Exercise 1 - Analytical data processing operations using R

S.DEIVANAYKI (21BCS003)

12.01.2024

R Markdown

#1.R program to get the details of objects in memory

```
a=5
b='Hii'
c=6.3
print(ls())
```

```
## [1] "a" "b" "c"
```

```
ans='W'
print(ls())
```

```
## [1] "a"   "ans" "b"   "c"
```

#2.R program to print no.s from 1 to 100 and print “Fizz” for mulitples of 3, print “Buzz” for multiples of 5 and print “FizzBuzz” for mulitples of both

```
for (i in 1:100)
{
  if(i%%3==0 && i%%5==0)
  {
    print("FizzBuzz")
  }
  else if(i%%3==0)
  {
    print("Fizz")
  }
  else if(i%%5==0)
  {
    print("Buzz")
  }
  else
  {
    print(i)
  }
}
```

```
## [1] 1
## [1] 2
## [1] "Fizz"
## [1] 4
```

```
## [1] "Buzz"
## [1] "Fizz"
## [1] 7
## [1] 8
## [1] "Fizz"
## [1] "Buzz"
## [1] 11
## [1] "Fizz"
## [1] 13
## [1] 14
## [1] "FizzBuzz"
```

#3. To create 3 vectors a,b,c with 3 integers combine 3 vectors to become 3*3 matrix where each column represent vector

```
n1<-c(12,2,5)
n2<-c(2,0,11)
n3<-c(3,20,7)
ans<-rbind(n1,n2,n3)
as.matrix(ans)
```

```
##    [,1] [,2] [,3]
## n1   12    2    5
## n2    2    0   11
## n3    3   20    7
```

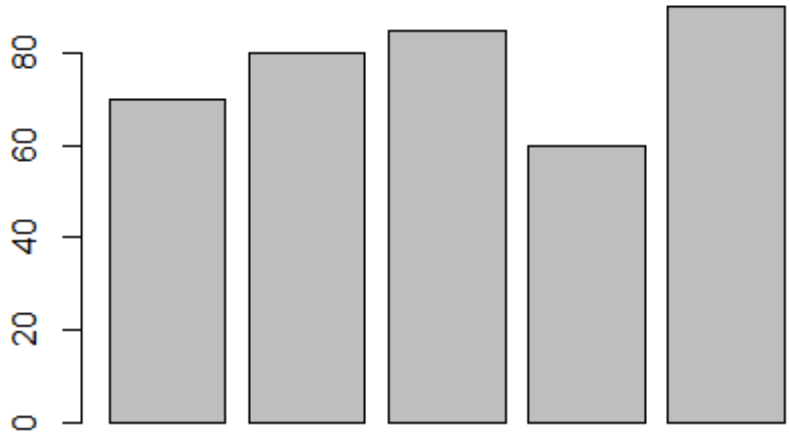
#4. R program to combine three arrays so that the 1st row of first array is followed by the first row of the second array and then first row of the 3rd array

```
a<-array(c(2,5,16,7,81,9),c(2,3))
b<-array(c(1,0,10,2,11,15),c(2,3))
c<-array(c(5,15,11,7,3,5),c(2,3))
d<-a[1,]
e<-b[1,]
f<-c[1,]
res<-array(c(d,e,f))
res
```

```
## [1]  2 16 81  1 10 11  5 11  3
```

#5. R program to create a simple bar plot of 5 subject marks

```
marks<-c(70,80,85,60,90)
barplot(marks)
```



#6. R program to concatenate 2 given factors into a single factors

```
a<-factor(c('a','b','a','c','z','b','x'))
b<-factor(c('z','z','a','t','z','b','x'))
c<-factor(c(a,b))
a

## [1] a b a c z b x
## Levels: a b c x z

b

## [1] z z a t z b x
## Levels: a b t x z

c

##  [1] a b a c z b x z z a t z b x
## Levels: a b c x z t
```

#7. R program to a) Create an empty data frame b) Create a data frame from 4 given vector c) Get the structure of a given dataframe. d) Get the statistical summary and nature of the data of a given data frame.

```
a<-data.frame()
a

## data frame with 0 columns and 0 rows
```

```r
b<-c(1,21)
c<-c(32,77)
d<-c(4,90)
e<-c(17,6)
ans<-data.frame(b,c,d,e)
ans
```

```
##    b  c  d  e
## 1  1 32  4 17
## 2 21 77 90  6
```

```r
dim(ans)
```

```
## [1] 2 4
```

```r
dimnames(ans)
```

```
## [[1]]
## [1] "1" "2"
##
## [[2]]
## [1] "b" "c" "d" "e"
```

```r
summary(ans)
```

```
##        b            c               d              e
##  Min.   : 1   Min.   :32.00   Min.   : 4.0   Min.   : 6.00
##  1st Qu.: 6   1st Qu.:43.25   1st Qu.:25.5   1st Qu.: 8.75
##  Median :11   Median :54.50   Median :47.0   Median :11.50
##  Mean   :11   Mean   :54.50   Mean   :47.0   Mean   :11.50
##  3rd Qu.:16   3rd Qu.:65.75   3rd Qu.:68.5   3rd Qu.:14.25
##  Max.   :21   Max.   :77.00   Max.   :90.0   Max.   :17.00
```

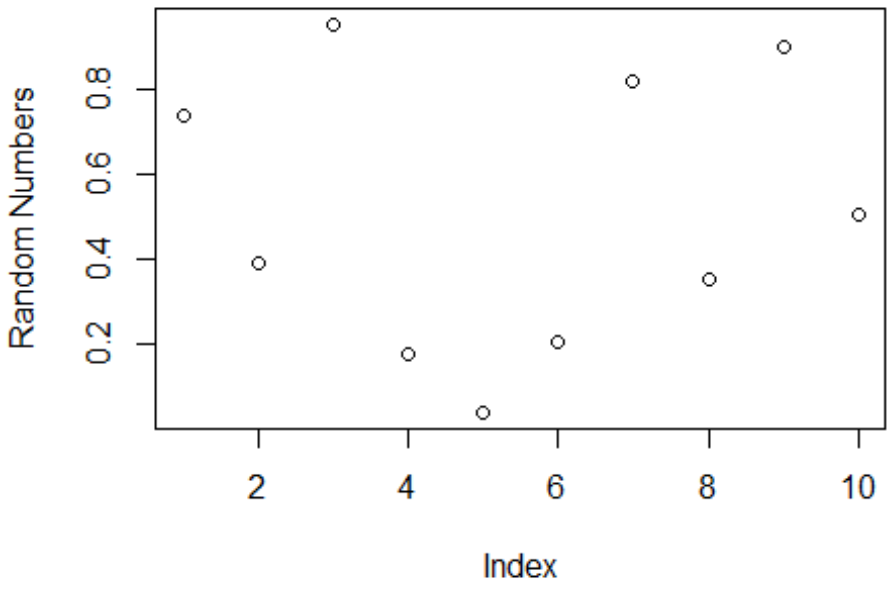
#8. R program to generate ten uniformly distributed numbers over(0,1)

```r
nums<-runif(10)
nums
```

```
##  [1] 0.74037575 0.39133301 0.95377288 0.17646391 0.04064874 0.20735133
##  [7] 0.81946828 0.35575717 0.89875647 0.50557859
```

#9. Display them using plot

```r
plot(nums,xlab="Index",ylab="Random Numbers")
```



#10. Generate 5 random integers 1:10 with or without replacement

```
r1<-sample(1:10,5,replace=TRUE)
r2<-sample(1:10,5,replace=FALSE)
r1

## [1]  7  8  3  6 10

r2

## [1] 10  8  9  7  6
```

#11. Select 2 random months from a list of months. Combine two string with comma separation.

```
mon<-sample(month.name,2)
paste(mon[1],',',mon[2])

## [1] "December , October"
```

#12. Prefix and Suffix a word with 'A'.

```
library(stringr)
m=sample(words,1)
paste('A',m,'A',sep="")

## [1] "AwhyA"
```

#13. Find number of characters in a string.

```
n=sample(words,1)
cnt<-nchar(n)
paste("Given String",n," and its number of characters is ",cnt)
```

```
## [1] "Given String case  and its number of characters is  4"
```

#14. Construct a vector of odd numbers 1 to 10 using seq().

```
vec<-seq(1,10,2)
vec
```

```
## [1] 1 3 5 7 9
```

#15. Find its log to the base 10 and sin values

```
vec
```

```
## [1] 1 3 5 7 9
```

```
logvalue<-log10(vec)
sinvalue<-sin(vec)
paste("Log10 value : ",logvalue)
```

```
## [1] "Log10 value :  0"                  "Log10 value :  0.477121254719662"
## [3] "Log10 value :  0.698970004336019" "Log10 value :  0.845098040014257"
## [5] "Log10 value :  0.954242509439325"
```

```
paste("Sine value : ",sinvalue)
```

```
## [1] "Sine value :  0.841470984807897"  "Sine value :  0.141120008059867"
## [3] "Sine value :  -0.958924274663138" "Sine value :  0.656986598718789"
## [5] "Sine value :  0.412118485241757"
```

#16. Find sum,mean,median,length,variance,standard deviation,quantile.

```
tot<-sum(vec)
avg<-mean(vec)
mid<-median(vec)
len<-length(vec)
v<-var(vec)
s<-sd(vec)
qval<-quantile(vec)
paste("Vector",vec)
```

```
## [1] "Vector 1" "Vector 3" "Vector 5" "Vector 7" "Vector 9"
```

```
paste("Sum of Vector                : ",tot)
```

```
## [1] "Sum of Vector                :  25"
```

```
paste("Mean of vecotr               : ",avg)
```

```
## [1] "Mean of vecotr               :  5"
```

```
paste("Median of vector             : ",mid)
```

```
## [1] "Median of vector             :  5"
```

```
paste("Length of vector             : ",len)
```

```
## [1] "Length of vector             :  5"
```

```
paste("Variance of vector           : ",v)
```

```
## [1] "Variance of vector           :  10"
```

```
paste("Standard deviation of vector : ",s)
```

```
## [1] "Standard deviation of vector :  3.16227766016838"
```

```
paste("Quantile                     : ",qval)
```

```
## [1] "Quantile                     :  1" "Quantile                     :
3"
## [3] "Quantile                     :  5" "Quantile                     :
7"
## [5] "Quantile                     :  9"
```

#17. Find summary.

```
summary(vec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       3       5       5       7       9
```

#18. Get today date and time

```
a<-Sys.time()
format(a,"%Y-%m-%d")
```

```
## [1] "2024-02-13"
```

```
format(a,"%H:%M:%S")
```

```
## [1] "14:49:15"
```

#19. Find your ages in days

```
  birthday<-as.Date("2003-12-03")
  cur_date<-Sys.Date()
  age_in_days<-as.numeric(difftime(cur_date,birthday,units = "days"))
  age_in_days
```

```
## [1] 7377
```

#20. Convert to lower and upper case

```
a<-sample(words,1)
upCase<-toupper(a)
```

```
lwCase<-tolower(a)
paste("UpperCase : ",upCase)

## [1] "UpperCase :  BLOKE"

paste("LowerCase : ",lwCase)

## [1] "LowerCase :  bloke"
```

#21. Extract last 3 characters in string

```
char3<-sample(words,1)
result<-substr(char3,nchar(char3)-2,nchar(char3))
paste("Analyzed Word : ",char3)

## [1] "Analyzed Word :  time"

paste("Extracted 3 Character : ",result)

## [1] "Extracted 3 Character :  ime"
```

#22. Sort the strings in lexicographic order

```
word_list<-sample(words,6)
paste("Words: ")

## [1] "Words: "

paste(word_list)

## [1] "in"     "over"   "fall"   "closes" "since"  "single"

paste("After Sorting..")

## [1] "After Sorting.."

sort(word_list)

## [1] "closes" "fall"   "in"     "over"   "since"  "single"
```

#23. Display strings that start with 'A'

```
word_list1<-sample(words,6)
paste("Words: ")

## [1] "Words: "

paste(word_list1)

## [1] "touch"  "moment" "side"   "suit"   "after"  "direct"

paste("Words that start with A ..")

## [1] "Words that start with A .."
```

```
sum(startsWith(word_list1,'A'))
```

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

#24. How many words end with a vowel

```
word_list2<-c("accept","grand","history","some","will","correct")
paste("words : ")
```

```
## [1] "words : "
```

```
paste(word_list)
```

```
## [1] "in"     "over"   "fall"   "closes" "since"  "single"
```

```
paste("Number of word that ends eith vowel..")
```

```
## [1] "Number of word that ends eith vowel.."
```

```
sum(str_detect(word_list2,"[aeiou]$"))
```

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

```
sum(endsWith(word_list2,c('a','e','i','o','u')))
```

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

#25. Find all words containing at least one vowel

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
a<-sample(words,6)
paste("Words ..")
```

```
## [1] "Words .."
```

```
paste(a)
```

```
## [1] "stuff"   "count"   "without" "worry"   "glass"   "little"
```

```
paste("Words contains atleast 1 vowel")
```

```
## [1] "Words contains atleast 1 vowel"
```

```
ind<-
unique(c(contains("a",vars=a),contains("e",vars=a),contains("i",vars=a),conta
ins("o",vars=a),contains("u",vars=a)))
word_list2<-c("accept","grand","history","some","will","correct","aaa")
paste("words : ")

paste(word_list)

## [1] "in"     "over"   "fall"   "closes" "since"  "single"

paste("Number of word that have atleast one consonant..")

word_list[str_detect(word_list2,"[bcdfghjklmnpqrstvwxyz]")]

## [1] "in"     "over"   "fall"   "closes" "since"  "single"
```

#26. Find all words containing only of consonants

```
a<-sample(words,6)
paste("Words ..")

## [1] "Words .."

paste(a)

## [1] "guy"    "claim"  "art"    "lad"    "brief"  "slight"

paste("Words contains only consonants")

## [1] "Words contains only consonants"

ind<-
unique(c(contains("a",vars=a),contains("e",vars=a),contains("i",vars=a),conta
ins("o",vars=a),contains("u",vars=a)))
a[ifelse(length(ind),FALSE,TRUE)]

## character(0)
```

R-code for following

```
x<-c(1:5,NA,7:10,NULL)
y<-c(1:5,Inf,7:10)
```

#27. Mark R ignore the NA values

```
xna<-na.omit(x)
yna<-na.omit(y)
xna

## [1]  1  2  3  4  5  7  8  9 10
## attr(,"na.action")
## [1] 6
## attr(,"class")
## [1] "omit"
```

```
yna
```

```
##  [1]   1   2   3   4   5 Inf   7   8   9  10
```

#28. What is the length of the NULL object?

```
length(is.null(x))
```

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

```
length(is.null(y))
```

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

#29.Construct a list of numeric vector, a logical, a string and another list.

```
l1<-list(1,2,11,7,5)
l2<-list(TRUE,FALSE,FALSE,TRUE,FALSE)
l3<-list("HPC","DS","OOAD","WUI")
l4<-list(1,2,7,list('a','b'),71,23)
l5<-list(1,'a',c(1,2,3),"Computer",TRUE,17.4)
paste("List of numeric vector..")
```

```
## [1] "List of numeric vector.."
```

```
l1
```

```
## [[1]]
## [1] 1
## [[2]]
## [1] 2
## [[3]]
## [1] 11
## [[4]]
## [1] 7
## [[5]]
## [1] 5
```

```
paste("List of logical values..")
```

```
## [1] "List of logical values.."
```

```
l2
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] FALSE
## [[3]]
## [1] FALSE
## [[4]]
## [1] TRUE
```

```
## [[5]]
## [1] FALSE
```

```
paste("List of string..")
```

```
## [1] "List of string.."
```

```
l3
```

```
## [[1]]
## [1] "HPC"
## [[2]]
## [1] "DS"
## [[3]]
## [1] "OOAD"
## [[4]]
## [1] "WUI"
```

```
paste("Nested List..")
```

```
## [1] "Nested List.."
```

```
l4
```

```
## [[1]]
## [1] 1
## [[2]]
## [1] 2
## [[3]]
## [1] 7
## [[4]]
## [[4]][[1]]
## [1] "a"
## [[4]][[2]]
## [1] "b"
## [[5]]
## [1] 71
## [[6]]
## [1] 23
```

```
paste("Heterogeneous List..")
```

```
## [1] "Heterogeneous List.."
```

```
l5
```

```
## [[1]]
## [1] 1
## [[2]]
## [1] "a"

## [[3]]
## [1] 1 2 3
```

```
## [[4]]
## [1] "Computer"
## [[5]]
## [1] TRUE
## [[6]]
## [1] 17.4
```

#30. Name some or all of the entries in our list by supplying argument names to list()

```
l<-list(num1=1,char1='a',vec=c(1,2,3),TRUE,7,char2='e',12.5)
l
```

```
## $num1
## [1] 1
## $char1
## [1] "a"
## $vec
## [1] 1 2 3
## [[4]]
## [1] TRUE
## [[5]]
## [1] 7
## $char2
## [1] "e"
## [[7]]
## [1] 12.5
```

#31. To create matrix(), diag()

```
a1<-matrix(c(1,2,3,4,5,6),nrow=2,ncol=3)
a1
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
a2<-diag(2)
a2
```

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

#32. Perform some operations

```
paste("Operations..")
```

```
## [1] "Operations.."
```

```
sum(a1)
```

```
## [1] 21
```

```r
min(a1)
```

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

```r
max(a1)
```

```
## [1] 6
```

```r
dim(a1)
```

```
## [1] 2 3
```

```r
t(a1)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
```

```r
mean(a1)
```

```
## [1] 3.5
```

#33. Construct a Dataframe and perform expansion on it.

```r
x<-data.frame(x=sample(1:10,20,replace =
TRUE),y=sample(10:20,20,replace=TRUE))
x
```

```
##     x  y
## 1   9 12
## 2   2 14
## 3   1 14
## 4   2 18
## 5   4 16
## 6   1 19
```

```r
z=sample(20:30,10,replace=TRUE)
cbind(x,z)
```

```
##     x  y  z
## 1   9 12 27
## 2   2 14 20
## 3   1 14 20
## 4   2 18 30
## 5   4 16 20
## 6   1 19 27
## 7   4 15 26
```