Series Temporales - Trabajo Final

Esteban Schab - Anabella De Battista

Febrero 2021

Contents

Ejercicio 1 - Modelos ARMA

Se requiere:

a. Identificación del modelo empleando los primeros 140 registros mediante ACF, PACF y AIC.

b. Realizar un pronóstico out-of sample, sobre los últimos 10 datos. Determinar la precisión del pronóstico de los dos mejores modelos calculados en en apartado (a), mediante el error cuadrático medio (mean squared error) y el error absoluto medio (mean absolute error).

Modelo 1

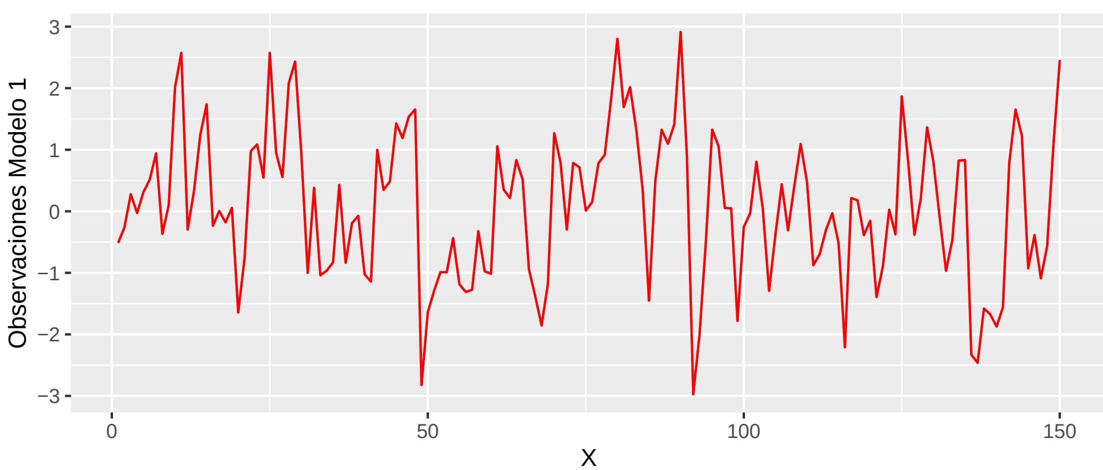
```
modelo1 <- read.csv("./data/modelo1.csv", header = TRUE, sep = ",", quote = "\"",
                    dec = ".", fill = TRUE)
m1_ts <- ts(modelo1[,2])
```

Estadísticas principales

```
summary(m1_ts)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.97293 -0.86605  0.05085  0.07211  0.91071  2.91216
```

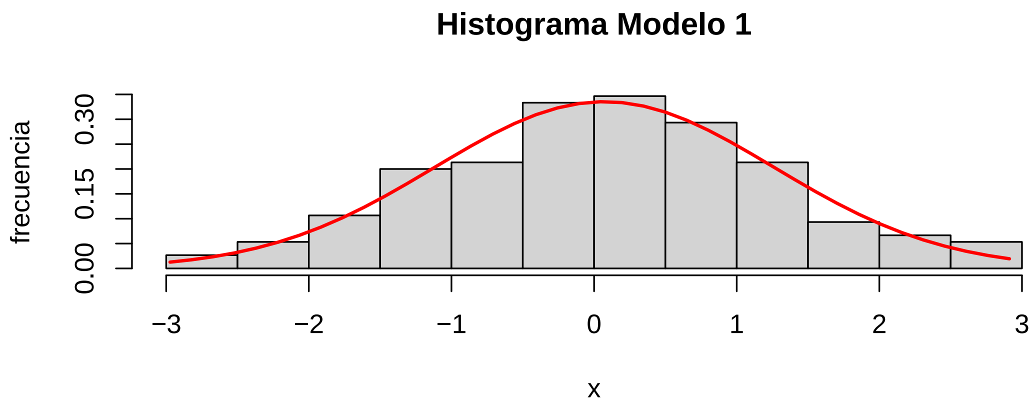
Gráfico de la serie temporal



En base al gráfico la serie aparenta ser constante en media y en varianza, por lo que se podría pensar que es estacionaria.

Histograma

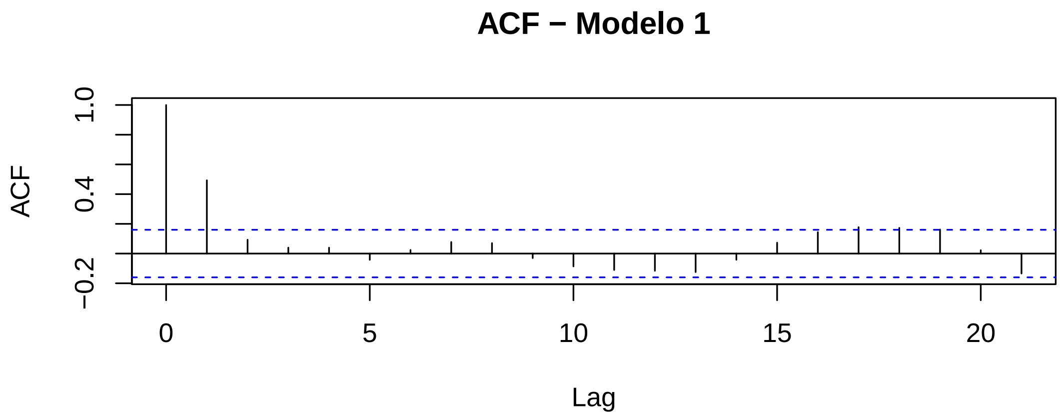
```
hist(m1_ts, prob=TRUE, main = "Histograma Modelo 1", xlab = "x", ylab = "frecuencia")
x <- seq(min(m1_ts), max(m1_ts), length = 40)
f <- dnorm(x, mean = mean(m1_ts), sd = sd(m1_ts))
lines(x, f, col = "red", lwd = 2)
```



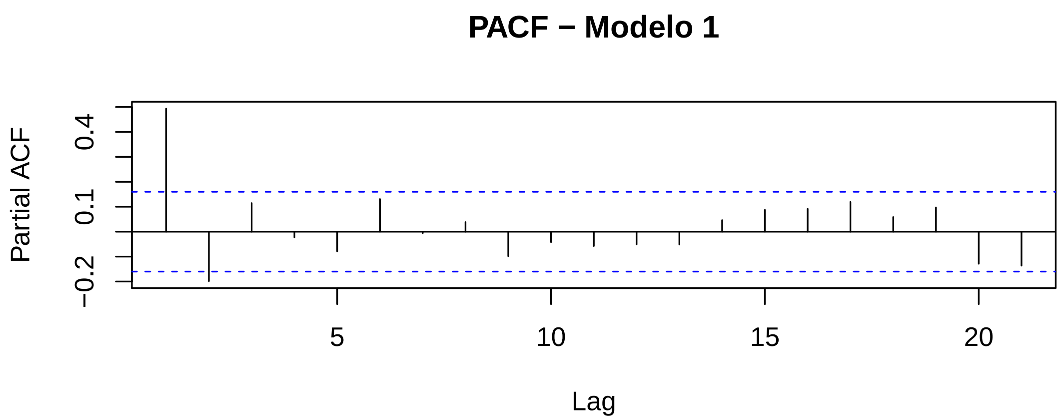
En base al histograma se ve que la distribución de los datos se ajusta a una curva normal.

Gráficas ACF y PACF

```
acf(m1_ts, main = "ACF - Modelo 1")
```



```r
pacf(m1_ts, main = "PACF - Modelo 1")
```



De acuerdo al gráfico de autocorrelación se puede apreciar que la serie es estacionaria ya que la función decae a medida que aumentan los rezagos en el tiempo. No existe componente estacional.

Si se analizan las gráficas ACF y PACF se deduce que es posible plantear varios modelos tentativos para el análisis, realizando combinaciones de p y q entre 0 y 2.

Previamente se descarta que exista componente estacional en la serie de datos.

```r
fit <- tbats(m1_ts)
s <- !is.null(fit$seasonal)
s
```

```
## [1] FALSE
```

El hecho de que **s=FALSE** indica que no hay estacionalidad en la serie de tiempo modelo1.

Verificación de que la serie es estacionaria

Aplicamos el test de estacionariedad ADF (Dickey-Fuller).

Planteamiento de Hipótesis Significancia $\alpha = 0.05$ H0: La serie es no estacionaria: tiene raíz unitaria H1: La serie es estacionaria: no tiene raíz unitaria

```r
library(tseries)
adf<-adf.test(m1_ts)
adf$p.value
```

```
## [1] 0.01
```

Se aplica el test KPSS para verificar si la serie es estacionaria.

```r
kpss<-kpss.test(m1_ts)
kpss$p.value
```

```
## [1] 0.1
```

Como el p-valor del Test de Dickey-Fuller da menor que α y en el caso del test KPSS da mayor que α, concluimos que la serie *es estacionaria*.

Configuración de parámetros para procesos ARMA, variando p y q.

```r
p.maximo <- 2  # orden maximo p
q.maximo <- 2  # orden maximo q
cantidad.modelos <- (p.maximo+1)*(q.maximo+1)

#combinaciones de p y q
orden.modelos <- permutations(p.maximo+1,2,0:p.maximo,repeats.allowed = TRUE)
resumen<- matrix(0,cantidad.modelos,6)  #matriz resumen (loglik, sigma2 y aic)
colnames(resumen) <- c("Loglik","Sigma2","AIC","AIC diff",
                       "Likelihood_model","Akaike weights")
 #nombre de las filas
nombres <- matrix(0,1,cantidad.modelos)
for (modelo.numero in 1:cantidad.modelos){
  nombres[1,modelo.numero] <- paste("p=",as.character(orden.modelos[modelo.numero,1]),
                                    "q=",as.character(orden.modelos[modelo.numero,2]))
}
rownames(resumen) <- nombres
```

Prueba de los modelos con los primeros 140 valores de la serie.

```r
fit_m1_ts <- window(m1_ts, start=1, end=140)

residuos.modelos <- matrix(0,length(fit_m1_ts),cantidad.modelos)
colnames(residuos.modelos)<- nombres
celda <- 0
for(p in 0:p.maximo){
  for(q in 0:q.maximo){
    celda= celda+1
    AR1 <- arima(fit_m1_ts, order = c(p,0,q))
    residuos.modelos[,celda] <- AR1$residuals
    resumen[celda,1] <- AR1$loglik
    resumen[celda,2] <- AR1$sigma2
    resumen[celda,3] <- AR1$aic
  }}
```

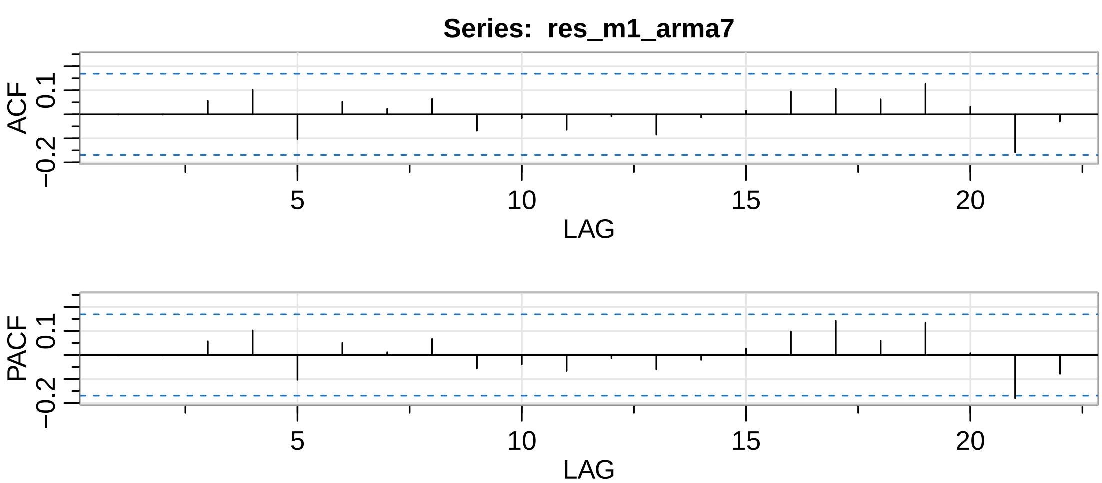
```r
resumen[,4]<-resumen[,3]-min(resumen[,3])
resumen[,5]<-exp(-0.5*resumen[,4])
resumen[,6]<-resumen[,5]/sum(resumen[,5])

# resumen de resultados
round(resumen,digits=4)
```

```
##                Loglik Sigma2      AIC AIC diff Likelihood_model Akaike weights
## p= 0 q= 0 -221.5203 1.3864 447.0407  42.6644           0.0000         0.0000
## p= 0 q= 1 -199.5710 1.0108 405.1420   0.7658           0.6819         0.1578
## p= 0 q= 2 -198.1881 0.9906 404.3763   0.0000           1.0000         0.2315
## p= 1 q= 0 -201.1960 1.0348 408.3919   4.0157           0.1343         0.0311
## p= 1 q= 1 -198.2574 0.9917 404.5147   0.1385           0.9331         0.2160
## p= 1 q= 2 -198.1979 0.9908 406.3958   2.0196           0.3643         0.0843
## p= 2 q= 0 -199.2133 1.0055 406.4265   2.0503           0.3587         0.0830
## p= 2 q= 1 -198.2573 0.9917 406.5147   2.1384           0.3433         0.0795
## p= 2 q= 2 -196.8725 0.9712 405.7450   1.3687           0.5044         0.1168
```

En base a los resultados se aprecia que para el dasaset *modelo1*, los dos mejores modelos son: arma3: ARMA(0,2) cuyo valor para *AIC diff* es = 0.0000 y arma5: ARMA(1,1) cuyo valor para *AIC diff* es = 0.1385.

Para validar el modelo se grafica el correlograma de los residuos para comprobar que son ruido blanco.

```r
# Calculo ACF y PACF para el mejor modelo ARMA para el dataset modelo1
res_m1_arma7 <- residuos.modelos[,3]
acf2(res_m1_arma7)
```




```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF      0    0 0.06  0.1 -0.1 0.05 0.02 0.06 -0.07 -0.02 -0.06 -0.01 -0.08
## PACF     0    0 0.06  0.1 -0.1 0.05 0.01 0.07 -0.06 -0.04 -0.07 -0.01 -0.06
```

```
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## ACF  -0.01  0.01   0.1  0.11  0.06  0.13  0.03 -0.16 -0.03
## PACF -0.02  0.03   0.1  0.14  0.06  0.13  0.01 -0.18 -0.08
```

En los correlogramas se aprecia que no hay ningún rezago significativo por lo que se puede decir que los residuos son ruido blanco.

Prueba con la función auto.arima

La función auto.arima permite determinar el mejor modelo ARIMA para un conjunto de datos. Al ejecutarla en este caso coinicide con el modelo identificado previamente.

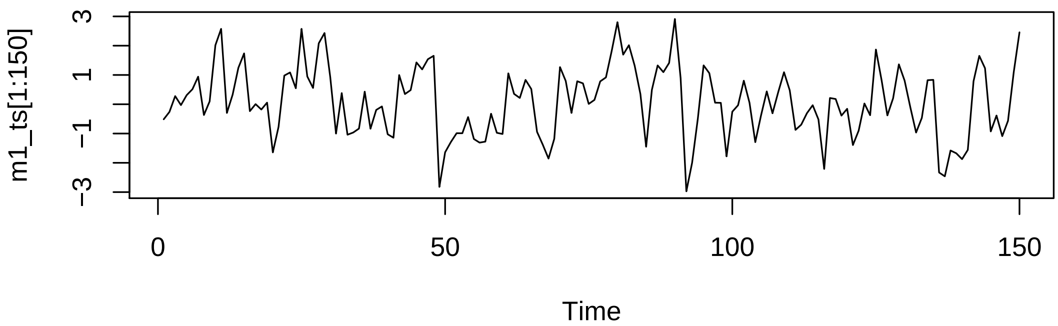
```
auto.arima(fit_m1_ts, stepwise = FALSE, approximation = FALSE)
```

```
## Series: fit_m1_ts
## ARIMA(0,0,2) with zero mean
##
## Coefficients:
##          ma1     ma2
##       0.6250  0.1314
## s.e.  0.0876  0.0782
##
## sigma^2 estimated as 1.006:  log likelihood=-198.24
## AIC=402.49   AICc=402.66   BIC=411.31
```

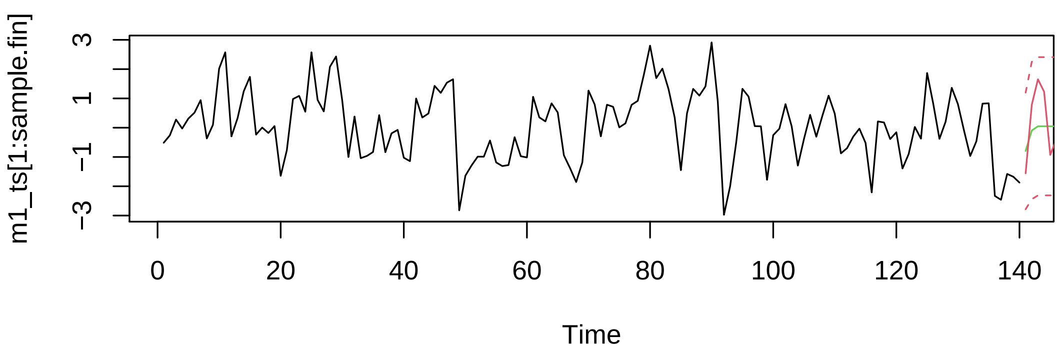
Por lo tanto estamos en condiciones de realizar un pronóstico para las siguientes 10 observaciones de la serie.

Pronóstico para Modelo 1 con ARMA(0,2)

```
#parametros p y q del mejor modelo
p <- 0
q <- 2
#fiteo el modelo con 140 datos (para despues hacer el out-of-sample forecast)
sample.fin <- 140
AR1_02 <- arima(m1_ts[1:sample.fin], order = c(p,0,q))
ts.plot(m1_ts[1:150])
```



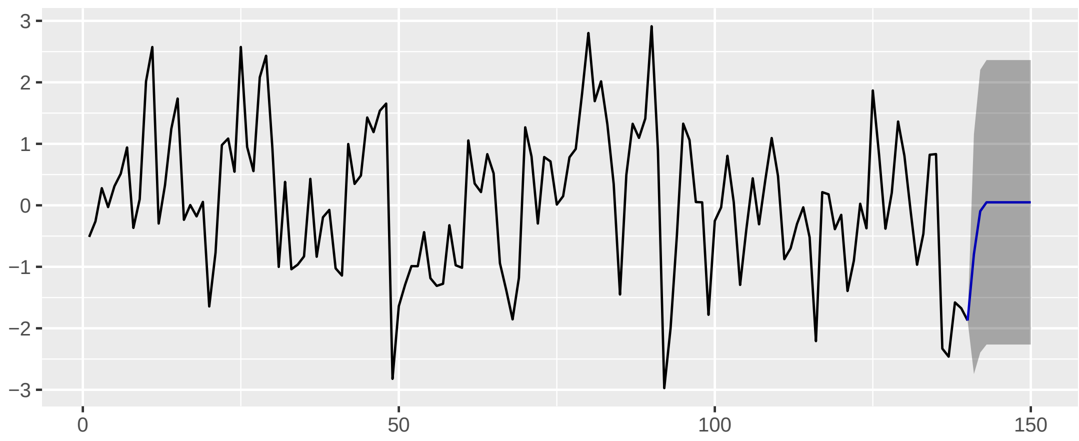

```
#plotting the m1_ts series plus the forecast and 95% prediction intervals
ts.plot(m1_ts[1:sample.fin], xlim = c(1, sample.fin))
AR1_02_forecast <- predict(AR1_02, n.ahead = length(m1_ts)-sample.fin)$pred
AR1_02_forecast_se <- predict(AR1_02, n.ahead = length(m1_ts)-sample.fin)$se
points(AR1_02_forecast, type = "l", col = 3)
points(AR1_02_forecast - 2*AR1_02_forecast_se, type = "l", col = 2, lty = 2)
points(AR1_02_forecast + 2*AR1_02_forecast_se, type = "l", col = 2, lty = 2)
points(sample.fin+1:150,m1_ts[sample.fin+1:150], type = "l", col = 2)
```



Otro método para pronóstico con ARMA(0,2)

```
library(forecast)
arima1_3<- arima(m1_ts[1:sample.fin], order = c(0,0,2))
```

```
forecast3<-forecast(arima1_3, level = c(95), h = 10)
autoplot(forecast3)
```



Comprobar los errores de los pronósticos con ARMA(2,0)

```
error(forecast=AR1_02_forecast, true=m1_ts[141:150], method=c("mse"))
```

```
## [1] 1.503304
```

```
error(forecast=AR1_02_forecast, true=m1_ts[141:150], method=c("mae"))
```

```
## [1] 1.104631
```

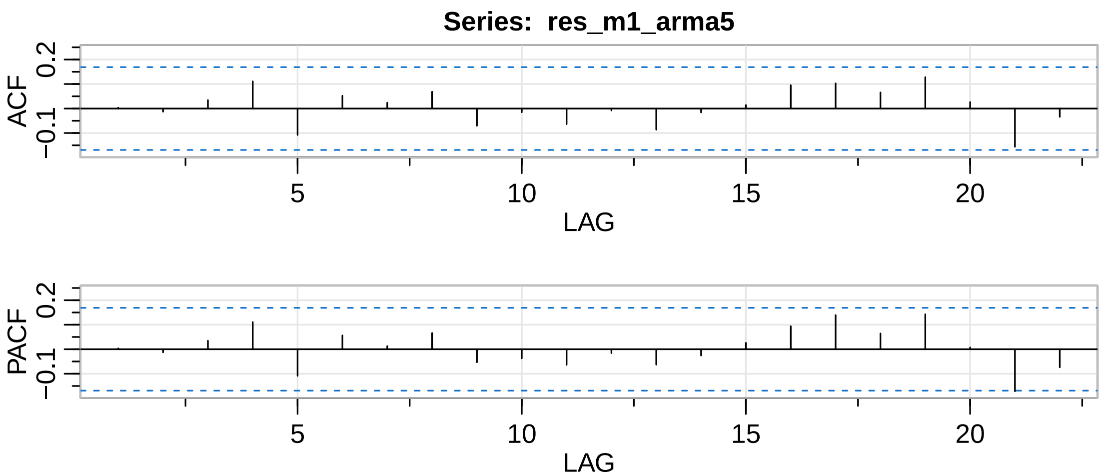
```
error(forecast=AR1_02_forecast, true=m1_ts[141:150], method=c("mape"))
```

```
## [1] 97.90185
```

Prueba del segundo mejor modelo ARMA(1,1)

Para validar el modelo se grafica el correlograma de los residuos para comprobar que son ruido blanco.

```
# Calculo ACF y PACF para el mejor modelo ARMA para el dataset modelo1
res_m1_arma5 <- residuos.modelos[,5]
acf2(res_m1_arma5)
```



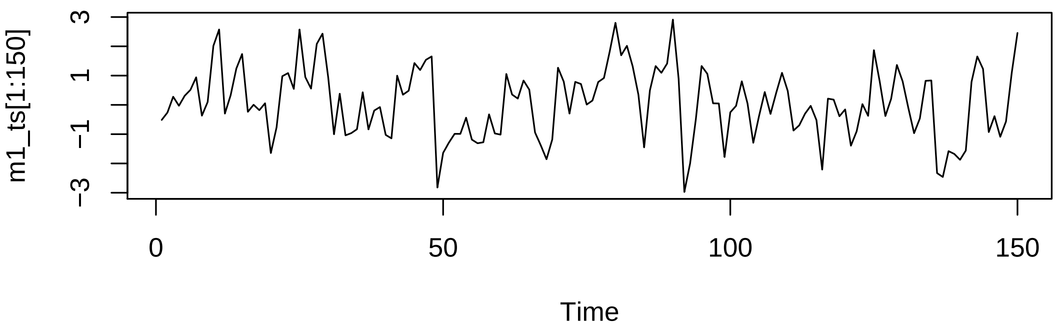

```
##      [,1]  [,2] [,3] [,4]  [,5] [,6] [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF     0 -0.01 0.03 0.11 -0.11 0.05 0.02 0.07 -0.07 -0.02 -0.06 -0.01 -0.09
## PACF    0 -0.01 0.03 0.11 -0.11 0.06 0.01 0.07 -0.05 -0.04 -0.06 -0.02 -0.06
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## ACF  -0.02  0.01  0.10  0.10  0.07  0.13  0.03 -0.16 -0.03
## PACF -0.03  0.03  0.09  0.14  0.06  0.14  0.01 -0.17 -0.07
```

En los correlogramas se aprecia que no hay ningún rezago significativo por lo que se puede decir que los residuos son ruido blanco.

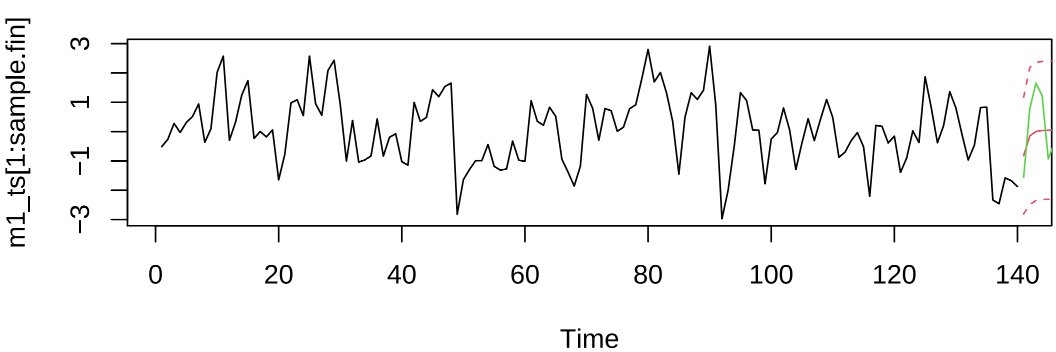
Por lo tanto estamos en condiciones de realizar un pronóstico para las siguientes 10 observaciones de la serie.

Pronóstico para Modelo 1 con ARMA(1,1)

```
#parametros p y q del mejor modelo
p <- 1
q <- 1
#fiteo el modelo con 140 datos (para despues hacer el out-of-sample forecast)
sample.fin <- 140
AR1_11 <- arima(m1_ts[1:sample.fin], order = c(p,0,q))
ts.plot(m1_ts[1:150])
```



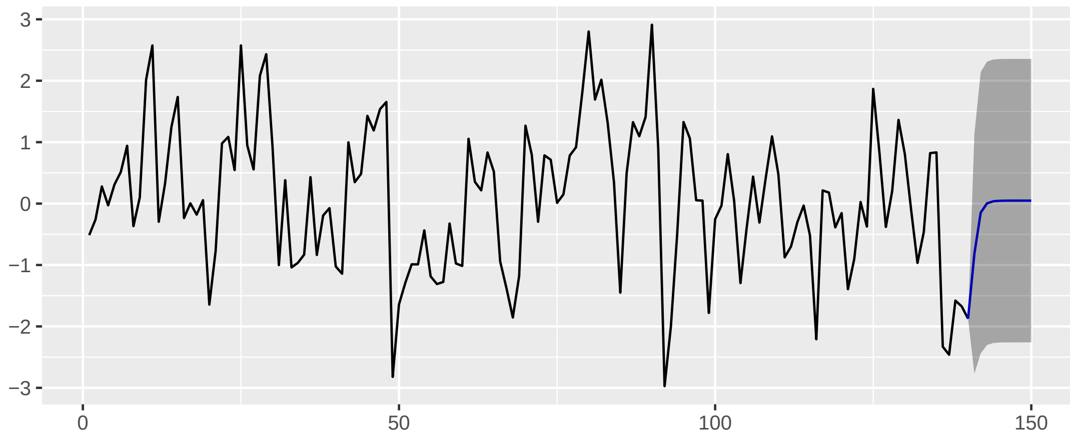

```
#plotting the m1_ts series plus the forecast and 95% prediction intervals
ts.plot(m1_ts[1:sample.fin], xlim = c(1, sample.fin))
AR1_11_forecast <- predict(AR1_11, n.ahead = length(m1_ts)-sample.fin)$pred
AR1_11_forecast_se <- predict(AR1_11, n.ahead = length(m1_ts)-sample.fin)$se
points(AR1_11_forecast, type = "l", col = 2)
points(AR1_11_forecast - 2*AR1_11_forecast_se, type = "l", col = 2, lty = 2)
points(AR1_11_forecast + 2*AR1_11_forecast_se, type = "l", col = 2, lty = 2)
points(sample.fin+1:150,m1_ts[sample.fin+1:150], type = "l", col = 3)
```



Otro método para pronóstico con ARMA(1,1)

```r
library(forecast)
arima5<- arima(m1_ts[1:sample.fin], order = c(1,0,1))
```

```r
forecast5<-forecast(arima5, level = c(95), h = 10)
autoplot(forecast5)
```



Comprobar los errores de los pronósticos con ARMA(1,1)

```r
error(forecast=AR1_11_forecast, true=m1_ts[141:150], method=c("mse"))
```

```
## [1] 1.525743
```

```r
error(forecast=AR1_11_forecast, true=m1_ts[141:150], method=c("mae"))
```

```
## [1] 1.112399
```

```r
error(forecast=AR1_11_forecast, true=m1_ts[141:150], method=c("mape"))
```

```
## [1] 98.66868
```