

# A8-Series de Tiempo no Estaiconales

AUTHOR

Alfredo García

## Leemos los datos

```
# Crear el marco de datos
datos <- data.frame(Año = rep(1:4, 4),
                    Trimestre = rep(1:4, times = 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
```

```
datos
```

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

Realiza el gráfico de dispersión. Observa la tendencia y los ciclos.

```
# Cargar la librería ggplot2 para crear gráficos
library(ggplot2)

# Crear una etiqueta única para cada trimestre independientemente del año
datos$Trimestre_unico <- interaction(datos$Año, datos$Trimestre, drop = TRUE)

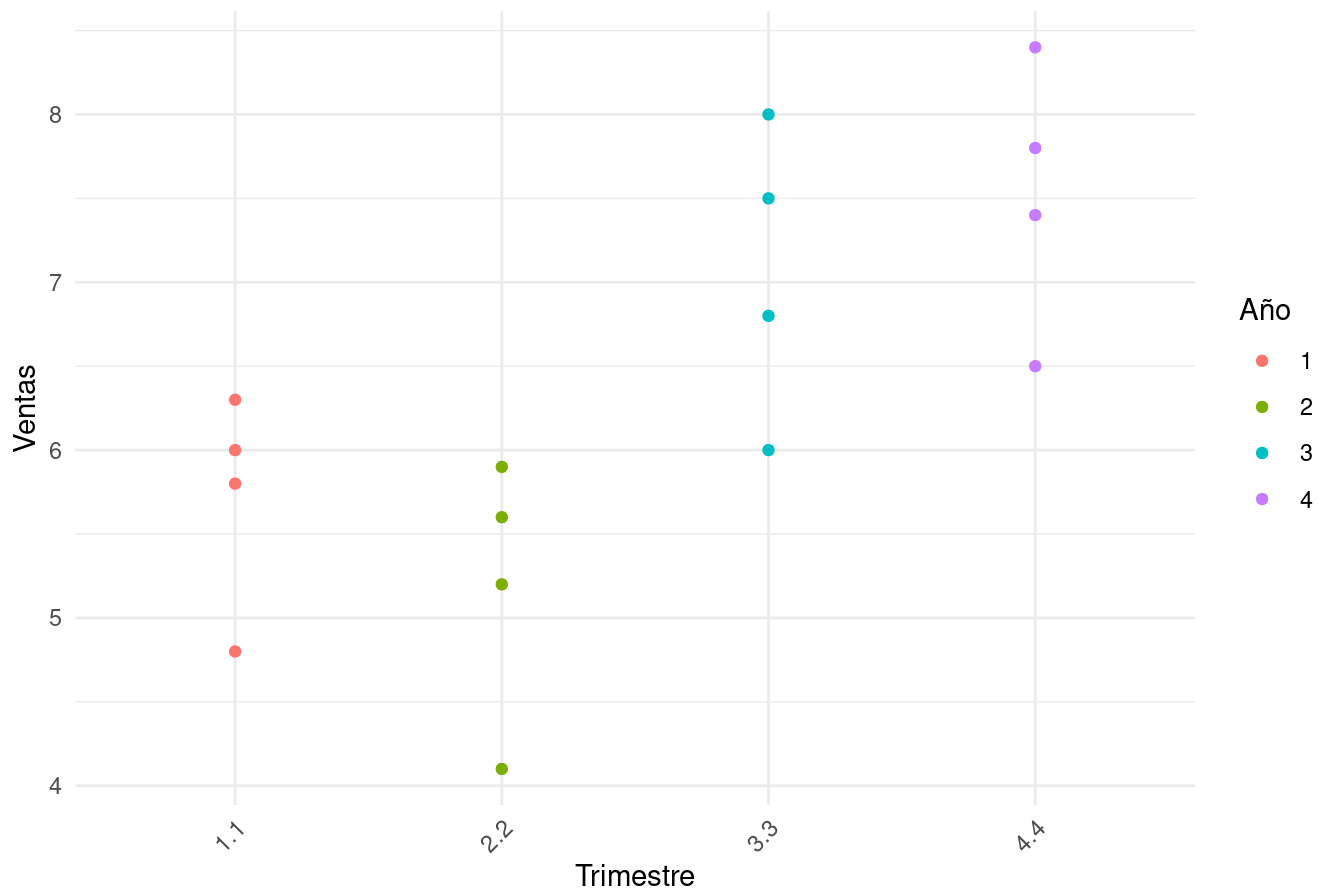
# Crear el gráfico de dispersión de la serie de tiempo
ggplot(datos, aes(x = Trimestre_unico, y = Ventas, color = factor(Año))) +
  geom_point() +
  labs(
    title = "Serie de Tiempo de Ventas por Trimestre",
    x = "Trimestre",
    y = "Ventas",
```

```

color = "Año"
) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

### Serie de Tiempo de Ventas por Trimestre



## Realiza el análisis de tendencia y estacionalidad

## Descompón la serie en sus 3 componentes e interprétalos

```

# Crear una serie de tiempo
library(forecast)

```

Registered S3 method overwritten by 'quantmod':

```

method      from
as.zoo.data.frame zoo

```

```

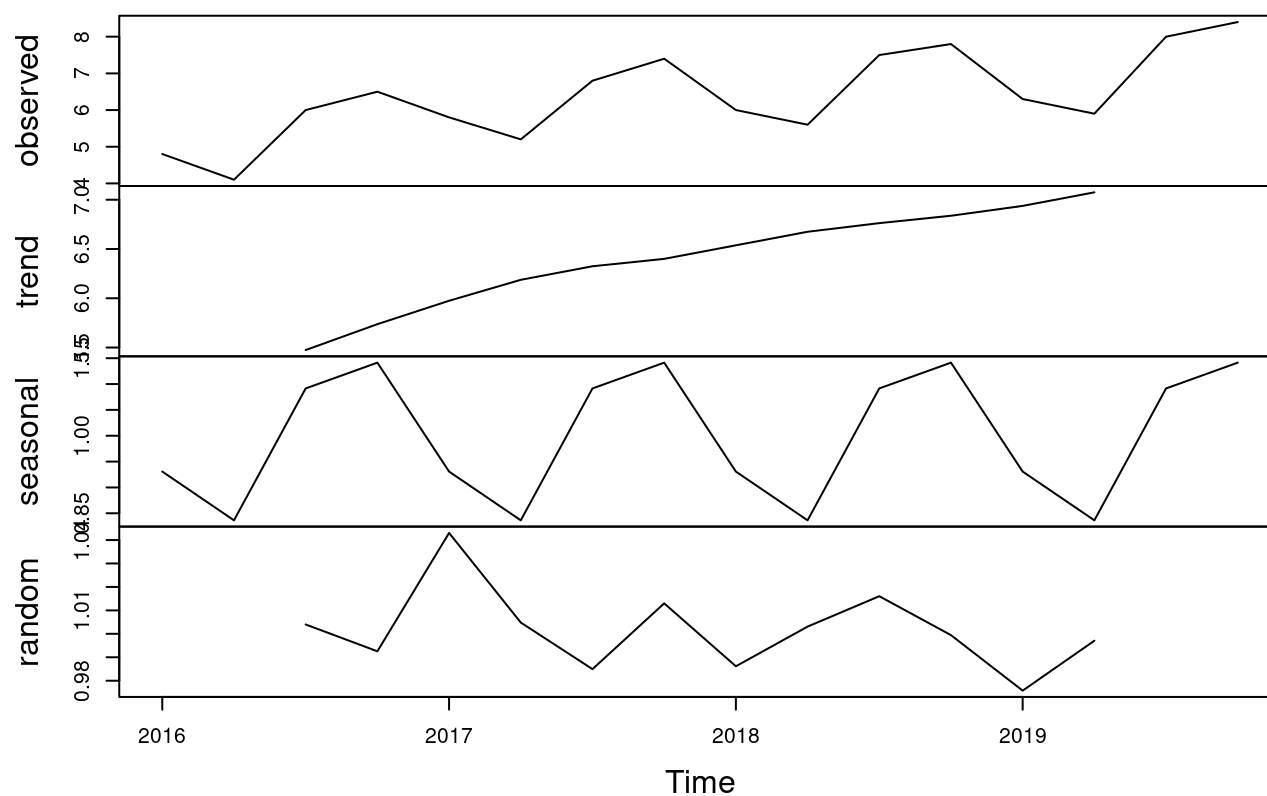
T <- ts(datos$Ventas, frequency = 4, start = c(2016, 1))

# Descomponer la serie de tiempo
D_m <- decompose(T, type = "multiplicative")

```

```
# Gráfico de dispersión
plot(D_m)
```

## Decomposition of multiplicative time series

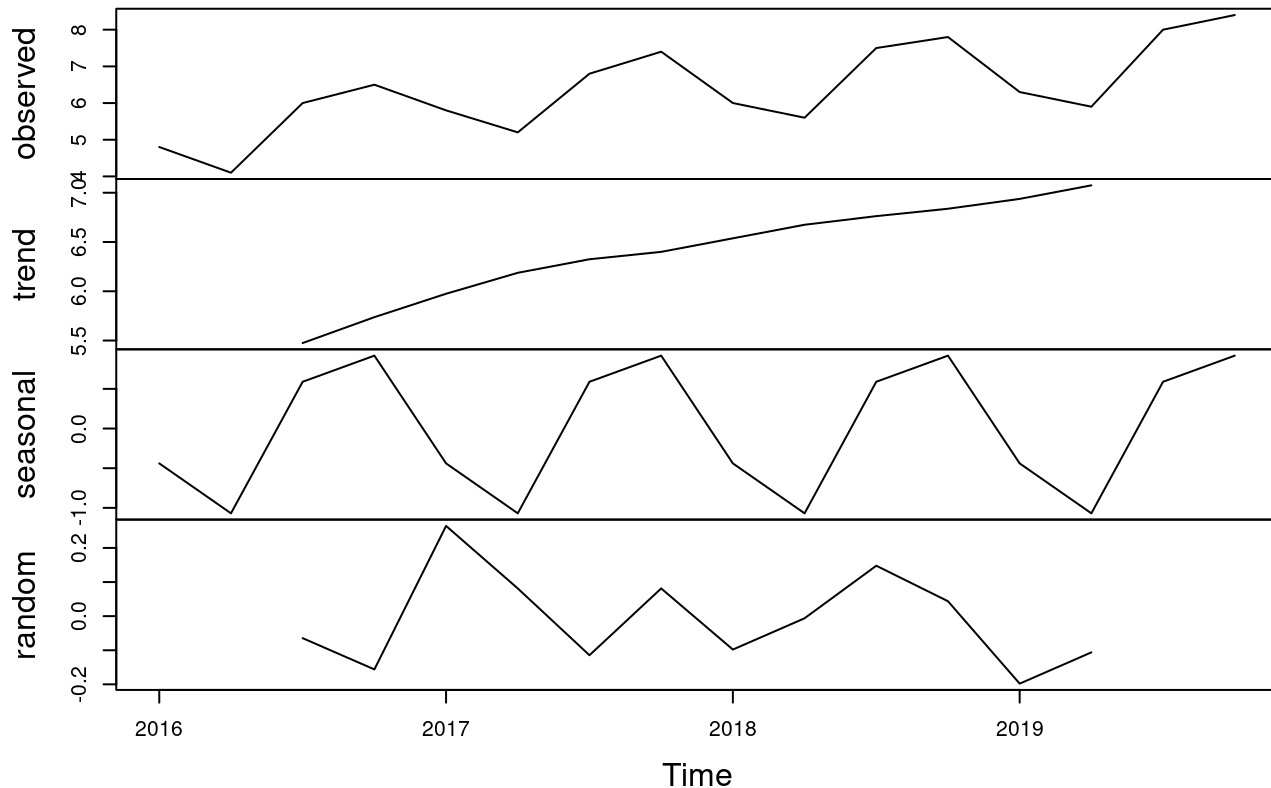


```
# Crear una serie de tiempo
library(forecast)
T <- ts(datos$Ventas, frequency = 4, start = c(2016, 1))

# Descomponer la serie de tiempo
D_a <- decompose(T, type = "additive")

# Gráfico de dispersión
plot(D_a)
```

## Decomposition of additive time series



**Tendencia:** Es la dirección general en la que la serie de tiempo parece estar creciendo o decreciendo a lo largo del tiempo. Si la serie de tiempo muestra un patrón ascendente o descendente a largo plazo, esta componente representará ese comportamiento.

**Estacionalidad:** Se refiere a patrones recurrentes o cíclicos que ocurren en la serie de tiempo en intervalos regulares. Por ejemplo, si hay fluctuaciones que se repiten cada cierto período (como estacionalidad anual, trimestral, mensual, etc.), este componente representará esas variaciones predecibles.

**Componente aleatorio:** También conocido como residuo, este componente captura cualquier variación que no pueda ser atribuida a la tendencia o a la estacionalidad. Representa el ruido o la variabilidad irregular que queda después de que se han tenido en cuenta la tendencia y la estacionalidad.

### Analiza el modelo lineal de la tendencia:

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

```
ventas_desestacionalizadas = (D_a$x)/(D_a$seasonal)
x3 = 1:16
y3 = ventas_desestacionalizadas
```

```
N3 = lm(y3~x3)
N3
```

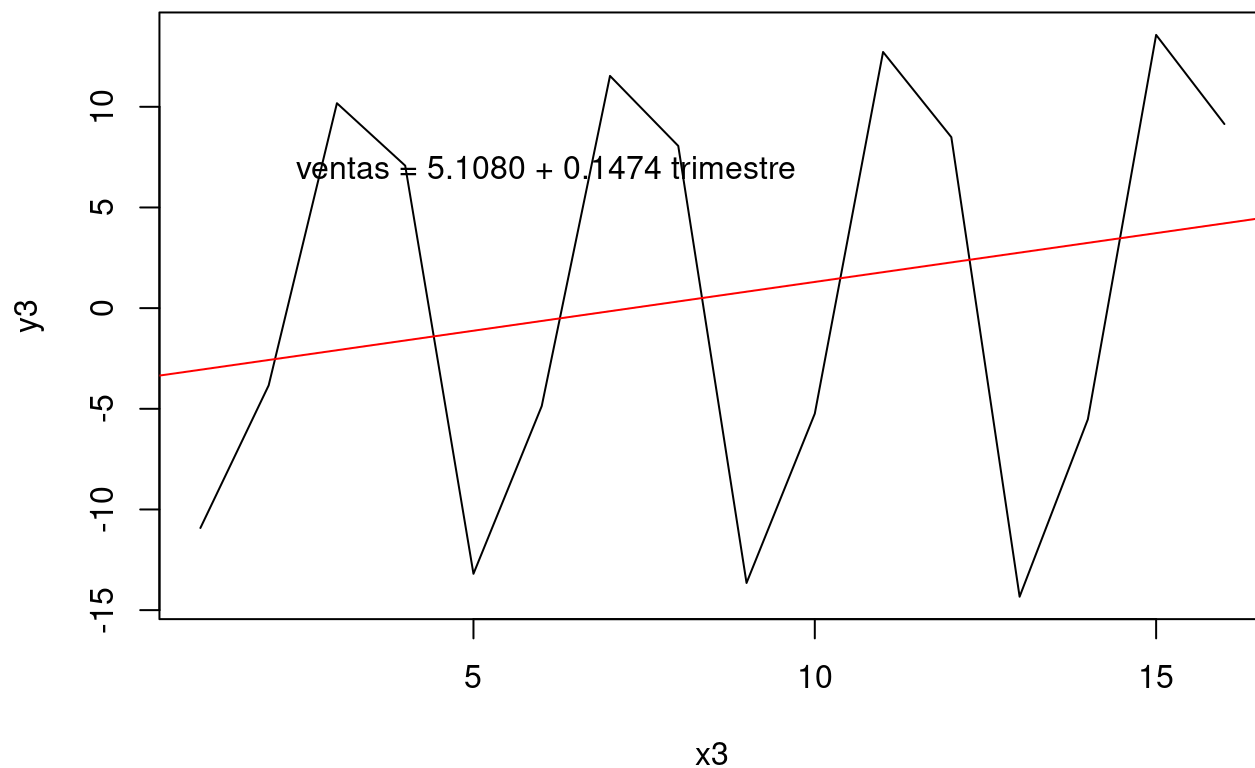
Call:

```
lm(formula = y3 ~ x3)
```

Coefficients:

(Intercept)	x3
-3.5443	0.4847

```
plot(x3, y3, type = "l")
abline(N3, col = "red")
text(6, 7, "ventas = 5.1080 + 0.1474 trimestre")
```



```
summary(N3)
```

Call:

```
lm(formula = y3 ~ x3)
```

Residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----

-17.088   -8.085   1.836   8.971   12.267

#### Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-3.5443	5.5166	-0.642	0.531
x3	0.4847	0.5705	0.850	0.410

Residual standard error: 10.52 on 14 degrees of freedom

Multiple R-squared: 0.04902, Adjusted R-squared: -0.0189

F-statistic: 0.7217 on 1 and 14 DF, p-value: 0.4099

#### multiplicative

```
ventas_desestacionalizadas = (D_m$x)/(D_m$seasonal)
x3 = 1:16
y3 = ventas_desestacionalizadas
N3 = lm(y3~x3)
N3
```

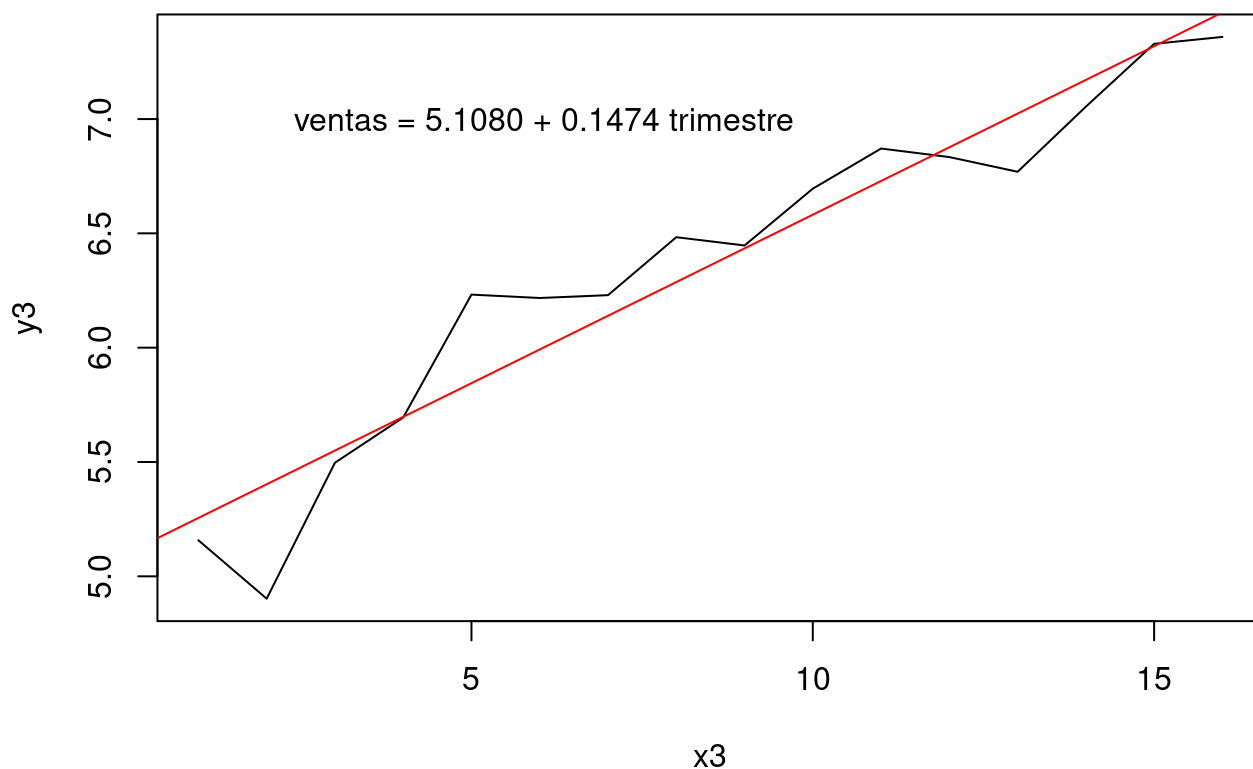
#### Call:

```
lm(formula = y3 ~ x3)
```

#### Coefficients:

		x3
(Intercept)	5.1080	0.1474

```
plot(x3, y3, type = "l")
abline(N3, col = "red")
text(6, 7, " ventas = 5.1080 + 0.1474 trimestre")
```



summary(N3)

Call:

```
lm(formula = y3 ~ x3)
```

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 ***
x3	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

Dibuja la recta junto con las ventas desestacionalizadas. Analiza la pertinencia del modelo lineal:

Significancia de  $\beta_1$  Variabilidad explicada por el modelo (c.d) Análisis de los residuos

```
# Obtener los residuos del modelo
residuos <- residuals(N3)

# Gráfico de los residuos
plot(x3, residuos, type = "b", xlab = "Trimestre", ylab = "Residuos",
     main = "Gráfico de Residuos")

# Línea horizontal en 0 para referencia
abline(h = 0, col = "red")
```



```
# Análisis estadístico de los residuos
summary(residuos)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-0.500706	-0.100074	0.003699	0.000000	0.120706	0.387173

## Prueba de normalidad

```
# Prueba de normalidad Shapiro-Wilk
shapiro.test(residuos)
```



### Shapiro-Wilk normality test

data: residuos

W = 0.96379, p-value = 0.7307

The null hypothesis states that the population is normally distributed i.e if the p-value is greater than 0.05, then the null hypothesis is accepted.

## Calcula el CME y el EPAM (promedio de los errores porcentuales) de la predicción de la serie de tiempo.

```
# Obtener predicciones del modelo
predicciones <- predict(N3)

# Calcular el Error Cuadrático Medio (CME)
residuos_cuadrados <- residuos^2
CME <- mean(residuos_cuadrados)

# Calcular el Error Porcentual Absoluto Medio (EPAM)
errores_porcentuales <- abs(residuos / y3) * 100
EPAM <- mean(errores_porcentuales, na.rm = TRUE)

# Mostrar los resultados
CME
```

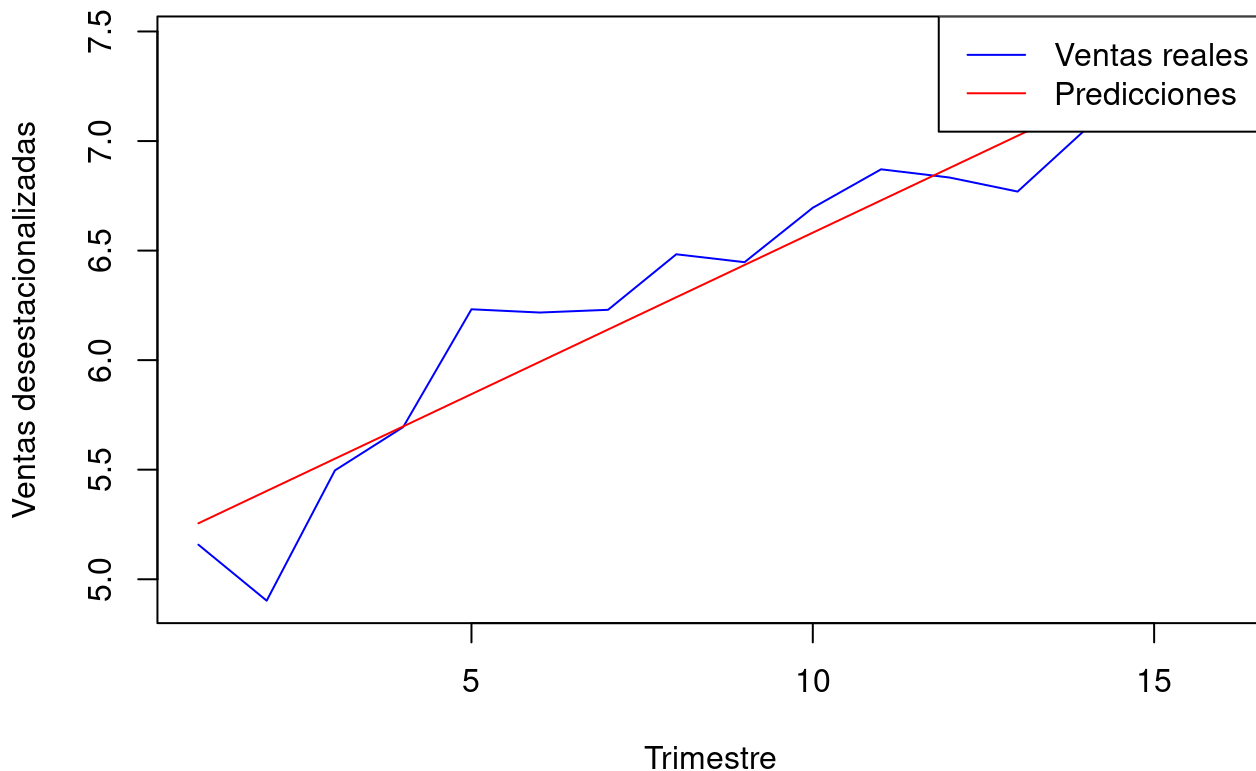
[1] 0.0397064

EPAM

[1] 2.439533

## Dibuja el gráfico de los valores de las ventas y las predicciones vs el tiempo

```
# Graficar valores reales de ventas y predicciones versus tiempo
plot(x3, y3, type = "l", col = "blue", ylim = range(c(y3, predicciones)), xlab = "Trimeste",
lines(x3, predicciones, col = "red")
legend("topright", legend = c("Ventas reales", "Predicciones"), col = c("blue", "red"), l
```



## Concluye sobre el modelo: de acuerdo al análisis de verificación de los supuestos, ¿es el mejor modelo que puedes obtener?

El mejor modelo en este caso es el multiplicativo, ya que, al hacer el modelo podemos darnos cuenta de que los coeficientes en el caso del aditivo no son significativos, por lo que, las conclusiones que saquemos de ahí no son confiables ( $\alpha = 0.05$ )

## Propón un posible mejor modelo para la tendencia de los datos.

Un mejor modelo para la tendencia de los datos si nos limitamos a la actividad podemos decir que la mejor alternativa es el multiplicativo que pasa todos los test y además este en su coeficiente  $B_1$  es significativo a diferencia del modelo anterior (aditivo). Y si podemos explorar en otras alternativas, diría que el uso de modelos como SARIMAX o SARIMA podrían ayudar en gran medida a mejorar el modelo, debido a la presencia de tendencia y que se pueden ver ciclos marcados en las graficas en donde descompusimos los componentes de la TS.

## Realiza el pronóstico para el siguiente año.

```
f = function(x) {5.1080 + 0.1474*x}
a1 = D_m$seasonal[1]
a2 = D_m$seasonal[2]
a3 = D_m$seasonal[3]
```

```
a4 = D_m$seasonal[4];
print(f(17)*a1*1000)
```

```
[1] 7085.872
```

```
print(f(18)*a2*1000)
```

```
[1] 6491.284
```

```
print(f(19)*a3*1000)
```

```
[1] 8632.585
```

```
print(f(20)*a4*1000)
```

```
[1] 9195.263
```

**Realiza el problema de “Un problemilla más” sobre las ventas trimestrales de libros de texto universitarios.**

```
ventas <- data.frame(
  trimestre = c(1, 2, 3, 4),
  a1 = c(1690, 940, 2625, 2500),
  a2 = c(1800, 900, 2900, 2360),
  a3 = c(1850, 1100, 2930, 2615)
)
```

**a) Encuentre los promedios móviles de cuatro trimestres y los promedios móviles centrados**

```
ventas$promedio_movil4 <- rowMeans(ventas[, c("a1", "a2", "a3")])
ventas$promedio_movil_centrado <- c(NA, head(ventas$promedio_movil4, -1) + tail(ventas$pr
print(ventas[, c("trimestre", "promedio_movil4", "promedio_movil_centrado")])
```

	trimestre	promedio_movil4	promedio_movil_centrado
1	1	1780.000	NA
2	2	980.000	1380.000
3	3	2818.333	1899.167
4	4	2491.667	2655.000

**b) Calcule los índices estacionales de los cuatro trimestres**

```
promedio_anual <- colMeans(ventas[, c("a1", "a2", "a3")])
indices_estacionales <- t(t(ventas[, c("a1", "a2", "a3")]) / promedio_anual) * 100
```

```
print(indices_estacionales)
```

	a1	a2	a3
[1,]	87.16957	90.45226	87.11006
[2,]	48.48485	45.22613	51.79517
[3,]	135.39652	145.72864	137.96351
[4,]	128.94907	118.59296	123.13125

**c) ¿Cuándo obtiene la editorial el mayor índice estacional? ¿Parece razonable este**

```
trimestre_max_indice <- which.max(apply(indices_estacionales, 2, max))  
trimestre_max_indice
```

a2  
2

Sí, es razonable porque podemos ver desde un inicio que se trata del trimestre en el que hay una mayor cantidad de vetas con respecto a los demás