Class 6: R functions

Morgan Black (PID A14904860)

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

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 assign a number to x or y within the function
add <- function(x,y=1){
  x+y
}

#then you don't have to put a y value when you use it because it defaults
#to the assigned value given in the function
add(10)
```

```
[1] 11
```

```r
#however, you can override the original y value by giving it a new one
add(10,10)
```

```
[1] 20
```

```r
#Make a function to generate a random nucleotide sequence of any length

nucleotides <- c("A", "C", "G", "T")
sequence <- sample(nucleotides, size = 10, replace = TRUE)

generate_dna <- function(length){
  nucleotides <- c("A", "C", "G", "T")
  sequence <- sample(nucleotides, size = length, replace = TRUE)

  return(sequence) #prints out the answer to the function
}

#try the function to get a random 9-base sequence
generate_dna(9)
```

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

```r
#Installed the bio3d package in the console. Call the amino acid table within
#the package and just subset the column that has the single-letter AA codes.
#The unique() pulls out only one of each amino acid and gets rid of duplicates
#from the table. The [1:20] subset at the end gets rid of the "X" code which is
#not a common canonical (?) amino acid. The paste() function returns the
#sequence without quotes or spaces or anything between each amino acid in the
#sequence

generate_protein <- function(length){
  amino_acids <- unique(bio3d::aa.table$aa1)[1:20]
  sequence <- sample(amino_acids, size = length, replace = TRUE)
  sequence <- paste(sequence, collapse = "")

  return(sequence) #prints out the answer to the function
}

generate_protein(22)
```

```
[1] "DREPEWIIYVRNMTEDEWDYST"
```

```
#Generate random protein sequences of length 6 to 12. Use the sapply() function
#to apply a function to a vector of values


answer <- sapply(6:12, generate_protein)
answer
```

```
[1] "HVFMSL"       "FILGRWG"      "PTMNQMFC"     "MCKSIGPQW"    "VKRNCYYGVY"
[6] "ANSDNLNSGTF"  "IQWSMDMEWKRV"
```

```
#Put the sequences together into FASTA format to directly search in blastp
#the "\n" puts in an enter/return/new line

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

```
>id.6
HVFMSL
>id.7
FILGRWG
>id.8
PTMNQMFC
>id.9
MCKSIGPQW
>id.10
VKRNCYYGVY
>id.11
ANSDNLNSGTF
>id.12
IQWSMDMEWKRV
```