A8 - Series de Tiempo No Estacionarias

Jorge Eduardo de León Reyna - A00829759

```
1 install.packages('Metrics')
2 install.packages('forecast')
```

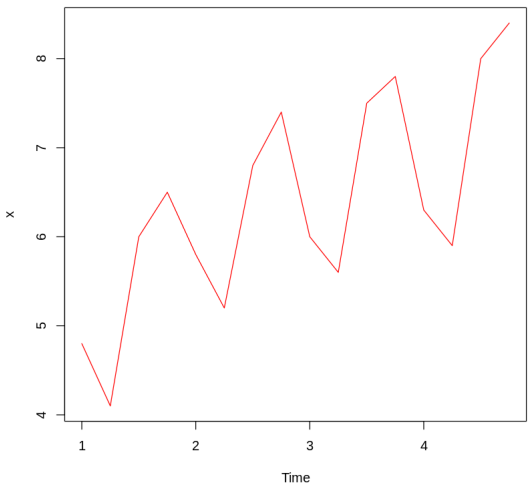
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘xts’, ‘TTR’, ‘quadprog’, ‘quantmod’, ‘fracdiff’, ‘lmtest’, ‘Rcpp’, ‘timeDate’, ‘tseries’,
```

```
1 install.packages('forecastHybrid')
2 install.packages('gbm')
3 install.packages('gnnforbm')
```

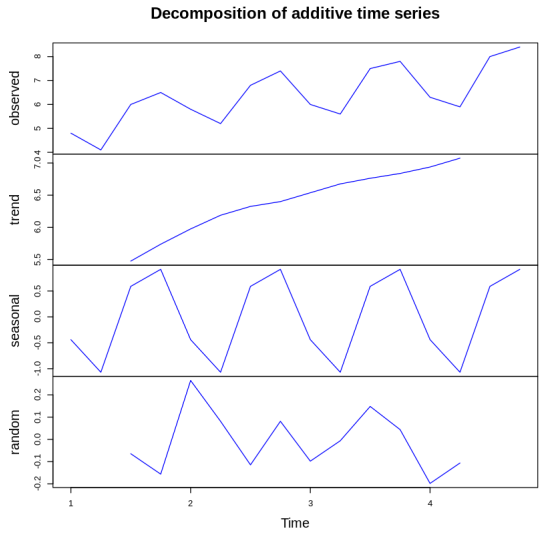
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘SparseM’, ‘RcppEigen’, ‘hts’, ‘iterators’, ‘thief’, ‘doParallel’, ‘foreach’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Warning message:
“package ‘gnnforbm’ is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages”
```

Problema 1

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



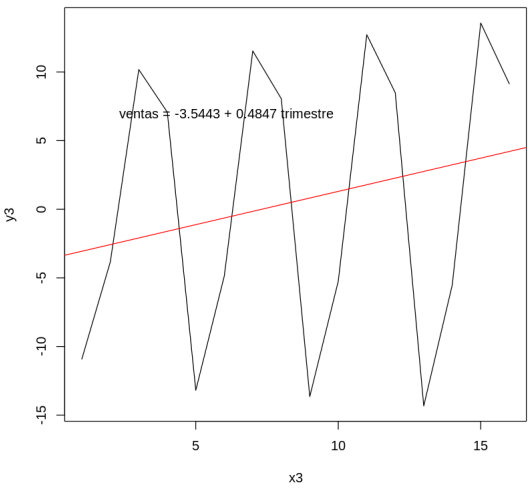
```
1 T = decompose(x)
2 plot(T, col ="blue")
```




```
1 ventas_desestacionalizadas = (T$x)/(T$seasonal)
2 x3 = 1:16
3 y3 = ventas_desestacionalizadas
4 N3 = lm(y3~x3)
5 N3
```

```
Call:
lm(formula = y3 ~ x3)

Coefficients:
(Intercept)           x3
    -3.5443       0.4847
```

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



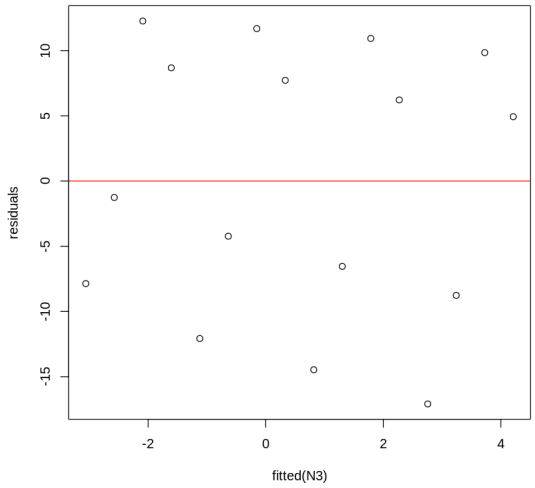

```
1 residuals <- residuals(N3)
2 summary(residuals)
```

```
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-17.088  -8.085   1.836   0.000   8.971  12.267
```

```
1 summary(N3)
```

```
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
```

```
plot(fitted(N3), residuals)
abline(h = 0, col = "red") # adds a horizontal line at 0
```




```
predictions <- predict(N3, newdata = y3)
library(Metrics)
```

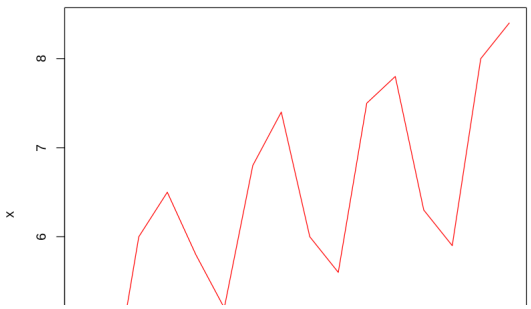
```
mape<-mape(y3, predictions)
mape
```

```
0.948857833283332
```

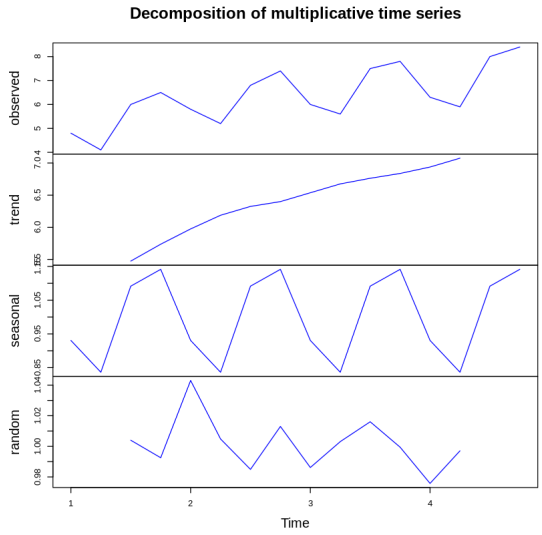
Notamos que los residuos generados por nuestro enfoque aditivo muestran niveles significativamente elevados. Específicamente, el modelo presenta un valor de p de 0.4099, indicando una elevada discrepancia, lo que desaconseja su utilización. Esta conclusión también se refleja directamente en el cálculo de la serie desestacionalizada, donde se observa una clara presencia de estacionalidad, a pesar de que se suponía que esta debería haber sido eliminada. Además, el error porcentual del 95% es excesivamente alto, motivando la exploración de otro modelo alternativo.

Problema 2

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




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




```
1 ventas_desestacionalizadas = (T$x)/(T$seasonal)
2 x3 = 1:16
3 y3 = ventas_desestacionalizadas
4 N3 = lm(y3~x3)
5 N3
```

```
Call:
lm(formula = y3 ~ x3)

Coefficients:
(Intercept)           x3
     5.1080       0.1474
```

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

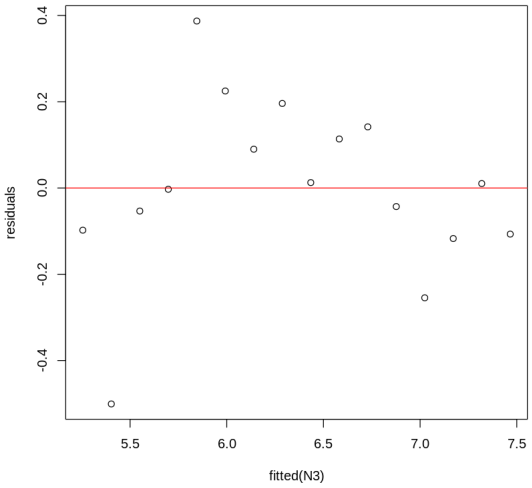

```
1 residuals <- residuals(N3)
2 summary(residuals)
```

```
      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
-0.500706 -0.100074  0.003699  0.000000  0.120706  0.387173 
```

```
1 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
```

```
1 plot(fitted(N3), residuals)
2 abline(h = 0, col = "red") # adds a horizontal line at 0
```




```
1 predictions <- predict(N3, newdata = y3)
2 library(Metrics)
3 mape<-mape(y3, predictions)
4 mape
```

```
0.024395328407222
```

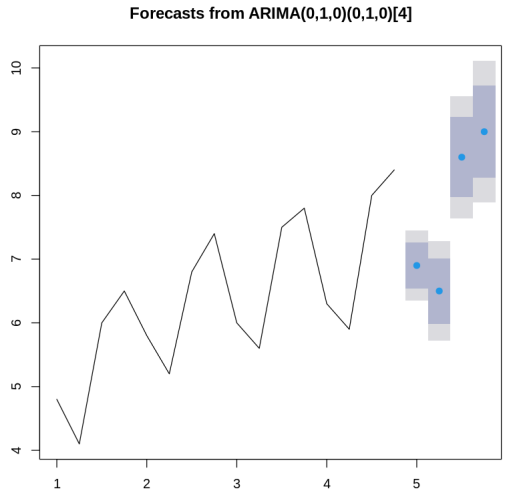
```
1 library(dplyr)
2 library(ggplot2)
3 library(forecast)
4 library(forecastHybrid)
5 library(gbm)
6 library(nnfor)
```

```
Error in library(forecastHybrid): there is no package called ‘forecastHybrid’
Traceback:

1. library(forecastHybrid)
```

```
1 sarima_ts<-auto.arima(x)
2 sarima_ts
```

```
Series: x
ARIMA(0,1,0)(0,1,0)[4]

sigma^2 = 0.08001:  log likelihood = -1.72
AIC=5.43   AICc=5.88   BIC=5.83
```

```
1  arima_model<-forecast::forecast(sarima_ts,h=4)
2 plot(arima_model)
```




```
1 fit<-nnetar(x,repeats=40,lambda=NULL)
2 fit
```

```
Series: x
Model:  NNAR(1,1,2)[4]
Call:   nnetar(y = x, repeats = 40, lambda = NULL)

Average of 40 networks, each of which is
a 2-2-1 network with 9 weights
options were - linear output units

sigma^2 estimated as 0.01048
```

```
1 nn_model<-forecast::forecast(fit,h=4)
2 #Plotting prediction and testing data (red for testing data)
3 plot(nn_model)
```

Problema 3

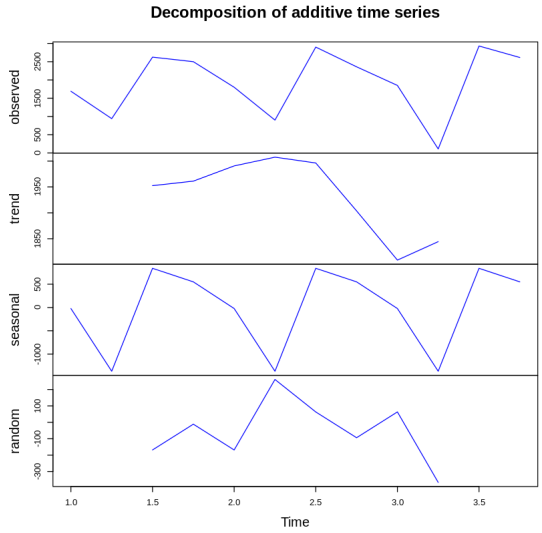
```
1 library(zoo)
```

```
Attaching package: ‘zoo’


The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric
```

```
1 ser = c(1690, 940, 2625, 2500, 1800, 900, 2900, 2360, 1850, 110, 2930, 2615) # Calculate the 4-period moving average
2 rollmean(ser, 4, fill = NA, align = "center")
```

<NA> · 1938.75 · 1966.25 · 1956.25 · 2025 · 1990 · 2002.5 · 1805 · 1812.5 · 1876.25 · <NA> · <NA>

```
1 x = ts(ser, frequency = 4)
2 T = decompose(x)
3 plot(T, col ="blue")
```




```
1 T$seasonal
```

A Time Series: 3 × 4

	Qtr1	Qtr2	Qtr3	Qtr4
1	-22.1875	-1368.4375	840.6250	550.0000
2	-22.1875	-1368.4375	840.6250	550.0000
3	-22.1875	-1368.4375	840.6250	550.0000

Notamos que el componente estacional más pronunciado corresponde al tercer trimestre, siendo su valor considerablemente elevado, alcanzando los 840.62. Este hallazgo resulta coherente, dado que este trimestre es consistentemente el período de mayor ingreso a lo largo de los tres años contemplados en nuestros datos.