Advanced Econometrics Project 2021/22

In RMarkdown

33636339

6 May 2022

Table of Contents

Goldsmiths College, University of London.

Advanced Econometrics.

TIME SERIES

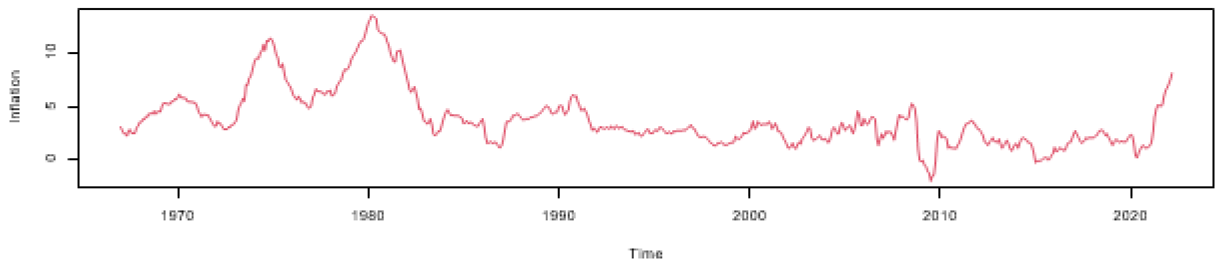
Words count: 1972

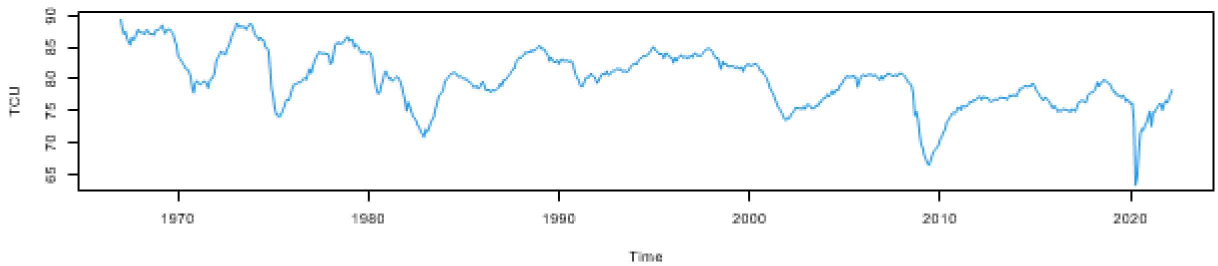
1. INTRODUCTION

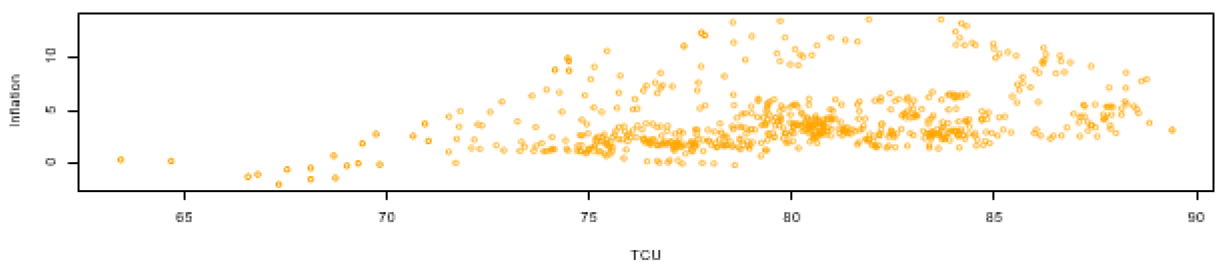
One of the main goals of central banks is to keep the inflation rate close to the set target to help the economy in achieving long term economic growth and stability in the whole economic and financial system. One indicator of economic performance is the Total Capacity Utilization Index, as can be considered as describing how efficiently the factors of production are being used. The scope of this project is to delve into the statistical relationship between TCU and Inflation, trying to capture possible correlations between the two variables, and investigate any possible causal mechanisms. This paper will try to explore the possibilities to build forecasting models that could help in predicting how inflation rate would react to a shock in TCU and vice versa. In this project, the data are fetched directly from the St. Louis FRED website. The Inflation Rate used for analysis is measured as monthly percentage change of CPI from one year ago; the TCU is calculated as percentage of capacity, with a monthly frequency, and it represents the percentage of resources used by firms and industries to produce demanded finished goods. Even though we expect to discover some sort of correlation between the two variables, we do not expect this to be strong. Particularly in regard to the causal mechanisms, it is believed that change in TCU do not cause significant and instant changes in Inflation Rate, as the Rate of Inflation depends on a variety of other factors; stronger the level of causation that might run in the opposite direction, as the capacity index might be directly influenced by the general level of prices which affects raw material costs, and might decrease demand for the final goods.

2. FETCHING THE DATA

```r
#The first step is plotting the time series in levels.

#Plot the data in your project

# The first graph shows the Inflation Rate, the second graph depicts the Total
#Capacity Utilization Rate, and the third graph shows the correlation between the
#two variables over time.
par(mfrow=c(3,1))
plot(Inflation, col=2)
plot(TCU, col=4)
plot(Inflation ~ TCU, col="orange")
```








```
cor(Inflation,TCU)

## [1] 0.3605283

#The correlation coefficient is 0.36 which suggest a fair level of correlation
between the two variables. By looking at the graph there seems to be a weak
positive trend, but further tests are required to learn more about the relationship
between the two variables.
```

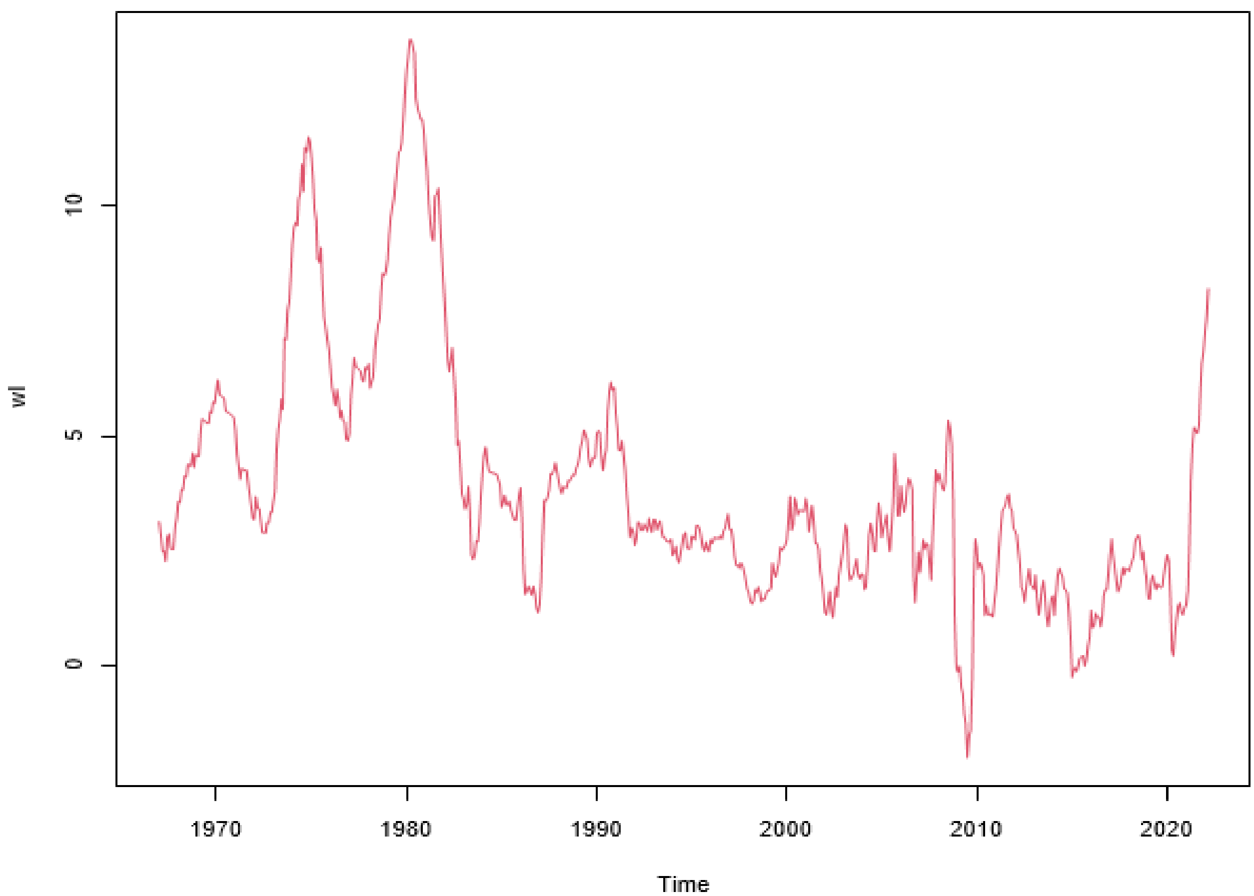
3. UNIT ROOT TESTS

Unit root tests are conducted to identify any potential unit root in the variables which will make them non-stationary. This is considered a problem in statistical modelling because the presence of unit root makes the time series stochastic. In this case the times series would be non-stationary, the mean and the variance will explode over time, making difficult to individuate potential causal relationships.

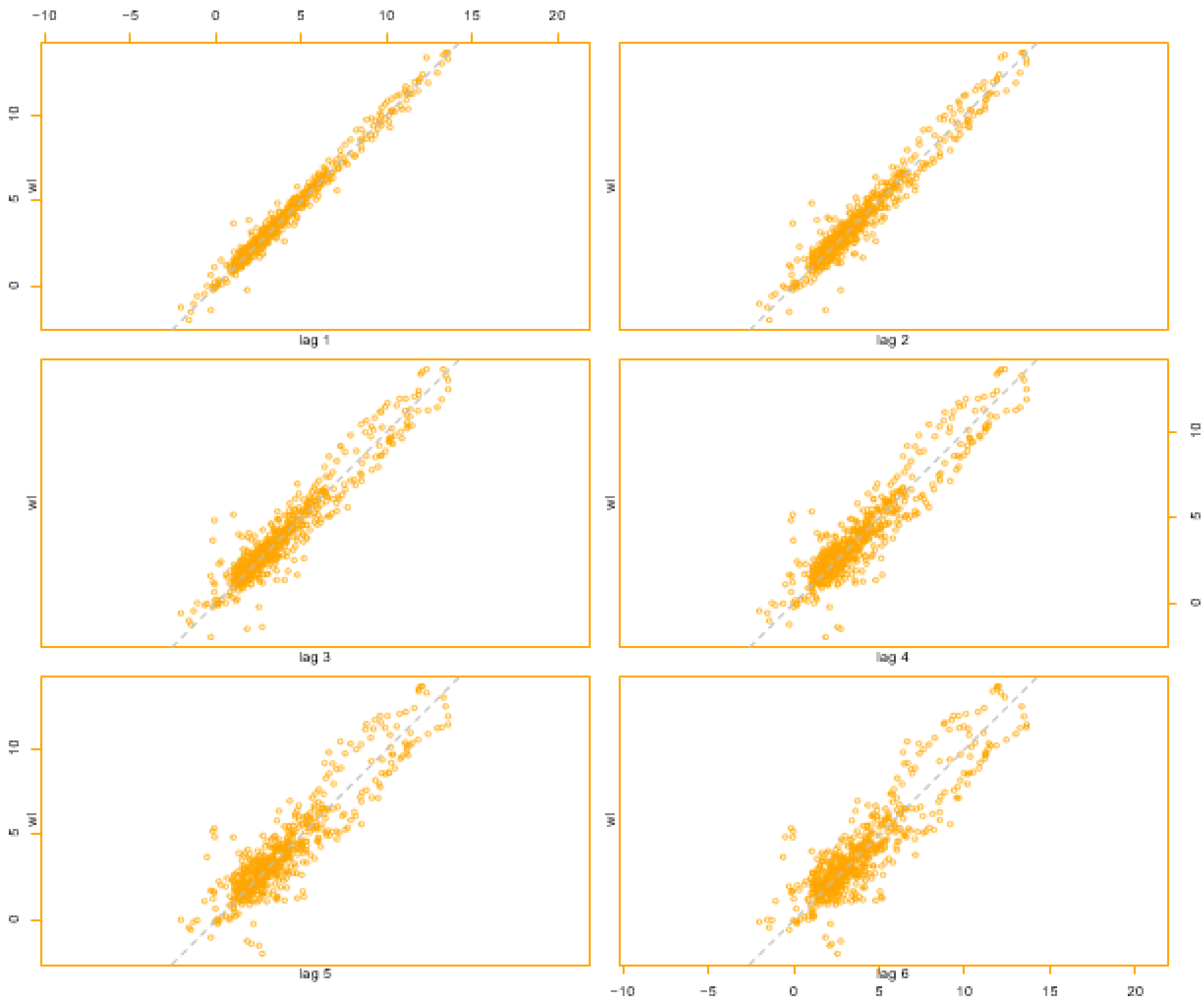
3.1 Model with variables in levels

3.1.1 Inflation

```
wl = Inflation

plot(wl, col=2)
```

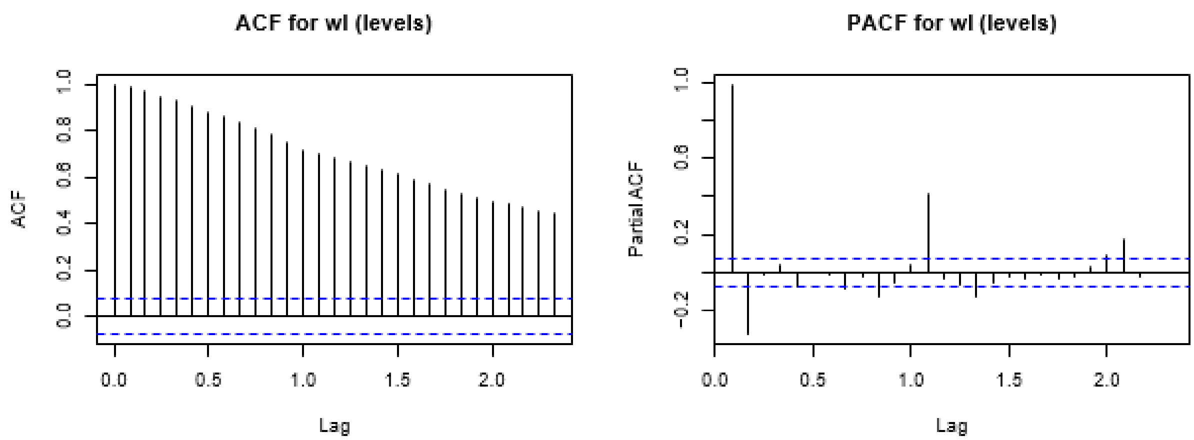



```
#The next plot shows the correlations in levels between the value of Inflation at
#time t and the the first 6 lags respectively:

lag.plot(wl, 6, do.lines=FALSE, col="orange")
```



The graph shows that there is a significant persistence suggesting long term correlation. In other words, the series suggests the presence of strong autocorrelation over time. This might imply the presence of a unit root.

```
#Next it is computed the auto-correlation functions (ACF) and the partial auto-
#correlation functions (PACF). The dashed lines indicate the bounds for statistical
#significance at the 10% level.

par(mfrow=c(2,2))
acf     (wl,         main="ACF for wl (levels)")
pacf    (wl,         main="PACF for wl (levels)")
```



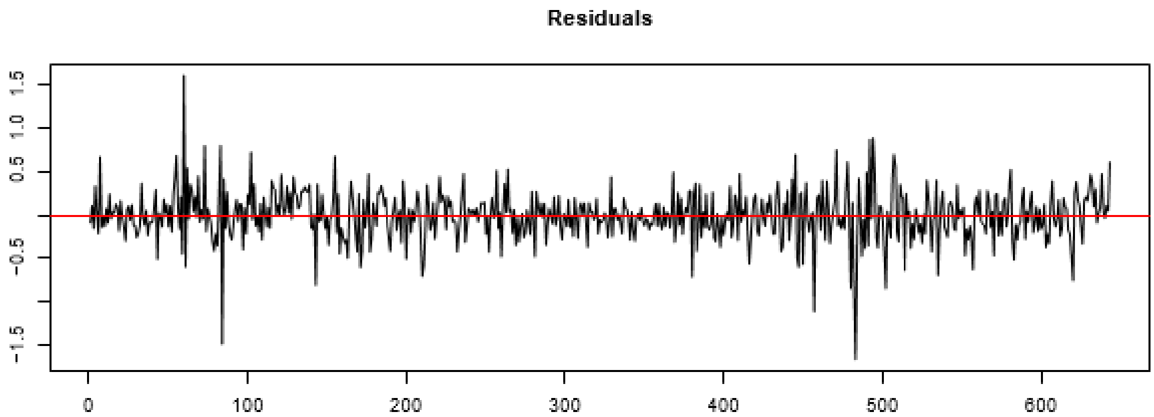
Note a very strong persistence given by the first graph, while the second graph suggests that inflation in levels might be a autoregressive process of order I[1].

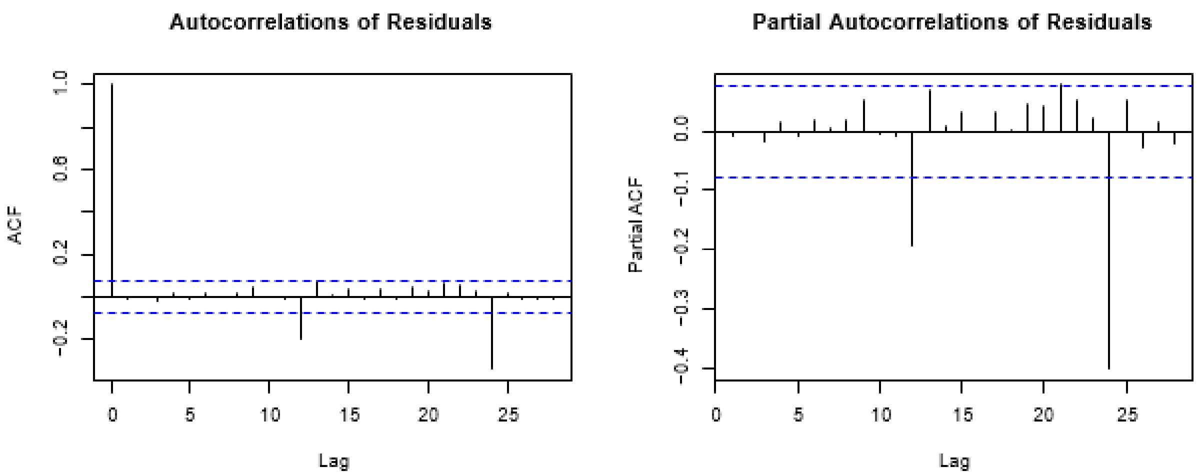
```
#Next step is conducting the three Unit Root tests (ADF, PP, KPSS) to determine
whether inflation #in levels have a process of unit root. The optimal maximum
number of lags are calculated using #the following formula:

max.lags = trunc(  (12 * (  (length(wl)/100)^(1/4)   ) ) )
max.lags

## [1] 19
```

Augmented Dickey-Fuller (ADF) test

```
#ADF: Ho = Residuals have a unit root = Non-stationary

wl.adf.drift <- ur.df(wl, selectlags="BIC", type="drift", lags=max.lags )
summary(wl.adf.drift)

##
## ###############################################
## # 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
```

```
## -1.6647 -0.1577  0.0023  0.1633  1.6169
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.053224   0.021344   2.494 0.012900 *
## z.lag.1      -0.012932   0.004633  -2.791 0.005411 **
## z.diff.lag1   0.461735   0.039411  11.716  < 2e-16 ***
## z.diff.lag2  -0.003637   0.043448  -0.084 0.933313
## z.diff.lag3   0.039029   0.043134   0.905 0.365905
## z.diff.lag4   0.035328   0.037133   0.951 0.341778
## z.diff.lag5   0.017493   0.036812   0.475 0.634817
## z.diff.lag6  -0.013282   0.036805  -0.361 0.718321
## z.diff.lag7   0.079936   0.036744   2.175 0.029968 *
## z.diff.lag8  -0.022462   0.036842  -0.610 0.542292
## z.diff.lag9   0.070908   0.036772   1.928 0.054265 .
## z.diff.lag10  0.069312   0.036817   1.883 0.060219 .
## z.diff.lag11  0.135759   0.036913   3.678 0.000255 ***
## z.diff.lag12 -0.537218   0.037414 -14.359  < 2e-16 ***
## z.diff.lag13  0.144639   0.043157   3.351 0.000852 ***
## z.diff.lag14 -0.012447   0.043530  -0.286 0.775026
## z.diff.lag15  0.154626   0.039972   3.868 0.000121 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.293 on 626 degrees of freedom
## Multiple R-squared:  0.431,  Adjusted R-squared:  0.4164
## F-statistic: 29.63 on 16 and 626 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -2.7912 3.9323
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78

#In this case, the null hypothesis is not rejected at the 5% as t-stat is -2.79, so the series is non-stationary and has a unit root.

plot(wl.adf.drift)
```

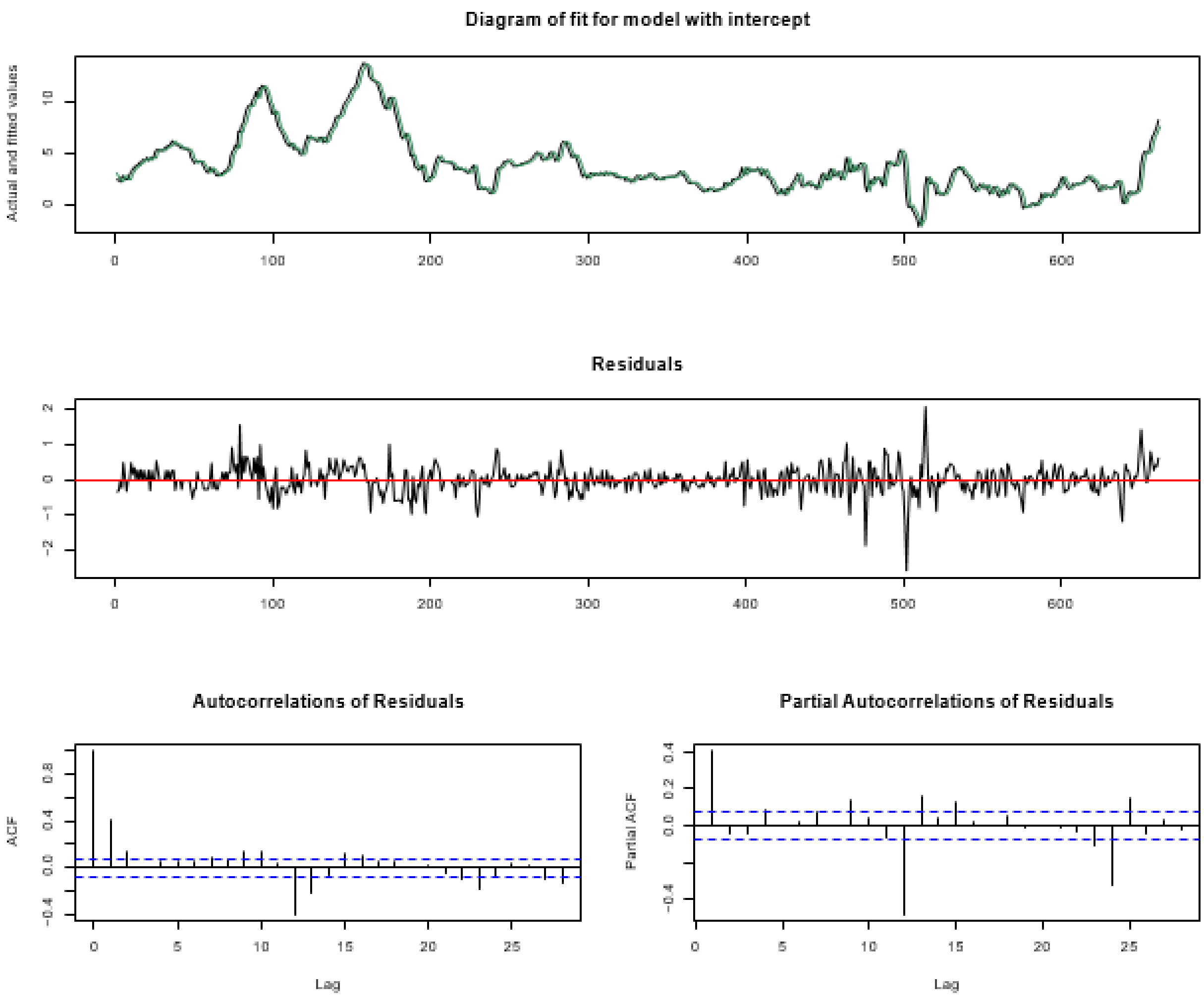




Philips-Perron (PP) test

```
#PP: Ho = Residuals have a unit root = Non-stationary

wl.pp <- ur.pp(wl, type="Z-tau", model="constant", lags="long")
summary(wl.pp)

##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
```

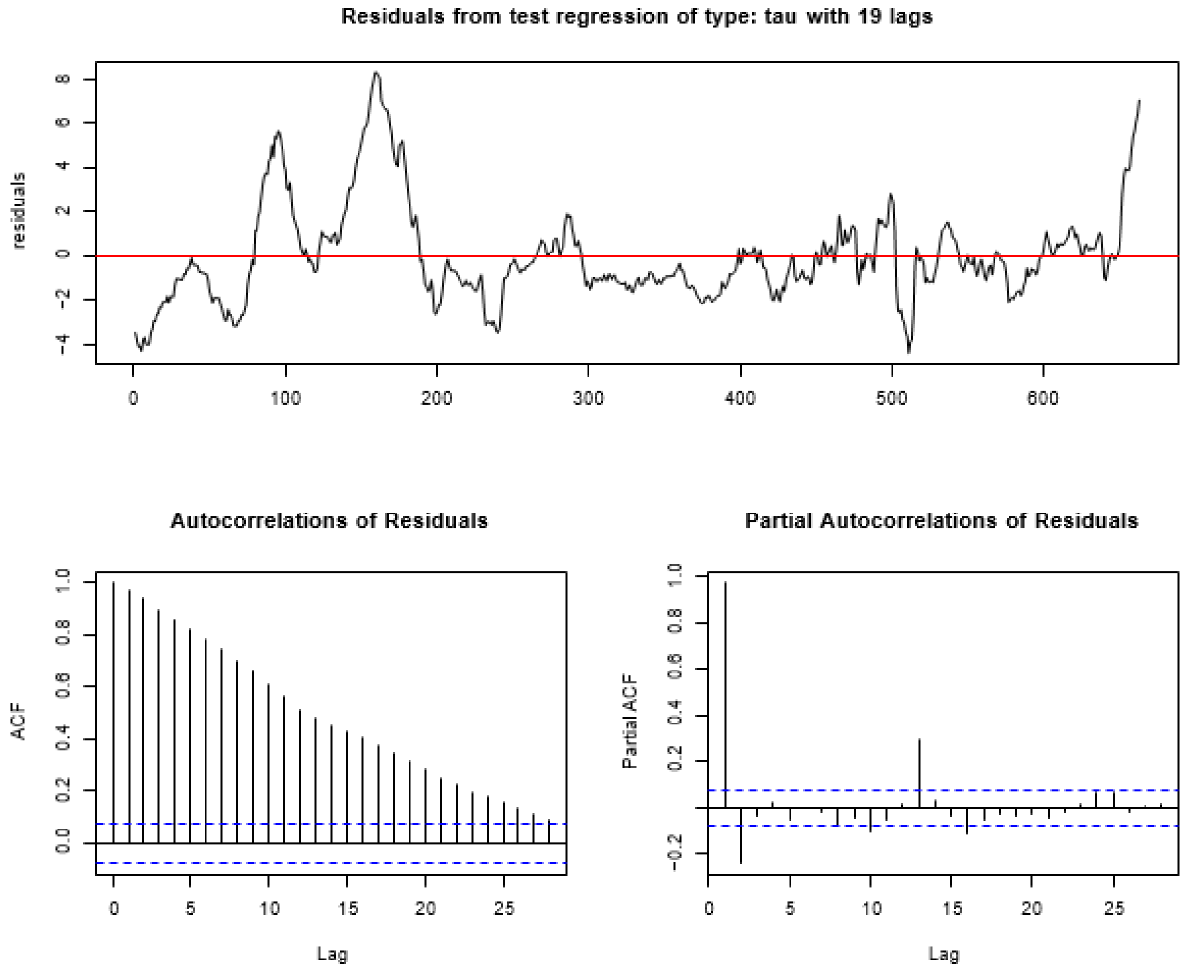
```
##      Min       1Q   Median       3Q      Max
## -2.57851 -0.21179 -0.01974  0.20530  2.08039
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.038532   0.025779   1.495    0.135
## y.l1        0.992015   0.005461 181.653   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3802 on 660 degrees of freedom
## Multiple R-squared:  0.9804, Adjusted R-squared:  0.9804
## F-statistic: 3.3e+04 on 1 and 660 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -2.5704
##
##           aux. Z statistics
## Z-tau-mu            2.2978
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.442629 -2.866255 -2.569282

#The t-stat is -2.57, which fails to reject the null hypothesis of non-stationarity
at the 5%.

plot(wl.pp)
```



KPSS test:

```
#Deterministic components: constant “mu”; or a constant with linear trend “tau”.
#KPSS: Ho = Residuals do not have a unit root = Stationary

wl.kpss <- ur.kpss(wl, type="mu",  lags="long" )
summary(wl.kpss)

## 
## ####################### 
## # KPSS Unit Root Test # 
## ####################### 
## 
## Test is of type: mu with 19 lags. 
## 
## Value of test-statistic is: 1.5785 
## 
## Critical value for a significance level of: 
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
wl.kpss <- ur.kpss(wl, type="tau", lags="long" )
print(summary(wl.kpss))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 19 lags.
##
## Value of test-statistic is: 0.1514
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216

#With a constant the null hypothesis of no unit root is rejected at any
#significance level; with tau it is also reject at 5% but on the borderline (t-stat
#0.15>0.146).

plot(wl.kpss)
```

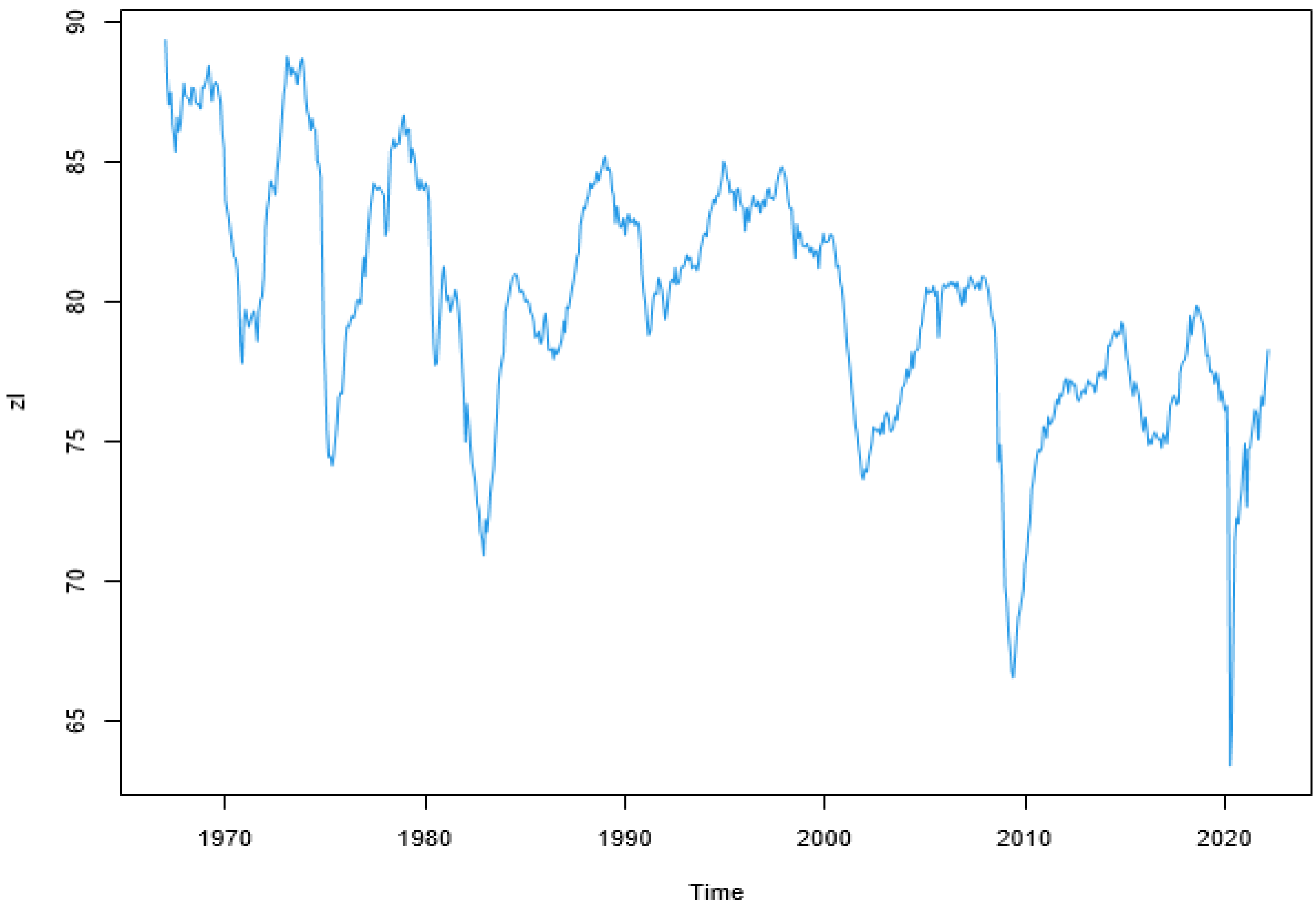


Summary

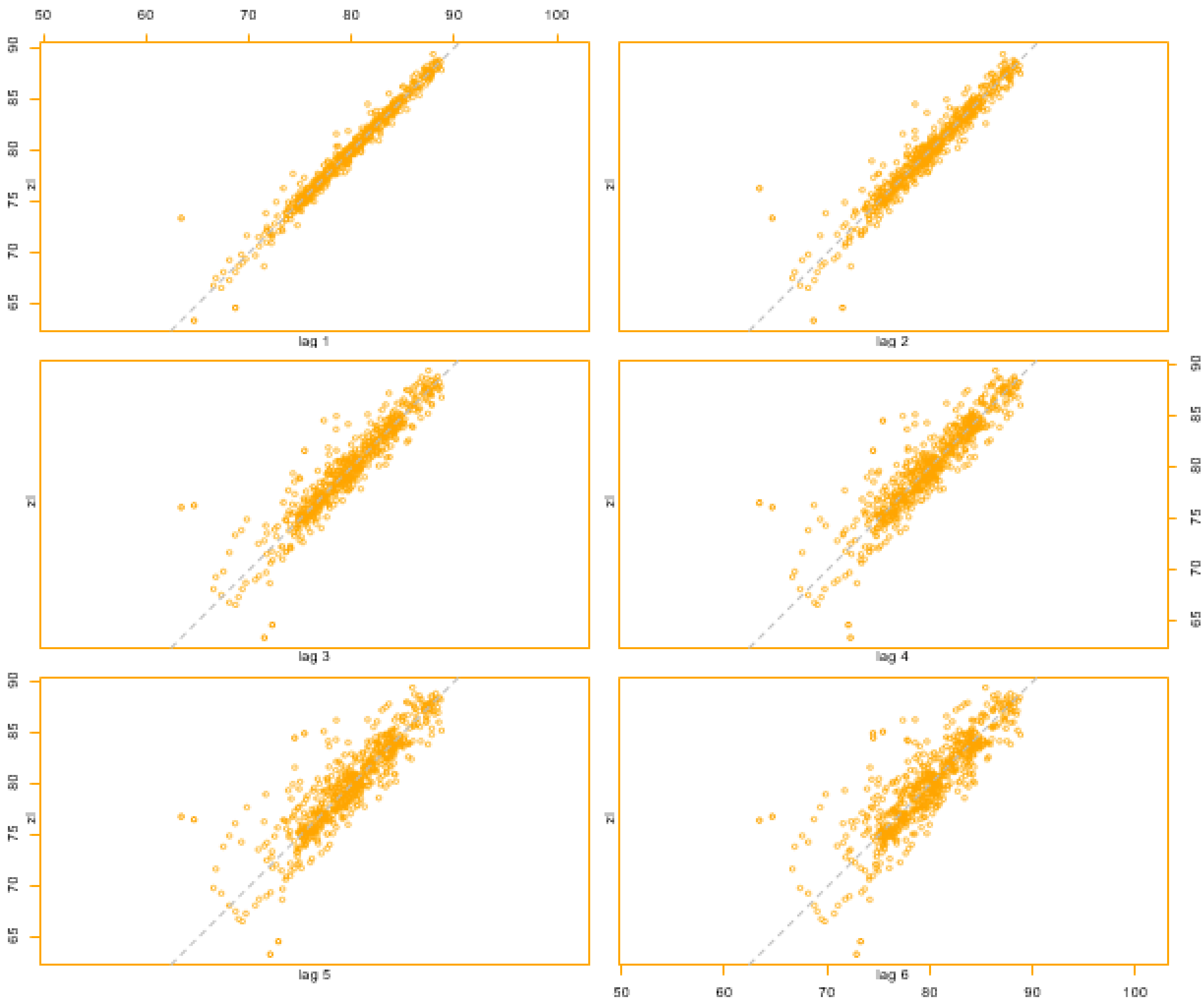
There are evidence that the residuals of the Inflation Time Series have a unit root and the series is not-stationary I(1) in levels.

3.1.2 TCU

```
zl = TCU

#Plot the chosen variable:

plot(zl, col = 4)
```

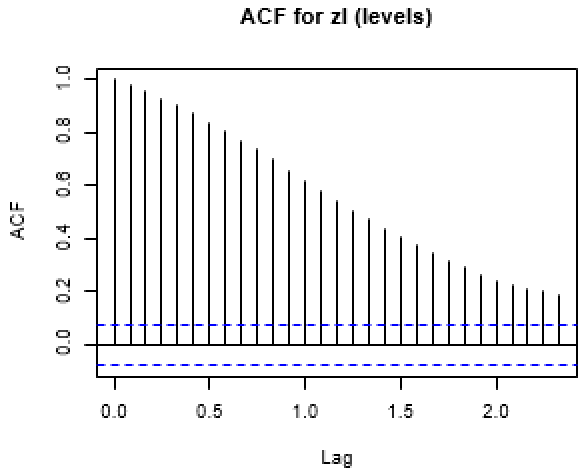


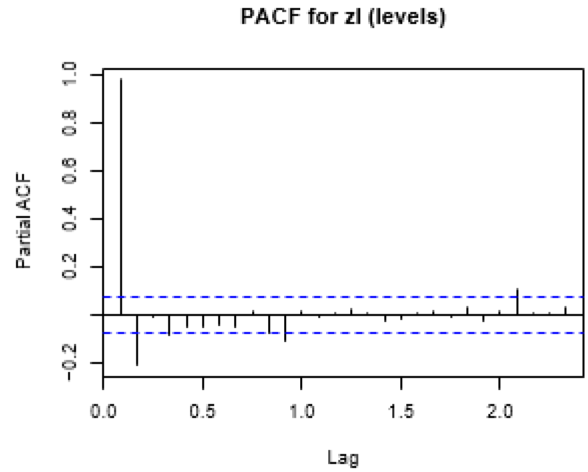

```
#The next plot shows the correlations in levels between the value of TCU at time t
#and the first 6 lags respectively:

lag.plot(zl, 6, do.lines=FALSE, col="orange")
```



Once again, the graph shows that there is a significant persistence and the presence of strong autocorrelation over time. This might suggest a unit root process.

```
#The auto-correlation functions (ACF) and the partial auto-correlation functions
#(PACF) are shown below:

par(mfrow=c(2,2))
acf     (zl,          main="ACF for zl (levels)")
pacf    (zl,          main="PACF for zl (levels)")
```





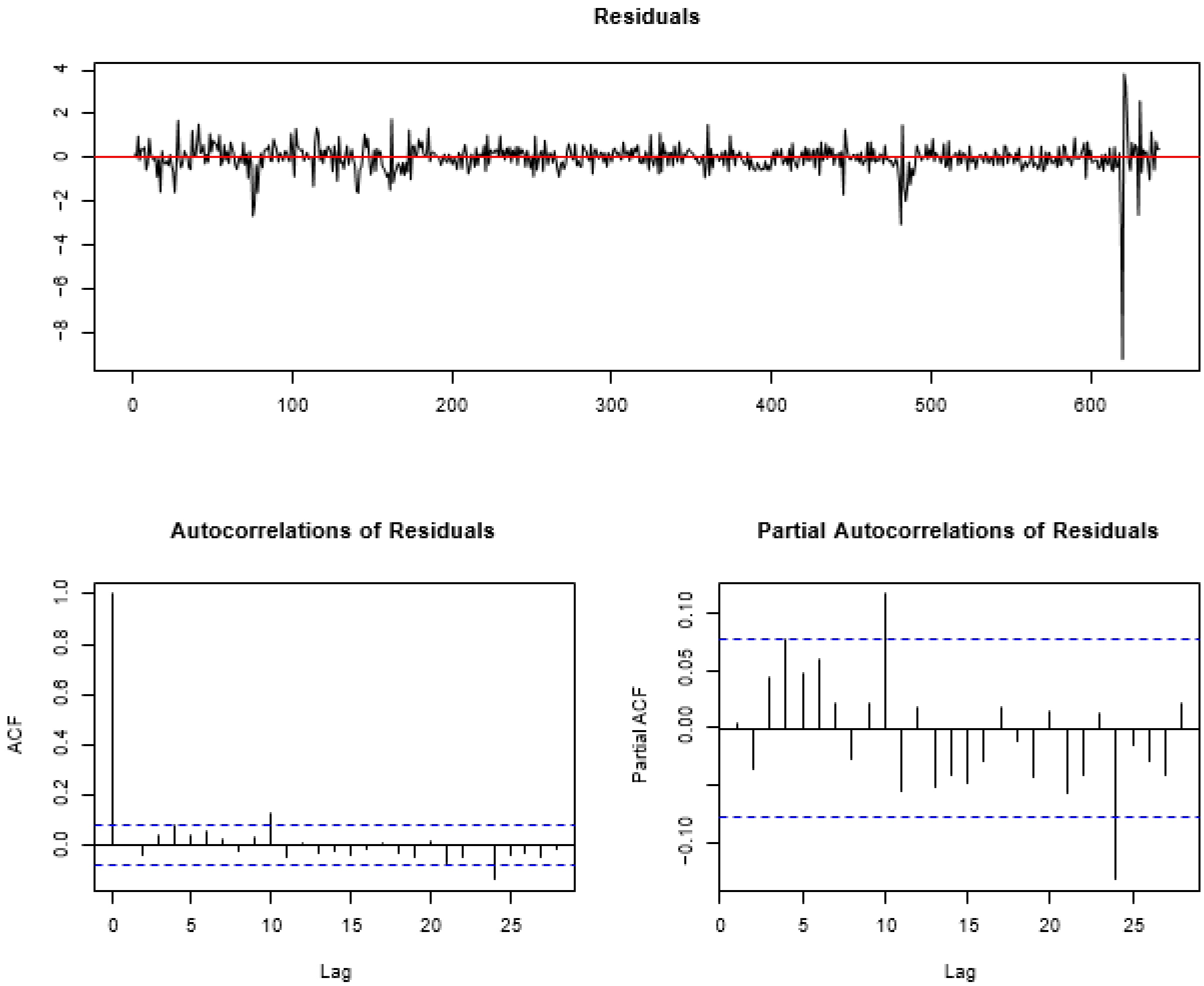
In the same way as in the first time series, there is a very strong persistence in the residuals, and the second graph suggests that TCU in levels might be a autoregressive process of order I[1].

```
#The Unit Root tests (ADF, PP, KPSS) are conducted to determine whether TCU in
#levels have a process of unit root. Optimal maximum lags are still 19:

 max.lags = trunc(  (12 * (  (length(wl)/100)^(1/4)  ) ) )
max.lags

## [1] 19
```

Augmented Dickey-Fuller (ADF)

```
#ADF: Ho = Residuals have a unit root = Non-stationary

zl.adf.drift <- ur.df(zl, selectlags="BIC", type="drift", lags=max.lags )
summary(zl.adf.drift)

##
## ###############################################
## # 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
## -9.2341 -0.3011  0.0271  0.3207  3.8488
##
```

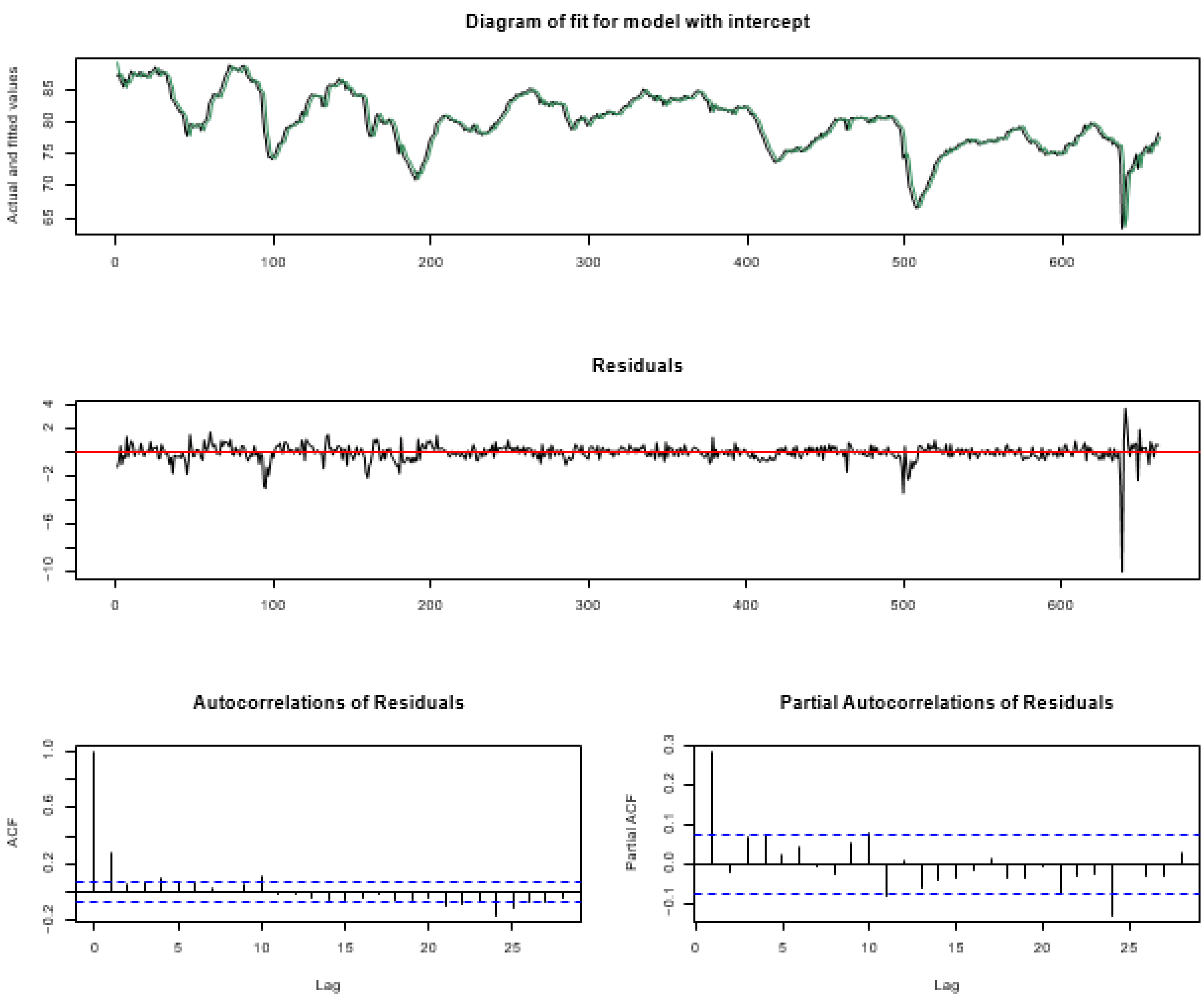
```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.693835   0.525794   3.221  0.00134 **
## z.lag.1     -0.021350   0.006581  -3.244  0.00124 **
## z.diff.lag   0.295130   0.037700   7.828 2.06e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7044 on 640 degrees of freedom
## Multiple R-squared:  0.09655,    Adjusted R-squared:  0.09372
## F-statistic:  34.2 on 2 and 640 DF,  p-value: 7.759e-15
##
##
## Value of test-statistic is: -3.244 5.3194
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78

# The null hypothesis of unit root is rejected at the 5% but fails to be rejected at the 1%.

plot(zl.adf.drift)
```



Philips-Perron (PP) test

```
#PP: Ho = Residuals have a unit root = Non-stationary

zl.pp <- ur.pp(zl, type="Z-tau", model="constant", lags="long")
summary(zl.pp)

##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

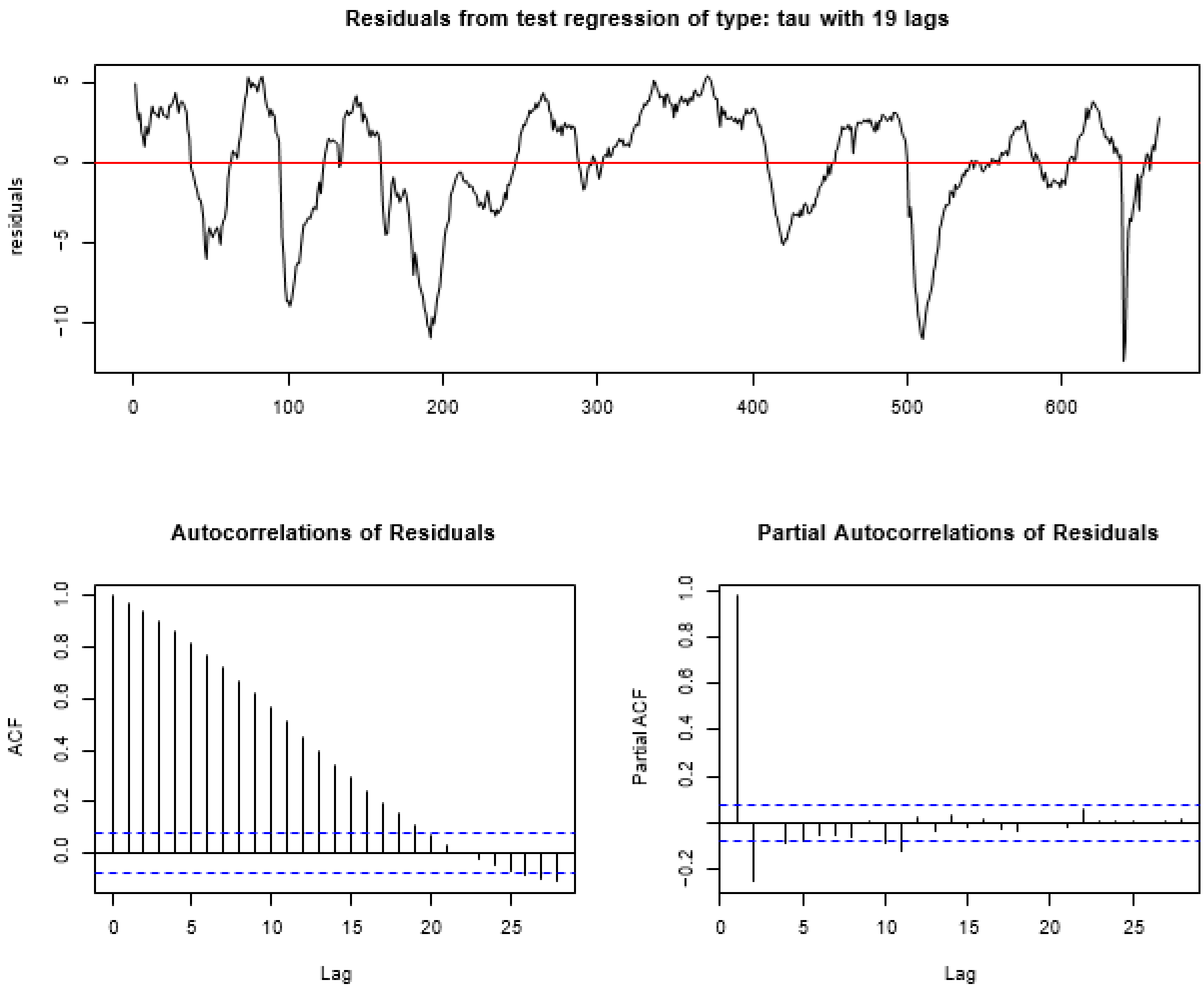
```
## -10.0680  -0.3112   0.0572   0.3570   3.7640
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.406049   0.525563   2.675  0.00765 **
## y.l1        0.982211   0.006561 149.701  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7346 on 660 degrees of freedom
## Multiple R-squared:  0.9714, Adjusted R-squared:  0.9713
## F-statistic: 2.241e+04 on 1 and 660 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -3.5943
##
##          aux. Z statistics
## Z-tau-mu            3.5676
##
## Critical values for Z statistics:
##                       1pct      5pct     10pct
## critical values -3.442629 -2.866255 -2.569282

# The PP test rejects the null hypothesis of unit root at all significance levels,
suggesting that TCU in levels is stationary.

plot(zl.pp)
```



KPSS tests on the chosen series

```
#Deterministic components: constant “mu”; or a constant with linear trend “tau”.

#KPSS: Ho = Residuals do not have a unit root = Stationary

 zl.kpss <- ur.kpss(zl, type="mu",  lags="long" )
summary(zl.kpss)

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 19 lags.
##
## Value of test-statistic is: 1.4505
##
## Critical value for a significance level of:
```

```
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

zl.kpss <- ur.kpss(zl, type="tau", lags="long" )
print(summary(zl.kpss))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 19 lags.
##
## Value of test-statistic is: 0.0948
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216



# The KPSS test rejects the null hypothesis of no unit root in the model with the
#constant at any significance levels, but do not reject it in the model with
#constant and linear trend at any significance levels.

plot(zl.kpss)
```



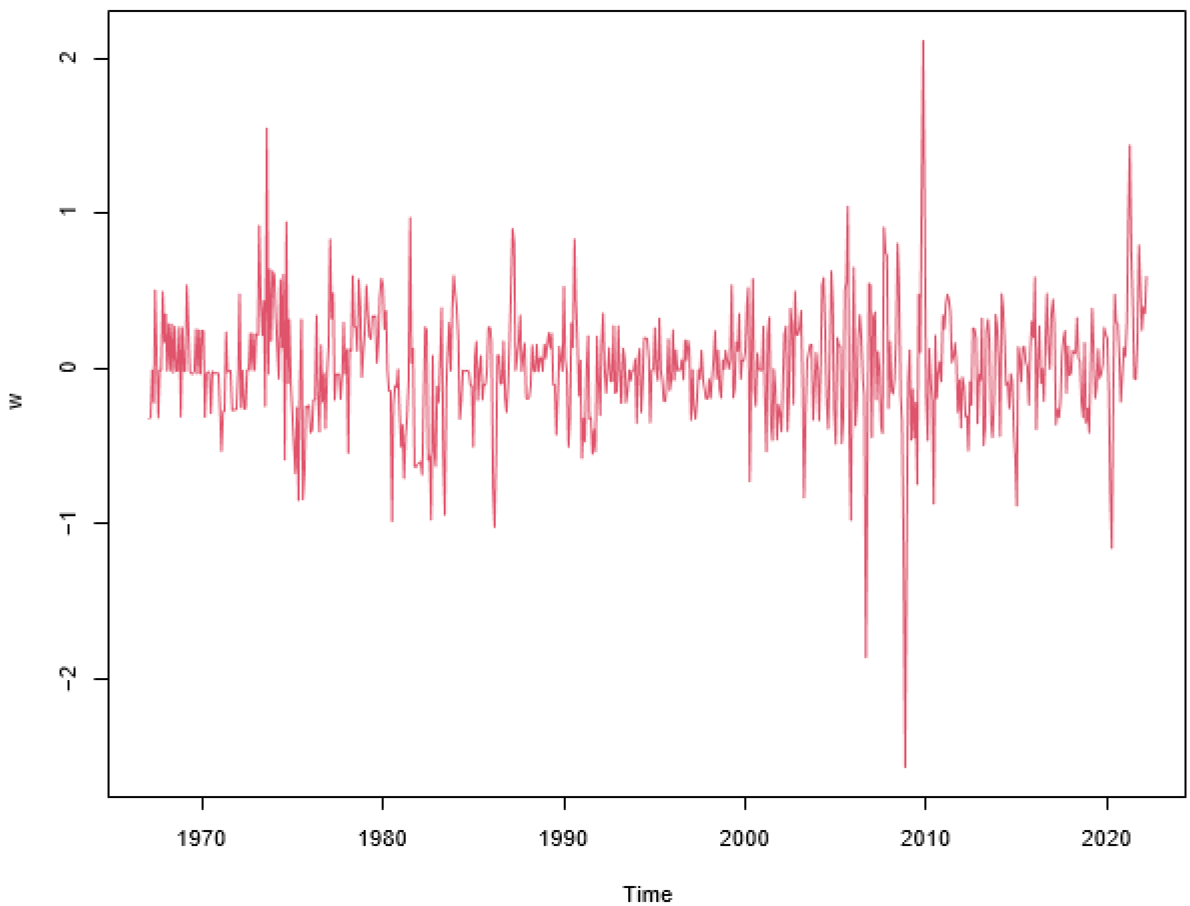
Summary

There is evidence that the residuals of the TCU Time Series in levels might have a unit root, but the results of the tests are ambiguous and inconsistent. Let us proceed by assuming that the series is not-stationary I[1] in levels.

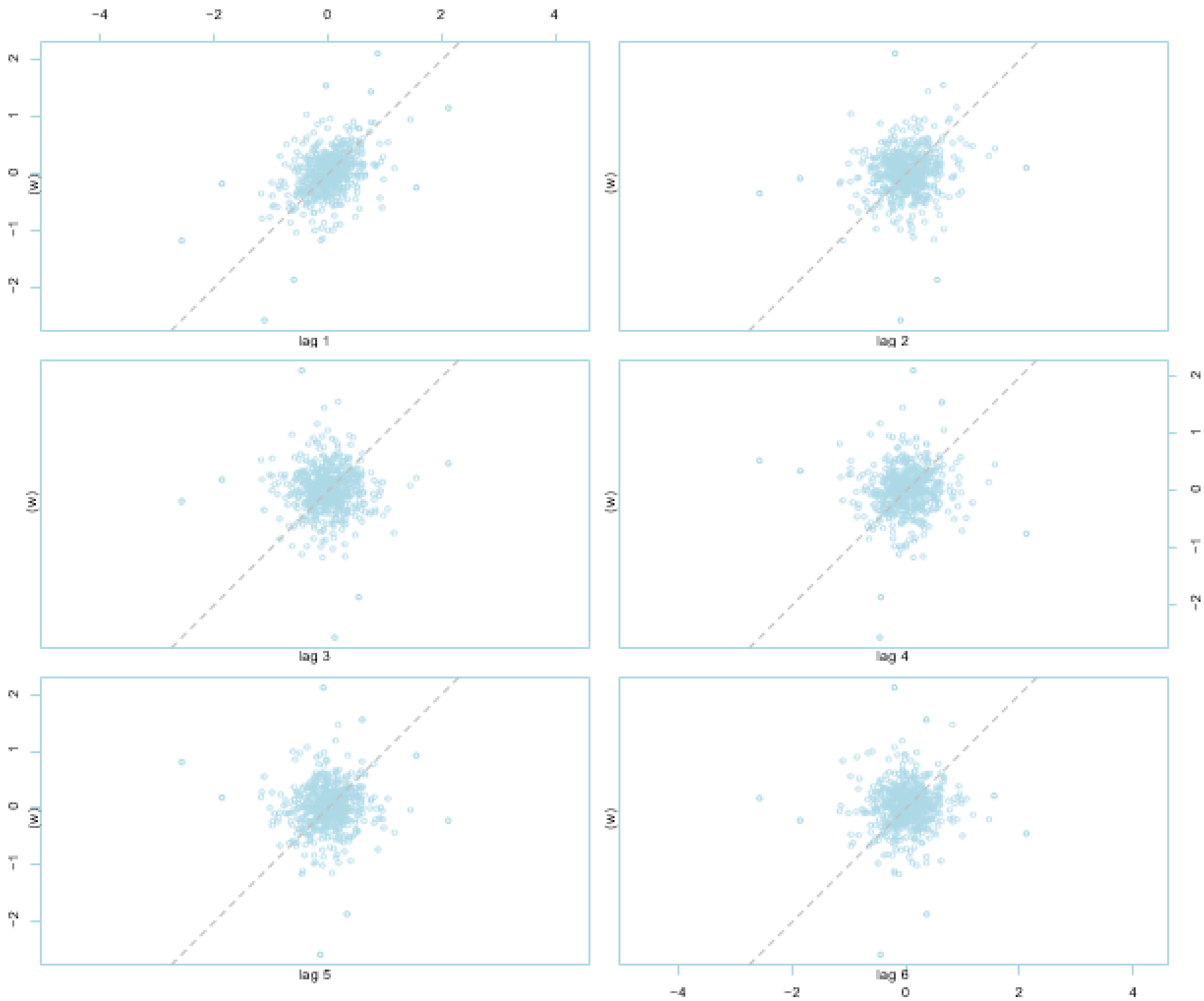
3.2 Model in first differences

3.2.1 Inflation

```
w = diff(Inflation)

#Plot of the variable:

plot(w, col = 2)
```

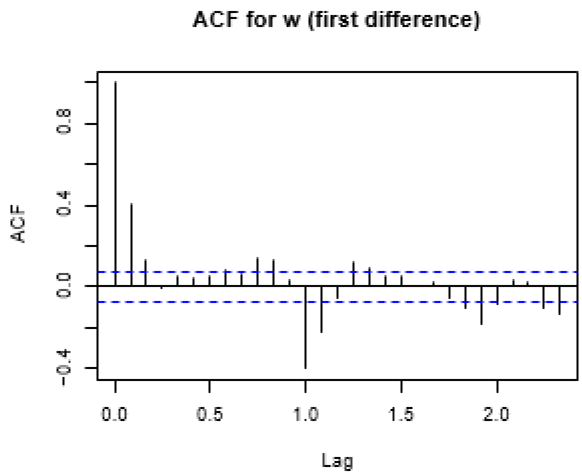


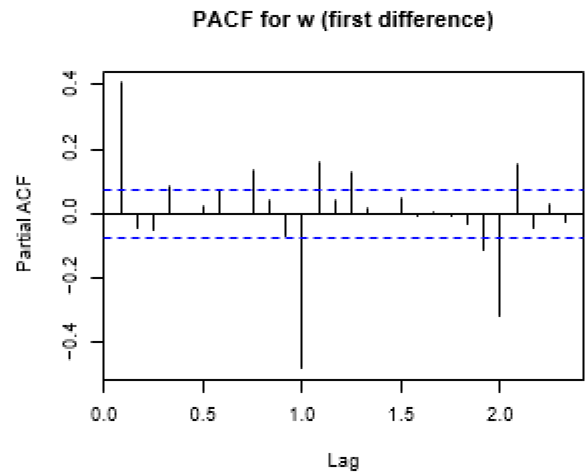

```
#Plot of the lag correlations in first differences:
lag.plot((w), 6, do.lines=FALSE, col="lightblue")
```



By running the model in first differences, the correlation is weakened, and the persistence disappears. This might imply that the process of unit root might have been removed.

```
par(mfrow=c(2,2))
acf     ((w),    main="ACF for w (first difference)")
pacf    ((w),    main="PACF for w (first difference)")
```





As it can be seen above, the autocorrelation has been reduced significantly by taking the first differences. There are still signs of autocorrelation of the residuals, but later in the paper further tests are done.

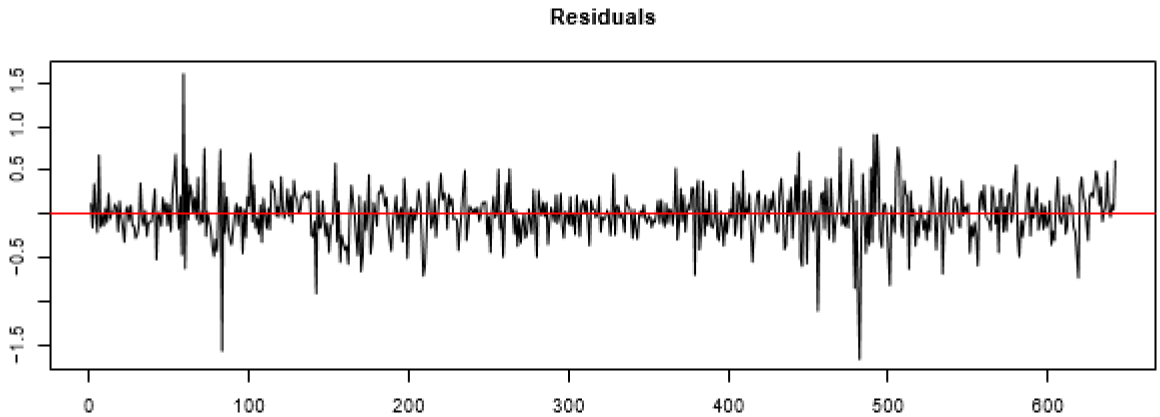
```
#Next step is to compute the three Unit Root tests (ADF, PP, KPSS) to determine
#whether Inflation in first differences has a process of unit root.

#Optimal maximum number of lags:

max.lags = trunc(  (12 * (  (length(w)/100)^(1/4)  ) ) )
max.lags

## [1] 19
```

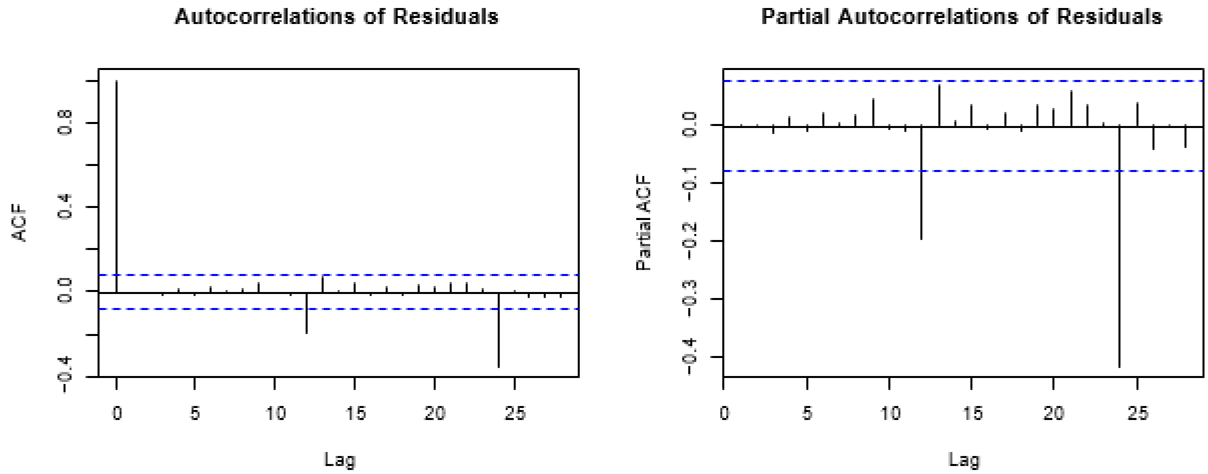
Augmented Dickey-Fuller(ADF) test

```
#ADF: Ho = Residuals have a unit root = Non-stationary

w.adf.drift <- ur.df(w, selectlags="BIC", type="drift", lags=max.lags)
print(summary(w.adf.drift))

##
## ###############################################
## # 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
## -1.65492 -0.15612  0.00057  0.16625  1.60708
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.003252   0.011638   0.279 0.780025
## z.lag.1      -0.478635   0.084014  -5.697 1.88e-08 ***
## z.diff.lag1  -0.062491   0.082403  -0.758 0.448519
## z.diff.lag2  -0.073195   0.078405  -0.934 0.350894
## z.diff.lag3  -0.043453   0.072052  -0.603 0.546677
## z.diff.lag4  -0.010724   0.071751  -0.149 0.881235
## z.diff.lag5   0.001653   0.070501   0.023 0.981305
## z.diff.lag6  -0.016958   0.068452  -0.248 0.804414
## z.diff.lag7   0.056639   0.065761   0.861 0.389409
## z.diff.lag8   0.027800   0.063396   0.439 0.661161
## z.diff.lag9   0.091425   0.060079   1.522 0.128581
## z.diff.lag10  0.153569   0.056928   2.698 0.007172 **
## z.diff.lag11  0.280657   0.053739   5.223 2.40e-07 ***
## z.diff.lag12 -0.266698   0.050319  -5.300 1.61e-07 ***
## z.diff.lag13 -0.126680   0.045966  -2.756 0.006023 **
## z.diff.lag14 -0.144420   0.040078  -3.603 0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2948 on 626 degrees of freedom
## Multiple R-squared:  0.5097, Adjusted R-squared:  0.4979
## F-statistic: 43.38 on 15 and 626 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -5.6971 16.2508
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```
# As expected, the null hypothesis of unit root is rejected at any significance level.

plot(w.adf.drift)
```

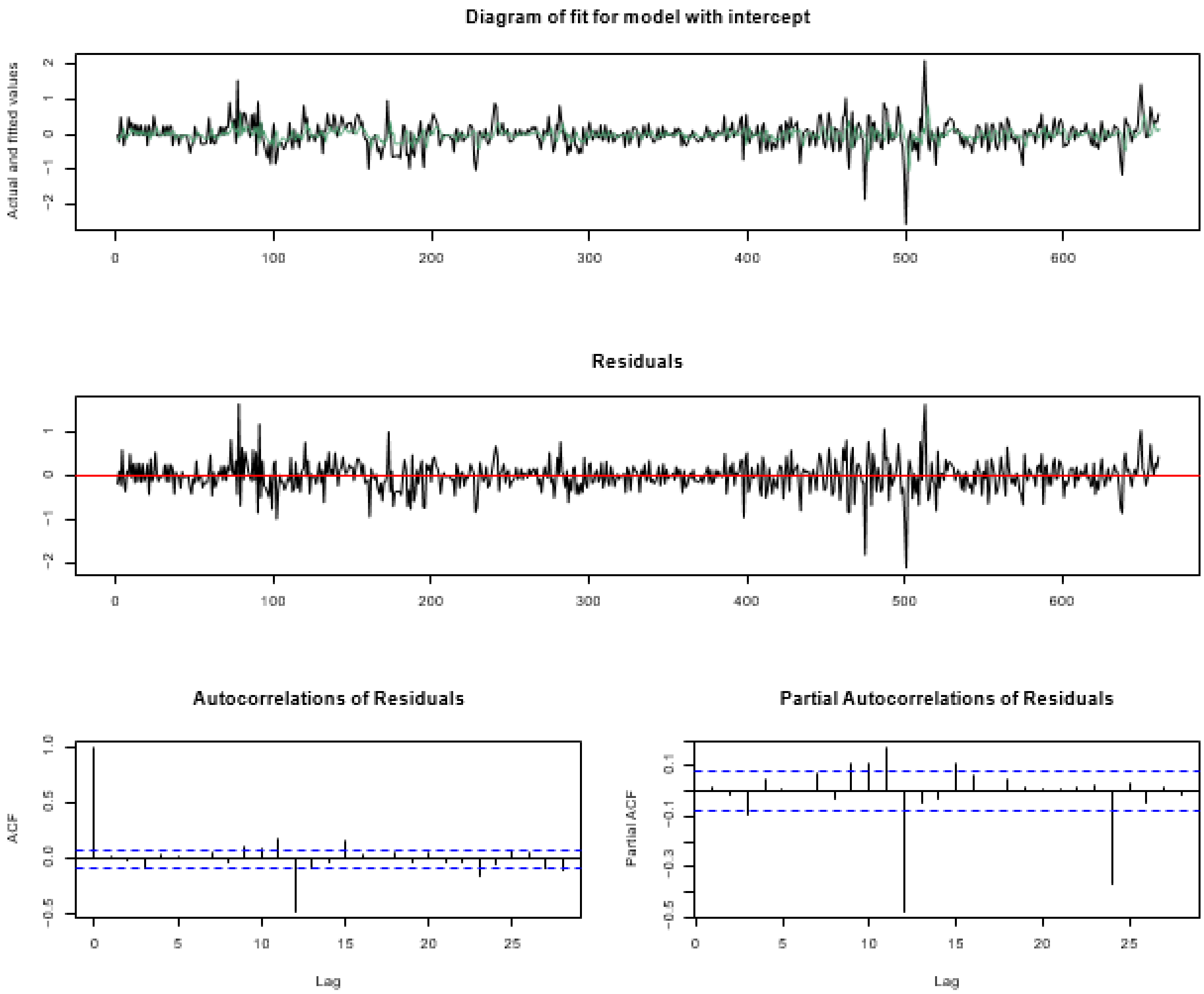




Philips-Perron (PP) test

```
#PP: Ho = series has a unit root = integrated I(1) process

w.pp <- ur.pp(w, type="Z-tau", model="constant", lags="long")
summary(w.pp)

##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

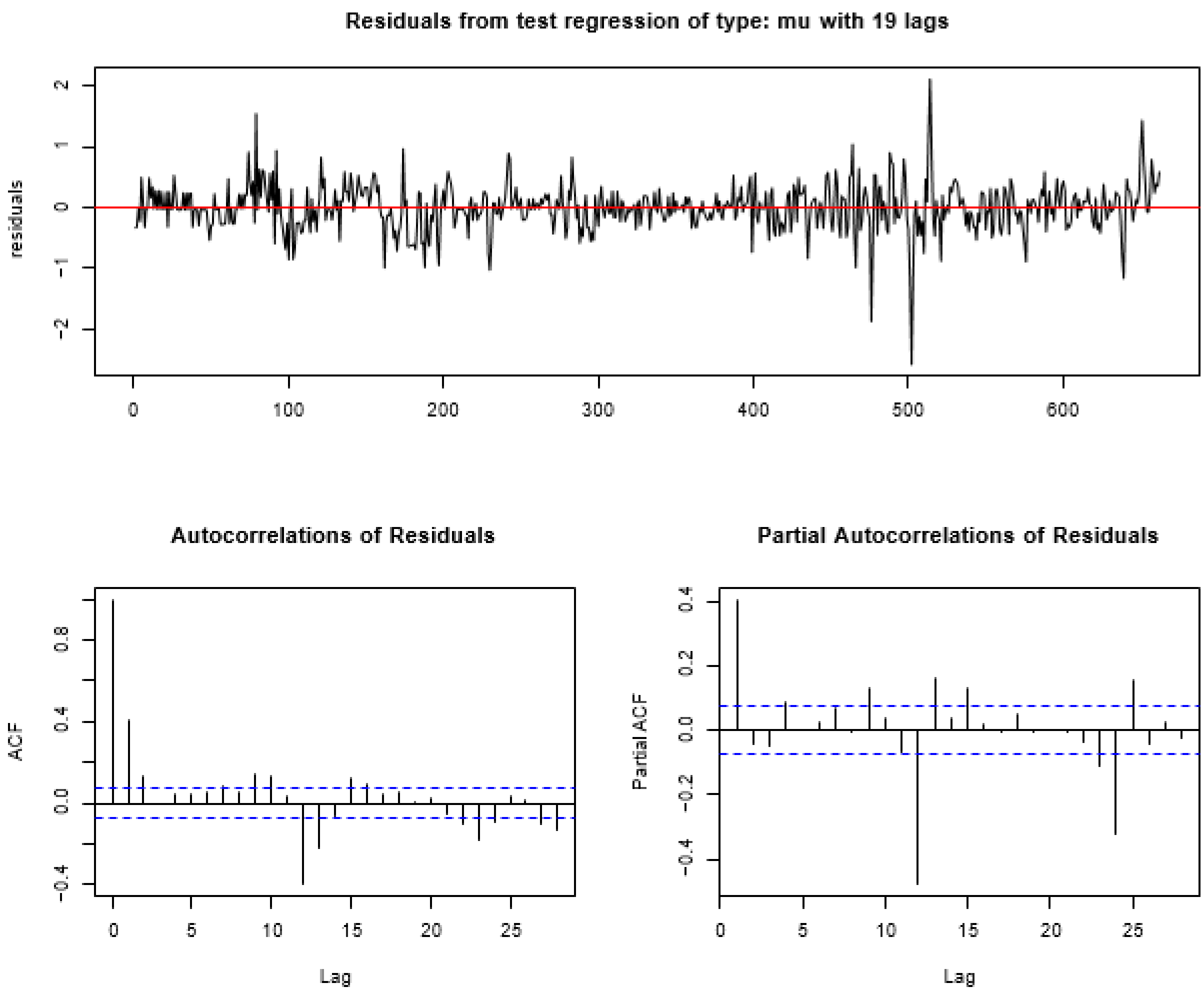
```
## -2.0967 -0.1717  0.0004  0.1844  1.6472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.005388   0.013530   0.398    0.691
## y.l1        0.407999   0.035618  11.455   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3478 on 659 degrees of freedom
## Multiple R-squared:  0.166,  Adjusted R-squared:  0.1648
## F-statistic: 131.2 on 1 and 659 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -16.7923
##
##          aux. Z statistics
## Z-tau-mu            0.4009
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.442643 -2.866261 -2.569286

#The PP test rejects the null hypothesis of unit root at any significance level.

plot(w.pp)
```



KPSS test

```
#Deterministic components: constant “mu”; or a constant with linear trend “tau”.

#KPSS: Ho = series does not have a unit root = Stationary

 w.kpss1 <- ur.kpss(w, type="mu",  lags="long" )
summary(w.kpss1)

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 19 lags.
##
## Value of test-statistic is: 0.0775
##
```

```
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

plot(w.kpss1)
```




```
w.kpss2 <- ur.kpss(w, type="tau", lags="long" )
print(summary(w.kpss2))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 19 lags.
##
## Value of test-statistic is: 0.065
##
```

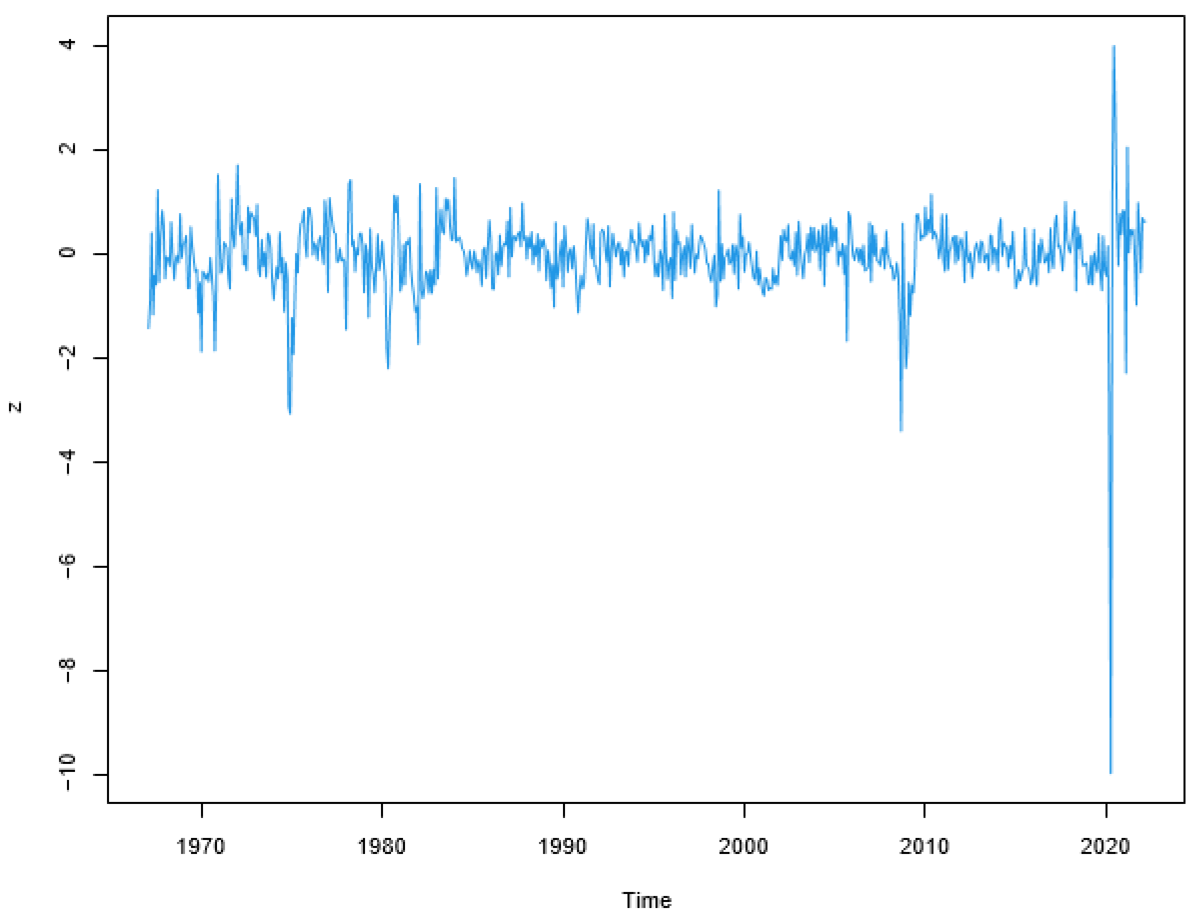
```
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216

# The KPSS test fails to reject the null hypothesis of stationarity in accordance
with the previous results.
```

Summary

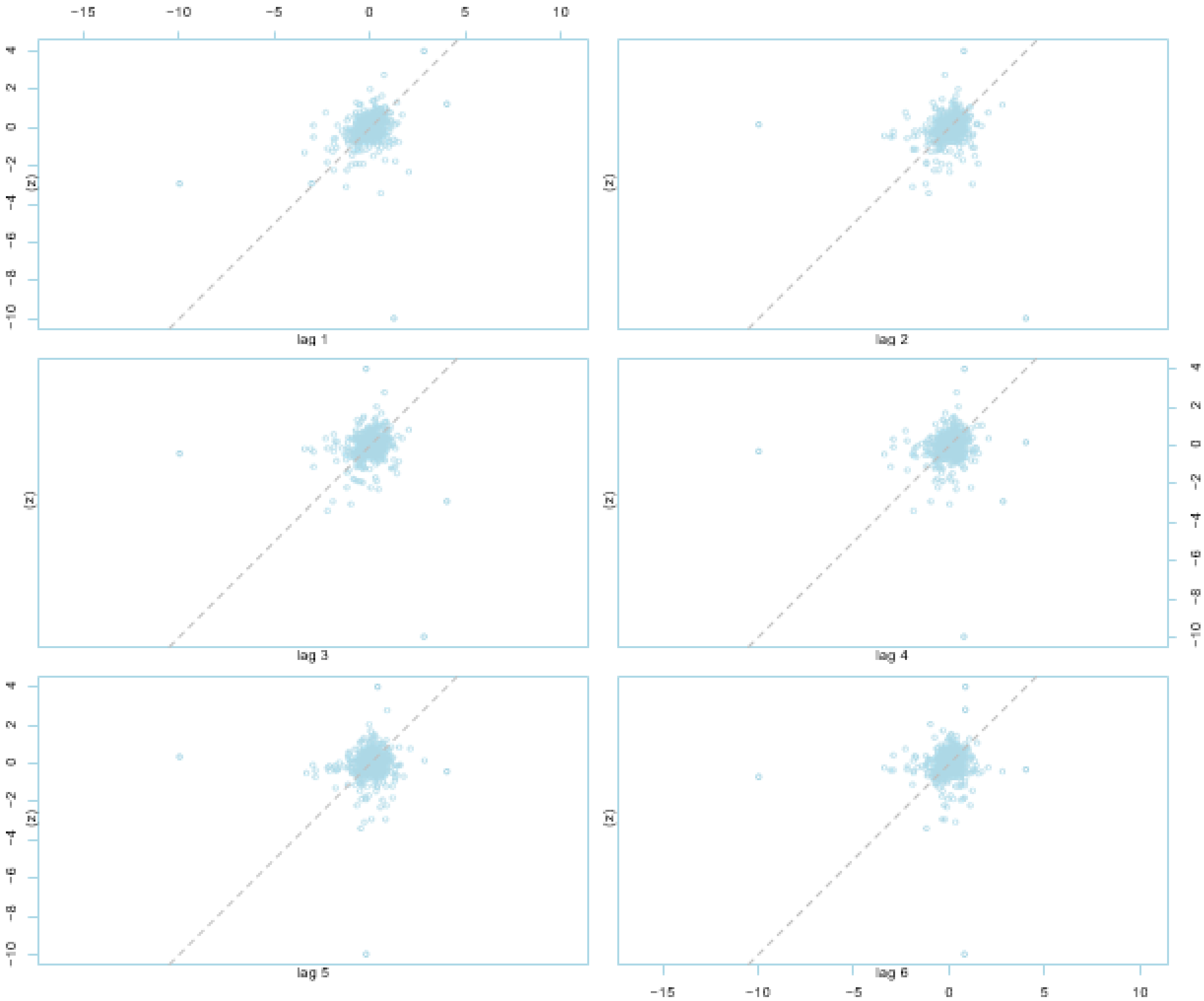
Inflation in first differences is stationary I[0].

3.2.2 TCU

```
z = diff(TCU)

#Plot of the chosen variable:

 plot(z, col = 4)
```

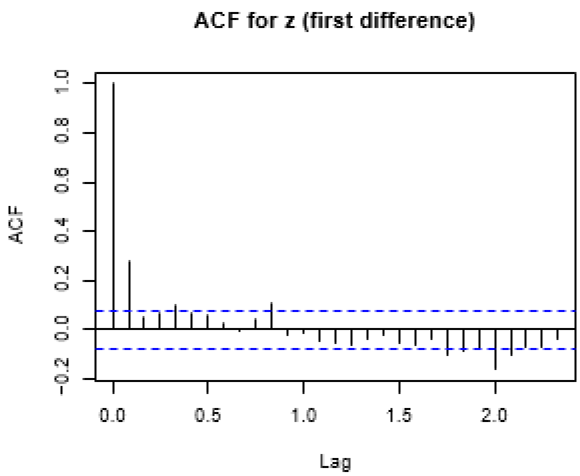


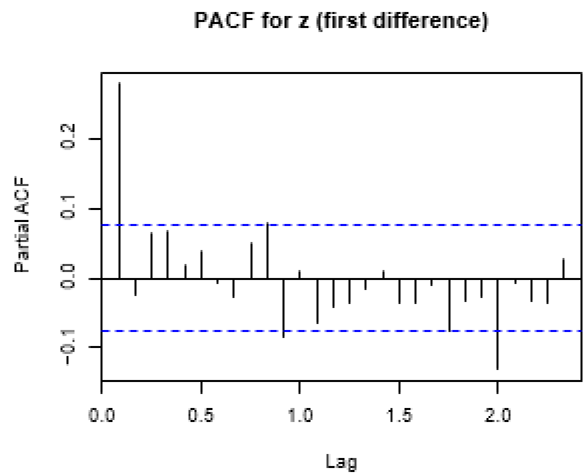
```
#Plot of the lag correlations in first differences:
lag.plot((z), 6, do.lines=FALSE, col="lightblue")
```



Once again, by running the model in first differences, the correlation is weakened, and the persistence disappears. This might imply that the process of unit root might have been removed.

```
par(mfrow=c(2,2))
acf     ((z),    main="ACF for z (first difference)")
pacf    ((z),    main="PACF for z (first difference)")
```





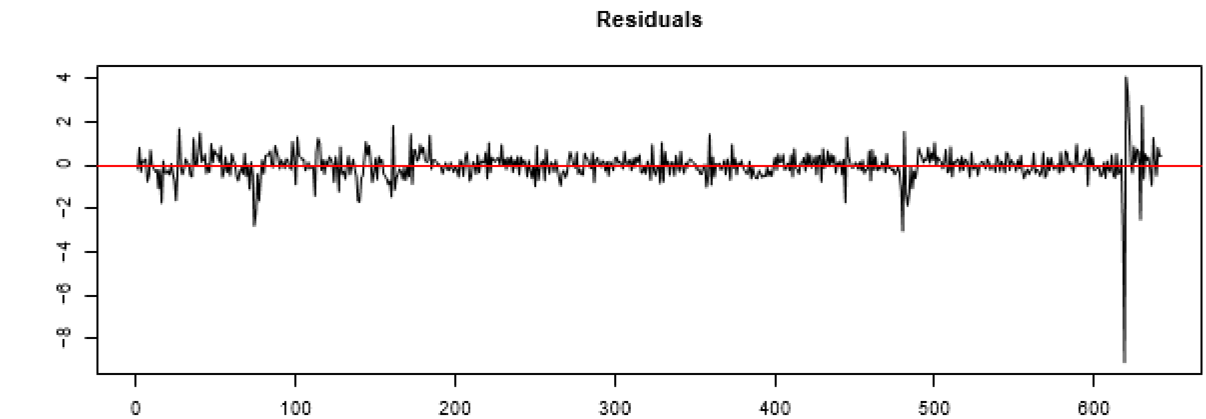
As shown above, the autocorrelation has been reduced significantly by taking the first differences. There are still signs of potential autocorrelation of the residuals, but later in the paper further tests are done.

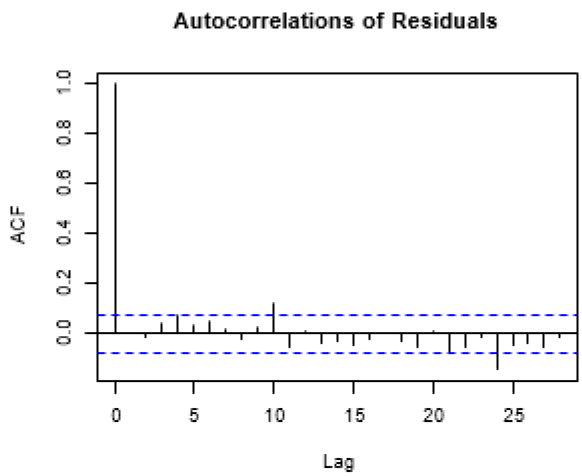
```
#Finally, the Unit Root tests (ADF, PP, KPSS) to determine whether TCU in first
#differences has a process of unit root. Optimal number of maximum lags:

max.lags = trunc(  (12 * (  (length(z)/100)^(1/4)  ) ) )
max.lags

## [1] 19
```

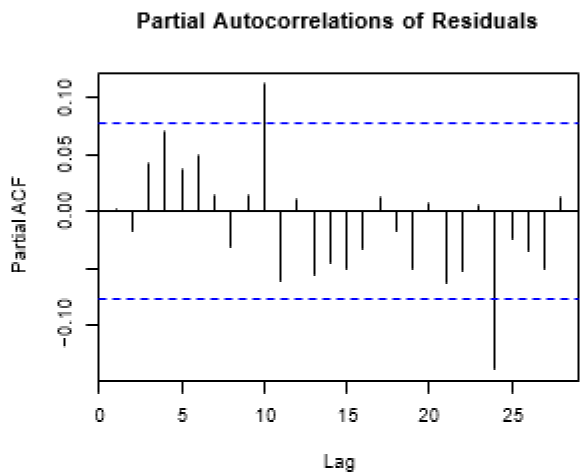
Augmented Dickey-Fuller(ADF) test

```
#ADF: Ho = Residuals have a unit root = Non-stationary

 z.adf.drift <- ur.df(z, selectlags="BIC", type="drift", lags=max.lags)
print(summary(z.adf.drift))

##
## ###############################################
## # 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
## -9.0963 -0.3009  0.0078  0.2997  4.1054
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -0.009808   0.028050  -0.350    0.727
## z.lag.1     -0.733657   0.047321 -15.504   <2e-16 ***
## z.diff.lag   0.027480   0.039582   0.694    0.488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7105 on 639 degrees of freedom
## Multiple R-squared:  0.3572, Adjusted R-squared:  0.3552
## F-statistic: 177.5 on 2 and 639 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -15.5039 120.1864
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78

# Even in this test the null hypothesis of unit root is rejected at any
significance level.

plot(z.adf.drift)
```



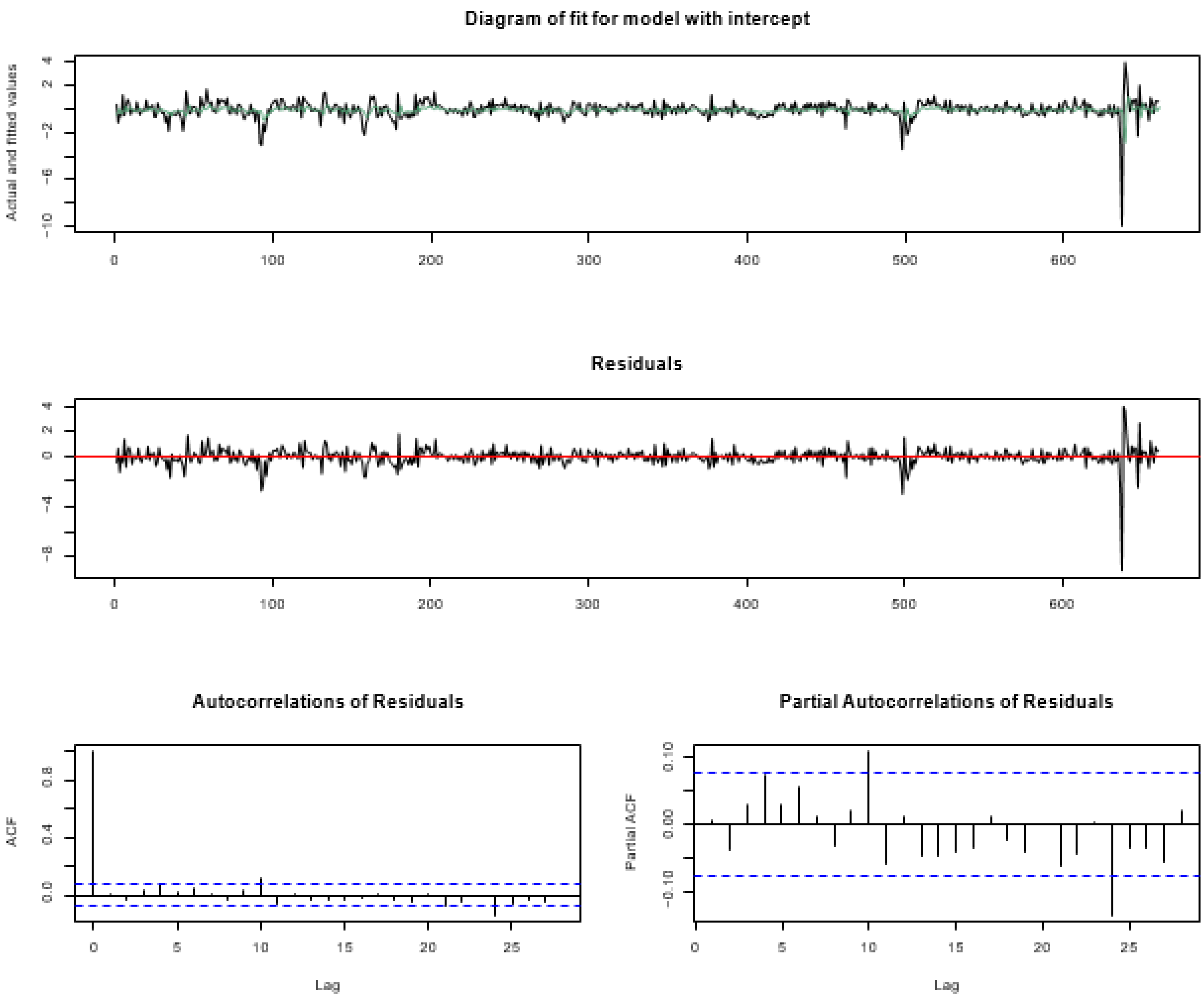




Philips-Perron (PP) test

```
#PP: Ho = Residuals have a unit root = Non-stationary

z.pp <- ur.pp(z, type="Z-tau", model="constant", lags="long")
summary(z.pp)

##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -9.1445 -0.3005  0.0098  0.3025  4.0372
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.00968    0.02754  -0.352    0.725
## y.l1         0.27896    0.03732   7.474 2.48e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7078 on 659 degrees of freedom
## Multiple R-squared:  0.07815,    Adjusted R-squared:  0.07675
## F-statistic: 55.87 on 1 and 659 DF,  p-value: 2.479e-13
##
##
## Value of test-statistic, type: Z-tau  is: -19.9491
##
##          aux. Z statistics
## Z-tau-mu           -0.3671
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.442643 -2.866261 -2.569286

# As expected, in the PP test the null hypothesis of unit root is rejected at any significance level.

plot(z.pp)
```



KPSS test

```
#Deterministic components: constant "mu"; or a constant with linear trend "tau".

#KPSS: Ho = Residuals do not have a unit root = Stationary

z.kpss <- ur.kpss(z, type="mu",  lags="long" )
summary(z.kpss)

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 19 lags.
##
## Value of test-statistic is: 0.0364
##
## Critical value for a significance level of:
```

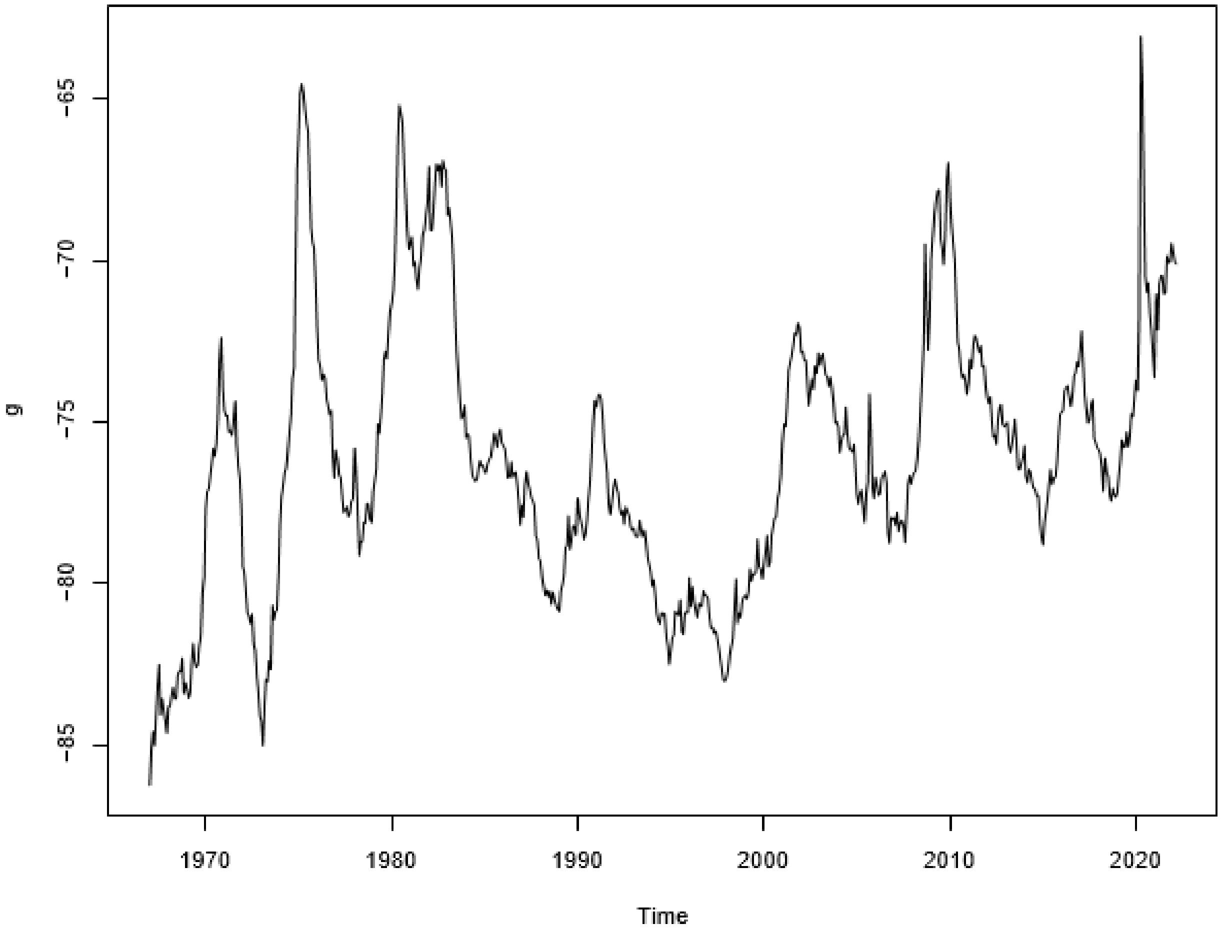
```
##                   10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

z.kpss <- ur.kpss(z, type="tau", lags="long" )
print(summary(z.kpss))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 19 lags.
##
## Value of test-statistic is: 0.0212
##
## Critical value for a significance level of:
##                   10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216

# Even the KPSS test fails to reject the null hypothesis of stationarity, which
confirm the previous results.
```

Summary

TCU in first differences is stationary I[0].

4. COINTEGRATION ANALYSIS

```
#Next, plot of the graph of the gap between the two series:

g=Inflation - TCU
plot(g)
```



The two series are not expected to be cointegrated as do not show a clear common trend, but after having discovered that both time series show signs of unit root in levels, but are stationary in first differences, the next step required is to test whether cointegration in the residuals exists. It could be found that there might be a vector of the linear combination of the two series whose residuals might be stationary, which would suggest the presence of cointegration.

4.1 Optimal Lag Selection For Cointegration tests

```
#First, let us check the optimal lag length according to the information criteria:

#Using the "VARS" package
VARselect(data.set, lag.max=10, type="none",  season = NULL, exogen =
NULL)$selection

## AIC(n)  HQ(n)  SC(n) FPE(n)
##      3      2      2      3

VARselect(data.set, lag.max=10, type="const", season = NULL, exogen =
NULL)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2

VARselect(data.set, lag.max=10, type="trend", season = NULL, exogen =
NULL)$selection

## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2

VARselect(data.set, lag.max=10, type="both",  season = NULL, exogen =
NULL)$selection

## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2

#According to the Hannan-Quinn criterion, the optimal number of lags is 2.

#Set the number of lags that will be used in our estimations
optimal.lags = 2
```

4.2 Engle-Granger Methodology

```
#In the Engle-Granger Methodology the Ho = Residuals are no-cointegrated.

coint.test(Inflation, TCU,           d = 0, nlag = NULL, output = TRUE)

## Response: Inflation
## Input: TCU
## Number of inputs: 1
## Model: y ~ X + 1
## -------------------------------
## Engle-Granger Cointegration Test
## alternative: cointegrated
##
## Type 1: no trend
##     lag      EG p.value
##  6.0000 -2.6428  0.0876
## -----
##  Type 2: linear trend
##     lag      EG p.value
##  6.0000  0.0228  0.1000
## -----
##  Type 3: quadratic trend
##     lag      EG p.value
##    6.00   -0.48    0.10
## -----------
## Note: p.value = 0.01 means p.value <= 0.01
##     : p.value = 0.10 means p.value >= 0.10

#The p-value of the Engle-Granger Cointegration test is lower than 1% rejecting the
null hypothesis of no cointegration.
```

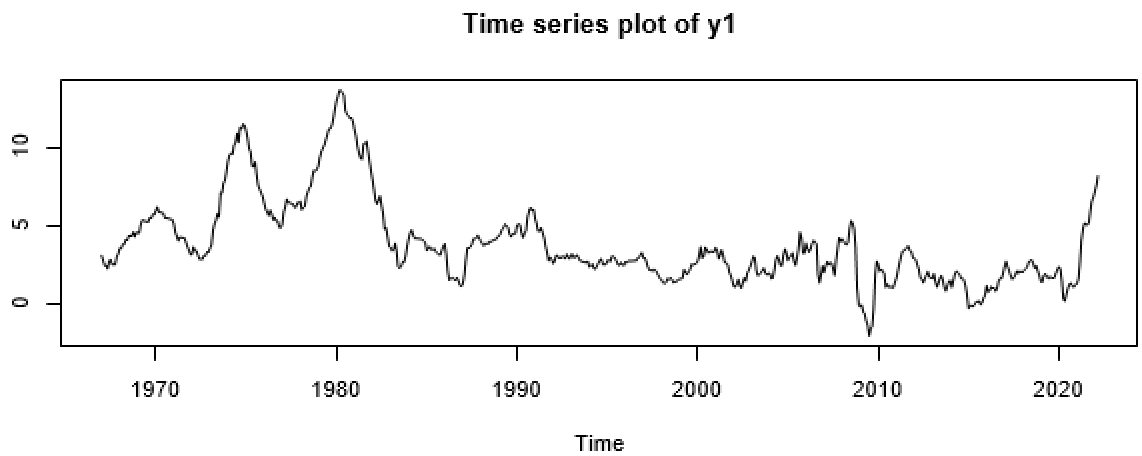
Summary

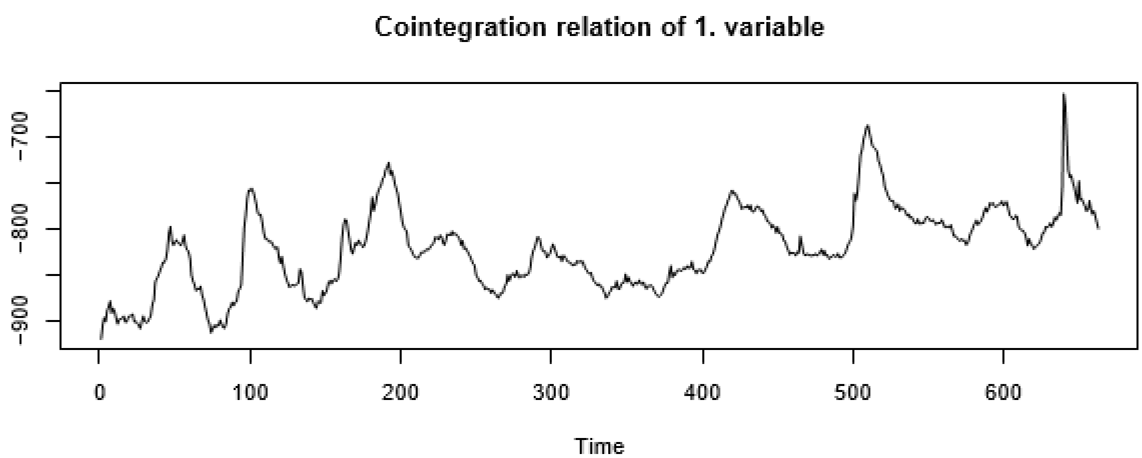
Inflation and TCU seem to be cointegrated in levels.

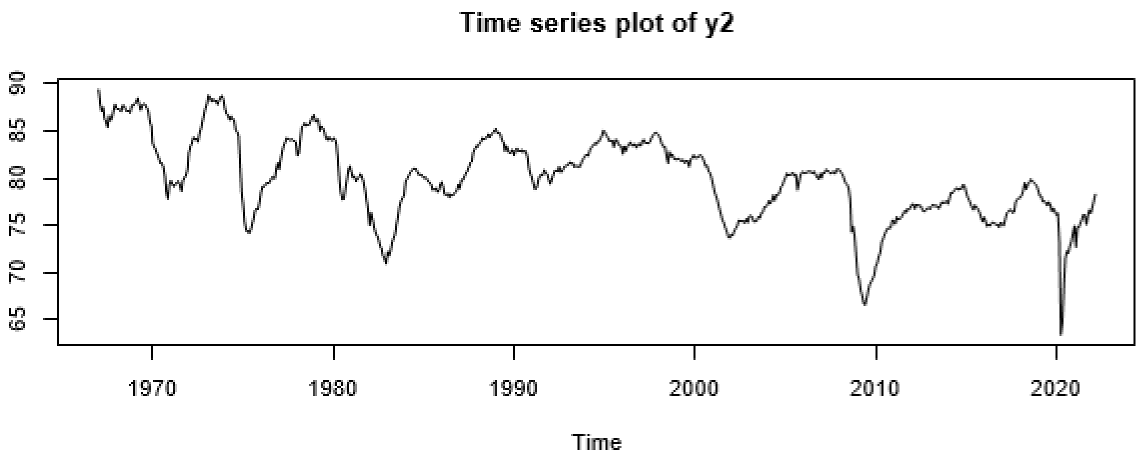
4.3 Johansen Test

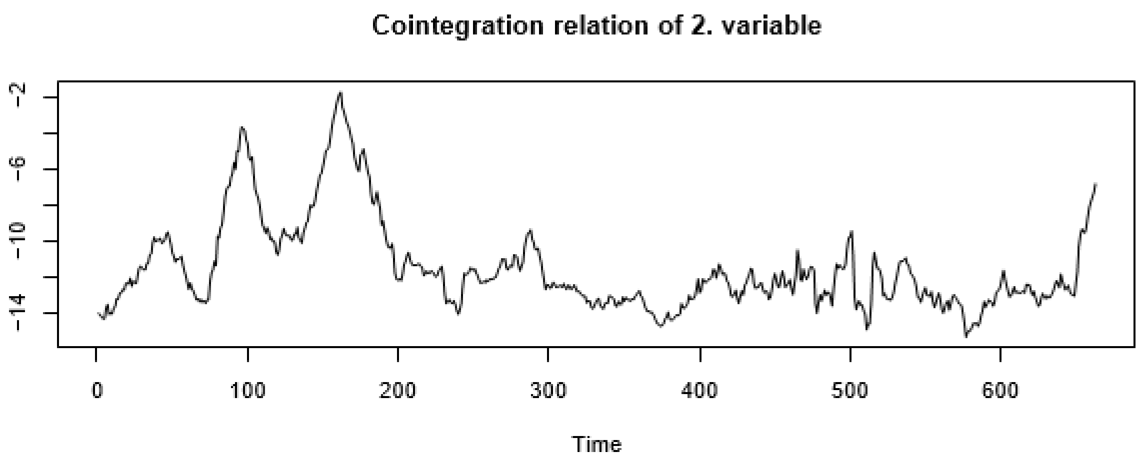
```
#In the Johansen test the null hypothesis is Ho = no cointegration (r=0).

#With constant in the cointegrating vector:
johansen.const = ca.jo(data.set, type="eigen", ecdet="const",  K = optimal.lags,
spec="longrun")
summary(johansen.const)

## 
## ###################### 
## # Johansen-Procedure # 
## ###################### 
## 
## Test type: maximal eigenvalue statistic (lambda max) , without linear trend and
constant in cointegration 
## 
## Eigenvalues (lambda):
## [1] 2.943962e-02 2.049640e-02 3.469447e-18
## 
## Values of teststatistic and critical values of test:
## 
##           test 10pct  5pct  1pct
## r <= 1 | 13.69  7.52  9.24 12.97
## r = 0  | 19.75 13.75 15.67 20.20
## 
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
## 
##                Inflation.l2     TCU.l2   constant
## Inflation.l2      1.00000  1.0000000   1.000000
## TCU.l2          -10.31221 -0.1915511  -1.891966
## constant        814.48319 11.3603475 197.589867
## 
## Weights W:
## (This is the loading matrix)
## 
##                Inflation.l2      TCU.l2      constant
## Inflation.d -0.0006959788 -0.01709435  1.116010e-19
## TCU.d        0.0021677903 -0.02522300 -1.054726e-17

johansen.const = ca.jo(data.set, type="trace", ecdet="const",  K = optimal.lags,
spec="longrun")
summary(johansen.const)
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1] 2.943962e-02 2.049640e-02 3.469447e-18
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 | 13.69  7.52  9.24 12.97
## r = 0  | 33.44 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                Inflation.l2     TCU.l2   constant
## Inflation.l2      1.00000  1.0000000   1.000000
## TCU.l2          -10.31221 -0.1915511  -1.891966
## constant        814.48319 11.3603475 197.589867
##
## Weights W:
## (This is the loading matrix)
##
##                Inflation.l2      TCU.l2      constant
## Inflation.d -0.0006959788 -0.01709435  1.116010e-19
## TCU.d        0.0021677903 -0.02522300 -1.054726e-17

plot(johansen.const)
```

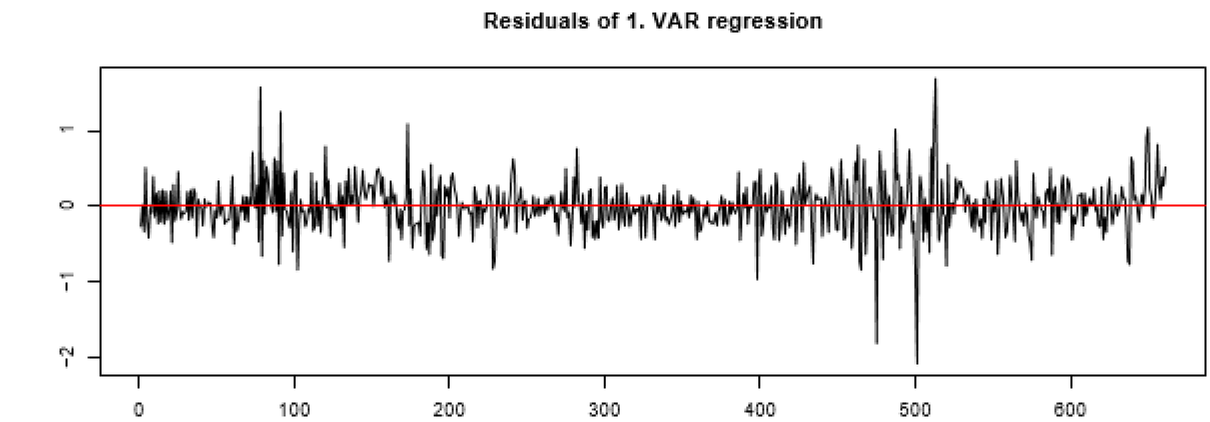

Time series plot of y1
0
5
10
1970
1980
1990
2000
2010
2020
Time

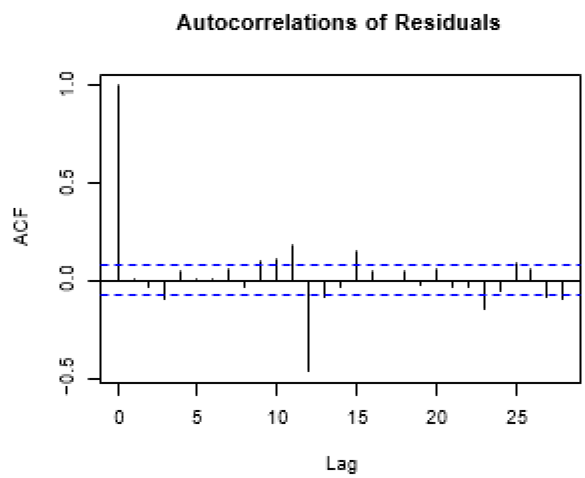

Cointegration relation of 1. variable
-900
-800
-700
0
100
200
300
400
500
600
Time

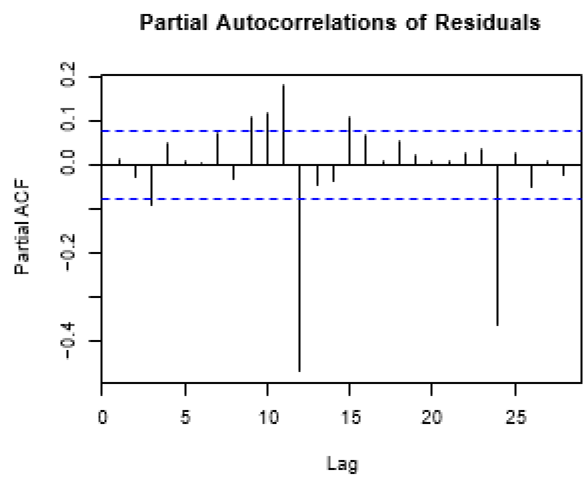


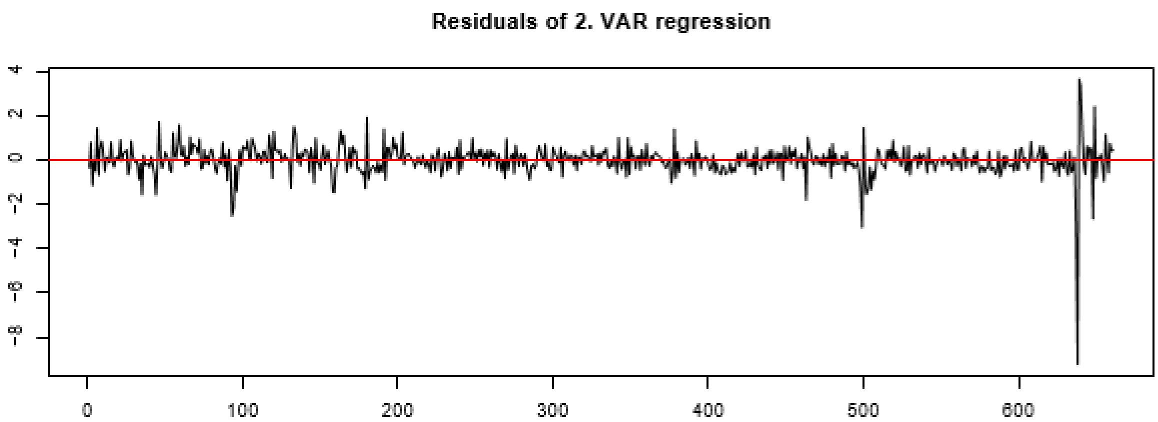


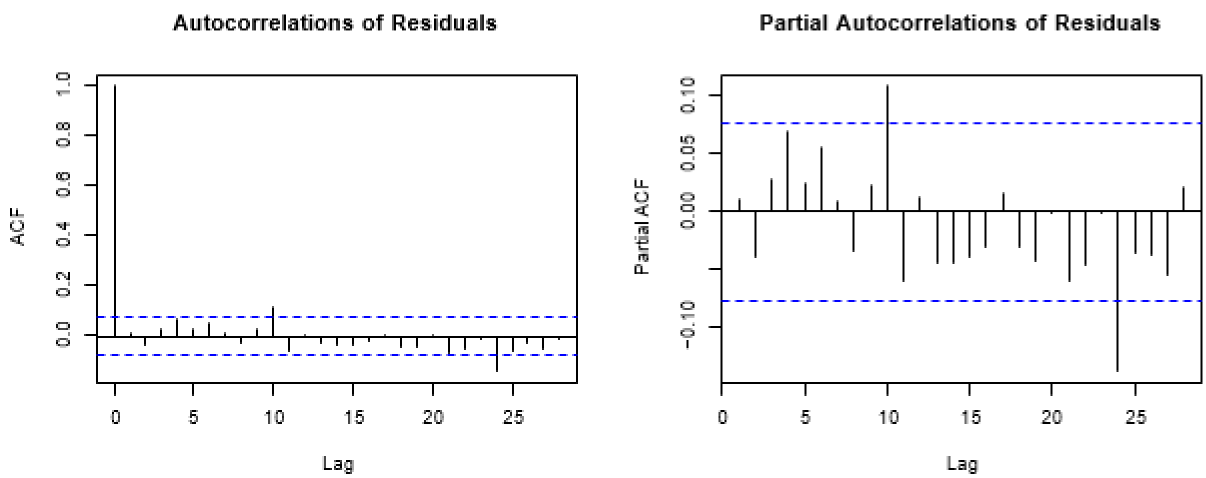

```
plotres(johansen.const)
```


Residuals of 1. VAR regression
1
0
-1
-2
0
100
200
300
400
500
600


Autocorrelations of Residuals
ACF
1.0
0.5
0.0
-0.5
0
5
10
15
20
25
Lag


Partial Autocorrelations of Residuals
Partial ACF
0.2
0.1
0.0
-0.2
-0.4
0
5
10
15
20
25
Lag





The t-tests in both tests are greater than the p-values at any significance level. This rejects the null hypothesis of zero cointegrating vectors (r=0) but also rejects the presence of only one cointegrating vector (r=1). Since the model contains two variables, it does not make sense to have more than one cointegrating relations.

Summary

The results of the tests are ambiguous and contrasting. Even though the Johansen test is more powerful than the Engle-Granger test, both tests are lowpower test. For these reasons, the analysis is now conducted by assuming that Inflation and TCU are no cointegrated in levels, as no strong supporting evidence for cointegration. Moreover, the residuals from the Johansen cointegration test seem not to be well behaved.

5. AUTO-REGRESSIVE DISTRIBUTED LAG (ARDL) MODEL: IMPACT OF A TCU SHOCK ON INFLATION RATE

```
#Recall the dataset to get rid of issue with "dynamac" package:
data.set = na.omit(
ts.intersect(

Inflation,
TCU,

dframe=TRUE))
```

The next step is to test how inflation rate reacts to shocks in the weakly exogenous variable TCU. Exhaustive results are not expected, as the inflation rate depends on a multitude of different other variables and cannot be explained only by changes in the Total Capacity Utilization Rate. Nevertheless, in the next steps the long-term effects of shocks in the TCU are explored.

```
#Because stochastic values are involved in the simulations, we set a seed
#to ensure our results are replicable.
set.seed(123)

lags = 2

ARDL = dynardl(
Inflation ~ TCU,
lags = list("TCU" = 1,          "Inflation" = 1              ),
diffs =    c("TCU"                                           ),
lagdiffs = list("TCU" = c(1:lags),  "Inflation" = c(1:lags)     ),

ec = TRUE,
constant = TRUE,
trend = FALSE,

simulate = TRUE,
shockvar = "TCU",

range = 50,
sims = 1000,
fullsims = TRUE,

data = data.set)

## [1] "Error correction (EC) specified; dependent variable to be run in
differences."
## [1] "TCU shocked by one standard deviation of TCU by default."
```

```
## [1] "dynardl estimating ..."
##   |

summary(ARDL)

##
## Call:
## lm(formula = as.formula(paste(paste(dvnamelist), "~", paste(colnames(IVs),
##     collapse = "+"), collapse = " ")))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90276 -0.19901 -0.00121  0.18187  1.68963
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.866047   0.265075  -3.267 0.001143 **
## l.1.Inflation  -0.014801   0.005408  -2.737 0.006370 **
## ld.1.Inflation  0.387860   0.039086   9.923  < 2e-16 ***
## ld.2.Inflation -0.049216   0.039119  -1.258 0.208809
## d.1.TCU         0.069500   0.019066   3.645 0.000288 ***
## l.1.TCU         0.011643   0.003396   3.429 0.000645 ***
## ld.1.TCU       -0.002605   0.019783  -0.132 0.895297
## ld.2.TCU        0.031149   0.019165   1.625 0.104592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3403 on 652 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.2095, Adjusted R-squared:  0.201
## F-statistic: 24.68 on 7 and 652 DF,  p-value: < 2.2e-16

# The adjusted R squared is 0.20 which is not very high suggesting a weak
significance of the impact of TCU on Inflation.
```

5.1 ARDL-bounds procedure

```
#Given the low power of the previous cointegration tests, the Long-run relationship
#is tested implementing the Pesaran, Shin, and Smith (2001) ARDL-bounds procedure:

pssbounds(ARDL)

##
##  PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
##
##  Observations: 660
##  Number of Lagged Regressors (not including LDV) (k): 1
##  Case: 3 (Unrestricted intercept; no trend)
##
##  -----------------------------------------------------
##  -                       F-test                      -
```

```
## ------------------------------------------------------
##                 <------- I(0) ------------ I(1) ----->
## 10% critical value       4.04            4.78
## 5% critical value        4.94            5.73
## 1% critical value        6.84            7.84
##
##
## F-statistic = 7.189
## ------------------------------------------------------
## -                       t-test                       -
## ------------------------------------------------------
##                 <------- I(0) ------------ I(1) ----->
## 10% critical value       -2.57            -2.91
## 5% critical value        -2.86            -3.22
## 1% critical value        -3.43            -3.82
##
##
## t statistic = -2.737
## ------------------------------------------------------
## F-statistic note: Asymptotic critical values used.
## t-statistic note: Asymptotic critical values used.
```

```
pssbounds(ARDL, restriction=TRUE)
```

```
##
## PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
##
## Observations: 660
## Number of Lagged Regressors (not including LDV) (k): 1
## Case: 2 (Intercept included in F-stat restriction; no trend)
##
## ------------------------------------------------------
## -                       F-test                       -
## ------------------------------------------------------
##                 <------- I(0) ------------ I(1) ----->
## 10% critical value       3.02            3.51
## 5% critical value        3.62            4.16
## 1% critical value        4.94            5.58
##
##
## F-statistic = 4.908
##
## ------------------------------------------------------
## F-statistic note: Asymptotic critical values used.
## t-statistic note: Critical values do not currently exist for Case II.
```

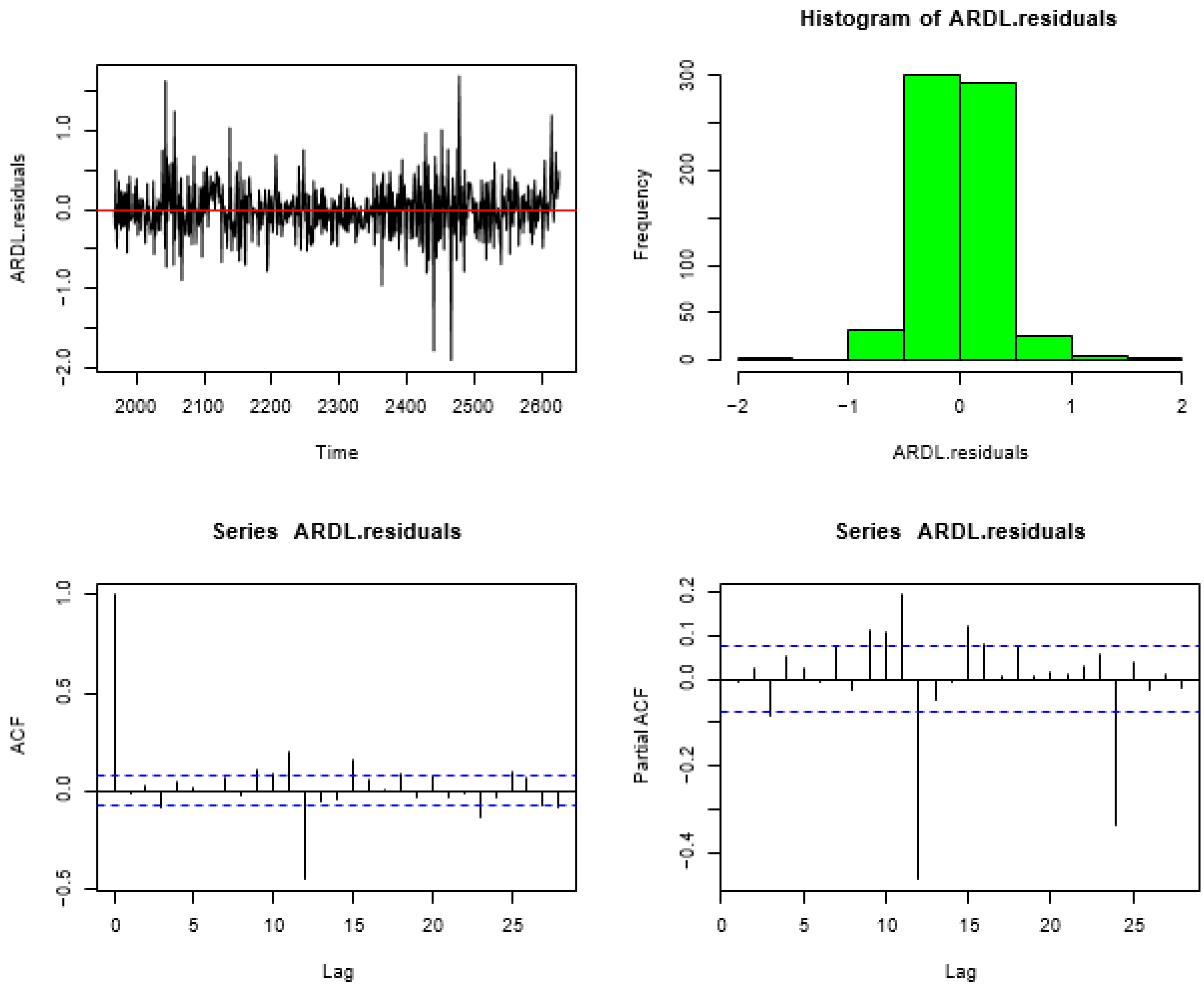
In the first version of the test, the F-test is 7.19, which falls between I[0] and I[1] at the 1% significance level, suggesting that results are inconclusive. # The t-test is -2.74, which is lower than I[0] at the 1% significance level, so the null hypothesis of no cointegration is not rejected. By turning the restriction on, the test suggests no cointegration, as the F-statistic of 4.90 is lower than I[0], but still on the borderline.

```
# Next step is to test for autocorrelation in the residuals:lags = 2

dynardl.auto.correlated(ARDL)

##
##  ------------------------------------------------------
##  Breusch-Godfrey LM Test
##  Test statistic: 3.213
##  p-value: 0.073
##  H_0: no autocorrelation up to AR 1
##
##  ------------------------------------------------------
##  Shapiro-Wilk Test for Normality
##  Test statistic: 0.955
##  p-value: 0
##  H_0: residuals are distributed normal
##
##  ------------------------------------------------------
##  Log-likelihood: -220.968
##  AIC: 459.937
##  BIC: 500.367
##  Note: AIC and BIC calculated with k = 8 on T = 660 observations.
##
##  ------------------------------------------------------
## Breusch-Godfrey test indicates we reject the null hypothesis of no
autocorrelation at p < 0.10.
##  Add lags to remove autocorrelation before running dynardl simulations.
## Shapiro-Wilk test indicates we reject the null hypothesis of normality at p <
0.01.
```

The p-value in the Breusch-Godfrey test is 0.073 so it fails to reject the null hypothesis of no autocorrelation at the 5%. The Shapiro-Wilk test for Normality suggests that the residuals are not normally distributed by rejecting the null hypothesis of normality in the residuals at less than 1%.

5.2 ARDL Model Residuals and Additional Tests

```
#Additional tests on the residuals are now computed:

ARDL.residuals =    ARDL$model$residuals
ARDL.residuals = ts(ARDL$model$residuals,  start=c(1967, 1) ,  frequency=1)

par(mfrow = c(2, 2))
plot( ARDL.residuals )
abline( h=0, col="red" )
hist( ARDL.residuals, col="green")
acf( ARDL.residuals )
pacf( ARDL.residuals )
```



The histogram of the residuals shows that the residuals are not normally distributed as the graph is slightly left skewed and there is a presence of few outliers. By looking at the Autocorrelation and Partial autocorrelation functions, it cannot be concluded with certainty that there is no autocorrelation in the residuals.

```
jarque.bera.test(ARDL.residuals)  # Jarque-Bera test for normal residuals

##
##  Jarque Bera Test
##
## data:  ARDL.residuals
## X-squared = 524.46, df = 2, p-value < 2.2e-16

bds.test(ARDL.residuals)        # BDS test Ho: series of i.i.d. random variable

##
##   BDS Test
##
## data:  ARDL.residuals
##
```

```
## Embedding dimension =  2 3
##
## Epsilon for close points =  0.1692 0.3384 0.5077 0.6769
##
## Standard Normal =
##       [ 0.1692 ] [ 0.3384 ] [ 0.5077 ] [ 0.6769 ]
## [ 2 ]     4.9513     5.0283     5.4672     6.0349
## [ 3 ]     6.2143     6.1547     6.4707     6.7919
##
## p-value =
##       [ 0.1692 ] [ 0.3384 ] [ 0.5077 ] [ 0.6769 ]
## [ 2 ]          0          0          0          0
## [ 3 ]          0          0          0          0
```

```
bptest(ARDL$model)              # Breusch-Pagan test against heteroskedasticity
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ARDL$model
## BP = 22.112, df = 7, p-value = 0.00243
```

The p-value in the Jarque Bera Test is close to 0, confirming that the residuals are not normally distributed. The Breush-Pagan test for heteroskedasticity suggests heteroskedasticity in the residuals, as the p-value is smaller than 0.01. We can conclude that the residuals are not well behaved.

5.3 ARDL Model with robust standard errors

```
#Since the residuals are not normally distributed and present heteroskedasticity,
#we can compute the ARDL model with robust standard errors that correct for
#heteroskedasticity and auto-correlation.

coeftest(ARDL$model, vcov.=vcovHAC)
```

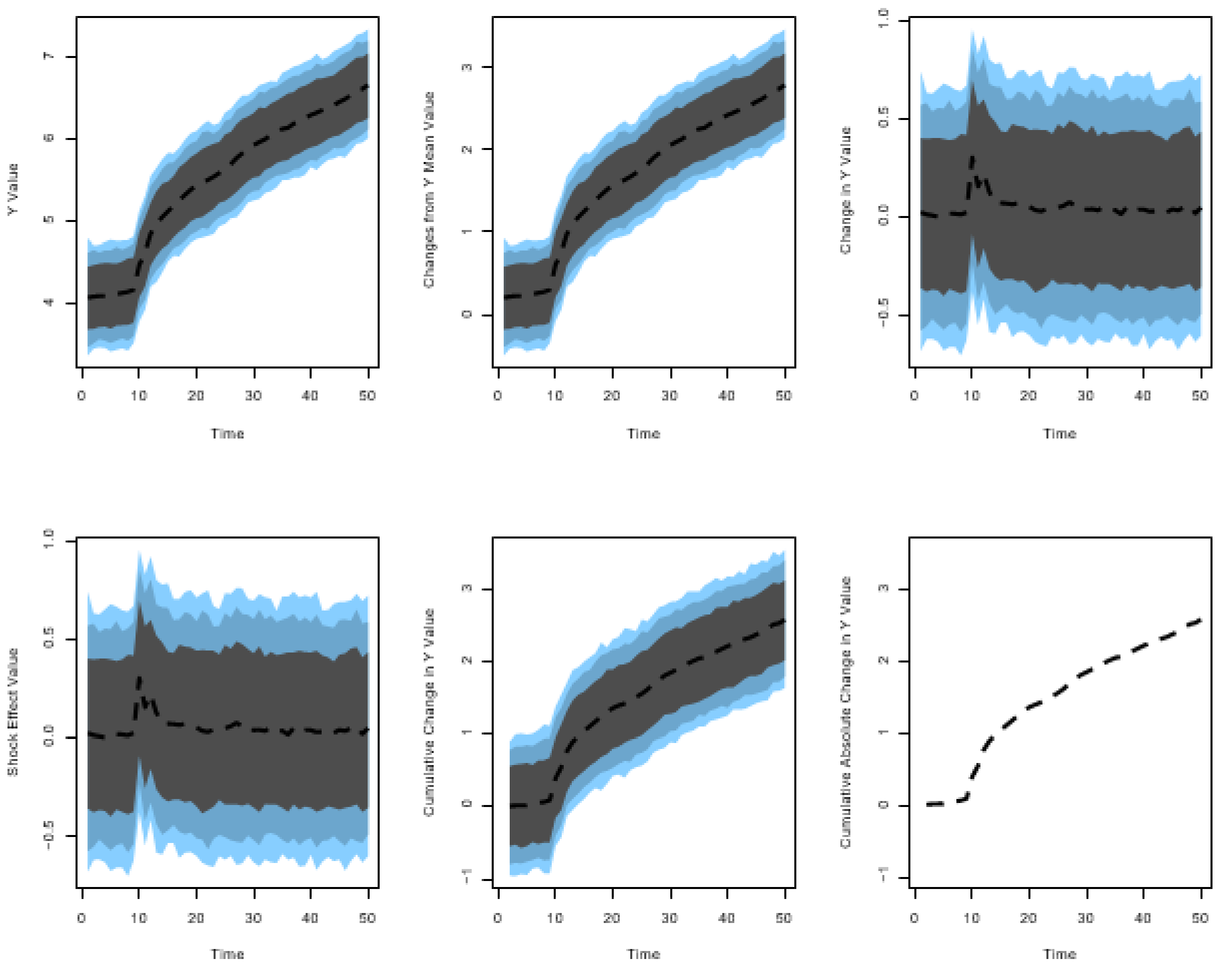
```
##
## t test of coefficients:
##
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.8660467  0.3222690 -2.6873 0.007386 **
## l.1.Inflation  -0.0148008  0.0052486 -2.8199 0.004949 **
## ld.1.Inflation  0.3878599  0.0604911  6.4118 2.76e-10 ***
## ld.2.Inflation -0.0492158  0.0497429 -0.9894 0.322833
## d.1.TCU         0.0695002  0.0212523  3.2702 0.001131 **
## l.1.TCU         0.0116428  0.0039875  2.9198 0.003623 **
## ld.1.TCU       -0.0026045  0.0225346 -0.1156 0.908022
## ld.2.TCU        0.0311486  0.0326627  0.9536 0.340617
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Impulse-Response plots

```
#It is possible now to plot the impulse response functions for the ARDL model. This
#is a counterfactual simulation of a response to a shock in some exogenous
#variable.

lags = 2

ARDL = dynardl(
Inflation ~ TCU,

lags = list("TCU" = 1,          "Inflation" = 1             ),
diffs =    c("TCU"                                           ),
lagdiffs = list("TCU" = c(1:lags),  "Inflation" = c(1:lags)     ),

ec = TRUE,
constant = TRUE,
trend = FALSE,

simulate = TRUE,
shockvar = "TCU",
range = 50,
sims = 1000,
fullsims = TRUE,

data = data.set)

## [1] "Error correction (EC) specified; dependent variable to be run in
differences."
## [1] "TCU shocked by one standard deviation of TCU by default."
## [1] "dynardl estimating ..."
##    |

summary(ARDL)

##
## Call:
## lm(formula = as.formula(paste(paste(dvnamelist), "~", paste(colnames(IVs),
##     collapse = "+"), collapse = " ")))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90276 -0.19901 -0.00121  0.18187  1.68963
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.866047   0.265075  -3.267 0.001143 **
## l.1.Inflation  -0.014801   0.005408  -2.737 0.006370 **
## ld.1.Inflation  0.387860   0.039086   9.923  < 2e-16 ***
## ld.2.Inflation -0.049216   0.039119  -1.258 0.208809
```

```
## d.1.TCU          0.069500   0.019066   3.645 0.000288 ***
## l.1.TCU          0.011643   0.003396   3.429 0.000645 ***
## ld.1.TCU        -0.002605   0.019783  -0.132 0.895297
## ld.2.TCU         0.031149   0.019165   1.625 0.104592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3403 on 652 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.2095, Adjusted R-squared:  0.201
## F-statistic: 24.68 on 7 and 652 DF,  p-value: < 2.2e-16
```

```
dynardl.all.plots(ARDL) # all plots together
```



By looking at the cumulative change in Inflation due to a permanent one-unit shock in TCU (+1%), the graph suggests that there is weak evidence of a positive cumulative impact that leads the inflation rate to rise over the years following the shock.

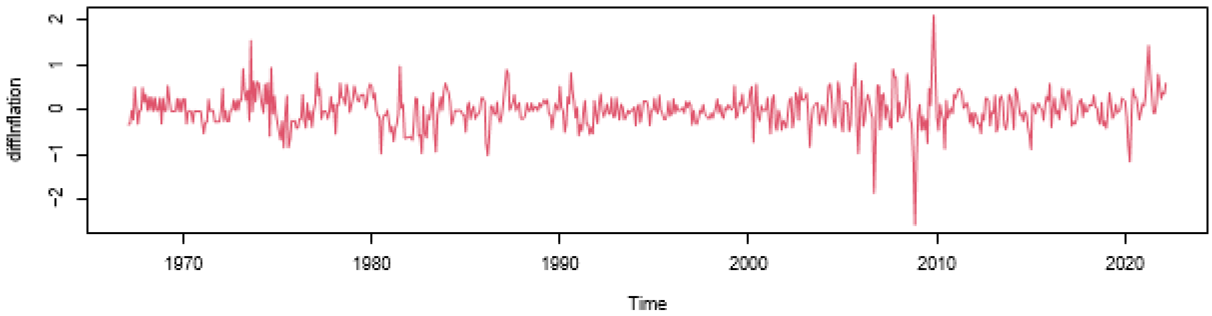
Summary

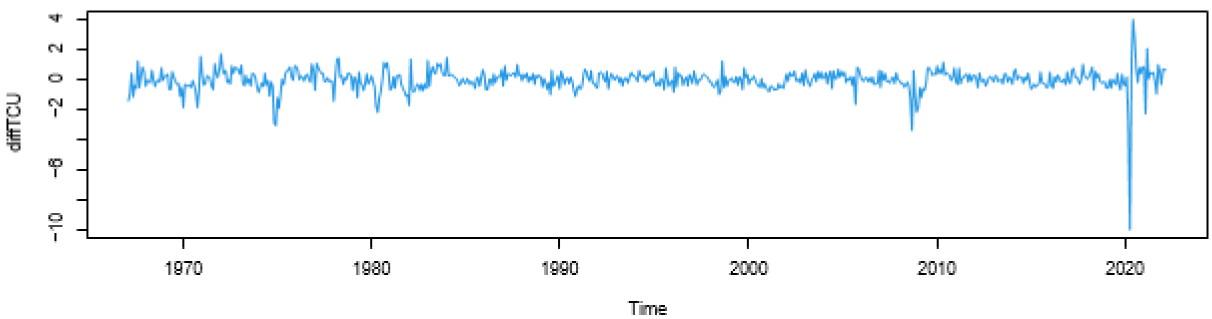
This evidence is weak: as previously mentioned, the ARDL model with only the Inflation and TCU is not specified correctly. The residuals are not white noise with 2 lags, and the residuals are not yet normally distributed. One possible solution would be to add more variables into the ARDL model that in conjunction with TCU could help in better explaining the behavior of inflation.

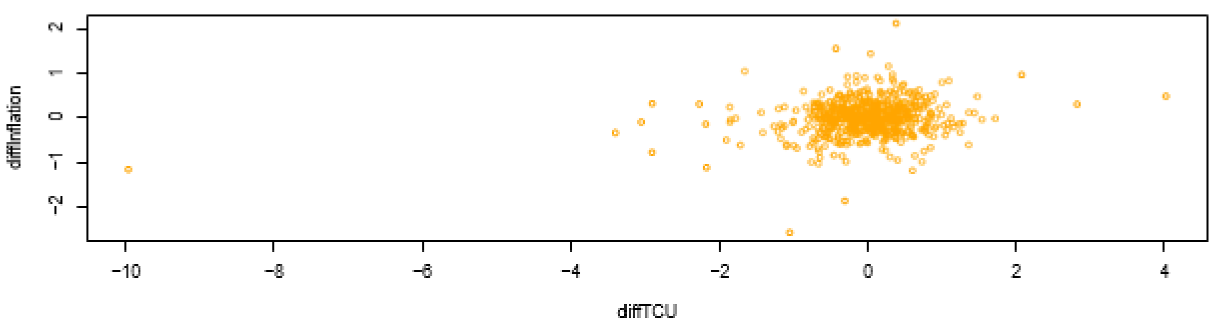
6. BIVARIATE VAR(p) MODEL

The ARDL model assumes only one dependent variable, but variables can be endogenous to one another. To solve for a problem of simultaneity, in the next step the VAR in first differences with only two variables is implemented assuming no cointegration.

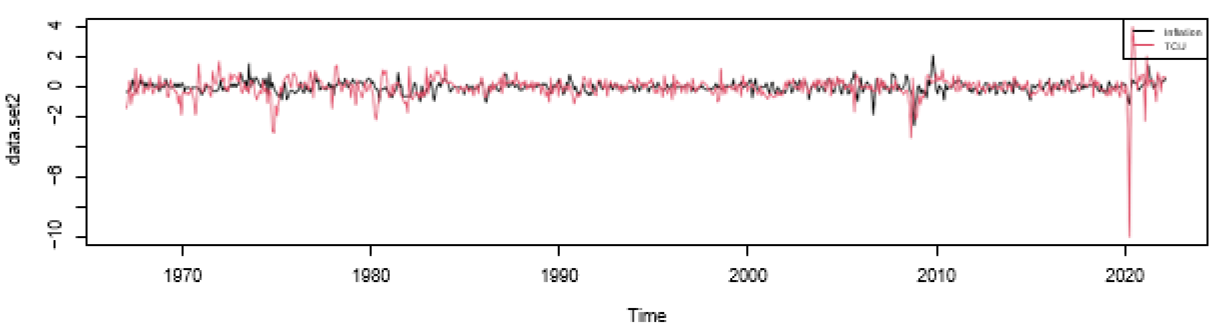
```
#Because previous tests suggested that the series have unit root, but the tests for
#cointegration do not provide enough strong evidence for cointegration, the next
#step is estimating the VAR in first differences.

data.set = na.omit(
ts.intersect(

Inflation,
TCU,

dframe=TRUE))

#Run the model in first differences
diffInflation = diff(Inflation)
diffTCU = diff(TCU)


data.set2 = na.omit(
ts.intersect(

diffInflation,
diffTCU,

dframe=TRUE))


diffInflation   = ts( data.set2$diffInflation,    start=c(1967, 2),   frequency=12)
diffTCU         = ts( data.set2$diffTCU,          start=c(1967, 2),   frequency=12)
data.set2=ts(data.set2, start=c(1967, 2), frequency=12)

par(mfrow=c(3,1))
plot(diffInflation)
plot(diffTCU)
plot(diffInflation ~ diffTCU)
```








```
plot(data.set2, plot.type="single", col=1:ncol(data.set))
legend("topright", colnames(data.set), col=1:ncol(data.set), lty=1, cex=.65)
```



```
#Check the optimal lag length according to the information criteria:
#Using the "VARS" package
VARselect(data.set2, lag.max=10, type="none",  season = NULL, exogen =
NULL)$selection

## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      1      1     10

VARselect(data.set2, lag.max=10, type="const", season = NULL, exogen =
NULL)$selection

## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      1      1     10

VARselect(data.set2, lag.max=10, type="trend", season = NULL, exogen =
NULL)$selection

## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      1      1     10

VARselect(data.set2, lag.max=10, type="both",  season = NULL, exogen =
NULL)$selection

## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      1      1     10

#According to the Hannan-Quinn criterium, the optimal lags are now 1.

optimal.lags = 1

#Correlation between the two variables:
cor(data.set)

##            Inflation       TCU
## Inflation 1.0000000 0.3605283
## TCU       0.3605283 1.0000000

# The correlation between the two variables in first difference is 0.36 which is
still not that strong.



#The reduced form of the VAR model is computed as follow:

var.model.const  <- VAR(data.set2, p=optimal.lags, type="const", exogen=NULL)
summary(var.model.const)

##
## VAR Estimation Results:
## =========================
## Endogenous variables: diffInflation, diffTCU
## Deterministic variables: const
## Sample size: 661
```

```
## Log Likelihood: -937.279
## Roots of the characteristic polynomial:
## 0.4265 0.2336
## Call:
## VAR(y = data.set2, p = optimal.lags, type = "const", exogen = NULL)
##
##
## Estimation results for equation diffInflation:
## ==============================================
## diffInflation = diffInflation.l1 + diffTCU.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## diffInflation.l1 0.395733   0.036214  10.928   <2e-16 ***
## diffTCU.l1       0.033362   0.018646   1.789    0.074 .
## const            0.006063   0.013513   0.449    0.654
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.3472 on 658 degrees of freedom
## Multiple R-Squared: 0.1701,  Adjusted R-squared: 0.1676
## F-statistic: 67.43 on 2 and 658 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation diffTCU:
## ========================================
## diffTCU = diffInflation.l1 + diffTCU.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## diffInflation.l1  0.14946    0.07365   2.029   0.0428 *
## diffTCU.l1        0.26439    0.03792   6.972  7.6e-12 ***
## const            -0.01095    0.02748  -0.398   0.6905
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.7062 on 658 degrees of freedom
## Multiple R-Squared: 0.08388, Adjusted R-squared: 0.0811
## F-statistic: 30.12 on 2 and 658 DF,  p-value: 3.034e-13
##
##
##
## Covariance matrix of residuals:
##               diffInflation diffTCU
## diffInflation       0.12057 0.03394
## diffTCU             0.03394 0.49865
##
## Correlation matrix of residuals:
##               diffInflation diffTCU
```

```
## diffInflation        1.0000  0.1384
## diffTCU              0.1384  1.0000

# The residual correlations in the contemporaneous correlation matrix are quite low
(0.1384), suggesting that the forecast errors of the endogenous variable are weakly
correlated.

plot(var.model.const)
```

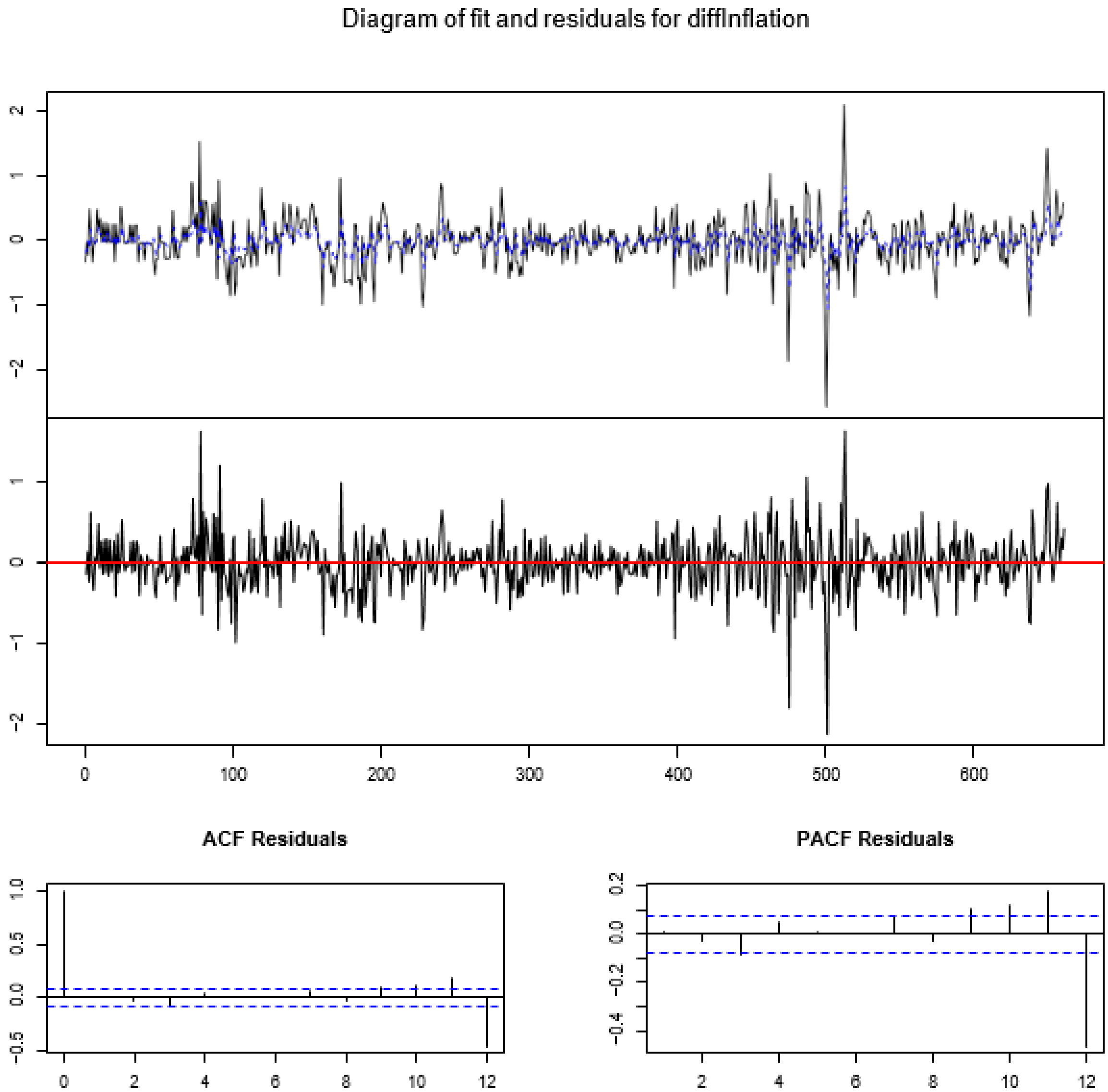
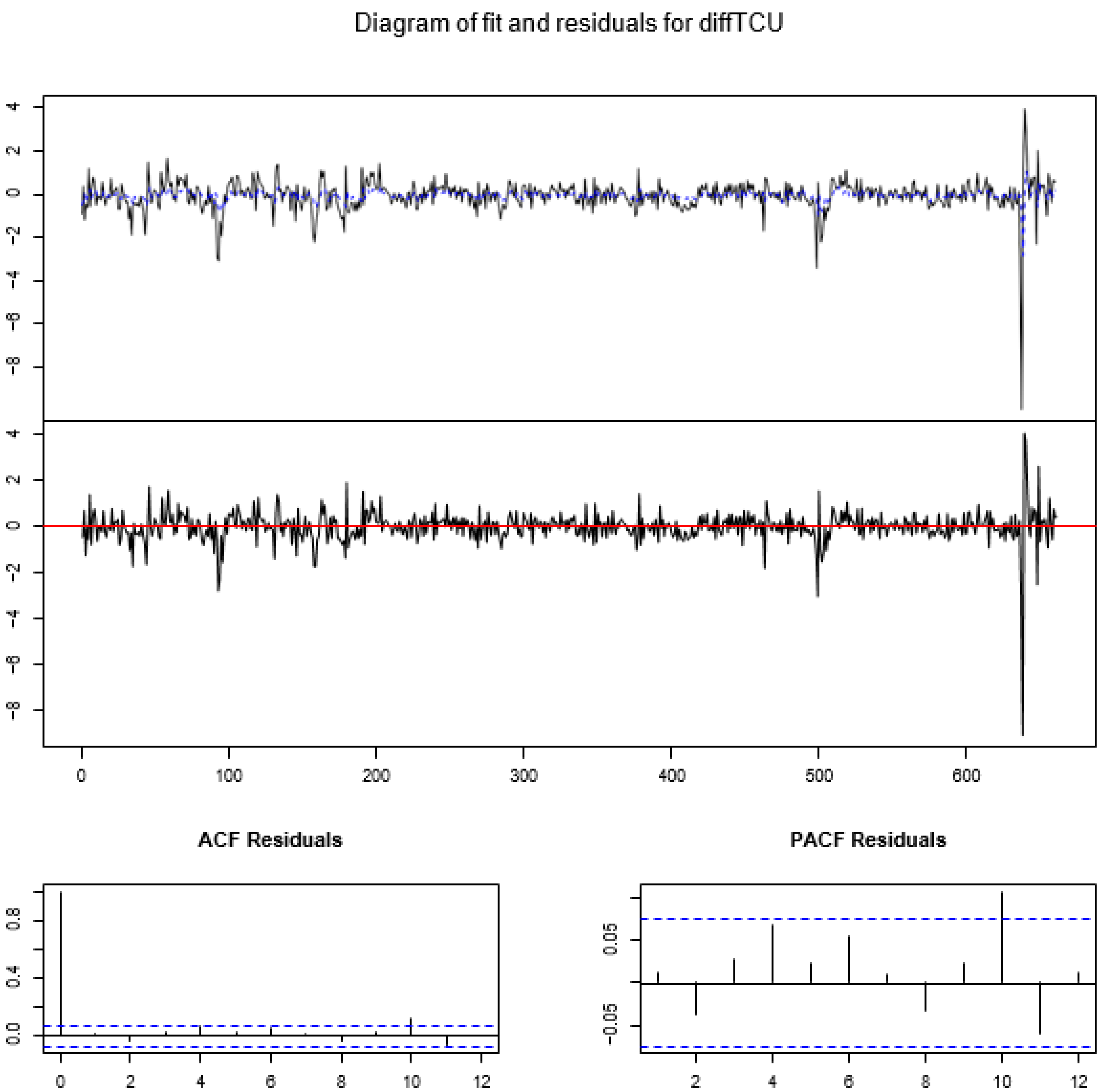


Diagram of fit and residuals for diffInflation
2
1
0
-1
-2
1
0
-1
-2
0
100
200
300
400
500
600
ACF Residuals
PACF Residuals
1.0
0.5
0.0
-0.5
0
2
4
6
8
10
12
0.2
0.0
-0.2
-0.4
2
4
6
8
10
12



6.1 Granger and Instantaneous causality test for the VAR model

Inflation –> TCU

```
causality(var.model.const, cause="diffInflation", boot=FALSE, boot.runs=1000)

## $Granger
##
##  Granger causality H0: diffInflation do not Granger-cause diffTCU
##
## data:  VAR object var.model.const
## F-Test = 4.1185, df1 = 1, df2 = 1316, p-value = 0.04262
##
##
```

```
## $Instant
##
##  H0: No instantaneous causality between: diffInflation and diffTCU
##
## data:  VAR object var.model.const
## Chi-squared = 12.426, df = 1, p-value = 0.0004233
```

TCU –> Inflation

```
causality(var.model.const, cause="diffTCU", boot=FALSE, boot.runs=1000)

## $Granger
##
##  Granger causality H0: diffTCU do not Granger-cause diffInflation
##
## data:  VAR object var.model.const
## F-Test = 3.2013, df1 = 1, df2 = 1316, p-value = 0.07381
##
##
## $Instant
##
##  H0: No instantaneous causality between: diffTCU and diffInflation
##
## data:  VAR object var.model.const
## Chi-squared = 12.426, df = 1, p-value = 0.0004233
```

The previous Granger causality tests do not control for Heteroskedasticity, so the next Granger tests includes the HC covar matrix using a robust heteroskedasticity variance-covariance matrix:

Inflation –> TCU

```
causality(var.model.const, cause="diffInflation", boot=FALSE, boot.runs=1000,
vcov.=vcovHC(var.model.const))

## $Granger
##
##  Granger causality H0: diffInflation do not Granger-cause diffTCU
##
## data:  VAR object var.model.const
## F-Test = 2.7224, df1 = 1, df2 = 1316, p-value = 0.09919
##
##
## $Instant
##
##  H0: No instantaneous causality between: diffInflation and diffTCU
##
## data:  VAR object var.model.const
## Chi-squared = 12.426, df = 1, p-value = 0.0004233

# The p-value is 0.10, suggesting that Inflation might Granger-cause TCU in first
differences.
```

TCU –> Inflation

```
causality(var.model.const, cause="diffTCU", boot=FALSE, boot.runs=1000,
vcov.=vcovHC(var.model.const))

## $Granger
##
##  Granger causality H0: diffTCU do not Granger-cause diffInflation
##
## data:  VAR object var.model.const
## F-Test = 1.0456, df1 = 1, df2 = 1316, p-value = 0.3067
##
##
## $Instant
##
##  H0: No instantaneous causality between: diffTCU and diffInflation
##
## data:  VAR object var.model.const
## Chi-squared = 12.426, df = 1, p-value = 0.0004233

# The p-value here is much larger (0.31), so the null hypothesis Ho: TCU do not
Granger-cause Inflation is not rejected.
```

Summary

From the Granger causality tests it could be observed that Inflation might Granger cause TCU in first differences but not the opposite. This means that past values of inflation could predict TCU, but past values of TCU would not predict the behavior of Inflation.

6.2 Recursive VAR

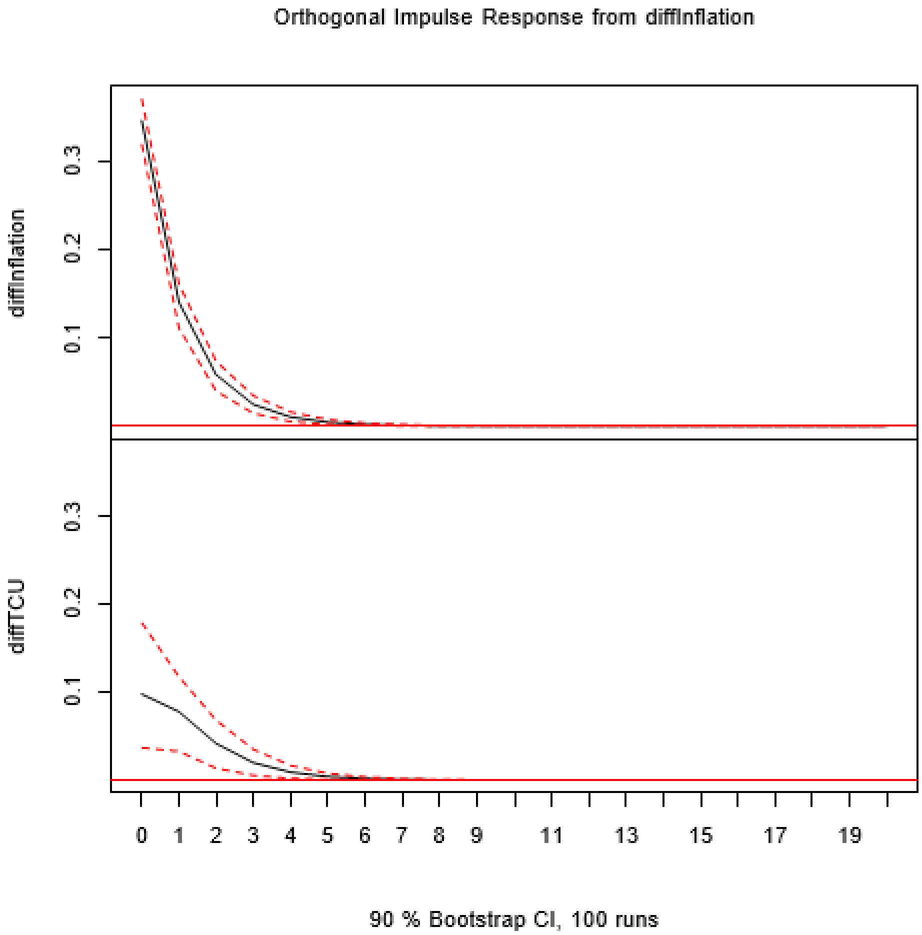
Now a VAR model that supposes that every variable is endogenous to one another is implemented by using the Cholesky decompositions for orthogonal errors. Because it has been previously stated that the causality runs from Inflation to TCU, the data are ordered accordingly and the ordered VAR with a constant is estimated.

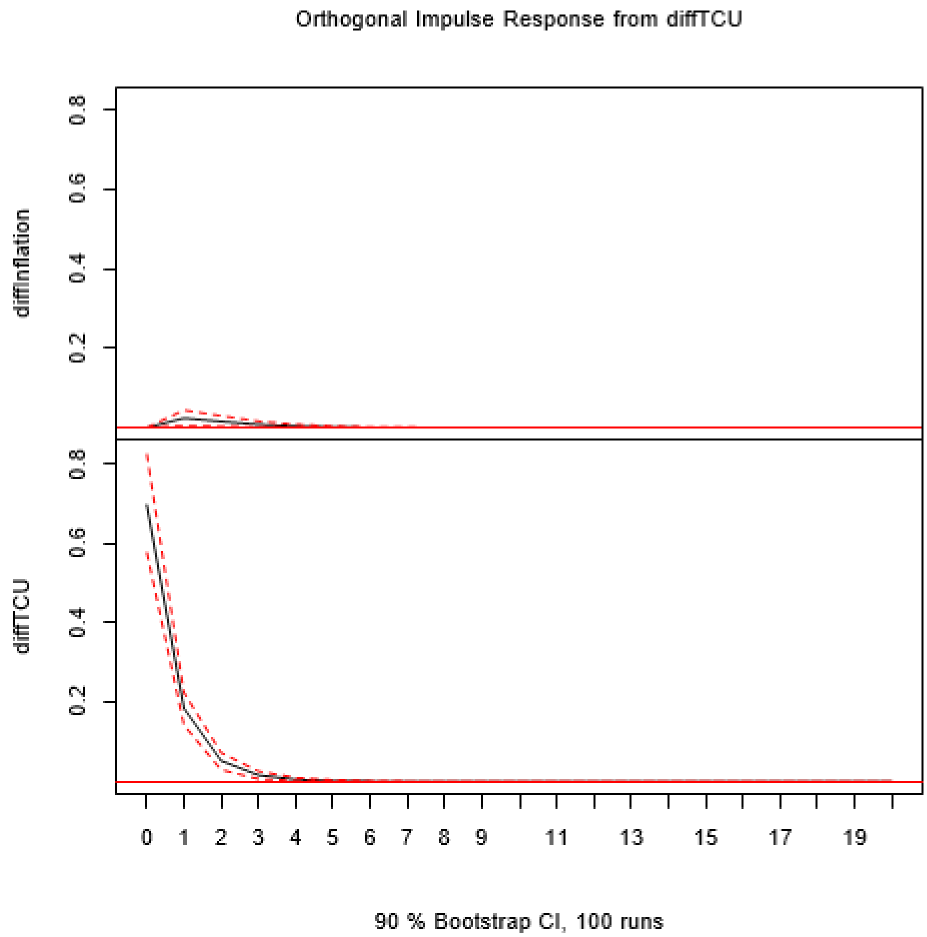
```
#Ordering: Inflation --> TCU
ordered.data.set2 = data.set2[, c("diffInflation","diffTCU")]

#Estimate the ordered VARs:
var.ordered.const  <- VAR(ordered.data.set2,  p=optimal.lags,   type="const",
exogen=NULL)
```

6.3 Forecast error variance decomposition (in percentage terms)

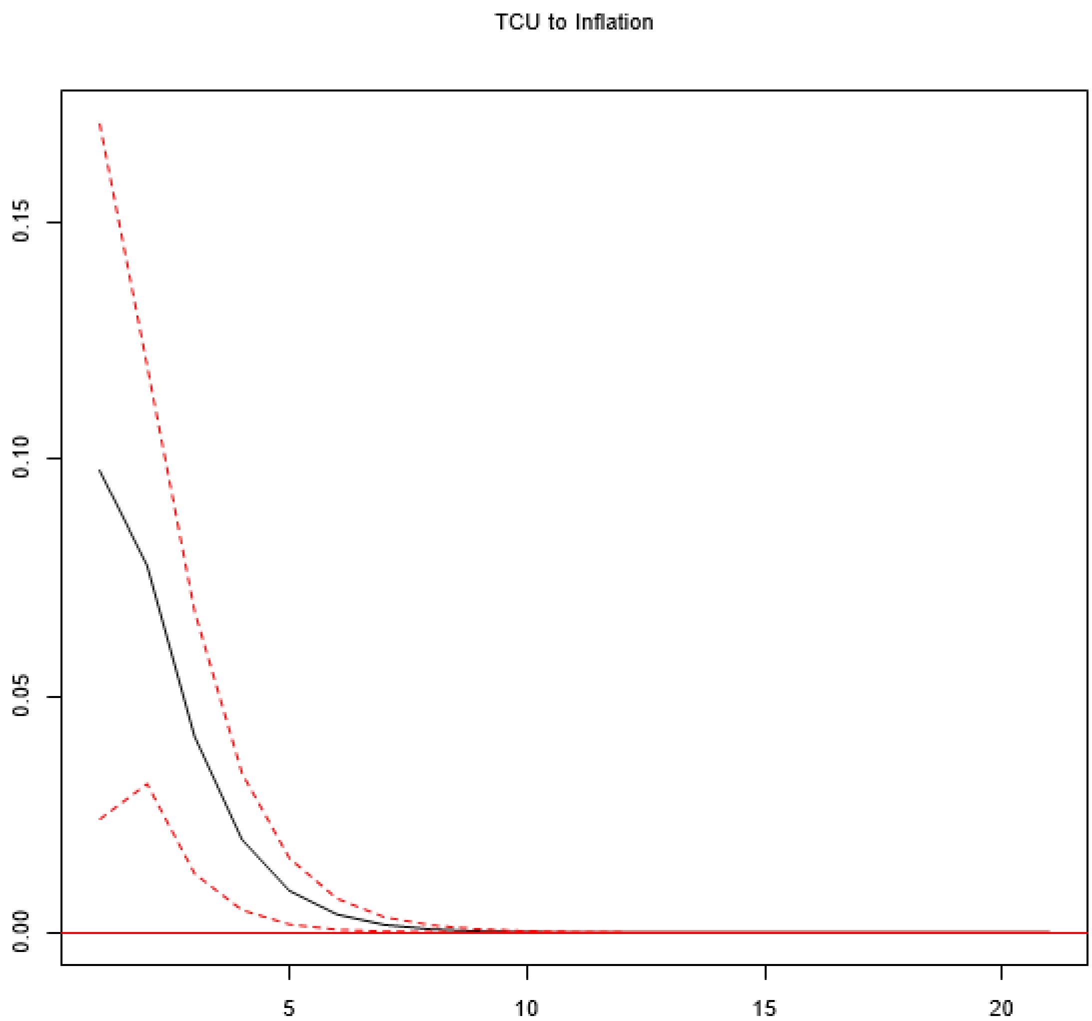
Short run (non-cumulative):

```
plot(irf(var.ordered.const, n.ahead=20, ortho=TRUE, cumulative=FALSE, boot=TRUE,
ci=0.90, runs=100))
```

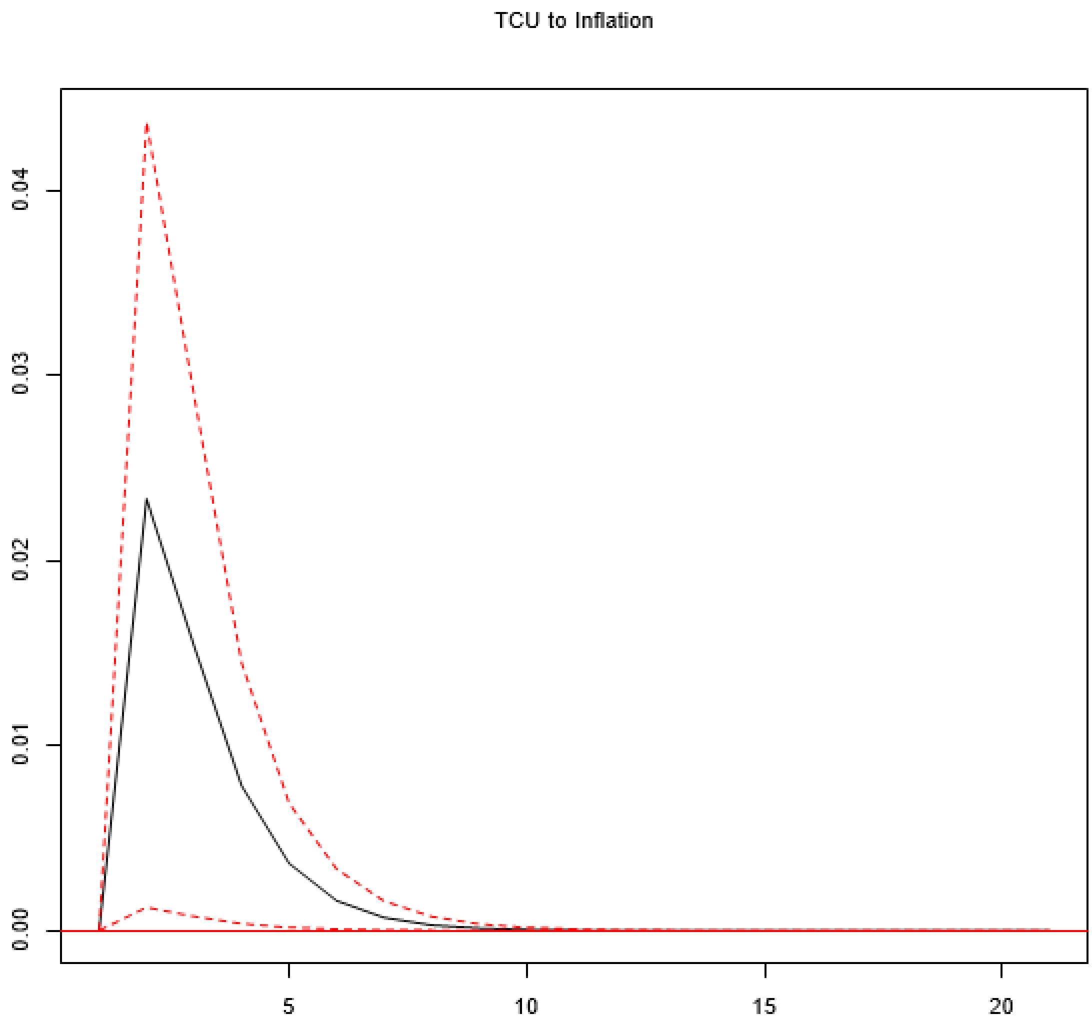

Orthogonal Impulse Response from diffInflation
diffInflation
0.1
0.2
0.3
diffTCU
0.1
0.2
0.3
0
1
2
3
4
5
6
7
8
9
11
13
15
17
19
90 % Bootstrap CI, 100 runs




```
plot(irf(var.ordered.const,  impulse="diffInflation", response="diffTCU",
n.ahead=20, ortho=TRUE, cumulative=FALSE, boot=TRUE, ci=0.90, runs=100, seed=NULL),
main="TCU to Inflation", xlab="Lag", ylab="", sub="", oma=c(3,0,3,0))
```



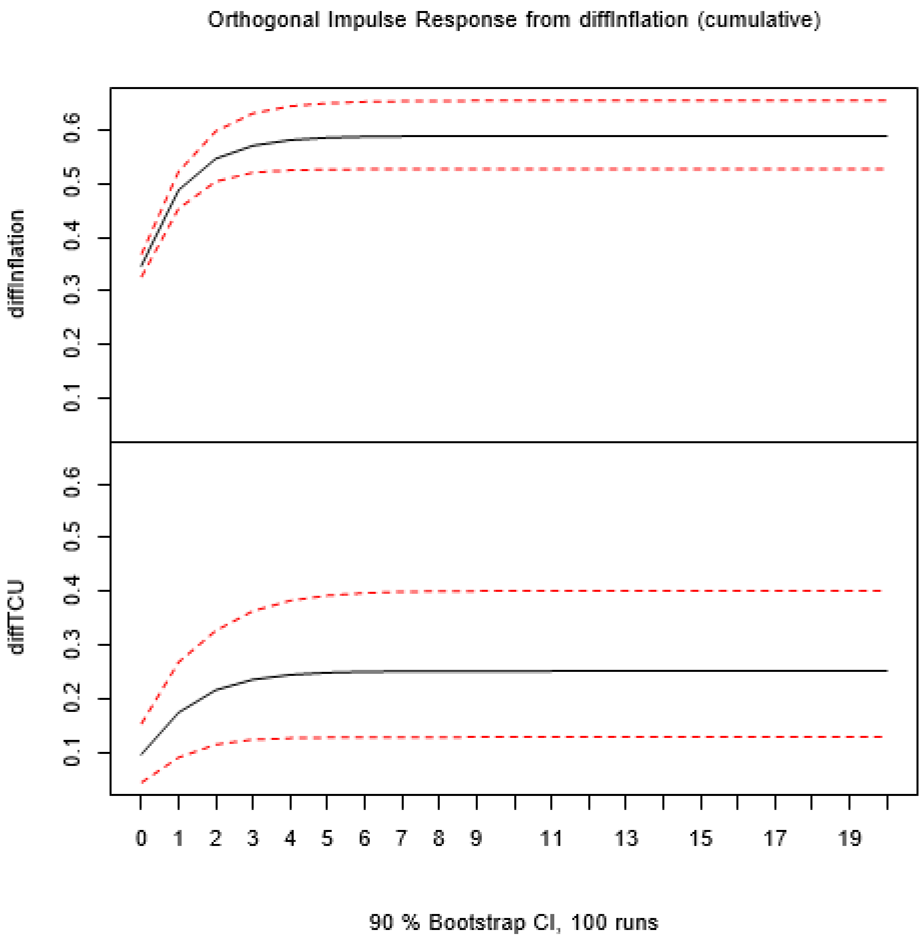

```
plot(irf(var.ordered.const,  impulse="diffTCU", response="diffInflation",
n.ahead=20, ortho=TRUE, cumulative=FALSE, boot=TRUE, ci=0.90, runs=100, seed=NULL),
main="TCU to Inflation", xlab="Lag", ylab="", sub="", oma=c(3,0,3,0))
```



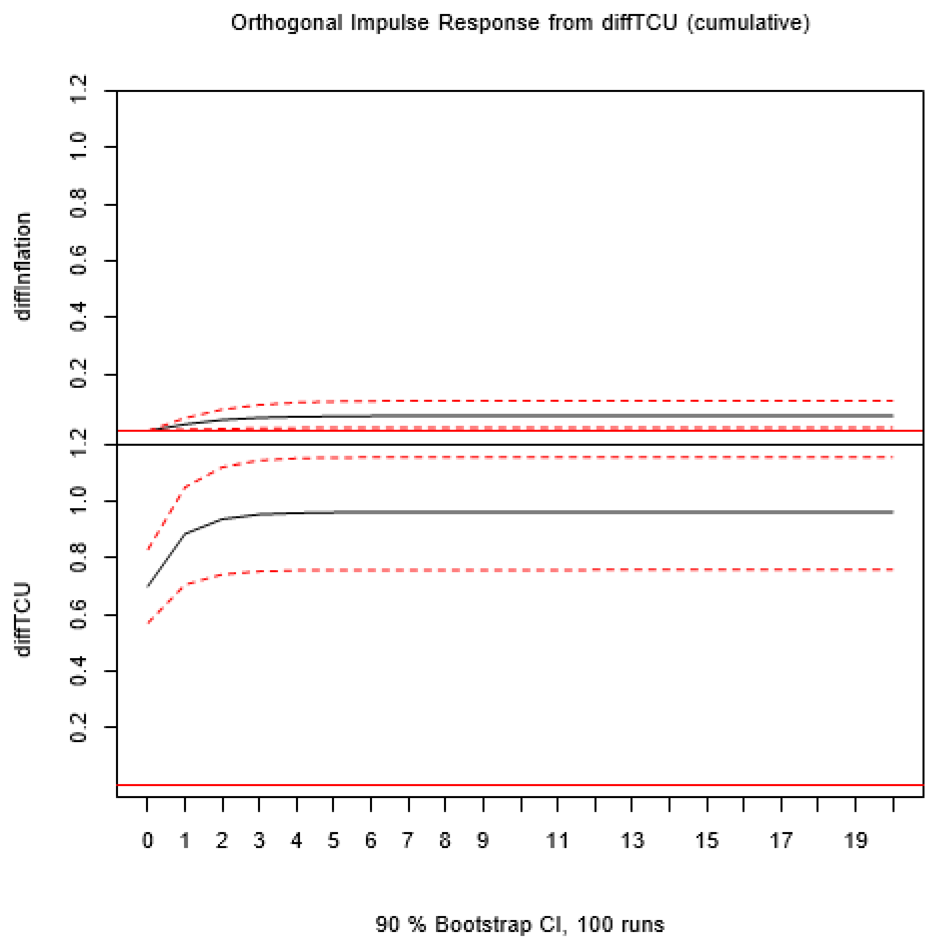
In the short run, the impulse response functions are returning to zero because the variables in first differences are now stationary. Because the variables are stationary, also the Granger-causality tests are meaningful.

Long run (cumulative):

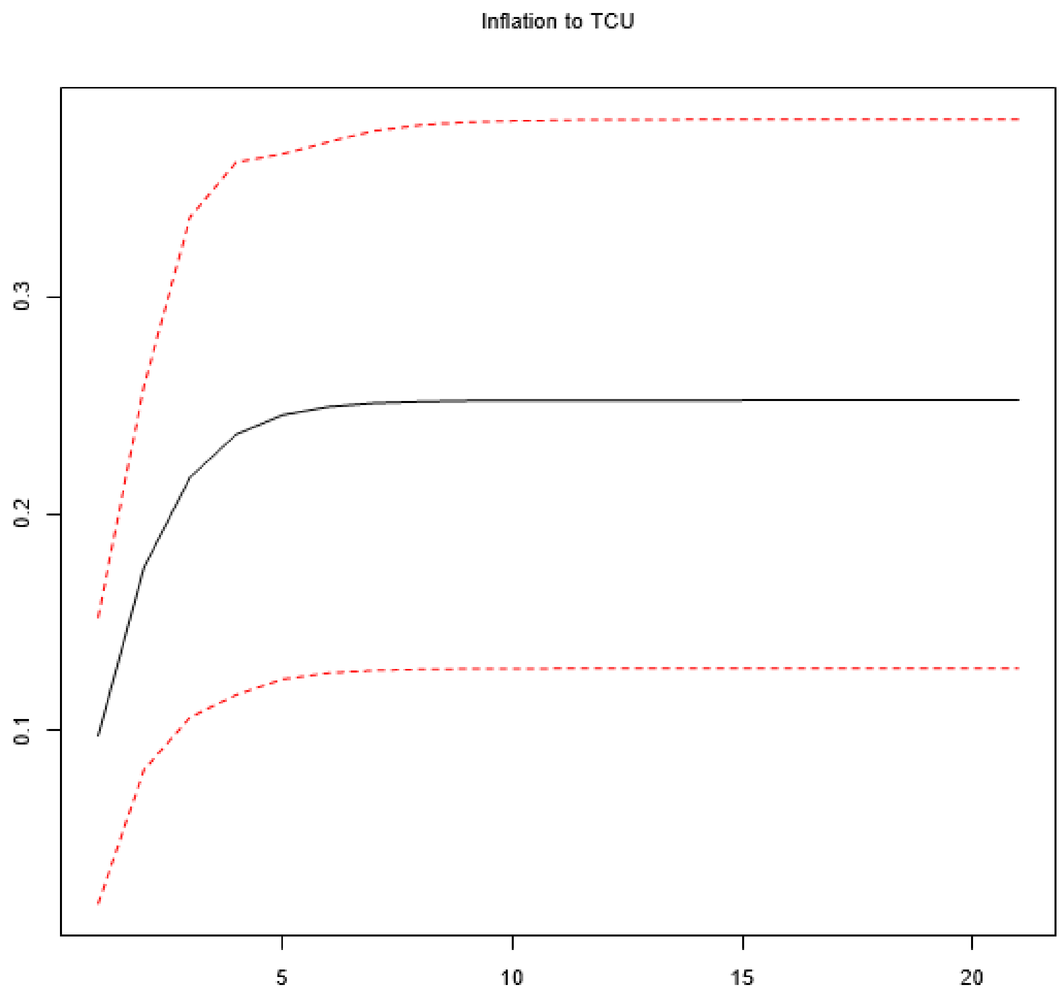
```
plot(irf(var.ordered.const, n.ahead=20, ortho=TRUE, cumulative=TRUE, boot=TRUE,
ci=0.90, runs=100))
```



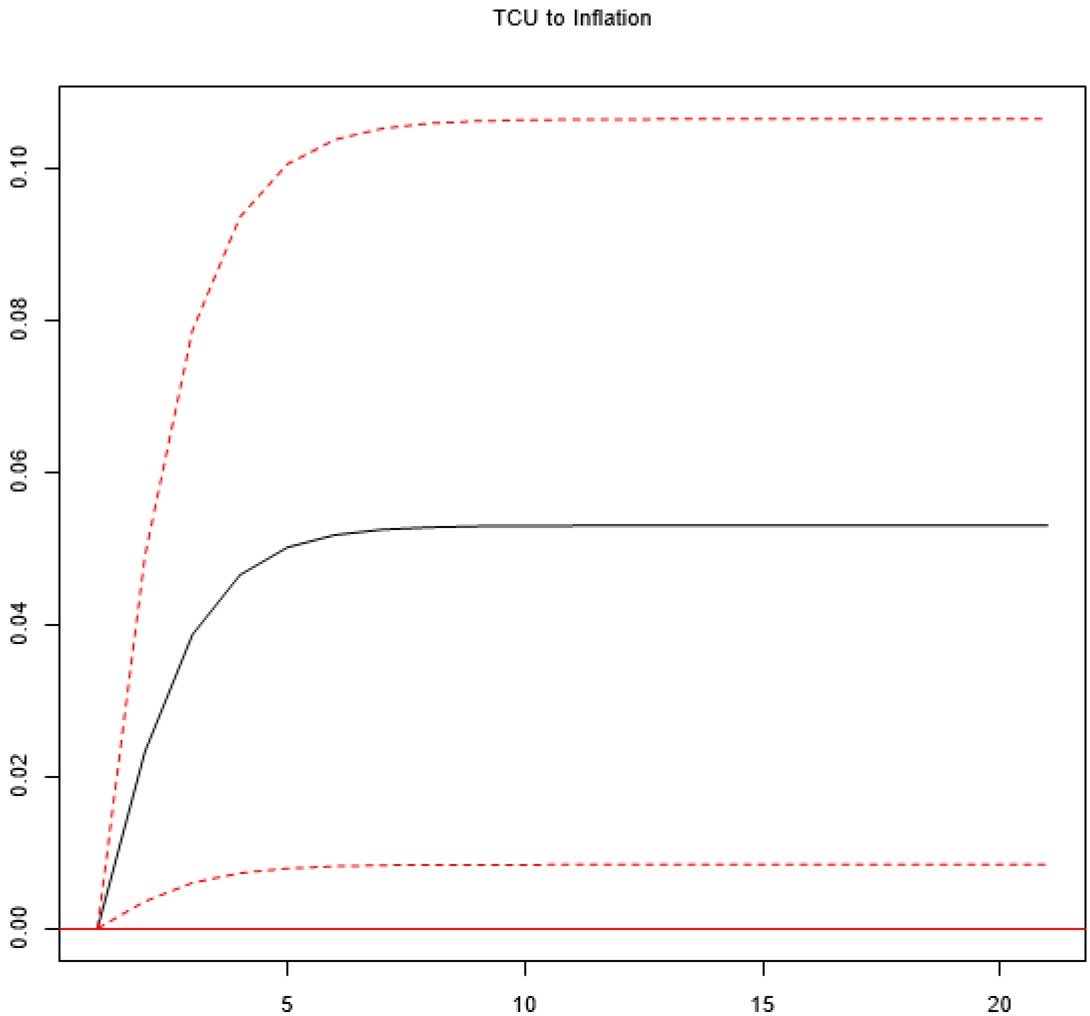
Orthogonal Impulse Response from diffInflation (cumulative)
diffInflation
diffTCU
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4
5
6
7
8
9
11
13
15
17
19
90 % Bootstrap CI, 100 runs




```
plot(irf(var.ordered.const,  impulse="diffInflation", response="diffTCU",
n.ahead=20, ortho=TRUE, cumulative=TRUE, boot=TRUE, ci=0.90, runs=100, seed=NULL),
main="Inflation to TCU", xlab="Lag", ylab="", sub="", oma=c(3,0,3,0))
```




```
plot(irf(var.ordered.const,  impulse="diffTCU", response="diffInflation",
n.ahead=20, ortho=TRUE, cumulative=TRUE, boot=TRUE, ci=0.90, runs=100, seed=NULL),
main="TCU to Inflation", xlab="Lag", ylab="", sub="", oma=c(3,0,3,0))
```



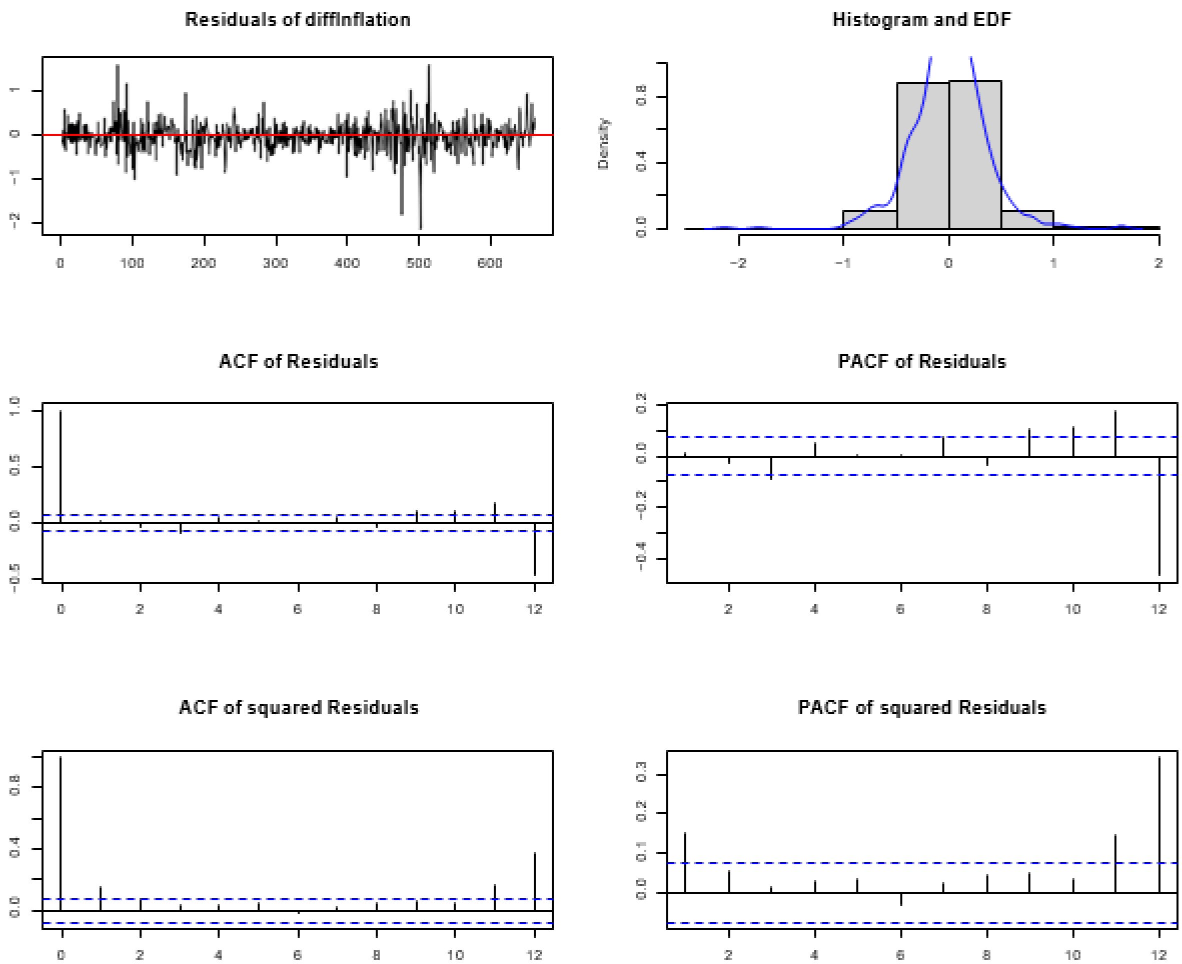
The long-run impulse functions, being cumulative, do not need to revert back to zero. These functions show that after an initial increase of 1% of TCU, Inflation raises of 0.05% and set at a new stable trend after the 5th lag. On the other side, an increase in 1% in Inflation rate, TCU raises of 0.25% and stabilize after 5 months.

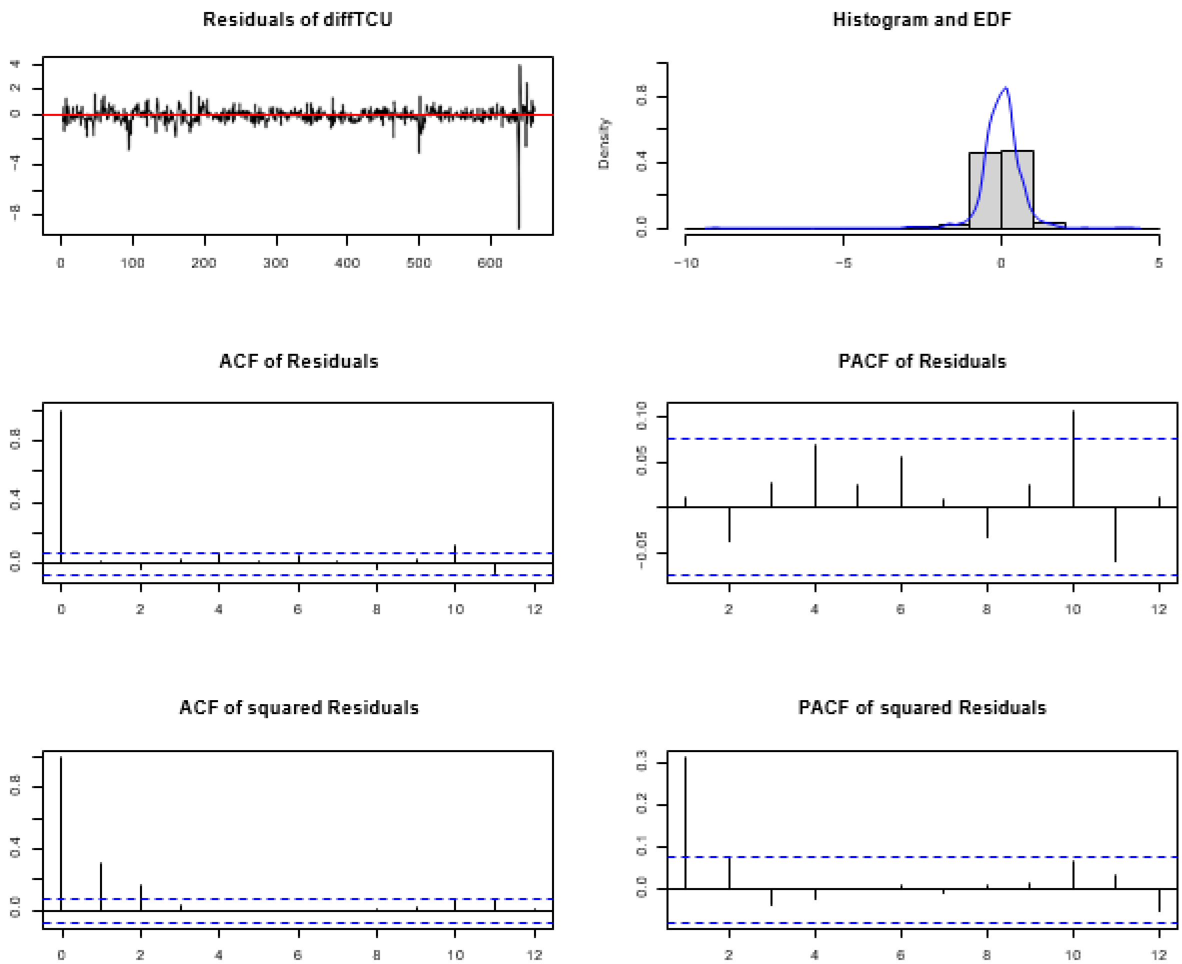
6.4 Diagnostic Testing for the estimated VAR models:

Serial correlation test

```
#Firstly, testing for serial correlation using a constant. The null hypothesis here
#is Ho = residuals do not have serial correlation.

serialtest <- serial.test(var.model.const, type = "PT.asymptotic")
serialtest

##
##  Portmanteau Test (asymptotic)
##
```

```
## data:  Residuals of VAR object var.model.const
## Chi-squared = 273.31, df = 60, p-value < 2.2e-16

serialtest <- serial.test(var.model.const, type = "PT.adjusted")
serialtest

##
##  Portmanteau Test (adjusted)
##
## data:  Residuals of VAR object var.model.const
## Chi-squared = 277.87, df = 60, p-value < 2.2e-16

serialtest <- serial.test(var.model.const, type = "BG")
serialtest

##
##  Breusch-Godfrey LM test
##
## data:  Residuals of VAR object var.model.const
## Chi-squared = 29.427, df = 20, p-value = 0.07968

serialtest <- serial.test(var.model.const, type = "ES")
serialtest

##
##  Edgerton-Shukur F test
##
## data:  Residuals of VAR object var.model.const
## F statistic = 1.4735, df1 = 20, df2 = 1294, p-value = 0.08126

plot(serialtest)
```


Residuals of diffInflation
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals

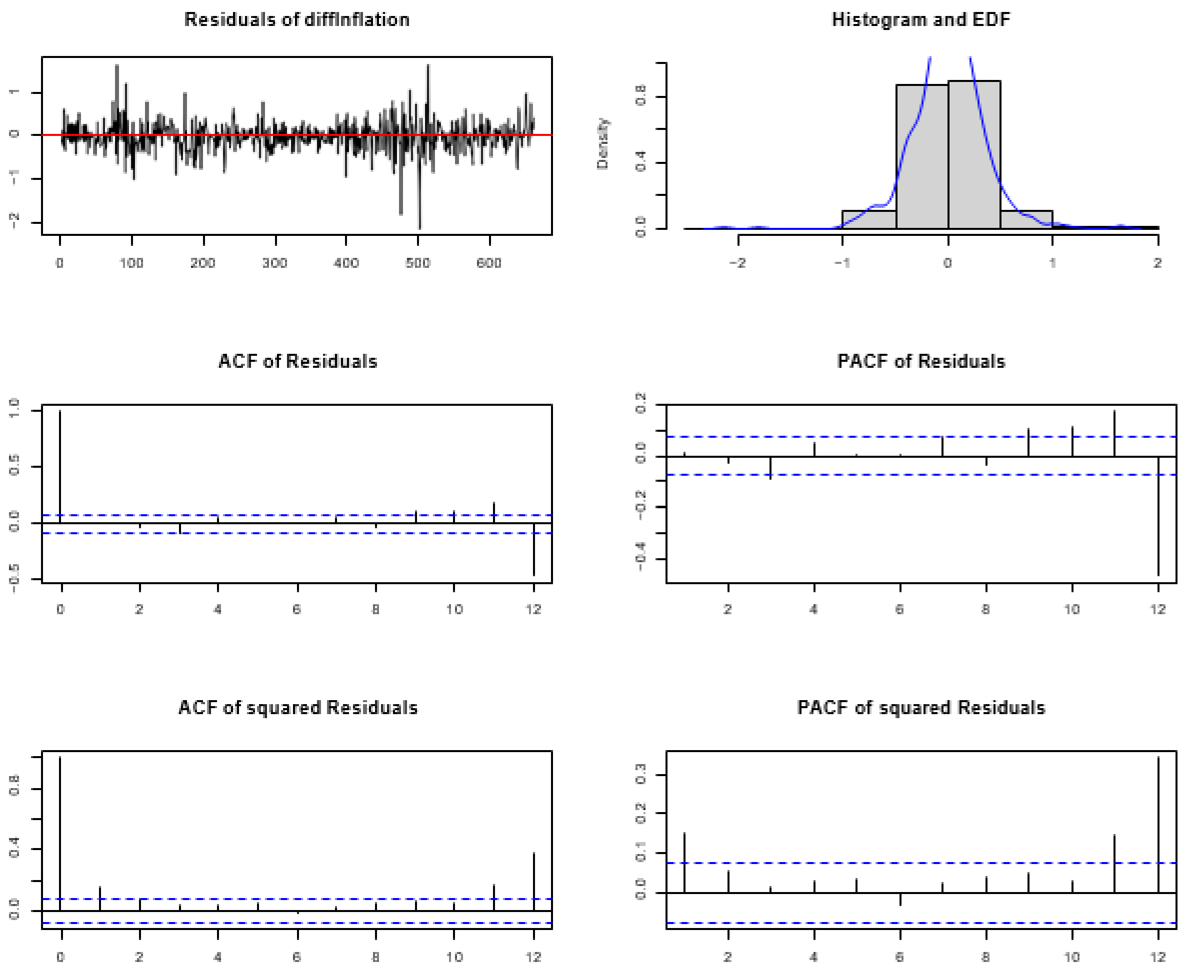


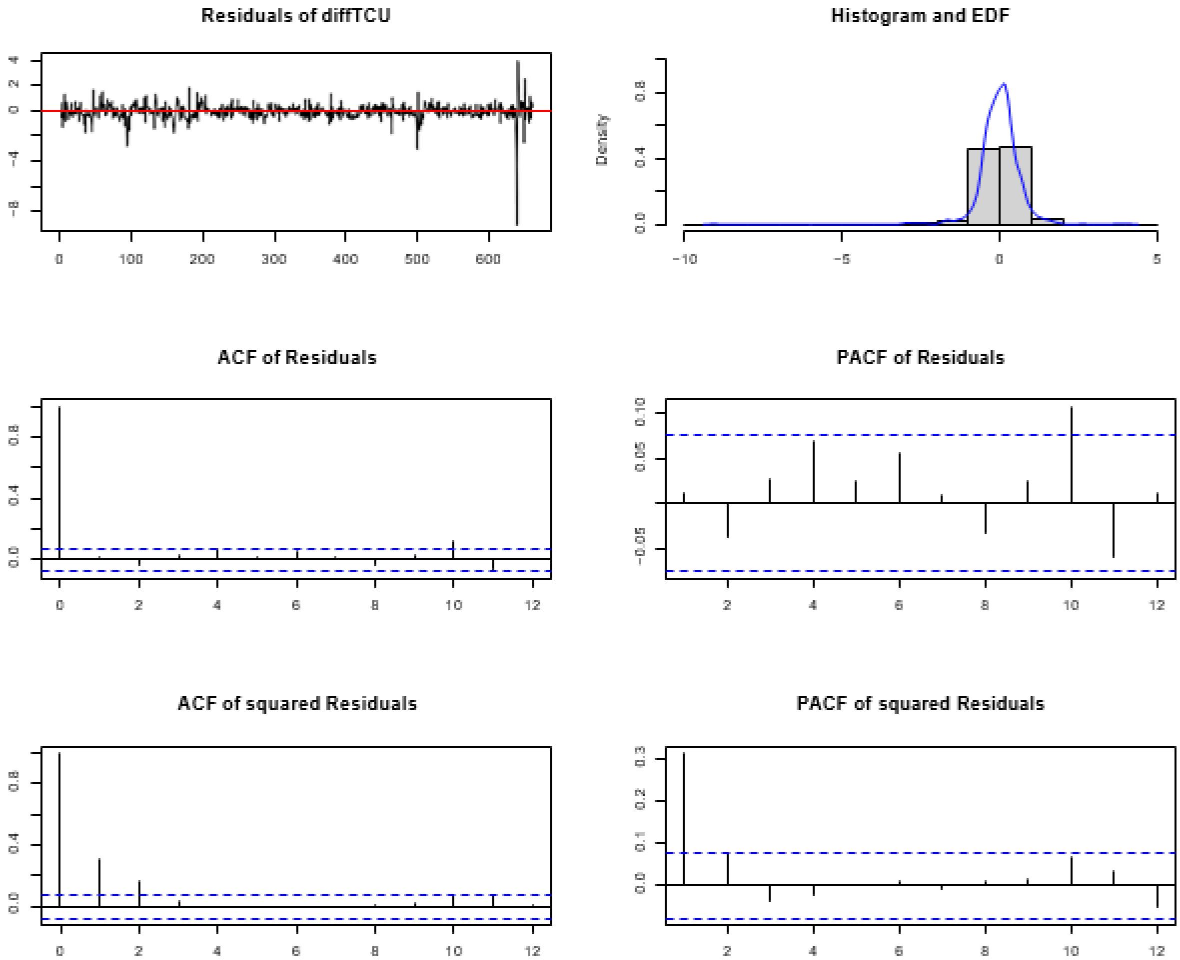
The p-values in the Portmanteau tests are both zero, so the null hypothesis of no serial correlation is rejected, while the Breusch-Godfrey and Edgerton-Shukur tests fail to reject the null hypothesis of no serial correlation at the 5%, but do so at the 10%. The tests show presence of serial correlation but the results are not convincing.

Normality test

```
#Next is to check for normality in the residuals by employing the Jarque-Baquera
#test.

normalitytest <- normality.test(var.ordered.const)
normalitytest

## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object var.ordered.const
## Chi-squared = 50247, df = 4, p-value < 2.2e-16
##
```

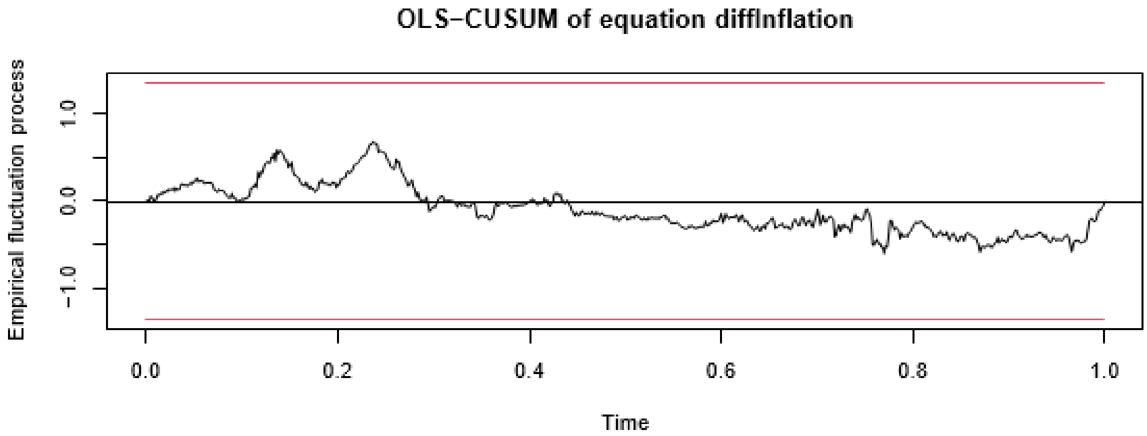
```
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object var.ordered.const
## Chi-squared = 952.78, df = 2, p-value < 2.2e-16
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object var.ordered.const
## Chi-squared = 49294, df = 2, p-value < 2.2e-16

# The p-value is zero, so it can be stated that the residuals are not normally distributed.

plot(normalitytest)
```

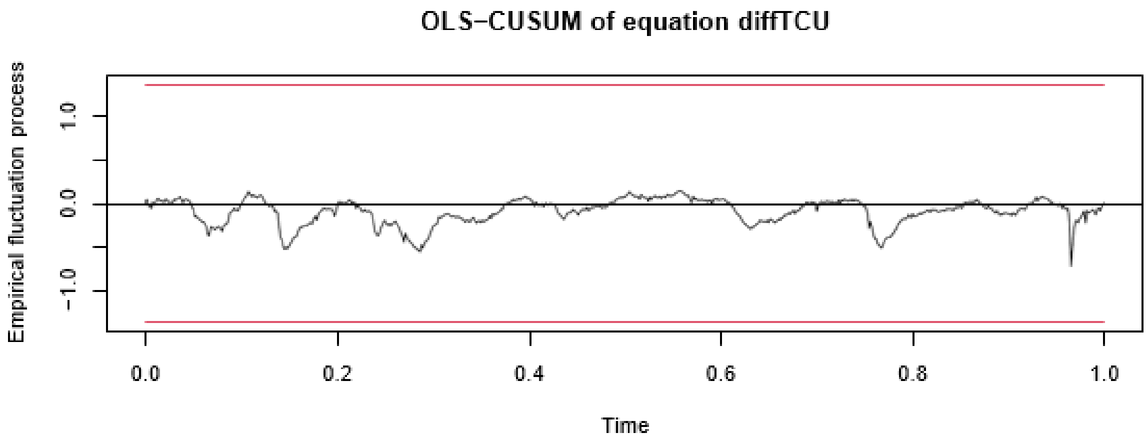

Residuals of diffInflation
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals



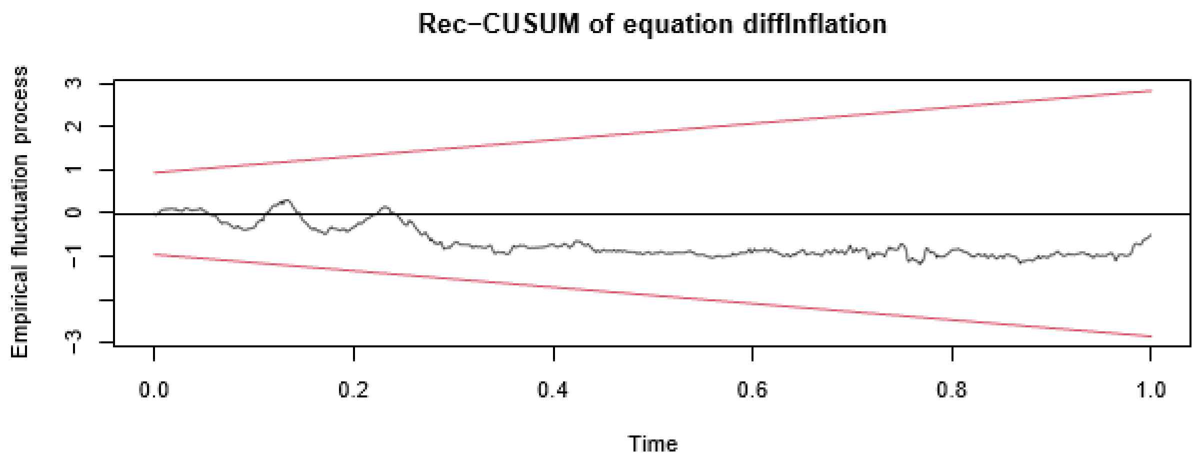
Stability

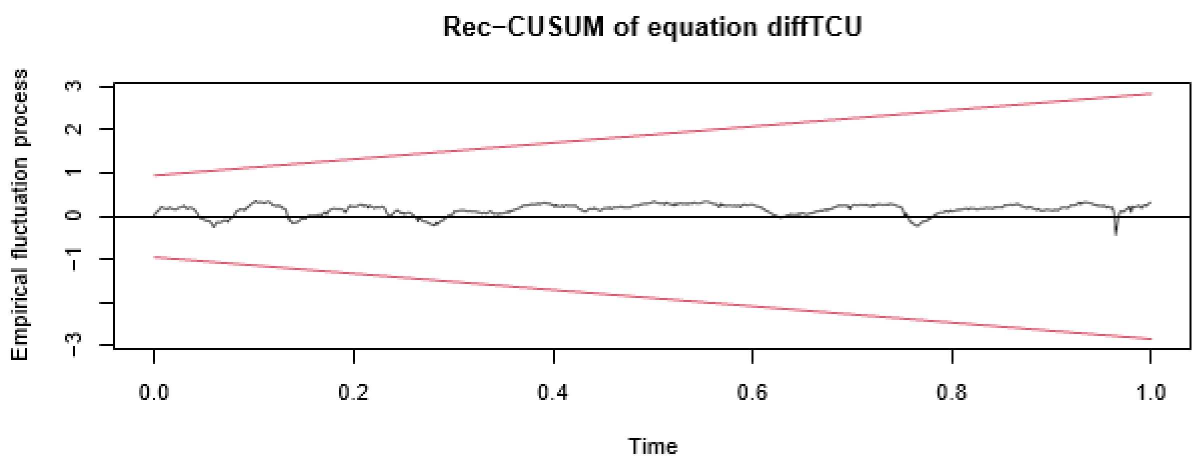
```
#Finally, let us check for the stability of the coefficients:

var.stabil <- stability(var.model.const, type = "OLS-CUSUM", dynamic = TRUE)
plot(var.stabil)
```



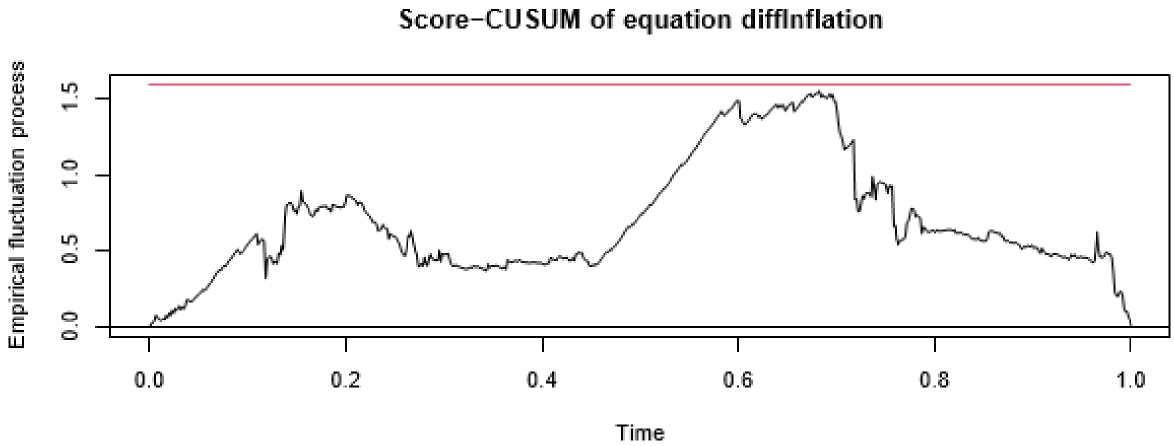


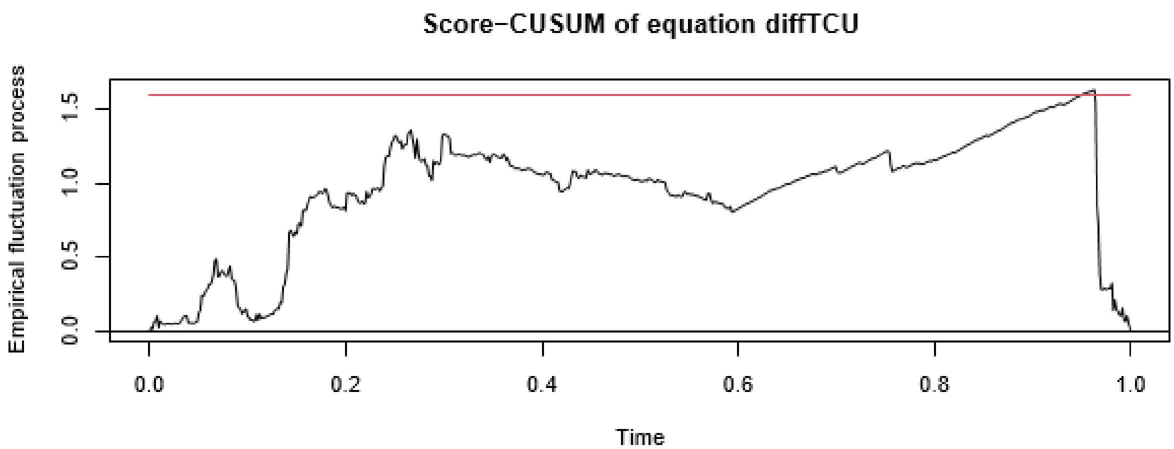

```
var.stabil <- stability(var.model.const, type = "Rec-CUSUM", dynamic = TRUE)
plot(var.stabil)
```






```
var.stabil <- stability(var.model.const, type = "Score-CUSUM")
plot(var.stabil)
```

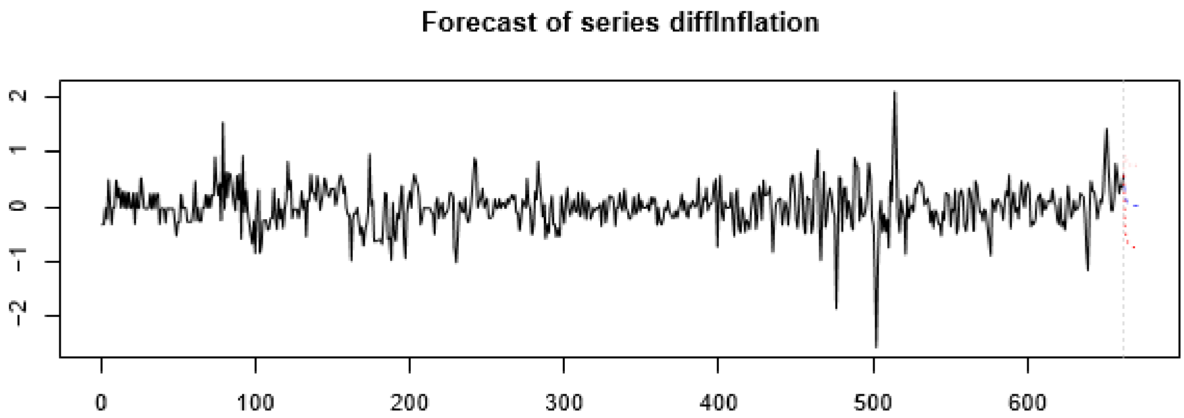


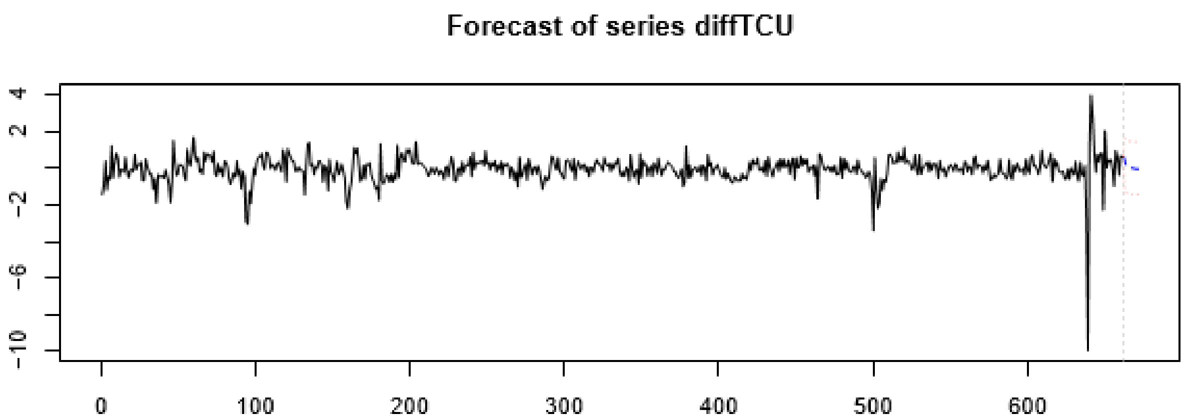


No issues in the stability of the coefficients, so the next step is to compute the Out-of-sample prediction.

Out-of-sample Prediction

```
#Number of periods ahead = 10.

var.prd.const <- predict(var.model.const, n.ahead = 10, ci = 0.95)
plot(var.prd.const)
```





The graphs above show a prediction of the two series for the next 10 months. The blue line represents the predicted value, while the red line the confidence interval.

7. ARIMA MODELS FOR FORECASTING

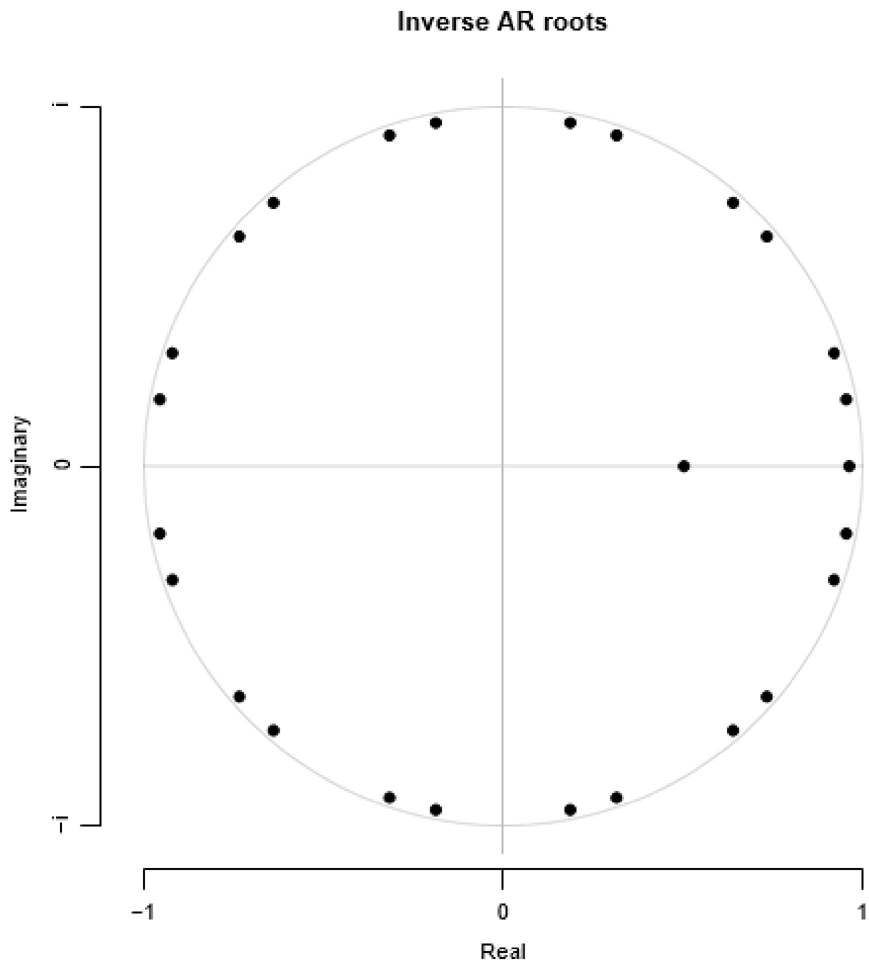
The forecasting model chosen in this project is the ARIMA model. Because the ARIMA model aims at using one variable and its own past values to identify how the variable can predict itself in the future, it is required to choose one time series at the time. Even if the time series has a unit root, it is possible to proceed using the time series in levels, as the purpose of the model is prediction and not to establish causality. Inflation has been chosen as the variable to be predicted, and now the accuracy of the model is tested.

```
#Transform the data set from first differences to levels once again
data.set = na.omit(
ts.intersect(
Inflation,
TCU,
```

```
dframe=TRUE))

Inflation   = ts( data.set$Inflation,    start=c(1967, 1),  frequency=12)
TCU         = ts( data.set$TCU,          start=c(1967, 1),  frequency=12)

data.set=ts(data.set, start=c(1967, 1), frequency=12)
```

7.1 Automatic ARIMA model selection

```
#The Automatic ARIMA and SARIMA model selection is employed to select the best
#ARIMA model according to either AIC, AICc or BIC value. The function conducts a
#search over possible models within the constraints provided.

arima.model = forecast::auto.arima(w,

D = 1,
stationary = FALSE,
ic = c("aicc", "aic", "bic"),
stepwise = FALSE,
approximation = FALSE,
seasonal = TRUE,
allowdrift = TRUE   )

arima.model

## Series: w
## ARIMA(2,0,0)(2,1,0)[12]
##
## Coefficients:
##          ar1      ar2     sar1     sar2
##       1.4654  -0.4838  -0.9813  -0.5205
## s.e.  0.0348   0.0348   0.0341   0.0336
##
## sigma^2 = 0.15:  log likelihood = -312.71
## AIC=635.41   AICc=635.5   BIC=657.8

plot(arima.model)
```

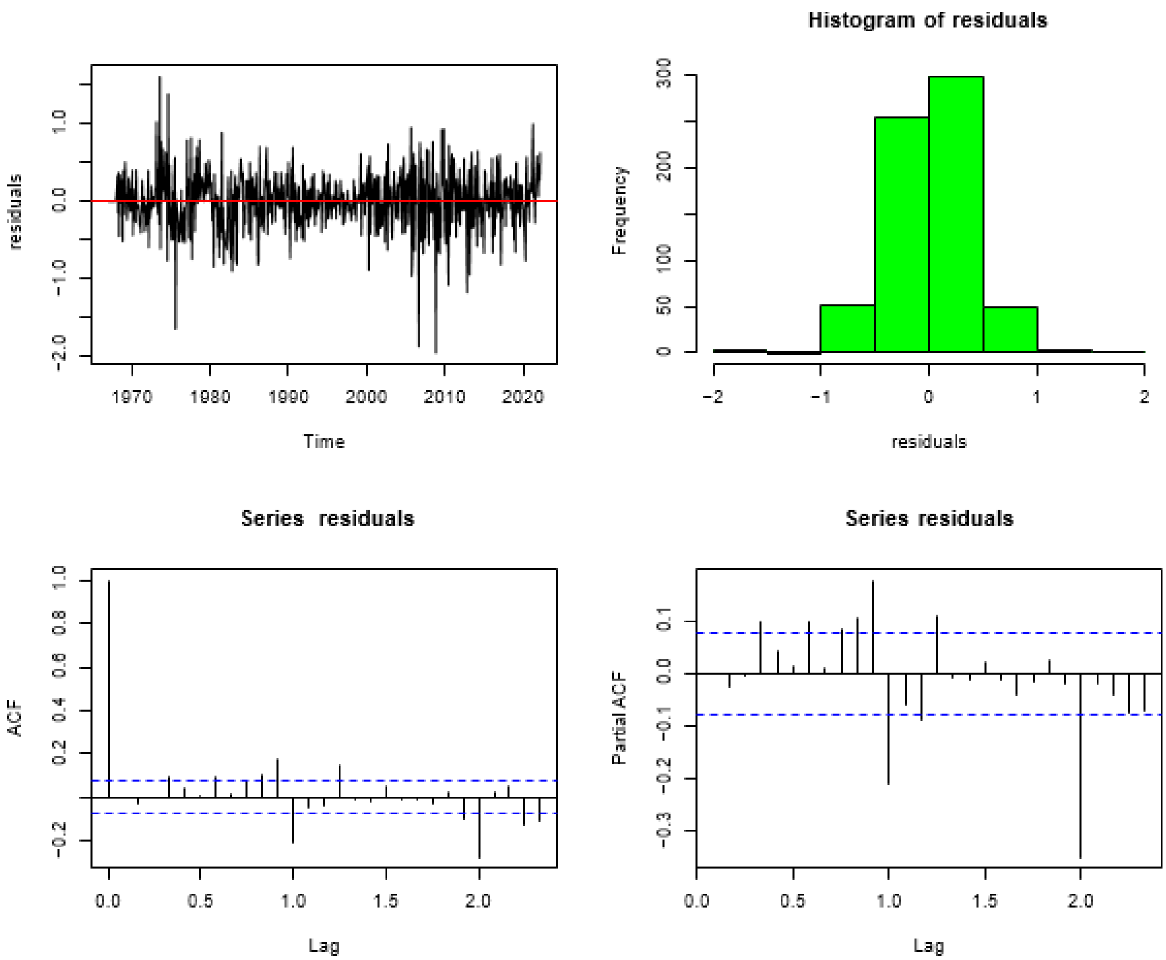


Auto.arima selects ARIMA(2,0,0)(2,1,0)[12] for the Inflation series, using seasonal dummies monthly. Moreover, as can be seen from the plot above, the unit roots are within the circle. The estimated ARIMA model to forecast future values of the time series can now be tested.

7.2 Tests

```
#After having selected the ARIMA(2,0,0)(2,1,0)[12] for the Inflation series, we
#conduct routine tests on the residuals.

#Plot the histogram of the residuals, together with the ACF and PACF of
#the residuals:

residuals = resid(arima.model)

par(mfrow = c(2, 2))
plot( residuals )
abline( h=0, col="red" )
```

```r
hist( residuals, col="green")
acf( residuals )
pacf( residuals )
```



Ljung-Box

```r
Box.test(resid(arima.model), lag = 2, fitdf=0)

##
##  Box-Pierce test
##
## data:  resid(arima.model)
## X-squared = 0.49627, df = 2, p-value = 0.7803

Box.test(resid(arima.model), lag = 1, type= "Ljung", fitdf=0)
```

```
##
##  Box-Ljung test
##
## data:  resid(arima.model)
## X-squared = 4.0951e-06, df = 1, p-value = 0.9984

Box.test(resid(arima.model), lag = 2, type= "Ljung", fitdf=0)

##
##  Box-Ljung test
##
## data:  resid(arima.model)
## X-squared = 0.49927, df = 2, p-value = 0.7791

Box.test(resid(arima.model), lag = 3, type= "Ljung", fitdf=0)

##
##  Box-Ljung test
##
## data:  resid(arima.model)
## X-squared = 0.51677, df = 3, p-value = 0.9152

Box.test(resid(arima.model), lag = 4, type= "Ljung", fitdf=0)

##
##  Box-Ljung test
##
## data:  resid(arima.model)
## X-squared = 7.0518, df = 4, p-value = 0.1332

Box.test(resid(arima.model), lag = 5, type= "Ljung", fitdf=0)

##
##  Box-Ljung test
##
## data:  resid(arima.model)
## X-squared = 8.2733, df = 5, p-value = 0.1418

Box.test(resid(arima.model), lag = 6, type= "Ljung", fitdf=0)

##
##  Box-Ljung test
##
## data:  resid(arima.model)
## X-squared = 8.3383, df = 6, p-value = 0.2144

# The ARIMA model has selected the series in first differences, so the residuals
#are not expected to be serially correlated. By running the Box-Ljung test, the p-
#values for the first 6 lags are greater than 0.01 failing to reject the null
#hypothesis of autocorrelation.
```

Shapiro-Wilk Normality Test

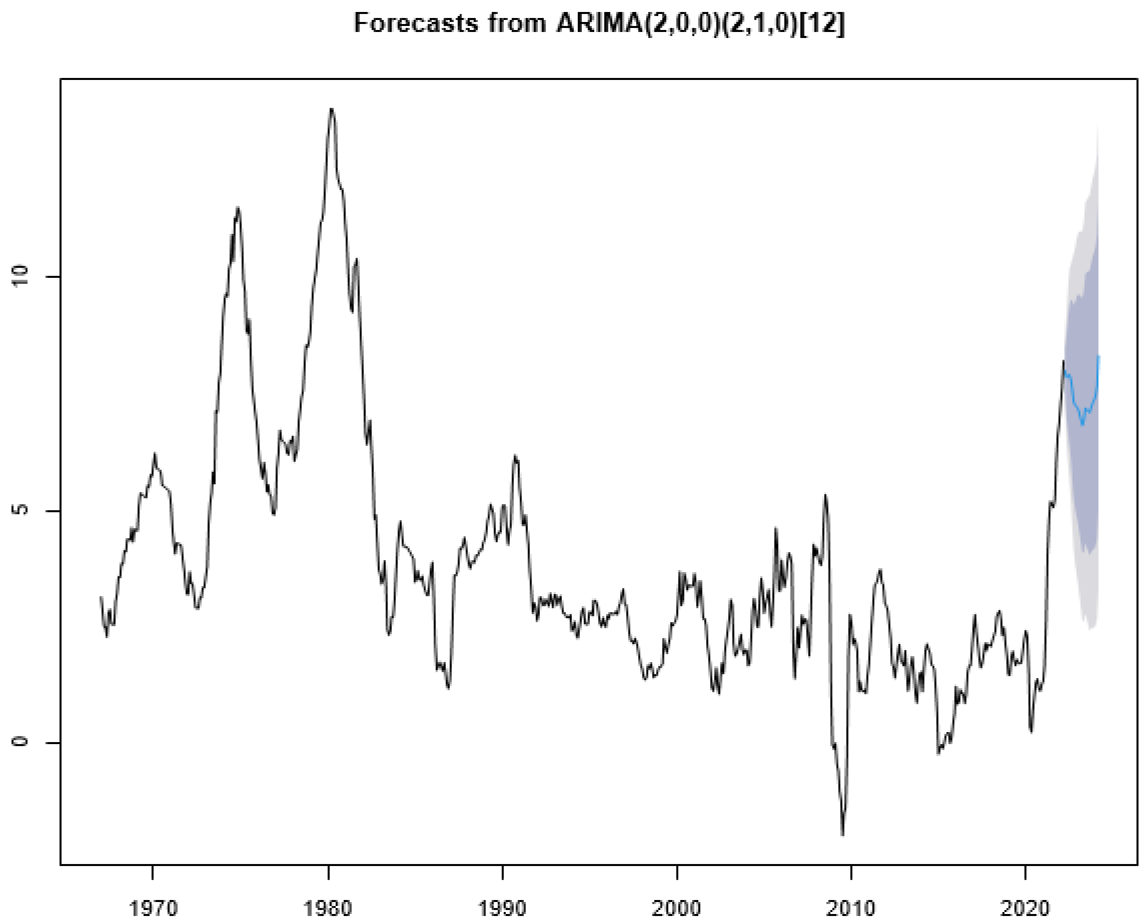
```
#Ho = series is normally distributed = kurtosis is zero and skewness is zero Ha =
#series is not normally distributed

shapiro.test(resid(arima.model))

##
##  Shapiro-Wilk normality test
##
## data:  resid(arima.model)
## W = 0.97353, p-value = 1.406e-09

# The Shapiro-Wilk test rejects the null hypothesis of normality, so the residuals
are not normally distributed.
```

Jarque-Bera Normality Test

```
#Ho = series is normally distributed = kurtosis is zero and skewness is zero Ha =
#series is not normally distributed Using package "tseries"

jarque.bera.test(residuals)

##
##  Jarque Bera Test
##
## data:  residuals
## X-squared = 208.92, df = 2, p-value < 2.2e-16

jarque.bera.test(resid(arima.model))

##
##  Jarque Bera Test
##
## data:  resid(arima.model)
## X-squared = 208.92, df = 2, p-value < 2.2e-16

# The Jarque-Bera test rejects the null hypothesis of normality in the residuals,
which confirms that the residuals are not normally distributed.
```

Out-of-sample forecasting

```
ARIMA.forecast = forecast::forecast(arima.model, h = 10)

plot(forecast::forecast(arima.model))
```

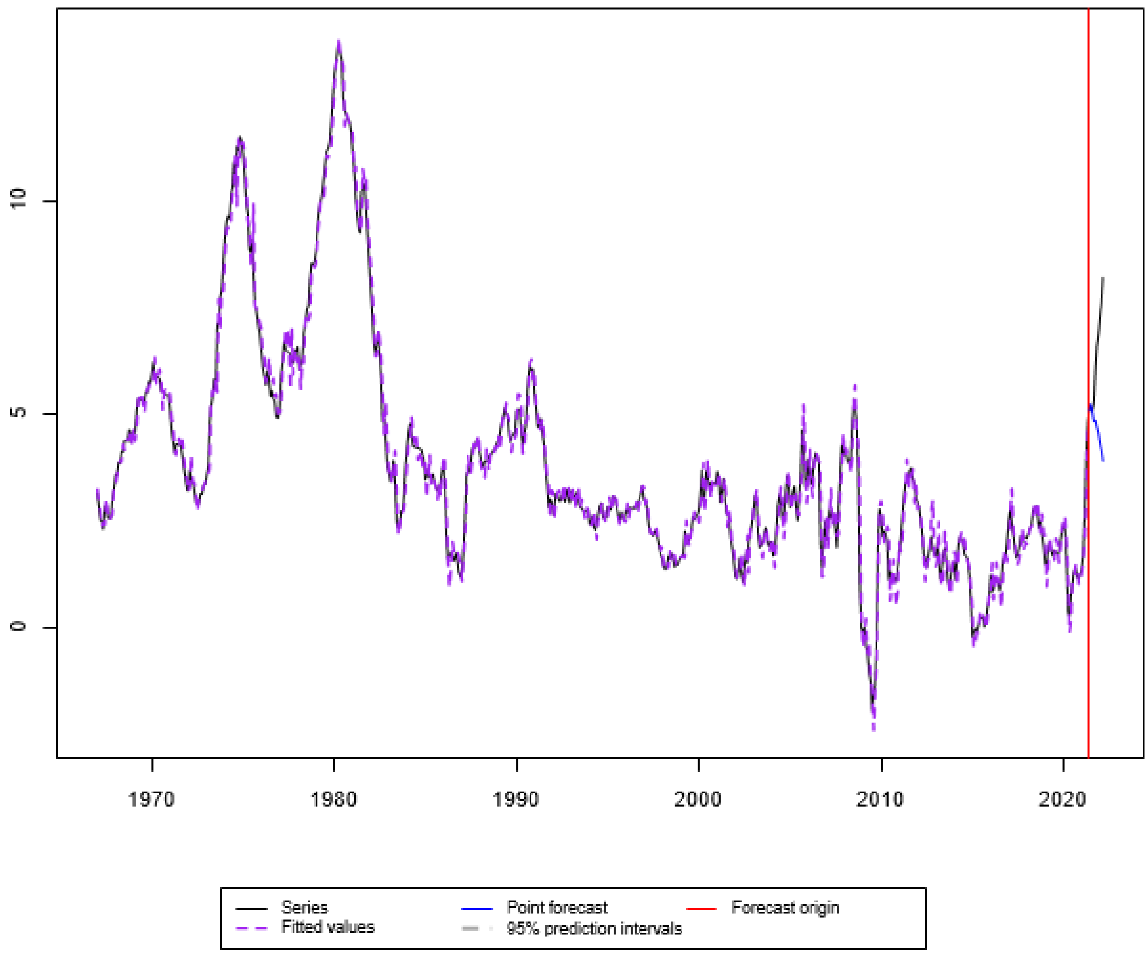


The plot above shows the prediction that we get with the model computed above. It predicts that inflation rate might decline in the next 4 months of roughly 2%, for then to rise back up close to 8%.

In-sample forecasting

```
library(smooth)

#The seasonal ARIMA model is now used to test whether the model can predict itself.
#The model excludes the last 10 months of the series, and tests whether can predict
#correctly the last 10 months of the series based on its own past values.

sarima.model = smooth::msarima(wl, orders=list(

ar = c(2,2),
i = c(0,1),
ma = c(1,0)),
lags = c(1,12),
h = 10,
holdout = TRUE)
```

```
summary(sarima.model)

## Time elapsed: 0.98 seconds
## Model estimated: SARIMA(2,0,1)[1](2,1,0)[12]
## Matrix of AR terms:
##         Lag 1  Lag 12
## AR(1)  1.4728 -1.0064
## AR(2) -0.4929 -0.5437
## Matrix of MA terms:
##         Lag 1
## MA(1) -0.0121
## Initial values were produced using backcasting.
##
## Loss function type: likelihood; Loss function value: 297.7593
## Error standard deviation: 0.3835
## Sample size: 653
## Number of estimated parameters: 6
## Number of degrees of freedom: 647
## Information criteria:
##      AIC     AICc      BIC     BICc
## 607.5186 607.6487 634.4081 634.8295
##
## Forecast errors:
## MPE: 22.9%; sCE: 428.4%; Asymmetry: 95%; MAPE: 23.3%
## MASE: 6.28; sMAE: 43.4%; sMSE: 33.3%; rMAE: 1.119; rRMSE: 1.203

values = sarima.model

greybox::graphmaker(wl,values$forecast,values$fitted,values$lower,values$upper,leve
l=0.95,legend=TRUE)
```



The model is not able to correctly predict the last 10 months that we discarded so can not be considered as a valid predictive model.

9. CONCLUSIONS

The results of the cointegration tests were ambiguous and contrasting, so it cannot be stated with certainty whether the series are cointegrated. Moreover, the ARDL models computed low R-squared, suggesting that the models are not correctly specified, and the residuals are not well behaved. This could also be the reason why the forecasting models turned out to be unreliable. It is suggested that using only one variable to predict Inflation Rate is not sufficient. On this note, on the other side, it could be observed from the Granger causality tests that Inflation might Granger cause TCU in first differences but not the opposite. This means that past values of inflation could predict TCU, but past values of TCU would not predict the behavior of Inflation. These conclusions suggest possible avenues for further research.

Optional part: PANEL DATA

1. INTRODUCTION

This section of the project aims at analysing the impact that Income Per Capita jointly with Safe Water Access, Health Expenditure, and Pregnant Women with anemia has on Maternal Mortality Rate across the given countries over the years. There are over 1,300 indicators available from WDI but for the purpose of this project we are only interested in the following indicators:

NY.GDP.MKTP.KD.ZG GDP growth (annual %)

SH.H2O.SAFE.ZS Improved watersource (% of population with access)

SH.XPD.TOTL.ZS Health expenditure,total (% of GDP)

SH.STA.MMRT Maternal mortality ratio (modeled estimate per 100,000 live births)

S H.PRG.ANEM Prevalence of anemia among pregnant women (%)

SH.MED.NUMW.P3 Nurses and midwives (per 1,000 people)

This dataset is used to understand the causes of maternal mortality rates around the world. It will be calculated how maternal mortality rates change over time and across countries and what factors are related to this change.

```
#Load the necessary packages:
library(lmtest)
library(texreg)
library(tidyr)
library(dplyr)
library(pdfetch)
library(foreign)
library(car)
library(gplots)
library(tseries)
library(sjPlot)
library(huxtable)
library(ivreg)
library(plm)
```

2. LOADING THE DATA

The variables available in the WDI file are the following:

Country (total of 192 countries)

Year (from 1960 to 2015)

IncomePerCapita (constant 2010 US)

GDPGrowth (annual %)

GDPPerCapita (constant 2010 US$)

OilRents (oil rents as % of GDP)

SafeWaterAccess (% of population with access to safe water)

NursesMidwives (nurses and midwives per 1,000 people)

PregnantWomenWithAnemia (pregnant women with anemia in %)

MaternalMortality (maternal mortality ratio, modeled estimate, per 100,000 live births)
HealthExpenditure (total health expenditure as % of GDP)

PovertyGap (poverty gap at $1.90 a day, 2011 PPP, in %)

InfantMortality (mortality rate, infant, per 1,000 live births)

Note that this is an unbalanced panel, as the dataset in use is missing a high number of data.

```
#View the countries covered by the wdi dataset
unique(wdi$Country)

##   [1] "Afghanistan"                    "Albania"
##   [3] "Algeria"                        "Andorra"
##   [5] "Angola"                         "Antigua and Barbuda"
##   [7] "Argentina"                      "Armenia"
##   [9] "Australia"                      "Austria"
##  [11] "Azerbaijan"                     "Bahamas, The"
##  [13] "Bahrain"                        "Bangladesh"
##  [15] "Barbados"                       "Belarus"
##  [17] "Belgium"                        "Belize"
##  [19] "Benin"                          "Bhutan"
##  [21] "Bolivia"                        "Bosnia and Herzegovina"
##  [23] "Botswana"                       "Brazil"
##  [25] "Brunei Darussalam"              "Bulgaria"
##  [27] "Burkina Faso"                   "Burundi"
##  [29] "Cabo Verde"                     "Cambodia"
##  [31] "Cameroon"                       "Canada"
##  [33] "Cayman Islands"                 "Central African Republic"
##  [35] "Chad"                           "Chile"
##  [37] "China"                          "Colombia"
##  [39] "Comoros"                        "Congo, Dem. Rep."
##  [41] "Congo, Rep."                    "Costa Rica"
##  [43] "Cote d'Ivoire"                  "Croatia"
##  [45] "Cuba"                           "Cyprus"
##  [47] "Czech Republic"                 "Denmark"
##  [49] "Djibouti"                       "Dominica"
##  [51] "Ecuador"                        "Egypt, Arab Rep."
##  [53] "El Salvador"                    "Equatorial Guinea"
##  [55] "Eritrea"                        "Estonia"
```

```
##  [57] "Ethiopia"                         "Fiji"
##  [59] "Finland"                          "France"
##  [61] "Gabon"                            "Gambia, The"
##  [63] "Georgia"                          "Germany"
##  [65] "Ghana"                            "Greece"
##  [67] "Grenada"                          "Guatemala"
##  [69] "Guinea"                           "Guinea-Bissau"
##  [71] "Guyana"                           "Haiti"
##  [73] "Honduras"                         "Hungary"
##  [75] "Iceland"                          "India"
##  [77] "Indonesia"                        "Iran, Islamic Rep."
##  [79] "Iraq"                             "Ireland"
##  [81] "Israel"                           "Italy"
##  [83] "Jamaica"                          "Japan"
##  [85] "Jordan"                           "Kazakhstan"
##  [87] "Kenya"                            "Kiribati"
##  [89] "Korea, Dem. Rep."                 "Korea, Rep."
##  [91] "Kuwait"                           "Kyrgyz Republic"
##  [93] "Lao PDR"                          "Latvia"
##  [95] "Lebanon"                          "Lesotho"
##  [97] "Liberia"                          "Libya"
##  [99] "Liechtenstein"                    "Lithuania"
## [101] "Luxembourg"                       "Macedonia, FYR"
## [103] "Madagascar"                       "Malawi"
## [105] "Malaysia"                         "Maldives"
## [107] "Mali"                             "Malta"
## [109] "Marshall Islands"                 "Mauritania"
## [111] "Mauritius"                        "Mexico"
## [113] "Micronesia, Fed. Sts."            "Moldova"
## [115] "Monaco"                           "Mongolia"
## [117] "Montenegro"                       "Morocco"
## [119] "Mozambique"                       "Myanmar"
## [121] "Namibia"                          "Nepal"
## [123] "Netherlands"                      "New Zealand"
## [125] "Nicaragua"                        "Niger"
## [127] "Nigeria"                          "Norway"
## [129] "Oman"                             "Pakistan"
## [131] "Palau"                            "Panama"
## [133] "Papua New Guinea"                 "Paraguay"
## [135] "Peru"                             "Philippines"
## [137] "Poland"                           "Portugal"
## [139] "Qatar"                            "Romania"
## [141] "Russian Federation"               "Rwanda"
## [143] "Samoa"                            "San Marino"
## [145] "Sao Tome and Principe"            "Saudi Arabia"
## [147] "Senegal"                          "Serbia"
## [149] "Seychelles"                       "Sierra Leone"
## [151] "Singapore"                        "Slovak Republic"
## [153] "Slovenia"                         "Solomon Islands"
## [155] "Somalia"                          "South Africa"
```

```
## [157] "South Sudan"                      "Spain"
## [159] "Sri Lanka"                        "St. Kitts and Nevis"
## [161] "St. Lucia"                        "St. Vincent and the Grenadines"
## [163] "Sudan"                            "Suriname"
## [165] "Swaziland"                        "Sweden"
## [167] "Switzerland"                      "Syrian Arab Republic"
## [169] "Tajikistan"                       "Tanzania"
## [171] "Thailand"                         "Timor-Leste"
## [173] "Togo"                             "Tonga"
## [175] "Trinidad and Tobago"              "Tunisia"
## [177] "Turkey"                           "Turkmenistan"
## [179] "Tuvalu"                           "Uganda"
## [181] "Ukraine"                          "United Arab Emirates"
## [183] "United Kingdom"                   "United States"
## [185] "Uruguay"                          "Uzbekistan"
## [187] "Vanuatu"                          "Venezuela, RB"
## [189] "Vietnam"                          "Yemen, Rep."
## [191] "Zambia"                           "Zimbabwe"
```

```
#Count the number of countries covered by the wdi dataset
length(unique(wdi$Country))
```

```
## [1] 192
```

The dataset countains data for 192 countries around the world. It follows below a table with all the key statistics of each variable in the dataset such as max, min, average, median etc.

```
#Summary of the key statistics of the dataset (max, min, average, median etc.)
summary(wdi)
```

```
##    Country               Year      IncomePerCapita     GDPGrowth
##  Length:10752       Min.   :1960   Min.   : -143.6   Min.   :-64.047
##  Class :character   1st Qu.:1974   1st Qu.:  631.0   1st Qu.:  1.442
##  Mode  :character   Median :1988   Median : 2161.2   Median :  3.954
##                     Mean   :1988   Mean   : 7288.3   Mean   :  3.916
##                     3rd Qu.:2001   3rd Qu.: 8838.2   3rd Qu.:  6.427
##                     Max.   :2015   Max.   :62248.6   Max.   :189.830
##                                    NA's   :6472      NA's   :2787
##   GDPPerCapita         OilRents       SafeWaterAccess  NursesMidwives
##  Min.   :    68.57   Min.   : 0.000   Min.   : 13.20   Min.   : 0.043
##  1st Qu.:   701.89   1st Qu.: 0.000   1st Qu.: 74.30   1st Qu.: 0.981
##  Median :  2352.74   Median : 0.000   Median : 91.20   Median : 3.660
##  Mean   :  8750.12   Mean   : 4.838   Mean   : 83.73   Mean   : 4.572
##  3rd Qu.:  9679.92   3rd Qu.: 1.503   3rd Qu.: 98.60   3rd Qu.: 6.419
##  Max.   :158602.52   Max.   :86.982   Max.   :100.00   Max.   :95.480
##  NA's   :2756        NA's   :3992     NA's   :6009     NA's   :10089
##  PregnantWomenWithAnemia MaternalMortality HealthExpenditure   PovertyGap
##  Min.   :12.00           Min.   :   3.0    Min.   : 0.715    Min.   : 0.400
##  1st Qu.:28.20           1st Qu.:  20.0    1st Qu.: 4.315    1st Qu.: 3.925
##  Median :34.30           Median :  78.0    Median : 5.815    Median : 8.100
##  Mean   :37.35           Mean   : 254.6    Mean   : 6.217    Mean   :11.645
```

```
##   3rd Qu.:44.83          3rd Qu.: 397.2    3rd Qu.: 7.640    3rd Qu.:16.000
##   Max.   :68.90          Max.   :2900.0    Max.   :22.533    Max.   :45.300
##   NA's   :6704           NA's   :6072      NA's   :7228      NA's   :10510
##   InfantMortality
##   Min.   :  1.50
##   1st Qu.: 15.60
##   Median : 40.70
##   Mean   : 54.38
##   3rd Qu.: 83.20
##   Max.   :276.90
##   NA's   :1369
```

3. POOLED OLS, FE, AND RE MODELS

Now we can estimate each model one at a time. For our analysis, we will include in the regression IncomePerCapita. We expect that increasing income per capita will reduce maternal mortality rate.

3.1 Pooled OLS Model

The first step is to run a Pooled OLS model that does not take into account the panel heterogeneity. This ignore both the panel and fixed effects, and the table that will be generated will simply stack the data and run OLS. We will then compare this simple OLS model with the Fixed Effects (FE) and Random Effects (RE) models thaat will be introduced later.

```
pooled_OLS = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita,
data   = wdi,
index  = c("Country", "Year"),
model  = "pooling")


summary(pooled_OLS)

## Pooling Model
##
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita, data = wdi, model = "pooling",
##     index = c("Country", "Year"))
##
## Unbalanced Panel: n = 161, T = 1-17, N = 2076
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -541.1501  -72.2114   -3.2121   49.7978 1791.4759
##
```

```
## Coefficients:
##                           Estimate  Std. Error  t-value Pr(>|t|)
## (Intercept)              5.9534e+02  5.0486e+01  11.7921   <2e-16 ***
## SafeWaterAccess         -1.1344e+01  3.7973e-01 -29.8743   <2e-16 ***
## HealthExpenditure        2.9464e+01  2.1764e+00  13.5376   <2e-16 ***
## PregnantWomenWithAnemia  1.1644e+01  5.5042e-01  21.1553   <2e-16 ***
## IncomePerCapita         -8.3848e-05  4.7592e-04  -0.1762   0.8602
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    249470000
## Residual Sum of Squares: 77869000
## R-Squared:      0.68786
## Adj. R-Squared: 0.68726
## F-statistic: 1140.97 on 4 and 2071 DF, p-value: < 2.22e-16
```

After running the pooled OLS model, the first thing that comes at our attention is the sign of the variable health expenditure: we expect a negative sign for this coefficient, as intuitively increasing health expenditure should decrease maternal mortality rate. The coefficient shows a positive sign which is arguably counterintuititve. Moreover, all variables except Income Per Capita are significant at the 0.1% level. This could be because an endogeneity problem might persist, or the pooled OLS model could not be specified correctly. Further test are required.

3.2 Fixed Effects Model

Our dataset includes a number of indicators that we believe could be related to maternal mortality rates such as healthcare expenditure, proportion of population with access to safe water. But we also know that a number of other factors could have a profound effect on maternal mortality rates but they are either not easily available or not observable at all. Some of these factors could include a lack of awareness of preventable diseases and attitudes towards prevention or prenatal care that vary greatly from country to country but don't change over time within the same country. Without access to datasets controlling for these factors, our model would suffer from omitted variable bias. But a fixed effect (FE) model allows us to control for these variables that are constant over time. Firstly, we run the within model, and the effect is only controlling for the individual effects and ignoring the time effects. Nevertheless, it will partially controlling for endogeneity problem by taking into account the unobserved effect.

```
fixed_effects = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita,
                    data   = wdi,
                    index  = c("Country", "Year"),
                    model  = "within",
                    effect = "individual")

summary(fixed_effects)
```

```
## Oneway (individual) effect Within Model
##
```

```
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita, data = wdi, effect =
"individual",
##     model = "within", index = c("Country", "Year"))
##
## Unbalanced Panel: n = 161, T = 1-17, N = 2076
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -564.5439  -12.9162    0.1942   11.6569  535.8183
##
## Coefficients:
##                            Estimate  Std. Error  t-value  Pr(>|t|)
## SafeWaterAccess         -8.39565735  0.54999320 -15.2650 < 2.2e-16 ***
## HealthExpenditure       -7.13888589  1.53446455  -4.6524 3.507e-06 ***
## PregnantWomenWithAnemia  6.08637540  0.78843707   7.7195 1.869e-14 ***
## IncomePerCapita          0.00325803  0.00097414   3.3445 0.0008402 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    10549000
## Residual Sum of Squares: 7206900
## R-Squared:      0.31681
## Adj. R-Squared: 0.25818
## F-statistic: 221.541 on 4 and 1911 DF, p-value: < 2.22e-16
```

By computing the country FE model we get the expected signs for the betas, and Income Per Capita coefficient is now significant at the 0.1% significance level too although still shows a positive sign. The adjusted R squared results quite low.

In addition to country FE, a number of factors could affect maternal mortality rates that are not specific to each country. For example, development of new vaccines or medicines to fight infections, or investments in awareness campaigns by aid organizations such as WHO about preventable diseases likely vary over time but have similar effects around the world. We now run a within Fixed Effect model that controls for the time effect.

```
time_effects = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita,
                   data = wdi,
                   index = c("Country", "Year"),
                   model = "within",
                   effect = "time")

summary(time_effects)

## Oneway (time) effect Within Model
##
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
```

```
##     PregnantWomenWithAnemia + IncomePerCapita, data = wdi, effect = "time",
##     model = "within", index = c("Country", "Year"))
##
## Unbalanced Panel: n = 161, T = 1-17, N = 2076
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -554.429  -71.522   -3.712   48.185 1806.502
##
## Coefficients:
##                            Estimate  Std. Error  t-value Pr(>|t|)
## SafeWaterAccess         -1.1353e+01  3.8119e-01 -29.7842   <2e-16 ***
## HealthExpenditure        2.9439e+01  2.2024e+00  13.3672   <2e-16 ***
## PregnantWomenWithAnemia  1.1662e+01  5.5271e-01  21.0996   <2e-16 ***
## IncomePerCapita         -8.5546e-05  4.7885e-04  -0.1786   0.8582
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    247430000
## Residual Sum of Squares: 77689000
## R-Squared:      0.68602
## Adj. R-Squared: 0.68297
## F-statistic: 1122.5 on 4 and 2055 DF, p-value: < 2.22e-16
```

But with time fixed effects, the model does not distinguish between observations across different countries. And we get unexpected signs for the betas. When we are not using country fixed effects, the fluctuations in the series over time make it seem as if an increase in HealthExpenditure is correlated with increase in MaternalMortality, but in fact it's just an artifact of time fixed effects. On the other side, it seems that an increase in income per capita over time contributes to lower maternal mortality, which is what we expect, but the coefficient is not statistically significant.

In order to control for both country and time fixed effects at the same time, we need to estimate a model using the the two ways effect:

```
twoway_effects = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita,
data = wdi,
index = c("Country", "Year"),
model = "within",
effect = "twoways")

summary(twoway_effects)

## Twoways effects Within Model
##
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita, data = wdi, effect = "twoways",
##     model = "within", index = c("Country", "Year"))
##
```

```
## Unbalanced Panel: n = 161, T = 1-17, N = 2076
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -567.582  -14.088    0.000   12.432  539.814
##
## Coefficients:
##                            Estimate Std. Error  t-value  Pr(>|t|)
## SafeWaterAccess          -7.9821605  0.6187568 -12.9003 < 2.2e-16 ***
## HealthExpenditure        -6.8293578  1.6087862  -4.2450 2.291e-05 ***
## PregnantWomenWithAnemia   5.5100821  0.9245709   5.9596 3.008e-09 ***
## IncomePerCapita           0.0042492  0.0011730   3.6225 0.0002994 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    8589400
## Residual Sum of Squares: 7173800
## R-Squared:      0.16481
## Adj. R-Squared: 0.085483
## F-statistic: 93.4894 on 4 and 1895 DF, p-value: < 2.22e-16
```

Once we include both individual (country) and time FE, we get the expected signs for the betas. Now we can create a consolidated table comparing the 3 FE models side-by-side and include the Pooled OLS model for comparison. The coefficient of all variables are statistically significant at the 0.1% significance level, but Income Per Capita still presents a positive sign. Nevertheless, the coefficient is close to zero, which might suggest thaat increasing income per capita might not have a direct effect on Maternal Mortality Rate.

```
screenreg(list(pooled_OLS,
fixed_effects,
time_effects,
twoway_effects),

custom.model.names = c("Pooled OLS",
"Country FE",
"Time FE",
"Two-way FE"))

##
## ===========================================================================
##                          Pooled OLS   Country FE   Time FE      Two-way FE
## ---------------------------------------------------------------------------
## (Intercept)               595.34 ***
##                           (50.49)
## SafeWaterAccess           -11.34 ***    -8.40 ***   -11.35 ***    -7.98 ***
##                            (0.38)       (0.55)       (0.38)       (0.62)
## HealthExpenditure          29.46 ***    -7.14 ***    29.44 ***    -6.83 ***
##                            (2.18)       (1.53)       (2.20)       (1.61)
## PregnantWomenWithAnemia    11.64 ***     6.09 ***    11.66 ***     5.51 ***
##                            (0.55)       (0.79)       (0.55)       (0.92)
```

```
## IncomePerCapita              -0.00         0.00 ***    -0.00         0.00 ***
##                              (0.00)       (0.00)       (0.00)       (0.00)
## ---------------------------------------------------------------------------
## R^2                            0.69         0.32         0.69         0.16
## Adj. R^2                       0.69         0.26         0.68         0.09
## Num. obs.                   2076         2076         2076         2076
## ===========================================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

All four explanatory variables are statistically significant when controlling for individual fixed effects and when controlling for both time and fixed effect, but income per capita dis not significant when controlling for time effect and in the pooled OLS. Out of the four models, the Two way FE model seems to be the most meaningful. The coefficients for our explanatory variable in the two-way fixed effect model are close to the country fixed effects indicating that these factors vary greatly across countries than they do across time.

3.3 Random Effects Model

In the RE model, the country-specific components are random and uncorrelated with the regressors. In this case we will compute the "Two-way RE" model, with both individual and time effects. The Error components will be Idiosyncratic, Individual, and Time.

```
RE_twoway = plm( MaternalMortality ~ SafeWaterAccess + HealthExpenditure + 
PregnantWomenWithAnemia + IncomePerCapita,

data   = wdi,
index  = c("Country", "Year"),
model  = "random",
effect = "twoways",
random.method = "swar" )

summary(RE_twoway)

## Twoways effects Random Effect Model 
##    (Swamy-Arora's transformation)
## 
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure + 
##     PregnantWomenWithAnemia + IncomePerCapita, data = wdi, effect = "twoways", 
##     model = "random", random.method = "swar", index = c("Country", 
##         "Year"))
## 
## Unbalanced Panel: n = 161, T = 1-17, N = 2076
## 
## Effects:
##                     var   std.dev share
## idiosyncratic  3785.631    61.527 0.101
## individual    33612.758   183.338 0.898
```

```
## time             22.014     4.692 0.001
## theta:
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## id    0.6818421 0.9188742 0.9188742 0.9121699 0.9188742 0.9188742
## time  0.2106073 0.2218063 0.2377456 0.2365045 0.2428450 0.2813444
## total 0.2103002 0.2216330 0.2374240 0.2360903 0.2425130 0.2810956
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -416.54  -77.74  -33.25    6.07    2.76 2169.25
##
## Coefficients:
##                            Estimate  Std. Error  z-value  Pr(>|z|)
## (Intercept)               7.5705e+02  1.0510e+00   720.32 < 2.2e-16 ***
## SafeWaterAccess          -8.9559e+00  8.1320e-03 -1101.32 < 2.2e-16 ***
## HealthExpenditure        -5.2444e+00  2.4669e-02  -212.59 < 2.2e-16 ***
## PregnantWomenWithAnemia   7.0088e+00  1.1942e-02   586.91 < 2.2e-16 ***
## IncomePerCapita           1.9027e-03  1.3284e-05   143.23 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    249470000
## Residual Sum of Squares: 97131000
## R-Squared:      0.64563
## Adj. R-Squared: 0.64495
## Chisq: 3796030 on 4 DF, p-value: < 2.22e-16
```

The coefficients seems to have the expected signs except for IncomePerCapita, and are all statistically significant at the 0.1%.

We create a table to compare the “FE two-way” and the “RE two-way” models, including also the Pooled OLS model:

```
screenreg(list(
pooled_OLS,
twoway_effects,
RE_twoway),

custom.model.names = c("Pooled OLS",
"Two-way FE",
"Two-way RE" ))

##
## ==============================================================
##                            Pooled OLS   Two-way FE   Two-way RE
## --------------------------------------------------------------
## (Intercept)                 595.34 ***                757.05 ***
##                             (50.49)                    (1.05)
## SafeWaterAccess             -11.34 ***    -7.98 ***    -8.96 ***
##                              (0.38)       (0.62)       (0.01)
```

```
## HealthExpenditure          29.46 ***    -6.83 ***    -5.24 ***
##                            (2.18)       (1.61)       (0.02)
## PregnantWomenWithAnemia    11.64 ***     5.51 ***     7.01 ***
##                            (0.55)       (0.92)       (0.01)
## IncomePerCapita            -0.00         0.00 ***     0.00 ***
##                            (0.00)       (0.00)       (0.00)
## ---------------------------------------------------------------
## R^2                         0.69         0.16         0.65
## Adj. R^2                    0.69         0.09         0.64
## Num. obs.                 2076         2076         2076
## s_idios                                              61.53
## s_id                                                183.34
## s_time                                                4.69
## ===============================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

3.4 Lagged Dependent Variables (LDV) and Dynamic Models

Another way to address auto-correlation (AR) is by modelling the time dependence directly. We can think of a dynamic model as one that takes into account whether changes in the predictor variables have an immediate effect on our dependent variable or whether the effects are distributed over time. We do so by introducing a lagged value of the dependent variable in the right hand side of the equation hence including a AR(1) process among the explanatory variables.

```
ldv_model = plm(MaternalMortality ~ lag(MaternalMortality) + SafeWaterAccess +
HealthExpenditure + PregnantWomenWithAnemia + IncomePerCapita,

data = wdi,
index = c("Country", "Year"),
model = "within",
effect = "twoways" )

summary(ldv_model)

## Twoways effects Within Model
##
## Call:
## plm(formula = MaternalMortality ~ lag(MaternalMortality) + SafeWaterAccess +
##     HealthExpenditure + PregnantWomenWithAnemia + IncomePerCapita,
##     data = wdi, effect = "twoways", model = "within", index = c("Country",
##         "Year"))
##
## Unbalanced Panel: n = 161, T = 1-17, N = 2076
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -8.0916e+01 -1.7451e+00  3.1817e-12  2.0932e+00  8.9342e+01
##
## Coefficients:
```

```
##                            Estimate  Std. Error  t-value  Pr(>|t|)
## lag(MaternalMortality)   0.97518878  0.00357320 272.9175 < 2.2e-16 ***
## SafeWaterAccess         -0.18781989  0.10156134  -1.8493 0.0645667 .
## HealthExpenditure       -0.37611749  0.25450797  -1.4778 0.1396217
## PregnantWomenWithAnemia -0.54779915  0.14731519  -3.7186 0.0002062 ***
## IncomePerCapita          0.00048670  0.00018528   2.6269 0.0086865 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    8589400
## Residual Sum of Squares: 177890
## R-Squared:      0.97929
## Adj. R-Squared: 0.97731
## F-statistic: 17911.3 on 5 and 1894 DF, p-value: < 2.22e-16
```

By includng the lagged value of the dependent variable, the significance of the variables drops, and the sign of some variables come out counterintuitive: Pregnant Women With Anemia has now negative signs. Moreover, Health Expenditure is statistically insignificant, and Safe Access To Water is statistically significant only at a 10% level.

We now create a table to compare all models constructed so far:

```
screenreg(list(
pooled_OLS,
twoway_effects,
RE_twoway,
ldv_model),

custom.model.names = c("Pooled OLS",
"Twoway FE",
"Twoway RE",
"LDV Model"))
```

```
##
## =============================================================================
##                           Pooled OLS    Twoway FE    Twoway RE    LDV Model
## -----------------------------------------------------------------------------
## (Intercept)                595.34 ***                757.05 ***
##                            (50.49)                    (1.05)
## SafeWaterAccess            -11.34 ***    -7.98 ***    -8.96 ***    -0.19
##                             (0.38)       (0.62)       (0.01)       (0.10)
## HealthExpenditure           29.46 ***    -6.83 ***    -5.24 ***    -0.38
##                             (2.18)       (1.61)       (0.02)       (0.25)
## PregnantWomenWithAnemia     11.64 ***     5.51 ***     7.01 ***    -0.55 ***
##                             (0.55)       (0.92)       (0.01)       (0.15)
## IncomePerCapita             -0.00         0.00 ***     0.00 ***     0.00 **
##                             (0.00)       (0.00)       (0.00)       (0.00)
## lag(MaternalMortality)                                              0.98 ***
##                                                                    (0.00)
## -----------------------------------------------------------------------------
```

```
## R^2                            0.69         0.16         0.65         0.98
## Adj. R^2                       0.69         0.09         0.64         0.98
## Num. obs.                   2076         2076         2076         2076
## s_idios                                                  61.53
## s_id                                                    183.34
## s_time                                                    4.69
## ==============================================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

By comparing all the models run so far, we can conclude that more work needs to be done on the regression models as the results are not convincing and ambiguous: probably more variables need to be included, we need to control for endogeneity problems, and also introduce Instrumental Variables. We will now conduct some tests before explore the data further.

4. TESTS

Poolability Test

The poolability test is a F test for individual fixed effects.

Null hypothesis: Pooled OLS better than FE
Alt hypothesis: Significant individual FE

```
pFtest(fixed_effects, pooled_OLS)

##
##  F test for individual effects
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## F = 117.11, df1 = 160, df2 = 1911, p-value < 2.2e-16
## alternative hypothesis: significant effects

pFtest(twoway_effects, pooled_OLS)

##
##  F test for twoways effects
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## F = 106.11, df1 = 176, df2 = 1895, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

The p-values are low so we reject the null hypothesis, and we use the FE model instead of Pooled OLS.

We need to check whether there are indeed any country fixed effects to begin with, so we run the plmtest() function which can test for the presence of individual or time effects.

Ho = no significant fixed effects

```
plmtest(fixed_effects, effect="individual")

##
##  Lagrange Multiplier Test - (Honda) for unbalanced panels
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## normal = 107.12, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

The p-value suggests that we can reject the null hypothesis and that there are indeed country fixed effects present in our model.

Let's run the same test on the time_effects model to see if there are time FE in our model. Ho = no time fixed effects

```
plmtest(time_effects, effect="time")

##
##  Lagrange Multiplier Test - time effects (Honda) for unbalanced panels
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## normal = -2.1126, p-value = 0.9827
## alternative hypothesis: significant effects
```

The p-value is very high and thus we cannot reject the null hypothesis of no time fixed effects.

We also test for FE in the twoway model:

```
plmtest(twoway_effects, effect="twoways")

##
##  Lagrange Multiplier Test - two-ways effects (Honda) for unbalanced
##  panels
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## normal = 74.253, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

The low p-value suggests that we need to control for both individual and time effects in the "within" FE model.

The conclusions of the second round of tests is that firstly, we should control for individual effects; secondly, time effects by themselves are not significant; lastly, the last test suggests that we should control for both time and individual effects at the same time.

Hausman Test

Let us compare the FE and RE models. The Hausman test checks for the correlation between the regressors and the (individual or time) unobserved effects. Note that the Hausman test by itself does not provide a definitive answer to whether we should run the FE or the RE models.

Ho: Errors are not correlated with the regressors No significant difference between FE and RE RE is more efficient and is preferred

Ha: Errors are correlated with the regressors FE is preferred

```
phtest(twoway_effects, RE_twoway)

##
##  Hausman Test
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## chisq = 9.1067, df = 4, p-value = 0.05849
## alternative hypothesis: one model is inconsistent
```

The p-value is not low so we can't reject Ho, and conclude that RE is better than FE. But if we include the lagged dependent variable (LDV) then the FE model is better than the RE model:

```
phtest(ldv_model, RE_twoway)

##
##  Hausman Test
##
## data:  MaternalMortality ~ lag(MaternalMortality) + SafeWaterAccess +  ...
## chisq = 14571, df = 4, p-value < 2.2e-16
## alternative hypothesis: one model is inconsistent
```

This could be related to the fact that by including the lagged dependent variable, a model with higher R squared is computed.

Serial Correlation Test

For time series data we need to address the potential for serial correlation (AR) in the error term. We will test for serial correlation with the Breusch-Godfrey test using the function "pbgtest()". Ho = no AR in the (idisyncratic) error terms

```
pbgtest(twoway_effects)

##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## chisq = 1493, df = 1, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

```
pbgtest(RE_twoway)

##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +  ...
## chisq = 1705.8, df = 1, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

```
pbgtest(ldv_model)

##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  MaternalMortality ~ lag(MaternalMortality) + SafeWaterAccess +  ...
## chisq = 804.95, df = 1, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

The p-values are low, so we rehect the null hypothesis of no serial correlation and conclude thwat the idyosincratic errors include an AutoRegressive component.

We can correct for serial correlation using "coeftest()" similarly to how we correct for heteroskedastic errors. We'll use the "vcovHC()" function for obtaining a heteroskedasticity-consistent covariance matrix, but since we're interested in correcting for autocorrelation (AR) as well, we will specify method = "arellano" which corrects for both heteroskedasticity and autocorrelation (HAC).

```
#HAC = heteroskedasticity and autocorrelation consistent estimators and standard
errors:
twoway_effects_hac = coeftest(twoway_effects,
                              vcov = vcovHC(twoway_effects,
                                            method = "arellano",
                                            type   = "HC3"))
```

Now create a consolidated table to compare the estimates:

```
screenreg(list(
  twoway_effects,
  twoway_effects_hac),

  custom.model.names = c("Twoway Fixed Effects",
                         "Twoway Fixed Effects (HAC)"))

##
## =========================================================================
##                             Twoway Fixed Effects  Twoway Fixed Effects (HAC)
## -------------------------------------------------------------------------
## SafeWaterAccess               -7.98 ***             -7.98 ***
##                               (0.62)                (2.06)
## HealthExpenditure             -6.83 ***             -6.83
```

```
##                              (1.61)              (7.65)
## PregnantWomenWithAnemia       5.51 ***            5.51
##                              (0.92)              (3.12)
## IncomePerCapita               0.00 ***            0.00 *
##                              (0.00)              (0.00)
## -------------------------------------------------------------------------
## R^2                           0.16
## Adj. R^2                      0.09
## Num. obs.                   2076
## =========================================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Variables that are initially significant might turn out to be insignificant once we correct for AR and HC using the HAC estimator. Notice that the coefficient estimates do not change but the standard deviations and p-values do change once we introduce the HAC covariance matrix to get the robust standard deviations. In this case HealthExpenditure and PregnantWomenWithAnemia are not statistically significant anymore, and IncomePerCapita is statistically significant only at the 5% significance level. Adjusted R squared are extremely low too, while the standard deviations have increased significantly.

Cross Sectional Dependence (XSD) Test

When we have panel data with many countries, there might be a factor that is affecting all the countries at the same time, but we might not have a variable to control for such factor. To test for cross sectional dependence and capture the effect of such missing variable, we use the Pesaran cross sectional dependence test: pcdtest(). Ho = no XSD

```
pcdtest(twoway_effects)

##
##  Pesaran CD test for cross-sectional dependence in panels
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +     IncomePerCapita
## z = 30.574, p-value < 2.2e-16
## alternative hypothesis: cross-sectional dependence

pcdtest(RE_twoway)

##
##  Pesaran CD test for cross-sectional dependence in panels
##
## data:  MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia +     IncomePerCapita
## z = 45.285, p-value < 2.2e-16
## alternative hypothesis: cross-sectional dependence

pcdtest(ldv_model)

##
##  Pesaran CD test for cross-sectional dependence in panels
```

```
##
## data:  MaternalMortality ~ lag(MaternalMortality) + SafeWaterAccess +
HealthExpenditure + PregnantWomenWithAnemia + IncomePerCapita
## z = 47.974, p-value < 2.2e-16
## alternative hypothesis: cross-sectional dependence
```

The p-values are low, so we do reject the null hypothesis of no XSD and conclude that there is a factor affecting all countries at the same time, but we do not have a specific variable that control for such effects.

If there is XSD the we need to correct for it. We can use the Beck and Katz (1995) method: "Panel Corrected Standard Errors (PCSE)". We can obtain Panel Corrected Standard Errors (PCSE) by first obtaining a robust variance-covariance matrix for panel models with the Beck and Katz (1995) method using the vcovBK() and passing it to the familiar coeftest() function:

```
twoway_effects_pcse = coeftest(twoway_effects,
                              vcov = vcovBK(twoway_effects,
                                            type = "HC3",
                                            cluster = "group"))

twoway_effects_pcse

##
## t test of coefficients:
##
##                            Estimate Std. Error t value  Pr(>|t|)
## SafeWaterAccess          -7.9821605  2.2462144 -3.5536 0.0003893 ***
## HealthExpenditure        -6.8293578  4.5809657 -1.4908 0.1361773
## PregnantWomenWithAnemia   5.5100821  3.1068766  1.7735 0.0763045 .
## IncomePerCapita           0.0042492  0.0033367  1.2734 0.2030157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results from PCSE are sensitive to the ratio between the number of time periods in the dataset (T) and the total number of observations (N). The coefficients have now lost significance except for SafeWaterAccess. In large datasets, the T/N ratio is small. To deal with the limitations of the Beck and Katz's PCSE method, we now implement the Discroll and Kraay (1998) SCC method. This allows us to to obtain heteroskedasticity and autocorrelation consistent (HAC) errors that are also robust to cross-sectional dependence. We can get SCC corrected covariance matrix using the vcovSCC() function:

```
twoway_effects_scc = coeftest(twoway_effects,
                             vcov = vcovSCC(twoway_effects,
                                            type = "HC3",
                                            cluster = "group"))

twoway_effects_scc

##
## t test of coefficients:
```

```
##
##                           Estimate Std. Error t value  Pr(>|t|)
## SafeWaterAccess         -7.9821605  1.9527163 -4.0877 4.538e-05 ***
## HealthExpenditure       -6.8293578  7.7192746 -0.8847   0.37642
## PregnantWomenWithAnemia  5.5100821  3.0548490  1.8037   0.07143 .
## IncomePerCapita          0.0042492  0.0019654  2.1620   0.03074 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Even by using the Discroll and Kraay method, the coefficients have still low significance. IncomePerCapita has gained significance at the 5%.

We can create a consolidated table with a comparison of all the different models computed so far.

```
huxreg.matrix  =  huxreg(

  "Pooled OLS" = pooled_OLS,
  "RE two-way effects" = RE_twoway,
  "Lagged dep. variable" = ldv_model,
  "FE two-way effects" = twoway_effects,
  "Arellano HAC FE" = twoway_effects_hac,
  "Beck-Katz two-way FE" = twoway_effects_pcse,
  "Driscoll-Kraay two-way FE" = twoway_effects_scc,
  "FE indiv. effects" = fixed_effects,
  "FE time effects" = time_effects,
  stars = c(`*` = 0.1, `**` = 0.05, `***` = 0.01))

## Original model not retained as part of coeftest object. For additional model
summary information (r.squared, df, etc.), consider passing `glance.coeftest()` an
object where the underlying model has been saved, i.e.`lmtest::coeftest(..., save =
TRUE)`.
## This message is displayed once per session.

huxreg.matrix
```

	Pooled OLS	RE two-way effects	Lagged dep. variable	FE two-way effects	Arellano HAC FE	Beck-Katz two-way FE	Driscoll-Kraay two-way FE	FE indiv. effects	FE time effects
(Intercept)	595.338 ***	757.050 ***							
	(50.486)	(1.051)							
SafeWaterAccess	-11.344 ***	-8.956 ***	-0.188 *	-7.982 ***	-7.982 ***	-7.982 ***	-7.982 ***	-8.396 ***	-11.353 ***
	(0.380)	(0.008)	(0.102)	(0.619)	(2.063)	(2.246)	(1.953)	(0.550)	(0.381)
HealthExpenditure	29.464 ***	-5.244 ***	-0.376	-6.829 ***	-6.829	-6.829	-6.829	-7.139 ***	29.439 ***
	(2.176)	(0.025)	(0.255)	(1.609)	(7.650)	(4.581)	(7.719)	(1.534)	(2.202)
PregnantWomenWithAnemia	11.644 ***	7.009 ***	-0.548 ***	5.510 ***	5.510 *	5.510 *	5.510 *	6.086 ***	11.662 ***
	(0.550)	(0.012)	(0.147)	(0.925)	(3.120)	(3.107)	(3.055)	(0.788)	(0.553)
IncomePerCapita	-0.000	0.002 ***	0.000 ***	0.004 ***	0.004 **	0.004	0.004 **	0.003 ***	-0.000
	(0.000)	(0.000)	(0.000)	(0.001)	(0.002)	(0.003)	(0.002)	(0.001)	(0.000)
lag(MaternalMortality)			0.975 ***						
			(0.004)						
N	2076	2076	2076	2076	2076	2076	2076	2076	2076
R2	0.688	0.646	0.979	0.165				0.317	0.686
logLik					-11403.074	-11403.074	-11403.074		
AIC					22816.150	22816.150	22816.150		

*** p < 0.01; ** p < 0.05; * p < 0.1.

Column names: names, Pooled OLS, RE two-way effects, Lagged dep. variable, FE two-way effects, Arellano HAC FE, Beck-Katz two-way FE, Driscoll-Kraay two-way FE, FE indiv. effects, FE time effects

5. INSTRUMENTAL VARIABLES, ENDOGENEITY, AND GMM ESTIMATION

5.1 Baseline non-IV model

The baseline model chosen for this project will be the Fixed effects (FE) model with two-way effects (individual and time effects) and with no instruments previously computed.

```
twoway_effects = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita,

                     data = wdi,
                     index = c("Country", "Year"),
                     model = "within",
                     effect = "twoways")

summary(twoway_effects)

## Twoways effects Within Model
##
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita, data = wdi, effect = "twoways",
##     model = "within", index = c("Country", "Year"))
```

```
##
## Unbalanced Panel: n = 161, T = 1-17, N = 2076
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -567.582  -14.088    0.000   12.432  539.814
##
## Coefficients:
##                             Estimate Std. Error  t-value  Pr(>|t|)
## SafeWaterAccess           -7.9821605  0.6187568 -12.9003 < 2.2e-16 ***
## HealthExpenditure         -6.8293578  1.6087862  -4.2450 2.291e-05 ***
## PregnantWomenWithAnemia    5.5100821  0.9245709   5.9596 3.008e-09 ***
## IncomePerCapita            0.0042492  0.0011730   3.6225 0.0002994 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    8589400
## Residual Sum of Squares: 7173800
## R-Squared:      0.16481
## Adj. R-Squared: 0.085483
## F-statistic: 93.4894 on 4 and 1895 DF, p-value: < 2.22e-16
```

We also compute the heteroskedasticity-robust standard errors.

```
coeftest(twoway_effects, vcov = vcovHC, type = "HC1")

##
## t test of coefficients:
##
##                             Estimate Std. Error t value Pr(>|t|)
## SafeWaterAccess           -7.9821605  2.0522964 -3.8894 0.000104 ***
## HealthExpenditure         -6.8293578  7.6138500 -0.8970 0.369851
## PregnantWomenWithAnemia    5.5100821  3.1095656  1.7720 0.076559 .
## IncomePerCapita            0.0042492  0.0020134  2.1104 0.034952 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(twoway_effects, vcov = vcovHC, type = "HC2")

##
## t test of coefficients:
##
##                             Estimate Std. Error t value  Pr(>|t|)
## SafeWaterAccess           -7.9821605  2.0567779 -3.8809 0.0001076 ***
## HealthExpenditure         -6.8293578  7.6282833 -0.8953 0.3707575
## PregnantWomenWithAnemia    5.5100821  3.1131044  1.7700 0.0768939 .
## IncomePerCapita            0.0042492  0.0020199  2.1037 0.0355396 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(twoway_effects, vcov = vcovHC, type = "HC3")
```

```
##
## t test of coefficients:
##
##                            Estimate Std. Error t value  Pr(>|t|)
## SafeWaterAccess          -7.9821605  2.0632821 -3.8687 0.0001131 ***
## HealthExpenditure        -6.8293578  7.6501470 -0.8927 0.3721262
## PregnantWomenWithAnemia   5.5100821  3.1196661  1.7662 0.0775164 .
## IncomePerCapita           0.0042492  0.0020284  2.0948 0.0363200 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(twoway_effects, vcov = vcovHC, type = "HC4")
```

```
##
## t test of coefficients:
##
##                            Estimate Std. Error t value  Pr(>|t|)
## SafeWaterAccess          -7.9821605  2.0726356 -3.8512 0.0001214 ***
## HealthExpenditure        -6.8293578  7.6754530 -0.8898 0.3737044
## PregnantWomenWithAnemia   5.5100821  3.1249081  1.7633 0.0780147 .
## IncomePerCapita           0.0042492  0.0020423  2.0806 0.0376037 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.2 Two-stage least squares (2SLS) with external IV

Suppose we have reasons to believe that one of the regressors is not strictly exogenous. In this case we would need to find an instrumental variable (IV) for it. Endogeneity makes OLS estimates biased and inconsistent, and therefore must be remedied. The instrument must be correlated with the regressor but not with the error (it must be exogenous). Also, the instrument must affect the dependent variable only indirectly via the regressor, and not directly. We must have at least the same number of instruments as endogenous regressors, or maybe more IV than endogenous regressors, but not less instruments than endogenous regressors. The 2SLS estimator is consistent and normally distributed when the sample size is large.

We expect correlation between the regressor and the instrument to be high, the correlation between the dependent variable and the instrument should be low, and the correlation between the instrument and the residuals of the non-IV baseline model should be close to zero

Suppose “HealthExpenditure” is endogenous to “MaternalMortality”. We now need to find an IV for “HealthExpenditure”. We will choose an external IV that is in the dataset but not already in the model. The instrument must be correlated with “HealthExpenditure” but uncorrelated with the error term, hence is must be an exogenous variable. As an IV for “HealthExpenditure” we use different variables at the time and compare the results.

Run the panel IV regression:

```
Two_Stage_IV = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita
```

```
                   | . - HealthExpenditure + NursesMidwives + GDPPerCapita,

                   data = wdi,
                   index = c("Country", "Year"),
                   model = "within",
                   effect = "twoways",
                   inst.method = "bvk" )

summary(Two_Stage_IV)

## Twoways effects Within Model
## Instrumental variable estimation
##
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita | . - HealthExpenditure +
##     NursesMidwives + GDPPerCapita, data = wdi, effect = "twoways",
##     model = "within", inst.method = "bvk", index = c("Country",
##         "Year"))
##
## Unbalanced Panel: n = 142, T = 1-9, N = 422
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -151.23968   -6.57974    0.25948    8.15175  228.57612
##
## Coefficients:
##                            Estimate  Std. Error z-value Pr(>|z|)
## SafeWaterAccess         -6.46314651  3.51927661 -1.8365 0.066284 .
## HealthExpenditure       -7.70532313 39.41837804 -0.1955 0.845021
## PregnantWomenWithAnemia  5.55182352  2.02471408  2.7420 0.006106 **
## IncomePerCapita          0.00029378  0.00162140  0.1812 0.856221
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    337830
## Residual Sum of Squares: 279410
## R-Squared:      0.17412
## Adj. R-Squared: -0.32708
## Chisq: 58.5151 on 4 DF, p-value: 5.9488e-12

Two_Stage_IV_1 = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita
                   | . - HealthExpenditure + OilRents + GDPPerCapita,

                   data = wdi,
                   index = c("Country", "Year"),
                   model = "within",
                   effect = "twoways",
                   inst.method = "bvk" )
```

```
summary(Two_Stage_IV_1)

## Twoways effects Within Model
## Instrumental variable estimation
##
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita | . - HealthExpenditure +
##     OilRents + GDPPerCapita, data = wdi, effect = "twoways",
##     model = "within", inst.method = "bvk", index = c("Country",
##         "Year"))
##
## Unbalanced Panel: n = 161, T = 1-17, N = 2068
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -567.632  -13.952    0.000   12.215  540.562
##
## Coefficients:
##                           Estimate Std. Error  z-value  Pr(>|z|)
## SafeWaterAccess         -8.0114345  0.7973446 -10.0476 < 2.2e-16 ***
## HealthExpenditure       -7.3101208 18.8479360  -0.3878 0.6981291
## PregnantWomenWithAnemia  5.5368377  0.9407855   5.8853 3.972e-09 ***
## IncomePerCapita          0.0041215  0.0012290   3.3534 0.0007982 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    8546800
## Residual Sum of Squares: 7130200
## R-Squared:      0.16575
## Adj. R-Squared: 0.086174
## Chisq: 356.047 on 4 DF, p-value: < 2.22e-16

Two_Stage_IV_2 = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita
                   | . - HealthExpenditure + OilRents,

                   data = wdi,
                   index = c("Country", "Year"),
                   model = "within",
                   effect = "twoways",
                   inst.method = "bvk" )

summary(Two_Stage_IV_2)

## Twoways effects Within Model
## Instrumental variable estimation
##
## Call:
```

```
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita | . - HealthExpenditure +
##     OilRents, data = wdi, effect = "twoways", model = "within",
##     inst.method = "bvk", index = c("Country", "Year"))
##
## Unbalanced Panel: n = 161, T = 1-17, N = 2068
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -662.25324  -42.50717   -0.25348   38.47951  587.33428
##
## Coefficients:
##                            Estimate  Std. Error z-value  Pr(>|z|)
## SafeWaterAccess         -10.2053061   1.5092363 -6.7619 1.362e-11 ***
## HealthExpenditure       -89.1261201  43.7172572 -2.0387  0.041481 *
## PregnantWomenWithAnemia   4.7632663   1.4826284  3.2127  0.001315 **
## IncomePerCapita           0.0057238   0.0019995  2.8626  0.004202 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    8546800
## Residual Sum of Squares: 16940000
## R-Squared:      0.046181
## Adj. R-Squared: -0.044804
## Chisq: 153.956 on 4 DF, p-value: < 2.22e-16
```

```
Two_Stage_IV_3 = plm(MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
PregnantWomenWithAnemia + IncomePerCapita
                   | . - HealthExpenditure + OilRents + GDPPerCapita +
NursesMidwives,

                   data = wdi,
                   index = c("Country", "Year"),
                   model = "within",
                   effect = "twoways",
                   inst.method = "bvk" )

summary(Two_Stage_IV_3)
```

```
## Twoways effects Within Model
## Instrumental variable estimation
##
## Call:
## plm(formula = MaternalMortality ~ SafeWaterAccess + HealthExpenditure +
##     PregnantWomenWithAnemia + IncomePerCapita | . - HealthExpenditure +
##     OilRents + GDPPerCapita + NursesMidwives, data = wdi, effect = "twoways",
##     model = "within", inst.method = "bvk", index = c("Country",
##         "Year"))
##
## Unbalanced Panel: n = 142, T = 1-9, N = 422
```

```
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -146.02370   -6.20834    0.50287    7.81695   216.24489
##
## Coefficients:
##                             Estimate  Std. Error z-value Pr(>|z|)
## SafeWaterAccess          -5.44077045  2.79083742 -1.9495 0.051234 .
## HealthExpenditure         4.74621631 29.52685182  0.1607 0.872296
## PregnantWomenWithAnemia   5.86430578  1.91622025  3.0604 0.002211 **
## IncomePerCapita           0.00018957  0.00160709  0.1180 0.906099
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    337830
## Residual Sum of Squares: 279580
## R-Squared:      0.17292
## Adj. R-Squared: -0.32901
## Chisq: 58.4664 on 4 DF, p-value: 6.0907e-12

huxreg.matrix  =  huxreg(

  "Two Stage IV " = Two_Stage_IV,
  "Two Stage IV 1" = Two_Stage_IV_1,
  "Two Stage IV 2" = Two_Stage_IV_2,
  "Two Stage IV 3" = Two_Stage_IV_3,
  "Twoway effects" = twoway_effects,

  stars = c(`*` = 0.1, `**` = 0.05, `***` = 0.01))

huxreg.matrix
```

	Two Stage IV	Two Stage IV 1	Two Stage IV 2	Two Stage IV 3	Twoway effects
SafeWaterAccess	-6.463 *	-8.011 ***	-10.205 ***	-5.441 *	-7.982 ***
	(3.519)	(0.797)	(1.509)	(2.791)	(0.619)
HealthExpenditure	-7.705	-7.310	-89.126 **	4.746	-6.829 ***
	(39.418)	(18.848)	(43.717)	(29.527)	(1.609)
PregnantWomenWithAnemia	5.552 ***	5.537 ***	4.763 ***	5.864 ***	5.510 ***
	(2.025)	(0.941)	(1.483)	(1.916)	(0.925)
IncomePerCapita	0.000	0.004 ***	0.006 ***	0.000	0.004 ***
	(0.002)	(0.001)	(0.002)	(0.002)	(0.001)
N	422	2068	2068	422	2076
R2	0.174	0.166	0.046	0.173	0.165

*** p < 0.01; ** p < 0.05; * p < 0.1.

The standard errors are much larger in the IV model. By running a combination of different Instrumental variables, the most convincing one seems to be IV 1, in whch only OilRents is used. The standard errors of most of the variables is pretty close to the two way effects model with no instrumental variables, although the standard error of HealthExpenditure has increase significantly. It could be argued that none of the variables available are actually instrumental variables, so further research need to be done. It could be suggested that variables that measure the average distance from the hospital of the average household, number of hospital per number of population, could be used for improving the model.

Below we can find heteroskedasticity-robust standard errors:

```
coeftest(Two_Stage_IV, vcov = vcovHC, type = "HC1")

##
## t test of coefficients:
##
##                            Estimate  Std. Error t value Pr(>|t|)
## SafeWaterAccess         -6.46314651  3.51230140 -1.8401  0.06688 .
## HealthExpenditure       -7.70532313 27.13093109 -0.2840  0.77663
## PregnantWomenWithAnemia  5.55182352  3.04310370  1.8244  0.06923 .
## IncomePerCapita          0.00029378  0.00115146  0.2551  0.79882
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(Two_Stage_IV, vcov = vcovHC, type = "HC2")

##
## t test of coefficients:
##
##                            Estimate  Std. Error t value Pr(>|t|)
## SafeWaterAccess         -6.46314651  3.67935231 -1.7566  0.08015 .
## HealthExpenditure       -7.70532313 29.36584170 -0.2624  0.79323
## PregnantWomenWithAnemia  5.55182352  3.07453062  1.8057  0.07211 .
## IncomePerCapita          0.00029378  0.00123808  0.2373  0.81262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(Two_Stage_IV, vcov = vcovHC, type = "HC3")

##
## t test of coefficients:
##
##                            Estimate  Std. Error t value Pr(>|t|)
## SafeWaterAccess         -6.46314651  3.88432585 -1.6639  0.09733 .
## HealthExpenditure       -7.70532313 32.02882362 -0.2406  0.81007
## PregnantWomenWithAnemia  5.55182352  3.12419845  1.7770  0.07672 .
## IncomePerCapita          0.00029378  0.00134216  0.2189  0.82691
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(Two_Stage_IV, vcov = vcovHC, type = "HC4")

##
## t test of coefficients:
##
##                            Estimate  Std. Error t value Pr(>|t|)
## SafeWaterAccess         -6.46314651  4.35396914 -1.4844   0.1389
```

```
## HealthExpenditure       -7.70532313 38.34641364 -0.2009   0.8409
## PregnantWomenWithAnemia  5.55182352  3.20489069  1.7323   0.0844 .
## IncomePerCapita          0.00029378  0.00158899  0.1849   0.8535
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.3 GMM Estimation with internal IVs

Now we will estimate the GMM model with lagged regressors as internal instruments. The generalized method of moments (GMM) is mainly used in panel data econometrics to estimate dynamic models with a lagged endogenous variable. In a GMM estimation, there are normal instruments and GMM instruments. All the variables of the model which are not used as GMM instruments are used as normal instruments with the same lag structure as the one specified in the model.

```
GMM = pgmm(         MaternalMortality ~

                      lag(MaternalMortality,       1:1) +
                      lag(SafeWaterAccess,         0:2) +
                      lag(HealthExpenditure,       0:2) +
                      lag(PregnantWomenWithAnemia, 0:2) +
                      lag(IncomePerCapita,         0:2)
                    | lag(MaternalMortality,       2:6),

                    data = wdi,
                    index = c("Country", "Year"),
                    model = "onestep",
                    effect = "individual"    )

summary(GMM)

## Oneway (individual) effect One-step model Difference GMM
##
## Call:
## pgmm(formula = MaternalMortality ~ lag(MaternalMortality, 1:1) +
##     lag(SafeWaterAccess, 0:2) + lag(HealthExpenditure, 0:2) +
##     lag(PregnantWomenWithAnemia, 0:2) + lag(IncomePerCapita,
##     0:2) | lag(MaternalMortality, 2:6), data = wdi, effect = "individual",
##     model = "onestep", index = c("Country", "Year"))
##
## Balanced Panel: n = 192, T = 56, N = 10752
##
## Number of Observations Used: 1650
## Residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -118.9926    0.0000    0.0000   -0.1755    0.0000  187.9692
##
## Coefficients:
```

```
##                                      Estimate  Std. Error z-value Pr(>|z|)
## lag(MaternalMortality, 1:1)        8.0389e-01  4.1741e-02 19.2590  < 2e-16 ***
## lag(SafeWaterAccess, 0:2)0        -1.7105e+02  8.7708e+01 -1.9502  0.05115 .
## lag(SafeWaterAccess, 0:2)1         5.6745e+01  4.6364e+01  1.2239  0.22099
## lag(SafeWaterAccess, 0:2)2         1.1206e+02  9.8676e+01  1.1357  0.25610
## lag(HealthExpenditure, 0:2)0      -6.2810e-01  7.6406e-01 -0.8221  0.41105
## lag(HealthExpenditure, 0:2)1      -2.2524e+00  1.3496e+00 -1.6689  0.09514 .
## lag(HealthExpenditure, 0:2)2      -1.5289e+00  1.6327e+00 -0.9364  0.34907
## lag(PregnantWomenWithAnemia, 0:2)0  3.1829e+01  3.0447e+01  1.0454  0.29585
## lag(PregnantWomenWithAnemia, 0:2)1  2.2681e+01  5.1436e+01  0.4410  0.65925
## lag(PregnantWomenWithAnemia, 0:2)2 -5.6499e+01  2.9085e+01 -1.9425  0.05207 .
## lag(IncomePerCapita, 0:2)0        -1.0940e-03  5.1526e-04 -2.1232  0.03374 *
## lag(IncomePerCapita, 0:2)1        -9.3662e-04  4.0254e-04 -2.3268  0.01998 *
## lag(IncomePerCapita, 0:2)2        -6.9003e-04  4.1353e-04 -1.6686  0.09519 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sargan test: chisq(258) = 103.895 (p-value = 1)
## Autocorrelation test (1): normal = 0.4657612 (p-value = 0.64139)
## Autocorrelation test (2): normal = 0.2775533 (p-value = 0.78136)
## Wald test for coefficients: chisq(13) = 826.4293 (p-value = < 2.22e-16)
```

By looking at the table, only the first lag of MaternalMortality is sufficiently significant. Results are not convincing, and suggest that different variables should be included in the model. Moreover, the Sargan test suggests that the instruments as a group are exogeneous and valid, as the p-value is greater than 0.25 (p-value = 1).

Computing the GMM using logs and lags of logs:

```
GMM.logs = pgmm(         log(MaternalMortality) ~

                           lag(log(MaternalMortality),       1:1) +
                           lag(log(SafeWaterAccess),         0:2) +
                           lag(log(HealthExpenditure),       0:2) +
                           lag(log(PregnantWomenWithAnemia), 0:2) +
                           lag(IncomePerCapita,              0:2)
                         | lag(log(MaternalMortality),       2:6),

                         data = wdi,
                         index = c("Country", "Year"),
                         model = "onestep",
                         effect = "individual"    )

summary(GMM.logs)

## Oneway (individual) effect One-step model Difference GMM
##
## Call:
## pgmm(formula = log(MaternalMortality) ~ lag(log(MaternalMortality),
##     1:1) + lag(log(SafeWaterAccess), 0:2) + lag(log(HealthExpenditure),
```

```
##     0:2) + lag(log(PregnantWomenWithAnemia), 0:2) + lag(IncomePerCapita,
##     0:2) | lag(log(MaternalMortality), 2:6), data = wdi, effect = "individual",
##     model = "onestep", index = c("Country", "Year"))
##
## Balanced Panel: n = 192, T = 56, N = 10752
##
## Number of Observations Used: 1650
## Residuals:
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.5205433  0.0000000  0.0000000 -0.0001039  0.0000000  0.3480286
##
## Coefficients:
##                                           Estimate  Std. Error z-value
## lag(log(MaternalMortality), 1:1)         4.0859e-01  4.1744e-01  0.9788
## lag(log(SafeWaterAccess), 0:2)0          2.4450e+01  3.0827e+01  0.7931
## lag(log(SafeWaterAccess), 0:2)1          7.5925e+00  2.0360e+01  0.3729
## lag(log(SafeWaterAccess), 0:2)2         -3.1187e+01  2.7099e+01 -1.1509
## lag(log(HealthExpenditure), 0:2)0        8.9176e-02  7.8840e-02  1.1311
## lag(log(HealthExpenditure), 0:2)1       -1.0618e-01  1.3216e-01 -0.8034
## lag(log(HealthExpenditure), 0:2)2       -1.5415e-01  1.3536e-01 -1.1388
## lag(log(PregnantWomenWithAnemia), 0:2)0  5.8392e+00  4.3806e+00  1.3330
## lag(log(PregnantWomenWithAnemia), 0:2)1 -1.1721e+01  8.6713e+00 -1.3516
## lag(log(PregnantWomenWithAnemia), 0:2)2  6.9480e+00  6.3435e+00  1.0953
## lag(IncomePerCapita, 0:2)0              -1.2497e-06  5.0786e-06 -0.2461
## lag(IncomePerCapita, 0:2)1              -1.7887e-06  5.7216e-06 -0.3126
## lag(IncomePerCapita, 0:2)2               2.4550e-07  5.0227e-06  0.0489
##                                         Pr(>|z|)
## lag(log(MaternalMortality), 1:1)          0.3277
## lag(log(SafeWaterAccess), 0:2)0           0.4277
## lag(log(SafeWaterAccess), 0:2)1           0.7092
## lag(log(SafeWaterAccess), 0:2)2           0.2498
## lag(log(HealthExpenditure), 0:2)0         0.2580
## lag(log(HealthExpenditure), 0:2)1         0.4217
## lag(log(HealthExpenditure), 0:2)2         0.2548
## lag(log(PregnantWomenWithAnemia), 0:2)0   0.1825
## lag(log(PregnantWomenWithAnemia), 0:2)1   0.1765
## lag(log(PregnantWomenWithAnemia), 0:2)2   0.2734
## lag(IncomePerCapita, 0:2)0                0.8056
## lag(IncomePerCapita, 0:2)1                0.7546
## lag(IncomePerCapita, 0:2)2                0.9610
##
## Sargan test: chisq(258) = 38.50207 (p-value = 1)
## Autocorrelation test (1): normal = -1.051932 (p-value = 0.29283)
## Autocorrelation test (2): normal = 1.76748 (p-value = 0.077148)
## Wald test for coefficients: chisq(13) = 250.706 (p-value = < 2.22e-16)
```

By computing the logs, none of the regressors are statistically significant, suggesting the the GMM model using logs is not a valid model to be used for analysis, and the elasticities are meaningless in this scenario.

6. COMPARISONS OF THE IV MODELS

We create a final table to compare the main models computed so far:

```
screenreg(list(
  twoway_effects,
  Two_Stage_IV_1,
  GMM,
  GMM.logs
),
custom.model.names = c(
  "FE two-way",
  "Two Stage IV 1",
  "GMM",
  "GMM.logs"
))

##
##
================================================================================
==========
##                                            FE two-way   Two Stage IV 1   GMM
GMM.logs
## -----------------------------------------------------------------------------
-------------
## SafeWaterAccess                              -7.98 ***    -8.01 ***
##                                              (0.62)       (0.80)
## HealthExpenditure                            -6.83 ***    -7.31
##                                              (1.61)      (18.85)
## PregnantWomenWithAnemia                       5.51 ***     5.54 ***
##                                              (0.92)       (0.94)
## IncomePerCapita                               0.00 ***     0.00 ***
##                                              (0.00)       (0.00)
## lag(MaternalMortality, 1:1)                                                 0.80
***
##                                                                             (0.04)
## lag(SafeWaterAccess, 0:2)0                                               -171.05
##                                                                           (87.71)
## lag(SafeWaterAccess, 0:2)1                                                 56.74
##                                                                           (46.36)
## lag(SafeWaterAccess, 0:2)2                                                112.06
##                                                                           (98.68)
## lag(HealthExpenditure, 0:2)0                                               -0.63
##                                                                            (0.76)
## lag(HealthExpenditure, 0:2)1                                               -2.25
##                                                                            (1.35)
## lag(HealthExpenditure, 0:2)2                                               -1.53
##                                                                            (1.63)
## lag(PregnantWomenWithAnemia, 0:2)0                                          31.83
##                                                                           (30.45)
```

```
## lag(PregnantWomenWithAnemia, 0:2)1                                                  22.68
##                                                                                   (51.44)
## lag(PregnantWomenWithAnemia, 0:2)2                                                -56.50
##                                                                                   (29.09)
## lag(IncomePerCapita, 0:2)0                                                         -0.00 *
-0.00
##                                                                                    (0.00)
(0.00)
## lag(IncomePerCapita, 0:2)1                                                         -0.00 *
-0.00
##                                                                                    (0.00)
(0.00)
## lag(IncomePerCapita, 0:2)2                                                         -0.00
0.00
##                                                                                    (0.00)
(0.00)
## lag(log(MaternalMortality), 1:1)
0.41
##
(0.42)
## lag(log(SafeWaterAccess), 0:2)0
24.45
##
(30.83)
## lag(log(SafeWaterAccess), 0:2)1
7.59
##
(20.36)
## lag(log(SafeWaterAccess), 0:2)2
-31.19
##
(27.10)
## lag(log(HealthExpenditure), 0:2)0
0.09
##
(0.08)
## lag(log(HealthExpenditure), 0:2)1
-0.11
##
(0.13)
## lag(log(HealthExpenditure), 0:2)2
-0.15
##
(0.14)
## lag(log(PregnantWomenWithAnemia), 0:2)0
5.84
##
(4.38)
## lag(log(PregnantWomenWithAnemia), 0:2)1
-11.72
```

```
##
(8.67)
## lag(log(PregnantWomenWithAnemia), 0:2)2
6.95
##
(6.34)
## ---------------------------------------------------------------------------------
-------------
## R^2                                          0.16         0.17
## Adj. R^2                                     0.09         0.09
## Num. obs.                                 2076         2068            10752
10752
## n                                                                        192
192
## T                                                                         56
56
## Num. obs. used                                                          1650
1650
## Sargan Test: chisq                                                       103.90
38.50
## Sargan Test: df                                                          258.00
258.00
## Sargan Test: p-value                                                       1.00
1.00
## Wald Test Coefficients: chisq                                            826.43
250.71
## Wald Test Coefficients: df                                                13
13
## Wald Test Coefficients: p-value                                            0.00
0.00
##
=================================================================================
==========
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Among the models computed through this project, the FE Two-Way model s the one that has a higher number of significant coefficient of the independent variables. Nevertheless, none of the models is correctly specified, and we recommend to do further research and include more exhaustive variables in the model that could help to better analyse the causes of MaternalMortality across countries over time.