

# class06: R functions

Sylvia Ho (PID: a18482382)

## background

functions are at the heart of using r everything insvolves calling and using functions all fns have at least 3 thigns:

1. a **name** the thing we use to call it
2. input **args** that are comma separated
3. **body** lines of code btwn curly {} that does work of fn

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

```
add(c(100,200,300))
```

```
[1] 101 201 301
```

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

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

```
[1] 110
```

```
log(10,base=10)
```

```
[1] 1
```

**n.b** input args can be either reqd or opt the latter have fall back default that is spec in fn w an equal sign

```
#add(x=100,y=200,z=300)
```

## a second fn

all fns in R look like this

```
name<fn(arg){ body }
```

thesample() fn in R samples randomly from arg

```
sample(1:10)
```

```
[1]  5  9  3 10  1  4  6  8  2  7
```

Q write a first version fn called `generate_dna()` that gens a user spec length n random dna seq

```
generate_dna<-function(n=6){  
  bases<-c("a","t","g","c")  
  sample(bases, size=n, replace = TRUE)  
}  
  
generate_dna(100)
```

```
[1] "c" "g" "t" "g" "t" "c" "a" "c" "t" "g" "a" "g" "g" "c" "t" "a" "a" "a"  
[19] "c" "t" "a" "c" "c" "c" "g" "t" "a" "a" "t" "a" "c" "t" "c" "c" "t" "a"  
[37] "t" "a" "t" "c" "c" "g" "t" "a" "a" "t" "t" "a" "t" "c" "a" "t" "g" "t"  
[55] "t" "a" "g" "t" "a" "c" "a" "a" "c" "a" "g" "g" "t" "a" "c" "a" "g" "t"  
[73] "g" "t" "t" "c" "c" "t" "a" "t" "g" "a" "a" "c" "c" "g" "t" "t" "a" "t"  
[91] "a" "t" "a" "c" "c" "g" "t" "a" "a" "a"
```

Q mod to return actual seq

```
generate_dna<-function(n=6){
  bases<-c("a","t","g","c")
  ans<-sample(bases, size=n, replace = TRUE)
  an<-paste(ans,collapse="")
  return(an)
}

generate_dna(10)
```

```
[1] "cagcgcacgt"
```

Q option FASTA or vector

```
generate_dna<-function(n=6, fasta = TRUE){
  bases<-c("a","t","g","c")
  ans<-sample(bases, size=n, replace = TRUE)

  if (fasta){
    ans<-paste(ans,collapse="")
  }
  return(ans)
}
```

```
generate_dna(10)
```

```
[1] "gtatgtaaac"
```

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

```
[1] "g" "t" "g" "t" "g" "g" "g" "a" "g" "g"
```

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

```

generate_protein<-function(n=6, fasta = TRUE){
aa<-c("A", "R", "N", "D", "C", "E", "Q", "G", "H", "I",
"L", "K", "M", "F", "P", "S", "T", "W", "Y", "V")

  ans<-sample(aa, size=n, replace = TRUE)

  if (fasta){
    ans<-paste(ans,collapse="")
  }
  return(ans)
}
generate_protein(10, fasta = F)

```

```
[1] "Q" "D" "G" "Y" "A" "K" "I" "Q" "W" "E"
```

Q gen protein gn to gen seqs between 6 to 12, check in uniqueue aka against NR database at NCBI

```

generate_protein<-function(n=6, fasta = TRUE){
aa<-c("A", "R", "N", "D", "C", "E", "Q", "G", "H", "I",
"L", "K", "M", "F", "P", "S", "T", "W", "Y", "V")

  ans<-sample(aa, size=n, replace = TRUE)

  if (fasta){
    ans<-paste(ans,collapse="")
  }
  return(ans)}

```

```

for (i in 6:12) {
  cat(">",i, "\n")
  cat(generate_protein(i), "\n")
}

```

```

> 6
KDSSLP
> 7
VRYCIAC

```

> 8  
HTETEVSQ  
> 9  
NYVQYMHVE  
> 10  
VGQEAENICL  
> 11  
ANCPCYCGESY  
> 12  
LTSFPCLEQPPD