# HWClass6

Sung Lien A16628474

```
#Can you improve this analysis code?
library(bio3d)
s1 <- read.pdb("4AKE")  # kinase with drug
```

```
  Note: Accessing on-line PDB file
```

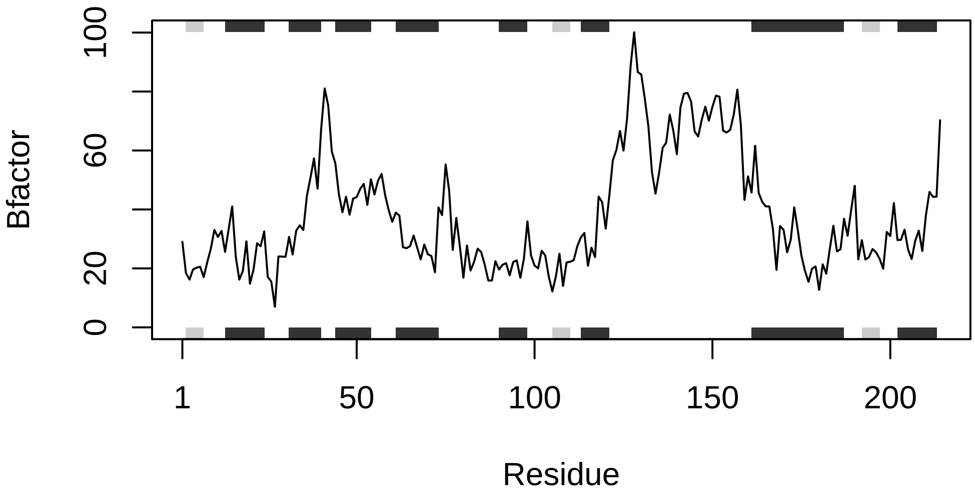
```
s2 <- read.pdb("1AKE")  # kinase no drug
```

```
  Note: Accessing on-line PDB file
   PDB has ALT records, taking A only, rm.alt=TRUE
```

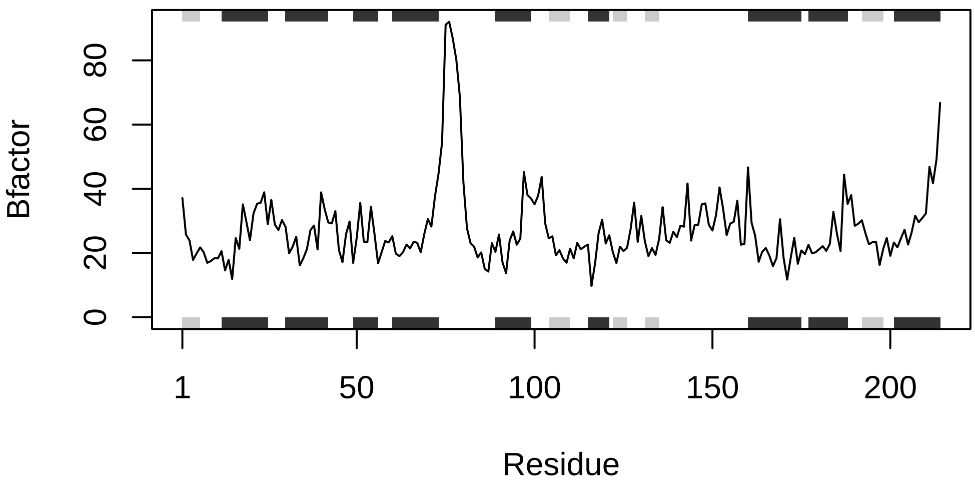
```
s3 <- read.pdb("1E4Y")  # kinase with drug
```

```
  Note: Accessing on-line PDB file
```

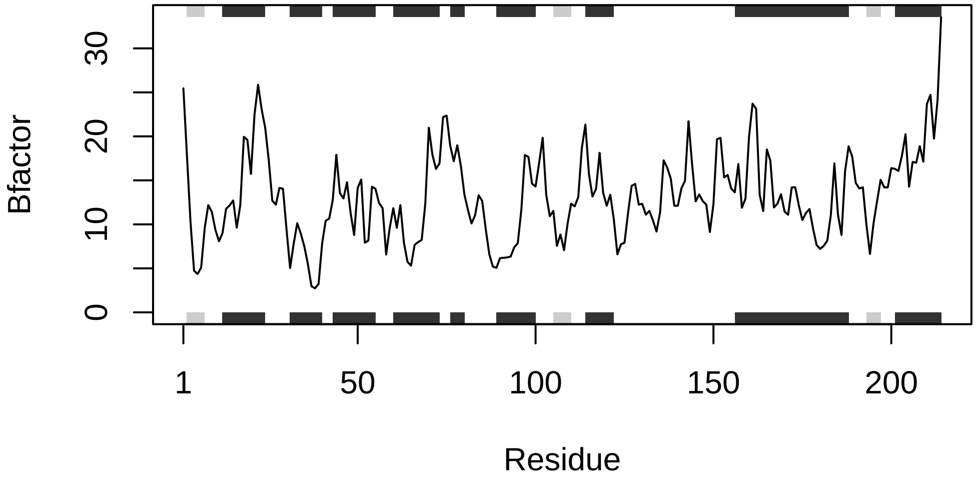
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



> Q6. How would you generalize the original code above to work with any set of input protein structures?

#comments on what are the inputs to the function. #Ans:input any protein PDB data

#what the function does and how to use it. #Ans: The function first reads PDB, second isolates chain A, third isolates atom B, fourth plots

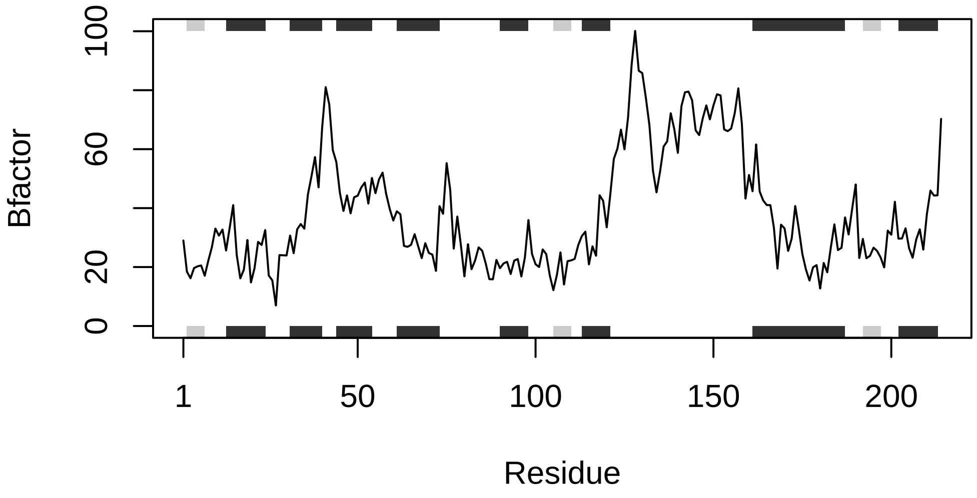
#what is the output of the function. #Ans: protein_analyze(), it outputs a plot that anazlyzes protein

```
protein_analyze <- function(x) {
# read pdb
x1 <- read.pdb (x)
# isolate chain A
x1.chainA <- trim.pdb(x1, chain="A", elety="CA")
# atom B
x1.b <- x1.chainA$atom$b
# plot
plotb3(x1.b, sse=x1.chainA, typ="l", ylab="Bfactor")
}
```

```r
protein_analyze("4AKE")
```

```
  Note: Accessing on-line PDB file
```

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/jc/h0473_4n28s051rz54q4nxv40000gn/T//RtmpLq4bDp/4AKE.pdb exists.
Skipping download
```


Bfactor
Residue