Programming Assigment 1

The folder contains 332 comma-separated-value (CSV) files containing pollution monitoring data for fine particulate matter (PM) air pollution at 332 locations in the United States. Each file contains data from a single monitor and the ID number for each monitor is contained in the file name. For example, data for monitor 200 is contained in the file "200.csv". Each file contains three variables:

Date: the date of the observation in YYYY-MM-DD format (year-month-day) sulfate: the level of sulfate PM in the air on that date (measured in micrograms per cubic meter) nitrate: the level of nitrate PM in the air on that date (measured in micrograms per cubic meter)

Taking a peek look at the data

```
sample_csv <- read.table("./specdata/002.csv", header=TRUE,sep=",")
head(sample_csv)
```

```
##         Date sulfate nitrate ID
## 1 2001-01-01      NA      NA  2
## 2 2001-01-02      NA      NA  2
## 3 2001-01-03      NA      NA  2
## 4 2001-01-04      NA      NA  2
## 5 2001-01-05      NA      NA  2
## 6 2001-01-06      NA      NA  2
```

A lot of NA values due to the data specificities

Part 1

Task 1:
Write a function named 'pollutantmean' that calculates the mean of a pollutant (sulfate or nitrate) across a specified list of monitors. The function 'pollutantmean' takes three arguments: 'directory', 'pollutant', and 'id'. Given a vector monitor ID numbers, 'pollutantmean' reads that monitors' particulate matter data from the directory specified in the 'directory' argument and returns the mean of the pollutant across all of the monitors, ignoring any missing values coded as NA. A prototype of the function is as follows

A little bit of testing

```
file_names <- dir("./specdata",full.names = TRUE) # getting the list of files
tables <- lapply(file_names,read.csv) # creating a dataframe
df <- do.call(rbind, tables)
ids_df <- subset(df, ID %in% c(1,2) & (!is.na(df$sulfate)), select = c("sulfate", "ID"))
mean(ids_df[["sulfate"]])
```

```
## [1] 4.402199
```

Creating function:

```
pollutantmean <- function(directory, pollutant, id = 1:332){
  ## 'directory' specifies the location of csv files
  ## 'pollutant' is character vector, either "sulfate" or "nitrate"
  ## 'id' is an integer vector of all ids to be used

  # function returns means of pollutant across all listed monitors (ignoring the NAs)

  file_names <- dir(paste("./",directory, sep=''),full.names = TRUE) # getting the list of files
  tables <- lapply(file_names,read.csv) # creating a list of tables
  df <- do.call(rbind, tables) # combining tables into one df

  ids_df <- subset(df, (ID %in% id) & (!is.na(df[[pollutant]])), select = c(pollutant, "ID"))  #select
  mean(ids_df[[pollutant]])
}
```

```
pollutantmean("specdata", "nitrate", 23)
```

```
## [1] 1.280833
```

```
pollutantmean("specdata", "nitrate", 70:72)
```

```
## [1] 1.706047
```

```
pollutantmean("specdata", "sulfate", 1:10)
```

```
## [1] 4.064128
```

The function seems to work well!
We can also save it to the file and then download it like this:

```
source("pollutantmean.R")
pollutantmean("specdata", "sulfate", 1:10)
```

```
## [1] 4.064128
```

Part 2

Task 2: Write a function that reads a directory full of files and reports the number of completely observed cases in each data file. The function should return a data frame where the first column is the name of the file and the second column is the number of complete cases.

For convinience, we'll make a new function called read_data to load all tables into one dataframe

```
read_data<-function(directory){
  file_names <- dir(paste("./",directory, sep=''),full.names = TRUE) # getting the list of files
  tables <- lapply(file_names,read.csv) # creating a list of tables
  df <- do.call(rbind, tables) # combining tables into one df
  df
}
```

```r
complete <- function(directory, id = 1:332){
  ## 'directory' specifies the location of csv files
  ## 'id' is an integer vector of all ids to be used

  # function returns the dataframe of the following structure:
  ## id nobs
  ## 1 117
  ## 2 1041

  ## where nobs -- the number of totally observed cases

  source("read_data.R")
  df <- read_data("specdata") ## reading data

  result = NULL # variable for a future dataframe with results

  for (i in id){
    ids_df <- subset(df, (ID == i) & (!is.na(df[["nitrate"]])) & (!is.na(df[["sulfate"]])), select = c(
    nobs<-nrow(ids_df) # getting the length of subset df
    result <- rbind(result,data.frame(i, nobs)) # saving results
  }
  result
}
```

```r
complete("specdata", 1)
```

```
##   i nobs
## 1 1  117
```

```r
complete("specdata", c(2, 4, 8, 10, 12))
```

```
##    i nobs
## 1  2 1041
## 2  4  474
## 3  8  192
## 4 10  148
## 5 12   96
```

```r
complete("specdata", 30:25)
```

```
##    i nobs
## 1 30  932
## 2 29  711
## 3 28  475
## 4 27  338
## 5 26  586
## 6 25  463
```

```r
complete("specdata", 3)
```

```
##   i nobs
## 1 3  243
```

All working right!

Part 3

Task 3: Write a function that takes a directory of data files and a threshold for complete cases and calculates the correlation between sulfate and nitrate for monitor locations where the number of completely observed cases (on all variables) is greater than the threshold. The function should return a vector of correlations for the monitors that meet the threshold requirement. If no monitors meet the threshold requirement, then the function should return a numeric vector of length 0.

```
corr <- function(directory, threshold=0){
  ## 'directory' specifies the location of csv files
  ## 'threshold' numeric value

  # function returns the list of correlations for ids that have nobs higher that threshold
  file_names <- dir(paste("./",directory, sep=''),full.names = TRUE) # getting the list of files
  tables <- lapply(file_names,read.csv) # creating a list of tables

  source("complete.R")
  nobs_df <- complete("specdata") ## getting the number of nods
  ids <- subset(nobs_df, nobs>threshold, select="i") ## selecting the ids for which nobs is higher than
  results <- vector("list", nrow(ids)) # empty vector for future results
  for (i in 1:nrow(ids)){
    temp<-data.frame(tables[ids[i,]]) #saving the corresponding table
    results[i]<-cor(temp$sulfate, temp$nitrate, use="complete.obs")
  }
  results

}
```

```
cr<-corr("specdata", 400)
head(cr)
```

```
## [[1]]
## [1] -0.01895754
##
## [[2]]
## [1] -0.04389737
##
## [[3]]
## [1] -0.06815956
##
## [[4]]
## [1] -0.07588814
##
## [[5]]
## [1] 0.7631288
##
## [[6]]
## [1] -0.1578286
```