Day 1 Worksheet

Some basic functions in R

Welcome to Big Data in Biology. Throughout this class we are going to learn how to manipulate large datasets and how to display data with graphs. Today we will learn some basic functions of R and then we will work through a script I have written to analyze the sediment and nutrient data from Kaneohe Bay found on Oahu, Hawai‘i. This is the largest marine bay in the Hawai‘ian islands and was the location where I did a lot of research on how coral reefs are responding to human pollution.

1. R is basically a number crunching program and we are going to learn some simple tasks to manipulate numbers and to create graphs from those numbers. First let’s just learn some of the features of R, and then we will dig into analyzing some data.
2. We will be using the program R studio, this allows us to see multiple panels that all contain different information all at the same time.
	1. Top left is the script panel. This is where you type your script. You can think of this as your note pad, it is where you want to write what you are doing, but R doesn’t actually do it until you say it is ready to run.
	2. Bottom left is the console. This is where R is doing the number crunching. What happens in the console is what is actually going on. However, the console is active but what happens there is not saved, it will always show the last action that you did.
	3. Top right has multiple tabs. We will use the environment tab the most, this shows you what is active in your R session. For us this is important because it shows the datasets that we have built.
	4. Bottom right also has multiple tabs. Here you can look at files, look at the packages you have installed and there is a pane that shows you the graphs you make. In the Files tab you can see the files that are in the directory where you are working. You can move among directories, and you can also use the gear symbol to set the working directory.
3. We will be working in the script pane, the top left.
4. First let’s learn to do some basic math. If you type some script R does not do that action. You need to tell R to run the script you have written. You can click the run button, or you can press ***command and return*** at the beginning of each line you want to run. The results of running your script will show up in the console (bottom left pane).
5. What is 768 \* 9234=
6. What is 392 / 21=
7. Sometimes we want to save a number for further manipulation. We can do this by assigning the number to a variable. Let’s make x = 99. In R you could type = but this has multiple functions. Instead, I want you to use the assignment symbol <-. In R the keyboard short cut for this symbol is ***option –***
8. Now let’s make y <- 4. What is x / y=
9. We can also make a variable equal to a column of numbers. To make a column we use ***c()*** Lets make z <- c(7,8,9) and v <- c(1,2,3). What is z \* v =
10. What is z \* y=
11. To make a table with more than one column we can create a data frame using the function data.frame (). First you need to define the variables.
12. Create a column called age and assign it the numbers 23, 24, 25.
13. Then create a column called name and assign it the names peter, paul, mary. When you are entering words into R you need to put them in quotes, in this case your name <- c(“peter”, “paul”, “mary”)
14. Now put age and name together in a data frame called one. The command for creating a table like this is ***data.frame ()***. Notice that the age associated with the names depends on the order of the variables that you input to create your original columns.
15. If you want to get information from a column in your data frame it is pretty easy, you can just use the $ to indicate the column you want. In our case we might want the maximum age in the dataset so we would type max (one$age).
16. If we want to add a new column of all the same numbers we can create a new variable, assign it a value, and then call out a new column. Let’s add the same zipcode to each name, 94018. First define zipcode as 94018, then name a new column (one$zip) and set it equal to zipcode. You will see that R automatically adds the column and a number for each row.
17. Now let’s get some basic information from our dataset. What are the column names in this dataset? We can use the command ***head ()*** to look at the headers for each column. Once we know the name of the columns we can call them out in the functions below.
18. What is the maximum age? Use the function ***max()*** to get your answer
19. What is the mean age? Make sure you call out the whole column. Use the function ***mean ()*** to get your answer.
20. What is the minimum age? Use the function ***min ()*** to get your answer.
21. In statistics we are often conserved with a central tendency (mean, median or mode) and the variance around that central value. To get a sense of how much distance there is between the individual data points and the mean you can measure the standard deviation, which is an important measure of variance in every dataset. To get the standard deviation in R you can use the function ***sd ()***
22. Often we want to get a summary of the central tendencies of a column from a data set. Use the function ***summary ()*** to get a summary of the age column.
23. Great job. These are the basic functions in R. There are many other math functions that r can do. A big part of learning to use R is searching for information on the internet. Search for calculating variance in R. Is there another math manipulation you want to try, google it and see what you can find.
24. Hopefully you can see that there is a lot of power in this program to create and manipulate datasets. The data frame that we created was quite small, only 3 rows, so now we can take a look at a dataset that I created with more variables and more samples.

­­­­­

An Example Dataset- Sedimentation Rates (Kbay\_sediment\_2015)

Getting data into R. This can happen by us typing the numbers as we did above, or datasets can be imported into R to be worked on. In general, when you are working on big data you will be importing the datasets into R. For this exercise you can retrieve the dataset and R-script from my website under day 1. It is called Kbay\_sediment\_2015.csv and is in a .csv format which is a basic table format (comma separated values) you can create using Microsoft Excel. The r-script is called rrw\_sed.r

1. Make a folder on the desktop of your computer called big\_data. Download the .csv and the .r file and place them in the big data folder. Download the r script too, today you can just follow the script I’ve built. For the rest of the class you will be writing your own scripts so make sure you ask if you have any questions about what I’ve done.
2. Open the .r file you just downloaded. You should see all 4 panes on your R studio.
3. Now we are going to tell R where to look for the dataset we are going to analyze. You want to use the function ***setwd ()*** =set working directory. This is a critical function that tells R where to find the data it needs to do its analysis. In our case we are going to set it to be our folder big\_data. Your function should look like setwd (“~/Desktop/big\_data”)
4. Now we need to import the data frame. You can call it what every you want, I called my sed.all to indicate it is the sediment data for all of the reefs. You should assign the .csv file to your data frame using the ***read.csv ()*** function. When you read csv, you must put the name of the file in quotation marks and you must spell out the entire name of the file. R is very literal, every letter (capital or lowercase) and symbol must exactly match what is shown.
5. In this case we are going to look at a dataset called Kbay\_sediment\_2015, which is sedimentation rates measured every month for 2015 in Kaneohe Bay, Oahu, HI. I measured the dry weight of sediments every month and produced a simple dataset of date (every month in 2015) , reef (9 different reef sites found in the bay), and the amount of sediment\_grams\_per\_day. To import the dataset make sure it is in the working directory and then run read.csv (“Kbay\_sediment\_2015.csv”)
6. Now in the right top pane in the environment tab you should be able to see a file called sed.all (or whatever you called yours). Click on that. Now you can see the data frame for our dataset.
7. If you just want to know what the column names are you can also use the header function, ***head ()*** it will tell you the title of each column. This is especially important if you want to call out a specific column for manipulation or summary.
8. This dataset isn’t too big but we will be working with datasets that are 1,000’s of rows long. A great way to rapidly see the properties of a data set is using the structure function, ***str ()***. Try that on our sed.all dataset and see what information we get.
9. When you look at the structure of a data set R will tell you how it sees your data. In this case the columns of date and site are factors, and sediment is a numerical format. That is fine but R is really good with dates, and we want to tell R that the date column is actually a date so it can take advantage of using those features. The function to convert a column to a date format is ***as.Date ()***. Run this function on our data set and now look at the structure of the data set. It should show up as a Date now. We’ll use this function again, but the format takes a little getting used to.
10. Often when you are working with a large data set you might want to target a specific group of data. To reduce the data set to what you are interested in we are going to use the function ***subset ()***. In the parenthesis of subset you need to indicate the data set (sed.all) and then define which data to include. In these 3 lines I have created a different subset for 3 different sites. You can see in the environment that we now have 3 more datasets, one for each reef. Now you can get a summary of the sediment data in each of these 3 reefs for 2015. Subset is one of the most important functions we will learn during this class.
11. Now we can look at graphing in base R. Plot is the function most often used for graphing, but it is a little clunky. Here is some script for making a line graph that compares the sedimentation rates at 3 of the reefs in Kaneohe Bay.
12. Often I save my graphs as .pdf files so they can be used in other programs. There are two lines before the plot function and 1 line under that I have turned off with the “#” sign. Do you see how you can use this feature to make notes as you are writing your r script. Remove these number signs. On your desktop you should have made a file called, Sed\_fig1.pdf.
13. Great job today! You’ve learned a ton of features about how to analyze and graph data in R. Tomorrow we are going to focus on using the graphing package ggplot. More on that tomorrow, however if you have finished early you can continue on to the bonus activity below.

Bonus Dataset-Nutrients in Kaneohe Bay (Kbay\_nutrient\_data\_2014\_2016.csv)

I have made another dataset that you can use to practice your r skills. This dataset has data from 9 reefs in Kaneohe Bay, but now has values for different nutrients that were dissolved in the seawater. These data span multiple time points from 2014 to 2016.

1. Use the functions from our analysis above to import the nutrient dataset into R. I’ve put a copy of this .csv file on my website under outreach Day1-BigData. Kbay\_nutrient\_data\_2014\_2016.csv
2. Answer the following questions by using the functions that we have already used for the sediment dataset above.
3. What was the highest, mean, lowest value of phosphate? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. What was the highest, mean, lowest value of ammonia? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. What was the mean ± standard deviation for silicate? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. At reef 44 what was the highest value for ammonia? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. At reef 25 what was the highest value for Nitrate + Nitrite? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. For both reef 42 and 44 what was the total silicate over the entire year?
9. Make a line graph of phosphate (y-axis) and date (x-axis) with a different color line for reefs 44, 42, 25, HIMB.
10. Make the same graph with all 4 reefs but for Silicate, Nitrate + Nitrite, and Ammonia.
11. Great work! See you tomorrow to work on using ggplot to create our graphs!