# Ejercicio de Time Series - Clase 1

Daniel Ferreira Zanchetta y Lais Silva Almeida Zanchetta

24/05/2020

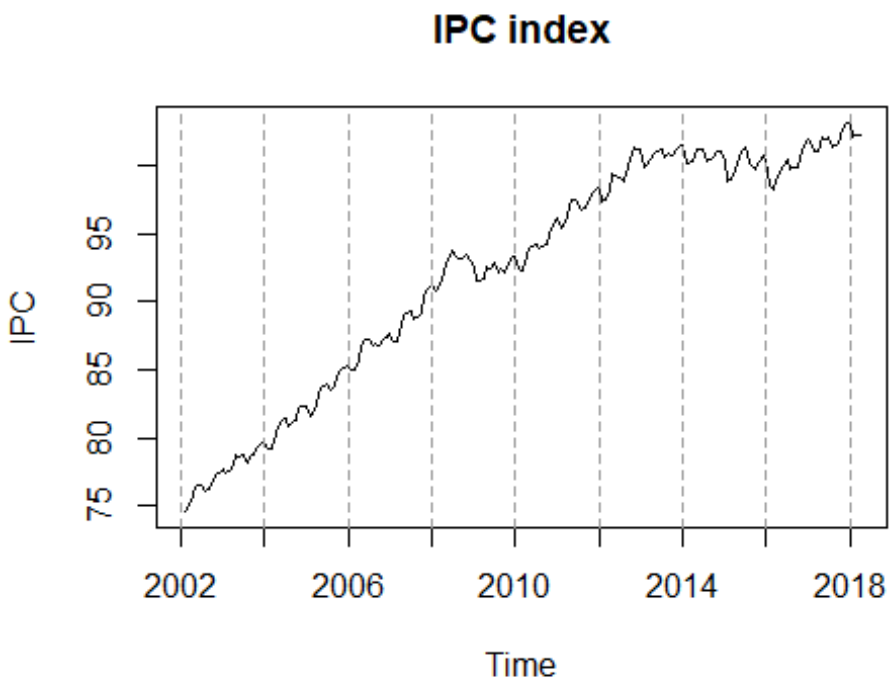
```
setwd("C:/Users/Daniel/Documents/Certificados & Faculdade/UPC Master Big Data/Data Analytics/Time Series/Sesion 08/05/20/Time_Series_Datasets/")
#Reading and formatting the file
goods <- read.csv2(file = "INE_IPC.csv", sep = ";", header = TRUE, nrows = 13, skip = 6)
```

*Exercise 1) Read the data, create 4 univariate time series (IPC index, Monthly variation of IPC index, Yearly variation of IPC index, Variation of IPC index from the begining of the year) and one 4-dimesional time series as the union of the previous 4 (use ts.union). Plot the 4-dim time series.*

```
cols <- colnames(goods)
end_block_IPCindex <- min(grep(".1",cols,fixed = TRUE))-1

for ( i in 2:end_block_IPCindex){
    cols[i] <-  paste0(cols[i],".0")
}

cols[782] <- sub(cols[782],"X.1","X.4")
```

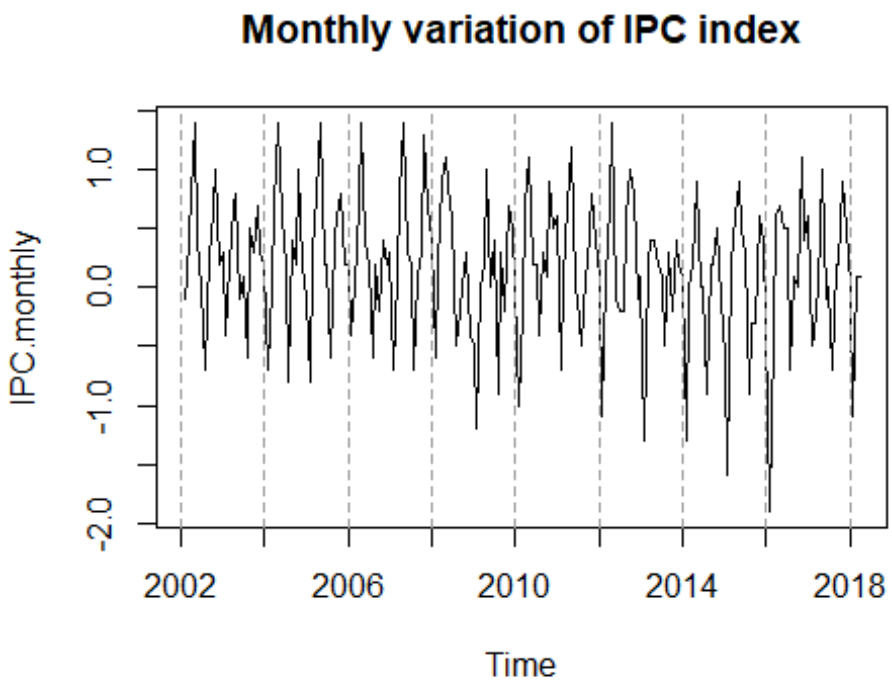
## IPC Index

```
end <- max(grep(".0",cols,fixed = TRUE))
cols.0 <- cols[end:2]

# IPC is only taking into consideration the "Indice geral" row
IPC <- as.numeric(
  t(read.csv2(file = "INE_IPC.csv", skip=7,
              dec = "." , header = FALSE,
              encoding = "UTF-8", row.names = 1,
              nrows=1)[1,(end-1):1])
)

year <- as.numeric(substr(cols.0,start=2,stop=5))
month <- as.numeric(substr(cols.0,start=7,stop=8))
tt <- year + month/12

IPC.ts <- ts(IPC, frequency = 12, start=c(year[1],month[1]))
plot(tt,IPC,type="l",main="IPC index", xlab="Time",lab=c(8,5,7))
abline(v=seq(2002,2018,by=2),lty=2,col=8)
```



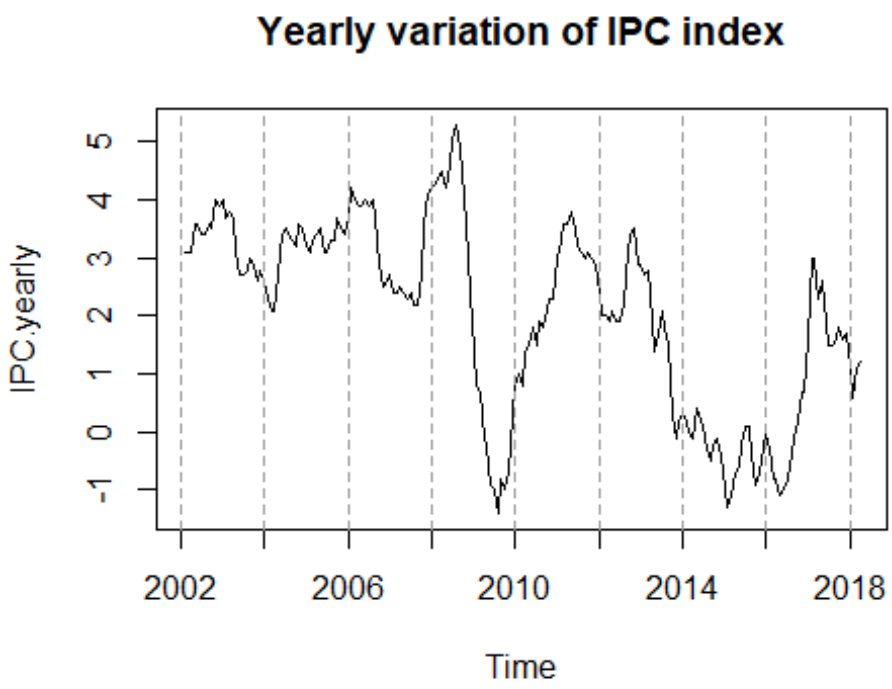

## Monthly variation of IPC index

```
#cols[782] <-  paste0(cols[782],".4")
end.1 <- min(grep(".1",cols,fixed = TRUE))
end.1.max <- max(grep(".1",cols,fixed = TRUE))
cols.1 <- cols[end.1.max:end.1]

# IPC is only taking into consideration the "Indice geral" row
IPC.monthly <- as.numeric(
  t(read.csv2(file = "INE_IPC.csv", skip=7,
              dec = "." , header = FALSE,
              encoding = "UTF-8", row.names = 1,
              nrows=1)[1,(end.1.max-1):(end.1-1)])
)

year.1 <- as.numeric(substr(cols.1,start=2,stop=5))
month.1 <- as.numeric(substr(cols.1,start=7,stop=8))
tt.1 <- year.1 + month.1/12

IPC.ts.monthly <- ts(IPC.monthly, frequency = 12,
start=c(year.1[1],month.1[1]))
plot(tt.1,IPC.monthly,type="l",main="Monthly variation of IPC index",
xlab="Time",lab=c(8,5,7))
abline(v=seq(2002,2018,by=2),lty=2,col=8)
```



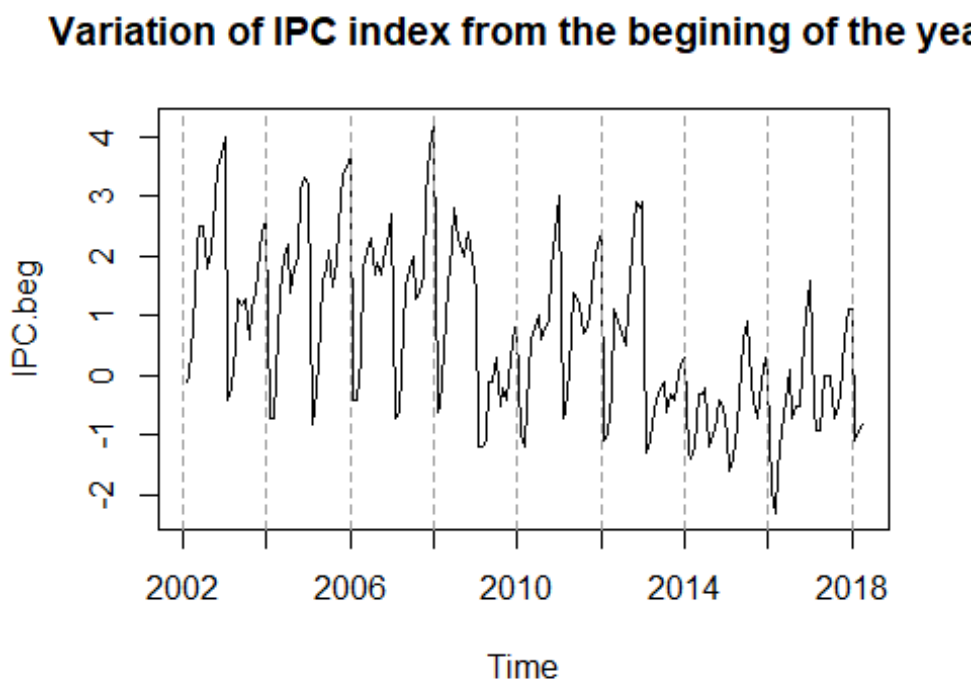

## Yearly variation of IPC index

```
end.2 <- min(grep(".2",cols,fixed = TRUE))
end.2.max <- max(grep(".2",cols,fixed = TRUE))
cols.2 <- cols[end.2.max:end.2]

# IPC is only taking into consideration the "Indice geral" row
IPC.yearly <- as.numeric(
  t(read.csv2(file = "INE_IPC.csv", skip=7,
              dec = "." , header = FALSE,
              encoding = "UTF-8", row.names = 1,
              nrows=1)[(end.2.max-1):(end.2-1)])
)

year.2 <- as.numeric(substr(cols.2,start=2,stop=5))
month.2 <- as.numeric(substr(cols.2,start=7,stop=8))
tt.2 <- year.2 + month.2/12

IPC.ts.yearly <- ts(IPC.yearly, frequency = 12,
start=c(year.2[1],month.2[1]))
plot(tt.2,IPC.yearly,type="l",main="Yearly variation of IPC index",
xlab="Time",lab=c(8,5,7))
abline(v=seq(2002,2018,by=2),lty=2,col=8)
```

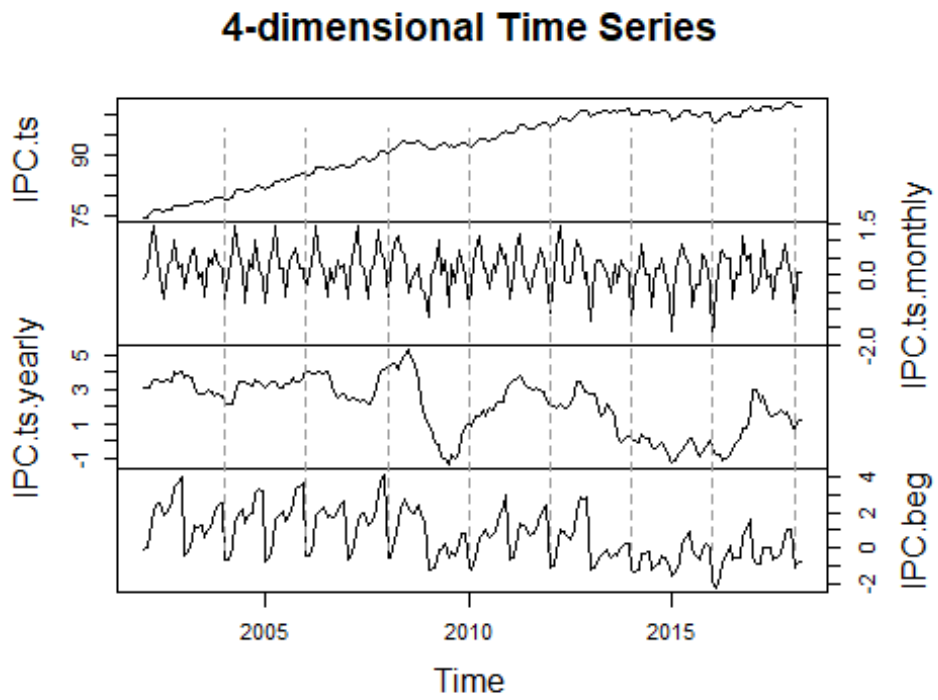



## Variation of IPC index from the begining of the year

```
end.3 <- min(grep(".3",cols,fixed = TRUE))
end.3.max <- max(grep(".3",cols,fixed = TRUE))
cols.3 <- cols[end.3.max:end.3]

# IPC is only taking into consideration the "Indice geral" row
IPC.beg <- as.numeric(
  t(read.csv2(file = "INE_IPC.csv", skip=7,
              dec = "." , header = FALSE,
              encoding = "UTF-8", row.names = 1,
              nrows=1)[(end.3.max-1):(end.3-1)])
)

year.3 <- as.numeric(substr(cols.3,start=2,stop=5))
month.3 <- as.numeric(substr(cols.3,start=7,stop=8))
tt.3 <- year.3 + month.3/12

IPC.ts.beg <- ts(IPC.beg, frequency = 12, start=c(year.3[1],month.3[1]))
plot(tt.3,IPC.beg,type="l",main="Variation of IPC index from the begining
of the year", xlab="Time",lab=c(8,5,7))
abline(v=seq(2002,2018,by=2),lty=2,col=8)
```




```
tsunion <- ts.union(IPC.ts,IPC.ts.monthly,IPC.ts.yearly,IPC.beg)
plot(tsunion,main="4-dimensional Time Series",yax.flip=TRUE)
abline(v=seq(2002,2018,by=2),lty=2,col=8)
```



*Exercise 2) Answer the following questions:*

## i. Which series have a seasonal pattern?

**Respuesta:** El time serie de "Variation of IPC index from beginning of the year" es aquello que tiene un patrón "seasonal". Esto porque se pueden observar picos de subida muy definidos, donde en parece hay una subida más pequeña en la mitad del año y otro más grande a finales del año. Quizás también se puede llegar a decir que, a simple vista, el "IPC Montly Variation" tiene patrones que se repiten.

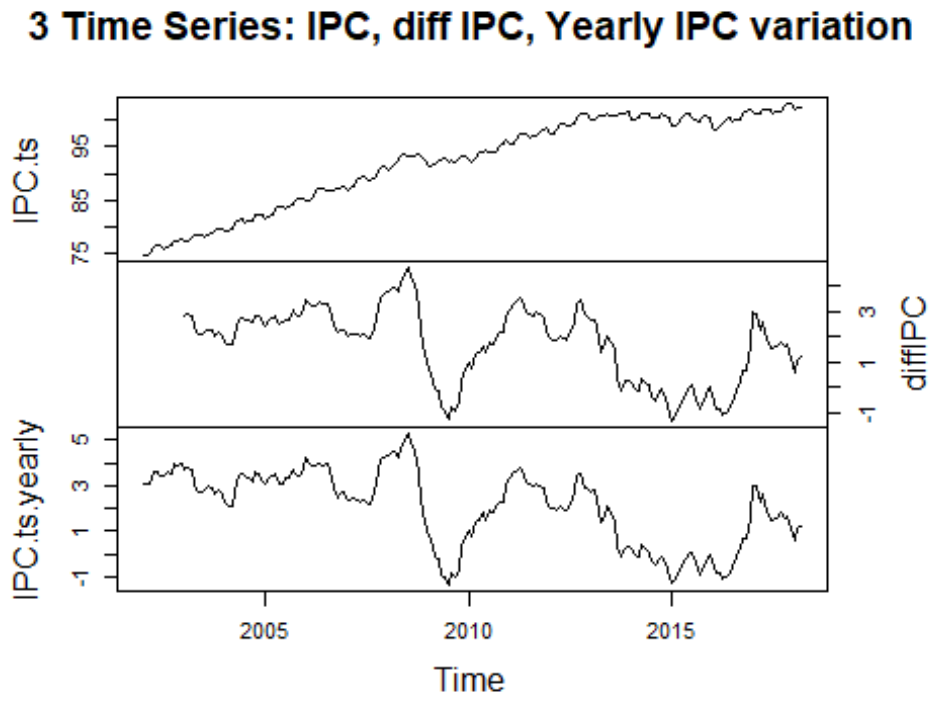
## ii. Which series are stationary?

**Respuesta:** El time serie de "Monthly variaton of IPC index" tiene un patrón estacionario. Esto porque hay subidas y bajadas periódicas que se presentan en forma regular en la serie de tiempo. Además de esto, parece que la media y la variancia se mantiene constante.

*Exercise 3) Compute the difference of order 12 of the IPC time series (using function diff). Then plot the union of 3 time series: IPC, its order 12 differences, and the series containg yearly variation of IPC index we read from file INE_IPC.csv.*

- Do you think that yearly variation of IPC index is stationary in variance?
  **Respuesta:** No, todavía seguimos creyendo que no es estacionaria en varianza, teniendo en cuenta lo que vimos en la diferencia de orden 12.

```
#op <- par(mfrow=c(2,2))
diffIPC <- diff(IPC.ts,lag = 12, differences = 1)
tsunionex3 <- ts.union(IPC.ts,diffIPC,IPC.ts.yearly)
plot(tsunionex3,main = "3 Time Series: IPC, diff IPC, Yearly IPC
variation", yax.flip = TRUE)
```



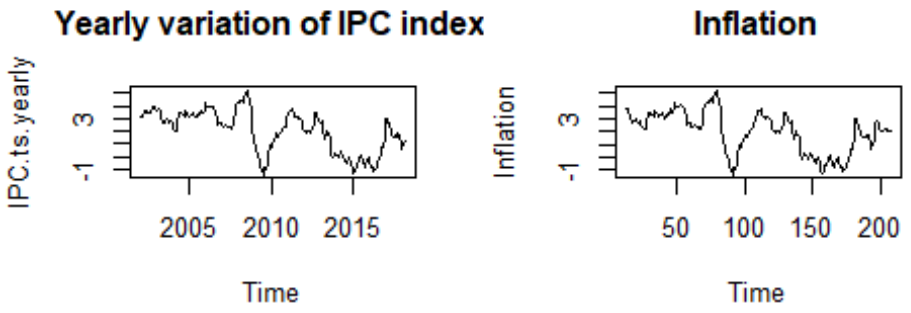

```
#par(op)
```

*Exercise 4) Compute the Inflation as Inflation <- 100*diff(IPC,lag=12)/lag(IPC,k=-12).*

```
Inflation <- 100*diff(IPC,lag=12)/lag(IPC,k=-12)

## Warning in 100 * diff(IPC, lag = 12)/lag(IPC, k = -12): comprimento do
objeto
## maior não é múltiplo do comprimento do objeto menor

classts <- function(x){if(class(x)=="ts") "Yes" else "No"}
cat("* Is inflation of class ts?", classts(Inflation))

## * Is inflation of class ts? No
```

- Is Inflation of class ts? **Respuesta:** No
- Why do you think Inflation is so similar to the yearly variation of IPC index?

```
op <- par(mfrow=c(2,2))
plot.ts(IPC.ts.yearly, main = "Yearly variation of IPC index")
plot.ts(Inflation, main = "Inflation")
par(op)
```




**Respuesta:** Llegan a ser muy similares porque el calculo de la Inflación es medido con relación a la tasa de crecimiento anual. Ambos son indicadores de precios. La diferencia entre IPC e inflación es en relación a la cantidad de productos que cada índice toman en consideración para medir los incrementos en los precios. Al final no llegan a ser exactamente iguales (unicamente son similares) pues la inflación es la subida mas generalizada de los precios, mientras que el IPC mide cestas de consumo en concreto.

*Exercise 5) Compute and plot the Autocorrelation Functions (ACFs) of the following time series:*

- The 4 we read from file INE_IPC.csv.
- The inflation time series.
- The IPC series after having taken difference of orders 1 (regular difference) and 12 (seasonal difference).

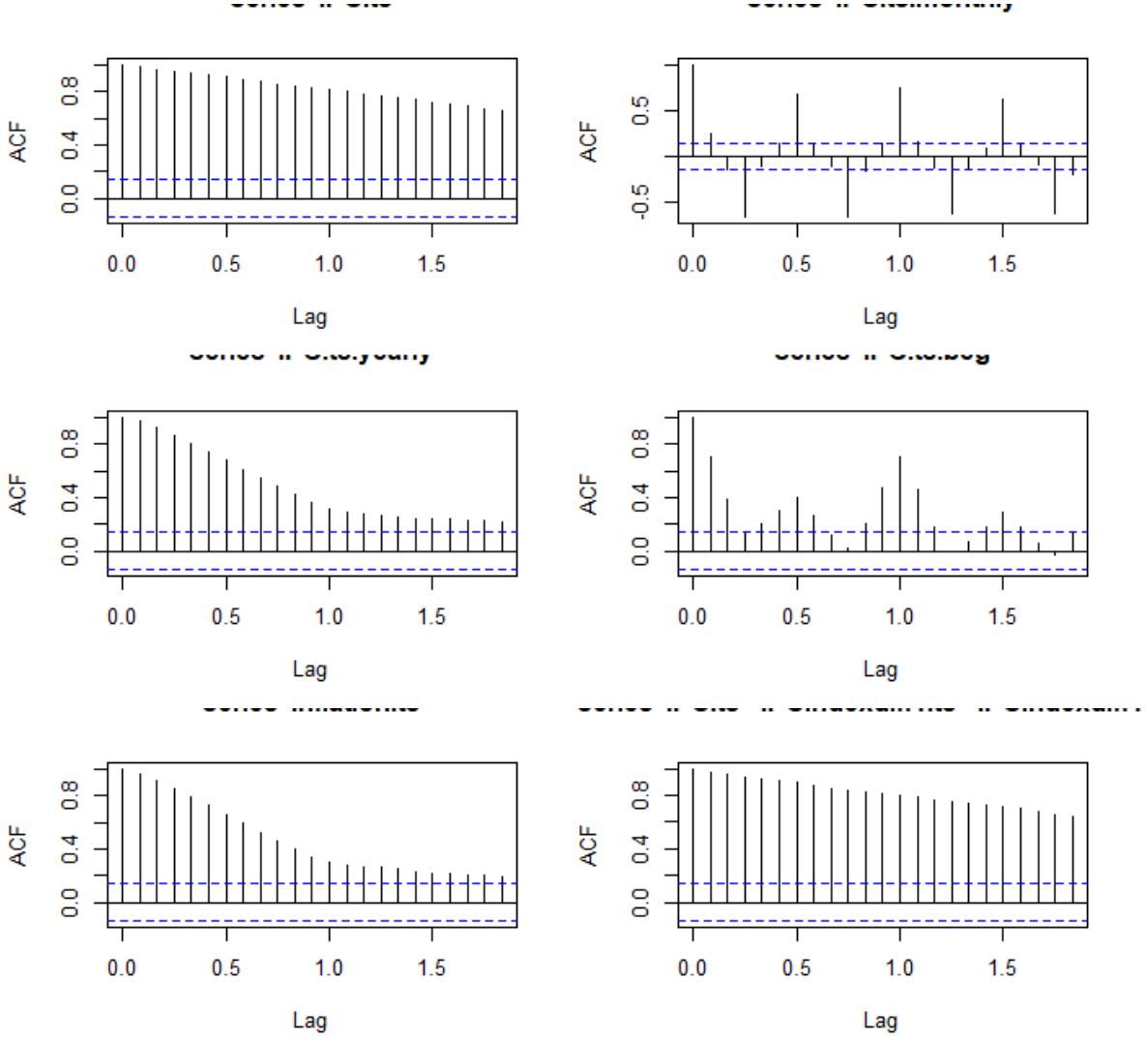
- Plot this series previously. Do you think it is stationary?

**Respuesta:** Las que parecen estacionarias son la de “Yearly Variation of IPC index”, “Variation of IPC index from the begining of the year”.

**Fuentes usada en la investigación del ejercicio:**

- https://coolstatsblog.com/2013/08/07/how-to-use-the-autocorreation-function-acf/
- https://otexts.com/fpp2/stationarity.html
- https://towardsdatascience.com/detecting-stationarity-in-time-series-data-d29e0a21e638

```
par(mfrow=c(2,2))
#4 time series we read
acf(IPC.ts)
acf(IPC.ts.monthly)
acf(IPC.ts.yearly)
acf(IPC.ts.beg)

#Inflation
Inflation.ts <- ts(Inflation, start=c(year[1], month[1]), frequency = 12)
acf(Inflation.ts)

#The IPC series after having taken difference of orders 1 (regular
difference) and 12 (seasonal difference)
IPCindexdiff1.ts <- ts(diff(IPC, lag=1), frequency = 12, start=c(year[1],
month[1]))
IPCindexdiff12.ts <- ts(diff(IPC, lag=12), frequency = 12,
start=c(year[1], month[1]))
#acf(IPCindexdiff1.ts)
#acf(IPCindexdiff12.ts)
acf(IPC.ts-IPCindexdiff1.ts-IPCindexdiff12.ts)

par(mfrow=c(1,1))
```




*Exercise 6) Do the structural decomposition of the following time series:*

- IPC
- Monthly variation of th IPC.
- Inflation.

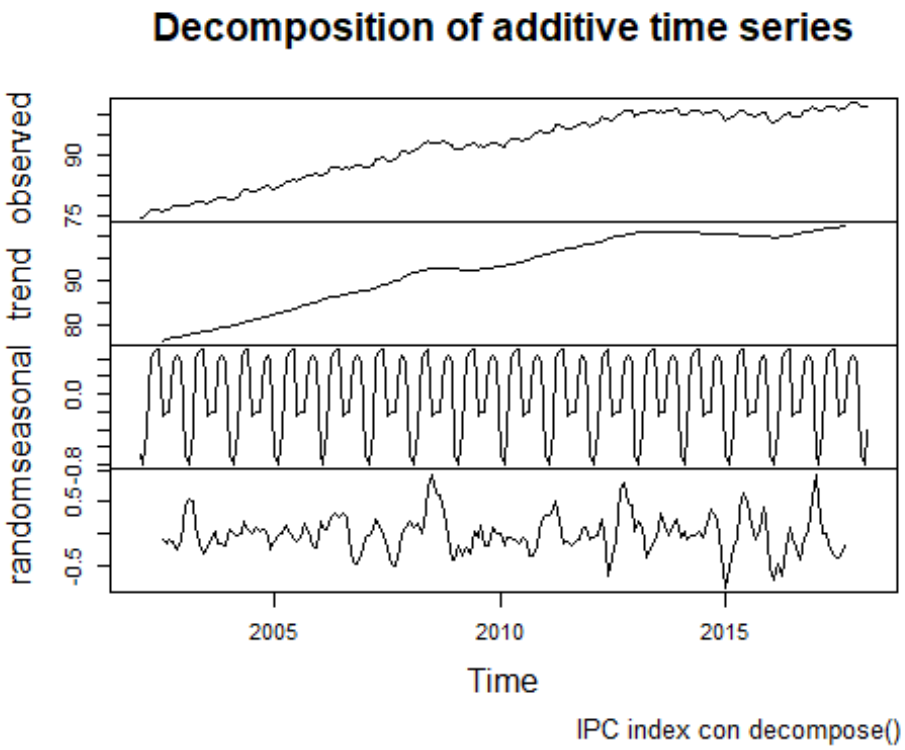
- The IPC series, differentiated regularly and seasonally.

- Use first decompose and stl, plot the obtaind components and do a barplot of the one-year seasonal component.

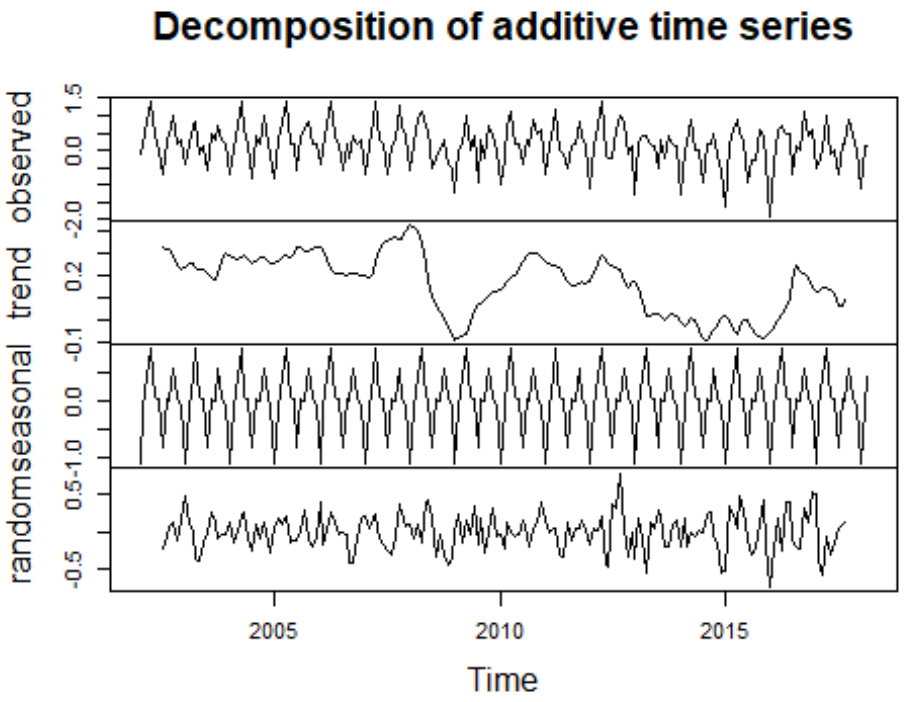
- Then use the function HoltWinters from package astsa.

### Decompose()

```
IPC.dec <- decompose(IPC.ts)
IPC.monthly.dec <- decompose(IPC.ts.monthly)
IPC.inflation.dec <- decompose(Inflation.ts)

IPCdiff_reg_season.ts <- (IPC.ts-IPCindexdiff1.ts-IPCindexdiff12.ts)
IPCdiff_reg_season.dec <- decompose(IPCdiff_reg_season.ts)
#IPC.yearly.dec <- decompose(IPC.ts.yearly)
#IPC.beg.dec <- decompose(IPC.ts.beg)

op<-par(mfrow=c(2,2))
plot(IPC.dec)
title(sub="IPC index con decompose()")
```

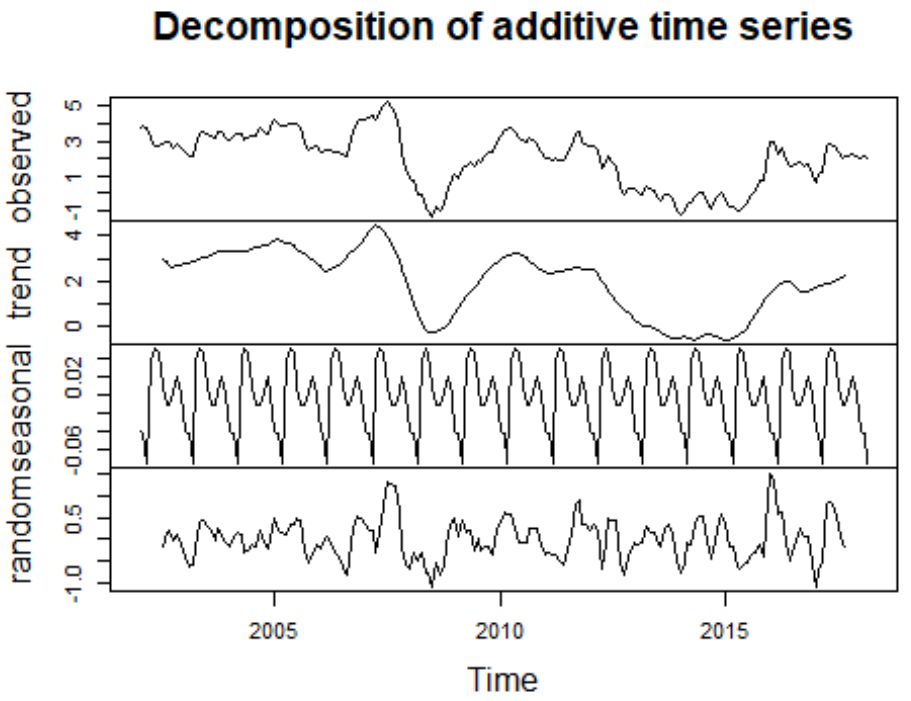



```
plot(IPC.monthly.dec)
title(sub="IPC montly con decompose()")
```



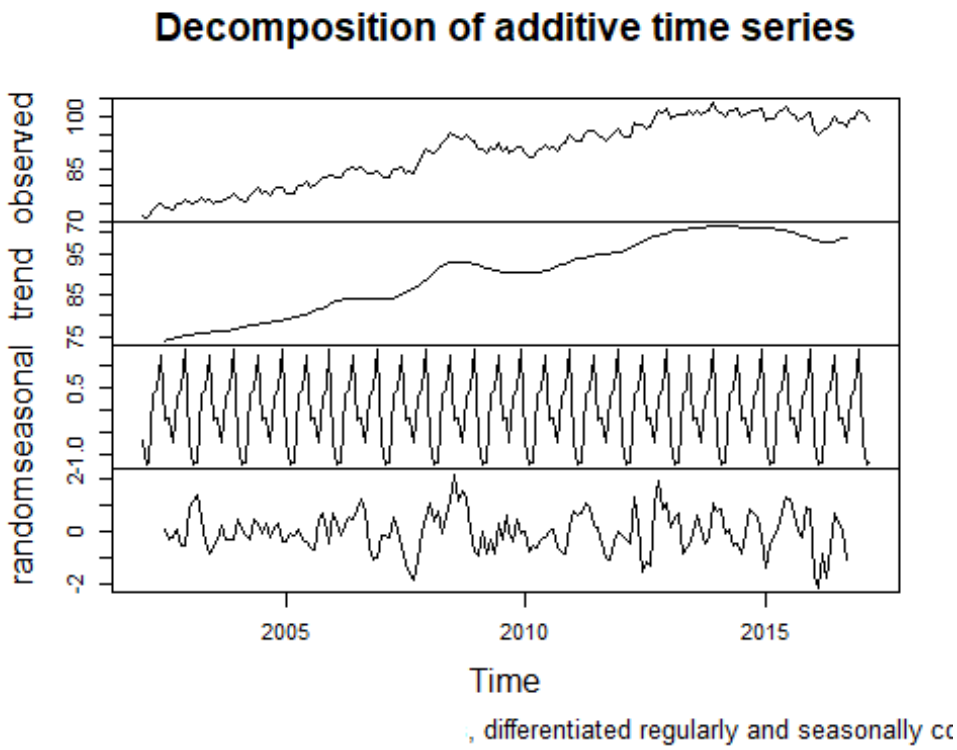

IPC montly con decompose()

```
plot(IPC.inflation.dec)
title(sub="Inflation con decompose()")
```




Inflation con decompose()

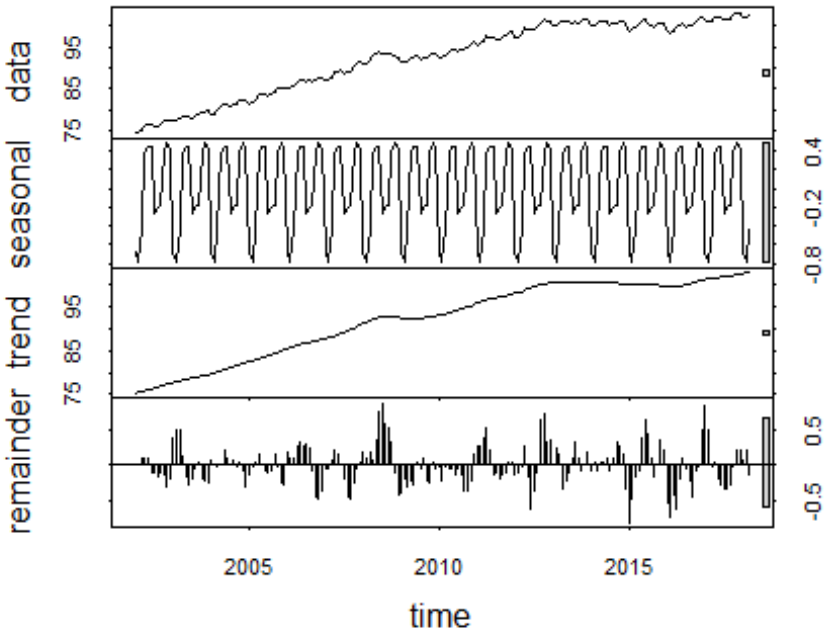
```
plot(IPCdiff_reg_season.dec)
title(sub="IPC series, differentiated regularly and seasonally con
decompose()")
```




```
#plot(IPC.yearly.dec)
#title(sub="IPC yearly con decompose()")

#plot(IPC.beg.dec)
#title(sub="IPC beginning of the year con decompose()")

par(op)
```

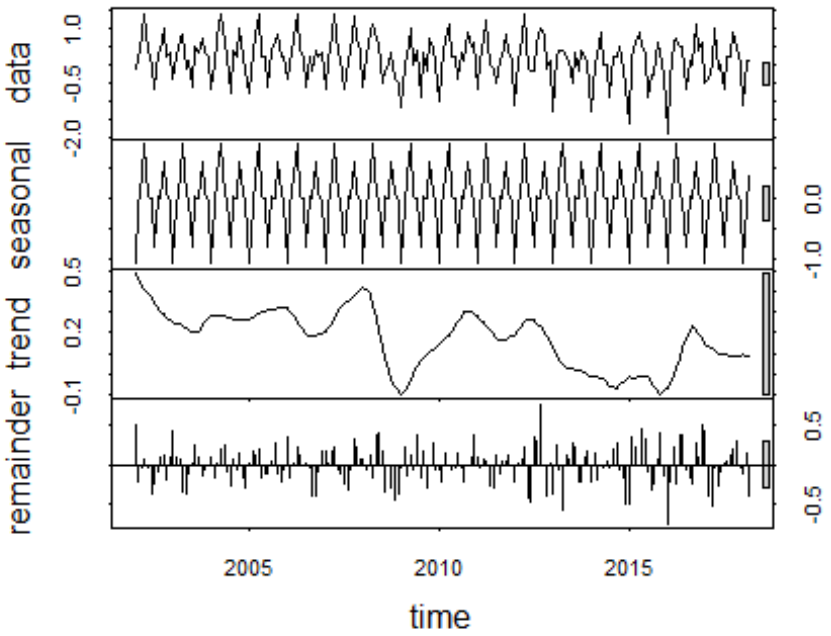
### Stl()

```
IPC.stl <- stl(IPC.ts, s.window = "period")
IPC.monthly.stl <- stl(IPC.ts.monthly, s.window = "period")

IPC.inflation.stl <- stl(Inflation.ts, s.window = "period")
IPCdiff_reg_season.stl <- stl(IPCdiff_reg_season.ts, s.window = "period")
#IPC.yearly.stl <- stl(IPC.ts.yearly, s.window = "period")
#IPC.beg.stl <- stl(IPC.ts.beg, s.window = "period")

op<-par(mfrow=c(2,2))
plot(IPC.stl)
title(sub="IPC index con stl()")
```



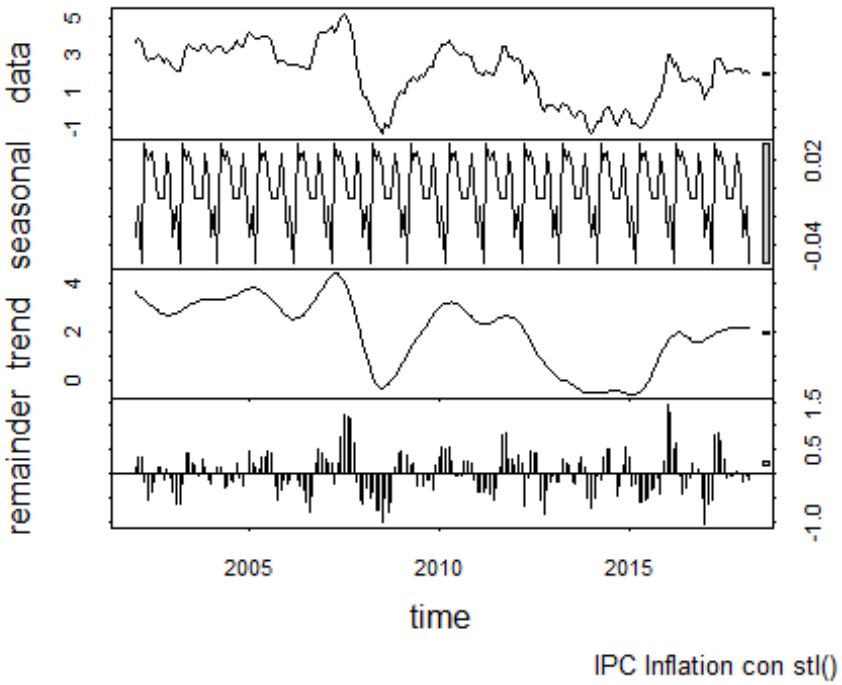

IPC index con stl()

```
plot(IPC.monthly.stl)
title(sub="IPC montly con stl()")
```



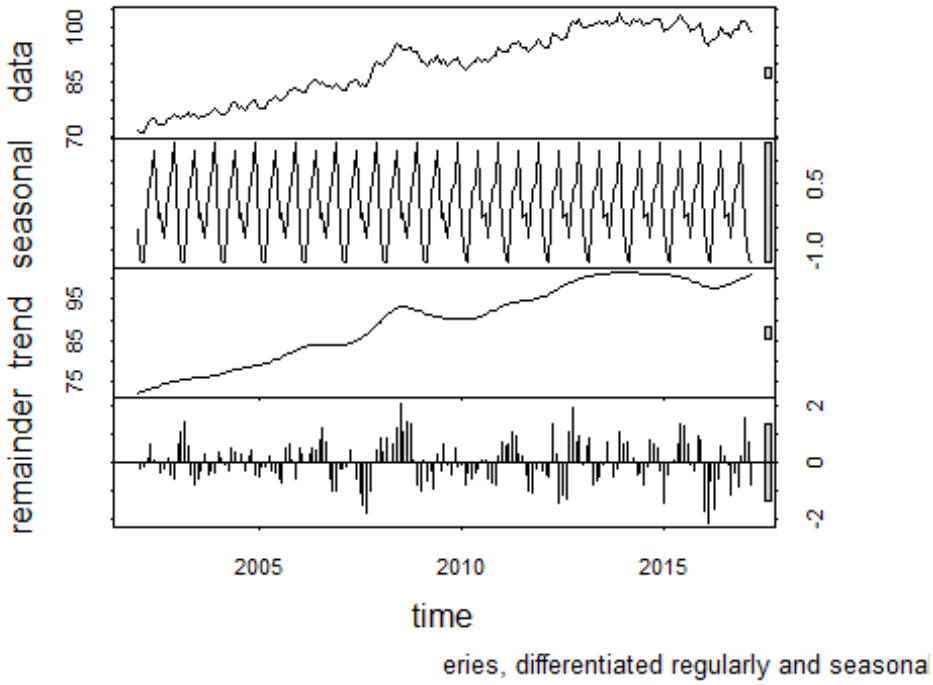

IPC montly con stl()

```
plot(IPC.inflation.stl)
title(sub="IPC Inflation con stl()")
```




IPC Inflation con stl()

```
plot(IPCdiff_reg_season.stl)
title(sub="IPC series, differentiated regularly and seasonally con
stl()")
```

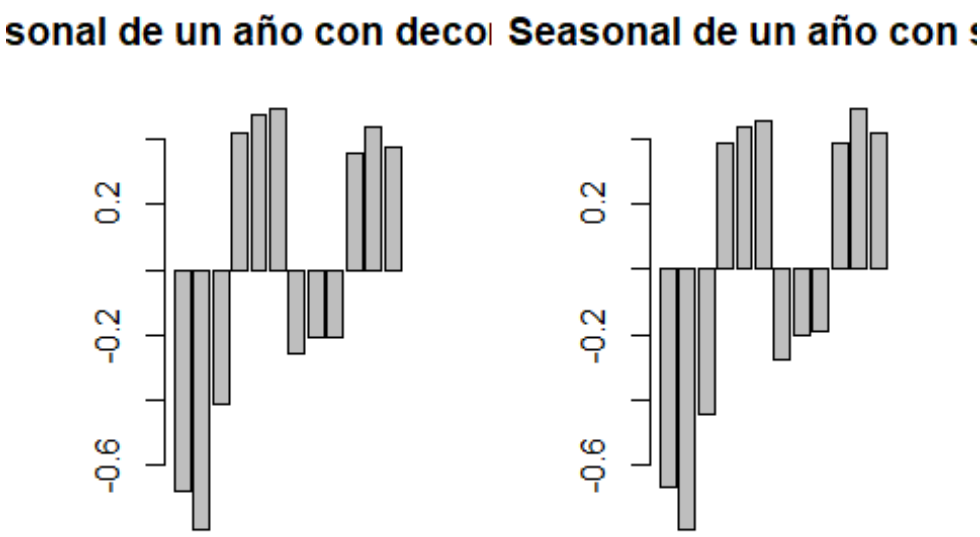



```
par(op)
```

### Barplot of one-year seasonal

*IPC Index*

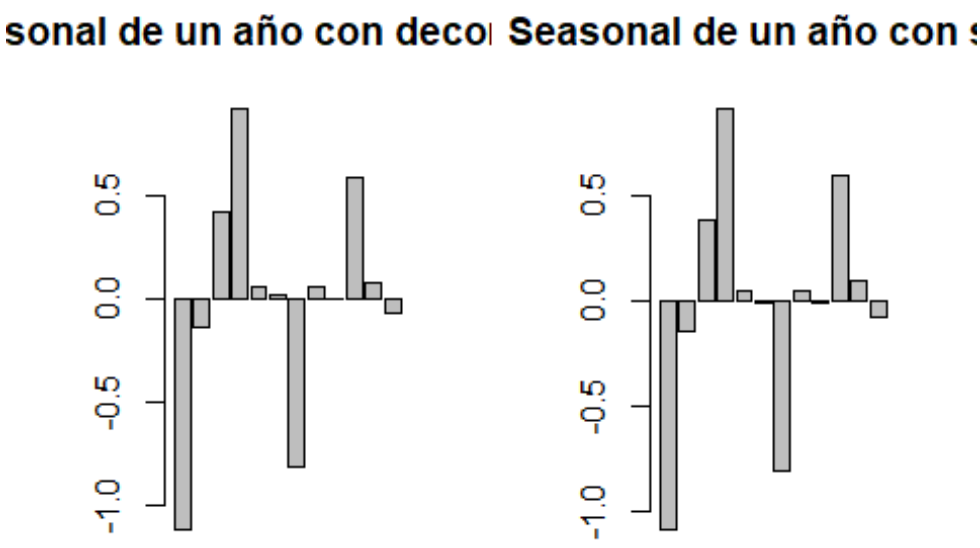
```
op<-par(mfrow=c(1,2))
#Pillamos el decompose de la time series de IPC index
barplot(IPC.dec$figure, main="Seasonal de un año con decompose()")
#Año de 2017
barplot(IPC.stl$time.series[181:192,1], main="Seasonal de un año con
stl()")
```




```
par(op)
```

*Monthly variation of th IPC*

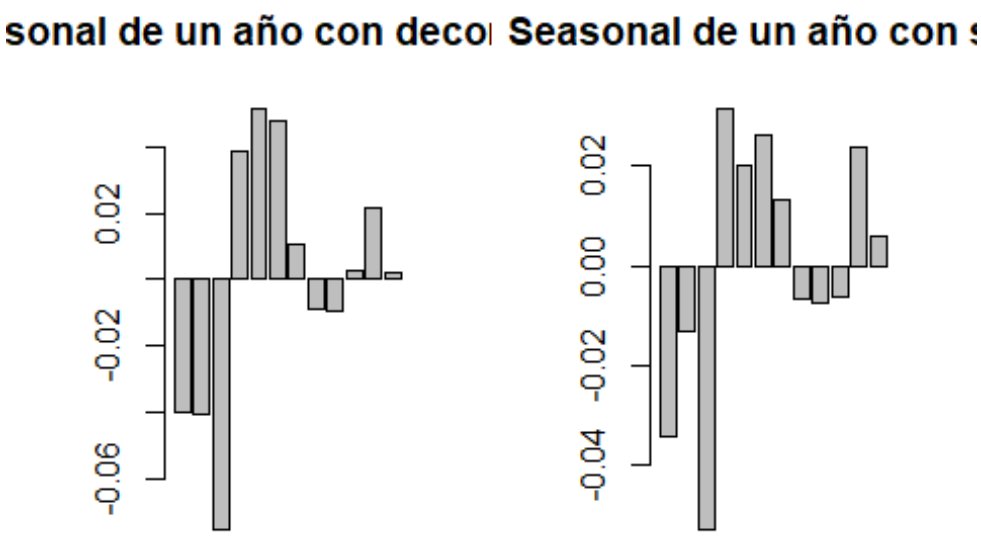
```
op<-par(mfrow=c(1,2))
#Pillamos el decompose de la time series de IPC index
barplot(IPC.monthly.dec$figure, main="Seasonal de un año con
decompose()")
#Año de 2017
barplot(IPC.monthly.stl$time.series[181:192,1], main="Seasonal de un año
con stl()")
```




```
par(op)
```

*Inflation*

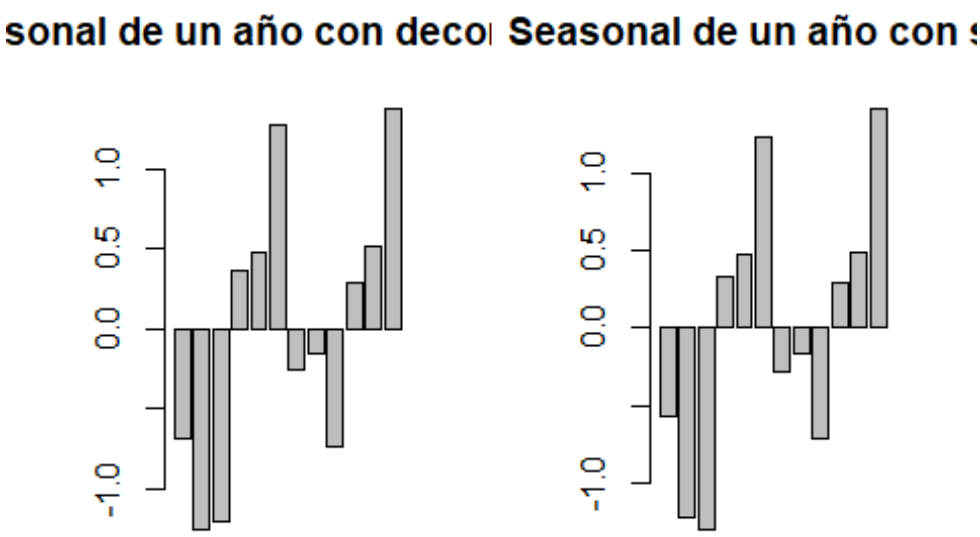
```
op<-par(mfrow=c(1,2))
#Pillamos el decompose de la time series de IPC index
barplot(IPC.inflation.dec$figure, main="Seasonal de un año con
decompose()")
#Año de 2017
barplot(IPC.inflation.stl$time.series[181:192,1], main="Seasonal de un
año con stl()")
```




```
par(op)
```

*The IPC series, differentiated regularly and seasonally*

```
op<-par(mfrow=c(1,2))
#Pillamos el decompose de la time series de IPC index
barplot(IPCdiff_reg_season.dec$figure, main="Seasonal de un año con
decompose()")
#Año de 2017
barplot(IPCdiff_reg_season.stl$time.series[1:12,1], main="Seasonal de un
año con stl()")
```




```
par(op)
```

### HoltWinters

```
library(astsa)

IPCholt <- HoltWinters(IPC.ts,beta=FALSE,gamma=FALSE)
IPCholt

## Holt-Winters exponential smoothing without trend and without seasonal
component.
##
## Call:
## HoltWinters(x = IPC.ts, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.9999593
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##      [,1]
## a 102.329

IPCmohtlyholt <- HoltWinters(IPC.ts.monthly,beta=FALSE,gamma=FALSE)
IPCmohtlyholt
```

```
## Holt-Winters exponential smoothing without trend and without seasonal
component.
##
## Call:
## HoltWinters(x = IPC.ts.monthly, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.03536348
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 0.06429138
```

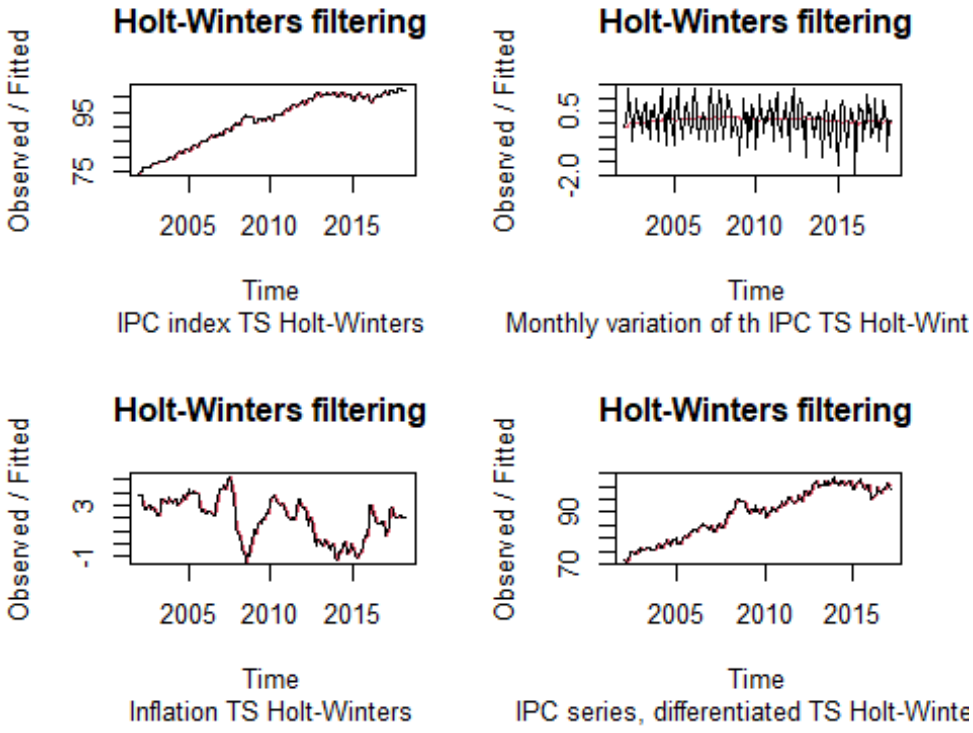
```
IPCinflationholt <- HoltWinters(Inflation.ts,beta=FALSE,gamma=FALSE)
IPCinflationholt
```

```
## Holt-Winters exponential smoothing without trend and without seasonal
component.
##
## Call:
## HoltWinters(x = Inflation.ts, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.9999404
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 1.976964
```

```
IPCseriesholt <-
HoltWinters(IPCdiff_reg_season.ts,beta=FALSE,gamma=FALSE)
IPCseriesholt
```

```
## Holt-Winters exponential smoothing without trend and without seasonal
component.
##
## Call:
## HoltWinters(x = IPCdiff_reg_season.ts, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.9676009
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 98.93973
```

```r
op <- par(mfrow=c(2,2))
plot(IPCholt)
title(sub="IPC index TS Holt-Winters")
plot(IPCmohtlyholt)
title(sub="Monthly variation of th IPC TS Holt-Winters")
plot(IPCinflationholt)
title(sub="Inflation TS Holt-Winters")
plot(IPCseriesholt)
title(sub="IPC series, differentiated TS Holt-Winters")
```




```r
par(op)
```

**Fuentes de investigación:**

- https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html
- https://docs.tibco.com/pub/enterprise-runtime-for-R/4.0.1/doc/html/Language_Reference/stats/HoltWinters.html

**Observaciones:** Hemos observado que en 3 de las time series el valor de **alpha** era muy proximo a 1 (0.99) en estos casos. Esto significa que los ultimos valores de la time series tienen un peso más elevado. Por otro lado, cuando alpha es muy bajo, significa que valores mas antiguos tienen peso más grande.