

Actividad-20-A8

Saúl Francisco Vázquez del Río

2024-11-12

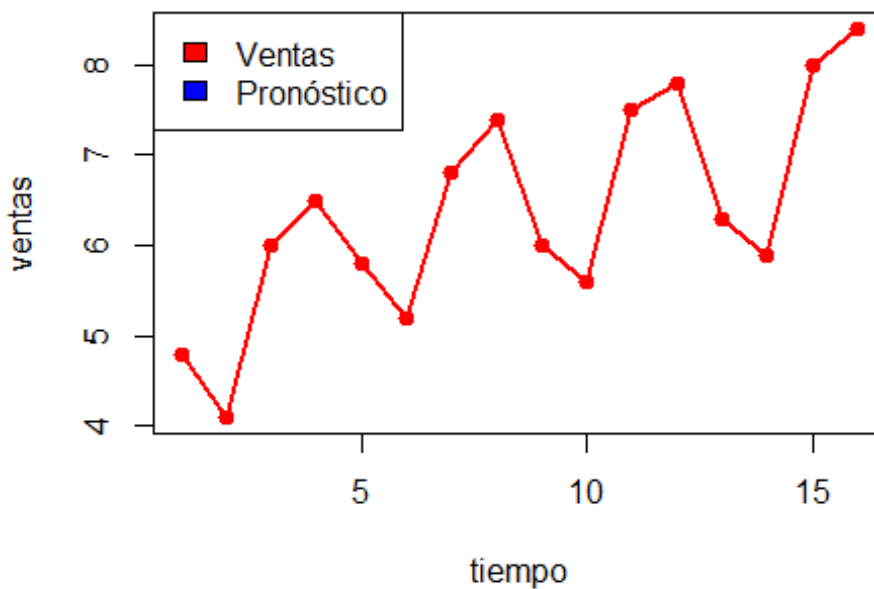
Realiza el análisis de tendencia y estacionalidad:

Identifica si es una serie estacionaria

```
# Definición de Los datos de ventas
ventas <- c(4.8, 4.1, 6, 6.5, 5.8, 5.2, 6.8, 7.4, 6, 5.6, 7.5, 7.8, 6.3,
5.9, 8, 8.4)

# Crear la serie temporal
x <- ts(ventas, frequency = 4, start = c(2016, 1))

# Graficar la serie temporal
tiempo <- 1:16
plot(tiempo, ventas, col = "red", type = "o", lwd = 2, pch = 19)
legend("topleft", legend = c("Ventas", "Pronóstico"), fill = c("red",
"blue"))
```



Grafica la serie para verificar su tendencia y estacionalidad Analiza su gráfico de autocorrelación

```

# Análisis de autocorrelación
library(forecast)

## Warning: package 'forecast' was built under R version 4.4.2

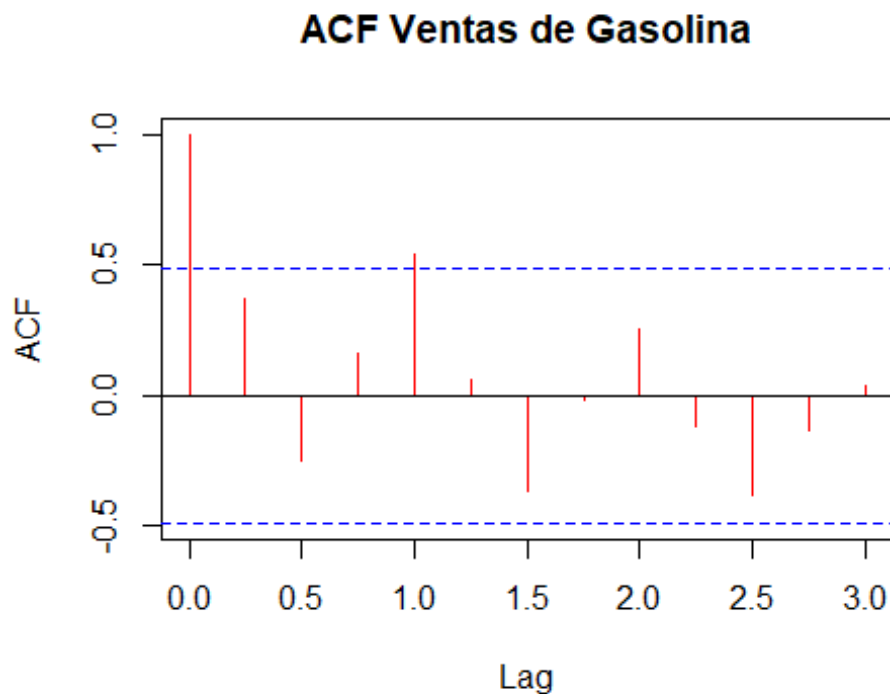
## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo

auto.arima(ventas)

## Series: ventas
## ARIMA(0,1,0)
##
## sigma^2 = 1.395: log likelihood = -23.78
## AIC=49.56   AICc=49.87   BIC=50.27

# Graficar la función de autocorrelación
acf(x, col = "red", main = "ACF Ventas de Gasolina")

```



```

qnorm(1 - 0.05 / 2) / sqrt(length(ventas))

## [1] 0.489991

```

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

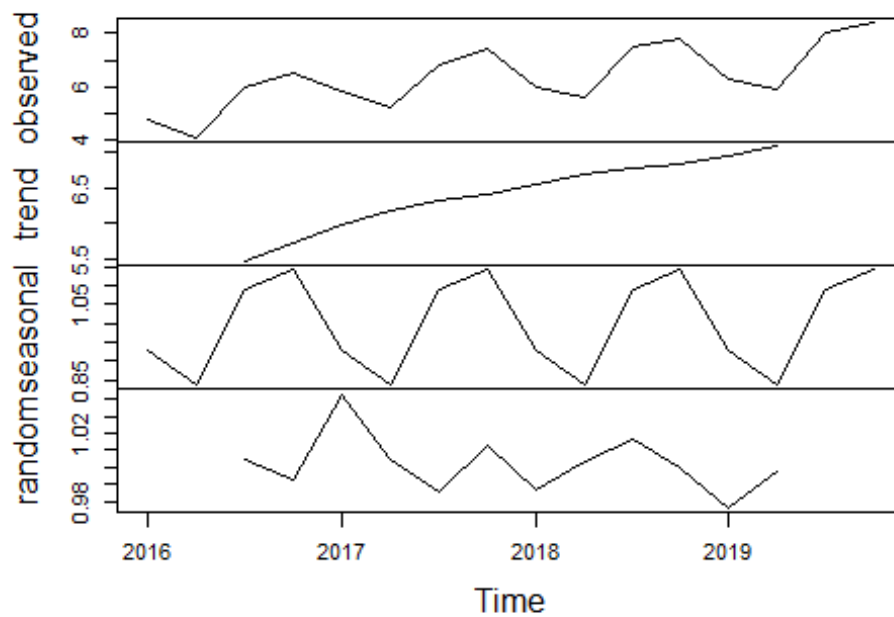
```

# Identificación de modelos aditivo y multiplicativo
# Descomposición multiplicativa

```

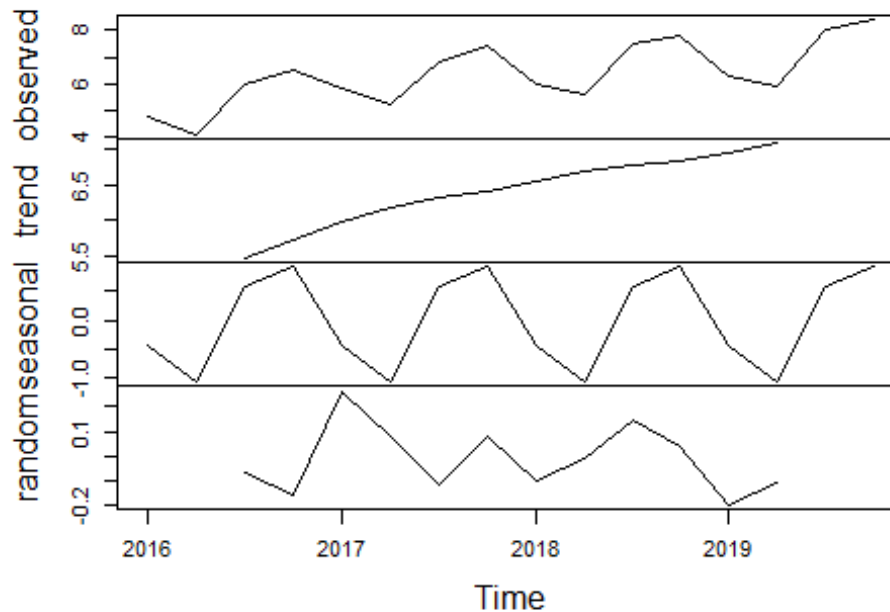
```
descomp_M <- decompose(x, type = "multiplicative")
plot(descomp_M)
```

Decomposition of multiplicative time series



```
# Descomposición aditiva
descomp_Ad <- decompose(x, type = "additive")
plot(descomp_Ad)
```

Decomposition of additive time series



Mostrar los índices estacionales

`descomp_M$seasonal`

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2016 0.9306617 0.8363763 1.0915441 1.1414179
## 2017 0.9306617 0.8363763 1.0915441 1.1414179
## 2018 0.9306617 0.8363763 1.0915441 1.1414179
## 2019 0.9306617 0.8363763 1.0915441 1.1414179
```

`descomp_Ad$seasonal`

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2016 -0.4395833 -1.0687500 0.5895833 0.9187500
## 2017 -0.4395833 -1.0687500 0.5895833 0.9187500
## 2018 -0.4395833 -1.0687500 0.5895833 0.9187500
## 2019 -0.4395833 -1.0687500 0.5895833 0.9187500
```

Calcula los índices estacionales y grafica la serie desestacionalizada

Calcular las ventas desestacionalizadas utilizando el modelo multiplicativo

```
ventas_desest_m <- ventas / descomp_M$seasonal
tiempo <- 1:length(ventas_desest_m)
```

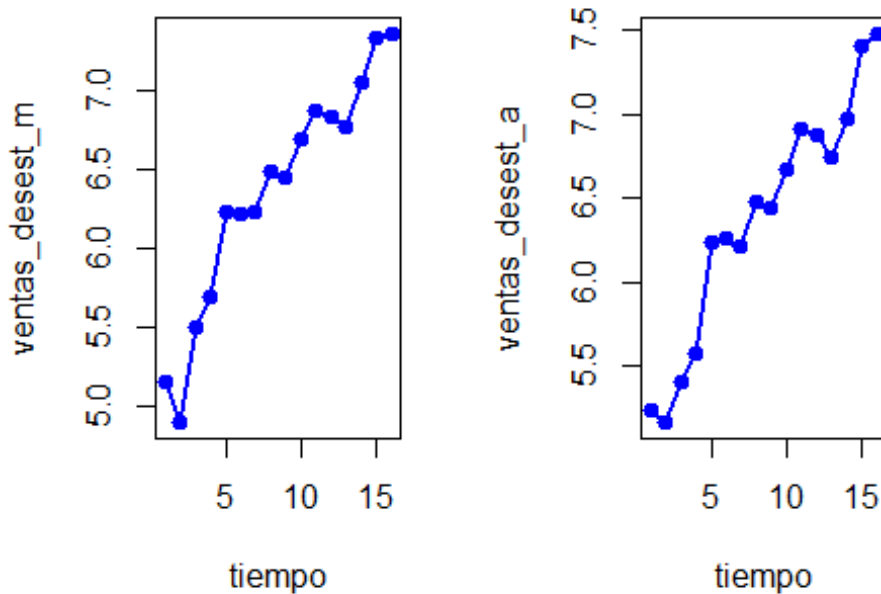
Calcular las ventas desestacionalizadas utilizando el modelo aditivo

```
ventas_desest_a <- ventas - descomp_Ad$seasonal
```

Graficar las ventas desestacionalizadas con ambos modelos

```
par(mfrow = c(1, 2))
plot(tiempo, ventas_desest_m, type = "o", col = "blue", lwd = 2, pch =
19, main = "Ventas Desestacionalizadas (Multiplicativo)")
plot(tiempo, ventas_desest_a, type = "o", col = "blue", lwd = 2, pch =
19, main = "Ventas Desestacionalizadas (Aditivo)")
```

Desestacionalizadas (Muntas Desestacionalizadas (



Analiza el modelo lineal de la tendencia

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

```
# Regresión lineal de las ventas desestacionalizadas (multiplicativo)
modelo_lineal_m <- lm(ventas_desest_m ~ tiempo)
summary(modelo_lineal_m)

##
## Call:
## lm(formula = ventas_desest_m ~ 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 ***
## ---
```

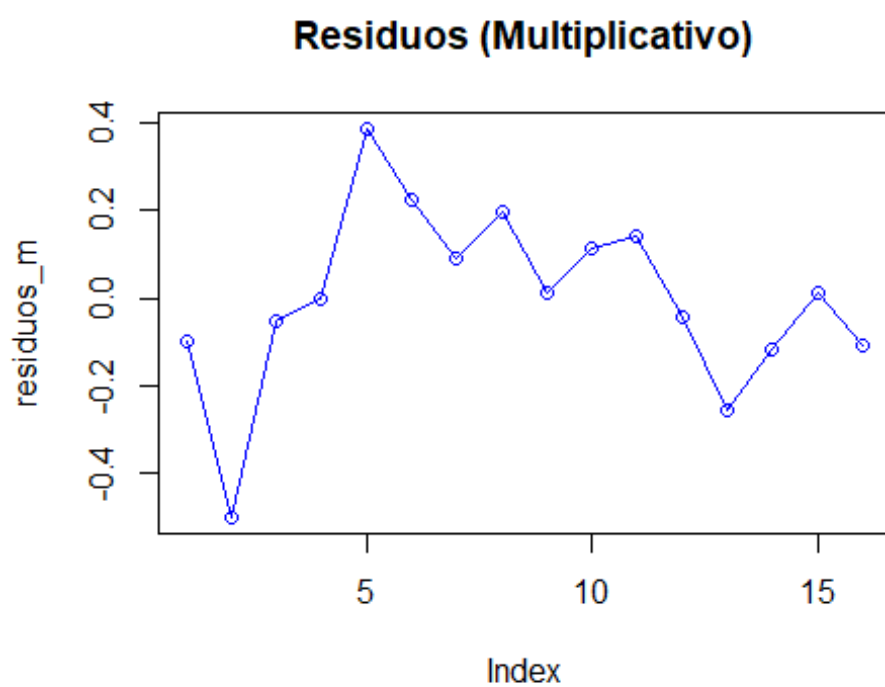
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.213 on 14 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9151
## F-statistic: 162.7 on 1 and 14 DF,  p-value: 4.248e-09

# Regresión lineal de Las ventas desestacionalizadas (aditivo)
modelo_lineal_a <- lm(ventas_desest_a ~ tiempo)
summary(modelo_lineal_a)

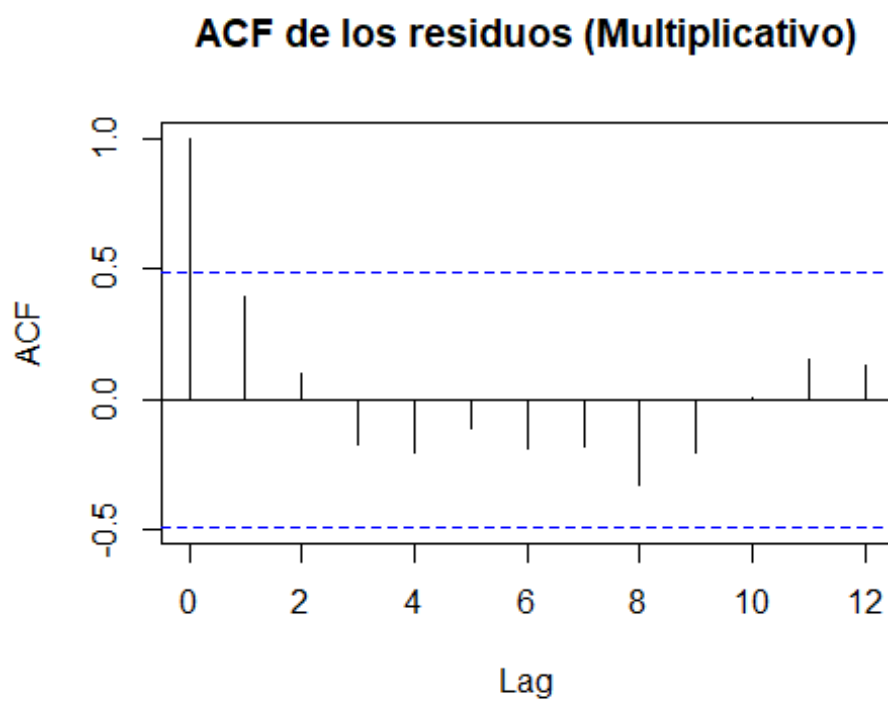
##
## Call:
## lm(formula = ventas_desest_a ~ tiempo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.2992 -0.1486 -0.0037  0.1005  0.3698
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.13917    0.10172   50.52  < 2e-16 ***
## tiempo       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
```

Analiza la significancia del modelo lineal, global e individual Haz el análisis de residuos

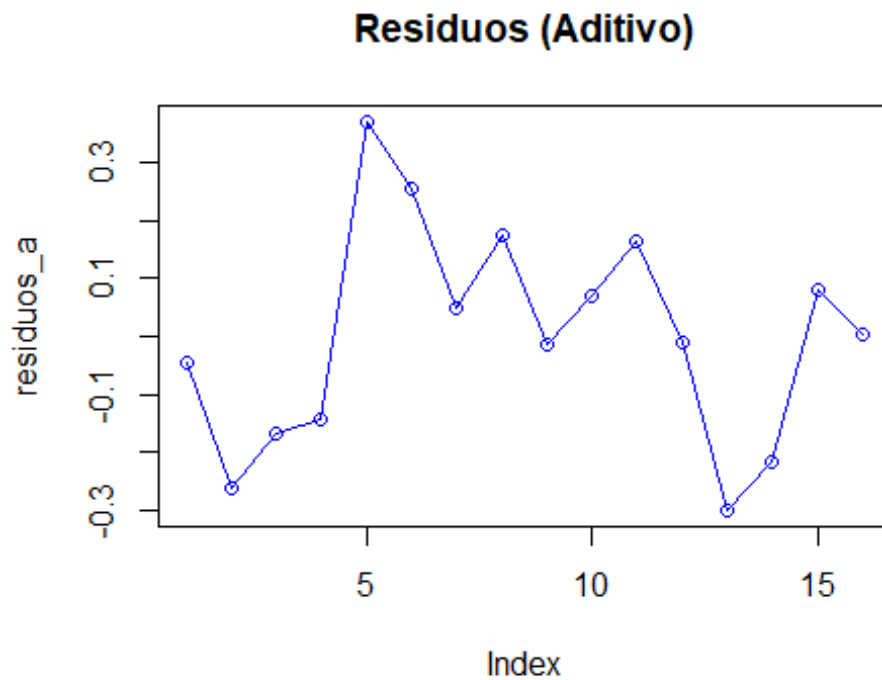
```
# Análisis de residuos (multiplicativo)
residuos_m <- residuals(modelo_lineal_m)
plot(residuos_m, main = "Residuos (Multiplicativo)", type = "o", col =
"blue")
```



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

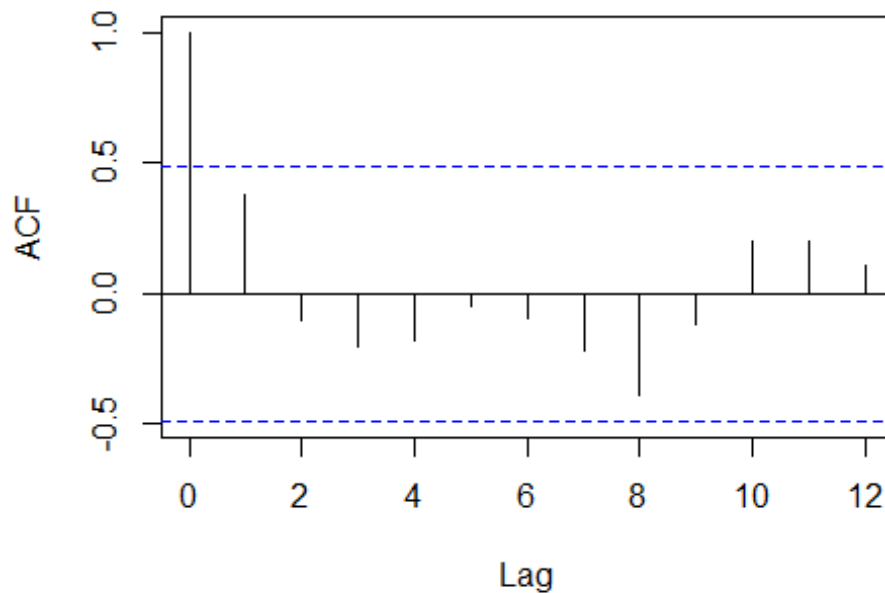


```
# Análisis de residuos (aditivo)
residuos_a <- residuals(modelo_lineal_a)
plot(residuos_a, main = "Residuos (Aditivo)", type = "o", col = "blue")
```



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


ACF de los residuos (Aditivo)



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

Cálculo del CME y EPAM (multiplicativo)

```
CME_m <- mean(residuos_m^2)
```

```
EPAM_m <- sqrt(CME_m)
```

Cálculo del CME y EPAM (aditivo)

```
CME_a <- mean(residuos_a^2)
```

```
EPAM_a <- sqrt(CME_a)
```

```
cat("CME (Multiplicativo):", CME_m, "\n")
```

```
## CME (Multiplicativo): 0.0397064
```

```
cat("EPAM (Multiplicativo):", EPAM_m, "\n")
```

```
## EPAM (Multiplicativo): 0.1992647
```

```
cat("CME (Aditivo):", CME_a, "\n")
```

```
## CME (Aditivo): 0.03291917
```

```
cat("EPAM (Aditivo):", EPAM_a, "\n")
```

```
## EPAM (Aditivo): 0.1814364
```

Explora un mejor modelo, por ejemplo un modelo cuadrático: $y =$

. Para ello transforma la variable ventas (recuerda que la regresión no lineal es una regresión lineal con una transformación).

```
# Modelo cuadrático (Multiplicativo)
modelo_cuadratico_m <- lm(ventas_desest_m ~ tiempo + I(tiempo^2))
summary(modelo_cuadratico_m)

##
## Call:
## lm(formula = ventas_desest_m ~ tiempo + I(tiempo^2))
##
## 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 ***
## I(tiempo^2) -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

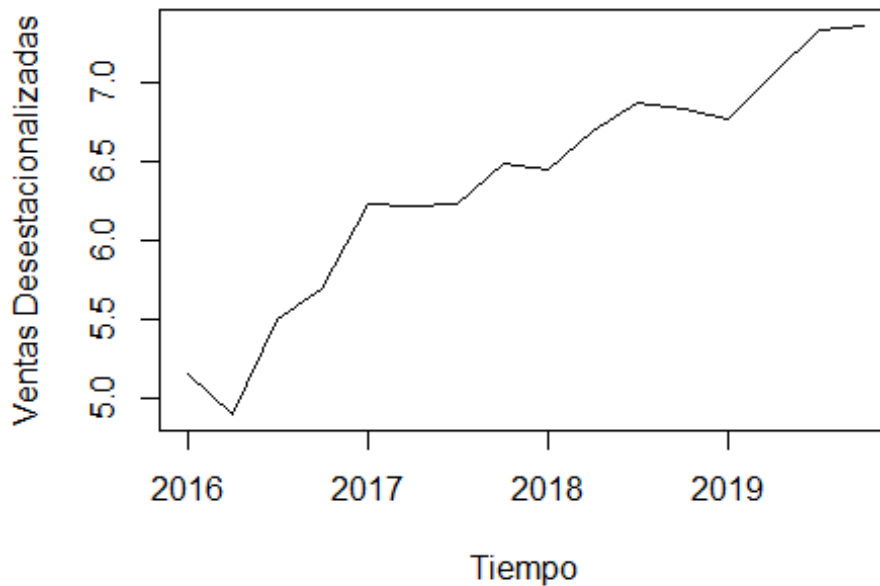
# Modelo cuadrático (Aditivo)
modelo_cuadratico_a <- lm(ventas_desest_a ~ tiempo + I(tiempo^2))
summary(modelo_cuadratico_a)

##
## Call:
## lm(formula = ventas_desest_a ~ tiempo + I(tiempo^2))
##
## 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 ***
## I(tiempo^2) -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

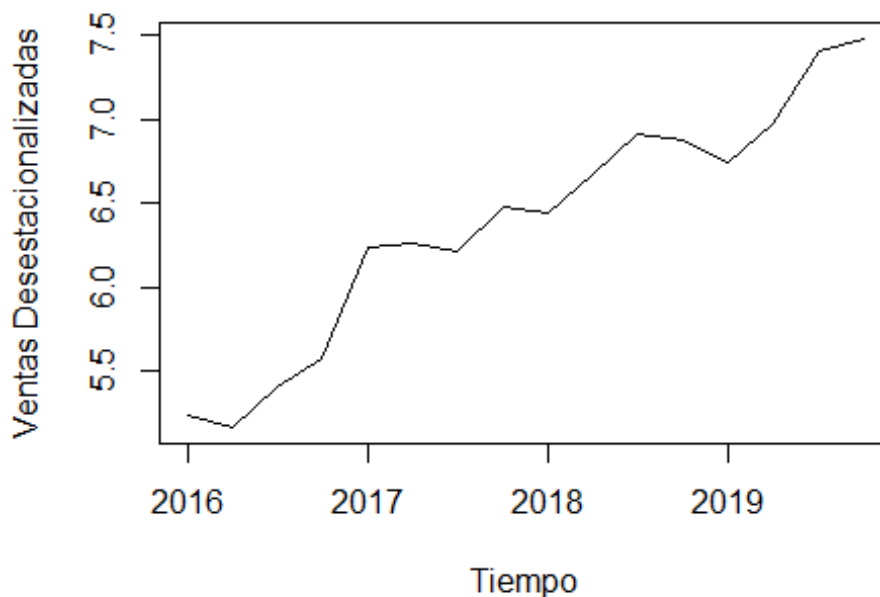
# Graficar el ajuste cuadrático (Multiplicativo)
plot(ventas_desest_m, main = "Ajuste de Modelo Cuadrático
(Multiplicativo)", xlab = "Tiempo", ylab = "Ventas Desestacionalizadas")
lines(tiempo, predict(modelo_cuadratico_m), col = "red", lwd = 2)
```

Ajuste de Modelo Cuadrático (Multiplicativo)



```
# Graficar el ajuste cuadrático (Aditivo)
plot(ventas_desest_a, main = "Ajuste de Modelo Cuadrático (Aditivo)",
xlab = "Tiempo", ylab = "Ventas Desestacionalizadas")
lines(tiempo, predict(modelo_cuadratico_a), col = "red", lwd = 2)
```

Ajuste de Modelo Cuadrático (Aditivo)



Concluye sobre el mejor modelo

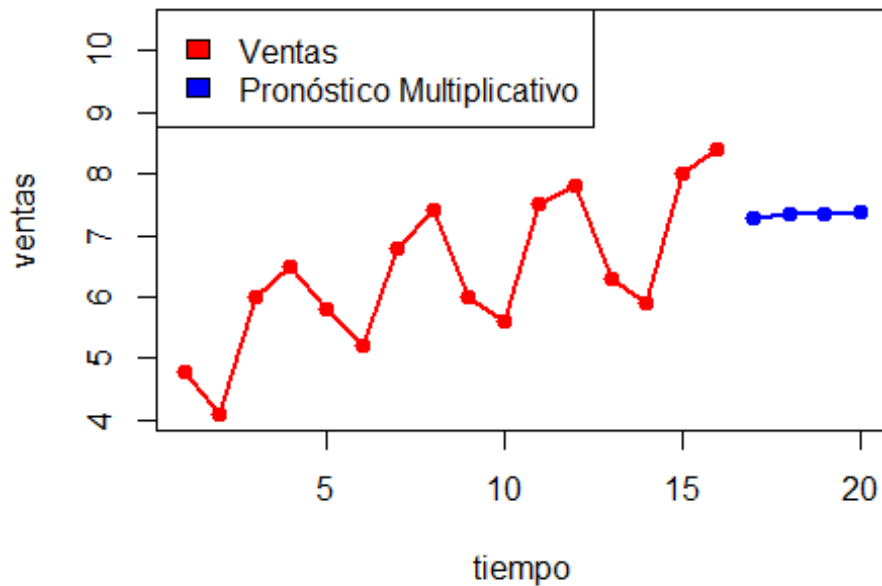
El mejor modelo es el modelo multiplicativo ya que este tiene una r cuadrada mayor al aditivo esta siendo la r cuadrada del modelo multiplicativo 0.9484 y la r cuadrada del modelo aditivo 0.9446 aunque estan a la par el modelo multiplicativo gana aunque no es mala idea usar el otro modelo ya que tiene buenos resultados.

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

```
# Pronóstico para el siguiente año (Multiplicativo)
nuevo_tiempo <- data.frame(tiempo = (length(ventas_desest_m) +
1):(length(ventas_desest_m) + 4))
pronostico_m <- predict(modelo_cuadratico_m, nuevo_tiempo)

# Pronóstico para el siguiente año (Aditivo)
pronostico_a <- predict(modelo_cuadratico_a, nuevo_tiempo)

# Graficar el pronóstico junto con los datos originales (Multiplicativo)
plot(tiempo, ventas, type = "o", col = "red", lwd = 2, pch = 19, xlim =
c(1, length(ventas) + 4), ylim = c(min(ventas), max(ventas) + 2))
lines(nuevo_tiempo$tiempo, pronostico_m, col = "blue", type = "o", pch =
19, lwd = 2)
legend("topleft", legend = c("Ventas", "Pronóstico Multiplicativo"), fill
= c("red", "blue"))
```



```
# Graficar el pronóstico junto con los datos originales (Aditivo)
plot(tiempo, ventas, type = "o", col = "red", lwd = 2, pch = 19, xlim =
c(1, length(ventas) + 4), ylim = c(min(ventas), max(ventas) + 2))
lines(nuevo_tiempo$tiempo, pronostico_a, col = "blue", type = "o", pch =
19, lwd = 2)
legend("topleft", legend = c("Ventas", "Pronóstico Aditivo"), fill =
c("red", "blue"))
```

