Neural additive VAR

Proposal

Marc Agustí (marc.agusti@barcelonagse.eu)
Patrick Altmeyer (patrick.altmeyer@barcelonagse.eu)
Ignacio Vidal-Quadras Costa (ignacio.vidalquadrascosta@barcelonagse.eu)

May, 2021

1 Literature review

1.1 Bussmann, Nys, and Latré (2020)

1.1.1 Motivation

- In many time series applications the functional dependence of some variable $X_t^{(i)}$ on past lags of some other variable $X_t^{(j)}$ can be expected to be non-linear.
- At the same time, dependencies with respect to multiple covariates can usually be modelled well through additive models (e.g. VAR).

1.1.2 Methodology

- Propose a Neural Additive Vector Autoregression (NAVAR) model for causal discovery in time series data
- Let the equation below denote the standard linear VAR where each variable in the system depends linearly on its own lags and those of its covariates:

$$\mathbf{X}_t{}^{(j)} = \beta^j + \sum_{i=1}^{N} \sum_{k=1}^{K} [A_k]_{ij} \mathbf{X}_{t-k}{}^{(i)} + \eta_t^j \tag{1}$$

Definition 1.1 (Granger causality)**.** Variable $X^{(i)}$ is said to Granger cause another variable $X^{(j)}$ if the past of the set of all (input) variables $\{X_{<t}^{(1)}, ..., X_{<t}^{(i)}, ..., X_{<t}^{(N)}\}$ yields better predictions of $X^{(j)}$ then if $X_{<t}^{(i)}$ was excluded.

- The NAVAR model instead allows for non-linear interactions between covariates where f_{ij} is the i-th output from a deep neural network that maps from all of j-th past lags (up to K) to all covariates:

$$\mathbf{X}_t{}^{(j)} = \beta^j + \sum_{i=1}^{N} f_{ij} \left(\mathbf{X}_{t-K:t-1}{}^{(i)} \right) + \eta_t^j \tag{2}$$

- Notice that if f is linear we are just back to the simple VAR case.
- In order to make the contributions comparable, every individual time series is normalized such that it has mean zero and standard deviation one before training.

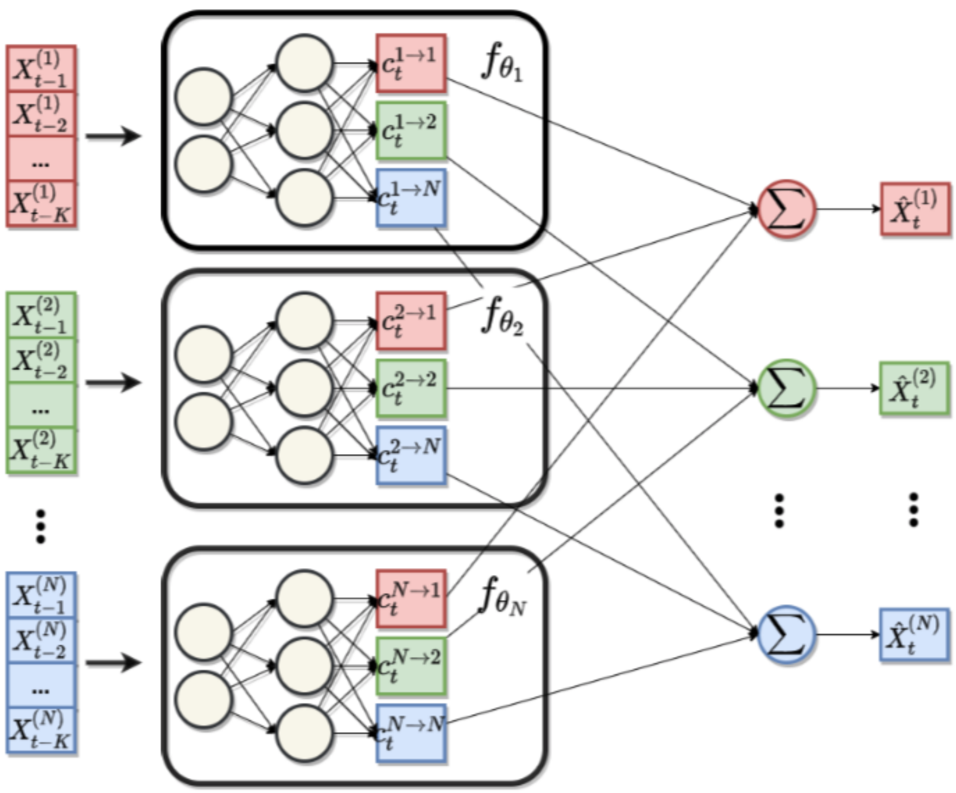


Figure 1: Graphical illustration of NAVAR model with MLPs. Source: Bussmann, Nys, and Latré (2020)

1.1.3 Experiments

1.1.3.1 Toy data set

1.1.3.2 CauseMe data sets

- where the performance of most methods declines as the number of variables N increases, the performance of NAVAR does not decrease.

1.2 Reduced form VAR analysis

```
library(data.table)
dt <- fread("data_VAR/preprocessed.csv")
```

Here I will use my `SVAA` package to run a couple of standard analyses in the VAR context.

1.2.1 Sanity checks

```
p <- 12
```

First, let us check if the VAR is stable. For this, I first run the reduced-form VAR with a conventional (?) choice of $p = 12$ lags (reflecting 1 year given the monthly frequency).

```
library(SVAA)
countries <- dt[,unique(country)] # run analysis by country
var_conventional <- lapply(
  countries,
  function(country) {
    dt_mod <- copy(dt)
    dt_mod <- dt_mod[country==country][,country:=NULL][,date:=NULL] # retain only VAR variables
    var <- VAR(dt_mod, lag=p)
    return(var)
  }
)
```

```
##          from  to    estimate         se    t_value      p_value signif
##   1: constant CPI  0.02912474 0.01284907  2.2666799 2.372824e-02      *
##   2:   CPI_l1 CPI  0.17853686 0.03913030  4.5626245 6.010235e-06    ***
##   3:   GDP_l1 CPI  0.24605213 0.03202246  7.6837368 5.516546e-14    ***
##   4:    IR_l1 CPI  0.11734348 0.02909679  4.0328673 6.144818e-05    ***
##   5:    UE_l1 CPI  0.01369769 0.03576938  0.3829445 7.018826e-01
##  ---
## 192:   UE_l11  UE  0.11675866 0.04021319  2.9034917 3.811850e-03     **
## 193:  CPI_l12  UE  0.03804748 0.04347475  0.8751627 3.817995e-01
## 194:  GDP_l12  UE  0.43948312 1.01302777  0.4338313 6.645508e-01
## 195:   IR_l12  UE -0.04803962 0.03145445 -1.5272759 1.271651e-01
## 196:   UE_l12  UE -0.12296896 0.03990691 -3.0813954 2.144897e-03     **
```

```
names(var_conventional) <- countries
```

Running the test I find that with the conventional choice, the VAR is not stable for the US.

```
sapply(var_conventional, VAR_stable)
```

```
## The VAR is not stable
```

```
##             US
## eigenvals   Complex,48
## test_result "The VAR is not stable"
```

Let's instead try lag-length selection first. Here, for the US, the more conservative measures suggest using just 5 lags while the less conservative AIC suggest using 10. A reasonable choice seems to be $p = 6$ reflecting half a year.

```
lag_selection <- lapply(
  countries,
  function(country) {
    dt_mod <- copy(dt)
    dt_mod <- dt_mod[country==country][,country:=NULL][,date:=NULL] # retain only VAR variables
    lag_selection <- VAR_lag_select(dt_mod)
    return(lag_selection)
  }
)
names(lag_selection) <- countries
lag_selection[["US"]]$proposed_lag_lengths
```

```
##     ic       min lag
## 1: BIC -13.33502   5
## 2: HQC -13.66302   5
## 3: AIC -13.93284  10
```

So, let's rerun the reduced-form VAR for the new choice of p.

```
p <- 6
vars <- lapply(
  countries,
  function(country) {
    dt_mod <- copy(dt)
    dt_mod <- dt_mod[country==country][,country:=NULL][,date:=NULL] # retain only VAR variables
    var <- VAR(dt_mod, lag=p)
    return(var)
  }
)
```

```
##          from  to      estimate         se      t_value      p_value signif
##   1: constant CPI  0.0431857121 0.01237511  3.489724186 5.138600e-04    ***
##   2:   CPI_l1 CPI  0.2251224358 0.03802795  5.919920476 5.047869e-09    ***
##   3:   GDP_l1 CPI  0.2463150320 0.03221869  7.645096860 6.914833e-14    ***
##   4:    IR_l1 CPI  0.1105931118 0.02907385  3.803869163 1.549178e-04    ***
##   5:    UE_l1 CPI -0.0170432633 0.03417671 -0.498680586 6.181613e-01
##   6:   CPI_l2 CPI  0.1921024383 0.03842752  4.999084809 7.289752e-07    ***
##   7:   GDP_l2 CPI -0.1481963984 0.08229045 -1.800894259 7.215046e-02      .
##   8:    IR_l2 CPI  0.0383690711 0.03053298  1.256643361 2.093026e-01
##   9:    UE_l2 CPI -0.0186312317 0.03380935 -0.551067379 5.817636e-01
##  10:   CPI_l3 CPI  0.0861958689 0.03927531  2.194657715 2.851639e-02      *
##  11:   GDP_l3 CPI -0.1561260448 0.09810664 -1.591391192 1.119737e-01
##  12:    IR_l3 CPI  0.0291610936 0.03021384  0.965156934 3.348000e-01
##  13:    UE_l3 CPI -0.0395494426 0.03344396 -1.182558734 2.373862e-01
##  14:   CPI_l4 CPI  0.0304363912 0.03952977  0.769961251 4.415830e-01
##  15:   GDP_l4 CPI -0.0624078516 0.09235957 -0.675705283 4.994513e-01
##  16:    IR_l4 CPI  0.0011568279 0.02970450  0.038944538 9.689457e-01
```

```
## 17:    UE_l4 CPI -0.0274036263 0.02849110  -0.961830974  3.364671e-01
## 18:   CPI_l5 CPI  0.1698993772 0.03908811   4.346574051  1.588202e-05    ***
## 19:   GDP_l5 CPI  0.0688914123 0.09445653   0.729345176  4.660349e-01
## 20:    IR_l5 CPI -0.0755229354 0.02912598  -2.592974486  9.713998e-03     **
## 21:    UE_l5 CPI -0.0065544529 0.02611902  -0.250945592  8.019299e-01
## 22:   CPI_l6 CPI  0.1728500844 0.03870155   4.466231528  9.281466e-06    ***
## 23:   GDP_l6 CPI -0.0251883409 0.05403958  -0.466109127  6.412826e-01
## 24:    IR_l6 CPI  0.0093463568 0.02803561   0.333374429  7.389516e-01
## 25:    UE_l6 CPI  0.0273273504 0.02431201   1.124026924  2.613875e-01
## 26: constant GDP  0.0423217923 0.01595934   2.651851753  8.186660e-03     **
## 27:   CPI_l1 GDP -0.0525455984 0.04904206  -1.071439389  2.843417e-01
## 28:   GDP_l1 GDP  0.7581307521 0.04155026  18.246113043  3.974650e-61    ***
## 29:    IR_l1 GDP  0.0214246437 0.03749457   0.571406629  5.679077e-01
## 30:    UE_l1 GDP  0.1152162544 0.04407539   2.614072527  9.139655e-03     **
## 31:   CPI_l2 GDP -0.1013266716 0.04955736  -2.044633993  4.126504e-02      *
## 32:   GDP_l2 GDP  0.2572011616 0.10612440   2.423581689  1.562113e-02      *
## 33:    IR_l2 GDP -0.0041495132 0.03937632  -0.105380941  9.161038e-01
## 34:    UE_l2 GDP  0.1435762774 0.04360163   3.292911032  1.041478e-03     **
## 35:   CPI_l3 GDP  0.0632550789 0.05065070   1.248848964  2.121384e-01
## 36:   GDP_l3 GDP -0.3465653966 0.12652147  -2.739182463  6.316197e-03     **
## 37:    IR_l3 GDP -0.0131187723 0.03896473  -0.336683239  7.364568e-01
## 38:    UE_l3 GDP  0.0833646162 0.04313040   1.932850573  5.365822e-02      .
## 39:   CPI_l4 GDP -0.0002761378 0.05097886  -0.005416711  9.956797e-01
## 40:   GDP_l4 GDP  0.2109397909 0.11910987   1.770968213  7.700171e-02      .
## 41:    IR_l4 GDP -0.0284522063 0.03830787  -0.742724761  4.578979e-01
## 42:    UE_l4 GDP  0.1006860680 0.03674304   2.740275801  6.295449e-03     **
## 43:   CPI_l5 GDP -0.0220468820 0.05040928  -0.437357568  6.619871e-01
## 44:   GDP_l5 GDP  0.1206631260 0.12181417   0.990550843  3.222477e-01
## 45:    IR_l5 GDP  0.0212274586 0.03756180   0.565134158  5.721638e-01
## 46:    UE_l5 GDP -0.0160360431 0.03368393  -0.476074052  6.341704e-01
## 47:   CPI_l6 GDP -0.0204465855 0.04991076  -0.409662870  6.821788e-01
## 48:   GDP_l6 GDP -0.1067822995 0.06969117  -1.532221264  1.259203e-01
## 49:    IR_l6 GDP  0.0260456473 0.03615563   0.720376036  4.715343e-01
## 50:    UE_l6 GDP  0.0183225579 0.03135354   0.584385544  5.591495e-01
## 51: constant  IR  0.0327061846 0.01761594   1.856624332  6.378515e-02      .
## 52:   CPI_l1  IR  0.1180682376 0.05413271   2.181088608  2.950945e-02      *
## 53:   GDP_l1  IR -0.0819947316 0.04586325  -1.787809166  7.423989e-02      .
## 54:    IR_l1  IR  0.3147603133 0.04138656   7.605374370  9.188437e-14    ***
## 55:    UE_l1  IR -0.0446711177 0.04865048  -0.918205019  3.588283e-01
## 56:   CPI_l2  IR -0.0194272494 0.05470150  -0.355150212  7.225842e-01
## 57:   GDP_l2  IR -0.0885342329 0.11714029  -0.755796590  4.500258e-01
## 58:    IR_l2  IR  0.1603269447 0.04346364   3.688760062  2.428499e-04    ***
## 59:    UE_l2  IR  0.0578800380 0.04812755   1.202638433  2.295235e-01
## 60:   CPI_l3  IR -0.1520850799 0.05590833  -2.720257958  6.685231e-03     **
## 61:   GDP_l3  IR  0.2850216056 0.13965461   2.040903685  4.163554e-02      *
## 62:    IR_l3  IR -0.0046613206 0.04300934  -0.108379273  9.137259e-01
## 63:    UE_l3  IR  0.0419416670 0.04760741   0.880990400  3.786258e-01
## 64:   CPI_l4  IR -0.2186975206 0.05627055  -3.886536111  1.113618e-04    ***
## 65:   GDP_l4  IR  0.0717730757 0.13147367   0.545912169  5.853005e-01
## 66:    IR_l4  IR -0.0644538618 0.04228429  -1.524297933  1.278866e-01
## 67:    UE_l4  IR  0.0533965638 0.04055703   1.316579738  1.884110e-01
## 68:   CPI_l5  IR  0.0704194849 0.05564185   1.265584873  2.060834e-01
## 69:   GDP_l5  IR -0.0251996210 0.13445868  -0.187415356  8.513894e-01
## 70:    IR_l5  IR  0.0847834974 0.04146078   2.044908333  4.123790e-02      *
```

```
##  71:    UE_l5  IR -0.0033291407 0.03718037  -0.089540271  9.286782e-01
##  72:   CPI_l6  IR  0.0987907177 0.05509158   1.793209011  7.337171e-02      .
##  73:   GDP_l6  IR  0.0002733226 0.07692523   0.003553094  9.971661e-01
##  74:    IR_l6  IR  0.0261618602 0.03990864   0.655543756  5.123334e-01
##  75:    UE_l6  IR  0.0113601868 0.03460809   0.328252314  7.428192e-01
##  76: constant  UE -0.0005307709 0.01380519  -0.038447195  9.693421e-01
##  77:   CPI_l1  UE -0.0006819526 0.04242251  -0.016075253  9.871789e-01
##  78:   GDP_l1  UE -2.1549221238 0.03594193 -59.955656856 9.745057e-278    ***
##  79:    IR_l1  UE  0.0447152271 0.03243366   1.378667199  1.684384e-01
##  80:    UE_l1  UE  0.0087952034 0.03812622   0.230686459  8.176259e-01
##  81:   CPI_l2  UE  0.0274669332 0.04286826   0.640728894  5.219088e-01
##  82:   GDP_l2  UE  1.7210801672 0.09180005  18.748139421  7.655302e-64    ***
##  83:    IR_l2  UE -0.0241289637 0.03406142  -0.708395730  4.789357e-01
##  84:    UE_l2  UE  0.1617104642 0.03771641   4.287535902  2.060601e-05    ***
##  85:   CPI_l3  UE -0.0953606168 0.04381402  -2.176486136  2.985298e-02      *
##  86:   GDP_l3  UE  0.4457241087 0.10944398   4.072623169  5.180640e-05    ***
##  87:    IR_l3  UE -0.0167011695 0.03370539  -0.495504406  6.203998e-01
##  88:    UE_l3  UE  0.0968419124 0.03730879   2.595686392  9.638411e-03     **
##  89:   CPI_l4  UE  0.0284352127 0.04409789   0.644820304  5.192552e-01
##  90:   GDP_l4  UE -1.2285839624 0.10303278 -11.924205020  5.720776e-30    ***
##  91:    IR_l4  UE -0.0392069064 0.03313719  -1.183169233  2.371443e-01
##  92:    UE_l4  UE  0.0808005371 0.03178358   2.542210334  1.123007e-02      *
##  93:   CPI_l5  UE  0.0412553663 0.04360519   0.946111353  3.444187e-01
##  94:   GDP_l5  UE  0.8751977210 0.10537206   8.305785578  5.135172e-16    ***
##  95:    IR_l5  UE -0.0356341319 0.03249183  -1.096710687  2.731456e-01
##  96:    UE_l5  UE -0.0040727933 0.02913737  -0.139779025  8.888749e-01
##  97:   CPI_l6  UE -0.0068293902 0.04317396  -0.158183097  8.743582e-01
##  98:   GDP_l6  UE -0.1853013731 0.06028447  -3.073782909  2.195865e-03     **
##  99:    IR_l6  UE -0.0333082507 0.03127545  -1.064996704  2.872452e-01
## 100:    UE_l6  UE -0.0430308769 0.02712154  -1.586594318  1.130568e-01
##          from  to      estimate         se       t_value       p_value signif
```

```r
names(vars) <- countries
```

For the US, the VAR is now stable.

```r
sapply(vars, VAR_stable)
```

```
## The VAR is stable.

##             US
## eigenvals   Complex,24
## test_result "The VAR is stable."
```

1.2.2 Reduced-form IRFs

Below I produce reduced-form IRFs with respect to the interest rate. Confidence bands are computed using bootstrapped standard errors. The results are not intuitive: the CPI increases in response to a shock to the interest rate, output falls but not significantly while unemployment seems completely unaffected.

```r
n_ahead <- 36
country <- "US"
irf_IR <- irf(
  vars[[country]],
  imp = "IR",
  structural = F,
```

```
  n_ahead = 36,
  n_bootstrap = 100
)
```

1.2.3 Granger causality

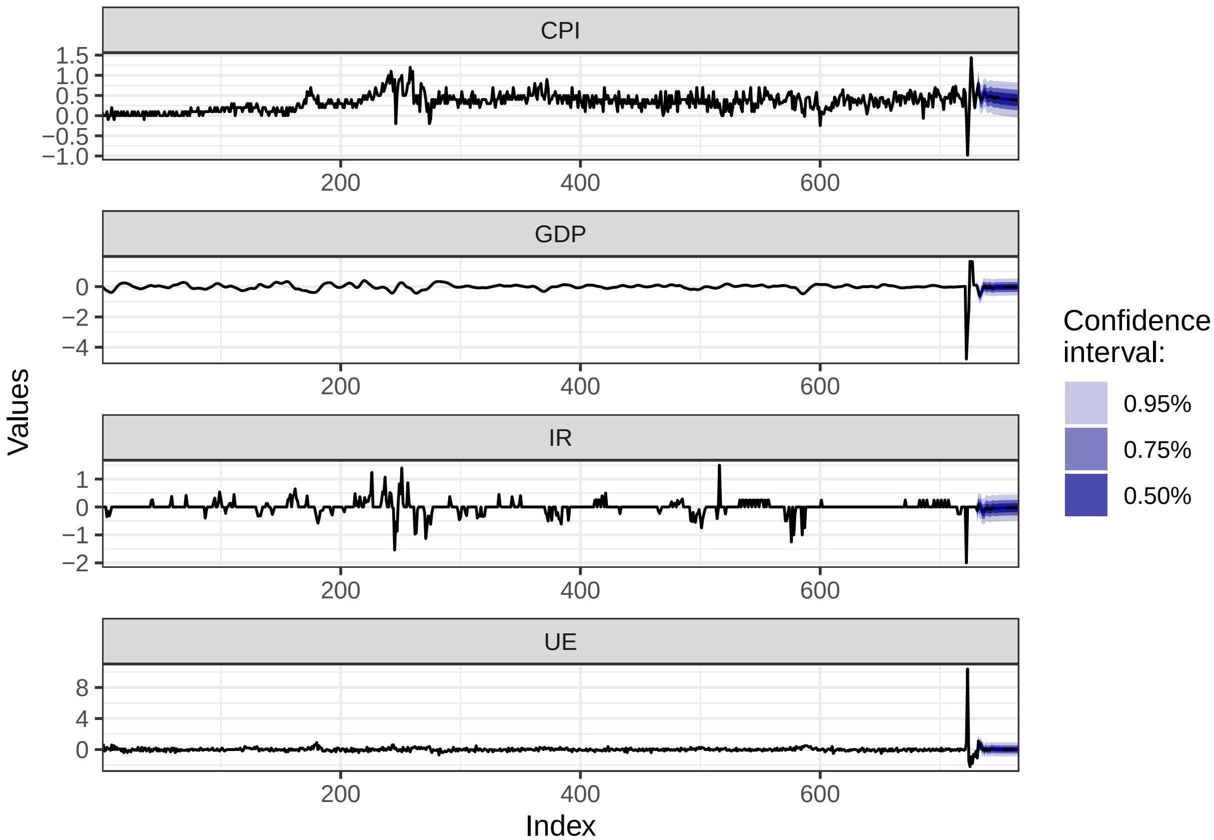
> We should also check for Granger causality as in Bussmann. The latter needs to be added to the package functioanlity. — Pat

1.2.4 Forecasts

Now let's look at forecasts for the economic indicators. I'm running an example for the US below, forecasting out to the conventional policy horizon of three years.

> Here we probably want to do some performance comparison with respect to in-sample and pseudo-out-of-sample forecasts of VAR vs. NAVAR using D Miliano (DM) - or whatever this test is called. — Pat

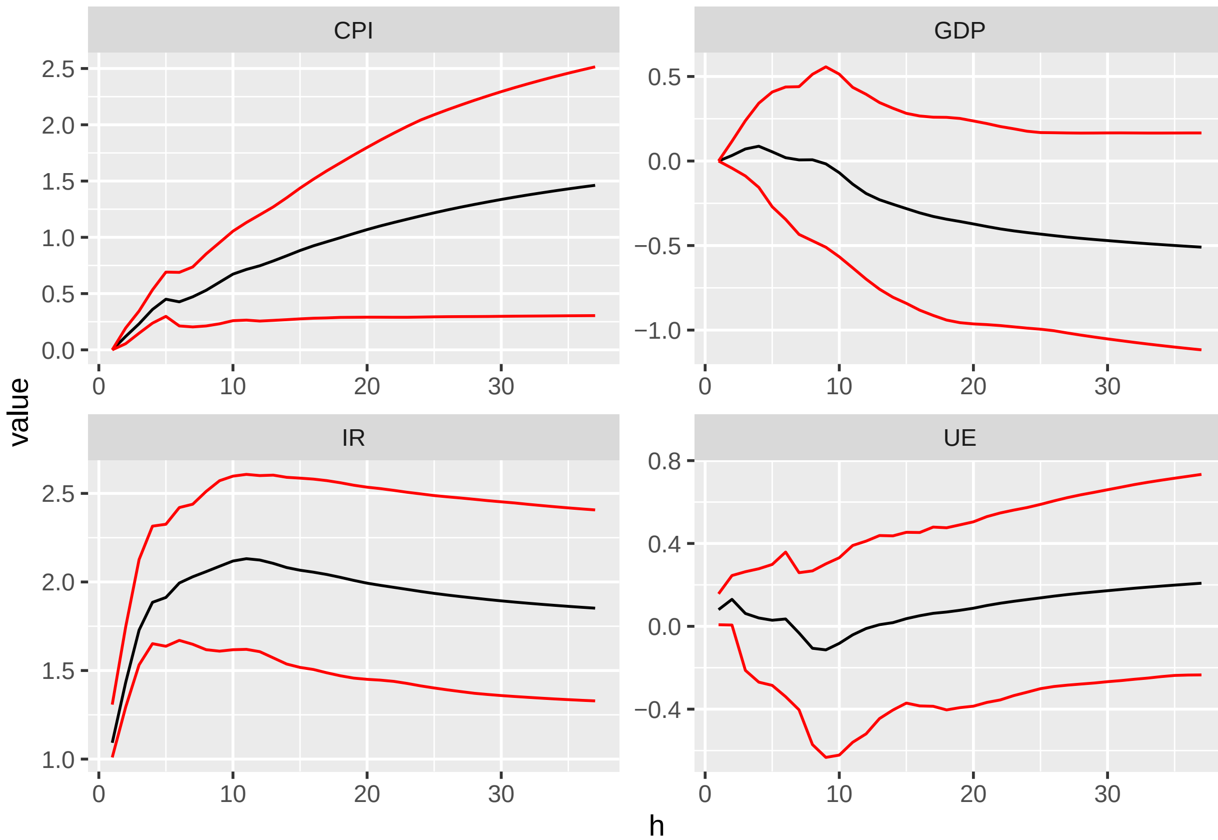
```
my_pred = VAR_predict(vars[[country]], n.ahead=n_ahead, plot = T, theme = theme_bw())
```



1.2.5 Structural IRFs

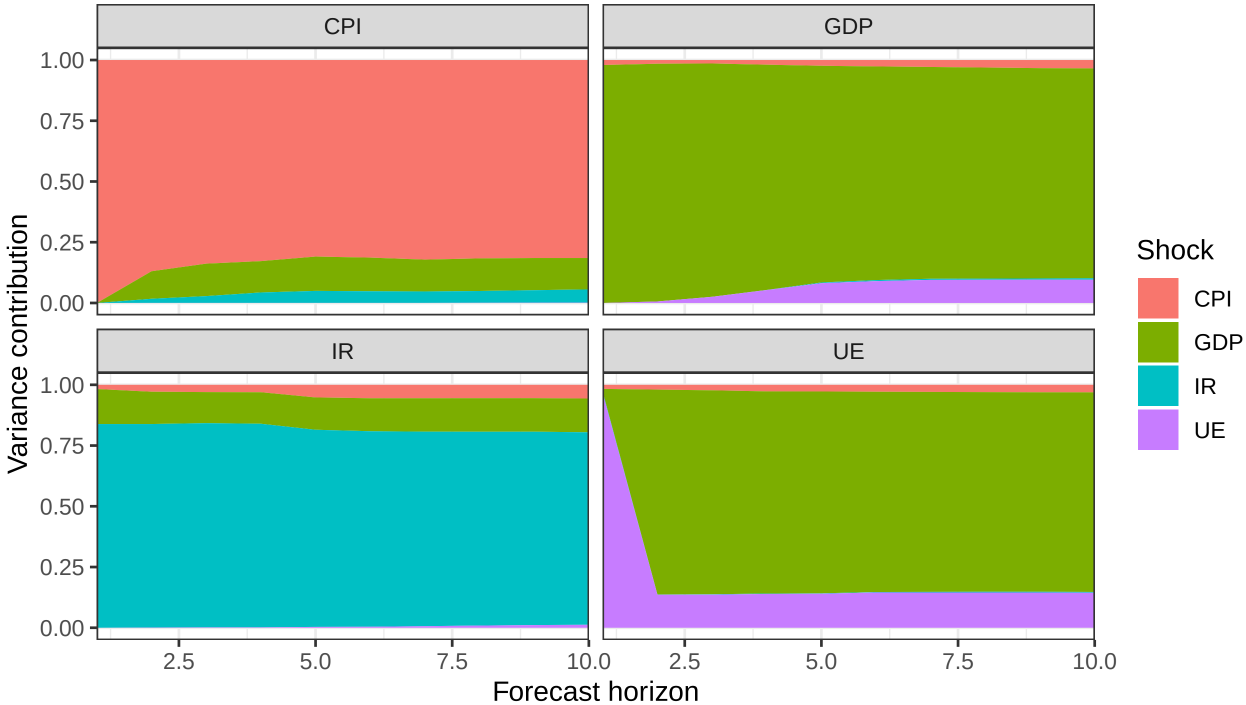
Running simple Cholesky-decomposed GIRF. Haven't taken ordering of variables into account though. — Pat

```
irf_IR <- irf(
  vars[[country]],
  imp = "IR",
  structural = TRUE,
  n_ahead = 36,
  n_bootstrap = 100
)
```



1.2.6 FEVD

```
fevd_output = fevd(vars[[country]])
fevd_output$plot
```



1.2.7 Historical decomposition

```r
hd(varresult = vars[[country]])
```

References

Bussmann, Bart, Jannes Nys, and Steven Latré. 2020. "Neural Additive Vector Autoregression Models for Causal Discovery in Time Series Data." *arXiv Preprint arXiv:2010.09429*.