AMERICAN INTERNATIONAL UNIVERSITY - BANGLADESH

Faculty of Science and Information Technology

Assignment Cover Sheet

Assign./Case Title:	MID TERM ASSIGNMENT		
Assign./Case No:	01	Date of Submission:	18 March 2024
Course Title:	INTRODUCTION TO DATA SCIENCE		
Course Code:	CSC 4180	Section:	C
Semester:	Spring 2023-24	Degree Program:	BSc [CSE]
Course Teacher:	TOHEDUL ISLAM		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 18

No	Name	ID	Signature
1	Ashik Ahamed	21-45368-2	
2	Srabone Raxit	21-45038-2	
3			
4			
5			
6			

Faculty use only

FACULTY COMMENTS	Marks Obtained	
	Total Marks	

Data Set Description

The dataset from the provided link "*https://archive.ics.uci.edu/dataset/863/maternal+health+ris*" is titled "Maternal Health Risk." It aims to assess the risk of maternal morbidity and mortality by providing a comprehensive array of demographic, medical, and health-related features for pregnant women. The dataset includes the following columns:

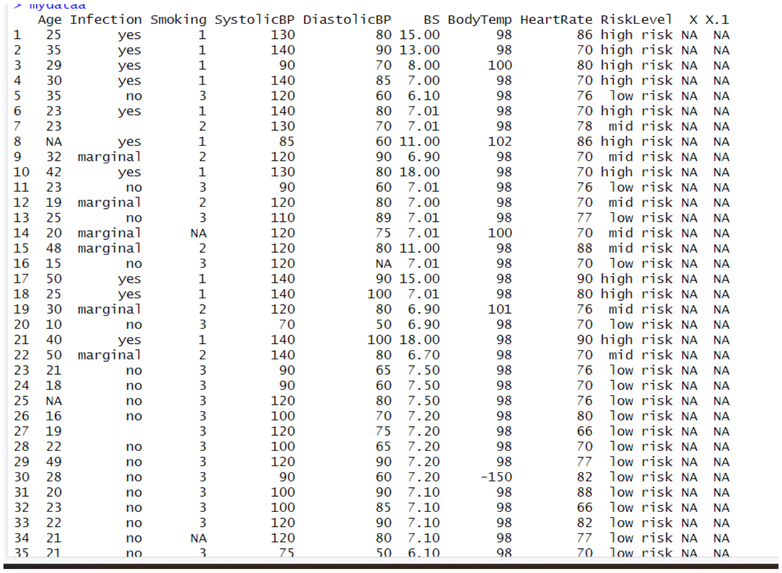
- Age: This column represents the age of the pregnant women, which is a key demographic factor in assessing maternal health risk.
- Infection: Indicates whether the pregnant women have any infections.
- Smoking: Indicates the smoking status of the pregnant women.
- SystolicBP: Represents the systolic blood pressure of pregnant women.
- DiastolicBP: Represents the diastolic blood pressure of pregnant women. Diastolic blood pressure, along with systolic blood pressure, is used to evaluate overall blood pressure.
- BS: Represents blood sugar levels or blood glucose levels.
- BodyTemp: Represents the body temperature of pregnant women.
- HeartRate: Represents the heart rate of pregnant women.
- RiskLevel: Indicates the risk level associated with each pregnant woman. Risk level assessment is crucial in prenatal care to identify high-risk pregnancies and provide appropriate management and interventions.

This dataset provides valuable insights into various maternal health risk factors and can be used to develop predictive models or risk assessment tools to improve maternal and fetal outcomes during pregnancy.

1. Dataset Include

```
mydataa<-read.csv("C:/Users/User/Desktop/Data Science/mid_project/Dataset_midterm_Section(C).csv",header=TRUE,sep=",")

mydataa
```

Output:



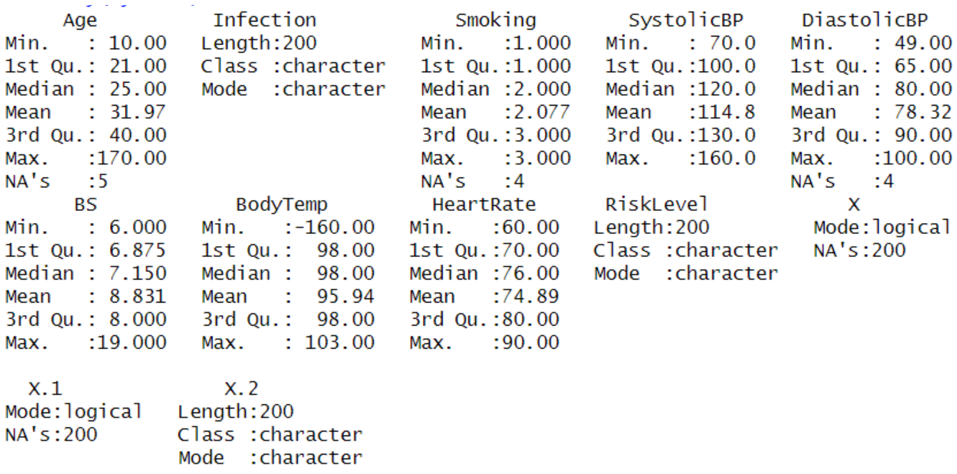
```
> mydataa
   Age Infection Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel  X X.1
1   25       yes       1        130          80 15.00       98        86 high risk NA  NA
2   35       yes       1        140          90 13.00       98        70 high risk NA  NA
3   29       yes       1         90          70  8.00      100        80 high risk NA  NA
4   30       yes       1        140          85  7.00       98        70 high risk NA  NA
5   35        no       3        120          60  6.10       98        76  low risk NA  NA
6   23       yes       1        140          80  7.01       98        70 high risk NA  NA
7   23                 2        130          70  7.01       98        78  mid risk NA  NA
8   NA       yes       1         85          60 11.00      102        86 high risk NA  NA
9   32  marginal       2        120          90  6.90       98        70  mid risk NA  NA
10  42       yes       1        130          80 18.00       98        70 high risk NA  NA
11  23        no       3         90          60  7.01       98        76  low risk NA  NA
12  19  marginal       2        120          80  7.00       98        70  mid risk NA  NA
13  25        no       3        110          89  7.01       98        77  low risk NA  NA
14  20  marginal      NA        120          75  7.01      100        70  mid risk NA  NA
15  48  marginal       2        120          80 11.00       98        88  mid risk NA  NA
16  15        no       3        120          NA  7.01       98        70  low risk NA  NA
17  50       yes       1        140          90 15.00       98        90 high risk NA  NA
18  25       yes       1        140         100  7.01       98        80 high risk NA  NA
19  30  marginal       2        120          80  6.90      101        76  mid risk NA  NA
20  10        no       3         70          50  6.90       98        70  low risk NA  NA
21  40       yes       1        140         100 18.00       98        90 high risk NA  NA
22  50  marginal       2        140          80  6.70       98        70  mid risk NA  NA
23  21        no       3         90          65  7.50       98        76  low risk NA  NA
24  18        no       3         90          60  7.50       98        70  low risk NA  NA
25  NA        no       3        120          80  7.50       98        76  low risk NA  NA
26  16        no       3        100          70  7.20       98        80  low risk NA  NA
27  19                 3        120          75  7.20       98        66  low risk NA  NA
28  22        no       3        100          65  7.20       98        70  low risk NA  NA
29  49        no       3        120          90  7.20       98        77  low risk NA  NA
30  28        no       3         90          60  7.20     -150        82  low risk NA  NA
31  20        no       3        100          90  7.10       98        88  low risk NA  NA
32  23        no       3        100          85  7.10       98        66  low risk NA  NA
33  22        no       3        120          90  7.10       98        82  low risk NA  NA
34  21        no      NA        120          80  7.10       98        77  low risk NA  NA
35  21        no       3         75          50  6.10       98        70  low risk NA  NA
```

At first, the dataset is included and stored in mydataa. Then mydataa is executed to show all the data.

2.

summary(mydataa)

Output:



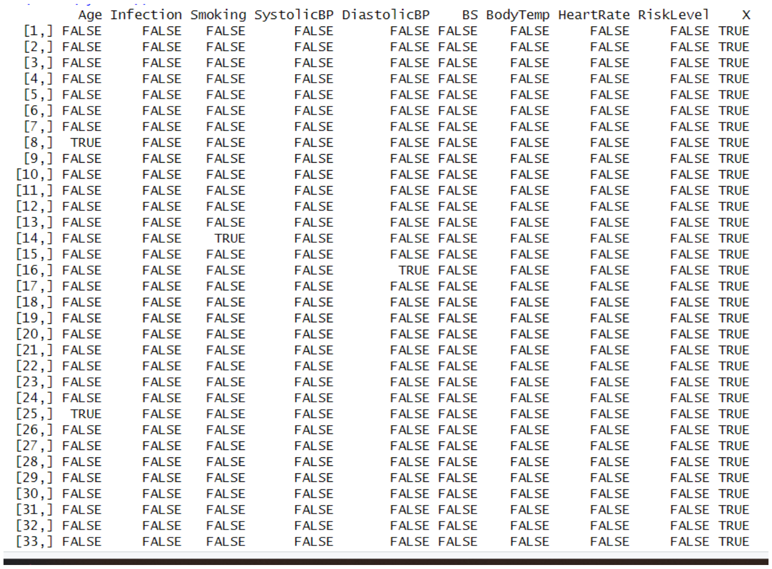
```
      Age           Infection            Smoking        SystolicBP       DiastolicBP    
 Min.   : 10.00   Length:200         Min.   :1.000   Min.   : 70.0   Min.   : 49.00  
 1st Qu.: 21.00   Class :character   1st Qu.:1.000   1st Qu.:100.0   1st Qu.: 65.00  
 Median : 25.00   Mode  :character   Median :2.000   Median :120.0   Median : 80.00  
 Mean   : 31.97                      Mean   :2.077   Mean   :114.8   Mean   : 78.32  
 3rd Qu.: 40.00                      3rd Qu.:3.000   3rd Qu.:130.0   3rd Qu.: 90.00  
 Max.   :170.00                      Max.   :3.000   Max.   :160.0   Max.   :100.00  
 NA's   :5                           NA's   :4                       NA's   :4       
       BS            BodyTemp         HeartRate      RiskLevel              X          
 Min.   : 6.000   Min.   :-160.00   Min.   :60.00   Length:200         Mode:logical  
 1st Qu.: 6.875   1st Qu.:  98.00   1st Qu.:70.00   Class :character   NA's:200      
 Median : 7.150   Median :  98.00   Median :76.00   Mode  :character                 
 Mean   : 8.831   Mean   :  95.94   Mean   :74.89                                    
 3rd Qu.: 8.000   3rd Qu.:  98.00   3rd Qu.:80.00                                    
 Max.   :19.000   Max.   : 103.00   Max.   :90.00                                    

   X.1               X.2           
 Mode:logical    Length:200        
 NA's:200        Class :character  
                 Mode  :character  
```

The function returns an overall summary like the minimum, maximum, mean, median, first & third quartiles for numerical values.

3.

is.na(mydataa)

Output:



```
        Age Infection Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel    X
 [1,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [2,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [3,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [4,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [5,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [6,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [7,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [8,]  TRUE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
 [9,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[10,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[11,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[12,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[13,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[14,] FALSE     FALSE    TRUE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[15,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[16,] FALSE     FALSE   FALSE      FALSE        TRUE FALSE    FALSE     FALSE     FALSE TRUE
[17,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[18,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[19,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[20,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[21,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[22,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[23,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[24,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[25,]  TRUE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[26,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[27,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[28,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[29,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[30,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[31,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[32,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
[33,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE
```

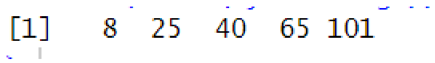
If there is any missing values, the function returns TRUE; otherwise FALSE for any numerical data. But for categorical data, it always return FALSE.

Here in dataset, for Infection and RiskLevel Column it is returning FALSE always.

4.

which(is.na(mydataa$Age))

Output:



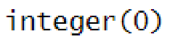
```
[1]   8  25  40  65 101
```

It returns the row number of missing values from AGE column.

5.

which(is.na(mydataa$Infection))

Output:



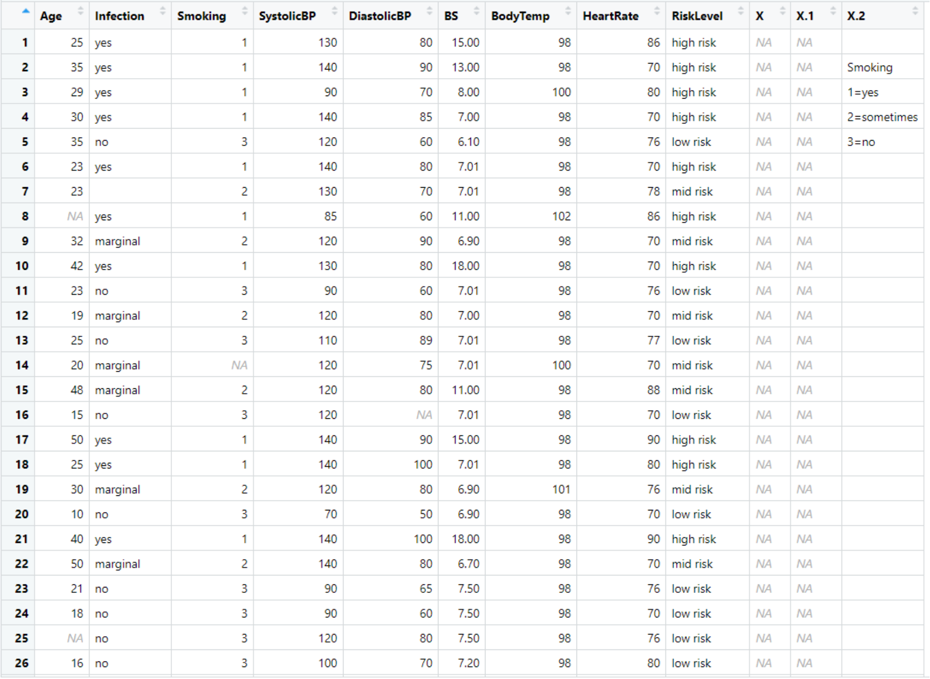
```
integer(0)
```

As the column Infection is filled with categorical data, it can't detect the missing values and showing the output.

6.

View(mydataa)

Output:



	Age	Infection	Smoking	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel	X	X.1	X.2
1	25	yes	1	130	80	15.00	98	86	high risk	NA	NA	
2	35	yes	1	140	90	13.00	98	70	high risk	NA	NA	Smoking
3	29	yes	1	90	70	8.00	100	80	high risk	NA	NA	1=yes
4	30	yes	1	140	85	7.00	98	70	high risk	NA	NA	2=sometimes
5	35	no	3	120	60	6.10	98	76	low risk	NA	NA	3=no
6	23	yes	1	140	80	7.01	98	70	high risk	NA	NA	
7	23		2	130	70	7.01	98	78	mid risk	NA	NA	
8	NA	yes	1	85	60	11.00	102	86	high risk	NA	NA	
9	32	marginal	2	120	90	6.90	98	70	mid risk	NA	NA	
10	42	yes	1	130	80	18.00	98	70	high risk	NA	NA	
11	23	no	3	90	60	7.01	98	76	low risk	NA	NA	
12	19	marginal	2	120	80	7.00	98	70	mid risk	NA	NA	
13	25	no	3	110	89	7.01	98	77	low risk	NA	NA	
14	20	marginal	NA	120	75	7.01	100	70	mid risk	NA	NA	
15	48	marginal	2	120	80	11.00	98	88	mid risk	NA	NA	
16	15	no	3	120	NA	7.01	98	70	low risk	NA	NA	
17	50	yes	1	140	90	15.00	98	90	high risk	NA	NA	
18	25	yes	1	140	100	7.01	98	80	high risk	NA	NA	
19	30	marginal	2	120	80	6.90	101	76	mid risk	NA	NA	
20	10	no	3	70	50	6.90	98	70	low risk	NA	NA	
21	40	yes	1	140	100	18.00	98	90	high risk	NA	NA	
22	50	marginal	2	140	80	6.70	98	70	mid risk	NA	NA	
23	21	no	3	90	65	7.50	98	76	low risk	NA	NA	
24	18	no	3	90	60	7.50	98	70	low risk	NA	NA	
25	NA	no	3	120	80	7.50	98	76	low risk	NA	NA	
26	16	no	3	100	70	7.20	98	80	low risk	NA	NA	

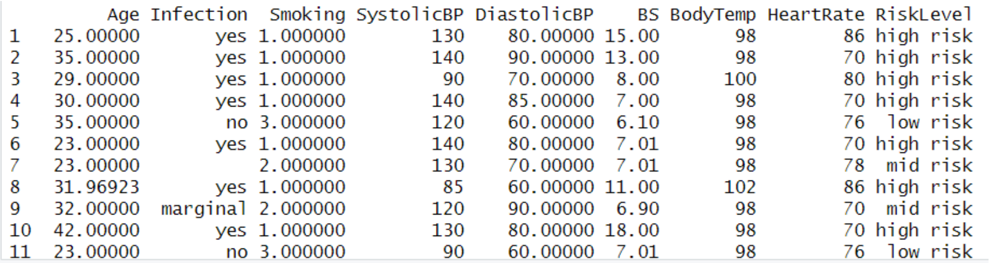
It open a separate window and show all the data like spreadsheets.

7.

newDataset <- mydataa[, c("Age", "Infection", "Smoking", "SystolicBP", "DiastolicBP",
                          "BS", "BodyTemp", "HeartRate", "RiskLevel")]

newDataset

Output:



```
          Age Infection  Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel
1   25.00000       yes 1.000000        130    80.00000 15.00       98        86 high risk
2   35.00000       yes 1.000000        140    90.00000 13.00       98        70 high risk
3   29.00000       yes 1.000000         90    70.00000  8.00      100        80 high risk
4   30.00000       yes 1.000000        140    85.00000  7.00       98        70 high risk
5   35.00000        no 3.000000        120    60.00000  6.10       98        76  low risk
6   23.00000       yes 1.000000        140    80.00000  7.01       98        70 high risk
7   23.00000           2.000000        130    70.00000  7.01       98        78  mid risk
8   31.96923       yes 1.000000         85    60.00000 11.00      102        86 high risk
9   32.00000  marginal 2.000000        120    90.00000  6.90       98        70  mid risk
10  42.00000       yes 1.000000        130    80.00000 18.00       98        70 high risk
11  23.00000        no 3.000000         90    60.00000  7.01       98        76  low risk
```

Make new data set using usable column and stored them on newDataset variable. This is the visual presentation.

8. Handle Missing Value replace by Average Value

newDataset <- newDataset %>%

mutate_all(~ ifelse(is.na(.), mean(., na.rm = TRUE), .))

Here,

- mutate_all() applies the specified function to all columns of the data frame.
- ~ is used to create an anonymous function (lambda function) that takes each column as input.
- ifelse() is used to check if the value is missing (is.na(.)) and replace it with the mean of the column (mean(., na.rm = TRUE)) if it is, otherwise keep the original value (.).

9.

mydataaAvg <- as.data.frame(newDataset)

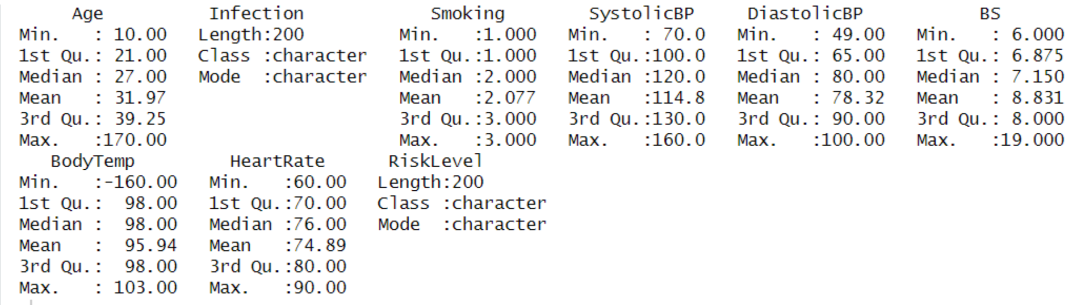
mydataaAvg

Here, the handled dataset is stored in new variable called mydataaAvg.

10.

summary(mydataaAvg)

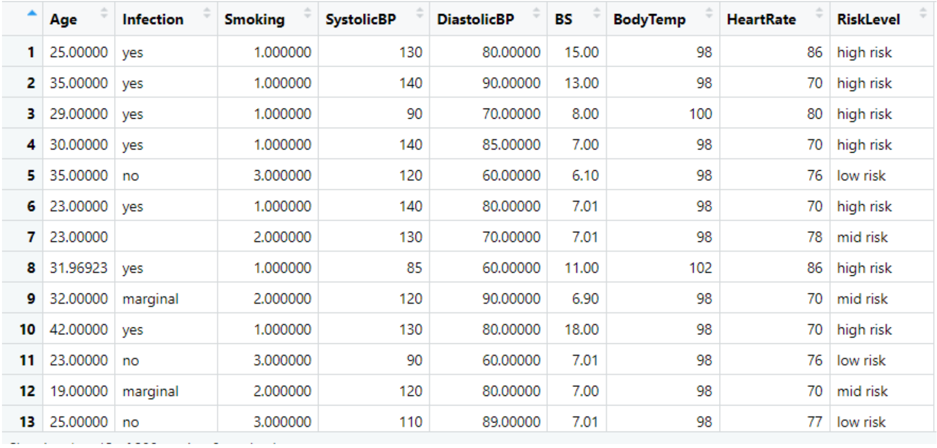
Output:



```
      Age           Infection            Smoking        SystolicBP      DiastolicBP           BS        
 Min.   : 10.00   Length:200         Min.   :1.000   Min.   : 70.0   Min.   : 49.00   Min.   : 6.000  
 1st Qu.: 21.00   Class :character   1st Qu.:1.000   1st Qu.:100.0   1st Qu.: 65.00   1st Qu.: 6.875  
 Median : 27.00   Mode  :character   Median :2.000   Median :120.0   Median : 80.00   Median : 7.150  
 Mean   : 31.97                      Mean   :2.077   Mean   :114.8   Mean   : 78.32   Mean   : 8.831  
 3rd Qu.: 39.25                      3rd Qu.:3.000   3rd Qu.:130.0   3rd Qu.: 90.00   3rd Qu.: 8.000  
 Max.   :170.00                      Max.   :3.000   Max.   :160.0   Max.   :100.00   Max.   :19.000  
    BodyTemp          HeartRate      RiskLevel        
 Min.   :-160.00   Min.   :60.00   Length:200        
 1st Qu.:  98.00   1st Qu.:70.00   Class :character  
 Median :  98.00   Median :76.00   Mode  :character  
 Mean   :  95.94   Mean   :74.89                     
 3rd Qu.:  98.00   3rd Qu.:80.00                     
 Max.   : 103.00   Max.   :90.00                     
```

View(mydataaAvg)

Output:



	Age	Infection	Smoking	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel
1	25.00000	yes	1.000000	130	80.00000	15.00	98	86	high risk
2	35.00000	yes	1.000000	140	90.00000	13.00	98	70	high risk
3	29.00000	yes	1.000000	90	70.00000	8.00	100	80	high risk
4	30.00000	yes	1.000000	140	85.00000	7.00	98	70	high risk
5	35.00000	no	3.000000	120	60.00000	6.10	98	76	low risk
6	23.00000	yes	1.000000	140	80.00000	7.01	98	70	high risk
7	23.00000		2.000000	130	70.00000	7.01	98	78	mid risk
8	31.96923	yes	1.000000	85	60.00000	11.00	102	86	high risk
9	32.00000	marginal	2.000000	120	90.00000	6.90	98	70	mid risk
10	42.00000	yes	1.000000	130	80.00000	18.00	98	70	high risk
11	23.00000	no	3.000000	90	60.00000	7.01	98	76	low risk
12	19.00000	marginal	2.000000	120	80.00000	7.00	98	70	mid risk
13	25.00000	no	3.000000	110	89.00000	7.01	98	77	low risk

The summary and View function showing the datas of mydataaAvg as like before.

11. Prepare Data before visualization

- numeric_columns <- mydataaAvg[, sapply(mydataaAvg, is.numeric)]

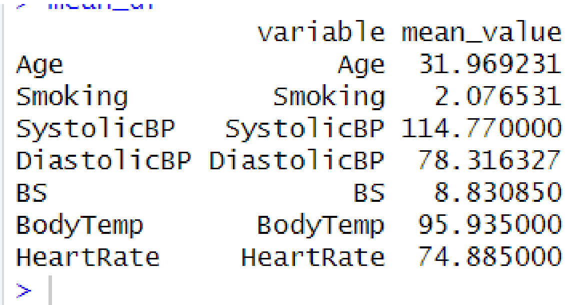
Filter those columns which have numeric values and stored in numeric_columns variable.

- mean_values <- colMeans(numeric_columns)

Calculate mean values for each numeric variable

- mean_df <- data.frame(variable = names(mean_values), mean_value = mean_values)
  mean_df

Output:



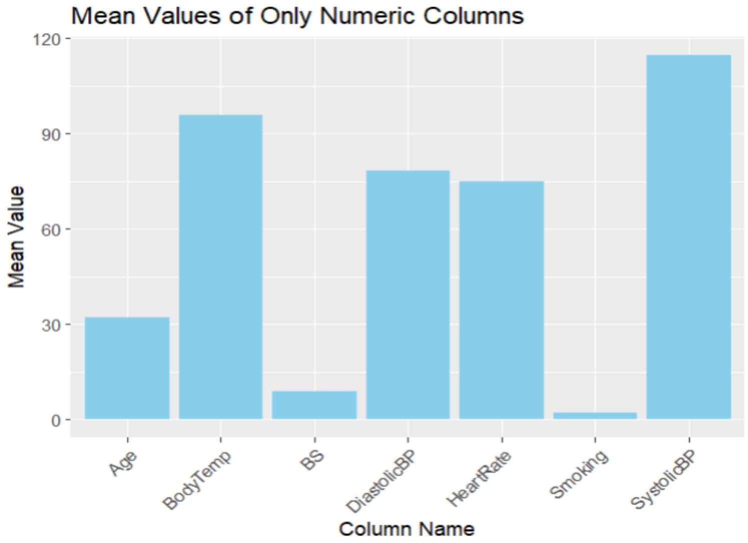
```
> mean_df
               variable mean_value
Age                 Age  31.969231
Smoking         Smoking   2.076531
SystolicBP   SystolicBP 114.770000
DiastolicBP DiastolicBP  78.316327
BS                   BS   8.830850
BodyTemp       BodyTemp  95.935000
HeartRate     HeartRate  74.885000
> 
```

Convert mean values to a data frame. We make average values from each column to generate a bar graph.

12. Visualization

```
ggplot(mean_df, aes(x = variable, y = mean_value)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Mean Values of Numeric Variables",
       x = "Variable",
       y = "Mean Value") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

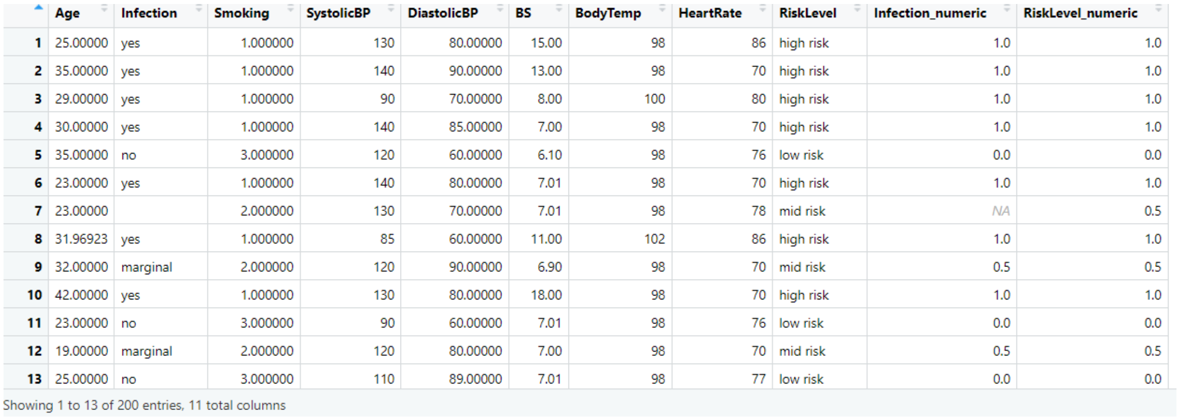
Output:



13. Convert Categorical Data to Numeric

```
# "Infection" column
mydataaAvgN <- mydataaAvg %>%
  mutate(Infection_numeric = case_when(
    Infection == "yes" ~ 1,
    Infection == "no" ~ 0,
    Infection == "marginal" ~ 0.5,
    TRUE ~ NA_real_  # For any other cases not specified
  ))
# "RiskLevel" column
mydataaAvgN <- mydataaAvg %>%
  mutate(RiskLevel_numeric = case_when(
    RiskLevel == "high risk" ~ 1,
    RiskLevel == "low risk" ~ 0,
    RiskLevel == "mid risk" ~ 0.5,
    TRUE ~ NA_real_  # For any other cases not specified
  ))
View(mydataaAvgN)
```

Output:



	Age	Infection	Smoking	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel	Infection_numeric	RiskLevel_numeric
1	25.00000	yes	1.000000	130	80.00000	15.00	98	86	high risk	1.0	1.0
2	35.00000	yes	1.000000	140	90.00000	13.00	98	70	high risk	1.0	1.0
3	29.00000	yes	1.000000	90	70.00000	8.00	100	80	high risk	1.0	1.0
4	30.00000	yes	1.000000	140	85.00000	7.00	98	70	high risk	1.0	1.0
5	35.00000	no	3.000000	120	60.00000	6.10	98	76	low risk	0.0	0.0
6	23.00000	yes	1.000000	140	80.00000	7.01	98	70	high risk	1.0	1.0
7	23.00000		2.000000	130	70.00000	7.01	98	78	mid risk	NA	0.5
8	31.96923	yes	1.000000	85	60.00000	11.00	102	86	high risk	1.0	1.0
9	32.00000	marginal	2.000000	120	90.00000	6.90	98	70	mid risk	0.5	0.5
10	42.00000	yes	1.000000	130	80.00000	18.00	98	70	high risk	1.0	1.0
11	23.00000	no	3.000000	90	60.00000	7.01	98	76	low risk	0.0	0.0
12	19.00000	marginal	2.000000	120	80.00000	7.00	98	70	mid risk	0.5	0.5
13	25.00000	no	3.000000	110	89.00000	7.01	98	77	low risk	0.0	0.0

Showing 1 to 13 of 200 entries, 11 total columns

The dataset is included with 2 new columns Infection_numeric and RiskLevel_numeric.

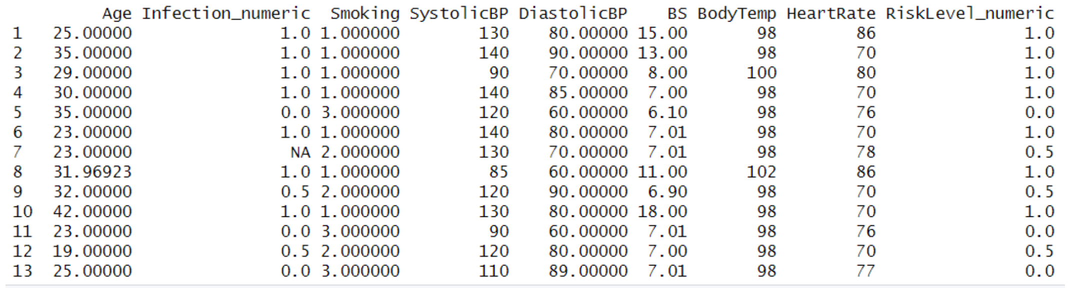
14. Truncating categorical data columns

newDatasetNumeric <- mydataaAvgN[, c("Age", "Infection_numeric", "Smoking", "SystolicBP", "DiastolicBP",

"BS", "BodyTemp", "HeartRate", "RiskLevel_numeric")]

newDatasetNumeric

Output:



```
        Age Infection_numeric  Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel_numeric
1  25.00000               1.0 1.000000        130    80.00000 15.00       98        86               1.0
2  35.00000               1.0 1.000000        140    90.00000 13.00       98        70               1.0
3  29.00000               1.0 1.000000         90    70.00000  8.00      100        80               1.0
4  30.00000               1.0 1.000000        140    85.00000  7.00       98        70               1.0
5  35.00000               0.0 3.000000        120    60.00000  6.10       98        76               0.0
6  23.00000               1.0 1.000000        140    80.00000  7.01       98        70               1.0
7  23.00000                NA 2.000000        130    70.00000  7.01       98        78               0.5
8  31.96923               1.0 1.000000         85    60.00000 11.00      102        86               1.0
9  32.00000               0.5 2.000000        120    90.00000  6.90       98        70               0.5
10 42.00000               1.0 1.000000        130    80.00000 18.00       98        70               1.0
11 23.00000               0.0 3.000000         90    60.00000  7.01       98        76               0.0
12 19.00000               0.5 2.000000        120    80.00000  7.00       98        70               0.5
13 25.00000               0.0 3.000000        110    89.00000  7.01       98        77               0.0
```

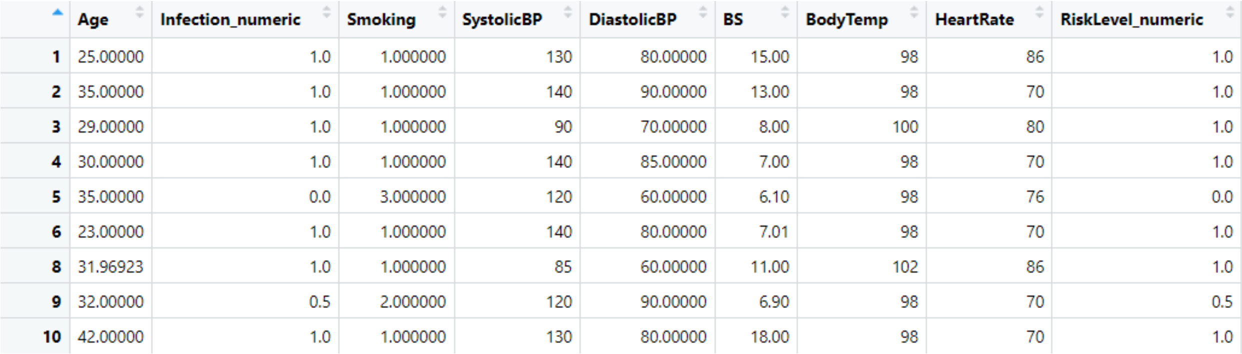
This is the new dataset only using Numerical datas. All categorical datas are truncated.

15. Deleting rows having missing values

newDatasetNumeric <- na.omit(newDatasetNumeric)

View(newDatasetNumeric)

Output:



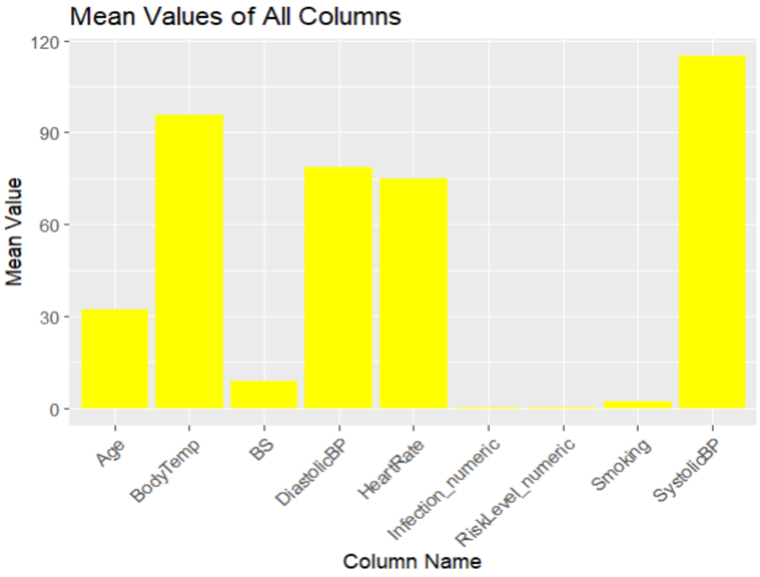
	Age	Infection_numeric	Smoking	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel_numeric
1	25.00000	1.0	1.000000	130	80.00000	15.00	98	86	1.0
2	35.00000	1.0	1.000000	140	90.00000	13.00	98	70	1.0
3	29.00000	1.0	1.000000	90	70.00000	8.00	100	80	1.0
4	30.00000	1.0	1.000000	140	85.00000	7.00	98	70	1.0
5	35.00000	0.0	3.000000	120	60.00000	6.10	98	76	0.0
6	23.00000	1.0	1.000000	140	80.00000	7.01	98	70	1.0
8	31.96923	1.0	1.000000	85	60.00000	11.00	102	86	1.0
9	32.00000	0.5	2.000000	120	90.00000	6.90	98	70	0.5
10	42.00000	1.0	1.000000	130	80.00000	18.00	98	70	1.0

Omit missing value row from the dataset.

16. Visualization of the New data set having no categorical data

```
numeric_columns2 <- newDatasetNumeric[, sapply(newDatasetNumeric, is.numeric)]
mean_values <- colMeans(numeric_columns2)
mean_df2 <- data.frame(variable = names(mean_values), mean_value = mean_values)
ggplot(mean_df2, aes(x = variable, y = mean_value)) +
  geom_bar(stat = "identity", fill = "yellow") +
  labs(title = "Mean Values of All Columns",
       x = "Column Name",
       y = "Mean Value") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Output:

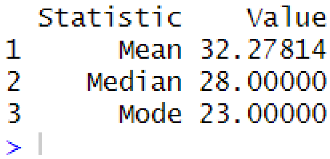


New bar graph using all columns where categorical datas are converted into numerical.

17. Mean, Median & Mode from Specific Columns

```
mean_age <- mean(newDatasetNumeric$Age)
median_age <- median(newDatasetNumeric$Age)
mode_age <- as.numeric(names(sort(-table(newDatasetNumeric$Age)))[1])

summary_stats <- data.frame(
  Statistic = c("Mean", "Median", "Mode"),
  Value = c(mean_age, median_age, mode_age)
)
print(summary_stats)
```

Output:



```
  Statistic    Value
1      Mean 32.27814
2    Median 28.00000
3      Mode 23.00000
> |
```

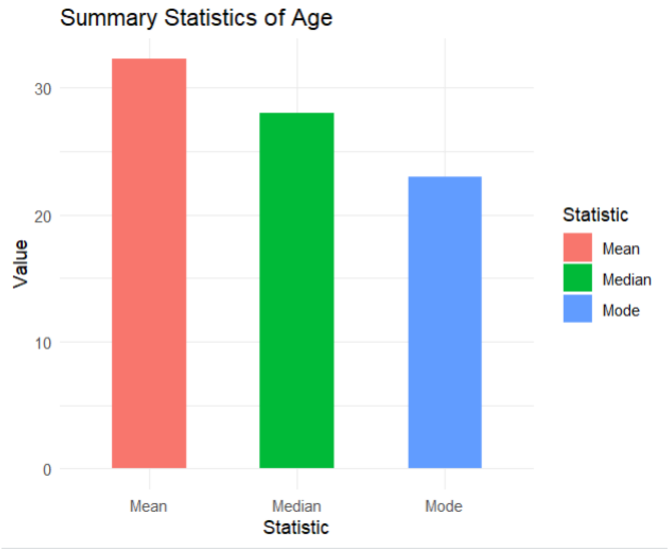
Here, the mean, median and mode values of Age column are stored in variables mean_age, median_age & mode_age. For calculating mode, the data are sorted in descending order. Then a data frame named summary_stats is created where Statistics hold variable names and Value hold the values. Then, the data frame is printed.

PS: We can do this for each column, but only Age column is showed here.

18. Graph Plot of summary_stats

```
library(ggplot2)
ggplot(summary_stats, aes(x = Statistic, y = Value, fill = Statistic)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "Summary Statistics of Age",
       x = "Statistic",
       y = "Value") +
  theme_minimal()
```

Output:



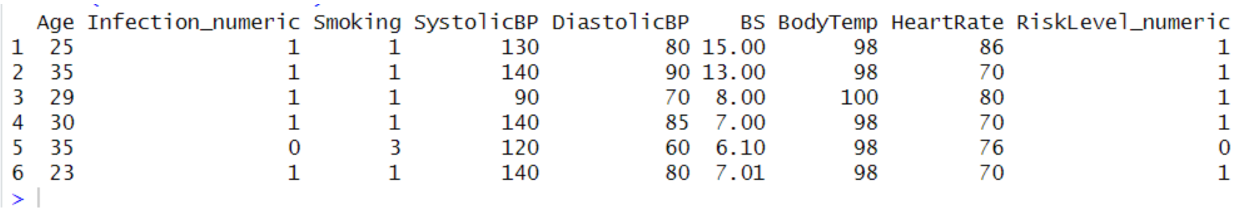
Here, library(ggplot2) helps to fill different colors to the bars with help of fill = Statistic.

19.

newDatasetNumeric$Smoking <- floor(newDatasetNumeric$Smoking)

head(newDatasetNumeric)

Output:



```
  Age Infection_numeric Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel_numeric
1  25                 1       1        130          80 15.00       98        86                 1
2  35                 1       1        140          90 13.00       98        70                 1
3  29                 1       1         90          70  8.00      100        80                 1
4  30                 1       1        140          85  7.00       98        70                 1
5  35                 0       3        120          60  6.10       98        76                 0
6  23                 1       1        140          80  7.01       98        70                 1
> 
```

Here, some values were in float after handing the missing values. So, the floats are converted to integer using floor in Smoking column.

20. Graph Plot

newDatasetNumeric$Smoking <- factor(newDatasetNumeric$Smoking, levels = c(1, 2, 3), labels = c("yes", "sometimes", "no"))

ggplot(newDatasetNumeric, aes(x = Smoking)) +

  geom_bar(width = 0.5, fill = "cyan") + # Adjust width of bars and fill color

  stat_count(aes(y = ..count.., label = ..count..), geom = "text", vjust = -0.5) + # Add count labels above bars

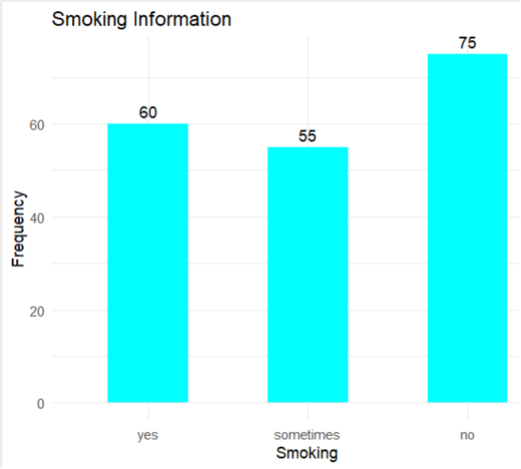
  labs(title = "Smoking Information",

       x = "Smoking",

       y = "Frequency") +

theme_minimal()

Output:



Here the Smoking column is plotted. From the dataset, we can see that there are 60 chain smokers, 55 people smoke sometimes and the number of non-smokers is only 75. Using the data set we can gather a lot of information according to our needs.

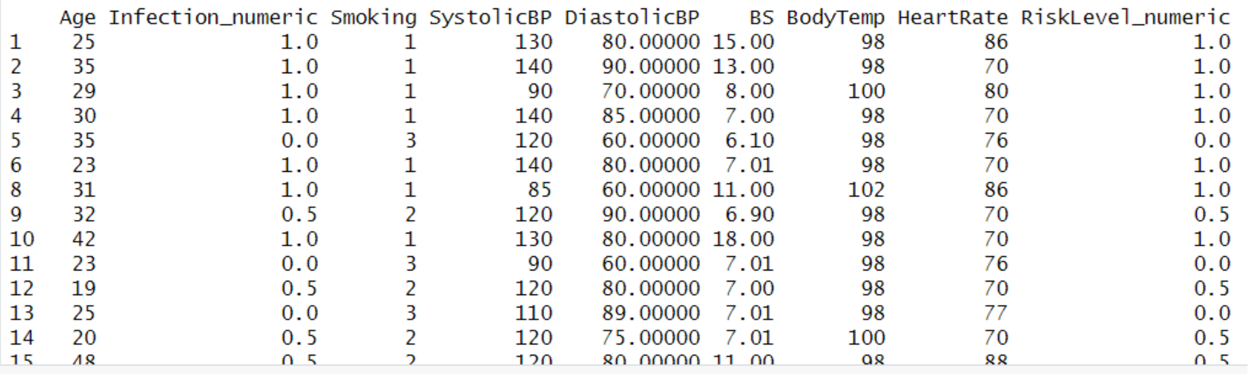
21. Data Modify

newDatasetNumeric <- newDatasetNumeric[newDatasetNumeric$BodyTemp >= 0,]

newDatasetNumeric$Age <- floor(newDatasetNumeric$Age)

newDatasetNumeric

Output:



```
     Age Infection_numeric Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel_numeric
1     25               1.0       1        130    80.00000 15.00       98        86               1.0
2     35               1.0       1        140    90.00000 13.00       98        70               1.0
3     29               1.0       1         90    70.00000  8.00      100        80               1.0
4     30               1.0       1        140    85.00000  7.00       98        70               1.0
5     35               0.0       3        120    60.00000  6.10       98        76               0.0
6     23               1.0       1        140    80.00000  7.01       98        70               1.0
8     31               1.0       1         85    60.00000 11.00      102        86               1.0
9     32               0.5       2        120    90.00000  6.90       98        70               0.5
10    42               1.0       1        130    80.00000 18.00       98        70               1.0
11    23               0.0       3         90    60.00000  7.01       98        76               0.0
12    19               0.5       2        120    80.00000  7.00       98        70               0.5
13    25               0.0       3        110    89.00000  7.01       98        77               0.0
14    20               0.5       2        120    75.00000  7.01      100        70               0.5
15    48               0.5       2        120    80.00000 11.00       98        88               0.5
```

Here, negative data from BodyTemp column is handled and Float numbers from Age column is converted to integer.

22. Extract Information from BodyTemp column

bodyTemp <- newDatasetNumeric$BodyTemp

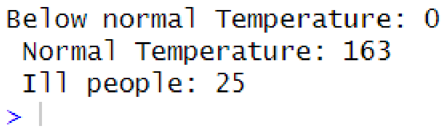
below_normal_count <- 0

normal_range_count <- 0

illness_count <- 0

```
for (temp in bodyTemp) {
 if (temp < 97) {
   below_normal_count <- below_normal_count + 1
 } else if (temp >= 97 & temp <= 99) {
   normal_range_count <- normal_range_count + 1
 } else {
   illness_count <- illness_count + 1
 }
}
cat("Below normal Temperature:", below_normal_count, "\n",
   "Normal Temperature:", normal_range_count, "\n",
   "Ill people:", illness_count, "\n")
```

Output:



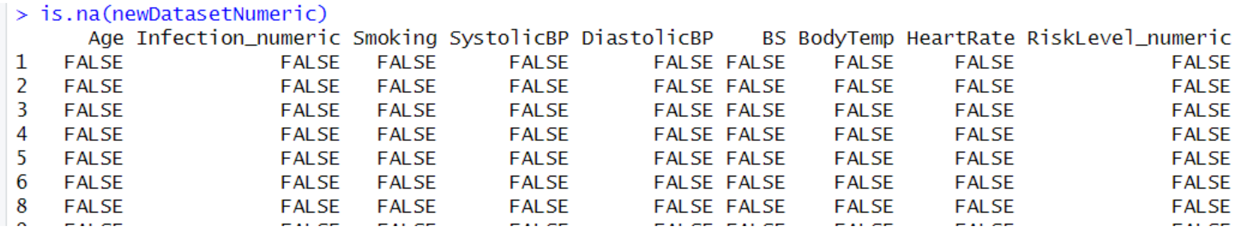
```
Below normal Temperature: 0
 Normal Temperature: 163
 Ill people: 25
> 
```

Here, we gave ranges for different body temperature in if else statements and count the number of people.

23. Missing data

is.na(newDatasetNumeric)

Output:

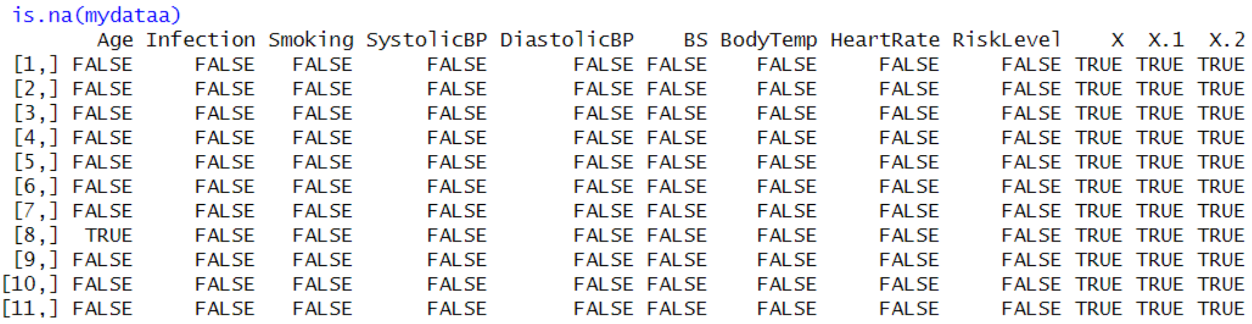


```
> is.na(newDatasetNumeric)
    Age Infection_numeric Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel_numeric
1 FALSE             FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE             FALSE
2 FALSE             FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE             FALSE
3 FALSE             FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE             FALSE
4 FALSE             FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE             FALSE
5 FALSE             FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE             FALSE
6 FALSE             FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE             FALSE
8 FALSE             FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE             FALSE
```

We can see that, there is no missing data in the dataset as returning FALSE.

is.na(mydataa)

Output:



```
is.na(mydataa)
        Age Infection Smoking SystolicBP DiastolicBP    BS BodyTemp HeartRate RiskLevel    X  X.1  X.2
 [1,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [2,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [3,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [4,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [5,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [6,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [7,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [8,]  TRUE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
 [9,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
[10,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
[11,] FALSE     FALSE   FALSE      FALSE       FALSE FALSE    FALSE     FALSE     FALSE TRUE TRUE TRUE
```

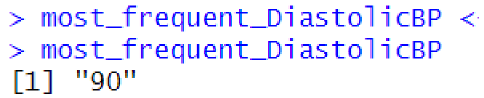
In the provided data set, we can see many TRUE which belongs to the missing data.

24.

most_frequent_DiastolicBP <- names(sort(table(newDatasetNumeric$DiastolicBP), decreasing = TRUE)[1])

most_frequent_DiastolicBP

Output:



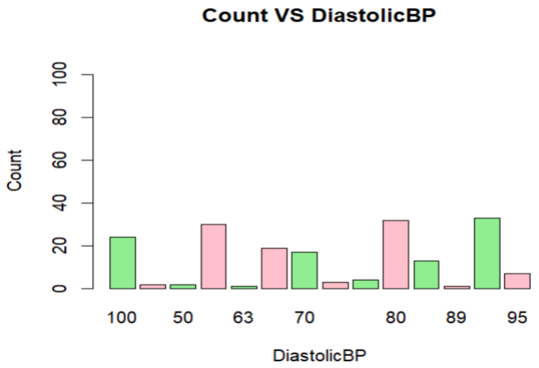
```
> most_frequent_DiastolicBP <
> most_frequent_DiastolicBP
[1] "90"
```

From the dataset, in the DiastolicBP column we can see that the most frequence BP is 90.

25.

barplot(table(newDatasetNumeric$DiastolicBP), main = "Count VS DiastolicBP", xlab = "DiastolicBP", ylab = "Count",ylim=c(0,110), col = c("lightgreen", "pink"))

Output:



Here, x axis indicates the DiastolicBP and y axis indicates the number of people. For example, we can see that around 25 people's Diastolic BP is 100.

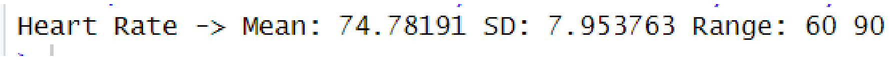
26.

heartRate_mean <- mean(newDatasetNumeric$HeartRate, na.rm = TRUE)

heartRate_sd <- sd(newDatasetNumeric$HeartRate, na.rm = TRUE) heartRate_range <- range(newDatasetNumeric$HeartRate, na.rm = TRUE)

cat("Heart Rate -> Mean:", heartRate_mean, "SD:", heartRate_sd, "Range:", heartRate_range, "\n")

Output:



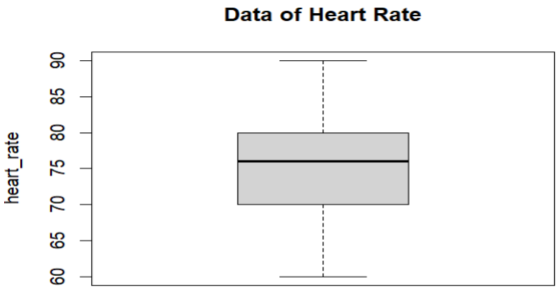
```
Heart Rate -> Mean: 74.78191 SD: 7.953763 Range: 60 90
```

Graph:

hist(newDatasetNumeric$HeartRate,main=" Data of Heart Rate", xlab="heart_rate", xlim = c(0,200),ylim=c(60,90), breaks=10)

boxplot(newDatasetNumeric$HeartRate, main = "Data of Heart Rate", ylab = "heart_rate")

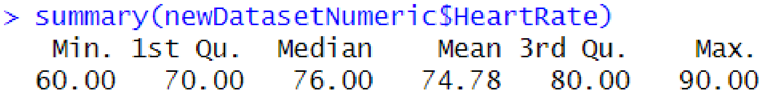
Output:



Summary of HeartRate:

summary(newDatasetNumeric$HeartRate)

Output:

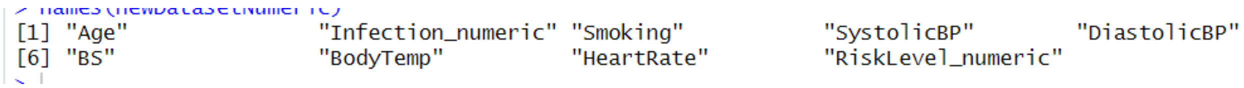


```
> summary(newDatasetNumeric$HeartRate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  60.00   70.00   76.00   74.78   80.00   90.00 
```

27.

names(newDatasetNumeric)

Output:



```
> names(newDatasetNumeric)
[1] "Age"               "Infection_numeric" "Smoking"           "SystolicBP"        "DiastolicBP"      
[6] "BS"                "BodyTemp"          "HeartRate"         "RiskLevel_numeric"
```

Shows column names from the dataset.

**********************THE END************************