Chirps project

Vincenzo Belcastro

January 15, 2024

Task

For each month, compute the number of rainy days, averaged over several years of data available, for the area of interest (see data attached), from the CHIRPS dataset (https://www.chc.ucsb.edu/data/chirps).

Expected output

- 12 rasters (one per month), clipped to the same polygon mask as the AOI (area of interest, attached), displaying the average number of rainy days per month, rounded to the nearest integer.
- Documentation of the data acquisition pipeline.

Input

Area of interest (aoi.shp)

Implementation

The idea is to fetch and process Chirps data by month.

Load required packages

```
# list of libraries loaded
library(chirps)
library(terra)
library(lubridate)
library(data.table)
library(ggplot2)
```

Define year range and load shapefile

- define year range (e.g. *c(2019:2021)*, *c(2019,2021)*)
- load shapefile (aoi.shp) to extract the area of interest

```
yy_range <- 2020:2022

# build spacial vector from shape file
sv <- vect("data/aoi.shp")
```

Fetch and process Chirps data

List of steps to fetch and process Chirps data for a month (multiple year):

- fetch Chirps data over years
- compute number of rainy days per year
- average number of rainy days across years

- mask area according to the area of interest

Functions to fetch data

The core function to fetch Chirps data is: `chirps_fetch_for_month()`. The function takes in input a year `yy` and a month `mm` as integers, and the `SpacialVector` object that encapsulate the area of interest. Data is fetcher via a call to `chirps::get_chirps()` specifying date range, server, and desidered resolution (0.25 is also possible).

```
# function to fetch chirps data for a given month and year
chirps_fetch_for_month <- function(yy, mm, sv) {
  # set first and last day of month
  from <- as.Date(paste(yy, mm, 1, sep = "-"))
  to <- get_last_day_of_month(yy, mm)

  # fetch chirps data
  get_chirps(sv, dates = c(from, to), server = "CHC", resolution = 0.05)
}
```

The function `chirps_fetch_month_over_years()` is just a wrapper to loop across years range.

```
# function to fetch chirps data for a specific month across multiple years
# return: a list of SpatRaster objects of lenght equal to the number of years
chirps_fetch_month_over_years <- function(yy_range, mm, sv) {
  .fetch_month <- function(yy, mm, sv) {
    chirps_fetch_for_month(yy, mm = mm, sv = sv)
  }

  lapply(yy_range, .fetch_month, mm = mm, sv = sv)
}
```

The output of this function is a list of `SpatialRasters` inheriting attributes of the `SpatialVector` in input, `sv`, such as *area extend* and *coordinate system*:

```
## class       : SpatRaster
## dimensions  : 14, 30, 31  (nrow, ncol, nlyr)
## resolution  : 0.25, 0.25  (x, y)
## extent      : 104, 111.5, 41.25, 44.75  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source(s)   : memory
## names       : 2020.01.01, 2020.01.02, 2020.01.03, 2020.01.04, 2020.01.05, 2020.01.06, ...
## min values  :        0.0,       0.00,       0.00,        0.0,        0.0,       0.00, ...
## max values  :        0.6,       0.65,       0.08,        1.6,        1.9,       0.18, ...
```

Spatial rasters in output contain, for each cell, the amount of rain per day (in millimiters).

Compute number of rainy days

To compute the number of rainy days is sufficient to sum the number of days with non-zero values

```
# function to compute number of rainy days for a list of spatial rasters
# return:  list of SpatRaster objects containing sums of rainy days
compute_nb_of_rainy_days <- function(daily_mm_of_rain) {
  lapply(daily_mm_of_rain, function(xx) sum(xx > 0))
}
```

Here an example of output:

```
## class       : SpatRaster
## dimensions  : 14, 30, 1  (nrow, ncol, nlyr)
## resolution  : 0.25, 0.25  (x, y)
## extent      : 104, 111.5, 41.25, 44.75  (xmin, xmax, ymin, ymax)
```

```
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source(s)   : memory
## name        : sum
## min value   :   0
## max value   :   7
```

Compute the avg number of rainy days over years

The average number of rainy days over multiple years is computed by

- flattening the list
- compute average values
- round to the closest integer
- apply masking to hide regions outside the area of interest

```
nb_rainy_days <- do.call(c, nb_rainy_days) # flatten list
avg_rainy_days <- round(mean(nb_rainy_days, na.rm = TRUE)) # get rounded avg
mask(avg_rainy_days, sv) # apply mask
```

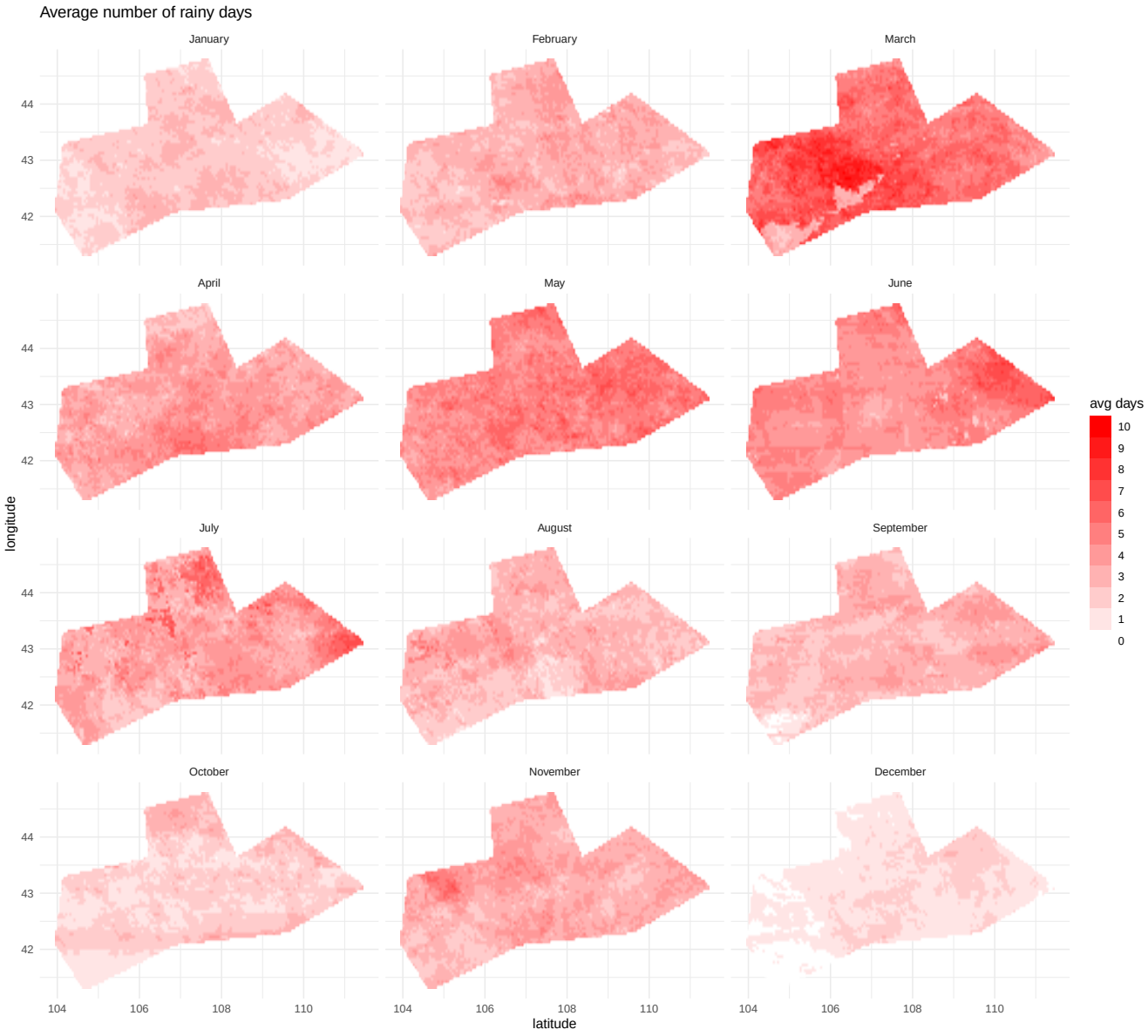
Aggregated output:

```
## class       : SpatRaster
## dimensions  : 70, 150, 12  (nrow, ncol, nlyr)
## resolution  : 0.05, 0.05  (x, y)
## extent      : 103.95, 111.45, 41.3, 44.8  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source(s)   : memory
## names       : January, February, March, April, May, June, ...
## min values  :       1,        0,     2,     1,   3,    2, ...
## max values  :       4,        5,    10,     7,   7,    8, ...
```

Plor rasters

To plot rasters I use the `ggplot2` package.

```
ggplot(data = df, aes(x = x, y = y, fill = factor(value))) +
  geom_raster() +
  # scale_fill_gradient(low="white", high = "red", limits = limits) +
  scale_fill_manual(values = rev(colors), breaks=limits[2]:0) +
  coord_quickmap() +
  theme_minimal () +
  labs(title = "Average number of rainy days", fill = "avg days") +
  xlab("latitude") +
  ylab("longitude") +
  facet_wrap(~variable, ncol=3)
```


Average number of rainy days
January
February
March
April
May
June
July
August
September
October
November
December
longitude
latitude
44
43
42
104
106
108
110
avg days
10
9
8
7
6
5
4
3
2
1
0