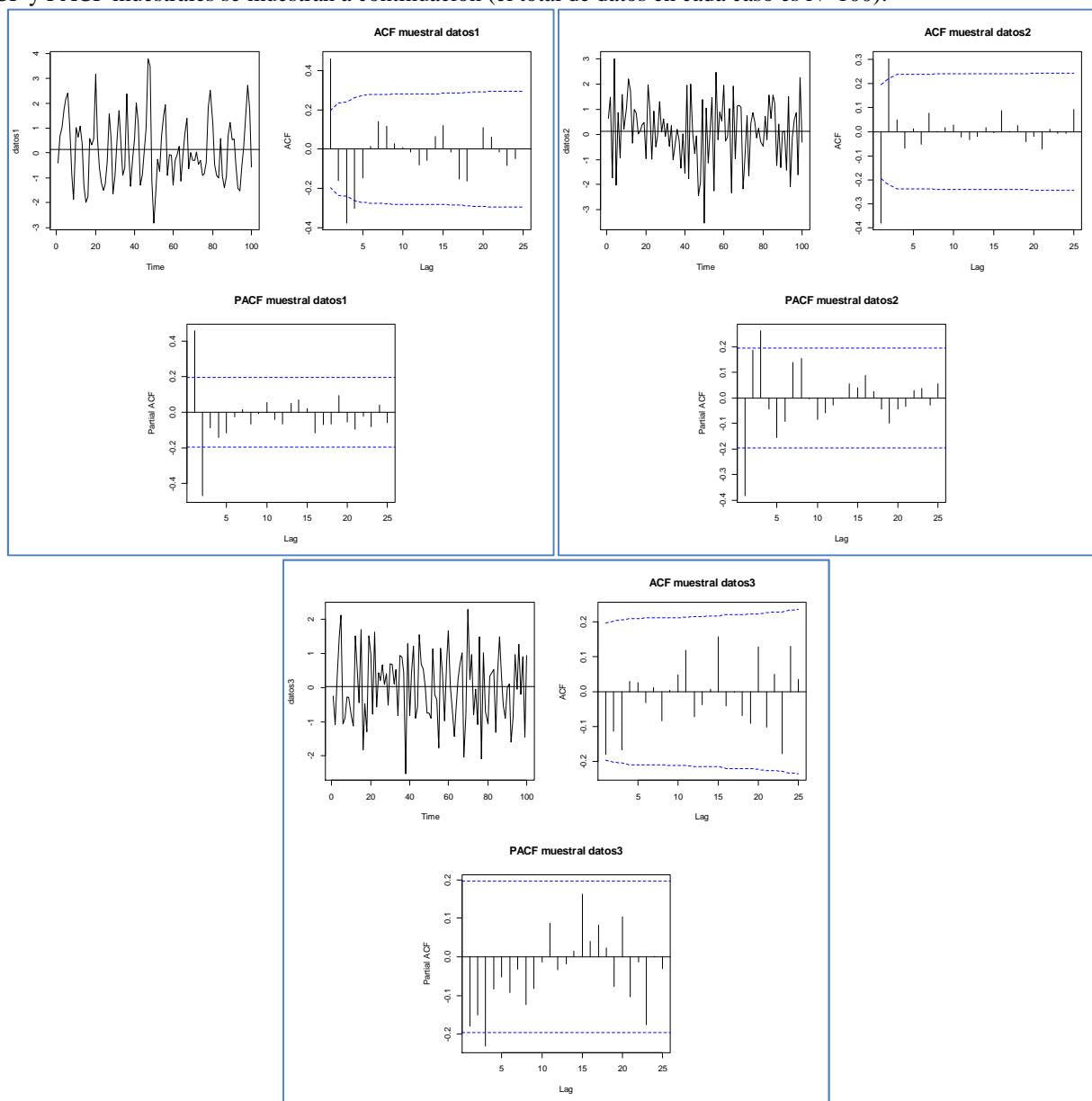


## Taller 8 Monitoría: Identificación, ajustes y pronósticos de modelos ARMA(p,q) de media cero

Considere las series simuladas de procesos estacionarios de media cero, cuyos datos están en los archivos AR2SIMUL.txt (que llamaremos datos1), MA2SIMULb.txt (que llamaremos datos2) y ARMA1.1SIMUL.txt (que llamaremos datos3), cuyas gráficas junto con sus ACF y PACF muestrales se muestran a continuación (el total de datos en cada caso es  $N=100$ ):



- Denote por  $Z_t$ ,  $W_t$ ,  $X_t$  a las variables asociadas a las series datos1, datos2 y datos3, respectivamente, y con base en estas gráficas, para cada serie de datos determine si el proceso del que provienen es un ruido blanco y en caso de no serlo, identifique si es estacionario y cuál modelo ARMA(p,q) (dé las ecuaciones en cada caso).
- Usando métodos automáticos y considerando el patrón observado en las ACF, identifique modelos ARMA para cada una de las tres series. Dé las ecuaciones de tales modelos identificados
  - Con criterio AIC y BIC en función R `auto.arima()`
  - Con la EACF, usando máximo p y q de 16
- Para cada serie ajuste modelos ARMA (sin validación cruzada) identificados y realice además lo siguiente:
  - Dé las ecuaciones ajustadas y de pronóstico. Calcule AIC y BIC como  $\exp(C^*(p))$
  - Analice residuos y valide supuestos sobre errores de ajuste (plantee claramente los elementos de cada prueba: hipótesis, estadísticos de prueba, criterio de decisión y conclusión)
  - Decida cuál modelo es mejor para cada serie.
  - Con mejor modelo en cada serie, pronostique para  $L=2$  períodos después de  $t=N=100$  (ver ANEXO 1)

```
R Console (64-bit)
Archivo Editar Misc Paquetes Ventanas Ayuda

> auto.arima(datos1,ic="aic")
Series: datos1
ARIMA(2,0,2) with zero mean

Coefficients:
      ar1      ar2      ma1      ma2
    1.0353 -0.5159 -0.4259 -0.2260
s.e.  0.1397  0.1293  0.1585  0.1525

sigma^2 estimated as 1.007: log likelihood=-140.72
AIC=291.45 AICc=292.09 BIC=304.47
> auto.arima(datos1,ic="bic")
Series: datos1
ARIMA(2,0,0) with zero mean

Coefficients:
      ar1      ar2
    0.6919 -0.4795
s.e.  0.0881  0.0881

sigma^2 estimated as 1.036: log likelihood=-143.06
AIC=292.12 AICc=292.37 BIC=299.93
> |
```

```
R Console (64-bit)
Archivo Editar Misc Paquetes Ventanas Ayuda

> auto.arima(datos2,ic="aic")
Series: datos2
ARIMA(0,0,2) with zero mean

Coefficients:
      ma1      ma2
    -0.3941  0.5650
s.e.  0.0815  0.0921

sigma^2 estimated as 1.183: log likelihood=-149.72
AIC=305.45 AICc=305.7 BIC=313.26
> auto.arima(datos2,ic="bic")
Series: datos2
ARIMA(0,0,2) with zero mean

Coefficients:
      ma1      ma2
    -0.3941  0.5650
s.e.  0.0815  0.0921

sigma^2 estimated as 1.183: log likelihood=-149.72
AIC=305.45 AICc=305.7 BIC=313.26
> |
```

```
R Console (64-bit)
Archivo Editar Misc Paquetes Ventanas Ayuda

> auto.arima(datos3,ic="aic")
Series: datos3
ARIMA(1,0,1) with zero mean

Coefficients:
      ar1      ma1
    0.4897 -0.7917
s.e.  0.1445  0.0934

sigma^2 estimated as 0.9378: log likelihood=-137.81
AIC=281.62 AICc=281.87 BIC=289.44
> auto.arima(datos3,ic="bic")
Series: datos3
ARIMA(0,0,0) with zero mean

sigma^2 estimated as 1.032: log likelihood=-143.46
AIC=288.92 AICc=288.96 BIC=291.53
> |
```

```
R Console (64-bit)
Archivo Editar Misc Paquetes Ventanas Ayuda

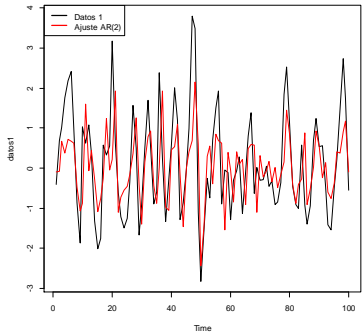
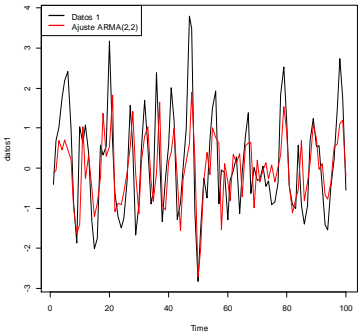
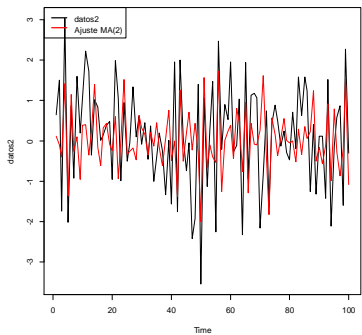
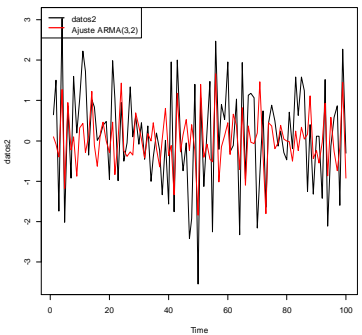
> eacf(datos1,ar.max=16,ma.max=16) #EACF
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
0 x o x x o o o o o o o o o o o o o
1 x x x o o o o o o o o o o o o o
2 o x o o o o o o o o o o o o o
3 x o o o o o o o o o o o o o o
4 x x o o o o o o o o o o o o o
5 x x o x o o o o o o o o o o o
6 o o x o o o o o o o o o o o o
7 o x x o o o o o o o o o o o o
8 o x x o o o o o o o o o o o o
9 o x o x o x o o o o o o o o o
10 x o o x o o o o o o o o o o o
11 x x o o x o o o o o o o o o o
12 x x x o o o o o o o o o o o o
13 x x x x o o o o o o o o o o o
14 x x x x o o x o o o o o o o o
15 x x x x o o x o o o o o o o o
16 x o o o o o o o o o o o o o o
```

```
R Console (64-bit)
Archivo Editar Misc Paquetes Ventanas Ayuda

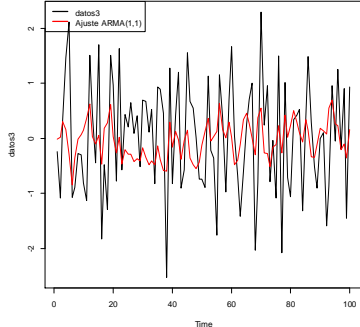
> eacf(datos2,ar.max=16,ma.max=16) #EACF
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
0 x x o o o o o o o o o o o o o
1 x x x o o o o o o o o o o o o
2 x x o x o o o o o o o o o o o
3 x x o o o o o o o o o o o o o
4 x x x o o o o o o o o o o o o
5 x x x x o o o o o o o o o o o
6 x x x x x o o o o o o o o o o
7 x x o o o o o o o o o o o o o
8 o x o o o o o o o o o o o o o
9 o x o x o o o o o o o o o o o
10 x o o x o o o o o o o o o o o
11 x o o x o o o o o o o o o o o
12 x o o o o o o o o o o o o o o
13 o x o o o o o o o o o o o o o
14 x x o o o o o o o o o o o o o
15 x x x x o o x o o o o o o o o
16 x x x x o o o o o o o o o o o
```

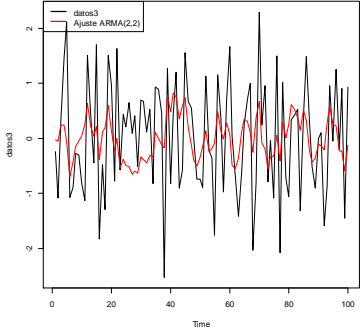
```
R Console (64-bit)
Archivo Editar Misc Paquetes Ventanas Ayuda

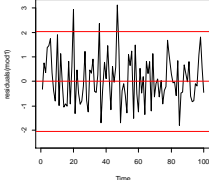
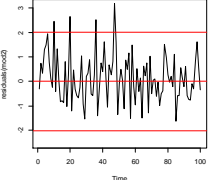
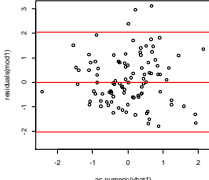
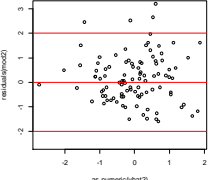
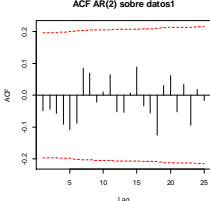
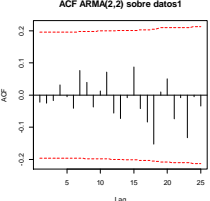
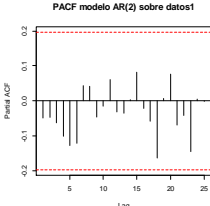
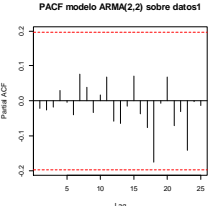
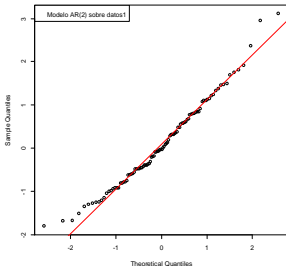
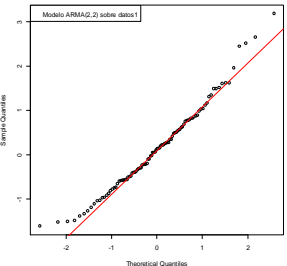
> eacf(datos3,ar.max=16,ma.max=16) #EACF
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
0 o o o o o o o o o o o o o o o
1 x o o o o o o o o o o o o o o
2 x x o o o o o o o o o o o o o
3 x o x o o o o o o o o o o o o
4 x o x o o o o o o o o o o o o
5 x x o o x o o o o o o o o o o
6 x x o o o o o o o o o o o o o
7 x x o o o x o o o o o o o o o
8 x o o x x o o o o o o o o o o
9 o o x x x o o o o o o o o o o
10 o x x o x o o o o o o o o o o
11 x o x o o o o o o o o o o o o
12 x o o o o o o o o o o x o o o
13 x o o x o o o o o o o x o o o
14 o o o x o o o o o o o x o o o
15 x x o x o o o x o o o o o o o
16 x x o x o o o x x o o o o o o
```

|   |   |
|---|---|
| <b>Ajuste AR(2) para datos1</b><br>t test of coefficients:<br>Estimate Std. Error t value Pr(> t )<br>ar1 0.691882 0.088101 7.8532 5.210e-12 ***<br>ar2 -0.479471 0.088111 -5.4416 3.895e-07 ***<br>---<br>Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'<br>0.1 ' ' 1 | <b>Ajuste ARMA(2,2) para datos1</b><br>t test of coefficients:<br>Estimate Std. Error t value Pr(> t )<br>ar1 1.03527 0.13971 7.4100 4.927e-11 ***<br>ar2 -0.51592 0.12931 -3.9897 0.0001293 ***<br>ma1 -0.42592 0.15849 -2.6874 0.0084886 **<br>ma2 -0.22599 0.15249 -1.4820 0.1416216<br>---<br>Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'<br>0.1 ' ' 1                                      |
| <b>Criterios de información en modelos para datos1</b><br>AIC      BIC<br>AR(2)          1.057195 1.113739<br>ARMA(2,2) 1.047466 1.162509   |   |
|    |   |
| <b>Ajuste MA(2) para datos2</b><br>t test of coefficients:<br>Estimate Std. Error t value Pr(> t )<br>ma1 -0.394121 0.081488 -4.8366 4.908e-06 ***<br>ma2 0.565021 0.092079 6.1363 1.794e-08 ***<br>---<br>Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'<br>0.1 ' ' 1 | <b>Ajuste ARMA(3,2) para datos2</b><br>t test of coefficients:<br>Estimate Std. Error t value Pr(> t )<br>ar1 0.086058 0.232457 0.3702 0.71205<br>ar2 -0.147569 0.163262 -0.9039 0.36835<br>ar3 -0.022441 0.159468 -0.1407 0.88839<br>ma1 -0.446300 0.208903 -2.1364 0.03522 *<br>ma2 0.675202 0.140913 4.7916 6.094e-06 ***<br>---<br>Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'<br>0.1 ' ' 1 |
| <b>Criterios de información en modelos para datos2</b><br>AIC      BIC<br>MA(2)          1.207119 1.271681<br>ARMA(3,2) 1.267583 1.443933   |   |
|    |   |

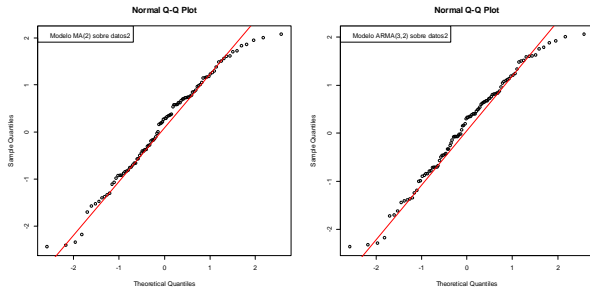
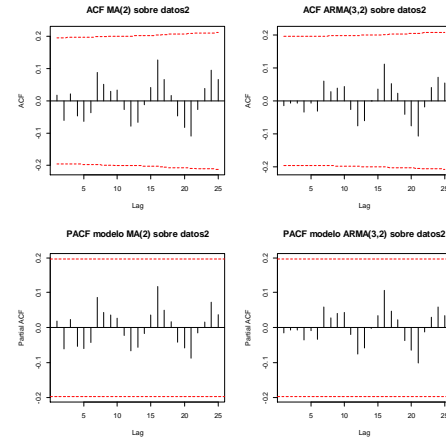
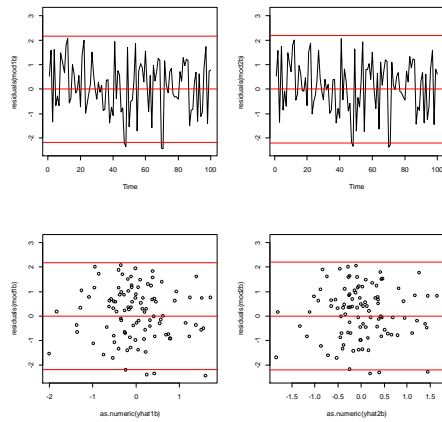
|   |  |
|---|--|
| <b>Ajuste ARMA(1,1) para datos3</b><br>t test of coefficients:<br>Estimate Std. Error t value Pr(> t )<br>ar1 0.489672 0.144539 3.3878 0.001016 **<br>ma1 -0.791686 0.093418 -8.4747 2.437e-13 ***<br>---<br>Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'<br>0.1 ' ' 1 | <b>Ajuste ARMA(2,2) para datos3</b><br>t test of coefficients:<br>Estimate Std. Error t value Pr(> t )<br>ar1 1.611587 0.077851 20.701 < 2.2e-16 ***<br>ar2 -0.685478 0.078484 -8.734 7.772e-14 ***<br>ma1 -1.948239 0.058234 -33.456 < 2.2e-16 ***<br>ma2 0.999906 0.059441 16.822 < 2.2e-16 ***<br>---<br>Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'<br>0.1 ' ' 1 |
| <b>Criterios de información en modelos para datos3</b><br>AIC BIC<br>ARMA(1,1) 0.9565337 1.007694<br>ARMA(2,2) 0.9204913 1.021589   |  |





| Resultados para validación supuestos modelos con datos1   |  |
|---|--|
|                                 |                                      |
|   <p>Shapiro-Wilk normality test<br/> data: residuals(mod1)<br/> W = 0.9785, p-value = 0.1015</p> <p>Shapiro-Wilk normality test<br/> data: residuals(mod2)<br/> W = 0.97529, p-value = 0.05682</p> | <pre>&gt; BP.LB.test(residuals(mod1),maxlag=25,type="Ljung") X.squared df p.value 6 3.812449 6 0.7020397 12 5.989751 12 0.9165978 18 9.743006 18 0.9399721 24 12.065896 24 0.9791555  &gt; BP.LB.test(residuals(mod2),maxlag=25,type="Ljung") X.squared df p.value 6 0.4210547 6 0.9986709 12 2.3801353 12 0.9985608 18 7.9018053 18 0.9800626 24 11.2835962 24 0.9868417 &gt;</pre> |

## Resultados para validación supuestos modelos con datos2



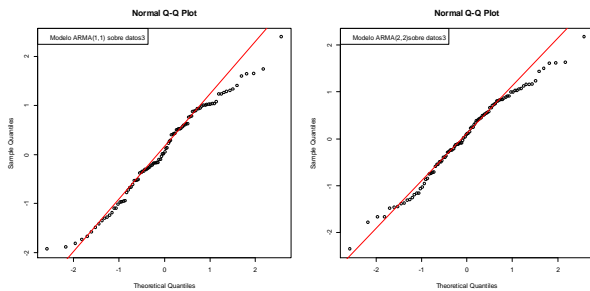
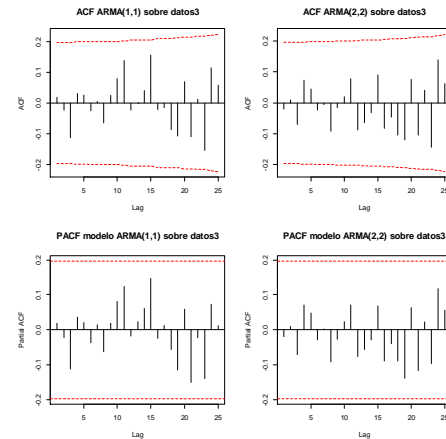
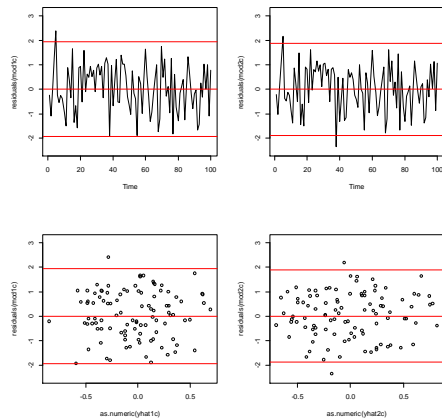
Shapiro-Wilk normality test  
data: residuals(mod1b)  
W = 0.97876, p-value = 0.1064

Shapiro-Wilk normality test  
data: residuals(mod2b)  
W = 0.9788, p-value = 0.1072

```
> BP.LB.test(residuals(mod1b),maxlag=25,type="Ljung")
      X.squared df    p.value
6      1.315446  6  0.9707991
12     3.527035 12  0.9905411
18     6.846811 18  0.9913558
24    11.000311 24  0.9890096
```

```
> BP.LB.test(residuals(mod2b),maxlag=25,type="Ljung")
      X.squared df    p.value
6      0.276793  6  0.9996016
12     1.892262 12  0.9995540
18     4.384403 18  0.9995421
24     7.748421 24  0.9993017
```

## Resultados para validación supuestos modelos con datos3



Shapiro-Wilk normality test  
data: residuals(mod1c)  
W = 0.98056, p-value = 0.147

Shapiro-Wilk normality test  
data: residuals(mod2c)  
W = 0.98408, p-value = 0.2719

```
> BP.LB.test(residuals(mod1c),maxlag=25,type="Ljung")
      X.squared df    p.value
6      1.657414  6  0.9483614
12     5.101319 12  0.9545089
18     9.201584 18  0.9548884
24    17.690512 24  0.8177585
```

```
> BP.LB.test(residuals(mod2c),maxlag=25,type="Ljung")
      X.squared df    p.value
6      1.409258  6  0.9652928
12     4.010927 12  0.9832384
18     7.963268 18  0.9791783
24    17.374201 24  0.8322697
```

| ANEXO 1  |            |            |  |            |           |
|--|------------|------------|--|------------|-----------|
| Algunos resultados con modelo AR(2) sobre datos1     |            |            | Algunos resultados con modelo MA(2) sobre datos2 |            |           |
|  | residuales | obs.       |  | residuales | obs.      |
| t=98   | 1.8114352  | 2.7287790  | t=98   | -1.4001133 | -1.599946 |
| t=99   | 0.5658022  | 1.7471550  | t=99   | 0.7363920  | 2.260711  |
| t=100  | -0.4490142 | -0.5485614 | t=100  | 0.7759357  | -0.305386 |
| Algunos resultados con modelo ARMA(1,1) sobre datos3 |            |            |  |            |           |
|  | residuales | obs        |  |            |           |
| t=98   | 1.0018306  | 0.8990827  |  |            |           |
| t=99   | -1.0947155 | -1.4475950 |  |            |           |
| t=100  | 0.7805878  | 0.9384119  |  |            |           |

Programa R usado:

I. Cargar librerías y definir funciones de usuario

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

#Creando función usuario para obtener test Box-Pierce y Ljung-Box
BP.LB.test=function(serie,maxlag,type){
  aux=floor(maxlag/6)
  X.squared=c(rep(NA,aux))
  df=c(rep(NA,aux))
  p.value=c(rep(NA,aux))
  for(i in 1:aux){
    test=Box.test(serie,lag=(6*i),type=type)
    X.squared[i]=test[[1]]
    df[i]=test[[2]]
    p.value[i]=test[[3]]
  }
  lag=6*c(1:aux)
  teste=as.data.frame(cbind(X.squared,df,p.value))
  rownames(teste)=lag
  teste
}

#Creando función usuario crit.inf.resid() para calcular  $C_n^*(p)$ 
crit.inf.resid=function(residuales,n.par,AIC="TRUE"){
  if(AIC=="TRUE"){
    #Calcula AIC
    CI=log(mean(residuales^2))+2*n.par/length(residuales)
  }
  if(AIC=="FALSE"){
    #Calcula BIC
    CI=log(mean(residuales^2))+n.par*log(length(residuales)/length(residuales))
  }
  CI
}
```

II. Programación con serie datos1

```
#datos1
#Leer AR2SIMUL.txt
datos1=scan(file.choose())
datos1=ts(datos1,freq=1)

layout(rbind(c(1,1,2,2),c(0,3,3,0)))
plot(datos1)
abline(h=mean(datos1))
acf(datos1,ci.type="ma",lag.max=25,main="ACF muestral datos1")
pacf(datos1,lag.max=25,main="PACF muestral datos1")

#Identificación modelos sobre datos1
eacf(datos1,ar.max=16,ma.max=16) #EACF
auto.arima(datos1,ic="aic")
auto.arima(datos1,ic="bic")

###Modelos sobre datos1
#Modelo AR(2) con media cero
mod1=Arima(datos1,order=c(2,0,0),include.mean=F,method="ML")
df1=length(datos1)-2 #Número parámetros es p=2
coeftest(mod1,df=df1)

yhat1=mod1$fitted
```

#### **#ARMA(2,2) con media cero**

```
mod2=Arima(datos1,order=c(2,0,2),include.mean=F,method="ML")
df2=length(datos1)-4 #Número parámetros es p=4
coeftest(mod2,df=df2)
```

```
yhat2=mod2$fitted
```

#### **#Medidas de ajuste**

```
aic1=exp(crit.inf.resid(residuals(mod1),n.par=2))
aic2=exp(crit.inf.resid(residuals(mod2),n.par=4))
bic1=exp(crit.inf.resid(residuals(mod1),n.par=2,AIC="FALSE"))
bic2=exp(crit.inf.resid(residuals(mod2),n.par=4,AIC="FALSE"))
```

```
criterios=data.frame(AIC=c(aic1,aic2),BIC=c(bic1,bic2),row.names=c("AR(2)","ARMA(2,2)"))
criterios
```

#### **#Gráficos de los ajustes**

```
win.graph(width=10,height=5)
layout(matrix(c(1,1,2,2),ncol=4))
plot(datos1)
lines(yhat1,col=2)
legend("topleft",legend=c("Datos 1","Ajuste AR(2)"),col=1:2,lwd=2)

plot(datos1)
lines(yhat2,col=2)
legend("topleft",legend=c("Datos 1","Ajuste ARMA(2,2)"),col=1:2,lwd=2)
```

#### **#Gráficos residuos**

```
win.graph()
layout(rbind(c(1,1,2,2),c(3,3,4,4)))
plot(residuals(mod1),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod1$sigma2),0,2*sqrt(mod1$sigma2)),col=2)

plot(residuals(mod2),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod2$sigma2),0,2*sqrt(mod2$sigma2)),col=2)

plot(as.numeric(yhat1),residuals(mod1),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod1$sigma2),0,2*sqrt(mod1$sigma2)),col=2)

plot(as.numeric(yhat2),residuals(mod2),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod2$sigma2),0,2*sqrt(mod2$sigma2)),col=2)
```

#### **#ACF y PACF residuales**

```
win.graph()
layout(rbind(c(1,1,2,2),c(3,3,4,4)))
acf(as.numeric(residuals(mod1)),ci.type="ma",lag.max=25,main="ACF AR(2) sobre datos1",ci.col=2)
acf(as.numeric(residuals(mod2)),ci.type="ma",lag.max=25,main="ACF ARMA(2,2) sobre datos1",ci.col=2)
pacf(as.numeric(residuals(mod1)),lag.max=25,main="PACF modelo AR(2) sobre datos1",ci.col=2)
pacf(as.numeric(residuals(mod2)),lag.max=25,main="PACF modelo ARMA(2,2) sobre datos1",ci.col=2)
```

#### **#Tests Ljun-Box**

```
BP.LB.test(residuals(mod1),maxlag=25,type="Ljung")
BP.LB.test(residuals(mod2),maxlag=25,type="Ljung")
```

#### **#tests de normalidad**

```
shapiro.test(residuals(mod1))
shapiro.test(residuals(mod2))
```

#### **#Gráficos de probabilidad normal**

```
win.graph(width=10,height=5)
layout(matrix(c(1,1,2,2),ncol=4))
qqnorm(residuals(mod1));qqline(residuals(mod1),col=2)
legend("topleft",legend="Modelo AR(2) sobre datos1")
qqnorm(residuals(mod2));qqline(residuals(mod2),col=2)
legend("topleft",legend="Modelo ARMA(2,2) sobre datos1")
```

### **III. Programación con serie datos2**

#### **#datos2**

##### **#Leer MA2SIMULb.txt**

```
datos2=scan(file.choose())
datos2=ts(datos2,freq=1)
```

```
win.graph()
layout(rbind(c(1,1,2,2),c(0,3,3,0)))
plot(datos2)
abline(h=mean(datos2))
acf(datos2,ci.type="ma",lag.max=25,main="ACF muestral datos2")
pacf(datos2,lag.max=25,main="PACF muestral datos2")
```

## **#Identificación con algunas funciones R de modelos sobre datos2**

```
eacf(datos2,ar.max=16,ma.max=16) #EACF
auto.arima(datos2,ic="aic")
auto.arima(datos2,ic="bic")
```

## **##Modelos sobre datos2**

### **#Modelo MA(2) con media cero**

```
mod1b=Arima(datos2,order=c(0,0,2),include.mean=F,method="ML")
df1b=length(datos2)-2 #Número parámetros es p=2
coeftest(mod1b,df=df1b)
```

```
yhat1b=mod1b$fitted
```

### **#ARMA(3,2) con media cero**

```
mod2b=Arima(datos2,order=c(3,0,2),include.mean=F,method="ML")
df2b=length(datos2)-5 #Número parámetros es p=5
coeftest(mod2b,df=df2b)
```

```
yhat2b=mod2b$fitted
```

## **#Medidas de ajuste**

```
aic1b=exp(crit.inf.resid(residuals(mod1b),n.par=2))
aic2b=exp(crit.inf.resid(residuals(mod2b),n.par=5))
bic1b=exp(crit.inf.resid(residuals(mod1b),n.par=2,AIC="FALSE"))
bic2b=exp(crit.inf.resid(residuals(mod2b),n.par=5,AIC="FALSE"))
```

```
criteriosb=data.frame(AIC=c(aic1b,aic2b),BIC=c(bic1b,bic2b),row.names=c("MA(2)","ARMA(3,2)"))
criteriosb
```

## **#Gráficos de los ajustes**

```
win.graph(width=10,height=5)
layout(matrix(c(1,1,2,2),ncol=4))
plot(datos2)
lines(yhat1b,col=2)
legend("topleft",legend=c("datos2","Ajuste MA(2)"),col=1:2,lwd=2)

plot(datos2)
lines(yhat2b,col=2)
legend("topleft",legend=c("datos2","Ajuste ARMA(3,2)"),col=1:2,lwd=2)
```

## **#Gráficos residuos**

```
win.graph()
layout(rbind(c(1,1,2,2),c(3,3,4,4)))
plot(residuals(mod1b),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod1b$sigma2),0,2*sqrt(mod1b$sigma2)),col=2)

plot(residuals(mod2b),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod2b$sigma2),0,2*sqrt(mod2b$sigma2)),col=2)

plot(as.numeric(yhat1b),residuals(mod1b),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod1b$sigma2),0,2*sqrt(mod1b$sigma2)),col=2)

plot(as.numeric(yhat2b),residuals(mod2b),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod2b$sigma2),0,2*sqrt(mod2b$sigma2)),col=2)
```

## **#ACF y PACF residuales**

```
win.graph()
layout(rbind(c(1,1,2,2),c(3,3,4,4)))
acf(as.numeric(residuals(mod1b)),ci.type="ma",lag.max=25,main="ACF MA(2) sobre datos2",ci.col=2)
acf(as.numeric(residuals(mod2b)),ci.type="ma",lag.max=25,main="ACF ARMA(3,2) sobre datos2",ci.col=2)
pacf(as.numeric(residuals(mod1b)),lag.max=25,main="PACF modelo MA(2) sobre datos2",ci.col=2)
pacf(as.numeric(residuals(mod2b)),lag.max=25,main="PACF modelo ARMA(3,2) sobre datos2",ci.col=2)
```

## **#Tests Ljun-Box**

```
BP.LB.test(residuals(mod1b),maxlag=25,type="Ljung")
BP.LB.test(residuals(mod2b),maxlag=25,type="Ljung")
```

## **#tests de normalidad**

```
shapiro.test(residuals(mod1b))
shapiro.test(residuals(mod2b))
```

## **#Gráficos de probabilidad normal**

```
win.graph(width=10,height=5)
layout(matrix(c(1,1,2,2),ncol=4))
qqnorm(residuals(mod1b));qqline(residuals(mod1b),col=2)
legend("topleft",legend="Modelo MA(2) sobre datos2")
qqnorm(residuals(mod2b));qqline(residuals(mod2b),col=2)
legend("topleft",legend="Modelo ARMA(3,2) sobre datos2")
```



## IV. Programación con serie datos3

```
##datos3
#Leer ARMA1.1SIMUL.txt
datos3=scan(file.choose())
datos3=ts(datos3,freq=1)

win.graph()
layout(rbind(c(1,1,2,2),c(0,3,3,0)))
plot(datos3)
abline(h=mean(datos3))
acf(datos3,ci.type="ma",lag.max=25,main="ACF muestral datos3")
pacf(datos3,lag.max=25,main="PACF muestral datos3")

#Identificación con algunas funciones R de modelos sobre datos 3
eacf(datos3,ar.max=16,ma.max=16)
auto.arima(datos3,ic="aic")
auto.arima(datos3,ic="bic")

##modelos sobre datos 3
#Modelo ARMA(1,1) con media cero
mod1c=Arima(datos3,order=c(1,0,1),include.mean=F,method="ML")
df1c=length(datos3)-2 #Número parámetros es p=2
coeftest(mod1c,df=df1c)

yhat1c=mod1c$fitted

#ARMA(2,2) con media cero
mod2c=Arima(datos3,order=c(2,0,2),include.mean=F,method="ML")
df2c=length(datos3)-4 #Número parámetros es p=4
coeftest(mod2c,df=df2c)

yhat2c=mod2c$fitted

#Medidas de ajuste
aic1c=exp(crit.inf.resid(residuals(mod1c),n.par=2))
aic2c=exp(crit.inf.resid(residuals(mod2c),n.par=4))
bic1c=exp(crit.inf.resid(residuals(mod1c),n.par=2,AIC="FALSE"))
bic2c=exp(crit.inf.resid(residuals(mod2c),n.par=4,AIC="FALSE"))

criteriosc=data.frame(AIC=c(aic1c,aic2c),BIC=c(bic1c,bic2c),row.names=c("ARMA(1,1)","ARMA(2,2)"))
criteriosc

#Gráficos de los ajustes
win.graph(width=10,height=5)
layout(matrix(c(1,1,2,2),ncol=4))
plot(datos3)
lines(yhat1c,col=2)
legend("topleft",legend=c("datos3","Ajuste ARMA(1,1)"),col=1:2,lwd=2)

plot(datos3)
lines(yhat2c,col=2)
legend("topleft",legend=c("datos3","Ajuste ARMA(2,2)"),col=1:2,lwd=2)

#Gráficos residuos
win.graph()
layout(rbind(c(1,1,2,2),c(3,3,4,4)))
plot(residuals(mod1c),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod1c$sigma2),0,2*sqrt(mod1c$sigma2)),col=2)

plot(residuals(mod2c),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod2c$sigma2),0,2*sqrt(mod2c$sigma2)),col=2)

plot(as.numeric(yhat1c),residuals(mod1c),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod1c$sigma2),0,2*sqrt(mod1c$sigma2)),col=2)

plot(as.numeric(yhat2c),residuals(mod2c),ylim=c(-2.5,3.1))
abline(h=c(-2*sqrt(mod2c$sigma2),0,2*sqrt(mod2c$sigma2)),col=2)

#ACF y PACF residuales
win.graph()
layout(rbind(c(1,1,2,2),c(3,3,4,4)))
acf(as.numeric(residuals(mod1c)),ci.type="ma",lag.max=25,main="ACF ARMA(1,1) sobre datos3",ci.col=2)
acf(as.numeric(residuals(mod2c)),ci.type="ma",lag.max=25,main="ACF ARMA(2,2) sobre datos3",ci.col=2)
pacf(as.numeric(residuals(mod1c)),lag.max=25,main="PACF modelo ARMA(1,1) sobre datos3",ci.col=2)
pacf(as.numeric(residuals(mod2c)),lag.max=25,main="PACF modelo ARMA(2,2) sobre datos3",ci.col=2)

#Tests Ljung-Box
BP.LB.test(residuals(mod1c),maxlag=25,type="Ljung")
BP.LB.test(residuals(mod2c),maxlag=25,type="Ljung")
```

#### **#tests de normalidad**

```
shapiro.test(residuals(mod1c))  
shapiro.test(residuals(mod2c))
```

#### **#Gráficos de probabilidad normal**

```
win.graph(width=10,height=5)  
layout(matrix(c(1,1,2,2),ncol=4))  
qqnorm(residuals(mod1c));qqline(residuals(mod1c),col=2)  
legend("topleft",legend="Modelo ARMA(1,1) sobre datos3")  
qqnorm(residuals(mod2c));qqline(residuals(mod2c),col=2)  
legend("topleft",legend="Modelo ARMA(2,2)sobre datos3")
```