Trabalho-Prova Econometria 3

William Y. N. Suzuki

16 de novembro de 2018

Este trabalho foi apresentado na disciplina de econometria 3 ministrada por Márcio P. Laurini no segundo semestre de 2018. Programa de pós-graduação em economia da FEA/RP USP.

Vamos começar trantando os dados:

```
library(readxl)
library(vars)
```

```
## Loading required package: MASS

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: urca

## Loading required package: lmtest
```

```
library(rmarkdown)
library(aTSA)
```

```
##
## Attaching package: 'aTSA'

## The following object is masked from 'package:vars':
##
##     arch.test

## The following object is masked from 'package:graphics':
##
##     identify
```

```
library(lmtest)
library(forecast)
```

```
##
## Attaching package: 'forecast'

## The following object is masked from 'package:aTSA':
##
##     forecast
```

Em seguida é importante lembrar de incluir o file "etf.xlsx" no diretório.

```
etf_data <- read_excel("etf.xlsx") #import file
head(etf_data)
```

```
## # A tibble: 6 x 3
##   X__1                etfcaixa           IBOV
##   <dttm>              <chr>              <chr>
## 1 2012-11-28 00:00:00 1                  56539
## 2 2012-11-29 00:00:00 1.0273080000000001 57852
## 3 2012-11-30 00:00:00 1.0323770000000001 57474
## 4 2012-12-03 00:00:00 1.0402940000000001 58202
## 5 2012-12-04 00:00:00 1.0411900000000001 57563
## 6 2012-12-05 00:00:00 1.0340469999999999 57678
```

```r
names(etf_data)
```

```
## [1] "X__1"     "etfcaixa" "IBOV"
```

Podemos ver que `etf_data$etfcaixa` e `etf_data$IBOV` *estão como character*.

Em seguida vamos fazer preparar os dados corrigindo o problema de missings e visualizar algumas características dos dados.

```r
#treatment for dates
date <- as.character(etf_data$X__1)
date <- as.Date(date,"%Y-%m-%d")
class(date)
```

```
## [1] "Date"
```

```r
head(date)
```

```
## [1] "2012-11-28" "2012-11-29" "2012-11-30" "2012-12-03" "2012-12-04"
## [6] "2012-12-05"
```

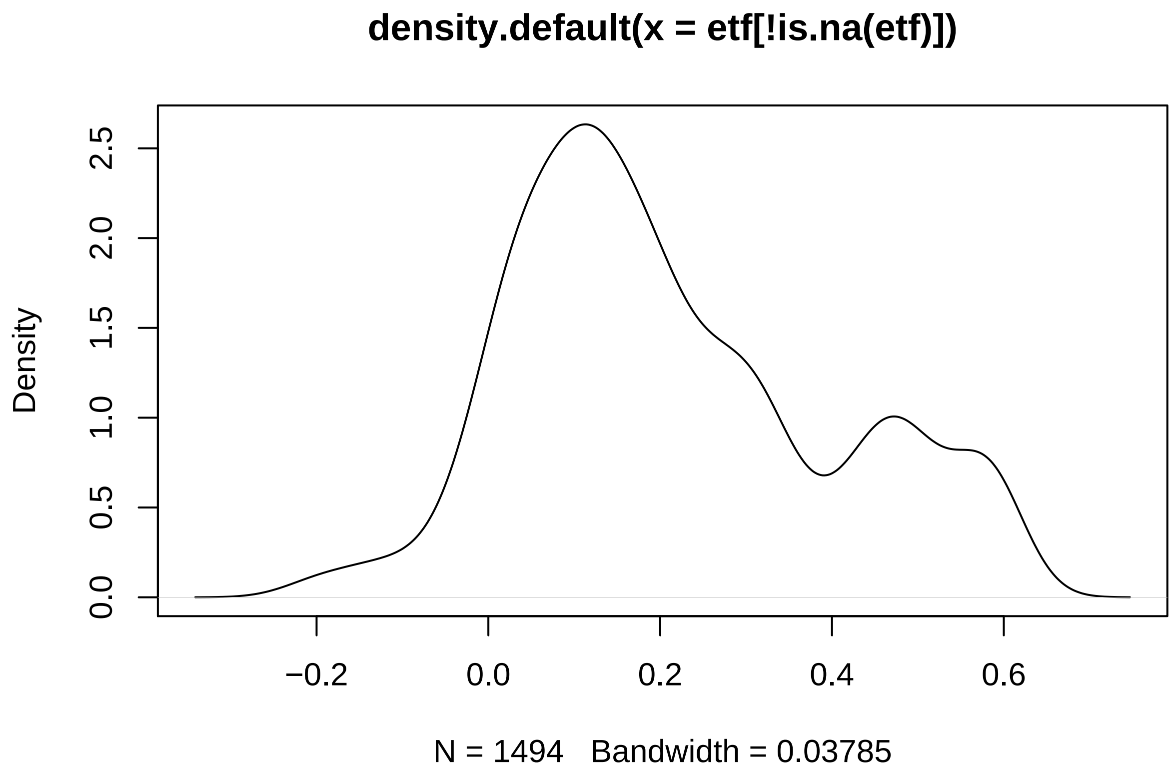
```r
#treatment for etf
etf<-log(as.numeric(etf_data$etfcaixa))
```

```
## Warning: NAs introduced by coercion
```

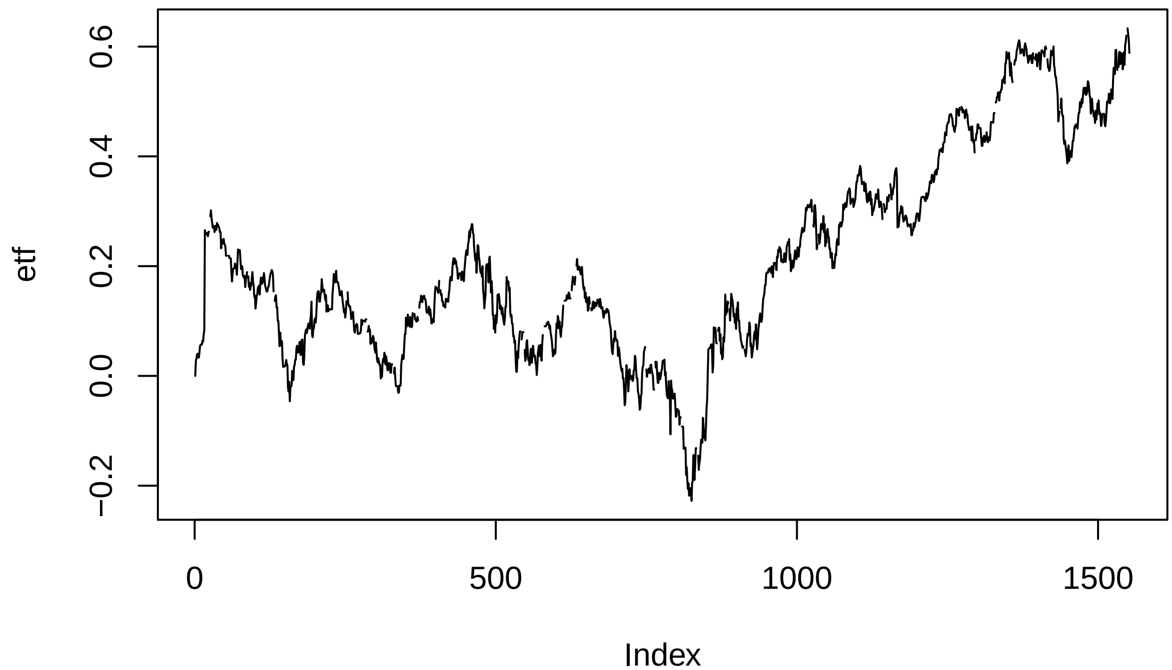
```r
sum(is.na(etf)) #number of missings
```

```
## [1] 59
```

```r
plot(density(etf[!is.na(etf)]))
```




```r
plot(etf,type = 'l') #plot graph of series
```




```r
table(is.na(etf)*1:length(etf)) #position of missings
```

```
##
##     0   20   25   54   55   88  111  132  253  281  286  329  330  363  364
## 1494    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##   372  407  542  547  579  580  613  625  633  657  724  749  764  803  808
##     1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##   834  835  868  887  912  986 1011 1026 1035 1109 1110 1143 1148 1154 1187
##     1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 1247 1272 1287 1296 1324 1329 1359 1360 1393 1415 1437 1508 1533 1548 1553
##     1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

```r
#treatment for missings: use flat method, where we input the last observation
#in the place of the missing
for (t in 1:length(etf)) {
  if (is.na(etf[t])==TRUE){etf[t] <- etf[t-1]}
}
```

Na última parte acima fizemos com que sempre um *missings data* fosse substituído pela última observação.

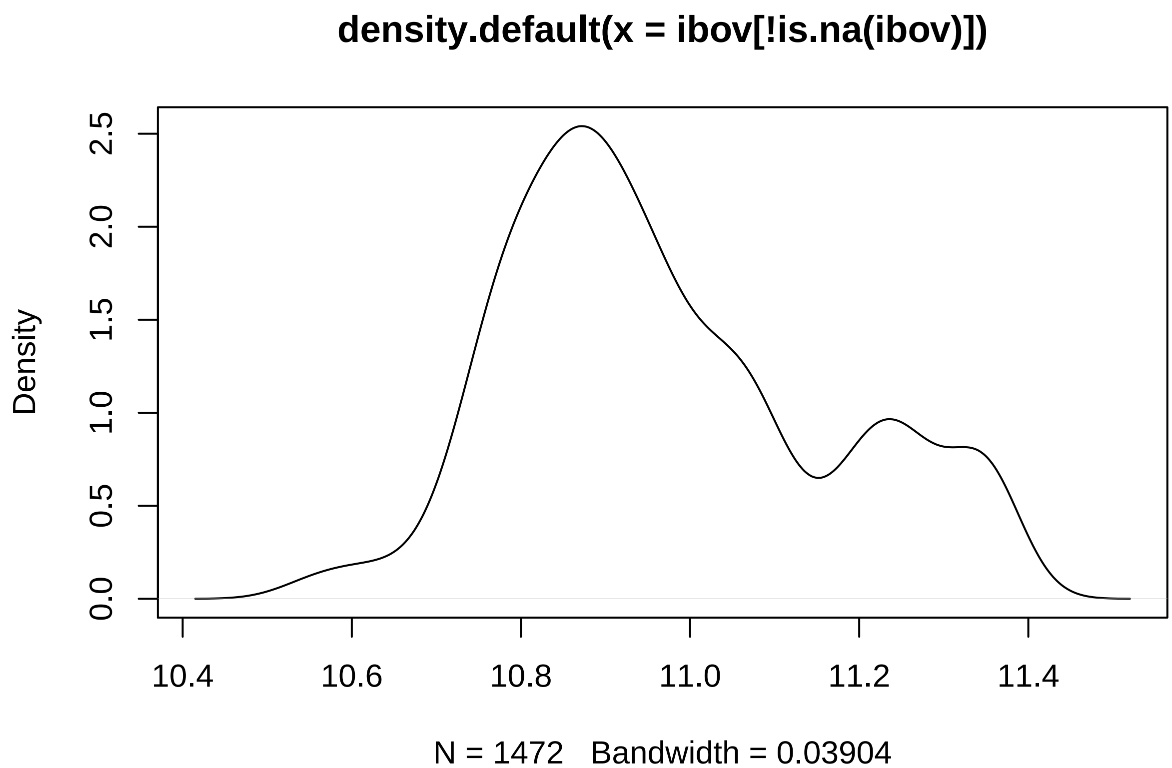
```r
#treatment for ibov data
ibov <- log(as.numeric(etf_data$IBOV))
```

```
## Warning: NAs introduced by coercion
```

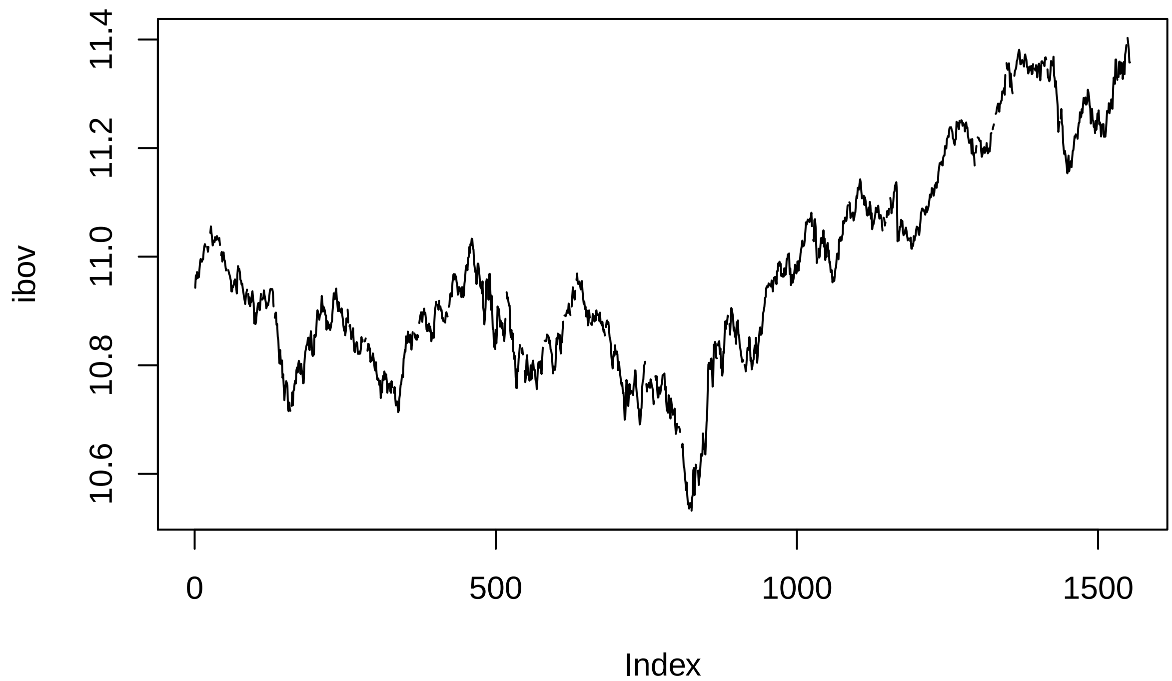
```r
sum(is.na(ibov)) #number of missings
```

```
## [1] 81
```

```r
plot(density(ibov[!is.na(ibov)]))
```




```r
plot(ibov,type = 'l')
```




```
table(is.na(ibov)*1:length(ibov)) #position of missings
```

```
##
##    0   19   20   24   25   43   54   55   88  111  132  160  253  256  280
## 1472    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  281  285  286  329  330  363  364  372  402  407  421  517  541  542  546
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  547  579  580  613  625  633  657  682  724  749  764  778  802  803  807
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  808  824  834  835  868  887  912  986 1011 1026 1035 1068 1086 1109 1110
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 1143 1148 1154 1187 1247 1272 1287 1296 1299 1324 1328 1329 1347 1359 1360
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 1393 1415 1437 1464 1508 1533 1548
##    1    1    1    1    1    1    1
```

```
#treatment for missings: use flat method where we imput the last observation
#in the place of the missing
for (t in 1:length(ibov)) {
  if (is.na(ibov[t])==TRUE){ibov[t] <- ibov[t-1]}
}

sum(is.na(ibov)) #verify number of missings
```

```
## [1] 0
```

Acabamos de tratar os dados da base sobre a bolsa.

```r
#check the length of vectors
length(date)
```

```
## [1] 1553
```

```r
length(ibov)
```

```
## [1] 1553
```

```r
length(etf)
```

```
## [1] 1553
```

```r
#merge the vectors
data_ibov <- cbind(date,ibov,etf)
head(data_ibov)
```

```
##       date     ibov        etf
## [1,] 15672 10.94269 0.00000000
## [2,] 15673 10.96564 0.02694179
## [3,] 15674 10.95909 0.03186391
## [4,] 15677 10.97167 0.03950337
## [5,] 15678 10.96064 0.04036429
## [6,] 15679 10.96263 0.03348023
```

Agora vamos começar a tratar os dados sobre a produção industrial.

```r
#editing data of industry
industria <- read_excel("prodindustrialcapital.xls") #import data
head(industria)
```

```
## # A tibble: 6 x 2
##   Data    Prodindustrialcapital
##   <chr>                   <dbl>
## 1 1991.01                  77.2
## 2 1991.02                  78.8
## 3 1991.03                  78.3
## 4 1991.04                  81.9
## 5 1991.05                  84.6
## 6 1991.06                  92.7
```

```r
names(industria)
```

```
## [1] "Data"                  "Prodindustrialcapital"
```

```r
#treatment for dates
date <- industria$Data
head(date)
```

```
## [1] "1991.01" "1991.02" "1991.03" "1991.04" "1991.05" "1991.06"
```

```r
class(date)
```

```
## [1] "character"
```

```r
date2 <- paste(date,"01",sep='.') #we need to include the day otherwise it does not work
head(date2)
```

```
## [1] "1991.01.01" "1991.02.01" "1991.03.01" "1991.04.01" "1991.05.01"
## [6] "1991.06.01"
```

```r
date3 <- as.Date(date2,"%Y.%m.%d")
head(date3)
```

```
## [1] "1991-01-01" "1991-02-01" "1991-03-01" "1991-04-01" "1991-05-01"
## [6] "1991-06-01"
```

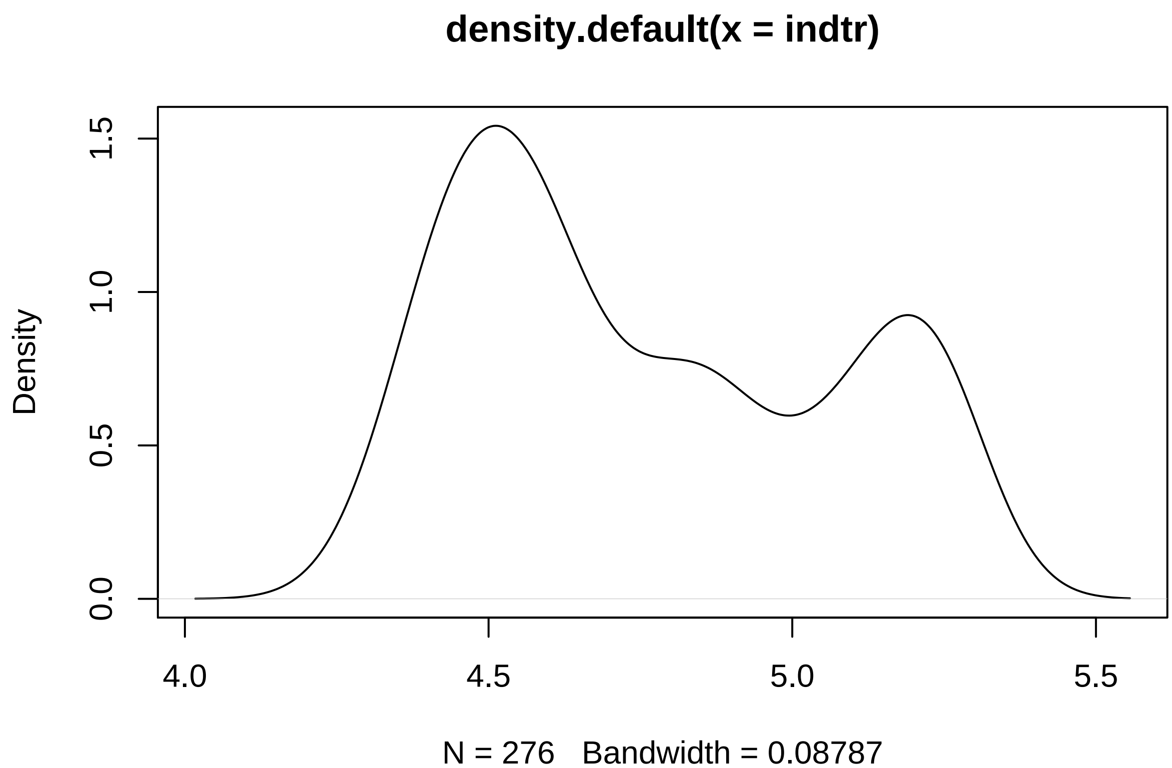
```r
class(date3)
```

```
## [1] "Date"
```

```r
#treatment for data about industrial production
indtr <- log(industria$Prodindustrialcapital)
head(indtr)
```

```
## [1] 4.347047 4.367040 4.361058 4.405255 4.438171 4.529584
```

```r
plot(density(indtr))
```




```r
plot(indtr,type = 'l')
```


```r
#check length of vectors
length(indtr)
```

```
## [1] 276
```

```r
length(date3)
```

```
## [1] 276
```

```r
#merge vectors
data_ind <- cbind(date3,indtr)
head(data_ind)
```

```
##      date3    indtr
## [1,]  7670 4.347047
## [2,]  7701 4.367040
## [3,]  7729 4.361058
## [4,]  7760 4.405255
## [5,]  7790 4.438171
## [6,]  7821 4.529584
```

Questão 2

Vamos fazer as análises.

```r
#build a VAR model with ibov and etf
```

```r
#selecting number of lags for VAR
VARselect(cbind(etf, ibov), lag.max =10, type="both")$selection
```

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

```r
VARselect(cbind(etf, ibov), lag.max =10, type="trend")$selection
```

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

```r
VARselect(cbind(etf, ibov), lag.max =10, type="const")$selection
```

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

```r
VARselect(cbind(etf, ibov), lag.max =10, type="none")$selection
```

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

```r
#using Schwartz criterion we use VAR(2)
```

Podemos ver que 2 lags acaba sendo escolhido pelo critério de Schwartz.

```r
#apply to a model VAR
model1 <- VAR(cbind(etf,ibov), p=2, type="both")
summary(model1)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: etf, ibov
## Deterministic variables: both
## Sample size: 1551
## Log Likelihood: 9963.452
## Roots of the characteristic polynomial:
## 0.994 0.919 0.2167 0.05968
## Call:
## VAR(y = cbind(etf, ibov), p = 2, type = "both")
##
##
## Estimation results for equation etf:
## ====================================
## etf = etf.l1 + ibov.l1 + etf.l2 + ibov.l2 + const + trend
##
##           Estimate Std. Error t value Pr(>|t|)
## etf.l1   5.178e-01  5.333e-02   9.709  < 2e-16 ***
## ibov.l1  4.865e-01  5.700e-02   8.536  < 2e-16 ***
## etf.l2   3.421e-01  5.163e-02   6.625 4.77e-11 ***
## ibov.l2 -3.548e-01  5.616e-02  -6.317 3.48e-10 ***
## const   -1.417e+00  2.210e-01  -6.412 1.91e-10 ***
## trend    2.008e-06  1.170e-06   1.716   0.0864 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.01481 on 1545 degrees of freedom
## Multiple R-Squared: 0.9936,  Adjusted R-squared: 0.9936
## F-statistic: 4.799e+04 on 5 and 1545 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation ibov:
## =====================================
## ibov = etf.l1 + ibov.l1 + etf.l2 + ibov.l2 + const + trend
##
##            Estimate Std. Error t value Pr(>|t|)
## etf.l1  -1.561e-01  5.049e-02  -3.091  0.00203 **
## ibov.l1  1.119e+00  5.397e-02  20.732  < 2e-16 ***
## etf.l2   1.149e-01  4.889e-02   2.350  0.01892 *
## ibov.l2 -8.460e-02  5.317e-02  -1.591  0.11180
## const   -3.685e-01  2.093e-01  -1.761  0.07841 .
## trend    2.743e-06  1.108e-06   2.476  0.01339 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01402 on 1545 degrees of freedom
## Multiple R-Squared: 0.9946,  Adjusted R-squared: 0.9945
## F-statistic: 5.643e+04 on 5 and 1545 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             etf      ibov
## etf  0.0002192 0.0001843
## ibov 0.0001843 0.0001965
##
## Correlation matrix of residuals:
##         etf   ibov
## etf  1.0000 0.8882
## ibov 0.8882 1.0000
```

```r
dim(resid(model1))
```

```
## [1] 1551    2
```

```r
serial.test(model1, lags.pt = 16, lags.bg = 5, type = c("PT.asymptotic") )
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object model1
## Chi-squared = 101.69, df = 56, p-value = 0.0001827
```

```r
m1 <- ca.jo(cbind(etf,ibov), type = "trace", ecdet = "trend", K = 2, spec = "transitory")
summary(m1)
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
```

```
## Test type: trace statistic , with linear trend in cointegration
##
## Eigenvalues (lambda):
## [1] 7.003165e-02 4.706615e-03 9.544944e-18
##
## Values of teststatistic and critical values of test:
##
##            test 10pct  5pct  1pct
## r <= 1 |   7.32 10.49 12.25 16.26
## r = 0  | 119.93 22.76 25.32 30.45
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                  etf.l1       ibov.l1     trend.l1
## etf.l1    1.000000e+00  1.0000000000  1.000000000
## ibov.l1  -9.777182e-01 -0.3761988713 -2.920494961
## trend.l1  3.528151e-06 -0.0002837902 -0.000420897
##
## Weights W:
## (This is the loading matrix)
##
##              etf.l1      ibov.l1      trend.l1
## etf.d  -0.13143873 -0.008709816 -3.981684e-17
## ibov.d -0.03113385 -0.010053201 -5.325602e-17
```

```
class(m1)
```

```
## [1] "ca.jo"
## attr(,"package")
## [1] "urca"
```

```
model2 <- vec2var(m1,r=1)
class(model2)
```

```
## [1] "vec2var"
```

```
model2$deterministic
```

```
##        constant      trend.l1
## etf  -1.3810821 -4.637356e-07
## ibov -0.3269555 -1.098449e-07
```

```
serial.test(model2, lags.pt = 16, lags.bg = 5, type = c("PT.asymptotic") )
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object model2
## Chi-squared = 101.04, df = 58, p-value = 0.0004011
```

Vamos usar 1 vetor de cointegração.

```
vec <- cajorls(m1, r = 1)
vec
```

```
## $rlm
##
## Call:
```

```
## lm(formula = substitute(form1), data = data.mat)
##
## Coefficients:
##           etf.d     ibov.d
## ect1      -0.13144  -0.03113
## constant  -1.38108  -0.32696
## etf.dl1   -0.35012  -0.12414
## ibov.dl1   0.36165   0.09255
##
##
## $beta
##                  ect1
## etf.l1    1.000000e+00
## ibov.l1  -9.777182e-01
## trend.l1  3.528151e-06
```

```
coef(summary(vec$rlm))
```

```
## Response etf.d :
##            Estimate Std. Error   t value     Pr(>|t|)
## ect1     -0.1314387 0.02087023 -6.297905 3.922559e-10
## constant -1.3810821 0.21935604 -6.296075 3.967940e-10
## etf.dl1  -0.3501187 0.05154876 -6.791992 1.572214e-11
## ibov.dl1  0.3616464 0.05609467  6.447072 1.520004e-10
##
## Response ibov.d :
##             Estimate Std. Error   t value   Pr(>|t|)
## ect1     -0.03113385 0.01977475 -1.574424 0.11559393
## constant -0.32695549 0.20784204 -1.573096 0.11590106
## etf.dl1  -0.12414336 0.04884296 -2.541684 0.01112887
## ibov.dl1  0.09255029 0.05315026  1.741295 0.08183068
```

Respondendo a questão **2.a) Obtenha a melhor representação VAR/VEC para estes dois ativos.** O modelo VAR é de ordem 2 e portanto o VEC é de ordem 1. Os estimadores que encontramos para o modelo VAR são:

```
model1$varresult
```

```
## $etf
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##     etf.l1     ibov.l1      etf.l2     ibov.l2       const       trend
##  5.178e-01   4.865e-01   3.421e-01  -3.548e-01  -1.417e+00   2.008e-06
##
##
## $ibov
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##     etf.l1     ibov.l1      etf.l2     ibov.l2       const       trend
## -1.561e-01   1.119e+00   1.149e-01  -8.460e-02  -3.685e-01   2.743e-06
```

Respondendo a **2.b) Determine a dinâmica de curto e longa prazo entre essas séries.** podemos ver que a relação de cointegração é dada por

```
vec$beta
```

```
##                   ect1
## etf.l1    1.000000e+00
## ibov.l1  -9.777182e-01
## trend.l1  3.528151e-06
```

essa é a relação de longo prazo. As relações de curto prazo são dados por:

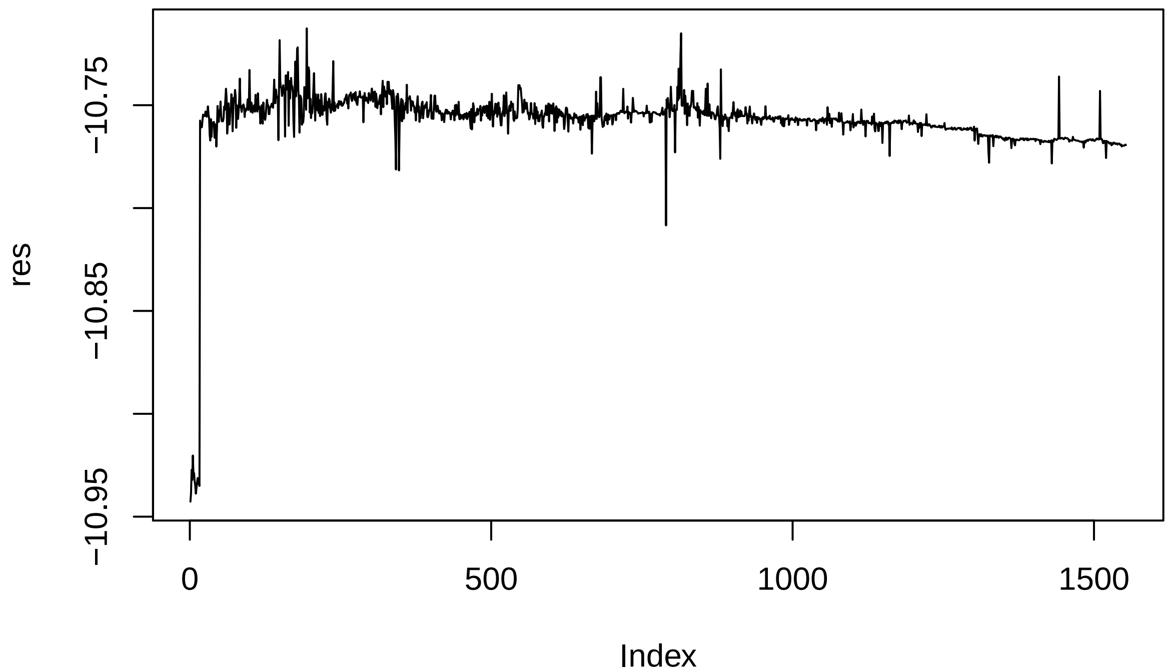
```
vec$rlm
```

```
##
## Call:
## lm(formula = substitute(form1), data = data.mat)
##
## Coefficients:
##           etf.d     ibov.d
## ect1      -0.13144  -0.03113
## constant  -1.38108  -0.32696
## etf.dl1   -0.35012  -0.12414
## ibov.dl1   0.36165   0.09255
```

para os coeficientes de etf.dl1 e ibov.dl1. E podemos ver que a velocidade de ajuste são -0.1314387 e -0.0311338 mas o p-valor do segundo é de 0.1155939, portanto podemos responder também **2. d) Verifique se o Ibovespa é exógena fraca nesse sistema.**, sim porque o coeficiente de ajuste para o ibovespa é estatíticamente igual a zero.

Teste para ajuste unitário no vetor de cointegração

```
res <- etf - ibov
plot(res,type="l")
```




```r
adf.test(res)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -0.615   0.459
## [2,]   1 -0.784   0.399
## [3,]   2 -0.850   0.375
## [4,]   3 -0.910   0.353
## [5,]   4 -0.878   0.365
## [6,]   5 -0.916   0.352
## [7,]   6 -0.873   0.367
## [8,]   7 -0.894   0.359
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -10.87    0.01
## [2,]   1 -10.19    0.01
## [3,]   2  -9.88    0.01
## [4,]   3 -10.22    0.01
## [5,]   4 -10.09    0.01
## [6,]   5 -10.66    0.01
## [7,]   6 -11.10    0.01
## [8,]   7 -11.79    0.01
## Type 3: with drift and trend
```

```
##      lag   ADF p.value
## [1,]   0 -11.1    0.01
## [2,]   1 -10.5    0.01
## [3,]   2 -10.2    0.01
## [4,]   3 -10.6    0.01
## [5,]   4 -10.6    0.01
## [6,]   5 -11.2    0.01
## [7,]   6 -11.8    0.01
## [8,]   7 -12.6    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

Vemos que o resíduo do vetor de cointegração quando estabelecemos valores iguais a (1,-1) são estacionários se ajustamos para tendência ou constante. No começo da séries há um quebra drática porque a *etf* acabou subindo muito mais do que o *ibov*. Possivelmente por causa disso é que se não colocarmos termos determinístas o teste de raiz unitária nos diz que temos não estacionariedade. Respondemos a questão **2. c) Teste se existe ajuste unitário no vetor de cointegração.**

Relação de causalidade de Granger

```
grangertest(ibov,etf,order = 2)
```

```
## Granger causality test
##
## Model 1: etf ~ Lags(etf, 1:2) + Lags(ibov, 1:2)
## Model 2: etf ~ Lags(etf, 1:2)
##   Res.Df Df      F    Pr(>F)
## 1   1546
## 2   1548 -2 47.391 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
grangertest(etf,ibov,order = 2)
```

```
## Granger causality test
##
## Model 1: ibov ~ Lags(ibov, 1:2) + Lags(etf, 1:2)
## Model 2: ibov ~ Lags(ibov, 1:2)
##   Res.Df Df      F   Pr(>F)
## 1   1546
## 2   1548 -2 5.2486 0.005349 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Respondendo a questão **2. e) Verifique as relações de causalidade de Granger nesse sistema**: parece que ambas variáveis Granger causam uma a outra, pesar de que *eft* tem um impacto menor em *ibov* quando comparado com a relação inversa. Seria de se esperar que *ibov* Granger-causa *etf* apenas.

Previsão

A previsão estática considera que estamos usando o último dado que foi observado, na previsão dinâmica vamos usar os dados previstos no passo anterior. Considere o modelo VEC com 1 lag

$$\Delta Y_t = c + \alpha\beta' Y_{t-1} + \Gamma_1 \Delta Y_{t-1} + u_t$$

Agora vamos construir as previsões na amostra com 30 passos, primeiro a previsão dinâmica com e sem considerar ibovespa como exógena fraca.

```
#dynamic forecasting
vec$rlm[[1]][1,] #estimated alpha coefficients
```

```
##       etf.d      ibov.d
## -0.13143873 -0.03113385
```

```
vec$rlm[[1]][2,] #constantes
```

```
##      etf.d     ibov.d
## -1.3810821 -0.3269555
```

```
t(vec$rlm[[1]][c(3,4),]) #matrix of coefficients of delta lag
```

```
##           etf.dl1   ibov.dl1
## etf.d  -0.3501187 0.36164642
## ibov.d -0.1241434 0.09255029
```

```
t(vec$beta) #transpose of cointegration vector
```

```
##      etf.l1    ibov.l1     trend.l1
## ect1      1 -0.9777182 3.528151e-06
```

```
#cbind(etf,ibov)
```

Vamos criar um vetor de diferenças porque usando apenas diff o R vai criar um vetor onde na primeira posição na verdade temos ΔY_2 não queremos isso, queremos que na posição 2 o valor seja ΔY_2.

```
n1 <- 30 #numero de previsoes, numero de observacoes da amostra observada a ser incluida na matriz de p
n2 <- length(etf)
etf.d <- c(0,diff(etf))
ibov.d <- c(0,diff(ibov))
```

O resultado fica: os 30 primeiros são valores da base original, e os outros 30 são valores que vamos preencher, todos eles compõem as últimas 60 posições da amostra original.

```
prv <- rbind(cbind(etf.d,ibov.d)[(n2-2*n1+1):(n2-n1),], cbind(rep(0,n1),rep(0,n1)))
prv
```

```
##               etf.d        ibov.d
##  [1,]  3.947030e-03  0.0039300008
##  [2,] -1.528501e-02 -0.0151478436
##  [3,]  2.225096e-02  0.0226453564
##  [4,] -1.624330e-02 -0.0166330385
##  [5,]  8.480730e-03  0.0082750506
##  [6,]  2.097671e-02  0.0216290827
##  [7,] -5.824805e-03 -0.0058751980
##  [8,]  1.147909e-02  0.0117497583
##  [9,] -2.520971e-02 -0.0256452106
## [10,]  3.937100e-03  0.0035723462
## [11,] -6.449320e-03 -0.0063425233
## [12,] -1.905952e-02 -0.0196281377
## [13,]  4.961734e-03  0.0050795792
## [14,]  1.712385e-02  0.0174747360
## [15,]  0.000000e+00  0.0000000000
## [16,]  2.593062e-04  0.0002661391
## [17,] -4.341894e-06 -0.0235607720
## [18,] -1.713270e-02  0.0062531366
## [19,] -5.642431e-03 -0.0058491007
```

```
## [20,]  9.643728e-03  0.0098913792
## [21,]  1.705683e-02  0.0178664353
## [22,]  1.844564e-02  0.0196664259
## [23,] -1.034703e-03 -0.0018572035
## [24,] -2.617279e-04 -0.0006736425
## [25,]  1.572418e-02  0.0168608638
## [26,] -1.753872e-02 -0.0185499075
## [27,] -4.259088e-06  0.0082490998
## [28,]  8.610010e-03  0.0003307344
## [29,]  1.662648e-02  0.0169418116
## [30,] -8.476412e-03 -0.0082547174
## [31,]  0.000000e+00  0.0000000000
## [32,]  0.000000e+00  0.0000000000
## [33,]  0.000000e+00  0.0000000000
## [34,]  0.000000e+00  0.0000000000
## [35,]  0.000000e+00  0.0000000000
## [36,]  0.000000e+00  0.0000000000
## [37,]  0.000000e+00  0.0000000000
## [38,]  0.000000e+00  0.0000000000
## [39,]  0.000000e+00  0.0000000000
## [40,]  0.000000e+00  0.0000000000
## [41,]  0.000000e+00  0.0000000000
## [42,]  0.000000e+00  0.0000000000
## [43,]  0.000000e+00  0.0000000000
## [44,]  0.000000e+00  0.0000000000
## [45,]  0.000000e+00  0.0000000000
## [46,]  0.000000e+00  0.0000000000
## [47,]  0.000000e+00  0.0000000000
## [48,]  0.000000e+00  0.0000000000
## [49,]  0.000000e+00  0.0000000000
## [50,]  0.000000e+00  0.0000000000
## [51,]  0.000000e+00  0.0000000000
## [52,]  0.000000e+00  0.0000000000
## [53,]  0.000000e+00  0.0000000000
## [54,]  0.000000e+00  0.0000000000
## [55,]  0.000000e+00  0.0000000000
## [56,]  0.000000e+00  0.0000000000
## [57,]  0.000000e+00  0.0000000000
## [58,]  0.000000e+00  0.0000000000
## [59,]  0.000000e+00  0.0000000000
## [60,]  0.000000e+00  0.0000000000
```

Montando o loop para a previsão dinâmica, onde usamos o valor previsto para a próxima previsão

```r
int1 <- 1:length(etf)
o_data <- cbind(etf,ibov,int1)#original data
o_data2 <- cbind(etf,ibov,int1) #original data holder to compare later
t <- 1
for (t in 1:n1) {
    prv[n1+t,] <- t(
    vec$rlm[[1]][2,] + #constantes
    vec$rlm[[1]][1,] %*% #alphas
      t(vec$beta) %*% #cointegration vector
      o_data[n2-n1-1+t,]  + #Y_{t-1}
      t(vec$rlm[[1]][c(3,4),])%*% #\Gamma
```

```r
     prv[n1+t-1,]  )#\Delta Y_{t-1}

o_data[n2-n1+t,c(1,2)] <- prv[n1+t,] + o_data[n2-n1-1+t,c(1,2)] #to make dynamic refresh of data
}
```

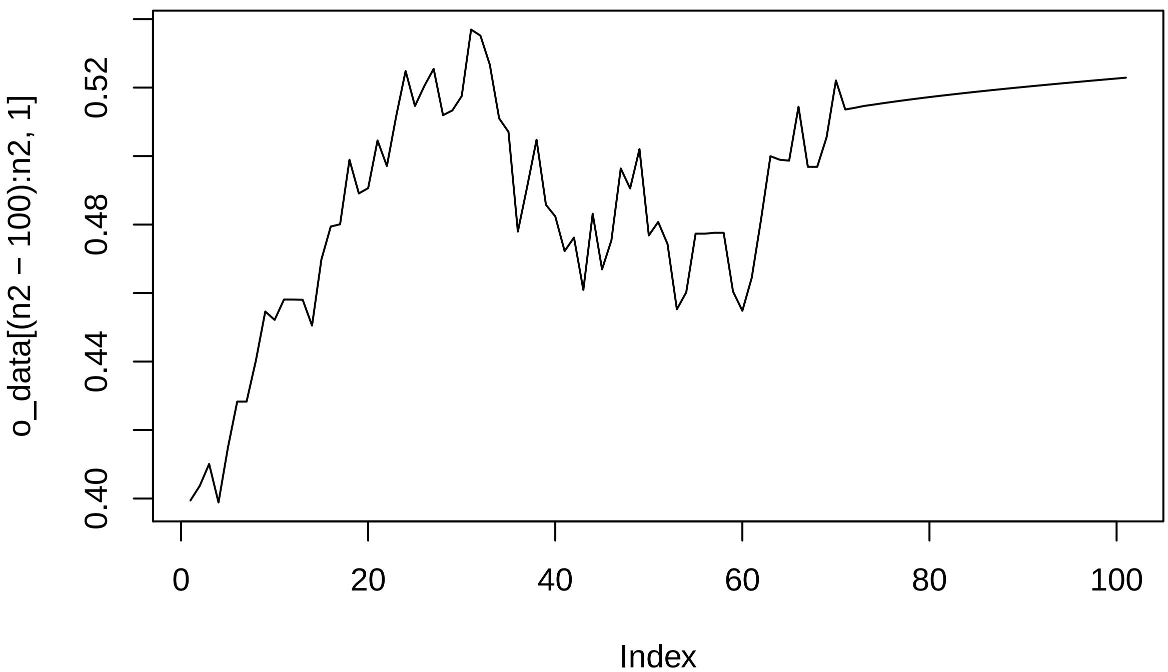
Podemos ver que os dados foram preenchidos

```r
prv
```

```
##              etf.d        ibov.d
##   [1,]  3.947030e-03  0.0039300008
##   [2,] -1.528501e-02 -0.0151478436
##   [3,]  2.225096e-02  0.0226453564
##   [4,] -1.624330e-02 -0.0166330385
##   [5,]  8.480730e-03  0.0082750506
##   [6,]  2.097671e-02  0.0216290827
##   [7,] -5.824805e-03 -0.0058751980
##   [8,]  1.147909e-02  0.0117497583
##   [9,] -2.520971e-02 -0.0256452106
##  [10,]  3.937100e-03  0.0035723462
##  [11,] -6.449320e-03 -0.0063425233
##  [12,] -1.905952e-02 -0.0196281377
##  [13,]  4.961734e-03  0.0050795792
##  [14,]  1.712385e-02  0.0174747360
##  [15,]  0.000000e+00  0.0000000000
##  [16,]  2.593062e-04  0.0002661391
##  [17,] -4.341894e-06 -0.0235607720
##  [18,] -1.713270e-02  0.0062531366
##  [19,] -5.642431e-03 -0.0058491007
##  [20,]  9.643728e-03  0.0098913792
##  [21,]  1.705683e-02  0.0178664353
##  [22,]  1.844564e-02  0.0196664259
##  [23,] -1.034703e-03 -0.0018572035
##  [24,] -2.617279e-04 -0.0006736425
##  [25,]  1.572418e-02  0.0168608638
##  [26,] -1.753872e-02 -0.0185499075
##  [27,] -4.259088e-06  0.0082490998
##  [28,]  8.610010e-03  0.0003307344
##  [29,]  1.662648e-02  0.0169418116
##  [30,] -8.476412e-03 -0.0082547174
##  [31,]  4.749998e-04  0.0005859283
##  [32,]  5.505314e-04  0.0002958116
##  [33,]  3.843566e-04  0.0002513390
##  [34,]  4.077706e-04  0.0002634270
##  [35,]  3.837369e-04  0.0002568525
##  [36,]  3.718804e-04  0.0002549892
##  [37,]  3.587832e-04  0.0002523627
##  [38,]  3.472283e-04  0.0002501473
##  [39,]  3.365162e-04  0.0002480709
##  [40,]  3.267004e-04  0.0002461730
##  [41,]  3.176816e-04  0.0002444281
##  [42,]  3.094003e-04  0.0002428262
##  [43,]  3.017951e-04  0.0002413550
##  [44,]  2.948110e-04  0.0002400040
##  [45,]  2.883972e-04  0.0002387633
```

```
## [46,]  2.825073e-04  0.0002376239
## [47,]  2.770983e-04  0.0002365776
## [48,]  2.721311e-04  0.0002356167
## [49,]  2.675695e-04  0.0002347343
## [50,]  2.633805e-04  0.0002339239
## [51,]  2.595335e-04  0.0002331797
## [52,]  2.560007e-04  0.0002324963
## [53,]  2.527564e-04  0.0002318687
## [54,]  2.497771e-04  0.0002312924
## [55,]  2.470411e-04  0.0002307631
## [56,]  2.445285e-04  0.0002302771
## [57,]  2.422211e-04  0.0002298307
## [58,]  2.401021e-04  0.0002294208
## [59,]  2.381562e-04  0.0002290444
## [60,]  2.363692e-04  0.0002286987
```

```r
plot(o_data[(n2-100):n2,1],type='l')
```




```r
plot(o_data[(n2-100):n2,2],type='l')
```

Vamos comparar os dados previstos com os dados verdadeiros:

```
(o_data - o_data2)[(n2-40):n2,]
```

```
##                etf         ibov int1
##  [1,]  0.000000000  0.000000000    0
##  [2,]  0.000000000  0.000000000    0
##  [3,]  0.000000000  0.000000000    0
##  [4,]  0.000000000  0.000000000    0
##  [5,]  0.000000000  0.000000000    0
##  [6,]  0.000000000  0.000000000    0
##  [7,]  0.000000000  0.000000000    0
##  [8,]  0.000000000  0.000000000    0
##  [9,]  0.000000000  0.000000000    0
## [10,]  0.000000000  0.000000000    0
## [11,]  0.000000000  0.000000000    0
## [12,]  0.009538766  0.009686299    0
## [13,] -0.026522739 -0.027324959    0
## [14,] -0.046436208 -0.047223049    0
## [15,] -0.041898382 -0.043102344    0
## [16,] -0.034310650 -0.035206155    0
## [17,] -0.077412221 -0.079641244    0
## [18,] -0.077751899 -0.079431049    0
## [19,] -0.048759398 -0.050805584    0
## [20,] -0.040197598 -0.041457341    0
## [21,] -0.039870897 -0.041211168    0
## [22,] -0.044755068 -0.046243247    0
```

```
## [23,] -0.071838113 -0.073892424    0
## [24,] -0.072001281 -0.074192118    0
## [25,] -0.049387228 -0.051348383    0
## [26,] -0.053543259 -0.055543934    0
## [27,] -0.068925758 -0.071523587    0
## [28,] -0.065478523 -0.067815201    0
## [29,] -0.038768450 -0.041010759    0
## [30,] -0.050775228 -0.052980399    0
## [31,] -0.069378098 -0.072020774    0
## [32,] -0.046031396 -0.049096570    0
## [33,] -0.082308803 -0.085063904    0
## [34,] -0.087256229 -0.091003156    0
## [35,] -0.098804301 -0.102094610    0
## [36,] -0.098557260 -0.101863847    0
## [37,] -0.111399802 -0.114880910    0
## [38,] -0.100851949 -0.104225728    0
## [39,] -0.089973976 -0.093172386    0
## [40,] -0.065880297 -0.068778366    0
## [41,] -0.065643928 -0.068795840    0
```

Vamos calcular o erro quadrático médio de previsão dinâmica usando os valores estimados dos coeficientes de ajuste, sem considerar exogeneidade fraca.

Para etf temos

```
eqm_dyn_beta_estimado_etf <- mean(((o_data - o_data2)[(n2-n1+1):n2,][,1])^2)
eqm_dyn_beta_estimado_etf
```

```
## [1] 0.004479993
```

Para ibov temos

```
eqm_dyn_beta_estimado_ibov <- mean(((o_data - o_data2)[(n2-n1+1):n2,][,2])^2)
eqm_dyn_beta_estimado_ibov
```

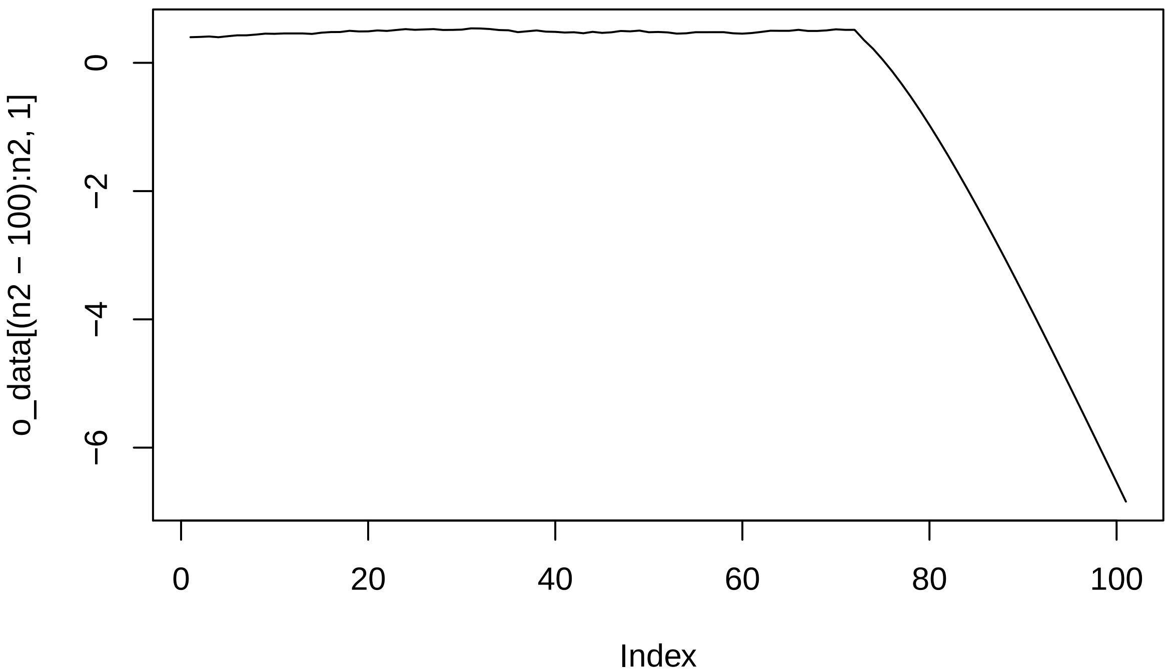
```
## [1] 0.004798857
```

Vamos agora calcular a previsão dinâmica mas agora considerando que ibov é fracamente exógeno.

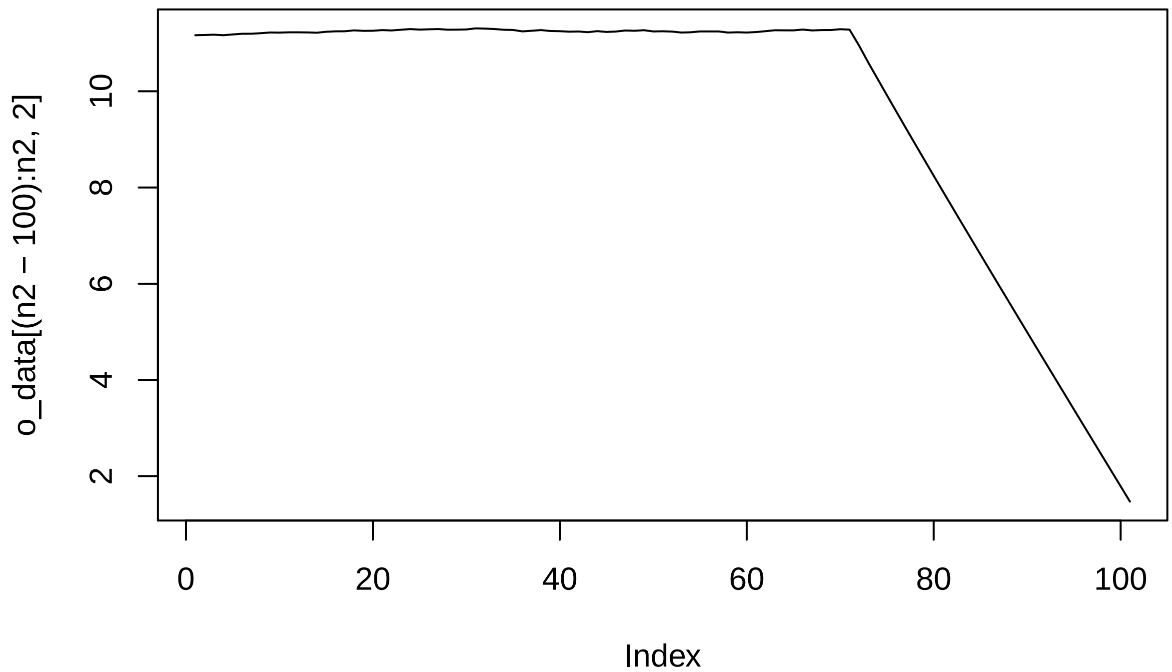
```
o_data <- cbind(etf,ibov,int1)#original data
o_data2 <- cbind(etf,ibov,int1) #original data holder to compare later
t <- 1
for (t in 1:n1) {
    prv[n1+t,] <- t(
    vec$rlm[[1]][2,] + #constantes
    c(vec$rlm[[1]][1,1],0) %*% #alphas  <- vamos alterar o alfa para ibov
      t(vec$beta) %*% #cointegration vector
      o_data[n2-n1-1+t,]  + #Y_{t-1}
      t(vec$rlm[[1]][c(3,4),])%*% #\Gamma
      prv[n1+t-1,]  )#\Delta Y_{t-1}

o_data[n2-n1+t,c(1,2)] <- prv[n1+t,] + o_data[n2-n1-1+t,c(1,2)] #to make dynamic refresh of data
}
```

```
plot(o_data[(n2-100):n2,1],type='l')
```




```
plot(o_data[(n2-100):n2,2],type='l')
```



Vamos calcular o erro quadrático médio de previsão dinâmica usando os valores estimados dos coeficientes de ajuste quando consideramos que ibov é fracamente exógeno.

Para etf temos

```
eqm_dyn_beta_estimado_etf <- mean(((o_data - o_data2)[(n2-n1):n2,][,1])^2)
eqm_dyn_beta_estimado_etf
```

```
## [1] 16.10504
```

Para ibov temos

```
eqm_dyn_beta_estimado_ibov <- mean(((o_data - o_data2)[(n2-n1):n2,][,2])^2)
eqm_dyn_beta_estimado_ibov
```

```
## [1] 33.96264
```

Previsão estática

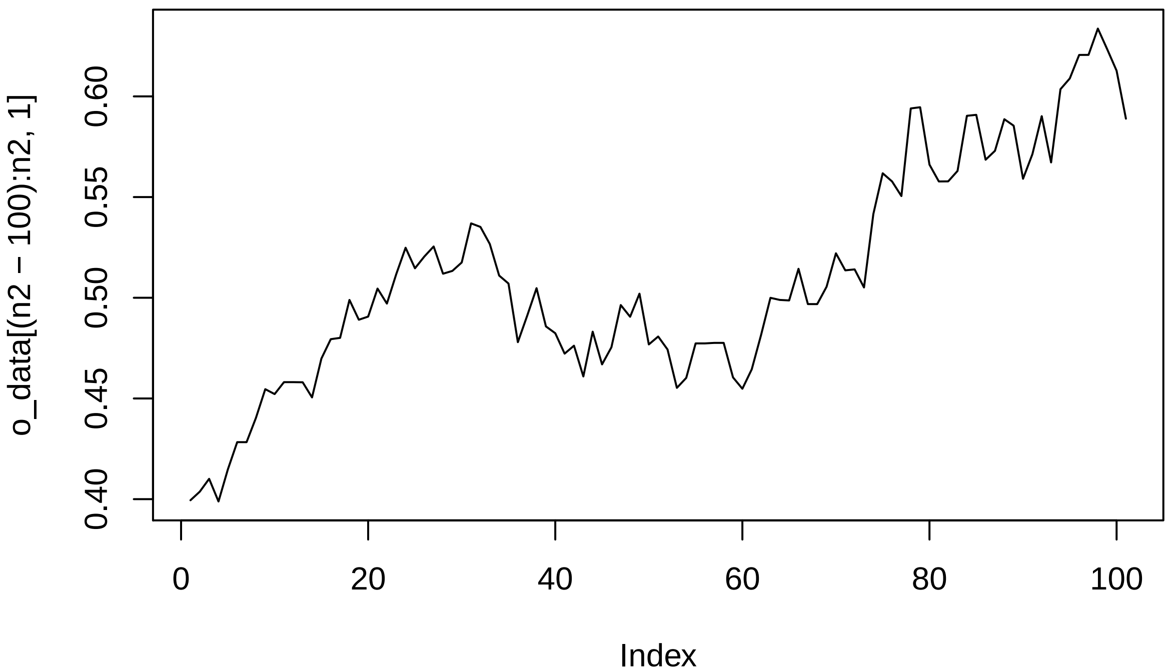
Vamos fazer agora a previsão estática, onde consideramos sempre um passo a frente o dados observado. E considerando que ibov não é fracamente exógeno.

```
o_data <- cbind(etf,ibov,int1)#original data
o_data2 <- cbind(etf,ibov,int1) #original data holder to compare later
t <- 1
for (t in 1:n1) {
    prv[n1+t,] <- t(
    vec$rlm[[1]][2,] + #constantes
```

```r
    vec$rlm[[1]][1,] %*% #alphas
      t(vec$beta) %*% #cointegration vector
      o_data2[n2-n1-1+t,]  + #Y_{t-1} #estamos usando um vetor que não é atualizado, mas sim os valores
      t(vec$rlm[[1]][c(3,4),])%*% #\Gamma
      prv[n1+t-1,]  )#\Delta Y_{t-1}

o_data[n2-n1+t,c(1,2)] <- prv[n1+t,] + o_data2[n2-n1-1+t,c(1,2)] #to make dynamic refresh of data
}
```

```r
plot(o_data[(n2-100):n2,1],type='l')
```




```r
plot(o_data[(n2-100):n2,2],type='l')
```

Vamos calcular o erro quadrático médio de previsão dinâmica usando os valores estimados dos coeficientes de ajuste sem considerarmos que ibov é fracamente exógeno.

Para etf temos

```
eqm_dyn_beta_estimado_etf <- mean(((o_data - o_data2)[(n2-n1):n2,][,1])^2)
eqm_dyn_beta_estimado_etf
```

```
## [1] 0.0003340067
```

Para ibov temos

```
eqm_dyn_beta_estimado_ibov <- mean(((o_data - o_data2)[(n2-n1):n2,][,2])^2)
eqm_dyn_beta_estimado_ibov
```

```
## [1] 0.000342492
```

Previsão estática, considerando fracamente exógeno

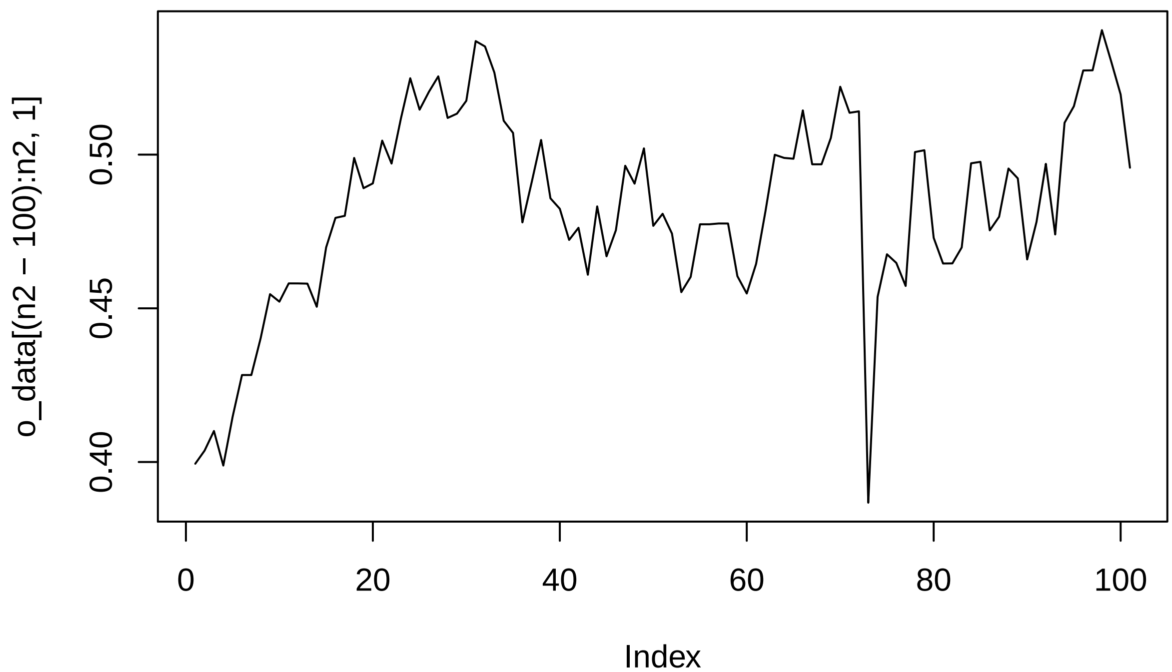
Vamos fazer agora a previsão estática, onde consideramos sempre um passo a frente o dados observado

```
o_data <- cbind(etf,ibov,int1)#original data
o_data2 <- cbind(etf,ibov,int1) #original data holder to compare later
t <- 1
for (t in 1:n1) {
    prv[n1+t,] <- t(
    vec$rlm[[1]][2,] + #constantes
    c(vec$rlm[[1]][1,1],0) %*% #alphas
```

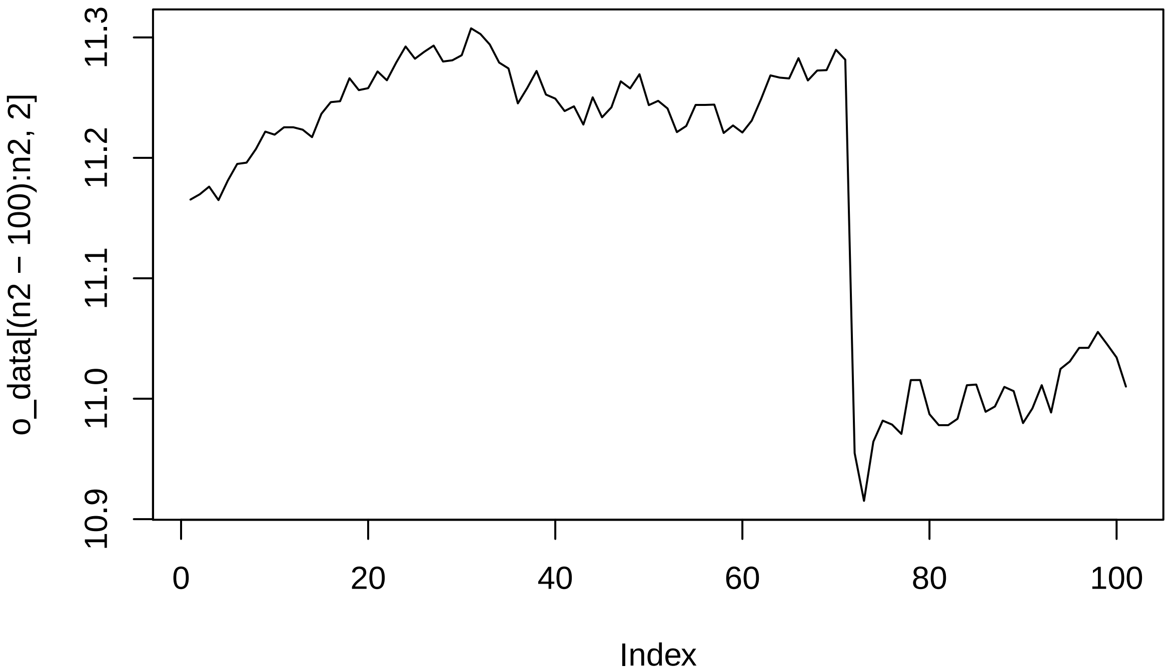
```
      t(vec$beta) %*% #cointegration vector
      o_data2[n2-n1-1+t,]  + #Y_{t-1} #estamos usando um vetor que não é atualizado, mas sim os valores
      t(vec$rlm[[1]][c(3,4),])%*% #\Gamma
      prv[n1+t-1,]  )#\Delta Y_{t-1}

o_data[n2-n1+t,c(1,2)] <- prv[n1+t,] + o_data2[n2-n1-1+t,c(1,2)] #to make dynamic refresh of data
}
```

```
plot(o_data[(n2-100):n2,1],type='l')
```




```
plot(o_data[(n2-100):n2,2],type='l')
```



Vamos calcular o erro quadrático médio de previsão dinâmica usando os valores estimados dos coeficientes de ajuste considerando que ibov é fracamente exógeno.

Para etf temos

```
eqm_dyn_beta_estimado_etf <- mean(((o_data - o_data2)[(n2-n1):n2,][,1])^2)
eqm_dyn_beta_estimado_etf
```

```
## [1] 0.009092334
```

Para ibov temos

```
eqm_dyn_beta_estimado_ibov <- mean(((o_data - o_data2)[(n2-n1):n2,][,2])^2)
eqm_dyn_beta_estimado_ibov
```

```
## [1] 0.1187443
```

Respondemos a questão **2. f) Obtenha as previsões estáticas e dinâmicas para as 30 últimas observações da amostra, considerando que ibovespa é exógena fraca e não considerando esta hipótese. Compare os resultados usando erro quadrático médio de previsão.** Vimos que o erro quadrático médio de previsão estática é melhor que seus correspondentes na previsão dinâmica, mas considerando que ibov é fracamente exógeno torna a previsão mais imprecisa.

2. g) Discuta a importância de exogeneidade fraca e causalidade de Granger no processo de previsão fora da amostra.

Questão 4

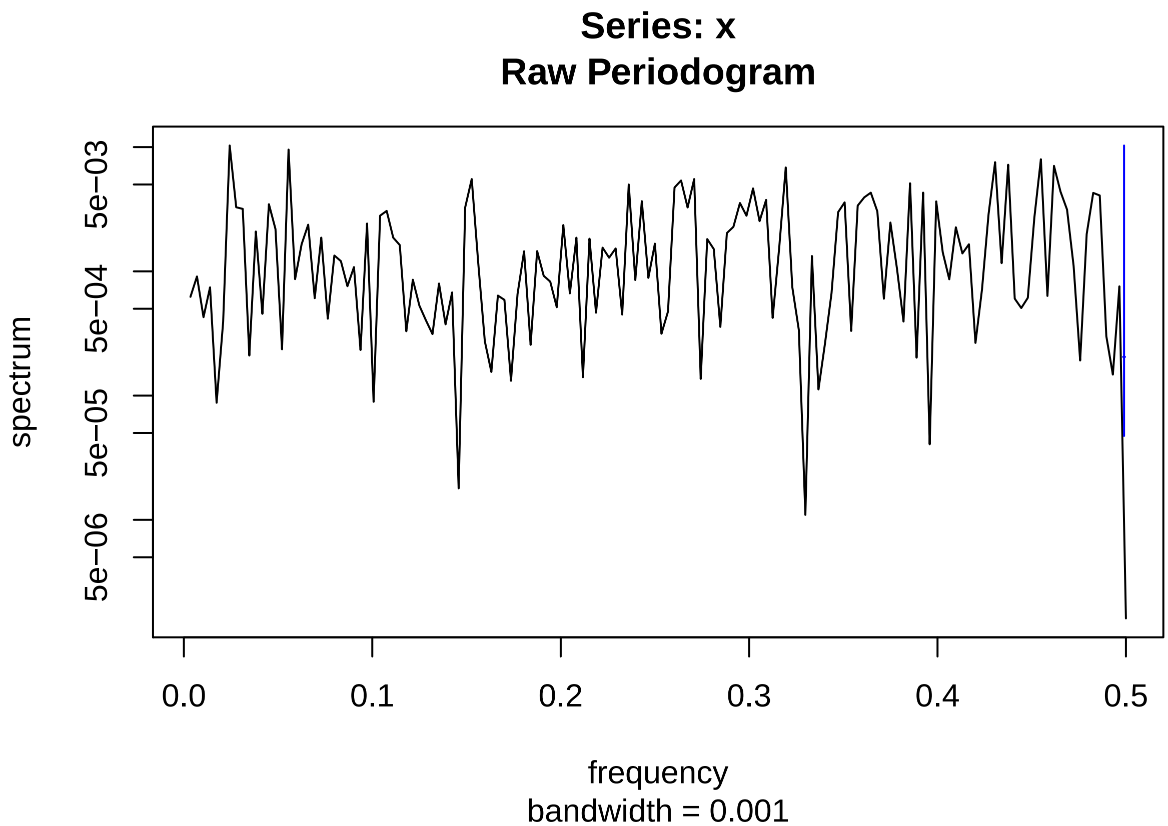
** Para a taxa de crescimento da produção industrial ** Vamos criar a variável taxa de crescimento

```r
gind <- diff(indtr)
head(gind)
```

```
## [1]  0.019992977 -0.005981564  0.044196433  0.032915884  0.091413554
## [6] -0.001079098
```

4. a) Obtenha uma estimação da densidade espectral desta série e interprete seus resultados

```r
spectrum(gind)
```



Parece que a série de crescimento industrial é composta por um quantidade variada de frequências, sem estar apontandando para alguma frequência ou faixas de frequências que melhor representa a série.

4. b) Realize a estimação de não-paramétrica da variância de longo prazo desta série, usando um estimador de Newey-West e um estimador Fixed-b. Discuta a diferença entre esses estimadores.

```r
var(gind)
```

```
## [1] 0.002010446
```

```r
m1 <- lm(gind~1)
b <- length(gind)
(length(gind)-1) * kernHAC( m1, kernel = "Quadratic Spectral", bw = bwNeweyWest) #estimadores Newey-Wes
```

```
##             (Intercept)
## (Intercept) 0.001915876
```

```r
(length(gind)-1) * kernHAC( m1, kernel = "Quadratic Spectral", bw = b)  #fixe-b
```

```
##              (Intercept)
## (Intercept) 9.892394e-06
```

```r
(length(gind)-1) * kernHAC( m1, kernel = "Bartlett", bw = bwNeweyWest)#estimadores Newey-West
```

```
##             (Intercept)
## (Intercept) 0.001815066
```

```r
(length(gind)-1) * kernHAC( m1, kernel = "Bartlett", bw = b) #fixe-b
```

```
##              (Intercept)
## (Intercept) 0.0001404447
```

```r
(length(gind)-1) * kernHAC( m1, kernel = "Parzen", bw = bwNeweyWest)#estimadores Newey-West
```

```
##             (Intercept)
## (Intercept) 0.001907289
```

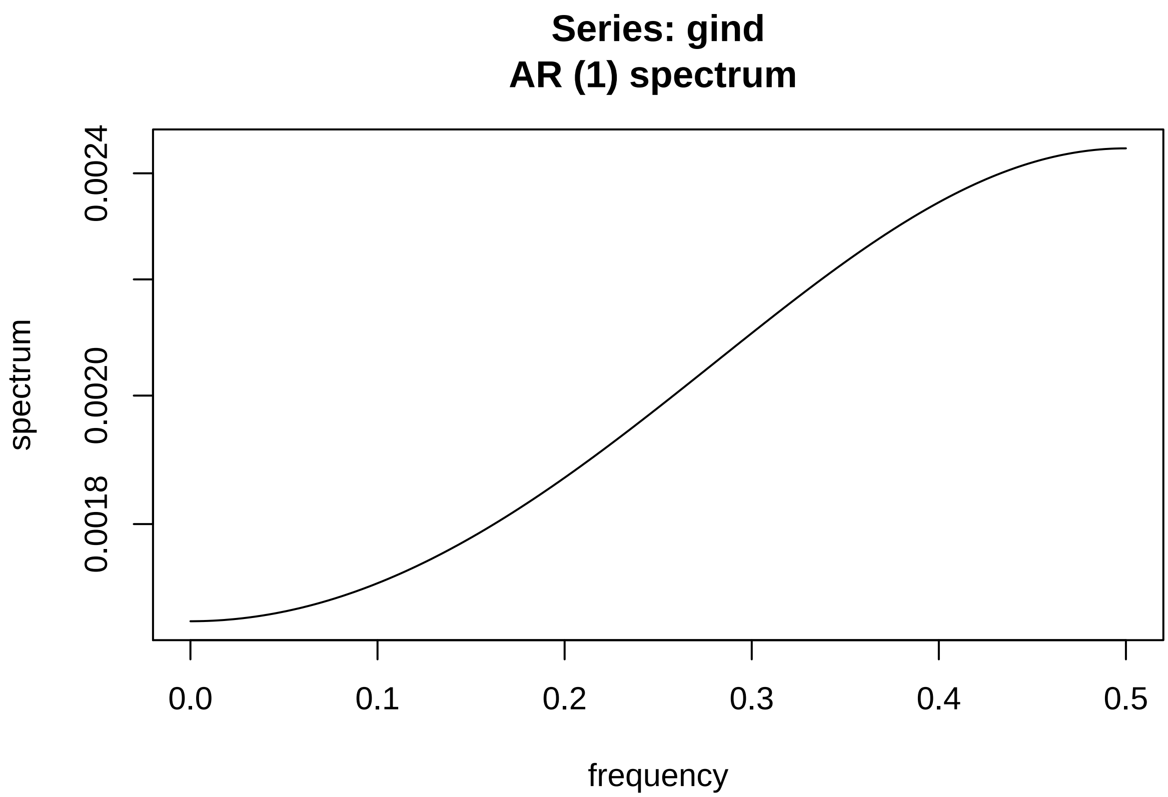
```r
(length(gind)-1) * kernHAC( m1, kernel = "Parzen", bw = b) #fixe-b
```

```
##              (Intercept)
## (Intercept) 0.0001195931
```

Vemos que os estimadores fixed-b são menores do que os de Newey-West. E a variância de longo prazo encontrada pelos métodos de Newey-West são próximos da variância amostral.

4. c) Compare o resultado do item anterior com a variância de longo prazo usando um estimador paramétrico

```r
l1 <- spec.ar(gind)
```




```r
l2<- ar(gind, order.max=10)
l2 #order selected = 1
```

```
##
## Call:
## ar(x = gind, order.max = 10)
##
## Coefficients:
##       1
## -0.0967
##
## Order selected 1  sigma^2 estimated as  0.001999
```

```r
mean(resid(l2)[!is.na(resid(l2))]^2)/(1-l2$ar^2) #variancia estimada pelo AR(1)
```

```
## [1] 0.002008547
```

O resultado paramétrico é proximo dos estimadores Newey-West.