**Mapping in R**

library(tidyverse)

library(sf)

library(tigris)

library(urbnmapr) # install from github first (see below)

When mapping there are a few **key tasks** you need to attend to:

1. importing/uploading boundary/shapefiles (perhaps shifting AK/HI)
2. filtering – either in the process of importing or later using dplyr or st\_intersection
3. joining or merging data, tyipcally with dplyr’s left\_join() function
4. plotting with geom\_sf()
5. other possibilities:
   1. statebins (squares for states)
   2. hexbin map (instead of usual states)
   3. cartogram (distorted by some other variable, such as population)
   4. geofacet – many small graphs in the layout of states, etc.
   5. small multiples with facet\_wrap()
   6. and more!

**tigris** – used to import boundary/line files from the Census Bureau’s TIGER/line shapefiles repository.

Set cb = TRUE (cartographic boundaries) because they’ll download in small data format.

Boundaries/lines that can be downloaded: blocks (will be huge unless you filter), block\_groups (also huge), tracts (pretty big), counties, states, urban\_areas, roads, rails, school\_districts (big), tribal\_block\_groups and tribal\_census\_tracts, zctas

Note: the states data will include Guam, American Samoa, United States Virgin Islands, Commonwealth of the Northern Mariana Islands, Puerto Rico. You may want to filter these out, as they may impact your map and there is often little data for them. See example below for one way to do this.

%>% shift\_geometry() # This will shift AK and HI over to the other 48 states

**Downloading boundary files using tigris**

Example:

states <- tigris::states(cb = TRUE) %>%

filter() %>% # impose a condition if you want just particular states

shift\_geometry() # if need to move AK and HI over

states <- tigris::states(cb = TRUE) %>%

filter(!NAME %in% c(“Guam”, “American Somoa”, etc……) %>%

shift\_geometry()

# tigris is pretty intuitive and will accept state/county names, fip codes, or combination of…

eptracts <- tigris::tracts(cb = TRUE, year = 2018, state = “Colorado”, county = c(“041”))

epdgtrcts <- tigris::trcts(cb = TRUE, year = 2018, state = “Colorado”,

county = c(“El Paso County”, “Douglas County”))

**Downloading boundary files (states or counties only) using Urban Institute’s urbnmapr**

devtools::install\_github(“UrbanInstitute/urbnmapr”)

statesui <- urbnmapr::get\_urbn\_map(“states”, sf = TRUE) %>%

shift\_geometry()

countiesui <- urbnmapr::get\_urbn\_map(“counties”, sf = TRUE) %>%

shift\_geometry()

**Mapping with sf()**

ggplot(data = df, aes()) +

geom\_sf() + # equivalent to other geoms; can add aes() here, etc.

theme\_void() # try it without theme\_void to see the difference

**Joining/Merging Data to Boundary File**

I recommend using left\_join (part of dplyr) to join .csv data to your boundary file.

\*\*Be sure to check your variable names that will be used for matching and how they are classified. You may need to reclassify first.

fulldata <- left\_join(bndrydata, csvdata, by = c(“bndrydatavar” = “csvdatavar”) # if two

variables are named different things

**Tasks to Excerise Your Skill Set**

1. Mapping Mining Data
   1. Use tigris() to import county boundaries for TN, KY, WV, and VA.
   2. Join the mining csv data to those boundaries
   3. Make a choropleth map using ggplot() + geom\_sf()
2. El Paso County Neighborhood Maps
   1. Upload and then import the zipfile (neighborhoods.zip in the EPneighbor folder on github) containing the EP neighborhood boundaries
   2. Import and then join the Epneighborhood.csv data to those boundaries.
   3. Make some plots:
      1. a choropleth of mortality rate;
      2. a choropleth of a poverty rate;
      3. a scatterplot of the two; use ggrepel to add neighborhood labels
      4. use patchwork to patch them together.
3. Life Expectancy at Census Tract Level in Colorado/Denver
   1. Use tigris() to import census tracts (where state is Colorado, year is 2018)
   2. Join life expectancy data to those tracts.
      1. Make a map of life expectancy
   3. Use tigris() to import urban areas
      1. filter for Denver—Aurora, CO
      2. Use st\_intersection() to get tracts that intersect with that urban area
      3. Make a choropleth