Class06 R Funtion

Xinyu Wen (PID: A17115443)

Table of contents

A first silly function 1
A second more fun function 2

Today we will get more exposure to functions in R. We call functions to do all our work and today we will learn how to write our own functions.

A first silly function

Note that arguments 2 and 3 have default values (because we set y=0 and z=0), so we don't have to supply them when we call our function.

```
add <- function(x, y=0, z=0) {x + y + z}
```

Can I just use this? Need to run the function first.

```
add(1, 1)
```

```
[1] 2
```

```
add(1, c(10, 100))
```

```
[1]  11 101
```

```
add(100)
```

```
[1] 100
```

```
add(100, 10, 1)
```

```
[1] 111
```

A second more fun function

Let's write a function that generates random nucleotide sequences.

We can make use of the inbuilt **sample()** function in R to help us here.

```
sample(x=1:10, size=9)
```

```
[1] 5 6 1 2 3 9 7 4 8
```

```
sample(x=1:10, size=11, replace=TRUE)
```

```
 [1] 10  8  7  4  1  2  7  6  8  5  6
```

> Q. Can you use `sample()` to generate a redom nucleotide sequence of length 5.

```
sample(x = c("A", "T", "C", "G"), size =5, replace = TRUE)
```

```
[1] "A" "A" "T" "T" "C"
```

> Q. Write a function `generate_dna()` that makes a nucleotide sequence of a user specified length.

Every function in R has at least three things:

- a **name** (in our case "generate_dna")
- one or more **input argument** (the length of sequence we want)
- a **body** (R code that does the work)

```
generate_dna <- function(length = 5) {
  bases <- c("A", "T", "C", "G")
  sample(bases, length, replace = TRUE)
  }
```

```
generate_dna()
```

```
[1] "C" "T" "C" "A" "T"
```

```
generate_dna(10)
```

```
 [1] "G" "A" "G" "A" "A" "A" "G" "T" "G" "C"
```

Q. Can you write a `generate_protein()` function that returns amnino acid sequence of a userrequested length?

```
aa <- bio3d::aa.table$aa1[1:20]
```

```
generate_protein <- function(length=5) {
  sample(aa, length, replace =T)
}
```

```
generate_protein(6)
```

```
[1] "L" "V" "M" "R" "E" "H"
```

I want my output of this functions to have no quotes, and rather be a continuous sting as opposed to individual elements.

```
bases <- c("A", "T", "C", "G")
paste(bases, collapse = "")
```

```
[1] "ATCG"
```

```
generate_protein <- function(length=5) {
  s <- sample(aa, size=length, replace=T)
  paste(s, collapse="")
}
generate_protein(6)
```

```
[1] "STSIDS"
```

Q. generate protein sequences from length 6 to 12?

We can use the useful utility function `sapply()` to help us "apply" our function over all the values 6 to 12.

```r
ans <- sapply(6:12, generate_protein)
```

```r
cat(paste(">ID", 6:12, sep ="", "\n", ans, "\n"))
```

```
>ID6
TECVII
 >ID7
HDQWIFA
 >ID8
VGPHLMQM
 >ID9
MISDGPREF
 >ID10
HWSYTMFGQH
 >ID11
HPMNWFWTDWG
 >ID12
LDFWATQFFEEY
```

> Q. Are any of these sequences unique in nature - i.e. never found in nature. We can search "refseq-protein" and look for 100% Identity and 100% coverage?

The shorter sequences (ID6:8) have 100% identity and coverage, although with high E value. Howevery, longer sequences (ID9:12) do not have 100% identity nor 100% coverage. Thus the longer sequences are unique in nature.