A8-Series de tiempo no estacionarias. Tendencia

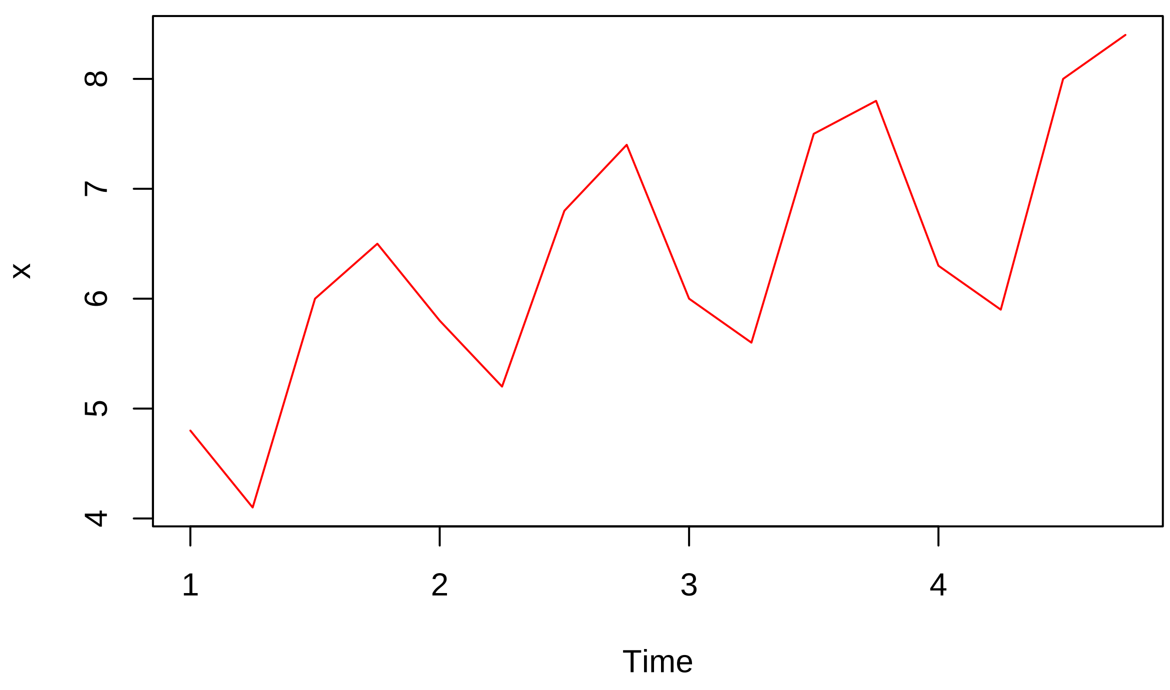
José Romo - A01197772

2023-11-14

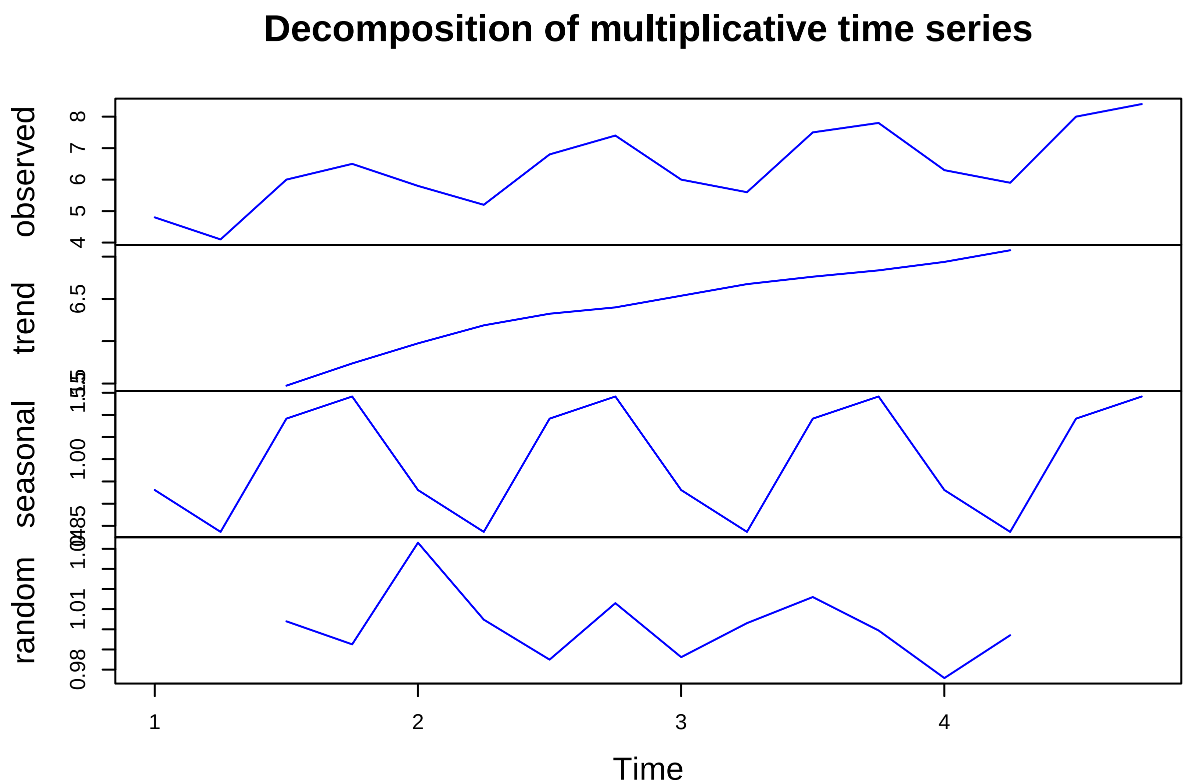
```
Ventas = c(4.8, 4.1, 6.0, 6.5, 5.8, 5.2, 6.8, 7.4, 6.0, 5.6, 7.5, 7.8, 6.3, 5.9, 8.0, 8.4)
```

Descomponsicion de la serie

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




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



Analisis de modelo lineal de la tendencia

Regresion lineal de la tendencia

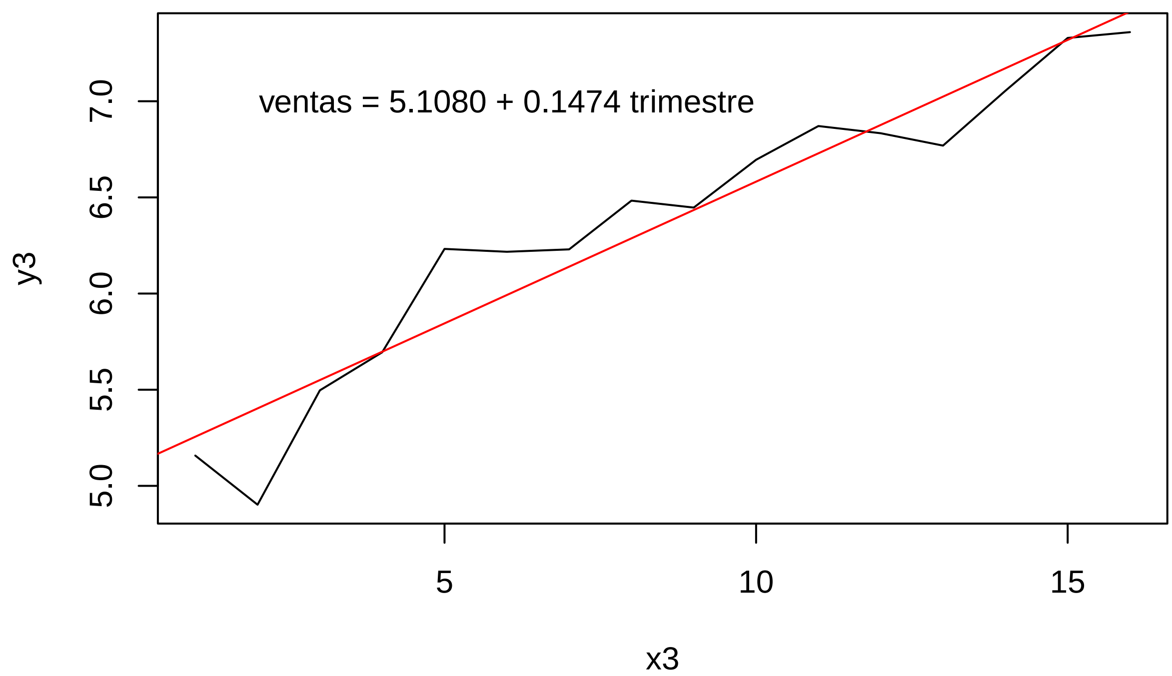
```
T$seasonal
```

```
##        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
```

```
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
```

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




```r
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
```

```r
f = function(x) {5.1080 + 0.1474*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] 7085.872
```

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

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

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

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

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

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

Significacion de Beta_1

```r
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
```

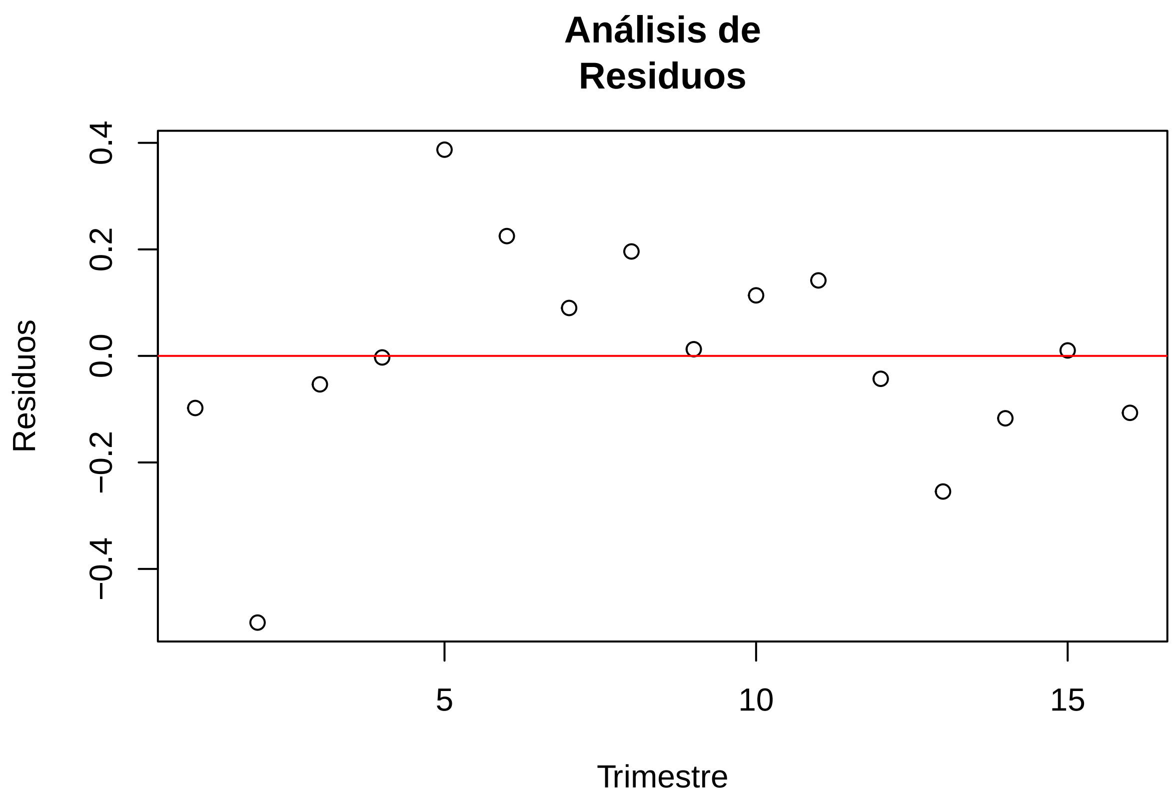
Variabilidad explicada

```r
cd = 1 - var(N3$residuals)/var(y3)
cd
```

```
## [1] 0.9207911
```

Analisis de los residuos

```r
plot(N3$residuals, ylab = "Residuos", xlab = "Trimestre", main = "Análisis de
Residuos")
abline(h = 0, col = "red")
```



Prueba de normalidad

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

```
##
##  Shapiro-Wilk normality test
##
## data:  N3$residuals
## W = 0.96379, p-value = 0.7307
```

Calculo del CME

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

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

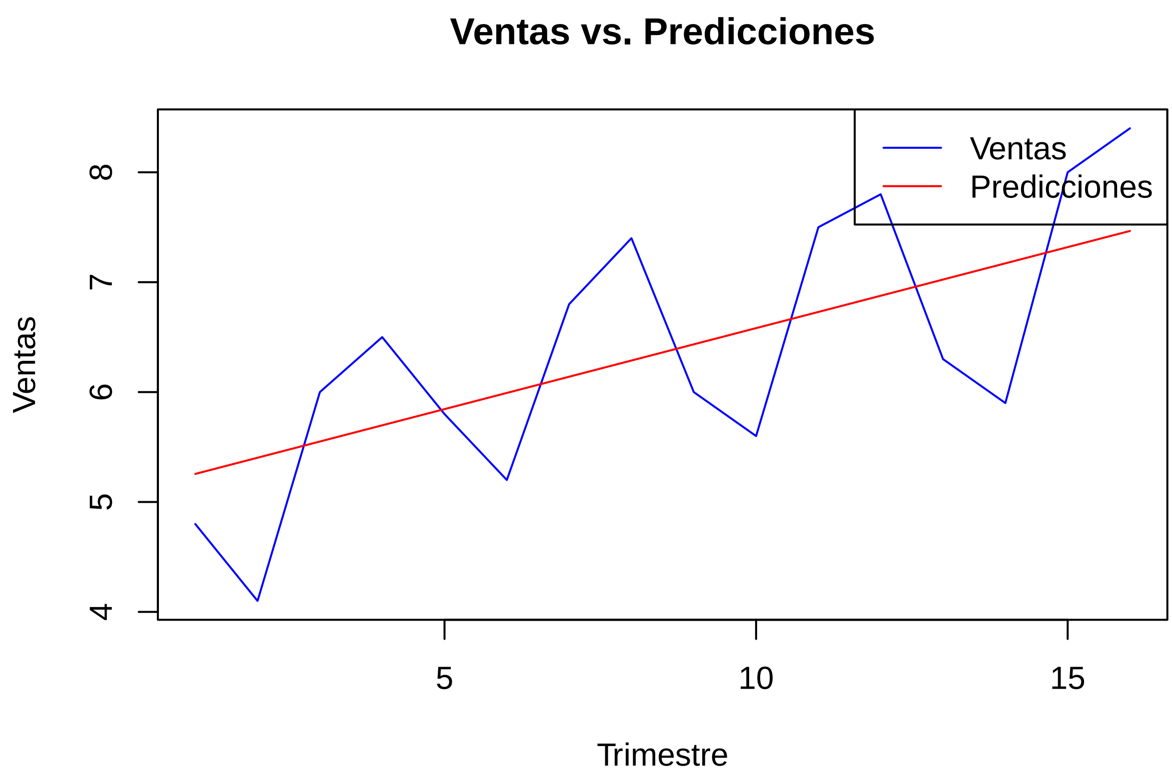
Calculo del EPAM

```
EPAM = mean(abs(N3$residuals/y3 * 100), na.rm = TRUE)
EPAM
```

```
## [1] 2.439533
```

Graficos de los valores de las ventas y las predicciones vs el tiempo

```
predVentas = predict(N3)
plot(Ventas, type = "l", col = "blue", ylab = "Ventas", xlab ="Trimestre",
main = "Ventas vs. Predicciones")
lines(predVentas, col = "red")
legend("topright", legend = c("Ventas", "Predicciones"), col = c("blue",
"red"), lty = 1)
```



Pronosticos del siguiente año

```
pronNextYear = 17:20
pronostico = predict(N3, newdata = data.frame(x3 = pronNextYear))
pronostico
```

```
##        1        2        3        4
## 7.613536 7.760918 7.908300 8.055682
```

Un problemilla mas

```
Ventas2 <- data.frame(
  Trimestre = rep(1:4, each = 3),
  A1 = c(1690, 940, 2625, 2500),
  A2 = c(1800, 900, 2900, 2360),
  A3 = c(1850, 1100, 2930, 2615)
)
```

Promedio moviles de 4 trimestres

```
Ventas2$PromedioMovil4 <- rowMeans(Ventas2[, c("A1", "A2", "A3")])
```

Promedio moviles centrados

```
Ventas2$PromedioMovilCentrado <- (c(NA, head(Ventas2$PromedioMovil4, -1)) + c(tail(Ventas2$PromedioMovil
```

Visualozar datos

```
Ventas2
```

```
##    Trimestre   A1   A2   A3 PromedioMovil4 PromedioMovilCentrado
## 1          1 1690 1800 1850       1780.000                    NA
## 2          1  940  900 1100        980.000              2299.167
## 3          1 2625 2900 2930       2818.333              1735.833
## 4          2 2500 2360 2615       2491.667              2299.167
## 5          2 1690 1800 1850       1780.000              1735.833
## 6          2  940  900 1100        980.000              2299.167
## 7          3 2625 2900 2930       2818.333              1735.833
## 8          3 2500 2360 2615       2491.667              2299.167
## 9          3 1690 1800 1850       1780.000              1735.833
## 10         4  940  900 1100        980.000              2299.167
## 11         4 2625 2900 2930       2818.333              1735.833
## 12         4 2500 2360 2615       2491.667                    NA
```

Indicies estacionales

```
indices_estacionales <- c(
  mean(Ventas2$A1) / mean(Ventas2$PromedioMovilCentrado[1:3]),
  mean(Ventas2$A2) / mean(Ventas2$PromedioMovilCentrado[4:6]),
  mean(Ventas2$A3) / mean(Ventas2$PromedioMovilCentrado[7:9])
)
```

Grafico

```
indices_estacionales
```

```
## [1]        NA 0.9425076 1.1040433
```

Trimestre con mayor indice estacional

```r
trimestre_max_estacional <- which.max(indices_estacionales)
```

Visaulizar resultados

```r
trimestre_max_estacional
```

```
## [1] 3
```