

# Class 6: R Functions

AE Pasquinelli

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

#run add to put in functions and be used again

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

```
[1] 20
```

```
add(10)
```

```
[1] 11
```

```
add(10, y=10)
```

```
[1] 20
```

## A second function

#Let's try something more interesting

```
#generate_dna < function
```

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

```
sequence <- sample(bases, size=10, replace=TRUE)
```

#These 2 snippets work and now I can make it into a function

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

```
generate_DNA(10)
```

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

```
generate_DNA(length=10)
```

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

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

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

#added paste to above to remove spaces

```
generate_protein(15)
```

```
[1] "DTYNVNFRNGFPMPM"
```

```
generate_protein(length=7)
```

```
[1] "TMGADQL"
```

```
generate_protein(length=6)
```

```
[1] "CHMVDP"
```

```
generate_protein(length=7)
```

```
[1] "RMNDDDV"
```

```
generate_protein(length=8)
```

```
[1] "RCEQSWLK"
```

```
generate_protein(length=9)
```

```
[1] "AASHTFWMV"
```

```
generate_protein(length=10)
```

```
[1] "FTLFELSIIF"
```

```
generate_protein(length=11)
```

```
[1] "ALYHIQAGKDI"
```

```
generate_protein(length=12)
```

```
[1] "HYAPCEIWIKTL"
```

```
generate_protein(length=13)
```

```
[1] "NECCISYGHKNT"
```

```
generate_protein()
```

```
[1] "YWCTVKWADCYEPSQRGVLQ"
```

generate protein seq of length 6 to 12

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

```
[1] "LSCYKT"      "GIDEILP"      "ACYEIFSA"      "PHYGYGLG"      "EVTTWPPEIK"
[6] "NMKSWKAIMP"  "QLNQMVHNEMFW"
```

```
paste(">id.", 6:12, sep = "")
```

```
[1] ">id.6" ">id.7" ">id.8" ">id.9" ">id.10" ">id.11" ">id.12"
```

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

```
[1] ">id.6\nLSCYKT\n"      ">id.7\nGIDEILP\n"      ">id.8\nACYEIFSA\n"
[4] ">id.9\nPHYGYGLG\n"    ">id.10\nEVTTWPPEIK\n"  ">id.11\nNMKSWKAIMP\n"
[7] ">id.12\nQLNQMVHNEMFW\n"
```

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

```
>id.6
LSCYKT
>id.7
GIDEILP
>id.8
ACYEIFSA
>id.9
PHYGYGLG
>id.10
EVTTWPPEIK
>id.11
NMKSWKAIMP
>id.12
QLNQMVHNEMFW
```

**Can you improve this analysis code?**

```
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

Note: Accessing on-line PDB file

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

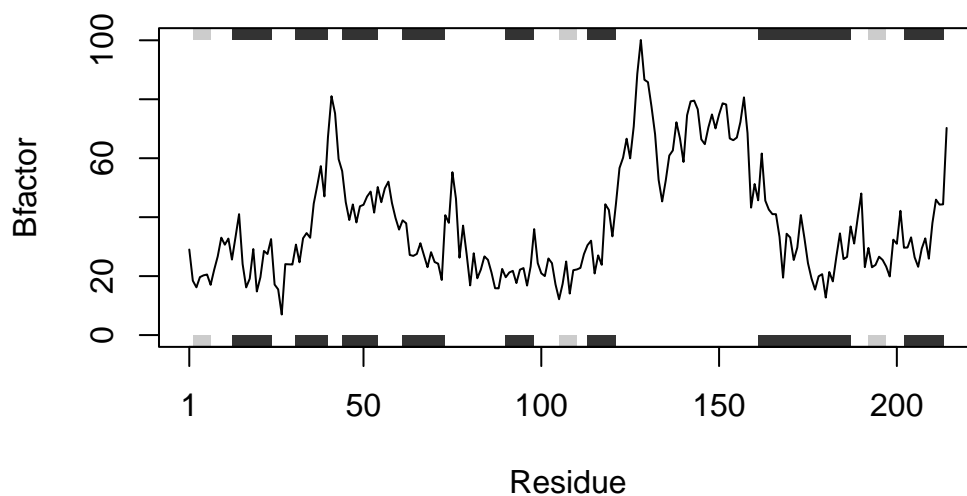
Note: Accessing on-line PDB file

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

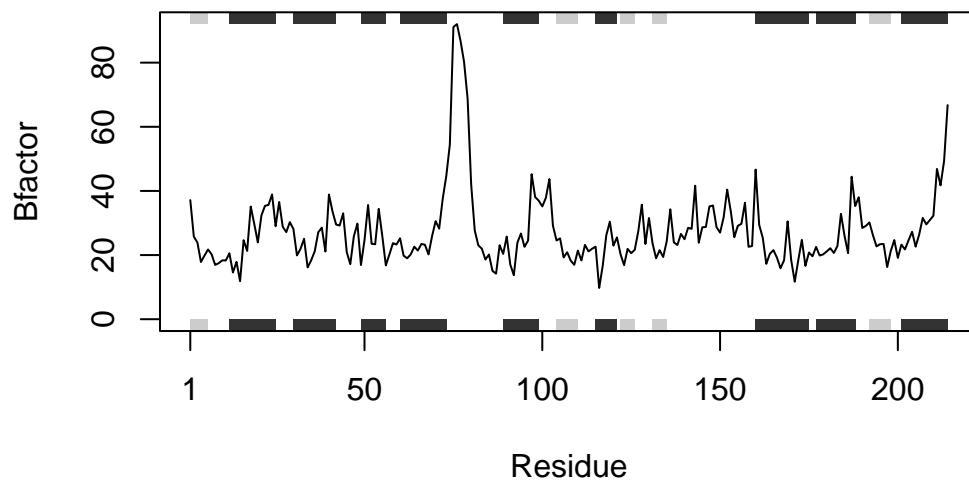
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

Note: Accessing on-line PDB file

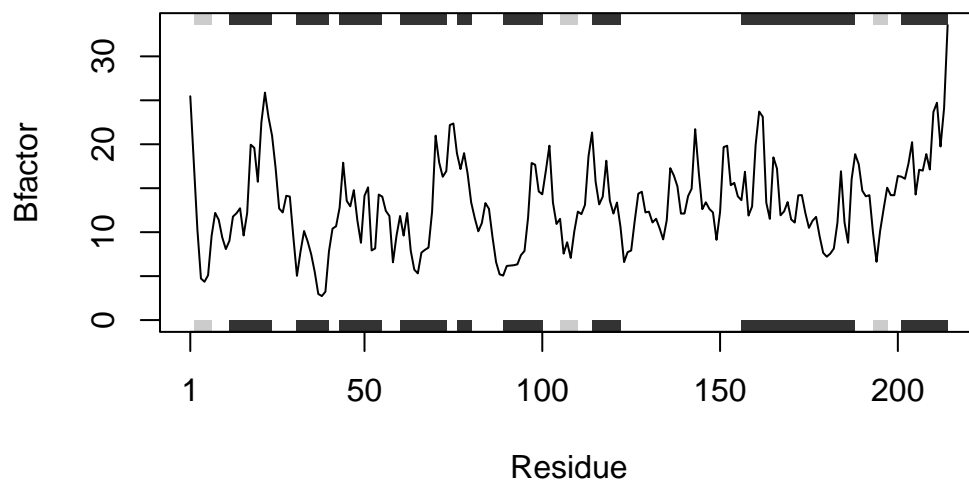
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



```

pdb_processing <- function (pdb) {
  s <- read.pdb(pdb)
  s.chainA <- trim.pdb(s, chain="A", elety="CA")
  s.b <- s.chainA$atom$b
  plotb3(s.b, sse=s.chainA, typ="l", ylab="Bfactor")
}

```

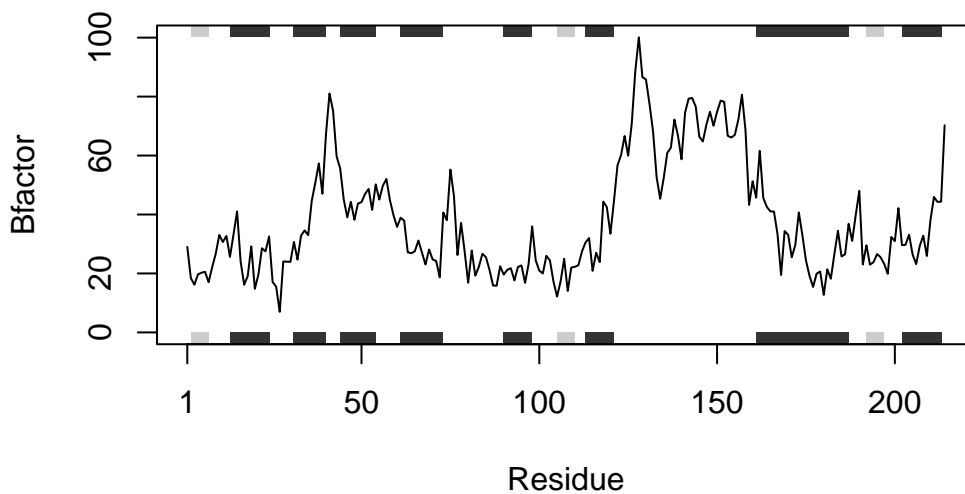
```

pdb_processing("4AKE")

```

Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):  
 /var/folders/sr/sqmg88lx1fg6x16w\_vr1cj1h0000gp/T//RtmpRxjs5N/4AKE.pdb exists.  
 Skipping download



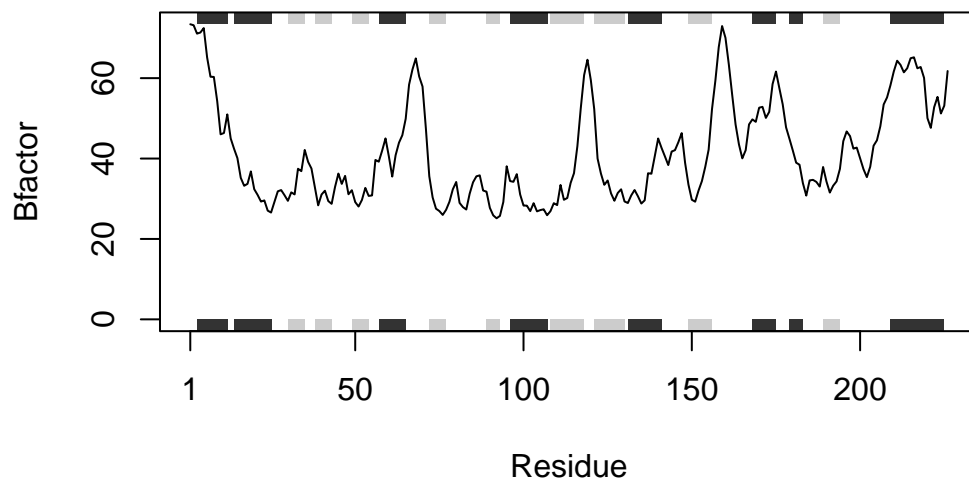
#use this code for TBP

```

pdb_processing("2CZR")

```

Note: Accessing on-line PDB file



#use apply to run multiple variables

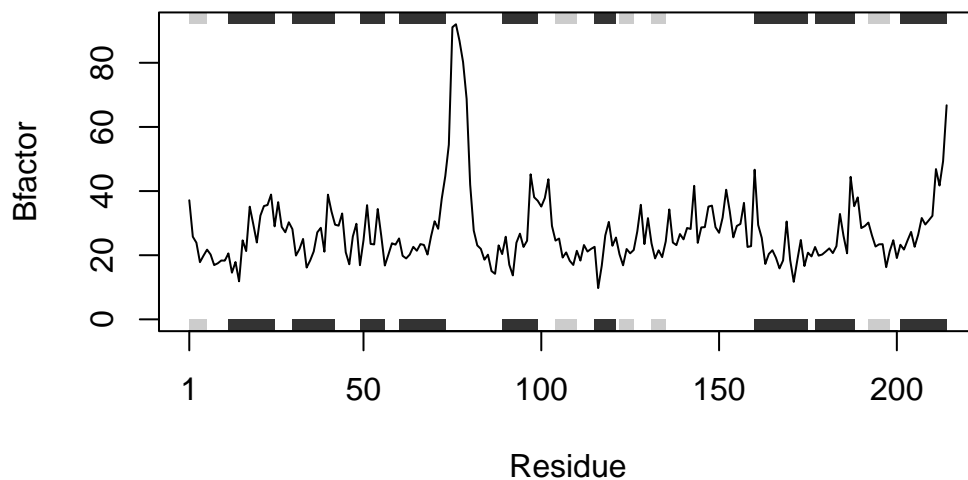
```
pdb_processing("1AKE")
```

Note: Accessing on-line PDB file

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):  
/var/folders/sr/sqmg88lx1fg6x16w_vr1cj1h0000gp/T//RtmpRxjs5N/1AKE.pdb exists.  
Skipping download
```

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

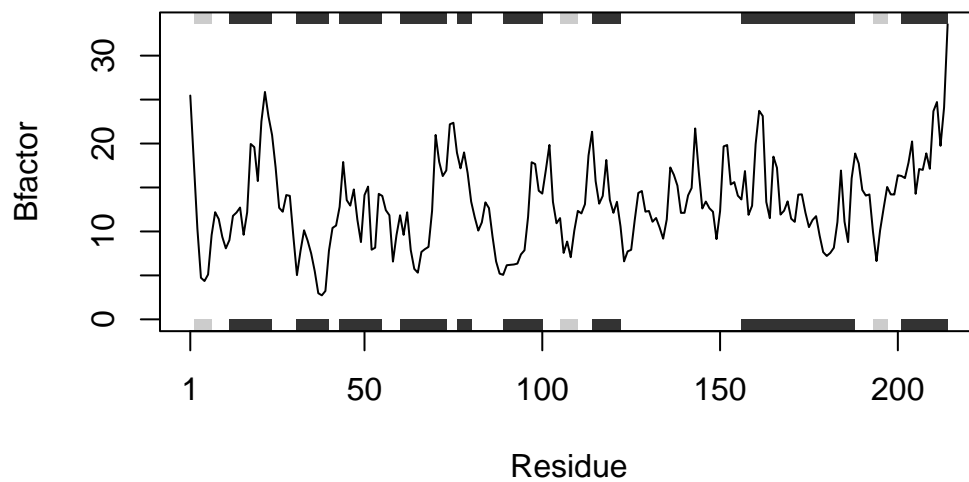




```
pdb_processing("1E4Y")
```

Note: Accessing on-line PDB file

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):  
/var/folders/sr/sqmg88lx1fg6x16w_vrlcj1h0000gp/T/RtmpRxjs5N/1E4Y.pdb exists.  
Skipping download
```



```
pdb_processing("1K1F")
```

Note: Accessing on-line PDB file

