Class 6: R Functions Lab

Ashley Mazon (A17478903)

All functions in R have at least 3 things:

- A **name**, we pick this and use it to call the function.
- Input **arguments**, there can be multiple comma seperated inputs to the function
- The **body**, lines of R code that do the work of the function

Our first wee function:

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

Let's test our function

```
add(c(1,2,3),y=10)
```

```
[1] 11 12 13
```

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

```
[1] 110
```

A second function

Let's try something more interesting. Make a seqeunce generation tool.

The `sample()` function could be useful here. ex. "give me 3 numbers between 1 through 10"

```
sample(1:10, size=3)
```

```
[1] 3 8 6
```

Change this to work with the nucleotides A C G and T and return 3 of them

```
n<- c("A","C","T","G")
sample(n, size =15, replace=TRUE)
```

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

Turn this snipet into a function that returns a user specified length dna sequence. Let's call it `generate_dna()`

```
generate_dna <- function(len=10,fasta=FALSE) {
  n<- c("A","C","T","G")
  v<- sample(n, size=len, replace =TRUE)

  #Make a single element vector
  s<- paste(v,collapse="")

  cat("Well done you!\n")

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

```
generate_dna(6,FALSE)
```

```
Well done you!

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

I want the option to return a single element character vector with my sequence all together like this: "GGAGTAC"

```
generate_dna(10,fasta=TRUE)
```

```
Well done you!

[1] "CGATGCCAGG"
```

A more advanced example

Make a third function that generates protein sequence of a user specified length and format

```
make_protein <- function(length = 10, fasta = TRUE) {
  aa <- c("A","R","N","D","C","Q","E","G","H","I","L","K","M","F","P","S","T","W","Y","V")
  seq <- sample(aa, size = length, replace = TRUE)
  if (fasta) {
    seq <- paste(seq, collapse = "")
    }
  return(seq)
}
```

```
make_protein(10)
```

```
[1] "RHSLHNIEFL"
```

Q. Generate random protein seqeunces between lengths 5 and 12 amino acids

One approach is to do this by brute force calling our function for each length 5 to 12.

Another approach is to write a `for()` loop to itterate over the input valued 5 to 12.

A very useful third R specific approach is to use the `sapply()` function

```
seq_lengths <- 5:12
for (i in seq_lengths) {
  cat(">", i, "\n")
  cat (make_protein(i))
  cat ("\n")
}
```

```
> 5
YMYQY
> 6
AGSMNM
> 7
ELTNATA
> 8
DESVHQFA
> 9
GIWCYKFGM
> 10
```

```
SVEEATEFAC
> 11
INSIDGHHGDF
> 12
WGHGTWGQRQHW
```

```
sapply(5:12, make_protein)
```

```
[1] "FQEVA"        "MVVGDF"       "FSIWVLV"      "NYNTCEQG"     "NPHPMTIQY"
[6] "ISKQHISMHV"   "GEIQTKKIHPP"  "MCANEIEYAYVA"
```

Key-Point: Writing functions in R is doable but not the easiest thing. Startinf with a working snippet of code and then using LLM tools to improve and generalize your function cosde is a productive approach.