Datos espacio-temporales II: Raster

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

02 febrero 2022

Índice

Hasta ahora (y en este momento también), el paquete de funciones más usado para trabajar con datos ráster es **raster**. Sin embargo, el nuevo paquete **terra** está llamado a sustituirlo por su mayor velocidad de procesamiento y su interoperabilidad con otros paquetes. La sintaxis es prácticamente la misma, por lo que no cuesta nada adaptarse a él.

1. Creación de objetos raster

Podemos crear fácilmente un objeto de clase raster dándole una estructura de latitud/longitud e introduciendo (o no) los datos dentro de este objeto.

```
library(terra)
```

```
## terra 1.5.17
```

```
x <- rast()
x
```

```
## class       : SpatRaster
## dimensions  : 180, 360, 1  (nrow, ncol, nlyr)
## resolution  : 1, 1  (x, y)
## extent      : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84
```

```
x <- rast(ncol = 36, nrow = 18, xmin = -1000, xmax = 1000, ymin = -100,
    ymax = 900)
res(x)  #nos permite ver la resolución espacial
```

```
## [1] 55.55556 55.55556
```

```
res(x) <- 100
```

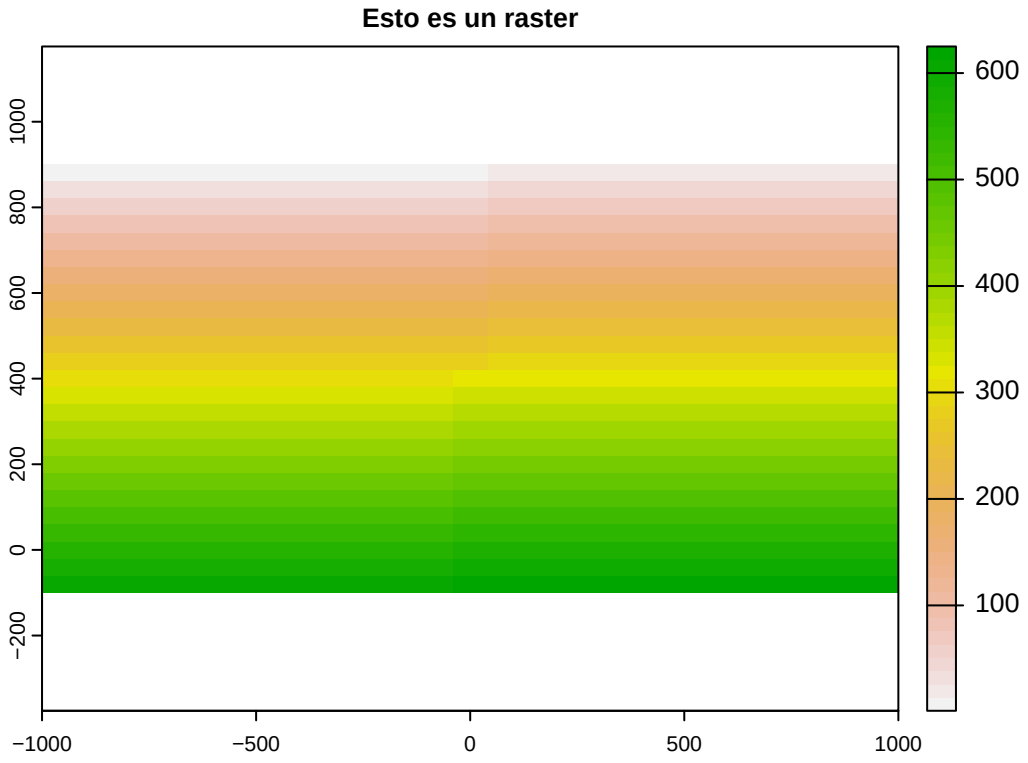
```
crs(x) <- "+proj=utm +zone=48 +datum=WGS84"
x
```

```
## class       : SpatRaster
## dimensions  : 10, 20, 1  (nrow, ncol, nlyr)
## resolution  : 100, 100  (x, y)
## extent      : -1000, 1000, -100, 900  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=48 +datum=WGS84 +units=m +no_defs
```

```
ncol(x) <- 25
nrow(x) <- 25

values(x) <- 1:ncell(x)
values(x)[1:30]
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30
```

```
plot(x, main = "Esto es un raster")
```



2. Importación

```
# Cargamos la temperatura media anual a 5min de res. esp.
# extraida de WorldClim
archivo <- "./data/wc2.1_5m_bio_1.tif"
```

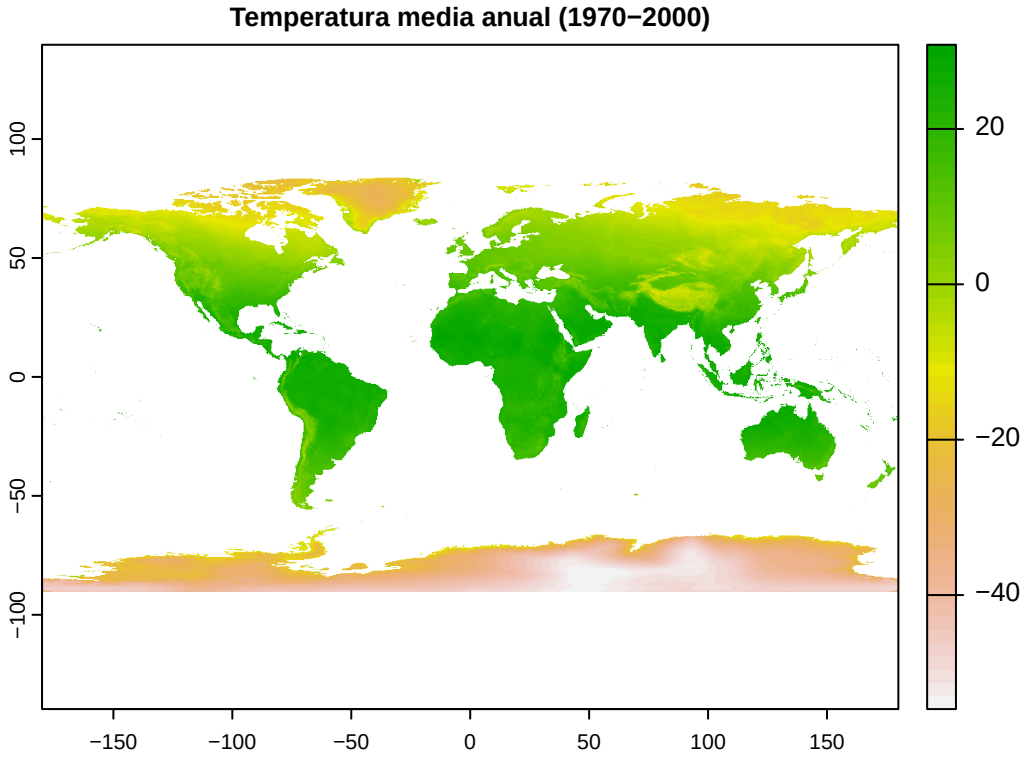
```
r <- rast(archivo)
sources(r)
```

```
## [1] "D:/OneDriveUSC/OneDrive - Universidade de Santiago de Compostela/Escritorio/GIS con R_Madrid/02_
```

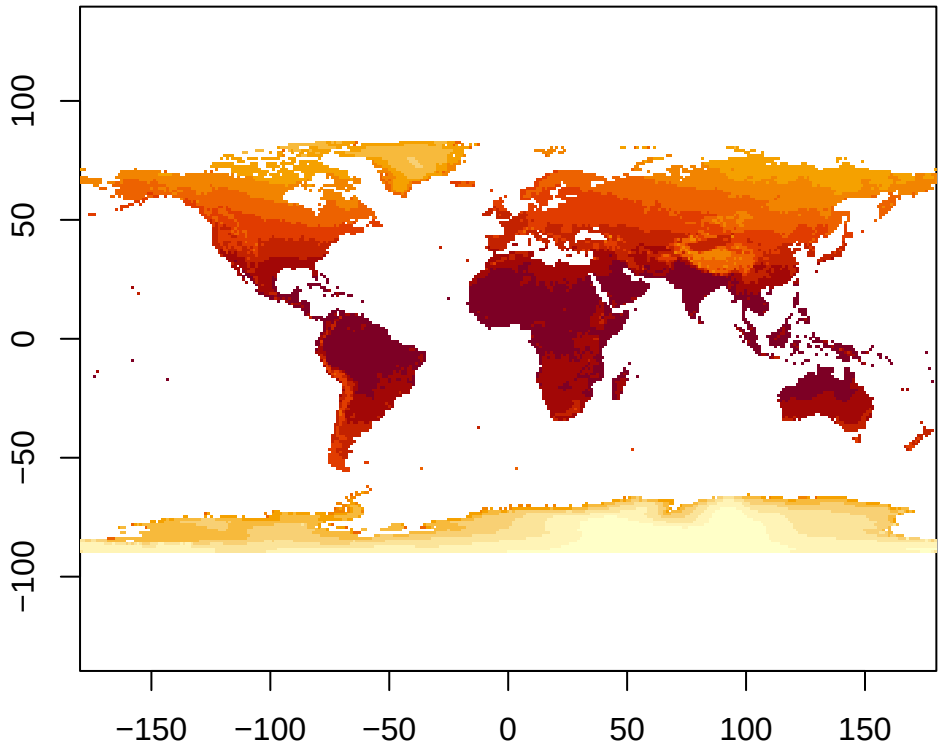
```
hasValues(r)
```

```
## [1] TRUE
```

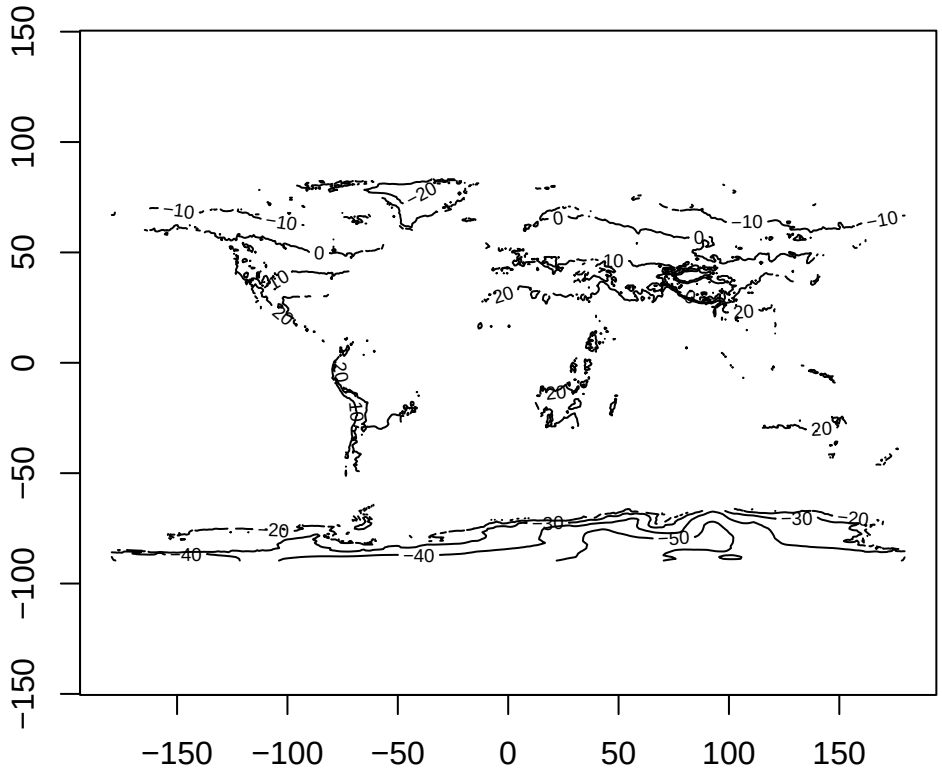
```
plot(r, main = "Temperatura media anual (1970-2000)")
```



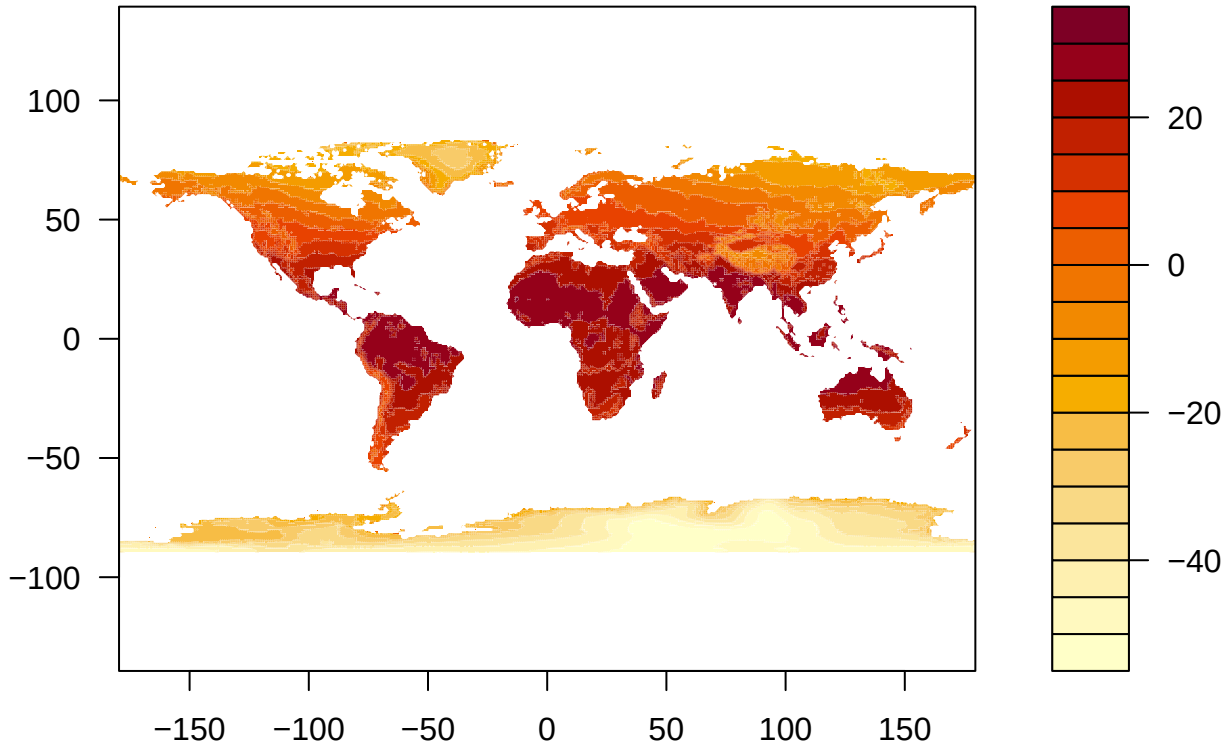

```
terra::image(r)
```




```
terra::contour(r)
```



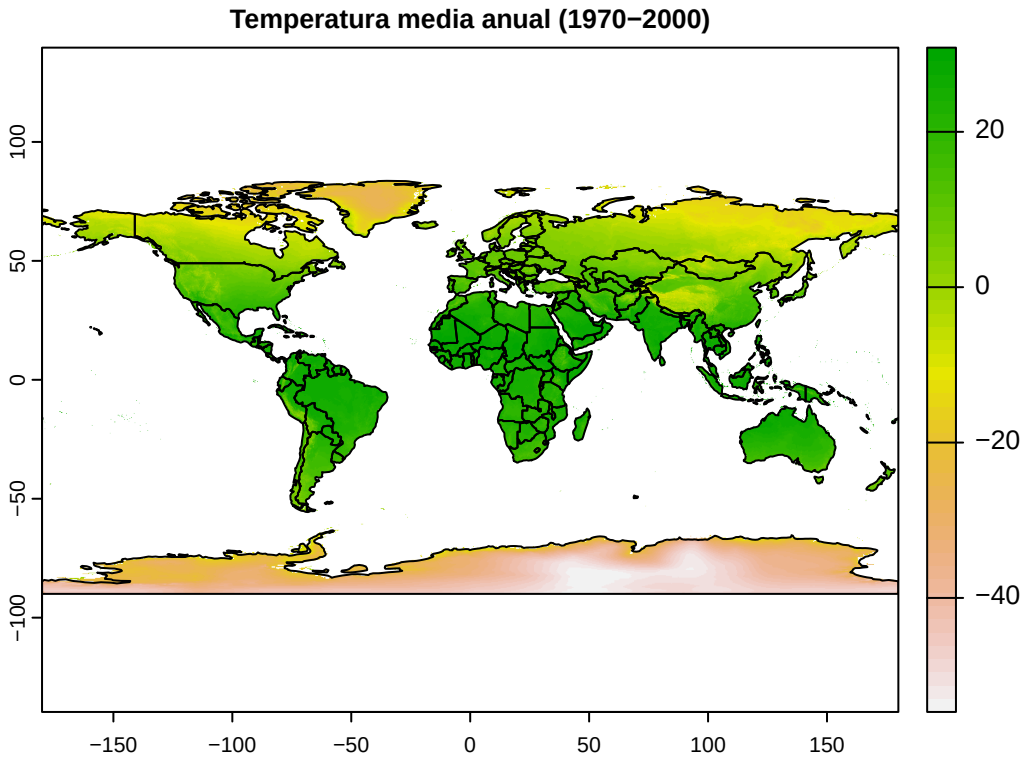

```
terra::contour(r, filled = TRUE)
```



Si queremos añadir en este caso los límites de los países para tenerlos como referencia, lo más sencillo es usar las capas ue viene por defecto en el paquete `maptools`.

```
library(rnaturalearth)

wld <- ne_countries(returnclass = "sf")

plot(r, main = "Temperatura media anual (1970-2000)")
plot(wld, add = TRUE, col = NA)
```

```
## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute
```



Los objetos *SpatRaster* permiten trabajar con objetos multi-capa, es decir, una superposición de capas raster tal y como nos podríamos encontrar por ejemplo en un archivo raster multibanda:

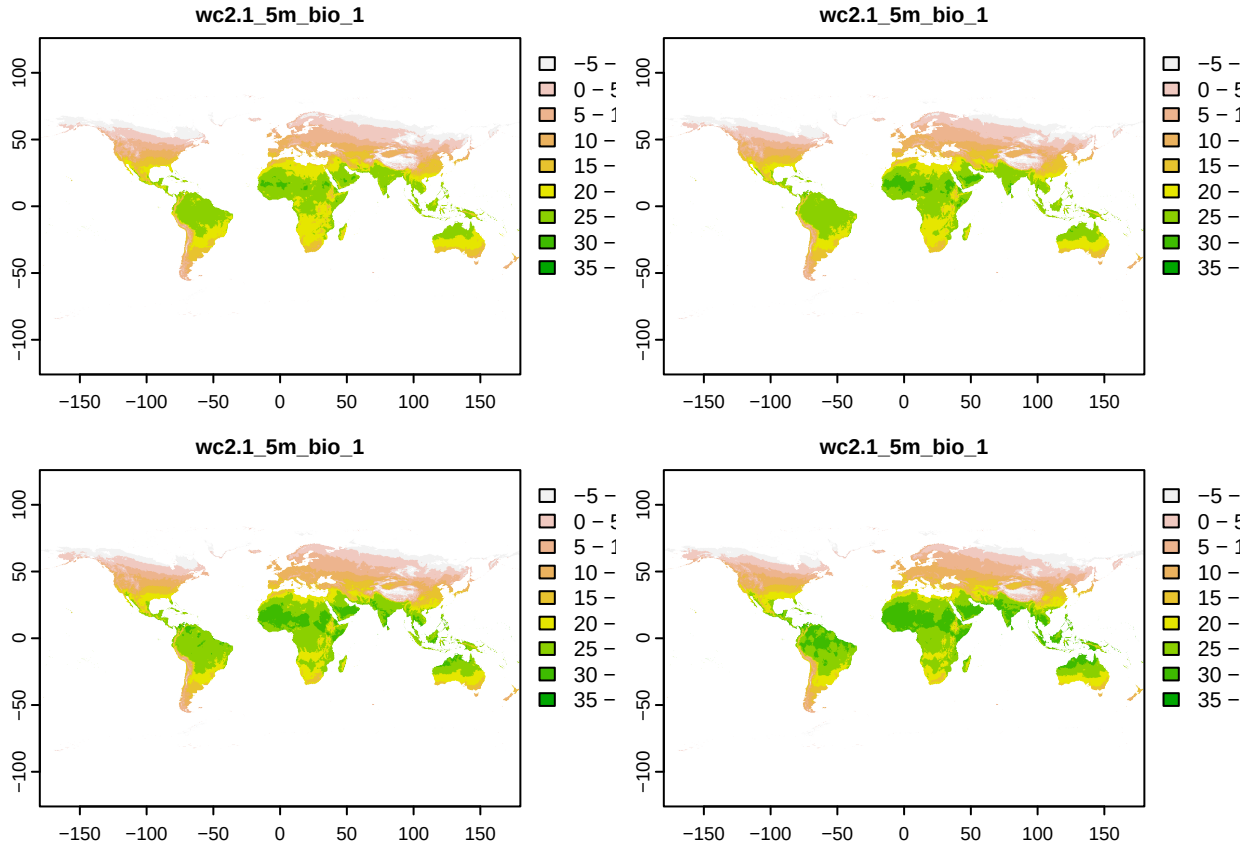
```
r1 <- r + 1
r2 <- r + 2
r3 <- r + 3
r4 <- r + 4

# Podemos combinarlos en uno solo
s <- c(r1, r2, r3, r4)
s
```

```
## class       : SpatRaster
## dimensions  : 2160, 4320, 4  (nrow, ncol, nlyr)
## resolution  : 0.08333333, 0.08333333  (x, y)
## extent      : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## sources     : memory
##               memory
##               memory
##               ... and 1 more source(s)
## names       : wc2.1_5m_bio_1, wc2.1_5m_bio_1, wc2.1_5m_bio_1, wc2.1_5m_bio_1
## min values  :      -53.73946,      -52.73946,      -51.73946,      -50.73946
## max values  :       32.05112,       33.05112,       34.05112,       35.05112
```

```
nlyr(s)
```

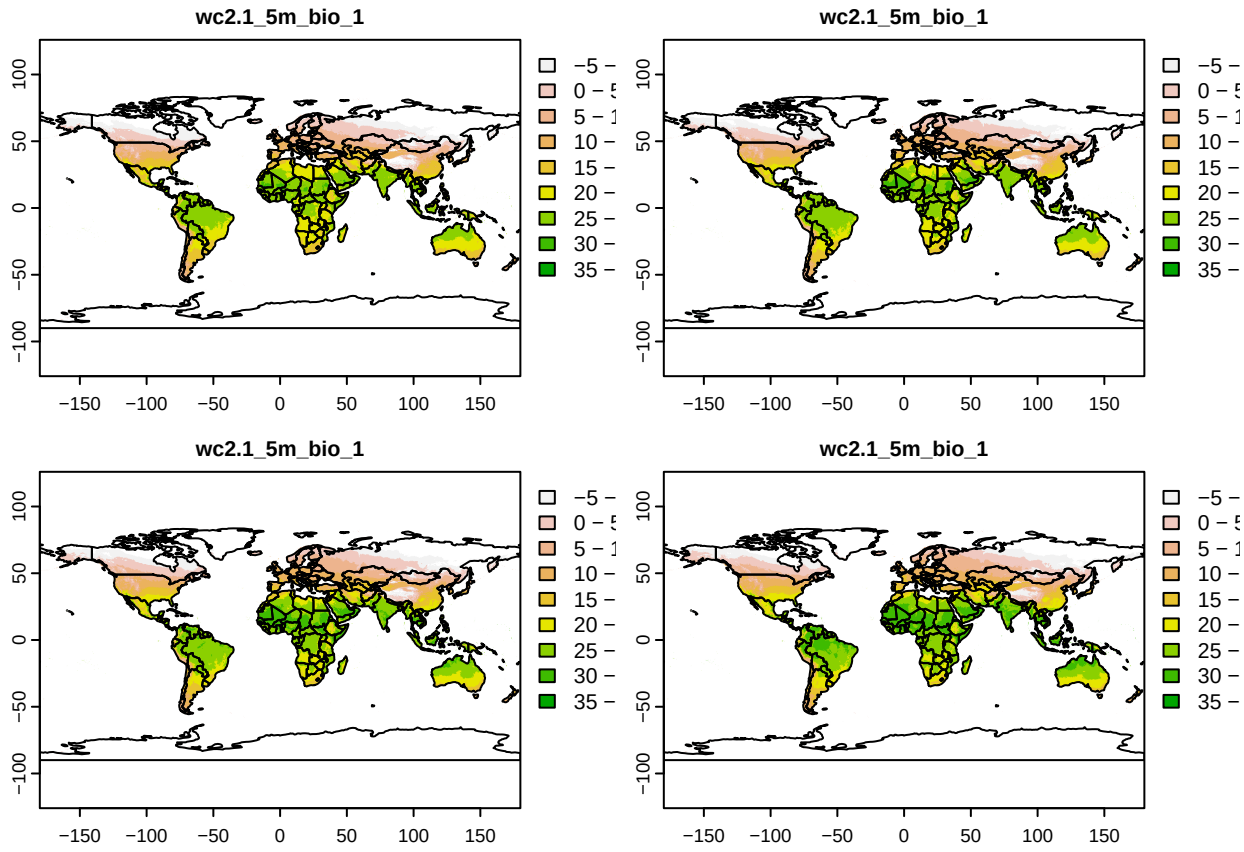
```
## [1] 4
```

```
brks <- seq(-5, 40, 5)
cols <- rev(terrain.colors(length(brks) - 1))
plot(s, breaks = brks, col = cols)
```




```
# si queremos añadir los países:
paises <- function() {
    plot(wld, add = TRUE, col = NA)
}
plot(s, breaks = brks, col = cols, fun = paises)
```

```
## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute

## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute

## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute

## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute
```




```
# en caso de que tengas problemas de RAM rm(r1, r2, r3, r4)
# gc()
```

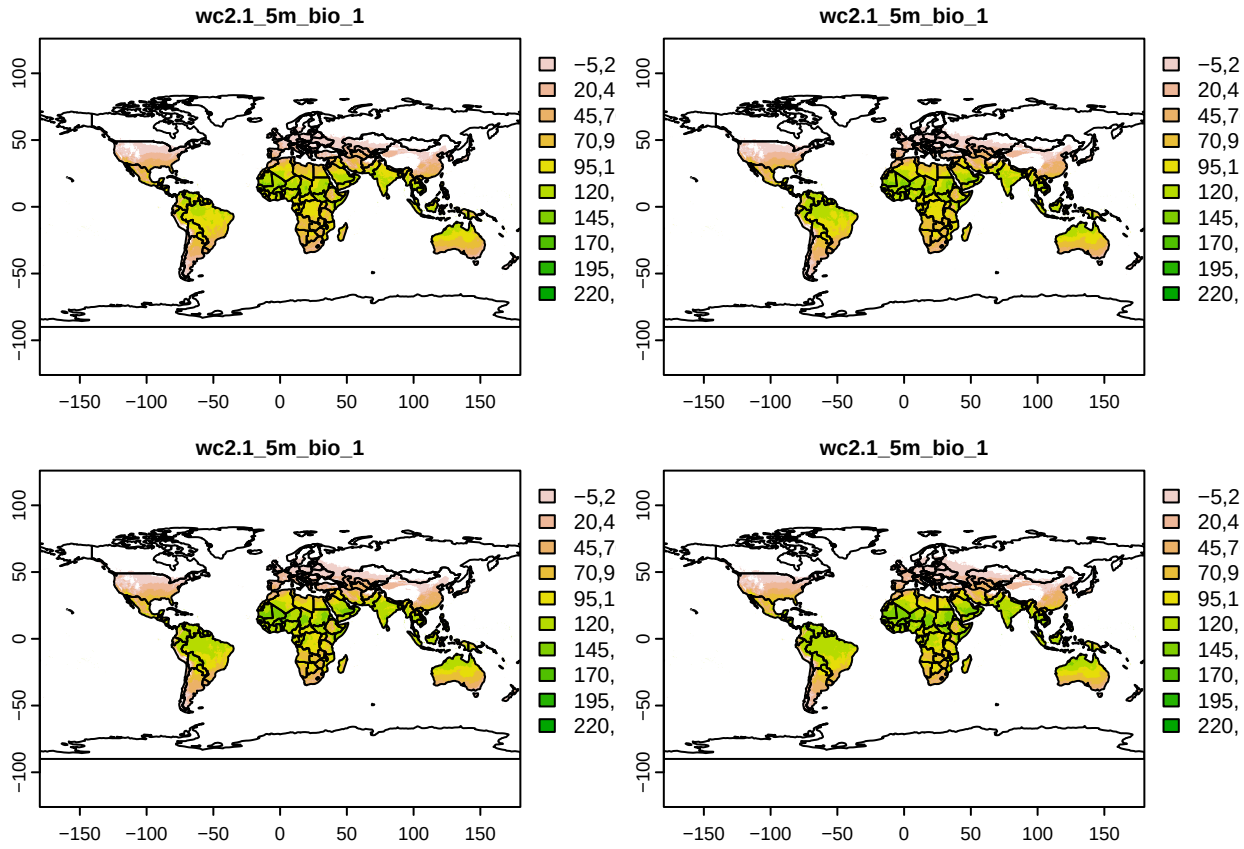
3. Manipulación

3.1. Álgebra raster

```
# dim(s)

s <- s * sqrt(r) + 5
s[s == 5] <- 15


sclas <- classify(s, seq(-5, 250, 25))
plot(sclas, type = "interval", breaks = 1:11, plg = list(legend = c("-5,20",
    "20,45", "45,70", "70,95", "95,120", "120,145", "145,170",
    "170,195", "195,220", "220,245")), fun = paises)
```

```
## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute

## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute

## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
```

```
## attribute

## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute
```

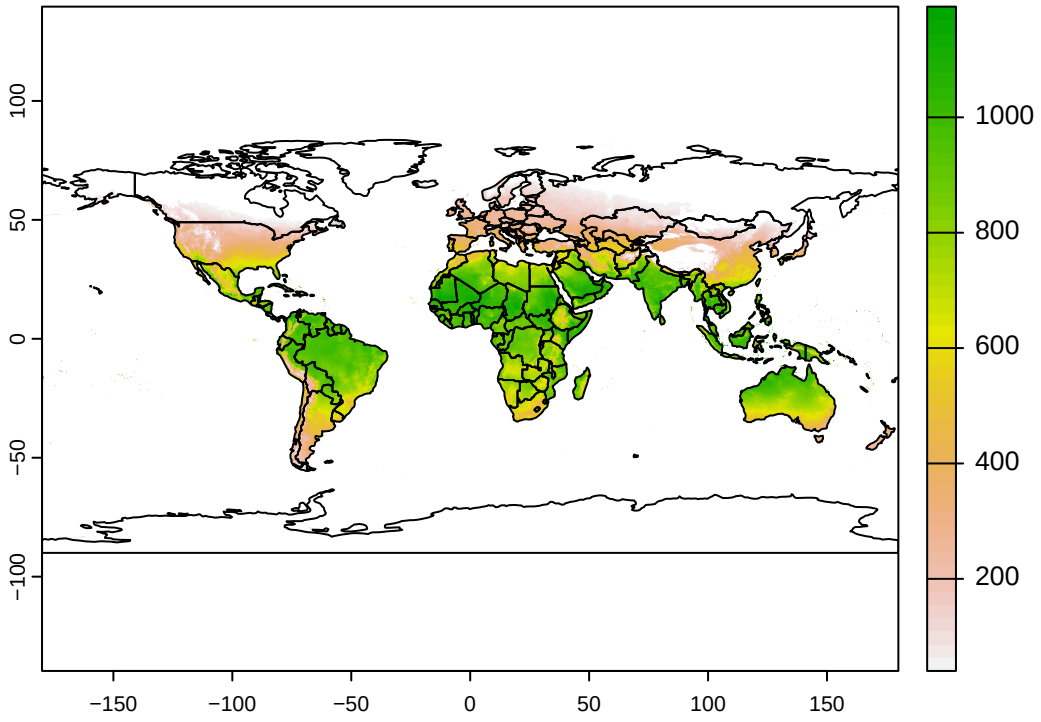



```
a <- mean(r, s, 10)
b <- sum(r, s)
all <- c(a, b)
all_sum <- sum(all)
all_sum
```

```
## class       : SpatRaster
## dimensions  : 2160, 4320, 1  (nrow, ncol, nlyr)
## resolution  : 0.08333333, 0.08333333  (x, y)
## extent      : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source      : memory
## name        :      sum
## min value   : 40.00227
## max value   : 1202.719
```

```
plot(all_sum, fun = paises)
```

```
## Warning in plot.sf(wld, add = TRUE, col = NA): ignoring all but the first
## attribute
```



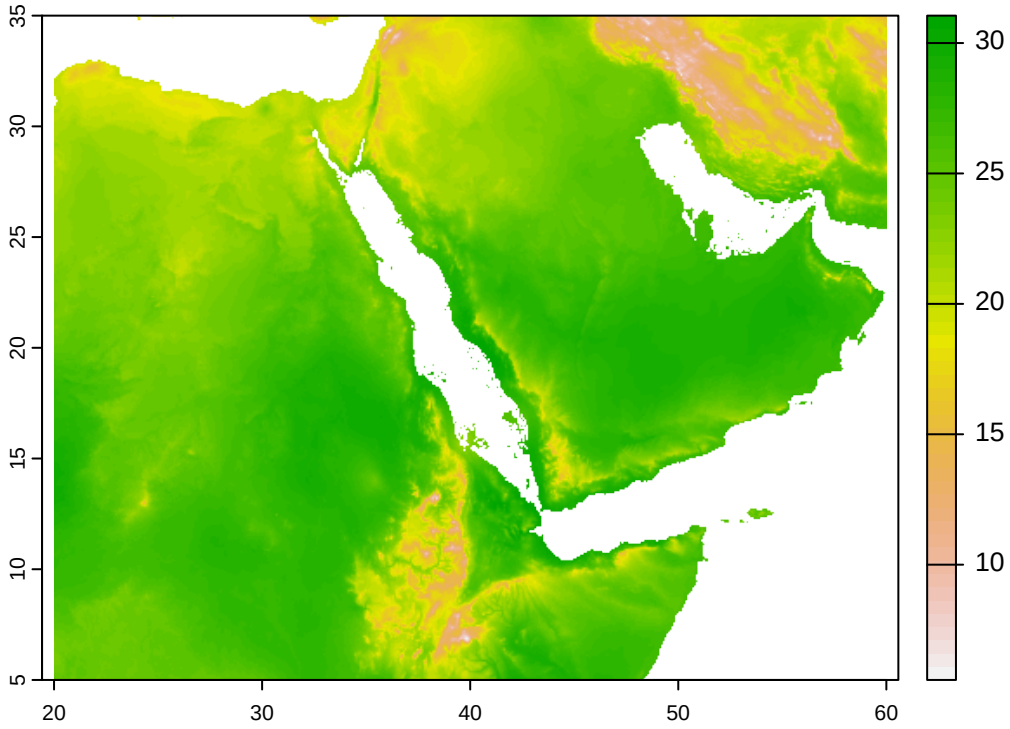
Usando `global( )` obtenemos el valor individual en lugar de un valor por cada pixel.

```
global(s, "mean", na.rm = TRUE)
```

```
##                       mean
## wc2.1_5m_bio_1   85.91929
## wc2.1_5m_bio_1.1 89.79189
## wc2.1_5m_bio_1.2 93.66448
## wc2.1_5m_bio_1.3 97.53708
```

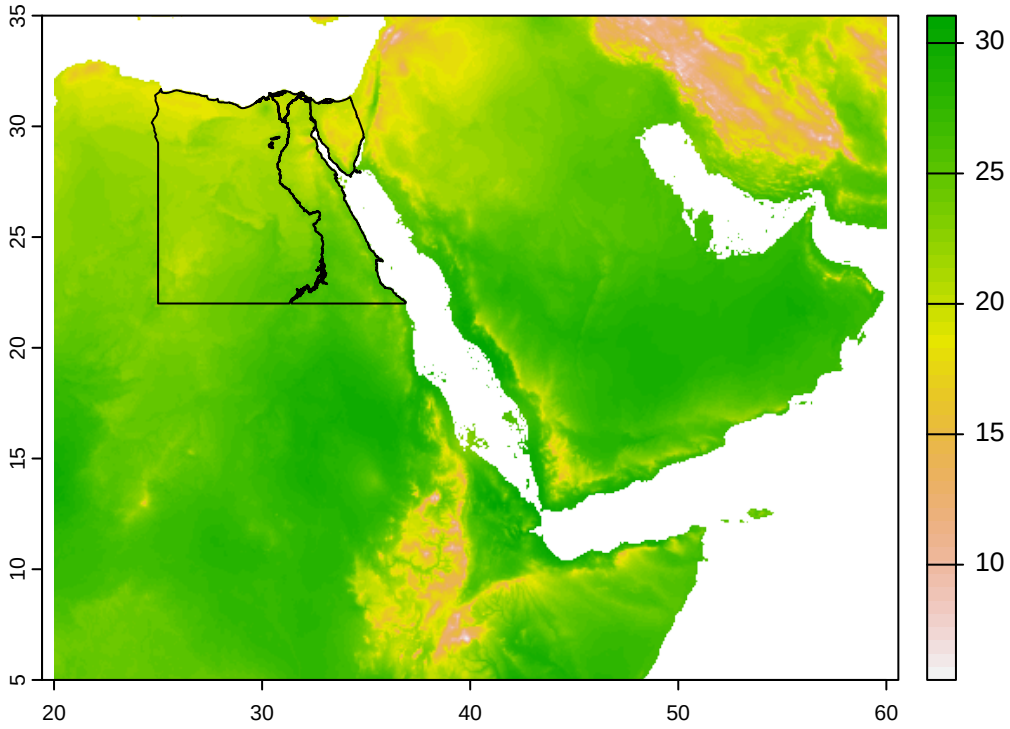
3.2. Modificando el aspecto del raster

```
r_recortado <- crop(r, ext(20, 60, 5, 35))  # xmin, xmax, ymin, ymax

plot(r_recortado)
```

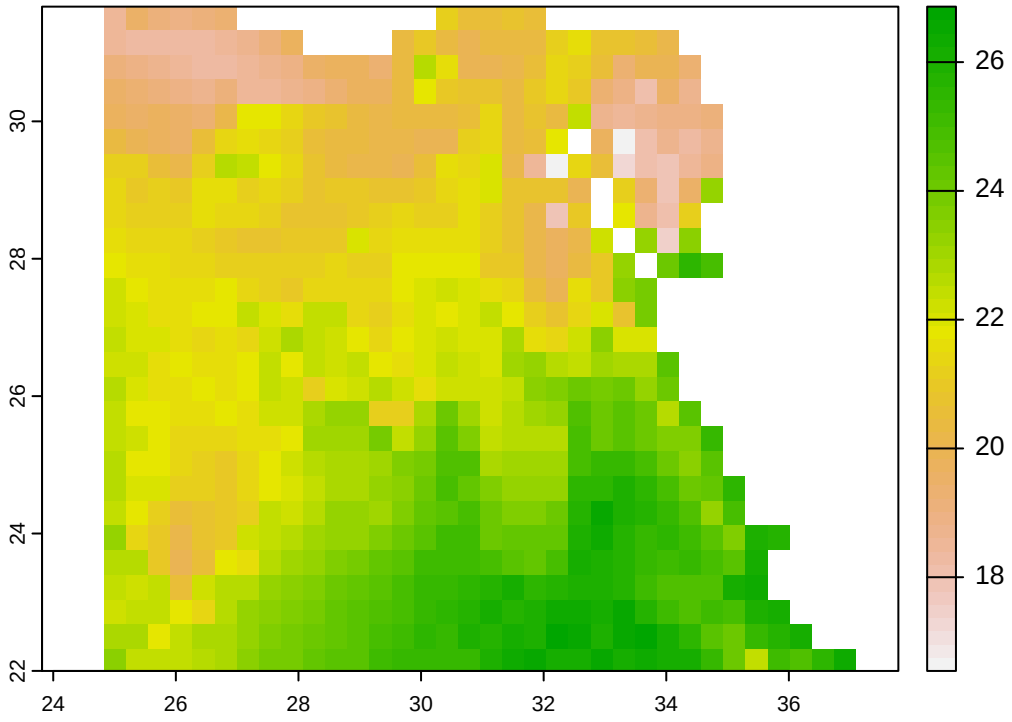


La función `crop( )` recorta toda la extensión del objeto que usamos como recorte. Si queremos recortar el raster en función de un shapefile, podemos hacerlo con la función `mask( )`.

```
# importar con terra datos vectoriales
egipto <- vect("./data/egy_admbnda_adm0_capmas_itos_20170421.shp")

plot(r_recortado)
terra::lines(egipto)
```

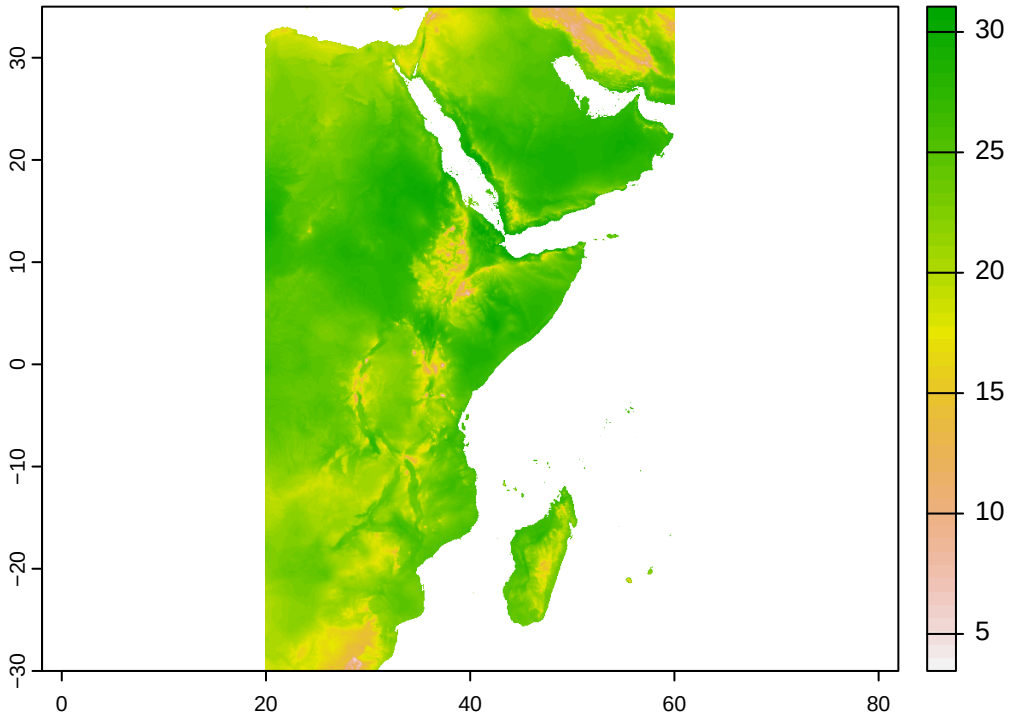



```
egipto_raster <- mask(r, egipto)
plot(egipto_raster, xlim = ext(egipto)[1:2], ylim = ext(egipto)[3:4])
```



Con la función `merge( )` podemos unir dos o más raster en un nuevo objeto.

```
rr1 <- crop(r, ext(20, 60, 5, 35))
rr2 <- crop(r, ext(20, 60, -30, 5))

m <- merge(rr1, rr2, filename = "test.grd", overwrite = TRUE)
plot(m)
```



Si queremos extraer la información de los raster a puntos concretos usamos `extract( )`.

```
library(sf)
```

```
## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1
```

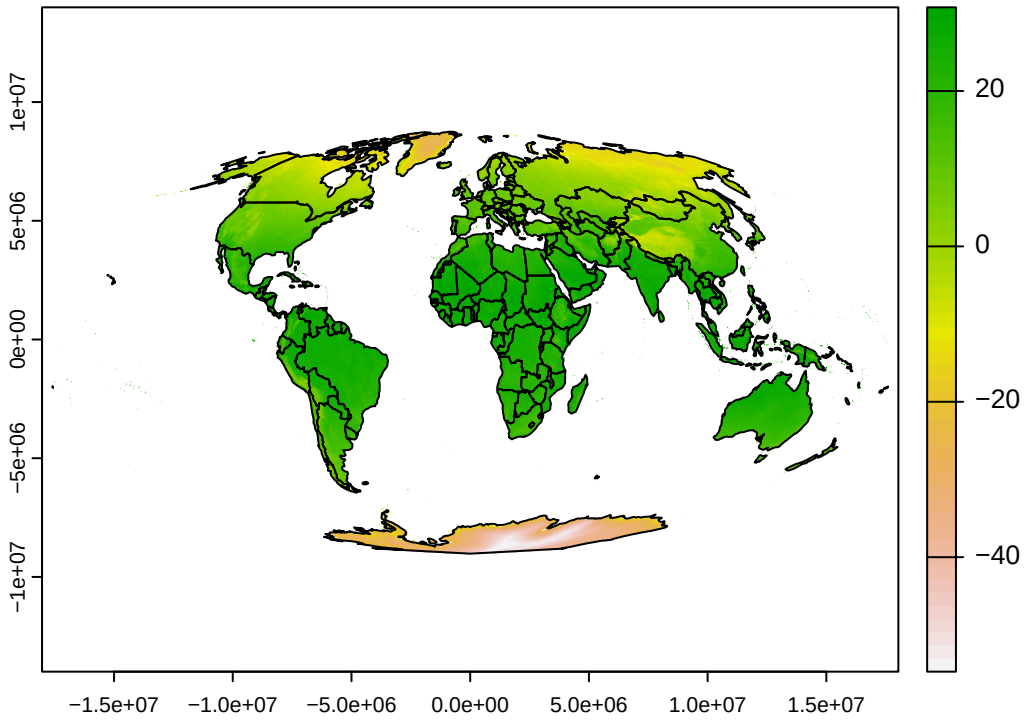
```
# creamos una serie de puntos aleatorios dentro del raster
set.seed(1234)
newpoints <- data.frame(lon = sample(-10:5, 10), lat = sample(30:45,
    10))

# creamos nuestro objeto espacial sf
newpoints <- st_as_sf(newpoints, coords = c("lon", "lat"), crs = 4326)
# extraemos los puntos
newpoints_val <- terra::extract(r, vect(newpoints))
newpoints_val
```

```
##    ID wc2.1_5m_bio_1
## 1   1       21.71921
## 2   2            NaN
## 3   3       16.00979
## 4   4       16.27262
## 5   5            NaN
## 6   6       15.69512
## 7   7            NaN
## 8   8       15.71944
## 9   9       13.73367
## 10 10       17.30329
```

3.3. Reproyección del raster

```
# define la proyección Mollweide https://epsg.io/54009
mollCRS <- "ESRI:54009"

# lee el erchivo
rast_bio <- rast("./data/wc2.1_5m_bio_1.tif")

# reproyecta ambos objetos
temp_moll <- terra::project(rast_bio, mollCRS)

wrld_moll <- st_transform(wld, mollCRS)

# plotea el mapa
plot(temp_moll)
plot(wrld_moll, add = TRUE, col = NA)
```

```
## Warning in plot.sf(wrld_moll, add = TRUE, col = NA): ignoring all but the first
## attribute
```



4. Exportación

La función `writeRaster( )` nos permite exportar nuestro objeto raster a GeoTiff.

```
writeRaster(r, "temperatura_mundial.tif", overwrite = TRUE)
```

5. Paquetes útiles

- Datos LiDAR. *{lidR}*
- Datos MDT Global. *{elevatr]*
- Datos meteorológicos del ECMWFR. *{ecmwfr}*