Getting Climate Data for the Morphology CURE

Alex Krohn

31 July 2020

Introduction

This tutorial is meant to help instructors, or advanced students, extract climate data from the specimen locality data for Research Goal 2 of the Morphology CURE. It assumes that you've already downloaded the specimen data using R, and thus have some familiarity using R.

To use this tutorial, you need a set of longitudes and latitudes. This could be from a dataset that you create (maybe from the test dataset) or from lat/longs that students send back to you with coreIDs to identify individuals. For purposes of repeatability, I will generate these climate data from a subset of 100 individuals from the test dataset.

This tutorial closely follows Shannon Carter's tutorial here.

```r
# Load the required libraries
library(tidyverse)
library(prism)
library(raster)
library(lubridate)
```

Load Your Specimen Data

```r
# Set your working directory to the CURE folder
setwd("CURE/")

# Read in the specimen data from the final datasheet from the test data
occ.data <- read_csv("final-metadata-spreadsheet.csv")

set.seed(25)

# Remove any specimens without the needed data
occ.subset <-
  occ.data %>%
  filter(!is.na(lat) & !is.na(lon) & !is.na(coll.date))

# Some dates have 00 as the day, which complicates things. Let's just split the date, and extract the y
occ.years <- str_split_fixed(occ.subset$coll.date, "-", 3)

# Finally, add in a year column, keeping only the core ID, lat, long and year column in the final dataf
occ.final <-
  occ.subset %>%
  mutate(year = occ.years[,1]) %>%
  dplyr::select(coreid, year, lat, lon) %>%
  filter(year > 1891) %>% # Only keep years after 1891, the first year of PRISM data
  slice_sample(n = 100)
```

Download Climate Data

We'll download a raster of global climate data for each year in our dataset. For our random subset, that's 57 different years (yours may different). Eventually, from the specimens' corresponding year of climate data, we will extract the average annual conditions (temperature in our case) at the specimens' location. You can choose to simplify this process by, instead of downloading data for each year that a specimen was collected, only downloading, say, 2010-2020, then taking the average of those measurements and extracting the value at the specimens' location. Or, you could get very specific by downloading global climate data for the exact day the specimen was collected (using `get_prims_dailys()`), and then extracting data at the specimens' locations. Those two approaches are beyond the scope of this tutorial. We'll take the road of middle complication.

```
# Set the file path to where the climate data will be stored.
options(prism.path = "CURE/")

# Create a directory for each year, then download the data. This will take a few minutes depending on y
for (y in unique(occ.final$year)) {
    dir.create(y, FALSE)
    options(prism.path = y)
    get_prism_annual(type = "tmean", years = as.numeric(y))
} # We're downloading mean temperature data here. See ?get_prism_annual() for different choices
```

Extract the Annual Temperature Values for Each Specimen

```
# Turn the lat lat longs into cordinates so they can accept a coordinate system
occ.latlong <-
  dplyr::select(occ.final, lon, lat)
coordinates(occ.latlong) <- c('lon', 'lat')

# Find the right raster, stack it, then extract the temperature data at each location
temps <- list()
for(i in 1:length(occ.final$year)) {
    options(prism.path = occ.final$year[i]) # Find it
    climate_data <- prism_stack(ls_prism_data()) # Stack it
    climate_crs <- climate_data@crs@projargs # Find the coordinate system
    proj4string(occ.latlong) <- CRS(climate_crs) # Put the lat longs in that coordinate system
    temps[[i]] <- extract(climate_data, occ.latlong[i]) # Extract the values
}

# Add the temp data to a final dataframe
temps.lats.longs <-
  occ.final %>%
  mutate(avg.temp = unlist(temps))

# Write the dataframe to csv
write_csv(temps.lats.longs, "/your/path/here/tempdata.csv")
```

Note: a few of the temperature values did not extract properly. They seem to be in plausible locations (Ireland, Canada etc.), so I'm not sure why they didn't work.