Actividad 8 - Series de tiempo no estacionarias: Tendencia

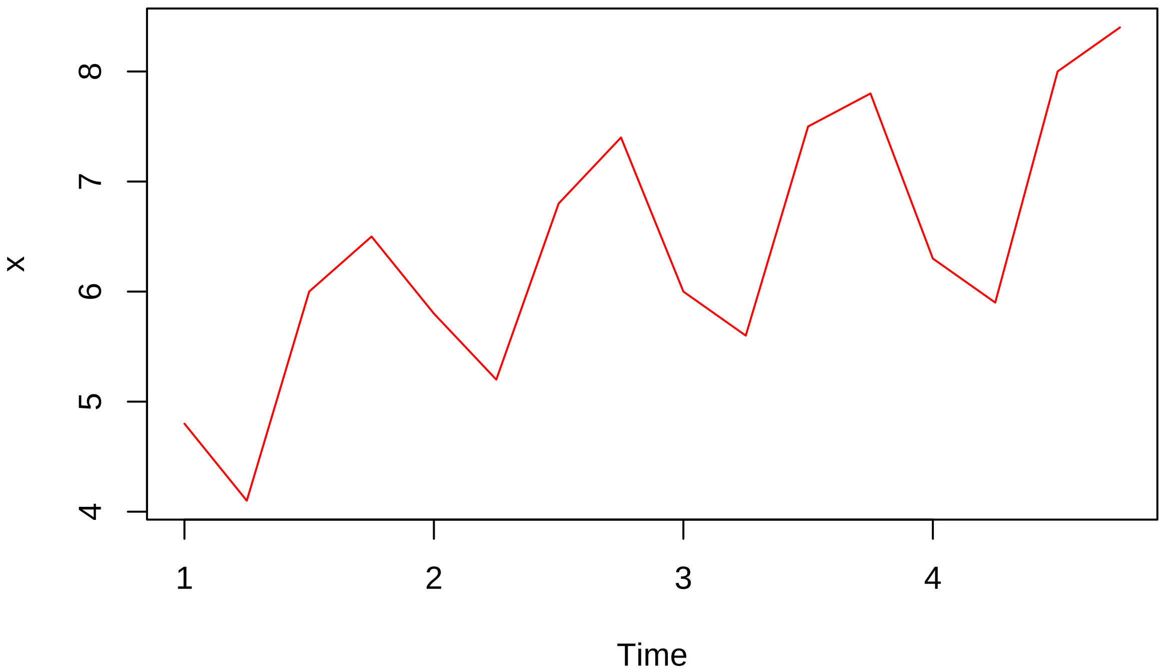
A00831314- Sofia Reyes

2023-11-14

Problema 1

Gráfico de Dispersión

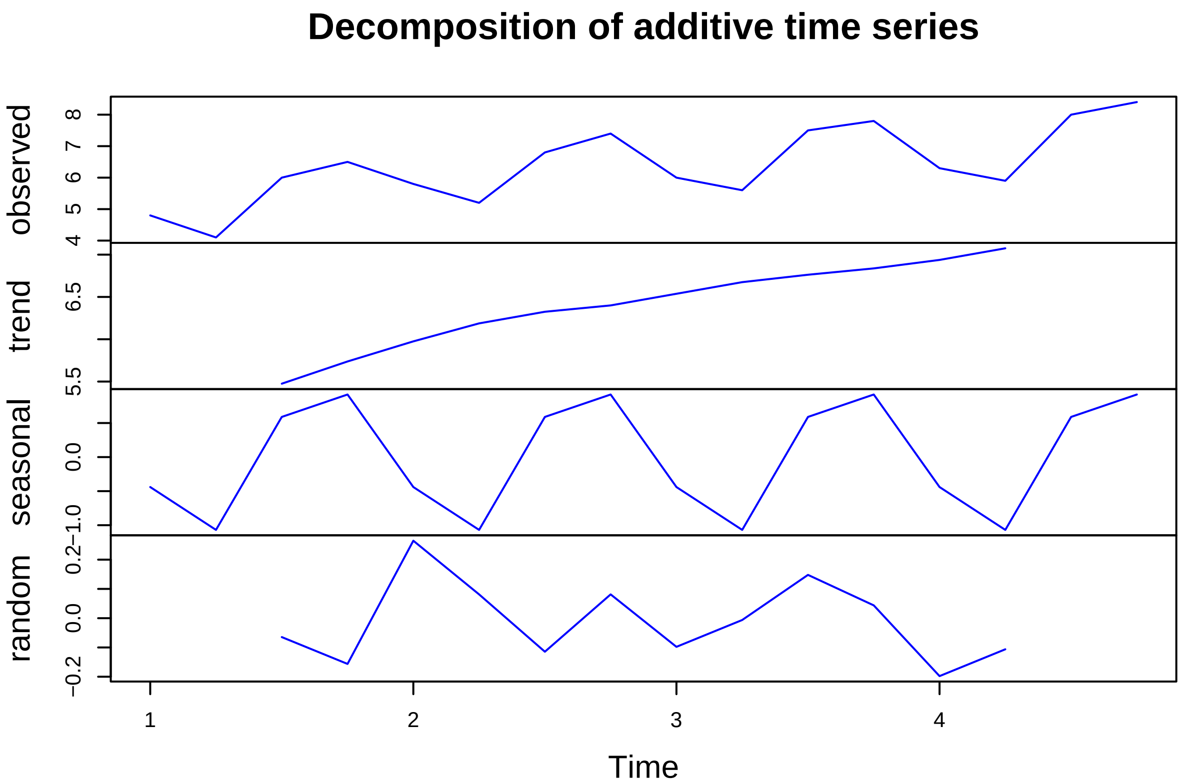
```
ventas = c(4.8, 4.1, 6.0, 6.5, 5.8, 5.2, 6.8, 7.4, 6.0, 5.6, 7.5, 7.8, 6.3, 5.9, 8.0, 8.4)
x = ts(ventas, frequency=4, start(c(2016,1)))
plot.ts(x, col = 'red')
```



Se puede observar una serie de tiempo no estacionaria, la cual tiene una tendencia positiva, ya que va aumentando la media año con año. Asi también existe una estacionalidad, donde la mayor venta de computadores se da en el último trimestre de cada año.

Análisis de tendencia y estacionalidad

```
T = decompose(x, type = 'a')
plot(T, col = 'blue')
```



La primera recta, muestra a la serie tal y como es, mientras que la segunda muestra la tendencia positiva que la serie de tiempo obtiene. Se puede observar como hasta la mitad del segundo año, hay un punto de inflexión, donde despúes de este último trimestre, la pendiente es menor. Finalmente, se observa la estacionalidad de los datos si es que la tendencia fuese nula, mostrando como en ciertos trimestres del año las ventas son mayores, y la fluctuación entre trimestres es muy parecida.

Modelo lineal de la tendencia

Desestacionalizacion

```
y1 = T$x/T$seasonal
x3 = 1:16
y3 = y1
N3 = lm(y3~x3)
N3
```

```
##
## Call:
## lm(formula = y3 ~ x3)
##
## Coefficients:
## (Intercept)           x3
##     -3.5443       0.4847
```

```
summary(N3)
```

***Análisis del modelo**

```
##
## 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
```

Al revisar el summary del modelo, se puede observar que los Betas no son significativos, al tener p-valores altos. Así también, la variabilidad explicada por la R cuadrada ajustada, es prácticamente nula. Por el lado del análisis de residuos, el error estandar residual es bastante alto, lo que significa que el modelo tiene grandes residuos y por ende no se está ajustando correctamente a la serie de tiempo.

Prueba de normalidad H0: los datos provienen de población normal H1: los datos no provienen de población normal alfa = 0.05 regla de decisión: si valor p< alfa, se rechaza H0

```r
shapiro.test(N3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  N3$residuals
## W = 0.90397, p-value = 0.09308
```

Al tener un p-valor mayor al de alfa, no se rechaza la Hipótesis nula y por ende los datos provienen de una población normal.

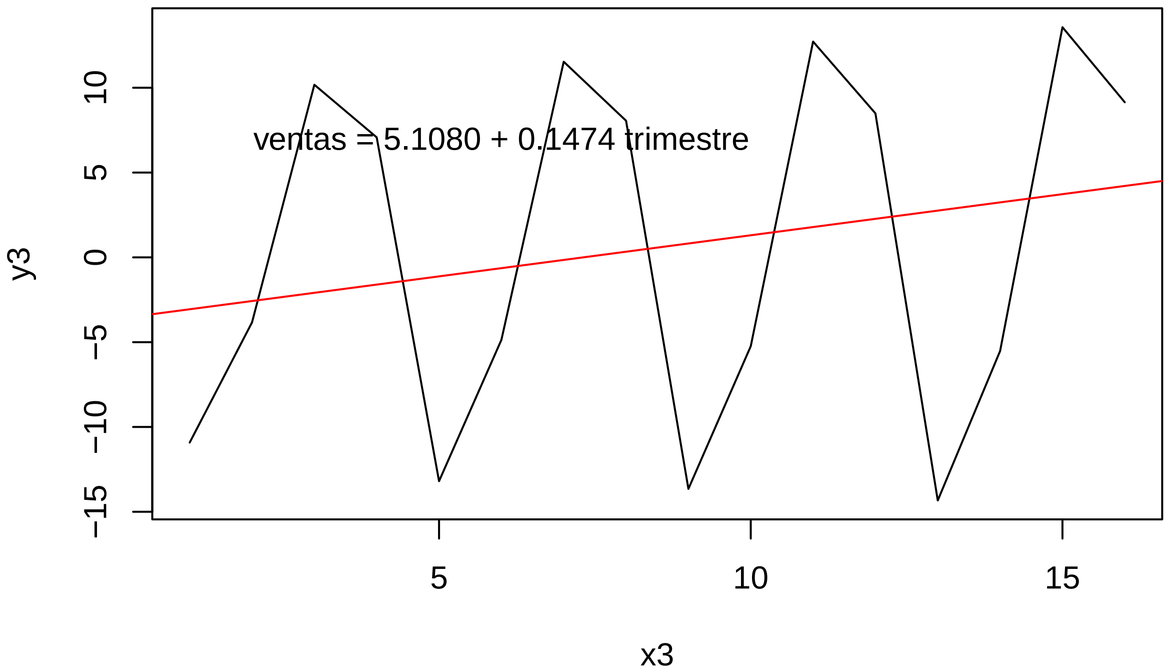
Cálculo del CME y el EPAM

```r
CME2=mean(N3$residuals^2,na.rm=TRUE)
CME2
```

```
## [1] 96.83152
```

Gráfico de Ventas vs Pronóstico

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



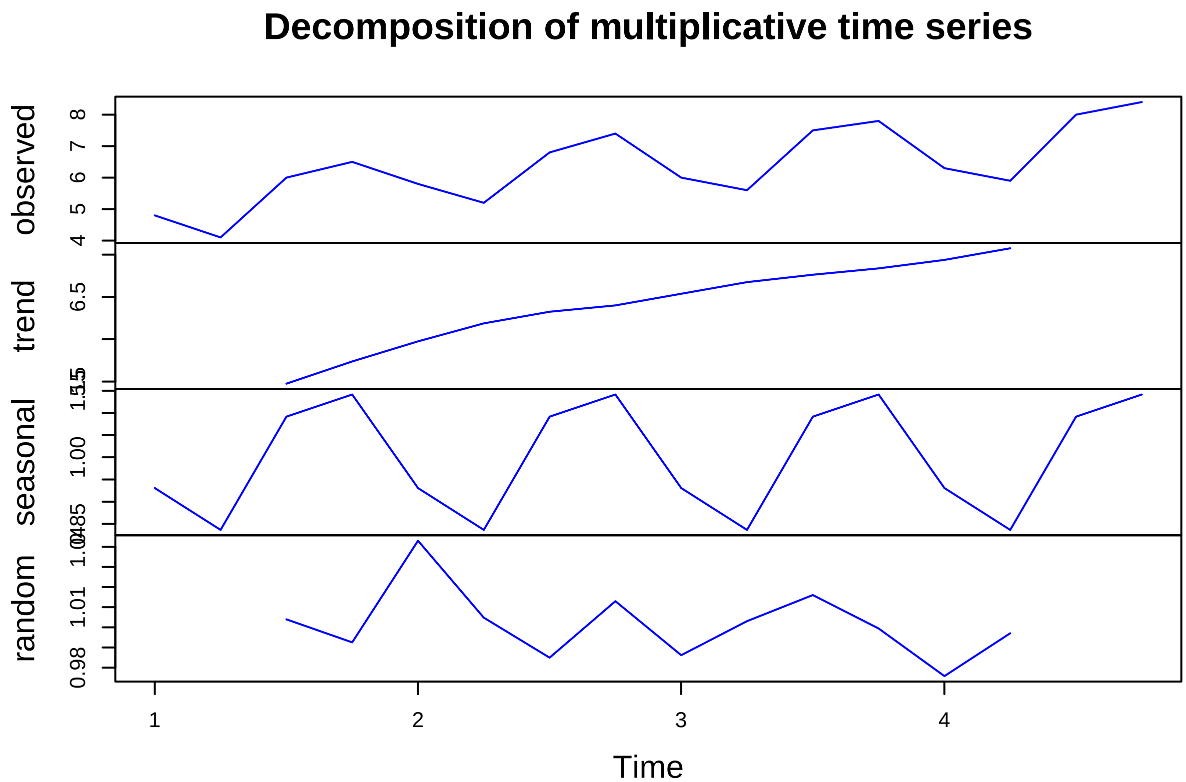
Concluye sobre el modelo: de acuerdo al análisis de verificación de los supuestos, ¿es el mejor modelo que puedes obtener?

Este no es el mejor modelo que se puede obtener ya que según el análisis realizado, este no se asemeja a los datos reales y sus predicciones generan residuos muy altos.

Propuesta de mejor modelo

Para generar un mejor modelo, se puede cambiar a un esquema multiplicativo:

```
T = decompose(x, type = 'm')
plot(T, col = 'blue')
```




```
y1 = T$x/T$seasonal
x3 = 1:16
y3 = y1
N3 = lm(y3~x3)
N3
```

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

```
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
```

Se puede ver como la R cuadrada aumenta sustancialmente.

Pronóstico siguiente año

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

```
## [1] 7085.872
```

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

```
## [1] 6491.284
```

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

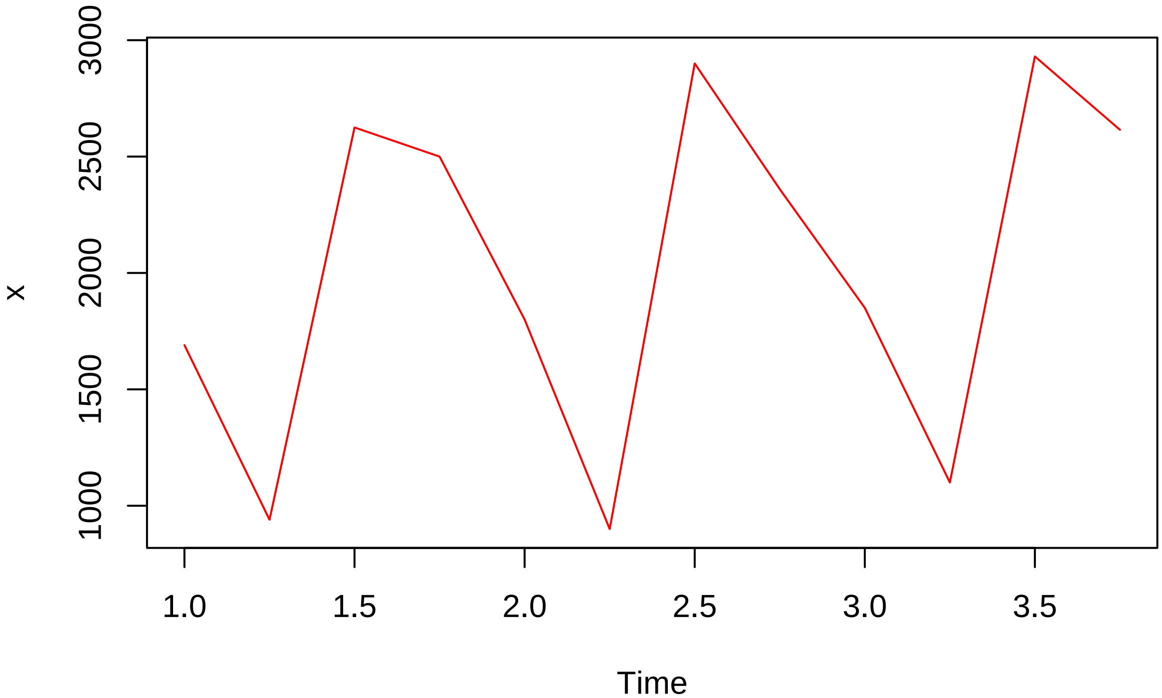
```
## [1] 8632.585
```

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

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

Problema 2

```
ventas = c(1690, 940, 2625, 2500, 1800, 900, 2900, 2360, 1850, 1100, 2930, 2615)
x = ts(ventas, frequency=4, start(c(2016,1)))
plot.ts(x, col = 'red')
```



##

Promedios móviles y promedios móviles centrados

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
k = 4
promedio_movil <- rollmean(ventas, k, fill = NA)
promedio_movil
```

```
##  [1]      NA 1938.75 1966.25 1956.25 2025.00 1990.00 2002.50 2052.50 2060.00
## [10] 2123.75      NA      NA
```

```
k = 2
promedio_movil_c <- rollmean(promedio_movil, k, fill = NA)
promedio_movil_c
```

```
##  [1]       NA 1952.500 1961.250 1990.625 2007.500 1996.250 2027.500 2056.250
##  [9] 2091.875       NA       NA       NA
```