hw_03.Rmd

Vinay K L

11 October, 2023

Q1

Find average using loop and colmeans

Solution

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

Using for loop

```r
#vector
num_lakes <- ncol(A)

#calculte average
average_values1 <- numeric(num_lakes)

#loop over it
for (i in 1:num_lakes) {
  average_values1[i] <- mean(A[, i])
}

#checking
print(average_values1)
```

```
##  [1] 0.4601492 0.4992815 0.5987037 0.4580486 0.4719578 0.4965216 0.5110536
##  [8] 0.4577936 0.5193423 0.4856413
```

Using colMeans()

```r
#vector
average_values2 <- colMeans(A)

#checking
print(average_values2)
```

```
##    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
```

Q2

Matrix using apply and nested loop

Solution

```r
# checking what's the results are first.
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"
```

```r
#get the array
x = array(1:27, dim = c(3, 3, 3))

#lets define how many rows and cols we need
nrows <- dim(x)[1]
ncols <- dim(x)[2]

#creating empty matrix to store the for loops result
matrix_c <- matrix(NA, nrow = nrows, ncol = ncols)

#looping over rows and cols
for (i in 1:nrows) {
  for (j in 1:ncols) {
    matrix_c[i, j] <- paste(x[i, j, ], collapse = ", ")
  }
}
print(matrix_c)
```

```
##      [,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"
```

Q3

Fibonacci sequence

Solution

```
## [1]       0       1       1       2       3       5       8      13      21      34
## [11]      55      89     144     233     377     610     987    1597    2584    4181
## [21]    6765   10946   17711   28657   46368   75025  121393  196418  317811  514229
```

Q4

Top 500 list and string

Solution

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

# I am combining the lines into single text since
#readLines is dragging the initial lines/headers and tail lines which also has numbers etc
text = paste(top105, collapse = "\n")

#extracting the ranking using gsub - removing all the other details associated with the ranking
ranking_numbers = gsub("(\\d+\\.)", "\\1",
                       unlist(regmatches(text,
                                         gregexpr("\\d+\\.", text))))

# removing empty strings if any
ranking_numbers = ranking_numbers[ranking_numbers != ""]

# checking
ranking_numbers
```

```
##    [1] "105."  "1991." "1."     "2."     "3."     "4."     "5."     "6."     "7."
##   [10] "8."    "9."     "10."    "11."    "12."    "13."    "14."    "15."    "16."
##   [19] "17."   "18."    "19."    "20."    "21."    "22."    "23."    "24."    "25."
##   [28] "26."   "27."    "28."    "29."    "30."    "31."    "32."    "33."    "34."
##   [37] "35."   "36."    "37."    "38."    "39."    "40."    "41."    "42."    "43."
##   [46] "44."   "45."    "46."    "47."    "48."    "49."    "50."    "51."    "52."
##   [55] "53."   "56."    "57."    "58."    "59."    "60."    "61."    "62."    "63."
##   [64] "64."   "65."    "66."    "67."    "68."    "69."    "70."    "71."    "72."
##   [73] "73."   "74."    "75."    "76."    "77."    "78."    "79."    "80."    "81."
##   [82] "82."   "83."    "83."    "84."    "85."    "86."    "87."    "88."    "89."
##   [91] "90."   "91."    "91."    "92."    "93."    "94."    "95."    "96."    "97."
##  [100] "97."   "98."    "99."    "100."   "101."   "102."   "103."   "104."   "105."
##  [109] "105."
```

Q 5

Remove traling . and find duplicates

Solution

```
#remove trailing dots
ranking_numbers <- sub("\\.$", "", ranking_numbers)

#checking
head(ranking_numbers, n =5)
```

```
## [1] "105"  "1991" "1"    "2"    "3"
```

```
#convert to numeric
ranking_numbers_nm <- as.numeric(ranking_numbers)

#more checking
head(ranking_numbers_nm, n =5)
```

```
## [1]  105 1991    1    2    3
```

```
#extracting duplicated rankings
duplicated_rankings <- ranking_numbers_nm[duplicated(ranking_numbers_nm)]

#final check
head(duplicated_rankings, n = 5)
```

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