

GRIDMET Raster Extraction

Alvin Sheng

9/28/2021

```
library(here)

## here() starts at /Users/Alvin/Documents/NCSU_Fall_2021/NIH_SIP/FirstStreet
library(raster)

## Loading required package: sp
library(exactextractr)
library(ggplot2)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v tibble 3.0.5      v dplyr 1.0.3
## v tidyr 1.1.2      v stringr 1.4.0
## v readr 1.4.0      v forcats 0.5.0
## 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 3.1.4, PROJ 6.3.1
library(stringr)

i_am("GRIDMET/gridmet_raster_extraction.Rmd")

## here() starts at /Users/Alvin/Documents/NCSU_Fall_2021/NIH_SIP/FirstStreet
```

Reading in the rasters

```
tmmx_files <- list.files(here("GRIDMET/tmmx"))

summer_tmmx_files <- tmmx_files[str_detect(tmmx_files, "summer")]

winter_tmmx_files <- tmmx_files[str_detect(tmmx_files, "winter")]

rmax_files <- list.files(here("GRIDMET/rmax"))
```

```

summer_rmax_files <- rmax_files[str_detect(rmax_files, "summer")]
winter_rmax_files <- rmax_files[str_detect(rmax_files, "winter")]

num_rast <- length(summer_tmmx_files)

# This raster array has 2 dimensions:
# first dimension is for the 4 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)
raster_array[[3]] <- vector("list", length = num_rast)
raster_array[[4]] <- vector("list", length = num_rast)

for (i in 1:num_rast) {

  raster_array[[1]][[i]] <- suppressWarnings(raster(here("GRIDMET/tmmx/"), summer_tmmx_files[i]))
  raster_array[[2]][[i]] <- suppressWarnings(raster(here("GRIDMET/tmmx/"), winter_tmmx_files[i]))
  raster_array[[3]][[i]] <- suppressWarnings(raster(here("GRIDMET/rmax/"), summer_rmax_files[i]))
  raster_array[[4]][[i]] <- suppressWarnings(raster(here("GRIDMET/rmax/"), winter_rmax_files[i]))

}

```

Stacking all four types of rasters

```

summer_tmmx <- stack(raster_array[[1]])
winter_tmmx <- stack(raster_array[[2]])
summer_rmax <- stack(raster_array[[3]])
winter_rmax <- stack(raster_array[[4]])

summer_tmmx_mean <- mean(summer_tmmx)
winter_tmmx_mean <- mean(winter_tmmx)
summer_rmax_mean <- mean(summer_rmax)
winter_rmax_mean <- mean(winter_rmax)

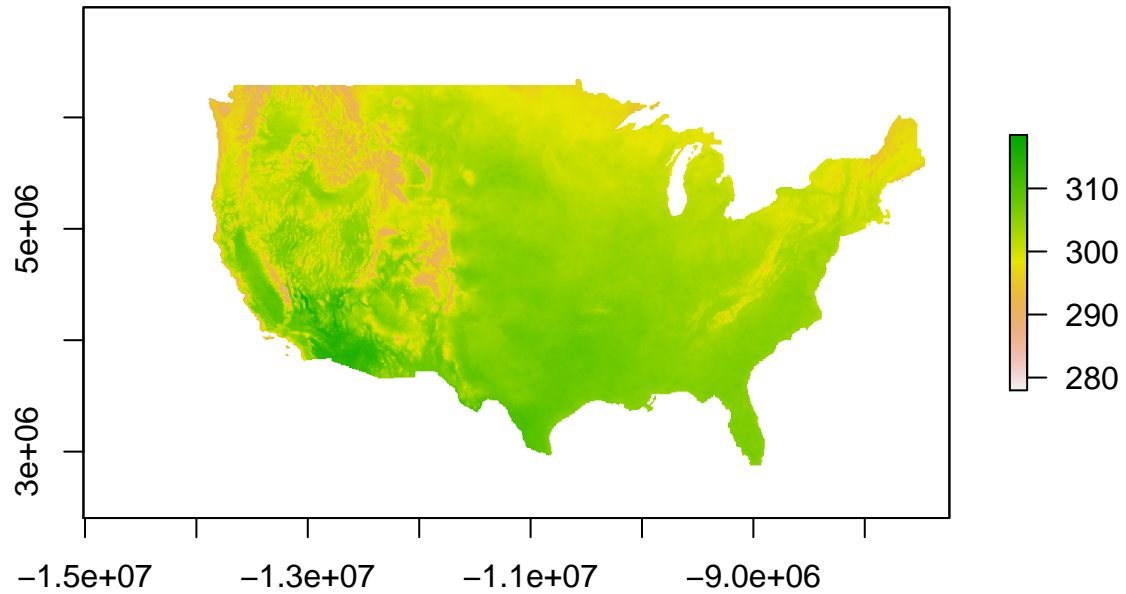
mean_array_list <- list(summer_tmmx_mean, winter_tmmx_mean, summer_rmax_mean, winter_rmax_mean)

```

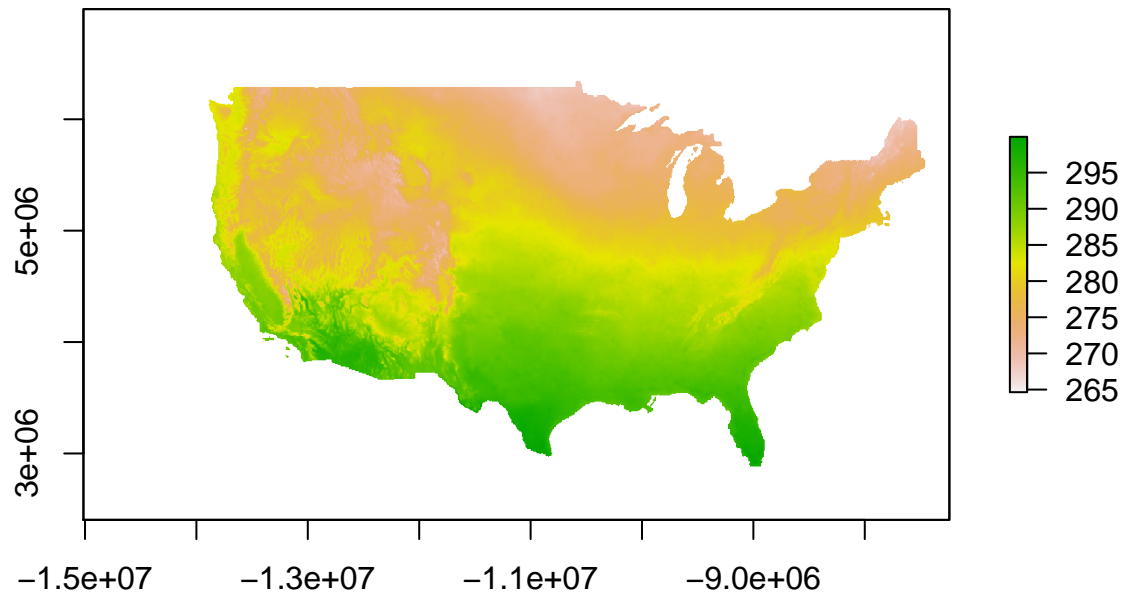
Plotting the rasters

Maximum Temperature

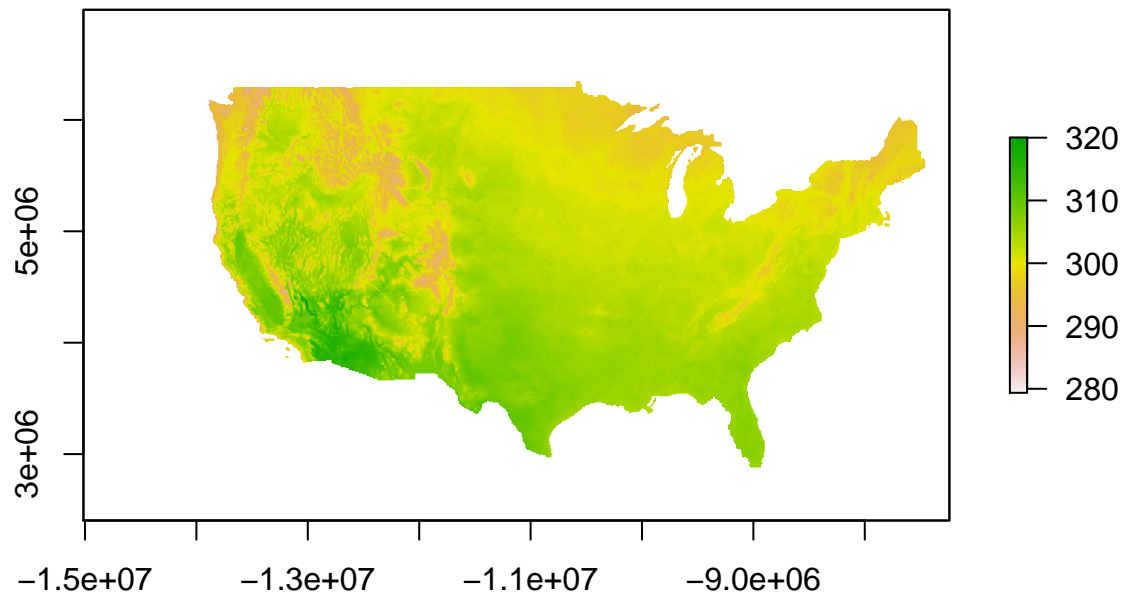
```
plot(raster_array[[1]][[1]]) # summer 2005
```



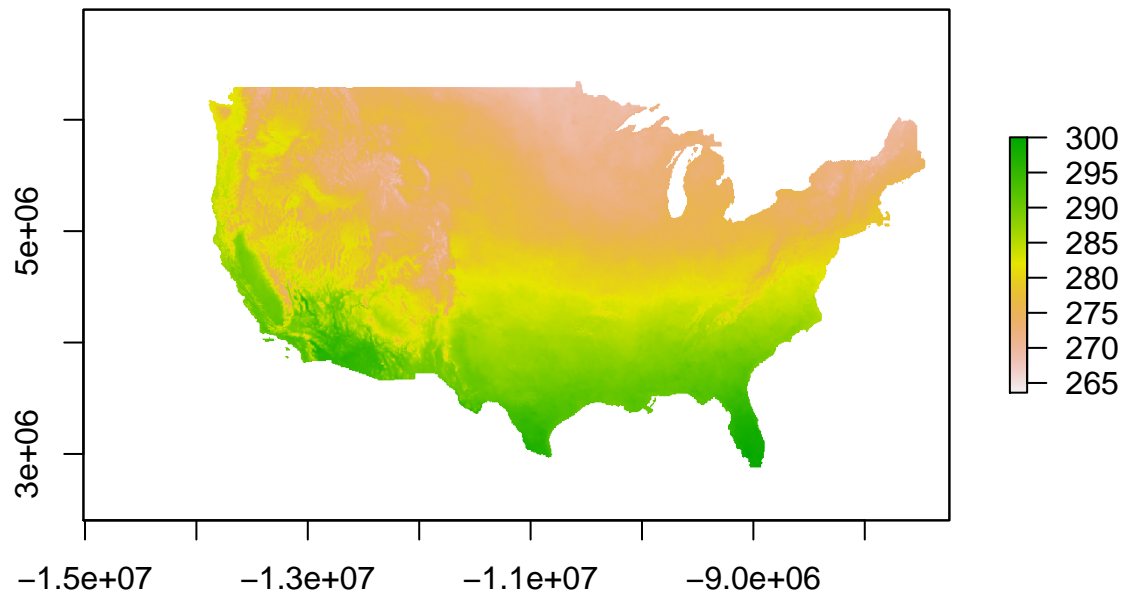
```
plot(raster_array[[2]][[1]]) # winter 2005
```



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

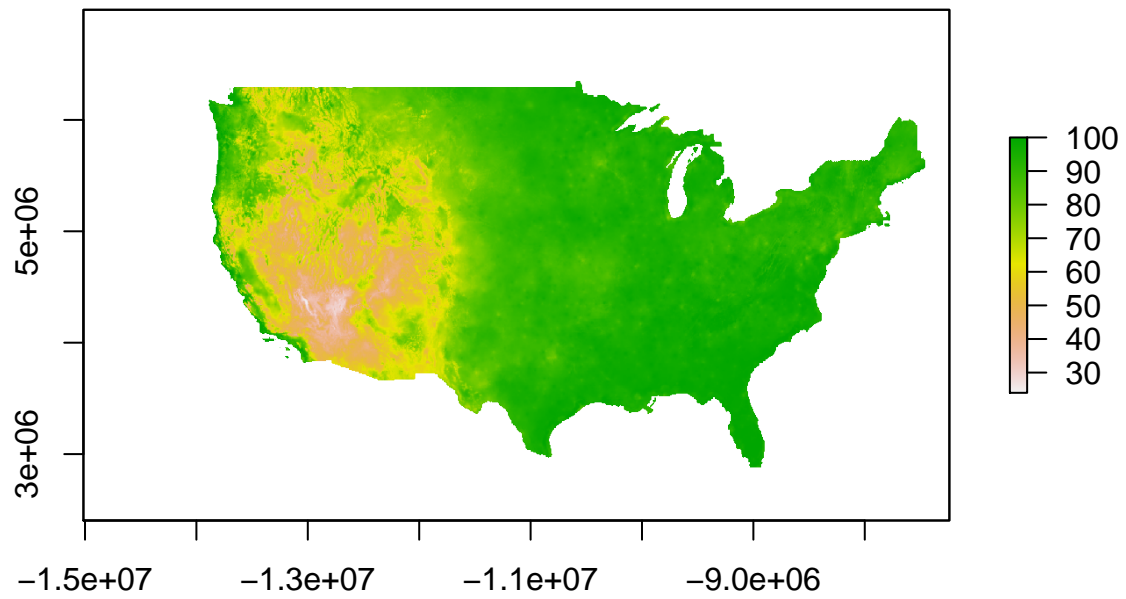


```
plot(raster_array[[2]][[num_rast]]) # winter 2020
```

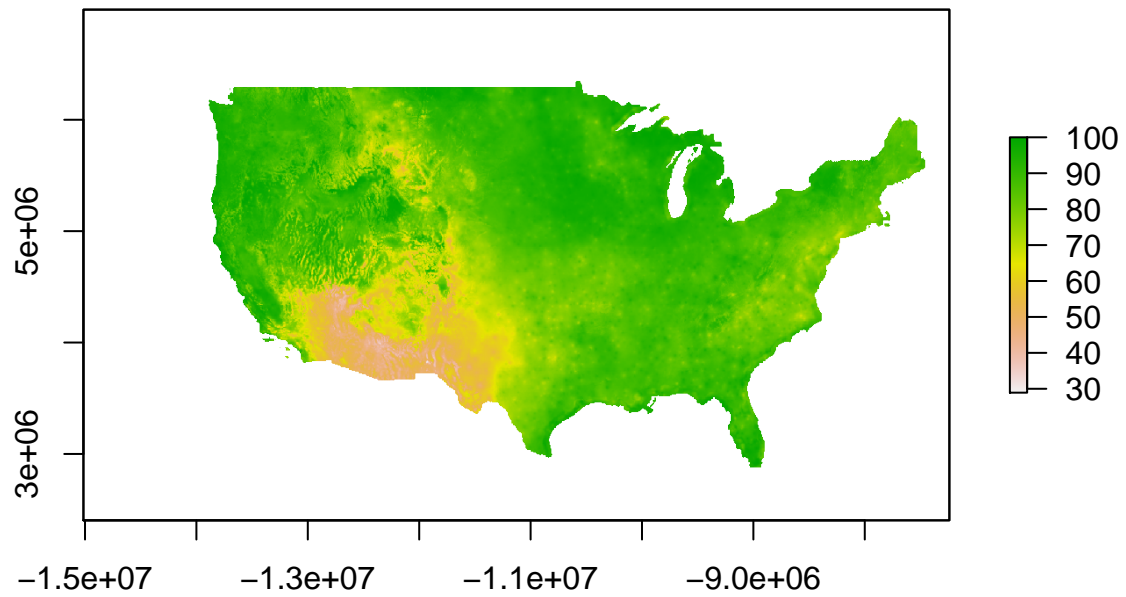


Maximum Relative Humidity

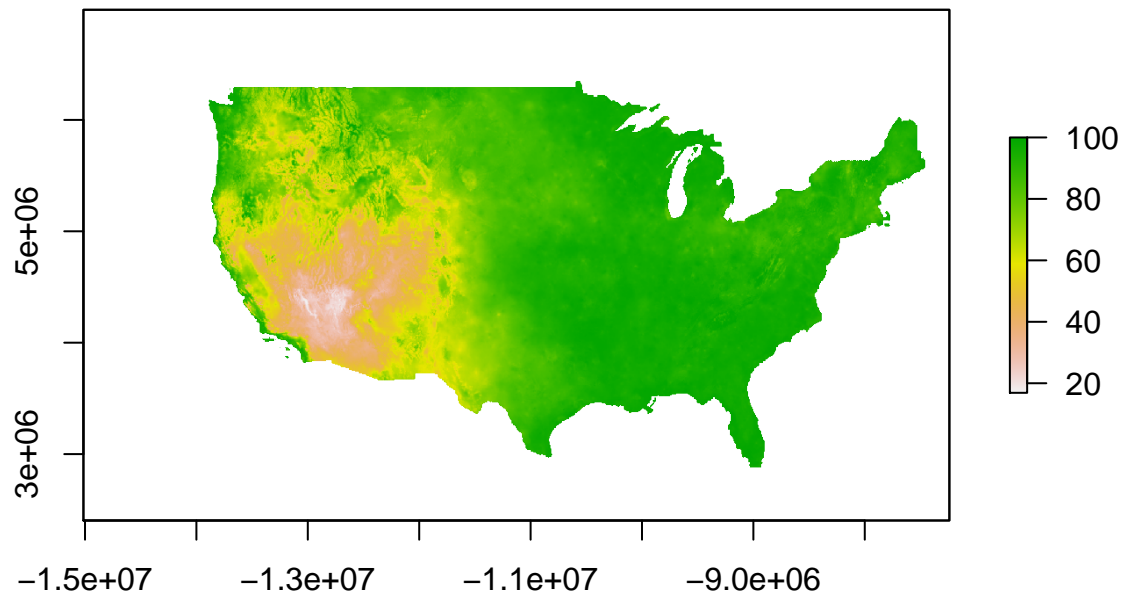
```
plot(raster_array[[3]][[1]]) # summer 2005
```



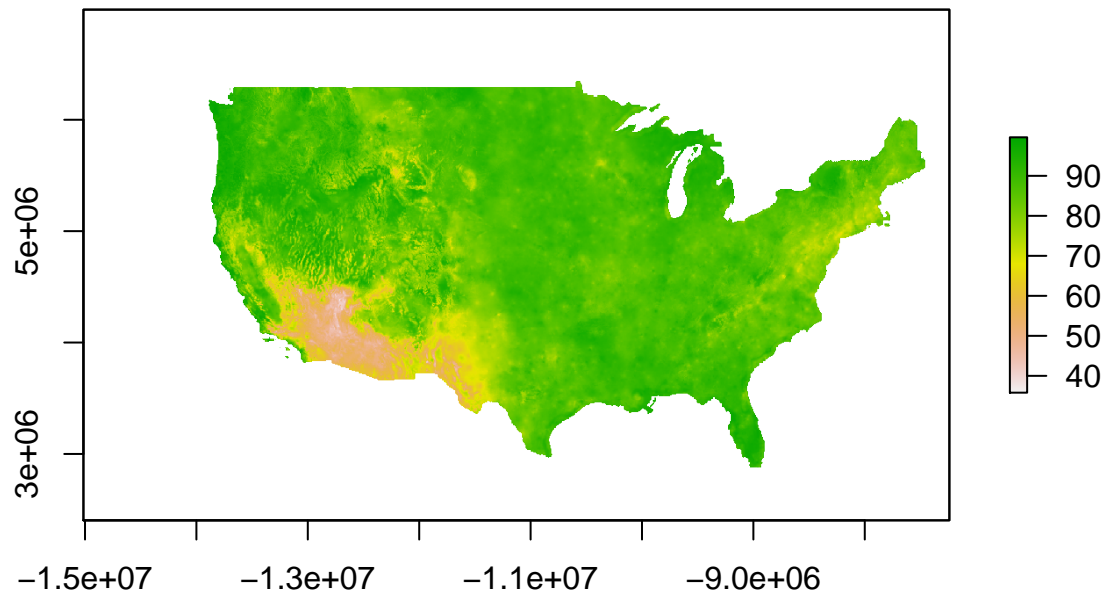
```
plot(raster_array[[4]][[1]]) # winter 2005
```



```
plot(raster_array[[3]][[num_rast]]) # summer 2020
```



```
plot(raster_array[[4]][[num_rast]]) # winter 2020
```



Extracting mean raster values

Reading in all the state shapefiles

```
ct_files <- list.files(here("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("census_tract_shapefiles", ct_files[i], paste0(ct_files[i], ".shp
```

```
})
```

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

Focus on North Carolina first, before reading in the other TIGER/LINE shapefiles

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

```
mean_df <- data.frame(fips = shp_df$GEOID10,  
                      summer_tmmx = rep(NA, num_ct), winter_tmmx = rep(NA, num_ct),  
                      summer_rmax = rep(NA, num_ct), winter_rmax = rep(NA, num_ct))
```

```
for (i in 1:nrow(shp_df)) { # looping over the fips  
  
  for (j in 1:4) { # looping over the variables  
  
    suppressWarnings(mean_df[i, j + 1] <- exact_extract(x = mean_array_list[[j]],  
                                                         y = shp_df[i, ], fun = "mean"))  
  
  }  
  
}
```

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