**### LEAFLET AND BACKGROUNDS**

## Choose favorite backgrounds in:

https://leaflet-extras.github.io/leaflet-providers/preview/

beware that some need extra options specified

# Packages

install.packages("leaflet")

library(leaflet)

**# Using Markers**

popup = c("Robin", "Jakub", "Jannes")

leaflet() %>%

addProviderTiles("Esri.WorldPhysical") %>%

#addProviderTiles("Esri.WorldImagery") %>%

addAwesomeMarkers(lng = c(-3, 23, 11),

lat = c(52, 53, 49),

popup = popup)

**## Setting View**

leaflet() %>%

addTiles() %>%

addProviderTiles("Esri.WorldImagery",

options = providerTileOptions(opacity=0.5)) %>%

setView(lng = 151.005006, lat = -33.9767231, zoom = 10)

**## Europe with layers control**

leaflet() %>%

addTiles() %>%

setView( lng = 2.34, lat = 48.85, zoom = 5 ) %>%

addProviderTiles("Esri.WorldPhysical", group = "Physical") %>%

addProviderTiles("Esri.WorldImagery", group = "Aerial") %>%

addProviderTiles("MtbMap", group = "Geo") %>%

addLayersControl(

baseGroups = c("Geo","Aerial", "Physical"),

options = layersControlOptions(collapsed = T))

**### PLOTTING IN LEAFLET USING ONLINE HOTSPRING DATA FROM CALIFORNIA**

### Example from UC Davis RUser group

### https://ryanpeek.github.io/2017-08-03-converting-XY-data-with-sf-package/

**# Packages**

install.packages("htmltab")

install.packages("leaflet")

**# Load libraries**

suppressMessages({

library(tidyverse)

library(sf)

library(htmltab)

})

**# Download data**

url <- "http://www.hotspringsdirectory.com/usa/ca/gps-usa-ca.html"

springs <- htmltab(url, which=1, header = 1,

colNames = c("STATE","LAT","LONG","SpringName","Temp\_F", "Temp\_C", "AREA", "USGS\_quad") )

**# Convert Lat/Long columns to numeric:**

sapply(springs, class) # dataframe colums are of character class now

cols.num <- c("LAT","LONG", "Temp\_F", "Temp\_C") # select columns which need to be numeric

springs[cols.num] <-sapply(springs[cols.num], as.numeric)

# beware that some NA's will appear where there are no temperatures available

springs$LONG <- springs$LONG\*(-1) # sanity check given the western hemisphere

head(springs)

# The chunk above does the following:

# Function from the htmltab package goes and grabs the first table on the page,

# parses it out into a dataframe, and adds custom column names.

# We should have 303 rows and 8 columns of data, including the temperature in °F and °C.

# Then identify the column classes and select columns we want to convert to numeric.

# Apply those changes to those columns using sapply

# Make sure our longitude is negative, so that these plot in the right hemisphere. :)

**# Create simple feature / Make the UTM cols spatial** (X/Easting/lon, Y/Northing/lat)

springs.SP <- st\_as\_sf(springs, coords = c("LONG", "LAT"), crs = 4326)

st\_crs(springs.SP)

**# to project the file to UTM 10N**

st\_crs(32610)

springs.SP <- st\_transform(springs.SP, crs=32610) # but tiles in Leaflet use Web Mercator 4326

**# To get projected coordinates (Northing&Easting) and feed them back into the csv**

springs.SP$utm\_E <- st\_coordinates(springs.SP)[,1]

springs.SP$utm\_N <- st\_coordinates(springs.SP)[,2]

st\_coordinates(springs.SP)

**# Coerce back to data.frame:**

springs.SP<-st\_set\_geometry(springs.SP, NULL)

plot(springs.SP)

**## PLOTTING in Leaflet**

# Reproject into Web Mercator for display in Leaflet

springs.SP <- st\_transform(springs.SP, crs=4326) # Base imagery is 3D

library(leaflet)

leaflet() %>%

addTiles() %>%

addProviderTiles("Esri.WorldTopoMap", group = "Topo") %>%

addProviderTiles("Esri.WorldImagery", group = "ESRI Aerial") %>%

addCircleMarkers(data=springs.SP, group="Hot Springs", radius = 4, opacity=1, fill = "darkblue",stroke=TRUE, fillOpacity = 0.75, weight=2, fillColor = "yellow",

popup = paste0("Spring Name: ", springs.SP$SpringName,

"<br> Temp\_F: ", springs.SP$Temp\_F,

"<br> Area: ", springs.SP$AREA)) %>%

addLayersControl(

baseGroups = c("Topo","ESRI Aerial", "Night"),

overlayGroups = c("Hot SPrings"),

options = layersControlOptions(collapsed = T))