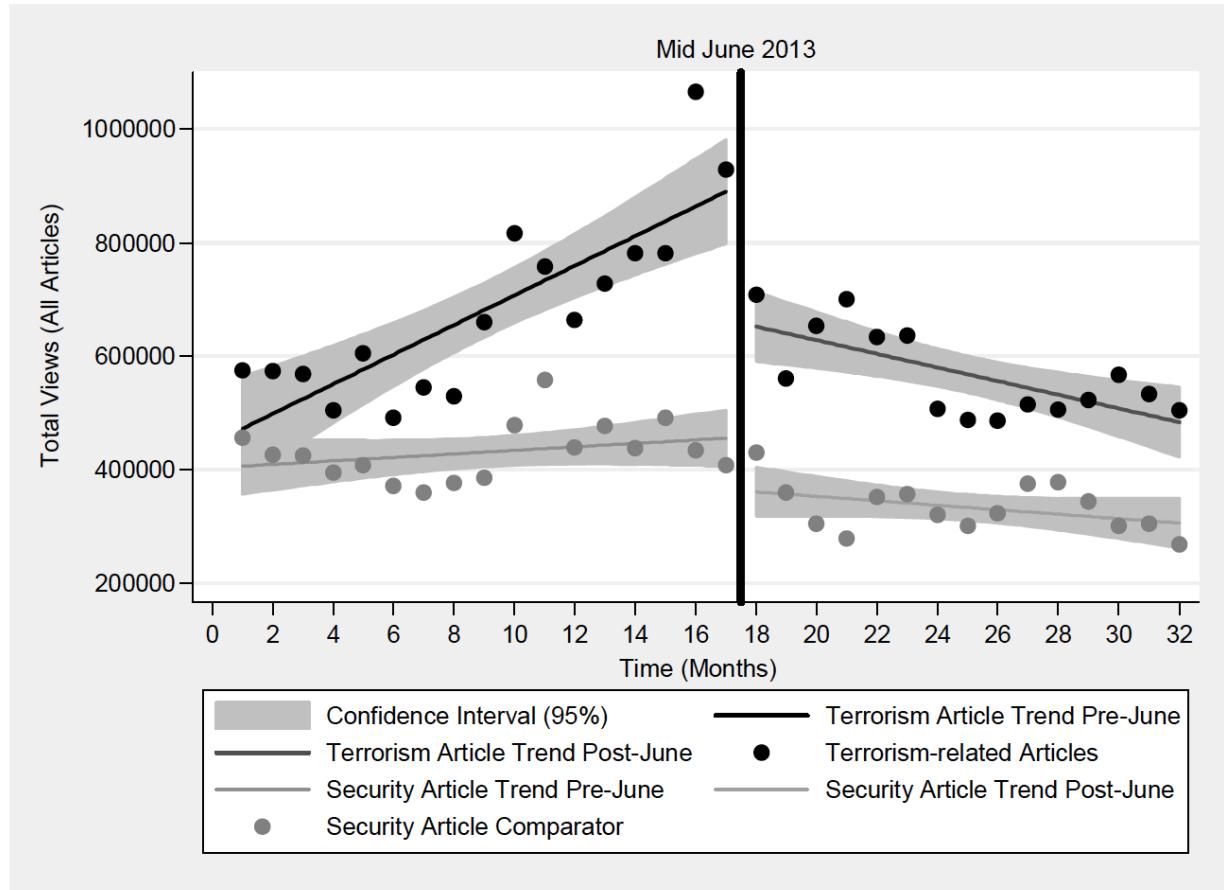


Figure 4: Original result for terrorism-related keywords

```

aes(x = month,
    y = prediction)) +
geom_smooth(aes(ymin = LoCI,
                 ymax = HiCI,
                 color = surveillance),
            stat = "identity") +
geom_point(data = df, aes(x=month, y = views)) +
geom_vline(xintercept = as.Date('2013-06-01'), linetype = 2, colour = 'blue') +
ylab('Views') +
xlab('Time (monthly)') +
scale_x_date(date_breaks = "6 month", labels = date_format("%Y-%b")) +
scale_y_continuous(labels = comma) +
ggtitle(gg_title)
}

lm_plot_topic(terrorism_data, 'Terrorism-related keywords trend before and after June 2013')

##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,

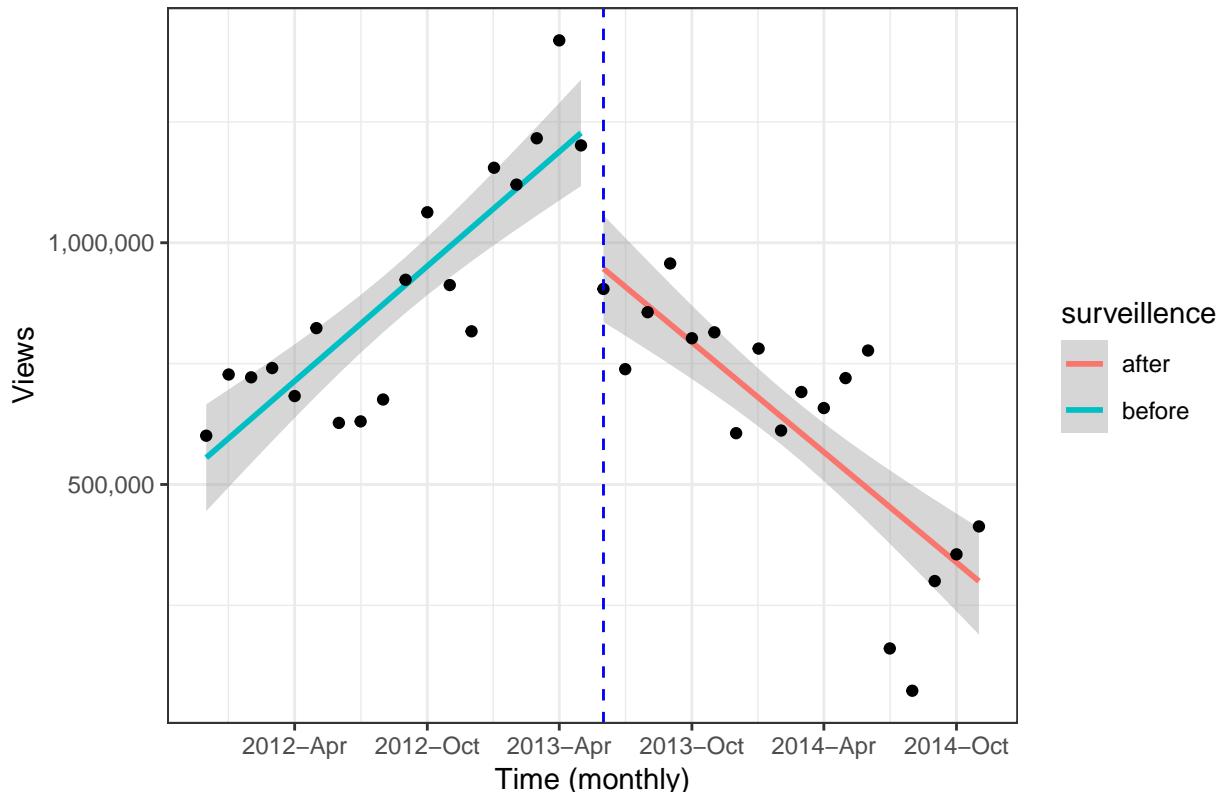
```

```

##      data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -341385 -76768 13782 87116 286130
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.074e+07 3.568e+06  5.813 1.87e-06 ***
## month                     -1.248e+03 2.214e+02 -5.638 3.10e-06 ***
## surveillancebefore        -4.008e+07 4.958e+06 -8.083 3.14e-09 ***
## month:surveillancebefore  2.548e+03 3.129e+02  8.142 2.68e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 148200 on 32 degrees of freedom
## Multiple R-squared:  0.7498, Adjusted R-squared:  0.7263
## F-statistic: 31.96 on 3 and 32 DF,  p-value: 9.546e-10

```

Terrorism-related keywords trend before and after June 2013



```
lm_plot_topic(popular_data, 'Popular keywords trend before and after June 2013')
```

```

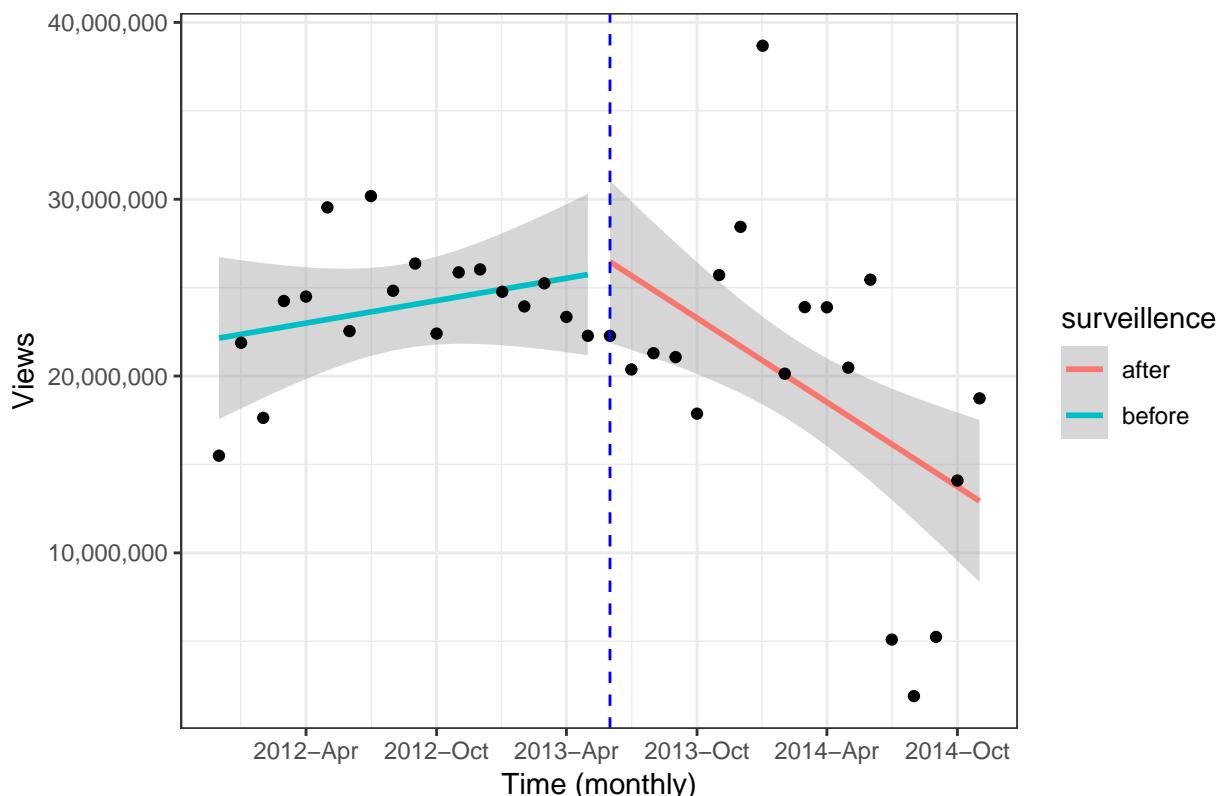
##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##      data = df)
##
## Residuals:
##
```

```

##      Min       1Q     Median       3Q      Max
## -13436279 -3492306     -1164    2864260   17808433
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             440615805 148068036   2.976  0.00553 **
## month                  -26118     9187   -2.843  0.00772 **
## surveillancebefore     -524944327 205785253  -2.551  0.01573 *
## month:surveillancebefore 33073     12987   2.547  0.01589 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6152000 on 32 degrees of freedom
## Multiple R-squared:  0.288, Adjusted R-squared:  0.2212
## F-statistic: 4.314 on 3 and 32 DF,  p-value: 0.01155

```

Popular keywords trend before and after June 2013



```
lm_plot_topic(infra_data, 'Infrastructure security-related keywords trend before and after June 2013')
```

```

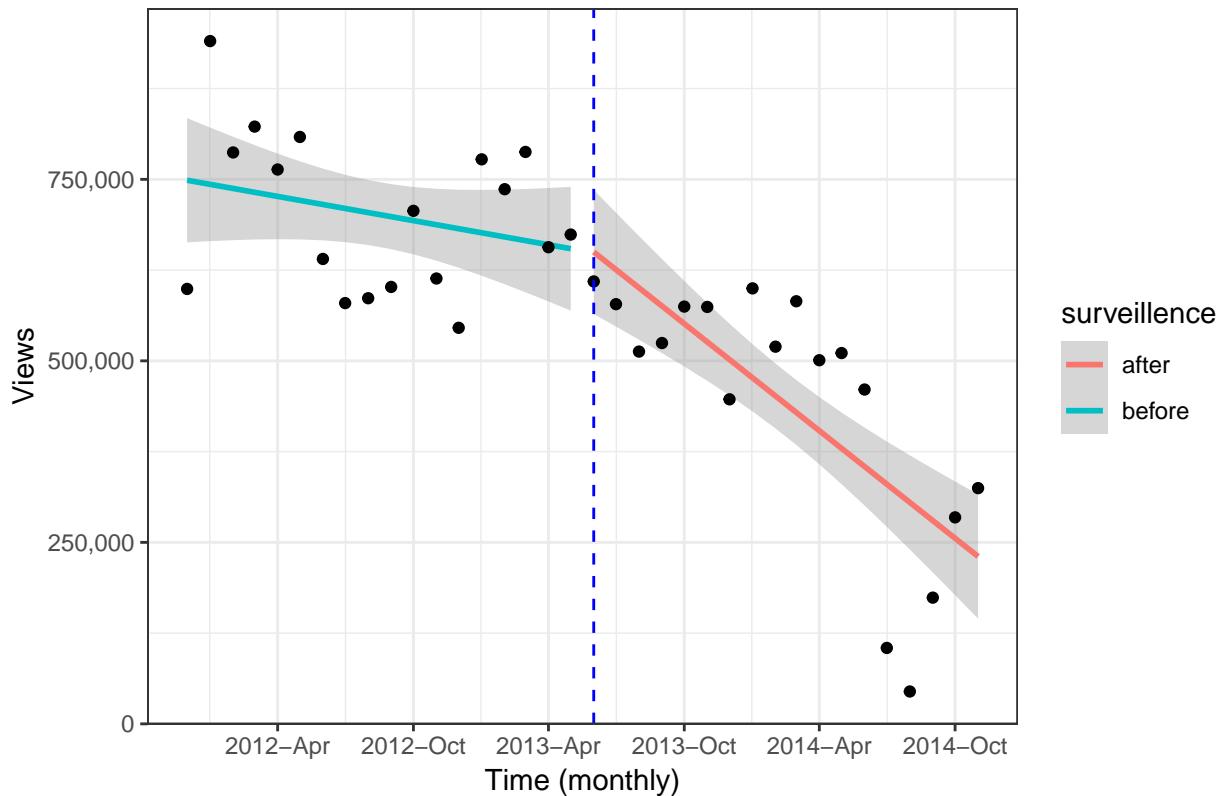
##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##      data = df)
##
## Residuals:
##      Min       1Q     Median       3Q      Max
## -260393  -78202   21543   91386  197325
## 
```

```

## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            13488540.4   2763780.6   4.880 2.81e-05 ***
## month                  -809.7      171.5  -4.721 4.46e-05 ***
## surveillancebefore    -9948117.6   3841107.8  -2.590  0.0143 *
## month:surveillancebefore  627.3      242.4   2.588  0.0144 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114800 on 32 degrees of freedom
## Multiple R-squared:  0.6862, Adjusted R-squared:  0.6567
## F-statistic: 23.32 on 3 and 32 DF,  p-value: 3.434e-08

```

Infrastructure security-related keywords trend before and after June 2014



3. Extended Analysis

3.1 Longer Trend Analysis

The link to our data broke on Saturday May 11th, 2019. Our original plots consisted data from 2012 to 2019. Our group tried to reconstruct the graph from data that was saved on our laptops before the link broke, which is until 2015. Luckily we have original graphs saved as png file. After each reproduced graphs, we included what they used to look like before the link broke, and full data was available.

```

monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize/views= sum/views))

```

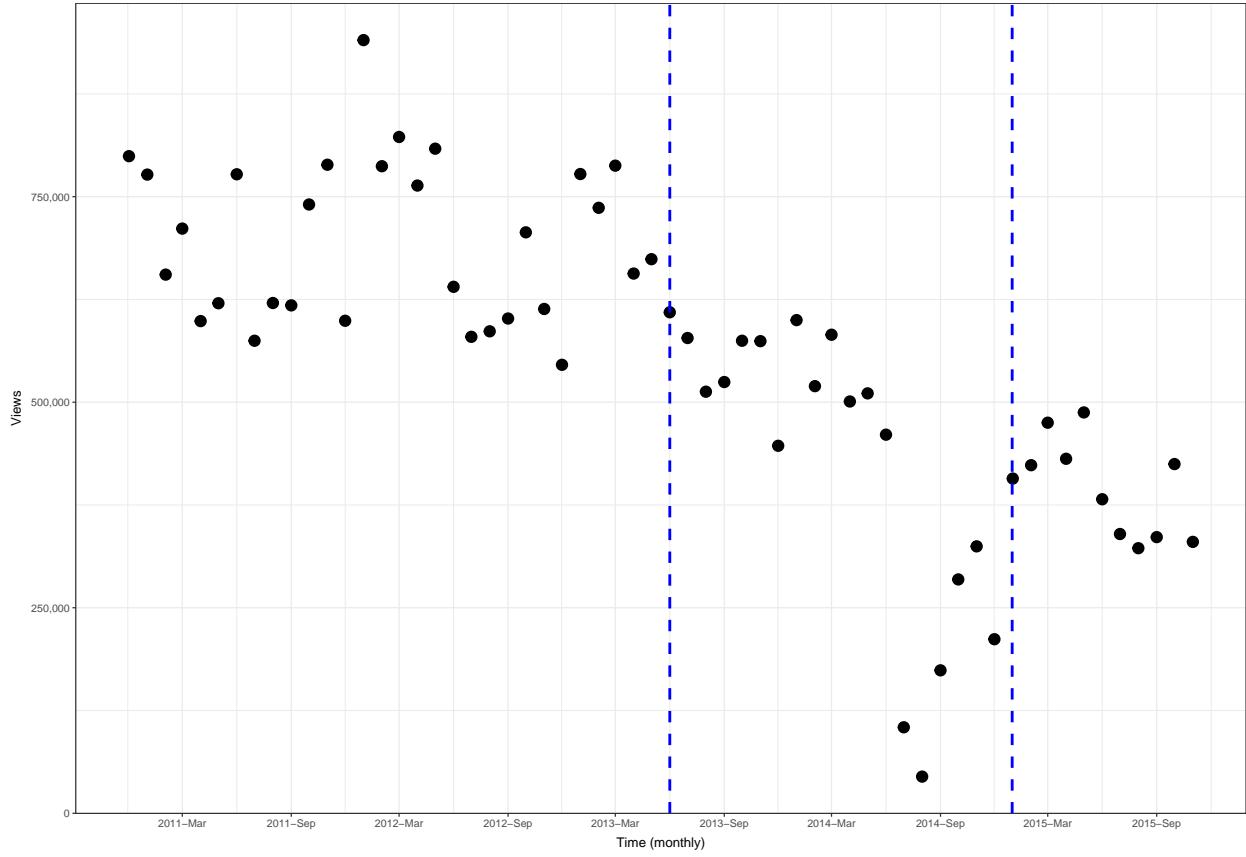
```

monthly_agg$surveillance <- 'before'
monthly_agg$surveillance[monthly_agg$month >= '2013-06-01'] <- 'after'
model <- lm(views ~ month + surveillance + month*surveillance, data = monthly_agg)
summary(model)

##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##      data = monthly_agg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -376110  -77479   25547   74033  242163
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             4.829e+06  1.305e+06   3.701 0.000491 ***
## month                  -2.708e+02  8.004e+01  -3.383 0.001316 **
## surveillancebefore     -3.686e+06  1.793e+06  -2.056 0.044497 *
## month:surveillancebefore 2.417e+02  1.131e+02   2.137 0.036990 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115400 on 56 degrees of freedom
## Multiple R-squared:  0.6412, Adjusted R-squared:  0.6219
## F-statistic: 33.35 on 3 and 56 DF,  p-value: 1.684e-12

monthly_agg$prediction <- predict(model, monthly_agg)
monthly_agg$se <- predict(model, monthly_agg,
                           se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
monthly_agg$LoCI <- monthly_agg$prediction - z.val * monthly_agg$se
monthly_agg$HiCI <- monthly_agg$prediction + z.val * monthly_agg$se
monthly_agg$month <- ymd(monthly_agg$month)
ggplot(monthly_agg,
       aes(x = month,
            y = views)) +
  geom_point(data = monthly_agg, aes(x=month, y = views)) +
  geom_vline(xintercept = as.Date('2013-06-01'), linetype = 2, colour = 'blue') +
  geom_vline(xintercept = as.Date('2014-12-31'), linetype = 2, colour = 'blue') +
  ylab('Views') +
  xlab('Time (monthly)') +
  scale_x_date(date_breaks = "6 month", labels = date_format("%Y-%b")) +
  theme_bw(base_size = 5) +
  scale_y_continuous(labels = comma)

```



First, we take out all the line graphs, and plot scattered graph according to data. The original graph with full data is shown above.

```
monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize/views=sum/views)
monthly_agg$surveillance <- 'before'
monthly_agg$surveillance[monthly_agg$month >= '2013-06-01'] <- 'after'

model <- lm/views ~ month + surveillance + month*surveillance, data = monthly_agg)
monthly_agg$prediction <- predict(model, monthly_agg)
monthly_agg$se <- predict(model, monthly_agg,
                           se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
monthly_agg$LoCI <- monthly_agg$prediction - z.val * monthly_agg$se
monthly_agg$HiCI <- monthly_agg$prediction + z.val * monthly_agg$se
monthly_agg$month <- ymd(monthly_agg$month)
ggplot(monthly_agg,
       aes(x = month,
            y = views)) + geom_point() + stat_smooth(se=T) +
  geom_point(data = monthly_agg, aes(x=month, y = views)) +
  geom_vline(xintercept = as.Date('2013-06-01'), linetype = 2, colour = 'blue') +
  geom_vline(xintercept = as.Date('2014-12-3'), linetype = 2, colour = 'blue') +
  ylab('Views') +
  xlab('Time (monthly)') +
  scale_x_date(date_breaks = "6 month", labels = date_format("%Y-%b")) +
  theme_bw(base_size = 5) +
```

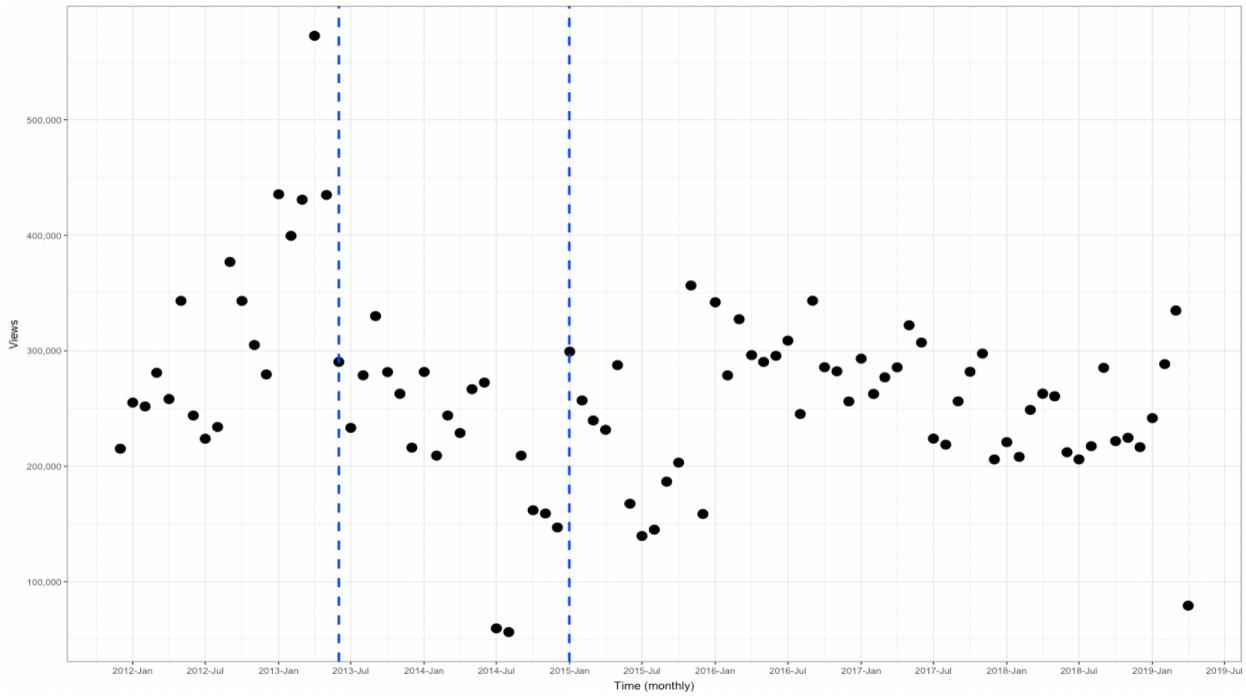
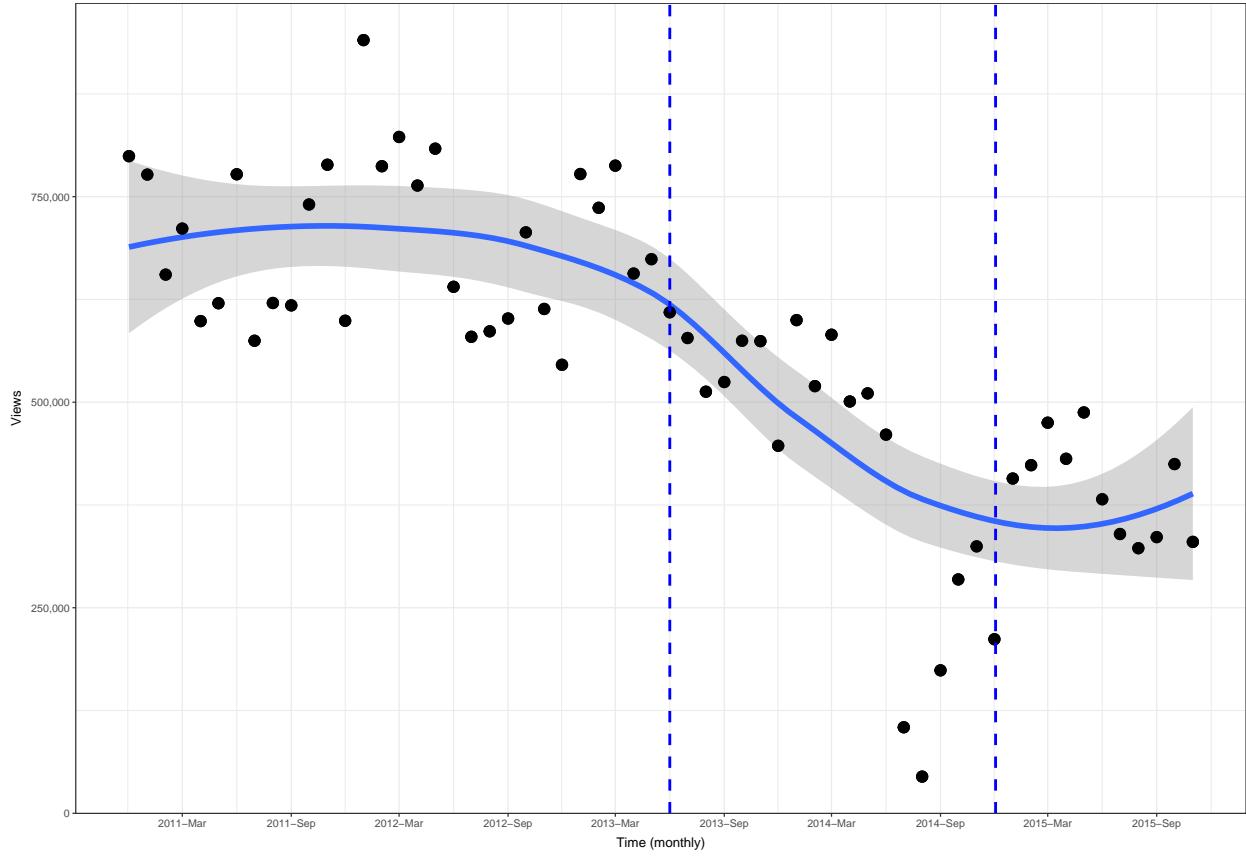


Figure 5: Original Scatter Plot with full data

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



Then, our group had fit a polynomial surface determined by one or more numerical predictors, using local fitting. The graph also displays confidence interval around as gray. The graph below shows rise in trend from 2015 January to July 2016. The graph trend again drops from July 2016 to Jan 2019. The views counts are similar at the beginning of 2016 and end of plot at 2019. This shows that there has been a “trend recovery,” but the trend again drops without second “Snowden Revelation.” This may mean that the decrease in trend from 2013 to 2015 may be due to other factors rather than due to NSA paranoia.

The original graph with full data is shown below.

```
monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize/views=sum(views))
monthly_agg$surveillance <- 'before'
monthly_agg$surveillance[monthly_agg$month >= '2013-06-01'] <- 'after'
model <- lm(views ~ month + surveillance + month*surveillance, data = monthly_agg)
monthly_agg$prediction <- predict(model, monthly_agg)
monthly_agg$se <- predict(model, monthly_agg,
                           se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
monthly_agg$LoCI <- monthly_agg$prediction - z.val * monthly_agg$se
monthly_agg$HiCI <- monthly_agg$prediction + z.val * monthly_agg$se
monthly_agg$month <- ymd(monthly_agg$month)
monthly_agg$month <- as.numeric(monthly_agg$month)
fit <- lm(views~poly(month,4,raw=TRUE),monthly_agg)
plot(views~month,monthly_agg)
curve(predict(fit,newdata=data.frame(month=x)),add=T)
```

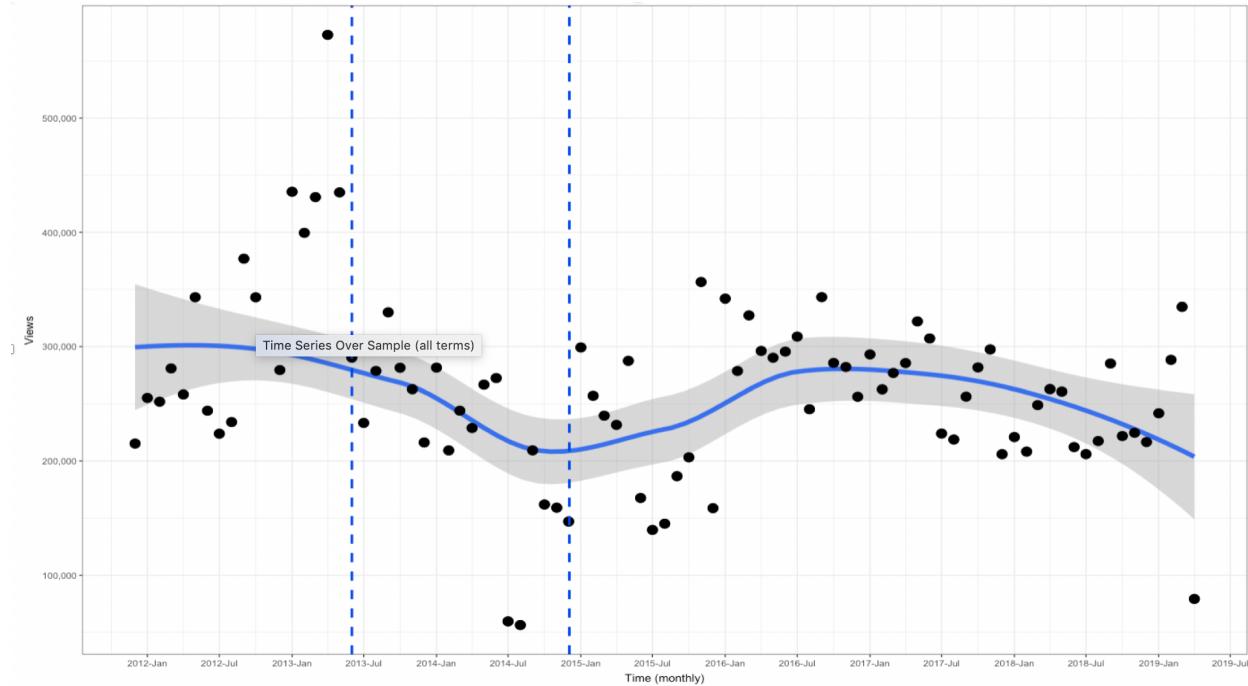
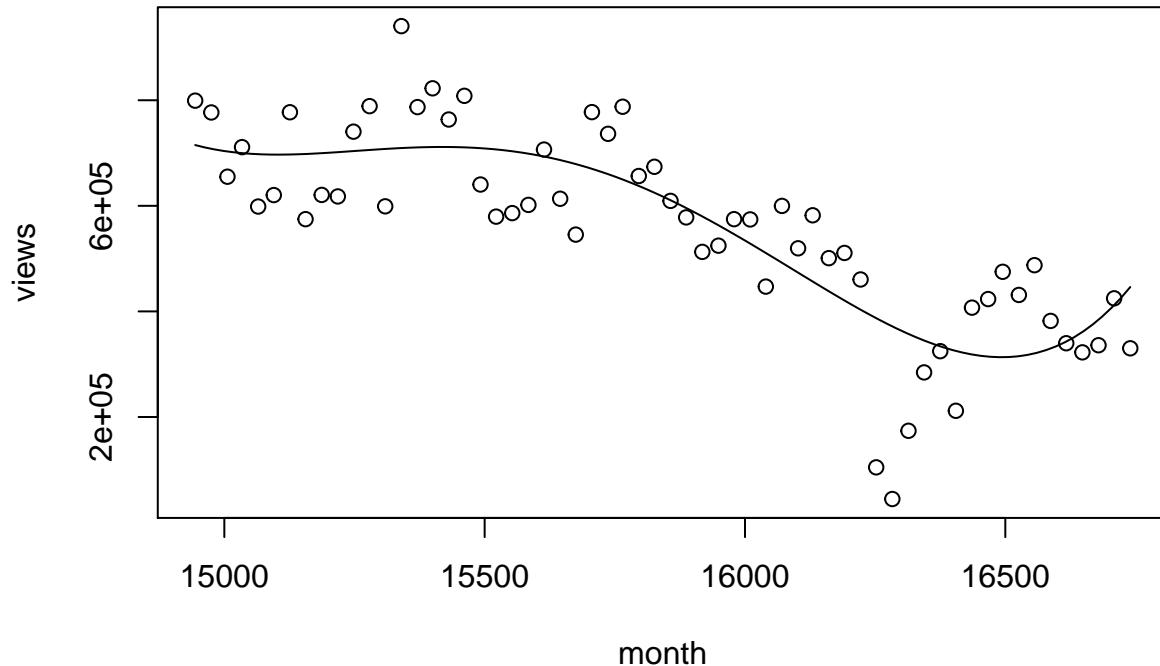


Figure 6: Original Stat Smooth Curve with full data



Then, our group had went further and fit a polynomial graph of degree 4. The curve above shows that there is sharper decrease in trend at the end of graph, far from 2013 region. This again backs our claim that there is a decrease in 2018 to 2019 without another “Snowden Revelation.” This means the decrease in trend after 2013 can be attributed to another resason than Snowden Revelation.

```
monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize/views= sum/views))
```

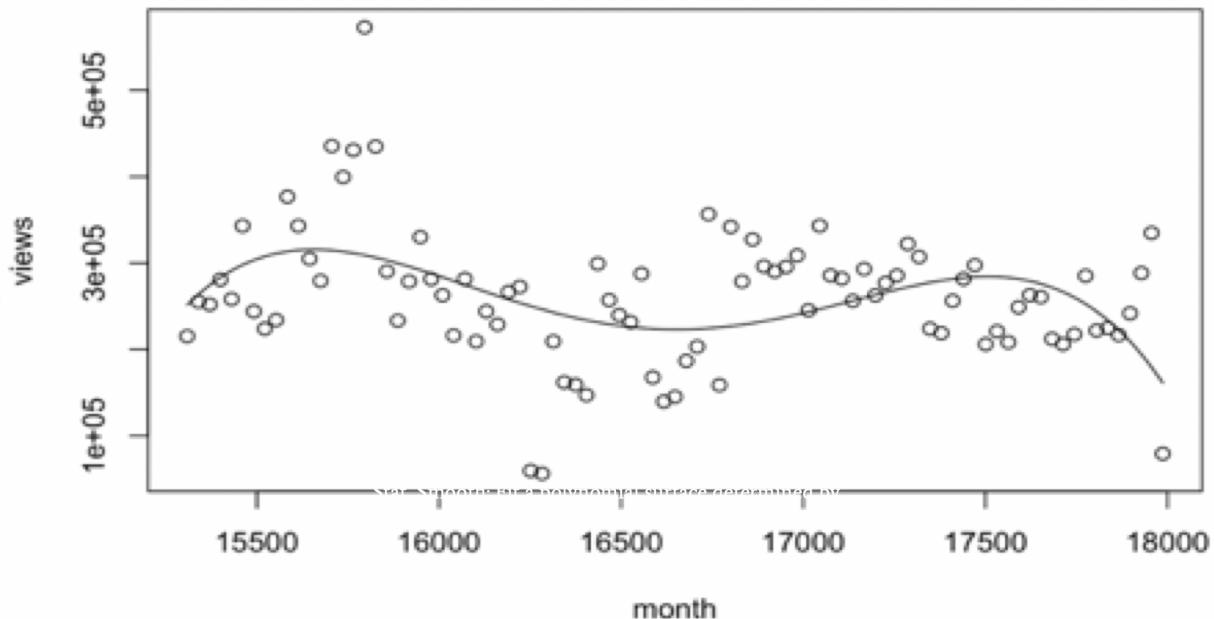


Figure 7: Original Polynomial curve n4 with full data

```

monthly_agg$surveillance <- 'before'
monthly_agg$surveillance[monthly_agg$month >= '2013-06-01'] <- 'after'

model <- lm/views ~ month + surveillance + month*surveillance, data = monthly_agg)
monthly_agg$prediction <- predict(model, monthly_agg)
monthly_agg$se <- predict(model, monthly_agg,
                           se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
monthly_agg$LoCI <- monthly_agg$prediction - z.val * monthly_agg$se
monthly_agg$HiCI <- monthly_agg$prediction + z.val * monthly_agg$se
monthly_agg$month <- ymd(monthly_agg$month)
monthly_agg$month <- as.numeric(monthly_agg$month)
fit <- lm/views~poly(month,5,raw=TRUE),monthly_agg)
plot/views~month,monthly_agg)
curve(predict(fit,newdata=data.frame(month=x)),add=T)

## Warning in predict.lm(fit, newdata = data.frame(month = x)): prediction
## from a rank-deficient fit may be misleading

```

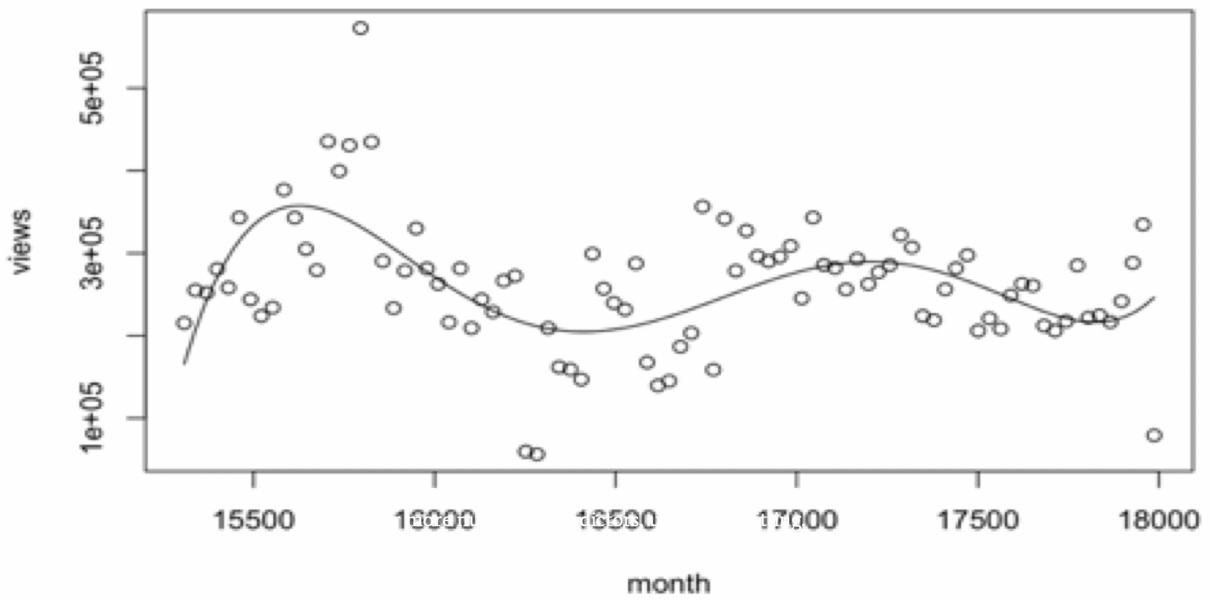
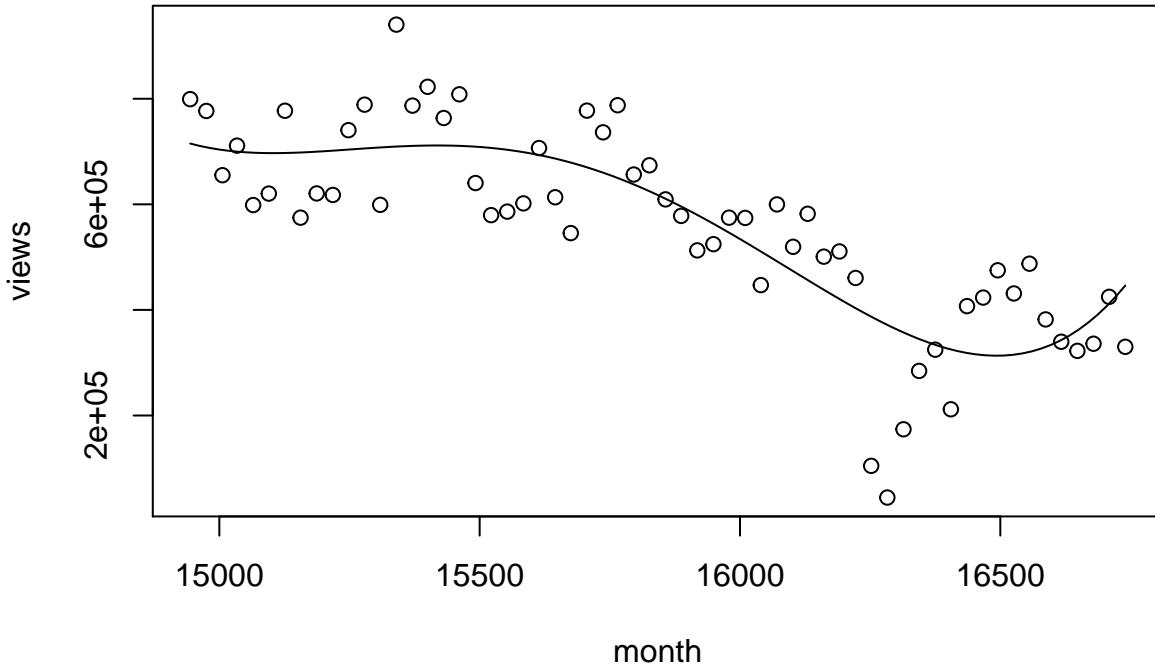


Figure 8: Original Polynomial curve n5 with full data



Then, our group had went further and fit a polynomial graph of degree 5 instead of 4. This may be overfitting, but wanted to analyze the graph to full extent. This graph also shows that there has been decrease in trend from 2018 to 2019, but end of the trend is actually increasing. This fluctuating trend shows that author of original paper's claim that chilling effect is not caused due to NSA paranoia.

```
monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize/views= sum/views)
monthly_agg$surveilence <- 'before'
```

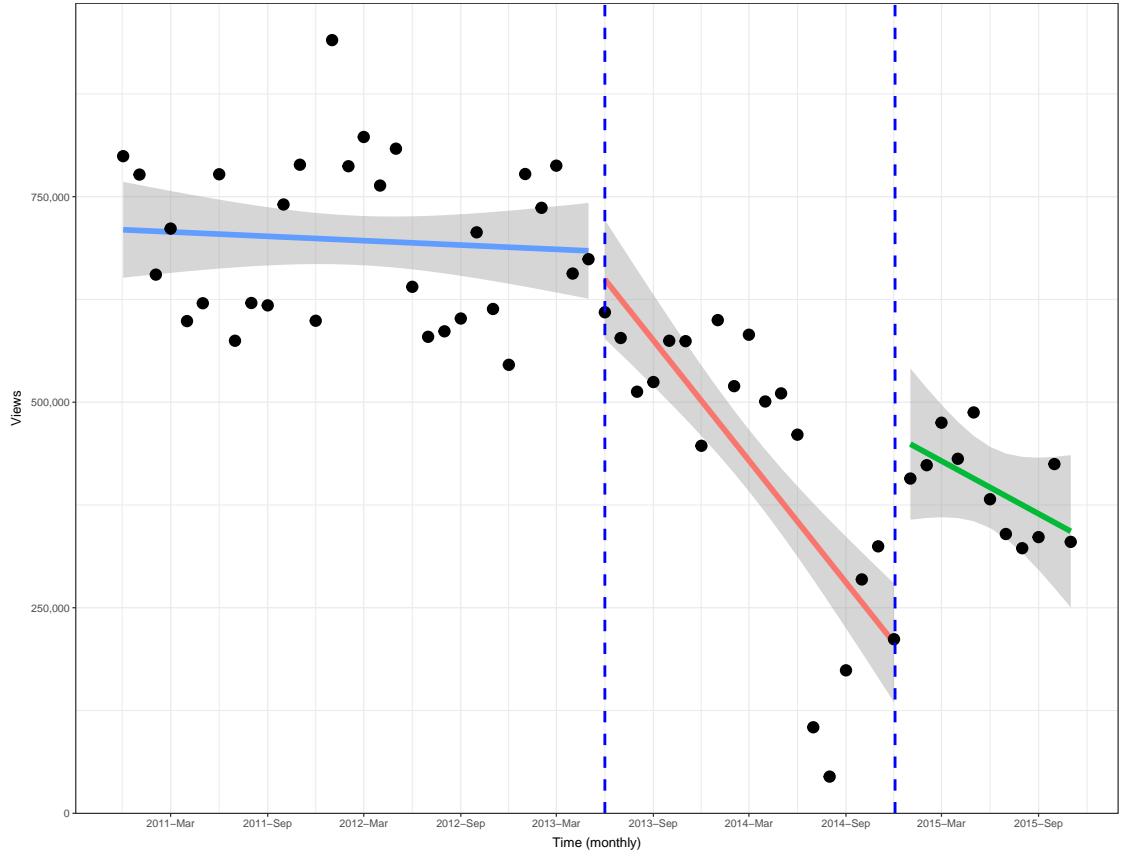
```

monthly_agg$surveillance[monthly_agg$month >= '2013-06-01'] <- 'after'
monthly_agg$surveillance[monthly_agg$month >= '2014-12-3'] <- 'after_after'
model <- lm(views ~ month + surveillance + month*surveillance, data = monthly_agg)
summary(model)

##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##      data = monthly_agg)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -261116 -60090  -3068   74682  242163 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             1.344e+07  2.212e+06  6.078  1.28e-07 ***
## month                  -8.068e+02  1.371e+02 -5.884  2.61e-07 ***
## surveillanceafter      -7.257e+06  5.632e+06 -1.289   0.203    
## surveillancebefore     -1.230e+07  2.453e+06 -5.014  6.11e-06 ***
## month:surveillanceafter 4.577e+02  3.410e+02  1.342   0.185    
## month:surveillancebefore 7.778e+02  1.535e+02  5.068  5.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 99590 on 54 degrees of freedom
## Multiple R-squared:  0.7424, Adjusted R-squared:  0.7186 
## F-statistic: 31.13 on 5 and 54 DF,  p-value: 9.146e-15

monthly_agg$prediction <- predict(model, monthly_agg)
monthly_agg$se <- predict(model, monthly_agg,
                           se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
monthly_agg$LoCI <- monthly_agg$prediction - z.val * monthly_agg$se
monthly_agg$HiCI <- monthly_agg$prediction + z.val * monthly_agg$se
monthly_agg$month <- ymd(monthly_agg$month)
ggplot(monthly_agg,
       aes(x = month,
            y = prediction)) +
  geom_smooth(aes(ymin = LoCI,
                  ymax = HiCI,
                  color = surveillance),
              stat = "identity") +
  geom_point(data = monthly_agg, aes(x=month, y = views)) +
  geom_vline(xintercept = as.Date('2013-06-01'), linetype = 2, colour = 'blue') +
  geom_vline(xintercept = as.Date('2014-12-3'), linetype = 2, colour = 'blue') +
  ylab('Views') +
  xlab('Time (monthly)') +
  scale_x_date(date_breaks = "6 month", labels = date_format("%Y-%b")) +
  theme_bw(base_size = 5) +
  scale_y_continuous(labels = comma)

```



The graph below shows that there is stable trend after January 2015. This graph shows that view after 2015 is higher than from 2013 to 2015. This is only graph that may support author's claim that there had been a trend recovery, and trend stays constant. However, the data is not segmented in equal time bins. This made our group explore further in equal time segments of trends after 2015, in the following works below.

```

monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize/views=sum/views)
monthly_agg$surveillance <- 'before'
monthly_agg$surveillance[monthly_agg$month >= '2013-06-01'] <- 'after'
monthly_agg$surveillance[monthly_agg$month >= '2014-12-3'] <- 'after_after'
monthly_agg$surveillance[monthly_agg$month >= '2016-06-3'] <- 'after_after_after'
monthly_agg$surveillance[monthly_agg$month >= '2016-06-3'] <- 'after_after_after_after'
model <- lm(views ~ month + surveillance + month*surveillance, data = monthly_agg)
summary(model)

##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##     data = monthly_agg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -261116  -60090   -3068   74682  242163 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 740000.0    10000.0  74.000  <2e-16 ***
## month        -1000.0     100.0 -10.000  <2e-16 ***
## surveillance  -1000.0     100.0 -10.000  <2e-16 ***
## month:surveillance  1000.0     100.0  10.000  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

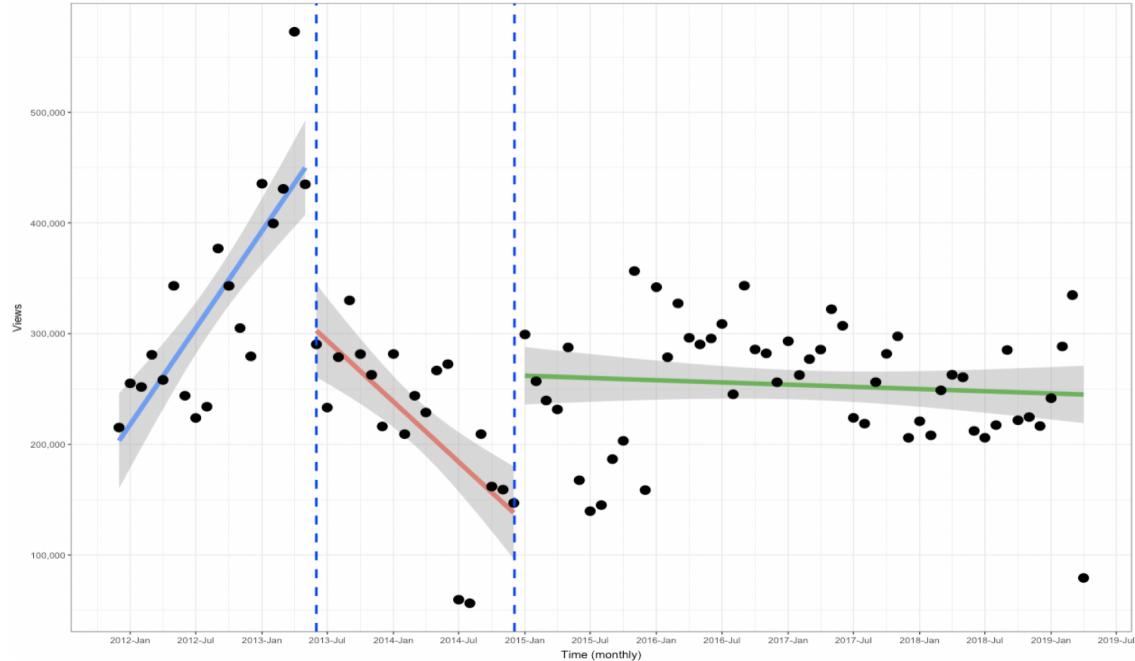


Figure 9: Original Graph with full data

```

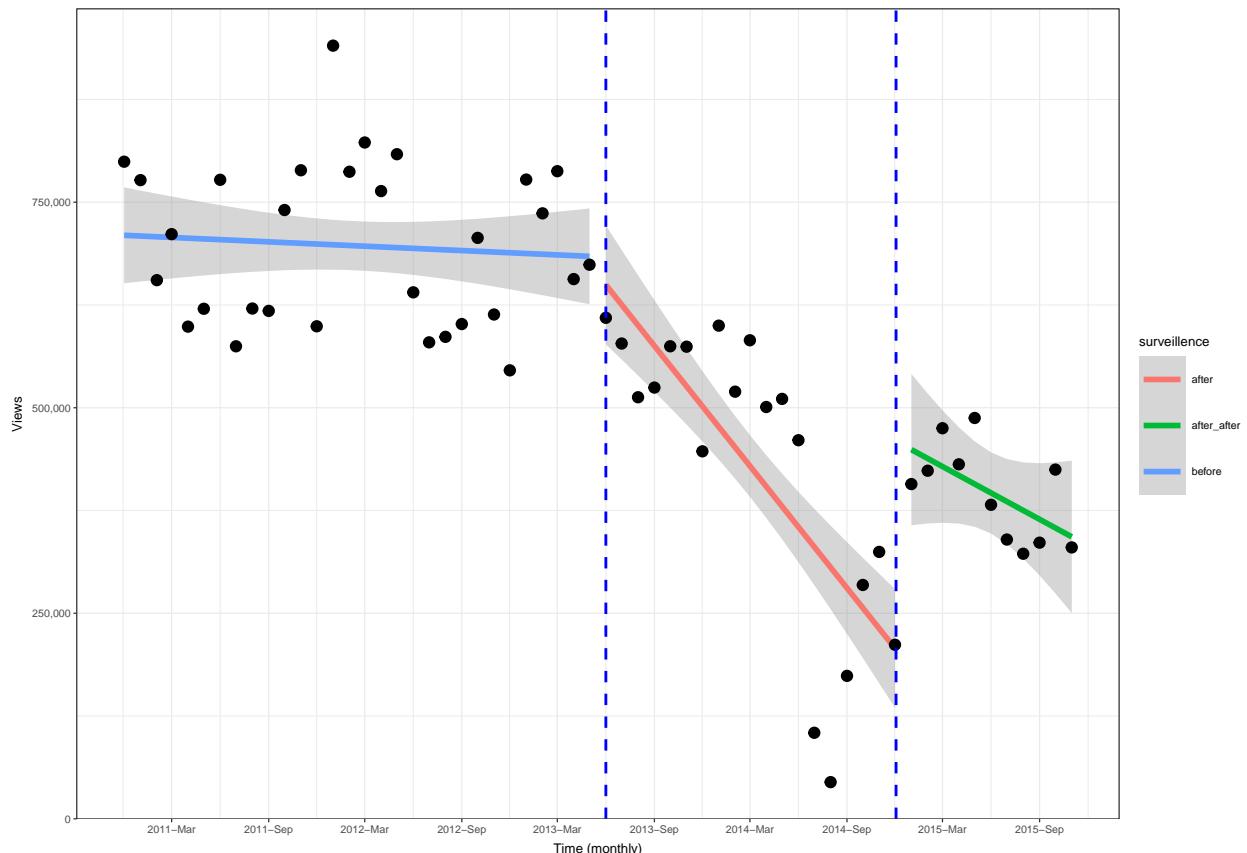
## (Intercept)           1.344e+07  2.212e+06   6.078  1.28e-07 ***
## month                  -8.068e+02  1.371e+02  -5.884  2.61e-07 ***
## surveillanceafter_after -7.257e+06  5.632e+06  -1.289   0.203
## surveillancebefore      -1.230e+07  2.453e+06  -5.014  6.11e-06 ***
## month:surveillanceafter_after 4.577e+02  3.410e+02   1.342   0.185
## month:surveillancebefore     7.778e+02  1.535e+02   5.068  5.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 99590 on 54 degrees of freedom
## Multiple R-squared:  0.7424, Adjusted R-squared:  0.7186
## F-statistic: 31.13 on 5 and 54 DF,  p-value: 9.146e-15

monthly_agg$prediction <- predict(model, monthly_agg)
monthly_agg$se <- predict(model, monthly_agg,
                           se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
monthly_agg$LoCI <- monthly_agg$prediction - z.val * monthly_agg$se
monthly_agg$HiCI <- monthly_agg$prediction + z.val * monthly_agg$se
monthly_agg$month <- ymd(monthly_agg$month)
ggplot(monthly_agg,
       aes(x = month,
            y = prediction)) +
  geom_smooth(aes(ymin = LoCI,
                  ymax = HiCI,
                  color = surveillance),
              stat = "identity") +
  geom_point(data = monthly_agg, aes(x=month, y = views)) +
  geom_vline(xintercept = as.Date('2013-06-01'), linetype = 2, colour = 'blue') +
  
```

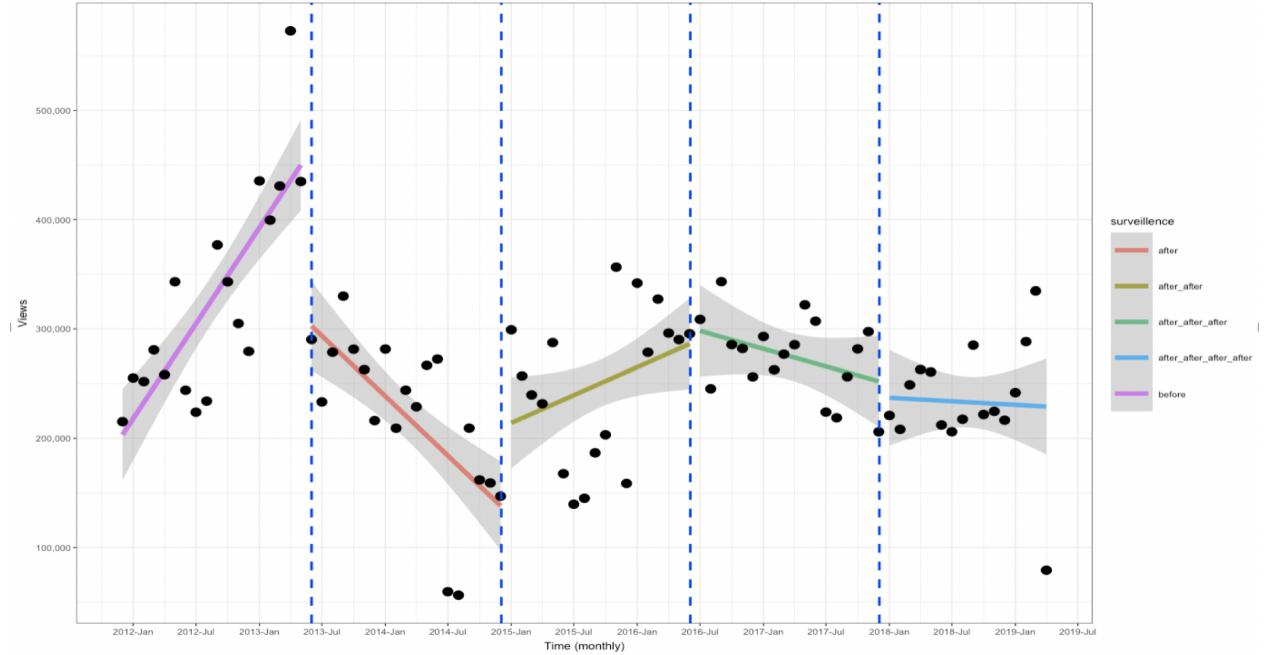
```

geom_vline(xintercept = as.Date('2014-12-3'), linetype = 2, colour = 'blue') +
geom_vline(xintercept = as.Date('2016-06-3'), linetype = 2, colour = 'blue') +
geom_vline(xintercept = as.Date('2016-06-3'), linetype = 2, colour = 'blue') +
ylab('Views') +
xlab('Time (monthly)') +
scale_x_date(date_breaks = "6 month", labels = date_format("%Y-%b")) +
theme_bw(base_size = 5) +
scale_y_continuous(labels = comma)

```



Now, our group had separated the data into equal segments. The graph above shows trend recovery from Jan 2015 to July 2016, trend decrease from July 2016 to Jan 2018, then a stable trend from Jan 2018 to July 2019. This fluctuations may again raise question to paper's author's claim that there exists chilling effect due to Snowden Revelation.



3.2 Per-keyword Analysis

3.2.1 Visualizing sample keywords

```
lm_plot_keyword <- function(input_df, article_name, gg_title){

df <- data.frame(input_df)
df <- df %>%
  group_by(article ,month=floor_date(date, "month")) %>%
  summarize/views=sum(views)) %>%
  filter(article == article_name)

df$surveillance <- 'before'
df$surveillance[df$month >= '2013-06-01'] <- 'after'

model <- lm(vIEWS ~ month + surveillance + month*surveillance, data = df)
print(summary(model))

df$prediction <- predict(model, df)
df$se <- predict(model, df,
                  se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
df$LoCI <- df$prediction - z.val * df$se
df$HiCI <- df$prediction + z.val * df$se

df$month <- ymd(df$month)

ggplot(df,
       aes(x = month,
```

```

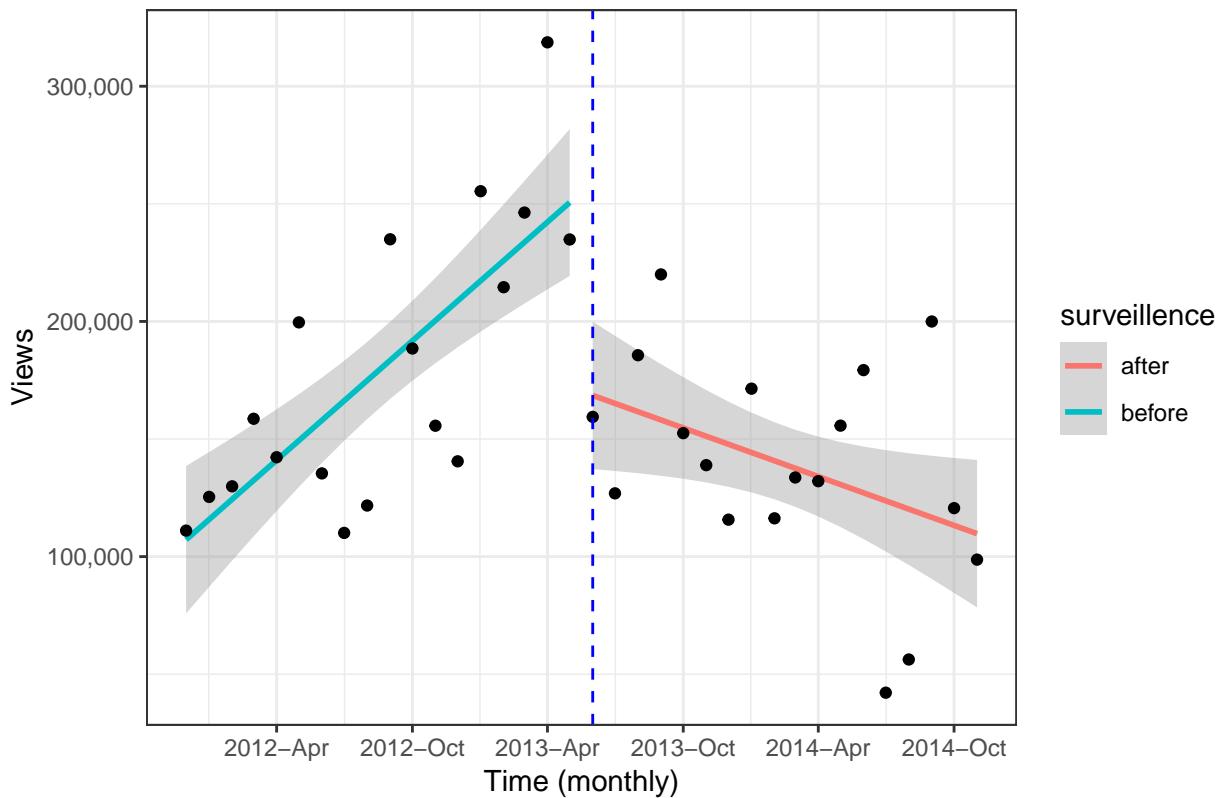
        y = prediction)) +
  geom_smooth(aes(ymin = LoCI,
                  ymax = HiCI,
                  color = surveillance),
              stat = "identity") +
  geom_point(data = df, aes(x=month, y = views)) +
  geom_vline(xintercept = as.Date('2013-06-01'), linetype = 2, colour = 'blue') +
  ylab('Views') +
  xlab('Time (monthly)') +
  scale_x_date(date_breaks = "6 month", labels = date_format("%Y-%b")) +
  scale_y_continuous(labels = comma) +
  ggtitle(gg_title)
}

lm_plot_keyword(terrorism_data, 'al-qaeda', 'Trend for \'al-qaeda\' before and after June 2013')

##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##      data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -81582 -23037  -2093  25332  83296
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.969e+06  1.013e+06   1.943  0.060805 .
## month                  -1.135e+02  6.285e+01  -1.806  0.080314 .
## surveillancebefore     -6.107e+06  1.408e+06  -4.338  0.000134 ***
## month:surveillancebefore 3.909e+02  8.884e+01   4.400  0.000113 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42080 on 32 degrees of freedom
## Multiple R-squared:  0.4909, Adjusted R-squared:  0.4432
## F-statistic: 10.29 on 3 and 32 DF,  p-value: 6.776e-05

```

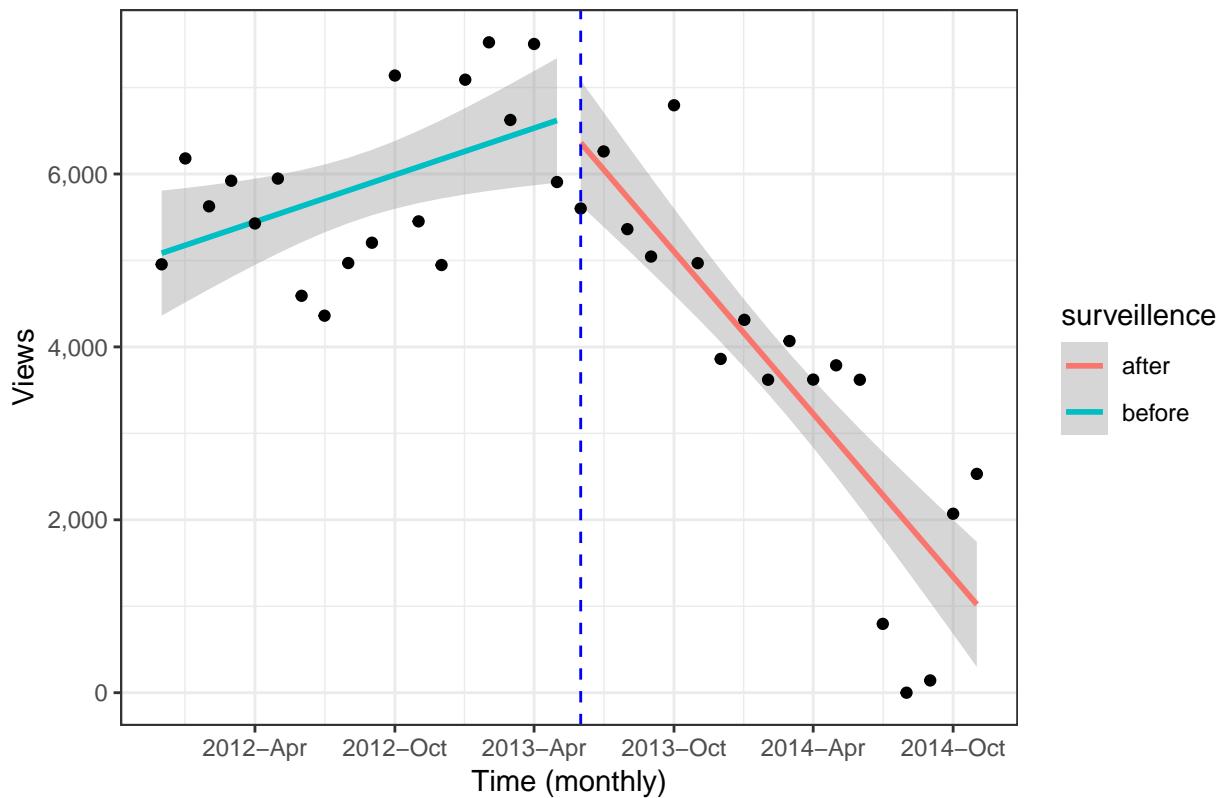
Trend for 'al–qaeda' before and after June 2013



```
lm_plot_keyword(terrorism_data, 'terror', 'Trend for \'terror\' before and after June 2013')

##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##      data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1969.2  -700.2   172.0   754.2  1692.1 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             1.698e+05  2.337e+04   7.268 2.94e-08 ***
## month                  -1.031e+01  1.450e+00  -7.110 4.56e-08 ***
## surveillancebefore     -2.102e+05  3.247e+04  -6.473 2.78e-07 ***
## month:surveillancebefore 1.328e+01  2.049e+00   6.479 2.73e-07 ***
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 970.8 on 32 degrees of freedom
## Multiple R-squared:  0.7564, Adjusted R-squared:  0.7336 
## F-statistic: 33.13 on 3 and 32 DF,  p-value: 6.22e-10
```

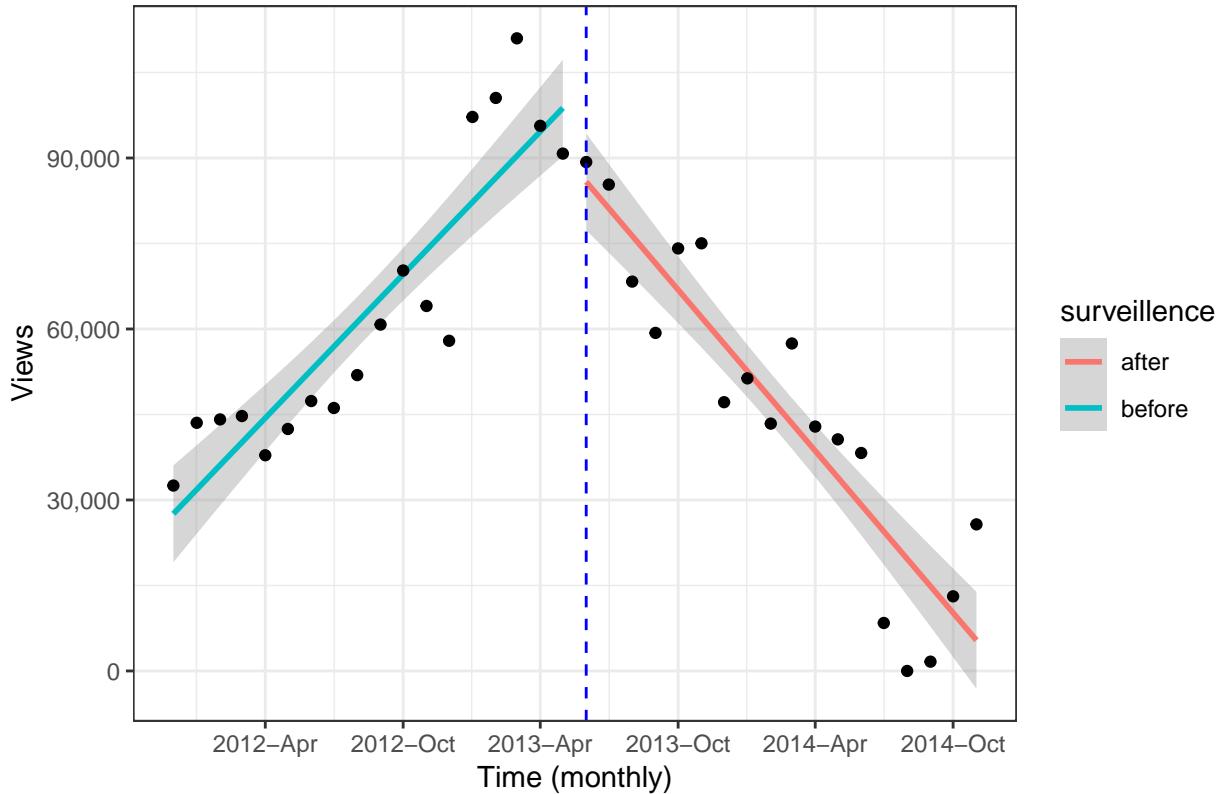
Trend for 'terror' before and after June 2013



```
lm_plot_keyword(terrorism_data, 'recruitment', 'Trend for \'recruitment\' before and after June 2013')

##
## Call:
## lm(formula = views ~ month + surveillance + month * surveillance,
##      data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -20046.9  -8343.9    843.7  7445.4  20621.8 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 2.548e+06  2.745e+05   9.282 1.36e-10 ***
## month       -1.553e+02  1.703e+01  -9.116 2.07e-10 ***
## surveillancebefore -4.629e+06  3.815e+05 -12.132 1.64e-13 ***
## month:surveillancebefore 2.930e+02  2.408e+01   12.169 1.52e-13 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 11410 on 32 degrees of freedom
## Multiple R-squared:  0.8417, Adjusted R-squared:  0.8268 
## F-statistic:  56.7 on 3 and 32 DF,  p-value: 6.647e-13
```

Trend for 'recruitment' before and after June 2013



3.2.2 Quantifying difference between models

```
model_diff <- function(input_df, article_name){

  df <- data.frame(input_df)
  df <- df %>%
    group_by(article ,month=floor_date(date, "month")) %>%
    summarize(views=sum(views)) %>%
    filter(article == article_name)

  df$surveillance <- 'before'
  df$surveillance[df$month >= '2013-06-01'] <- 'after'

  model <- lm(views ~ month + surveillance + month*surveillance, data = df)

  interaction_rmse <- sqrt(mean((model$residuals)^2))
  int_pred <- predict(model, df %>% filter(surveillance != 'before'))

  df2 <- df %>% filter(surveillance == 'before')
  model2 <- lm(views ~ month, data = df2)

  null_pred <- predict(model2, df %>% filter(surveillance != 'before'))
  null_rmse <- sqrt(mean((null_pred - df$views[df$surveillance != 'before'])^2))
  return(c(mean(int_pred - null_pred), interaction_rmse-null_rmse))
}
```

```

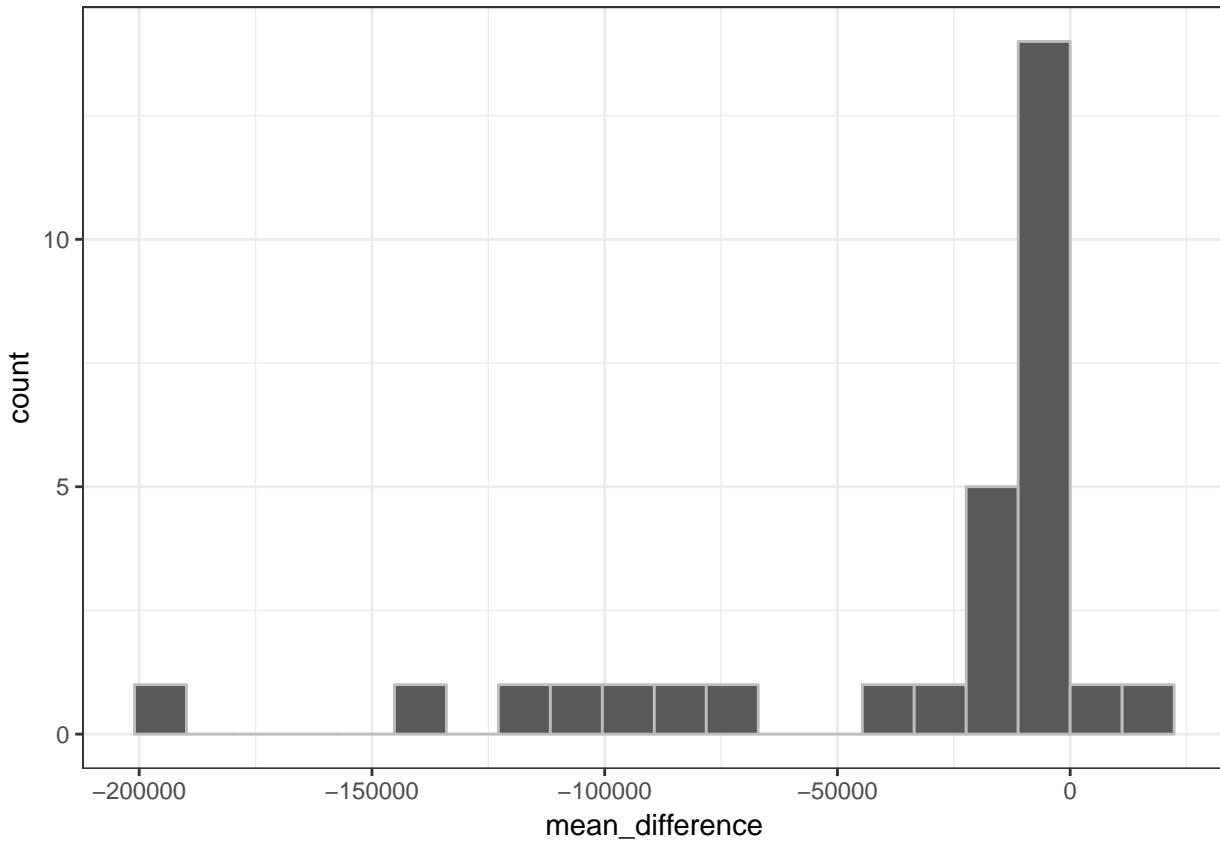
diff <- c()
diff_rmses <- c()

for(i in unique(terrorism_data$article)){
  res = model_diff(terrorism_data, i)
  diff <- c(diff, res[1])
  diff_rmses <- c(diff_rmses, res[2])
}

diff_results <- data.frame("mean_difference" = diff, "rmse_difference" = diff_rmses)

ggplot(diff_results, aes(x = mean_difference)) +
  geom_histogram(boundary = 0, bins = 20, color = 'grey')

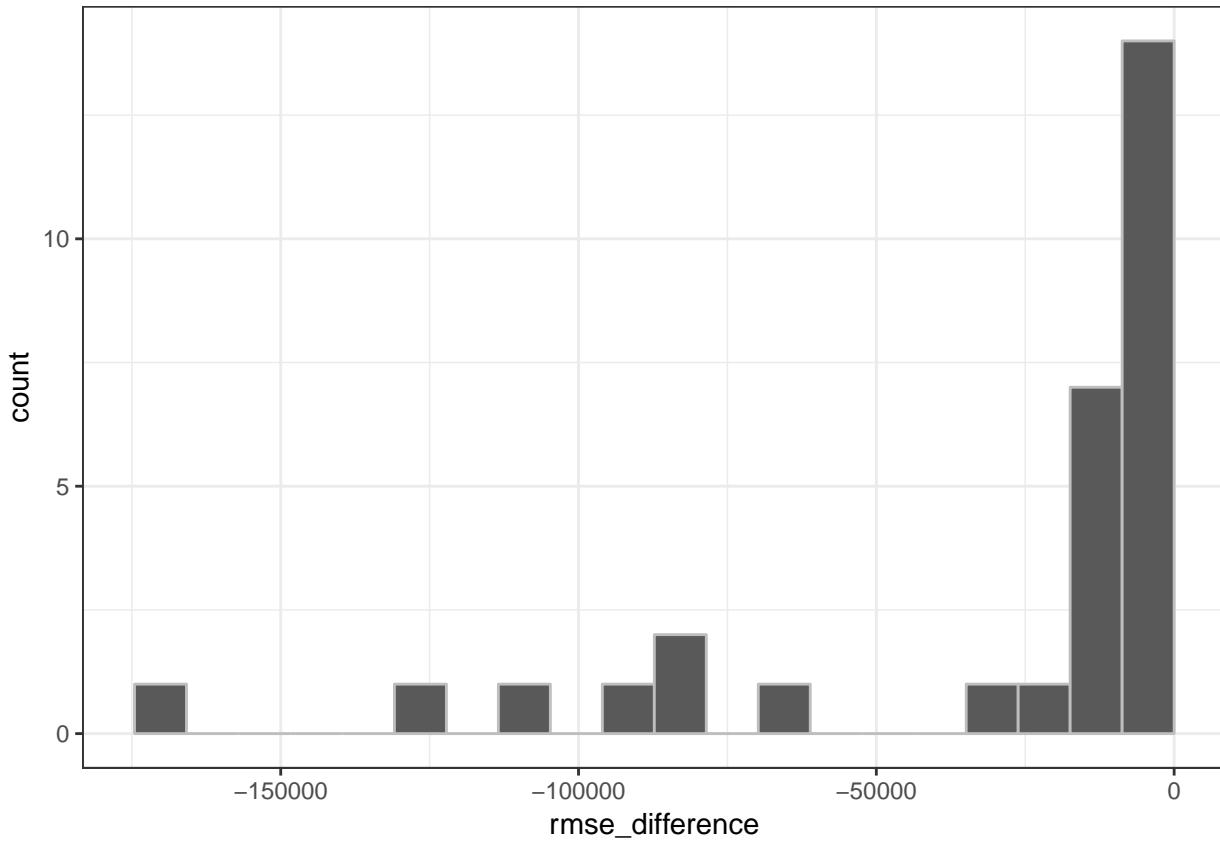
```



```

ggplot(diff_results, aes(x = rmse_difference)) +
  geom_histogram(boundary = 0, bins = 20, color = 'grey')

```



3.3 Time-series Analysis

4. Summary

In our longer trend analysis, our group had found that trend fluctuations shown in polynomial fit graphs ($n=4,5$) show that there the chilling effect in 2013 may have been caused due to other reasons rather than NSA paranoia. When our group had separated the data into equal time segments. The graph shows trend recovery from Jan 2015 to July 2016, trend decrease from July 2016 to Jan 2018, then a stable trend from Jan 2018 to July 2019. Again, this fluctuations raise concern to paper's author's claim that there exists chilling effect due to Snowden Revelation.

The following is a list of all packages used to generate these results. (Leave at very end of file.)

```
sessionInfo()

## R version 3.5.1 (2018-07-02)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS 10.14.4
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
```

```

## attached base packages:
## [1] stats      graphics   grDevices utils     datasets   methods    base
##
## other attached packages:
## [1] bindr_0.2.2       wikipediatrend_2.1.1 lubridate_1.7.4
## [4] forcats_0.3.0     stringr_1.4.0       dplyr_0.7.7
## [7] purrr_0.2.5       readr_1.1.1        tidyverse_1.2.1
## [10] tibble_2.1.1      ggplot2_3.1.1      scales_1.0.0
## [13] scales_1.0.0
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.1         cellranger_1.1.0 compiler_3.5.1   pillar_1.3.1
## [5] plyr_1.8.4         bindr_0.1.1        tools_3.5.1     digest_0.6.18
## [9] jsonlite_1.6       evaluate_0.12     nlme_3.1-137   gtable_0.3.0
## [13] lattice_0.20-35   pkgconfig_2.0.2   rlang_0.3.4    cli_1.1.0
## [17] rstudioapi_0.8     yaml_2.2.0        haven_1.1.2    hellno_0.0.1
## [21] withr_2.1.2       xml2_1.2.0        httr_1.4.0     knitr_1.20
## [25] hms_0.4.2         rprojroot_1.3-2   grid_3.5.1     tidyselect_0.2.5
## [29] glue_1.3.1         R6_2.4.0          readxl_1.1.0   rmarkdown_1.10
## [33] modelr_0.1.2      magrittr_1.5       backports_1.1.2 htmltools_0.3.6
## [37] rvest_0.3.3        assertthat_0.2.1  colorspace_1.4-1 labeling_0.3
## [41] stringi_1.4.3     lazyeval_0.2.2    munsell_0.5.0   broom_0.5.0
## [45] crayon_1.3.4

```