The Data Set: specdata

The Data set 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)

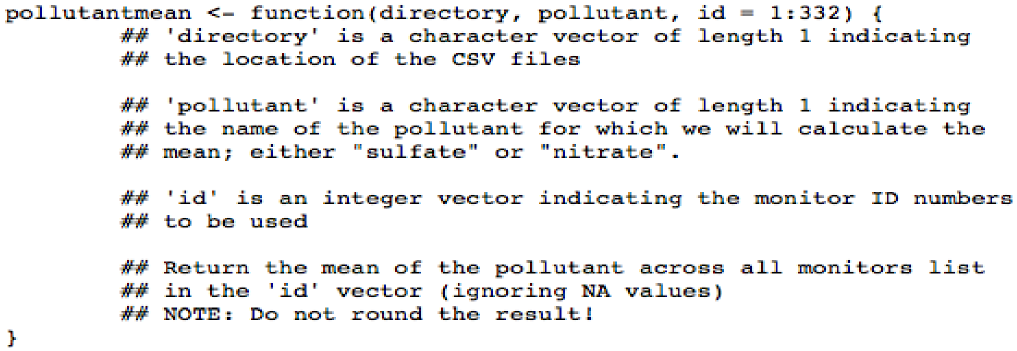
Problem Statement 1

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.

Arguments of the pollutantmean function:

- -directory: path that leads the location where the .csv files have been stored.
- -pollutant: the pollutant(sulfate/nitrate) whose mean we are interested to calculate across the selected .csv files
- -id: It is an integer vector where each element corresponds to the individual .csv file. Its default value has been set to cover all the 332 files of the dataset.

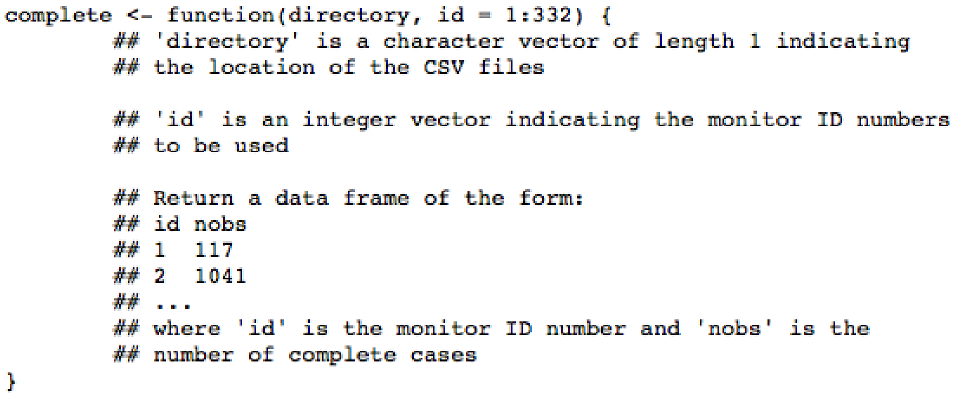
A prototype of the function should look as follows –



```
pollutantmean <- function(directory, pollutant, id = 1:332) {
        ## 'directory' is a character vector of length 1 indicating
        ## the location of the CSV files

        ## 'pollutant' is a character vector of length 1 indicating
        ## the name of the pollutant for which we will calculate the
        ## mean; either "sulfate" or "nitrate".

        ## 'id' is an integer vector indicating the monitor ID numbers
        ## to be used

        ## Return the mean of the pollutant across all monitors list
        ## in the 'id' vector (ignoring NA values)
        ## NOTE: Do not round the result!
}
```

Problem Statement 2

Write a function that reads through the given data set and generates the total number of complete cases in each of the files. A prototype of the Function should look as follows –



```
complete <- function(directory, id = 1:332) {
        ## 'directory' is a character vector of length 1 indicating
        ## the location of the CSV files

        ## 'id' is an integer vector indicating the monitor ID numbers
        ## to be used

        ## Return a data frame of the form:
        ## id nobs
        ## 1  117
        ## 2  1041
        ## ...
        ## where 'id' is the monitor ID number and 'nobs' is the
        ## number of complete cases
}
```

Problem Statement 3

Write a function that reads through the given dataset looking for monitor locations which exceed a given threshold number of complete cases. For monitor locations for which the number of complete cases exceeds the threshold limit, the function should calculate the correlation between sulfate and nitrate for monitor locations. 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. A prototype of the function should look as follows -

```
corr <- function(directory, threshold = 0) {
        ## 'directory' is a character vector of length 1 indicating
        ## the location of the CSV files

        ## 'threshold' is a numeric vector of length 1 indicating the
        ## number of completely observed observations (on all
        ## variables) required to compute the correlation between
        ## nitrate and sulfate; the default is 0

        ## Return a numeric vector of correlations
        ## NOTE: Do not round the result!
}
```