# Series de tiempo

**Mario A. Guerra U.**
**Cc:1152451112**

Consider a contrived set of data generated by tossing a fair coin, letting $x_t = 1$ when a head obtained and $x_t = 0$ when a tail obtained. Construt $y_t$ as

$$y_t = 5 + x_t - 0,65x_(t-1).$$

a. Compare the sample ACF you obtain to the actual ACF, with n = 10, 100, 200, 500 and 1000.

b. For each n = 10, 100, 200, 500 and 1000, simulate 1000 replications, and compute the sample ACF to lag 10, verify the Large Sample Distribution of the ACF.

```
library(tseries)
library(portes)
library(forecast)
library(fma)
library(expsmooth)
library(zoo)
library(lmtest)
library(fpp)

# Definicion de los parametros del proceso
theta0_ma1 <- 5
theta1_ma1 <- 0.65
var_RB_ma1 <- 2
# chequeo de la invertibilidad del modelo\
# coefma1c(theta1_ma1)
# armaRoots(coefma1)

# simulacion


a <- rep(0,1000)

for(i in 1:1000){
  z_ma1 <- arima.sim(n = 11, list(ma = theta1_ma1),
                   mean=theta0_ma1, sd = sqrt(var_RB_ma1), n.start=1000)
a[i] <- acf(z_ma1,plot=FALSE, lag.max = 11)[10,1]

}
```

```r
anue1 <- mean(as.numeric(a))
anue1

## [1] -0.01404775
```

El promedio de todos los resagos numeros 10 para n=10

```r
a <- rep(0,1000)

for(i in 1:1000){
  z_ma2 <- arima.sim(n = 101, list(ma = theta1_ma1),
                     mean=theta0_ma1, sd = sqrt(var_RB_ma1), n.start=1000)
a[i] <- acf(z_ma2,plot=FALSE, lag.max = 101)[10,1]

}

anue2 <- mean(as.numeric(a))
anue2

## [1] -0.01929081
```

El promedio de todos los resagos numeros 10 para n=100

```r
a <- rep(0,1000)

for(i in 1:1000){
  z_ma3 <- arima.sim(n = 201, list(ma = theta1_ma1),
                     mean=theta0_ma1, sd = sqrt(var_RB_ma1), n.start=1000)
a[i] <- acf(z_ma3,plot=FALSE, lag.max = 201)[10,1]

}

anue3 <- mean(as.numeric(a))
anue3

## [1] -0.009472503
```

El promedio de todos los resagos numeros 10 para n=200

```r
a <- rep(0,1000)

for(i in 1:1000){
  z_ma4 <- arima.sim(n = 501, list(ma = theta1_ma1),
                     mean=theta0_ma1, sd = sqrt(var_RB_ma1), n.start=1000)
a[i] <- acf(z_ma4,plot=FALSE, lag.max = 501)[10,1]
```

```
}

anue4 <- mean(as.numeric(a))
anue4

## [1] -0.002764429
```

El promedio de todos los resagos numeros 10 para n=500

```
a <- rep(0,1000)


for(i in 1:1000){
  z_ma5 <- arima.sim(n = 1001, list(ma = theta1_ma1),
                     mean=theta0_ma1, sd = sqrt(var_RB_ma1), n.start=1000)

a[i] <- acf(z_ma4,plot=FALSE, lag.max = 501)[10,1]

}

anue5 <- mean(as.numeric(a))
anue5

## [1] -0.04605323
```

El promedio de todos los resagos numeros 10 para n=1000.

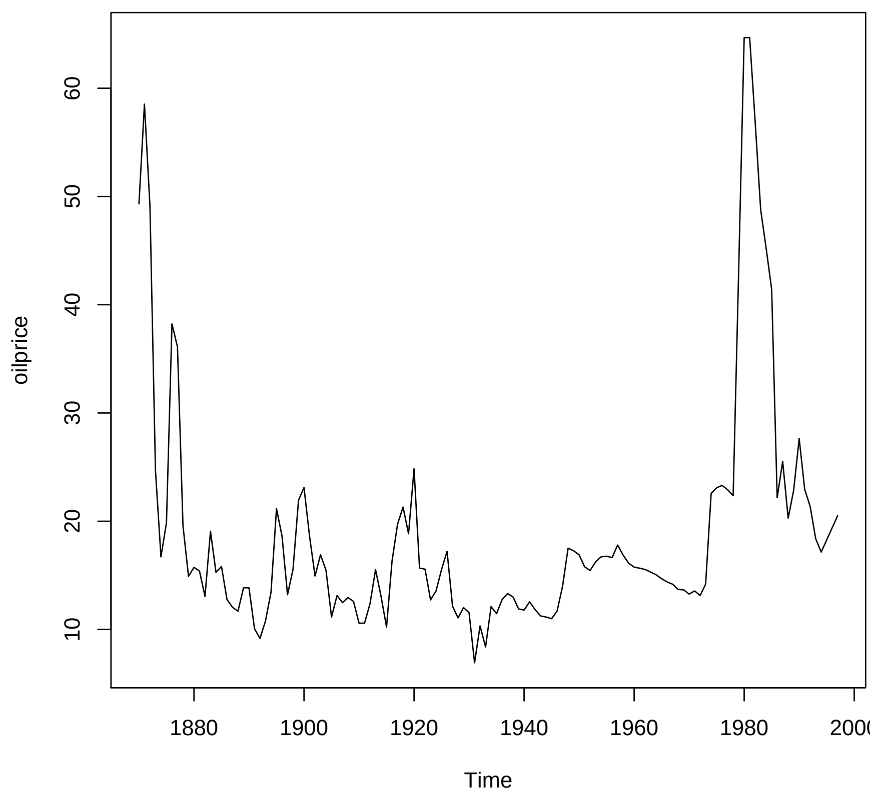
punto 6:

Crude oil prices in dollars per barrel are in oil; see Appendix R for more details. Fit an $ARIMA(p, d, q)$ model to the growth rate performing all necessary diagnostics. Comment.

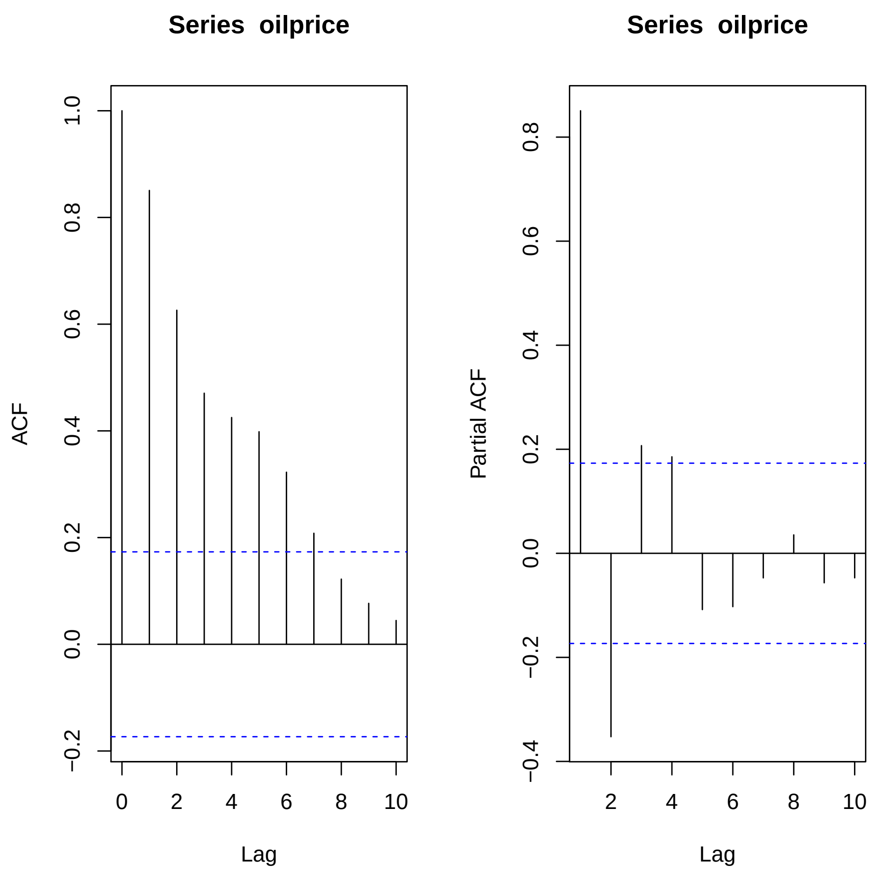
```
oilprice

## Time Series:
## Start = 1870
## End = 1997
## Frequency = 1
##   [1] 49.32 58.53 49.09 24.68 16.71 19.86 38.23 36.11 19.59 14.91 15.74
##  [12] 15.40 13.06 19.07 15.28 15.82 12.77 12.05 11.69 13.85 13.85 10.07
##  [23]  9.17 10.79 13.44 21.17 18.64 13.21 15.54 21.94 23.11 18.64 14.94
##  [34] 16.90 15.46 11.15 13.13 12.48 12.95 12.59 10.58 10.58 12.39 15.53
##  [45] 13.06 10.22 16.33 19.72 21.31 18.84 24.84 15.67 15.57 12.73 13.56
##  [56] 15.54 17.22 12.14 11.07 12.02 11.55  6.92 10.33  8.38 12.11 11.46
##  [67] 12.75 13.32 13.00 11.90 11.79 12.55 11.84 11.25 11.15 10.99 11.70
##  [78] 14.01 17.51 17.27 16.90 15.79 15.45 16.24 16.71 16.77 16.64 17.80
```

```
##  [89] 16.87 16.13 15.76 15.66 15.54 15.30 15.05 14.69 14.39 14.18 13.70
## [100] 13.66 13.27 13.56 13.14 14.19 22.57 23.09 23.31 22.91 22.36 43.06
## [111] 64.67 64.68 56.90 48.79 45.21 41.40 22.17 25.52 20.28 22.87 27.63
## [122] 22.95 21.35 18.38 17.15 18.27 19.40 20.52
```

```
plot.ts(oilprice)
```

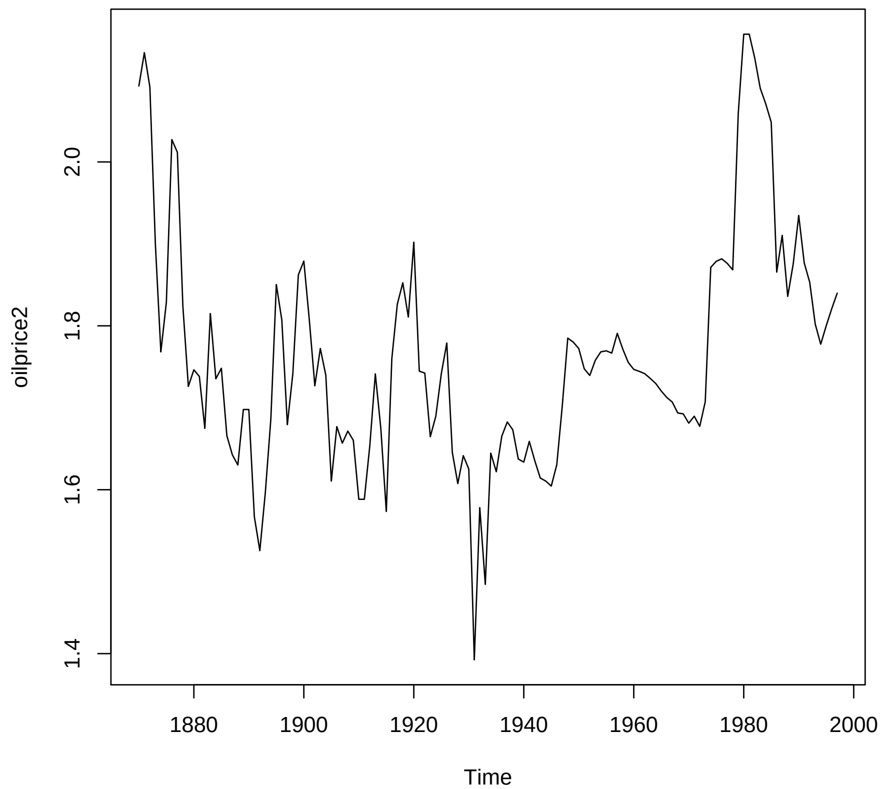



```
par(mfrow = c(1,2))
acf(oilprice,lag.max = 10)
pacf(oilprice,lag.max = 10)
```

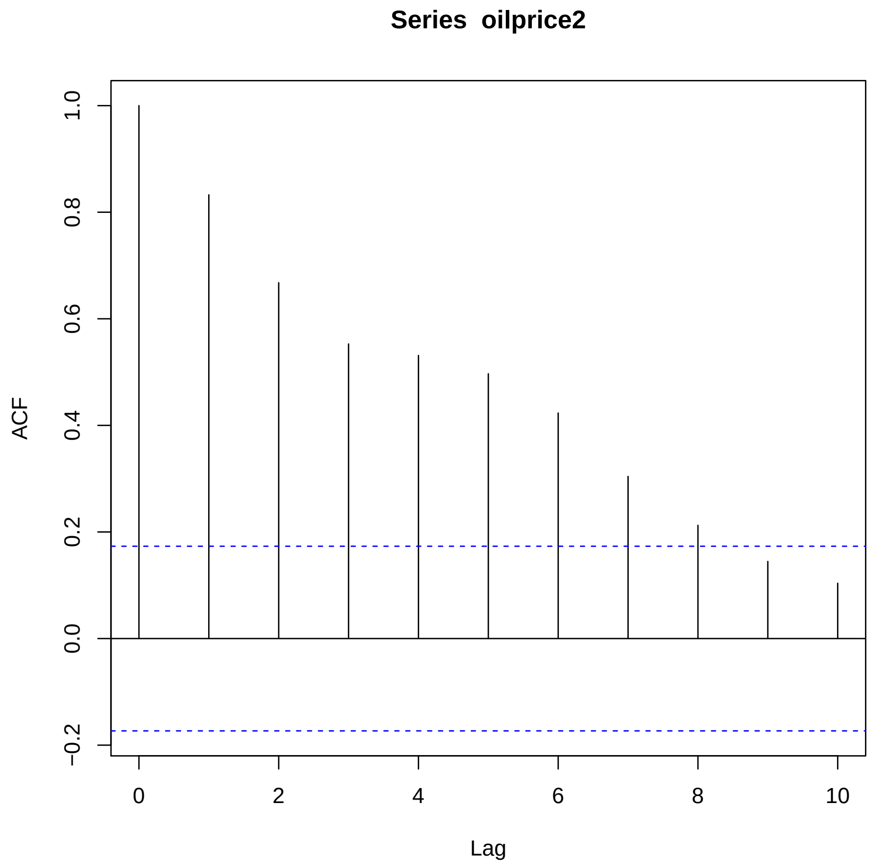



Se observa que la acf decae de manera muy lenta lo que posiblemente sugiere una diferenciacion , ademas que cuando graficamos la serie de tiempo se ve que no es estacionaria en varianza por lo que posiblemente utilizaremos una transformacion para estabilizarla .

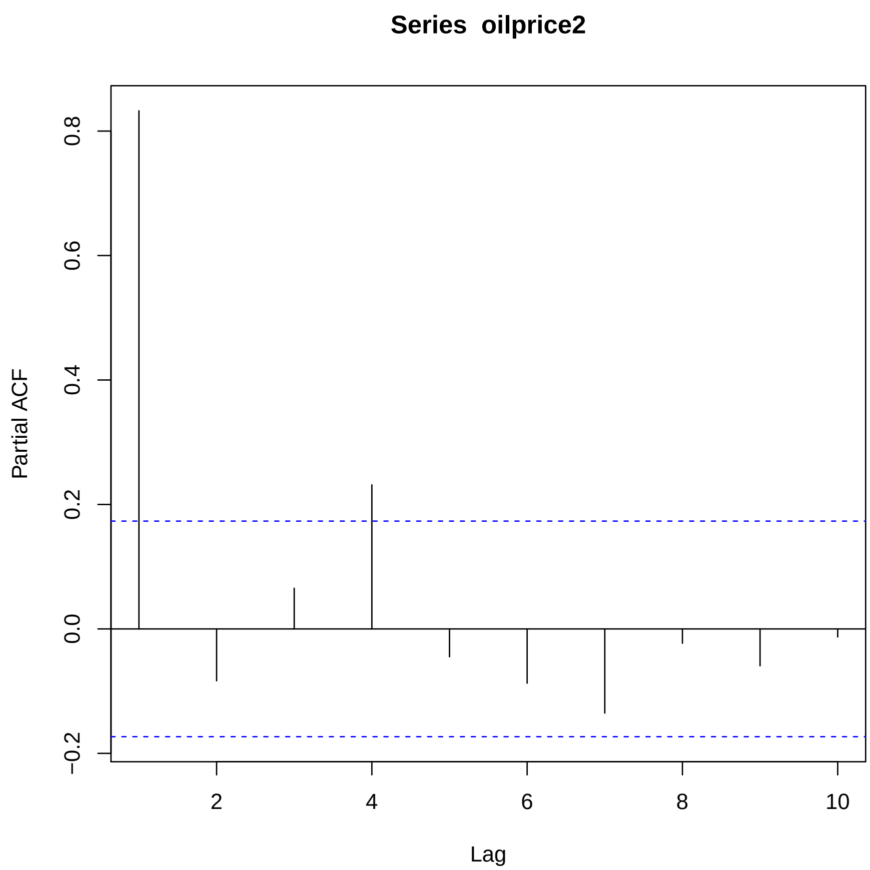
```
lambda <- BoxCox.lambda(oilprice)
lambda

## [1] -0.3607872

oilprice2 <- (oilprice^(lambda) -1)/lambda
plot.ts(oilprice2)
```




```r
acf(oilprice2,lag.max = 10)
```




```
pacf(oilprice2,lag.max = 10)
```




Utilizamos el metodo de Box-Cox para encontrar un lambda adecuado y utilizamos la transformacion para estabilizarla , vemos que la acf decae de manera muy lenta.Ahora miremos los posibles modelos que sepresentarian esta serie:

```
lambda <- BoxCox.lambda(oilprice)
oilprice2 <- (oilprice^(lambda) -1)/lambda
auto.arima(oilprice2)

## Series: oilprice2
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.9846  -0.9787  -1.0402  0.9401
## s.e.  0.0305   0.0285   0.0463  0.0781
##
## sigma^2 estimated as 0.005334:  log likelihood=153.12
```

```
## AIC=-296.24   AICc=-295.75   BIC=-282.02

modelo_autoarima <- arima(oilprice2,c(2,1,2))
modelo_autoarima

##
## Call:
## arima(x = oilprice2, order = c(2, 1, 2))
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.9846  -0.9787  -1.0402  0.9401
## s.e.  0.0305   0.0285   0.0463  0.0781
##
## sigma^2 estimated as 0.005166:  log likelihood = 153.12,  aic = -296.24

modelo_1 <- arima(oilprice2,c(2,1,1))
modelo_1

##
## Call:
## arima(x = oilprice2, order = c(2, 1, 1))
##
## Coefficients:
##          ar1      ar2      ma1
##       0.4457  -0.1838  -0.4983
## s.e.  0.2319   0.0928   0.2252
##
## sigma^2 estimated as 0.005491:  log likelihood = 150.23,  aic = -292.47

modelo_2 <- arima(oilprice2,c(1,1,1))
modelo_2

##
## Call:
## arima(x = oilprice2, order = c(1, 1, 1))
##
## Coefficients:
##          ar1      ma1
##       0.8319  -0.9496
## s.e.  0.1309   0.0938
##
## sigma^2 estimated as 0.005552:  log likelihood = 149.4,  aic = -292.81

modelo_3 <- arima(oilprice2,c(1,1,0))
modelo_3
```

```
##
## Call:
## arima(x = oilprice2, order = c(1, 1, 0))
##
## Coefficients:
##           ar1
##       -0.0245
## s.e.   0.0885
##
## sigma^2 estimated as 0.005779:  log likelihood = 147.05,  aic = -290.09
```

```
modelo_4 <- arima(oilprice2,c(0,1,0))
modelo_4
```

```
##
## Call:
## arima(x = oilprice2, order = c(0, 1, 0))
##
##
## sigma^2 estimated as 0.005782:  log likelihood = 147.01,  aic = -292.02
```

Se observa que a medida que comenzamos a reducir los grados del $arima(p, d, q)$ que aroja el autoarima , los $AIC$ son junto con las $\sigma^2$ estimados pero la variacion es minima y el $\sigma^2$ estimado varia solo en los decimales, ademas se ve que las desviaciones estandar de los parametros aumentan en comparacion a los que arroja el autoarima, por lo que aun no tenemos evidencia para quedarnos con un modelo,asique procederemos a realizar el diagnostico de residuales y mirar si la estimacion de los parametros es significativa.

```
LjungBox(modelo_autoarima)
```

```
##  lags statistic df    p-value
##     5  3.903608  1 0.04818254
##    10  7.244395  6 0.29883596
##    15 25.827004 11 0.00688564
##    20 28.663054 16 0.02630965
##    25 32.227554 21 0.05551554
##    30 34.612975 26 0.12027546
```

```
# preguntar por que aparece NA
#H_0: modelo adecuado= independencia de la serie de tiempo, si p-value<alpha,
#rechaza H_0=correlacion

LjungBox(modelo_autoarima)
```

```
##  lags statistic df    p-value
```

```
##     5  3.903608  1 0.04818254
##    10  7.244395  6 0.29883596
##    15 25.827004 11 0.00688564
##    20 28.663054 16 0.02630965
##    25 32.227554 21 0.05551554
##    30 34.612975 26 0.12027546
```

```r
#vemos que hay un resago que es significativamente estadistico.
LjungBox(modelo_1)
```

```
##  lags statistic df    p-value
##     5  5.400206  2 0.06719861
##    10  8.481316  7 0.29206869
##    15 23.191707 12 0.02614181
##    20 28.646371 17 0.03792940
##    25 33.769230 22 0.05184192
##    30 34.508017 27 0.15177147
```

```r
#eEste modelo es adecuado
LjungBox(modelo_2)
```

```
##  lags statistic df    p-value
##     5  10.13767  3 0.01743138
##    10  12.35985  8 0.13585575
##    15  24.57866 13 0.02620591
##    20  32.43109 18 0.01953833
##    25  40.06122 23 0.01512804
##    30  40.85073 28 0.05542116
```

```r
#Esto modelo no es adecuado puesto que hay varios resagos signi
#cativos.
LjungBox(modelo_3)
```

```
##  lags statistic df     p-value
##     5  13.71132  4 0.008275737
##    10  15.82853  9 0.070549908
##    15  30.84065 14 0.005835290
##    20  39.31354 19 0.004022568
##    25  47.92050 24 0.002581953
##    30  48.71633 29 0.012371841
```

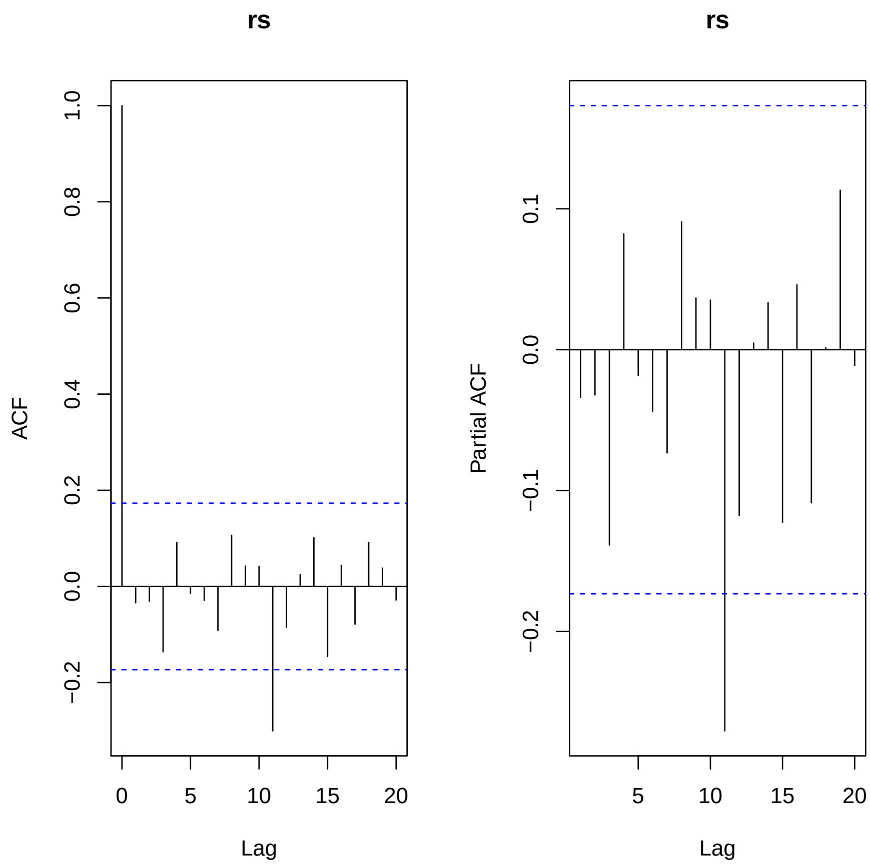
```r
#Esto modelo no es adecuado puesto que hay varios resagos signi
#cativos.
LjungBox(modelo_3)
```

```
##  lags statistic df     p-value
##     5  13.71132  4 0.008275737
```

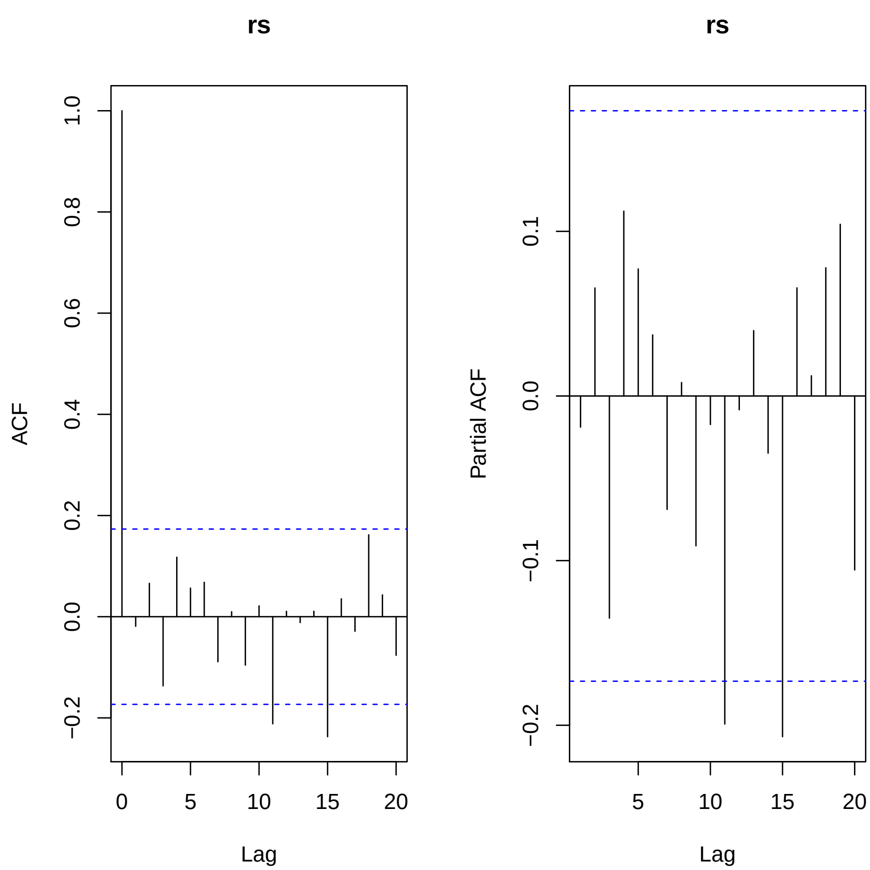
```
##    10  15.82853  9 0.070549908
##    15  30.84065 14 0.005835290
##    20  39.31354 19 0.004022568
##    25  47.92050 24 0.002581953
##    30  48.71633 29 0.012371841

#Esto modelo no es adecuado puesto que hay varios resagos signi
#cativos
LjungBox(modelo_4)

##  lags statistic df     p-value
##     5  13.38148  5 0.020054457
##    10  15.59562 10 0.111808290
##    15  30.73165 15 0.009540167
##    20  39.00873 20 0.006650747
##    25  47.46115 25 0.004324309
##    30  48.25026 30 0.018714785

#Esto modelo no es adecuado puesto que hay varios resagos signi
#cativos
```

El unico modelo que salio adecuado bajo este test es el $modelo_1$

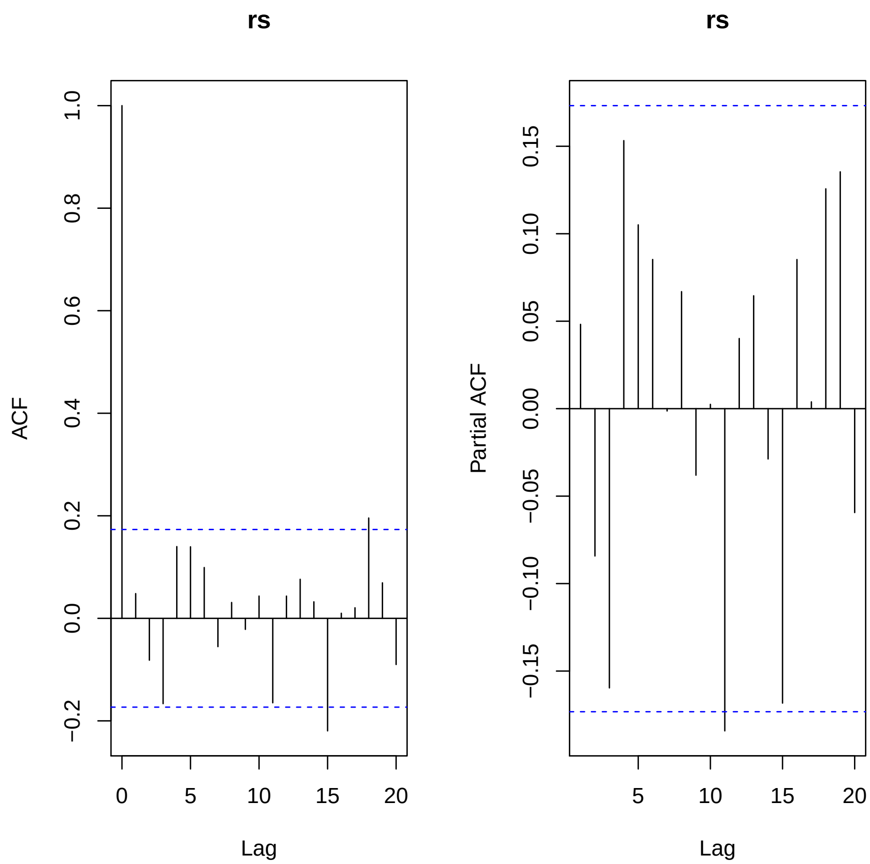
```
#diagnstico
#Ruido blanco

par(mfrow = c(1,2))
acf(main="rs",residuals(modelo_autoarima)/sqrt(as.numeric(modelo_autoarima[2] )),lag.max =20
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelo_autoarima)/sqrt(as.numeric(modelo_autoarima[2] )),lag.max =2
```



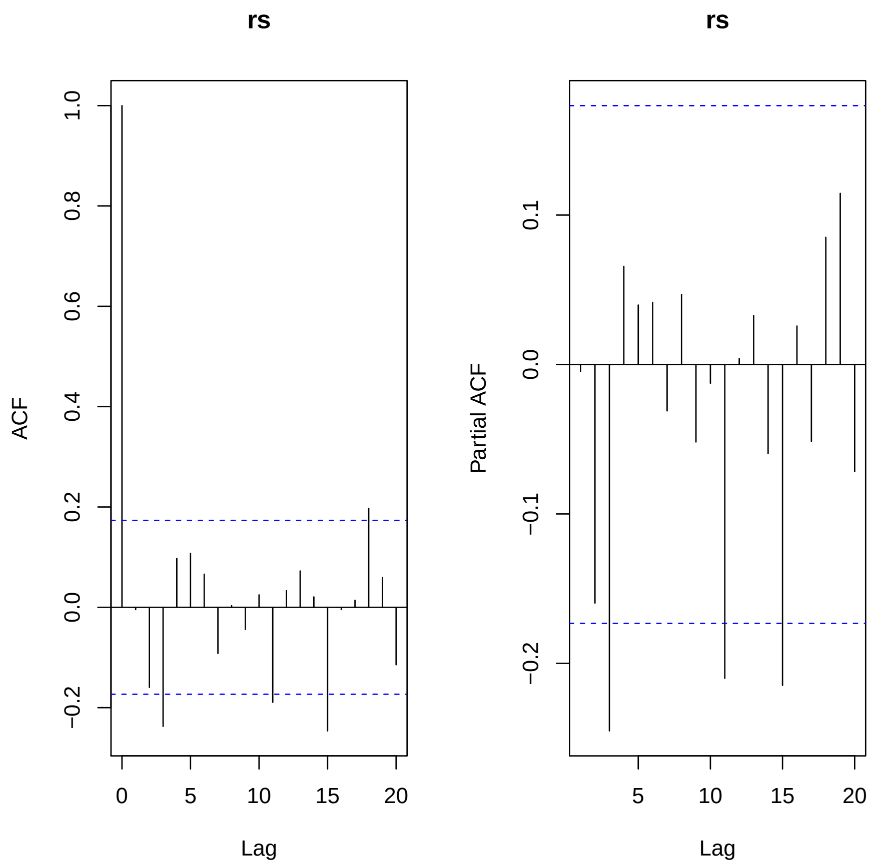

```r
#El resago 11 se sale de la banda


###########################################

#modelo_1

acf(main="rs",residuals(modelo_1)/sqrt(as.numeric(modelo_1[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelo_1)/sqrt(as.numeric(modelo_1[2] )),lag.max =20)
```



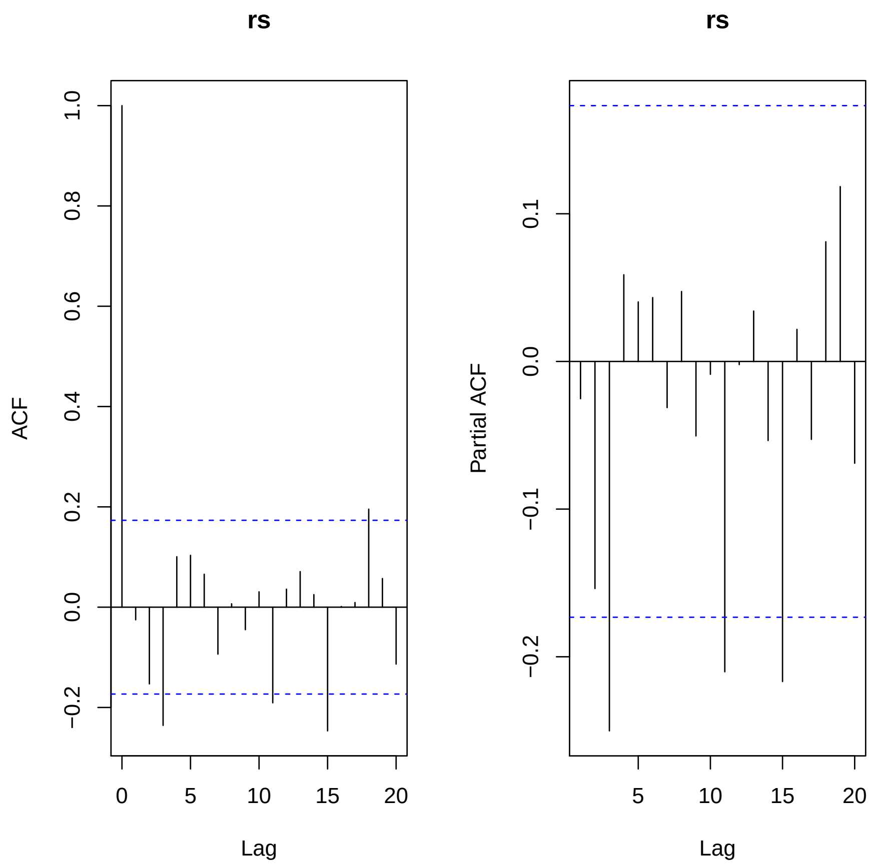

```
# el resago 11y 15 se salen de la banda de confianza

##########################################

#modelo_2
acf(main="rs",residuals(modelo_2)/sqrt(as.numeric(modelo_2[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelo_2)/sqrt(as.numeric(modelo_2[2] )),lag.max =20)
```




```
#en este modelo se salen de la banda el resago 11 y 15 pero de una manera muy leve

############################################
#modelo_3
acf(main="rs",residuals(modelo_3)/sqrt(as.numeric(modelo_3[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelo_3)/sqrt(as.numeric(modelo_3[2] )),lag.max =20)
```




```
##vemos que hay muchos resagos por fuera de la banda


########################################
#modelo_4
acf(main="rs",residuals(modelo_4)/sqrt(as.numeric(modelo_4[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelo_4)/sqrt(as.numeric(modelo_4[2] )),lag.max =20)
```




```
##vemos que hay muchos resagos por fuera de la banda
```

se observa que algunos modelos no cumplen que los valores de las acf sean estadisticamente cercanas a cero, por lo que esto dara peso a la hora de escoger un modelo. el $modelo_2$ fue el que mejor cumplio las espectativas de los acf y pacf estandarizados.

Ahora miremos la significancia de los parametros:

```
#modelo_autoarima
t <- (modelo_autoarima$coef[1])/(sqrt(modelo_autoarima$var.coef[1,1]))
abs(t)>qnorm(.95)

##  ar1
## TRUE

t <- (modelo_autoarima$coef[2])/(sqrt(modelo_autoarima$var.coef[2,2]))
```

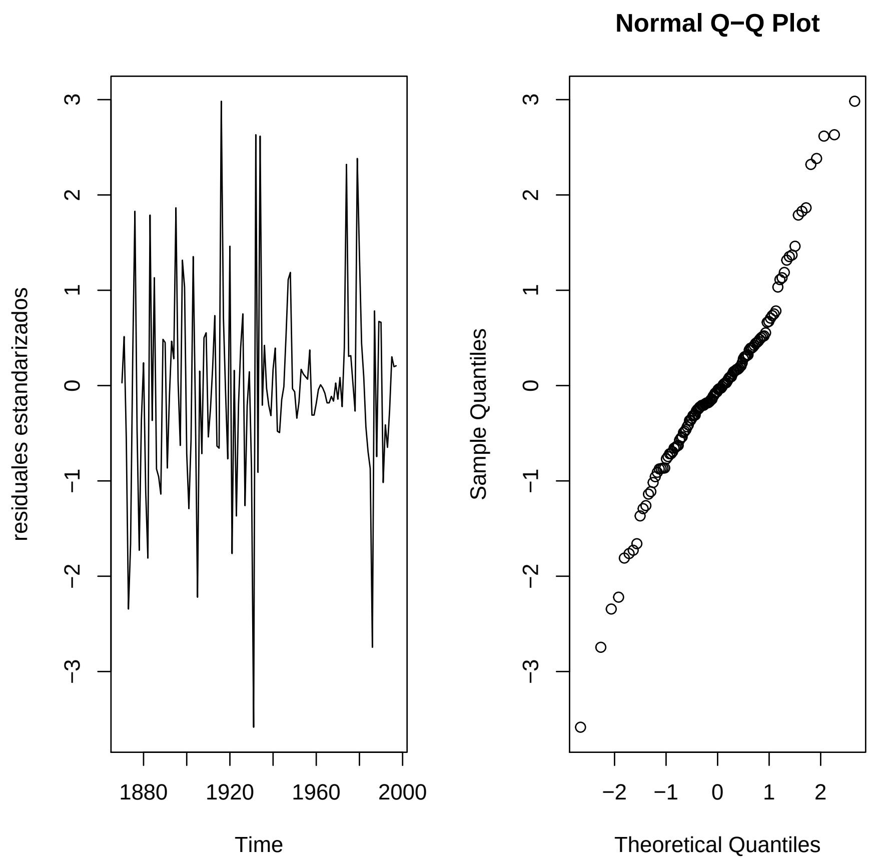
```r
abs(t)>qnorm(.95)

##  ar2
## TRUE

t <- (modelo_autoarima$coef[3])/(sqrt(modelo_autoarima$var.coef[3,3]))
abs(t)>qnorm(.95)

##  ma1
## TRUE

t <- (modelo_autoarima$coef[4])/(sqrt(modelo_autoarima$var.coef[4,4]))
abs(t)>qnorm(.95)

##  ma2
## TRUE

#todos los parametros son significativamente estadisticos
#####################################################################3
#modelo_1:
t <- (modelo_1$coef[1])/(sqrt(modelo_1$var.coef[1,1]))
abs(t)>qnorm(.95)

##  ar1
## TRUE

t <- (modelo_1$coef[2])/(sqrt(modelo_1$var.coef[2,2]))
abs(t)>qnorm(.95)

##  ar2
## TRUE

t <- (modelo_1$coef[3])/(sqrt(modelo_1$var.coef[3,3]))
abs(t)>qnorm(.95)

##  ma1
## TRUE

#vemos que todos los parametros son estadisticamente significativos

######################################################################
#modelo_2:
t <- (modelo_2$coef[1])/(sqrt(modelo_2$var.coef[1,1]))
abs(t)>qnorm(.95)

##  ar1
## TRUE
```

```r
t <- (modelo_2$coef[2])/(sqrt(modelo_2$var.coef[2,2]))
abs(t)>qnorm(.95)

##  ma1
## TRUE

#se observa que los parametros son estadisticamente significativos

##################################################################
#modelo_3:
t <- (modelo_3$coef[1])/(sqrt(modelo_3$var.coef[1,1]))
abs(t)>qnorm(.95)

##   ar1
## FALSE

#el parametro no es estadisticamnete significativo
```

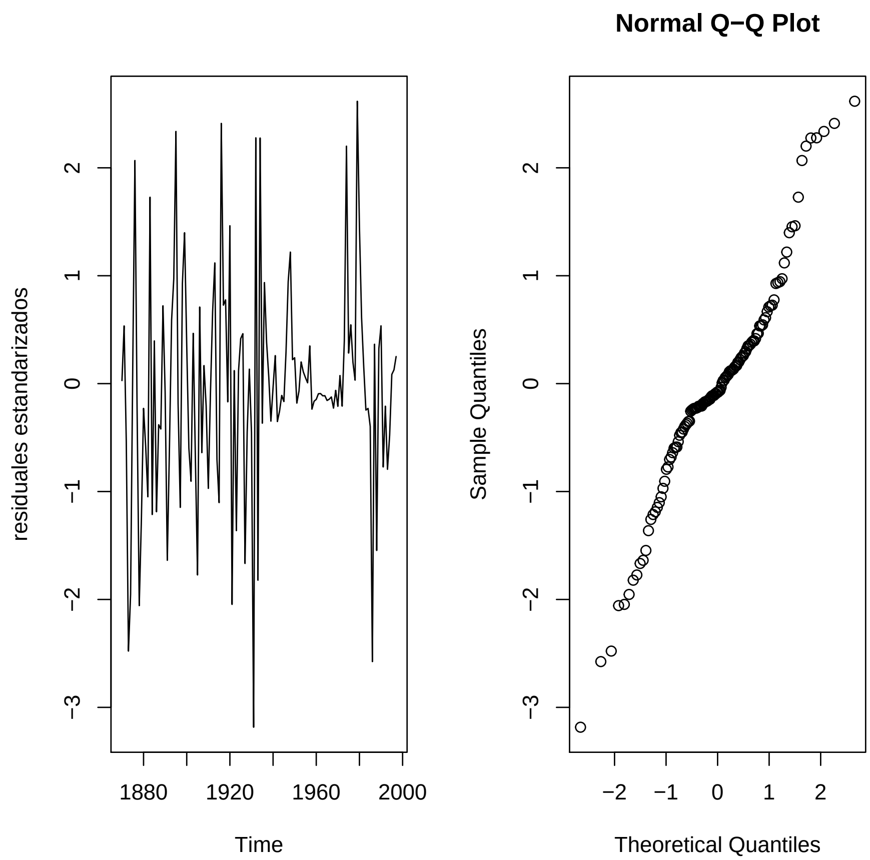
Apartir de lo anteriror nos quedaremos con los modelos $modelo_{(autoarima)}$, $modelo_1$, $modelo_2$, puesto que los otros dos modelos los parametros nos nos estadisticamente significativos por lo que de aqui en adelnate procederemos con los tres modelos dichos anteriormente.

Acontinuacion se mostrara si se cumple la normalidad de los residuales:

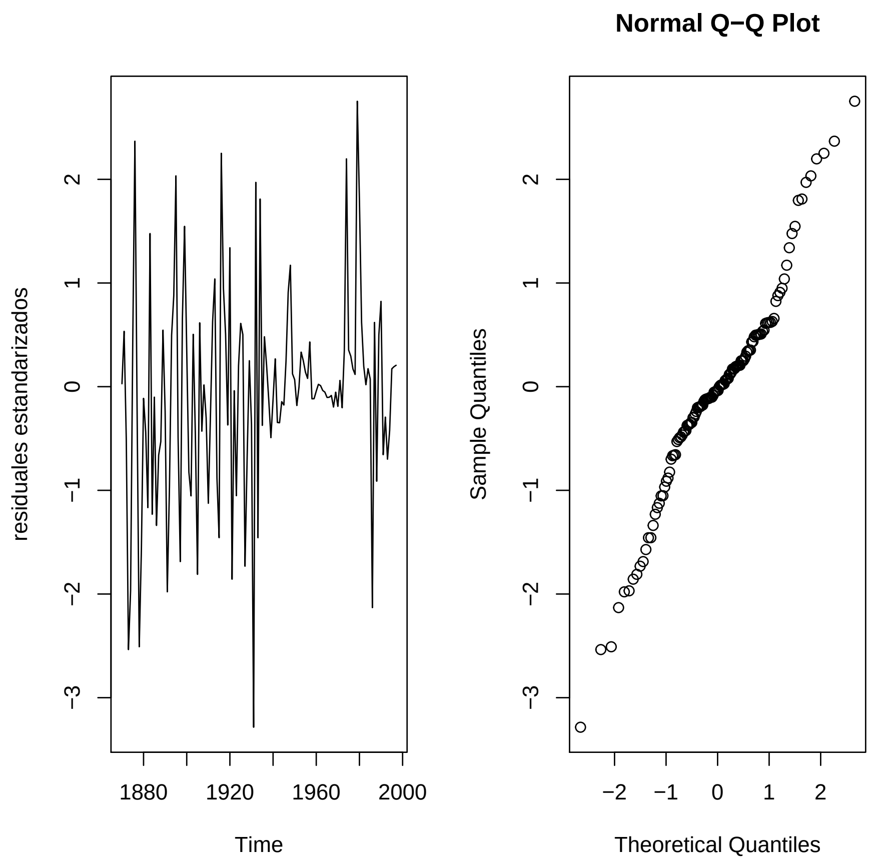
```r
par(mfrow = c(1,2))
#modelo_autoarima:
plot(ylab="residuales estandarizados",
     residuals(modelo_autoarima)/sqrt(as.numeric(modelo_autoarima[2] )))
shapiro.test(residuals(modelo_autoarima)/sqrt(as.numeric(modelo_autoarima[2] ))) #Segun del

##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo_autoarima)/sqrt(as.numeric(modelo_autoarima[2]))
## W = 0.95214, p-value = 0.0001881

qqnorm(residuals(modelo_autoarima)/sqrt(as.numeric(modelo_autoarima[2])))
```




```
#######################################################################3
#mdoelo_1
plot(ylab="residuales estandarizados",
     residuals(modelo_1)/sqrt(as.numeric(modelo_1[2] )))
shapiro.test(residuals(modelo_1)/sqrt(as.numeric(modelo_1[2] )))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo_1)/sqrt(as.numeric(modelo_1[2]))
## W = 0.95589, p-value = 0.0003716

#Segun del test de shapiro-wilk no hay normalidad.
qqnorm(residuals(modelo_1)/sqrt(as.numeric(modelo_1[2])))
```




```
#######################################################################
#modelo_2

plot(ylab="residuales estandarizados",
     residuals(modelo_2)/sqrt(as.numeric(modelo_2[2] )))
shapiro.test(residuals(modelo_2)/sqrt(as.numeric(modelo_2[2] )))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo_2)/sqrt(as.numeric(modelo_2[2]))
## W = 0.96069, p-value = 0.0009224

#Segun del test de shapiro-wilk no hay normalidad.
qqnorm(residuals(modelo_2)/sqrt(as.numeric(modelo_2[2])))
```




```
######################################################################
```

Veamos que tanto acierta en tendencia los modelos , partiremos la serie en un70-30 y que apartir del 70 pronostique , luego comparamos las estimaciones con los datos reales para mirar cual de todos los modelos es mas adecuado.

```
a <- c()
setentaporcidatos <- floor(70*length(oilprice2)/100)

d=89

while(d >88 & d<(length(oilprice2))){
  s1 <- ts(oilprice2[1:d],start=1,end=d,frequency=1)
  mod1=arima(s1,order=c(2,1,2))
  fore1=forecast(mod1, h=1)
  x <- fore1$mean[1]
```

```
  a <- c(a,x)
  d=d+1
}
```

MAE del modelo propuesto bajo las $ACF$ y $PACF$:

```
mae <- mean(abs(a-oilprice2[90:128]))
mae

## [1] 0.04328013
```

MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```
mse <- mean((a-oilprice2[90:128])^2)
mse

## [1] 0.004021019
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```
#at=aciertos en tendencia

h <- oilprice2[90:128]
d <- c()
for (i in 1:(length(a)-1)) {
  d[i] <- (a[i+1]-a[i])/i #medias para los pronosticos

}
m <- c()
for (i in 1:(length(a)-1)) {
  m[i] <- (h[i+1]-h[i])/i
}
l <- c()
y <- c()
x <- d*m

for (i in 1:length(x)) {
  if(x[i]<0){
    y <- c(y,x[i])}

}
 #cantidad de prodcutos negativos que en realidad son los que no aciertan en tendencia

diferencia <- length(oilprice2)-length(oilprice2[90:128])
# se miraran los 72 pronosticos ya que los datos se partieron en un 70-30
```

```
l <- diferencia-length(y)
#cantidad de productos positivos osea los que si
#aciertan en tendencia
porcentajeaciertotendencia <- l*100/diferencia
porcentajeaciertotendencia

## [1] 80.89888
```

Ahora veamos con el otro $modelo_1$:
MAE del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 0.0404032
```

MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 0.004067387
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 86.51685
```

Mejoro un poco el acierto en tendencia para el $modelo_1$, veamos que pasa con el $modelo_2$:

MAE del modelo propuesto bajo las $ACF$ y $PACF$:

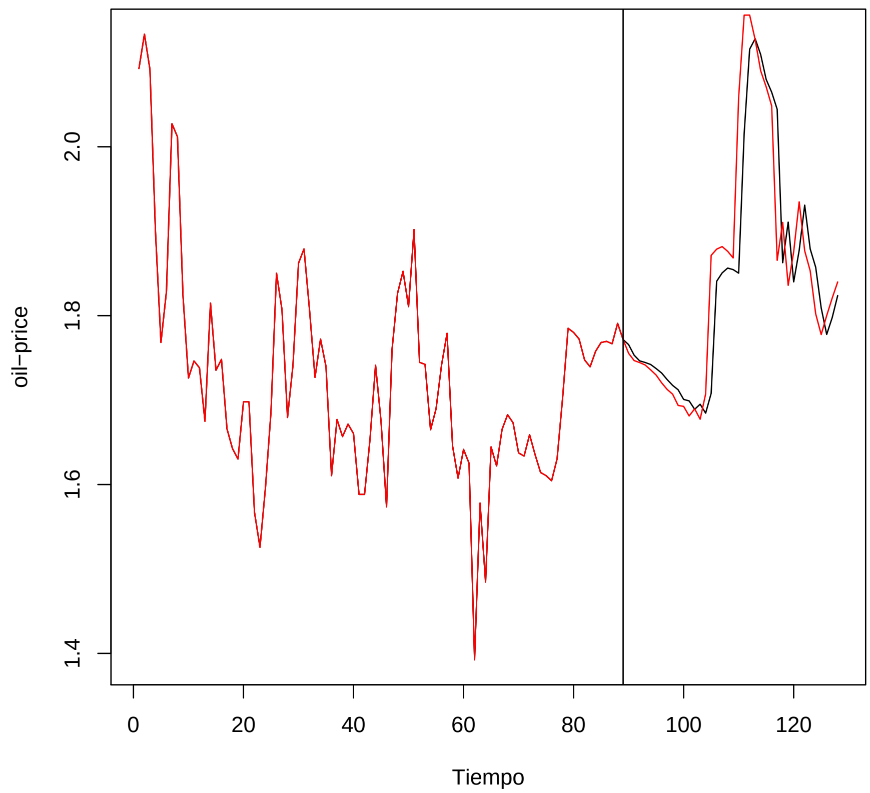
```
## [1] 0.03791126
```

MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 0.003862609
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 87.64045
```




Se observa que el $modelo_2$ tuvo un mejor acierto en tendencia que los otros dos modelos ,la serie negra representa el pronsotico bajo el $modelo_2$ apartir del 70% de la serie.
Se sabe que siempre hay que tratar de dejar el modelo mas simple , asi que nos quedaremos con el $modelo_2$ que al igual que los otros dos modelos cumplen ciertos supuestos pero es mejor quedarnos con elmodelo mas simple.

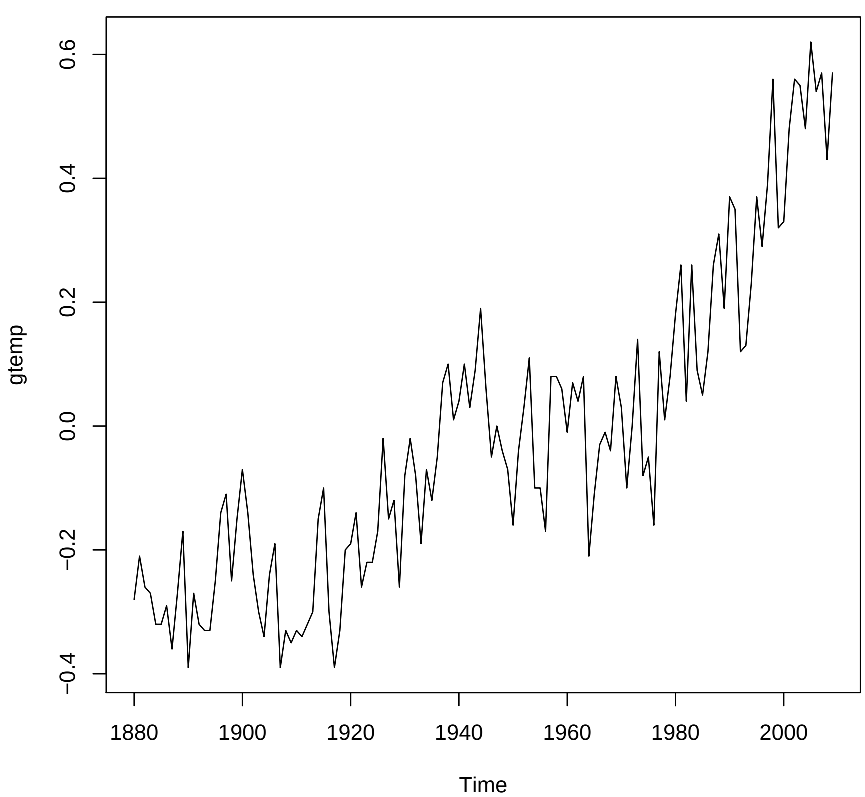
Modelo final: $Modelo_2 = arima(oilprice2, c(1, 1, 1))$

punto 7:
Fit an $ARIMA(p, d, q)$ model to the global temperature data gtemp performing all of the necessary diagnostics. After deciding on an appropriate model, forecast (with limits) the next 10 years. Comment.

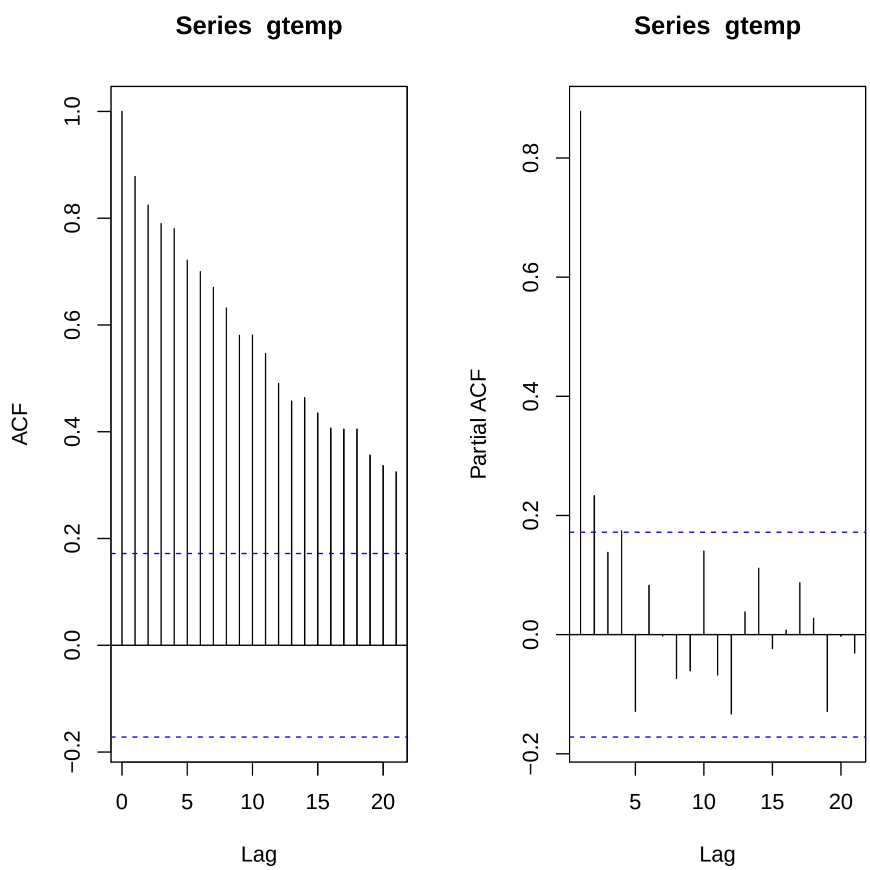
```
library(astsa)

##
```

```
## Attaching package:  'astsa'
## The following object is masked from 'package:fpp':
##
##     oil
## The following objects are masked from 'package:fma':
##
##     chicken, sales
## The following object is masked from 'package:forecast':
##
##     gas
```

```
data("gtemp")
plot.ts(gtemp)
```



```
par(mfrow = c(1,2))
acf(gtemp)
pacf(gtemp)
```




Observamos que la acf de la serie decae de manera muy lenta , lo que sugiere diferenciarla , pero antes de esto notemos que tambien la serie no es estacionaria ne varianza , por lo que primero procedemos a realizar una trnasformacion, de entrada descartamos las transfromaciones logaritmo y raiz debido a los valores negativos de la serie.

Procedemos con Box-Cox:

```
la <- BoxCox.lambda(gtemp)
```

Ahora que tenemos el lambda y vemos que es cercano a 1 entonces no realizamos ninguna transformacion.

```r
auto.arima(gtemp)

## Series: gtemp
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1   drift
##       0.2570  -0.7854  0.0064
## s.e.  0.1177   0.0707  0.0025
##
## sigma^2 estimated as 0.009381:  log likelihood=119.34
## AIC=-230.68   AICc=-230.35   BIC=-219.24

modeloauto <- arima(gtemp,c(1,1,1)) #, fixed = c(NA,0,NA,NA))
modeloauto

##
## Call:
## arima(x = gtemp, order = c(1, 1, 1))
##
## Coefficients:
##          ar1      ma1
##       0.2256  -0.7158
## s.e.  0.1235   0.0792
##
## sigma^2 estimated as 0.009539:  log likelihood = 116.83,  aic = -227.65

modelopro <- arima(gtemp, c(1,1,0))
modelopro

##
## Call:
## arima(x = gtemp, order = c(1, 1, 0))
##
## Coefficients:
##           ar1
##       -0.2925
## s.e.   0.0845
##
## sigma^2 estimated as 0.01096:  log likelihood = 108.03,  aic = -212.07

modelopro1 <- arima(gtemp,c(0,1,0))
modelopro1

##
```

```
## Call:
## arima(x = gtemp, order = c(0, 1, 0))
##
##
## sigma^2 estimated as 0.01199:  log likelihood = 102.31,  aic = -202.62
```

```
modelopro2 <- arima(gtemp,c(2,1,1))
modelopro2
```

```
##
## Call:
## arima(x = gtemp, order = c(2, 1, 1))
##
## Coefficients:
##          ar1      ar2      ma1
##       0.1576  -0.1490  -0.6252
## s.e.  0.1403   0.1057   0.1190
##
## sigma^2 estimated as 0.009398:  log likelihood = 117.76,  aic = -227.52
```

```
modelopro3 <- arima(gtemp,c(2,1,2))
modelopro3
```

```
##
## Call:
## arima(x = gtemp, order = c(2, 1, 2))
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.8891  -0.3449  -1.3721  0.5701
## s.e.  0.3061   0.1416   0.3019  0.2306
##
## sigma^2 estimated as 0.009306:  log likelihood = 118.35,  aic = -226.71
```

Dado el modelo propuesto por el autoarima , si ponemos a variar los parametros ar y ma mayores a 1 la desviacion estandar de las estimaciones de los parametros aumentan , los $AIC$ y $\sigma^2$ estimado no varian mucho .

Ahora procedamos a mirar si para cada modelo un grupo de autocrrelaciones son diferentes de cero.

```
LjungBox(modeloauto)
```

```
##  lags statistic df    p-value
##     5  7.997216  3 0.04606927
```

```
##    10 13.920403  8 0.08386381
##    15 22.375475 13 0.04981020
##    20 26.601863 18 0.08677187
##    25 30.910365 23 0.12501119
##    30 38.883533 28 0.08281446
```

```r
LjungBox(modelopro)
```

```
##  lags statistic df      p-value
##     5  23.04473  4 1.240471e-04
##    10  30.40011  9 3.749714e-04
##    15  42.69856 14 9.571511e-05
##    20  48.27883 19 2.337726e-04
##    25  55.68050 24 2.532656e-04
##    30  67.15144 29 7.362011e-05
```

```r
LjungBox(modelopro1)
```

```
##  lags statistic df      p-value
##     5  26.53305  5 7.031443e-05
##    10  37.11424 10 5.407497e-05
##    15  46.23956 15 4.870143e-05
##    20  53.75365 20 6.291787e-05
##    25  59.64799 25 1.168890e-04
##    30  71.93005 30 2.664933e-05
```

```r
LjungBox(modelopro2)
```

```
##  lags statistic df    p-value
##     5  7.057352  2 0.02934374
##    10 13.246382  7 0.06632809
##    15 20.063485 12 0.06589453
##    20 24.486525 17 0.10680703
##    25 27.684671 22 0.18636815
##    30 34.448840 27 0.15338420
```

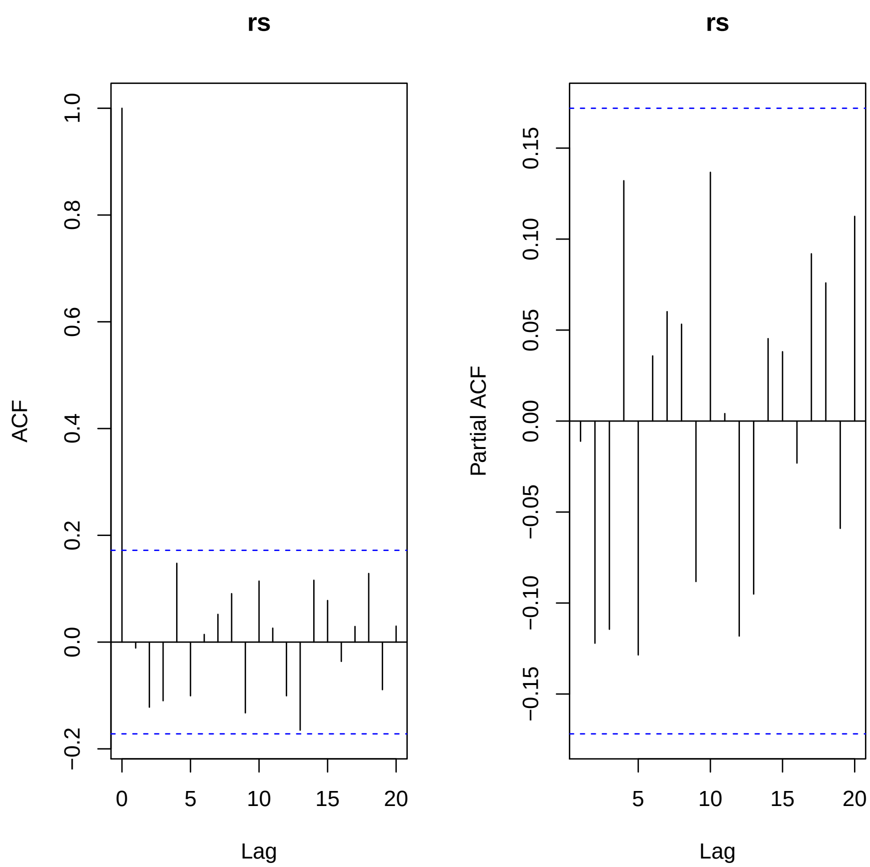
```r
LjungBox(modelopro3)
```

```
##  lags statistic df    p-value
##     5  6.749964  1 0.00937496
##    10 12.457061  6 0.05251539
##    15 19.698745 11 0.04964734
##    20 24.575829 16 0.07766049
##    25 28.199351 21 0.13458751
##    30 35.345250 26 0.10438346
```

Observemos que para algunos modelos propuestos hay un grupo de resagos distintos de algo estadisticamente cero.

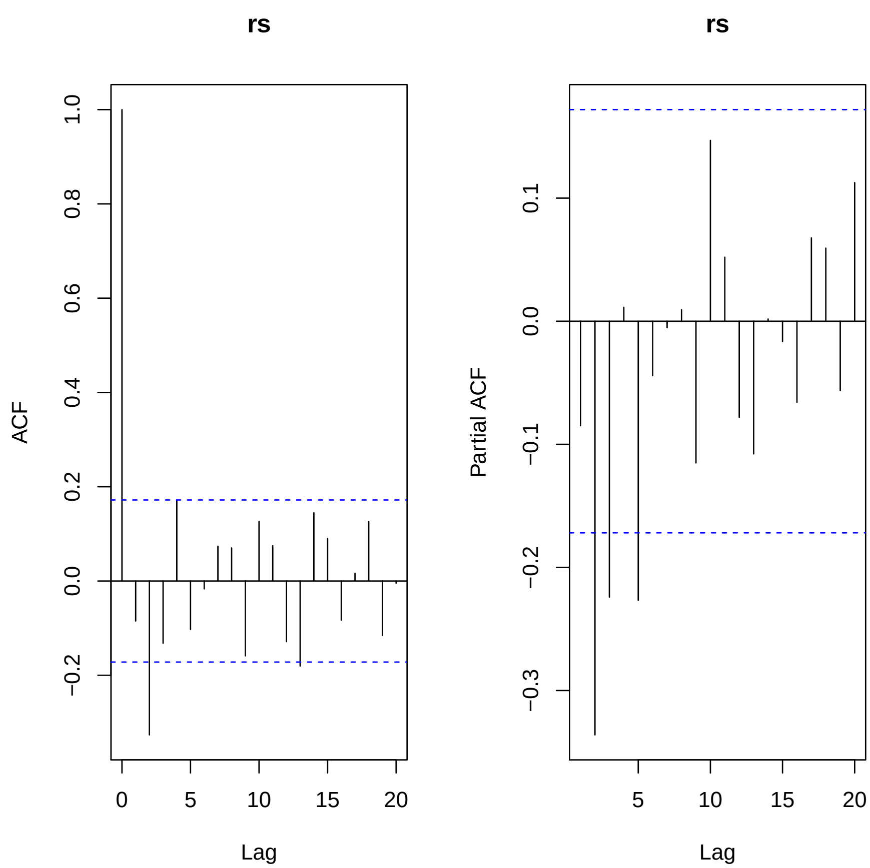
Procedamos a realizar el diagnostico ruido blanco para cada modelo:

```
#diagnstico
#Ruido blanco
#modelo_auto
par(mfrow = c(1,2))
acf(main="rs",residuals(modeloauto)/sqrt(as.numeric(modeloauto[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modeloauto)/sqrt(as.numeric(modeloauto[2] )),lag.max =20)
```

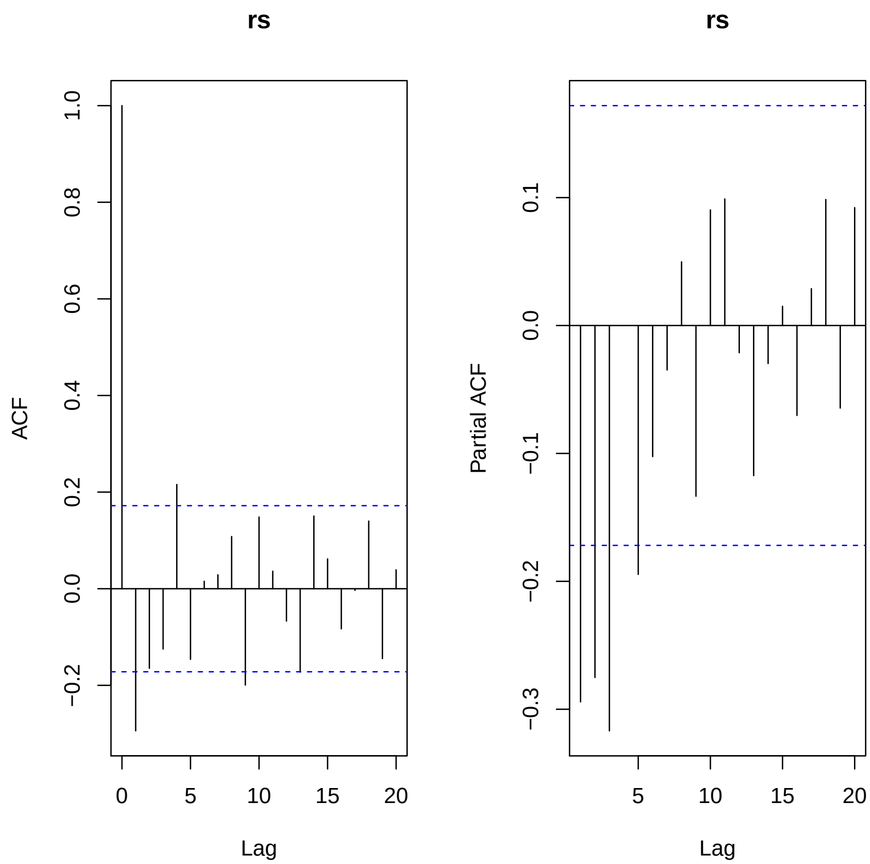



```
#todos los resagos estan dentro de la franja
#############################################
#modelopro
acf(main="rs",residuals(modelopro)/sqrt(as.numeric(modelopro[2] )),lag.max =20)
```

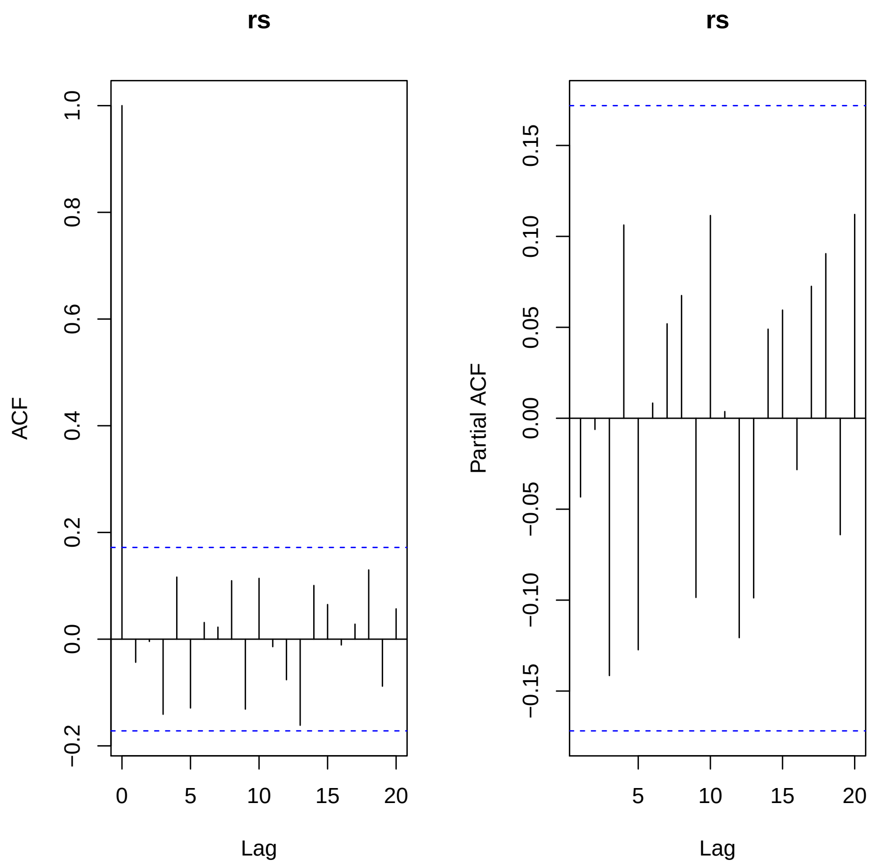
```
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelopro)/sqrt(as.numeric(modelopro[2] )),lag.max =20)
```



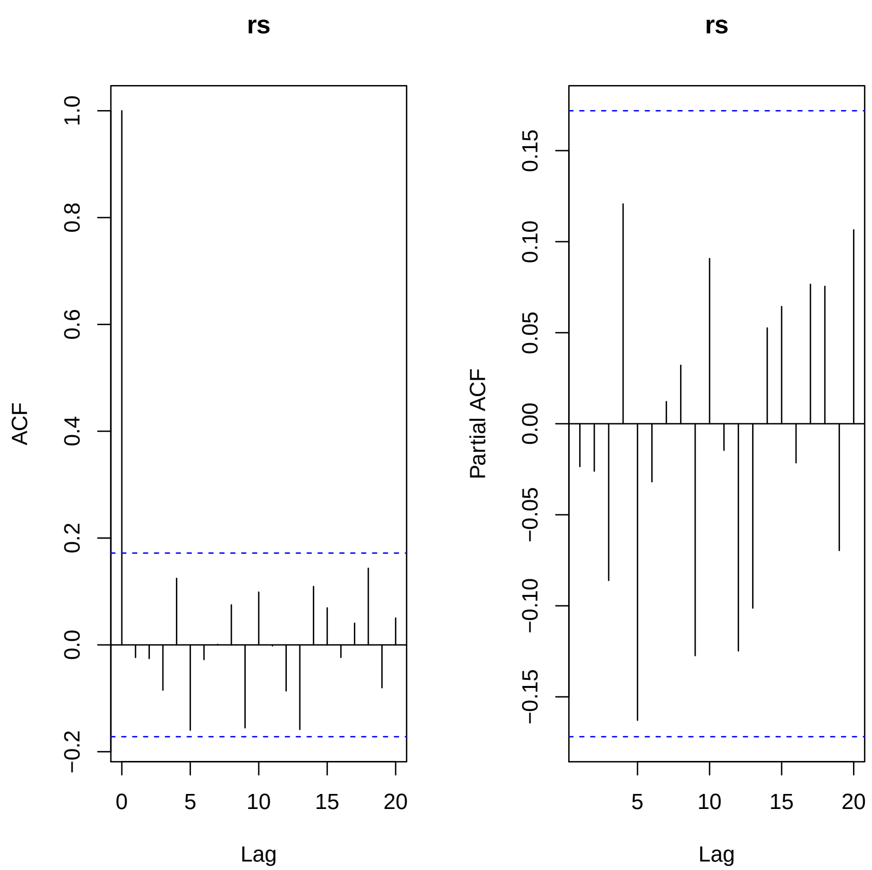

```
#problemas en los primeros resagos
###################################################3
#modelopro1
acf(main="rs",residuals(modelopro1)/sqrt(as.numeric(modelopro1[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelopro1)/sqrt(as.numeric(modelopro1[2] )),lag.max =20)
```




```
#problemas en los primeros resagos
####################################################
# modelopro2
acf(main="rs",residuals(modelopro2)/sqrt(as.numeric(modelopro2[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelopro2)/sqrt(as.numeric(modelopro2[2] )),lag.max =20)
```




```r
#los resagos estan dentro de las bandas
##################################################
#modelopro3
acf(main="rs",residuals(modelopro3)/sqrt(as.numeric(modelopro3[2] )),lag.max =20)
### lo dividimos para estandarizados sqrt
pacf(main="rs",residuals(modelopro3)/sqrt(as.numeric(modelopro3[2] )),lag.max =20)
```




```
#los resagos estan dentro de las bandas
```

Observamos que los ultimos dos modelos presentan mejores representaciones de las *acf* y *pacf* estandarizados , junto con la del modelo propuesto por el autoarima, debido que los resagos estan por dentro de las bandas.

```
#adecuacidad del modelo  H_0:modelo adecuado

LjungBox(residuals(modeloauto)/sqrt(as.numeric(modeloauto[2] ))) ###verifica si son indepen

##  lags statistic df    p-value
##     5  7.997216  5 0.15638910
##    10 13.920403 10 0.17665332
##    15 22.375475 15 0.09833003
##    20 26.601863 20 0.14684141
##    25 30.910365 25 0.19200402
##    30 38.883533 30 0.12835937
```

```
#como, p-valor muy grande no rechaza. son independientes
LjungBox(residuals(modelopro)/sqrt(as.numeric(modelopro[2] )))

##  lags statistic df      p-value
##     5  23.04473  5 0.0003309837
##    10  30.40011 10 0.0007365898
##    15  42.69856 15 0.0001753394
##    20  48.27883 20 0.0003887044
##    25  55.68050 25 0.0004004257
##    30  67.15144 30 0.0001156072

LjungBox(residuals(modelopro1)/sqrt(as.numeric(modelopro1[2] )))

##  lags statistic df      p-value
##     5  26.53305  5 7.031443e-05
##    10  37.11424 10 5.407497e-05
##    15  46.23956 15 4.870143e-05
##    20  53.75365 20 6.291787e-05
##    25  59.64799 25 1.168890e-04
##    30  71.93005 30 2.664933e-05

LjungBox(residuals(modelopro2)/sqrt(as.numeric(modelopro2[2] )))

##  lags statistic df   p-value
##     5  7.057352  5 0.2164098
##    10 13.246382 10 0.2102208
##    15 20.063485 15 0.1695108
##    20 24.486525 20 0.2217854
##    25 27.684671 25 0.3225723
##    30 34.448840 30 0.2632460

LjungBox(residuals(modelopro3)/sqrt(as.numeric(modelopro3[2] )))

##  lags statistic df   p-value
##     5  6.749964  5 0.2399098
##    10 12.457061 10 0.2556306
##    15 19.698745 15 0.1837963
##    20 24.575829 20 0.2181430
##    25 28.199351 25 0.2987580
##    30 35.345250 30 0.2302951
```

En el modelo dado por el auto.arima junto con los dos ultimos modelos son los que se adecuan a estos residuales mejor.

Procedamos a mirar la significancia de los parametros de cada modelo propuesto:

```r
# modeloauto
t <- (modeloauto$coef[1])/(sqrt(modeloauto$var.coef[1,1]))
abs(t)>qnorm(.95)

##  ar1
## TRUE

t <- (modeloauto$coef[2])/(sqrt(modeloauto$var.coef[2,2]))
abs(t)>qnorm(.95)

##  ma1
## TRUE

#los dos parametros son estadisticamente significativos
##########
# modelopro
t <- (modelopro$coef[1])/(sqrt(modelopro$var.coef[1,1]))
abs(t)>qnorm(.95)

##  ar1
## TRUE

#el parametro es significativo
########################
#modelopro2
t <- (modelopro2$coef[1])/(sqrt(modelopro2$var.coef[1,1]))
abs(t)>qnorm(.95)

##   ar1
## FALSE

t <- (modelopro2$coef[2])/(sqrt(modelopro2$var.coef[2,2]))
abs(t)>qnorm(.95)

##   ar2
## FALSE

t <- (modelopro2$coef[3])/(sqrt(modelopro2$var.coef[3,3]))
abs(t)>qnorm(.95)

##  ma1
## TRUE

#vemos que los parametros del ar no son significativamente estadisticos
#########################
# modelopro3
t <- (modelopro3$coef[1])/(sqrt(modelopro3$var.coef[1,1]))
abs(t)>qnorm(.95)
```

```
##  ar1
## TRUE
```

```
t <- (modelopro3$coef[2])/(sqrt(modelopro3$var.coef[2,2]))
abs(t)>qnorm(.95)
```

```
##  ar2
## TRUE
```

```
t <- (modelopro3$coef[3])/(sqrt(modelopro3$var.coef[3,3]))
abs(t)>qnorm(.95)
```

```
##  ma1
## TRUE
```

```
t <- (modelopro3$coef[4])/(sqrt(modelopro3$var.coef[4,4]))
abs(t)>qnorm(.95)
```

```
##  ma2
## TRUE
```

```
#todos los parametros son estadisticamente significativos
```

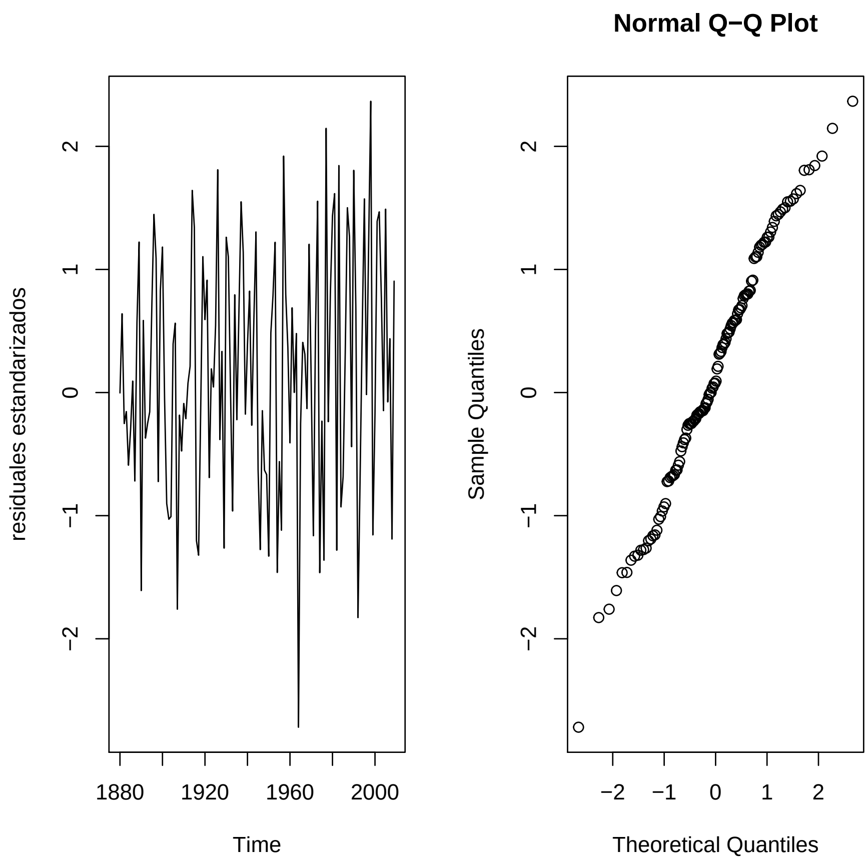
Se observa que en todos los modelos menos en el modelopro2 los parametros son estadisticamente significativos.

Veamos que cumple normalidad en los residuales:

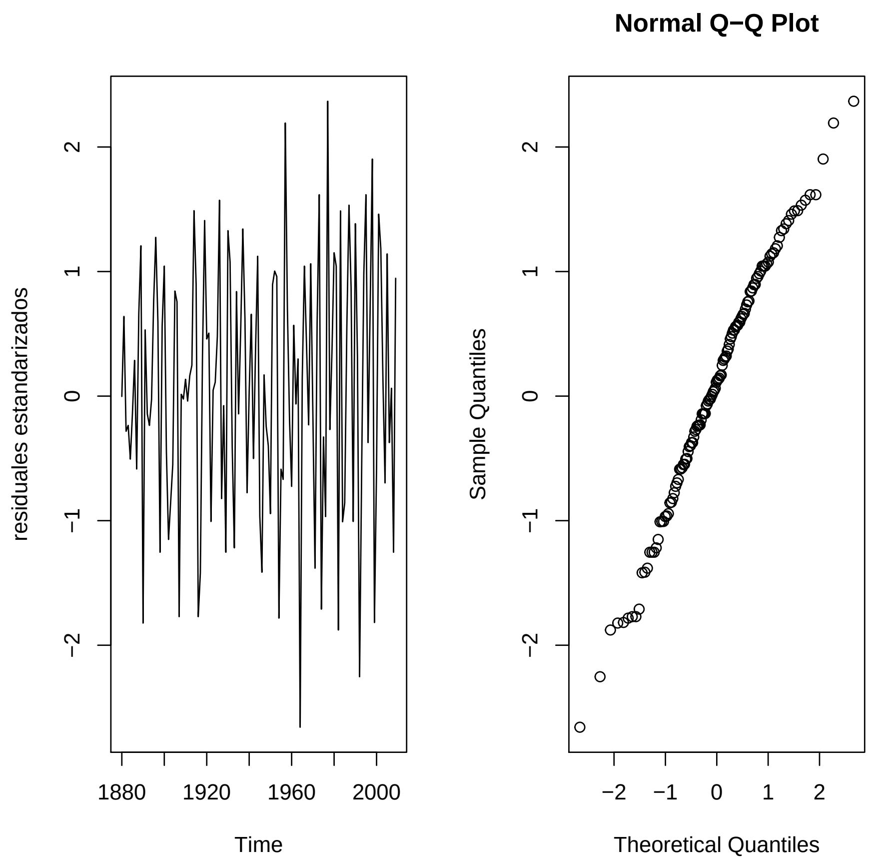
```
par(mfrow = c(1,2))
#modelo_autoarima:
plot(ylab="residuales estandarizados",
     residuals(modeloauto)/sqrt(as.numeric(modeloauto[2] )))
shapiro.test(residuals(modeloauto)/sqrt(as.numeric(modeloauto[2] )))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modeloauto)/sqrt(as.numeric(modeloauto[2]))
## W = 0.98884, p-value = 0.3774
```

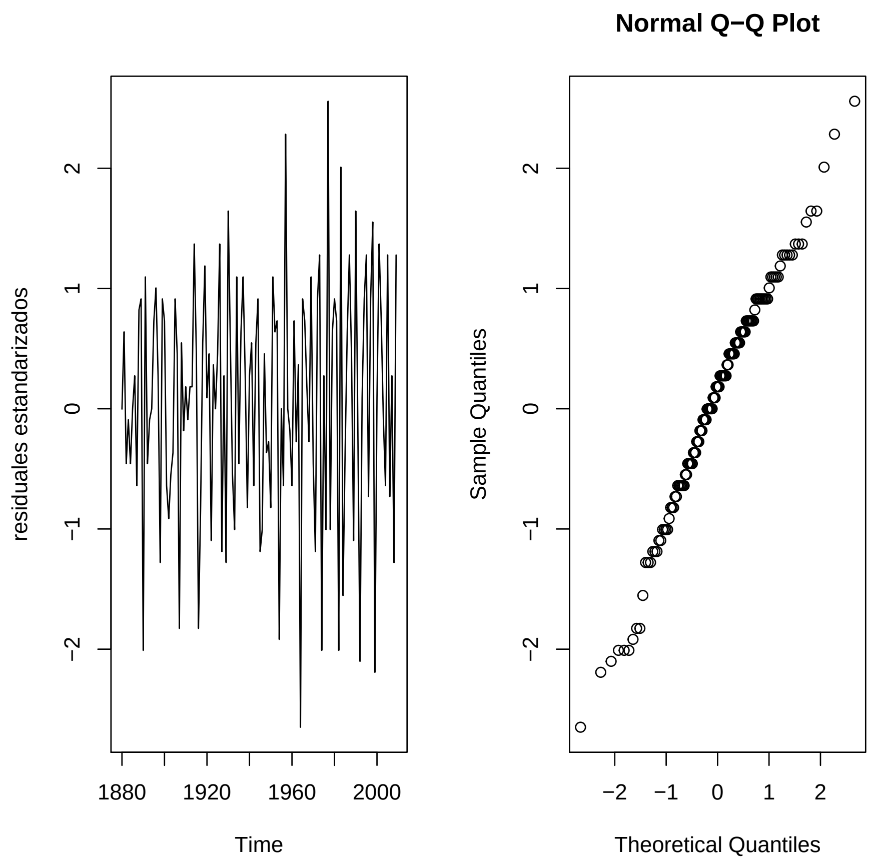
```
#Segun del test de shapiro-wilk hay normalidad.
qqnorm(residuals(modeloauto)/sqrt(as.numeric(modeloauto[2])))
```



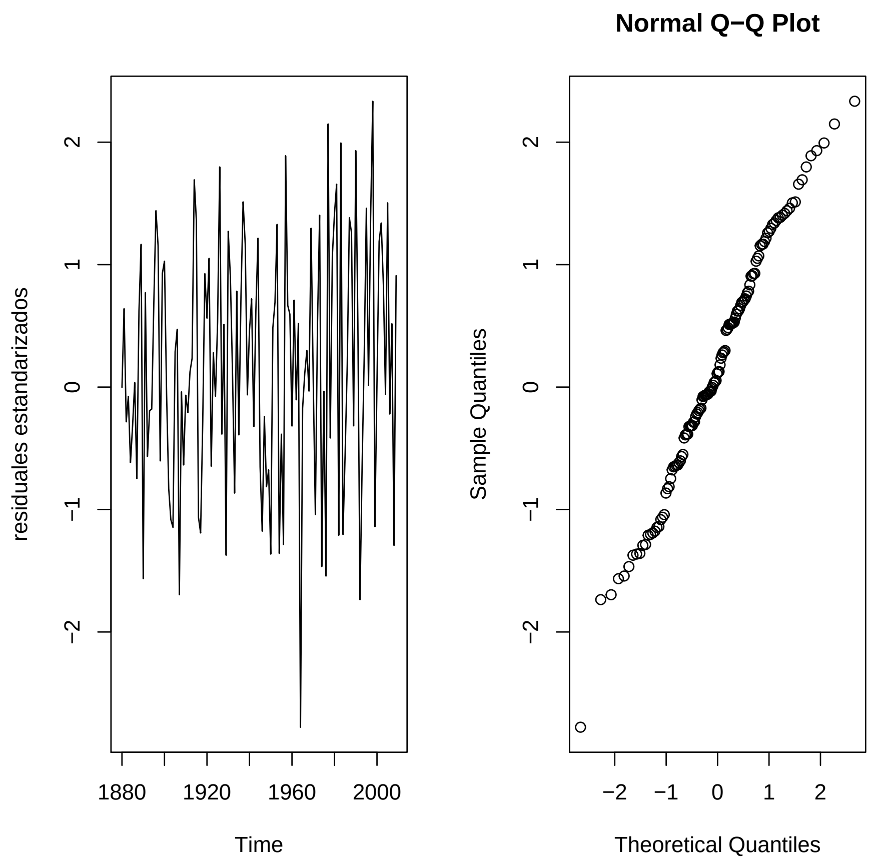

```
#vemos que tiene un comportamiento aleatorio y se le puede asociar una posible recta a los d
#############################################
#modelopro:
plot(ylab="residuales estandarizados",
     residuals(modelopro)/sqrt(as.numeric(modelopro[2] )))
shapiro.test(residuals(modelopro)/sqrt(as.numeric(modelopro[2] )))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelopro)/sqrt(as.numeric(modelopro[2]))
## W = 0.98943, p-value = 0.4244

#Segun del test de shapiro-wilk hay normalidad.
qqnorm(residuals(modelopro)/sqrt(as.numeric(modelopro[2])))
```



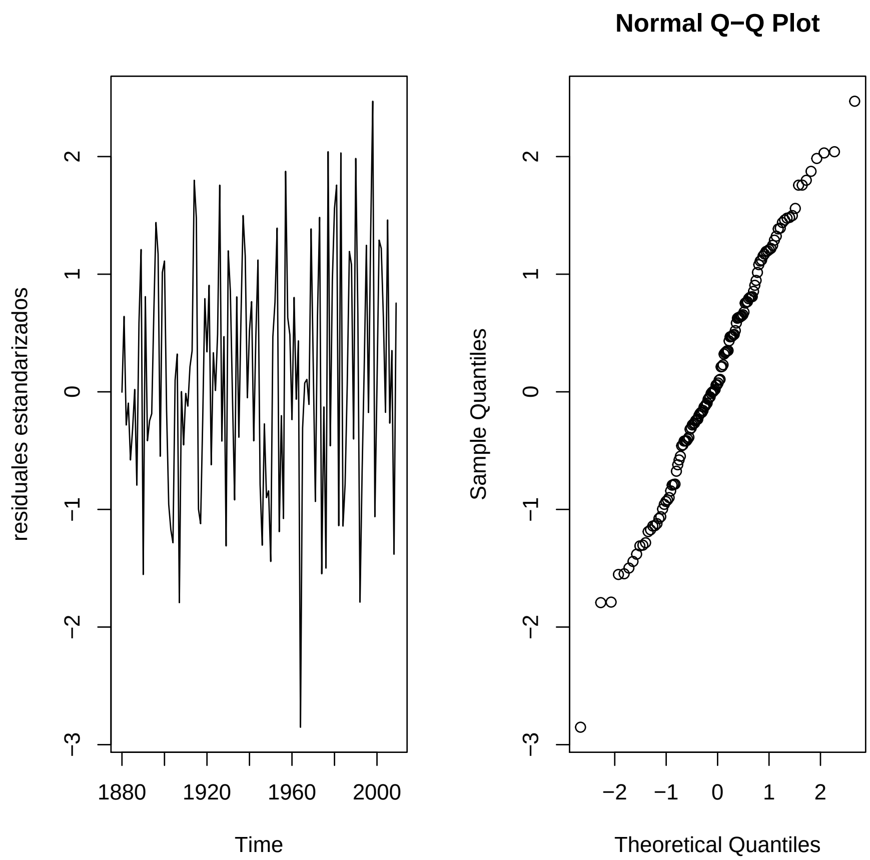

```
# en este modelo tambien se observa una aleatoriedad y
#al parecer se acomoda mas a una posible recta en el normal
#q-q plot pero hay que mirar las colas tambien
####################################################
#modelopro1:
plot(ylab="residuales estandarizados",
     residuals(modelopro1)/sqrt(as.numeric(modelopro1[2] )))
shapiro.test(residuals(modelopro1)/sqrt(as.numeric(modelopro1[2] )))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelopro1)/sqrt(as.numeric(modelopro1[2]))
## W = 0.98465, p-value = 0.1515

#Segun del test de shapiro-wilk  hay normalidad.
qqnorm(residuals(modelopro1)/sqrt(as.numeric(modelopro1[2])))
```




```
#con este modelo si se tiene un poco de cuidado puesto que
#si se observa un comoportamiento aleatorio pero debido a las
#colas no se le podria una adecuada recta
##################################################
#modelopro2:
plot(ylab="residuales estandarizados",
     residuals(modelopro2)/sqrt(as.numeric(modelopro2[2] )))
shapiro.test(residuals(modelopro2)/sqrt(as.numeric(modelopro2[2] )))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelopro2)/sqrt(as.numeric(modelopro2[2]))
## W = 0.98874, p-value = 0.3696

#Segun del test de shapiro-wilk  hay normalidad.
qqnorm(residuals(modelopro2)/sqrt(as.numeric(modelopro2[2])))
```




```
#ciertos puntos se alejan mas de la recta que se le podria
#asociar fijandonos un poco en las colas
###############################################
#modelopro3
plot(ylab="residuales estandarizados",
     residuals(modelopro3)/sqrt(as.numeric(modelopro3[2] )))
shapiro.test(residuals(modelopro3)/sqrt(as.numeric(modelopro3[2] )))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelopro3)/sqrt(as.numeric(modelopro3[2]))
## W = 0.99179, p-value = 0.6471

#Segun del test de shapiro-wilk  hay normalidad.
qqnorm(residuals(modelopro3)/sqrt(as.numeric(modelopro3[2])))
```




```
#se observa el mismo estilo de conclusion que en los otros modelos
```

En los anteriores graficos observamos que tenian comportamientos aleatorios los residuales pero estacionarios , el problema resulto ser en el grafico normal q-q plot que no se le puede asociar una debida recta debio a la variabilidad que podria ocasionar.

Veamos que tanto acierta en tendencia los modelos , partiremos la serie en un70-30 y que apartir del 70 pronostique y actualice la serie con el nuevo pronostico siempre, luego comparamos las estimaciones con los datos reales para mirar cual de todos los modelos es mas adecuado.

primer miremos el modeloauto

```r
#modeloauto
a <- c()
setentaporcidatos <- floor(70*length(gtemp)/100)

d=91

while(d >90 & d<(length(gtemp))){
  s1 <- ts(gtemp[1:d],start=1,end=d,frequency=1)
  mod1=arima(s1,order=c(1,1,1))
  fore1=forecast(mod1, h=1)
  x <- fore1$mean[1]
  a <- c(a,x)
  d=d+1
}
```

MAE del modelo propuesto bajo las $ACF$ y $PACF$:

```r
mae <- mean(abs(a-gtemp[92:130]))
mae

## [1] 0.1031003
```

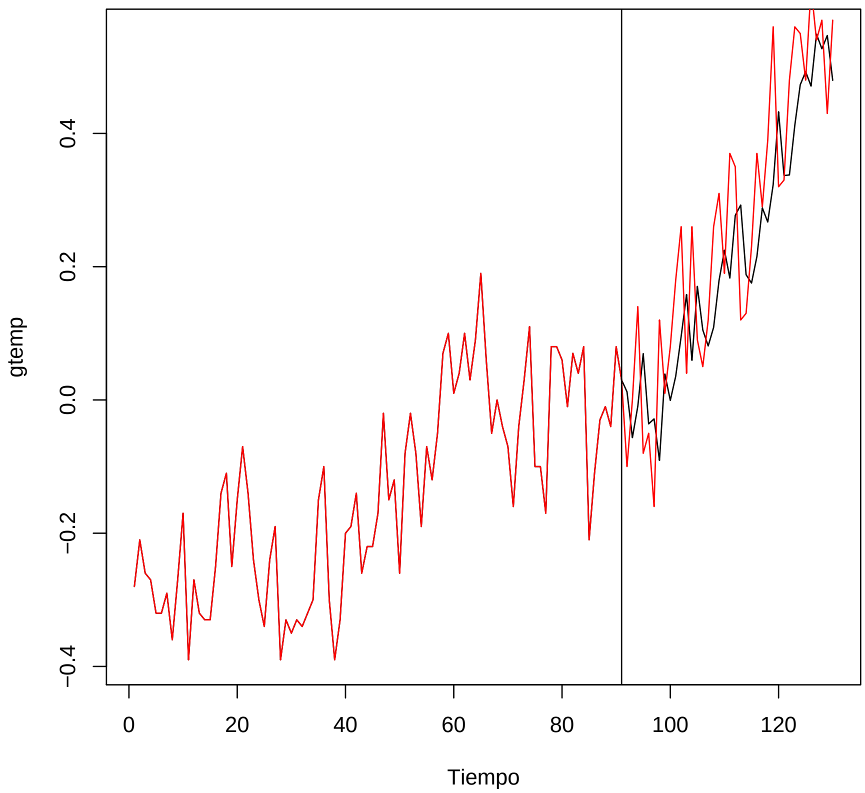
MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```r
mse <- mean((a-gtemp[92:130])^2)
mse

## [1] 0.01442046
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```r
#at=aciertos en tendencia

h <- gtemp[92:130]
d <- c()
for (i in 1:(length(a)-1)) {
  d[i] <- (a[i+1]-a[i])/i #medias para los pronosticos

}
m <- c()
for (i in 1:(length(a)-1)) {
  m[i] <- (h[i+1]-h[i])/i
}
l <- c()
y <- c()
```

```r
x <- d*m

for (i in 1:length(x)) {
  if(x[i]<0){
    y <- c(y,x[i])}

}
 #cantidad de prodcutos negativos que en realidad son los que no aciertan en tendencia

diferencia <- length(gtemp)-length(gtemp[92:130])
# se miraran los 72 pronosticos ya que los datos se partieron en un 70-30
l <- diferencia-length(y)
#cantidad de productos positivos osea los que si
#aciertan en tendencia
porcentajeaciertotendencia <- l*100/diferencia
porcentajeaciertotendencia

## [1] 69.23077

a <- c(gtemp[1:91],a)
ts.plot(a,ylab="gtemp",xlab="Tiempo")
lines(as.numeric(gtemp), col=2)
abline(v=91)
```




Modelopro

MAE del modelo propuesto bajo las $ACF$ y $PACF$:

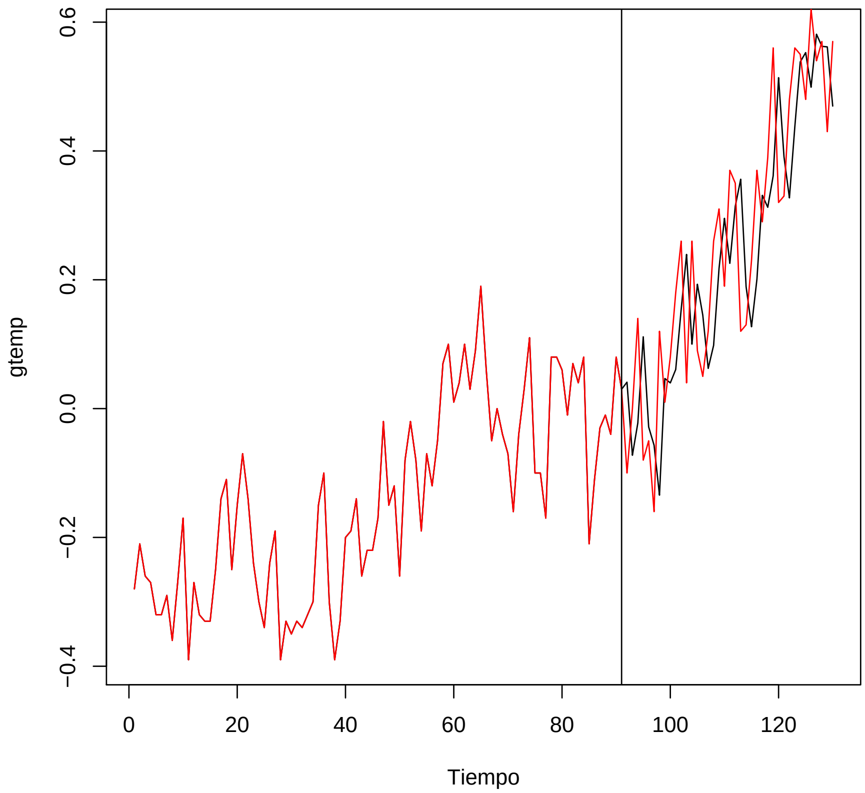
```
## [1] 0.1103908
```

MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 0.0159821
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 70.32967
```




Veamos el modelopro1

MAE del modelo propuesto bajo las $ACF$ y $PACF$:

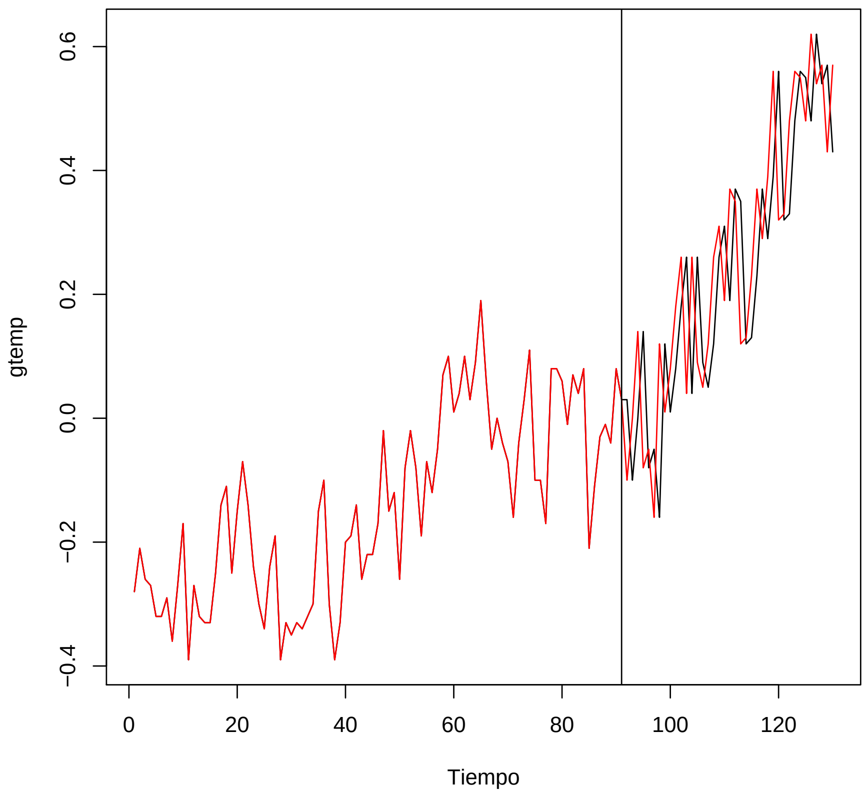
```
## [1] 0.1158974
```

MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 0.01812308
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 72.52747
```




modelopro2

MAE del modelo propuesto bajo las $ACF$ y $PACF$:

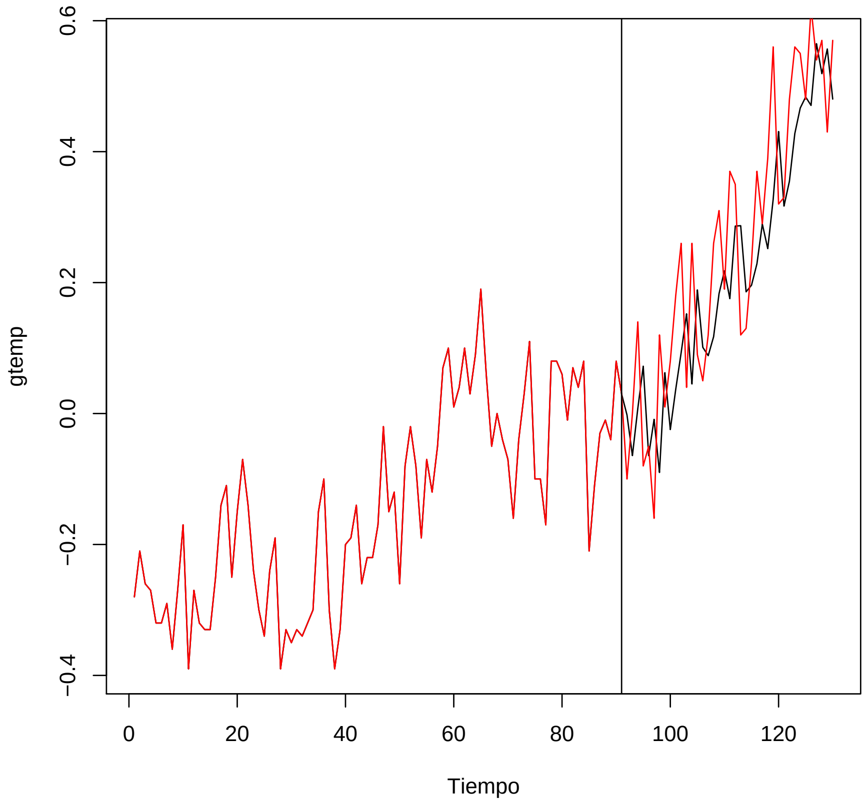
```
## [1] 0.1034154
```

MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 0.01442308
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 70.32967
```




modelopro3:

MAE del modelo propuesto bajo las $ACF$ y $PACF$:

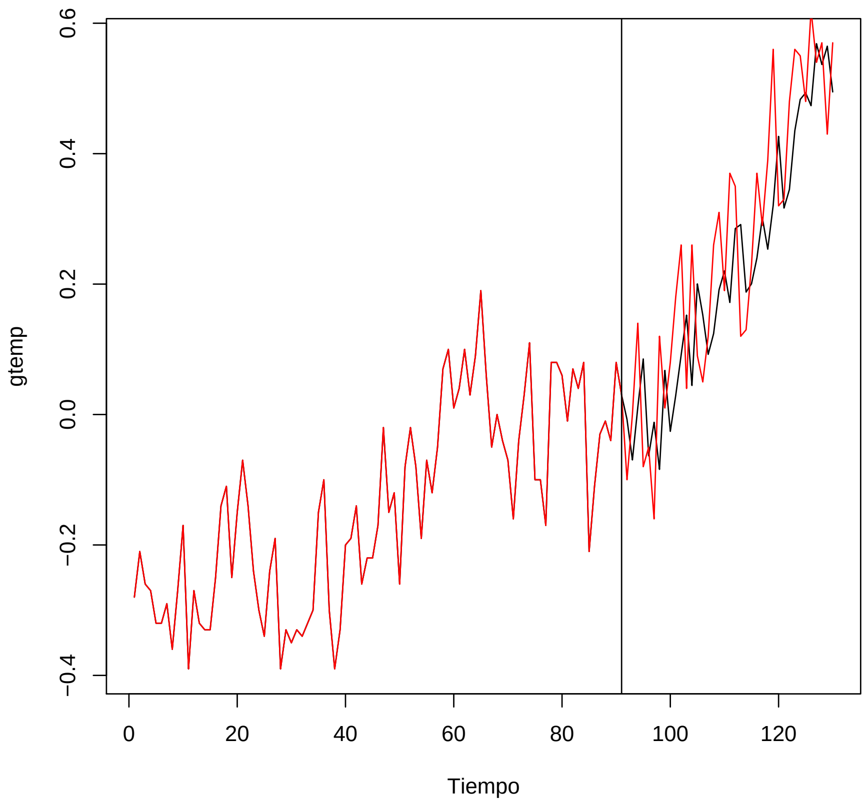
```
## [1] 0.1044964
```

MSE del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 0.01461293
```

Porcentaje de acierto en tendencia del modelo propuesto bajo las $ACF$ y $PACF$:

```
## [1] 70.32967
```




Debido a que el modelo $modelopro3$ suma mas cosas a su favor , hablando en contexto de que cumple con la significancia de parametros junto con su respectivo analisis de residuales junto con un buen $AIC$ y $\sigma^2$ estimado aceptable.

$$Modelofinal : modelopro3 = arima(gtemp, c(2, 1, 2)$$

```
h <- length(gtemp)
d <- length(gtemp)
x<- h-1
y <- h+10
a <- c()
while(d >x & d<y){
  s1 <- ts(gtemp[1:d],start=1,end=d,frequency=1)
  mod1=arima(s1,order=c(2,1,2))
  fore1=forecast(mod1, h=1)
```

```
  x <- fore1$mean[1]
  a <- c(a,x)
  d=d+1
}
a

##  [1] 0.5669683 0.5574338 0.5500418 0.5467524 0.5463780 0.5471791 0.5480197
##  [8] 0.5484904 0.5486187 0.5485705
```

vector de pronosticos un paso a la vez actualizando la serie en cada pronostico

```
h <- length(gtemp)
d <- length(gtemp)
x<- h-1
y <- h+10
a <- c()
while(d >x & d<y){
  s1 <- ts(gtemp[1:d],start=1,end=d,frequency=1)
  mod1=arima(s1,order=c(2,1,2))
  fore1=forecast(mod1, h=2)
  x <- fore1$mean[1]
  a <- c(a,x)
  d=d+2
}
a

## [1] 0.5669683 0.5500418 0.5463780 0.5480197 0.5486187
```

vector de pronosticos dos pasos a la vez actualizando la serie en cada pronostico