UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

Máster en Software de Sistemas Distribuidos y Empotrados

Ciencia de Datos

Ejercicio 3 – Series Temporales – ARIMA

Alejandro Casanova Martín

N.º de matrícula: bu0383

Madrid, 29 de abril 2024

Índice

1. Preparación de datos

Para cada uno de los tres archivos *real-daily-wages-in-pounds-engla.csv*, *monthly-milk-production-pounds-p.csv* y *highest-mean-monthly-level-lake-.csv*:

a) Cargarlos en un dataframe.

```
> #install.packages("forecast")       # install, if necessary
> library(forecast)
> # a)   Cargarlos en un dataframe.
> script_dir <- "C:/Users/alex/Desktop/Máster Software Embebido/2
  Segundo Semestre/2 Ciencia de Datos/Ejercicios" # Actualizar con el
  directorio correcto
> setwd(script_dir); getwd()
[1] "C:/Users/alex/Desktop/Máster Software Embebido/2 Segundo
Semestre/2 Ciencia de Datos/Ejercicios"
> # Dataset de salarios diarios por año en libras
> wages_input <- as.data.frame(read.csv("Ficheros/real-daily-wages-in-
  pounds-engla.csv", header=TRUE))
> wages_input <- wages_input[-736,] # Descartamos el último elemento,
  ya que no es útil
> head(wages_input)
  Year Real.daily.wages.in.pounds..England..1260...1994
1 1260                                             4.41
2 1261                                             4.63
3 1262                                             4.38
4 1263                                             4.52
5 1264                                             4.42
6 1265                                             4.64
> # Dataset de producción mensual de leche en libras
> milk_input <- as.data.frame(read.csv("Ficheros/monthly-milk-
  production-pounds-p.csv"))
> names(milk_input) <- c("date", "Monthly.milk.production.in.pounds") #
  Nombramos las columnas
> head(milk_input)
     date Monthly.milk.production.in.pounds
1 1962-02                             609.8
2 1962-03                             628.4
3 1962-04                             665.6
4 1962-05                             713.8
5 1962-06                             707.2
6 1962-07                             628.4
> # Dataset de nivel de agua medio por mes
> lake_input <- as.data.frame(read.csv("Ficheros/highest-mean-monthly-
  level-lake-.csv", header=TRUE))
> lake_input <- lake_input[-c(97,98),] # Descartamos el último
  elemento, ya que no es útil
> head(lake_input)
  Year Highest.mean.monthly.level..Lake.Michigan..1860.to.Dec.1955
1 1860                                                        83.3
2 1861                                                        83.5
3 1862                                                        83.2
4 1863                                                        82.6
5 1864                                                        82.2
6 1865                                                        82.1
```

b) Separar los últimos valores (un 10% aproximadamente es suficiente) para usarlos como test del modelo que entrenaremos con el resto de los valores.

```
> ### Para el dataset de salarios ##################################
> n_wages <- nrow(wages_input) # Tamaño del dataset
```

```
> n_wages_train <- floor(n_wages*0.9) # Tamaño del dataset para el
  modelado
> n_wages_test <- n_wages - n_wages_train # Tamaño del dataset para
  testing
> start_year_wages <- as.numeric(wages_input[1,1]) # Año inicial del
  dataset
> start_year_wages_test <- start_year_wages + n_wages_train # Año
  inicial del dataset de testing
> wages_input_train <- wages_input[1:n_wages_train,]
> wages_input_test <- wages_input[(n_wages_train+1):n_wages,]

> ### Para el dataset de producción de leche #################
> start_year_milk <- 1962 # Año inicial del dataset
> start_offset_milk <- 1
> n_milk <- nrow(milk_input) # Tamaño del dataset
> n_milk_train <- floor(n_milk*0.9) # # Tamaño del dataset para el
  modelado (143)
> n_milk_test <- n_milk - n_milk_train # Tamaño del dataset para
  testing
> # Redondea al próximo múltiplo de 12 y resta 1 (empieza en febrero)
> n_milk_train <- n_milk_train + (12 - n_milk_train %% 12) -
  start_offset_milk
> start_year_milk_test <- start_year_milk + round(n_milk_train / 12) #
  Año inicial del dataset de testing (1974)
> milk_input_train <- milk_input[1:n_milk_train,]
> milk_input_test <- milk_input[(n_milk_train+1):n_milk,]

> ### Para el dataset de niveles de agua del lago Michigan ###########
> n_lake <- nrow(lake_input) # Tamaño del dataset
> n_lake_train <- floor(n_lake*0.9) # Tamaño del dataset para el
  modelado
> n_lake_test <- n_lake - n_lake_train # Tamaño del dataset para
  testing
> start_year_lake <- as.numeric(lake_input[1,1]) # Año inicial del
  dataset
> start_year_lake_test <- start_year_lake + n_lake_train # Año inicial
  del dataset de testing
> lake_input_train <- lake_input[1:n_lake_train,]
> lake_input_test <- lake_input[(n_lake_train+1):n_lake,]
```

c) Crear series temporales con los primeros valores (para entrenar el modelo) y los últimos valores (para test), fijando el inicio adecuado para cada serie temporal y, en su caso, la frecuencia.

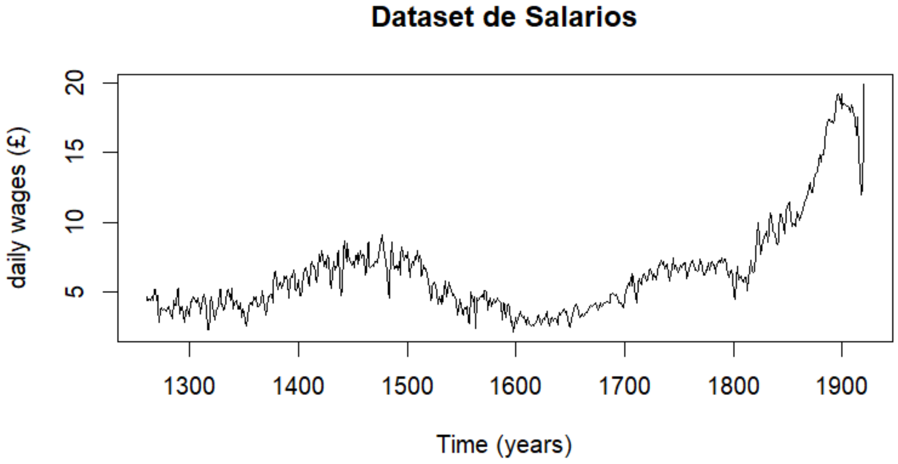
```
> ### Para el dataset de salarios ##################################
> wages <- ts(as.numeric(wages_input_train[,2]),
  start=start_year_wages)
> wages_test <- ts(as.numeric(wages_input_test[,2]),
  start=start_year_wages_test)

> ### Para el dataset de producción de leche #########################
> milk <- ts(as.numeric(milk_input_train[,2]), start=c(start_year_milk,
  2), frequency = 12)
> milk_test <-
  ts(as.numeric(milk_input_test[,2]),start=start_year_milk_test,
  frequency = 12)
> ### Para el dataset de niveles de agua del lago Michigan ############
> lake <- ts(as.numeric(lake_input_train[,2]),start=start_year_lake)
> lake_test <-
  ts(as.numeric(lake_input_test[,2]),start=start_year_lake_test)
```

2. Búsqueda y evaluación del modelo ARIMA

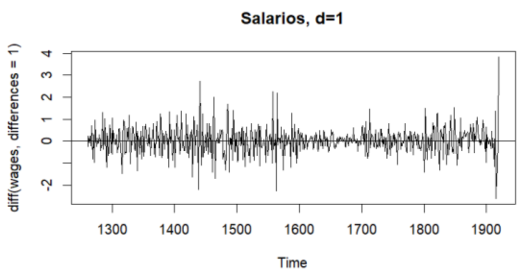
d) Analizar los valores apropiados de d (y en su caso D y s) para el ajuste de un modelo ARIMA o ARIMA estacional, para asegurar que la serie temporal sea estacionaria.

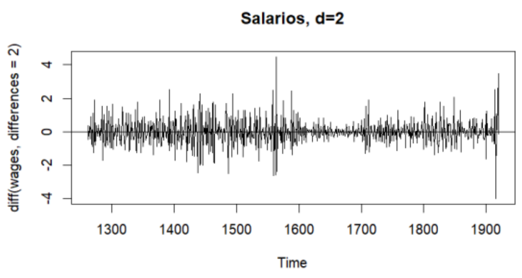
Primero para el dataset de salarios:

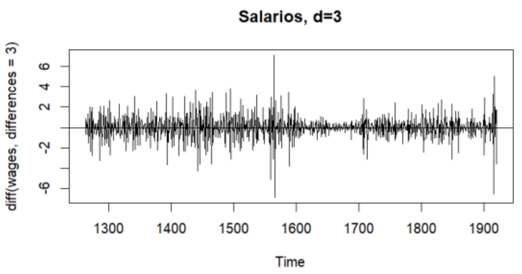
```
> plot(wages, xlab = "Time (years)",
+      ylab = "daily wages (£)",
+      main = "Dataset de Salarios")
```




```
> # Comprobamos las condiciones para serie estacionaria
> plot(diff(wages,differences=1), main= "Salarios, d=1")
> abline(a=0, b=0)
> var(diff(wages,differences=1))
[1] 0.3795782
> plot(diff(wages,differences=2), main= "Salarios, d=2")
> abline(a=0, b=0)
> var(diff(wages,differences=2))
[1] 0.6666833
> plot(diff(wages,differences=3), main= "Salarios, d=3")
> abline(a=0, b=0)
> var(diff(wages,differences=3))
[1] 1.787241
```



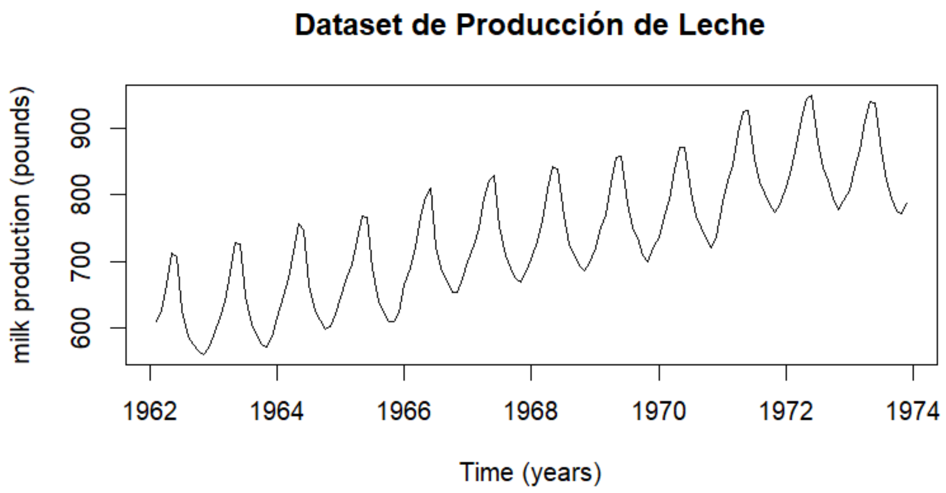




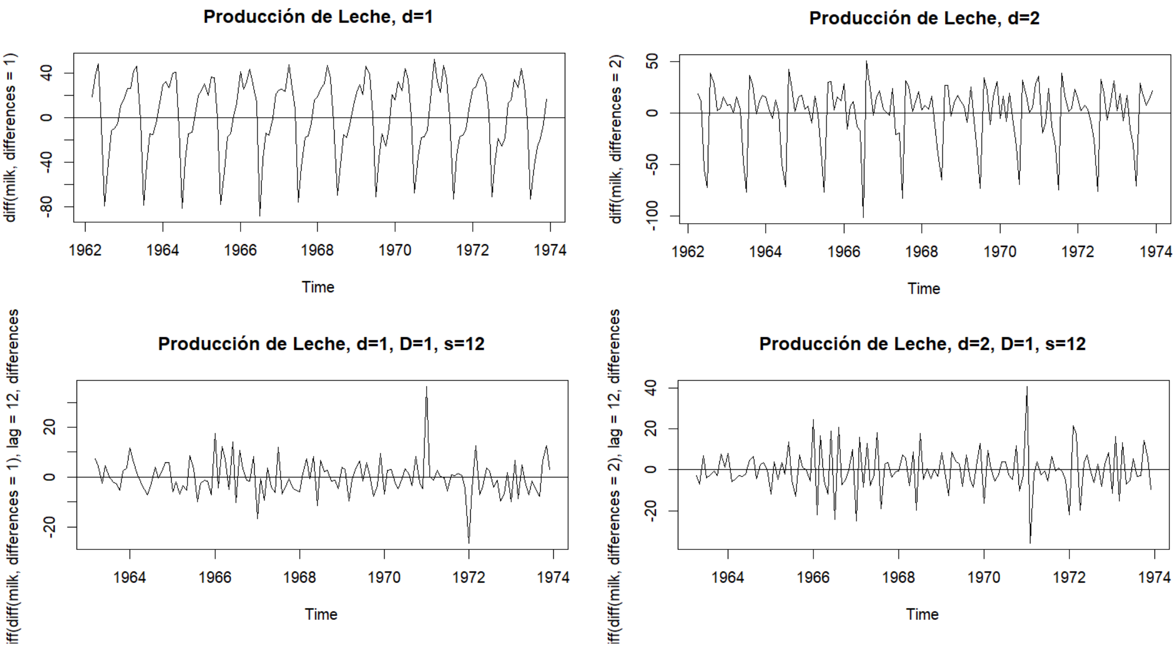
A la vista de las gráficas y los valores de varianza, nos decantaremos por d = 1 o por d = 2. Según el aspecto que tengan las gráficas del ACF y PACF, escogeremos uno u otro.

Para el dataset de producción de leche:

```
> plot(milk, xlab = "Time (years)",
+      ylab = "milk production (pounds)",
+      main = "Dataset de Producción de Leche")
```



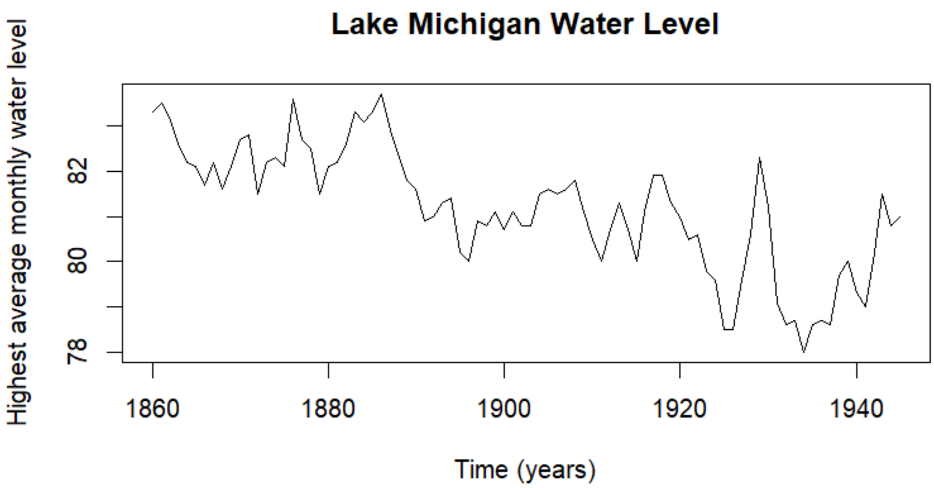

```
> # Comprobamos las condiciones para serie estacionaria
> plot(diff(milk,differences=1), main= "Producción de Leche, d=1")
> abline(a=0, b=0)
> var(diff(milk,differences=1))
[1] 1155.648
> plot(diff(milk,differences=2), main= "Producción de Leche, d=2")
> abline(a=0, b=0)
> var(diff(milk,differences=2))
[1] 901.8661
> plot(diff(diff(milk,differences=1),lag=12,differences=1), main=
   "Producción de Leche, d=1, D=1, s=12")
> abline(a=0, b=0)
> var(diff(diff(milk,differences=1),lag=12,differences=1))
[1] 51.80197
> plot(diff(diff(milk,differences=2),lag=12,differences=1), main=
   "Producción de Leche, d=2, D=1, s=12")
> abline(a=0, b=0)
> var(diff(diff(milk,differences=2),lag=12,differences=1))
[1] 116.2855
```



A la vista de las gráficas y los valores de la varianza, nos decantamos por D=1, d=1 y s=12.

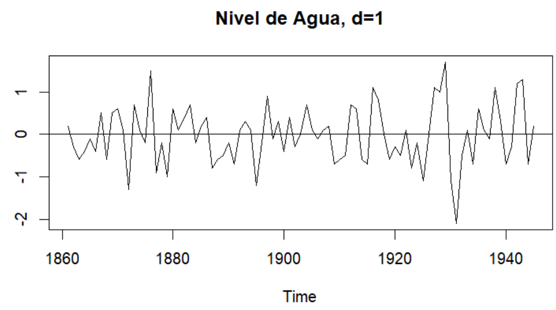
Para el dataset de nivel de agua del lago Michigan:

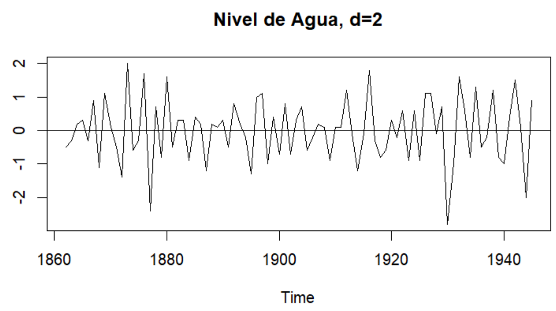
```
> plot(lake, xlab = "Time (years)",
+      ylab = "Highest average monthly water level",
+      main = "Lake Michigan Water Level")
```

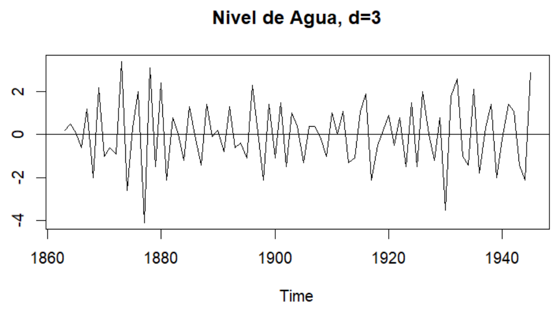



```
> plot(diff(lake,differences=1), main="Nivel de Agua, d=1", ylab="")
> abline(a=0, b=0)
> var(diff(lake,differences=1))
[1] 0.4698543
> plot(diff(lake,differences=2), main="Nivel de Agua, d=3", ylab="")
> abline(a=0, b=0)
> var(diff(lake,differences=2))
[1] 0.8751807
> plot(diff(lake,differences=3), main="Nivel de Agua, d=2", ylab="")
```

```
> abline(a=0, b=0)
> var(diff(lake,differences=3))
[1] 2.363371
```





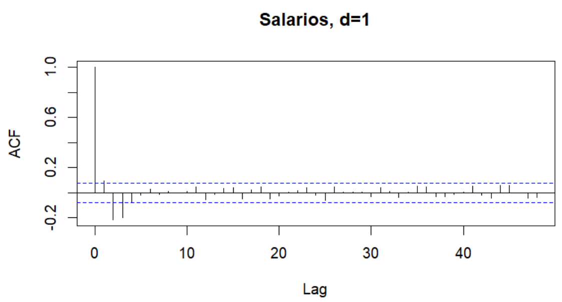


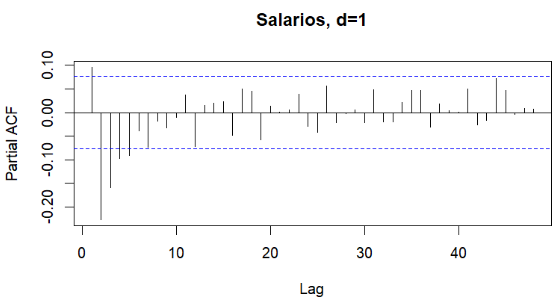
A la vista de las gráficas y los valores de varianza, decidiremos entre d=1 y d=2 en función de las gráficas del ACF y PACF.

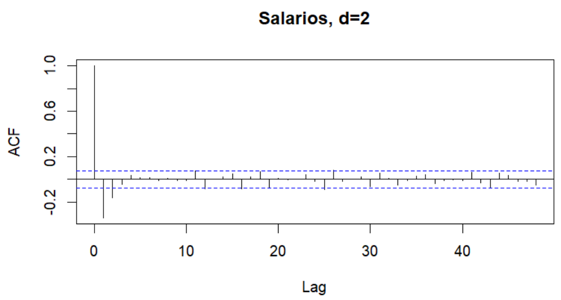
e) Con dichos valores de d (y en su caso, D y S), obtener las gráficas de ACF y PACF para evaluar los valores apropiados para p, q (y, en su caso, P y Q).

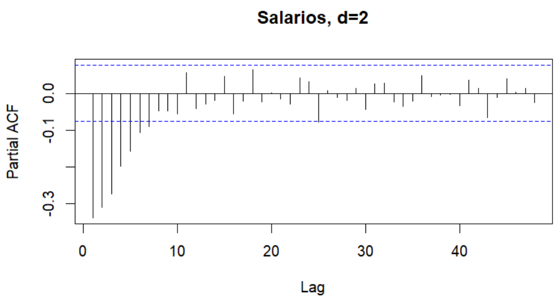
Para el dataset de salarios:

```
> acf(diff(wages,differences=1), lag.max=48, main="Salarios, d=1")
> pacf(diff(wages,differences=1), lag.max=48, main="Salarios, d=1")
> acf(diff(wages,differences=2), lag.max=48, main="Salarios, d=1")
> pacf(diff(wages,differences=2), lag.max=48, main="Salarios, d=1")
```



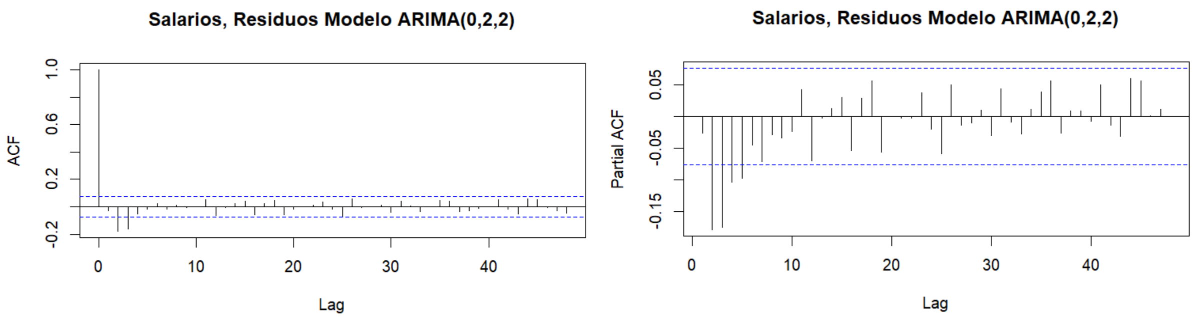






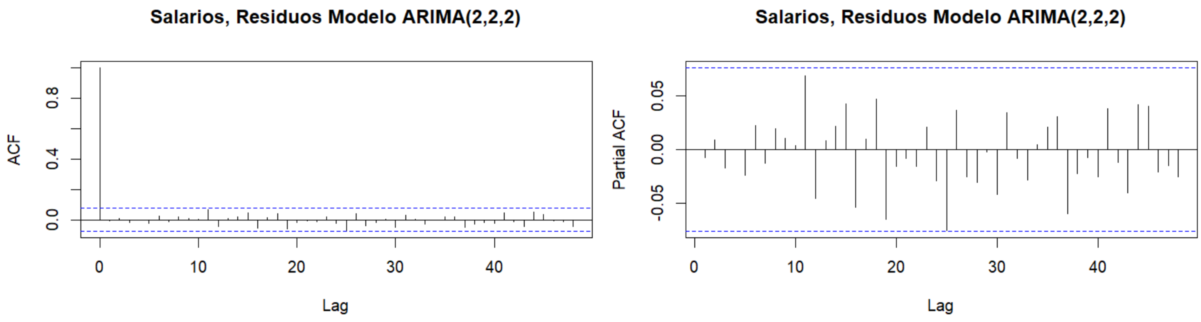
Para d=2 el comportamiento es más claro, y se corresponde con un modelo MA (2). Ajustamos un modelo ARIMA (0,2,2), y analizamos sus residuos:

```
> arima_wages_1 <- arima(wages, order=c(0,2,2))
> arima_wages_1

Call:
arima(x = wages, order = c(0, 2, 2))

Coefficients:
          ma1      ma2
      -0.8407  -0.1593
s.e.   0.0464   0.0453

sigma^2 estimated as 0.3735:  log likelihood = -613.69,  aic = 1233.38
> AIC(arima_wages_1, k = log(length(wages))) # BIC
[1] 1246.862
> # Examinamos el ACF y PACF de los residuos
> acf(arima_wages_1$residuals, lag.max=48, main="Salarios, Residuos
  Modelo ARIMA(0,2,2)")
> pacf(arima_wages_1$residuals, lag.max=48, main="Salarios, Residuos
  Modelo ARIMA(0,2,2)")
```



Como los residuos no son satisfactorios, probamos con un modelo ARIMA (2,2,2).

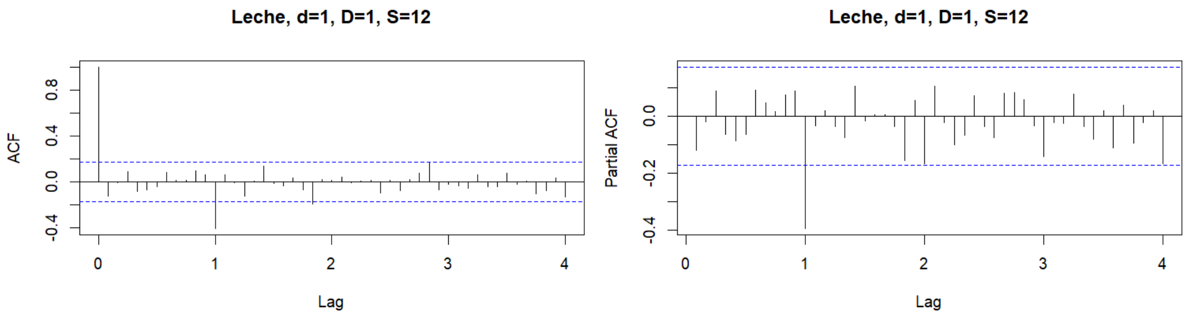
```
> arima_wages_2 <- arima(wages, order=c(2,2,2))
> arima_wages_2

Call:
arima(x = wages, order = c(2, 2, 2))

Coefficients:
         ar1      ar2      ma1     ma2
      0.7648  -0.3258  -1.7332  0.7362
s.e.  0.0589   0.0404   0.0498  0.0500

sigma^2 estimated as 0.3324:  log likelihood = -574.97,  aic = 1159.94
> AIC(arima_wages_2, k = log(length(wages))) # BIC
[1] 1182.407
> # Examinamos el ACF y PACF de los residuos
> acf(arima_wages_2$residuals, lag.max=48, main="Salarios, Residuos
  Modelo ARIMA(2,2,2)")
> pacf(arima_wages_2$residuals, lag.max=48, main="Salarios, Residuos
  Modelo ARIMA(2,2,2)")
```



En este caso, tanto el PACF y ACF de los residuos, como el AIC y e BIC del modelo tienen mejor aspecto, por lo que nos decantamos por el modelo ARIMA (2,2,2).

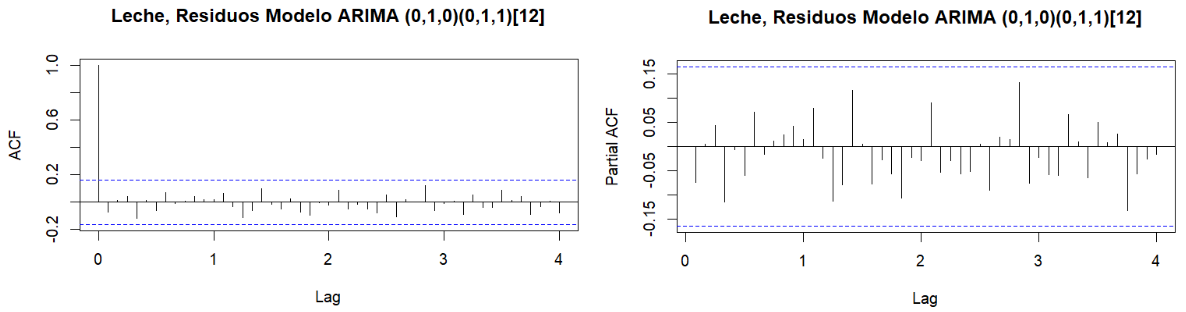
Para el dataset de producción de leche:

```
> acf(diff(diff(milk,differences=1),lag=12,differences=1), lag.max=48,
+     main="Leche, d=1, D=1, S=12")
> pacf(diff(diff(milk,differences=1),lag=12,differences=1),
   lag.max=48, main="Leche, d=1, D=1, S=12")
```



A las vista de las gráficas, modelamos la componente estacional ajustando un modelo ARIMA (0,1,0) x (0,1,1) [12].

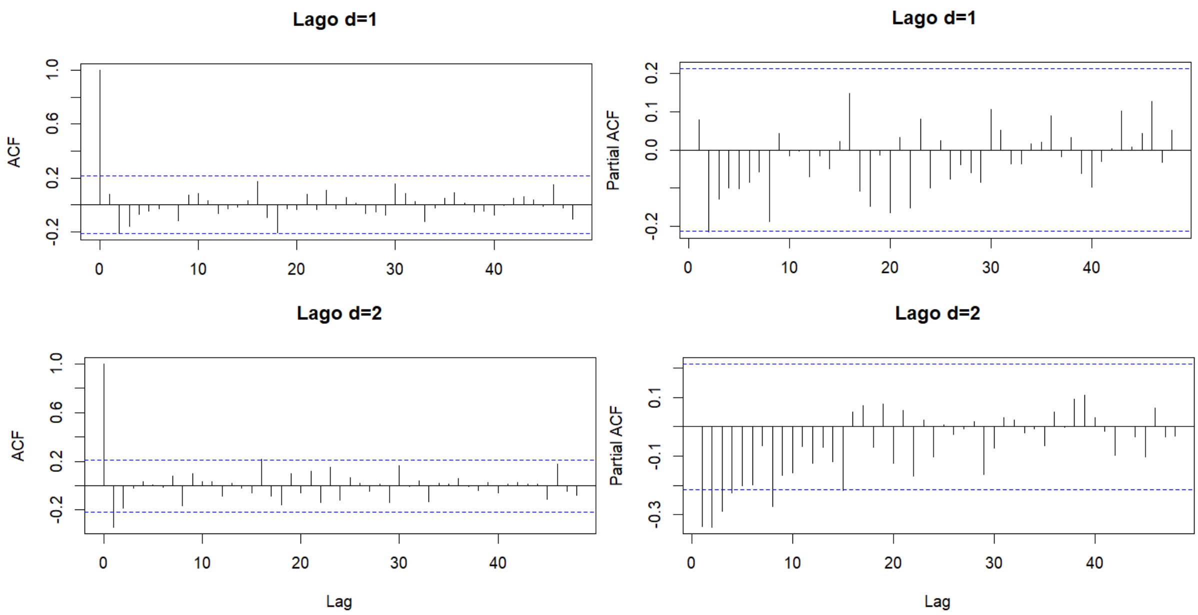
```
> arima_milk_1 <- arima (milk, order=c(0,1,0),
+                  seasonal = list(order=c(0,1,1), period=12))
> arima_milk_1

Call:
arima(x = milk, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 1),
period = 12))

Coefficients:
         sma1
      -0.6861
s.e.   0.0844

sigma^2 estimated as 35.98:  log likelihood = -421.18,  aic = 846.36
> AIC(arima_milk_1, k = log(length(milk))) # BIC
[1] 852.2825
> acf(arima_milk_1$residuals, lag.max=48, main="Leche, Residuos Modelo
   ARIMA (0,1,0)(0,1,1)[12]")
> pacf(arima_milk_1$residuals, lag.max=48, main="Leche, Residuos Modelo
   ARIMA (0,1,0)(0,1,1)[12]")
```



A la vista de las gráficas, no es necesario añadir más parámetros. Nos quedamos con el modelo ARIMA (0,1,0) x (0,1,1) [12].

Para el dataset del lago Michigan:

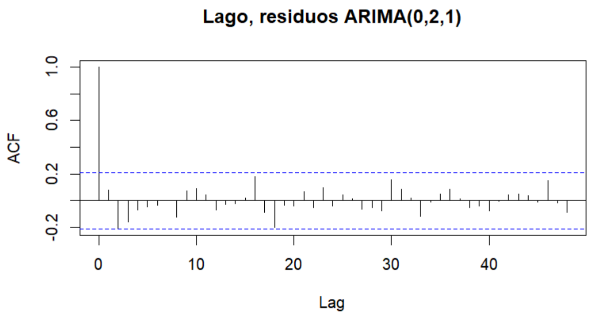
```
> acf(diff(lake,differences=1), lag.max=48, main="Lago d=1")
> pacf(diff(lake,differences=1), lag.max=48, main="Lago d=1")
> acf(diff(lake,differences=2), lag.max=48, main="Lago d=2")
> pacf(diff(lake,differences=2), lag.max=48, main="Lago d=2")
```

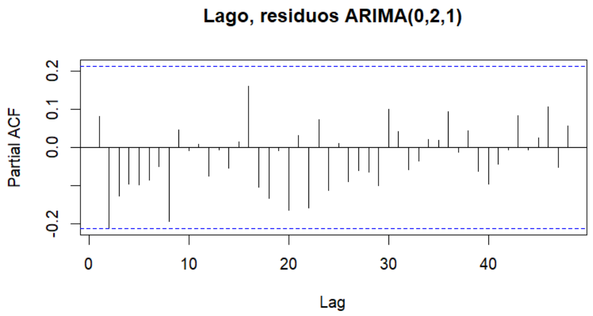


A la vista de las gráficas, nos decantamos por d=2, que ofrece una respuesta más representativa de un modelo MA (1). A continuación, ajustamos dicho modelo, y analizamos el PACF y ACF de sus residuos.

```
> arima_lake_1 <- arima(lake, order=c(0,2,1))
> arima_lake_1

Call:
arima(x = lake, order = c(0, 2, 1))

Coefficients:
          ma1
       -1.000
s.e.    0.034

sigma^2 estimated as 0.4699:  log likelihood = -89.69,  aic = 183.38
> AIC(arima_lake_1, k = log(length(lake))) # BIC
```

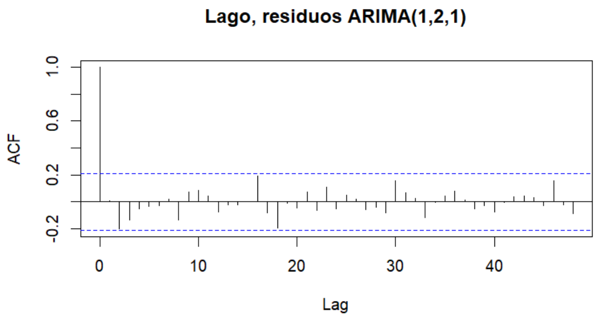
```
[1] 188.2853
> # Examinamos el ACF y PACF de los residuos
> acf(arima_lake_1$residuals, lag.max=48, main="Lago, residuos
  ARIMA(0,2,1)")
> pacf(arima_lake_1$residuals, lag.max=48, main="Lago, residuos
  ARIMA(0,2,1)")
```

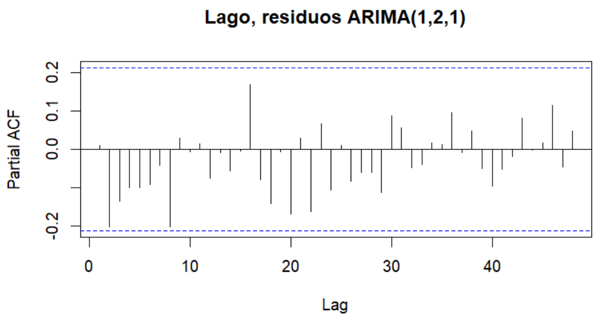




Los residuos presentan un aspecto mejorable, por lo que probamos con un modelo ARIMA (1,2,1):

```
> arima_lake_2 <- arima(lake, order=c(1,2,1))
> arima_lake_2

Call:
arima(x = lake, order = c(1, 2, 1))

Coefficients:
         ar1      ma1
      0.0905   -1.000
s.e.  0.1090    0.033

sigma^2 estimated as 0.467:  log likelihood = -89.34,  aic = 184.69
> AIC(arima_lake_2, k = log(length(lake))) # BIC
[1] 192.0524
> # Examinamos el ACF y PACF de los residuos
> acf(arima_lake_2$residuals, lag.max=48, main="Lago, residuos
  ARIMA(1,2,1)")
> pacf(arima_lake_2$residuals, lag.max=48, main="Lago, residuos
  ARIMA(1,2,1)")
```





Los residuos tienen un aspecto un poco mejor, por lo que nos quedamos con este modelo.

f) Comparar los resultados obtenidos con la solución proporcionada por el comando *auto.arima* de R.

Para el dataset de salarios, hemos obtenido con autoarima el mismo modelo que con el ajuste manual (p=2, q=2, d=2):

```
> arima_wages_3=auto.arima(wages, d=2, max.order=4,
+                          trace=TRUE, approx=FALSE,
+                          allowdrift=FALSE, stepwise=FALSE)
> arima_wages_3
Series: wages
ARIMA(2,2,2)

Coefficients:
         ar1      ar2      ma1     ma2
      0.7648  -0.3258  -1.7332  0.7362
s.e.  0.0589   0.0404   0.0498  0.0500

sigma^2 = 0.3344:  log likelihood = -574.97
AIC=1159.94   AICc=1160.03   BIC=1182.39
```

Para el dataset de producción de leche:

```
> arima_milk_2=auto.arima(milk, d=1, D=1, max.order=4,
+                          trace=TRUE, approx=FALSE,
+                          allowdrift=FALSE, stepwise=FALSE)
> arima_milk_2 # Best model: ARIMA (0,1,0) (0,1,1) [12]
Series: milk
ARIMA(0,1,0)(0,1,1)[12]

Coefficients:
         sma1
      -0.6861
s.e.   0.0844

sigma^2 = 36.3:  log likelihood = -421.18
AIC=846.36   AICc=846.45   BIC=852.09
```

También en este caso, el modelo obtenido con autoarima coincide con el ajustado manualmente. Se trata de un modelo ARIMA (0,1,0) x (0,1,1) [12].

Para el dataset del lago Michigan:

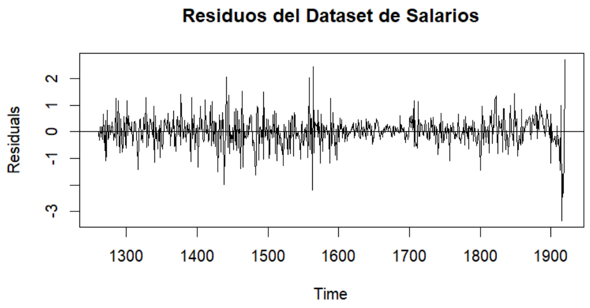
```
> arima_lake_3=auto.arima(lake, d=1, max.order=4,
+                         trace=TRUE, approx=FALSE,
+                         allowdrift=FALSE, stepwise=FALSE)
> arima_lake_3
Series: lake
ARIMA(2,1,1)

Coefficients:
         ar1      ar2      ma1
      0.8647  -0.2535  -0.8636
s.e.  0.1298   0.1099   0.0920

sigma^2 = 0.427:  log likelihood = -83.16
AIC=174.33   AICc=174.83   BIC=184.1
```

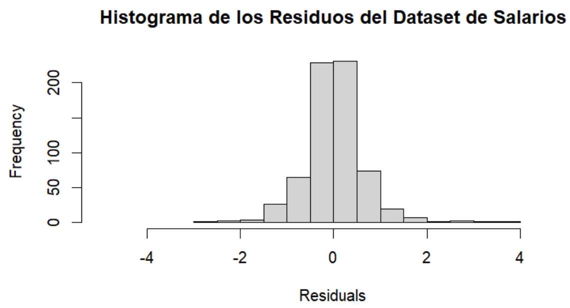
En este caso, el modelo generado con autoarima presenta valores de AIC y BIC mejores que los del modelo ajustado manualmente, por lo que nos quedaremos con este. Se trata de un modelo ARIMA (2,1,1).

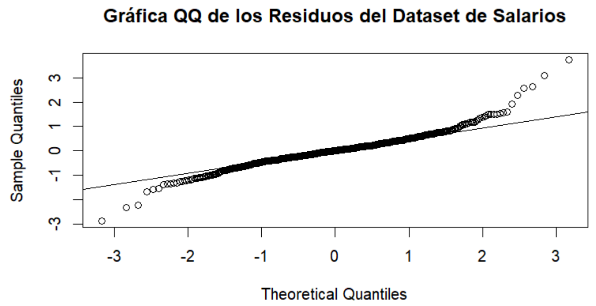
g) Analizar la normalidad de los residuos.

Para el dataset de salarios:

```
> plot(arima_wages_2$residuals, ylab = "Residuals", main="Residuos del
  Dataset de Salarios")
> abline(a=0, b=0)
> hist(arima_wages_1$residuals, xlab="Residuals", xlim=c(-5,5),
+      main="Histograma de los Residuos del Dataset de Salarios")
> qqnorm(arima_wages_1$residuals, main="Gráfica QQ de los Residuos del
  Dataset de Salarios")
> qqline(arima_wages_1$residuals)
```



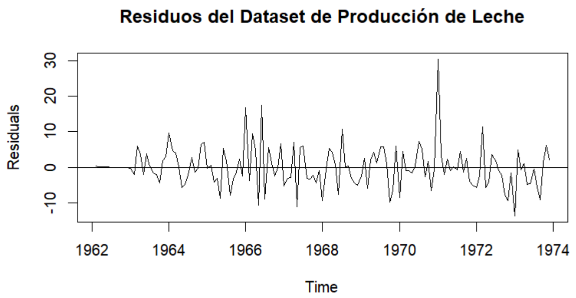


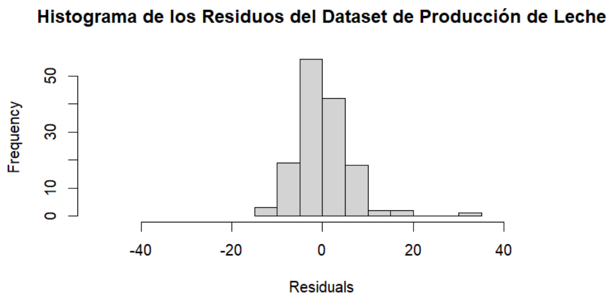


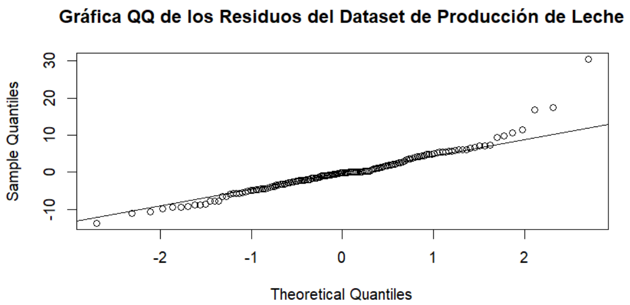
En el caso del dataset de salarios, los residuos parecen ajustarse más o menos a una distribución normal, por lo que podemos esperar una buena predicción.

Para el dataset de producción de leche:

```
> plot(arima_milk_1$residuals, ylab = "Residuals", main="Residuos del
  Dataset de Producción de Leche")
> abline(a=0, b=0)
> hist(arima_milk_1$residuals, xlab="Residuals", xlim=c(-50,50),
+      main="Histograma de los Residuos del Dataset de Producción de
  Leche")
> qqnorm(arima_milk_1$residuals, main="Gráfica QQ de los Residuos del
  Dataset de Producción de Leche")
> qqline(arima_milk_1$residuals)
```



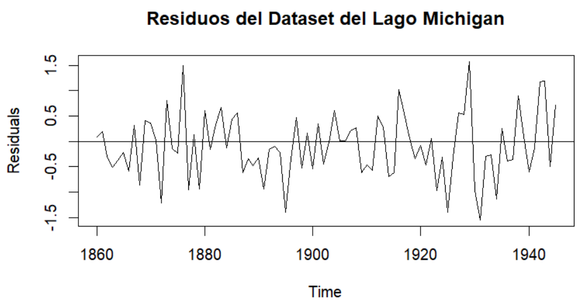


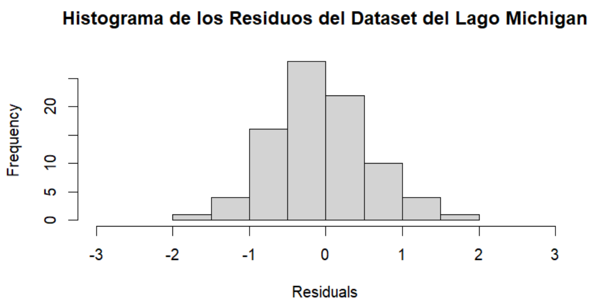


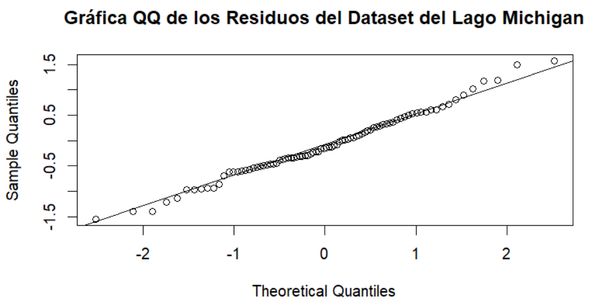
En este caso, los residuos parecen aproximarse también a una distribución normal, por lo que esperaremos buenas predicciones del modelo.

Finalmente, para el dataset del Lago Michigan:

```
> plot(arima_lake_3$residuals, ylab = "Residuals", main="Residuos del
  Dataset del Lago Michigan")
> abline(a=0, b=0)
> hist(arima_lake_3$residuals, xlab="Residuals", xlim=c(-3,3),
+      main="Histograma de los Residuos del Dataset del Lago Michigan")
> qqnorm(arima_lake_3$residuals, main="Gráfica QQ de los Residuos del
  Dataset del Lago Michigan")
> qqline(arima_lake_3$residuals)
```





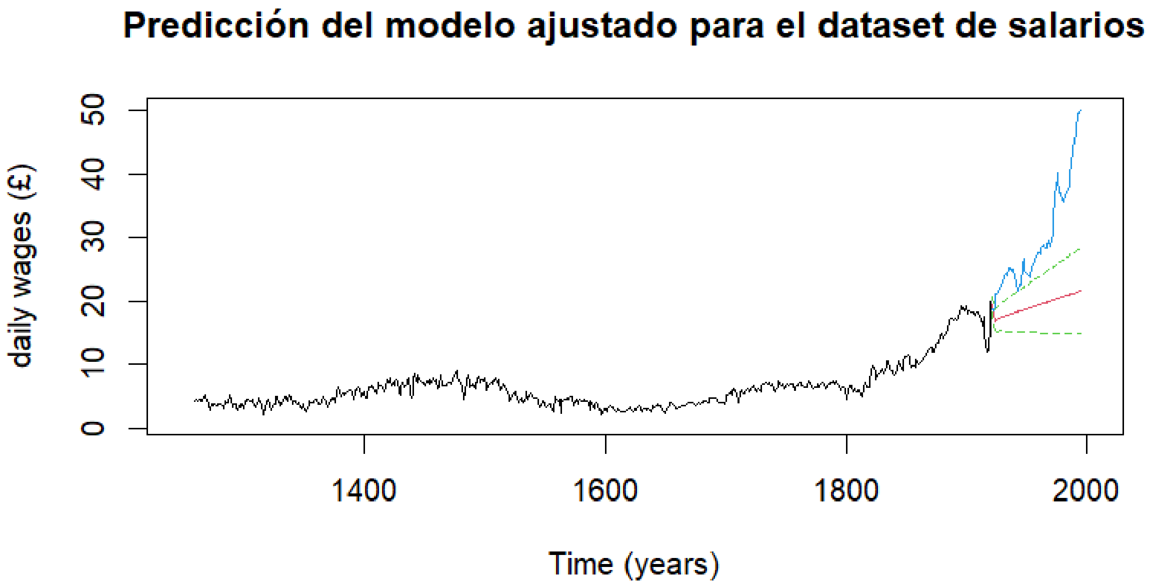


También en este caso los residuos parecen ajustarse a la distribución normal, por lo que las predicciones del modelo prometen ser buenas.

h) Representar gráficamente las predicciones del modelo, con sus intervalos de confianza, y compararlos con los datos reales que se reservaron para test.

Para el dataset de salarios:

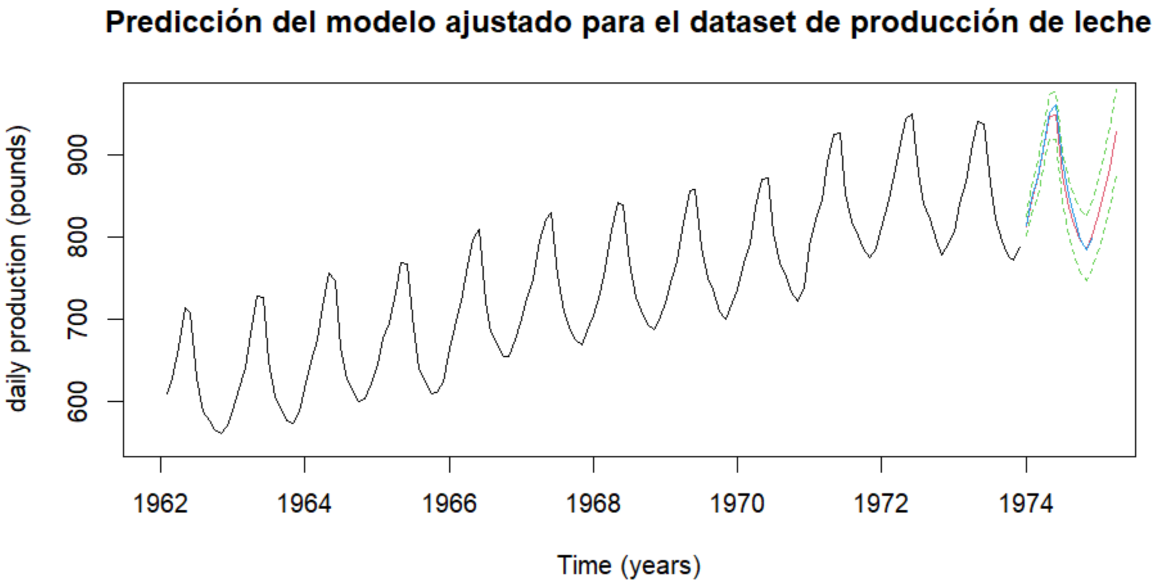
```
> arima_wages_3.predict <- predict(arima_wages_3, n.ahead=n_wages_test)
> plot(wages, xlab = "Time (years)",
+      ylab = "daily wages (£)",
+      ylim = c(1, 50),
+      xlim = c(1250, 2000),
+      main = "Predicción del modelo ajustado para el dataset de
  salarios")
> lines(arima_wages_3.predict$pred, col=2)
> lines(arima_wages_3.predict$pred+1.96*arima_wages_3.predict$se,
  col=3, lty=2)
> lines(arima_wages_3.predict$pred-1.96*arima_wages_3.predict$se,
  col=3, lty=2)
> lines(wages_test, col=4)
```



Los intervalos de confianza parecen ser considerablemente estrechos, lo que indica una predicción potencialmente buena. Sin embargo, en este caso la predicción se ha alejado significativamente de los datos reales, que se han disparado saliéndose rápidamente de los márgenes de confianza. Aunque el modelo parecía ser bueno, en este caso los datos presentaban un crecimiento repentino muy difícil de predecir sólo en base a los valores previos de la serie. Los datos seleccionados para testing se corresponden a los años 20 y posteriores del siglo XX, en los que se dio un fuerte crecimiento económico debido a la revolución industrial. Como ya se ha comentado, este crecimiento difícilmente podía predecirse sólo a base de los datos previos de salarios, y también contradice la hipótesis de estacionariedad de la que se partió para construir el modelo.

Para el dataset de producción de leche:

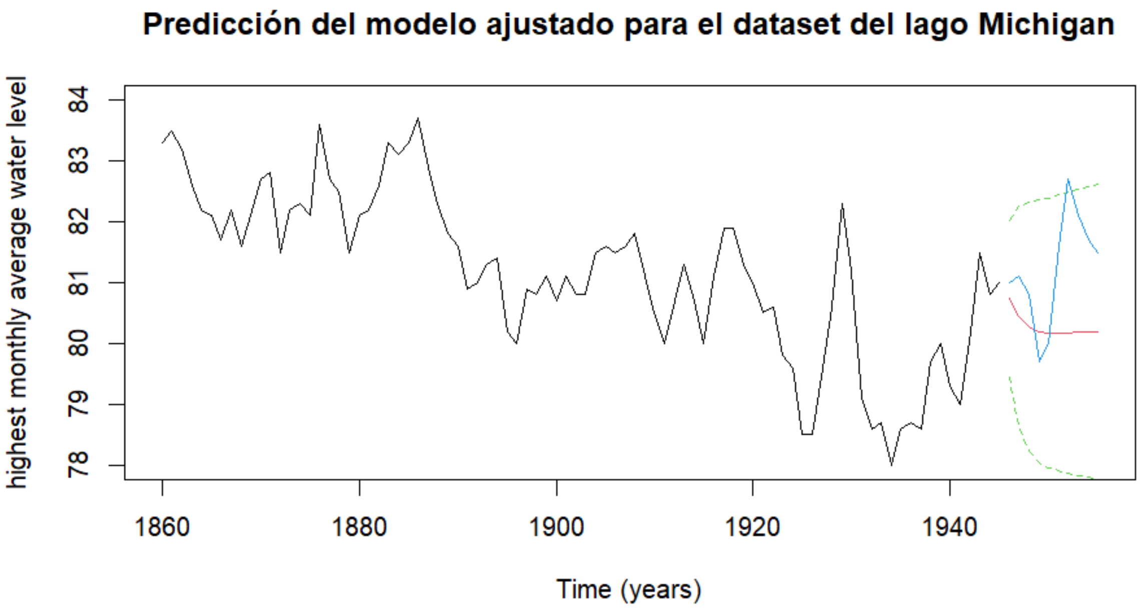
```
> arima_milk.predict <- predict(arima_milk_1, n.ahead=n_milk_test)
> plot(milk, xlab = "Time (years)",
+      ylab = "daily production (pounds)",
+      ylim = c(550, 970),
+      xlim = c(1962, 1975),
+      main = "Predicción del modelo ajustado para el dataset de
  producción de leche")
> lines(arima_milk.predict$pred, col=2)
> lines(arima_milk.predict$pred+1.96*arima_milk.predict$se, col=3,
  lty=2)
> lines(arima_milk.predict$pred-1.96*arima_milk.predict$se, col=3,
  lty=2)
> lines(milk_test, col=4)
```



En este caso, la predicción ha sido significativamente buena. Tanto la predicción como los datos reales han quedado contenidos en los márgenes de confianza, y el ajuste ha sido muy bueno.

Para el dataset del lago Michigan:

```
> arima_lake.predict <- predict(arima_lake_3, n.ahead=n_lake_test)
> plot(lake, xlab = "Time (years)",
+      ylab = "highest monthly average water level",
+      ylim = c(78, 84),
+      xlim = c(1860, 1955),
+      main = "Predicción del modelo ajustado para el dataset del lago
  Michigan")
> lines(arima_lake.predict$pred, col=2)
> lines(arima_lake.predict$pred+1.96*arima_lake.predict$se, col=3,
  lty=2)
> lines(arima_lake.predict$pred-1.96*arima_lake.predict$se, col=3,
  lty=2)
> lines(lake_test, col=4)
```



En este caso, los intervalos de confianza son demasiado amplios, lo que limita la utilidad del modelo. La predicción no se ajusta bien a los datos reales que, además, sobrepasan ligeramente el margen de confianza. Podemos intuir que la naturaleza del dataset es considerablemente imprevisible y esporádica, además de que cuenta con alto nivel de agregación, y una resolución baja (dado que, por cada año, la medida tomada corresponde al valor medio de un único mes; el más alto de los doce meses que componen el año). Aunque el modelo no ha conseguido capturar las fluctuaciones del dataset, sí que parece haber asimilado la tendencia generalmente decreciente de los datos.