Problem Set 1

Martin Aragoneses

July 22, 2015

Introduction

This is an R Markdown document I've written for "Programming Assignment 1: Air Pollution" https://class.coursera.org/rprog-015/assignment/view?assignment_id=3. The folder "specdata" 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)

Part 1

The function named 'pollutantmean' calculates the mean of a pollutant (sulfate or nitrate) across a specified list of monitors, taking 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' (in our case `"/Users/Martin/Desktop/data/github/Data-Science-R/Coursera/2/specdata"`) argument and returns the mean of the pollutant across all of the monitors, ignoring any missing values coded as NA.

```
# First we set the working directory:
setwd("/Users/Martin/Desktop/data/github/Data-Science-R/Coursera/2")
```

```
pollutantmean <- function(directory, pollutant = "nitrate", 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" (both in the datasets).

  ## 'id' is an integer vector indicating the monitor ID numbers
  ## to be used (recall there are 322 of them). By default, we
  ## use all of them, but we could select just a subset.

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

  ## We create a list with all the files in the directory

  file_list <- as.character(list.files(directory)) # specdatafiles
  file_paths <- paste(directory, file_list, sep="/")
```

```
  ## We initialize where the raw and the clean data are going to be
  data_raw <- data.frame(Date = character(), sulfate = numeric(), nitrate = numeric(), ID = numeric())
  data_clean <- numeric()

  ## We go through a loop that reads the .csv's for each of the wanted IDs
  for(i in id) {

    data_raw <- rbind(data_raw, read.csv(file_paths[i], header = TRUE))
  }

  if(pollutant == "nitrate") {
     # Removing missing values of the variable wanted
      good <- !is.na(data_raw[, 3])
      data_clean <- data_raw[good, ]

      # And finally calculate its mean
      mean(data_clean[, 3])

  } else {
    good <- !is.na(data_raw[, 2])
    data_clean <- data_raw[good, ]

    # And finally calculate its mean
    mean(data_clean[, 2])
  }

}
        # Removing missing values of the variable wanted
       ### good <- !is.na(data_raw$pollutant)
       ### data_clean <- data_raw[good, ]

        # And finally calculate its mean
       ### mean(data_clean$pollutant)
```

And after saving the above as a file "pollutantmean.R", we can source it and call the function.

```
source("pollutantmean.R")

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

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

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

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

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

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

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.

```
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
  file_list <- as.character(list.files(directory)) # specdatafiles
  file_paths <- paste(directory, file_list, sep="/")


  numfiles <- length(file_list)
  data_raw <- data.frame(Date = character(), sulfate = numeric(), nitrate = numeric(), ID = numeric())

  x <- data.frame(ID = id, nobs = numeric(length(id)))
  good <- list()

  for(i in 1:max(id)) {

    good[i] <- sum(!is.na(read.csv(file_paths[i], header = TRUE)[, 2]) & !is.na(read.csv(file_paths[i],
    x[i, 2] <- good[i]
  }
  x <- data.frame(ID = id, nobs = x[id, 2])
  x
}
```

After saving the above as "complete.R", we can run:

```
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
```

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.

```
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 (use 'cor' function for two vectors)
        ## NOTE: Do not round the result!
    tres <- rep(threshold, times = 332)
    x <- complete(directory)$nobs > tres
    correls <- numeric()
    ids <- 1:332
    where <- ids[x]

    good <- list()

    ## Recall from the first function:

    file_list <- as.character(list.files(directory)) # specdatafiles
    file_paths <- paste(directory, file_list, sep="/")
```

```
    ## We initialize where the raw and the clean data are going to be
    data_raw <- data.frame(Date = character(), sulfate = numeric(), nitrate = numeric(), ID = numeric())
    data_clean <- numeric()

    ## And now we loop:

    for(i in where) {
        data_raw <- read.csv(file_paths[i], header = TRUE)
        good <- !is.na(data_raw[, 2]) & !is.na(data_raw[, 3])
        data_clean <- data_raw[good, ]
        correls <- c(correls, cor(data_clean[,c(2, 3)])[1, 2])
    }
    correls
}
```

And after saving the above as "corr.R" and sourcing it (along with "complete.R") we can run different cases:

```
source("complete.R")
source("corr.R")

cr <- corr("specdata", 150)
head(cr)
```

```
## [1] -0.01895754 -0.14051254 -0.04389737 -0.06815956 -0.12350667 -0.07588814
```

```
summary(cr)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.21060 -0.04999  0.09463  0.12530  0.26840  0.76310
```

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

```
## [1] -0.01895754 -0.04389737 -0.06815956 -0.07588814  0.76312884 -0.15782860
```

```
summary(cr)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.17620 -0.03109  0.10020  0.13970  0.26850  0.76310
```

```
cr <- corr("specdata", 5000)
summary(cr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##
```

```
length(cr)
```

```
## [1] 0
```

```
cr <- corr("specdata") ## Recall default threshold was 0
head(cr)
```

```
## [1] -0.22255256 -0.01895754 -0.14051254 -0.04389737 -0.06815956 -0.12350667
```

```
summary(cr)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.00000 -0.05282  0.10720  0.13680  0.27830  1.00000
```

```
length(cr)
```

```
## [1] 323
```