ASHR_SVAR

pte

2025-05-15

Contents

1 Setup

1.1 Load packages & functions

```
rm(list=ls())
require(tinytex) #LaTeX
require(ggplot2) #plots
require(AEC) #JP-Renne functions
require(AER) #NW formula
require(forecast) #time series stuff
require(expm) #matrix exponents
require(here) #directory finder
require(stringr) # analysis of strings, important for the detection in tweets
```

```r
require(dplyr) #data management
require(lubridate) #data dates management
require(zoo) #for lagging
require(jtools) #tables
require(huxtable) #tables
require(lmtest) #reg tests
require(vroom) #for loading data
require(data.table) #for data filtering
require(sysid) #for ARMA-X modeling
require(sandwhich) #regression errors
require(stargazer) #nice reg tables
require(tidytext) #text mining
require(textstem) #lemmatization
require(quanteda) #tokenization
require(texreg) #arima tables
require(vars) #VAR models
require(xts) #time series objects
require(tseries) #includes adf test
require(quantmod)
require(TSA)
require(aTSA)
require(tibble)
require(FinTS)
require(kableExtra)
require(writexl)
require(purrr)

getwd()
#setwd("...") -> set wd at base repo folder

#load helper functions
source(here("helperfunctions/data_loaders.R"))
source(here("helperfunctions/date_selector.R"))
source(here("helperfunctions/plotters.R"))
source(here("helperfunctions/quick_arma.R"))
source(here("helperfunctions/r.vol_calculators.R"))
source(here("helperfunctions/truths_cleaning_function.R"))
source(here("helperfunctions/armax_functions.R"))
source(here("helperfunctions/var_irf.R"))
```

1.2 Load Data

```r
#load final dataset
source(here("helperfunctions/full_data.R"))

#select timeframe
Vdata = filter(data,between(timestamp, as.Date('2014-01-01'), as.Date('2025-05-07')))
```

2 Some SVAR estimations

(this is not an exhaustive list of our VAR estimations, you can find more by going on /modeling/VAR/VAR_SPY_TRUE or VAR_ASHR_TRUE or VAR_VGK_TRUE)

2.1 Dummy variable

here we use a dummy variable which equal to one if Trump has made a post or 0 otherwise, taking into account the closed hour market posts.

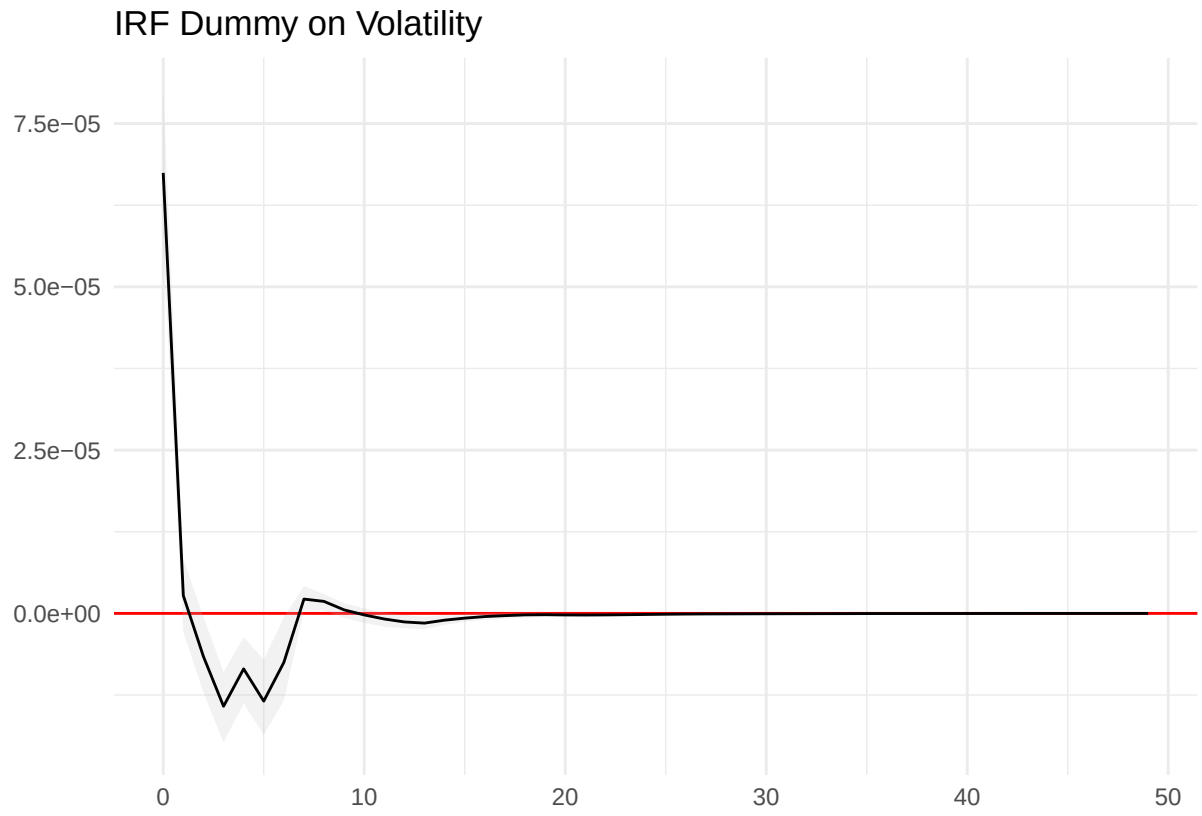
```
y = cbind(Vdata$dummy, Vdata$ASHR_vol)
colnames(y)[1:2] <- c("dummy", "vol")
est.VAR <- VAR(y,p=6)
mod_vol <- est.VAR$varresult$vol
screenreg(mod_vol, digits = 6)
```

```
##
## ============================
##            Model 1
## ----------------------------
## dummy.l1      -0.000006 ***
##               (0.000001)
## vol.l1         0.282482 ***
##               (0.007140)
## dummy.l2      -0.000005 ***
##               (0.000001)
## vol.l2         0.072926 ***
##               (0.007410)
## dummy.l3      -0.000006 ***
##               (0.000001)
## vol.l3         0.047892 ***
##               (0.007418)
## dummy.l4      -0.000004 ***
##               (0.000001)
## vol.l4         0.056084 ***
##               (0.007416)
## dummy.l5      -0.000006 ***
##               (0.000001)
## vol.l5         0.059763 ***
##               (0.007410)
## dummy.l6      -0.000005 ***
##               (0.000001)
## vol.l6         0.109466 ***
##               (0.007136)
## const          0.000095 ***
##               (0.000005)
## ----------------------------
## R^2            0.178208
## Adj. R^2       0.177714
## Num. obs.  19965
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
Omega <- var(residuals(est.VAR))

#make the B matrix
loss <- function(param){
  #Define the restriction
  B <- matrix(c(param[1], param[2], 0, param[3]), ncol = 2)

  #Make BB' approximatively equal to omega
  X <- Omega - B %*% t(B)

  #loss function
  loss <- sum(X^2)
  return(loss)
}

res.opt <- optim(c(1, 0, 1), loss, method = "BFGS")
B.hat <- matrix(c(res.opt$par[1], res.opt$par[2], 0, res.opt$par[3]), ncol = 2)

print(cbind(Omega,B.hat %*% t(B.hat)))
```

```
##              dummy          vol
## dummy 8.5222425009 1.968292e-04 8.5222418314 1.968705e-04
## vol   0.0001968292 1.631171e-07 0.0001968705 2.428876e-05
```

```r
#irf creation
irf_res <- irf(est.VAR, impulse = "dummy", response = "vol",
                  bmat=b.hat, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf <- extract_varirf(irf_res)

#the plot
single_varirf %>%
  ggplot(aes(x=period, y=irf_dummy_vol, ymin=lower_dummy_vol, ymax=upper_dummy_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF Dummy on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```




```
#does volatility Granger cause dummy mentions
grangertest(y[,c("vol","dummy")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	4.06	0.000446

```
#does dummy mentions Granger cause volatility
grangertest(y[,c("dummy", "vol")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	22.2	3.64e-26

2.2 Number of Post

```r
y2 = cbind(Vdata$N , Vdata$ASHR_vol)
colnames(y2)[1:2] <- c("N", "vol")
est.VAR2 <- VAR(y2,p=6)
mod_vol2 <- est.VAR2$varresult$vol
screenreg(mod_vol2, digits = 6)
```

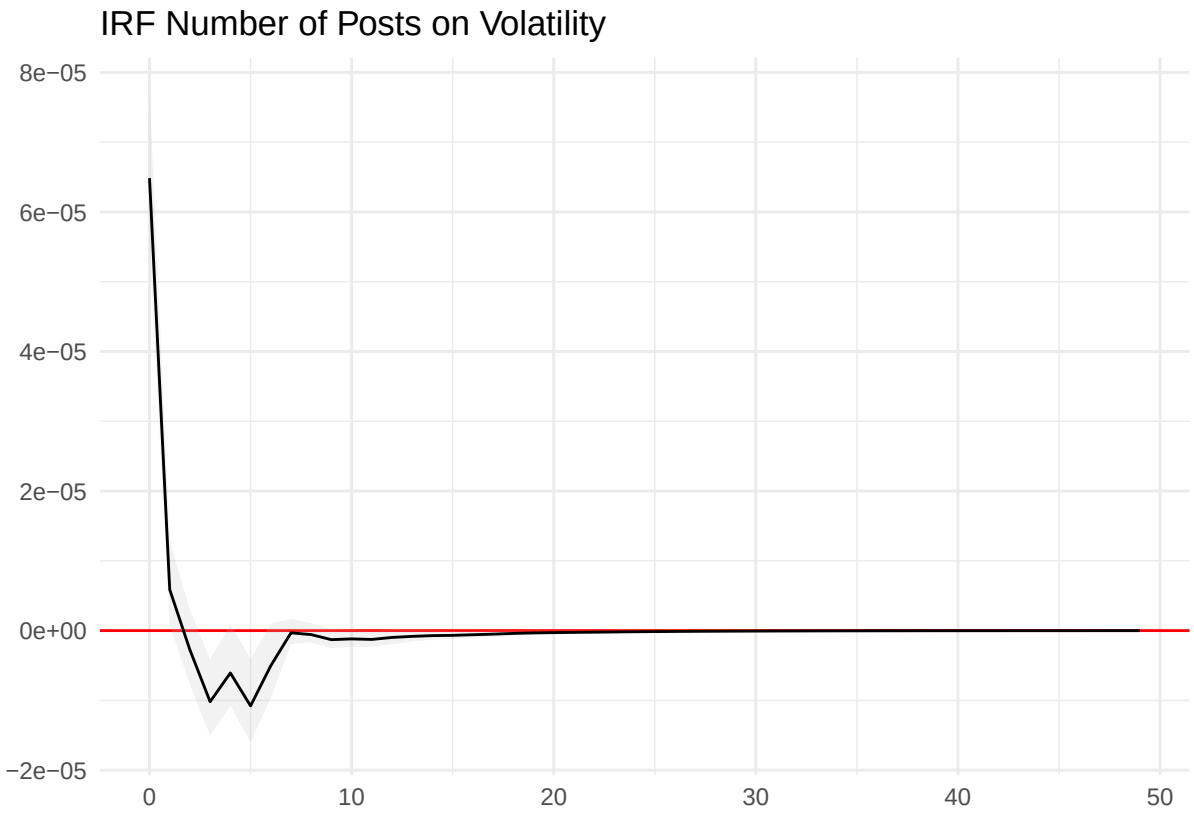
```
##
## ============================
##             Model 1
## ----------------------------
## N.l1          -0.000001 ***
##               (0.000000)
## vol.l1         0.282497 ***
##               (0.007132)
## N.l2          -0.000001 ***
##               (0.000000)
## vol.l2         0.072640 ***
##               (0.007403)
## N.l3          -0.000002 ***
##               (0.000000)
## vol.l3         0.047738 ***
##               (0.007409)
## N.l4          -0.000001 **
##               (0.000000)
## vol.l4         0.056237 ***
##               (0.007408)
## N.l5          -0.000002 ***
##               (0.000000)
## vol.l5         0.059528 ***
##               (0.007403)
## N.l6          -0.000001 ***
##               (0.000000)
## vol.l6         0.109380 ***
##               (0.007128)
## const          0.000081 ***
##               (0.000004)
## ----------------------------
## R^2            0.176168
## Adj. R^2       0.175673
## Num. obs.  19965
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
Omega2 <- var(residuals(est.VAR2))

#make the B matrix
loss2 <- function(param2){
  #Define the restriction
  B2 <- matrix(c(param2[1], param2[2], 0, param2[3]), ncol = 2)

  #Make BB' approximatively equal to omega
  X2 <- Omega2 - B2 %*% t(B2)
```

```
  #loss function
  loss2 <- sum(X2^2)
  return(loss2)
}

res.opt2 <- optim(c(1, 0, 1), loss2, method = "BFGS")
B.hat2 <- matrix(c(res.opt2$par[1], res.opt2$par[2], 0, res.opt2$par[3]), ncol = 2)

print(cbind(Omega2,B.hat2 %*% t(B.hat2)))
```

```
##                N          vol
## N   7.988455e+01 5.794262e-04 7.988455e+01 5.793217e-04
## vol 5.794262e-04 1.635220e-07 5.793217e-04 3.880898e-07
```

```
#irf creation
irf_res2 <- irf(est.VAR2, impulse = "N", response = "vol",
                  bmat=b.hat2, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf2 <- extract_varirf(irf_res2)

#the plot
single_varirf2 %>%
  ggplot(aes(x=period, y=irf_n_vol, ymin=lower_n_vol, ymax=upper_n_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF Number of Posts on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```




```
#does volatility Granger cause N mentions
grangertest(y2[,c("vol","N")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	3.17	0.00413

```
#does N mentions Granger cause volatility
grangertest(y2[,c("N", "vol")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	13.9	8.02e-16

2.3 Tariff

```r
y3 = cbind(Vdata$tariff , Vdata$ASHR_vol)
colnames(y3)[1:2] <- c("tariff", "vol")
est.VAR3 <- VAR(y3,p=6)
mod_vol3 <- est.VAR3$varresult$vol
screenreg(mod_vol3, digits = 6)
```

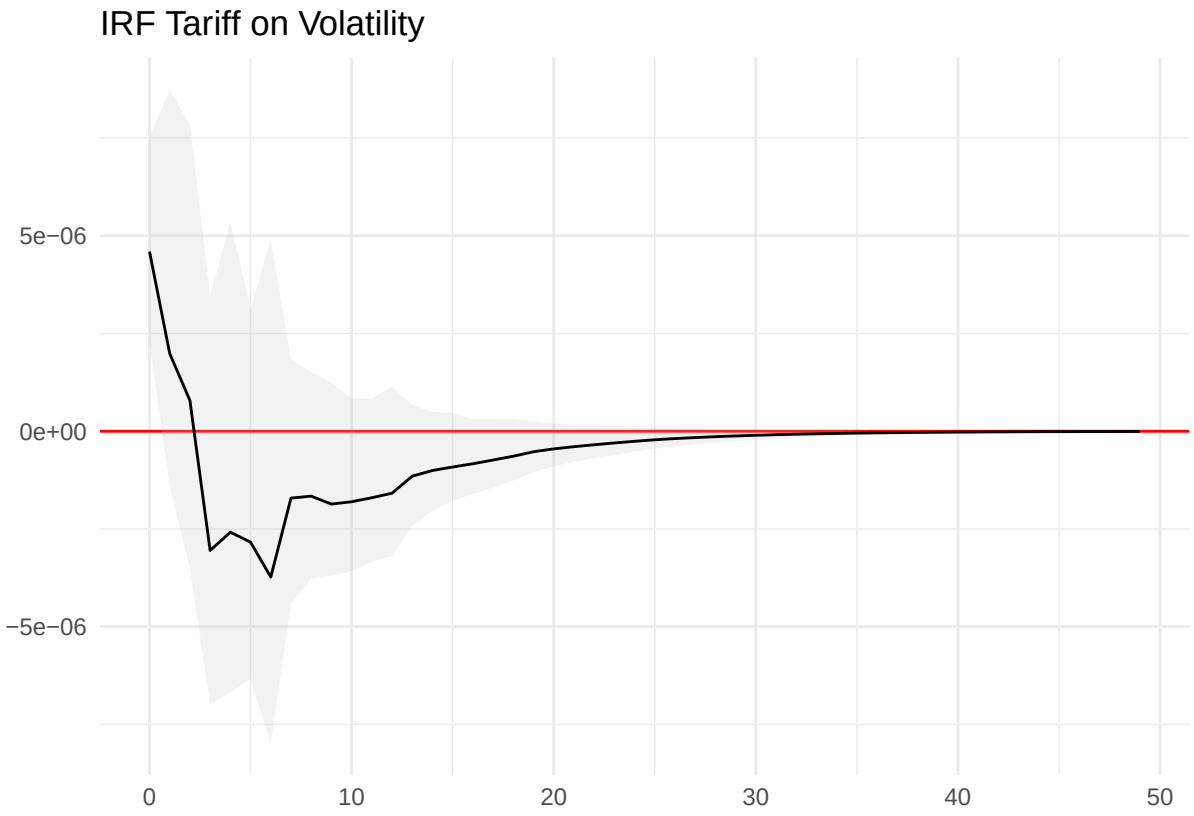
```
##
## ============================
##             Model 1
## ----------------------------
## tariff.l1      0.000002
##               (0.000008)
## vol.l1         0.280773 ***
##               (0.007038)
## tariff.l2     -0.000000
##               (0.000008)
## vol.l2         0.072461 ***
##               (0.007302)
## tariff.l3     -0.000010
##               (0.000008)
## vol.l3         0.045252 ***
##               (0.007309)
## tariff.l4     -0.000005
##               (0.000008)
## vol.l4         0.056243 ***
##               (0.007309)
## tariff.l5     -0.000005
##               (0.000008)
## vol.l5         0.056910 ***
##               (0.007302)
## tariff.l6     -0.000008
##               (0.000008)
## vol.l6         0.109191 ***
##               (0.007037)
## const          0.000061 ***
##               (0.000003)
## ----------------------------
## R^2            0.172906
## Adj. R^2       0.172409
## Num. obs.  19965
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
Omega3 <- var(residuals(est.VAR3))

#make the B matrix
loss3 <- function(param3){
  #Define the restriction
  B3 <- matrix(c(param3[1], param3[2], 0, param3[3]), ncol = 2)

  #Make BB' approximatively equal to omega
  X3 <- Omega3 - B3 %*% t(B3)
```

```
  #loss function
  loss3 <- sum(X3^2)
  return(loss3)
}

res.opt3 <- optim(c(1, 0, 1), loss3, method = "BFGS")
B.hat3 <- matrix(c(res.opt3$par[1], res.opt3$par[2], 0, res.opt3$par[3]), ncol = 2)

print(cbind(Omega3,B.hat3 %*% t(B.hat3)))
```

```
##              tariff          vol
## tariff 1.437847e-01 1.743059e-06 1.437843e-01 1.794967e-06
## vol    1.743059e-06 1.641694e-07 1.794967e-06 1.830018e-05
```

```
#irf creation
irf_res3 <- irf(est.VAR3, impulse = "tariff", response = "vol",
                  bmat=b.hat3, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf3 <- extract_varirf(irf_res3)

#the plot
single_varirf3 %>%
  ggplot(aes(x=period, y=irf_tariff_vol, ymin=lower_tariff_vol, ymax=upper_tariff_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF Tariff on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```




```
#does volatility Granger cause tariff mentions
grangertest(y3[,c("vol","tariff")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	0.722	0.632

```
#does tariff mentions Granger cause volatility
grangertest(y3[,c("tariff", "vol")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	0.715	0.638

2.4 Trade

```r
y4 = cbind(Vdata$trade , Vdata$ASHR_vol)
colnames(y4)[1:2] <- c("trade", "vol")
est.VAR4 <- VAR(y4,p=6)
mod_vol4 <- est.VAR4$varresult$vol
screenreg(mod_vol4, digits = 6)
```

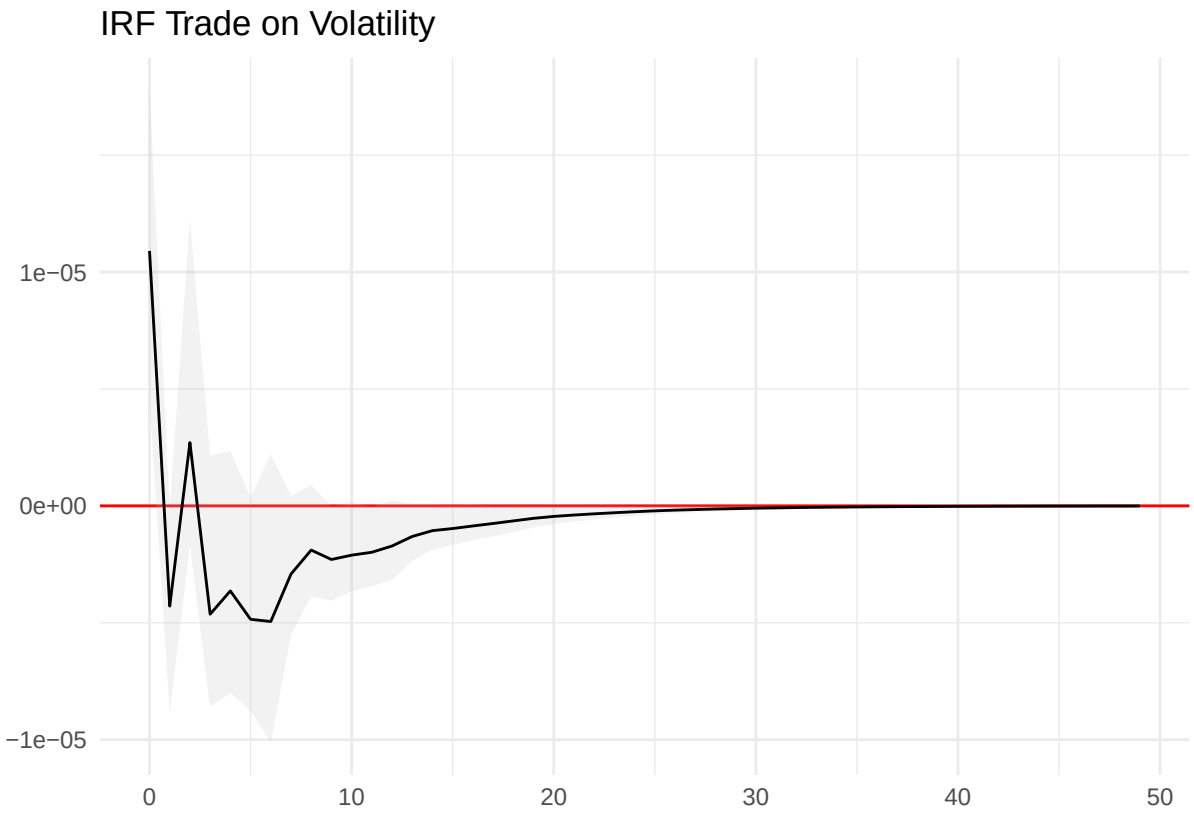
```
##
## ============================
##             Model 1
## ----------------------------
## trade.l1      -0.000025 *
##               (0.000010)
## vol.l1         0.281371 ***
##               (0.007039)
## trade.l2       0.000012
##               (0.000010)
## vol.l2         0.071544 ***
##               (0.007306)
## trade.l3      -0.000019
##               (0.000010)
## vol.l3         0.045746 ***
##               (0.007312)
## trade.l4      -0.000009
##               (0.000010)
## vol.l4         0.056738 ***
##               (0.007316)
## trade.l5      -0.000013
##               (0.000010)
## vol.l5         0.056226 ***
##               (0.007311)
## trade.l6      -0.000013
##               (0.000010)
## vol.l6         0.109845 ***
##               (0.007043)
## const          0.000062 ***
##               (0.000003)
## ----------------------------
## R^2            0.173422
## Adj. R^2       0.172925
## Num. obs.  19965
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
Omega4 <- var(residuals(est.VAR4))

#make the B matrix
loss4 <- function(param4){
  #Define the restriction
  B4 <- matrix(c(param4[1], param4[2], 0, param4[3]), ncol = 2)

  #Make BB' approximatively equal to omega
  X4 <- Omega4 - B4 %*% t(B4)
```

```
  #loss function
  loss4 <- sum(X4^2)
  return(loss4)
}

res.opt4 <- optim(c(1, 0, 1), loss4, method = "BFGS")
B.hat4 <- matrix(c(res.opt4$par[1], res.opt4$par[2], 0, res.opt4$par[3]), ncol = 2)

print(cbind(Omega4,B.hat4 %*% t(B.hat4)))
```

```
##              trade          vol
## trade 8.360782e-02 3.151165e-06 8.360540e-02 6.905593e-07
## vol   3.151165e-06 1.640670e-07 6.905593e-07 4.318409e-05
```

```
#irf creation
irf_res4 <- irf(est.VAR4, impulse = "trade", response = "vol",
                  bmat=b.hat4, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf4 <- extract_varirf(irf_res4)

#the plot
single_varirf4 %>%
  ggplot(aes(x=period, y=irf_trade_vol, ymin=lower_trade_vol, ymax=upper_trade_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF Trade on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```




```
#does volatility Granger cause trade mentions
grangertest(y4[,c("vol","trade")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	5.36	1.55e-05

```
#does trade mentions Granger cause volatility
grangertest(y4[,c("trade", "vol")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	2.79	0.0103

3 China

```r
y5 = cbind(Vdata$china , Vdata$ASHR_vol)
colnames(y5)[1:2] <- c("china", "vol")
est.VAR5 <- VAR(y5,p=6)
mod_vol5 <- est.VAR5$varresult$vol
screenreg(mod_vol5, digits = 6)
```

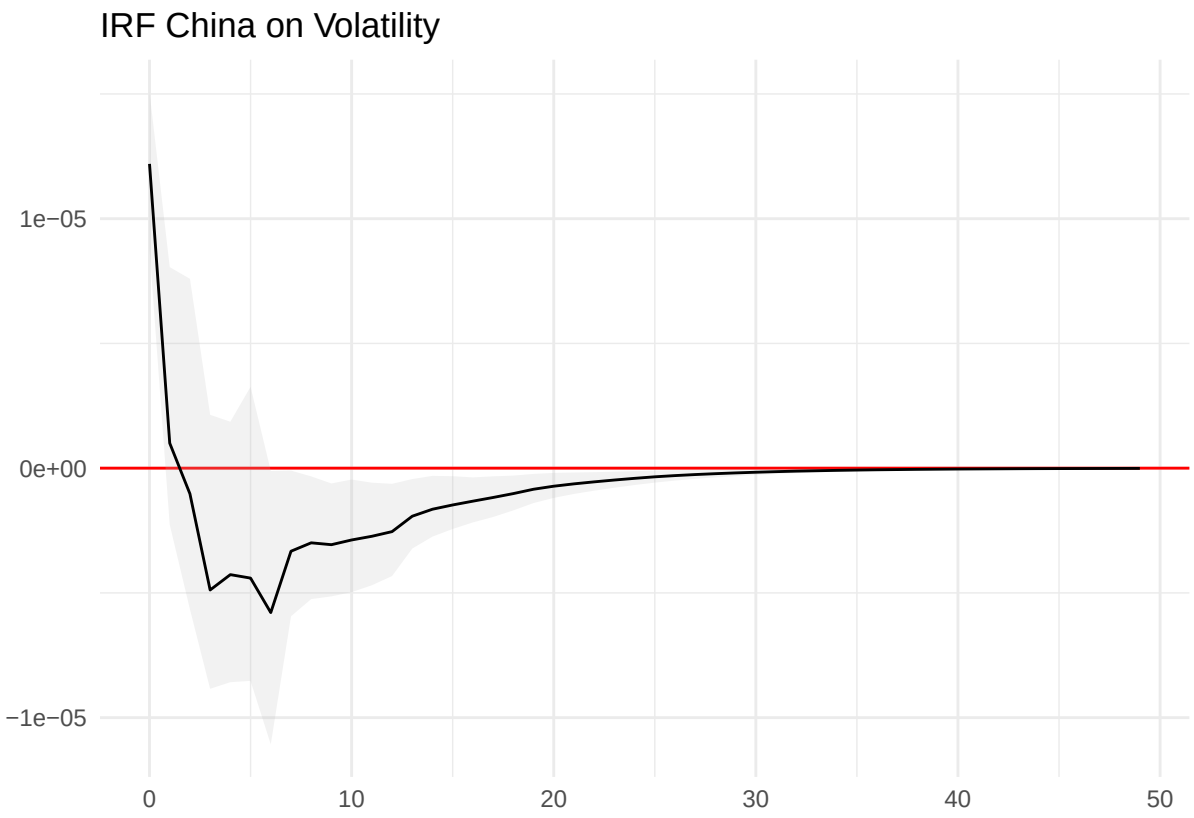
```
##
## ============================
##             Model 1
## ----------------------------
## china.l1      -0.000005
##               (0.000006)
## vol.l1         0.280637 ***
##               (0.007040)
## china.l2      -0.000004
##               (0.000006)
## vol.l2         0.072261 ***
##               (0.007305)
## china.l3      -0.000011
##               (0.000006)
## vol.l3         0.045298 ***
##               (0.007312)
## china.l4      -0.000007
##               (0.000006)
## vol.l4         0.056264 ***
##               (0.007312)
## china.l5      -0.000007
##               (0.000006)
## vol.l5         0.056857 ***
##               (0.007305)
## china.l6      -0.000010
##               (0.000006)
## vol.l6         0.109272 ***
##               (0.007040)
## const          0.000063 ***
##               (0.000003)
## ----------------------------
## R^2            0.173206
## Adj. R^2       0.172709
## Num. obs.  19965
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
Omega5 <- var(residuals(est.VAR5))

#make the B matrix
loss5 <- function(param5){
  #Define the restriction
  B5 <- matrix(c(param5[1], param5[2], 0, param5[3]), ncol = 2)

  #Make BB' approximatively equal to omega
  X5 <- Omega5 - B5 %*% t(B5)
```

```
  #loss function
  loss5 <- sum(X5^2)
  return(loss5)
}

res.opt5 <- optim(c(1, 0, 1), loss5, method = "BFGS")
B.hat5 <- matrix(c(res.opt5$par[1], res.opt5$par[2], 0, res.opt5$par[3]), ncol = 2)

print(cbind(Omega5,B.hat5 %*% t(B.hat5)))
```

```
##              china          vol
## china 2.022865e-01 5.483724e-06 2.022859e-01 5.147609e-06
## vol   5.483724e-06 1.641099e-07 5.147609e-06 8.314668e-06
```

```
#irf creation
irf_res5 <- irf(est.VAR5, impulse = "china", response = "vol",
                  bmat=b.hat5, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf5 <- extract_varirf(irf_res5)

#the plot
single_varirf5 %>%
  ggplot(aes(x=period, y=irf_china_vol, ymin=lower_china_vol, ymax=upper_china_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF China on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```




```
#does volatility Granger cause china mentions
grangertest(y5[,c("vol","china")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	1.08	0.372

```
#does china mentions Granger cause volatility
grangertest(y5[,c("china", "vol")], order = 6)
```

Res.Df	Df	F	Pr(>F)
2e+04			
2e+04	-6	1.92	0.0733

4 Interaction number of post and tariff

here is an example of our interaction

```r
#interaction
##N and tariff, 2 variables

int1 = Vdata$tariff * Vdata$N

y12 = cbind(int1, Vdata$ASHR_vol)
colnames(y12)[1:2] <- c("interaction", "vol")
est.VAR12 <- VAR(y12,p=6)
mod_vol12 <- est.VAR12$varresult$vol
screenreg(mod_vol12, digits = 6)
```

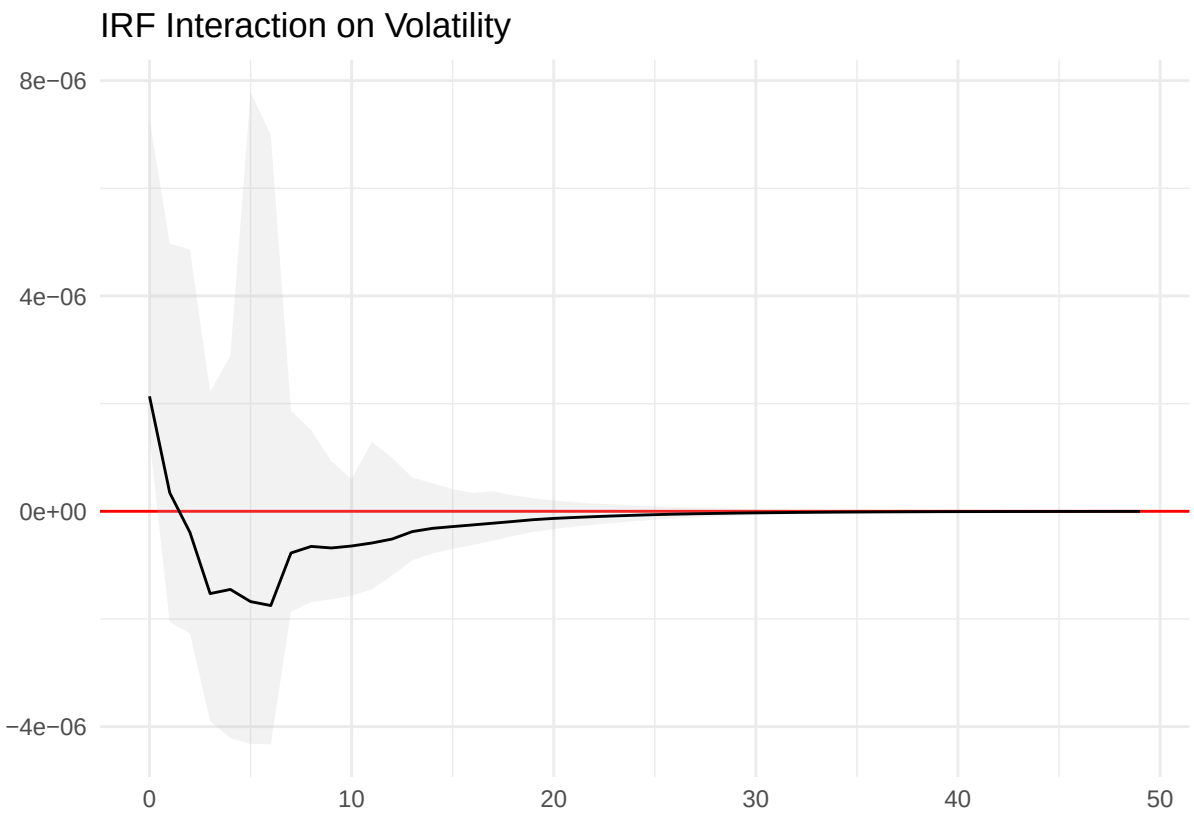
```
##
## ================================
##                  Model 1
## --------------------------------
## interaction.l1     -0.000000
##                    (0.000000)
## vol.l1              0.280904 ***
##                    (0.007037)
## interaction.l2     -0.000000
##                    (0.000000)
## vol.l2              0.072496 ***
##                    (0.007302)
## interaction.l3     -0.000000
##                    (0.000000)
## vol.l3              0.045218 ***
##                    (0.007309)
## interaction.l4     -0.000000
##                    (0.000000)
## vol.l4              0.056276 ***
##                    (0.007309)
## interaction.l5     -0.000000
##                    (0.000000)
## vol.l5              0.056856 ***
##                    (0.007302)
## interaction.l6     -0.000000
##                    (0.000000)
## vol.l6              0.109078 ***
##                    (0.007037)
## const               0.000060 ***
##                    (0.000003)
## --------------------------------
## R^2                 0.172766
## Adj. R^2            0.172269
## Num. obs.       19965
## ================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
Omega12 <- var(residuals(est.VAR12))
#make the B matrix
loss12 <- function(param12){
  #Define the restriction
  B12 <- matrix(c(param12[1], param12[2], 0, param12[3]), ncol = 2)
```

```r
  #Make BB' approximatively equal to omega
  X12 <- Omega12 - B12 %*% t(B12)

  #loss function
  loss12 <- sum(X12^2)
  return(loss12)
}

res.opt12 <- optim(c(1, 0, 1), loss12, method = "BFGS")
B.hat12 <- matrix(c(res.opt12$par[1], res.opt12$par[2], 0, res.opt12$par[3]), ncol = 2)

print(cbind(Omega12,B.hat12 %*% t(B.hat12)))
```

```
##              interaction          vol
## interaction 7.087119e+02 5.683122e-05 7.087119e+02 5.688691e-05
## vol         5.683122e-05 1.641972e-07 5.688691e-05 7.197072e-06
```

```r
#irf creation
irf_res12 <- irf(est.VAR12, impulse = "interaction", response = "vol",
                 bmat=b.hat12, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf12 <- extract_varirf(irf_res12)

#the plot
single_varirf12 %>%
  ggplot(aes(x=period, y=irf_interaction_vol, ymin=lower_interaction_vol, ymax=upper_interaction_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF Interaction on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```



5 Terms

Here we look for the first and second mandate effect of posts. We will use the tariff variable as a proxy for the posts

5.1 First mandate

```
# First and Second Mandate

#first term
Vdata_f = filter(data,between(timestamp, as.Date('2017-01-20'), as.Date('2021-01-20')))

#second term
Vdata_s = filter(data,between(timestamp, as.Date('2025-01-20'), as.Date('2025-05-07')))
```

```
y_f_d = cbind(Vdata_f$tariff, Vdata_f$ASHR_vol)
colnames(y_f_d)[1:2] <- c("tariff", "vol")
est.VAR_f_d <- VAR(y_f_d,p=6)
mod_vol_f_d <- est.VAR_f_d$varresult$vol
screenreg(mod_vol_f_d, digits = 6)
```

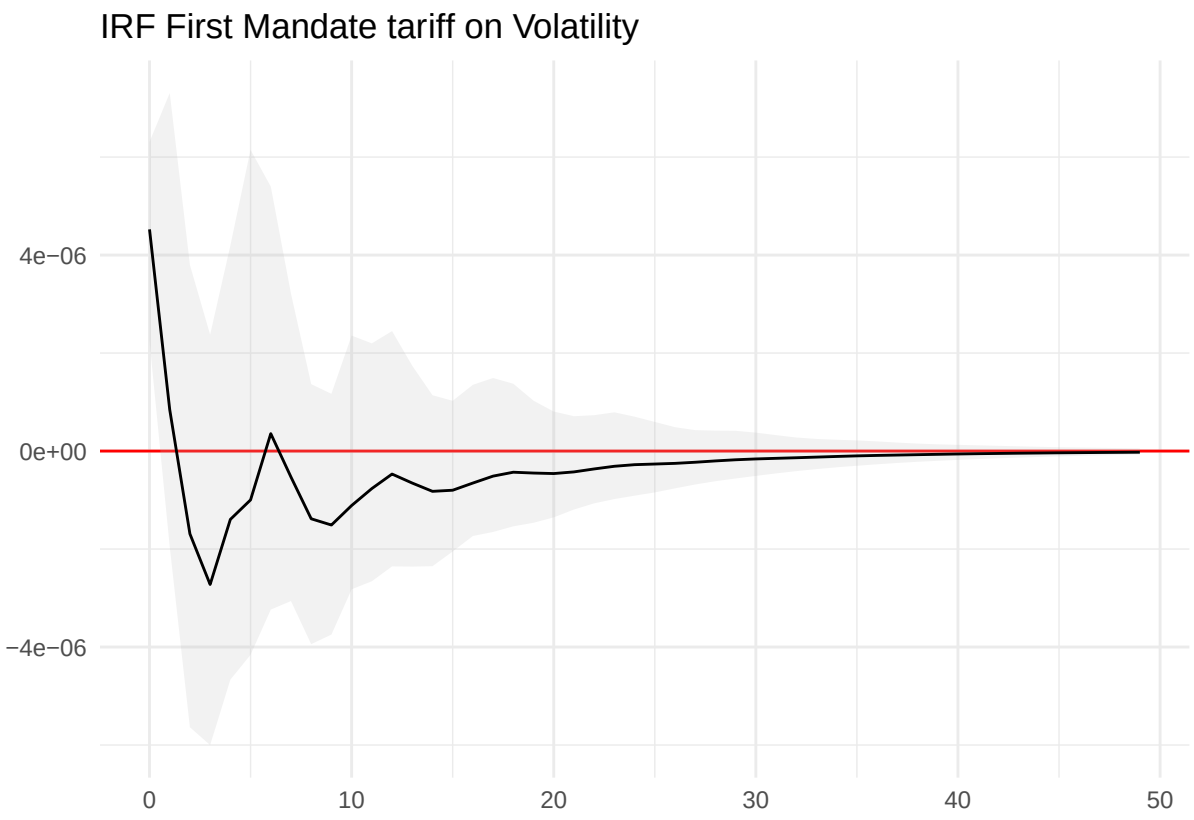
```
##
```

```
## ==========================
##            Model 1
## --------------------------
## tariff.l1    -0.000001
##              (0.000008)
## vol.l1        0.245435 ***
##              (0.011754)
## tariff.l2    -0.000007
##              (0.000008)
## vol.l2        0.075223 ***
##              (0.012070)
## tariff.l3    -0.000007
##              (0.000008)
## vol.l3        0.060102 ***
##              (0.012082)
## tariff.l4    -0.000002
##              (0.000008)
## vol.l4        0.060634 ***
##              (0.012082)
## tariff.l5    -0.000002
##              (0.000008)
## vol.l5        0.085747 ***
##              (0.012070)
## tariff.l6     0.000001
##              (0.000008)
## vol.l6        0.173709 ***
##              (0.011754)
## const         0.000033 ***
##              (0.000003)
## --------------------------
## R^2           0.218236
## Adj. R^2      0.216900
## Num. obs.  7036
## ==========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
Omega_f_d <- var(residuals(est.VAR_f_d))
#make the B matrix
loss_f_d <- function(param_f_d){
  #Define the restriction
  B_f_d <- matrix(c(param_f_d[1], param_f_d[2], 0, param_f_d[3]), ncol = 2)

  #Make BB' approximatively equal to omega
  X_f_d <- Omega_f_d - B_f_d %*% t(B_f_d)

  #loss function
  loss_f_d <- sum(X_f_d^2)
  return(loss_f_d)
}

res.opt_f_d <- optim(c(1, 0, 1), loss_f_d, method = "BFGS")
B.hat_f_d <- matrix(c(res.opt_f_d$par[1], res.opt_f_d$par[2], 0, res.opt_f_d$par[3]), ncol = 2)

print(cbind(Omega_f_d,B.hat_f_d %*% t(B.hat_f_d)))
```

```
##              tariff          vol
## tariff 9.834364e-02 1.417639e-06  9.834195e-02 -3.060188e-07
## vol    1.417639e-06 4.865146e-08 -3.060188e-07  3.476919e-05
```

```
#irf creation
irf_res_f_d <- irf(est.VAR_f_d, impulse = "tariff", response = "vol",
                   bmat=b.hat_f_d, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf_f_d <- extract_varirf(irf_res_f_d)

#the plot
single_varirf_f_d %>%
  ggplot(aes(x=period, y=irf_tariff_vol, ymin=lower_tariff_vol, ymax=upper_tariff_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF First Mandate tariff on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```




```
#does vol granger cause tariff
grangertest(y_f_d[,c("vol","tariff")], order = 6)
```

Res.Df	Df	F	Pr(>F)
7.02e+03			
7.03e+03	-6	0.384	0.889

```
#does tariff granger cause vol
grangertest(y_f_d[,c("tariff", "vol")], order = 6)
```

Res.Df	Df	F	Pr(>F)
7.02e+03			
7.03e+03	-6	0.346	0.912

5.2 second mandate

```
y_s_d = cbind(Vdata_s$tariff, Vdata_s$ASHR_vol)
colnames(y_s_d)[1:2] <- c("tariff", "vol")
est.VAR_s_d <- VAR(y_s_d,p=6)
mod_vol_s_d <- est.VAR_s_d$varresult$vol
screenreg(mod_vol_s_d, digits = 6)
```

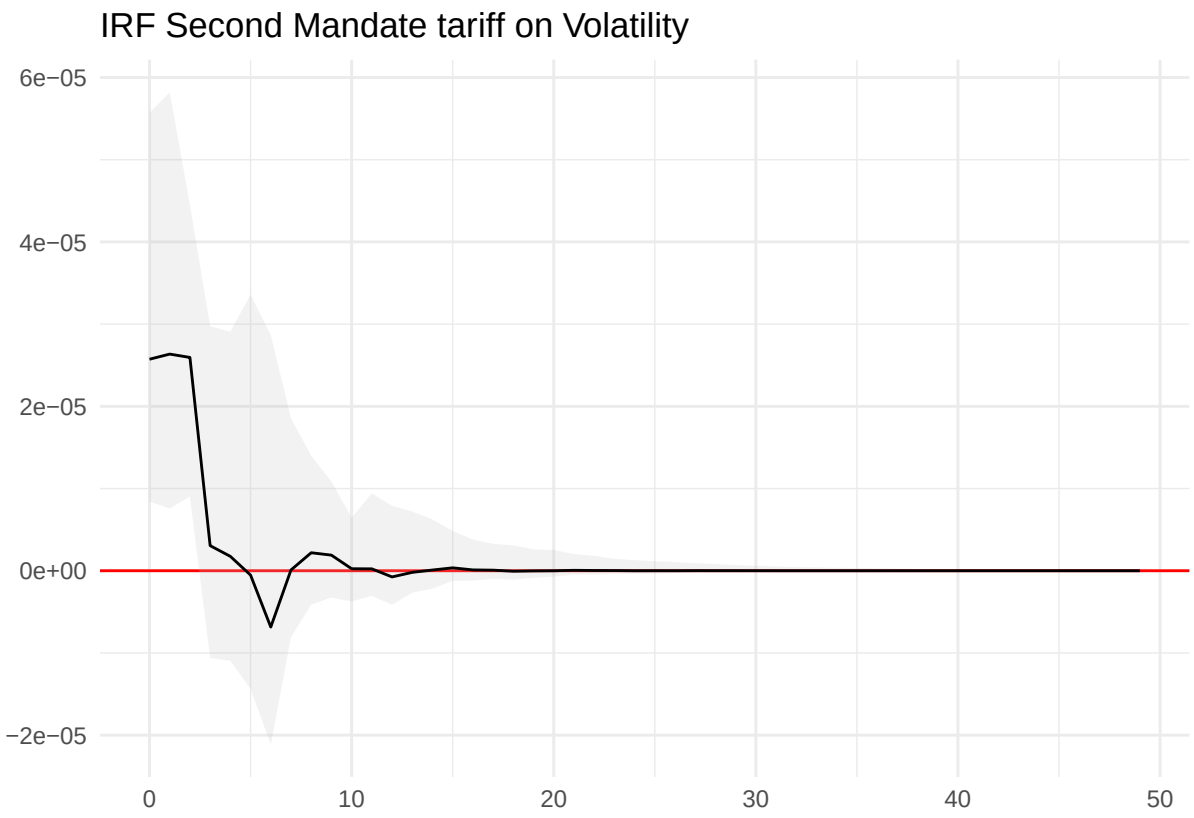
```
##
## ========================
##            Model 1
## ------------------------
## tariff.l1    0.000008
##             (0.000005)
## vol.l1       0.443019 ***
##             (0.044731)
## tariff.l2    0.000006
##             (0.000005)
## vol.l2       0.075886
##             (0.049172)
## tariff.l3   -0.000005
##             (0.000005)
## vol.l3      -0.062004
##             (0.049326)
## tariff.l4   -0.000001
##             (0.000005)
## vol.l4       0.075354
##             (0.049365)
## tariff.l5   -0.000001
##             (0.000005)
## vol.l5      -0.003760
##             (0.049339)
## tariff.l6   -0.000006
```

```
##             (0.000005)
## vol.l6       0.123405 **
##             (0.044686)
## const        0.000033 **
##             (0.000011)
## -------------------------
## R^2          0.287673
## Adj. R^2     0.270543
## Num. obs.  512
## =========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
Omega_s_d <- var(residuals(est.VAR_s_d))
#make the B matrix
loss_s_d <- function(param_s_d){
  #Define the restriction
  B_s_d <- matrix(c(param_s_d[1], param_s_d[2], 0, param_s_d[3]), ncol = 2)

  #Make BB' approximatively equal to omega
  X_s_d <- Omega_s_d - B_s_d %*% t(B_s_d)

  #loss function
  loss_s_d <- sum(X_s_d^2)
  return(loss_s_d)
}

res.opt_s_d <- optim(c(1, 0, 1), loss_s_d, method = "BFGS")
B.hat_s_d <- matrix(c(res.opt_s_d$par[1], res.opt_s_d$par[2], 0, res.opt_s_d$par[3]), ncol = 2)

print(cbind(Omega_s_d,B.hat_s_d %*% t(B.hat_s_d)))
```

```
##              tariff          vol
## tariff 3.774830e+00 4.940214e-05 3.77483e+00 4.944550e-05
## vol    4.940214e-05 4.060693e-08 4.94455e-05 2.889062e-05
```

```
#irf creation
irf_res_s_d <- irf(est.VAR_s_d, impulse = "tariff", response = "vol",
                   bmat=b.hat_s_d, n.ahead = 7 * 7, boot = TRUE, ci = 0.95)

#function to extract relevant objects for plotting
single_varirf_s_d <- extract_varirf(irf_res_s_d)

#the plot
single_varirf_s_d %>%
  ggplot(aes(x=period, y=irf_tariff_vol, ymin=lower_tariff_vol, ymax=upper_tariff_vol)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=0.2) +
  geom_line() +
  theme_light() +
  ggtitle("IRF Second Mandate tariff on Volatility")+
  ylab("")+
  xlab("") +
  theme_minimal()
```




```
#does vol granger cause tariff
grangertest(y_s_d[,c("vol","tariff")], order = 6)
```

Res.Df	Df	F	Pr(>F)
499			
505	-6	1.49	0.179

```
#does tariff granger cause vol
grangertest(y_s_d[,c("tariff", "vol")], order = 6)
```

Res.Df	Df	F	Pr(>F)
499			
505	-6	1.18	0.313