lab 6 homework

Shivani Lakkaraju

first install the bio3d package in the console only:

```
# install.packages("bio3d")
```

Load the required package

```
library(bio3d)
```

This function reads a Protein Data Bank (PDB) file, trims it to a specified chain and atom type, extracts B-factor values, and generates a B-factor plot and trimmed PDB object containing only the specified chain and atom type as the *output*. The *inputs* arguments are:

- pdb_id = The PDB identifier (e.g., "4AKE") of the protein structure to analyze.
- chain = The chain identifier within the PDB to analyze (default is "A").
- elety = The atom type to consider for analysis (default is "CA" for alpha carbon).
- ylab = Label for the y-axis in the plot (default is "B-factor").

```
analyze_protein_pdb <- function(pdb_id, chain = "A", elety = "CA",
  ylab = "B-factor") {

  # Step 1: Read the PDB File
  pdb <- read.pdb(pdb_id)
```

```
  # Step 2: Trim the PDB to Specified Chain and Atom Type
  pdb_chain <- trim.pdb(pdb, chain = chain, elety = elety)

  # Step 3: Extract B-factors
  b_factors <- pdb_chain$atom$b

  # Step 4: Plot B-factors
  plot_title <- paste("B-factor Plot for", pdb_id)
  plotb3(b_factors, sse = pdb_chain, typ = "l", ylab = ylab, main = plot_title)

  # Return the Trimmed PDB Object
  return(pdb_chain)
}
```

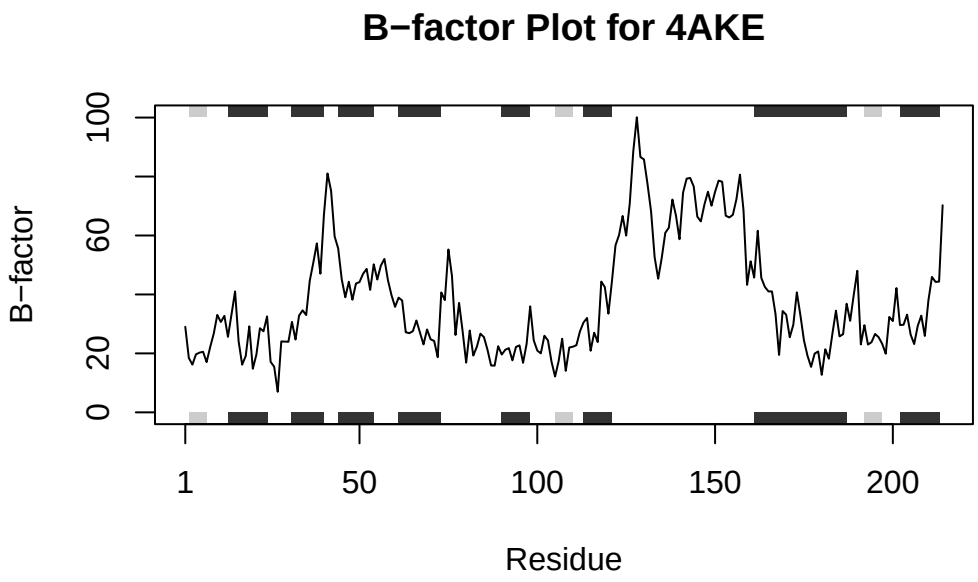
To analyze multiple PDB IDs simultaneously and with less copy/paste errors, we will create a vector of PDB IDs that we want to analyze with the above function:

```
pdb_ids <- c("4AKE", "1AKE", "1E4Y", "2AKE", "3AKE")
# Add more IDs as required
```

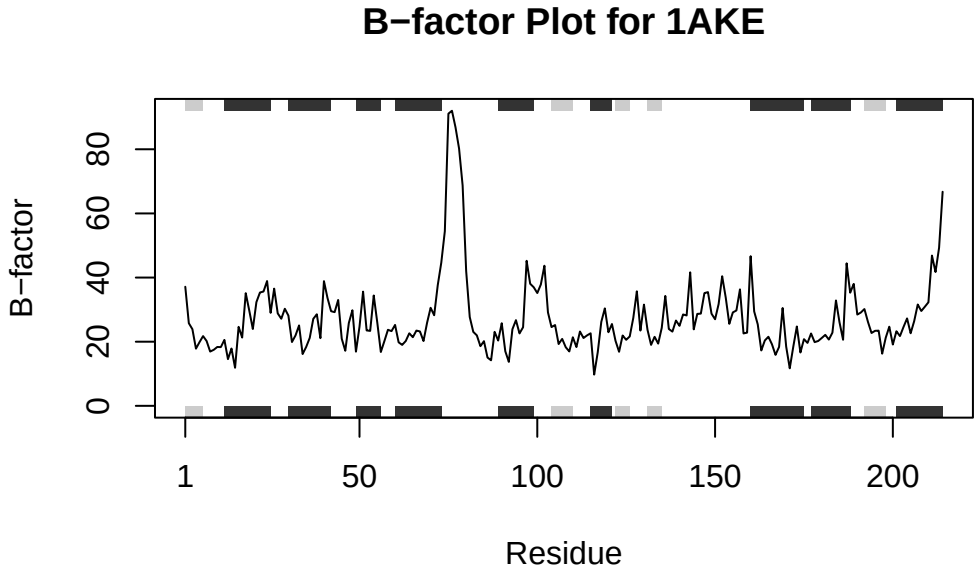
Then, we apply the function to each PDB ID via vectorization using the 'lapply()' function!

```
results <- lapply(pdb_ids, function(pdb) {
  analyze_protein_pdb(
    pdb_id = pdb,
    chain = "A",
    elety = "CA",
    ylab = "B-factor"
  )
})
```

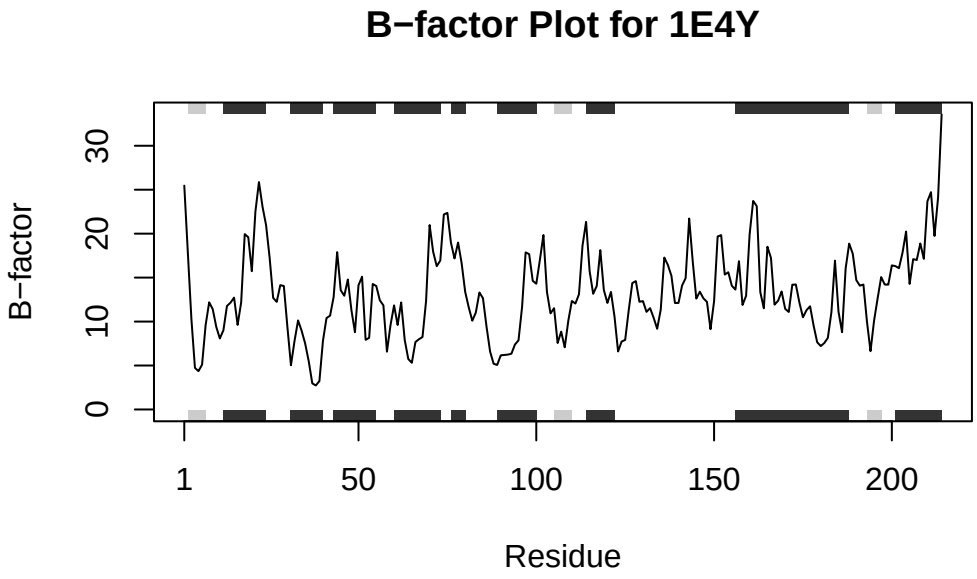
Note: Accessing on-line PDB file



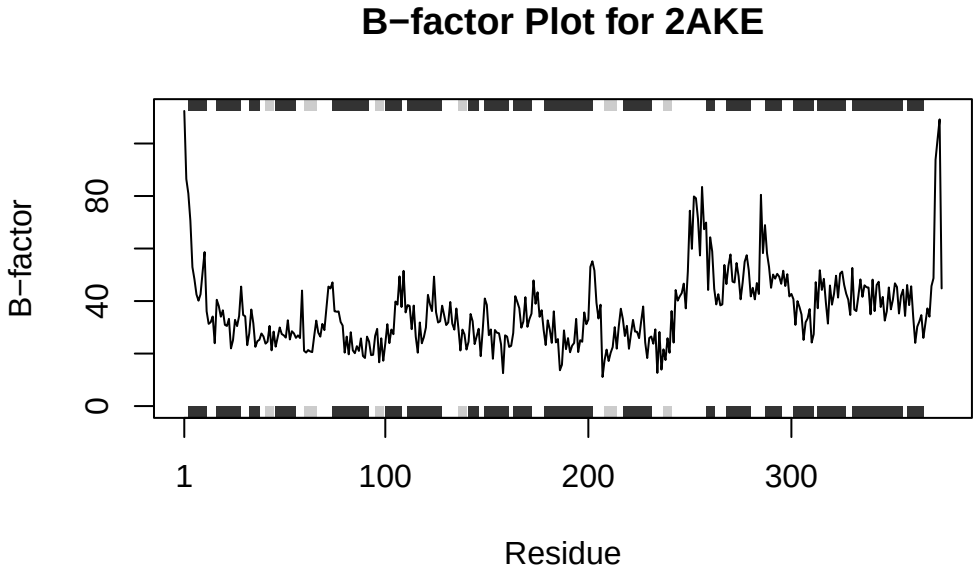

```
Note: Accessing on-line PDB file
 PDB has ALT records, taking A only, rm.alt=TRUE
```



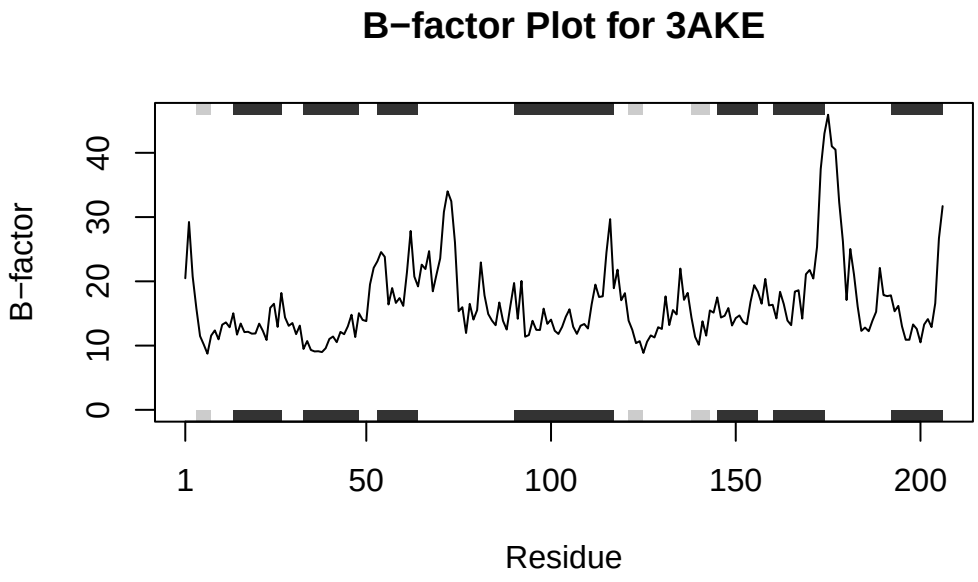
```
Note: Accessing on-line PDB file
```




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



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



Assign names to the list elements corresponding to PDB IDs:

```
names(results) <- pdb_ids
```