

# Taller 2 - Predicción

Adaptado del taller 2022 de Federico Molina

Series Cronológicas 2024

Abril 2024

## 1. Exploración de los datos

En este taller, se trabajará con la serie del Producto Interno Bruto correspondiente al sector “Comercio, alojamiento y suministro de comidas y bebidas”, medido en millones de pesos a precios constantes de 2016. Se cuenta con datos trimestrales entre marzo de 2016 y diciembre de 2023. De esta manera, se cuenta con 32 observaciones.

Antes de pasar a la etapa de modelización, es importante explorar los datos mediante distintos gráficos. Los mismos permiten visualizar distintos aspectos como:

- Tendencias y/o media no constante
- Varianza no constante
- Valores atípicos
- Estacionalidad

```
# Datos trimestrales (32 observaciones)  
frequency(comercio)
```

```
## [1] 4
```

```
length(comercio)
```

```
## [1] 32
```

```
# No es conveniente usar la función summary() porque no considera  
# la estructura temporal de los datos  
# summary(comercio)
```

```
# Máximo valor observado y fecha en la que ocurrió  
max(comercio)
```

```
## [1] 63814.26
```

```
time(comercio)[which.max(comercio)]
```

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

```
# Mínimo valor observado y fecha en la que ocurrió
min(comercio)
```

```
## [1] 45673.51
```

```
time(comercio)[which.min(comercio)]
```

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

```
# 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.

```
# La función ggmonthplot() permite observar el comportamiento por
# trimestre, mensual, etc. de una serie y tener un primer indicio
# de si existe un componente estacional o no
ggmonthplot(comercio) +
```

```
labs(x = "Trimestres",
     y = "PIB Comercio")
```

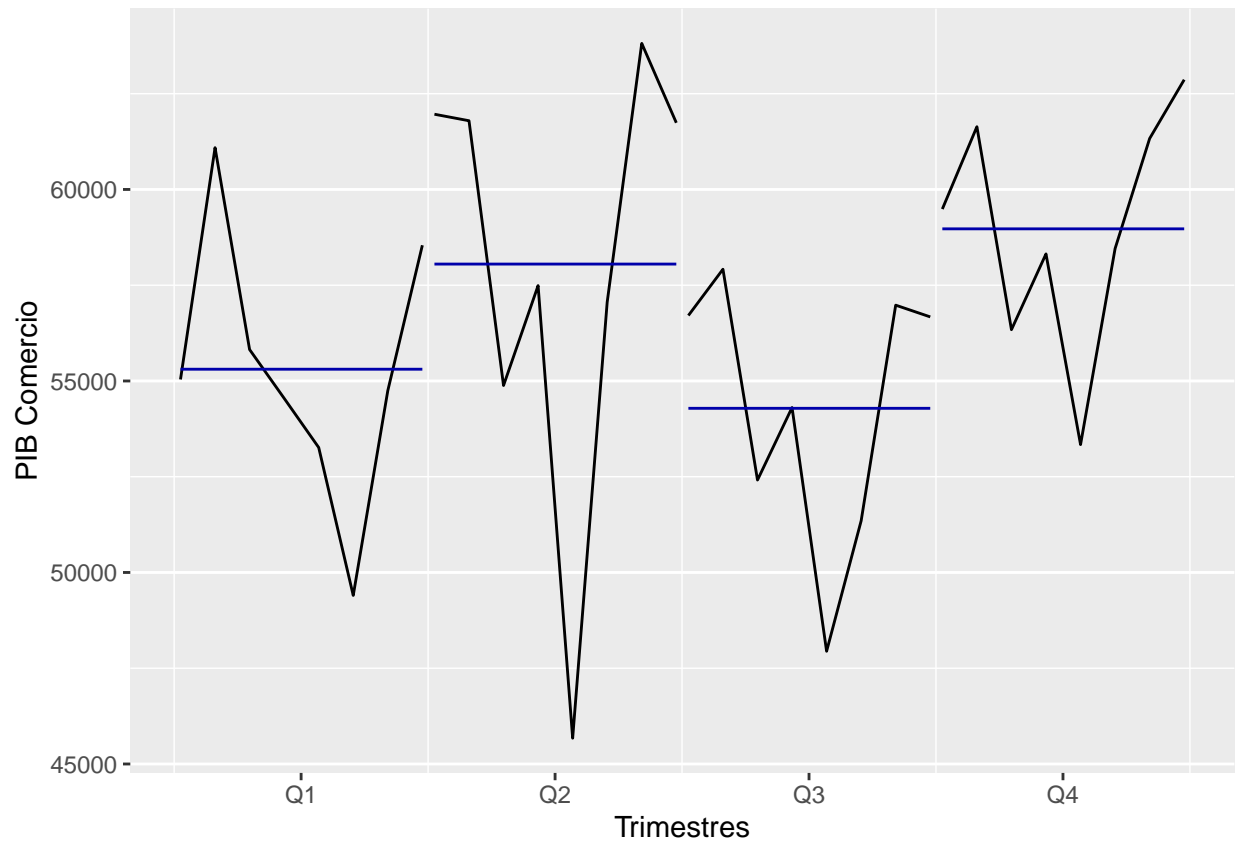


Figura 2: Comportamiento trimestral del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

*# Otras formas:*

```
# R base: función monthplot
# monthplot(comercio, xlab = "Trimestres",
#           ylab = "PIB Comercio")
```

```
# ggplot: Función ggsubseriesplot()
# ggsubseriesplot(comercio) +
#   labs(x = "Trimestres",
#        y = "PIB Comercio")
```

*# La función ggseasonplot() permite comparar el comportamiento por año*  
*# de una serie y advertir tendencias, estacionalidad, etc.*

```
ggseasonplot(comercio) +
  labs(color = "Año",
       x = "Trimestre",
       y = "PIB Comercio",
       title = "")
```

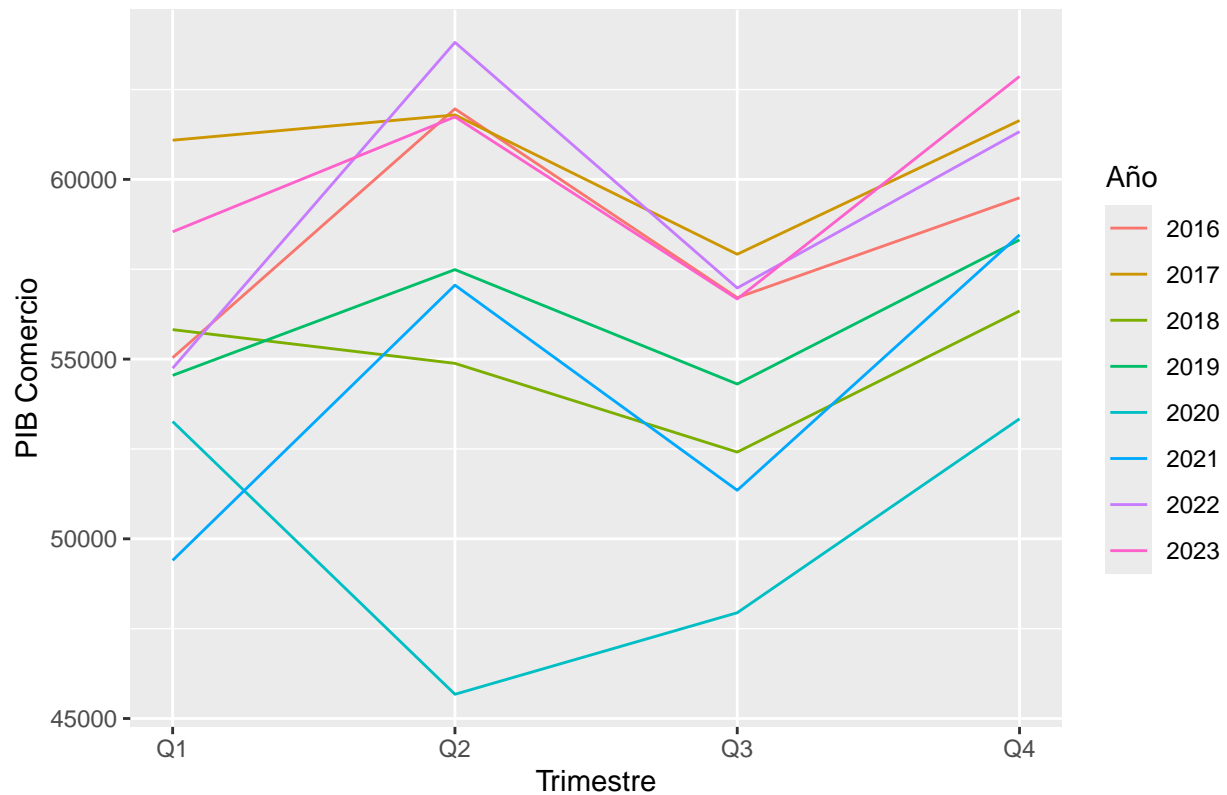


Figura 3: Comportamiento por año del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

## 2. Identificación y estimación del modelo

En la etapa de exploración, se obtiene una primera idea acerca del comportamiento de la serie y de si es estacionaria o no. Para verificarlo, en primera instancia se estima la Función de Autocorrelación. Si se observa una caída relativamente rápida en las autocorrelaciones simples, no se descarta que el proceso sea estacionario y se pasa a la etapa de identificación. Para ello, se utiliza también la Función de Autocorrelación Parcial.

```
# 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)
```

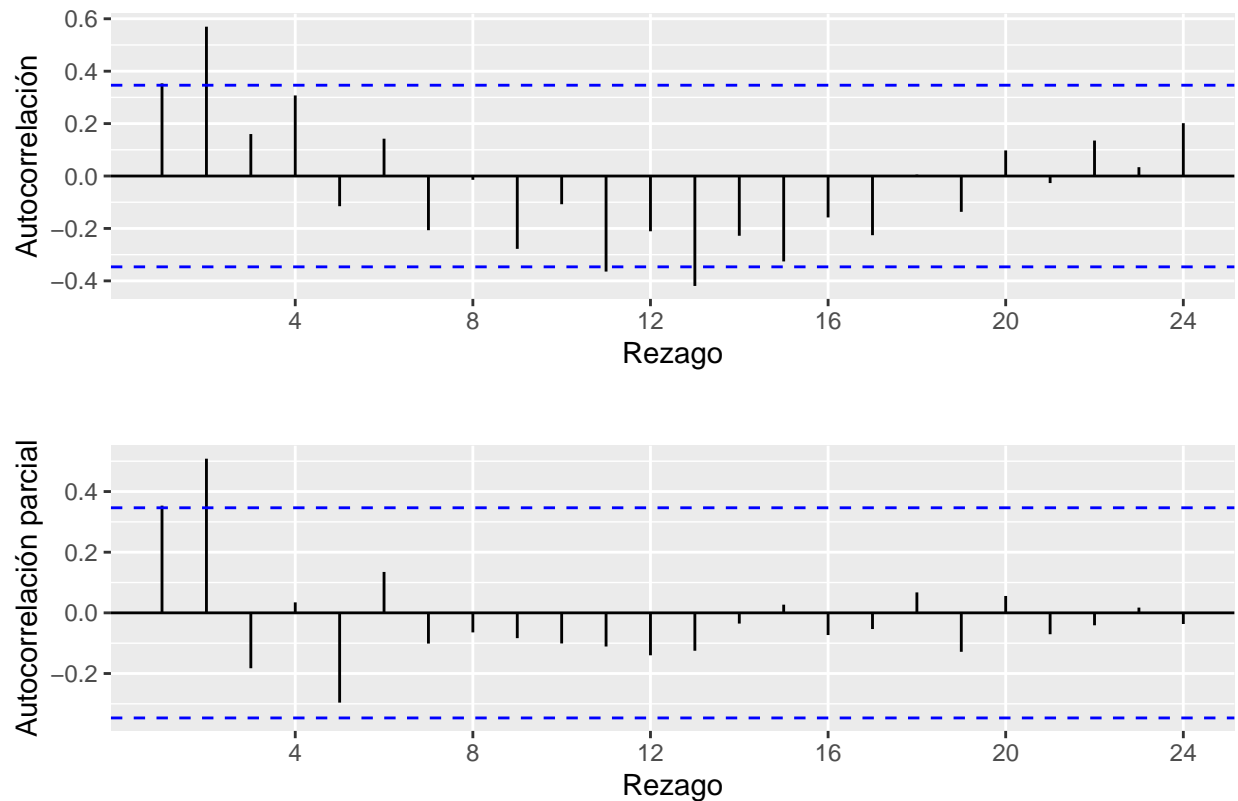


Figura 4: Funciones de Autocorrelación y Autocorrelación Parcial estimadas del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023.

## 2.1. Serie original

```
# Dada la forma de la FAC y la FACP, dos posibles modelos son AR(2) y MA(2).

# AR(2)
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

summary(modelo1)

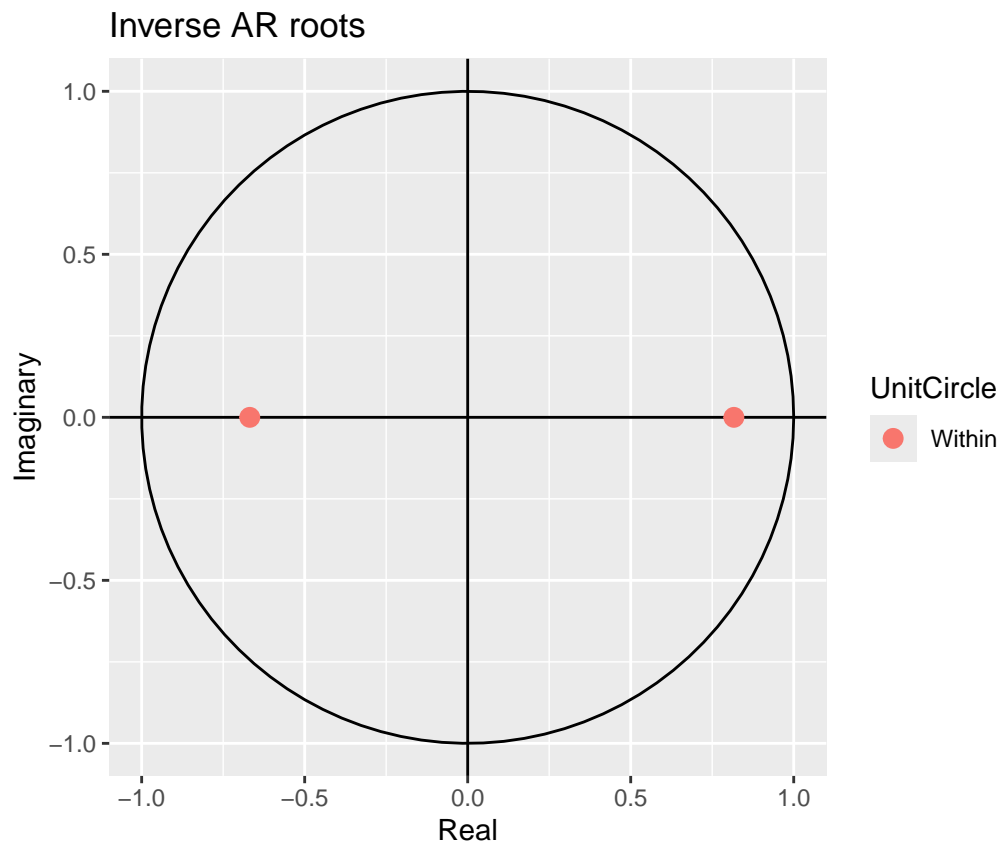
## Series: comercio
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      mean
##          0.1481  0.5458 57151.079
## s.e.      0.1488  0.1518 1767.707
##
```

```
## sigma^2 = 12609736: log likelihood = -305.84
## AIC=619.68 AICc=621.16 BIC=625.54
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -38.74251 3380.469 2531.523 -0.4620817 4.61813 0.6065934 0.1349752
```

```
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
```

```
autoplot(modelo1)
```



```
# 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

```

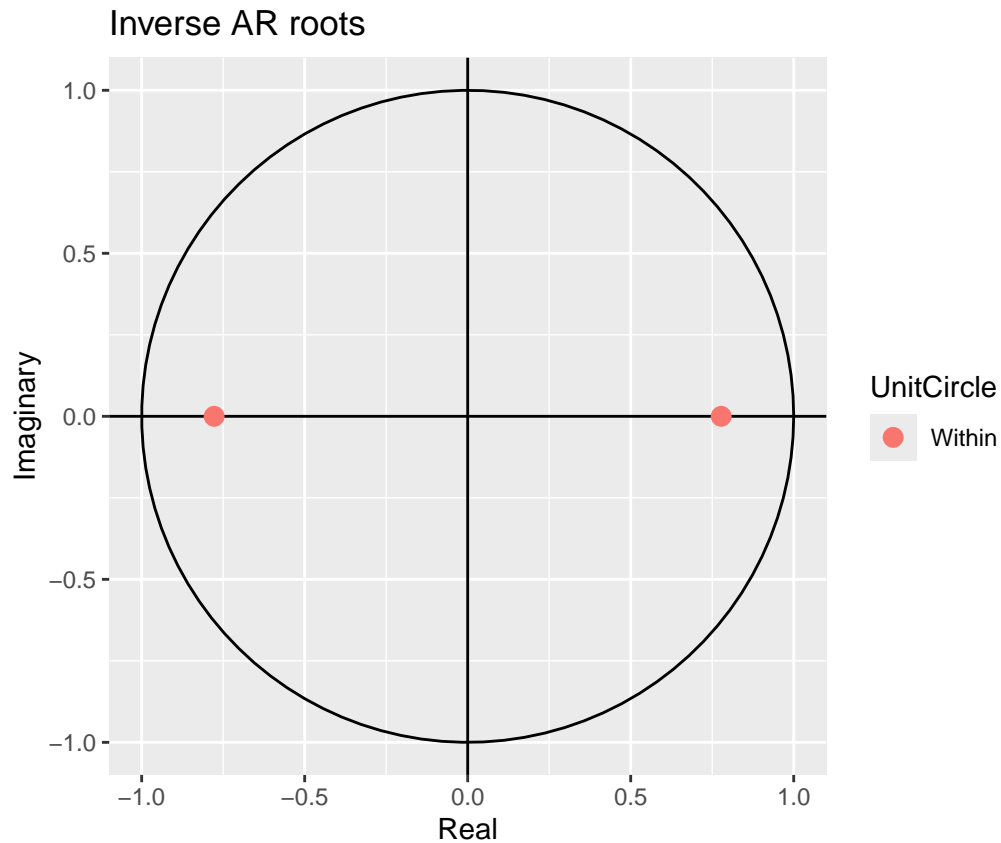
```
coeftest(modelo1)
```

```

##
## 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

```

```
autoplot(modelo1)
```



```
# MA(2)
modelo2 <- Arima(y = comercio,
                 order = c(0, 0, 2),
                 lambda = NULL)
```

```
summary(modelo2)
```

```
## Series: comercio
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1      ma2      mean
##          0.1665  0.4681 56765.828
## s.e.  0.1851  0.1394 1017.362
##
## sigma^2 = 14245372: log likelihood = -307.64
## AIC=623.27  AICc=624.75  BIC=629.13
##
## Training set error measures:
##              ME    RMSE    MAE    MPE    MAPE    MASE    ACF1
## Training set -26.40074 3593.03 2768.523 -0.517579 5.032565 0.6633826 0.09846745
```

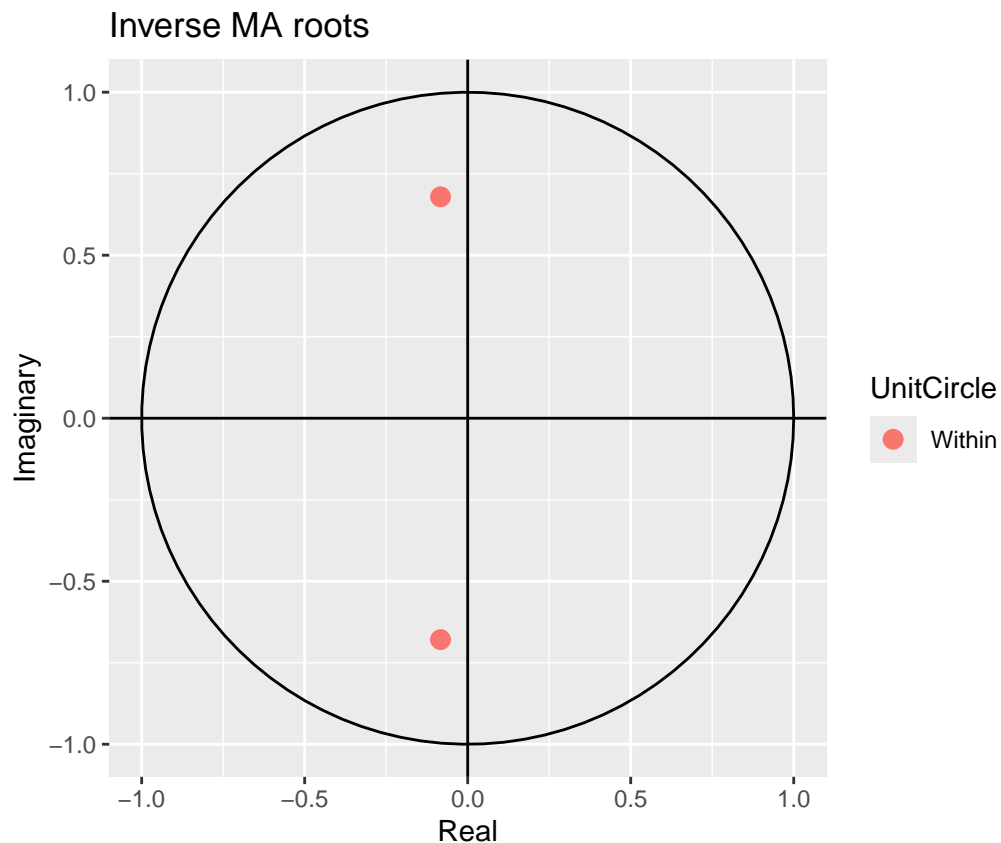
```
coeftest(modelo2)
```

```
##
```



```
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ma1      1.6650e-01 1.8515e-01  0.8993 0.3684988
## ma2      4.6811e-01 1.3937e-01  3.3587 0.0007831 ***
## intercept 5.6766e+04 1.0174e+03 55.7971 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
autoplot(modelo2)
```



```
# Probamos un MA(2) con theta1 = 0
modelo2 <- Arima(y = comercio,
  order = c(0, 0, 2),
  lambda = NULL,
  fixed = c(0, NA, NA))

summary(modelo2)
```

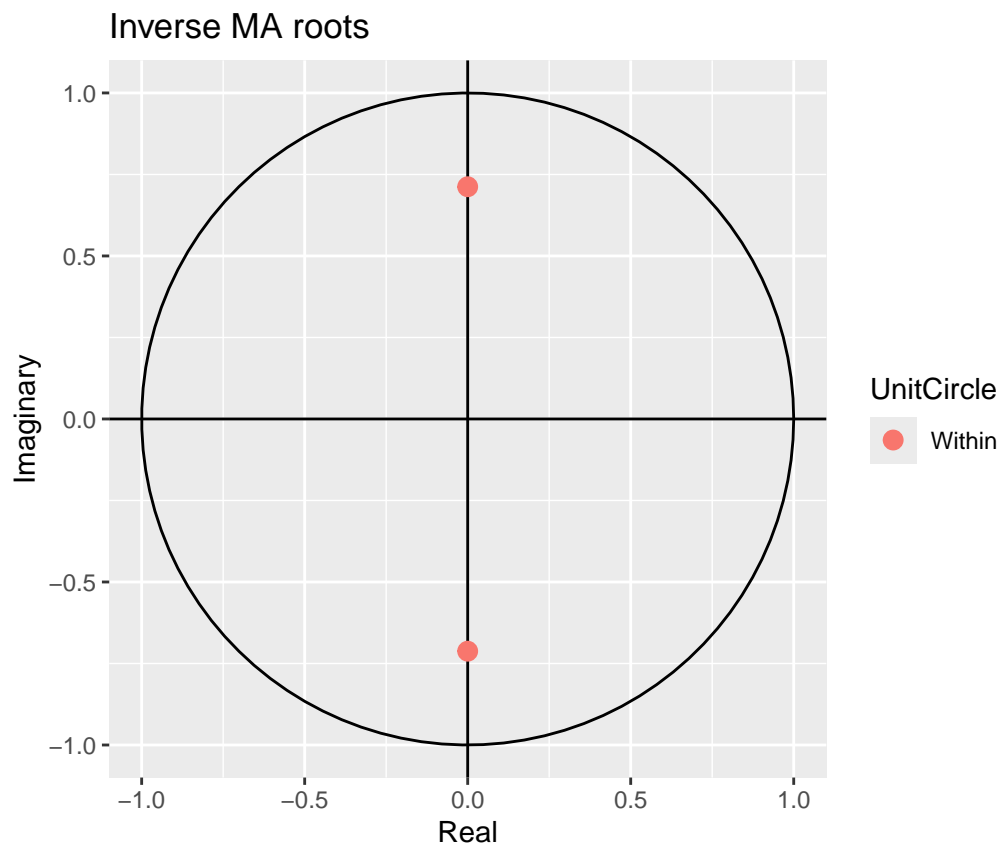
```
## Series: comercio
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##      ma1      ma2      mean
##      0  0.5073 56757.6694
```

```
## s.e.    0  0.1302    948.7529
##
## sigma^2 = 14087541:  log likelihood = -308.04
## AIC=622.09  AICc=622.94  BIC=626.48
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -41.59821 3634.153 2685.229 -0.5520701 4.905562 0.6434239
##           ACF1
## Training set 0.2436876
```

```
coeftest(modelo2)
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ma2          5.0733e-01 1.3019e-01  3.8969 9.745e-05 ***
## intercept 5.6758e+04 9.4875e+02 59.8234 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
autoplot(modelo2)
```



```
# Nos quedamos con los modelos sin phi1 ni theta1
```

```
# Graficamos la serie y los valores ajustados
```

```
# Modelo 1
```

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

```
# Modelo 2
```

```
fit2 <- autoplot(comercio) +  
  autolayer(modelo2$fitted, color = "blue") +  
  labs(x = "Fecha",  
       y = "PIB Comercio",  
       title = "Modelo MA(2)")
```

```
grid.arrange(fit1, fit2)
```

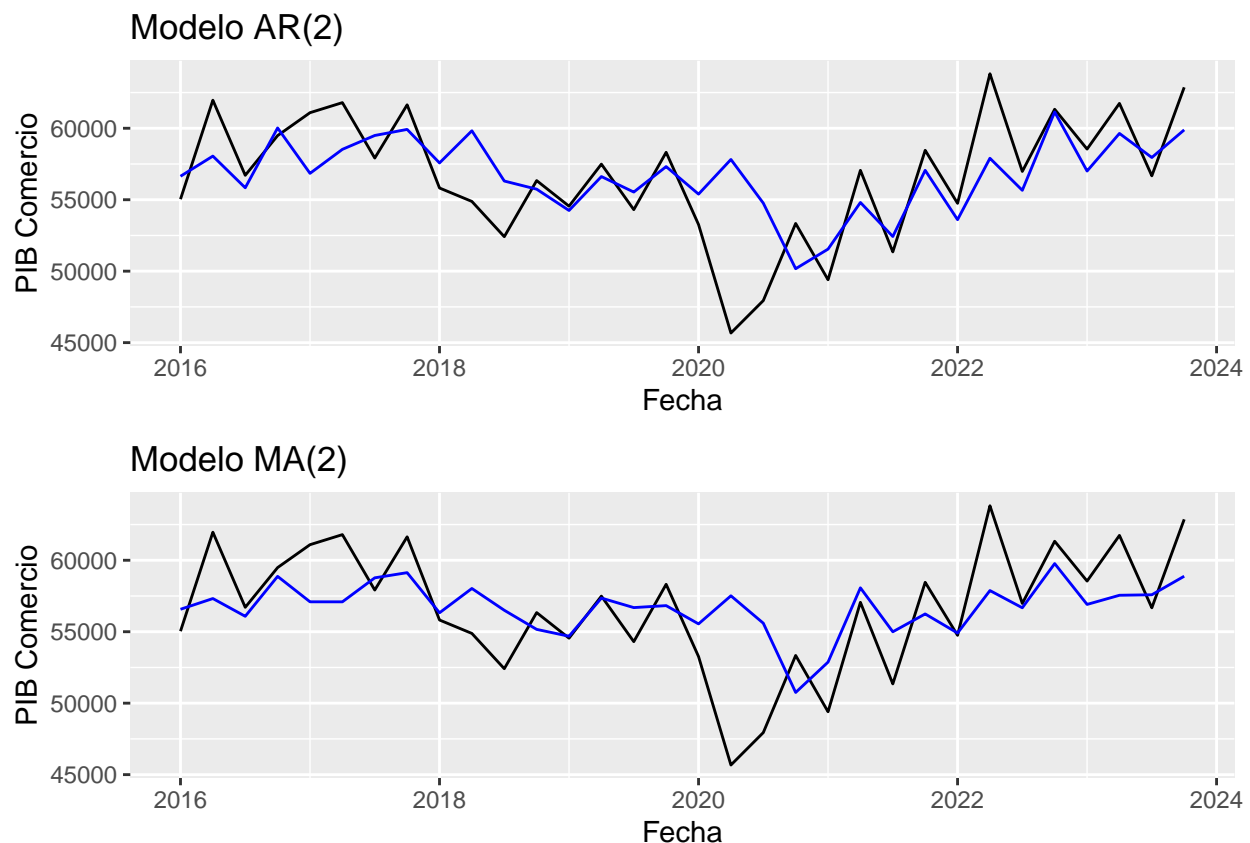


Figura 5: PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023 y valores ajustados para modelos AR(2) y MA(2). La línea negra corresponde a los valores reales y la azul a los ajustados.

## 2.2. Serie transformada

```
# Dada la forma de la FAC y la FACP, dos posibles modelos son AR(1) y AR(2)

# AR(2)
modelo1_trans <- Arima(y = comercio, # Datos para estimar
  order = c(2, 0, 0), # Orden del modelo (suponemos estacionariedad)
  lambda = 0, # Aplicamos una transformación logarítmica
  biasadj = TRUE, # Ajustamos el sesgo en los valores ajustados
  #dado por las transformaciones
  fixed = c(0, NA, NA))
```

```
summary(modelo1_trans)
```

```
## Series: comercio
## ARIMA(2,0,0) with non-zero mean
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1          ar2          mean
##           0  0.5808  10.9483
## s.e.        0  0.1422   0.0249
##
## sigma^2 = 0.004286: log likelihood = 42.45
## AIC=-78.91  AICc=-78.05  BIC=-74.51
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -78.85164 3441.052 2502.201 -0.5519437 4.590366 0.5995676
##              ACF1
## Training set 0.2759179
```

```
coeftest(modelo1_trans)
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error  z value  Pr(>|z|)
## ar2         0.580770   0.142243   4.0829 4.447e-05 ***
## intercept 10.948255   0.024891 439.8513 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# MA(2)
modelo2_trans <- Arima(y = comercio,
  order = c(0, 0, 2),
  lambda = 0,
  biasadj = TRUE,
  fixed = c(0, NA, NA))
```

```
summary(modelo2_trans)
```

```
## Series: comercio
## ARIMA(0,0,2) with non-zero mean
## Box Cox transformation: lambda= 0
##
## Coefficients:
##      ma1      ma2      mean
##      0  0.4839  10.9435
## s.e.    0  0.1304   0.0172
##
## sigma^2 = 0.004755: log likelihood = 40.94
## AIC=-75.88   AICc=-75.02   BIC=-71.48
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -31.3535 3641.158 2700.82 -0.5368607 4.929653 0.6471597 0.2494239
```

```
coeftest(modelo2_trans)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma2      0.483938   0.130399   3.7112 0.0002063 ***
## intercept 10.943474   0.017171 637.3242 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Graficamos la serie y los valores ajustados
```

```
# Modelo 1
```

```
fit1_trans <- autoplot(comercio) +
  autolayer(modelo1_trans$fitted, color = "blue") +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "Modelo AR(2)")
```

```
# Modelo 2
```

```
fit2_trans <- autoplot(comercio) +
  autolayer(modelo2_trans$fitted, color = "blue") +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "Modelo MA(2)")
```

```
grid.arrange(fit1_trans, fit2_trans)
```

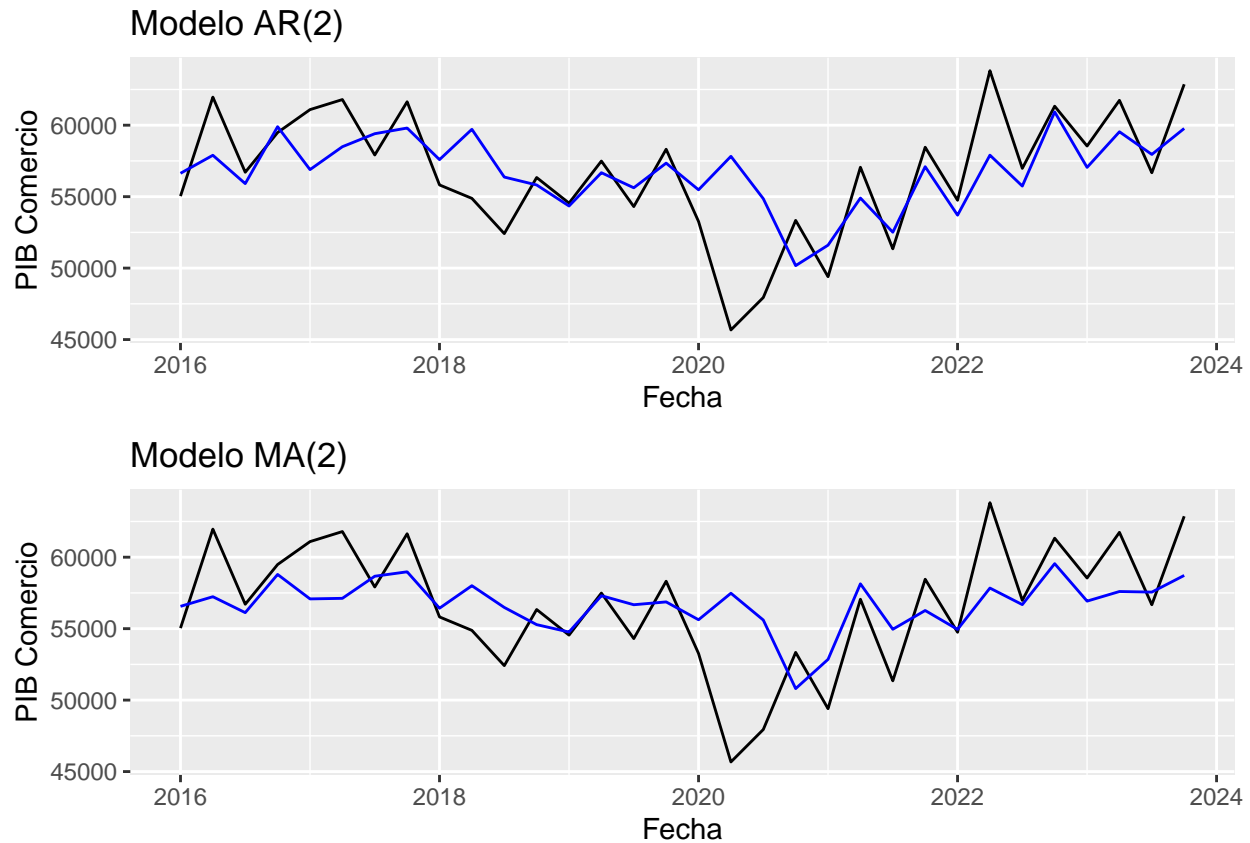


Figura 6: PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) entre 2016 y 2023 y valores ajustados para modelos AR(2) y MA(2) (modelo estimado para la serie transformada). La línea negra corresponde a los valores reales y la azul a los ajustados.

### 3. Predicción

En lo que sigue, se supone que los modelos fueron correctamente identificados y estimados, por lo que puede pasarse directamente a la etapa de predicción. No se considera la posibilidad de que la serie no sea estacionaria y/o que tenga estacionalidad. Es decir, que se saltea la etapa de diagnóstico y no se verifican los distintos supuestos acerca de los residuos (media nula, varianza constante, no autocorrelación y normalidad).

#### 3.1. Serie original

##### 3.1.1. Cálculo de predicciones

```
# Obtenemos las predicciones con la función forecast() del paquete forecast
# Por defecto, se predice a 10 pasos para las series anuales y a 2 años
# para las series de mayor frecuencia (ver argumento h en ?forecast)

# Modelo 1
pred1 <- forecast(modelo1)
pred1 # Por defecto, tenemos la estimación puntual y los
```

##		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
##	2024 Q1	56822.93	52285.17	61360.69	49883.02	63762.84
##	2024 Q2	60567.50	56029.73	65105.26	53627.59	67507.40
##	2024 Q3	56913.65	51610.83	62216.48	48803.68	65023.62
##	2024 Q4	59177.87	53875.05	64480.70	51067.90	67287.85
##	2025 Q1	56968.51	51412.19	62524.82	48470.86	65466.16
##	2025 Q2	58337.61	52781.30	63893.93	49839.96	66835.26
##	2025 Q3	57001.68	51355.52	62647.83	48366.63	65636.73
##	2025 Q4	57829.53	52183.38	63475.69	49194.48	66464.58

```
# intervalos de confianza al 80% y 95%
```

```
# Modelo 2
```

```
pred2 <- forecast(modelo2)
```

```
pred2
```

##		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
##	2024 Q1	56294.65	51484.56	61104.75	48938.25	63651.06
##	2024 Q2	58777.83	53967.73	63587.92	51421.42	66134.23
##	2024 Q3	56757.67	51363.96	62151.38	48508.71	65006.63
##	2024 Q4	56757.67	51363.96	62151.38	48508.71	65006.63
##	2025 Q1	56757.67	51363.96	62151.38	48508.71	65006.63
##	2025 Q2	56757.67	51363.96	62151.38	48508.71	65006.63
##	2025 Q3	56757.67	51363.96	62151.38	48508.71	65006.63
##	2025 Q4	56757.67	51363.96	62151.38	48508.71	65006.63

### 3.1.2. Gráfico de predicciones

```
# Graficamos las predicciones obtenidas
```

```
# Modelo 1
```

```
grafico_pred1 <- autoplot(pred1) +  
  labs(x = "Fecha",  
        y = "PIB Comercio",  
        title = "")
```

```
# Modelo 2
```

```
grafico_pred2 <- autoplot(pred2) +  
  labs(x = "Fecha",  
        y = "PIB Comercio",  
        title = "")
```

```
grid.arrange(grafico_pred1, grafico_pred2)
```

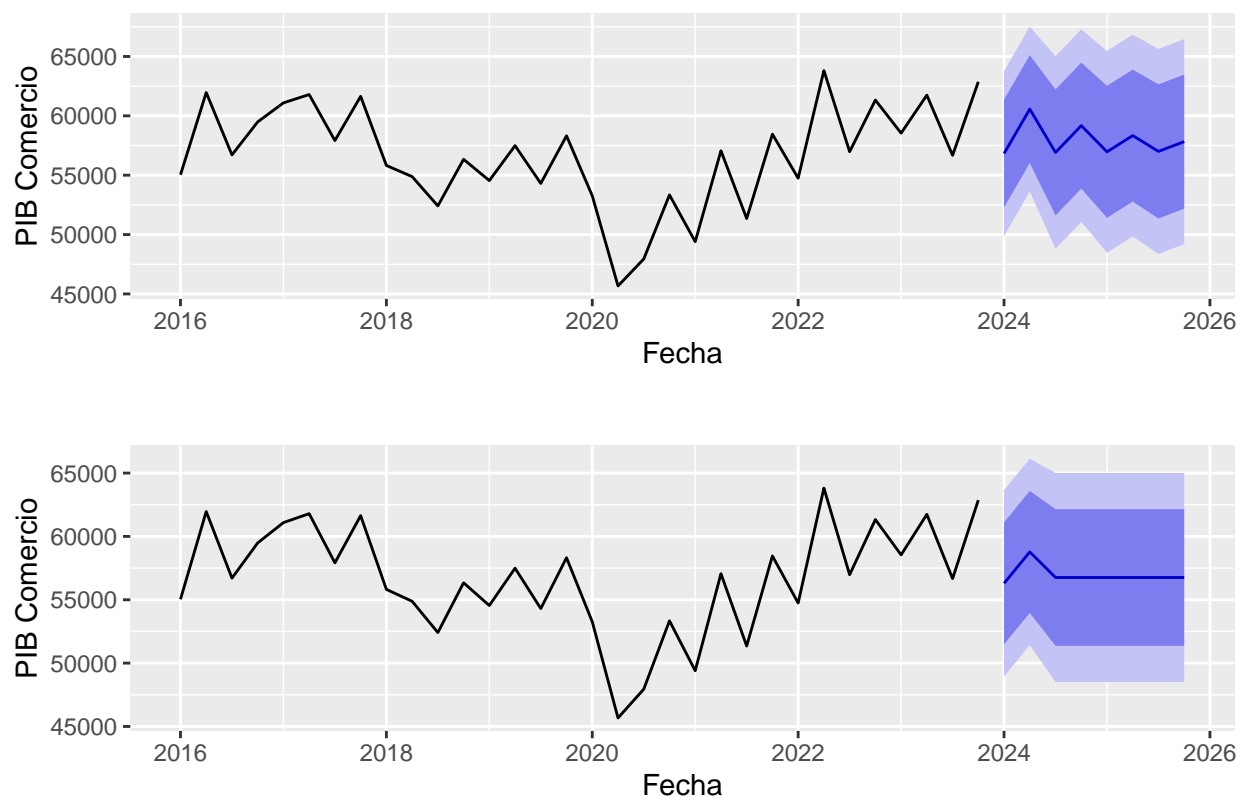


Figura 7: Predicciones a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para modelos AR(2) y MA(2). Se consideraron intervalos de confianza al 80 % y 95 %.

```
# Otra forma: usar la función plot() de R base
```

```
# Modelo 1
# plot(pred1,
#       xlab = "Fecha",
#       ylab = "PIB Comercio",
#       main = "")
```

```
# Modelo 2
# plot(pred2,
#       xlab = "Fecha",
#       ylab = "PIB Comercio",
#       main = "")
```

```
# Graficamos las predicciones puntuales obtenidas (media condicional sin
# intervalos de confianza)
```

```
# Modelo 1
grafico_pred1_sinPI <- autoplot(pred1, PI = FALSE) +
  labs(x = "Fecha",
       y = "PIB Comercio",
```



```

    title = "")

# Modelo 2
grafico_pred2_sinPI <- autoplot(pred2, PI = FALSE) +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")

grid.arrange(grafico_pred1_sinPI, grafico_pred2_sinPI)

```

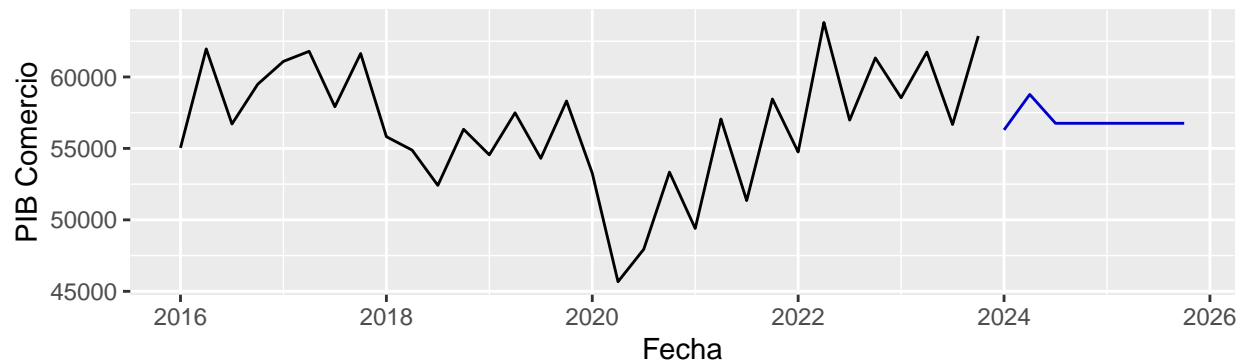
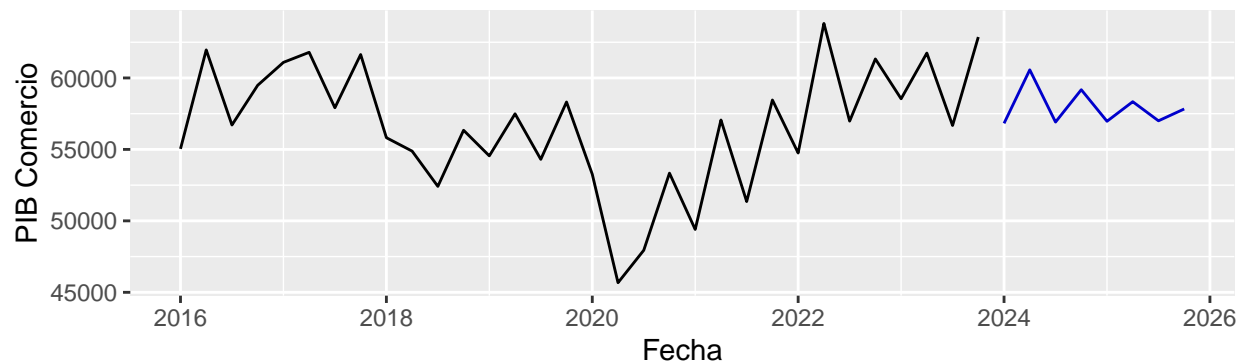


Figura 8: Predicciones a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para modelos AR(2) y MA(2).

```

# Otra forma: usar la función plot() de R base

# Modelo 1
# plot(pred1,
#       xlab = "Fecha",
#       ylab = "PIB Comercio",
#       main = "", PI = FALSE)

# Modelo 2
# plot(pred2,
#       xlab = "Fecha",

```

```
#      ylab = "PIB Comercio",
#      main = "", PI = FALSE)
```

```
# Graficamos las predicciones obtenidas para distintos intervalos de confianza
```

```
# Modelo 1
```

```
pred1_int <- forecast(modelo1, level = seq(50, 90, by = 10))
grafico_pred1_int <- autoplot(pred1_int) +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")
```

```
# Modelo 2
```

```
pred2_int <- forecast(modelo2, level = seq(50, 90, by = 10))
grafico_pred2_int <- autoplot(pred2_int) +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")
```

```
grid.arrange(grafico_pred1_int, grafico_pred2_int)
```

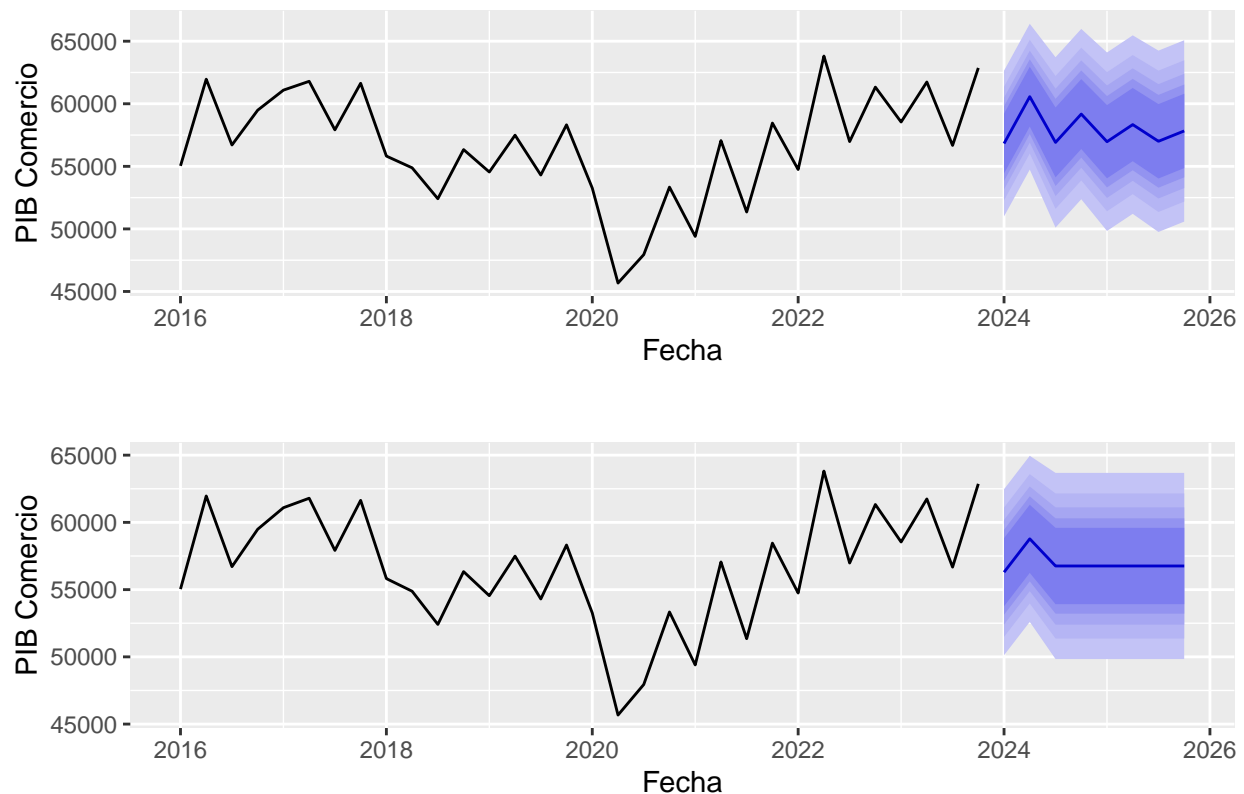


Figura 9: Predicciones a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para modelos AR(2) y MA(2). Se consideraron intervalos de confianza del 50 %, 60 %, 70 %, 80 % y 90 %.

```
# Graficamos las predicciones obtenidas mediante un fan chart
```

```
# Modelo 1
```

```
pred1_fan <- forecast(modelo1, fan = TRUE)
grafico_pred1_fan <- autoplot(pred1_fan) +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")
```

```
# Modelo 2
```

```
pred2_fan <- forecast(modelo2, fan = TRUE)
grafico_pred2_fan <- autoplot(pred2_fan) +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")
```

```
grid.arrange(grafico_pred1_fan, grafico_pred2_fan)
```

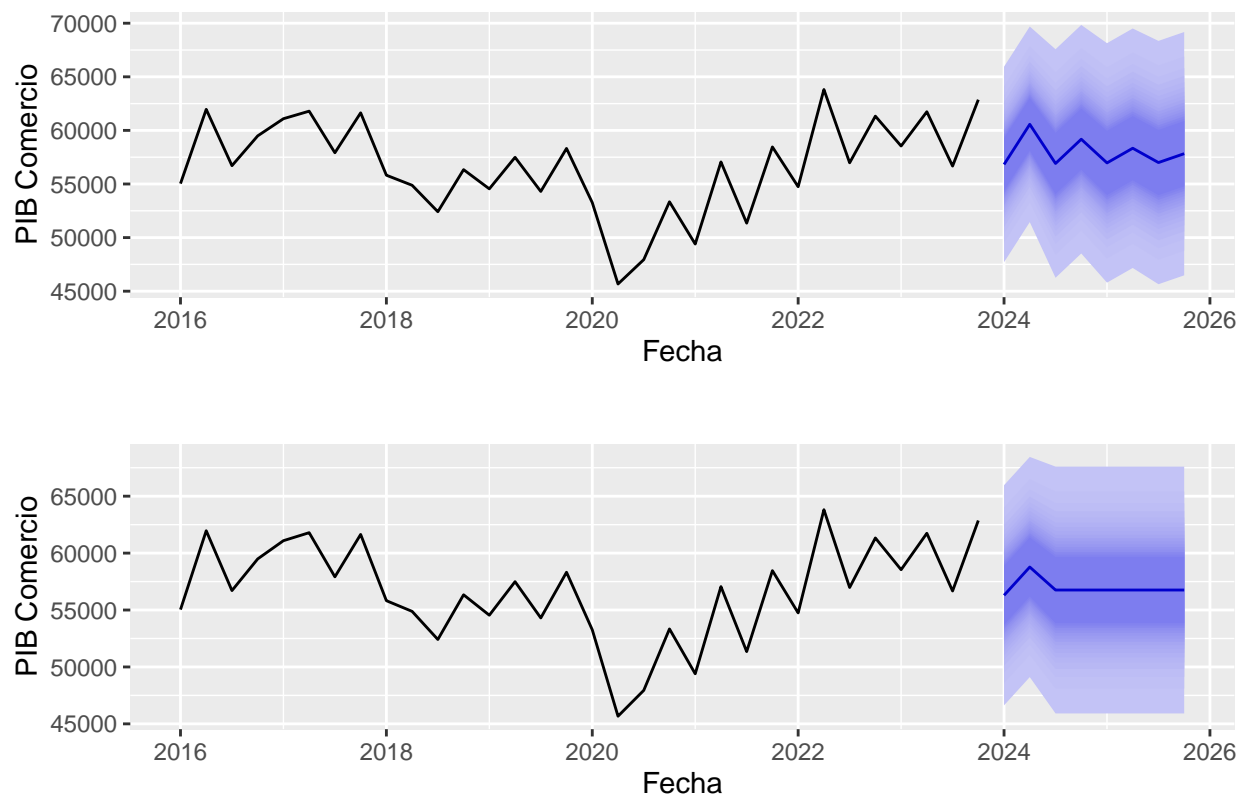


Figura 10: Predicciones a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para modelos AR(2) y MA(2). Se consideraron intervalos de confianza entre el 51 % y 99 %.

### 3.1.3. Bootstrap

*# Si los residuos no cumplen con el supuesto de normalidad, una alternativa para  
# obtener su distribución es emplear técnicas de bootstrap.*

*# Modelo 1*

```
pred1_boot <- forecast(modelo1, bootstrap = TRUE, npaths = 5000)
```

*# Se toman 5000 muestras con reposición de la muestra original*

```
grafico_pred1_boot <- autoplot(pred1_boot) +
```

```
  labs(x = "Fecha",  
        y = "PIB Comercio",  
        title = "")
```

*# Modelo 2*

```
pred2_boot <- forecast(modelo2, bootstrap = TRUE, npaths = 5000)
```

```
grafico_pred2_boot <- autoplot(pred2_boot) +
```

```
  labs(x = "Fecha",  
        y = "PIB Comercio",  
        title = "")
```

```
grid.arrange(grafico_pred1_boot, grafico_pred2_boot)
```

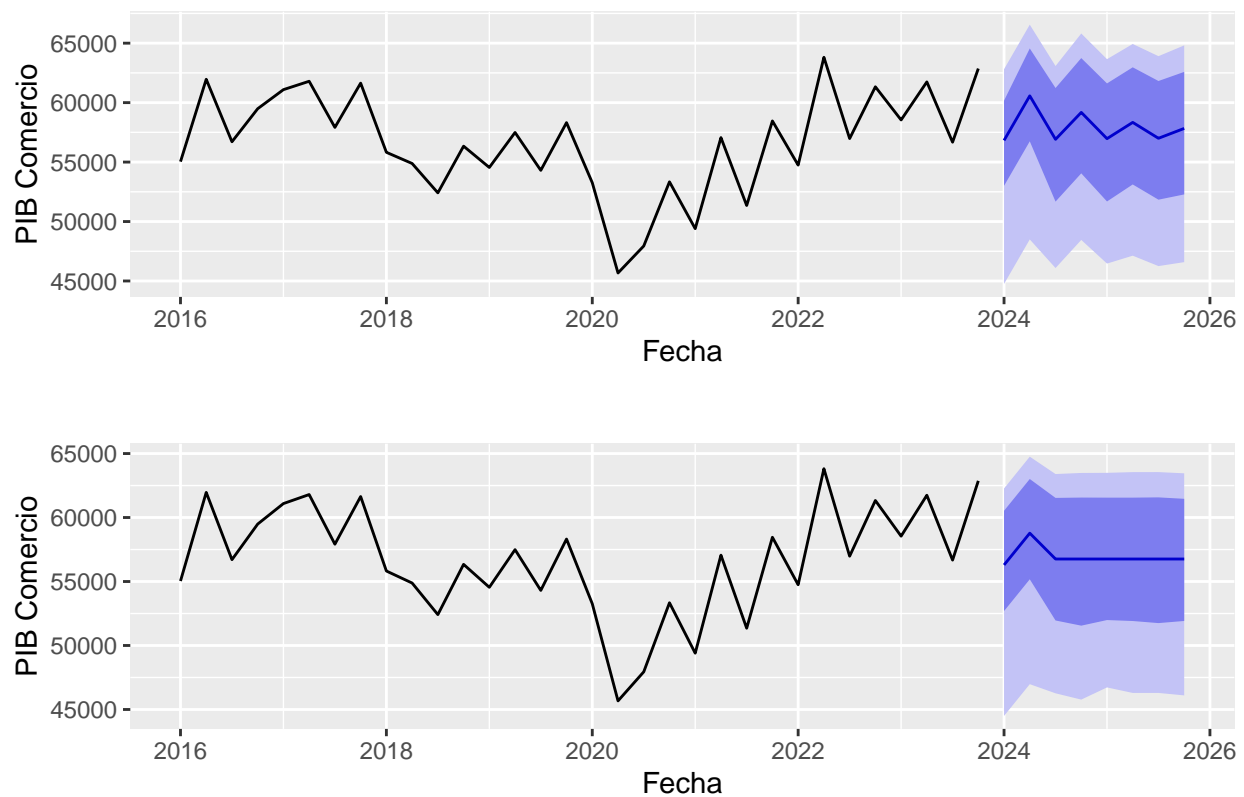


Figura 11: Predicciones a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para modelos AR(2) y MA(2). Se consideraron intervalos de confianza al 80 % y 95 % (la distribución de los residuos fue obtenida mediante técnicas de bootstrap).

```
# Al utilizar el bootstrap, no cambia la predicción puntual,  
# sino solamente los intervalos de confianza
```

```
# Modelo 1  
identical(pred1$mean, pred1_boot$mean)
```

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

```
identical(pred1$lower, pred1_boot$lower)
```

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

```
identical(pred1$upper, pred1_boot$upper)
```

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

```
# Modelo 2  
identical(pred2$mean, pred2_boot$mean)
```

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

```
identical(pred2$lower, pred1_boot$lower)
```

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

```
identical(pred2$upper, pred2_boot$upper)
```

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

```
grafico_pred1_comparacion <- autoplot(pred1, PI = FALSE) +  
  autolayer(pred1$lower, color = "black") +  
  autolayer(pred1$upper, color = "black") +  
  autolayer(pred1_boot$lower, color = "red") +  
  autolayer(pred1_boot$upper, color = "red") +  
  labs(x = "Fecha",  
        y = "PIB Comercio",  
        title = "",  
        color = "Intervalo")  
  
grafico_pred2_comparacion <- autoplot(pred2, PI = FALSE) +  
  autolayer(pred2$lower, color = "black") +  
  autolayer(pred2$upper, color = "black") +  
  autolayer(pred2_boot$lower, color = "red") +  
  autolayer(pred2_boot$upper, color = "red") +  
  labs(x = "Fecha",  
        y = "PIB Comercio",  
        title = "",  
        color = "Intervalo")  
  
grid.arrange(grafico_pred1_comparacion, grafico_pred2_comparacion)
```

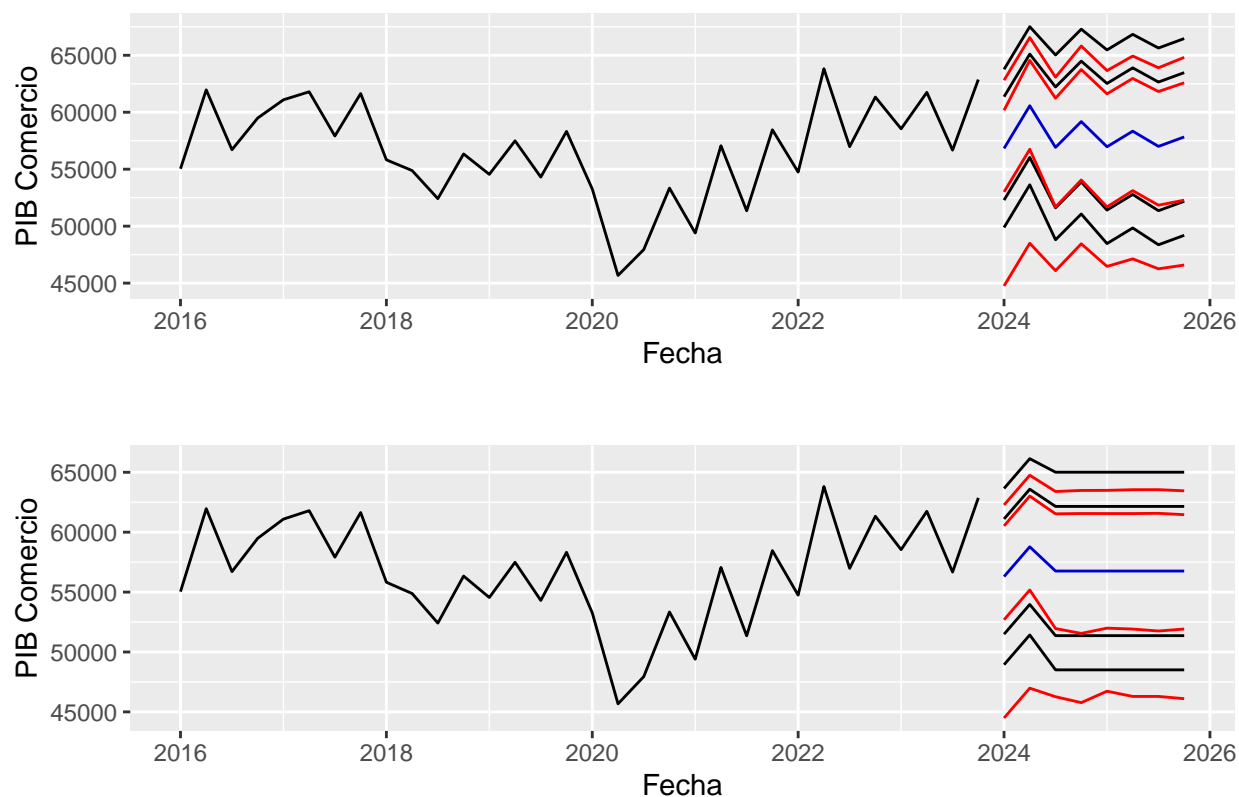


Figura 12: Predicciones a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para modelos AR(2) y MA(2). Se consideraron intervalos de confianza al 80 % y 95 % con (líneas rojas) y sin bootstrap (líneas negras).

### 3.2. Serie transformada

```
# Obtenemos las predicciones a 8 pasos para la serie transformada
# Si no ajustamos el sesgo por transformación y volvemos a la
# serie original aplicando la transformación inversa, obtenemos
# intervalos de confianza asimétricos y la predicción puntual resulta
# ser la mediana condicional. Al ajustar las predicciones, se lleva
# la predicción puntual a la media condicional.

# Modelo 1
pred1_trans_sin_aj <- forecast(modelo1_trans, biasadj = FALSE)
pred1_trans <- forecast(modelo1_trans, biasadj = TRUE)
# Ajustamos el sesgo por transformación

# Modelo 2
pred2_trans_sin_aj <- forecast(modelo2_trans, biasadj = FALSE)
pred2_trans <- forecast(modelo2_trans, biasadj = TRUE)
# Ajustamos el sesgo por transformación
```

```
# Graficamos las predicciones obtenidas

# Modelo 1
grafico_pred1_trans <- autoplot(pred1_trans) +
  autolayer(pred1_trans_sin_aj$mean, color = "red") +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")

# Modelo 2
grafico_pred2_trans <- autoplot(pred2_trans) +
  autolayer(pred2_trans_sin_aj$mean, color = "red") +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")

grid.arrange(grafico_pred1_trans, grafico_pred2_trans)
```

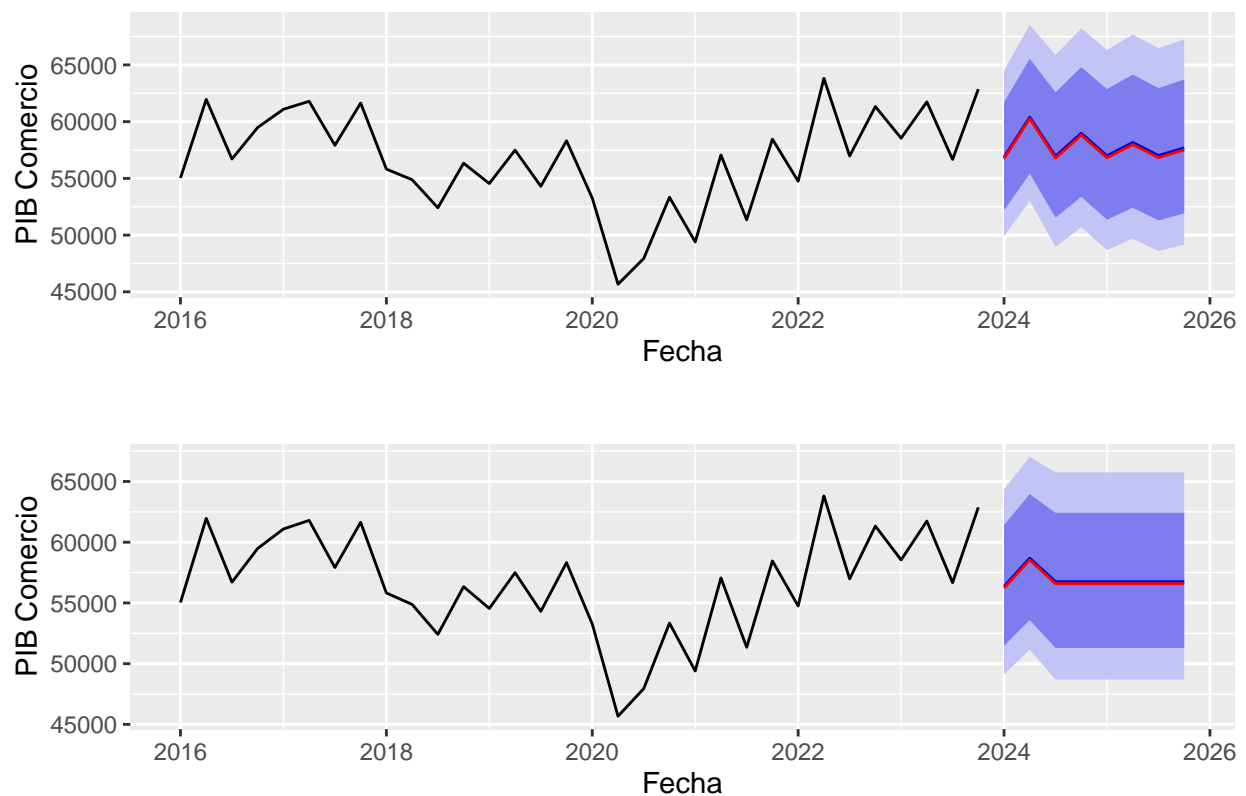


Figura 13: Predicciones a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para los modelos AR(2) y MA(2). Se consideraron intervalos de confianza al 80 % y 95 %. La línea azul corresponde a las predicciones ajustadas, y la roja a las sin ajustar.

```
# Comparamos las predicciones puntuales con y sin ajuste
```



```

# Modelo 1
comparacion_ajuste1 <- data.frame(Fecha = seq(from = as.Date("2024-01-01"),
                                             to = as.Date("2025-12-31"),
                                             by = "quarter"),
                                sin_ajustar = pred1_trans_sin_aj$mean,
                                ajustada = pred1_trans$mean) %>%
  pivot_longer(cols = c(sin_ajustar, ajustada),
               names_to = "Serie",
               values_to = "Valores") %>%
  mutate(Serie = recode(Serie,
                        "sin_ajustar" = "Sin ajustar",
                        "ajustada" = "Ajustada"))

grafico_comparacion_ajuste1 <- ggplot(comparacion_ajuste1) +
  geom_line(aes(x = Fecha, y = Valores, color = Serie)) +
  labs(color = "Predicción")

# Modelo 2
comparacion_ajuste2 <- data.frame(Fecha = seq(from = as.Date("2024-01-01"),
                                             to = as.Date("2025-12-31"),
                                             by = "quarter"),
                                sin_ajustar = pred2_trans_sin_aj$mean,
                                ajustada = pred2_trans$mean) %>%
  pivot_longer(cols = c(sin_ajustar, ajustada),
               names_to = "Serie",
               values_to = "Valores") %>%
  mutate(Serie = recode(Serie,
                        "sin_ajustar" = "Sin ajustar",
                        "ajustada" = "Ajustada"))

grafico_comparacion_ajuste2 <- ggplot(comparacion_ajuste2) +
  geom_line(aes(x = Fecha, y = Valores, color = Serie)) +
  labs(color = "Predicción")

grid.arrange(grafico_comparacion_ajuste1, grafico_comparacion_ajuste2)

```

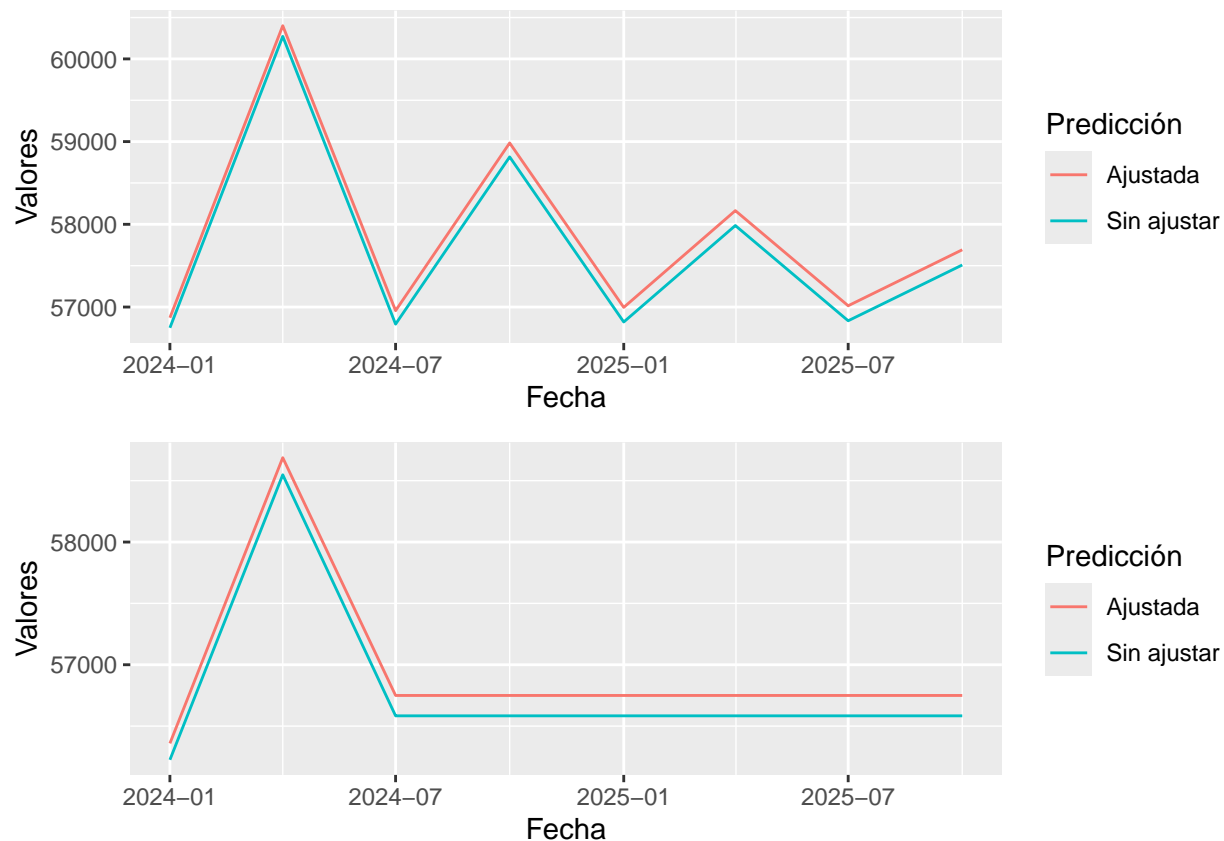


Figura 14: Predicciones puntuales a 8 pasos del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para los modelos AR(2) y MA(2) con y sin ajuste del sesgo por transformación.

## 4. Validación de las predicciones

Dado que el modelo siempre tiene un mejor ajuste dentro de la muestra que se usó para estimarlo que para nuevos datos, es importante evaluar la calidad de las predicciones en un nuevo conjunto de observaciones.

La manera más simple de validar los pronósticos obtenidos es dividiendo la muestra en dos partes:

- **Datos de entrenamiento:** También llamado “*training set*”, se compone de la primera parte de la muestra y se utiliza para estimar el modelo identificado.
- **Datos de prueba:** Se lo conoce como “*test set*” y se lo utiliza para evaluar las predicciones obtenidas mediante el *training set* a partir de distintas medidas de error.

### 4.1. 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)

# Modelo 1
accuracy(modelo1)
```

```
##               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
```

```
# Modelo 2
accuracy(modelo2)
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -41.59821 3634.153 2685.229 -0.5520701 4.905562 0.6434239
##               ACF1
## Training set 0.2436876
```

## 4.2. 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)
```

```
# Estimamos los modelos para el training set (series sin transformar)
```

```
# Modelo 1
modelo1_train <- Arima(y = train_comercio,
  order = c(2, 0, 0),
  lambda = NULL,
  fixed = c(0, NA, NA))
```

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

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

```
# Modelo 1
pred1_test <- forecast(modelo1_train, h = n)
```

```
# Modelo 2
pred2_test <- forecast(modelo2_train, h = n)
```

```
# Graficamos las predicciones obtenidas
```

```
# Modelo 1
grafico_pred1_test <- autoplot(pred1_test) +
  autolayer(comercio, color = "black") +
```

```

labs(x = "Fecha",
     y = "PIB Comercio",
     title = "")

# Modelo 2
grafico_pred2_test <- autoplot(pred2_test) +
  autolayer(comercio, color = "black") +
  labs(x = "Fecha",
       y = "PIB Comercio",
       title = "")

grid.arrange(grafico_pred1_test, grafico_pred2_test)

```

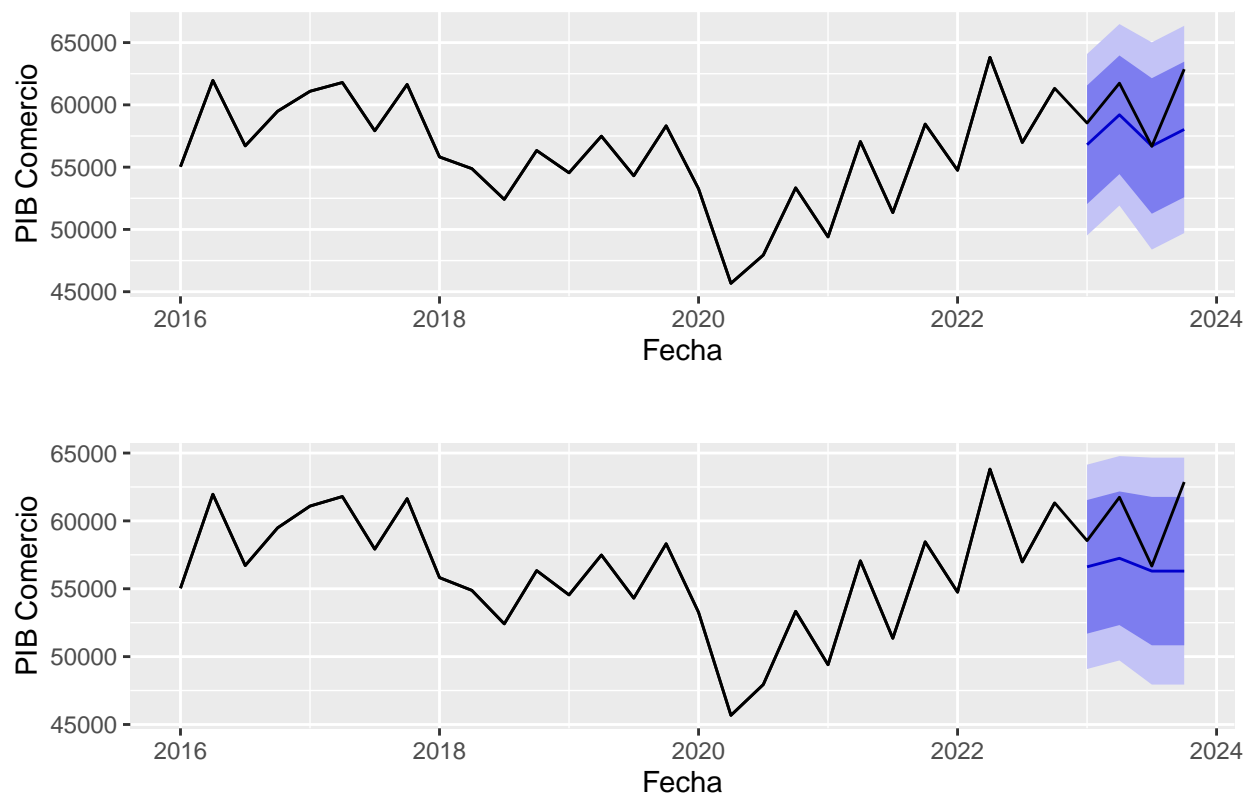


Figura 15: Predicciones en el conjunto de prueba del PIB del sector Comercio, alojamiento y suministro de comidas y bebidas (millones de pesos a precios constantes de 2016) para modelos AR(2) y MA(2). La línea azul corresponde a las predicciones.

```

# 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)

# Modelo 1
accuracy(pred1_test, test_comercio)

```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -98.67626 3580.095 2600.855 -0.6200449 4.804871 0.5719184
## Test set     2273.17083 2867.727 2287.110  3.6842910 3.708887 0.5029271
##               ACF1 Theil's U
## Training set  0.3001828      NA
## Test set      -0.5442531 0.6506197
```

*# Modelo 2*

```
accuracy(pred2_test, test_comercio)
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -56.43657 3700.954 2697.614 -0.5944737 4.969193 0.5931954
## Test set     3339.86519 4097.298 3339.865  5.4180524 5.418052 0.7344240
##               ACF1 Theil's U
## Training set  0.2622495      NA
## Test set      -0.6490699 0.945089
```