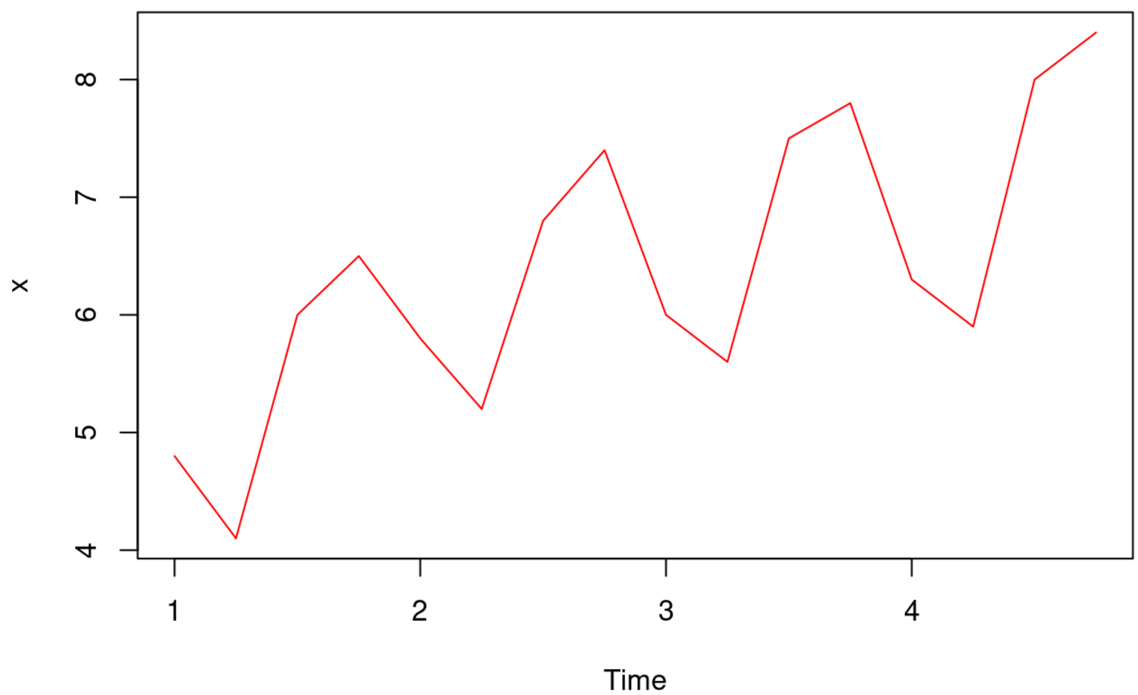
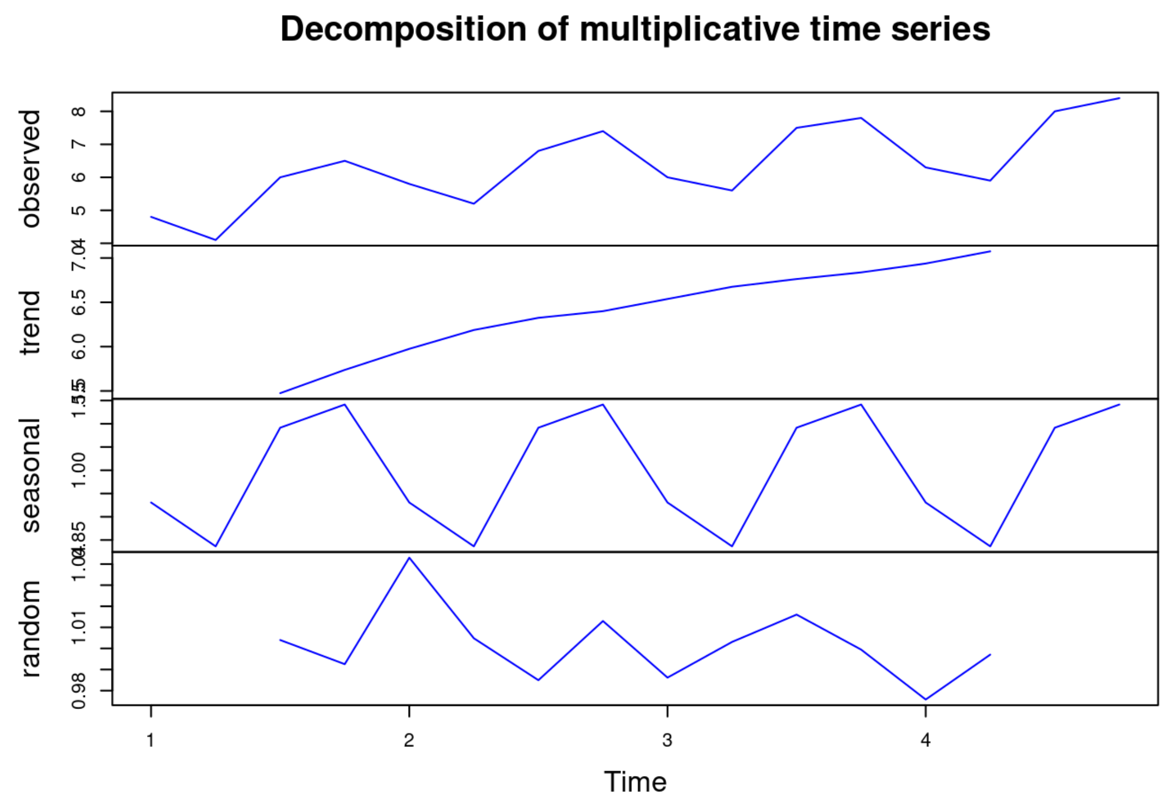
seriesp2

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




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



La primer gráfica es de los datos observados. La segunda muestra la tendencia, que en este caso es creciente. La tercera son los ciclos, donde se observa claramente que hay un comportamiento ciclico y la ultima grafica son los residuos.

Indices estacionales

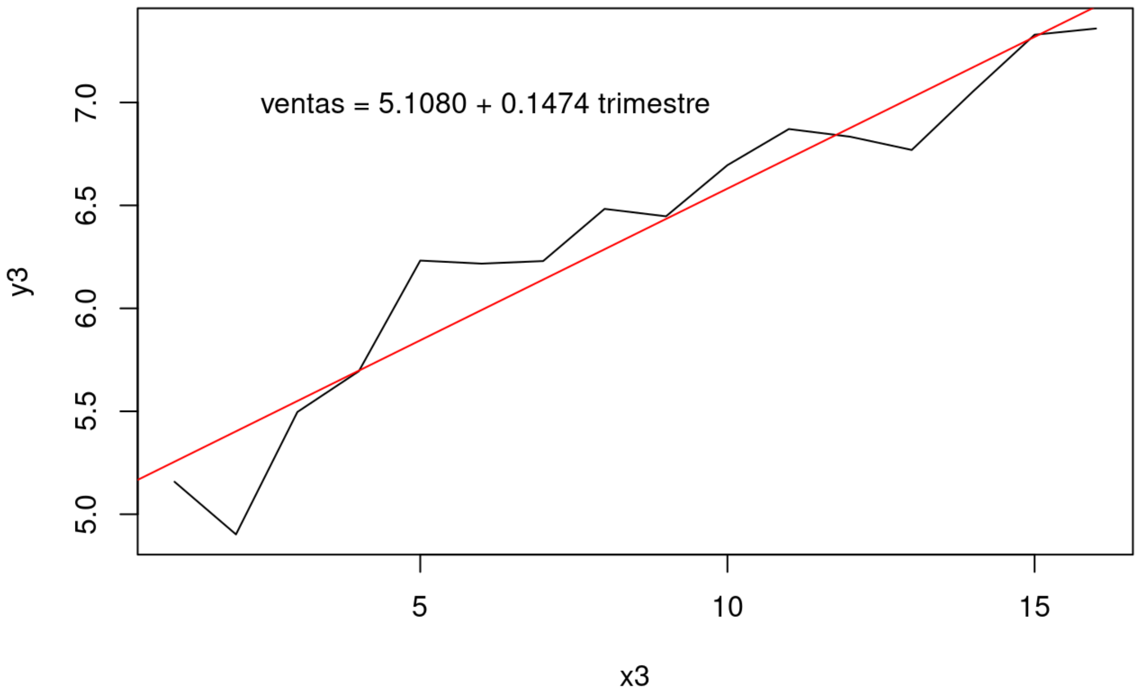
```
T$seasonal
```

```
##        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
```

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



En la grafica se observa que hay residuos tanto positivos como negativos.

```
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
```

El resumen del modelo indica que tanto el intercepto como B1 son significantes para este modelo.

Predicciones

```
e=NA
g=NA
f = function(x) 5.1080 + 0.1474*x
for(i in 1:16){
  g[i]=f(i)*T$seasonal[i]
  e[i]=ventas[i]-g[i]
}
CME_tendencia=mean(e^2,na.rm="TRUE")
cat("El CME del metodo de proyeccion de tendencia es:", CME_tendencia,"\n")
```

```
## El CME del metodo de proyeccion de tendencia es: 0.03302078
```

En este primer modelo se obtuvo un error CME de 0.033 lo cual es un error pequeño.

La grafica de las predicciones y los valores observados muestra que el modelo hace estimaciones bastante cercanas a la realidad.

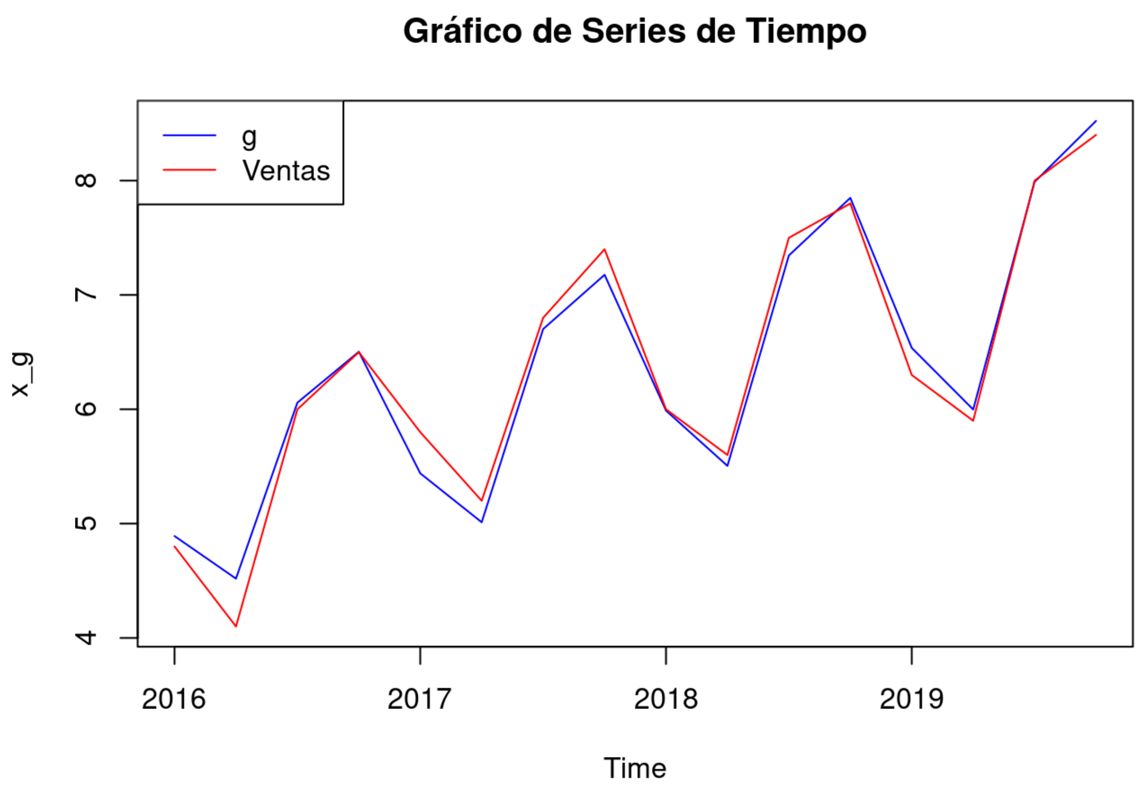
```
#library(ggplot2)

# Supongamos que tienes las variables definidas
#g <- c(2, 4, 3, 5, 6, 7, 5, 4, 6, 8, 7, 6, 5, 4, 3, 2)
#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)

# Crear series temporales
x_g <- ts(g, frequency = 4, start = c(2016, 1))
x_ventas <- ts(ventas, frequency = 4, start = c(2016, 1))

# Graficar en la misma gráfica
plot.ts(x_g, col = "blue", ylim = range(c(g, ventas)), main = "Gráfico de Series de Tiempo")
lines(x_ventas, col = "red")

# Agregar leyenda
legend("topleft", legend = c("g", "Ventas"), col = c("blue", "red"), lty = 1)
```



Normalidad alpha:

0.05

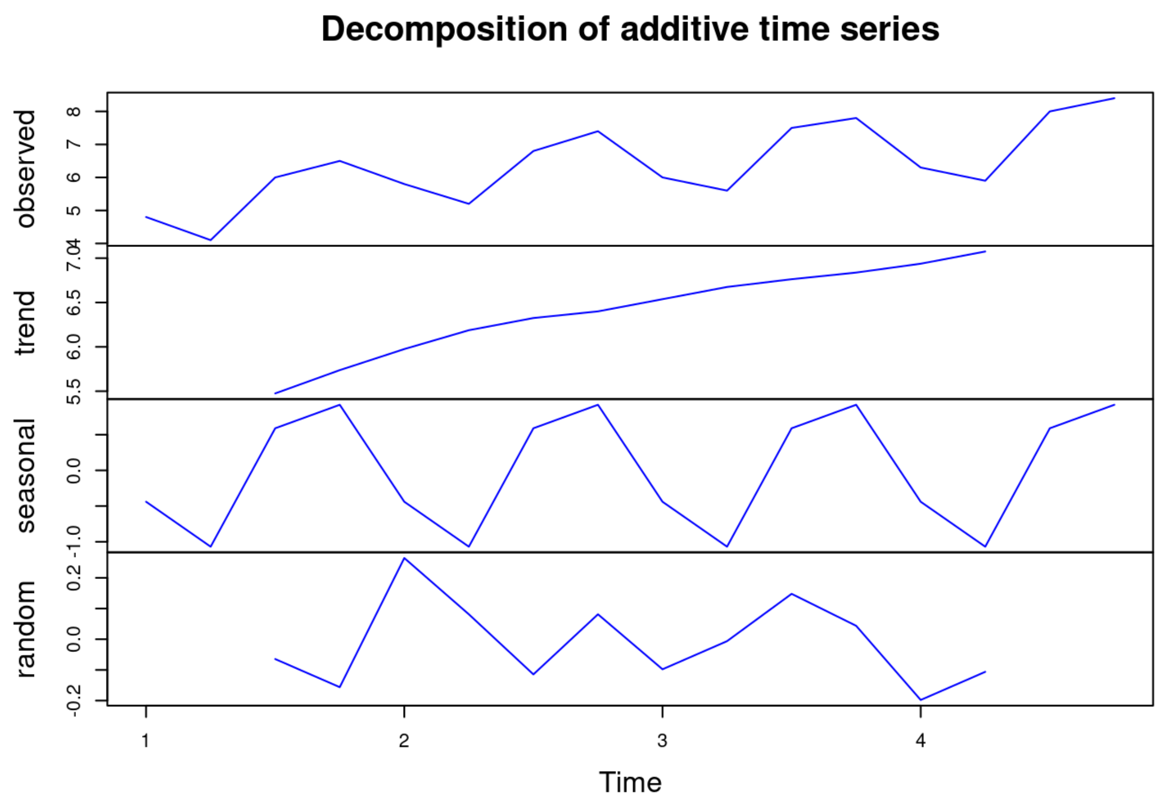
```
shapiro.test(N3$residuals)
```

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

Como el p-value es mayor a alpha entoces no se rechaza H0 y podemos concluir que los residuos si siguen una distribucion normal.

modelo 2

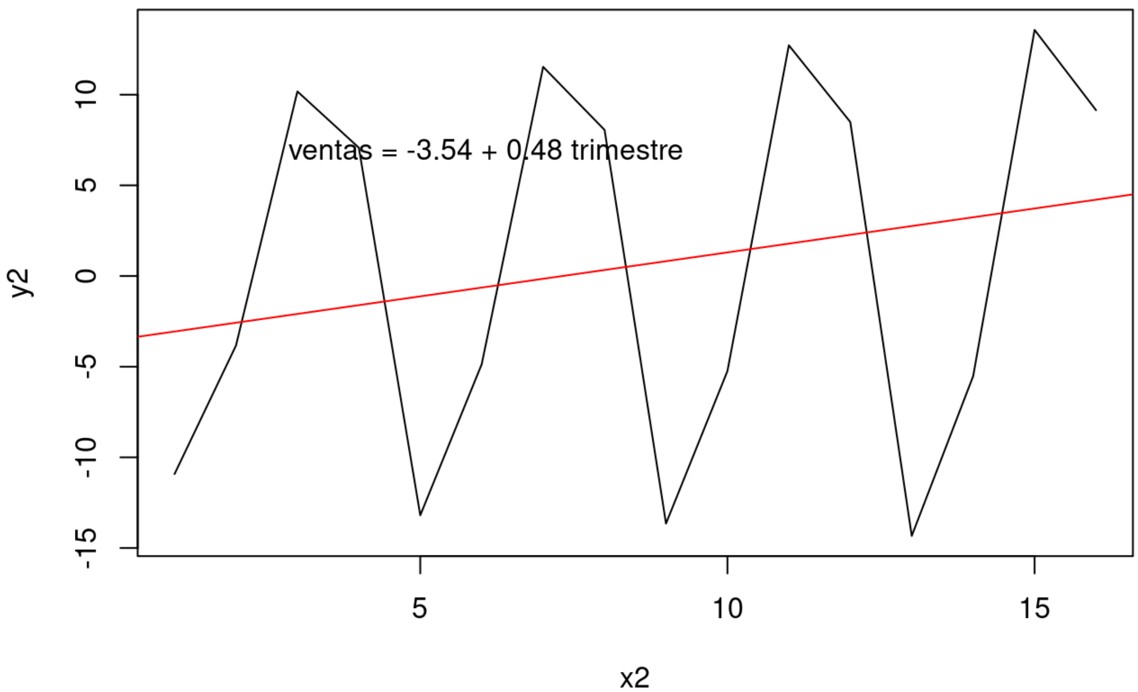
```
T2 = decompose(x)
plot(T2, col ="blue")
```




```
ventas_desestacionalizadas = (T2$x)/(T2$seasonal)
x2 = 1:16
y2 = ventas_desestacionalizadas
N2 = lm(y2~x2)
N2
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## Coefficients:
## (Intercept)           x2
##     -3.5443       0.4847
```

```
plot(x2, y2, type = "l")
abline(N2, col = "red")
text(6, 7, " ventas = -3.54 + 0.48 trimestre")
```




```
summary(N2)
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## 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
## x2            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
```

Como observamos en el resumen este segundo modelo donde se usa el aditivo no es significativo en B1.

```
e2=NA
g2=NA
f2 = function(x) -3.54 + 0.48*x
for(i in 1:16){
  g2[i]=f2(i)*T2$seasonal[i]
  e2[i]=ventas[i]-g2[i]
}
CME_tendencia=mean(e2^2,na.rm="TRUE")
cat("El CME del metodo de proyeccion de tendencia es:", CME_tendencia,"\n")
```

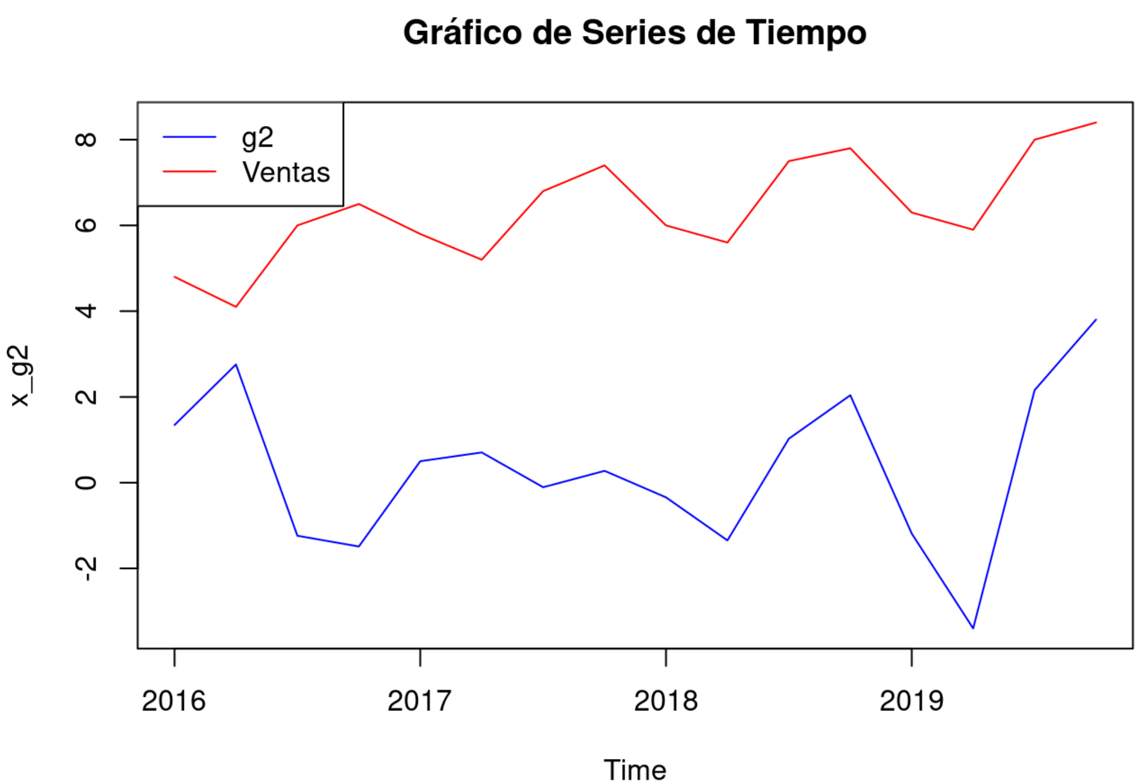
```
## El CME del metodo de proyeccion de tendencia es: 39.87707
```

Calculando el error CME confirmamos que el modelo no es bueno, ya que tiene un error bastante grande

```
x_g2 <- ts(g2, frequency = 4, start = c(2016, 1))
x_ventas <- ts(ventas, frequency = 4, start = c(2016, 1))

# Graficar en la misma gráfica
plot.ts(x_g2, col = "blue", ylim = range(c(g2, ventas)), main = "Gráfico de Series de Tiempo")
lines(x_ventas, col = "red")

# Agregar leyenda
legend("topleft", legend = c("g2", "Ventas"), col = c("blue", "red"), lty = 1)
```



Finalmente con la grafica concluimos que el segundo modelo no es bueno para hacer estimaciones.