Assignment 1

andrew

11/8/2021

R AirPolution

Part 1

Function to get the mean for pollutants through various data sets in “specdata’

```
pollutantmean <- function(directory, pollutant, id = 1:332) {
        file_list <- list.files(path = directory)
        dataset <- data.frame()
        for (i in id) {
                temp_data <- read.csv(paste(directory, file_list[i], sep = ""))
                dataset <- rbind(dataset, temp_data)
        }

        mean(dataset[[pollutant]], na.rm = TRUE)
}
```

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

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

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

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

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

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

## Part 2

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 = specdata, id = 1:332) {
        file_list <- list.files(path = directory)
        dataset <- data.frame()
        for (i in id) {
                i_data <- read.csv(paste(directory, file_list[i], sep = ""))
                i_nobs <- c(i,nrow(i_data[complete.cases(i_data),]))
                dataset <- rbind(dataset, i_nobs)
                colnames(dataset) <- c("id", "nobs")
        }
        dataset
}
```

```
complete(id = 1)
```

```
##   id nobs
## 1  1  117
```

```
complete(id = c(2, 4, 8, 10, 12))
```

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

```
complete(id = 30:25)
```

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

```
complete(id = 3)
```

```
##   id nobs
## 1  3  243
```

/ ## Part 3 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 return as vector of correlations for the monitors that meet the threshold requirement. If no monitors meet the threshold requirement, then the function returns a numeric vector of length 0.

```
corr <- function(directory = specdata, threshold = 0){
        id = 1:332
        file_list <- list.files(path = directory)
        dataset <-  c()
        for (i in id) {
                temp_data <- read.csv(paste(directory, file_list[i], sep = ""))
                clean_data <- temp_data[complete.cases(temp_data),]
                if (nrow(clean_data) > threshold){
                        dataset <- c(dataset, cor(clean_data$nitrate, clean_data$sulfate))
                }
        }
        dataset
}
```

```
test <- corr(threshold = 150)
head(test)
```

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

```
summary(test)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.21057 -0.04999  0.09463  0.12525  0.26844  0.76313
```

```
test <- corr(threshold = 400)
head(test)
```

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

```
summary(test)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.17623 -0.03109  0.10021  0.13969  0.26849  0.76313
```

```
test <- corr(threshold = 5000)
summary(test)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```
length(test)
```

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

```
test <- corr()
summary(test)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.00000 -0.05282  0.10718  0.13684  0.27831  1.00000
```

```
length(test)
```

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

Assignment 1 Quiz

/

1.

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

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

2.

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

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

3.

```
pollutantmean(specdata, "sulfate", 34)
```

```
## [1] 1.477143
```

4.

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

```
## [1] 1.702932
```

5.

```r
cc <- complete(specdata, c(6, 10, 20, 34, 100, 200, 310))
print(cc$nobs)
```

```
## [1] 228 148 124 165 104 460 232
```

6.

```r
cc <- complete(specdata, 54)
print(cc$nobs)
```

```
## [1] 219
```

7.

```r
RNGversion("3.5.1")
```

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```r
set.seed(42)
cc <- complete(specdata, 332:1)
use <- sample(332, 10)
print(cc[use, "nobs"])
```

```
##  [1] 711 135  74 445 178  73  49   0 687 237
```

8.

```r
cr <- corr(specdata)
cr <- sort(cr)
RNGversion("3.5.1")
```

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```r
set.seed(868)
out <- round(cr[sample(length(cr), 5)], 4)
print(out)
```

```
## [1]  0.2688  0.1127 -0.0085  0.4586  0.0447
```

9.

```r
cr <- corr(specdata, 129)
cr <- sort(cr)
n <- length(cr)
RNGversion("3.5.1")
```

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```
set.seed(197)
out <- c(n, round(cr[sample(n, 5)], 4))
print(out)
```

```
## [1] 243.0000   0.2540   0.0504  -0.1462  -0.1680   0.5969
```

10.

```
cr <- corr(specdata, 2000)
n <- length(cr)
cr <- corr(specdata, 1000)
cr <- sort(cr)
print(c(n, round(cr, 4)))
```

```
## [1]  0.0000 -0.0190  0.0419  0.1901
```