R Plotting and Mapping Tutorial for the Institute for Tribal Environmental Professionals

Downloading this tutorial

If you are familiar with Github and `git clone` you can clone this repository to start. Or, you can download this repository as a zip file and then unzip it (preferably to the Desktop so that you can easily find it.)

What is RMarkdown?

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Plotting and Mapping Temperature Station Data in R

We will be plotting hourly, multi-annual temperature data from the Mattamiscontis watershed in Maine and associating it with a spatial plot of the site locations. Our goal is to understand how trends in temperature vary across both time and space. We will make use of modern R libraries that let us do more with less lines of code. In the end we will be able to generate a report from our code with just our plots and description, which will hopefully look good enough to submit to your colleague/manager/agency/publication. We'll also check out the mapplots library if we have time and think about how we could adapt a ready made example to our own data.

To start we will need the following packages installed.

Data Description and Tasks To Do in R (courtesy of Angie Reed)

Our data/ is located in the PINTempData folder. It contains …

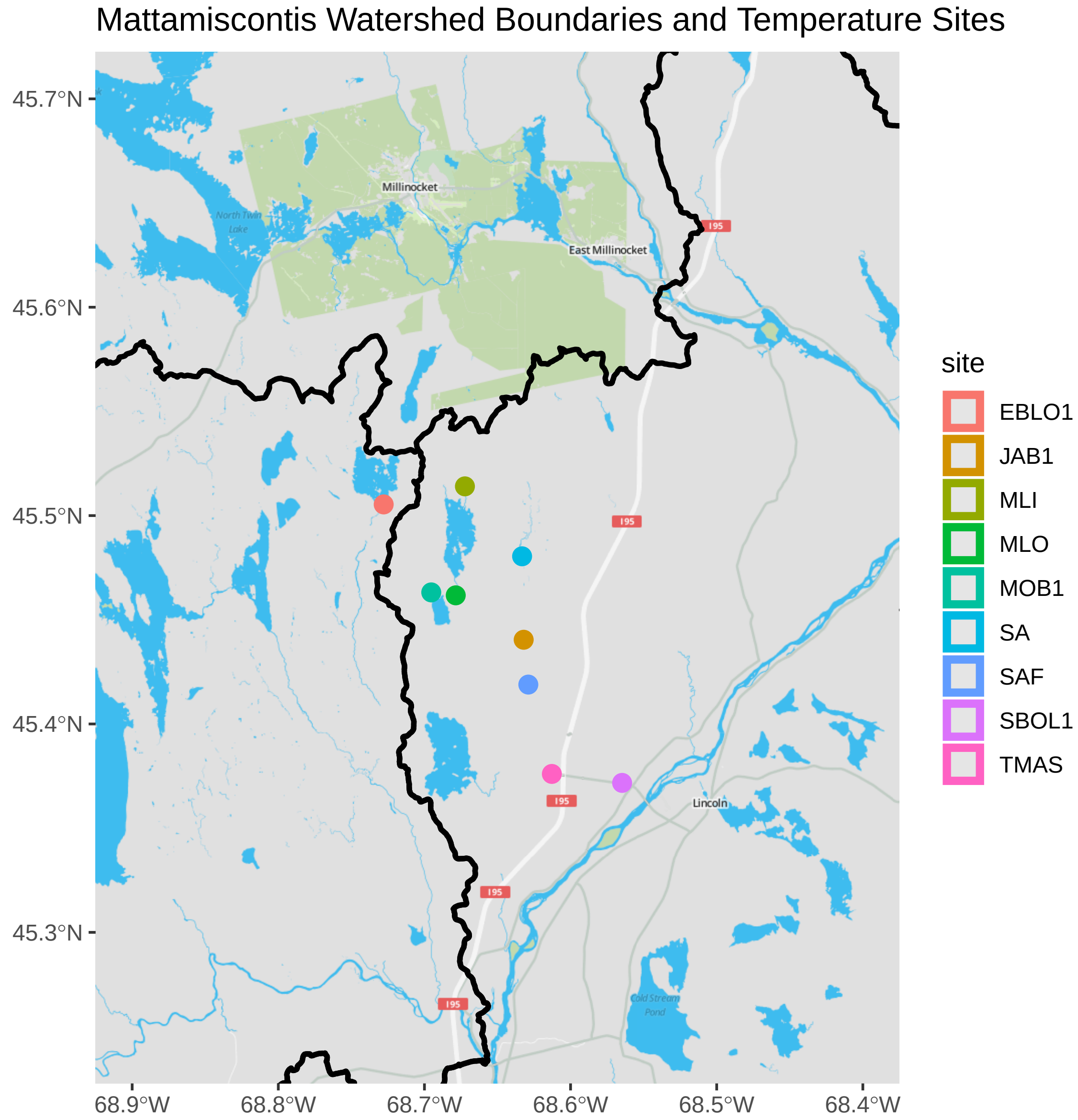
1. An excel file that has coordinates for each of the Penobscot Indian Nation's temperature sensor sites that have data in the csv files

    - The file called PIN_MattamiscontisSites_LDRTimes.xlsx has the lat and long values for each temperature sensor site, along with some other information. There are sometimes multiple rows for the same site, which has to do with keeping track of when we downloaded the file and what times to clip out of the data. **We need to get just one unique row entry for each site so that we can plot each site once along with the watershed boundary.**

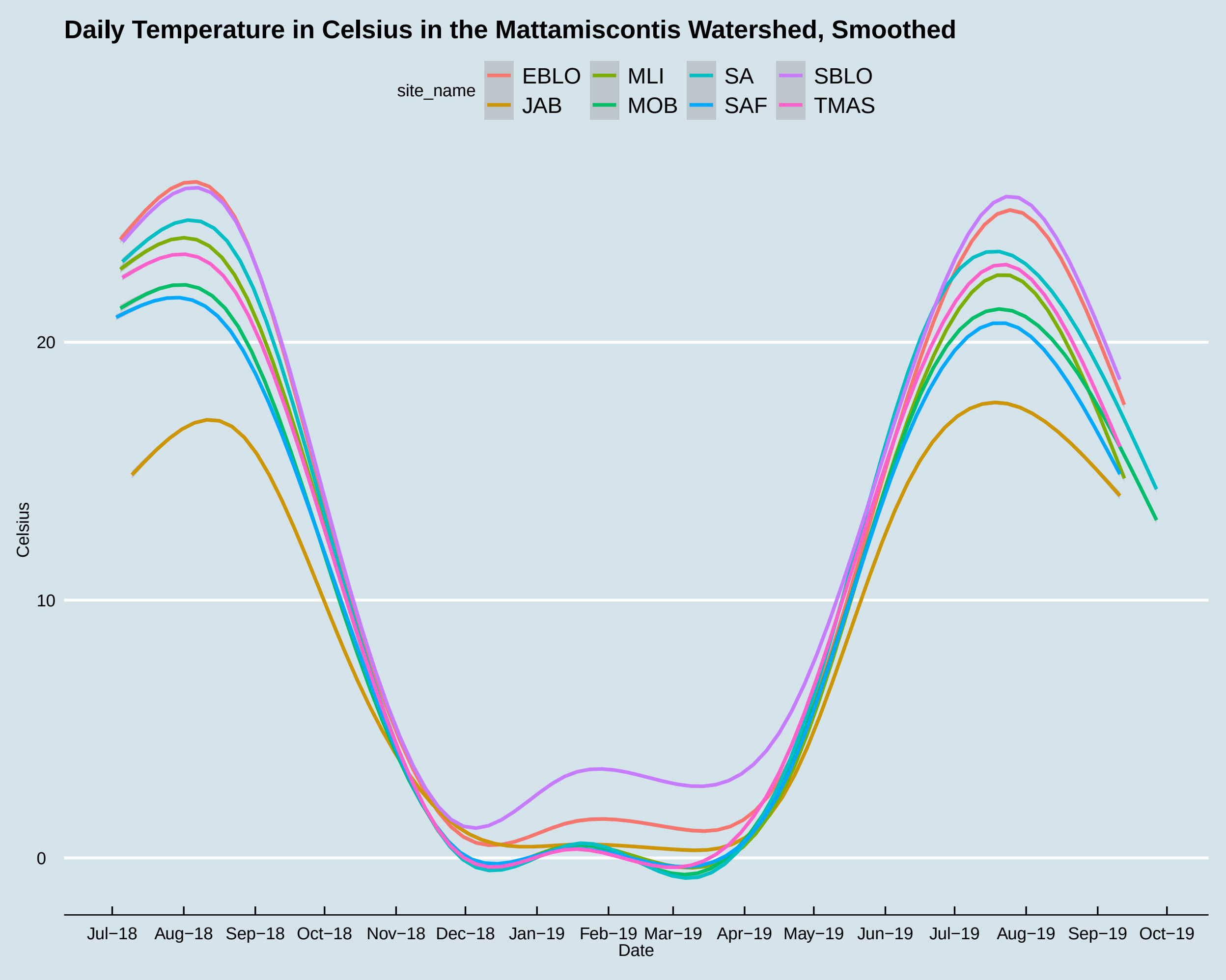
2. a zipped folder inside of that which has the files needed for the watershed boundary shapefile. **We need to plot this with the station data locations to give context to our timeseries plot**
3. A bunch of csv files with temperature station data in them
    - Each of these files has the site name at the start, before the first underscore. These site names match the sites in the file described in #3.
    - These files come from the temperature recorders in csv (comma seperated value) format. **We need to ignore the first two rows, ignore the first column and rename the remaining two columns to Date and Celsius so that the data re ready to be plotted**
    - Once our data is cleaned, **we need to plot the temperature over time with a legend that matches our spatial plot of the site locations and watershed boundary**

Throughout this tutorial we will breakdown what is going on in each code chunk line by line. We will create new code chunks on the fly with the *Ctrl+Alt+I* keyboard shortcut to inspect the output of code lines.

Note: Non-important messages and warnings that are generated by our code have been ignored so that only the plots are presented in the report.

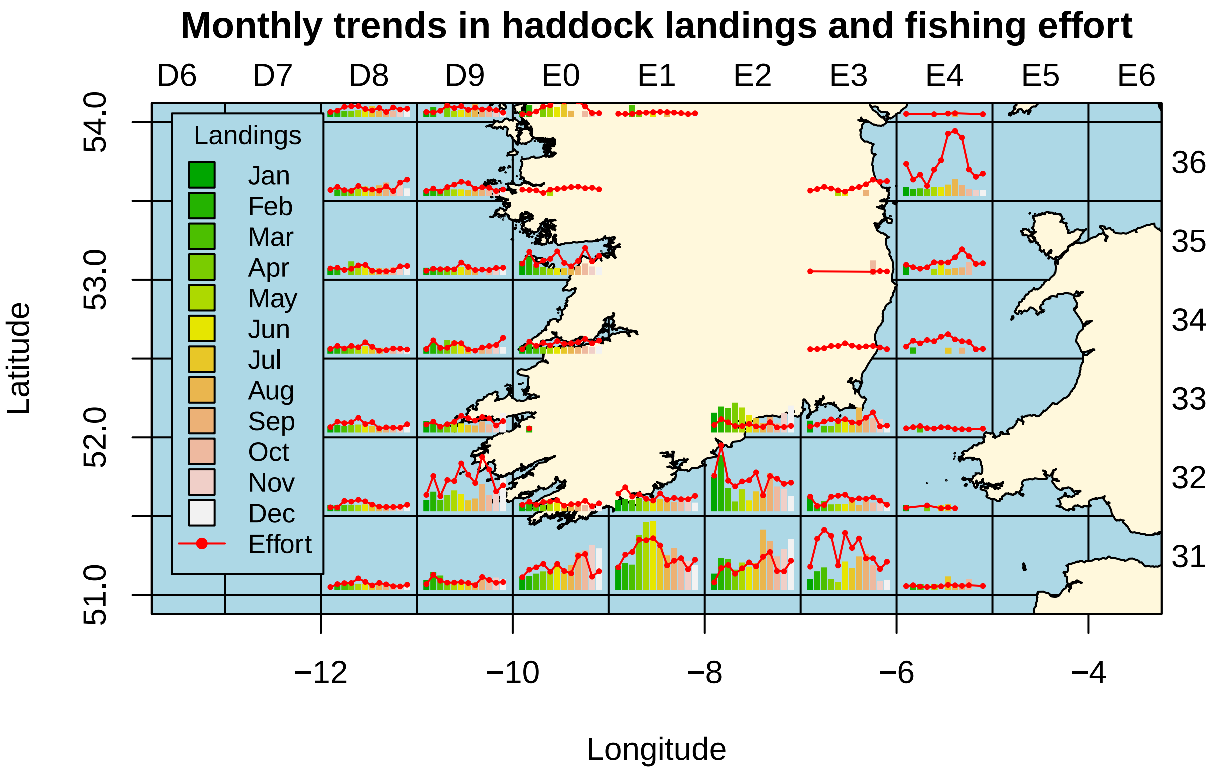


attribution: Tiles courtesy of http://openstreetmap.se/ OpenStreetMap Sweden; Map data © OpenStreetMap contributors



And here is an example of including code and the resulting plot in a report.

```
# This example is sourced from
# https://www.rdocumentation.org/packages/mapplots/versions/1.5.1/topics/draw.xy
data(effort)
data(coast)
xlim <- c(-12,-5)
ylim <- c(51,54)
col <- terrain.colors(12)
effort$col <- col[match(effort$Month,1:12)]
basemap(xlim, ylim, main = "Monthly trends in haddock landings and fishing effort")
draw.rect(lty=1, col=1)
draw.shape(coast, col="cornsilk")
draw.xy(effort$Lon, effort$Lat, effort$Month, effort$LiveWeight, width=1, height=0.5,
 col=effort$col, type="h",lwd=3, border=NA)
draw.xy(effort$Lon, effort$Lat, effort$Month, effort$Effort, width=1, height=0.5, col="red",
 type="l", border=NA)
draw.xy(effort$Lon, effort$Lat, effort$Month, effort$Effort, width=1, height=0.5, col="red",
 type="p",cex=0.4,pch=16, border=NA)
legend("topleft", c(month.abb,"Effort"), pch=c(rep(22,12),16), pt.bg=c(col,NA),
 pt.cex=c(rep(2,12),0.8),col=c(rep(1,12),2), lty=c(rep(NA,12),1), bg="lightblue",
 inset=0.02, title="Landings", cex=0.8)
```



Final Advice

The resources recommended here are high-quality, free, and there are different options depending on whether you are completely new to R/Rstudio or want to learn about a particular topic: https://www.tidyverse.org/learn/

When using R, generally expect that someone else has already writtena function to help you out. In the course of making this lesson I discovered the packages `readxl` for reading excel data, `scales` for controlling the frequency and placement of x axis labels, flipTime for converting characters to Date objects, and basemapR for fetching basemaps from online providers to use in in ggplots. Your browser is your best friend when it comes to coding and websites like StackOverflow provide many useful code examples and answers to common R questions.