

R_Programming Assgn1

The point of this assignment is to know how to do a simple for loop, combining all the files into a same file, subsetting the data with conditions you wish, and then return the results.

```
library(data.table)

#Example: Create a blank vector, and duplicate it.

nums <- c(2,4,6,8)
dup <- numeric() #the blank one
for (num in nums) {
  dup<- c(dup,nums) #the new dup= blank dup + nums
}

#part 1

pollutantmean <- function(directory, pollutant, id= 1:332) {

  filelist<- list.files(path=directory, pattern= ".csv", full.names=TRUE)#Read the list of files,
  values<- numeric() #Create a blank numeric vector, create before the for loop

  for (i in id){
    data<-read.csv(filelist[i]) # read the filelist
    values <- c(values, data[[pollutant]]) # add the value into the blank vector you created before
  }
  mean(values,na.rm=TRUE)
}

pollutantmean("/users/andrewhu/desktop/Coursera/specdata","sulfate")

## [1] 3.189369
pollutantmean("specdata", "sulfate", 1:10)

## [1] 4.064128
pollutantmean("specdata", "nitrate", 70:72)

## [1] 1.706047
pollutantmean("specdata", "sulfate", 34)

## [1] 1.477143
pollutantmean("specdata", "nitrate")

## [1] 1.702932

#part2

#Making examples
data<- read.csv("specdata/001.csv")
```

```

#get the number of complete cases
sum(complete.cases(data)) #117 complete cases in 001.csv

## [1] 117

#Checking the length of filelist
filelist<- list.files(path="specdata", pattern=".csv",full.names = TRUE)
length(filelist) # check to get all the files

## [1] 332

complete<- function(directory, id= 1:332) {
  filelist<-list.files(path=directory,pattern=".csv", full.names = TRUE) #list all the files
  nob<- numeric() #create a empty vector

  for (i in id) {
    data<- read.csv(filelist[i])
    nob<- c(nob,sum(complete.cases(data))) #add value into the original blank vector, nob
  }
  nob #return the nob
  data.frame(id,nob) #Create a simple df
}

complete("specdata",1:10)

##      id nob
## 1     1 117
## 2     2 1041
## 3     3 243
## 4     4 474
## 5     5 402
## 6     6 228
## 7     7 442
## 8     8 192
## 9     9 275
## 10    10 148

cc <- complete("specdata", c(6, 10, 20, 34, 100, 200, 310))
print(cc$nob)

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

cc <- complete("specdata", 54)
print(cc$nob)

## [1] 219

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

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

#partIII
corr<- function(directory, threshold=0) {

  filelist= list.files(path =directory, pattern =".csv", full.names = TRUE )#create a filelist

```

```

    dat<- numeric() # blank numeric vector

    for (i in 1:length(filelist)) {
      temp<- read.csv(filelist[i]) #reading files
      temp<- temp[complete.cases(temp),] #subsetting complete cases
      nrow<- nrow(temp) #counting the rows
      if (nrow > threshold) {
        dat<- c(dat, cor(temp$sulfate,temp$nitrate))    } #if nrow> threshold then return the correlati
      }
    }
  }

cr <- corr("specdata")
cr <- sort(cr)
set.seed(868)
out <- round(cr[sample(length(cr), 5)], 4)
print(out)

## [1] 0.2688 0.1127 -0.0085 0.4586 0.0447

cr <- corr("specdata")
cr <- sort(cr)
set.seed(868)
out <- round(cr[sample(length(cr), 5)], 4)
print(out)

## [1] 0.2688 0.1127 -0.0085 0.4586 0.0447

cr <- corr("specdata", 129)
cr <- sort(cr)
n <- length(cr)
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

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

```