4423 Advanced Macroeconometrics 1 - Assignment 1

Unterweger L., Pirich G., Tijani S.

2023-04-19

Excercise 1: FRED-MD

Download the current version of the FRED-MD database and load it into R (or another statistical software of your choice). Note that the second line in the CSV-file denotes the suggested transformation, you have to remove it.

```
##    sasdate      RPI W875RX1 DPCERA3M086SBEA CMRMTSPLx  RETAILx
## 2 1/1/1959 2442.158  2293.2          17.272  292266.4 18235.77
## 3 2/1/1959 2451.778  2301.5          17.452  294424.7 18369.56
## 4 3/1/1959 2467.594  2318.5          17.617  293418.7 18523.06
## 5 4/1/1959 2483.671  2334.9          17.553  299322.8 18534.47
## 6 5/1/1959 2498.026  2350.4          17.765  301364.3 18679.66
## 7 6/1/1959 2505.788  2357.4          17.831  301348.8 18849.75
```

Subquestion (a)

Create a function that takes a vector containing observations of a time series as input and returns a dataframe with the following transformed series in its columns as output: - the original time series in its raw form; -the log-transformed time series; - month-on-month growth rates in percent; - year-on-year growth rates in percent; - the first lag of the year-on-year growth rates of the time series

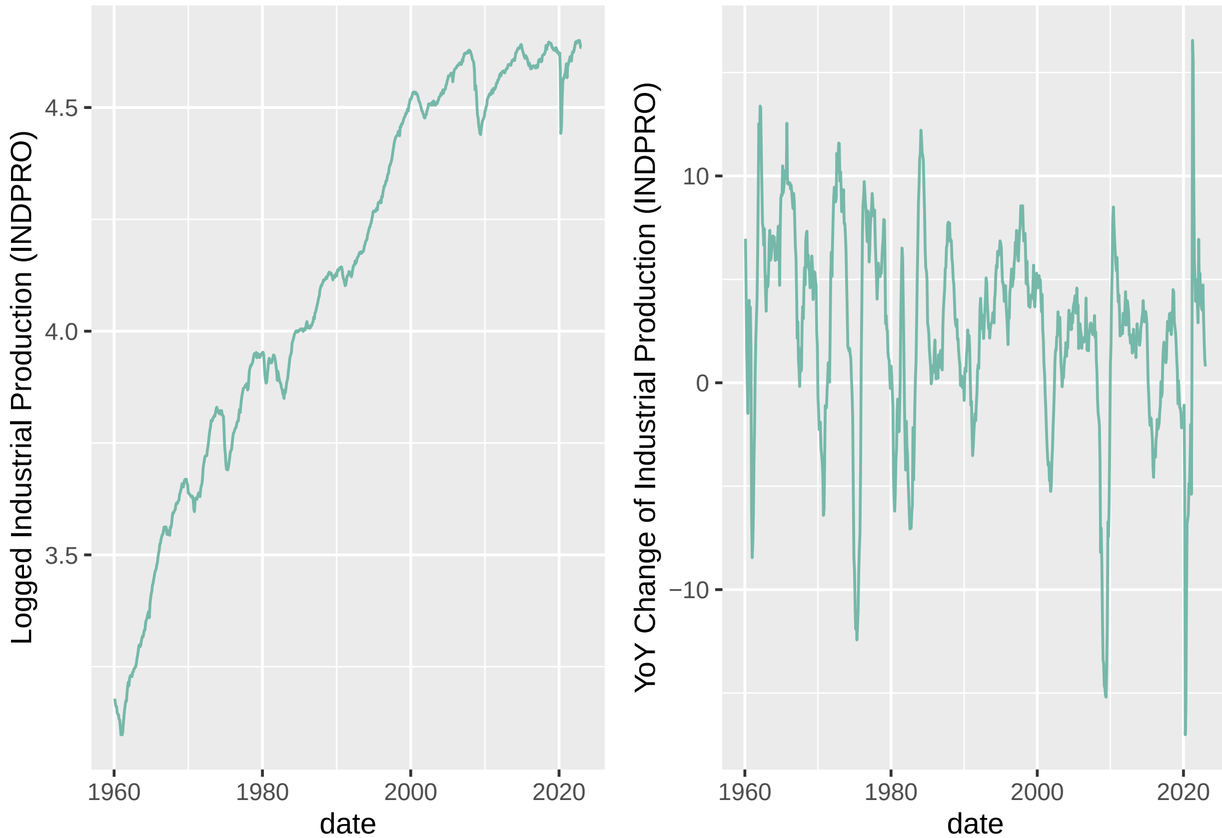
```r
ts_transform <- function(x) {
  output <- data.frame(matrix(NA,    # Create empty data frame
                              nrow = NROW(x),
                              ncol = 0))
  output$raw <- x
  output$log <- log(x)
  output$mom <- ((x-lag(x))/lag(x))*100
  output$yoy <- ((x-lag(x, 12))/lag(x, 12))*100
  output$yoy_1stlag <- lag((x-lag(x, 12))/lag(x, 12))*100
  return(output)
}
```

Subquestion (b)

Use the created function to create a dataframe with the various transformation for US industrial production (mnemonic INDPRO), plot the logged time series and the yearly changes produced by the function. Briefly describe the properties of the time series.

```
##         date     raw      log       mom yoy yoy_1stlag
## 1 1959-01-01 22.0151 3.091729        NA  NA         NA
## 2 1959-02-01 22.4463 3.111126 1.9586556  NA         NA
## 3 1959-03-01 22.7696 3.125426 1.4403265  NA         NA
## 4 1959-04-01 23.2547 3.146507 2.1304722  NA         NA
```

```
## 5 1959-05-01 23.6050 3.161459 1.5063622  NA          NA
## 6 1959-06-01 23.6319 3.162597 0.1139589  NA          NA
```

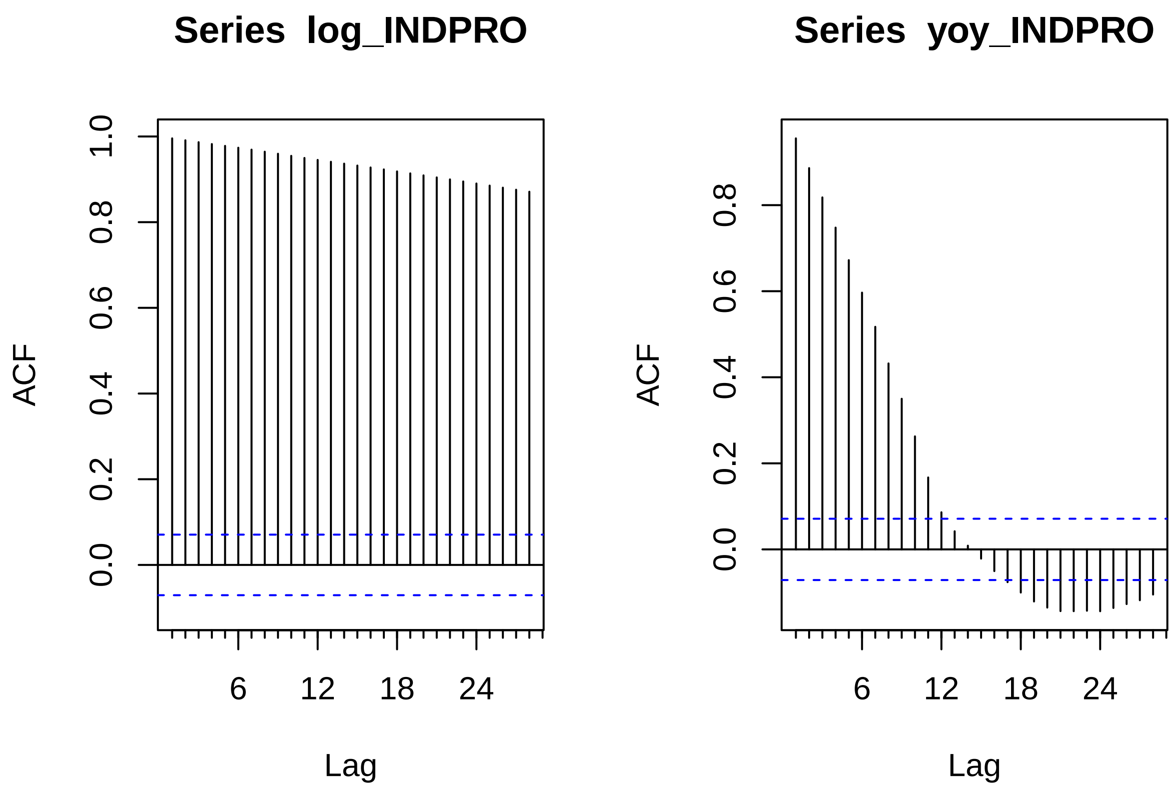


The series of the logged production follows an upward trend showing the increase in the data over time. There are no cyclical or seasonal components. For instance, there are only a few observable downward shifts in the series as the subprime crisis around 2007, and the covid crisis around 2020. The time series does not seem to possess constant first and second moments. Finally, the time series show a slight curvature indicating a decreasing growth in the data over time. By analyzing the graph, the time series does not appear to be stationary.

The time series of the yearly growth rate is experiencing significant shifts over time, and although the variance is significant it seems stable. Besides, no upward, or downward trend can be observed, i.e., the series seems to evolve around a constant mean. It also seems that negative yearly growth rates are followed by negative growth rates (the same can be said for positive yearly growth rates), the data appears to be cylclical. The times series looks graphically stationary.

Subquestion (c)

Using suitable functions from the stats and urca package, assess the properties of both logged industrial production and its yearly growth rate. Plot the autocorrelation function and perform Dickey-Fuller tests to test for a unit root (note the different specifications, i.e. including a drift or a trend), interpret the results.



The ACF plot of the logged production shows a substantial autocorrelation over the entire period which only decays at a very slow-paced overtime. Thus, as in the analysis of the plotted time series, there is strong evidence against the stationarity.

The ACF plot of the yearly growth rate of industrial production has a substantial positive autocorrelation until lag 12 which decreased rapidly. The autocorrelation between lags 12 and 18 is not substantial. Finally, there is an alternation with the autocorrelation which is negative and slightly above the substantial level from lag 18. Thus, the time series looks stationary.

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.3097  -0.5880   0.0529   0.7056  15.5830
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## z.lag.1 -0.035909   0.009356  -3.838 0.000134 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.405 on 755 degrees of freedom
## Multiple R-squared:  0.01914,    Adjusted R-squared:  0.01784
## F-statistic: 14.73 on 1 and 755 DF,  p-value: 0.0001344
##
##
## Value of test-statistic is: -3.838
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.143587 -0.003563  0.000648  0.004666  0.058851
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## z.lag.1 4.514e-04  8.747e-05   5.161 3.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01002 on 767 degrees of freedom
## Multiple R-squared:  0.03356,    Adjusted R-squared:  0.0323
## F-statistic: 26.64 on 1 and 767 DF,  p-value: 3.132e-07
##
##
## Value of test-statistic is: 5.161
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

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

```
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.142265 -0.003909  0.000693  0.004596  0.059784
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0128789  0.0032689   3.940  8.9e-05 ***
## z.lag.1     -0.0026467  0.0007911  -3.345 0.000861 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009922 on 766 degrees of freedom
## Multiple R-squared:  0.0144, Adjusted R-squared:  0.01311
## F-statistic: 11.19 on 1 and 766 DF,  p-value: 0.0008614
##
##
## Value of test-statistic is: -3.3455 21.3309
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78

##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.142730 -0.003882  0.000696  0.004660  0.059091
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.870e-02  1.047e-02   1.786   0.0746 .
## z.lag.1     -4.412e-03  3.120e-03  -1.414   0.1578
## tt           3.726e-06  6.370e-06   0.585   0.5587
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009927 on 765 degrees of freedom
## Multiple R-squared:  0.01484,	Adjusted R-squared:  0.01227
## F-statistic: 5.762 on 2 and 765 DF,  p-value: 0.003282
##
##
## Value of test-statistic is: -1.4141 14.3224 5.7624
```

```
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34
```

Yoy_prod: The test statistic value is smaller than the critical value (-3.838<-2.58), therefore we can reject the null hypothesis, the time series has no unit root and conclude that the time series is stationary.

Log_prod:\ **Test without drift nor trend**: We cannot reject that the times series has a unit root as 5.161>-2.58. Thus, we cannot find evidence for stationarity.

Test with drift and no trend: The first null hypothesis cannot be rejected as -3.34>-3.43, there is a unit root. The second null hypothesis cannot be rejected either given that 21.33>6.43, thus there is a unit root and no drift. No evidence for stationarity.

Test with both drift and trend: The first null hypothesis (tau3) is not rejected as -1.41>-3.96, a unit root is present. The second null hypothesis (phi3) is rejected given that 5.76<6.09, there is a unit root and there is a trend. The third null hypothesis (phi2) is not rejected 14.32>6.09, a unit root is present and there is no trend nor drift. This is inconsistent with the second null hypothesis.

Subquestion (d)

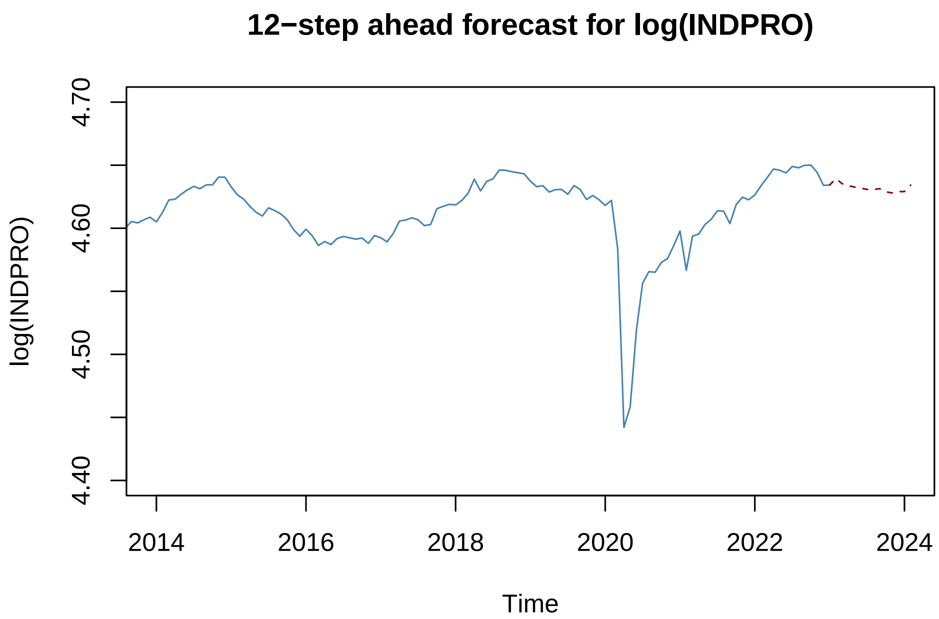
Estimate a suitable AR model (e.g. using the ar.ols() function) for the stationary time series (as determined in the previous point). How is the lag order determined by default? Use the estimated model to produceforecasts for the next year and plot them. Interpret their behaviour (i.e. are they converging towards a certain value? What could that be?). Use the produced forecasts to also forecast the change in the original time series.

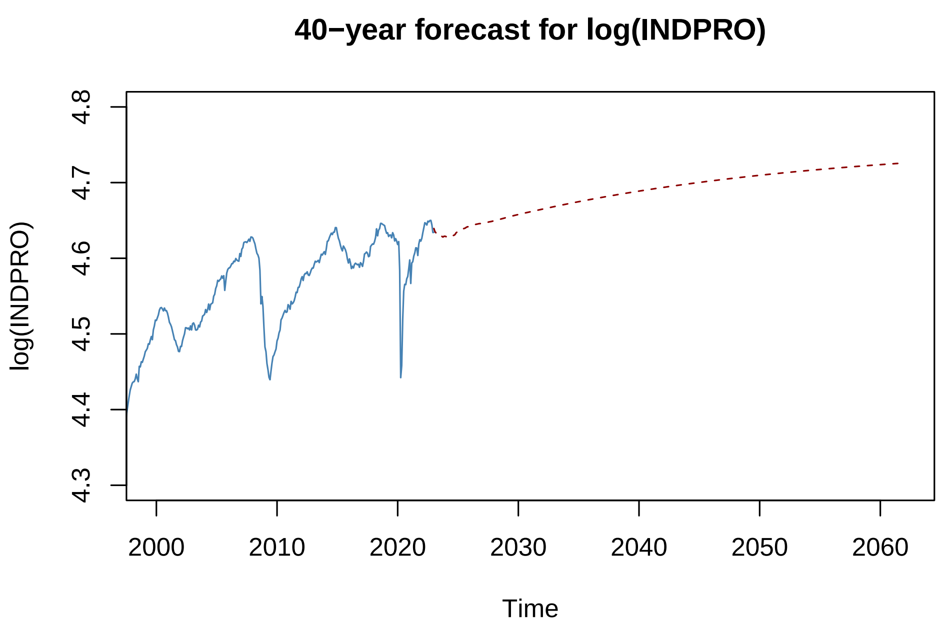
```
##
## Call:
## ar.ols(x = industrial_prod$log, na.action = na.omit, demean = TRUE,     intercept = TRUE)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
##  1.2637  -0.3294   0.0921   0.0518  -0.0941   0.0529  -0.0165  -0.0789
##       9       10       11       12       13       14       15       16
##  0.0961   0.0492  -0.1650   0.0850  -0.0609   0.0292   0.0281  -0.0108
##      17       18       19       20       21       22       23       24
##  0.0137   0.0036  -0.0460   0.0232  -0.0230   0.0326   0.0277  -0.1539
##      25
##  0.1268
##
## Intercept: 0.001837 (0.0004059)
##
## Order selected 25  sigma^2 estimated as  7.871e-05
```

```
##
## Call:
## ar.ols(x = industrial_prod$yoy, na.action = na.omit, demean = FALSE,     intercept = FALSE)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
##  1.2925  -0.3567   0.0895   0.0171  -0.0735   0.0812  -0.0743  -0.0481
##       9       10       11       12       13       14       15       16
##  0.1293   0.0448  -0.1444  -0.6125   0.8058  -0.2258   0.1347  -0.0914
##      17       18       19       20       21       22       23       24
```

```
##  0.0290   0.0253  -0.1213   0.0522   0.0187   0.1260  -0.0878  -0.5378
##      25       26
##  0.6397  -0.1440
##
## Order selected 26  sigma^2 estimated as  1.042
```

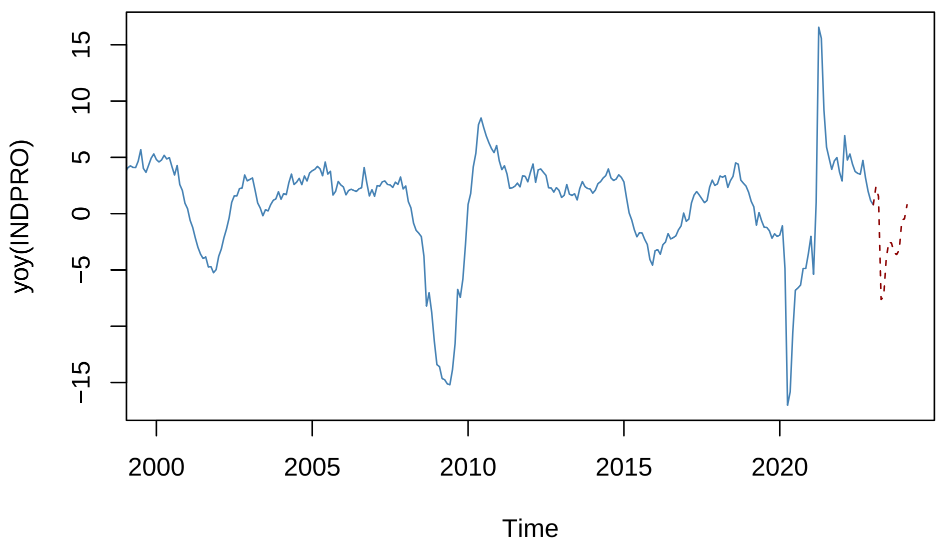
By default the lag order is determined by the AIC information criterion. For the log of INDPRO a lag order of 25 is estimated. The model for the year on year changes is estimated with lag order 26.



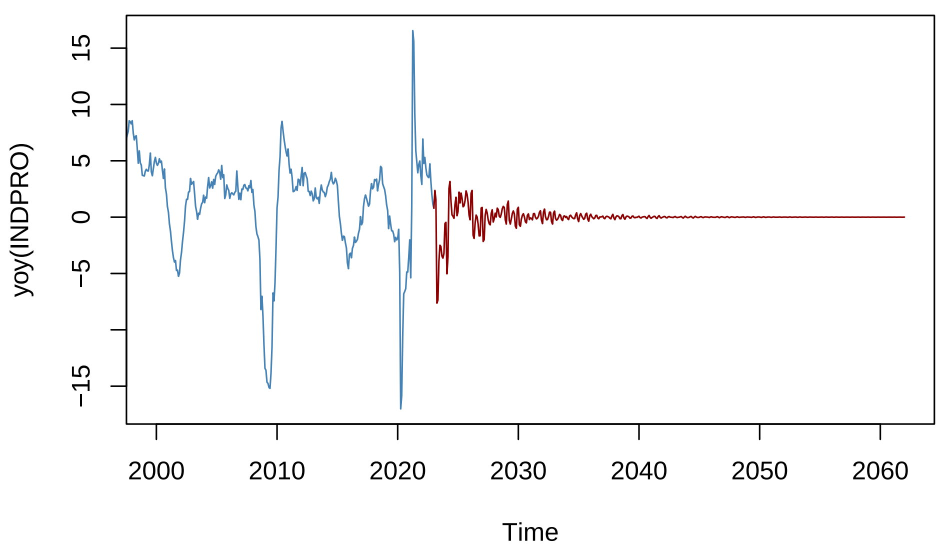


The predicted forecast seems to be stable with no significant shift occurring. When extending the forecasting period, the data is slightly upward trending and we can observe a divergence effect. It indicates that the value of the log production is expected to increase after every period.

12−step ahead forecast for yoy(INDPRO)

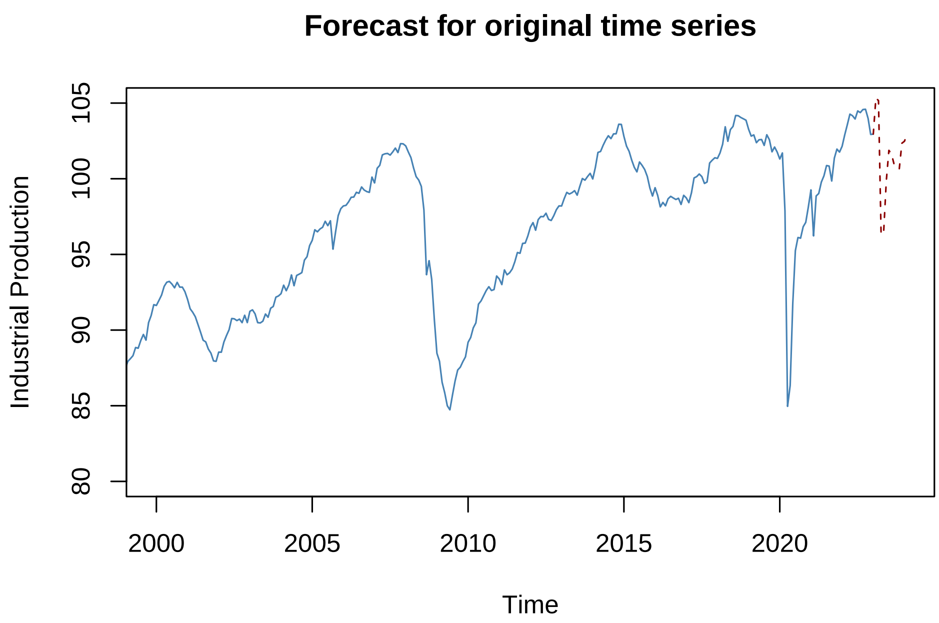


Forecast for yoy(INDPRO)



The forecast of the yearly growth rate shows predicted variation in the short term, however less significant than in the anterior periods. Also, in the long run, the yearly growth rate exhibits a convergence to 0.

```
##  [1] 105.33552 105.16440  96.31155  96.49389  99.83880 101.87643 101.63979
##  [8] 100.95871 100.78447 100.64529 102.34614 102.47634
```



Subquestion (e) - Bonus question

We start by defining the function.

```
rmse_ar <- function(data, lag, hold_period) {
  #Remove holdout period from the end of the sample
  data_train <- data[1:(length(data) - hold_period)]

  # Estimate the AR model
  ar_model <- ar.ols(data_train, order.max = lag, demean = TRUE, intercept = TRUE)

  # Forecast for the holdout period
  data_test <- data[(length(data) - hold_period + 1):length(data)]
  ar_forecasts <- predict(ar_model, n.ahead = hold_period)

  # Compute the RMSE
  rmse <- sqrt(mean((data_test - ar_forecasts$pred)^2))
  return(rmse)
}


#RMSE for 50 different lag orders and returning the minimal
rmse <- list()
for (x in 1:50) {
  rmse[[x]] <- rmse_ar(na.omit(industrial_prod$yoy), x, 6)
}
which.min((rmse))
```

```
## [1] 16
```

```
#RMSE for 50 different lag orders and returning the minimal
for (x in 1:50) {
  rmse[[x]] <- rmse_ar(na.omit(industrial_prod$yoy), x, 12)
```

```
}
which.min((rmse))
```

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

```
#RMSE for 50 different lag orders and returning the minimal
rmse <- list()
for (x in 1:50) {
  rmse[[x]] <- rmse_ar(na.omit(industrial_prod$log), x, 6)
}
which.min((rmse))
```

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

```
#RMSE for 50 different lag orders and returning the minimal
for (x in 1:50) {
  rmse[[x]] <- rmse_ar(na.omit(industrial_prod$log), x, 12)
}
which.min((rmse))
```

```
## [1] 11
```

We run the function for the two time series and compare the RMSE for AR models up to order 50. For a holdout period of 6 months we find that the year on year growth rates are best predicted with an AR(16) model. For a holdout period of 12 months, an AR(1) model produces the lowest RMSE.

For the log(industrial production) the lowest RMSE is produced by an AR(1) model for a 6 month holdout period. Over 12 months an AR(11) model serves as the best predictor.

Remarkably, based on the forecast performance, the optimal lag order is significantly lower than the lag order chosen by the AIC or BIC criterion. The AIC would have selected a model of order 25 (for the logged time series) and 26 (for the year on year growth rates).

Excercise 2 - Killian and Park (2009)

Read Kilian & Park (2009), who discuss the effects of oil price shocks on the US stock market, focus on Sections 2 and 3.1-3.3. Load the provided data by Kilian & Park (2009), which contains a measure of change in oil production, a measure of real economic activity, the real price of oil, and changes in real US dividend growth from 1973M1 to 2016M12.

```
##        GOPC     GRA      RPO    USSR
## 1  11.8773 34.5887 -46.3143 -1.3498
## 2   1.4191 40.0667 -46.6013 -0.3862
## 3   1.1777 42.5462 -45.3973  1.2771
## 4  27.4551 46.6761 -42.1724 -2.4366
## 5 -13.1104 50.6190 -39.8859 -0.2239
## 6  36.2581 51.5436 -39.3027  0.6786
```

Here, *GOPC*, *GRA*, *RPO* and *USSR* refer to *Global Oil Production Change*, *Global Real Activity*, *Real Price of Oil* and *U.S. Stock Returns* respectively. ## Subquestion (a) *Using the the packages vars in R (or an equivalent one in another language), estimate the VAR described in section 2.2 using the variables in the same order as specified by Kilian and Park (2009).*

```
mod1 <- VAR(data, p=24, type="const")
```

Due to the enormity of the VAR consisting of 24 lags and 4 time series, we decided to exclude the output at this point of the assignment.

Subquestion (b)

Using the estimated VAR, compute impulse response functions (take a look at the irf() function in vars, it uses the same identification scheme as Kilian & Park (2009) propose (recursive ordering based on a Cholesky decomposition of the vcov-matrix of the errors) by default. Replicate Figure 1 and the lower panel in figure 3 of Kilian & Park (2009). Interpret the results.

Figure 1

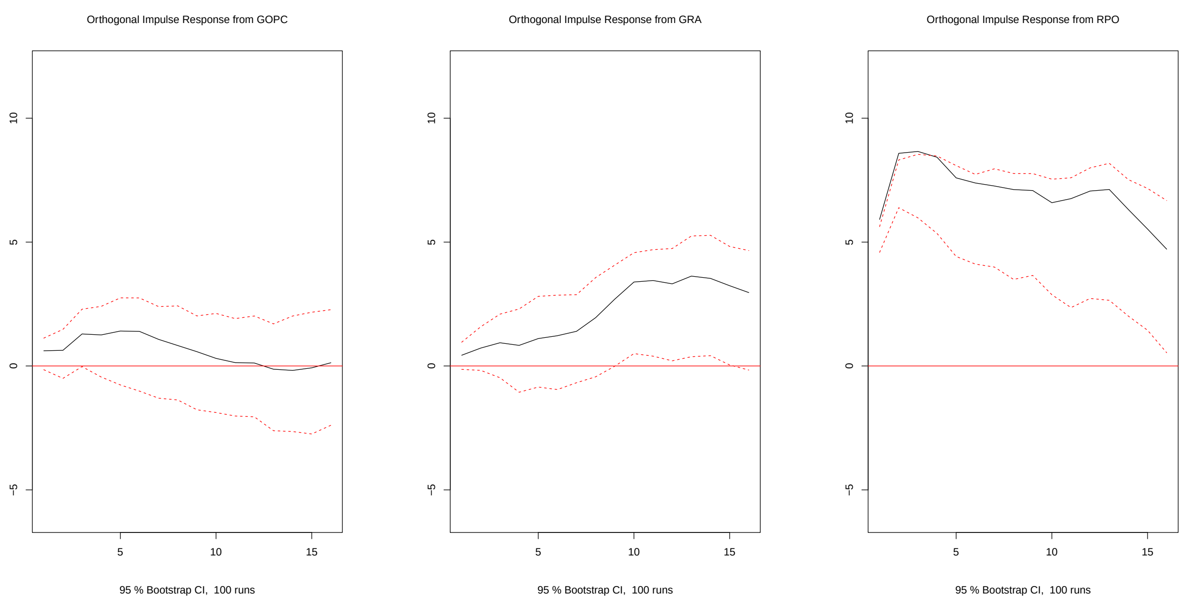
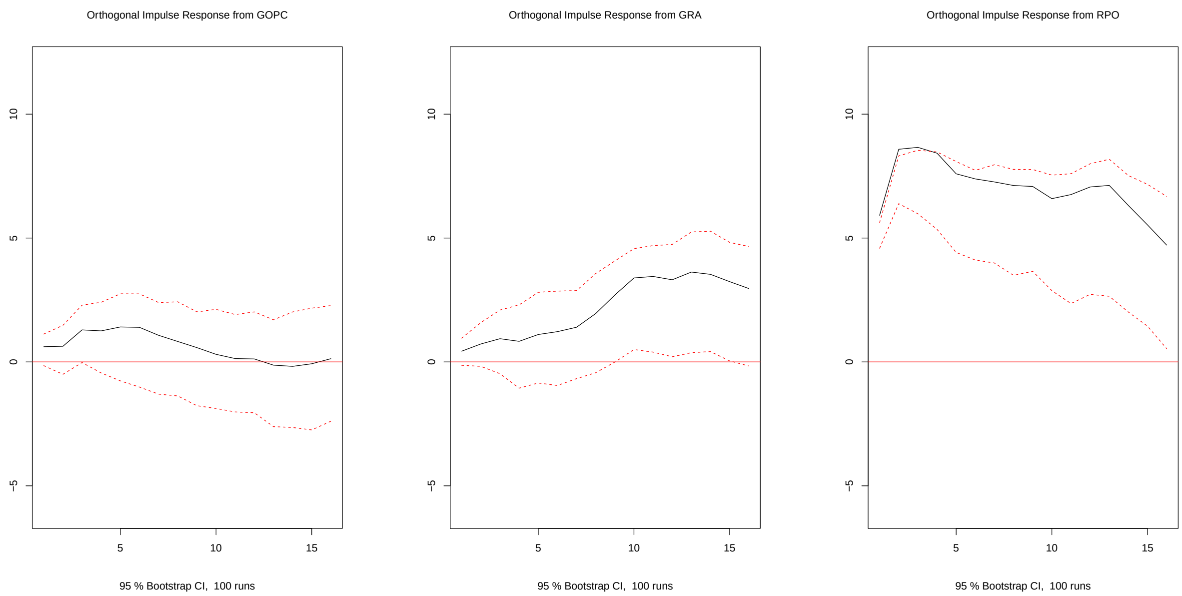


Figure 3



INTERPRETATION!

Subquestion (c)

Calculate forecast error variance decompositions for the included variables (take a look at the fevd() function in vars). Replicate Table 2 of Kilian and Park (2009). Interpret the results.

```
##        GOPC  GRA  RPO  USSR
## h=1    0.20 0.16 1.69 97.94
## h=2    0.55 0.36 2.09 97.00
## h=3    0.76 0.48 2.12 96.64
## h=12   2.80 6.83 4.53 85.84
## h=Inf  6.63 8.38 7.93 77.06
```

INTERPRETATION!

Subquestion (d)

Note that the dataset provided misses US stock market returns (due to the licensing of the underlying time series). Look for alternative data on the US stock market, create a variable similar to the one used by Kilian & Park (2009). Re-estimate the model and replicate Figure 1 again as well as the top panel of Figure 3 and Table 1.5 Interpret the results.

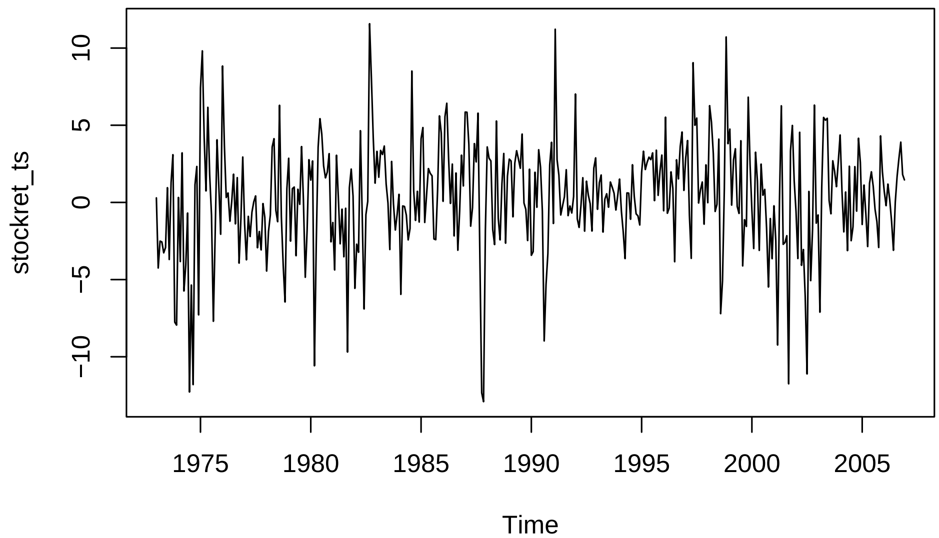
In this exercise we decided to use the S&P 500 returns as well as the CPI data from the FRED database to create monthly real stock returns.

```
## 'data.frame':    770 obs. of  3 variables:
##  $ sasdate : chr  "1/1/1959" "2/1/1959" "3/1/1959" "4/1/1959" ...
##  $ S.P.500 : num  55.6 54.8 56.2 57.1 58 ...
##  $ CPIAUCSL: num  29 29 29 29 29 ...
```

```
##  [1]         NA -1.4937564  2.6413340  1.6392707  1.2990903 -1.1037107
##  [7]  3.8305679 -0.6720489 -4.1961193 -0.4295228
```

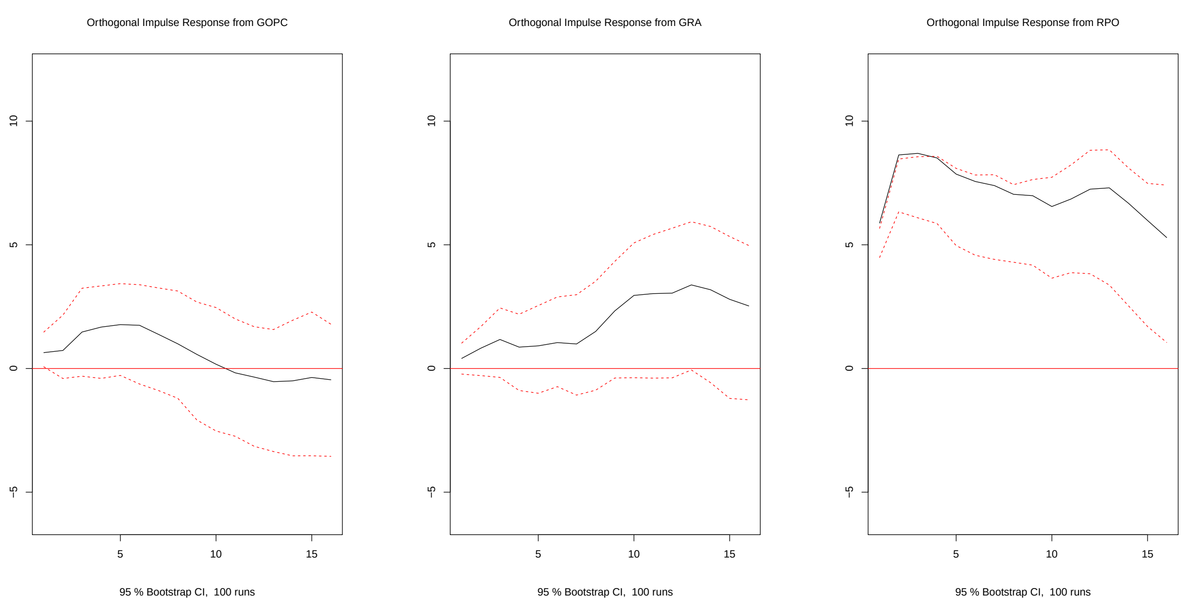
```
##             Jan        Feb        Mar        Apr        May        Jun
```

```
## 1959        NA -1.493756  2.641334  1.639271  1.299090 -1.103711
```



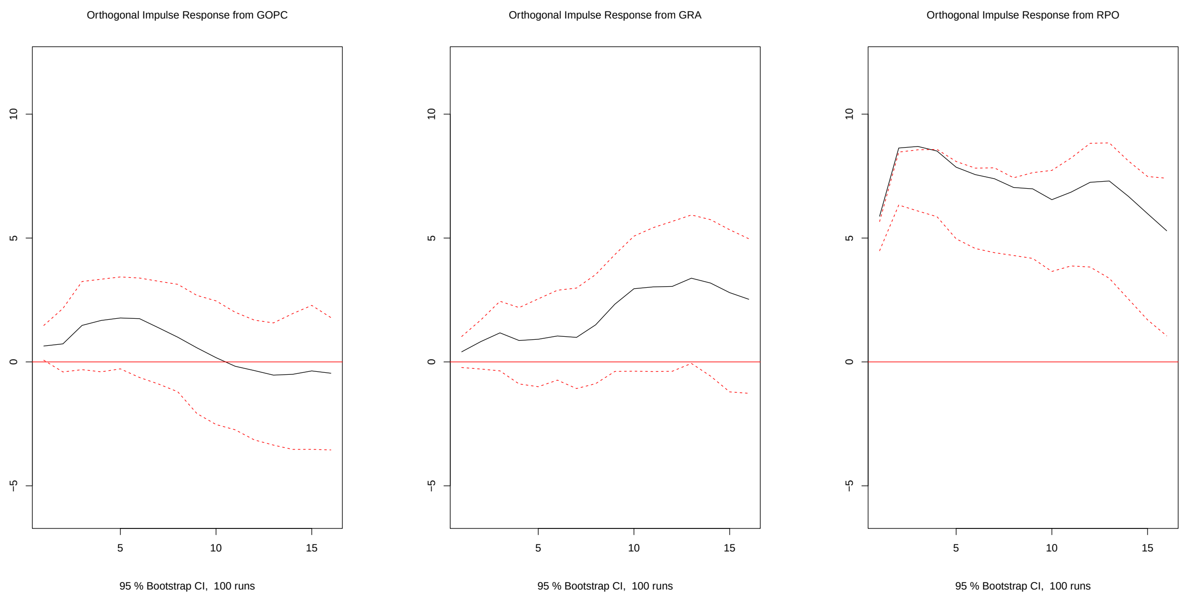
Using this data, we replicate the VAR given above. Again, the estimated coefficients are given in the appendix.

Figure 1



INTERPRETATION

Top Panel of Figure 3



INTERPRETATION

Table 1

Similar to before, we replicate table 1 using our data on S&P500 returns.

```
##        GOPC  GRA  RPO SP500 Returns
## h=1    0.38 0.01 2.64         96.97
## h=2    0.43 0.25 6.34         92.97
## h=3    0.89 0.48 6.68         91.96
## h=12   1.78 2.35 7.42         88.45
## h=Inf  5.84 6.17 9.04         78.96
```

INTERPRETATION