Lab2

m.pryidun

2022-11-11

##Task 2. Working with files ##Data The zip file specdata.zip [2.4MB] containing the data can be downloaded from data folder is course repository. The 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) In each file 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 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.You can see some example output from this function below. pollutantmean(“specdata”, “sulfate”, 1:10) [1] 4.064128 pollutantmean(“specdata”, “nitrate”, 70:72) [1] 1.706047 pollutantmean(“specdata”, “nitrate”, 23) [1] 1.280833

pollutantmean <- function(directory, pollutant, id = 1:332) {  
 means <- c()  
   
 for(monitor in id){  
 path <- paste(getwd(), "/", directory, "/", sprintf("%03d", monitor), ".csv", sep = "")  
 data <- read.csv(path)  
 selected\_data <- data[pollutant]  
 means <- c(means, selected\_data[!is.na(selected\_data)])  
 }  
   
 mean(means)  
}  
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 named complete 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. You can see some example output from this function below.

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 <- function(directory, id = 1:332){  
 results <- data.frame(id=numeric(0), nobs=numeric(0))  
 for(monitor in id){  
 path <- paste(getwd(), "/", directory, "/", sprintf("%03d", monitor), ".csv", sep = "")  
 data <- read.csv(path)  
 selected\_data <- data[(!is.na(data$sulfate)), ]  
 selected\_data <- selected\_data[(!is.na(selected\_data$nitrate)), ]  
 nobs <- nrow(selected\_data)  
 results <- rbind(results, data.frame(id=monitor, nobs=nobs))  
 }  
 results  
}  
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

##Part 3 Write a function named corr 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 re- quirement. If no monitors meet the threshold requirement, then the function should return a numeric vector of length 0. For this function you will need to use the ‘cor’ function in R which calculates the correlation between two vectors.

corr <- function(directory, threshold = 0){  
 cor\_results <- numeric(0)  
   
 complete\_cases <- complete(directory)  
 complete\_cases <- complete\_cases[complete\_cases$nobs>=threshold, ]  
   
 if(nrow(complete\_cases)>0){  
 for(monitor in complete\_cases$id){  
 path <- paste(getwd(), "/", directory, "/", sprintf("%03d", monitor), ".csv", sep = "")  
 data <- read.csv(path)  
 selected\_data <- data[(!is.na(data$sulfate)), ]  
 selected\_data <- selected\_data[(!is.na(selected\_data$nitrate)), ]  
 sulfate <- selected\_data["sulfate"]  
 nitrate <- selected\_data["nitrate"]  
 cor\_results <- c(cor\_results, cor(sulfate, nitrate))  
 }  
 }  
 cor\_results  
}  
corr("specdata")

## [1] -0.222552561 -0.018957541 -0.140512544 -0.043897372 -0.068159562  
## [6] -0.123506666 -0.075888144 -0.159673652 -0.086841940 0.161379334  
## [11] 0.763128837 -0.078813785 -0.545372688 -0.222748447 -0.178085309  
## [16] -0.461271662 -0.157828603 -0.223464778 -0.156998919 -0.111576389  
## [21] -0.044898818 0.050363116 0.117249264 0.259057178 0.133274607  
## [26] 0.366201078 0.580751264 0.006863930 0.726693888 0.057741676  
## [31] 0.115338086 0.465754012 0.515804375 0.412693537 0.375631176  
## [36] 0.315725317 0.244560561 0.594426499 0.553514976 0.469712603  
## [41] 0.614340566 -0.297694992 0.181938817 0.460513619 0.405022501  
## [46] 0.432921867 0.434789780 0.374124729 0.088421364 0.118136697  
## [51] -0.091022820 -0.033091304 0.440660466 -0.029683708 0.268525390  
## [56] 0.277220958 -0.049108453 0.322627410 0.091139374 -0.025750053  
## [61] 0.120521602 -0.061746831 0.041306963 -0.146202136 0.243249343  
## [66] -0.162485185 -0.097254393 0.089262856 0.290081750 0.381281107  
## [71] 0.568403991 0.711864008 0.442047154 0.268203237 0.190644585  
## [76] 0.227222983 0.229238882 0.005635506 0.010352587 0.018628108  
## [81] 0.115582197 -0.034698733 -0.064750174 0.096614297 0.236536500  
## [86] 0.002864405 0.107184775 0.128477284 -0.042533572 0.246440304  
## [91] -0.137041337 0.211610424 0.136609030 0.118975253 -0.010010172  
## [96] 0.098073855 0.066928310 0.100212474 -0.063984344 0.058257528  
## [101] -0.067907770 -0.109445392 -0.066525489 -0.129245884 -0.111066409  
## [106] 0.195026555 0.068983843 -0.089441210 -0.114090325 -0.106280702  
## [111] -0.176855164 -0.116984680 0.019138583 0.100643502 -0.073858484  
## [116] 0.036665921 -0.107957809 0.487760251 -0.167955891 0.296744105  
## [121] 0.347421569 0.146528765 0.362414577 0.093330832 0.198915192  
## [126] 0.214695603 0.164602262 0.180626975 0.297391960 0.400610675  
## [131] 0.176508543 0.139158631 0.231984399 0.275465461 0.550891819  
## [136] 0.227615918 -1.000000000 0.275903634 0.299630040 0.248143145  
## [141] 0.298344178 -0.056325366 -0.178114558 0.002032940 -0.022802183  
## [146] 0.304689837 -0.001202233 0.085217423 -0.076409023 0.010021716  
## [151] 0.016411646 -0.038785934 -0.075297768 0.041917773 -0.019090637  
## [156] 0.193324040 0.347127724 0.596929143 0.064760688 0.113596590  
## [161] -0.220905861 1.000000000 -0.181659703 -0.143750037 -0.017703373  
## [166] 0.284905360 0.305506111 0.150031306 0.673240137 0.134895077  
## [171] 0.172850003 0.286076203 -0.106687748 0.244744168 0.911838775  
## [176] 0.337120085 0.424798956 0.095921881 0.022899033 0.143330735  
## [181] 0.087196218 0.408741028 0.425176879 0.361728434 -0.035090337  
## [186] -0.082388453 0.524933220 -0.094742313 -0.087573726 -0.060405837  
## [191] -0.092398269 -0.183197353 0.124650112 -0.053001162 -0.039911536  
## [196] 0.010158287 0.451828854 0.295793699 0.615268727 -0.075214053  
## [201] 0.132207405 0.089547098 -0.019086127 -0.137259840 -0.045552626  
## [206] -0.055514425 0.044727139 -0.009589442 0.211599525 -0.073972834  
## [211] 0.125764712 -0.133770703 0.112668377 0.138387891 -0.003207550  
## [216] -0.052643174 0.042168144 -0.067460173 -0.030882797 0.017805647  
## [221] 0.026138073 -0.050287543 0.016535643 0.585781324 0.199919014  
## [226] 0.695524534 0.482158286 0.355110474 0.589606340 0.368038099  
## [231] -0.029094866 -0.074495323 -0.288270835 0.262101561 0.892224353  
## [236] 0.116029371 0.202651963 0.072781519 -0.005386993 0.258826380  
## [241] 0.144110820 0.101915017 0.023020993 0.074594252 0.256665139  
## [246] 0.162401158 -0.003454405 0.190141976 0.184581239 0.120596460  
## [251] 0.711555731 -0.176233152 -0.144699131 0.147074115 0.273520382  
## [256] 0.153333302 0.109557323 -0.092863394 -0.212957125 -0.182752126  
## [261] 0.261566498 -0.008836513 0.356592359 0.253286446 -0.089133895  
## [266] -0.017185129 -0.156323514 -0.042538204 0.010235676 -0.009912754  
## [271] -0.042910367 -0.210567709 -0.155957816 0.717309882 NA  
## [276] NA 0.046211272 NA -0.060808231 0.184083723  
## [281] -0.008546646 -0.081160145 0.017684029 -0.037848590 -0.123781156  
## [286] NA 0.160865053 0.191467036 NA 0.127708002  
## [291] NA NA NA NA -0.081211759  
## [296] 0.263104105 0.756946045 0.785964505 0.615095781 0.598343330  
## [301] 0.506535631 0.191834811 -0.024723462 -0.110563022 -0.150627164  
## [306] -0.002500089 -0.166201361 0.225537295 0.619349867 0.531380642  
## [311] 0.345317319 0.520115665 0.466673962 0.518820173 0.394191512  
## [316] 0.213529393 0.260561149 0.379446208 0.562418583 -0.123172036  
## [321] -0.061565518 -0.180133963 0.458643619 0.728610818 0.253978075  
## [326] 0.139867175 0.316429404 0.268780500 0.279397143 0.267260662  
## [331] 0.287133842 0.146211226