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)

2019-05-13 17:39:30

Contents

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 -ment Trouble	Browser Delete	Privacy Sensi- tive	Avoid- 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 surveilance 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 terrism-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

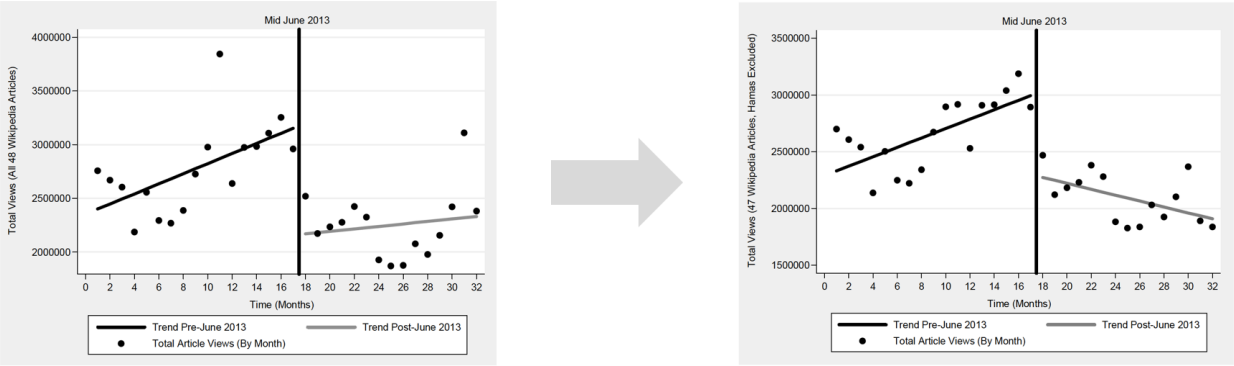


Figure 3: Difference in regression results before and after removing outliers

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 unual 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

Theinterpretation 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 for 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 Keyword-level 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_2005_present.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")
```

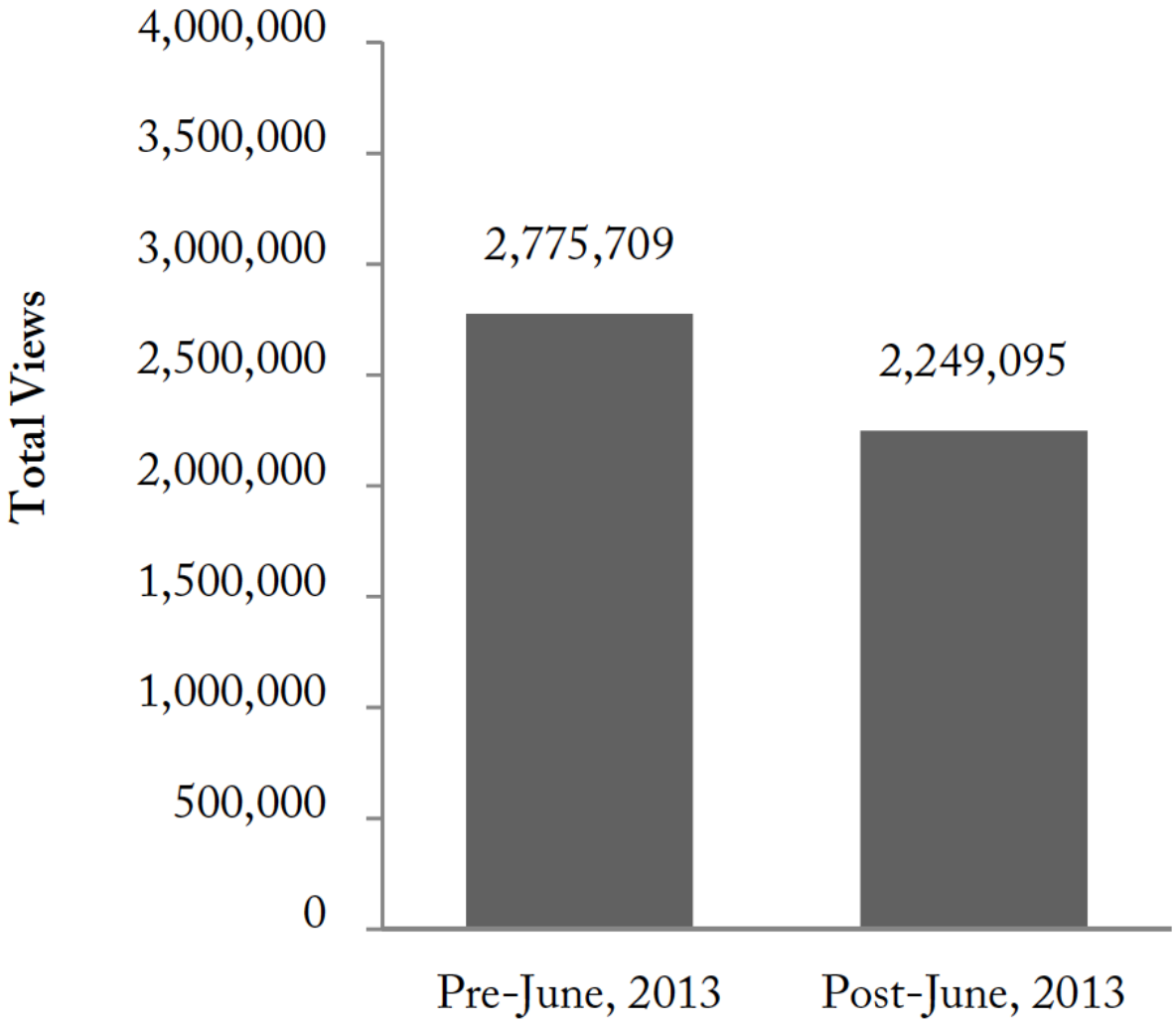
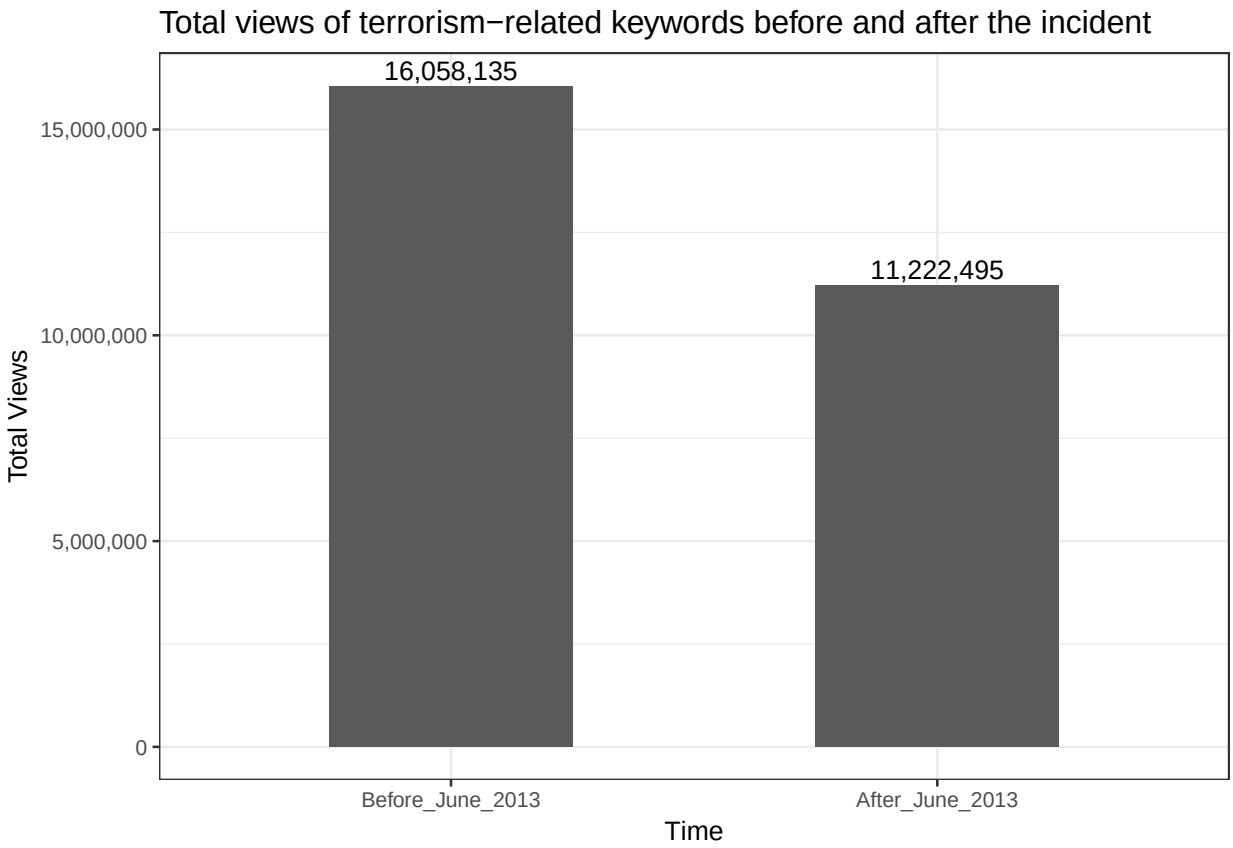


Figure 4: Bar chart comparing monthly views before and after June 2013



This plot aims to vaguely prove the simple belief that the behaviors tend to change after the incident indicating 'chilling effect' which can be observed by the decrease of the views on Wikipedia article. Our relplication has the same trend of decreasing volume, however the number are not exactly the same because we count the number of views 1.5 years before and after the incident. Moreover, it is not clear which keywords or if all of traffic is used to produced the original graph. This chart sparks the idea of existence of the effect that will be investigated further in following sections.

2.4.2 Linear model with interactions: Analysis and Plots

Based on the linear model with interactions discussed earlier, we decide to implement it on three of the topics, terrorism, infrastructure security, and most popular topics. Our replications include both plots and regression analysis for eaxh topics. Keywords within each topic can be found in the appendix of the paper. We have an assumption that the difference between results are mostly due to the difference in data. We did not perform any treatment to outliers before any analysis since the method is not cleary established.

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

model <- lm(views ~ month + surveillence + month*surveillence, 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 = surveillence),
              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 + surveillence + month * surveillence,
##     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 ***
## surveillencebefore       -4.008e+07  4.958e+06  -8.083 3.14e-09 ***
## month:surveillencebefore  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
```

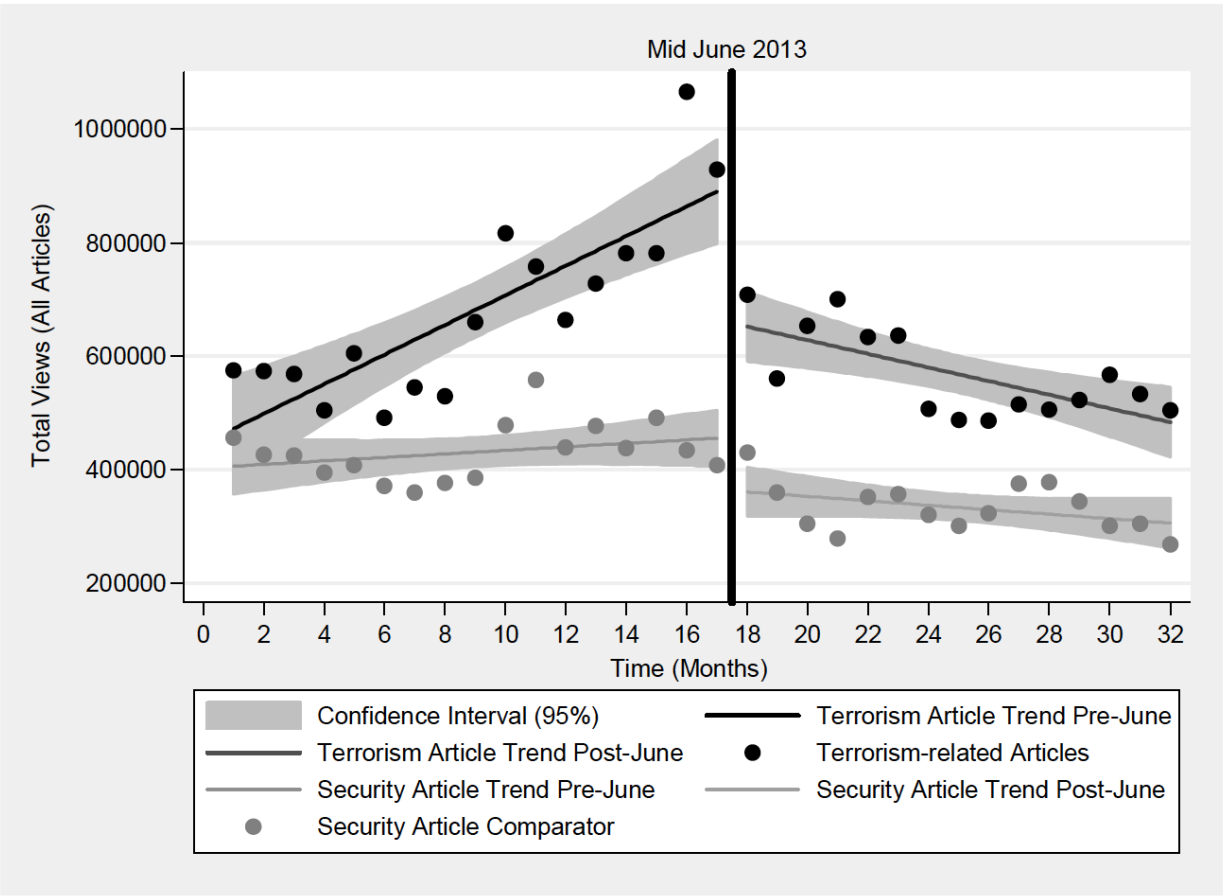
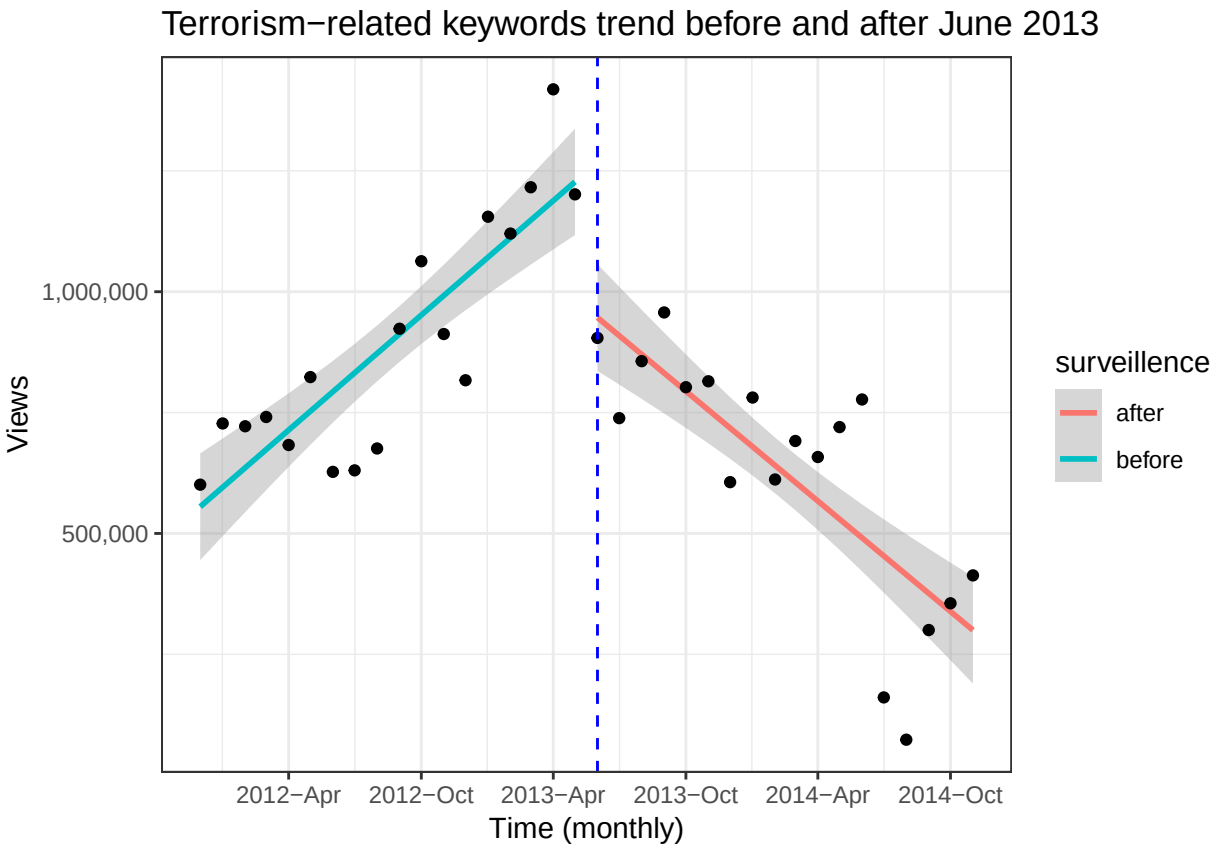


Figure 5: Original result for terrorism-related keywords



The result for terrorism-related articles are very similar to the paper. We got the plot where the trend change from increasing to decreasing and the level is shifted down around 200,000 views. Both coefficients for change in trend and level are significant which is the same as in the original study.

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

```
##
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     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 **
## surveillencebefore       -524944327  205785253  -2.551  0.01573 *
## month:surveillencebefore      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
```

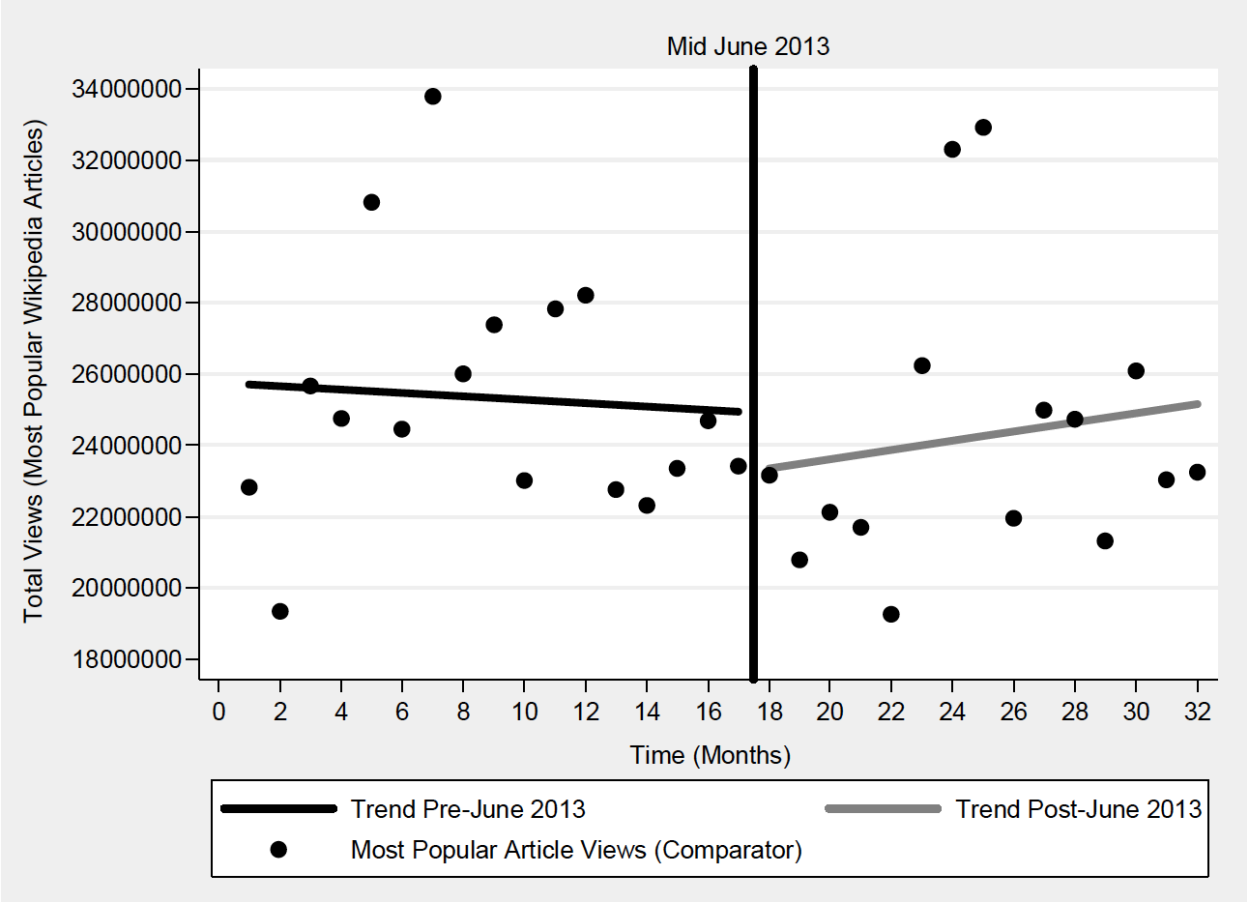
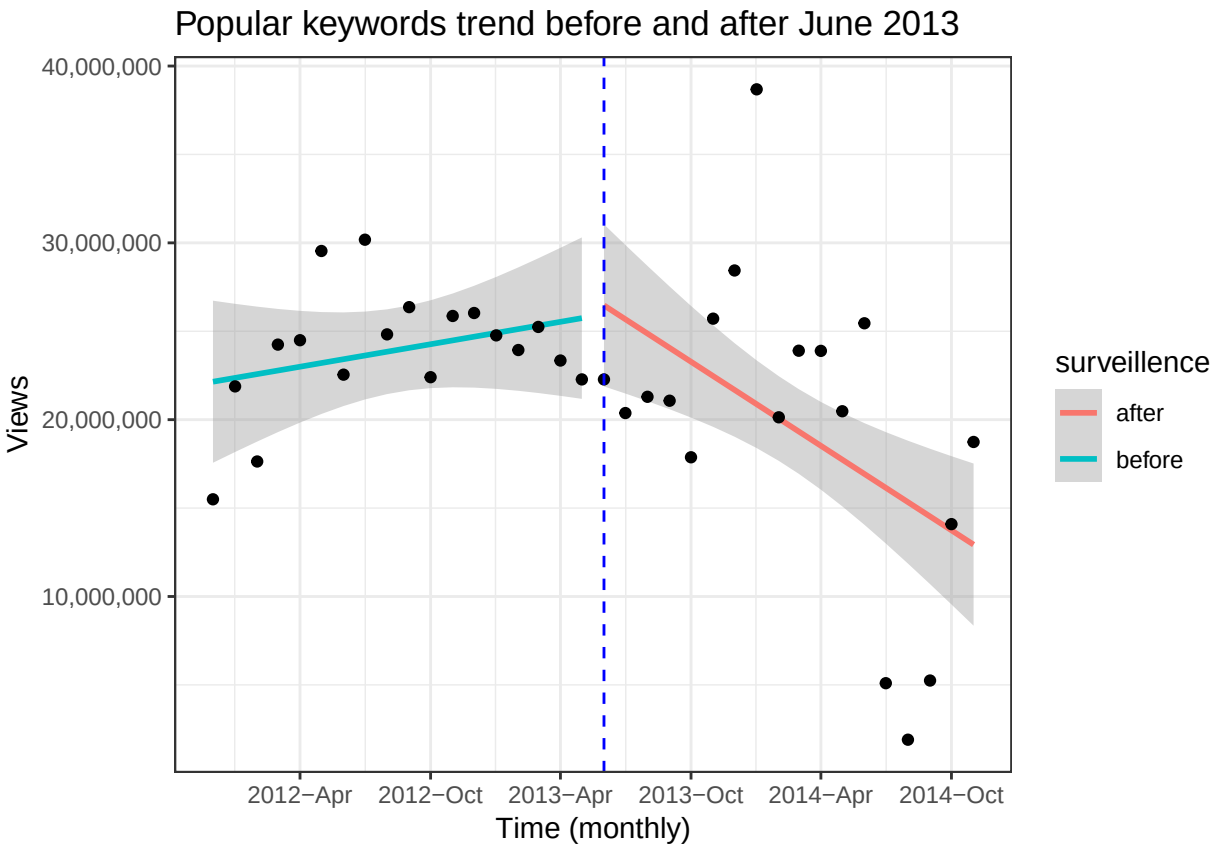


Figure 6: Original result for popular keywords



For most popular articles, our results are quite different from the reference where the trend went slightly down before the event and went up after. Our result is more similar to the terrorism case in term of the trend, but with less change. However, there is almost no shift in level which is same as in the paper. The regression coefficients are significant with lower level, unlike those in the paper which is not significant. However, we would like to mention that the data is quite different both general and outliers.

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

```
##
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     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 ***
## surveillencebefore       -9948117.6  3841107.8  -2.590   0.0143 *
## month:surveillencebefore       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
```

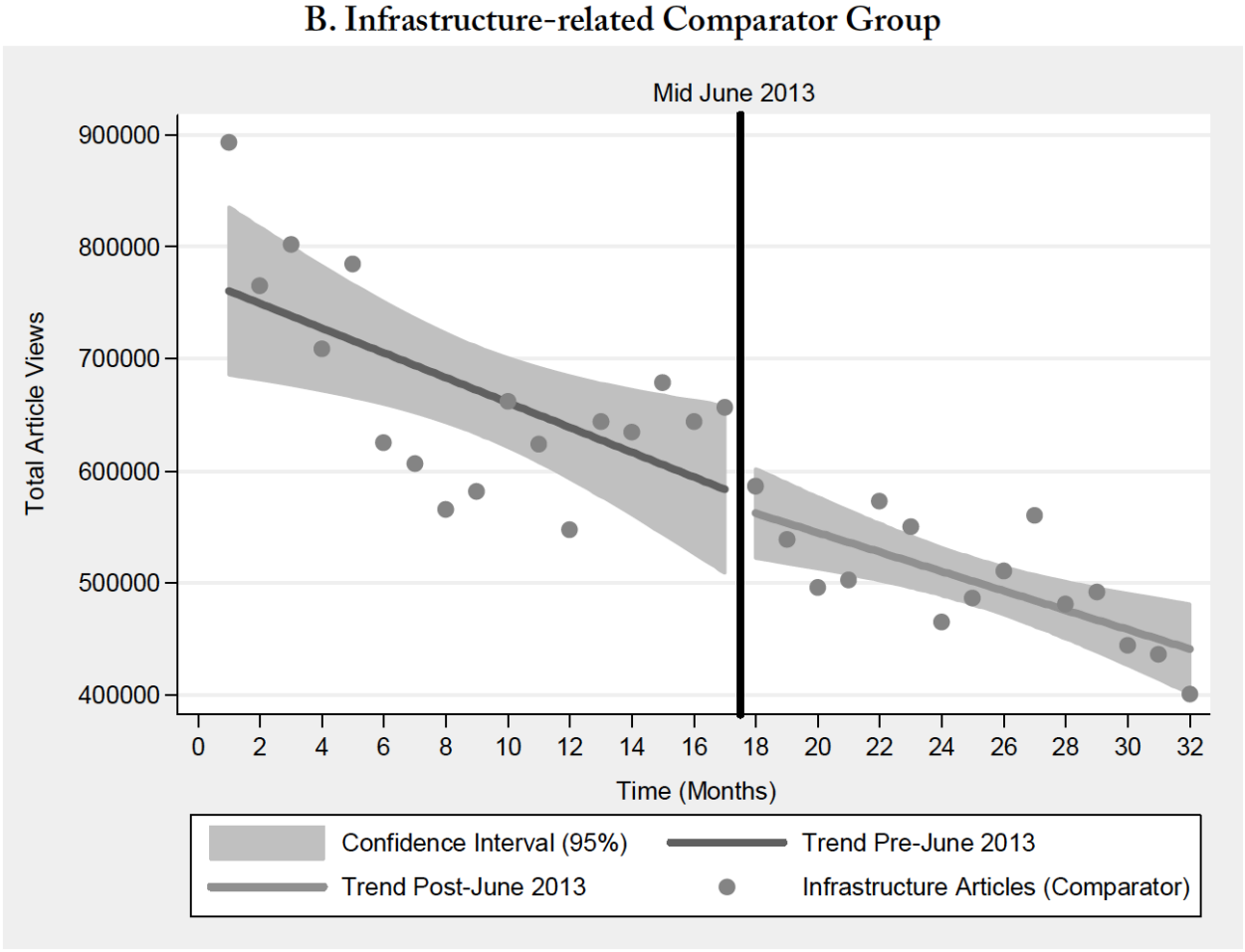
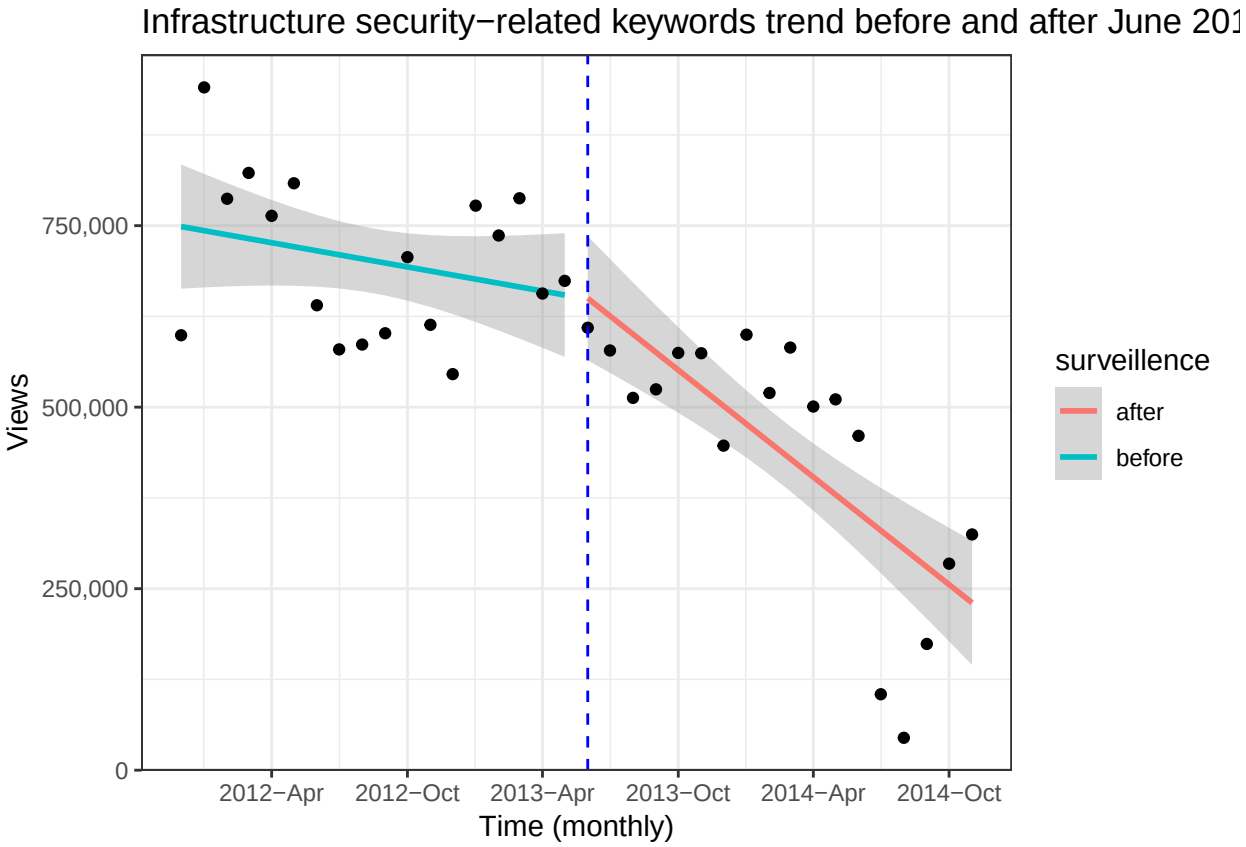


Figure 7: Original result for infra-structure security keywords

```
## F-statistic: 23.32 on 3 and 32 DF,  p-value: 3.434e-08
```



The result is similar to the paper that the trend keeps going down even after the incident. Our result shows slightly steeper trend which may be caused by the few outliers at July to September 2014. Our regression give the significance for the coefficients while they are not significant in the study.

In summary, the reproduced analysis shows that the results can be reproduced using the same regression model. However, it also points out the importance of the way we deal with outliers which greatly affects the analysis. The difference in data source also causes the discrepancy among results. Finally, it raises an interesting point about coefficients that which can be significant regardless of the change in trend and level. We attempt to quantify the change in section 3.2.

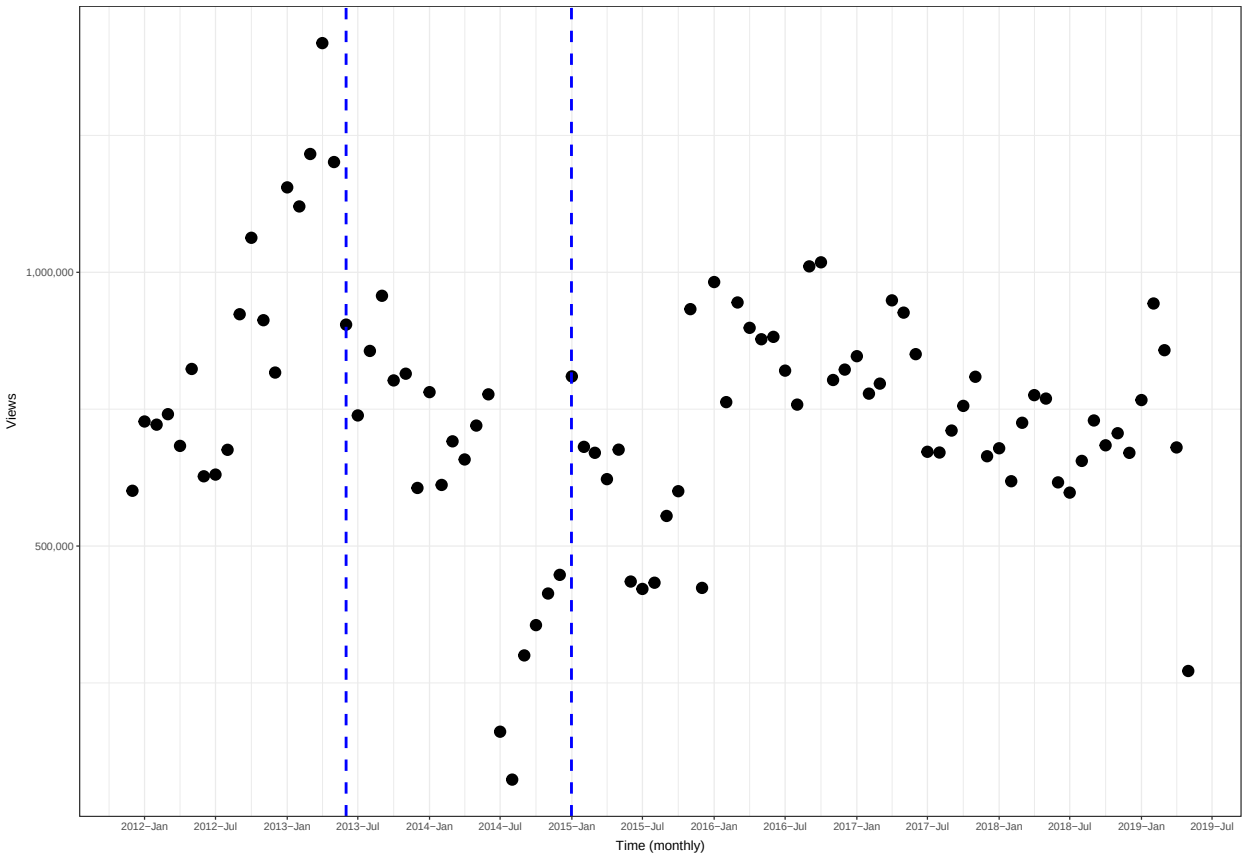
3. Extended Analysis

3.1 Longer Trend Analysis and Trend Recovery

```
terrorism_data_long <- terrorism_data_2005_present %>% filter(date >= '2011-12-01')

monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize(views=sum(views))
monthly_agg$surveillence <- 'before'
monthly_agg$surveillence[monthly_agg$month >= '2013-06-01'] <- 'after'
model <- lm(views ~ month + surveillence + month*surveillence, data = monthly_agg)
summary(model)
```

```
##
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     data = monthly_agg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -601429  -73999   21591  113603  308109
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -4.991e+04  5.804e+05  -0.086    0.932
## month                      4.451e+01  3.424e+01   1.300    0.197
## surveillencebefore        -1.929e+07  4.309e+06  -4.477 2.31e-05 ***
## month:surveillencebefore   1.255e+03  2.764e+02   4.541 1.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 183800 on 86 degrees of freedom
## Multiple R-squared:  0.3133, Adjusted R-squared:  0.2893
## F-statistic: 13.08 on 3 and 86 DF,  p-value: 4.099e-07
```

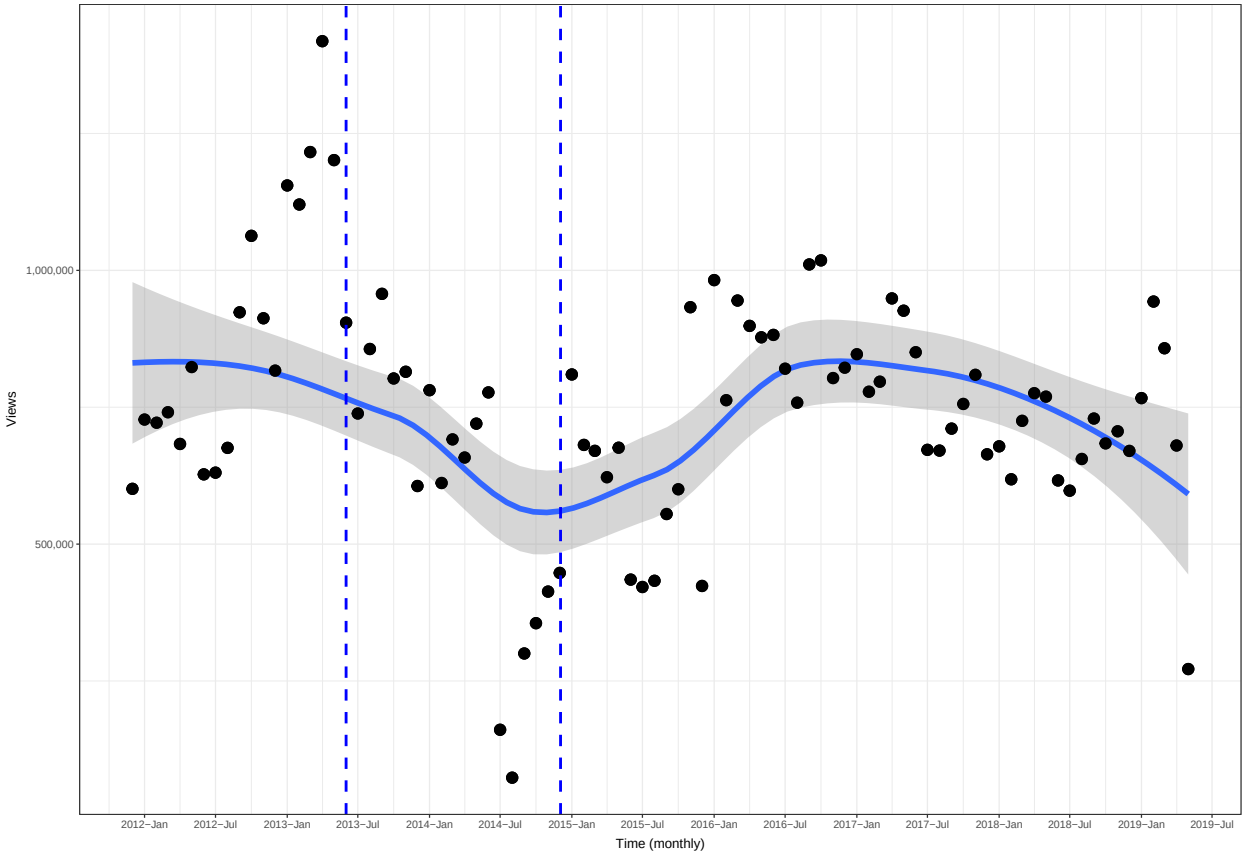
```
monthly_agg$prediction <- predict(model, monthly_agg)
monthly_agg$se <- predict(model, monthly_agg,
                      se.fit = TRUE)$se.fit
z.val <- qnorm(1 - (1 - 0.90)/2)
monthly_agg$LoCI <- monthly_agg$prediction - z.val * monthly_agg$se
monthly_agg$HiCI <- monthly_agg$prediction + z.val *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

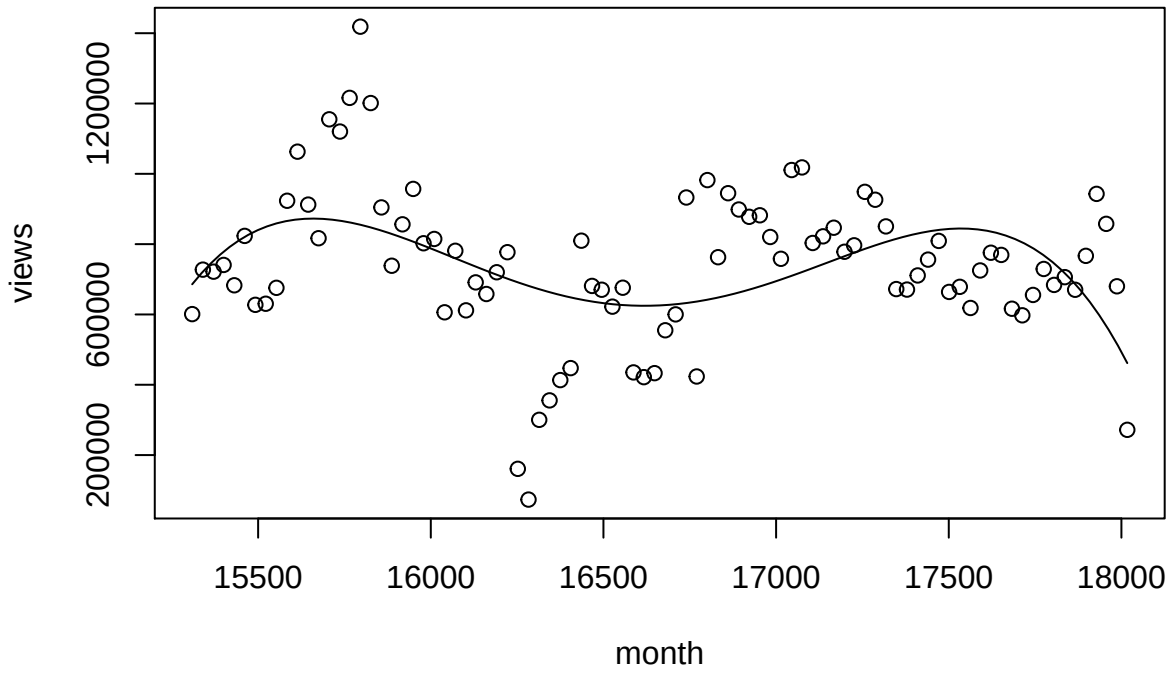
```
monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize(views=sum(views))
monthly_agg$surveillence <- 'before'
monthly_agg$surveillence[monthly_agg$month >= '2013-06-01'] <- 'after'

model <- lm(views ~ month + surveillence + month*surveillence, 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) +
  scale_y_continuous(labels = comma)
```

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



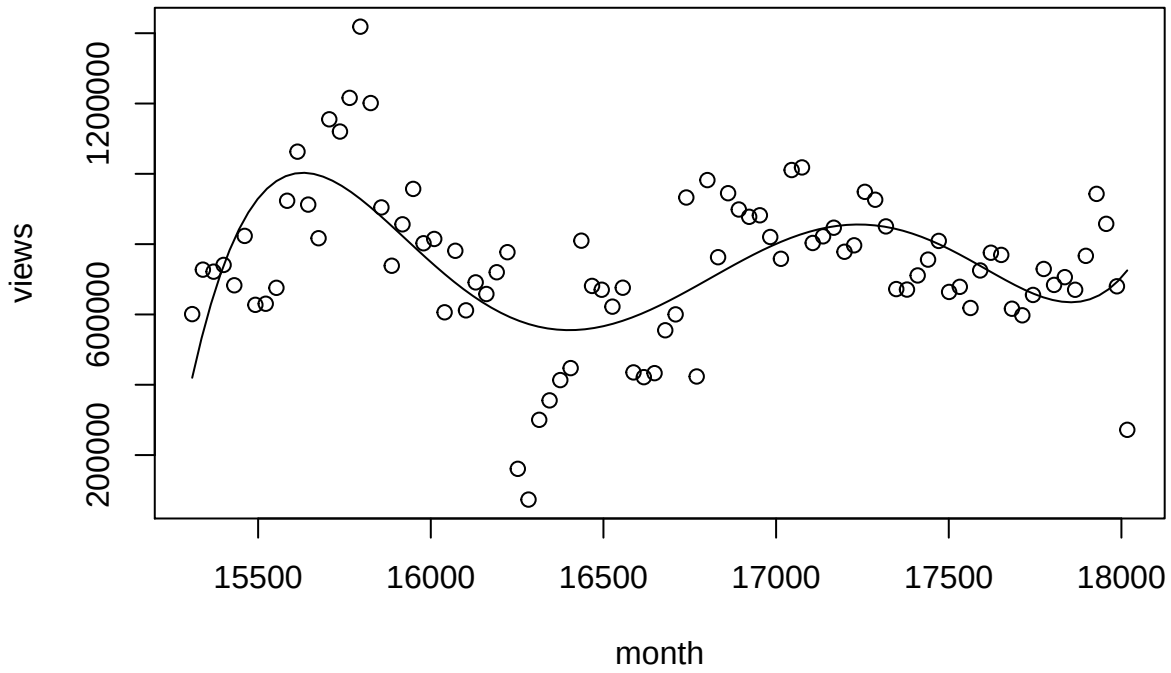
Then, our group had fit a polynomial surface determined by one or more numerical predictors, using local fitting. The graph also displays confidence interval around as gray. The graph below shows rise in trend from 2015 January to July 2016. The graph trend again drops from July 2016 to Jan 2019. The views counts are similar at the beginning of 2016 and end of plot at 2019. This shows that there has been a "trend reovery," 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.

```
monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize(views=sum(views))
monthly_agg$surveillence <- 'before'
monthly_agg$surveillence[monthly_agg$month >= '2013-06-01'] <- 'after'
model <- lm(views ~ month + surveillence + month*surveillence, data = monthly_agg)
monthly_agg$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)
```



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

model <- lm(views ~ month + surveillence + month*surveillence, 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)
```



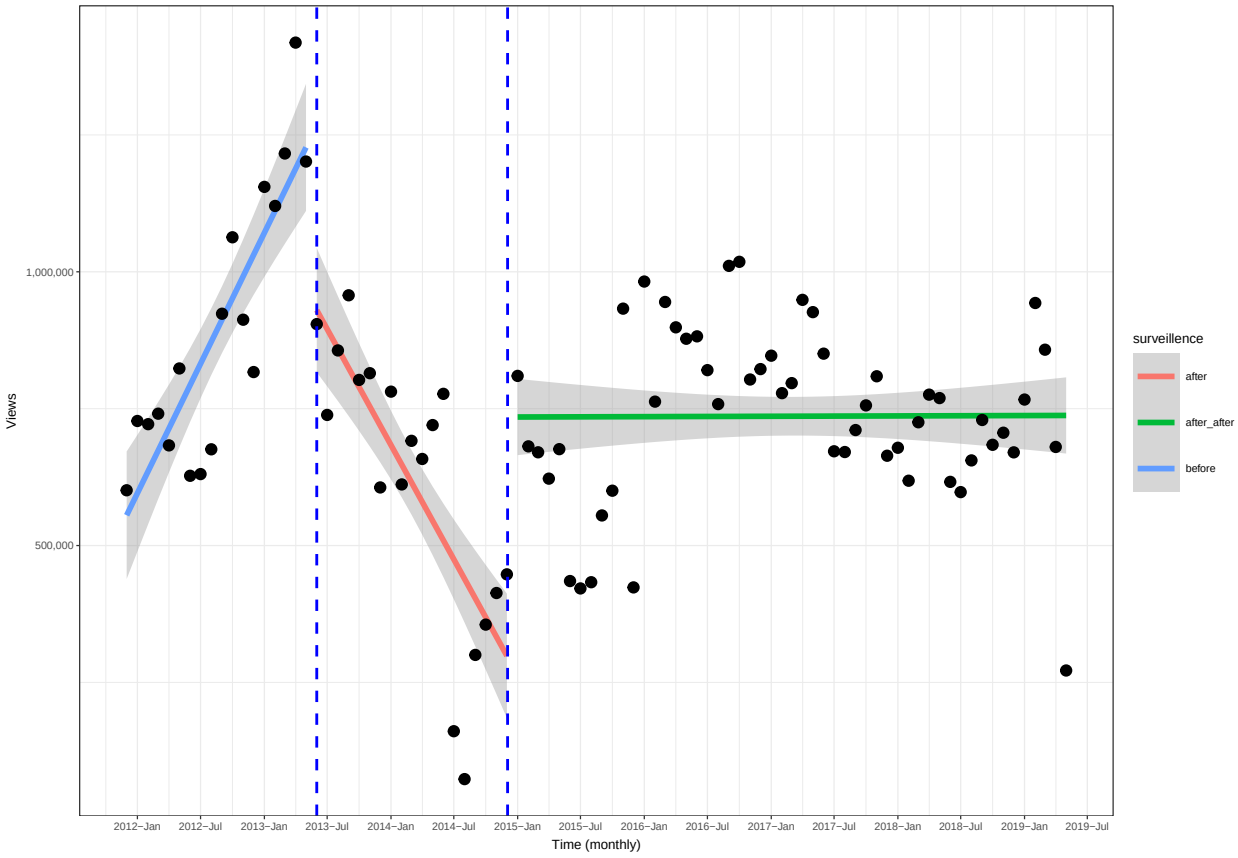
Then, our group had went further and fit a polynomial graph of degree 5 instead of 4. This may be overfitting, but wanted to analyze the graph to full extent. This graph also shows that there has been decrease in trend from 2018 to 2019, but end of the trend is actually increasing. This fluctuating trend 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$surveillence <- 'before'
monthly_agg$surveillence[monthly_agg$month >= '2013-06-01'] <- 'after'
monthly_agg$surveillence[monthly_agg$month >= '2014-12-3'] <- 'after_after'
model <- lm(views ~ month + surveillence + month*surveillence, data = monthly_agg)
summary(model)
```

```
##
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     data = monthly_agg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -465805  -71267   16265   86126  282306
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.920e+07  3.472e+06   5.529 3.56e-07 ***
## month                         -1.152e+03  2.153e+02  -5.353 7.40e-07 ***
## surveillenceafter_after       -1.849e+07  3.562e+06  -5.192 1.43e-06 ***
## surveillencebefore            -3.854e+07  5.025e+06  -7.670 2.77e-11 ***
## month:surveillenceafter_after  1.154e+03  2.201e+02   5.242 1.16e-06 ***
## month:surveillencebefore       2.452e+03  3.174e+02   7.724 2.16e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 156300 on 84 degrees of freedom
## Multiple R-squared:  0.5146, Adjusted R-squared:  0.4857
## F-statistic: 17.81 on 5 and 84 DF,  p-value: 5.357e-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 = prediction)) +
  geom_smooth(aes(ymin = LoCI,
                  ymax = HiCI,
                  color = surveillence),
              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)
```

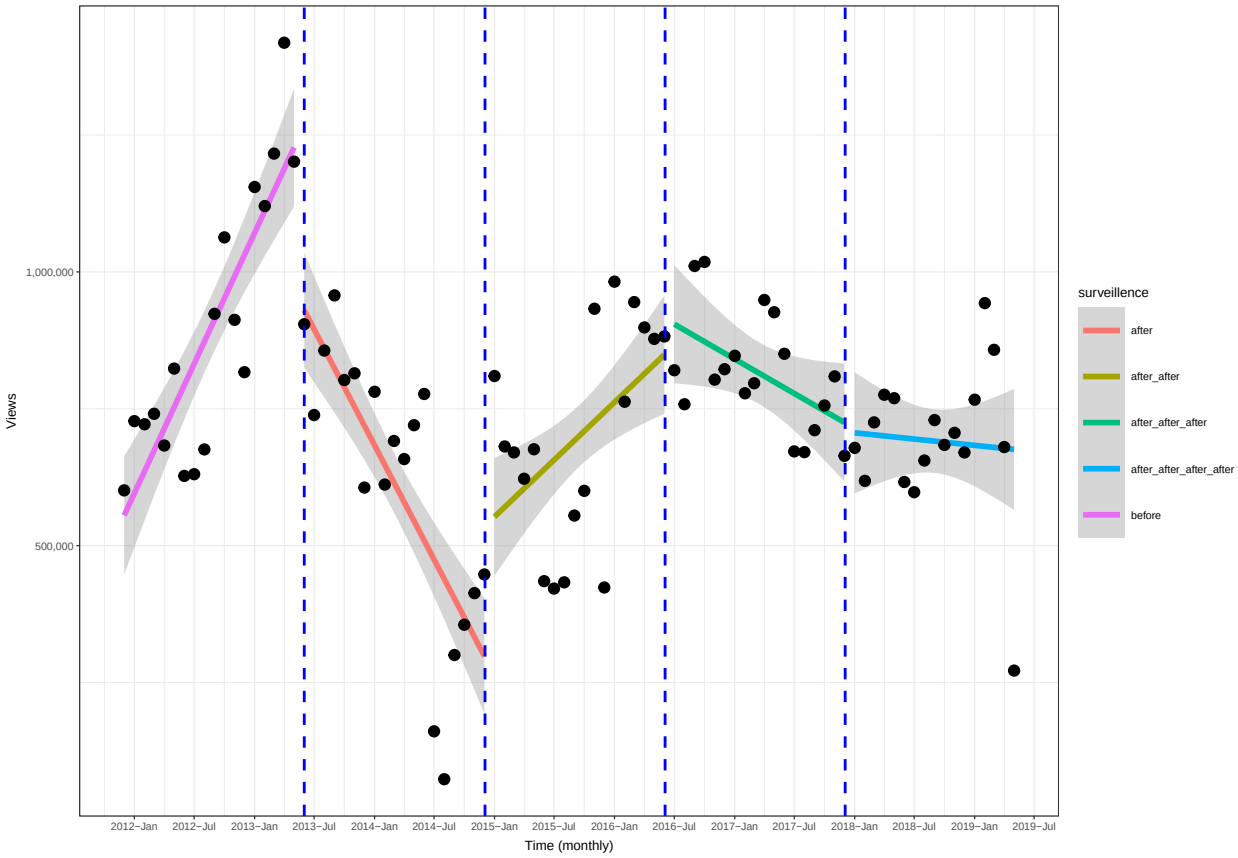


The graph below shows that there is stable trend after January 2015. This graph shows that view after 2015 is higher than from 2013 to 2015. This is only graph that may support author's claim that there had been a trend recovery, and trend stays constant. However, the data is not segmented in equal time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

```
monthly_agg <- terrorism_data_long %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize(views=sum(views))
monthly_agg$surveillence <- 'before'
monthly_agg$surveillence[monthly_agg$month >= '2013-06-01'] <- 'after'
monthly_agg$surveillence[monthly_agg$month >= '2014-12-3'] <- 'after_after'
monthly_agg$surveillence[monthly_agg$month >= '2016-06-3'] <- 'after_after_after'
monthly_agg$surveillence[monthly_agg$month >= '2017-12-3'] <- 'after_after_after_after'
model <- lm(views ~ month + surveillence + month*surveillence, data = monthly_agg)
summary(model)
```

```
##
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     data = monthly_agg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -403876  -80140   10295   83680  267670
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                             1.920e+07  3.220e+06   5.962
## month                                  -1.152e+03  1.996e+02  -5.771
## surveillenceafter_after                -2.806e+07  4.838e+06  -5.800
## surveillenceafter_after_after          -1.241e+07  4.931e+06  -2.516
## surveillenceafter_after_after_after    -1.739e+07  5.290e+06  -3.288
## surveillencebefore                     -3.854e+07  4.660e+06  -8.270
## month:surveillenceafter_after           1.725e+03  2.943e+02   5.860
## month:surveillenceafter_after_after     8.055e+02  2.945e+02   2.735
## month:surveillenceafter_after_after_after  1.089e+03  3.092e+02   3.523
## month:surveillencebefore                2.452e+03  2.944e+02   8.328
##                                        Pr(>|t|)
## (Intercept)                            6.42e-08 ***
## month                                  1.43e-07 ***
## surveillenceafter_after                1.27e-07 ***
## surveillenceafter_after_after          0.013862 *
## surveillenceafter_after_after_after    0.001501 **
## surveillencebefore                     2.39e-12 ***
## month:surveillenceafter_after          9.86e-08 ***
## month:surveillenceafter_after_after    0.007685 **
## month:surveillenceafter_after_after_after 0.000708 ***
## month:surveillencebefore               1.84e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 145000 on 80 degrees of freedom
## Multiple R-squared:  0.6024, Adjusted R-squared:  0.5576
## F-statistic: 13.46 on 9 and 80 DF,  p-value: 7.218e-13
```

```
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 = surveillence),
              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('2017-12-3'), 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)
```



Now, our group had seperated 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 Keyword-level Analysis

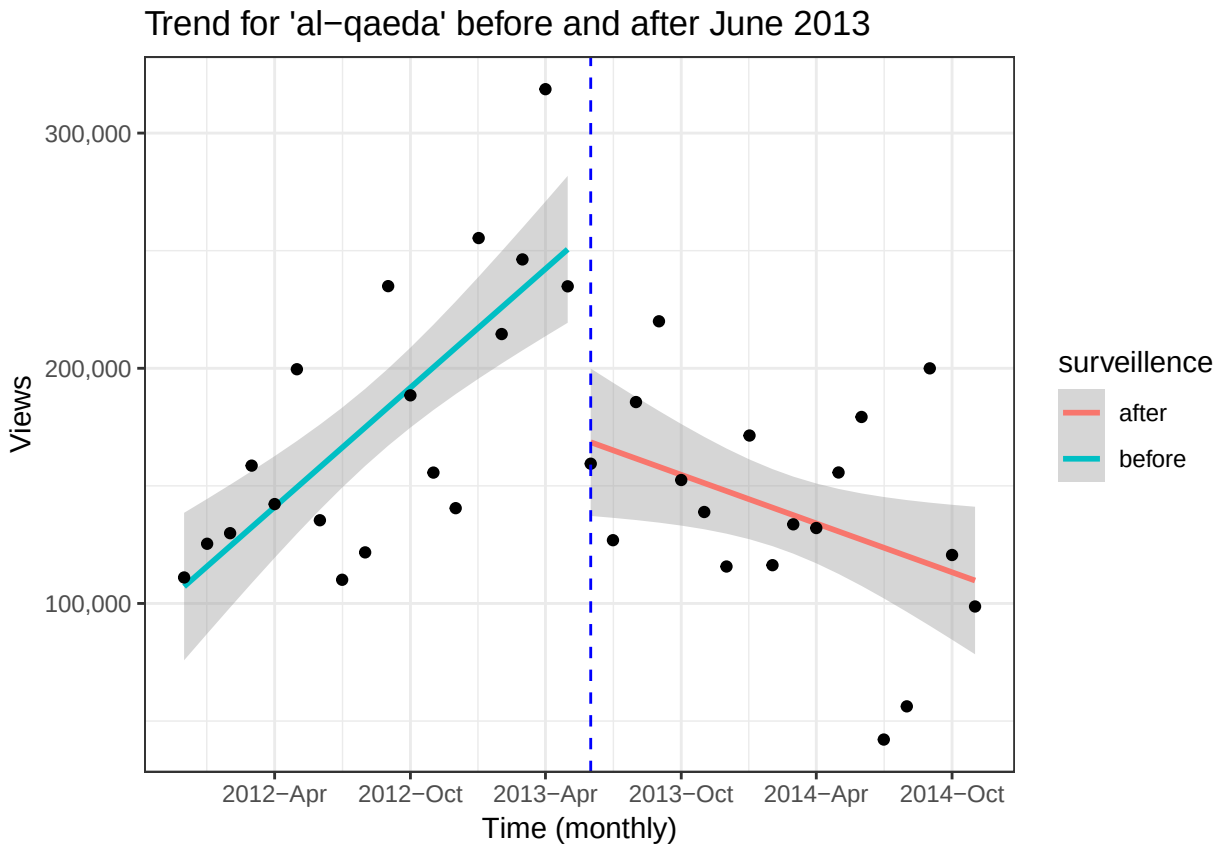
In this section, we will explore further on the trend of terrorism-related topics by looking into the trend of each article. The first part is visualizing some sample topics and the second part is quantifying the difference between the interaction model and the null model which use the trend of data before June 2013 to predict the views after the incident.

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

model <- lm(views ~ month + surveillence + month*surveillence, 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 = surveillence),
              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')
```

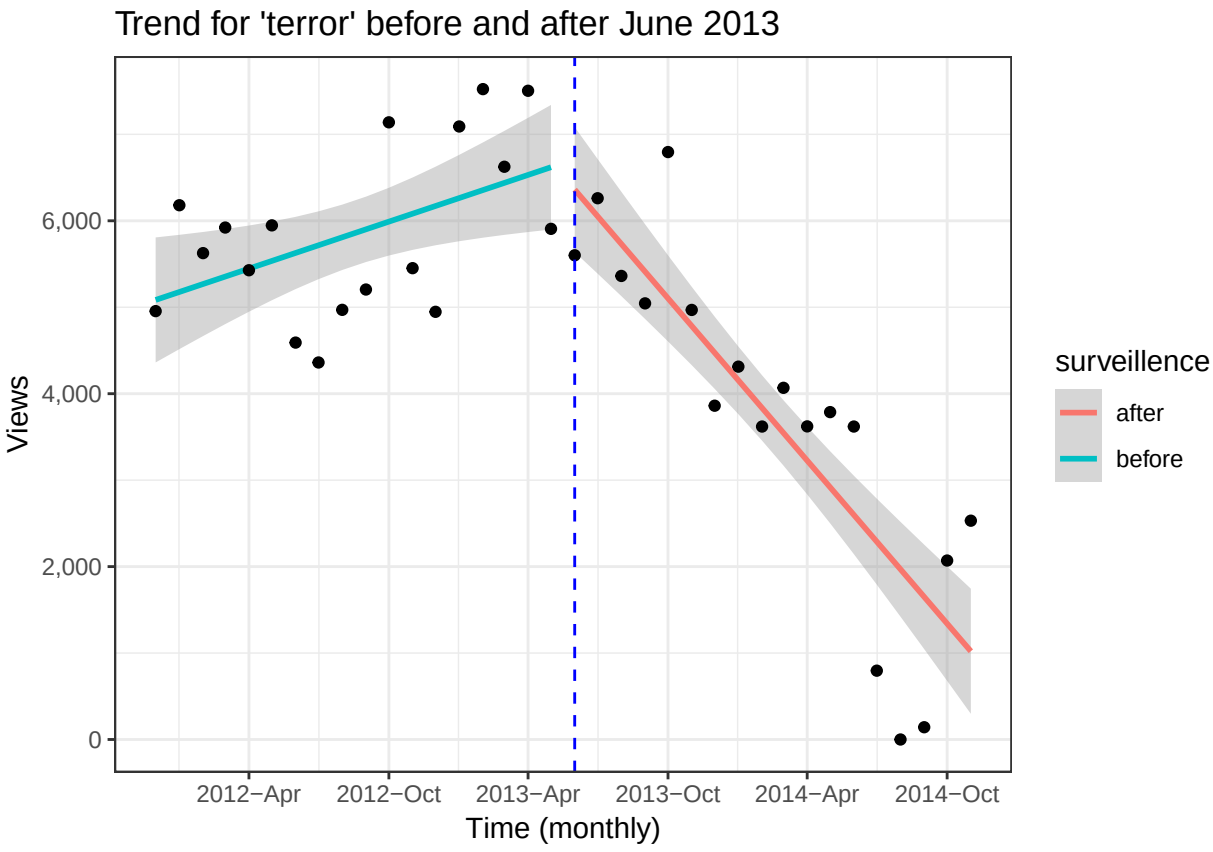
```
##
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -81582 -23037  -2093  25332  83296
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.969e+06  1.013e+06   1.943 0.060805 .
## month                     -1.135e+02  6.285e+01  -1.806 0.080314 .
## surveillencebefore        -6.107e+06  1.408e+06  -4.338 0.000134 ***
## month:surveillencebefore   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
```




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

```
##
```

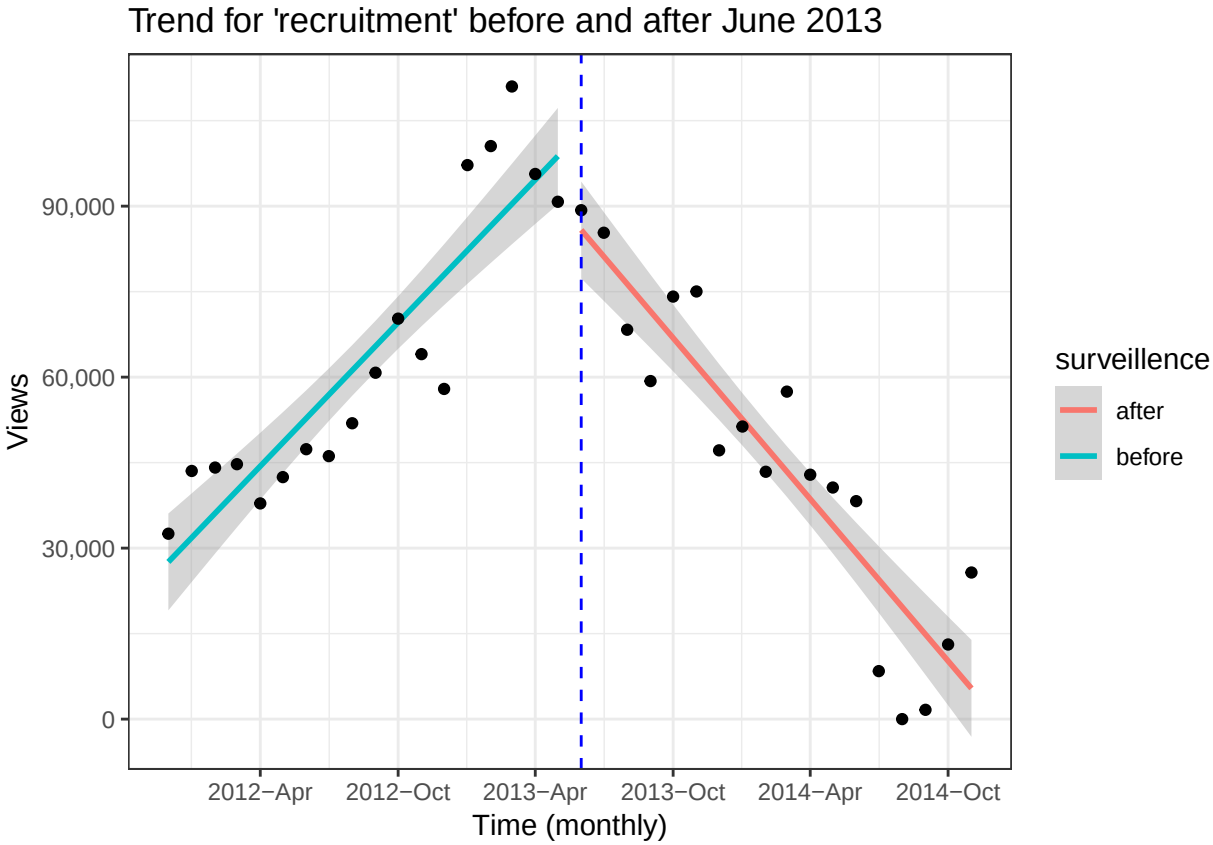
```
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1969.2  -700.2   172.0   754.2  1692.1
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.698e+05  2.337e+04   7.268 2.94e-08 ***
## month                     -1.031e+01  1.450e+00  -7.110 4.56e-08 ***
## surveillencebefore        -2.102e+05  3.247e+04  -6.473 2.78e-07 ***
## month:surveillencebefore   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
```




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

```
##
## Call:
## lm(formula = views ~ month + surveillence + month * surveillence,
##     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 ***
## surveillencebefore        -4.629e+06  3.815e+05 -12.132 1.64e-13 ***
## month:surveillencebefore   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
```



We can see from the plots that all the articles have the shift in trend. One of them, al-qaeda, has the significant shift in level. However, the coefficients for all articles are significant. Therefore, it is difficult to tell the different by just looking at coefficients and their significance. Moreover, it would not be scalelable if we have to visually analyze all the plots. We come up with a way to quantify the change in the following section.

3.2.2 Quantifying difference between models

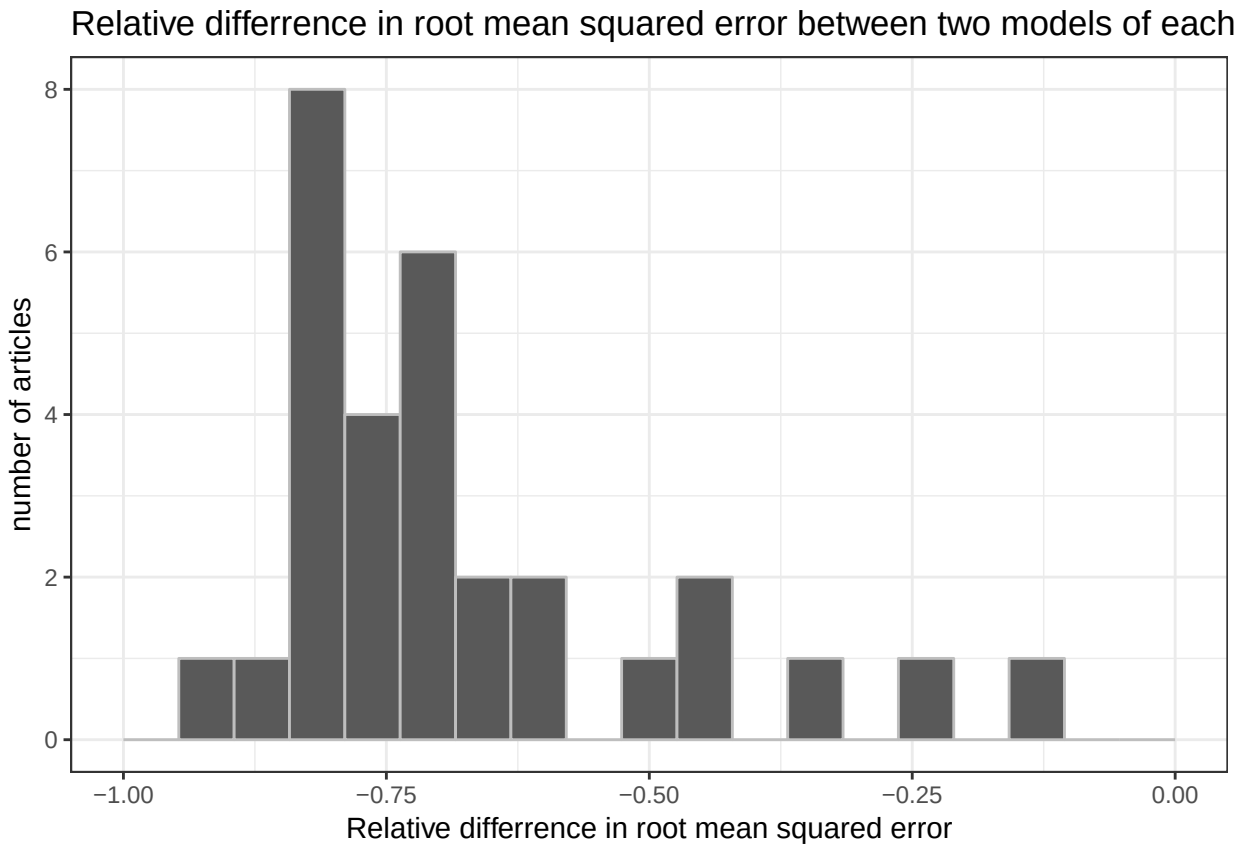
There are two quantities that we are interested in: relative differrence in root mean squared error and average of relative change in prediction.

Relative differrence in root mean squared error gives us the sense how better the model with interaction predicts the trend afterward than the model without. We use the relative term because the number of views for each topic is not equal.

Average of relative change in prediction tells us about the change in trend whether it is higher or lower and also give us some idea about magnitude of changes.

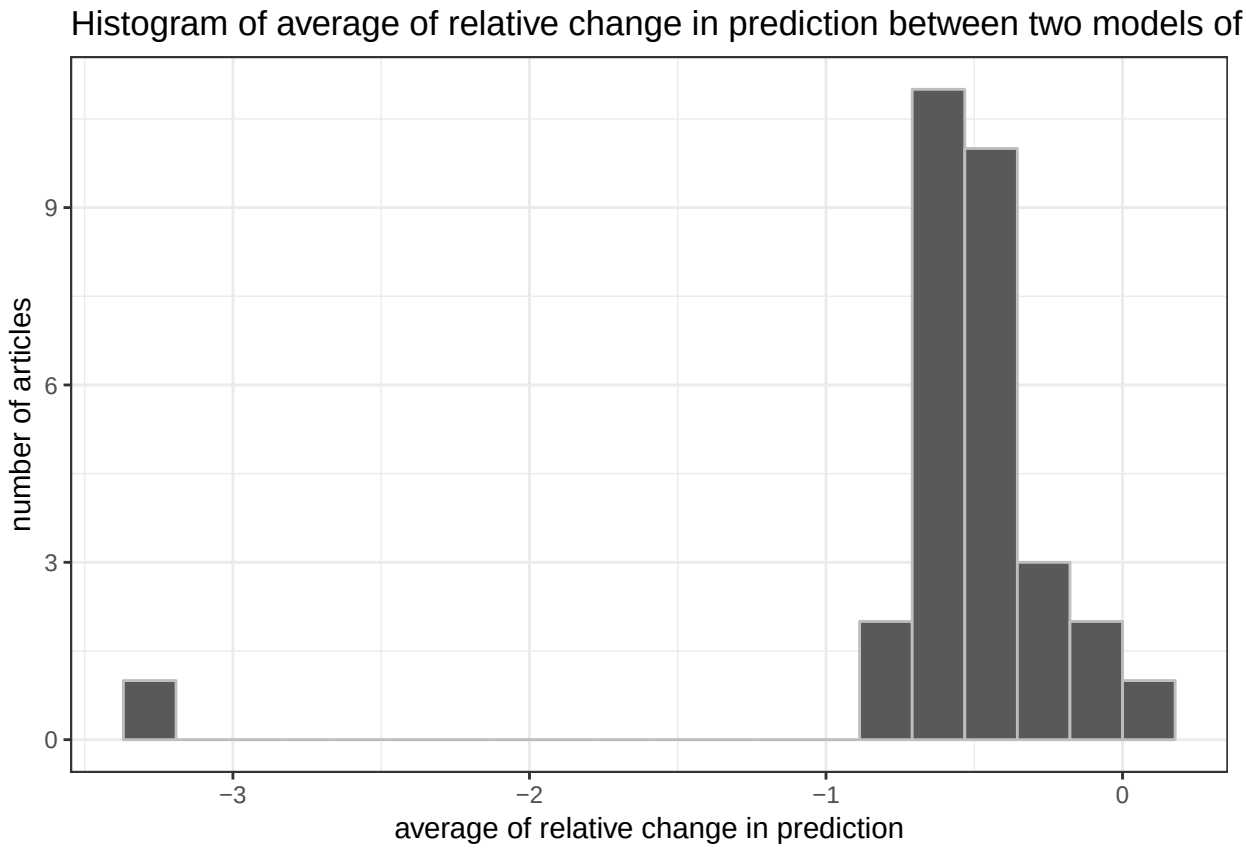
```
model_diff <- function(input_df, article_name){

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

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

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

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

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

null_pred <- predict(model2, df %>% filter(surveillence != 'before'))
null_rmse <- sqrt(mean((null_pred - df$views[df$surveillence != 'before'])^2))
return(c(mean(((int_pred - null_pred)/ null_pred)), (interaction_rmse - null_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 = rmse_difference)) +
  geom_histogram(boundary = 0, bins = 20, color = 'grey') +
  xlim(c(-1, 0)) +
  ylab("number of articles") +
  xlab("Relative differrence in root mean squared error") +
  ggtitle("Relative differrence in root mean squared error between two models of each article")
```



As expected, the model with interaction always performs better because it has an ability to fit two parts of the data seperately. What interesting here is that the improvement of RMSE is not that high. This is similar to the problem we discussed when we learn about regression that sometimes the results are significant, but the question is do we care about the change? As we see from the longer trend analysis that if we consider the longer period of the data, the trend might diminish.

```
ggplot(diff_results, aes(x = mean_difference)) +
  geom_histogram(boundary = 0, bins = 20, color = 'grey') +
  ylab("number of articles") +
  xlab("average of relative change in prediction") +
  ggtitle("Histogram of average of relative change in prediction between two models of each article")
```



The histogram of average of relative change in prediction tells us that most of the trends go down or have less slope after the incident. One of the topic actually has higher number of views. There is another topic that its view decreases much more than the other. What we can investigate further in the future is that we can include number of views for each topic in our analysis to take into account the weighted effect on the overall trend.

3.3 Time-series Analysis

In this part of the analysis, we wish to explore just how valid the claims of the paper are, that search trends have changed in magnitude and direction pre- and post-Snowden.

We begin our analysis by pulling data using the terrorism keyword list as specified in the paper. We sample the data from December 1, 2011 to December 30, 2015. We also examine trends from the ranges of December 1, 2008 to December 31, 2018, as well as December 1, 2007 to May 1, 2018.

```
require(wikipediatrend)
require(tidyverse)
require(lubridate)
require(astsa)
```

```
## Loading required package: astsa
```

```
require(scales)
```

The trend data from December 1, 2011 to December 31, 2015 is stored in wiki_monthly.

```
wiki_monthly <- terrorism_data_2005_present %>%
  filter(date >= '2011-12-01' & date <= '2015-12-31')
```

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

```
## # A tibble: 49 x 2
##    month       views
##    <date>      <dbl>
##  1 2011-12-01 600949
##  2 2012-01-01 727496
##  3 2012-02-01 721693
##  4 2012-03-01 740862
##  5 2012-04-01 682937
##  6 2012-05-01 823435
##  7 2012-06-01 627388
##  8 2012-07-01 630508
##  9 2012-08-01 675752
## 10 2012-09-01 923390
## # ... with 39 more rows
```

The trend data from December 1, 2008 to December 31, 2018 is stored in wiki_monthly_expanded.

```
wiki_monthly_expanded <- terrorism_data_2005_present %>%
  filter(date >= '2008-12-01' & date <= '2018-12-31')
wiki_monthly_expanded <- wiki_monthly_expanded %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize(views=sum(views))
wiki_monthly_expanded
```

```
## # A tibble: 121 x 2
##    month       views
##    <date>      <dbl>
##  1 2008-12-01 613577
##  2 2009-01-01 777986
##  3 2009-02-01 664745
##  4 2009-03-01 746472
##  5 2009-04-01 710235
##  6 2009-05-01 801382
##  7 2009-06-01 651064
##  8 2009-07-01 551122
##  9 2009-08-01 652483
## 10 2009-09-01 499376
## # ... with 111 more rows
```

The trend data from December 1, 2007 to May 1, 2019 is stored in wiki_montly_expanded_more.

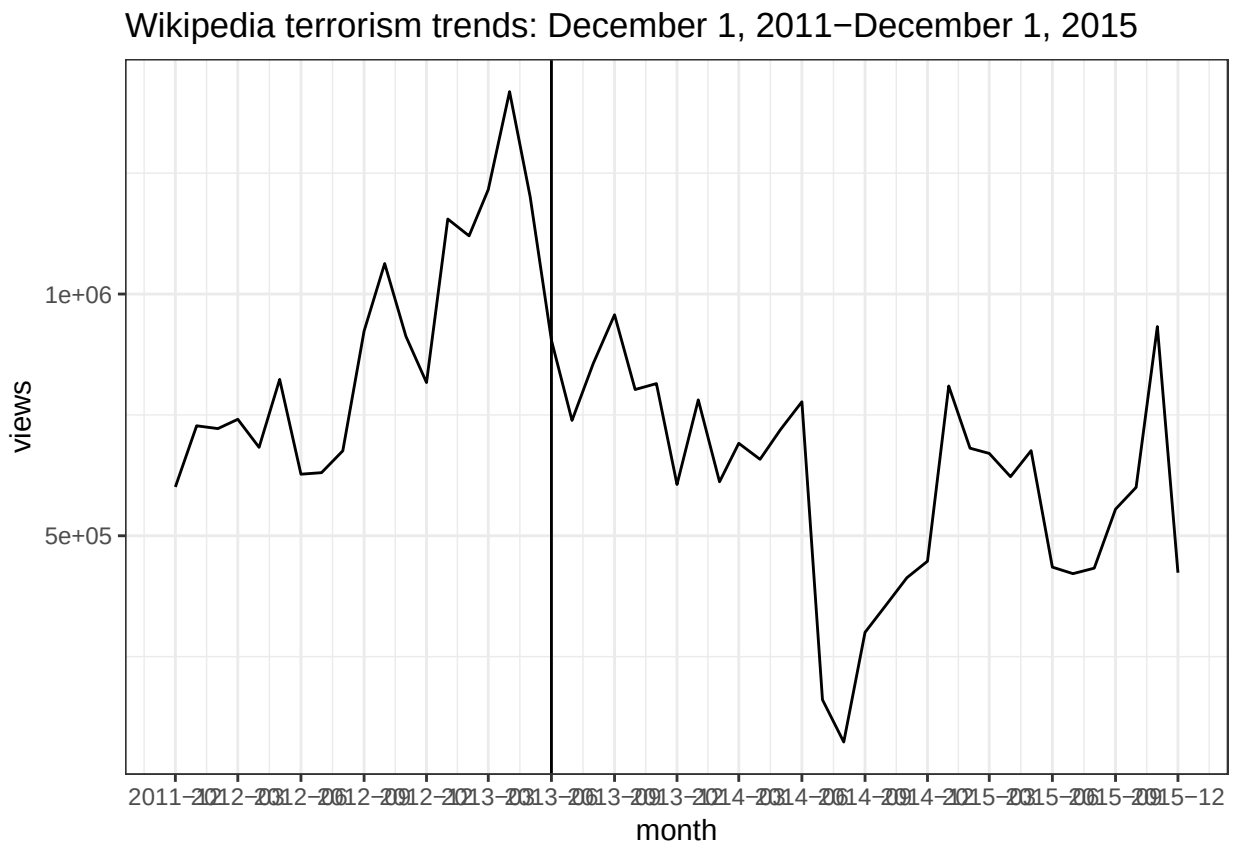
```
wiki_monthly_expanded_more <- terrorism_data_2005_present %>%
  filter(date >= '2007-12-01' & date <= '2019-05-01')
wiki_monthly_expanded_more <- wiki_monthly_expanded_more %>%
  group_by(month=floor_date(date, "month")) %>%
  summarize(views=sum(views))
wiki_monthly_expanded_more
```

```
## # A tibble: 138 x 2
##    month       views
##    <date>      <dbl>
```

```
##  1 2007-12-01 355180
##  2 2008-01-01 567114
##  3 2008-02-01 570456
##  4 2008-03-01 622434
##  5 2008-04-01 585878
##  6 2008-05-01 685887
##  7 2008-06-01 560223
##  8 2008-07-01 552874
##  9 2008-08-01 457203
## 10 2008-09-01 608728
## # ... with 128 more rows
```

When we look at the plot for wiki_monthly (December 1, 2011-December 31, 2015):

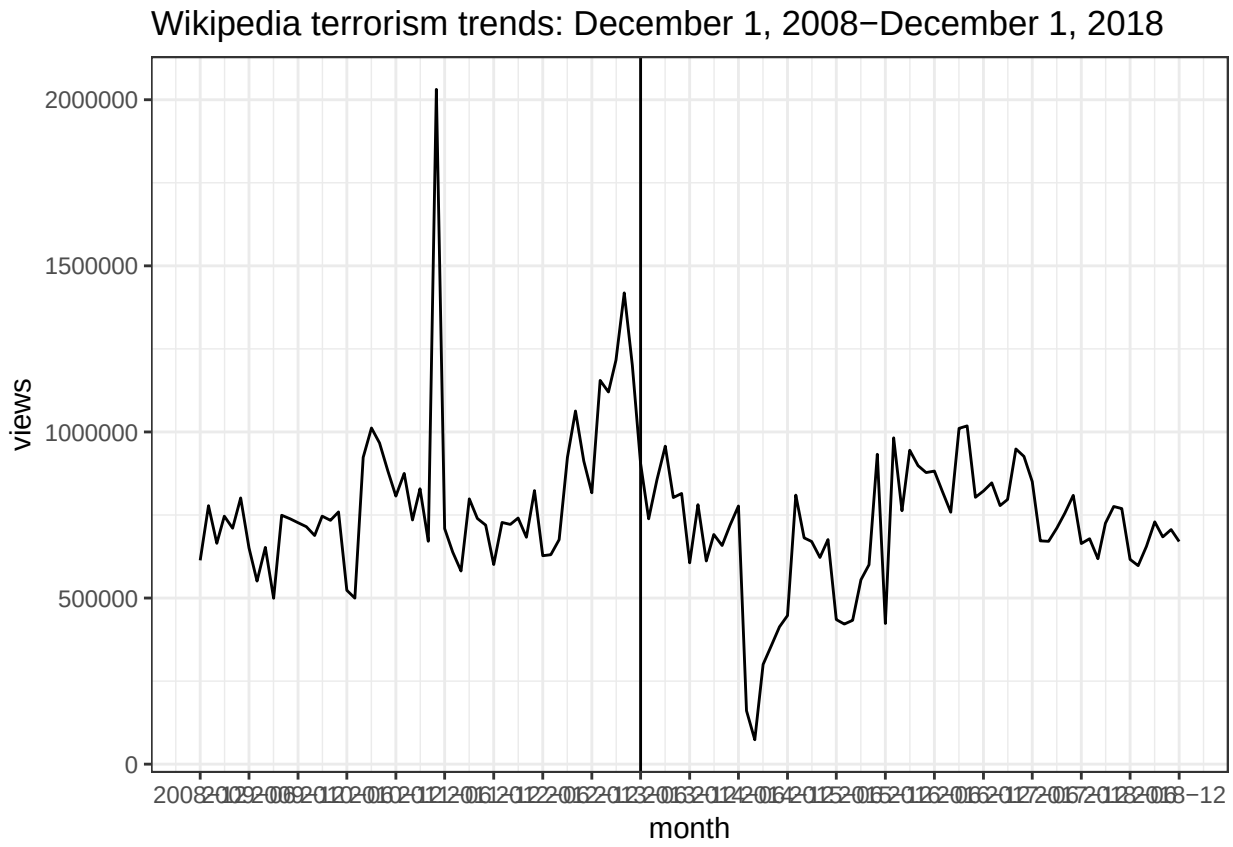
```
ggplot(wiki_monthly, aes(month, views)) +
  geom_line() +
  ggtitle('Wikipedia terrorism trends: December 1, 2011-December 1, 2015') +
  geom_vline(xintercept=as.numeric(as.Date('2013-06-01'))) +
  scale_x_date(labels=date_format('%Y-%m'),
               breaks=wiki_monthly$month[seq(1, length(wiki_monthly$month),
                                             by=3)])
```



there does appear to be a shift pre- and post- Snowden. However, based on some of the data post-Snowden, we need to examine further. Are these trends outliers, or is there credible evidence of significant drops in searches for these terms? We chose to sample over a larger time frame (the wiki_monthly_expanded data from December 1, 2008 to December 31, 2018) to see if this was the case.

```
ggplot(wiki_monthly_expanded, aes(month, views)) +
  geom_line() +
```

```
  ggtitle('Wikipedia terrorism trends: December 1, 2008-December 1, 2018') +
  geom_vline(xintercept=as.numeric(as.Date('2013-06-01'))) +
  scale_x_date(labels=date_format('%Y-%m'),
               breaks=wiki_monthly_expanded$month[seq(1, length(wiki_monthly_expanded$month),
                                                      by=6)])
```

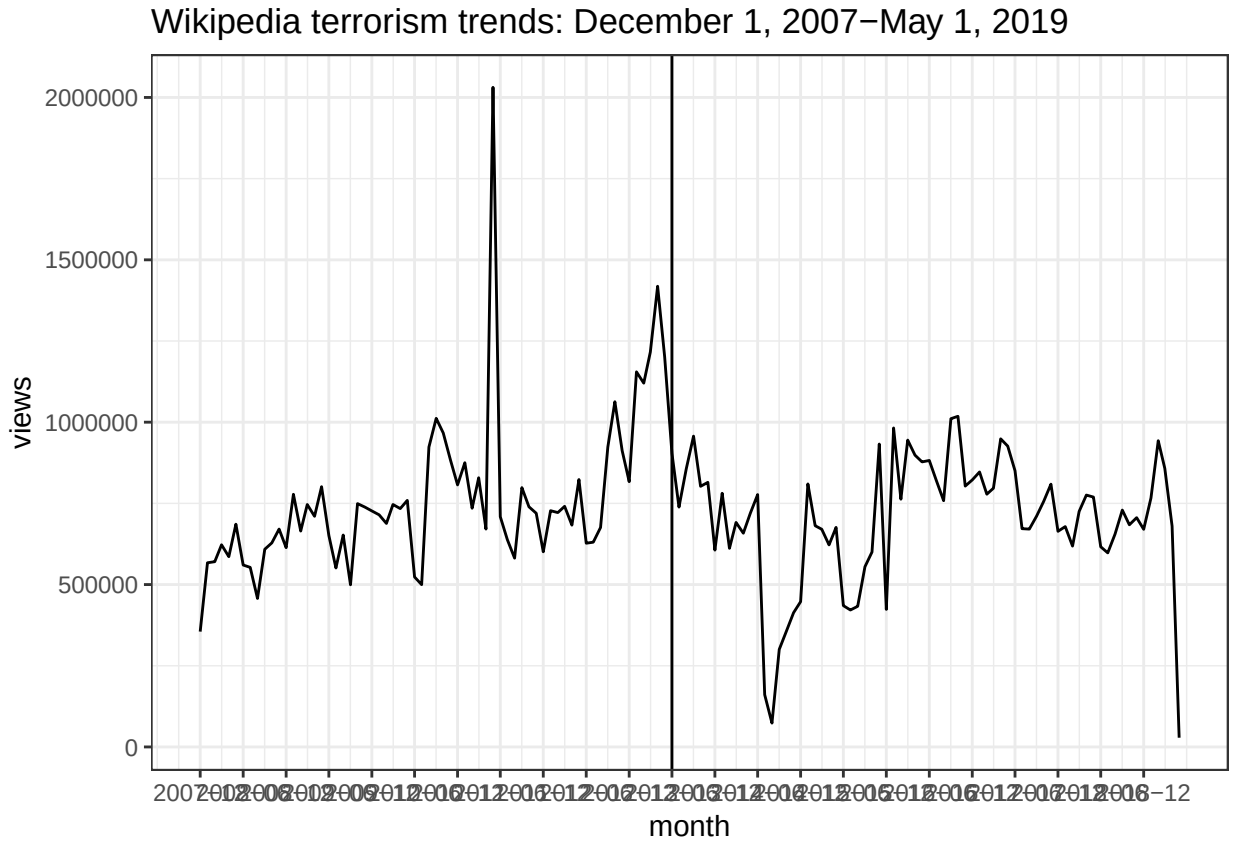


When we examine the above plot, now it becomes far less obvious that the trend the paper claims exists. For one, it appears that searches, while fluctuating, are still centered around the same mean. In time-series analysis, we call this a stationary distribution (not perfectly, but about). It is true that search terms did increase from 2011 to 2013 and then decrease from 2013-2014, however, it does not appear that this shift is monumentous enough to warrant the paper's claim. The high point right before Snowden seems like a high outlier, and a few points in 2014 look like low outliers.

All in all, the mean seems fairly centered, with a few outlier points that do not disrupt the stationary distriubtion.

We see further evidence of this stationary distribution when we sample between December 1, 2007 and May 1, 2019 (the wiki_monthly_expanded_more dataset):

```
ggplot(wiki_monthly_expanded_more, aes(month, views)) +
  geom_line() +
  ggtitle('Wikipedia terrorism trends: December 1, 2007-May 1, 2019') +
  geom_vline(xintercept=as.numeric(as.Date('2013-06-01'))) +
  scale_x_date(labels=date_format('%Y-%m'),
               breaks=wiki_monthly_expanded_more$month[seq(1, length(wiki_monthly_expanded_more$month),
                                                      by=6)])
```



Because this graph was drawn at the very beginning of May, there are relatively few search terms for the terrorism terms listed. Otherwise, we see mostly the same stationary distribution.

An interesting extension we looked at was to see if we could model this data with a time series model to see if we can capture the patterns in this time series. AR models work very well as a simple example because, in a nutshell, they use the nth previous time series data points to predict the future (and take into account an error term known as white noise). More complicated models such as ARMA, ARIMA, GARCH, and e-GARCH exist, however, for the purposes of exploration, we will stick to the baseline which has been shown to work well (like linear regression for regression analysis).

The reason we chose to do a time series analysis is because the paper does not take a very strong time series based approach to their analysis. They used what is called an interrupted time series approach, which is another way of saying "we looked at the data before and after a certain event to look for changes in patterns over time". However, the paper did not take into account other factors which may have caused this trend change, such as seasonality.

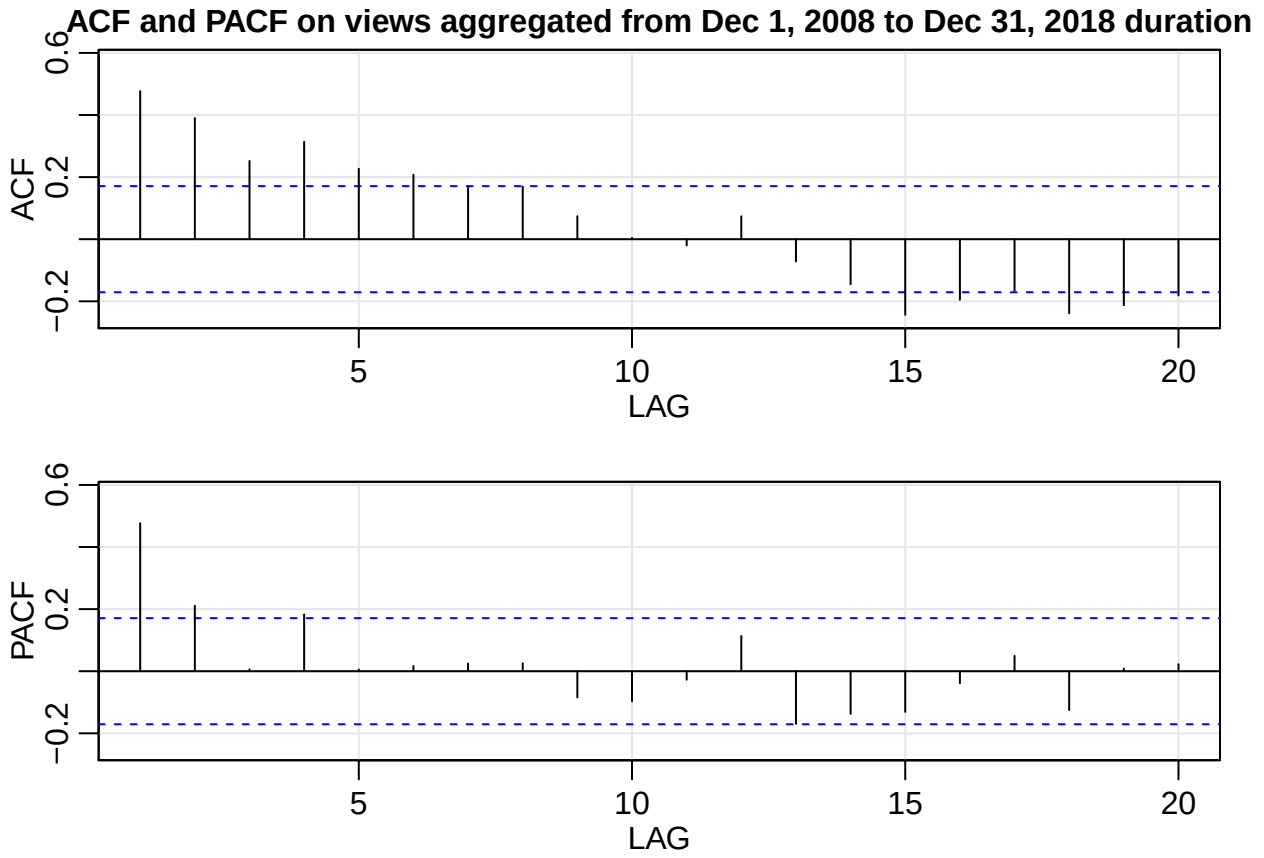
We use the wiki_monthly_expanded_more (December 1, 2007 to May 1, 2019) dataset, however, we first drop the datapoint for May 1, 2019, as only that day's data is being used to model the total number of views for May 2019. We attempt to model a simple AR(1) model on the data to start off (where the 1 means a 1st order regression):

```
time_series_data <- wiki_monthly_expanded_more %>% filter(month < as.Date('2019-05-01'))
time_series_data
```

```
## # A tibble: 137 x 2
##    month       views
##    <date>      <dbl>
##  1 2007-12-01 355180
```

```
##  2 2008-01-01 567114
##  3 2008-02-01 570456
##  4 2008-03-01 622434
##  5 2008-04-01 585878
##  6 2008-05-01 685887
##  7 2008-06-01 560223
##  8 2008-07-01 552874
##  9 2008-08-01 457203
## 10 2008-09-01 608728
## # ... with 127 more rows
```

```
acf2(time_series_data$views, max.lag=20,
     main='ACF and PACF on views aggregated from Dec 1, 2008 to Dec 31, 2018 duration')
```



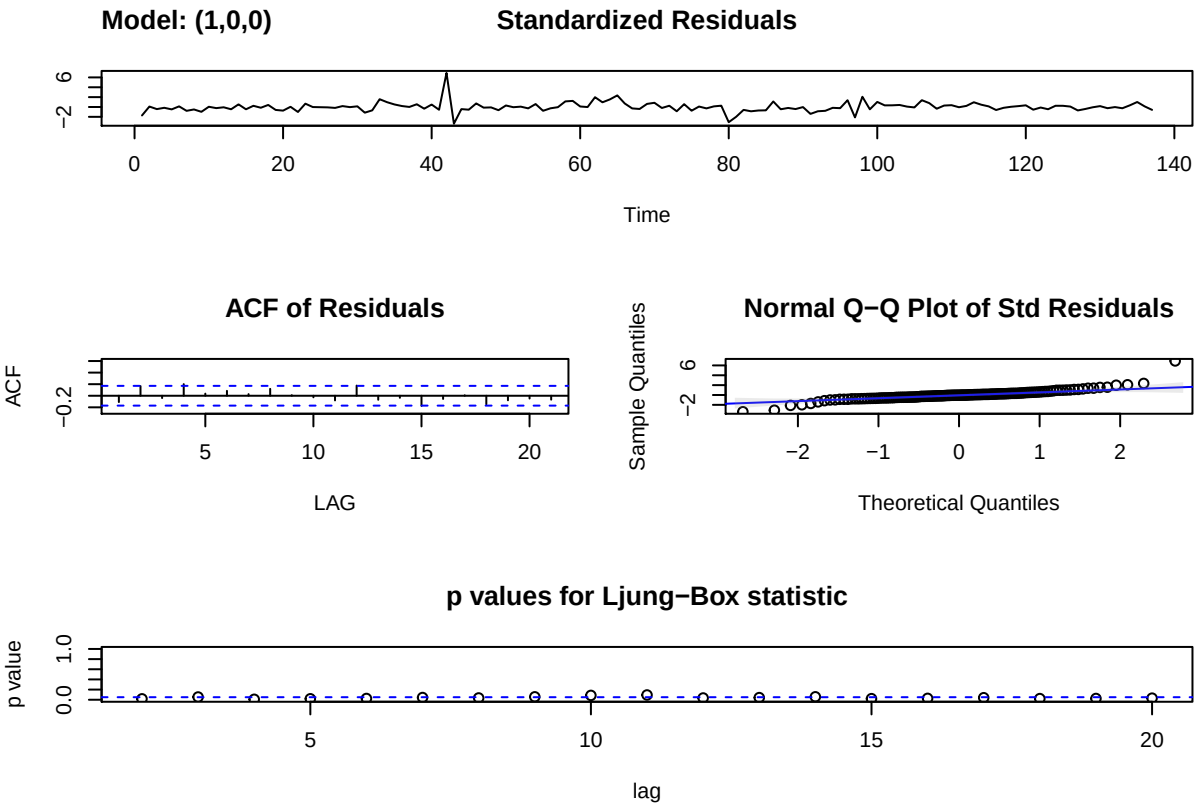

```
##          ACF  PACF
##  [1,]  0.48  0.48
##  [2,]  0.39  0.21
##  [3,]  0.25  0.01
##  [4,]  0.31  0.18
##  [5,]  0.23  0.01
##  [6,]  0.21  0.02
##  [7,]  0.17  0.02
##  [8,]  0.17  0.03
##  [9,]  0.07 -0.08
## [10,]  0.00 -0.10
## [11,] -0.02 -0.03
## [12,]  0.07  0.11
## [13,] -0.07 -0.17
```

```
## [14,] -0.15 -0.14
## [15,] -0.24 -0.13
## [16,] -0.20 -0.04
## [17,] -0.17  0.05
## [18,] -0.24 -0.13
## [19,] -0.21  0.01
## [20,] -0.18  0.02
```

Because the autocorrelation plot, or ACF, plot seems to monotonically decrease for the most part, an AR(1) model should work decently.

```
ts_fit <- sarima(time_series_data$views, p=1, d=0, q=0)
```

```
## initial  value 12.292051
## iter   2 value 12.159712
## iter   3 value 12.159630
## iter   4 value 12.159617
## iter   5 value 12.159608
## iter   6 value 12.159607
## iter   6 value 12.159607
## final  value 12.159607
## converged
## initial  value 12.168337
## iter   2 value 12.168200
## iter   3 value 12.168092
## iter   4 value 12.168078
## iter   5 value 12.168076
## iter   5 value 12.168076
## iter   5 value 12.168076
## final  value 12.168076
## converged
```




```
ts_fit
```

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      xmean
##       0.4842  732860.19
## s.e.  0.0752   31654.53
##
## sigma^2 estimated as 3.7e+10:  log likelihood = -1861.42,  aic = 3728.84
##
## $degrees_of_freedom
## [1] 135
##
## $ttable
##          Estimate         SE t.value p.value
## ar1        0.4842     0.0752  6.4377       0
## xmean 732860.1910 31654.5330 23.1518       0
##
## $AIC
## [1] 25.3634
##
## $AICc
## [1] 25.37931
```

```
##
## $BIC
## [1] 24.40603
```

From the above plot, we notice a few problems with our AR(1) fit.

ACF of residuals: not inside the confidence interval for all lags Normal Q-Q plot: not normal (this is an assumption the model makes) *Ljung-Box statistic: low p-values means we reject the null hypothesis that there are no more correlations we need to take into account. This means that our AR(1) model is too simple.

The easy fix is to make a more complicated model. However, before we fit the model, we should examine the time series for outliers. We use the tsoutliers function to identify and replace outliers in our dataset:

```
require(tsoutliers)
```

```
## Loading required package: tsoutliers
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

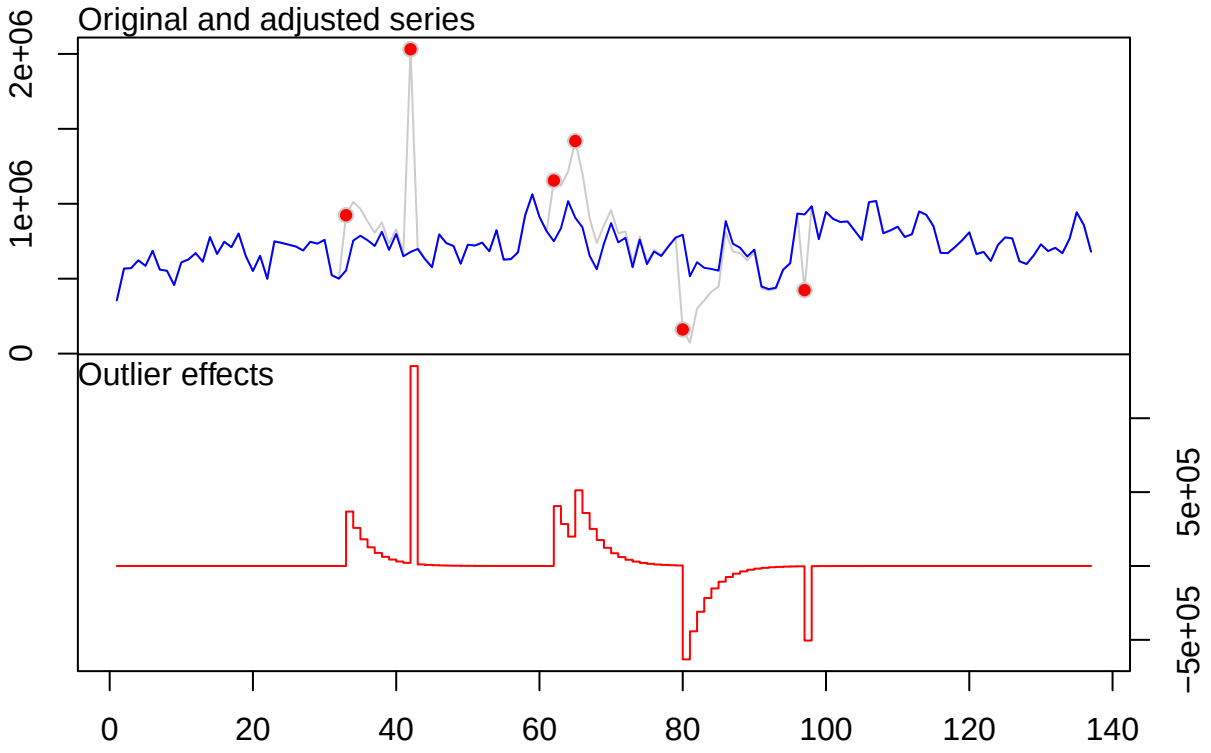
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
outlier_ts <- ts(time_series_data$views,frequency=1)
data_outliers <- tso(outlier_ts)
data_outliers
```

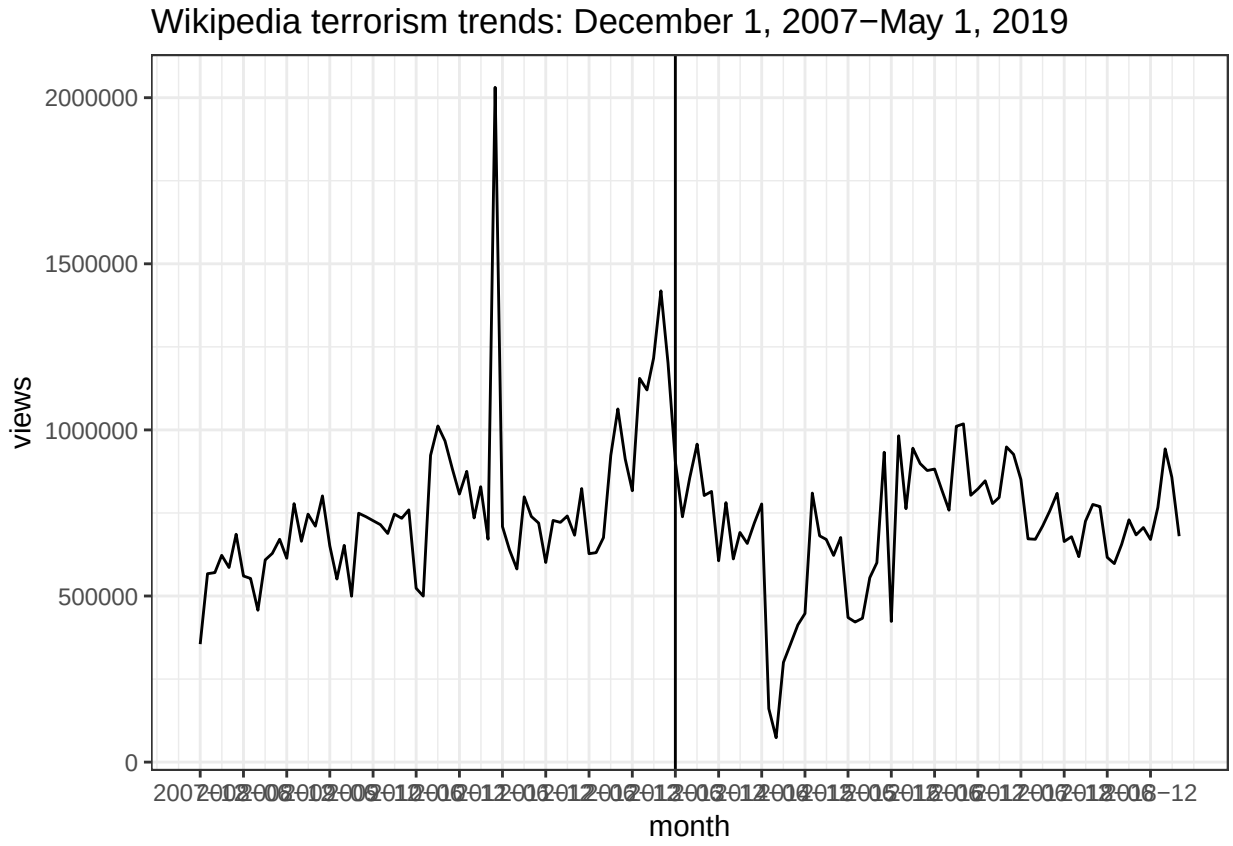
```
## Series: outlier_ts
## Regression with ARIMA(1,1,1) errors
##
## Coefficients:
##          ar1      ma1      TC33        AO42      TC62      TC65       TC80
##       0.5628  -0.9527  368263.4  1338245.65  405322.0  373205.9  -634685.7
## s.e.  0.0994   0.0482  104283.9    90577.68  107711.1  105433.1   106919.0
##            AO97
##       -503383.1
## s.e.    91773.4
##
## sigma^2 estimated as 1.198e+10:  log likelihood=-1767.49
## AIC=3552.98   AICc=3554.41   BIC=3579.19
##
## Outliers:
##   type ind time coefhat  tstat
## 1   TC  33   33  368263  3.531
## 2   AO  42   42 1338246 14.775
## 3   TC  62   62  405322  3.763
## 4   TC  65   65  373206  3.540
## 5   TC  80   80 -634686 -5.936
## 6   AO  97   97 -503383 -5.485
```

```
plot(data_outliers)
```



For comparison, we plot the original time series data below:

```
ggplot(time_series_data, aes(month, views)) +
  geom_line() +
  ggtitle('Wikipedia terrorism trends: December 1, 2007-May 1, 2019') +
  geom_vline(xintercept=as.numeric(as.Date('2013-06-01'))) +
  scale_x_date(labels=date_format('%Y-%m'),
               breaks=time_series_data$month[seq(1, length(time_series_data$month),
                                                 by=6)])
```



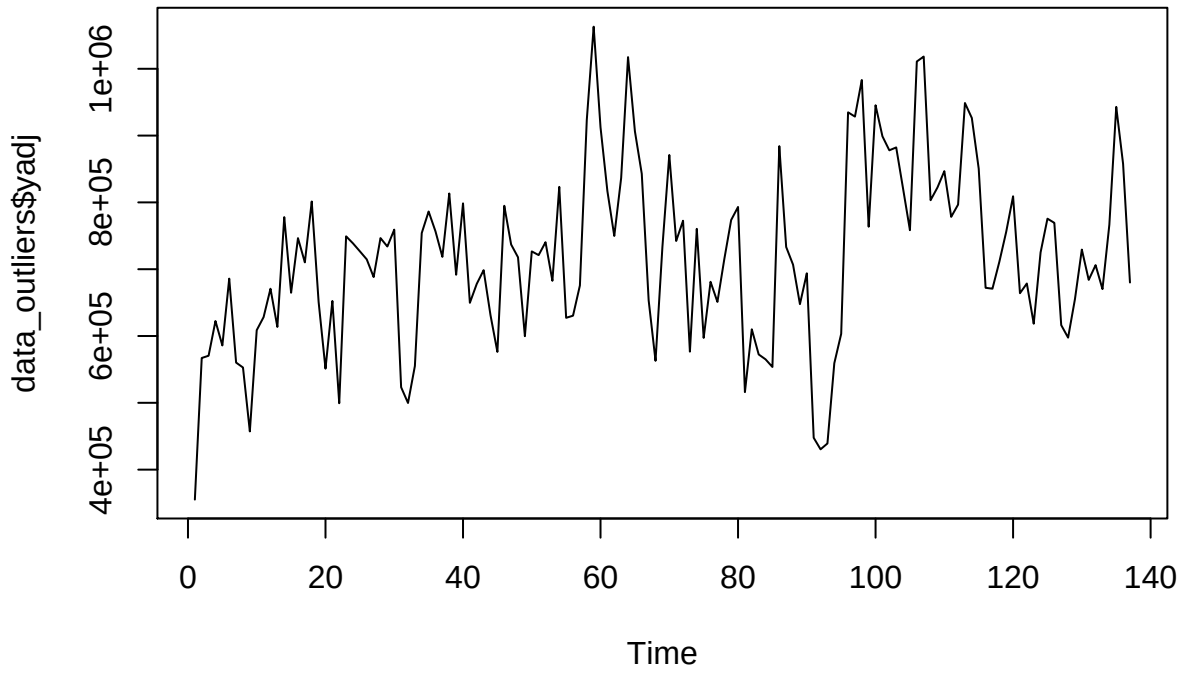
This outlier plot shows that the claimed chilling effect seems to have been influenced by an anomalous increase in terrorism searches in the months just preceding pre-Snowden. This presents a problem for the chilling effect claim. It is quite possible that there still would have been a decrease in terrorism related searches. However, the claim that there was a significant increase in search volume in the months immediately prior to pre-Snowden is weakened.

It should also be noticed that it is unclear whether the sudden drop off in search volume from July 2014-December 2014 is actually caused by the chilling effect. While these are identified as low outliers, there is not the gradual decrease in search volume as specified by the paper. In the immediate months following post-Snowden, search terms simply regressed back to the mean and stayed there until the sudden nosedive in July 2014.

At the same time, the time series analysis also shows that it is difficult to model views over time on a website due to frequent fluctuations which may result in outliers in the data. We choose to take the stance that, even if there was a chilling effect, it was potentially biased by points immediately pre-Snowden and rebounded quite well after an anomalous drop in terrorism search volume from July 1, 2014 to the paper's right endpoint on November 30, 2014. Thus, we use the outlier corrected time-series data to model continued trends.

When we see the plot of the outlier corrected time-series data, we now see that the data is even more stationary, which is to be expected.

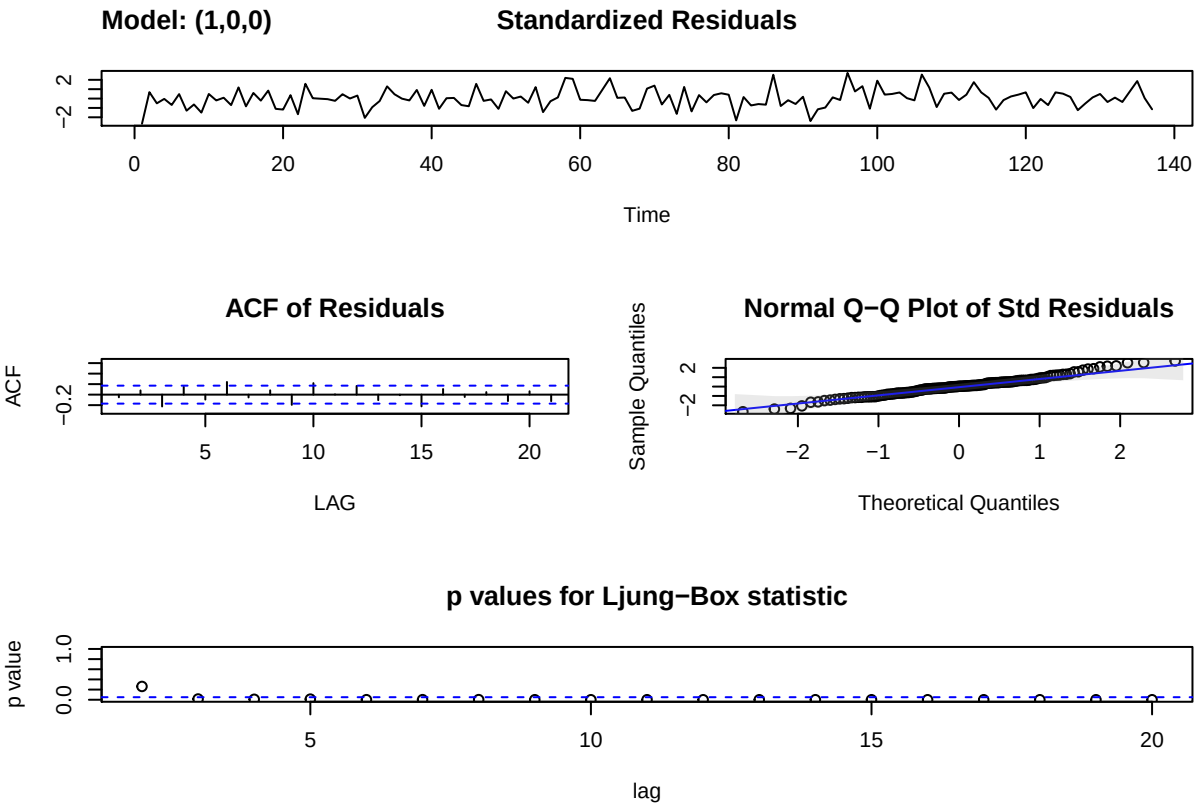
```
plot(data_outliers$yadj)
```



We now attempt to plot the same AR(1) model to the outlier-adjusted values:

```
ts_fit_adj <- sarima(data_outliers$yadj, p=1, d=0, q=0)
```

```
## initial  value 11.762338
## iter   2 value 11.543145
## iter   3 value 11.542401
## iter   4 value 11.542141
## iter   5 value 11.542116
## iter   6 value 11.542081
## iter   6 value 11.542081
## final  value 11.542081
## converged
## initial  value 11.570162
## iter   2 value 11.569285
## iter   3 value 11.568932
## iter   4 value 11.568816
## iter   5 value 11.568735
## iter   6 value 11.568722
## iter   7 value 11.568721
## iter   8 value 11.568721
## iter   8 value 11.568721
## iter   8 value 11.568721
## final  value 11.568721
## converged
```




```
ts_fit_adj
```

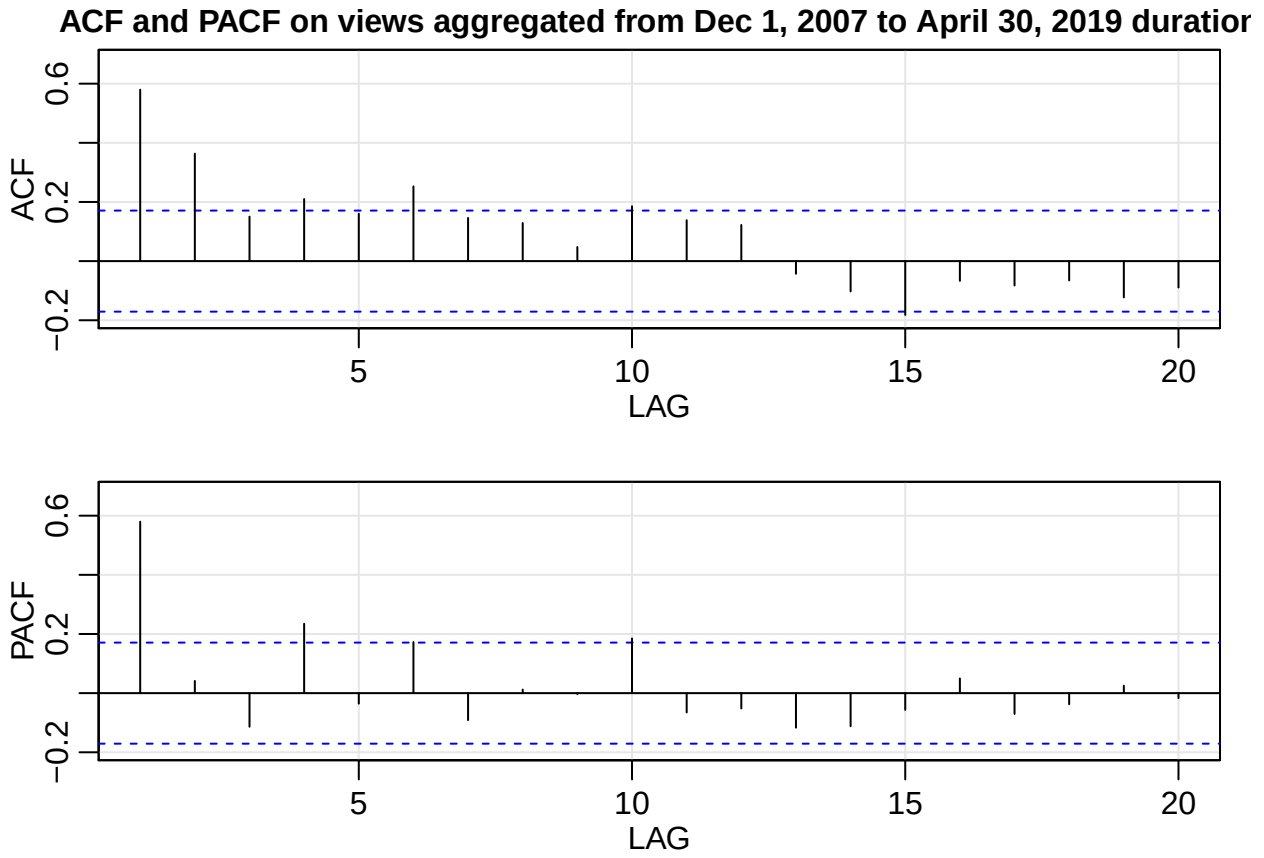
```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      xmean
##       0.6095  712768.31
## s.e.  0.0699   22867.43
##
## sigma^2 estimated as 1.114e+10:  log likelihood = -1779.31,  aic = 3564.62
##
## $degrees_of_freedom
## [1] 135
##
## $ttable
##          Estimate         SE t.value p.value
## ar1        0.6095     0.0699  8.7154       0
## xmean 712768.3124 22867.4287 31.1696       0
##
## $AIC
## [1] 24.16325
##
## $AICc
## [1] 24.17916
```

```
##
## $BIC
## [1] 23.20588
```

Clearly, we still need a more complicated model as indicated by the Ljung-Box statistic, however, we now see that the Q-Q plot looks much more normal (though by no means perfect) and the standardized residual plot looks much more stationary.

Due to a the high autocorrelation values for lags 0 and 1, we choose to add a moving average component of 2 (which attempts to capture autocorrelation). Additionaly, we arbitrarily increase the degree of autoregression to 2 to account for slightly positive autocorrelation for lag 1. We thus obtain an ARMA(2, 2) model.

```
acf2(data_outliers$yadj, max.lag=20,
     main='ACF and PACF on views aggregated from Dec 1, 2007 to April 30, 2019 duration')
```



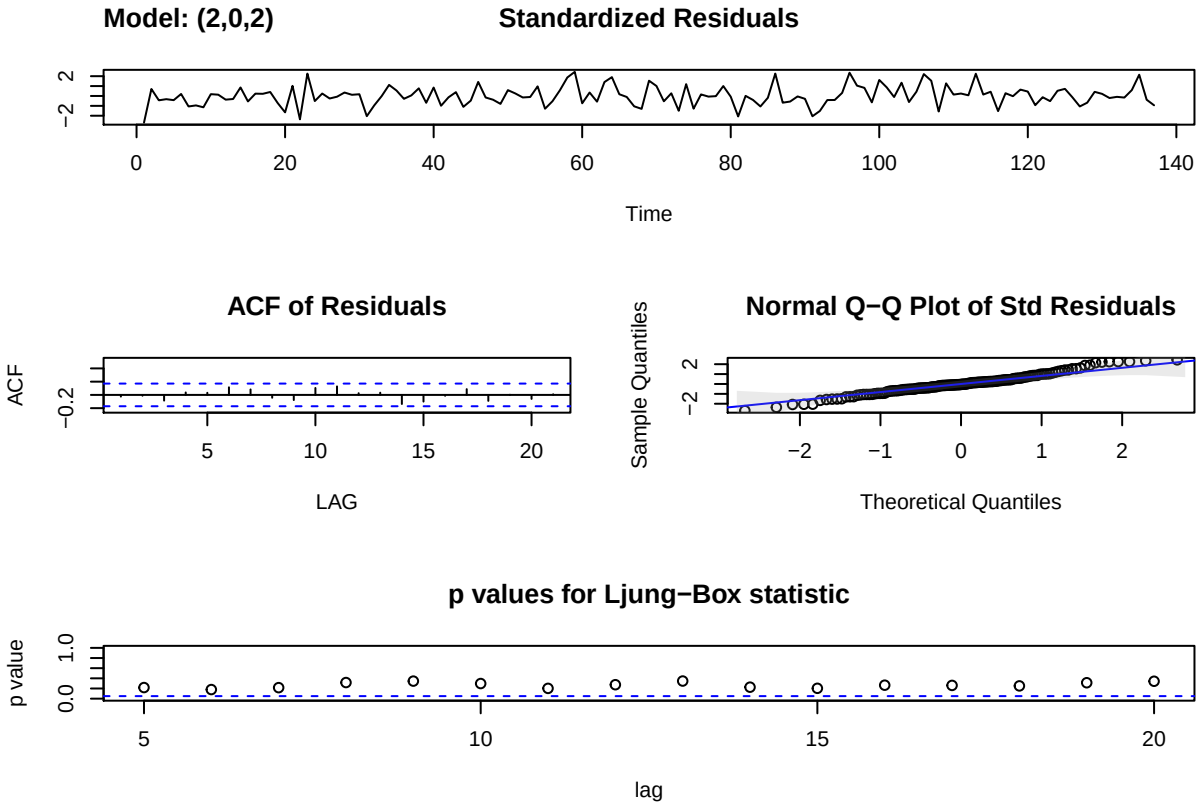

```
##          ACF  PACF
##  [1,]  0.58  0.58
##  [2,]  0.36  0.04
##  [3,]  0.15 -0.11
##  [4,]  0.21  0.23
##  [5,]  0.16 -0.04
##  [6,]  0.25  0.17
##  [7,]  0.15 -0.09
##  [8,]  0.13  0.01
##  [9,]  0.05  0.00
## [10,]  0.19  0.18
## [11,]  0.14 -0.07
## [12,]  0.12 -0.05
## [13,] -0.04 -0.12
```

```
## [14,] -0.10 -0.11
## [15,] -0.18 -0.06
## [16,] -0.07  0.05
## [17,] -0.08 -0.07
## [18,] -0.07 -0.04
## [19,] -0.12  0.03
## [20,] -0.09 -0.02
```

```
ts_fit_adj_ma <- sarima(data_outliers$yadj, p=2, d=0, q=2)
```

```
## initial  value 11.760969
## iter   2 value 11.705380
## iter   3 value 11.576270
## iter   4 value 11.529076
## iter   5 value 11.526442
## iter   6 value 11.524971
## iter   7 value 11.521912
## iter   8 value 11.518442
## iter   9 value 11.514908
## iter  10 value 11.507860
## iter  11 value 11.498440
## iter  12 value 11.493199
## iter  13 value 11.492481
## iter  14 value 11.487625
## iter  15 value 11.487306
## iter  16 value 11.483797
## iter  17 value 11.480672
## iter  18 value 11.474938
## iter  19 value 11.474154
## iter  20 value 11.469996
## iter  21 value 11.469258
## iter  22 value 11.466638
## iter  23 value 11.462579
## iter  24 value 11.461483
## iter  25 value 11.453926
## iter  26 value 11.448776
## iter  26 value 11.448776
## iter  27 value 11.448756
## iter  27 value 11.448756
## iter  28 value 11.448753
## iter  28 value 11.448753
## iter  28 value 11.448753
## final  value 11.448753
## converged
## initial  value 11.520477
## iter   2 value 11.517306
## iter   3 value 11.514021
## iter   4 value 11.511703
## iter   5 value 11.511100
## iter   6 value 11.510684
## iter   7 value 11.510625
## iter   8 value 11.510617
## iter   9 value 11.510614
## iter  10 value 11.510603
## iter  11 value 11.510599
```

```
## iter  12 value 11.510589
## iter  13 value 11.510588
## iter  14 value 11.510582
## iter  15 value 11.510572
## iter  16 value 11.510569
## iter  17 value 11.510567
## iter  18 value 11.510561
## iter  19 value 11.510548
## iter  20 value 11.510523
## iter  21 value 11.510486
## iter  22 value 11.510463
## iter  23 value 11.510459
## iter  24 value 11.510459
## iter  25 value 11.510458
## iter  26 value 11.510457
## iter  26 value 11.510457
## iter  26 value 11.510457
## final  value 11.510457
## converged
```




```
ts_fit_adj_ma
```

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
```

```
## Coefficients:
##           ar1     ar2     ma1     ma2     xmean
##       -0.4282  0.5617  1.1290  0.1701  713086.0
## s.e.   0.1056  0.1130  0.1201  0.1423   22295.9
##
## sigma^2 estimated as 9.843e+09:  log likelihood = -1771.33,  aic = 3554.65
##
## $degrees_of_freedom
## [1] 132
##
## $ttable
##          Estimate         SE t.value p.value
## ar1       -0.4282     0.1056 -4.0562  0.0001
## ar2        0.5617     0.1130  4.9707  0.0000
## ma1        1.1290     0.1201  9.4019  0.0000
## ma2        0.1701     0.1423  1.1954  0.2341
## xmean 713085.9643 22295.8978 31.9828  0.0000
##
## $AIC
## [1] 24.08303
##
## $AICc
## [1] 24.10235
##
## $BIC
## [1] 23.1896
```

Although our model appears to have captured all of the serial autocorrelation according to the Ljung-Box statistic, our Q-Q plot for standard residuals looks much worse. Clearly, we need to be careful with tuning our parameters.

Accounting for the removal of outliers, it is clear that even a simple ARMA model performs as a decent baseline for future time series models. We thus conclude that, looking at a larger time frame, even though a chilling effect may have taken place, it was only a small blip in the grand scheme of things. Overall, terrorism search trends over time have not changed.

4. Summary

From the original study and the reproduced results, we can conclude that there exists the 'Chilling Effect'. The Mass Surveillance does affect how people use the Internet (Wikipedia), if we use the coefficients and their significance as a metric. However, we also found that the outlliers and data processing affect the results greatly, especially for the control groups: infra structure-related and popular articles which do not have significance change in trend and level in the paper, but the changes are significant in our results.

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 seperated 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.

We investigate further into keyword-level analysis to quantify and decompose the total trend for terrorism-related keywords. We found that the model with interaction always gives better RMSE for all keywords. However, the small improvement raise the question about the impact that the surveilence actually causes.

However, it is difficult to investigate further since the regression only captures trend not the exact views as discussed earlier. Moreover, we also found that all but one keywords have significant decrease in their trends. This can be analyzed further in future work whether the slopes change their sign or just decrease in magnitude.

Finally, we attempted to use a more rigorous time series analysis to see if the paper's claimed trends hold over a longer period of time. We suspected, and confirmed, using time series analysis that some datapoints post-Snowden are indeed low outliers. However, this may or may not indicate the presence of the chilling effect, as these low outliers occured well after Snowden's revelations and appeared to be more of a sudden drop than a gradual decline. Additionally, we discovered that the increasing trends in the months immediately pre-Snowden were more anomalous than the norm considering the grand scheme of things. Using a time series model corrected for outliers, we show that, while there may have been a chilling effect, that the effects appeared more temporary than continuous. We can thus conclude that the chilling effect applies for a very short window pre- and post- Snowden but not for larger time frames.

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

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.3
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/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] tsoutliers_0.6-8     astsa_1.8            wikipediatrend_2.1.1
##  [4] lubridate_1.7.4      forcats_0.4.0        stringr_1.4.0
##  [7] dplyr_0.8.0.1        purrr_0.3.2          readr_1.3.1
## [10] tidyr_0.8.3          tibble_2.1.1         ggplot2_3.1.1
## [13] tidyverse_1.2.1      scales_1.0.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.1        lattice_0.20-38   zoo_1.8-5
##  [4] assertthat_0.2.1  digest_0.6.18     lmtest_0.9-37
##  [7] utf8_1.1.4        R6_2.4.0          cellranger_1.1.0
## [10] plyr_1.8.4        backports_1.1.4   evaluate_0.13
## [13] httr_1.4.0        pillar_1.3.1      rlang_0.3.4
## [16] curl_3.3          lazyeval_0.2.2    readxl_1.3.1
## [19] rstudioapi_0.10   TTR_0.23-4        fracdiff_1.4-2
## [22] hellno_0.0.1      rmarkdown_1.12    labeling_0.3
## [25] munsell_0.5.0     broom_0.5.2       compiler_3.6.0
## [28] modelr_0.1.4      xfun_0.6          pkgconfig_2.0.2
## [31] forecast_8.7      urca_1.3-0        htmltools_0.3.6
## [34] nnet_7.3-12       tidyselect_0.2.5  quadprog_1.5-6
## [37] fansi_0.4.0       crayon_1.3.4      withr_2.1.2
## [40] grid_3.6.0        nlme_3.1-139      jsonlite_1.6
## [43] gtable_0.3.0      magrittr_1.5      quantmod_0.4-14
```

```
## [46] cli_1.1.0         stringi_1.4.3     tseries_0.10-46
## [49] timeDate_3043.102 xml2_1.2.0        xts_0.11-2
## [52] generics_0.0.2    tools_3.6.0       glue_1.3.1
## [55] hms_0.4.2         parallel_3.6.0    yaml_2.2.0
## [58] colorspace_1.4-1  rvest_0.3.3       knitr_1.22
## [61] haven_2.1.0
```