SVAR Example Using R

Emmanuel Davila

7/4/2022

This is a simple example of how to make a Structural Vector Autoregression in R. I used this video by Justin Eloriagaas as a guide to make this.

Loading Packages and the Data Set

First we will load the relevant packages to perform our analysis.

```
library(urca)
library(vars)
library(mFilter)
library(tseries)
library(TSstudio)
library(forecast)
library(tidyverse)
```

Next we will load our data set. All data was obtained from the **Federal Reserve Economic Database (FRED)** website. All data is quarterly from Q1 1970 to Q1 2022. I will provide the .csv file. The data includes the US output gap, US CPI, and the Federal Funds Rate. You will need to replace *~/Documents/Rprojects/svardataex.csv* with your own file path.

```
macro <- read.csv("~/Documents/Rprojects/svardataex.csv")
head(macro)
```

```
##         DATE output_gap core_cpi fed_funds_rate
## 1 1970-01-01 -0.1084213 6.916757       8.573333
## 2 1970-04-01 -0.7935408 7.010863       7.886667
## 3 1970-07-01 -0.6734349 7.107600       6.706667
```

```
## 4 1970-10-01 -2.4913922 7.284990       5.566667
## 5 1971-01-01 -0.6061156 6.312204       3.856667
## 6 1971-04-01 -0.8391273 5.235265       4.566667
```

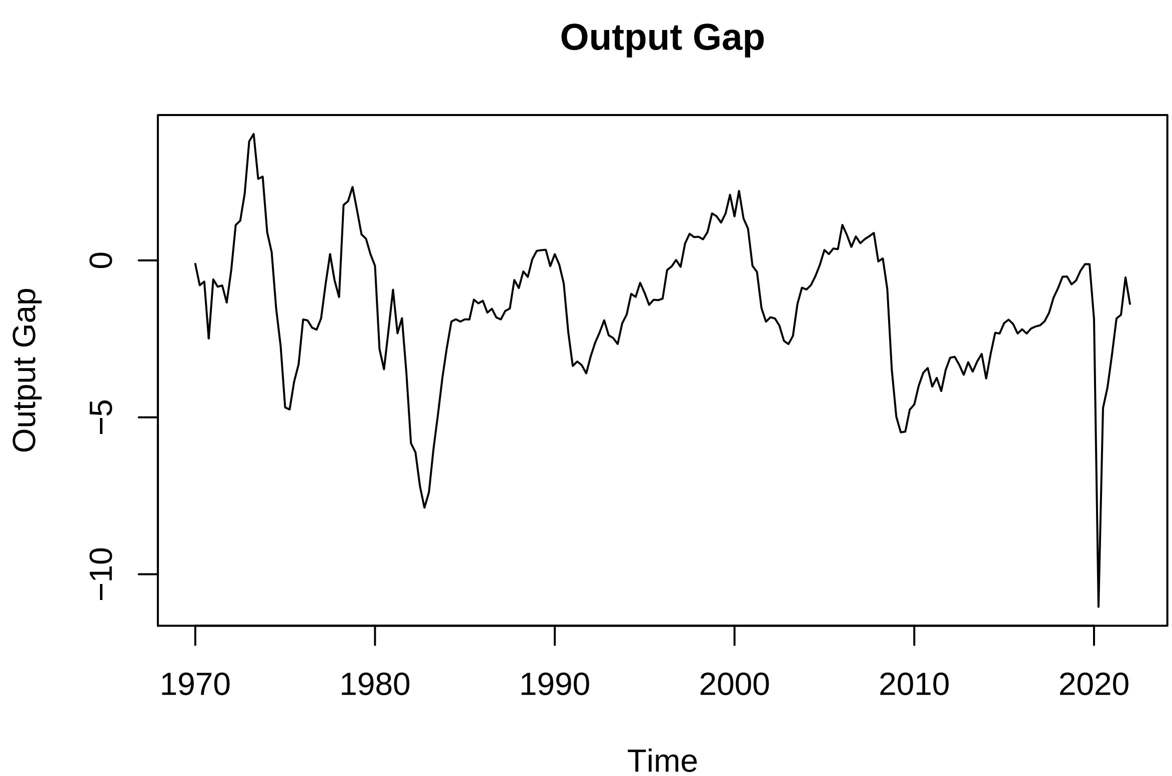
Creating Time Series Objects and Plotting Them

Now that the data is loaded onto R, we will transform the data into Time Series Objects using ts(). We use *start = c(1970,1,1), frequency = 4* to signify that our data starts on 1970 January 1, and the frequency is quarterly.

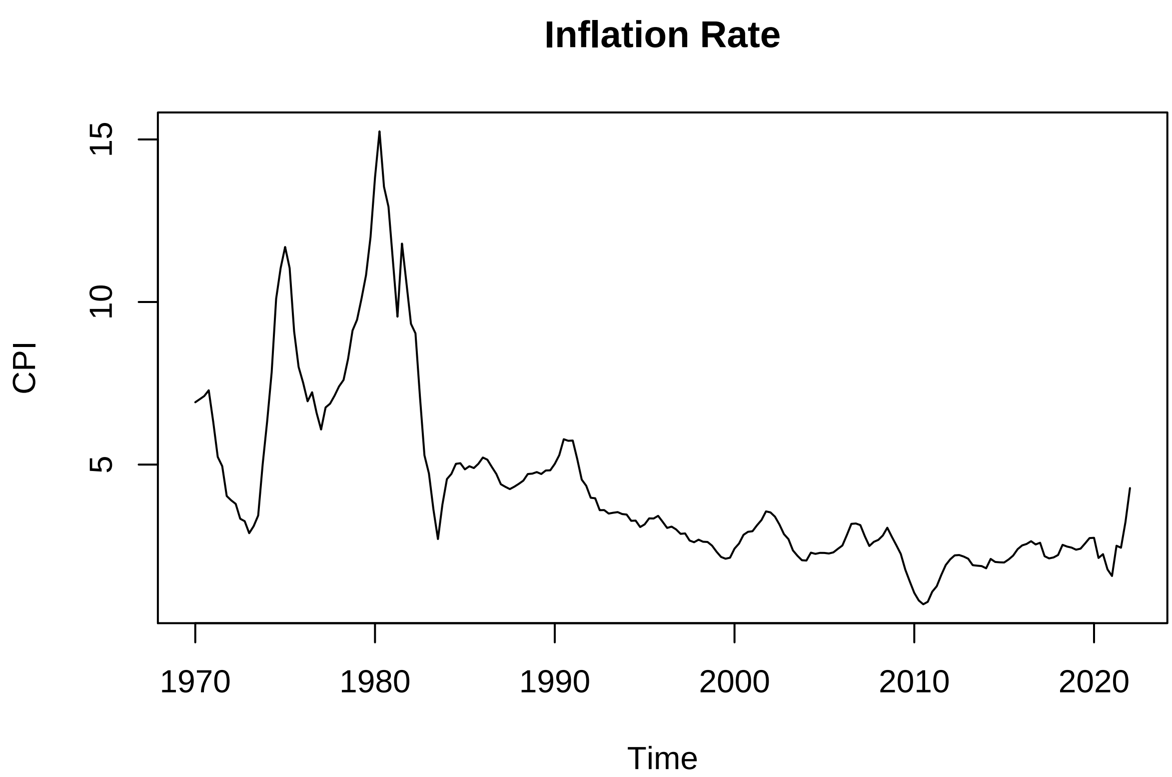
```
y <- ts(macro$output_gap, start = c(1970,1,1), frequency = 4)
pi <- ts(macro$core_cpi, start = c(1970,1,1), frequency = 4)
r <- ts(macro$fed_funds_rate, start = c(1970,1,1), frequency = 4)
```

Lets plot the three series' to make sure they look like they're supposed to.

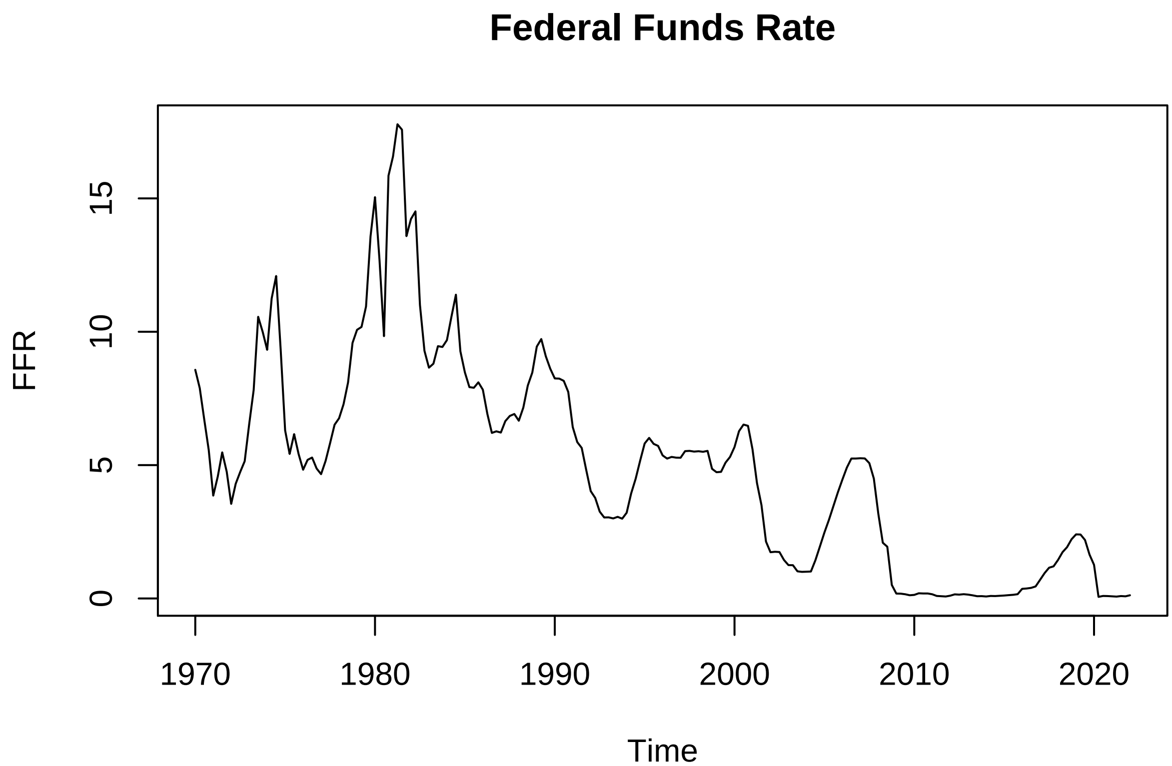
```
plot(y, main = "Output Gap", xlab = "Time", ylab = "Output Gap")
```



```r
plot(pi, main = "Inflation Rate", xlab = "Time", ylab = "CPI")
```




```r
plot(r, main = "Federal Funds Rate", xlab = "Time", ylab = "FFR")
```



Applying Restrictions to the SVAR

One of the most important parts in creating our SVAR is imposing the restrictions. This is done by creating a matrix and ordering it according to economic principles and intuition. The matrix represents contemporaneous shocks affecting the variables in the systems.

$$\begin{bmatrix} y & 0 & 0 \\ a_{21} & pi & 0 \\ a_{31} & a_{32} & ffr \end{bmatrix}$$

The figure above is what our matrix will look like. The output gap (first column) can influence both inflation (second column) and the ffr (third column) in the same period. Inflation influences the ffr but not the output gap in the same period (hence the free variable to the left of pi). The ffr does not influence either in the same period (hence the two free variables to the left of ffr). Therefore, the zeros in the upper triangular part of the matrix are our restrictions.

Now we can start creating our matrix. First, we create an **Identity** matrix.

```r
amat <- diag(3)
amat
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

Then, we assign the free variables to the lower triangular part.

```r
amat[2,1] <- NA
amat[3,1] <- NA
amat[3,2] <- NA
amat
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]   NA    1    0
## [3,]   NA   NA    1
```

Once we estimate our SVAR, the free variables will fill up and replace the *NA*s.

Building the Model

It's finally time to build our model. We will follow the economic intuition we discussed previously to bind our variables.

```r
svar1 <- cbind(y, pi, r)
colnames(svar1) <- c('OutputGap', 'CPI', 'FFR')
head(svar1)
```

```
##          OutputGap      CPI      FFR
## 1970 Q1 -0.1084213 6.916757 8.573333
## 1970 Q2 -0.7935408 7.010863 7.886667
## 1970 Q3 -0.6734349 7.107600 6.706667
## 1970 Q4 -2.4913922 7.284990 5.566667
## 1971 Q1 -0.6061156 6.312204 3.856667
## 1971 Q2 -0.8391273 5.235265 4.566667
```

We also have to select the number of lags that we will use in the model. For this, we will use the *VARselect()* function. Since this is quarterly data, we expect a lag order of around four to six. So we will specify a max of eight lags, just to be safe.

```
lagselect <- VARselect(svar1, lag.max = 8, type = "both")
lagselect$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      5      5      5
```

So according to AIC, SBIC, HQIC, and FPE selection criteria, the optimal number of lags for the model is 5.

Estimating the Model

Now that the framework for our model is built, we can estimate it. First, we'll need to estimate a regular VAR so we can later impose our restrictions.

```
varmodel1 <- VAR(svar1, p = 5, season = NULL, exog = NULL, type = "const")
varmodel1
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation OutputGap:
## ==============================================
## Call:
## OutputGap = OutputGap.l1 + CPI.l1 + FFR.l1 + OutputGap.l2 + CPI.l2 + FFR.l2 + OutputGap.l3 + CPI.l3 +
##
## OutputGap.l1       CPI.l1       FFR.l1 OutputGap.l2       CPI.l2       FFR.l2
##  0.842956960 -0.439279337  0.173594464  0.097186443  0.202050873 -0.265117708
## OutputGap.l3       CPI.l3       FFR.l3 OutputGap.l4       CPI.l4       FFR.l4
## -0.002834393  0.150369652  0.244324916 -0.027695943  0.292564090 -0.115241846
## OutputGap.l5       CPI.l5       FFR.l5        const
## -0.050803371 -0.246264524 -0.051731186  0.036463149
##
##
```

```
## Estimated coefficients for equation CPI:
## ========================================
## Call:
## CPI = OutputGap.l1 + CPI.l1 + FFR.l1 + OutputGap.l2 + CPI.l2 + FFR.l2 + OutputGap.l3 + CPI.l3 + FFR.l
##
## OutputGap.l1       CPI.l1       FFR.l1 OutputGap.l2       CPI.l2       FFR.l2
##  -0.02007062   1.33795307   0.21375829   0.05837871  -0.24977274  -0.22559912
## OutputGap.l3       CPI.l3       FFR.l3 OutputGap.l4       CPI.l4       FFR.l4
##  -0.01203928   0.06263685   0.22608996  -0.06688984  -0.46508368  -0.30651109
## OutputGap.l5       CPI.l5       FFR.l5        const
##   0.06390800   0.27257443   0.10393326   0.15725146
##
##
## Estimated coefficients for equation FFR:
## ========================================
## Call:
## FFR = OutputGap.l1 + CPI.l1 + FFR.l1 + OutputGap.l2 + CPI.l2 + FFR.l2 + OutputGap.l3 + CPI.l3 + FFR.l
##
## OutputGap.l1       CPI.l1       FFR.l1 OutputGap.l2       CPI.l2       FFR.l2
##  0.209560911 -0.888582863  1.286835586 -0.093964499  1.578368132 -0.493494214
## OutputGap.l3       CPI.l3       FFR.l3 OutputGap.l4       CPI.l4       FFR.l4
##  0.035193389 -0.659733356  0.212214032 -0.073074344 -0.105423940  0.157574952
## OutputGap.l5       CPI.l5       FFR.l5        const
## -0.002292005  0.152373213 -0.239273255  0.150436608
```

Next, we will estimate the actual SVAR model. In the *Amat=* option, we will input the matrix we created earlier for our restrictions (amat).

```
svarmodel1 <- SVAR(varmodel1, Amat = amat, Bmat = NULL, hessian = TRUE,
              estmethod = c("scoring", "direct"))
svarmodel1
```

```
##
## SVAR Estimation Results:
## ========================
```

```
## 
## 
## Estimated A matrix:
##           OutputGap     CPI FFR
## OutputGap   1.00000  0.0000   0
## CPI        -0.05298  1.0000   0
## FFR        -0.22168 -0.3883   1
```

Impulse Response Functions

Telling the Story

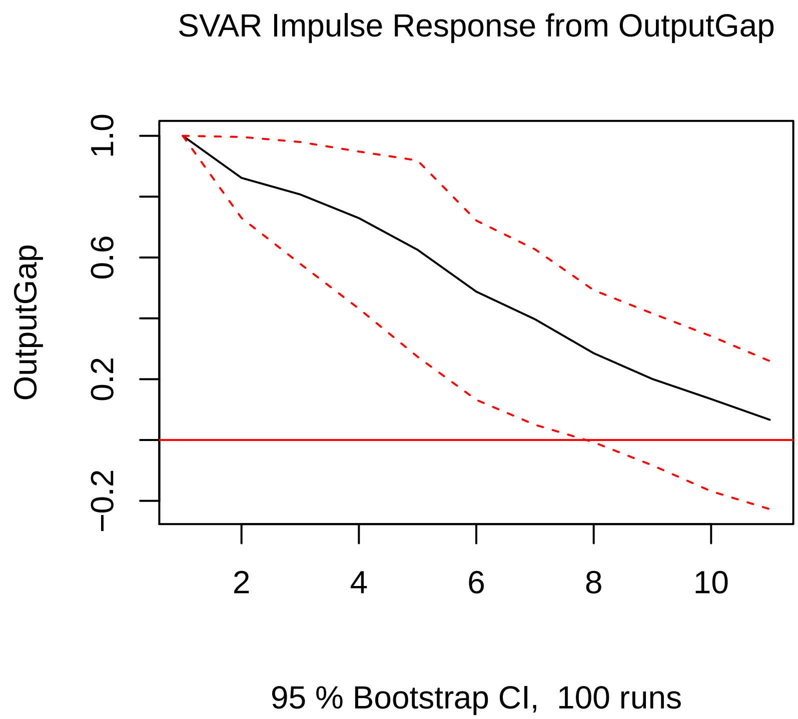
The final step is to plot our impulse response functions (irf). First, we will see what a positive exogenous shock does to the output gap.

```r
svarog <- irf(svarmodel1, impulse = "OutputGap", response = "OutputGap")
svarog
```

```
## 
## Impulse response coefficients
## $OutputGap
##        OutputGap
##  [1,] 1.00000000
##  [2,] 0.86173770
##  [3,] 0.80732473
##  [4,] 0.72946798
##  [5,] 0.62525715
##  [6,] 0.48772686
##  [7,] 0.39692689
##  [8,] 0.28532192
##  [9,] 0.20058702
## [10,] 0.13461002
## [11,] 0.06647378
## 
## 
```

```
## Lower Band, CI= 0.95
## $OutputGap
##          OutputGap
##  [1,]  1.000000000
##  [2,]  0.730051102
##  [3,]  0.579280442
##  [4,]  0.431813332
##  [5,]  0.273728330
##  [6,]  0.132116548
##  [7,]  0.050083523
##  [8,] -0.007366506
##  [9,] -0.083227537
## [10,] -0.168311912
## [11,] -0.227318489
##
##
## Upper Band, CI= 0.95
## $OutputGap
##       OutputGap
##  [1,] 1.0000000
##  [2,] 0.9963486
##  [3,] 0.9796882
##  [4,] 0.9484044
##  [5,] 0.9189320
##  [6,] 0.7215943
##  [7,] 0.6269484
##  [8,] 0.4926216
##  [9,] 0.4160313
## [10,] 0.3415395
## [11,] 0.2594622
```

```
plot(svarog)
```



As expected, a positive shock to the output gap causes the gap to increase initially. But almost immediately it starts to slowly decrease.

Now let's see how inflation (CPI) responds to a positive shock caused by the output gap.

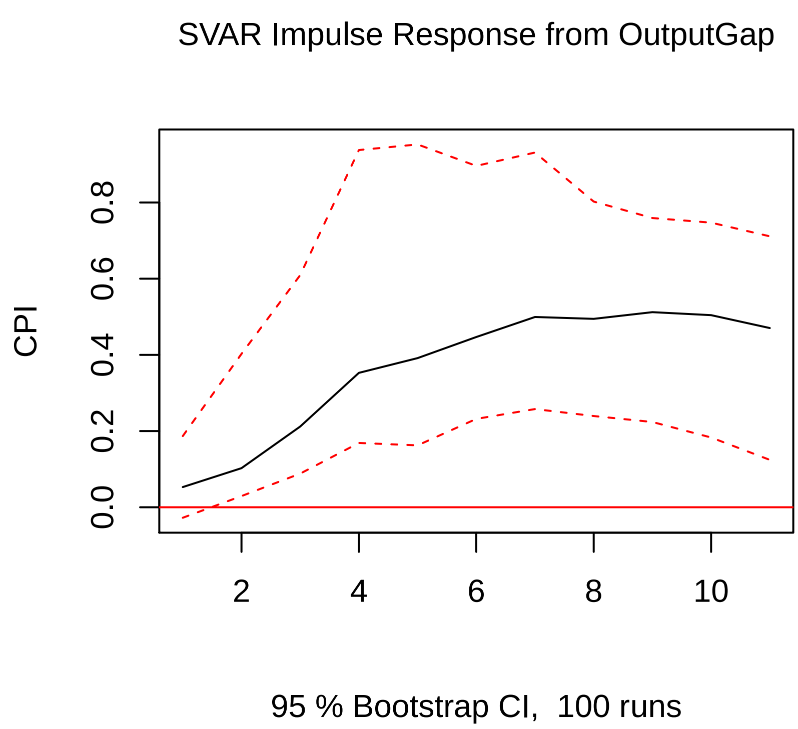
```
svarinf <- irf(svarmodel1, impulse = "OutputGap", response = "CPI")
svarinf
```

```
##
## Impulse response coefficients
## $OutputGap
##              CPI
##  [1,] 0.05298243
##  [2,] 0.10260226
##  [3,] 0.21184353
##  [4,] 0.35277403
##  [5,] 0.39149439
##  [6,] 0.44691534
```

```
##  [7,] 0.49942720
##  [8,] 0.49458326
##  [9,] 0.51210215
## [10,] 0.50428497
## [11,] 0.47038999
##
##
## Lower Band, CI= 0.95
## $OutputGap
##               CPI
##  [1,] -0.02761610
##  [2,]  0.02928873
##  [3,]  0.08832140
##  [4,]  0.16875705
##  [5,]  0.16247343
##  [6,]  0.23198781
##  [7,]  0.25769751
##  [8,]  0.23945914
##  [9,]  0.22378355
## [10,]  0.18307648
## [11,]  0.12433903
##
##
## Upper Band, CI= 0.95
## $OutputGap
##             CPI
##  [1,] 0.1867979
##  [2,] 0.4025774
##  [3,] 0.6093061
##  [4,] 0.9376694
##  [5,] 0.9524677
##  [6,] 0.8961468
##  [7,] 0.9311767
```

```
##  [8,] 0.8024332
##  [9,] 0.7592196
## [10,] 0.7471364
## [11,] 0.7112564
```

```r
plot(svarinf)
```



Inflation increases due to the output gap increase. The increase in the output gap overheats the economy, causing inflationary pressures.

Now we'll see what the Federal Reserve will do to limit inflation.

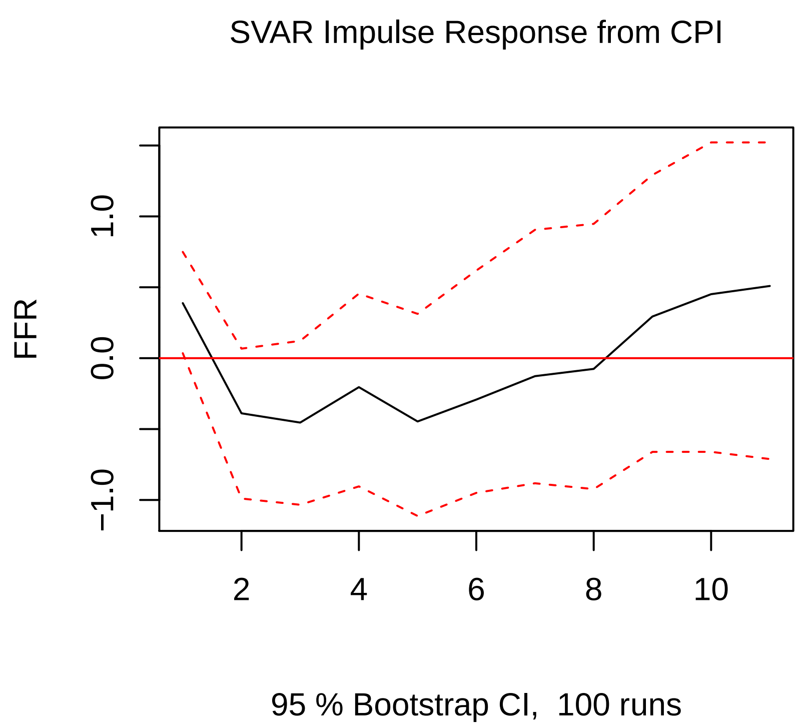
```r
svarffr <- irf(svarmodel1, impulse = "CPI", response = "FFR")
svarffr
```

```
##
## Impulse response coefficients
## $CPI
##                FFR
```

```
##  [1,]  0.38832273
##  [2,] -0.38887536
##  [3,] -0.45425562
##  [4,] -0.20444170
##  [5,] -0.44628795
##  [6,] -0.29202345
##  [7,] -0.12681977
##  [8,] -0.07559652
##  [9,]  0.29389181
## [10,]  0.45133447
## [11,]  0.50910146
##
##
## Lower Band, CI= 0.95
## $CPI
##               FFR
##  [1,]  0.03497692
##  [2,] -0.98986753
##  [3,] -1.03387645
##  [4,] -0.90421321
##  [5,] -1.11298171
##  [6,] -0.95004801
##  [7,] -0.88222789
##  [8,] -0.92336099
##  [9,] -0.66089465
## [10,] -0.66050975
## [11,] -0.71135722
##
##
## Upper Band, CI= 0.95
## $CPI
##              FFR
##  [1,] 0.74950272
```

```
##  [2,] 0.06719104
##  [3,] 0.12204065
##  [4,] 0.45514903
##  [5,] 0.31263806
##  [6,] 0.61812573
##  [7,] 0.90521555
##  [8,] 0.94726417
##  [9,] 1.29178730
## [10,] 1.52180704
## [11,] 1.52153583
```

```
plot(svarffr)
```



This response is unexpected. Usually, central banks will fight inflation by increasing their policy rate. Here we see that the ffr is decreasing. This might be an error, or possibly there is a story to tell. This could be the result of the Volcker Disinflation period.

Hopefully this helps anyone wanting to see an example of how to use SVARs in R.