

Lab 2: Data analysis in R: Oceanographic section

ES 383 September 2017

This lab focuses on processing and plotting data and some data analysis. The tools you develop here can be used to analyze and plot the cruise data throughout the semester, as well as your independent research data. Be sure to save any scripts that you develop so that you can continue to use them.

You will use files from your first sampling trip along the Damariscotta Estuary. The data set used here is in the following files, corresponding to the CTD data from the four sampled stations:

```
20170912Colbysta1.csv  
20170912Colbysta2.csv  
20170912Colbysta3.csv  
20170912Colbysta4.csv
```

You'll use these scripts:

```
TempContour.R  
TSdiagram.R
```

Place all of those files in the folder that you'll be using in the RStudio server.

You will need to install packages in R. After you've opened R, type these lines:

```
install.packages('fields')  
install.packages('plot3D')  
install.packages('marelac')
```

I've also posted additional R resources in the moodle site.

After covering the material below, we will address the following statements/questions using this data set:

- 1- Make section plots of each of the variables measured by the CTD. Use a logarithmic scale where appropriate. Describe how the structure of the water column changes moving from in the river out to sea. Explain any patterns that you see.
- 2- Which measured variables are significantly correlated with each other (i.e. $p < 0.05$)? Explain the strongest correlations.
- 3- Which stations have a statistically significant difference between the surface and bottom temperatures? (Use the top and bottom 10 m.)
- 4- As we go through the semester, we'll continue to plot our data and revisit these questions.

For-loops

For-loops can save repetition. If you're doing something over and over again, often times a for-loop can save a lot of work. Here is an example using the CTD data from the previous lab:

```
CTD <- read.csv('Lab01example.csv')

for (i in 1:10)                # The variable i will loop through values 1 to 10
{                               # Whatever is inside {} will run for each i
  plot(CTD[,i],-CTD[,1],
       xlab=colnames(CTD)[i],   # This references column names in CTD
       ylab='Depth (m)',
       type='l',col='blue')     # Plotting command: note where i appears
  cat("press a key to continue") # Displays a message to the screen
  blah <- readline()            # Pauses between plots (press a key)
}
```

Note: You can paste this into an R-script and run it.

Here is the same for-loop, expanded to save each plot:

```
for (i in 1:10)
{
  plot(CTD[,i],-CTD[,1],
       xlab=colnames(CTD)[i],
       ylab='Depth (m)',
       type='l',col='blue')
  cat("press a key to continue")
  blah <- readline()
  fname <- paste('Lab2ExamplePlot',toString(i),'.png',sep='')
  dev.copy(png,fname)
  dev.off()
}
```

You can also put for-loops nested inside each other:

```
for (i in 1:10)
{
  for (j in 1:10)
  {
    plot(CTD[,i],CTD[,j],
         xlab=colnames(CTD)[i],
         ylab=colnames(CTD)[j])
    cat("press a key to continue")
    blah <- readline()
  }
}
```

Note: Try this command: `pairs(CTD)`

Basic Statistics

Comparing averages

Example: Is the average of the first 50 temperature measurements in the profile significantly different from the average of the next 50 temperature measurements?

```
t.test(CTD[1:50,2],CTD[51:100,2])  
wilcox.test(CTD[1:50,2],CTD[51:100,2])
```

The output p-value gives the statistical significance. The p-value is defined as the probability of obtaining a result equal to or "more extreme" than what was actually observed, when the null hypothesis is true. The Wilcox test is for when the data are not normally distributed.

If you type:

```
tempstats <- t.test(CTD[1:50,2],CTD[51:100,2])
```

Then `tempstats` is a list that contains all the information in the test. To get the p-value:

```
tempstats$p.value
```

Correlation

Example: Is light correlated with chlorophyll?

```
LightChl <- cor.test(CTD[,5],CTD[,6])  
LightChl$estimate      # gives the correlation estimate  
LightChl$p.value       # gives the p-value
```

Note: a significant correlation only suggests that the relationship is there. There may be better relationships. For example:

```
cor.test(log10(CTD[,5]),log10(CTD[,6]))
```

Relationships might also be nonlinear, and correlation does not imply causation.

T-S Diagram

I've written a short R function that plots a T-S diagram from your CTD data. To use the function, you'll need to type:

```
source('TSdiagram.R')
```

Then you can use the function to create a T-S diagram like this:

```
TSdiagram(CTD[,2],CTD[,3])
```

You can use a for-loop to plot the four profiles:

```
for (i in 1:4) {  
  CTD <- read.csv(paste('20170912Colbysta',toString(i),'.csv',sep=""))  
  TSdiagram(CTD[,2],CTD[,3],xlim=c(31,33),ylim=c(5,15))  
  cat("press a key to continue")  
  blah <- readline()  
}
```

Challenge: work out how to plot all four profiles on the same T-S diagram.

Section plot

This is the code from the included script TempContour.R

It makes a section plot of your cruise from last week. We will walk through this in lab.

```
library('fields')
lat <- c(43+54.130/60,43+51.8/60,43+48.56/60,43+44.87/60)
TEM <- matrix(NA,nrow=100,ncol=4)
for (i in 1:4)
{
  CTD <- read.csv(paste('20170912Colbysta',toString(i),'.csv',sep=""))
  for (j in 1:100)
  {
    TEM[j,i] <- mean(CTD[ceiling(CTD[,1])==j,2])
  }
}
image.plot(x=lat[4:1],y=-100:-1,t(TEM[100:1,4:1]),
           xlim=c(8,20),
           col=tim.colors(12),
           xlab='latitude (deg N)',
           ylab='Depth (m)',
           breaks=8:20,las=1)
```

There are multiple commands for doing color plots like this. Here is another option for section plots using `filled.contour`.

```
TEM2 <- TEM
TEM2[is.na(TEM)] <- -1
filled.contour(x=lat[4:1],y=-100:-1,t(TEM2[100:1,4:1]),
              levels=seq(8,20),
              color=function(x) rev(rainbow(x)))
```

- Contour plotting tries to interpolate the data so that the image is smoother. It doesn't work as well with our data because there are only four profiles with very different depths, so the results are a little choppy.