Intro to R: Week 5

Topics Covered: Date/Time Data and Data Manipulation

Before we begin, you will need to install several packages. Please do so before class to make sure that everything is working properly.

```
install.packages("lubridate")
install.packages("reshape2")
install.packages("dplyr")
install.packages("ggplot2")
```

Part 1: Dealing with Date/Time Data

Task 1: Introducing the `lubridate` package

Step 1.1 Load the lubridate library. Use `?lubridate` to see what this package has to offer.

Step 1.2 What is the current system time? What is the current time in London? Hint: use `now()` and `with_tz()`.

Step 1.3 On what day of the week were you born? Hint: use `ymd()` and `wday()`. On what day of the week was your 21st birthday? Hint: use `years()`.

Task 2: Importing, formatting, and binning data from the SIO Pier

Step 2.1 Read in the shore stations data from the SIO pier, plot the temperature measurements, then add a column with a POSIX timestamp. Hint: use `paste()` and `ymd_hms()`.

Step 2.2 Use `plot()` or `ggplot()` to plot time v. temperature for the SIO pier record.

Step 2.3 These measurements were taken every six minutes, and they're noisy. Write a function that uses `interval()` and `within()` to bin data by a specified number of hours. Hint: the structure of your function will be very similar to the binning function we created last week for CTD data.

Step 2.4 Now calculate the average temperature in 24 hour bins and plot the day of the year vs. temperature using plot or ggplot2. Hint: use `yday()`.

Part 2: Data Manipulation

There are whole books written about data manipulation (also called data wrangling) in R, so this will be a very very brief overview to introduce you to a few of the packages and functions available.

Task 3: Handy data manipulation functions in base R

Step 3.1 Bin your SIO pier data by 1 hr, then use the `apply()` and `fivenum()` function to return the minimum, first quartile, median, third quartile, and maximum values of average temperature, average chlorophyll, and average salinity from your binned SIO pier dataset.

This is a very minimal example of the `apply` family of functions. Basically, anything that you can do with a `for` loop you can also do with `apply`.

Step 3.2 Use `table()` to determine how many measurements were collected in each month of the SIO pier record.

Task 4: Using the `reshape2` package

Note Many of the functions in the `reshape2` package have analogs in the `dplyr` package. I'm more familiar with the `reshape2` package so that's what I'll present here, but if you're interested in data wrangling definitely check out `dplyr`!

Step 4.1 Use the `melt()` and `dcast()` functions to generate a data frame of average temperature, salinity, and chlorophyll by month using your binned one hour data frame. This should not take more than two lines of code. *Bonus:* Use ggplot to create a three-panel timeseries plot of temperature, chlorophyll, and salinity.

Step 4.2 Read in the islands dataset and create a data frame of average percent hard corals, soft corals, etc. by island. Again, manipulating the data should not take more than two lines of code. *Bonus* Use `ggplot2` to create pie charts of average cover on each island.

Task 5: Regular Expressions and Partial String Matching

Step 5.1 Load the file SampleIndexes.csv and extract the samples which come from the cruise with identifier 1311COFI. Hint: use `grep()`.

Step 5.2 Which of the CalCOFI sample indexes begin with the sequence ATC?

Step 5.3 Were any samples labeled with indexes GCCGCG, GGGCCA, or CATTTT? Hint: use `%in%` or `match()`.