

GRIDMET Raster Extraction

Alvin Sheng

9/28/2021

```
library(here)

## Warning in readLines(f, n): line 1 appears to contain an embedded nul
## Warning in readLines(f, n): incomplete final line found on '/Volumes/
## ALVINDRIVE2/flood-risk-health-effects/._flood-risk-health-effects.Rproj'
## here() starts at /Volumes/ALVINDRIVE2/flood-risk-health-effects

library(raster)

## Loading required package: sp

library(exactextractr)
library(ggplot2)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.0
## v readr   2.1.1      v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ----- tidyverse_conflicts() --
## x tidyr::extract() masks raster::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x dplyr::select()  masks raster::select()

library(sf)

## Linking to GEOS 3.8.1, GDAL 3.2.1, PROJ 7.2.1

library(stringr)

i_am("GRIDMET/gridmet_raster_extraction.Rmd")

## here() starts at /Volumes/ALVINDRIVE2/flood-risk-health-effects
```

Reading in the rasters

```
tmmx_files <- list.files(here("GRIDMET/all_year_tmmx"))
rmax_files <- list.files(here("GRIDMET/all_year_rmax"))
```

```

num_rast <- length(tmmx_files)

# This raster array has 2 dimensions:
# first dimension is for the 2 variables, second dimension is for the years

raster_array <- list()

# TBC: put a list of rasters across the years instead of just one raster

raster_array[[1]] <- vector("list", length = num_rast)
raster_array[[2]] <- vector("list", length = num_rast)

for (i in 1:num_rast) {

  raster_array[[1]][[i]] <- suppressWarnings(raster(here("GRIDMET/all_year_tmmx", tmmx_files[i])))

  raster_array[[2]][[i]] <- suppressWarnings(raster(here("GRIDMET/all_year_rmax", rmax_files[i])))

}

```

Stacking all four types of rasters

```
tmmx <- stack(raster_array[[1]])
```

```
rmax <- stack(raster_array[[2]])
```

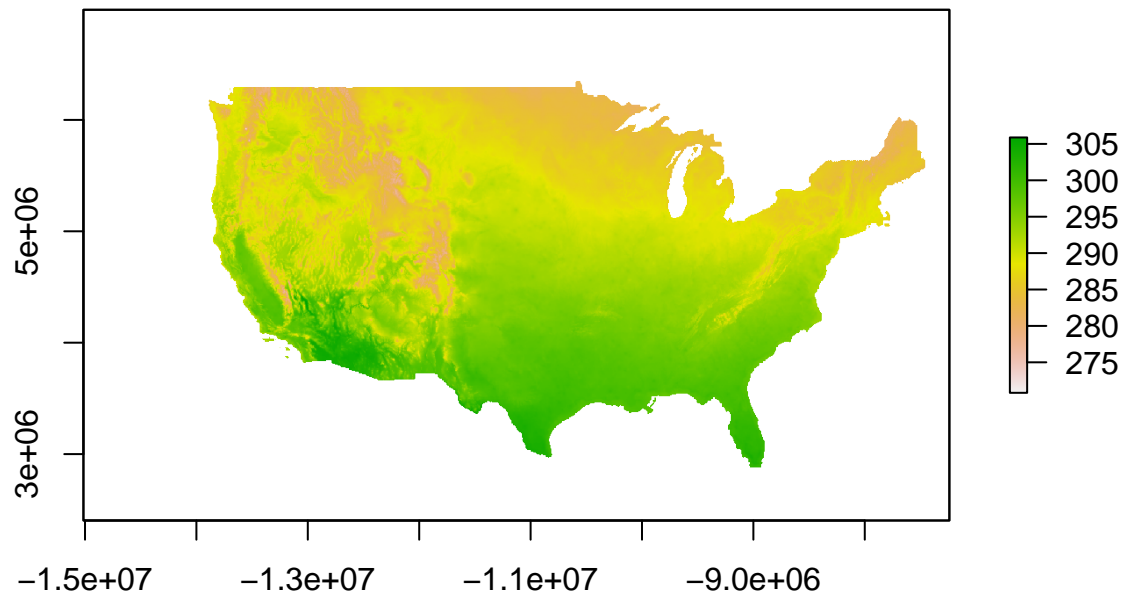
```
tmmx_mean <- mean(tmmx)
```

```
rmax_mean <- mean(rmax)
```

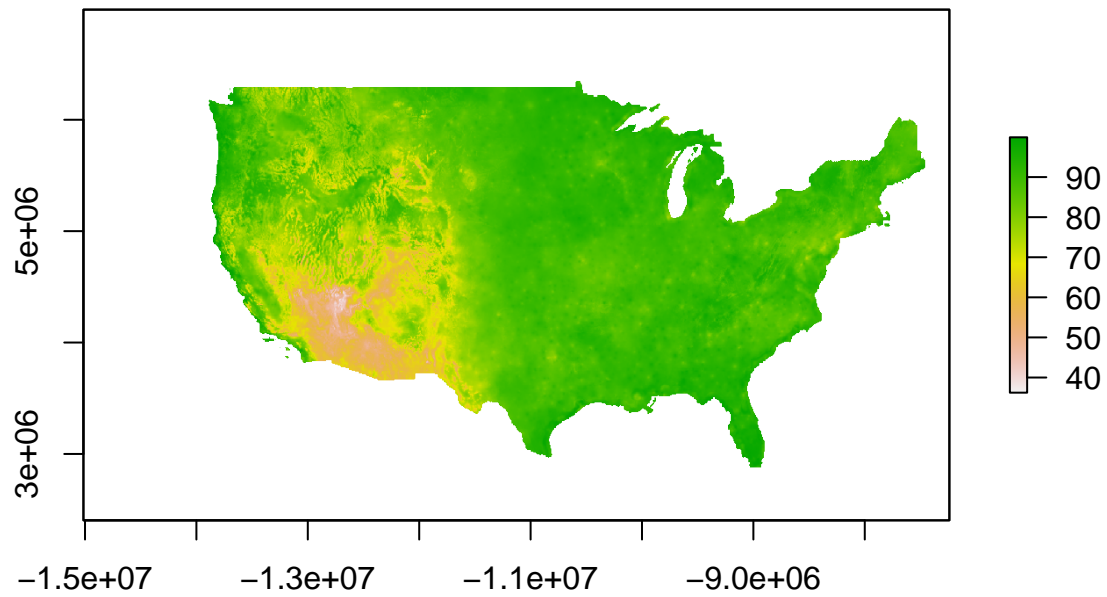
```
mean_array_list <- list(tmmx_mean, rmax_mean)
```

Plotting the rasters

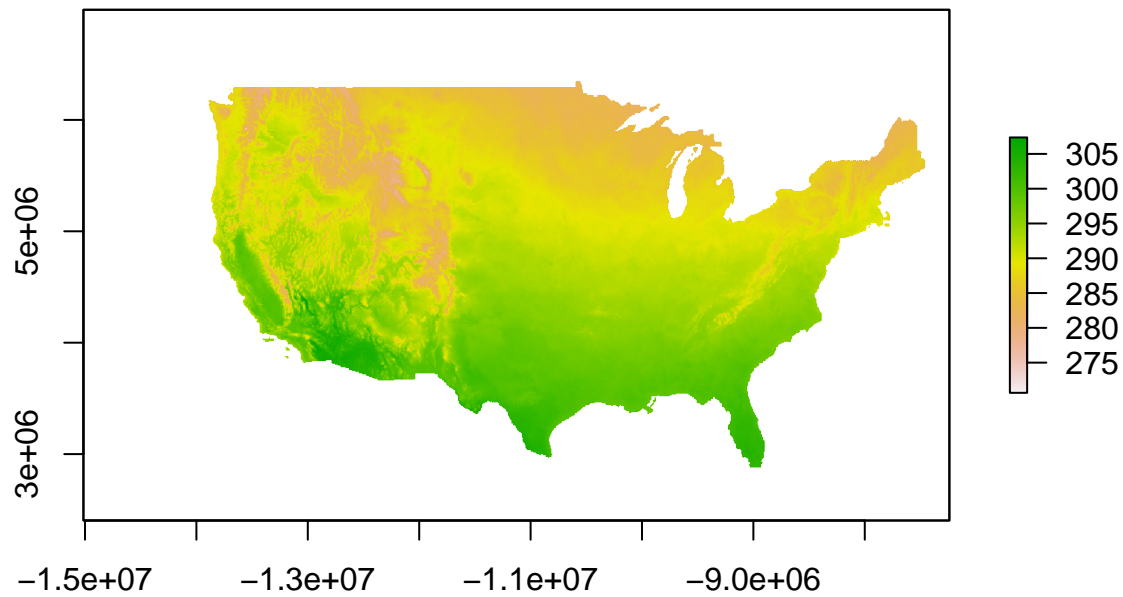
```
plot(raster_array[[1]][[1]]) # tmmx 2005
```



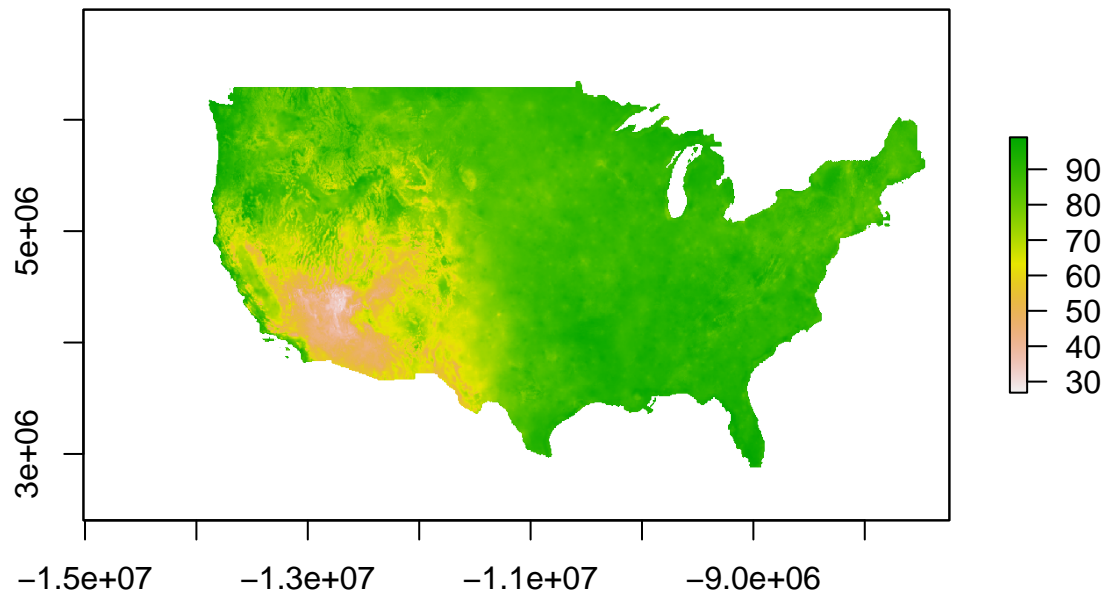
```
plot(raster_array[[2]][[1]]) # rmax 2005
```



```
plot(raster_array[[1]][[num_rast]]) # tmmx 2020
```



```
plot(raster_array[[2]][[num_rast]]) # rmax 2020
```



Extracting mean raster values

Reading in all the state shapefiles

```
ct_files <- list.files(here("imported_data/census_tract_shapefiles/"))
```

```
shp_list <- vector("list", length = length(ct_files))
```

```
for (i in 1:length(ct_files)) {
```

```
  shp_list[[i]] <- st_read(dsn = here("imported_data/census_tract_shapefiles", ct_files[i], paste0(ct_f
```

```
}
```

```
# making names consistent between 2019 versions and 2010 versions
```

```
names(shp_list[[which(ct_files == "t1_2019_46_tract")]]) <- names(shp_list[[which(ct_files == "t1_2010_46_tract")]])  
names(shp_list[[which(ct_files == "t1_2019_51_tract")]]) <- names(shp_list[[which(ct_files == "t1_2010_51_tract")]])
```

```
shp_df <- do.call("rbind", shp_list)
```

```
num_ct <- nrow(shp_df)
```

```
mean_df <- data.frame(fips = shp_df$GEOID10,  
                     tmmx = rep(NA, num_ct), rmax = rep(NA, num_ct))
```

```
for (i in 1:nrow(shp_df)) { # looping over the fips
```

```
  for (j in 1:2) { # looping over the variables
```

```
    suppressWarnings(mean_df[i, j + 1] <- exact_extract(x = mean_array_list[[j]],  
                                                         y = shp_df[i, ], fun = "mean"))
```

```
  }
```

```
}
```

The above loop takes a long time.

```
saveRDS(mean_df, file = here("intermediary_data/mean_df_GRIDMET.rds"))
```