

# Series temporales con zoo

Oscar Perpiñán Lamigueiro

Universidad Politécnica de Madrid

① Lectura de ficheros (sencillo)

② Lectura de datos (real)

③ Datos agregados

④ Datos desde una URL

# Descargamos datos de SIAR

- ▶ <http://eportal.magrama.gob.es/websiar/Inicio.aspx>
- ▶ **Estación:** Aranjuez, Madrid
- ▶ **Período:** 01/01/2004 a 31/12/2011
- ▶ **Variables:** Temperatura, Humedad, Viento, Lluvia, Radiación, ET

# Lectura de datos con read.table

- ▶ Primero configuramos el directorio de trabajo

```
## Entre las comillas hay que indicar el directorio en el que está el  
## repositorio (será visible la carpeta data/)  
setwd('~github/intro/')
```

## Lectura de datos con read.table

- ▶ A continuación leemos el fichero (usamos la versión final)

```
dats <- read.table('data/aranjuez.csv')  
head(dats)
```

```
dats <- read.table('data/aranjuez.csv', sep=',')  
head(dats)
```

```
dats <- read.table('data/aranjuez.csv', sep=',', header=TRUE)  
head(dats)
```

```
aranjuez <- read.csv('data/aranjuez.csv')  
head(aranjuez)
```

```
class(aranjuez)  
names(aranjuez)
```

# Visualización de datos

```
library(lattice)

xyplot(Radiation ~ TempAvg, data=aranjuez)

xyplot(Radiation ~ TempAvg, data=aranjuez,
       type=c('p', 'r'))

xyplot(Radiation ~ TempAvg + TempMax + TempMin,
       data = aranjuez, xlab='Temperature',
       type=c('p', 'r'), auto.key=TRUE,
       pch=16, alpha=0.5)
```

## Visualización de datos (advanced!)

```
library(RColorBrewer)

humidClass <- cut(aranjuez$HumidAvg, 4)
myPal <- brewer.pal(n=4, 'GnBu')

xyplot(Radiation ~ TempAvg + TempMax + TempMin,
       groups=humidClass, outer=TRUE,
       data = aranjuez, xlab='Temperature',
       layout=c(3, 1),
       scales=list(relation='free'),
       auto.key=list(space='right'),
       par.settings=custom.theme(pch=16,
                                alpha=0.8, col=myPal))
```

# Transformamos a serie temporal

```
library(zoo)

fecha <- as.POSIXct(aranjuez[,1],
                    format='%Y-%m-%d')

head(fecha)

aranjuez <- zoo(aranjuez[, -1], fecha)
class(aranjuez)
head(aranjuez)
```



## Leemos directamente como serie temporal

```
aranjuez <- read.zoo('data/aranjuez.csv',  
                    sep=',', header=TRUE)
```

```
header(aranjuez)
```

```
names(aranjuez)
```

```
summary(index(aranjuez))
```

① Lectura de ficheros (sencillo)

② Lectura de datos (real)

③ Datos agregados

④ Datos desde una URL

## Ahora con la versión original

- ▶ Primero descomprimos el archivo

```
unzip('data/InformeDatos.zip', exdir='data')
```

- ▶ Y ahora abrimos teniendo en cuenta codificación, separadores, etc.

```
aranjuez <- read.table("data/M03_Aranjuez_01_01_2004_31_12_2011.csv",  
                      fileEncoding = 'UTF-16LE',  
                      header = TRUE, fill = TRUE,  
                      sep = ';', dec = ",")
```

- ▶ Vemos el contenido

```
head(aranjuez)  
summary(aranjuez)  
names(aranjuez)
```

## Convertimos a serie temporal

- ▶ Sólo nos interesan algunas variables (indexamos por columnas)

```
tt <- as.Date(aranjuez$Fecha, format='%d/%m/%Y')
aranjuez <- zoo(aranjuez[, c(6, 7, 9, 11, 12, 16,
                           17, 19, 20, 22)],
               order.by=tt)
```

## Ajustamos los nombres (opcional)

```
names(aranjuez) <- c('TempAvg', 'TempMax',  
                    'TempMin', 'HumidAvg',  
                    'HumidMax', 'WindAvg',  
                    'WindMax', 'Radiation',  
                    'Rain', 'ET')
```

## Nuevamente mostramos datos

- ▶ Método simple

```
xyplot(aranjuez)
```

- ▶ Seleccionamos variables y superponemos

```
xyplot(aranjuez[,c("TempAvg", "TempMax", "TempMin")],  
       superpose=TRUE)
```

- ▶ Para cruzar variables hay que convertir a `data.frame`

```
xyplot(TempAvg ~ Radiation,  
       data=as.data.frame(aranjuez))
```

# Limpieza de datos

## ► Conversión de Unidades (MJ -> Wh)

```
aranjuez$G0 <- aranjuez$Radiation/3.6*1000  
xyplot(aranjuez$G0)
```

## ► Filtrado de datos

```
aranjuezClean <- within(as.data.frame(aranjuez), {  
  TempMin[TempMin>40] <- NA  
  HumidMax[HumidMax>100] <- NA  
  WindAvg[WindAvg>10] <- NA  
  WindMax[WindMax>10] <- NA  
})  
  
aranjuez <- zoo(aranjuezClean, index(aranjuez))
```

① Lectura de ficheros (sencillo)

② Lectura de datos (real)

③ Datos agregados

④ Datos desde una URL



# Media anual

- ▶ Primero definimos una función para extraer el año

```
Year <- function(x) as.numeric(format(x, "%Y"))  
  
Year(index(aranjuez))
```

- ▶ Y la empleamos para agrupar con aggregate

```
aranjuezY <- aggregate(aranjuez$G0, by=Year,  
                      FUN=mean, na.rm=TRUE)  
  
aranjuezY  
class(aranjuezY)
```

```
G0y <- aggregate(aranjuez$G0, by=Year,  
                FUN=mean, na.rm=TRUE)  
  
G0y
```

## Medias anuales usando cut

```
aggregate(aranjuez$G0, by=function(tt) cut(tt, 'year'),  
          FUN=mean, na.rm=TRUE)
```

# Medias mensuales

## ► Meses como números

```
Month <- function(x)as.numeric(format(x, "%m"))
```

```
Month(index(aranjuez))
```

```
G0m <- aggregate(aranjuez$G0, by=Month,  
                FUN=mean, na.rm=TRUE)
```

```
G0m
```

## ► Meses como etiquetas

```
months(index(aranjuez))
```

```
G0m <- aggregate(aranjuez$G0, by=months,  
                FUN=mean, na.rm=TRUE)
```

```
G0m
```

## Medias mensuales para cada año

- ▶ La función para agrupar es `as.yearmon`

```
as.yearmon(index(aranjuez))
```

```
G0ym <- aggregate(aranjuez$G0, by=as.yearmon,  
                  FUN=mean, na.rm=TRUE)
```

```
G0ym
```

① Lectura de ficheros (sencillo)

② Lectura de datos (real)

③ Datos agregados

④ Datos desde una URL

## Ejemplo: Lanai-Hawaii

```
URL <- "http://www.nrel.gov/midc/apps/plot.pl?site=LANAI&start=20090722&edy=19&
emo=11&eyr=2010&zenloc=19&year=2010&month=11&day=1&endyear=2010&endmonth=11&
endday=19&time=1&inst=3&inst=4&inst=5&inst=10&type=data&first=3&math=0&second
=-1&value=0.0&global=-1&direct=-1&diffuse=-1&user=0&axis=1"
## URL <- "data/NREL-Hawaii.csv"
```

```
DATE,HST,Global Horizontal [W/m^2],Direct Normal [W/m^2],Diffuse Horizontal [W/m^2],Air Temperature [deg C]
11/1/2010,06:32,4.87621,0,4.87621,14.67
11/1/2010,06:33,5.14142,0,5.14142,14.54
11/1/2010,06:34,1.42216,0,1.42216,14.43
11/1/2010,06:35,1.95135,0,1.95135,14.4
11/1/2010,06:36,2.44687,0,2.44687,14.55
11/1/2010,06:37,3.16990,0,3.16990,14.95
11/1/2010,06:38,3.99677,0,3.99677,15.45
11/1/2010,06:39,4.88811,0,4.88811,15.71
11/1/2010,06:40,5.85428,0,5.85428,15.8
11/1/2010,06:41,8.27598,0,8.27598,15.87
```

# Leemos como serie temporal

- ▶ Leemos con `read.zoo`

```
lat <- 20.77
lon <- -156.9339
hawaii <- read.zoo(URL,
  col.names = c("date", "hour",
    "G0", "B", "D0", "Ta"),
  ## Dia en columna 1, Hora en columna 2
  index = list(1, 2),
  ## Obtiene escala temporal de estas dos columnas
  FUN = function(d, h) as.POSIXct(
    paste(d, h),
    format = "%m/%d/%Y %H:%M",
    tz = "HST"),
  header=TRUE, sep=",")
```

- ▶ Añadimos Directa en el plano Horizontal

```
hawaii$B0 <- with(hawaii, G0-D0)
```

## Mostramos datos como serie temporal

```
xyplot(hawaii)  
xyplot(hawaii[,c('G0', 'D0', 'B0')],  
       superpose=TRUE)
```



## Mostramos relaciones entre variables

```
xyplot(Ta ~ G0 + D0 + B0,  
       data=as.data.frame(hawaii),  
       type=c('p', 'smooth'),  
       par.settings=custom.theme(  
         alpha=.5, pch=16,  
         lwd=3, col.line='black'),  
       outer=TRUE, layout=c(3, 1),  
       scales=list(x=list(relation='free')))
```

# Irradiación horaria

## ► Primer intento

```
hour <- function(x) as.numeric(format(x, '%H'))
```

```
G0h <- aggregate(hawaii$G0, by=hour,  
                FUN=sum, na.rm=1)/1000
```

```
G0h
```

# Irradiación horaria

► Mejor así

```
hour <- function(x) as.POSIXct(format(x,  
                                     '%Y-%m-%d %H:00:00'))
```

```
G0h <- aggregate(hawaii$G0, by=hour,  
                FUN=sum, na.rm=1)/60
```

```
G0h
```

# Irradiación diaria

- ▶ A partir de la horaria

```
G0d <- aggregate(G0h,  
                 by=function(x)format(x, '%Y-%m-%d'),  
                 sum)/1000
```

- ▶ A partir de la minutaria

```
day <- function(x)format(x, '%Y-%m-%d')  
G0d <- aggregate(hawaii$G0, by=day,  
                 sum)/60/1000  
  
G0d  
  
truncDay <- function(x)as.POSIXct(trunc(x, units='day'))  
G0d <- aggregate(hawaii$G0, by=truncDay,  
                 sum)/60/1000  
  
G0d
```

## Más complicado: agrupar por 30 minutos

```
halfHour <- function(tt, delta=30){  
  tt <- as.POSIXlt(tt)  
  gg <- tt$min %/% delta  
  tt <- modifyList(tt, list(min=gg*delta))  
  as.POSIXct(tt)  
}
```

```
hawaii30 <- aggregate(hawaii, by=halfHour,  
                      FUN=sum)/60  
  
head(hawaii30)
```