

class 6: R functions

Shivani (PID: A12652928)

Today we are getting more exposure to R functions.

Let's start with a simple function to add some numbers:

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

can we use this function? only if you remember to send it to the “R brain” by clicking the green arrow in the code box

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

```
[1] 2
```

```
add(c(100,200), 1)
```

```
[1] 101 201
```

```
add(100)
```

```
[1] 100
```

```
log(10)
```

```
[1] 2.302585
```

```
log(10, base=10)
```

```
[1] 1
```

```
add(100, 1, 200)
```

```
[1] 301
```

'sample()' function

Q. what does it do?

The sample function randomly selects a sample of desired size from a vector.

```
sample(1:10, 1)
```

```
[1] 4
```

what if i want 11 things taken from my vector 1:10

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

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

side note:

```
seq(5, 50, by=3)
```

```
[1] 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50
```

generate DNA sequences

Q. write a function to make random nucleotide sequence of a user specified size/length.

```
x <- c("A", "C", "G", "T")
sample(x, size=9, replace = TRUE)
```

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

All functions in R have 3 things: - **name** (we pick this) - input **arguments** (length of the output sequence) - **body** (line by line work)

```
generate_dna <- function(length=9) {
  #the nucleotides to draw/sample from
  bases <- c("A", "C", "G", "T")
  #draw n=length nucleotides to make sequence
  ans <- sample(bases, size=length, replace = TRUE)
  #concatenate
  ans <- paste(ans, collapse="")
  # to explicitly print the result
  return(ans)
}
generate_dna(length=450)
```

```
[1] "CGCCTAGCATGGTATGGGCCTGACTTTATCGCGAGCGTCTAAGTTCCAGCGTGGAGAGGTGACTTTGTACAAGTCCGTTTATGCTAG"
```

I want the ability to switch between these 2 output formats, with an extra argument that controls the function with TRUE/FALSE.

```
generate_dna <- function(length=9, collapse=FALSE) {
  #the nucleotides to draw/sample from
  bases <- c("A", "C", "G", "T")
  #draw n=length nucleotides to make sequence
  ans <- sample(bases, size=length, replace = TRUE)
  #concatenate
  if(collapse) {
    ans <- paste(ans, collapse="")
  }
  # to explicitly print the result
```

```

    return(ans)
}
generate_dna(length=450)

```

```

[1] "T" "G" "G" "C" "A" "C" "G" "T" "G" "C" "G" "G" "A" "T" "A" "T" "G" "G"
[19] "A" "A" "A" "G" "C" "T" "G" "C" "G" "G" "A" "G" "T" "T" "C" "C" "A" "T"
[37] "T" "T" "G" "C" "G" "A" "C" "T" "C" "C" "C" "T" "A" "T" "G" "C" "T" "A"
[55] "A" "C" "C" "A" "A" "C" "T" "G" "A" "G" "C" "C" "T" "G" "C" "A" "T" "T"
[73] "G" "T" "T" "G" "A" "C" "G" "C" "G" "C" "C" "G" "C" "T" "G" "A" "G" "G"
[91] "T" "G" "T" "C" "T" "A" "G" "A" "G" "G" "G" "T" "A" "C" "C" "G" "G" "C"
[109] "A" "G" "T" "A" "A" "G" "C" "T" "A" "T" "A" "C" "C" "T" "A" "T" "G" "T"
[127] "A" "C" "A" "G" "G" "G" "A" "G" "G" "G" "G" "T" "A" "G" "G" "A" "T" "A"
[145] "C" "A" "A" "T" "C" "G" "A" "G" "C" "C" "T" "C" "C" "G" "C" "C" "G" "C"
[163] "C" "G" "A" "C" "A" "C" "T" "A" "C" "T" "G" "G" "C" "C" "A" "A" "G" "C"
[181] "G" "T" "A" "C" "A" "T" "G" "A" "A" "T" "C" "C" "A" "T" "A" "G" "G" "C"
[199] "T" "G" "T" "G" "C" "T" "G" "A" "G" "A" "T" "C" "C" "T" "A" "C" "T" "C"
[217] "G" "G" "C" "A" "C" "G" "G" "T" "A" "G" "T" "T" "A" "C" "G" "G" "G" "A"
[235] "A" "C" "G" "T" "C" "T" "A" "G" "T" "C" "G" "C" "C" "A" "G" "C" "G" "G"
[253] "G" "G" "C" "T" "C" "T" "C" "T" "A" "C" "G" "T" "C" "T" "T" "G" "C" "C"
[271] "G" "C" "A" "T" "C" "A" "G" "C" "G" "G" "G" "C" "G" "A" "G" "T" "C" "G"
[289] "A" "A" "A" "C" "G" "G" "G" "C" "T" "G" "T" "G" "A" "A" "T" "T" "C" "T"
[307] "A" "C" "G" "A" "C" "T" "A" "C" "C" "A" "C" "A" "A" "A" "A" "T" "A" "A"
[325] "T" "A" "A" "A" "G" "C" "T" "T" "G" "A" "T" "A" "A" "A" "C" "G" "A" "G"
[343] "G" "G" "A" "C" "C" "C" "T" "A" "T" "A" "T" "T" "G" "A" "T" "C" "C" "G"
[361] "A" "C" "A" "T" "T" "T" "C" "T" "T" "A" "T" "A" "A" "G" "T" "T" "C" "C"
[379] "T" "T" "C" "A" "C" "C" "A" "C" "G" "T" "T" "T" "T" "T" "T" "C" "A" "C"
[397] "A" "T" "G" "G" "C" "A" "A" "T" "G" "T" "G" "G" "T" "C" "C" "T" "G" "G"
[415] "C" "T" "G" "T" "C" "G" "C" "C" "T" "C" "C" "G" "T" "T" "A" "A" "C" "T"
[433] "G" "C" "T" "T" "T" "G" "C" "G" "T" "C" "T" "T" "G" "C" "G" "A" "T" "G"

```

```
generate_dna(5, collapse = TRUE)
```

```
[1] "CGGGC"
```

Q. add the ability to add a message if user is happy/sad, controlled by a new input parameter called 'mood'.

```
cat('hi')
```

```
hi
```

```

generate_dna <- function(length=9, collapse=FALSE, mood=FALSE) {
  #the nucleotides to draw/sample from
  bases <- c("A", "C", "G", "T")
  #draw n=length nucleotides to make sequence
  ans <- sample(bases, size=length, replace = TRUE)
  #concatenate
  if(collapse) {
    ans <- paste(ans, collapse="")
  }
  # mood
  if(mood) {
    cat("HAPPY")
  }
  # to explicitly print the result
  return(ans)
}
generate_dna(length=450, mood=T)

```

HAPPY

```

[1] "C" "G" "T" "A" "C" "C" "A" "T" "T" "T" "A" "G" "T" "A" "G" "A" "C" "A"
[19] "T" "C" "T" "A" "G" "C" "A" "G" "G" "T" "G" "T" "G" "T" "T" "G" "C" "G"
[37] "A" "A" "T" "T" "T" "T" "A" "G" "T" "G" "G" "C" "T" "G" "G" "T" "T" "G"
[55] "A" "C" "T" "G" "A" "T" "G" "G" "G" "T" "A" "G" "A" "C" "T" "T" "T" "T"
[73] "C" "A" "A" "T" "T" "C" "C" "C" "A" "G" "C" "G" "G" "G" "T" "A" "T" "G"
[91] "C" "T" "C" "A" "C" "C" "A" "C" "C" "C" "T" "C" "G" "A" "T" "C" "G" "T"
[109] "C" "A" "C" "T" "T" "C" "T" "T" "A" "G" "A" "A" "C" "A" "T" "G" "C" "G"
[127] "A" "T" "A" "C" "A" "C" "C" "G" "C" "G" "T" "G" "G" "T" "G" "A" "C" "C"
[145] "G" "T" "T" "T" "G" "A" "A" "T" "T" "C" "A" "C" "G" "C" "C" "A" "C" "C"
[163] "T" "G" "A" "G" "T" "T" "A" "T" "T" "A" "G" "G" "T" "C" "A" "T" "A" "T"
[181] "G" "T" "G" "G" "G" "A" "T" "C" "G" "T" "A" "G" "C" "A" "T" "G" "G" "A"
[199] "G" "T" "T" "T" "G" "T" "A" "G" "G" "A" "G" "G" "A" "G" "C" "G" "C" "C"
[217] "A" "T" "T" "G" "T" "C" "T" "C" "C" "G" "G" "G" "G" "G" "A" "A" "G" "T"
[235] "G" "A" "C" "A" "A" "G" "A" "T" "C" "A" "G" "T" "C" "G" "G" "T" "G" "C"
[253] "C" "G" "G" "G" "C" "T" "T" "C" "C" "G" "C" "A" "A" "T" "C" "C" "T" "T"
[271] "T" "T" "A" "T" "C" "T" "T" "G" "G" "T" "T" "T" "T" "C" "C" "C" "C" "A"
[289] "A" "A" "G" "T" "C" "C" "G" "G" "T" "A" "C" "A" "G" "T" "G" "G" "T" "A"
[307] "G" "C" "C" "A" "C" "C" "G" "G" "G" "T" "C" "T" "C" "T" "G" "C" "T" "A"
[325] "A" "T" "C" "C" "T" "C" "A" "T" "G" "G" "C" "G" "A" "G" "C" "G" "T" "G"
[343] "A" "T" "C" "A" "G" "T" "A" "T" "T" "A" "G" "T" "A" "G" "A" "C" "T" "G"
[361] "A" "T" "T" "G" "A" "G" "T" "T" "T" "A" "C" "G" "C" "C" "G" "T" "A" "G"

```

```
[379] "T" "G" "A" "C" "C" "G" "A" "G" "T" "C" "G" "G" "A" "G" "T" "A" "T" "T"
[397] "A" "T" "C" "C" "C" "G" "A" "G" "C" "C" "G" "T" "G" "G" "T" "A" "G" "T"
[415] "G" "T" "T" "C" "A" "G" "C" "C" "T" "C" "C" "C" "G" "C" "C" "C" "A" "G"
[433] "G" "G" "G" "T" "C" "A" "G" "G" "A" "C" "C" "A" "C" "G" "G" "C" "G" "G"
```

```
s <- generate_dna(4, mood = T)
```

HAPPY

```
s
```

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

I would like my function to print a single element vector: “GATGATCT”. To help with this, I can use the ‘paste()’ function.

```
paste(s, collapse="")
```

```
[1] "GCCC"
```

```
generate_dna(4, mood=T)
```

HAPPY

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

Q. write a function with the ability to output random amino acid sequences of a user defined length.

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

```
[1] 20
```

```

generate_protein <- function(length=6, collapse=FALSE) {
  #the nucleotides to draw/sample from
  aa <- c("A", "R", "N", "D", "C", "E", "Q", "G", "H", "I", "L", "K", "M", "F", "P", "S", "T")
  #draw n=length nucleotides to make sequence
  ans <- sample(aa, size=length, replace = TRUE)
  #concatenate
  if(collapse) {
    ans <- paste(ans, collapse="")
  }
  # to explicitly print the result
  return(ans)
}
generate_protein(length=6, collapse=T)

```

```
[1] "INENWA"
```

Q. generate protein sequences from length 6-12 amino acids.

```
generate_protein(length=6, collapse=T)
```

```
[1] "DAFSTA"
```

This does NOT work since the function is not vectorized for 'length'.

In particular, the 'sapply()' can do this. It is used to apply a function to each element of a vector or list and simplify the result.

```
sapply(6:12, generate_protein, collapse=T)
```

```

[1] "HVTTWQ"      "MNAFCLG"      "GEEVRDLM"      "SAADMNEND"      "NRHPKLRQAT"
[6] "MLYAICMVPTY" "RTMIWQKPLWMP"

```

Q. are any of these sequences unique / never found in nature?

To make this accessible, let's convert our sequences to FASTA format: >id. 6 VLEEGC >id. 7 MHRYEYP

```

myseqs <- sapply(6:12, generate_protein, collapse=T)
myseqs

```

```
[1] "NMAHLY"      "PCLNLPQ"      "QGSFLTHT"      "HKATLLGIY"      "EGITNRQGHF"
[6] "YTMQRMLRTIT" "VEQSDTCMELFQ"
```

The functions 'paste()' and 'cat()' will help here:

```
cat( paste(">id.", 6:12, "\n", myseqs, "\n", sep=""), sep="")
```

```
>id.6
NMAHLY
>id.7
PCLNLPQ
>id.8
QGSFLTHT
>id.9
HKATLLGIY
>id.10
EGITNRQGHF
>id.11
YTMQRMLRTIT
>id.12
VEQSDTCMELFQ
```

```
library(bio3d)

myseqs <- sapply(6:12, generate_protein, collapse=T)
myseqs
```

```
[1] "KCNQTC"      "QFKIPHN"      "FQCDCFNE"      "NGEPYINKV"      "MCLFQDWLDQ"
[6] "KIKRGWHGTNL" "YIHERLNGIMSW"
```

```
#as.fasta(myseqs[1])
```