

Series temporales, práctica 1: conjunto de datos meteorológicos de Granada-aeropuerto Chauchina medidos por AEMET.

Carlos Manuel Sequí Sánchez

Problema 2. ¿Qué valores de temperatura máxima, a escala diaria, se espera para la primera semana de Marzo de 2018?

Primeramente leemos el conjunto de datos que contiene los siguientes atributos:

- Columna 1 : Identificador Estación
- Columna 2 : Fecha
- Columna 3 : Temperatura Máxima (°C)
- Columna 4 : Hora Temperatura Máxima
- Columna 5 : Temperatura mínima (°C)
- Columna 6 : Hora Temperatura mínima
- Columna 7 : Temperatura Media (°C)
- Columna 8 : Racha máxima de viento (Km/h)
- Columna 9 : Hora de Racha Máxima
- Columna 10 : Velocidad media de Viento (Km/h)
- Columna 11 : Hora de Velocidad Máxima de viento
- Columna 12 : Precipitacion Total diaria (mm)
- Columna 13 : Precipitacion de 0 a 6 horas (mm)
- Columna 14 : Precipitacion de 6 a 12 horas (mm)
- Columna 15 : Precipitacion de 12 a 18 horas (mm)
- Columna 16 : Precipitacion de 18 a 24 horas (mm)

Librerías...

```
library(tseries)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##   date
```

Leemos el dataset y, como solo nos interesa la fecha y la temperatura máxima nos quedamos con tan solo esos datos.

```

datos = read.csv("5530E.csv", header = TRUE, sep=";")
datos = datos[,c("Fecha", "Tmax")]
datos$Fecha = as.Date(datos$Fecha)

```

Veamos los valores NA...

```

apply(datos, 2, function(atributo){sum(is.na(atributo))})

```

```

## Fecha Tmax
##      0   124

```

Eliminamos las instancias(días) donde hay al menos algún valor NA de temperatura máxima

```

datos = datos[complete.cases(datos),]

```

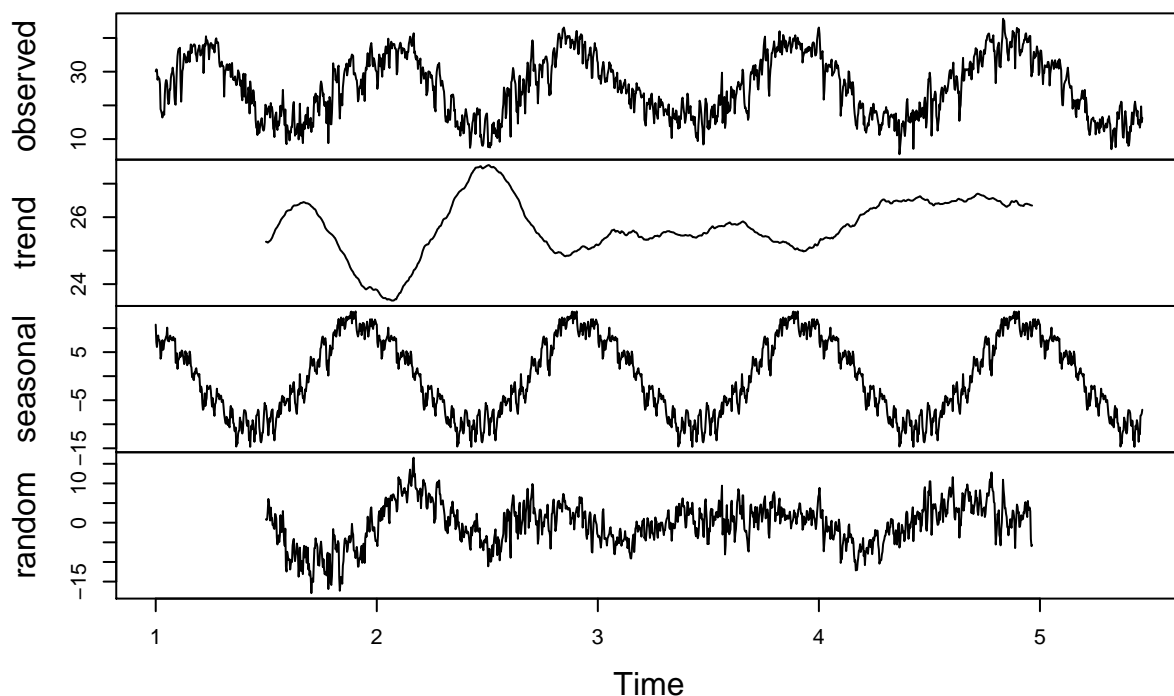
Obtenemos ahora la cantidad de datos a predecir y la serie temporal en sí

```

Npred = 7 # cantidad de datos a predecir (temperaturas máximas de marzo y abril)
serie = datos$Tmax
serie.ts = ts(serie, frequency = 365) # frequency set to 12 to set stationarity each 12 months
plot(decompose(serie.ts))

```

Decomposition of additive time series



Observamos en la gráfica:

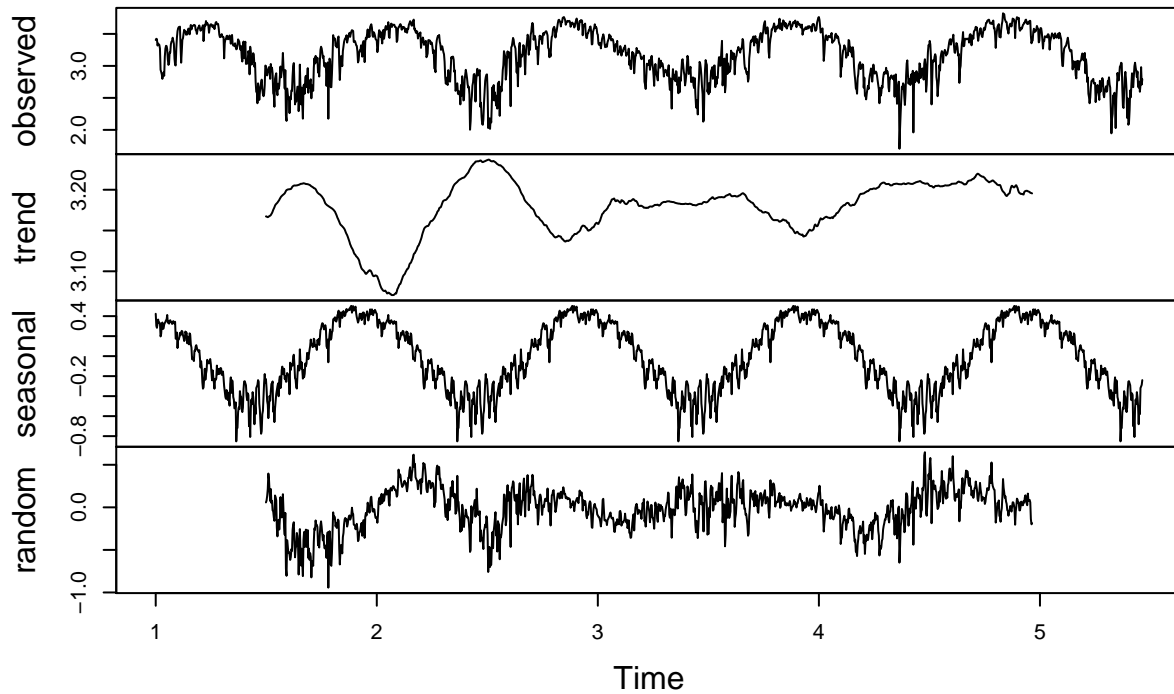
- los valores de la serie
- la tendencia calculada mediante filtros
- la estacionalidad repetida cada 12 instantes de tiempo
- lo que queda de la serie al eliminar tendencia y estacionalidad

Probamos, como hemos hecho en experimentos anteriores, a realizar una transformación logarítmica de la

serie para reducir la varianza y así evitar problemas con el cálculo de la estacionariedad:

```
serie.ts.log = log(serie.ts)
serie.log = log(serie)
plot(decompose(serie.ts.log))
```

Decomposition of additive time series



El cambio es relativamente muy pequeño, es posible que tenga mucho que ver la gran cantidad de datos, siendo difícil distinguir la variación entre la gráfica anterior y esta. Aún así, por precaución, tomaremos esta serie por válida.

Observamos los datos de estacionalidad que nos servirán en el futuro para eliminársela a la serie temporal.

```
head(decompose(serie.ts.log)$seasonal)
```

```
## [1] 0.4237612 0.3026066 0.2830425 0.3237159 0.3660154 0.3323542
```

Por simplicidad en el uso de la nomenclatura utilizare la variable serie en lugar de serie.log

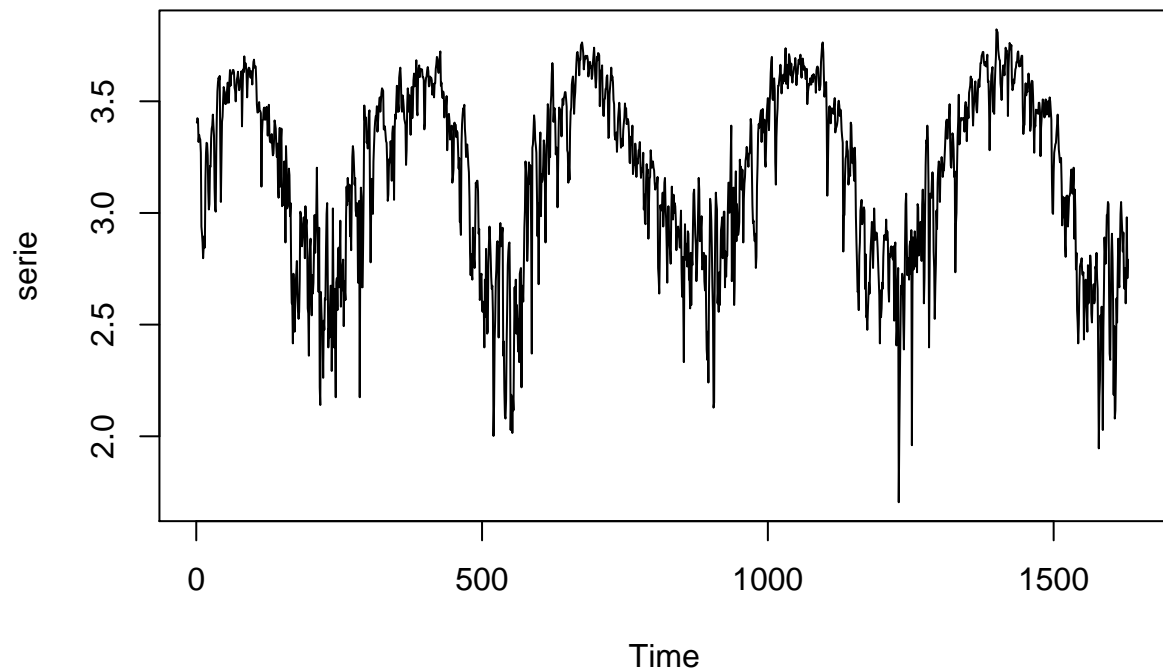
```
serie = serie.log
```

Calculamos los instantes de tiempo de la serie

```
# para train
tiempo = 1:length(serie.ts.log)
# para test
tiempoTs = (tiempo[length(tiempo)]+1):(tiempo[length(tiempo)]+Npred)
```

Representamos la serie

```
plot.ts(serie, xlim=c(1, tiempoTs[length(tiempoTs)]))
```



Modelamos la tendencia, la cual podemos aparentemente ajustarla de forma lineal, sabiendo que:

$\text{serie} = \text{parametroA} * \text{tiempo} + \text{parametroB}$

Con ayuda de la función `lm` calculamos dichos parámetros:

```
parametros.H1 = lm(serie ~ tiempo)
parametros.H1
```

```
##
## Call:
## lm(formula = serie ~ tiempo)
##
## Coefficients:
## (Intercept)      tiempo
##  3.213e+00   -5.016e-05
```

Tomamos Intercept como termino independiente (`parametroB`) y el otro como el que multiplica al tiempo para poder calcular la serie (`parametroA`). Para modelar la tendencia usamos la fórmula descrita antes: