ARMA-X Analysis Tutorial

Contents

Data

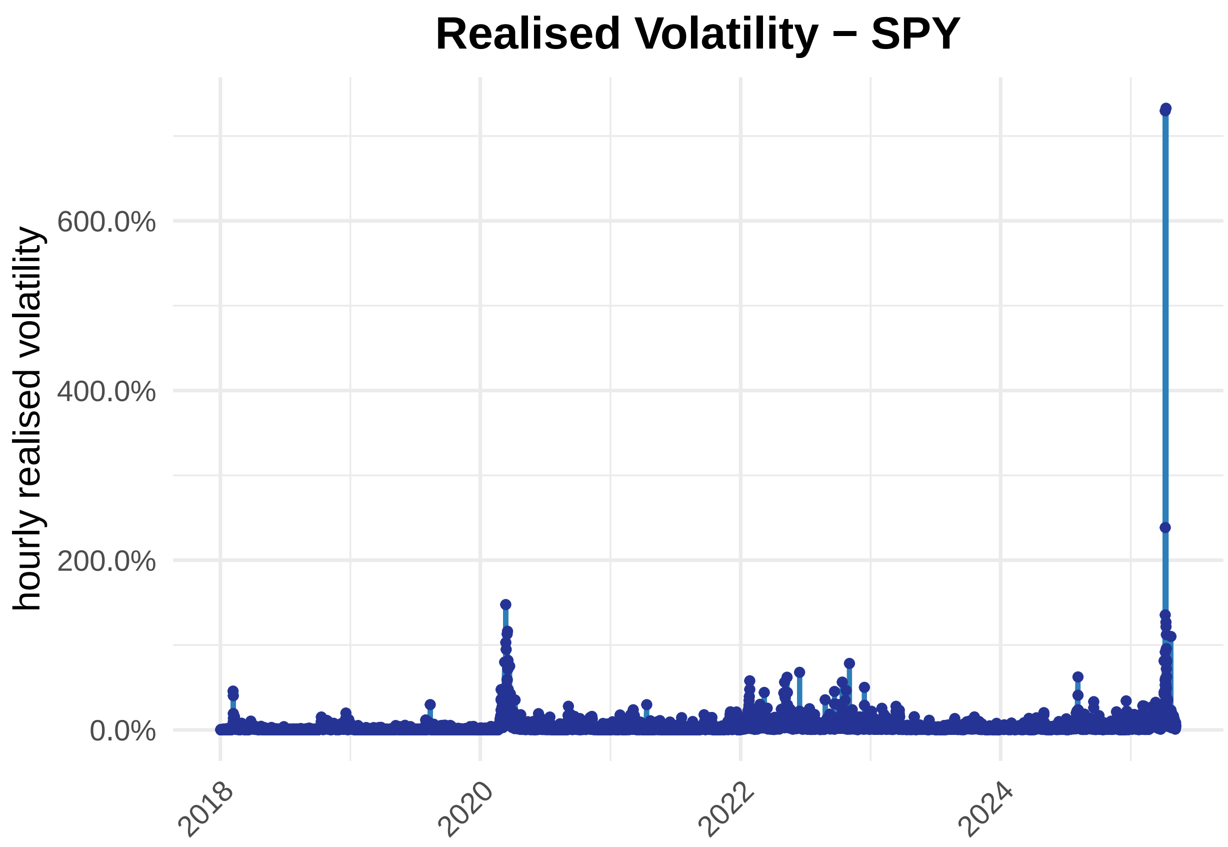
Load Base Data

```
# 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")
social_hourly$adjusted_time = as.POSIXct(social_hourly$adjusted_time,format = "%Y-%m-%d %H:%M:%S")

#select timeframe
SPY = filter(SPY,between(timestamp, as.Date('2018-01-01'), as.Date('2025-05-07')))
VGK = filter(VGK,between(timestamp, as.Date('2018-01-01'), as.Date('2025-05-07')))
ASHR = filter(ASHR,between(timestamp, as.Date('2018-01-01'), as.Date('2025-05-07')))
social = filter(social,between(timestamp, as.Date('2018-01-01'), as.Date('2025-05-07')))
social_hourly = filter(social_hourly,between(timestamp, as.Date('2018-01-01'), as.Date('2025-05-07')))
```

Volatility

```
#find hourly volatility
SPY_volatility = dplyr::select(SPY,timestamp,r_vol_h)

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

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



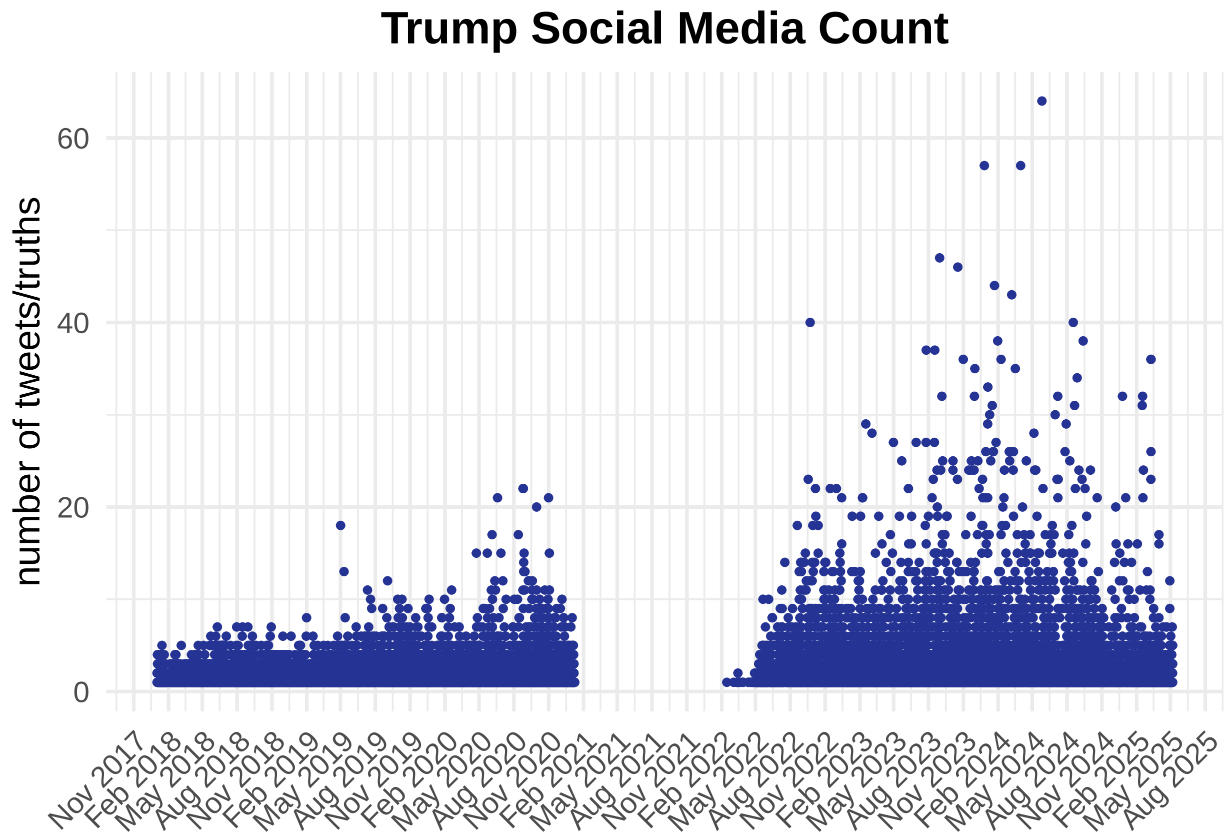
```r
#find hourly volatility
VGK_volatility = dplyr::select(VGK,timestamp,r_vol_h)

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

```r
#find hourly volatility
ASHR_volatility = dplyr::select(ASHR,timestamp,r_vol_h)

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

Number of Posts

```r
#find count
tweetcount = dplyr::select(social_hourly,timestamp,adjusted_time,N)
```

```
#for taking count of closed market hours
tweetcount2 <- tweetcount %>%
  group_by(adjusted_time) %>%
  summarise(N = sum(N))

#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

```
#find dummy
tweetdummy = dplyr::select(social_hourly,timestamp,adjusted_time,dummy)

#for taking count of closed market hours
tweetdummy2 <- tweetdummy %>%
  group_by(adjusted_time) %>%
  summarise(dummy = sum(dummy))
#peculiar interpretation for dummy: if dummy>1 it means that there were x
#out-hours which had tweets in them
```

Number of Tweets Mentioning Tariffs

```
#find count
tariff = dplyr::select(social_hourly,timestamp,adjusted_time,total_tariff)

#for taking count of closed market hours
tariff2 <- tariff %>%
  group_by(adjusted_time) %>%
  summarise(total_tariff = sum(total_tariff))
```

Number of Tweets Mentioning Trade

```
#find count
trade = dplyr::select(social_hourly,timestamp,adjusted_time,total_trade)

#for taking count of closed market hours
trade2 <- trade %>%
  group_by(adjusted_time) %>%
  summarise(total_trade = sum(total_trade))
```

Number of Tweets Mentioning China

```
#find count
china = dplyr::select(social_hourly,timestamp,adjusted_time,total_china)

#for taking count of closed market hours
china2 <- china %>%
  group_by(adjusted_time) %>%
  summarise(total_china = sum(total_china))
```

Proportion of Positive

```
#find count
positive = dplyr::select(social_hourly,timestamp,adjusted_time,prop_positive)

#how to count outside hours? since proportion?
```

Proportion of Negative

```
#find count
negative = dplyr::select(social_hourly,timestamp,adjusted_time,prop_negative)
```

Merge

```
#merge our dependant and independant vars

#case 1: ignore tweets outside trading hours
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, select(tweetdummy, -adjusted_time), by="timestamp")
armax_data = left_join(armax_data, select(tweetcount, -adjusted_time), by="timestamp")
armax_data = left_join(armax_data, select(tariff, -adjusted_time), by="timestamp")
armax_data = left_join(armax_data, select(trade, -adjusted_time), by="timestamp")
armax_data = left_join(armax_data, select(china, -adjusted_time), by="timestamp")
armax_data = left_join(armax_data, select(positive, -adjusted_time), by="timestamp")
armax_data = left_join(armax_data, select(negative, -adjusted_time), by="timestamp")

rm(armax_data)
#case 2: push tweets made outside market hours to the next open hour
armax_data = left_join(SPY_volatility, VGK_volatility, by="timestamp")
armax_data = left_join(armax_data, ASHR_volatility, by="timestamp")
armax_data <- armax_data %>%
  left_join(tweetdummy2, by = c("timestamp" = "adjusted_time"))
armax_data <- armax_data %>%
  left_join(tweetcount2, by = c("timestamp" = "adjusted_time"))
armax_data <- armax_data %>%
  left_join(tariff2, by = c("timestamp" = "adjusted_time"))
armax_data <- armax_data %>%
  left_join(trade2, by = c("timestamp" = "adjusted_time"))
armax_data <- armax_data %>%
  left_join(china2, by = c("timestamp" = "adjusted_time"))

#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$total_china[is.na(armax_data$total_china)] = 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 Univariate ARMA-X Models

Tweet Dummy as Exogenous

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

================================= Model 1
——————————————- ar1 0.9812 ***(0.0023)***
ma1 -0.6788 (0.0091)
ma2 -0.2104 ***(0.0108)***
ma3 -0.0105
(0.0100)
ma4 0.0322 (0.0088)
intercept 0.0325 ***(0.0061)***
dummy_lag_0 0.0012 (0.0003)
dummy_lag_1 0.0007 *
(0.0003)
dummy_lag_2 -0.0003
(0.0003)
dummy_lag_3 -0.0009 ** (0.0003)
dummy_lag_4 -0.0007 *
(0.0003)
dummy_lag_5 -0.0006 *
(0.0003)
dummy_lag_6 0.0000
(0.0003)
dummy_lag_7 0.0008 ** (0.0003)
——————————————- AIC -24010.3169
AICc -24010.2797
BIC -23898.3247
Log Likelihood 12020.1584
Num. obs. 12915
================================= *** p < 0.001; ** p < 0.01; * p < 0.05

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

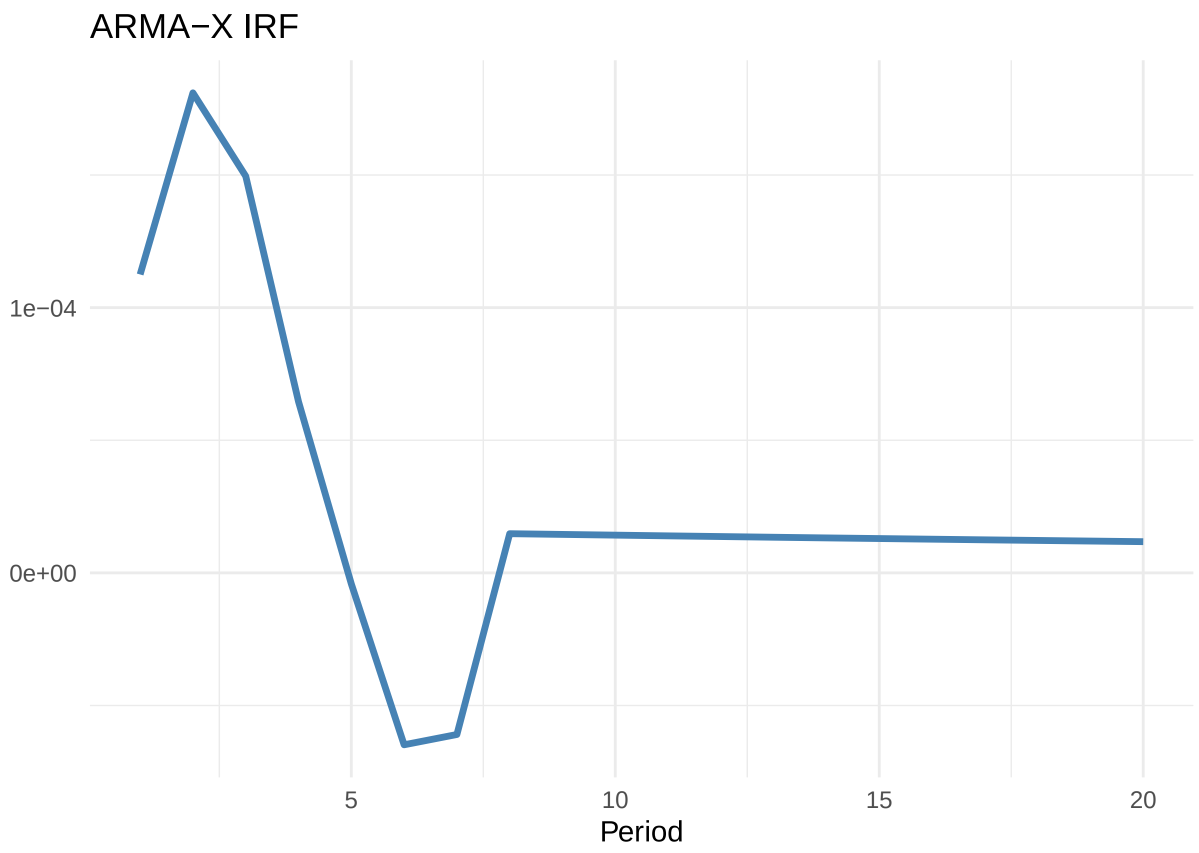
================================= Model 1
——————————————- ar1 0.3536 ***(0.0088)***
ar2 0.0393 (0.0093)
ar3 0.0970 ***(0.0092)***
ar4 0.1025 (0.0093)
ar5 0.0778 ***(0.0088)***
intercept 0.0291 (0.0027)
dummy_lag_0 0.0019 ***(0.0003)***
dummy_lag_1 0.0011 (0.0003)
dummy_lag_2 0.0001
(0.0003)
——————————————- AIC -23390.9170

AICc -23390.9000
BIC -23316.2517
Log Likelihood 11705.4585
Num. obs. 12920
================================== *** p < 0.001; ** p < 0.01; * p < 0.05

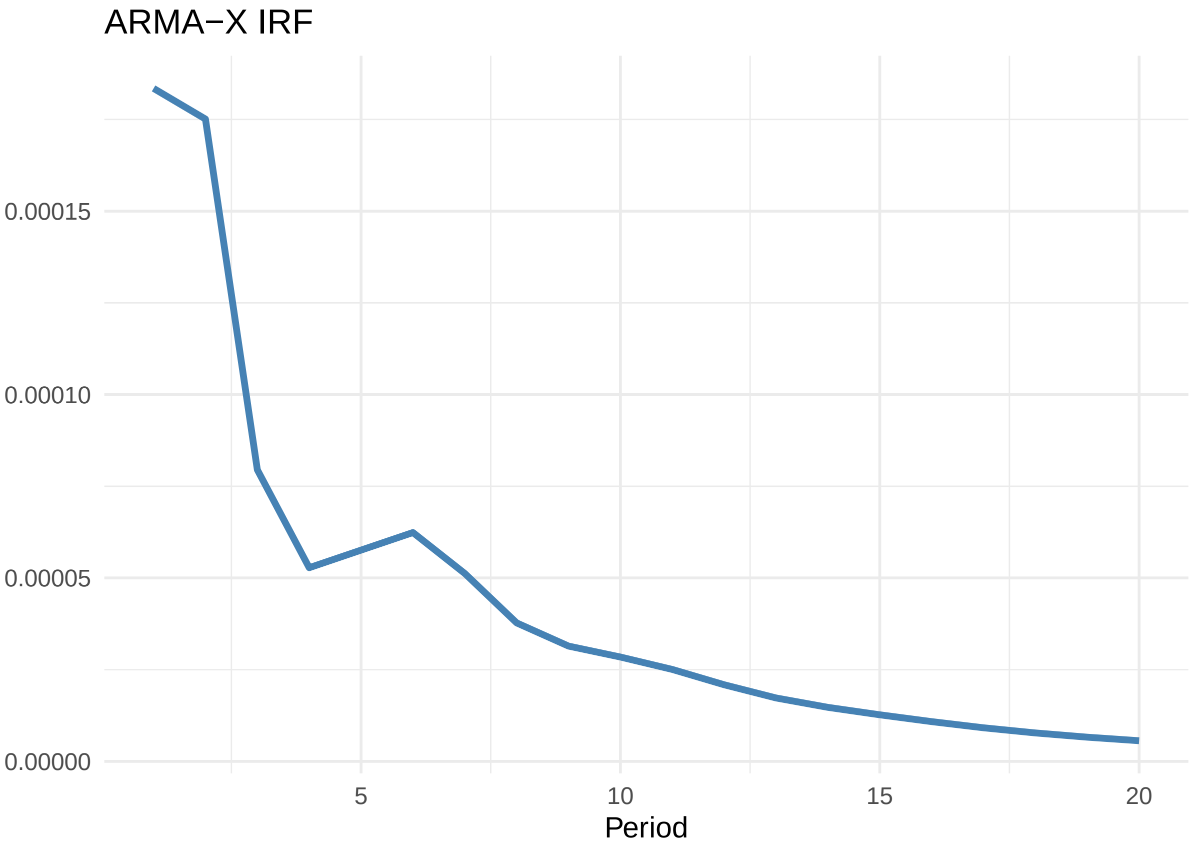
```r
#auto.armax.r selects the lowest AIC checking all 3 p,q,r values
res3 = auto.armax.r(armax_data$SPY_vol, x=armax_data$dummy,
                    max_p = 7, max_q = 7, max_r = 3, criterion = "AIC", latex=F)
```

================================== Model 1
—————————————- ar1 -0.8978 ***(0.0157)***
ar2 -0.5791 (0.0171)
ar3 -0.1483 ***(0.0155)***
ar4 0.3603 (0.0119)
ar5 0.6161 ***(0.0152)***
ar6 0.8037 (0.0150)
ar7 0.6210 ***(0.0125)***
ma1 1.1949 (0.0122)
ma2 0.9904 ***(0.0176)***
ma3 0.5643 (0.0204)
ma4 -0.0241
(0.0181)
ma5 -0.4948 ***(0.0165)***
ma6 -0.8424 (0.0138)
ma7 -0.7519 ***(0.0084)***
intercept 0.0303 (0.0059)
dummy_lag_0 0.0015 ***(0.0002)***
dummy_lag_1 0.0007 *(0.0002)*
—————————————- AIC -24830.1606
AICc -24830.1076
BIC -24695.7617
Log Likelihood 12433.0803
Num. obs. 12921
================================== ** p < 0.001; ** p < 0.01; * p < 0.05

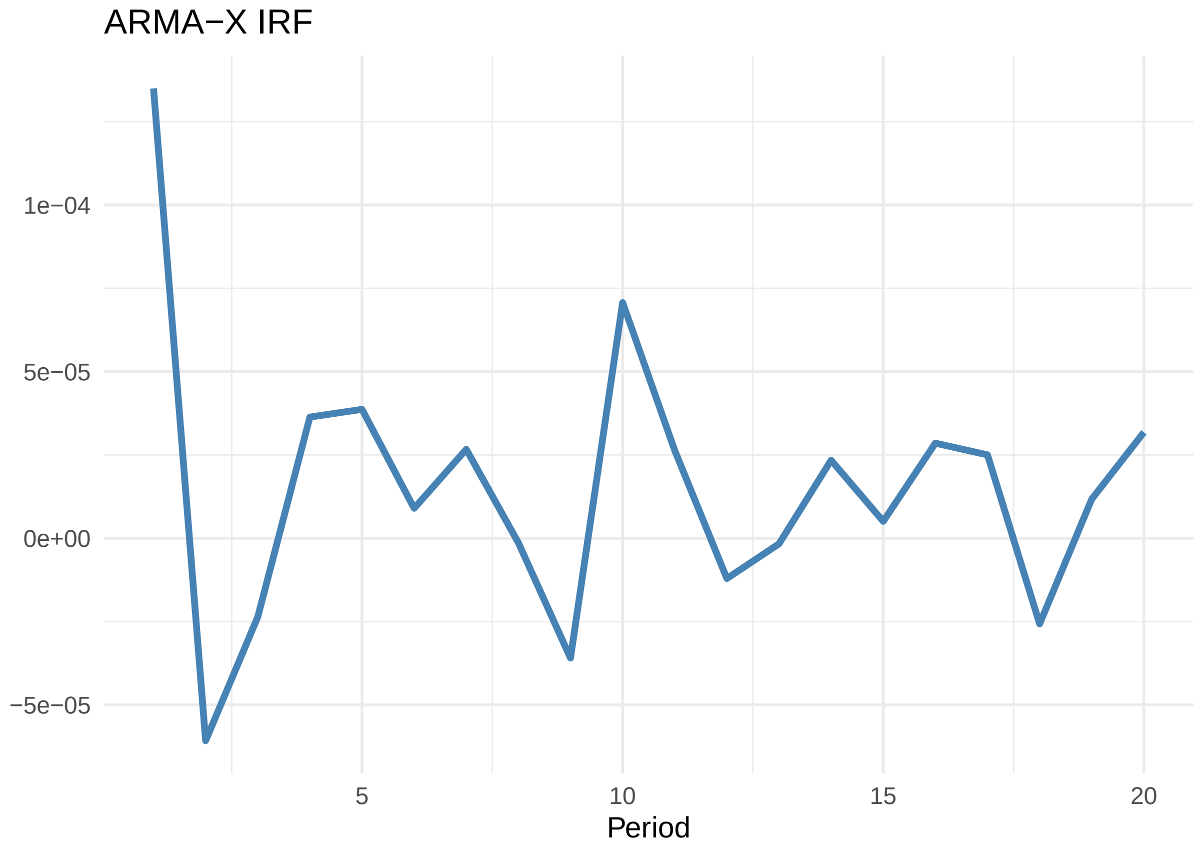
```r
#we want to plot the IRFs of these models
nb.periods = 20

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



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



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



Tweet Count as Exogenous

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

================================ Model 1
————————————- ar1 0.9812 ***(0.0023)***
ma1 -0.6781 (0.0091)
ma2 -0.2117 ***(0.0108)***
ma3 -0.0108
(0.0100)
ma4 0.0330 (0.0088)
intercept 0.0328 ***(0.0060)***
N_lag_0 0.0003 *(0.0001)*
N_lag_1 0.0002
(0.0001)
N_lag_2 -0.0001
(0.0001)
N_lag_3 -0.0003 ** (0.0001)
N_lag_4 -0.0002 *
(0.0001)
N_lag_5 -0.0002
(0.0001)

N_lag_6 0.0000
(0.0001)
N_lag_7 0.0003 ** (0.0001)
———————————- AIC -23990.1944
AICc -23990.1572
BIC -23878.2023
Log Likelihood 12010.0972
Num. obs. 12915
================================= *** p < 0.001; ** p < 0.01; * p < 0.05

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

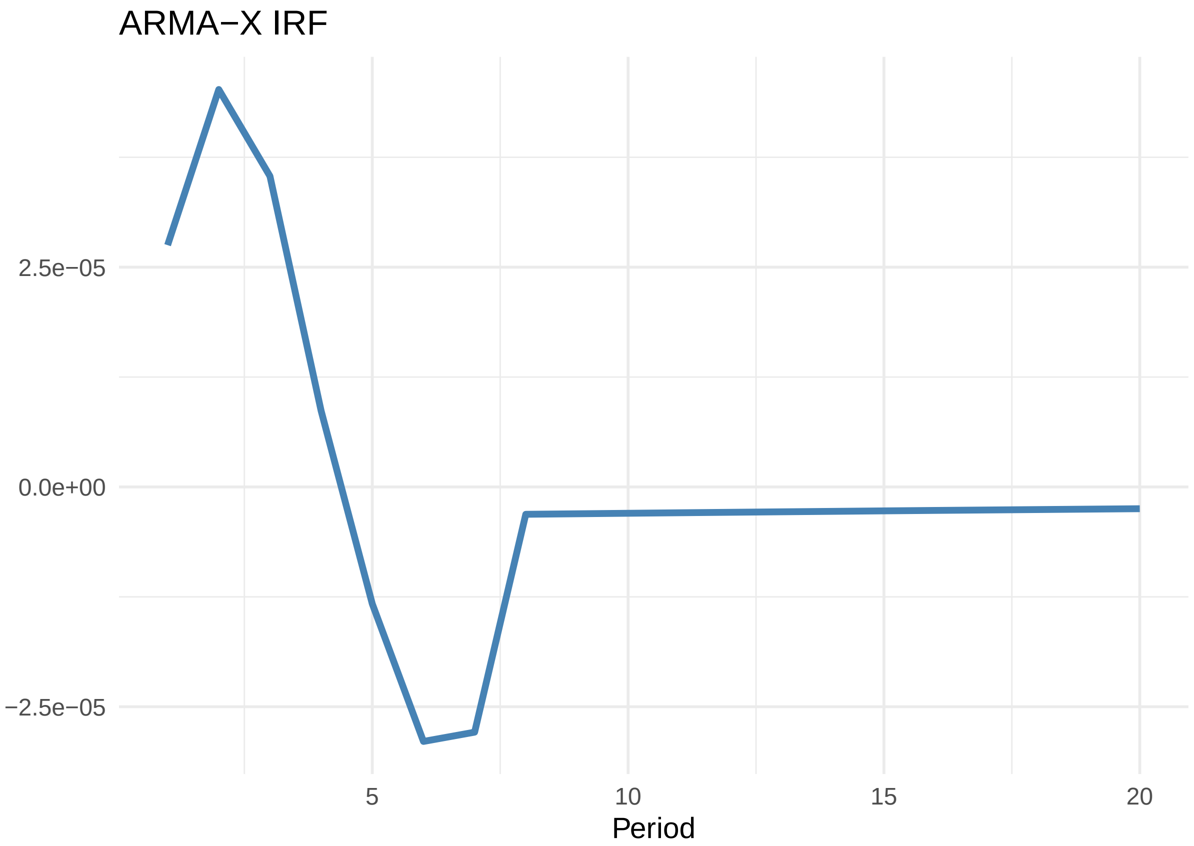
================================= Model 1
———————————- ar1 0.3547 ***(0.0088)***
ar2 0.0386 (0.0093)
ar3 0.0968 ***(0.0092)***
ar4 0.1019 (0.0093)
ar5 0.0778 ***(0.0088)***
intercept 0.0303 (0.0027)
N_lag_0 0.0005 ***(0.0001)***
N_lag_1 0.0003 *(0.0001)*
N_lag_2 0.0000
(0.0001)
———————————- AIC -23367.8843
AICc -23367.8672
BIC -23293.2189
Log Likelihood 11693.9421
Num. obs. 12920
================================= ** 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$SPY_vol, x=armax_data$N,
                    max_p = 7, max_q = 7, max_r = 3, criterion = "AIC", latex=F)
```

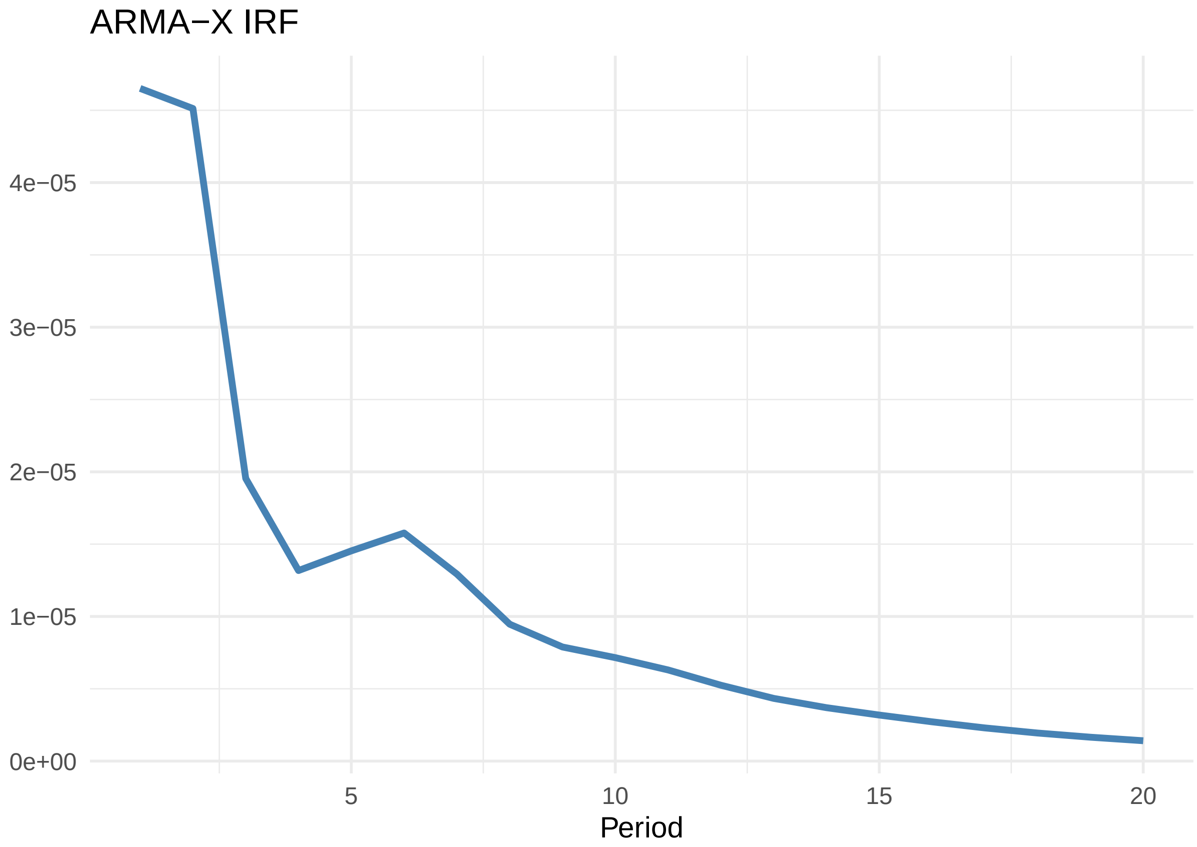
================================= Model 1
———————————- ar1 -0.8928 ***(0.0156)***
ar2 -0.5733 (0.0166)
ar3 -0.1386 ***(0.0152)***
ar4 0.3651 (0.0118)
ar5 0.6207 ***(0.0148)***
ar6 0.8064 (0.0148)
ar7 0.6184 ***(0.0127)***
ma1 1.1905 (0.0121)
ma2 0.9826 ***(0.0168)***
ma3 0.5509 (0.0196)
ma4 -0.0369 *
(0.0173)
ma5 -0.5071 ***(0.0158)***
ma6 -0.8523 (0.0133)
ma7 -0.7548 ***(0.0084)***
intercept 0.0310 (0.0066)

N_lag_0 0.0004 ***(0.0001)***
N_lag_1 0.0002 *(0.0001)*
——————————- AIC -24815.5647
AICc -24815.5117
BIC -24681.1657
Log Likelihood 12425.7823
Num. obs. 12921
================================= ** p < 0.001; ** p < 0.01; * p < 0.05

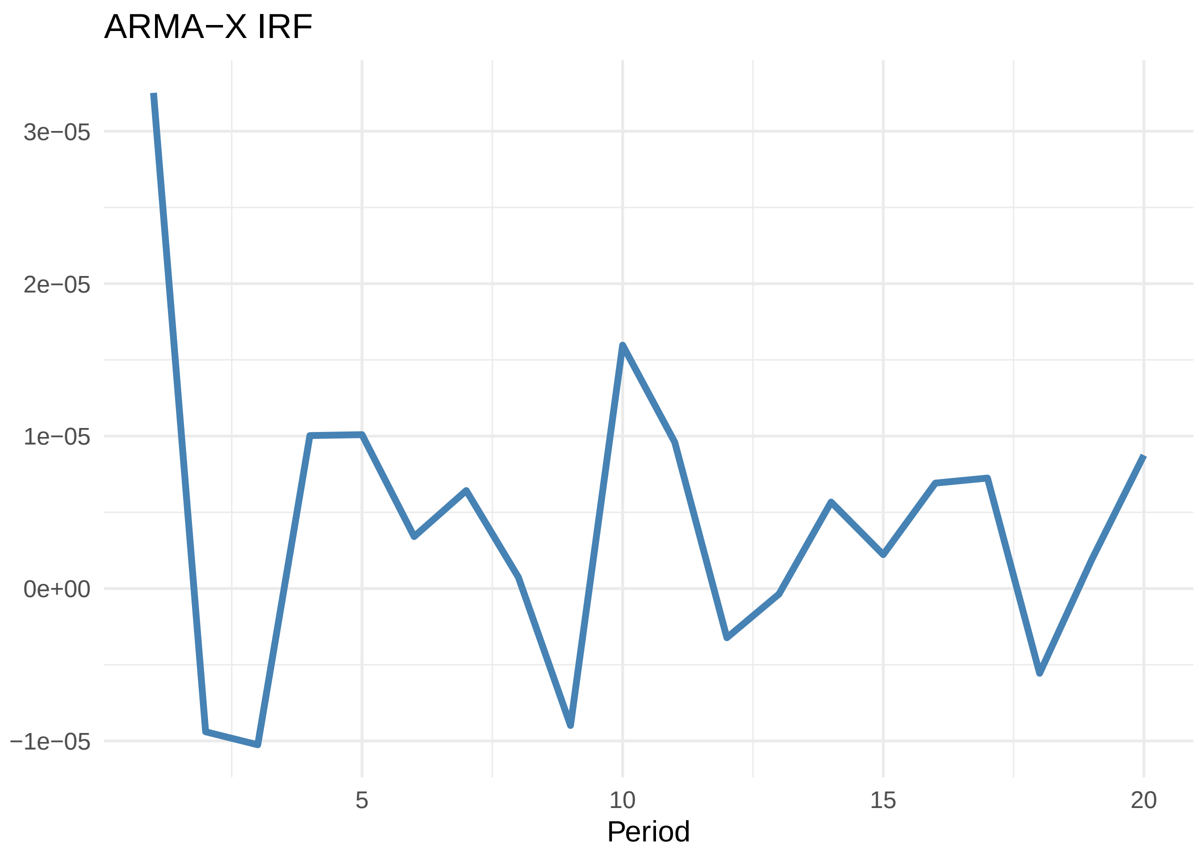
```r
#we want to plot the IRFs of these models
nb.periods = 20

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



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



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



Tariff as Exogenous

```
#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=2,
                  latex=F,max.p = 6, max.q = 6, max.d=0)
```

==================================== Model 1
——————————————— ar1 1.6305 ***(0.1227)***
ar2 -0.7981 (0.1433)
ar3 0.1575 ***(0.0238)***
ma1 -1.3306 (0.1236)
ma2 0.3989 ***(0.1112)***
intercept 0.0314 (0.0057)
total_tariff_lag_0 0.0044 *
(0.0018)
total_tariff_lag_1 0.0204 ***(0.0019)***
total_tariff_lag_2 0.0112 (0.0018)
——————————————— AIC -24097.0141
AICc -24096.9971
BIC -24022.3488
Log Likelihood 12058.5070
Num. obs. 12920
==================================== *** p < 0.001; ** p < 0.01; * p < 0.05

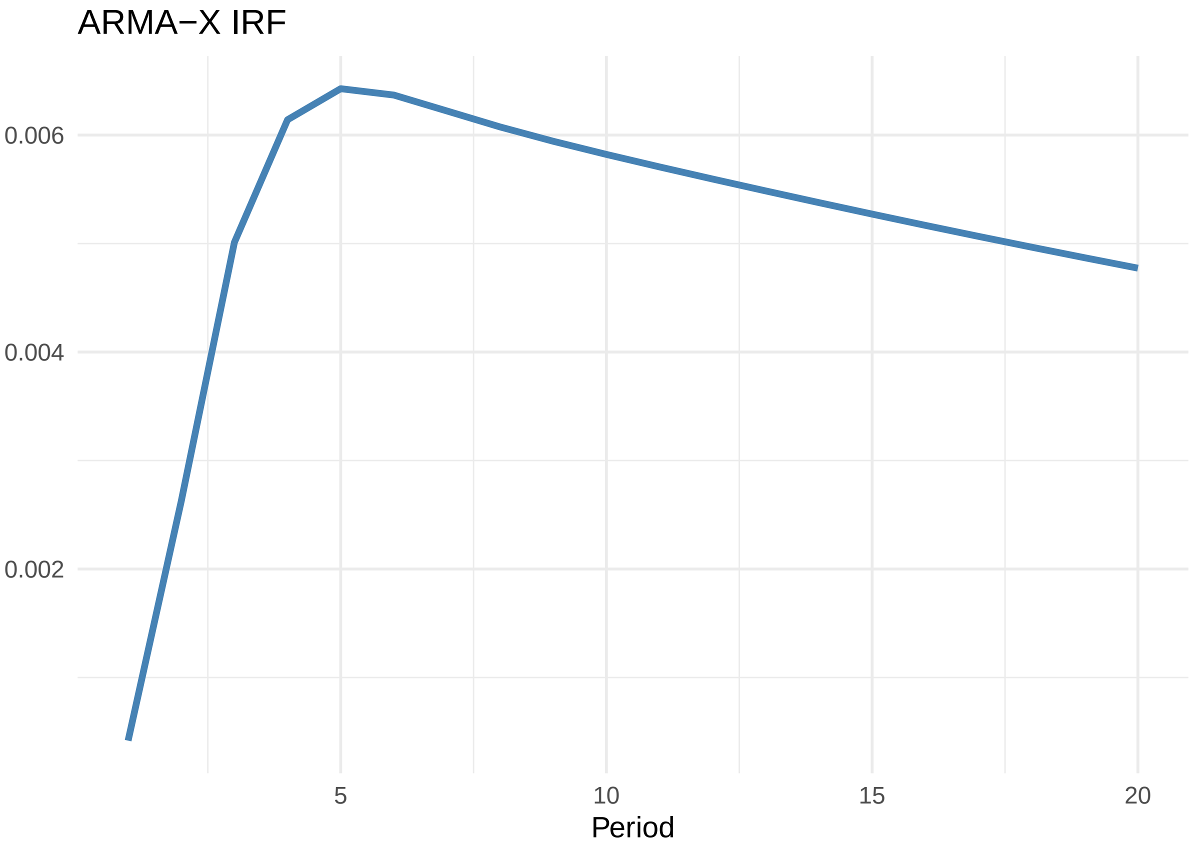
```r
#armax enables a custom armax specification with p,q,r
res2 = armax(armax_data$SPY_vol, xreg=armax_data$total_tariff, nb.lags=2,
                   p=5, q=0, d=0, latex=F)
```

```
====================================== Model 1
———————————————————— ar1 0.3538 (0.0088)
ar2 0.0402 (0.0093)
ar3 0.0877 (0.0093)
ar4 0.0955 (0.0093)
ar5 0.0825 (0.0088)
intercept 0.0313 (0.0025)
total_tariff_lag_0 0.0047 ** (0.0018)
total_tariff_lag_1 0.0202 (0.0019)
total_tariff_lag_2 0.0110 (0.0018)
———————————————————— AIC -23454.7592
AICc -23454.7422
BIC -23380.0939
Log Likelihood 11737.3796
Num. obs. 12920
====================================== *** p < 0.001; ** p < 0.01; * p < 0.05
```

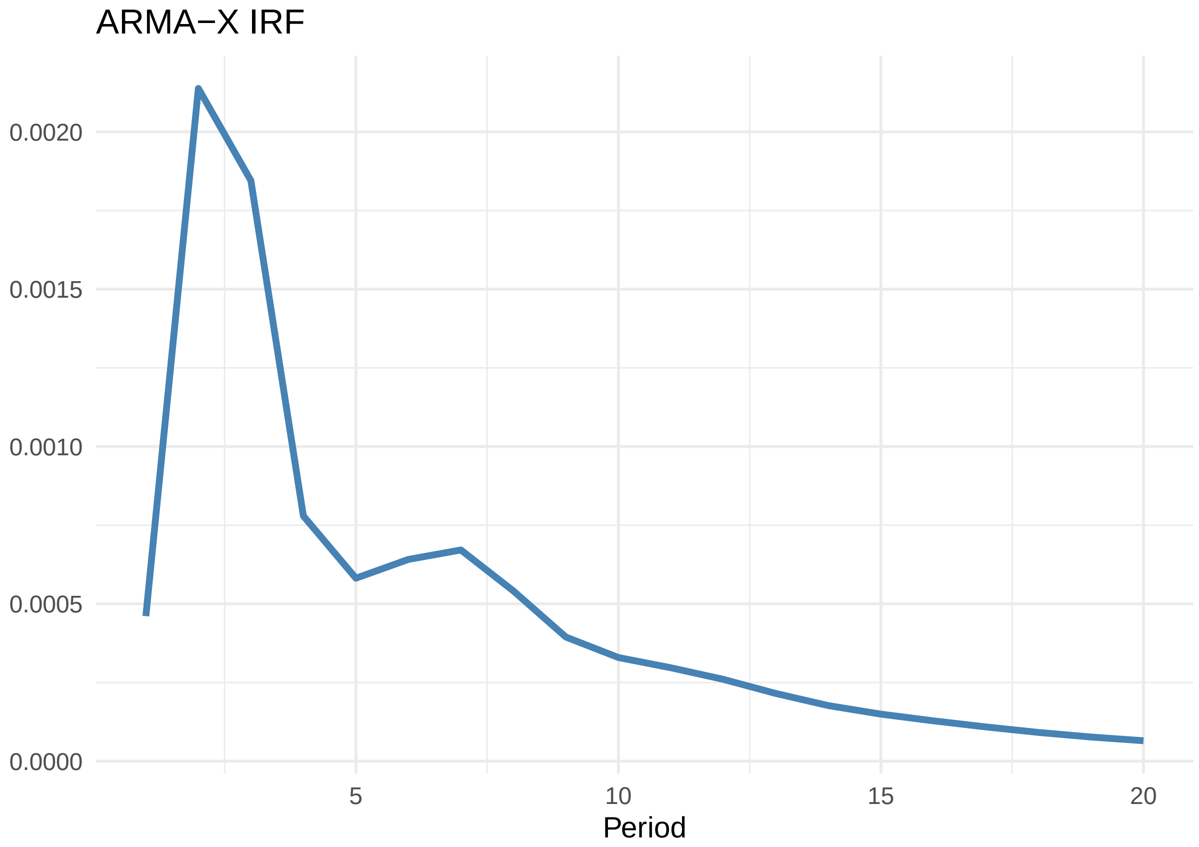
```r
#auto.armax.r selects the lowest AIC checking all 3 p,q,r values
res3 = auto.armax.r(armax_data$SPY_vol, x=armax_data$total_tariff,
                max_p = 6, max_q = 6, max_r = 6, criterion = "AIC", latex=F)
```

```
====================================== Model 1
———————————————————— ar1 -0.6528 (0.0121)
ar2 0.0290
(0.0155)
ar3 0.0152
(0.0109)
ar4 0.1308 (0.0140)
ar5 0.6401 (0.0134)
ar6 0.6896 (0.0094)
ma1 0.9531 (0.0092)
ma2 0.2800 (0.0160)
ma3 0.1789 (0.0147)
ma4 0.0650 (0.0124)
ma5 -0.6161 (0.0130)
ma6 -0.8008 (0.0071)
intercept 0.0316 (0.0058)
total_tariff_lag_0 0.0072 (0.0016)
total_tariff_lag_1 0.0165 (0.0017)
total_tariff_lag_2 0.0076 (0.0017)
total_tariff_lag_3 -0.0026
(0.0016)
———————————————————— AIC -24958.5939
AICc -24958.5409
BIC -24824.1977
Log Likelihood 12497.2969
Num. obs. 12919
====================================== *** p < 0.001; ** p < 0.01; * p < 0.05
```

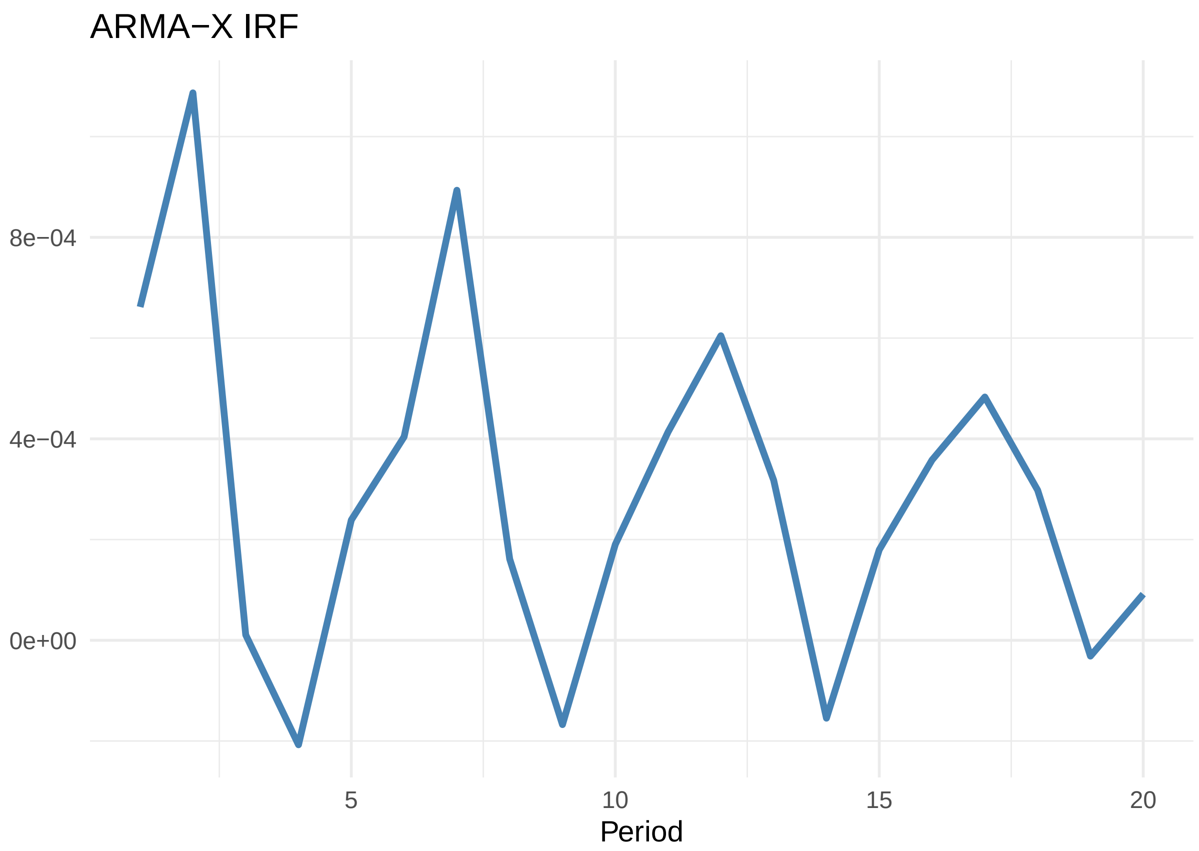
```
#we want to plot the IRFs of these models
nb.periods = 20

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



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



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



Interaction Terms

Dummy * Tariff

```r
#create X matrix with both exogenous regressors
X = cbind(armax_data$total_tariff,armax_data$N*armax_data$total_tariff)
colnames(X) <- c("Tariff","Tariff*Count")
head(X)

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

#we want to plot the IRFs of these models
nb.periods = 20

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