CS 544 Module 1 Assignment

Phillip Escandon - [escandon@bu.edu](mailto:escandon@bu.edu)

24 January, 2021

## Part 1

### a. Vectors

How many students took the exam

scores <- c(59,46,76,60,49,65,82,68,99,52)  
numberOfStudents <- length(scores)  
numberOfStudents

## [1] 10

Firt two items of the vector

scores[c(1,2)]

## [1] 59 46

First and last items in the scores vector

fl <- scores[c(1,length(scores))]  
fl

## [1] 59 52

Access the middle two items

mid = length(scores)/2  
scores[c(mid,mid+1)]

## [1] 49 65

### b. Median

Use median

median(scores)

## [1] 62.5

Less than or equal to median

scores<=median(scores)

## [1] TRUE TRUE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE

Greater than the median

scores>median(scores)

## [1] FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE FALSE

Using the sum function- find the number of elements LESS THAN the median

sum(scores<=median(scores))

## [1] 5

Using the sum function to find the number of elements above the median

s = sum(scores>median(scores))  
s

## [1] 5

### c. Using logical expressions

Using logical expressions find the values <= and > to the median

scores[scores<=median(scores)]

## [1] 59 46 60 49 52

scores[scores>median(scores)]

## [1] 76 65 82 68 99

### d. Logical Indexing

Logical indexing with TRUE and FALSE to find odd and even **index** values in vector

scores[c(TRUE,FALSE)]

## [1] 59 76 49 82 99

scores[c(FALSE,TRUE)]

## [1] 46 60 65 68 52

### e. Numeric Expression

# need the ODD INDEX of the vector not the odd values!  
od<- seq(1,length(scores),2)  
ev<- seq(2,length(scores),2)  
scores[od]

## [1] 59 76 49 82 99

scores[ev]

## [1] 46 60 65 68 52

### f. Create a 2x5 matrix called scores.matrix

scores.matrix <- matrix(scores, nrow = 2, ncol = 5, byrow = TRUE)  
scores.matrix

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 59 46 76 60 49  
## [2,] 65 82 68 99 52

### g. First and last column of the matrix

scores.matrix[ ,c(1,ncol(scores.matrix))]

## [,1] [,2]  
## [1,] 59 49  
## [2,] 65 52

### h. Assign column names for the scores.matrix student 1-5, quiz 1-2

cnames <- paste("Student",1:ncol(scores.matrix),sep = "\_")  
rnames <- paste("Quiz",1:nrow(scores.matrix),sep = "\_")  
dimnames(scores.matrix) <- list(rnames,cnames)  
scores.matrix

## Student\_1 Student\_2 Student\_3 Student\_4 Student\_5  
## Quiz\_1 59 46 76 60 49  
## Quiz\_2 65 82 68 99 52

### i. Show the first and last columns with names using code from g.

scores.matrix[ ,c(1,ncol(scores.matrix))]

## Student\_1 Student\_5  
## Quiz\_1 59 49  
## Quiz\_2 65 52

## Part 2

### a. Create a dataframe -dow

month <- c("Jan","Feb","Mar","Apr","May")  
open <- c(28639,28320,25591,21227,24121)  
high <- c(29374,29569,27102,24765,24359)  
low <- c(28170,24681,18214,20735,23361)  
close <- c(28256,25409,21917,24346,24331)

dow <- data.frame(month,open,high,low,close)  
dow

## month open high low close  
## 1 Jan 28639 29374 28170 28256  
## 2 Feb 28320 29569 24681 25409  
## 3 Mar 25591 27102 18214 21917  
## 4 Apr 21227 24765 20735 24346  
## 5 May 24121 24359 23361 24331

### b. Show the summary() on the above data frame.

summary(dow[c(2:ncol(dow))])

## open high low close   
## Min. :21227 Min. :24359 Min. :18214 Min. :21917   
## 1st Qu.:24121 1st Qu.:24765 1st Qu.:20735 1st Qu.:24331   
## Median :25591 Median :27102 Median :23361 Median :24346   
## Mean :25580 Mean :27034 Mean :23032 Mean :24852   
## 3rd Qu.:28320 3rd Qu.:29374 3rd Qu.:24681 3rd Qu.:25409   
## Max. :28639 Max. :29569 Max. :28170 Max. :28256

### c. Slice data frame columns with month, open and close

dow[c("month","open","close")]

## month open close  
## 1 Jan 28639 28256  
## 2 Feb 28320 25409  
## 3 Mar 25591 21917  
## 4 Apr 21227 24346  
## 5 May 24121 24331

### d. Dataframe rows sliced, first and last row, any size dataframe

# data frame[ row , col]  
lastRow <- nrow(dow)  
dow[ c(1,lastRow), ]

## month open high low close  
## 1 Jan 28639 29374 28170 28256  
## 5 May 24121 24359 23361 24331

### e. Dataframe sliced by first and last row, columns MONTH, HIGH and LOW

Should work for any size dataframe

# data frame[ row , col ]  
lastRow <- nrow(dow)  
dow[ c(1,lastRow), c("month","high","low")]

## month high low  
## 1 Jan 29374 28170  
## 5 May 24359 23361

### f. Show all rows of data frame whose LOW is greater than 22,000. Any size dataframe.

Show logical indexing and one solution using the subset()

# dataframe [ row , col]  
dow

## month open high low close  
## 1 Jan 28639 29374 28170 28256  
## 2 Feb 28320 29569 24681 25409  
## 3 Mar 25591 27102 18214 21917  
## 4 Apr 21227 24765 20735 24346  
## 5 May 24121 24359 23361 24331

# logical indexing  
dow[ dow$low> 22000 , ]

## month open high low close  
## 1 Jan 28639 29374 28170 28256  
## 2 Feb 28320 29569 24681 25409  
## 5 May 24121 24359 23361 24331

# subset( df, rules)  
subset(dow, dow$low > 22000)

## month open high low close  
## 1 Jan 28639 29374 28170 28256  
## 2 Feb 28320 29569 24681 25409  
## 5 May 24121 24359 23361 24331

### g. Show all rows whose OPEN and LOW > 25,000.

Show logical indexing and one solution using the subset()

# indexing  
dow[c(dow$open>25000 & dow$low > 25000) , ]

## month open high low close  
## 1 Jan 28639 29374 28170 28256

# subset  
subset(dow,c(dow$open>25000 & dow$low > 25000))

## month open high low close  
## 1 Jan 28639 29374 28170 28256

### h. Add new column called VOLATILITY showing the difference between high and low

dow$volatility <- dow$high - dow$low  
dow

## month open high low close volatility  
## 1 Jan 28639 29374 28170 28256 1204  
## 2 Feb 28320 29569 24681 25409 4888  
## 3 Mar 25591 27102 18214 21917 8888  
## 4 Apr 21227 24765 20735 24346 4030  
## 5 May 24121 24359 23361 24331 998

### i. Show the rows with max volatilty. Use subset() and max()

max(dow$volatility)

## [1] 8888

subset(dow, dow$volatility == max(dow$volatility))

## month open high low close volatility  
## 3 Mar 25591 27102 18214 21917 8888

### j. Show the rows with minimum volatility. Use logical indexing and min()

Do no use the subset function

#subset(x, subset, select, drop = FALSE, ...)  
subset(dow,dow$volatility == min(dow$volatility))

## month open high low close volatility  
## 5 May 24121 24359 23361 24331 998