

# Class 6: R functions

Mudit

My first function

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

Can I just use it?

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

```
[1] 2
```

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

```
[1] 101
```

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

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

```
add(10)
```

```
[1] 11
```

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

```
[1] 2
```

```

GENERATE_DNA <- function(l = 1){

  bases <- c('A', 'T', 'G', 'C')
  sequence1 <- sample(bases, l, replace = TRUE)
  #paste0(sample(bases, l, replace = TRUE), collapse = "")
  sequence1 <- paste0(sample(bases, l, replace = TRUE), collapse = "")
  return(sequence1)

}

```

```
GENERATE_DNA(100)
```

```
[1] "CAGACATCTTAATATGAGTGTGAAGAAAGATAATAACTCACTTATTAATCCGCGTTTTCTGGGGGTTCCGTTTGCTATGCGCAATGGG"
```

```

GENERATE_PROTEIN <- function(l = 1){

  amino <- unique(bio3d::aa.table$aa1)
  bases <- c('A', 'T', 'G', 'C')
  #sample(bases, l, replace = TRUE)
  #paste0(sample(bases, l, replace = TRUE), collapse = "")
  sequence1 <- paste0(sample(amino, l, replace = TRUE), collapse = "")
  return(sequence1)

}

```

```
GENERATE_DNA(100)
```

```
[1] "ACCCCGACCACTGCGCGTGATTGGATATACTCCCGCGAGCTCATCGGCTGGGACCGATGCACGTCTAATGACCCGAAGAACCGG"
```

```
unique(bio3d::aa.table$aa1)[1:20]
```

```

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

```

```
View(bio3d::aa.table)
```

Generate random protein sequences of length 6 to 12

```
answer<-sapply(6:12, GENERATE_PROTEIN)
cat(paste(">id.", 6:12, "\n", answer, sep="'), sep="\n")
```

```
>id.6
XSGFPS
>id.7
IDTHTEC
>id.8
TCRYNPYP
>id.9
GHCLSSRWS
>id.10
YVHSCMEGIR
>id.11
QPYTLLYEAYF
>id.12
AQTETVVTWGIV
```

```
#grade() <- function(x = c(1, 1, 1, 1)){
#   x<-sort(x)
#}
```

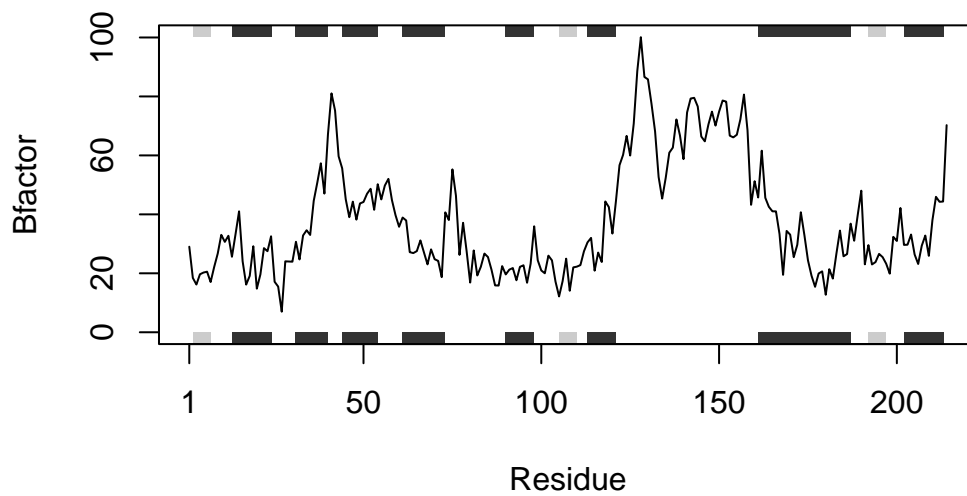
```
df <- read.csv("student_homework.csv")
```

#Input: function takes the PDB ID character sequence as an input  
 #function plots chain A flexibility of the protein (whose pdb ID is given as an argument).  
 #output: It plots the B-factor vs residue position

```
func_x <- function(pdbID){
  #calling the bio3d library
  library(bio3d)
  # Reading the pdb ID and assigning it to object 's'
  s <- read.pdb(pdbID)
  #trims the structure and include chain A's alpha carbon atoms
  s.chainA <- trim.pdb(s, chain="A", elety="CA")
  #extracts the B-factors from the trimmed structure
  s.b <- s.chainA$atom$b
  #ploting secondary structure elements against b factors
  plotb3(s.b, sse=s.chainA, typ="l", ylab="Bfactor")
}
```

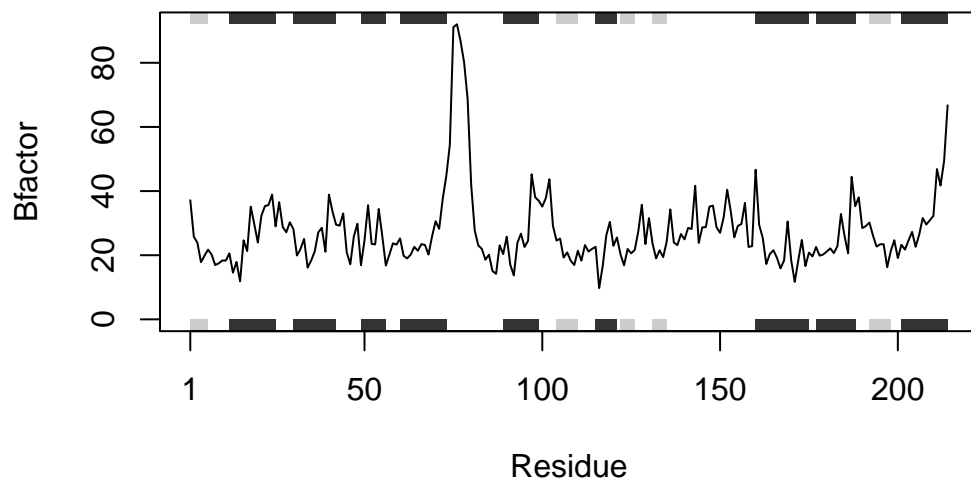
```
#| warning: false
#vector of pdbIDs
seq_vec <- c("4AKE", "1AKE", "1E4Y")
#looping over
sapply(seq_vec, func_x)
```

Note: Accessing on-line PDB file

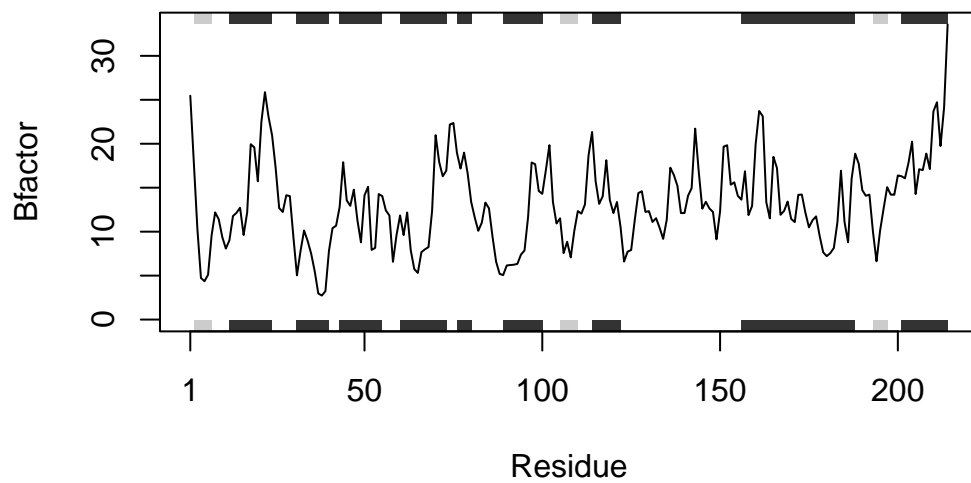


Note: Accessing on-line PDB file

PDB has ALT records, taking A only, rm.alt=TRUE



Note: Accessing on-line PDB file



\$`4AKE`

NULL

\$`1AKE`

NULL

\$`1E4Y`

NULL

```
s1 <- read.pdb("4AKE") # kinase with drug
```

Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):  
/var/folders/pq/n\_hbs121309bf8n2d4n71v100000gn/T//RtmpjBh9hW/4AKE.pdb exists.  
Skipping download

```
s2 <- read.pdb("1AKE") # kinase no drug
```

Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):  
/var/folders/pq/n\_hbs121309bf8n2d4n71v100000gn/T//RtmpjBh9hW/1AKE.pdb exists.  
Skipping download

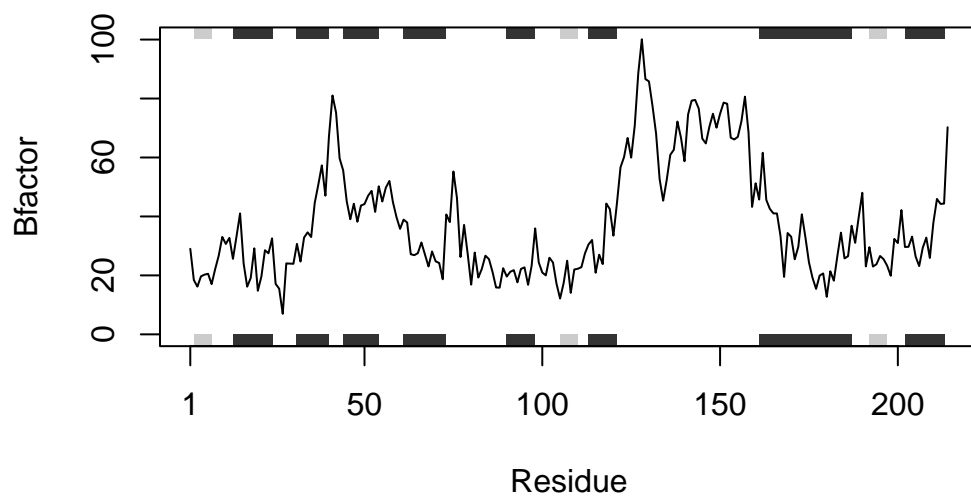
PDB has ALT records, taking A only, rm.alt=TRUE

```
s3 <- read.pdb("1E4Y") # kinase with drug
```

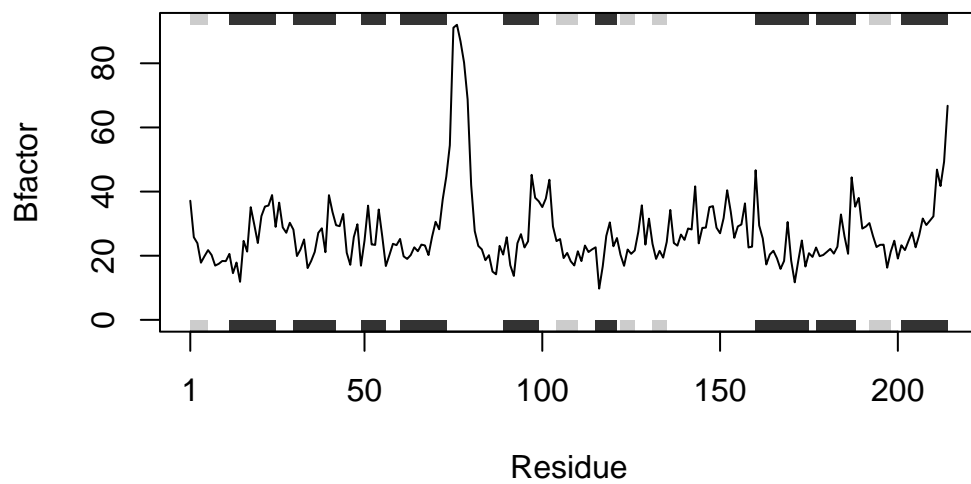
Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):  
/var/folders/pq/n\_hbs121309bf8n2d4n71v100000gn/T//RtmpjBh9hW/1E4Y.pdb exists.  
Skipping download

```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")  
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")  
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")  
s1.b <- s1.chainA$atom$b  
s2.b <- s2.chainA$atom$b  
s3.b <- s3.chainA$atom$b  
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



```
plotb3(s2.b, sse=s2.chainA, typ="l", ylab="Bfactor")
```



```
plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")
```

