Note rolling a dice

w.Lu

13/01/2020

Project 1: Weighted dice

we assume to rolling 2 dice in one time .

```
Roll1 <- function(){
  die <- 1:6 # we store a vector called "die" with numbers 1 to 6
  dice <- sample(die, size = 2, replace = TRUE) # Sampling with replacement is an easy way to
create independent random samples.
  sum(dice)
}

# Now we call the fuction
Roll1()
```

```
## [1] 10
```

```
Roll1()
```

```
## [1] 4
```

Let’s try another time

```
Roll2 <- function(bones = 1:4) {
  dice <- sample(bones, size = 2, replace = TRUE)
  sum(dice) }
  Roll2()
```

```
## [1] 5
```

A third try

```
  Roll3 <- function(bones = 1:6) {
    dice <- sample(bones, size = 2, replace = TRUE)
    sum(dice)
  }

  x <- c(-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1)
  x
```

```
##  [1] -1.0 -0.8 -0.6 -0.4 -0.2  0.0  0.2  0.4  0.6  0.8  1.0
```

```
x <- c(-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1)
y <- x^3
y
```

```
##  [1] -1.000 -0.512 -0.216 -0.064 -0.008  0.000  0.008  0.064  0.216  0.512
## [11]  1.000
```

```
Roll3()
```

```
## [1] 3
```

```
Roll3()
```

```
## [1] 6
```

Summary

As you’ve seen, R is a language that you can use to talk to your computer. You write commands in R and run them at the command line for your computer to read. Your computer will sometimes talk back–for example, when you commit an error–but it usually just does what you ask and then displays the result. The two most important components of the R language are objects, which store data, and functions, which manipulate data. R also uses a host of operators like +, -, *, /, and <- to do basic tasks. As a data scientist, you will use R objects to store data in your computer’s memory, and you will use functions to automate tasks and do complicated calculations.