

SCS2111 - Tutorial 2

Introduction to R

Index No: -18001149

1. Determine the type of the following constant.

```
> typeof(10)
[1] "double"
> typeof(2e-4)
[1] "double"
> typeof(2i+3)
[1] "complex"
> typeof(5L)
[1] "integer"
> typeof(0*F)
[1] "double"
> |
```

- 2) i. Create a vector which contains 10, 12, 14, 16, 18, 20 values.
ii. Create sequence with 6 equidistant numbers between 10 and 20. Hint: use *seq()*
iii. Check whether above two vectors are equal and identical.
iv. Combine above two vectors you created in question **b. i.** and **b. ii.**

```
> justice<-rbind(batman,m)
> justice
      [,1] [,2] [,3] [,4] [,5] [,6]
batman  10  12  14  16  18  20
m       10  12  14  16  18  20
```

- v. Create vectors below and find the mean and median of those vectors.

`x<-c(1:3, NA, NULL, 5)`

1. `y<-c(1:3, Inf, runif(3))`

```
> batman<-c(10,12,14,16,18,20)
> batman
[1] 10 12 14 16 18 20
> seq(10,20,length=6)
[1] 10 12 14 16 18 20
> m<-seq(10,20,length = 6)
> m
[1] 10 12 14 16 18 20
> batman==m
[1] TRUE TRUE TRUE TRUE TRUE TRUE
> Justics<-cbind(batman,m)
> Justics
      batman  m
[1,]    10 10
[2,]    12 12
[3,]    14 14
[4,]    16 16
[5,]    18 18
[6,]    20 20
> |

> x<-c(1:3,NA,NULL,5)
> y<-c(1:3,Inf,runif(3))
Error: unexpected ',' in "y<-(1:3,"
> y<-c(1:3,Inf,runif(3))
> x
[1] 1 2 3 NA 5
> y
[1] 1.0000000 2.0000000 3.0000000      Inf 0.6492208
> avg.x<-mean(x)
> avg.x
[1] NA
> median(x)
[1] NA
> avg.y<-mean(y)
> avg.y
[1] Inf
> y.median<-y(median)
Error in y(median) : could not find function "y"
> y.median<-median(y)
> y.median
[1] 1
> |
```

b.

i. Create three vectors A = 1:4 B= 5:8 C=9:10.

ii. Combine A, B, C vectors. Observe the short vector C. Hint: use *cbind()*

```
> a<-seq(1,4,length=1)
> a
[1] 1
> a<-seq(1,4)
> a
[1] 1 2 3 4
> b<-seq(5,8)
> b
[1] 5 6 7 8
> c<-seq(9,10)
> c
[1] 9 10
> hat<-cbind(a,b,c)
> hat
      a b  c
[1,] 1 5  9
[2,] 2 6 10
[3,] 3 7  9
[4,] 4 8 10
> |
```

iii. Create a sequence of 12 equidistant numbers as you wish.

```
> bat<-seq(20,56,length = 12)
> bat
[1] 20.00000 23.27273 26.54545 29.81818 33.09091 36.36364 39.63636
[8] 42.90909 46.18182 49.45455 52.72727 56.00000
> # matrix(bat)
```

iv. Make above sequence into 4*3 dimension matrix.

v. Multiply above matrix and combined vector matrix which you created in question c. ii. and

c. iv.

```
> wonder<-matrix(bat,nrow=4,byrow=FALSE)
> wonder
      [,1] [,2] [,3]
[1,] 20.00000 33.09091 46.18182
[2,] 23.27273 36.36364 49.45455
[3,] 26.54545 39.63636 52.72727
[4,] 29.81818 42.90909 56.00000
> wonder*hat
      a      b      c
[1,] 20.00000 165.4545 415.6364
[2,] 46.54545 218.1818 494.5455
[3,] 79.63636 277.4545 474.5455
[4,] 119.27273 343.2727 560.0000
> |
```

c.

i. Create the vector V1 using random 16 numbers between 1 and 10.

ii. Convert V1 to 8*2 dimension matrix M1.

iii. Multiply M1 and transpose of M1.

- iv. Convert V1 to 4*4 dimension matrix M2.
v. Multiply M2 and transpose of M2.

```
> v1<-runif(16,min=0,max=10)
> v1
[1] 9.9674413 1.1704605 8.9977963 0.2656878 2.8439813 9.7302696
[7] 9.9114664 0.9163694 5.3688920 3.8800367 2.6627303 3.1587078
[13] 5.6268865 6.7006471 2.1164177 6.3676539
> m1<-matrix(v1,nrow=8,byrow=TRUE)
> m1
      [,1]      [,2]
[1,] 9.967441 1.1704605
[2,] 8.997796 0.2656878
[3,] 2.843981 9.7302696
[4,] 9.911466 0.9163694
[5,] 5.368892 3.8800367
[6,] 2.662730 3.1587078
[7,] 5.626887 6.7006471
[8,] 2.116418 6.3676539
> M1<-matrix(v1,nrow=2,byrow=FALSE)
> M1
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 9.967441 8.9977963 2.843981 9.9114664 5.368892 2.662730
[2,] 1.170460 0.2656878 9.730270 0.9163694 3.880037 3.158708
      [,7]      [,8]
[1,] 5.626887 2.116418
[2,] 6.700647 6.367654
> m1*M1
Error in m1 * M1 : non-conformable arrays
> M1*m1
Error in M1 * m1 : non-conformable arrays
> M1*m1
Error in M1 * m1 : non-conformable arrays
> m2<-matrix(v1,nrow=4,byrow=TRUE)
> m2
      [,1]      [,2]      [,3]      [,4]
[1,] 9.967441 1.170460 8.997796 0.2656878
[2,] 2.843981 9.730270 9.911466 0.9163694
[3,] 5.368892 3.880037 2.662730 3.1587078
[4,] 5.626887 6.700647 2.116418 6.3676539
> M2<-matrix(v1,nrow=4,byrow=FALSE)
> M2
      [,1]      [,2]      [,3]      [,4]
[1,] 9.9674413 2.8439813 5.368892 5.626887
[2,] 1.1704605 9.7302696 3.880037 6.700647
[3,] 8.9977963 9.9114664 2.662730 2.116418
[4,] 0.2656878 0.9163694 3.158708 6.367654
> m2*M2
      [,1]      [,2]      [,3]      [,4]
[1,] 99.349887 3.328768 48.308197 1.494995
[2,] 3.328768 94.678147 38.456854 6.140268
[3,] 48.308197 38.456854 7.090133 6.685145
[4,] 1.494995 6.140268 6.685145 40.547017
> |
```

3) Run below commands using R. What are the tasks of those commands?

i. `A<-array(month.name)`

ii. `factor(A)`

iii. `A[-3]`

iv. `dim(A)<-c(3,4)`

v. `U<-list(month=A, num=1:12, fun1 = runif, fun2= factor)`

vi. `U`

vii. `DF<-data.frame(U$month, U$num)`

viii. `DF`

ix. `head(DF, 3)`

x. `dim(DF)`

```
> A<-array(month.name)
> A
[1] "January" "February" "March" "April" "May"
[6] "June" "July" "August" "September" "October"
[11] "November" "December"
> factor(A)
[1] January February March April May June
[7] July August September October November December
12 Levels: April August December February January July ... September
> A[-3]
[1] "January" "February" "April" "May" "June"
[6] "July" "August" "September" "October" "November"
[11] "December"
> dim(A)<-c(3,4)
> dim(a)
NULL
> dim(A)<-c(3,4)
> U<-list(month=A, num=1:12, fun1 = runif, fun2= factor )
```

```
      [,1]      [,2]      [,3]      [,4]
[1,] "January" "April" "July"      "October"
[2,] "February" "May" "August" "November"
[3,] "March" "June" "September" "December"
```

```
$num
[1] 1 2 3 4 5 6 7 8 9 10 11 12
```

```
$fun1
function (n, min = 0, max = 1)
.Call(C_runif, n, min, max)
<bytecode: 0x0000020740fc9e90>
<environment: namespace:stats>
```

```
$fun2
function (x = character(), levels, labels = levels, exclude = NA,
ordered = is.ordered(x), nmax = NA)
{
  if (is.null(x))
    x <- character()
  nx <- names(x)
  if (missing(levels)) {
    y <- unique(x, nmax = nmax)
    ind <- order(y)
    levels <- unique(as.character(y)[ind])
  }
  force(ordered)
  if (!is.character(x))
    x <- as.character(x)
  levels <- levels[is.na(match(levels, exclude))]
  f <- match(x, levels)
  if (!is.null(nx))
    names(f) <- nx
  if (missing(labels)) {
    levels(f) <- as.character(levels)
  }
  else {
    nlab <- length(labels)
    if (nlab == length(levels)) {
      nlevs <- unique(xlevs <- as.character(labels))
      at <- attributes(f)
      at$levels <- nlevs
      f <- match(xlevs, nlevs)[f]
      attributes(f) <- at
    }
    else if (nlab == 1L)
      levels(f) <- paste0(labels, seq_along(levels))
    else stop(gettextf("invalid 'labels'; length %d should be 1 or
%d",
      nlab, length(levels)), domain = NA)
  }
  class(f) <- c(if (ordered) "ordered", "factor")
  f
}
<bytecode: 0x000002073edb2b38>
<environment: namespace:base>
```

```
> DF<-data.frame(U$month, U$num)
> DF
```

	x1	x2	x3	x4	U. num
1	January	April	July	October	1
2	February	May	August	November	2
3	March	June	September	December	3
4	January	April	July	October	4
5	February	May	August	November	5
6	March	June	September	December	6
7	January	April	July	October	7
8	February	May	August	November	8
9	March	June	September	December	9
10	January	April	July	October	10
11	February	May	August	November	11
12	March	June	September	December	12

```
> head(DF, 3)
```

	x1	x2	x3	x4	U. num
1	January	April	July	October	1
2	February	May	August	November	2
3	March	June	September	December	3

```
> dim(DF)
[1] 12 5
> |
```

```
> DF<-data.frame(U$month, U$num)
> DF
```

	x1	x2	x3	x4	U. num
1	January	April	July	October	1
2	February	May	August	November	2
3	March	June	September	December	3
4	January	April	July	October	4
5	February	May	August	November	5
6	March	June	September	December	6
7	January	April	July	October	7
8	February	May	August	November	8
9	March	June	September	December	9
10	January	April	July	October	10
11	February	May	August	November	11
12	March	June	September	December	12

```
> head(DF, 3)
```

	x1	x2	x3	x4	U. num
1	January	April	July	October	1
2	February	May	August	November	2
3	March	June	September	December	3

```
> dim(DF)
[1] 12 5
> write.table(DataToBeExplore,file = "Df.txt",sep = ",")
Error in is.data.frame(x) : object 'DataToBeExplore' not found
> write.table(DF,file = "Df.txt",sep = ",")
. |
```

<input type="checkbox"/>	Custom Office Templates		
<input type="checkbox"/>	CyberLink		
<input type="checkbox"/>	d1.txt	10.7 KB	Jan 15, 2021, 3:35 PM
<input type="checkbox"/>	d2.txt	10.7 KB	Jan 15, 2021, 3:36 PM
<input type="checkbox"/>	desktop.ini	402 B	Jul 22, 2020, 8:54 PM
<input type="checkbox"/>	Df.txt	559 B	Jan 22, 2021, 4:58 PM
<input type="checkbox"/>	Downloads		
<input type="checkbox"/>	File.txt	62 B	Dec 7, 2020, 5:28 PM
<input type="checkbox"/>	FilmoraScreen		
<input type="checkbox"/>	Fragments		
<input type="checkbox"/>	Game Rush		
<input type="checkbox"/>	GitHub		
<input type="checkbox"/>	My Music		
<input type="checkbox"/>	My Pictures		
<input type="checkbox"/>	My Shapes		
<input type="checkbox"/>	My Videos		
<input type="checkbox"/>	mvscrip.txt	108 B	Jan 15, 2021, 4:37 PM