CLASS 6 R FUNCTIONS!

Karolina A19106745

Table of contents

Background

All functions in R have at least 3 things:

- A **name** that we use to call the function
- One or more input **arguments**
- The **body** the lines of R code that do the work

Our first function

Let's write a silly wee function to `add` some numbers(the input arguments)

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

Now we can use this function

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

```
[1] 101
```

```
add(x=c(100, 1, 100), y=1)
```

```
[1] 101   2 101
```

Q. What if I give a multiple element vector to `x1 andy`‘

```
add(x=c(100, 1), y=c(100, 1))
```

```
[1] 200   2
```

Q. What if I give three inputs to the function?

```
#add(x=c(100, 1), y=1, z=1)
```

Q. What if I give only one input to the add function?

```
addnew<- function(x, y=1){
x + y
}
```

```
addnew(x=100)
```

```
[1] 101
```

```
addnew(c(100,1), 100)
```

```
[1] 200 101
```

If we write our function with input arguments having default value then the user does not have to use a user specified value.

A second function

Let’s try something more interesting: Make a sequence generating tool..

The `sample()` function can be a useful starting point here:

```
sample(1:10, size=4)
```

```
[1] 8 1 5 6
```

Q. Generate 9 random numbers taken from the input vector x=1:10?

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

```
[1]  3  5  1  7  8 10  2  4  9
```

Q. Generate 12 random numbers taken from the input vector x=1:10?

```
#sample(1:10, size=12)
```

```
sample(1:10, size=12, replace=TRUE)
```

```
 [1]  8  8  9  7  7  7  3 10  3  5  2  8
```

Q. Write code for the `sample()` function that generates nucleotide sequences of length 6?

```
sample(x=c("A", "G", "C", "T"), size=6, replace=TRUE)
```

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

Q. Write a first function `generate_dna()` that returns a user specified length DNA sequence:

```
generate_dna <- function(length) {

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

}
```

```
generate_dna(12)
```

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

Key-Points Every function in R looks fundamentally the same in terms of its structure. Basically 3 things: name, input, and body

```
name <- function(input){

  body
}
```

Functions can have multiple inputs. These can be **required** or **optional** arguments. With optional arguments having a set default value.

Q. Modify and improve our `generate_dna()` function to return it's generated sequence in a more standard formt like "AGTAGTA" rather than the vector "A", "C", "G", "A"

```
generate_dna <- function(length=6, fasta=TRUE) {

ans <- sample(x=c("A", "G", "C", "T"), size=length, replace=TRUE)


if(fasta) {
  cat("Single-element vector output")
  ans <- paste(ans, collapse = "")

} else {

  cat("Multi-element vector output")
}

  return(ans)
}

generate_dna()
```

```
Single-element vector output

[1] "ATGCCA"
```

The `paste()` function - it's job is to join up or stick together (a.k.a paste) input strings together

```
paste("alice", "loves R", sep=" ")
```

```
[1] "alice loves R"
```

Flow control means where the R brain goes in your code

```
good_mood <- TRUE

if(good_mood) {

  cat("Great!")


} else {

  cat("Bummer!")

}
```

```
Great!
```

```
good_mood <- FALSE

if(good_mood) {

  cat("Great!")


} else {

  cat("Bummer!")

}
```

```
Bummer!
```

A Protein generating function

> Q. Write a function, called 'generate_protein()', that generates a user specified length protein sequence

There are 20 natural amino-acids

```
aa <- c("A", "R", "N", "D", "B", "C", "Q", "E", "G", "H", "I", "L", "K", "M", "F", "P", "S",
```

```
generate_protein <- function(length) {
  # The amino-acids to sample from
  aa <- c("A", "R", "N", "D", "B", "C", "Q", "E", "G", "H", "I", "L", "K", "M", "F", "P", "S

  # Draw n-length amino acids to make our sequence
 ans <-  sample(aa, size=length, replace=T)
  ans <- paste(ans, collapse = "")
}
```

```
myseq<- generate_protein(42)
myseq
```

```
[1] "BDDIIQARQGGRPCNSWMGPALPLVWFMHHRGHLTYFKNVKP"
```

Q. Use that function to generate random protein sequences between length 6 and 12

```
generate_protein(6)
generate_protein(7)
generate_protein(8)
generate_protein(9)
generate_protein(10)
generate_protein(11)
generate_protein(12)
```

```
for(i in 6:12) {

  # FASTA ID line ">id"

  cat(">", i, sep="", "\n")

  # Our protein sequence line
  cat(generate_protein(i), "\n")

}
```

```
>6
ATKAPQ
>7
RGBGEGR
>8
PSMSISST
>9
VWBIEWCPH
>10
AKNBMSSTLC
>11
ETWTREPPENB
>12
PLYFWKFTRWBT
```

Q. Are any of your sequences unique i.e not found anywhere in nature?

Yes! Amino acids 6 through 12.