20231021_Class06HW_SB

Savannah Bogus A69027475

Section 1:

A

```
#making a function that will apply the formula that section A uses to a vector, x
Rescale<-function(x){
  rng<-range(x)
  (x-rng[1])/(rng[2]-rng[1])
}
#testing
Rescale(1:10)
```

```
 [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
 [8] 0.7777778 0.8888889 1.0000000
```

```
#defining the dataframe
df <- data.frame(a=1:10, b=seq(200,400,length=10),c=11:20,d=NA)
```

```
#applying the function to the dataframe provided
apply(df,2,Rescale)
```

```
              a         b         c  d
 [1,] 0.0000000 0.0000000 0.0000000 NA
 [2,] 0.1111111 0.1111111 0.1111111 NA
 [3,] 0.2222222 0.2222222 0.2222222 NA
 [4,] 0.3333333 0.3333333 0.3333333 NA
 [5,] 0.4444444 0.4444444 0.4444444 NA
 [6,] 0.5555556 0.5555556 0.5555556 NA
 [7,] 0.6666667 0.6666667 0.6666667 NA
 [8,] 0.7777778 0.7777778 0.7777778 NA
 [9,] 0.8888889 0.8888889 0.8888889 NA
[10,] 1.0000000 1.0000000 1.0000000 NA
```

B

Can I improve this code?

```r
library(bio3d)
#kinase with no drug
s1<-read.pdb("4AKE")
```

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

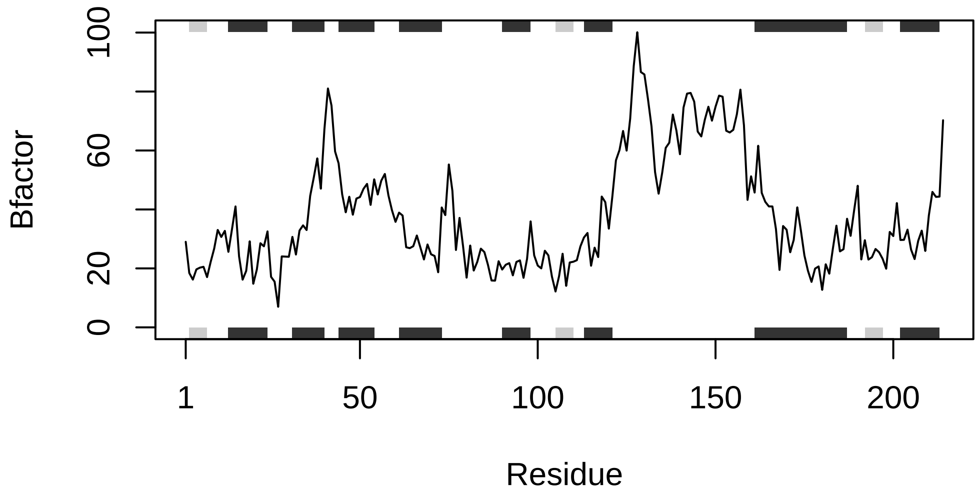
```r
#kinase no drug
s2<-read.pdb("1AKE")
```

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

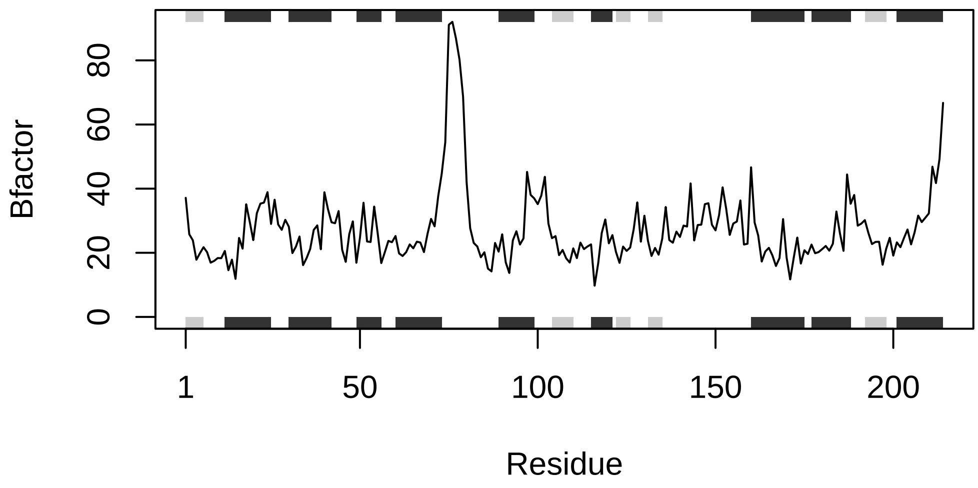
```r
#kinase with drug
s3<-read.pdb("1E4Y")
```

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

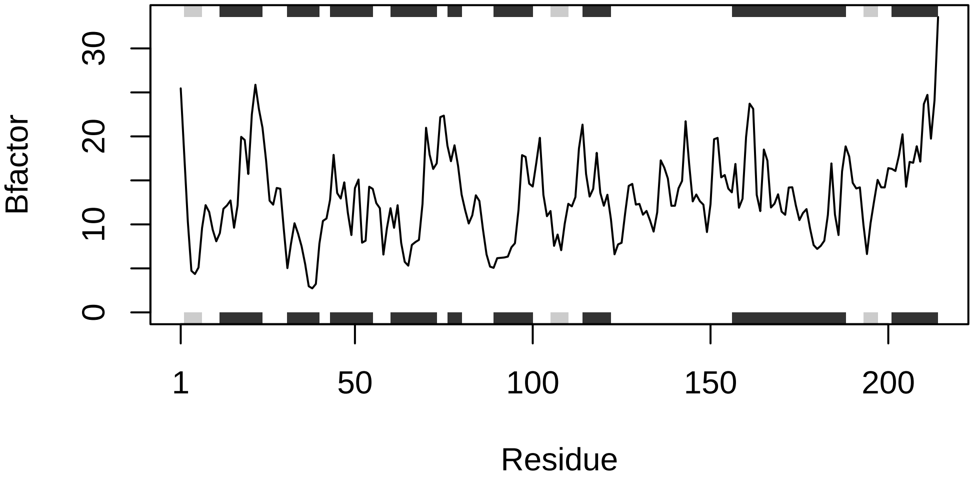
```r
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

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")
```



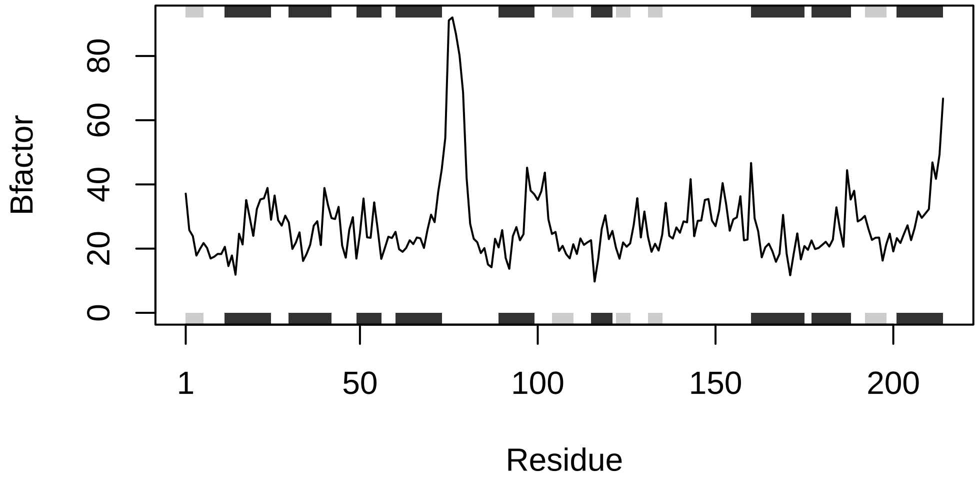
Inititally, yes, I can improve the code by getting rid of errors.

But, I can also improve the code by writing a function that does each of the 4 parts associated with the code that was written. There's a code to read from pdb for the protein, there's a code to trim, there's a code to get just the b factor, and there's a code to plot, so my function should also have these four parts. Something that seems obvious but that I was not understanding well is that I need to define each piece as we go so that I can call it later as the function progresses to combine each part later.

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

}
HWfunct("1AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/mk/4xp631s90rl36bqn0tcbxyf40000gn/T//RtmpuAeTTL/1AKE.pdb exists.
Skipping download

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



“‘