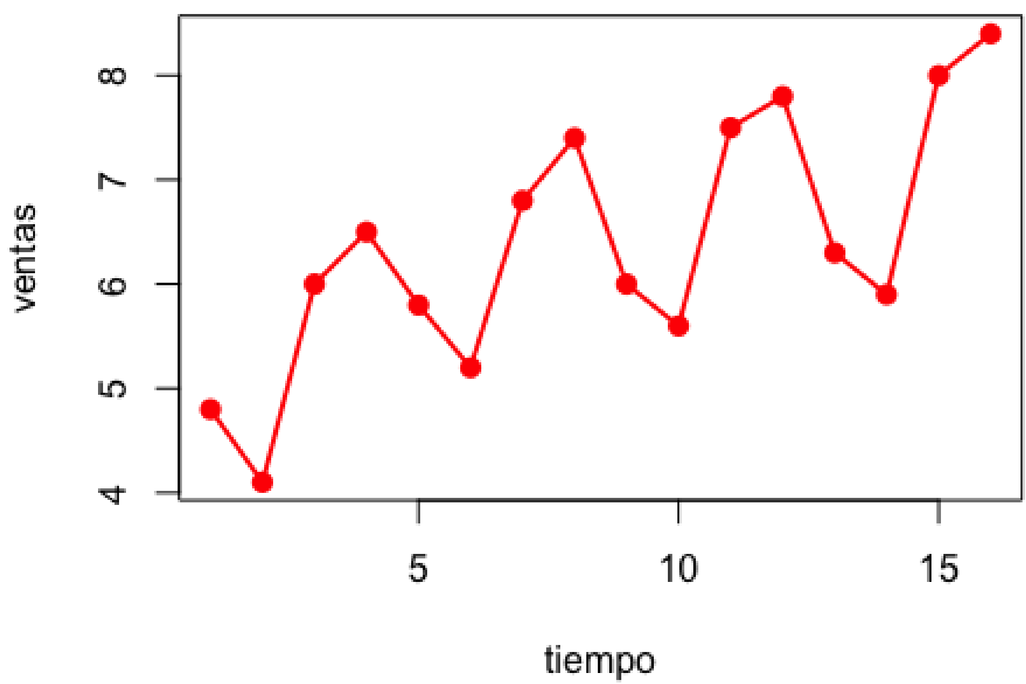
SeriesDeTiempo

Cleber Perez

2024-11-12

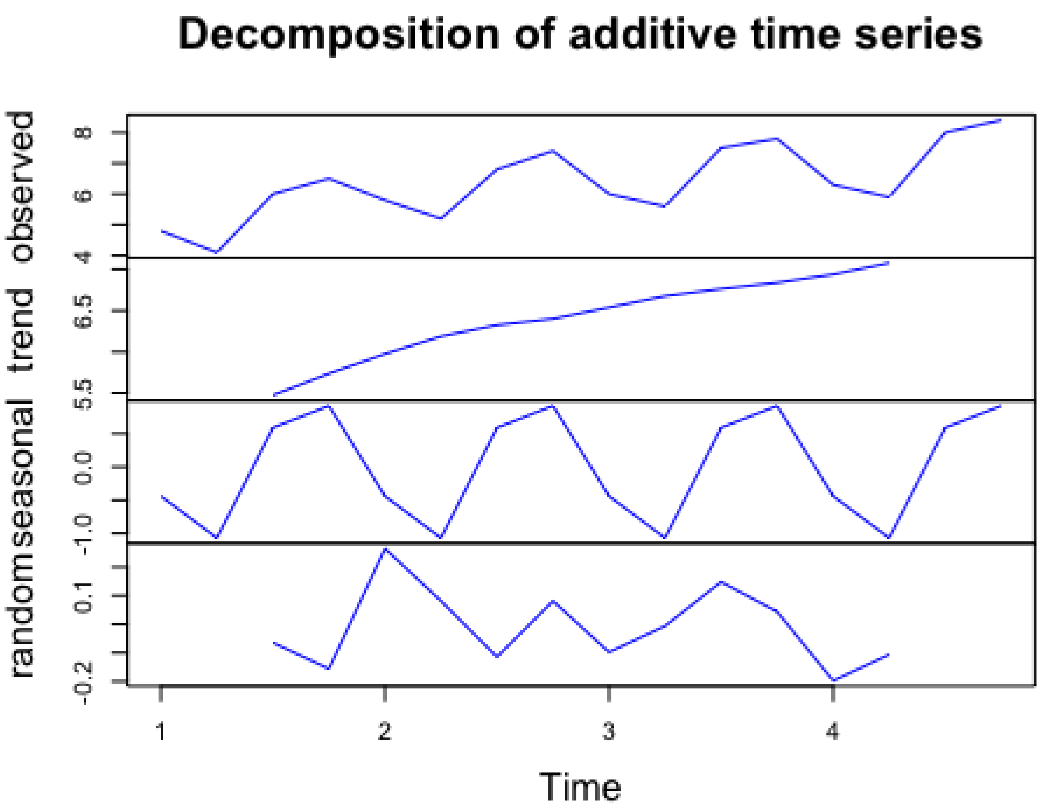
Realiza el análisis de tendencia y estacionalidad

```
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)
x= ts(ventas, frequency = 4, start(c(2016,1)))
tiempo = 1:16
plot(tiempo, ventas, col ="red", type = "o", lwd = 2, pch = 19)
```

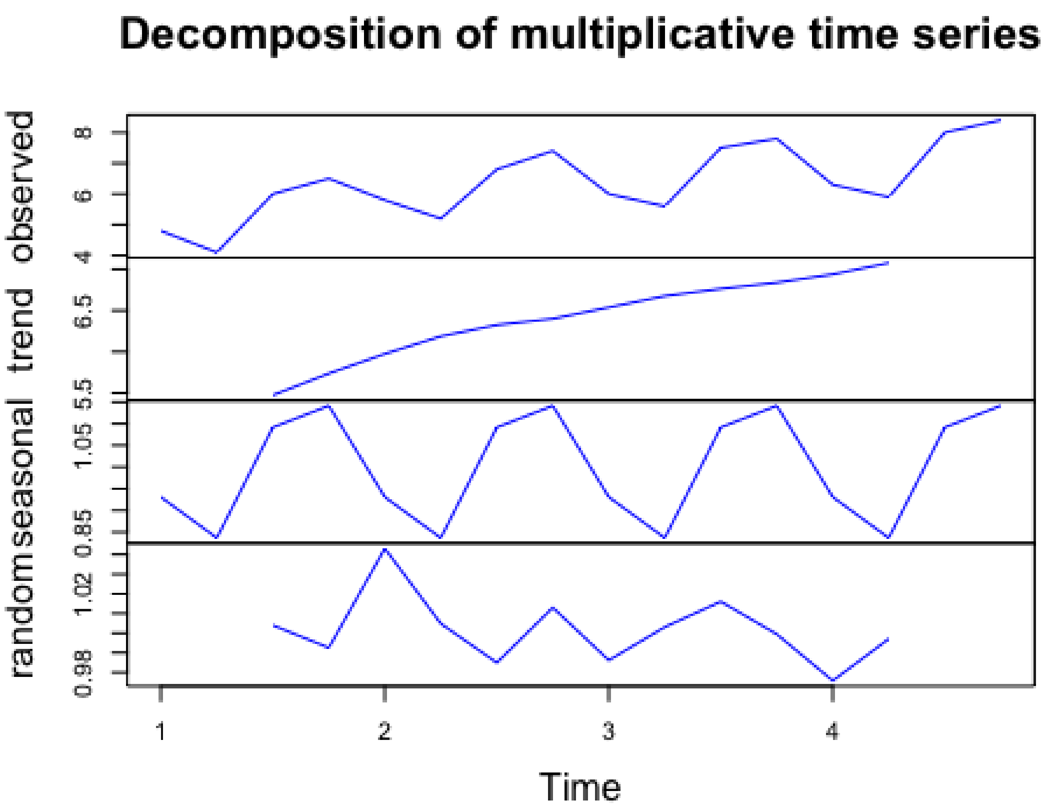


Descomposicón de serie aditiva y multiplicativa

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




```r
T2 = decompose(x, type="m")
plot(T2, col ="blue")
```



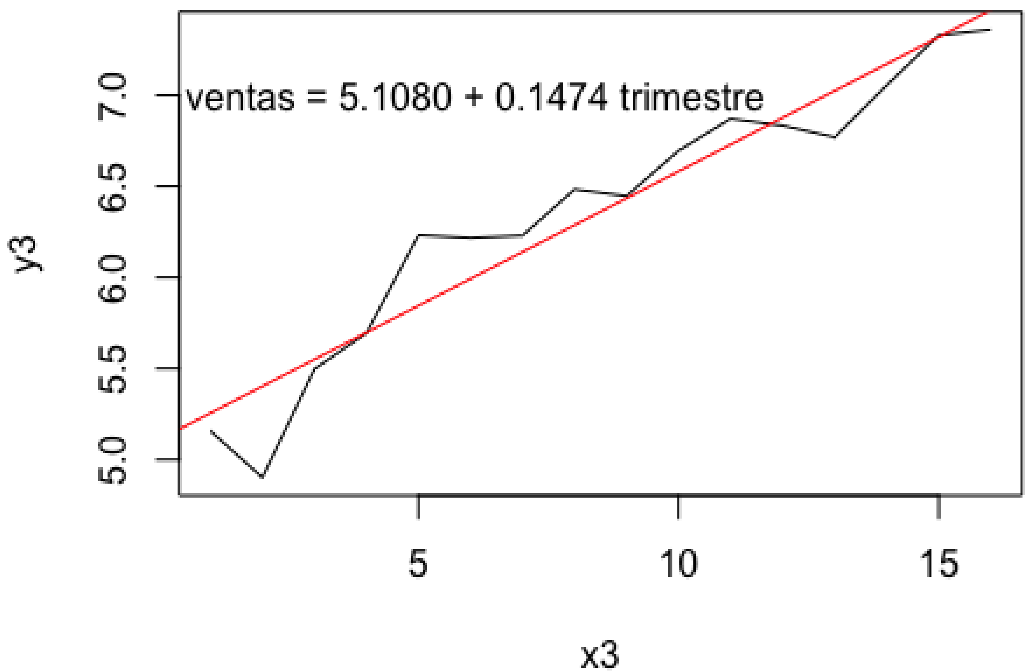
Calcula los índices estacionales y grafica la serie desestacionalizada

```
T2$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
```

Analiza el modelo lineal de la tendencia

```
ventas_desestacionalizadas = (T2$x)/(T2$seasonal)
x3 = 1:16
y3 = ventas_desestacionalizadas
N3 = lm(y3~x3)
N3

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

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



Pronóstico para el siguiente año

```r
f = function(x) {5.1080 + 0.1474*x}
# Los índices estacionales son:
a1 = T2$seasonal[1]
a2 =T2$seasonal[2]
a3 = T2$seasonal[3]
a4 = T2$seasonal[4];
f(17)*a1*1000

## [1] 7085.872

f(18)*a2*1000

## [1] 6491.284

f(19)*a3*1000

## [1] 8632.585

f(20)*a4*1000
```

```
## [1] 9195.263
```

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

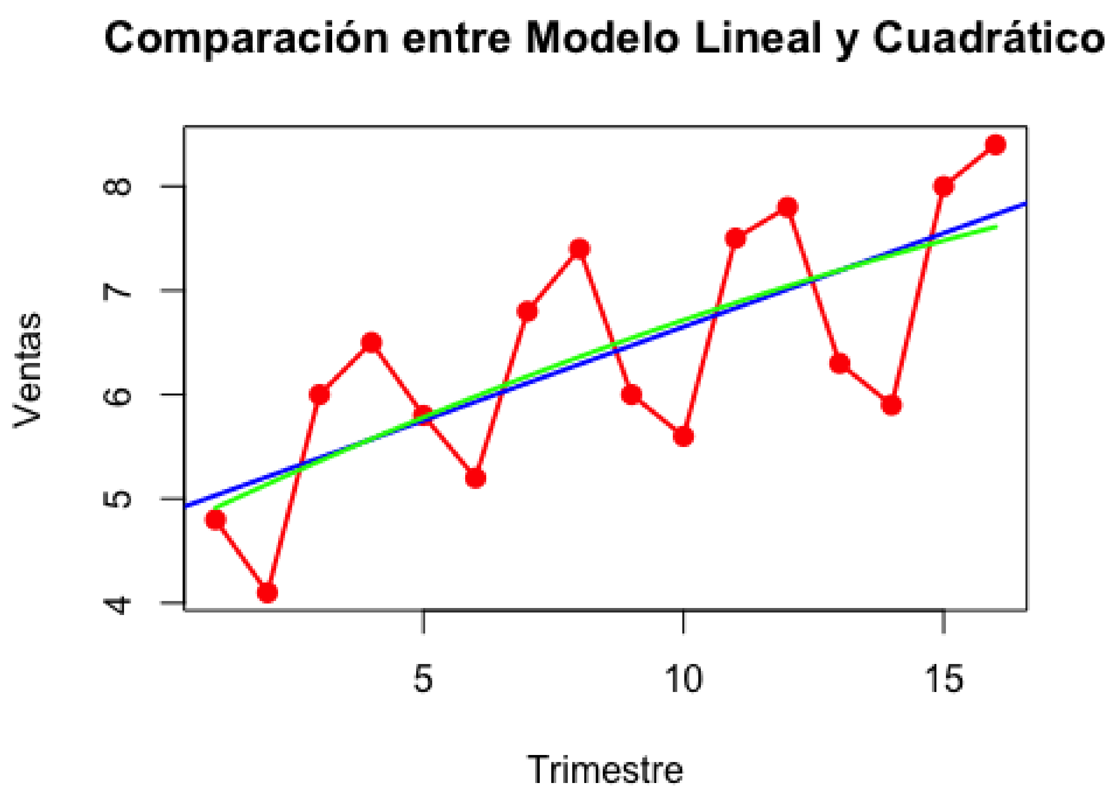
```
modelo = lm(x ~ tiempo)
predicciones = predict(modelo)
cme = mean((x - predicciones)^2)
cme

## [1] 0.672898

epam = mean(abs((x - predicciones) / x)) * 100
epam

## [1] 12.16897
```

Including Plots

```
library(tseries)

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

adf.test(ventas, k=0)

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

R Markdown

```
plot(tiempo, ventas, col = "red", type = "o", lwd = 2, pch = 19, ylab =
"Ventas", xlab = "Trimestre")
title("Comparación entre Modelo Lineal y Cuadrático")
modelol <- lm(ventas ~ tiempo)
abline(modelol, col = "blue", lwd = 2)
modeloc <- lm(ventas ~ tiempo + I(tiempo^2))
tiempo_pred <- seq(1, 16, length.out = 100)
predc <- predict(modeloc, newdata = data.frame(tiempo = tiempo_pred))
lines(tiempo_pred, predc, col = "green", lwd = 2)
```




```
pred_lineal <- predict(modelol)
pred_cuadratico <- predict(modeloc)
```

Including Plots

```
cme_lineal <- mean((ventas - pred_lineal)^2)
epam_lineal <- mean(abs((ventas - pred_lineal) / ventas)) * 100
cme_cuadratico <- mean((ventas - pred_cuadratico)^2)
epam_cuadratico <- mean(abs((ventas - pred_cuadratico) / ventas)) * 100

cat("CME (Modelo lineal):", cme_lineal, "\n")

## CME (Modelo lineal): 0.672898

cat("EPAM (Modelo lineal):", epam_lineal, "%\n")

## EPAM (Modelo lineal): 12.16897 %

cat("CME (Modelo cuadrático):", cme_cuadratico, "\n")

## CME (Modelo cuadrático): 0.6687373

cat("EPAM (Modelo cuadrático):", epam_cuadratico, "%\n")

## EPAM (Modelo cuadrático): 12.03663 %
```