

# Act8\_SeriesTiempo

Frida Cano

2023-11-21

## Series de tiempo no estacionarias - tendencia

Usa los datos de las ventas de televisores para familiarizarte con el análisis de tendencia de una serie de tiempo:

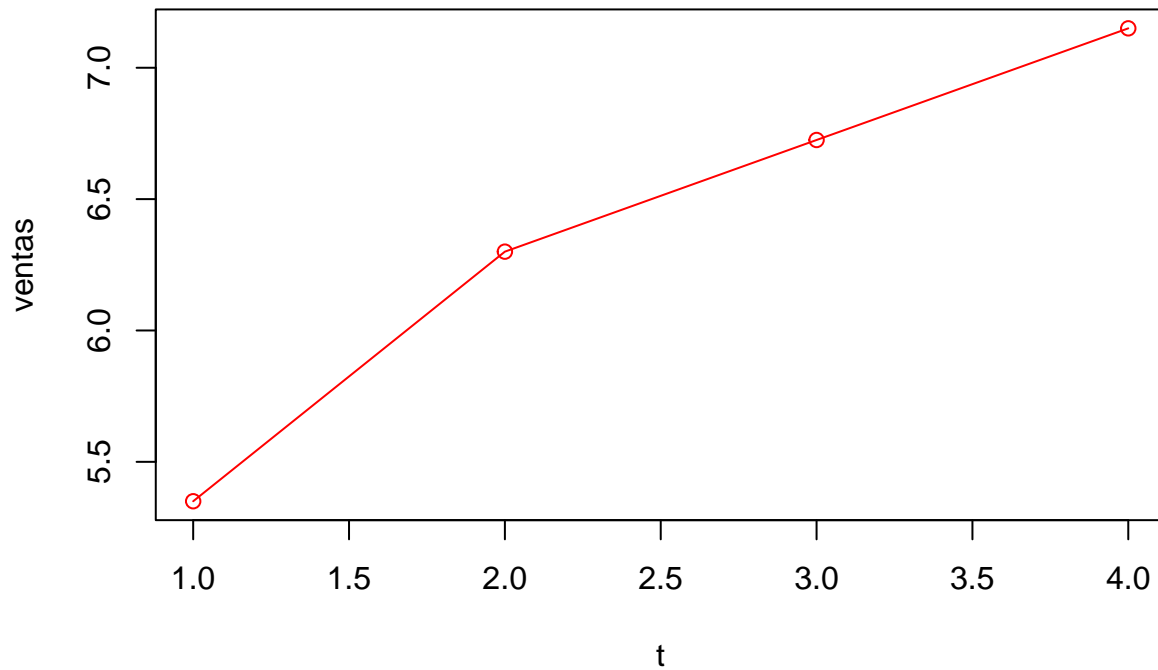
Año	1				2				3				4			
Trimestre	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
Ventas (miles)	4.8	4.1	6.0	6.5	5.8	5.2	6.8	7.4	6.0	5.6	7.5	7.8	6.3	5.9	8.0	8.4

```
t = 1:4  
  
y1 <- c(4.8,4.1,6.0,6.5)  
y2 <- c(5.8,5.2,6.8,7.4)  
y3 <- c(6.0,5.6,7.5,7.8)  
y4 <- c(6.3,5.9,8.0,8.4)  
m_y1 <- mean(y1)  
m_y2 <- mean(y2)  
m_y3 <- mean(y3)  
m_y4 <- mean(y4)  
ventas <- c(m_y1,m_y2,m_y3,m_y4)
```

## Gráfico de dispersión de los datos

```
plot(t,ventas,type="o",col="red",main = "Ventas por año")
```

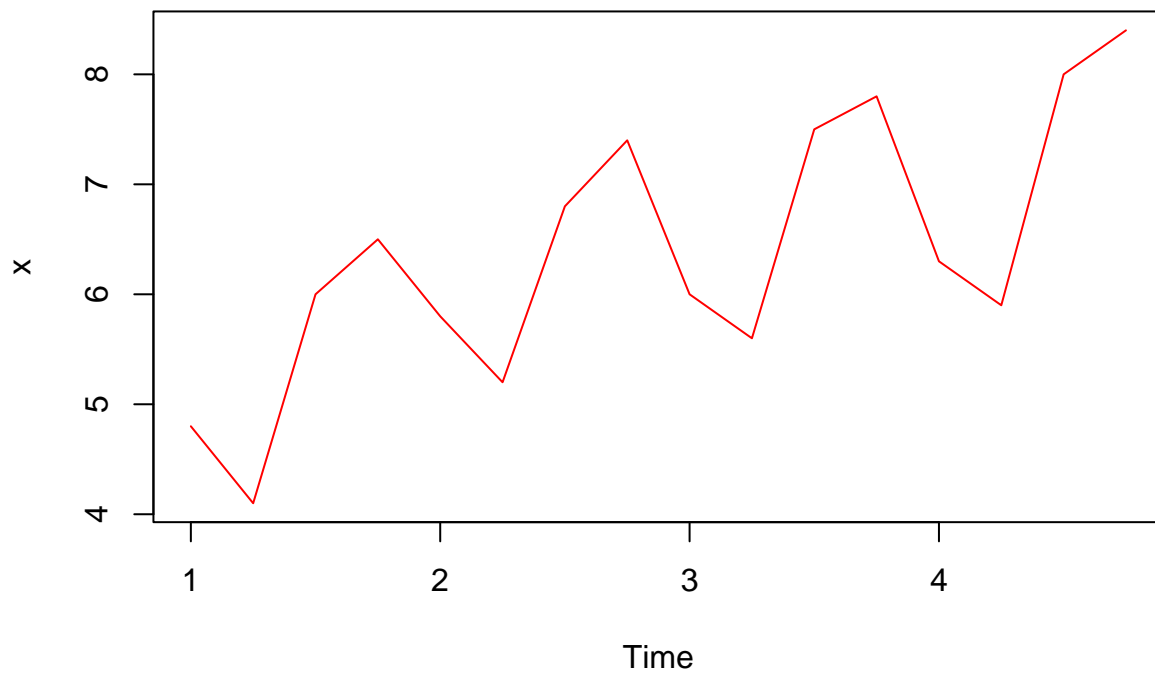
## Ventas por año



##

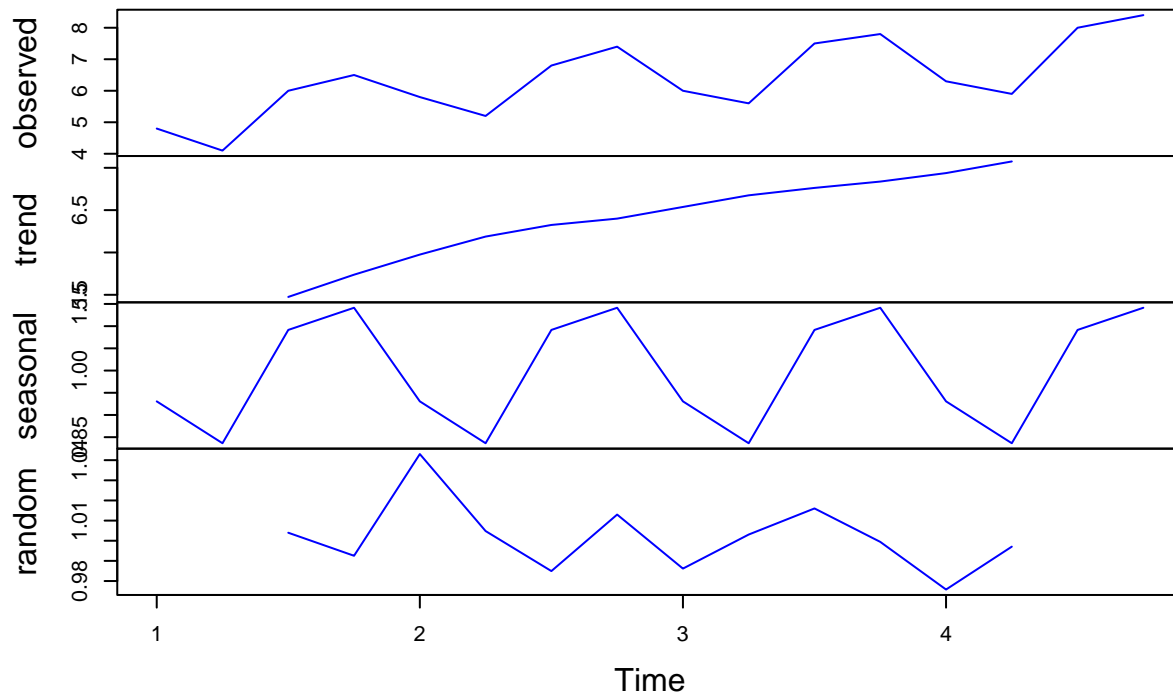
Análisis de tendencia y estacionalidad

```
#Tendencia
ventas_compl <- 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)
x = ts(ventas_compl,frequency = 4,start=c(2016,1))
plot.ts(x, col="red")
```



```
T = decompose(x,type="m")
plot(T,col="blue")
```

## Decomposition of multiplicative time series



```
T$seasonal
```

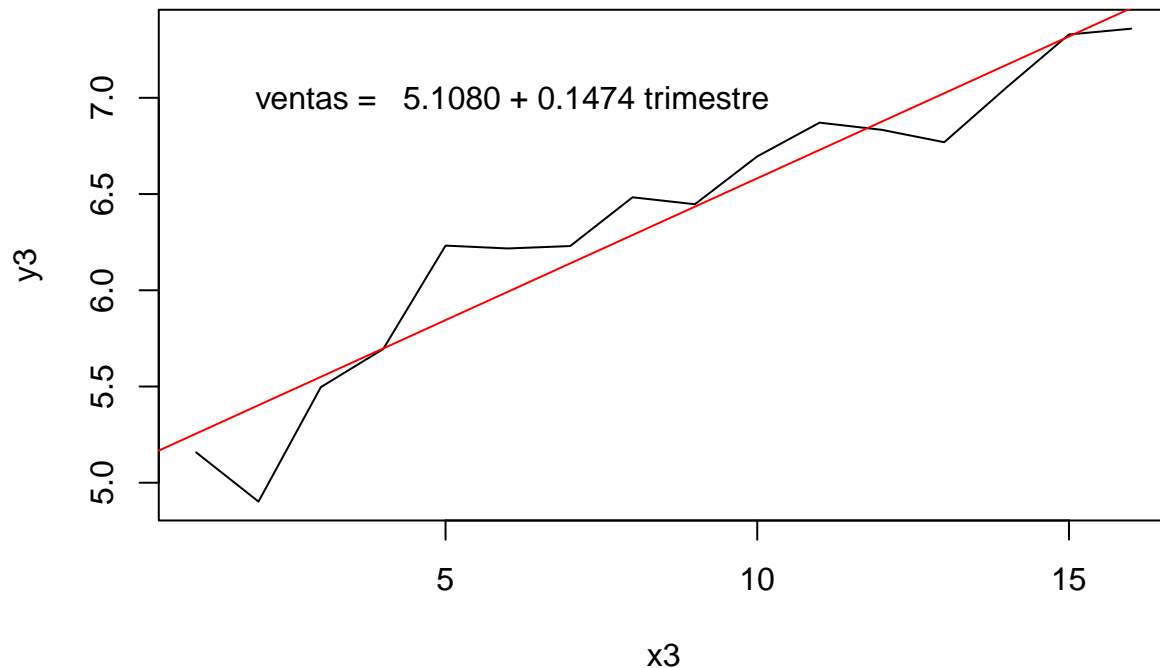
```
##          Qtr1      Qtr2      Qtr3      Qtr4
## 1 0.9306617 0.8363763 1.0915441 1.1414179
## 2 0.9306617 0.8363763 1.0915441 1.1414179
## 3 0.9306617 0.8363763 1.0915441 1.1414179
## 4 0.9306617 0.8363763 1.0915441 1.1414179
```

## Análisis del modelo lineal de la tendencia

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

##
## Call:
## lm(formula = y3 ~ x3)
##
## Coefficients:
## (Intercept)          x3
##      5.1080      0.1474

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



```
# Obtener la variabilidad explicada (R²)
r_squared <- summary(N3)$r.squared
print(paste("Coeficiente de determinación (R²): ", r_squared))
```

```
## [1] "Coeficiente de determinación (R²): 0.920791119635599"
```

```
# Análisis de los residuos
residuos <- residuals(N3)
```

```
# Prueba de normalidad de los residuos (Shapiro-Wilk)
shapiro_test <- shapiro.test(residuos)
print("Prueba de normalidad de los residuos:")
```

```
## [1] "Prueba de normalidad de los residuos:"
```

```
print(shapiro_test)
```

```
##
## Shapiro-Wilk normality test
##
## data:  residuos
## W = 0.96379, p-value = 0.7307
```

1. En la primera representación gráfica, observamos los datos trazados sin realizar ningún tipo de ajuste.
2. En la segunda gráfica, se evidencia la dirección hacia la cual se inclinan los datos; en nuestro contexto, se trata de una tendencia ascendente.
3. La tercera representación gráfica pone de manifiesto los patrones recurrentes en la serie temporal. En nuestro caso, se identifica una frecuencia de 4 datos por temporada, lo que resulta en la visualización de picos y valles.
4. Esta gráfica representa el ruido de la serie, por lo tanto, no debería exhibir un patrón claro como tal.

## Error Cuadrático Medio (CME)

```
# Crear un conjunto de datos con valores de x dentro del intervalo deseado
nuevos_x <- data.frame(x3)

# Obtener los valores predichos para los nuevos valores de x
valores_predichos <- predict(N3, nuevos_x)

# Mostrar los valores predichos
e <- (y3-valores_predichos)

CME2=mean(e^2,na.rm=TRUE)
CME2
```

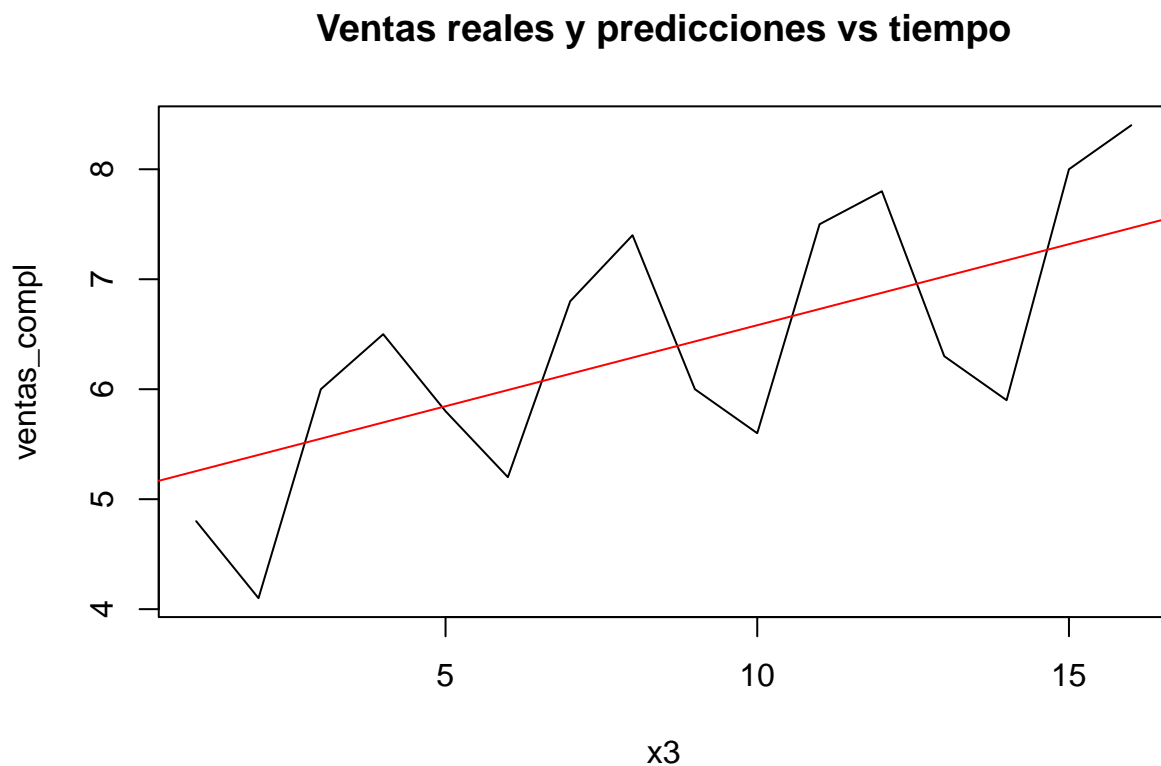
```
## [1] 0.0397064
```

## Error Porcentual Absoluto Medio

```
epam <- mean(abs((y3 - valores_predichos) / y3) * 100)
print(paste("EPAM:", epam))
```

```
## [1] "EPAM: 2.4395328407222"
```

```
plot(x3, ventas_compl, type = "l",main = "Ventas reales y predicciones vs tiempo")
abline(N3, col = "red")
```



## Un problemilla mas

A continuación, se presentan los datos correspondientes a los últimos tres años de ventas trimestrales (número de ejemplares vendidos) de un libro de texto universitario.

Trimestre	Año 1	Año 2	Año 3
1	1690	1800	1850
2	940	900	1100
3	2625	2900	2930
4	2500	2360	2615

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

```
t <- c(1,2,3,4,5,6,7,8,9,10,11,12)
y <- c(1690,940,2625,2500,1800,900,2900,2360,1850,1100,2930,2615)
n <- length(y)
a = 4
p = NA
e = NA
for(i in 1:(n-a)){
  p[i+a]=(y[i]+y[i+1]+y[i+2])/a;
  e[i+a]=p[i+a]-y[i+a]
}
T = data.frame(t,y,p,e^2)
T
```

```
##      t      y      p      e.2
## 1    1 1690     NA     NA
## 2    2  940     NA     NA
## 3    3 2625     NA     NA
## 4    4 2500     NA     NA
## 5    5 1800 1313.75 236439.1
## 6    6  900 1516.25 379764.1
## 7    7 2900 1731.25 1365976.6
## 8    8 2360 1300.00 1123600.0
## 9    9 1850 1400.00 202500.0
## 10  10 1100 1540.00 193600.0
## 11  11 2930 1777.50 1328256.2
## 12  12 2615 1327.50 1657656.2
```