R Functions

Aileen Andrade (PID A17033749)

We call functions to do all our work. Today we will get more exposure to functions in R and learn how to write our own.

A first silly function

Note that arguments 2 and 3 have default values (because we set y=0 and z=0) so we don't have to supply them when we call our function.

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

Can I just use this?

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

```
[1] 2
```

```
add(1, c(10,100))
```

```
[1]  11 101
```

```
add(100)
```

```
[1] 100
```

```
add(100,10,1)
```

```
[1] 111
```

A second, more fun function

Let's write a function that generates random nucleotide sequences.

We can make use of the in-built `sample()` function in R to help us here.

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

```
[1] 6
```

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

```
 [1]  3 10  1  6  9  6  9  2  6  6  3
```

> Q. Can you use `sample()` to generate a random nucleotide sequence of length 5.

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

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

> Q. Generate a function `generate_dna()` that makes a nucleotide sequence of a user specified length.

Every function in R has at least 3 things:

- A **name** (in our case "generate_dna")
- One or more **input arguments** (the "length" of sequence we want)
- A **body** (R code that does the work)

```
generate_dna <- function(length=5) {
  sample(x=c("A","T","C","G"), size=length, replace=T)
}
```

```
generate_dna(10)
```

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

> Q. Can you write a `generate_protein()` function that returns an amino acid sequence of a user requested length?

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

generate_protein <- function(length=5) {
  sample(x=aa, size=length, replace=T)
}
```

```
generate_protein(20)
```

```
 [1] "T" "T" "Q" "T" "E" "V" "Q" "H" "I" "L" "M" "A" "D" "F" "A" "V" "I" "T" "L"
[20] "Q"
```

I want my output of this function to not be a vector with one amino acid per element, but rather a one element single string.

```
bases <- c("A","T","C","G")
paste(bases, collapse="")
```

```
[1] "ATCG"
```

```
generate_protein <- function(length=5) {
  s <- sample(x=aa, size=length, replace=T)
  paste(s, collapse="")
}
```

```
generate_protein()
```

```
[1] "PFNST"
```

Q. Generate protein sequences from legnth 6 to 12

```
generate_protein(6)
```

```
[1] "QNSFWH"
```

```
generate_protein(7)
```

```
[1] "IPPPYVR"
```

```
generate_protein(8)
```

```
[1] "FYKTGYNL"
```

We can use the useful utility function `sapply()` to help us "apply" our function over all the values 6 to 12.

```
ans <- sapply(c(6:12), "generate_protein")
```

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

```
>ID.6
HKFDFD
 >ID.7
KQPAMGC
 >ID.8
VERRFIYT
 >ID.9
ADTSNVFHS
 >ID.10
KQTQDRDQME
 >ID.11
MDVEWKFYVRF
 >ID.12
EQNYDTYRGNHD
```

> Q. Are any of these sequences unique in nature - i.e. never found in nature? We can search "refseq-protein" and look or 100% identity.

A BLASTp search into the Refseq_protein database did not show complete matches in nature.