

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
```

```
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

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

##
## Adjuntando el paquete: 'lubridate'
## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union

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 ($\mu\text{g}/\text{m}^3$) (partículas en suspensión con diámetro aerodinámico $\leq 2.5 \mu\text{m}$) 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.

```
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 $\mu\text{g}/\text{m}^3$, una mediana de 94.49 $\mu\text{g}/\text{m}^3$ y un máximo extremo de 685.36 $\mu\text{g}/\text{m}^3$.

```
# 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 ")
```

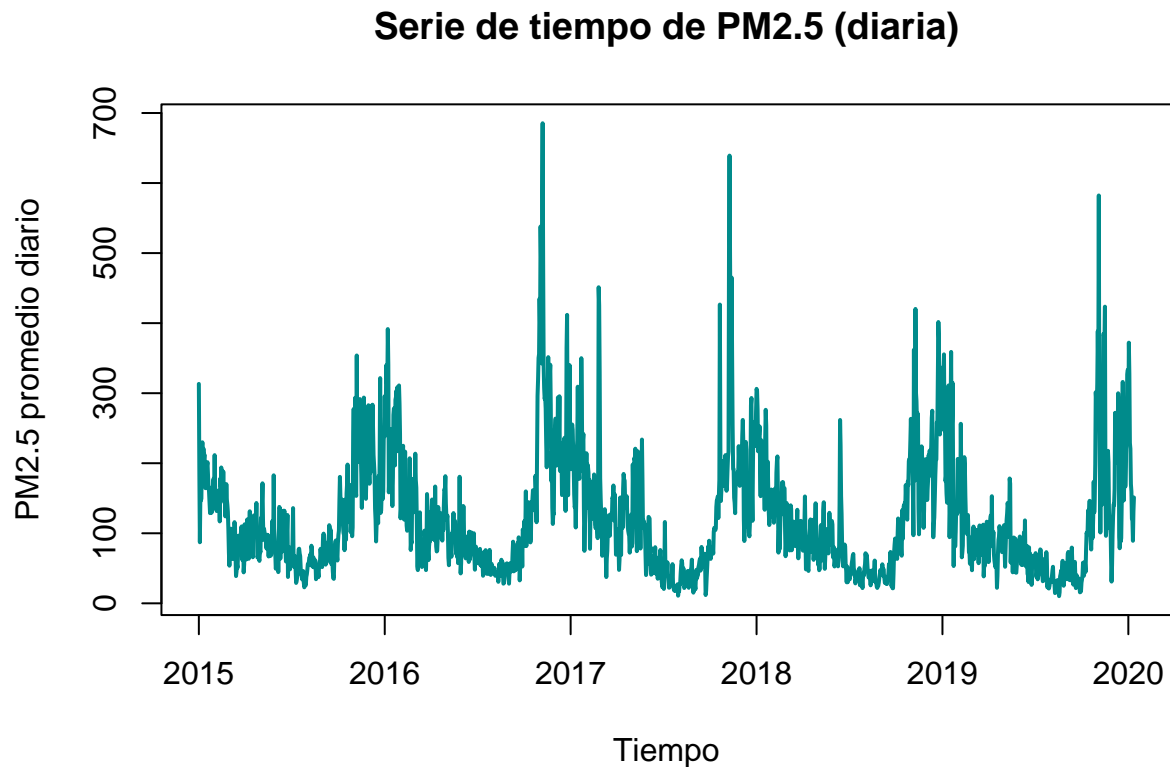
Table 2: Estadísticas descriptivas de PM2.5

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

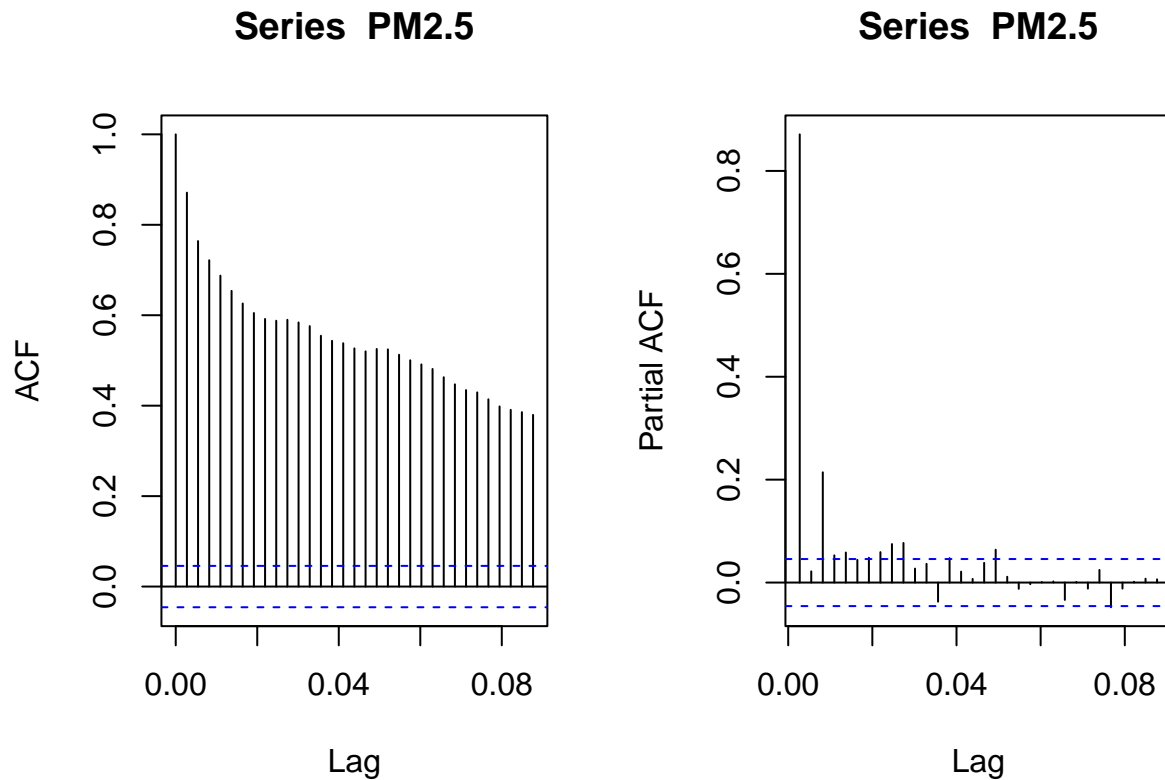
La gráfica de la serie de tiempo muestra una variabilidad, con estacionalidad clara (patrones repetidos, que parecieran ser anuales). Aunque no se observa una tendencia, la varianza no es constante puesto que los picos

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, obteniendo $\lambda \approx 0.257$, cercano a cero. Por lo cual se aplicó la transformación logarítmica.

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

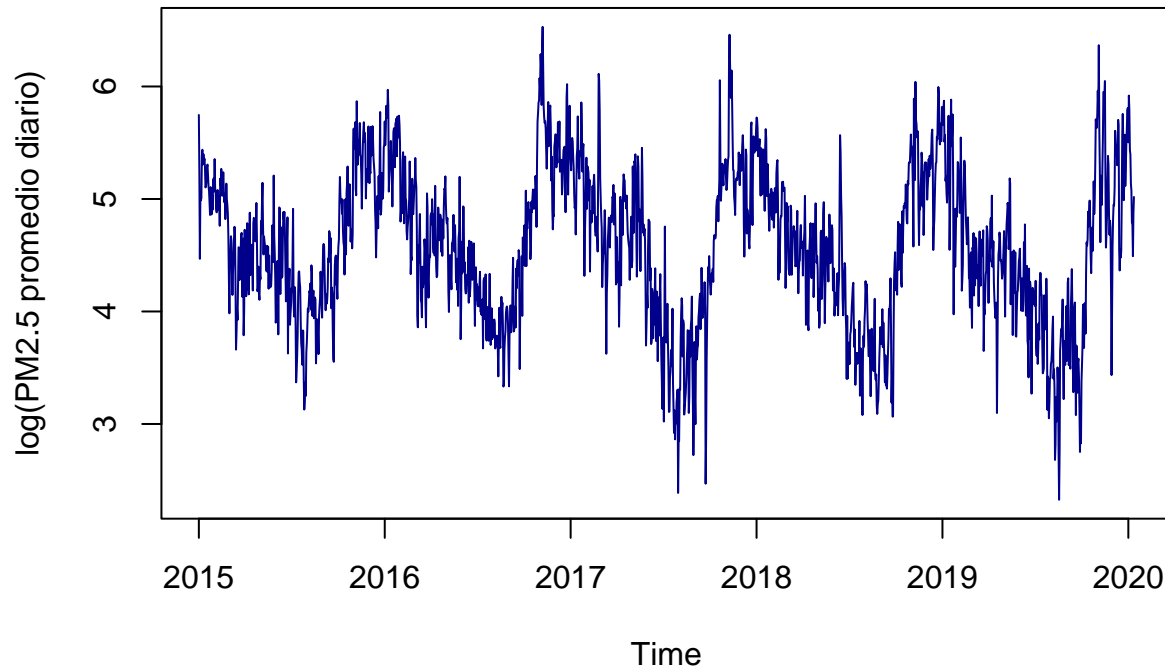
```
## [1] 0.2571413
```

Dada la heterocedasticidad, se evaluo una transformación para estabilizar la varianza. Se estimó el parámetro de Box-Cox, obteniendo $\lambda = 0.257$ (cercana a 0). Por lo cual se aplicó la transformación logarítmica.

```
# 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")
```

Serie transformada logarítmicamente



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

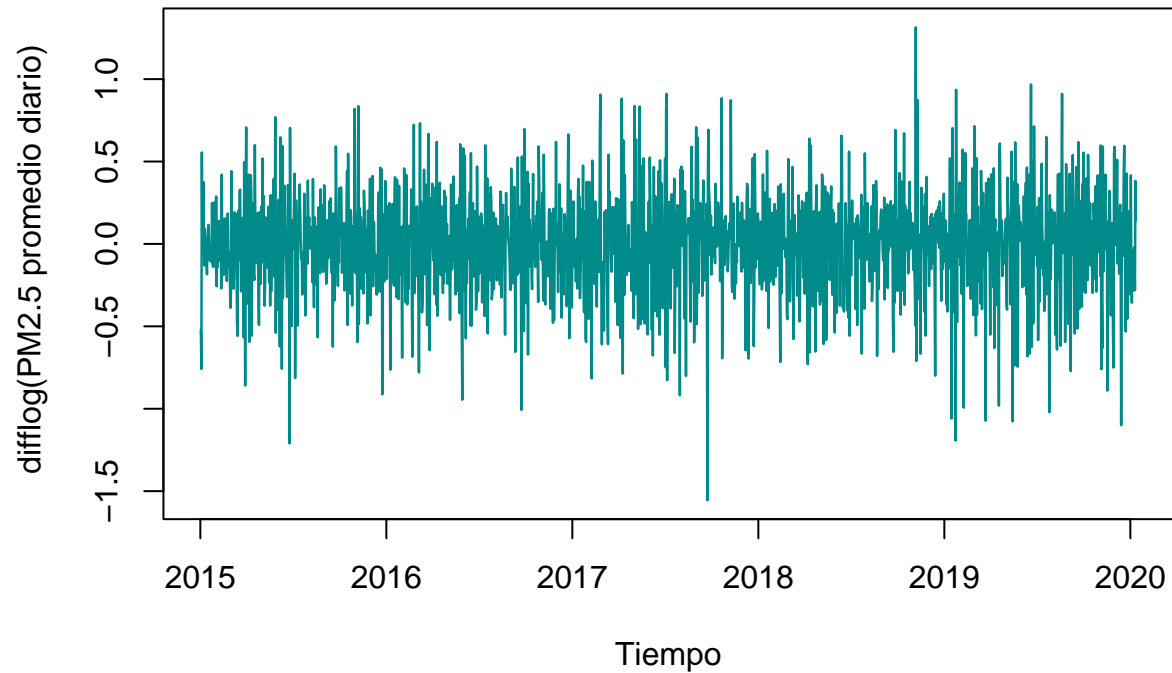
```
adf.test(l.PM2.5, alternative = "stationary")
```

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

Luego de aplicar la transformación logarítmica a la serie de PM2.5, el test de Dickey-Fuller Aumentado (ADF) arrojó un estadístico de -3.65 con un p-valor de 0.027, lo que indicaría estacionariedad a un nivel de significancia del 5%. Sin embargo, la inspección de la función de autocorrelación (ACF) revela un decaimiento lento, señal de persistencia temporal y posible estacionalidad anual. Por tanto, la serie log-transformada aún no puede considerarse estrictamente estacionaria, y se recomienda aplicar una diferenciación regular ($d = 1$)

```
d.l.PM2.5 <- diff(l.PM2.5) # Retorno de PM2.5  
ts.plot(d.l.PM2.5, col = "darkcyan", lwd = 1.25,  
        ylab = "difflog(PM2.5 promedio diario)", xlab = "Tiempo",  
        main = "Serie con diff de la transformación logaritmo")
```

Serie con diff de la transformación logaritmo



```
adf.test(d.l.PM2.5)
```

```
## Warning in adf.test(d.l.PM2.5): p-value smaller than printed p-value
```

```
##
```

```
## Augmented Dickey-Fuller Test
```

```
##
```

```
## data: d.l.PM2.5
```

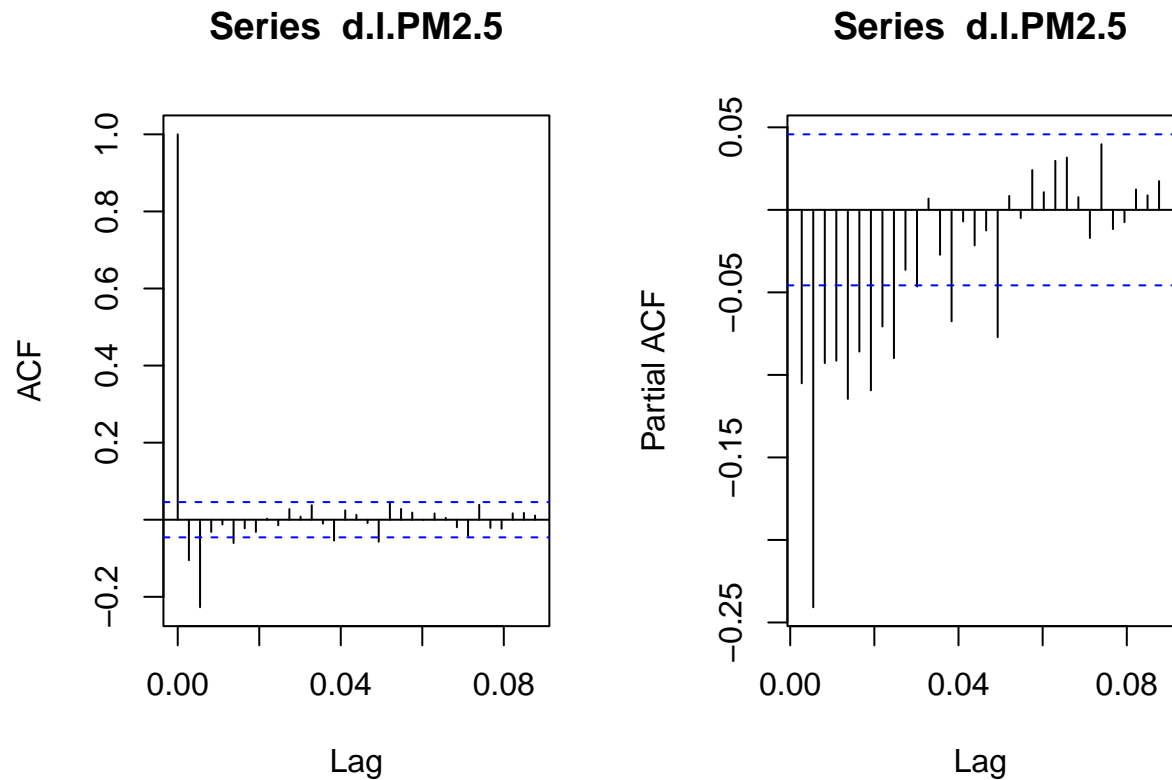
```
## Dickey-Fuller = -16.31, Lag order = 12, p-value = 0.01
```

```
## alternative hypothesis: stationary
```

```
par(mfrow=c(1,2))
```

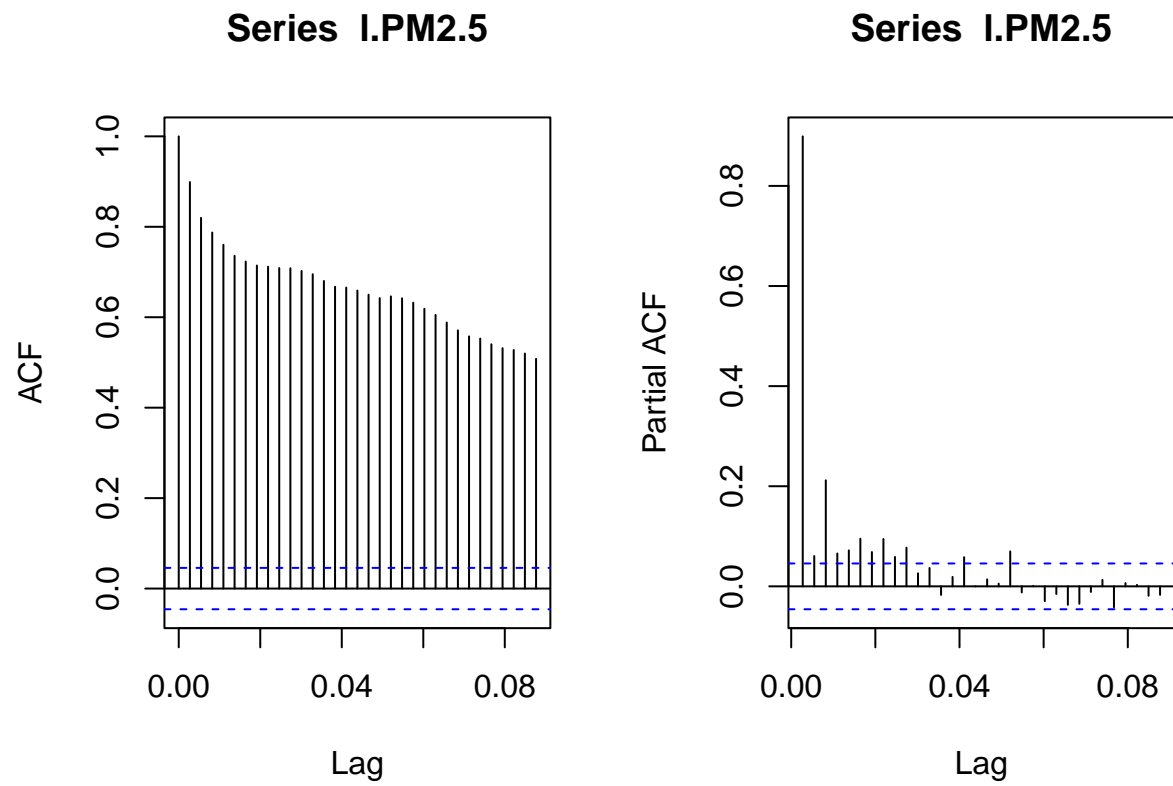
```
acf(d.l.PM2.5)
```

```
pacf(d.l.PM2.5)
```

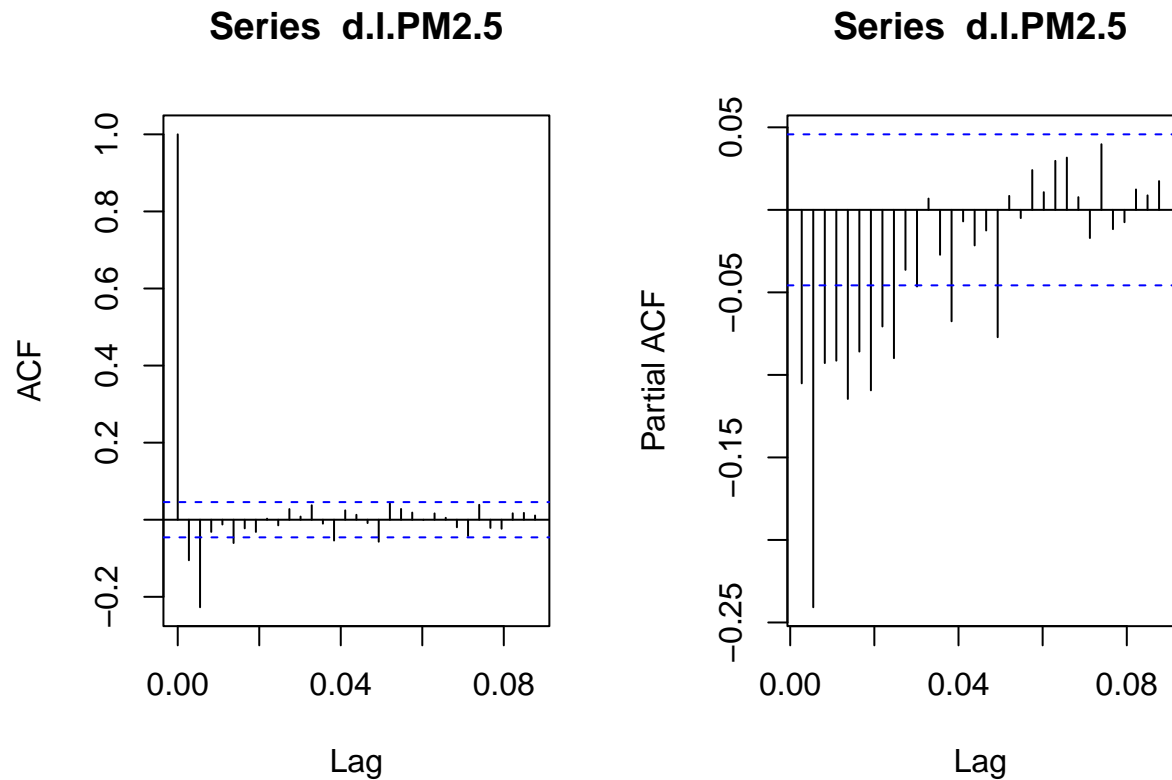



Luego de aplicar la transformación logarítmica, se realizó una diferenciación de primer orden (`difflog(PM2.5)`) con el fin de eliminar la tendencia presente en la serie. La inspección de la serie transformada muestra oscilaciones alrededor de una media constante, sin tendencias visibles. El test de Dickey–Fuller Aumentado (ADF) arrojó un estadístico de -16.31 con un p-valor < 0.01 , indicando que la serie diferenciada es estacionaria. Asimismo, la función de autocorrelación (ACF) se corta rápidamente tras el primer rezago, confirmando la ausencia de dependencia de largo plazo. En consecuencia, la serie transformada y diferenciada cumple los supuestos de estacionariedad necesarios para proceder con la identificación de un modelo ARIMA. # Modelos

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



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



Se presentan estos 2 modelos el primero para la serie transformada con log dado que, según el PACF (Función de Autocorrelación Parcial) de esta, se observa que el último rezago significativo es 2 (lo que sugiere un AR(1) o AR(2)). Por otro lado, con el ACF (Función de Autocorrelación) de la serie se observa un decaimiento lento, por lo que no se puede definir con certeza cuál sería la estructura de MA (Media Móvil). Por ello, se prueba el ARMA(2,2) (este último es el sugerido por la función `auto.arima()`), el otro modelo es un modelo ARIMA(2,1,2), ya que sería interesante comparar la serie sin diferenciar y diferenciandola.

```
arma2.2 <- arima(x = l.PM2.5, order = c(2,0,2))
arma2.2
```

```
##
## Call:
## arima(x = l.PM2.5, order = c(2, 0, 2))
##
## Coefficients:
##          ar1          ar2          ma1          ma2  intercept
##          1.3806   -0.3876   -0.5955   -0.2087         4.5934
## s.e.    0.0569    0.0557    0.0584    0.0368         0.1806
##
## sigma^2 estimated as 0.0822:  log likelihood = -312.7,  aic = 637.39
```

```
arma2.1.2 <- arima(x = l.PM2.5, order = c(2,1,2))
arma2.1.2
```

```
##
## Call:
## arima(x = l.PM2.5, order = c(2, 1, 2))
##
```

```
## Coefficients:
##          ar1      ar2      ma1      ma2
##      0.2156  0.1336 -0.4276 -0.3811
## s.e.  0.1156  0.0800   0.1111   0.1004
##
## sigma^2 estimated as 0.08253:  log likelihood = -315.45,  aic = 640.9

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

  # Modelo AR(9)
  "ARMA(2,2)" = c(coef(arma2.2)[5], coef(arma2.2)[-5], arma2.2$loglik, arma2.2$aic, BIC(arma2.2)),

  # Valor t (significancia) para AR(9)
  "t-valor ARMA(2,2)" = c((arma2.2$coef / sqrt(diag(arma2.2$var.coef)))[5],
                          (arma2.2$coef / sqrt(diag(arma2.2$var.coef)))[-5],
                          NA, NA, NA),

  # Modelo MA(9)
  "ARIMA(2,1,2)" = c(NA, coef(arima2.1.2),
                    arima2.1.2$loglik, arima2.1.2$aic, BIC(arima2.1.2)),

  # Valor t (significancia) para MA(9)
  "t-valor ARIMA(2,1,2)" = c(NA, arima2.1.2$coef / sqrt(diag(arima2.1.2$var.coef)),
                             NA, 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 | ARMA.2.2. | t.valor.ARMA.2.2. | ARIMA.2.1.2. | t.valor.ARIMA.2.1.2. |
|------------|-----------|-------------------|--------------|----------------------|
| Intercepto | 4.593 | 25.438 | NA | NA |
| ar1 | 1.381 | 24.282 | 0.216 | 1.866 |
| ar2 | -0.388 | -6.963 | 0.134 | 1.671 |
| ma1 | -0.595 | -10.203 | -0.428 | -3.849 |
| ma2 | -0.209 | -5.675 | -0.381 | -3.794 |
| logLik | -312.696 | NA | -315.449 | NA |
| AIC | 637.392 | NA | 640.897 | NA |
| BIC | 670.488 | NA | 668.474 | NA |

Se presentan a continuación las formulas de los modelos con los coeficientes estimados.

- *Formula ARMA(2,2)*

$$Z_t = 4.593Z_{t-1} - 0.388Z_{t-2} + a_t - 0.595a_{t-1} - 0.209a_{t-2} + 4.593$$

- *Formula ARIMA(2,1,2)*

$$\nabla Z_t = 0.216Z_{t-1} + 0.134Z_{t-2} + a_t - 0.428a_{t-1} - 0.381a_{t-2}$$

Se presentan a continuación las formulas de los modelos con las estimaciones de los coeficientes significativos:

- *Formula ARMA(2,2)*

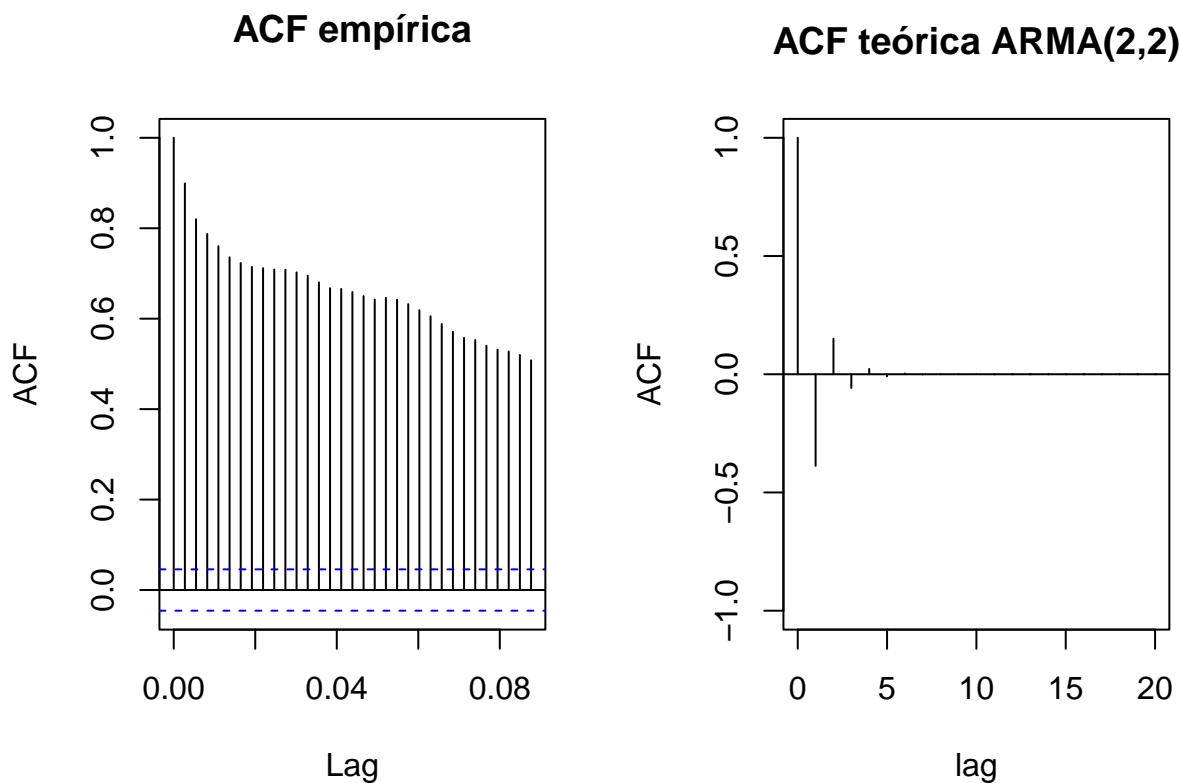
$$Z_t = 4.593Z_{t-1} - 0.388Z_{t-2} + a_t - 0.595a_{t-1} - 0.209a_{t-2} + 4.593$$

- *Formula ARIMA(2,1,2)*

$$\nabla Z_t = 0.216Z_{t-1} + a_t - 0.428a_{t-1} - 0.381a_{t-2}$$

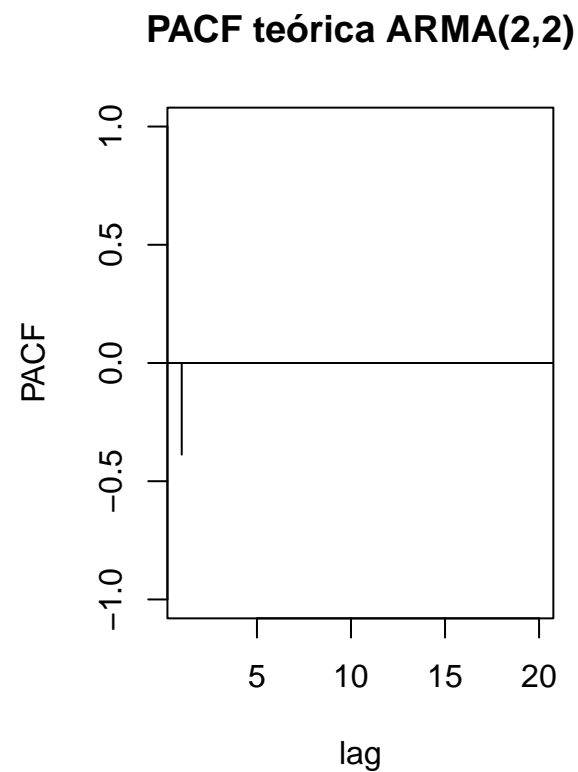
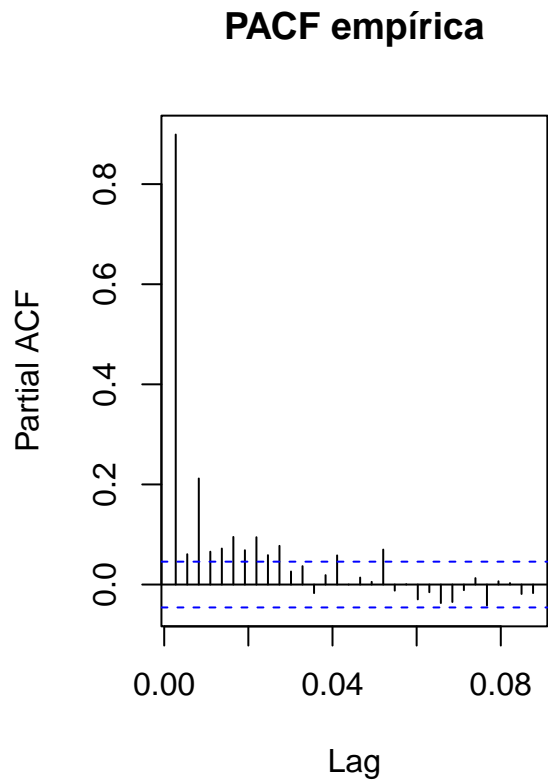
Para el modelo ARIMA(2,1,2) parece “reducirse” a un ARIMA(1,1,2) # Validación modelos ## ARMA(2,2)

```
par(mfrow = c(1, 2))
acf(1.PM2.5, main = "ACF empírica")
plotArmaTrueacf(list(ar = arma2.2$coef["ar2"]), main = "ACF teórica ARMA(2,2)")
```



```
## $roots
## [1] -0.3875542
##
## $acf
##          0          1          2          3          4
## 1.000000e+00 -3.875542e-01 1.501982e-01 -5.820996e-02 2.255951e-02
##          5          6          7          8          9
## -8.743034e-03 3.388400e-03 -1.313188e-03 5.089317e-04 -1.972386e-04
```

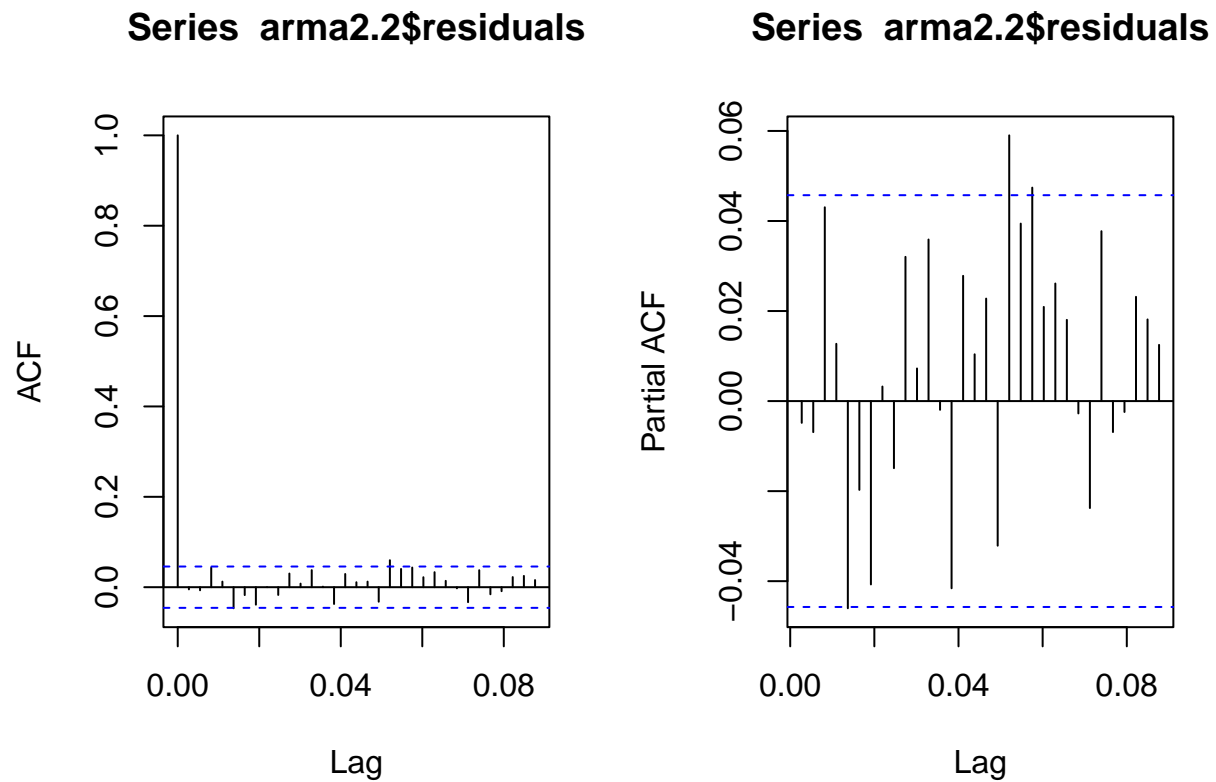
```
##          10          11          12          13          14
## 7.644065e-05 -2.962489e-05 1.148125e-05 -4.449607e-06 1.724464e-06
##          15          16          17          18          19
## -6.683232e-07 2.590115e-07 -1.003810e-07 3.890307e-08 -1.507705e-08
##          20
## 5.843172e-09
##
## $periodicity
## [1] damping period
## <0 rows> (o 0- extensión row.names)
pacf(1.PM2.5, main = "PACF empírica")
plotArmaTrueacf(list(ar = arma2.2$coef["ar2"]), pacf = TRUE, main = "PACF teórica ARMA(2,2)")
```



```
## $roots
## [1] -0.3875542
##
## $pacf
## [1] -0.3875542 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [7] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [13] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [19] 0.0000000 0.0000000
##
## $periodicity
## [1] damping period
## <0 rows> (o 0- extensión row.names)
```

```
par(mfrow=c(1,2))
acf(arma2.2$residuals)

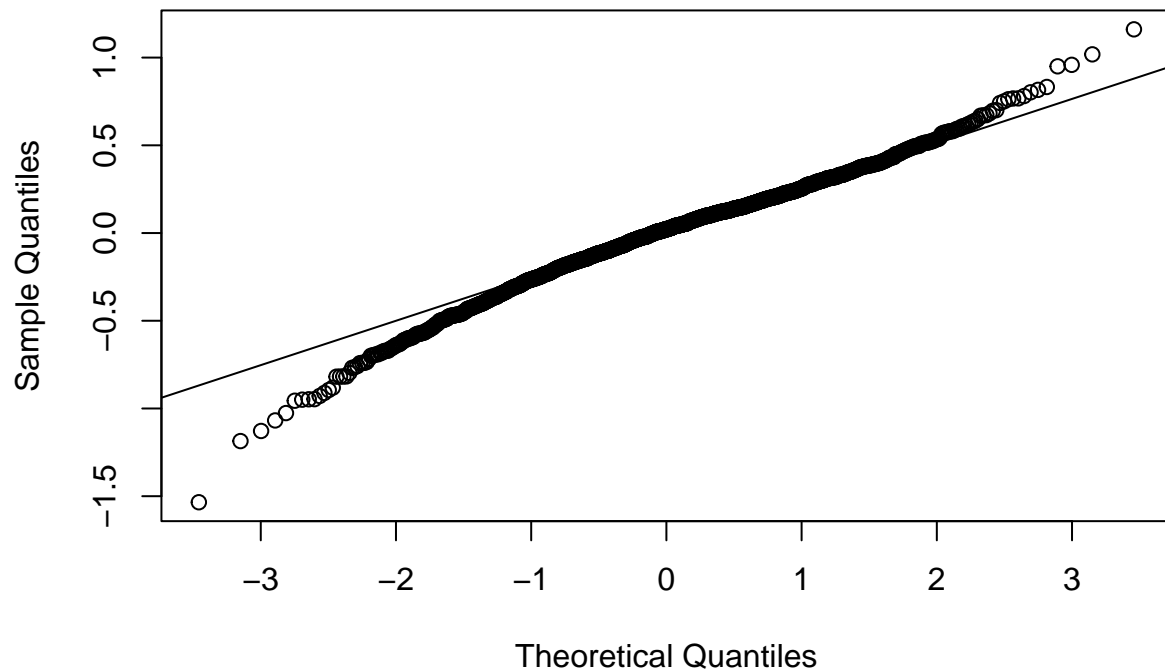
pacf(arma2.2$residuals)
```



Los residuales del modelo ARMA(2,2) parecen comportarse como un ruido blanco, aunque se ve un rezago en el PACF, este se encuentra muy lejos de los primeros, por lo que no se toma tanta importancia

```
qqnorm(arma2.2$residuals)
qqline(arma2.2$residuals)
```

Normal Q-Q Plot

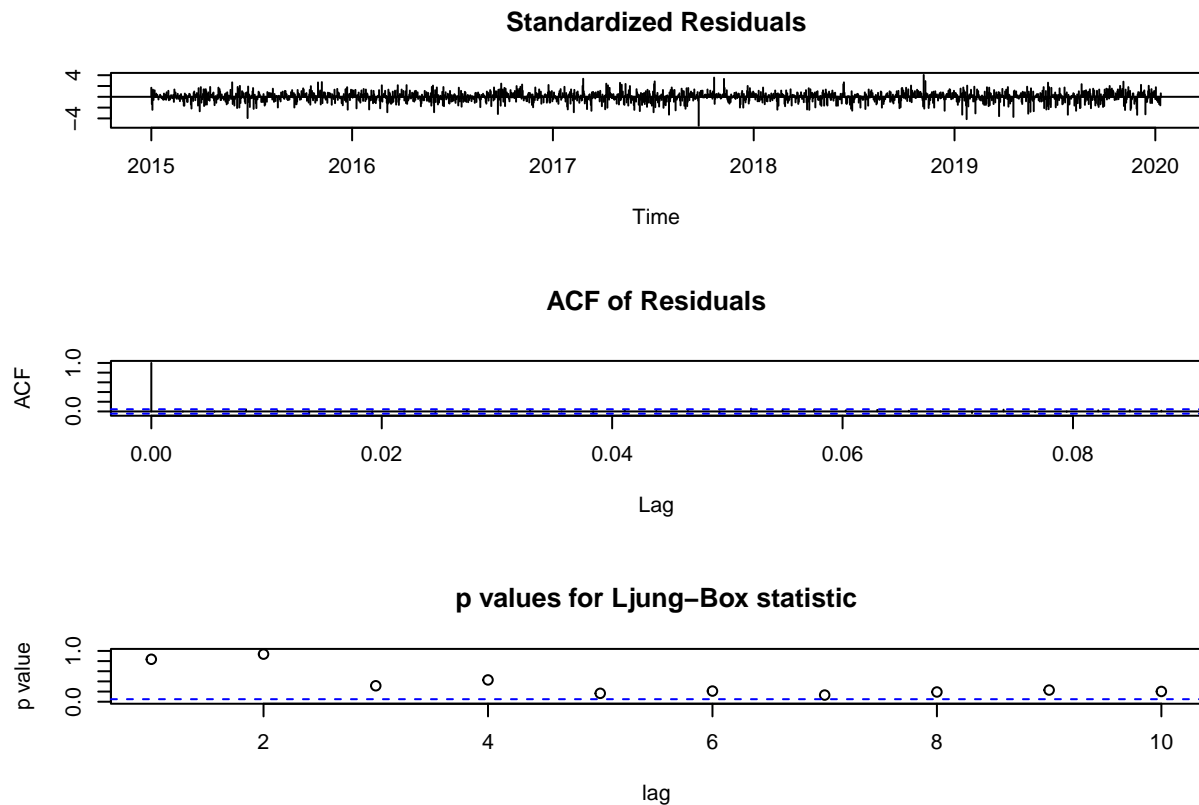


Los residuales aunque en su mayoría se encuentran alrededor de la línea, parece ser que se alejan mucho en las colas, lo que podría indicar no normalidad.

```
Box.test(arma2.2$residuals)
```

```
##  
## Box-Pierce test  
##  
## data: arma2.2$residuals  
## X-squared = 0.042943, df = 1, p-value = 0.8358
```

```
tsdiag(arma2.2)
```

La prueba de Ljung box muestra que los residuales son independientes ya que su p-valor aproximadamente de 0.84, mayor al nivel de significancia 5%, no se rechaza la hipótesis nula la cual indica independencia.

```
ks.test(arma2.2$residuals, "pnorm", mean = 0, sd = sd(arma2.2$residuals))
```

```
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: arma2.2$residuals
## D = 0.03942, p-value = 0.006631
## alternative hypothesis: two-sided
```

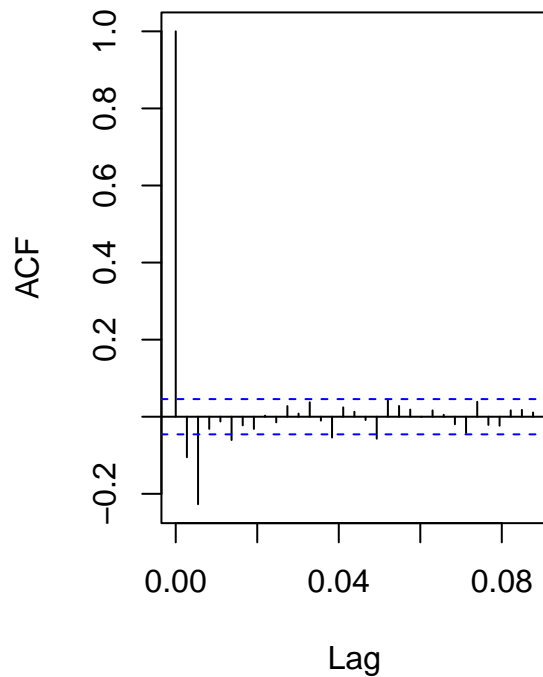
```
shapiro.test(arma2.2$residuals)
```

```
##
## Shapiro-Wilk normality test
##
## data: arma2.2$residuals
## W = 0.98539, p-value = 9.803e-13
```

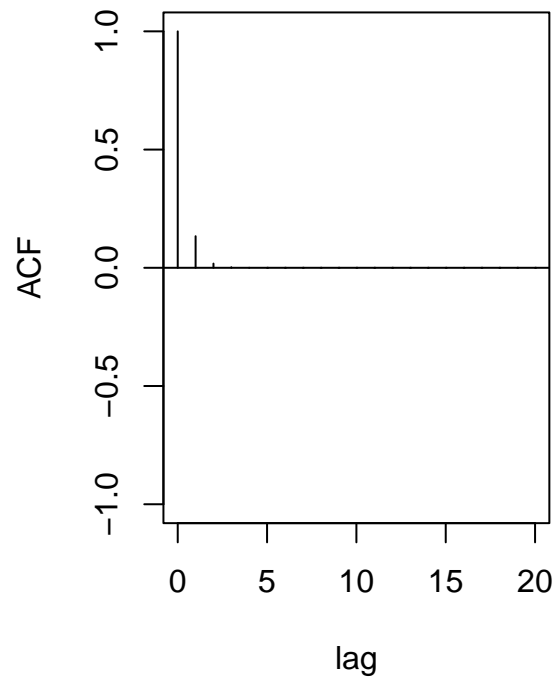
Respecto a la normalidad, el p-valor fue menor al nivel de significancia por lo que se rechaza la hipótesis nula de que los residuales provienen de una distribución normal. ## ARIMA(2,1,2)

```
par(mfrow = c(1, 2))
acf(d.l.PM2.5, main = "ACF empírica")
plotArmaTrueacf(list(ar = arima2.1.2$coef["ar2"]), main = "ACF teórica ARIMA(2,1,2)")
```

ACF empírica

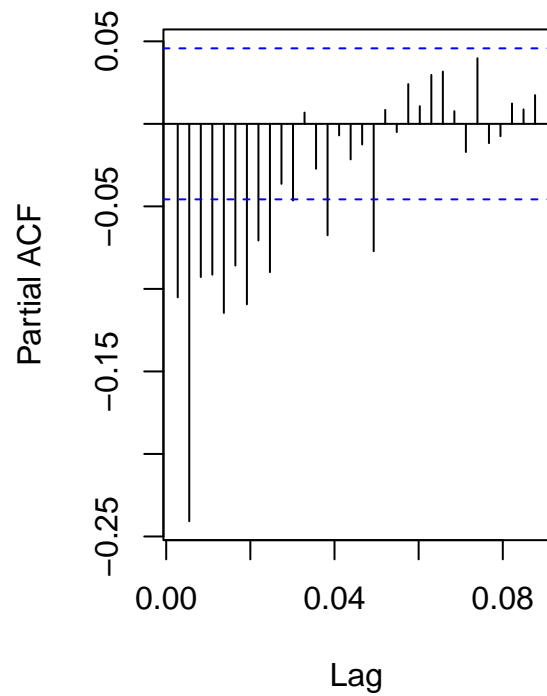


ACF teórica ARiMA(2,1,2)

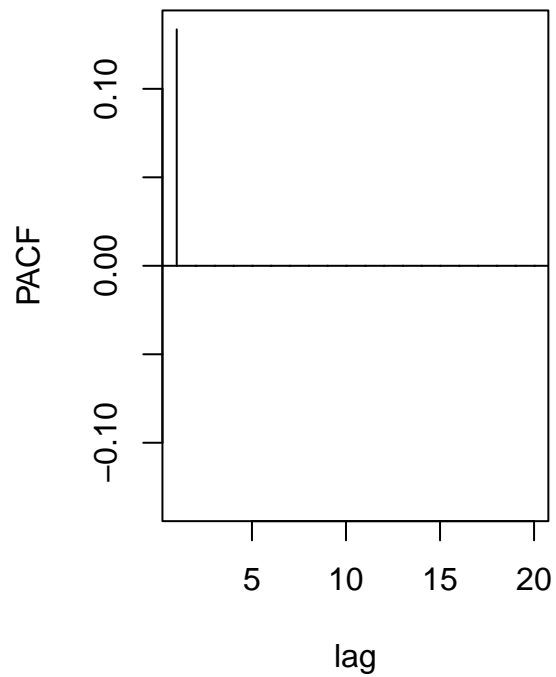


```
## $roots
## [1] 0.1336186
##
## $acf
##          0          1          2          3          4          5
## 1.000000e+00 1.336186e-01 1.785394e-02 2.385619e-03 3.187632e-04 4.259270e-05
##          6          7          8          9         10         11
## 5.691178e-06 7.604474e-07 1.016099e-07 1.357698e-08 1.814138e-09 2.424026e-10
##          12         13         14         15         16         17
## 3.238951e-11 4.327842e-12 5.782803e-13 7.726903e-14 1.032458e-14 1.379557e-15
##          18         19         20
## 1.843345e-16 2.463052e-17 3.291096e-18
##
## $periodicity
## [1] damping period
## <0 rows> (o 0- extensión row.names)
pacf(d.l.PM2.5, main = "PACF empírica")
plotArmaTrueacf(list(ar = arima2.1.2$coef["ar2"]), pacf = TRUE, main = "PACF teórica ARIMA(2,1,2)")
```

PACF empírica



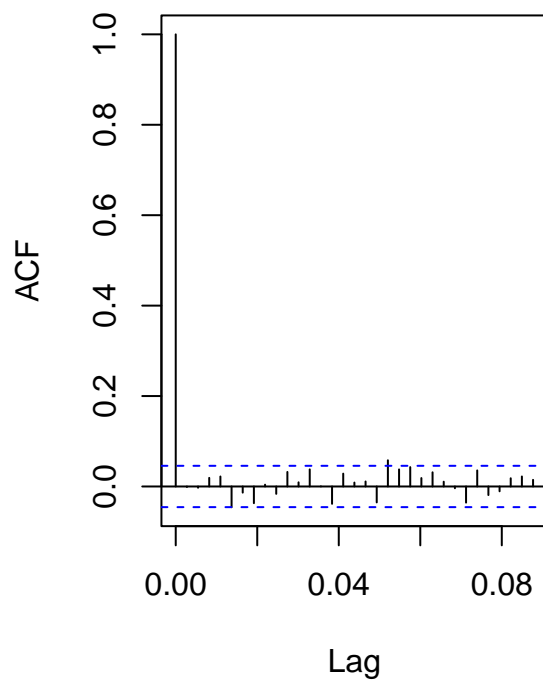
PACF teórica ARIMA(2,1,2)



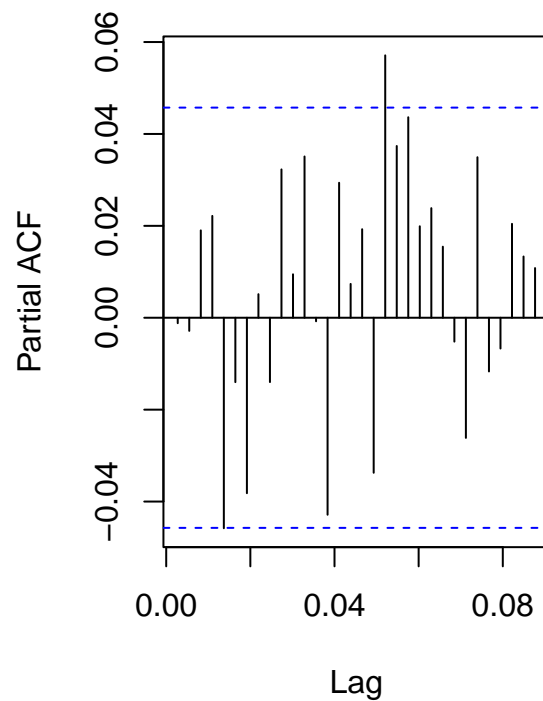
```
## $roots
## [1] 0.1336186
##
## $pacf
## [1] 0.1336186 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [8] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [15] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##
## $periodicity
## [1] damping period
## <0 rows> (o 0- extensión row.names)

par(mfrow=c(1,2))
acf(arima2.1.2$residuals)
pacf(arima2.1.2$residuals)
```

Series arima2.1.2\$residuals



Series arima2.1.2\$residuals

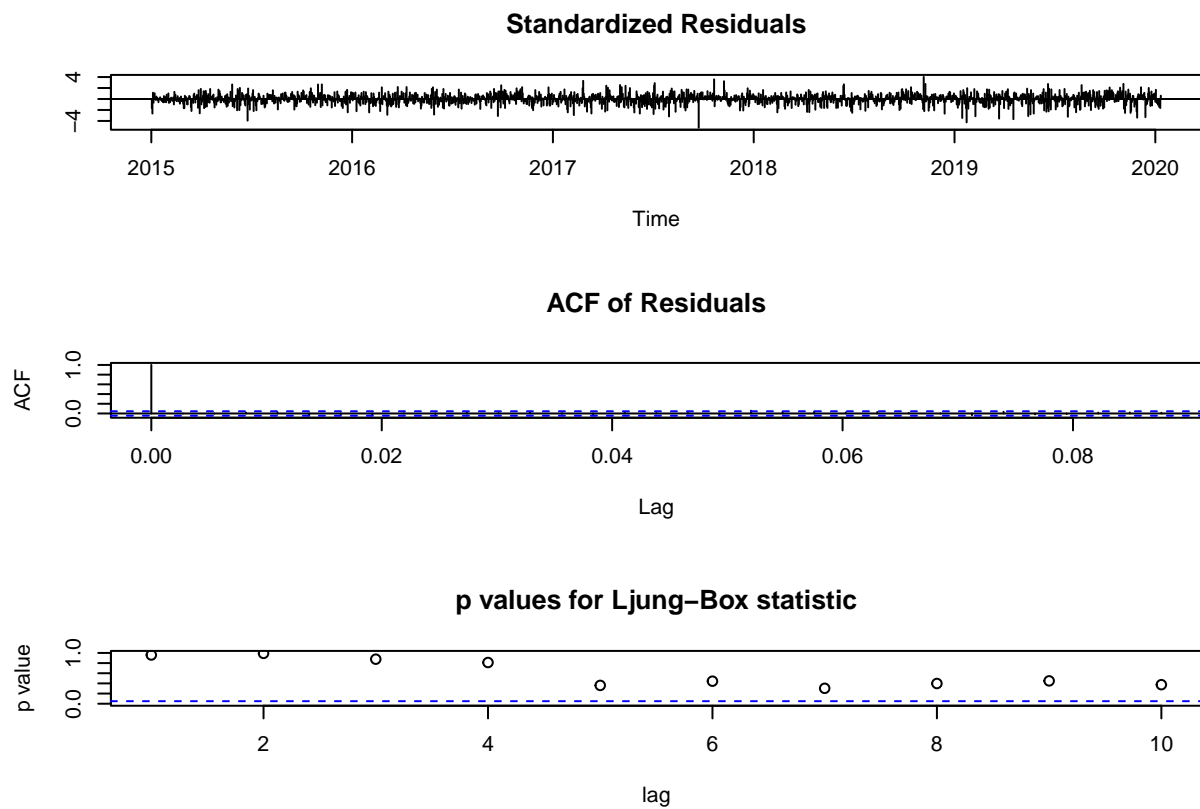


Los residuales del modelo ARIMA(2,1,2), al igual que con el anterior, parecen comportarse como un ruido blanco, e incluso tambien presentan un rezago en el PACF, con el mismo comportamiento del anterior, al ser tan lejano de los primeros valores, no se lo toma importancia.

```
Box.test(arima2.1.2$residuals)
```

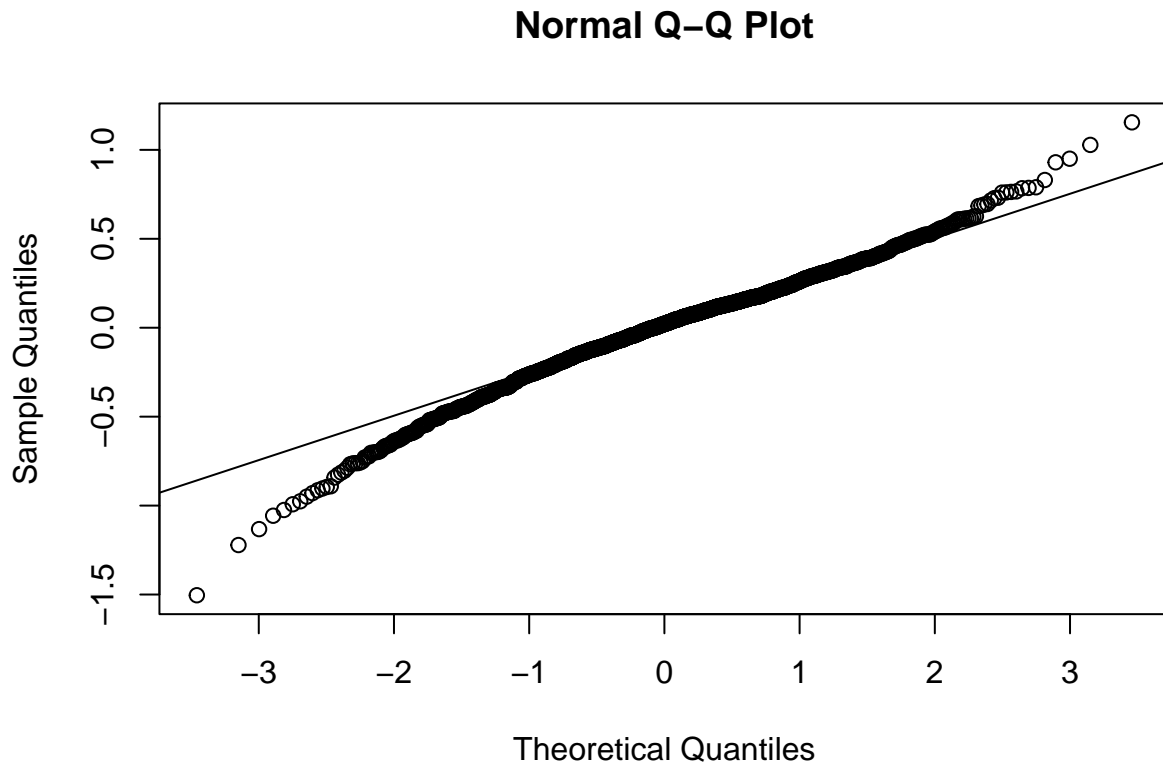
```
##
## Box-Pierce test
##
## data:  arima2.1.2$residuals
## X-squared = 0.0025837, df = 1, p-value = 0.9595
```

```
tsdiag(arima2.1.2)
```



La prueba de Ljung box muestra que los residuales son independientes ya que su p-valor aproximadamente de 0.95, mayor al nivel de significancia 5%, por lo que no se rechaza la hipótesis nula la cual indica independencia.

```
qqnorm(arima2.1.2$residuals)
qqline(arima2.1.2$residuals)
```



Al igual que con el modelo ARMA(2,2) parece haber problemas de normalidad ya que los datos suelen alejarse de la línea en las colas.

```
ks.test(arma2.2$residuals, "pnorm", mean = 0, sd = sd(arima2.1.2$residuals))
```

```
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: arma2.2$residuals
## D = 0.039498, p-value = 0.006483
## alternative hypothesis: two-sided
```

```
shapiro.test(arima2.1.2$residuals)
```

```
##
## Shapiro-Wilk normality test
##
## data: arima2.1.2$residuals
## W = 0.9858, p-value = 1.652e-12
```

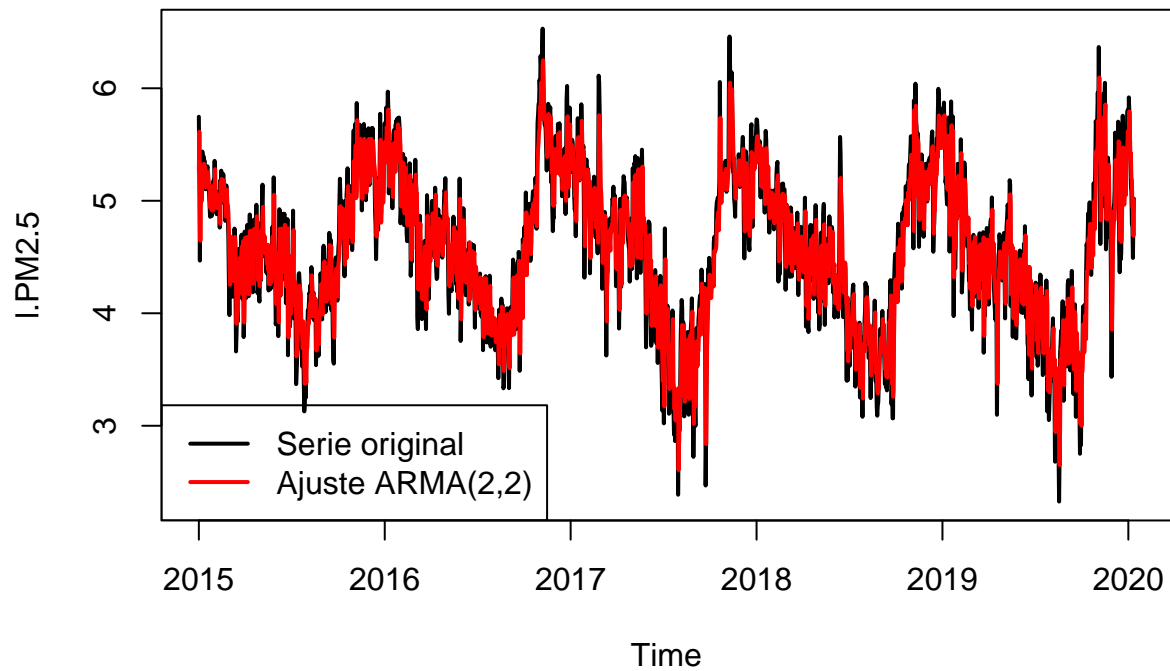
Efectivamente, al realizar la prueba de normalidad el p-valor fue menor al nivel de significancia por lo que se rechaza la hipótesis nula de que los residuales provienen de una distribución normal.

Gráficas de los modelos y la serie

```
plot(1.PM2.5, type = "l", col = "black", lwd = 2, main = "Ajuste del modelo ARMA(2,2)")
lines(fitted(arma2.2), col = "red", lwd = 2)
legend("bottomleft", legend = c("Serie original", "Ajuste ARMA(2,2)"),
```

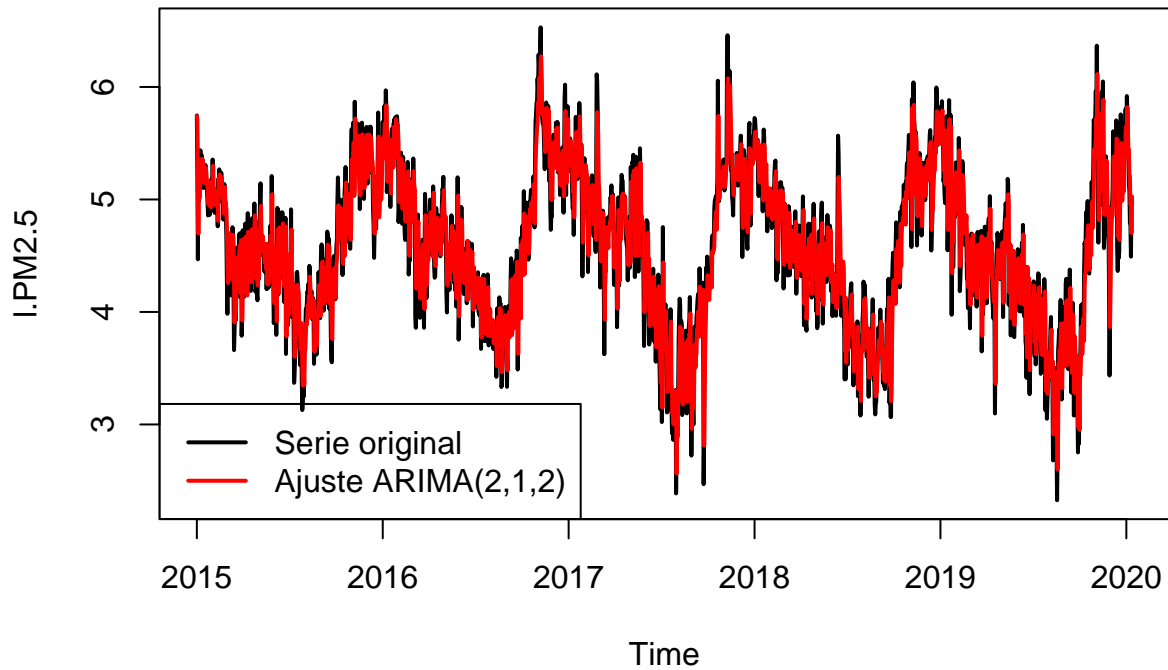
```
col = c("black", "red"), lwd = 2)
```

Ajuste del modelo ARMA(2,2)



```
plot(l.PM2.5, type = "l", col = "black", lwd = 2, main = "Ajuste del modelo ARIMA(2,1,2)")
lines(fitted(arima2.1.2), col = "red", lwd = 2)
legend("bottomleft", legend = c("Serie original", "Ajuste ARIMA(2,1,2)"),
      col = c("black", "red"), lwd = 2)
```

Ajuste del modelo ARIMA(2,1,2)



Ambos modelos parecen ajustarse bien a los datos. # Selección del mejor modelo Recordando que el modelo ARMA(2,2) obtuvo mejores AIC, y loglik que el modelo ARIMA(2,1,2), y observando que ambos modelos tienen un comportamiento similar en los residuales y que ambos parecen ajustarse bien a los datos, por criterio de parsimonia el mejor modelo es el ARMA(2,2)