Econometrics II - Problem 7

William Radaic Peron

September 20, 2020

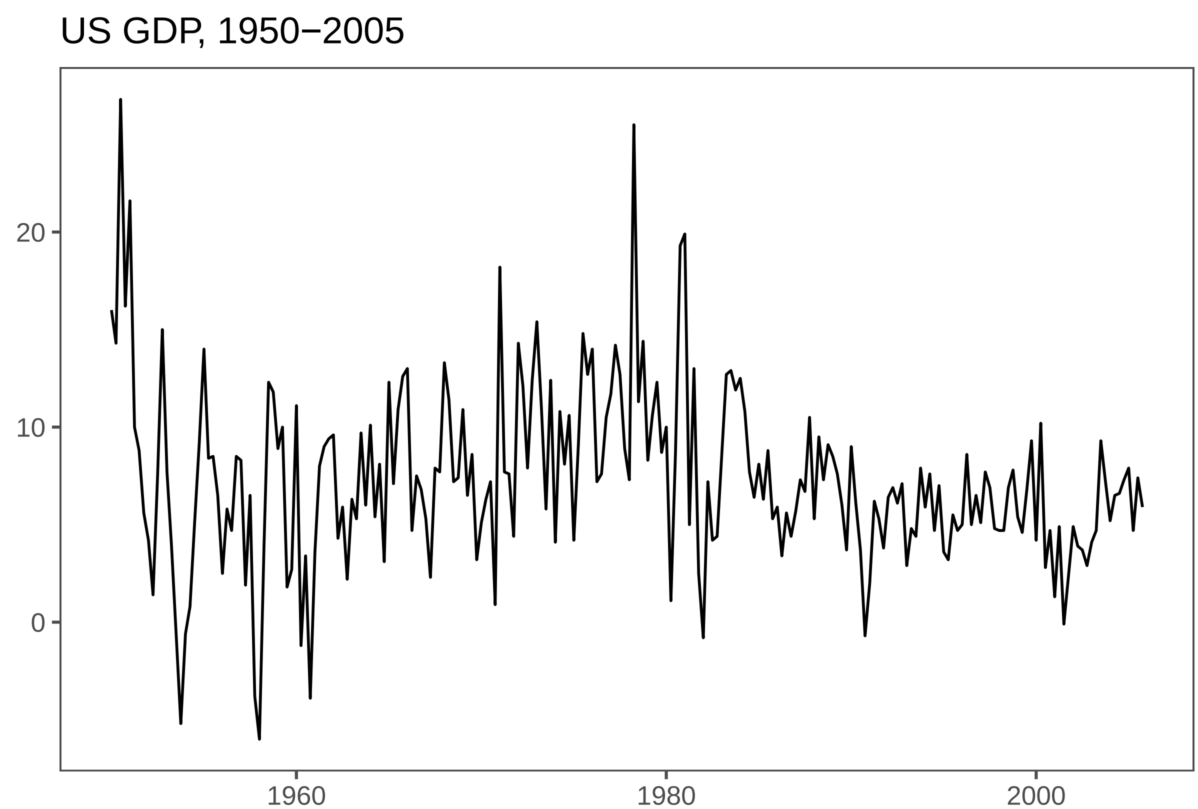
```
library(readxl)
library(ggplot2)
library(forecast)
library(dynlm)
library(ggthemes)
library(strucchange)
```

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

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

```
library(lmtest)
library(car)
library(dplyr)

df <- read_excel("G:/My Drive/FGV EESP/4o SEMESTRE/Econometria II/QuantEconEESP/QuantEconEESP/pibeua_rea
```

```
## New names:
## * `` -> ...1
```

```
series <- ts(df$`Cresimento percentual`[13:236], start = c(1950,
    1), end = c(2005, 4), frequency = 4)  # 1950-2005

autoplot(series) + theme_few() + ggtitle("US GDP, 1950-2005")
```




```r
# We can clearly see a reduction in variance during the 80s.

df$observ <- 1:length(df$`PIB nominal`)

# Suppose that the break happens at time t = 153 (Q1, 1985).

df$d <- as.numeric(df$observ > 152)

m1 <- lm(series ~ df$d[13:236])

summary(m1)
```

```
##
## Call:
## lm(formula = series ~ df$d[13:236])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2379  -2.0571  -0.1379   2.2871  18.5621
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.2379     0.3728  22.098  < 2e-16 ***
## df$d[13:236]  -2.5057     0.6088  -4.116 5.43e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.411 on 222 degrees of freedom
## Multiple R-squared:  0.0709, Adjusted R-squared:  0.06672
## F-statistic: 16.94 on 1 and 222 DF,  p-value: 5.435e-05
```

```r
m2 <- dynlm(series ~ df$d[13:236] + L(series, 1) + L(series,
    1) * df$d[13:236])

summary(m2)
```

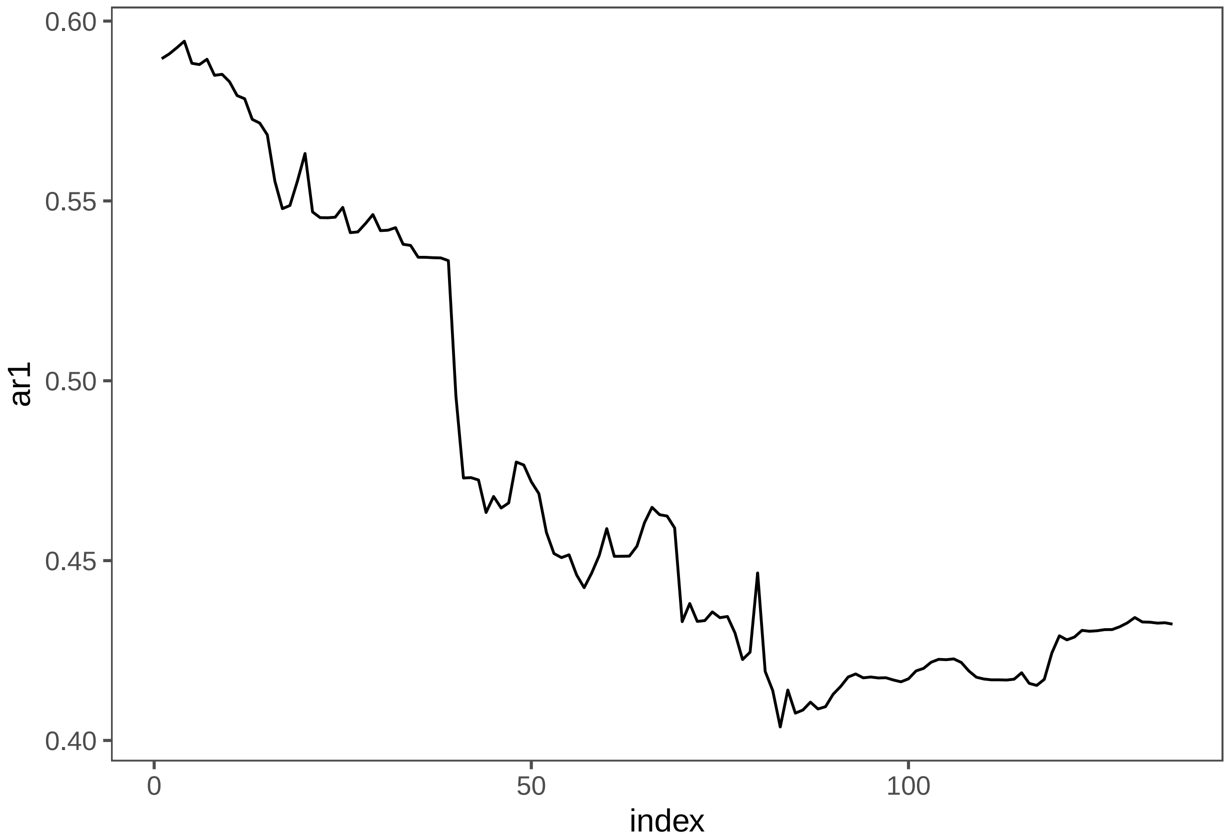
```
##
## Time series regression with "ts" data:
## Start = 1950(2), End = 2005(4)
##
## Call:
## dynlm(formula = series ~ df$d[13:236] + L(series, 1) + L(series,
##     1) * df$d[13:236])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2567  -2.1410  -0.0335   2.0335  17.7119
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.76430    0.63168   7.542 1.22e-12 ***
## df$d[13:236]                -0.40507    1.40339  -0.289    0.773
## L(series, 1)                 0.41421    0.06436   6.436 7.63e-10 ***
## df$d[13:236]:L(series, 1) -0.17495    0.21438  -0.816    0.415
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.033 on 219 degrees of freedom
## Multiple R-squared:  0.2209, Adjusted R-squared:  0.2103
## F-statistic:  20.7 on 3 and 219 DF,  p-value: 7.576e-12
```

```r
# ARIMA model

arima_unr <- arima(series, order = c(1, 0, 0))

arima_r1 <- arima(series[1:152], order = c(1, 0, 0))

arima_r2 <- arima(series[153:length(series)], order = c(1, 0,
    0))

ssr_unr <- sum(arima_unr$residuals^2)

ssr_r1 <- sum(arima_r1$residuals^2)

ssr_r2 <- sum(arima_r2$residuals^2)

# We will now define the Chow test for the null H0: \beta_m1
# - \beta_m2 = 0 (no structural break)

chow <- function(SSR_unr, SSR_r1, SSR_r2, t, n) {
```

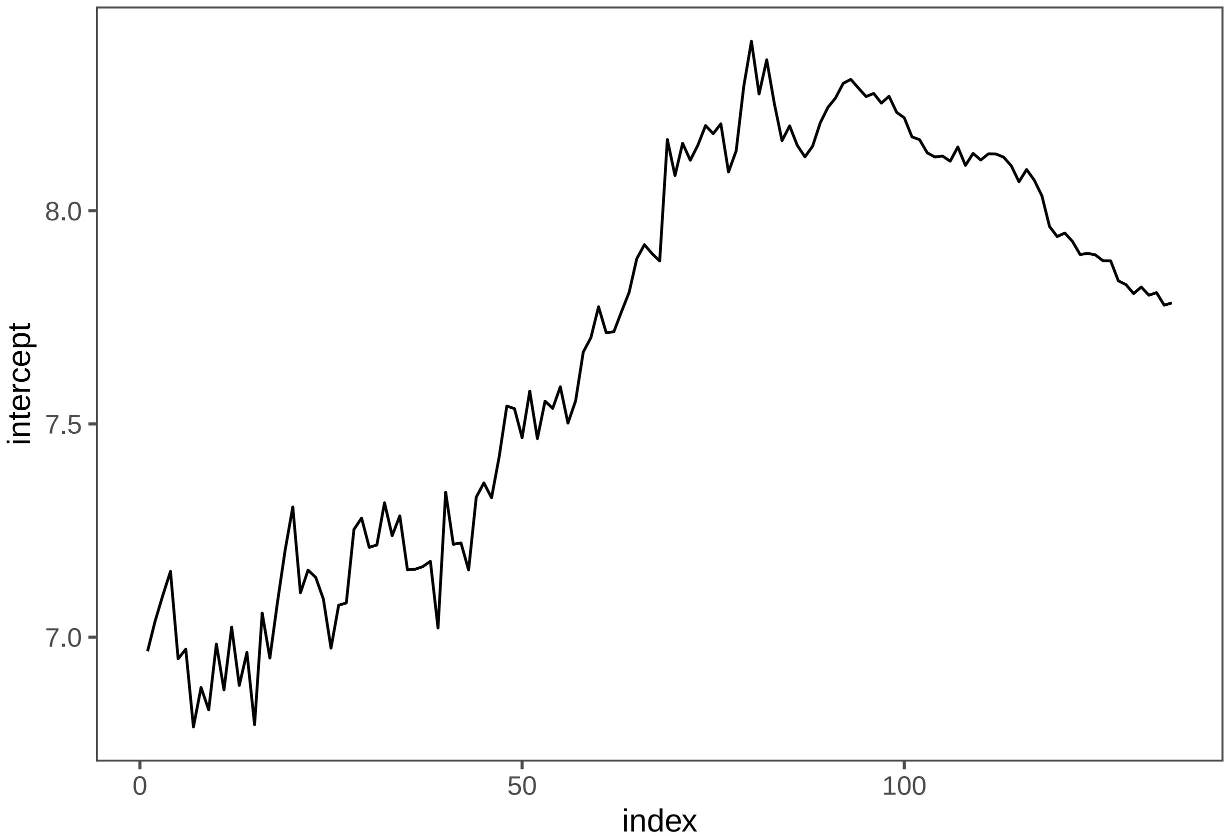
```r
    ((SSR_unr - SSR_r1 - SSR_r2)/n)/((SSR_r1 + SSR_r2)/(t - 2 *
        n))

}


chow(SSR_unr = ssr_unr, SSR_r1 = ssr_r1, SSR_r2 = ssr_r2, t = length(series),
    n = length(arima_unr$coef))  # T statistic (n, T - 2n).
```

```
## [1] 3.736208
```

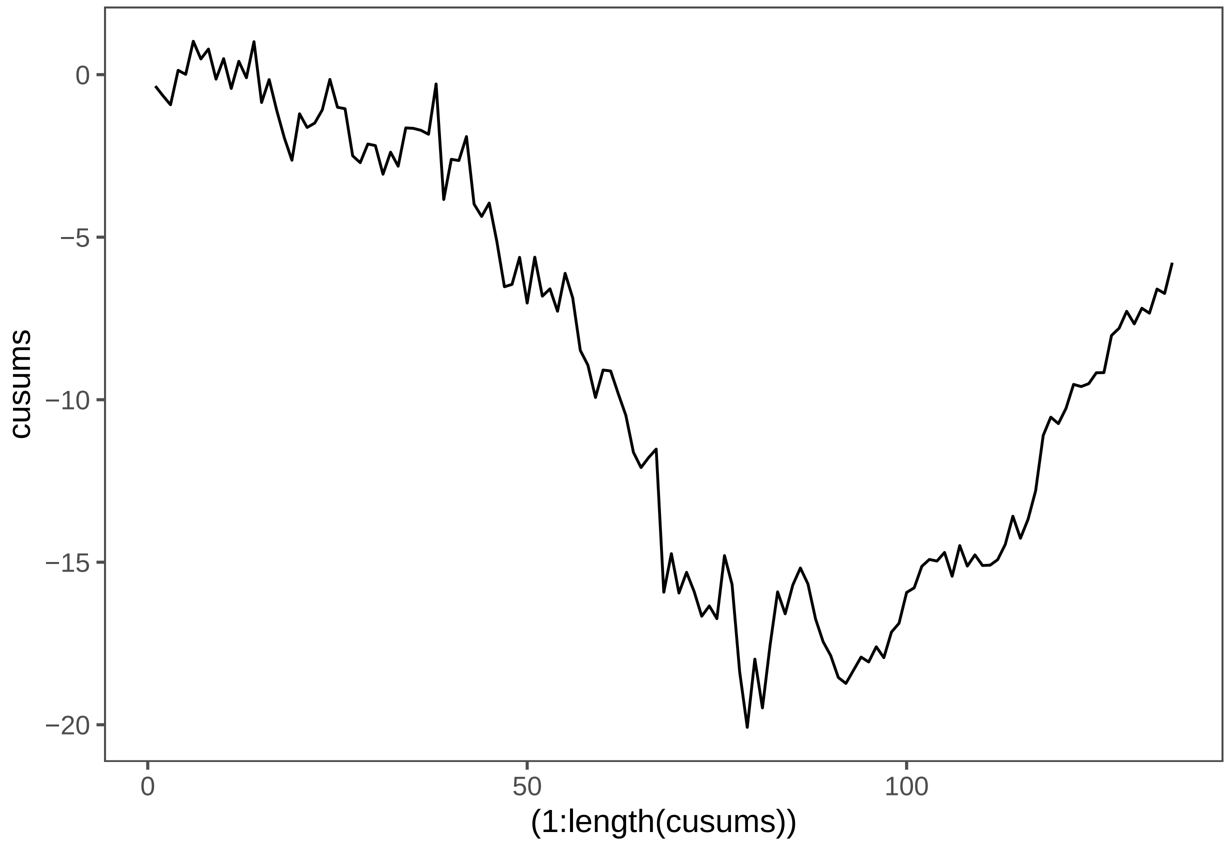
```r
# Now, suppose that we do not know when the break happened.

t0 = 45
tf = 180  # Boundaries for the process.

models = list(NA)

coefs = matrix(NA, nrow = length(t0:tf), ncol = 2)

forecasts = list(NA)

ci = data.frame(matrix(NA, nrow = length(t0:tf), ncol = 5))

e = matrix(NA, nrow = length(t0:tf), ncol = 1)


# 1. Plotting coefficients.

for (i in (1:(tf - t0))) {

    models[[i]] = arima(series[1:(i + t0)], order = c(1, 0, 0))

    coefs[i, ] = models[[i]]$coef

    forecasts[[i]] = forecast(series[1:(i + t0)], model = models[[i]],
        h = 1)

    e[i, ] = forecasts[[i]]$mean - series[(i + t0 + 1)]

}

coefs = data.frame(coefs)

coefs = data.frame(coefs, (1:length(coefs$X1)))

df.coefs = na.omit(data.frame(coefs))

names(df.coefs) = c("ar1", "intercept", "index")

ggplot(df.coefs, aes(x = index, y = ar1)) + geom_line() + theme_few()
```




```r
ggplot(df.coefs, aes(x = index, y = intercept)) + geom_line() +
    theme_few()
```




```r
# 2. Cusum test.

cusums = matrix(NA, nrow = length(e), ncol = 1)

e = na.omit(e)

for (i in 1:(length(e))) {

    cusums[i, ] = sum(e[1:i])/sd(e)

}


cusums = na.omit(cusums)

df.cusums = data.frame(cusums)

ggplot(df.cusums, aes(x = (1:length(cusums)), y = cusums)) +
    geom_line() + theme_few()
```

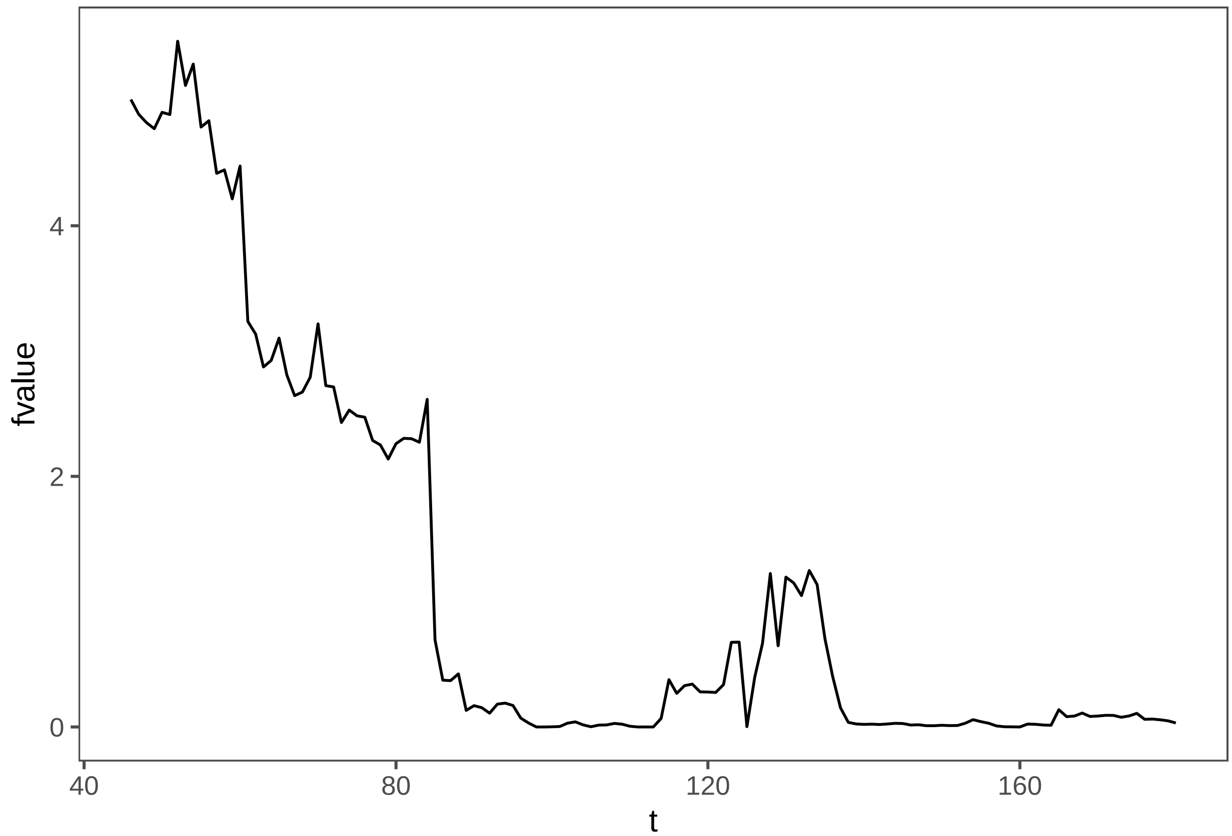



```
# 3. Iterative F-tests.

models_unr = list(NA)

models_r = list(NA)


# dummy <- df$d[13:236]

num_series = as.numeric(series)

f_values = matrix(NA, nrow = length(num_series), ncol = 2)

hyp = c(0, -1, 0, 1)

rhs = 0

length((t0:tf))
```

```
## [1] 136
```

```
dummies = data.frame(matrix(NA, ncol = 224, nrow = 224))

for (i in (13:236)) {
    dummies[i] = as.numeric(df$observ[13:236] >= i)
}
```

```r
for (i in (1:(tf - t0))) {

    adummy <- rep(0, 1)
    j <- 0
    k <- t0 + i
    while (j <= length(num_series)) {
        if (j > k) {
            # nÃ£o sei se isso deveria ser maior ou igual ou sÃ³ maior
            adummy[j] <- 1
        } else {
            adummy[j] <- 0
        }
        j <- j + 1
    }
    models_unr[[i]] <- dynlm(num_series ~ adummy + lag(num_series) +
        lag(num_series) * adummy)
    f_values[i, 1] = linearHypothesis(models_unr[[i]], hyp, rhs)$F[1]
    f_values[i, 2] = linearHypothesis(models_unr[[i]], hyp, rhs)$F[2]  #eu acho que funcionou?????

}

df.f_values <- data.frame(f_values)

df.f_values <- df.f_values$X2

df.f_values <- na.omit(df.f_values)

df.f_values <- data.frame(df.f_values, ((t0 + 1):tf))

names(df.f_values) <- c("fvalue", "t")

ggplot(df.f_values, aes(x = t, y = fvalue)) + geom_line() + theme_few()
```




```r
linearHypothesis(models_unr[[10]], hyp, rhs)
```

```
## Linear hypothesis test
##
## Hypothesis:
## - adummy  + adummy:lag(num_series) = 0
##
## Model 1: restricted model
## Model 2: num_series ~ adummy + lag(num_series) + lag(num_series) * adummy
##
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    220 3673.8
## 2    219 3595.2  1    78.591 4.7874 0.02973 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```