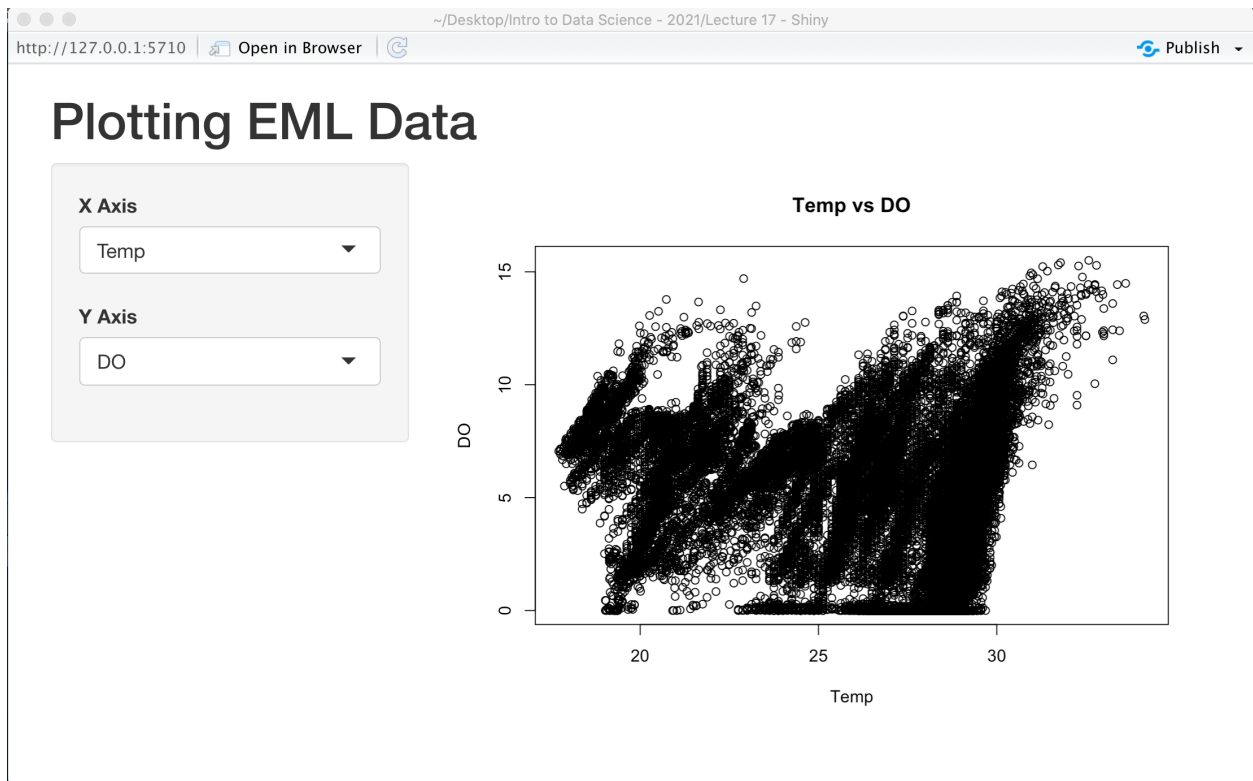# Lecture 16: In Class Exercise

Creating A Shiny App

For today's class we will be creating a Shiny app that uses our 'EML' data set to build a collection of plots. This interactive app will take in two different variables from the viewer and will display the corresponding visualization in real time. The final product will look similar to this:

1. We will build this app incrementally using the template provided on the current lecture page ('ShinyTemplate.R'). First, we need to load the `mowater` library and bring in the EML data set. (If you do not have the EML libray, you can load the `eml.rda` file in the `dat/` directory.)

What are the R commands to execute this? These should be placed after the `library(shiny)` command and before the `ui` interface.

2. Next, let's set the title for the app. What is the code that will set the title to be "Plotting EML Data" in the user interface?

3. Using the template, insert some text into the *Shiny app* that will serve as place holders for the elements of our final product. If we look at our final product image (above), we see that there are two elements in the side bar and one plot in the main window. Insert code within your shiny app that corresponds to each of these elements. When you run your app, it should look like this:

http://127.0.0.1:5710   Open in Browser

Publish

User Input #1 - Pull Down

User Input #2 - Pull Down

Main Plot

Inside of the **ui**, what code accomplished this task?

4. Now, start filling in the appropriate pieces. First we need to add control widgets that will allow you to select the variables on the x-axis and the y-axis. For a list of all of the control widgets, you can refer to this webpage, which includes sample code. In this case, choose the 'Select Box' option. The arguments for the `Select Box` widget should include the following:

- **inputId**: The "tag" or label that will be used for this element in the `server` part of the app.
- **label**: The label that will appear for this element directly on the app.
- **choices**: The choices, or list of values to select from.
- **selected**: The default choice to show in the app initially.

Repeat this process for the second 'Select Box', but make the default value a different one then in the first box. What is your updated **ui** code?

5. Because we are creating scatterplots of the different variables, for simplicity, remove some of the variables that don't make sense to plot. So instead of supplying all of the variables names to the `choices` argument, we need to remove some of the options (i.e., Date.Time and Depth). What is the updated code?

6. Finally, we need to add to the server code to create the plot based on the input lables in the 'Select Boxes' widget. To accomplish this, we need to create an output we will call 'plot1' and call the *renderplot()* function. It is within *renderplot()* that we will create our plot call, specifying the x and y as the values from the x and y select boxes. Once completed, make sure to update your **ui** to show the plot in the main panel using the call **plotOutput('plot1')**.

Going Further: If you complete everything, and your group still has time, other suggested additions that you can make are to

- have both the main title and the x/y labels update with the current plotting variables
- put alpha blending on the points
- change the plotting character
- add another widget to select the depth at which the x and y variables are plotted

Submission

For this in class exercise you will be required to submit your *.R code for your shiny app. If you separated out the server and UI code, please merge these into one file before submission.