**R Objects – 6 Basic Types of Atomic Vectors**

* doubles – stores regular numbers (positive or negative, decimals or none). When you assign a number to an object, this is the default.

Ruler <- c(1:12)

* integers – numbers without a decimal component. This are specified when setting an object by using the capital letter L next to the number.

Int\_example <- c(-1L,1L,2L,3L)

* characters – stores a small amount of text. These are specified by using quotes around the word(s). As long as there are quotes around, a number may be stored as a character (e.g., “1” is actually a character and not a double)

text <- c(“hello”, “world”)

* logicals – stores TRUE or FALSE (in R, these are Boolean data). Anytime you type TRUE or FALSE in capital letters.

logic <- c(TRUE, TRUE, FALSE)

* complex – stores complex numbers. Anytime an imaginary term is added to a number, the vector will read this as a complex object. (NOT COMMON USED FOR DATA SCIENCE)

comp <- c(1 + 2x, 3 + 4y, 5 + 6z)

* raw – stores raw bytes of data (NOT COMMON USED FOR DATA SCIENCE)

**For most users, you will not be entering your own data directly into R (in the next module, we will discuss how to import data). However, I do want you to understand how things work within R**

**Vectors**:

* Although we have not named them yet, you have worked with atomic vectors within R. These are just a simple vector of data that you have assigned to an object. For example, when you used the command *ruler <- c(1:12)*, you created a vector of information that includes the numbers 1,2,3,4,5,6,7,8,9,10,11,12.
* The problem with working with vectors is each atomic vector can contain only one type of data. For example:

*example1 <- c(1, 2, “hello”, “world”)*

will result in everything stored in the vector *example1* will be considered characters. To overcome this limitation, we create data frames.

**Data Frames**:

* A data frame is a two dimensional version of a list where each column within the data frame can contain a different type of data (although the within a column must still be the same). Think about working with an Excel spreadsheet. These are much better to work with when doing data analysis. Let’s create a simple data frame that will store UNCW Soccer Game Schedule:

fall\_2018 <- data.frame(Date=c("9/10","10/2","10/15"), Opponent=c("Northeastern","Hofstra","Elon"),UNCW\_Goals=c(2,4,1),Opp\_Goal=c(1,2,0))