Who are the financial profiteers of war?

University of Applied Sciences Lucerne

Master of Science in Applied Information and Data Science

Time Series in Finance (TSA01)

Authors: Michèle Gerber and Damien Grossniklaus
Date of Submission: 22. December 2023

Contents

```r
install.packages("readr")
install.packages("colorspace")
install.packages("vctrs")
install.packages("gtrendsR")
```

```
## Warning in register(): Can't find generic `scale_type` in package ggplot2 to
## register S3 method.
```

TODO: * Ticker Section Mentioning in Charts and Code –> in Einleitung erwähnt und in Grafik angepasst * Nach differencing nochmals Stationarity testen –> ergänzt * Casality testing andere Seite –> Einfluss Ticker auf Trends ausschliessen –> in Code ergänzt * Add Inflation as Controll Variable

Potential structure

- Context and Goal (Literature?)
- Methods
- Data
  - Import data (indices & Google words)
  - Describe data
  - Preprocessing
- VAR & Causality Testing
- Results & Discussion

1. Context and Goal

With recent outbreaks of war (e.g., in Israel or in the Ukraine), the question arises of how stock markets react to this and if there are sectors that profit from such crises. Specifically, this case study examines if several indices from specific economic sectors perform well in situations of crisis. The following six indices are included in the analysis:

- **XLK**: Technology Select Sector
- **XLF**: Financial Select Sector
- **XLV**: Health Care Select Sector
- **XLY**: Consumer Discretionary Select Sector
- **XLE**: Energy Select Sector
- **XLU**: Utilities Select Sector

Essentially our research question is the following: *Who are the financial profiteers of war?*. To answer this question, the performance of the sector-specific indices should be compared to an overall crisis measure. As an overall crisis measure the Google Trend data for the search word *"war"* is used. We postulate, that an increased search for this term points to a higher crisis. So in conclusion, we want to examine if there are positive correlations between our crisis measure and stock market profits in the different indices.

2. Methods

Our research question is of an exploratory nature. Therefore we don't have a hypothesis going in, as to which indices will perform better under situations of crisis. The method we will us is vector autoregression (VAR). This allows us to measure the influence of our crisis measure on the performance of the different indices. For each index we will calculate a separate model. Furthermore, to address potential spurious correlations a Granger causality test will be conducted for every ticker. However, before a VAR and causality test

can be carried out the data in form of different time series has to be imported, cleaned and pre-processed. Concretely, potential seasonal effects and trends in time series have to be identified and adjusted for to achieve stationarity, which characterizes a stabilized mean and variance for the time series over time and will ultimately allow more reliable and accurate modeling. To test the stationarity of a time the "*Augmented Dickey-Fuller Test*" will be carried out. If a time series is non-stationary, lagged differences will be calculated. Lastly, to deal with right skewness and achieve normal distributions in the data, the time series will first be logarithmized before applying the difference.

3. Data

Financial Data

The financial data used for the paper at hand comes from Yahoo Finance (2023)(**SOURCE**). The data can be retrieved using the `getSynmbols.yahoo()` function from the `quantmod` package. The data represents the weekly adjusted closing prices for the tickers introduced in chapter 1 for the time frame from December 2021 until December 2023. Mentioned time frame was chosen as it captures the start of the Russian invasion of Ukraine as well as the start of the Isreal-Hammas war. Additionally, the time frame and periodicity is considered of good balance between having enough data points as well as keeping the dataset manageable and focused. The decision to use the adjusted closing prices instead of the closing price can be justified that adjusted prices are adapted to address any splits, dividends or capital gain distributions. Lastly, the weekly periodicity of the ticker data was chosen as the Google Trends data comes in weekly measurements.

Google Trends Data

lorem ipsum

https://data.oecd.org/price/inflation-cpi.htm

Import Time Series Data

TODO: - Justify why Adjusted values and not close values - Timeframe: December 2021 until December 2023

```
# Ticker Data
ticker_data <- NULL
tickers_index <- c("XLK", "XLF", "XLV", "XLY", "XLE", "XLU")

for (Ticker in tickers_index){
  ticker_data <- cbind(ticker_data,
                       getSymbols.yahoo(Ticker, from="2021-12-01", to="2023-12-01",
                                        periodicity = "weekly",auto.assign=FALSE)[,6])
}

# Check length and min max date
length(ticker_data$XLK.Adjusted)
```

```
## [1] 105
```

```
summary(ticker_data)[c(1, 6)]
```

```
## [1] "Min.   :2021-11-29  " "Max.   :2023-11-27  "
```

```
head(ticker_data)
```

```
##            XLK.Adjusted XLF.Adjusted XLV.Adjusted XLY.Adjusted XLE.Adjusted
## 2021-11-29     162.1687     36.56124     126.3468     196.2840     50.84306
## 2021-12-06     171.7439     37.54185     130.4269     201.5340     52.77941
## 2021-12-13     164.8694     37.09962     133.6638     191.9975     50.13306
## 2021-12-20     169.8387     37.32074     135.0012     199.7643     50.62176
## 2021-12-27     171.0750     37.72171     137.0669     201.2943     51.83763
## 2022-01-03     163.2626     39.76960     130.7044     196.4106     57.29226
##            XLU.Adjusted
## 2021-11-29     63.42516
## 2021-12-06     65.07195
## 2021-12-13     65.88126
## 2021-12-20     65.60835
## 2021-12-27     67.88812
## 2022-01-03     66.77845
```

```
# Google Trends data
gtrends_war_web <- gtrends(
  keyword = "war",
  time = "2021-11-30 2023-12-01",
  gprop = "web"
)$interest_over_time
gtrends_war_web
```

```
# Too many requests exclude?
gtrends_war_news <- gtrends(
  keyword = "war",
  time = "2021-12-01 2023-12-01",
  gprop = "news"
)$interest_over_time
```

```
# If gtrends does not work --> response 429
library(lubridate)

gtrends_war_web <- read.csv('gtrends_war_web.csv')
gtrends_war_web <- gtrends_war_web %>%
  mutate(date = ymd(date))
head(gtrends_war_web)
```

```
##   X       date hits keyword   geo                  time gprop category
## 1 1 2021-12-05   33     war world 2021-11-30 2023-12-01   web        0
## 2 2 2021-12-12   35     war world 2021-11-30 2023-12-01   web        0
## 3 3 2021-12-19   31     war world 2021-11-30 2023-12-01   web        0
## 4 4 2021-12-26   31     war world 2021-11-30 2023-12-01   web        0
## 5 5 2022-01-02   34     war world 2021-11-30 2023-12-01   web        0
## 6 6 2022-01-09   39     war world 2021-11-30 2023-12-01   web        0
```

Google Trends delivers data always from Sunday weekly. This has to be put to the weekly data from the financial tickers, which always starts at Monday.

```
# Increase date by one day to match ticker data
gtrends_war_web$date <- gtrends_war_web$date + days(1)

# Check length and min max date
length(gtrends_war_web$date)
```

```
## [1] 104
```

```
min(gtrends_war_web$date)
```

```
## [1] "2021-12-06"
```

```
max(gtrends_war_web$date)
```

```
## [1] "2023-11-27"
```

```
# Remove first date of ticker data to match gtrends data
ticker_data <- ticker_data[-1,]
ticker_data
```

```
##            XLK.Adjusted XLF.Adjusted XLV.Adjusted XLY.Adjusted XLE.Adjusted
## 2021-12-06     171.7439     37.54185     130.4269     201.5340     52.77941
## 2021-12-13     164.8694     37.09962     133.6638     191.9975     50.13306
## 2021-12-20     169.8387     37.32074     135.0012     199.7643     50.62176
## 2021-12-27     171.0750     37.72171     137.0669     201.2943     51.83763
## 2022-01-03     163.2626     39.76960     130.7044     196.4106     57.29226
## 2022-01-10     163.0954     39.44117     130.4320     193.4961     60.25307
## 2022-01-17     151.8491     36.90063     125.9568     177.6438     58.32901
## 2022-01-24     155.4601     37.40294     126.9394     175.3596     61.29916
## 2022-01-31     157.0344     38.73599     128.7100     181.5626     64.35338
## 2022-02-07     152.4001     38.74566     126.7156     177.7029     65.76375
## 2022-02-14     149.9895     37.88593     124.0500     177.2303     63.55947
## 2022-02-21     151.8688     37.77001     127.3966     173.4789     64.33471
## 2022-02-28     147.3427     35.96362     128.9240     168.9596     70.26569
## 2022-03-07     141.7442     35.16186     125.4412     164.1546     71.77878
## 2022-03-14     152.5772     37.67341     133.2727     179.0814     69.00475
## 2022-03-21     155.6766     38.17572     132.5723     180.5386     73.55341
## 2022-03-28     156.2008     37.06440     134.7424     182.6600     72.66339
## 2022-04-04     150.2371     36.74438     139.3877     176.9898     74.99246
## 2022-04-11     144.5493     35.77461     135.2987     175.9248     75.29420
## 2022-04-18     141.0401     35.07638     130.4485     173.3806     71.85246
## 2022-04-25     139.4037     33.46657     127.1499     160.6201     70.86237
## 2022-05-02     138.5659     33.68962     126.6522     155.9557     78.18905
## 2022-05-09     133.9230     32.51620     125.5006     150.2066     76.17114
## 2022-05-16     129.1915     31.92464     126.6522     138.4619     77.10464
## 2022-05-23     139.4432     34.49453     130.7901     151.6266     83.47897
## 2022-05-30     137.9942     33.81569     126.6912     150.8772     84.40307
## 2022-06-06     129.2210     31.54644     122.4656     142.4655     83.64870
## 2022-06-13     122.8433     29.99482     117.0006     134.2117     69.29707
## 2022-06-20     131.5474     31.36218     126.0862     144.9506     67.50547
## 2022-06-27     125.9199     31.08089     127.0889     138.3867     69.20649
## 2022-07-04     131.3160     31.27588     128.1277     144.6730     67.65225
## 2022-07-11     130.8317     30.98340     127.5985     143.1607     65.40194
## 2022-07-18     135.6151     31.92908     127.2359     152.9558     67.84296
## 2022-07-25     142.5333     32.86502     129.7545     160.9421     74.77505
## 2022-08-01     145.3005     32.84552     128.8235     162.4050     69.68324
## 2022-08-08     148.8485     34.66865     130.9502     167.9400     74.82272
## 2022-08-15     146.3778     34.08369     130.2641     165.5876     75.76671
```

```
## 2022-08-22     138.2440     32.87477     124.7466     157.8286     78.98959
## 2022-08-29     131.2962     32.06558     122.4926     153.6872     76.24346
## 2022-09-05     135.5954     33.49873     127.8925     162.5730     76.85371
## 2022-09-12     127.2047     32.23131     124.9034     156.1088     74.79412
## 2022-09-19     122.2928     30.27170     120.2778     144.7224     67.20410
## 2022-09-26     117.6806     29.75470     119.1632     141.0932     69.42124
## 2022-10-03     119.6522     30.27413     120.6490     139.3103     78.84834
## 2022-10-10     115.7190     30.39174     121.9084     133.9816     77.38319
## 2022-10-17     123.1892     31.54821     124.5059     141.0437     83.79324
## 2022-10-24     128.3807     33.50834     130.7243     143.5793     86.02953
## 2022-10-31     119.8702     33.23392     128.7073     136.1904     88.12122
## 2022-11-07     131.9077     35.15484     130.9900     143.7081     89.76951
## 2022-11-14     130.8674     34.66481     132.4069     139.6471     88.32364
## 2022-11-21     132.3635     35.39006     134.9454     141.2715     88.55499
## 2022-11-28     134.1072     35.21364     137.5331     144.5698     87.05128
## 2022-12-05     129.6984     33.84156     135.7620     138.1020     79.69659
## 2022-12-12     126.3001     33.02810     133.3514     132.6346     81.31597
## 2022-12-19     123.4468     33.27312     133.9123     128.1972     83.91855
## 2022-12-26     123.6423     33.72831     134.2169     128.2804     85.18214
## 2023-01-02     123.9304     34.89204     134.0391     131.1706     85.26978
## 2023-01-09     129.6535     35.62183     133.8217     138.7486     87.59727
## 2023-01-16     130.4980     34.86245     132.3002     138.0335     88.11341
## 2023-01-23     135.8237     35.75004     131.2628     146.8829     88.84379
## 2023-01-30     140.8612     36.08535     131.0949     150.3193     83.71163
## 2023-02-06     139.4801     35.98673     130.8973     147.1411     87.85046
## 2023-02-13     138.9237     35.88811     130.4033     149.5446     82.28008
## 2023-02-20     135.2176     35.16818     126.9651     142.8803     82.44563
## 2023-02-27     139.2416     35.49363     127.6172     145.3136     84.97763
## 2023-03-06     134.8996     32.47583     122.6575     137.2589     80.46873
## 2023-03-13     142.5403     30.55272     124.3470     140.3775     74.95678
## 2023-03-20     145.0243     30.56259     125.7697     140.6158     75.85271
## 2023-03-27     150.3732     31.86395     128.4026     148.9748     81.50535
## 2023-04-03     148.4413     31.70538     132.4294     144.3822     83.62096
## 2023-04-10     148.0231     32.58746     133.4709     146.3448     85.83498
## 2023-04-17     147.1369     32.92443     133.2031     146.8130     83.62096
## 2023-04-24     150.1939     32.87487     132.4393     147.3012     83.76856
## 2023-05-01     150.6022     32.04235     132.4989     146.6337     78.94692
## 2023-05-08     150.3134     31.61618     131.1103     147.2713     77.26427
## 2023-05-15     156.8159     32.30995     130.2375     150.9772     78.36636
## 2023-05-22     164.0851     31.82431     126.4884     151.3957     77.51027
## 2023-05-29     166.1961     32.50817     129.2556     156.4066     78.62221
## 2023-06-05     165.3099     32.85506     129.3548     160.5908     80.02934
## 2023-06-12     172.4596     33.28123     131.1301     165.7512     79.56685
## 2023-06-19     167.9388     32.44870     130.9219     164.8247     76.12283
## 2023-06-26     173.4894     33.55998     132.1878     169.5085     80.57540
## 2023-07-03     170.8451     33.44052     128.4730     169.0693     80.19819
## 2023-07-10     175.6149     34.09758     131.1620     174.6094     80.85336
## 2023-07-17     175.7546     35.10308     135.6835     170.6764     83.68248
## 2023-07-24     177.5607     35.04335     134.6378     172.4533     85.22112
## 2023-07-31     170.7952     34.75464     131.8591     171.2454     86.28329
## 2023-08-07     166.5443     34.76460     135.1158     169.4187     89.24146
## 2023-08-14     164.6084     33.82878     132.9845     162.3712     88.15944
## 2023-08-21     168.3504     33.83874     132.8650     164.3677     86.94837
## 2023-08-28     175.8244     34.53562     133.0244     169.3688     90.07530
```

```
## 2023-09-04     172.4217     34.14736     131.5205     168.4504     91.37571
## 2023-09-11     168.5400     34.67500     131.6500     171.4850     91.33600
## 2023-09-18     164.0596     33.51020     129.6183     160.6443     88.65578
## 2023-09-25     163.9300     33.17000     128.7400     160.9800     90.39000
## 2023-10-02     168.2200     33.04000     130.0100     160.6100     85.73000
## 2023-10-09     168.6300     33.21000     130.1900     158.9700     89.59000
## 2023-10-16     163.9400     32.20000     128.0600     151.6900     90.26000
## 2023-10-23     161.1200     31.45000     123.1400     149.5900     84.63000
## 2023-10-30     171.7600     33.78000     127.4100     160.2200     86.68000
## 2023-11-06     179.5200     33.91000     126.2700     161.4000     83.41000
## 2023-11-13     182.8000     35.01000     128.2500     167.1400     84.69000
## 2023-11-20     184.4100     35.38000     131.1300     168.2400     84.93000
## 2023-11-27     185.1600     35.90000     131.3100     168.7800     84.58000
##            XLU.Adjusted
## 2021-12-06     65.07195
## 2021-12-13     65.88126
## 2021-12-20     65.60835
## 2021-12-27     67.88812
## 2022-01-03     66.77845
## 2022-01-10     65.82055
## 2022-01-17     65.28942
## 2022-01-24     64.45483
## 2022-01-31     64.94799
## 2022-02-07     63.57278
## 2022-02-14     62.80456
## 2022-02-21     64.09442
## 2022-02-28     67.23369
## 2022-03-07     66.79742
## 2022-03-14     67.15782
## 2022-03-21     69.02621
## 2022-03-28     72.08437
## 2022-04-04     73.47832
## 2022-04-11     72.64767
## 2022-04-18     70.89094
## 2022-04-25     68.03619
## 2022-05-02     68.94322
## 2022-05-09     68.19850
## 2022-05-16     68.49450
## 2022-05-23     71.96026
## 2022-05-30     70.98640
## 2022-06-06     68.11257
## 2022-06-13     61.93528
## 2022-06-20     65.88799
## 2022-06-27     69.14847
## 2022-07-04     67.19479
## 2022-07-11     67.13706
## 2022-07-18     66.81947
## 2022-07-25     71.16952
## 2022-08-01     71.43900
## 2022-08-08     73.83537
## 2022-08-15     74.77852
## 2022-08-22     72.86333
## 2022-08-29     71.79507
## 2022-09-05     74.47055
```

```
## 2022-09-12     71.66998
## 2022-09-19     69.06186
## 2022-09-26     63.47388
## 2022-10-03     61.77827
## 2022-10-10     60.21831
## 2022-10-17     61.34226
## 2022-10-24     65.31483
## 2022-10-31     65.00476
## 2022-11-07     65.96400
## 2022-11-14     66.71007
## 2022-11-21     68.73511
## 2022-11-28     68.82231
## 2022-12-05     68.62854
## 2022-12-12     68.28941
## 2022-12-19     68.70604
## 2022-12-26     68.86770
## 2023-01-02     69.35612
## 2023-01-09     69.67847
## 2023-01-16     67.62711
## 2023-01-23     67.29498
## 2023-01-30     66.33767
## 2023-02-06     66.14229
## 2023-02-13     66.89447
## 2023-02-20     65.06777
## 2023-02-27     64.71609
## 2023-03-06     62.92846
## 2023-03-13     65.41943
## 2023-03-20     64.16907
## 2023-03-27     66.61012
## 2023-04-03     68.69630
## 2023-04-10     67.79098
## 2023-04-17     68.50934
## 2023-04-24     67.87955
## 2023-05-01     67.92875
## 2023-05-08     67.92875
## 2023-05-15     65.05534
## 2023-05-22     63.55957
## 2023-05-29     64.08112
## 2023-06-05     65.33086
## 2023-06-12     66.22635
## 2023-06-19     63.99256
## 2023-06-26     64.91399
## 2023-07-03     64.86439
## 2023-07-10     66.31264
## 2023-07-17     67.90971
## 2023-07-24     66.50112
## 2023-07-31     63.47564
## 2023-08-07     64.05097
## 2023-08-14     62.97966
## 2023-08-21     63.11853
## 2023-08-28     62.13649
## 2023-09-04     62.67215
## 2023-09-11     64.41801
## 2023-09-18     62.78127
```

```
## 2023-09-25     58.93000
## 2023-10-02     57.25000
## 2023-10-09     59.30000
## 2023-10-16     58.03000
## 2023-10-23     58.73000
## 2023-10-30     61.86000
## 2023-11-06     60.32000
## 2023-11-13     62.29000
## 2023-11-20     62.69000
## 2023-11-27     62.76000
```

Visualize Ticker Data

```
# Visualization
ticker_data_df <- fortify.zoo(ticker_data)
colnames(ticker_data_df)[1] <- "Date"
ticker_data_df <- gather(ticker_data_df, key="Ticker", value="Value",
                         c("XLK.Adjusted", "XLF.Adjusted", "XLV.Adjusted",
                           "XLY.Adjusted", "XLE.Adjusted", "XLU.Adjusted"), -Date)

ticker_data_df
```

```
##           Date       Ticker     Value
## 1   2021-12-06 XLK.Adjusted 171.74388
## 2   2021-12-13 XLK.Adjusted 164.86940
## 3   2021-12-20 XLK.Adjusted 169.83865
## 4   2021-12-27 XLK.Adjusted 171.07501
## 5   2022-01-03 XLK.Adjusted 163.26263
## 6   2022-01-10 XLK.Adjusted 163.09535
## 7   2022-01-17 XLK.Adjusted 151.84912
## 8   2022-01-24 XLK.Adjusted 155.46011
## 9   2022-01-31 XLK.Adjusted 157.03441
## 10  2022-02-07 XLK.Adjusted 152.40010
## 11  2022-02-14 XLK.Adjusted 149.98949
## 12  2022-02-21 XLK.Adjusted 151.86879
## 13  2022-02-28 XLK.Adjusted 147.34274
## 14  2022-03-07 XLK.Adjusted 141.74420
## 15  2022-03-14 XLK.Adjusted 152.57722
## 16  2022-03-21 XLK.Adjusted 155.67659
## 17  2022-03-28 XLK.Adjusted 156.20084
## 18  2022-04-04 XLK.Adjusted 150.23708
## 19  2022-04-11 XLK.Adjusted 144.54932
## 20  2022-04-18 XLK.Adjusted 141.04010
## 21  2022-04-25 XLK.Adjusted 139.40373
## 22  2022-05-02 XLK.Adjusted 138.56587
## 23  2022-05-09 XLK.Adjusted 133.92302
## 24  2022-05-16 XLK.Adjusted 129.19147
## 25  2022-05-23 XLK.Adjusted 139.44319
## 26  2022-05-30 XLK.Adjusted 137.99416
## 27  2022-06-06 XLK.Adjusted 129.22102
## 28  2022-06-13 XLK.Adjusted 122.84328
## 29  2022-06-20 XLK.Adjusted 131.54738
```

```
## 30  2022-06-27 XLK.Adjusted 125.91987
## 31  2022-07-04 XLK.Adjusted 131.31601
## 32  2022-07-11 XLK.Adjusted 130.83174
## 33  2022-07-18 XLK.Adjusted 135.61514
## 34  2022-07-25 XLK.Adjusted 142.53326
## 35  2022-08-01 XLK.Adjusted 145.30052
## 36  2022-08-08 XLK.Adjusted 148.84854
## 37  2022-08-15 XLK.Adjusted 146.37778
## 38  2022-08-22 XLK.Adjusted 138.24405
## 39  2022-08-29 XLK.Adjusted 131.29625
## 40  2022-09-05 XLK.Adjusted 135.59537
## 41  2022-09-12 XLK.Adjusted 127.20467
## 42  2022-09-19 XLK.Adjusted 122.29279
## 43  2022-09-26 XLK.Adjusted 117.68063
## 44  2022-10-03 XLK.Adjusted 119.65222
## 45  2022-10-10 XLK.Adjusted 115.71897
## 46  2022-10-17 XLK.Adjusted 123.18918
## 47  2022-10-24 XLK.Adjusted 128.38068
## 48  2022-10-31 XLK.Adjusted 119.87019
## 49  2022-11-07 XLK.Adjusted 131.90773
## 50  2022-11-14 XLK.Adjusted 130.86745
## 51  2022-11-21 XLK.Adjusted 132.36348
## 52  2022-11-28 XLK.Adjusted 134.10718
## 53  2022-12-05 XLK.Adjusted 129.69838
## 54  2022-12-12 XLK.Adjusted 126.30012
## 55  2022-12-19 XLK.Adjusted 123.44678
## 56  2022-12-26 XLK.Adjusted 123.64227
## 57  2023-01-02 XLK.Adjusted 123.93041
## 58  2023-01-09 XLK.Adjusted 129.65347
## 59  2023-01-16 XLK.Adjusted 130.49802
## 60  2023-01-23 XLK.Adjusted 135.82367
## 61  2023-01-30 XLK.Adjusted 140.86118
## 62  2023-02-06 XLK.Adjusted 139.48009
## 63  2023-02-13 XLK.Adjusted 138.92368
## 64  2023-02-20 XLK.Adjusted 135.21759
## 65  2023-02-27 XLK.Adjusted 139.24161
## 66  2023-03-06 XLK.Adjusted 134.89964
## 67  2023-03-13 XLK.Adjusted 142.54033
## 68  2023-03-20 XLK.Adjusted 145.02431
## 69  2023-03-27 XLK.Adjusted 150.37315
## 70  2023-04-03 XLK.Adjusted 148.44135
## 71  2023-04-10 XLK.Adjusted 148.02310
## 72  2023-04-17 XLK.Adjusted 147.13686
## 73  2023-04-24 XLK.Adjusted 150.19392
## 74  2023-05-01 XLK.Adjusted 150.60220
## 75  2023-05-08 XLK.Adjusted 150.31340
## 76  2023-05-15 XLK.Adjusted 156.81587
## 77  2023-05-22 XLK.Adjusted 164.08508
## 78  2023-05-29 XLK.Adjusted 166.19614
## 79  2023-06-05 XLK.Adjusted 165.30989
## 80  2023-06-12 XLK.Adjusted 172.45963
## 81  2023-06-19 XLK.Adjusted 167.93877
## 82  2023-06-26 XLK.Adjusted 173.48943
## 83  2023-07-03 XLK.Adjusted 170.84508
```

```
## 84  2023-07-10 XLK.Adjusted 175.61488
## 85  2023-07-17 XLK.Adjusted 175.75459
## 86  2023-07-24 XLK.Adjusted 177.56073
## 87  2023-07-31 XLK.Adjusted 170.79518
## 88  2023-08-07 XLK.Adjusted 166.54425
## 89  2023-08-14 XLK.Adjusted 164.60840
## 90  2023-08-21 XLK.Adjusted 168.35040
## 91  2023-08-28 XLK.Adjusted 175.82443
## 92  2023-09-04 XLK.Adjusted 172.42169
## 93  2023-09-11 XLK.Adjusted 168.53999
## 94  2023-09-18 XLK.Adjusted 164.05957
## 95  2023-09-25 XLK.Adjusted 163.92999
## 96  2023-10-02 XLK.Adjusted 168.22000
## 97  2023-10-09 XLK.Adjusted 168.63000
## 98  2023-10-16 XLK.Adjusted 163.94000
## 99  2023-10-23 XLK.Adjusted 161.12000
## 100 2023-10-30 XLK.Adjusted 171.75999
## 101 2023-11-06 XLK.Adjusted 179.52000
## 102 2023-11-13 XLK.Adjusted 182.80000
## 103 2023-11-20 XLK.Adjusted 184.41000
## 104 2023-11-27 XLK.Adjusted 185.16000
## 105 2021-12-06 XLF.Adjusted  37.54185
## 106 2021-12-13 XLF.Adjusted  37.09962
## 107 2021-12-20 XLF.Adjusted  37.32074
## 108 2021-12-27 XLF.Adjusted  37.72171
## 109 2022-01-03 XLF.Adjusted  39.76960
## 110 2022-01-10 XLF.Adjusted  39.44117
## 111 2022-01-17 XLF.Adjusted  36.90063
## 112 2022-01-24 XLF.Adjusted  37.40294
## 113 2022-01-31 XLF.Adjusted  38.73599
## 114 2022-02-07 XLF.Adjusted  38.74566
## 115 2022-02-14 XLF.Adjusted  37.88593
## 116 2022-02-21 XLF.Adjusted  37.77001
## 117 2022-02-28 XLF.Adjusted  35.96362
## 118 2022-03-07 XLF.Adjusted  35.16186
## 119 2022-03-14 XLF.Adjusted  37.67341
## 120 2022-03-21 XLF.Adjusted  38.17572
## 121 2022-03-28 XLF.Adjusted  37.06440
## 122 2022-04-04 XLF.Adjusted  36.74438
## 123 2022-04-11 XLF.Adjusted  35.77461
## 124 2022-04-18 XLF.Adjusted  35.07638
## 125 2022-04-25 XLF.Adjusted  33.46657
## 126 2022-05-02 XLF.Adjusted  33.68962
## 127 2022-05-09 XLF.Adjusted  32.51620
## 128 2022-05-16 XLF.Adjusted  31.92464
## 129 2022-05-23 XLF.Adjusted  34.49453
## 130 2022-05-30 XLF.Adjusted  33.81569
## 131 2022-06-06 XLF.Adjusted  31.54644
## 132 2022-06-13 XLF.Adjusted  29.99482
## 133 2022-06-20 XLF.Adjusted  31.36218
## 134 2022-06-27 XLF.Adjusted  31.08089
## 135 2022-07-04 XLF.Adjusted  31.27588
## 136 2022-07-11 XLF.Adjusted  30.98340
## 137 2022-07-18 XLF.Adjusted  31.92908
```

```
## 138 2022-07-25 XLF.Adjusted  32.86502
## 139 2022-08-01 XLF.Adjusted  32.84552
## 140 2022-08-08 XLF.Adjusted  34.66865
## 141 2022-08-15 XLF.Adjusted  34.08369
## 142 2022-08-22 XLF.Adjusted  32.87477
## 143 2022-08-29 XLF.Adjusted  32.06558
## 144 2022-09-05 XLF.Adjusted  33.49873
## 145 2022-09-12 XLF.Adjusted  32.23131
## 146 2022-09-19 XLF.Adjusted  30.27170
## 147 2022-09-26 XLF.Adjusted  29.75470
## 148 2022-10-03 XLF.Adjusted  30.27413
## 149 2022-10-10 XLF.Adjusted  30.39174
## 150 2022-10-17 XLF.Adjusted  31.54821
## 151 2022-10-24 XLF.Adjusted  33.50834
## 152 2022-10-31 XLF.Adjusted  33.23392
## 153 2022-11-07 XLF.Adjusted  35.15484
## 154 2022-11-14 XLF.Adjusted  34.66481
## 155 2022-11-21 XLF.Adjusted  35.39006
## 156 2022-11-28 XLF.Adjusted  35.21364
## 157 2022-12-05 XLF.Adjusted  33.84156
## 158 2022-12-12 XLF.Adjusted  33.02810
## 159 2022-12-19 XLF.Adjusted  33.27312
## 160 2022-12-26 XLF.Adjusted  33.72831
## 161 2023-01-02 XLF.Adjusted  34.89204
## 162 2023-01-09 XLF.Adjusted  35.62183
## 163 2023-01-16 XLF.Adjusted  34.86245
## 164 2023-01-23 XLF.Adjusted  35.75004
## 165 2023-01-30 XLF.Adjusted  36.08535
## 166 2023-02-06 XLF.Adjusted  35.98673
## 167 2023-02-13 XLF.Adjusted  35.88811
## 168 2023-02-20 XLF.Adjusted  35.16818
## 169 2023-02-27 XLF.Adjusted  35.49363
## 170 2023-03-06 XLF.Adjusted  32.47583
## 171 2023-03-13 XLF.Adjusted  30.55272
## 172 2023-03-20 XLF.Adjusted  30.56259
## 173 2023-03-27 XLF.Adjusted  31.86395
## 174 2023-04-03 XLF.Adjusted  31.70538
## 175 2023-04-10 XLF.Adjusted  32.58746
## 176 2023-04-17 XLF.Adjusted  32.92443
## 177 2023-04-24 XLF.Adjusted  32.87487
## 178 2023-05-01 XLF.Adjusted  32.04235
## 179 2023-05-08 XLF.Adjusted  31.61618
## 180 2023-05-15 XLF.Adjusted  32.30995
## 181 2023-05-22 XLF.Adjusted  31.82431
## 182 2023-05-29 XLF.Adjusted  32.50817
## 183 2023-06-05 XLF.Adjusted  32.85506
## 184 2023-06-12 XLF.Adjusted  33.28123
## 185 2023-06-19 XLF.Adjusted  32.44870
## 186 2023-06-26 XLF.Adjusted  33.55998
## 187 2023-07-03 XLF.Adjusted  33.44052
## 188 2023-07-10 XLF.Adjusted  34.09758
## 189 2023-07-17 XLF.Adjusted  35.10308
## 190 2023-07-24 XLF.Adjusted  35.04335
## 191 2023-07-31 XLF.Adjusted  34.75464
```

```
## 192 2023-08-07 XLF.Adjusted  34.76460
## 193 2023-08-14 XLF.Adjusted  33.82878
## 194 2023-08-21 XLF.Adjusted  33.83874
## 195 2023-08-28 XLF.Adjusted  34.53562
## 196 2023-09-04 XLF.Adjusted  34.14736
## 197 2023-09-11 XLF.Adjusted  34.67500
## 198 2023-09-18 XLF.Adjusted  33.51020
## 199 2023-09-25 XLF.Adjusted  33.17000
## 200 2023-10-02 XLF.Adjusted  33.04000
## 201 2023-10-09 XLF.Adjusted  33.21000
## 202 2023-10-16 XLF.Adjusted  32.20000
## 203 2023-10-23 XLF.Adjusted  31.45000
## 204 2023-10-30 XLF.Adjusted  33.78000
## 205 2023-11-06 XLF.Adjusted  33.91000
## 206 2023-11-13 XLF.Adjusted  35.01000
## 207 2023-11-20 XLF.Adjusted  35.38000
## 208 2023-11-27 XLF.Adjusted  35.90000
## 209 2021-12-06 XLV.Adjusted 130.42690
## 210 2021-12-13 XLV.Adjusted 133.66380
## 211 2021-12-20 XLV.Adjusted 135.00124
## 212 2021-12-27 XLV.Adjusted 137.06689
## 213 2022-01-03 XLV.Adjusted 130.70436
## 214 2022-01-10 XLV.Adjusted 130.43196
## 215 2022-01-17 XLV.Adjusted 125.95676
## 216 2022-01-24 XLV.Adjusted 126.93938
## 217 2022-01-31 XLV.Adjusted 128.70999
## 218 2022-02-07 XLV.Adjusted 126.71561
## 219 2022-02-14 XLV.Adjusted 124.04996
## 220 2022-02-21 XLV.Adjusted 127.39661
## 221 2022-02-28 XLV.Adjusted 128.92401
## 222 2022-03-07 XLV.Adjusted 125.44116
## 223 2022-03-14 XLV.Adjusted 133.27271
## 224 2022-03-21 XLV.Adjusted 132.57225
## 225 2022-03-28 XLV.Adjusted 134.74243
## 226 2022-04-04 XLV.Adjusted 139.38770
## 227 2022-04-11 XLV.Adjusted 135.29868
## 228 2022-04-18 XLV.Adjusted 130.44846
## 229 2022-04-25 XLV.Adjusted 127.14992
## 230 2022-05-02 XLV.Adjusted 126.65221
## 231 2022-05-09 XLV.Adjusted 125.50065
## 232 2022-05-16 XLV.Adjusted 126.65221
## 233 2022-05-23 XLV.Adjusted 130.79005
## 234 2022-05-30 XLV.Adjusted 126.69124
## 235 2022-06-06 XLV.Adjusted 122.46560
## 236 2022-06-13 XLV.Adjusted 117.00056
## 237 2022-06-20 XLV.Adjusted 126.08620
## 238 2022-06-27 XLV.Adjusted 127.08888
## 239 2022-07-04 XLV.Adjusted 128.12772
## 240 2022-07-11 XLV.Adjusted 127.59848
## 241 2022-07-18 XLV.Adjusted 127.23589
## 242 2022-07-25 XLV.Adjusted 129.75453
## 243 2022-08-01 XLV.Adjusted 128.82353
## 244 2022-08-08 XLV.Adjusted 130.95016
## 245 2022-08-15 XLV.Adjusted 130.26414
```

```
## 246 2022-08-22 XLV.Adjusted 124.74664
## 247 2022-08-29 XLV.Adjusted 122.49260
## 248 2022-09-05 XLV.Adjusted 127.89252
## 249 2022-09-12 XLV.Adjusted 124.90343
## 250 2022-09-19 XLV.Adjusted 120.27776
## 251 2022-09-26 XLV.Adjusted 119.16323
## 252 2022-10-03 XLV.Adjusted 120.64896
## 253 2022-10-10 XLV.Adjusted 121.90839
## 254 2022-10-17 XLV.Adjusted 124.50594
## 255 2022-10-24 XLV.Adjusted 130.72435
## 256 2022-10-31 XLV.Adjusted 128.70731
## 257 2022-11-07 XLV.Adjusted 130.99002
## 258 2022-11-14 XLV.Adjusted 132.40688
## 259 2022-11-21 XLV.Adjusted 134.94539
## 260 2022-11-28 XLV.Adjusted 137.53311
## 261 2022-12-05 XLV.Adjusted 135.76204
## 262 2022-12-12 XLV.Adjusted 133.35144
## 263 2022-12-19 XLV.Adjusted 133.91228
## 264 2022-12-26 XLV.Adjusted 134.21690
## 265 2023-01-02 XLV.Adjusted 134.03906
## 266 2023-01-09 XLV.Adjusted 133.82170
## 267 2023-01-16 XLV.Adjusted 132.30022
## 268 2023-01-23 XLV.Adjusted 131.26285
## 269 2023-01-30 XLV.Adjusted 131.09489
## 270 2023-02-06 XLV.Adjusted 130.89729
## 271 2023-02-13 XLV.Adjusted 130.40331
## 272 2023-02-20 XLV.Adjusted 126.96513
## 273 2023-02-27 XLV.Adjusted 127.61719
## 274 2023-03-06 XLV.Adjusted 122.65755
## 275 2023-03-13 XLV.Adjusted 124.34700
## 276 2023-03-20 XLV.Adjusted 125.76968
## 277 2023-03-27 XLV.Adjusted 128.40260
## 278 2023-04-03 XLV.Adjusted 132.42944
## 279 2023-04-10 XLV.Adjusted 133.47087
## 280 2023-04-17 XLV.Adjusted 133.20306
## 281 2023-04-24 XLV.Adjusted 132.43935
## 282 2023-05-01 XLV.Adjusted 132.49886
## 283 2023-05-08 XLV.Adjusted 131.11031
## 284 2023-05-15 XLV.Adjusted 130.23747
## 285 2023-05-22 XLV.Adjusted 126.48836
## 286 2023-05-29 XLV.Adjusted 129.25557
## 287 2023-06-05 XLV.Adjusted 129.35475
## 288 2023-06-12 XLV.Adjusted 131.13014
## 289 2023-06-19 XLV.Adjusted 130.92186
## 290 2023-06-26 XLV.Adjusted 132.18779
## 291 2023-07-03 XLV.Adjusted 128.47304
## 292 2023-07-10 XLV.Adjusted 131.16200
## 293 2023-07-17 XLV.Adjusted 135.68347
## 294 2023-07-24 XLV.Adjusted 134.63776
## 295 2023-07-31 XLV.Adjusted 131.85915
## 296 2023-08-07 XLV.Adjusted 135.11578
## 297 2023-08-14 XLV.Adjusted 132.98453
## 298 2023-08-21 XLV.Adjusted 132.86502
## 299 2023-08-28 XLV.Adjusted 133.02437
```

```
## 300 2023-09-04 XLV.Adjusted 131.52054
## 301 2023-09-11 XLV.Adjusted 131.65001
## 302 2023-09-18 XLV.Adjusted 129.61833
## 303 2023-09-25 XLV.Adjusted 128.74001
## 304 2023-10-02 XLV.Adjusted 130.00999
## 305 2023-10-09 XLV.Adjusted 130.19000
## 306 2023-10-16 XLV.Adjusted 128.06000
## 307 2023-10-23 XLV.Adjusted 123.14000
## 308 2023-10-30 XLV.Adjusted 127.41000
## 309 2023-11-06 XLV.Adjusted 126.27000
## 310 2023-11-13 XLV.Adjusted 128.25000
## 311 2023-11-20 XLV.Adjusted 131.13000
## 312 2023-11-27 XLV.Adjusted 131.31000
## 313 2021-12-06 XLY.Adjusted 201.53398
## 314 2021-12-13 XLY.Adjusted 191.99751
## 315 2021-12-20 XLY.Adjusted 199.76433
## 316 2021-12-27 XLY.Adjusted 201.29425
## 317 2022-01-03 XLY.Adjusted 196.41057
## 318 2022-01-10 XLY.Adjusted 193.49612
## 319 2022-01-17 XLY.Adjusted 177.64384
## 320 2022-01-24 XLY.Adjusted 175.35956
## 321 2022-01-31 XLY.Adjusted 181.56259
## 322 2022-02-07 XLY.Adjusted 177.70291
## 323 2022-02-14 XLY.Adjusted 177.23029
## 324 2022-02-21 XLY.Adjusted 173.47894
## 325 2022-02-28 XLY.Adjusted 168.95958
## 326 2022-03-07 XLY.Adjusted 164.15465
## 327 2022-03-14 XLY.Adjusted 179.08138
## 328 2022-03-21 XLY.Adjusted 180.53859
## 329 2022-03-28 XLY.Adjusted 182.66002
## 330 2022-04-04 XLY.Adjusted 176.98979
## 331 2022-04-11 XLY.Adjusted 175.92477
## 332 2022-04-18 XLY.Adjusted 173.38058
## 333 2022-04-25 XLY.Adjusted 160.62010
## 334 2022-05-02 XLY.Adjusted 155.95573
## 335 2022-05-09 XLY.Adjusted 150.20665
## 336 2022-05-16 XLY.Adjusted 138.46188
## 337 2022-05-23 XLY.Adjusted 151.62665
## 338 2022-05-30 XLY.Adjusted 150.87721
## 339 2022-06-06 XLY.Adjusted 142.46555
## 340 2022-06-13 XLY.Adjusted 134.21169
## 341 2022-06-20 XLY.Adjusted 144.95059
## 342 2022-06-27 XLY.Adjusted 138.38669
## 343 2022-07-04 XLY.Adjusted 144.67299
## 344 2022-07-11 XLY.Adjusted 143.16071
## 345 2022-07-18 XLY.Adjusted 152.95580
## 346 2022-07-25 XLY.Adjusted 160.94211
## 347 2022-08-01 XLY.Adjusted 162.40498
## 348 2022-08-08 XLY.Adjusted 167.94005
## 349 2022-08-15 XLY.Adjusted 165.58765
## 350 2022-08-22 XLY.Adjusted 157.82864
## 351 2022-08-29 XLY.Adjusted 153.68724
## 352 2022-09-05 XLY.Adjusted 162.57298
## 353 2022-09-12 XLY.Adjusted 156.10884
```

```
## 354 2022-09-19 XLY.Adjusted 144.72240
## 355 2022-09-26 XLY.Adjusted 141.09320
## 356 2022-10-03 XLY.Adjusted 139.31035
## 357 2022-10-10 XLY.Adjusted 133.98161
## 358 2022-10-17 XLY.Adjusted 141.04369
## 359 2022-10-24 XLY.Adjusted 143.57932
## 360 2022-10-31 XLY.Adjusted 136.19035
## 361 2022-11-07 XLY.Adjusted 143.70805
## 362 2022-11-14 XLY.Adjusted 139.64713
## 363 2022-11-21 XLY.Adjusted 141.27150
## 364 2022-11-28 XLY.Adjusted 144.56979
## 365 2022-12-05 XLY.Adjusted 138.10197
## 366 2022-12-12 XLY.Adjusted 132.63457
## 367 2022-12-19 XLY.Adjusted 128.19722
## 368 2022-12-26 XLY.Adjusted 128.28041
## 369 2023-01-02 XLY.Adjusted 131.17059
## 370 2023-01-09 XLY.Adjusted 138.74863
## 371 2023-01-16 XLY.Adjusted 138.03352
## 372 2023-01-23 XLY.Adjusted 146.88286
## 373 2023-01-30 XLY.Adjusted 150.31931
## 374 2023-02-06 XLY.Adjusted 147.14108
## 375 2023-02-13 XLY.Adjusted 149.54462
## 376 2023-02-20 XLY.Adjusted 142.88031
## 377 2023-02-27 XLY.Adjusted 145.31361
## 378 2023-03-06 XLY.Adjusted 137.25885
## 379 2023-03-13 XLY.Adjusted 140.37746
## 380 2023-03-20 XLY.Adjusted 140.61583
## 381 2023-03-27 XLY.Adjusted 148.97484
## 382 2023-04-03 XLY.Adjusted 144.38225
## 383 2023-04-10 XLY.Adjusted 146.34482
## 384 2023-04-17 XLY.Adjusted 146.81303
## 385 2023-04-24 XLY.Adjusted 147.30119
## 386 2023-05-01 XLY.Adjusted 146.63373
## 387 2023-05-08 XLY.Adjusted 147.27132
## 388 2023-05-15 XLY.Adjusted 150.97725
## 389 2023-05-22 XLY.Adjusted 151.39566
## 390 2023-05-29 XLY.Adjusted 156.40665
## 391 2023-06-05 XLY.Adjusted 160.59077
## 392 2023-06-12 XLY.Adjusted 165.75121
## 393 2023-06-19 XLY.Adjusted 164.82472
## 394 2023-06-26 XLY.Adjusted 169.50851
## 395 2023-07-03 XLY.Adjusted 169.06929
## 396 2023-07-10 XLY.Adjusted 174.60944
## 397 2023-07-17 XLY.Adjusted 170.67644
## 398 2023-07-24 XLY.Adjusted 172.45326
## 399 2023-07-31 XLY.Adjusted 171.24542
## 400 2023-08-07 XLY.Adjusted 169.41867
## 401 2023-08-14 XLY.Adjusted 162.37122
## 402 2023-08-21 XLY.Adjusted 164.36766
## 403 2023-08-28 XLY.Adjusted 169.36876
## 404 2023-09-04 XLY.Adjusted 168.45039
## 405 2023-09-11 XLY.Adjusted 171.48499
## 406 2023-09-18 XLY.Adjusted 160.64427
## 407 2023-09-25 XLY.Adjusted 160.98000
```

```
## 408 2023-10-02 XLY.Adjusted 160.61000
## 409 2023-10-09 XLY.Adjusted 158.97000
## 410 2023-10-16 XLY.Adjusted 151.69000
## 411 2023-10-23 XLY.Adjusted 149.59000
## 412 2023-10-30 XLY.Adjusted 160.22000
## 413 2023-11-06 XLY.Adjusted 161.39999
## 414 2023-11-13 XLY.Adjusted 167.14000
## 415 2023-11-20 XLY.Adjusted 168.24001
## 416 2023-11-27 XLY.Adjusted 168.78000
## 417 2021-12-06 XLE.Adjusted  52.77941
## 418 2021-12-13 XLE.Adjusted  50.13306
## 419 2021-12-20 XLE.Adjusted  50.62176
## 420 2021-12-27 XLE.Adjusted  51.83763
## 421 2022-01-03 XLE.Adjusted  57.29226
## 422 2022-01-10 XLE.Adjusted  60.25307
## 423 2022-01-17 XLE.Adjusted  58.32901
## 424 2022-01-24 XLE.Adjusted  61.29916
## 425 2022-01-31 XLE.Adjusted  64.35338
## 426 2022-02-07 XLE.Adjusted  65.76375
## 427 2022-02-14 XLE.Adjusted  63.55947
## 428 2022-02-21 XLE.Adjusted  64.33471
## 429 2022-02-28 XLE.Adjusted  70.26569
## 430 2022-03-07 XLE.Adjusted  71.77878
## 431 2022-03-14 XLE.Adjusted  69.00475
## 432 2022-03-21 XLE.Adjusted  73.55341
## 433 2022-03-28 XLE.Adjusted  72.66339
## 434 2022-04-04 XLE.Adjusted  74.99246
## 435 2022-04-11 XLE.Adjusted  75.29420
## 436 2022-04-18 XLE.Adjusted  71.85246
## 437 2022-04-25 XLE.Adjusted  70.86237
## 438 2022-05-02 XLE.Adjusted  78.18905
## 439 2022-05-09 XLE.Adjusted  76.17114
## 440 2022-05-16 XLE.Adjusted  77.10464
## 441 2022-05-23 XLE.Adjusted  83.47897
## 442 2022-05-30 XLE.Adjusted  84.40307
## 443 2022-06-06 XLE.Adjusted  83.64870
## 444 2022-06-13 XLE.Adjusted  69.29707
## 445 2022-06-20 XLE.Adjusted  67.50547
## 446 2022-06-27 XLE.Adjusted  69.20649
## 447 2022-07-04 XLE.Adjusted  67.65225
## 448 2022-07-11 XLE.Adjusted  65.40194
## 449 2022-07-18 XLE.Adjusted  67.84296
## 450 2022-07-25 XLE.Adjusted  74.77505
## 451 2022-08-01 XLE.Adjusted  69.68324
## 452 2022-08-08 XLE.Adjusted  74.82272
## 453 2022-08-15 XLE.Adjusted  75.76671
## 454 2022-08-22 XLE.Adjusted  78.98959
## 455 2022-08-29 XLE.Adjusted  76.24346
## 456 2022-09-05 XLE.Adjusted  76.85371
## 457 2022-09-12 XLE.Adjusted  74.79412
## 458 2022-09-19 XLE.Adjusted  67.20410
## 459 2022-09-26 XLE.Adjusted  69.42124
## 460 2022-10-03 XLE.Adjusted  78.84834
## 461 2022-10-10 XLE.Adjusted  77.38319
```

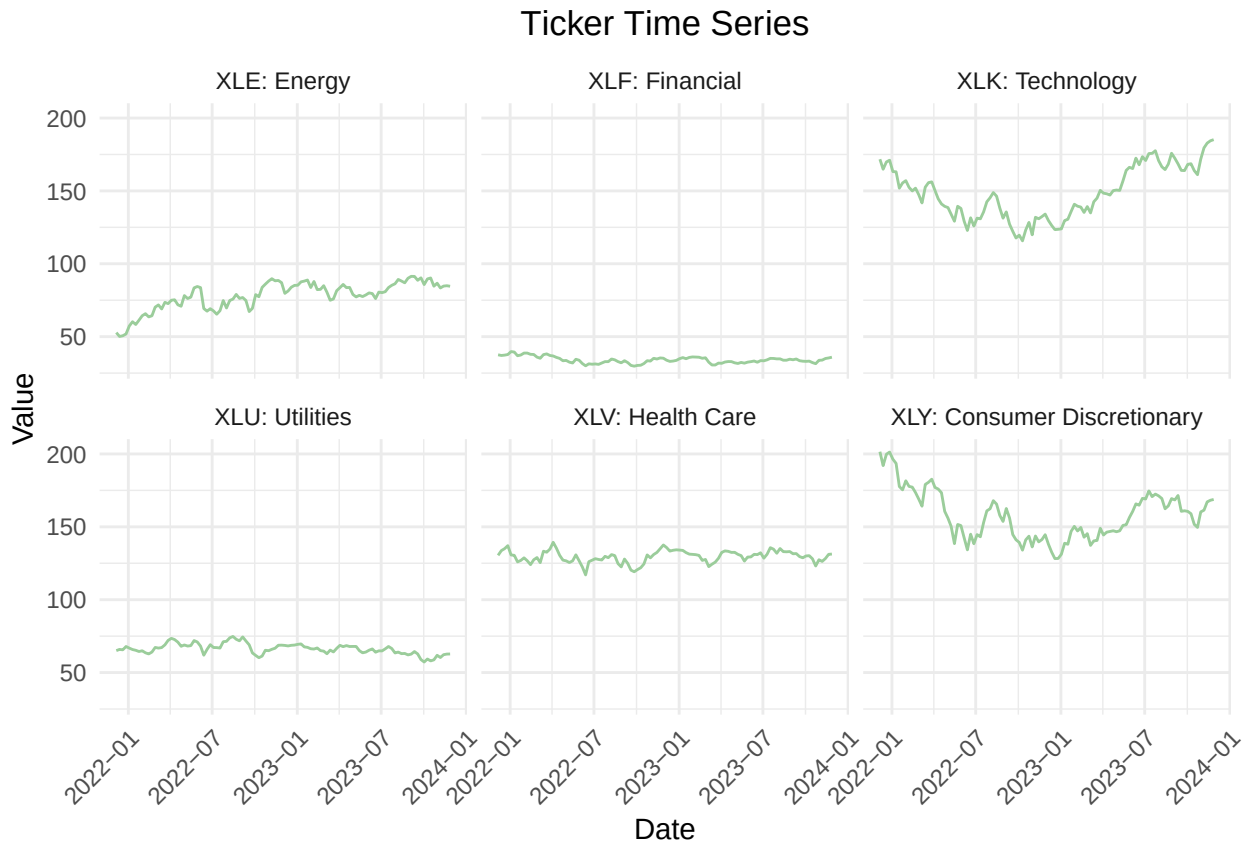
```
## 462 2022-10-17 XLE.Adjusted  83.79324
## 463 2022-10-24 XLE.Adjusted  86.02953
## 464 2022-10-31 XLE.Adjusted  88.12122
## 465 2022-11-07 XLE.Adjusted  89.76951
## 466 2022-11-14 XLE.Adjusted  88.32364
## 467 2022-11-21 XLE.Adjusted  88.55499
## 468 2022-11-28 XLE.Adjusted  87.05128
## 469 2022-12-05 XLE.Adjusted  79.69659
## 470 2022-12-12 XLE.Adjusted  81.31597
## 471 2022-12-19 XLE.Adjusted  83.91855
## 472 2022-12-26 XLE.Adjusted  85.18214
## 473 2023-01-02 XLE.Adjusted  85.26978
## 474 2023-01-09 XLE.Adjusted  87.59727
## 475 2023-01-16 XLE.Adjusted  88.11341
## 476 2023-01-23 XLE.Adjusted  88.84379
## 477 2023-01-30 XLE.Adjusted  83.71163
## 478 2023-02-06 XLE.Adjusted  87.85046
## 479 2023-02-13 XLE.Adjusted  82.28008
## 480 2023-02-20 XLE.Adjusted  82.44563
## 481 2023-02-27 XLE.Adjusted  84.97763
## 482 2023-03-06 XLE.Adjusted  80.46873
## 483 2023-03-13 XLE.Adjusted  74.95678
## 484 2023-03-20 XLE.Adjusted  75.85271
## 485 2023-03-27 XLE.Adjusted  81.50535
## 486 2023-04-03 XLE.Adjusted  83.62096
## 487 2023-04-10 XLE.Adjusted  85.83498
## 488 2023-04-17 XLE.Adjusted  83.62096
## 489 2023-04-24 XLE.Adjusted  83.76856
## 490 2023-05-01 XLE.Adjusted  78.94692
## 491 2023-05-08 XLE.Adjusted  77.26427
## 492 2023-05-15 XLE.Adjusted  78.36636
## 493 2023-05-22 XLE.Adjusted  77.51027
## 494 2023-05-29 XLE.Adjusted  78.62221
## 495 2023-06-05 XLE.Adjusted  80.02934
## 496 2023-06-12 XLE.Adjusted  79.56685
## 497 2023-06-19 XLE.Adjusted  76.12283
## 498 2023-06-26 XLE.Adjusted  80.57540
## 499 2023-07-03 XLE.Adjusted  80.19819
## 500 2023-07-10 XLE.Adjusted  80.85336
## 501 2023-07-17 XLE.Adjusted  83.68248
## 502 2023-07-24 XLE.Adjusted  85.22112
## 503 2023-07-31 XLE.Adjusted  86.28329
## 504 2023-08-07 XLE.Adjusted  89.24146
## 505 2023-08-14 XLE.Adjusted  88.15944
## 506 2023-08-21 XLE.Adjusted  86.94837
## 507 2023-08-28 XLE.Adjusted  90.07530
## 508 2023-09-04 XLE.Adjusted  91.37571
## 509 2023-09-11 XLE.Adjusted  91.33600
## 510 2023-09-18 XLE.Adjusted  88.65578
## 511 2023-09-25 XLE.Adjusted  90.39000
## 512 2023-10-02 XLE.Adjusted  85.73000
## 513 2023-10-09 XLE.Adjusted  89.59000
## 514 2023-10-16 XLE.Adjusted  90.26000
## 515 2023-10-23 XLE.Adjusted  84.63000
```

```
## 516 2023-10-30 XLE.Adjusted  86.68000
## 517 2023-11-06 XLE.Adjusted  83.41000
## 518 2023-11-13 XLE.Adjusted  84.69000
## 519 2023-11-20 XLE.Adjusted  84.93000
## 520 2023-11-27 XLE.Adjusted  84.58000
## 521 2021-12-06 XLU.Adjusted  65.07195
## 522 2021-12-13 XLU.Adjusted  65.88126
## 523 2021-12-20 XLU.Adjusted  65.60835
## 524 2021-12-27 XLU.Adjusted  67.88812
## 525 2022-01-03 XLU.Adjusted  66.77845
## 526 2022-01-10 XLU.Adjusted  65.82055
## 527 2022-01-17 XLU.Adjusted  65.28942
## 528 2022-01-24 XLU.Adjusted  64.45483
## 529 2022-01-31 XLU.Adjusted  64.94799
## 530 2022-02-07 XLU.Adjusted  63.57278
## 531 2022-02-14 XLU.Adjusted  62.80456
## 532 2022-02-21 XLU.Adjusted  64.09442
## 533 2022-02-28 XLU.Adjusted  67.23369
## 534 2022-03-07 XLU.Adjusted  66.79742
## 535 2022-03-14 XLU.Adjusted  67.15782
## 536 2022-03-21 XLU.Adjusted  69.02621
## 537 2022-03-28 XLU.Adjusted  72.08437
## 538 2022-04-04 XLU.Adjusted  73.47832
## 539 2022-04-11 XLU.Adjusted  72.64767
## 540 2022-04-18 XLU.Adjusted  70.89094
## 541 2022-04-25 XLU.Adjusted  68.03619
## 542 2022-05-02 XLU.Adjusted  68.94322
## 543 2022-05-09 XLU.Adjusted  68.19850
## 544 2022-05-16 XLU.Adjusted  68.49450
## 545 2022-05-23 XLU.Adjusted  71.96026
## 546 2022-05-30 XLU.Adjusted  70.98640
## 547 2022-06-06 XLU.Adjusted  68.11257
## 548 2022-06-13 XLU.Adjusted  61.93528
## 549 2022-06-20 XLU.Adjusted  65.88799
## 550 2022-06-27 XLU.Adjusted  69.14847
## 551 2022-07-04 XLU.Adjusted  67.19479
## 552 2022-07-11 XLU.Adjusted  67.13706
## 553 2022-07-18 XLU.Adjusted  66.81947
## 554 2022-07-25 XLU.Adjusted  71.16952
## 555 2022-08-01 XLU.Adjusted  71.43900
## 556 2022-08-08 XLU.Adjusted  73.83537
## 557 2022-08-15 XLU.Adjusted  74.77852
## 558 2022-08-22 XLU.Adjusted  72.86333
## 559 2022-08-29 XLU.Adjusted  71.79507
## 560 2022-09-05 XLU.Adjusted  74.47055
## 561 2022-09-12 XLU.Adjusted  71.66998
## 562 2022-09-19 XLU.Adjusted  69.06186
## 563 2022-09-26 XLU.Adjusted  63.47388
## 564 2022-10-03 XLU.Adjusted  61.77827
## 565 2022-10-10 XLU.Adjusted  60.21831
## 566 2022-10-17 XLU.Adjusted  61.34226
## 567 2022-10-24 XLU.Adjusted  65.31483
## 568 2022-10-31 XLU.Adjusted  65.00476
## 569 2022-11-07 XLU.Adjusted  65.96400
```

```
## 570 2022-11-14 XLU.Adjusted  66.71007
## 571 2022-11-21 XLU.Adjusted  68.73511
## 572 2022-11-28 XLU.Adjusted  68.82231
## 573 2022-12-05 XLU.Adjusted  68.62854
## 574 2022-12-12 XLU.Adjusted  68.28941
## 575 2022-12-19 XLU.Adjusted  68.70604
## 576 2022-12-26 XLU.Adjusted  68.86770
## 577 2023-01-02 XLU.Adjusted  69.35612
## 578 2023-01-09 XLU.Adjusted  69.67847
## 579 2023-01-16 XLU.Adjusted  67.62711
## 580 2023-01-23 XLU.Adjusted  67.29498
## 581 2023-01-30 XLU.Adjusted  66.33767
## 582 2023-02-06 XLU.Adjusted  66.14229
## 583 2023-02-13 XLU.Adjusted  66.89447
## 584 2023-02-20 XLU.Adjusted  65.06777
## 585 2023-02-27 XLU.Adjusted  64.71609
## 586 2023-03-06 XLU.Adjusted  62.92846
## 587 2023-03-13 XLU.Adjusted  65.41943
## 588 2023-03-20 XLU.Adjusted  64.16907
## 589 2023-03-27 XLU.Adjusted  66.61012
## 590 2023-04-03 XLU.Adjusted  68.69630
## 591 2023-04-10 XLU.Adjusted  67.79098
## 592 2023-04-17 XLU.Adjusted  68.50934
## 593 2023-04-24 XLU.Adjusted  67.87955
## 594 2023-05-01 XLU.Adjusted  67.92875
## 595 2023-05-08 XLU.Adjusted  67.92875
## 596 2023-05-15 XLU.Adjusted  65.05534
## 597 2023-05-22 XLU.Adjusted  63.55957
## 598 2023-05-29 XLU.Adjusted  64.08112
## 599 2023-06-05 XLU.Adjusted  65.33086
## 600 2023-06-12 XLU.Adjusted  66.22635
## 601 2023-06-19 XLU.Adjusted  63.99256
## 602 2023-06-26 XLU.Adjusted  64.91399
## 603 2023-07-03 XLU.Adjusted  64.86439
## 604 2023-07-10 XLU.Adjusted  66.31264
## 605 2023-07-17 XLU.Adjusted  67.90971
## 606 2023-07-24 XLU.Adjusted  66.50112
## 607 2023-07-31 XLU.Adjusted  63.47564
## 608 2023-08-07 XLU.Adjusted  64.05097
## 609 2023-08-14 XLU.Adjusted  62.97966
## 610 2023-08-21 XLU.Adjusted  63.11853
## 611 2023-08-28 XLU.Adjusted  62.13649
## 612 2023-09-04 XLU.Adjusted  62.67215
## 613 2023-09-11 XLU.Adjusted  64.41801
## 614 2023-09-18 XLU.Adjusted  62.78127
## 615 2023-09-25 XLU.Adjusted  58.93000
## 616 2023-10-02 XLU.Adjusted  57.25000
## 617 2023-10-09 XLU.Adjusted  59.30000
## 618 2023-10-16 XLU.Adjusted  58.03000
## 619 2023-10-23 XLU.Adjusted  58.73000
## 620 2023-10-30 XLU.Adjusted  61.86000
## 621 2023-11-06 XLU.Adjusted  60.32000
## 622 2023-11-13 XLU.Adjusted  62.29000
## 623 2023-11-20 XLU.Adjusted  62.69000
```

```
## 624 2023-11-27 XLU.Adjusted  62.76000
```

```
# Line Chart
# New facet label names for ticker_data
ticker.labs <- c("XLK: Technology", "XLF: Financial", "XLV: Health Care",
                 "XLY: Consumer Discretionary", "XLE: Energy", "XLU: Utilities")
names(ticker.labs) <- c("XLK.Adjusted", "XLF.Adjusted", "XLV.Adjusted",
                        "XLY.Adjusted", "XLE.Adjusted", "XLU.Adjusted")

# Create the plot
ggplot(ticker_data_df, aes(Date, Value)) +
  geom_line(color='darkseagreen3') +
  facet_wrap(~Ticker, labeller = labeller(Ticker = ticker.labs)) +
  ggtitle("Ticker Time Series") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

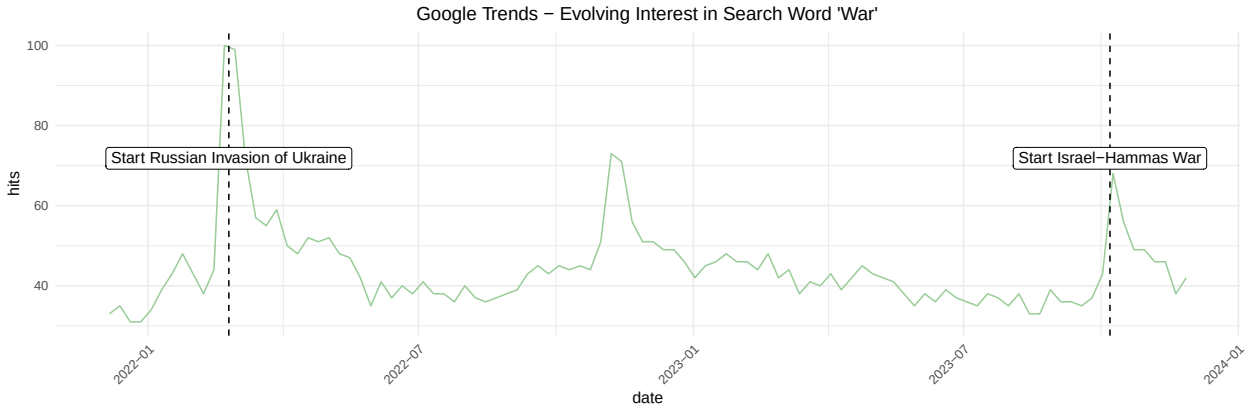



```
# Histogram
ggplot(ticker_data_df, aes(Value)) +
  geom_histogram(color='darkseagreen3', fill='darkseagreen3') +
  facet_wrap(~Ticker, labeller = labeller(Ticker = ticker.labs)) +
  ggtitle("Ticker Distributions") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

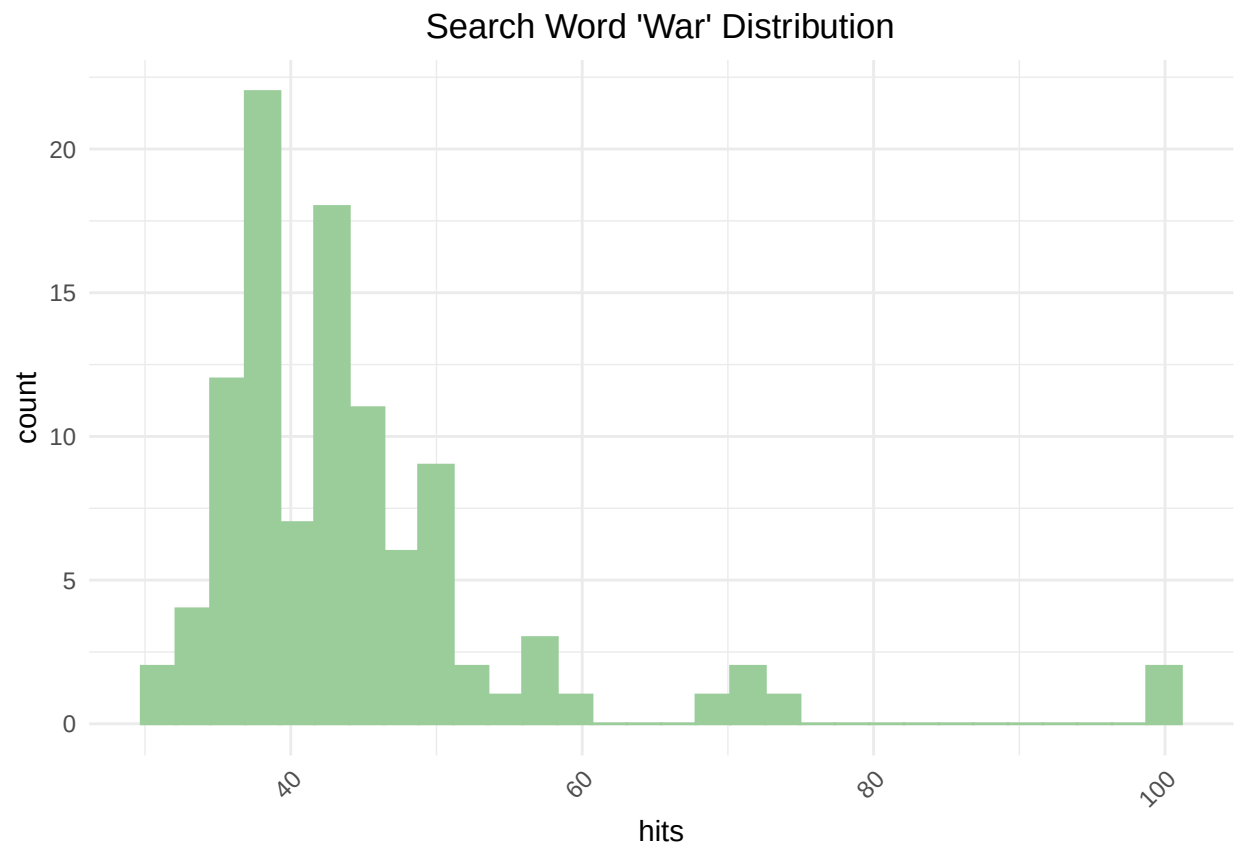
Visualize Google Trends Data

```
# Line chart

# Convert 'date' column to POSIXct format
gtrends_war_web$date <- as.POSIXct(gtrends_war_web$date, format = "%Y-%m-%d")

ggplot(gtrends_war_web, aes(date, hits)) +
  geom_line(color='darkseagreen3') +
  theme_minimal() +
  ggtitle("Google Trends - Evolving Interest in Search Word 'War'") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5)) +
  geom_vline(xintercept = as.POSIXct("2022-02-24"), linetype="dashed",
             color = "black", size=0.5) +
  annotate('label', x=as.POSIXct("2022-02-24"), y=80,
           label="Start Russian Invasion of Ukraine", vjust=2, color="black") +
  geom_vline(xintercept = as.POSIXct("2023-10-07"), linetype="dashed",
               color = "black", size=0.5) +
  annotate('label', x=as.POSIXct("2023-10-07"), y=80,
             label="Start Israel-Hammas War", vjust=2, color="black")
```




```
# Histogram
ggplot(gtrends_war_web, aes(hits)) +
  geom_histogram(color='darkseagreen3', fill='darkseagreen3') +
  ggtitle("Search Word 'War' Distribution") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Stationarity test and adjustment

```
# Check stationarity
sapply(ticker_data, adf.test)
```

Ticker Data

```
##             XLK.Adjusted                    XLF.Adjusted
## statistic   -1.952864                       -2.839779
## parameter   4                               4
## alternative "stationary"                    "stationary"
## p.value     0.5962105                       0.228463
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "X[[i]]"                        "X[[i]]"
##             XLV.Adjusted                    XLY.Adjusted
## statistic   -3.79224                       -2.573108
## parameter   4                               4
## alternative "stationary"                    "stationary"
## p.value     0.02194769                      0.3390346
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "X[[i]]"                        "X[[i]]"
##             XLE.Adjusted                    XLU.Adjusted
## statistic   -3.427281                       -3.526364
## parameter   4                               4
## alternative "stationary"                    "stationary"
## p.value     0.05371989                      0.04313629
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "X[[i]]"                        "X[[i]]"
```

Non-stationary are XLK, XLF, XLY, XLE. Make them stationary. Only log the others.

```
ticker_data_differenced <- NULL
ticker_data_differenced$XLK.Adjusted <- na.omit(diff(log(ticker_data$XLK.Adjusted)))
ticker_data_differenced$XLF.Adjusted <- na.omit(diff(log(ticker_data$XLF.Adjusted)))
ticker_data_differenced$XLY.Adjusted <- na.omit(diff(log(ticker_data$XLY.Adjusted)))
ticker_data_differenced$XLE.Adjusted <- na.omit(diff(log(ticker_data$XLE.Adjusted)))

# Already stationary ticker series
ticker_data_differenced$XLV.Adjusted <- log(ticker_data$XLV.Adjusted)
ticker_data_differenced$XLU.Adjusted <- log(ticker_data$XLU.Adjusted)

# For non differenced time series remove first row
ticker_data_differenced$XLV.Adjusted <-  ticker_data_differenced$XLV.Adjusted[-1,]
ticker_data_differenced$XLU.Adjusted <-  ticker_data_differenced$XLU.Adjusted[-1,]
```

```
# Check stationarity again after diff-log transformation
sapply(ticker_data_differenced, adf.test)
```

```
## Warning in FUN(X[[i]], ...): p-value smaller than printed p-value
```

```
## Warning in FUN(X[[i]], ...): p-value smaller than printed p-value

## Warning in FUN(X[[i]], ...): p-value smaller than printed p-value

## Warning in FUN(X[[i]], ...): p-value smaller than printed p-value

##             XLK.Adjusted                    XLF.Adjusted
## statistic   -4.849782                       -5.398199
## parameter   4                               4
## alternative "stationary"                    "stationary"
## p.value     0.01                            0.01
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "X[[i]]"                        "X[[i]]"
##             XLY.Adjusted                    XLE.Adjusted
## statistic   -4.882529                       -5.159398
## parameter   4                               4
## alternative "stationary"                    "stationary"
## p.value     0.01                            0.01
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "X[[i]]"                        "X[[i]]"
##             XLV.Adjusted                    XLU.Adjusted
## statistic   -3.793162                       -3.567446
## parameter   4                               4
## alternative "stationary"                    "stationary"
## p.value     0.02191663                      0.03947994
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "X[[i]]"                        "X[[i]]"
```

After the diff-log-transformation all the time series are stationary.

```
adf.test(gtrends_war_web$hits)
```

Google Trends Data

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gtrends_war_web$hits
## Dickey-Fuller = -3.5584, Lag order = 4, p-value = 0.04027
## alternative hypothesis: stationary
```

Not stationary and distribution has long tail on right –> Log Diff. Although p value is slightly below 0.05.

```
logged_diff_hits <- na.omit(diff(log(gtrends_war_web$hits)))

corresponding_dates <- gtrends_war_web[-1,]$date # +1 due to diff reduction

gtrends_war_web_differenced <- data.frame(
  date = corresponding_dates,
  hits = logged_diff_hits
```

```
)
# Check stationarity after diff-log-transformation
adf.test(gtrends_war_web_differenced$hits)
```

```
## Warning in adf.test(gtrends_war_web_differenced$hits): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gtrends_war_web_differenced$hits
## Dickey-Fuller = -5.2254, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Check that differenced time series have same length.

```
# Check length of time series
length(gtrends_war_web_differenced$date) == length(ticker_data_differenced$XLK.Adjusted)
```

```
## [1] TRUE
```

```
length(gtrends_war_web_differenced$date)
```

```
## [1] 103
```

```
length(ticker_data_differenced$XLK.Adjusted)
```

```
## [1] 103
```

```
min(gtrends_war_web_differenced$date)
```

```
## [1] "2021-12-13 01:00:00 CET"
```

```
max(gtrends_war_web_differenced$date)
```

```
## [1] "2023-11-27 01:00:00 CET"
```

Create final dataframe

```
# Make tickers in list to dataframe
ticker_data_differenced_df <- do.call(cbind.data.frame, ticker_data_differenced)

# Make date index to column for merging
ticker_data_differenced_df$date <- rownames(ticker_data_differenced_df)

# Make date in gtrends to character in order to merge
gtrends_war_web_differenced$date <- format(gtrends_war_web_differenced$date, format = "%Y-%m-%d")

# Merge
war_ticker_df <- left_join(gtrends_war_web_differenced, ticker_data_differenced_df)
```

```
## Joining, by = "date"
```

4. VAR & Causality Testing

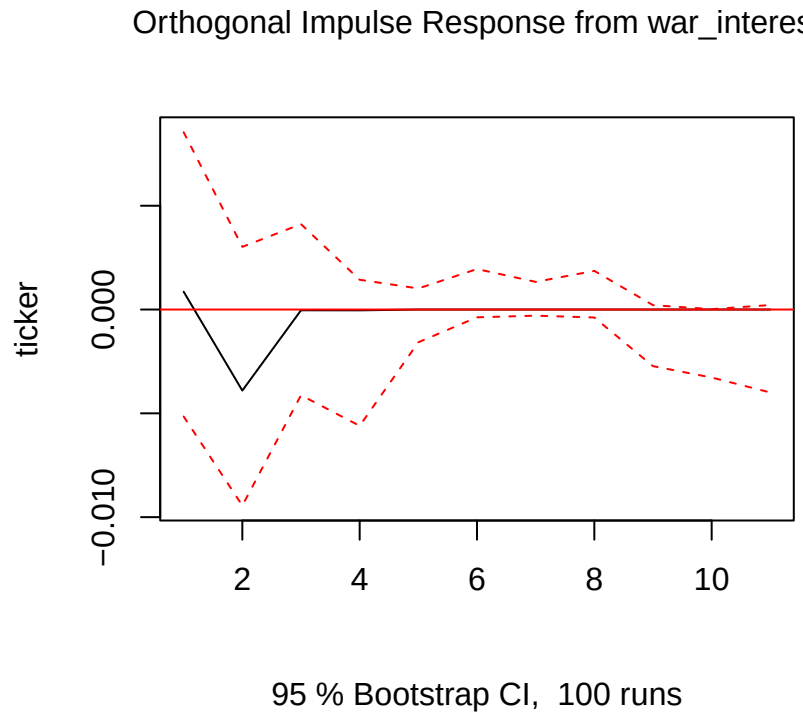
TODO: - Check lag.max what to apply - Control variable?

```
# Create ticker list
ticker_cols <- colnames(war_ticker_df)[3:8]

for (Ticker in ticker_cols) {

  # Create data for VAR
  print(Ticker)
  data_for_var <- cbind(war_interest=war_ticker_df$hits, ticker=war_ticker_df[[Ticker]])

  # Run VAR model
  VAR_est <- VAR(data_for_var, ic = "AIC", lag.max = 24)
  coefs <-coeftest(VAR_est)
  summ <- summary(VAR_est)

  print(coefs)
  print(summ)

  # Run Granger Causality Test
  causal <- causality(VAR_est, cause="war_interest")["Granger"]
  print(causal)

  # Run Granger Causality Test other way (check if index has influence on crisis measure)
  causal_ticker <- causality(VAR_est, cause="ticker")["Granger"]
  print(causal_ticker)

  # Impulse response functions
  plot(irf(VAR_est, impulse="war_interest", response="ticker"))

}
```

```
## [1] "XLK.Adjusted"
##
## t test of coefficients:
##
##                                Estimate Std. Error t value Pr(>|t|)
## war_interest:(Intercept)      0.0018754  0.0136893  0.1370   0.8913
## war_interest:war_interest.l1  0.0590052  0.1003184  0.5882   0.5578
## war_interest:ticker.l1       -0.2429640  0.3888822 -0.6248   0.5336
## ticker:(Intercept)            0.0012100  0.0034886  0.3469   0.7294
## ticker:war_interest.l1       -0.0279391  0.0255650 -1.0929   0.2771
## ticker:ticker.l1             -0.0479634  0.0991024 -0.4840   0.6295
##
##
## VAR Estimation Results:
## ========================
## Endogenous variables: war_interest, ticker
## Deterministic variables: const
## Sample size: 102
## Log Likelihood: 256.754
## Roots of the characteristic polynomial:
```

```
## 0.1037 0.09271
## Call:
## VAR(y = data_for_var, lag.max = 24, ic = "AIC")
##
##
## Estimation results for equation war_interest:
## =============================================
## war_interest = war_interest.l1 + ticker.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## war_interest.l1  0.059005   0.100318   0.588    0.558
## ticker.l1       -0.242964   0.388882  -0.625    0.534
## const            0.001875   0.013689   0.137    0.891
##
##
## Residual standard error: 0.1382 on 99 degrees of freedom
## Multiple R-Squared: 0.007273,    Adjusted R-squared: -0.01278
## F-statistic: 0.3627 on 2 and 99 DF,  p-value: 0.6967
##
##
## Estimation results for equation ticker:
## =======================================
## ticker = war_interest.l1 + ticker.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## war_interest.l1 -0.027939   0.025565  -1.093    0.277
## ticker.l1       -0.047963   0.099102  -0.484    0.629
## const            0.001210   0.003489   0.347    0.729
##
##
## Residual standard error: 0.03522 on 99 degrees of freedom
## Multiple R-Squared: 0.01439, Adjusted R-squared: -0.005526
## F-statistic: 0.7225 on 2 and 99 DF,  p-value: 0.4881
##
##
##
## Covariance matrix of residuals:
##              war_interest    ticker
## war_interest    0.0191051 0.0001187
## ticker          0.0001187 0.0012407
##
## Correlation matrix of residuals:
##              war_interest  ticker
## war_interest      1.00000 0.02438
## ticker            0.02438 1.00000
##
##
## $Granger
##
##  Granger causality H0: war_interest do not Granger-cause ticker
##
## data:  VAR object VAR_est
## F-Test = 1.1944, df1 = 1, df2 = 198, p-value = 0.2758
##
```

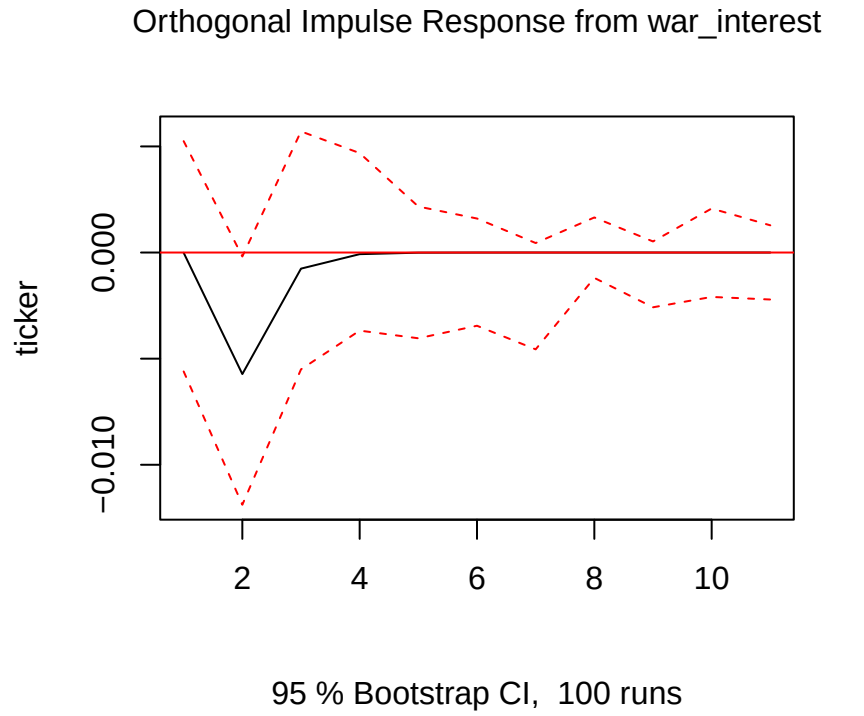
```
##
## $Granger
##
##  Granger causality H0: ticker do not Granger-cause war_interest
##
## data:  VAR object VAR_est
## F-Test = 0.39034, df1 = 1, df2 = 198, p-value = 0.5328
```




```
## [1] "XLF.Adjusted"
##
## t test of coefficients:
##
##                                Estimate  Std. Error t value Pr(>|t|)
## war_interest:(Intercept)      0.00170637  0.01371596  0.1244   0.9012
## war_interest:war_interest.l1  0.05805078  0.10051706  0.5775   0.5649
## war_interest:ticker.l1       -0.00138063  0.44072279 -0.0031   0.9975
## ticker:(Intercept)           -0.00022179  0.00306846 -0.0723   0.9425
## ticker:war_interest.l1       -0.04134066  0.02248709 -1.8384   0.0690 .
## ticker:ticker.l1              0.07443020  0.09859593  0.7549   0.4521
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## VAR Estimation Results:
## ========================
```

```
## Endogenous variables: war_interest, ticker
## Deterministic variables: const
## Sample size: 102
## Log Likelihood: 269.609
## Roots of the characteristic polynomial:
## 0.07738 0.0551
## Call:
## VAR(y = data_for_var, lag.max = 24, ic = "AIC")
##
##
## Estimation results for equation war_interest:
## =============================================
## war_interest = war_interest.l1 + ticker.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## war_interest.l1  0.058051   0.100517   0.578    0.565
## ticker.l1       -0.001381   0.440723  -0.003    0.998
## const            0.001706   0.013716   0.124    0.901
##
##
## Residual standard error: 0.1385 on 99 degrees of freedom
## Multiple R-Squared: 0.003359,    Adjusted R-squared: -0.01677
## F-statistic: 0.1668 on 2 and 99 DF,  p-value: 0.8466
##
##
## Estimation results for equation ticker:
## =======================================
## ticker = war_interest.l1 + ticker.l1 + const
##
##                   Estimate Std. Error t value Pr(>|t|)
## war_interest.l1 -0.0413407  0.0224871  -1.838    0.069 .
## ticker.l1        0.0744302  0.0985959   0.755    0.452
## const           -0.0002218  0.0030685  -0.072    0.943
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.03098 on 99 degrees of freedom
## Multiple R-Squared: 0.03879, Adjusted R-squared: 0.01937
## F-statistic: 1.997 on 2 and 99 DF,  p-value: 0.1411
##
##
##
## Covariance matrix of residuals:
##              war_interest     ticker
## war_interest    1.918e-02 -1.410e-06
## ticker         -1.410e-06  9.599e-04
##
## Correlation matrix of residuals:
##              war_interest     ticker
## war_interest    1.0000000 -0.0003285
## ticker         -0.0003285  1.0000000
##
##
```

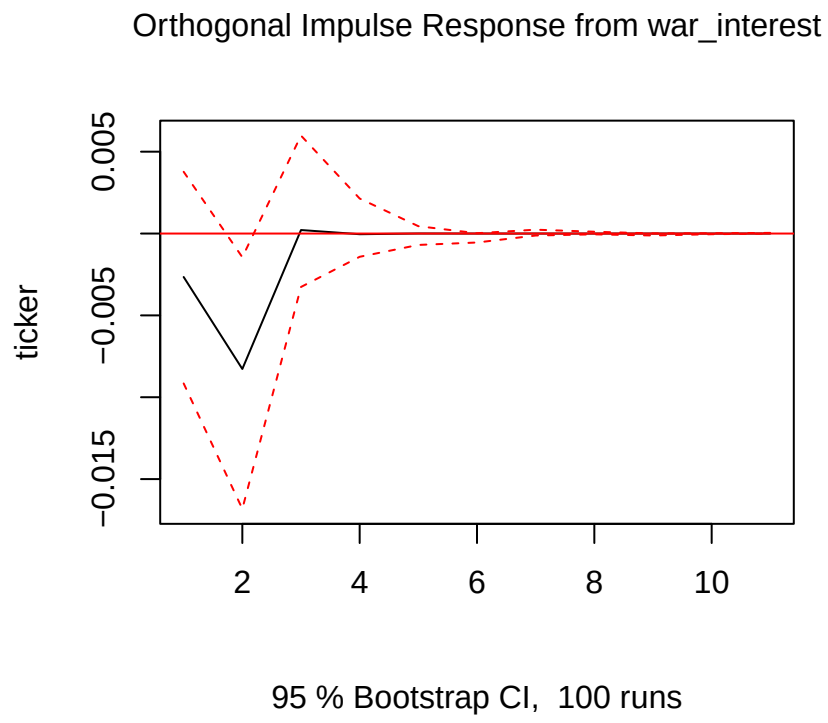
```
## $Granger
##
##  Granger causality H0: war_interest do not Granger-cause ticker
##
## data:  VAR object VAR_est
## F-Test = 3.3798, df1 = 1, df2 = 198, p-value = 0.0675
##
##
## $Granger
##
##  Granger causality H0: ticker do not Granger-cause war_interest
##
## data:  VAR object VAR_est
## F-Test = 9.8135e-06, df1 = 1, df2 = 198, p-value = 0.9975
```




```
## [1] "XLY.Adjusted"
##
## t test of coefficients:
##
##                                Estimate Std. Error t value Pr(>|t|)
## war_interest:(Intercept)      0.0017293  0.0137281  0.1260  0.90001
## war_interest:war_interest.l1  0.0583668  0.1008952  0.5785  0.56425
## war_interest:ticker.l1        0.0127692  0.3645436  0.0350  0.97213
## ticker:(Intercept)           -0.0013305  0.0036677 -0.3628  0.71756
## ticker:war_interest.l1       -0.0613923  0.0269556 -2.2775  0.02491 *
```

```
## ticker:ticker.l1             -0.0857372  0.0973932 -0.8803  0.38082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## VAR Estimation Results:
## ========================
## Endogenous variables: war_interest, ticker
## Deterministic variables: const
## Sample size: 102
## Log Likelihood: 251.769
## Roots of the characteristic polynomial:
## 0.08007 0.0527
## Call:
## VAR(y = data_for_var, lag.max = 24, ic = "AIC")
##
##
## Estimation results for equation war_interest:
## ==============================================
## war_interest = war_interest.l1 + ticker.l1 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## war_interest.l1 0.058367   0.100895   0.578    0.564
## ticker.l1       0.012769   0.364544   0.035    0.972
## const           0.001729   0.013728   0.126    0.900
##
##
## Residual standard error: 0.1385 on 99 degrees of freedom
## Multiple R-Squared: 0.003371,    Adjusted R-squared: -0.01676
## F-statistic: 0.1675 on 2 and 99 DF,  p-value: 0.8461
##
##
## Estimation results for equation ticker:
## ========================================
## ticker = war_interest.l1 + ticker.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## war_interest.l1 -0.061392   0.026956  -2.278   0.0249 *
## ticker.l1       -0.085737   0.097393  -0.880   0.3808
## const           -0.001330   0.003668  -0.363   0.7176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.037 on 99 degrees of freedom
## Multiple R-Squared: 0.05402, Adjusted R-squared: 0.03491
## F-statistic: 2.827 on 2 and 99 DF,  p-value: 0.06401
##
##
##
## Covariance matrix of residuals:
##              war_interest     ticker
## war_interest    0.0191802 -0.0003676
## ticker         -0.0003676  0.0013690
```

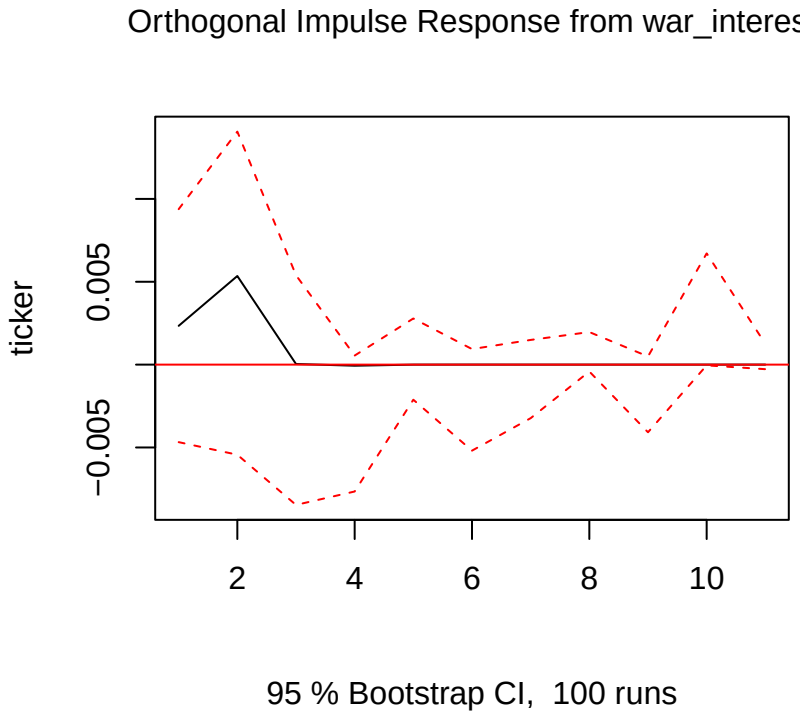
```
##
## Correlation matrix of residuals:
##              war_interest   ticker
## war_interest      1.00000 -0.07174
## ticker           -0.07174  1.00000
##
##
## $Granger
##
##  Granger causality H0: war_interest do not Granger-cause ticker
##
## data:  VAR object VAR_est
## F-Test = 5.1871, df1 = 1, df2 = 198, p-value = 0.02382
##
##
## $Granger
##
##  Granger causality H0: ticker do not Granger-cause war_interest
##
## data:  VAR object VAR_est
## F-Test = 0.0012269, df1 = 1, df2 = 198, p-value = 0.9721
```




```
## [1] "XLE.Adjusted"
##
## t test of coefficients:
```

```
##
##                                Estimate Std. Error t value Pr(>|t|)
## war_interest:(Intercept)      0.0035701  0.0136526  0.2615   0.7943
## war_interest:war_interest.l1  0.0661948  0.0997379  0.6637   0.5084
## war_interest:ticker.l1       -0.4018670  0.2927685 -1.3726   0.1730
## ticker:(Intercept)            0.0053230  0.0046244  1.1511   0.2525
## ticker:war_interest.l1        0.0398860  0.0337831  1.1806   0.2406
## ticker:ticker.l1             -0.0537286  0.0991663 -0.5418   0.5892
##
##
## VAR Estimation Results:
## ========================
## Endogenous variables: war_interest, ticker
## Deterministic variables: const
## Sample size: 102
## Log Likelihood: 229.351
## Roots of the characteristic polynomial:
## 0.1117 0.1117
## Call:
## VAR(y = data_for_var, lag.max = 24, ic = "AIC")
##
##
## Estimation results for equation war_interest:
## =============================================
## war_interest = war_interest.l1 + ticker.l1 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## war_interest.l1  0.06620    0.09974   0.664    0.508
## ticker.l1       -0.40187    0.29277  -1.373    0.173
## const            0.00357    0.01365   0.261    0.794
##
##
## Residual standard error: 0.1372 on 99 degrees of freedom
## Multiple R-Squared: 0.02197, Adjusted R-squared: 0.002215
## F-statistic: 1.112 on 2 and 99 DF,  p-value: 0.3329
##
##
## Estimation results for equation ticker:
## =======================================
## ticker = war_interest.l1 + ticker.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## war_interest.l1  0.039886   0.033783   1.181    0.241
## ticker.l1       -0.053729   0.099166  -0.542    0.589
## const            0.005323   0.004624   1.151    0.252
##
##
## Residual standard error: 0.04647 on 99 degrees of freedom
## Multiple R-Squared: 0.01607, Adjusted R-squared: -0.003805
## F-statistic: 0.8086 on 2 and 99 DF,  p-value: 0.4484
##
##
##
## Covariance matrix of residuals:
```

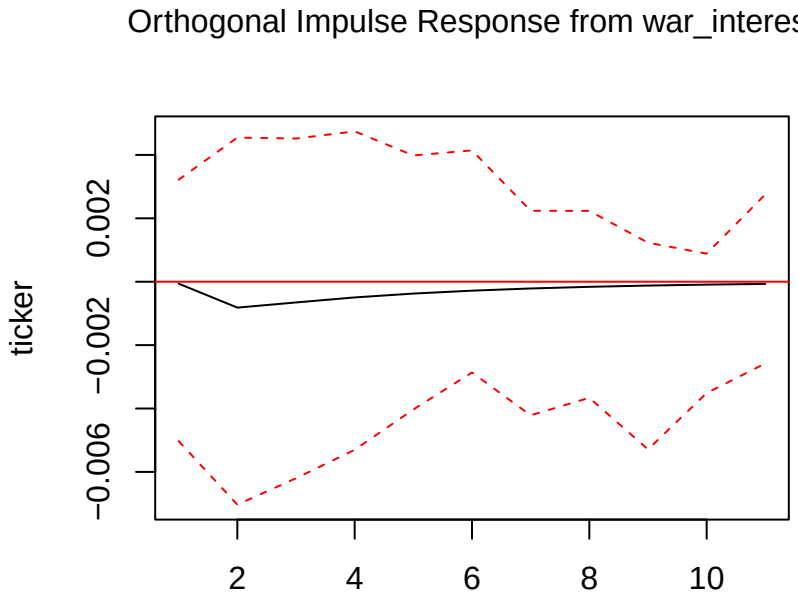
```
##              war_interest   ticker
## war_interest     0.018822 0.000321
## ticker           0.000321 0.002159
##
## Correlation matrix of residuals:
##              war_interest  ticker
## war_interest      1.00000 0.05034
## ticker            0.05034 1.00000
##
##
## $Granger
##
##  Granger causality H0: war_interest do not Granger-cause ticker
##
## data:  VAR object VAR_est
## F-Test = 1.3939, df1 = 1, df2 = 198, p-value = 0.2392
##
##
## $Granger
##
##  Granger causality H0: ticker do not Granger-cause war_interest
##
## data:  VAR object VAR_est
## F-Test = 1.8842, df1 = 1, df2 = 198, p-value = 0.1714
```




```
## [1] "XLV.Adjusted"
```

```
##
## t test of coefficients:
##
##                                Estimate Std. Error t value  Pr(>|t|)
## war_interest:(Intercept)      1.8501368  2.0648763  0.8960 0.3724243
## war_interest:war_interest.l1  0.0515319  0.1003650  0.5134 0.6087842
## war_interest:ticker.l1       -0.3800757  0.4245724 -0.8952 0.3728534
## ticker:(Intercept)            1.2078777  0.3191859  3.7842 0.0002642 ***
## ticker:war_interest.l1       -0.0055966  0.0155143 -0.3607 0.7190642
## ticker:ticker.l1              0.7515997  0.0656298 11.4521 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## VAR Estimation Results:
## ========================
## Endogenous variables: war_interest, ticker
## Deterministic variables: const
## Sample size: 102
## Log Likelihood: 308.138
## Roots of the characteristic polynomial:
## 0.7546 0.04851
## Call:
## VAR(y = data_for_var, lag.max = 24, ic = "AIC")
##
##
## Estimation results for equation war_interest:
## =============================================
## war_interest = war_interest.l1 + ticker.l1 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## war_interest.l1  0.05153    0.10037   0.513    0.609
## ticker.l1       -0.38008    0.42457  -0.895    0.373
## const            1.85014    2.06488   0.896    0.372
##
##
## Residual standard error: 0.1379 on 99 degrees of freedom
## Multiple R-Squared: 0.01136, Adjusted R-squared: -0.008611
## F-statistic: 0.5689 on 2 and 99 DF,  p-value: 0.568
##
##
## Estimation results for equation ticker:
## =======================================
## ticker = war_interest.l1 + ticker.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## war_interest.l1 -0.005597   0.015514  -0.361 0.719064
## ticker.l1        0.751600   0.065630  11.452  < 2e-16 ***
## const            1.207878   0.319186   3.784 0.000264 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02132 on 99 degrees of freedom
```

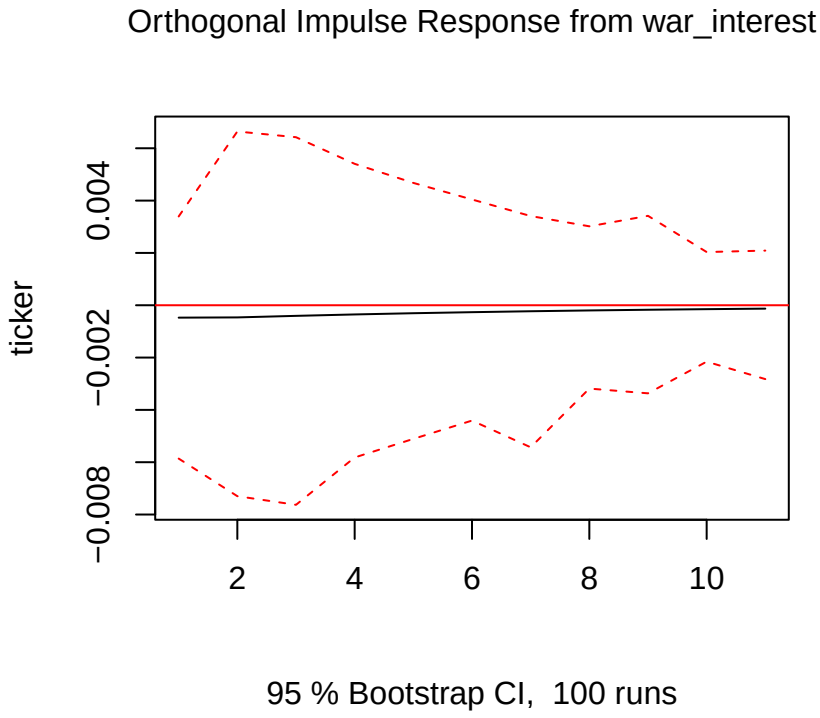
```
## Multiple R-Squared: 0.5725,  Adjusted R-squared: 0.5639
## F-statistic: 66.29 on 2 and 99 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##              war_interest     ticker
## war_interest    1.903e-02 -8.359e-06
## ticker         -8.359e-06  4.546e-04
##
## Correlation matrix of residuals:
##              war_interest    ticker
## war_interest     1.000000 -0.002842
## ticker          -0.002842  1.000000
##
##
## $Granger
##
##  Granger causality H0: war_interest do not Granger-cause ticker
##
## data:  VAR object VAR_est
## F-Test = 0.13013, df1 = 1, df2 = 198, p-value = 0.7187
##
##
## $Granger
##
##  Granger causality H0: ticker do not Granger-cause war_interest
##
## data:  VAR object VAR_est
## F-Test = 0.80138, df1 = 1, df2 = 198, p-value = 0.3718
```




```
## [1] "XLU.Adjusted"
##
## t test of coefficients:
##
##                                Estimate  Std. Error t value Pr(>|t|)
## war_interest:(Intercept)      1.29705199  1.04473348  1.2415  0.21735
## war_interest:war_interest.l1  0.04383161  0.10039046  0.4366  0.66334
## war_interest:ticker.l1       -0.30890095  0.24911652 -1.2400  0.21791
## ticker:(Intercept)            0.55600546  0.21504014  2.5856  0.01118 *
## ticker:war_interest.l1       -0.00039089  0.02066362 -0.0189  0.98495
## ticker:ticker.l1              0.86729398  0.05127628 16.9141  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## VAR Estimation Results:
## ========================
## Endogenous variables: war_interest, ticker
## Deterministic variables: const
## Sample size: 102
## Log Likelihood: 279.692
## Roots of the characteristic polynomial:
## 0.8674 0.04369
## Call:
## VAR(y = data_for_var, lag.max = 24, ic = "AIC")
##
```

```
##
## Estimation results for equation war_interest:
## ============================================
## war_interest = war_interest.l1 + ticker.l1 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## war_interest.l1  0.04383    0.10039   0.437    0.663
## ticker.l1       -0.30890    0.24912  -1.240    0.218
## const            1.29705    1.04473   1.242    0.217
##
##
## Residual standard error: 0.1374 on 99 degrees of freedom
## Multiple R-Squared: 0.0186,  Adjusted R-squared: -0.001225
## F-statistic: 0.9382 on 2 and 99 DF,  p-value: 0.3948
##
##
## Estimation results for equation ticker:
## =======================================
## ticker = war_interest.l1 + ticker.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## war_interest.l1 -0.0003909  0.0206636  -0.019   0.9849
## ticker.l1        0.8672940  0.0512763  16.914   <2e-16 ***
## const            0.5560055  0.2150401   2.586   0.0112 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02829 on 99 degrees of freedom
## Multiple R-Squared: 0.7455,  Adjusted R-squared: 0.7403
## F-statistic:   145 on 2 and 99 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##              war_interest     ticker
## war_interest    1.889e-02 -6.518e-05
## ticker         -6.518e-05  8.002e-04
##
## Correlation matrix of residuals:
##              war_interest   ticker
## war_interest      1.00000 -0.01677
## ticker           -0.01677  1.00000
##
##
## $Granger
##
##  Granger causality H0: war_interest do not Granger-cause ticker
##
## data:  VAR object VAR_est
## F-Test = 0.00035784, df1 = 1, df2 = 198, p-value = 0.9849
##
##
## $Granger
```

```
##
##  Granger causality H0: ticker do not Granger-cause war_interest
##
## data:  VAR object VAR_est
## F-Test = 1.5376, df1 = 1, df2 = 198, p-value = 0.2164
```



5. Results & Discussion