VIT-AP UNIVERSITY, ANDHRA PRADESH

CSE4027 – Data Analytics - Lab Sheet :3

Academic year: 2022-2023
Semester: Fall
Faculty Name: Dr Syed Khasim
Student name: Majjiga Jaswanth

Branch/ Class: B.Tech/M.Tech
Date: 23/09/22
School: SCOPE
Reg. no.: 20BCD7171

LAB3

Questions:

1. Find out mean of all columns of BMI dataset using for loop.

```
> setwd("C:/Users/Sashank K/Documents")
> d=read.csv("bmi_data.csv")
> meancol <- NULL
> for (m in names(d)){
+     if(class(d[,m]) == 'integer' | class(d[,m]) == 'numeric'){
+         meancol[m] <- mean(d[,m],na.rm = TRUE)
+     }
+ }
> meancol
            Age Height.Inches. Weight.Pounds.             BMI
       26.49960       67.99297      127.07821       19.32137
>
```

2. Create two vectors - height and weight of 20 students and convert it to a matrix

```
> Height <-as.vector(unlist(head(d,20)$Height.Inches))
> Weight <-as.vector(unlist(head(d,20)$Weight.Pounds))
> matrix_of_20 <- matrix(c(Height, Weight), ncol = 2)
> print(matrix_of_20)
          [,1]     [,2]
 [1,] 65.78331 112.9925
 [2,] 71.51521 136.4873
 [3,] 69.39874 153.0269
 [4,] 68.21660 142.3354
 [5,] 67.78781 144.2971
 [6,] 68.69784 123.3024
 [7,] 69.80204 141.4947
 [8,] 70.01472 136.4623
 [9,] 67.90265 112.3723
[10,] 66.78236 120.6672
[11,]       NA 127.4516
[12,] 67.62333 114.1430
[13,] 68.30248 125.6107
[14,] 67.11656 122.4618
[15,] 68.27967 116.0866
[16,] 71.09160 139.9975
[17,] 66.46100 129.5023
[18,] 68.64927 142.9733
[19,] 71.23033 137.9025
[20,] 67.13118 124.0449
> is.matrix(matrix_of_20)
[1] TRUE
>
```

3. Convert matrix into data frame and find the weight of 12th student

```
> dataframe_of_20 = as.data.frame(matrix_of_20)
> dataframe_of_20
          V1       V2
1   65.78331 112.9925
2   71.51521 136.4873
3   69.39874 153.0269
4   68.21660 142.3354
5   67.78781 144.2971
6   68.69784 123.3024
7   69.80204 141.4947
8   70.01472 136.4623
9   67.90265 112.3723
10  66.78236 120.6672
11        NA 127.4516
12  67.62333 114.1430
13  68.30248 125.6107
14  67.11656 122.4618
15  68.27967 116.0866
16  71.09160 139.9975
17  66.46100 129.5023
18  68.64927 142.9733
19  71.23033 137.9025
20  67.13118 124.0449
> dataframe_of_20[12, 2]
[1] 114.143
>
```

4. Categorize high and low based on bmi factor using if condition

```
> a<-d$BMI
> limit = length(a)
>
> for(i in d){
+     if(is.na(a[i]))
+     {
+         print(paste("NA"))}
+     else if(a[i]>19){
+         print(paste(a[i],"High"))
+     }
+     else
+     {
+         print(paste(a[i],"Low"))
+     }
+ }
[1] "NA"
   [1] "21.58686108 High" "19.10443381 High" "19.9523732 High"  "19.44857419 High" "21.32956107 High" "21.27149797 High"
   [7] "19.10443381 High" "19.10913596 High" "20.30278552 High" "21.80594279 High" "19.21760089 High" "20.30278552 High"
  [13] "21.58686108 High" "21.80594279 High" "19.9523732 High"  "17.08049361 High" "19.35214792 High" "15.02605989 High"
  [19] "19.10913596 High" "19.60075929 High" "21.27149797 High" "19.60075929 High" "15.02605989 High" "19.10443381 High"
  [25] "17.20559874 High" "19.44857419 High" "21.58686108 High" "21.80594279 High" "18.69791655 High" "15.02605989 High"
  [31] "21.80594279 High" "21.80594279 High" "20.19157888 High" "21.58686108 High" "20.19157888 High" "19.9523732 High"
  [37] "18.69791655 High" "21.58686108 High" "20.19157888 High" "17.08049361 High" "17.08049361 High" "15.02605989 High"
  [43] "21.32956107 High" "19.10913596 High" "20.30278552 High" "19.9523732 High"  "21.32956107 High" "15.02605989 High"
  [49] "21.58686108 High" "19.21760089 High" "19.21760089 High" "20.30278552 High" "17.20559874 High" "18.69791655 High"
  [55] "19.9523732 High"  "19.44857419 High" "20.30278552 High" "20.19157888 High" "21.32956107 High" "19.60075929 High"
  [61] "21.27149797 High" "21.58686108 High" "19.10913596 High" "19.44857419 High" "21.27149797 High" "20.19157888 High"
  [67] "20.30278552 High" "19.21760089 High" "15.02605989 High" "20.19157888 High" "17.08049361 High" "19.10443381 High"
  [73] "19.21760089 High" "21.27149797 High" "19.10913596 High" "19.60075929 High" "19.10913596 High" "19.21760089 High"
  [79] "21.58686108 High" "19.10913596 High" "19.21760089 High" "17.08049361 High" "15.02605989 High" "21.58686108 High"
  [85] "21.27149797 High" "19.60075929 High" "19.44857419 High" "19.9523732 High"  "18.69791655 High" "19.9523732 High"
  [91] "21.27149797 High" "19.35214792 High" "19.21760089 High" "19.22967413 High" "20.19157888 High" "19.60075929 High"
  [97] "21.32956107 High" "19.21760089 High" "17.08049361 High" "21.58686108 High" "19.10913596 High" "21.58686108 High"
 [103] "18.69791655 High" "21.58686108 High" "21.32956107 High" "15.02605989 High" "21.58686108 High" "20.30278552 High"
 [109] "19.9523732 High"  "18.69791655 High" "17.20559874 High" "19.21760089 High" "19.60075929 High" "19.44857419 High"
```

```
[505] "21.34085156 Low" "16.55910885 Low" "16.55910885 Low" "20.1689512 Low"  "20.34486076 Low" "20.34486076 Low" "16.11066092 Low"
[512] "16.11066092 Low" "20.34486076 Low" "20.34486076 Low" "16.55910885 Low" "NA Low"          "16.11066092 Low" "21.00269566 Low"
[519] "20.1689512 Low"  "16.11066092 Low" "20.34486076 Low" "21.00269566 Low" "20.34486076 Low" "18.97625203 Low" "20.1689512 Low"
[526] "20.34486076 Low" "16.55910885 Low" "16.11066092 Low" "18.97625203 Low" "16.55910885 Low" "16.11066092 Low" "21.00269566 Low"
[533] "18.97625203 Low" "16.11066092 Low" "16.55910885 Low" "16.11066092 Low" "21.00269566 Low" "16.11066092 Low" "20.34486076 Low"
[540] "21.34085156 Low" "16.55910885 Low" "16.55910885 Low" "20.34486076 Low" "20.34486076 Low" "16.55910885 Low" "21.34085156 Low"
[547] "16.55910885 Low" "16.11066092 Low" "20.34486076 Low" "18.97625203 Low" "16.55910885 Low" "21.34085156 Low" "21.00269566 Low"
[554] "20.34486076 Low" "20.34486076 Low" "21.00269566 Low" "21.00269566 Low" "16.11066092 Low" "21.00269566 Low" "16.11066092 Low"
[561] "NA Low"          "21.00269566 Low" "21.00269566 Low" "20.34486076 Low" "16.55910885 Low" "21.34085156 Low" "20.1689512 Low"
[568] "21.00269566 Low" "16.11066092 Low" "16.11066092 Low" "16.55910885 Low" "NA Low"          "20.34486076 Low" "20.34486076 Low"
[575] "16.11066092 Low" "20.34486076 Low" "18.97625203 Low" "16.11066092 Low" "18.97625203 Low" "20.34486076 Low" "16.11066092 Low"
[582] "21.00269566 Low" "NA Low"          "16.11066092 Low" "18.97625203 Low" "16.11066092 Low" "20.34486076 Low" "18.97625203 Low"
[589] "19.55484811 Low" "NA Low"          "20.34486076 Low" "20.34486076 Low" "21.00269566 Low" "16.55910885 Low" "16.11066092 Low"
[596] "18.97625203 Low" "16.55910885 Low" "21.00269566 Low" "NA Low"          "16.11066092 Low" "18.97625203 Low" "18.77453309 Low"
[603] "NA Low"          "21.00269566 Low" "16.55910885 Low" "16.11066092 Low" "20.34486076 Low" "20.34486076 Low" "16.55910885 Low"
[610] "NA Low"          "18.97625203 Low" "20.34486076 Low" "20.1689512 Low"  "16.11066092 Low" "16.55910885 Low" "NA Low"
[617] "16.11066092 Low" "NA Low"          "21.00269566 Low" "20.34486076 Low" "21.00269566 Low" "20.1689512 Low"  "16.55910885 Low"
[624] "16.55910885 Low" "18.97625203 Low" "16.55910885 Low" "16.55910885 Low" "NA Low"          "NA Low"          "20.1689512 Low"
[631] "20.34486076 Low" "21.00269566 Low" "NA Low"          "20.34486076 Low" "21.34085156 Low" "21.00269566 Low" "18.97625203 Low"
[638] "16.55910885 Low" "21.34085156 Low" "16.11066092 Low" "18.97625203 Low" "16.11066092 Low" "21.34085156 Low" "16.55910885 Low"
[645] "16.55910885 Low" "21.00269566 Low" "16.55910885 Low" "16.11066092 Low" "21.34085156 Low" "20.34486076 Low" "16.11066092 Low"
[652] "21.00269566 Low" "18.97625203 Low" "16.55910885 Low" "21.34085156 Low" "21.00269566 Low" "21.34085156 Low" "20.34486076 Low"
[659] "21.00269566 Low" "20.34486076 Low" "20.34486076 Low" "21.00269566 Low" "16.55910885 Low" "18.97625203 Low" "18.97625203 Low"
[666] "20.34486076 Low" "NA Low"          "18.97625203 Low" "16.11066092 Low" "20.34486076 Low" "NA Low"          "20.34486076 Low"
[673] "16.11066092 Low" "21.00269566 Low" "NA Low"          "NA Low"          "21.34085156 Low" "20.34486076 Low" "16.11066092 Low"
[680] "NA Low"          "21.00269566 Low" "21.00269566 Low" "20.34486076 Low" "16.55910885 Low" "18.77453309 Low" "20.34486076 Low"
[687] "NA Low"          "16.55910885 Low" "20.34486076 Low" "21.00269566 Low" "21.00269566 Low" "21.00269566 Low" "21.00269566 Low"
[694] "16.11066092 Low" "NA Low"          "NA Low"          "16.55910885 Low" "21.00269566 Low" "18.97625203 Low" "16.11066092 Low"
[701] "NA Low"          "21.00269566 Low" "20.34486076 Low" "21.00269566 Low" "18.97625203 Low" "18.97625203 Low" "20.34486076 Low"
[708] "18.77453309 Low" "16.55910885 Low" "16.11066092 Low" "20.1689512 Low"  "21.00269566 Low" "20.34486076 Low" "NA Low"
```

5. Using switch case categorize into

   Category

   BMI range - weight/height

   Severe Thinness < 16

   Moderate Thinness 16 - 17

   Mild Thinness 17 - 18.5

   Normal 18.5 - 25

   Overweight 25 - 30

   Obese Class I 30 - 35

   Obese Class II 35 - 40

   Obese Class III > 40

output:

```
> BMI_range <- data.frame(matrix(nrow = nrow(d)))
> for(i in 1:nrow(BMI_range)) {
+     if(!is.na(d$BMI[i])) {
+         bmi <- d$BMI[i]
+         x <- as.numeric(cut(bmi, c(-Inf,16,17,18.5,25,30,35,40,Inf)), right = TRUE)
+         BMI_range[i,] <- switch (x,
+                                  "Severe Thinness",
+                                  "Moderate Thinnes",
+                                  "Mild Thinness",
+                                  "Normal",
+                                  "Overweight",
+                                  "Obese Class I",
+                                  "Obese Class II",
+                                  "Obese Class III",
+         )
+     }
+ }
> View(BMI_range)
```

	matrix.nrow...nrow.d..
1	Mild Thinness
2	Normal
3	Normal
4	Normal
5	Normal
6	Mild Thinness
7	*NA*
8	Normal
9	Mild Thinness
10	Normal
11	*NA*
12	Mild Thinness
13	Normal
14	Normal
15	Mild Thinness
16	Normal
17	Normal
18	Normal
19	Normal
20	Normal
21	Normal
22	Normal
23	Mild Thinness
24	Normal
25	Normal
26	Normal

.....

	matrix.nrow...nrow.d..
24976	Normal
24977	Moderate Thinnes
24978	Moderate Thinnes
24979	Normal
24980	Normal
24981	Normal
24982	Mild Thinness
24983	Mild Thinness
24984	Normal
24985	Normal
24986	Normal
24987	Mild Thinness
24988	Normal
24989	Moderate Thinnes
24990	Normal
24991	Mild Thinness
24992	Mild Thinness
24993	Normal
24994	Normal
24995	Normal
24996	Mild Thinness
24997	Normal
24998	Normal
24999	Normal
25000	Normal