01_Bernardin_Mini_Project_2021

Jessica Bernardin

10/20/2021

Contents

Goals for Mini Project 1

1. Articulate an interesting research question based on a dataset you'd like to learn more about.
2. Develop a spatial database that contains potentially relevant explanatory variables that you'd like to explore in the context of that research question.
3. Demonstrate an understanding of the various workflow elements involved in designing and constructing a spatial database for subsequent visualization and analysis.

Research Question

- Using iNaturalist geotagged observations of *Sarracenia purpurea* plants in North America as a response variable, can predictors like elevation, precipitation, and mean monthly air temperature help inform where plants may be located?
- Or are geographic features like watershed boundaries and land use more useful predictors for *S. purpurea* populations?

Data Sets

Species Occurence – Response Variable

- GBIF stands for the Global Biodiversity Information facility
- They provide open access data about all kinds of living creatures!
- I was able to download a global species occurrence dataset for *Sarracenia purpurea* dataset citation
- This is the original species page, you are able to sort and filter the type data you are interested in and download a csv file, all of the data is open source. Species Page

Elevation Data – Predictor Variable

- Average elevation by county with codes for county and state USGS Elevation Data
- Also contains latitude longitude data for each observation.
- I found this .txt file of elevation data on the USGS website and I thought it would be an interesting predictor for species occurrence.

Land Use Data – Predictor Variable

- I found land use data by state on the USDA website Land Use Data
- This data is divided into different categories of land use in acres for each state in the US.
- As wetlands are converted to urban landscapes and agricultural land, it will be interesting to see how species occurrence correlates with land use type in each state on the east coast.

Watershed Data – Regions

- A shapefile with the watershed data for the United States was downloaded from the USDS website.
- Watershed Data
- This data set (North American Atlas – Basin Watersheds) has a scale of 1:10,000,000. Watersheds will be an interesting way to divide up the landscape and might shed light on future questions like nutrient and pesticide runoff into wetlands.

State Boundries – Regions

- In addition to watersheds, state boundries will be used to help orient the viewer along with helping to summarize the data.

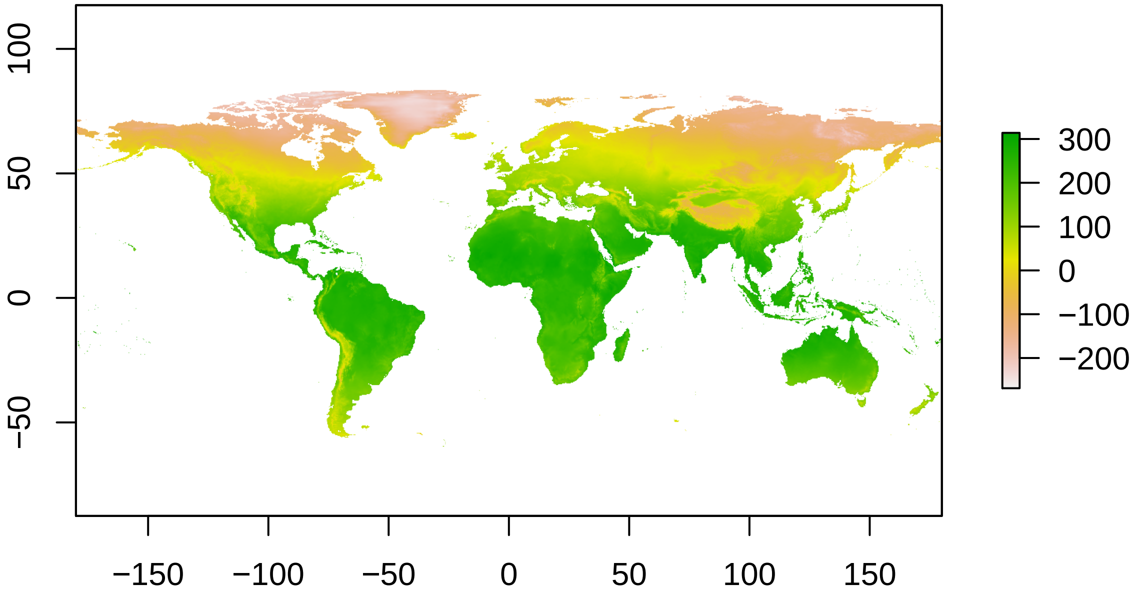
- The cartographic boundary files are build from the Census Bureau's MAF/TIGER geographic database and are available for download as shapefiles State Boundries.

Climate Data – Predictor Variables

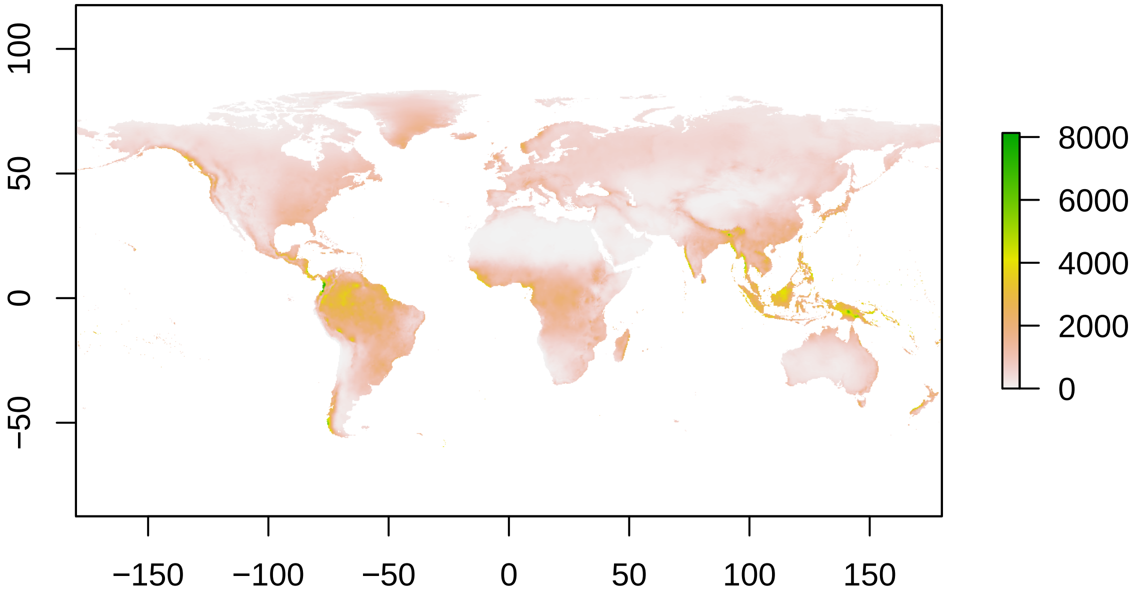
- Lastly, after trying several different ways to get raster data for climate variables I decided to use NCEP North American Regional Reanalysis data.
- I tried downloading GRID files and also the `rnoaa` package but I couldn’t get either to work.
- Here I used the package `ncdf4` to get the mean monthly air temperature and precipitation for January and July, from 2006 to now. I use the 2006-2007 data for my files.
- I have not included the raw data because they are very large files that won’t fit on github, but I have included the two scripts in Project that show how I got the .tif files. I used two examples I found on google to help me with the `ncdf4 package`.
- For this miniproject, I am reading in the two raster files that I created from the scripts called “01_Precip_Raster.R’ and”01_Temp_Raster.R“. I have also included their metadata in the project”air.mon.mean_metadata.txt" and “precip.mon.mean_metadata.txt”.
- Climate Data
- None of the above climate data approaches worked, so I ended up downloading some worldclim data within R. I can’t compare different months but that is ok.

```r
#dependent variable, location of iNat purple pitcher plant observations from GBIF
pitcher <- read.csv("gbif_sarracenia.csv", sep = "\t")

#predictor variable, elevation data for the US


elevation <- read.table("Elevation.US.txt", header = TRUE,
                 sep = "|",
                 na.strings = "",
                 comment.char = "",
                 quote = "\"",
                 fill = FALSE)

#predictor variable, land use data for the US
landuse <- read.csv("MajorLandUse.csv", header = TRUE)

#using worldclim data instead
r <- getData("worldclim",var="bio",res=10)
temp.rast <- r[[1]]
names(temp.rast) <- "Temp"

precip.rast <- r[[12]]
names(precip.rast) <- "Prec"
```

```r
plot(temp.rast)
```



```r
plot(precip.rast)
```



```r
#watersheds
#Projected CRS: Sphere_ARC_INFO_Lambert_Azimuthal_Equal_Area
watersheds <- st_read("NA_Watersheds_Shapefile/watershed_p_v2.shp")
```

```
## Reading layer `watershed_p_v2' from data source
##   `/Users/jessicabernardin/Desktop/R_spatial_Fall 2021/miniproject/01_Bernardin_Spati
##   using driver `ESRI Shapefile'
## Simple feature collection with 2301 features and 38 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -5761945 ymin: -3920000 xmax: 4462000 ymax: 4907000
## Projected CRS: Sphere_ARC_INFO_Lambert_Azimuthal_Equal_Area
```

```
#state boundaries
#NAD83
state <- st_read("us_state_20m/cb_2018_us_state_20m.shp")
```

```
## Reading layer `cb_2018_us_state_20m' from data source
##   `/Users/jessicabernardin/Desktop/R_spatial_Fall 2021/miniproject/01_Bernardin_Spati
##   using driver `ESRI Shapefile'
## Simple feature collection with 52 features and 9 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -179.1743 ymin: 17.91377 xmax: 179.7739 ymax: 71.35256
## Geodetic CRS:  NAD83
```

```
#census data
#https://data.ers.usda.gov/reports.aspx?ID=17827

population <- read.csv("state_population.csv")
```

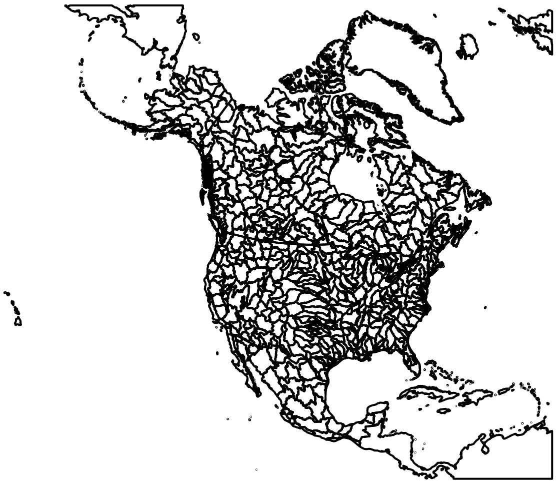
Making the Database

```
#summarize county elevation to state ave elevation
state.elevation <- elevation %>%
  group_by(STATE_ALPHA) %>%
  summarise(mean_elevation = mean(ELEV_IN_M, na.rm = TRUE))

state.elevation <- state.elevation %>%
  rename(Code = STATE_ALPHA)

st.elev.pop <- left_join(population, state.elevation, by = "Code")

#filter the land use data to only year 2007
landuse_07 <- filter(landuse, Year == "2007")
landuse_07 <- landuse_07 %>%
  rename(state = Region.or.State)

#combine with other state data
state.df <- left_join(st.elev.pop, landuse_07, by = c("state"))

# check geometries for polygons
st_is_valid(state) # TRUE
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
st_make_valid(watersheds) # TRUE
```

```
## Simple feature collection with 2301 features and 38 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -5761945 ymin: -3920000 xmax: 4462000 ymax: 4907000
## Projected CRS: Sphere_ARC_INFO_Lambert_Azimuthal_Equal_Area
## First 10 features:
##    OBJECTID UIDENT       NAW1_EN         NAW1_SP         NAW1_FR
## 1         1    116          <NA>            <NA>            <NA>
## 2         2    216          <NA>            <NA>            <NA>
## 3         3    316          <NA>            <NA>            <NA>
## 4         4    416          <NA>            <NA>            <NA>
## 5         5    516          <NA>            <NA>            <NA>
## 6         6    616          <NA>            <NA>            <NA>
## 7         7    716          <NA>            <NA>            <NA>
## 8         8    816 Pacific Ocean Océano Pacífico Océan Pacifique
## 9         9    916          <NA>            <NA>            <NA>
## 10       10   1016          <NA>            <NA>            <NA>
##                   NAW2_EN                 NAW2_SP                  NAW2_FR
## 1                    <NA>                    <NA>                     <NA>
## 2                    <NA>                    <NA>                     <NA>
## 3                    <NA>                    <NA>                     <NA>
## 4                    <NA>                    <NA>                     <NA>
## 5                    <NA>                    <NA>                     <NA>
## 6                    <NA>                    <NA>                     <NA>
## 7                    <NA>                    <NA>                     <NA>
## 8  Pacific Ocean Seaboard Litoral Océano Pacífico Littoral Océan Pacifique
## 9                    <NA>                    <NA>                     <NA>
## 10                   <NA>                    <NA>                     <NA>
##    NAW3_EN NAW3_SP NAW3_FR NAW4_EN NAW4_SP NAW4_FR TRANS_BND MAP_COLOR INTERNAL
## 1     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 2     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 3     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 4     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 5     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 6     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 7     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 8     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999        50     -999
## 9     <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
## 10    <NA>    <NA>    <NA>    <NA>    <NA>    <NA>      -999         2     -999
##    FED_REF COUNTRY USA_REG USA_SUB USA_ACC USA_REG_NA USA_SUB_NA USA_ACC_NA
## 1     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 2     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 3     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 4     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 5     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 6     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 7     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 8     <NA>     MEX    -999    -999    -999       <NA>       <NA>       <NA>
## 9     <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
## 10    <NA>      FN    -999    -999    -999       <NA>       <NA>       <NA>
##    CAN_MDA CAN_SDA CAN_MDA_EN CAN_MDA_FR CAN_SDA_EN CAN_SDA_FR MEX_FC_RH
## 1     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 2     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 3     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 4     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 5     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 6     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 7     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 8     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 9     <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
## 10    <NA>    <NA>       <NA>       <NA>       <NA>       <NA>      -999
##    MEX_REG_RH EDIT_DATE EDIT comments Shape_Leng Shape_Area
## 1        <NA>  20081020  NEW     <NA>  119520.03  503482858
## 2        <NA>  20081020  NEW     <NA>   60671.90   84676627
## 3        <NA>  20081020  NEW     <NA>   16445.06   14507299
## 4        <NA>  20081020  NEW     <NA>   17833.95   18542525
## 5        <NA>  20081020  NEW     <NA>   35729.36   31403643
## 6        <NA>  20081020  NEW     <NA>   33990.62   48070997
## 7        <NA>  20081020  NEW     <NA>   33497.94   49937434
## 8        <NA>  20081020  NEW     <NA>   11209.46    8723358
## 9        <NA>  20081020  NEW     <NA>   19510.78   16384482
## 10       <NA>  20081020  NEW     <NA>   64896.73  255217180
##                          geometry
## 1  POLYGON ((2120932 -3913909,...
## 2  POLYGON ((2166910 -3888448,...
## 3  POLYGON ((2179850 -3860513,...
## 4  POLYGON ((2035409 -3847843,...
## 5  POLYGON ((2044844 -3837379,...
## 6  POLYGON ((2027031 -3831261,...
## 7  POLYGON ((2404638 -3753810,...
## 8  POLYGON ((-1054379 -3751093...
## 9  POLYGON ((2400779 -3742210,...
## 10 POLYGON ((2427767 -3744710,...
#plot(st_geometry(state))
plot(st_geometry(watersheds))
```



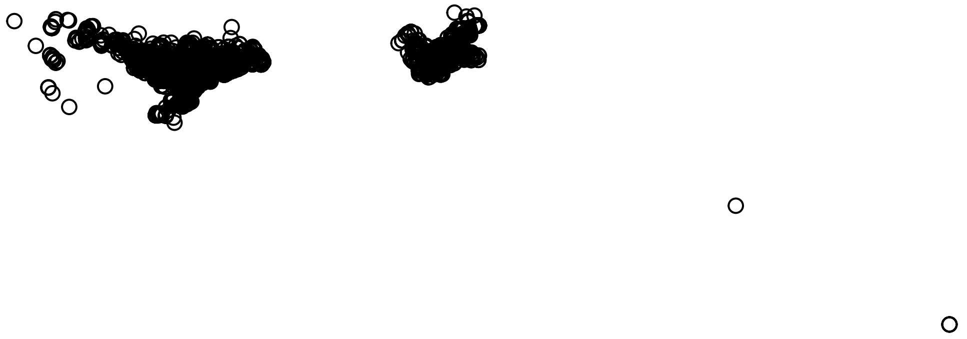
```
#check crs
st_crs(state) == st_crs(watersheds) #FALSE
```

```
## [1] FALSE
```

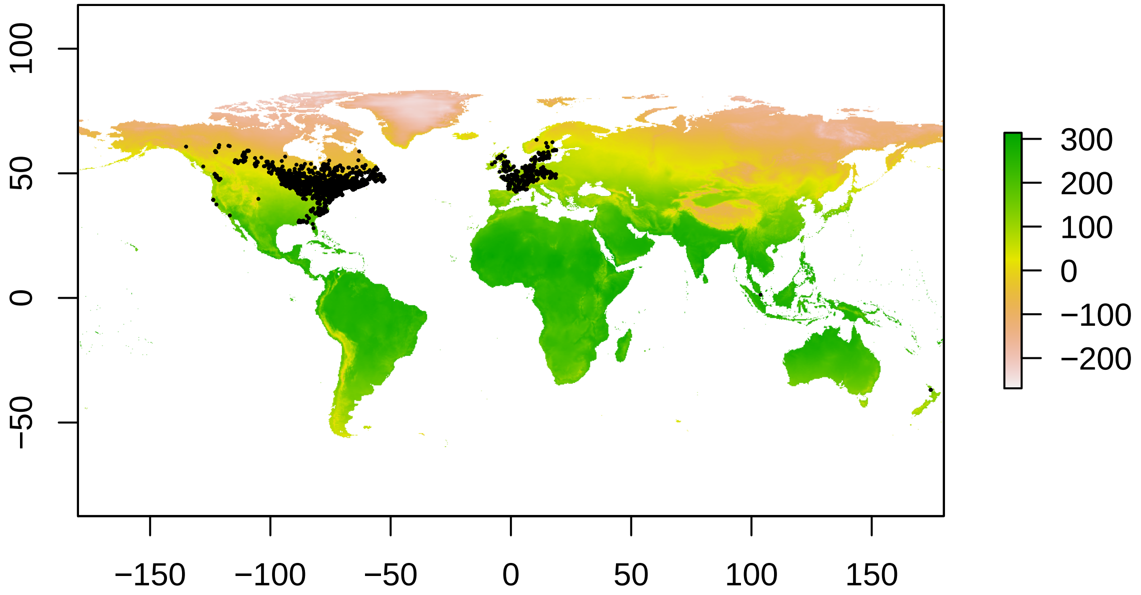
```
#reproject
state <- state %>%
  st_transform(., crs = st_crs(watersheds))

#recheck
st_crs(state) == st_crs(watersheds) #TRUE
```

```
## [1] TRUE
```

```
#Bind state sf with the state tabular data
state.df.sf <- left_join(state, state.df, by = c("STUSPS" = "Code"))

#make pitcher data a shape file

pitcher.sf <- st_as_sf(pitcher, coords = c("decimalLongitude", "decimalLatitude"), crs =

plot(st_geometry(pitcher.sf))
```



```
pitcher.sf.t <- st_transform(pitcher.sf, crs = st_crs(r))

plot(temp.rast)
  plot(st_geometry(pitcher.sf.t),add=T, pch = 20, cex = .2)
```



```
#reproject pitcher.sf
st_crs(state) == st_crs(pitcher.sf)
```

```
## [1] FALSE
```

```
pitcher.sf <- pitcher.sf %>%
  st_transform(., crs = st_crs(watersheds))

st_crs(watersheds) == st_crs(pitcher.sf) #TRUE
```

```
## [1] TRUE
```

#decided not to join the watershed data to the state/elev./population data yet #this chunk takes a while to run

```
####
#pitcher.sf point geometry
```

```
#watershed sf polygons
#state.df.sf state info (population, elevation, land use, state polygons)

#join sf objects
#watershed.state <- st_join(watersheds, state.df.sf, join=st_is_within_distance, dist=

#get all the data to the raster crs
#watershed.state <- watershed.state %>%
  #st_transform(., crs = st_crs(temp.rast))

#st_crs(temp.rast) == st_crs(watershed.state)
```

```
#raster extract
#temp and precip

###DIDNT RUN THIS BECAUSE I COULDN'T GET THE CROP FUNCTION TO FIND THE EXTENT OF Y, EV
###THE PITCHER DATA ISN'T THAT MUCH OF A SMALLER EXTENT THAN THE RASTER ANYWAY SO MAYB
#pitcher.buff <- pitcher.sf.t %>% st_buffer(., 25000)
#pitcher.buf.vect <- as(pitcher.buff, "SpatVector")
#head(pitcher.buf.vect)
#plot(pitcher.buf.vect)
#st_crs(pitcher.buf.vect) == st_crs(temp.rast)
#st_crs(pitcher.sf.t) == st_crs(r)
#a <- vect(pitcher.sf.t)
#a.extent <- ext(-135.0208, 174.5552, -36.9, 63.434)
#temp.crop.a <- terra::crop(temp.rast, a.extent)
#temp.rast
#crop the rasters to just the area where the pitcher plants are
#temp.crop <- crop(temp.rast, extent(pitcher.buf.vect))
#??`crop,SpatRaster-method`
#precip.crop <- crop(precip.rast, pitcher.buf.vect)

#extract the pixels where the buffered pitcher plant points are, average the values an
temp.extract <- terra::extract(temp.rast, pitcher.sf.t, fun = mean, na.rm=TRUE)
pitcher.sf.t$temp <- temp.extract
precip.extract <- extract(precip.rast, pitcher.sf.t, fun = mean, na.rm=TRUE)
pitcher.sf.t$precip <- precip.extract

#Adding the extracted raster data back to the main dataset
#state.df.sf (has state polygons, pop, elev., landuse data)
#watershed (watershed polygons)
#pitcher.sf.t (pitcher points, temp data, precip data from rasters)

#I had combined these sf objects(watershed and state.df.sf) earlier but I noticed I lo
```

```
#summary.watershed.state.pitchers <-  watershed.state.pitchers %>%
 #group_by(watershed) %>%
 #mutate(mean_precipitation = mean(precip.extract, na.rm = TRUE))
```

References