Analise de Series Temporais - Trabalho 1

Gustavo Hotta (A56865193), Rafael Furlan (A56871487), Ricardo Squassina Lee (A56843646)

1- Utilizando o arquivo “Serie\_Dados.csv” realize as seguintes etapas:

Serie\_Dados <- read.csv("Serie\_Dados.csv", sep=";")  
#Serie\_Dados

1. Crie a série temporal dos retornos Ln, ou seja, r=Ln(P\_t+1 /P\_t)

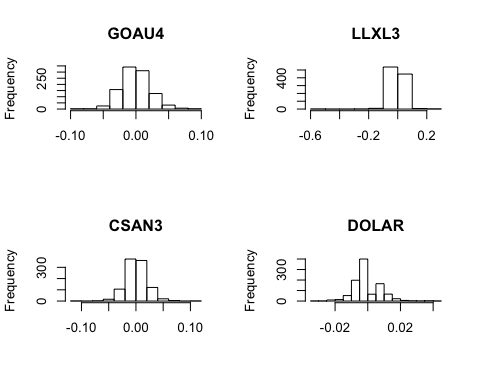
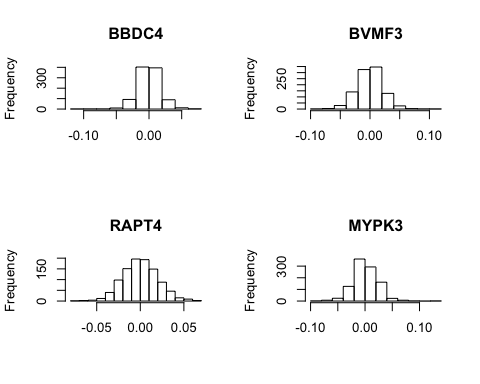
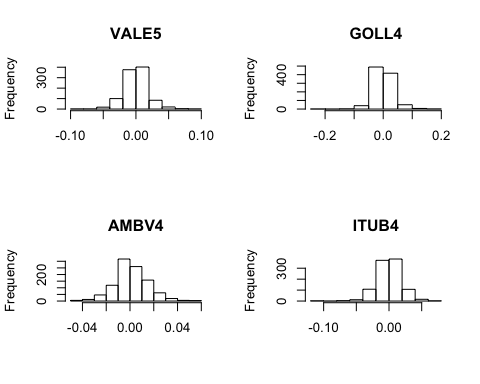
Serie\_Dados.LN <- log(Serie\_Dados[2:13]/rbind(NA,Serie\_Dados[2:13][-nrow(Serie\_Dados[2:13]),]))  
Serie\_Dados.LN <- Serie\_Dados.LN[-1,]  
#Serie\_Dados.LN

A maior parte dos estudos financeiros concentram-se na análise de séries temporal dos retornos ao invés do uso da série dos preços dos ativos. A razão de utilizarmos série de retornos tem dois fatores, as informações de retornos atendem aos interesses de investidores e a série de retornos possui propriedades estatísticas mais interessantes do que séries dos preços.

1. Para cada ação construa o histograma dos retornos. Comente o resultado dos histogramas, verifique também o desvio padrão e a média de cada série

Histogramas:

par(mfrow = c(2, 2))  
  
  
for (col in 1:ncol(Serie\_Dados.LN)) {  
 hist(Serie\_Dados.LN[,col], main = names(Serie\_Dados.LN[col]), xlab = "")  
}



Desvio Padrao e Media:

sapply(Serie\_Dados.LN, function(cl) list(Media=mean(cl,na.rm=TRUE), DesvioPadrao=sd(cl,na.rm=TRUE)))

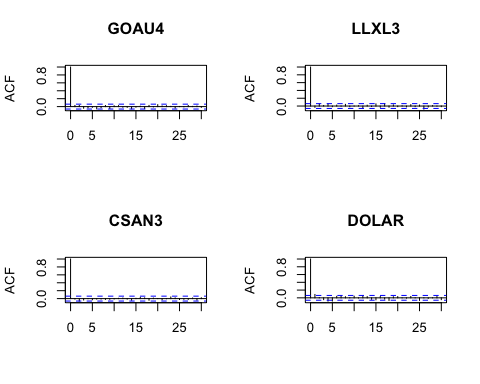
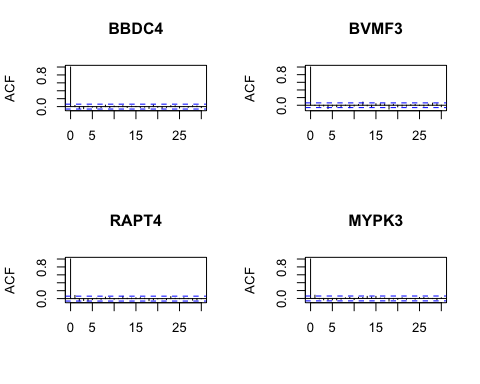
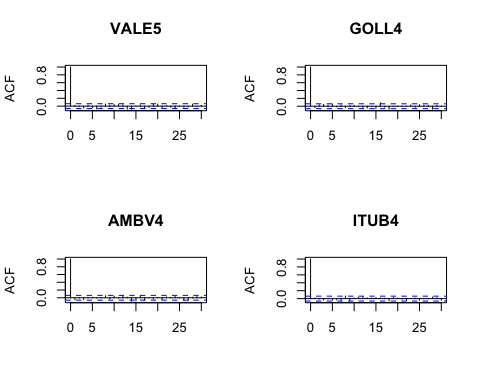
## VALE5 GOLL4 AMBV4 ITUB4   
## Media 0.0001293889 -0.000491604 0.001271904 -3.197417e-05  
## DesvioPadrao 0.01838984 0.0324803 0.01426466 0.01833354   
## BBDC4 BVMF3 RAPT4 MYPK3   
## Media 0.000209532 9.431336e-05 0.0003498633 0.001024337  
## DesvioPadrao 0.01708826 0.02193809 0.02011623 0.02255594   
## GOAU4 LLXL3 CSAN3 DOLAR   
## Media -0.0001969214 -0.001223711 0.0008827886 0.0002516147  
## DesvioPadrao 0.02253222 0.04117323 0.01928199 0.007719852

Graficamente, as variações dos retornos da carteira apresentam uma distribuição normal. Os melhores retornos para os piores são: AMBV4; MYPK3; BVMF3; CSAN3; RAPT4; DÓLAR; BBDC4; VALE5; ITUB4; GOAU4; GOLL4; LLXL3. Os últimos quatro retornos são negativos. Os maiores desvios padrões para os menores são: LLXL3; GOLL4; MYPK3; GOAU4; BVMF3; RAPT4; CSAN3; VALE5; ITUB4; BBDC4; AMBV4; DÓLAR. Podemos analisar que o melhor e mais seguro retorno é a AMBR4, porque tem o melhor retorno e um menor desvio padrão.

1. Calcule o ACF e PACF de cada série de retornos. Comente os resultados.

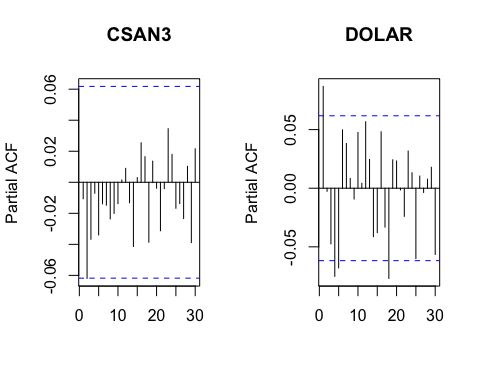
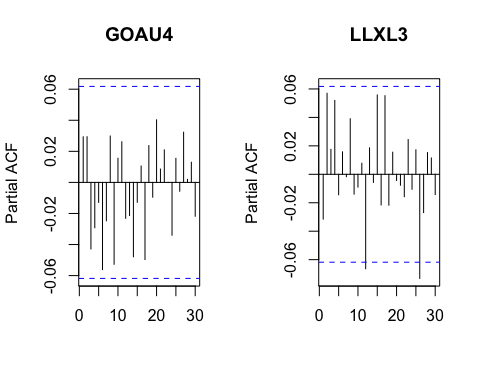
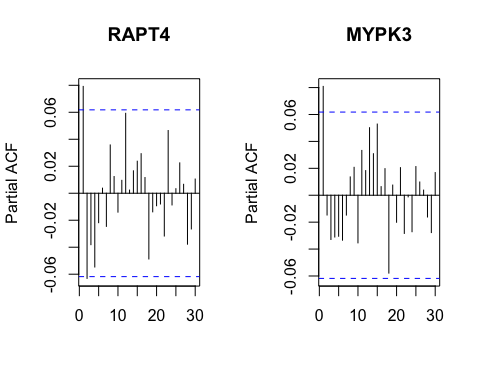
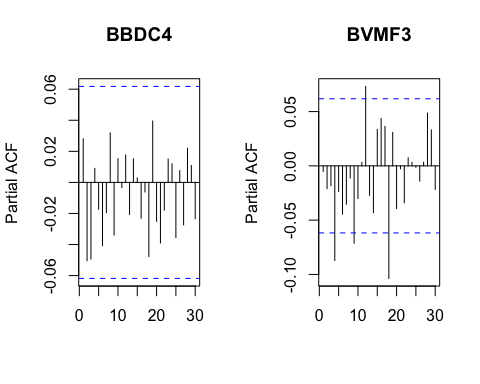
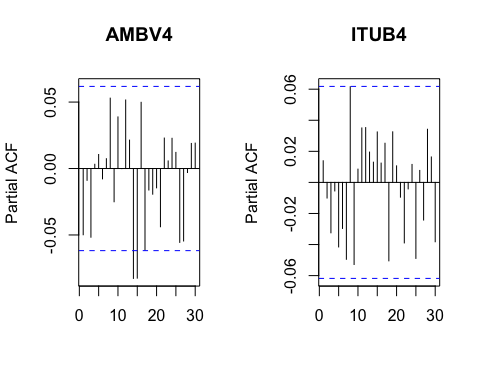
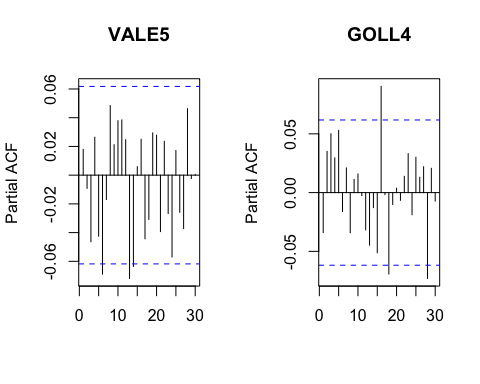
ACF

par(mfrow = c(2, 2))  
  
for (col in 1:ncol(Serie\_Dados.LN)) {  
 acf(Serie\_Dados.LN[,col], main = names(Serie\_Dados.LN[col]), xlab = "")  
}



PACF

par(mfrow = c(1, 2))  
  
for (col in 1:ncol(Serie\_Dados.LN)) {  
 pacf(Serie\_Dados.LN[,col], main = names(Serie\_Dados.LN[col]), xlab = "")  
}



Nos gráficos de ACF, podemos analisar que todos os retornos convergem rapidamente para zero, então podemos concluir que eles são estacionários como AR(p). O PACF (função de autocorrelação parcial) nos dá correlação entre a variável no instante t e uma de suas defasagens, retirando os efeitos das outras defasagens. Graficamente cada retorno apresenta um padrão.

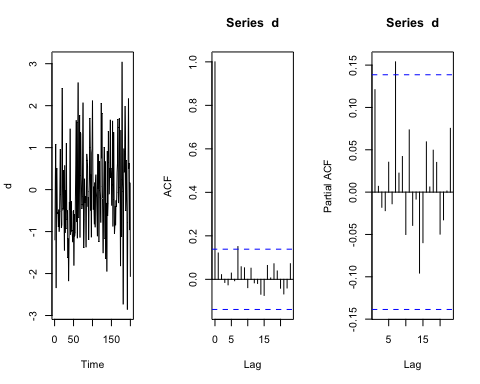
2- Para cada um dos processos abaixo gere 200 observações. Faça um gráfico da série, ACF e PACF. Comente os resultados.

Definindo uma semente para os numeros aleatorios serem sempre os mesmos:

set.seed(1234)

1. Série aleatória, observações iid da distribuição N(0,1)

par(mfrow = c(1, 3))  
  
d <- ts(rnorm(200, 0, 1))  
plot(d)  
acf(d)  
pacf(d)

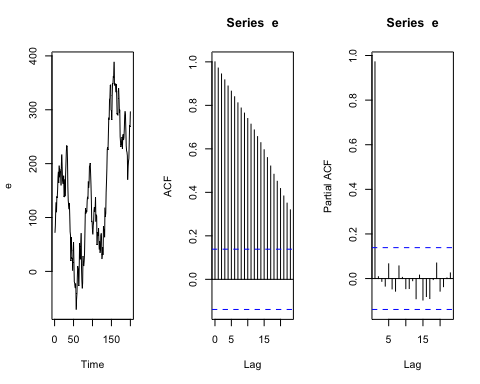


A serie é estacionária, mas por ser iid tem a PACF igual a 0

1. Série com tendência estocástica,

Neste caso o coeficiente tem que ser menor que 1 para rodar, então ar = 0.99999

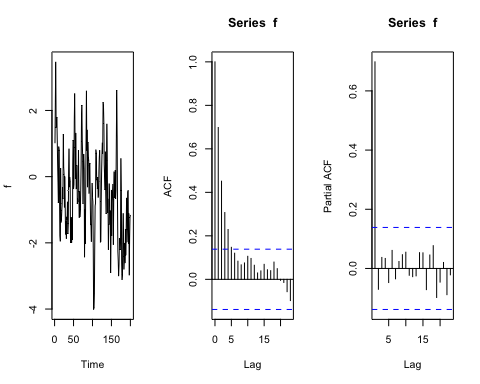
par(mfrow = c(1, 3))  
  
e <- arima.sim(model = list(ar= 0.99999), n=200, innov = rnorm(200,1,25))  
plot(e)  
acf(e)  
pacf(e)



A série não é estacionaria

1. Serie com correlação de curto-prazo,

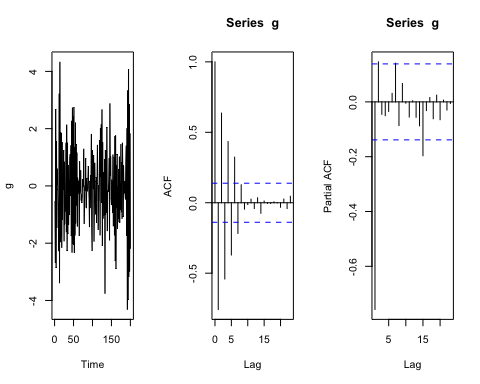
par(mfrow = c(1, 3))  
  
f <- arima.sim(model = list(ar = 0.7), n = 200, innov = rnorm(200, 0, 1))  
plot(f)  
acf(f)  
pacf(f)



A série é estacionária, ACF com decaimento e grafico de pacf com pico em 1

1. Série com correlações negativas

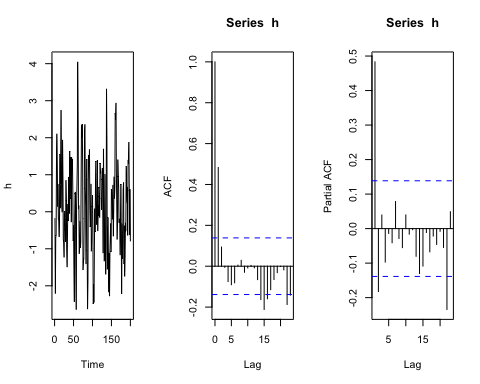
par(mfrow = c(1, 3))  
  
g <- arima.sim(model = list(ar = -0.8), n = 200, innov = rnorm(200, 0, 1))  
plot(g)  
acf(g)  
pacf(g)



A série é estacionária, acf com decaimento oscilando, pacf pico em 1

1. Medias moveis,

par(mfrow = c(1, 3))  
  
h <- arima.sim(model = list(ma = 0.6), n = 200, innov = rnorm(200, 0, 1))  
plot(h)  
acf(h)  
pacf(h)



A série é estacionaria, acf igual a 0 em K>1 e pacf com decaimento oscilando

3- Utilize a série abaixo para resolver cada item.

An example of a time series that can probably be described using an additive model with a trend and no seasonality is the time series of the annual diameter of women’s skirts at the hem, from 1866 to 1911. The data is available in the file <http://robjhyndman.com/tsdldata/roberts/skirts.dat> (original data from Hipel and McLeod, 1994).

skirts <- read.table("http://robjhyndman.com/tsdldata/roberts/skirts.dat", header = TRUE, skip = 3)

1. Faça a leitura da série de dados e os tratamentos necessários para considerar a mesma como uma série temporal

skirts.ts<-ts(skirts, frequency=1, start=c(1866))  
#skirts.ts

1. Faça a decomposição da série do item (a): Sazonalidade, Tendência e Aleatória.

skirts.components <- ifelse(frequency(skirts.ts)>1,  
 decompose(skirts.ts,type = c("additive", "multiplicative")),  
 print("Nao e' possivel decompor uma serie anual, para ser feita a decomposicao a serie deveria ter, no minimo, 2 periodos"))

## [1] "Nao e' possivel decompor uma serie anual, para ser feita a decomposicao a serie deveria ter, no minimo, 2 periodos"

#plot(skirts.components)

1. Calcule a ACF e PACF da série e da primeira diferença

par(mfrow = c(2, 2))  
  
acf(skirts.ts)  
acf(diff(skirts.ts))  
pacf(skirts.ts)  
pacf(diff(skirts.ts))



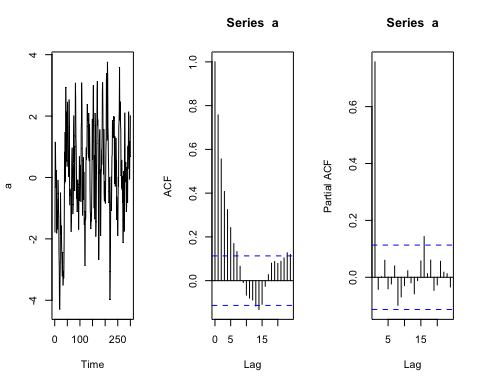
Os ACF da série e da primeira diferença não apresentam muita diferença quando comparados entre si. No gráfico PACF na série apresenta um lag com valor negativo no tempo 2 e na primeira diferença apresenta um lag com valor positivo no tempo 6.

4- Usando a função arima.sim gere as seguintes simulações (300 ptos):

set.seed(1234)

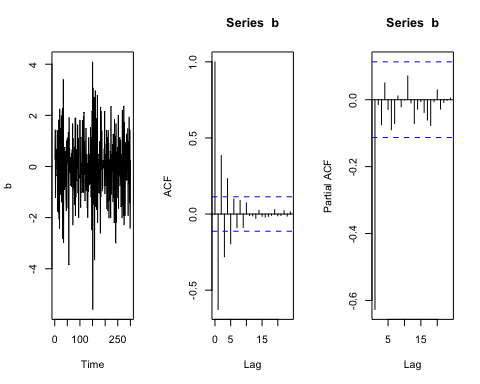
1. Processo AR(1) onde θ0=0, θ1=0.7

par(mfrow = c(1, 3))  
  
a <- arima.sim(n=300,list(ar = c(.7)))  
plot(a)  
acf(a)  
pacf(a)



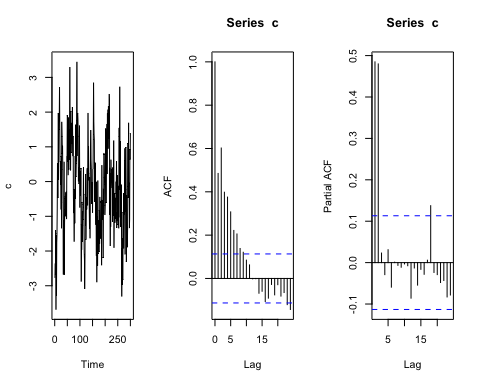
1. Processo AR(1) onde θ0=0, θ1= -0.7

par(mfrow = c(1, 3))  
  
b <- arima.sim(n=300,list(ar = c(-.7)))  
plot(b)  
acf(b)  
pacf(b)



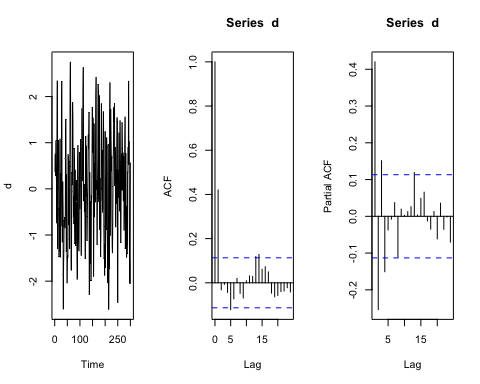
1. Processo AR(2) onde θ0=0, θ1=0.3 e θ2=0.5

par(mfrow = c(1, 3))  
  
c <- arima.sim(n=300,list(ar = c(.3,.5)))  
plot(c)  
acf(c)  
pacf(c)



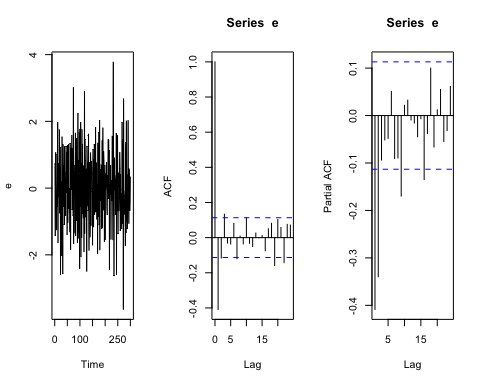
1. Processo MA(1) onde θ0=0, θ1=0.6

par(mfrow = c(1, 3))  
  
d <- arima.sim(n=300,list(ma = c(.6)))  
plot(d)  
acf(d)  
pacf(d)



1. Processo MA(1) onde θ0=0, θ1=-0.6

par(mfrow = c(1, 3))  
  
e <- arima.sim(n=300,list(ma = c(-.6)))  
plot(e)  
acf(e)  
pacf(e)



Para cada simulação, plote o gráfico da série, calcule o ACF e PACF. Usando estes resultados conclua como deve ser o comportamento da ACF de PACF de um modelo autoregressivo( AR.)

5- Obtenha a série histórica do PIB Brasil no site: <http://www.bcb.gov.br/pre/portalCidadao/cadsis/series.asp?idpai=PORTALBCB> Código da série: 1232

Lendo o arquivo do PIB (Editado fora do R por comodidade)

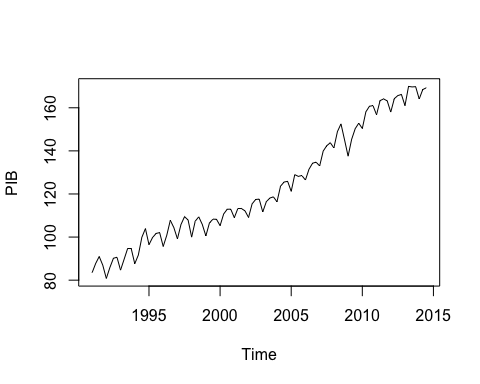
PIB <- read.csv("PIB2.csv")  
#PIB

Criando uma Serie Temporal baseado no PIB (data inicio 1 Trimestre de 1991 e frequencia trimestral)

PIB.ts<-ts(PIB, frequency=4, start=c(1991,1))  
#PIB.ts

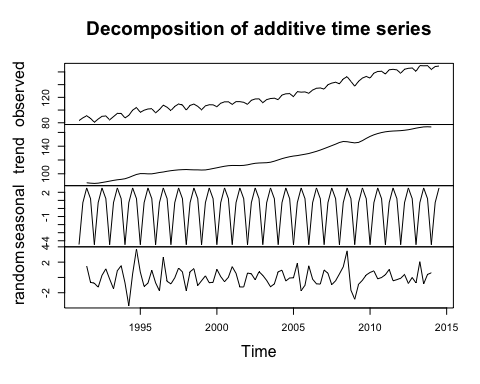
1. Plote o gráfico da série usando o R

plot.ts(PIB.ts)



1. Faça a decomposição da série em: Sazonalidade, Tendência e Aleatória.

PIB.decomposto <- decompose(PIB.ts)  
#PIB.decomposto  
plot(PIB.decomposto)



1. Usando o índice dos últimos 12 anos, encontre uma projeção para o PIB(índice) do próximo semestre usando um modelo AR(1). Neste caso use a série das diferenças.

Usando Predict:

if(!require(forecast)) {  
 install.packages("forecast")  
 library(forecast)  
}

## Loading required package: forecast

PIB.dif <- diff(PIB.ts[(nrow(PIB.ts)-12\*4):nrow(PIB.ts)])  
PIB.predict <- predict(auto.arima(PIB.dif),ahead = 2)  
PIB.predict

## $pred  
## Time Series:  
## Start = 49   
## End = 49   
## Frequency = 1   
## [1] 3.342682  
##   
## $se  
## Time Series:  
## Start = 49   
## End = 49   
## Frequency = 1   
## [1] 3.779094

Usando Forecast:

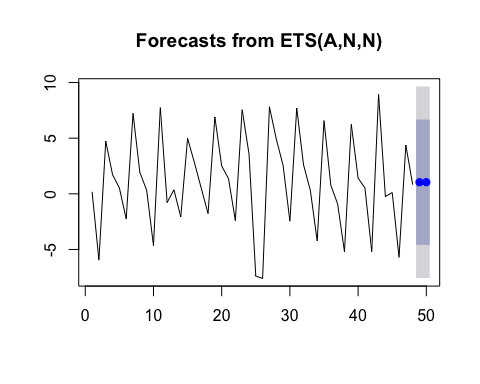
etsfit.PIB.dif <- ets(PIB.dif)  
etsfit.PIB.dif

## ETS(A,N,N)   
##   
## Call:  
## ets(y = PIB.dif)   
##   
## Smoothing parameters:  
## alpha = 1e-04   
##   
## Initial states:  
## l = 1.0451   
##   
## sigma: 4.3916  
##   
## AIC AICc BIC   
## 331.8250 332.3705 337.4386

accuracy(etsfit.PIB.dif)

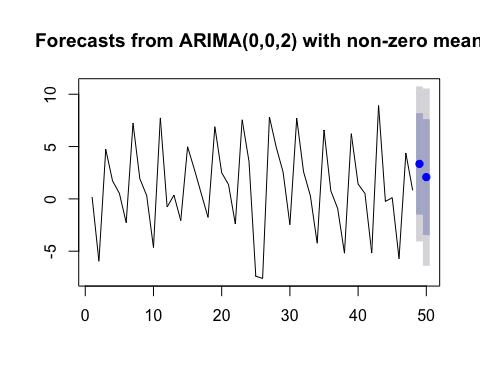
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.03456298 4.299119 3.433323 38.41953 136.8852 0.6138396  
## ACF1  
## Training set -0.2304208

fcast.PIB.dif <- forecast(etsfit.PIB.dif, h = 2, level = c(80,95))  
#fcast.PIB.dif  
plot(fcast.PIB.dif)



Usando ARIMA:

arimafit.PIB.dif <- auto.arima(PIB.dif)  
fcast.ARIMA.PIB.dif <- forecast(arimafit.PIB.dif, h = 2, level = c(80,95))  
#fcast.ARIMA.PIB.dif  
plot(fcast.ARIMA.PIB.dif)



Usando Forecast (sem usar a diferença):

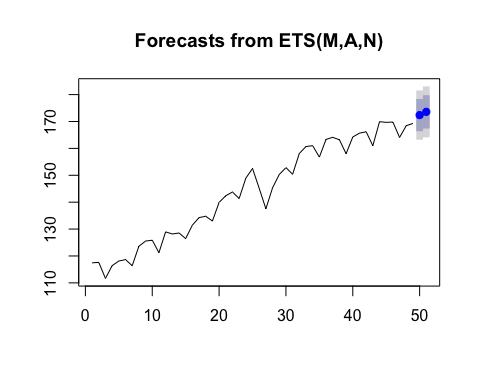
etsfit.PIB.ts <- ets(PIB.ts[(nrow(PIB.ts)-12\*4):nrow(PIB.ts)])  
etsfit.PIB.ts

## ETS(M,A,N)   
##   
## Call:  
## ets(y = PIB.ts[(nrow(PIB.ts) - 12 \* 4):nrow(PIB.ts)])   
##   
## Smoothing parameters:  
## alpha = 0.2265   
## beta = 1e-04   
##   
## Initial states:  
## l = 113.1102   
## b = 1.2088   
##   
## sigma: 0.0271  
##   
## AIC AICc BIC   
## 329.4144 330.8098 338.8735

accuracy(etsfit.PIB.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.1070473 3.774304 2.966711 -0.1073658 2.082458 0.8403785  
## ACF1  
## Training set 0.1977401

fcast.PIB.ts <- forecast(etsfit.PIB.ts, h = 2, level = c(80,95))  
#fcast.PIB.ts  
plot(fcast.PIB.ts)



Usando ARIMA (sem usar a diferença):

arimafit.PIB.ts <- auto.arima(PIB.ts[(nrow(PIB.ts)-12\*4):nrow(PIB.ts)])  
fcast.ARIMA.PIB.ts <- forecast(arimafit.PIB.ts, h = 2, level = c(80,95))  
#fcast.ARIMA.PIB.ts  
plot(fcast.ARIMA.PIB.ts)

