ARMA-X Analysis Tutorial

Contents

Data

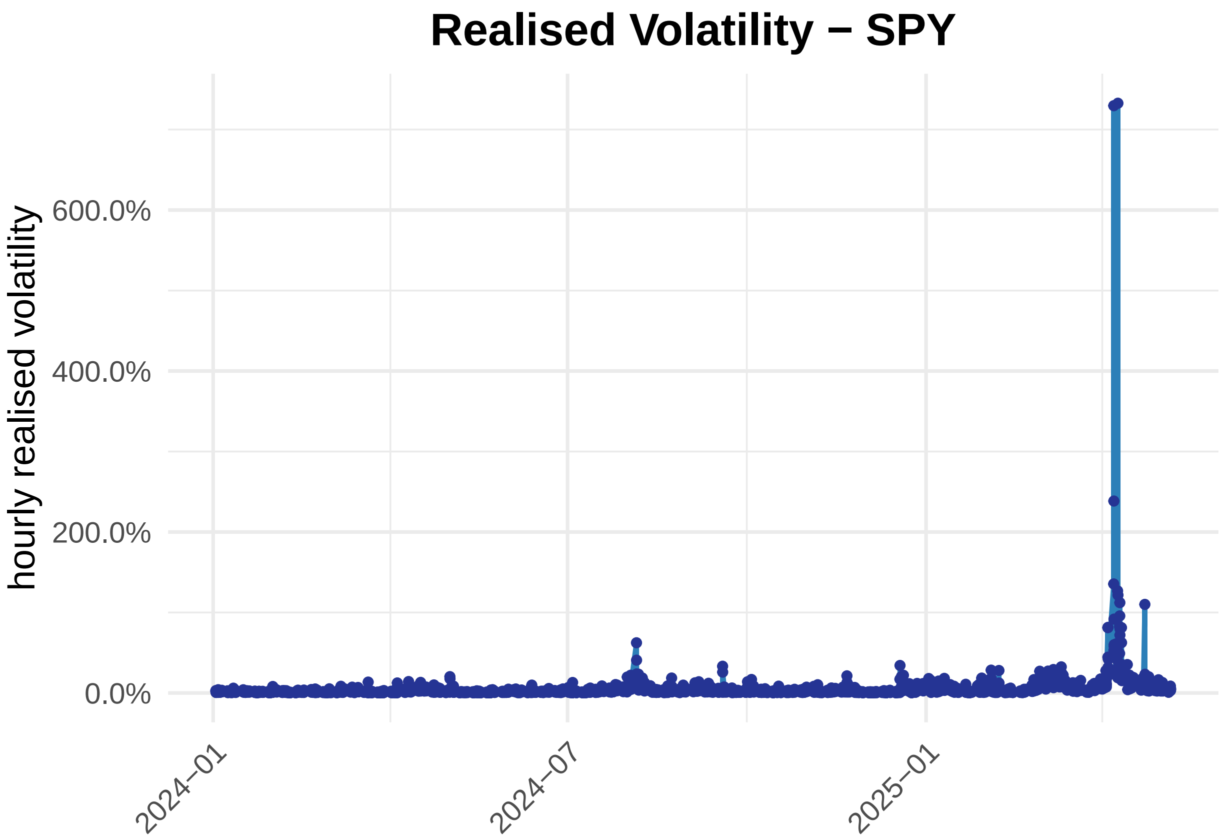
Load Base Data

```
# 1. Load Political Social Media

#contains posts from Twitter & TruthSocial
social <- read.csv(here("data/mothership", "social.csv"))

social_hourly <- read.csv(here("data/mothership", "socialhourly.csv"))


# 2. Load Financial

#S&P500
SPY <- read.csv(here("data/mothership", "SPY.csv"))

#STOXX50
VGK <- read.csv(here("data/mothership", "VGK.csv"))

#CSI 300 (China)
ASHR <- read.csv(here("data/mothership", "ASHR.CSV"))
```

```
#make posixct
SPY$timestamp = as.POSIXct(SPY$timestamp,format = "%Y-%m-%d %H:%M:%S")
VGK$timestamp = as.POSIXct(VGK$timestamp,format = "%Y-%m-%d %H:%M:%S")
ASHR$timestamp = as.POSIXct(ASHR$timestamp,format = "%Y-%m-%d %H:%M:%S")
social$timestamp = as.POSIXct(social$timestamp,format = "%Y-%m-%d %H:%M:%S")
social_hourly$timestamp = as.POSIXct(social_hourly$timestamp,format = "%Y-%m-%d %H:%M:%S")
```

Volatility

```
#find hourly volatility
#NOTE: this ignores tweets made outside trading hours!!
SPY_volatility_alltime = dplyr::select(SPY,timestamp,r_vol_h)

#aggregating per hour
SPY_volatility_alltime = SPY_volatility_alltime %>%
          mutate(timestamp = floor_date(timestamp, unit = "hour")) %>%
          distinct(timestamp, .keep_all = TRUE)

#select time period
SPY_volatility = filter(SPY_volatility_alltime,
                  between(timestamp,
                          as.Date('2024-01-01'),
                          as.Date('2025-05-07')))

#plot
hvol_plotter(SPY_volatility,breaks="3 month",
             title="Realised Volatility - SPY")
```

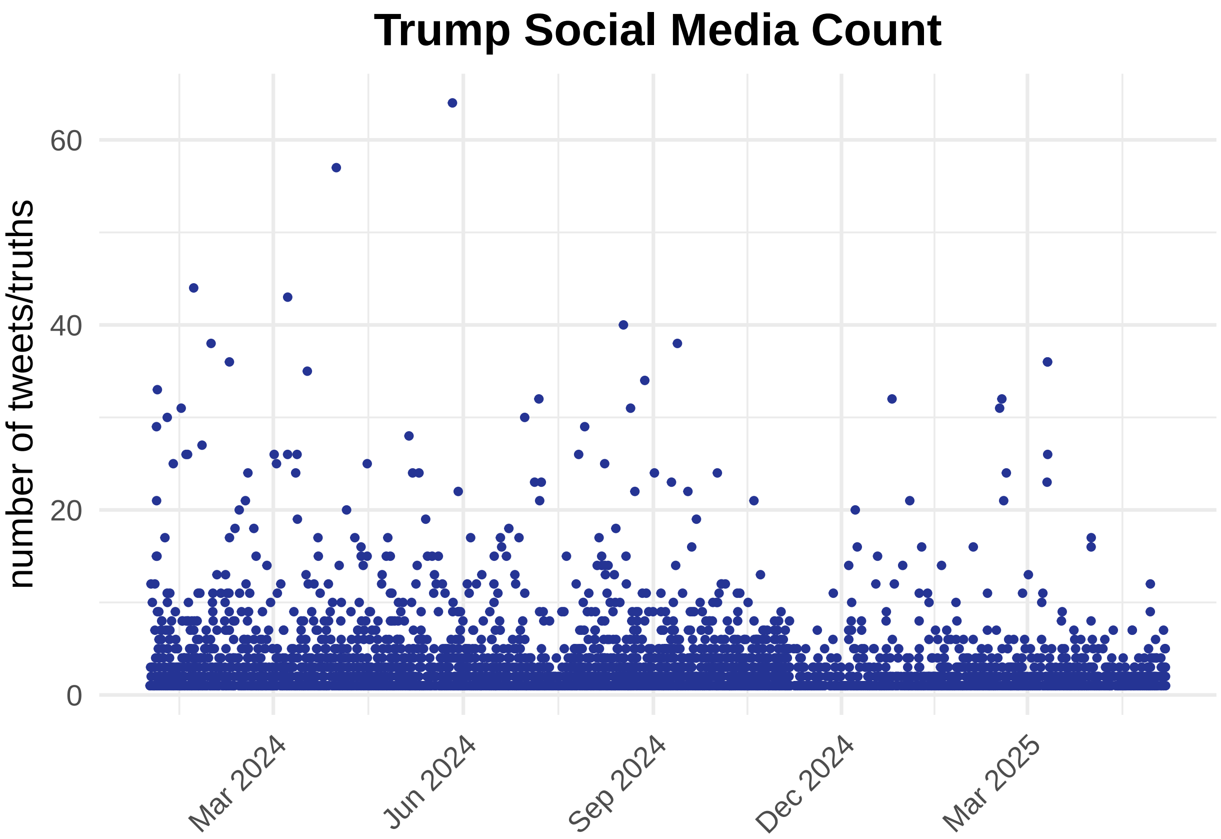


```r
#find hourly volatility
#NOTE: this ignores tweets made outside trading hours!!
VGK_volatility_alltime = dplyr::select(VGK,timestamp,r_vol_h)

#aggregating per hour
VGK_volatility_alltime = VGK_volatility_alltime %>%
          mutate(timestamp = floor_date(timestamp, unit = "hour")) %>%
          distinct(timestamp, .keep_all = TRUE)

#select time period
VGK_volatility = filter(VGK_volatility_alltime,
                  between(timestamp,
                          as.Date('2024-01-01'),
                          as.Date('2025-05-07')))
```

```r
#find hourly volatility
#NOTE: this ignores tweets made outside trading hours!!
ASHR_volatility_alltime = dplyr::select(ASHR,timestamp,r_vol_h)

#aggregating per hour
ASHR_volatility_alltime = ASHR_volatility_alltime %>%
          mutate(timestamp = floor_date(timestamp, unit = "hour")) %>%
          distinct(timestamp, .keep_all = TRUE)

#select time period
ASHR_volatility = filter(ASHR_volatility_alltime,
                  between(timestamp,
                          as.Date('2024-01-01'),
                          as.Date('2025-05-07')))
```

Number of Posts

```
#find count
tweetcount_alltime = dplyr::select(social_hourly,timestamp,N)

#select time period
tweetcount = filter(tweetcount_alltime,
                  between(timestamp,
                          as.Date('2024-01-01'),
                          as.Date('2025-05-07')))

#plot
ggplot(tweetcount, aes(x = timestamp, y = N)) +
    geom_point(color = "#253494", size = 1) +
    scale_x_datetime(date_labels = "%b %Y", date_breaks = "3 month") +
    labs(title = "Trump Social Media Count",
         x = NULL,
         y = "number of tweets/truths") +
    theme_minimal(base_size = 14) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1),
          plot.title = element_text(face = "bold", hjust = 0.5))
```



Dummy for Social Media Post

```r
#find dummy
tweetdummy_alltime = dplyr::select(social_hourly,timestamp,dummy)

#select time period
tweetdummy = filter(tweetdummy_alltime,
                    between(timestamp,
                            as.Date('2024-01-01'),
                            as.Date('2025-05-07')))
```

Number of Tweets Mentioning Tariffs

```r
#find count
tariff_alltime = dplyr::select(social_hourly,timestamp,total_tariff)

#select time period
tariff = filter(tariff_alltime,
                    between(timestamp,
                            as.Date('2024-01-01'),
                            as.Date('2025-05-07')))
```

Number of Tweets Mentioning Trade

```r
#find count
trade_alltime = dplyr::select(social_hourly,timestamp,total_trade)

#select time period
trade = filter(trade_alltime,
                  between(timestamp,
                          as.Date('2024-01-01'),
                          as.Date('2025-05-07')))
```

Proportion of Positive

```r
#find count
positive_alltime = dplyr::select(social_hourly,timestamp,prop_positive)

#select time period
positive = filter(positive_alltime,
                  between(timestamp,
                          as.Date('2024-01-01'),
                          as.Date('2025-05-07')))
```

Proportion of Negative

```r
#find count
negative_alltime = dplyr::select(social_hourly,timestamp,prop_negative)

#select time period
negative = filter(negative_alltime,
                  between(timestamp,
                          as.Date('2024-01-01'),
                          as.Date('2025-05-07')))
```

Merge

```r
#merge our dependant and independant vars
armax_data = left_join(SPY_volatility, VGK_volatility, by="timestamp")
armax_data = left_join(armax_data, ASHR_volatility, by="timestamp")
armax_data = left_join(armax_data, tweetdummy, by="timestamp")
armax_data = left_join(armax_data, tweetcount, by="timestamp")
armax_data = left_join(armax_data, tariff, by="timestamp")
armax_data = left_join(armax_data, trade, by="timestamp")
armax_data = left_join(armax_data, positive, by="timestamp")
armax_data = left_join(armax_data, negative, by="timestamp")

#rename volatility columns
names(armax_data)[2] <- "SPY_vol"
names(armax_data)[3] <- "VGK_vol"
names(armax_data)[4] <- "ASHR_vol"

#convert NA to zeroes
armax_data$N[is.na(armax_data$N)] = 0
armax_data$dummy[is.na(armax_data$dummy)] = 0
armax_data$total_tariff[is.na(armax_data$total_tariff)] = 0
armax_data$total_trade[is.na(armax_data$total_trade)] = 0
armax_data$prop_positive[is.na(armax_data$prop_positive)] = 0
armax_data$prop_negative[is.na(armax_data$prop_negative)] = 0
```

S&P500 ARMA-X Tariff Models

Finding Model

```
#auto.armax selects the lowest AIC value given r (exogenous variable lags)
res1 = auto.armax(armax_data$SPY_vol,xreg=armax_data$total_tariff,nb.lags=7,
                latex=F,max.p = 7, max.q = 7, max.d=0)
```

```
##
## =================================
##                      Model 1
## ---------------------------------
## ar1                    0.9758 ***
##                       (0.0063)
## ma1                   -0.6906 ***
##                       (0.0217)
## ma2                   -0.1800 ***
##                       (0.0214)
## intercept              0.0543 *
##                       (0.0228)
## total_tariff_lag_0    -0.0066
##                       (0.0113)
## total_tariff_lag_1    -0.0131
##                       (0.0116)
## total_tariff_lag_2     0.0359 **
##                       (0.0117)
## total_tariff_lag_3    -0.0049
##                       (0.0117)
## total_tariff_lag_4     0.0044
##                       (0.0117)
## total_tariff_lag_5     0.0037
##                       (0.0116)
## total_tariff_lag_6    -0.0188
##                       (0.0115)
## total_tariff_lag_7    -0.0141
##                       (0.0112)
## ---------------------------------
## AIC                 -674.3212
## AICc                -674.1655
## BIC                 -599.4019
## Log Likelihood       350.1606
## Num. obs.           2352
## =================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
#armax enables a custom armax specification with p,q,r
res2 = armax(armax_data$SPY_vol, xreg=armax_data$total_tariff, nb.lags=2,
                   p=5, q=0, d=0, latex=F)
```

```
##
## =================================
##                      Model 1
## ---------------------------------
## ar1                    0.3224 ***
##                       (0.0206)
## ar2                    0.0329
##                       (0.0219)
## ar3                    0.1113 ***
##                       (0.0224)
## ar4                    0.0896 ***
##                       (0.0223)
## ar5                    0.0460 *
##                       (0.0208)
## intercept              0.0539 ***
##                       (0.0110)
## total_tariff_lag_0    -0.0128
##                       (0.0115)
## total_tariff_lag_1    -0.0250 *
##                       (0.0122)
## total_tariff_lag_2     0.0312 **
##                       (0.0114)
## ---------------------------------
## AIC                 -597.9298
## AICc                -597.8360
## BIC                 -540.2783
## Log Likelihood       308.9649
## Num. obs.           2357
## =================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
#auto.armax.r selects the lowest AIC checking all 3 p,q,r values
res3 = auto.armax.r(armax_data$SPY_vol, x=armax_data$total_tariff,
                max_p = 7, max_q = 7, max_r = 3, criterion = "AIC", latex=F)
```

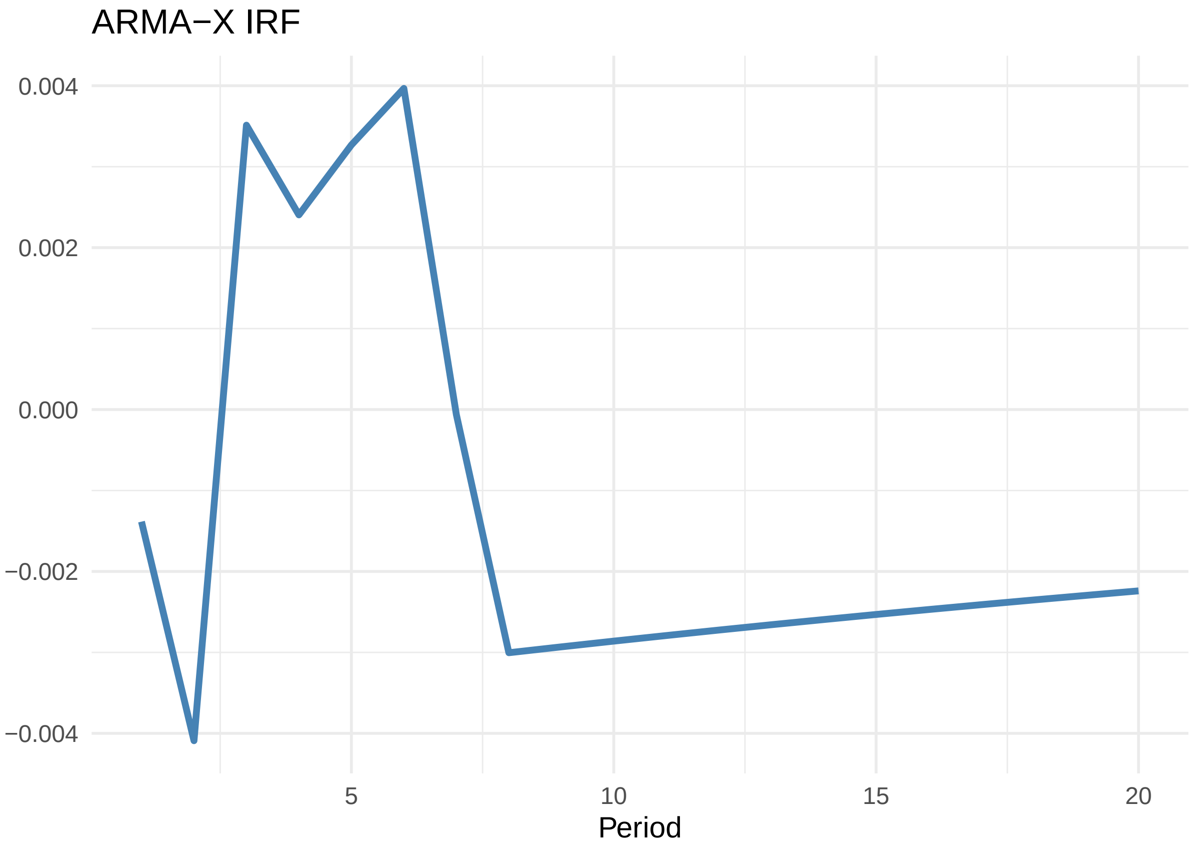
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning in sqrt(diag(model$var.coef)): NaNs produced
```

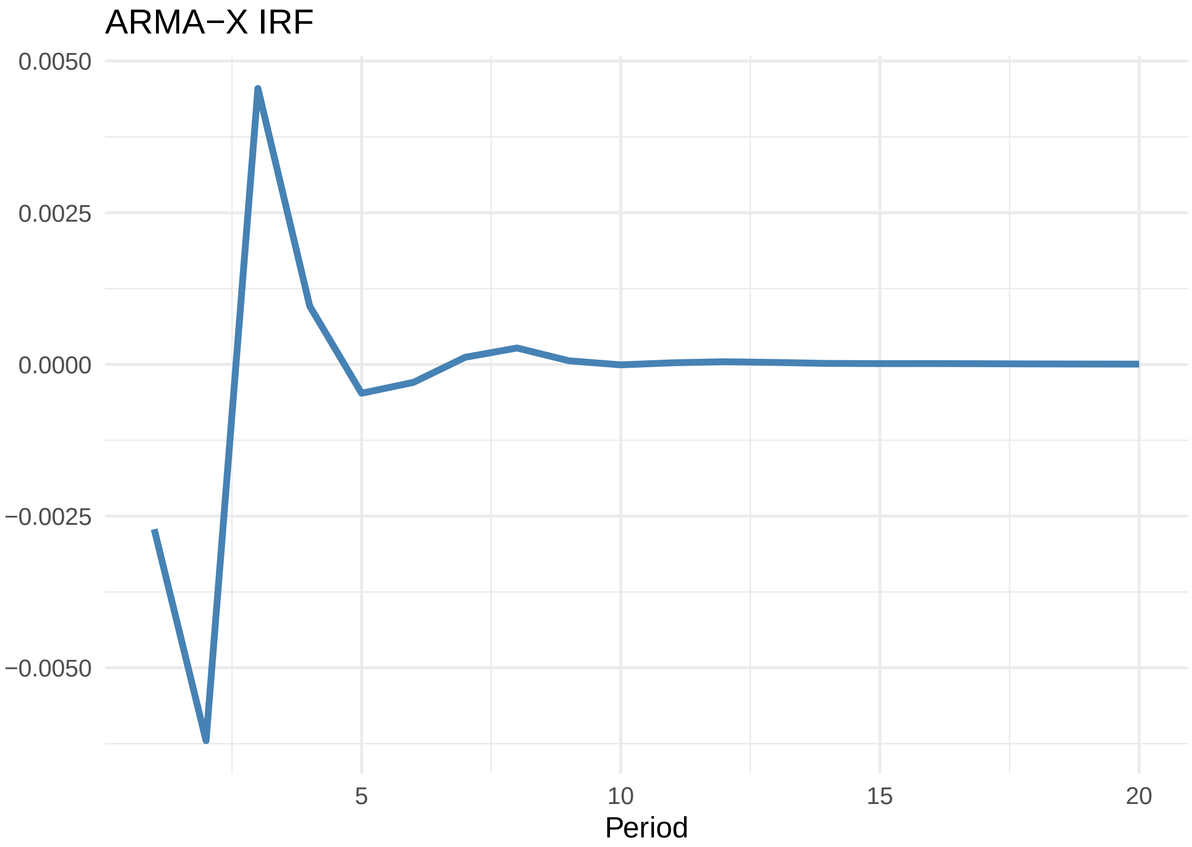
```
##
## =================================
##                      Model 1
## ---------------------------------
## ar1                    1.0801
##
## ar2                   -0.4288
##
## ar3                    0.4976 ***
##                       (0.0488)
## ar4                   -0.3019 ***
##                       (0.0281)
## ar5                    0.7924 ***
##                       (0.0293)
## ar6                   -0.6814 ***
##                       (0.0150)
## ma1                   -0.7735 ***
##                       (0.0094)
## ma2                    0.1292 ***
##                       (0.0303)
## ma3                   -0.3723 ***
##                       (0.0240)
## ma4                    0.2371 ***
##                       (0.0211)
## ma5                   -0.9602 ***
##                       (0.0095)
## ma6                    0.6554
##
## ma7                    0.2287 ***
##                       (0.0181)
## intercept              0.0507 ***
##                       (0.0138)
## total_tariff_lag_0     0.0030
##                       (0.0086)
## total_tariff_lag_1    -0.0101
##                       (0.0094)
## total_tariff_lag_2     0.0112
##                       (0.0088)
## ---------------------------------
## AIC                 -940.4859
## AICc                -940.1933
## BIC                 -836.7133
## Log Likelihood       488.2429
## Num. obs.           2357
## =================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Plotting IRFs

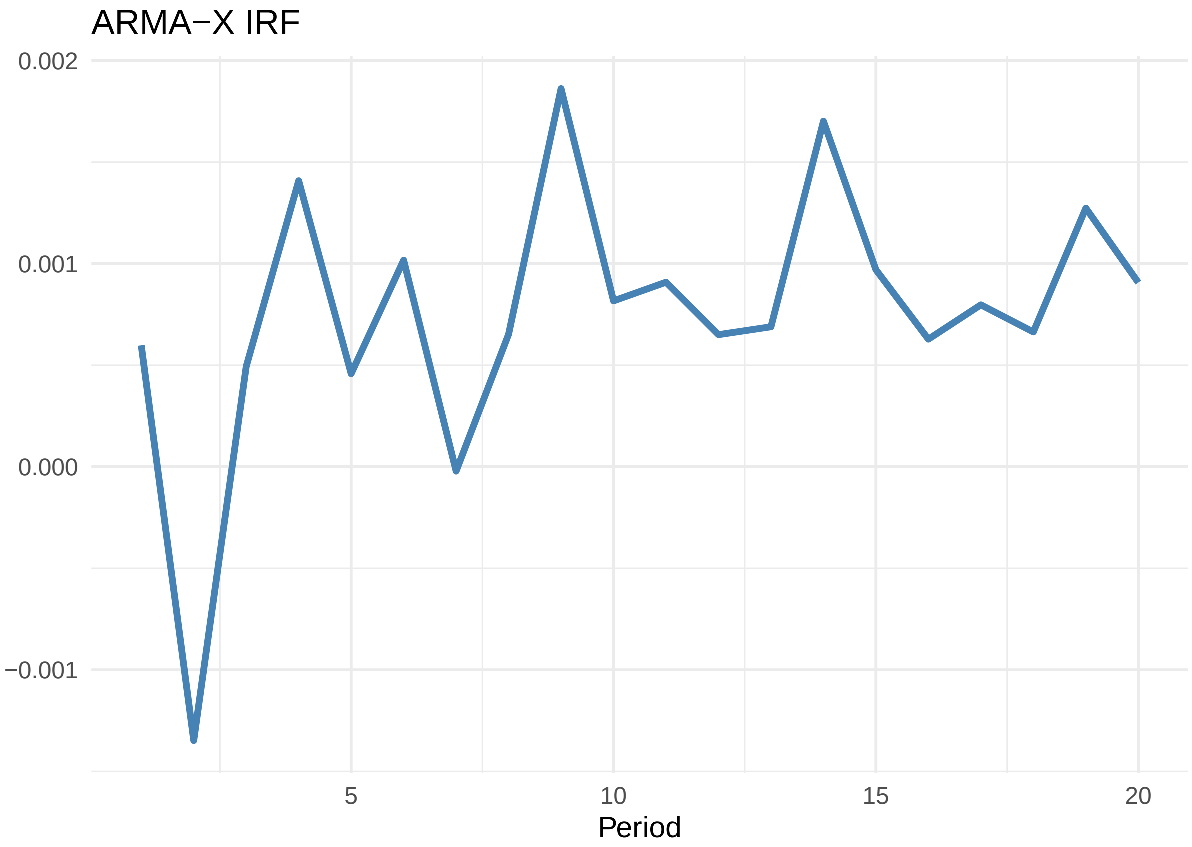
```
nb.periods = 20

irf.plot(res1,nb.periods)
```



```
irf.plot(res2,nb.periods)
```



```
irf.plot(res3$model,nb.periods)
```


ARMA−X IRF
0.002
0.001
0.000
−0.001
5
10
15
20
Period