Class 6: R functions

AUTHOR
Tiffany 15700705

My first function :-)

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

Can I use it?

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

```
[1] 2
```

```
add(x=1, y=100)
```

```
[1] 101
```

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

```
[1] 101   2 101
```

You can also give y a default function so you only specify x

```
add <- function(x, y=1) {
  x + y
}
add(10)
```

```
[1] 11
```

You can still override the default y=1 if you specify y. The y=1 is the default y placeholder, if you spell out a value for y, it will use your new value instead.

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

```
[1] 20
```

What if you wanted to have 3 inputs, but not always use the third?

```
add <- function(x, y=1, z=0){
  x + y + z
}
add(1,1,1)
```

```
[1] 3
```

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

```
[1] 2
```

Make a function `generate_DNA()` that makes a random nucleotide sequence of any length

```
#before writing as a function, try on its own
bases <- c("A", "G", "C", "T")
#look at ?sample to see documentation. use this to grab bases in random order
sample(bases, size = 10, replace = TRUE)
```

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

Now, you can turn it into a function!

```
generate_DNA <- function(y) {
  bases <- c("A", "G", "C", "T")
  sample(bases, size = y, replace = TRUE)
}
generate_DNA(y=10)
```

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

```
generate_DNA(15)
```

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

Barry's version of function:

```
generate_DNA_barry <- function(length) {
  bases <- c("A", "G", "C", "T")
  sequence <- sample(bases, size = length, replace = TRUE)
  return(sequence)
}
generate_DNA_barry(10)
```

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

```
#to only grab this from the package, not everything in this package
bio3d::aa.table$aa1
```

```
 [1] "A" "R" "N" "D" "C" "Q" "E" "G" "H" "I" "L" "K" "M" "F" "P" "S" "T" "W" "Y"
[20] "V" "X" "D" "R" "C" "C" "C" "C" "C" "C" "C" "C" "H" "E" "H" "H" "H" "H" "H"
```

```
[39] "H" "D" "K" "K" "M" "K" "M" "C" "F" "Y" "S" "T"
```

```
unique(bio3d::aa.table$aa1)
```

```
 [1] "A" "R" "N" "D" "C" "Q" "E" "G" "H" "I" "L" "K" "M" "F" "P" "S" "T" "W" "Y"
[20] "V" "X"
```

Write a function to generate a random amino acid sequence

```
generate_protein <- function(length) {
  amino <- unique(bio3d::aa.table$aa1)
  sequence <- sample(amino, length, replace= TRUE)
  return(sequence)
}
generate_protein(6)
```

```
[1] "Y" "R" "P" "G" "C" "C"
```

Generate random sequences of length 6 to 12 using `sapply()` where x can be a vector, to apply the function to a vector.

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

```
[[1]]
[1] "M" "V" "K" "X" "I" "R"

[[2]]
[1] "Q" "X" "R" "S" "W" "I" "X"

[[3]]
[1] "K" "G" "W" "Y" "Y" "V" "E" "M"

[[4]]
[1] "F" "G" "E" "K" "L" "Y" "A" "N" "F"

[[5]]
 [1] "P" "M" "A" "D" "H" "I" "H" "N" "P" "T"

[[6]]
 [1] "F" "N" "C" "V" "W" "T" "W" "Y" "F" "E" "A"

[[7]]
 [1] "L" "D" "A" "G" "Y" "E" "N" "A" "I" "R" "D" "I"
```

Want to change this sequence format into something that can be pasted into **BLAST** (FASTA format). Use `paste()` to remove spaces in between aminmo acids via *collapse* argument.

```
generate_protein <- function(length) {
  amino <- unique(bio3d::aa.table$aa1)
  sequence <- sample(amino, length, replace= TRUE)
  sequence <- paste(sequence, collapse ="")
  return(sequence)
}
answer <- sapply(6:12, generate_protein)
answer
```

```
[1] "YTXXXR"       "CEDAASX"      "TPPTIXFG"     "IKCEVIETK"    "SXPIKWLLYT"  
[6] "VFLCCDQSCSK"  "SLXDTXAHEECQ"
```

```
#using paste again to add unique IDs to the above sequences and format as FASTA
cat(paste(">id.", 6:12, "\n", answer, sep = ""), sep = "\n")
```

```
>id.6
YTXXXR
>id.7
CEDAASX
>id.8
TPPTIXFG
>id.9
IKCEVIETK
>id.10
SXPIKWLLYT
>id.11
VFLCCDQSCSK
>id.12
SLXDTXAHEECQ
```