

# **Series Temporales. Práctica 2**

Departamento de Estadística e Investigación Operativa.

Grado en Estadística. Universidad de Sevilla

## **Análisis de componentes no observadas**

<b>1. Introducción</b>	<b>2</b>
<b>2. Definición de una serie temporal en R</b>	<b>2</b>
<b>3. Descomposición de la serie en componentes</b>	<b>3</b>
<b>4. Cálculo de la tendencia</b>	<b>5</b>
<b>5. Cálculo de predicciones</b>	<b>6</b>

## Análisis de componentes no observadas

### Análisis de componentes no observadas

## 1 Introducción

En esta práctica se presentan las instrucciones necesarias para realizar una análisis clásico de componentes no observadas de una serie temporal en R:

1. definición de una serie temporal,
2. representación de la serie,
3. descomposición de la serie temporal en componentes,
4. cálculo de la tendencia secular y
5. predicciones.

## 2 Definición de una serie temporal en R

Introducimos los datos del siguiente ejemplo relativo a las ventas trimestrales de una fábrica de calzado expresadas en cientos de miles de euros para los años 2002, 2003 y 2004:

Años ↓	2002	2003	2004
1 <sup>o</sup>	150	155	160
2 <sup>o</sup>	165	170	180
3 <sup>o</sup>	125	135	140
4 <sup>o</sup>	170	165	180

```
datos1 = c(150, 165, 125, 170, 155, 170, 135, 165, 160, 180, 140, 180)
datos1
```

También se usarán los datos de paro vistos en la práctica anterior.

```
paro <- read.csv("paro.csv", header=F, dec=".", sep=";")
```

La **función ts** crea un objeto de clase "ts" (serie de tiempo) a partir de un vector (serie de tiempo única) o una matriz (serie multivariada).

En su forma básica, se crea la serie temporal con el comando **ts(data,frequency,start)** donde: **data** es la serie de datos, **frequency** es el número de estaciones, y **start** es un vector cuyas componentes indican el año y la estación de inicio, respectivamente.

```
serie1=ts(data=datos1,frequency=4,start=c(2002,1))
serie1
```

*Ejemplo 1.*

```
ts(1:10, start = 1959)
ts(1:47, frequency = 12, start = c(1959, 2))
ts(1:10, frequency = 4, start = c(1959, 2))
ts(matrix(rpois(36, 5), 12, 3), start=c(1961, 1), frequency=12)
z <- ts(matrix(rnorm(300), 100, 3), start = c(1961, 1), frequency = 12)
plot(z)
paro <- ts(paro[,2], start=2003,freq=4)
plot(paro)
```

### 3 Descomposición de la serie en componentes

La **función decompose()** tiene como objetivo descomponer una serie temporal en las componentes estacional, tendencia e irregular, usando las medias móviles. Puede usar el modelo multiplicativo y el modelo aditivo.

La forma de utilizar esta función es la siguiente.

```
decompose(x, type = c("additive", "multiplicative"), filter = NULL)
```

donde

- **x**: Es la serie que se quiere descomponer.

- **type:** Tipo de componente estacional
- **filter:** Un vector con los coeficientes del filtro, usados para filtrar la componente estacional. NULL se usa para medias móviles con ventana simétrica.

*Ejemplo 2.* Descomponer la serie de paro:

```
m <- decompose(paro)
plot(m)      #representa las componentes de la serie
m
m$x
m$figure
plot(m$figure, type="h")
m$trend
plot(m$trend, type="l")
```

Para el ejemplo inicial (según el esquema multiplicativo):

```
componentes=decompose(serie1,type=c("multiplicative"))
```

La serie de medias móviles centradas ha quedado almacenada en el valor **trend**.

```
mmcent=componentes$trend
mmcent
```

Los índices de variación estacional han quedado almacenados en el valor **varest1**.

```
varest1=componentes$figure #Índice para cada estación
varest1
```

Los índices de variación estacional repetidos para cada año han quedado almacenados en el valor **seasonal**.

```
varest2=componentes$seasonal
#Índice para cada estación repetidos para cada año
varest2
```

## 4 Cálculo de la tendencia

El objetivo de **la función `lm()`** es ajustar modelos lineales. Se puede usar para ejecutar regresiones.

La forma de utilizar esta función es la siguiente:

```
lm(formula, data, subset, weights, na.action,  
   method = "qr",  
   model = TRUE, x = FALSE, y = FALSE, qr = TRUE,  
   singular.ok = TRUE, contrasts = NULL, offset, ...)
```

donde

- **formula:** es una fórmula que describe simbólicamente el modelo a ajustar. Un modelo tiene la forma **respuesta** ~ **modelo**, siendo **respuesta** el vector respuesta, y **modelo** es una serie de términos que especifican una predicción lineal para **respuesta**.
- **data:** Es opcional. Sirve para especificar el dataframe que contiene las variables del modelo.

Aplicaremos la metodología explicada en clase para obtener la tendencia. Se hará con el ejemplo del problema inicial.

```
Ejemplo 3. datos1 = c(150, 165, 125, 170, 155, 170, 135, 165, 160, 180, 140, 180)  
datos1  
serie1=ts(data=datos1,frequency=4,start=c(2002,1))  
serie1  
plot(serie1,type="l",ylim=c(min(serie1),max(serie1)))  
componentes=decompose(serie1,type=c("multiplicative"))  
mmcent=componentes$trend  
mmcent  
varest1=componentes$figure # Índice para cada estación  
varest1  
varest2=componentes$seasonal  
#Índice para cada estación repetidos para cada año  
varest2
```

Obtenemos la serie desestacionalizada.

```
serie_des=serie1/varest2 #serie desestacionalizada
serie_des
```

Para calcular la recta de tendencia consideraremos la serie de instantes de tiempo dada por  $T = \{1, 2, 3, 4, \dots, 14\}$  sino como  $T = \{2002, 2002,25, 2002,5, 2002,75, 2003, \dots, 2004,75\}$ .

```
Time=time(serie_des)
#Proporciona los tiempos en los que ha sido medida la serie
Time
```

La tendencia secular es la recta  $tendencia = a + b * t$  donde  $a$  y  $b$  son los coeficientes de la recta de regresión de la serie desestacionalizada:

```
rTendT=lm(serie_des~Time) #recta de regresión
rTendT

coef=as.vector(rTendT$coefficients)
#extraigo los coeficientes
coef

a=coef[1]
b=coef[2]
a
b
```

Vamos a representar la serie temporal junto con Representamos la serie y su tendencia. Recordemos que **lty** es el tipo de línea, **col** el color de la línea.

## 5 Cálculo de predicciones

El objetivo de **la función predict()** es predecir valores según un modelo lineal.

La forma de utilizar esta función es la siguiente:

```
predict(object, newdata, se.fit = FALSE, scale = NULL,
df = Inf, interval = c("none", "confidence", "prediction"),
level = 0.95, type = c("response", "terms"),
terms = NULL, na.action = na.pass,
pred.var = res.var/weights, weights = 1, ...)
```

donde

- **object:** Es un objeto de la clase `lm`. Object of class inheriting from "lm"
- **newdata:** Un dataframe opcional para buscar valores de las variables con las que predecir
- **interval:** Tipo de cálculo de intervalos.
- **level:** Nivel de confianza.

Seleccionamos un instante  $t$  donde queremos realizar la predicción. Por ejemplo, el trimestre siguiente al último de nuestra serie.

Los datos almacenados en la serie temporal *serie1* termina en el cuarto trimestre. El trimestre siguiente a predecir sería el primero, por lo que la estación y el instante de tiempo a utilizar en la predicción serían  $frec = 1$  y  $t = 2005$ .

```
frec=1
t=2005
(tsecular=a+b*t) #calculo la tendencia secular
(prediccion=tsecular*varest1[frec]) #predicción (modelo multiplicativo)
```

Otra forma de hacer las predicciones es considerar los índices de tiempo: 1, 2, 3, ... A continuación se recoge cómo se haría con R:

```
(Time = 1:12)
rTendT=lm(serie_des~Time) #recta de regresión
rTendT
(srTendT = summary(rTendT))
(Rcuadrado = srTendT$r.squared)
coef=as.vector(rTendT$coefficients) #se extraen los coeficientes
coef
a=coef[1]
b=coef[2]
a
b
(varest1bien = varest1[c(1,2,3,4)])
frec=c(1:4)
t=13:16
(tsecular=a+b*t) #cálculo la tendencia secular
(prediccion=tsecular*varest1bien[frec]) #predicción
```