Welcome to our UCSB Library Collaboratory GeoSpatial R Carpentries Etherpad!

https://ucsbcarpentry.github.io/2021-05-13-GeospatialR/

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try https://etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

---

JonJab, UCSB, downtown SB
Kristi Liu, UCSB, 1000 Oaks
Maddie Frick UCSB Isla Vista
Mari Cajandig (Noleta)
Mario, York University, Toronto Canada
Casey O'Hara, UCSB Bren School, east side SB
Jenny Couture, Salem, MA
Sravan, UC Berkeley
Evan Hobson, UCSB, downtown SB (Westside)
Greg Janée, UCSB, biking distance of UCSB so I lose
Dave Hunter (Montecito)
Wai-Yin, freelance learner, Los Angeles
Nargol Ghazian, Toronto, York University
Leonardo, UCSB Bren School, West Campus housing
z
Michael Topper, Noleta
Daniela Soleri, Noleta - also biking distance to library
Reetika Sindhi, New Delhi, India
Jai Uparkar, Cupertino, CA
Ann, UCSB, Cheadle Center for Biodiversity and Restoration
Dana, the Mesa in Santa Barbara

1. How do we get the Max and Min from the dataframe version?

dsm_harv_df %>%
summarize(min(HARV_dsmCrop)) ### +1
summary (DSM_HARV_df)? ### +1
min(DSM_HARV_df) ### almost... which variable? min(DSM_HARV_df$HARV_dsmCrop).
min(DSM_HARV_df) does return the min in this specific case, but only because the x and y values are

huge... so it works, but not like you'd think it is working...

```
 2 layer map"
 # 2 layers
ggplot() +
  geom_raster(data = DSM_HARV_df ,
              aes(x = x, y = y,
                  fill = HARV_dsmCrop)) +
  geom_raster(data = DSM_hill_HARV_df,
              aes(x = x, y = y,
                  alpha = HARV_DSMhill)) +
  scale_fill_viridis_c() +
  scale_alpha(range = c(0.15, 0.65), guide = "none") +
  ggtitle("Elevation with hillshade") +
  coord_quickmap()
```

Final Day 1 Challenge
Use the files in the NEON_RemoteSensing/SJER/ directory to create a Digital Terrain Model map and Digital Surface Model map of the San Joaquin Experimental Range field site.

Make sure to:

include hillshade in the maps,
label axes on the DSM map and exclude them from the DTM map,
include a title for each map,
experiment with various alpha values and color palettes to represent the data.

Workshop R Scripts:
https://drive.google.com/drive/folders/12l0Oi7kQ82b6729owzOkJZ_n468ayQqG?usp=sharing

---

Day 2 Notes and Challenges

this is a ggplot2 book that includes a mapping chapter that I find to be an excellent reference:
https://ggplot2-book.org/maps.html
O-Reilly has a book specifically about geospatial R:
https://learning.oreilly.com/library/view/learning-r-for/9781783984367/
UCSB Library has free access to O-Reilly books.

Day 3

To force your text attributes to be factors in R4+, use this:
lines_HARV <- st_read("data/NEON-DS-Site-Layout-Files/HARV/HARV_roads.shp", stringsAsFactors = TRUE)

if factors, find unique values with levels()
if text, use unique()

Good ggplot2 theme reference:
Here's the code that makes it work: lines_HARV <-
st_read("data/NEON-DS-Site-Layout-Files/HARV/HARV_roads.shp", stringsAsFactors = TRUE)

Juliano calls a function for his favorite ggplot parameters:
In my case, I created a function that has all the specifics I like for a plot to look like (for example, no axis line, size of labels, background style, etc.). I just call the function at the end of the ggplo() +… so all my plots are the same. You can check it put here:
https://github.com/jepa/MyFunctions/blob/master/R/my_ggtheme_p.R

Student question about overlaying a manual polygon on top of spatially-explicit data via ggplot
Example uses code from second challenge from episode 9:

```
ggplot() +
  geom_sf(data = ne_state_boundary_us, aes(color = "color"), show.legend = "line") +
  scale_color_manual(name = "", labels = "State Boundary", values = c("color" = "gray18")) +
  geom_sf(data = point_HARV, aes(shape = "shape"), color = "purple", size = 3) +
  scale_shape_manual(name = "", labels = "Fish Tower",
                     values = c("shape" = 22),
                     guide = guide_legend(override.aes = list(linetype = 0))) +
  geom_polygon(data = data.frame("x" = c(-83, -83, -70, -70), "y" = c(40,45,45,40)),
               mapping = aes(x = x, y = y), color = "orange", fill = NA) +
  ggtitle("Fish Tower Location") +
  coord_sf()
```

where geom_polygon is defined by a dataframe that gives the coordinates for the bounding box of the polygon x = c(xmin, xmin, xmax, xmax), y = c(ymin, ymax, ymin, ymax) (because using x and y together we get: lower left point, upper left point, lower right point, upper right point -- aka boundary box of the polygon we want to create)