

BIOS 4120: Introduction to Biostatistics
Breheny

Lab #1

For the first day of lab we will introduce the programming language **R**, which will be used throughout the course. Computers are powerful tools for statisticians and it will be helpful for you to be familiar with the basics of programming. **R** can be used to do simple calculations, create plots and figures, and to run statistical analyses. We will provide you with all the programming you need to know for this class in lab; however, we cannot cover everything that **R** can do. If you have any questions or are curious about other aspects of **R**, there is a lot of material online that is worth reading.

1 Downloading and Installing R

We will be using a version of **R** called RStudio. On the lab computers, you can just open up RStudio. However, to get it on your personal computer it is a two-step process.

- 1) To install **R** on your personal computer, go to <http://cran.r-project.org>
- 2) Then install RStudio on your personal computer, go to <http://www.rstudio.com>

Note: You have to install **R** first then install RStudio. This is very important! The instructions are pretty clear on the websites; if you need help please contact a TA to assist you after Lab or during office hours

2 Using R as a calculator

Adding, subtracting, multiplication, division, power (+, -, *, /, ^)

```
> 4 + 5 - (24/6)
[1] 5
> (6 - 4) * 3
[1] 6
> 5 ^ 2
[1] 25
```

Sequences

```
> 1:8
[1] 1 2 3 4 5 6 7 8
> seq(1,10,2)
[1] 1 3 5 7 9
```

Storing scalars and vectors: You can name numeric or character variables any way you want. Spaces are not allowed in naming of variables. It is recommended to only use letters (uppercase or lowercase) and underscores when naming a variable. R is case sensitive, which means capital letters are different from lower case letters. Please also use the assignment operator (<-) instead of an equals sign. This is standard programming practice.

Note: Refrain from using function names as the name for your variable. For example, the letters c and t are functions in R. If you are not sure if a letter or word is a function, just type it in the console and hit enter. If it is a function name you should see an output saying its a function. If it is not a function name it will result an error.

```
> t
function (x)
UseMethod("t")
<bytecode: 0x000000000820d508>
<environment: namespace:base>

> x <- 1:8
> x
[1] 1 2 3 4 5 6 7 8
> x <- 6
> x + 3
[1] 9
> x <- x + 10
> x
[1] 16
```

Common functions: mean(), sd(), min(), max(), sqrt(), exp(), log(), sum()
Mean, standard deviation, minimum, maximum, square root, exponential function, logarithm, summation, respectively. There are many more functions available in R but we will learn the necessary functions as we progress through the course.

```
> x <- 1:9
> mean(x)
[1] 5
> sum(x)
[1] 45
> sqrt(9)
[1] 3
> min(x)
[1] 1
> max(x)
[1] 9
> max_x <- max(x)
> max_x
[1] 9
```

Indexing: Lets say we have a vector, x , and it contains the values 2, 3, and 5. We want to extract the second position of vector x and store it in a new variable called b .

```
> x <- c(2,3,5)
> b <- x[2]
> b
[1] 3
```

The square brackets after a vector variable indicate we want to extract a certain position or positions of x . In this case, we extract the second position in x (which is 3) and store it in a new variable called b .

Comments: Often in programming languages, you can provide comments within code that explains what certain code does or if you just want to leave notes for yourself. In R, to start a comment it is `#` and everything on the same line immediately to the left of `#` is commented out.

```
> # Example of a comment. This line does NOT get run by the program
```

In the R Script, comments are green. Remember, the `#` starts a comment ONLY on the same line.

Help function: The help function is very useful in R. If you are not sure what a function does, type a question mark directly followed by the name of the function. The help page for that function will be brought up in the bottom right of RStudio. Additionally, there are lots of help pages and documentations on the internet for R and its functions.

```
> ?max
```

Recommendations: Create a folder for this class in your H drive. When lab begins, open up a new R Script and save it as Lab (lab number). Follow along with the coding by your instructor and use comments to make notes of what is taught. When the lab session is over, save your R Script so it can be referenced to later in the semester, if needed. Please also space out characters and numbers to make your code more readable!

3 Reading in Data

There are many different ways to read in data of all formats into R. In this class we will only use data that is saved as 'tab-delimited'. You can find all the datasets we will use in class at <http://myweb.uiowa.edu/pbreheny/4120/s15/data.html>. The following code is an example of importing data into R.

```
> titanic <- read.table("http://myweb.uiowa.edu/pbreheny/data/titanic.txt", header = TRUE)
```

The `'header = TRUE'` argument tells **R** that the first row of variables are the headers. Note that the directory of the data has to be in quotations. You can now either click on your dataset in the top right corner of RStudio to view the data, or use:

```
> View(titanic)
```

Please note that the `'V'` on `'View'` is capitalized.

This concludes Lab 1 ■