class06

Maria Tavares (PID A69036242)

Table of contents

All functions in R have at least 3 things: - A **name**, we pick this and use it to call our function, - Input **arguments** (there can be multiple), - The **body** lines of R code that do the work. ##Maria's first R function Write a function to add some numbers:

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

Now we can call this function:

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

```
[1] 110
```

A second function

Write a function to generate random nucleotide sequences of a user specified length: The 'sample()' function can be helpful here.

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

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

I want the a 1 element long character vector that

```r
v <- sample (c("A", "C", "G", "T"), size = 50, replace = TRUE)
paste(v, collapse = "")
```

```
[1] "CTCACGCGTCGAAGACCCACAGCAACAGGAGTCATTTCCGCCCCCTCACG"
```

Turn this into my first wee function

```r
generate_dna <- function(size = 50) {
  v <- sample(c("A", "C", "G", "T"), size = size, replace = TRUE)
  paste (v, collapse = "")
}
```

Test it

```r
generate_dna(60)
```

```
[1] "CGGTGCGAAACCACACTACCTGCCGCATGAAAACCGATATATTAACGTTAGCTGATGTTC"
```

```r
fasta <- FALSE
if(fasta) {
  cat("HELLO You!")
}
```

Add the ability to return a multi-element f=vector or a single element fasta like vector.

```r
generate_fasta <- function(size = 50, fasta = TRUE) {
  v <- sample(c("A", "C", "G", "T"), size = size, replace = TRUE)
  s <- paste (v, collapse = "")

  if(fasta) {
    return (s)
  } else {
    return (v)
  }
}
```

```r
generate_fasta(10, fasta=TRUE)
```

```
[1] "TGCAACAGGG"
```

A protein generating function

```r
generate_protein <- function(size = 50, fasta = TRUE) {
  aa <- c("A", "R", "N", "D", "C", "Q", "E", "G", "H", "I",
          "L", "K", "M", "F", "P", "S", "T", "W", "Y", "V")
  v <- sample(aa, size = size, replace = TRUE)
  s <- paste(v, collapse = "")

  if(fasta) {
    return(s)
  } else {
    return(v)
  }
}
```

```r
generate_protein(6)
```

```
[1] "REVPQI"
```

Use our new `generate_protein()` function to make random protein sequences of lenght 6 to 12 (one length 6, one length 7…etc)

One way to do this is "brute force"

```r
generate_protein(6)
```

```
[1] "KWIPAL"
```

```r
generate_protein(7)
```

```
[1] "ELTTRAW"
```

```r
generate_protein(8)
```

```
[1] "VGWEQPNM"
```

A second way is to use a `for()` loop:

```r
lenghts <- 6:12
lenghts
```

```
[1]  6  7  8  9 10 11 12
```

```r
for (i in lenghts) {
  cat(">", i, "\n", sep = "")
  aa <- generate_protein(i)
  cat(aa)
  cat("\n")
}
```

```
>6
PLGVIM
>7
WSLFFNM
>8
AFEIIRRG
>9
TYDIQPHPA
>10
FIGEGCGSCT
>11
PGTHIVTQIWA
>12
CCHCMGLVIQAG
```

A third, and better way to solve this is the `apply()` family of functiond, specifically `sapply()` function in this case.

```r
sapply(6:12, generate_protein)
```

```
[1] "SYDDLS"       "FHKKDQV"      "PVPHNRPC"     "GQGVRIRIM"    "VVPNEWDKRV"  
[6] "IITIANLCKNS"  "CHRWIVDAWCYE"
```