R Notebook

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.

To start we will need the following packages installed.

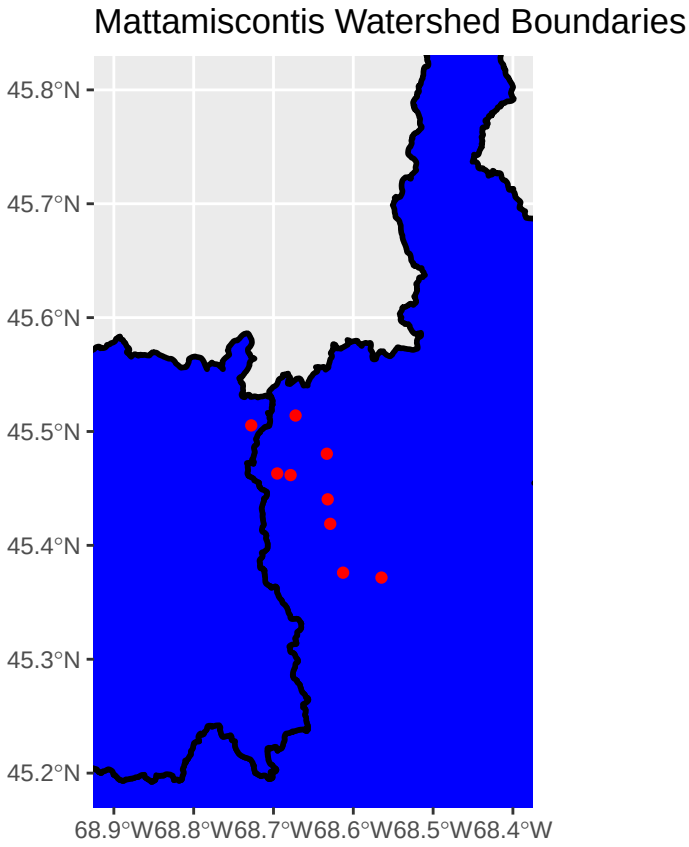
Data Description (courtesy of Angie Reed)

1. a bunch of csv files with data in them

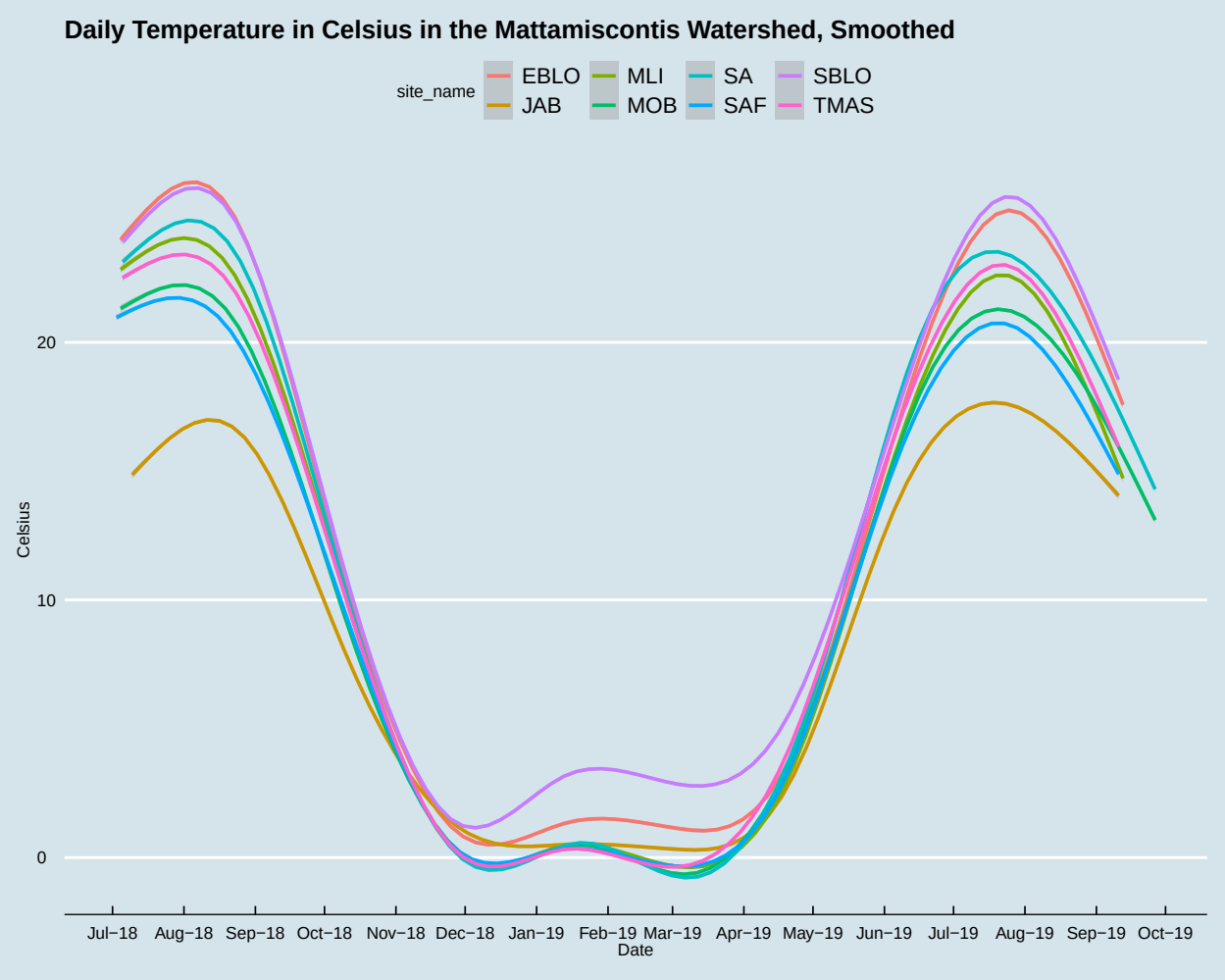
a. Each of these files has the site name at the start, before the first underscore. i. These site names match one (or more) listed in column A of the file described in #3.
b. These files are what get created when we export raw sensor data from our temperature recorders to a csv. They add stupid stuff to the first row and to the column header for the temperature data. You could ignore the first two rows, delete the first column and rename the remaining two columns Datetime and Temperature. They should all be in degrees C.
c. For the mapping exercise perhaps we could just plot the temperature over time and add each of the graphs for each site to their corresponding location?

2. a zipped folder inside of that which has the files needed for the watershed boundary shapefile
3. an excel file that has coordinates for each of the sites that have data in the csv files

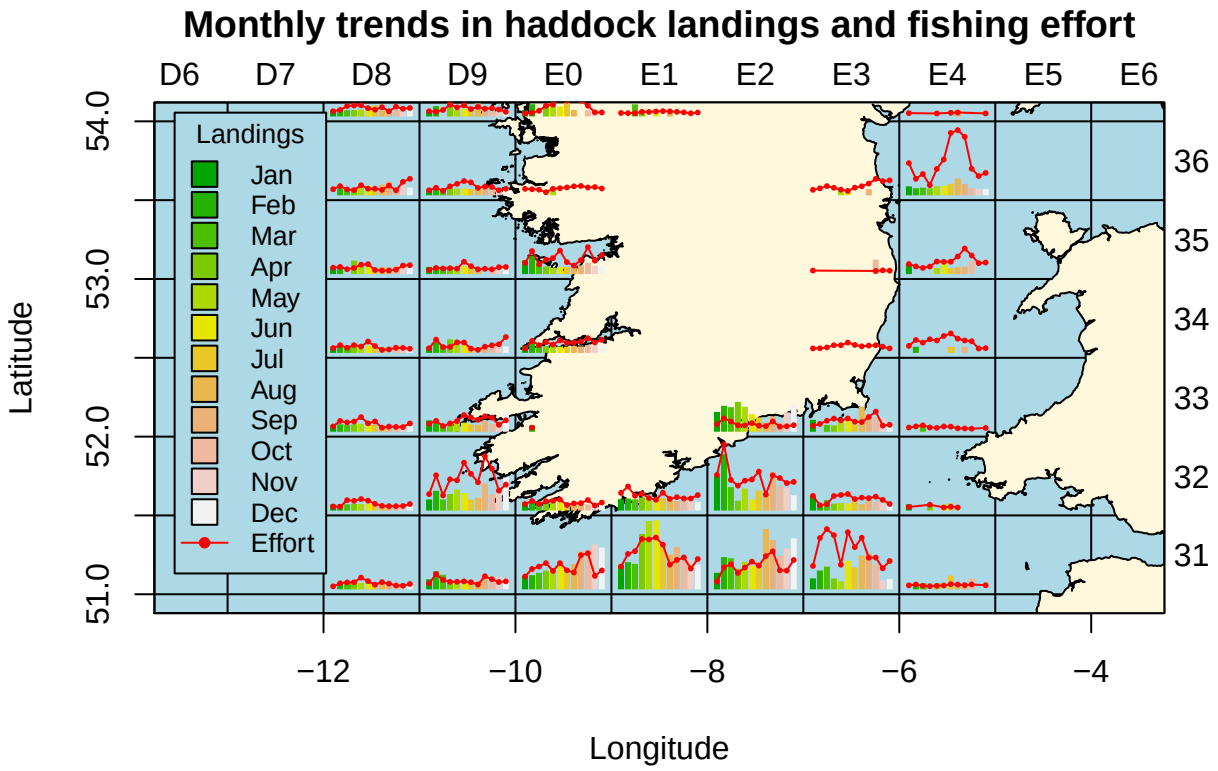
a. The file called PIN_MattamiscontisSites_LDRTimes.xlsx has the lat and long values for each site, along with some other information. There are sometimes multiple rows for the same site, which has to do with keeping track of when we downloaded the file and what times to clip out of the data. So perhaps you could just get the unique row for each site to get the lat and longs?




```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 24 rows containing non-finite values (stat_smooth).
```




```
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)
```


Monthly trends in haddock landings and fishing effort
D6
D7
D8
D9
E0
E1
E2
E3
E4
E5
E6
54.0
53.0
52.0
51.0
36
35
34
33
32
31
Landings
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Effort
−12
−10
−8
−6
−4
Latitude
Longitude