Taller 5 - simulación de datos atípicos

Adaptado del taller 2022 de Federico Molina

Series Cronológicas 2024

Mayo 2024

1. Introducción

En este taller, se simula un modelo AR(2) y se introduce distintos tipos de *outliers* (aditivos, cambio transitorio y cambio de nivel) y se analiza su impacto en el proceso de modelización. La presencia de un dato atípico afectará la estimación del modelo y probablemente dejará de cumplirse el supuesto de normalidad de los residuos, con lo cual también la inferencia sobre los coeficientes y la capacidad predictiva del modelo se verán afectadas. Para remediar esta situación, será necesario intervenir el modelo mediante la inclusión de un regresor externo en la estimación del modelo.

2. Simulación e introducción de un dato atípico

```r
# Simulamos un proceso AR(2) con media 2 y desvío estándar 0,07
# Todavía no introducimos el cambio de nivel
phi_1 <-  -0.5
phi_2 <- 0.4
media <- 2
desvio <- 0.07
N <- 200

yt <- arima.sim(n = N,
                model = list(ar = c(phi_1, phi_2)),
                mean = media,
                sd = desvio)
```

```r
# Graficamos la serie simulada, su FAC y su FACP
grafico_yt <- autoplot(yt) +
  labs(x = "Observación",
       y = "Valor") +
  theme(panel.grid.minor = element_blank())

yt_acf <- ggAcf(yt, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

yt_pacf <- ggAcf(yt, lag.max = 50, type = "partial") +
```

```r
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grafico_yt / (yt_acf + yt_pacf)
```

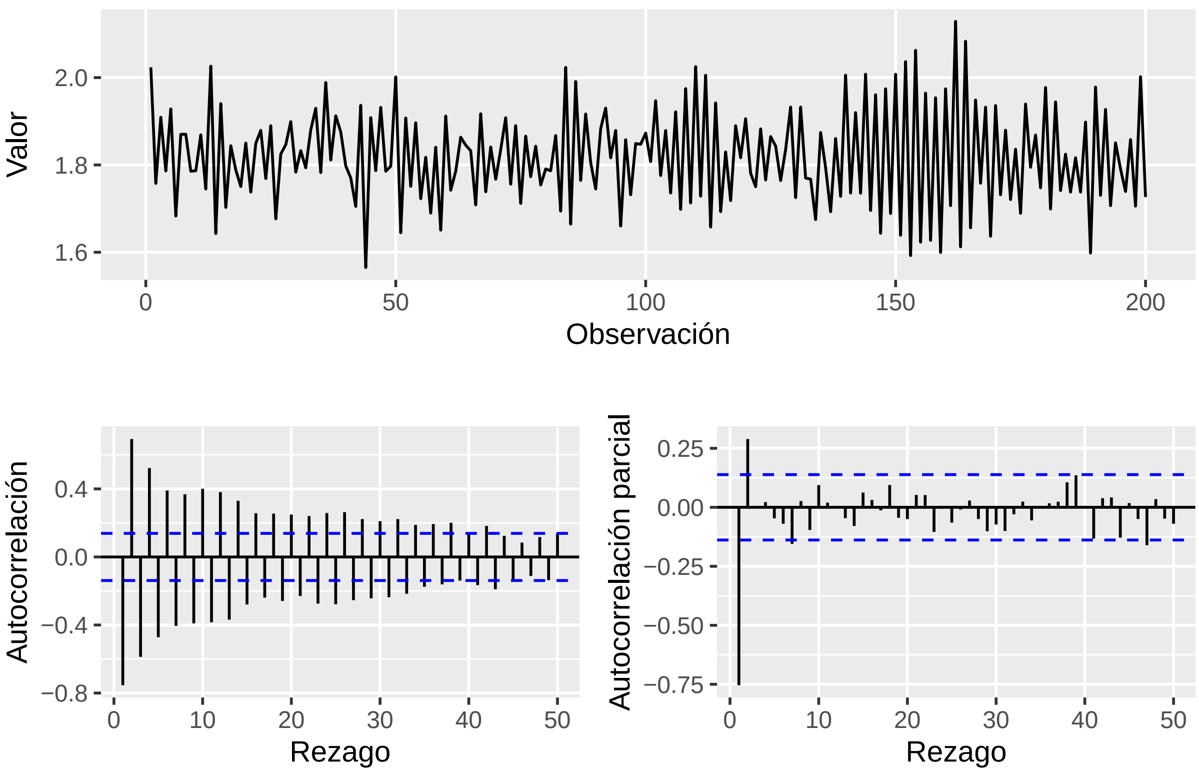


Figura 1: Simulación de un proceso AR(2).

2.1. Atípico aditivo (AO)

Cuando se advierte un punto raro que afecta únicamente una observación de la serie, se está en presencia de un *outlier* de tipo AO. Sea W_A el impacto asociado al punto raro, I_t^h una variable externa que indica su ocurrencia en el momento h y $\Psi(L)\varepsilon_t$ un proceso estacionario, el proceso resultante Z_t puede escribirse como:

$$Z_t = W_A I_t^h + \Psi(L)\varepsilon_t$$

La variable I_t^h vale uno en el momento h y cero en todos los demás:

$$I_t = \begin{cases} 1 & \text{si } t = h \\ 0 & \text{en otro caso} \end{cases}$$

```r
# Introducimos un AO en la observación 100
quiebre <- 100
I <- rep(0, N)
```

```r
I[quiebre] <- 1
zt_AO <- yt
zt_AO <- 0.4 * I + yt
```

```r
# Graficamos la serie con un AO, su FAC y su FACP
grafico_zt_AO <- autoplot(zt_AO, color = "red") +
  labs(x = "Observación",
       y = "Valor") +
  theme(panel.grid.minor = element_blank()) +
  autolayer(yt, color = "black")

zt_AO_acf <- ggAcf(zt_AO, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

zt_AO_pacf <- ggAcf(zt_AO, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grafico_zt_AO / (zt_AO_acf + zt_AO_pacf)
```

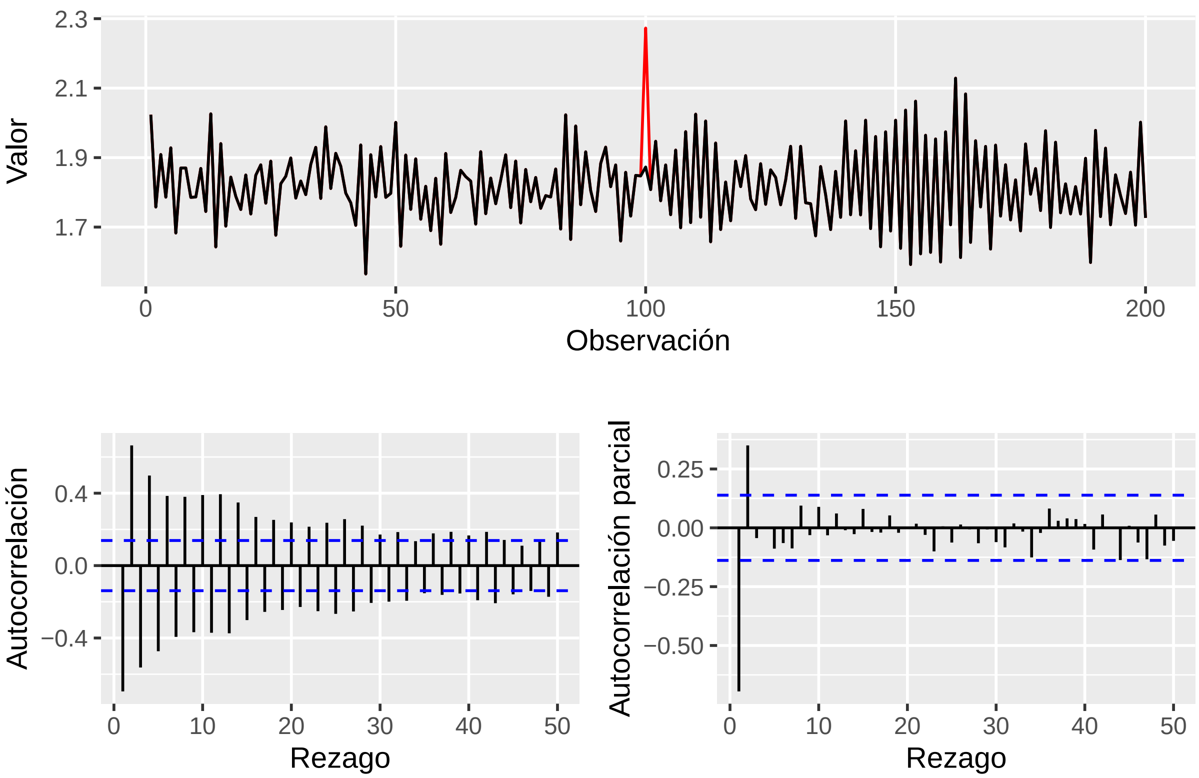


Figura 2: Efecto de la introducción de un dato atípico AO en el proceso AR(2).

2.2. Cambio transitorio (TC)

Los valores atípicos del tipo TC corresponden a cambios transitorios que tienen un impacto máximo al comienzo y luego se diluyen de forma exponencial a medida que transcurre el tiempo. Sea W_{TC} el impacto asociado al punto raro, I_t^h una variable externa que indica su ocurrencia a partir del momento h y $\Psi(L)\varepsilon_t$ un proceso estacionario, el proceso resultante Z_t puede escribirse como:

$$Z_t = W_{TC} I_t^h + \Psi(L)\varepsilon_t$$

La variable I_t^h vale uno en el momento h, y luego decrece exponencialmente a una tasa δ:

$$I_t = \begin{cases} \delta^{t-h} & \text{si } t \geq h \\ 0 & \text{en otro caso} \end{cases}$$

```
# Introducimos un TC a partir de la observación 100
quiebre <- 100
zt_TC <- yt
I <- rep(0, N)
delta <- 0.7
I[quiebre] <- 1
for (i in (quiebre + 1):length(I)) {
  I[i] <- delta * I[i - 1]
}
zt_TC <- 0.4 * I + yt
```

```
# Graficamos la serie con un AO, su FAC y su FACP
grafico_zt_TC <- autoplot(zt_TC, color = "red") +
  labs(x = "Observación",
       y = "Valor") +
  theme(panel.grid.minor = element_blank()) +
  autolayer(yt, color = "black")

zt_TC_acf <- ggAcf(zt_TC, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

zt_TC_pacf <- ggAcf(zt_TC, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grafico_zt_TC / (zt_TC_acf + zt_TC_pacf)
```

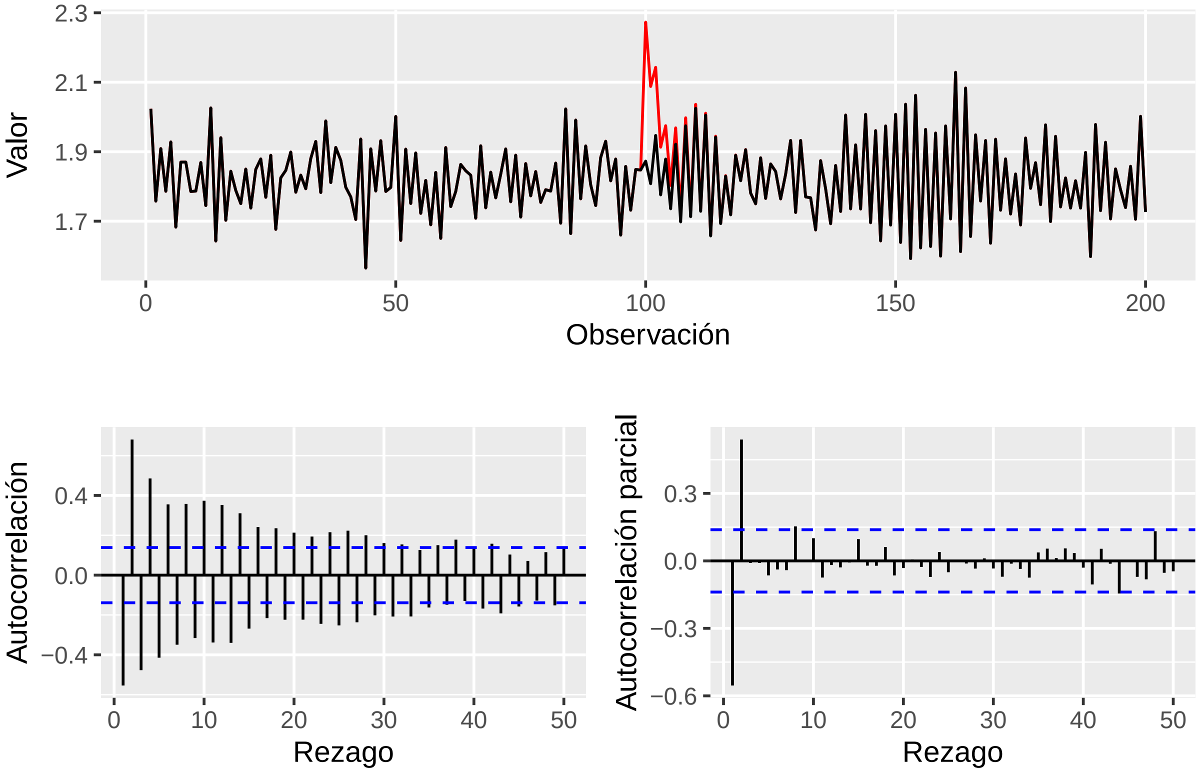


Figura 3: Efecto de la introducción de un dato atípico TC en el proceso AR(2).

2.3. Cambio de nivel (LS)

Cuando a partir de cierta observación se produce un cambio de nivel en la serie, se dice que hay un valor atípico del tipo LS. Sea W_L el impacto asociado al punto raro, S_t^h una variable externa que indica su ocurrencia a partir del momento h y $\Psi(L)\varepsilon_t$ un proceso estacionario, el proceso resultante Z_t puede escribirse como:

$$Z_t = W_L S_t^h + \Psi(L)\varepsilon_t$$

La variable S_t^h es de tipo "escalón". Así, si se supone que el cambio de nivel ocurre en el momento h, la variable indicatriz S_t tendrá la forma:

$$S_t = \begin{cases} 1 & \text{si } t \geq h \\ 0 & \text{en otro caso} \end{cases}$$

A partir de esto, queda claro que la esperanza del proceso será diferente dependiendo de si se lo considera antes o después del cambio de nivel, por lo que el proceso dejará de ser estacionario. Se tiene, entonces:

$$E(Z_t) = \begin{cases} W_L & \text{si } t \geq h \\ 0 & \text{en otro caso} \end{cases}$$

```
# Introducimos un LS a partir de la observación 100
quiebre <- 100
I <- rep(0, N)
I[quiebre:N] <- 1
```

```
zt_LS <- yt
zt_LS <- 0.4 * I + yt
```

```
# Graficamos la serie con un LS, su FAC y su FACP
grafico_zt_LS <- autoplot(zt_LS, color = "red") +
  labs(x = "Observación",
       y = "Valor") +
  theme(panel.grid.minor = element_blank()) +
  autolayer(yt, color = "black")

# Al introducir un cambio de nivel, el proceso deja de tener una media constante y se vuelve no estacio
zt_LS_acf <- ggAcf(zt_LS, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

zt_LS_pacf <- ggAcf(zt_LS, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grafico_zt_LS / (zt_LS_acf + zt_LS_pacf)
```

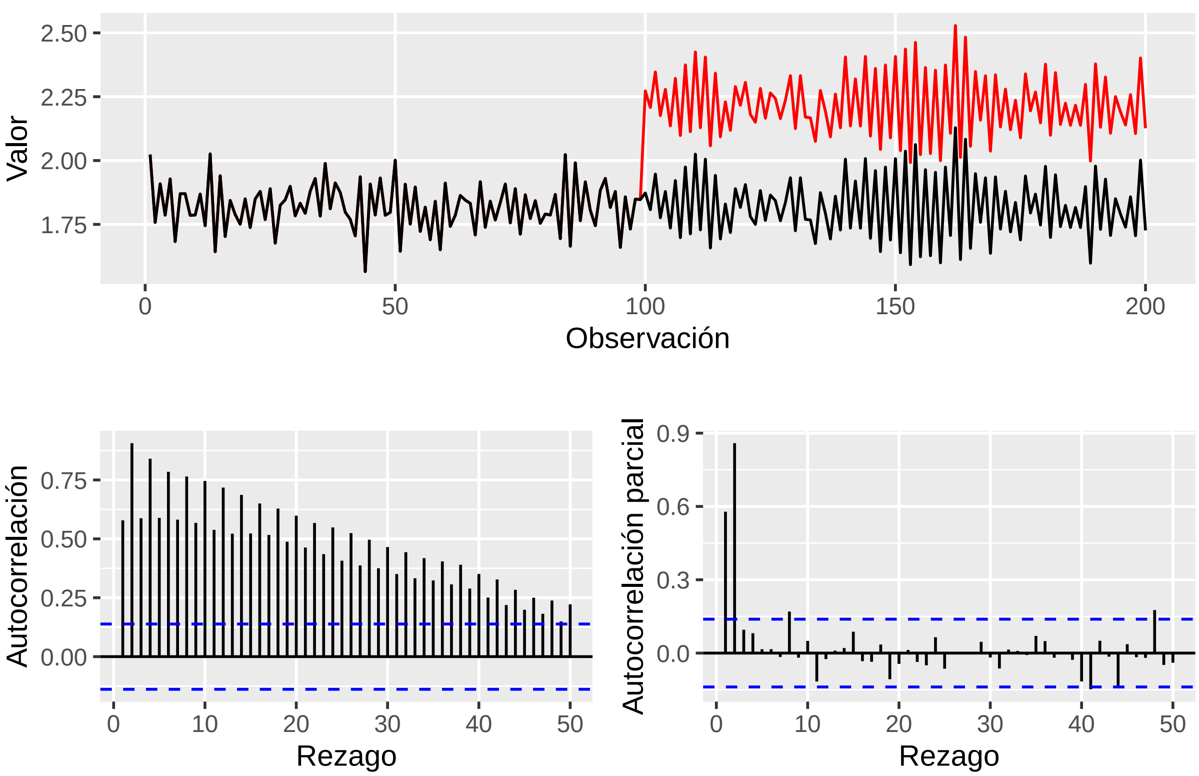


Figura 4: Efecto de la introducción de un dato atípico LS en el proceso AR(2).

3. Estimación y diagnóstico

3.1. Atípico aditivo (AO)

```r
# Estimamos un modelo AR(2)
modelo_AO <- Arima(y = zt_AO,
                   order = c(2, 0, 0),
                   lambda = NULL)

summary(modelo_AO)
```

```
## Series: zt_AO
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##           ar1     ar2    mean
##       -0.4509  0.3621  1.8223
## s.e.   0.0662  0.0666  0.0050
##
## sigma^2 = 0.005999:  log likelihood = 228.86
## AIC=-449.72   AICc=-449.51   BIC=-436.52
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE      MASE
## Training set 9.423771e-05 0.07686838 0.05767618 -0.1662334 3.155508 0.3204652
##                    ACF1
## Training set 0.02399651
```

```r
coefci(modelo_AO)
```

```
##                2.5 %     97.5 %
## ar1       -0.5806966 -0.3211251
## ar2        0.2315723  0.4926275
## intercept  1.8125727  1.8321234
```

```r
coeftest(modelo_AO)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1       -0.4509109  0.0662184  -6.8094 9.798e-12 ***
## ar2        0.3620999  0.0665969   5.4372 5.413e-08 ***
## intercept  1.8223480  0.0049875 365.3820 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Guardamos los residuos del modelo
residuos_AO <- modelo_AO$residuals

# Buscamos el residuo máximo del modelo
max(abs(residuos_AO))
```

```
## [1] 0.4522831
```

```r
which.max(abs(residuos_AO))
```

```
## [1] 100
```

```r
# Graficamos los residuos
grafico_residuos <- residuos_AO %>% autoplot() +
  labs(x = "Fecha",
       y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")

residuos_AO_est <- residuos_AO/sqrt(modelo_AO$sigma2)
grafico_residuos_est <- residuos_AO_est %>%
  autoplot() +
  labs(x = "Fecha",
        y = "Residuos estandarizados") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")

grid.arrange(grafico_residuos, grafico_residuos_est)
```

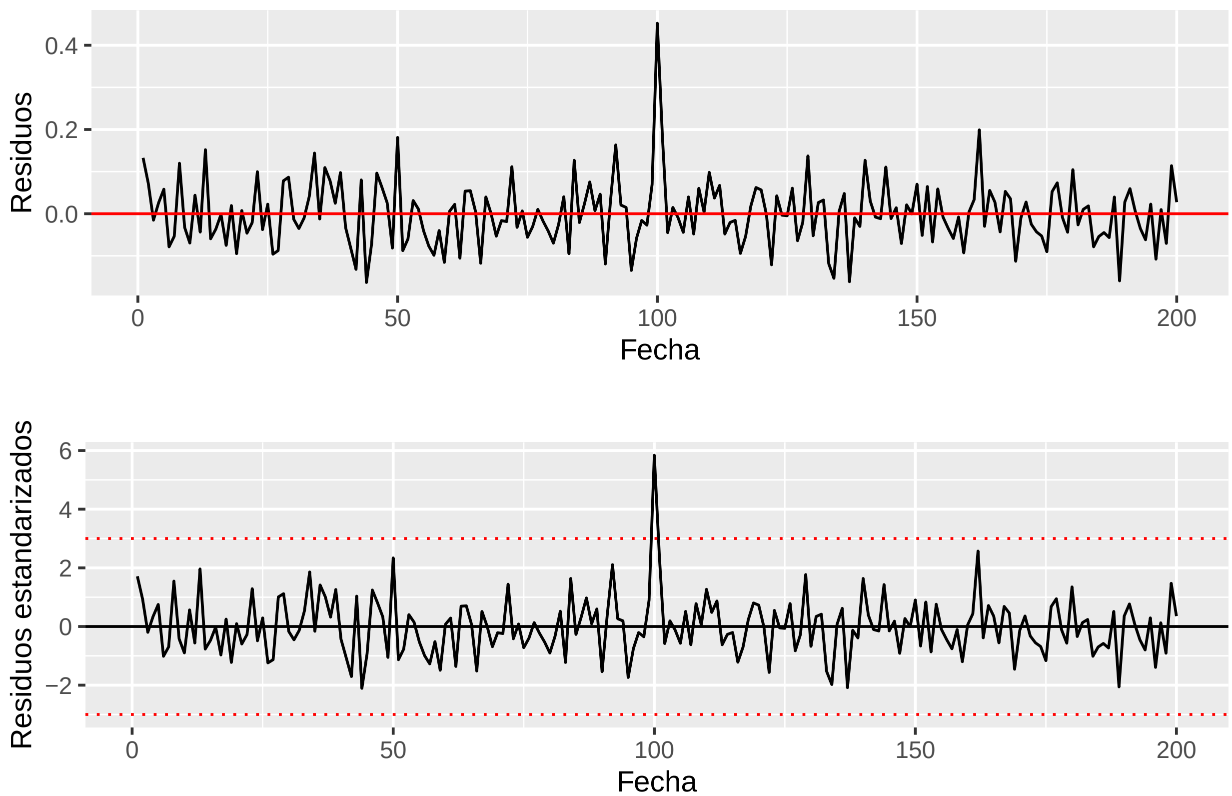


Figura 5: Residuos de un modelo AR(2) para la serie simulada con un AO.

```r
# FAC
residuos_AO_acf <- ggAcf(residuos_AO, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_AO_pacf <- ggAcf(residuos_AO, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_AO_acf, residuos_AO_pacf)
```

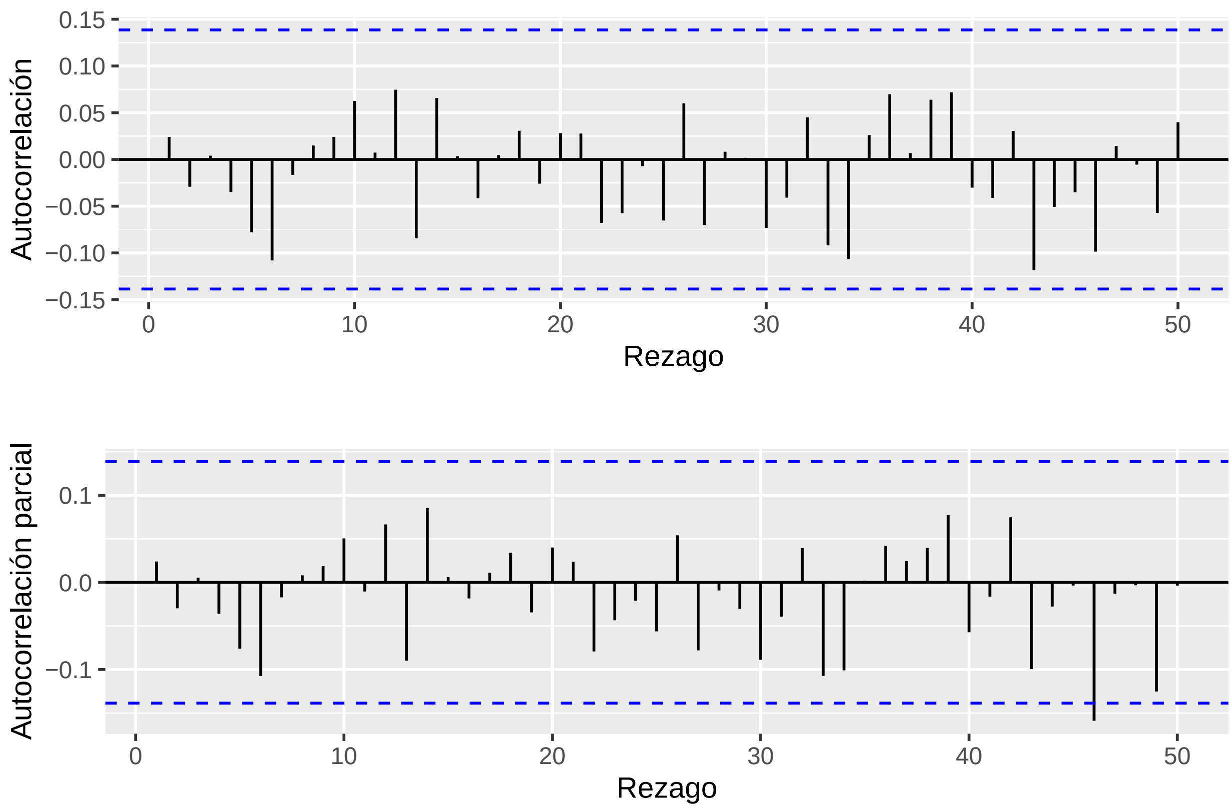


Figura 6: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) para la serie simulada con un AO.

```r
# Test de Ljung-Box
# No se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos_AO,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_AO
## X-squared = 5.2938, df = 8, p-value = 0.7258
```

```r
Box.test(residuos_AO,
         lag = 30,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_AO
## X-squared = 16.145, df = 28, p-value = 0.9636
```

```r
Box.test(residuos_AO,
         lag = 50,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_AO
## X-squared = 34.944, df = 48, p-value = 0.9203
```

```r
# Armamos el QQ-plot de los residuos
ggplot(residuos_AO, aes(sample = residuos_AO)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

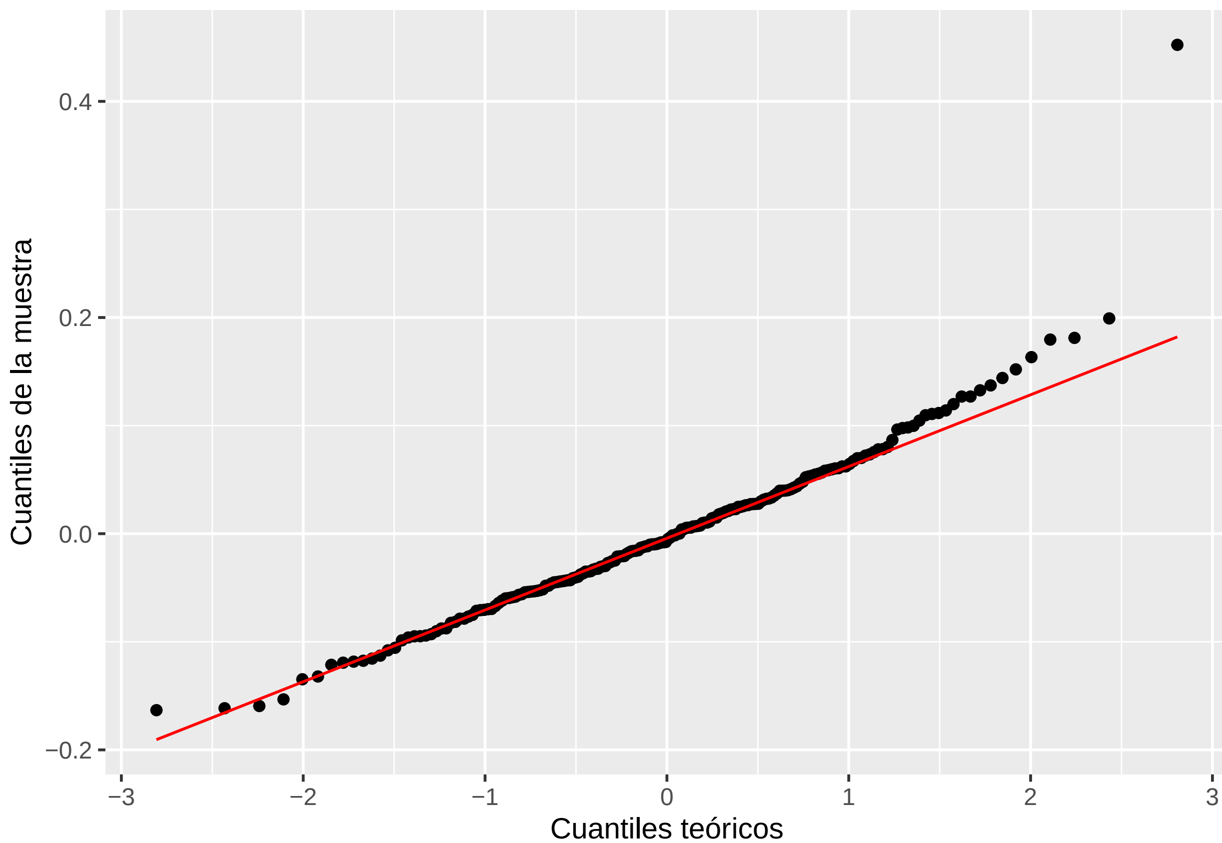


Figura 7: QQ-plot de los residuos de un modelo AR(2) para la serie simulada con un AO.

```
# Hacemos un histograma de los residuos
ggplot(data = residuos_AO) +
  geom_histogram(aes(x = residuos_AO,y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_AO),
                            sd = sd(residuos_AO)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

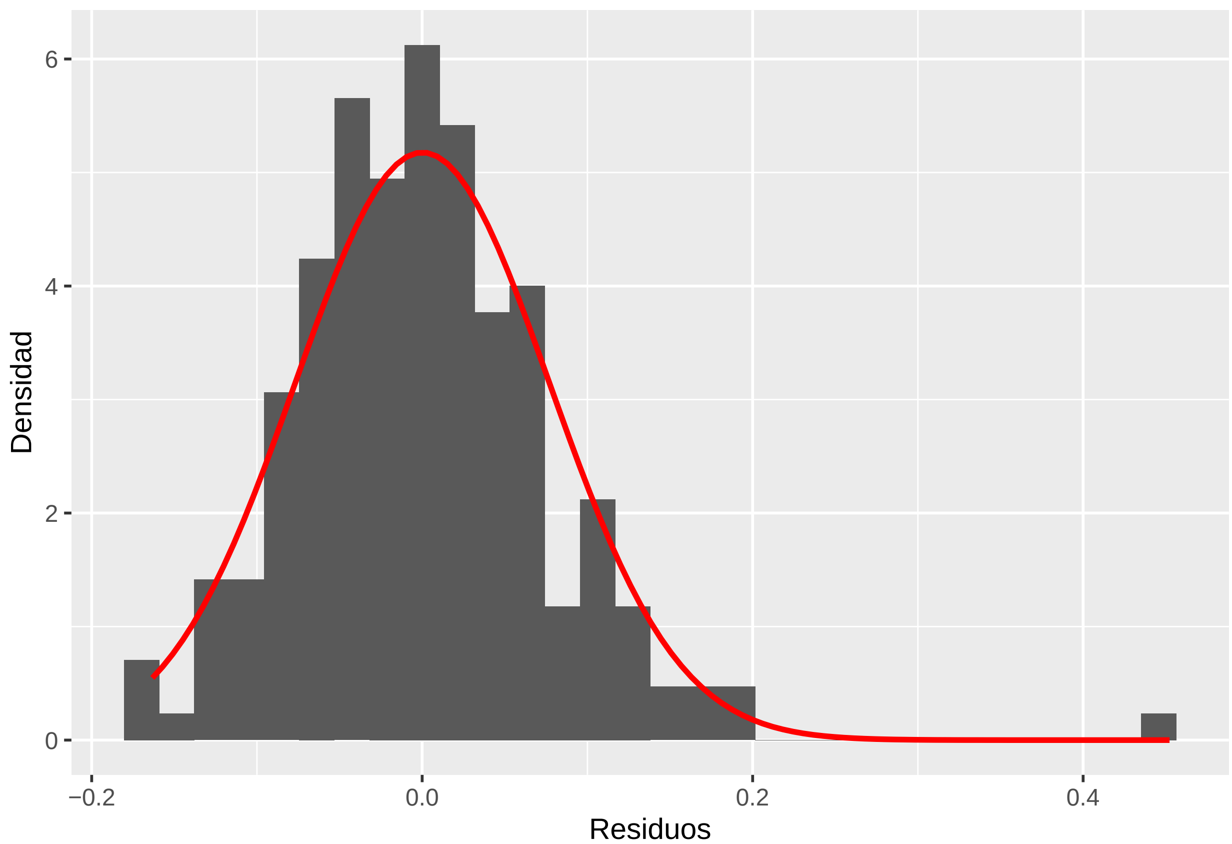


Figura 8: Histograma de los residuos de un modelo AR(2) para la serie simulada con un AO. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

```r
# Tests de Shapiro y Jarque-Bera
# Se rechaza la hipótesis nula de normalidad dado que hay un outlier
shapiro.test(residuos_AO)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_AO
## W = 0.94474, p-value = 6.06e-07
```

```r
JarqueBera.test(residuos_AO)
```

```
##
##  Jarque Bera Test
##
## data:  residuos_AO
## X-squared = 253.73, df = 2, p-value < 2.2e-16
##
##
##  Skewness
##
```

```
## data:  residuos_AO
## statistic = 1.1265, p-value = 7.822e-11
##
##
##  Kurtosis
##
## data:  residuos_AO
## statistic = 8.037, p-value < 2.2e-16
```

3.2. Cambio transitorio (TC)

```r
# Estimamos un modelo AR(2)
modelo_TC <- Arima(y = zt_TC,
                   order = c(2, 0, 0),
                   lambda = NULL)

summary(modelo_TC)
```

```
## Series: zt_TC
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##           ar1     ar2    mean
##       -0.2534  0.5532  1.8275
## s.e.   0.0588  0.0591  0.0083
##
## sigma^2 = 0.006941:  log likelihood = 214.19
## AIC=-420.39   AICc=-420.18   BIC=-407.2
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE      MASE
## Training set -0.00013461 0.08268582 0.06188069 -0.2052263 3.372387 0.3470175
##                    ACF1
## Training set 0.0137427
```

```r
coefci(modelo_TC)
```

```
##                 2.5 %     97.5 %
## ar1       -0.3687407 -0.1381249
## ar2        0.4372862  0.6690467
## intercept  1.8112107  1.8437528
```

```r
coeftest(modelo_TC)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1       -0.2534328  0.0588317  -4.3078 1.649e-05 ***
## ar2        0.5531664  0.0591237   9.3561 < 2.2e-16 ***
```

```
## intercept  1.8274817  0.0083017 220.1338 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Guardamos los residuos del modelo
residuos_TC <- modelo_TC$residuals

# Buscamos el residuo máximo del modelo
max(abs(residuos_TC))
```

```
## [1] 0.4386786
```

```r
which.max(abs(residuos_TC))
```

```
## [1] 100
```

```r
# Graficamos los residuos
grafico_residuos <- residuos_TC %>% autoplot() +
  labs(x = "Fecha",
      y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")

residuos_TC_est <- residuos_TC/sqrt(modelo_TC$sigma2)
grafico_residuos_est <- residuos_TC_est %>%
  autoplot() +
  labs(x = "Fecha",
       y = "Residuos estandarizados") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")

grid.arrange(grafico_residuos, grafico_residuos_est)
```

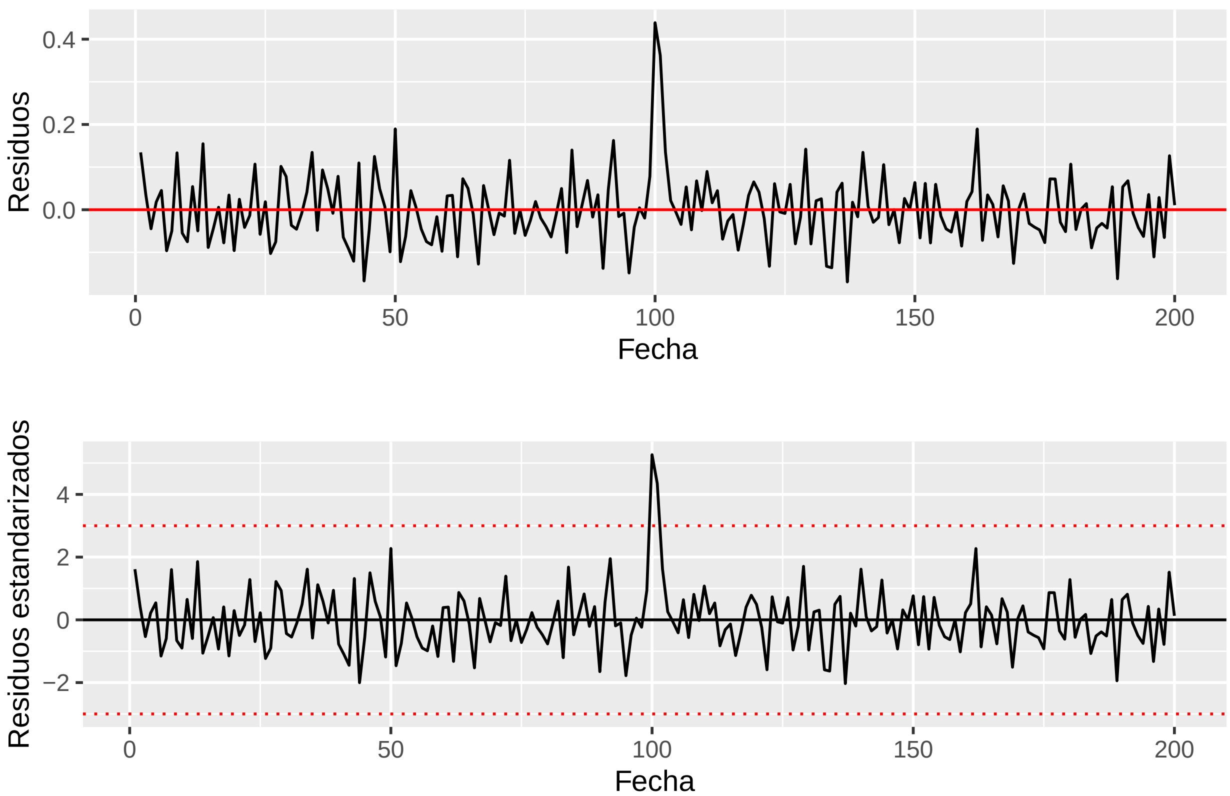
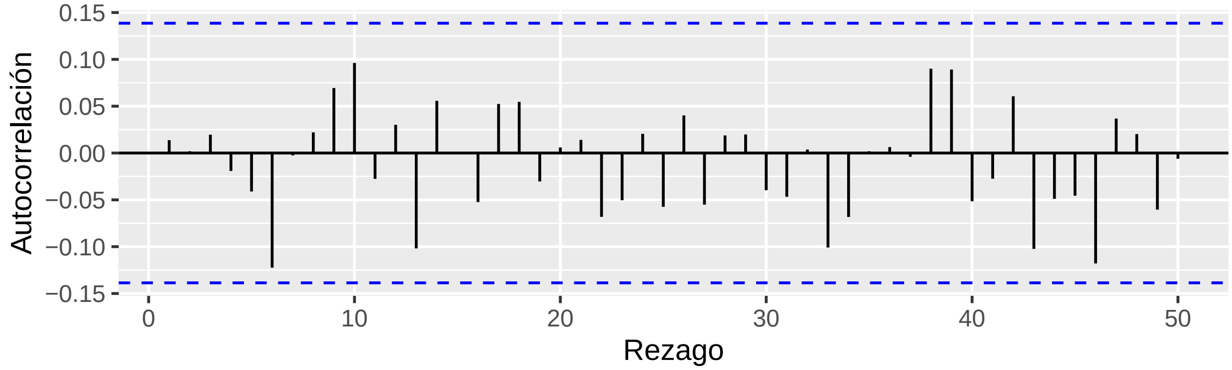


Figura 9: Residuos de un modelo AR(2) para la serie simulada con un TC.

```
# FAC
residuos_TC_acf <- ggAcf(residuos_TC, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_TC_pacf <- ggAcf(residuos_TC, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_TC_acf, residuos_TC_pacf)
```



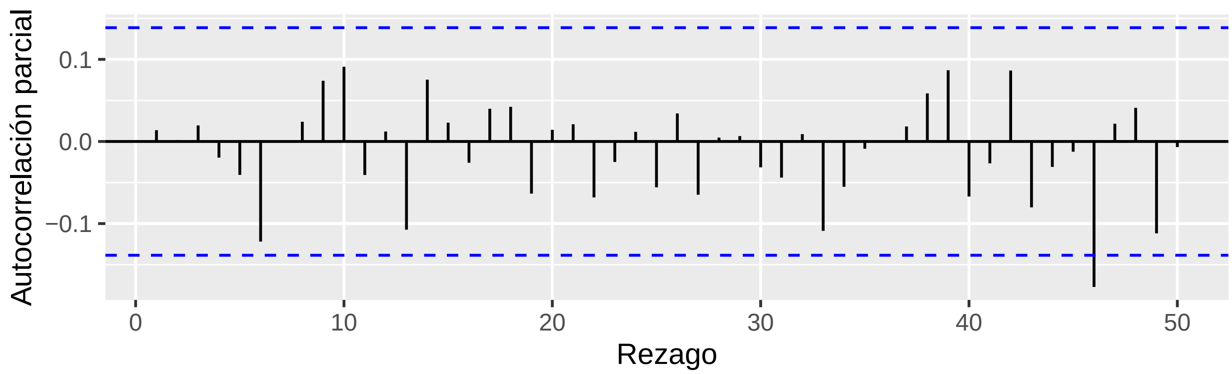


Figura 10: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) para la serie simulada con un TC.

```
# Test de Ljung-Box
# No se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos_TC,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_TC
## X-squared = 6.7533, df = 8, p-value = 0.5635
```

```
Box.test(residuos_TC,
         lag = 30,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
```

```
## data:  residuos_TC
## X-squared = 16.281, df = 28, p-value = 0.9615
```

```r
Box.test(residuos_TC,
         lag = 50,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_TC
## X-squared = 35.187, df = 48, p-value = 0.9157
```

```r
# Armamos el QQ-plot de los residuos
ggplot(residuos_TC, aes(sample = residuos_TC)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

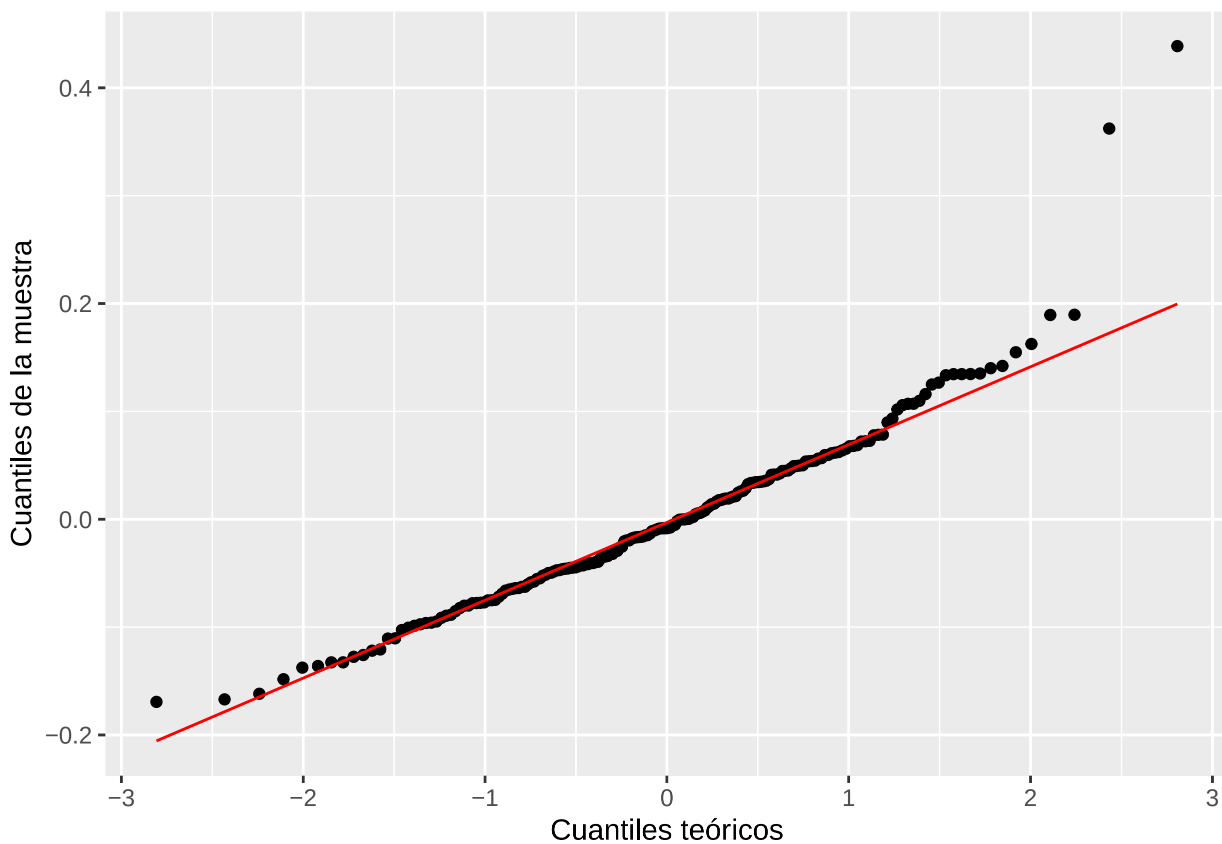


Figura 11: QQ-plot de los residuos de un modelo AR(2) para la serie simulada con un TC.

```r
# Hacemos un histograma de los residuos
ggplot(data = residuos_TC) +
  geom_histogram(aes(x = residuos_TC,y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_TC),
                            sd = sd(residuos_TC)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

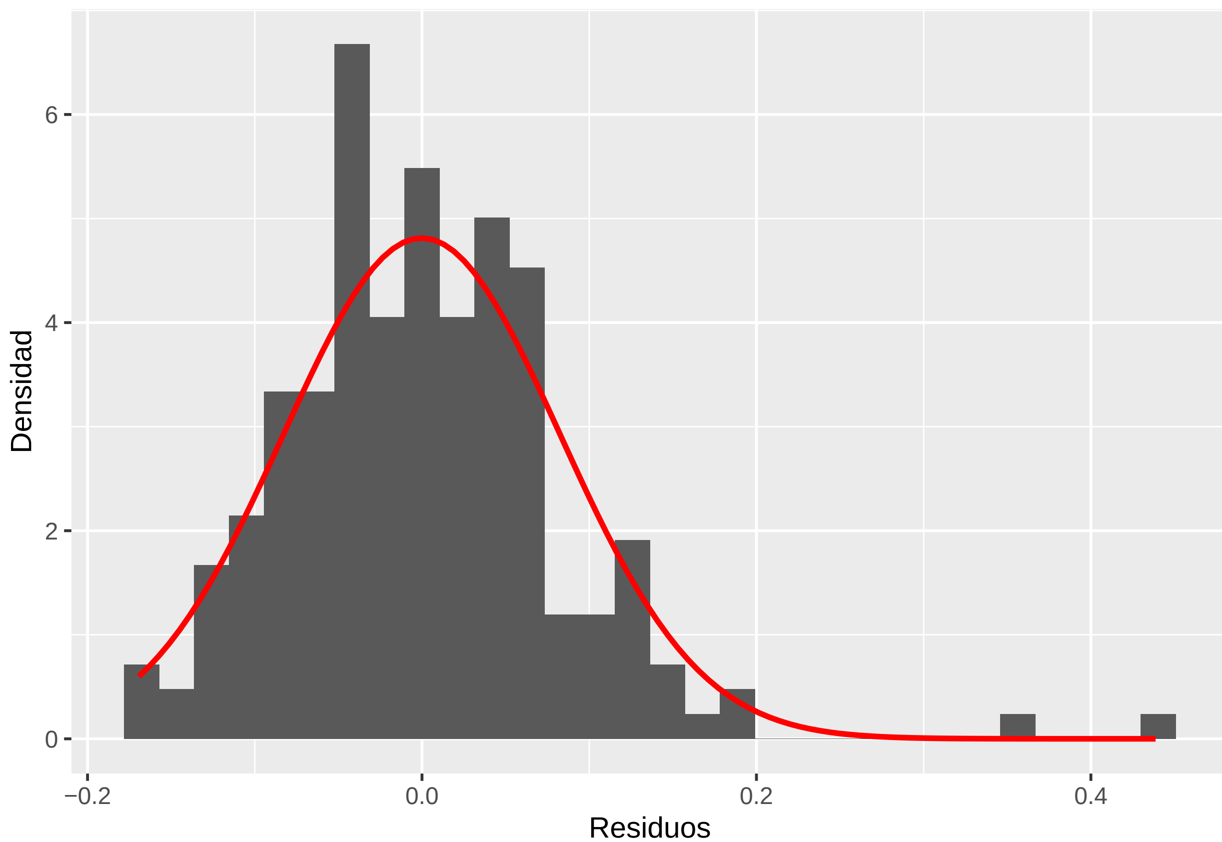


Figura 12: Histograma de los residuos de un modelo AR(2) para la serie simulada. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

```r
# Tests de Shapiro y Jarque-Bera
# Se rechaza la hipótesis nula de normalidad dado que hay un outlier
shapiro.test(residuos_TC)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_TC
## W = 0.93664, p-value = 1.181e-07
```

```r
JarqueBera.test(residuos_TC)
```

```
##
##  Jarque Bera Test
##
## data:  residuos_TC
## X-squared = 211.69, df = 2, p-value < 2.2e-16
##
##
##  Skewness
##
## data:  residuos_TC
## statistic = 1.1996, p-value = 4.325e-12
##
##
##  Kurtosis
##
## data:  residuos_TC
## statistic = 7.4323, p-value < 2.2e-16
```

3.3. Cambio de nivel (LS)

```r
# Estimamos un modelo AR(2)
modelo_LS <- Arima(y = zt_LS,
                   order = c(2, 0, 0),
                   lambda = NULL)

summary(modelo_LS)
```

```
## Series: zt_LS
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2    mean
##       0.0699  0.8759  2.0376
## s.e.  0.0333  0.0336  0.1029
##
## sigma^2 = 0.00837:  log likelihood = 194.39
## AIC=-380.78   AICc=-380.57   BIC=-367.59
##
## Training set error measures:
##                       ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.00196612 0.09079692 0.07072268 -0.113052 3.553996 0.3973935
##                    ACF1
## Training set -0.1065125
```

```r
coefci(modelo_LS)
```

```
##                2.5 %    97.5 %
## ar1      0.004632131 0.1352471
```

```
## ar2       0.810033961 0.9418400
## intercept 1.835951075 2.2391649
```

```r
coeftest(modelo_LS)
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1       0.069940   0.033321   2.099  0.03582 *
## ar2       0.875937   0.033625  26.050  < 2e-16 ***
## intercept 2.037558   0.102863  19.808  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Guardamos los residuos del modelo
residuos_LS <- modelo_LS$residuals

# Buscamos el residuo máximo del modelo
max(abs(residuos_LS))
```

```
## [1] 0.4139718
```

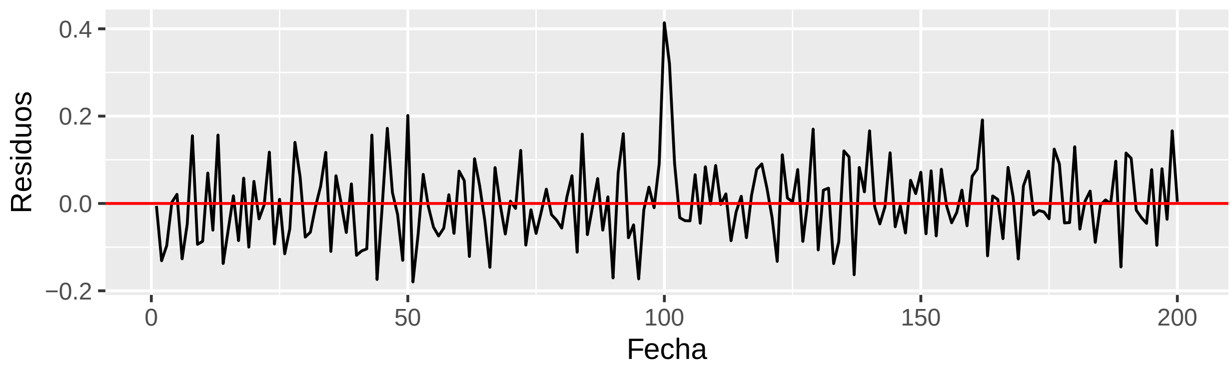
```r
which.max(abs(residuos_LS))
```

```
## [1] 100
```

```r
# Graficamos los residuos
grafico_residuos <- residuos_LS %>% autoplot() +
  labs(x = "Fecha",
       y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")

residuos_LS_est <- residuos_LS/sqrt(modelo_LS$sigma2)
grafico_residuos_est <- residuos_LS_est %>%
  autoplot() +
  labs(x = "Fecha",
       y = "Residuos estandarizados") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")

grid.arrange(grafico_residuos, grafico_residuos_est)
```



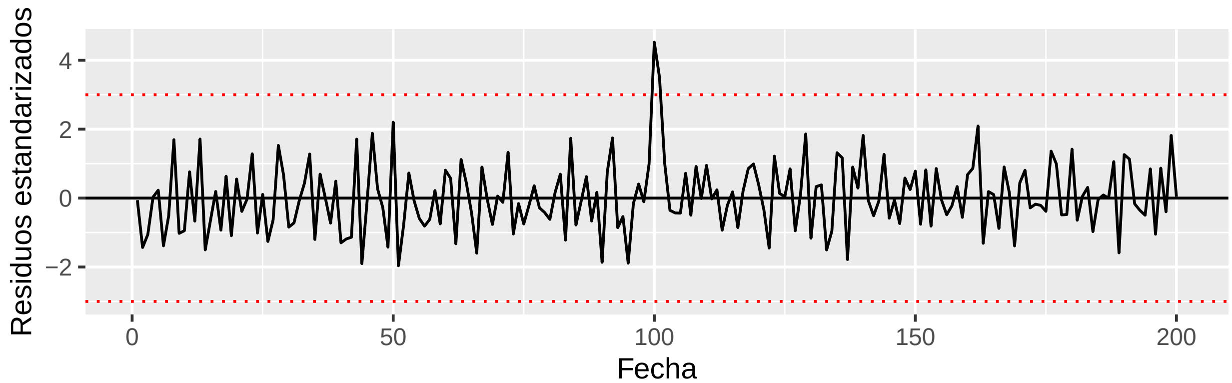


Figura 13: Residuos de un modelo AR(2) para la serie simulada con un LS.

```
# FAC
residuos_LS_acf <- ggAcf(residuos_LS, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_LS_pacf <- ggAcf(residuos_LS, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_LS_acf, residuos_LS_pacf)
```

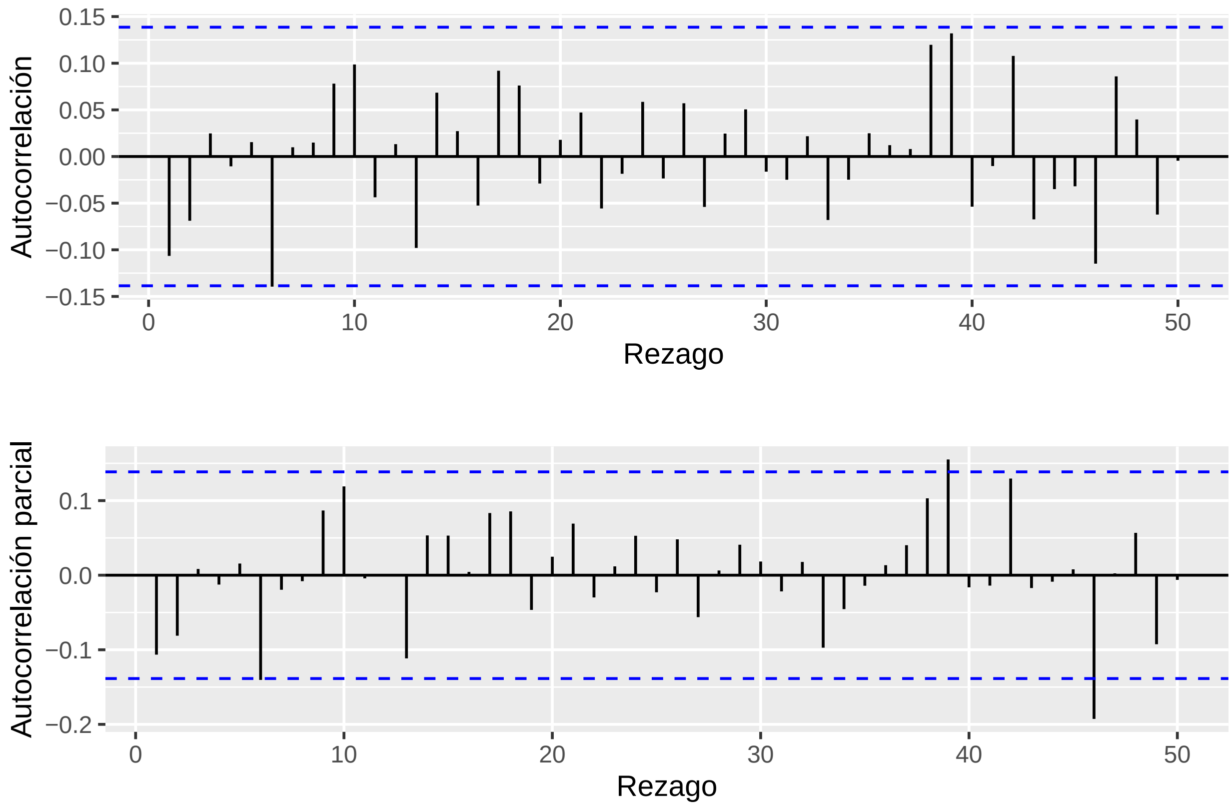


Figura 14: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) para la serie simulada con un LS.

```r
# Test de Ljung-Box
# No se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos_LS,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_LS
## X-squared = 10.945, df = 8, p-value = 0.2048
```

```r
Box.test(residuos_LS,
         lag = 30,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
```

```
## data:  residuos_LS
## X-squared = 23.107, df = 28, p-value = 0.7277
```

```r
Box.test(residuos_LS,
         lag = 50,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_LS
## X-squared = 45.137, df = 48, p-value = 0.5909
```

```r
# Armamos el QQ-plot de los residuos
ggplot(residuos_LS, aes(sample = residuos_LS)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

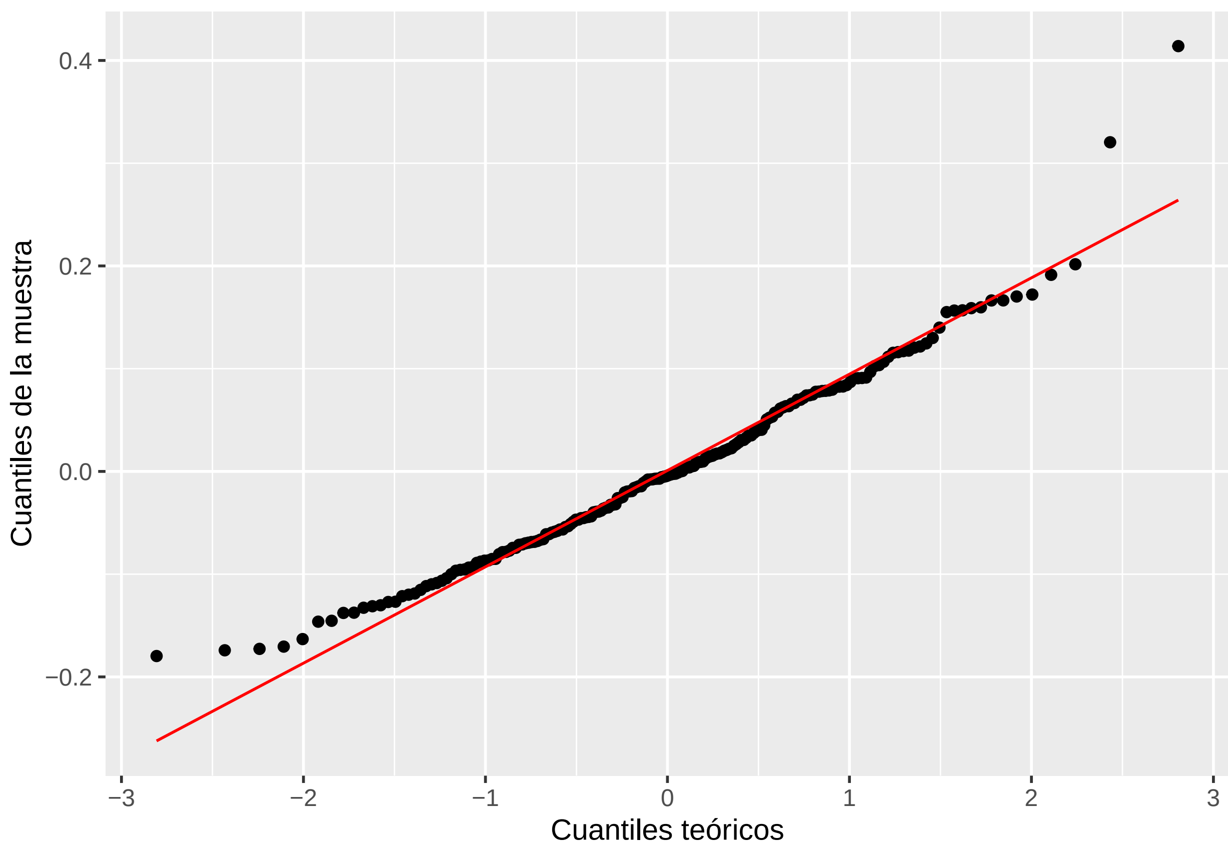


Figura 15: QQ-plot de los residuos de un modelo AR(2) para la serie simulada con un LS.

```r
# Hacemos un histograma de los residuos
ggplot(data = residuos_LS) +
  geom_histogram(aes(x = residuos_LS,y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_LS),
                            sd = sd(residuos_LS)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

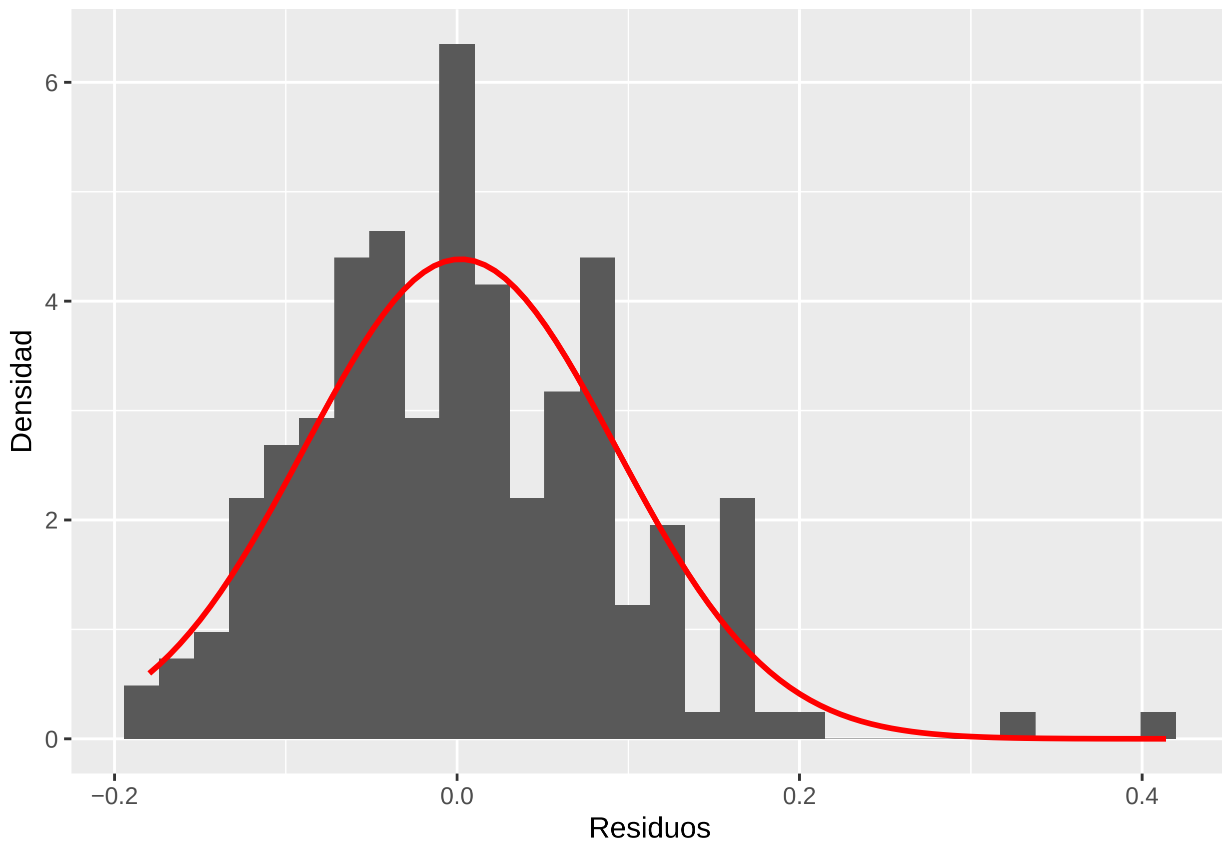


Figura 16: Histograma de los residuos de un modelo AR(2) para la serie simulada. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

```r
# Tests de Shapiro y Jarque-Bera
# Se rechaza la hipótesis nula de normalidad dado que hay un outlier
shapiro.test(residuos_LS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_LS
## W = 0.9701, p-value = 0.0002903
```

```r
JarqueBera.test(residuos_LS)
```

```
##
##  Jarque Bera Test
##
## data:  residuos_LS
## X-squared = 37.932, df = 2, p-value = 5.797e-09
##
##
##  Skewness
##
## data:  residuos_LS
## statistic = 0.70583, p-value = 4.6e-05
##
##
##  Kurtosis
##
## data:  residuos_LS
## statistic = 4.5997, p-value = 3.876e-06
```

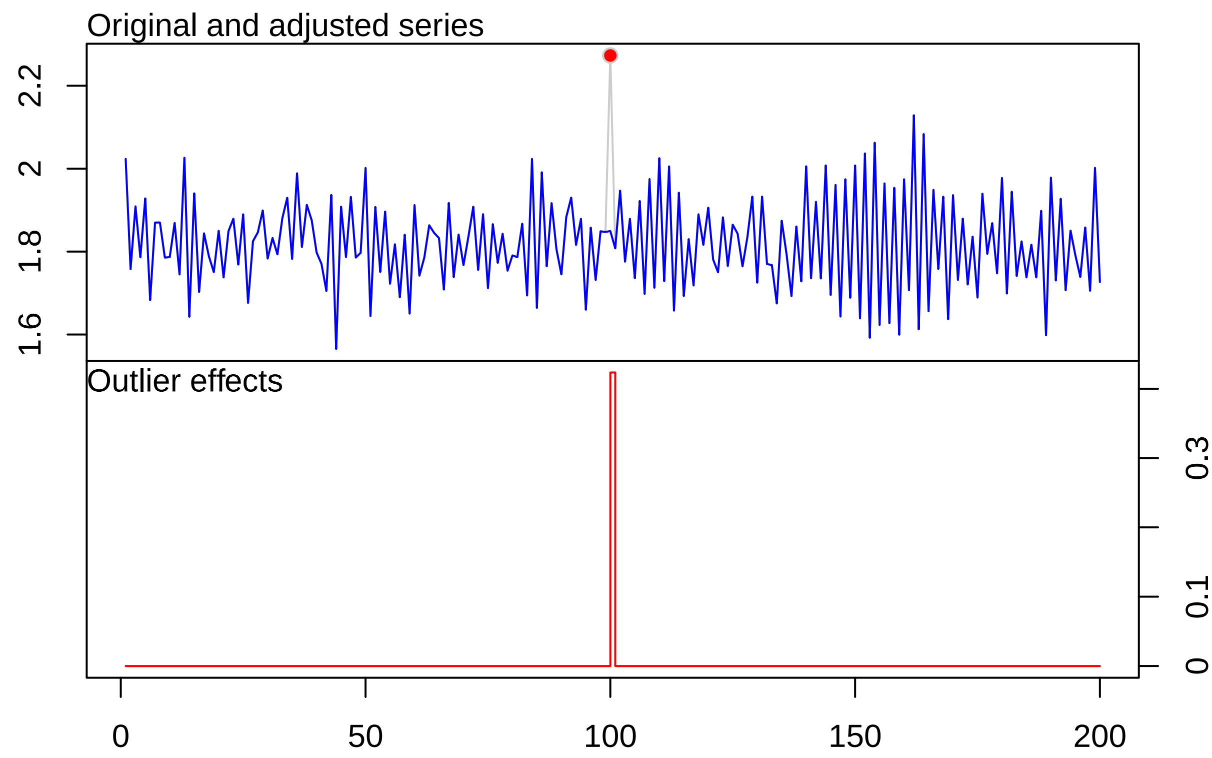
4. Reestimación y diagnóstico

4.1. Atípico aditivo (AO)

```r
# Probamos una función de detección automática de outliers
auto_ao <- tso(zt_AO, tsmethod = "arima",
               args.tsmethod = list(order = c(2, 0, 0),
                                    seasonal = list(order = c(0, 0, 0))))
auto_ao
```

```
##
## Call:
## list(method = NULL)
##
## Coefficients:
##           ar1     ar2  intercept   AO100
##       -0.5406  0.2960     1.8201  0.4235
## s.e.   0.0688  0.0694     0.0039  0.0594
##
## sigma^2 estimated as 0.004729:  log likelihood = 251.08,  aic = -492.15
##
## Outliers:
##   type ind time coefhat tstat
## 1   AO 100  100  0.4235 7.125
```

```r
# Graficamos el efecto del outlier AO
plot.tsoutliers(auto_ao)
```




```r
# Obtenemos la indicatriz para incluir como regresor externo
xreg_AO <- outliers.effects(auto_ao$outliers, length(zt_AO))
head(xreg_AO)
```

```
##      AO100
## [1,]     0
## [2,]     0
## [3,]     0
## [4,]     0
## [5,]     0
## [6,]     0
```

```r
# Creamos la indicatriz para intervenir el modelo
AO <- rep(0, length(zt_AO))
AO[quiebre] <- 1
AO
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```r
# Estimamos un modelo AR(2)
modelo_AO2 <- Arima(y = zt_AO,
                    order = c(2, 0, 0),
                    lambda = NULL,
                    xreg = AO)

summary(modelo_AO2)
```

```
## Series: zt_AO
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##           ar1     ar2  intercept    xreg
##       -0.5406  0.2960     1.8201  0.4235
## s.e.   0.0688  0.0694     0.0039  0.0594
##
## sigma^2 = 0.004826:  log likelihood = 251.08
## AIC=-492.15   AICc=-491.84   BIC=-475.66
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE      MASE
## Training set 0.0001573565 0.06876915 0.05451332 -0.1339013 2.998114 0.3028915
##                     ACF1
## Training set 0.008129161
```

```r
coefci(modelo_AO2)
```

```
##                2.5 %     97.5 %
## ar1       -0.6753633 -0.4058608
## ar2        0.1600010  0.4320579
## intercept  1.8124316  1.8277993
## xreg       0.3069661  0.5399348
```

```r
coeftest(modelo_AO2)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1       -0.5406121  0.0687519  -7.8632 3.744e-15 ***
## ar2        0.2960294  0.0694035   4.2653 1.996e-05 ***
## intercept  1.8201155  0.0039204 464.2666 < 2.2e-16 ***
## xreg       0.4234504  0.0594319   7.1250 1.041e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Guardamos los residuos del modelo
residuos_AO2 <- modelo_AO2$residuals

# Buscamos el residuo máximo del modelo
max(abs(residuos_AO2))
```

```
## [1] 0.2016174
```

```r
which.max(abs(residuos_AO2))
```

```
## [1] 162
```

```r
# Graficamos los residuos
grafico_residuos <- residuos_AO2 %>% autoplot() +
  labs(x = "Fecha",
       y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")

residuos_AO2_est <- residuos_AO2/sqrt(modelo_AO2$sigma2)
grafico_residuos_est <- residuos_AO2_est %>%
  autoplot() +
  labs(x = "Fecha",
        y = "Residuos estandarizados") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")

grid.arrange(grafico_residuos, grafico_residuos_est)
```

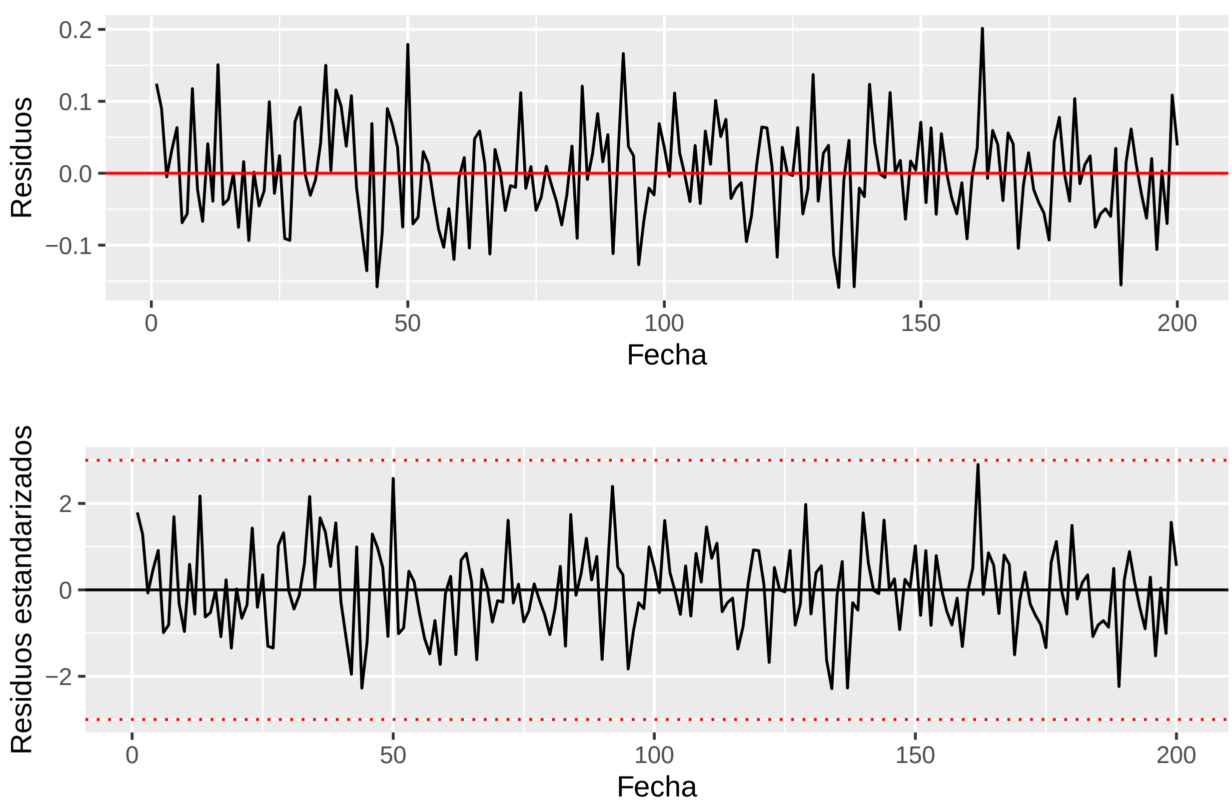


Figura 17: Residuos de un modelo AR(2) intervenido para la serie simulada con un AO.

```r
# FAC
residuos_AO2_acf <- ggAcf(residuos_AO2, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_AO2_pacf <- ggAcf(residuos_AO2, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_AO2_acf, residuos_AO2_pacf)
```

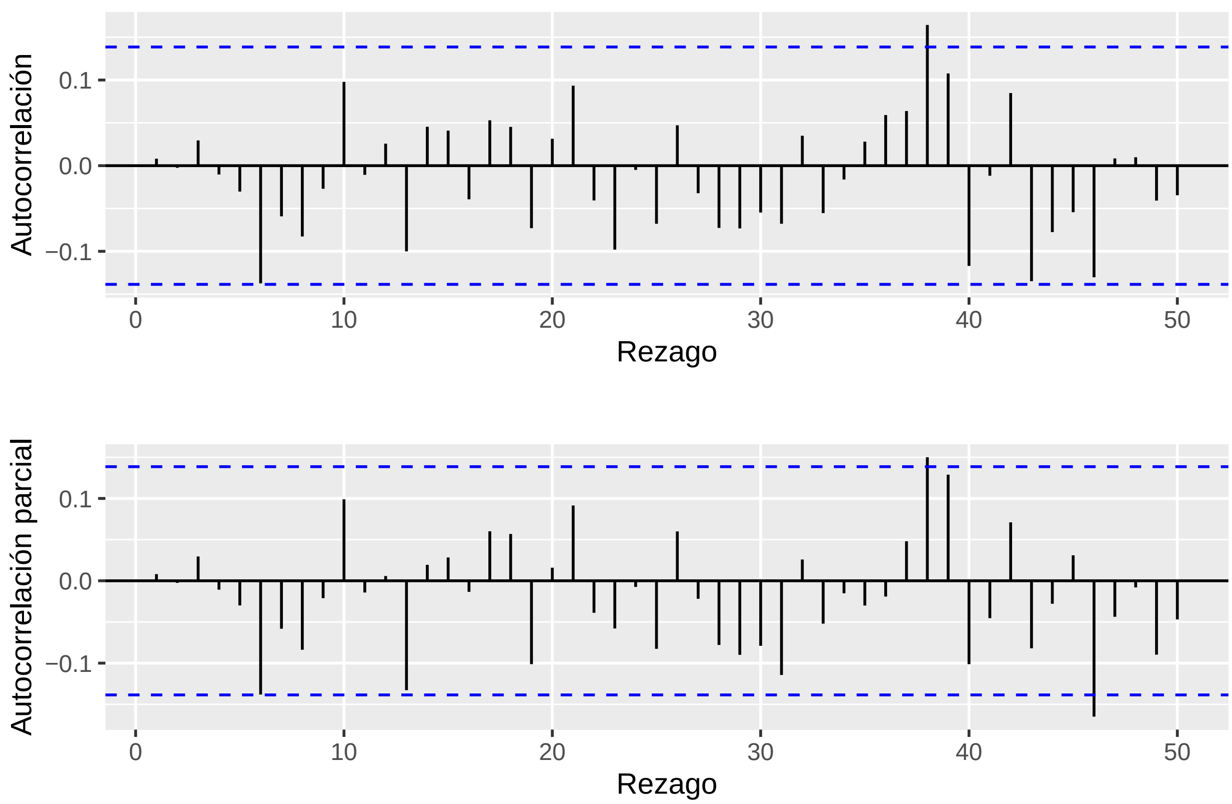


Figura 18: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) intervenido para la serie simulada con un AO.

```r
# Test de Ljung-Box
# No se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos_AO2,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_A02
## X-squared = 8.695, df = 8, p-value = 0.3687
```

```r
Box.test(residuos_A02,
         lag = 30,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_A02
## X-squared = 24.239, df = 28, p-value = 0.6688
```

```r
Box.test(residuos_A02,
         lag = 50,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_A02
## X-squared = 55.753, df = 48, p-value = 0.2062
```

```r
# Armamos el QQ-plot de los residuos
ggplot(residuos_A02, aes(sample = residuos_A02)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

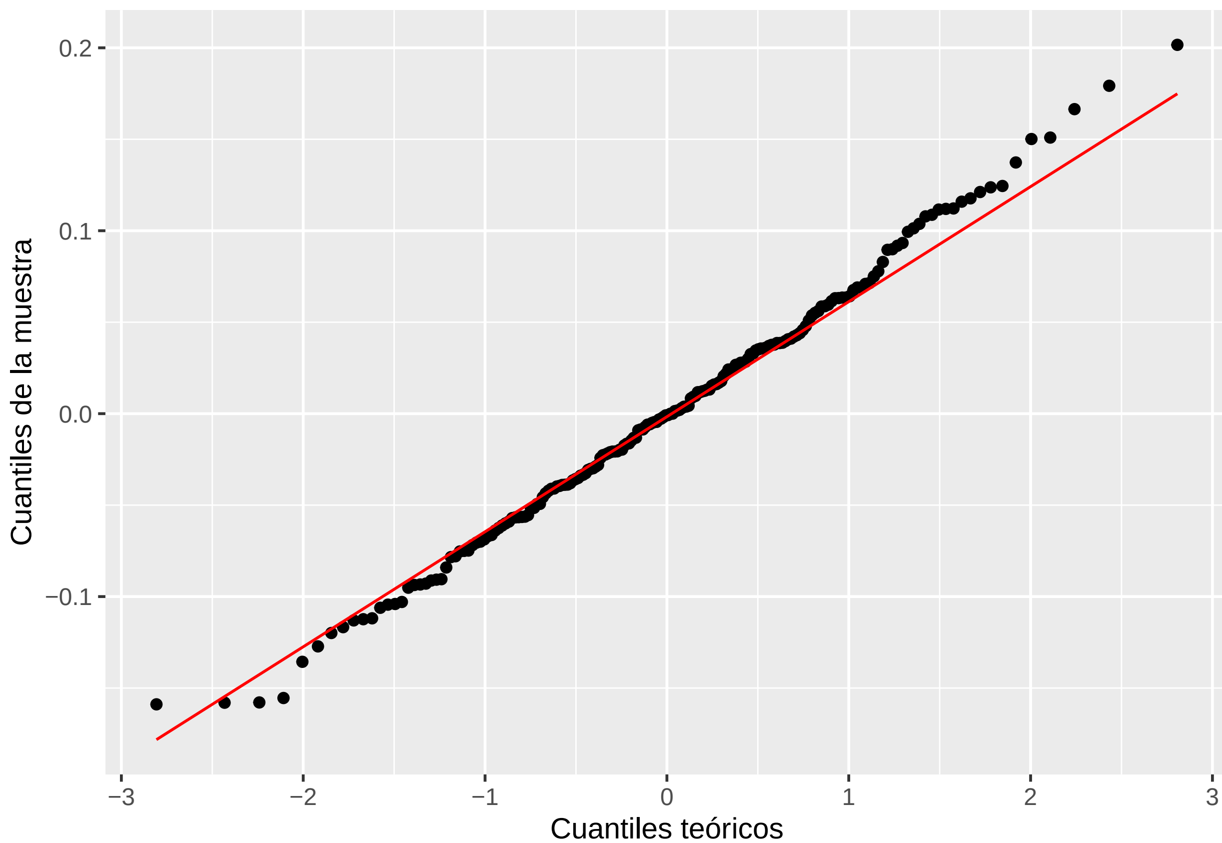


Figura 19: QQ-plot de los residuos de un modelo AR(2) intervenido para la serie simulada con un AO.

```
# Hacemos un histograma de los residuos
ggplot(data = residuos_AO2) +
  geom_histogram(aes(x = residuos_AO2,y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_AO2),
                            sd = sd(residuos_AO2)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

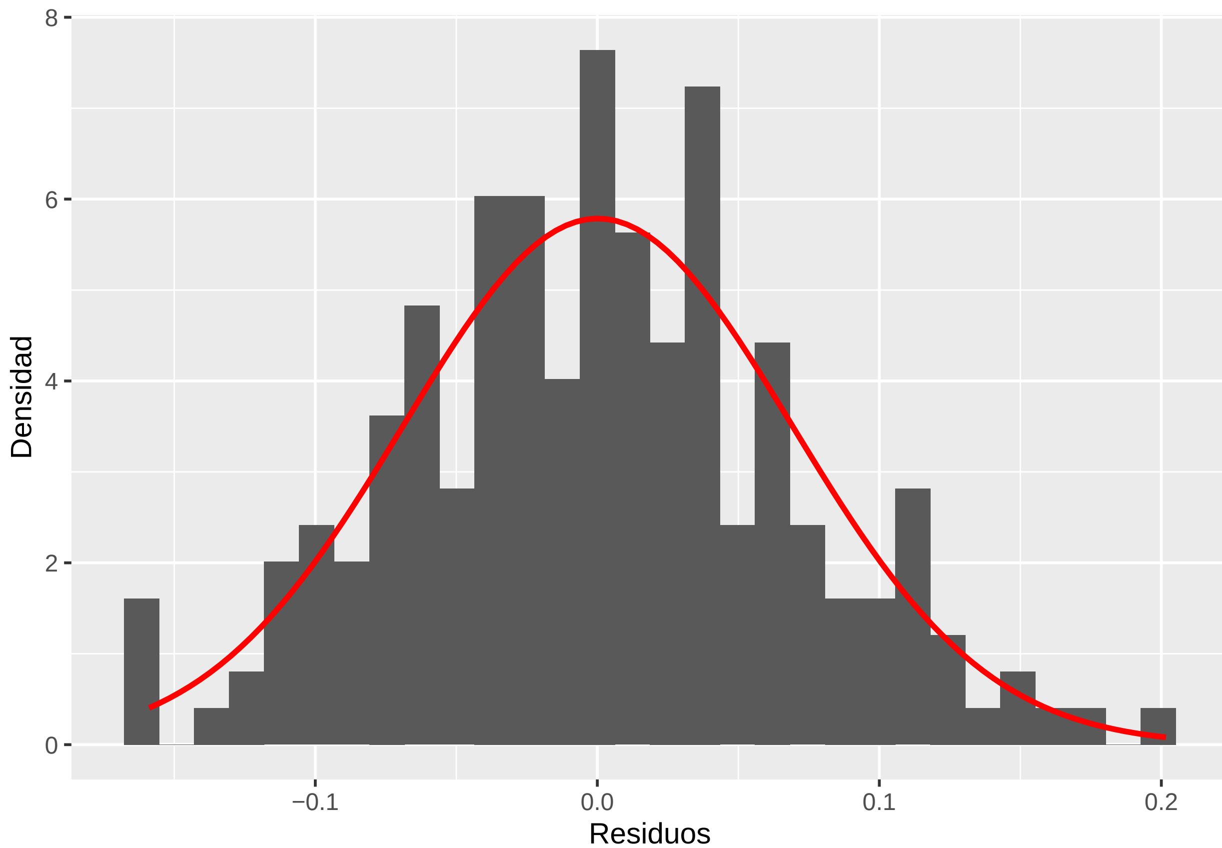


Figura 20: Histograma de los residuos de un modelo AR(2) intervenido para la serie simulada con un AO. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

```
# Tests de Shapiro y Jarque-Bera
# No se rechaza la hipótesis nula de normalidad
shapiro.test(residuos_AO2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_AO2
## W = 0.99532, p-value = 0.7962
```

```
JarqueBera.test(residuos_AO2)
```

```
##
##  Jarque Bera Test
##
## data:  residuos_AO2
## X-squared = 0.80651, df = 2, p-value = 0.6681
##
##
##  Skewness
##
```

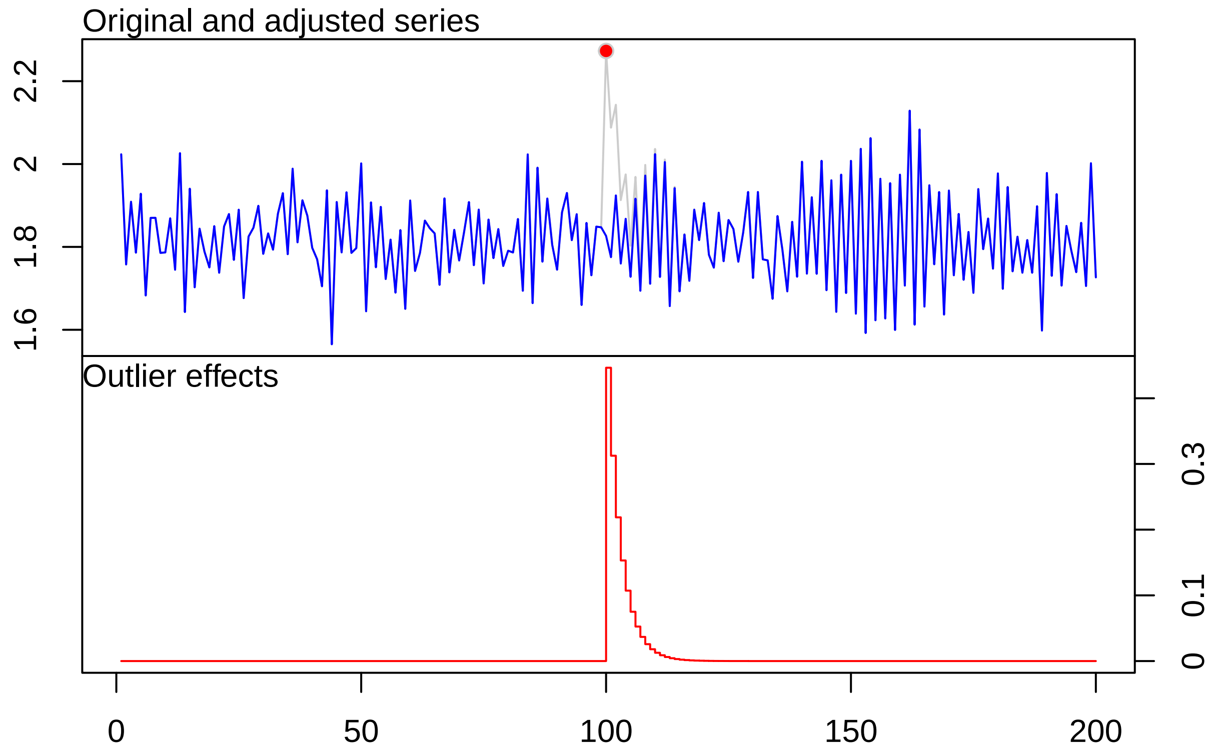
```
## data:  residuos_AO2
## statistic = 0.15315, p-value = 0.3766
##
##
##  Kurtosis
##
## data:  residuos_AO2
## statistic = 2.9455, p-value = 0.8751
```

4.2. Cambio transitorio (TC)

```r
# Probamos una función de detección automática de outliers
auto_tc <- tso(zt_TC, tsmethod = "arima",
               args.tsmethod = list(order = c(2, 0, 0),
                                    seasonal = list(order = c(0, 0, 0))))
auto_tc
```

```
##
## Call:
## list(method = NULL)
##
## Coefficients:
##           ar1     ar2  intercept   TC100
##       -0.5506  0.2857     1.8194  0.4465
## s.e.   0.0687  0.0693     0.0039  0.0388
##
## sigma^2 estimated as 0.0047:  log likelihood = 251.69,  aic = -493.39
##
## Outliers:
##   type ind time coefhat tstat
## 1   TC 100  100  0.4465  11.5
```

```r
# Graficamos el efecto del outlier TC
plot.tsoutliers(auto_tc)
```




```
# Obtenemos la indicatriz para incluir como regresor externo
xreg_TC <- outliers.effects(auto_tc$outliers, length(zt_TC))
head(xreg_TC)
```

```
##      TC100
## [1,]     0
## [2,]     0
## [3,]     0
## [4,]     0
## [5,]     0
## [6,]     0
```

```
# Creamos la indicatriz para intervenir el modelo
TC <- rep(0, length(zt_TC))
delta <- 0.7
TC[quiebre] <- 1
for (i in (quiebre + 1):length(TC)) {
  TC[i] <- delta * TC[i - 1]
}
round(TC,2)
```

```
##   [1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [16] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [31] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [46] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```
##  [61] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [76] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [91] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.70 0.49 0.34 0.24 0.17
## [106] 0.12 0.08 0.06 0.04 0.03 0.02 0.01 0.01 0.01 0.00 0.00 0.00 0.00 0.00 0.00
## [121] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [136] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [151] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [166] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [181] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [196] 0.00 0.00 0.00 0.00 0.00
```

```r
# Estimamos un modelo AR(2)
modelo_TC2 <- Arima(y = zt_TC,
                    order = c(2, 0, 0),
                    lambda = NULL,
                    xreg = TC)

summary(modelo_TC2)
```

```
## Series: zt_TC
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##           ar1     ar2  intercept    xreg
##       -0.5506  0.2857     1.8194  0.4465
## s.e.   0.0687  0.0693     0.0039  0.0388
##
## sigma^2 = 0.004796:  log likelihood = 251.69
## AIC=-493.39   AICc=-493.08   BIC=-476.9
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE      MASE
## Training set 0.0001658699 0.06855692 0.05441391 -0.1313324 2.987752 0.3051449
##                     ACF1
## Training set 0.009479703
```

```r
coefci(modelo_TC2)
```

```
##                2.5 %     97.5 %
## ar1       -0.6853190 -0.4159589
## ar2        0.1499958  0.4214962
## intercept  1.8118210  1.8270689
## xreg       0.3703493  0.5225574
```

```r
coeftest(modelo_TC2)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1       -0.5506389  0.0687156  -8.0133 1.117e-15 ***
## ar2        0.2857460  0.0692616   4.1256 3.698e-05 ***
```

```
## intercept  1.8194450  0.0038898 467.7429 < 2.2e-16 ***
## xreg       0.4464534  0.0388293  11.4978 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Guardamos los residuos del modelo
residuos_TC2 <- modelo_TC2$residuals

# Buscamos el residuo máximo del modelo
max(abs(residuos_TC2))
```

```
## [1] 0.2029134
```

```r
which.max(abs(residuos_TC2))
```

```
## [1] 162
```

```r
# Graficamos los residuos
grafico_residuos <- residuos_TC2 %>% autoplot() +
  labs(x = "Fecha",
      y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")

residuos_TC2_est <- residuos_TC2/sqrt(modelo_TC2$sigma2)
grafico_residuos_est <- residuos_TC2_est %>%
  autoplot() +
  labs(x = "Fecha",
       y = "Residuos estandarizados") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")

grid.arrange(grafico_residuos, grafico_residuos_est)
```

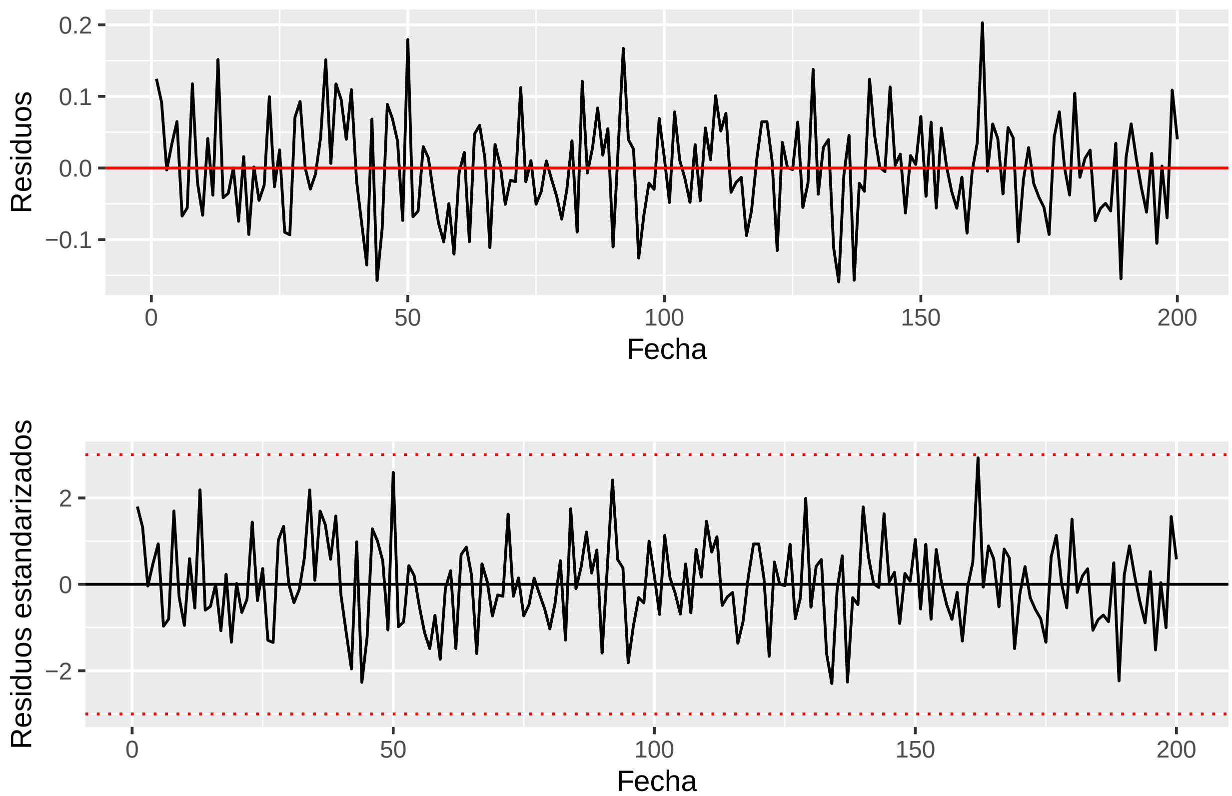


Figura 21: Residuos de un modelo AR(2) intervenido para la serie simulada con un TC.

```r
# FAC
residuos_TC2_acf <- ggAcf(residuos_TC2, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_TC2_pacf <- ggAcf(residuos_TC2, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_TC2_acf, residuos_TC2_pacf)
```

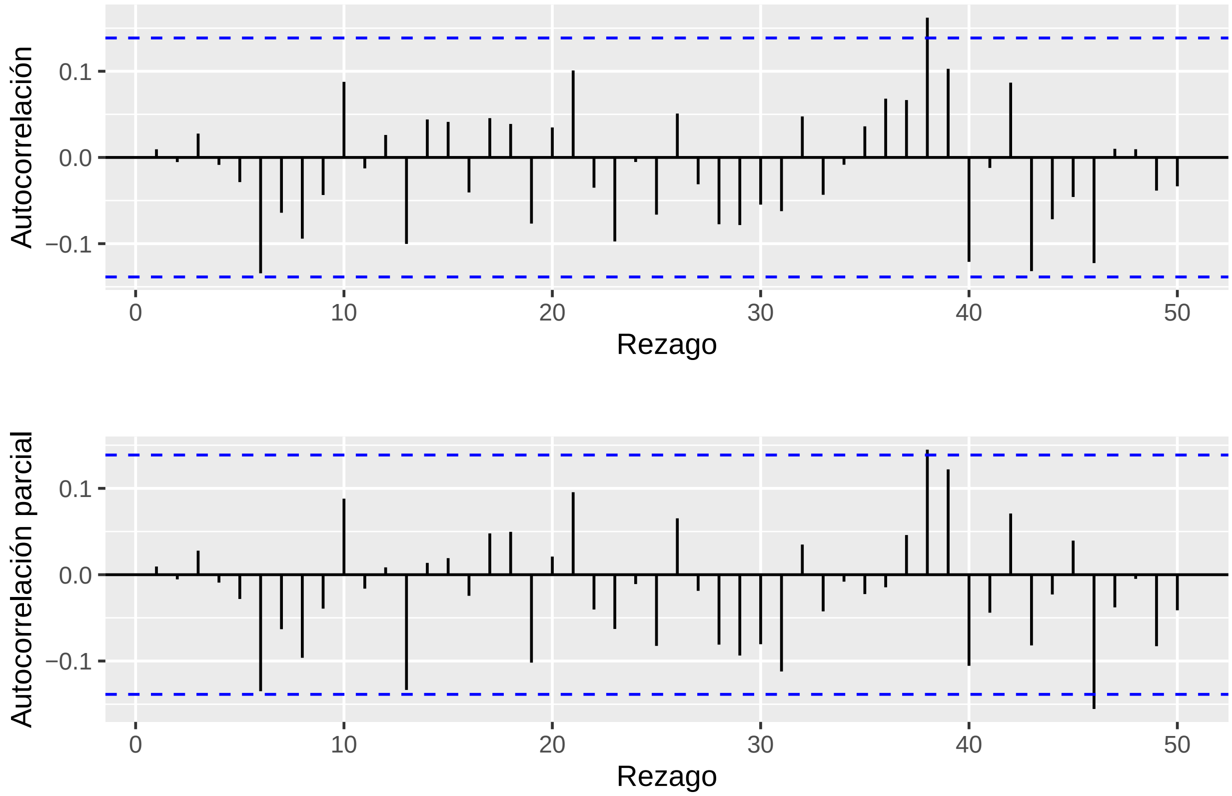


Figura 22: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) intervenido para la serie simulada con un TC.

```
# Test de Ljung-Box
# No se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos_TC2,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_TC2
## X-squared = 8.8907, df = 8, p-value = 0.3516
```

```
Box.test(residuos_TC2,
         lag = 30,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
```

```
## data:  residuos_TC2
## X-squared = 24.944, df = 28, p-value = 0.6309
```

```r
Box.test(residuos_TC2,
         lag = 50,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_TC2
## X-squared = 55.356, df = 48, p-value = 0.2169
```

```r
# Armamos el QQ-plot de los residuos
ggplot(residuos_TC2, aes(sample = residuos_TC2)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

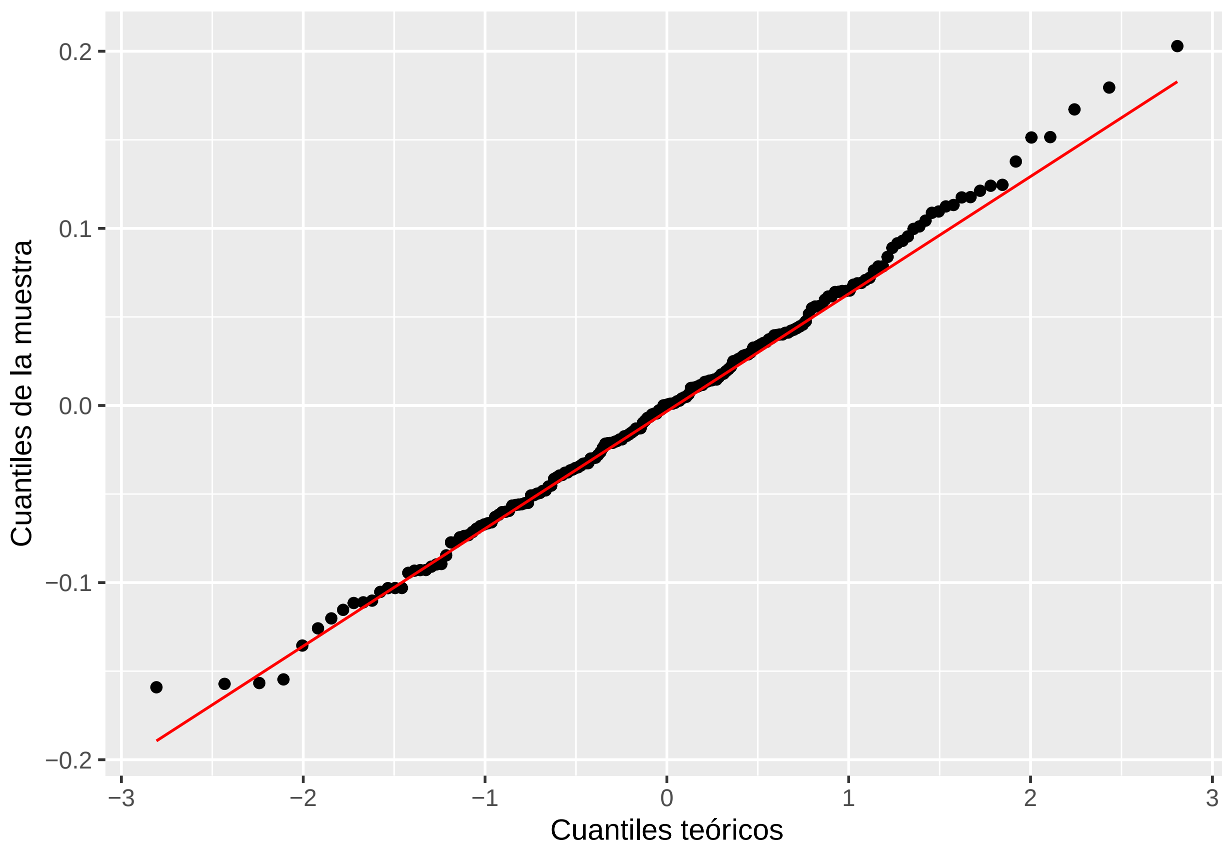


Figura 23: QQ-plot de los residuos de un modelo AR(2) intervenido para la serie simulada con un TC.

```r
# Hacemos un histograma de los residuos
ggplot(data = residuos_TC2) +
  geom_histogram(aes(x = residuos_TC2,y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_TC2),
                            sd = sd(residuos_TC2)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

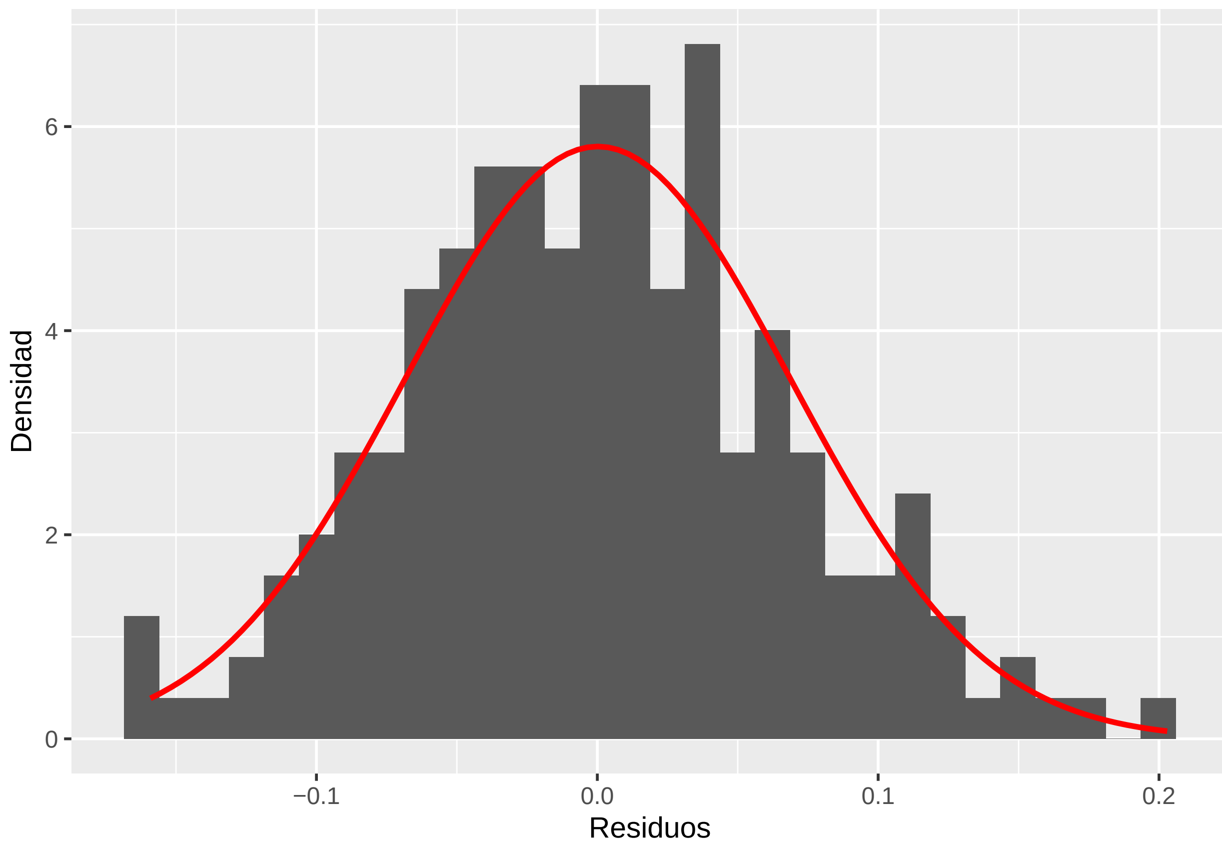


Figura 24: Histograma de los residuos de un modelo AR(2) intervenido para la serie simulada con un TC. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

```r
# Tests de Shapiro y Jarque-Bera
# No se rechaza la hipótesis nula de normalidad
shapiro.test(residuos_TC2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_TC2
## W = 0.99552, p-value = 0.8226
```

```r
JarqueBera.test(residuos_TC2)
```

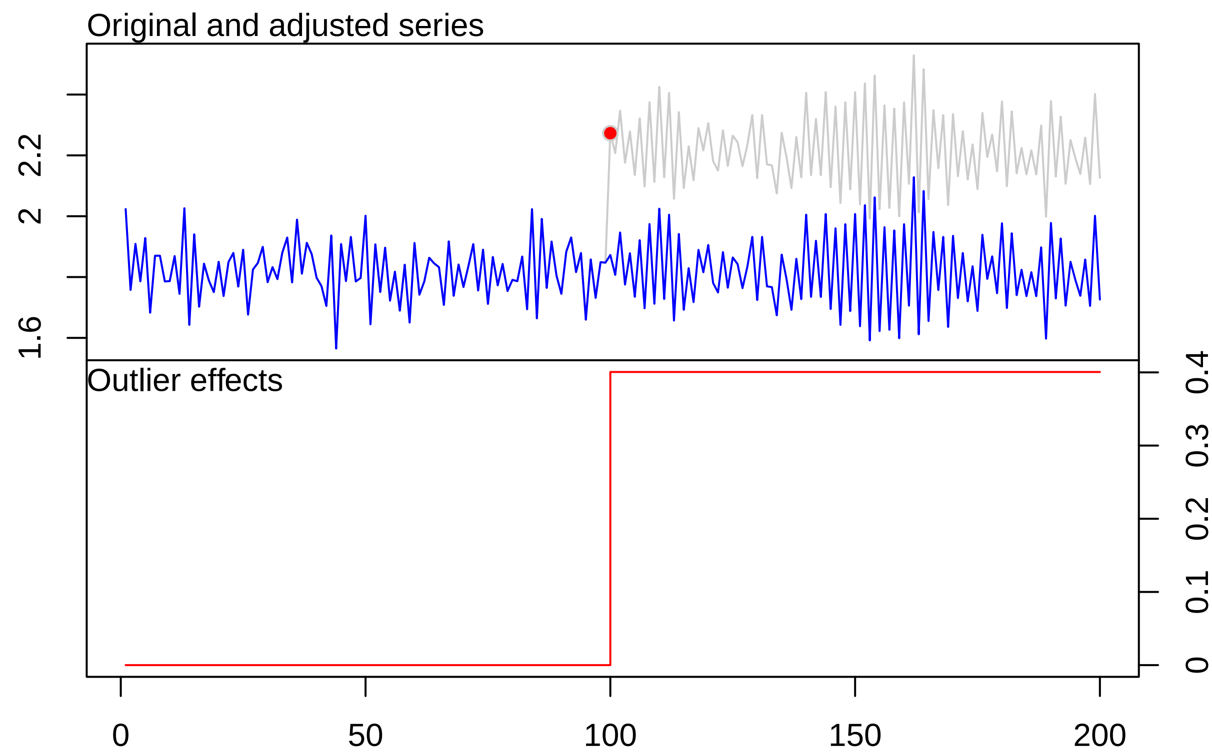
```
##
##  Jarque Bera Test
##
## data:  residuos_TC2
## X-squared = 0.94041, df = 2, p-value = 0.6249
##
##
##  Skewness
##
## data:  residuos_TC2
## statistic = 0.16692, p-value = 0.3352
##
##
##  Kurtosis
##
## data:  residuos_TC2
## statistic = 2.9626, p-value = 0.9141
```

4.3. Cambio de nivel (LS)

```r
# Probamos una función de detección automática de outliers
auto_ls <- tso(zt_LS, tsmethod = "arima",
               args.tsmethod = list(order = c(2, 0, 0),
                                    seasonal = list(order = c(0, 0, 0))))
auto_ls
```

```
##
## Call:
## list(method = NULL)
##
## Coefficients:
##           ar1     ar2  intercept   LS100
##       -0.5369  0.3002     1.8200  0.4005
## s.e.   0.0680  0.0685     0.0056  0.0079
##
## sigma^2 estimated as 0.004733:  log likelihood = 251,  aic = -492
##
## Outliers:
##   type ind time coefhat tstat
## 1   LS 100  100  0.4005    51
```

```r
# Graficamos el efecto del outlier LS
plot.tsoutliers(auto_ls)
```




```r
# Obtenemos la indicatriz para incluir como regresor externo
xreg_LS <- outliers.effects(auto_ls$outliers, length(zt_LS))
head(xreg_LS)
```

```
##      LS100
## [1,]     0
## [2,]     0
## [3,]     0
## [4,]     0
## [5,]     0
## [6,]     0
```

```r
# Creamos la indicatriz para intervenir el modelo
LS <- rep(0, length(zt_LS))
LS[quiebre:N] <- 1
LS
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```r
# Estimamos un modelo AR(2)
modelo_LS2 <- Arima(y = zt_LS,
                    order = c(2, 0, 0),
                    lambda = NULL,
                    xreg = LS)

summary(modelo_LS2)
```

```
## Series: zt_LS
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##           ar1     ar2  intercept    xreg
##       -0.5369  0.3002     1.8200  0.4005
## s.e.   0.0680  0.0685     0.0056  0.0079
##
## sigma^2 = 0.004829:  log likelihood = 251
## AIC=-492   AICc=-491.69   BIC=-475.51
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE      MASE
## Training set 0.0001589623 0.06879479 0.05461582 -0.1082987 2.752769 0.3068884
##                     ACF1
## Training set 0.007982117
```

```r
coefci(modelo_LS2)
```

```
##                2.5 %     97.5 %
## ar1       -0.6702893 -0.4035667
## ar2        0.1659780  0.4344104
## intercept  1.8090054  1.8309069
## xreg       0.3851544  0.4159437
```

```r
coeftest(modelo_LS2)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1       -0.5369280  0.0680427  -7.8910 2.997e-15 ***
## ar2        0.3001942  0.0684789   4.3837 1.167e-05 ***
## intercept  1.8199561  0.0055872 325.7364 < 2.2e-16 ***
## xreg       0.4005491  0.0078545  50.9959 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Guardamos los residuos del modelo
residuos_LS2 <- modelo_LS2$residuals

# Buscamos el residuo máximo del modelo
max(abs(residuos_LS2))
```

```
## [1] 0.2009112
```

```
which.max(abs(residuos_LS2))
```

```
## [1] 162
```

```
# Graficamos los residuos
grafico_residuos <- residuos_LS2 %>% autoplot() +
  labs(x = "Fecha",
      y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")

residuos_LS2_est <- residuos_LS2/sqrt(modelo_LS2$sigma2)
grafico_residuos_est <- residuos_LS2_est %>%
  autoplot() +
  labs(x = "Fecha",
       y = "Residuos estandarizados") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")

grid.arrange(grafico_residuos, grafico_residuos_est)
```

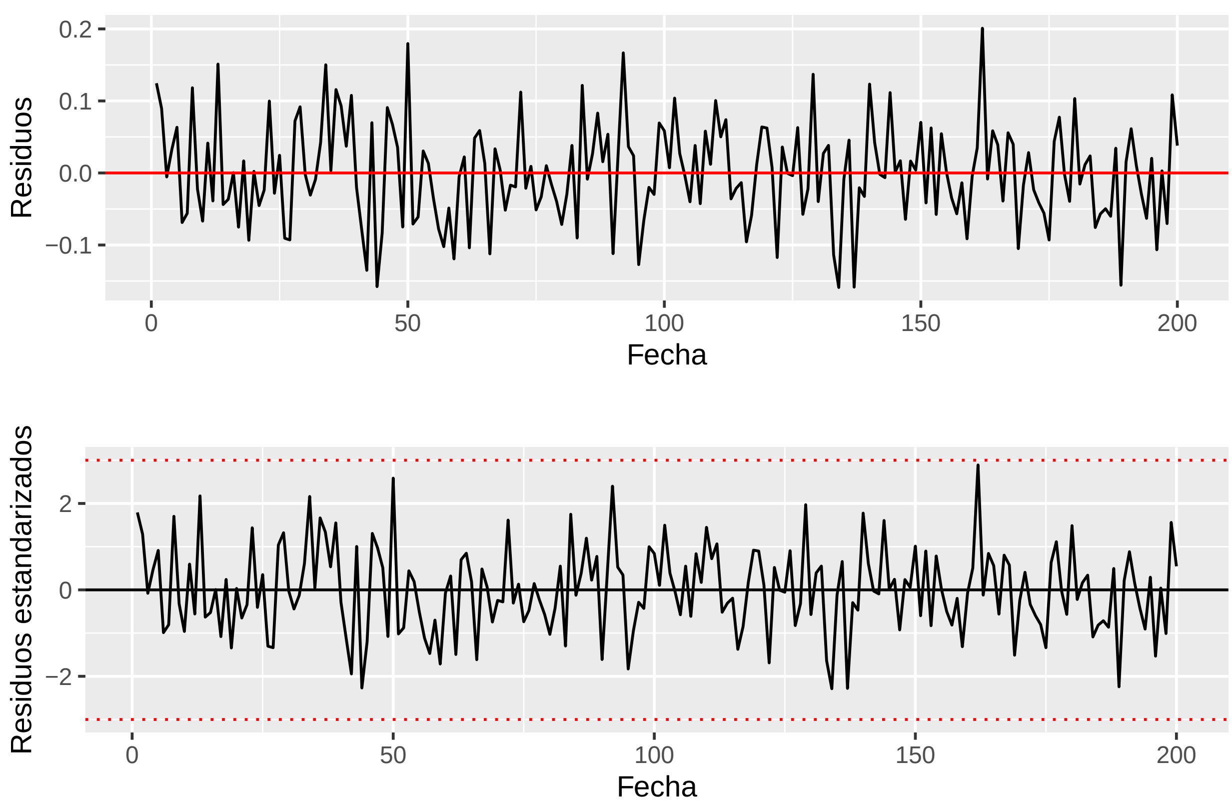


Figura 25: Residuos de un modelo AR(2) intervenido para la serie simulada con un LS.

```r
# FAC
residuos_LS2_acf <- ggAcf(residuos_LS2, lag.max = 50, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_LS2_pacf <- ggAcf(residuos_LS2, lag.max = 50, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_LS2_acf, residuos_LS2_pacf)
```

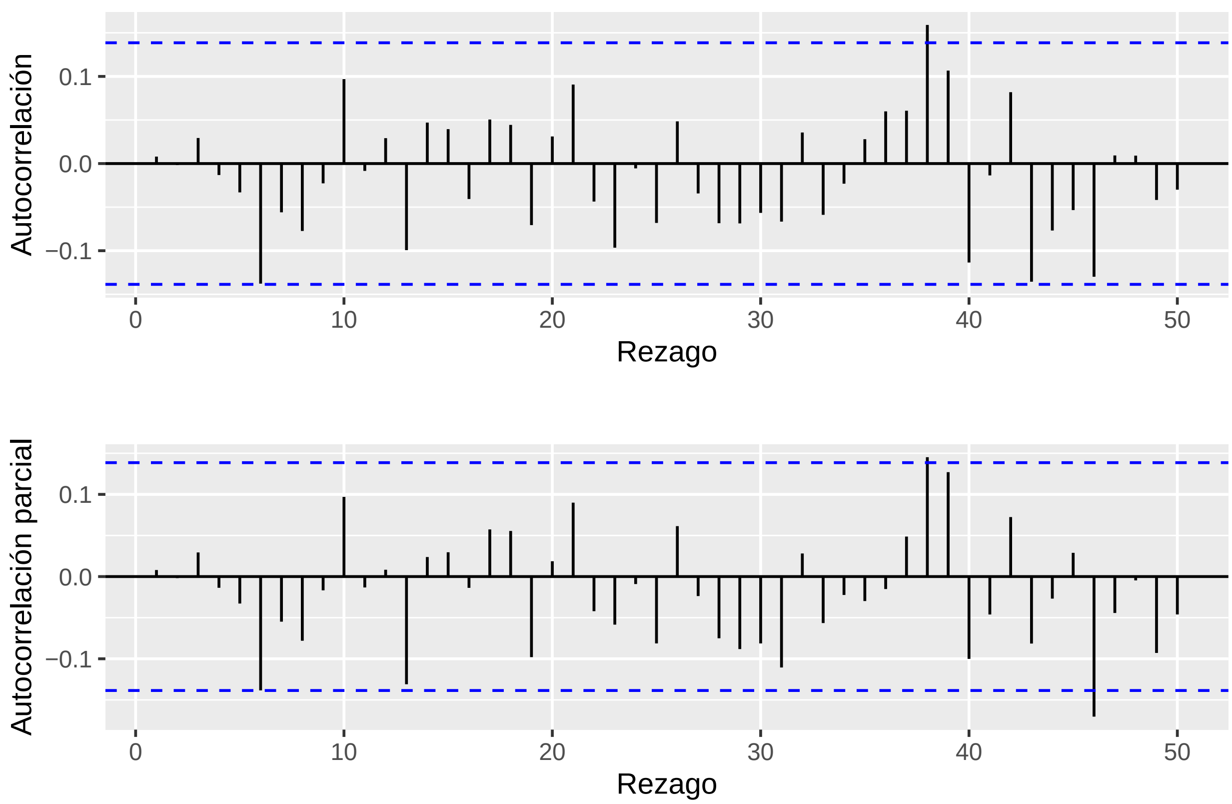


Figura 26: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) intervenido para la serie simulada con un LS.

```r
# Test de Ljung-Box
# No se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos_LS2,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_LS2
## X-squared = 8.4226, df = 8, p-value = 0.3933
```

```r
Box.test(residuos_LS2,
         lag = 30,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_LS2
## X-squared = 23.545, df = 28, p-value = 0.7053
```

```r
Box.test(residuos_LS2,
         lag = 50,
         type = "Ljung-Box",
         fitdf = 2) # p + q
```

```
##
##  Box-Ljung test
##
## data:  residuos_LS2
## X-squared = 54.211, df = 48, p-value = 0.2496
```

```r
# Armamos el QQ-plot de los residuos
ggplot(residuos_LS2, aes(sample = residuos_LS2)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

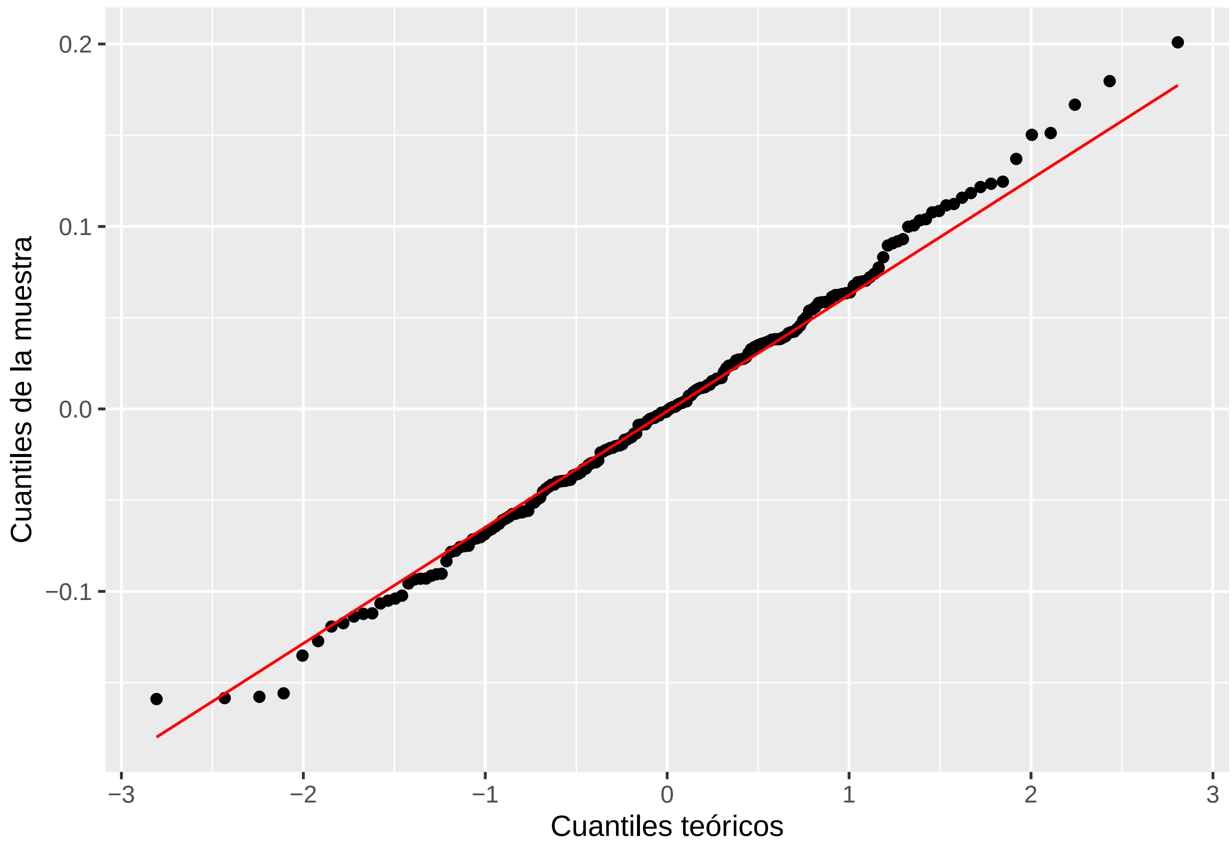


Figura 27: QQ-plot de los residuos de un modelo AR(2) intervenido para la serie simulada con un LS.

```
# Hacemos un histograma de los residuos
ggplot(data = residuos_LS2) +
  geom_histogram(aes(x = residuos_LS2,y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_LS2),
                            sd = sd(residuos_LS2)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

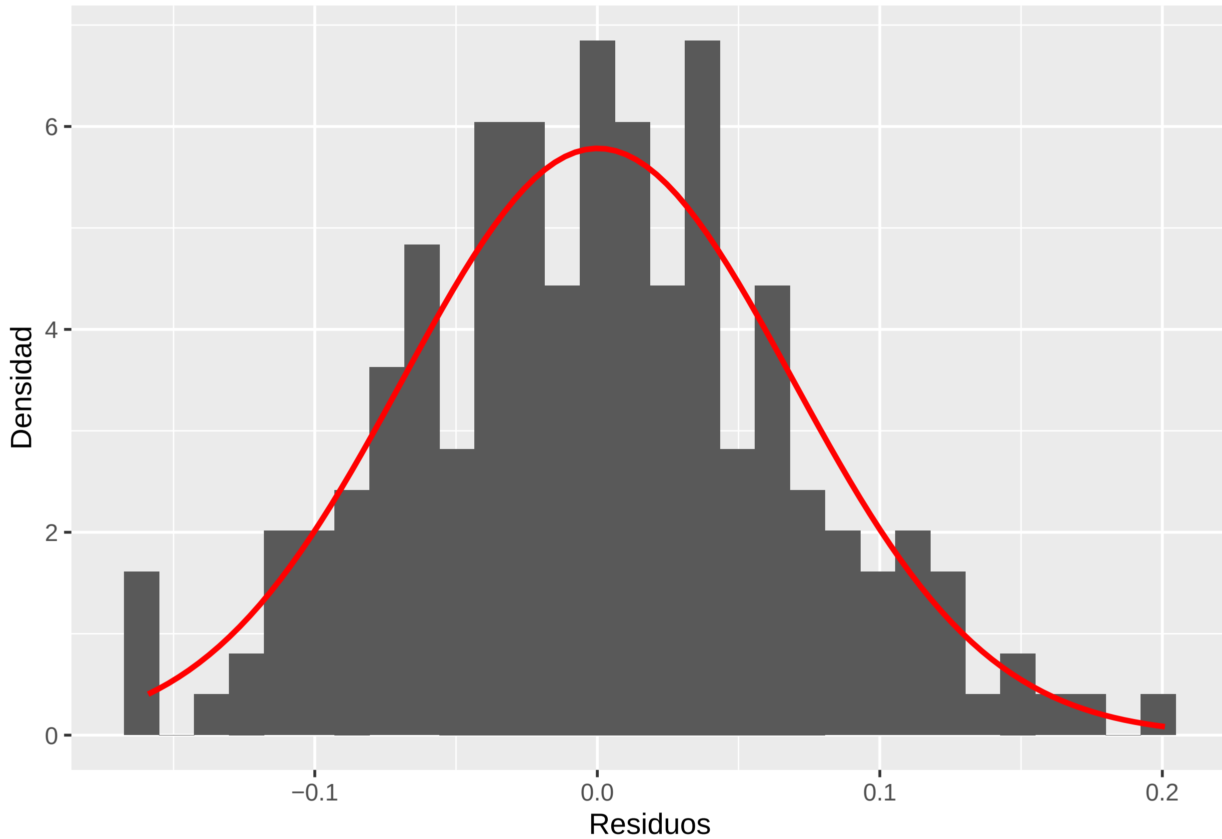


Figura 28: Histograma de los residuos de un modelo AR(2) intervenido para la serie simulada con un LS. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

```r
# Tests de Shapiro y Jarque-Bera
# No se rechaza la hipótesis nula de normalidad
shapiro.test(residuos_LS2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_LS2
## W = 0.99558, p-value = 0.8306
```

```r
JarqueBera.test(residuos_LS2)
```

```
##
##  Jarque Bera Test
##
## data:  residuos_LS2
## X-squared = 0.76558, df = 2, p-value = 0.682
##
##
##  Skewness
##
```

```
## data:  residuos_LS2
## statistic = 0.1482, p-value = 0.3922
##
##
##  Kurtosis
##
## data:  residuos_LS2
## statistic = 2.9366, p-value = 0.8548
```

5. Predicción

5.1. Atípico aditivo (AO)

```r
# Creamos una indicatriz para incluir en las predicciones (consideramos predicciones a 10 pasos)
AO_pred <- rep(0, 10)
AO_pred
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

```r
# Obtenemos las predicciones (debemos incluir un regresor externo)
predicciones_AO <- forecast(modelo_AO2, h = 10, xreg = AO_pred)
predicciones_AO
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 201       1.924570 1.835544 2.013596 1.788416 2.060723
## 202       1.735932 1.634730 1.837135 1.581156 1.890708
## 203       1.896547 1.782596 2.010499 1.722274 2.070821
## 204       1.753875 1.632234 1.875515 1.567842 1.939907
## 205       1.878552 1.750961 2.006144 1.683418 2.073687
## 206       1.768915 1.637019 1.900810 1.567197 1.970632
## 207       1.865094 1.729947 2.000242 1.658404 2.071785
## 208       1.780642 1.643050 1.918235 1.570213 1.991072
## 209       1.854770 1.715320 1.994220 1.641499 2.068041
## 210       1.789696 1.648832 1.930560 1.574263 2.005129
```

```r
# Graficamos las predicciones mediante un fan chart
predicciones_AO <- forecast(modelo_AO2, h = 10, xreg = AO_pred, fan = TRUE)
autoplot(predicciones_AO) +
  labs(x = "Fecha",
       y = "Valor",
       title = "")
```

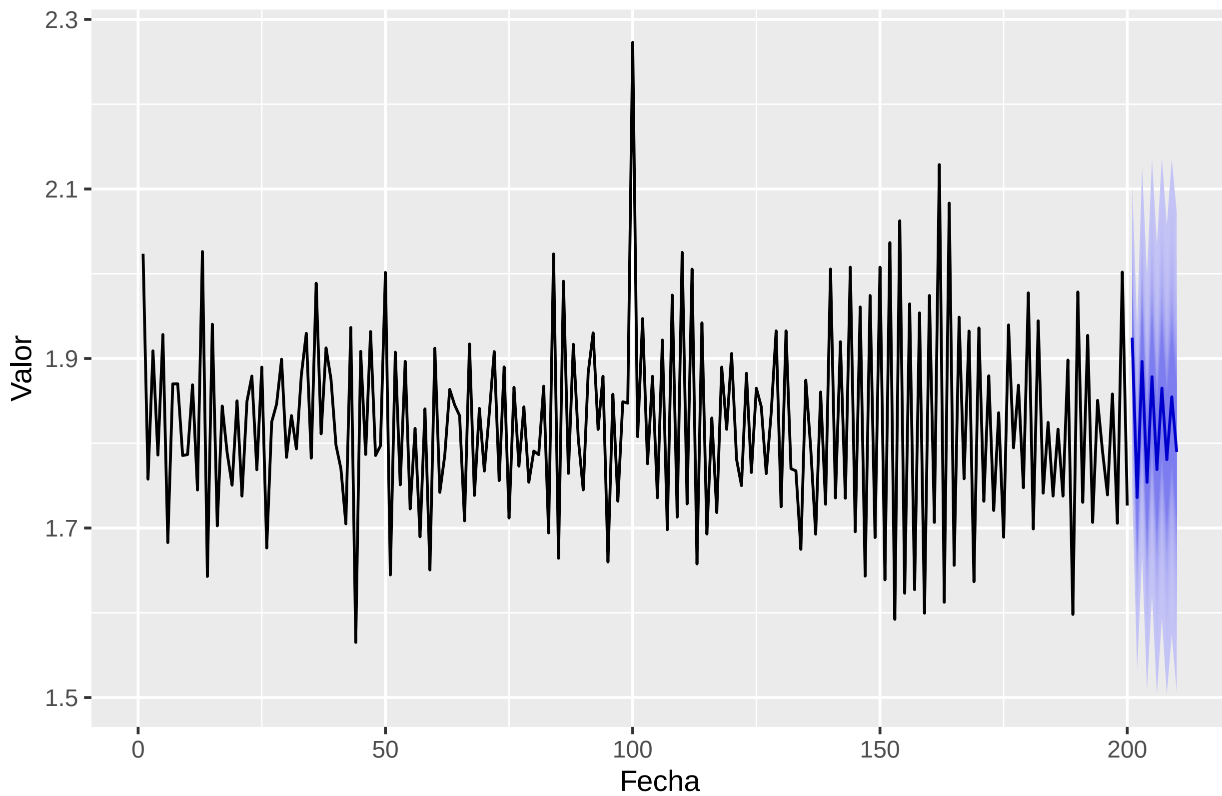


Figura 29: Predicciones a 10 pasos de un AR(2) con un AO. Se consideraron intervalos de confianza entre el 51 % y 99 %.

5.2. Cambio transitorio (TC)

```
# Creamos una indicatriz para incluir en las predicciones (consideramos predicciones a 10 pasos)
TC_pred <- rep(0, 10)
deltaa <- 0.7
TC_pred[1] <- TC[length(TC)]*delta
for (i in 2:length(TC_pred)) {
  TC_pred[i] <- TC_pred[i-1]*delta
}
TC_pred
```

```
##  [1] 2.264134e-16 1.584893e-16 1.109425e-16 7.765978e-17 5.436185e-17
##  [6] 3.805329e-17 2.663730e-17 1.864611e-17 1.305228e-17 9.136596e-18
```

```
# Obtenemos las predicciones (debemos incluir un regresor externo)
predicciones_TC <- forecast(modelo_TC2, h = 10, xreg = TC_pred)
predicciones_TC
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 201       1.922790 1.834039 2.011541 1.787057 2.058523
```

```
## 202       1.735980 1.634663 1.837296 1.581029 1.890930
## 203       1.894935 1.780930 2.008940 1.720579 2.069291
## 204       1.754027 1.632272 1.875783 1.567818 1.940237
## 205       1.877037 1.749348 2.004727 1.681753 2.072322
## 206       1.769040 1.637054 1.901025 1.567185 1.970894
## 207       1.863657 1.728440 1.998874 1.656860 2.070454
## 208       1.780697 1.643055 1.918339 1.570191 1.991203
## 209       1.853414 1.713935 1.992894 1.640100 2.066729
## 210       1.789668 1.648794 1.930542 1.574220 2.005116
```

```r
# Graficamos las predicciones mediante un fan chart
predicciones_TC <- forecast(modelo_TC2, h = 10, xreg = TC_pred, fan = TRUE)
autoplot(predicciones_TC) +
  labs(x = "Fecha",
       y = "Valor",
       title = "")
```

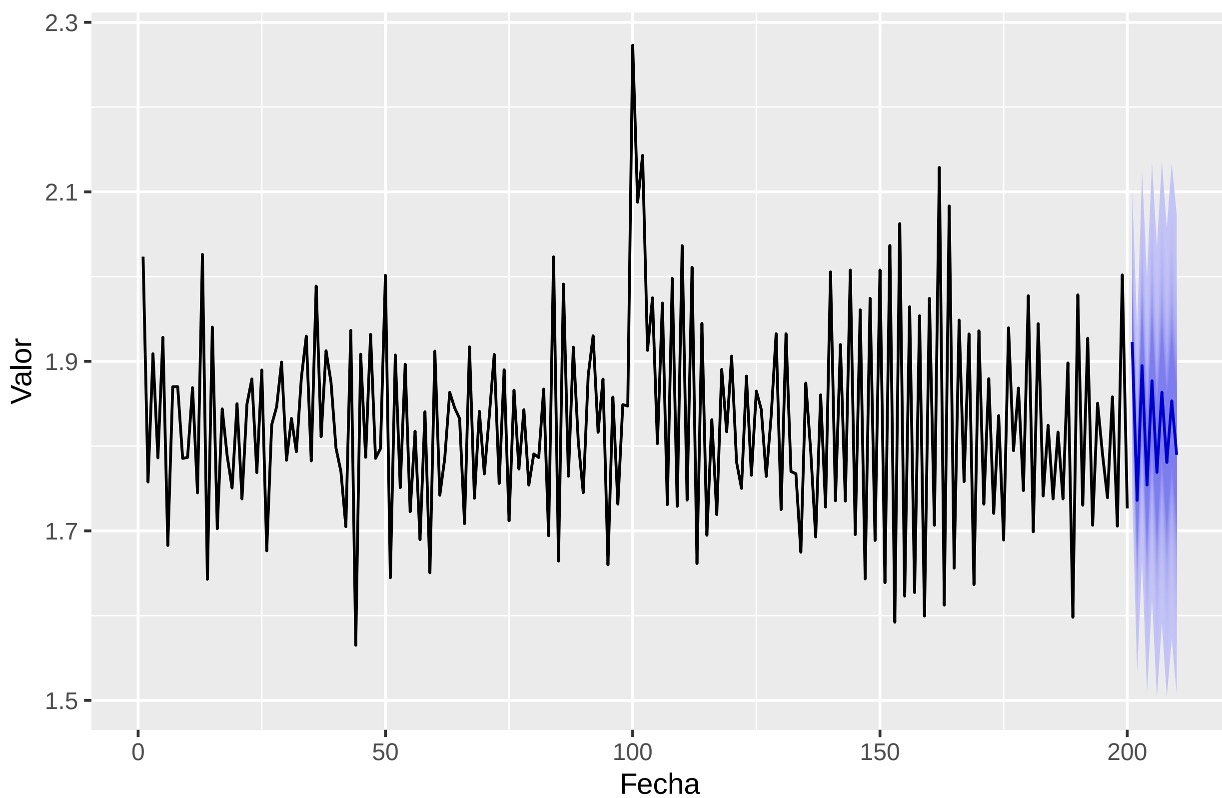


Figura 30: Predicciones a 10 pasos de un AR(2) con un TC. Se consideraron intervalos de confianza entre el 51 % y 99 %.

5.3. Cambio de nivel (LS)

```r
# Creamos una indicatriz para incluir en las predicciones (consideramos predicciones a 10 pasos)
LS_pred <- rep(1, 10)
LS_pred
```

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

```r
# Obtenemos las predicciones (debemos incluir un regresor externo)
predicciones_LS <- forecast(modelo_LS2, h = 10, xreg = LS_pred)
predicciones_LS
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 201       2.325465 2.236405 2.414524 2.189260 2.461669
## 202       2.135929 2.034844 2.237014 1.981333 2.290525
## 203       2.297425 2.183561 2.411288 2.123285 2.471564
## 204       2.153816 2.032281 2.275351 1.967944 2.339688
## 205       2.279404 2.151901 2.406906 2.084405 2.474402
## 206       2.168861 2.037043 2.300680 1.967262 2.370460
## 207       2.265915 2.130828 2.401002 2.059317 2.472513
## 208       2.180620 2.043072 2.318168 1.970259 2.390981
## 209       2.255552 2.116132 2.394973 2.042327 2.468778
## 210       2.189714 2.048866 2.330562 1.974305 2.405123
```

```r
# Graficamos las predicciones mediante un fan chart
predicciones_LS <- forecast(modelo_LS2, h = 10, xreg = LS_pred, fan = TRUE)
autoplot(predicciones_LS) +
  labs(x = "Fecha",
       y = "Valor",
       title = "")
```

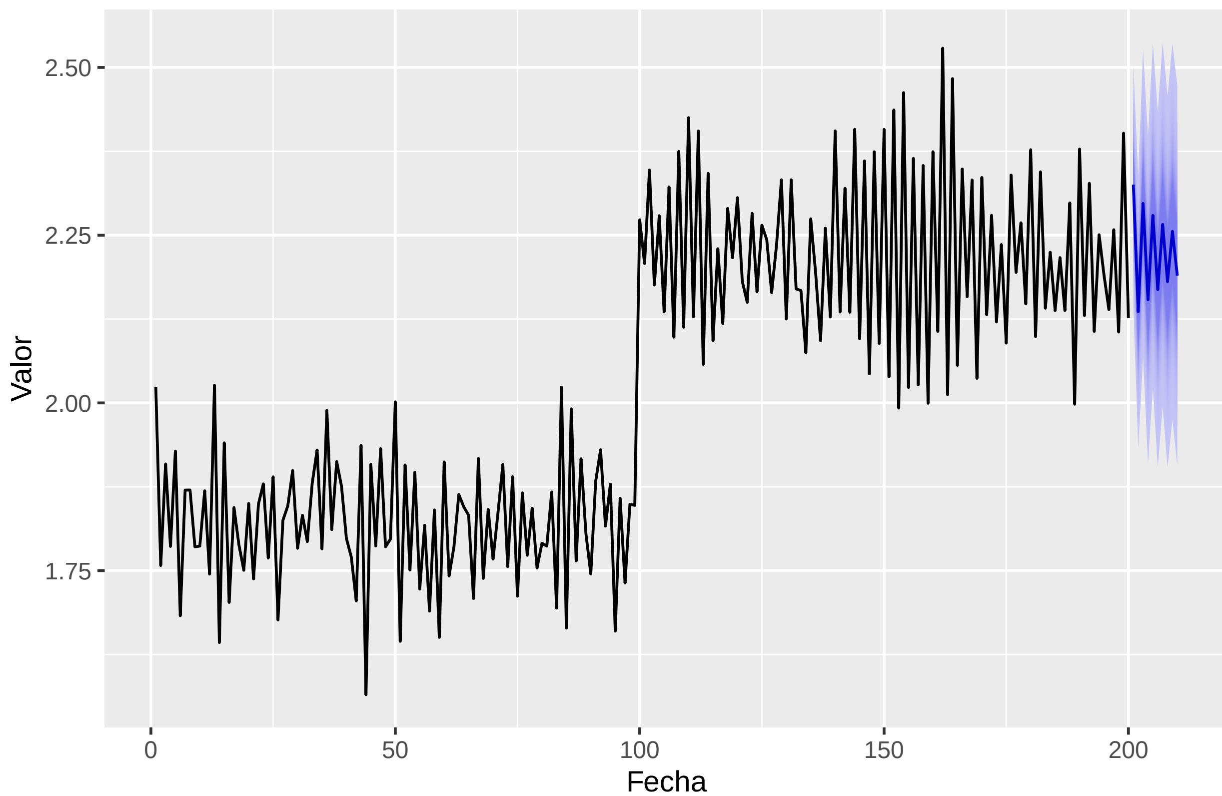


Figura 31: Predicciones a 10 pasos de un AR(2) con un LS. Se consideraron intervalos de confianza entre el 51 % y 99 %.

6. Validación de las predicciones

6.1. Atípico aditivo (AO)

```r
# Definimos una muestra de entrenamiento ("training set") hasta la observación 150 inclusive
train_AO <- window(zt_AO, end = 150)
train_AO_reg <- AO[1:150]

# Dejamos las observaciones 151-200 como conjunto de entrenamiento ("test set")
test_AO <- window(zt_AO, start = 151)
n_AO <- length(test_AO)
```

```r
# Estimamos el modelo para el training set (incluimos una variable indicatriz)
modeloAO_train <- Arima(y = train_AO,
                        order = c(2, 0, 0),
                        lambda = NULL,
                        xreg = train_AO_reg)
```

```r
# Creamos una variable indicatriz para predecir en el test set
AO_test <- AO[151:N]
AO_test
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0 0 0 0 0 0 0
```

```r
# Predecimos fuera de la muestra (el horizonte de predicción será igual al largo del test set)
pred_test_AO <- forecast(modeloAO_train, h = n_AO, xreg = AO_test)
```

```r
# Graficamos las predicciones obtenidas
autoplot(pred_test_AO) +
  autolayer(zt_AO, color = "black") +
  labs(x = "Fecha",
       y = "Valor",
       title = "")
```

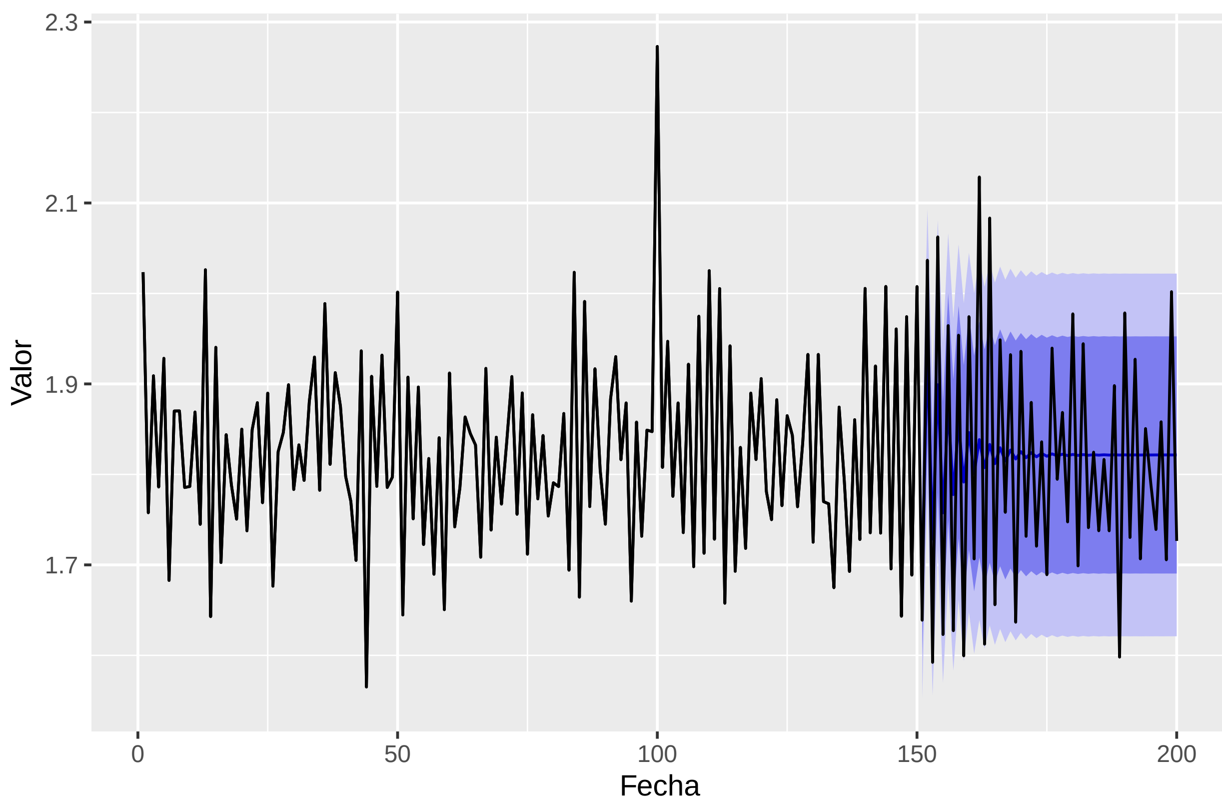


Figura 32: Predicciones en el conjunto de prueba un modelo AR(2) con un AO. La línea azul corresponde a las predicciones.

```r
# Obtenemos los errores de predicción fuera de la muestra
# El segundo argumento de la función accuracy() corresponde al verdadero valor de la serie (conjunto de
accuracy(pred_test_AO, test_AO)
```

```
##                         ME       RMSE        MAE        MPE     MAPE      MASE
## Training set  0.0002704409 0.07012768 0.05576004 -0.1338502 3.060690 0.3558654
## Test set     -0.0068195190 0.12433623 0.10887289 -0.8986085 6.028522 0.6948361
##                      ACF1 Theil's U
## Training set  0.007233479        NA
## Test set     -0.835869876 0.4541751
```

6.2. Cambio transitorio (TC)

```
# Definimos una muestra de entrenamiento ("training set") hasta la observación 150 inclusive
train_TC <- window(zt_TC, end = 150)
train_TC_reg <- TC[1:150]
round(train_TC_reg, 2)
```

```
##   [1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [16] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [31] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [46] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [61] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [76] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##  [91] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.70 0.49 0.34 0.24 0.17
## [106] 0.12 0.08 0.06 0.04 0.03 0.02 0.01 0.01 0.01 0.00 0.00 0.00 0.00 0.00 0.00
## [121] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [136] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```
# Dejamos las observaciones 151-200 como conjunto de entrenamiento ("test set")
test_TC <- window(zt_TC, start = 151)
n_TC <- length(test_TC)
```

```
# Estimamos el modelo para el training set (incluimos una variable indicatriz)
modeloTC_train <- Arima(y = train_TC,
                        order = c(2, 0, 0),
                        lambda = NULL,
                        xreg = train_TC_reg)
```

```
# Creamos una variable indicatriz para predecir en el test set
TC_test <- TC[151:N]
round(TC_test, 2)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0 0 0 0 0 0 0
```

```
# Predecimos fuera de la muestra (el horizonte de predicción será igual al largo del test set)
pred_test_TC <- forecast(modeloTC_train, h = n_TC, xreg = TC_test)
```

```
# Graficamos las predicciones obtenidas
autoplot(pred_test_TC) +
  autolayer(zt_TC, color = "black") +
  labs(x = "Fecha",
       y = "Valor",
       title = "")
```

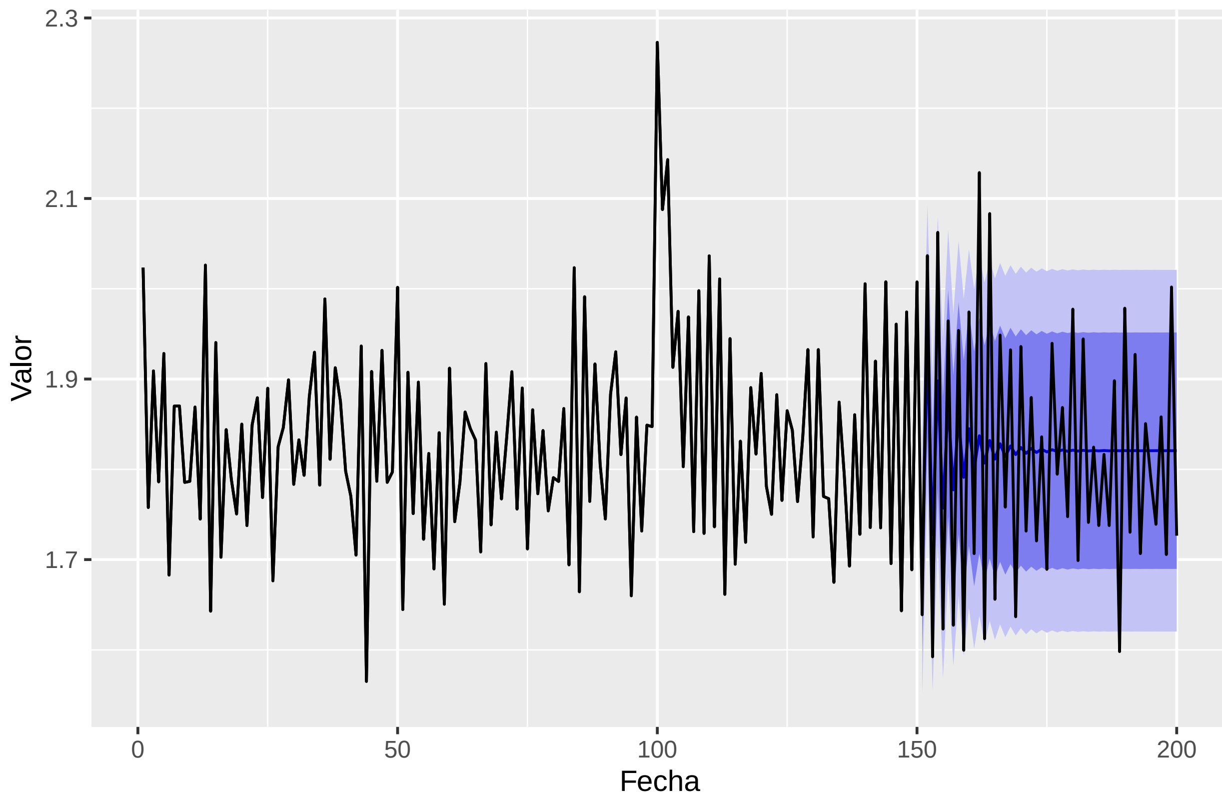


Figura 33: Predicciones en el conjunto de prueba un modelo AR(2) con un TC. La línea azul corresponde a las predicciones.

```
# Obtenemos los errores de predicción fuera de la muestra
# El segundo argumento de la función accuracy() corresponde al verdadero valor de la serie (conjunto de
accuracy(pred_test_TC, test_TC)
```

```
##                         ME       RMSE        MAE        MPE     MAPE      MASE
## Training set  0.0002838797 0.06986611 0.05563267 -0.1303790 3.046907 0.3601324
## Test set     -0.0059259658 0.12442469 0.10891045 -0.8495589 6.027208 0.7050208
##                      ACF1 Theil's U
## Training set  0.008959694        NA
## Test set     -0.836198604 0.4549729
```

6.3. Cambio de nivel (LS)

```
# Definimos una muestra de entrenamiento ("training set") hasta la observación 150 inclusive
train_LS <- window(zt_LS, end = 150)
train_LS_reg <- LS[1:150]

# Dejamos las observaciones 151-200 como conjunto de entrenamiento ("test set")
test_LS <- window(zt_LS, start = 151)
n_LS <- length(test_LS)
```

```r
# Estimamos el modelo para el training set (incluimos una variable indicatriz)
modeloLS_train <- Arima(y = train_LS,
                        order = c(2, 0, 0),
                        lambda = NULL,
                        xreg = train_LS_reg)
```

```r
# Creamos una variable indicatriz para predecir en el test set
LS_test <- LS[151:N]
LS_test
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1
```

```r
# Predecimos fuera de la muestra (el horizonte de predicción será igual al largo del test set)
pred_test_LS <- forecast(modeloLS_train, h = n_LS, xreg = LS_test)
```

```r
# Graficamos las predicciones obtenidas
autoplot(pred_test_LS) +
  autolayer(zt_LS, color = "black") +
  labs(x = "Fecha",
       y = "Valor",
       title = "")
```

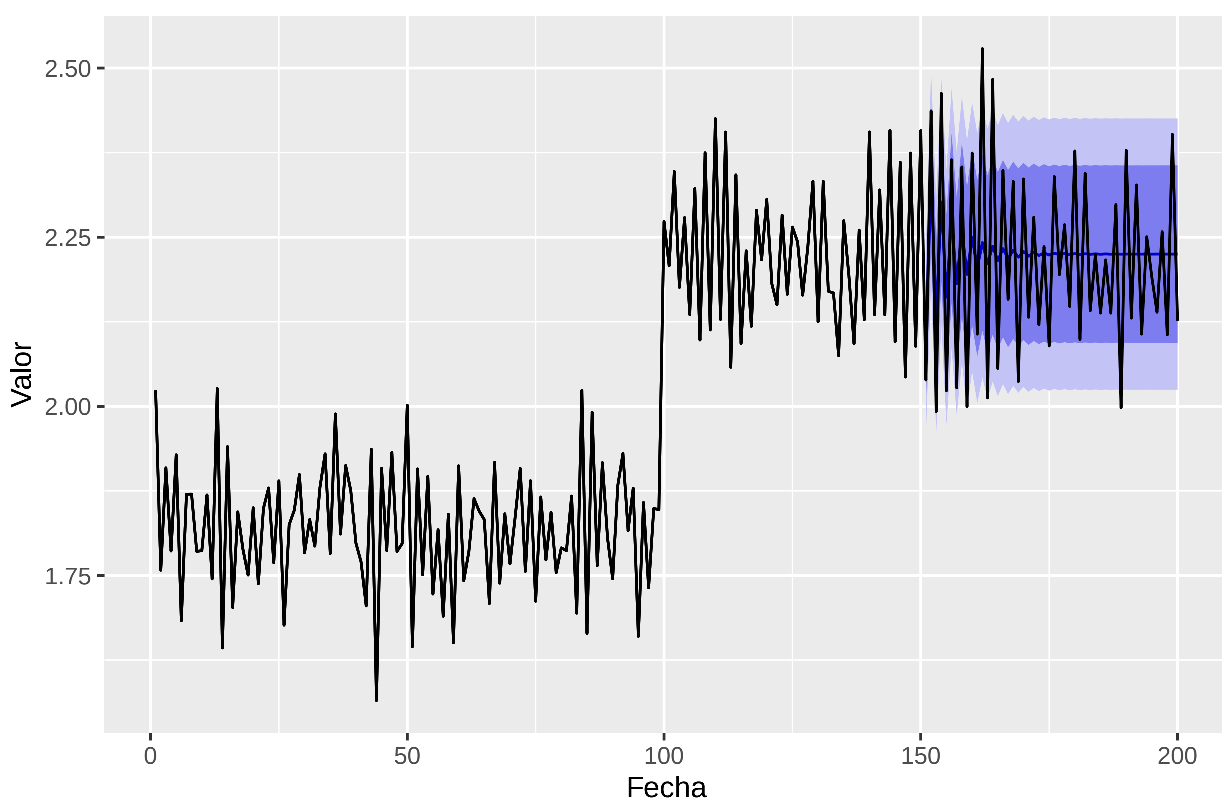


Figura 34: Predicciones en el conjunto de prueba un modelo AR(2) con un LS. La línea azul corresponde a las predicciones.

```r
# Obtenemos los errores de predicción fuera de la muestra
# El segundo argumento de la función accuracy() corresponde al verdadero valor de la serie (conjunto de
accuracy(pred_test_LS, test_LS)
```

```
##                         ME       RMSE        MAE        MPE     MAPE      MASE
## Training set  0.0002608218 0.07010192 0.05572413 -0.1159675 2.892470 0.3618356
## Test set     -0.0103463688 0.12462835 0.10924335 -0.8189376 4.954617 0.7093539
##                      ACF1 Theil's U
## Training set  0.007554034        NA
## Test set     -0.835942194 0.4538215
```