Projekt 5. Semester

Simon Thomsen

4. Jan. 2020

Contents

Funktioner vi bruger

```
arpdiag <- function(series, lags) {
x <- mat.or.vec(lags, 1)
y <- mat.or.vec(lags, 1)
y <- y + 0.05
for (i in 1:lags) {
x[i] <- Box.test(series, lag = i, type = "Ljung-Box")$p.value
}
plot(x, xlab = "Lags", ylab = "p-value H0: no
Autocorrelation", type = "p",
main = "Ljung-Box Test for
Autocorrelation", ylim = c(0, 1))
axis(1, 1:lags)
abline(0.05, 0, lty = 2, col = "blue")
}

##Hamids function for to extracting the lags of my variables used to find structural breaks

lagged <- function(x, k) {
if (k>0) {
return (c(rep(NA, k), x)[1 : length(x)] );
}
else {
return (c(x[(-k+1) : length(x)], rep(NA, -k)));
}
}
```

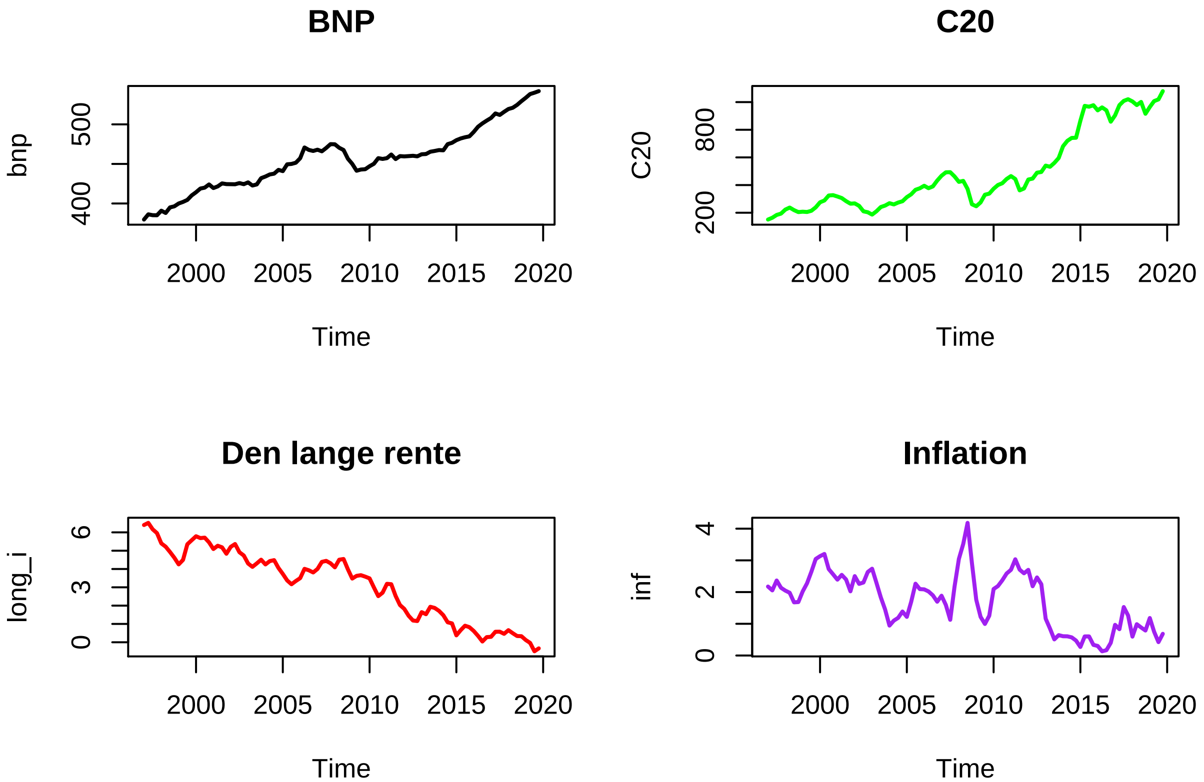
```
library(readxl)
##Fed2 <- read_excel("C:/Users/pjkss/OneDrive/Skrivebord/Projekt 5. semester/Fed2.xlsx")
##Fed2 <- read_excel("C:/Users/Simon ik mig/Desktop/projekt 5. semester/DData/Fed2.xlsx")
##Fed2 <- read_excel("C:/Users/Methling/Dropbox/Uni/5. semester/Projekt/Data/Fed2.xlsx")
Fed2 <- read_excel("/Users/Kristoffer/Desktop/Universitet/5. semester/Econometrics/WD/Fed2.xlsx")
```

We take the variables.

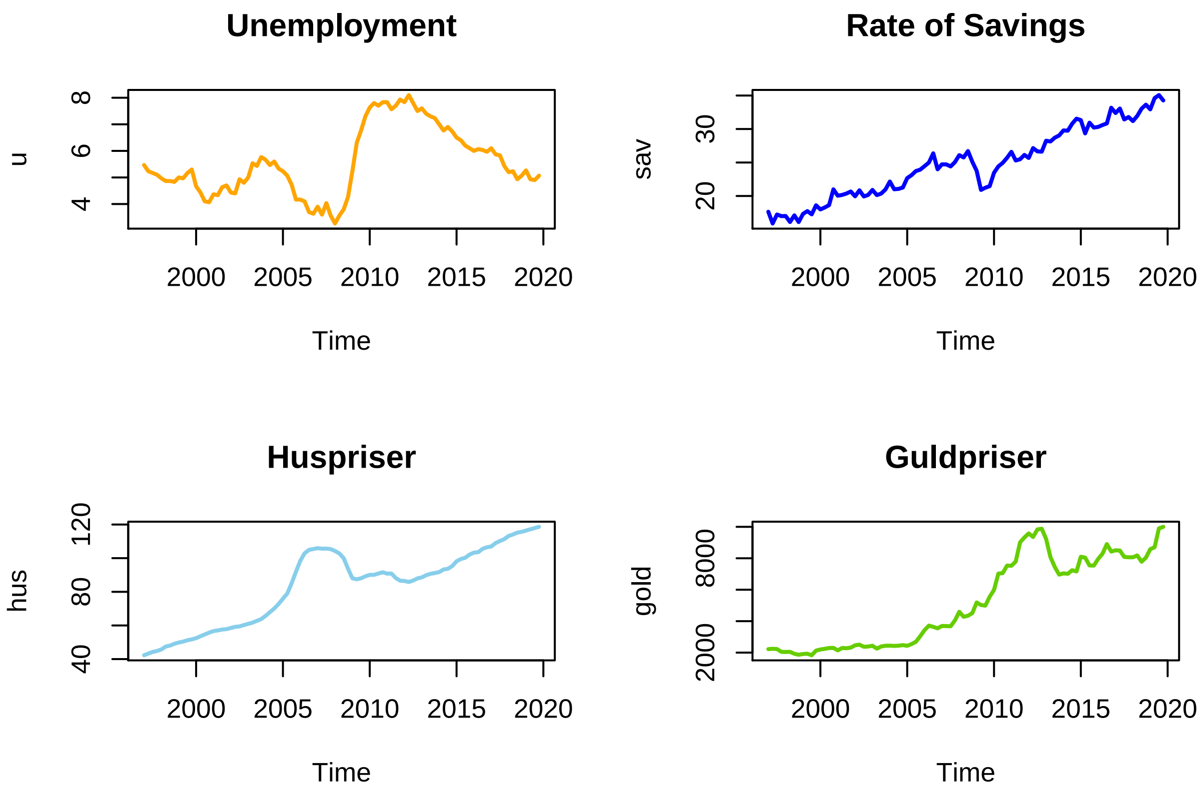
```
bnp= ts(Fed2$q_bnp, start = c(1997,1), end = c(2019, 4), frequency = 4)

C20= ts(Fed2$q_C20, start = c(1997,1), end = c(2019, 4), frequency = 4)

long_i= ts(Fed2$q_long_i, start = c(1997,1), end = c(2019, 4), frequency = 4)

inf= ts(Fed2$q_inf, start = c(1997,1), end = c(2019, 4), frequency = 4)

hus= ts(Fed2$q_house, start = c(1997,1), end = c(2019, 4), frequency = 4)

u = ts(Fed2$q_u, start = c(1997,1), end = c(2019, 4), frequency = 4)

sav = ts(Fed2$q_sav1, start = c(1997,1), end = c(2019, 4), frequency = 4)

short_i= ts(Fed2$q_short_i, start = c(1997,1), end = c(2019, 4), frequency = 4)

gold = ts(Fed2$q_gold, start = c(1997,1), end = c(2019, 4), frequency = 4)
```

Vi plotter data.

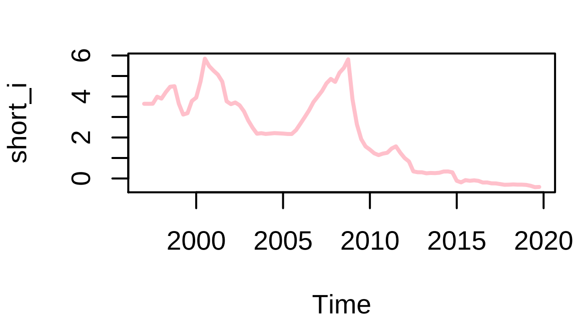
```
par(mfrow=c(2,2))
plot(bnp, col = "black", lwd = "2", main = "BNP" )
plot(C20, col = "green", lwd = "2", main = "C20" )
plot(long_i, col = "red", lwd = "2", main = "Den lange rente")
plot(inf, col = "purple", lwd = "2", main = "Inflation")
```




```
plot(u, col = "orange", lwd = "2", main = "Unemployment")
plot(sav, col = "blue", lwd = "2", main = "Rate of Savings")
plot(hus, col = "skyblue", lwd = "2", main = "Huspriser")
plot(gold, col = "chartreuse3", lwd = "2", main = "Guldpriser")
```




```r
plot(short_i, col = "pink", lwd = "2", main = "Den korte rente")
```



Vi tjekker for seasonality

```r
##bnp
Q_bnp=ordered(cycle(bnp))
summary(y.reg <- lm(bnp ~ Q_bnp))
```

```
##
## Call:
## lm(formula = bnp ~ Q_bnp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -72.717 -30.461   1.509  17.668  84.278
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 455.18913    4.14129 109.915   <2e-16 ***
```

```
## Q_bnp.L       3.93937    8.28258   0.476    0.636
## Q_bnp.Q      -0.23913    8.28258  -0.029    0.977
## Q_bnp.C       0.04278    8.28258   0.005    0.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.72 on 88 degrees of freedom
## Multiple R-squared:  0.002574,   Adjusted R-squared:  -0.03143
## F-statistic: 0.07569 on 3 and 88 DF,  p-value: 0.9729
```

```
##C20
Q_C20=ordered(cycle(C20))
summary(y.reg <- lm(C20 ~ Q_C20))
```

```
##
## Call:
## lm(formula = C20 ~ Q_C20)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -326.4 -223.7 -103.5  211.8  586.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 488.4328    30.0588  16.249   <2e-16 ***
## Q_C20.L      13.5618    60.1176   0.226    0.822
## Q_C20.Q      -8.9651    60.1176  -0.149    0.882
## Q_C20.C       0.2447    60.1176   0.004    0.997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 288.3 on 88 degrees of freedom
## Multiple R-squared:  0.0008305,  Adjusted R-squared:  -0.03323
## F-statistic: 0.02438 on 3 and 88 DF,  p-value: 0.9948
```

```
##Lang Rente
Q_long_i=ordered(cycle(long_i))
summary(y.reg <- lm(long_i ~ Q_long_i))
```

```
##
## Call:
## lm(formula = long_i ~ Q_long_i)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5984 -1.8650  0.4796  1.4560  3.3161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.12612    0.20554  15.210   <2e-16 ***
## Q_long_i.L  -0.14683    0.41107  -0.357    0.722
## Q_long_i.Q  -0.05520    0.41107  -0.134    0.893
## Q_long_i.C   0.02351    0.41107   0.057    0.955
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.971 on 88 degrees of freedom
## Multiple R-squared:  0.001689,   Adjusted R-squared:  -0.03234
## F-statistic: 0.04963 on 3 and 88 DF,  p-value: 0.9853
```

```r
##Inflation
Q_inf=ordered(cycle(inf))
summary(y.reg <- lm(inf ~ Q_inf))
```

```
##
## Call:
## lm(formula = inf ~ Q_inf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6021 -0.8007  0.1454  0.6356  2.4515
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.731669   0.094809  18.265   <2e-16 ***
## Q_inf.L     -0.043528   0.189619  -0.230    0.819
## Q_inf.Q     -0.002765   0.189619  -0.015    0.988
## Q_inf.C     -0.011632   0.189619  -0.061    0.951
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9094 on 88 degrees of freedom
## Multiple R-squared:  0.0006436,  Adjusted R-squared:  -0.03343
## F-statistic: 0.01889 on 3 and 88 DF,  p-value: 0.9964
```

```r
##Huspriser
Q_hus=ordered(cycle(hus))
summary(y.reg <- lm(hus ~ Q_hus))
```

```
##
## Call:
## lm(formula = hus ~ Q_hus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.905 -23.765   6.444  19.963  34.210
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 83.4988722  2.4070235  34.690   <2e-16 ***
## Q_hus.L      1.8852220  4.8140470   0.392    0.696
## Q_hus.Q     -0.0754624  4.8140470  -0.016    0.988
## Q_hus.C      0.0004524  4.8140470   0.000    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.09 on 88 degrees of freedom
## Multiple R-squared:  0.001742,   Adjusted R-squared:  -0.03229
## F-statistic: 0.0512 on 3 and 88 DF,  p-value: 0.9846
```

```r
##Unemployment
Q_u=ordered(cycle(u))
```

```r
summary(y.reg <- lm(u ~ Q_u))
```

```
##
## Call:
## lm(formula = u ~ Q_u)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3130 -0.8446 -0.3319  0.8562  2.5348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.568116   0.135675  41.040   <2e-16 ***
## Q_u.L       -0.016203   0.271351  -0.060    0.953
## Q_u.Q       -0.004348   0.271351  -0.016    0.987
## Q_u.C       -0.012963   0.271351  -0.048    0.962
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.301 on 88 degrees of freedom
## Multiple R-squared:  6.937e-05,  Adjusted R-squared:  -0.03402
## F-statistic: 0.002035 on 3 and 88 DF,  p-value: 0.9999
```

```r
##Rate of Savings
Q_sav=ordered(cycle(sav))
summary(y.reg <- lm(sav ~ Q_sav))
```

```
##
## Call:
## lm(formula = sav ~ Q_sav)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0022 -4.3671 -0.0218  4.8926 10.1837
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.7867     0.5583  44.394   <2e-16 ***
## Q_sav.L       0.3940     1.1167   0.353    0.725
## Q_sav.Q       0.1706     1.1167   0.153    0.879
## Q_sav.C      -0.1380     1.1167  -0.124    0.902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.355 on 88 degrees of freedom
## Multiple R-squared:  0.00185,   Adjusted R-squared:  -0.03218
## F-statistic: 0.05436 on 3 and 88 DF,  p-value: 0.9832
```

```r
##Huspriser
Q_hus=ordered(cycle(hus))
summary(y.reg <- lm(hus ~ Q_hus))
```

```
##
## Call:
## lm(formula = hus ~ Q_hus)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.905 -23.765   6.444  19.963  34.210
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 83.4988722  2.4070235  34.690   <2e-16 ***
## Q_hus.L      1.8852220  4.8140470   0.392    0.696
## Q_hus.Q     -0.0754624  4.8140470  -0.016    0.988
## Q_hus.C      0.0004524  4.8140470   0.000    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.09 on 88 degrees of freedom
## Multiple R-squared:  0.001742,   Adjusted R-squared:  -0.03229
## F-statistic: 0.0512 on 3 and 88 DF,  p-value: 0.9846
```

```
##Kort Rente
Q_short_i=ordered(cycle(short_i))
summary(y.reg <- lm(short_i ~ Q_short_i))
```

```
##
## Call:
## lm(formula = short_i ~ Q_short_i)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5838 -1.8810  0.0307  1.5817  3.6844
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.153933   0.202764  10.623   <2e-16 ***
## Q_short_i.L  0.001044   0.405528   0.003    0.998
## Q_short_i.Q  0.003611   0.405528   0.009    0.993
## Q_short_i.C -0.012094   0.405528  -0.030    0.976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.945 on 88 degrees of freedom
## Multiple R-squared:  1.108e-05,  Adjusted R-squared:  -0.03408
## F-statistic: 0.0003251 on 3 and 88 DF,  p-value: 1
```

1 CCF tests

Fjerner trend og normaliserer.

```
lambda = 1600
hp = hpfilter(bnp, lambda)
bnp_cycle = hp$cycle + mean(hp$trend)

hp1 = hpfilter(C20, lambda)
c20_cycle = hp1$cycle + mean(hp1$trend)

hp2 = hpfilter(inf, lambda)
```

```
inf_cycle = hp2$cycle + mean(hp2$trend)

hp3 = hpfilter(sav, lambda)
sav_cycle = hp3$cycle + mean(hp3$trend)

hp4 = hpfilter(u, lambda)
u_cycle = hp4$cycle + mean(hp4$trend)

hp5 = hpfilter(long_i, lambda)
long_i_cycle = hp5$cycle + mean(hp5$trend)

hp6 = hpfilter(short_i, lambda)
short_i_cycle = hp6$cycle + mean(hp6$trend)

hp7 = hpfilter(hus, lambda)
hus_cycle = hp7$cycle + mean(hp7$trend)

hp8 = hpfilter(gold, lambda)
guld_cycle = hp8$cycle + mean(hp8$trend)
```

CCF test for independent variables.

```
##Long_i
cc_long_i_inf = ccf(inf_cycle, long_i_cycle, 1, pl = F);cc_long_i_inf##0.172
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.147 0.172 0.141
```

```
cc_long_i_bnp = ccf(bnp_cycle, long_i_cycle, 1, pl = F);cc_long_i_bnp##0.202
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.239 0.202 0.119
```

```
cc_long_i_sav = ccf(sav_cycle, long_i_cycle, 1, pl = F);cc_long_i_sav##0.077
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.096 0.077 0.062
```

```
cc_long_i_u = ccf(u_cycle, long_i_cycle, 1, pl = F);cc_long_i_u##-0.280
```

```
##
## Autocorrelations of series 'X', by lag
##
##  -0.25   0.00   0.25
## -0.278 -0.280 -0.259
```

```
cc_long_i_hus = ccf(hus_cycle, long_i_cycle, 1, pl = F);cc_long_i_hus##0.132
```

```
##
## Autocorrelations of series 'X', by lag
##
##  -0.25   0.00   0.25
##  0.239  0.132 -0.018
```

```
cc_long_i_short_i = ccf(short_i_cycle, long_i_cycle, 1, pl = F);cc_long_i_short_i##0.431
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.272 0.431 0.512
```

```
cc_long_i_guld = ccf(guld_cycle, long_i_cycle, 1, pl = F);cc_long_i_guld##-0.204
```

```
##
## Autocorrelations of series 'X', by lag
##
##  -0.25   0.00   0.25
## -0.243 -0.349 -0.297
```

```
##BNP
cc_bnp_sav = ccf(sav_cycle, bnp_cycle, 1, pl = F);cc_bnp_sav##0.655
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.513 0.655 0.618
```

```
cc_bnp_u = ccf(u_cycle, bnp_cycle, 1, pl = F);cc_bnp_u##-0.671
```

```
##
## Autocorrelations of series 'X', by lag
##
##  -0.25   0.00   0.25
## -0.495 -0.671 -0.788
```

```
cc_bnp_inf = ccf(inf_cycle, bnp_cycle, 1, pl = F);cc_bnp_inf##0.407
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.241 0.407 0.473
```

```
cc_bnp_hus = ccf(hus_cycle, bnp_cycle, 1, pl = F);cc_bnp_hus##0.801
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.812 0.801 0.716
```

```
cc_bnp_short_i = ccf(short_i_cycle, bnp_cycle, 1, pl = F);cc_bnp_short_i##0.505
```

```
##
## Autocorrelations of series 'X', by lag
```

```
##
## -0.25  0.00  0.25
## 0.340 0.555 0.719
```

```r
cc_bnp_guld = ccf(guld_cycle, bnp_cycle, 1, pl = F);cc_bnp_guld##0.136
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.135 0.147 0.120
```

```r
##inflation
cc_inf_sav = ccf(sav_cycle, inf_cycle, 1, pl = F);cc_inf_sav##0.305
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.279 0.305 0.251
```

```r
cc_inf_u = ccf(u_cycle, inf_cycle, 1, pl = F);cc_inf_u##-0.253
```

```
##
## Autocorrelations of series 'X', by lag
##
##  -0.25   0.00   0.25
## -0.151 -0.253 -0.269
```

```r
cc_inf_hus = ccf(hus_cycle, inf_cycle, 1, pl = F);cc_inf_hus##0.177
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.237 0.177 0.056
```

```r
cc_inf_short_i = ccf(short_i_cycle, inf_cycle, 1, pl = F);cc_inf_short_i##0.324
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.183 0.324 0.417
```

```r
cc_inf_guld = ccf(guld_cycle, inf_cycle, 1, pl = F);cc_inf_guld##0.407
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.304 0.358 0.377
```

```r
#Savings in percent of BNP
cc_sav_u = ccf(u_cycle, sav_cycle, 1, pl = F);cc_sav_u##-0.513
```

```
##
## Autocorrelations of series 'X', by lag
##
```

```
## -0.25   0.00   0.25
## -0.384 -0.513 -0.571
```

```
cc_sav_hus = ccf(hus_cycle, sav_cycle, 1, pl = F);cc_sav_hus##0.505
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.521 0.505 0.443
```

```
cc_sav_short_i = ccf(short_i_cycle, sav_cycle, 1, pl = F);cc_sav_short_i##0.403
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.271 0.403 0.485
```

```
cc_sav_guld = ccf(guld_cycle, sav_cycle, 1, pl = F);cc_sav_guld##-0.019
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.032 0.025 0.008
```

```
##Arbejdsløshed
cc_u_hus = ccf(hus_cycle, u_cycle, 1, pl = F);cc_u_hus##-0.712
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25   0.00   0.25
## -0.795 -0.712 -0.574
```

```
cc_u_short_i = ccf(short_i_cycle, u_cycle, 1, pl = F);cc_u_short_i##-0.810
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25   0.00   0.25
## -0.700 -0.810 -0.826
```

```
cc_u_guld = ccf(guld_cycle, u_cycle, 1, pl = F);cc_u_guld##0.251
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
## 0.185 0.237 0.284
```

```
##Huspris
cc_hus_short_i = ccf(short_i_cycle, hus_cycle, 1, pl = F);cc_hus_short_i##0.432
```

```
##
## Autocorrelations of series 'X', by lag
##
## -0.25  0.00  0.25
```

```
## 0.233 0.432 0.606
```

```
cc_hus_guld = ccf(guld_cycle, hus_cycle, 1, pl = F);cc_hus_guld##-0.123
```

```
##
## Autocorrelations of series 'X', by lag
##
##  -0.25   0.00   0.25
## -0.076 -0.088 -0.097
```

```
##Den korte rente
cc_short_i_guld = ccf(guld_cycle, short_i_cycle, 1, pl = F); cc_short_i_guld##-0.119
```

```
##
## Autocorrelations of series 'X', by lag
##
##  -0.25   0.00   0.25
## -0.092 -0.125 -0.129
```

2 ADF Test

2.1 ADF BNP

```
model_bnp_start <- dynlm(diff(bnp) ~ 1+ L(bnp, 1) + L(diff(bnp), 1:12) + trend(diff(bnp)))
summary(model_bnp_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(bnp) ~ 1 + L(bnp, 1) + L(diff(bnp), 1:12) +
##     trend(diff(bnp)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1179 -2.0573 -0.2851  1.9877  9.7854
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            18.45065   18.46227   0.999   0.3214
## L(bnp, 1)              -0.04636    0.04734  -0.979   0.3311
## L(diff(bnp), 1:12)1     0.21666    0.11911   1.819   0.0736 .
## L(diff(bnp), 1:12)2     0.21526    0.12107   1.778   0.0802 .
## L(diff(bnp), 1:12)3     0.12639    0.12351   1.023   0.3100
## L(diff(bnp), 1:12)4    -0.03581    0.12325  -0.291   0.7723
## L(diff(bnp), 1:12)5    -0.15857    0.11915  -1.331   0.1880
## L(diff(bnp), 1:12)6     0.18727    0.12082   1.550   0.1261
## L(diff(bnp), 1:12)7     0.05234    0.12027   0.435   0.6649
## L(diff(bnp), 1:12)8    -0.15766    0.11715  -1.346   0.1831
## L(diff(bnp), 1:12)9     0.22777    0.11643   1.956   0.0548 .
## L(diff(bnp), 1:12)10    0.00768    0.12238   0.063   0.9502
## L(diff(bnp), 1:12)11    0.07399    0.12032   0.615   0.5408
## L(diff(bnp), 1:12)12   -0.26986    0.11682  -2.310   0.0241 *
## trend(diff(bnp))        0.29503    0.24029   1.228   0.2240
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.501 on 64 degrees of freedom
## Multiple R-squared:  0.3101, Adjusted R-squared:  0.1592
## F-statistic: 2.055 on 14 and 64 DF,  p-value: 0.02685
```

```
res_model_bnp_start= model_bnp_start$residuals
```

Test for normality

```
shapiro.test(model_bnp_start$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model_bnp_start$residuals
## W = 0.98383, p-value = 0.4183
```

```
jarque.bera.test(res_model_bnp_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_bnp_start
## X-squared = 2.2876, df = 2, p-value = 0.3186
```

Vi fjerner lags med F-test

```
bnp_vars_1 <- str_c("L(diff(bnp), 1:12)", 3:12)

linearHypothesis(model_bnp_start,bnp_vars_1, rep(0, length(bnp_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(bnp),12)3 = 0
## L(diff(bnp),12)4 = 0
## L(diff(bnp),12)5 = 0
## L(diff(bnp),12)6 = 0
## L(diff(bnp),12)7 = 0
## L(diff(bnp),12)8 = 0
## L(diff(bnp),12)9 = 0
## L(diff(bnp),12)10 = 0
## L(diff(bnp),12)11 = 0
## L(diff(bnp),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(bnp) ~ 1 + L(bnp, 1) + L(diff(bnp), 1:12) + trend(diff(bnp))
##
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1     74 1001.54
## 2     64  784.43 10    217.11 1.7713 0.08418 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_bnp= ur.df(bnp, lags = 2, type = "trend")
summary(model_bnp)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8527 -1.9277  0.0508  2.2916 12.1558
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.53568   12.03895   1.706   0.0917 .
## z.lag.1     -0.05041    0.03072  -1.641   0.1045
## tt           0.07401    0.04350   1.702   0.0925 .
## z.diff.lag1  0.14283    0.10576   1.351   0.1805
## z.diff.lag2  0.27004    0.10523   2.566   0.0121 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.618 on 84 degrees of freedom
## Multiple R-squared:  0.1083, Adjusted R-squared:  0.06589
## F-statistic: 2.552 on 4 and 84 DF,  p-value: 0.04494
##
##
## Value of test-statistic is: -1.641 3.1117 1.4557
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Vi skal tage Diff af bnp

```
bnp_diff=diff(bnp)
```

Vi refitter vores model igen

```
model_bnp_diff_start <- dynlm(diff(bnp_diff) ~ 1+ L(bnp_diff, 1) + L(diff(bnp_diff), 1:12))
summary(model_bnp_diff_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(bnp_diff) ~ 1 + L(bnp_diff, 1) + L(diff(bnp_diff),
##     1:12))
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -8.1089 -2.3614 -0.0989  1.7731  9.0454
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.14181    0.59740   1.911   0.0604 .
## L(bnp_diff, 1)           -0.73885    0.27873  -2.651   0.0101 *
## L(diff(bnp_diff), 1:12)1 -0.02490    0.26081  -0.095   0.9242
## L(diff(bnp_diff), 1:12)2  0.15170    0.25582   0.593   0.5553
## L(diff(bnp_diff), 1:12)3  0.26521    0.24895   1.065   0.2907
## L(diff(bnp_diff), 1:12)4  0.17947    0.24241   0.740   0.4618
## L(diff(bnp_diff), 1:12)5  0.01612    0.22635   0.071   0.9434
## L(diff(bnp_diff), 1:12)6  0.18497    0.21636   0.855   0.3958
## L(diff(bnp_diff), 1:12)7  0.17987    0.20644   0.871   0.3868
## L(diff(bnp_diff), 1:12)8  0.02350    0.18839   0.125   0.9011
## L(diff(bnp_diff), 1:12)9  0.23638    0.17343   1.363   0.1777
## L(diff(bnp_diff), 1:12)10  0.20781    0.16834   1.234   0.2215
## L(diff(bnp_diff), 1:12)11  0.24420    0.15385   1.587   0.1174
## L(diff(bnp_diff), 1:12)12 -0.09088    0.11923  -0.762   0.4487
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.528 on 64 degrees of freedom
## Multiple R-squared:  0.568,  Adjusted R-squared:  0.4803
## F-statistic: 6.474 on 13 and 64 DF,  p-value: 1.227e-07
```

```
res_model_bnp_diff_start= model_bnp_diff_start$residuals
```

Vi tester for Normality

```
shapiro.test(res_model_bnp_diff_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_bnp_diff_start
## W = 0.99011, p-value = 0.8143
```

```
jarque.bera.test(res_model_bnp_diff_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_bnp_diff_start
## X-squared = 0.66613, df = 2, p-value = 0.7167
```

Vi kan nu fjerne lags med F-test

```
bnp_diff_vars_1 <- str_c("L(diff(bnp_diff), 1:12)", 2:12)

linearHypothesis(model_bnp_diff_start,bnp_diff_vars_1, rep(0, length(bnp_diff_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(bnp_diff),12)2 = 0
## L(diff(bnp_diff),12)3 = 0
```

```
## L(diff(bnp_diff),12)4 = 0
## L(diff(bnp_diff),12)5 = 0
## L(diff(bnp_diff),12)6 = 0
## L(diff(bnp_diff),12)7 = 0
## L(diff(bnp_diff),12)8 = 0
## L(diff(bnp_diff),12)9 = 0
## L(diff(bnp_diff),12)10 = 0
## L(diff(bnp_diff),12)11 = 0
## L(diff(bnp_diff),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(bnp_diff) ~ 1 + L(bnp_diff, 1) + L(diff(bnp_diff), 1:12)
##
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1     75 1037.09
## 2     64  796.62 11    240.47 1.7563 0.08099 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vi fjerner variablerne

```
model_bnp_diff <- dynlm(diff(bnp_diff) ~ 1+ L(bnp_diff, 1) + L(diff(bnp_diff), 1))
summary(model_bnp_diff)
```

```
##
## Time series regression with "ts" data:
## Start = 1997(4), End = 2019(4)
##
## Call:
## dynlm(formula = diff(bnp_diff) ~ 1 + L(bnp_diff, 1) + L(diff(bnp_diff),
##     1))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7872  -1.7263   0.1369   1.9562  11.4490
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.1442     0.4539   2.521   0.0135 *
## L(bnp_diff, 1)         -0.6499     0.1365  -4.763 7.67e-06 ***
## L(diff(bnp_diff), 1)   -0.2354     0.1032  -2.281   0.0250 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.637 on 86 degrees of freedom
## Multiple R-squared:  0.461,  Adjusted R-squared:  0.4484
## F-statistic: 36.77 on 2 and 86 DF,  p-value: 2.882e-12
```

```
model_bnp_diff1 <- ur.df(bnp_diff, lags = 1, type = "drift")
summary(model_bnp_diff)
```

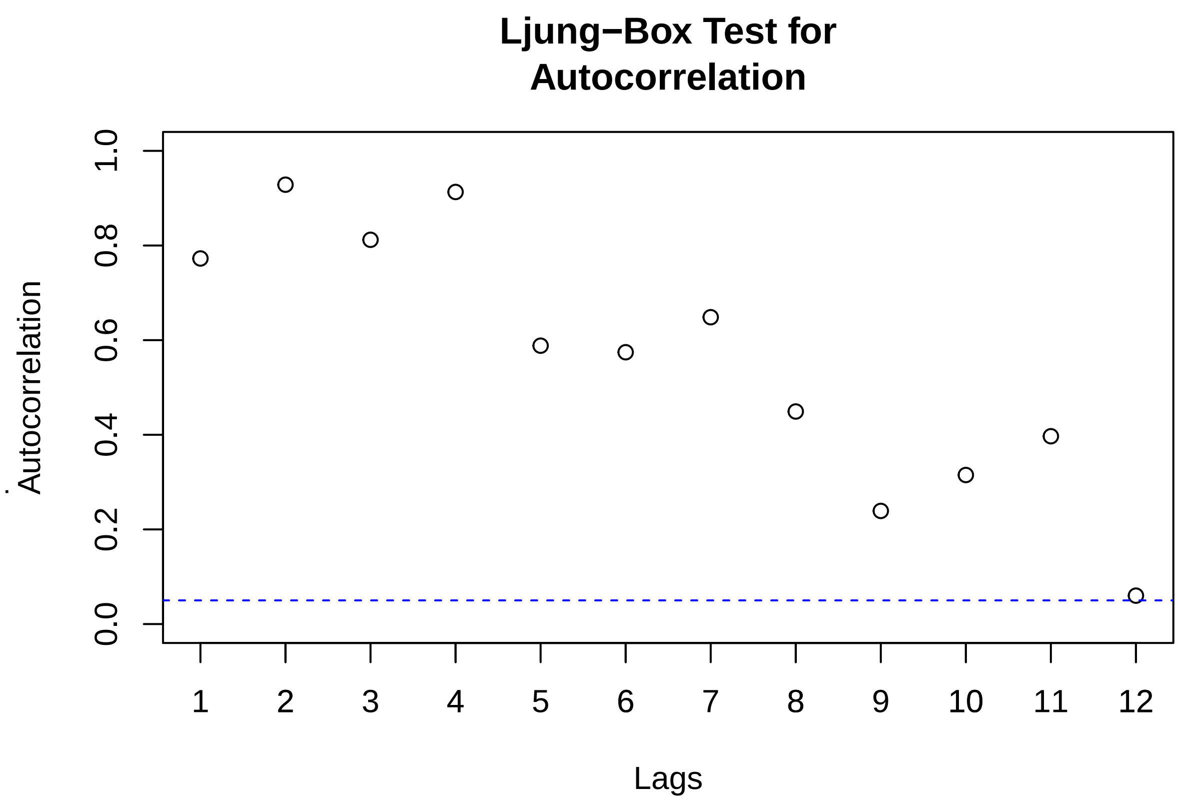
```
##
## Time series regression with "ts" data:
## Start = 1997(4), End = 2019(4)
##
## Call:
```

```
## dynlm(formula = diff(bnp_diff) ~ 1 + L(bnp_diff, 1) + L(diff(bnp_diff),
##     1))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7872  -1.7263   0.1369   1.9562  11.4490
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.1442     0.4539   2.521   0.0135 *
## L(bnp_diff, 1)       -0.6499     0.1365  -4.763 7.67e-06 ***
## L(diff(bnp_diff), 1)  -0.2354     0.1032  -2.281   0.0250 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.637 on 86 degrees of freedom
## Multiple R-squared:  0.461,  Adjusted R-squared:  0.4484
## F-statistic: 36.77 on 2 and 86 DF,  p-value: 2.882e-12
```

```
res_model_bnp_diff= residuals(model_bnp_diff)
```

Vi tjekker for Serial correlation

```
arpdiag(res_model_bnp_diff, 12)
```




```
Box.test(res_model_bnp_diff, lag=8, type = "Ljung")
```

```
##
##  Box-Ljung test
##
```

```
## data:  res_model_bnp_diff
## X-squared = 7.8405, df = 8, p-value = 0.4492
```

```
Box.test(res_model_bnp_diff, lag=12, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  res_model_bnp_diff
## X-squared = 20.392, df = 12, p-value = 0.06002
```

```
Box.test(res_model_bnp_diff, lag=16, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  res_model_bnp_diff
## X-squared = 21.946, df = 16, p-value = 0.1449
```

Vi kan se BNP er I(1)

2.2 ADF lange rente

```
model_long_i_start <- dynlm(diff(long_i) ~ 1+ L(long_i, 1) + L(diff(long_i), 1:12) + trend(diff(long_i)
summary(model_long_i_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(long_i) ~ 1 + L(long_i, 1) + L(diff(long_i),
##     1:12) + trend(diff(long_i)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61439 -0.18545 -0.01712  0.16861  0.58461
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.205689   0.625464   1.928   0.0583 .
## L(long_i, 1)           -0.186269   0.090077  -2.068   0.0427 *
## L(diff(long_i), 1:12)1  0.280535   0.129371   2.168   0.0338 *
## L(diff(long_i), 1:12)2 -0.094438   0.134461  -0.702   0.4850
## L(diff(long_i), 1:12)3  0.032453   0.129792   0.250   0.8034
## L(diff(long_i), 1:12)4 -0.009222   0.128791  -0.072   0.9431
## L(diff(long_i), 1:12)5  0.039068   0.126929   0.308   0.7592
## L(diff(long_i), 1:12)6 -0.049245   0.126201  -0.390   0.6977
## L(diff(long_i), 1:12)7  0.061722   0.123190   0.501   0.6181
## L(diff(long_i), 1:12)8  0.031471   0.123395   0.255   0.7995
## L(diff(long_i), 1:12)9 -0.004516   0.120299  -0.038   0.9702
## L(diff(long_i), 1:12)10  0.140329   0.118834   1.181   0.2420
## L(diff(long_i), 1:12)11  0.080520   0.116599   0.691   0.4923
## L(diff(long_i), 1:12)12 -0.115223   0.115058  -1.001   0.3204
## trend(diff(long_i))     -0.056067   0.026492  -2.116   0.0382 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2761 on 64 degrees of freedom
## Multiple R-squared:  0.2093, Adjusted R-squared:  0.03632
## F-statistic:  1.21 on 14 and 64 DF,  p-value: 0.2905
```

```
res_model_long_i_start = model_long_i_start$residuals
```

Test for normality

```
shapiro.test(res_model_long_i_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_long_i_start
## W = 0.98891, p-value = 0.7325
```

```
jarque.bera.test(res_model_long_i_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_long_i_start
## X-squared = 1.102, df = 2, p-value = 0.5764
```

Da denne p-v?rdi > 10%-signifikansniveau (0.1), kan vi ikke forkaste nul-hypotesen. Dermed er residualerne normalfordelt.

Vi fjerner lags med F-test

```
long_i_vars_1 <- str_c("L(diff(long_i), 1:12)", 2:12)

linearHypothesis(model_long_i_start, long_i_vars_1, rep(0, length(long_i_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(long_i),12)2 = 0
## L(diff(long_i),12)3 = 0
## L(diff(long_i),12)4 = 0
## L(diff(long_i),12)5 = 0
## L(diff(long_i),12)6 = 0
## L(diff(long_i),12)7 = 0
## L(diff(long_i),12)8 = 0
## L(diff(long_i),12)9 = 0
## L(diff(long_i),12)10 = 0
## L(diff(long_i),12)11 = 0
## L(diff(long_i),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(long_i) ~ 1 + L(long_i, 1) + L(diff(long_i), 1:12) + trend(diff(long_i))
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     75 5.2899
## 2     64 4.8783 11   0.41157 0.4909 0.9022
```

```r
model_long_i = ur.df(long_i, lags = 1, type = "trend")
summary(model_long_i)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64625 -0.18764 -0.04663  0.16850  0.70014
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.105358   0.345626   3.198 0.001937 **
## z.lag.1     -0.182941   0.053357  -3.429 0.000933 ***
## tt          -0.012559   0.003851  -3.261 0.001592 **
## z.diff.lag   0.329223   0.102296   3.218 0.001819 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2752 on 86 degrees of freedom
## Multiple R-squared:  0.1684, Adjusted R-squared:  0.1394
## F-statistic: 5.806 on 3 and 86 DF,  p-value: 0.00116
##
##
## Value of test-statistic is: -3.4286 5.1762 5.8801
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Denne proces starter fra det h?jeste lag, som reduceres tilstr?kkeligt. Vi reducere ned til f?rste signifikante lag for modellen. Her finder vi at det 1. lag er signifikant, og derfor inkluderes 1 lag i modellen. Efter at have fundet en model uden "serial correlation", s? tester vi nul-hypotesen, ved at kigge p? t-stat og sammenligner denne med de kritiske v?rdi (tau-v?rdier), som vi direkte kan opn? gennem ur.df-modellen. Vi kan tjekke dette resultat for antal lags, gennem AIC og BIC for modellen.

Hvis $t < \tau$, s? forkaster vi $H_0 : \pi = 0$, og dermed forkaster vi at der er en unit-root i tidsserien for DK-kvartal-long_i. Af vores ADF-test f?r vi at $t = -3.4286$ og $\tau = -3.45$ (p? 5%-signifikansniveau). Af denne ?rsag kan vi lige n?jagtig ikke forkaste nul-hypotesen, og der er dermed en unit-root i tidsserien for DK-kvartal-long_i.

Vi differentiere long_i

```r
long_i_diff= diff(long_i)
```

```r
#First diff, I(1)

model_long_i_diff_start <- dynlm(diff(long_i_diff) ~ 1+ L(long_i_diff, 1) + L(diff(long_i_diff), 1:12) +
summary(model_long_i_diff_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(long_i_diff) ~ 1 + L(long_i_diff, 1) + L(diff(long_i_diff),
##     1:12) + trend(diff(long_i_diff)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62000 -0.18930 -0.01483  0.20561  0.51644
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -0.087074   0.082350  -1.057  0.29439
## L(long_i_diff, 1)             -1.788881   0.540080  -3.312  0.00154 **
## L(diff(long_i_diff), 1:12)1    0.927739   0.506218   1.833  0.07158 .
## L(diff(long_i_diff), 1:12)2    0.721027   0.486569   1.482  0.14336
## L(diff(long_i_diff), 1:12)3    0.652264   0.458230   1.423  0.15954
## L(diff(long_i_diff), 1:12)4    0.544143   0.421661   1.290  0.20160
## L(diff(long_i_diff), 1:12)5    0.481686   0.378525   1.273  0.20786
## L(diff(long_i_diff), 1:12)6    0.344673   0.339508   1.015  0.31389
## L(diff(long_i_diff), 1:12)7    0.319635   0.297248   1.075  0.28634
## L(diff(long_i_diff), 1:12)8    0.273904   0.258122   1.061  0.29268
## L(diff(long_i_diff), 1:12)9    0.198496   0.219583   0.904  0.36946
## L(diff(long_i_diff), 1:12)10   0.274494   0.183265   1.498  0.13918
## L(diff(long_i_diff), 1:12)11   0.279693   0.145320   1.925  0.05879 .
## L(diff(long_i_diff), 1:12)12   0.128055   0.118481   1.081  0.28390
## trend(diff(long_i_diff))      -0.003129   0.005965  -0.524  0.60178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2848 on 63 degrees of freedom
## Multiple R-squared:  0.5001, Adjusted R-squared:  0.389
## F-statistic: 4.502 on 14 and 63 DF,  p-value: 1.577e-05
```

```r
res_model_long_i_diff_start = model_long_i_diff_start$residuals
```

Test for normality

```r
shapiro.test(res_model_long_i_diff_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_long_i_diff_start
## W = 0.98398, p-value = 0.4354
```

```r
jarque.bera.test(res_model_long_i_diff_start)
```

```
##
```

```
##  Jarque Bera Test
##
## data:  res_model_long_i_diff_start
## X-squared = 1.5443, df = 2, p-value = 0.462
```

Vi fjerner lags med F-test

```r
long_i_diff_vars_1 <- str_c("L(diff(long_i_diff), 1:12)", 2:12)

linearHypothesis(model_long_i_diff_start, long_i_diff_vars_1, rep(0, length(long_i_diff_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(long_i_diff),12)2 = 0
## L(diff(long_i_diff),12)3 = 0
## L(diff(long_i_diff),12)4 = 0
## L(diff(long_i_diff),12)5 = 0
## L(diff(long_i_diff),12)6 = 0
## L(diff(long_i_diff),12)7 = 0
## L(diff(long_i_diff),12)8 = 0
## L(diff(long_i_diff),12)9 = 0
## L(diff(long_i_diff),12)10 = 0
## L(diff(long_i_diff),12)11 = 0
## L(diff(long_i_diff),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(long_i_diff) ~ 1 + L(long_i_diff, 1) + L(diff(long_i_diff),
##     1:12) + trend(diff(long_i_diff))
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     74 5.7143
## 2     63 5.1085 11   0.60577 0.6791 0.7529
```

S? vi ender med

```r
model_long_i_diff <- ur.df(diff(long_i), lags = 1, type = "drift")
summary(model_long_i_diff)
```

```
##
## ###############################################
## # 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
## -0.63892 -0.19786 -0.02784  0.18081  0.78567
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

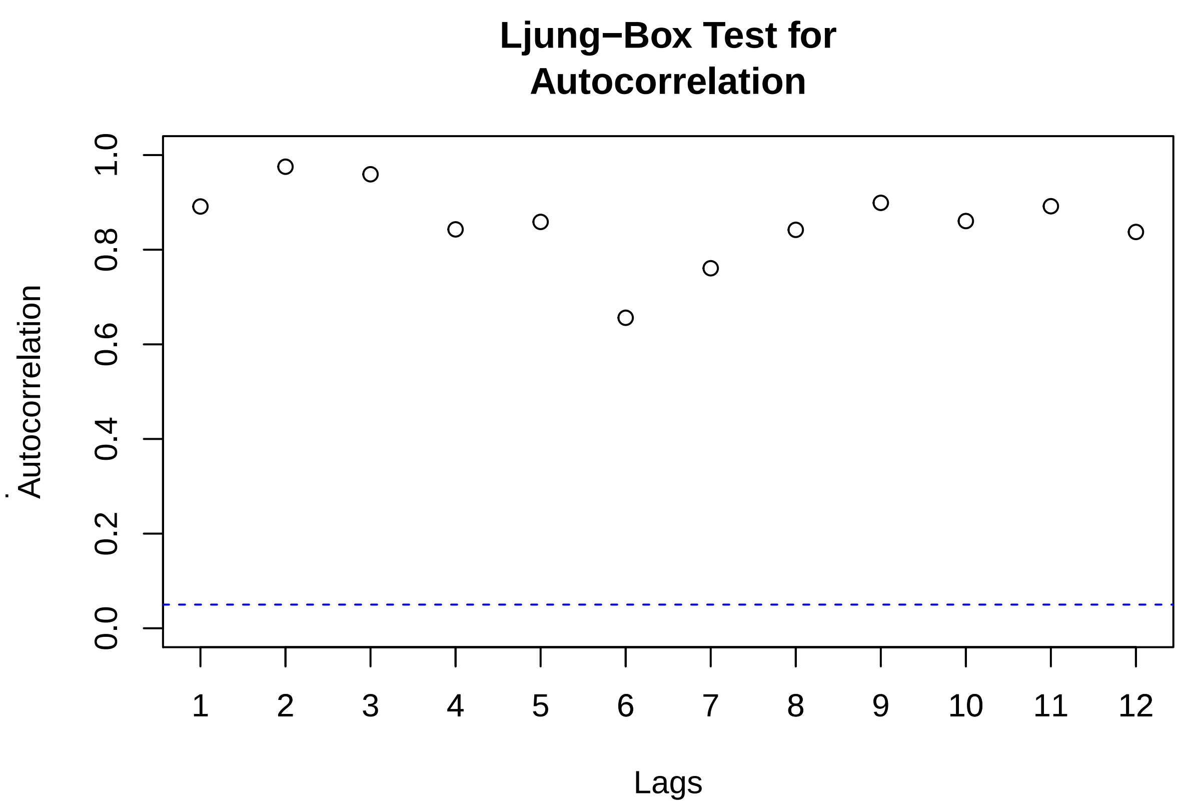
```
## (Intercept) -0.06302    0.03200  -1.969   0.0522 .
## z.lag.1     -0.88672    0.13076  -6.781 1.43e-09 ***
## z.diff.lag   0.17133    0.10707   1.600   0.1132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2872 on 86 degrees of freedom
## Multiple R-squared:  0.3974, Adjusted R-squared:  0.3834
## F-statistic: 28.36 on 2 and 86 DF,  p-value: 3.47e-10
##
##
## Value of test-statistic is: -6.7812 23.0007
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

```
res_model_long_i_diff=model_long_i_diff@testreg$residuals
```

Da vi finder, at der er en unit-root i tidsserien, s? kan vi differentiere p? long_i, og gentage processen, for at unders?ge om der herefter er en unit-root i tidsserien. Hvis $t < \tau$, s? forkaster vi $H_0 : \pi = 0$, og dermed forkaster vi at der er en unit-root i tidsserien for DK-kvartal-long_i. Af vores I(1)-ADF-test f?r vi at $t = -6.7812$ og $\tau = -3.51$ (p? 1%-signifikansniveau). Af denne ?rsag kan vi forkaste nul-hypotesen, og der er dermed ikke l?ngere en unit-root i tidsserien for DK-kvartal-long_i.

Tjekke for serial correlation

```
arpdiag(res_model_long_i_diff, 12)
```



```r
Box.test(res_model_bnp_diff, lag=8, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  res_model_bnp_diff
## X-squared = 7.8405, df = 8, p-value = 0.4492
```

No serial correlation

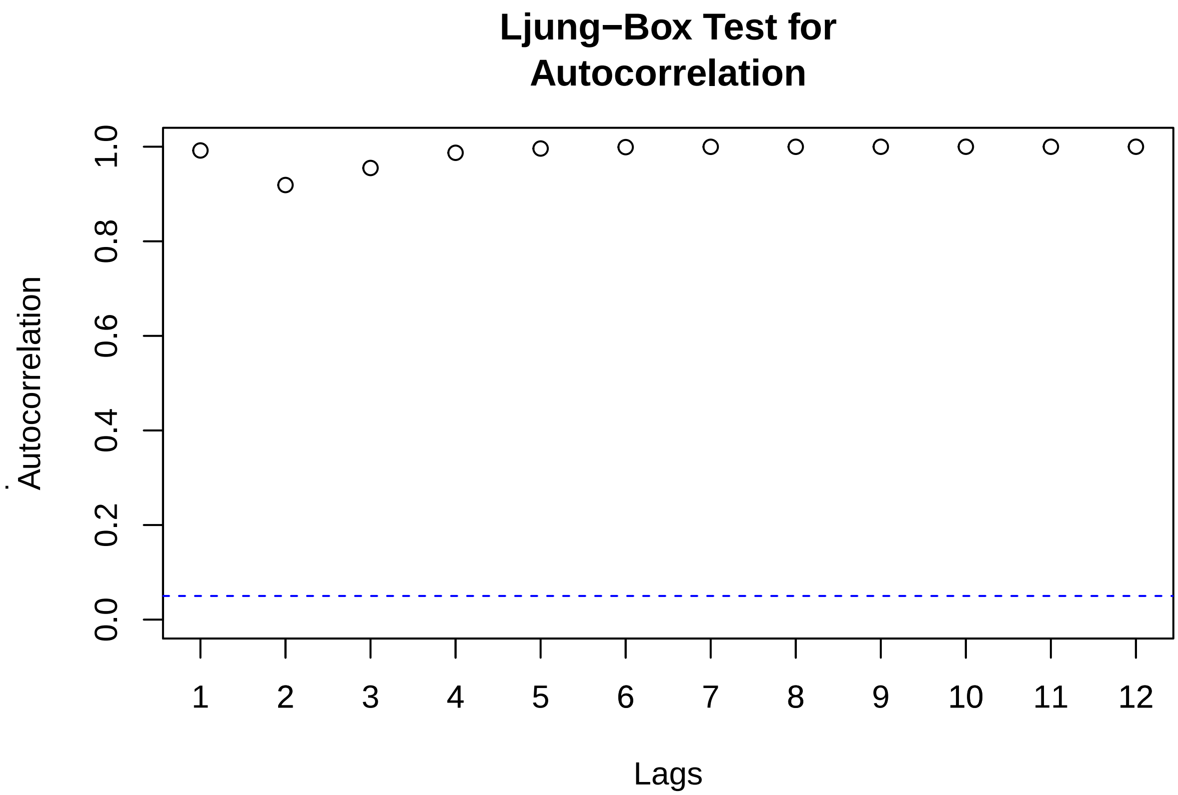
2.3 ADF test C20

```r
LC20 = log(C20)
model_LC20_start <- dynlm(diff(LC20) ~ 1+ L(LC20, 1) + L(diff(LC20), 1:12) + trend(diff(LC20)))
summary(model_LC20_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(LC20) ~ 1 + L(LC20, 1) + L(diff(LC20), 1:12) +
##     trend(diff(LC20)))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.293993 -0.027918  0.008472  0.043508  0.145808
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.867455   0.349614   2.481 0.015733 *
## L(LC20, 1)            -0.172522   0.069584  -2.479 0.015808 *
## L(diff(LC20), 1:12)1   0.461085   0.120997   3.811 0.000314 ***
## L(diff(LC20), 1:12)2   0.013536   0.134431   0.101 0.920113
## L(diff(LC20), 1:12)3   0.072299   0.133581   0.541 0.590222
## L(diff(LC20), 1:12)4   0.115588   0.133173   0.868 0.388662
## L(diff(LC20), 1:12)5   0.087536   0.134339   0.652 0.516988
## L(diff(LC20), 1:12)6  -0.041322   0.134409  -0.307 0.759512
## L(diff(LC20), 1:12)7   0.064623   0.132137   0.489 0.626471
## L(diff(LC20), 1:12)8  -0.059337   0.131160  -0.452 0.652506
## L(diff(LC20), 1:12)9   0.055983   0.129930   0.431 0.668011
## L(diff(LC20), 1:12)10  0.027014   0.129313   0.209 0.835185
## L(diff(LC20), 1:12)11  0.164179   0.126400   1.299 0.198644
## L(diff(LC20), 1:12)12 -0.036819   0.120928  -0.304 0.761760
## trend(diff(LC20))      0.014619   0.005582   2.619 0.011000 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07941 on 64 degrees of freedom
## Multiple R-squared:  0.2841, Adjusted R-squared:  0.1276
## F-statistic: 1.815 on 14 and 64 DF,  p-value: 0.05548
```

```r
res_model_LC20_start= model_LC20_start$residuals
```

```r
arpdiag(res_model_LC20_start, 12)
```



Test for normality

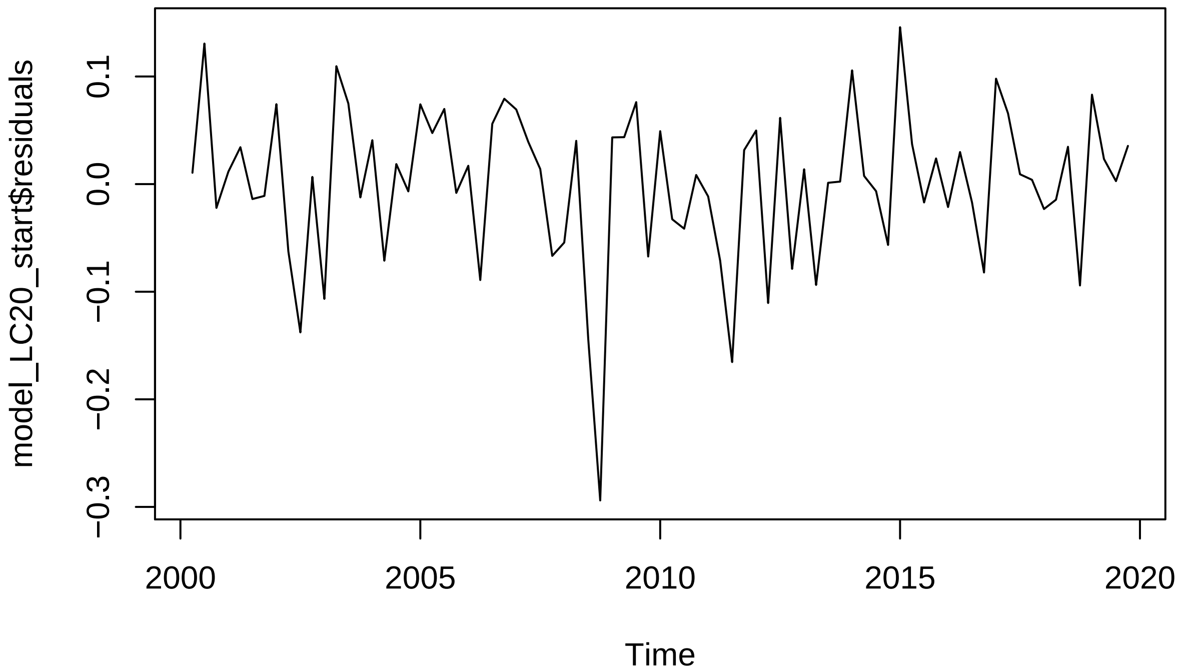
```r
shapiro.test(res_model_LC20_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_LC20_start
## W = 0.94624, p-value = 0.002282
```

```r
jarque.bera.test(res_model_LC20_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_LC20_start
## X-squared = 33.19, df = 2, p-value = 6.206e-08
```

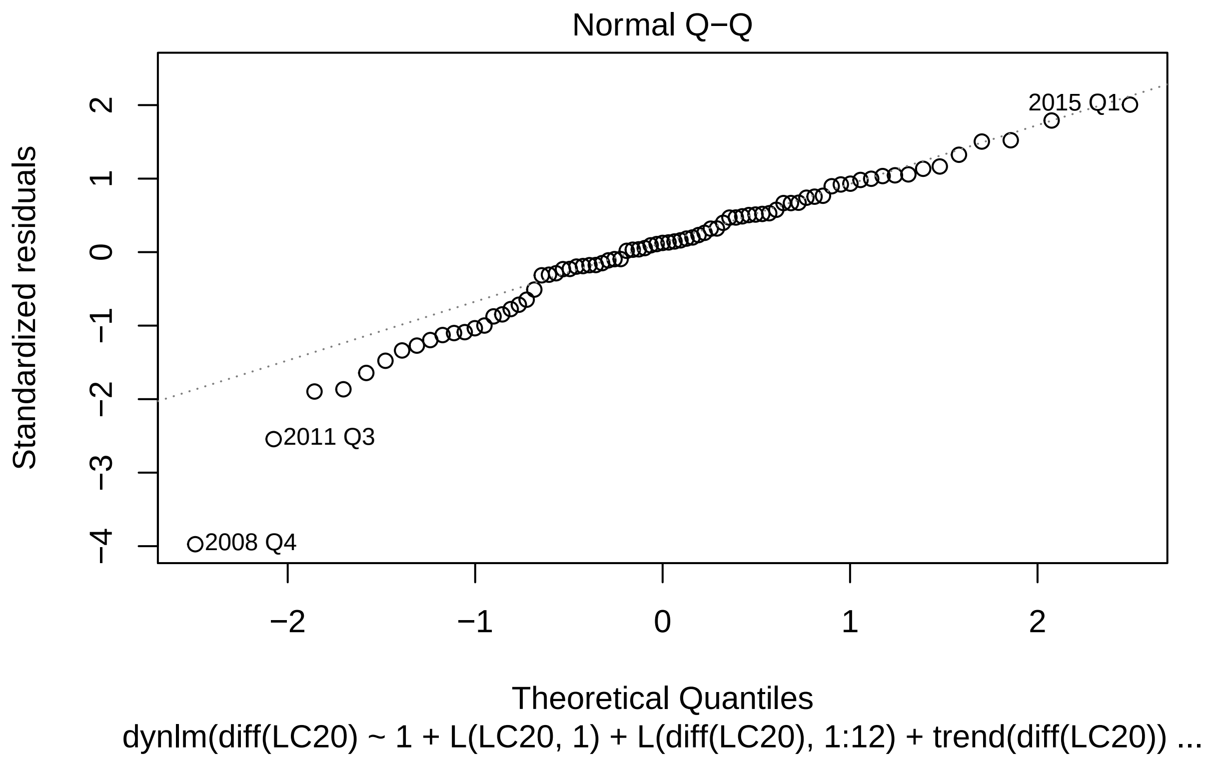
```r
#Residualerne er ikke normaltfordelte.. Derfor dummies. Først bestemmes hvilke der skal laves dummies f
plot(model_LC20_start$residuals)
```




```
model_LC20_start$residuals
```

```
##              Qtr1         Qtr2         Qtr3         Qtr4
## 2000              0.010530471  0.130622554 -0.022102365
## 2001  0.011618181  0.034287279 -0.013869102 -0.010979773
## 2002  0.074258069 -0.062611397 -0.137806132  0.006657105
## 2003 -0.106641514  0.109551290  0.074870556 -0.012375225
## 2004  0.040773005 -0.071164568  0.018583327 -0.006760549
## 2005  0.074144804  0.047445856  0.069822080 -0.008178376
## 2006  0.017017752 -0.089149539  0.056006534  0.079312952
## 2007  0.069334907  0.039252666  0.014005966 -0.066695333
## 2008 -0.054323864  0.040202378 -0.143896794 -0.293993350
## 2009  0.043410872  0.043605546  0.076163504 -0.067288522
## 2010  0.049197806 -0.032669832 -0.041453661  0.008472041
## 2011 -0.011461483 -0.071179738 -0.165384626  0.031619501
## 2012  0.049803282 -0.110493273  0.061545495 -0.078769604
## 2013  0.013784203 -0.093699469  0.001239131  0.002309140
## 2014  0.105688714  0.007584302 -0.006555492 -0.056559424
## 2015  0.145807751  0.037140228 -0.017045966  0.023836183
## 2016 -0.021305523  0.029798408 -0.017108701 -0.082154823
## 2017  0.097969043  0.065719960  0.009150304  0.003987853
## 2018 -0.023165954 -0.014503368  0.034690122 -0.094194300
## 2019  0.083117338  0.023324357  0.002777062  0.035501762
```

```
plot(model_LC20_start,2)
```




```
#Sidste plot viser at der skal laves dummies for 2008Q3 - 2008Q4 og for 2011Q3.
dummy_LC20_2008=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2008,2), du

dummy_LC20_2011=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2011,3), du

dummies_LC20 = cbind(dummy_LC20_2008, dummy_LC20_2011)
```

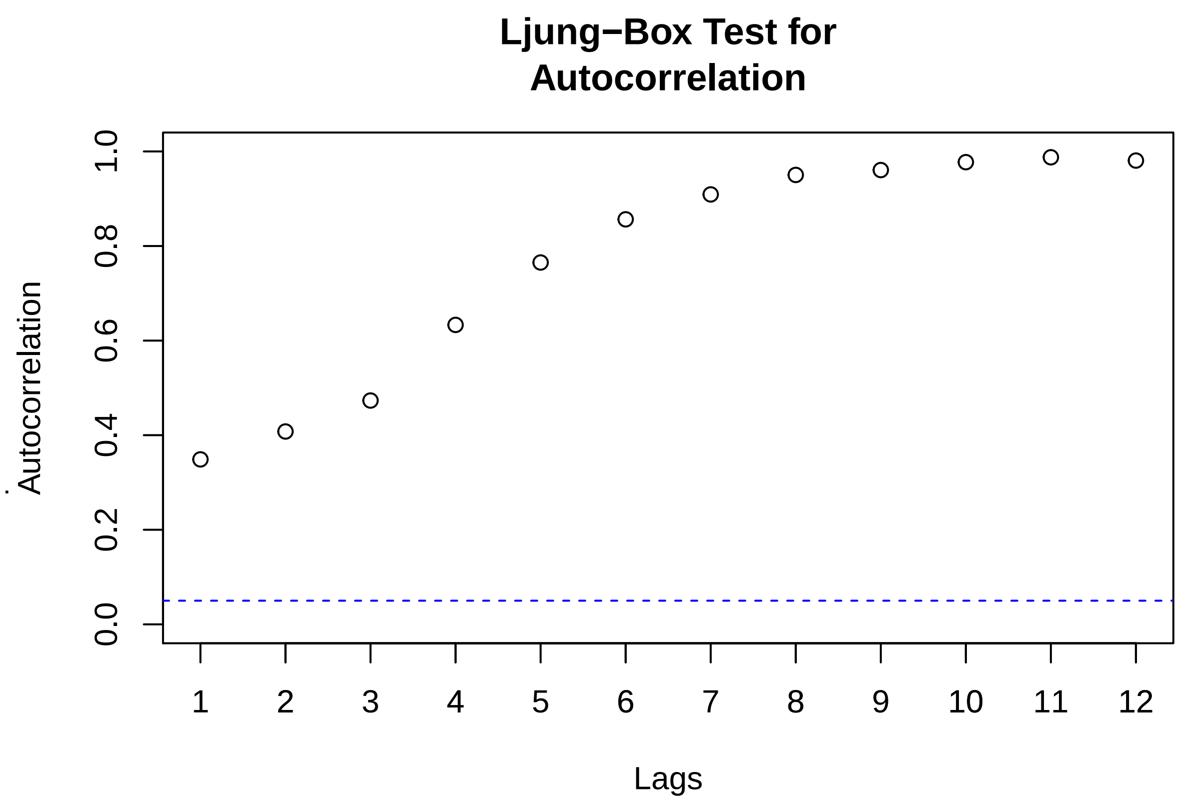
Ny model med de nye dummies

```
model_LC20_start_2 <- dynlm(diff(LC20) ~ 1+ L(LC20, 1) + L(diff(LC20), 1:12) + trend(diff(LC20)) + dummi
summary(model_LC20_start_2)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(LC20) ~ 1 + L(LC20, 1) + L(diff(LC20), 1:12) +
##     trend(diff(LC20)) + dummies_LC20)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.174655 -0.031554  0.003558  0.037636  0.180869
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.9423997  0.3120518   3.020 0.003668 **
## L(LC20, 1)                  -0.1848317  0.0621067  -2.976 0.004159 **
## L(diff(LC20), 1:12)1         0.3715166  0.1088197   3.414 0.001133 **
## L(diff(LC20), 1:12)2         0.0339224  0.1193928   0.284 0.777262
## L(diff(LC20), 1:12)3         0.0487654  0.1190755   0.410 0.683560
```

```
## L(diff(LC20), 1:12)4         0.0968473  0.1174451   0.825 0.412750
## L(diff(LC20), 1:12)5         0.1059385  0.1184772   0.894 0.374690
## L(diff(LC20), 1:12)6         0.0354424  0.1200850   0.295 0.768870
## L(diff(LC20), 1:12)7         0.0372442  0.1175187   0.317 0.752368
## L(diff(LC20), 1:12)8         0.0122271  0.1195356   0.102 0.918858
## L(diff(LC20), 1:12)9         0.0557159  0.1146985   0.486 0.628850
## L(diff(LC20), 1:12)10        0.0296633  0.1147692   0.258 0.796908
## L(diff(LC20), 1:12)11        0.0416843  0.1224615   0.340 0.734716
## L(diff(LC20), 1:12)12        0.0009176  0.1069084   0.009 0.993179
## trend(diff(LC20))            0.0154239  0.0049871   3.093 0.002972 **
## dummies_LC20dummy_LC20_2008 -0.1520051  0.0437418  -3.475 0.000938 ***
## dummies_LC20dummy_LC20_2011 -0.2388251  0.0855104  -2.793 0.006938 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06998 on 62 degrees of freedom
## Multiple R-squared:  0.4614, Adjusted R-squared:  0.3225
## F-statistic:  3.32 on 16 and 62 DF,  p-value: 0.0003458
```

```r
res_model_LC20_start_2= model_LC20_start_2$residuals

#Ljungbox test
arpdiag(res_model_LC20_start_2, 12)
```




```r
#Normality
shapiro.test(res_model_LC20_start_2)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  res_model_LC20_start_2
## W = 0.98402, p-value = 0.4286
```

```r
jarque.bera.test(res_model_LC20_start_2)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_LC20_start_2
## X-squared = 1.8027, df = 2, p-value = 0.406
```

Normaliteten er meget bedre nu.

Vi fjerner lags med F-test

```r
LC20_vars_1 <- str_c("L(diff(LC20), 1:12)", c(2:12))

linearHypothesis(model_LC20_start_2,LC20_vars_1, rep(0, length(LC20_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(LC20),12)2 = 0
## L(diff(LC20),12)3 = 0
## L(diff(LC20),12)4 = 0
## L(diff(LC20),12)5 = 0
## L(diff(LC20),12)6 = 0
## L(diff(LC20),12)7 = 0
## L(diff(LC20),12)8 = 0
## L(diff(LC20),12)9 = 0
## L(diff(LC20),12)10 = 0
## L(diff(LC20),12)11 = 0
## L(diff(LC20),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(LC20) ~ 1 + L(LC20, 1) + L(diff(LC20), 1:12) + trend(diff(LC20)) +
##     dummies_LC20
##
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     73 0.31663
## 2     62 0.30362 11  0.013005 0.2414 0.9933
```

```r
model_LC20 <- dynlm(diff(LC20) ~ 1+ L(LC20, 1) + L(diff(LC20), 1) + trend(diff(LC20)) + dummies_LC20)
summary(model_LC20)
```

```
##
## Time series regression with "ts" data:
## Start = 1997(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(LC20) ~ 1 + L(LC20, 1) + L(diff(LC20), 1) +
##     trend(diff(LC20)) + dummies_LC20)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.179606 -0.030694  0.000109  0.036806  0.190755
##
```

```
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  0.673831   0.180155   3.740 0.000335 ***
## L(LC20, 1)                  -0.127611   0.035041  -3.642 0.000467 ***
## L(diff(LC20), 1)             0.360599   0.086552   4.166 7.48e-05 ***
## trend(diff(LC20))            0.010100   0.002918   3.461 0.000848 ***
## dummies_LC20dummy_LC20_2008 -0.157459   0.039605  -3.976 0.000148 ***
## dummies_LC20dummy_LC20_2011 -0.230804   0.066780  -3.456 0.000862 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.0659 on 84 degrees of freedom
## Multiple R-squared:  0.424,  Adjusted R-squared:  0.3897 
## F-statistic: 12.36 on 5 and 84 DF,  p-value: 5.377e-09
```

```
model_LC20_ur= ur.df(LC20, lags = 1, type = "trend")
summary(model_LC20_ur)
```

```
## 
## ############################################### 
## # Augmented Dickey-Fuller Test Unit Root Test # 
## ############################################### 
## 
## Test regression trend 
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.313248 -0.026724  0.005439  0.048484  0.150741 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.643446   0.203945   3.155  0.00221 ** 
## z.lag.1     -0.123303   0.039663  -3.109  0.00255 ** 
## tt           0.002409   0.000825   2.920  0.00446 ** 
## z.diff.lag   0.452697   0.095789   4.726 8.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.0748 on 86 degrees of freedom
## Multiple R-squared:  0.2402, Adjusted R-squared:  0.2137 
## F-statistic: 9.063 on 3 and 86 DF,  p-value: 2.804e-05
## 
## 
## Value of test-statistic is: -3.1087 4.0258 4.8354 
## 
## Critical values for test statistics: 
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```r
PP.test(LC20)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  LC20
## Dickey-Fuller = -2.5931, Truncation lag parameter = 3, p-value = 0.3318
```

Godt nok viser ADF testen kun lige akkurat, at Log C20 er I(1), men Phillip Perron testen siger klart, at den er I(1), så derfor konkluderes det, at vi tager diff for at teste om den er I(2).

```r
LC20_diff=diff(LC20) #change back to LC20
```

Vi refitter vores model igen

```r
model_LC20_diff_start <- dynlm(diff(LC20_diff) ~ 1+ L(LC20_diff, 1) + L(diff(LC20_diff), 1:12))
summary(model_LC20_diff_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(LC20_diff) ~ 1 + L(LC20_diff, 1) + L(diff(LC20_diff),
##     1:12))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.295005 -0.040910  0.002422  0.054980  0.176295
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.02038    0.01204   1.692  0.09553 .
## L(LC20_diff, 1)            -1.11275    0.37634  -2.957  0.00435 **
## L(diff(LC20_diff), 1:12)1   0.50268    0.35184   1.429  0.15795
## L(diff(LC20_diff), 1:12)2   0.40253    0.33796   1.191  0.23803
## L(diff(LC20_diff), 1:12)3   0.36494    0.31782   1.148  0.25513
## L(diff(LC20_diff), 1:12)4   0.37560    0.29770   1.262  0.21164
## L(diff(LC20_diff), 1:12)5   0.35795    0.27174   1.317  0.19245
## L(diff(LC20_diff), 1:12)6   0.21962    0.25377   0.865  0.39004
## L(diff(LC20_diff), 1:12)7   0.20596    0.22829   0.902  0.37034
## L(diff(LC20_diff), 1:12)8   0.07853    0.20527   0.383  0.70329
## L(diff(LC20_diff), 1:12)9   0.07678    0.18487   0.415  0.67930
## L(diff(LC20_diff), 1:12)10  0.04336    0.16228   0.267  0.79017
## L(diff(LC20_diff), 1:12)11  0.15777    0.13827   1.141  0.25811
## L(diff(LC20_diff), 1:12)12  0.04396    0.12392   0.355  0.72397
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08345 on 64 degrees of freedom
## Multiple R-squared:  0.3674, Adjusted R-squared:  0.239
## F-statistic:  2.86 on 13 and 64 DF,  p-value: 0.002671
```

```r
res_model_LC20_diff_start= model_LC20_diff_start$residuals
```

Vi tester for serial correlation

```r
shapiro.test(res_model_LC20_diff_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_LC20_diff_start
## W = 0.96805, p-value = 0.04716
```

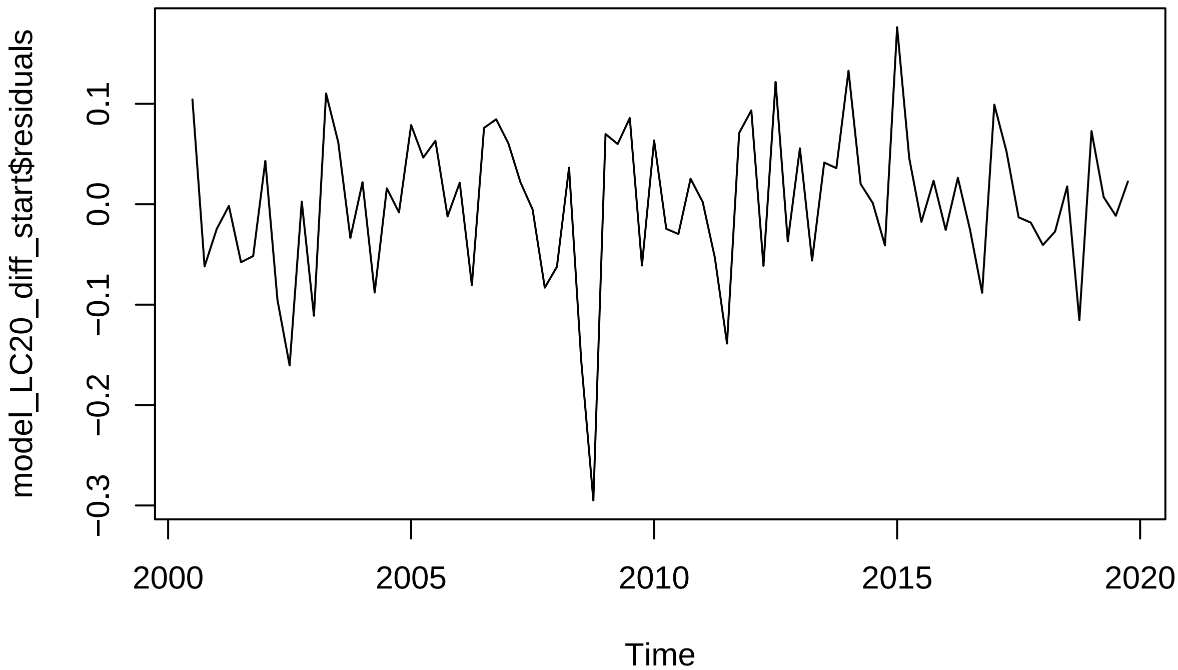
```r
jarque.bera.test(res_model_LC20_diff_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_LC20_diff_start
## X-squared = 17.878, df = 2, p-value = 0.0001311
```

```r
#Residualerne er ikke normaltfordelte.. Derfor dummies. Først bestemmes hvilke der skal laves dummies f
plot(model_LC20_diff_start$residuals)

model_LC20_diff_start$residuals
```

```
##              Qtr1         Qtr2         Qtr3         Qtr4
## 2000                           0.104357266 -0.061907041
## 2001 -0.024676481 -0.001716363 -0.057753790 -0.051626844
## 2002  0.043129791 -0.095549257 -0.160614806  0.002642338
## 2003 -0.111051823  0.110341744  0.061772133 -0.033455653
## 2004  0.021886682 -0.087954009  0.015866920 -0.008146884
## 2005  0.078856559  0.046503732  0.063141167 -0.012130977
## 2006  0.021470595 -0.080531270  0.076012117  0.084566200
## 2007  0.060810335  0.021831503 -0.005299601 -0.083079214
## 2008 -0.062276489  0.036486143 -0.155764184 -0.295005199
## 2009  0.069901684  0.059954716  0.085869627 -0.061035628
## 2010  0.063601107 -0.024520294 -0.029677545  0.025430648
## 2011  0.002201346 -0.053216282 -0.138776377  0.070978941
## 2012  0.093411631 -0.061365562  0.121724023 -0.036979556
## 2013  0.055720661 -0.056090572  0.041492109  0.035910751
## 2014  0.132891013  0.020166375  0.001088729 -0.041007430
## 2015  0.176295142  0.045818457 -0.017682880  0.023383949
## 2016 -0.025641634  0.026319335 -0.025344448 -0.088150923
## 2017  0.099179932  0.052757523 -0.013038996 -0.018275563
## 2018 -0.040616499 -0.027220290  0.017732210 -0.115537464
## 2019  0.072896898  0.007092071 -0.011509781  0.022733507
```

```r
plot(model_LC20_diff_start$residuals)
```




```
#Sidste plot viser at der skal laves dummies for 2008Q3 - 2008Q4 og for 2015Q1.
dummy_LC20_diff_2002Q3=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(200

dummy_LC20_diff_2003Q1=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(200

dummy_LC20_diff_2008Q4=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(200

dummy_LC20_diff_2008Q3=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(200

dummy_LC20_diff_2011Q3=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(201

dummy_LC20_diff_2015=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2015,

dummies_LC20_diff = cbind(dummy_LC20_diff_2008Q4,dummy_LC20_diff_2008Q3,dummy_LC20_diff_2015, dummy_LC2
```

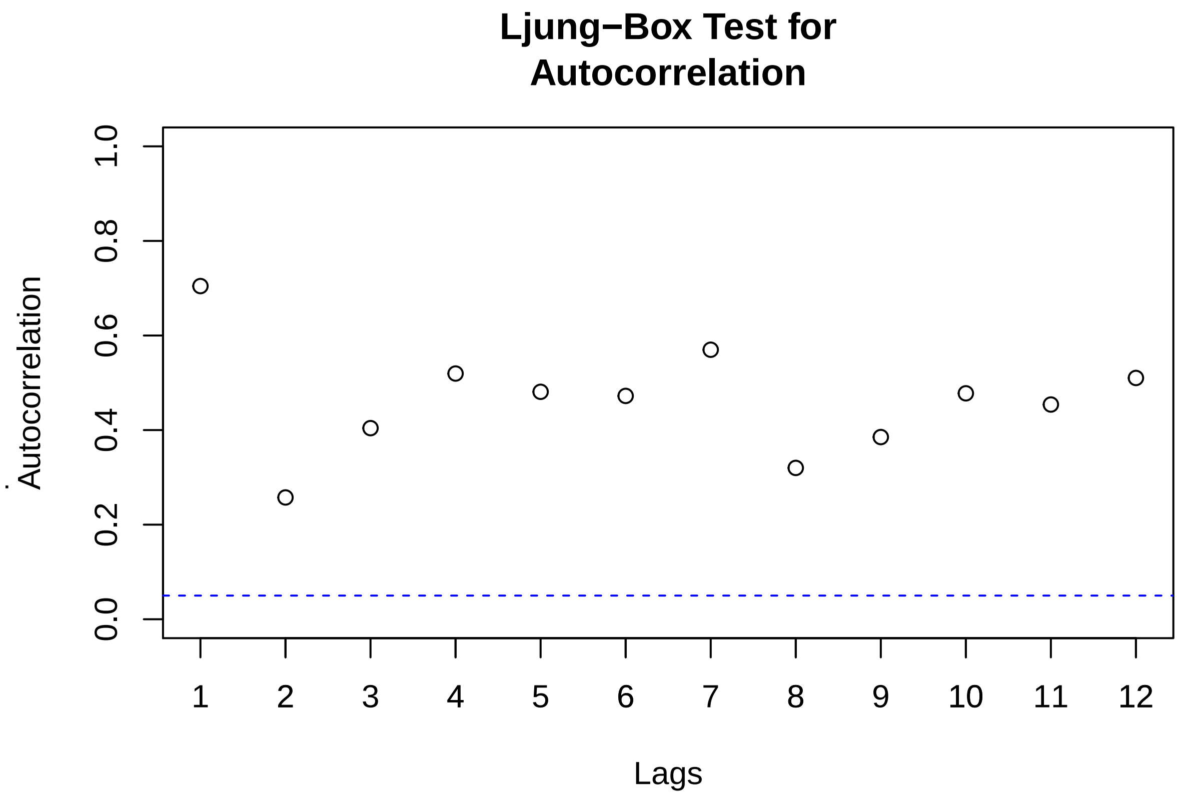
Ny model med de nye dummies

```
model_LC20_diff_start_2 <- dynlm(diff(LC20_diff) ~ 1+ L(LC20_diff, 1) + L(diff(LC20_diff), 1:12) + dummi
summary(model_LC20_diff_start_2)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(LC20_diff) ~ 1 + L(LC20_diff, 1) + L(diff(LC20_diff),
##     1:12) + dummies_LC20_diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12085 -0.02791  0.00000  0.03291  0.12210
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                               0.038365   0.008921   4.301 6.61e-05
## L(LC20_diff, 1)                          -1.463923   0.272280  -5.377 1.41e-06
## L(diff(LC20_diff), 1:12)1                 0.682755   0.254339   2.684  0.00946
## L(diff(LC20_diff), 1:12)2                 0.622433   0.243848   2.553  0.01335
## L(diff(LC20_diff), 1:12)3                 0.463132   0.230612   2.008  0.04928
## L(diff(LC20_diff), 1:12)4                 0.368549   0.214068   1.722  0.09046
## L(diff(LC20_diff), 1:12)5                 0.365038   0.195778   1.865  0.06731
## L(diff(LC20_diff), 1:12)6                 0.219616   0.183883   1.194  0.23721
## L(diff(LC20_diff), 1:12)7                 0.193547   0.165744   1.168  0.24769
## L(diff(LC20_diff), 1:12)8                 0.186039   0.150827   1.233  0.22238
## L(diff(LC20_diff), 1:12)9                 0.113406   0.141051   0.804  0.42468
## L(diff(LC20_diff), 1:12)10                0.082024   0.123626   0.663  0.50965
## L(diff(LC20_diff), 1:12)11                0.124065   0.102721   1.208  0.23203
## L(diff(LC20_diff), 1:12)12               -0.038654   0.088837  -0.435  0.66510
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.367697   0.063957  -5.749 3.52e-07
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.179909   0.062167  -2.894  0.00535
## dummies_LC20_diffdummy_LC20_diff_2015    0.173408   0.064175   2.702  0.00902
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.207062   0.063000  -3.287  0.00172
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.203051   0.071748  -2.830  0.00638
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.130034   0.065055  -1.999  0.05032
##
## (Intercept)                              ***
## L(LC20_diff, 1)                          ***
## L(diff(LC20_diff), 1:12)1                **
## L(diff(LC20_diff), 1:12)2                *
## L(diff(LC20_diff), 1:12)3                *
## L(diff(LC20_diff), 1:12)4                .
## L(diff(LC20_diff), 1:12)5                .
## L(diff(LC20_diff), 1:12)6
## L(diff(LC20_diff), 1:12)7
## L(diff(LC20_diff), 1:12)8
## L(diff(LC20_diff), 1:12)9
## L(diff(LC20_diff), 1:12)10
## L(diff(LC20_diff), 1:12)11
## L(diff(LC20_diff), 1:12)12
## dummies_LC20_diffdummy_LC20_diff_2008Q4 ***
## dummies_LC20_diffdummy_LC20_diff_2008Q3 **
## dummies_LC20_diffdummy_LC20_diff_2015   **
## dummies_LC20_diffdummy_LC20_diff_2002Q3 **
## dummies_LC20_diffdummy_LC20_diff_2011Q3 **
## dummies_LC20_diffdummy_LC20_diff_2003Q1 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05912 on 58 degrees of freedom
## Multiple R-squared:  0.7123, Adjusted R-squared:  0.618
## F-statistic: 7.556 on 19 and 58 DF,  p-value: 1.033e-09
```

```r
res_model_LC20_diff_start_2= model_LC20_diff_start_2$residuals

#Ljungbox test
arpdiag(res_model_LC20_diff_start_2, 12)
```




```r
#Normality
shapiro.test(res_model_LC20_diff_start_2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_LC20_diff_start_2
## W = 0.98794, p-value = 0.6769
```

```r
jarque.bera.test(res_model_LC20_diff_start_2)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_LC20_diff_start_2
## X-squared = 0.024042, df = 2, p-value = 0.9881
```

Vi kan nu fjerne lags med F-test

```r
LC20_diff_vars_1 <- str_c("L(diff(LC20_diff), 1:12)", c(1:12))

linearHypothesis(model_LC20_diff_start_2,LC20_diff_vars_1, rep(0, length(LC20_diff_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(LC20_diff),12)1 = 0
## L(diff(LC20_diff),12)2 = 0
## L(diff(LC20_diff),12)3 = 0
## L(diff(LC20_diff),12)4 = 0
## L(diff(LC20_diff),12)5 = 0
```

```
## L(diff(LC20_diff),12)6 = 0
## L(diff(LC20_diff),12)7 = 0
## L(diff(LC20_diff),12)8 = 0
## L(diff(LC20_diff),12)9 = 0
## L(diff(LC20_diff),12)10 = 0
## L(diff(LC20_diff),12)11 = 0
## L(diff(LC20_diff),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(LC20_diff) ~ 1 + L(LC20_diff, 1) + L(diff(LC20_diff), 1:12) +
##     dummies_LC20_diff
##
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     70 0.24920
## 2     58 0.20272 12  0.046478 1.1081 0.3714
```

```r
model_LC20_diff <- dynlm(diff(LC20_diff) ~ 1+ L(LC20_diff, 1) + dummies_LC20_diff)
summary(model_LC20_diff)
```

```
##
## Time series regression with "ts" data:
## Start = 1997(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(LC20_diff) ~ 1 + L(LC20_diff, 1) + dummies_LC20_diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12239 -0.04302  0.00000  0.03742  0.13308
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              0.025672   0.006925   3.707 0.000380
## L(LC20_diff, 1)                         -0.752905   0.078925  -9.539 6.10e-15
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.342087   0.062469  -5.476 4.65e-07
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.176300   0.060972  -2.891 0.004907
## dummies_LC20_diffdummy_LC20_diff_2015    0.127334   0.060998   2.088 0.039947
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.181759   0.061428  -2.959 0.004034
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.219336   0.061235  -3.582 0.000577
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.104703   0.061144  -1.712 0.090602
##
## (Intercept)                             ***
## L(LC20_diff, 1)                         ***
## dummies_LC20_diffdummy_LC20_diff_2008Q4 ***
## dummies_LC20_diffdummy_LC20_diff_2008Q3 **
## dummies_LC20_diffdummy_LC20_diff_2015   *
## dummies_LC20_diffdummy_LC20_diff_2002Q3 **
## dummies_LC20_diffdummy_LC20_diff_2011Q3 ***
## dummies_LC20_diffdummy_LC20_diff_2003Q1 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06061 on 82 degrees of freedom
## Multiple R-squared:  0.6087, Adjusted R-squared:  0.5753
## F-statistic: 18.22 on 7 and 82 DF,  p-value: 2.12e-14
```

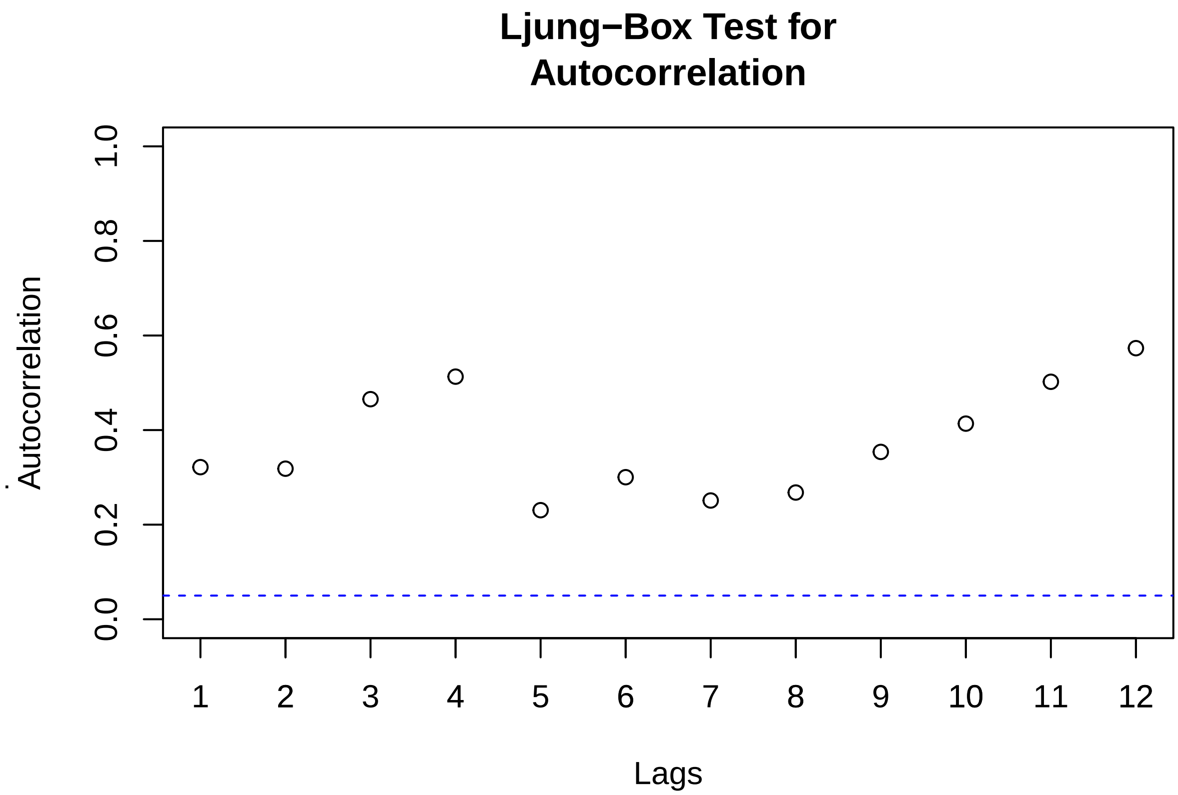
```r
model_LC20_ur_diff <- ur.df(LC20_diff, lags = 0, type = "drift")
summary(model_LC20_ur_diff)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.307717 -0.048031  0.003448  0.050532  0.143947
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.012615   0.008479   1.488     0.14
## z.lag.1     -0.607365   0.097808  -6.210 1.71e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07799 on 88 degrees of freedom
## Multiple R-squared:  0.3047, Adjusted R-squared:  0.2968
## F-statistic: 38.56 on 1 and 88 DF,  p-value: 1.707e-08
##
##
## Value of test-statistic is: -6.2098 19.2812
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

```r
res_model_LC20_diff= residuals(model_LC20_diff)
```

Vi tjekker for Serial correlation

```r
arpdiag(res_model_LC20_diff, 12)
```



C20

indekset er altså I(1).

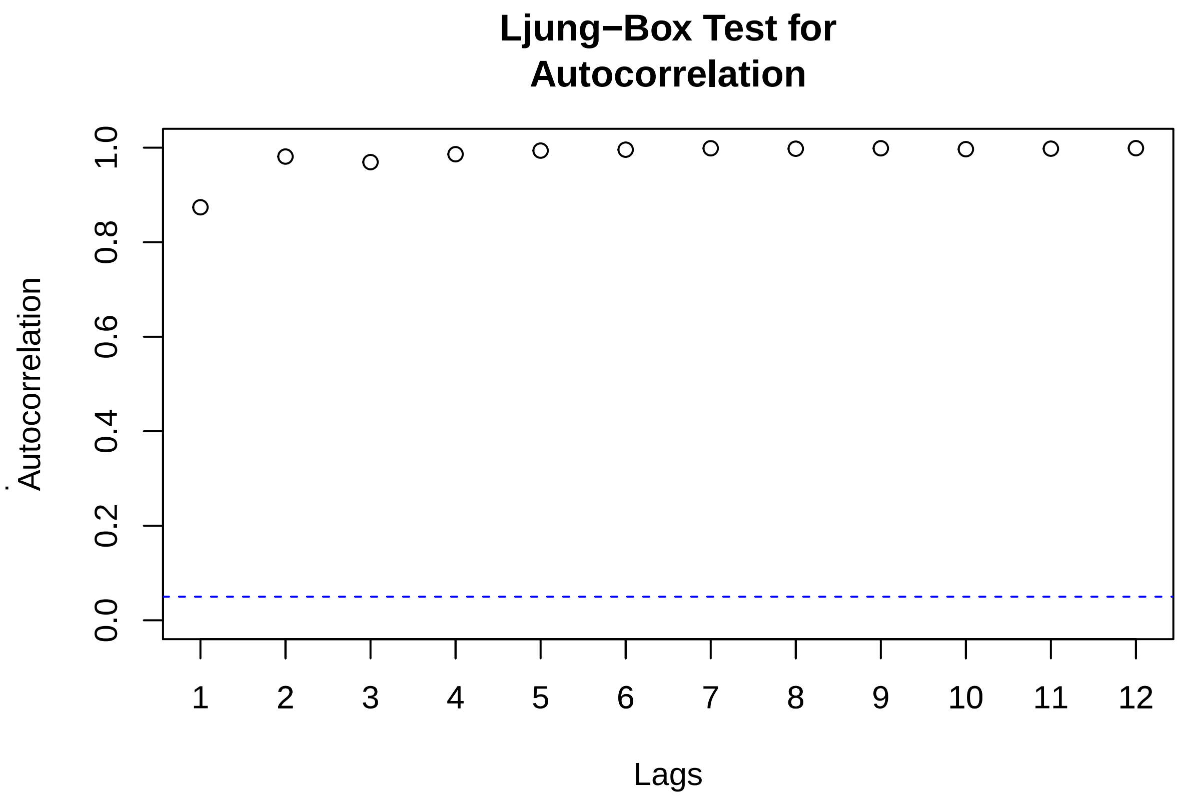
2.4 ADF Unemployment

```
model_u_start <- dynlm(diff(u) ~ 1+ L(u, 1) + L(diff(u), 1:12) + trend(diff(u)))
summary(model_u_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(u) ~ 1 + L(u, 1) + L(diff(u), 1:12) + trend(diff(u)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74978 -0.15406  0.00223  0.14055  0.56326
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.400480   0.177701   2.254  0.02765 *
## L(u, 1)             -0.086321   0.040695  -2.121  0.03779 *
## L(diff(u), 1:12)1    0.308545   0.115719   2.666  0.00970 **
## L(diff(u), 1:12)2    0.254136   0.121522   2.091  0.04048 *
## L(diff(u), 1:12)3    0.396537   0.123624   3.208  0.00209 **
## L(diff(u), 1:12)4   -0.292692   0.134044  -2.184  0.03267 *
## L(diff(u), 1:12)5   -0.030627   0.139482  -0.220  0.82690
## L(diff(u), 1:12)6    0.101808   0.139000   0.732  0.46658
## L(diff(u), 1:12)7    0.172463   0.139532   1.236  0.22097
```

```
## L(diff(u), 1:12)8   0.003235   0.139895   0.023  0.98162
## L(diff(u), 1:12)9   0.084922   0.131885   0.644  0.52193
## L(diff(u), 1:12)10 -0.068543   0.127285  -0.538  0.59210
## L(diff(u), 1:12)11  0.032103   0.126302   0.254  0.80017
## L(diff(u), 1:12)12  0.041029   0.125087   0.328  0.74398
## trend(diff(u))      0.007022   0.007190   0.977  0.33245
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2543 on 64 degrees of freedom
## Multiple R-squared:  0.4123, Adjusted R-squared:  0.2837
## F-statistic: 3.207 on 14 and 64 DF,  p-value: 0.0007364
```

```r
arpdiag(residuals(model_u_start), 12)
```



Test for normality

```r
shapiro.test(residuals(model_u_start))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_u_start)
## W = 0.98759, p-value = 0.6457
```

```r
jarque.bera.test(residuals(model_u_start))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model_u_start)
```

```
## X-squared = 1.6929, df = 2, p-value = 0.4289
```

Der er ingen problemer med normality i start modellen.

Vi fjerner lags med F-test

```r
u_vars_1 <- str_c("L(diff(u), 1:12)", 5:12)

linearHypothesis(model_u_start,u_vars_1, rep(0, length(u_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(u),12)5 = 0
## L(diff(u),12)6 = 0
## L(diff(u),12)7 = 0
## L(diff(u),12)8 = 0
## L(diff(u),12)9 = 0
## L(diff(u),12)10 = 0
## L(diff(u),12)11 = 0
## L(diff(u),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(u) ~ 1 + L(u, 1) + L(diff(u), 1:12) + trend(diff(u))
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     72 4.4508
## 2     64 4.1389  8   0.31191 0.6029  0.772
```

F-testen er klart bestået

```r
model_u <- dynlm(diff(u) ~ L(u, 1) + L(diff(u), 1:4))
summary(model_u)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(u) ~ L(u, 1) + L(diff(u), 1:4))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73175 -0.13604 -0.00485  0.15396  0.67429
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.24077    0.12681   1.899  0.06118 .
## L(u, 1)           -0.04292    0.02216  -1.937  0.05627 .
## L(diff(u), 1:4)1  0.29051    0.10324   2.814  0.00614 **
## L(diff(u), 1:4)2  0.17710    0.10122   1.750  0.08397 .
## L(diff(u), 1:4)3  0.40028    0.10310   3.882  0.00021 ***
## L(diff(u), 1:4)4 -0.26886    0.10868  -2.474  0.01545 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2481 on 81 degrees of freedom
```

```
## Multiple R-squared:  0.3392, Adjusted R-squared:  0.2985
## F-statistic: 8.318 on 5 and 81 DF,  p-value: 2.212e-06
```

```
model_u2= ur.df(u, lags = 4, type = "drift")
summary(model_u2)
```

```
##
## ###############################################
## # 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
## -0.73175 -0.13604 -0.00485  0.15396  0.67429
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.24077    0.12681   1.899  0.06118 .
## z.lag.1     -0.04292    0.02216  -1.937  0.05627 .
## z.diff.lag1  0.29051    0.10324   2.814  0.00614 **
## z.diff.lag2  0.17710    0.10122   1.750  0.08397 .
## z.diff.lag3  0.40028    0.10310   3.882  0.00021 ***
## z.diff.lag4 -0.26886    0.10868  -2.474  0.01545 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2481 on 81 degrees of freedom
## Multiple R-squared:  0.3392, Adjusted R-squared:  0.2985
## F-statistic: 8.318 on 5 and 81 DF,  p-value: 2.212e-06
##
##
## Value of test-statistic is: -1.9367 1.8756
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

Der er altså unitroot i dataet for unemployment, så der bliver nu taget diff.

Vi skal tage Diff af guld

```
u_diff=diff(u)
```

Vi refitter vores model igen

```
model_u_diff_start <- dynlm(diff(u_diff) ~ 0+ L(u_diff, 1) + L(diff(u_diff), 1:12))
summary(model_u_diff_start)
```

```
##
## Time series regression with "ts" data:
```

```
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(u_diff) ~ 0 + L(u_diff, 1) + L(diff(u_diff),
##     1:12))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67871 -0.16303 -0.02551  0.15241  0.65902
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## L(u_diff, 1)            -0.65767    0.21437  -3.068  0.00314 **
## L(diff(u_diff), 1:12)1  -0.07613    0.21208  -0.359  0.72077
## L(diff(u_diff), 1:12)2   0.15162    0.20479   0.740  0.46175
## L(diff(u_diff), 1:12)3   0.50417    0.19671   2.563  0.01270 *
## L(diff(u_diff), 1:12)4   0.15928    0.19863   0.802  0.42554
## L(diff(u_diff), 1:12)5   0.07553    0.19227   0.393  0.69575
## L(diff(u_diff), 1:12)6   0.15192    0.18673   0.814  0.41885
## L(diff(u_diff), 1:12)7   0.28504    0.17688   1.611  0.11192
## L(diff(u_diff), 1:12)8   0.21696    0.16524   1.313  0.19380
## L(diff(u_diff), 1:12)9   0.21777    0.15549   1.401  0.16611
## L(diff(u_diff), 1:12)10  0.16549    0.15631   1.059  0.29364
## L(diff(u_diff), 1:12)11  0.17974    0.14565   1.234  0.22161
## L(diff(u_diff), 1:12)12  0.18481    0.12024   1.537  0.12916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2567 on 65 degrees of freedom
## Multiple R-squared:  0.5235, Adjusted R-squared:  0.4282
## F-statistic: 5.494 on 13 and 65 DF,  p-value: 1.379e-06
```

```
res_model_u_diff_start= model_u_diff_start$residuals
```

Vi tester for Normality

```
shapiro.test(res_model_u_diff_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_u_diff_start
## W = 0.98184, p-value = 0.331
```

```
jarque.bera.test(res_model_u_diff_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_u_diff_start
## X-squared = 1.8917, df = 2, p-value = 0.3883
```

Der er ikke problemer med normality

Vi kan nu fjerne lags med F-test

```r
u_diff_vars_1 <- str_c("L(diff(u_diff), 1:12)", 4:12)

linearHypothesis(model_u_diff_start,u_diff_vars_1, rep(0, length(u_diff_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(u_diff),12)4 = 0
## L(diff(u_diff),12)5 = 0
## L(diff(u_diff),12)6 = 0
## L(diff(u_diff),12)7 = 0
## L(diff(u_diff),12)8 = 0
## L(diff(u_diff),12)9 = 0
## L(diff(u_diff),12)10 = 0
## L(diff(u_diff),12)11 = 0
## L(diff(u_diff),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(u_diff) ~ 0 + L(u_diff, 1) + L(diff(u_diff), 1:12)
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     74 4.6889
## 2     65 4.2839  9   0.40501 0.6828 0.7216
```

F-testen er ikke signifikant, så vi kan fjerne lags

```r
model_u_diff <- dynlm(diff(u_diff) ~ 0+L(u_diff, 1) + L(diff(u_diff), 1:3))
summary(model_u_diff)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(u_diff) ~ 0 + L(u_diff, 1) + L(diff(u_diff),
##     1:3))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67162 -0.13589 -0.01095  0.15507  0.70688
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## L(u_diff, 1)            -0.48783    0.12844  -3.798 0.000277 ***
## L(diff(u_diff), 1:3)1 -0.20642    0.13918  -1.483 0.141827
## L(diff(u_diff), 1:3)2 -0.04359    0.12915  -0.338 0.736575
## L(diff(u_diff), 1:3)3  0.32904    0.10523   3.127 0.002436 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2507 on 83 degrees of freedom
## Multiple R-squared:  0.4727, Adjusted R-squared:  0.4473
## F-statistic:  18.6 on 4 and 83 DF,  p-value: 6.021e-11
```

We remove lag 2.

```r
model_u_diff_reduced <- dynlm(diff(u_diff) ~ 0+L(u_diff, 1) + L(diff(u_diff), c(1,3)))
summary(model_u_diff_reduced)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(u_diff) ~ 0 + L(u_diff, 1) + L(diff(u_diff),
##     c(1, 3)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66958 -0.13044 -0.00332  0.15208  0.70233
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## L(u_diff, 1)              -0.50943    0.11077  -4.599 1.49e-05 ***
## L(diff(u_diff), c(1, 3))1 -0.17291    0.09702  -1.782   0.0783 .
## L(diff(u_diff), c(1, 3))3  0.35139    0.08133   4.320 4.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2493 on 84 degrees of freedom
## Multiple R-squared:  0.472,  Adjusted R-squared:  0.4531
## F-statistic: 25.03 on 3 and 84 DF,  p-value: 1.153e-11
```

```r
model_u_diff1 <- ur.df(u_diff, lags = 2, type = "none")
summary(model_u_diff1)
```

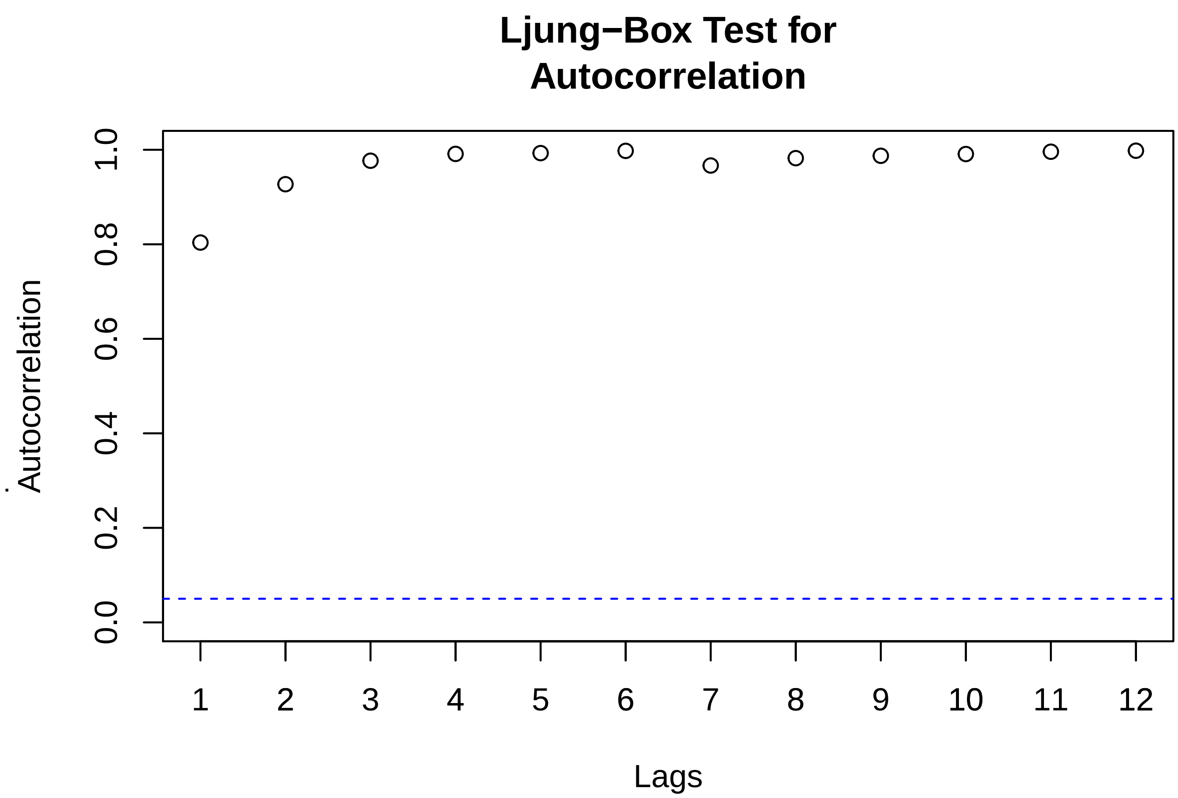
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67559 -0.16025 -0.02572  0.16334  0.75004
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1       -0.3691     0.1281  -2.881  0.00501 **
## z.diff.lag1  -0.4189     0.1269  -3.301  0.00141 **
## z.diff.lag2  -0.2991     0.1045  -2.862  0.00530 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2619 on 85 degrees of freedom
```

```
## Multiple R-squared:  0.4107, Adjusted R-squared:  0.3899
## F-statistic: 19.74 on 3 and 85 DF,  p-value: 8.402e-10
##
##
## Value of test-statistic is: -2.8812
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```

```
res_model_u_diff_reduced= residuals(model_u_diff_reduced)
```

We reject that Unemployment is I(2) so we can accept H_1 and conclude that Unempoyment is I(1)

```
arpdiag(res_model_u_diff_reduced, lag=12)
```




```
Box.test(res_model_u_diff_reduced,12)
```

```
##
##  Box-Pierce test
##
## data:  res_model_u_diff_reduced
## X-squared = 2.2713, df = 12, p-value = 0.9989
```

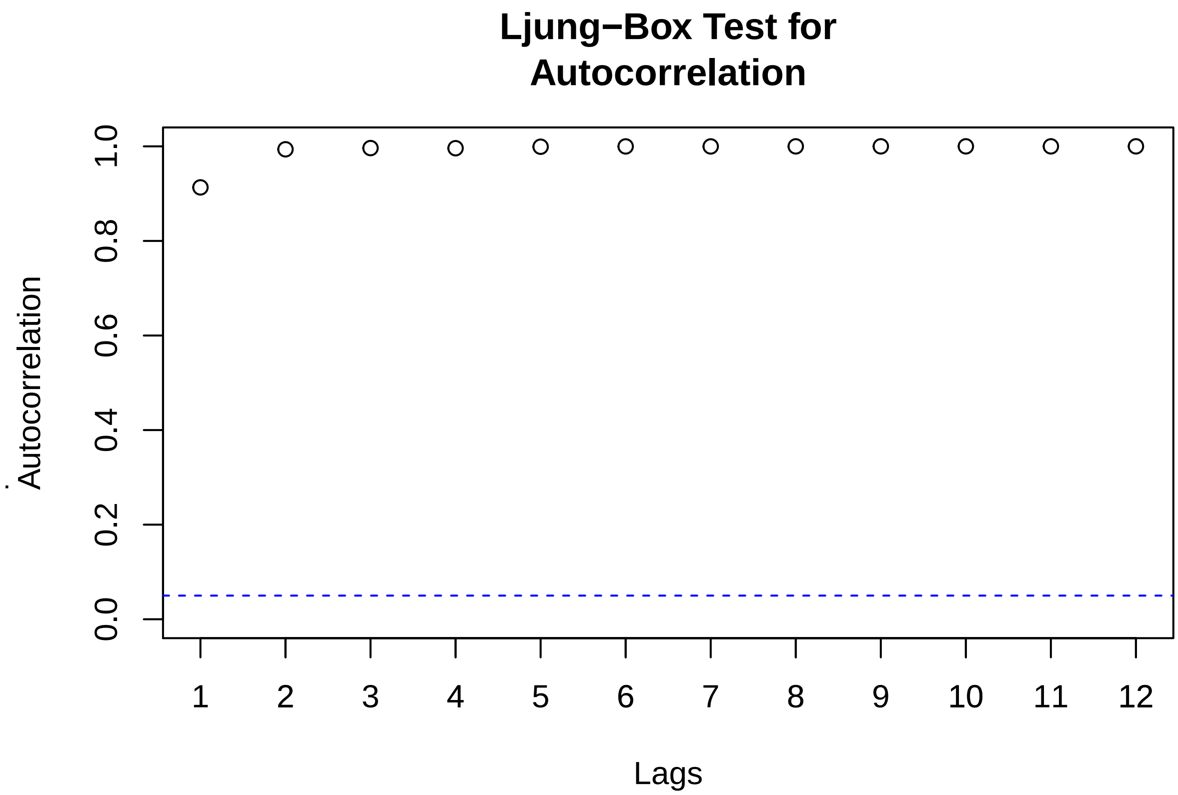
Der er altså ingen seriekorrelation.

2.5 ADF inf

```
model_inflation_start <- dynlm(diff(inf) ~ 1+ L(inf, 1) + L(diff(inf), 1:12) + trend(diff(inf)))
summary(model_inflation_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(inf) ~ 1 + L(inf, 1) + L(diff(inf), 1:12) +
##     trend(diff(inf)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74161 -0.23605 -0.04631  0.24055  0.95229
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.56218    0.30282   1.856  0.06799 .
## L(inf, 1)            -0.20196    0.09684  -2.086  0.04101 *
## L(diff(inf), 1:12)1   0.34077    0.13187   2.584  0.01205 *
## L(diff(inf), 1:12)2   0.11791    0.13774   0.856  0.39518
## L(diff(inf), 1:12)3   0.22883    0.13779   1.661  0.10166
## L(diff(inf), 1:12)4  -0.47964    0.13983  -3.430  0.00106 **
## L(diff(inf), 1:12)5   0.06383    0.13902   0.459  0.64769
## L(diff(inf), 1:12)6   0.22624    0.13909   1.627  0.10874
## L(diff(inf), 1:12)7   0.13099    0.14236   0.920  0.36097
## L(diff(inf), 1:12)8  -0.27918    0.14244  -1.960  0.05435 .
## L(diff(inf), 1:12)9   0.01896    0.12417   0.153  0.87913
## L(diff(inf), 1:12)10  0.10215    0.12721   0.803  0.42494
## L(diff(inf), 1:12)11  0.25213    0.12720   1.982  0.05177 .
## L(diff(inf), 1:12)12 -0.14976    0.12954  -1.156  0.25195
## trend(diff(inf))     -0.01847    0.01126  -1.641  0.10576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.354 on 64 degrees of freedom
## Multiple R-squared:  0.4192, Adjusted R-squared:  0.2922
## F-statistic:   3.3 on 14 and 64 DF,  p-value: 0.0005521
```

```r
arpdiag(residuals(model_inflation_start), 12)
```



Ingen seriekorrelation

Test for normality

```
shapiro.test(residuals(model_inflation_start))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_inflation_start)
## W = 0.98416, p-value = 0.436
```

```
jarque.bera.test(residuals(model_inflation_start))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model_inflation_start)
## X-squared = 2.1977, df = 2, p-value = 0.3333
```

Der er ingen problemer med normality i start modellen.

Vi fjerner lags med F-test

```
inf_vars_1 <- str_c("L(diff(inf), 1:12)", c(2,3,5,6,7,9,10,11,12))

linearHypothesis(model_inflation_start,inf_vars_1, rep(0, length(inf_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(inf),12)2 = 0
## L(diff(inf),12)3 = 0
```

```
## L(diff(inf),12)5 = 0
## L(diff(inf),12)6 = 0
## L(diff(inf),12)7 = 0
## L(diff(inf),12)9 = 0
## L(diff(inf),12)10 = 0
## L(diff(inf),12)11 = 0
## L(diff(inf),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(inf) ~ 1 + L(inf, 1) + L(diff(inf), 1:12) + trend(diff(inf))
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     73 9.1620
## 2     64 8.0182  9    1.1438 1.0144 0.4382
```

F-testen er bestået

```
model_inf <- dynlm(diff(inf) ~ 1+ L(inf, 1) + L(diff(inf), c(1,4,8))+ trend(diff(inf)))
summary(model_inf)
```

```
##
## Time series regression with "ts" data:
## Start = 1999(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(inf) ~ 1 + L(inf, 1) + L(diff(inf), c(1,
##     4, 8)) + trend(diff(inf)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82652 -0.24177 -0.03947  0.24335  0.95923
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.322468   0.203546   1.584 0.117235
## L(inf, 1)                 -0.106124   0.063143  -1.681 0.096878 .
## L(diff(inf), c(1, 4, 8))1  0.253376   0.098480   2.573 0.012009 *
## L(diff(inf), c(1, 4, 8))4 -0.483208   0.118296  -4.085 0.000107 ***
## L(diff(inf), c(1, 4, 8))8 -0.258223   0.108721  -2.375 0.020034 *
## trend(diff(inf))          -0.013014   0.008676  -1.500 0.137712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3478 on 77 degrees of freedom
## Multiple R-squared:  0.3455, Adjusted R-squared:  0.303
## F-statistic: 8.128 on 5 and 77 DF,  p-value: 3.369e-06
```

```
model_inf2= ur.df(inf, lags = 8, type = "trend")
summary(model_inf2)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
```

```
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71840 -0.24386 -0.01883  0.22096  0.94186
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.528976   0.254292   2.080 0.041068 *
## z.lag.1     -0.178370   0.082885  -2.152 0.034749 *
## tt          -0.004763   0.002445  -1.948 0.055348 .
## z.diff.lag1  0.325334   0.119839   2.715 0.008296 **
## z.diff.lag2  0.049896   0.124934   0.399 0.690794
## z.diff.lag3  0.167680   0.125307   1.338 0.185057
## z.diff.lag4 -0.458431   0.126138  -3.634 0.000519 ***
## z.diff.lag5  0.037404   0.117673   0.318 0.751509
## z.diff.lag6  0.118746   0.117831   1.008 0.316942
## z.diff.lag7  0.013197   0.117889   0.112 0.911182
## z.diff.lag8 -0.204626   0.117268  -1.745 0.085263 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3518 on 72 degrees of freedom
## Multiple R-squared:  0.3736, Adjusted R-squared:  0.2866
## F-statistic: 4.295 on 10 and 72 DF,  p-value: 0.0001044
##
##
## Value of test-statistic is: -2.152 1.7619 2.4459
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Der er altså unitroot i dataet for inflation, så der bliver nu taget diff.

Vi skal tage Diff af inflation

```r
inf_diff=diff(inf)
```

Vi refitter vores model igen

```r
model_inf_diff_start <- dynlm(diff(inf_diff) ~ 0+ L(inf_diff, 1) + L(diff(inf_diff), 1:12))
summary(model_inf_diff_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(inf_diff) ~ 0 + L(inf_diff, 1) + L(diff(inf_diff),
##     1:12))
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91290 -0.25312 -0.04441  0.13831  0.99355
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## L(inf_diff, 1)           -1.46268    0.51049  -2.865  0.00561 **
## L(diff(inf_diff), 1:12)1  0.69486    0.47926   1.450  0.15191
## L(diff(inf_diff), 1:12)2  0.66363    0.46103   1.439  0.15482
## L(diff(inf_diff), 1:12)3  0.74671    0.43407   1.720  0.09014 .
## L(diff(inf_diff), 1:12)4  0.11871    0.40401   0.294  0.76983
## L(diff(inf_diff), 1:12)5  0.11528    0.35834   0.322  0.74871
## L(diff(inf_diff), 1:12)6  0.25946    0.33070   0.785  0.43556
## L(diff(inf_diff), 1:12)7  0.28996    0.29904   0.970  0.33583
## L(diff(inf_diff), 1:12)8 -0.08424    0.25528  -0.330  0.74246
## L(diff(inf_diff), 1:12)9 -0.08269    0.20154  -0.410  0.68294
## L(diff(inf_diff), 1:12)10 -0.01688    0.18167  -0.093  0.92626
## L(diff(inf_diff), 1:12)11  0.19063    0.15700   1.214  0.22907
## L(diff(inf_diff), 1:12)12 -0.03955    0.12980  -0.305  0.76159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3647 on 65 degrees of freedom
## Multiple R-squared:  0.5983, Adjusted R-squared:  0.518
## F-statistic: 7.447 on 13 and 65 DF,  p-value: 1.074e-08
```

```r
res_model_inf_diff_start= model_inf_diff_start$residuals
```

Vi tester for Normality

```r
shapiro.test(res_model_inf_diff_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_inf_diff_start
## W = 0.98631, p-value = 0.572
```

```r
jarque.bera.test(res_model_inf_diff_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_inf_diff_start
## X-squared = 2.7541, df = 2, p-value = 0.2523
```

Der er ikke problemer med normality

Vi kan nu fjerne lags med F-test

```r
inf_diff_vars_1 <- str_c("L(diff(inf_diff), 1:12)", c(4:12))

linearHypothesis(model_inf_diff_start,inf_diff_vars_1, rep(0, length(inf_diff_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
```

```
## L(diff(inf_diff),12)4 = 0
## L(diff(inf_diff),12)5 = 0
## L(diff(inf_diff),12)6 = 0
## L(diff(inf_diff),12)7 = 0
## L(diff(inf_diff),12)8 = 0
## L(diff(inf_diff),12)9 = 0
## L(diff(inf_diff),12)10 = 0
## L(diff(inf_diff),12)11 = 0
## L(diff(inf_diff),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(inf_diff) ~ 0 + L(inf_diff, 1) + L(diff(inf_diff), 1:12)
##
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     74 10.2063
## 2     65  8.6447  9    1.5616 1.3047 0.2519
```

F-testen er ikke signifikant, så vi kan fjerne lagsne

```r
model_inf_diff <- dynlm(diff(inf_diff) ~ 0+L(inf_diff, 1) + L(diff(inf_diff), 1:3))
summary(model_inf_diff)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(inf_diff) ~ 0 + L(inf_diff, 1) + L(diff(inf_diff),
##     1:3))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98602 -0.28796 -0.02525  0.21119  1.01454
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## L(inf_diff, 1)          -1.20888    0.16385  -7.378 1.11e-10 ***
## L(diff(inf_diff), 1:3)1  0.46293    0.14433   3.207  0.00190 **
## L(diff(inf_diff), 1:3)2  0.35705    0.12152   2.938  0.00427 **
## L(diff(inf_diff), 1:3)3  0.47603    0.09808   4.853 5.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3563 on 83 degrees of freedom
## Multiple R-squared:  0.5187, Adjusted R-squared:  0.4955
## F-statistic: 22.36 on 4 and 83 DF,  p-value: 1.487e-12
```

```r
model_inf_diff1 <- ur.df(inf_diff, lags = 3, type = "none")
summary(model_inf_diff1)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
```

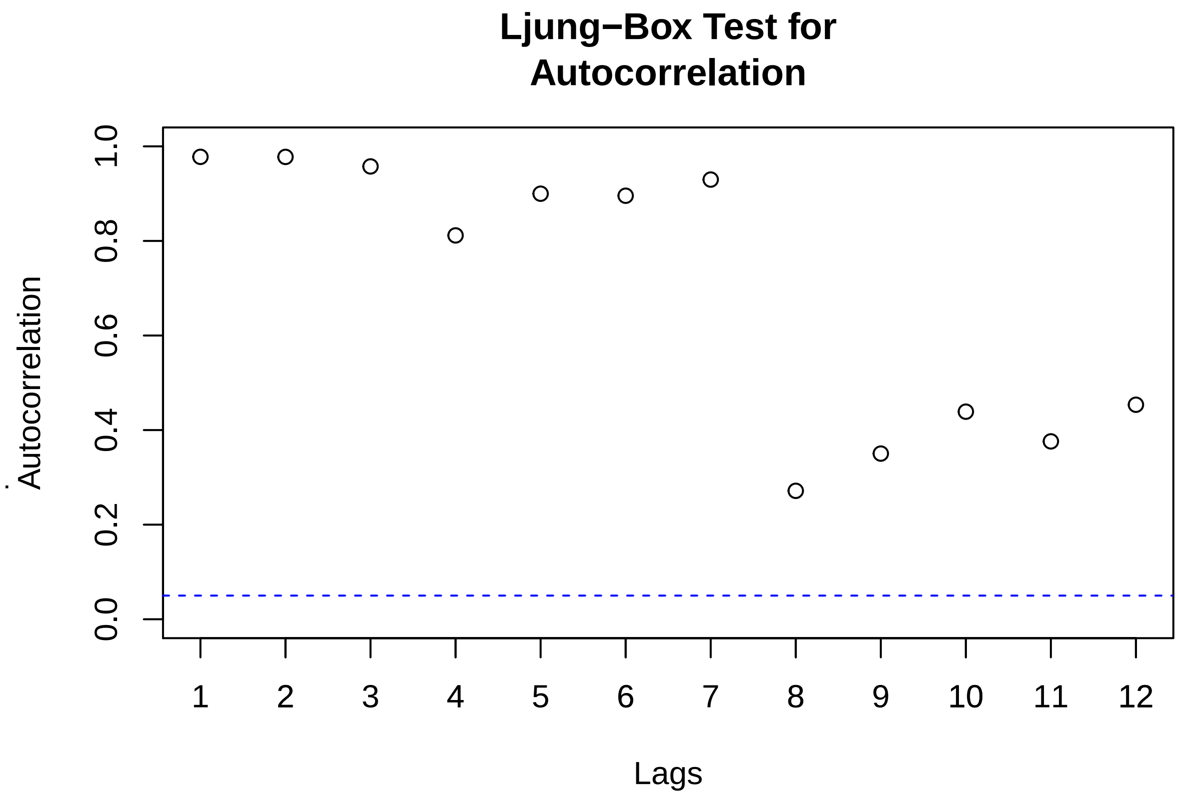
```
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98602 -0.28796 -0.02525  0.21119  1.01454
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.20888    0.16385  -7.378 1.11e-10 ***
## z.diff.lag1  0.46293    0.14433   3.207  0.00190 **
## z.diff.lag2  0.35705    0.12152   2.938  0.00427 **
## z.diff.lag3  0.47603    0.09808   4.853 5.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3563 on 83 degrees of freedom
## Multiple R-squared:  0.5187, Adjusted R-squared:  0.4955
## F-statistic: 22.36 on 4 and 83 DF,  p-value: 1.487e-12
##
##
## Value of test-statistic is: -7.378
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```

```r
res_model_inf_diff= residuals(model_inf_diff)
```

Der er altså ikke længere unitroot og inflation er I(1)

Tester for seriekorrelation

```r
arpdiag(res_model_inf_diff, lag=12)
```



Der er altså ingen seriekorrelation.

2.6 ADF Rate of Savings

```
model_sav_start <- dynlm(diff(sav) ~ 1+ L(sav, 1) + L(diff(sav), 1:12) + trend(diff(sav)))
summary(model_sav_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(sav) ~ 1 + L(sav, 1) + L(diff(sav), 1:12) +
##     trend(diff(sav)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6706 -0.5305  0.1353  0.4703  2.2579
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.2899365  2.1974922   2.407   0.0190 *
## L(sav, 1)            -0.3288502  0.1460389  -2.252   0.0278 *
## L(diff(sav), 1:12)1   0.0143054  0.1601595   0.089   0.9291
## L(diff(sav), 1:12)2   0.2309553  0.1530228   1.509   0.1361
## L(diff(sav), 1:12)3   0.1209383  0.1539007   0.786   0.4349
## L(diff(sav), 1:12)4   0.0991502  0.1568920   0.632   0.5297
## L(diff(sav), 1:12)5  -0.0575451  0.1532497  -0.375   0.7085
## L(diff(sav), 1:12)6   0.0710776  0.1462651   0.486   0.6287
```

```
## L(diff(sav), 1:12)7  -0.0063959  0.1428841  -0.045   0.9644
## L(diff(sav), 1:12)8   0.0607923  0.1344237   0.452   0.6526
## L(diff(sav), 1:12)9   0.2204181  0.1308677   1.684   0.0970 .
## L(diff(sav), 1:12)10  0.2135175  0.1338259   1.595   0.1155
## L(diff(sav), 1:12)11  0.0004622  0.1360113   0.003   0.9973
## L(diff(sav), 1:12)12 -0.0868393  0.1263920  -0.687   0.4945
## trend(diff(sav))      0.2490465  0.1104292   2.255   0.0275 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.948 on 64 degrees of freedom
## Multiple R-squared:  0.2511, Adjusted R-squared:  0.08722
## F-statistic: 1.532 on 14 and 64 DF,  p-value: 0.1251
```

```r
res_model_sav_start= model_sav_start$residuals
```

Test for normality

```r
shapiro.test(res_model_sav_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_sav_start
## W = 0.98208, p-value = 0.3331
```

```r
jarque.bera.test(res_model_sav_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_sav_start
## X-squared = 2.8261, df = 2, p-value = 0.2434
```

Vi fjerner lags med F-test

```r
sav_vars_1 <- str_c("L(diff(sav), 1:12)", 3:12)

linearHypothesis(model_sav_start,sav_vars_1, rep(0, length(sav_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(sav),12)3 = 0
## L(diff(sav),12)4 = 0
## L(diff(sav),12)5 = 0
## L(diff(sav),12)6 = 0
## L(diff(sav),12)7 = 0
## L(diff(sav),12)8 = 0
## L(diff(sav),12)9 = 0
## L(diff(sav),12)10 = 0
## L(diff(sav),12)11 = 0
## L(diff(sav),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(sav) ~ 1 + L(sav, 1) + L(diff(sav), 1:12) + trend(diff(sav))
##
```

```
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     74 64.562
## 2     64 57.511 10    7.0508 0.7846 0.6432
```

```
model_sav_reduced <- dynlm(diff(sav) ~ 1+ L(sav, 1) + L(diff(sav), 1:2) + trend(diff(sav)))
summary(model_sav_reduced)
```

```
##
## Time series regression with "ts" data:
## Start = 1997(4), End = 2019(4)
##
## Call:
## dynlm(formula = diff(sav) ~ 1 + L(sav, 1) + L(diff(sav), 1:2) +
##     trend(diff(sav)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1693 -0.5208  0.0472  0.5751  2.1472
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.71444    1.31737   3.579 0.000577 ***
## L(sav, 1)           -0.28935    0.08275  -3.497 0.000755 ***
## L(diff(sav), 1:2)1 -0.02284    0.11413  -0.200 0.841864
## L(diff(sav), 1:2)2  0.19200    0.10558   1.819 0.072535 .
## trend(diff(sav))     0.22456    0.06456   3.478 0.000802 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9077 on 84 degrees of freedom
## Multiple R-squared:  0.1748, Adjusted R-squared:  0.1355
## F-statistic: 4.448 on 4 and 84 DF,  p-value: 0.00261
```

We do an F-test again

```
sav_vars_2 <- str_c("L(diff(sav), 1:2)", c(1,2))

linearHypothesis(model_sav_reduced,sav_vars_2, rep(0, length(sav_vars_2)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(sav),2)1 = 0
## L(diff(sav),2)2 = 0
##
## Model 1: restricted model
## Model 2: diff(sav) ~ 1 + L(sav, 1) + L(diff(sav), 1:2) + trend(diff(sav))
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     86 72.589
## 2     84 69.213  2     3.376 2.0486 0.1353
```

```
model_sav_reduced <- dynlm(diff(sav) ~ 1+ L(sav, 1) + L(diff(sav), 2) + trend(diff(sav)))
summary(model_sav_reduced)
```

```
##
## Time series regression with "ts" data:
```

```
## Start = 1997(4), End = 2019(4)
##
## Call:
## dynlm(formula = diff(sav) ~ 1 + L(sav, 1) + L(diff(sav), 2) +
##     trend(diff(sav)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.13133 -0.56171 -0.01096  0.59068  2.14127
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.82823    1.18162   4.086  9.9e-05 ***
## L(sav, 1)         -0.29680    0.07350  -4.038 0.000118 ***
## L(diff(sav), 2)    0.19936    0.09841   2.026 0.045932 *
## trend(diff(sav))   0.23009    0.05802   3.965 0.000152 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9026 on 85 degrees of freedom
## Multiple R-squared:  0.1744, Adjusted R-squared:  0.1453
## F-statistic: 5.986 on 3 and 85 DF,  p-value: 0.0009456
```

```
model_sav= ur.df(sav, lags = 1, type = "trend")
summary(model_sav)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5919 -0.5948  0.0639  0.6025  2.1721
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.94713    1.26820   3.112  0.00252 **
## z.lag.1      -0.23651    0.07922  -2.986  0.00369 **
## tt            0.04593    0.01554   2.956  0.00402 **
## z.diff.lag   -0.11695    0.10609  -1.102  0.27339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9198 on 86 degrees of freedom
## Multiple R-squared:  0.1461, Adjusted R-squared:  0.1163
## F-statistic: 4.905 on 3 and 86 DF,  p-value: 0.003404
##
##
```

```
## Value of test-statistic is: -2.9856 5.1186 4.4865
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Vi skal tage Diff af bnp

```
sav_diff=diff(sav)
```

Vi refitter vores model igen

```
model_sav_diff_start <- dynlm(diff(sav_diff) ~ 1+ L(sav_diff, 1) + L(diff(sav_diff), 1:12))
summary(model_sav_diff_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(sav_diff) ~ 1 + L(sav_diff, 1) + L(diff(sav_diff),
##     1:12))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.80533 -0.58245  0.00586  0.58238  2.11751
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.408109   0.179861   2.269   0.0266 *
## L(sav_diff, 1)           -1.972647   0.710511  -2.776   0.0072 **
## L(diff(sav_diff), 1:12)1   0.755815   0.674409   1.121   0.2666
## L(diff(sav_diff), 1:12)2   0.784940   0.639527   1.227   0.2242
## L(diff(sav_diff), 1:12)3   0.688738   0.607489   1.134   0.2611
## L(diff(sav_diff), 1:12)4   0.553112   0.570530   0.969   0.3360
## L(diff(sav_diff), 1:12)5   0.293353   0.519483   0.565   0.5743
## L(diff(sav_diff), 1:12)6   0.204137   0.460870   0.443   0.6593
## L(diff(sav_diff), 1:12)7   0.049452   0.405935   0.122   0.9034
## L(diff(sav_diff), 1:12)8   0.002051   0.347029   0.006   0.9953
## L(diff(sav_diff), 1:12)9   0.121725   0.297362   0.409   0.6836
## L(diff(sav_diff), 1:12)10  0.221600   0.251629   0.881   0.3818
## L(diff(sav_diff), 1:12)11  0.090995   0.202502   0.449   0.6547
## L(diff(sav_diff), 1:12)12 -0.078201   0.126912  -0.616   0.5400
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9821 on 64 degrees of freedom
## Multiple R-squared:  0.6558, Adjusted R-squared:  0.5859
## F-statistic:  9.38 on 13 and 64 DF,  p-value: 1.807e-10
```

```
res_model_sav_diff_start= model_sav_diff_start$residuals
```

Vi tester for Normality

```r
shapiro.test(res_model_sav_diff_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_sav_diff_start
## W = 0.98671, p-value = 0.5971
```

```r
jarque.bera.test(res_model_sav_diff_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_sav_diff_start
## X-squared = 2.5554, df = 2, p-value = 0.2787
```

Vi kan nu fjerne lags med F-test

```r
sav_diff_vars_1 <- str_c("L(diff(sav_diff), 1:12)", 1:12)

linearHypothesis(model_sav_diff_start,sav_diff_vars_1, rep(0, length(sav_diff_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(sav_diff),12)1 = 0
## L(diff(sav_diff),12)2 = 0
## L(diff(sav_diff),12)3 = 0
## L(diff(sav_diff),12)4 = 0
## L(diff(sav_diff),12)5 = 0
## L(diff(sav_diff),12)6 = 0
## L(diff(sav_diff),12)7 = 0
## L(diff(sav_diff),12)8 = 0
## L(diff(sav_diff),12)9 = 0
## L(diff(sav_diff),12)10 = 0
## L(diff(sav_diff),12)11 = 0
## L(diff(sav_diff),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(sav_diff) ~ 1 + L(sav_diff, 1) + L(diff(sav_diff), 1:12)
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     76 74.411
## 2     64 61.730 12    12.682 1.0957 0.3791
```

Vi fjerner variablerne

```r
model_sav_diff <- dynlm(diff(sav_diff) ~ 1+ L(sav_diff, 1))
summary(model_sav_diff)
```

```
##
## Time series regression with "ts" data:
## Start = 1997(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(sav_diff) ~ 1 + L(sav_diff, 1))
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4221 -0.5377  0.0587  0.5966  2.1926
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.2496     0.1026   2.432    0.017 *
## L(sav_diff, 1)  -1.2349     0.1018 -12.126   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9555 on 88 degrees of freedom
## Multiple R-squared:  0.6256, Adjusted R-squared:  0.6213
## F-statistic:   147 on 1 and 88 DF,  p-value: < 2.2e-16
```

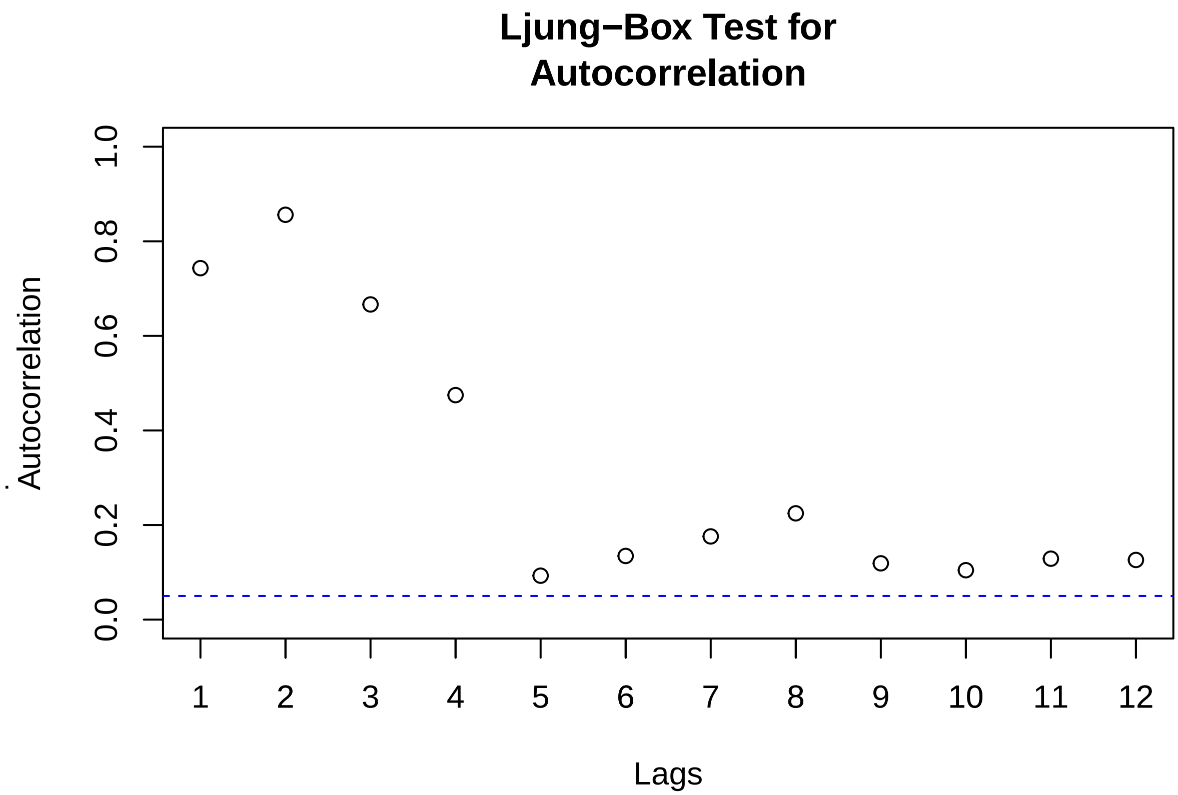
```
model_sav_diff1 <- ur.df(sav_diff, lags = 0, type = "drift")
summary(model_sav_diff1)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4221 -0.5377  0.0587  0.5966  2.1926
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2496     0.1026   2.432    0.017 *
## z.lag.1      -1.2349     0.1018 -12.126   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9555 on 88 degrees of freedom
## Multiple R-squared:  0.6256, Adjusted R-squared:  0.6213
## F-statistic:   147 on 1 and 88 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -12.1257 73.5213
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

```
res_model_sav_diff= residuals(model_sav_diff)
```

Vi tjekker for Serial correlation

```r
arpdiag(res_model_sav_diff, 12)
```




```r
Box.test(res_model_sav_diff, lag=12, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  res_model_sav_diff
## X-squared = 17.669, df = 12, p-value = 0.1261
```

Vi kan se Savings rate er I(1)

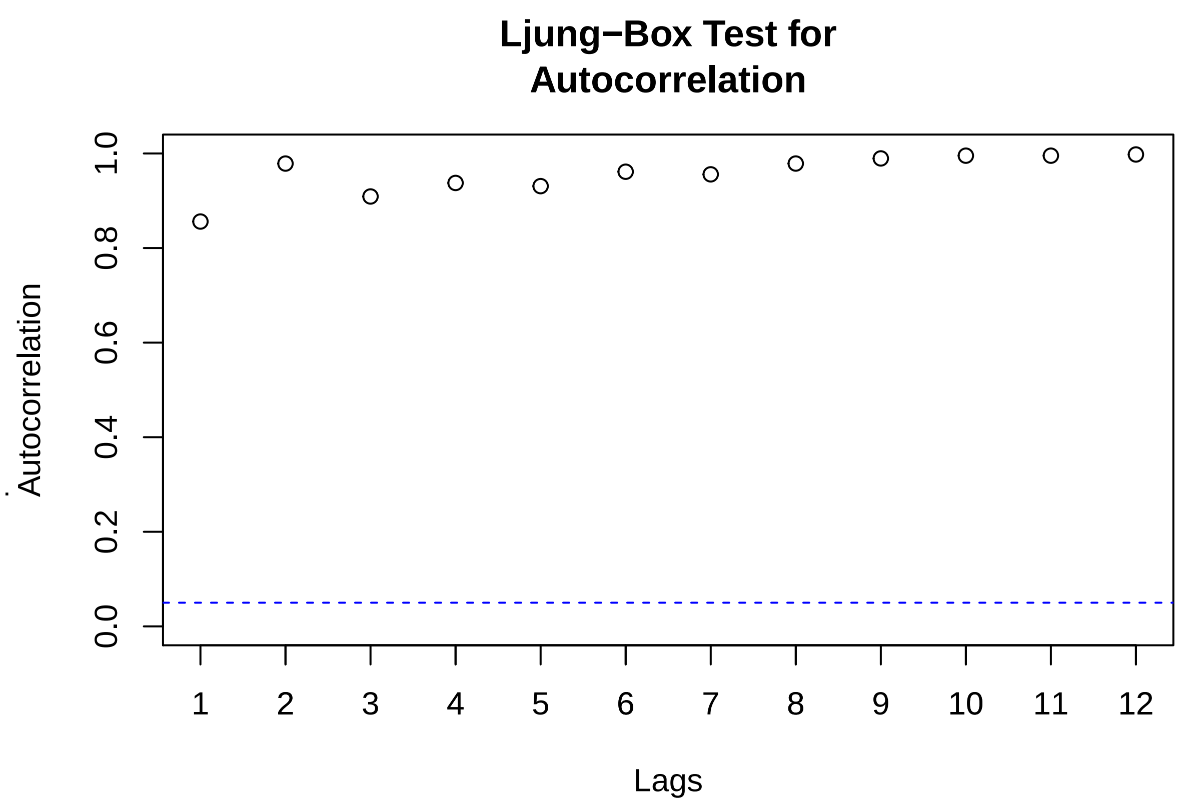
2.7 ADF Test for den korte rente

```r
model_short_i_start <- dynlm(diff(short_i) ~ 1+ L(short_i, 1) + L(diff(short_i), 1:12) + trend(diff(sho
summary(model_short_i_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(short_i) ~ 1 + L(short_i, 1) + L(diff(short_i),
##     1:12) + trend(diff(short_i)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66953 -0.10685  0.00534  0.11112  0.76373
##
```

```
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.927229   0.373671   2.481  0.01572 *
## L(short_i, 1)           -0.171144   0.064311  -2.661  0.00983 **
## L(diff(short_i), 1:12)1  0.514111   0.118134   4.352 4.96e-05 ***
## L(diff(short_i), 1:12)2  0.115334   0.130590   0.883  0.38045
## L(diff(short_i), 1:12)3  0.053172   0.131039   0.406  0.68627
## L(diff(short_i), 1:12)4  0.143655   0.129819   1.107  0.27262
## L(diff(short_i), 1:12)5 -0.061052   0.128794  -0.474  0.63710
## L(diff(short_i), 1:12)6  0.027862   0.128216   0.217  0.82866
## L(diff(short_i), 1:12)7  0.169221   0.126703   1.336  0.18642
## L(diff(short_i), 1:12)8  0.005276   0.124134   0.043  0.96623
## L(diff(short_i), 1:12)9  0.004922   0.123604   0.040  0.96836
## L(diff(short_i), 1:12)10 0.065771   0.121237   0.543  0.58936
## L(diff(short_i), 1:12)11 0.124807   0.121168   1.030  0.30687
## L(diff(short_i), 1:12)12 -0.085436  0.113796  -0.751  0.45553
## trend(diff(short_i))    -0.045837   0.018089  -2.534  0.01374 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3362 on 64 degrees of freedom
## Multiple R-squared:  0.3667, Adjusted R-squared:  0.2282
## F-statistic: 2.647 on 14 and 64 DF,  p-value: 0.004249
```

```r
arpdiag(residuals(model_short_i_start), 12)
```



Test for normality

```r
shapiro.test(residuals(model_short_i_start))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_short_i_start)
## W = 0.81912, p-value = 1.978e-08
```

```r
jarque.bera.test(residuals(model_short_i_start))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model_short_i_start)
## X-squared = 419.74, df = 2, p-value < 2.2e-16
```

Der er problemer med normality i dataet, så der vil nu blive lavet dummies for at forsøge at løse problemet

```r
model_short_i_start$residuals
```

```
##              Qtr1         Qtr2         Qtr3         Qtr4
## 2000               0.483783071  0.635295376 -0.695543429
## 2001  0.032828166 -0.062246023 -0.189566957 -0.439315720
## 2002  0.178268507 -0.043659118 -0.138336316 -0.146241863
## 2003 -0.526624594 -0.217048116  0.099892956  0.068712660
## 2004 -0.159240727 -0.027813543 -0.011137440 -0.217450653
## 2005 -0.129562332 -0.094148032 -0.104002032  0.104694693
## 2006  0.127811656  0.091719179  0.119864184  0.243282543
## 2007  0.142096197  0.270973326  0.390995431  0.242945178
## 2008  0.015221484  0.763731231  0.334840067  0.679028085
## 2009 -1.669530442 -0.151404359  0.117551875  0.190006202
## 2010  0.299642097 -0.132710855 -0.032025980  0.420783565
## 2011  0.094960862  0.267140983  0.154217047  0.005280434
## 2012 -0.081076238 -0.065717453 -0.445698621  0.132053514
## 2013 -0.062350993 -0.160428615  0.014868859 -0.109700703
## 2014 -0.066975716  0.078499920 -0.025786981 -0.084785209
## 2015 -0.409532294  0.069287801  0.055971231 -0.139118375
## 2016  0.007566525 -0.120730700 -0.121404615  0.067611047
## 2017 -0.077213495 -0.019156684 -0.007919509  0.007588550
## 2018 -0.012418817  0.005341360  0.010966798  0.028351670
## 2019  0.026721019  0.015641216  0.003794844  0.097792139
```

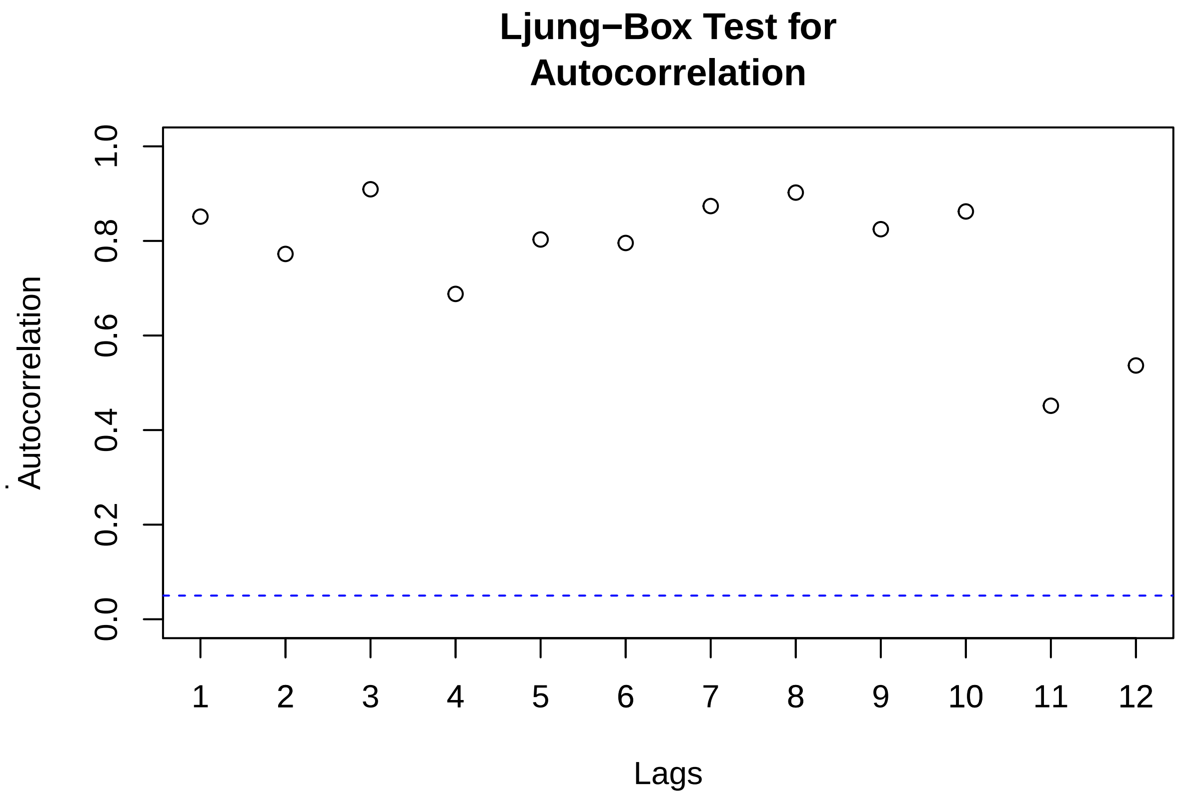
```r
#Sidste plot viser at der skal laves dummies for 2008Q2-Q4.

dummy_short_2008=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2008,2), d

dummy_short_20082=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2008,4),

dummy_short_2009=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2009,1), d

dummies_short = cbind(dummy_short_2008, dummy_short_20082, dummy_short_2009)
```

Modellen køres nu igen med dummies

```r
model_short_i_start2 <- dynlm(diff(short_i) ~ 1+ L(short_i, 1) + L(diff(short_i), 1:12) + trend(diff(sho
summary(model_short_i_start2)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(short_i) ~ 1 + L(short_i, 1) + L(diff(short_i),
##     1:12) + trend(diff(short_i)) + dummies_short)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73271 -0.06949  0.00000  0.09837  0.53873
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.463396   0.280432   1.652  0.10358
## L(short_i, 1)                  -0.086227   0.049504  -1.742  0.08658 .
## L(diff(short_i), 1:12)1         0.592361   0.078642   7.532 2.81e-10 ***
## L(diff(short_i), 1:12)2        -0.005731   0.087515  -0.065  0.94800
## L(diff(short_i), 1:12)3         0.139048   0.088873   1.565  0.12286
## L(diff(short_i), 1:12)4        -0.014918   0.088738  -0.168  0.86705
## L(diff(short_i), 1:12)5        -0.095558   0.085381  -1.119  0.26745
## L(diff(short_i), 1:12)6         0.050839   0.084948   0.598  0.55174
## L(diff(short_i), 1:12)7         0.129294   0.083967   1.540  0.12877
## L(diff(short_i), 1:12)8        -0.028093   0.082169  -0.342  0.73360
## L(diff(short_i), 1:12)9         0.010868   0.081875   0.133  0.89484
## L(diff(short_i), 1:12)10        0.029150   0.080285   0.363  0.71780
## L(diff(short_i), 1:12)11        0.129547   0.080172   1.616  0.11129
## L(diff(short_i), 1:12)12       -0.087561   0.075486  -1.160  0.25058
## trend(diff(short_i))           -0.022880   0.013718  -1.668  0.10046
## dummies_shortdummy_short_2008   0.648709   0.237023   2.737  0.00811 **
## dummies_shortdummy_short_20082  0.536489   0.252576   2.124  0.03773 *
## dummies_shortdummy_short_2009  -2.012334   0.263842  -7.627 1.93e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2223 on 61 degrees of freedom
## Multiple R-squared:  0.7361, Adjusted R-squared:  0.6625
## F-statistic: 10.01 on 17 and 61 DF,  p-value: 6.304e-12
```

```
arpdiag(residuals(model_short_i_start2), 12)
```



Dummiesne er signifikante og der er ingen seriekorrelation

```
shapiro.test(residuals(model_short_i_start2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_short_i_start2)
## W = 0.91381, p-value = 5.451e-05
```

```
jarque.bera.test(residuals(model_short_i_start2))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model_short_i_start2)
## X-squared = 31.994, df = 2, p-value = 1.128e-07
```

Men der er stadig store problemer med normalfordelingen af residuals, så det er ikke muligt at anvende den korte rente i en model.

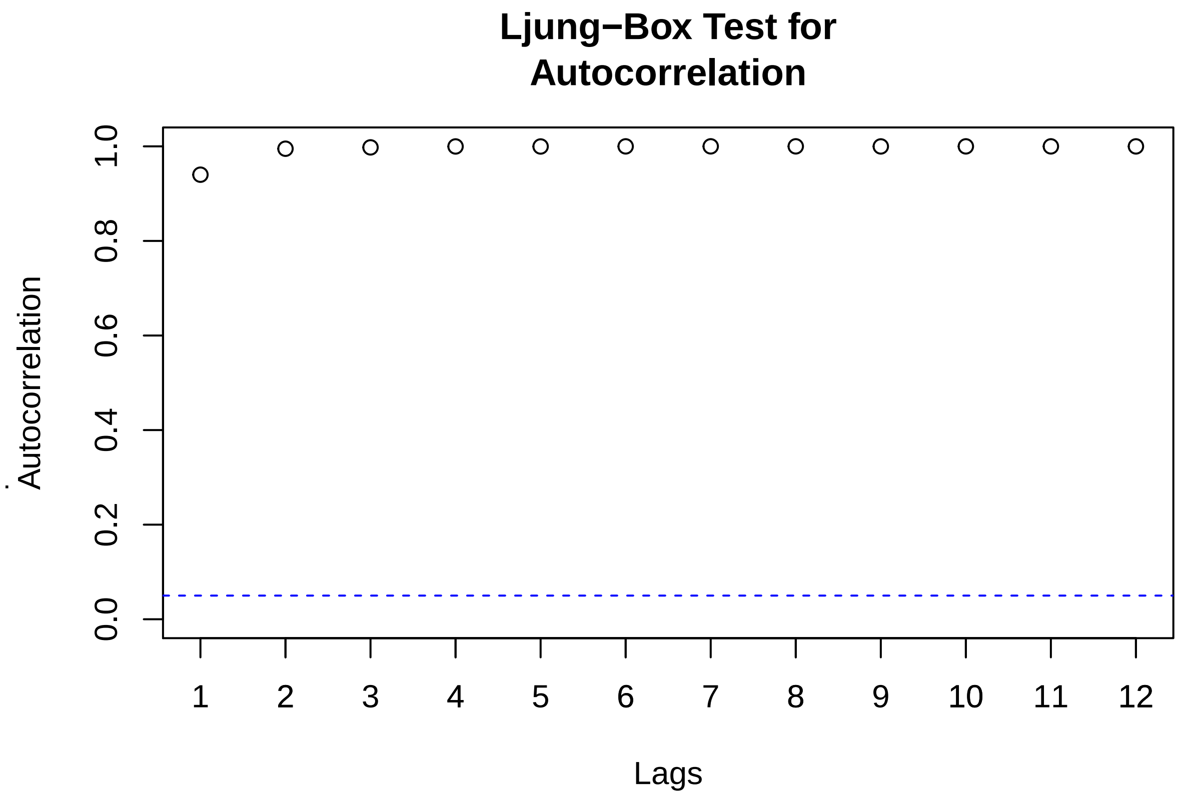
2.8 ADF Test for huspriser

```
model_hus_start <- dynlm(diff(hus) ~ 1+ L(hus, 1) + L(diff(hus), 1:12) + trend(diff(hus)))
summary(model_hus_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
```

```
## dynlm(formula = diff(hus) ~ 1 + L(hus, 1) + L(diff(hus), 1:12) +
##     trend(diff(hus)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3271 -0.3961 -0.0684  0.4528  3.7602
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.91000    0.83806   2.279   0.0260 *
## L(hus, 1)             -0.03349    0.01687  -1.985   0.0514 .
## L(diff(hus), 1:12)1    1.02348    0.12157   8.419 5.89e-12 ***
## L(diff(hus), 1:12)2   -0.33233    0.17308  -1.920   0.0593 .
## L(diff(hus), 1:12)3    0.30585    0.17749   1.723   0.0897 .
## L(diff(hus), 1:12)4   -0.40928    0.18209  -2.248   0.0281 *
## L(diff(hus), 1:12)5    0.29226    0.18990   1.539   0.1287
## L(diff(hus), 1:12)6    0.11180    0.19046   0.587   0.5593
## L(diff(hus), 1:12)7   -0.25495    0.19005  -1.341   0.1845
## L(diff(hus), 1:12)8    0.20896    0.18853   1.108   0.2718
## L(diff(hus), 1:12)9   -0.09909    0.18233  -0.543   0.5887
## L(diff(hus), 1:12)10   0.16863    0.17953   0.939   0.3511
## L(diff(hus), 1:12)11  -0.24827    0.17622  -1.409   0.1637
## L(diff(hus), 1:12)12   0.10874    0.12537   0.867   0.3890
## trend(diff(hus))       0.08886    0.05384   1.650   0.1038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.061 on 64 degrees of freedom
## Multiple R-squared:  0.7505, Adjusted R-squared:  0.696
## F-statistic: 13.75 on 14 and 64 DF,  p-value: 2.658e-14
```

```
arpdiag(residuals(model_hus_start), 12)
```



Test for normality

```
shapiro.test(residuals(model_hus_start))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_hus_start)
## W = 0.87554, p-value = 1.476e-06
```

```
jarque.bera.test(residuals(model_hus_start))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model_hus_start)
## X-squared = 94.315, df = 2, p-value < 2.2e-16
```

Der er problemer med normality i dataet, så der vil nu blive lavet dummies for at forsøge at løse problemet

```
model_hus_start$residuals
```

```
##              Qtr1         Qtr2         Qtr3         Qtr4
## 2000              -0.306521087 -0.157198588 -0.424740873
## 2001 -0.246755984 -0.748714811 -0.178108268 -0.726699522
## 2002  0.127757467 -0.585594957 -0.592299886  0.031640629
## 2003 -0.457329553 -0.068436959 -0.305898195 -0.023159868
## 2004  0.564443842 -0.035663326  0.158647328  0.481445105
## 2005  0.804731423  0.202139400  3.336872930  1.466111685
## 2006  1.372358101 -0.883604995 -0.322956395 -0.125943319
## 2007  0.338048943 -0.329173414  0.553440669  0.007482986
```

```
## 2008 -0.397639907  0.053968296 -0.941018448 -2.847390169
## 2009 -0.080117936  3.760210465  0.280427318  0.299610769
## 2010 -0.505714063  0.961052671  0.616930511 -0.455553929
## 2011 -0.821523745  0.623947623 -3.327112839  0.606571461
## 2012 -0.097051444 -0.060636497  0.621884689 -0.277333564
## 2013  0.174441481 -0.350003453 -0.087798798 -0.389455906
## 2014 -0.394611253  0.667009479 -1.573710176  1.190329393
## 2015  0.612806903 -0.939606459 -0.532328870  0.746781470
## 2016  0.013392965 -1.340854640  1.336168393 -0.811805226
## 2017 -0.042255757  0.574063674 -0.178446296  0.424056542
## 2018  0.032318305 -0.112006973  0.342914189 -0.892348611
## 2019  0.808289439 -0.260618499  0.264785808 -0.221338896
```

```
#Sidste plot viser at der skal laves dummies for 2005Q3-2006Q1, 2008Q1, 2009Q2 og for 2011Q3.
dummy_hus_2005=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2005,3), du

dummy_hus_2008=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2008,4), du

dummy_hus_2009=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2009,2), du

dummy_hus_2011=create_dummy_ts(start_basic = c(1997, 1), end_basic=c(2019,4), dummy_start=c(2011,3), du

dummies_hus = cbind(dummy_hus_2005, dummy_hus_2008, dummy_hus_2009, dummy_hus_2011)
```

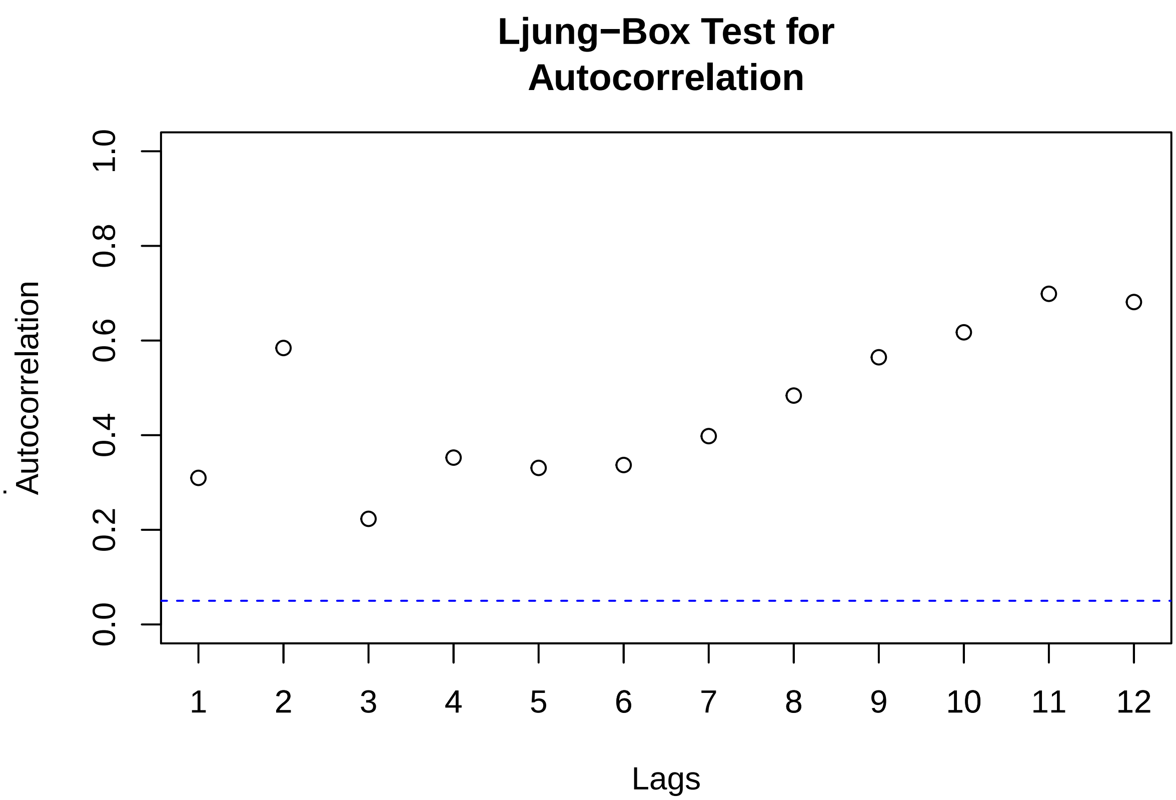
Modellen køres nu igen med dummies

```
model_hus_start2 <- dynlm(diff(hus) ~ 1+ L(hus, 1) + L(diff(hus), 1:12) + trend(diff(hus)) + dummies_hu
summary(model_hus_start2)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(hus) ~ 1 + L(hus, 1) + L(diff(hus), 1:12) +
##     trend(diff(hus)) + dummies_hus)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.42650 -0.27808 -0.03659  0.36240  1.10751
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.8722363  0.4701969   3.982 0.000187 ***
## L(hus, 1)               -0.0336678  0.0096733  -3.480 0.000939 ***
## L(diff(hus), 1:12)1      0.8298181  0.0732070  11.335  < 2e-16 ***
## L(diff(hus), 1:12)2     -0.2413257  0.1029729  -2.344 0.022431 *
## L(diff(hus), 1:12)3      0.3023082  0.1004001   3.011 0.003806 **
## L(diff(hus), 1:12)4     -0.3735038  0.1000462  -3.733 0.000422 ***
## L(diff(hus), 1:12)5      0.2781228  0.1048119   2.654 0.010180 *
## L(diff(hus), 1:12)6     -0.0006277  0.1088573  -0.006 0.995419
## L(diff(hus), 1:12)7     -0.1405171  0.1072385  -1.310 0.195080
## L(diff(hus), 1:12)8      0.1429132  0.1052636   1.358 0.179653
## L(diff(hus), 1:12)9      0.0908859  0.1042128   0.872 0.386620
## L(diff(hus), 1:12)10    -0.0231194  0.1064696  -0.217 0.828831
```

```
## L(diff(hus), 1:12)11      -0.2828624  0.1010587  -2.799 0.006884 **
## L(diff(hus), 1:12)12       0.1208107  0.0716975   1.685 0.097183 .
## trend(diff(hus))           0.0999603  0.0309611   3.229 0.002019 **
## dummies_husdummy_hus_2005  2.7469339  0.4142593   6.631 1.06e-08 ***
## dummies_husdummy_hus_2008 -2.7705200  0.6788857  -4.081 0.000134 ***
## dummies_husdummy_hus_2009  3.1308637  0.7291958   4.294 6.52e-05 ***
## dummies_husdummy_hus_2011 -4.5258880  0.8015813  -5.646 4.75e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5824 on 60 degrees of freedom
## Multiple R-squared:  0.9295, Adjusted R-squared:  0.9084
## F-statistic: 43.96 on 18 and 60 DF,  p-value: < 2.2e-16
```

```
arpdiag(residuals(model_hus_start2), 12)
```



Dummiesne er signifikante og der er ingen seriekorrelation

```
shapiro.test(residuals(model_hus_start2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_hus_start2)
## W = 0.98009, p-value = 0.2542
```

```
jarque.bera.test(residuals(model_hus_start2))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model_hus_start2)
```

```
## X-squared = 0.017021, df = 2, p-value = 0.9915
```

Vi fjerner lags med F-test

```
hus_vars_1 <- str_c("L(diff(hus), 1:12)", 6:12)

linearHypothesis(model_hus_start,hus_vars_1, rep(0, length(hus_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(hus),12)6 = 0
## L(diff(hus),12)7 = 0
## L(diff(hus),12)8 = 0
## L(diff(hus),12)9 = 0
## L(diff(hus),12)10 = 0
## L(diff(hus),12)11 = 0
## L(diff(hus),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(hus) ~ 1 + L(hus, 1) + L(diff(hus), 1:12) + trend(diff(hus))
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     71 75.477
## 2     64 72.035  7    3.4417 0.4368 0.8754
```

F-testen er klart bestået

```
model_hus <- dynlm(diff(hus) ~ 1+ L(hus, 1) + L(diff(hus), 1:5) + trend(diff(gold)) + dummies_hus)
summary(model_hus)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(hus) ~ 1 + L(hus, 1) + L(diff(hus), 1:5) +
##     trend(diff(gold)) + dummies_hus)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33345 -0.29840 -0.00268  0.30647  1.19959
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.885928   0.388134   4.859 6.43e-06 ***
## L(hus, 1)                 -0.037713   0.007738  -4.874 6.07e-06 ***
## L(diff(hus), 1:5)1         0.781080   0.070328  11.106  < 2e-16 ***
## L(diff(hus), 1:5)2        -0.169087   0.092109  -1.836 0.070415 .
## L(diff(hus), 1:5)3         0.301117   0.089019   3.383 0.001150 **
## L(diff(hus), 1:5)4        -0.384803   0.087002  -4.423 3.29e-05 ***
## L(diff(hus), 1:5)5         0.244315   0.065274   3.743 0.000357 ***
## trend(diff(gold))          0.119603   0.025737   4.647 1.43e-05 ***
## dummies_husdummy_hus_2005  2.859963   0.422492   6.769 2.64e-09 ***
## dummies_husdummy_hus_2008 -3.574831   0.644168  -5.550 4.24e-07 ***
## dummies_husdummy_hus_2009  3.306363   0.710422   4.654 1.40e-05 ***
```

```
## dummies_husdummy_hus_2011 -2.960795   0.620076  -4.775 8.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6021 on 74 degrees of freedom
## Multiple R-squared:  0.9072, Adjusted R-squared:  0.8934
## F-statistic: 65.76 on 11 and 74 DF,  p-value: < 2.2e-16
```

```r
model_hus2= ur.df(hus, lags = 5, type = "trend")
summary(model_hus2)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2238 -0.4176 -0.0910  0.3839  3.7502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.72071    0.63089   2.727  0.00788 **
## z.lag.1     -0.03292    0.01234  -2.667  0.00929 **
## tt           0.02429    0.01025   2.369  0.02029 *
## z.diff.lag1  0.98550    0.10571   9.323 2.56e-14 ***
## z.diff.lag2 -0.29520    0.14451  -2.043  0.04446 *
## z.diff.lag3  0.32112    0.14455   2.222  0.02921 *
## z.diff.lag4 -0.42974    0.14429  -2.978  0.00386 **
## z.diff.lag5  0.29822    0.10739   2.777  0.00687 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.002 on 78 degrees of freedom
## Multiple R-squared:  0.7292, Adjusted R-squared:  0.7049
## F-statistic: 30.01 on 7 and 78 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -2.6673 2.8798 3.5626
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

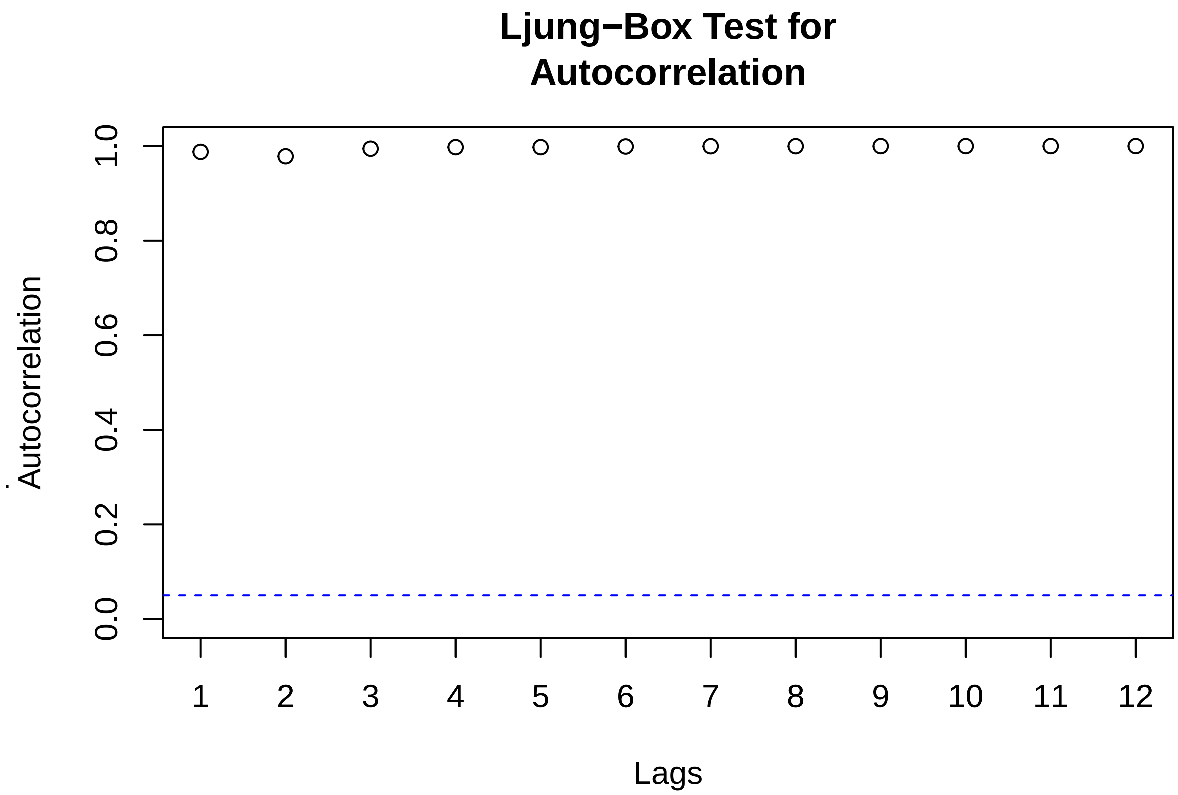
Der er altså ikke unitroot i dataet for huspriser, da t-stat er -4.874 og tau-værdi er -3.45, så huspriser er I(0).

2.9 ADF Test for guld

```r
model_gold_start <- dynlm(diff(gold) ~ 1+ L(gold, 1) + L(diff(gold), 1:12) + trend(diff(gold)))
summary(model_gold_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(4)
##
## Call:
## dynlm(formula = diff(gold) ~ 1 + L(gold, 1) + L(diff(gold), 1:12) +
##     trend(diff(gold)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -886.27 -162.10  -16.93  164.36 1073.60
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -3.95468  108.64009  -0.036   0.9711
## L(gold, 1)              -0.11986    0.05484  -2.186   0.0325 *
## L(diff(gold), 1:12)1     0.26730    0.12282   2.176   0.0332 *
## L(diff(gold), 1:12)2     0.19321    0.13125   1.472   0.1459
## L(diff(gold), 1:12)3     0.01178    0.13298   0.089   0.9297
## L(diff(gold), 1:12)4     0.12704    0.13347   0.952   0.3448
## L(diff(gold), 1:12)5     0.15362    0.13417   1.145   0.2565
## L(diff(gold), 1:12)6    -0.10087    0.13724  -0.735   0.4650
## L(diff(gold), 1:12)7     0.03576    0.13514   0.265   0.7921
## L(diff(gold), 1:12)8    -0.07890    0.13541  -0.583   0.5621
## L(diff(gold), 1:12)9     0.09625    0.13532   0.711   0.4795
## L(diff(gold), 1:12)10    0.17688    0.13474   1.313   0.1940
## L(diff(gold), 1:12)11   -0.15726    0.13641  -1.153   0.2532
## L(diff(gold), 1:12)12    0.02548    0.13439   0.190   0.8502
## trend(diff(gold))       54.59663   24.42784   2.235   0.0289 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 370.8 on 64 degrees of freedom
## Multiple R-squared:  0.2256, Adjusted R-squared:  0.05623
## F-statistic: 1.332 on 14 and 64 DF,  p-value: 0.2143
```

```r
arpdiag(residuals(model_gold_start), 12)
```



Test for normality

```
shapiro.test(residuals(model_gold_start))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_gold_start)
## W = 0.97266, p-value = 0.08754
```

```
jarque.bera.test(residuals(model_gold_start))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model_gold_start)
## X-squared = 4.9008, df = 2, p-value = 0.08626
```

Der er ingen problemer med normality i start modellen.

Vi fjerner lags med F-test

```
gold_vars_1 <- str_c("L(diff(gold), 1:12)", 2:12)

linearHypothesis(model_gold_start,gold_vars_1, rep(0, length(gold_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(gold),12)2 = 0
## L(diff(gold),12)3 = 0
## L(diff(gold),12)4 = 0
```

```
## L(diff(gold),12)5 = 0
## L(diff(gold),12)6 = 0
## L(diff(gold),12)7 = 0
## L(diff(gold),12)8 = 0
## L(diff(gold),12)9 = 0
## L(diff(gold),12)10 = 0
## L(diff(gold),12)11 = 0
## L(diff(gold),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(gold) ~ 1 + L(gold, 1) + L(diff(gold), 1:12) + trend(diff(gold))
##
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1     75 10101951
## 2     64  8799191 11   1302760 0.8614 0.5813
```

F-testen er klart bestået

```
model_gold <- dynlm(diff(gold) ~ 1+ L(gold, 1) + L(diff(gold), 1) + trend(diff(gold)))
summary(model_gold)
```

```
##
## Time series regression with "ts" data:
## Start = 1997(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(gold) ~ 1 + L(gold, 1) + L(diff(gold), 1) +
##     trend(diff(gold)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -894.51 -156.56  -21.66  107.90 1193.46
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       64.57770   77.41505   0.834   0.4065
## L(gold, 1)        -0.07604    0.03385  -2.246   0.0272 *
## L(diff(gold), 1)   0.27040    0.10273   2.632   0.0101 *
## trend(diff(gold)) 33.68176   14.33585   2.349   0.0211 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 345.2 on 86 degrees of freedom
## Multiple R-squared:  0.1178, Adjusted R-squared:  0.08707
## F-statistic: 3.829 on 3 and 86 DF,  p-value: 0.0126
```

```
model_gold2= ur.df(gold, lags = 1, type = "trend")
summary(model_gold2)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
```

```
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -894.51 -156.56  -21.66  107.90 1193.46
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 64.57770   77.41505   0.834   0.4065
## z.lag.1     -0.07604    0.03385  -2.246   0.0272 *
## tt           8.42044    3.58396   2.349   0.0211 *
## z.diff.lag   0.27040    0.10273   2.632   0.0101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 345.2 on 86 degrees of freedom
## Multiple R-squared:  0.1178, Adjusted R-squared:  0.08707
## F-statistic: 3.829 on 3 and 86 DF,  p-value: 0.0126
##
##
## Value of test-statistic is: -2.2463 2.8668 2.7838
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Der er altså unitroot i dataet for guld, så der bliver nu taget diff.

Vi skal tage Diff af guld

```r
gold_diff=diff(gold)
```

Vi refitter vores model igen

```r
model_gold_diff_start <- dynlm(diff(gold_diff) ~ 0+ L(gold_diff, 1) + L(diff(gold_diff), 1:12))
summary(model_gold_diff_start)
```

```
##
## Time series regression with "ts" data:
## Start = 2000(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(gold_diff) ~ 0 + L(gold_diff, 1) + L(diff(gold_diff),
##     1:12))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -960.72  -66.90   50.71  257.75 1039.80
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## L(gold_diff, 1)             -6.955e-01  2.973e-01  -2.339   0.0224 *
```

```
## L(diff(gold_diff), 1:12)1  -3.334e-02  2.911e-01  -0.115   0.9092
## L(diff(gold_diff), 1:12)2   1.104e-01  2.809e-01   0.393   0.6957
## L(diff(gold_diff), 1:12)3   8.858e-02  2.733e-01   0.324   0.7469
## L(diff(gold_diff), 1:12)4   1.808e-01  2.621e-01   0.690   0.4928
## L(diff(gold_diff), 1:12)5   2.748e-01  2.492e-01   1.103   0.2742
## L(diff(gold_diff), 1:12)6   1.147e-01  2.360e-01   0.486   0.6287
## L(diff(gold_diff), 1:12)7   1.032e-01  2.226e-01   0.464   0.6445
## L(diff(gold_diff), 1:12)8   5.377e-05  2.150e-01   0.000   0.9998
## L(diff(gold_diff), 1:12)9   9.432e-02  2.035e-01   0.464   0.6445
## L(diff(gold_diff), 1:12)10  2.443e-01  1.821e-01   1.342   0.1844
## L(diff(gold_diff), 1:12)11  6.519e-02  1.659e-01   0.393   0.6957
## L(diff(gold_diff), 1:12)12  1.153e-01  1.369e-01   0.843   0.4025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 388.9 on 65 degrees of freedom
## Multiple R-squared:  0.4288, Adjusted R-squared:  0.3145
## F-statistic: 3.753 on 13 and 65 DF,  p-value: 0.0001805
```

```
res_model_gold_diff_start= model_gold_diff_start$residuals
```

Vi tester for Normality

```
shapiro.test(res_model_gold_diff_start)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_model_gold_diff_start
## W = 0.97915, p-value = 0.2301
```

```
jarque.bera.test(res_model_gold_diff_start)
```

```
##
##  Jarque Bera Test
##
## data:  res_model_gold_diff_start
## X-squared = 3.2071, df = 2, p-value = 0.2012
```

Der er ikke problemer med normality

Vi kan nu fjerne lags med F-test

```
gold_diff_vars_1 <- str_c("L(diff(gold_diff), 1:12)", 1:12)

linearHypothesis(model_gold_diff_start,gold_diff_vars_1, rep(0, length(gold_diff_vars_1)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(diff(gold_diff),12)1 = 0
## L(diff(gold_diff),12)2 = 0
## L(diff(gold_diff),12)3 = 0
## L(diff(gold_diff),12)4 = 0
## L(diff(gold_diff),12)5 = 0
## L(diff(gold_diff),12)6 = 0
## L(diff(gold_diff),12)7 = 0
## L(diff(gold_diff),12)8 = 0
```

```
## L(diff(gold_diff),12)9 = 0
## L(diff(gold_diff),12)10 = 0
## L(diff(gold_diff),12)11 = 0
## L(diff(gold_diff),12)12 = 0
##
## Model 1: restricted model
## Model 2: diff(gold_diff) ~ 0 + L(gold_diff, 1) + L(diff(gold_diff), 1:12)
##
##   Res.Df      RSS Df Sum of Sq     F Pr(>F)
## 1     77 11107224
## 2     65  9829699 12   1277525 0.704 0.7419
```

F-testen er ikke signifikant, så vi kan fjerne lagsne

```r
model_gold_diff <- dynlm(diff(gold_diff) ~ 1+L(gold_diff, 1))
summary(model_gold_diff)
```

```
##
## Time series regression with "ts" data:
## Start = 1997(3), End = 2019(4)
##
## Call:
## dynlm(formula = diff(gold_diff) ~ 1 + L(gold_diff, 1))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1065.18  -161.08   -50.34   129.33  1105.14
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       65.1368    38.1548   1.707   0.0913 .
## L(gold_diff, 1)   -0.7536     0.1033  -7.296 1.24e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 352.2 on 88 degrees of freedom
## Multiple R-squared:  0.3769, Adjusted R-squared:  0.3698
## F-statistic: 53.23 on 1 and 88 DF,  p-value: 1.241e-10
```

```r
model_gold_diff1 <- ur.df(gold_diff, lags = 0, type = "drift")
summary(model_gold_diff1)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1065.18  -161.08   -50.34   129.33  1105.14
```

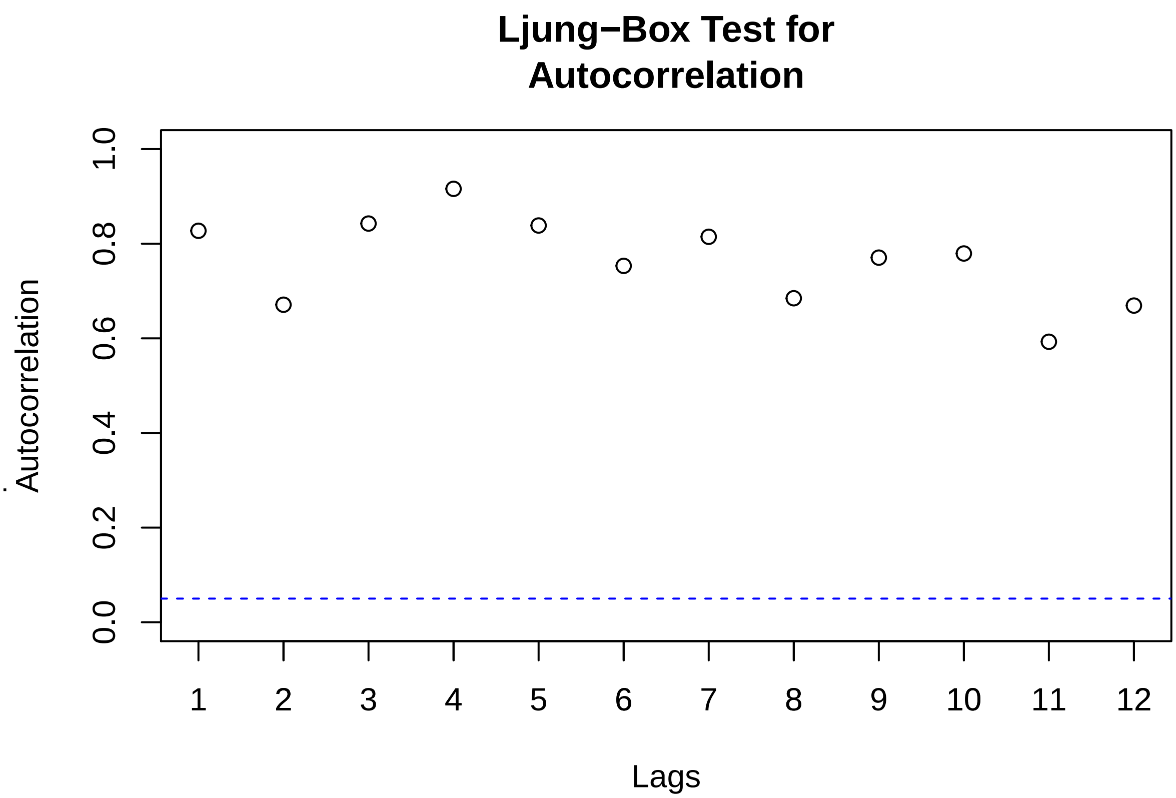
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  65.1368    38.1548   1.707   0.0913 .
## z.lag.1      -0.7536     0.1033  -7.296 1.24e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 352.2 on 88 degrees of freedom
## Multiple R-squared:  0.3769, Adjusted R-squared:  0.3698
## F-statistic: 53.23 on 1 and 88 DF,  p-value: 1.241e-10
##
##
## Value of test-statistic is: -7.2956 26.6135
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

```
res_model_gold_diff= residuals(model_gold_diff)
```

Der er altså ikke længere unitroot og gold er I(1)

Tester fpr seriekorrelation

```
arpdiag(res_model_gold_diff, lag=12)
```



Der er altså ingen seriekorrelation.

3 Engle- Granger test

Level- level

```
eg_test=lm(LC20~inf+bnp+long_i+u)

options(scipen = 999)

summary(eg_test)
```

```
##
## Call:
## lm(formula = LC20 ~ inf + bnp + long_i + u)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45046 -0.12534  0.00369  0.11631  0.34748
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept)  3.072877   0.795000   3.865    0.000213 ***
## inf         -0.061926   0.028758  -2.153    0.034060 *
## bnp          0.007659   0.001362   5.625 0.000000222 ***
## long_i      -0.110971   0.034314  -3.234    0.001726 **
## u           -0.012687   0.022003  -0.577    0.565692
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1728 on 87 degrees of freedom
## Multiple R-squared:  0.9082, Adjusted R-squared:  0.9039
## F-statistic: 215.1 on 4 and 87 DF,  p-value: < 0.00000000000000022
```

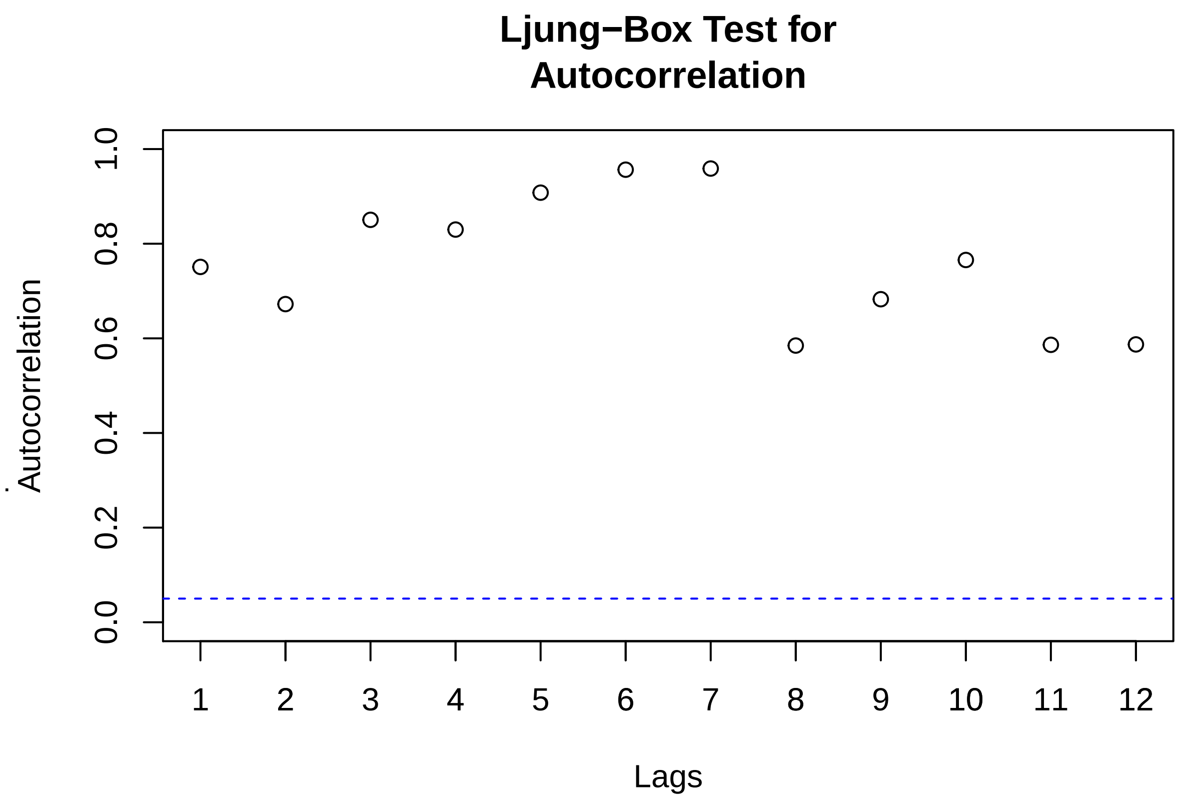
We should obtain the error term

```
res_eg_test=eg_test$residuals

model_eg_test <- ur.df(res_eg_test, lags = 1, type = "drift")

res_model_eg_test=model_eg_test@testreg$residuals

summary(model_eg_test)
```

```
##
## ###############################################
## # 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
## -0.35723 -0.05122  0.00685  0.06576  0.17232
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0003477  0.0094486  -0.037 0.970728
## z.lag.1     -0.2296713  0.0588577  -3.902 0.000187 ***
## z.diff.lag   0.3634609  0.1013066   3.588 0.000551 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08959 on 87 degrees of freedom
## Multiple R-squared:  0.1979, Adjusted R-squared:  0.1795
## F-statistic: 10.74 on 2 and 87 DF,  p-value: 0.00006805
##
##
## Value of test-statistic is: -3.9021 7.6212
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

```
arpdiag(res_model_eg_test,12)
```



we need to look at the Engel Granger critical values. for a model with drift and no trend we have a critical value when n=100 on -4.4185 on 1% significans level.

We conclude there is no long run relationship.

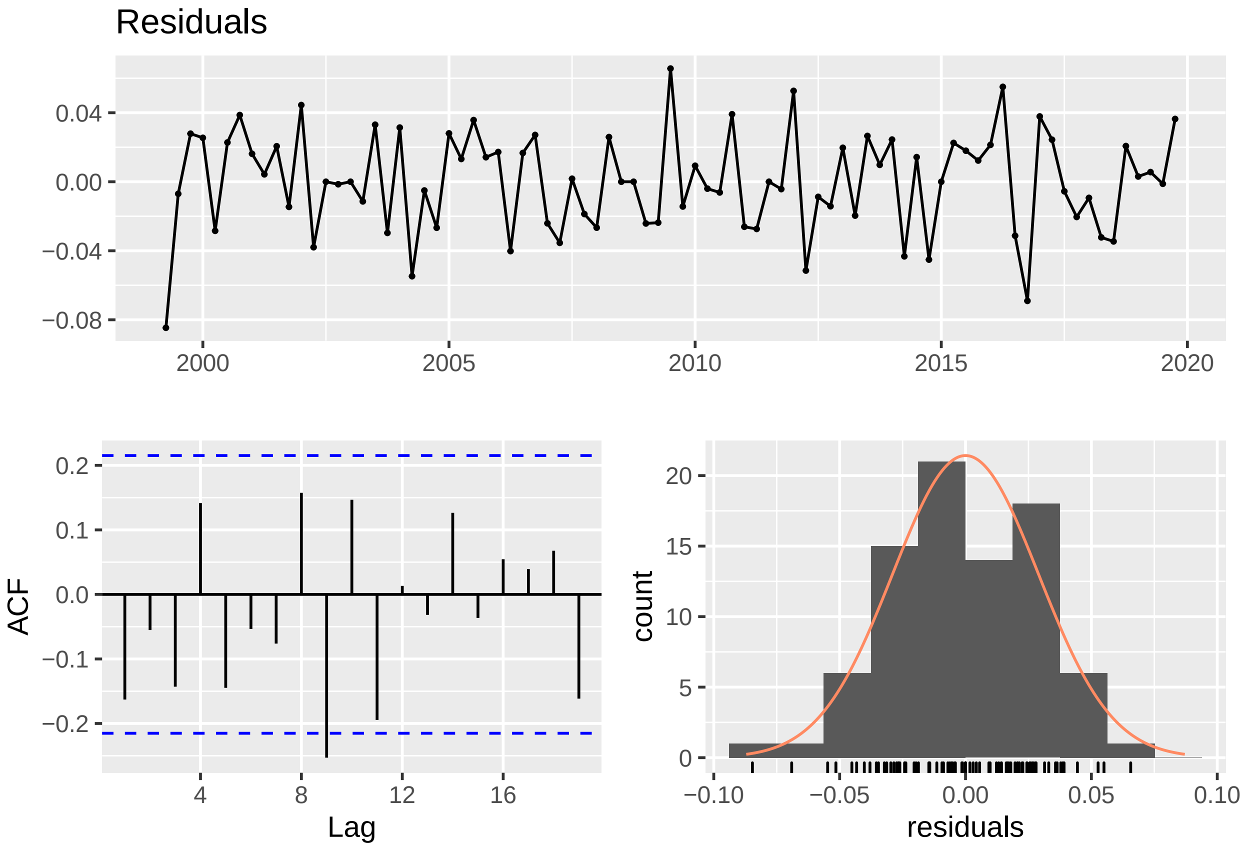
4 ADRL Bounds-test

First we create the ECM model

We start by using 8 lags.

```
ecm1 <- dynlm(LC20_diff~L(LC20_diff, 1:8)+ L(long_i_diff, 0:8)+L(u_diff, 0:8)+ L(inf_diff, 0:8) +L(bnp_

checkresiduals(residuals(ecm1))
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
summary(ecm1)
```

```
##
## Time series regression with "ts" data:
## Start = 1999(2), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1:8) + L(long_i_diff,
##     0:8) + L(u_diff, 0:8) + L(inf_diff, 0:8) + L(bnp_diff, 0:8) +
##     L(LC20, 1) + L(long_i, 1) + L(inf, 1) + L(bnp, 1) + L(u,
##     1) + dummies_LC20_diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08465 -0.02207  0.00000  0.02194  0.06561
##
```

```
## Coefficients:
##                                          Estimate  Std. Error t value
## (Intercept)                            1.72260478  1.40872921   1.223
## L(LC20_diff, 1:8)1                     0.36005987  0.13658572   2.636
## L(LC20_diff, 1:8)2                     0.08026892  0.16597663   0.484
## L(LC20_diff, 1:8)3                    -0.06129215  0.15808487  -0.388
## L(LC20_diff, 1:8)4                     0.17695226  0.16396081   1.079
## L(LC20_diff, 1:8)5                     0.01113001  0.18363744   0.061
## L(LC20_diff, 1:8)6                    -0.08854608  0.18509030  -0.478
## L(LC20_diff, 1:8)7                     0.03556717  0.14820203   0.240
## L(LC20_diff, 1:8)8                    -0.03708236  0.15978950  -0.232
## L(long_i_diff, 0:8)0                   0.02188408  0.04071717   0.537
## L(long_i_diff, 0:8)1                   0.04922656  0.06054266   0.813
## L(long_i_diff, 0:8)2                   0.04613456  0.05617063   0.821
## L(long_i_diff, 0:8)3                   0.02178785  0.05602324   0.389
## L(long_i_diff, 0:8)4                   0.09001036  0.04420204   2.036
## L(long_i_diff, 0:8)5                   0.05757621  0.04391136   1.311
## L(long_i_diff, 0:8)6                  -0.01818848  0.04133316  -0.440
## L(long_i_diff, 0:8)7                   0.03617482  0.04032173   0.897
## L(long_i_diff, 0:8)8                  -0.04791888  0.03915953  -1.224
## L(u_diff, 0:8)0                       -0.08548197  0.05668590  -1.508
## L(u_diff, 0:8)1                        0.00346195  0.06362397   0.054
## L(u_diff, 0:8)2                       -0.01696604  0.05175131  -0.328
## L(u_diff, 0:8)3                        0.09251513  0.06001928   1.541
## L(u_diff, 0:8)4                       -0.00439156  0.06112933  -0.072
## L(u_diff, 0:8)5                       -0.08794257  0.07304473  -1.204
## L(u_diff, 0:8)6                       -0.07634857  0.05471310  -1.395
## L(u_diff, 0:8)7                        0.00489987  0.05171108   0.095
## L(u_diff, 0:8)8                        0.01248219  0.04973028   0.251
## L(inf_diff, 0:8)0                      0.03015897  0.03455647   0.873
## L(inf_diff, 0:8)1                      0.03082670  0.03523812   0.875
## L(inf_diff, 0:8)2                      0.03293766  0.03497871   0.942
## L(inf_diff, 0:8)3                      0.05895581  0.03938018   1.497
## L(inf_diff, 0:8)4                      0.06187048  0.03961437   1.562
## L(inf_diff, 0:8)5                      0.00895720  0.03051175   0.294
## L(inf_diff, 0:8)6                      0.02054072  0.02828998   0.726
## L(inf_diff, 0:8)7                      0.02521899  0.02663935   0.947
## L(inf_diff, 0:8)8                      0.05891344  0.03538718   1.665
## L(bnp_diff, 0:8)0                     -0.00411073  0.00269853  -1.523
## L(bnp_diff, 0:8)1                     -0.00378522  0.00271899  -1.392
## L(bnp_diff, 0:8)2                     -0.00157602  0.00275034  -0.573
## L(bnp_diff, 0:8)3                     -0.00127692  0.00291172  -0.439
## L(bnp_diff, 0:8)4                     -0.00344096  0.00292942  -1.175
## L(bnp_diff, 0:8)5                      0.00268688  0.00309781   0.867
## L(bnp_diff, 0:8)6                     -0.00561835  0.00327027  -1.718
## L(bnp_diff, 0:8)7                     -0.00545575  0.00410874  -1.328
## L(bnp_diff, 0:8)8                     -0.00364197  0.00393119  -0.926
## L(LC20, 1)                            -0.23660682  0.12964872  -1.825
## L(long_i, 1)                          -0.05849883  0.06533649  -0.895
## L(inf, 1)                             -0.02781225  0.03364121  -0.827
## L(bnp, 1)                              0.00008314  0.00135090   0.062
## L(u, 1)                               -0.00328463  0.03234133  -0.102
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.33542234  0.10510561  -3.191
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.16898653  0.09544005  -1.771
```

```
## dummies_LC20_diffdummy_LC20_diff_2015    0.26862883  0.07769728   3.457
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.07560396  0.08525590  -0.887
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.10994133  0.10056871  -1.093
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.15032168  0.08654861  -1.737
##                                         Pr(>|t|)
## (Intercept)                              0.23197
## L(LC20_diff, 1:8)1                       0.01373 *
## L(LC20_diff, 1:8)2                       0.63256
## L(LC20_diff, 1:8)3                       0.70127
## L(LC20_diff, 1:8)4                       0.29003
## L(LC20_diff, 1:8)5                       0.95212
## L(LC20_diff, 1:8)6                       0.63622
## L(LC20_diff, 1:8)7                       0.81215
## L(LC20_diff, 1:8)8                       0.81823
## L(long_i_diff, 0:8)0                     0.59535
## L(long_i_diff, 0:8)1                     0.42328
## L(long_i_diff, 0:8)2                     0.41865
## L(long_i_diff, 0:8)3                     0.70039
## L(long_i_diff, 0:8)4                     0.05163 .
## L(long_i_diff, 0:8)5                     0.20084
## L(long_i_diff, 0:8)6                     0.66341
## L(long_i_diff, 0:8)7                     0.37757
## L(long_i_diff, 0:8)8                     0.23164
## L(u_diff, 0:8)0                          0.14317
## L(u_diff, 0:8)1                          0.95701
## L(u_diff, 0:8)2                          0.74556
## L(u_diff, 0:8)3                          0.13485
## L(u_diff, 0:8)4                          0.94326
## L(u_diff, 0:8)5                          0.23906
## L(u_diff, 0:8)6                          0.17426
## L(u_diff, 0:8)7                          0.92521
## L(u_diff, 0:8)8                          0.80372
## L(inf_diff, 0:8)0                        0.39050
## L(inf_diff, 0:8)1                        0.38939
## L(inf_diff, 0:8)2                        0.35472
## L(inf_diff, 0:8)3                        0.14597
## L(inf_diff, 0:8)4                        0.12998
## L(inf_diff, 0:8)5                        0.77133
## L(inf_diff, 0:8)6                        0.47404
## L(inf_diff, 0:8)7                        0.35220
## L(inf_diff, 0:8)8                        0.10751
## L(bnp_diff, 0:8)0                        0.13931
## L(bnp_diff, 0:8)1                        0.17524
## L(bnp_diff, 0:8)2                        0.57137
## L(bnp_diff, 0:8)3                        0.66448
## L(bnp_diff, 0:8)4                        0.25040
## L(bnp_diff, 0:8)5                        0.39339
## L(bnp_diff, 0:8)6                        0.09724 .
## L(bnp_diff, 0:8)7                        0.19535
## L(bnp_diff, 0:8)8                        0.36243
## L(LC20, 1)                               0.07909 .
## L(long_i, 1)                             0.37851
## L(inf, 1)                                0.41564
## L(bnp, 1)                                0.95138
```

```
## L(u, 1)                                   0.91986
## dummies_LC20_diffdummy_LC20_diff_2008Q4  0.00358 **
## dummies_LC20_diffdummy_LC20_diff_2008Q3  0.08792 .
## dummies_LC20_diffdummy_LC20_diff_2015    0.00182 **
## dummies_LC20_diffdummy_LC20_diff_2002Q3  0.38302
## dummies_LC20_diffdummy_LC20_diff_2011Q3  0.28397
## dummies_LC20_diffdummy_LC20_diff_2003Q1  0.09381 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0506 on 27 degrees of freedom
## Multiple R-squared:  0.8826, Adjusted R-squared:  0.6435
## F-statistic: 3.691 on 55 and 27 DF,  p-value: 0.0002233
```

We check for normal distribution in the error term

```
res_ecm1= ecm1$residuals

shapiro.test(res_ecm1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_ecm1
## W = 0.98876, p-value = 0.6926
```

```
jarque.bera.test(res_ecm1)
```

```
##
##  Jarque Bera Test
##
## data:  res_ecm1
## X-squared = 1.3919, df = 2, p-value = 0.4986
```

We fail to reject, so we can use the F and T-statistic. But not by a lot.

We will now remove insignificant parameters. We can test if they are significant by running an F-test. on the restricted model and unrestricted model.

```
    LC20_vars_1_ecm <- str_c("L(LC20_diff, 1:8)", c(3:8))
    long_i_vars_1_ecm <- str_c("L(long_i_diff, 0:8)", c(6:8))
    inf_vars_1_ecm <- str_c("L(inf_diff, 0:8)", c(4:8))
    bnp_vars_1_ecm <- str_c("L(bnp_diff, 0:8)", c(5:8))
    u_vars_1_ecm <- str_c("L(u_diff, 0:8)", c(4:8))



    vars_out_1_ecm <- c(bnp_vars_1_ecm, inf_vars_1_ecm, long_i_vars_1_ecm, LC20_vars_1_ecm, u_vars_1_ec
    vars_out_1_ecm
```

```
##  [1] "L(bnp_diff, 0:8)5"    "L(bnp_diff, 0:8)6"    "L(bnp_diff, 0:8)7"
##  [4] "L(bnp_diff, 0:8)8"    "L(inf_diff, 0:8)4"    "L(inf_diff, 0:8)5"
##  [7] "L(inf_diff, 0:8)6"    "L(inf_diff, 0:8)7"    "L(inf_diff, 0:8)8"
## [10] "L(long_i_diff, 0:8)6" "L(long_i_diff, 0:8)7" "L(long_i_diff, 0:8)8"
## [13] "L(LC20_diff, 1:8)3"   "L(LC20_diff, 1:8)4"   "L(LC20_diff, 1:8)5"
## [16] "L(LC20_diff, 1:8)6"   "L(LC20_diff, 1:8)7"   "L(LC20_diff, 1:8)8"
## [19] "L(u_diff, 0:8)4"      "L(u_diff, 0:8)5"      "L(u_diff, 0:8)6"
## [22] "L(u_diff, 0:8)7"      "L(u_diff, 0:8)8"
```

```r
    linearHypothesis(ecm1, vars_out_1_ecm, rep(0, length(vars_out_1_ecm)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(bnp_diff, 0:8)5 = 0
## L(bnp_diff, 0:8)6 = 0
## L(bnp_diff, 0:8)7 = 0
## L(bnp_diff, 0:8)8 = 0
## L(inf_diff, 0:8)4 = 0
## L(inf_diff, 0:8)5 = 0
## L(inf_diff, 0:8)6 = 0
## L(inf_diff, 0:8)7 = 0
## L(inf_diff, 0:8)8 = 0
## L(long_i_diff, 0:8)6 = 0
## L(long_i_diff, 0:8)7 = 0
## L(long_i_diff, 0:8)8 = 0
## L(LC20_diff,8)3 = 0
## L(LC20_diff,8)4 = 0
## L(LC20_diff,8)5 = 0
## L(LC20_diff,8)6 = 0
## L(LC20_diff,8)7 = 0
## L(LC20_diff,8)8 = 0
## L(u_diff, 0:8)4 = 0
## L(u_diff, 0:8)5 = 0
## L(u_diff, 0:8)6 = 0
## L(u_diff, 0:8)7 = 0
## L(u_diff, 0:8)8 = 0
##
## Model 1: restricted model
## Model 2: LC20_diff ~ L(LC20_diff, 1:8) + L(long_i_diff, 0:8) + L(u_diff,
##     0:8) + L(inf_diff, 0:8) + L(bnp_diff, 0:8) + L(LC20, 1) +
##     L(long_i, 1) + L(inf, 1) + L(bnp, 1) + L(u, 1) + dummies_LC20_diff
##
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1     50 0.115095
## 2     27 0.069131 23  0.045965 0.7805  0.725
```

We remove these lags

```r
ecm1_reduced1 <- dynlm(LC20_diff~L(LC20_diff, 1:2)+ L(long_i_diff, 0:5)+L(u_diff, 0:3)+ L(inf_diff, 0:3

summary(ecm1_reduced1)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(3), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1:2) + L(long_i_diff,
##     0:5) + L(u_diff, 0:3) + L(inf_diff, 0:3) + L(bnp_diff, 0:4) +
##     L(LC20, 1) + L(long_i, 1) + L(inf, 1) + L(bnp, 1) + L(u,
##     1) + dummies_LC20_diff)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.10249 -0.01847  0.00000  0.02672  0.07852
##
## Coefficients:
##                                          Estimate  Std. Error t value
## (Intercept)                             1.74201505  0.46373896   3.756
## L(LC20_diff, 1:2)1                      0.37790469  0.09747588   3.877
## L(LC20_diff, 1:2)2                      0.11893168  0.09710887   1.225
## L(long_i_diff, 0:5)0                    0.01587110  0.02373671   0.669
## L(long_i_diff, 0:5)1                    0.04112423  0.02654462   1.549
## L(long_i_diff, 0:5)2                    0.07978787  0.02608437   3.059
## L(long_i_diff, 0:5)3                    0.03700000  0.02450308   1.510
## L(long_i_diff, 0:5)4                    0.08521871  0.02506287   3.400
## L(long_i_diff, 0:5)5                    0.02838365  0.02536438   1.119
## L(u_diff, 0:3)0                         0.00940431  0.03285216   0.286
## L(u_diff, 0:3)1                         0.05810505  0.03091541   1.879
## L(u_diff, 0:3)2                        -0.01528868  0.02976403  -0.514
## L(u_diff, 0:3)3                         0.00791535  0.03062481   0.258
## L(inf_diff, 0:3)0                       0.01331651  0.01894565   0.703
## L(inf_diff, 0:3)1                       0.01940805  0.01946426   0.997
## L(inf_diff, 0:3)2                       0.01161751  0.01841805   0.631
## L(inf_diff, 0:3)3                       0.01407565  0.01852621   0.760
## L(bnp_diff, 0:4)0                      -0.00354786  0.00192751  -1.841
## L(bnp_diff, 0:4)1                      -0.00178424  0.00191001  -0.934
## L(bnp_diff, 0:4)2                       0.00043055  0.00197311   0.218
## L(bnp_diff, 0:4)3                       0.00114148  0.00204832   0.557
## L(bnp_diff, 0:4)4                      -0.00093406  0.00199992  -0.467
## L(LC20, 1)                             -0.23220571  0.04398810  -5.279
## L(long_i, 1)                           -0.07099458  0.02078439  -3.416
## L(inf, 1)                               0.00102277  0.01520952   0.067
## L(bnp, 1)                              -0.00006946  0.00069925  -0.099
## L(u, 1)                                -0.00745587  0.01036945  -0.719
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.35828293  0.06894246  -5.197
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.19370881  0.06476282  -2.991
## dummies_LC20_diffdummy_LC20_diff_2015    0.19945373  0.05434178   3.670
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.17551959  0.05643877  -3.110
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.18778920  0.06267155  -2.996
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.12659411  0.05571212  -2.272
##                                          Pr(>|t|)
## (Intercept)                              0.000430 ***
## L(LC20_diff, 1:2)1                       0.000294 ***
## L(LC20_diff, 1:2)2                       0.226094
## L(long_i_diff, 0:5)0                     0.506633
## L(long_i_diff, 0:5)1                     0.127273
## L(long_i_diff, 0:5)2                     0.003481 **
## L(long_i_diff, 0:5)3                     0.136979
## L(long_i_diff, 0:5)4                     0.001287 **
## L(long_i_diff, 0:5)5                     0.268171
## L(u_diff, 0:3)0                          0.775794
## L(u_diff, 0:3)1                          0.065681 .
## L(u_diff, 0:3)2                          0.609623
## L(u_diff, 0:3)3                          0.797052
## L(inf_diff, 0:3)0                        0.485207
## L(inf_diff, 0:3)1                        0.323241
```

```
## L(inf_diff, 0:3)2                          0.530903
## L(inf_diff, 0:3)3                          0.450759
## L(bnp_diff, 0:4)0                          0.071275 .
## L(bnp_diff, 0:4)1                          0.354461
## L(bnp_diff, 0:4)2                          0.828104
## L(bnp_diff, 0:4)3                          0.579685
## L(bnp_diff, 0:4)4                          0.642381
## L(LC20, 1)                               0.00000247 ***
## L(long_i, 1)                               0.001228 **
## L(inf, 1)                                  0.946639
## L(bnp, 1)                                  0.921247
## L(u, 1)                                    0.475285
## dummies_LC20_diffdummy_LC20_diff_2008Q4 0.00000331 ***
## dummies_LC20_diffdummy_LC20_diff_2008Q3   0.004212 **
## dummies_LC20_diffdummy_LC20_diff_2015     0.000563 ***
## dummies_LC20_diffdummy_LC20_diff_2002Q3   0.003010 **
## dummies_LC20_diffdummy_LC20_diff_2011Q3   0.004149 **
## dummies_LC20_diffdummy_LC20_diff_2003Q1   0.027152 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04797 on 53 degrees of freedom
## Multiple R-squared:  0.7988, Adjusted R-squared:  0.6774
## F-statistic: 6.577 on 32 and 53 DF,  p-value: 0.000000001105
```

```
res_ecm1_reduced1= residuals(ecm1_reduced1)
```

We will now remove insignificant parameters. We can test if they are significant by running an F-test. on the restricted model and unrestricted model.

```
    LC20_vars_2_ecm <- str_c("L(LC20_diff, 1:2)", c(2))
    long_i_vars_2_ecm <- str_c("L(long_i_diff, 0:5)", c(1,2,3,5))
    inf_vars_2_ecm <- str_c("L(inf_diff, 0:3)", c(0:3))
    bnp_vars_2_ecm <- str_c("L(bnp_diff, 0:4)", c(0:4))
    u_vars_2_ecm <- str_c("L(u_diff, 0:3)", c(2:3))

    vars_out_2_ecm <- c(bnp_vars_2_ecm, inf_vars_2_ecm, long_i_vars_2_ecm, LC20_vars_2_ecm,u_vars_2_ecm)
    vars_out_2_ecm
```

```
##  [1] "L(bnp_diff, 0:4)0"    "L(bnp_diff, 0:4)1"    "L(bnp_diff, 0:4)2"
##  [4] "L(bnp_diff, 0:4)3"    "L(bnp_diff, 0:4)4"    "L(inf_diff, 0:3)0"
##  [7] "L(inf_diff, 0:3)1"    "L(inf_diff, 0:3)2"    "L(inf_diff, 0:3)3"
## [10] "L(long_i_diff, 0:5)1" "L(long_i_diff, 0:5)2" "L(long_i_diff, 0:5)3"
## [13] "L(long_i_diff, 0:5)5" "L(LC20_diff, 1:2)2"   "L(u_diff, 0:3)2"
## [16] "L(u_diff, 0:3)3"
```

```
    linearHypothesis(ecm1_reduced1, vars_out_2_ecm, rep(0, length(vars_out_2_ecm)))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L(bnp_diff, 0:4)0 = 0
## L(bnp_diff, 0:4)1 = 0
## L(bnp_diff, 0:4)2 = 0
## L(bnp_diff, 0:4)3 = 0
## L(bnp_diff, 0:4)4 = 0
```

```
## L(inf_diff, 0:3)0 = 0
## L(inf_diff, 0:3)1 = 0
## L(inf_diff, 0:3)2 = 0
## L(inf_diff, 0:3)3 = 0
## L(long_i_diff, 0:5)1 = 0
## L(long_i_diff, 0:5)2 = 0
## L(long_i_diff, 0:5)3 = 0
## L(long_i_diff, 0:5)5 = 0
## L(LC20_diff,2)2 = 0
## L(u_diff, 0:3)2 = 0
## L(u_diff, 0:3)3 = 0
##
## Model 1: restricted model
## Model 2: LC20_diff ~ L(LC20_diff, 1:2) + L(long_i_diff, 0:5) + L(u_diff,
##     0:3) + L(inf_diff, 0:3) + L(bnp_diff, 0:4) + L(LC20, 1) +
##     L(long_i, 1) + L(inf, 1) + L(bnp, 1) + L(u, 1) + dummies_LC20_diff
##
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     69 0.17110
## 2     53 0.12197 16  0.049134 1.3344 0.2121
```

We can now remove the lags

```
ecm1_reduced2 <- dynlm(LC20_diff~L(LC20_diff, 1)+ L(long_i_diff,4)+ L(LC20,1) + L(long_i,1)+ L(inf,1)+L

summary(ecm1_reduced2)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff,
##     4) + L(LC20, 1) + L(long_i, 1) + L(inf, 1) + L(bnp, 1) +
##     L(u, 1) + dummies_LC20_diff)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.103298 -0.027071  0.001718  0.028772  0.120779
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                             0.6489148  0.2737462   2.370
## L(LC20_diff, 1)                         0.3136766  0.0721621   4.347
## L(long_i_diff, 4)                       0.0647013  0.0197272   3.280
## L(LC20, 1)                             -0.1780332  0.0336884  -5.285
## L(long_i, 1)                           -0.0253787  0.0112880  -2.248
## L(inf, 1)                              -0.0014806  0.0096309  -0.154
## L(bnp, 1)                               0.0010812  0.0004862   2.224
## L(u, 1)                                 0.0074219  0.0068371   1.086
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.3056379  0.0557376  -5.484
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.1399927  0.0536476  -2.609
## dummies_LC20_diffdummy_LC20_diff_2015    0.1525339  0.0510645   2.987
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.1637867  0.0508980  -3.218
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.1943903  0.0545483  -3.564
```

```
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.1743013  0.0519990  -3.352
##                                            Pr(>|t|)
## (Intercept)                                0.020407 *
## L(LC20_diff, 1)                         0.000043986 ***
## L(long_i_diff, 4)                          0.001594 **
## L(LC20, 1)                              0.000001255 ***
## L(long_i, 1)                               0.027577 *
## L(inf, 1)                                  0.878246
## L(bnp, 1)                                  0.029262 *
## L(u, 1)                                    0.281257
## dummies_LC20_diffdummy_LC20_diff_2008Q4 0.000000569 ***
## dummies_LC20_diffdummy_LC20_diff_2008Q3    0.010995 *
## dummies_LC20_diffdummy_LC20_diff_2015      0.003833 **
## dummies_LC20_diffdummy_LC20_diff_2002Q3    0.001927 **
## dummies_LC20_diffdummy_LC20_diff_2011Q3    0.000649 ***
## dummies_LC20_diffdummy_LC20_diff_2003Q1    0.001274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04961 on 73 degrees of freedom
## Multiple R-squared:  0.7047, Adjusted R-squared:  0.6521
## F-statistic:  13.4 on 13 and 73 DF,  p-value: 0.00000000000001557
```

```
res_ecm1_reduced2= residuals(ecm1_reduced2)
```

We will now create our restricted model. to see if the "niveau parameters" are significant. We can now set up our hypothesis $H_0 : \theta_1 = \phi_1 = \phi_2 = 0$ (No Long Run Relation) And our $H_1 : \theta_1 \neq \phi_1 \neq \phi_2 = 0$ (long-Run-Relationship)

```
ecm1_restrict <- dynlm(LC20_diff~L(LC20_diff, 1)+ L(long_i_diff, c(4))+dummies_LC20_diff)

anova(ecm1_restrict, ecm1_reduced2)
```

```
## Analysis of Variance Table
##
## Model 1: LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff, c(4)) + dummies_LC20_diff
## Model 2: LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff, 4) + L(LC20, 1) +
##     L(long_i, 1) + L(inf, 1) + L(bnp, 1) + L(u, 1) + dummies_LC20_diff
##   Res.Df     RSS Df Sum of Sq      F      Pr(>F)
## 1     78 0.27400
## 2     73 0.17968  5  0.094324 7.6642 0.000007777 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With 4 explaining variables and on a 5% significans level we have a lower bound critical value on 2.86 and an upper bound critical value on 3.79 So we can now reject H_0 and Accept H_1 that there is a Long-run relationship between Inflation, BNP and the long interest rate on The growth rate of C20 indeks.

4.1 Long run estimate

We will use the following function:

$$-\frac{\phi_1}{\theta_1}$$

We start by finding θ_1 and ϕ_1. For consumption, Investment and BNP. Then we wil use the above function to calculate the long run estimater.

```r
ecm1_reduced2
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff,
##     4) + L(LC20, 1) + L(long_i, 1) + L(inf, 1) + L(bnp, 1) +
##     L(u, 1) + dummies_LC20_diff)
##
## Coefficients:
##                             (Intercept)
##                                0.648915
##                         L(LC20_diff, 1)
##                                0.313677
##                       L(long_i_diff, 4)
##                                0.064701
##                              L(LC20, 1)
##                               -0.178033
##                            L(long_i, 1)
##                               -0.025379
##                               L(inf, 1)
##                               -0.001481
##                               L(bnp, 1)
##                                0.001081
##                                 L(u, 1)
##                                0.007422
## dummies_LC20_diffdummy_LC20_diff_2008Q4
##                               -0.305638
## dummies_LC20_diffdummy_LC20_diff_2008Q3
##                               -0.139993
##   dummies_LC20_diffdummy_LC20_diff_2015
##                                0.152534
## dummies_LC20_diffdummy_LC20_diff_2002Q3
##                               -0.163787
## dummies_LC20_diffdummy_LC20_diff_2011Q3
##                               -0.194390
## dummies_LC20_diffdummy_LC20_diff_2003Q1
##                               -0.174301
```

```r
phi_bnp=ecm1_reduced2$coefficients[7]; phi_bnp
```

```
##   L(bnp, 1)
## 0.001081195
```

```r
phi_long_i= ecm1_reduced2$coefficients[5]; phi_long_i
```

```
## L(long_i, 1)
##  -0.02537873
```

```r
phi_inf= ecm1_reduced2$coefficients[6]; phi_inf
```

```
##    L(inf, 1)
## -0.001480576
```

```r
phi_u= ecm1_reduced2$coefficients[8]; phi_u
```

```
##     L(u, 1)
## 0.007421869
```

```r
theta_1=ecm1_reduced2$coefficients[4]; theta_1
```

```
## L(LC20, 1)
## -0.1780332
```

```r
lr_LC20_bnp= -phi_bnp/theta_1; lr_LC20_bnp
```

```
##   L(bnp, 1)
## 0.006072996
```

```r
lr_LC20_long_i= -phi_long_i/theta_1; lr_LC20_long_i
```

```
## L(long_i, 1)
##   -0.1425505
```

```r
lr_LC20_inf= -phi_inf/theta_1; lr_LC20_inf
```

```
##    L(inf, 1)
## -0.008316289
```

```r
lr_LC20_u= -phi_u/theta_1; lr_LC20_u
```

```
##    L(u, 1)
## 0.04168811
```

We can now see the long run relationship between the macroeconmoic variables and LC20 (Growth rate in C20 indeks)

We will now test if our long-run estimators are statistical significant.

We know that you cannot just divide the standard error of one variable by another if they are statistically dependent.

We can compute the statistical significance of our long-run estimator by performing the following:

4.2 LR_LC20_bnp

```r
nlWaldtest(ecm1_reduced2, c("b[7]/b[4]=0"))
```

```
##
##  Wald Chi-square test of a restriction on model parameters
##
## data:  ecm1_reduced2
## Chisq = 6.4208, df = 1, p-value = 0.01128
```

4.3 LR_LC20_long_i

```r
nlWaldtest(ecm1_reduced2, c("b[5]/b[4]=0"))
```

```
##
##  Wald Chi-square test of a restriction on model parameters
##
## data:  ecm1_reduced2
## Chisq = 5.9956, df = 1, p-value = 0.01434
```

4.4 LR_LC20_inf

```
nlWaldtest(ecm1_reduced2, c("b[6]/b[4]=0"))
```

```
##
##  Wald Chi-square test of a restriction on model parameters
##
## data:  ecm1_reduced2
## Chisq = 0.023881, df = 1, p-value = 0.8772
```

4.5 LR_LC20_u

```
nlWaldtest(ecm1_reduced2, c("b[8]/b[4]=0"))
```

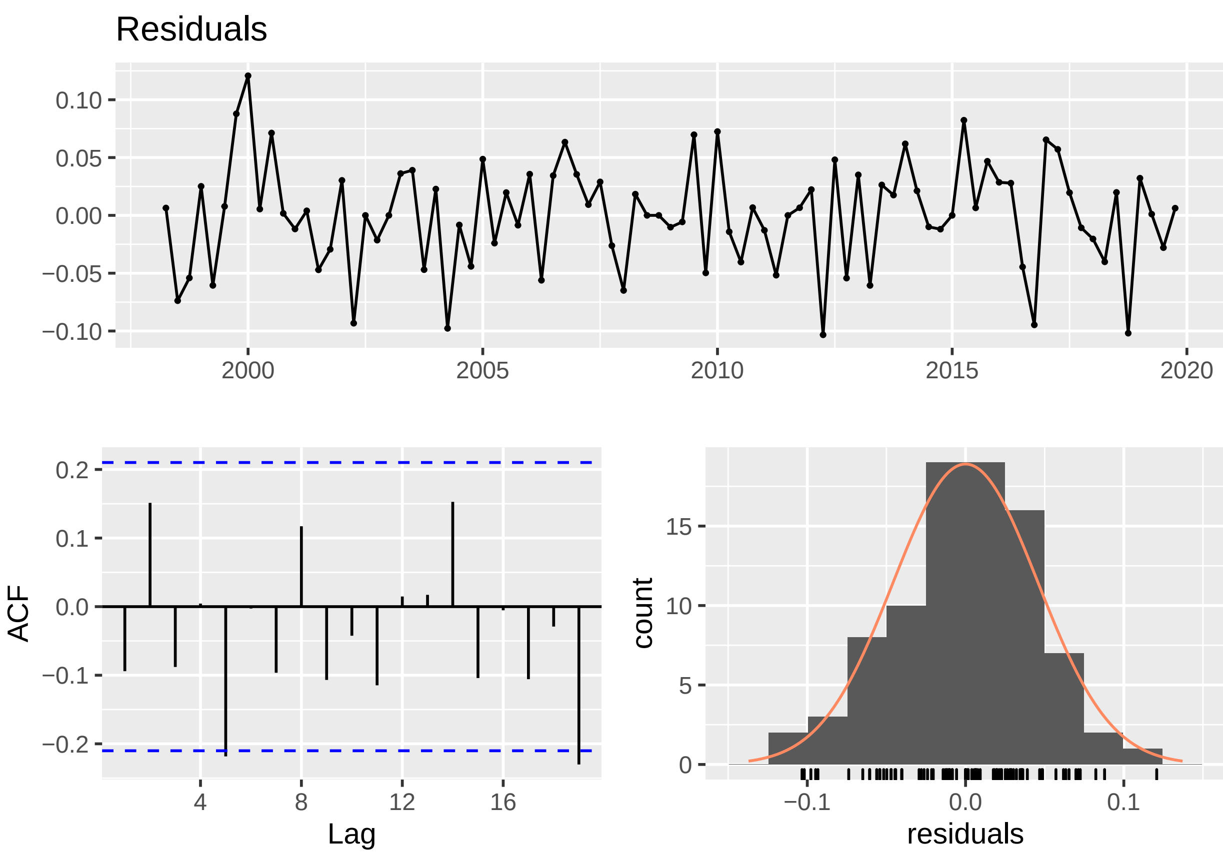
```
##
##  Wald Chi-square test of a restriction on model parameters
##
## data:  ecm1_reduced2
## Chisq = 1.0643, df = 1, p-value = 0.3022
```

5 Diagnostics

5.1 ARDL -Bounds test

5.1.1 Check Residuals

```
checkresiduals(ecm1_reduced2)
```



```
##
##  Breusch-Godfrey test for serial correlation of order up to 17
##
## data:  Residuals
## LM test = 27.343, df = 17, p-value = 0.05322
```

5.1.2 Serial correlation

```r
Box.test(res_ecm1_reduced2, lag=20, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  res_ecm1_reduced2
## X-squared = 24.366, df = 20, p-value = 0.2268
```

```r
Box.test(res_ecm1_reduced2, lag=16, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  res_ecm1_reduced2
## X-squared = 16.694, df = 16, p-value = 0.4056
```

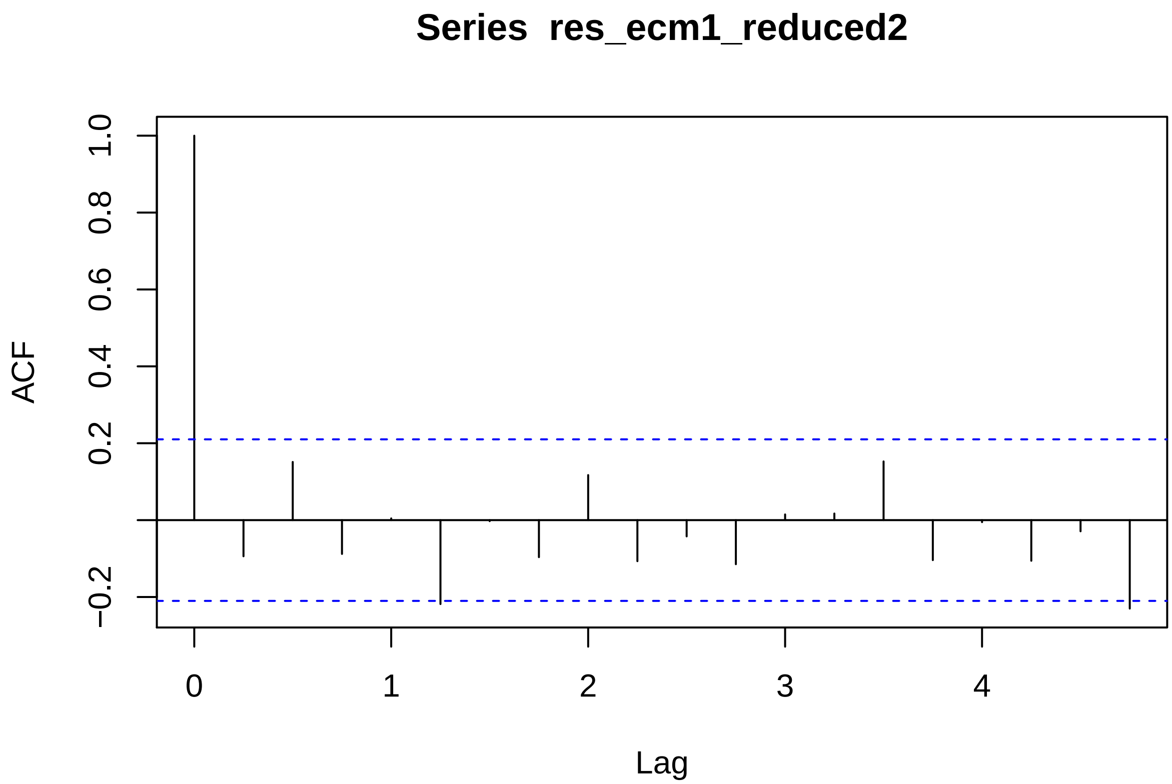
```r
Box.test(res_ecm1_reduced2, lag=12, type = "Ljung")
```

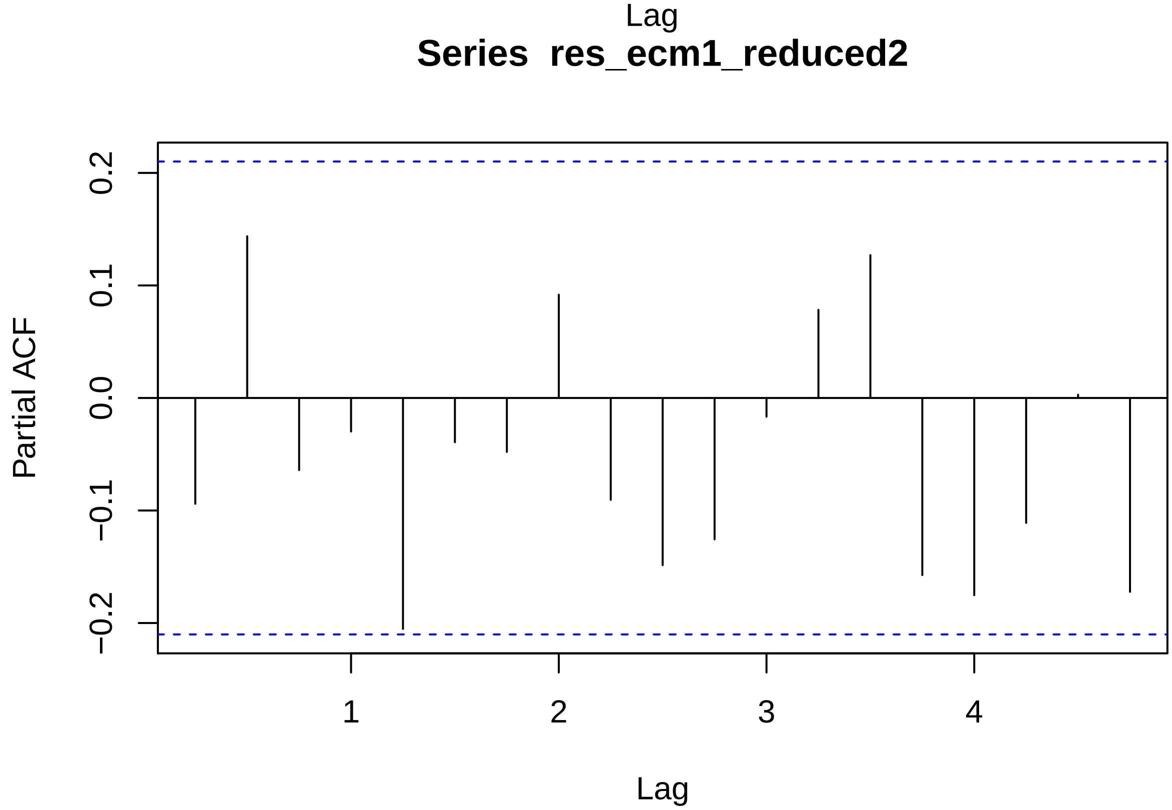
```
##
##  Box-Ljung test
##
## data:  res_ecm1_reduced2
## X-squared = 13.021, df = 12, p-value = 0.3675
```

Looks fine

We will just check the ACF and PACF

```r
acf(res_ecm1_reduced2); pacf(res_ecm1_reduced2)
```

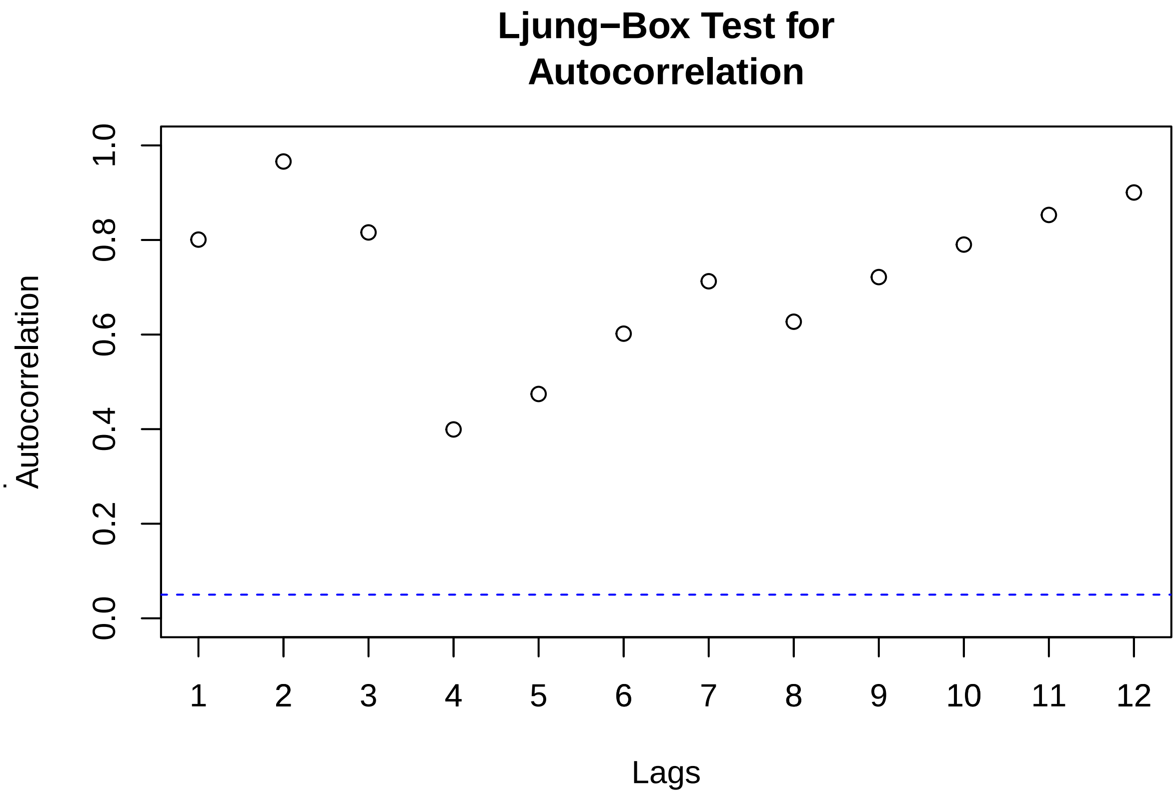




Also looks fine

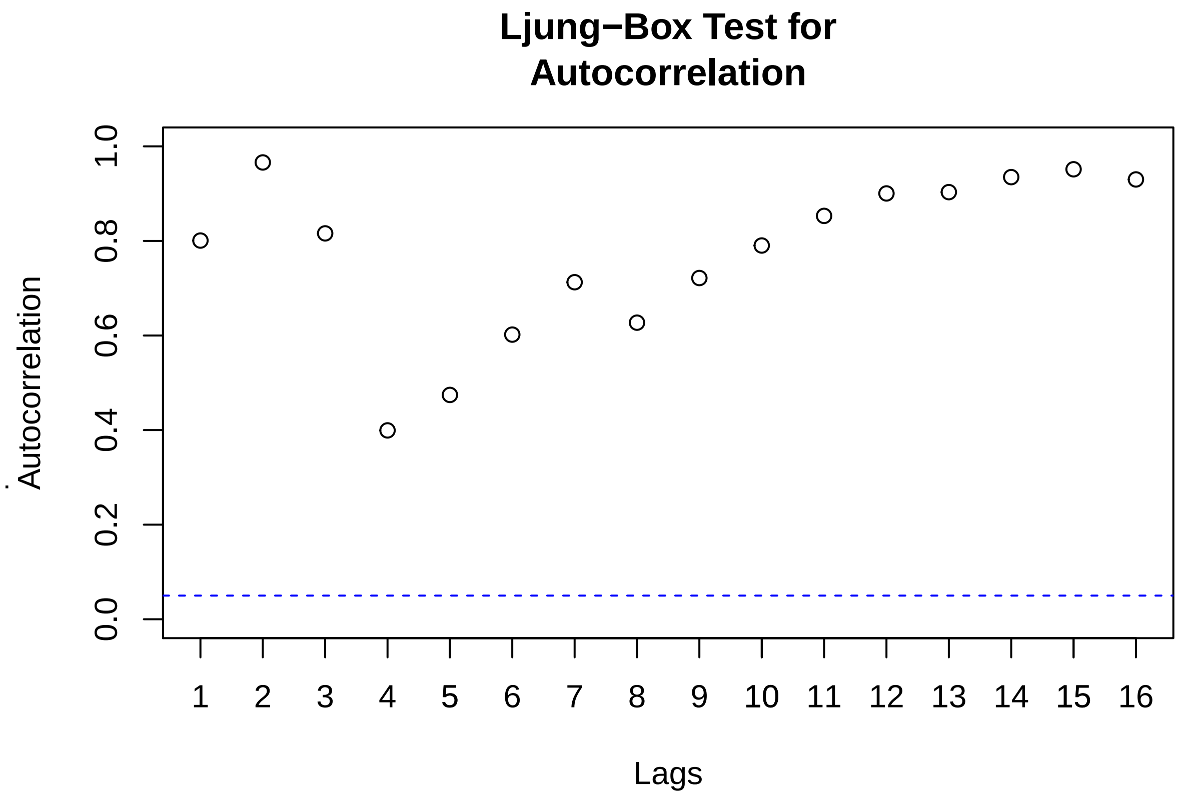
5.1.3 ARCH -Test

```
arpdiag((res_ecm1_reduced2^2), lags=12); Box.test(res_ecm1_reduced2^2, lag = 12, type = "Ljung-Box")
```



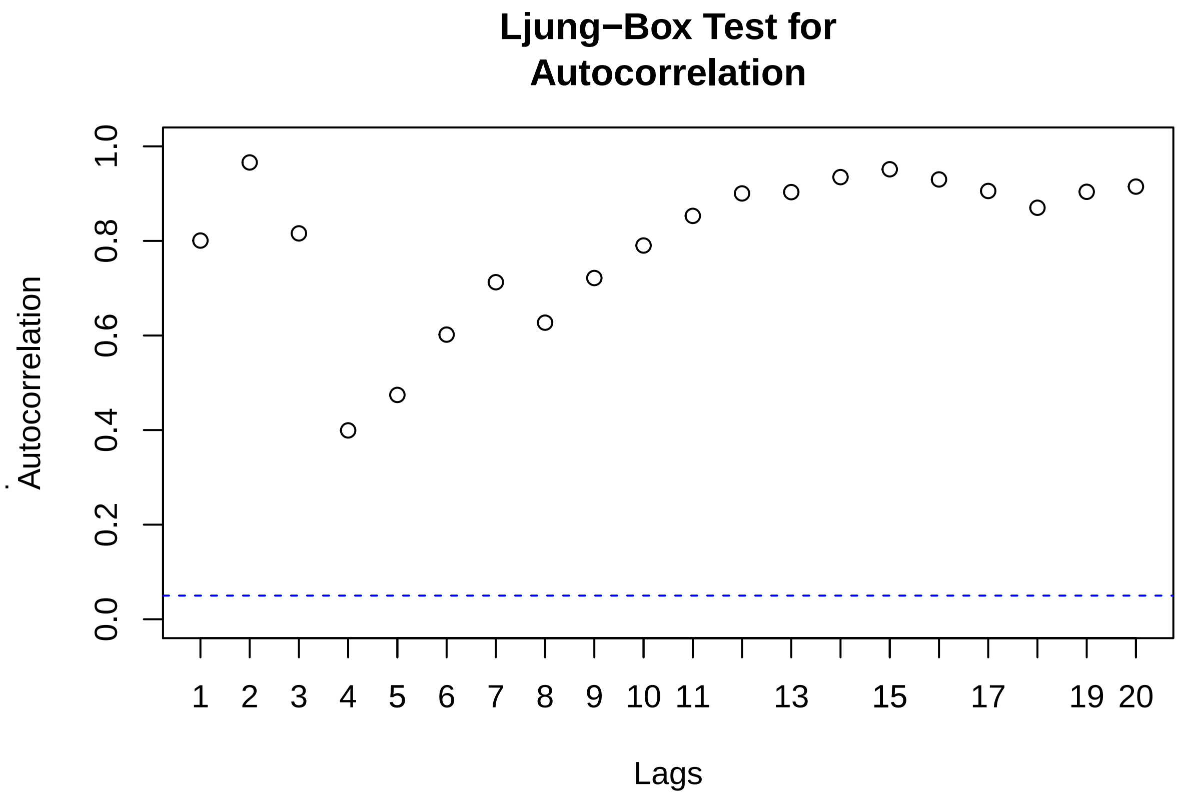

```
##
##  Box-Ljung test
##
## data:  res_ecm1_reduced2^2
## X-squared = 6.2955, df = 12, p-value = 0.9005
```

```
arpdiag((res_ecm1_reduced2^2), lags=16); Box.test(res_ecm1_reduced2^2, lag = 16, type = "Ljung-Box")
```




```
##
##  Box-Ljung test
##
## data:  res_ecm1_reduced2^2
## X-squared = 8.5709, df = 16, p-value = 0.93
```

```r
arpdiag((res_ecm1_reduced2^2), lags=20); Box.test(res_ecm1_reduced2^2, lag = 20, type = "Ljung-Box")
```



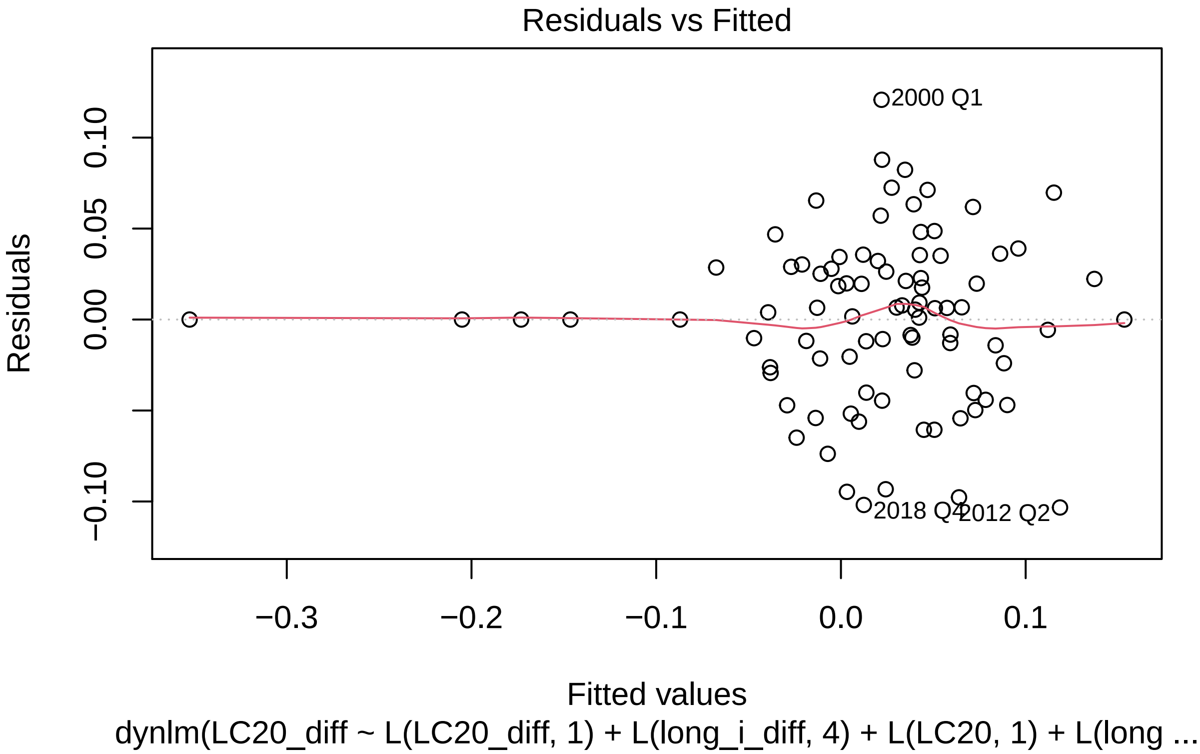

```
##
##  Box-Ljung test
##
## data:  res_ecm1_reduced2^2
## X-squared = 12.034, df = 20, p-value = 0.9149
```

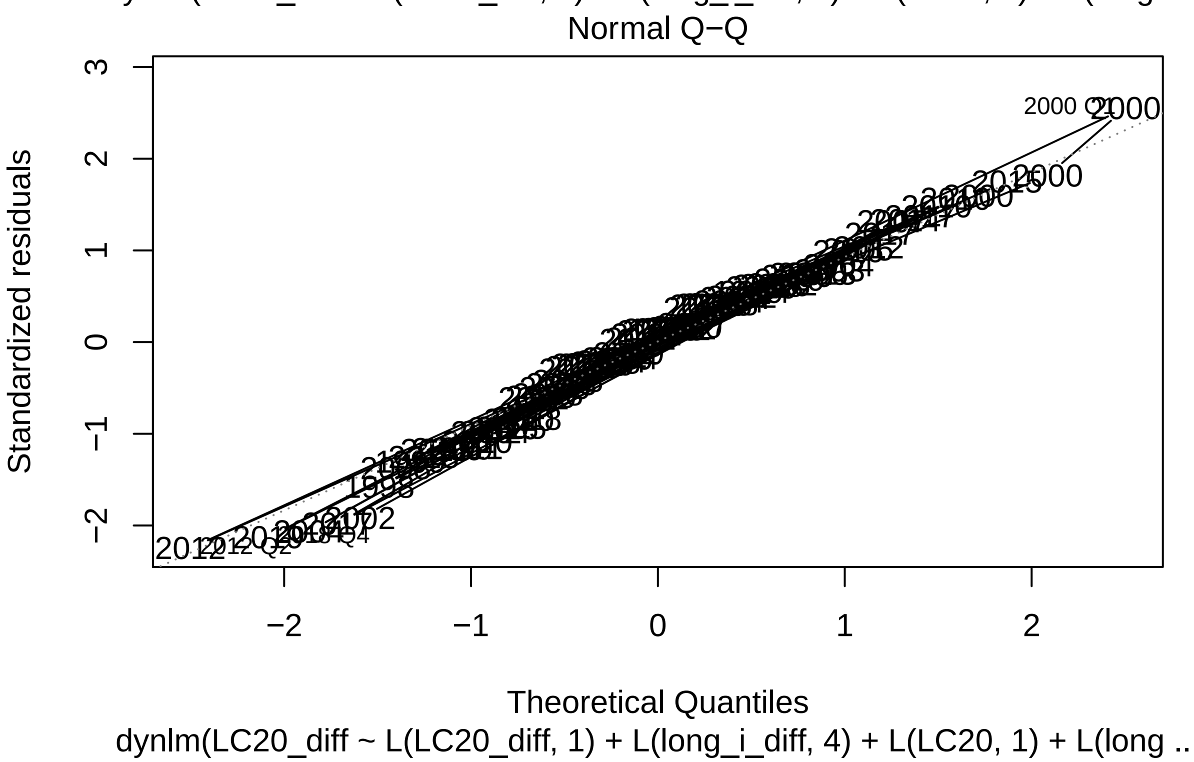
Looks fine

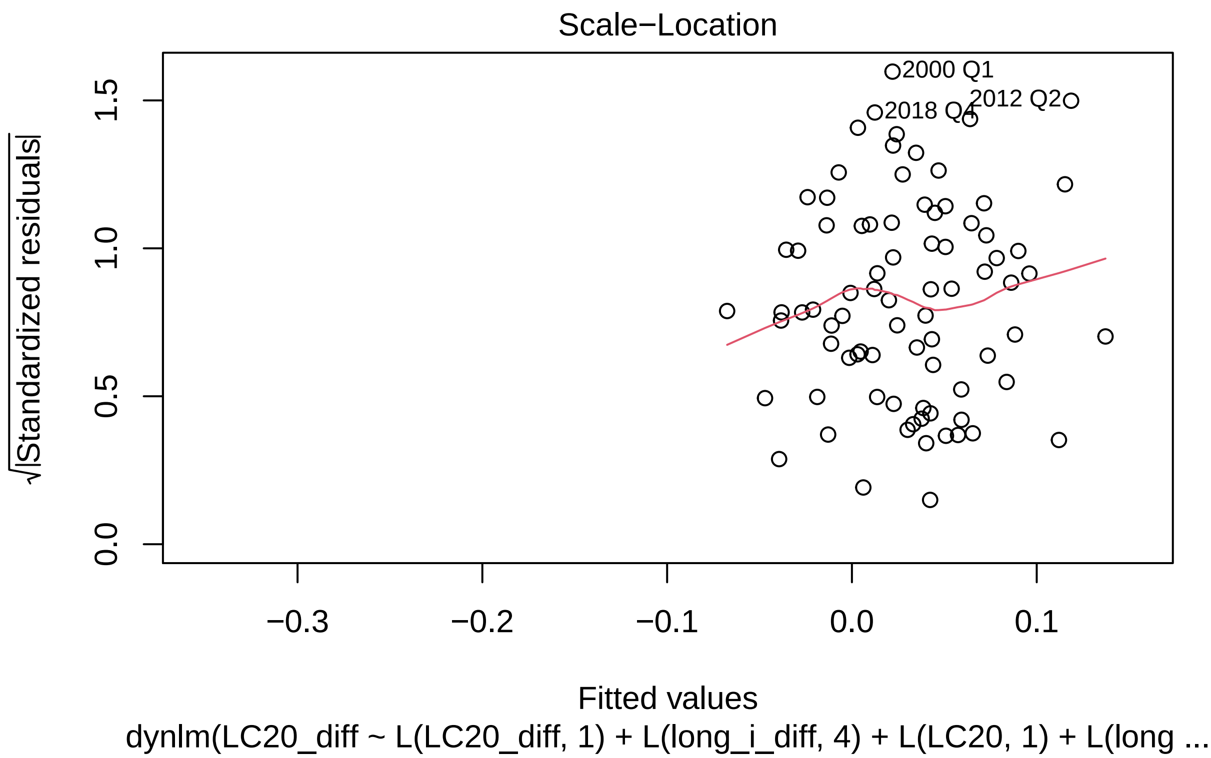
5.1.4 We can also look for the regular OLS assumptions

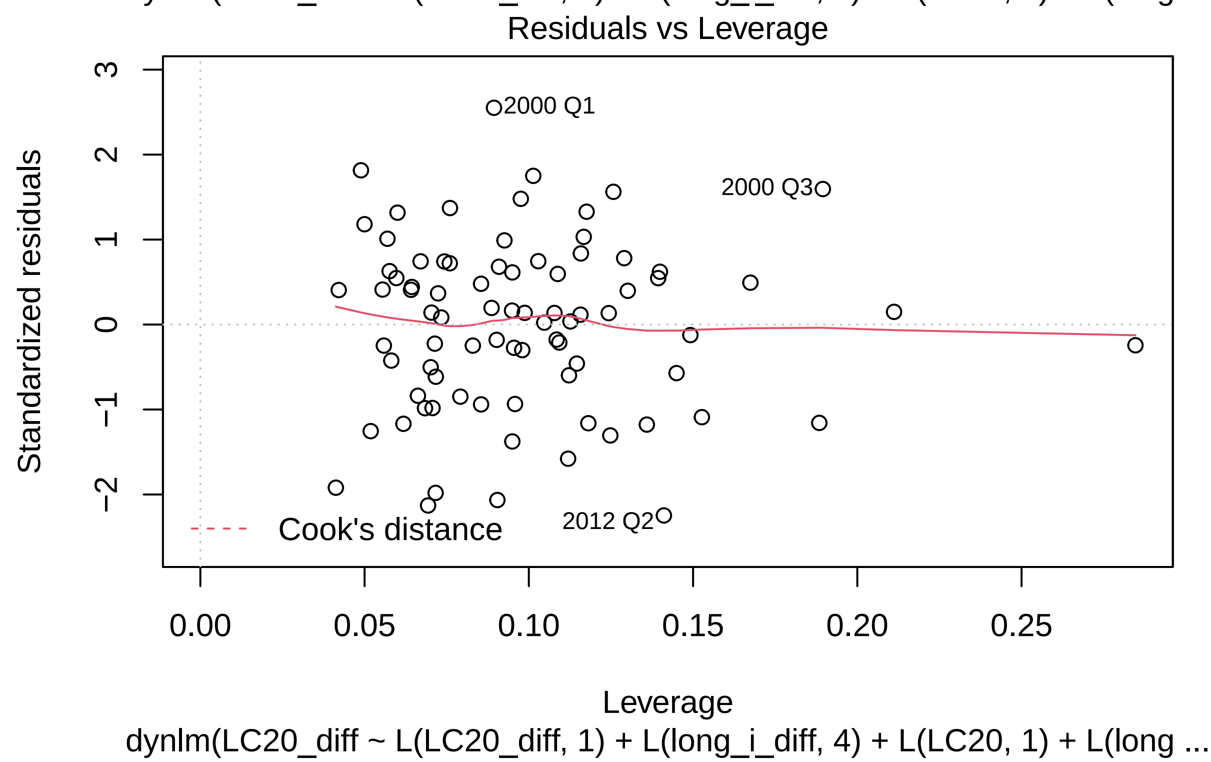
```
plot(ecm1_reduced2)
```

```
## Warning: not plotting observations with leverage one:
##   18, 20, 42, 43, 54, 68
```


Residuals vs Fitted
Residuals
0.10
0.05
0.00
−0.10
2000 Q1
2018 Q4
2012 Q2
−0.3
−0.2
−0.1
0.0
0.1
Fitted values
dynlm(LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff, 4) + L(LC20, 1) + L(long ...


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
2000 Q1
2012 Q2
−2
−1
0
1
2
Theoretical Quantiles
dynlm(LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff, 4) + L(LC20, 1) + L(long ...





5.1.4.1 RESET test

We do this to test if the model os linear in parameters.

The Residuals vs. Fitted shows that there should be a linear relationship.

```
y_hat_ecm=ecm1_reduced2$fitted.values
y_hat_sq= y_hat_ecm^2
```

```r
y_hat_kub= y_hat_ecm^3

ecm1_reduced2_reset <- dynlm(LC20_diff~L(LC20_diff, 1)+ L(long_i_diff,4)+ L(LC20,1) + L(long_i,1)+ L(in

summary(ecm1_reduced2_reset)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff,
##     4) + L(LC20, 1) + L(long_i, 1) + L(inf, 1) + L(bnp, 1) +
##     L(u, 1) + dummies_LC20_diff + y_hat_sq + y_hat_kub)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.102002 -0.024689  0.001659  0.032063  0.119521
##
## Coefficients:
##                                           Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                              0.7096374  0.2998977   2.366  0.020701
## L(LC20_diff, 1)                          0.3354173  0.0838913   3.998  0.000154
## L(long_i_diff, 4)                        0.0715209  0.0238070   3.004  0.003677
## L(LC20, 1)                              -0.1946488  0.0465616  -4.180 0.0000818
## L(long_i, 1)                            -0.0277415  0.0123016  -2.255  0.027211
## L(inf, 1)                               -0.0012635  0.0099158  -0.127  0.898964
## L(bnp, 1)                                0.0011783  0.0005275   2.234  0.028661
## L(u, 1)                                  0.0082767  0.0071772   1.153  0.252703
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.6516190  2.6093738  -0.250  0.803523
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.1606893  0.2554392  -0.629  0.531322
## dummies_LC20_diffdummy_LC20_diff_2015    0.1920347  0.0959816   2.001  0.049244
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.2001039  0.3853188  -0.519  0.605153
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.2574013  0.6013621  -0.428  0.669925
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.1867745  0.0855525  -2.183  0.032332
## y_hat_sq                                -0.3049635  5.4542163  -0.056  0.955568
## y_hat_kub                               -8.8229740 45.4388543  -0.194  0.846595
##
## (Intercept)                             *
## L(LC20_diff, 1)                         ***
## L(long_i_diff, 4)                       **
## L(LC20, 1)                              ***
## L(long_i, 1)                            *
## L(inf, 1)
## L(bnp, 1)                               *
## L(u, 1)
## dummies_LC20_diffdummy_LC20_diff_2008Q4
## dummies_LC20_diffdummy_LC20_diff_2008Q3
## dummies_LC20_diffdummy_LC20_diff_2015   *
## dummies_LC20_diffdummy_LC20_diff_2002Q3
## dummies_LC20_diffdummy_LC20_diff_2011Q3
## dummies_LC20_diffdummy_LC20_diff_2003Q1 *
## y_hat_sq
## y_hat_kub
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05021 on 71 degrees of freedom
## Multiple R-squared:  0.7058, Adjusted R-squared:  0.6437
## F-statistic: 11.36 on 15 and 71 DF,  p-value: 0.0000000000001859
```

Dermed er den linrar n?r de uafh?ngige variabler stiger Vi ved allerede den er linear over tid da den er stationear.

5.1.4.2 Testing for Heteroskedasticity Because we can not use the bp.test for the dynlm function we will perform my dynamic regression using simple lm() function. For that i need to extract the lags of my variables

```r
summary(ecm1_reduced2)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff,
##     4) + L(LC20, 1) + L(long_i, 1) + L(inf, 1) + L(bnp, 1) +
##     L(u, 1) + dummies_LC20_diff)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.103298 -0.027071  0.001718  0.028772  0.120779
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                              0.6489148  0.2737462   2.370
## L(LC20_diff, 1)                          0.3136766  0.0721621   4.347
## L(long_i_diff, 4)                        0.0647013  0.0197272   3.280
## L(LC20, 1)                              -0.1780332  0.0336884  -5.285
## L(long_i, 1)                            -0.0253787  0.0112880  -2.248
## L(inf, 1)                               -0.0014806  0.0096309  -0.154
## L(bnp, 1)                                0.0010812  0.0004862   2.224
## L(u, 1)                                  0.0074219  0.0068371   1.086
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.3056379  0.0557376  -5.484
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.1399927  0.0536476  -2.609
## dummies_LC20_diffdummy_LC20_diff_2015    0.1525339  0.0510645   2.987
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.1637867  0.0508980  -3.218
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.1943903  0.0545483  -3.564
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.1743013  0.0519990  -3.352
##                                            Pr(>|t|)
## (Intercept)                                0.020407 *
## L(LC20_diff, 1)                          0.000043986 ***
## L(long_i_diff, 4)                          0.001594 **
## L(LC20, 1)                               0.000001255 ***
## L(long_i, 1)                               0.027577 *
## L(inf, 1)                                  0.878246
## L(bnp, 1)                                  0.029262 *
## L(u, 1)                                    0.281257
## dummies_LC20_diffdummy_LC20_diff_2008Q4 0.000000569 ***
```

```
## dummies_LC20_diffdummy_LC20_diff_2008Q3    0.010995 *
## dummies_LC20_diffdummy_LC20_diff_2015      0.003833 **
## dummies_LC20_diffdummy_LC20_diff_2002Q3    0.001927 **
## dummies_LC20_diffdummy_LC20_diff_2011Q3    0.000649 ***
## dummies_LC20_diffdummy_LC20_diff_2003Q1    0.001274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04961 on 73 degrees of freedom
## Multiple R-squared:  0.7047, Adjusted R-squared:  0.6521
## F-statistic:  13.4 on 13 and 73 DF,  p-value: 0.00000000000001557
```

```
##For C20

LC20_1 <- lagged(LC20,1)
LC20_d <- LC20 - lagged(LC20,1)
LC20_d1 <- lagged(LC20_d, 1)

##For Long_i
long_i_1 <- lagged(long_i,1)
long_i_d <- long_i - lagged(long_i,1)
long_i_d4 <- lagged(long_i_d, 4)

##for Inflation

inf_1 <- lagged(inf,1)


##For BNP

bnp_1 <- lagged(bnp,1)


##For Unemployment

u_1 <- lagged(u,1)



##We make them for the time periode we use.

mydata <- cbind( LC20_1, LC20_d1, LC20_d,long_i_1,long_i_d, long_i_d4, inf_1, bnp_1, u_1, dummy_LC20_di
mydata <- ts(mydata, start=c(1997,1), freq=4); data <- na.omit(mydata)
```

```
##We create the lm function as bptest cant work with dynlm so the model i taken from the strucchange pa

model_struc= lm(LC20_d~ LC20_d1+ long_i_d4+ LC20_1+ long_i_1+ inf_1+ bnp_1+ u_1+ dummy_LC20_diff_2008Q3
summary(model_struc)
```

```
##
## Call:
## lm(formula = LC20_d ~ LC20_d1 + long_i_d4 + LC20_1 + long_i_1 +
##     inf_1 + bnp_1 + u_1 + dummy_LC20_diff_2008Q3 + dummy_LC20_diff_2008Q4 +
##     dummy_LC20_diff_2015 + dummy_LC20_diff_2002Q3 + dummy_LC20_diff_2011Q3 +
##     dummy_LC20_diff_2003Q1)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.103298 -0.027071  0.001718  0.028772  0.120779
##
## Coefficients:
##                         Estimate Std. Error t value    Pr(>|t|)
## (Intercept)            0.6489148  0.2737462   2.370    0.020407 *
## LC20_d1                0.3136766  0.0721621   4.347 0.000043986 ***
## long_i_d4              0.0647013  0.0197272   3.280    0.001594 **
## LC20_1                -0.1780332  0.0336884  -5.285 0.000001255 ***
## long_i_1              -0.0253787  0.0112880  -2.248    0.027577 *
## inf_1                 -0.0014806  0.0096309  -0.154    0.878246
## bnp_1                  0.0010812  0.0004862   2.224    0.029262 *
## u_1                    0.0074219  0.0068371   1.086    0.281257
## dummy_LC20_diff_2008Q3 -0.1399927  0.0536476  -2.609    0.010995 *
## dummy_LC20_diff_2008Q4 -0.3056379  0.0557376  -5.484 0.000000569 ***
## dummy_LC20_diff_2015    0.1525339  0.0510645   2.987    0.003833 **
## dummy_LC20_diff_2002Q3 -0.1637867  0.0508980  -3.218    0.001927 **
## dummy_LC20_diff_2011Q3 -0.1943903  0.0545483  -3.564    0.000649 ***
## dummy_LC20_diff_2003Q1 -0.1743013  0.0519990  -3.352    0.001274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04961 on 73 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.7047, Adjusted R-squared:  0.6521
## F-statistic:  13.4 on 13 and 73 DF,  p-value: 0.00000000000001557
```

```
bptest(model_struc)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_struc
## BP = 8.549, df = 13, p-value = 0.8061
```

We fail to reject that the model is homoskedastistic when the independent variables is changing.

To look for heteroskedasticity when we look acrros time we use ARCH.

5.1.5 Normality

```
#First reduction
shapiro.test(res_ecm1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_ecm1
## W = 0.98876, p-value = 0.6926
```

```
jarque.bera.test(res_ecm1)
```

```
##
##  Jarque Bera Test
##
```

```
## data:  res_ecm1
## X-squared = 1.3919, df = 2, p-value = 0.4986
```

```
#Second reduction
shapiro.test(res_ecm1_reduced1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_ecm1_reduced1
## W = 0.98341, p-value = 0.3384
```

```
jarque.bera.test(res_ecm1_reduced1)
```

```
##
##  Jarque Bera Test
##
## data:  res_ecm1_reduced1
## X-squared = 1.8322, df = 2, p-value = 0.4001
```

```
#Third reduction
jarque.bera.test(res_ecm1_reduced2)
```

```
##
##  Jarque Bera Test
##
## data:  res_ecm1_reduced2
## X-squared = 0.41971, df = 2, p-value = 0.8107
```

```
shapiro.test(res_ecm1_reduced2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_ecm1_reduced2
## W = 0.98554, p-value = 0.4463
```

5.1.6 Change in mean

```
break_points= breakpoints(LC20_diff~1)
summary(break_points)
```

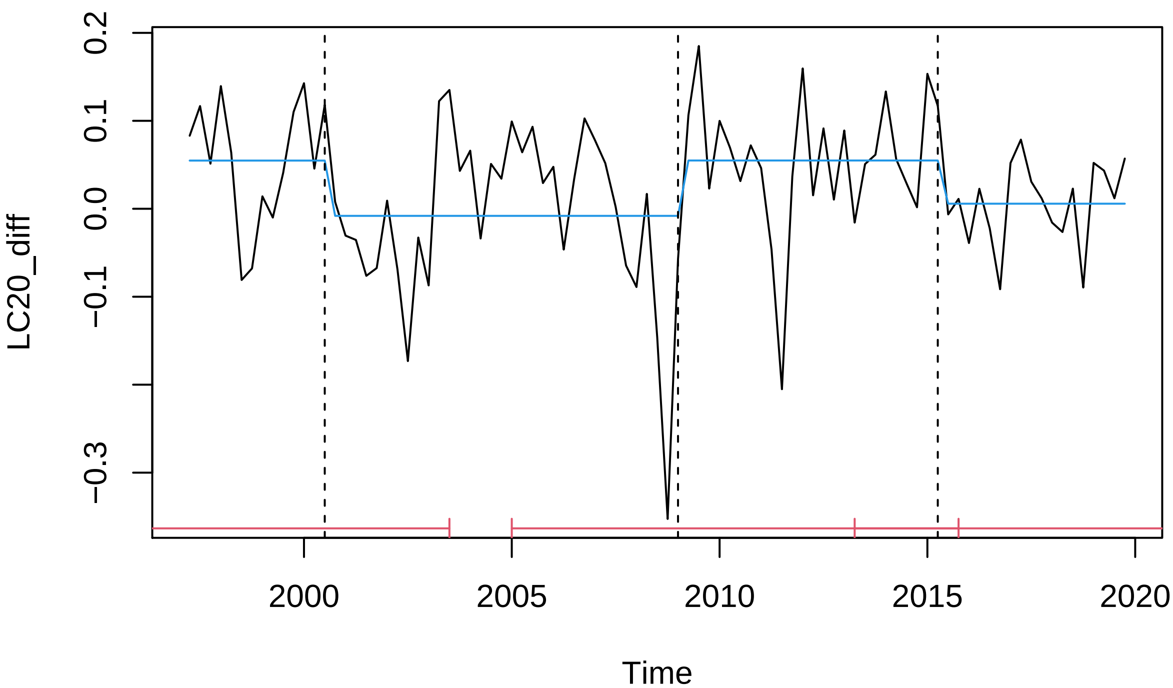
```
##
##   Optimal (m+1)-segment partition:
##
## Call:
## breakpoints.formula(formula = LC20_diff ~ 1)
##
## Breakpoints at observation number:
##
## m = 1   14
## m = 2   14    48
## m = 3   14    48    73
## m = 4   14    48    73
## m = 5   14 27 41 58 73
##
## Corresponding to breakdates:
```

```
##
## m = 1   2000(3)
## m = 2   2000(3)         2009(1)
## m = 3   2000(3)         2009(1)         2015(2)
## m = 4   2000(3)         2009(1)         2015(2)
## m = 5   2000(3) 2003(4) 2007(2) 2011(3) 2015(2)
##
## Fit:
##
## m   0         1         2         3         4         5
## RSS    0.6371    0.6189    0.5848    0.5595    0.5469    0.5290
## BIC -184.2506 -177.8668 -174.0019 -168.9929 -162.0502 -156.0568
```

```r
plot(LC20_diff); lines(fitted(break_points, breaks=3), col=4); lines(confint(break_points, breaks = 3))
```

```
## Warning: Confidence intervals outside data time interval
##   from 1997(2) to 2019(4) (91 observations)
```

```
## Warning: Overlapping confidence intervals
```



5.1.7 Structural Breaks

Because we can not use the strucchange package for the dynlm function we will perform my dynamic regression using simple lm() function. For that i need to extract the lags of my variables

```r
summary(ecm1_reduced2)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff,
##     4) + L(LC20, 1) + L(long_i, 1) + L(inf, 1) + L(bnp, 1) +
```

```
##     L(u, 1) + dummies_LC20_diff)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.103298 -0.027071  0.001718  0.028772  0.120779
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                             0.6489148  0.2737462   2.370
## L(LC20_diff, 1)                         0.3136766  0.0721621   4.347
## L(long_i_diff, 4)                       0.0647013  0.0197272   3.280
## L(LC20, 1)                             -0.1780332  0.0336884  -5.285
## L(long_i, 1)                           -0.0253787  0.0112880  -2.248
## L(inf, 1)                              -0.0014806  0.0096309  -0.154
## L(bnp, 1)                               0.0010812  0.0004862   2.224
## L(u, 1)                                 0.0074219  0.0068371   1.086
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.3056379  0.0557376  -5.484
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.1399927  0.0536476  -2.609
## dummies_LC20_diffdummy_LC20_diff_2015    0.1525339  0.0510645   2.987
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.1637867  0.0508980  -3.218
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.1943903  0.0545483  -3.564
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.1743013  0.0519990  -3.352
##                                           Pr(>|t|)
## (Intercept)                               0.020407 *
## L(LC20_diff, 1)                        0.000043986 ***
## L(long_i_diff, 4)                         0.001594 **
## L(LC20, 1)                             0.000001255 ***
## L(long_i, 1)                              0.027577 *
## L(inf, 1)                                 0.878246
## L(bnp, 1)                                 0.029262 *
## L(u, 1)                                   0.281257
## dummies_LC20_diffdummy_LC20_diff_2008Q4 0.000000569 ***
## dummies_LC20_diffdummy_LC20_diff_2008Q3    0.010995 *
## dummies_LC20_diffdummy_LC20_diff_2015      0.003833 **
## dummies_LC20_diffdummy_LC20_diff_2002Q3    0.001927 **
## dummies_LC20_diffdummy_LC20_diff_2011Q3    0.000649 ***
## dummies_LC20_diffdummy_LC20_diff_2003Q1    0.001274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04961 on 73 degrees of freedom
## Multiple R-squared:  0.7047, Adjusted R-squared:  0.6521
## F-statistic:  13.4 on 13 and 73 DF,  p-value: 0.00000000000001557
```

We will now create the model with the lm function.

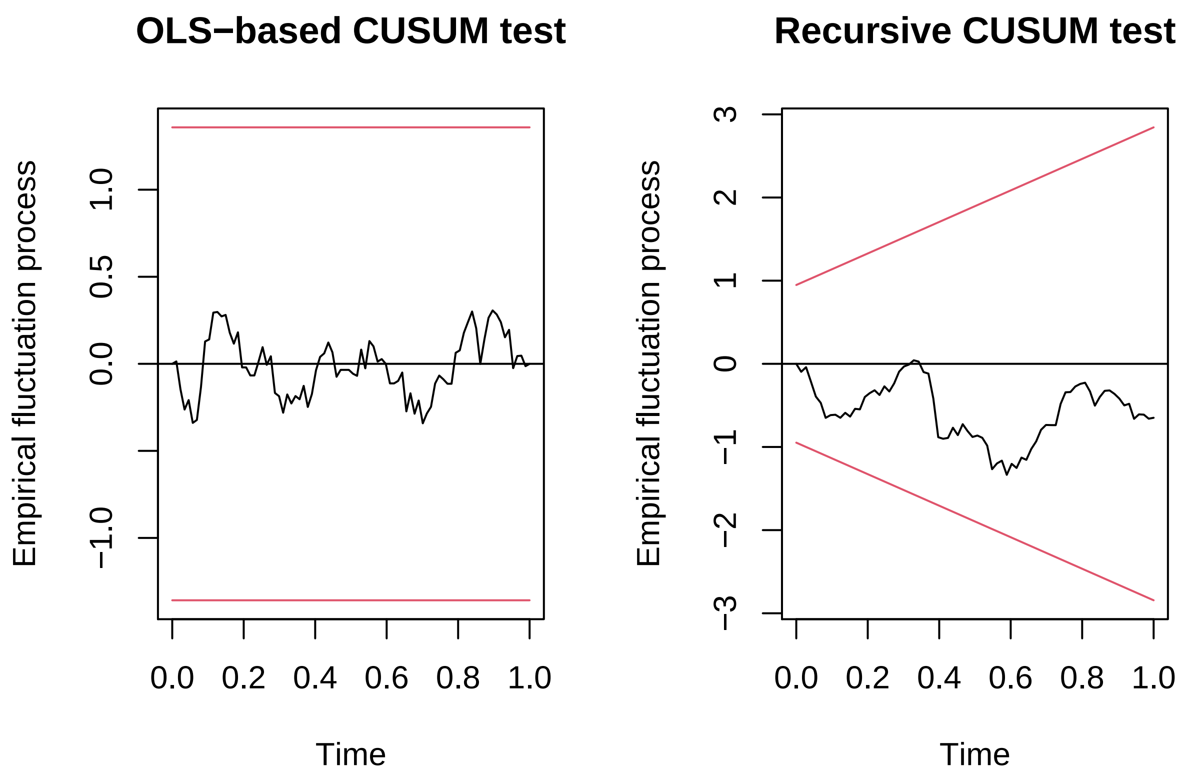
```r
##Created for testing for Homoskedasticity

summary(model_struc)
```

```
##
## Call:
## lm(formula = LC20_d ~ LC20_d1 + long_i_d4 + LC20_1 + long_i_1 +
##     inf_1 + bnp_1 + u_1 + dummy_LC20_diff_2008Q3 + dummy_LC20_diff_2008Q4 +
##     dummy_LC20_diff_2015 + dummy_LC20_diff_2002Q3 + dummy_LC20_diff_2011Q3 +
```

```
##     dummy_LC20_diff_2003Q1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.103298 -0.027071  0.001718  0.028772  0.120779
##
## Coefficients:
##                         Estimate Std. Error t value    Pr(>|t|)
## (Intercept)            0.6489148  0.2737462   2.370    0.020407 *
## LC20_d1                0.3136766  0.0721621   4.347 0.000043986 ***
## long_i_d4              0.0647013  0.0197272   3.280    0.001594 **
## LC20_1                -0.1780332  0.0336884  -5.285 0.000001255 ***
## long_i_1              -0.0253787  0.0112880  -2.248    0.027577 *
## inf_1                 -0.0014806  0.0096309  -0.154    0.878246
## bnp_1                  0.0010812  0.0004862   2.224    0.029262 *
## u_1                    0.0074219  0.0068371   1.086    0.281257
## dummy_LC20_diff_2008Q3 -0.1399927  0.0536476  -2.609    0.010995 *
## dummy_LC20_diff_2008Q4 -0.3056379  0.0557376  -5.484 0.000000569 ***
## dummy_LC20_diff_2015    0.1525339  0.0510645   2.987    0.003833 **
## dummy_LC20_diff_2002Q3 -0.1637867  0.0508980  -3.218    0.001927 **
## dummy_LC20_diff_2011Q3 -0.1943903  0.0545483  -3.564    0.000649 ***
## dummy_LC20_diff_2003Q1 -0.1743013  0.0519990  -3.352    0.001274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04961 on 73 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.7047, Adjusted R-squared:  0.6521
## F-statistic:  13.4 on 13 and 73 DF,  p-value: 0.00000000000001557
```

We will now use the strucchange package

```
par(mfrow=c(1,2))
struc_c <- (LC20_d~ LC20_d1+ long_i_d4+ LC20_1+ long_i_1+ inf_1+ bnp_1+ u_1+ dummy_LC20_diff_2008Q3+ du
ols_cusum_c <- efp(struc_c, type="OLS-CUSUM", data=Fed2)
rec_cusum_c <- efp(struc_c, type="Rec-CUSUM", data=Fed2)
plot(ols_cusum_c)
plot(rec_cusum_c)
```



We can see that the parameters are stable

5.2 SR estimater og Speed of Adjustment

Vi kan ligeledes f? SR forhold og speed of adjustment fra SR til LR.

Vores unrestricted model:

$$\Delta Z_t = \beta_i \Delta Z_{t-i} + \alpha_i \Delta X_{t-i} + \gamma_i \Delta Y_{t-i} + \theta_1 Z_{t-1} + \mu + \phi_1 X_{t-1} + \phi_2 Y_{t-1} + \varepsilon_t$$

Vi får dermed restricted model således:

$$\Delta Z_t = \beta_i \Delta Z_{t-i} + \alpha_i \Delta X_{t-i} + \gamma_i \Delta Y_{t-i} + \hat{\varepsilon}_{t-1} + \varepsilon_t$$

Her er $\hat{\varepsilon}_{t-1}$ vores lagged residual fra vores long run static regression.

Vi udleder f?rst vores residual fra vores long run static regression og inkorporerer den herefter i vores ecm som en lagged værdi. For at få residualet substituerer vi long run effekterne ud af modellen.

```
lrm=lm(LC20~inf+bnp+long_i+u)
error <- residuals(lrm)
error <- ts(error, start = c(1997,1) , frequency = 4)
ecm3 <- dynlm(LC20_diff~L(LC20_diff, 1)+ L(long_i_diff,4)+ L(error, 1) + dummies_LC20_diff)
summary(ecm3)
```

```
##
## Time series regression with "ts" data:
## Start = 1998(2), End = 2019(4)
##
## Call:
## dynlm(formula = LC20_diff ~ L(LC20_diff, 1) + L(long_i_diff,
##     4) + L(error, 1) + dummies_LC20_diff)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.12068 -0.02624  0.00000  0.02564  0.12388
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                              0.024518   0.006034   4.063
## L(LC20_diff, 1)                          0.350976   0.070275   4.994
## L(long_i_diff, 4)                        0.053607   0.019146   2.800
## L(error, 1)                             -0.183152   0.034483  -5.311
## dummies_LC20_diffdummy_LC20_diff_2008Q4 -0.304778   0.053012  -5.749
## dummies_LC20_diffdummy_LC20_diff_2008Q3 -0.151953   0.051659  -2.941
## dummies_LC20_diffdummy_LC20_diff_2015    0.157917   0.051615   3.060
## dummies_LC20_diffdummy_LC20_diff_2002Q3 -0.170709   0.051755  -3.298
## dummies_LC20_diffdummy_LC20_diff_2011Q3 -0.165950   0.052588  -3.156
## dummies_LC20_diffdummy_LC20_diff_2003Q1 -0.167726   0.052894  -3.171
##                                            Pr(>|t|)
## (Intercept)                                0.000116 ***
## L(LC20_diff, 1)                          0.000003593 ***
## L(long_i_diff, 4)                          0.006458 **
## L(error, 1)                              0.000001023 ***
## dummies_LC20_diffdummy_LC20_diff_2008Q4 0.000000172 ***
## dummies_LC20_diffdummy_LC20_diff_2008Q3    0.004313 **
## dummies_LC20_diffdummy_LC20_diff_2015      0.003050 **
## dummies_LC20_diffdummy_LC20_diff_2002Q3    0.001474 **
## dummies_LC20_diffdummy_LC20_diff_2011Q3    0.002285 **
## dummies_LC20_diffdummy_LC20_diff_2003Q1    0.002182 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05103 on 77 degrees of freedom
## Multiple R-squared:  0.6704, Adjusted R-squared:  0.6319
## F-statistic:  17.4 on 9 and 77 DF,  p-value: 0.000000000000002649
```

Speed of Adjustment

Vores lagged residual fra vores long run static regression er -0.183 og denne er meget statistisk signifikant. Dette betyder, at speed of adjustment fra SR til LR er forholdsvis langsom. Hvis eksogene chok rammer modellen, så går vi tilbage mod ligevægt med en hastighed på 18.3% hvert kvartal, således at vi er tilbage til long run ligevægt efter kvartaler.

Short run estimater

Det ses, at ved en stigning på 1 % i C20-indekset med 1 lag, vil den nutidige værdi af C20-indekset stige med 0.351%.

Det ses, at ved en stigning på 1 enhed i den lange rente med 4 lags, vil den nutidige værdi af C20-indekset stige med 5.36%.