IE 423 Project Part 1

Baran Kırkgöz & Orkan Çelikhisar

2023-10-31

Introduction

In this project, we aim to analyze 6 stocks of Borsa Istanbul and detect outliers by using Boxplot and 3-sigma limits. After this step, we compared each stock with relevant google trends data. For google trends, we used the names of the companie, eg. for MGROS stock data we used "Migros" search volume.

Identification of Outliers

First necesssary packages and libraries are installed.

There are some errors but it is ok.

Now the data is retrieved.

```
data <- read_csv("all_ticks_wide.csv.gz")
```

```
## Rows: 50012 Columns: 61
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (60): AEFES, AKBNK, AKSA, AKSEN, ALARK, ALBRK, ANACM, ARCLK, ASELS, ASU...
## dttm  (1): timestamp
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Let's organize the data in choronological order and select 6 sample stock.

Banking: Akbank, Is Bankası / Retail: Migros, Arçelik / Energy: Aygaz, Tüpraş

```
data <- data %>% arrange(timestamp)
selected_stocks <- c("AKBNK", "ISCTR", "MGROS", "ARCLK", "AYGAZ", "TUPRS")
```

Filter the data for the selected stocks and for a duration of at least 2 years

```
data_filtered <- data %>%
  select(all_of(c("timestamp", selected_stocks)))
```

Realize that working with the long format is easier Totally forget the fact that the data was given to us in both formats Ask ChatGPT how to transform stock data from wide format to long format Convert data to long format

```
data_long <- data_filtered %>%
  pivot_longer(cols = -timestamp, names_to = "name", values_to = "value")
```

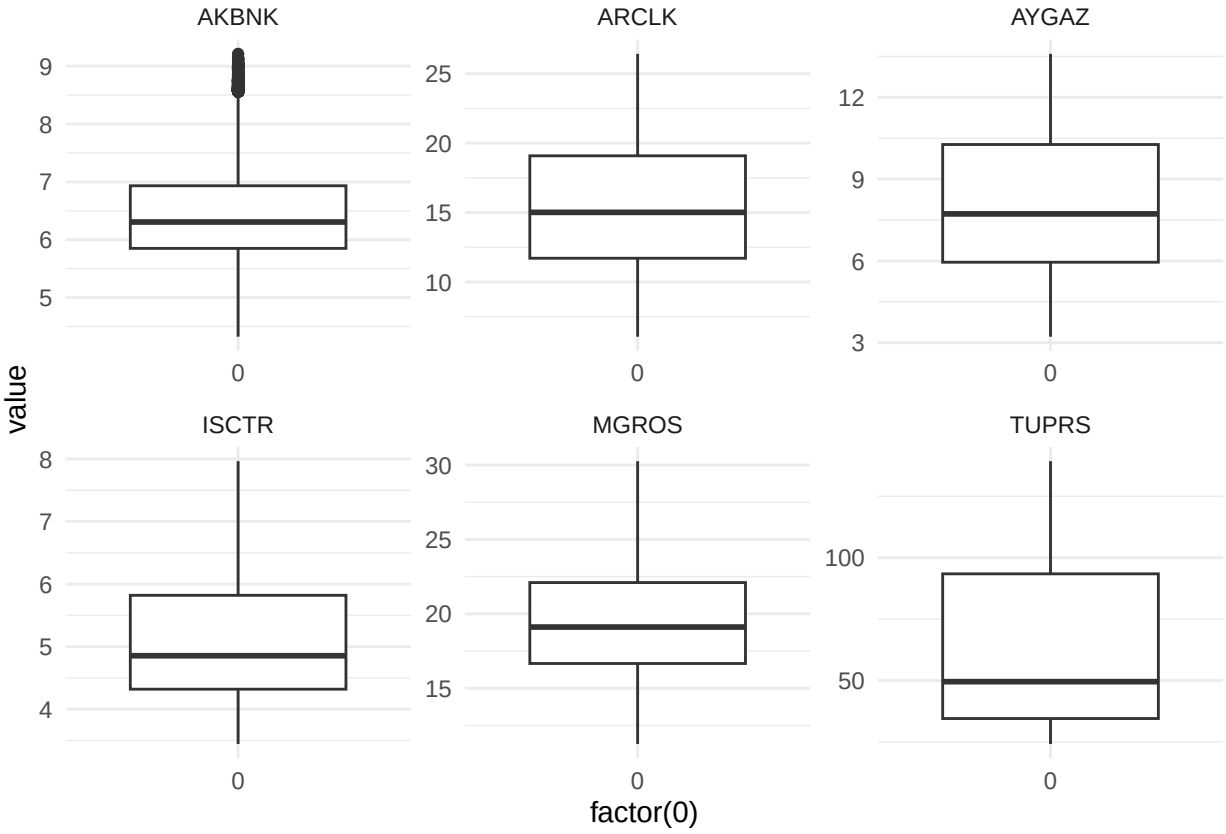
It seems that in the dataset there's a 0.0001 value for every stock at the exact same timestamp This is most probably a placeholder To be able to conduct an accurate analysis, we have removed the rows with the values 0.0001

```
data_long <- data_long %>%
  filter(value != 0.0001)
```

Identification of Outliers using Boxplots and 3-Sigma Limits: Generate boxplots for the chosen indices

```
p <- ggplot(data_long, aes(x = factor(0), y = value)) +
  geom_boxplot() +
  facet_wrap(~name, scales = "free") +
  theme_minimal()

print(p)
```



3-Sigma rule

```
outliers <- data_long %>%
  group_by(name) %>%
  mutate(month = format(timestamp, "%Y-%m"),
         mean_price = mean(value, na.rm = TRUE),
         sd_price = sd(value, na.rm = TRUE),
         lower_limit = mean_price - 3 * sd_price,
         upper_limit = mean_price + 3 * sd_price) %>%
  filter(value < lower_limit | value > upper_limit)

print(outliers)
```

```
## # A tibble: 0 x 8
## # Groups:   name [0]
```

```
## # i 8 variables: timestamp <dttm>, name <chr>, value <dbl>, month <chr>,
## #   mean_price <dbl>, sd_price <dbl>, lower_limit <dbl>, upper_limit <dbl>
```

After removing the 0.0001 placeholders, according to the 3-Sigma rule there are no outliers. Let's try the 3-Sigma rule for monthly variations now in order to achieve a higher sensitivity.

3-Sigma rule for monthly variations

```
outliers <- data_long %>%
  group_by(name, month = format(timestamp, "%Y-%m")) %>%
  mutate(mean_price = mean(value, na.rm = TRUE),
         sd_price = sd(value, na.rm = TRUE),
         lower_limit = mean_price - 3 * sd_price,
         upper_limit = mean_price + 3 * sd_price) %>%
  filter(value < lower_limit | value > upper_limit)

outliers
```

```
## # A tibble: 434 x 8
## # Groups:   name, month [73]
##    timestamp           name  value month   mean_price sd_price lower_limit
##    <dttm>              <chr> <dbl> <chr>        <dbl>    <dbl>       <dbl>
##  1 2012-10-04 09:15:00 MGROS 18.4  2012-10      19.1     0.232       18.4
##  2 2012-10-04 11:00:00 MGROS 18.4  2012-10      19.1     0.232       18.4
##  3 2012-11-01 07:45:00 ARCLK  7.67 2012-11       7.06    0.201        6.46
##  4 2012-11-01 08:00:00 ARCLK  7.71 2012-11       7.06    0.201        6.46
##  5 2012-11-01 08:30:00 ARCLK  7.71 2012-11       7.06    0.201        6.46
##  6 2012-11-01 08:45:00 ARCLK  7.67 2012-11       7.06    0.201        6.46
##  7 2012-11-01 09:45:00 ARCLK  7.74 2012-11       7.06    0.201        6.46
##  8 2012-11-01 10:00:00 ARCLK  7.74 2012-11       7.06    0.201        6.46
##  9 2012-11-01 10:15:00 ARCLK  7.74 2012-11       7.06    0.201        6.46
## 10 2012-11-01 12:00:00 ARCLK  7.74 2012-11       7.06    0.201        6.46
## # i 424 more rows
## # i 1 more variable: upper_limit <dbl>
```

Now there are a lot of outliers in detailed timestamps. This has allowed us to carefully examine and correlate historical events and outliers.

Convert the timestamp to date and then create a 6-monthly grouping variable

```
data_long <- data_long %>%
  mutate(date = as.Date(timestamp),
         six_month_period = ifelse(month(date) %in% 1:6, paste(year(date), "H1", sep="-"), paste(year(da
```

3-Sigma rule for 6-monthly variations

```
outliers_6months <- data_long %>%
  group_by(name, six_month_period) %>%
  mutate(mean_price = mean(value, na.rm = TRUE),
         sd_price = sd(value, na.rm = TRUE),
         lower_limit = mean_price - 3 * sd_price,
         upper_limit = mean_price + 3 * sd_price) %>%
  filter(value < lower_limit | value > upper_limit)
```

```
outliers_6months
```

```
## # A tibble: 216 x 9
## # Groups:   name, six_month_period [17]
##    timestamp           name  value date       six_month_period mean_price
##    <dttm>              <chr> <dbl> <date>     <chr>                  <dbl>
##  1 2012-12-10 15:00:00 MGROS  23.2 2012-12-10 2012-H2                19.8
##  2 2012-12-13 07:45:00 MGROS  24.4 2012-12-13 2012-H2                19.8
##  3 2012-12-13 08:00:00 MGROS  23.5 2012-12-13 2012-H2                19.8
##  4 2013-06-24 13:15:00 MGROS  17.7 2013-06-24 2013-H1                22.3
##  5 2013-06-24 13:30:00 MGROS  17.6 2013-06-24 2013-H1                22.3
##  6 2013-06-24 13:45:00 MGROS  17.6 2013-06-24 2013-H1                22.3
##  7 2013-06-24 14:00:00 MGROS  17.6 2013-06-24 2013-H1                22.3
##  8 2013-06-24 14:15:00 MGROS  17.4 2013-06-24 2013-H1                22.3
##  9 2013-06-24 14:30:00 MGROS  17.5 2013-06-24 2013-H1                22.3
## 10 2013-06-25 06:30:00 MGROS  17.7 2013-06-25 2013-H1                22.3
## # i 206 more rows
## # i 3 more variables: sd_price <dbl>, lower_limit <dbl>, upper_limit <dbl>
```

Now there are much less outliers but still not a very small amount

Judging from the outliers we've received, we can actually observe the mid-2019 stock price peaks of Migros due to covid-19, as well as the fluctuations of Akbank and Isbank due to the instability of the TRY exchange rate and the rash economic policies that were followed by the Turkish Central Bank.

I personally was interested to see the effect of the total IT Malfunction of Akbank on July 2021, though it seems the dataset is not up-to-date.

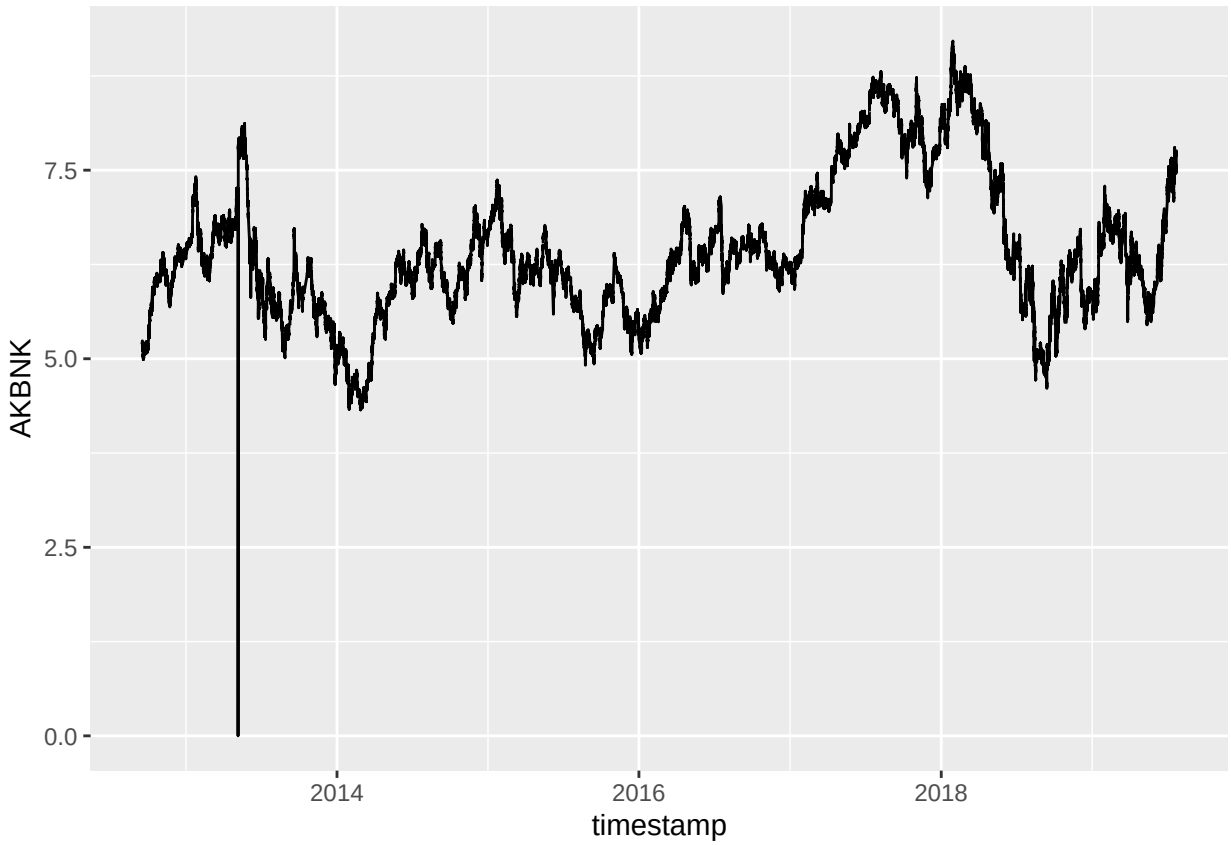
Google Trends Data

Now to investigate the relationship between stocks and search volume of the related words the google trends data will be analyzed for each of the selected stock. The google trends data and the stock data will be analyzed to understand whether there is a correlation between them.

Akbank

To understand Akbank stock data, the keyword "Akbank" is selected for the years between 2012-2019. As it can be seen from the two graphs, there is no significant relationship between the trend of AKBNK stock and Akbank search volume on google.

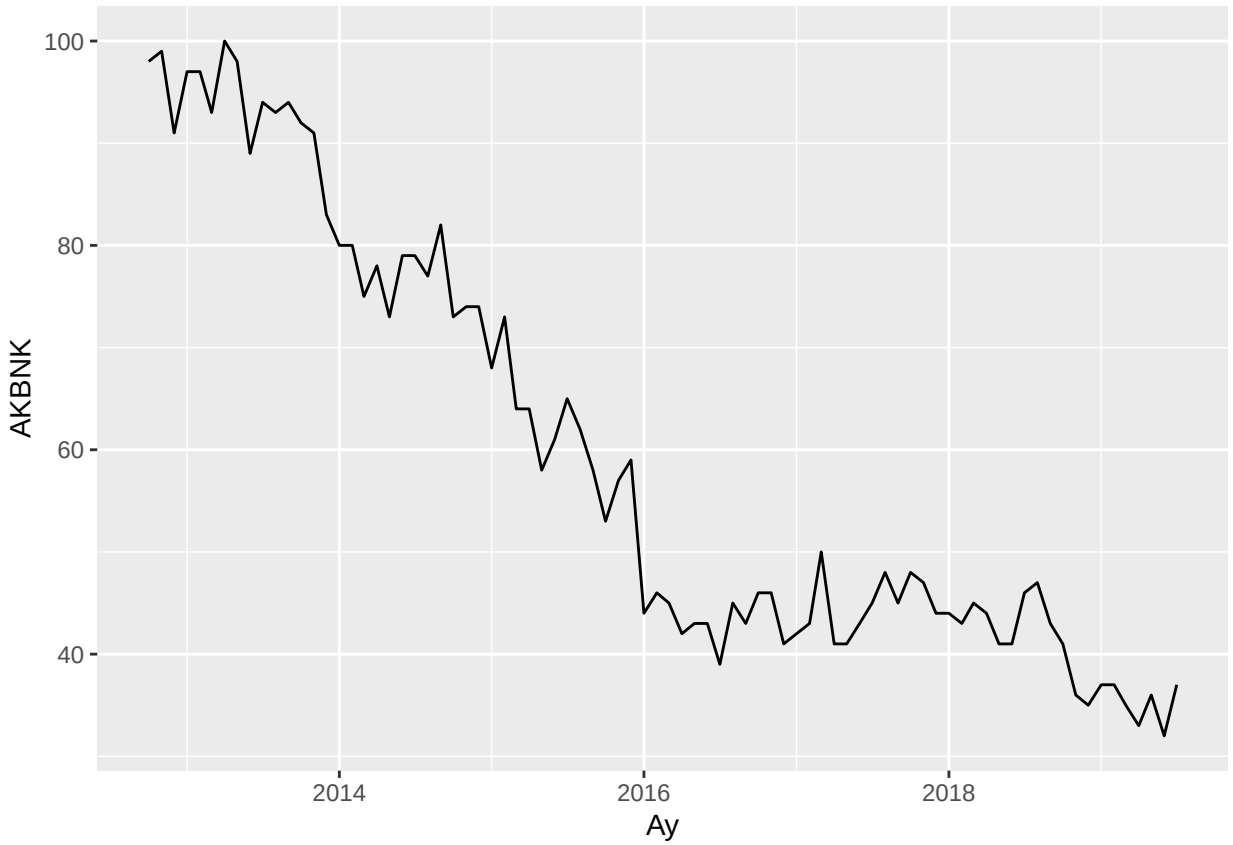
```
g <- ggplot(data_filtered, aes(x=timestamp, y=AKBNK)) +
  geom_line()
g
```



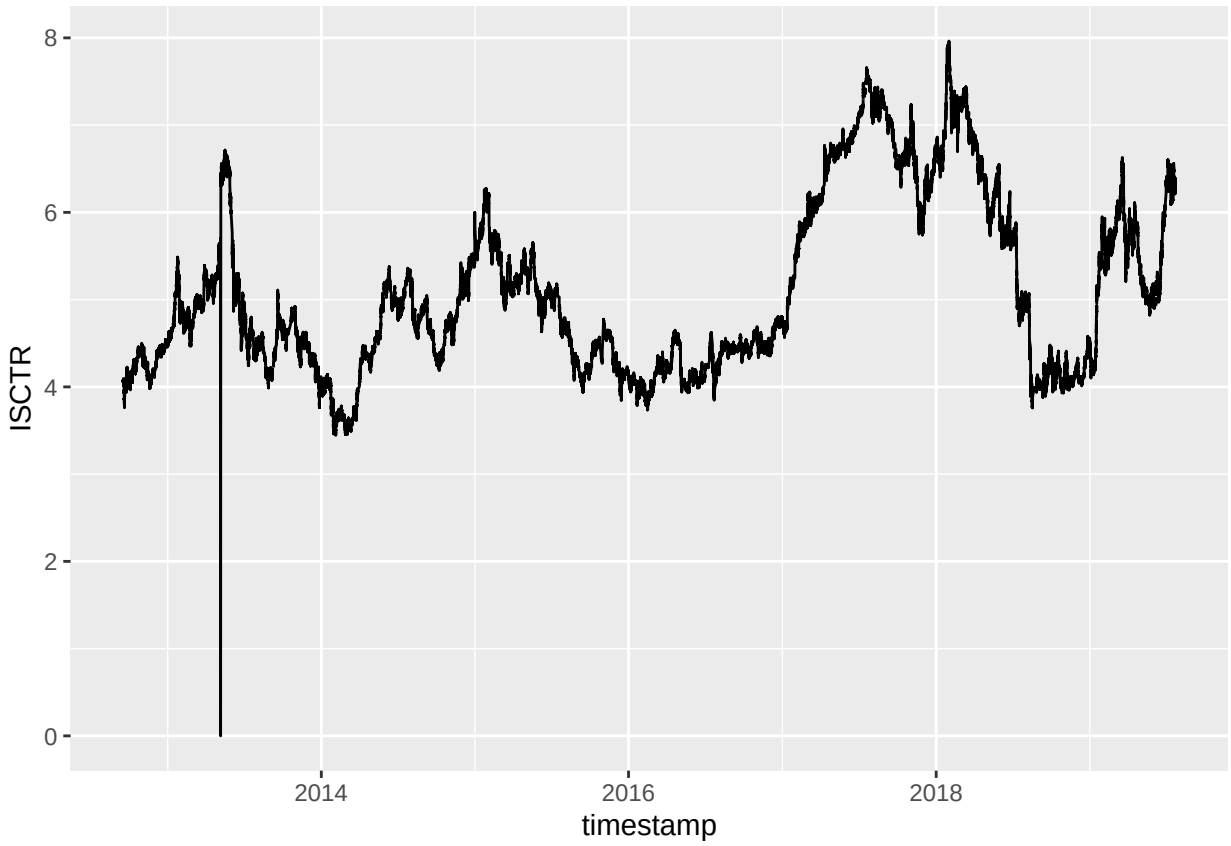

```
AKBNK_data <- read_csv("multiTimeline (4).csv")
```

```
## Rows: 82 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Ay
## dbl (1): AKBNK
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
AKBNK_data$Ay<- ym(AKBNK_data$Ay)
a <- ggplot(AKBNK_data, aes(x=Ay, y=AKBNK)) +
  geom_line()
a
```




```
x <- ggplot(data_filtered, aes(x=timestamp, y=ISCTR)) +
  geom_line()
x
```



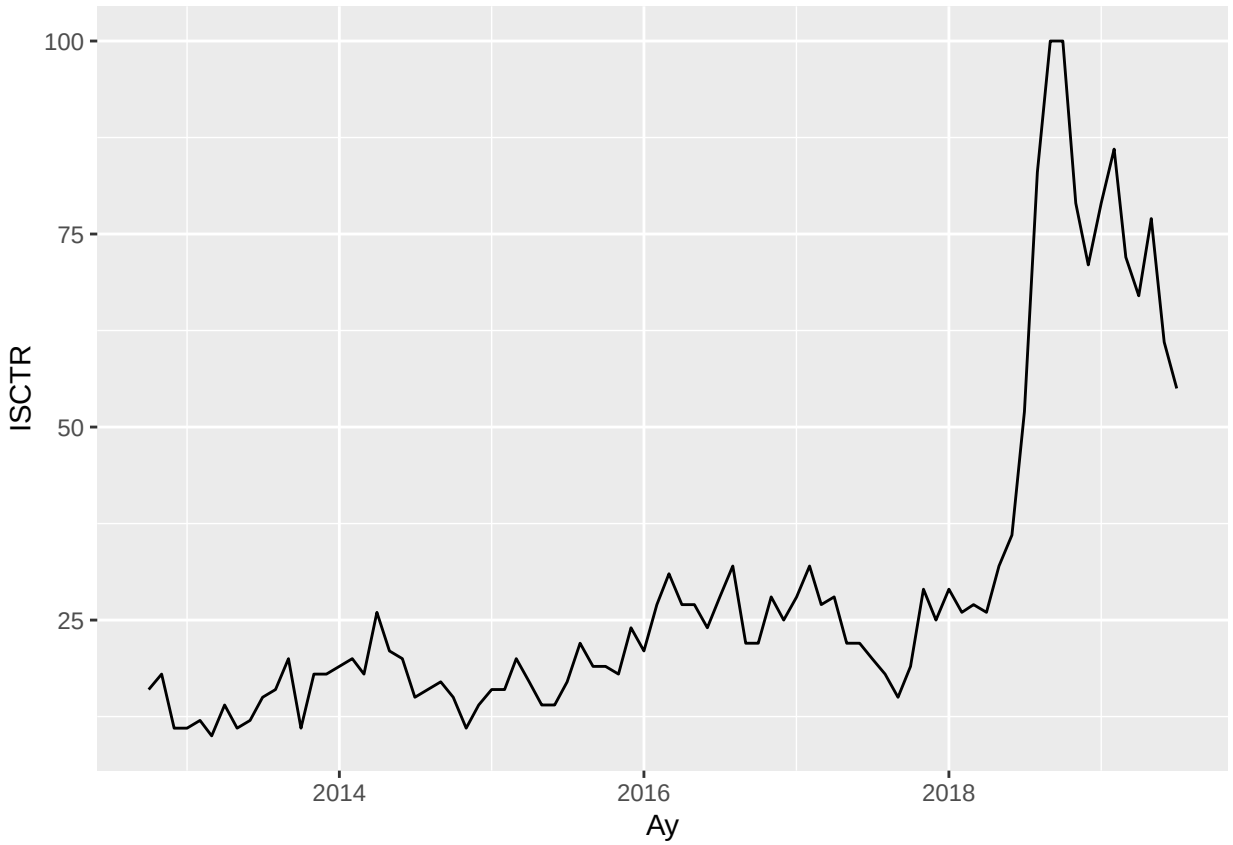
ISCTR Data

When the same analysis is performed for ISCTR stock, the positive coveriance around year 2018-2019 can be seen. Most probably İş Bankası was performing very well after 2018 and the effect of it can be seen on graphs.

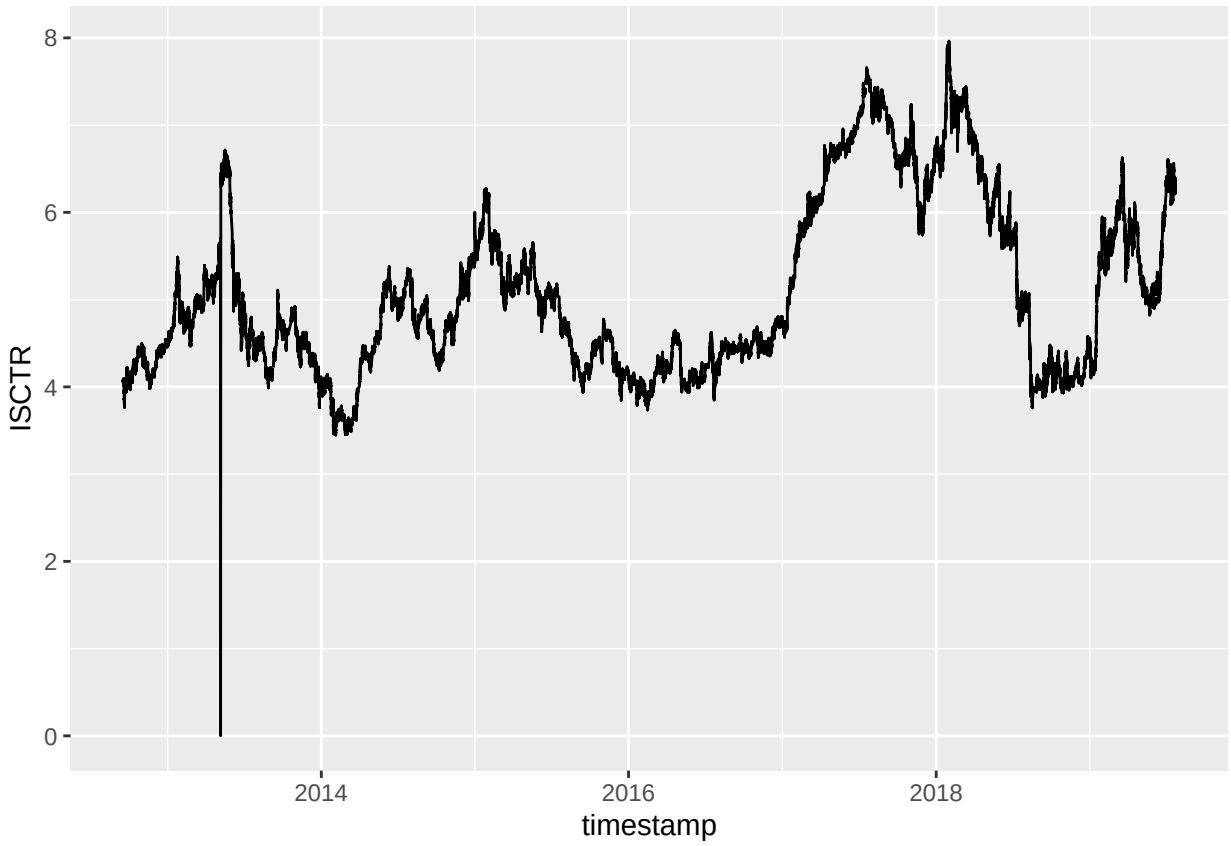
```
ISCTR_data <- read_csv("multiTimeline (5).csv")
```

```
## Rows: 82 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Ay
## dbl (1): ISCTR
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
ISCTR_data$Ay<- ym(ISCTR_data$Ay)
ISCTRgraph <- ggplot(ISCTR_data, aes(x=Ay, y=ISCTR)) +
  geom_line()
ISCTRgraph
```




```
y <- ggplot(data_filtered, aes(x=timestamp, y=ISCTR)) +
  geom_line()
y
```



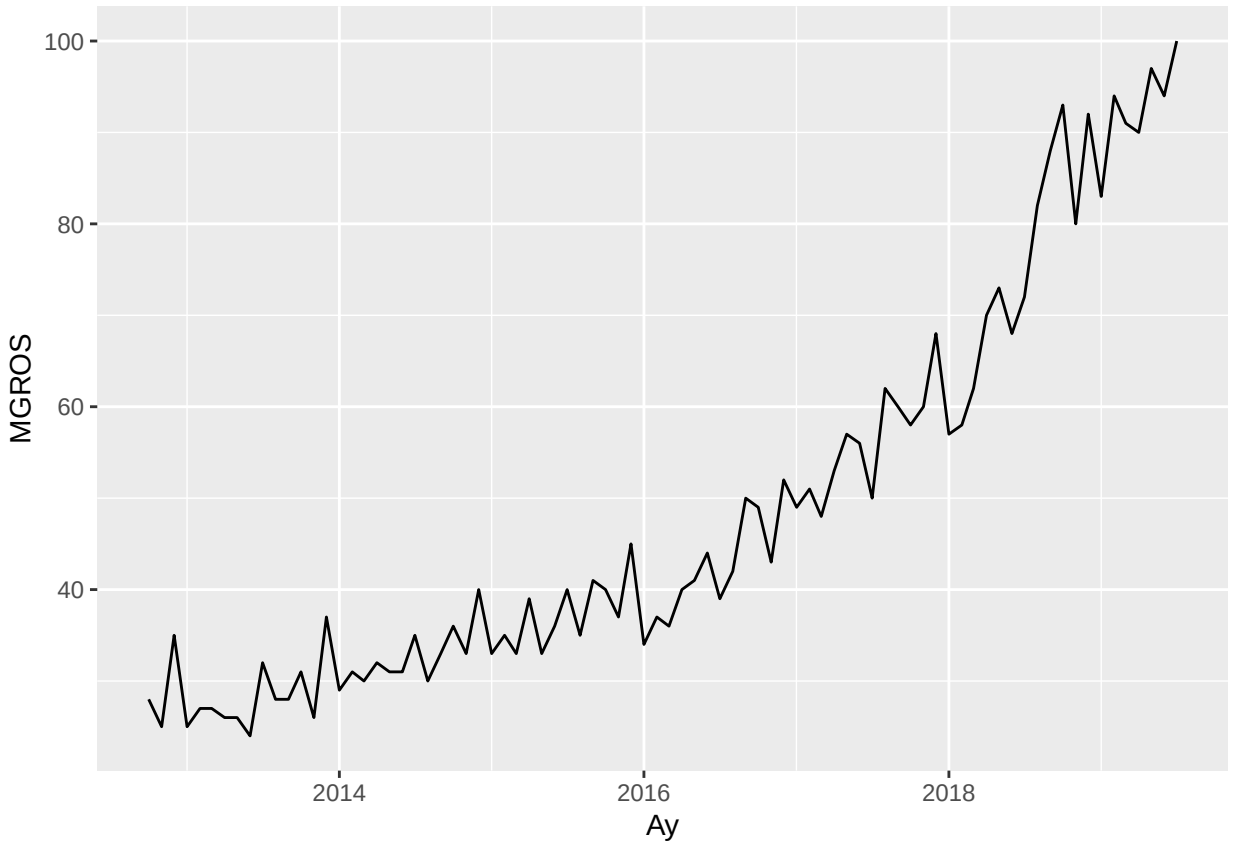
MGROS

When two graphs are analyzed, the covarience till 2018 can be detected. After 2018, the MGROS stock continues to climb whereas search volume of Migros starts to decline.

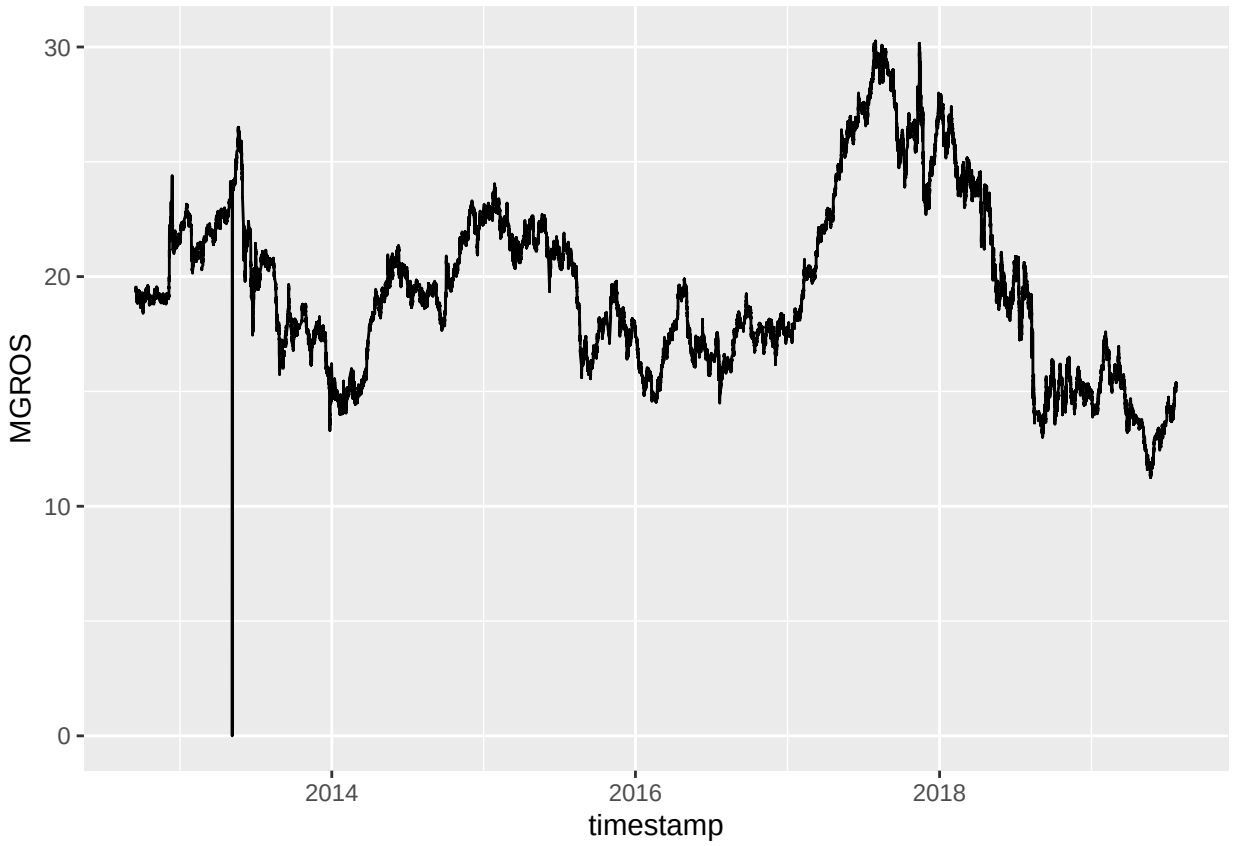
```
MGROS_data <- read_csv("multiTimeline (6).csv")
```

```
## Rows: 82 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Ay
## dbl (1): MGROS
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
MGROS_data$Ay<- ym(MGROS_data$Ay)
MGROSgraph <- ggplot(MGROS_data, aes(x=Ay, y=MGROS)) +
  geom_line()
MGROSgraph
```



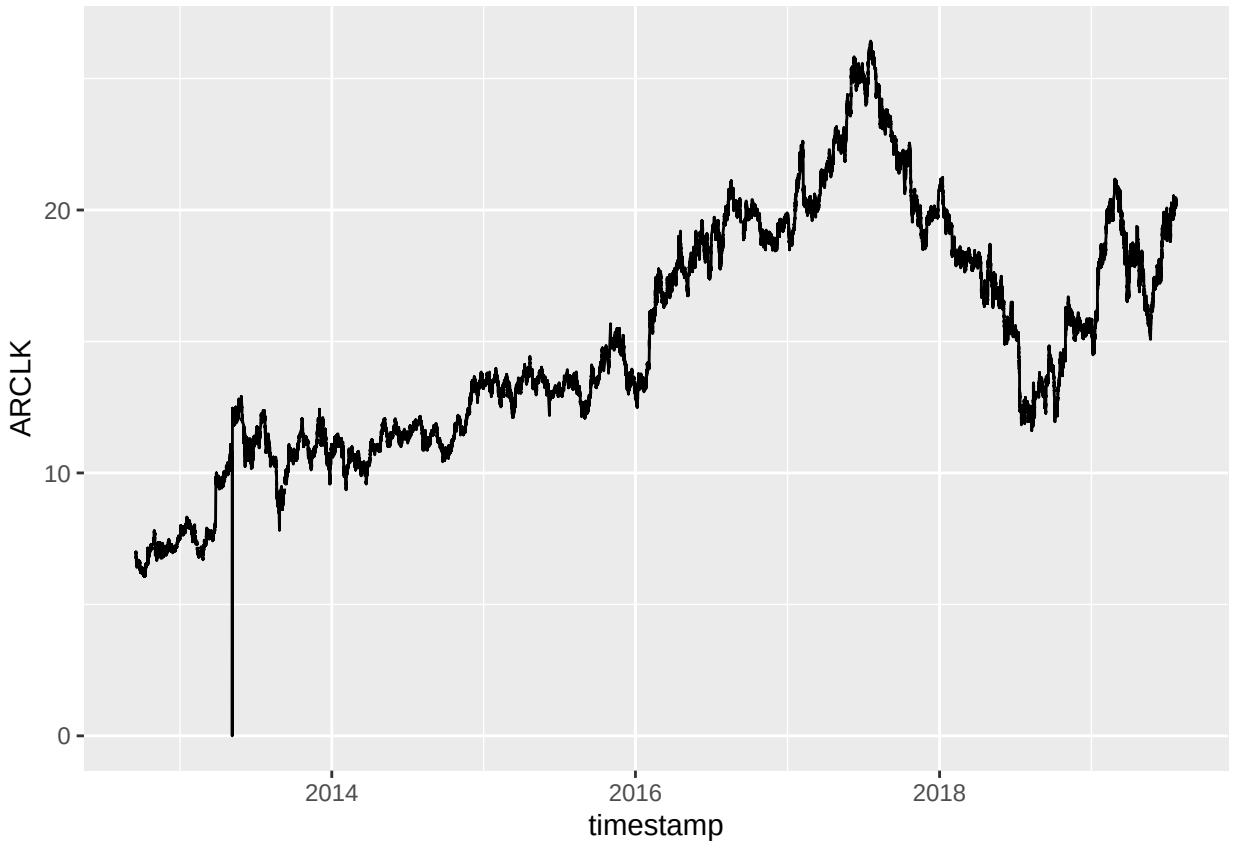

```
l <- ggplot(data_filtered, aes(x=timestamp, y=MGROS)) +
  geom_line()
l
```



ARCLK

Except years2018- 2019, both of the graphs are very similiar. The high coveriance can be seen easily. They both start to incline from 2012 till 2018 with little fluctuations. In 2018-2019 the search volume still increases wheras the stock goes down.

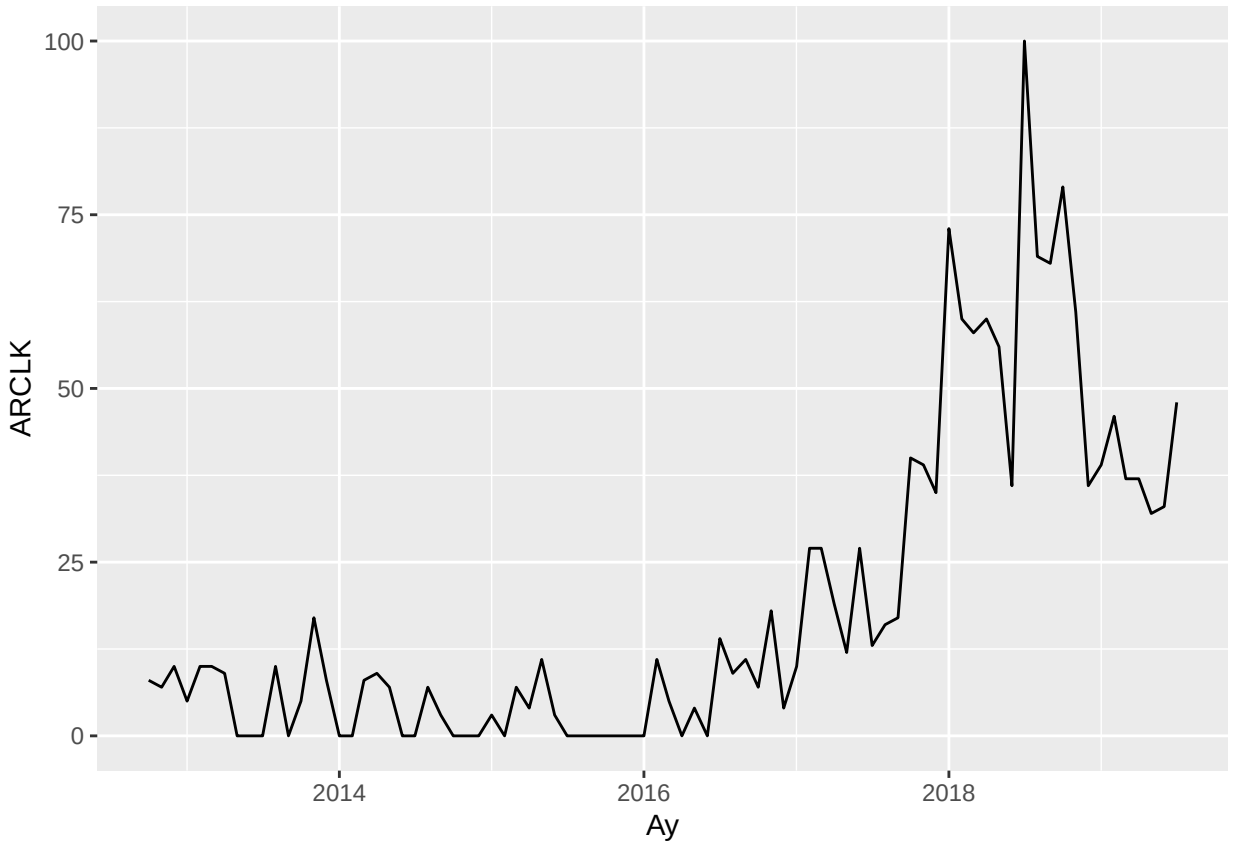
```
k <- ggplot(data_filtered, aes(x=timestamp, y=ARCLK)) +
  geom_line()
k
```




```
ARCLK_data <- read_csv("multiTimeline (7).csv")
```

```
## Rows: 82 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Ay
## dbl (1): ARCLK
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

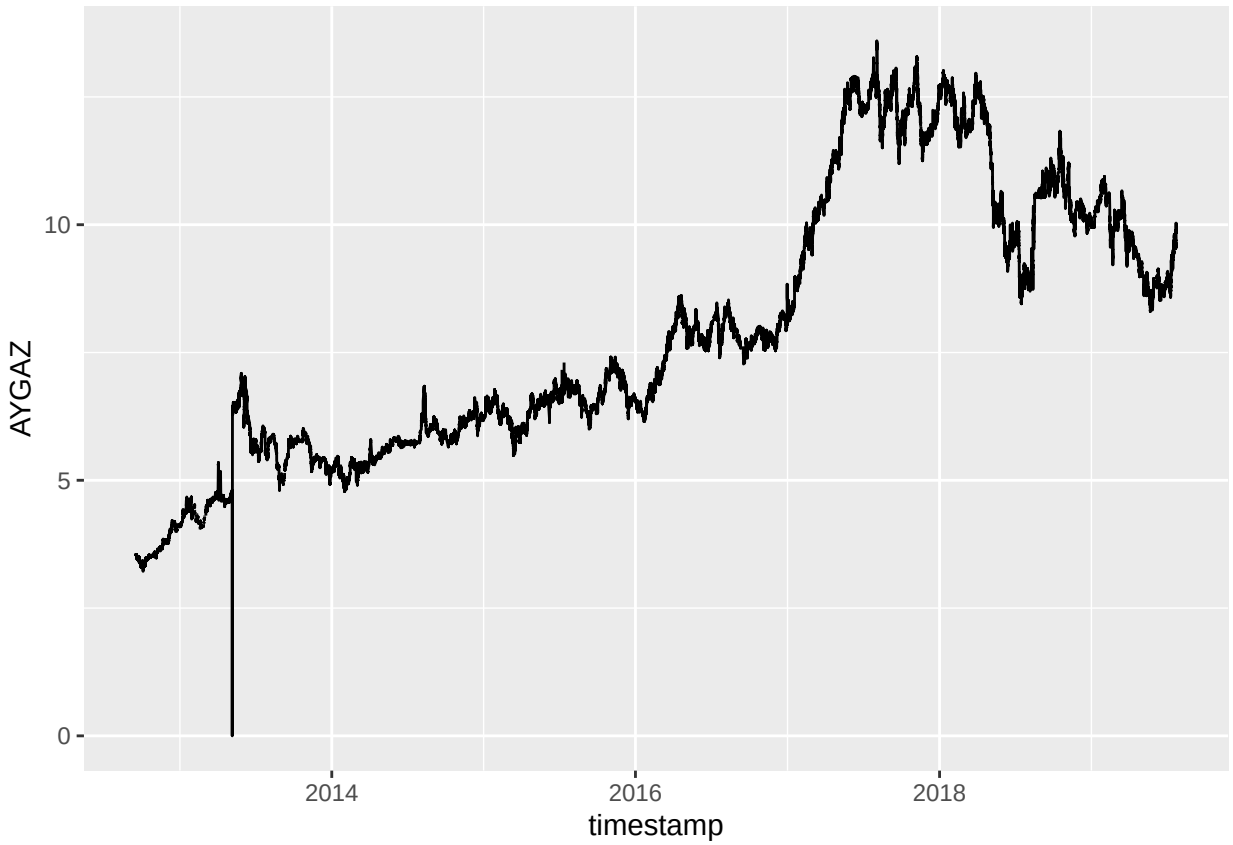
```
ARCLK_data$Ay<- ym(ARCLK_data$Ay)
ARCLKgraph <- ggplot(ARCLK_data, aes(x=Ay, y=ARCLK)) +
  geom_line()
ARCLKgraph
```



AYGAZ

The relationship between Aygaz stock and search volume is really strong. The graphs are very similiar expect the peak level of search volume in 2020.

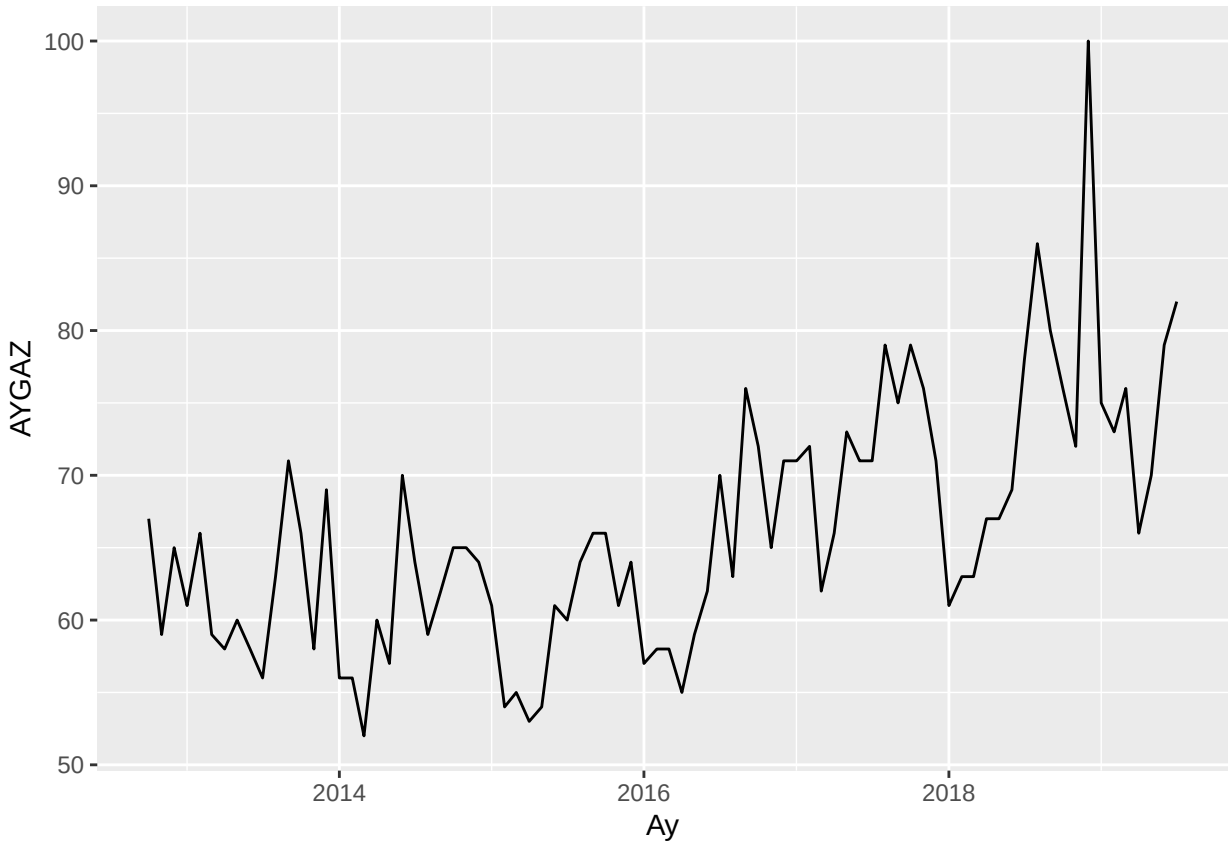
```
m <- ggplot(data_filtered, aes(x=timestamp, y=AYGAZ)) +
  geom_line()
m
```




```
AYGAZ_data <- read_csv("multiTimeline (8).csv")
```

```
## Rows: 82 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Ay
## dbl (1): AYGAZ
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

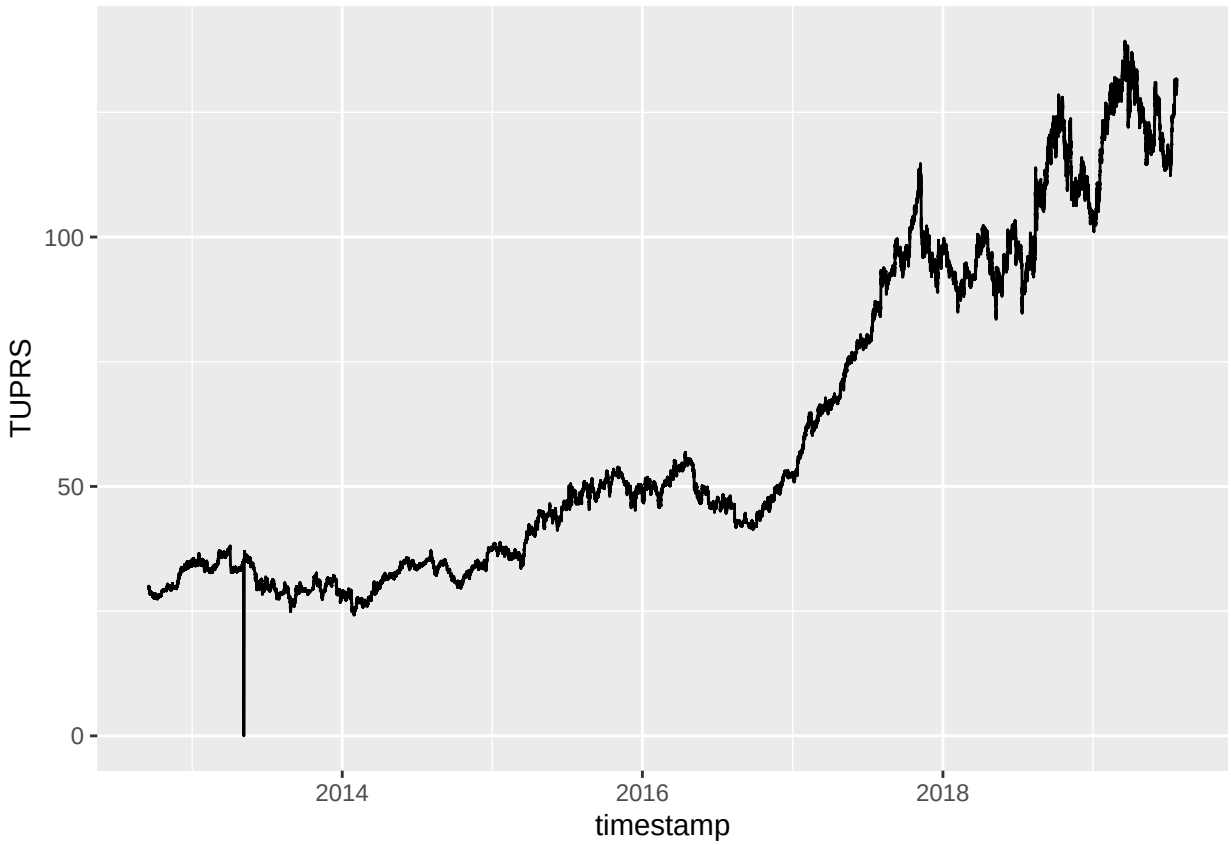
```
AYGAZ_data$Ay<- ym(AYGAZ_data$Ay)
AYGAZgraph <- ggplot(AYGAZ_data, aes(x=Ay, y=AYGAZ)) +
  geom_line()
AYGAZgraph
```



TUPRS

The two graphs are very similar after 2016, both of them start to increase with varience.

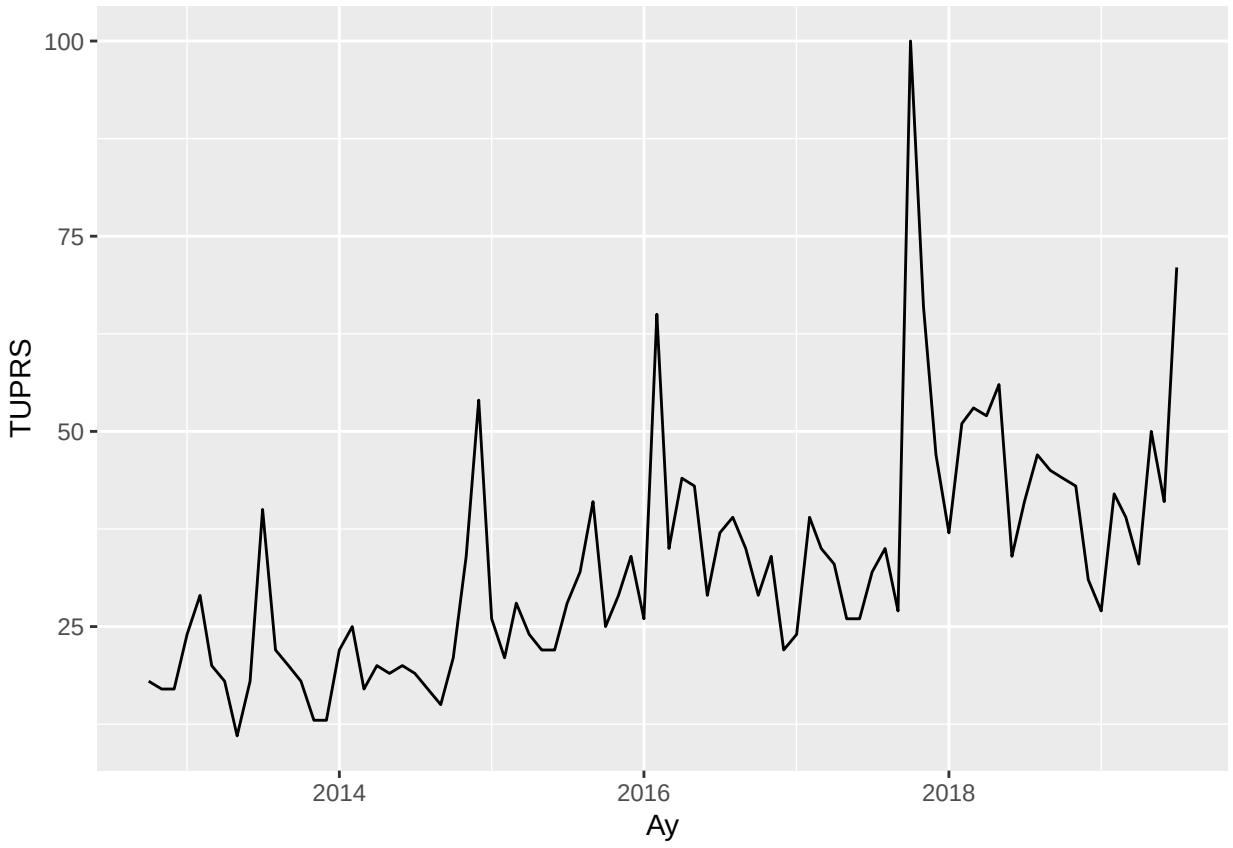
```
n <- ggplot(data_filtered, aes(x=timestamp, y=TUPRS)) +
  geom_line()
n
```




```
TUPRS_data <- read_csv("multiTimeline (9).csv")
```

```
## Rows: 82 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Ay
## dbl (1): TUPRS
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
TUPRS_data$Ay<- ym(TUPRS_data$Ay)
TUPRSgraph <- ggplot(TUPRS_data, aes(x=Ay, y=TUPRS)) +
  geom_line()
TUPRSgraph
```


100
75
50
25
TUPRS
2014
2016
2018
Ay