Vector Autoregression (VAR)

Vipul Bhatt

Application: Estimating VAR model for monetary policy rule

In this workbook we will estimate a vector autoregression model (VAR) for three variables: inflation (π_t), unemployment (u_t), and the federal funds rate (i_t). We will use quarterly data from 1960Q1 through 2019Q4. This file is available on Canvas as "var.csv". Figure 1 below plots these three variables:

```
library(readr)
## import data
data <- read_csv("C:/Users/bhattvx/Dropbox/In Class/SP 2023/datasets/var.csv")

### declare time series for each variable
inf=ts(data$inflation, start=c(1960,1), end=c(2019,4), frequency=4)
u=ts(data$u, start=c(1960,1), end=c(2019,4), frequency=4)
ffr=ts(data$ffr, start=c(1960,1), end=c(2019,4), frequency=4)

### plot them in same plot
ts.plot(cbind(inf,u,ffr), col=c("black", "red","steelblue"),
        lty=c("solid", "dashed", "dotted"))
legend("topright", bty = "n", legend=c("Inflation", "Unemployment", "FFR"),
       fill=c("black", "red","steelblue"))
```

1. First step is to test for stationarity for each variable, and determine if we need to difference our data to obtain stationarity. From Figure 1 it is clear that our data has no apparent trend and also the long run mean for each variable is non-zero. In this case it make sense to test for unit root using "drift" version of the ADF test. We find that all three variables are I(1), i.e., first difference stationary. Hence, we will estimate our model using first difference of each time series in our data.

```
library(urca)
### test for unit root in levels for each
### below I just use model with drift as there is
### is no apparent trend in data and reasonable
### to assume each has non-zero mean

summary(ur.df(inf,type="drift", lags=10, selectlags="AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
```

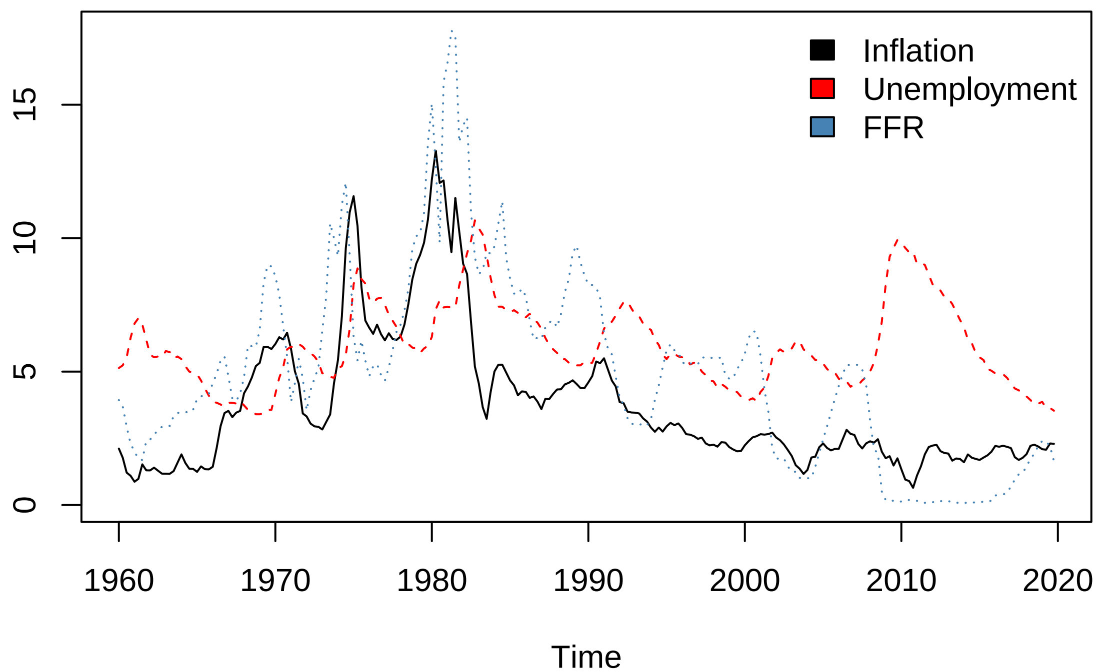


Figure 1: Inflation, Unemployment, and FFR

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.61224 -0.16247 -0.00995  0.15939  1.70351
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09493    0.04966   1.911  0.05725 .
## z.lag.1     -0.02455    0.01117  -2.197  0.02904 *
## z.diff.lag1  0.57998    0.06631   8.747 6.03e-16 ***
## z.diff.lag2  0.06055    0.07328   0.826  0.40952
## z.diff.lag3  0.20564    0.07338   2.802  0.00553 **
## z.diff.lag4 -0.64675    0.07445  -8.687 8.93e-16 ***
## z.diff.lag5  0.35715    0.08318   4.294 2.64e-05 ***
## z.diff.lag6  0.08583    0.07363   1.166  0.24505
## z.diff.lag7  0.04734    0.07288   0.650  0.51667
## z.diff.lag8 -0.33032    0.07290  -4.531 9.66e-06 ***
## z.diff.lag9  0.16774    0.06642   2.525  0.01227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3892 on 218 degrees of freedom
## Multiple R-squared:  0.4692, Adjusted R-squared:  0.4449
## F-statistic: 19.27 on 10 and 218 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -2.1975 2.4163
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
summary(ur.df(u,type="drift", lags=10 ,selectlags="AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88126 -0.12419 -0.02254  0.10964  1.00881
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.191956   0.070719   2.714 0.007172 **
## z.lag.1     -0.032579   0.011477  -2.839 0.004960 **
## z.diff.lag1  0.676332   0.066131  10.227  < 2e-16 ***
```

```
## z.diff.lag2  0.061236   0.077793   0.787 0.432039
## z.diff.lag3  0.059846   0.077902   0.768 0.443188
## z.diff.lag4 -0.183053   0.077528  -2.361 0.019102 *
## z.diff.lag5  0.006242   0.078489   0.080 0.936689
## z.diff.lag6  0.160703   0.076972   2.088 0.037976 *
## z.diff.lag7  0.078135   0.077692   1.006 0.315671
## z.diff.lag8 -0.262872   0.077572  -3.389 0.000833 ***
## z.diff.lag9  0.142716   0.066003   2.162 0.031687 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2296 on 218 degrees of freedom
## Multiple R-squared:  0.5088, Adjusted R-squared:  0.4863
## F-statistic: 22.59 on 10 and 218 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -2.8385 4.0649
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
summary(ur.df(ffr,type="drift", lags=10,selectlags="AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6743 -0.2349 -0.0083  0.2521  6.2828
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14954    0.09790   1.527  0.12807
## z.lag.1     -0.03033    0.01597  -1.899  0.05882 .
## z.diff.lag1  0.31372    0.06576   4.771 3.34e-06 ***
## z.diff.lag2 -0.20776    0.06856  -3.031  0.00273 **
## z.diff.lag3  0.22760    0.06921   3.289  0.00117 **
## z.diff.lag4 -0.03423    0.07062  -0.485  0.62838
## z.diff.lag5  0.18326    0.06945   2.639  0.00891 **
## z.diff.lag6 -0.09098    0.06869  -1.325  0.18669
## z.diff.lag7 -0.18127    0.06635  -2.732  0.00681 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.816 on 220 degrees of freedom
```

```
## Multiple R-squared:  0.2098, Adjusted R-squared:  0.181
## F-statistic:   7.3 on 8 and 220 DF,  p-value: 1.356e-08
##
##
## Value of test-statistic is: -1.8994 1.8094
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```r
summary(ur.df(na.omit(diff(inf)),type="drift", lags=10, selectlags="AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71725 -0.13774  0.01437  0.15478  1.72679
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.003305   0.025676   0.129 0.897715
## z.lag.1      -0.740019   0.122643  -6.034 6.88e-09 ***
## z.diff.lag1   0.314182   0.117483   2.674 0.008062 **
## z.diff.lag2   0.362500   0.112838   3.213 0.001517 **
## z.diff.lag3   0.504831   0.102420   4.929 1.65e-06 ***
## z.diff.lag4  -0.148690   0.102989  -1.444 0.150262
## z.diff.lag5   0.239242   0.099336   2.408 0.016861 *
## z.diff.lag6   0.314226   0.094573   3.323 0.001047 **
## z.diff.lag7   0.264596   0.075931   3.485 0.000596 ***
## z.diff.lag8  -0.043817   0.076316  -0.574 0.566464
## z.diff.lag9   0.163756   0.071917   2.277 0.023766 *
## z.diff.lag10  0.157096   0.066788   2.352 0.019565 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3877 on 216 degrees of freedom
## Multiple R-squared:  0.5012, Adjusted R-squared:  0.4758
## F-statistic: 19.73 on 11 and 216 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -6.0339 18.2053
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
```

```
## phi1  6.52  4.63  3.81
```

```r
summary(ur.df(na.omit(diff(u)),type="drift", lags=10,selectlags="AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.92158 -0.12877 -0.01645  0.11596  0.99050
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.003938   0.015508  -0.254   0.7998
## z.lag.1     -0.415090   0.085519  -4.854 2.31e-06 ***
## z.diff.lag1  0.090873   0.087244   1.042   0.2988
## z.diff.lag2  0.138802   0.086031   1.613   0.1081
## z.diff.lag3  0.181411   0.084151   2.156   0.0322 *
## z.diff.lag4 -0.028439   0.079238  -0.359   0.7200
## z.diff.lag5 -0.041926   0.072671  -0.577   0.5646
## z.diff.lag6  0.109765   0.070143   1.565   0.1191
## z.diff.lag7  0.173351   0.067942   2.551   0.0114 *
## z.diff.lag8 -0.106454   0.066058  -1.612   0.1085
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2338 on 218 degrees of freedom
## Multiple R-squared:  0.2516, Adjusted R-squared:  0.2207
## F-statistic: 8.144 on 9 and 218 DF,  p-value: 2.09e-10
##
##
## Value of test-statistic is: -4.8538 11.7797
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```r
summary(ur.df(na.omit(diff(ffr)),type="drift", lags=10,selectlags="AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
```

```
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0022 -0.2359  0.0189  0.2898  6.1627
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.005598   0.054480  -0.103 0.918258
## z.lag.1     -0.896181   0.134179  -6.679 1.94e-10 ***
## z.diff.lag1  0.199657   0.125990   1.585 0.114471
## z.diff.lag2 -0.021112   0.120425  -0.175 0.860994
## z.diff.lag3  0.194027   0.109312   1.775 0.077284 .
## z.diff.lag4  0.145077   0.099389   1.460 0.145800
## z.diff.lag5  0.310186   0.078321   3.960 0.000101 ***
## z.diff.lag6  0.202211   0.065996   3.064 0.002457 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8226 on 220 degrees of freedom
## Multiple R-squared:  0.4736, Adjusted R-squared:  0.4569
## F-statistic: 28.28 on 7 and 220 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -6.679 22.3081
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
### create differenced data and drop missing observation

d_inf=na.omit(diff(inf))
d_u=na.omit(diff(u))
d_ffr=na.omit(diff(ffr))
```

2. Step 2 is to select the optimal number of lags for our VAR model. Do this we first have to declare an **ordering**. In this example I assume the following ordering for Cholesky decomposition: inflation, unemployment, ffr. This implies, inflation is the most "exogenous" variable followed by unemployment, and then ffr. Using SC criterion (which is the same is BIC) we find that the best model is VAR(4). We store the estimation result in an object named "varmodel". You can use "summary(varmodel)" to see estimated coefficients for each equation. We will use these coefficients to generate relevant output later.

```
library(vars)

### declare vector of y:
yvector= ts.union(d_inf,d_u,d_ffr)

## select optimal lag:
VARselect(yvector,lag.max=8,type=c("const"))
```

```
## $selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      8      4      4      8
##
## $criteria
##                   1            2            3            4            5
## AIC(n) -5.148960013 -5.333639960 -5.479863267 -5.657504739 -5.669709412
## HQ(n)  -5.076832907 -5.207417525 -5.299545502 -5.423091645 -5.381200989
## SC(n)  -4.970133119 -5.020692895 -5.032796032 -5.076317333 -4.954401836
## FPE(n)  0.005805499  0.004826739  0.004170577  0.003492465  0.003451167
##                   6            7            8
## AIC(n) -5.724291239 -5.685951264 -5.796020257
## HQ(n)  -5.381687486 -5.289252181 -5.345225845
## SC(n)  -4.874863493 -4.702403347 -4.678352169
## FPE(n)  0.003269318  0.003399204  0.003047398
```

```
## estimate VAR(4)
varmodel=VAR(yvector,p=4, type=c("const"))
summary(varmodel)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: d_inf, d_u, d_ffr
## Deterministic variables: const
## Sample size: 235
## Log Likelihood: -301.313
## Roots of the characteristic polynomial:
## 0.8124 0.8124 0.7868 0.7868 0.7742 0.7742 0.7045 0.6171 0.6171 0.5466 0.5445 0.5445
## Call:
## VAR(y = yvector, p = 4, type = c("const"))
##
##
## Estimation results for equation d_inf:
## ======================================
## d_inf = d_inf.l1 + d_u.l1 + d_ffr.l1 + d_inf.l2 + d_u.l2 + d_ffr.l2 + d_inf.l3 + d_u.l3 + d_ffr.l3 +
##
##           Estimate Std. Error t value Pr(>|t|)
## d_inf.l1  0.388015   0.064750   5.993 8.26e-09 ***
## d_u.l1   -0.182179   0.113201  -1.609  0.10896
## d_ffr.l1  0.165884   0.035924   4.618 6.57e-06 ***
## d_inf.l2  0.097274   0.067633   1.438  0.15177
## d_u.l2    0.154877   0.129698   1.194  0.23370
## d_ffr.l2 -0.023927   0.038379  -0.623  0.53363
## d_inf.l3  0.104303   0.071026   1.469  0.14338
## d_u.l3   -0.095865   0.130015  -0.737  0.46169
## d_ffr.l3  0.195821   0.035813   5.468 1.22e-07 ***
## d_inf.l4 -0.258927   0.062411  -4.149 4.76e-05 ***
## d_u.l4   -0.017824   0.111721  -0.160  0.87339
## d_ffr.l4 -0.108712   0.036449  -2.983  0.00318 **
## const     0.002488   0.024010   0.104  0.91757
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.3675 on 222 degrees of freedom
```

```
## Multiple R-Squared: 0.5208,  Adjusted R-squared: 0.4949
## F-statistic: 20.11 on 12 and 222 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation d_u:
## ====================================
## d_u = d_inf.l1 + d_u.l1 + d_ffr.l1 + d_inf.l2 + d_u.l2 + d_ffr.l2 + d_inf.l3 + d_u.l3 + d_ffr.l3 + d
##
##           Estimate Std. Error t value Pr(>|t|)
## d_inf.l1  0.133874   0.040600   3.297  0.00114 **
## d_u.l1    0.600285   0.070981   8.457 3.69e-15 ***
## d_ffr.l1 -0.009796   0.022526  -0.435  0.66407
## d_inf.l2 -0.025422   0.042409  -0.599  0.54948
## d_u.l2    0.102806   0.081326   1.264  0.20751
## d_ffr.l2  0.056415   0.024065   2.344  0.01995 *
## d_inf.l3  0.024648   0.044536   0.553  0.58052
## d_u.l3    0.126525   0.081524   1.552  0.12209
## d_ffr.l3  0.008983   0.022456   0.400  0.68951
## d_inf.l4 -0.036478   0.039134  -0.932  0.35228
## d_u.l4   -0.118944   0.070053  -1.698  0.09093 .
## d_ffr.l4  0.003940   0.022855   0.172  0.86327
## const    -0.006698   0.015055  -0.445  0.65683
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2304 on 222 degrees of freedom
## Multiple R-Squared: 0.5113,  Adjusted R-squared: 0.4849
## F-statistic: 19.36 on 12 and 222 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation d_ffr:
## ======================================
## d_ffr = d_inf.l1 + d_u.l1 + d_ffr.l1 + d_inf.l2 + d_u.l2 + d_ffr.l2 + d_inf.l3 + d_u.l3 + d_ffr.l3 +
##
##           Estimate Std. Error t value Pr(>|t|)
## d_inf.l1 -0.763440   0.129108  -5.913 1.26e-08 ***
## d_u.l1   -0.939495   0.225718  -4.162 4.51e-05 ***
## d_ffr.l1  0.341338   0.071632   4.765 3.41e-06 ***
## d_inf.l2  0.752875   0.134858   5.583 6.88e-08 ***
## d_u.l2    0.092406   0.258613   0.357  0.72120
## d_ffr.l2 -0.239104   0.076527  -3.124  0.00202 **
## d_inf.l3  0.010997   0.141622   0.078  0.93818
## d_u.l3   -0.402894   0.259245  -1.554  0.12158
## d_ffr.l3  0.074955   0.071410   1.050  0.29502
## d_inf.l4 -0.131013   0.124445  -1.053  0.29359
## d_u.l4    0.558502   0.222767   2.507  0.01289 *
## d_ffr.l4  0.204245   0.072678   2.810  0.00539 **
## const    -0.007773   0.047874  -0.162  0.87116
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.7328 on 222 degrees of freedom
```

```
## Multiple R-Squared: 0.3592,  Adjusted R-squared: 0.3245
## F-statistic: 10.37 on 12 and 222 DF,  p-value: 3.688e-16
##
##
##
## Covariance matrix of residuals:
##          d_inf      d_u    d_ffr
## d_inf 0.135071  0.005545  0.07640
## d_u   0.005545  0.053106 -0.05353
## d_ffr 0.076405 -0.053528  0.53702
##
## Correlation matrix of residuals:
##         d_inf     d_u   d_ffr
## d_inf 1.00000  0.06547  0.2837
## d_u   0.06547  1.00000 -0.3170
## d_ffr 0.28369 -0.31696  1.0000
```

3. We can use our estimated model to describe our data. Below we use our model to test for Granger-causality, as well as for computing forecast for next 4 quarters for each variable. I also show how to manually compute RMSE for each forecast (in-sample accuracy measure).

```
causality(varmodel,cause="d_inf")$Granger
```

```
##
##  Granger causality H0: d_inf do not Granger-cause d_u d_ffr
##
## data:  VAR object varmodel
## F-Test = 7.6808, df1 = 8, df2 = 666, p-value = 7.197e-10
```

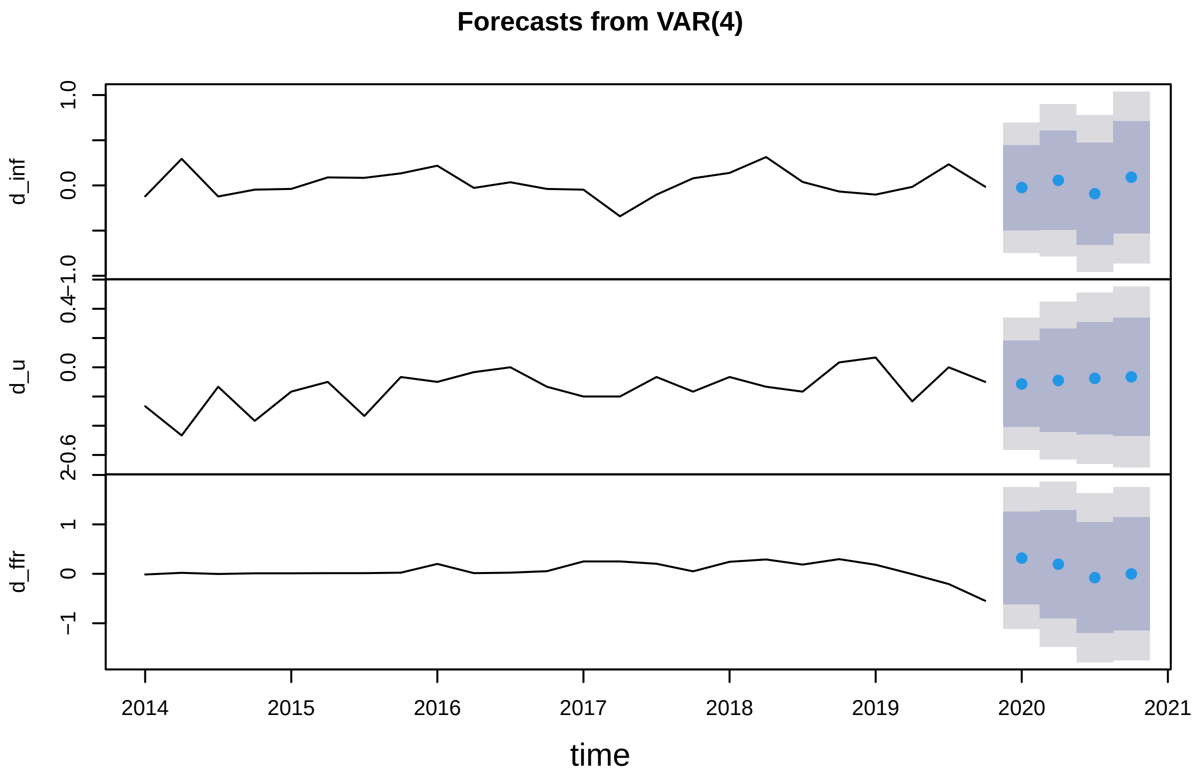
```
causality(varmodel,cause="d_u")$Granger
```

```
##
##  Granger causality H0: d_u do not Granger-cause d_inf d_ffr
##
## data:  VAR object varmodel
## F-Test = 4.0378, df1 = 8, df2 = 666, p-value = 0.0001061
```

```
causality(varmodel,cause="d_ffr")$Granger
```

```
##
##  Granger causality H0: d_ffr do not Granger-cause d_inf d_u
##
## data:  VAR object varmodel
## F-Test = 7.251, df1 = 8, df2 = 666, p-value = 2.993e-09
```

```
library(forecast)
fcast = forecast(varmodel, h = 4)
plot(fcast,include=24)
```

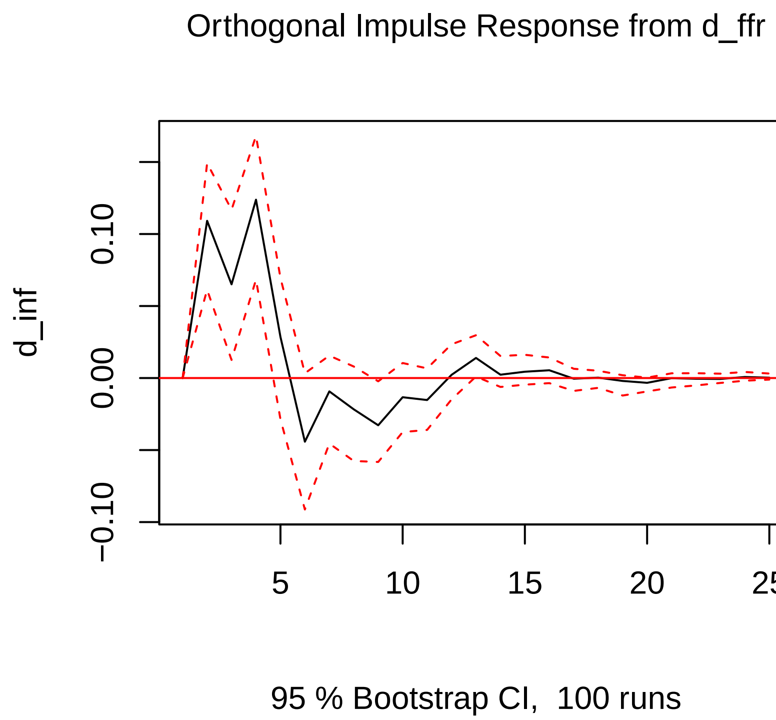



```r
### residuals for each variable of the var model
e=residuals(varmodel)

### manually compute RMSE-insample for each variable
sqrt(colMeans(e^2))
```

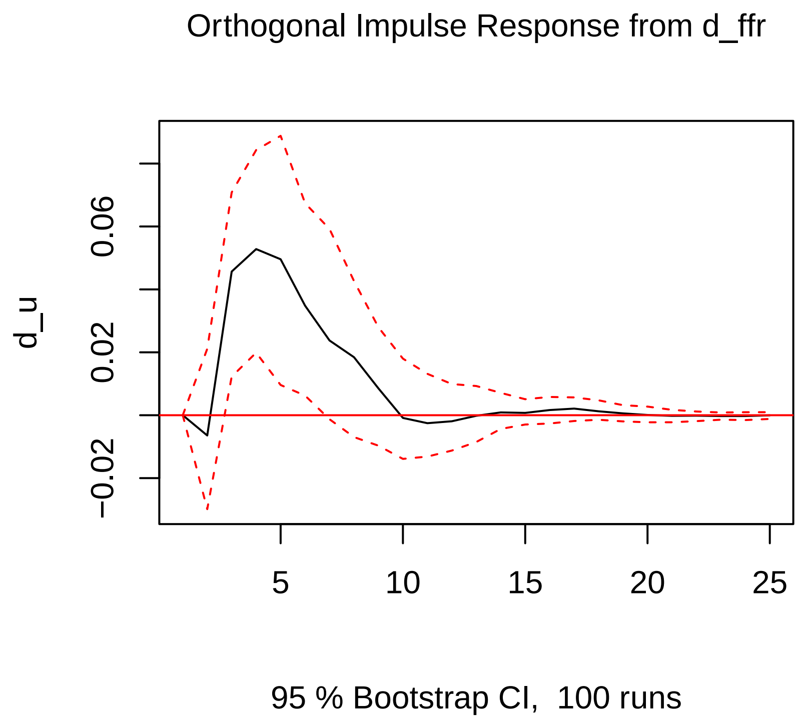
```
##     d_inf       d_u     d_ffr
## 0.3572096 0.2239837 0.7122595
```

4. We can also compute impulse response functions using our varmodel. Below, I show the impulse response functions for inflation and unemployment for orthogonal shock to FFR of size 1. Note that R only codes positive shock, and hence this is an example of a contractionary monetary policy where the FED is raising the FFR.

```r
### compute response for 24 quarters to an orthogonal shock to ffr
impulse_inf=irf(varmodel,impulse="d_ffr",ortho = TRUE, n.ahead=24, response=c("d_inf"))
impulse_u=irf(varmodel,impulse="d_ffr", ortho=TRUE, n.ahead=24, response=c("d_u"))


## plot IRF
plot(impulse_inf)
```




```
plot(impulse_u)
```



5. Finally we can compute variance decomposition for each variable.

```
fevd(varmodel,n.ahead=12)
```

```
## $d_inf
##            d_inf        d_u      d_ffr
##  [1,] 1.0000000 0.00000000 0.00000000
##  [2,] 0.8981289 0.03718522 0.06468585
##  [3,] 0.8602072 0.05726245 0.08253039
##  [4,] 0.7705285 0.09537619 0.13409528
##  [5,] 0.7255697 0.14964626 0.12478404
##  [6,] 0.7203375 0.15123984 0.12842263
##  [7,] 0.7155638 0.15668277 0.12775346
##  [8,] 0.7201577 0.15338904 0.12645328
##  [9,] 0.7148129 0.15592931 0.12925778
## [10,] 0.7140539 0.15631267 0.12963340
## [11,] 0.7122080 0.15769104 0.13010098
## [12,] 0.7116119 0.15886764 0.12952047
##
## $d_u
##             d_inf       d_u        d_ffr
##  [1,] 0.004286249 0.9957138 0.0000000000
##  [2,] 0.044564601 0.9548888 0.0005465932
##  [3,] 0.082514447 0.8937163 0.0237692641
##  [4,] 0.096492999 0.8540469 0.0494600699
##  [5,] 0.108172152 0.8208444 0.0709834110
##  [6,] 0.109060265 0.8095169 0.0814228796
```

```
##  [7,] 0.108378001 0.8054457 0.0861763377
##  [8,] 0.107808811 0.8034022 0.0887889653
##  [9,] 0.108917138 0.8020112 0.0890716739
## [10,] 0.109682961 0.8013571 0.0889599690
## [11,] 0.109854620 0.8011559 0.0889895121
## [12,] 0.109923211 0.8010651 0.0890117349
##
## $d_ffr
##            d_inf       d_u     d_ffr
##  [1,] 0.08047974 0.1130697 0.8064506
##  [2,] 0.12822328 0.2072577 0.6645190
##  [3,] 0.12394656 0.2087824 0.6672711
##  [4,] 0.12894879 0.2276674 0.6433838
##  [5,] 0.13413906 0.2274240 0.6384370
##  [6,] 0.13329299 0.2260663 0.6406407
##  [7,] 0.13305101 0.2275967 0.6393523
##  [8,] 0.13300708 0.2256277 0.6413652
##  [9,] 0.13553895 0.2254947 0.6389664
## [10,] 0.13514005 0.2248825 0.6399774
## [11,] 0.13554487 0.2248215 0.6396336
## [12,] 0.13598977 0.2246462 0.6393640
```