

A8-Series de tiempo

Luis Maximiliano López Ramírez

2024-11-12

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

```
# Crear Los vectores de datos
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)

# Crear el DataFrame
tabla <- data.frame(Año, Trimestre, Ventas)

# Mostrar la tabla
tabla
```

##	Año	Trimestre	Ventas
## 1	1	1	4.8
## 2	1	2	4.1
## 3	1	3	6.0
## 4	1	4	6.5
## 5	2	1	5.8
## 6	2	2	5.2
## 7	2	3	6.8
## 8	2	4	7.4
## 9	3	1	6.0
## 10	3	2	5.6
## 11	3	3	7.5
## 12	3	4	7.8
## 13	4	1	6.3
## 14	4	2	5.9
## 15	4	3	8.0
## 16	4	4	8.4

1. Realiza el análisis de tendencia y estacionalidad:

Identifica si es una serie estacionaria

```
# Cargar La Librería
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.3.3
```

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

# Suponiendo que tus datos de ventas están en el vector `Ventas`:
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_ts <- ts(Ventas, frequency = 4) # frequency=4 asume que es
trimestral

# Aplicar el test de Dickey-Fuller aumentado
adf_test <- adf.test(Ventas_ts, k = 0)

# Mostrar los resultados del test
print(adf_test)

##
## Augmented Dickey-Fuller Test
##
## data: Ventas_ts
## Dickey-Fuller = -3.2388, Lag order = 0, p-value = 0.1004
## alternative hypothesis: stationary

# Regla de decisión
if (adf_test$p.value < 0.05) {
  cat("La serie es estacionaria (rechazamos la hipótesis nula de raíz
unitaria).\n")
} else {
  cat("La serie no es estacionaria (no se puede rechazar la hipótesis
nula de raíz unitaria).\n")
}

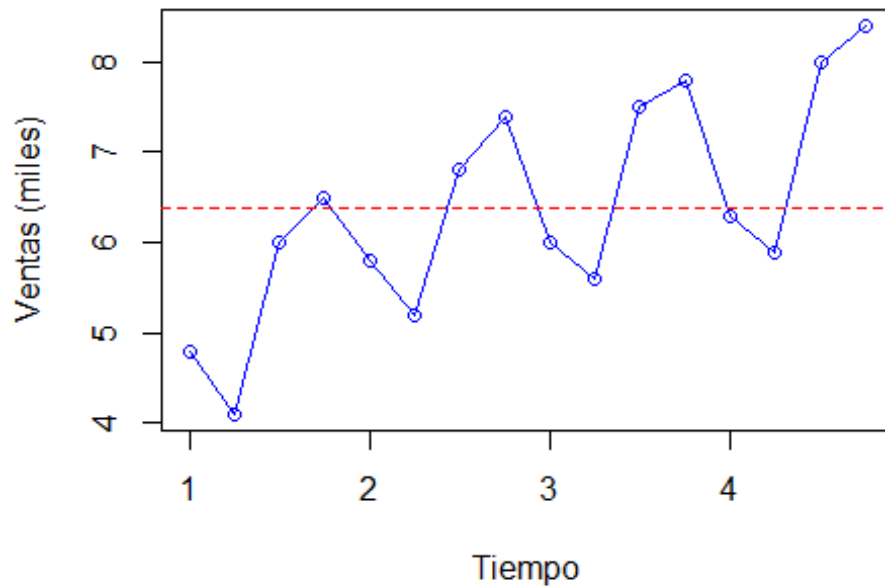
## La serie no es estacionaria (no se puede rechazar la hipótesis nula de
raíz unitaria).
```

Grafica la serie para verificar su tendencia y estacionalidad

```
# Crear Los datos de ventas y convertirlos en una serie temporal
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)

# Graficar la serie temporal original
plot(Ventas_ts, main = "Serie Temporal de Ventas", xlab = "Tiempo", ylab =
"Ventas (miles)", col = "blue", type = "o")
abline(h = mean(Ventas), col = "red", lty = 2) # Línea de media
```

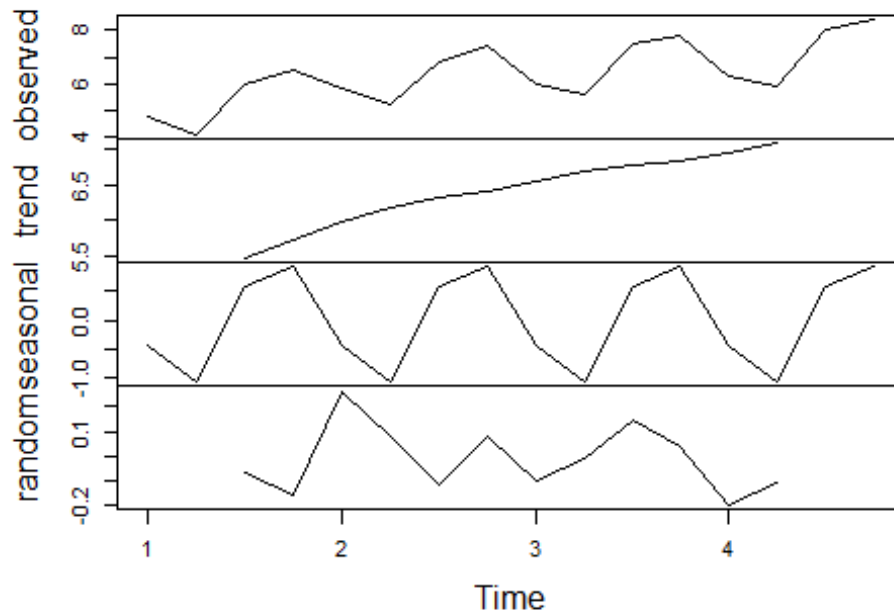
Serie Temporal de Ventas



```
# Descomponer la serie temporal para analizar tendencia y estacionalidad
descomposicion <- decompose(Ventas_ts, type = "additive")

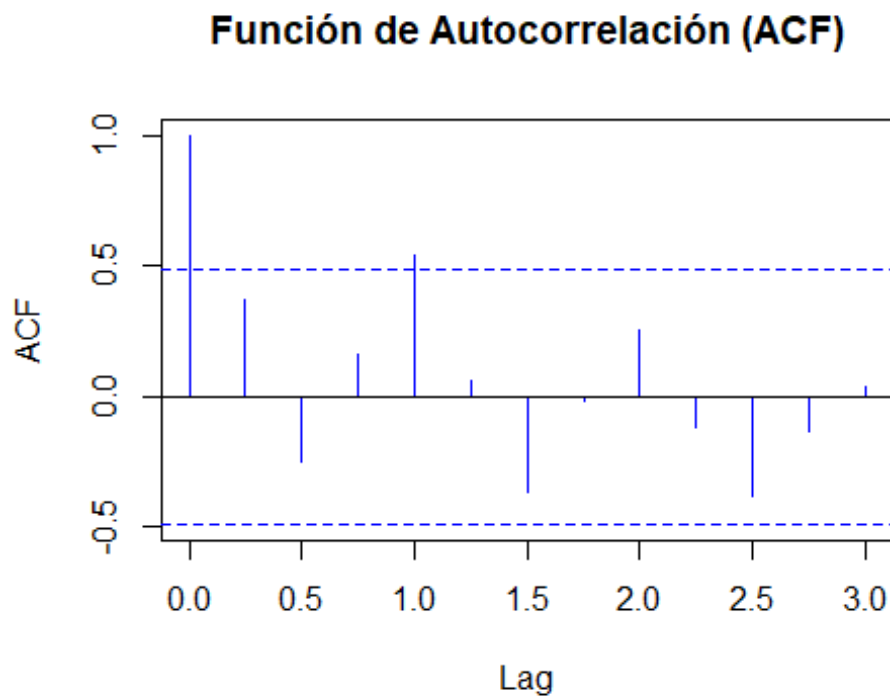
# Graficar la descomposición de la serie temporal
plot(descomposicion)
```

Decomposition of additive time series



Analiza su gráfico de autocorrelación

```
# Graficar ACF y PACF
acf(Ventas_ts, main = "Función de Autocorrelación (ACF)", col = "blue")
# Gráfico ACF
```

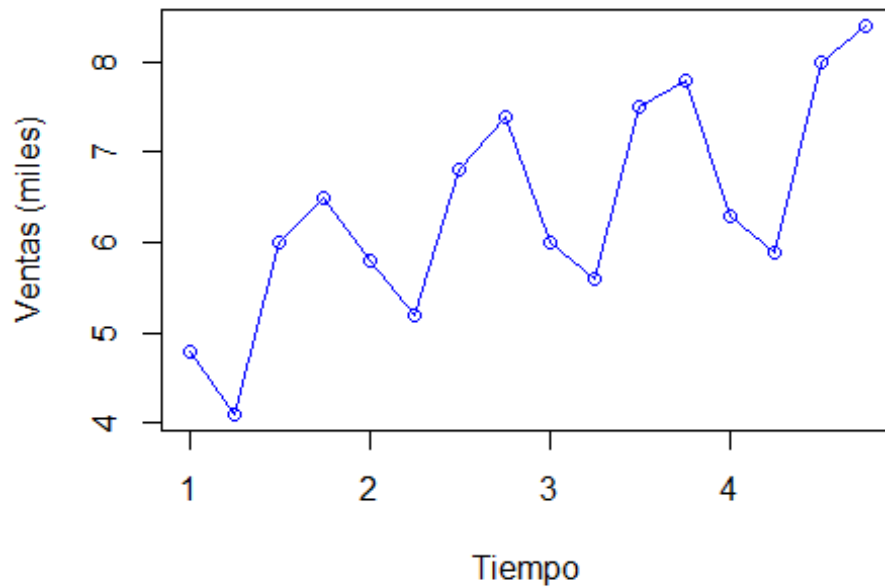


La gráfica de autocorrelación (ACF) muestra una autocorrelación significativa en el primer retraso (lag 1), seguida de una rápida disminución hacia cero en los lags posteriores, lo que indica que la serie temporal tiene características estacionarias. La ausencia de una autocorrelación persistente sugiere que no hay una tendencia fuerte ni estacionalidad de largo plazo, lo cual respalda la adecuación de un modelo estacionario para esta serie.

Identifica si el modelo puede ser sumativo o multiplicativo (puedes probar con ambos para ver con cuál es mejor el modelo)

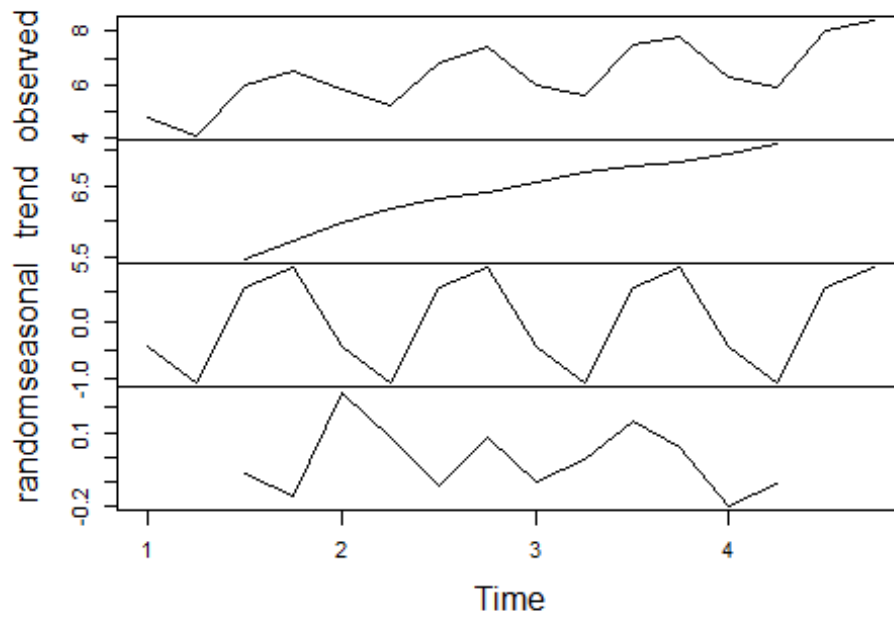
```
# Graficar la serie temporal
plot(Ventas_ts, main = "Serie Temporal de Ventas", xlab = "Tiempo", ylab = "Ventas (miles)", col = "blue", type = "o")
```

Serie Temporal de Ventas



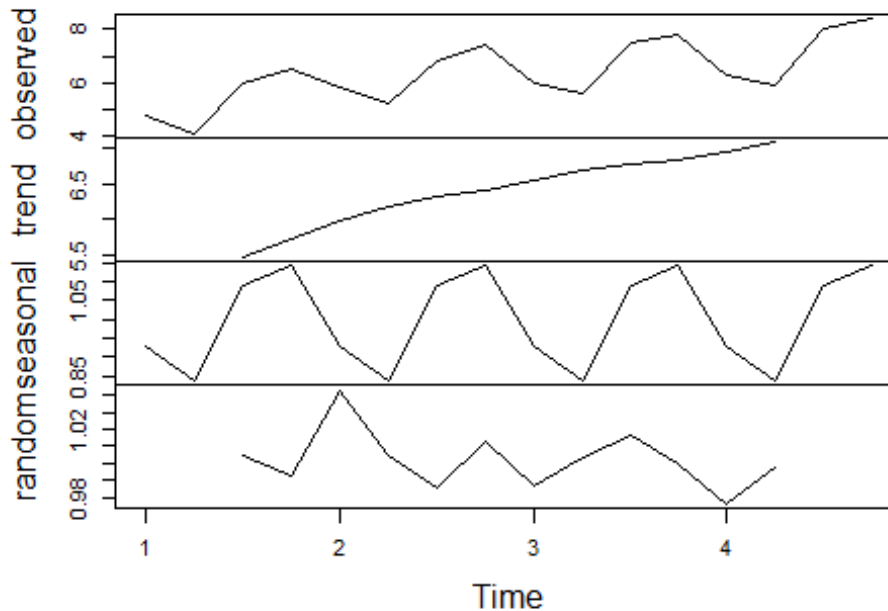
```
# Descomposición aditiva
decomp_aditiva <- decompose(Ventas_ts, type = "additive")
plot(decomp_aditiva) # Sin argumento `main`
```

Decomposition of additive time series



```
# Descomposición multiplicativa
decomp_multiplicativa <- decompose(Ventas_ts, type = "multiplicative")
plot(decomp_multiplicativa) # Sin argumento `main`
```

Decomposition of multiplicative time series



```
# Calcular la varianza de la componente aleatoria de ambos modelos
varianza_aditiva <- var(na.omit(decomp_aditiva$random))
varianza_multiplicativa <- var(na.omit(decomp_multiplicativa$random))

# Mostrar los resultados
cat("Varianza del modelo aditivo:", varianza_aditiva, "\n")

## Varianza del modelo aditivo: 0.01886364

cat("Varianza del modelo multiplicativo:", varianza_multiplicativa, "\n")

## Varianza del modelo multiplicativo: 0.000305848

# Comparación
if (varianza_aditiva < varianza_multiplicativa) {
  cat("El modelo aditivo tiene menor varianza en la componente aleatoria
y es un mejor ajuste.\n")
} else if (varianza_multiplicativa < varianza_aditiva) {
  cat("El modelo multiplicativo tiene menor varianza en la componente
aleatoria y es un mejor ajuste.\n")
} else {
  cat("Ambos modelos tienen varianzas similares en la componente
aleatoria.\n")
}
```

```
## El modelo multiplicativo tiene menor varianza en la componente aleatoria y es un mejor ajuste.
```

La comparación de varianzas entre los modelos aditivo y multiplicativo muestra que la varianza de la componente aleatoria en el modelo multiplicativo (0.000305848) es significativamente menor que en el modelo aditivo (0.01886364). Esto sugiere que el modelo multiplicativo ajusta mejor la serie temporal, ya que presenta menos variabilidad en los errores. Por lo tanto, el modelo multiplicativo es el más adecuado para capturar la estructura de los datos en esta serie temporal.

2. Calcula los índices estacionales y grafica la serie desestacionalizada

```
# Descomposición aditiva
decomp_aditiva <- decompose(Ventas_ts, type = "additive")
indices_estacionales_aditivos <- decomp_aditiva$seasonal
serie_desestacionalizada_aditiva <- Ventas_ts -
indices_estacionales_aditivos

# Descomposición multiplicativa
decomp_multiplicativa <- decompose(Ventas_ts, type = "multiplicative")
indices_estacionales_multiplicativos <- decomp_multiplicativa$seasonal
serie_desestacionalizada_multiplicativa <- Ventas_ts /
indices_estacionales_multiplicativos

# Imprimir Los índices estacionales para ambos modelos
cat("Índices estacionales (modelo aditivo):\n")

## Índices estacionales (modelo aditivo):

print(indices_estacionales_aditivos)

##           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

cat("\nÍndices estacionales (modelo multiplicativo):\n")

##
## Índices estacionales (modelo multiplicativo):

print(indices_estacionales_multiplicativos)

##           Qtr1           Qtr2           Qtr3           Qtr4
## 1 0.9306617 0.8363763 1.0915441 1.1414179
## 2 0.9306617 0.8363763 1.0915441 1.1414179
## 3 0.9306617 0.8363763 1.0915441 1.1414179
## 4 0.9306617 0.8363763 1.0915441 1.1414179

# Graficar la serie original y las series desestacionalizadas para ambos modelos
```

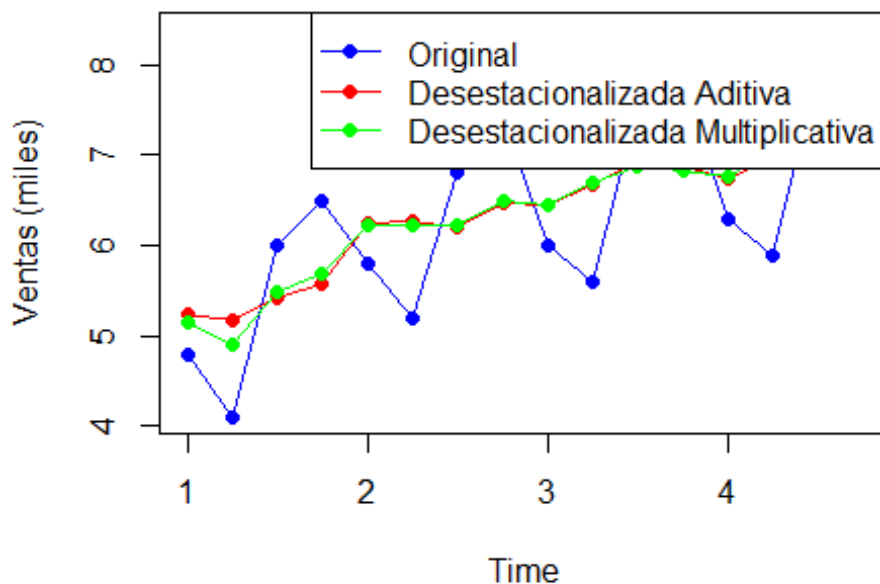


```

plot(Ventas_ts, main = "Serie Temporal Original y Desestacionalizada
(Modelos Aditivo y Multiplicativo)", ylab = "Ventas (miles)", col =
"blue", type = "o", pch = 16)
lines(serie_desestacionalizada_aditiva, col = "red", type = "o", pch =
16)
lines(serie_desestacionalizada_multiplicativa, col = "green", type = "o",
pch = 16)
legend("topright", legend = c("Original", "Desestacionalizada Aditiva",
"Desestacionalizada Multiplicativa"), col = c("blue", "red", "green"),
lty = 1, pch = 16)

```

oral Original y Desestacionalizada (Modelos Aditivo



3. Analiza el modelo lineal de la tendencia

Realiza la regresión lineal de la tendencia (ventas desestacionalizadas vs tiempo)

```

# Crear un vector de tiempo
tiempo <- 1:length(Ventas_ts)

# Regresión lineal de la tendencia para el modelo aditivo
modelo_tendencia_aditivo <- lm(serie_desestacionalizada_aditiva ~ tiempo)
summary(modelo_tendencia_aditivo)

##
## Call:
## lm(formula = serie_desestacionalizada_aditiva ~ 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 ***
## 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

# Regresión lineal de la tendencia para el modelo multiplicativo
modelo_tendencia_multiplicativo <-
lm(serie_desestacionalizada_multiplicativa ~ tiempo)
summary(modelo_tendencia_multiplicativo)

##
## Call:
## lm(formula = serie_desestacionalizada_multiplicativa ~ tiempo)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -0.5007 -0.1001  0.0037  0.1207  0.3872
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.10804    0.11171   45.73 < 2e-16 ***
## tiempo      0.14738    0.01155   12.76 4.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.213 on 14 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9151
## F-statistic: 162.7 on 1 and 14 DF, p-value: 4.248e-09

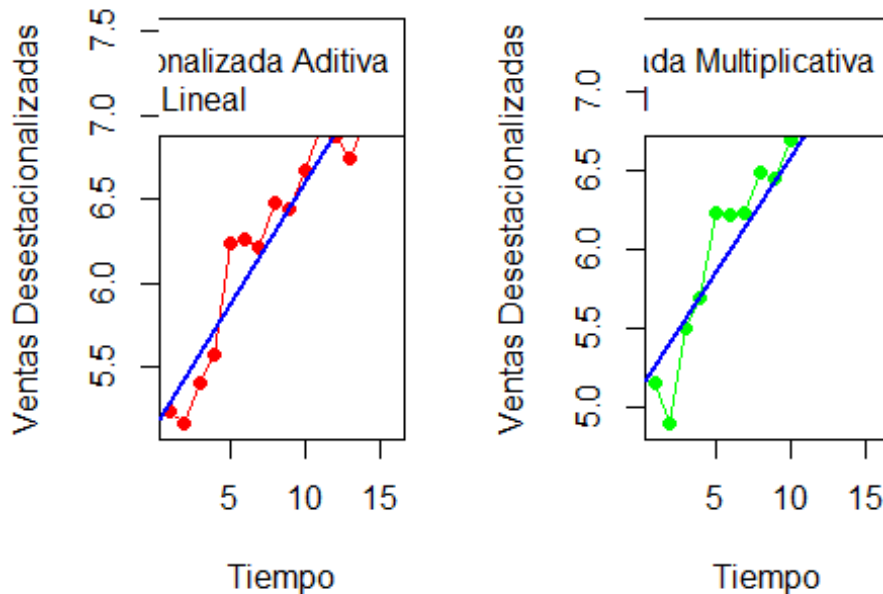
# Graficar las series desestacionalizadas y sus líneas de tendencia
par(mfrow = c(1, 2)) # Dividir la ventana gráfica en 1 fila y 2 columnas

# Gráfico para el modelo aditivo
plot(tiempo, serie_desestacionalizada_aditiva, main = "Tendencia (Modelo
Aditivo)", ylab = "Ventas Desestacionalizadas", xlab = "Tiempo", col =
"red", type = "o", pch = 16)
abline(modelo_tendencia_aditivo, col = "blue", lwd = 2)
legend("topright", legend = c("Desestacionalizada Aditiva", "Tendencia
Lineal"), col = c("red", "blue"), lty = 1, pch = 16)

# Gráfico para el modelo multiplicativo
```

```
plot(tiempo, serie_desestacionalizada_multiplicativa, main = "Tendencia
(Modelo Multiplicativo)", ylab = "Ventas Desestacionalizadas", xlab =
"Tiempo", col = "green", type = "o", pch = 16)
abline(modelo_tendencia_multiplicativo, col = "blue", lwd = 2)
legend("topright", legend = c("Desestacionalizada Multiplicativa",
"Tendencia Lineal"), col = c("green", "blue"), lty = 1, pch = 16)
```

Tendencia (Modelo Aditivo) Tendencia (Modelo Multiplicativo)



```
# Restaurar la configuración gráfica
par(mfrow = c(1, 1))
```

Analiza la significancia del modelo lineal, global e individual

```
# Análisis de significancia global e individual para el modelo aditivo
cat("Análisis del modelo aditivo:\n")
```

```
## Análisis del modelo aditivo:
```

```
resumen_aditivo <- summary(modelo_tendencia_aditivo)
print(resumen_aditivo)
```

```
##
## Call:
## lm(formula = serie_desestacionalizada_aditiva ~ 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 ***
## 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

# Extraer valores p individuales y globales para el modelo aditivo
cat("\nSignificancia global (modelo aditivo):\n")

##
## Significancia global (modelo aditivo):

cat("Valor p del modelo aditivo:", pf(resumen_aditivo$fstatistic[1],
resumen_aditivo$fstatistic[2], resumen_aditivo$fstatistic[3], lower.tail
= FALSE), "\n")

## Valor p del modelo aditivo: 1.399111e-09

cat("\nSignificancia individual (modelo aditivo):\n")

##
## Significancia individual (modelo aditivo):

print(resumen_aditivo$coefficients[,4])

## (Intercept)      tiempo
## 3.015451e-17 1.399111e-09

# Análisis de significancia global e individual para el modelo
multiplicativo
cat("\nAnálisis del modelo multiplicativo:\n")

##
## Análisis del modelo multiplicativo:

resumen_multiplicativo <- summary(modelo_tendencia_multiplicativo)
print(resumen_multiplicativo)

##
## Call:
## lm(formula = serie_desestacionalizada_multiplicativa ~ tiempo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.5007 -0.1001  0.0037  0.1207  0.3872
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)

```

```
## (Intercept)  5.10804    0.11171   45.73 < 2e-16 ***
## tiempo      0.14738    0.01155   12.76 4.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.213 on 14 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9151
## F-statistic: 162.7 on 1 and 14 DF,  p-value: 4.248e-09

# Extraer valores p individuales y globales para el modelo multiplicativo
cat("\nSignificancia global (modelo multiplicativo):\n")

##
## Significancia global (modelo multiplicativo):

cat("Valor p del modelo multiplicativo:",
pf(resumen_multiplicativo$fstatistic[1],
resumen_multiplicativo$fstatistic[2],
resumen_multiplicativo$fstatistic[3], lower.tail = FALSE), "\n")

## Valor p del modelo multiplicativo: 4.247717e-09

cat("\nSignificancia individual (modelo multiplicativo):\n")

##
## Significancia individual (modelo multiplicativo):

print(resumen_multiplicativo$coefficients[,4])

## (Intercept)      tiempo
## 1.209866e-16  4.247717e-09
```

Ambos modelos, aditivo y multiplicativo, son estadísticamente significativos tanto a nivel global como individual. En el modelo aditivo, el valor p global es 1.4×10^{-9} , mientras que los valores p de los coeficientes son 3.0×10^{-17} para el intercepto y 1.4×10^{-9} para la pendiente, indicando que todos son significativos. En el modelo multiplicativo, el valor p global es 4.25×10^{-9} , con valores p de 1.2×10^{-16} para el intercepto y 4.25×10^{-9} para la pendiente, también significativos. El R^2 ajustado es de 0.9275 en el modelo aditivo (con error estándar residual de 0.194) y de 0.9151 en el modelo multiplicativo (con error estándar residual de 0.213). **Esto sugiere que el modelo aditivo es un mejor ajuste para los datos.**

Haz el análisis de residuos

```
# Análisis de residuos para el modelo multiplicativo
cat("\nAnálisis de residuos para el modelo multiplicativo:\n")

##
## Análisis de residuos para el modelo multiplicativo:

par(mfrow = c(2, 2)) # Configuración para mostrar varios gráficos
```

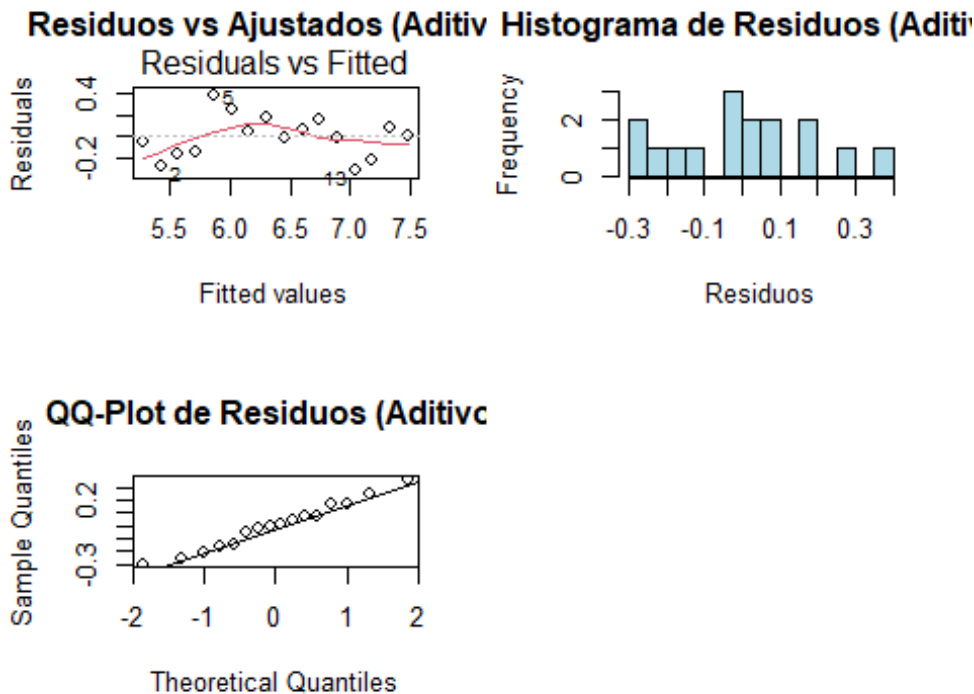
```

# Gráfico de residuos
plot(modelo_tendencia_aditivo, which = 1, main = "Residuos vs Ajustados
(Aditivo)")
# Histograma de Los residuos
hist(resid(modelo_tendencia_aditivo), main = "Histograma de Residuos
(Aditivo)", xlab = "Residuos", col = "lightblue", breaks = 10)
# QQ-plot de Los residuos
qqnorm(resid(modelo_tendencia_aditivo), main = "QQ-Plot de Residuos
(Aditivo)")
qqline(resid(modelo_tendencia_aditivo))
# Prueba de Shapiro-Wilk para normalidad
shapiro_test_aditivo <- shapiro.test(resid(modelo_tendencia_aditivo))
print(shapiro_test_aditivo)

##
##  Shapiro-Wilk normality test
##
## data:  resid(modelo_tendencia_aditivo)
## W = 0.97816, p-value = 0.9473

# Restaurar la configuración de gráficos
par(mfrow = c(1, 1))

```



```

# Análisis de residuos para el modelo multiplicativo
cat("\nAnálisis de residuos para el modelo multiplicativo:\n")

##
## Análisis de residuos para el modelo multiplicativo:

```

```

par(mfrow = c(2, 2)) # Configuración para mostrar varios gráficos

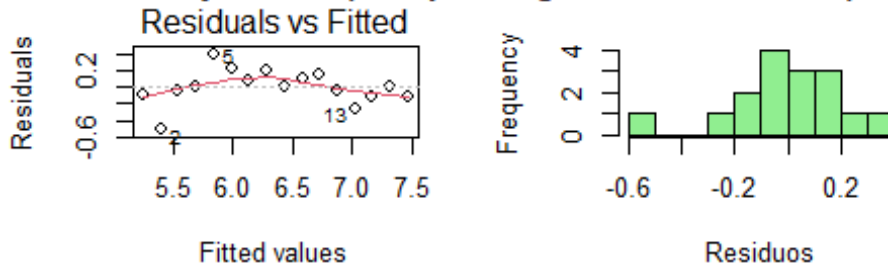
# Gráfico de residuos
plot(modelo_tendencia_multiplicativo, which = 1, main = "Residuos vs
Ajustados (Multiplicativo)")
# Histograma de Los residuos
hist(resid(modelo_tendencia_multiplicativo), main = "Histograma de
Residuos (Multiplicativo)", xlab = "Residuos", col = "lightgreen", breaks
= 10)
# QQ-plot de Los residuos
qqnorm(resid(modelo_tendencia_multiplicativo), main = "QQ-Plot de
Residuos (Multiplicativo)")
qqline(resid(modelo_tendencia_multiplicativo))
# Prueba de Shapiro-Wilk para normalidad
shapiro_test_multiplicativo <-
shapiro.test(resid(modelo_tendencia_multiplicativo))
print(shapiro_test_multiplicativo)

##
## Shapiro-Wilk normality test
##
## data: resid(modelo_tendencia_multiplicativo)
## W = 0.96379, p-value = 0.7307

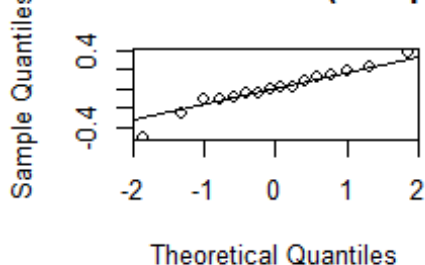
# Restaurar la configuración de gráficos
par(mfrow = c(1, 1))

```

Residuos vs Ajustados (Multiplicativo) **Histograma de Residuos (Multiplicativo)**



QQ-Plot de Residuos (Multiplicativo)



Los resultados del análisis de residuos indican que ambos modelos, aditivo y multiplicativo, cumplen con el supuesto de normalidad en los residuos. Para el modelo aditivo, el test de Shapiro-Wilk arroja un valor W de 0.97816 y un valor p de 0.9473, mientras que para el modelo multiplicativo, W es 0.96379 con un valor p de 0.7307. Dado que ambos valores p son significativamente mayores a 0.05, no se rechaza la hipótesis de normalidad en ninguno de los modelos, lo que sugiere que los residuos están distribuidos normalmente en ambos casos. Esto respalda la validez de los modelos al no presentar problemas significativos de no-normalidad en los residuos.

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

```
# Predicciones de la tendencia para el modelo aditivo y multiplicativo
pred_tendencia_aditivo <- predict(modelo_tendencia_aditivo, newdata =
data.frame(tiempo = tiempo))
pred_tendencia_multiplicativo <- predict(modelo_tendencia_multiplicativo,
newdata = data.frame(tiempo = tiempo))

# Reestacionalizar las predicciones para obtener la serie en su forma
original
# Para el modelo aditivo (suma los índices estacionales)
predicciones_aditivo <- pred_tendencia_aditivo + decomp_aditiva$seasonal

# Para el modelo multiplicativo (multiplica por los índices estacionales)
predicciones_multiplicativo <- pred_tendencia_multiplicativo *
decomp_multiplicativa$seasonal

# Mostrar las predicciones
cat("Predicciones del modelo aditivo:\n")

## Predicciones del modelo aditivo:

print(predicciones_aditivo)

##          Qtr1      Qtr2      Qtr3      Qtr4
## 1 4.845711 4.362672 6.167132 6.642426
## 2 5.430221 4.947181 6.751642 7.226936
## 3 6.014730 5.531691 7.336152 7.811446
## 4 6.599240 6.116201 7.920662 8.395956

cat("\nPredicciones del modelo multiplicativo:\n")

##
## Predicciones del modelo multiplicativo:

print(predicciones_multiplicativo)

##          Qtr1      Qtr2      Qtr3      Qtr4
## 1 4.891022 4.518779 6.058275 6.503308
## 2 5.439673 5.011846 6.701770 7.176206
## 3 5.988324 5.504913 7.345266 7.849104
## 4 6.536975 5.997981 7.988762 8.522002
```



```

# Calcular CME y EPAM para cada modelo
# CME y EPAM para el modelo aditivo
CME_aditivo <- mean((Ventas - predicciones_aditivo)^2)
EPAM_aditivo <- mean(abs((Ventas - predicciones_aditivo) / Ventas)) * 100

# CME y EPAM para el modelo multiplicativo
CME_multiplicativo <- mean((Ventas - predicciones_multiplicativo)^2)
EPAM_multiplicativo <- mean(abs((Ventas - predicciones_multiplicativo) /
Ventas)) * 100

# Mostrar resultados de error
cat("\nCuadrado Medio del Error (CME) - Aditivo:", CME_aditivo, "\n")

##
## Cuadrado Medio del Error (CME) - Aditivo: 0.03291917

cat("Error Porcentual Absoluto Medio (EPAM) - Aditivo:", EPAM_aditivo,
"%\n")

## Error Porcentual Absoluto Medio (EPAM) - Aditivo: 2.492108 %

cat("\nCuadrado Medio del Error (CME) - Multiplicativo:",
CME_multiplicativo, "\n")

##
## Cuadrado Medio del Error (CME) - Multiplicativo: 0.0330211

cat("Error Porcentual Absoluto Medio (EPAM) - Multiplicativo:",
EPAM_multiplicativo, "%\n")

## Error Porcentual Absoluto Medio (EPAM) - Multiplicativo: 2.439533 %

```

5. Explora un mejor modelo, por ejemplo un modelo cuadrático. Para ello transforma la variable ventas (recuerda que la regresión no lineal es una regresión lineal con una transformación).

```

# Crear un vector de tiempo y su cuadrado
tiempo <- 1:length(Ventas_ts)
tiempo_cuadrado <- tiempo^2

# Ajustar el modelo cuadrático para la tendencia en el modelo aditivo
modelo_cuadratico_aditivo <- lm(serie_desestacionalizada_aditiva ~ tiempo
+ tiempo_cuadrado)

# Ajustar el modelo cuadrático para la tendencia en el modelo
multiplicativo
modelo_cuadratico_multiplicativo <-
lm(serie_desestacionalizada_multiplicativa ~ tiempo + tiempo_cuadrado)

# Resumen de Los modelos
cat("Modelo cuadrático para la serie desestacionalizada aditiva:\n")

```

```
## Modelo cuadrático para la serie desestacionalizada aditiva:

summary(modelo_cuadratico_aditivo)

##
## Call:
## lm(formula = serie_desestacionalizada_aditiva ~ tiempo +
tiempo_cuadrado)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30333 -0.13440 -0.01928  0.11368  0.33301
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.930833   0.155679  31.673 1.08e-13 ***
## tiempo        0.215572   0.042149   5.115 0.000199 ***
## tiempo_cuadrado -0.004085   0.002410  -1.695 0.113918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1822 on 13 degrees of freedom
## Multiple R-squared:  0.9446, Adjusted R-squared:  0.9361
## F-statistic: 110.8 on 2 and 13 DF,  p-value: 6.805e-09

cat("\nModelo cuadrático para la serie desestacionalizada
multiplicativa:\n")

##
## Modelo cuadrático para la serie desestacionalizada multiplicativa:

summary(modelo_cuadratico_multiplicativo)

##
## Call:
## lm(formula = serie_desestacionalizada_multiplicativa ~ tiempo +
tiempo_cuadrado)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36986 -0.07058 -0.00100  0.11345  0.33110
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.790283   0.152429  31.426 1.20e-13 ***
## tiempo        0.253302   0.041269   6.138 3.56e-05 ***
## tiempo_cuadrado -0.006231   0.002360  -2.640  0.0204 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1784 on 13 degrees of freedom
```

```

## Multiple R-squared:  0.9484, Adjusted R-squared:  0.9405
## F-statistic: 119.6 on 2 and 13 DF,  p-value: 4.268e-09

# Predicciones usando el modelo cuadrático
pred_tendencia_cuadratica_aditivo <- predict(modelo_cuadratico_aditivo,
newdata = data.frame(tiempo = tiempo, tiempo_cuadrado = tiempo_cuadrado))
pred_tendencia_cuadratica_multiplicativo <-
predict(modelo_cuadratico_multiplicativo, newdata = data.frame(tiempo =
tiempo, tiempo_cuadrado = tiempo_cuadrado))

# Reestacionalizar Las predicciones para obtener la serie en su forma
original
predicciones_cuadraticas_aditivo <- pred_tendencia_cuadratica_aditivo +
decomp_aditiva$seasonal
predicciones_cuadraticas_multiplicativo <-
pred_tendencia_cuadratica_multiplicativo * decomp_multiplicativa$seasonal

# Calcular CME y EPAM para el modelo cuadrático en ambos casos
CME_cuadratico_aditivo <- mean((Ventas -
predicciones_cuadraticas_aditivo)^2)
EPAM_cuadratico_aditivo <- mean(abs((Ventas -
predicciones_cuadraticas_aditivo) / Ventas)) * 100

CME_cuadratico_multiplicativo <- mean((Ventas -
predicciones_cuadraticas_multiplicativo)^2)
EPAM_cuadratico_multiplicativo <- mean(abs((Ventas -
predicciones_cuadraticas_multiplicativo) / Ventas)) * 100

# Mostrar Los resultados de error para el modelo cuadrático
cat("\nCuadrado Medio del Error (CME) - Cuadrático Aditivo:",
CME_cuadratico_aditivo, "\n")

##
## Cuadrado Medio del Error (CME) - Cuadrático Aditivo: 0.02696192

cat("Error Porcentual Absoluto Medio (EPAM) - Cuadrático Aditivo:",
EPAM_cuadratico_aditivo, "%\n")

## Error Porcentual Absoluto Medio (EPAM) - Cuadrático Aditivo: 2.331231
%

cat("\nCuadrado Medio del Error (CME) - Cuadrático Multiplicativo:",
CME_cuadratico_multiplicativo, "\n")

##
## Cuadrado Medio del Error (CME) - Cuadrático Multiplicativo: 0.02193668

cat("Error Porcentual Absoluto Medio (EPAM) - Cuadrático
Multiplicativo:", EPAM_cuadratico_multiplicativo, "%\n")

```

```
## Error Porcentual Absoluto Medio (EPAM) - Cuadrático Multiplicativo:  
1.949349 %
```

6. Concluye sobre el mejor modelo

Comparando los resultados de los modelos lineal y cuadrático, observamos que el modelo cuadrático ofrece un mejor ajuste, especialmente en su versión multiplicativa. El modelo cuadrático multiplicativo tiene un Cuadrado Medio del Error (CME) de 0.02193668 y un Error Porcentual Absoluto Medio (EPAM) de 1.949349 %, ambos valores más bajos en comparación con el modelo lineal multiplicativo, que presenta un CME de 0.0330211 y un EPAM de 2.439533 %. Esto sugiere que el modelo cuadrático captura mejor las variaciones en la serie temporal desestacionalizada.

En términos de significancia, los términos lineales y cuadráticos del modelo cuadrático multiplicativo son significativos, con un valor p de 0.0204 para el término cuadrático, lo que respalda la inclusión de este término para mejorar el ajuste. El modelo cuadrático aditivo también reduce los errores (CME de 0.02696192 y EPAM de 2.331231 %) en comparación con su versión lineal, aunque el término cuadrático en este caso no es significativo ($p = 0.1139$).

En conclusión, el modelo cuadrático multiplicativo es el mejor ajuste para esta serie, ya que logra reducir significativamente los errores de predicción y muestra una significancia estadística en todos sus términos. Este modelo es más adecuado para capturar la tendencia en los datos, proporcionando predicciones más precisas en comparación con los modelos lineales.

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

```
# Datos de ventas  
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_ts <- ts(Ventas, frequency = 4) # Suponiendo datos trimestrales  
  
# Descomposición multiplicativa  
decomp_multiplicativa <- decompose(Ventas_ts, type = "multiplicative")  
serie_desestacionalizada_multiplicativa <- Ventas_ts /  
decomp_multiplicativa$seasonal  
  
# Crear variables tiempo y tiempo cuadrado  
tiempo <- 1:length(Ventas_ts)  
tiempo_cuadrado <- tiempo^2  
  
# Ajustar el modelo cuadrático multiplicativo  
modelo_cuadratico_multiplicativo <-  
lm(serie_desestacionalizada_multiplicativa ~ tiempo + tiempo_cuadrado)  
  
# Extender las variables de tiempo para el próximo año (4 trimestres  
adicionales)
```

```

tiempo_extendido <- c(tiempo, (length(tiempo) + 1):(length(tiempo) + 4))
tiempo_cuadrado_extendido <- tiempo_extendido^2

# Realizar el pronóstico de la tendencia para el siguiente año
pred_tendencia_cuadratica_extendida <- predict(
  modelo_cuadratico_multiplicativo,
  newdata = data.frame(tiempo = tiempo_extendido, tiempo_cuadrado =
tiempo_cuadrado_extendido)
)

# Repetir los índices estacionales para el próximo año
indices_estacionales_extendidos <-
rep(decomp_multiplicativa$seasonal[1:4], length.out =
length(tiempo_extendido))

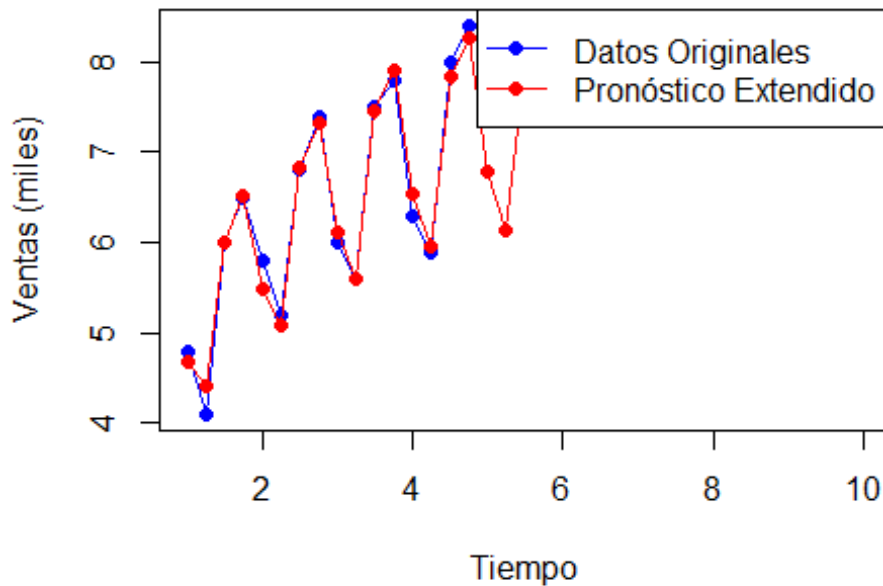
# Aplicar los índices estacionales para obtener el pronóstico en su forma
original
predicciones_extendidas <- pred_tendencia_cuadratica_extendida *
indices_estacionales_extendidos

# Crear un objeto de serie temporal para el pronóstico extendido
Ventas_pronosticadas <- ts(predicciones_extendidas, start =
start(Ventas_ts), frequency = 4)

# Graficar la serie original y el pronóstico del mejor modelo (modelo
cuadrático multiplicativo)
plot(Ventas_ts, main = "Pronóstico de Ventas para el Próximo Año", ylab =
"Ventas (miles)", xlab = "Tiempo",
col = "blue", type = "o", pch = 16, xlim = c(1, 10)) # Limitar el
eje x hasta 10
lines(Ventas_pronosticadas, col = "red", type = "o", pch = 16)
legend("topright", legend = c("Datos Originales", "Pronóstico
Extendido"), col = c("blue", "red"), lty = 1, pch = 16)

```

Pronóstico de Ventas para el Próximo Año



Al observar la gráfica, se puede concluir que el modelo cuadrático multiplicativo se ajusta bien a los datos históricos de ventas y proyecta un pronóstico razonable para el próximo año. Las predicciones (en rojo) siguen de cerca el comportamiento de la serie original (en azul), manteniendo el patrón estacional. Este ajuste indica que el modelo captura adecuadamente tanto la tendencia como la estacionalidad de la serie, proporcionando una proyección que refleja el comportamiento cíclico observado. Este resultado sugiere que el modelo cuadrático multiplicativo es adecuado para realizar pronósticos a corto plazo para esta serie temporal.