Class06 HW Q 6

AUTHOR
Joseph Kesler

Given code:

```
# Can you improve this analysis code?
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.1.3
```

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

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

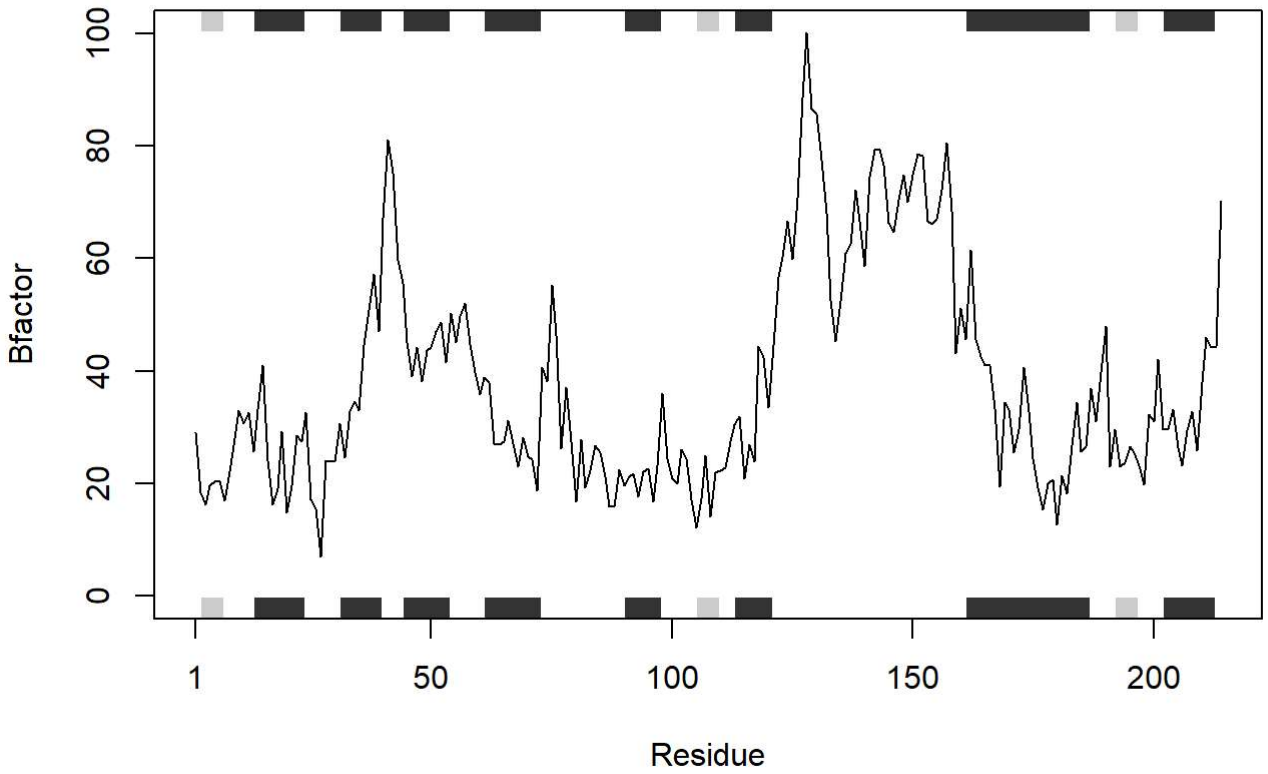
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

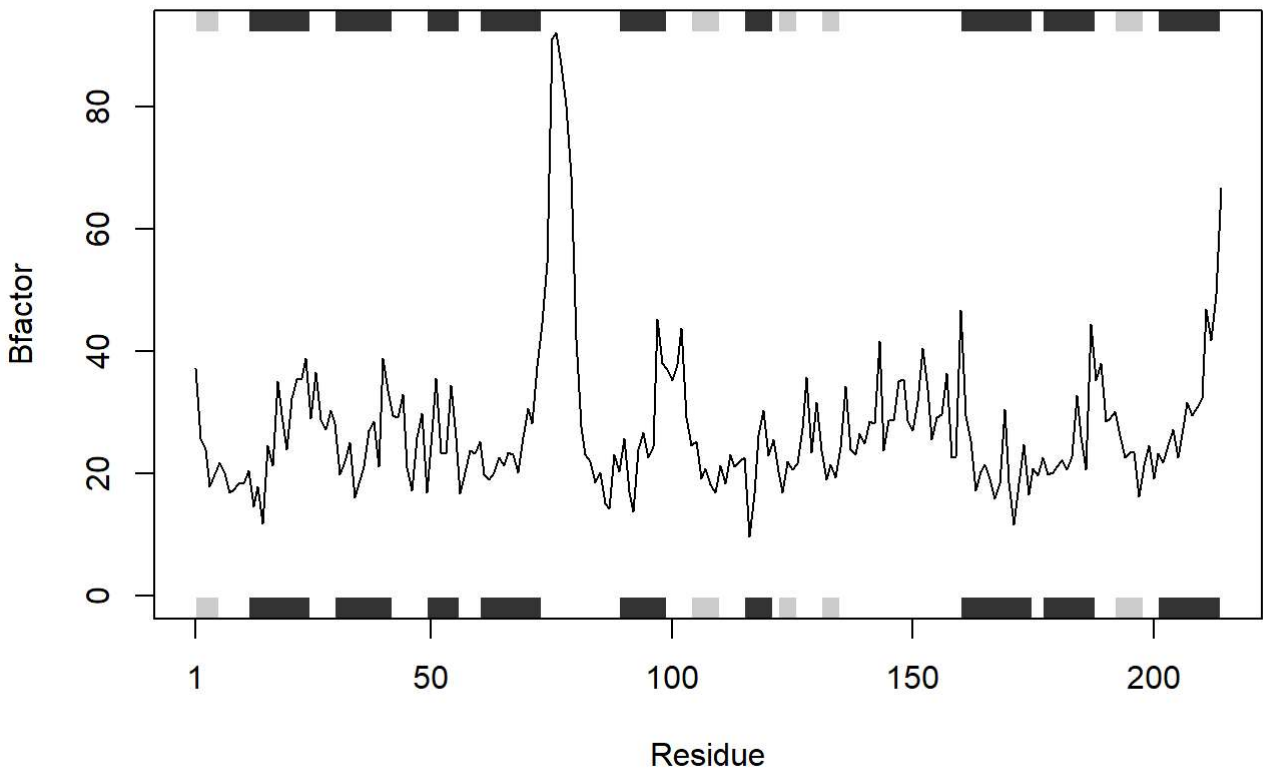
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

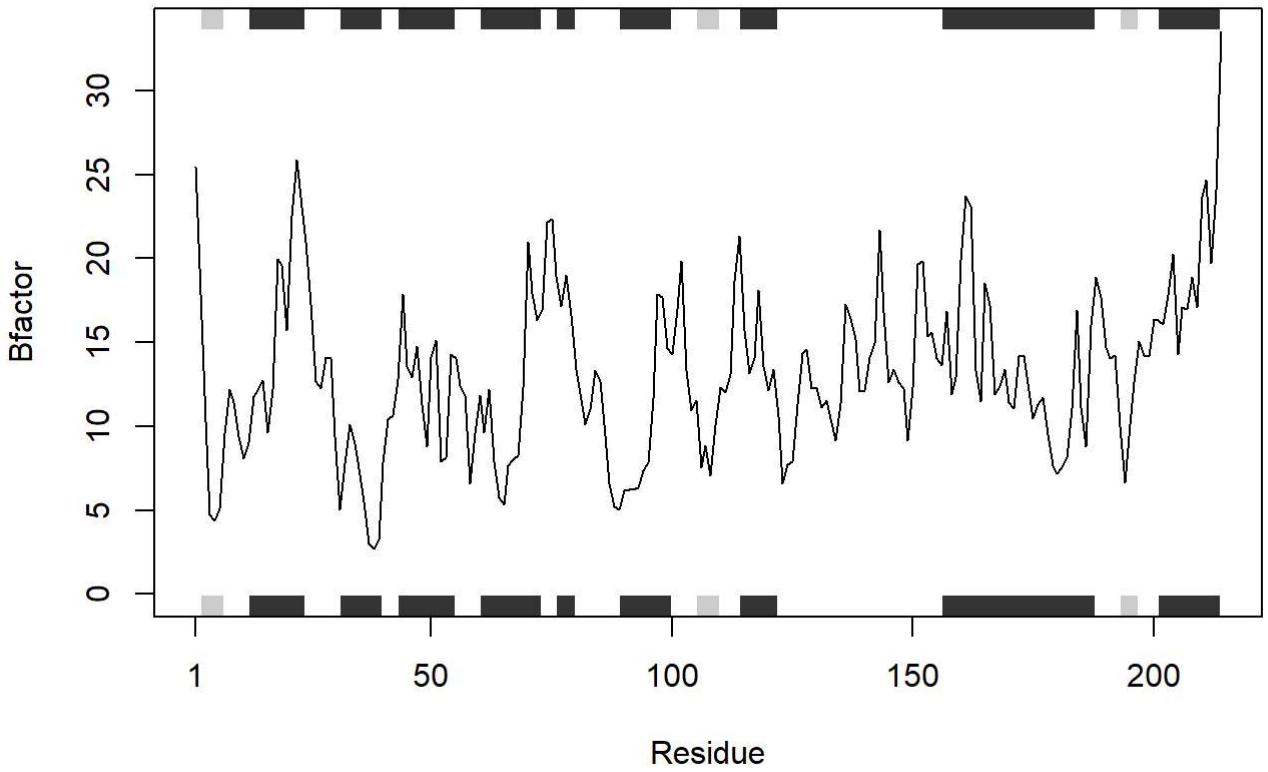
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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




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



> Question 6. How would you generalize the following code to work with any set of input protien structures?

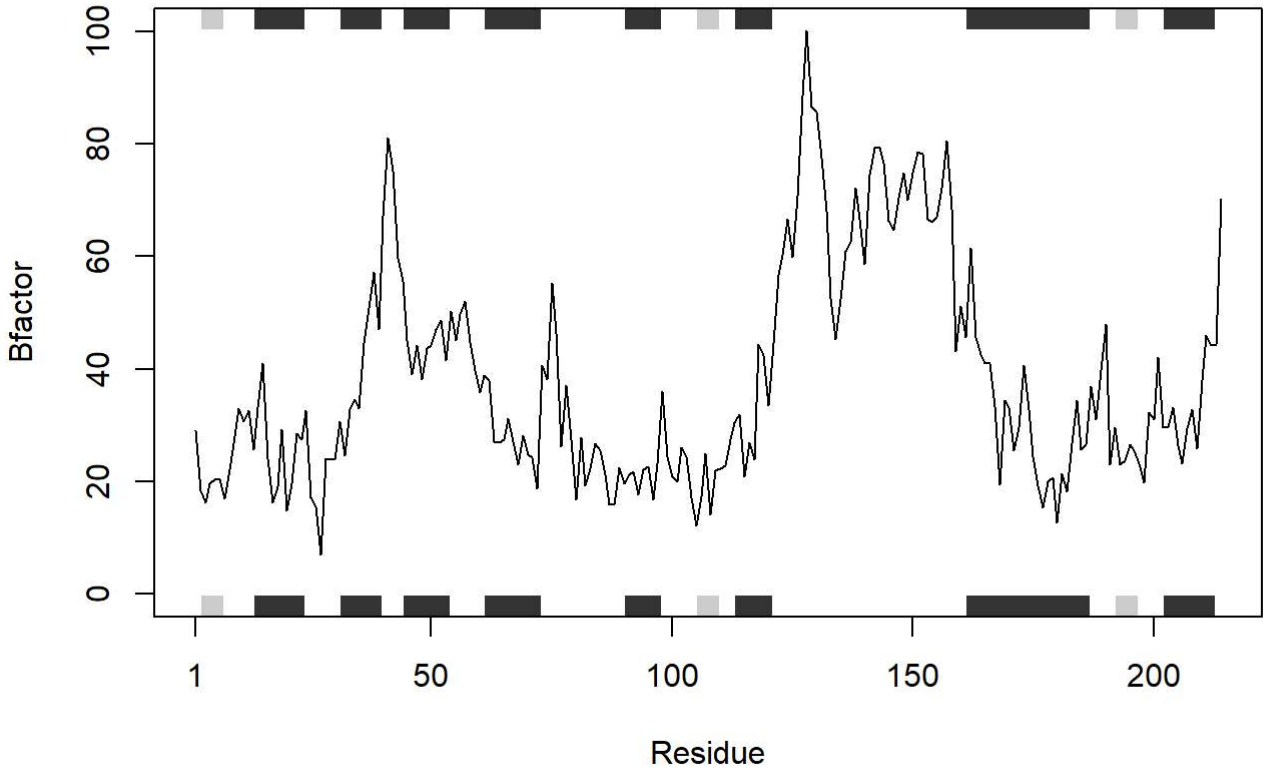
First, lets simplify the variable s1 to be x so we can substitute any variable for it in the future when our function is complete.

```
x <- s1
```

Now, lets continue to extrapolate s1 as x to generalize the whole code snippet.

```
x.chainA <- trim.pdb(x, chain="A", elety="CA")
x.b <- x.chainA$atom$b
```

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

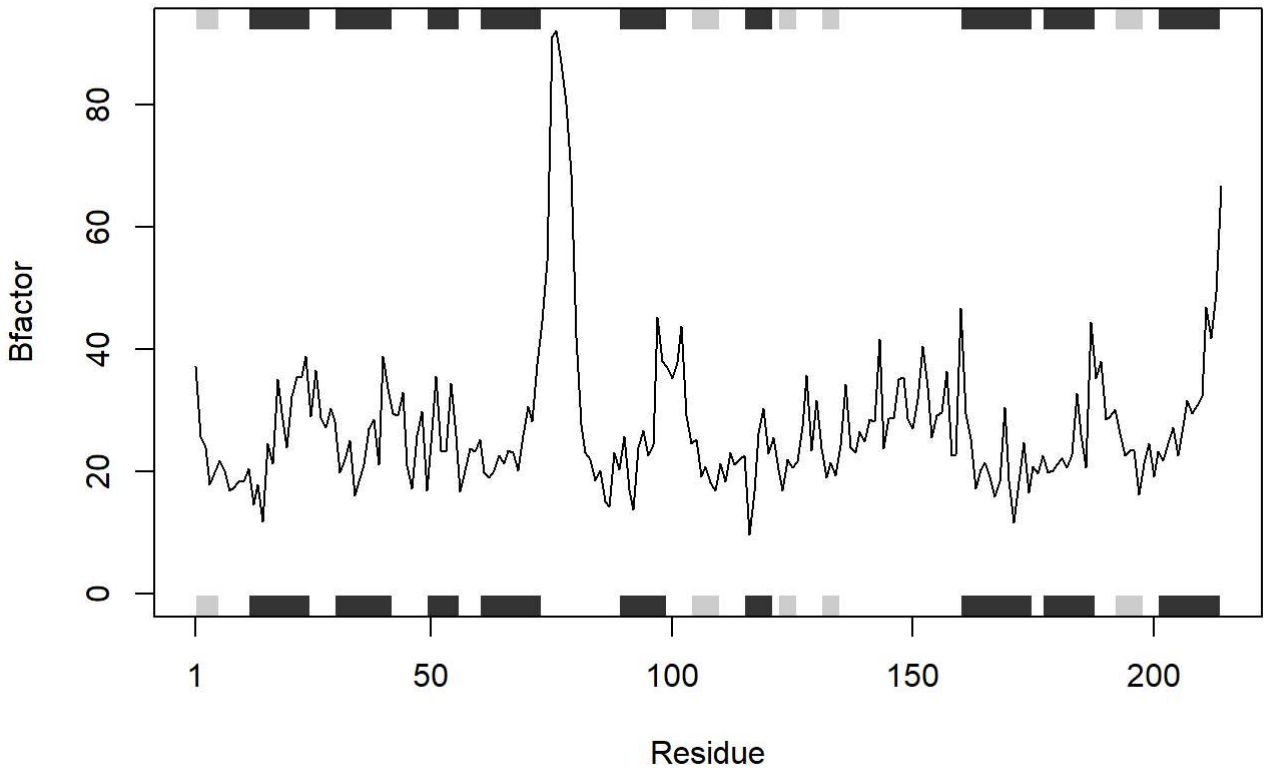


Now we can set up a function where x can be substituted for any protien.

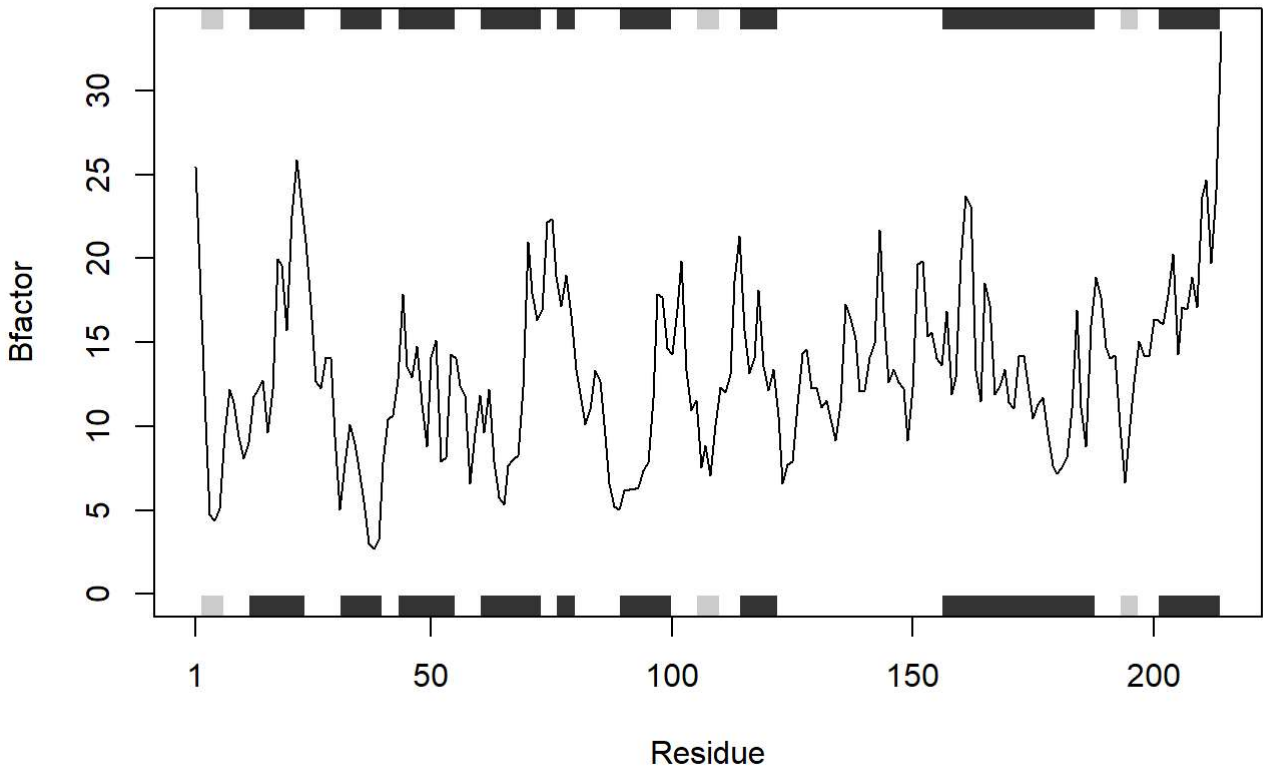
```
protien <- function(x){
  x.chainA <- trim.pdb(x, chain="A", elety="CA")
x.b <- x.chainA$atom$b
plotb3(x.b, sse=x.chainA, typ="l", ylab="Bfactor")
}
```

Now we can test out our function with s2 and s3!

```
protien(s2)
```




```
protien(s3)
```



Wonderful