

01 Introduction to R

Statistics

To access R and R Studio you can go to: rstudio.cloud And log in through your Saint Ann's gmail After you log in, you will see four panes. For now, we will focus our attention on the console window in the lower left. Down the line you will likely find yourself spending most of your time writing scripts in the upper left pane.

Computation

R can be used for computation, data manipulation, visualizations and simulations (among other things) but, mostly simply, R can be used as a calculator. Try typing the following into the console. [Note: As you work through this lab, take the time to make sure that you understand what each line of code is doing and try typing out your own lines of code using the same functions.]

```
3*2
4^2
5/2
5 %/% 2
5 %% 2
5*(3+7)^2
sqrt(25)
25^(1/2)
```

How are /, %/% and %% different from each other?

R can also perform operations on vectors or matrices. Let's try creating vectors, assigning them to variables and then performing calculations on them.

```
x <- 1:10
x
x^2

y <- seq(from=10,to=100,by=5)
y
1/y

z <- c(2,3,5)
z
10*z
```

Use R to calculate:

- The first 10 multiples of 9

- The first 20 perfect squares.

Scientific Notation

Scientific notation allows us to more readily understand and perform calculation on really large and really small numbers. You can enter numbers into R in scientific notation and R will automatically produce large and small number in scientific notation. Try entering each of the following:

```
1000000000000  
0.0000000000000565  
6.2e9  
5e-10  
6.2e9*5e-10  
factorial(15)
```

Functions

R has many built in functions and many more that can be added as parts of packages. To find out how a function works you can type “?” and then the name of the function. This will pull up instructions within the help tab in your lower right window. The most useful parts of these instructions are often the examples given at the bottom. Try running each of the following lines one at a time and

```
x <- 2:10  
x  
  
sum(x)  
  
?sum  
  
min(x)  
max(x)  
length(x)  
mean(x)  
cumsum(x)  
cumprod(x)
```

Challenge

If you place one grain of rice on the first space on a chessboard and two on the second space and four on the third and so on (doubling every time), how many grains of rice will you need to fill the chessboard (which has 64 spaces)? [Try writing one line of R code that will solve this problem.]

Logical Statements

You can also enter logical statements into R – that is, statements that will return true or false. You can then do mathematics on these values with true taking the value 1 and false the value 0. Try entering each of the following statements in turn and make sure that you understand why you are getting each output.

```
x <- c(20,25,23,27, 28)
x > 23
x >= 23
x == 20
7*(x >= 23)+3
sum(x>=23)
x %% 5
(x %% 5) == 0
sum((x %% 5) == 0)
```

You can make your logical statements more complicated by introducing ands and ors (& and | in R).

```
x <- c(20,25,23,27, 28)
(x == 20) & (x == 28)
(x == 20) | (x == 28)

(x %% 5 ==0 ) | (x %% 7 ==0)

sum((x %% 5 ==0) | (x %% 7 ==0))
```

Going back to the rice and chessboard scenario, if you have one million grains of rice how many spaces can you fill before running out? [Try to write one line of code to solve this problem.]

Writing your own Functions

You can write your own functions in R. Here are three examples of code to create functions (each followed by two examples of the function in use).

```
Add_Them <- function(x, y){x+y}

#examples
Add_Them(5,7)
Add_Them(10,20)

Mean <- function(x){sum(x)/length(x)}

#examples
Mean(2:10)
Mean(c(20,25,23,27,28))

#examples
Differences <- function(x, y){x - y}
guesses <- c(20,25,23,27,28)
values <- c(21,23,24,25,25)
Differences(guesses, values)
```

Challenge:

Write a function that computes the root mean square error of a series of guesses given the guesses and the true values. The following code may help you get started. Note that the root mean square error should be 1.949359 given the values provided below.

```
guesses <- c(20,25,23,27, 28)
values <- c(21, 23, 24, 25, 25)

RMSE <- function(x,y){
  #formula here
}

RMSE(guesses, values)
```