Actividad 8: Series de tiempo

Daniela Jiménez Téllez

2024-11-12

Problema

Para los datos de las ventas de televisores analiza la serie de tiempo más apropiada:

Importación de datos

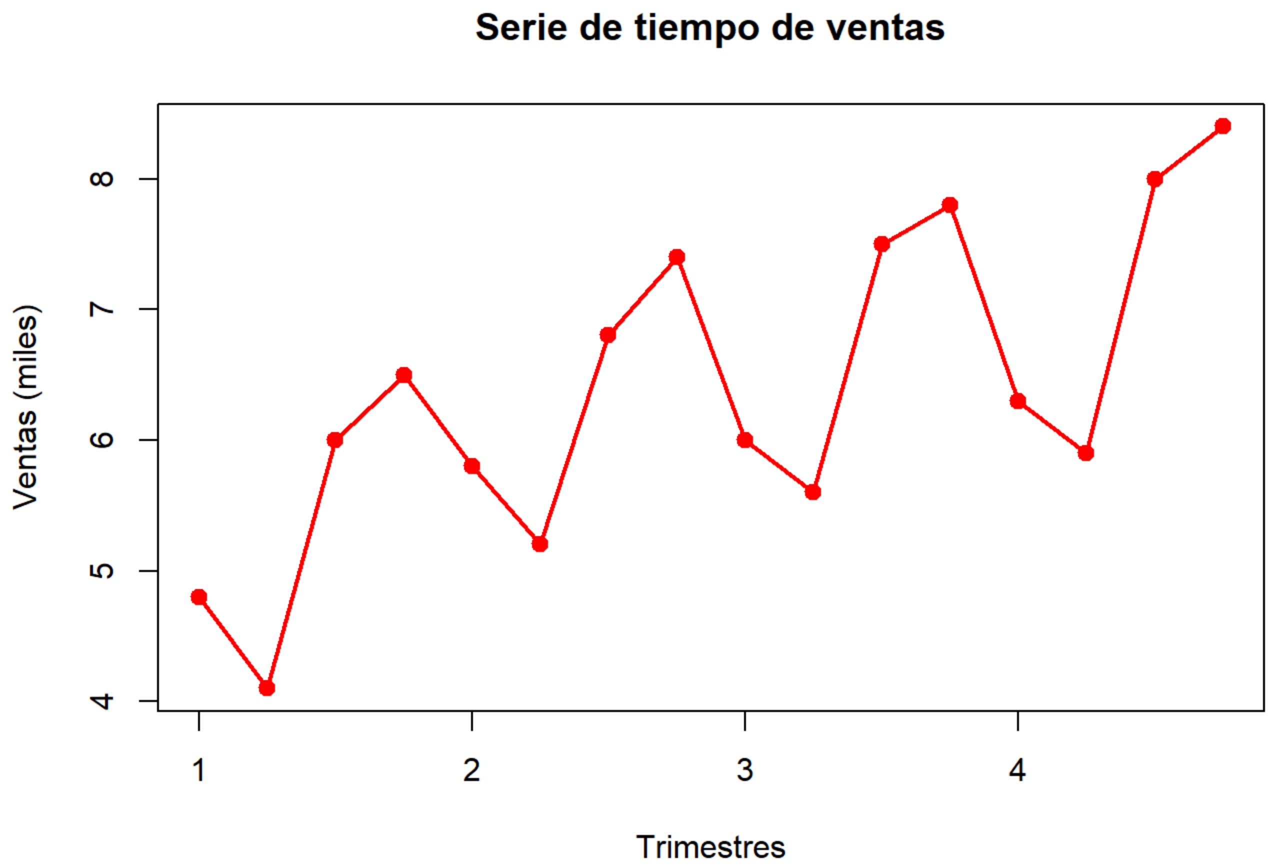
```
ventas_data <- data.frame(
  Año = c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4),
  Trimestre = c(1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4),
  Ventas = c(4.8, 4.1, 6.0, 6.5, 5.8, 5.2, 6.8, 7.4, 6.0, 5.6, 7.5, 7.8, 6.3, 5.9, 8.0, 8.4))

ventas_data$Tiempo <- 1:nrow(ventas_data)
```

Instrucciones

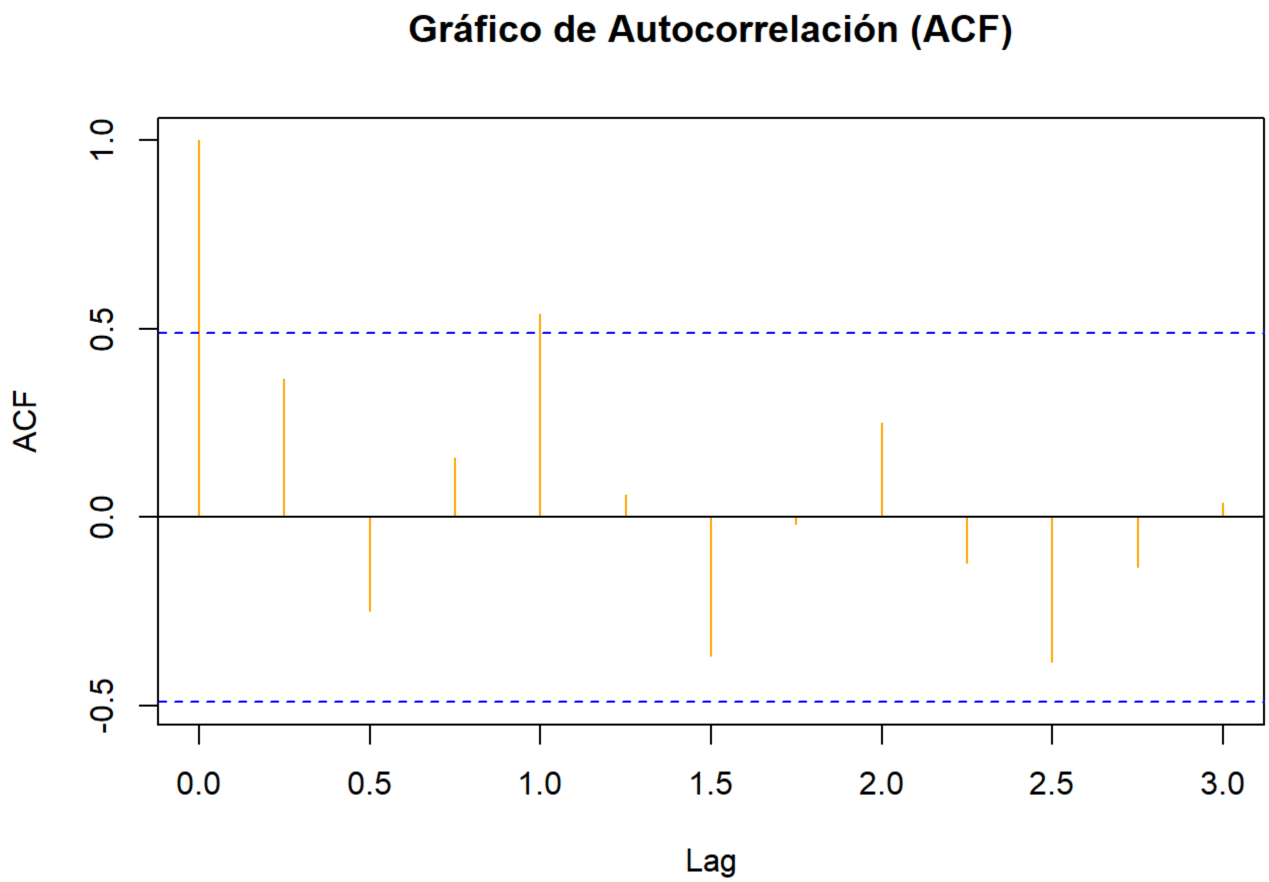
1. Realiza el análisis de tendencia y estacionalidad:

- Identifica si es una serie estacionaria
- Grafica la serie para verificar su tendencia y estacionalidad
- Analiza su gráfico de autocorrelación
- Identifica si el modelo puede ser sumativo o multiplicativo (puedes probar con ambos para ver con cuál es mejor el modelo)

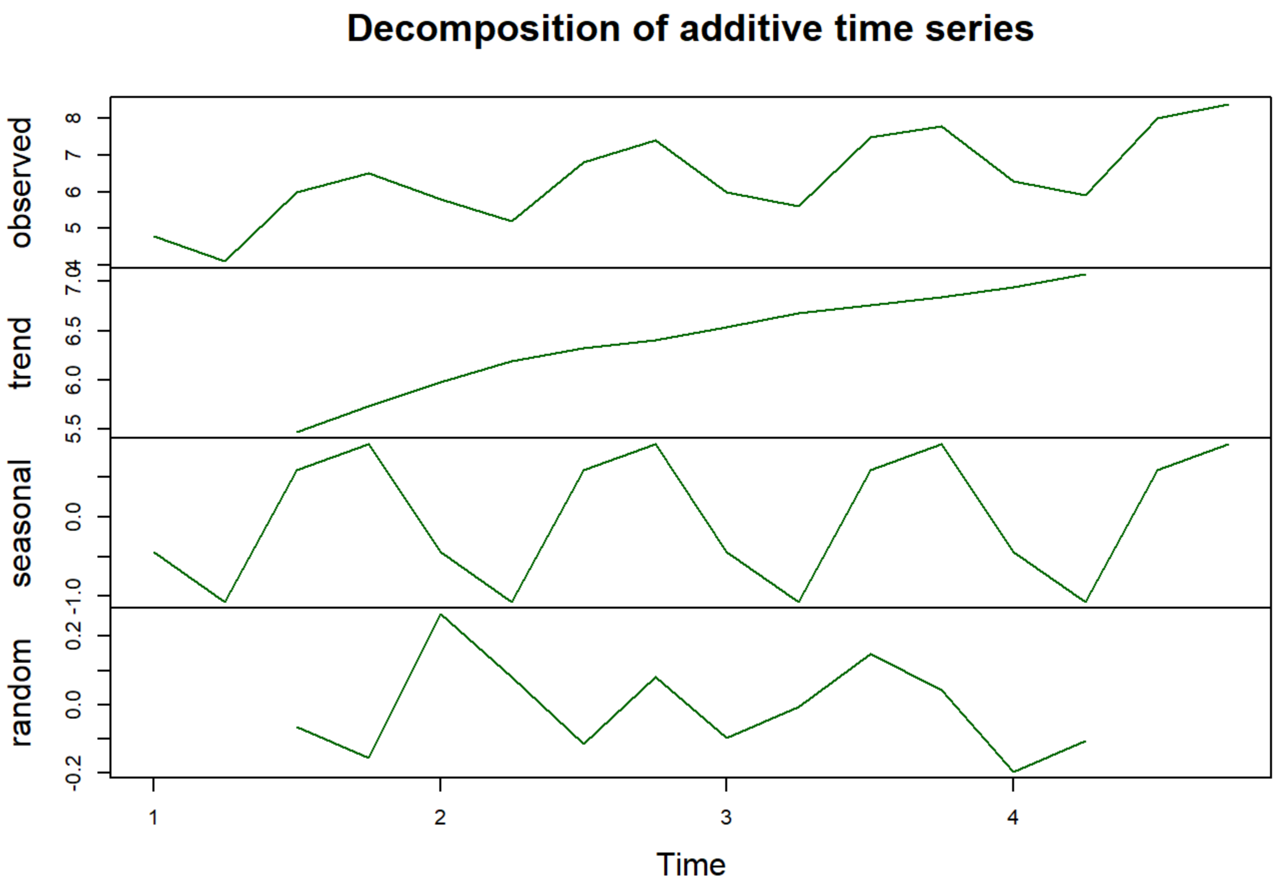
```
ventas_ts <- ts(ventas_data$Ventas, start = c(1, 1), frequency = 4)

# Gráfica de la serie

plot(ventas_ts, main = "Serie de tiempo de ventas", xlab = "Trimestres", ylab = "Ventas (mile
s)", type = "o", pch = 19, lwd = 2, col = "red")
```



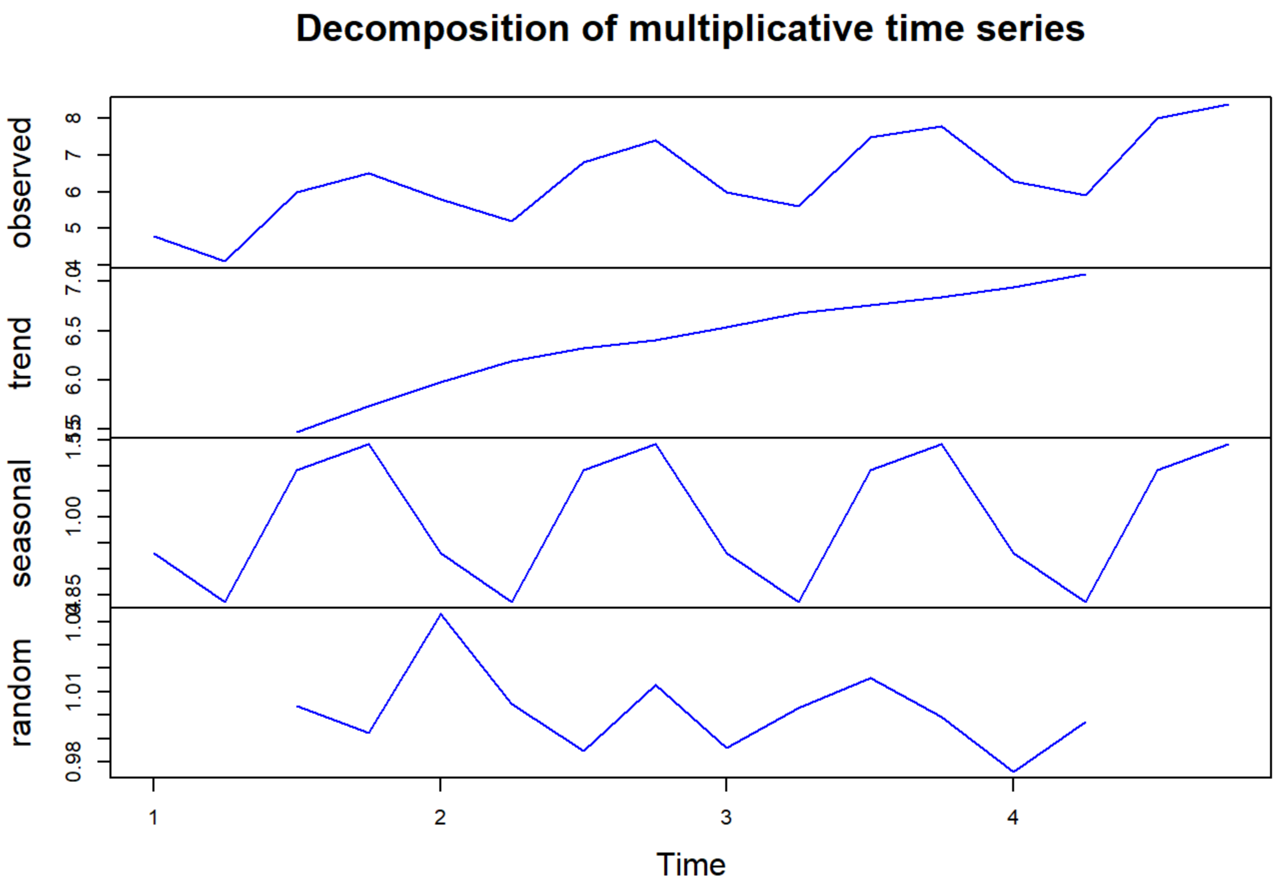

```
# Gráfico de correlación

acf(ventas_ts, main = "Gráfico de Autocorrelación (ACF)", col = "orange")
```




```r
# Descomposición aditiva

ventas_decomp_add <- decompose(ventas_ts, type = "additive")
plot(ventas_decomp_add, col = "darkgreen")
```




```
# Descomposición multiplicativa

ventas_decomp_mult <- decompose(ventas_ts, type = "multiplicative")
plot(ventas_decomp_mult, col = "blue")
```



De acuerdo a la gráfica, se puede ver que esta serie es no estacionaria con tendencia creciente y media dependiente del tiempo, lo que nos dice que los valores aumentan constantemente a lo largo del tiempo. Además, en la gráfica de autocorrelación muestra picos importantes en el principio de la gráfica. Esto indica que la serie no cumple con la condición de estacionariedad, ya que sus estadísticas no son constantes en el tiempo y están influenciadas por patrones estacionales y tendencias.

```
resumen_ventas <- ventas_data %>%
  group_by(Año) %>%
  summarise(
    Media = mean(Ventas),
    DesviacionEstandar = sd(Ventas))

print(resumen_ventas)
```

```
## # A tibble: 4 × 3
##     Año Media DesviacionEstandar
##   <dbl> <dbl>              <dbl>
## 1     1  5.35              1.10
## 2     2  6.3               0.987
## 3     3  6.72              1.09
## 4     4  7.15              1.23
```

Ahora, de acuerdo a esta definición de la Universidad de Barcelona, "si a medida que aumenta (o disminuye) el valor medio la desviación típica aumenta (o disminuye) el esquema de agregación adecuado es el multiplicativo; si las desviaciones típicas se mantienen aproximadamente iguales sea cual sea el valor de la media, el modelo adecuado es el aditivo". Observando los cálculos anteriores, se puede decir que los valores de la desviación estándar indican que no existe un incremento proporcional a la media de cada año. Las desviaciones estándar son relativamente estables y no aumentan consistentemente con el valor medio. Es por esto que el modelo es aditivo.

2. Calcula los índices estacionales y grafica la serie desestacionalizada

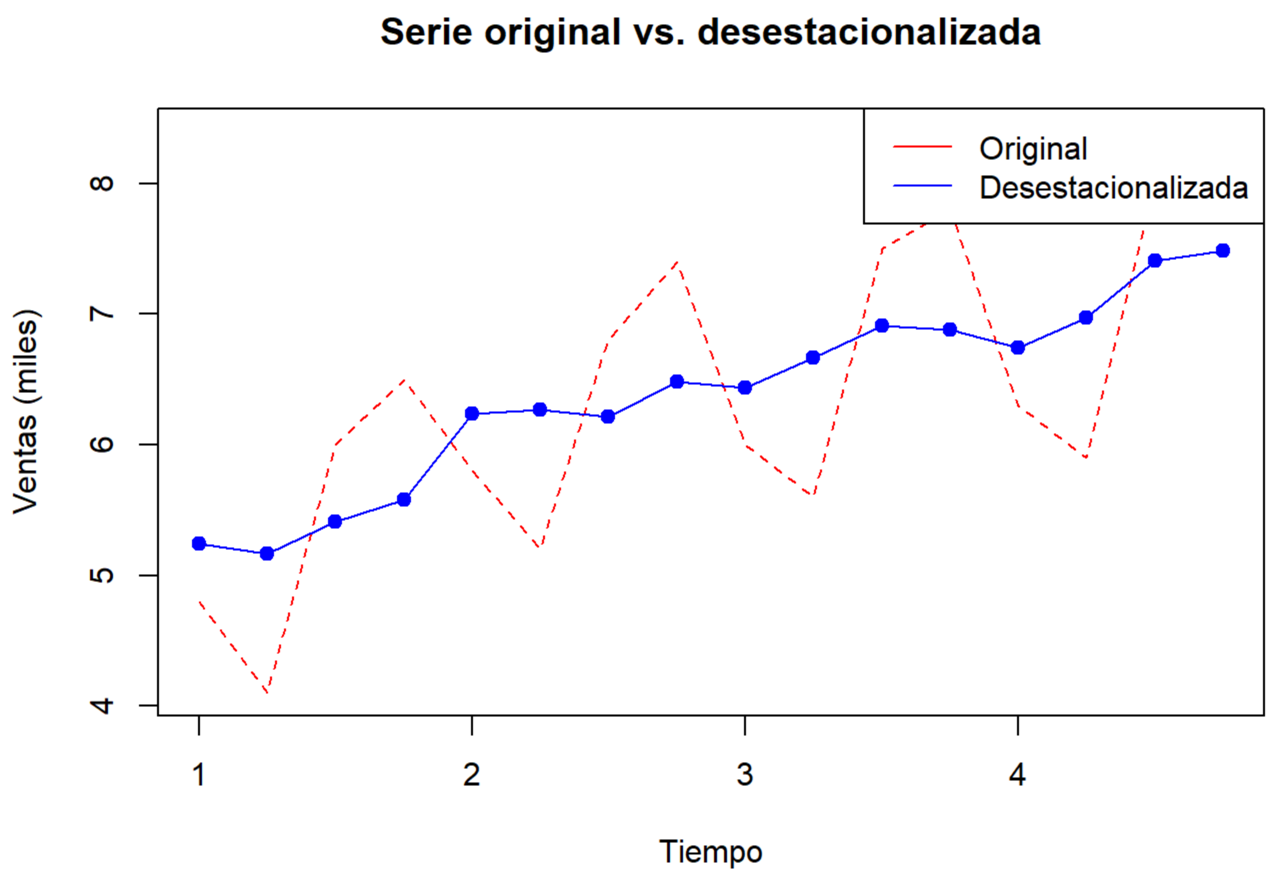
```
indices_estacionales <- ventas_decomp_add$seasonal

print("Índices Estacionales:")
```

```
## [1] "Índices Estacionales:"
```

```
print(indices_estacionales)
```

```
##        Qtr1       Qtr2       Qtr3       Qtr4
## 1 -0.4395833 -1.0687500  0.5895833  0.9187500
## 2 -0.4395833 -1.0687500  0.5895833  0.9187500
## 3 -0.4395833 -1.0687500  0.5895833  0.9187500
## 4 -0.4395833 -1.0687500  0.5895833  0.9187500
```

```
ventas_desestacionalizadas <- ventas_ts - indices_estacionales

plot(ventas_ts, main = "Serie original vs. desestacionalizada", xlab = "Tiempo", ylab = "Ventas
(miles)", col = "red", lty = 2)
lines(ventas_desestacionalizadas, col = "blue", type = "o", pch = 19)
legend("topright", legend = c("Original", "Desestacionalizada"), col = c("red", "blue"), lty =
1)
```



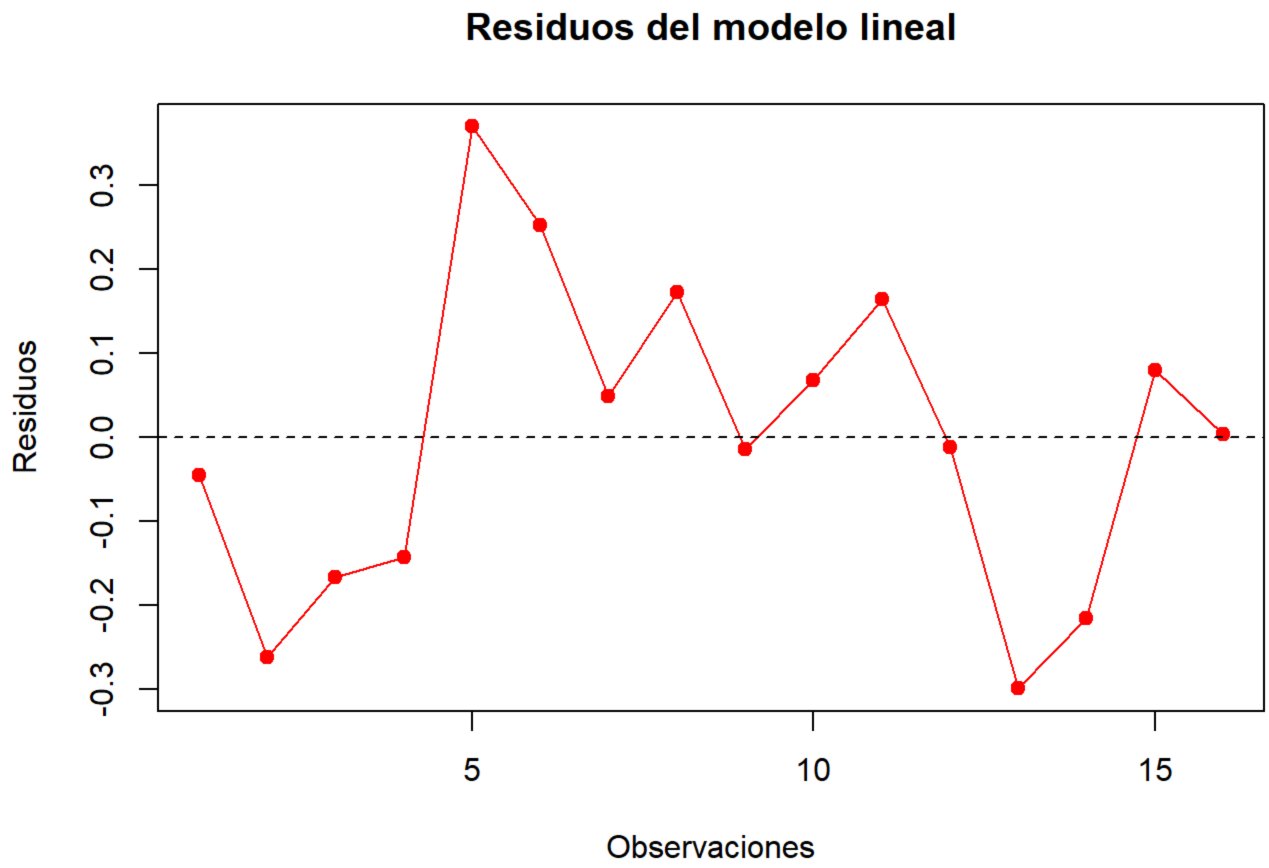
3. Analiza el modelo lineal de la tendencia

- Realiza la regresión lineal de la tendencia (ventas desestacionalizadas vs tiempo)
- Analiza la significancia del modelo lineal, global e individual
- Haz el análisis de residuos

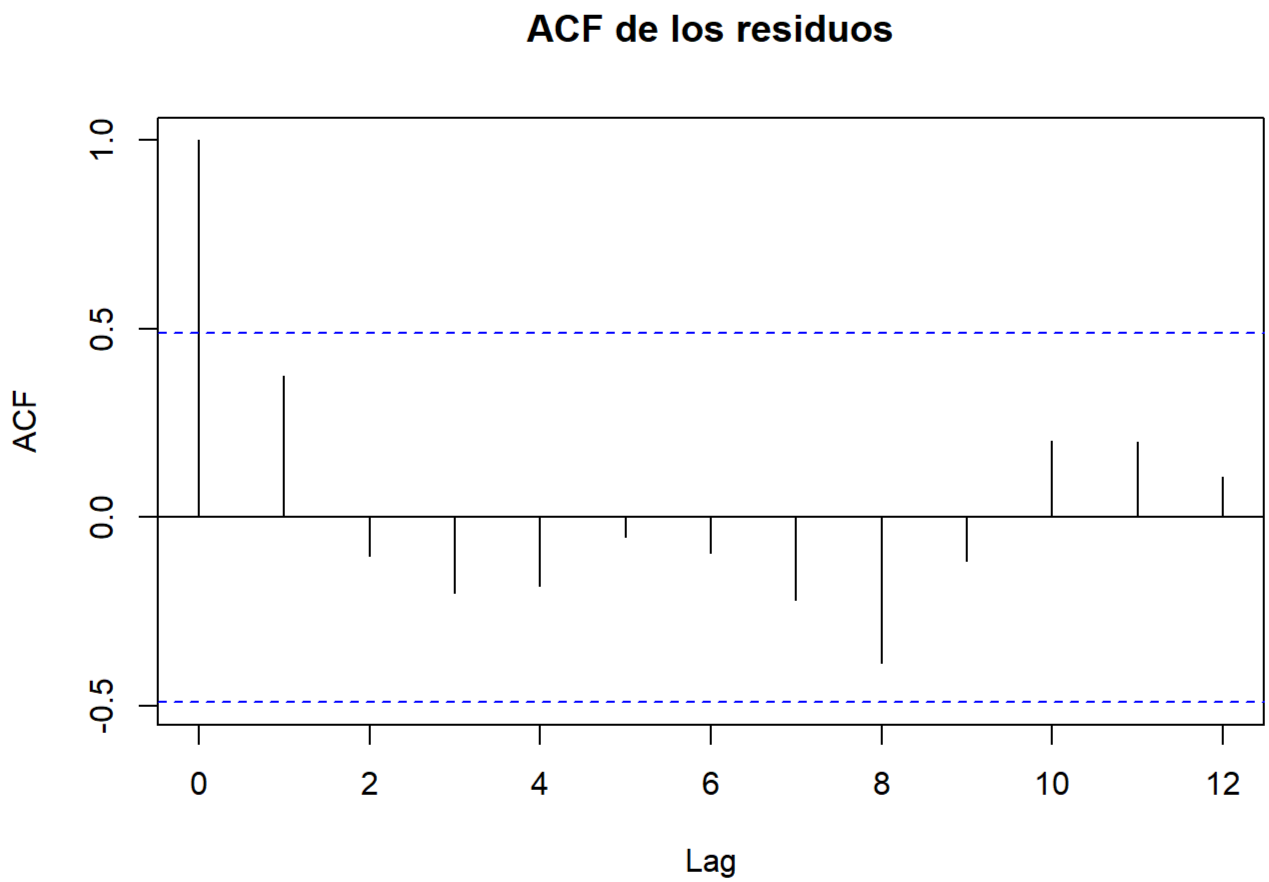
```
# Regresión lineal

modelo_lineal <- lm(ventas_desestacionalizadas ~ ventas_data$Tiempo)
summary(modelo_lineal)
```

```
##
## Call:
## lm(formula = ventas_desestacionalizadas ~ ventas_data$Tiempo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2992 -0.1486 -0.0037  0.1005  0.3698
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.13917    0.10172   50.52  < 2e-16 ***
## ventas_data$Tiempo  0.14613    0.01052   13.89  1.4e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.194 on 14 degrees of freedom
## Multiple R-squared:  0.9324, Adjusted R-squared:  0.9275
## F-statistic:   193 on 1 and 14 DF,  p-value: 1.399e-09
```

```
# Significancia del modelo

# Análisis de residuos

residuos <- residuals(modelo_lineal)
plot(residuos, type = "o", pch = 19, col = "red", main = "Residuos del modelo lineal", xlab = "O
bservaciones", ylab = "Residuos")
abline(h = 0, col = "black", lty = 2)
```




```
acf(residuos, main = "ACF de los residuos")
```



4. Calcula el CME y el EPAM de la predicción de la serie de tiempo

```
# Predicciones de la tendencia lineal

predicciones_lineales <- predict(modelo_lineal)
errores <- ventas_desestacionalizadas - predicciones_lineales

# Calcular CME y EPAM

CME <- mean(errores^2, na.rm = TRUE)
EPAM <- mean(abs(errores / ventas_desestacionalizadas)) * 100

print(paste("CME:", round(CME, 4)))
```

```
## [1] "CME: 0.0329"
```

```
print(paste("EPAM:", round(EPAM, 4), "%"))
```

```
## [1] "EPAM: 2.3413 %"
```

5. Explora un mejor modelo, por ejemplo un modelo cuadrático: $y = \beta_0 + \beta_1 x + \beta_2 x^2$. Para ello transforma la variable ventas (recuerda que la regresión no lineal es una regresión lineal con una tranformación).

```
# Modelo cuadrático

tiempo = 1:length(ventas_desestacionalizadas)

tiempo_cuadrado = tiempo^2

modelo_cuadratico = lm(ventas_desestacionalizadas ~ tiempo + tiempo_cuadrado)

predicciones_cuadraticas = predict(modelo_cuadratico)

# CME y EPAM

CME_cuadratico <- mean((ventas_desestacionalizadas - predicciones_cuadraticas)^2)
EPAM_cuadratico <- mean(abs((ventas_desestacionalizadas - predicciones_cuadraticas) / ventas_des
estacionalizadas) * 100)

print(paste("CME:", round(CME_cuadratico, 4)))
```

```
## [1] "CME: 0.027"
```

```
print(paste("EPAM:", round(EPAM_cuadratico, 4), "%"))
```

```
## [1] "EPAM: 2.2299 %"
```

6. Concluye sobre el mejor modelo

De acuerdo con los resultados anteriores, podemos concluir que el mejor modelo es el modelo cuadrático, ya que presenta un CME menor, con un valor de 0.027 en comparación con el 0.0329 del modelo lineal. Esto indica que el modelo cuadrático tiene una mayor precisión al predecir los datos, ya que un CME menor significa que las predicciones están, en promedio, más cerca de los valores reales. Además, el rendimiento del modelo cuadrático en cuanto al indicador EPAM (2.2299 %) es ligeramente mejor que el del modelo lineal (2.3413 %).

7. Realiza el pronóstico para el siguiente año y grafícalo junto con los pronósticos previos y los datos originales.

```
n_periodos <- length(ventas_desestacionalizadas)
tiempo_futuro <- (n_periodos + 1):(n_periodos + 4)
tiempo_futuro_cuadrado <- tiempo_futuro^2

datos_futuros <- data.frame(tiempo = tiempo_futuro, tiempo_cuadrado = tiempo_futuro_cuadrado)

pronostico_futuro <- predict(modelo_cuadratico, newdata = datos_futuros)

plot(tiempo, ventas_desestacionalizadas, type = "l", col = "blue", ylim = range(c(ventas_desesta
cionalizadas, predicciones_cuadraticas, pronostico_futuro)), xlab = "Tiempo", ylab = "Ventas des
estacionalizadas", main = "Pronóstico de Ventas Desestacionalizadas")
lines(tiempo, predicciones_cuadraticas, col = "red", lty = 2)

legend("topleft", legend = c("Datos Originales", "Pronóstico Cuadrático"),
       col = c("blue", "red"), lty = c(1, 2, 2), bty = "n")
```