2021bcs0175
Hazarathayya

# R PROGRAMMING LAB

Q1. Create a vector named as rainfall 10 elements and perform following operation
Rainfall = (0.1, 0.6, 33.8, 1.9. 9.7, 4.5, 0.3, 0.4, 0.1, 0.0)
I. Display the vector
II. Find the mean and standard deviation of rainfall
III. Find the cumulative sum
IV. Find highest rainfall
V. Creating a subset of rainfall having data greater 25
VI. Find mean rainfall for days where rainfall was at least 5
VII. Find subset of vector where rainfall is either exactly 0 or exactly 0.5

```
> Rainfall = c(0.1, 0.6, 33.8, 1.9, 9.7, 4.5, 0.3, 0.4, 0.1, 0.0)
> Rainfall
 [1]  0.1  0.6 33.8  1.9  9.7  4.5  0.3  0.4  0.1  0.0
>
> # Calculate mean and standard deviation
> Rainfall.mean = mean(Rainfall)
> Rainfall.standardDeviation = sd(Rainfall)
> Rainfall.mean
[1] 5.14
> Rainfall.standardDeviation
[1] 10.52417
>
> # Calculate Sum
> Rainfall.sum = sum(Rainfall)
> Rainfall.sum
[1] 51.4
>
> # Highest Rainfall
> Rainfall.max = max(Rainfall)
> Rainfall.max
[1] 33.8
>
> # Create a subset having data greater than 25
> Rainfall.greaterThan25 = subset(Rainfall, Rainfall>25)
> Rainfall.greaterThan25
[1] 33.8
>
> # mean rainfall of subset rainfall > 5
> Rainfall.meanofRainfallGreaterThan5 = mean(subset(Rainfall, Rainfall>5))
> Rainfall.meanofRainfallGreaterThan5
[1] 21.75
>
> # subset of rainfall either exactly 0 or 0.5
> Rainfall.subset1 = subset(Rainfall, Rainfall==0 | Rainfall == 0.5)
> Rainfall.subset1
[1] 0
```

Q2. Create two vectors as cylindrical length and cylindrical diameter in centimetre
I. Length = (2.1, 3.4, 2.8, 2.9, 3.1)
II. Diameter = (0.3, 0.5, 0.6, 0.9, 1)
III. Calculate correlation between both vectors
IV. Calculate volume of each cylinder and find the mean and standard deviation of volume
V. Convert each measurement for centimetre to meter. Find volume with new values. Find mean and standard deviation of new volume
VI. Calculate difference between old and new volume.

```
> # Q2.
> length = c(2.1, 3.4, 2.8, 2.9, 3.1)
> diameter = c(0.3, 0.5, 0.6, 0.9, 1)
>
> # Correlation between length and diameter
> corr = cor(length, diameter)
> corr
[1] 0.5069582
>
> # Calculate Volume, mean and sd of volume
> vol = pi*(diameter/2)*(diameter/2)*length
> vol
[1] 0.1484403 0.6675884 0.7916813 1.8449003 2.4347343
> vol.mean = mean(vol)
> vol.sd = sd(vol)
> vol.mean
[1] 1.177469
> vol.sd
[1] 0.9345976
>
>
>
> # Change the measurements from centimeter to meter
> length.m = length*0.01
> length.m
[1] 0.021 0.034 0.028 0.029 0.031
> diameter.m = diameter*0.01
> diameter.m
[1] 0.003 0.005 0.006 0.009 0.010
> vol.m = pi*(diameter.m/2)*(diameter.m/2)*length.m
> vol.m
[1] 1.484403e-07 6.675884e-07 7.916813e-07 1.844900e-06 2.434734e-06
> vol.m.mean = mean(vol.m)
> vol.m.mean
[1] 1.177469e-06

> vol.m.sd = sd(vol.m)
> vol.m.sd
[1] 9.345976e-07
>
> # difference between volume in meters and centimeters
> diff = vol.m - vol
> diff
[1] -0.1484401 -0.6675878 -0.7916806 -1.8448984 -2.4347319
>
```

Q3. Create two sets as x and y with five elements
x = (2, 3, 6, 10, 12)
y = (3, 5, 2, 0, 24)
I. Find the union of set x and y
II. Find the difference between set x and y
III. Construct vector of all values and compare(by subtraction) it with c(x, y)

```
>
> # Q3.
> x = c(2, 3, 6, 10, 12)
> y = c(3, 5, 2, 0, 24)
>
> # Union of x, y
> union.x.y = union(x, y)
> union.x.y
[1]  2  3  6 10 12  5  0 24
>
> # difference between set x and y
> diffofxy = diff(x, y)
Error in diff.default(x, y) :
  'lag' and 'differences' must be integers >= 1
> diffofxy
Error: object 'diffofxy' not found
>
> # Construct vector of all values and compare(by subtraction) it with c(x, y)
>
> Combined = c(x, y)
> Combined
 [1]  2  3  6 10 12  3  5  2  0 24
> Combined - x
 [1]  0  0  0  0  0  1  2 -4 -10  12
> Combined - y
 [1] -1 -2  4 10 -12  0  0  0  0  0
>
```

Q4. Create a matrix of 10×10 with values between 0 and 1.

I. Calculate row means and standard deviation across row means

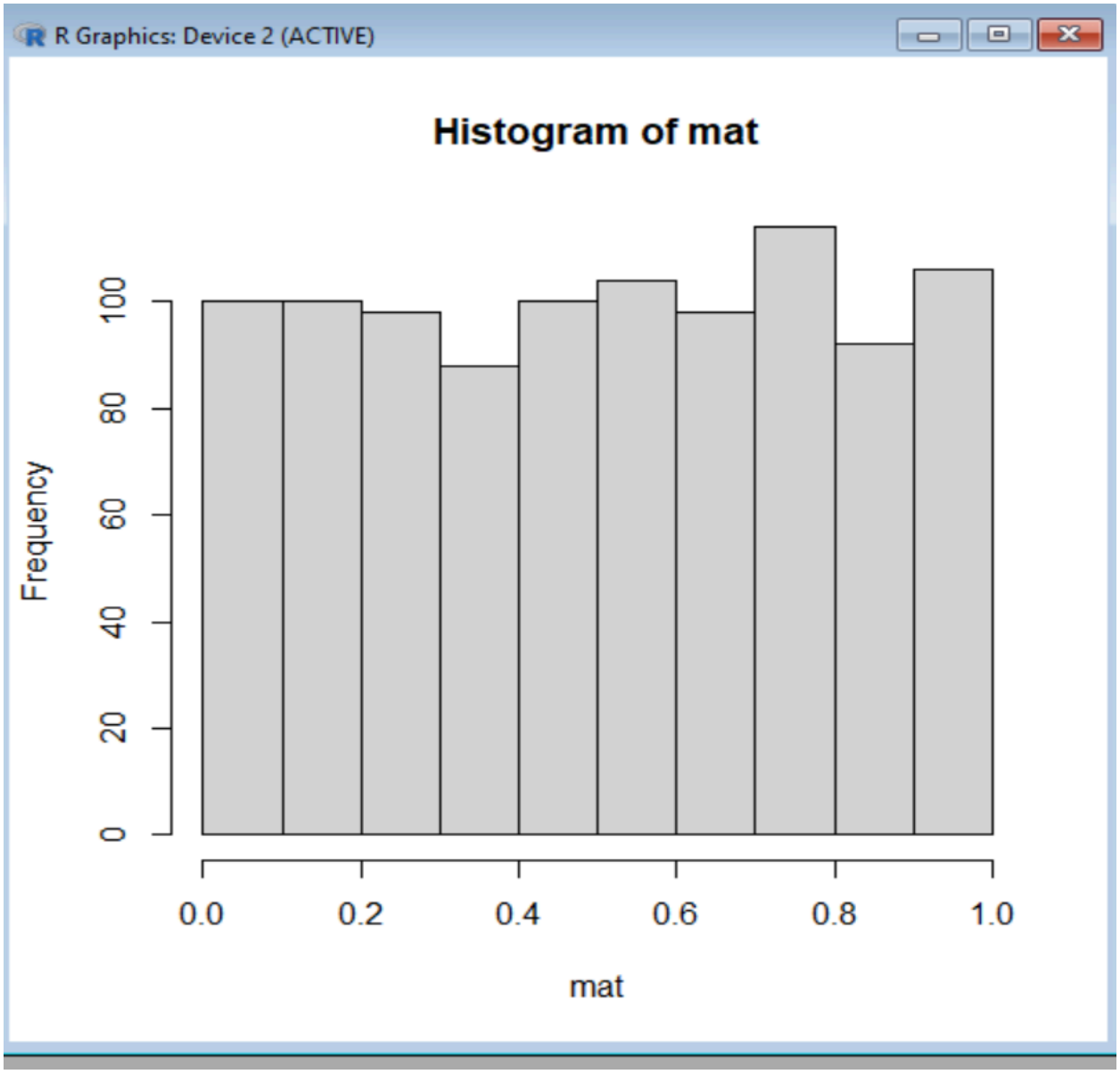
```
> # Q4.
> # Create a matrix of 10×10 with values between 0 and 1.
> mat = matrix(runif(100), 10, 10)
> mat
           [,1]       [,2]       [,3]      [,4]       [,5]       [,6]      [,7]
 [1,] 0.4773072 0.74839875 0.99453639 0.2201086 0.98058558 0.18840646 0.6691936
 [2,] 0.6622326 0.76978917 0.80702780 0.1751860 0.41374474 0.87478642 0.2181383
 [3,] 0.3353039 0.86392574 0.94033176 0.9229783 0.06157466 0.18096556 0.2401334
 [4,] 0.6746523 0.47138804 0.68629637 0.7271311 0.73036692 0.22729506 0.5100269
 [5,] 0.4698453 0.93846160 0.24114281 0.6643710 0.14369030 0.03036305 0.6191662
 [6,] 0.7343156 0.75814923 0.04602721 0.4964077 0.06192258 0.49424343 0.2311352
 [7,] 0.5898053 0.16490306 0.43945141 0.2199452 0.51778180 0.23153257 0.5483100
 [8,] 0.9107146 0.49257886 0.77934000 0.9546578 0.19631620 0.60246438 0.6537922
 [9,] 0.7489664 0.03760153 0.12756623 0.2455369 0.66659316 0.98159044 0.9309570
[10,] 0.1732784 0.79507394 0.63754906 0.6055494 0.65902214 0.77940956 0.9028807
           [,8]       [,9]      [,10]
 [1,] 0.27920091 0.98969130 0.91771135
 [2,] 0.14506429 0.82689736 0.63499805
 [3,] 0.33633729 0.95691994 0.33404346
 [4,] 0.33048582 0.22038764 0.26084244
 [5,] 0.64321556 0.53944514 0.51309798
 [6,] 0.81963899 0.06029438 0.01104557
 [7,] 0.27388888 0.04005492 0.90932531
 [8,] 0.92510367 0.83704635 0.73496445
 [9,] 0.09825632 0.57740002 0.31016610
[10,] 0.62571078 0.63794292 0.39754448
> # I. Calculate row means and standard deviation across row means
> mean = matrix(0, dim(mat)[1])
> for(x in 1:dim(mat)[1]){
+ mean[x] = mean(mat[x,])
+ }
> mean
           [,1]
 [1,] 0.6465140
 [2,] 0.5527865
 [3,] 0.5172514
 [4,] 0.4838873
 [5,] 0.4802799
 [6,] 0.3713180
 [7,] 0.3934998
 [8,] 0.7086978
 [9,] 0.4724634
[10,] 0.6213961
> std = matrix(0, dim(mat)[2])
> for(x in 1:dim(mat)[1]){
+ std[x] = sd(mat[x,])
+ }
> std
           [,1]
 [1,] 0.3316911
 [2,] 0.2885651
 [3,] 0.3580854
 [4,] 0.2125696
 [5,] 0.2722842
 [6,] 0.3267480
 [7,] 0.2568996
 [8,] 0.2342446
 [9,] 0.3528463
[10,] 0.2074997
>
```

II. Update matrix size as 100×10 and find the column means

```
>
> # II. Update matrix size as 100×10 and find the column means
> mat = matrix(runif(100*10), 100, 10)
> mat
             [,1]        [,2]        [,3]       [,4]       [,5]        [,6]
  [1,] 0.18052882 0.129351665 0.281023914 0.07954610 0.64037127 0.237715911
  [2,] 0.07413301 0.295603134 0.749943423 0.56366063 0.73045628 0.435443395
  [3,] 0.13321588 0.101136556 0.264283398 0.54534578 0.17214692 0.960552709
  [4,] 0.07146258 0.272146015 0.576604237 0.36270739 0.85925632 0.415783741
  [5,] 0.69618780 0.473339200 0.370324577 0.45215654 0.40036011 0.881637322
  [6,] 0.96110682 0.921951404 0.313945670 0.64827880 0.24039922 0.205186487
  [7,] 0.40269620 0.920513716 0.303385211 0.61114020 0.74441615 0.662817269
  [8,] 0.54860286 0.682027614 0.758240031 0.74361527 0.09040464 0.482158899
  [9,] 0.44292978 0.196207820 0.923926756 0.24798771 0.95810637 0.782120949
 [10,] 0.44810052 0.377558424 0.662448991 0.69962426 0.31763066 0.960277302
 [11,] 0.94423417 0.143406736 0.135647196 0.81255817 0.93235797 0.373534066
 [12,] 0.49983394 0.357219750 0.874241897 0.44131582 0.13353568 0.821676635
 [13,] 0.22492043 0.822382084 0.654830208 0.90964363 0.30516752 0.679755931
 [14,] 0.80423684 0.781810199 0.980540757 0.68284819 0.92512568 0.894262919
 [15,] 0.06934701 0.970331848 0.040335020 0.24273496 0.39853520 0.141082554
 [16,] 0.81103292 0.475400107 0.602461624 0.02540555 0.80211801 0.667741456
 [17,] 0.09195146 0.485421972 0.161909269 0.33362034 0.58260245 0.594422956
 [18,] 0.54136382 0.671552882 0.656336645 0.70836550 0.78919215 0.796959519
 [19,] 0.60271927 0.039747029 0.607945061 0.82536172 0.45275909 0.364051151
 [20,] 0.63717509 0.177991494 0.051344265 0.79663895 0.33793698 0.261970151
 [21,] 0.48968639 0.174099489 0.803303421 0.18157219 0.95279616 0.205933424
 [22,] 0.24481028 0.701105792 0.115914223 0.55605379 0.84351071 0.866696431
 [23,] 0.92588633 0.378910437 0.331331531 0.04175698 0.06219766 0.161251135
 [24,] 0.87842397 0.354536125 0.680916745 0.72617517 0.67981156 0.825804939
 [25,] 0.80064219 0.549745136 0.877407057 0.47982153 0.09427271 0.769268511
 [26,] 0.32974204 0.096628246 0.706183012 0.45334135 0.83770555 0.137065190
 [27,] 0.99246704 0.761611464 0.897087922 0.65222457 0.18118587 0.253136061
 [28,] 0.76626337 0.421935598 0.792591530 0.70431625 0.67684861 0.875706383
 [29,] 0.94796157 0.590852294 0.236923555 0.78473764 0.08540563 0.016138965
 [30,] 0.30031501 0.050209772 0.192341094 0.97896277 0.59245670 0.464204679
 [31,] 0.97139321 0.950244181 0.584138890 0.97815596 0.10311977 0.561902932
 [32,] 0.79785989 0.376867216 0.080663757 0.96803425 0.76970764 0.212288126
 [33,] 0.87891434 0.204111739 0.928011000 0.98971397 0.42820336 0.431037120
```

```
[100,] 0.729128062 0.44039119 0.319705358 0.757262820
> mean = matrix(0, dim(mat)[2])
> for(x in 1:dim(mat)[2]){
+ mean[x] = mean(mat[,x])
+ }
> mean
            [,1]
 [1,] 0.5354785
 [2,] 0.5114318
 [3,] 0.5213747
 [4,] 0.5164764
 [5,] 0.4567002
 [6,] 0.5144536
 [7,] 0.5288734
 [8,] 0.4844621
 [9,] 0.4826745
[10,] 0.5033064
>
```

III. Find the histogram of matrix



Q5. Load cereal dataset .

I. display starting rows with head and display details with str function

```
>
> #Q5. Load cereal dataset .
> # I. display starting rows with head and display details with str function
> df = read.csv("C:\\Users\\91630\\OneDrive\\Documents\\College\\SEM VI\\Data warehousing and data mining\\Labl\\cereal.csv")
> head(df)
                        name mfr type calories protein fat sodium fiber carbo
1                  100% Bran   N    C       70       4   1    130  10.0   5.0
2          100% Natural Bran   Q    C      120       3   5     15   2.0   8.0
3                   All-Bran   K    C       70       4   1    260   9.0   7.0
4 All-Bran with Extra Fiber   K    C       50       4   0    140  14.0   8.0
5             Almond Delight   R    C      110       2   2    200   1.0  14.0
6   Apple Cinnamon Cheerios   G    C      110       2   2    180   1.5  10.5
  sugars potass vitamins shelf weight cups   rating
1      6    280       25     3      1 0.33 68.40297
2      8    135        0     3      1 1.00 33.98368
3      5    320       25     3      1 0.33 59.42551
4      0    330       25     3      1 0.50 93.70491
5      8     -1       25     3      1 0.75 34.38484
6     10     70       25     1      1 0.75 29.50954
```

Data: df

| | name | mfr | type | calories | protein |
|---|---|---|---|---|---|
| 1 | 100% Bran | N | C | 70 | 4 |
| 2 | 100% Natural Bran | Q | C | 120 | 3 |
| 3 | All-Bran | K | C | 70 | 4 |
| 4 | All-Bran with Extra Fiber | K | C | 50 | 4 |
| 5 | Almond Delight | R | C | 110 | 2 |
| 6 | Apple Cinnamon Cheerios | G | C | 110 | 2 |
| 7 | Apple Jacks | K | C | 110 | 2 |
| 8 | Basic 4 | G | C | 130 | 3 |
| 9 | Bran Chex | R | C | 90 | 2 |
| 10 | Bran Flakes | P | C | 90 | 3 |
| 11 | Cap'n'Crunch | Q | C | 120 | 1 |
| 12 | Cheerios | G | C | 110 | 6 |
| 13 | Cinnamon Toast Crunch | G | C | 120 | 1 |
| 14 | Clusters | G | C | 110 | 3 |
| 15 | Cocoa Puffs | G | C | 110 | 1 |
| 16 | Corn Chex | R | C | 110 | 2 |
| 17 | Corn Flakes | K | C | 100 | 2 |
| 18 | Corn Pops | K | C | 110 | 1 |
| 19 | Count Chocula | G | C | 110 | 1 |

II. Add new variable as total corbs which is sum of carbs and sugar

```
> # II. Add new variable as total corbs which is sum of carbs and sugar
> df$totalcarbs = df$sugars + df$carbo
> df$totalcarbs
 [1] 11.0 16.0 12.0  8.0 22.0 20.5 25.0 26.0 21.0 18.0 24.0 18.0 22.0 20.0 25.0
[16] 25.0 23.0 25.0 25.0 17.0 21.0 24.0 21.0 23.0 24.0 25.0 21.0 22.0 26.0 25.0
[31] 26.0 24.0 20.0 20.0 17.0 23.0 21.5 25.0 23.0 29.0 24.0 18.0 24.0 19.0 27.0
[46] 27.0 30.0 21.0 24.0 28.0 20.0 23.5 25.0 23.0 13.0 10.0 20.0 -2.0 26.0 18.5
[61] 21.0 25.0 25.0 16.0 19.0 20.0 24.0 19.0 20.0 24.0 29.0 19.0 24.0 25.0 20.0
[76] 20.0 24.0
>
```

III. Counting cereals of type hot

```
>
> # III. Counting cereals of type hot
> sum(df$type == "H")
[1] 3
>
>
```

IV. Calculate unique manufacture

```
> unique(df$mfr)
[1] "N" "Q" "K" "R" "G" "P" "A"
> length(unique(df$mfr))
[1] 7
>
```

V. Calculate subset of cereals having calories less than 60 and vitamins greater than 20.

```
> # V. Calculate subset of cereals having calories less than 60 and vitamins greater than 20.
> subset(df, df$calories<60 & df$vitamins>20)
                      name mfr type calories protein fat sodium fiber carbo
4 All-Bran with Extra Fiber   K    C       50       4   0    140    14     8
  sugars potass vitamins shelf weight cups   rating totalcarbs
4      0    330       25     3      1  0.5 93.70491          8
>
```

VI. Rename column manufacturer and producer

```
>
> # VI. Rename column manufacturer and producer.
> names(df)[2] = "manufacturer and producer"
> names(df)
 [1] "name"                          "manufacturer and producer"
 [3] "type"                          "calories"
 [5] "protein"                       "fat"
 [7] "sodium"                        "fiber"
 [9] "carbo"                         "sugars"
[11] "potass"                        "vitamins"
[13] "shelf"                         "weight"
[15] "cups"                          "rating"
[17] "totalcarbs"
```

**THE END**