R PROGRAMMING & MATLAB LAB

(ACADEMIC YEAR : 2017-2018)

I SEMESTER

ASSIGNMENT 2

ASHISH CHANDRAKANT DUSANE

M. TECH. (ACDS)

COMPUTER ENGG. DEPARTMENT

{ PRN : 170101261004 }

Section I : Integer Vectors :-

1. Create two integer vectors of equal lengths and apply mathematical operations like – (Subtraction), *(Multiplication) .

```
> a<-c(2,4,6,8,10)
> b<-c(1,2,4,5,6)
> a-b
[1] 1 2 2 3 4
> a*b
[1]  2  8 24 40 60
```

OR

```
> z<-a+b
> z
[1]  3  6 10 13 16
> z<-a*b
> z
[1]  2  8 24 40 60
```

2. Create two different integer vectors of unequal length and apply the basic math functions (+,-,/,%,^,*) and check the output.

```
> a<-c(10,20,30,40)
> b<-c(4,6,8,10,12)

> a+b
[1] 14 26 38 50 22
Warning message:
In a + b : longer object length is not a multiple of shorter object length

> a-b
[1]  6 14 22 30 -2
Warning message:
In a - b : longer object length is not a multiple of shorter object length

> a/b
[1] 2.5000000 3.3333333 3.7500000 4.0000000 0.8333333
```

Warning message:
In a/b : longer object length is not a multiple of shorter object length

> a*b
[1] 40 120 240 400 120
Warning message:
In a * b : longer object length is not a multiple of shorter object length

> a^b
[1] 1.000000e+04 6.400000e+07 6.561000e+11 1.048576e+16 1.000000e+12
Warning message:
In a^b : longer object length is not a multiple of shorter object length

3. Apply following functions factorial, logarithm, sin(x), cos(x), tan(x), acos(x), asin(x), atan(x) on single integer and check the output.

> a<-c(10)

> factorial (a)
[1] 3628800

> log (a)
[1] 2.302585

> a<-c(0.5)
> a
[1] 0.5

> asin(a)
[1] 0.5235988

> acos(a)
[1] 1.047198

> atan(a)
[1] 0.4636476

4. Create integer vector of 100 elements starting from 201 to 300 and store in an object and apply the following functions (sum, mean, min, max, length, median, mode) .

> a<-c(201:300)

> a
[1] 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218
[19] 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236
[37] 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254
[55] 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272
[73] 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290
[91] 291 292 293 294 295 296 297 298 299 300

> sum(a)
[1] 25050

> mean(a)
[1] 250.5

> min(a)
[1] 201

> max(a)
[1] 300

> length(a)
[1] 100

> median(a)
[1] 250.5

> mode(a)
[1] "numeric"

5. Create integer vector with one or more NA and apply sum, mean, min, max, length, median, mode functions.

```
> b<-c(10,20,30,65,NA,NA)
> b
[1] 10 20 30 65 NA NA

> sum(b)
[1] NA

> mean(b)
[1] NA

> min(b)
[1] NA

> max(b)
[1] NA

> length(b)
[1] 6


> median(b)
[1] NA

> mode(b)
[1] "numeric"
```

6. Create the integer vector with the following values
Output: 0.10 0.11 0.12 0.13 0.14 0.15 0.16 . . . . . 0.95 0.96 0.97 0.98 0.99 1.00

```
> b<-seq(0.10,1.00,by=0.01)
> b
 [1] 0.10 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.20 0.21 0.22 0.23 0.24
[16] 0.25 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39
```

[31] 0.40 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54
[46] 0.55 0.56 0.57 0.58 0.59 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69
[61] 0.70 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82 0.83 0.84
[76] 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99
[91] 1.00

7. Retrieve every odd value from the integer created in question 4
Output: 201 203 205 207..... 293 295 297 299

> a<-c(201:300)
> a<-seq(201,by=2,len=50)
> a
[1] 201 203 205 207 209 211 213 215 217 219 221 223 225 227 229 231 233 235 237
[20] 239 241 243 245 247 249 251 253 255 257 259 261 263 265 267 269 271 273 275
[39] 277 279 281 283 285 287 289 291 293 295 297 299

8. Create integer vector as follows by using rep() command and store it in different objects
i. **1 2 2 3 3 3 4 4 4 4 5 5 5 5 5**
> a<-c(1,2,3,4,5)
> a
[1] 1 2 3 4 5

> rep(a,1:5)
> rep(a,1:5)
[1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5

ii. **1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5**
> rep(1:5,5)
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

iii. **1 1 1 1 1 2 2 2 2 3 3 3 4 4 5**
> rep(a,5:1)

[1] 1 1 1 1 1 2 2 2 2 3 3 3 4 4 5

iv. **5 8 5 8 5 8 5 8**
> a<-c(5,8)
> a
[1] 5 8

> rep(a,3)
[1] 5 8 5 8 5 8

v. **5 5 5 5 8 8 8 8**
> rep(a,c(4,4))
[1] 5 5 5 5 8 8 8 8

Section II : Character Vectors :~

9. Create a simple character vector of five or six values and give names to it.
> a<-c("One","Two","Hundred","Thousand","Five")
> a
[1] "One" "Two" "Hundred" "Thousand" "Five"

10. Create a simple character vector using scan() command.
> a<-scan(what="character")
1: One
2: Two
3: Hundred
4: Thousand
5: Five
6:
Read 5 items
> a
[1] "One" "Two" "Hundred" "Thousand" "Five"

11. Apply length function on the character vector created in the question 9

> length(a)
[1] 5

12. Retrieve first and last elements from the character vector created in the question 9

> cat(head(a,n=1)); cat("\n"); cat(tail(a,n=1))
One
Five

13. Create the following vector (of type factor) and typecast it to numeric and check the output.
"boy","girl","boy","girl","boy","boy"

> z<-factor(c("Boy","Girl","Boy","Girl","Girl"));
> z
[1] Boy Girl Boy Girl Girl
Levels: Boy Girl
> y<-factor(z)
> y
[1] Boy Girl Boy Girl Girl
Levels: Boy Girl
> as.numeric(y)
[1] 1 2 1 2 2

<u>Section III : Matrices :-</u>

14. Create 4x4 matrix without using matrix() function. Matrix elements should comprise of following values
from 16 to 31

> p<-c(16,17,18,19)
> q<-c(20,21,22,23)
> r<-c(24,25,26,27)
> s<-c(28,29,30,31)
> t<-cbind(p,q,r,s)
> t
     p q r s

[1,] 16 20 24 28
[2,] 17 21 25 29
[3,] 18 22 26 30
[4,] 19 23 27 31

15. Create the following matrices and Create the variables a = 3 and b = 4.5.

> p<-c(1,4,7)
> q<-c(2,5,8)
> r<-c(3,6,10)
> A<-cbind(p,q,r)
> A

```
     p q  r
[1,] 1  2  3
[2,] 4  5  6
[3,] 7  8 10
```

> e<-c(1,2,3)
> f<-c(4,5,6)
> g<-c(7,8,9)
> B<-cbind(e,f,g)

> B

```
     e f g
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

> y<-c(1,2,3)
> y
[1] 1 2 3

> a<-3
> a

```
[1] 3

> b<-4.5
> b
[1] 4.5
```

i. **a 2 + 1/b**

```
> a*a+1/b
[1] 9.222222
```

ii. **a *A ## Multiplication with a scalar**

```
> a*A
     p  q  r
[1,]  3  6  9
[2,] 12 15 18
[3,] 21 24 30
```

iii. **A B Matrix multiplication (R - command %*%)**

```
> A*B
     p  q  r
[1,]  1  8 21
[2,]  8 25 48
[3,] 21 48 90
```

iv. **Transpose A. (R - command t())**

```
> t(A)
  [,1] [,2] [,3]
p    1    4    7
q    2    5    8
r    3    6   10
```

v. **Fill the first row of B matrix with ones**

```
> B[1,]<-c(1,1,1)
> B
     e f g
[1,] 1 1 1
[2,] 2 5 8
```

[3,] 3 6 9

vi. Row and Column sums of all the three matrices

> A+B+y

```
     p  q  r
[1,]  3  4  5
[2,]  8 12 16
[3,] 13 17 22
```

vii. Row and Column means of all the three matrices

> mean(A)

[1] 5.111111

> mean(B)

[1] 4

> mean(y)

[1] 2

16. Create 5x5 matrix using matrix() function. Fill the matrix row wise.

> matrix(1,5,5)

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

17. Create 4x4 matrix using cbind() function

```
> a<-c(1,1,1,1)
> b<-c(2,2,2,2)
> c<-c(3,3,3,3)
> d<-c(4,4,4,4)
> e<-cbind(a,b,c,d)
```

> e

```
     a b c d
[1,] 1 2 3 4
[2,] 1 2 3 4
[3,] 1 2 3 4
[4,] 1 2 3 4
```

18. Create 5x5 matrix using rbind() function

```
> a<-c(1,1,1,1,1)
> b<-c(2,2,2,2,2)
> c<-c(3,3,3,3,3)
> f<-c(4,4,4,4,4)
> e<-c(5,5,5,5,5)
> g<-rbind(a,b,c,f,e)
```

> g

```
  [,1] [,2] [,3] [,4] [,5]
a    1    1    1    1    1
b    2    2    2    2    2
c    3    3    3    3    3
f    4    4    4    4    4
e    5    5    5    5    5
```

19. Apply matrix multiplication and matrix addition on the matrices created in question 11 and question 13 and check the output.
INVALID Question

20. Consider any one of the matrix created above and do the following.

I. Multiply with any integer

> A

```
     p q  r
[1,] 1 2  3
[2,] 4 5  6
[3,] 7 8 10
```

> A*2

```
     p  q  r
[1,]  2  4  6
[2,]  8 10 12
[3,] 14 16 20
```

II. Add the values of matrix with any integer

> A

```
     p q  r
[1,] 1 2  3
[2,] 4 5  6
[3,] 7 8 10
```

> A+10

```
     p  q  r
[1,] 11 12 13
[2,] 14 15 16
[3,] 17 18 20
```

III. Retrieve the value in the second row 4 column

> t

```
     p  q  r  s
[1,] 16 20 24 28
[2,] 17 21 25 29
[3,] 18 22 26 30
[4,] 19 23 27 31
```

> t[2,4]

```
 s
29
```

IV. **Retrieve the values in the second row**

> t[2,]

```
 p  q  r  s
17 21 25 29
```

V. **Apply sum() function to any one of the row and check the output**

> A

```
  p q  r
[1,] 1 2  3
[2,] 4 5  6
[3,] 7 8 10
```

> e<-sum(A[2,])

```
> e
[1] 15
```

VI. **Apply sum() function to any one of the column and check the output**

```
> e<-sum(A[,3])
> e
[1] 19
```

21. Create the following matrix.

matrix4 <- matrix(c(3,4,6,8,5,4,8,4,7,1,2,2,5,4,6,4,7,5,2,5), nrow=5)

> matrix4

```
     [,1] [,2] [,3] [,4]
[1,]    3    4    2    4
[2,]    4    8    2    7
[3,]    6    4    5    5
[4,]    8    7    4    2
[5,]    5    1    6    5
```

Now, create the another matrix by using the elements (shown in bold-italics) from matrix4. Use indexing.

> a<-c(matrix4[2,2],matrix4[2,3])
> b<-c(matrix4[3,2],matrix4[3,3])
> c<-c(matrix4[4,2],matrix4[4,3])
> d<-rbind(a,b,c)

```
> d
  [,1] [,2]
a    8    2
b    4    5
```

```
c  7  4
```

22. Append rows of matrices created in question 15 and create another matrix. Check the dimensions of newly created matrix

```
> a<-c(1,4,7)
> b<-c(2,5,8)
> c<-c(3,6,10)
> d<-rbind(a,b,c)

> d
  [,1] [,2] [,3]
a    1    4    7
b    2    5    8
c    3    6   10

> e<-c(4,7,11)
> d<-rbind(a,b,c,e)


> d
  [,1] [,2] [,3]
a    1    4    7
b    2    5    8
c    3    6   10
e    4    7   11
```

> dim(d)
[1] 4 3

23. Append columns of matrices created in question 15 and create another matrix. Check the dimensions of newly created matrix

```
> a<-c(1,4,7)
> b<-c(2,5,8)
> c<-c(3,6,10)
> d<-cbind(a,b,c)
> d
```

```
     a b  c
[1,] 1 2  3
[2,] 4 5  6
[3,] 7 8 10
> e<-c(4,7,11)
> d<-cbind(a,b,c,e)
> d
     a b  c  e
[1,] 1 2  3  4
[2,] 4 5  6  7
[3,] 7 8 10 11
```

> dim(d)
[1] 3 4

24. Check the following commands and note the difference

- **A<-matrix(1:16, nrow=3,ncol=4)**

```
> A<-matrix(1:16, nrow=3,ncol=4)
```

Warning message: In matrix(1:16, nrow = 3, ncol = 4) : data length [16] is not a sub-multiple or multiple of the number of rows [3]

- **x<-1:16;dim(x)<-c(3,4)**

```
> x<-1:16;dim(x)<-c(3,4)
```

Error in dim(x) <- c(3, 4) : dims [product 12] do not match the length of object [16]

25. Create 3x10 matrix in such a way that first row elements are "1", second row elements are only "2" and third row elements are only "3". Use rep() command.

> A<-matrix(,3,10)
> A

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA
[2,]   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA
```

[3,] NA NA NA NA NA NA NA NA NA NA

> A[1,]<-rep(1,1,by=10)
> A[2,]<-rep(2,1,by=10)
> A[3,]<-rep(3,1,by=10)
> A

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    1    1    1    1    1    1    1     1
[2,]    2    2    2    2    2    2    2    2    2     2
[3,]    3    3    3    3    3    3    3    3    3     3
```

Section IV : Data Frames :~

26. Create two data frames as shown and store in objects called dataframe1 and dataframe2

>dataframe1<-
data.frame("WaterDepth"=c(50,75,100,150,200),"Temperature"=c(18,15,12,5,4))

> dataframe1

	Water.Depth	Temperature
1	50	18
2	75	15
3	100	12
4	150	5
5	200	4

>dataframe2<-
data.frame("Name"=c("Adam","Barbara","Charlie","Debbie","Elvis"),"Age"=c(39,23,30,31,28),"Location"=c("Bristol","Swansea","Derby","Belfast","Inverness"))

> dataframe2

	Name	Age	Location
1	Adam	39	Bristol
2	Barbara	23	Swansea
3	Charlie	30	Derby
4	Debbie	31	Belfast
5	Elvis	28	Inverness

27. Consider dataframe1 and do the following :

i. **Retrieve the rows whose values (Water Depth) are greater than and equal to 100**

> dataframe1[dataframe1$Water.Depth > 100,]

	Water.Depth	Temperature
4	150	5
5	200	4

ii. **Retrieve the rows whose values are above 5**

> dataframe1[dataframe1 > 5]

[1] 50 75 100 150 200 18 15 12

iii. **Retrieve the rows whose Temperature values are above 5**

```
> dataframe1[dataframe1$Temperature > 5,]
  Water.Depth Temperature
1          50          18
2          75          15
3         100          12
```

28. Consider dataframe2 and do the following :

i. **Retrieve the rows whose name value is equal to "Adam"**

```
> adam_dataframe2<-subset(dataframe2,Name=="Adam")
> adam_dataframe2
   Name  Age  Location
1  Adam   39   Bristol
```

ii. Retrieve the rows whose ages are between 20 and 30

```
> dataframe2[dataframe2$Age>20 & dataframe2$Age<30,]
    Name   Age  Location
```

2 Barbara 23 Swansea

5 Elvis 28 Inverness

iii.**Find the mean of the ages**

> mean(dataframe2$Age)

[1] 30.2

29. Create a 5x6 matrix "x" using matrix() fucntion. Create single integer vector "y" of 6 random elements from (1 to 6). Add the integer vector to the matrix using cbind() function and create a data frame using data.frame() function.

> data=matrix(1:6,5,6)

> data

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

> y

[1] 1 2 3 4 5 6

> z<-c(1,2,3,4,5,6)

> p<-c(1,2,3,4,5,6)

> q<-c(1,2,3,4,5,6)

> r<-c(1,2,3,4,5,6)

> s<-c(1,2,3,4,5,6)

> t<-cbind(data,y)

Warning message:

In cbind(data, y) :

number of rows of result is not a multiple of vector length (arg 2)

> t

y

[1,] 1 6 5 4 3 2 1

[2,] 2 1 6 5 4 3 2

[3,] 3 2 1 6 5 4 3

[4,] 4 3 2 1 6 5 4

[5,] 5 4 3 2 1 6 5

> dataframe3<-data.frame(t)

> dataframe3

```
  V1 V2 V3 V4 V5 V6 y
1  1  6  5  4  3  2 1
2  2  1  6  5  4  3 2
3  3  2  1  6  5  4 3
4  4  3  2  1  6  5 4
5  5  4  3  2  1  6 5
```

Extract the rows whose :

i. **7th column value is equal to 2**

```
> dataframe3[dataframe3$y==2,]
  V1 V2 V3 V4 V5 V6 y
2  2  1  6  5  4  3 2
```

ii. **7th column value is >2 and <=5**

```
> dataframe3[dataframe3$y>2 & dataframe3$y<=5,]
```

iii. V1 V2 V3 V4 V5 V6 y

iv. 3 3 2 1 6 5 4 3

v. 4 4 3 2 1 6 5 4

vi. 5 5 4 3 2 1 6 5

30. Load mtcars data frame into R console and retrieve the following data from the data table

i. Get the headers, dimensions and structure of mtcars data frame

> head(mtcars)

```
                   mpg cyl disp  hp drat    wt  qsec vs am
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0
                  gear carb
Mazda RX4            4    4
Mazda RX4 Wag        4    4
Datsun 710           4    1
Hornet 4 Drive       3    1
Hornet Sportabout    3    2
Valiant              3    1
```

> dim(mtcars)

[1] 32 11

> str(mtcars)

'data.frame': 32 obs. of 11 variables:

$ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...

$ cyl : num 6 6 4 6 8 6 8 4 4 6 ...

$ disp: num 160 160 108 258 360 ...

$ hp : num 110 110 93 110 175 105 245 62 95 123 ...

$ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...

$ wt : num 2.62 2.88 2.32 3.21 3.44 ...

$ qsec: num 16.5 17 18.6 19.4 17 ...

$ vs : num 0 0 1 1 0 1 0 1 1 1 ...

$ am : num 1 1 1 0 0 0 0 0 0 0 ...

$ gear: num 4 4 4 3 3 3 3 4 4 4 ...

$ carb: num 4 4 1 1 2 1 4 2 2 4 ...

ii. Get first 15 rows and 15 columns

> head(mtcars,15)

```
                   mpg cyl  disp  hp drat    wt  qsec vs am
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0
Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0

                  gear carb
Mazda RX4            4    4
Mazda RX4 Wag        4    4
Datsun 710           4    1
Hornet 4 Drive       3    1
Hornet Sportabout    3    2
Valiant              3    1
Duster 360           3    4
Merc 240D            4    2
```

```
Merc 230              4    2
Merc 280              4    4
Merc 280C             4    4
Merc 450SE            3    3
Merc 450SL            3    3
Merc 450SLC           3    3
Cadillac Fleetwood    3    4
```

> head(mtcars,,15)

```
                   mpg cyl disp  hp drat    wt  qsec vs am
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0

                  gear carb
Mazda RX4            4    4
Mazda RX4 Wag        4    4
Datsun 710           4    1
Hornet 4 Drive       3    1
Hornet Sportabout    3    2
Valiant              3    1
```

Get first column of first 20 rows

> head(mtcars [1],20)

```
                   mpg
Mazda RX4         21.0
Mazda RX4 Wag     21.0
Datsun 710        22.8
Hornet 4 Drive    21.4
Hornet Sportabout 18.7
Valiant           18.1
Duster 360        14.3
Merc 240D         24.4
Merc 230          22.8
Merc 280          19.2
Merc 280C         17.8
Merc 450SE        16.4
Merc 450SL        17.3
Merc 450SLC       15.2
Cadillac Fleetwood  10.4
Lincoln Continental 10.4
Chrysler Imperial   14.7
Fiat 128          32.4
Honda Civic       30.4
Toyota Corolla    33.9
```

iii. Get second column of first 20 rows

> head(mtcars [2],20)

```
                  cyl
Mazda RX4           6
Mazda RX4 Wag       6
Datsun 710          4
Hornet 4 Drive      6
Hornet Sportabout   8
Valiant             6
Duster 360          8
Merc 240D           4
Merc 230            4
Merc 280            6
Merc 280C           6
Merc 450SE          8
Merc 450SL          8
Merc 450SLC         8
Cadillac Fleetwood  8
Lincoln Continental 8
Chrysler Imperial   8
Fiat 128            4
Honda Civic         4
Toyota Corolla      4
```

iv. Get only second column

> mtcars [2]

```
                  cyl
Mazda RX4           6
Mazda RX4 Wag       6
Datsun 710          4
Hornet 4 Drive      6
Hornet Sportabout   8
Valiant             6
Duster 360          8
Merc 240D           4
Merc 230            4
Merc 280            6
Merc 280C           6
Merc 450SE          8
Merc 450SL          8
Merc 450SLC         8
Cadillac Fleetwood  8
Lincoln Continental 8
```

```
Chrysler Imperial     8
Fiat 128              4
Honda Civic           4
Toyota Corolla        4
Toyota Corona         4
Dodge Challenger      8
AMC Javelin           8
Camaro Z28            8
Pontiac Firebird      8
Fiat X1-9             4
Porsche 914-2         4
Lotus Europa          4
Ford Pantera L        8
Ferrari Dino          6
Maserati Bora         8
Volvo 142E            4
```

v. Get only alternate rows and all its corresponding columns

> mtcars[seq(1, nrow(mtcars),2),]

```
                   mpg cyl  disp  hp drat    wt  qsec vs am
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0
Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0
Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0
Honda Civic       30.4   4  75.7  52 4.93 1.615 18.52  1  1
Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0
AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30  0  0
Pontiac Firebird  19.2   8 400.0 175 3.08 3.845 17.05  0  0
Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0  1
Ford Pantera L    15.8   8 351.0 264 4.22 3.170 14.50  0  1
Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1

                  gear carb
Mazda RX4            4    4
Datsun 710           4    1
Hornet Sportabout    3    2
Duster 360           3    4
Merc 230             4    2
Merc 280C            4    4
Merc 450SL           3    3
Cadillac Fleetwood   3    4
Chrysler Imperial    3    4
```

```
Honda Civic            4    2
Toyota Corona          3    1
AMC Javelin            3    2
Pontiac Firebird       3    2
Porsche 914-2          5    2
Ford Pantera L         5    4
Maserati Bora          5    8
```

vi. Get the value of third row, fifth column
> head(mtcars [5],3)

	drat
Mazda RX4	3.90
Mazda RX4 Wag	3.90
Datsun 710	3.85

31. Write a command for replacing the NA's with sum values in the following data.frame

> foo<-data.frame("Day"=c(1,3,5,7),"Od"=c("NA","NA","NA","NA"))

> foo

	Day	Od
1	1	NA
2	3	NA
3	5	NA
4	7	NA

> foo[,2]<-c("SUM")
> foo

```
  Day  Od
1   1   SUM
2   3   SUM
3   5   SUM
4   7   SUM
```

"OR"

> foo<-data.frame("Day"=c(1,3,5,7),"Od"=c("NA","NA","NA","NA"))

> foo

```
  Day Od
1   1 NA
2   3 NA
3   5 NA
4   7 NA
```

> sum(foo$Day)

[1] 16

> foo[,2]<-c(16)

> foo

	Day	Od
1	1	16
2	3	16
3	5	16
4	7	16