

Caso práctico I: Anomalías de horas de sol

Curso de Invierno UCM - Analizando datos con R | Dr. Dominic Royé

02 febrero 2022

Índice

1. Los formatos multidimensionales	1
2. Paquetes	1
3. Importación de archivos	2
4. Cálculo de variables derivadas	2
5. Anomalías temporales	4
6. Extracción de datos puntuales y series temporales	4
7. Visualización de las horas de sol	5

1. Los formatos multidimensionales

NetCDF es un formato para crear y distribuir arrays de datos en formato *grid* o rejilla. Originalmente fue creado para guardar y distribuir información climática como la generada para los modelos de simulaciones o reanálisis, pero después se extendió a otros tipos de datos. Las librerías de netCDF son mantenidas por Unidata <http://www.unidata.ucar.edu/software/netcdf/> y existen varias aplicaciones para visualizar este tipo de archivos como Panoply, <http://www.giss.nasa.gov/tools/panoply/>. Los paquetes `ncdf`, `ncdf4` y `raster` en R se utilizan para crear y almacenar datos en este formato.

Sin embargo, el paquete `ncdf4`, que es el más utilizado, no es muy intuitivo y se hace un poco complicado para niveles de iniciación, así que trabajaremos a través de un caso de estudio utilizando el paquete **terra** que ya hemos visto al trabajar con ráster.

2. Paquetes

El único nuevo paquete introducido en este ejemplo es `{fs}` que proporciona una interfaz uniforme y multiplataforma para las operaciones del sistema de archivos.

```
library(lubridate)
library(tidyverse)
library(terra)
library(rnaturalearth)
library(fs) # nuevo paquete
library(sf)
```

3. Importación de archivos

Vamos a trabajar con la suma de horas de sol mensuales en la Península Ibérica y norte de África. Como se trata de varios archivos que hemos extraído de https://wui.cmsaf.eu/safira/action/viewProduktDetails?eid=21846_22004&fid=28 y https://wui.cmsaf.eu/safira/action/viewProduktDetails?eid=22018_22019&fid=27, lo primero que tenemos que hacer es descomprimirlos.

```
comprimidos <- dir_ls("data", regexp = "tar")

walk(comprimidos, untar, exdir = "./data/archivos")
```

Ahora ya podemos cargarlos en nuestra sesión como un SpatRaster.

```
files <- dir_ls("./data/archivos")
# head(files)
r <- rast(files)
r

## class      : SpatRaster
## dimensions : 301, 301, 456 (nrow, ncol, nlyr)
## resolution : 0.05, 0.05 (x, y)
## extent     : -10.025, 5.025, 29.975, 45.025 (xmin, xmax, ymin, ymax)
## coord. ref.: lon/lat WGS 84
## sources    : SDUms198301010000003UD1000101UD.nc
##             SDUms198302010000003UD1000101UD.nc
##             SDUms198303010000003UD1000101UD.nc
##             ... and 453 more source(s)
## varnames   : SDU (Sunshine Duration)
##             SDU (Sunshine Duration)
##             SDU (Sunshine Duration)
##             ...
## names      : SDU, SDU, SDU, SDU, SDU, SDU, ...
## unit       : h, h, h, h, h, h, ...
## time       : 1983-01-01 to 2020-12-01
```

4. Cálculo de variables derivadas

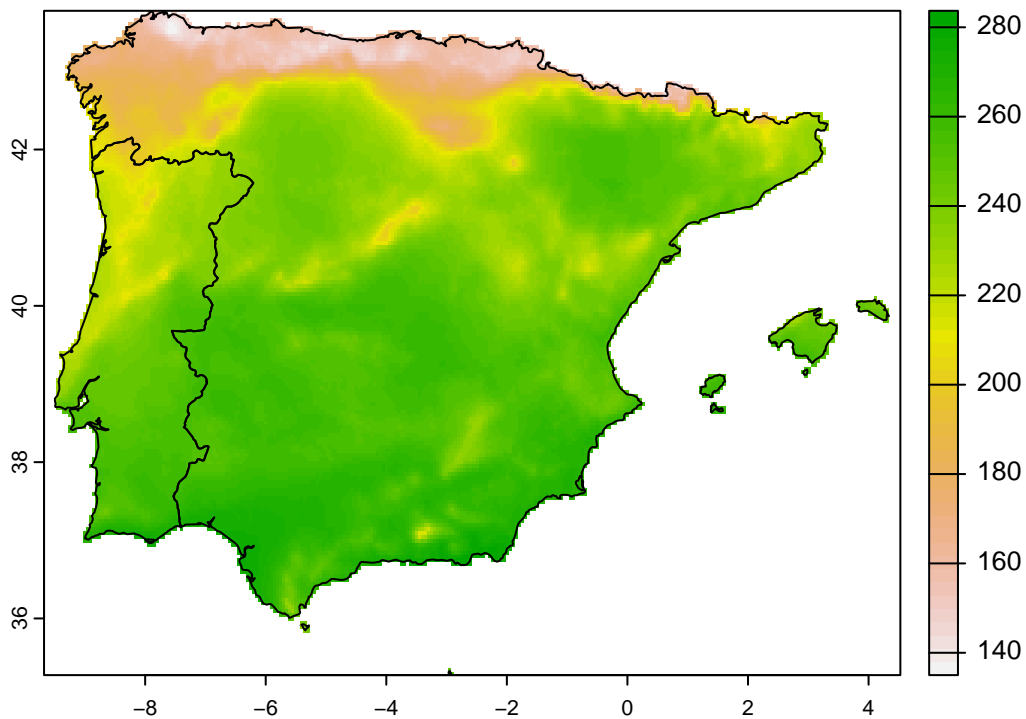
Una vez que tenemos todas las capas en un solo objeto, podemos operar con él a nuestro antojo. Por ejemplo, si hacemos la media de todas ellas, obtendremos la media mensual de horas de sol para cada pixel. La función `app()` es muy útil.

```
# media de todos los meses
r_media_mens <- app(r, mean)

# límites
pi <- ne_countries(scale = 10, country = c("Spain", "Portugal"),
  returnclass = "sf") %>%
  st_crop(xmin = -10, xmax = 5, ymin = 35, ymax = 44)

# recortar y enmascarar
r_media_mens <- crop(r_media_mens, pi) %>%
  mask(vect(pi))

# mapa
plot(r_media_mens)
plot(pi, add = TRUE, col = NA)
```



Podemos hacer lo mismo para todos los años, aunque el proceso es algo diferente, ya que primero tenemos que sumar todas las horas de sol de cada año, y después hacer la media de todos ellos.

```
# media de todos los años
dates <- seq(ymd("1983-01-01"), ymd("2020-12-01"), "month")
years <- year(dates)

# suma anual
r_year <- tapp(r, index = years, fun = sum)
r_year

## class      : SpatRaster
## dimensions  : 301, 301, 38  (nrow, ncol, nlyr)
## resolution  : 0.05, 0.05  (x, y)
## extent     : -10.025, 5.025, 29.975, 45.025  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84
## source      : memory
## names       : X1983, X1984, X1985, X1986, X1987, X1988, ...
## min values  : 1533.868, 1320.441, 1552.508, 1397.506, 1377.239, 1484.483, ...
## max values  : 3677.823, 3767.841, 3714.458, 3752.956, 3568.362, 3656.469, ...

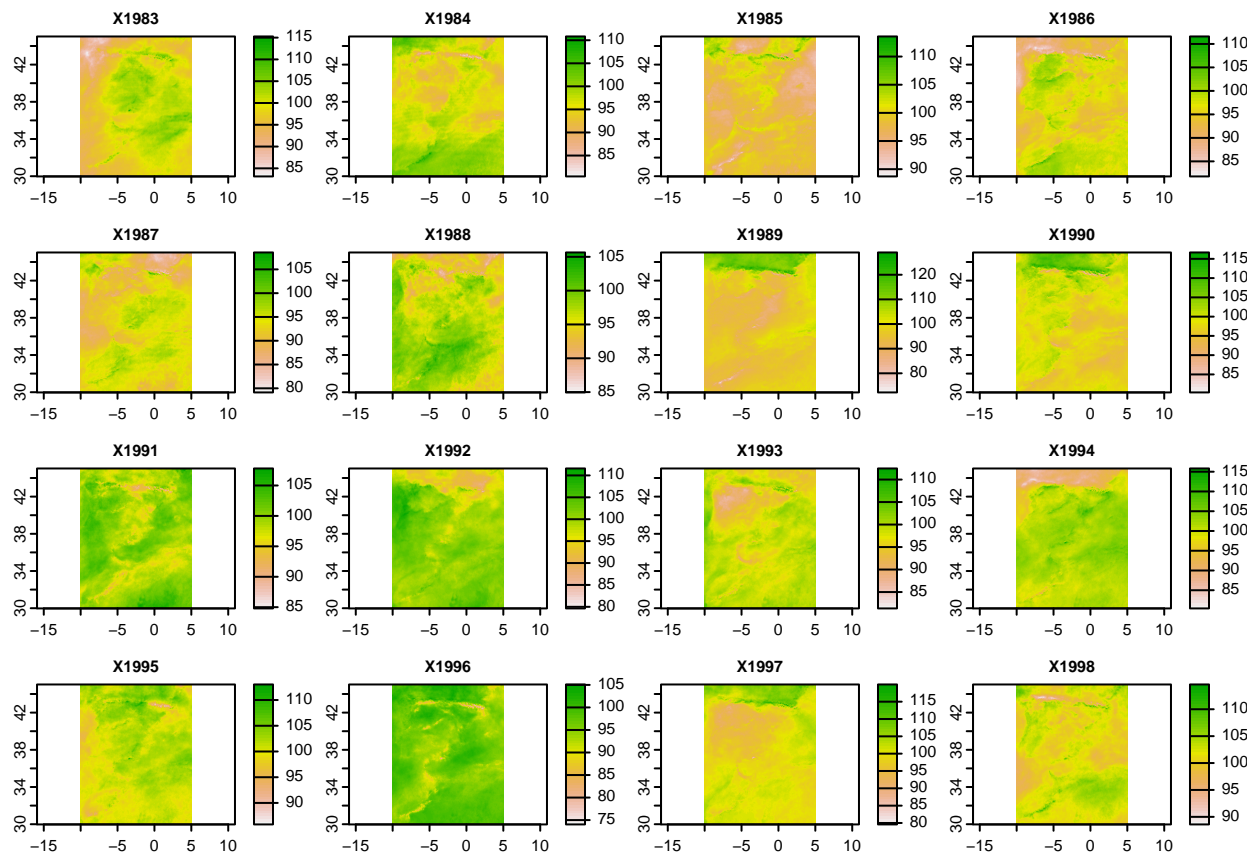
# promedio anual
r_media_year <- app(r_year, mean)
```

5. Anomalías temporales

Una vez que tenemos la media, podemos calcular, por ejemplo, las anomalías anuales. Es decir, la desviación de cada año respecto a la media del total.

```
# anomalías anuales
r_anom_year <- r_year * 100/r_media_year

# mapa
plot(r_anom_year)
```



6. Extracción de datos puntuales y series temporales

También podemos extraer toda esta información para puntos concretos que nos interesen. Por ejemplo, para un conjunto de ciudades.

```
# puntos de interés
puntos <- data.frame(ciudad = c("SAN", "ZGZ", "SEV", "OPO", "MAR",
                                "MAL"), lon = c(-8.5, -0.9, -6, -8.3, -8, 2.6), lat = c(42.8,
                                41.6, 37.4, 41.2, 31.6, 39.6))

# extracción
media_anual <- terra::extract(r_media_year, puntos[, 2:3])
head(media_anual)

##   ID      mean
## 1  1 2229.647
```

```
## 2 2 3019.635
## 3 3 3261.630
## 4 4 2596.018
## 5 5 3359.162
## 6 6 3017.600

# añadir la extracción a la tabla
puntos <- mutate(puntos, media_anual[, 2]) %>%
  rename(horas_sol = 4)

puntos

##   ciudad lon lat horas_sol
## 1   SAN -8.5 42.8 2229.647
## 2   ZGZ -0.9 41.6 3019.635
## 3   SEV -6.0 37.4 3261.630
## 4   OPO -8.3 41.2 2596.018
## 5   MAR -8.0 31.6 3359.162
## 6   MAL  2.6 39.6 3017.600
```

7. Visualización de las horas de sol

En este ejemplo práctico haremos un gráfico de facetas con las anomalías de horas de sol desde el año 1983 al 2020. Aquí introducimos dos nuevos paquetes: *{ggtext}* para una representación de texto mejorado en *ggplot2* y *{classInt}* para hacer categorización basado en diferentes algoritmos.

```
# paquetes adicionales
library(ggtext)
library(classInt)
```

El primer paso consiste en convertir el raster multidimensional en un *data.frame* con coordenadas y columnas de cada capa.

```
# recortamos y enmascaramos
r_anom_year <- crop(r_anom_year, pi) %>%
  mask(vect(pi))

# raster a data.frame
anom_mat <- as.data.frame(r_anom_year, xy = TRUE)

# renombramos las columnas
names(anom_mat)[3:40] <- str_c("yr", 1983:2020)

# estructura final str(anom_mat)
```

En el siguiente paso, debemos convertir la tabla en un formato *tidy*, tabla larga y limpiar los nombres de las antiguas columnas.

```
# formato largo de tabla
anom_mat <- pivot_longer(anom_mat, 3:40, names_to = "yr", values_to = "anom") %>%
  mutate(yr = str_remove(yr, "yr") %>%
    as.numeric())

# resumen
summary(anom_mat)
```

```
##           x           y           yr           anom
```

```
## Min.      :-9.450   Min.      :35.30   Min.      :1983   Min.      : 74.04
## 1st Qu.   :-6.550   1st Qu. :38.70   1st Qu. :1992   1st Qu. : 96.86
## Median    :-4.350   Median   :40.40   Median   :2002   Median  :100.00
## Mean      :-4.139   Mean     :40.28   Mean     :2002   Mean    :100.00
## 3rd Qu.   :-1.950   3rd Qu. :41.90   3rd Qu. :2011   3rd Qu. :103.00
## Max.      : 4.300   Max.     :43.75   Max.     :2020   Max.    :125.98
```

Antes de pasar a construir los mapas, creamos las clases de las anomalías. En este caso son elegidas de forma manual con apoyo en la función `classIntervals`.

```
# vector de las anomalías
anom_val <- anom_mat$anom/100
# posibles clases
classInt::classIntervals(anom_val[anom_val > 1.03], 4)

## style: quantile
##      [1.03,1.039712) [1.039712,1.052932) [1.052932,1.070045) [1.070045,1.259833]
##                60765                60764                60765                60765

# clases en ratio (NO%)
cutp <- c(0.7, 0.9, 0.93, 0.95, 0.97, 1.03, 1.05, 1.08, 1.1,
          1.15, 1.2, 1.3)

# nuestros colores
col_reds <- colorRampPalette(c("#67001f", "#b2182b", "#d6604d",
                              "#f4a582", "#fddbc7"))
col_rb <- c(col_reds(6), "#f7f7f7", "#d1e5f0", "#92c5de", "#4393c3",
            "#2166ac") # con azules

# clasificamos las anomalías
anom_mat <- mutate(anom_mat, anom_cut = cut(anom/100, cutp)) # ratio!!
```

En la construcción de este mapa haremos uso del paquete `ggtext` para mejorar el aspecto de texto, lo que nos permite en dar color a diferentes partes del texto o formatearlo en base a *markdown* o *html*.

```
# primera parte
g <- ggplot() +
  geom_raster(data = anom_mat,
             aes(x, y,
                 fill = anom_cut)) +
  geom_sf(data = pi,
          fill = "transparent",
          colour = "grey70",
          size = 0.05) +
  facet_wrap(yr ~ .,
             ncol = 7,
             strip.position = "bottom") +
  coord_sf(xlim = c(-10, 5), ylim = c(35.96, 44))

# segunda parte
g <- g + scale_fill_manual(values = rev(col_rb),
                          labels = scales::percent_format(accuracy = 1)) +
  guides(fill = guide_colorsteps(barwidth = 20, barheight = .4))

# tercera parte
```

```

g <- g + labs(x = "",
              y = "",
              fill = "",
              title = "¿Qué años fueron
<span style='color:#b2182b;'><strong>más</strong></span> o
<span style='color:#2166ac;'><strong>menos</strong></span>
soleados en la Península Ibérica?",
              subtitle = "Anomalía de horas anuales de sol 1983-2020.
Período normal de referencia 1983-2010.",
              caption = "Datos: EUMETSAT/CMSAF")

# definiciones finales
g <- g + theme_minimal() +
  theme(legend.position = "top",
        panel.grid = element_blank(),
        panel.spacing = unit(1, "lines"),
        axis.text = element_blank(),
        legend.justification = "left",
        plot.caption = element_text(size = 12,
                                     margin = margin(b = 5, t = 10, unit = "pt")),

        plot.title = element_textbox(size = 18,
                                     margin = margin(b = 1, t = 2, unit = "pt")),

        plot.subtitle = element_textbox(size = 12,
                                     margin = margin(b = 7, t = 2, unit = "pt")),

        strip.text = element_text(margin = margin(t = 0.3, unit = "line"),
                                   size = 12, hjust = 0.3, face = "bold"))

```

Para finalizar exportamos nuestro gráfico a png.

```

ggsave("horas_sol_anomalias.png", g, width = 15, height = 12,
       units = "in", bg = "white")

```

Otra posibilidad sería mostrar anomalías manuales de un año concreto como aquí de 2021.

No sólo eso, una variante sería mostrar las diferencias en un gráfico de densidades.