Assignment 13

Jeff Oliver

2024-04-21

Assignment Details

Purpose

The goal of this assignment is to practice working with geospatial raster and vector data.

Task

Write R code to successfully answer each question below.

Criteria for Success

- Code is within the provided code chunks or new code chunks are created where necessary
- Code chunks run without errors
- Code chunks have brief comments indicating which code is answering which part of the question
- Code will be assessed as follows:
  - Produces the correct answer using the requested approach: 100%
  - Generally uses the right approach, but a minor mistake results in an incorrect answer: 90%
  - Attempts to solve the problem and makes some progress using the core concept, but returns the wrong answer and does not demonstrate comfort with the core concept: 50%
  - Answer demonstrates a lack of understanding of the core concept: 0%
- Any questions requiring written answers are answered with sufficient detail

Due Date

This Assignment is Optional! April 22 at midnight MST

Assignment Exercises

1. Working with raster data in terra (20 pts)

In this exercise, you will work with the same precipitation data as in class, but this time focus in on the Himalayan region of Asia.

a. Load in the terra package. What version does it say you are using?

```r
install.packages("terra", repos = "https://cran.r-project.org/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/n4/w8q6d2092t75r2qr437ntcrm0000gn/T//RtmpAxFcQh/downloaded_packages
```

```r
library(terra)
```

```
## terra 1.7.71
```

- terra 1.7.21

b. Load in the precipitation data ("global_precipitation.tif"), what is the EPSG used in these data?

```r
prec <- rast("image_files/global_precipitation.tif")
prec
```

```
## class       : SpatRaster
## dimensions  : 4320, 8640, 1  (nrow, ncol, nlyr)
## resolution  : 0.04166667, 0.04166667  (x, y)
## extent      : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source      : global_precipitation.tif
## name        : global_precipitation
## min value   :                    0
## max value   :                11246
```

- EPSG: 4326

c. Define a geographic extent object for the Himalayan region.

```r
himalayan_ext <- ext(c(69, 97, 27, 40))
```

d. Crop the precipitation data to the Himalayan extent; what are resulting dimensions?

```r
prec_him <- crop(prec, himalayan_ext)
```

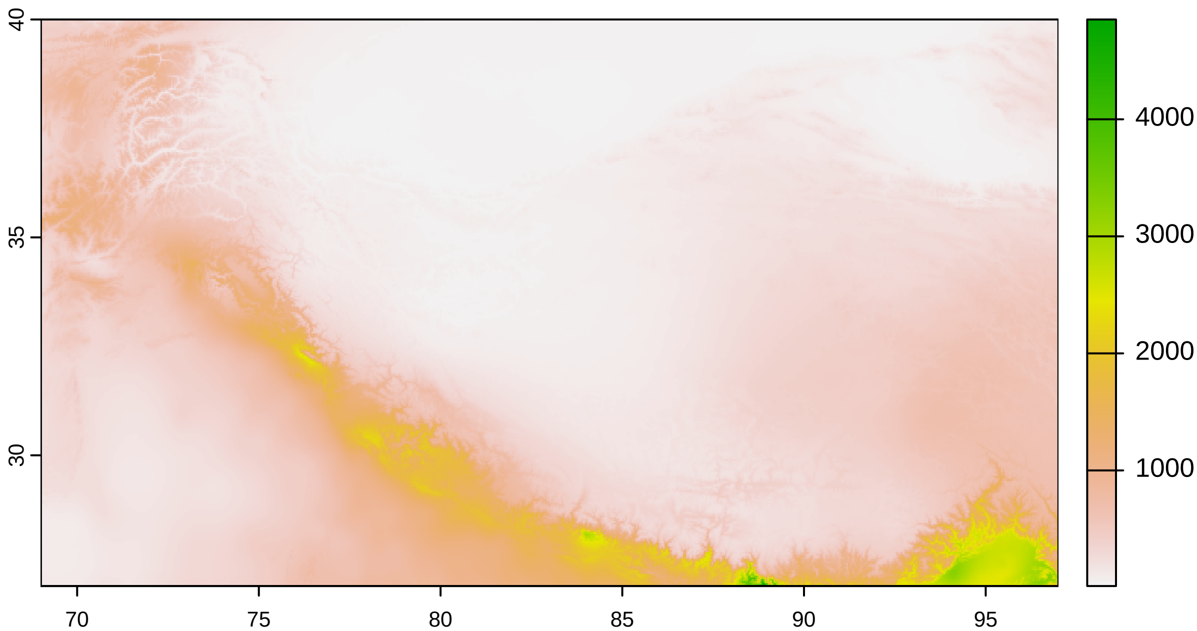
- 312 rows, 672 columns, 1 layer.

2. Printing maps (20 points)

Still working with the precipitation data, this exercise focuses on visualizing the data.

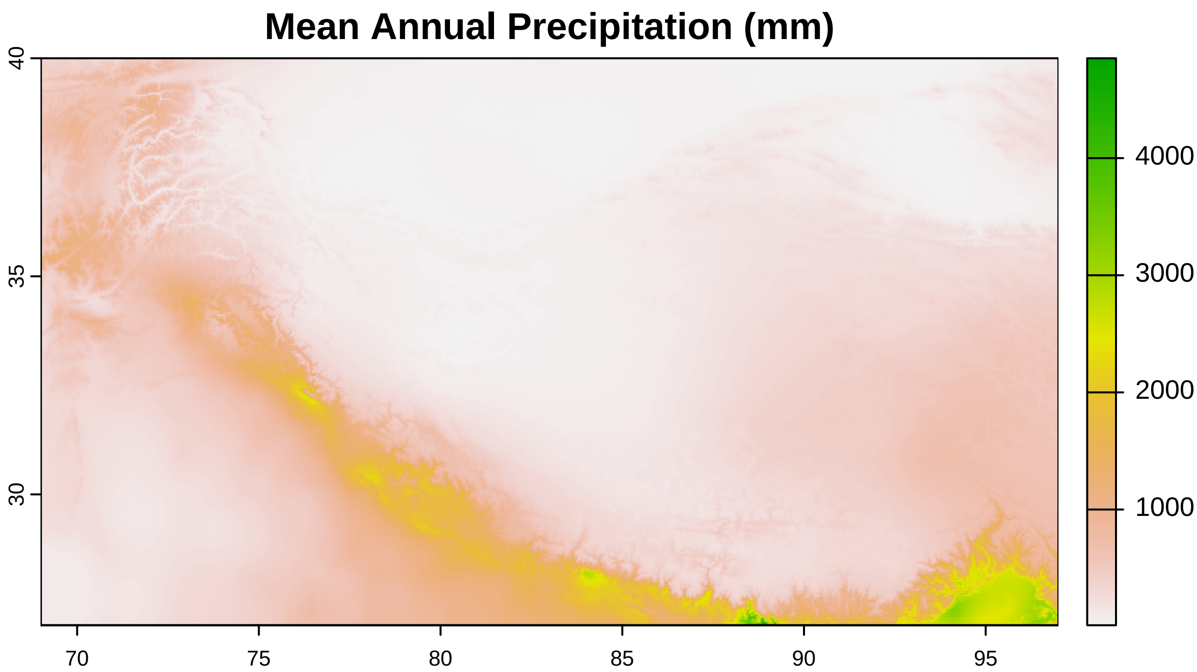
a. Plot the precipitation data for the Himalayan region.

```r
plot(prec_him)
```



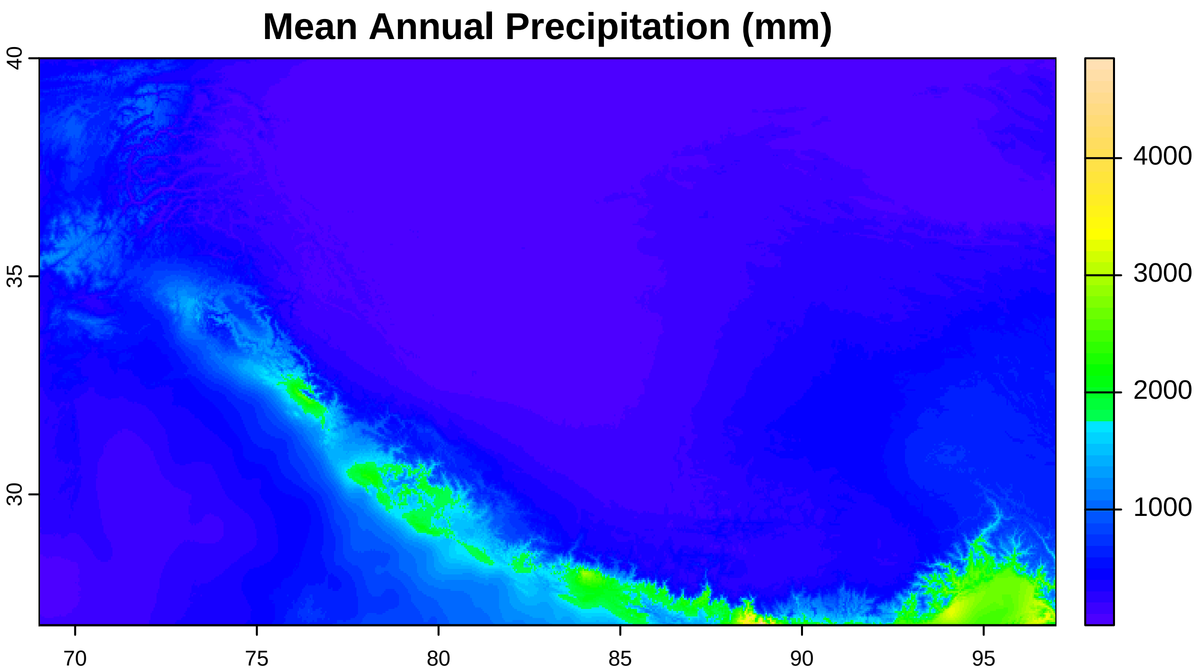
b. Add a title to the plot.

```r
plot(prec_him, main = "Mean Annual Precipitation (mm)")
```



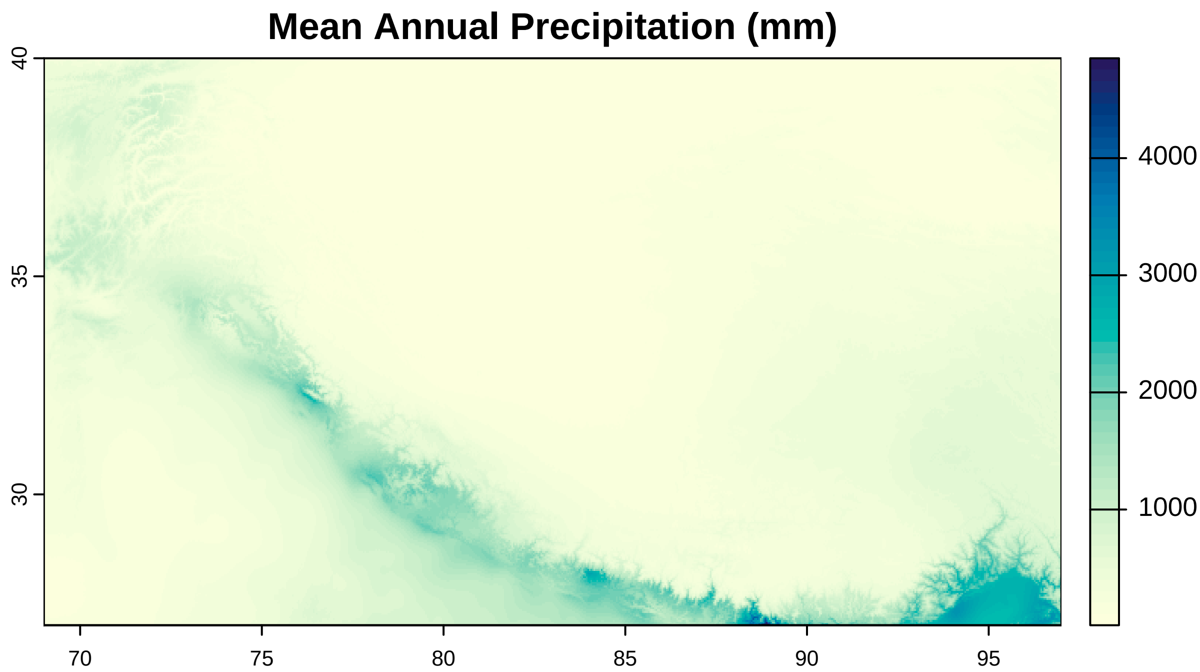
c. Change the palette to that of `topo.colors` (use `?topo.colors` for details) and print the resultant plot.

```
plot(prec_him, main = "Mean Annual Precipitation (mm)", col =topo.colors(n=50))
```



d. Change the palette to one from ColorBrewer for sequential (continuous) data.

```
plot(prec_him, main = "Mean Annual Precipitation (mm)", col = rev(hcl.colors(n=50, palette= "YlGnBu")))
```



3. Working with vector data in terra (20 points)

a. Create a data frame of three peaks in the Himalayan region: (lat, lon)

- K2: 35.93, 76.50
- Annapurna: 28.53, 83.81
- Mount Everest: 28.05, 86.91

```
peaks <- data.frame(name = c("K2", "Annapuna", "Mount Everest"),
                    lat = c(35.93, 28.53, 28.05),
                    lon = c(76.50, 83.81, 86.91))
peaks
```

```
##            name   lat   lon
## 1            K2 35.93 76.50
## 2      Annapuna 28.53 83.81
## 3 Mount Everest 28.05 86.91
```

b. Create a SpatVector object from the data frame with peak data.

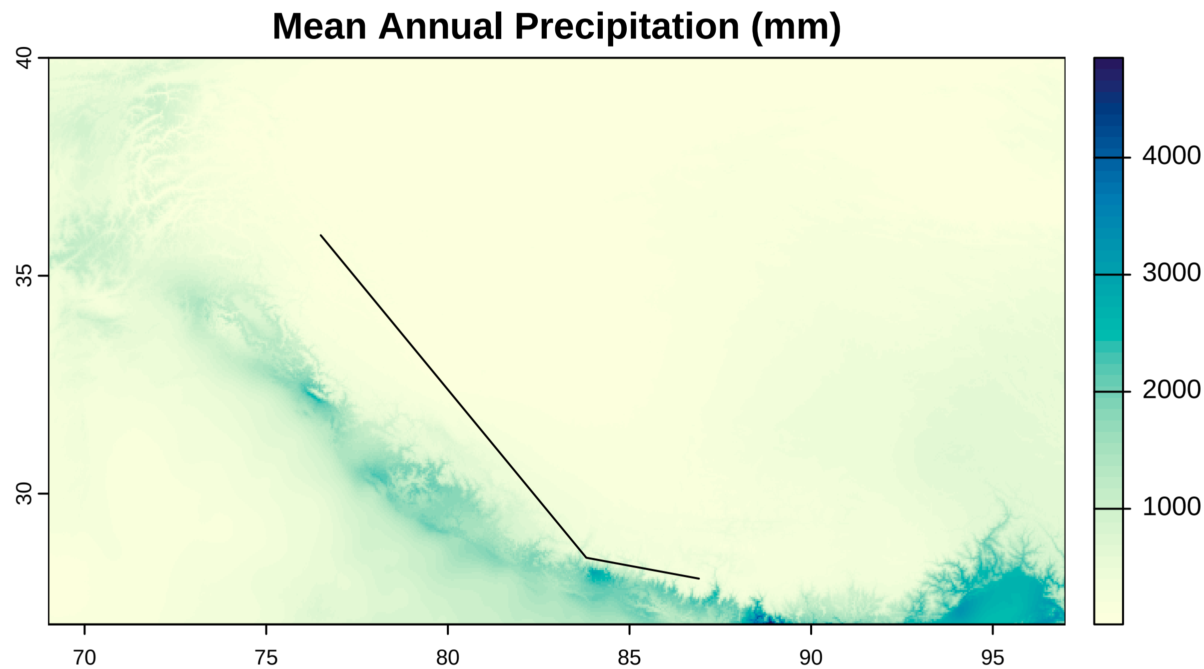
```
peaks_vect <- vect(peaks, crs = crs(prec_him))
```

c. Convert the points SpatVector object to a lines object.

```r
line <- as.lines(peaks_vect)
```

d. Add the lines object to the precipitation plot of the Himalayan region.

```r
plot(prec_him, main = "Mean Annual Precipitation (mm)", col = rev(hcl.colors(n=50, palette= "YlGnBu")))
plot(line, add = TRUE)
```



4. Modifying raster values (20 points)

a. Load in the elevation data ("global_elevation.tif"), what are the units for these data?

```r
elev <- rast("image_files/global_elevation.tif")
elev
```

```
## class       : SpatRaster
## dimensions  : 4320, 8640, 1  (nrow, ncol, nlyr)
## resolution  : 0.04166667, 0.04166667  (x, y)
## extent      : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source      : global_elevation.tif
## name        : global_elevation
## min value   :             -415
## max value   :             7412
```

- Meters

b. Convert the units of the raster to feet.

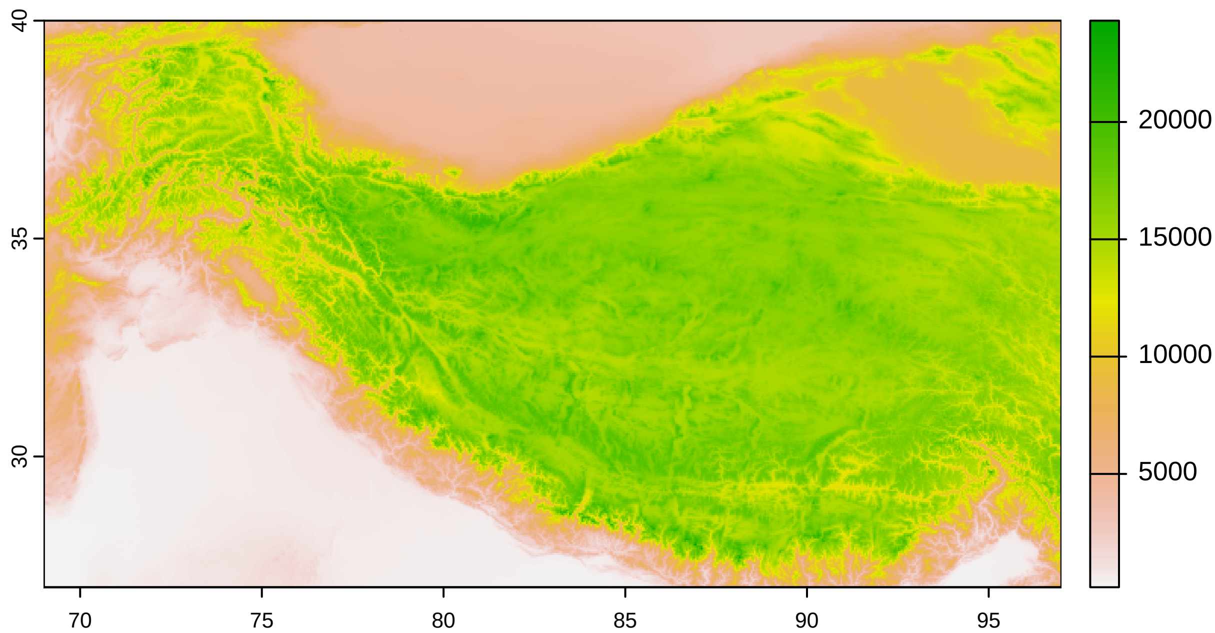
```
elev_f <- elev * 3.281
```

c. Crop the elevation data to the same geographic extent as precipitation data.

```
elev_him <- crop(elev_f, himalayan_ext)
```

d. Plot the Himalayan elevation data.

```
plot(elev_him)
```

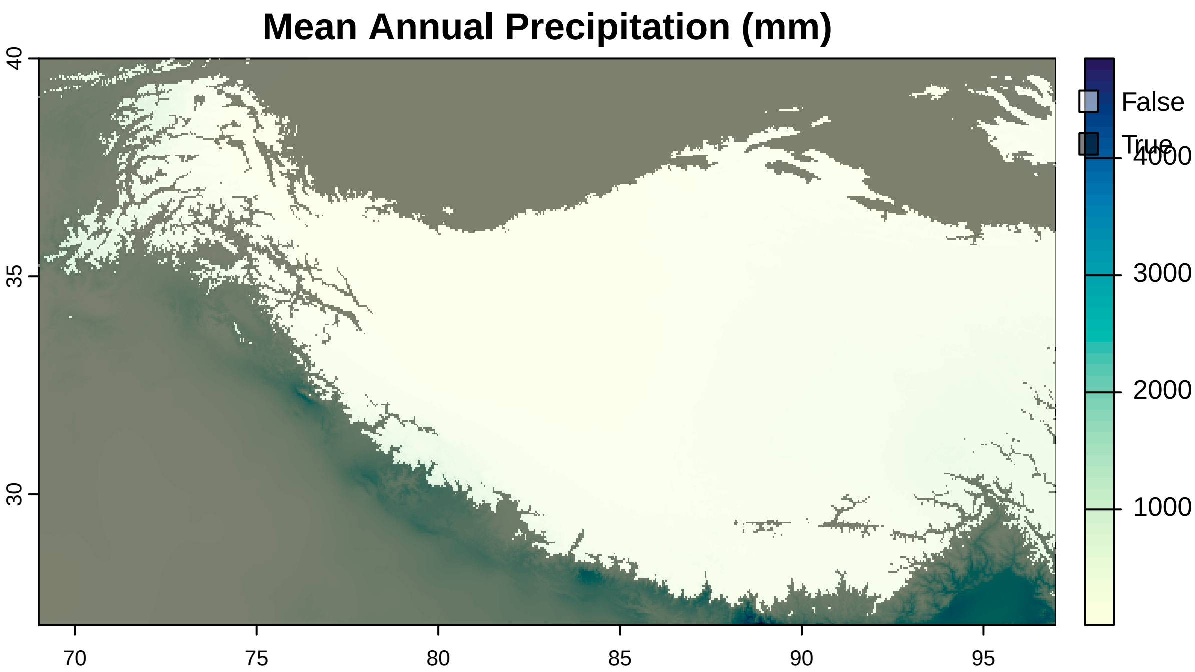


5. Converting raster objects to spatial vector objects (20 points)

For this last exercise, create a plot that shows precipitation for the Himalayan region, using a color palette from ColorBrewer AND add a layer that is partially (50%) transparent to shade out regions below 13,000 feet.

```
elev_max <- 13000
him_max <- elev_him < elev_max

plot(prec_him, main = "Mean Annual Precipitation (mm)", col = rev(hcl.colors(n=50, palette= "YlGnBu")))
plot(him_max, col = c("transparent", "black"),alpha=0.5, add=TRUE)
```


Mean Annual Precipitation (mm)
False
True
4000
3000
2000
1000
40
35
30
70
75
80
85
90
95