

# Quick maps in R

@cjfiscus

2024/04/19

## load required packages

This broadly follows this tutorial.

```
## install libraries if not already installed, load libraries
if (!require("pacman")) install.packages("pacman")

## Loading required package: pacman
pacman::p_load(
  stars, # spatiotemporal data handling
  raster, # raster data handling
  terra, # raster data handling
  sf, # vector data handling
  dplyr, # data wrangling
  stringr, # string manipulation
  lubridate, # dates handling
  data.table, # data wrangling
  patchwork, # arranging figures
  tigris, # county border
  colorspace, # color scale
  viridis, # arranging figures
  tidyr, # reshape
  ggspatial, # north arrow and scale bar
  ggplot2, # make maps
  rnaturalearth # polygons
)
```

## Make a basic map

First pull in data to make a basic map and restrict coordinates to what you want to show

```
# basic world map with country borders
world_sf <- ne_countries(returnclass = "sf", scale = "large")

# set view window coords here
window_coord<-st_sfc(st_point(c(-120,20)), # left bottom
                      st_point(c(-95, 40))) # right, top

window_coord_sf<-window_coord %>% st_coordinates()
```

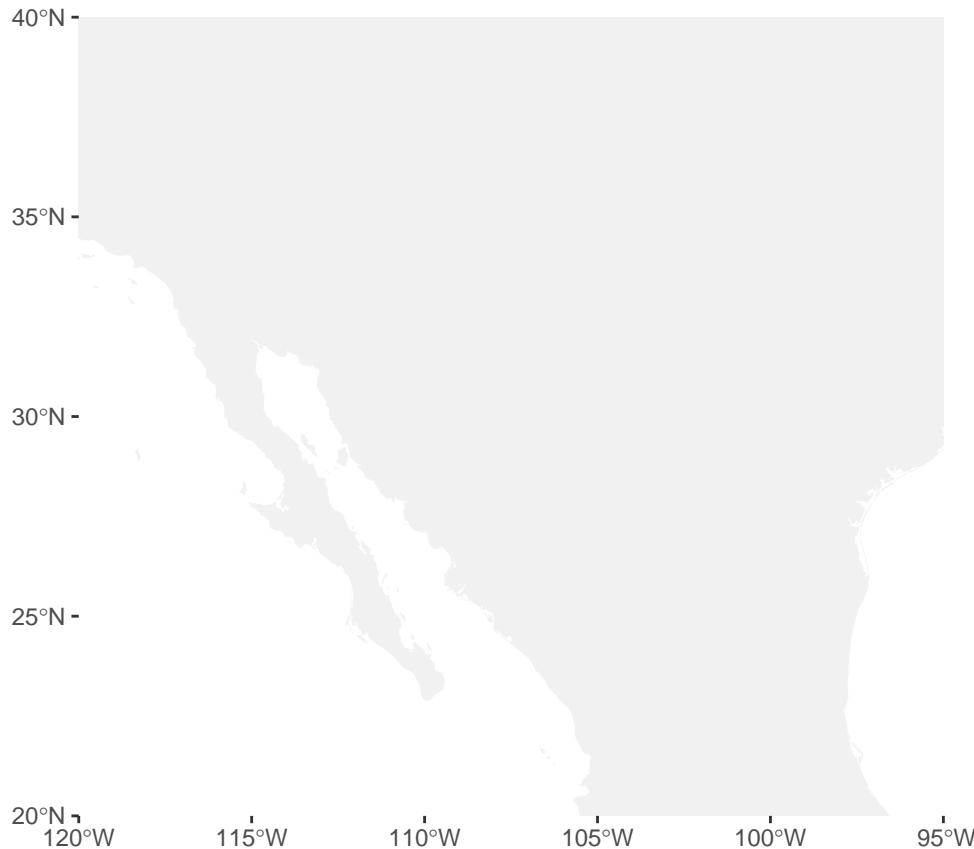
Then plot the map:

```
## plot map
g<-ggplot() +
```

```

geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
coord_sf(xlim>window_coord_sf[, "X"],
         ylim>window_coord_sf[, "Y"], expand=F) +
theme(panel.background = element_rect(fill = 'white'))
## plot map
print(g)

```



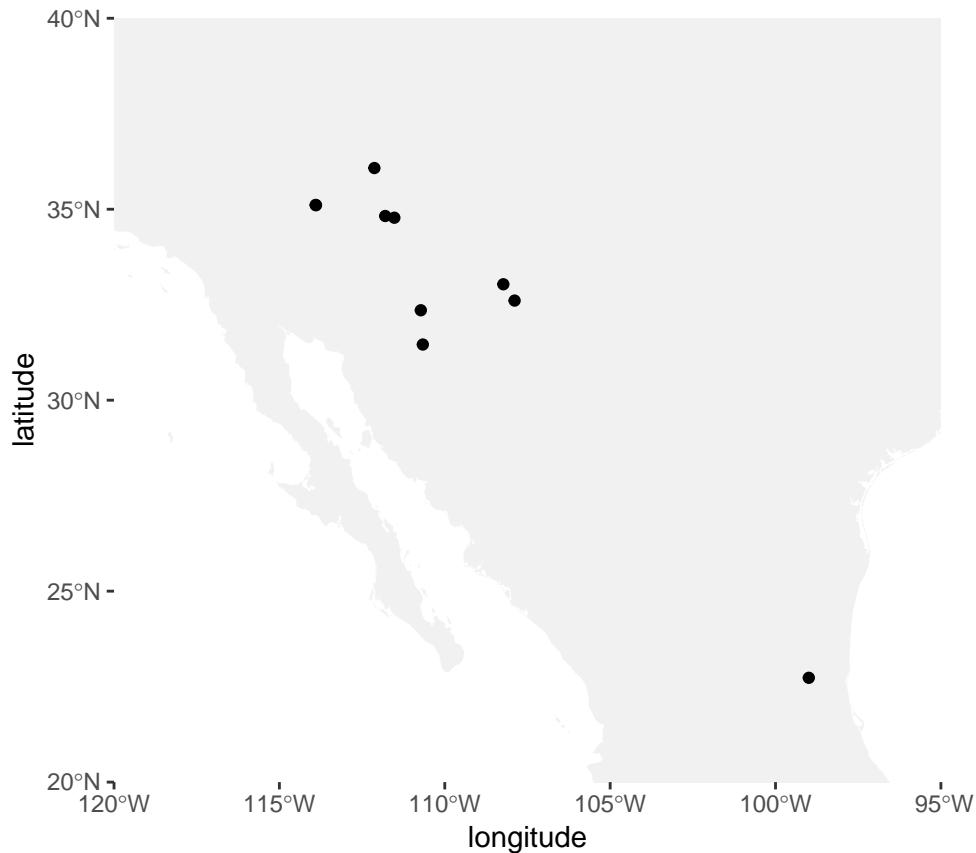
You can add points corresponding to sample collection sites from a df onto the map like this:

```

## import coords to plot
df<-read.delim("./data/coords.txt")

## plot map
g<-ggplot() +
  geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
  coord_sf(xlim>window_coord_sf[, "X"],
           ylim>window_coord_sf[, "Y"], expand=F) +
  geom_point(data=df, aes(x=longitude, y=latitude)) + # code that adds points!
  theme(panel.background = element_rect(fill = 'white'))
## plot map
print(g)

```



## plotting raster layers

To plot raster layers, must first stack layers in R and then convert to df.

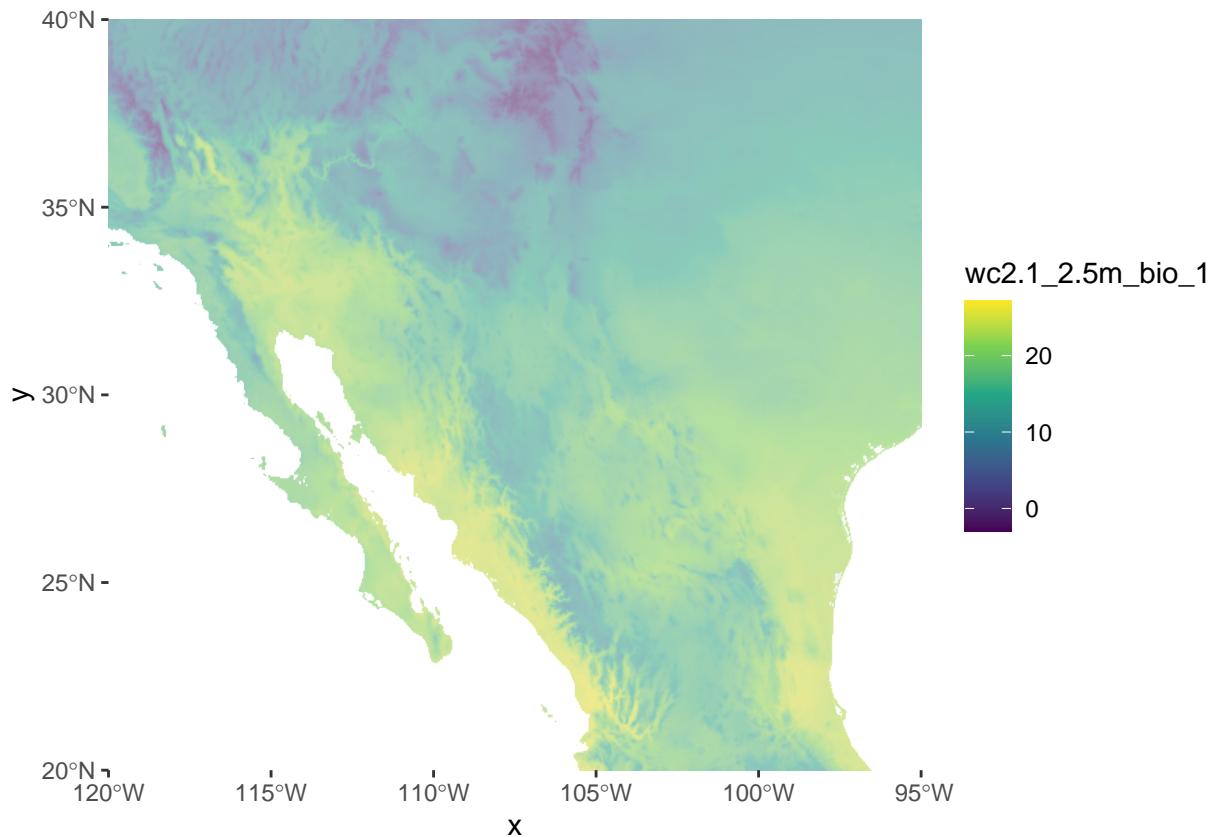
As an example, we're using a subset of the worldclim 2 bioclimatic variables. The bioclimatic variables can be downloaded at various resolutions here. Sample layer is included in data folder for this particular map.

Here's plotting just bio1 across the landscape:

```
## import raster files into raster stack
files<-list.files(path=".~/data/", pattern=".tif", full.names=T)
bioclim<-stack(files)

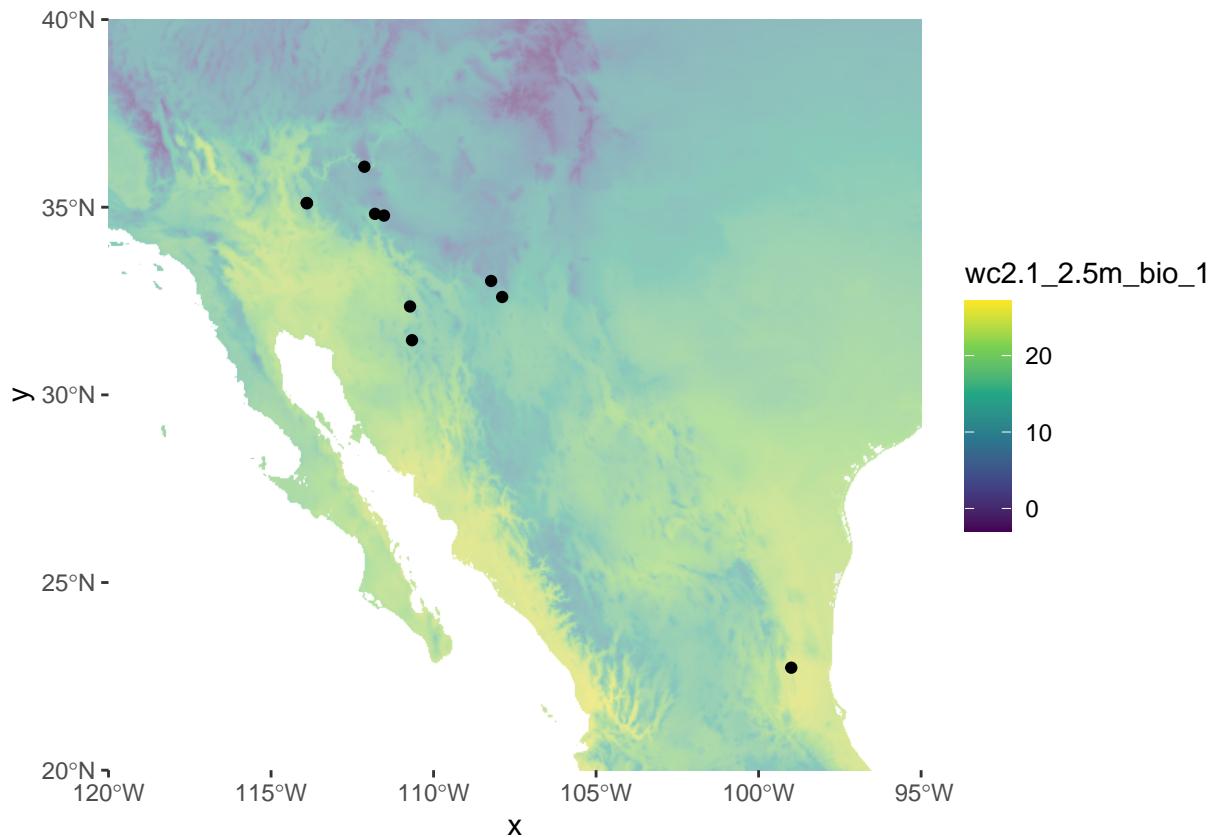
## convert to data.frame
bioclim_df<-as.data.frame(bioclim, xy=TRUE) %>% na.omit()

## map with just 1 bioclim variable
g<-ggplot() +
  geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
  coord_sf(xlim>window_coord_sf[, "X"], ylim>window_coord_sf[, "Y"], expand=F) +
  geom_raster(data=bioclim_df, aes(x=x, y=y, fill=wc2.1_2.5m_bio_1)) + ## fill corresponds to raster layer
  scale_fill_viridis_c() +
  theme(panel.background = element_rect(fill = 'white'), legend.position="right")
print(g)
```



Here's how you can plot samples on top of bio1:

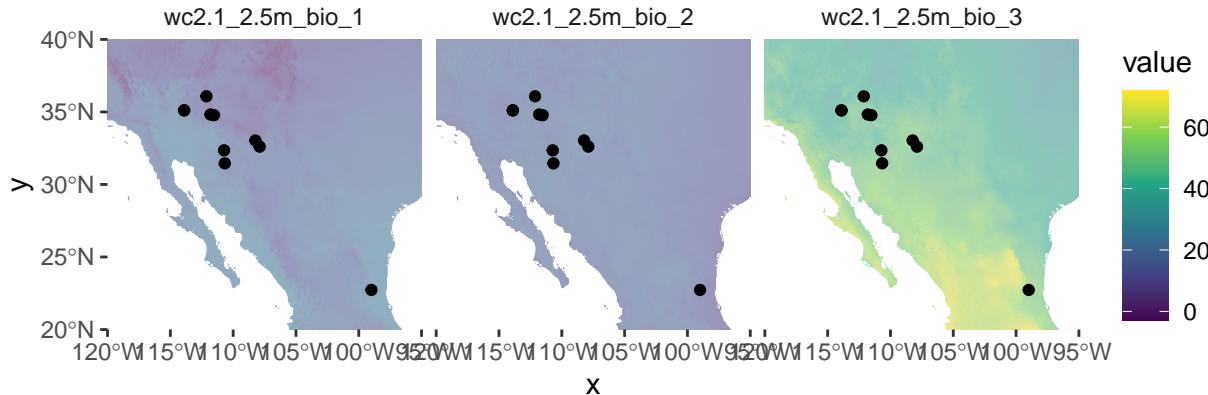
```
## plot points on top of map
g<-ggplot() +
  geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
  coord_sf(xlim>window_coord_sf[, "X"],
            ylim>window_coord_sf[, "Y"], expand=F) +
  geom_raster(data=bioclim_df, aes(x=x, y=y, fill=wc2.1_2.5m_bio_1)) +
  geom_point(data=df, aes(x=longitude, y=latitude)) +
  scale_fill_viridis_c() +
  theme(panel.background = element_rect(fill = 'white'), legend.position="right")
print(g)
```



Finally, let's plot samples on top of multiple bioclim variables using facets:

```
## first convert df from wide to long
bioclim_df2<-bioclim_df %>% pivot_longer(cols=starts_with("wc2.1_2.5m"), names_to=c("bio"), values_to=""

g<-ggplot() +
  geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
  coord_sf(xlim>window_coord_sf[, "X"],
            ylim>window_coord_sf[, "Y"], expand=F) +
  geom_raster(data=bioclim_df2, aes(x=x, y=y, fill=value)) + # notice changes to df and fill
  geom_point(data=df, aes(x=longitude, y=latitude)) +
  scale_fill_viridis_c() +
  theme(panel.background = element_rect(fill = 'white'), legend.position="right",
        strip.background = element_blank()) + # removes grey in facet box
  facet_wrap(bio~.)
print(g)
```



This is pretty good except that all facets are forced to share the same legend. Since the variables are on different scales this makes it difficult to see the variation for each variable. This problem can be solved by merely making a separate plot for each variable and then plotting them together with patchwork.

```
## same code as earlier (bio1)
g1<-ggplot() +
  geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
  coord_sf(xlim>window_coord_sf[, "X"],
            ylim>window_coord_sf[, "Y"], expand=F) +
  geom_raster(data=bioclim_df, aes(x=x, y=y, fill=wc2.1_2.5m_bio_1)) +
  geom_point(data=df, aes(x=longitude, y=latitude)) +
  scale_fill_viridis_c() +
  theme(panel.background = element_rect(fill = 'white'), legend.position="bottom")

## bio 2
g2<-ggplot() +
  geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
  coord_sf(xlim>window_coord_sf[, "X"],
            ylim>window_coord_sf[, "Y"], expand=F) +
  geom_raster(data=bioclim_df, aes(x=x, y=y, fill=wc2.1_2.5m_bio_2)) +
  geom_point(data=df, aes(x=longitude, y=latitude)) +
  scale_fill_viridis_c() +
  theme(panel.background = element_rect(fill = 'white'), legend.position="bottom")

## bio 3
g3<-ggplot() +
  geom_sf(data=world_sf, lwd=0, alpha=0.5, color=NA) +
  coord_sf(xlim>window_coord_sf[, "X"],
            ylim>window_coord_sf[, "Y"], expand=F) +
  geom_raster(data=bioclim_df, aes(x=x, y=y, fill=wc2.1_2.5m_bio_3)) +
  geom_point(data=df, aes(x=longitude, y=latitude)) +
  scale_fill_viridis_c() +
  theme(panel.background = element_rect(fill = 'white'), legend.position="bottom")

## put together
p_load(patchwork)

## put plots side by side with "/"
g1 | g2 | g3
```

