CSE463 Introduction to Bioinformatics

Spring'17

Assignment 2 Due: Mar 31

You are allowed to collaborate with other students provided that you do not exchange any written code. Write the solution on your own and mention the names of students you have collaborated with. Failure to do so may earn you a zero in this assignment.

For this first programming assignment you will write three functions that are meant to interact with dataset that accompanies this assignment. The dataset is contained in a zip file `specdata.zip` attached with the assignment.

Data

The `specdata.zip` file 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)

For this programming assignment you will need to unzip this file and create the directory '`specdata`'. Once you have unzipped the zip file, do not make any modifications to the files in the '`specdata`' directory. In each file you'll notice that there are many days where either sulfate or nitrate (or both) are missing (coded as `NA`). This is common with air pollution monitoring data in the United States.

Part 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 monitor's 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

```
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)

}
```

Some example output from this function is shown below.

```
source("pollutantmean.R")
pollutantmean("specdata", "sulfate", 1:10)
[1] 4.064
pollutantmean("specdata", "nitrate", 70:72)
[1] 1.706
pollutantmean("specdata", "nitrate", 23)
[1] 1.281
```

The function that you write should be able to match this output. Please save your code to a file named `pollutantmean.R`.

Part 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. A prototype of this function 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

}
```

Some example output from this function is shown below.

```
source("complete.R")
complete("specdata", 1)
##   id nobs
## 1  1  117
complete("specdata", c(2, 4, 8, 10, 12))
##   id nobs
## 1  2 1041
## 2  4  474
## 3  8  192
## 4 10  148
## 5 12   96
```

```
complete("specdata", 30:25)
##   id nobs
## 1 30  932
## 2 29  711
## 3 28  475
## 4 27  338
## 5 26  586
## 6 25  463
complete("specdata", 3)
##   id nobs
## 1  3  243
```

The function that you write should be able to match this output. Please save your code to a file named `complete.R`.

Part 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. A prototype of this function follows

For this function you will need to use the '`cor`' function in R which calculates the correlation between two vectors. Please read the help page for this function via '`?cor`' and make sure that you know how to use it.

```
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

}
```

Some example output from this function is shown below.

```
source("corr.R")
source("complete.R")
cr <- corr("specdata", 150)
head(cr)
## [1] -0.01896 -0.14051 -0.04390 -0.06816 -0.12351 -0.07589
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.2110 -0.0500  0.0946  0.1250  0.2680  0.7630
```

```
cr <- corr("specdata", 400)
head(cr)
## [1] -0.01896 -0.04390 -0.06816 -0.07589  0.76313 -0.15783
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.1760 -0.0311  0.1000  0.1400  0.2680  0.7630
cr <- corr("specdata", 5000)
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##
length(cr)
## [1] 0
cr <- corr("specdata")
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.0000 -0.0528  0.1070  0.1370  0.2780  1.0000
length(cr)
## [1] 323
```

The function that you write should be able to match this output. Please save your code to a file named `corr.R`.