Taller 5 - Datos atípicos

Series Cronológicas 2024

Mayo 2024

1. Exploración de los datos

```
# Graficamos la serie PIB Comercio
autoplot(comercio) +
  labs(x = "Fecha",
       y = "PIB Comercio") +
  scale_x_continuous(breaks = 2016:2024) +
  theme(panel.grid.minor = element_blank())
```

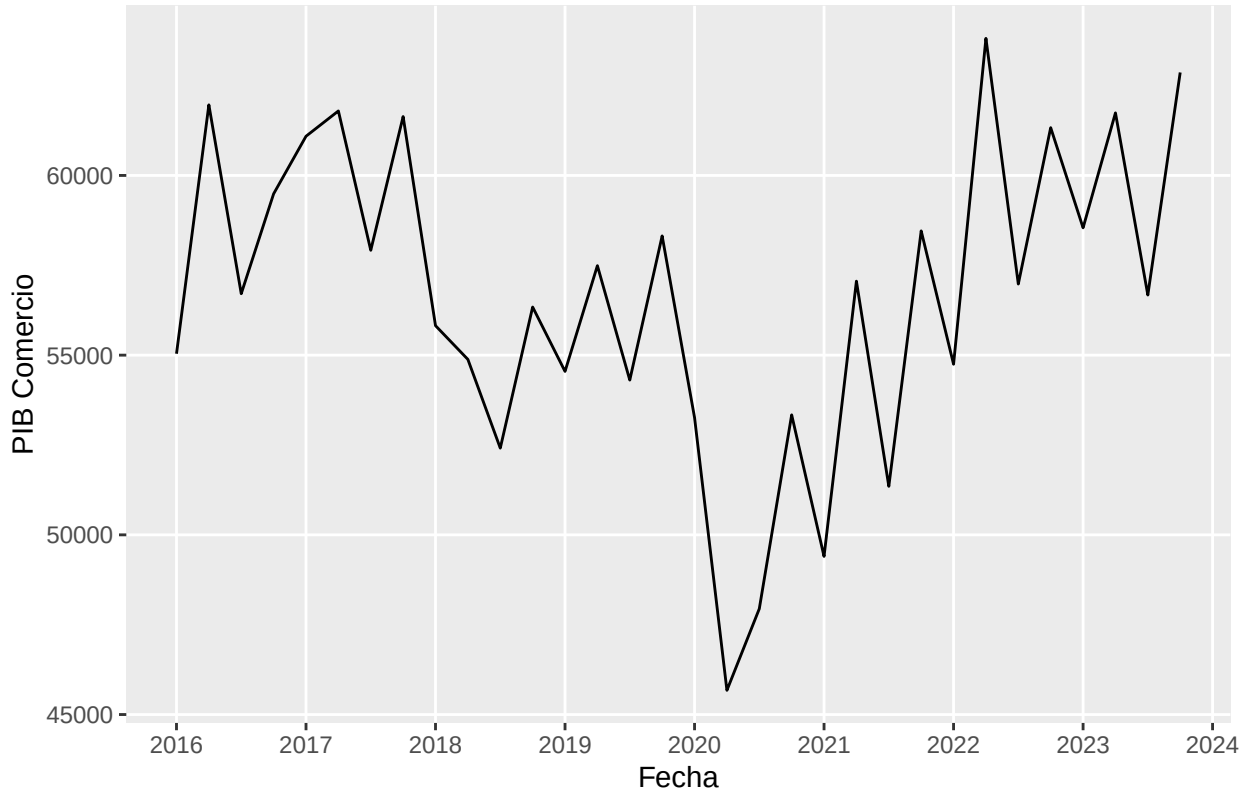


Figura 1: Evolución del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

2. Identificación y estimación del modelo

```
# FAC (no descartamos estacionariedad)
comercio_acf <- ggAcf(comercio, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
comercio_pacf <- ggAcf(comercio, lag.max = 24, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(comercio_acf, comercio_pacf)
```

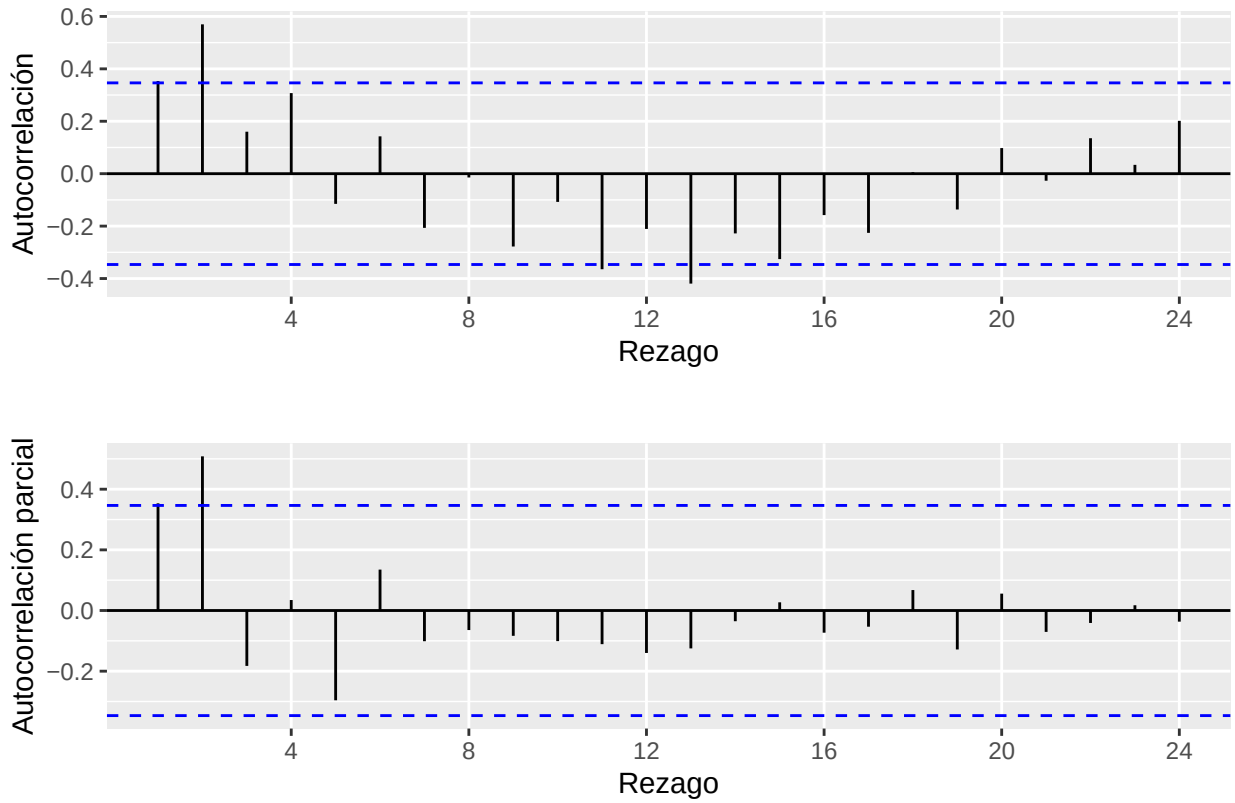


Figura 2: Funciones de Autocorrelación y Autocorrelación Parcial estimadas del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```
# Dada la forma de la FAC y la FACP, un posible modelo es un AR(2).

# AR(2)
```

```r
modelo1 <- Arima(y = comercio, # Datos para estimar
                 order = c(2, 0, 0), # Orden del modelo (suponemos estacionariedad)
                 lambda = NULL) # Trabajamos con la serie sin transformar

coeftest(modelo1)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ar1       1.4814e-01 1.4875e-01  0.9959 0.3193194
## ar2       5.4577e-01 1.5178e-01  3.5958 0.0003234 ***
## intercept 5.7151e+04 1.7677e+03 32.3306 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Probamos un AR(2) con phi1 = 0
modelo1 <- Arima(y = comercio,
                 order = c(2, 0, 0),
                 lambda = NULL,
                 fixed = c(0, NA, NA))

summary(modelo1)
```

```
## Series: comercio
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##       ar1     ar2       mean
##         0  0.6047  57052.412
## s.e.    0  0.1396   1418.146
##
## sigma^2 = 12537510:  log likelihood = -306.34
## AIC=618.67   AICc=619.53   BIC=623.07
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -73.20484 3428.398 2497.484 -0.5355904 4.582298 0.5984372
##                    ACF1
## Training set 0.2758011
```

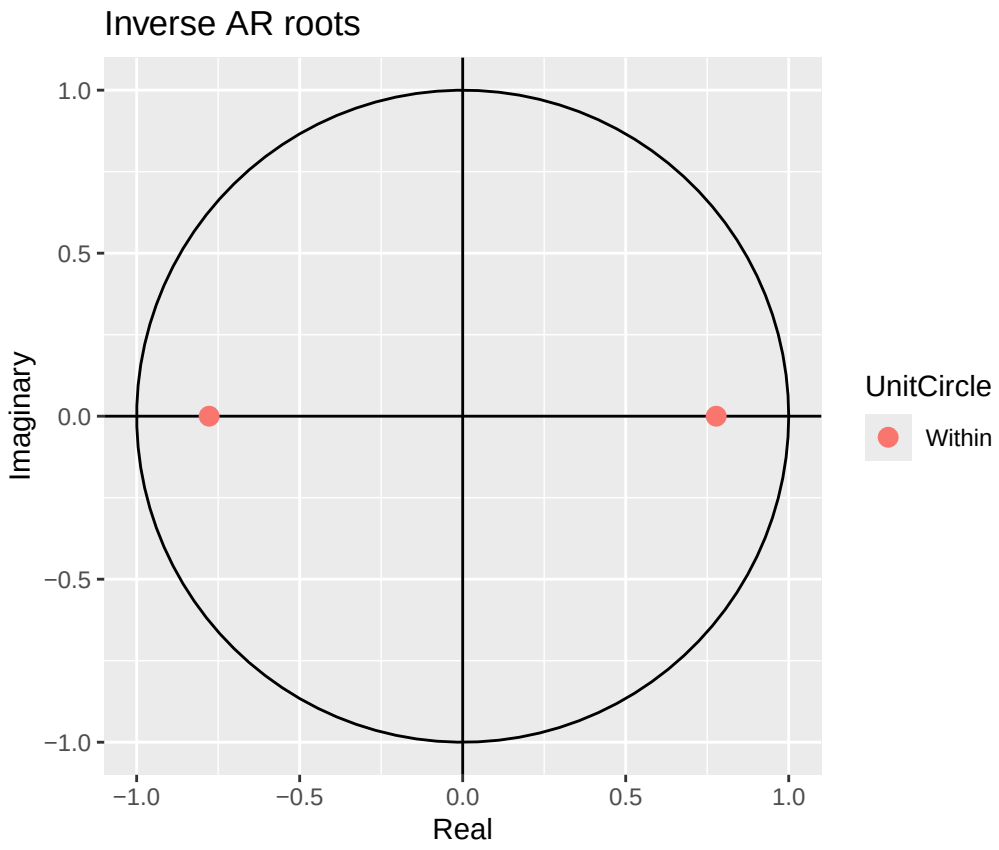
```r
coeftest(modelo1)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ar2       6.0467e-01 1.3960e-01  4.3315 1.481e-05 ***
## intercept 5.7052e+04 1.4181e+03 40.2303 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefci(modelo1)
```

```
##                   2.5 %       97.5 %
## ar2       3.310641e-01 8.782737e-01
## intercept 5.427290e+04 5.983193e+04
```

```
autoplot(modelo1)
```




```
# Nos quedamos con un modelo sin phi1
```

```
# Graficamos la serie y los valores ajustados

fit1 <- autoplot(comercio) +
  autolayer(modelo1$fitted, color = "blue") +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "Modelo AR(2)")

fit1
```

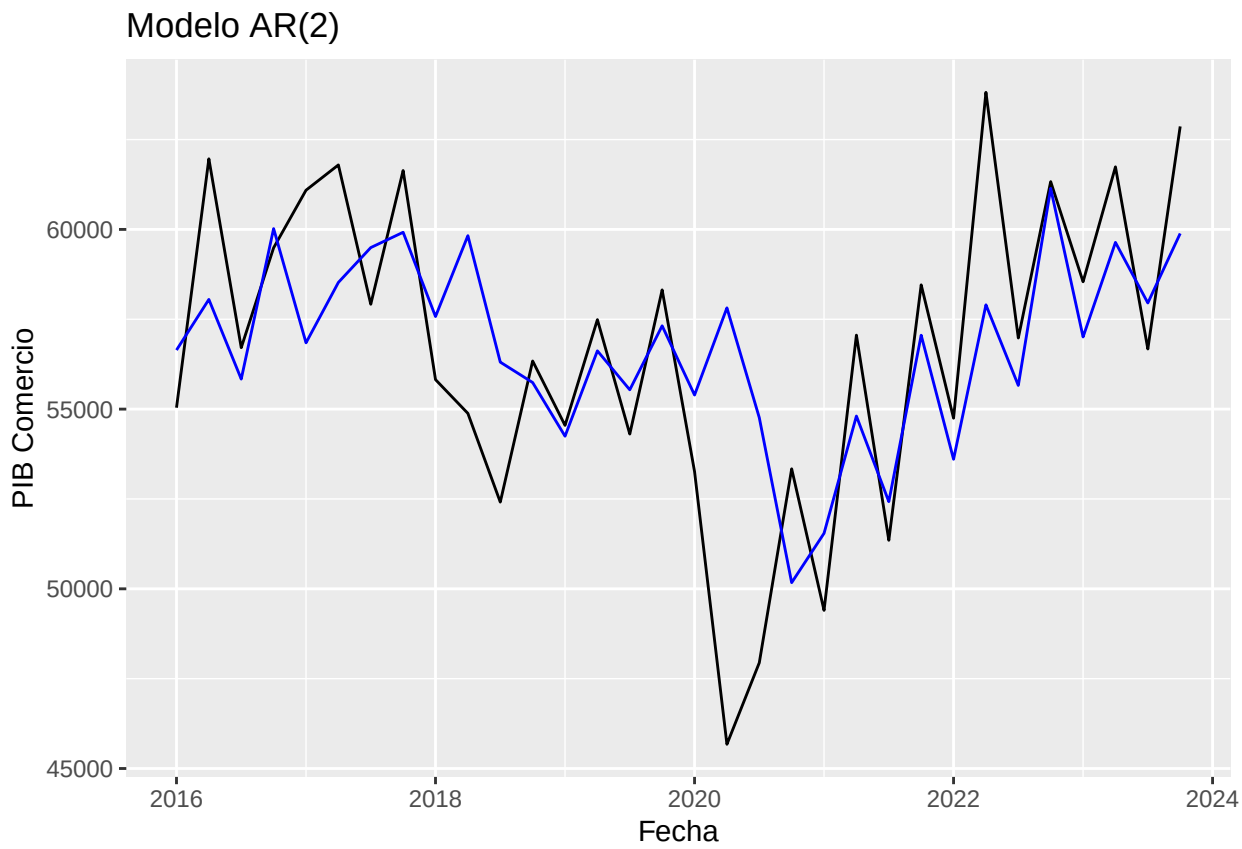


Figura 3: PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023 y valores ajustados para un modelo AR(2). La línea negra corresponde a los valores reales y la azul a los ajustados.

3. Diagnóstico del modelo

3.1. Análisis gráfico de los residuos

```
# Guardamos los residuos del modelo
residuos1 <- modelo1$residuals

# Buscamos los residuos máximos y mínimos

max(residuos1)
```

```
## [1] 5912.371
```

```
which.max(residuos1)
```

```
## [1] 26
```

```r
time(residuos1)[which.max(residuos1)] # Junio de 2022
```

```
## [1] 2022.25
```

```r
min(residuos1)
```

```
## [1] -12143.76
```

```r
which.min(residuos1)
```

```
## [1] 18
```

```r
time(residuos1)[which.min(residuos1)] # Junio de 2020
```

```
## [1] 2020.25
```

```r
# Graficamos los residuos
residuos1 %>% autoplot() +
  labs(x = "Fecha",
       y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")
```

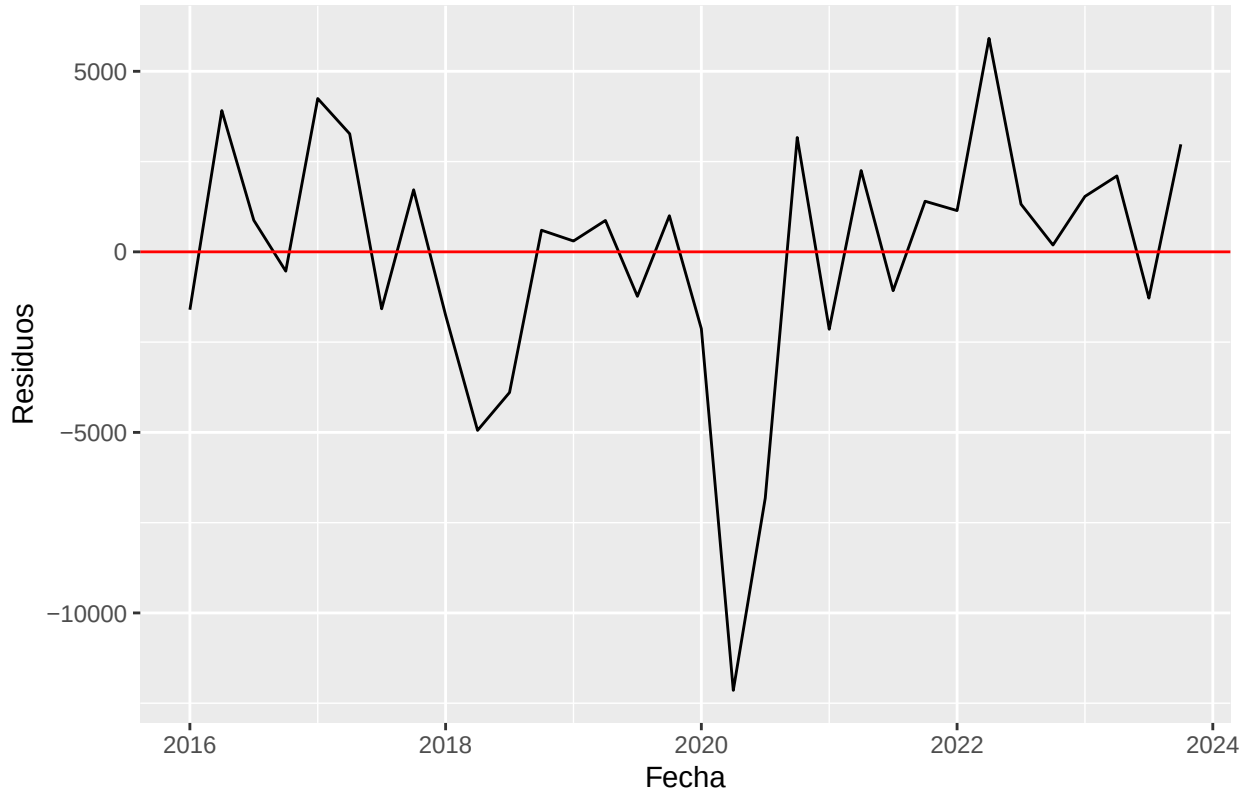


Figura 4: Residuos de un modelo AR(2) para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```r
# Residuos estandarizados
residuos1_est <- residuos1/sqrt(modelo1$sigma2)
residuos1_est %>% autoplot() +
  labs(x = "Fecha",
       y = "Residuos estandarizados",
       title = "PIB comercio") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")
```

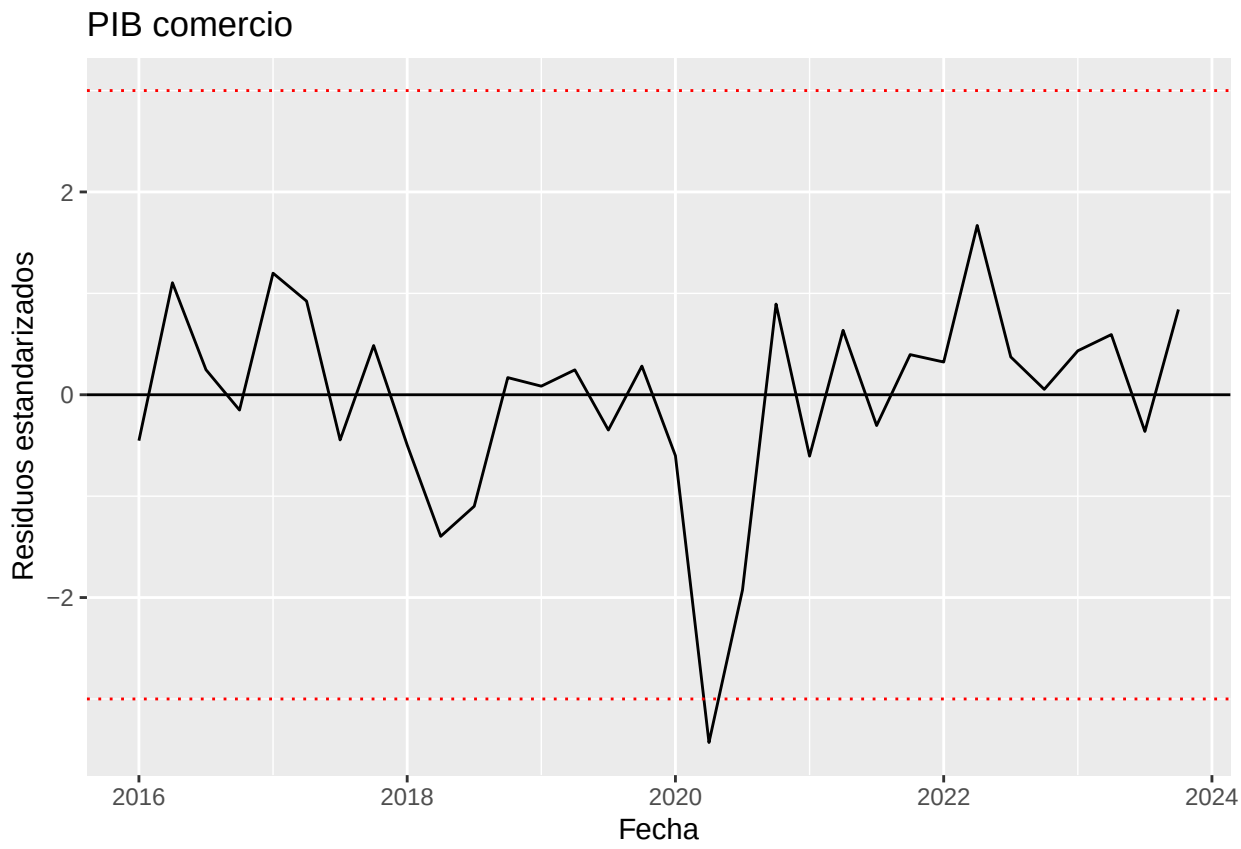


Figura 5: Residuos estandarizados de un modelo AR(2) para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```r
time(residuos1_est)[which.max(residuos1_est)] # Junio de 2022
```

```
## [1] 2022.25
```

3.2. Autocorrelación de los residuos

3.2.1. FAC y FACP de los residuos

```r
# FAC
residuos_acf <- ggAcf(residuos1, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_pacf <- ggAcf(residuos1, lag.max = 24, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_acf, residuos_pacf)
```

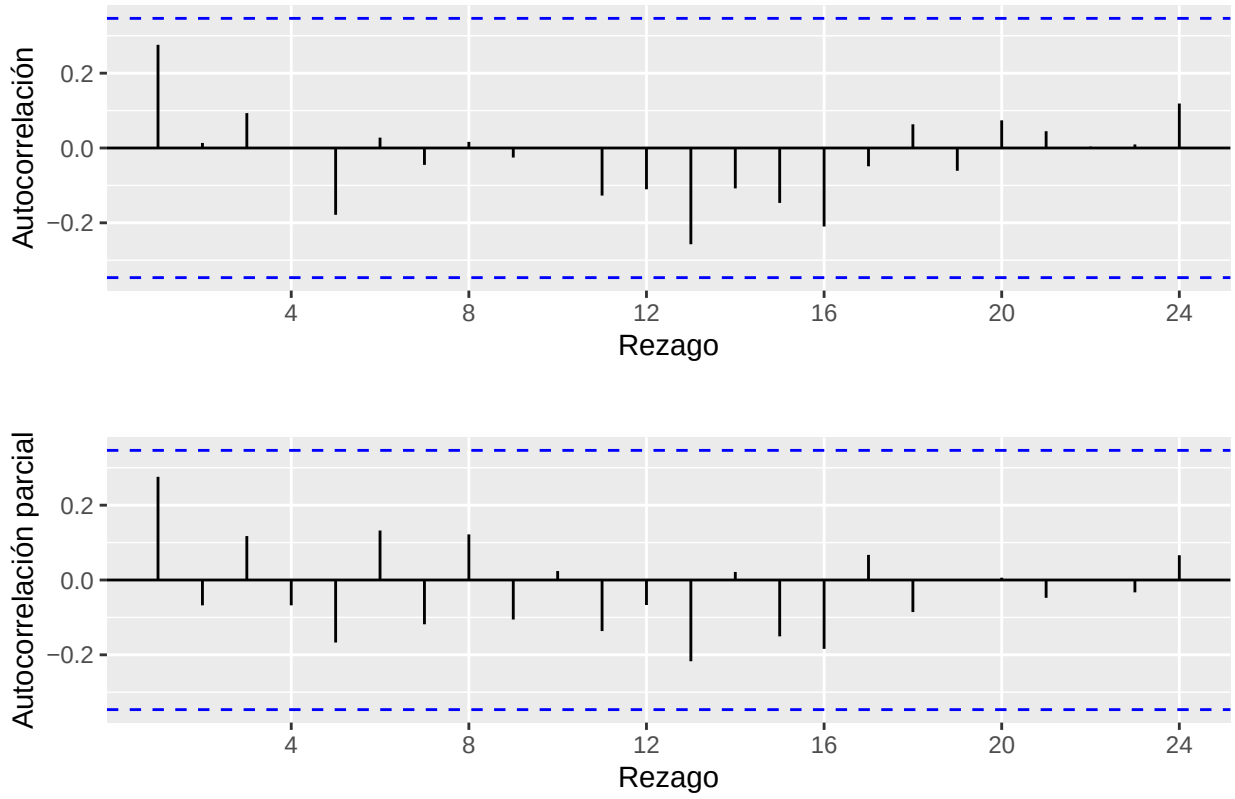


Figura 6: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```r
# Otra posibilidad es graficar a la vez los residuos, su FAC y su FACP
# checkresiduals(residuos1)
# tsdisplay(residuos1)
```

3.2.2. Contraste de autocorrelación de los residuos

```
# Test de Ljung-Box

# Se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos1,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q de un modelo ARMA(p,q)
```

```
##
##  Box-Ljung test
##
## data:  residuos1
## X-squared = 4.4506, df = 8, p-value = 0.8144
```

3.3. Normalidad de los residuos

3.3.1. QQ-plot de los residuos

```
# Armamos el QQ-plot de los residuos
ggplot(residuos1, aes(sample = residuos1)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

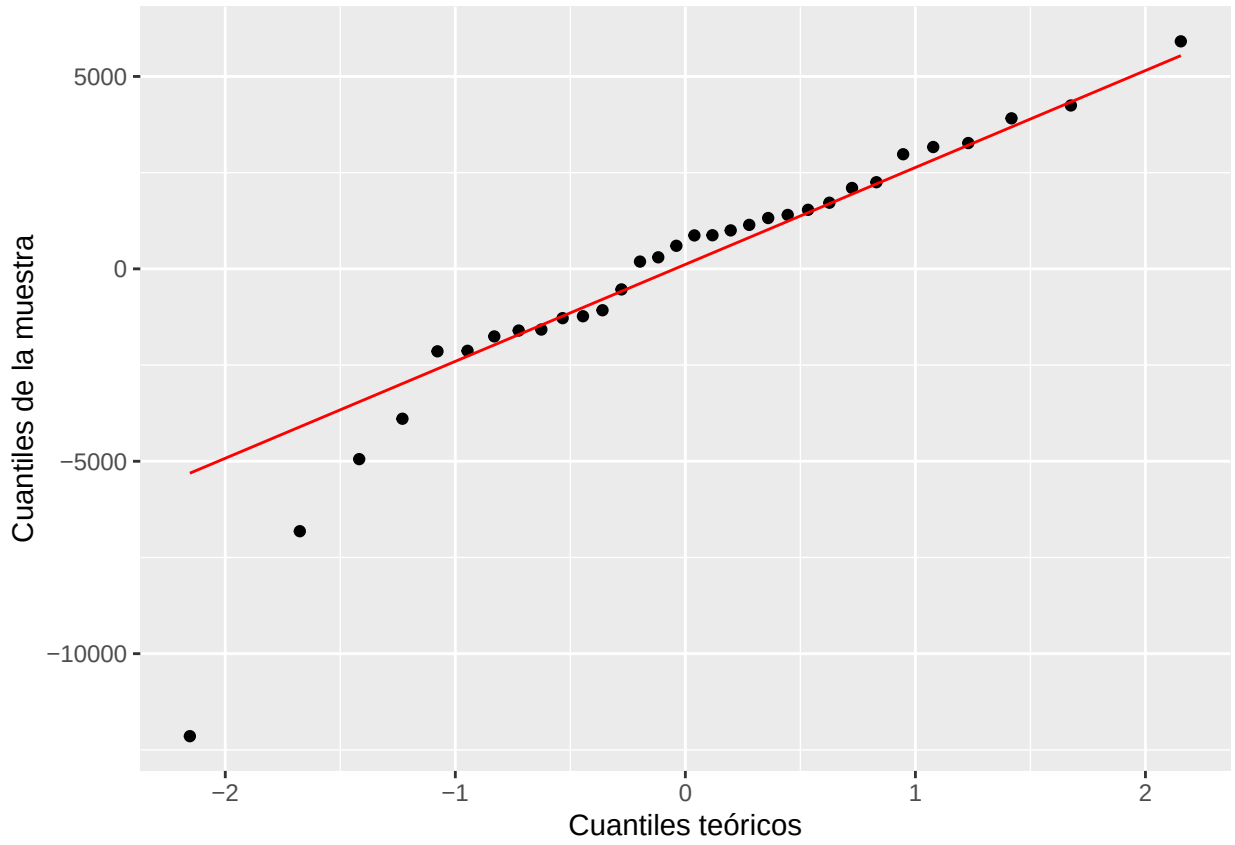


Figura 7: QQ-plot de los residuos de un modelo AR(2) para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

3.3.2. Histograma de los residuos

```
# Hacemos un histograma de los residuos
ggplot(data = residuos1) +
  geom_histogram(aes(x = residuos1, y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos1),
                            sd = sd(residuos1)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

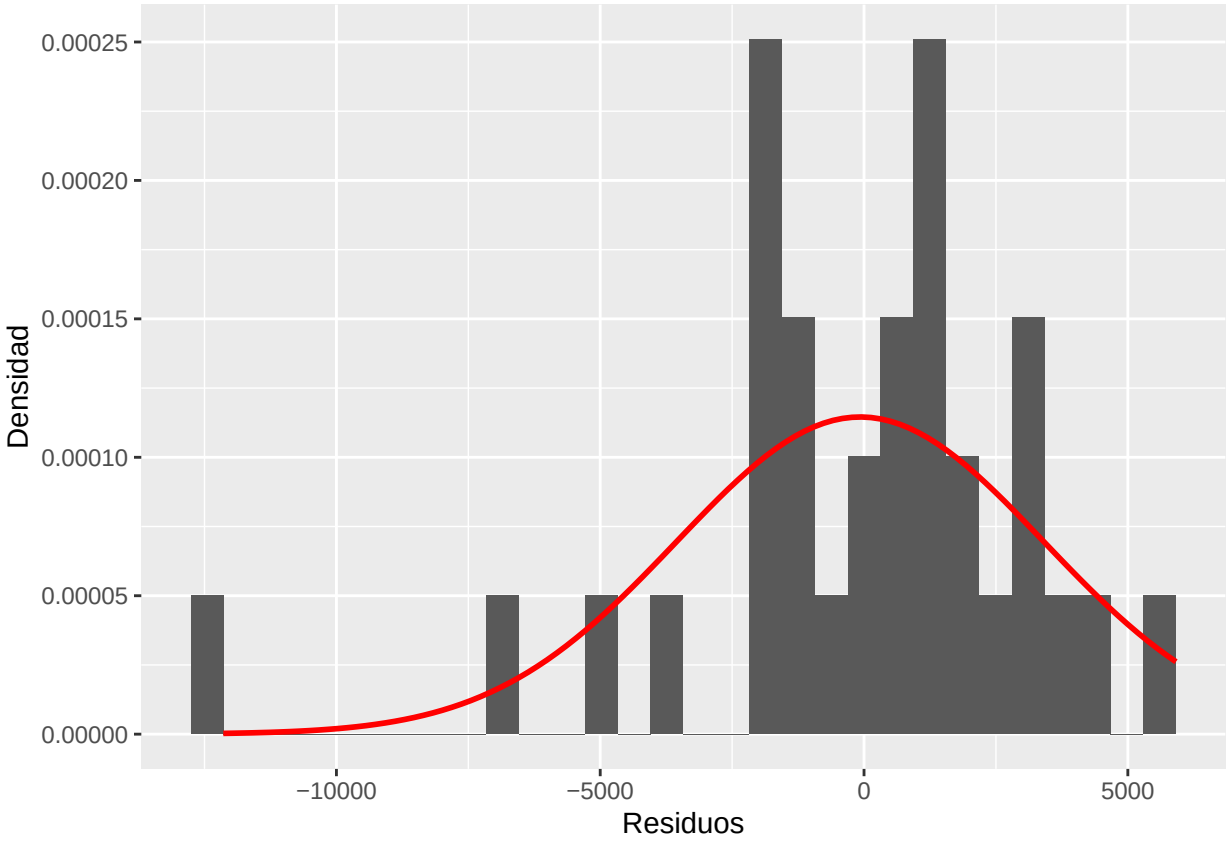


Figura 8: Histograma de los residuos de un modelo AR(2) para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

3.3.3. Contrastes de normalidad de los residuos

```
# Tests de Shapiro y Jarque-Bera
# Se rechaza la hipótesis nula de normalidad dado que hay outliers
shapiro.test(residuos1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos1
## W = 0.9036, p-value = 0.007657
```

```
JarqueBera.test(residuos1)
```

```
##
##  Jarque Bera Test
##
## data:  residuos1
## X-squared = 21.904, df = 2, p-value = 1.753e-05
```

```
##
##
##  Skewness
##
## data:  residuos1
## statistic = 1.3726, p-value = 0.001525
##
##
##  Kurtosis
##
## data:  residuos1
## statistic = 5.9819, p-value = 0.0005749
```

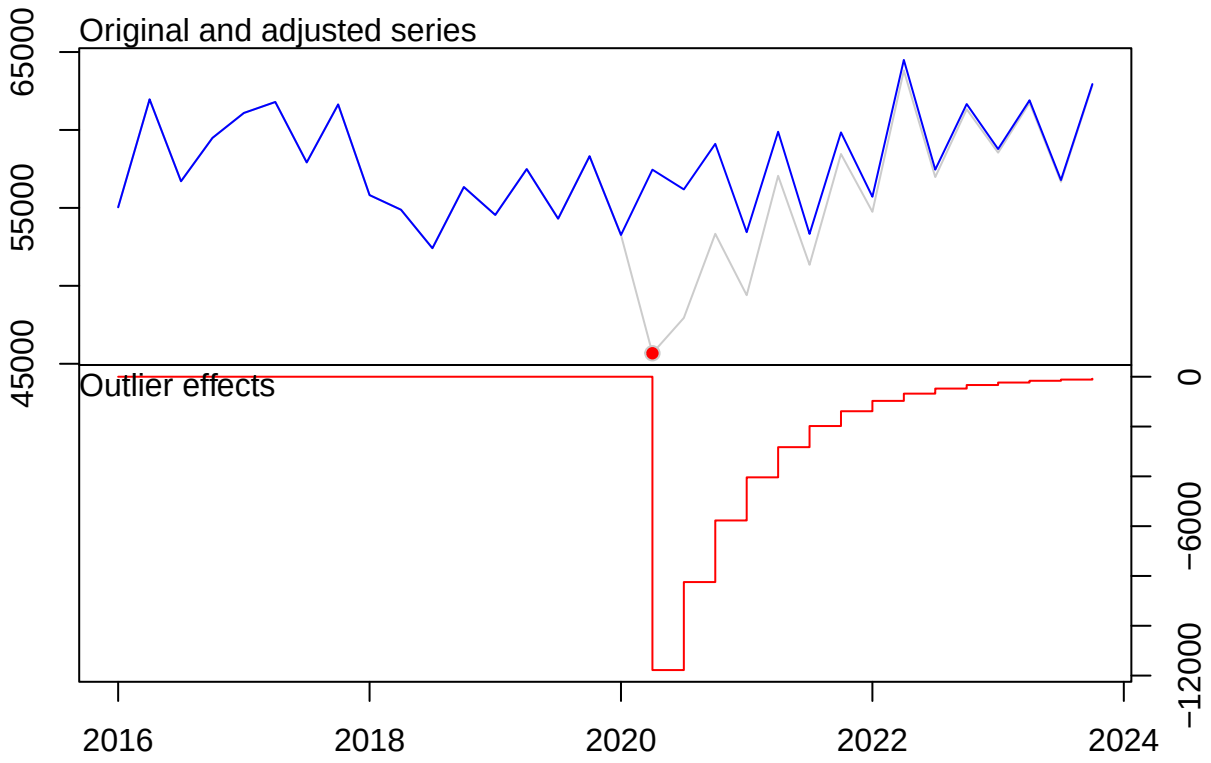
4. Intervención de outliers

4.1. Identificación de outliers

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

```
##
## Call:
## list(method = NULL)
##
## Coefficients:
##           ar1     ar2  intercept        TC18
##       -0.0608  0.7017  58124.747  -11778.166
## s.e.   0.1229  0.1260   1021.529    1822.925
##
## sigma^2 estimated as 5125278:  log likelihood = -293.3,  aic = 596.6
##
## Outliers:
##   type ind    time coefhat  tstat
## 1   TC  18 2020:02  -11778 -6.461
```

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




```
# Obtenemos la indicatriz para incluir como regresor externo
xreg <- outliers.effects(outliers_comercio$outliers, length(comercio))
xreg
```

```
##              TC18
##  [1,] 0.000000000
##  [2,] 0.000000000
##  [3,] 0.000000000
##  [4,] 0.000000000
##  [5,] 0.000000000
##  [6,] 0.000000000
##  [7,] 0.000000000
##  [8,] 0.000000000
##  [9,] 0.000000000
## [10,] 0.000000000
## [11,] 0.000000000
## [12,] 0.000000000
## [13,] 0.000000000
## [14,] 0.000000000
## [15,] 0.000000000
## [16,] 0.000000000
## [17,] 0.000000000
## [18,] 1.000000000
## [19,] 0.700000000
## [20,] 0.490000000
## [21,] 0.343000000
```

```
## [22,] 0.240100000
## [23,] 0.168070000
## [24,] 0.117649000
## [25,] 0.082354300
## [26,] 0.057648010
## [27,] 0.040353607
## [28,] 0.028247525
## [29,] 0.019773267
## [30,] 0.013841287
## [31,] 0.009688901
## [32,] 0.006782231
```

4.2. Reestimación del modelo

```
# Reestimamos el modelo
modelo2 <- Arima(y = comercio,
                 order = c(2, 0, 0),
                 lambda = NULL,
                 xreg = xreg,
                 fixed = c(0, NA, NA, NA))

summary(modelo2)
```

```
## Series: comercio
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##       ar1     ar2  intercept        TC18
##         0  0.7067  58168.139  -11654.131
## s.e.    0  0.1260   1216.847    1828.239
##
## sigma^2 = 5701242:  log likelihood = -293.42
## AIC=594.84   AICc=596.33   BIC=600.71
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -20.82493 2273.049 1848.262 -0.1958915 3.267152 0.4428731
##                    ACF1
## Training set 0.1054408
```

```
coefci(modelo2)
```

```
##                   2.5 %        97.5 %
## ar2        4.598268e-01     0.9535668
## intercept  5.578316e+04 60553.1160998
## TC18      -1.523741e+04 -8070.8489250
```

```
coeftest(modelo2)
```

```
##
```

```
## z test of coefficients:
##
##              Estimate  Std. Error z value  Pr(>|z|)
## ar2         7.0670e-01  1.2596e-01  5.6106 2.016e-08 ***
## intercept   5.8168e+04  1.2168e+03 47.8023 < 2.2e-16 ***
## TC18       -1.1654e+04  1.8282e+03 -6.3745 1.835e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.3. Diagnóstico del modelo

```
# Guardamos los residuos del modelo
residuos2 <- modelo2$residuals

# Buscamos los residuos máximos y mínimos

max(residuos2)
```

```
## [1] 5144.677
```

```
which.max(residuos2)
```

```
## [1] 26
```

```
time(residuos2)[which.max(residuos2)] # Junio de 2022
```

```
## [1] 2022.25
```

```
min(residuos2)
```

```
## [1] -5739.595
```

```
which.min(residuos2)
```

```
## [1] 10
```

```
time(residuos1)[which.min(residuos2)] # Junio de 2018
```

```
## [1] 2018.25
```

```
# Graficamos los residuos
residuos2 %>% autoplot() +
  labs(x = "Fecha",
       y = "Residuos") +
  geom_hline(yintercept = 0, color = "red")
```

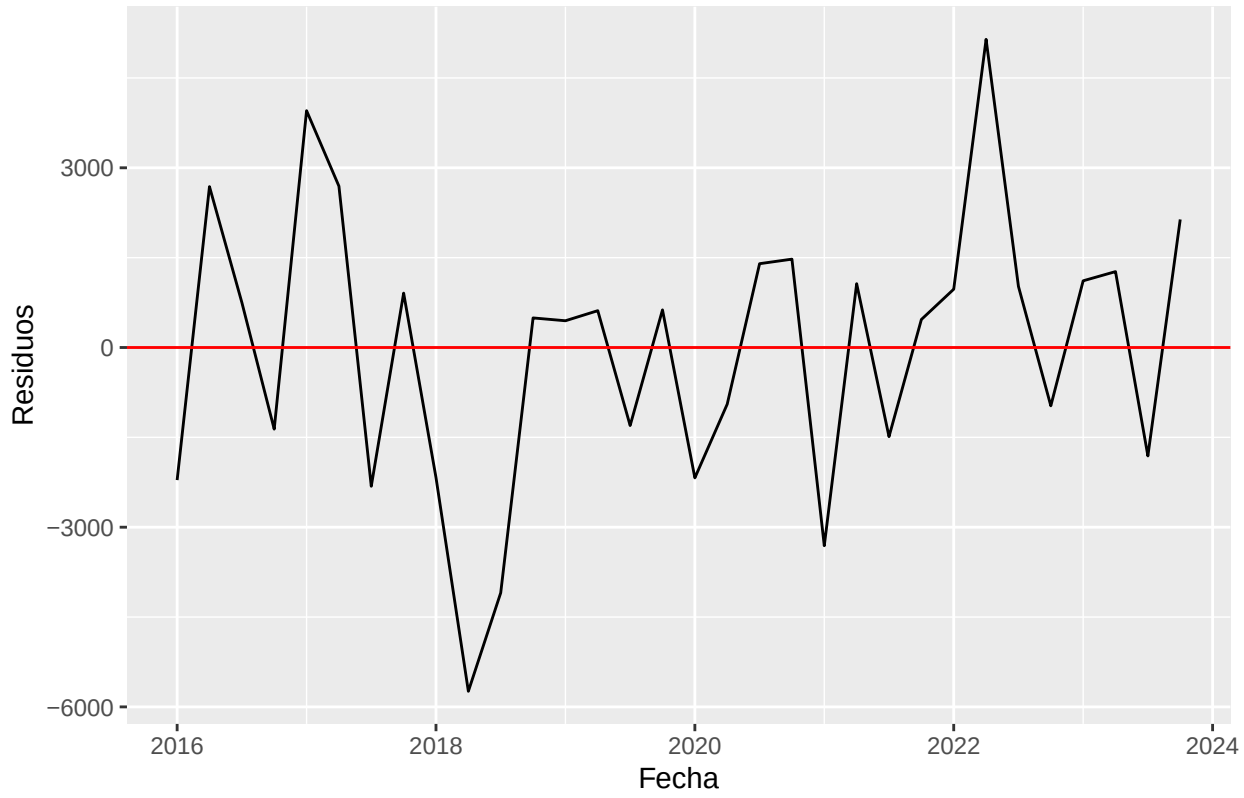


Figura 9: Residuos de un modelo AR(2) intervenido para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```
# Residuos estandarizados
residuos2_est <- residuos2/sqrt(modelo2$sigma2)
residuos2_est %>% autoplot() +
  labs(x = "Fecha",
       y = "Residuos estandarizados",
       title = "PIB comercio") +
  geom_hline(yintercept = 0, color = "black") +
  geom_hline(yintercept = 3, color = "red", linetype = "dotted") +
  geom_hline(yintercept = -3, color = "red", linetype = "dotted")
```

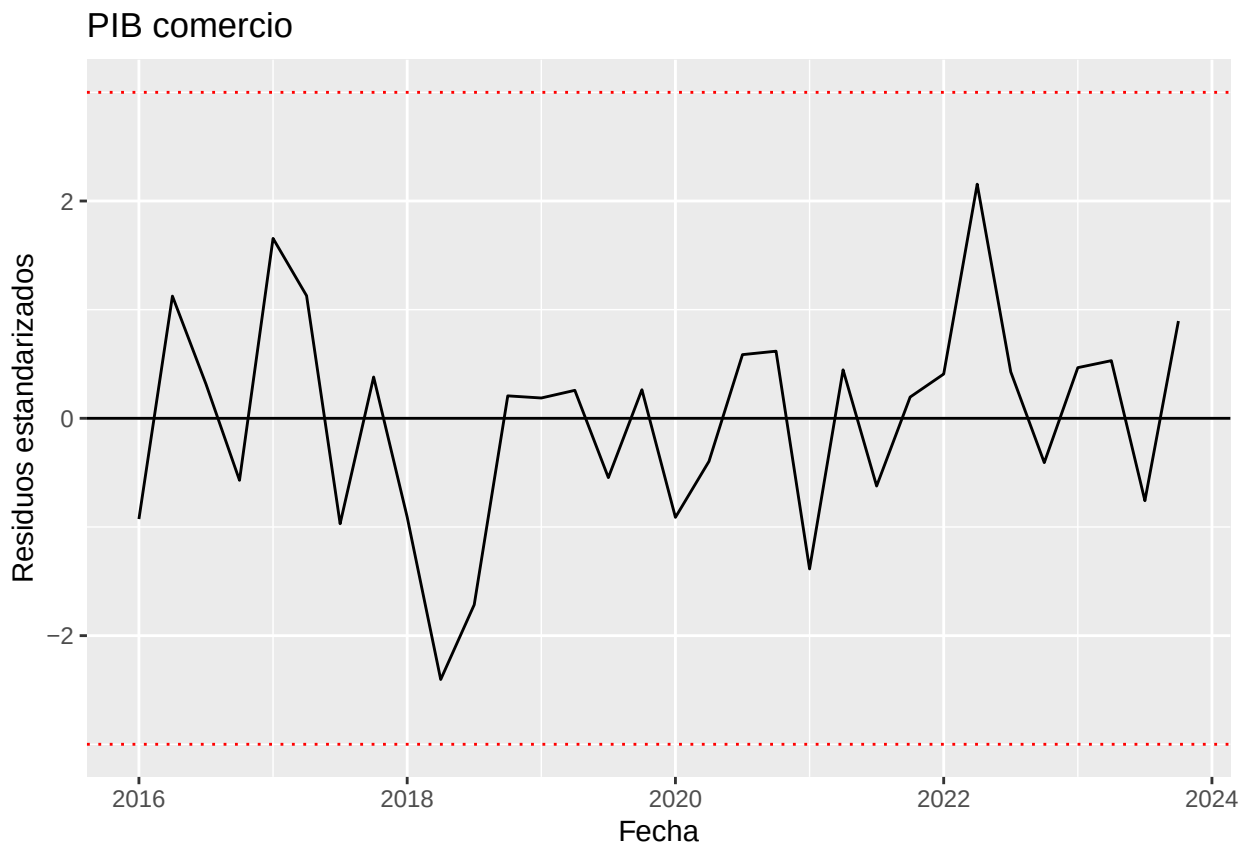


Figura 10: Residuos estandarizados de un modelo AR(2) intervenido para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```
# FAC
residuos_acf <- ggAcf(residuos2, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
residuos_pacf <- ggAcf(residuos2, lag.max = 24, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(residuos_acf, residuos_pacf)
```

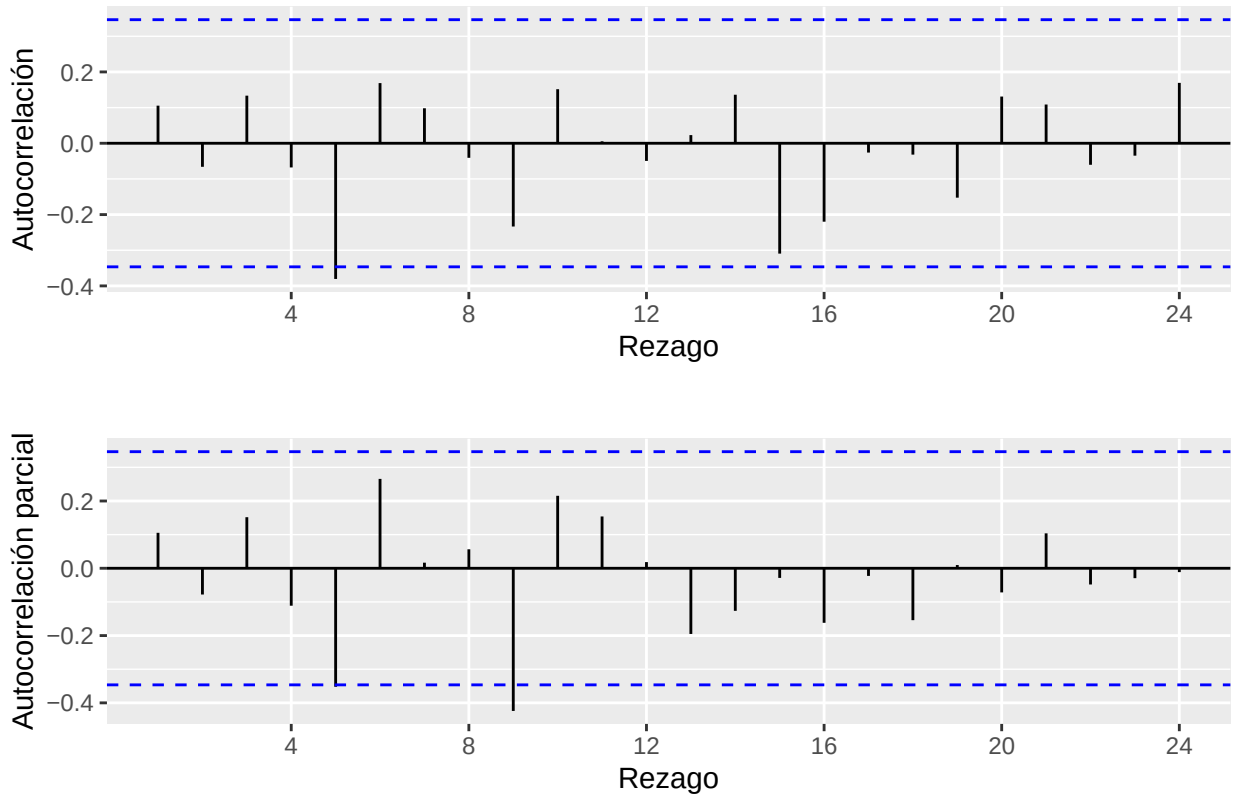


Figura 11: Funciones de Autocorrelación y Autocorrelación Parcial estimadas de los residuos de un modelo AR(2) intervenido para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```
# Test de Ljung-Box

# Se rechaza la hipótsis nula de no autocorrelación de los residuos
Box.test(residuos2,
         lag = 10,
         type = "Ljung-Box",
         fitdf = 2) # p + q de un modelo ARMA(p,q)
```

```
##
##  Box-Ljung test
##
## data:  residuos2
## X-squared = 12.625, df = 8, p-value = 0.1254
```

```
# Armamos el QQ-plot de los residuos
ggplot(residuos2, aes(sample = residuos2)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")
```

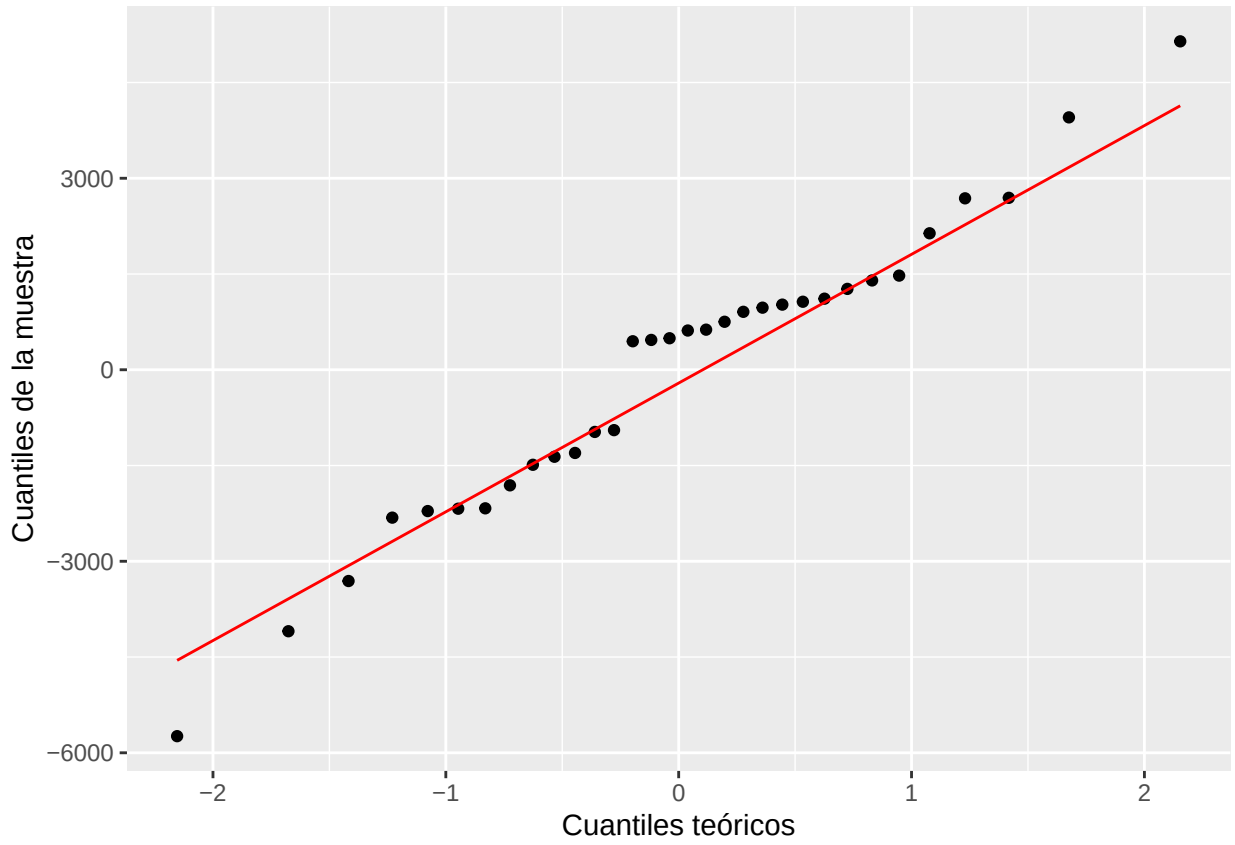


Figura 12: QQ-plot de los residuos de un modelo AR(2) intervenido para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

```
# Hacemos un histograma de los residuos
ggplot(data = residuos2) +
  geom_histogram(aes(x = residuos2, y = ..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos2),
                            sd = sd(residuos2)),
                col = "red",
                size = 1) +
  labs(x = "Residuos",
       y = "Densidad")
```

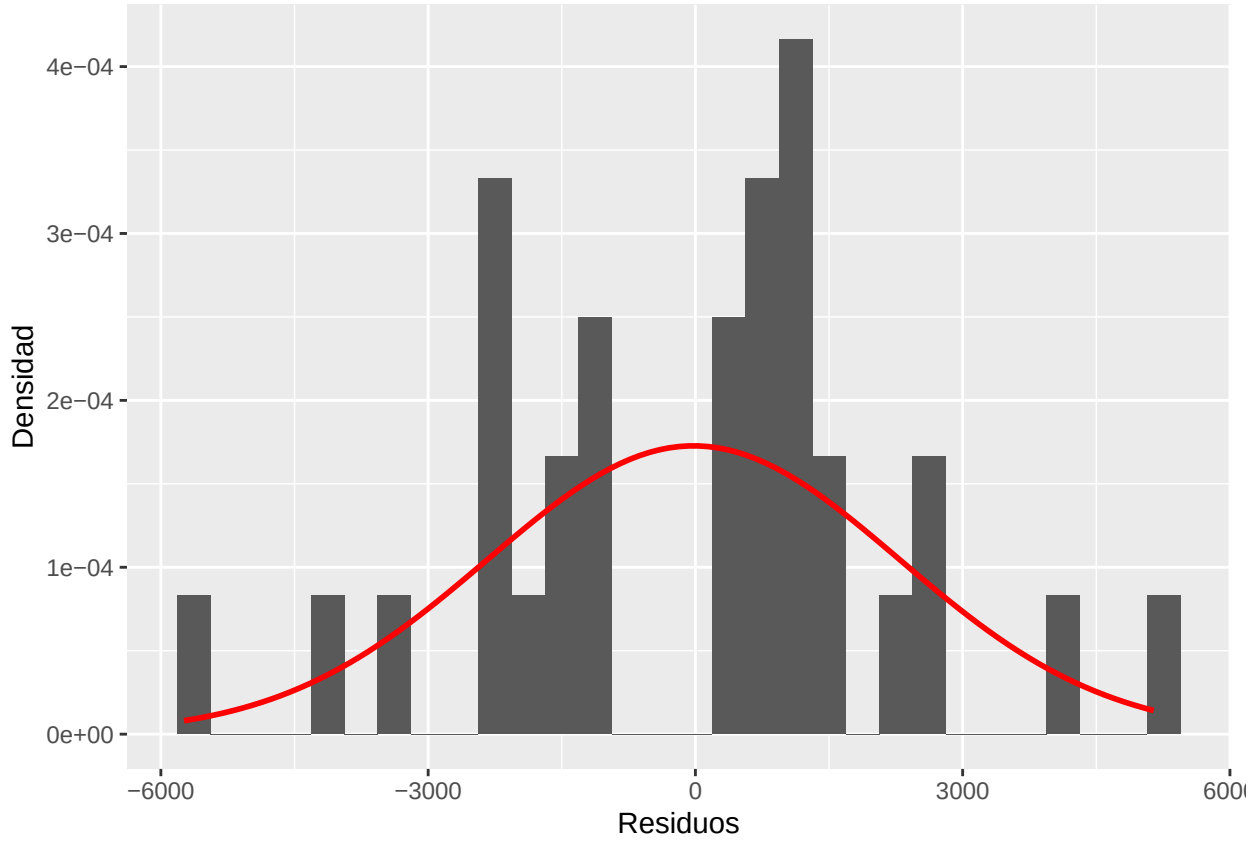


Figura 13: Histograma de los residuos de un modelo AR(2) intervenido para el PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023. La línea roja corresponde a una densidad normal con media y desvío muestrales igual al de los residuos.

```
# Tests de Shapiro y Jarque-Bera
# No se rechaza la hipótesis nula de normalidad dado que hay outliers
shapiro.test(residuos2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos2
## W = 0.97268, p-value = 0.5765
```

```
JarqueBera.test(residuos2)
```

```
##
##  Jarque Bera Test
##
## data:  residuos2
## X-squared = 0.28349, df = 2, p-value = 0.8678
##
##
##  Skewness
```

```
##
## data:  residuos2
## statistic = 0.21844, p-value = 0.6139
##
##
##  Kurtosis
##
## data:  residuos2
## statistic = 3.1475, p-value = 0.8648
```

5. Validación del modelo

5.1. Errores de predicción a un paso dentro de la muestra

```
# Obtenemos medidas de los errores de predicción a un paso dentro de la muestra (residuos)

accuracy(modelo2)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -20.82493 2273.049 1848.262 -0.1958915 3.267152 0.4428731
##                    ACF1
## Training set 0.1054408
```

5.2. Ajuste fuera de la muestra

```
# Definimos una muestra de entrenamiento ("training set") hasta 2022 inclusive
train_comercio <- window(comercio, end = c(2022,4))

# Dejamos los datos de 2023 como conjunto de entrenamiento ("test set")
test_comercio <- window(comercio, start = 2023)
n <- length(test_comercio)

# Cortamos el regresor correspondiente al TC
xreg_train <- xreg[1:length(train_comercio)]
xreg_test <- xreg[(length(train_comercio)+1):length(comercio)]
```

```
# Estimamos los modelos para el training set

modelo2_train <- Arima(y = train_comercio,
                  order = c(2, 0, 0),
                  lambda = NULL,
                  xreg = xreg_train,
                  fixed = c(0, NA, NA, NA))
```

```
# Predecimos fuera de la muestra (el horizonte de predicción
# será igual al largo del test set)

pred2_test <- forecast(modelo2_train, h = n, xreg = xreg_test)
```

```r
# Graficamos las predicciones obtenidas

grafico_pred2_test <- autoplot(pred2_test) +
  autolayer(comercio, color = "black") +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")

grafico_pred2_test
```

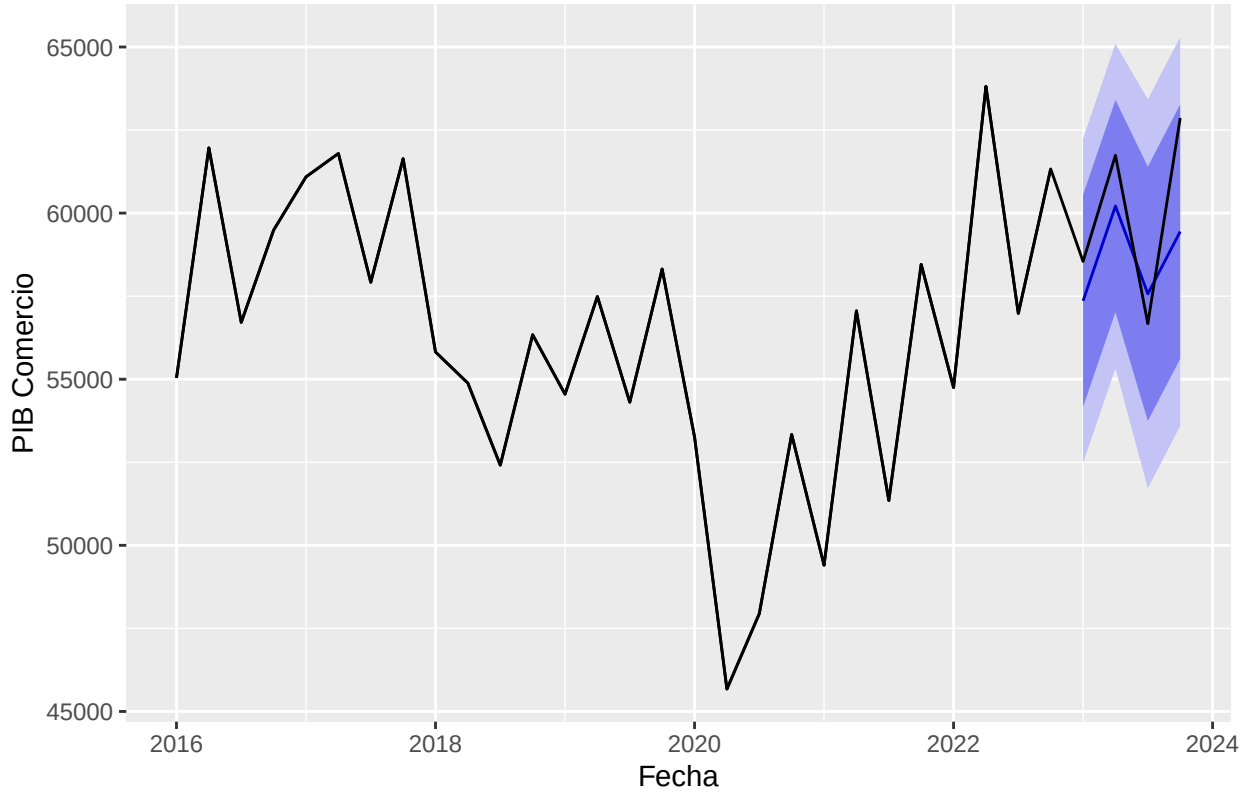


Figura 14: Predicciones en el conjunto de prueba del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para un modelo AR(2) intervenido. La línea azul corresponde a las predicciones.

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

accuracy(pred2_test, test_comercio)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set  -41.41985 2353.976 1879.526 -0.2429185 3.338918 0.4133009
## Test set     1308.88479 2015.379 1757.701  2.0888510 2.880793 0.3865121
##                    ACF1 Theil's U
```

```
## Training set  0.1338293        NA
## Test set     -0.5506493 0.4576695
```