Homework3_IDS

justyn koenig

10/17/2021

Question 1

```
set.seed(12) # to be reproducible
A = matrix(data = runif(n = 1:500), nrow = 50, ncol = 10)
colnames(A) = paste("lake", 1:10, sep = "_")
data1 <- data
data1
```

```
## function (..., list = character(), package = NULL, lib.loc = NULL,
##     verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## {
##     fileExt <- function(x) {
##         db <- grepl("\\.[^.]+\\.(gz|bz2|xz)$", x)
##         ans <- sub(".*\\.", "", x)
##         ans[db] <- sub(".*\\.([^.]+\\.)(gz|bz2|xz)$", "\\1\\2",
##             x[db])
##         ans
##     }
##     my_read_table <- function(...) {
##         lcc <- Sys.getlocale("LC_COLLATE")
##         on.exit(Sys.setlocale("LC_COLLATE", lcc))
##         Sys.setlocale("LC_COLLATE", "C")
##         read.table(...)
##     }
##     names <- c(as.character(substitute(list(...))[-1L]), list)
##     if (!is.null(package)) {
##         if (!is.character(package))
##             stop("’package’ must be a character string or NULL")
##         if (FALSE) {
##             if (any(package %in% "base"))
##                 warning("datasets have been moved from package ’base’ to package ’datasets’")
##             if (any(package %in% "stats"))
##                 warning("datasets have been moved from package ’stats’ to package ’datasets’")
##             package[package %in% c("base", "stats")] <- "datasets"
##         }
##     }
##     paths <- find.package(package, lib.loc, verbose = verbose)
##     if (is.null(lib.loc))
##         paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##             paths)
##     paths <- unique(normalizePath(paths[file.exists(paths)]))
##     paths <- paths[dir.exists(file.path(paths, "data"))]
##     dataExts <- tools:::.make_file_exts("data")
##     if (length(names) == 0L) {
##         db <- matrix(character(), nrow = 0L, ncol = 4L)
##         for (path in paths) {
##             entries <- NULL
##             packageName <- if (file_test("-f", file.path(path,
##                 "DESCRIPTION")))
##                 basename(path)
##             else "."
##             if (file_test("-f", INDEX <- file.path(path, "Meta",
##                 "data.rds"))) {
##                 entries <- readRDS(INDEX)
##             }
##             else {
##                 dataDir <- file.path(path, "data")
##                 entries <- tools::list_files_with_type(dataDir,
##                   "data")
##                 if (length(entries)) {
##                   entries <- unique(tools::file_path_sans_ext(basename(entries)))
##                   entries <- cbind(entries, "")
##                 }
##             }
##             if (NROW(entries)) {
##                 if (is.matrix(entries) && ncol(entries) == 2L)
##                   db <- rbind(db, cbind(packageName, dirname(path),
##                     entries))
##                 else warning(gettextf("data index for package %s is invalid and will be ignored",
##                   sQuote(packageName)), domain = NA, call. = FALSE)
##             }
##         }
##         colnames(db) <- c("Package", "LibPath", "Item", "Title")
##         footer <- if (missing(package))
##             paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRUE))")),
##                 "\n", "to list the data sets in all *available* packages.")
##         else NULL
##         y <- list(title = "Data sets", header = NULL, results = db,
##             footer = footer)
##         class(y) <- "packageIQR"
##         return(y)
##     }
##     paths <- file.path(paths, "data")
##     for (name in names) {
##         found <- FALSE
##         for (p in paths) {
##             tmp_env <- if (overwrite)
##                 envir
##             else new.env()
##             if (file_test("-f", file.path(p, "Rdata.rds"))) {
##                 rds <- readRDS(file.path(p, "Rdata.rds"))
##                 if (name %in% names(rds)) {
##                   found <- TRUE
##                    if (verbose)
##                      message(sprintf("name=%s:\t found in Rdata.rds",
##                        name), domain = NA)
##                    thispkg <- sub(".*/([^/]*)/data$", "\\1", p)
##                    thispkg <- sub("_.*$", "", thispkg)
##                    thispkg <- paste0("package:", thispkg)
##                    objs <- rds[[name]]
##                    lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                      filter = function(x) x %in% objs)
##                    break
##                  }
##                  else if (verbose)
##                    message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s\n",
##                      name, paste(names(rds), collapse = ",")),
##                      domain = NA)
##              }
##              if (file_test("-f", file.path(p, "Rdata.zip"))) {
##                  warning("zipped data found for package ", sQuote(basename(dirname(p))),
##                    ".\nThat is defunct, so please re-install the package.",
##                    domain = NA)
##                  if (file_test("-f", fp <- file.path(p, "filelist")))
##                    files <- file.path(p, scan(fp, what = "", quiet = TRUE))
##                  else {
##                    warning(gettextf("file 'filelist' is missing for directory %s",
##                      sQuote(p)), domain = NA)
##                    next
##                  }
##              }
##              else {
##                  files <- list.files(p, full.names = TRUE)
##              }
##              files <- files[grep(name, files, fixed = TRUE)]
##              if (length(files) > 1L) {
##                  o <- match(fileExt(files), dataExts, nomatch = 100L)
##                  paths0 <- dirname(files)
##                  paths0 <- factor(paths0, levels = unique(paths0))
##                  files <- files[order(paths0, o)]
##              }
##              if (length(files)) {
##                  for (file in files) {
##                    if (verbose)
##                      message("name=", name, ":\t file= ...", .Platform$file.sep,
##                        basename(file), "::\t", appendLF = FALSE,
##                        domain = NA)
##                    ext <- fileExt(file)
##                    if (basename(file) != paste0(name, ".", ext))
##                      found <- FALSE
##                    else {
##                      found <- TRUE
##                      zfile <- file
##                      zipname <- file.path(dirname(file), "Rdata.zip")
##                      if (file.exists(zipname)) {
##                        Rdatadir <- tempfile("Rdata")
##                        dir.create(Rdatadir, showWarnings = FALSE)
##                         topic <- basename(file)
##                         rc <- .External(C_unzip, zipname, topic,
##                           Rdatadir, FALSE, TRUE, FALSE, FALSE)
##                         if (rc == 0L)
##                           zfile <- file.path(Rdatadir, topic)
##                       }
##                       if (zfile != file)
##                         on.exit(unlink(zfile))
##                       switch(ext, R = , r = {
##                         library("utils")
##                         sys.source(zfile, chdir = TRUE, envir = tmp_env)
##                       }, RData = , rdata = , rda = load(zfile,
##                         envir = tmp_env), TXT = , txt = , tab = ,
##                         tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                         txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                           header = TRUE, as.is = FALSE), envir = tmp_env),
##                         CSV = , csv = , csv.gz = , csv.bz2 = ,
##                         csv.xz = assign(name, my_read_table(zfile,
##                           header = TRUE, sep = ";", as.is = FALSE),
##                           envir = tmp_env), found <- FALSE)
##                     }
##                     if (found)
##                       break
##                   }
##                   if (verbose)
##                     message(if (!found)
##                       "*NOT* ", "found", domain = NA)
##               }
##               if (found)
##                   break
##           }
##           if (!found) {
##               warning(gettextf("data set %s not found", sQuote(name)),
##                   domain = NA)
##           }
##           else if (!overwrite) {
##               for (o in ls(envir = tmp_env, all.names = TRUE)) {
##                   if (exists(o, envir = envir, inherits = FALSE))
##                     warning(gettextf("an object named %s already exists and will not be overwritten",
##                       sQuote(o)))
##                   else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                     envir = envir)
##               }
##               rm(tmp_env)
##           }
##       }
##       invisible(names)
## }
## <bytecode: 0x000000001646f0d8>
## <environment: namespace:utils>
```

```
for(i in 1:ncol(A)){
  print(mean(A[, i]))
}
```

```
## [1] 0.4601492
## [1] 0.4992815
## [1] 0.5987037
## [1] 0.4580486
## [1] 0.4719578
## [1] 0.4965216
## [1] 0.5110536
## [1] 0.4577936
## [1] 0.5193423
## [1] 0.4856413
```

```
colMeans(A)
```

```
##    lake_1    lake_2    lake_3    lake_4    lake_5    lake_6    lake_7    lake_8
## 0.4601492 0.4992815 0.5987037 0.4580486 0.4719578 0.4965216 0.5110536 0.4577936
##    lake_9   lake_10
## 0.5193423 0.4856413
```

Question 2

```
x = array(1:27, dim = c(3, 3, 3))
apply(X = x, MARGIN = c(1, 2),
      FUN = paste, collapse = ", ")
```

```
##      [,1]        [,2]        [,3]
## [1,] "1, 10, 19" "4, 13, 22" "7, 16, 25"
## [2,] "2, 11, 20" "5, 14, 23" "8, 17, 26"
## [3,] "3, 12, 21" "6, 15, 24" "9, 18, 27"
```

```
x = array(1:27, dim = c(3, 3, 3))
y = array(0, dim=c(3, 3, 3))
for (i in 1:3){
  for (j in 1:3) {
    for (k in 1:3) {
      y[i,j,] = paste(x[i,j,],collapse = ", ")
    }
  }
}
y
```

```
## , , 1
##
##      [,1]        [,2]        [,3]
## [1,] "1, 10, 19" "4, 13, 22" "7, 16, 25"
## [2,] "2, 11, 20" "5, 14, 23" "8, 17, 26"
## [3,] "3, 12, 21" "6, 15, 24" "9, 18, 27"
##
## , , 2
##
```

```
##      [,1]        [,2]        [,3]
## [1,] "1, 10, 19" "4, 13, 22" "7, 16, 25"
## [2,] "2, 11, 20" "5, 14, 23" "8, 17, 26"
## [3,] "3, 12, 21" "6, 15, 24" "9, 18, 27"
##
## , , 3
##
##      [,1]        [,2]        [,3]
## [1,] "1, 10, 19" "4, 13, 22" "7, 16, 25"
## [2,] "2, 11, 20" "5, 14, 23" "8, 17, 26"
## [3,] "3, 12, 21" "6, 15, 24" "9, 18, 27"
```

Question 3

```
N <- 30
for (i in 1:30){
  N[1] = 0
  N[2] = 1
  N[i+2] = N[i]+N[i+1]
}
N
```

```
##  [1]       0       1       1       2       3       5       8      13      21
## [10]      34      55      89     144     233     377     610     987    1597
## [19]    2584    4181    6765   10946   17711   28657   46368   75025  121393
## [28]  196418  317811  514229  832040 1346269
```

Question 4

```
top105 = readLines("http://www.textfiles.com/music/ktop100.txt")
top105 = top105[-c(64, 65)] # missing No. 54 and 55
```

```
pattern <- "^[0-9]*\\.*[0-9]\\."
library("stringr")
A = str_extract(top105,pattern)
A
```

```
##   [1] NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
##  [11] "1."  "2."  "3."  "4."  "5."  "6."  "7."  "8."  "9."  "10."
##  [21] "11." "12." "13." "14." "15." "16." "17." "18." "19." "20."
##  [31] "21." "22." "23." "24." "25." "26." "27." "28." "29." "30."
##  [41] "31." "32." "33." "34." "35." "36." "37." "38." "39." "40."
##  [51] "41." "42." "43." "44." "45." "46." "47." "48." "49." "50."
##  [61] "51." "52." "53." "56." "57." "58." "59." "60." "61." "62."
##  [71] "63." "64." "65." "66." "67." "68." "69." "70." "71." "72."
##  [81] "73." "74." "75." "76." "77." "78." "79." "80." "81." "82."
##  [91] "83." "83." "84." "85." "86." "87." "88." "89." "90." "91."
## [101] "91." "92." "93." "94." "95." "96." "97." "97." "98." "99."
## [111] "100." "101." "102." "103." "104." "105." "105." NA     NA     NA
## [121] NA
```

```
B = grep("^[0-9]*\\.*[0-9]\\.*[0-9]", A, value =TRUE)
B
```

```
##  [1] "10."  "11."  "12."  "13."  "14."  "15."  "16."  "17."  "18."  "19."
## [11] "20."  "21."  "22."  "23."  "24."  "25."  "26."  "27."  "28."  "29."
## [21] "30."  "31."  "32."  "33."  "34."  "35."  "36."  "37."  "38."  "39."
## [31] "40."  "41."  "42."  "43."  "44."  "45."  "46."  "47."  "48."  "49."
## [41] "50."  "51."  "52."  "53."  "56."  "57."  "58."  "59."  "60."  "61."
## [51] "62."  "63."  "64."  "65."  "66."  "67."  "68."  "69."  "70."  "71."
## [61] "72."  "73."  "74."  "75."  "76."  "77."  "78."  "79."  "80."  "81."
## [71] "82."  "83."  "83."  "84."  "85."  "86."  "87."  "88."  "89."  "90."
## [81] "91."  "91."  "92."  "93."  "94."  "95."  "96."  "97."  "97."  "98."
## [91] "99."  "100." "101." "102." "103." "104." "105." "105."
```

Question 5

```
C = as.numeric(B)
is.numeric(C)
```

```
## [1] TRUE
```

```
D <- gregexpr(pattern = "(\\d{1,3})", text = C)
E = regmatches(C, D)
C[duplicated(C, incomparables =FALSE)]
```

```
## [1]  83  91  97 105
```