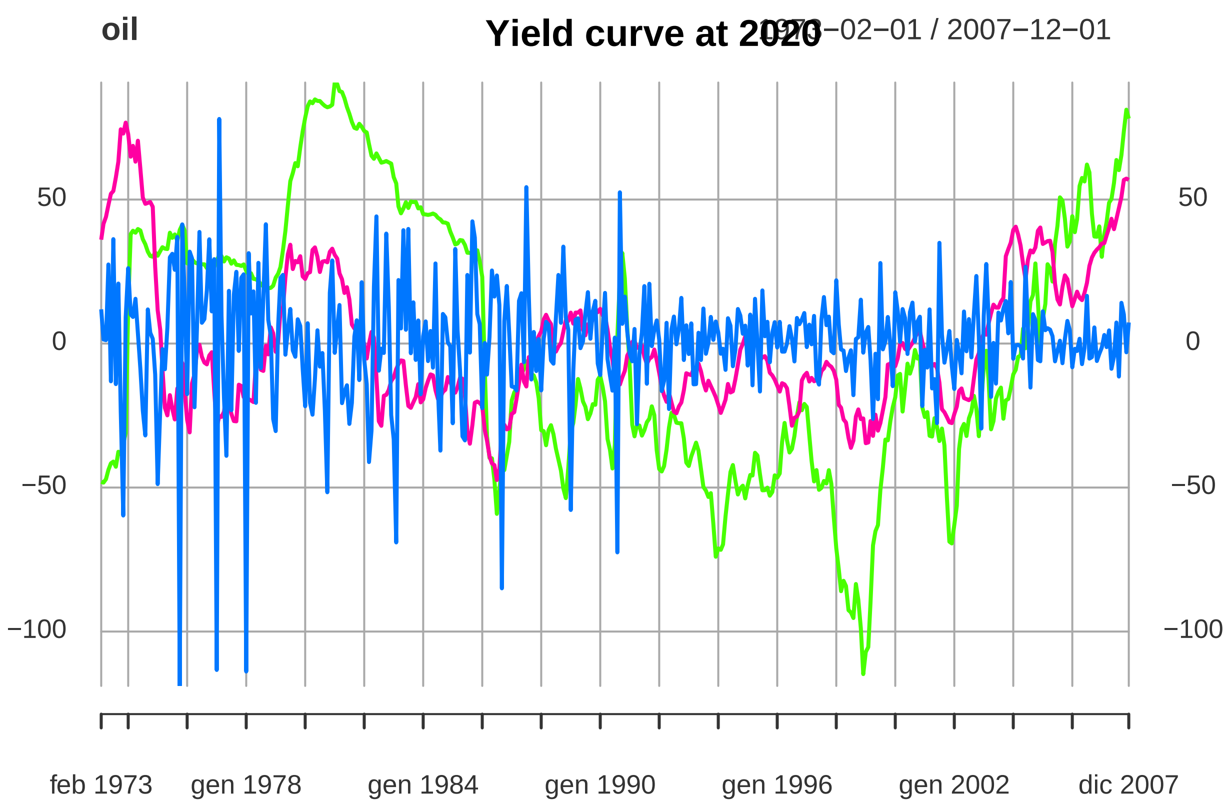
Assignment

Point 1

The plot below represents three monthly time series, in the order:

1. $\Delta\, prod$: % change in global crude oil production
2. rpo_t: the real price of oil
3. rea_t : index of the real economic activity

from 1973:1 to 2007:12.



From the acf we can clearly see the presence of an autocorrelation process. From the partial autocorrelation function we can infer that it's probably first-order autocorrelation since the only significant column is the first one (also the second one, but it has a negative sign).

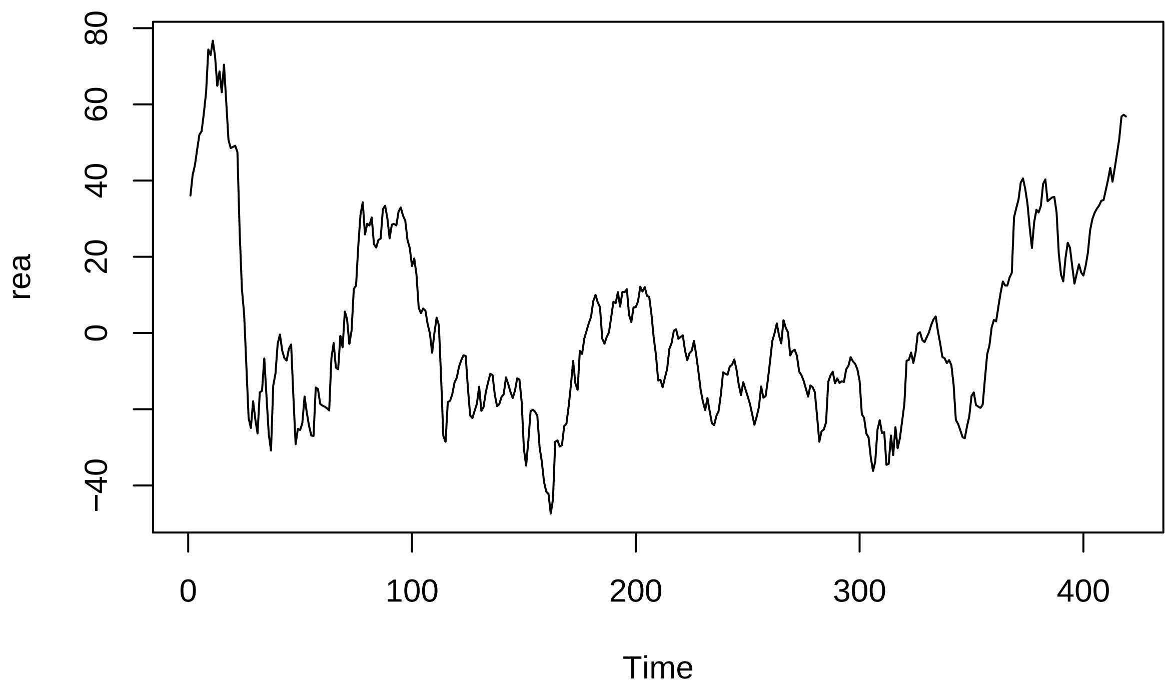
In order to test if the rea_t is an $I(1)$, we will use an ADF test with a minimun lag $= 1$. We will perform the test using four different specifications of the process:

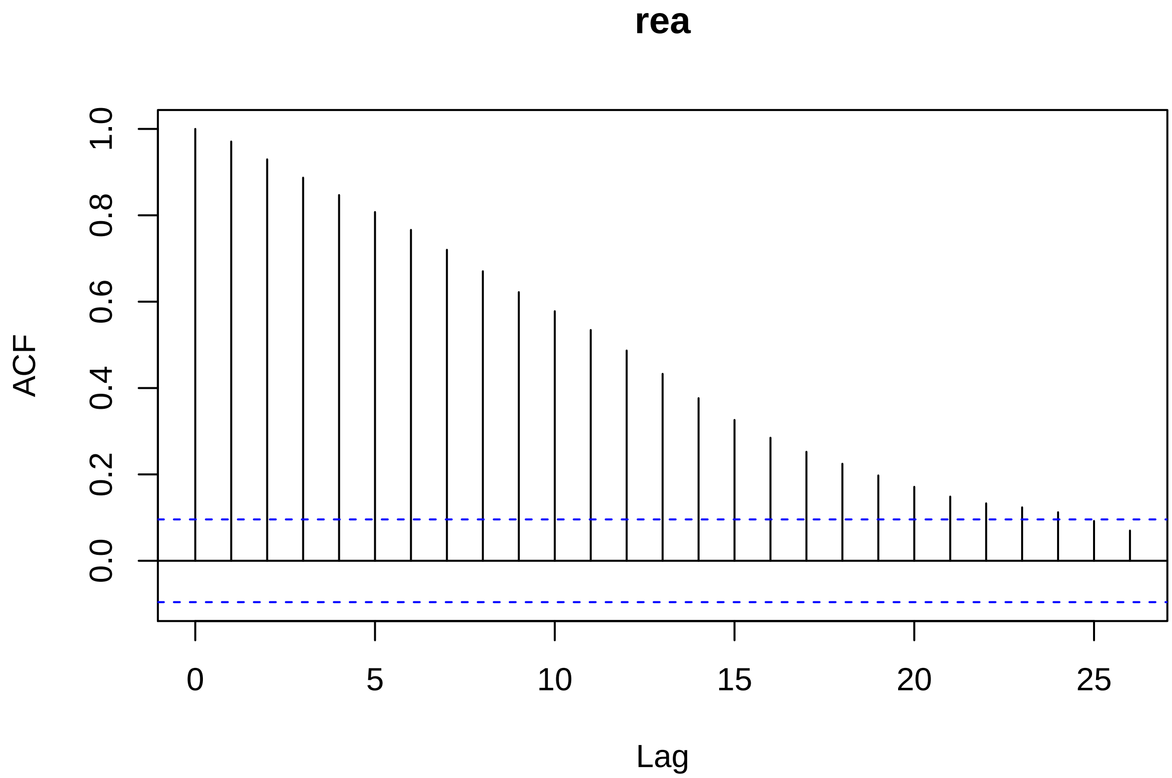
1. No constant, no trend
2. Constant
3. Constant with trend

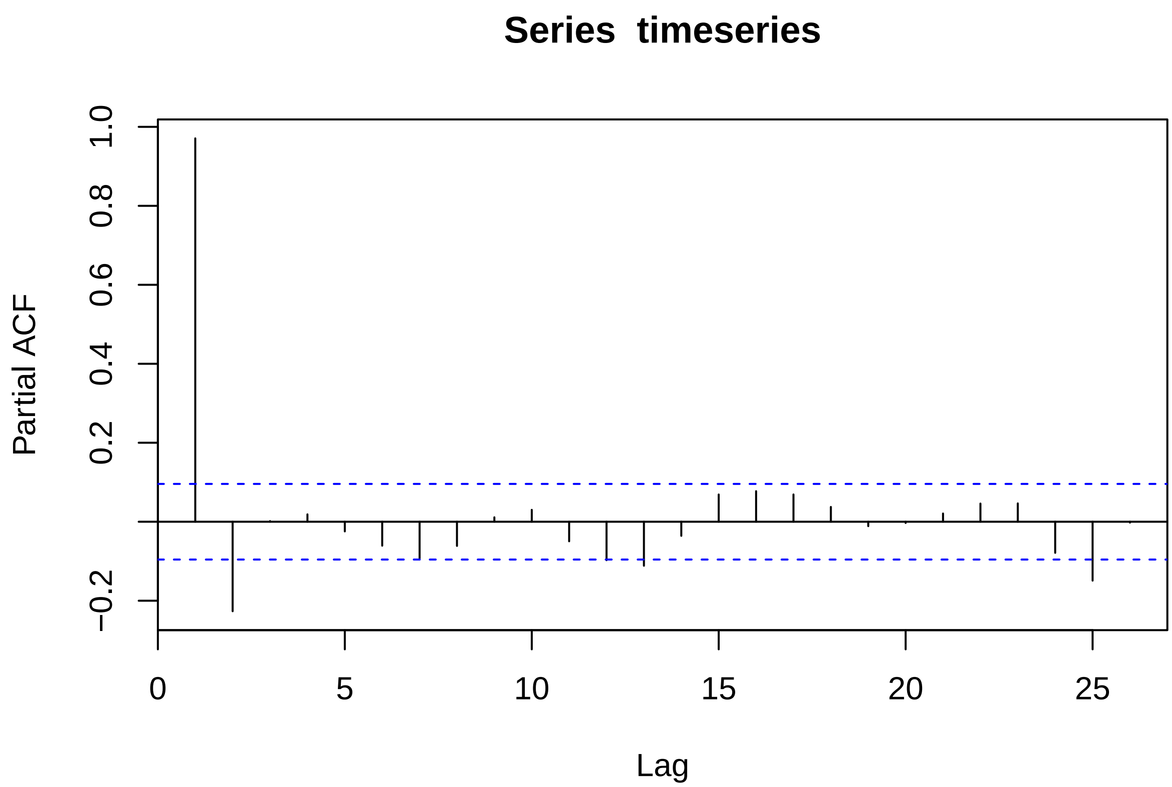
First, we print the rea_t time series graph. Then, we perform the different types of the test with a maximum lag order of 12:

$$rea_t = \alpha + \delta_1 rea_{t-1} + ... + \delta_1 2 rea_{t-12}$$

The criteria for selection of the lag order is selecting the one which has lower BIC:




rea
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
Lag




```
## [1] "Without constant and without time trend"
```

```
##
## === Test statistics =========
##              tau1
## statistic -3.0561
##
## === Test critical values ====
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
## === Combined output =========
## [1] "-3.06 [1]***"
```

```
## [1] "Max lag : 12"
```

```
## [1] "Lag used: 1"
```

```
## [1] "BIC: 2400.42837419751"
```

```
## [1] "With constant and without time trend"
```

```
##
## === Test statistics =========
```

```
##              tau2   phi1
## statistic -3.0642 4.6954
##
## === Test critical values ====
##       1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
##
## === Combined output =========
## [1] "-3.06 [1]**"

## [1] "Max lag : 12"

## [1] "Lag used: 1"

## [1] "BIC: 2406.39318640431"

## [1] "With constant and with time trend"

##
## === Test statistics =========
##              tau3   phi2   phi3
## statistic -3.2836 4.5302 6.7945
##
## === Test critical values ====
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
##
## === Combined output =========
## [1] "-3.28 [1]*"

## [1] "Max lag : 12"

## [1] "Lag used: 1"

## [1] "BIC: 2408.28425811698"
```

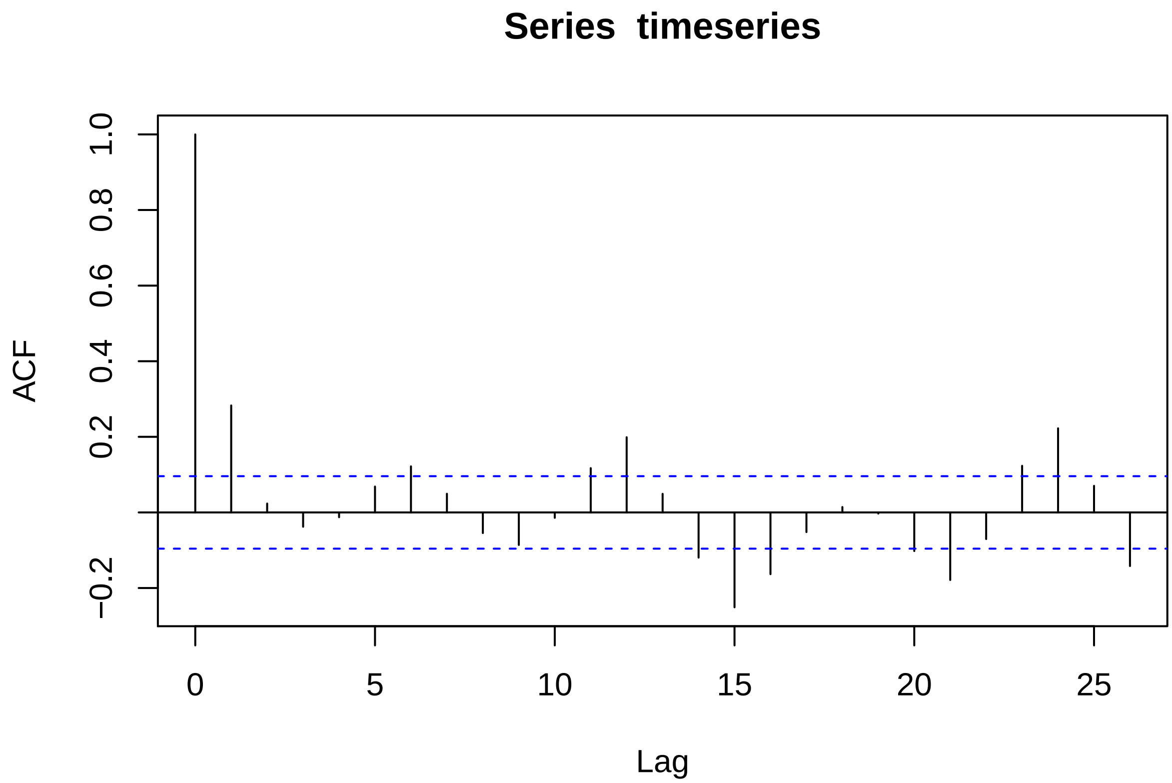
The results of the ADF tests shows that the process is stationary with the simplest specification (without constant and time trend), up to the third significance level (over 1%). However, the other possible specification, which add a constant and then also a time trend present higher p-values (also, it's got the lowest BIC), thus the specification we are going to select is the first one. This proves that the process is $I(1)$, since the first specification includes only one lag (without any constant and time trend). This is consistent with what we should expect, since rea_t is computed as a percentage deviation from the mean (it's basically an indicator of the business cycle).

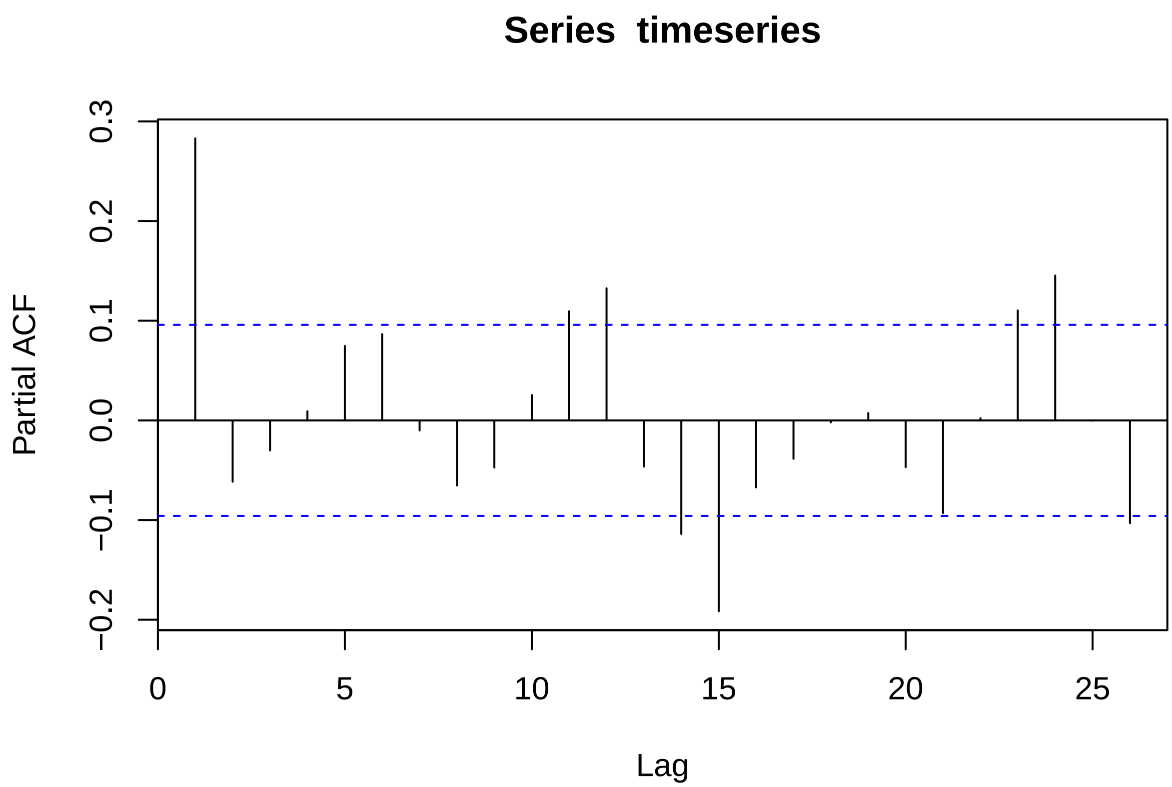
Thus, the specification we select in the end is:

$$\delta rea_t = \delta_1 rea_{t-1} + ... + \delta_1 2 rea_{t-12}$$

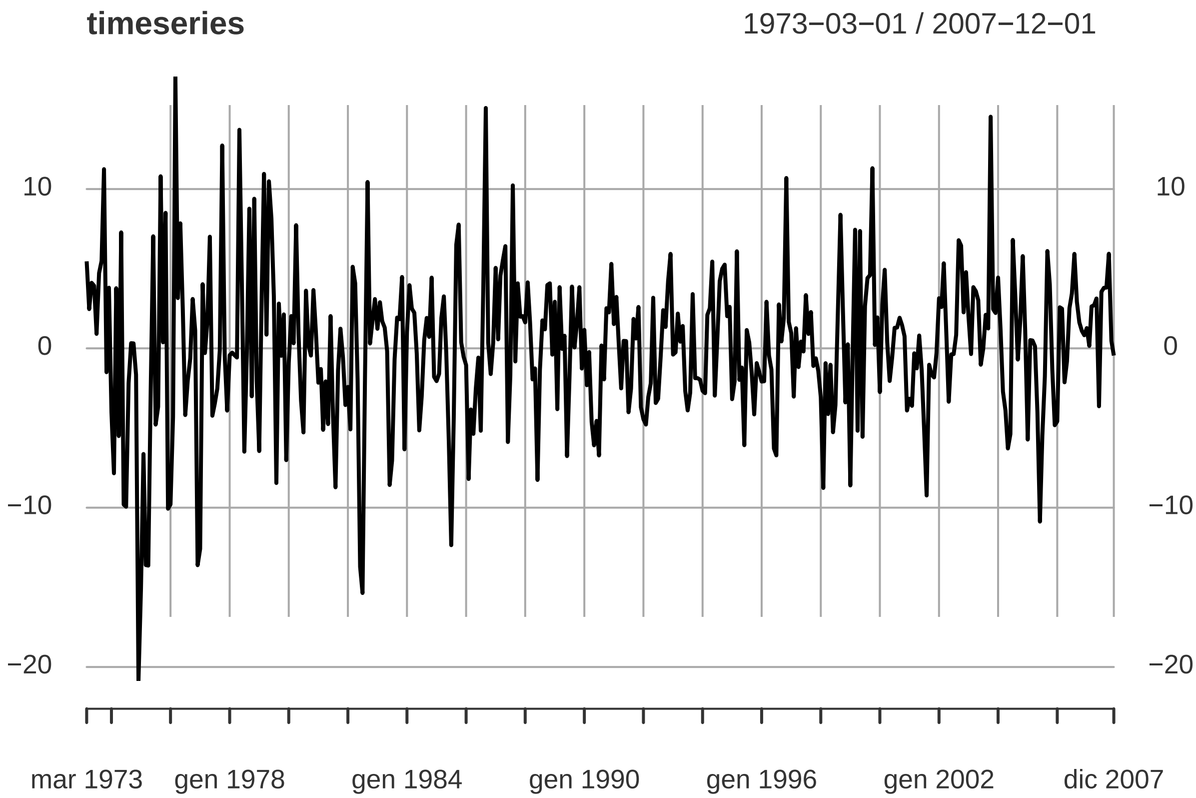
Point 2

We take the first difference of the time series *rea* and check if it is stationary with an adf test. Before that we print the time series of the first differences, its acf and pacf to understand the correct specification for the ADF test.

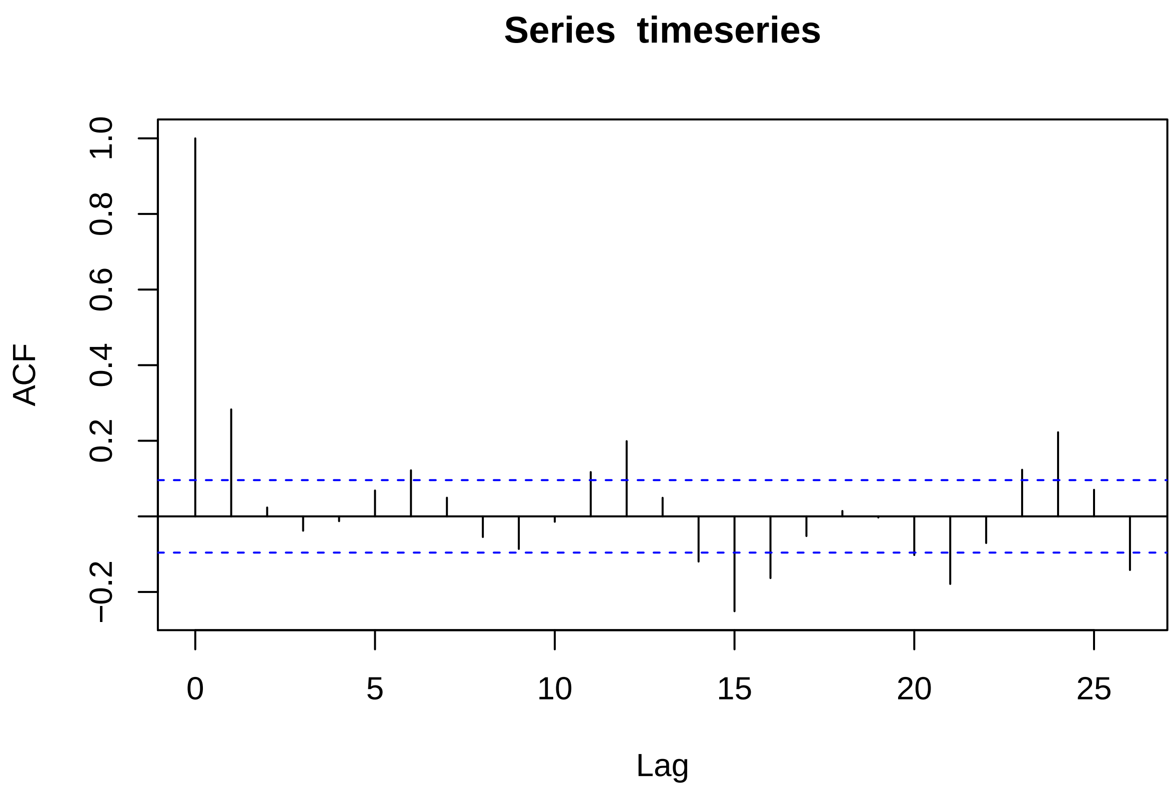




Series timeseries
Partial ACF
0.3
0.2
0.1
0.0
−0.1
−0.2
0
5
10
15
20
25
Lag



timeseries
1973−03−01 / 2007−12−01
10
0
−10
−20
mar 1973
gen 1978
gen 1984
gen 1990
gen 1996
gen 2002
dic 2007



The graphs above indicate the stationarity of the process. Indeed the acf, when the $lag > 2$ shows an autcorrelation that is not statisticaly different from 0 (except for a few peaks). As for the partial autocorrelation, it is statistically different only for some lags>10 (except, of course, for lag = 1). From the plot of the time series we can see a mean reverting process, and so I will opt for the specifications without constant and time trend, because it is less restrictive. So the test will have the following specifications:

$$\Delta rea_t = \delta_1 \Delta reat_{t-1} + ... + \delta_1 2 \Delta reat_{t-12}$$

$$\Delta rea_t = \alpha + \delta_1 \Delta reat_{t-1} + ... + \delta_1 2 \Delta reat_{t-12}$$

$$\Delta rea_t = \alpha + \beta * t + \delta_1 \Delta reat_{t-1} + ... + \delta_1 2 \Delta reat_{t-12}$$

The test will be performed with all possible three specification, and the specification with lower adf will be selected.

```
## [1] "Without constant and without time trend"
```

```
##
## === Test statistics =========
##               tau1
## statistic -12.928
##
## === Test critical values ====
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
## === Combined output =========
## [1] "-12.93 [1]***"
```

```
## [1] "Max lag : 12"

## [1] "Lag used: 1"

## [1] "BIC: 2401.03338921915"

## [1] "With constant and without time trend"

##
## === Test statistics =========
##               tau2   phi1
## statistic -12.913 83.372
##
## === Test critical values ====
##       1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
##
## === Combined output =========
## [1] "-12.91 [1]***"

## [1] "Max lag : 12"

## [1] "Lag used: 1"

## [1] "BIC: 2407.05552268091"

## [1] "With constant and with time trend"

##
## === Test statistics =========
##               tau3   phi2   phi3
## statistic -13.095 57.158 85.736
##
## === Test critical values ====
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
##
## === Combined output =========
## [1] "-13.09 [1]***"

## [1] "Max lag : 12"

## [1] "Lag used: 1"

## [1] "BIC: 2409.43562388553"
```

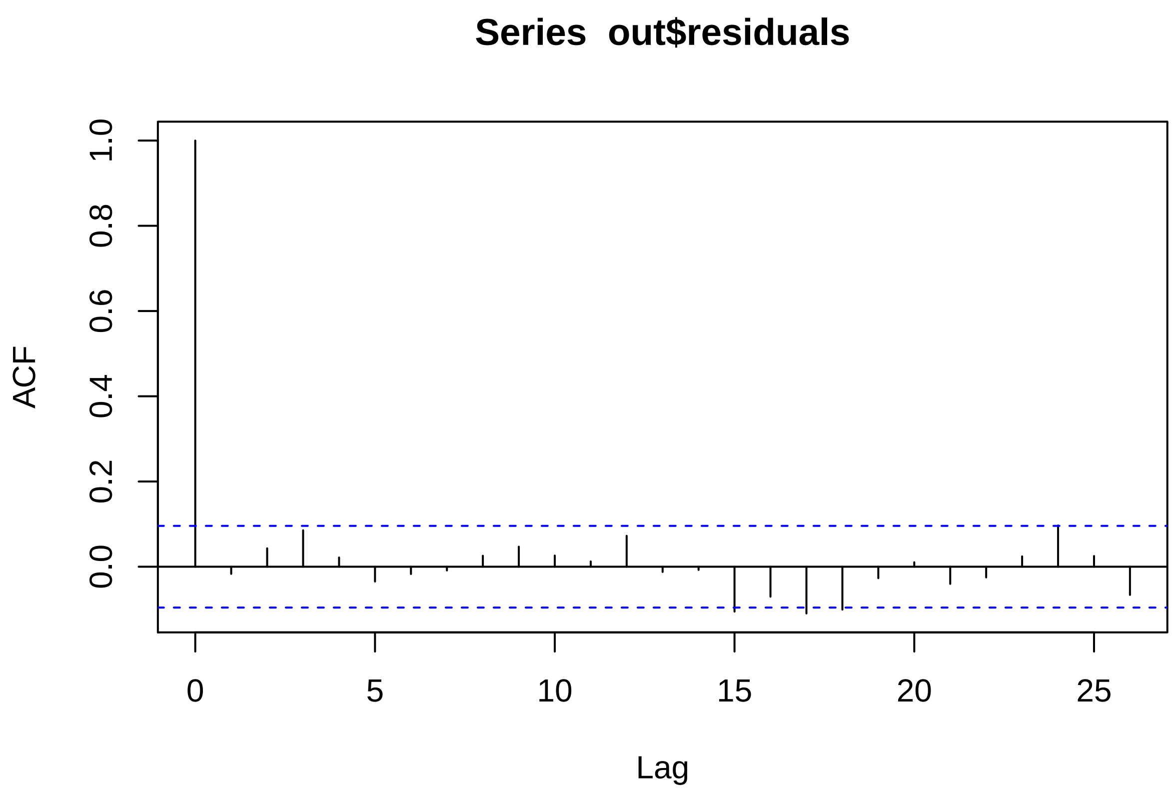
The test above shows another time the stationarity of the process, since with all specifications we reject the null hypothesis of non-stationarity up and beyond the 1% significance level. Furthermore, we select the simplest specification yet another time, since even if more complex specifications yield lower values of the p-value, the BIC increases, and with the first specification we already have a p-value that is asyntotically equal to zero.

Point 3

We select the best arma model setting the hyper-paramenters (p,q), using the BIC criteria, through the "best_arima" and the "bic_score" functions reported below

```r
# Function calculating the BIC score

bic_score <- function(k, n, l) {
    x <-  2 * k * log(n) - 2 * l
    return(x)
}

# Best arima model selected with the BIC criterion
bestarima <- function(timeseries, maxlag) {
    plag    <- 1:maxlag
    qlag    <- 1:maxlag

    model1   <- matrix(NA, nrow = 0, ncol = 3)
    colnames(model1) <- c("p", "q", "BIC")
    for (p in plag) {
       for (q in qlag) {
        out <- tryCatch(
        {
            # Just to highlight: if you want to use more than one
            # R expression in the "try" part then you'll have to
            # use curly brackets.
            # 'tryCatch()' will return the last evaluated expression
            # in case the "try" part was completed successfully

            arima(timeseries, order = c(p, 0, q))
            # The return value of `readLines()` is the actual value
            # that will be returned in case there is no condition
            # (e.g. warning or error).
            # You don't need to state the return value via `return()` as code
            # in the "try" part is not wrapped inside a function (unlike that
            # for the condition handlers for warnings and error below)
        },
        error=function(cond) {
            # Choose a return value in case of error
            return(NA)
        },
        warning=function(cond) {
            # Choose a return value in case of warning
            return(NA)
        }
    )
    if(any(!is.na(out))){
        x <- arima(timeseries, order = c(p, 0, q))
        x_bic <- bic_score(length(x$coef), x$nobs, x$loglik)



       } else {
          x_bic <- 9999
```

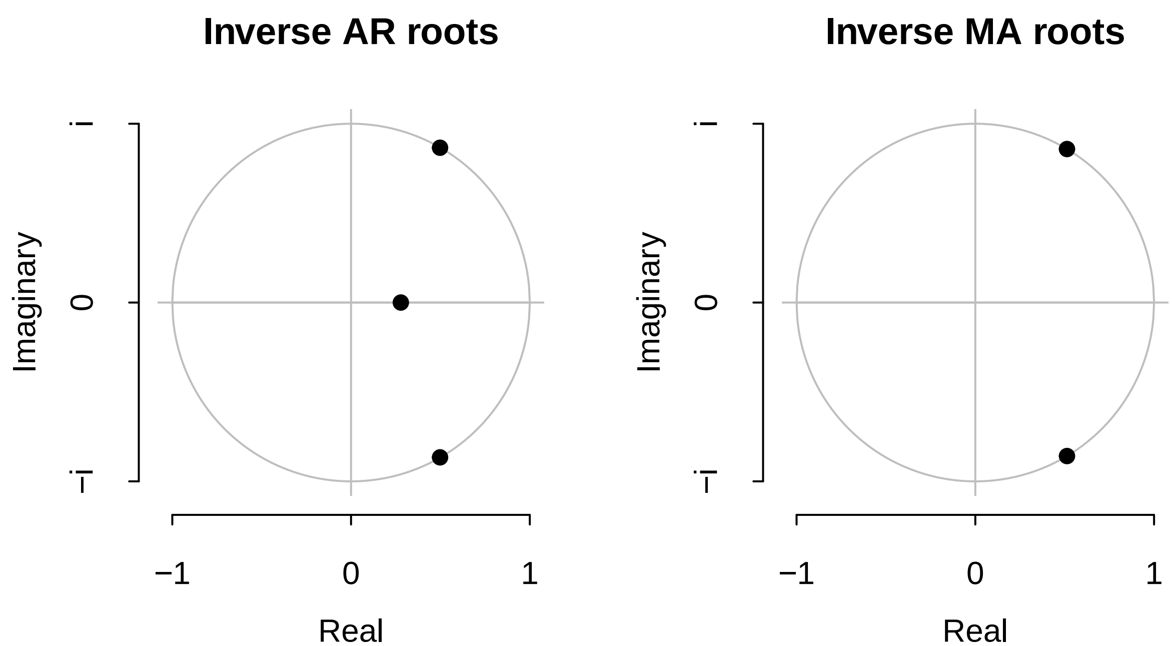
```
        }
        model1 <- rbind(model1, c(p, q, x_bic))
    }
    }
    p <- model1[which.min(model1[, "BIC"]), "p"]
    q <- model1[which.min(model1[, "BIC"]), "q"]
    out <- arima(timeseries, order = c(p, 0, q))
    acf(out$residuals)
   return(c(p, 0, q))
}
```

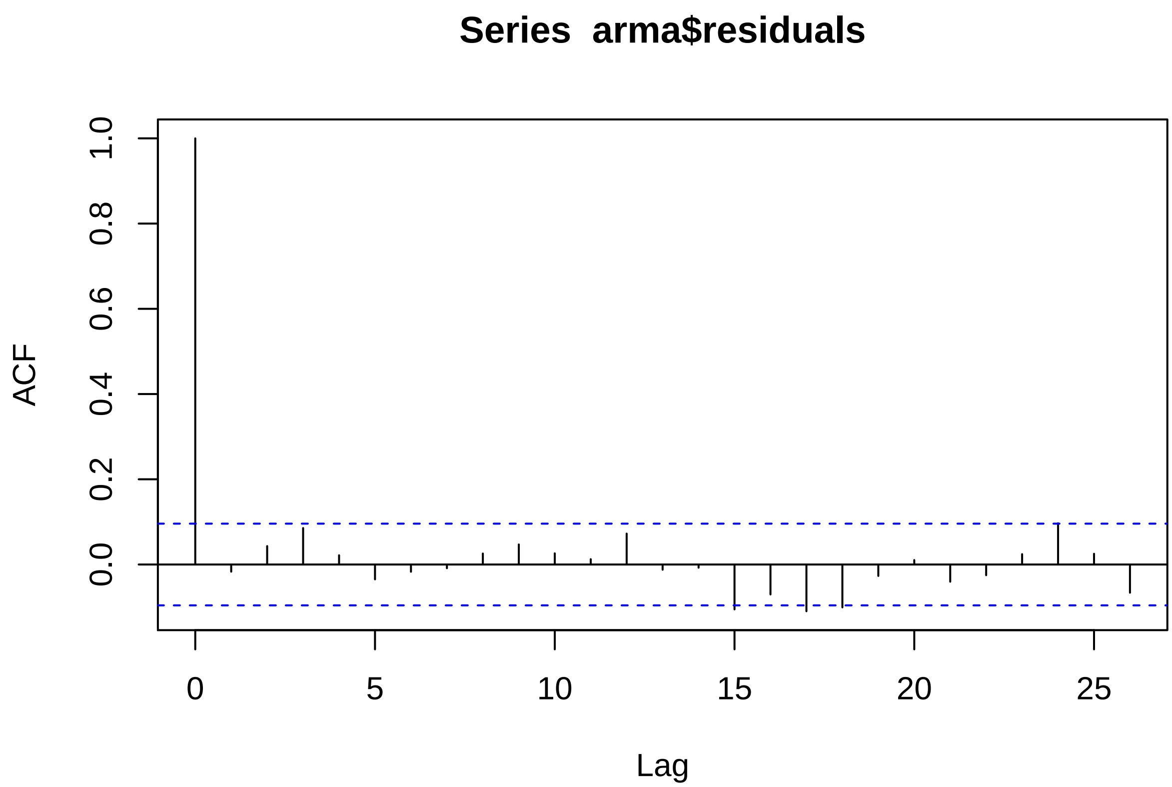


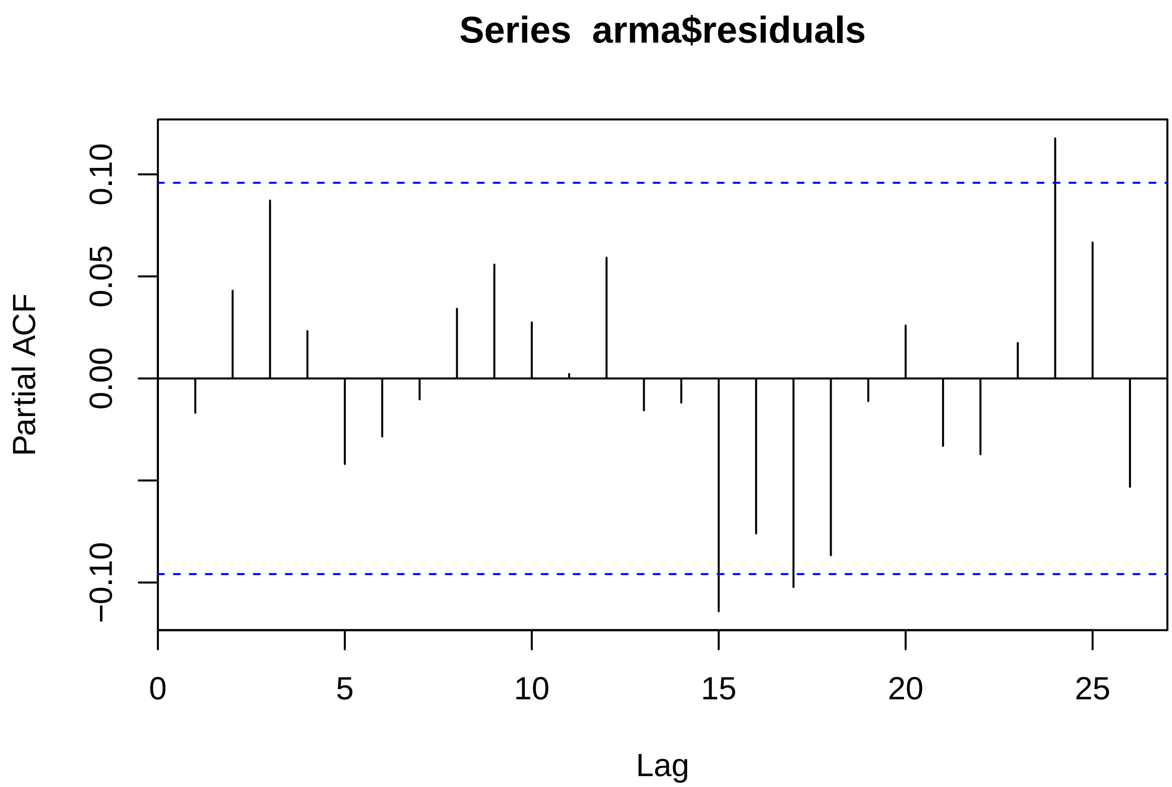

```
## [1] "BIC: 2488.2795280015"
```

```
##
## Call:
## arima(x = timeseries, order = best_arima, method = "ML")
##
## Coefficients:
##          ar1      ar2    ar3      ma1    ma2  intercept
##        1.274  -1.275  0.278  -1.026  1.000      0.047
## s.e.  0.048   0.048  0.048   0.014  0.021      0.284
##
## sigma^2 estimated as 18.6:  log likelihood = -1207.9,  aic = 2429.8
##
## Training set error measures:
```

```
##                       ME   RMSE    MAE    MPE  MAPE    MASE      ACF1
## Training set 0.0046073 4.3127 3.2826 65.463 204.7 0.76972 -0.016815
```




Series arma$residuals
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
Lag



The autcorrelation function of the residuals it is not statistically different from 0, it looks like white noise. So the arma model adopted is one the fit perfectly the time series:

$$y_t = \theta_1 y_{t-1} + \theta_2 y_{t-2} + \theta_3 y_{t-3} + \beta_1 \epsilon_{t-1} + \beta_2 \epsilon_{t-2} + \epsilon_t$$

The issue regarding this model is an overfitting one, since all the point in the timeseries has been used to fit the model, as opposite to the usual practice. But the aim of this model is not to provide a prediction for the series, but instead the understading of the process in the specific time span of the series.

Point 4

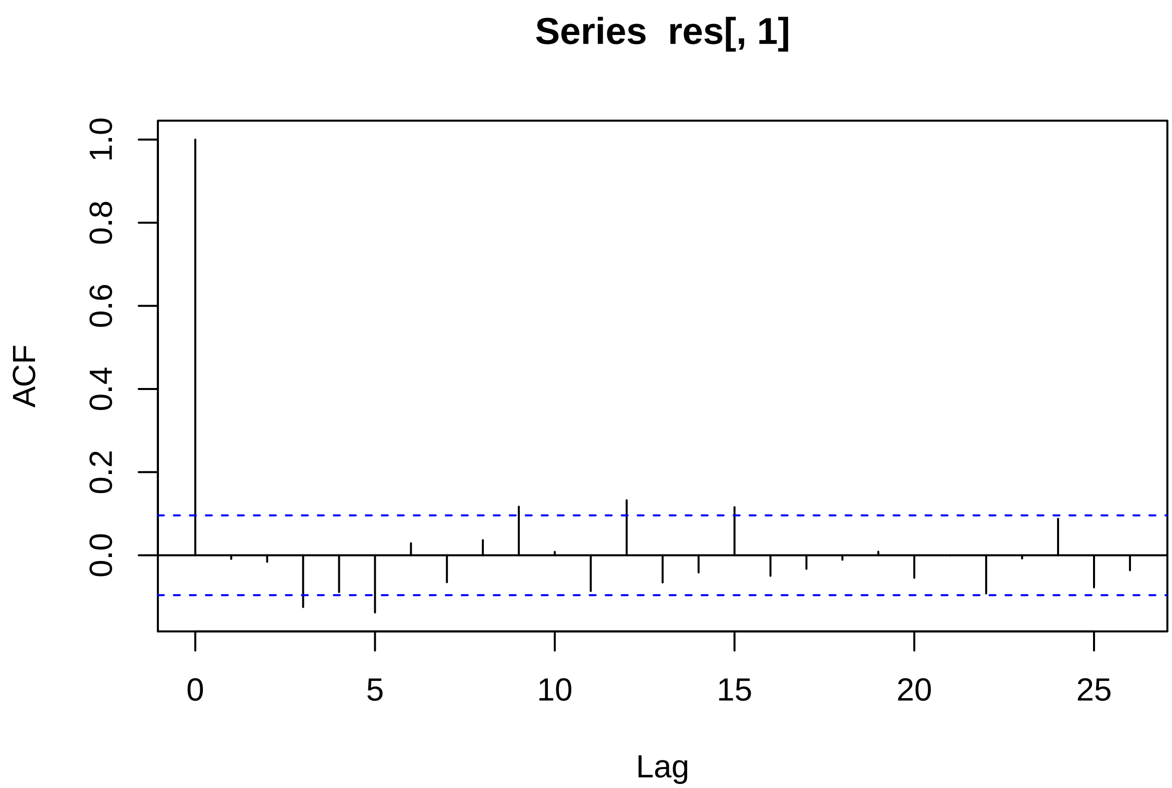
```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag  ADF p.value
## [1,]   0 22.3    0.99
## [2,]   1 32.7    0.99
## [3,]   2 41.8    0.99
## [4,]   3 50.3    0.99
## [5,]   4 59.7    0.99
## [6,]   5 66.9    0.99
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 22.3    0.99
```

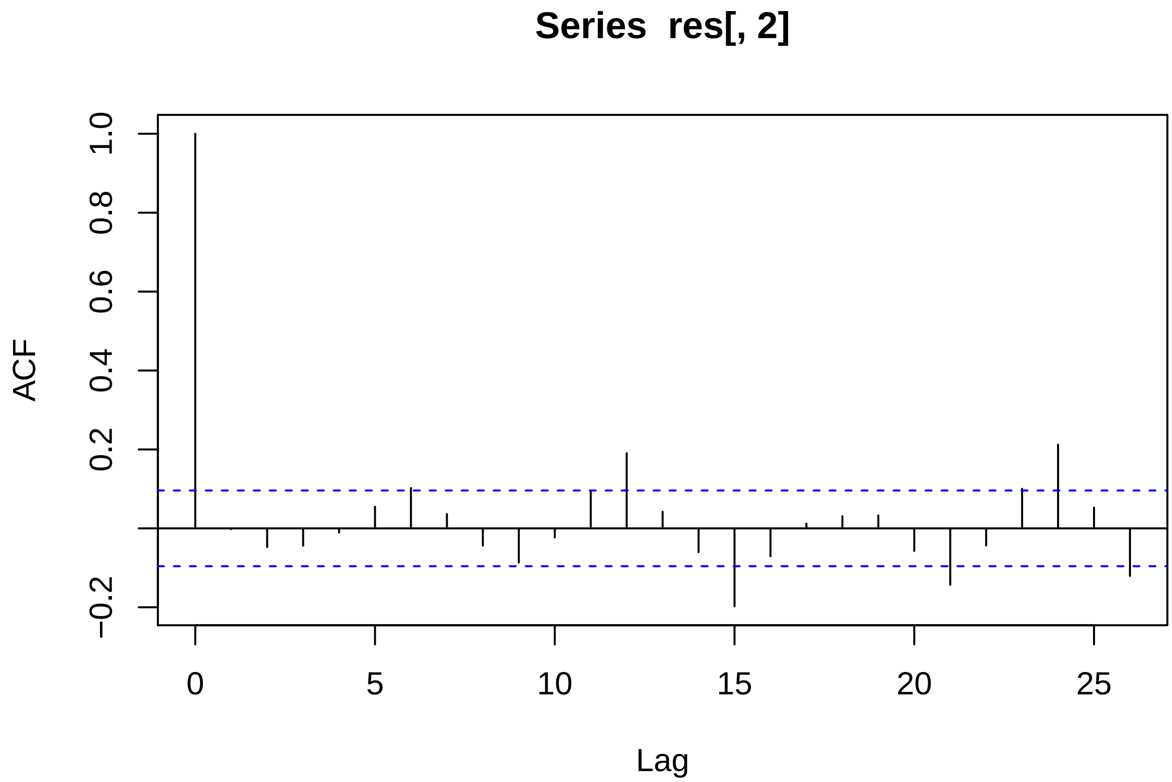
```
## [2,]   1 32.7    0.99
## [3,]   2 41.9    0.99
## [4,]   3 50.5    0.99
## [5,]   4 60.1    0.99
## [6,]   5 67.4    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 22.3    0.99
## [2,]   1 32.7    0.99
## [3,]   2 41.9    0.99
## [4,]   3 50.5    0.99
## [5,]   4 60.1    0.99
## [6,]   5 67.5    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

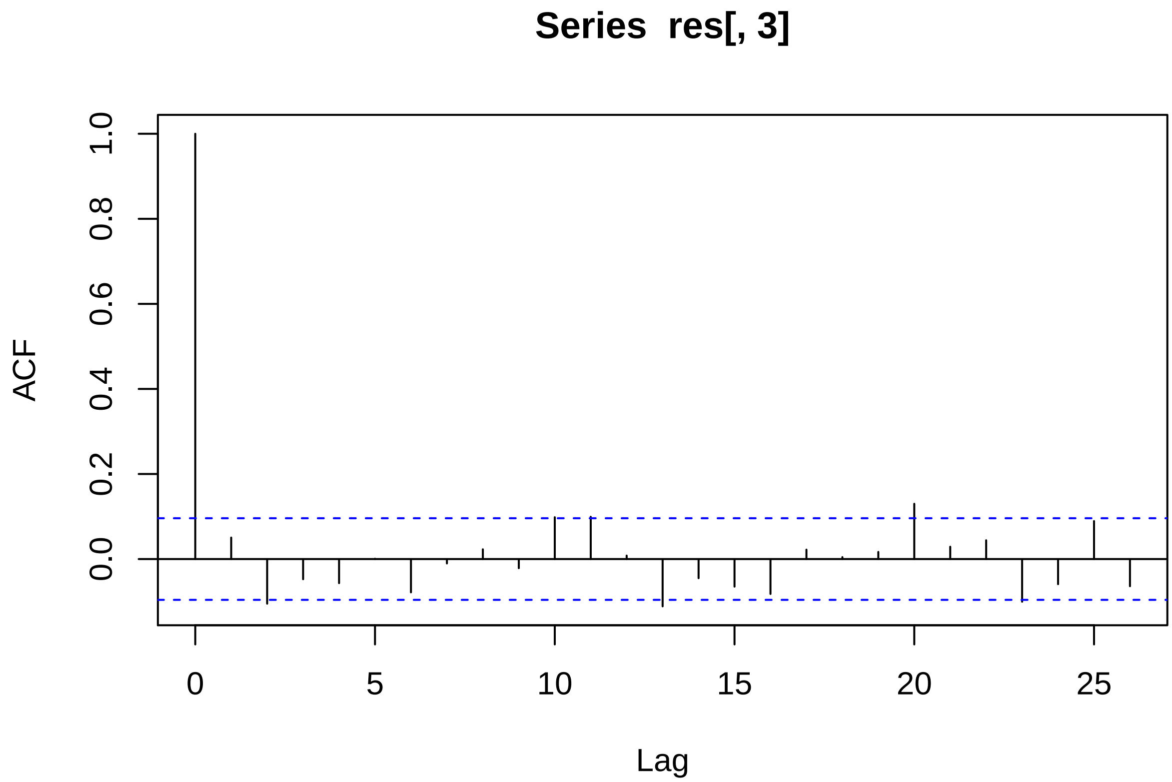
```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag  ADF p.value
## [1,]   0 2.47   0.990
## [2,]   1 1.42   0.960
## [3,]   2 1.60   0.973
## [4,]   3 1.63   0.975
## [5,]   4 1.56   0.970
## [6,]   5 1.35   0.955
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 2.46    0.99
## [2,]   1 1.41    0.99
## [3,]   2 1.60    0.99
## [4,]   3 1.63    0.99
## [5,]   4 1.56    0.99
## [6,]   5 1.35    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 2.47    0.99
## [2,]   1 1.42    0.99
## [3,]   2 1.61    0.99
## [4,]   3 1.64    0.99
## [5,]   4 1.57    0.99
## [6,]   5 1.36    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

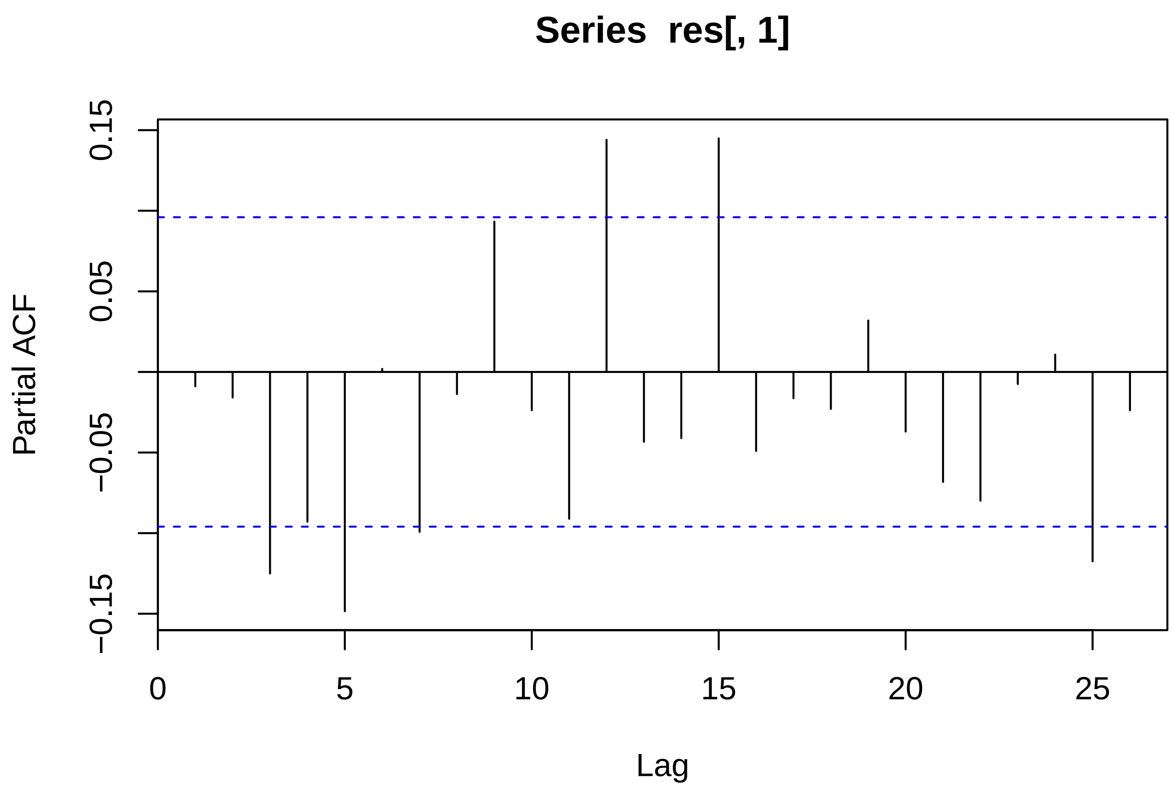
```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 1.852   0.984
## [2,]   1 0.579   0.811
## [3,]   2 0.886   0.899
```

```
## [4,]   3 0.933   0.906
## [5,]   4 1.072   0.923
## [6,]   5 1.081   0.924
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 1.847   0.990
## [2,]   1 0.579   0.989
## [3,]   2 0.886   0.990
## [4,]   3 0.933   0.990
## [5,]   4 1.071   0.990
## [6,]   5 1.081   0.990
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 2.137    0.99
## [2,]   1 0.714    0.99
## [3,]   2 1.070    0.99
## [4,]   3 1.145    0.99
## [5,]   4 1.313    0.99
## [6,]   5 1.332    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

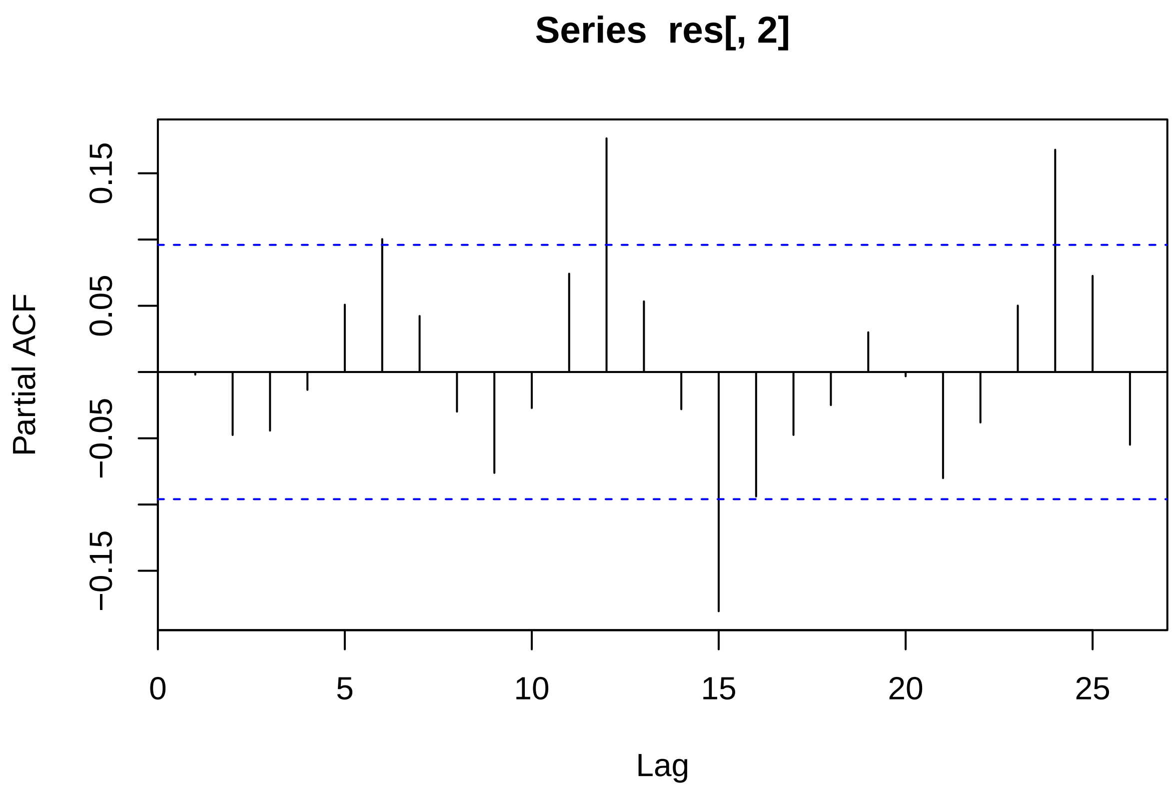


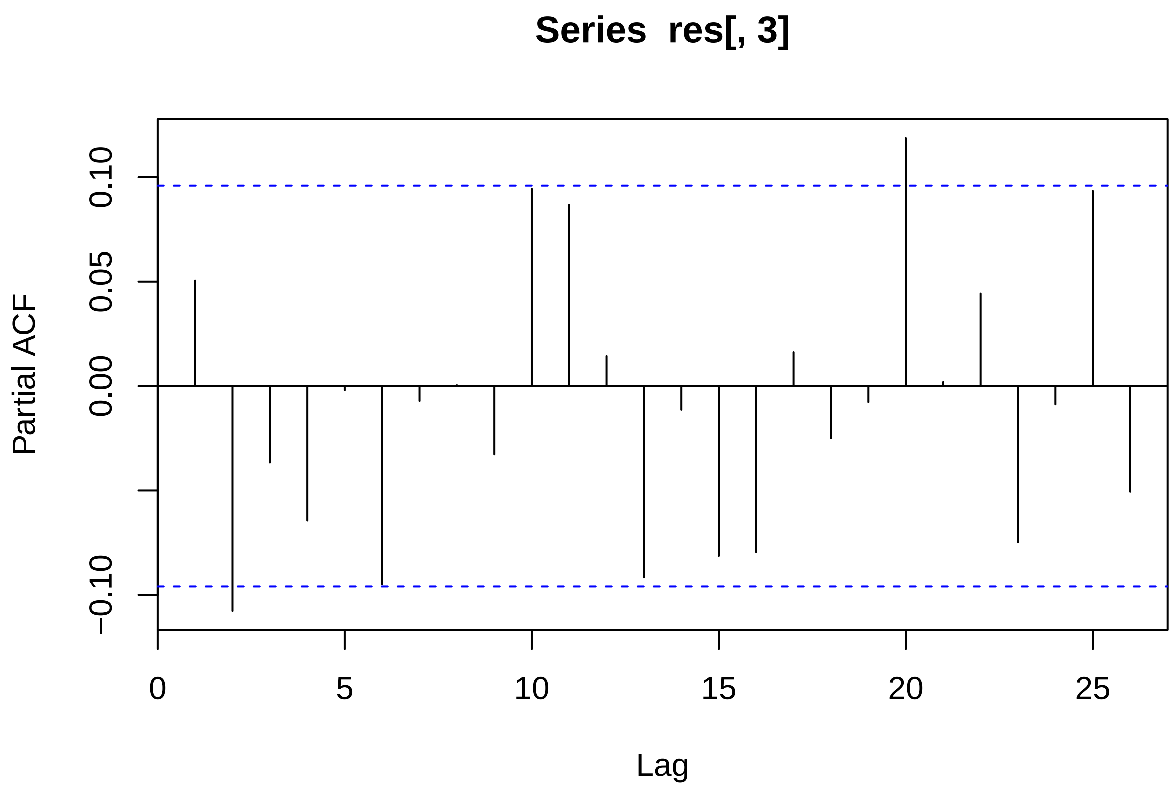

Series res[, 2]
ACF
1.0
0.8
0.6
0.4
0.2
−0.2
0
5
10
15
20
25
Lag


Series res[, 3]
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
Lag



Series res[, 1]
Partial ACF
0.15
0.05
−0.05
−0.15
0
5
10
15
20
25
Lag


Series res[, 2]
Partial ACF
0.15
0.05
−0.05
−0.15
0
5
10
15
20
25
Lag



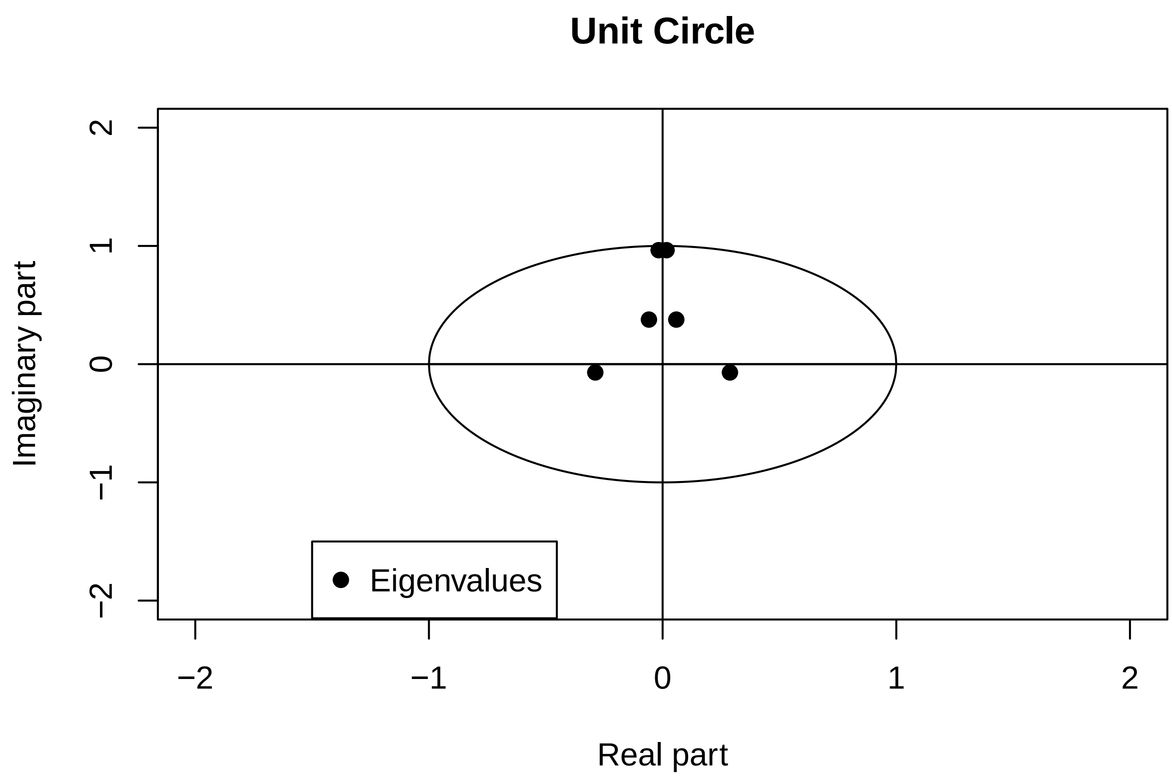

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: Dprod, rea, rpo
## Deterministic variables: const
## Sample size: 417
## Log Likelihood: -4416.681
## Roots of the characteristic polynomial:
## 0.964 0.964 0.381 0.381 0.297 0.297
## Call:
## VAR(y = oil, type = "const", lag.max = 3, ic = "HQ")
##
##
## Estimation results for equation Dprod:
## ======================================
## Dprod = Dprod.l1 + rea.l1 + rpo.l1 + Dprod.l2 + rea.l2 + rpo.l2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## Dprod.l1  -0.1058     0.0493   -2.15    0.032 *
## rea.l1     0.2499     0.2108    1.19    0.236
## rpo.l1     0.0319     0.1483    0.21    0.830
## Dprod.l2  -0.0771     0.0491   -1.57    0.117
## rea.l2    -0.3005     0.2122   -1.42    0.158
## rpo.l2    -0.0380     0.1473   -0.26    0.796
## const      0.9912     1.0064    0.98    0.325
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 20.5 on 410 degrees of freedom
## Multiple R-Squared: 0.0217,  Adjusted R-squared: 0.0074
## F-statistic: 1.52 on 6 and 410 DF,  p-value: 0.171
##
##
## Estimation results for equation rea:
## ====================================
## rea = Dprod.l1 + rea.l1 + rpo.l1 + Dprod.l2 + rea.l2 + rpo.l2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## Dprod.l1  0.00103    0.01098    0.09    0.925
## rea.l1    1.26124    0.04695   26.87  < 2e-16 ***
## rpo.l1    0.07752    0.03302    2.35    0.019 *
## Dprod.l2  0.01928    0.01094    1.76    0.079 .
## rea.l2   -0.28884    0.04727   -6.11  2.3e-09 ***
## rpo.l2   -0.07709    0.03281   -2.35    0.019 *
## const    -0.02085    0.22413   -0.09    0.926
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.56 on 410 degrees of freedom
## Multiple R-Squared: 0.964,   Adjusted R-squared: 0.964
## F-statistic: 1.85e+03 on 6 and 410 DF,  p-value: <2e-16
##
##
## Estimation results for equation rpo:
## ====================================
## rpo = Dprod.l1 + rea.l1 + rpo.l1 + Dprod.l2 + rea.l2 + rpo.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## Dprod.l1  3.29e-03   1.50e-02    0.22    0.826
## rea.l1    3.83e-02   6.39e-02    0.60    0.550
## rpo.l1    1.39e+00   4.50e-02   30.81   <2e-16 ***
## Dprod.l2 -3.41e-02   1.49e-02   -2.29    0.022 *
## rea.l2   -6.91e-06   6.44e-02    0.00    1.000
## rpo.l2   -4.09e-01   4.47e-02   -9.15   <2e-16 ***
## const     2.06e-01   3.05e-01    0.68    0.500
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 6.22 on 410 degrees of freedom
## Multiple R-Squared: 0.982,   Adjusted R-squared: 0.981
## F-statistic: 3.67e+03 on 6 and 410 DF,  p-value: <2e-16
##
##
##
## Covariance matrix of residuals:
##        Dprod   rea   rpo
## Dprod 419.79  6.49 -4.89
```

```
## rea      6.49 20.82  1.53
## rpo     -4.89  1.53 38.63
##
## Correlation matrix of residuals:
##          Dprod    rea     rpo
## Dprod  1.0000 0.0694 -0.0384
## rea    0.0694 1.0000  0.0540
## rpo   -0.0384 0.0540  1.0000
```

```
##           Dprod     rea     rpo
## Dprod 419.7911  6.4860 -4.8947
## rea     6.4860 20.8229  1.5306
## rpo    -4.8947  1.5306 38.6291
```

```
##            Dprod      rea       rpo
## Dprod  1.000000 0.069373 -0.038437
## rea    0.069373 1.000000  0.053968
## rpo   -0.038437 0.053968  1.000000
```



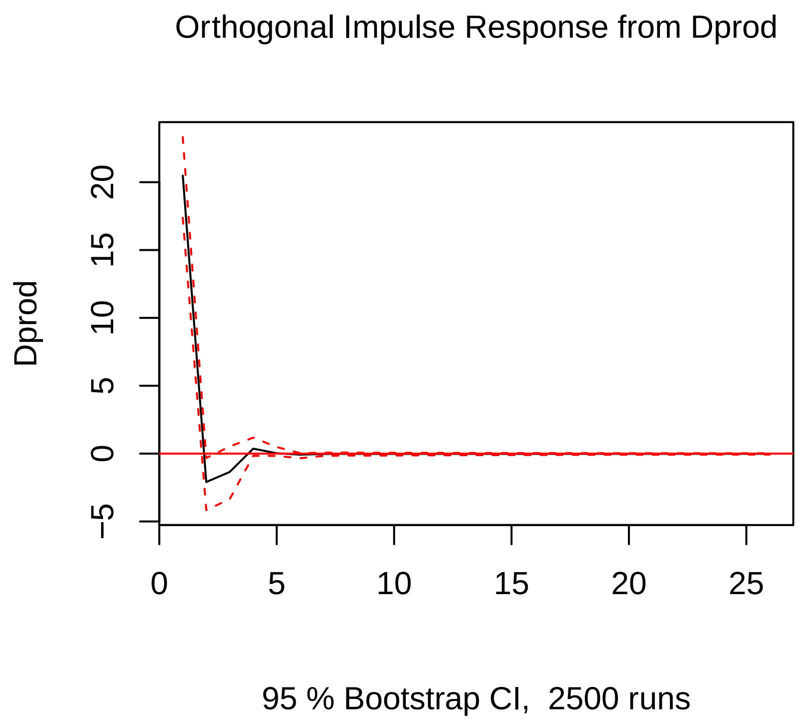
Point 5

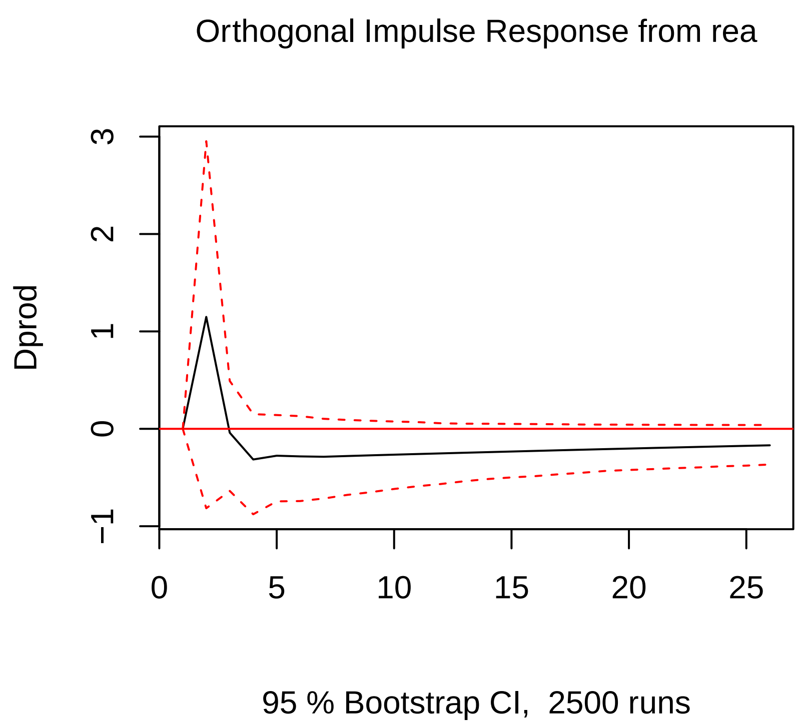
We report below the mapping from the text of the assignment:

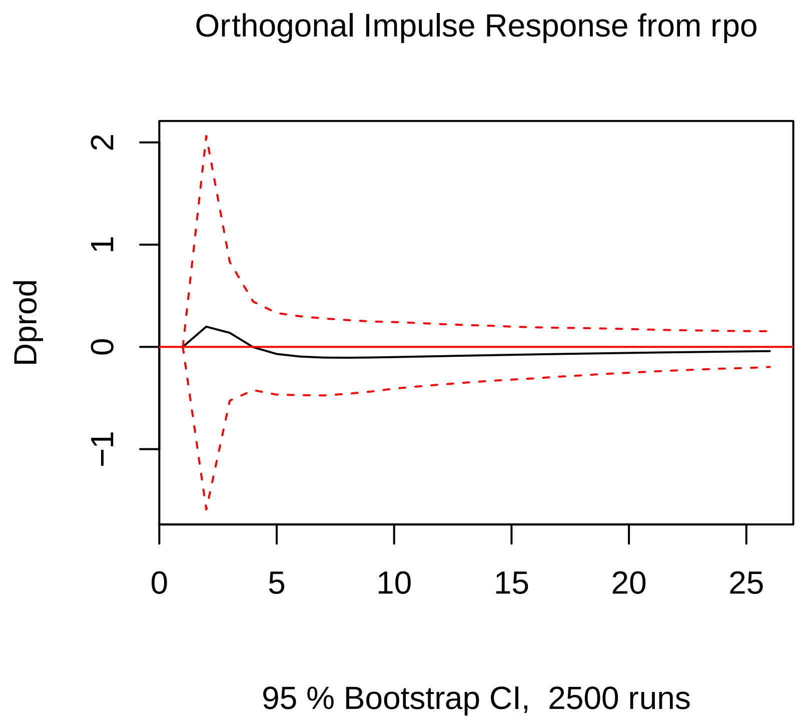
$$\begin{bmatrix} e_t^{\triangle \prod} \\ e_t^{rea} \\ e_t^{rpo} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{32} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} u_t^{oil\ supply\ shock} \\ u_t^{agg\ demand\ shock} \\ u_t^{oil\ specific\ demand\ shock} \end{bmatrix}$$

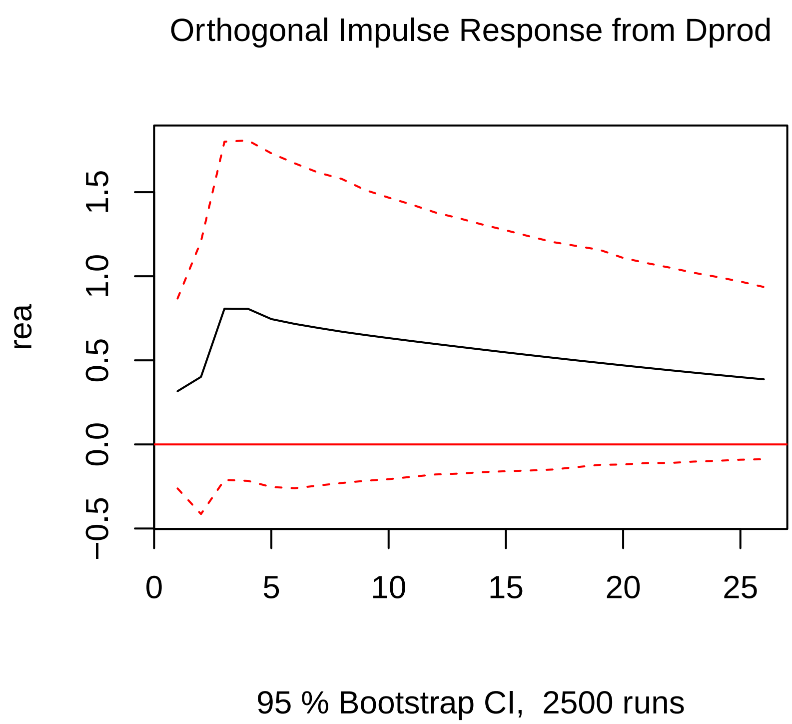
Based on the assumptions made in the text, we set c_{12}, c_{13}, and c_{23} equal to zero. This is consistent with the fact that we need to have a lower triangular matrix C, since in a Cholesky decomposition we need this restriction in order to get a unique solution.

Point 6

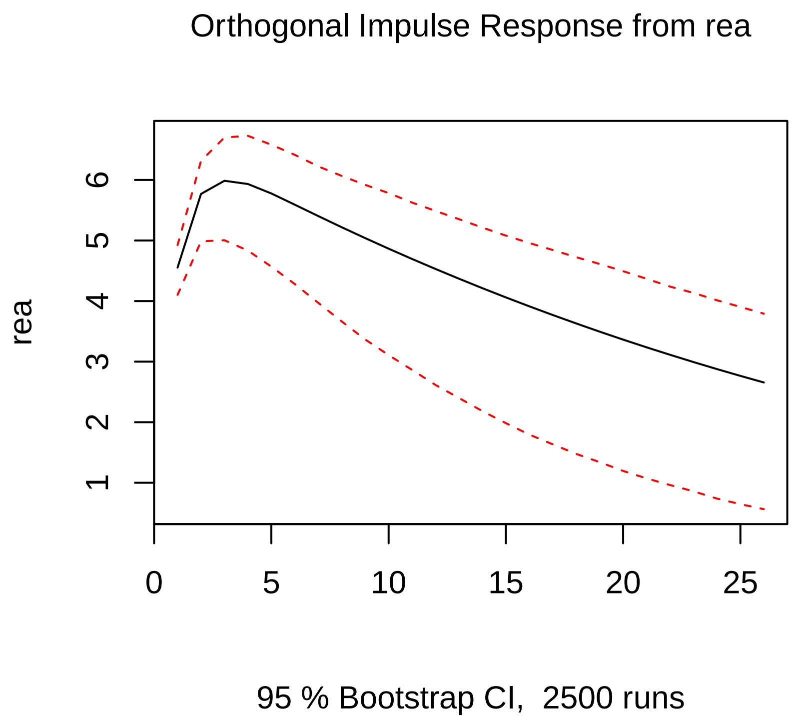

Orthogonal Impulse Response from Dprod
Dprod
−5
0
5
10
15
20
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs

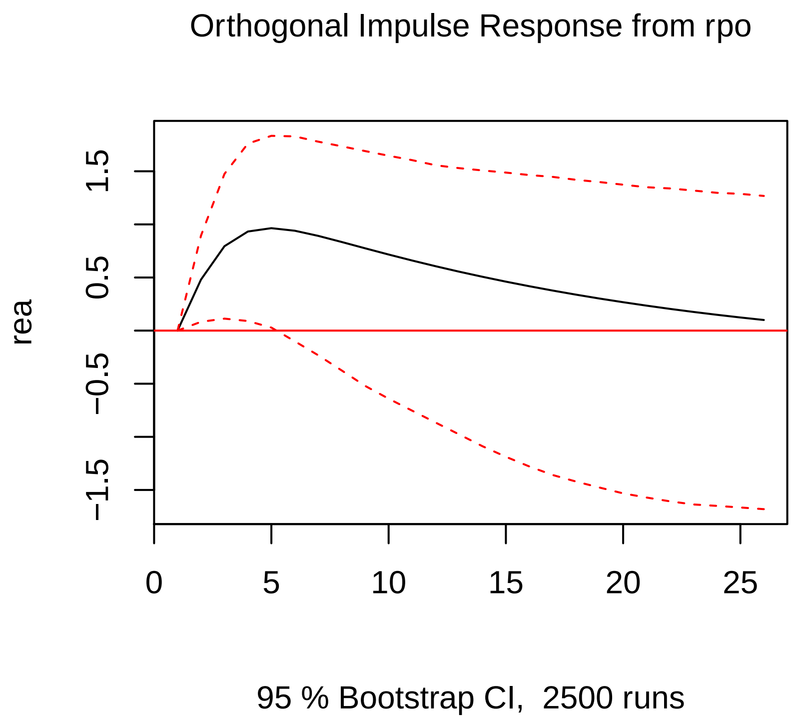

Orthogonal Impulse Response from rea
Dprod
−1
0
1
2
3
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs

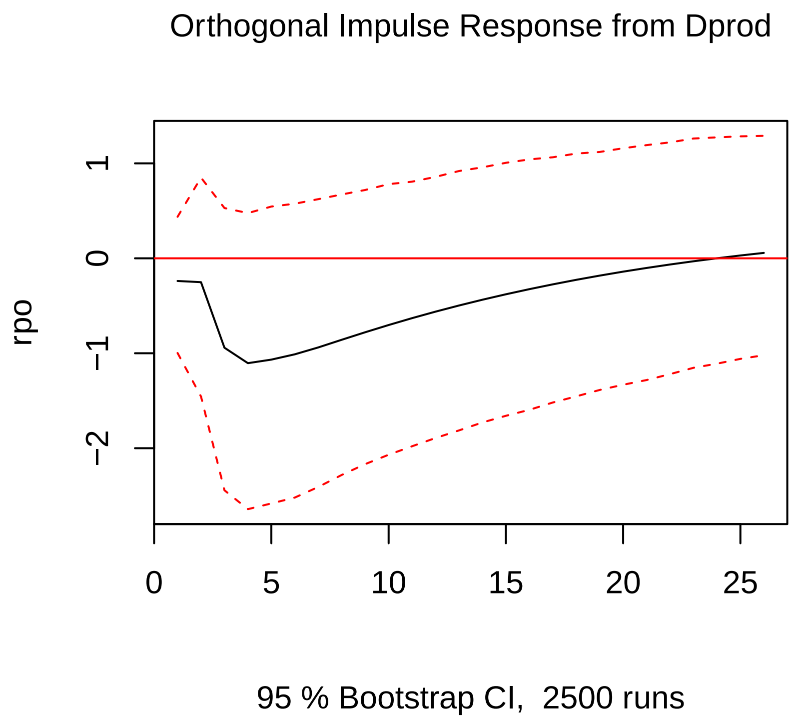

Orthogonal Impulse Response from rpo
Dprod
2
1
0
−1
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs

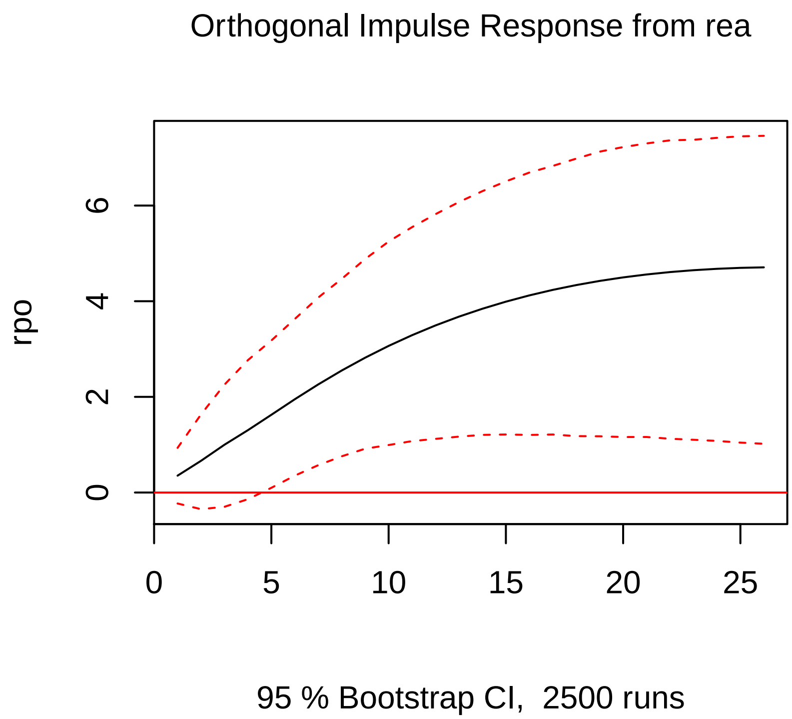


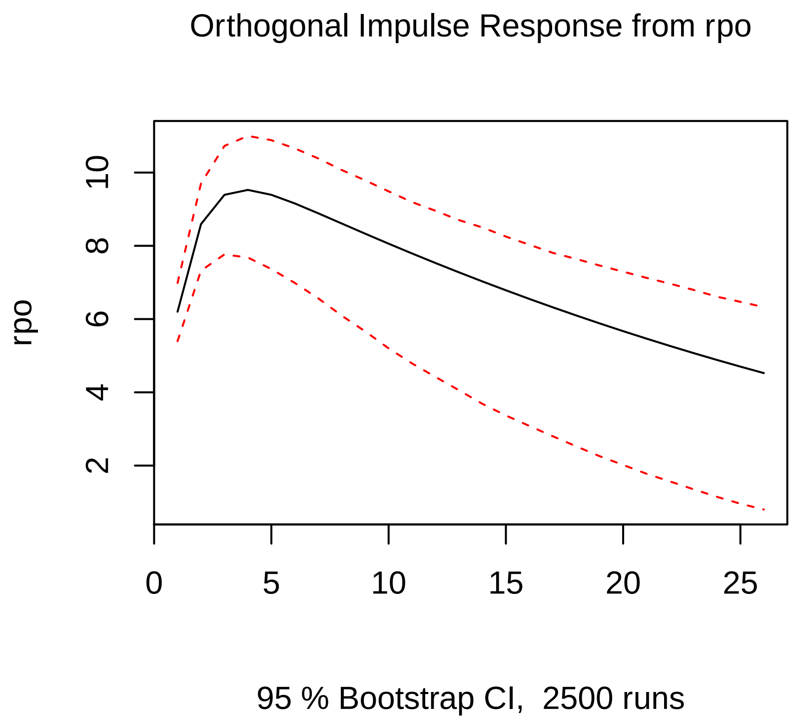
Orthogonal Impulse Response from Dprod
rea
−0.5
0.0
0.5
1.0
1.5
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs


Orthogonal Impulse Response from rea
rea
1
2
3
4
5
6
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs


Orthogonal Impulse Response from rpo
rea
1.5
0.5
−0.5
−1.5
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs


Orthogonal Impulse Response from Dprod
rpo
1
0
−1
−2
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs


Orthogonal Impulse Response from rea
rpo
0
2
4
6
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs


Orthogonal Impulse Response from rpo
rpo
2
4
6
8
10
0
5
10
15
20
25
95 % Bootstrap CI, 2500 runs