Stat 291 - Recitation 3

Orçun Oltulu

12 / 11 / 2021

Exercise 1: Arrays

Exercise 1.1:

Part A.

Create the following array and name it as array1.

```
array1 <- array(c(seq(10,120,10),seq(-10,-120,-10)), dim = c(2,2,3,2),
                dimnames = list(c(),c(),c(),c("Positive","Negative")))
array1
```

```
## , , 1, Positive
##
##      [,1] [,2]
## [1,]   10   30
## [2,]   20   40
##
## , , 2, Positive
##
##      [,1] [,2]
## [1,]   50   70
## [2,]   60   80
##
## , , 3, Positive
##
##      [,1] [,2]
## [1,]   90  110
## [2,]  100  120
##
## , , 1, Negative
##
##      [,1] [,2]
## [1,]  -10  -30
## [2,]  -20  -40
##
## , , 2, Negative
##
##      [,1] [,2]
## [1,]  -50  -70
## [2,]  -60  -80
##
## , , 3, Negative
##
##      [,1] [,2]
## [1,]  -90 -110
## [2,] -100 -120
```

Part B.

Print only the ‘Negative’ blocks of array1.

```
array1[,,,"Negative"]
```

```
## , , 1
##
##      [,1] [,2]
## [1,]  -10  -30
## [2,]  -20  -40
##
## , , 2
##
##      [,1] [,2]
## [1,]  -50  -70
## [2,]  -60  -80
##
## , , 3
##
##      [,1] [,2]
## [1,]  -90 -110
## [2,] -100 -120
```

Part C.

Print the second layer of array1.

```
array1[,,2,]
```

```
## , , Positive
##
```

```
##      [,1] [,2]
## [1,]   50   70
## [2,]   60   80
##
## , , Negative
##
##      [,1] [,2]
## [1,]  -50  -70
## [2,]  -60  -80
```

Part D.

Print only the first columns of 'Positive' blocks, then find the mean of the elements inside this 2x3 matrix.

```
array1[,1,,1]
```

```
##      [,1] [,2] [,3]
## [1,]   10   50   90
## [2,]   20   60  100
```

```
mean(array1[,1,,1])
```

```
## [1] 55
```

Part E.

Multiply the 'Positive' and 'Negative' blocks in the first layer.

```
array1[,,1,"Positive"] %*% array1[,,1,"Negative"]
```

```
##       [,1]  [,2]
## [1,]  -700 -1500
## [2,] -1000 -2200
```

Exercise 1.2:

Part A.

Create an Array with 2 layers of 4x4 matrices whose elements are all zeros.

```
q1_array <- array(0, dim = c(4,4,2))
q1_array
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
## [4,]    0    0    0    0
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
## [4,]    0    0    0    0
```

Part B.

Now, change the elements in the first layer with the first 16 Letters in the alphabet and change the elements in the second layer with integers 1 to 16.

```r
q1_array[,,1] <- LETTERS[1:16]
q1_array[,,2] <- 1:16
q1_array
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,] "A"  "E"  "I"  "M"
## [2,] "B"  "F"  "J"  "N"
## [3,] "C"  "G"  "K"  "O"
## [4,] "D"  "H"  "L"  "P"
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,] "1"  "5"  "9"  "13"
## [2,] "2"  "6"  "10" "14"
## [3,] "3"  "7"  "11" "15"
## [4,] "4"  "8"  "12" "16"
```

Part C.

Name rows as {r.1, r.2, r.3, r.4}, columns as {c.1, c.2, c.3, c.4}

```r
rownames(q1_array) <- c("r.1","r.2","r.3","r.4")
colnames(q1_array) <- paste("c.", 1:4, sep="")
```

Part D.

Convert the second layer to numeric, and find the mean of the elements in it.

```
mean(as.numeric(q1_array[,,2]))
```

```
## [1] 8.5
```

Part E.

Replace diagonals of both layers with NA.

```
diag(q1_array[,,1]) <- rep(NA,4)
diag(q1_array[,,2]) <- rep(NA,4)
q1_array
```

```
## , , 1
##
##     c.1 c.2 c.3 c.4
## r.1 NA  "E" "I" "M"
## r.2 "B" NA  "J" "N"
## r.3 "C" "G" NA  "O"
## r.4 "D" "H" "L" NA
##
## , , 2
##
##     c.1 c.2 c.3  c.4
## r.1 NA  "5" "9"  "13"
## r.2 "2" NA  "10" "14"
## r.3 "3" "7" NA   "15"
## r.4 "4" "8" "12" NA
```

Exercise 2: Factors:

Exercise 2.1:

Part A.

Create the following vector,

{0,0,1,1,1,0,1,0,0,1,1}

Assume that 0 represents "No", 1 represents "Yes" for a specific question.

```
myvec <- c(0,0,1,1,1,0,1,0,1,1)
myvec
```

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

Part B.

Convert the vector into a factor object and save it in **myfactor** variable.

```
myfactor <- factor(myvec)
myfactor
```

```
##  [1] 0 0 1 1 1 0 1 0 1 1
## Levels: 0 1
```

Part C.

Now, instead of using 0 and 1, use No and Yes respectively for your factor object and create **myfactor2** variable.

```
myfactor2 <- factor(myvec, levels = 0:1, labels = c("No","Yes"))
myfactor2
```

```
##  [1] No  No  Yes Yes Yes No  Yes No  Yes Yes
## Levels: No Yes
```

Part D.

Print the total number of yes and no anwers.

```
table(myfactor2)
```

```
## myfactor2
##  No Yes
##   4   6
```

Part E.

Find the proportion of Yes and No answers.

```
table(myfactor2) / length(myfactor2)
```

```
## myfactor2
##  No Yes
## 0.4 0.6
```

Exercise 2.2:

Part A.

Create a 'letter_grades' vector,

```
letter_grades <- c("AA","BA","BB","CB","CC","DC","DD","FD","FF")
letter_grades
```

```
## [1] "AA" "BA" "BB" "CB" "CC" "DC" "DD" "FD" "FF"
```

Part B.

Assume that the following vector is the letter grades for 10 students and name it 'stud.grades'.

{"AA", "AA", "BB", "BA", "FF", "DC", "FD", "BB", "DD", "AA"}

```
stud.grades <- c("AA", "AA", "BB", "BA", "FF", "DC", "FD", "BB", "DD", "AA")
stud.grades
```

```
##  [1] "AA" "AA" "BB" "BA" "FF" "DC" "FD" "BB" "DD" "AA"
```

Part C.

Create a factor from stud.grades vector and name it 'grades_factor'.

```
grades_factor <- factor(stud.grades)
grades_factor
```

```
##  [1] AA AA BB BA FF DC FD BB DD AA
## Levels: AA BA BB DC DD FD FF
```

Part D.

Assume that there was a mistake while assigning grades, replace the grade of 3rd student with "CC" and 10th student with "BA".

```
grades_factor[c(3,10)] <- c("CC","BA") # gives an error !!
# and assigns NA to the third student.
# because there is not a CC level for this factor.
grades_factor
```

```
##  [1] AA   AA   <NA> BA   FF   DC   FD   BB   DD   BA
## Levels: AA BA BB DC DD FD FF
```

```
# Let's add all grade levels to grades_factor
levels(grades_factor) <- letter_grades
```

```
grades_factor[c(3,10)] <- c("CC","BA")
grades_factor
```

```
##  [1] AA AA CC BA DD CB DC BB CC BA
## Levels: AA BA BB CB CC DC DD FD FF
```

Exercise 2.3:

Create a vector called agreement that contains the following values: {“Disagree”, “Neither agree or disagree”, “Somewhat agree”, “Agree”, “Strongly Agree”}.

Next, convert the vector to a factor variable called agreementfactor.

```
agreement = c("Disagree", "Neither agree or disagree",
              "Somewhat agree", "Agree", "Strongly Agree")
agreement
```

```
## [1] "Disagree"                  "Neither agree or disagree"
## [3] "Somewhat agree"            "Agree"
## [5] "Strongly Agree"
```

```
agreementfactor <- factor(agreement)
agreementfactor
```

```
## [1] Disagree                  Neither agree or disagree
## [3] Somewhat agree            Agree
## [5] Strongly Agree
## 5 Levels: Agree Disagree Neither agree or disagree ... Strongly Agree
```

Part B.

Sort the factor variable from disagree to strongly agree.

```
agreementfactor2 <- factor(agreement, levels = agreement, ordered = TRUE)
sort(agreementfactor2)
```

```
## [1] Disagree                  Neither agree or disagree
## [3] Somewhat agree            Agree
## [5] Strongly Agree
## 5 Levels: Disagree < Neither agree or disagree < Somewhat agree < ... < Strongly Agr
```

```
#agreementfactor2[2] < agreementfactor2[3]
```

Part C.

Assume that you asked 20 people “What do you think about the idea of pineapple on pizza? 1. Strongly Disaggree, ..., 3. No idea, ..., 5. Strongly Agree.”.

```
mysample <- c(1, 1, 3, 2, 1, 3, 1, 3, 4, 1, 1, 5, 3, 3, 1, 4, 1, 1, 1, 2)
```

You collected the data {1, 1, 3, 2, 1, 3, 1, 3, 4, 1, 1, 5, 3, 3, 1, 4, 1, 1, 1, 2}

Create a factor from the sample above without using any option.

```
myfactor <- factor(mysample)
myfactor
```

```
##  [1] 1 1 3 2 1 3 1 3 4 1 1 5 3 3 1 4 1 1 1 2
## Levels: 1 2 3 4 5
```

Part D.

Now, create another factor from the sample above but this time assign labels to your sample.

```
mysample_factor <- factor(mysample, labels = agreement, ordered = TRUE)
levels(mysample_factor)
```

```
## [1] "Disagree"                   "Neither agree or disagree"
## [3] "Somewhat agree"             "Agree"
## [5] "Strongly Agree"
```

```
sort(mysample_factor)
```

```
##  [1] Disagree                  Disagree
##  [3] Disagree                  Disagree
##  [5] Disagree                  Disagree
##  [7] Disagree                  Disagree
##  [9] Disagree                  Disagree
## [11] Neither agree or disagree Neither agree or disagree
## [13] Somewhat agree            Somewhat agree
## [15] Somewhat agree            Somewhat agree
## [17] Somewhat agree            Agree
## [19] Agree                     Strongly Agree
## 5 Levels: Disagree < Neither agree or disagree < Somewhat agree < ... < Strongly Agr
```

Part E.

Use 'table()' function to comment on the people's ideas about pineapple pizza.

```
table(mysample_factor)
```

```
## mysample_factor
##                  Disagree Neither agree or disagree            Somewhat agree
##                        10                         2                         5
##                     Agree            Strongly Agree
##                         2                         1
```

Exercise 3: Lists

Exercise 3.1:

Create a list name it as your name, and add some elements based on your personal info;

- First element in your list is simply a vector which consists of your age, your height, how many hours you slept last night.
- Second element is going to be related with your music taste; write down your favorite 3 bands/singers and their songs. Create a 3x2 matrix, on the columns you will have your favorite band and songs
- Third, create another vector in which you have the courses that you are enrolled for this semester. (e.g. "STAT291")

Mine looks like this, (please use the same names, 'Info', 'Music', 'Courses')

```
orcun <- list(Info = c("Age" = 24,"Height" = 172,"HoursSlept"= 5),
              Music = cbind(bands=c("Megadeath","Dio","Black Sabbath"),
                            songs=c("Symphony of Destruction","Killing the Dragon","Par
              Courses = c("STAT291"))
orcun
```

```
## $Info
##        Age     Height HoursSlept
##         24        172          5
##
## $Music
##      bands           songs
## [1,] "Megadeath"     "Symphony of Destruction"
## [2,] "Dio"           "Killing the Dragon"
## [3,] "Black Sabbath" "Paranoid"
##
## $Courses
## [1] "STAT291"
```

Part A.

Get the first element of your list.

```
orcun[[1]]
```

```
##        Age     Height HoursSlept
##         24        172          5
```

Part B.

Print your 'Age' and 'Height'

```
orcun$Info[c("Age","Height")]
```

```
##    Age Height
##     24    172
```

Part C.

Print only the 'Music' matrix.

```
orcun[["Music"]]
```

```
##      bands           songs
## [1,] "Megadeath"     "Symphony of Destruction"
## [2,] "Dio"           "Killing the Dragon"
## [3,] "Black Sabbath" "Paranoid"
```

```
# or
orcun$Music
```

```
##      bands           songs
## [1,] "Megadeath"     "Symphony of Destruction"
## [2,] "Dio"           "Killing the Dragon"
## [3,] "Black Sabbath" "Paranoid"
```

Part D.

Add one more brand/singer and song to your music matrix, and print the updated version.

```
orcun$Music <- rbind(orcun$Music, c("Rick Astley", "Never Gonna Give You Up"))
orcun$Music
```

```
##      bands           songs
## [1,] "Megadeath"     "Symphony of Destruction"
## [2,] "Dio"           "Killing the Dragon"
## [3,] "Black Sabbath" "Paranoid"
## [4,] "Rick Astley"   "Never Gonna Give You Up"
```

Part E.

Print the number of courses you are enrolled.

```
length(orcun$Courses)
```

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

Part F.

Add your 3 favorite food as a new element to your list and name that element as "Food".

```
orcun[["Food"]] <- c("Pizza","Hamburger","Baklava")
```

Part G.

Get "Food" in alphabetic order.

```r
orcun[["Food"]] <- sort(orcun[["Food"]])
orcun
```

```
## $Info
##         Age      Height HoursSlept 
##          24         172          5 
## 
## $Music
##      bands           songs                       
## [1,] "Megadeath"     "Symphony of Destruction"   
## [2,] "Dio"           "Killing the Dragon"        
## [3,] "Black Sabbath" "Paranoid"                  
## [4,] "Rick Astley"   "Never Gonna Give You Up"   
## 
## $Courses
## [1] "STAT291"
## 
## $Food
## [1] "Baklava"   "Hamburger" "Pizza"
```

Exercise 3.2:

Recall the example from previous recitation;

Consider you arrive to a store to buy some groceries and you want to group the items in your basket. There are mainly 3 groups in your list which are fruits, diary products and others.

There are banana, apple, peach and grape in your basket and they cost {12, 8, 15, 10} respectively.

Also, you bought Milk, eggs, yogurt and cheese which cost you {10, 15, 12, 25} respectively.

Finally, you got some candy, tooth paste, ketchup and toilet paper which cost you {5, 17, 9, 22} respectively.

Now, create your shopping list in R and for each item include their cost.

```r
shopping <- list(Fruit=c("Banana" = 12, "Apple" = 8,
                         "Peach" = 15, "Grape" = 10),
                 Diary=c("Milk" = 10, "Eggs" = 15,
                         "Yogurt" = 12, "Cheese" = 25),
                 Others=c("Candy" = 5,"Tooth Paste" = 17,
                          "Ketchup" = 9, "Toilet Paper" = 22))


shopping
```

```
## $Fruit
## Banana  Apple  Peach  Grape
##     12      8     15     10
##
## $Diary
##   Milk   Eggs Yogurt Cheese
##     10     15     12     25
##
## $Others
##        Candy  Tooth Paste      Ketchup Toilet Paper
##            5           17            9           22
```

Part A.

How much money did you pay for Diary ?

```
sum(shopping$Diary)
```

```
## [1] 62
```

Part B.

Which fruit did cost you the most ?

```
names(which.max(shopping[["Fruit"]]))
```

```
## [1] "Peach"
```

```
# alternatively
names(sort(shopping[["Fruit"]],decreasing = T)[1])
```

```
## [1] "Peach"
```

Part C.

Print out the cost of each element in your basket in increasing order.

```
sort(unlist(shopping))
```

```
##        Others.Candy         Fruit.Apple      Others.Ketchup         Fruit.Grape
##                   5                   8                   9                  10
##          Diary.Milk        Fruit.Banana        Diary.Yogurt         Fruit.Peach
##                  10                  12                  12                  15
##          Diary.Eggs  Others.Tooth Paste Others.Toilet Paper        Diary.Cheese
##                  15                  17                  22                  25
```

Part D.

Which group's mean is the smallest ?

```
which.min(c(mean(shopping[[1]]),
            mean(shopping[[2]]),
            mean(shopping[[3]])))
```

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

Part E.

Add more items in and take out some from your basket before you left the store and print the updated version. (You will pick what to add and what to remove.)

```
shopping[[1]] <- c(shopping[[1]], "Orange" = 7, "Blueberry" = 18)
shopping[[2]] <- c(shopping[[2]], "Ice Cream" = 20)
shopping[[3]] <- shopping[[3]][setdiff(names(shopping[[3]]),"Candy")]
shopping
```

```
## $Fruit
##    Banana     Apple     Peach     Grape    Orange Blueberry
##        12         8        15        10         7        18
##
## $Diary
##      Milk      Eggs    Yogurt    Cheese Ice Cream
##        10        15        12        25        20
##
## $Others
##   Tooth Paste       Ketchup Toilet Paper
##            17             9           22
```