Series de Tiempo no Estacionarias

Iker Ledesma - A01653115

17 de noviembre de 2022

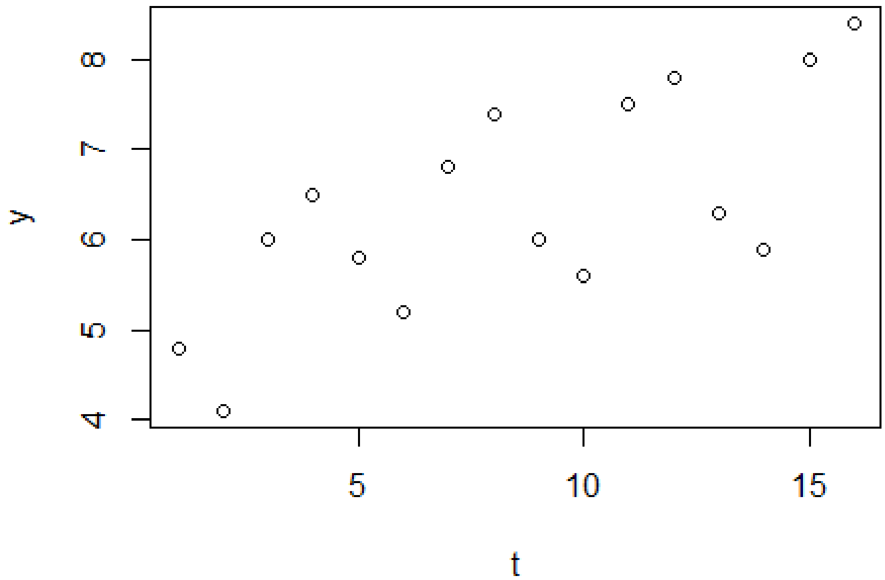
Base de Datos

```
df = data.frame(
  "trimestre" = 1:16,
  "Ventas (miles)" = 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)
)
t = 1:16
y = 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)
df
```

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

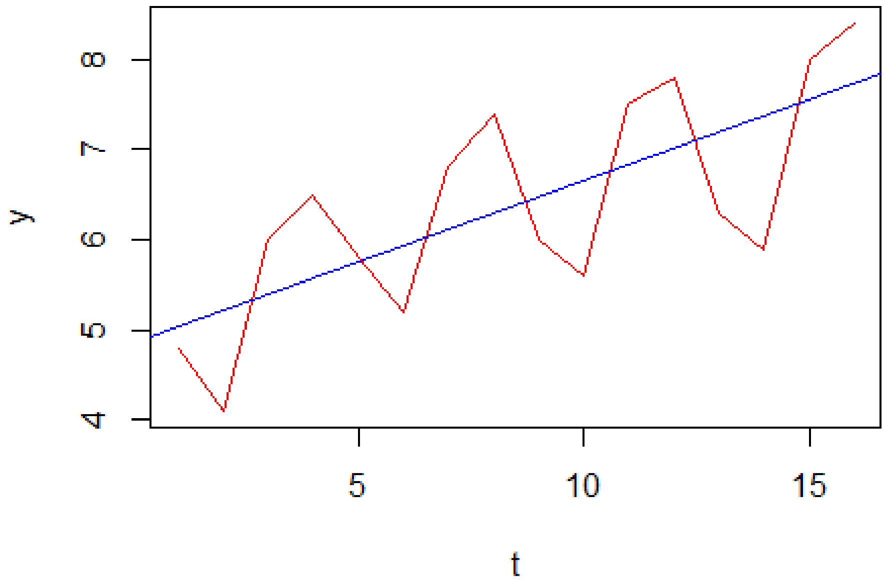
Gráfica de dispersión

```
plot(t, y)
```



Gráfica lineal

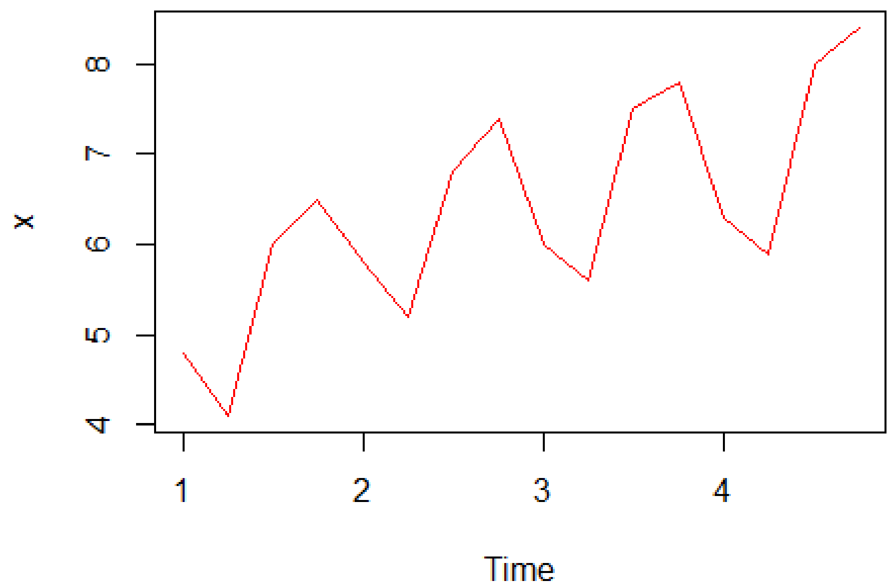
```
N = lm(y~t)
plot(t, y, type= "l", col = "red")
abline(N, col = "blue")
```



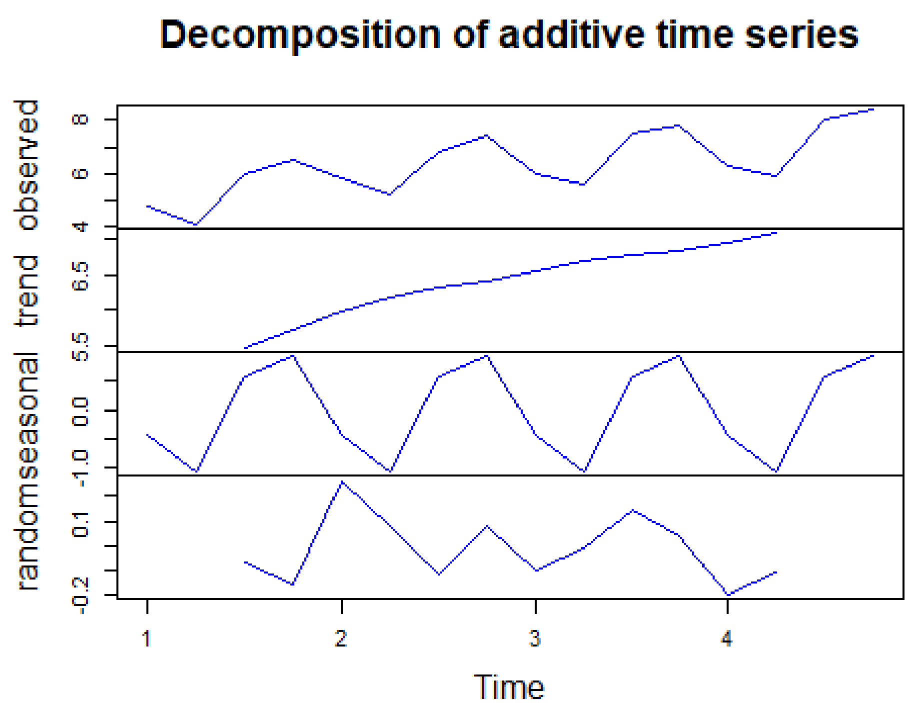

```
N

##
## Call:
## lm(formula = y ~ t)
##
## Coefficients:
## (Intercept)            t
##      4.8525       0.1799
```

Descompocisión

```
x= ts(y, frequency = 4, start(c(2016,1)))
plot.ts(x, col = "red")
```




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

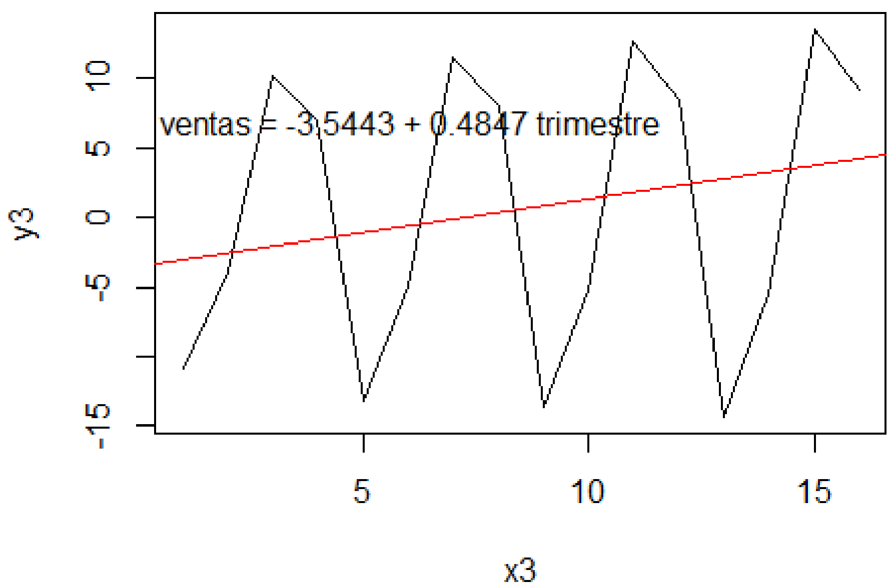


Modelo Lineal de la Tendencia

```
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
##     -3.5443       0.4847
```

```
plot(x3, y3, type = "l")
abline(N3, col = "red")
text(6, 7, " ventas = -3.5443 + 0.4847 trimestre")
```

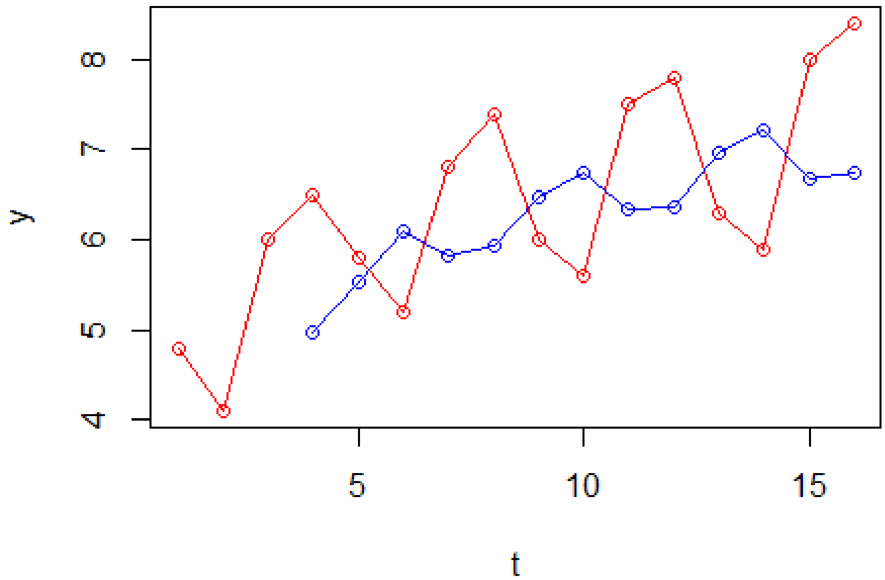



```
T$seasonal
```

```
##          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
```

Cálculo de CME y EPAM

```
n = 16
p1 = NA
e1 = NA
x = 1:16


for(i in 1:(n-3)){p1[i+3]=(y[i]+y[i+1]+y[i+2])/3; e1[i+3] = p1[i+3] -
y[i+3]}

CME1=mean(e1^2,na.rm=TRUE)
CME1

## [1] 1.378889

EPAM = mean(abs((y-p1)/y)) * 100
EPAM

## [1] NA

T1=data.frame(t,p1,y,e1^2)
T1

##     t       p1   y      e1.2
## 1   1       NA 4.8         NA
## 2   2       NA 4.1         NA
## 3   3       NA 6.0         NA
## 4   4 4.966667 6.5 2.35111111
## 5   5 5.533333 5.8 0.07111111
## 6   6 6.100000 5.2 0.81000000
## 7   7 5.833333 6.8 0.93444444
## 8   8 5.933333 7.4 2.15111111
## 9   9 6.466667 6.0 0.21777778
## 10 10 6.733333 5.6 1.28444444
## 11 11 6.333333 7.5 1.36111111
## 12 12 6.366667 7.8 2.05444444
## 13 13 6.966667 6.3 0.44444444
## 14 14 7.200000 5.9 1.69000000
## 15 15 6.666667 8.0 1.77777778
## 16 16 6.733333 8.4 2.77777778

plot(t, y, type='o', col='red')
lines(x,p1[x],type='o',col='blue')
```



Predicción quinto año

```
f = function(x) {-3.5443 + 0.4847*x}
# Los ídices estacionales son:
a1  = T$seasonal[1]
a2 =T$seasonal[2]
a3  = T$seasonal[3]
a4  = T$seasonal[4];
f(17)*a1*1000

## [1] -2064.108

f(18)*a2*1000

## [1] -5536.446

f(19)*a3*1000

## [1] 3339.99

f(20)*a4*1000

## [1] 5650.037
```