Analise de Series Temporais - Trabalho 1

Iniciando o ambiente

cat("\014")

rm(list = ls())

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

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

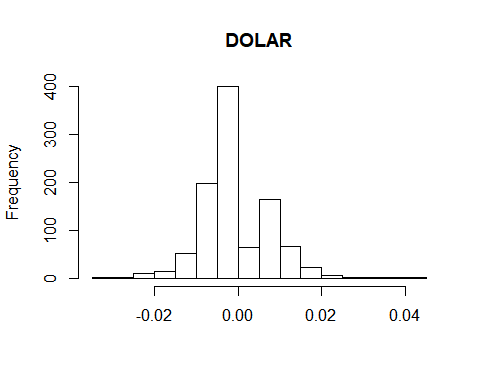
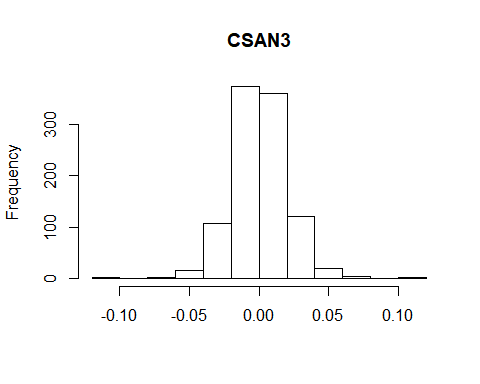
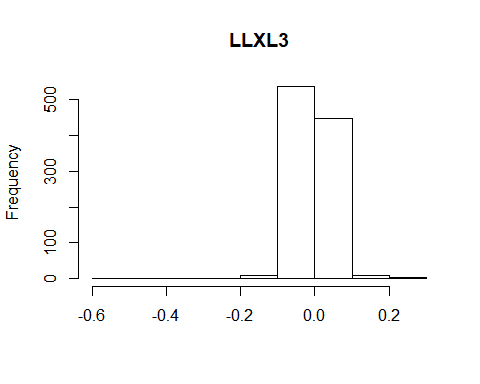
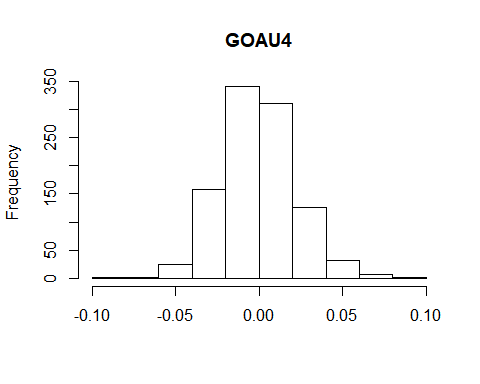
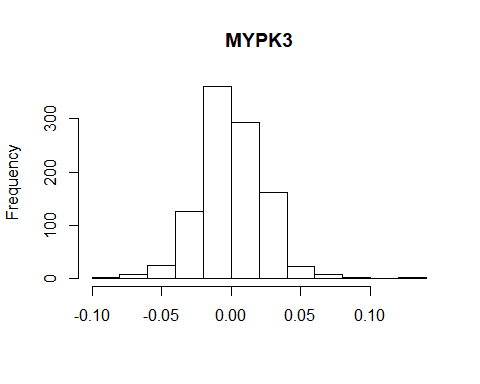
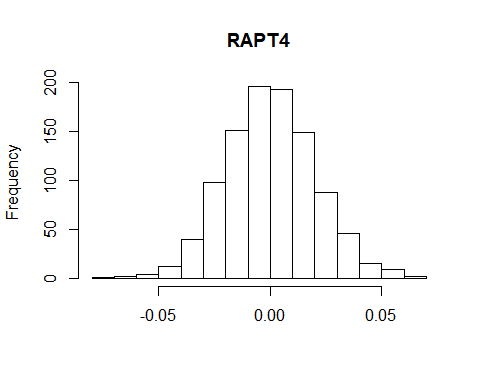
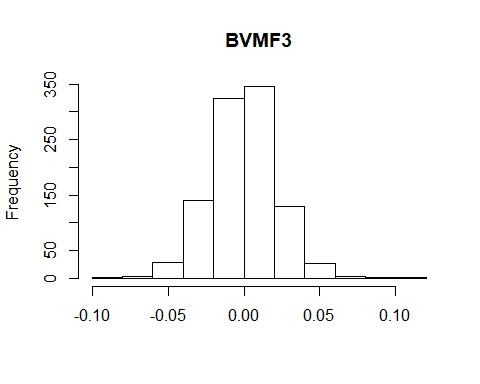
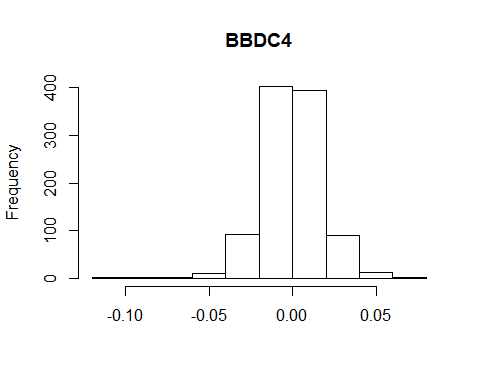
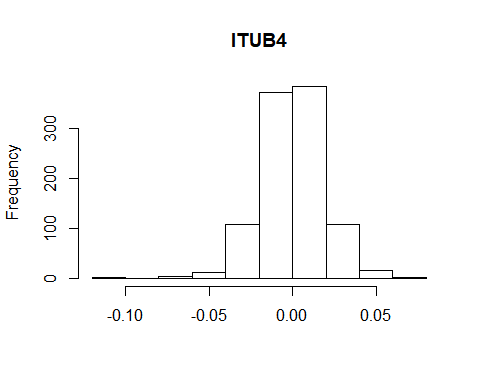
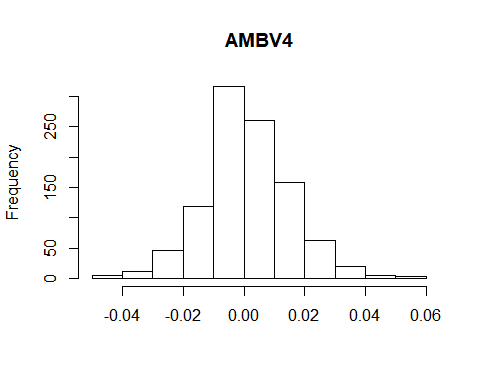
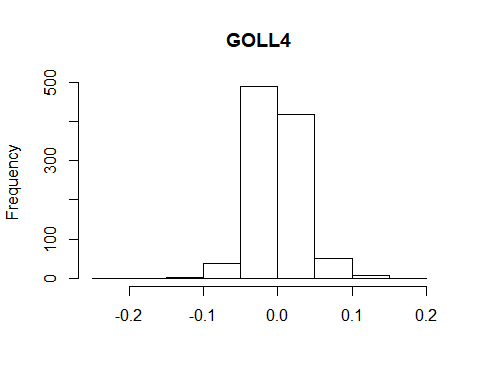
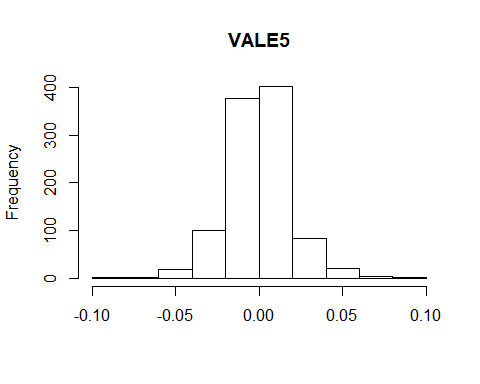
1. Crie a série temporal dos retornos Ln, ou seja, r=Ln(P\_t+1 /P\_t)

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,]

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:

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



Desvio Padrao e Media:

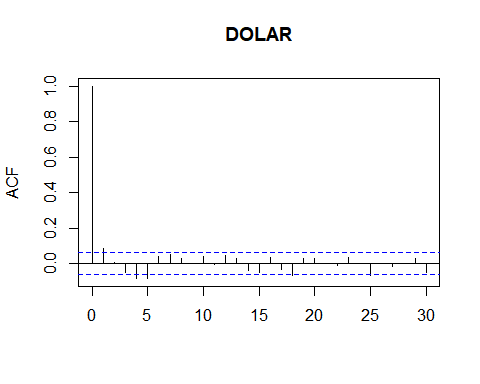
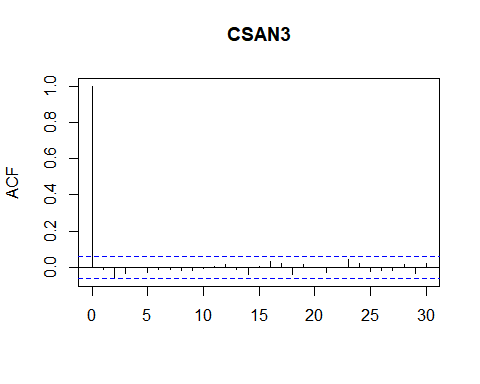
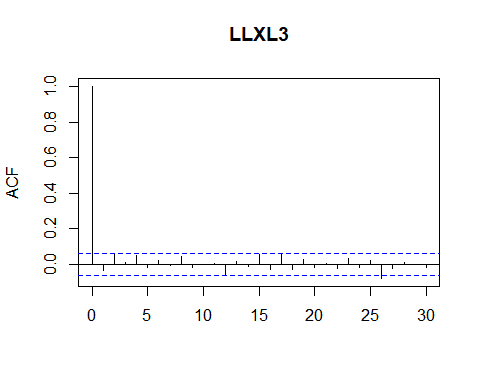
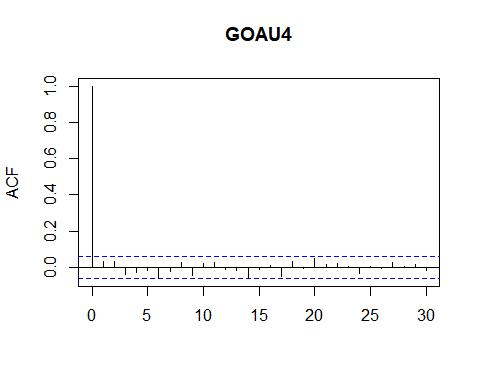
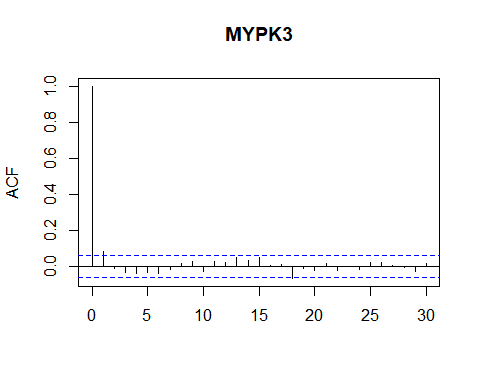
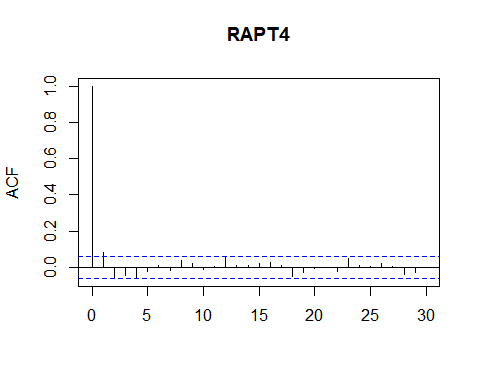
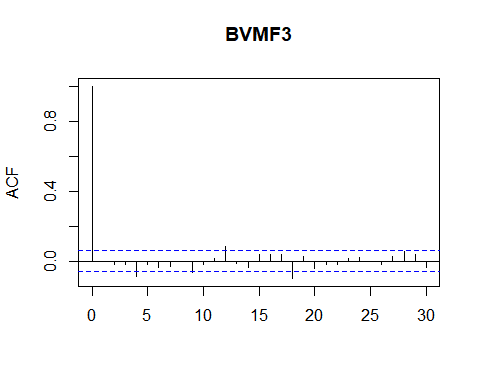
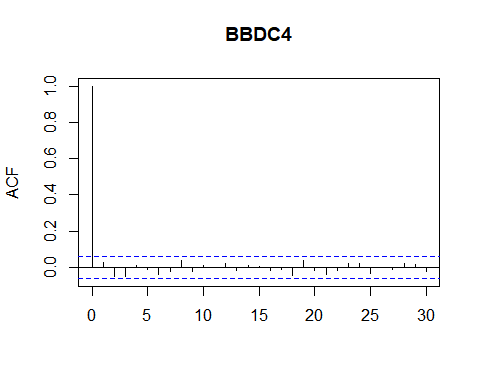
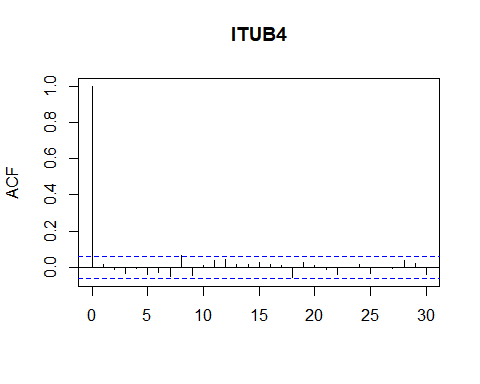
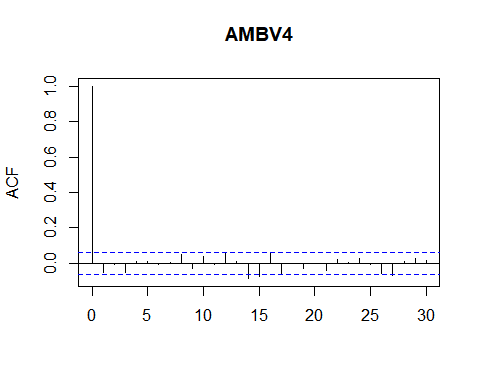
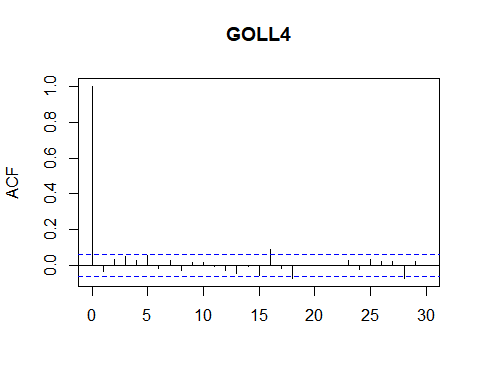
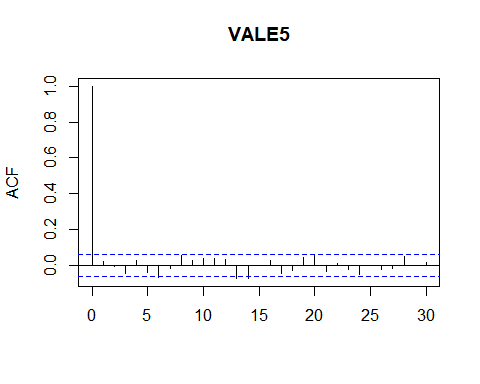
sapply(Serie\_Dados.LN, function(cl) list(Media=mean(cl,na.rm=TRUE), DesvioPadrao=sd(cl,na.rm=TRUE)))

## 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

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

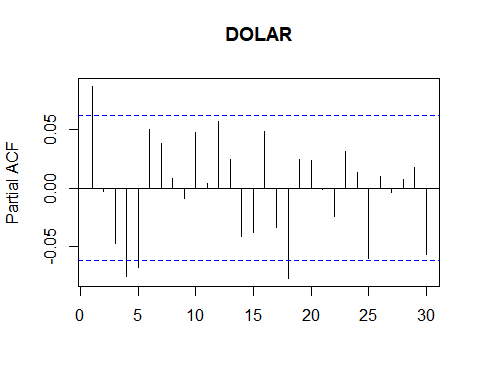
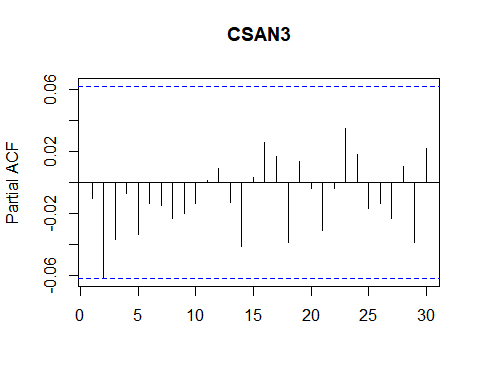
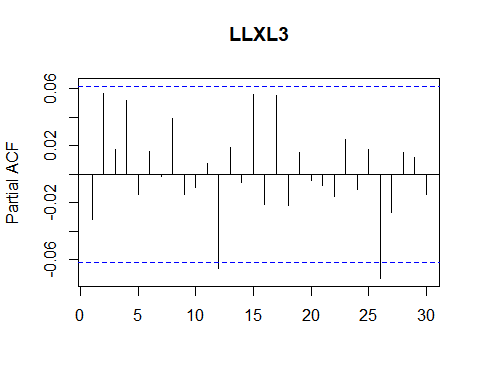
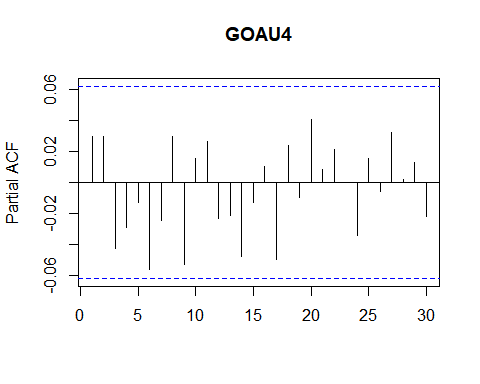
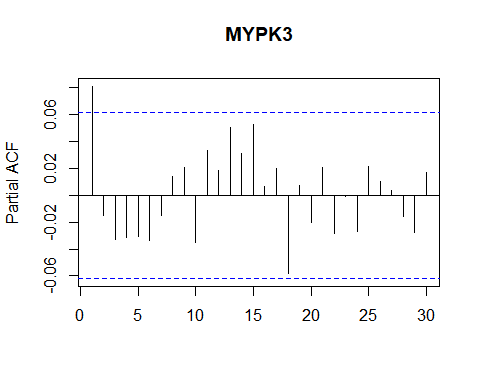
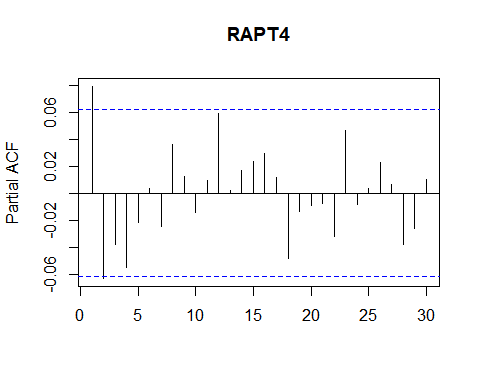
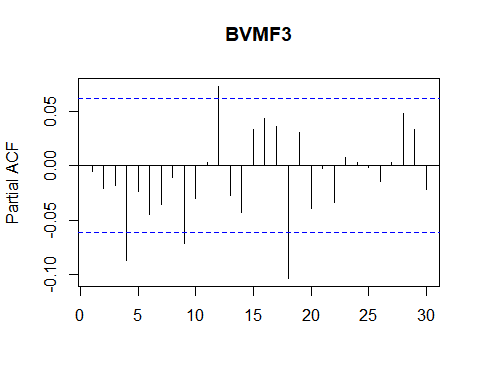
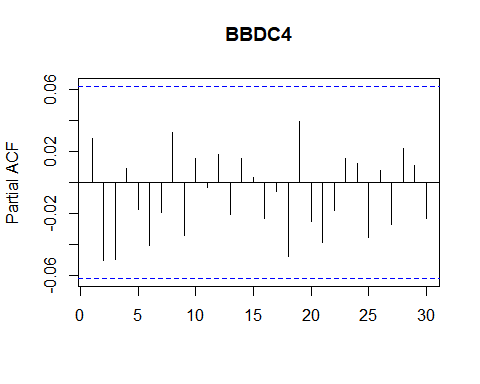
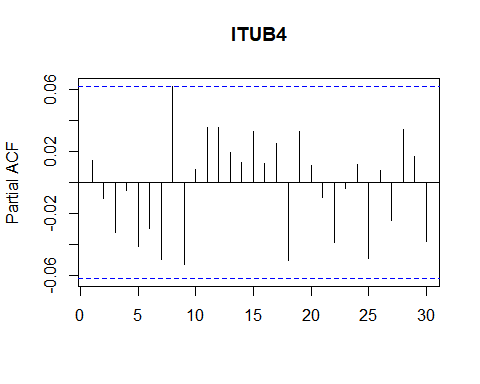
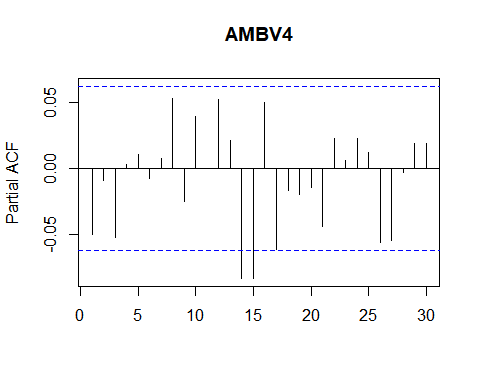
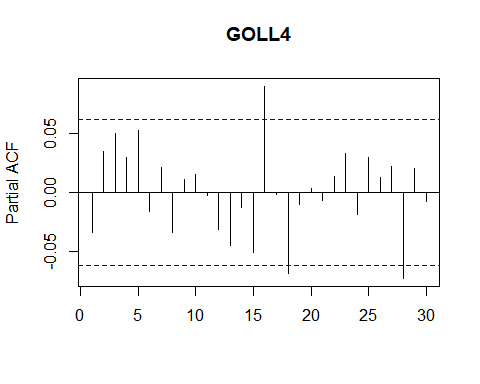
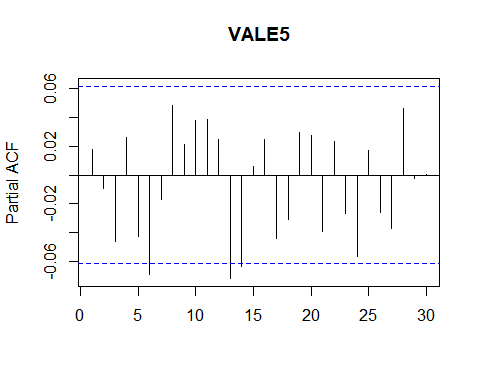
ACF

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



PACF

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



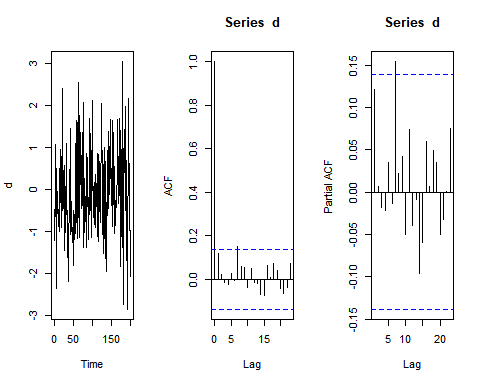
2- Para cada um dos processos abaixo gere 200 observações. Faça um gráfico da série, ACF e PACF. Comente os resultados.

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

set.seed(1234)

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

#x.iid = data.frame(t = x[2:200],  
# t\_1 = x[1:199])  
#x.iid  
#x.iid.mod = lm(t~t\_1,data = x.iid)  
#summary(x.iid.mod)  
#plot(x.iid.mod)  
#acf(x.iid)  
#pacf(x.iid)  
  
par(mfrow = c(1, 3))  
  
d <- ts(rnorm(200, 0, 1)) #ts: time series  
plot(d)  
acf(d)  
pacf(d)

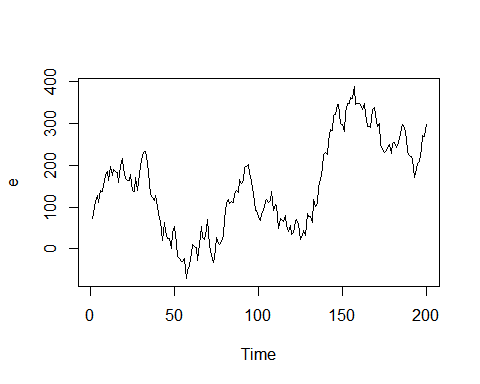


A serie é estacionária, mas por ser iid tem a PACF igual a 0

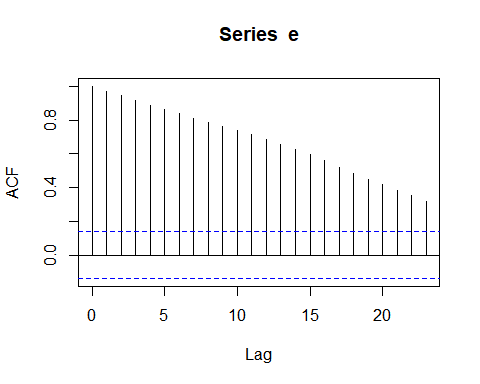
1. Série com tendência estocástica,

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

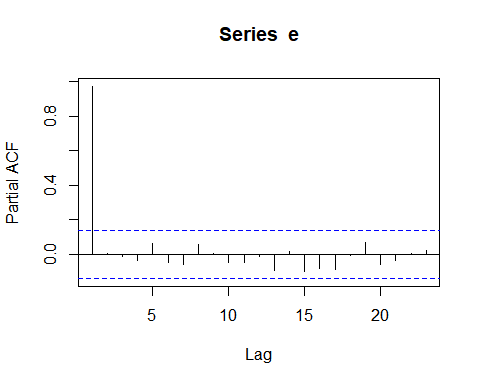
e <- arima.sim(model = list(ar= 0.99999), n=200, innov = rnorm(200,1,25))  
plot(e)



acf(e)



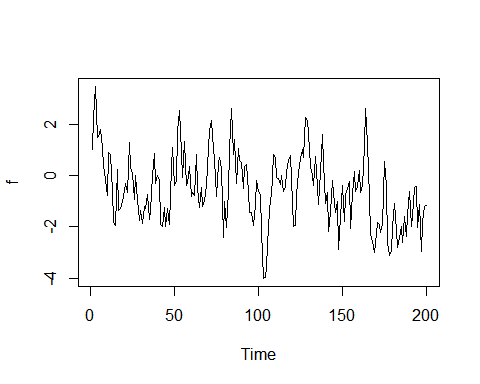
pacf(e)



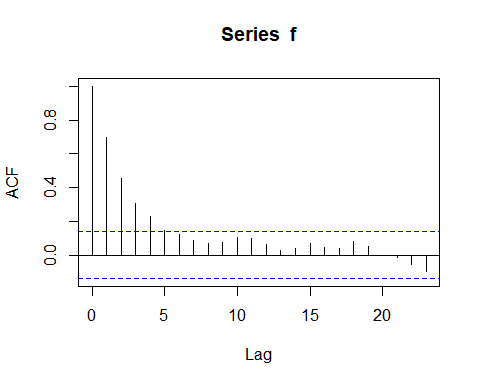
A série não é estacionaria

1. Serie com correlação de curto-prazo,

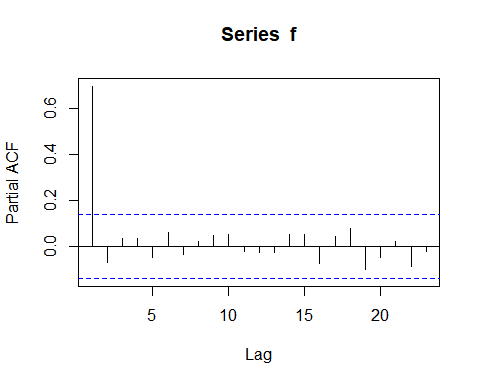
f <- arima.sim(model = list(ar = 0.7), n = 200, innov = rnorm(200, 0, 1))  
plot(f)



acf(f)



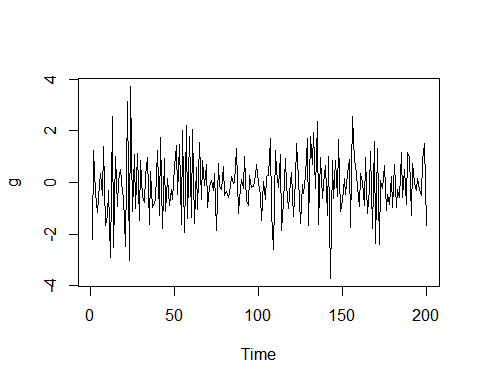
pacf(f)



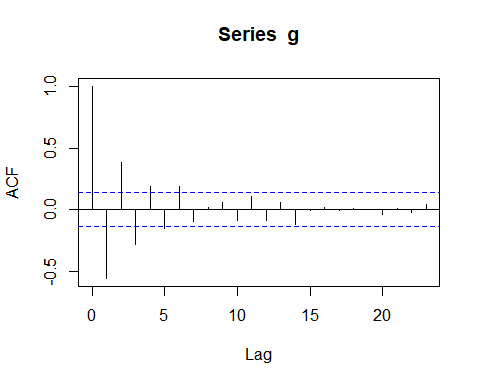
A série é estacionária, ACF com decaimento e grafico de pacf com pico em 1

1. Série com correlações negativas

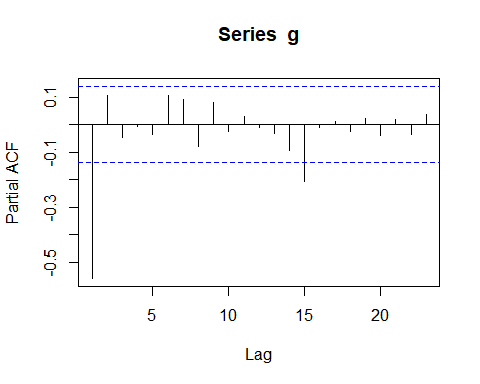
g <- arima.sim(model = list(ar = -0.7), n = 200, innov = rnorm(200, 0, 1))  
plot(g)



acf(g)



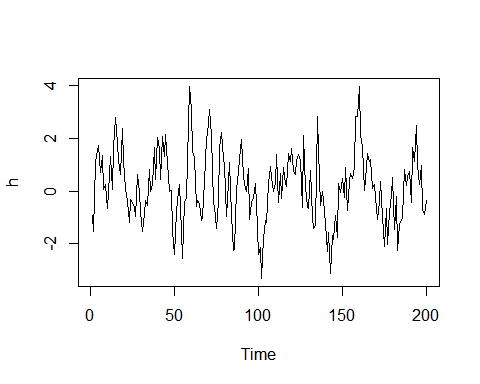
pacf(g)



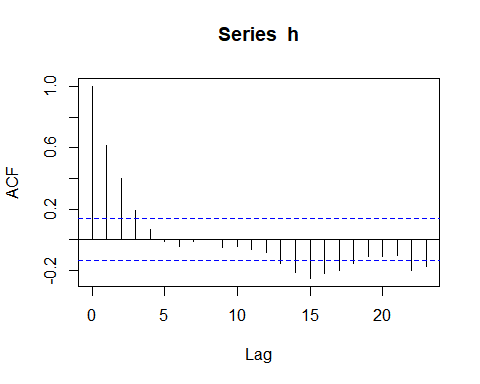
A série é estacionária, acf com decaimento oscilando, pacf pico em 1

1. Medias moveis,

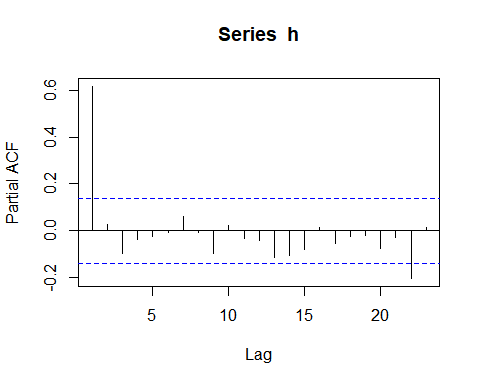
h <- arima.sim(model = list(ar = 0.6), n = 200, innov = rnorm(200, 0, 1))  
plot(h)



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

## Time Series:  
## Start = 1866   
## End = 1911   
## Frequency = 1   
## SKIRTS  
## [1,] 608  
## [2,] 617  
## [3,] 625  
## [4,] 636  
## [5,] 657  
## [6,] 691  
## [7,] 728  
## [8,] 784  
## [9,] 816  
## [10,] 876  
## [11,] 949  
## [12,] 997  
## [13,] 1027  
## [14,] 1047  
## [15,] 1049  
## [16,] 1018  
## [17,] 1021  
## [18,] 1012  
## [19,] 1018  
## [20,] 991  
## [21,] 962  
## [22,] 921  
## [23,] 871  
## [24,] 829  
## [25,] 822  
## [26,] 820  
## [27,] 802  
## [28,] 821  
## [29,] 819  
## [30,] 791  
## [31,] 746  
## [32,] 726  
## [33,] 661  
## [34,] 620  
## [35,] 588  
## [36,] 568  
## [37,] 542  
## [38,] 551  
## [39,] 541  
## [40,] 557  
## [41,] 556  
## [42,] 534  
## [43,] 528  
## [44,] 529  
## [45,] 523  
## [46,] 531

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

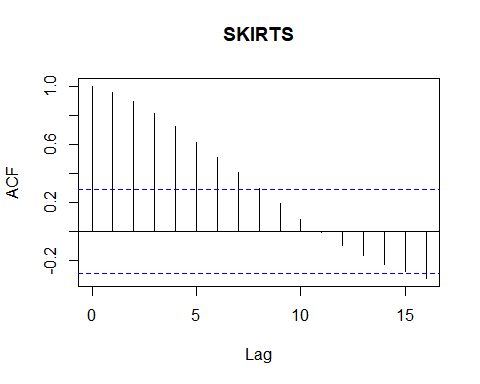
skirts.components <- ifelse(frequency(skirts.ts)>1,  
 decompose(skirts.ts,type = c("additive", "multiplicative")),  
 print("Nao e' possivel decompor uma serie anual, para ser feita a decomposicao a serie deveria ter, no minimo, 2 periodos"))

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

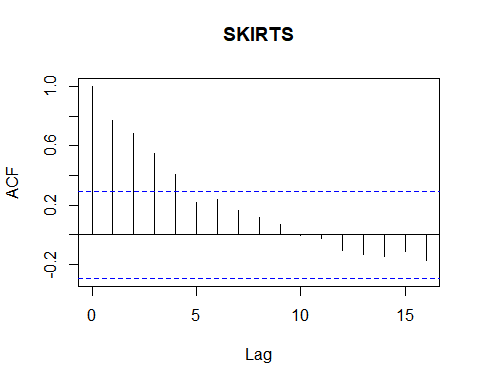
#plot(skirts.components)

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

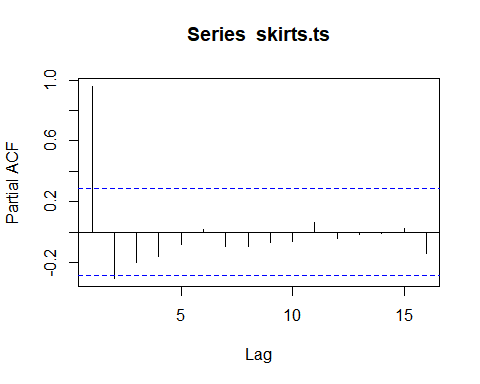
acf(skirts.ts)



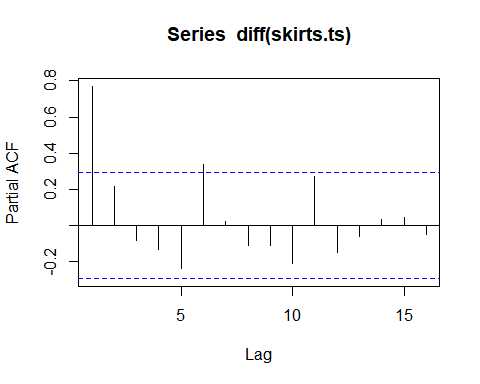
acf(diff(skirts.ts))



pacf(skirts.ts)



pacf(diff(skirts.ts))

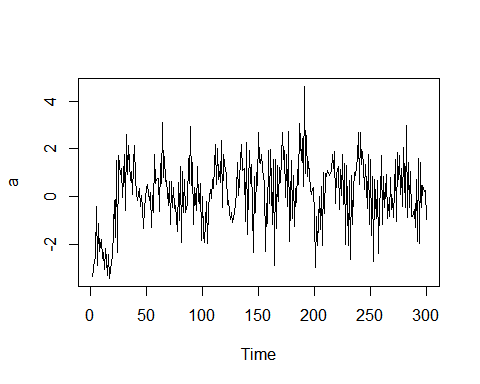


4- Usando a função arima.sim gere as seguintes simulações (300 ptos):

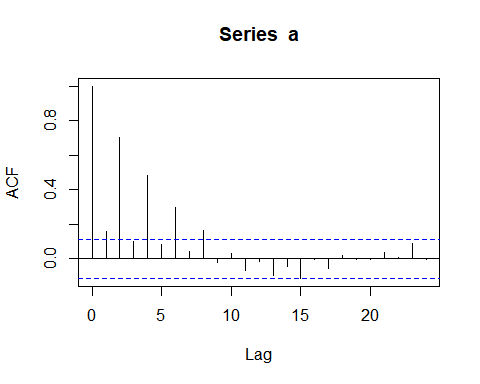
set.seed(1234)

1. Processo AR(1) onde θ0=0, θ1=0.7

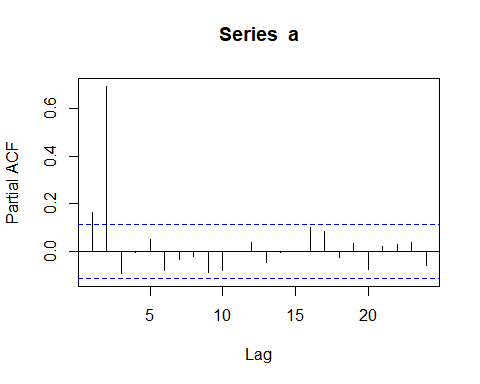
a <- arima.sim(n=300,list(ar = c(0,.7)))  
plot(a)



acf(a)

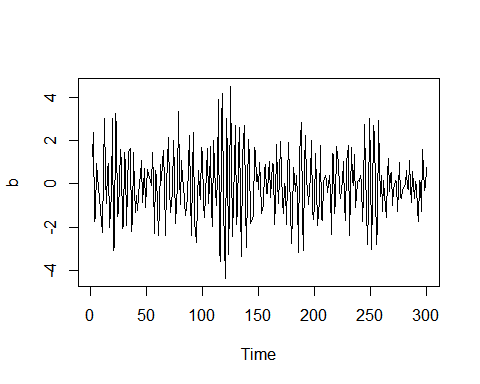


pacf(a)

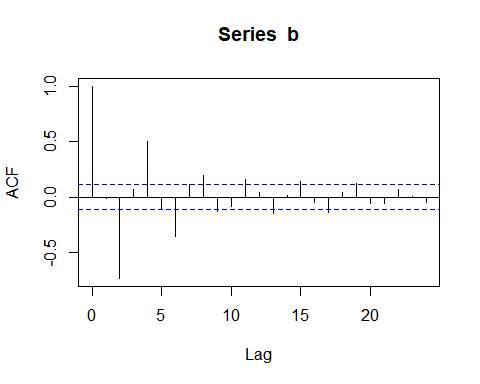


1. Processo AR(1) onde θ0=0, θ1= -0.7

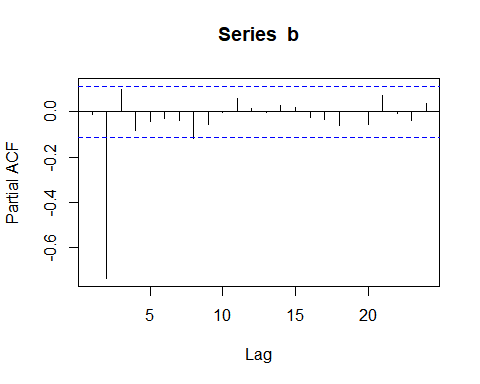
b <- arima.sim(n=300,list(ar = c(0,-.7)))  
plot(b)



acf(b)

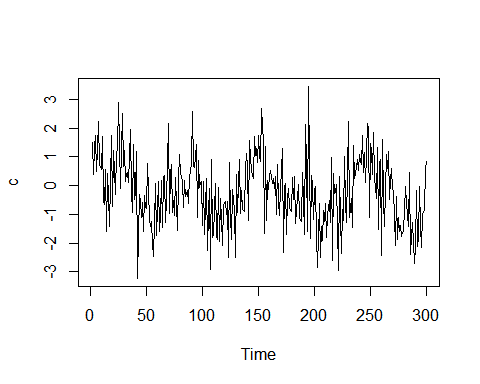


pacf(b)

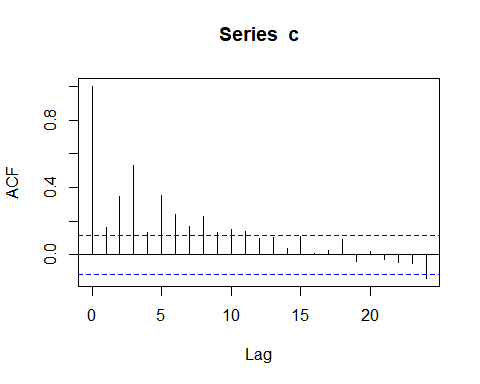


1. Processo AR(2) onde θ0=0, θ1=0.3 e θ2=0.5

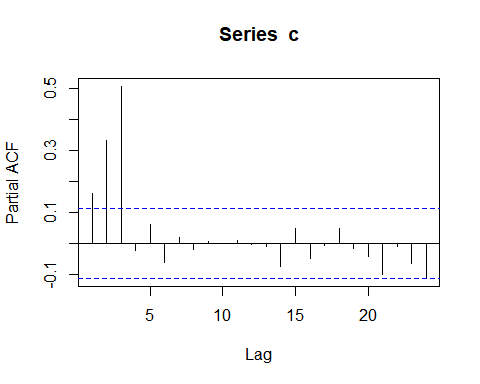
c <- arima.sim(n=300,list(ar = c(0,.3,.5)))  
plot(c)



acf(c)

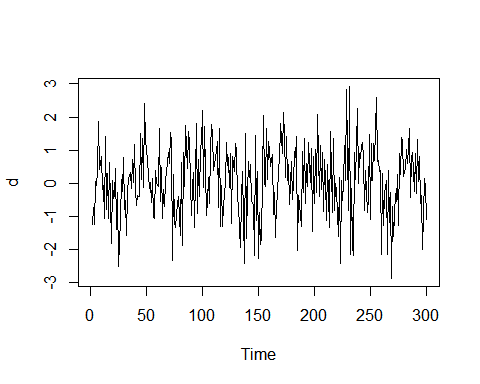


pacf(c)

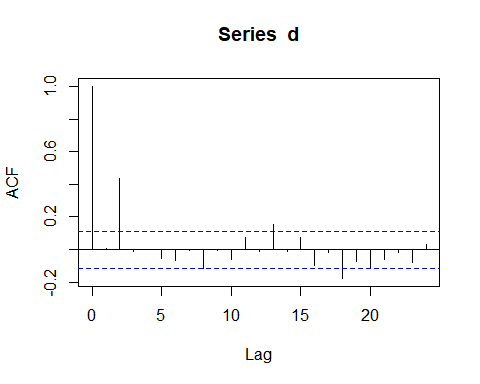


1. Processo MA(1) onde θ0=0, θ1=0.6

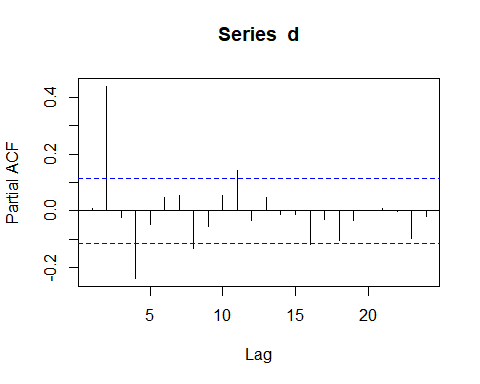
d <- arima.sim(n=300,list(ma = c(0,.6)))  
plot(d)



acf(d)

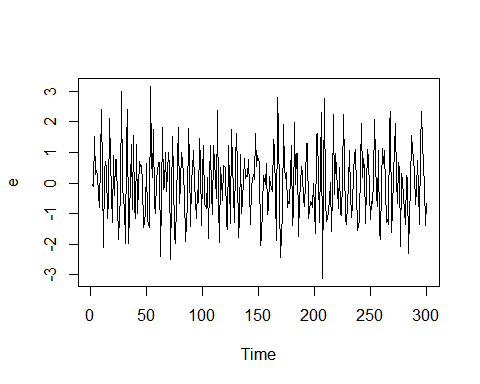


pacf(d)

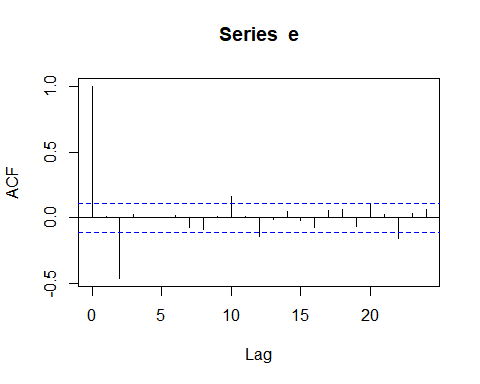


1. Processo MA(1) onde θ0=0, θ1=-0.6

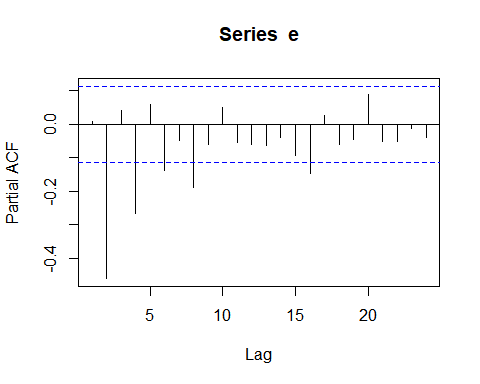
e <- arima.sim(n=300,list(ma = c(0,-.6)))  
plot(e)



acf(e)



pacf(e)



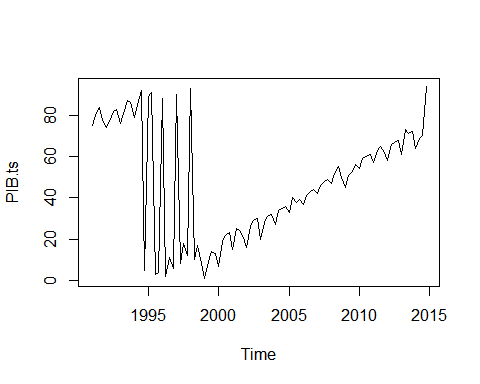
Para cada simulação, plote o gráfico da série, calcule o ACF e PACF. Usando estes resultados conclua como deve ser o comportamento da ACF de PACF de um modelo autoregressivo( AR.)

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

PIB <- read.csv("PIB.csv", sep=";")  
#View(PIB)  
PIB.ts<-ts(PIB$X1232.PIB, frequency=4, start=c(1991,1))

1. Plote o gráfico da série usando o R

plot.ts(PIB.ts)

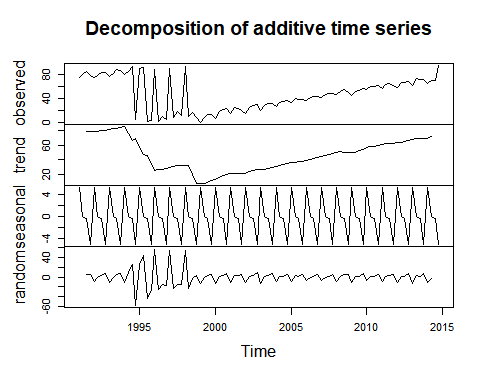


1. Faça a decomposição da série em: Sazonalidade, Tendência e Aleatória.

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

## $x  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1991 75 80 84 78  
## 1992 74 77 82 83  
## 1993 76 81 87 86  
## 1994 79 85 92 5  
## 1995 89 91 3 4  
## 1996 88 2 11 6  
## 1997 90 8 18 12  
## 1998 93 10 17 8  
## 1999 1 9 14 13  
## 2000 7 19 22 23  
## 2001 15 25 24 21  
## 2002 16 26 29 30  
## 2003 20 28 31 32  
## 2004 27 34 35 36  
## 2005 33 40 38 39  
## 2006 37 41 43 44  
## 2007 42 46 48 49  
## 2008 47 52 55 50  
## 2009 45 51 53 56  
## 2010 54 59 60 61  
## 2011 57 63 65 62  
## 2012 58 66 67 68  
## 2013 61 73 71 72  
## 2014 64 69 70 94  
##   
## $seasonal  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1991 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1992 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1993 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1994 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1995 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1996 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1997 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1998 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 1999 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2000 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2001 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2002 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2003 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2004 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2005 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2006 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2007 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2008 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2009 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2010 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2011 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2012 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2013 5.2853261 0.1440217 -0.3614130 -5.0679348  
## 2014 5.2853261 0.1440217 -0.3614130 -5.0679348  
##   
## $trend  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1991 NA NA 79.125 78.625  
## 1992 78.000 78.375 79.250 80.000  
## 1993 81.125 82.125 82.875 83.750  
## 1994 84.875 75.375 66.500 68.500  
## 1995 58.125 46.875 46.625 35.375  
## 1996 25.250 26.500 27.000 28.000  
## 1997 29.625 31.250 32.375 33.000  
## 1998 33.125 32.500 20.500 8.875  
## 1999 8.375 8.625 10.000 12.000  
## 2000 14.250 16.500 18.750 20.500  
## 2001 21.500 21.500 21.375 21.625  
## 2002 22.375 24.125 25.750 26.500  
## 2003 27.000 27.500 28.625 30.250  
## 2004 31.500 32.500 33.750 35.250  
## 2005 36.375 37.125 38.000 38.625  
## 2006 39.375 40.625 41.875 43.125  
## 2007 44.375 45.625 46.875 48.250  
## 2008 49.875 50.875 50.750 50.375  
## 2009 50.000 50.500 52.375 54.500  
## 2010 56.375 57.875 58.875 59.750  
## 2011 60.875 61.625 61.875 62.375  
## 2012 63.000 64.000 65.125 66.375  
## 2013 67.750 68.750 69.625 69.500  
## 2014 68.875 71.500 NA NA  
##   
## $random  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1991 NA NA 5.2364130 4.4429348  
## 1992 -9.2853261 -1.5190217 3.1114130 8.0679348  
## 1993 -10.4103261 -1.2690217 4.4864130 7.3179348  
## 1994 -11.1603261 9.4809783 25.8614130 -58.4320652  
## 1995 25.5896739 43.9809783 -43.2635870 -26.3070652  
## 1996 57.4646739 -24.6440217 -15.6385870 -16.9320652  
## 1997 55.0896739 -23.3940217 -14.0135870 -15.9320652  
## 1998 54.5896739 -22.6440217 -3.1385870 4.1929348  
## 1999 -12.6603261 0.2309783 4.3614130 6.0679348  
## 2000 -12.5353261 2.3559783 3.6114130 7.5679348  
## 2001 -11.7853261 3.3559783 2.9864130 4.4429348  
## 2002 -11.6603261 1.7309783 3.6114130 8.5679348  
## 2003 -12.2853261 0.3559783 2.7364130 6.8179348  
## 2004 -9.7853261 1.3559783 1.6114130 5.8179348  
## 2005 -8.6603261 2.7309783 0.3614130 5.4429348  
## 2006 -7.6603261 0.2309783 1.4864130 5.9429348  
## 2007 -7.6603261 0.2309783 1.4864130 5.8179348  
## 2008 -8.1603261 0.9809783 4.6114130 4.6929348  
## 2009 -10.2853261 0.3559783 0.9864130 6.5679348  
## 2010 -7.6603261 0.9809783 1.4864130 6.3179348  
## 2011 -9.1603261 1.2309783 3.4864130 4.6929348  
## 2012 -10.2853261 1.8559783 2.2364130 6.6929348  
## 2013 -12.0353261 4.1059783 1.7364130 7.5679348  
## 2014 -10.1603261 -2.6440217 NA NA  
##   
## $figure  
## [1] 5.2853261 0.1440217 -0.3614130 -5.0679348  
##   
## $type  
## [1] "additive"  
##   
## attr(,"class")  
## [1] "decomposed.ts"

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[49:96])  
PIB.predict <- predict(auto.arima(PIB.dif),ahead = 1)  
PIB.predict

## $pred  
## Time Series:  
## Start = 48   
## End = 48   
## Frequency = 1   
## [1] -6.898466  
##   
## $se  
## Time Series:  
## Start = 48   
## End = 48   
## Frequency = 1   
## [1] 5.194559

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 = 1.5736   
##   
## sigma: 5.4275  
##   
## AIC AICc BIC   
## 343.9129 344.4711 349.4634

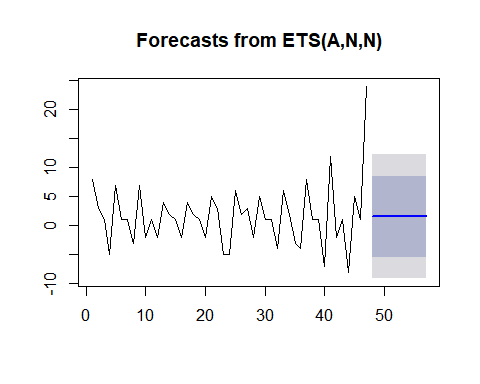
accuracy(etsfit.PIB.dif)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.001407463 5.310812 3.681257 59.63687 88.9061 0.6271772  
## ACF1  
## Training set -0.2199867

fcast.PIB.dif <- forecast(etsfit.PIB.dif)  
fcast.PIB.dif

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 48 1.57364 -5.382041 8.529321 -9.064157 12.21144  
## 49 1.57364 -5.382041 8.529321 -9.064157 12.21144  
## 50 1.57364 -5.382041 8.529321 -9.064157 12.21144  
## 51 1.57364 -5.382042 8.529321 -9.064157 12.21144  
## 52 1.57364 -5.382042 8.529322 -9.064157 12.21144  
## 53 1.57364 -5.382042 8.529322 -9.064157 12.21144  
## 54 1.57364 -5.382042 8.529322 -9.064157 12.21144  
## 55 1.57364 -5.382042 8.529322 -9.064157 12.21144  
## 56 1.57364 -5.382042 8.529322 -9.064157 12.21144  
## 57 1.57364 -5.382042 8.529322 -9.064157 12.21144

plot(fcast.PIB.dif)

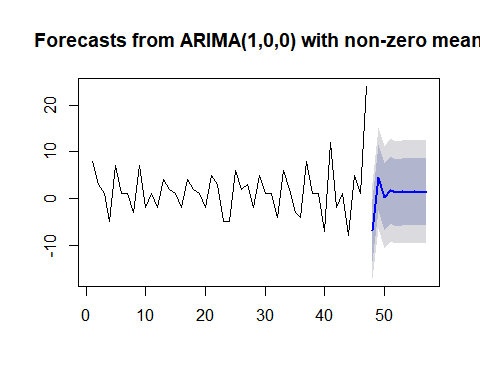


Usando ARIMA:

arimafit.PIB.dif <- auto.arima(PIB.dif)  
fcast.ARIMA.PIB.dif <- forecast(arimafit.PIB.dif)  
fcast.ARIMA.PIB.dif

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 48 -6.8984662 -13.555561 -0.2413712 -17.079614 3.282682  
## 49 4.4612251 -2.631513 11.5539636 -6.386182 15.308633  
## 50 0.2848823 -6.864703 7.4344677 -10.649465 11.219230  
## 51 1.8202972 -5.336937 8.9775316 -9.125748 12.766343  
## 52 1.2558084 -5.902459 8.4140760 -9.691817 12.203434  
## 53 1.4633403 -5.695067 8.6217476 -9.484499 12.411179  
## 54 1.3870421 -5.771384 8.5454682 -9.560826 12.334910  
## 55 1.4150928 -5.743336 8.5735215 -9.532779 12.362965  
## 56 1.4047801 -5.753649 8.5632091 -9.543092 12.352653  
## 57 1.4085715 -5.749858 8.5670006 -9.539301 12.356444

plot(fcast.ARIMA.PIB.dif)



Usando Forecast (sem usar a diferença):

etsfit.PIB.ts <- ets(PIB.ts[49:96])  
etsfit.PIB.ts

## ETS(M,A,N)   
##   
## Call:  
## ets(y = PIB.ts[49:96])   
##   
## Smoothing parameters:  
## alpha = 0.0296   
## beta = 0.0296   
##   
## Initial states:  
## l = 25.1972   
## b = 1.3539   
##   
## sigma: 0.0825  
##   
## AIC AICc BIC   
## 326.5554 327.9840 335.9114

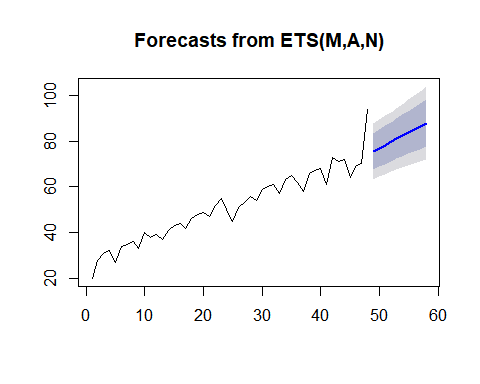
accuracy(etsfit.PIB.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01216636 4.313633 2.950698 -1.043802 5.960771 0.7456065  
## ACF1  
## Training set -0.04256709

fcast.PIB.ts <- forecast(etsfit.PIB.ts)  
fcast.PIB.ts

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 49 75.54042 67.55617 83.52466 63.32957 87.75127  
## 50 76.91158 68.76863 85.05453 64.45801 89.36515  
## 51 78.28274 69.96423 86.60126 65.56067 91.00482  
## 52 79.65391 71.13682 88.17100 66.62814 92.67967  
## 53 81.02507 72.28071 89.76944 67.65172 94.39842  
## 54 82.39623 73.39078 91.40169 68.62358 96.16889  
## 55 83.76740 74.46259 93.07221 69.53692 97.99787  
## 56 85.13856 75.49243 94.78469 70.38608 99.89105  
## 57 86.50972 76.47736 96.54209 71.16654 101.85291  
## 58 87.88089 77.41520 98.34658 71.87500 103.88678

plot(fcast.PIB.ts)



Usando ARIMA (sem usar a diferença):

arimafit.PIB.ts <- auto.arima(PIB.ts[49:96])  
fcast.ARIMA.PIB.ts <- forecast(arimafit.PIB.ts)  
fcast.ARIMA.PIB.ts

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 49 87.10154 80.44444 93.75863 76.92039 97.28269  
## 50 91.56276 83.68634 99.43918 79.51681 103.60871  
## 51 91.84764 82.45912 101.23617 77.48913 106.20615  
## 52 93.66794 83.13325 104.20263 77.55652 109.77936  
## 53 94.92375 83.30514 106.54236 77.15461 112.69289  
## 54 96.38709 83.79469 108.97949 77.12868 115.64550  
## 55 97.77413 84.27211 111.27616 77.12457 118.42370  
## 56 99.18923 84.83715 113.54130 77.23963 121.13883  
## 57 100.59401 85.43878 115.74923 77.41609 123.77193  
## 58 102.00258 86.08492 117.92024 77.65862 126.34654

plot(fcast.ARIMA.PIB.ts)

