Problem Set 3 Questions 1 and 6

Pedro Scatimburgo

19/06/2022

Preamble

This file contains Questions 1 and 6 of Problem Set 3. As usual, we start by cleaning the environment and loading the main packages. To estimate the ADL models in Question 1, I rely on `dynlm::dynlm:`.

```r
# Clears the environment
rm(list=ls())

# Loads main packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(vars)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: strucchange
```

```
## Loading required package: sandwich
```

```
##
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':
##
##     boundary
```

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```
library(dynlm)
```

Question 1

Data loading and manipulation

`data_brazil.csv` is the file we will use throughout this problem set. To make my life easier, I create new variables, `gdp`, `exrate` and `ipc`, to store the GDP, exchange rate and inflation rate, respectively, since the name of the original columns are a bit cumbersome. I also filter the dataframe to include only the desired period.

```
data_brazil <- readr::read_csv("data\\data_brazil.csv") %>%
  filter(date >= 1942, date <= 2019) %>%
  mutate(
    gdp = real_gdp_growth_pct,
    exrate = exchange_rate_real_dolar_annual_average,
    ipc = ipc_fipe_pct
  ) %>%
  dplyr::select(date,gdp,exrate,ipc)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   date = col_double(),
##   real_gdp_growth_pct = col_double(),
##   exchange_rate_real_dolar_annual_average = col_double(),
##   ipc_fipe_pct = col_double()
## )
```

Since `dynlm` requires a `ts` object, I will convert the whole dataframe:

```
data_brazil_ts <- data_brazil %>%
  dplyr::select(-date) %>%
  ts(start = 1942,
     end = 2019,
     frequency = 1)
```

Since the `dynlm` function uses a standard error estimator that is not robust to homoskedasticity, I will use `lmtest::coeftest` to use a robust estimator for all models.

Model 1: Run an ADL(2,1) using GDP growth as your dependent variable and Exchange Rate as your predictor.

Report the estimated coefficient and their standard errors.

```
adl_gdp_exrate <- dynlm(data = data_brazil_ts,
                        formula = gdp ~ L(gdp,1) + L(gdp,2) + L(exrate,1),
                        start = 1942,
                        end = 2019)
```

```r
adl_gdp_exrate_coef <- lmtest::coeftest(x = adl_gdp_exrate, vcov.=sandwich)
adl_gdp_exrate_coef
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.48983    0.86524  2.8776 0.005270 **
## L(gdp, 1)      0.32306    0.10094  3.2005 0.002042 **
## L(gdp, 2)      0.22989    0.11087  2.0735 0.041707 *
## L(exrate, 1) -0.59143    0.34043 -1.7373 0.086612 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`(Intercept)` is the estimated coefficient for the intercept, `L(gdp, 1)` is the estimated for the first lag of the GDP, `L(gdp, 2)` the estimated coefficient for the second lag of the GDP and `L(exrate, 1)` for the first lag of the exchange rate. The following models will use the same naming convention.

Predict GDP growth in 2020 using model 1.

We can predict the GDP growth in 2020 simply by using $\hat{gdp}_{2020} = \hat{\alpha} + \hat{\delta}_1 gdp_{2019} + \hat{\delta}_2 gdp_{2018} + \hat{\delta}_3 exrate_{2019}$, where $\hat{\alpha}$ is '`(Intercept)`, $\hat{\delta}_1$ is `L(gdp, 1)`, $\hat{\delta}_2$ is `L(gdp, 2)` and $\hat{\delta}_3$ is `L(exrate, 1)`. Then:

```r
adl_gdp_exrate_forecast <- adl_gdp_exrate_coef[1,1] +
  adl_gdp_exrate_coef[2,1]*data_brazil$gdp[data_brazil$date==2019] +
  adl_gdp_exrate_coef[3,1]*data_brazil$gdp[data_brazil$date==2018] +
  adl_gdp_exrate_coef[4,1]*data_brazil$exrate[data_brazil$date==2019]

adl_gdp_exrate_forecast
```

```
## [1] 0.9613436
```

The $ADL(2,1)$ model predicted a `0.96%` growth for the GDP in 2020.

Model 2: Run an ADL(2,2) using GDP growth as your dependent variable and inflation as your predictor.

Report the estimated coefficient and their standard errors.

```r
adl_gdp_ipc <- dynlm(data = data_brazil_ts,
                           formula = gdp ~ L(gdp,1) + L(gdp,2) + L(ipc,1) + L(ipc,2),
                           start = 1942,
                           end = 2019)

adl_gdp_ipc_coef <- lmtest::coeftest(x = adl_gdp_ipc, vcov.=sandwich)
adl_gdp_ipc_coef
```

```
##
## t test of coefficients:
##
##                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept)  1.77736836  0.63995811  2.7773 0.007005 **
## L(gdp, 1)    0.36770394  0.10718933  3.4304 0.001008 **
## L(gdp, 2)    0.26443638  0.09803498  2.6974 0.008724 **
## L(ipc, 1)    0.00030256  0.00130581  0.2317 0.817434
```

```
## L(ipc, 2)   -0.00085232  0.00061697 -1.3815 0.171464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Predict GDP growth in 2020 using model 2.

```
adl_gdp_ipc_forecast <- adl_gdp_ipc_coef[1,1] +
  adl_gdp_ipc_coef[2,1]*data_brazil$gdp[data_brazil$date==2019] +
  adl_gdp_ipc_coef[3,1]*data_brazil$gdp[data_brazil$date==2018] +
  adl_gdp_ipc_coef[4,1]*data_brazil$ipc[data_brazil$date==2019] +
  adl_gdp_ipc_coef[5,1]*data_brazil$ipc[data_brazil$date==2018]

adl_gdp_ipc_forecast
```

```
## [1] 2.696677
```

The $ADL(2,2)$ model predicted a `2.7%` growth for the GDP in 2020.

Model 3: Run an general time series regression model using GDP growth as your dependent variable and two lags of GDP growth, Exchange Rate and Infation as your predictors.

Report the estimated coefficient and their standard errors.

```
adl_gdp_exrate_ipc <- dynlm(data = data_brazil_ts,
                            formula = gdp ~ L(gdp,1) + L(gdp,2) + L(exrate,1) + L(exrate,2) + L(ipc,1) + L(i
                            start = 1942,
                            end = 2019)

adl_gdp_exrate_ipc_coef <- lmtest::coeftest(x = adl_gdp_exrate_ipc, vcov.=sandwich)
adl_gdp_exrate_ipc_coef
```

```
##
## t test of coefficients:
##
##                  Estimate  Std. Error t value Pr(>|t|)
## (Intercept)    3.19991758  1.02696738  3.1159 0.002672 **
## L(gdp, 1)      0.28205296  0.11054765  2.5514 0.012949 *
## L(gdp, 2)      0.19536040  0.11156176  1.7511 0.084366 .
## L(exrate, 1) -1.49747998  1.16585380 -1.2844 0.203281
## L(exrate, 2)  0.75094685  1.17754703  0.6377 0.525766
## L(ipc, 1)    -0.00044145  0.00150625 -0.2931 0.770342
## L(ipc, 2)    -0.00086988  0.00086997 -0.9999 0.320855
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Predict GDP growth in 2020 using model 3.

```
adl_gdp_exrate_ipc_forecast <- adl_gdp_exrate_ipc_coef[1,1] +
  adl_gdp_exrate_ipc_coef[2,1]*data_brazil$gdp[data_brazil$date==2019] +
  adl_gdp_exrate_ipc_coef[3,1]*data_brazil$gdp[data_brazil$date==2018] +
  adl_gdp_exrate_ipc_coef[4,1]*data_brazil$exrate[data_brazil$date==2019] +
  adl_gdp_exrate_ipc_coef[5,1]*data_brazil$exrate[data_brazil$date==2018] +
  adl_gdp_exrate_ipc_coef[6,1]*data_brazil$ipc[data_brazil$date==2019] +
```

```r
  adl_gdp_exrate_ipc_coef[7,1]*data_brazil$ipc[data_brazil$date==2018]

  adl_gdp_exrate_ipc_forecast
```

```
## [1] 0.7249797
```

The GTS model predicted a `0.72%` growth for the GDP in 2020.

Model 4: Run an ARMA(2,0) using GDP growth as your dependent variable.

To run an ARMA(2,0), I use the `stats::arima` function, which is the canonical method for estimating ARIMA(p,d,q).

Report the estimated coefficient and their standard errors.

Note that the `arima` does not report the actual Intercept, but the mean of the stochastic process insead. We must calculate the intercept using the formula: $\alpha = \mu(1 - \delta_1 - \delta_2)$

```r
arma_gdp <- arima(x = data_brazil_ts[,"gdp"], order = c(2,0,0))

arma_gdp_coef <- coeftest(arma_gdp)

arma_gdp_coef
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value  Pr(>|z|)
## ar1        0.32626    0.11210  2.9104  0.003610 **
## ar2        0.29376    0.11280  2.6042  0.009209 **
## intercept  4.47135    1.00641  4.4429 8.877e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
arma_gdp_coef_intercept <- arma_gdp_coef[3]*(1-arma_gdp_coef[1]-arma_gdp_coef[2])

arma_gdp_coef_intercept
```

```
## [1] 1.699016
```

Predict GDP growth in 2020 using model 4.

```r
arma_gdp_forecast <- arma_gdp_coef_intercept +
  arma_gdp_coef[1,1]*data_brazil$gdp[data_brazil$date==2019] +
  arma_gdp_coef[1,2]*data_brazil$gdp[data_brazil$date==2018]

arma_gdp_forecast
```

```
## [1] 2.297263
```

The $ARMA(2,0)$ model predicted a `2.3%` growth for the GDP in 2020.

Which model generate the prediction that is closest to the realized value?

All models performed poorly. The realized GDP growth in 2020 was -3.88%, thus the General Time Series Model, which predicted a GDP growth of `0.72%`, generated the prediction that is closest to the realized value.

Question 6

Since I have a lot of stuff in my environment, which I find rather confusing, I will delete everything but `data_brazil` and `data_brazil_ts`.

```
rm(list=c(ls()[c(-14,-15)]))
```

6.1.

We will use the `vars::VAR` function to estimate the models. Since all models are the same but for the number of lags `p`, I will use a `for` loop to estimate all models at once. Therefore, in this section, I report all coefficients and standard errors of items 6a) to 6c) and also predict the GDP growth in 2020 using each of the three models.

I wasn't sure about including or not a time trend. Following the code used in class, I chose not to, so we have `type = "const"`.

```
for(p in 1:3){

  gdp_VAR <- VAR(y = data_brazil_ts, p = p, type = "const")

  cat(paste0("","\n"))
  cat(paste0("Results for the VAR(",p,") model:","\n"))
  stargazer::stargazer(gdp_VAR$varresult, type = "text")
  cat(paste0("","\n"))

  gdp_VAR_forecast <- predict(gdp_VAR, n.ahead = 1)
  cat(paste0("The GDP growth prediction using the VAR(",p,") model is: ",
             round(gdp_VAR_forecast[["fcst"]][["gdp"]][1],3),"%\n"))


}
```

```
##
## Results for the VAR(1) model:
##
## ==============================================================
##                                    Dependent variable:
##                                ------------------------------
##                                              y
##                                   (1)        (2)        (3)
## --------------------------------------------------------------
## gdp.l1                          0.292**    -0.001   -17.725*
##                                 (0.111)    (0.007)   (9.499)
##
## exrate.l1                      -1.164***  1.041***  -61.510*
##                                 (0.411)    (0.024)  (35.044)
##
## ipc.l1                          -0.002*    0.0001*  0.632***
##                                 (0.001)   (0.0001)   (0.087)
##
## const                          4.478***     0.011   182.530**
##                                 (0.874)    (0.052)  (74.463)
##
## --------------------------------------------------------------
## Observations                      77         77         77
```

```
## R2                             0.301     0.971      0.522
## Adjusted R2                    0.273     0.970      0.503
## Residual Std. Error (df = 73)   3.457     0.204     294.664
## F Statistic (df = 3; 73)      10.496*** 810.941*** 26.622***
## ===============================================================
## Note:                           *p<0.1; **p<0.05; ***p<0.01
##
## The GDP growth prediction using the VAR(1) model is: 0.234%
##
## Results for the VAR(2) model:
##
## ===============================================================
##                                     Dependent variable:
##                               ------------------------------
##                                              y
##                                 (1)       (2)        (3)
## ---------------------------------------------------------------
## gdp.l1                        0.282**    0.001     -18.671*
##                               (0.121)   (0.007)    (10.306)
##
## exrate.l1                      -1.497   1.334***   -194.682
##                               (2.051)   (0.121)   (175.155)
##
## ipc.l1                        -0.0004   0.0001*    0.707***
##                               (0.001)   (0.0001)   (0.120)
##
## gdp.l2                         0.195*   -0.0002    -13.218
##                               (0.117)   (0.007)    (9.982)
##
## exrate.l2                      0.751    -0.309**   112.603
##                               (2.122)   (0.125)   (181.280)
##
## ipc.l2                         -0.001   -0.00005    -0.173
##                               (0.001)   (0.0001)   (0.117)
##
## const                         3.200***   0.0001   288.045***
##                               (1.085)   (0.064)    (92.642)
##
## ---------------------------------------------------------------
## Observations                     76        76         76
## R2                             0.349     0.973      0.557
## Adjusted R2                    0.292     0.971      0.519
## Residual Std. Error (df = 69)  3.414     0.201     291.590
## F Statistic (df = 6; 69)      6.153*** 416.164*** 14.482***
## ===============================================================
## Note:                           *p<0.1; **p<0.05; ***p<0.01
##
## The GDP growth prediction using the VAR(2) model is: 0.725%
##
## Results for the VAR(3) model:
##
## ==============================================================
##                                     Dependent variable:
##                               -----------------------------
```

```
##                                              y
##                                  (1)       (2)        (3)
## -------------------------------------------------------------
## gdp.l1                         0.269**    0.003     -12.038
##                                (0.128)   (0.007)    (9.569)
##
## exrate.l1                       -1.557   1.342***   -97.002
##                                (2.181)   (0.127)   (162.545)
##
## ipc.l1                          -0.001    0.0001   0.800***
##                                (0.001)   (0.0001)   (0.111)
##
## gdp.l2                          0.235*    0.001     -15.233
##                                (0.131)   (0.008)    (9.755)
##
## exrate.l2                       0.941     -0.341   -199.978
##                                (3.544)   (0.206)   (264.062)
##
## ipc.l2                          -0.001   -0.00004  -0.568***
##                                (0.002)   (0.0001)   (0.136)
##
## gdp.l3                          -0.043    -0.012     1.963
##                                (0.123)   (0.007)    (9.176)
##
## exrate.l3                       -0.141    0.008     239.065
##                                (2.294)   (0.133)   (170.953)
##
## ipc.l3                         -0.0004   -0.00003  0.495***
##                                (0.001)   (0.0001)   (0.108)
##
## const                          3.269**    0.058    217.662**
##                                (1.292)   (0.075)    (96.271)
##
## -------------------------------------------------------------
## Observations                      75        75        75
## R2                              0.352     0.974      0.667
## Adjusted R2                     0.262     0.970      0.621
## Residual Std. Error (df = 65)   3.496     0.203     260.490
## F Statistic (df = 9; 65)      3.926*** 270.968*** 14.460***
## =============================================================
## Note:                            *p<0.1; **p<0.05; ***p<0.01
##
## The GDP growth prediction using the VAR(3) model is: 0.799%
```

Which model generate the prediction that is closest to the realized value?

The $VAR(1)$ model, which predicted a growth of `0.234%`.

6.2.

Please note that item d) was compiled after the plots.

Choose the order of your variables and justify your exclusion restrictions

The exchange rate should be the last variable: it is very a very volatile variable that quickly responds to unexpected shocks. The inflation rate should be the first, most exogenous variable: since prices are sticky (because of, say, menu costs), they don't react quickly to shocks in the exchange rate or the domestic GDP. Finally, GDP should be the second variable. Let's properly order the dataframe, create the `ts` object and estimate our $VAR(2)$ model:

```
var2 <- data_brazil %>%
  dplyr::select(ipc, gdp, exrate) %>%
  ts(start = 1942, end = 2019, frequency = 1) %>%
  VAR(p = 2, type = "both")
```

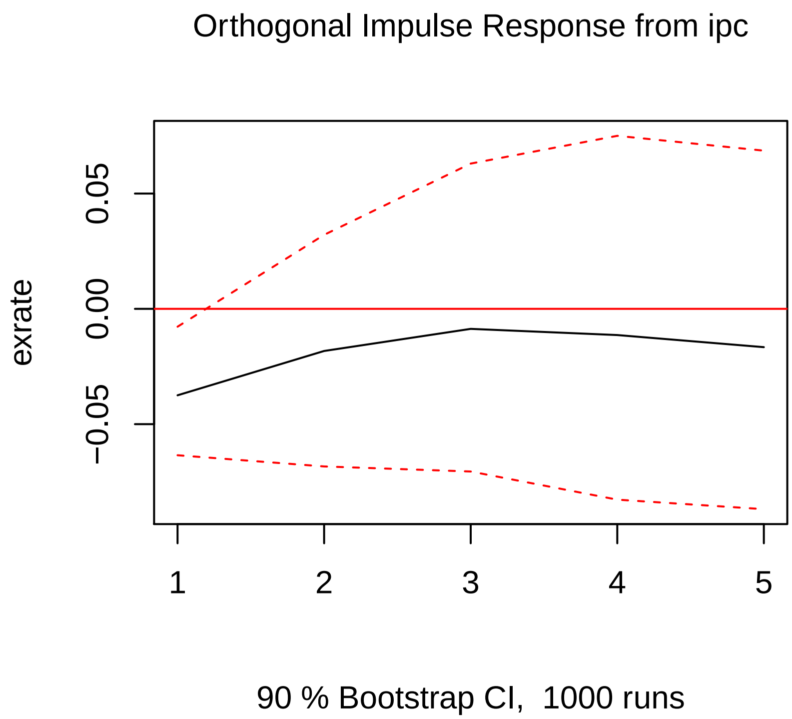
Estimate and plot all nine structural impulse response functions and their 90%-confdence intervals based on 1,000 bootstrap repetitions.

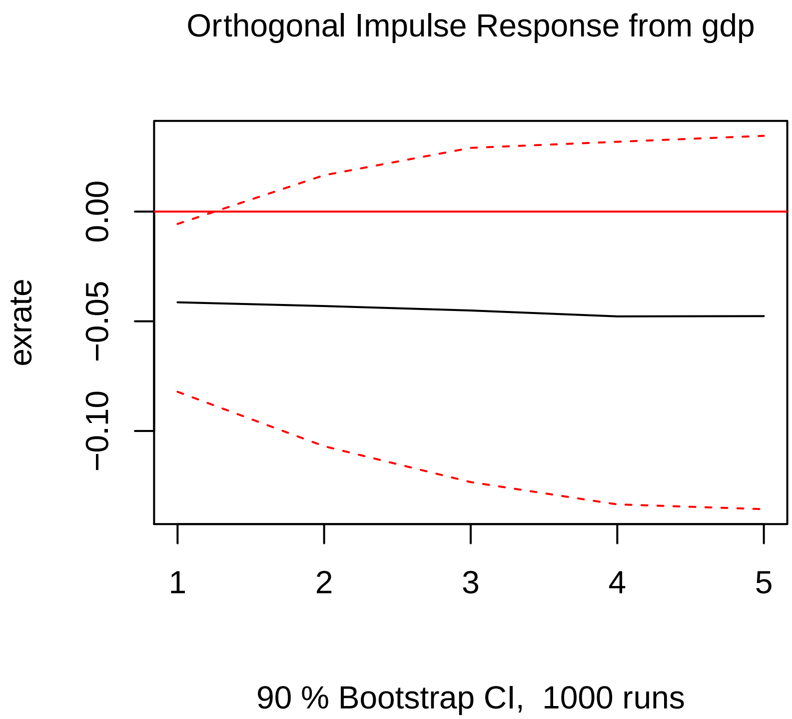
Since the number of periods ahead was not specified, I have decided to use `n.ahead = 4`, because in `code01-var-example.R`, Possebom plots the IRF for 16 quarters. Note that we have yearly data here.

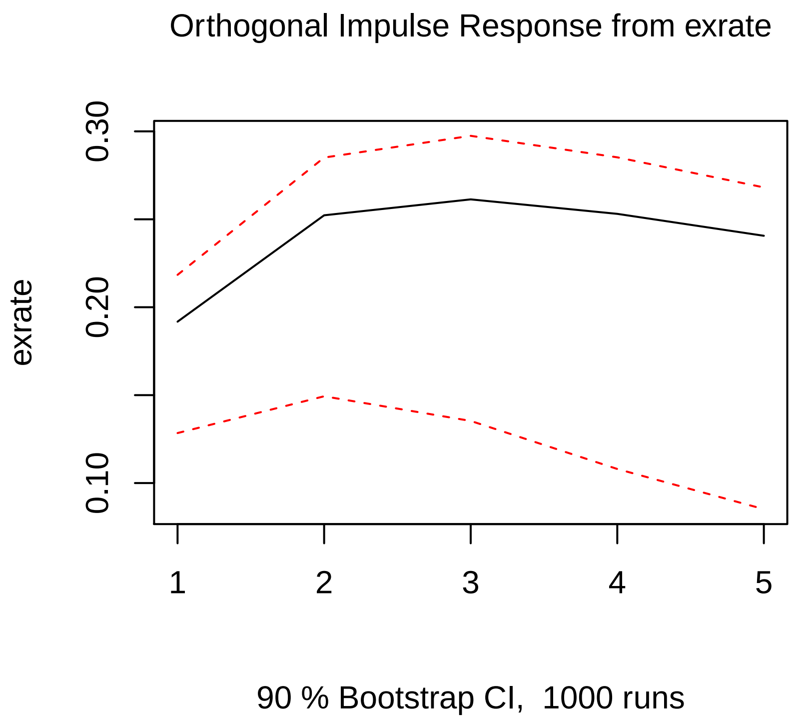
I create three sets of three plots each. We could plot only three figures, but this would mess with the y-scale.

```
structural_irf_exrate <- irf(x = var2,
                      n.ahead = 4,
                      ortho = T,
                      boot = T,
                      runs = 1000,
                      ci = .9,
                      response = "exrate")

plot(structural_irf_exrate)
```

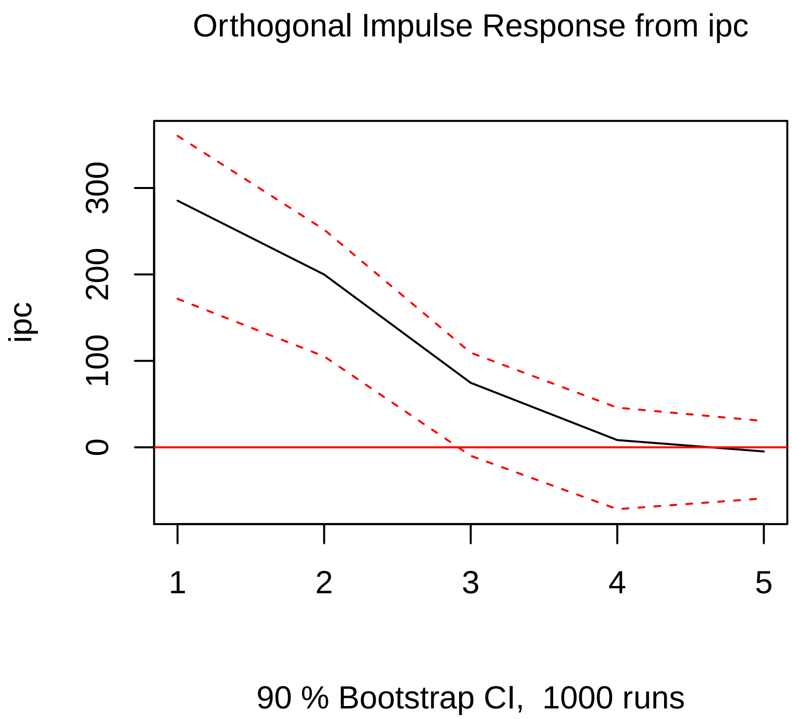

Orthogonal Impulse Response from ipc
exrate
0.05
0.00
−0.05
1
2
3
4
5
90 % Bootstrap CI, 1000 runs

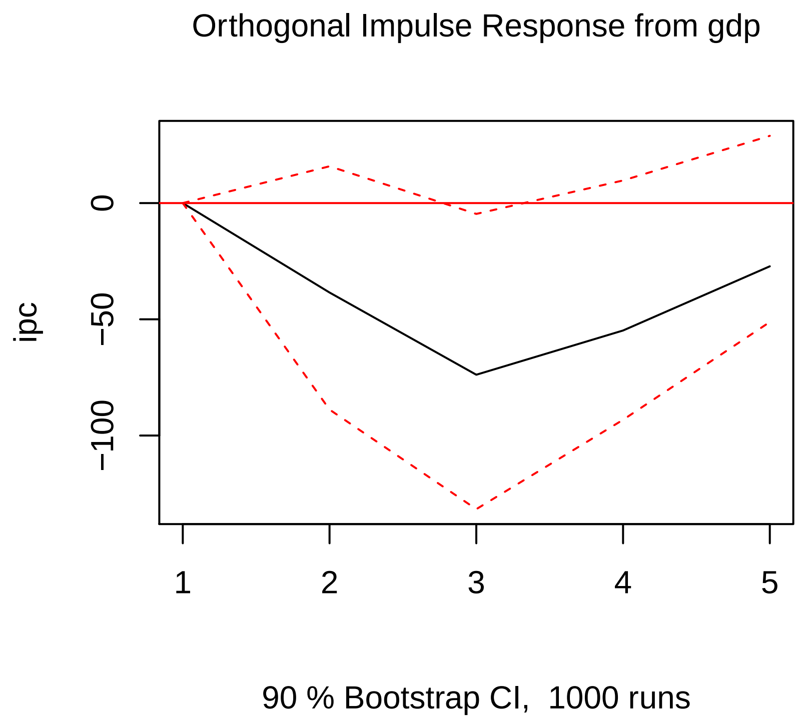

Orthogonal Impulse Response from gdp
exrate
0.00
−0.05
−0.10
1
2
3
4
5
90 % Bootstrap CI, 1000 runs

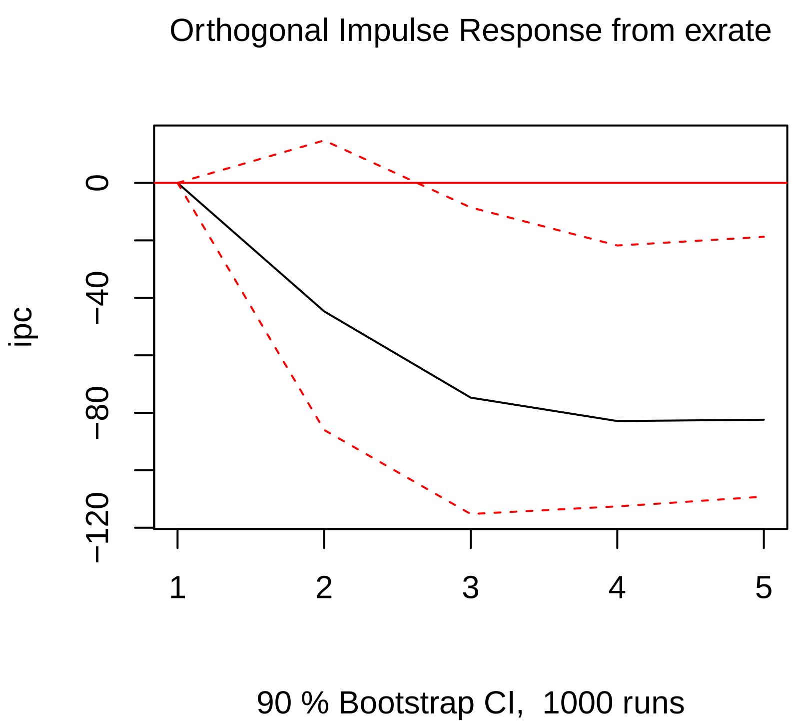



```r
structural_irf_ipc <- irf(x = var2,
                          n.ahead = 4,
                          ortho = T,
                          boot = T,
                          runs = 1000,
                          ci = .9,
                          response = "ipc")

plot(structural_irf_ipc)
```

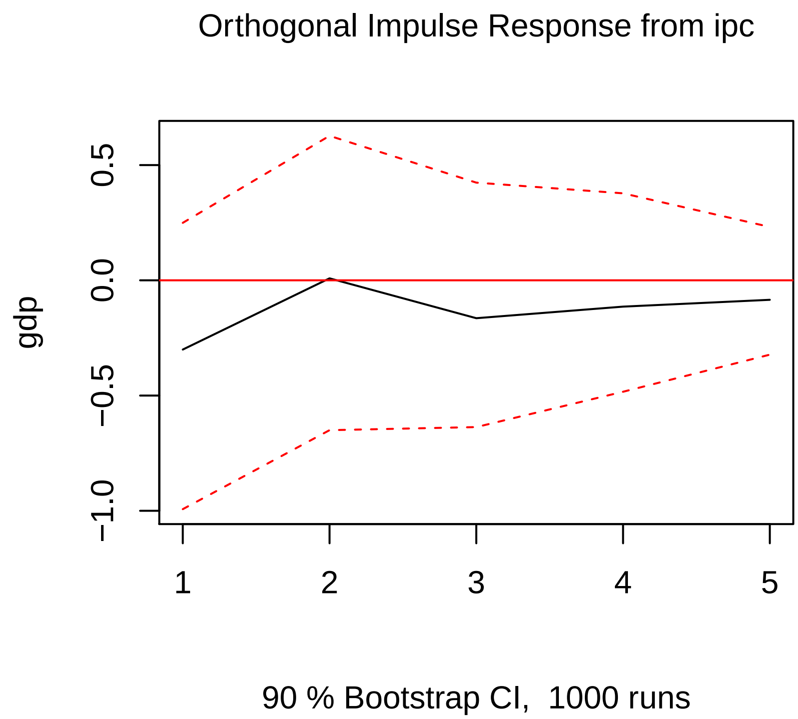

Orthogonal Impulse Response from ipc
ipc
300
200
100
0
1
2
3
4
5
90 % Bootstrap CI, 1000 runs

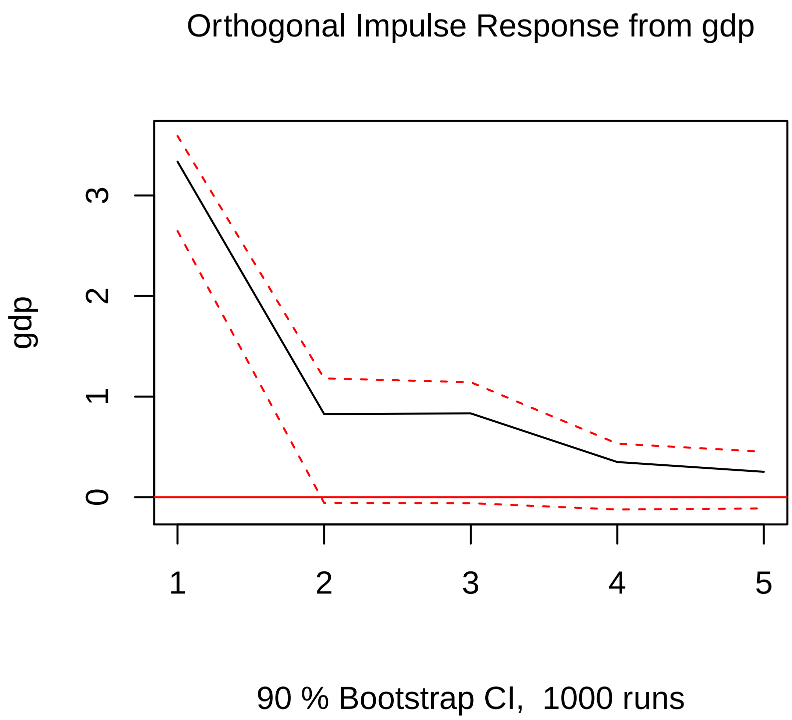

Orthogonal Impulse Response from gdp
ipc
0
−50
−100
1
2
3
4
5
90 % Bootstrap CI, 1000 runs



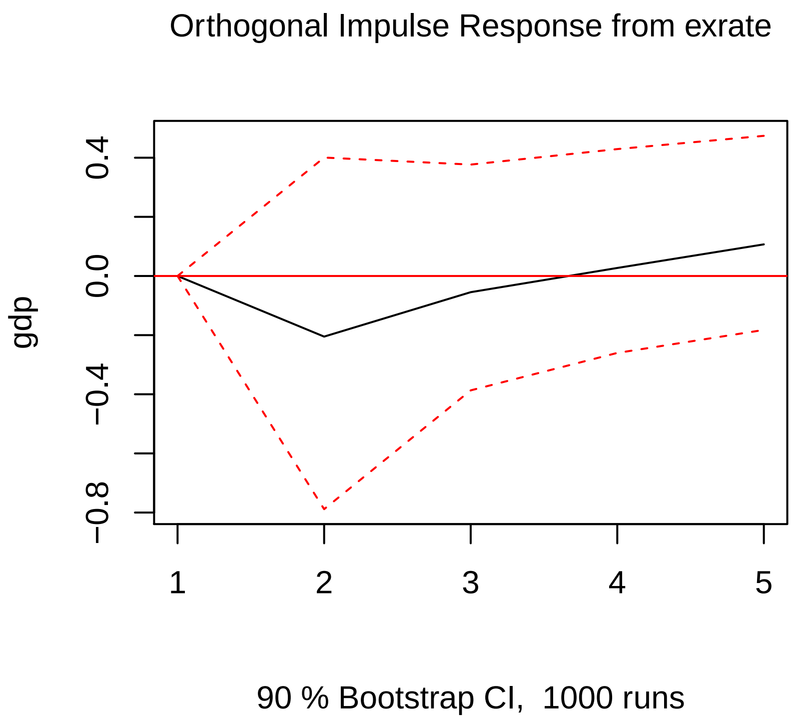

```
structural_irf_gdp <- irf(x = var2,
                          n.ahead = 4,
                          ortho = T,
                          boot = T,
                          runs = 1000,
                          ci = .9,
                          response = "gdp")

plot(structural_irf_gdp)
```


Orthogonal Impulse Response from ipc
gdp
0.5
0.0
−0.5
−1.0
1
2
3
4
5
90 % Bootstrap CI, 1000 runs



Orthogonal Impulse Response from gdp
gdp
0
1
2
3
1
2
3
4
5
90 % Bootstrap CI, 1000 runs



Do you believe your results are credible? Justify your answer.

I don't think these results are credible. From 1942 to 2019, there were several structural breaks and meaningful institutinal and economical changes in the Brazilian economy, such as periods of hyperinflation, the latin-american debt crisis in the early 1980s, the successive currency reforms and the mid 2010s recession. Without accounting for these changes, it's is very implausible to assume that the impulse response functions are actually reflecting a accurate effect of a shock in a certain variable into another.