Class 6: R Functions Lab

Michael Romero (A18135877)

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 separated inputs ot 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
}
```

Lets 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 sequence generation tool.

The `sample()` function could be useful here.

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

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

change this to work with the nucleotides A C G and T

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

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

TUrn this snippet 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!")

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

```
generate_dna(5)
```

```
Well done you!
```

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

```
s<-generate_dna(15)
```

```
Well done you!
```

```
s
```

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

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

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

```
Well done you!
```

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

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

```
Well done you!
```

```
[1] "CTTATTTCCC"
```

A more advanced example

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

```
generate_dna<-function(size=20, fasta=FALSE) {
  n<-c("A","C","T","G")
  v<- sample(n, size=size, replace = TRUE)

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

  cat("Well done you!")

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

```
generate_dna(6)
```

```
Well done you!
```

```
[1] "A" "A" "G" "A" "G" "G"
```

```
generate_protein<-function(size=15, 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=size, replace=TRUE)

  if (fasta) {
    return(paste(seq, collapse=""))
  } else{
    return(seq)
    }
}
```

```
generate_protein(5)
```

```
[1] "DLGWT"
```

> Q. generate random protein sequences 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.

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

```
[1] "IIKMR"        "NFFCHH"       "VGQLLRV"      "RKSFAIHS"     "VHCFIYDEG"
[6] "CSDFCVIDKE"   "DNWCVREVFCI"  "CSWHQIYMGQVT"
```

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