# CALIDAD DE AIRE

María Paula Camargo Rincón  Laura Katherin Martinez Castiblanco
Yudy Vanessa Puerres Rosero

2025-10-20

```
rm(list=ls())
```

## librerías

```
#install.packages("readxl")
#install.packages("knitr")
#install.packages("forecast")
#install.packages("FinTS")
#install.packages("dplyr")
#install.packages("ggplot2")
#install.packages(lubridate")
#install.packages("zoo")
#install.packages("nortest")
library(readxl) # Para leer el csv
library(knitr) # Para obtener la tabla en el pdf a partir de una tabla en r
library(forecast) # para el lambda de box-cox
```

```
## Warning: package 'forecast' was built under R version 4.5.1
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries) # prueba de estacionariedad adf
library(FinTS)
```

```
## Warning: package 'FinTS' was built under R version 4.5.1
```

```
## Cargando paquete requerido: zoo
```

```
##
## Adjuntando el paquete: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Adjuntando el paquete: 'FinTS'
```

```
## The following object is masked from 'package:forecast':
##
##     Acf
```

```r
library(dplyr) # para manejo de datos
```

```
##
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2) # para gráficas
library(lubridate) # fechas
```

```
##
## Adjuntando el paquete: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(zoo) #imputación de datos faltantes
library(nortest) # para prueba kolmogorov
```

# Descripción de los datos:

## Base de datos

La base de datos utilizada proviene del conjunto de datos público disponible en la plataforma Kaggle (Rohan Rao, 2020). Este conjunto fue recopilado originalmente por el Central Pollution Control Board (CPCB), organismo oficial del Gobierno de la India encargado del monitoreo ambiental.

Para este estudio, se utilizó la versión diaria del conjunto de datos, enfocándose exclusivamente en la variable PM2.5 (µg/m³) (partículas en suspensión con diámetro aerodinámico $\leq$ 2.5 µm) registrada en la ciudad de Delhi. La serie abarca el período comprendido entre el 1 de enero de 2015 y el 6 de diciembre de 2020, con observaciones diarias.

```r
aire <- read_excel("Datos_India.xlsx")
knitr::kable(head(aire,10),
             caption = "Primeros 10 datos",
             digits = 4)  # redondea a 4 decimales
```

Table 1: Primeros 10 datos

| City | Date | PM2.5 | AQI_Bucket |
|---|---|---|---|
| Delhi | 2015-01-01 | 313.22 | Severe |
| Delhi | 2015-02-01 | 186.18 | Severe |
| Delhi | 2015-03-01 | 87.18 | Moderate |
| Delhi | 2015-04-01 | 151.84 | Very Poor |
| Delhi | 2015-05-01 | 146.60 | Very Poor |
| Delhi | 2015-06-01 | 149.58 | Very Poor |
| Delhi | 2015-07-01 | 217.87 | Very Poor |
| Delhi | 2015-08-01 | 229.90 | Very Poor |
| Delhi | 2015-09-01 | 201.66 | Very Poor |

| City | Date | PM2.5 | AQI_Bucket |
|---|---|---|---|
| Delhi | 2015-10-01 | 221.02 | Very Poor |

Además de PM2.5, el conjunto incluye el Índice de Calidad del Aire (AQI) y su clasificación categórica (*AQI_Bucket*), que divide la calidad del aire en seis niveles: *Good*, *Satisfactory*, *Moderate*, *Poor*, *Very Poor* y *Severe*. Sin embargo, dado que el objetivo del análisis es modelar la dinámica temporal de la concentración de PM2.5, se trabajó únicamente con la variable numérica continua.

```
# Convert 'City' and 'AQI_Bucket' to factor variables
aire$City <- as.factor(aire$City)
aire$AQI_Bucket <- as.factor(aire$AQI_Bucket)

# Convert 'Date' to a date variable, specifying the format
aire$Date <- as.Date(aire$Date, format = "%d/%m/%Y")

summary(aire)
```

```
##     City           Date                PM2.5               AQI_Bucket
##  Delhi:2009   Min.   :2015-01-01   Min.   : 10.24   Good        : 21
##               1st Qu.:2016-05-17   1st Qu.: 57.09   Moderate    :519
##               Median :2017-10-01   Median : 94.62   Poor        :542
##               Mean   :2017-10-04   Mean   :117.20   Satisfactory:158
##               3rd Qu.:2019-02-15   3rd Qu.:153.03   Severe      :239
##               Max.   :2020-12-06   Max.   :685.36   Very Poor   :520
##                                    NA's   :2        NA's        : 10
```

```
str(aire)
```

```
## tibble [2,009 x 4] (S3: tbl_df/tbl/data.frame)
##  $ City      : Factor w/ 1 level "Delhi": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date      : Date[1:2009], format: "2015-01-01" "2015-02-01" ...
##  $ PM2.5     : num [1:2009] 313.2 186.2 87.2 151.8 146.6 ...
##  $ AQI_Bucket: Factor w/ 6 levels "Good","Moderate",..: 5 5 2 6 6 6 6 6 6 6 ...
```

## Evaluación de base de datos

```
# Group by Date and count the number of entries
daily_counts <- aire %>%
  group_by(Date) %>%
  summarise(count = n())

# Filter for dates with more than one entry
dates_with_duplicates <- daily_counts %>%
  filter(count > 1)


# Display the dates with duplicate entries
if (nrow(dates_with_duplicates) > 0) {
  cat("Días con mas de un valor de  PM2.5:\n")
  print(dates_with_duplicates)
} else {
  cat("No hay días con mas de un valor de PM2.5\n")
}
```

```
## No hay días con mas de un valor de PM2.5
```

El conjunto inicial contiene 2,009 observaciones. Se identificaron 2 valores faltantes en la variable `PM2.5` y 10 en `AQI_Bucket`. Dado que el AQI no se utilizará en el modelado, se centró la atención en imputar los valores ausentes de PM2.5. Se aplicó interpolación lineal mediante la función `na.approx()` del paquete `zoo`, asumiendo que la evolución de la contaminación en el corto plazo no es drástica.

```
aire[!complete.cases(aire), ]
```

```
## # A tibble: 11 x 4
##    City  Date       PM2.5 AQI_Bucket
##    <fct> <date>     <dbl> <fct>
##  1 Delhi 2016-07-24  59.4 <NA>
##  2 Delhi 2017-06-23  44.1 <NA>
##  3 Delhi 2017-12-08  NA   Good
##  4 Delhi 2017-08-13  NA   <NA>
##  5 Delhi 2017-08-14  26.5 <NA>
##  6 Delhi 2017-08-22  46   <NA>
##  7 Delhi 2017-08-23  36.5 <NA>
##  8 Delhi 2017-08-26  62.3 <NA>
##  9 Delhi 2017-08-27  34.3 <NA>
## 10 Delhi 2017-08-28  23.8 <NA>
## 11 Delhi 2017-08-29  15.2 <NA>
```

```
# Usar interpolación lineal para rellenar valores faltantes en 'PM2.5'
# Podemos usar la función na.approx del paquete zoo para la interpolación lineal.
aire$`PM2.5` <- na.approx(aire$`PM2.5`)

# Verificar que no haya más valores faltantes en 'PM2.5'
missing_pm25_after_imputation <- sum(is.na(aire$`PM2.5`))
cat("Número de datos faltantes en 'PM2.5' después de la imputación:", missing_pm25_after_imputation, "\n
```

```
## Número de datos faltantes en 'PM2.5' después de la imputación: 0
```

Las estadísticas resumen de la serie de PM2.5 revelan una distribución altamente asimétrica hacia la derecha, con una media de 117.10 µg/m³, una mediana de 94.49 µg/m³ y un máximo extremo de 685.36 µg/m³.

```
# Estadísticas descriptivas
summary_stats <- aire %>%
  summarise(
    Media = mean(PM2.5, na.rm = TRUE),
    Mediana = median(PM2.5, na.rm = TRUE),
    Desv_Est = sd(PM2.5, na.rm = TRUE),
    Min = min(PM2.5, na.rm = TRUE),
    Max = max(PM2.5, na.rm = TRUE)
  )

kable(summary_stats, digits = 3, caption = "Estadísticas descriptivas de PM2.5 ")
```

Table 2: Estadísticas descriptivas de PM2.5

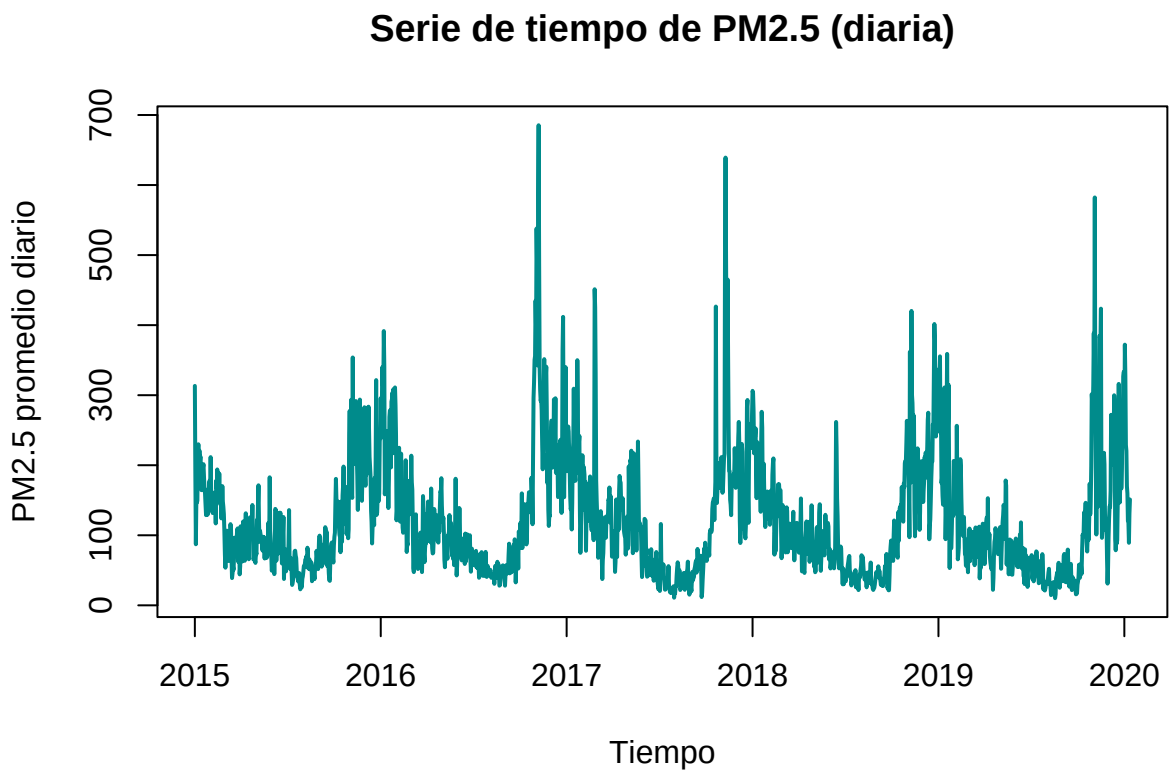
| Media | Mediana | Desv_Est | Min | Max |
|---|---|---|---|---|
| 117.104 | 94.49 | 82.924 | 10.24 | 685.36 |

La gráfica de la serie de tiempo muestra una variabilidad, con estacionalidad clara (patrones repetidos, que parecieran ser anuales). Aunque no se observa una tendencia, la varianza no es constante puesto que los picos

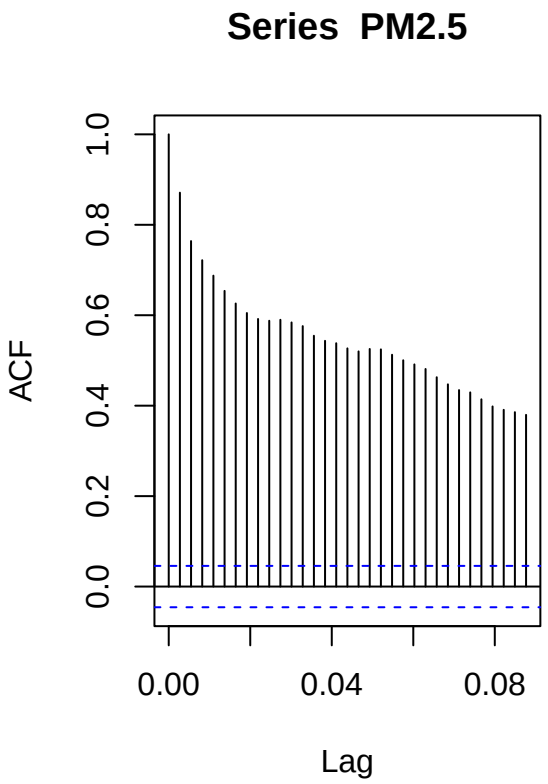
son más pronunciados en ciertos períodos del año, lo que sugiere heterocedasticidad.

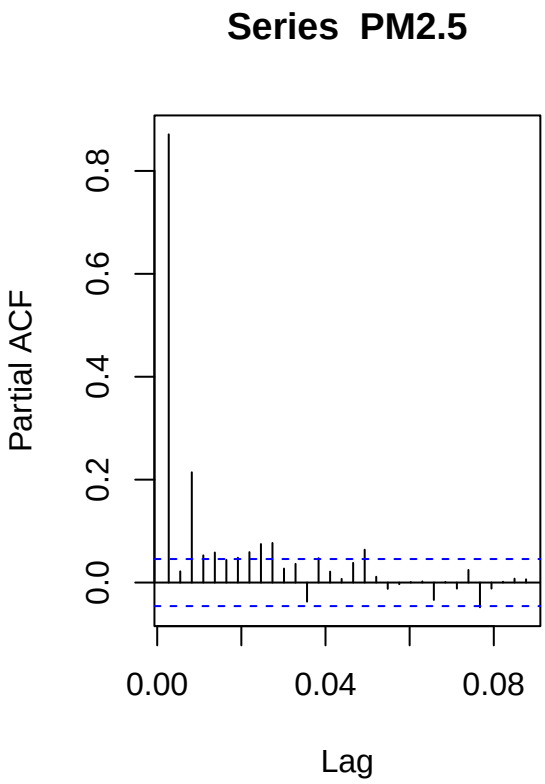
```
# Creación de la serie
start_date <- min(aire$Date)
end_date <- max(aire$Date)
PM2.5 <- ts(aire$`PM2.5`, start = c(year(start_date), month(start_date), day(start_date)), end = c(year

# Gráfica
ts.plot(PM2.5, col = "darkcyan", lwd = 2,
        ylab = "PM2.5 promedio diario", xlab = "Tiempo",
        main = "Serie de tiempo de PM2.5 (diaria)")
```




```
par(mfrow=c(1,2))
acf(PM2.5) # decaimiento lento indica que la serie no es estacionaria
pacf(PM2.5)
```




Para evaluar formalmente la estacionariedad en media (la cual segun gráficos parece ser estacionaria), se usa la prueba de Dickey-Fuller (ADF). El estadístico resultante fue -4.7414 con un p-valor $< 0.01$, lo que permite rechazar la hipótesis nula de raíz unitaria. Por lo tanto, la serie se considera estacionaria.

```
adf_result <- adf.test(PM2.5, alternative = "stationary")
```

```
## Warning in adf.test(PM2.5, alternative = "stationary"): p-value smaller than
## printed p-value
```

```
cat("Prueba ADF:\n Estadístico =", round(adf_result$statistic, 4),
    ", p-valor =", adf_result$p.value, "\n")
```

```
## Prueba ADF:
##  Estadístico = -4.7414 , p-valor = 0.01
```

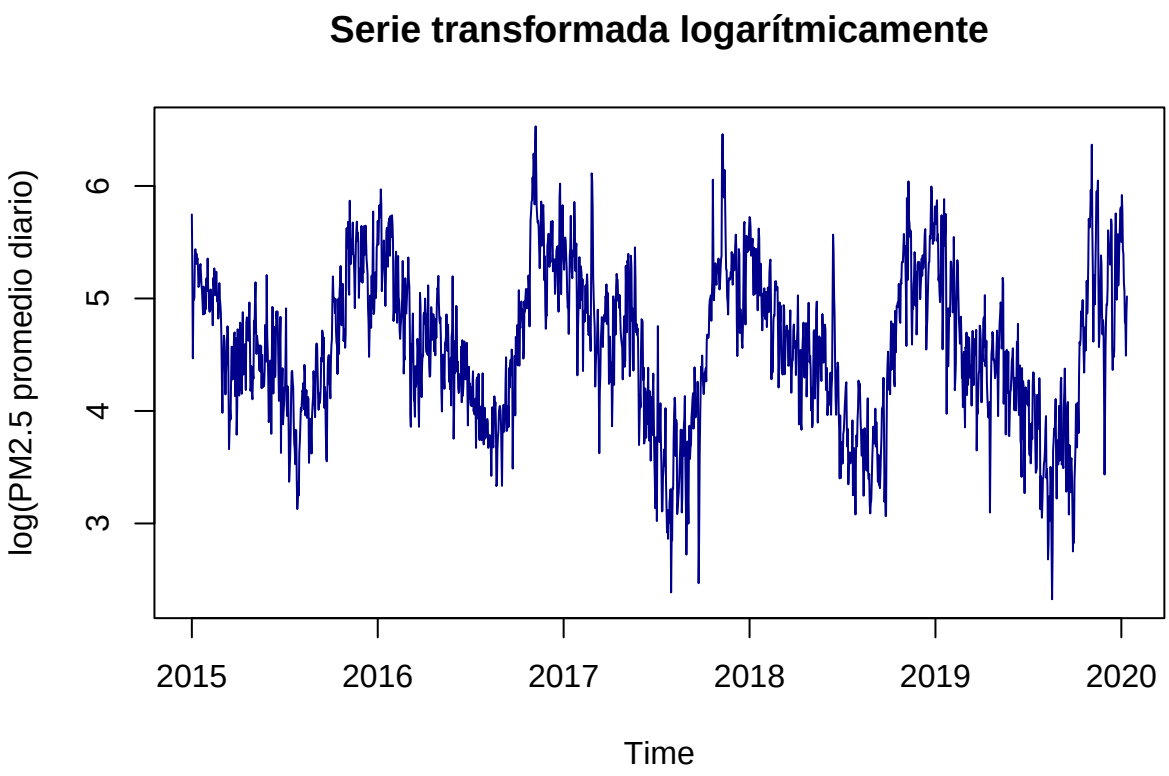
Dada la heterocedasticidad, se evaluo una transformación para estabilizar la varianza. Se estimó el parámetro de Box-Cox, obteniéndo $\lambda \approx 0.257$, cercano a cero. Por lo cual se aplicó la transformación logarítmica.

```
lambda <- BoxCox.lambda(PM2.5)
lambda
```

```
## [1] 0.2571413
```

Como el lambda obtenido fue cercano a 0, se aplicará la transformación de logaritmica.

```
# Transformación logarítmica
l.PM2.5 <- log(PM2.5)
ts.plot(l.PM2.5, col = "darkblue",
        ylab = "log(PM2.5 promedio diario)",
        main = "Serie transformada logarítmicamente")
```



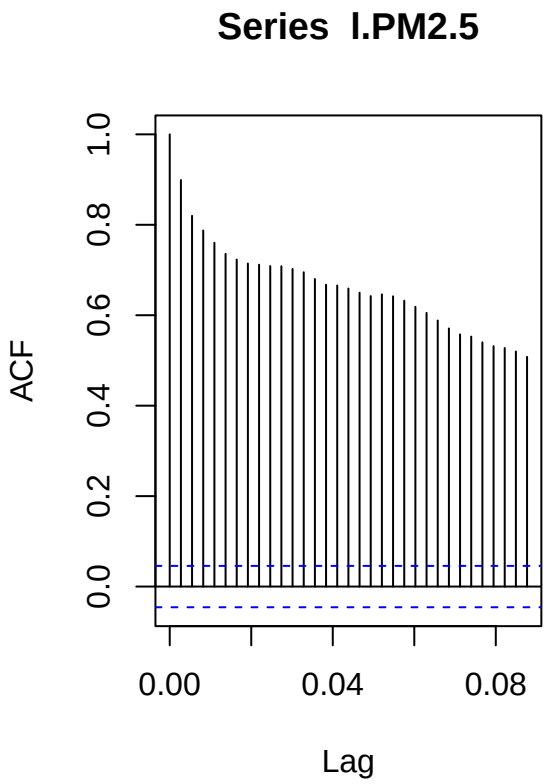

La serie transformada presenta una varianza más homogénea y mantiene la estacionariedad en media (prueba ADF con p-valor $\approx 0.027$), cumpliendo así los supuestos de estacionariedad para el ajuste de modelos ARIMA.

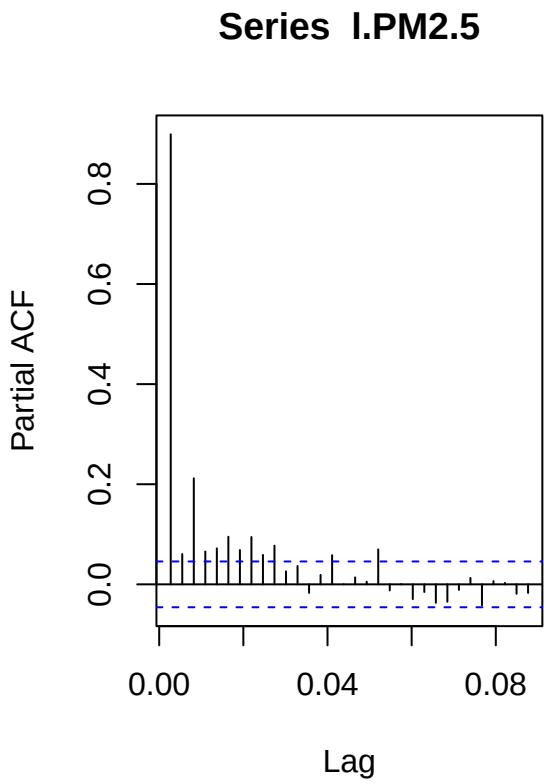
```
adf.test(l.PM2.5,alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  l.PM2.5
## Dickey-Fuller = -3.6549, Lag order = 12, p-value = 0.02739
## alternative hypothesis: stationary
```

## Modelos

```
par(mfrow=c(1,2))
acf(l.PM2.5)
pacf(l.PM2.5)
```




Pareciera que un ARMA(3,2) es adecuado, tal vez AR(9).

```
ar9 <- arima(x = l.PM2.5,order = c(9,0,0))
ar9
```

```
##
## Call:
## arima(x = l.PM2.5, order = c(9, 0, 0))
##
## Coefficients:
##          ar1      ar2     ar3     ar4      ar5     ar6     ar7     ar8     ar9
##       0.7895  -0.1245  0.1391  0.0114  -0.0150  0.0431  0.0007  0.0491  0.0606
## s.e.  0.0233   0.0297  0.0299  0.0301   0.0301  0.0301  0.0299  0.0298  0.0234
##       intercept
##          4.5994
## s.e.     0.1420
##
## sigma^2 estimated as 0.08234:  log likelihood = -314.24,  aic = 650.49
```

```
arma3.2 <- arima(x = l.PM2.5,order = c(3,0,2))
arma3.2
```

```
##
## Call:
## arima(x = l.PM2.5, order = c(3, 0, 2))
##
## Coefficients:
##          ar1      ar2      ar3      ma1      ma2  intercept
```

```
##       1.2050  -0.0898  -0.1225  -0.4223  -0.3684     4.6052
## s.e.  0.1165   0.1898   0.0793   0.1121   0.1004     0.1817
##
## sigma^2 estimated as 0.0821:  log likelihood = -311.59,  aic = 637.18
```

```
parametros <- c("Intercepto",
                paste0("ar", 1:9),
                paste0("ma", 1:2),
                "logLik", "AIC")
resumen <- data.frame(
  "Parámetro" = parametros,

  # Modelo AR(9)
  "AR(9)" = c(coef(ar9)[10],coef(ar9)[-10], rep(NA, 2), ar9$loglik, ar9$aic),

  # Valor t (significancia) para AR(9)
  "t-valor AR(9)" = c((ar9$coef / sqrt(diag(ar9$var.coef)))[10],
                      (ar9$coef / sqrt(diag(ar9$var.coef)))[-10],
                      rep(NA, 2), NA, NA),

  # Modelo MA(9)
  "ARMA(3,2)" = c(coef(arma3.2)[6],coef(arma3.2)[1:3],
                  rep(NA,6), coef(arma3.2)[4:5],
                  arma3.2$loglik, arma3.2$aic),

  # Valor t (significancia) para MA(9)
  "t-valor MA(9)" = c((arma3.2$coef / sqrt(diag(arma3.2$var.coef)))[6],
                      (arma3.2$coef / sqrt(diag(arma3.2$var.coef)))[1:3],
                      rep(NA,6),
                      (arma3.2$coef / sqrt(diag(arma3.2$var.coef)))[4:5],
                      NA, NA)
)
knitr::kable(resumen,
             caption = "Tabla de resumen de los modelos",
             digits = 3,
             row.names = F)
```

Table 3: Tabla de resumen de los modelos

| Parámetro | AR.9. | t.valor.AR.9. | ARMA.3.2. | t.valor.MA.9. |
|---|---|---|---|---|
| Intercepto | 4.599 | 32.387 | 4.605 | 25.338 |
| ar1 | 0.789 | 33.894 | 1.205 | 10.346 |
| ar2 | -0.125 | -4.195 | -0.090 | -0.473 |
| ar3 | 0.139 | 4.655 | -0.122 | -1.544 |
| ar4 | 0.011 | 0.378 | NA | NA |
| ar5 | -0.015 | -0.499 | NA | NA |
| ar6 | 0.043 | 1.434 | NA | NA |
| ar7 | 0.001 | 0.022 | NA | NA |
| ar8 | 0.049 | 1.651 | NA | NA |
| ar9 | 0.061 | 2.595 | NA | NA |
| ma1 | NA | NA | -0.422 | -3.766 |
| ma2 | NA | NA | -0.368 | -3.671 |
| logLik | -314.243 | NA | -311.590 | NA |
| AIC | 650.487 | NA | 637.181 | NA |

**Formula AR(9)**

$$Z_t = 0.789Z_{t-1} - 0.125Z_{t-2} + 0.139Z_{t-3} + 0.061Z_{t-9} + a_t + 4.599$$

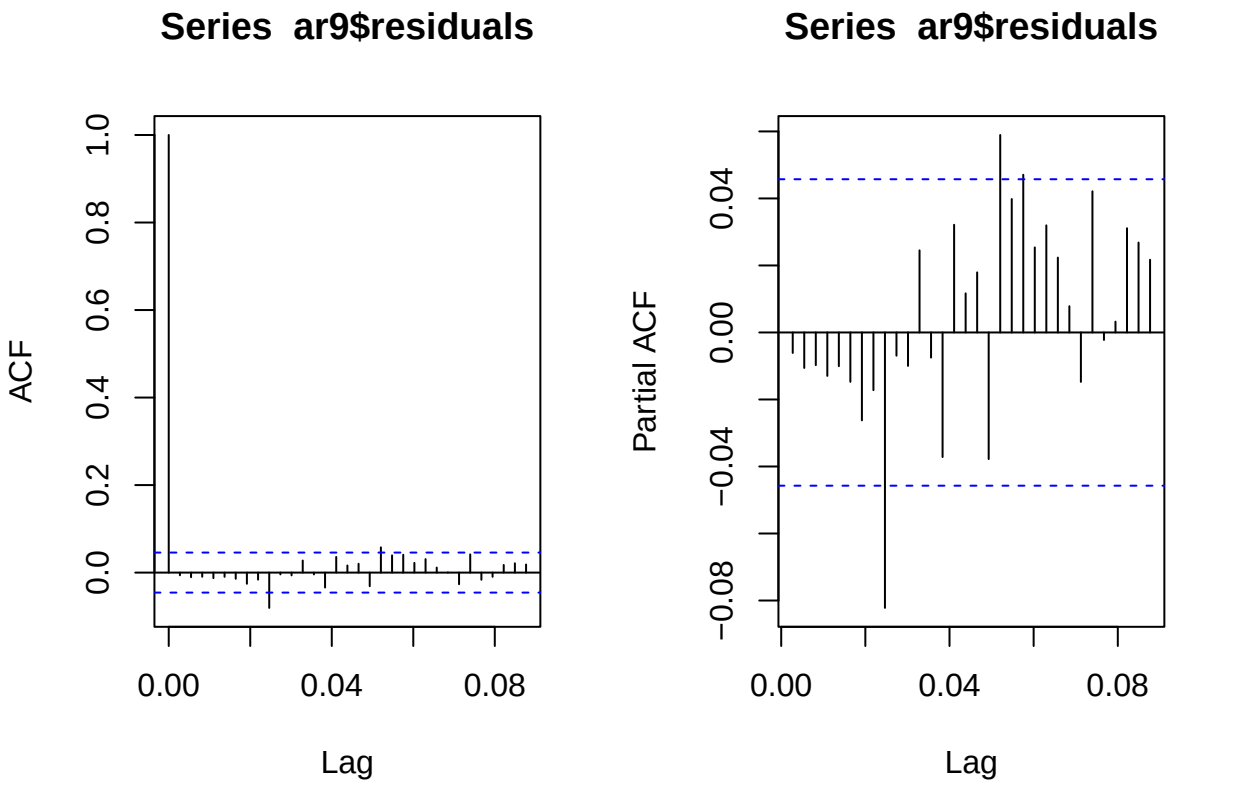
**Formula AR(9)**

$$Z_t = 1.205Z_{t-1} + a_t - 0.422a_{t-1} - 0.368 + 4.605$$

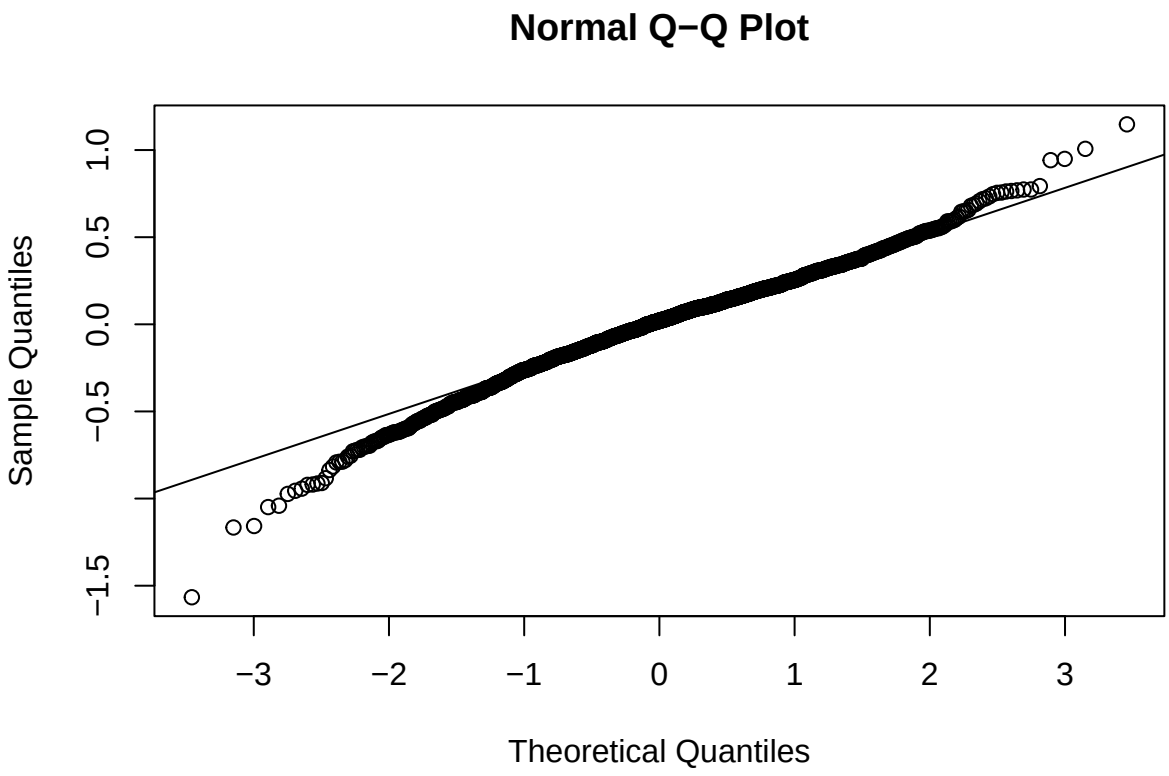
# Validación modelos

## AR(9)

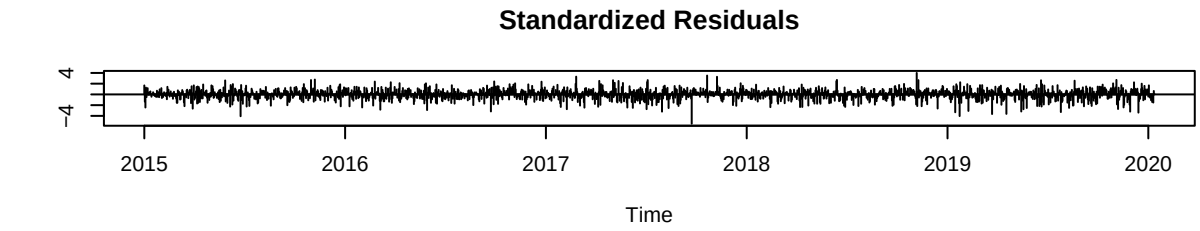
```
par(mfrow=c(1,2))
acf(ar9$residuals)
pacf(ar9$residuals)
```

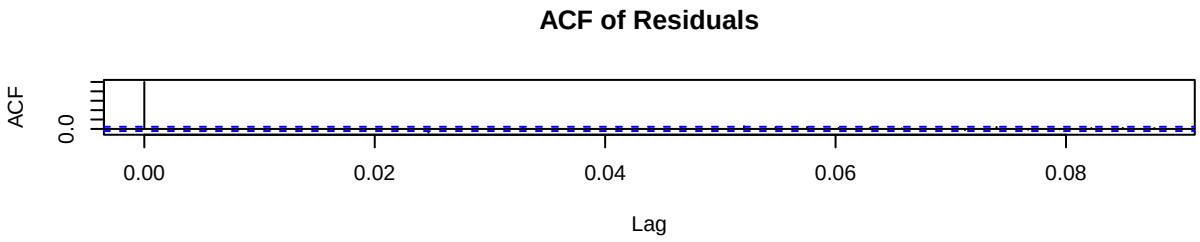


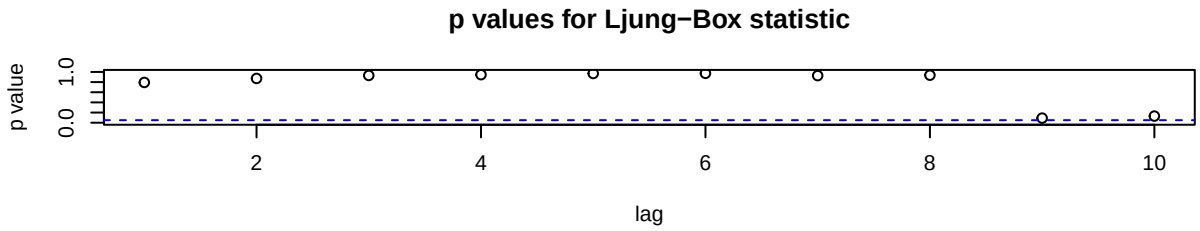

```
qqnorm(ar9$residuals)
qqline(ar9$residuals)
```


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−1.5
−0.5
0.0
0.5
1.0
−3
−2
−1
0
1
2
3

```
tsdiag(ar9)
```




```
ks.test(ar9$residuals, "pnorm", mean = 0, sd = sd(ar9$residuals))
```

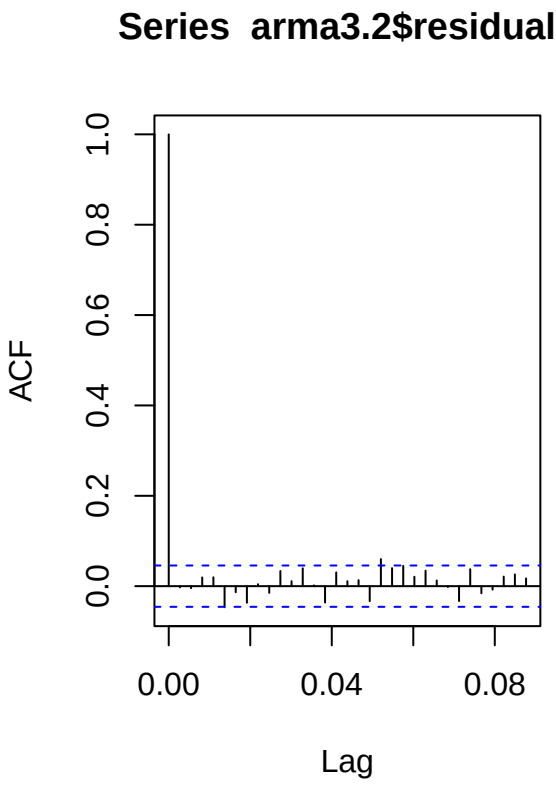
```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  ar9$residuals
## D = 0.036861, p-value = 0.01359
## alternative hypothesis: two-sided
```

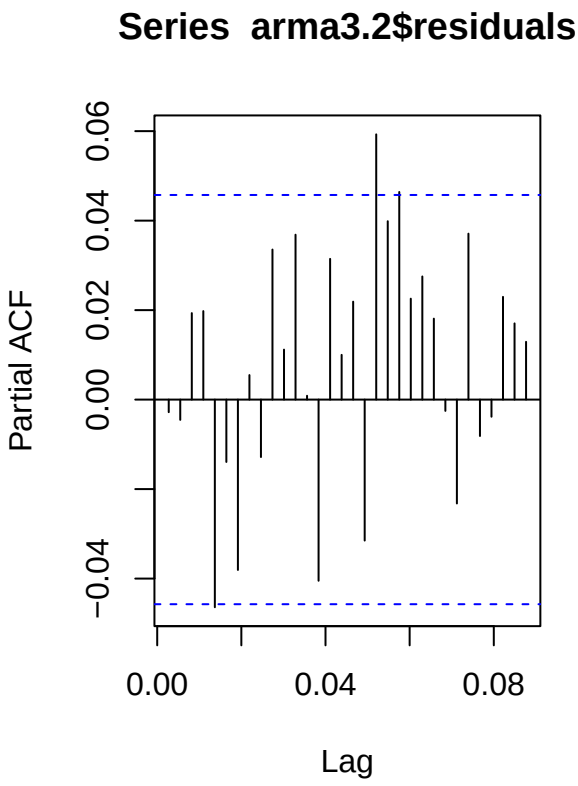
```
shapiro.test(ar9$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ar9$residuals
## W = 0.98596, p-value = 2.032e-12
```

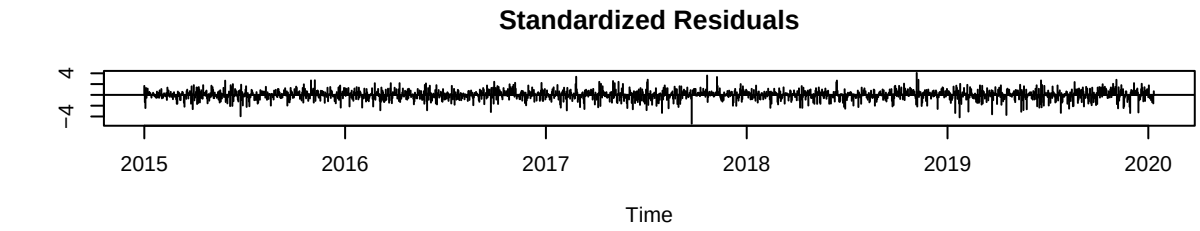
## ARMA(3,2)

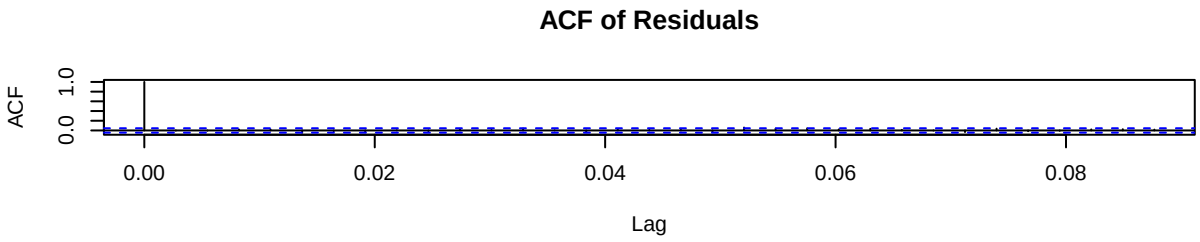
```
par(mfrow=c(1,2))
acf(arma3.2$residuals)
pacf(arma3.2$residuals)
```

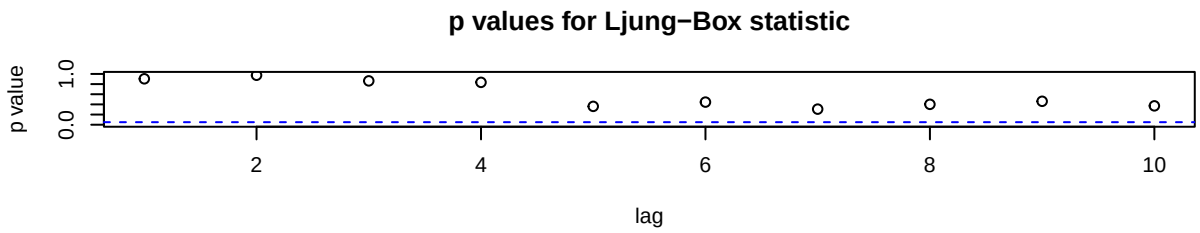


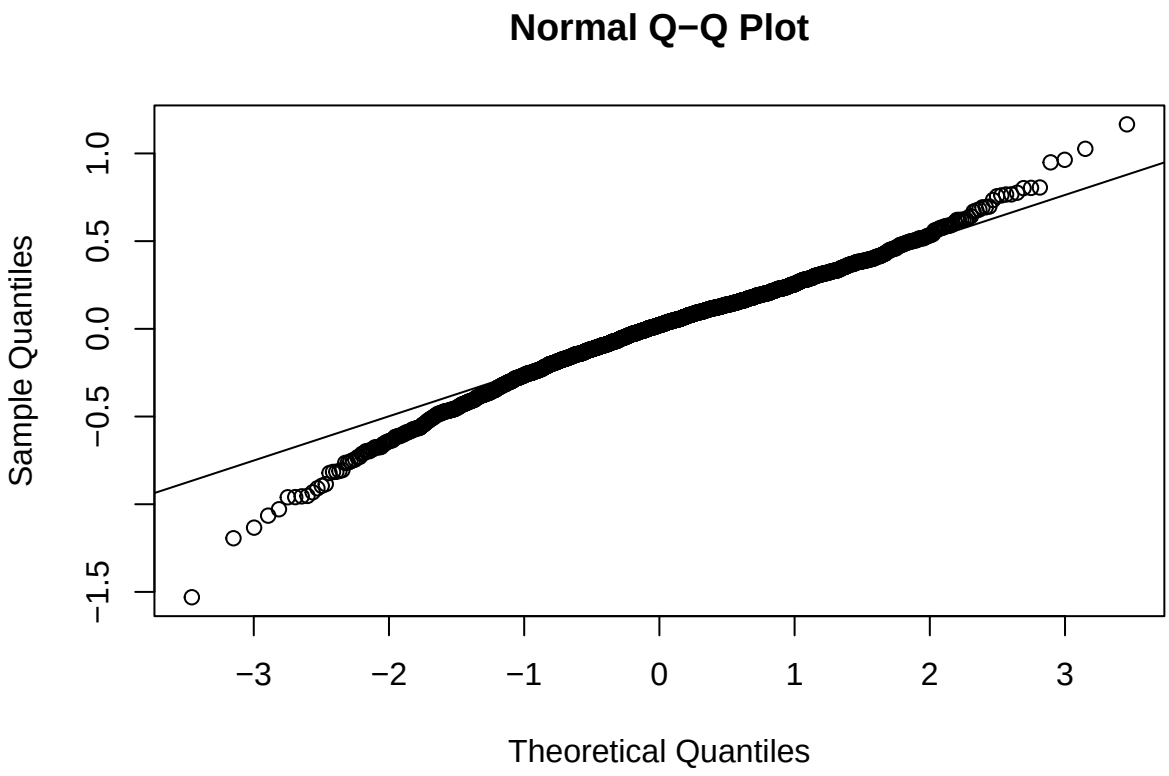

```
tsdiag(arma3.2)
```




```
qqnorm(arma3.2$residuals)
qqline(arma3.2$residuals)
```




Parace que hay problemas con las colas.

```
ks.test(ar9$residuals, "pnorm", mean = 0, sd = sd(arma3.2$residuals))
```

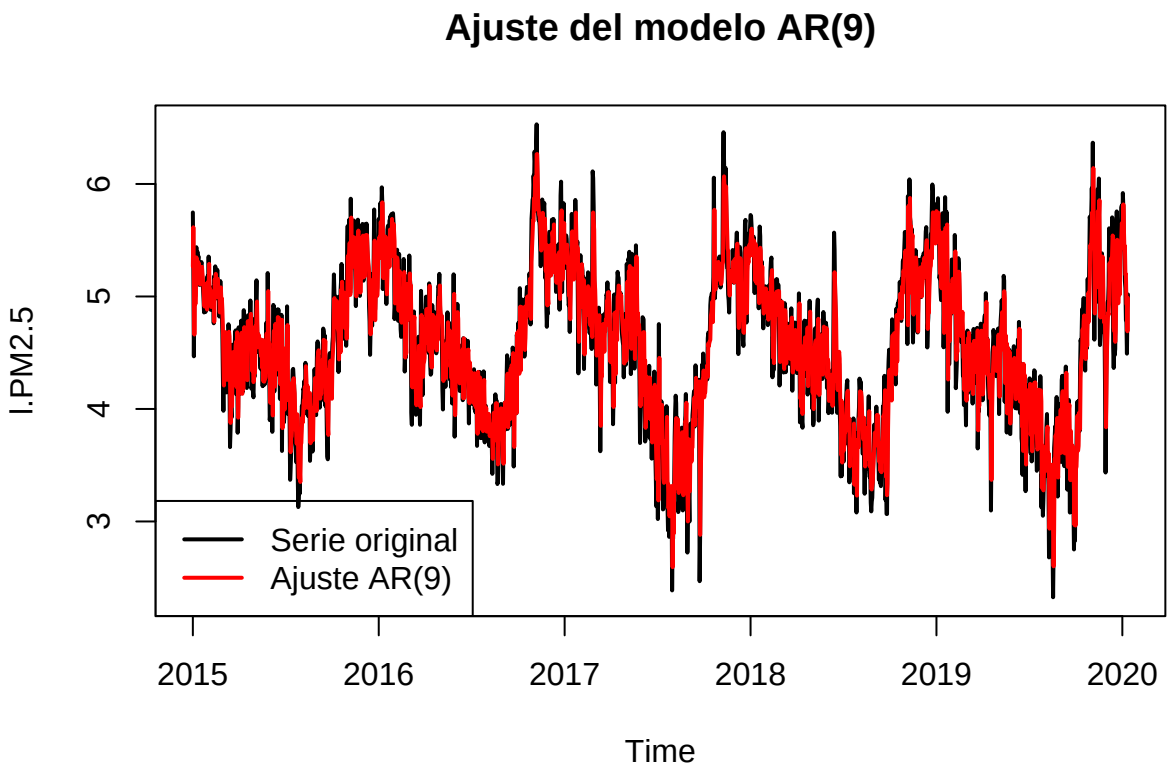
```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  ar9$residuals
## D = 0.036723, p-value = 0.0141
## alternative hypothesis: two-sided
```

```
shapiro.test(arma3.2$residuals)
```

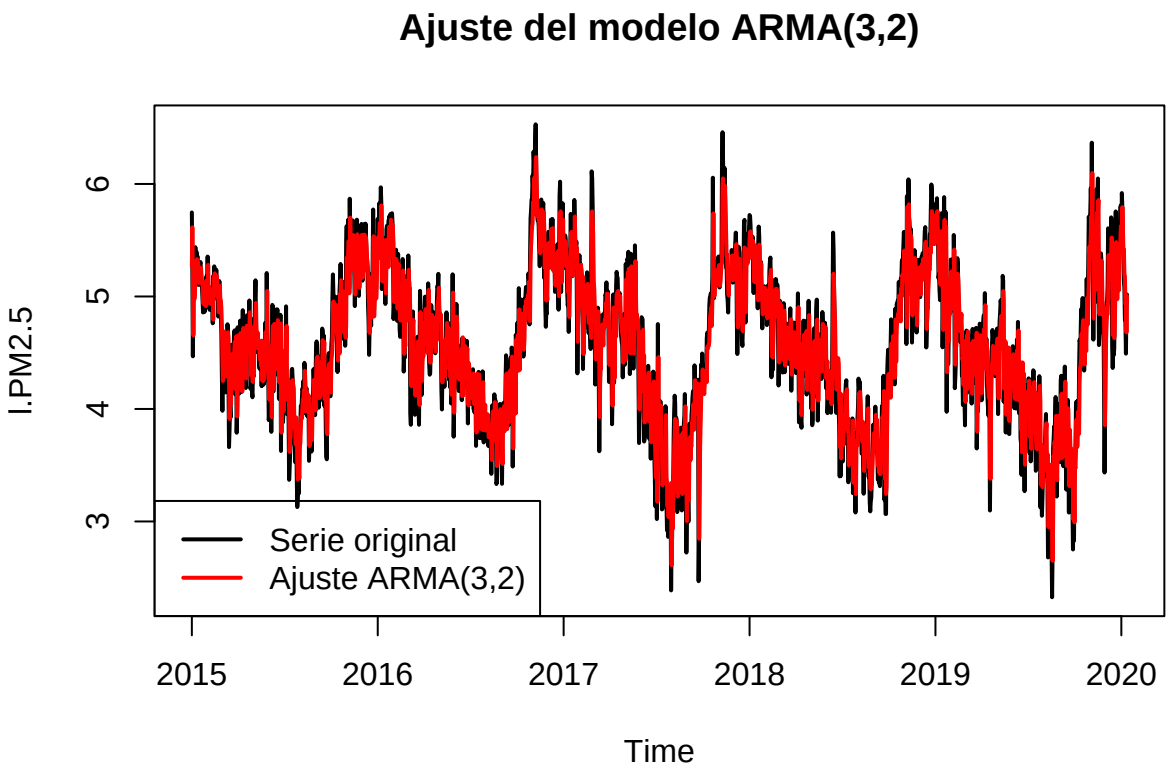
```
##
##  Shapiro-Wilk normality test
##
## data:  arma3.2$residuals
## W = 0.9851, p-value = 6.863e-13
```

### Gráficas de los modelos y la serie

```
plot(l.PM2.5, type = "l", col = "black", lwd = 2, main = "Ajuste del modelo AR(9)")
lines(fitted(ar9), col = "red", lwd = 2)
legend("bottomleft", legend = c("Serie original", "Ajuste AR(9)"),
       col = c("black", "red"), lwd = 2)
```




```
plot(l.PM2.5, type = "l", col = "black", lwd = 2, main = "Ajuste del modelo ARMA(3,2)")
lines(fitted(arma3.2), col = "red", lwd = 2)
legend("bottomleft", legend = c("Serie original", "Ajuste ARMA(3,2)"),
       col = c("black", "red"), lwd = 2)
```


Ajuste del modelo ARMA(3,2)
l.PM2.5
3
4
5
6
2015
2016
2017
2018
2019
2020
Time
Serie original
Ajuste ARMA(3,2)