AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Assignment Cover Sheet

Assignment Title:	**Data Preparation Steps for Heart Failure Clinical Records Dataset**		
Assignment No:	01	Date of Submission:	18 March 2024
Course Title:	Introduction to Data Science		
Course Code:	CSC4180	Section:	B
Semester:	Spring 2023-24	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 formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. 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 them arterial 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.: 03

No	Name	ID	Program	Signature
1	**Noshin Farzana**	**21-44647-1**	**BSc [CSE]**	
2	**Avijit Saha Anto**	**21-44630-1**	**BSc [CSE]**	

Faculty use only		
FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

TABLE OF CONTENT

ABOUT THE DATASET

The medical history of 199 patients who had heart failure was gathered over the follow-up period and is included in this dataset. Each patient profile includes 13 clinical variables. These are- age(age of the patient in years), anaemia (decrease of red blood cells or hemoglobin), creatinine_phosphokinase (level of the CPK enzyme in the blood in mcg/L), diabetes (if the patient has diabetes), ejection_fraction(percentage of blood leaving the heart at each contraction), high_blood_pressure (if the patient has hypertension), platelets (platelets in blood in kiloplatelets/mL), serum_creatinine (level of serum creatinine in blood in mg/dL), serum_sodium (level of serum sodium in blood in mEq/L), sex (woman or man), smoking (if the patient smokes or not), time (follow-up period in days) and DEATH_EVENT (if the patient died during the follow-up period). There are some missing data in age, anaemia, creatinine_phosphokinase and diabetes column. Also, creatinine_phosphokinase and sex column contain some invalid values. Using this dataset, Machine Learning Model can be developed to predict heart failure outcomes based on the patient's medical history. Moreover, this can be helpful for medical professionals for identifying patients who may be at high risk of heart failure.

IMPORTING THE DATASET

Code:

```
dataset <- read.csv("E:/Data Science Midterm Project/Dataset_midterm_Section(B).csv",na.strings=c((“”)),header= TRUE, sep = ",")

dataset
```

Output:

Console Terminal Background Jobs

R 4.3.2 · ~/

```
> dataset <- read.csv("E:/Data Science Midterm Project/Dataset_midterm_Section(B).csv",na.strings=c((“”)),header= TRUE, sep = ",
> dataset
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
1	75	0	582	0	20	1	265000	1.90	130	Male	0	4	1
2	55	0	7861	0	38	0	263358	1.10	136	Male	0	6	1
3	65	0	146	0	20	0	162000	1.30	129	Male	1	7	1
4	50	1	111	0	20	0	210000	1.90	137	Male	0	7	1
5	65	1	160	1	20	0	327000	2.70	116	Female	0	8	1
6	90	1	47	0	40	1	204000	2.10	132	Male	1	8	1
7	75	1	246	NA	15	0	127000	1.20	137	Male	0	10	1
8	60	NA	315	1	60	0	454000	1.10	131	Male	1	10	1
9	65	0	157	0	65	0	263358	1.50	138	Female	0	10	1
10	NA	1	123	0	35	1	388000	9.40	133	Maleee	1	10	1
11	75	1	81	0	38	1	368000	4.00	131	Male	1	10	1
12	62	0	231	0	25	1	NA	0.90	140	Male	NA	10	1
13	45	1	981	0	30	0	136000	1.10	137	Male	0	11	1
14	50	1	168	0	38	1	276000	1.10	137	Male	0	11	1
15	49	1	80	0	30	1	427000	1.00	138	Female	0	12	0
16	82	1	379	0	50	0	47000	1.30	136	Male	0	13	1
17	87	1	<NA>	0	38	0	262000	0.90	140	Male	0	14	1
18	45	0	582	0	14	0	166000	0.80	127	Male	NA	14	1
19	70	1	125	0	25	1	237000	1.00	140	Female	0	15	1
20	48	1	582	1	55	0	87000	1.90	121	Female	0	15	1
21	65	1	52	0	25	1	276000	1.30	137	Female	0	16	0
22	65	NA	128	1	30	1	297000	1.60	136	Female	0	20	1
23	68	1	220	0	35	1	289000	0.90	140	Male	1	20	1
24	53	0	63	1	60	0	368000	0.80	135	Male	0	22	0
25	75	0	582	1	30	1	263358	1.83	134	Female	0	23	1
26	80	0	148	1	38	0	149000	1.90	144	Male	1	23	1
27	95	1	112	0	40	1	NA	1.00	138	Female	0	24	1
28	70	0	122	1	45	1	284000	1.30	136	Male	1	26	1
29	NA	1	60	0	38	0	153000	5.80	134	Male	0	26	1
30	82	0	70	1	30	0	200000	1.20	132	Male	1	26	1
31	94	0	582	1	38	1	263358	1.83	134	Male	0	27	1
32	85	0	23	0	45	0	360000	3.00	132	Male	0	28	1
33	50	1	249	1	35	1	319000	1.00	128	Femmale	0	28	1
34	50	1	159	1	30	0	302000	1.20	138	Female	0	29	0
35	65	0	94	1	50	1	188000	1.00	140	Male	0	29	1
36	NA	NA	582	1	35	0	228000	3.50	134	Male	0	30	1
37	90	1	60	NA	50	0	226000	1.00	134	Male	0	30	1
38	82	1	855	1	50	1	321000	1.00	145	Female	0	30	1
39	60	0	2656	1	30	0	305000	2.30	137	Male	0	30	0
40	60	NA	235	1	38	0	329000	3.00	142	Female	0	30	1
41	70	0	582	0	20	1	263358	1.83	134	Male	1	31	1
42	50	0	124	1	30	1	153000	1.20	136	Female	1	32	1
43	70	0	571	1	45	1	185000	1.20	139	Male	1	33	1
44	72	0	127	1	50	1	218000	1.00	134	Male	NA	33	0
45	60	1	588	1	60	0	NA	1.10	142	Female	0	33	1
46	50	0	582	1	38	0	310000	1.90	135	Male	1	35	1
47	51	0	1380	0	25	1	271000	0.90	130	Male	0	38	1
48	60	NA	582	1	38	1	451000	0.60	138	Male	1	40	1
49	80	1	553	0	20	1	140000	4.40	133	Male	0	41	1
50	NA	1	129	0	30	0	395000	1.00	140	Female	0	42	1
51	68	1	<NA>	0	25	1	166000	1.00	138	Male	0	43	1
52	53	1	91	0	20	1	418000	1.40	139	Female	0	43	1
53	60	0	3964	1	62	0	263358	6.80	146	Female	0	43	1
54	70	1	69	1	50	1	351000	1.00	134	Female	0	44	1
55	60	1	260	1	38	0	255000	2.20	132	Female	1	45	1
56	95	1	371	0	30	0	461000	2.00	132	Male	0	50	1
57	70	1	75	0	35	0	223000	2.70	138	Male	1	54	0
58	60	1	607	0	40	0	216000	0.60	138	Male	1	54	0
59	49	0	789	0	20	1	319000	1.10	136	Male	1	55	1
60	72	0	364	1	20	1	254000	1.30	136	Male	1	59	1
61	45	0	7702	1	25	1	390000	1.00	139	Male	0	60	1
62	50	0	318	0	40	1	216000	2.30	131	Female	0	60	1
63	55	0	109	0	35	0	254000	1.10	139	Male	1	60	0
64	45	0	<NA>	0	35	0	385000	1.00	145	Male	0	61	1
65	45	0	582	0	80	0	263358	1.18	137	Female	0	63	0
66	60	0	68	0	20	0	119000	2.90	127	Male	1	64	1
67	NA	1	250	1	15	0	213000	1.30	136	Female	0	65	1
68	72	1	110	0	25	0	274000	1.00	140	Male	1	65	1
69	70	0	161	0	25	0	244000	1.20	142	Female	0	66	1
70	65	0	113	1	25	0	497000	1.83	135	Male	0	67	1
71	41	0	148	0	40	0	374000	0.80	140	Male	1	68	0
72	58	0	582	1	35	0	122000	0.90	139	Male	1	71	0
73	85	0	5882	0	35	0	243000	1.00	132	Male	1	72	1
74	65	0	224	1	50	0	149000	1.30	137	Male	1	72	0
75	69	0	582	0	20	0	NA	1.20	134	Male	1	73	1
76	60	1	47	0	20	0	204000	0.70	139	Male	1	73	1

Initially, the XLSX to CSV format conversion of the given dataset was performed. Next, Rstudio was used to import the dataset by using the read.csv. There are 4 parameters in read.csv method. These are- location of csv file, na.strings = c("") which replaces empty strings with NAs, header to declare the first row as the columns and sep which separates value using comma.

DATA EXPLORATION

Show Attributes Name

Code:

names (dataset)

Output:

```
> names (dataset)
 [1] "age"                "anaemia"            "creatinine_phosphokinase" "diabetes"
 [5] "ejection_fraction"  "high_blood_pressure" "platelets"                "serum_creatinine"
 [9] "serum_sodium"       "sex"                "smoking"                  "time"
[13] "DEATH_EVENT"
>
```

The above command is used to show the attributes name of given dataset.

Show Data Types for each Attribute

Code:

str(dataset)

Output:

```
> str(dataset)
'data.frame':   199 obs. of  13 variables:
 $ age                     : num  75 55 65 50 65 90 75 60 65 NA ...
 $ anaemia                 : int  0 0 0 1 1 1 1 NA 0 1 ...
 $ creatinine_phosphokinase: chr  "582" "7861" "146" "111" ...
 $ diabetes                : int  0 0 0 0 1 0 NA 1 0 0 ...
 $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
 $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
 $ platelets               : num  265000 263358 162000 210000 327000 ...
 $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
 $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
 $ sex                     : chr  "Male" "Male" "Male" "Male" ...
 $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
 $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
 $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
>
```

The above command is used to show the data type for each attributes of given dataset.

Show Summary of Dataset

Code:

summary(dataset)

Output:

```
> summary(dataset)
      age            anaemia       creatinine_phosphokinase    diabetes      ejection_fraction high_blood_pressure
 Min.   : 40.00   Min.   :0.0000   Length:199                 Min.   :0.0000   Min.   :14.00     Min.   :0.000
 1st Qu.: 52.00   1st Qu.:0.0000   Class :character           1st Qu.:0.0000   1st Qu.:30.00     1st Qu.:0.000
 Median : 60.00   Median :0.0000   Mode  :character           Median :0.0000   Median :38.00     Median :0.000
 Mean   : 63.13   Mean   :0.4794                              Mean   :0.4051   Mean   :37.97     Mean   :0.397
 3rd Qu.: 70.00   3rd Qu.:1.0000                              3rd Qu.:1.0000   3rd Qu.:45.00     3rd Qu.:1.000
 Max.   :170.00   Max.   :1.0000                              Max.   :1.0000   Max.   :80.00     Max.   :1.000
 NA's   :5        NA's   :5                                   NA's   :4
   platelets      serum_creatinine  serum_sodium       sex                smoking            time         DEATH_EVENT
 Min.   : 47000   Min.   :0.600    Min.   :116.0   Length:199         Min.   :0.0000   Min.   :  4.00   Min.   :0.0000
 1st Qu.:216000   1st Qu.:0.900    1st Qu.:134.0   Class :character   1st Qu.:0.0000   1st Qu.: 42.50   1st Qu.:0.0000
 Median :263000   Median :1.100    Median :137.0   Mode  :character   Median :0.0000   Median : 86.00   Median :0.0000
 Mean   :262560   Mean   :1.441    Mean   :136.6                      Mean   :0.3316   Mean   : 84.03   Mean   :0.4472
 3rd Qu.:304000   3rd Qu.:1.550    3rd Qu.:139.0                      3rd Qu.:1.0000   3rd Qu.:114.00   3rd Qu.:1.0000
 Max.   :850000   Max.   :9.400    Max.   :146.0                      Max.   :1.0000   Max.   :186.00   Max.   :1.0000
 NA's   :6                                                           NA's   :3
>
```

The above command is used to show the mean, median and max for each attributes of given dataset.

Show Numeric Attributes

Code:

dataset %>% summarise_if(is.numeric,sd)

Output:

```
> dataset %>% summarise_if(is.numeric,sd)
  age anaemia diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium smoking     time
1  NA      NA       NA          12.67859           0.4905068        NA         1.050314     4.294505      NA 48.05394
  DEATH_EVENT
1   0.4984622
>
```

The above command is used to show numeric attributes of given dataset.

Standard Deviation

❖ Standard Deviation for age

Code:

```
ageStandardDeviation = sd(dataset$age, na.rm = TRUE)
cat("Standard Deviation of age: ", ageStandardDeviation)
```

Output:

```
> ageStandardDeviation = sd(dataset$age, na.rm = TRUE)
> cat("Standard Deviation of age: ", ageStandardDeviation)
Standard Deviation of age:  16.05917
>
```

Here, sd is used to calculate standard deviation. The age attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in ageStandardDeviation variable, then the result is printed using cat function.

❖ Standard Deviation for creatinine_phosphokinase

Code:

```
creatinine_phosphokinaseStandardDeviation = sd(dataset$creatinine_phosphokinase, na.rm =
TRUE)
cat("Standard Deviation of creatinine_phosphokinase: ",
creatinine_phosphokinaseStandardDeviation)
```

Output:

```
> creatinine_phosphokinaseStandardDeviation = sd(dataset$creatinine_phosphokinase, na.rm = TRUE)
> cat("Standard Deviation of creatinine_phosphokinase: ", creatinine_phosphokinaseStandardDeviation)
Standard Deviation of creatinine_phosphokinase:  1113.677
>
```

Here, sd is used to calculate standard deviation. The creatinine_phosphokinase attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in creatinine_phosphokinaseStandardDeviation variable, then the result is printed using cat function.

❖ **Standard Deviation for ejection_fraction**

Code:

```
ejection_fractionStandardDeviation = sd(dataset$ejection_fraction, na.rm = TRUE)
cat("Standard Deviation of ejection_fraction: ", ejection_fractionStandardDeviation)
```

Output:

```
> ejection_fractionStandardDeviation = sd(dataset$ejection_fraction, na.rm = TRUE)
> cat("Standard Deviation of ejection_fraction: ", ejection_fractionStandardDeviation)
Standard Deviation of ejection_fraction:  12.67859
>
```

Here, sd is used to calculate standard deviation. The ejection_fraction attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in ejection_fractionStandardDeviation variable, then the result is printed using cat function.

❖ **Standard Deviation for platelets**

Code:

```
plateletsStandardDeviation = sd(dataset$platelets, na.rm = TRUE)
cat("Standard Deviation of platelets: ", plateletsStandardDeviation)
```

Output:

```
> plateletsStandardDeviation = sd(dataset$platelets, na.rm = TRUE)
> cat("Standard Deviation of platelets: ", plateletsStandardDeviation)
Standard Deviation of platelets:  93694.7
>
```

Here, sd is used to calculate standard deviation. The platelets attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in platelets Deviation variable, then the result is printed using cat function.

❖ **Standard Deviation for serum_creatinine**

Code:

```
serum_creatinineStandardDeviation = sd(dataset$serum_creatinine, na.rm = TRUE)
cat("Standard Deviation of serum_creatinine: ", serum_creatinineStandardDeviation)
```

Output:

```
> serum_creatinineStandardDeviation = sd(dataset$serum_creatinine, na.rm = TRUE)
> cat("Standard Deviation of serum_creatinine: ", serum_creatinineStandardDeviation)
Standard Deviation of serum_creatinine:  1.050314
```

Here, sd is used to calculate standard deviation. The serum_creatinine attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in serum_creatinineDeviation variable, then the result is printed using cat function.

❖ **Standard Deviation for serum_sodium**

Code:

```
serum_sodiumStandardDeviation = sd(dataset$serum_sodium, na.rm = TRUE)
cat("Standard Deviation of serum_sodium: ", serum_sodiumStandardDeviation)
```

Output:

```
> serum_sodiumStandardDeviation = sd(dataset$serum_sodium, na.rm = TRUE)
> cat("Standard Deviation of serum_sodium: ", serum_sodiumStandardDeviation)
Standard Deviation of serum_sodium:  4.294505
>
```

Here, sd is used to calculate standard deviation. The serum_sodium attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in serum_sodiumDeviation variable, then the result is printed using cat function.

❖ **Standard Deviation for time**

Code:

```
timeStandardDeviation = sd(dataset$time, na.rm = TRUE)
cat("Standard Deviation of time: ", timeStandardDeviation)
```

Output:

```
> timeStandardDeviation = sd(dataset$time, na.rm = TRUE)
> cat("Standard Deviation of time: ", timeStandardDeviation)
Standard Deviation of time:  48.05394
>
```

Here, sd is used to calculate standard deviation. The timeattribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in timeStandardDeviation variable, then the result is printed using cat function.

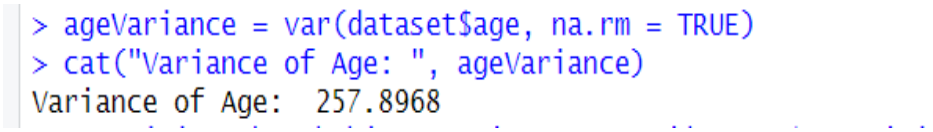
Variance

❖ Variance for age

Code:

```
ageVariance = var(dataset$age, na.rm = TRUE)
cat("Variance of Age: ", ageVariance)
```

Output:



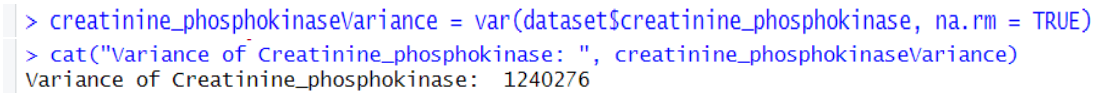
Here, var is used to calculate variance. The age attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in ageVariance variable, then the result is printed using the cat method.

❖ Variance for creatinine_phosphokinase

Code:

```
creatinine_phosphokinaseVariance = var(dataset$creatinine_phosphokinase, na.rm = TRUE)
cat("Variance of Creatinine_phosphokinase: ", creatinine_phosphokinaseVariance)
```

Output:



Here, var is used to calculate variance. The creatinine_phosphokinase attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in creatinine_phosphokinaseVariance variable, then the result is printed using the cat method.

❖ **Variance for ejection_fraction**

Code:

```
ejection_fractionVariance = var(dataset$ejection_fraction, na.rm = TRUE)
cat("Variance of Ejection_fraction: ", ejection_fractionVariance)
```

Output:

```
> ejection_fractionVariance = var(dataset$ejection_fraction, na.rm = TRUE)
> cat("Variance of Ejection_fraction: ", ejection_fractionVariance)
Variance of Ejection_fraction:  160.7466
>
```

Here, var is used to calculate variance. The ejection_fraction attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in ejection_fractionVariance variable, then the result is printed using the cat method.

❖ **Variance for platelets**

Code:

```
plateletsVariance = var(dataset$platelets, na.rm = TRUE)
cat("Variance of Platelets: ", plateletsVariance)
```

Output:

```
> plateletsVariance = var(dataset$platelets, na.rm = TRUE)
> cat("Variance of Platelets: ", plateletsVariance)
Variance of Platelets:  8778696980
>
```

Here, var is used to calculate variance. The platelets attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in plateletsVariance variable, then the result is printed using the cat method.

❖ **Variance for serum_creatinine**

Code:

```
serum_creatinineVariance = var(dataset$serum_creatinine, na.rm = TRUE)
cat("Variance of Serum_creatinine: ", serum_creatinineVariance)
```

Output:

```
> serum_creatinineVariance = var(dataset$serum_creatinine, na.rm = TRUE)
> cat("Variance of Serum_creatinine: ", serum_creatinineVariance)
Variance of Serum_creatinine:  1.103159
```

Here, var is used to calculate variance. The serum_creatinine attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in serum_creatinineVariance variable, then the result is printed using the cat method.

❖ **Variance for serum_sodium**

Code:

```
serum_sodiumVariance = var(dataset$serum_sodium, na.rm = TRUE)
cat("Variance of Serum_sodium: ", serum_sodiumVariance)
```

Output:

```
> serum_sodiumVariance = var(dataset$serum_sodium, na.rm = TRUE)
> cat("Variance of Serum_sodium: ", serum_sodiumVariance)
Variance of Serum_sodium:  18.44277
```

Here, var is used to calculate variance. The serum_sodium attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in serum_sodiumVariance variable, then the result is printed using the cat method.

❖ **Variance for time**

Code:

```
timeVariance = var(dataset$time, na.rm = TRUE)
cat("Variance of Time: ", timeVariance)
```

Output:

```
> timeVariance = var(dataset$time, na.rm = TRUE)
> cat("Variance of Time: ", timeVariance)
Variance of Time:  2309.181
>
```

Here, var is used to calculate variance. The time attribute of the dataset was passed as a parameter with na.rm = TRUE which does not consider the NA values while calculating. Finally, the result was stored in timeVariance variable, then the result is printed using the cat method.

DATA VISUALIZATION

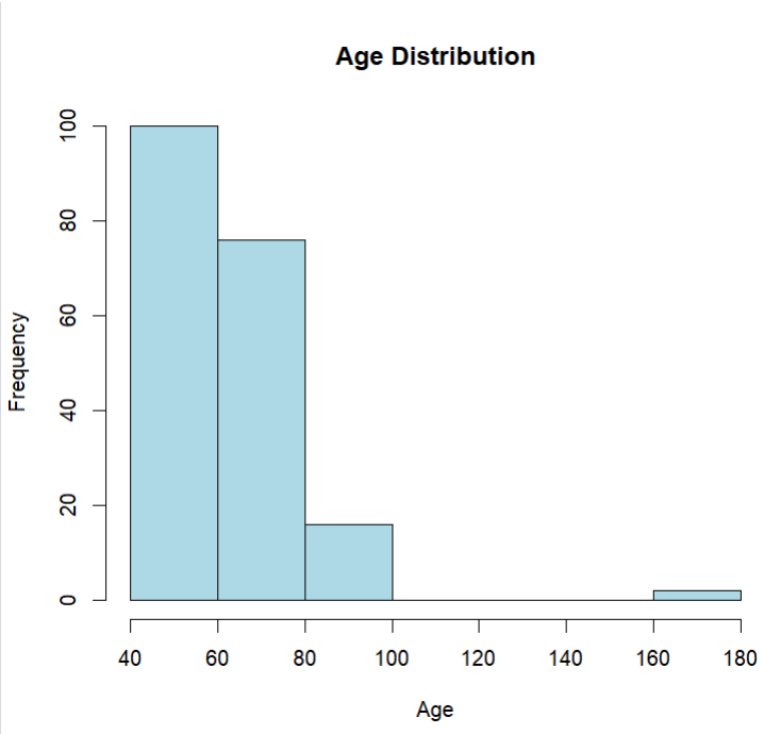
Histogram

❖ **Histogram for age**

Code:

```
hist(dataset$age, main = "Age Distribution", xlab = "Age", ylab="Frequency" ,col = "lightblue")
```

Output:



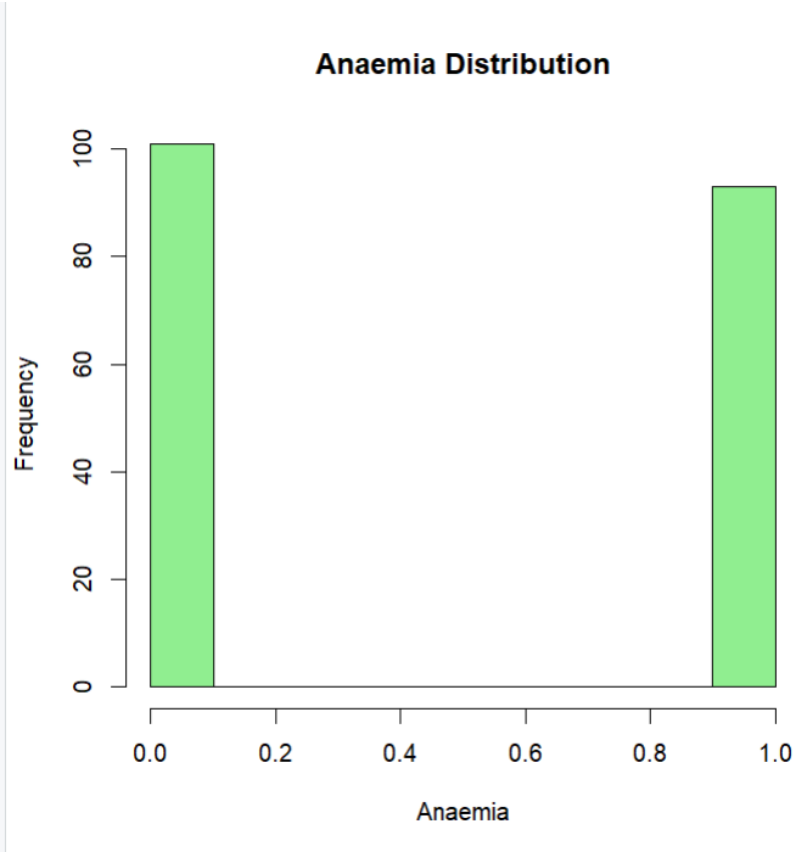
Here, hist is used in order to create histograms. dataset$age was passed as an argument to create a histogram of the age attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for anaemia**

Code:

```
hist(dataset$anaemia, main = "Anaemia Distribution", xlab = "Anaemia", ylab="Frequency" ,col = "lightgreen")
```

Output:



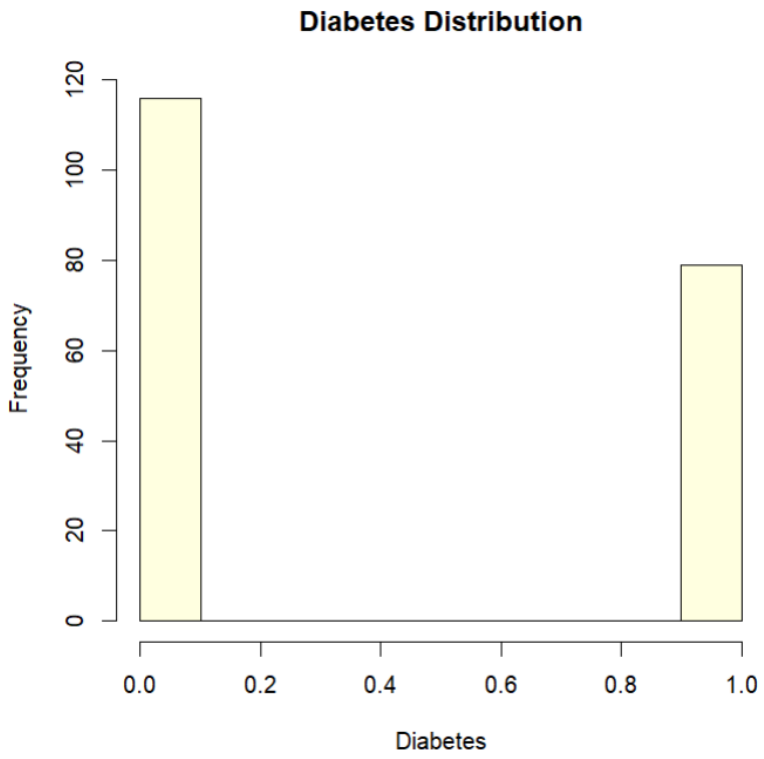
Here, hist is used in order to create histograms. dataset$anaemia was passed as an argument to create a histogram of the anaemia attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for diabetes**

Code:

```
hist(dataset$diabetes, main = "Diabetes Distribution", xlab = "Diabetes", ylab="Frequency" ,col = "lightyellow")
```

Output:



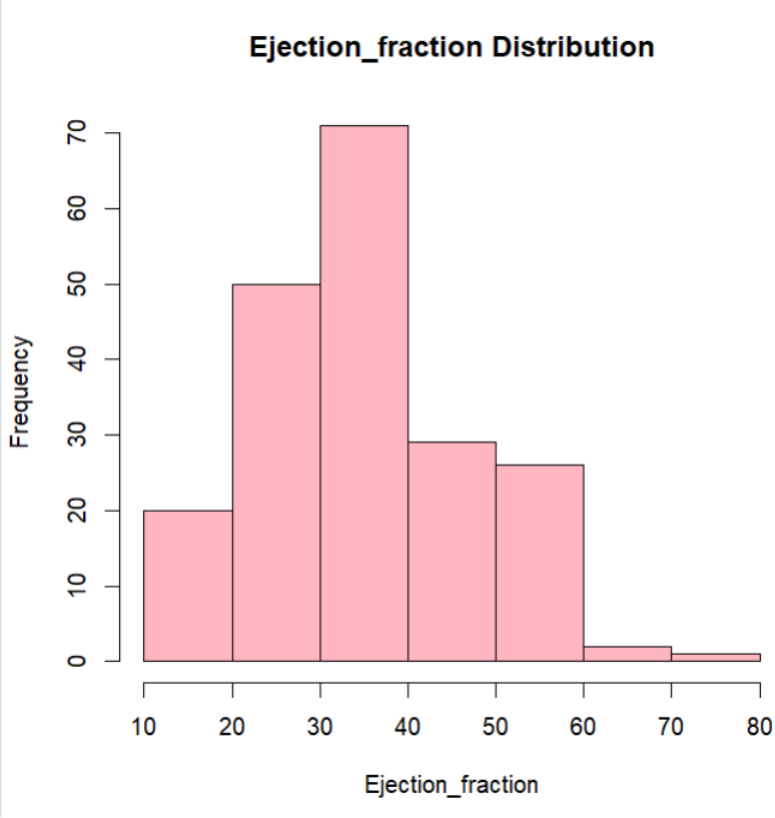
Here, hist is used in order to create histograms. dataset$diabetes was passed as an argument to create a histogram of the diabetes attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for ejection_fraction**

Code:

```
hist(dataset$ejection_fraction, main = "Ejection_fraction Distribution", xlab =
"Ejection_fraction", ylab="Frequency" ,col = "lightpink")
```

Output:



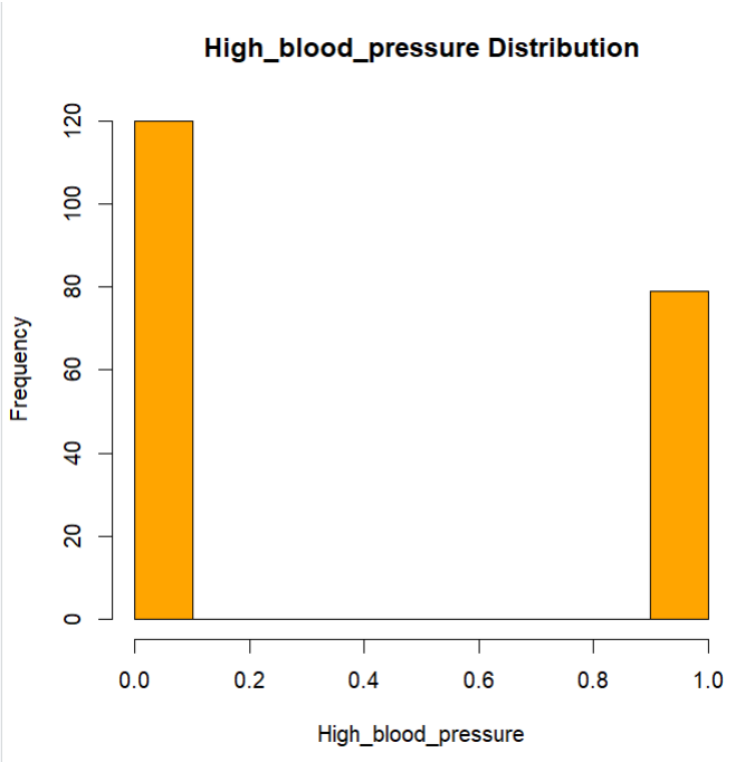
Here, hist is used in order to create histograms. dataset$ejection_fraction was passed as an argument to create a histogram of the ejection_fraction attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for high_blood_pressure**

Code:

```
hist(dataset$high_blood_pressure, main = "High_blood_pressure Distribution", xlab =
"High_blood_pressure", ylab="Frequency" ,col = "orange")
```

Output:



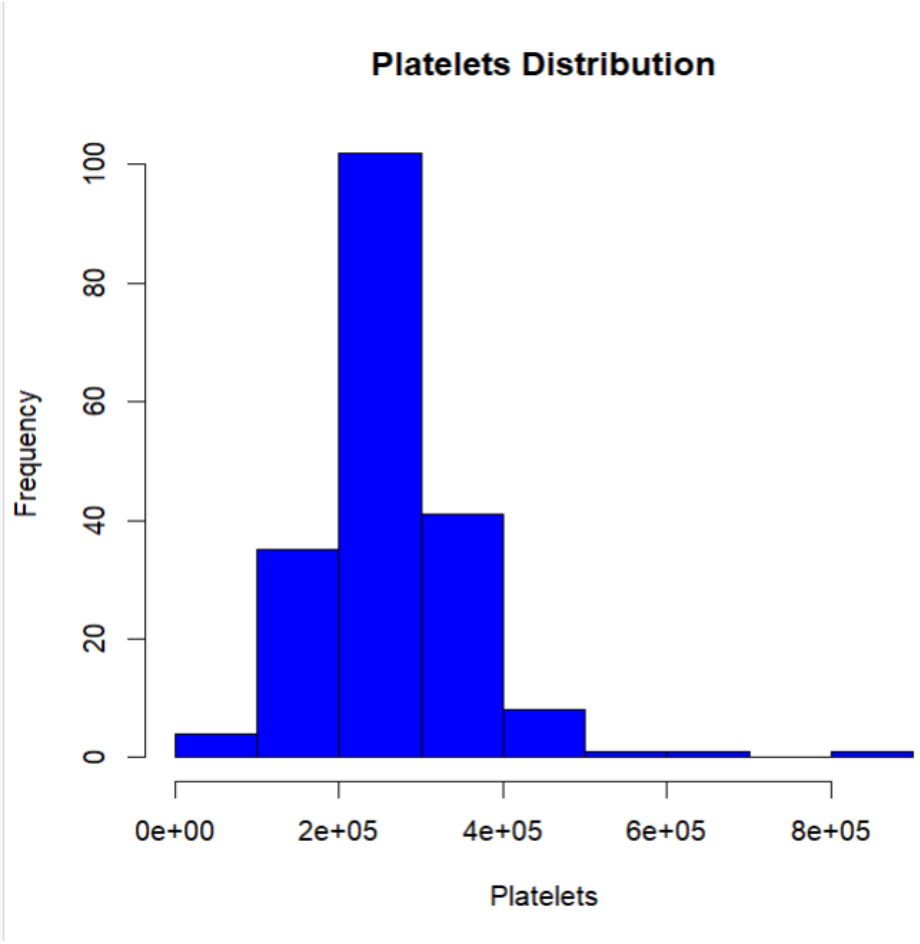
Here, hist is used in order to create histograms. dataset$high_blood_pressure was passed as an argument to create a histogram of the high_blood_pressure attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for platelets**

Code:

```
hist(dataset$platelets, main = "Platelets Distribution", xlab = "Platelets", ylab="Frequency" ,col = "blue")
```

Output:



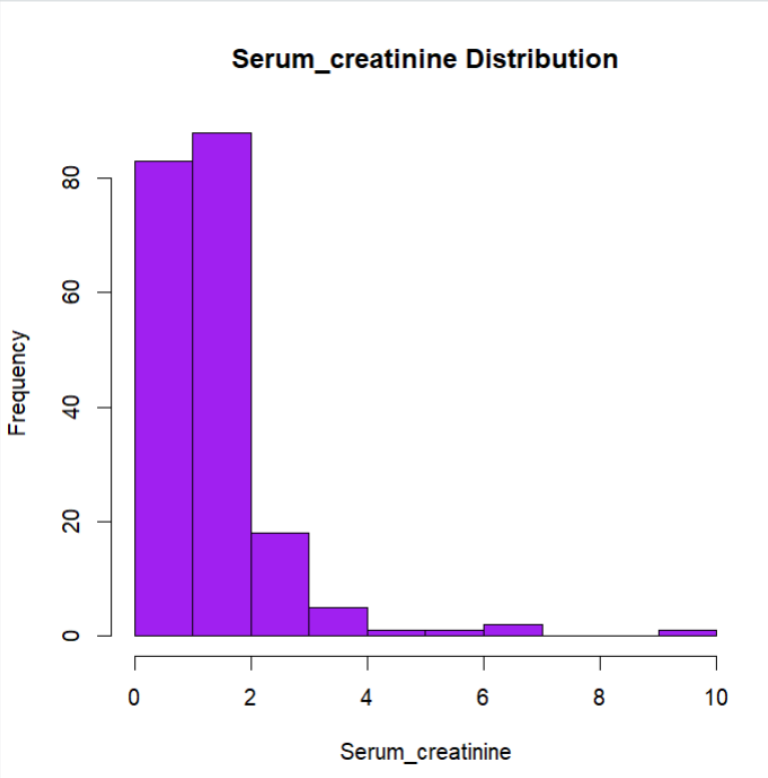
Here, hist is used in order to create histograms. dataset$platelets was passed as an argument to create a histogram of the platelets attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for serum_creatinine**

Code:

```
hist(dataset$serum_creatinine, main = "Serum_creatinine Distribution", xlab = "Serum_creatinine", ylab="Frequency" ,col = "purple")
```

Output:



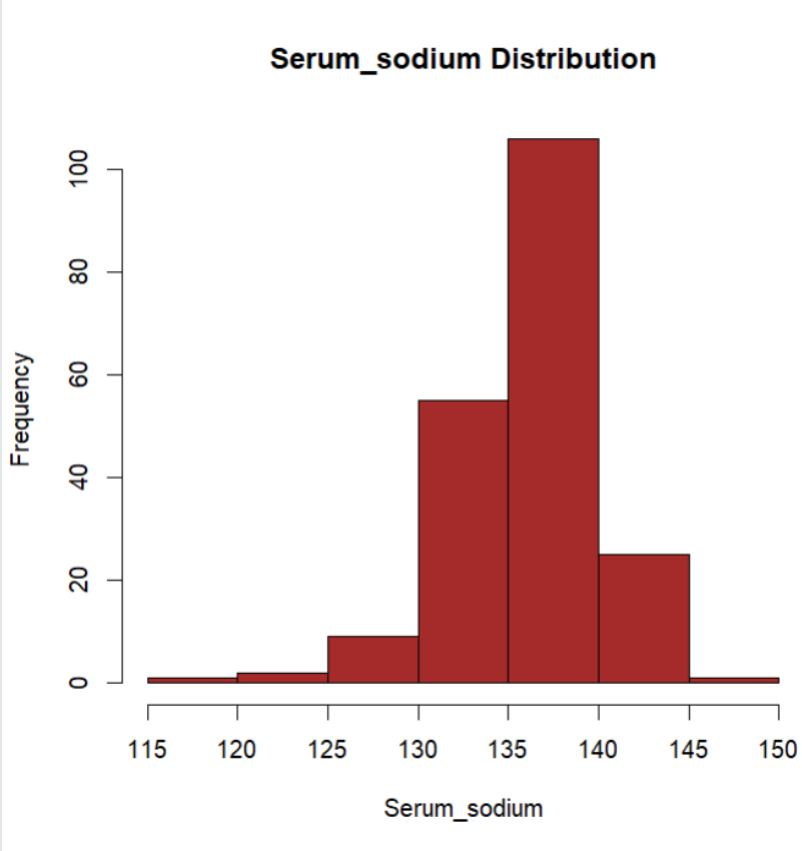
Here, hist is used in order to create histograms. dataset$serum_creatinine was passed as an argument to create a histogram of the serum_creatinine attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for serum_sodium**

Code:

```
hist(dataset$serum_sodium, main = "Serum_sodium Distribution", xlab = "Serum_sodium",
ylab="Frequency" ,col = "brown")
```

Output:



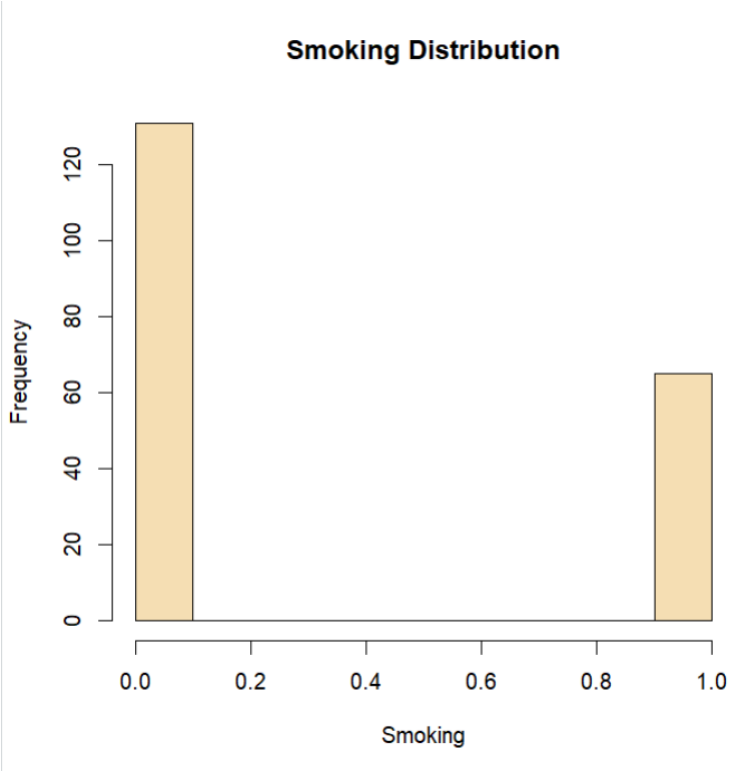
Here, hist is used in order to create histograms. dataset$serum_sodium was passed as an argument to create a histogram of the serum_sodium attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for smoking**

Code:

```
hist(dataset$smoking, main = "Smoking Distribution", xlab = "Smoking", ylab="Frequency" ,col = "wheat")
```

Output:



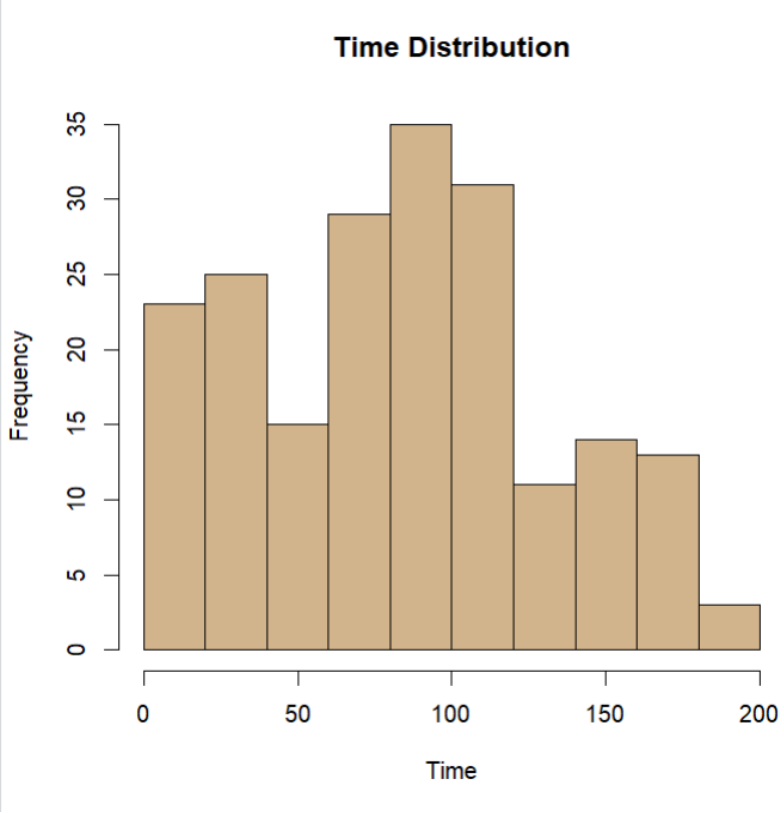
Here, hist is used in order to create histograms. dataset$smoking was passed as an argument to create a histogram of the smoking attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for time**

Code:

```
hist(dataset$time, main = "Time Distribution", xlab = "Time", ylab="Frequency" ,col = "tan")
```

Output:



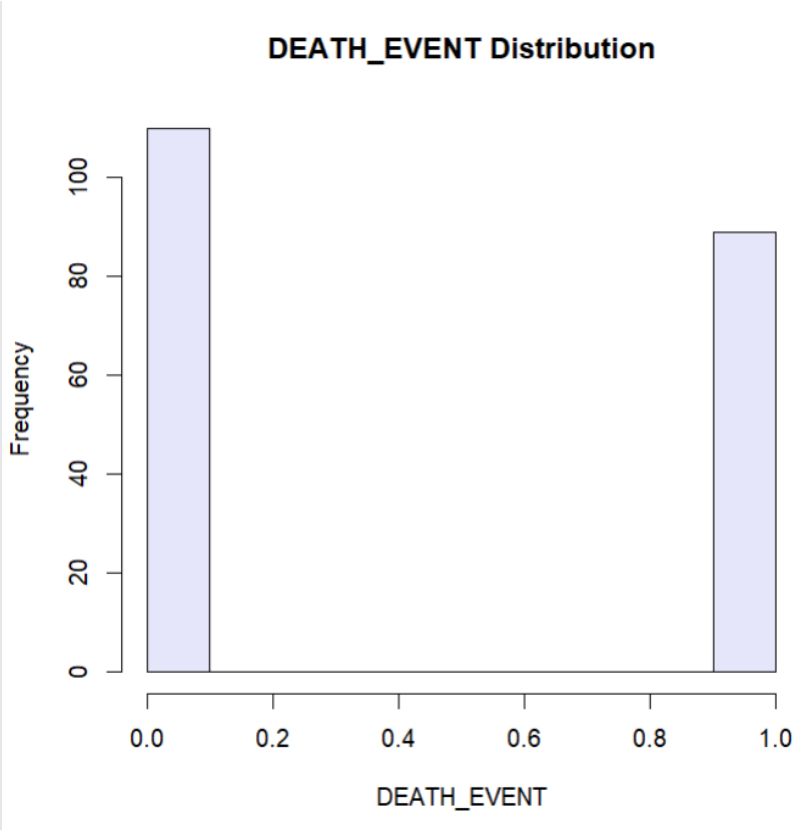
Here, hist is used in order to create histograms. dataset$time was passed as an argument to create a histogram of the time attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Histogram for DEATH_EVENT**

Code:

```
hist(dataset$DEATH_EVENT, main = "DEATH_EVENT Distribution", xlab =
"DEATH_EVENT", ylab="Frequency" ,col = "lavender")
```

Output:



Here, hist is used in order to create histograms. dataset$DEATH_EVENT was passed as an argument to create a histogram of the DEATH_EVENT attribute, main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

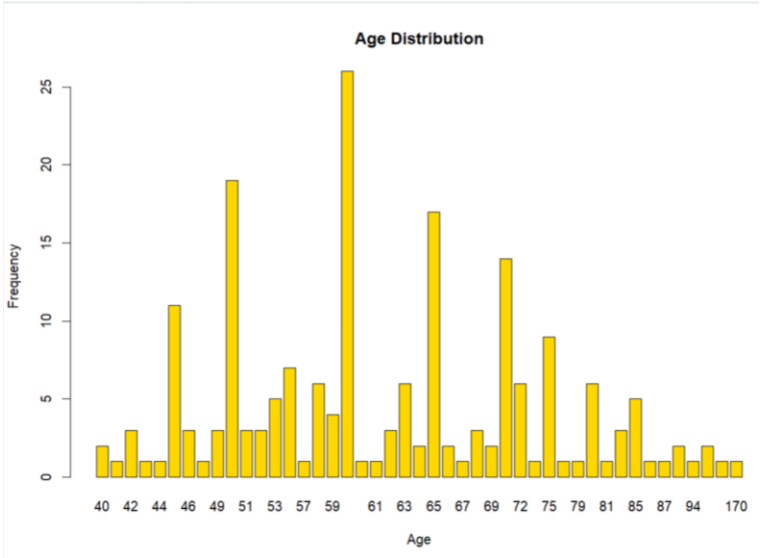
Bar Graph

❖ **Bar Graph for age**

Code:

```
barplot(table(dataset$age), main="Age Distribution", xlab="Age", ylab="Frequency",col =
"gold")
```

Output:



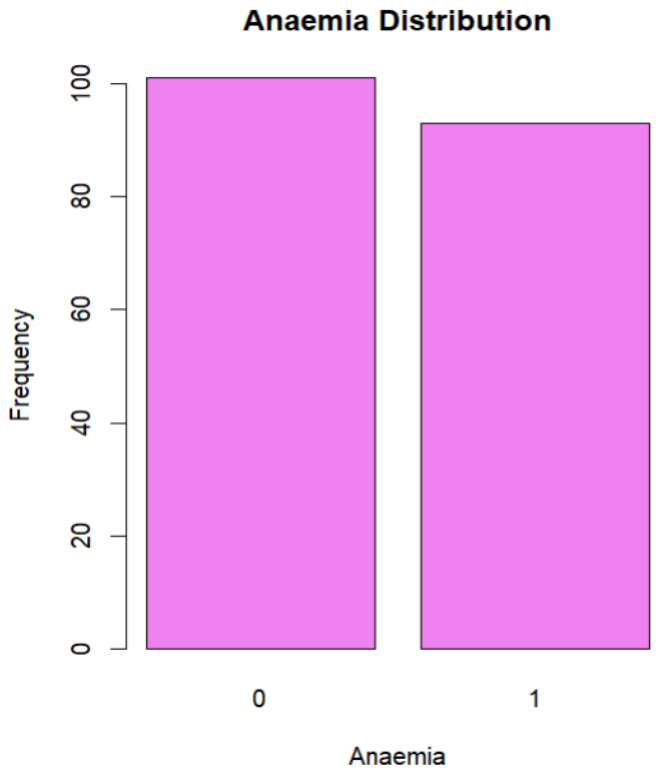
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$age to tabulate the age attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for anaemia**

Code:

```
barplot(table(dataset$anaemia), main="Anaemia Distribution", xlab="Anaemia",
ylab="Frequency",col = "violet")
```

Output:



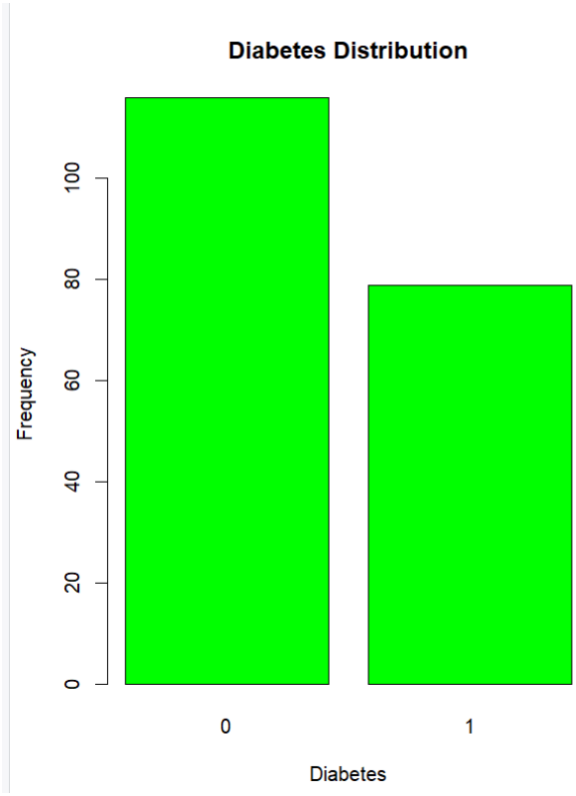
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$anaemia to tabulate the anaemia attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for diabetes**

Code:

```
barplot(table(dataset$diabetes), main="Diabetes Distribution", xlab="Diabetes",
ylab="Frequency",col = "green")
```

Output:



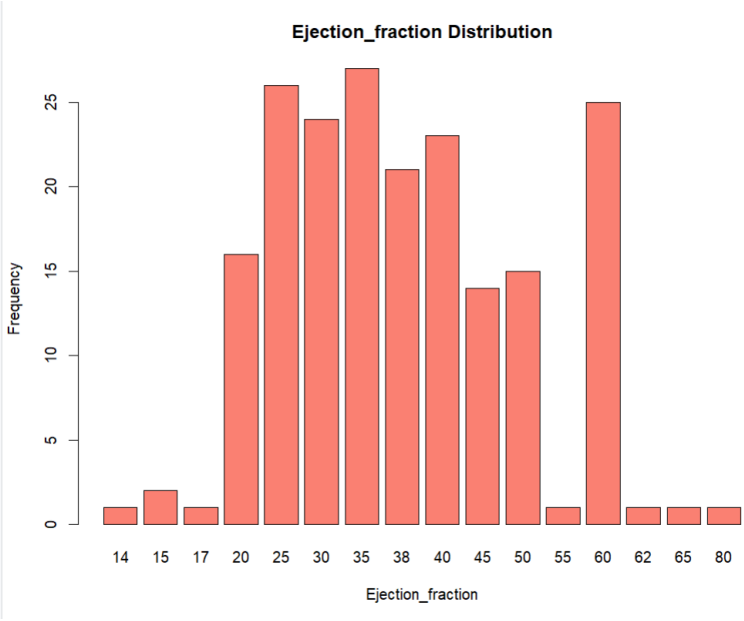
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$diabetes to tabulate the diabetes attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for ejection_fraction**

Code:

```
barplot(table(dataset$ejection_fraction), main="Ejection_fraction Distribution",
xlab="Ejection_fraction", ylab="Frequency",col = "salmon")
```

Output:



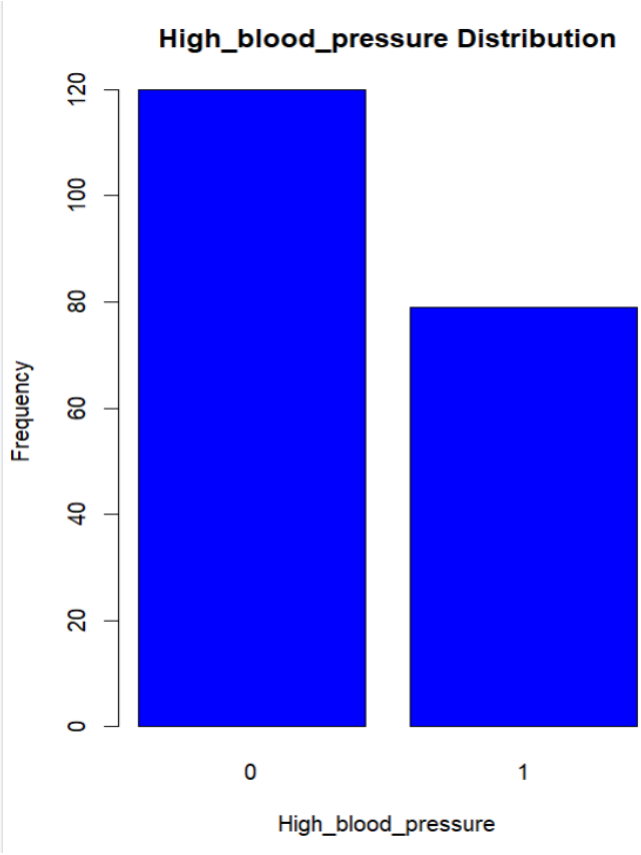
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$ejection_fraction to tabulate the ejection_fraction attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for high_blood_pressure**

Code:

```
barplot(table(dataset$high_blood_pressure), main="High_blood_pressure Distribution",
xlab="High_blood_pressure", ylab="Frequency",col = "blue")
```

Output:



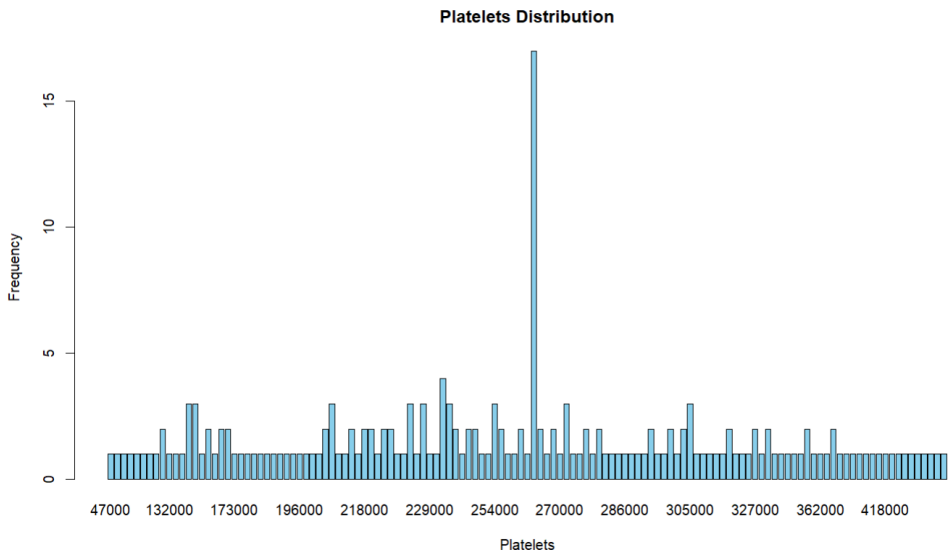
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$high_blood_pressure to tabulate the high_blood_pressure attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for platelets**

Code:

```
barplot(table(dataset$platelets), main="Platelets Distribution", xlab="Platelets",
ylab="Frequency",col = "skyblue")
```

Output:



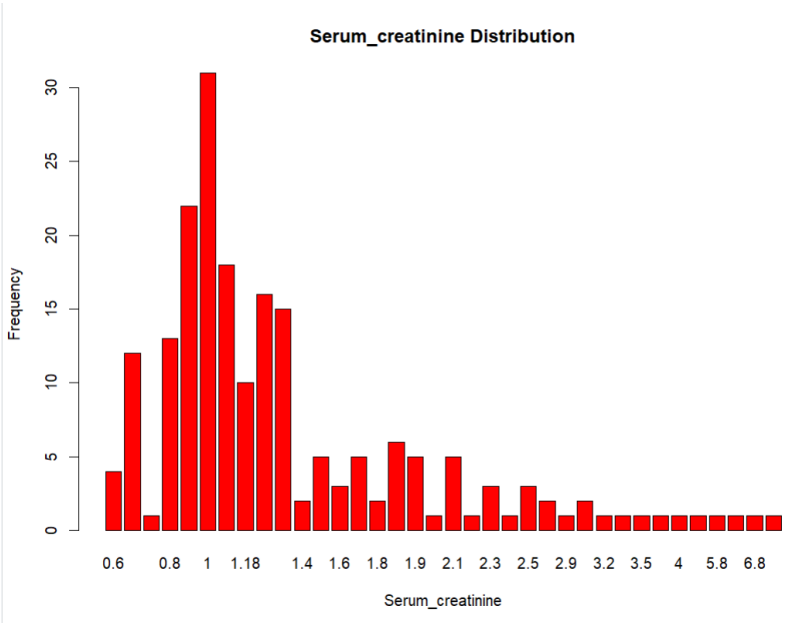
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$platelets to tabulate the platelets attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for serum_creatinine**

Code:

```
barplot(table(dataset$serum_creatinine), main="Serum_creatinine Distribution",
xlab="Serum_creatinine", ylab="Frequency",col = "red")
```

Output:



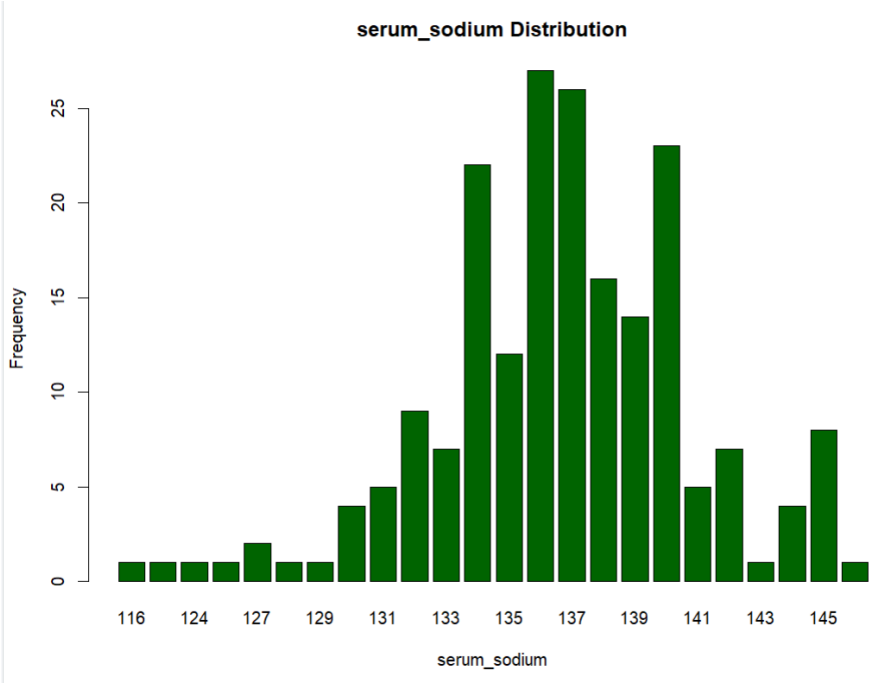
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$serum_creatinine to tabulate the serum_creatinine attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for serum_sodium**

Code:

```
barplot(table(dataset$serum_sodium), main="serum_sodium Distribution",
xlab="serum_sodium", ylab="Frequency",col = "darkgreen")
```

Output:



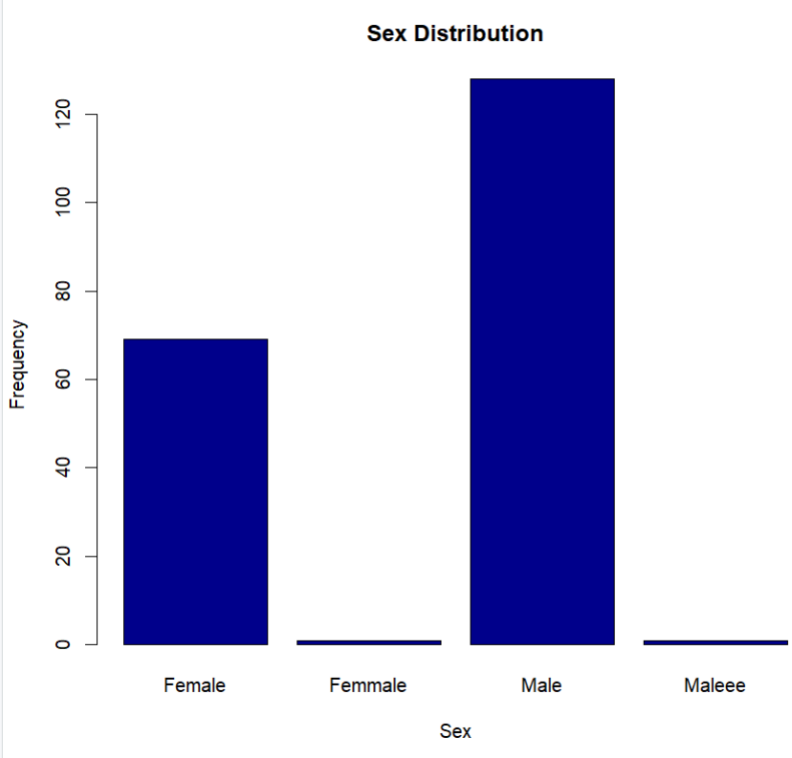
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$serum_sodium to tabulate the serum_sodium attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for sex**

Code:

```
barplot(table(dataset$sex), main="Sex Distribution", xlab="Sex", ylab="Frequency",col =
"darkblue")
```

Output:



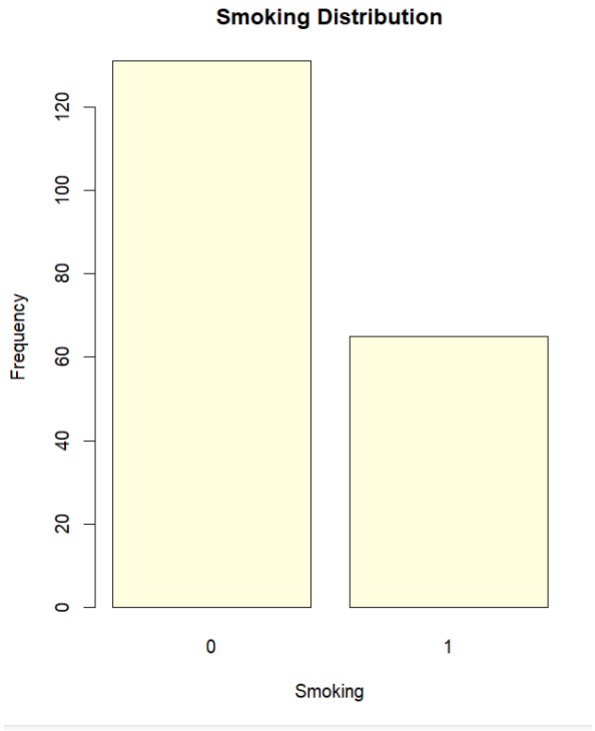
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$sex to tabulate the sex attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for smoking**

Code:

```
barplot(table(dataset$smoking), main="Smoking Distribution", xlab="Smoking",
ylab="Frequency",col = "lightyellow")
```

Output:



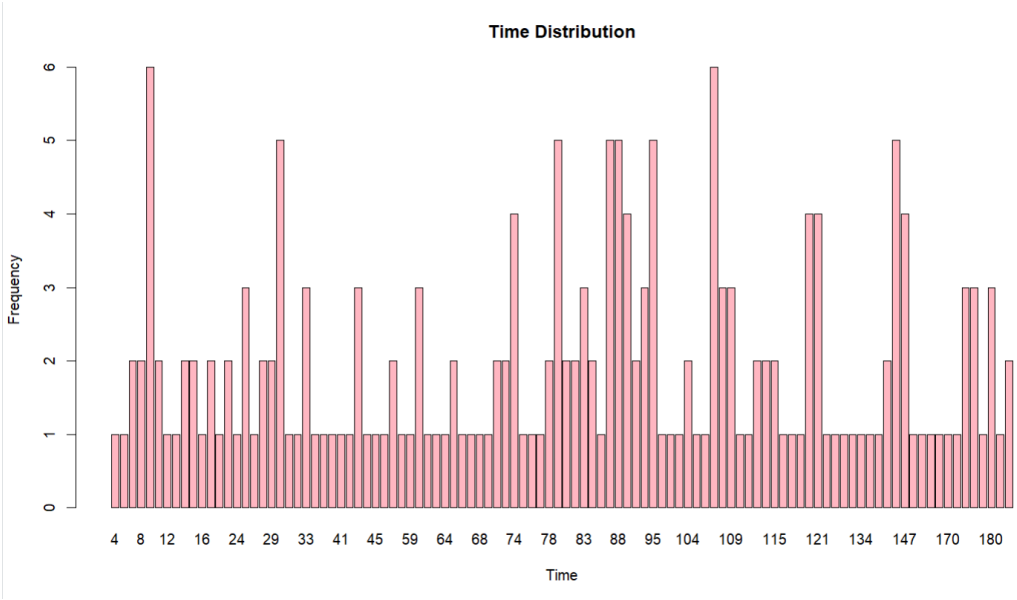
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$smoking to tabulate the smoking attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for time**

Code:

barplot(table(dataset$time), main="Time Distribution", xlab="Time", ylab="Frequency",col = "lightpink")

Output:



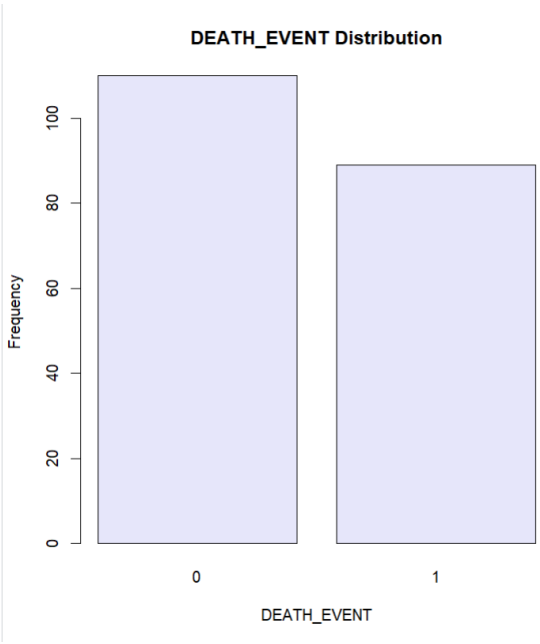
Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$time to tabulate the time attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

❖ **Bar Graph for DEATH_EVENT**

Code:

```
barplot(table(dataset$DEATH_EVENT), main="DEATH_EVENT Distribution",
xlab="DEATH_EVENT", ylab="Frequency",col = "lavender")
```

Output:



Here, barplot is used in order to create bar plots. The table method was used with an argument dataset$DEATH_EVENT to tabulate the DEATH_EVENT attribute with variable name and the frequency in the form of a table. The result was passed into the barplot method along with main to add the title, xlab to provide a label for the x-axis and ylab to provide a label for the y-axis.

MISSING VALUES
Detecting NA values in Dataset

❖ Show Missing Values for Whole Dataset

Code:

```
is.na(dataset)
sum(is.na(dataset))
```

Output:

```
> is.na(dataset)
       age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium   sex smoking  time DEATH_EVENT
  [1,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [2,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [3,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [4,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [5,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [6,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [7,] FALSE   FALSE                  FALSE     TRUE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [8,] FALSE    TRUE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
  [9,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [10,]  TRUE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [11,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [12,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE      TRUE            FALSE        FALSE FALSE    TRUE FALSE       FALSE
 [13,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [14,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [15,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [16,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [17,] FALSE   FALSE                   TRUE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [18,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE    TRUE FALSE       FALSE
 [19,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [20,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [21,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [22,] FALSE    TRUE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [23,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [24,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [25,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [26,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [27,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE      TRUE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [28,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [29,]  TRUE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [30,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [31,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [32,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [33,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [34,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [35,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [36,]  TRUE    TRUE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [37,] FALSE   FALSE                  FALSE     TRUE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [38,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [39,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [40,] FALSE    TRUE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [41,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [42,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [43,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [44,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE    TRUE FALSE       FALSE
 [45,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE      TRUE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [46,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [47,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [48,] FALSE    TRUE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [49,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [50,]  TRUE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [51,] FALSE   FALSE                   TRUE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [52,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [53,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [54,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [55,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [56,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [57,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [58,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [59,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [60,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [61,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [62,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [63,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [64,] FALSE   FALSE                   TRUE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [65,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [66,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [67,]  TRUE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [68,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [69,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [70,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [71,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [72,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [73,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [74,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [75,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE      TRUE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
 [76,] FALSE   FALSE                  FALSE    FALSE            FALSE              FALSE     FALSE            FALSE        FALSE FALSE   FALSE FALSE       FALSE
```

The above command is used to show missing values in whole dataset. When a NA value exists, the is.na method returns TRUE. Otherwise, it returns FALSE. The number of NA values in the dataset can be obtained by using sum(is.na(dataset)).

❖ Show Missing Values for each Attribute

Code:

colSums(is.na(dataset))

Output:

```
> colSums(is.na(dataset))
                      age                 anaemia creatinine_phosphokinase                diabetes         ejection_fraction
                        5                       5                         4                       4                         0
     high_blood_pressure                 platelets          serum_creatinine            serum_sodium                       sex
                        0                       6                         0                       0                         0
                  smoking                    time              DEATH_EVENT
                        3                       0                         0
```

The above command is used to show missing values for each attribute of given dataset.

❖ Show Missing Values for age

Code:

which(is.na(dataset$age))

Output:

```
> which(is.na(dataset$age))
[1] 10 29 36 50 67
>
```

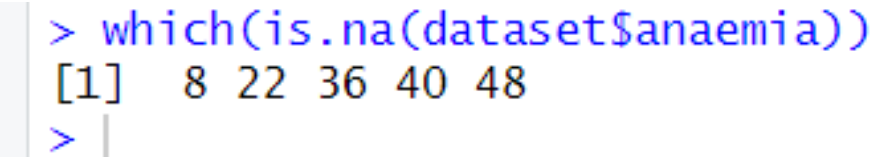
The above command is used to show missing values for age (shows the instance number where value is missing).

❖ Show Missing Values for anaemia

Code:

```
which(is.na(dataset$anaemia))
```

Output:



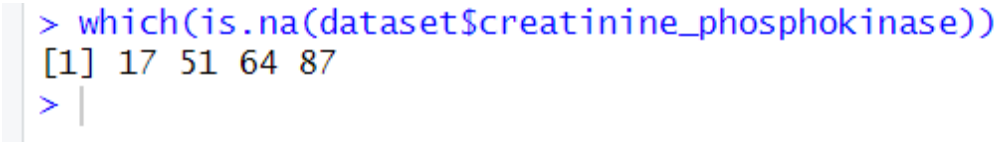
The above command is used to show missing values for anaemia (shows the instance number where value is missing).

❖ Show Missing Values for creatinine_phosphokinase

Code:

```
which(is.na(dataset$creatinine_phosphokinase))
```

Output:



The above command is used to show missing values for creatinine_phosphokinase (shows the instance number where value is missing).

❖ Show Missing Values for diabetes

Code:

```
which(is.na(dataset$diabetes))
```

Output:

```
> which(is.na(dataset$diabetes))
[1]   7  37 128 164
>
```

The above command is used to show missing values for diabetes (shows the instance number where value is missing).

❖ Show Missing Values for ejection_fraction

Code:

```
which(is.na(dataset$ejection_fraction))
```

Output:

```
> which(is.na(dataset$ejection_fraction))
integer(0)
>
```

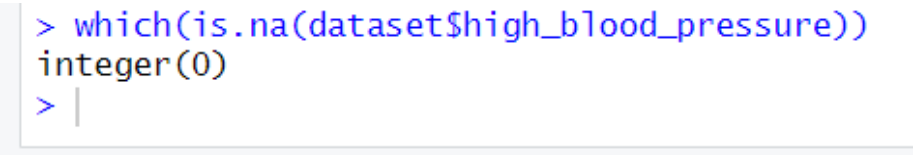
The above command is used to show missing values for ejection_fraction (shows the instance number where value is missing). We can see from the output there is no missing values in ejection_fraction.

❖ Show Missing Values for high_blood_pressure

Code:

```
which(is.na(dataset$high_blood_pressure))
```

Output:



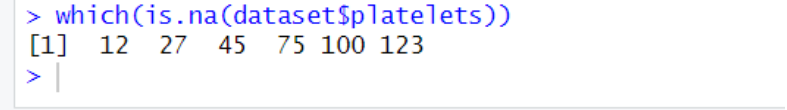
The above command is used to show missing values for high_blood_pressure (shows the instance number where value is missing). We can see from the output there is no missing values in high_ blood_pressure.

❖ Show Missing Values for platelets

Code:

```
which(is.na(dataset$platelets))
```

Output:



The above command is used to show missing values for platelets (shows the instance number where value is missing).

❖ Show Missing Values for serum_creatinine

Code:

```
which(is.na(dataset$serum_creatinine))
```

Output:

```
> which(is.na(dataset$serum_creatinine))
integer(0)
>
```

The above command is used to show missing values for serum_creatinine (shows the instance number where value is missing). We can see from the output there is no missing values in serum_creatinine.

❖ Show Missing Values for serum_sodium

Code:

```
which(is.na(dataset$serum_sodium))
```

Output:

```
> which(is.na(dataset$serum_sodium))
integer(0)
>
```

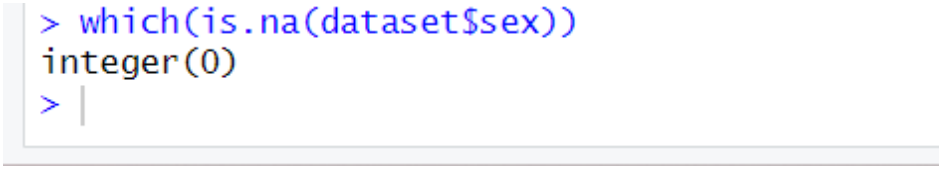
The above command is used to show missing values for serum_sodium (shows the instance number where value is missing). We can see from the output there is no missing values in serum_sodium.

❖ Show Missing Values for sex

Code:

```
which(is.na(dataset$sex))
```

Output:



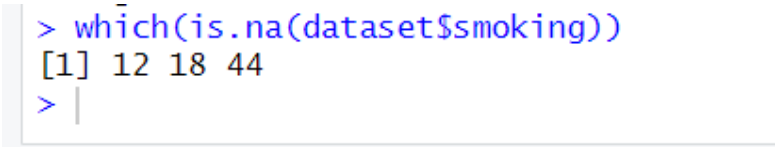
The above command is used to show missing values for sex (shows the instance number where value is missing). We can see from the output there is no missing values in sex.

❖ Show Missing Values for smoking

Code:

```
which(is.na(dataset$smoking))
```

Output:



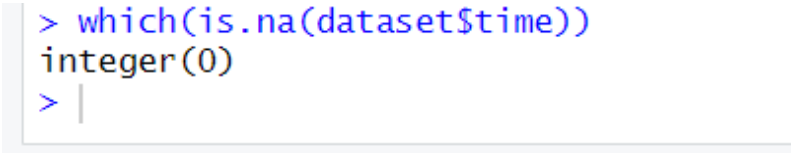
The above command is used to show missing values for smoking (shows the instance number where value is missing).

❖ Show Missing Values for time

Code:

```
which(is.na(dataset$time))
```

Output:



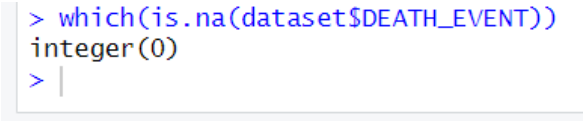
The above command is used to show missing values for time (shows the instance number where value is missing). We can see from the output there is no missing values in time.

❖ Show Missing Values for DEATH_EVENT

Code:

```
which(is.na(dataset$DEATH_EVENT))
```

Output:



The above command is used to show missing values for DEATH_EVENT (shows the instance number where value is missing). We can see from the output there is no missing values in DEATH_EVENT.

Discard Instances

Code:

```
dataset1 <- na.omit(dataset)
dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> dataset1
    age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium       sex smoking time DEATH_EVENT
1    75       0                     582        0                20                   1    265000            1.90           130      Male       0    4           1
2    55       0                    7861        0                38                   0    263358            1.10           136      Male       0    6           1
3    65       0                     146        0                20                   0    162000            1.30           129      Male       1    7           1
4    50       1                     111        0                20                   0    210000            1.90           137      Male       0    7           1
5    65       1                     160        1                20                   0    327000            2.70           116    Female       0    8           1
6    90       1                      47        0                40                   1    204000            2.10           132      Male       1    8           1
9    65       0                     157        0                65                   0    263358            1.50           138    Female       0   10           1
11   75       1                      81        0                38                   1    368000            4.00           131      Male       1   10           1
13   45       1                     981        0                30                   0    136000            1.10           137      Male       0   11           1
14   50       1                     168        0                38                   1    276000            1.10           137      Male       0   11           1
15   49       1                      80        0                30                   1    427000            1.00           138    Female       0   12           0
16   82       1                     379        0                50                   0     47000            1.30           136      Male       0   13           1
19   70       1                     125        0                25                   1    237000            1.00           140    Female       0   15           1
20   48       1                     582        1                55                   0     87000            1.90           121    Female       0   15           1
21   65       1                      52        0                25                   1    276000            1.30           137    Female       0   16           0
23   68       1                     220        0                35                   1    289000            0.90           140      Male       1   20           1
24   53       0                      63        1                60                   0    368000            0.80           135      Male       0   22           0
25   75       0                     582        1                30                   1    263358            1.83           134    Female       0   23           1
26   80       0                     148        1                38                   0    149000            1.90           144      Male       1   23           1
28   70       0                     122        1                45                   1    284000            1.30           136      Male       1   26           1
30   82       0                      70        1                30                   0    200000            1.20           132      Male       1   26           1
31   94       0                     582        1                38                   1    263358            1.83           134      Male       0   27           1
32   85       0                      23        0                45                   0    360000            3.00           132      Male       0   28           1
33   50       1                     249        1                35                   1    319000            1.00           128    Femmale      0   28           1
34   50       1                     159        1                30                   0    302000            1.20           138    Female       0   29           0
35   65       0                      94        1                50                   1    188000            1.00           140      Male       0   29           1
38   82       1                     855        1                50                   1    321000            1.00           145    Female       0   30           1
39   60       0                    2656        1                30                   0    305000            2.30           137      Male       0   30           0
41   70       0                     582        0                20                   1    263358            1.83           134      Male       1   31           1
42   50       0                     124        1                30                   1    153000            1.20           136    Female       1   32           1
43   70       0                     571        1                45                   1    185000            1.20           139      Male       1   33           1
46   50       0                     582        1                38                   0    310000            1.90           135      Male       1   35           1
47   51       0                    1380        0                25                   1    271000            0.90           130      Male       0   38           1
49   80       1                     553        0                20                   1    140000            4.40           133      Male       0   41           1
52   53       1                      91        0                20                   1    418000            1.40           139    Female       0   43           1
53   60       0                    3964        1                62                   0    263358            6.80           146    Female       0   43           1
54   70       1                      69        1                50                   1    351000            1.00           134    Female       0   44           1
55   60       1                     260        1                38                   0    255000            2.20           132    Female       1   45           1
56   95       1                     371        0                30                   0    461000            2.00           132      Male       0   50           1
57   70       1                      75        0                35                   0    223000            2.70           138      Male       1   54           0
58   60       1                     607        0                40                   0    216000            0.60           138      Male       1   54           0
59   49       0                     789        0                20                   1    319000            1.10           136      Male       1   55           1
60   72       0                     364        1                20                   1    254000            1.30           136      Male       1   59           1
61   45       0                    7702        1                25                   1    390000            1.00           139      Male       0   60           1
62   50       0                     318        0                40                   1    216000            2.30           131    Female       0   60           1
63   55       0                     109        0                35                   0    254000            1.10           139      Male       1   60           0
65   45       0                     582        0                80                   0    263358            1.18           137    Female       0   63           0
66   60       0                      68        0                20                   0    119000            2.90           127      Male       1   64           1
68   72       1                     110        0                25                   0    274000            1.00           140      Male       1   65           1
69   70       0                     161        0                25                   0    244000            1.20           142    Female       0   66           1
70   65       0                     113        1                25                   0    497000            1.83           135      Male       0   67           1
71   41       0                     148        0                40                   0    374000            0.80           140      Male       1   68           0
72   58       0                     582        1                35                   0    122000            0.90           139      Male       1   71           0
73   85       0                    5882        0                35                   0    243000            1.00           132      Male       1   72           1
74   65       0                     224        1                50                   0    149000            1.30           137      Male       1   72           0
76   60       1                      47        0                20                   0    204000            0.70           139      Male       1   73           1
77   70       0                      92        0                60                   1    317000            0.80           140    Female       1   74           0
78   42       0                     102        1                40                   0    237000            1.20           140      Male       0   74           0
79   75       1                     203        1                38                   1    283000            0.60           131      Male       1   74           0
80   55       0                     336        0                45                   1    324000            0.90           140    Female       0   74           0
81   70       0                      69        0                40                   0    293000            1.70           136    Female       0   75           0
82   67       0                     582        0                50                   0    263358            1.18           137      Male       1   76           0
83   60       1                      76        1                25                   0    196000            2.50           132    Female       0   77           1
84   79       1                     55X        0                50                   1    172000            1.80           133      Male       0   78           0
85   59       1                     280        1                25                   1    302000            1.00           141    Female       0   78           1
86   51       0                      78        0                50                   0    406000            0.70           140      Male       0   79           0
88   65       1                      68        1                60                   1    304000            0.80           140      Male       0   79           0
89   44       0                      84        1                40                   1    235000            0.70           139      Male       0   79           0
90   57       1                     115        0                25                   1    181000            1.10           144      Male       0   79           0
91   70       0                      66        1                45                   0    249000            0.80           136      Male       1   80           0
92   60       0                     897        1                45                   0    297000            1.00           133      Male       0   80           0
93   42       0                     582        0                60                   0    263358            1.18           137    Female       0   82           0
94   60       1                     154        0                25                   0    210000            1.70           135      Male       0   82           1
95   58       0                     144        1                38                   1    327000            0.70           142    Female       0   83           0
96   58       1                     133        0                60                   1    219000            1.00           141      Male       0   83           0
97   63       1                     514        1                25                   1    254000            1.30           134      Male       0   83           0
```

All instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset 1,after the removal of the NA values.

Replace by Most Frequent Value

❖ Replacing NA values for age using Mode

Code:

```
missingValues <- which(is.na(dataset$age))
ageMode <- names(sort(table(dataset$age[!is.na(dataset$age)]), decreasing = TRUE)[1])
dataset$age[missingValues] <- ageMode
cat("Age Mode:", ageMode)
dataset$age
```

Output:

```
> missingValues <- which(is.na(dataset$age))
> ageMode <- names(sort(table(dataset$age[!is.na(dataset$age)]), decreasing = TRUE)[1])
> dataset$age[missingValues] <- ageMode
> cat("Age Mode:", ageMode)
Age Mode: 60> dataset$age
  [1] "75"    "55"    "65"    "50"    "65"    "90"    "75"    "60"    "65"    "60"    "75"    "62"    "45"    "50"    "49"    "82"    "87"    "45"    "70"    "48"
 [21] "65"    "65"    "68"    "53"    "75"    "80"    "95"    "70"    "60"    "82"    "94"    "85"    "50"    "50"    "65"    "60"    "90"    "82"    "60"    "60"
 [41] "70"    "50"    "70"    "72"    "60"    "50"    "51"    "60"    "80"    "60"    "68"    "53"    "60"    "70"    "60"    "95"    "70"    "60"    "49"    "72"
 [61] "45"    "50"    "55"    "45"    "45"    "60"    "60"    "72"    "70"    "65"    "41"    "58"    "85"    "65"    "69"    "60"    "70"    "42"    "75"    "55"
 [81] "70"    "67"    "60"    "79"    "59"    "51"    "55"    "65"    "44"    "57"    "70"    "60"    "42"    "60"    "58"    "58"    "63"    "70"    "60"    "63"
[101] "65"    "75"    "80"    "42"    "60"    "72"    "55"    "45"    "63"    "45"    "85"    "55"    "50"    "70"    "60"    "58"    "60"    "85"    "65"    "86"
[121] "60"    "66"    "60"    "60"    "60"    "43"    "46"    "58"    "61"    "53"    "53"    "60"    "46"    "63"    "81"    "75"    "65"    "68"    "62"    "50"
[141] "80"    "46"    "50"    "161"   "72"    "50"    "52"    "64"    "75"    "60"    "72"    "62"    "50"    "50"    "65"    "60"    "52"    "50"    "85"    "59"
[161] "66"    "45"    "63"    "50"    "45"    "80"    "53"    "59"    "65"    "70"    "51"    "52"    "70"    "50"    "65"    "60"    "69"    "49"    "63"    "55"
[181] "40"    "59"    "65"    "75"    "58"    "170"   "50"    "60"    "60.66" "40"    "80"    "64"    "50"    "73"    "45"    "77"    "45"    "65"    "50"
>
```

First, NA values of age attribute were stored in missingValues using the which(is.na(dataset$age)) method. Then table method returns data in a tabular format with variable name. dataset$age[!is.na(dataset$age)] was passed as parameter to tabulate only the age attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in ageMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of age attribute were replaced by ageMode.

❖ **Replacing NA values for anaemia using Mode**

Code:

```
missingValues <- which(is.na(dataset$anaemia))
anaemiaMode <- names(sort(table(dataset$anaemia[!is.na(dataset$anaemia)]), decreasing =
TRUE)[1])
dataset$anaemia[missingValues] <- anaemiaMode
cat("Anaemia Mode:", anaemiaMode)
dataset$anaemia
```

Output:

```
> missingValues <- which(is.na(dataset$anaemia))
> anaemiaMode <- names(sort(table(dataset$anaemia[!is.na(dataset$anaemia)]), decreasing = TRUE)[1])
> dataset$anaemia[missingValues] <- anaemiaMode
> cat("Anaemia Mode:", anaemiaMode)
Anaemia Mode: 0> dataset$anaemia
  [1] "0" "0" "0" "1" "1" "1" "1" "0" "0" "1" "1" "0" "1" "1" "1" "1" "1" "0" "1" "1" "1" "0" "1" "0" "0" "0" "1" "0" "1" "0" "0" "0" "1" "1" "0" "0" "1" "1" "0" "0"
 [41] "0" "0" "0" "0" "1" "0" "0" "0" "1" "1" "1" "1" "0" "1" "1" "1" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "1" "0"
 [81] "0" "0" "1" "1" "1" "0" "0" "1" "0" "1" "0" "0" "0" "1" "0" "1" "1" "1" "1" "1" "1" "0" "0" "0" "0" "1" "0" "1" "0" "0" "0" "0" "0" "1" "1" "1" "1" "1" "1" "0"
[121] "1" "1" "0" "1" "0" "1" "0" "1" "0" "1" "1" "1" "0" "0" "0" "0" "1" "1" "0" "0" "0" "1" "0" "1" "1" "0" "0" "0" "1" "0" "0" "0" "0" "0" "1" "1" "1" "0" "1" "1"
[161] "1" "1" "1" "1" "0" "0" "0" "0" "0" "0" "1" "0" "1" "1" "0" "1" "0" "1" "1" "0" "0" "1" "0" "0" "1" "1" "0" "0" "1" "0" "0" "1" "1" "1" "0" "1" "0" "0" "1"
>
```

First, NA values of anaemia attribute were stored in missingValues using the which(is.na(dataset$anaemia)) method. Then table method returns data in a tabular format with variable name. dataset$anaemia [!is.na(dataset$anaemia)] was passed as parameter to tabulate only the anaemia attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in anaemiaMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of anaemia attribute were replaced by anaemiaMode.

❖ **Replacing NA values for creatinine_phosphokinase using Mode**

Code:

```
missingValues <- which(is.na(dataset$creatinine_phosphokinase))

creatinine_phosphokinaseMode <-
names(sort(table(dataset$creatinine_phosphokinase[!is.na(dataset$creatinine_phosphokinase)]),
decreasing = TRUE)[1])

dataset$creatinine_phosphokinase[missingValues] <- creatinine_phosphokinaseMode

cat("Creatinine_phosphokinase Mode:", creatinine_phosphokinaseMode)

dataset$creatinine_phosphokinase
```

Output:

```
> missingValues <- which(is.na(dataset$creatinine_phosphokinase))
> creatinine_phosphokinaseMode <- names(sort(table(dataset$creatinine_phosphokinase[!is.na(dataset$creatinine_phosphokinase)]), decreasing = TRUE)[1])
> dataset$creatinine_phosphokinase[missingValues] <- creatinine_phosphokinaseMode
> cat("Creatinine_phosphokinase Mode:", creatinine_phosphokinaseMode)
Creatinine_phosphokinase Mode: 582> dataset$creatinine_phosphokinase
  [1] "582"  "7861" "146"  "111"  "160"  "47"   "246"  "315"  "157"  "123"  "81"   "231"  "981"  "168"  "80"   "379"  "582"  "582"  "125"  "582"  "52"   "128"  "220"
 [24] "63"   "582"  "148"  "112"  "122"  "60"   "70"   "582"  "23"   "249"  "159"  "94"   "582"  "60"   "855"  "2656" "235"  "582"  "124"  "571"  "127"  "588"  "582"
 [47] "1380" "582"  "553"  "129"  "582"  "91"   "3964" "69"   "260"  "371"  "75"   "607"  "789"  "364"  "7702" "318"  "109"  "582"  "582"  "68"   "250"  "110"  "161"
 [70] "113"  "148"  "582"  "5882" "224"  "582"  "47"   "92"   "102"  "203"  "336"  "69"   "582"  "76"   "55X"  "280"  "78"   "582"  "68"   "84"   "115"  "66"   "897"
 [93] "582"  "154"  "144"  "133"  "514"  "59"   "156"  "61"   "305"  "582"  "898"  "5209" "53"   "328"  "748"  "1876" "936"  "292"  "129"  "60"   "369"  "143"  "754"
[116] "400"  "96"   "102"  "113"  "582"  "737"  "68"   "96"   "582"  "582"  "358"  "168"  "200"  "248"  "270"  "1808" "1082" "719"  "193"  "4540" "582"  "59"   "646"
[139] "281"  "1548" "805"  "291"  "482"  "84"   "943"  "185"  "132"  "1610" "582"  "2261" "233"  "30"   "115"  "1846" "335"  "231"  "58"   "250"  "910"  "129"  "72"
[162] "130"  "582"  "2334" "2442" "776"  "196"  "66"   "582"  "835"  "582"  "3966" "171"  "115"  "198"  "95"   "1419" "69"   "122"  "835"  "478"  "176"  "395"  "99"
[185] "145"  "104"  "582"  "1896" "151"  "244"  "582"  "62"   "121"  "231"  "582"  "418"  "582"  "167"  "582"
>
```

First, NA values of creatinine_phosphokinase attribute were stored in missingValues using the which(is.na(dataset$creatinine_phosphokinase)) method. Then table method returns data in a tabular format with variable name. dataset$creatinine_phosphokinase [!is.na(dataset$creatinine_phosphokinase)] was passed as parameter to tabulate only the creatinine_phosphokinase attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in creatinine_phosphokinaseMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of creatinine_phosphokinase attribute were replaced by creatinine_phosphokinaseMode.

❖ **Replacing NA values for diabetes using Mode**

Code:

```
missingValues <- which(is.na(dataset$diabetes))

diabetesMode <- names(sort(table(dataset$diabetes[!is.na(dataset$diabetes)]), decreasing = TRUE)[1])

dataset$diabetes[missingValues] <- diabetesMode

cat("Diabetes Mode:", diabetesMode)

dataset$diabetes
```

Output:

```
> missingValues <- which(is.na(dataset$diabetes))
> diabetesMode <- names(sort(table(dataset$diabetes[!is.na(dataset$diabetes)]), decreasing = TRUE)[1])
> dataset$diabetes[missingValues] <- diabetesMode
> cat("Diabetes Mode:", diabetesMode)
Diabetes Mode: 0> dataset$diabetes
  [1] "0" "0" "0" "0" "1" "0" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "1" "0" "1" "1" "1" "0" "1" "0" "1" "1" "0" "1" "1" "1" "1" "0" "1" "1" "1"
 [41] "0" "1" "1" "1" "1" "1" "0" "1" "0" "0" "0" "0" "1" "1" "1" "0" "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "1" "0" "0" "1" "0" "1" "0" "1" "0" "0" "0" "1" "1" "0"
 [81] "0" "0" "1" "0" "1" "0" "0" "1" "1" "0" "1" "1" "0" "0" "1" "0" "1" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "1" "0" "1" "0" "0" "1" "0" "1" "0" "1" "0" "1" "0"
[121] "0" "1" "1" "0" "0" "0" "1" "0" "0" "1" "0" "1" "0" "0" "0" "0" "1" "0" "1" "0" "0" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "1" "0" "1" "0" "0" "0" "0"
[161] "0" "0" "0" "0" "1" "1" "0" "1" "1" "0" "1" "0" "0" "0" "1" "0" "0" "0" "1" "0" "1" "1" "1" "0" "0" "1" "0" "1" "1" "0" "1" "0" "1" "1" "0" "0" "1" "0" "1"
>
```

First, NA values of diabetes attribute were stored in missingValues using the which(is.na(dataset$diabetes)) method. Then table method returns data in a tabular format with variable name. dataset$ diabetes [!is.na(dataset$diabetes)] was passed as parameter to tabulate only the diabetes attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in diabetes Mode. cat method was used to print the mode (most frequent value) and lastly all the NA values of diabetes attribute were replaced by diabetes Mode.

❖ **Replacing NA values for ejection_fraction using Mode**

Code:

```
missingValues <- which(is.na(dataset$ejection_fraction))

ejection_fractionMode <-
names(sort(table(dataset$ejection_fraction[!is.na(dataset$ejection_fraction)]), decreasing =
TRUE)[1])

dataset$ejection_fraction[missingValues] <- ejection_fractionMode

cat("Ejection_fraction Mode:", ejection_fractionMode)

dataset$ejection_fraction
```

Output:

```
> missingValues <- which(is.na(dataset$ejection_fraction))
> ejection_fractionMode <- names(sort(table(dataset$ejection_fraction[!is.na(dataset$ejection_fraction)]), decreasing = TRUE)[1])
> dataset$ejection_fraction[missingValues] <- ejection_fractionMode
> cat("Ejection_fraction Mode:", ejection_fractionMode)
Ejection_fraction Mode: 35> dataset$ejection_fraction
  [1] "20" "38" "20" "20" "20" "40" "15" "60" "65" "35" "38" "25" "30" "38" "30" "50" "38" "14" "25" "55" "25" "30" "35" "60" "30" "38" "40" "45" "38" "30" "38" "45"
 [33] "35" "30" "50" "35" "50" "50" "30" "38" "20" "30" "45" "50" "60" "38" "25" "38" "20" "30" "25" "20" "62" "50" "38" "30" "35" "40" "20" "20" "25" "40" "35" "35"
 [65] "80" "20" "15" "25" "25" "25" "40" "35" "35" "50" "20" "20" "60" "40" "38" "45" "40" "50" "25" "50" "25" "50" "35" "60" "40" "25" "45" "45" "60" "25" "38" "60"
 [97] "25" "60" "25" "40" "25" "45" "25" "30" "50" "30" "45" "35" "38" "35" "60" "35" "25" "60" "40" "40" "60" "60" "60" "38" "60" "38" "38" "30" "40" "50" "17" "60"
[129] "30" "35" "60" "45" "40" "60" "35" "40" "60" "25" "35" "30" "38" "35" "30" "40" "25" "30" "30" "60" "30" "35" "45" "60" "45" "35" "35" "25" "35" "25" "50" "45"
[161] "40" "35" "40" "35" "30" "38" "60" "20" "40" "35" "35" "40" "60" "20" "35" "60" "40" "50" "60" "40" "30" "25" "25" "38" "25" "30" "50" "25" "40" "45" "35" "60"
[193] "40" "30" "20" "45" "38" "30" "20"
>
```

First, NA values of ejection_fraction attribute were stored in missingValues using the which(is.na(dataset$ejection_fraction)) method. Then table method returns data in a tabular format with variable name. dataset$ejection_fraction [!is.na(dataset$ejection_fraction)] was passed as parameter to tabulate only the ejection_fraction attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in ejection_fraction Mode. cat method was used to print the mode (most frequent value) and lastly all the NA values of ejection_fraction attribute were replaced by ejection_fraction Mode.

❖ **Replacing NA values for high_blood_pressure using Mode**

Code:

```
missingValues <- which(is.na(dataset$high_blood_pressure))

high_blood_pressureMode <-
names(sort(table(dataset$high_blood_pressure[!is.na(dataset$high_blood_pressure)]), decreasing
= TRUE)[1])

dataset$high_blood_pressure[missingValues] <- high_blood_pressureMode

cat("High_blood_pressure Mode:", high_blood_pressureMode)

dataset$high_blood_pressure
```

Output:

```
> missingValues <- which(is.na(dataset$high_blood_pressure))
> high_blood_pressureMode <- names(sort(table(dataset$high_blood_pressure[!is.na(dataset$high_blood_pressure)]), decreasing = TRUE)[1])
> dataset$high_blood_pressure[missingValues] <- high_blood_pressureMode
> cat("High_blood_pressure Mode:", high_blood_pressureMode)
High_blood_pressure Mode: 0> dataset$high_blood_pressure
  [1] "1" "0" "0" "0" "0" "1" "0" "0" "0" "1" "1" "1" "0" "1" "1" "0" "0" "0" "1" "0" "1" "1" "1" "0" "1" "0" "1" "1" "0" "0" "1" "0" "1" "0" "1" "0" "0" "1" "0" "0"
 [41] "1" "1" "1" "1" "0" "0" "1" "1" "1" "0" "1" "1" "0" "1" "0" "0" "0" "0" "1" "1" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "1" "1"
 [81] "0" "0" "0" "1" "1" "0" "1" "1" "1" "1" "0" "0" "0" "0" "1" "1" "1" "0" "1" "0" "0" "1" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "1" "0" "1" "0" "1" "0"
[121] "1" "1" "0" "1" "0" "0" "1" "0" "1" "0" "1" "0" "1" "1" "0" "0" "0" "0" "0" "1" "0" "0" "0" "1" "1" "0" "0" "0" "0" "1" "1" "1" "1" "0" "1" "0" "0" "0" "0" "1"
[161] "1" "0" "0" "0" "0" "1" "0" "0" "0" "1" "0" "0" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0" "1" "1" "0" "0" "0" "0" "1" "0" "1" "0" "1"
>
```

First, NA values of high_blood_pressure attribute were stored in missingValues using the which(is.na(dataset$high_blood_pressure)) method. Then table method returns data in a tabular format with variable name. dataset$high_blood_pressure[!is.na(dataset$high_blood_pressure)] was passed as parameter to tabulate only the high_blood_pressure attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in high_blood_pressureMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of high_blood_pressure attribute were replaced by high_blood_pressure Mode.

❖ **Replacing NA values for serum_creatinine using Mode**

Code:

```
missingValues <- which(is.na(dataset$serum_creatinine))

serum_creatinineMode <-
names(sort(table(dataset$serum_creatinine[!is.na(dataset$serum_creatinine)]), decreasing =
TRUE)[1])

dataset$serum_creatinine[missingValues] <- serum_creatinineMode

cat("Serum_creatinine Mode:", serum_creatinineMode)

dataset$serum_creatinine
```

Output:

```
> missingValues <- which(is.na(dataset$serum_creatinine))
> serum_creatinineMode <- names(sort(table(dataset$serum_creatinine[!is.na(dataset$serum_creatinine)]), decreasing = TRUE)[1])
> dataset$serum_creatinine[missingValues] <- serum_creatinineMode
> cat("Serum_creatinine Mode:", serum_creatinineMode)
Serum_creatinine Mode: 1> dataset$serum_creatinine
  [1] "1.9"  "1.1"  "1.3"  "1.9"  "2.7"  "2.1"  "1.2"  "1.1"  "1.5"  "9.4"  "4"    "0.9"  "1.1"  "1.1"  "1"    "1.3"  "0.9"  "0.8"  "1"    "1.9"  "1.3"  "1.6"  "0.9"
 [24] "0.8"  "1.83" "1.9"  "1"    "1.3"  "5.8"  "1.2"  "1.83" "3"    "1"    "1.2"  "1"    "3.5"  "1"    "1"    "2.3"  "3"    "1.83" "1.2"  "1.2"  "1"    "1.1"  "1.9"
 [47] "0.9"  "0.6"  "4.4"  "1"    "1"    "1.4"  "6.8"  "1"    "2.2"  "2"    "2.7"  "0.6"  "1.1"  "1.3"  "1"    "2.3"  "1.1"  "1"    "1.18" "2.9"  "1.3"  "1"    "1.2"
 [70] "1.83" "0.8"  "0.9"  "1"    "1.3"  "1.2"  "0.7"  "0.8"  "1.2"  "0.6"  "0.9"  "1.7"  "1.18" "2.5"  "1.8"  "1"    "0.7"  "1.1"  "0.8"  "0.7"  "1.1"  "0.8"  "1"
 [93] "1.18" "1.7"  "0.7"  "1"    "1.3"  "1.1"  "1.2"  "1.1"  "1.1"  "1.18" "1.1"  "1"    "2.3"  "1.7"  "1.3"  "0.9"  "1.1"  "1.3"  "1.2"  "1.2"  "1.6"  "1.3"  "1.2"
[116] "1"    "0.7"  "3.2"  "0.9"  "1.83" "1.5"  "1"    "0.75" "0.9"  "3.7"  "1.3"  "2.1"  "0.8"  "0.7"  "3.4"  "0.7"  "6.1"  "1.18" "1.3"  "1.18" "1.18" "0.9"  "2.1"
[139] "1"    "0.8"  "1.1"  "0.9"  "0.9"  "0.9"  "1.7"  "0.7"  "0.7"  "1"    "1.83" "0.9"  "2.5"  "0.9"  "0.9"  "1.18" "0.8"  "1.7"  "1.4"  "1"    "1.3"  "1.1"  "1.2"
[162] "0.8"  "0.9"  "0.9"  "1.1"  "1.3"  "0.7"  "2.4"  "1"    "0.8"  "1.5"  "0.9"  "1.1"  "0.8"  "0.9"  "1"    "1"    "1"    "1.2"  "0.7"  "0.9"  "1"    "1.2"  "2.5"
[185] "1.2"  "1.5"  "0.6"  "2.1"  "1"    "0.9"  "2.1"  "1.5"  "0.7"  "1.18" "1.6"  "1.8"  "1.18" "0.8"  "1"
>
```

First, NA values of serum_creatinine attribute were stored in missingValues using the which(is.na(dataset$serum_creatinine)) method. Then table method returns data in a tabular format with variable name. dataset$serum_creatinine [!is.na(dataset$serum_creatinine)] was passed as parameter to tabulate only the serum_creatinine attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in serum_creatinineMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of serum_creatinineattribute were replaced by serum_creatinineMode.

❖ **Replacing NA values for serum_sodium using Mode**

Code:

```
missingValues <- which(is.na(dataset$serum_sodium))

serum_sodiumMode <-
names(sort(table(dataset$serum_sodium[!is.na(dataset$serum_sodium)]), decreasing =
TRUE)[1])

dataset$serum_sodium[missingValues] <- serum_sodiumMode

cat("Serum_sodium Mode:", serum_sodiumMode)

dataset$serum_sodium
```

Output:

```
> missingValues <- which(is.na(dataset$serum_sodium))
> serum_sodiumMode <- names(sort(table(dataset$serum_sodium[!is.na(dataset$serum_sodium)]), decreasing = TRUE)[1])
> dataset$serum_sodium[missingValues] <- serum_sodiumMode
> cat("Serum_sodium Mode:", serum_sodiumMode)
Serum_sodium Mode: 136> dataset$serum_sodium
  [1] "130" "136" "129" "137" "116" "132" "137" "131" "138" "133" "131" "140" "137" "137" "138" "136" "140" "127" "140" "121" "137" "136" "140" "135" "134" "144"
 [27] "138" "136" "134" "132" "134" "132" "128" "138" "140" "134" "134" "145" "137" "142" "134" "136" "139" "134" "142" "135" "130" "138" "133" "140" "138" "139"
 [53] "146" "134" "132" "132" "138" "138" "136" "136" "139" "131" "139" "145" "137" "127" "136" "140" "142" "135" "140" "139" "132" "137" "134" "139" "140" "140"
 [79] "131" "140" "136" "137" "132" "133" "141" "140" "137" "140" "139" "144" "136" "133" "137" "135" "142" "141" "134" "136" "137" "140" "141" "137" "144" "140"
[105] "143" "138" "137" "138" "133" "142" "132" "135" "136" "137" "126" "139" "136" "138" "140" "134" "135" "136" "140" "145" "134" "135" "124" "137" "136" "145"
[131] "138" "131" "137" "145" "137" "137" "137" "130" "136" "138" "134" "140" "132" "141" "139" "141" "136" "137" "134" "136" "135" "139" "134" "137" "136" "140"
[157] "136" "136" "134" "139" "134" "139" "137" "142" "139" "135" "133" "134" "138" "133" "136" "140" "145" "139" "137" "138" "135" "140" "145" "140" "136" "136"
[183] "136" "134" "137" "136" "134" "144" "136" "140" "134" "135" "130" "142" "135" "145" "137" "138" "134"
>
```

First, NA values of serum_sodium attribute were stored in missingValues using the which(is.na(dataset$ serum_sodium)) method. Then table method returns data in a tabular format with variable name. dataset$serum_sodium[!is.na(dataset$serum_sodium)] was passed as parameter to tabulate only the serum_sodium attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in serum_sodiumMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of serum_sodium attribute were replaced by serum_sodiumMode.

❖ **Replacing NA values for sex using Mode**

Code:

```
missingValues <- which(is.na(dataset$sex))
sexMode <- names(sort(table(dataset$sex[!is.na(dataset$sex)]), decreasing = TRUE)[1])
dataset$sex[missingValues] <- sexMode
cat("Sex Mode:", sexMode)
dataset$sex
```

Output:

```
> missingValues <- which(is.na(dataset$sex))
> sexMode <- names(sort(table(dataset$sex[!is.na(dataset$sex)]), decreasing = TRUE)[1])
> dataset$sex[missingValues] <- sexMode
> cat("Sex Mode:", sexMode)
Sex Mode: Male> dataset$sex
  [1] "Male"    "Male"    "Male"    "Male"    "Female"  "Male"    "Male"    "Male"    "Female"  "Maleee"  "Male"    "Male"    "Male"    "Male"    "Female"  "Male"
 [17] "Male"    "Male"    "Female"  "Female"  "Female"  "Female"  "Male"    "Male"    "Female"  "Male"    "Female"  "Male"    "Male"    "Male"    "Male"    "Male"
 [33] "Femmale" "Female"  "Male"    "Male"    "Male"    "Female"  "Male"    "Female"  "Male"    "Female"  "Male"    "Male"    "Female"  "Male"    "Male"    "Male"
 [49] "Male"    "Female"  "Male"    "Female"  "Female"  "Female"  "Female"  "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Female"  "Male"    "Male"
 [65] "Female"  "Male"    "Female"  "Male"    "Female"  "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Female"  "Male"    "Male"    "Female"
 [81] "Female"  "Male"    "Female"  "Male"    "Female"  "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Female"  "Male"    "Female"  "Male"
 [97] "Male"    "Female"  "Female"  "Female"  "Male"    "Male"    "Male"    "Male"    "Female"  "Female"  "Male"    "Male"    "Male"    "Male"    "Male"    "Male"
[113] "Male"    "Female"  "Male"    "Female"  "Female"  "Female"  "Female"  "Female"  "Male"    "Female"  "Female"  "Female"  "Male"    "Female"  "Female"  "Female"
[129] "Male"    "Male"    "Male"    "Male"    "Female"  "Male"    "Male"    "Male"    "Female"  "Male"    "Female"  "Male"    "Male"    "Female"  "Female"  "Female"
[145] "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Female"  "Male"    "Male"    "Male"    "Female"  "Male"    "Female"  "Male"    "Male"    "Male"
[161] "Male"    "Male"    "Male"    "Female"  "Male"    "Female"  "Male"    "Male"    "Female"  "Female"  "Male"    "Male"    "Male"    "Male"    "Male"    "Male"
[177] "Male"    "Female"  "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Male"    "Female"  "Female"  "Female"  "Female"  "Male"    "Female"
[193] "Male"    "Male"    "Male"    "Male"    "Female"  "Female"  "Female"
>
```

First, NA values of sex attribute were stored in missingValues using the which(is.na(dataset$ sex)) method. Then table method returns data in a tabular format with variable name. dataset$sex [!is.na(dataset$sex)] was passed as parameter to tabulate only the sex attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in sexMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of sex attribute were replaced by sexMode.

❖ **Replacing NA values for smoking using Mode**

Code:

```
missingValues <- which(is.na(dataset$smoking))

smokingMode <- names(sort(table(dataset$smoking[!is.na(dataset$smoking)]), decreasing = TRUE)[1])

dataset$smoking[missingValues] <- smokingMode

cat("Smoking Mode:", smokingMode)

dataset$smoking
```

Output:

```
> missingValues <- which(is.na(dataset$smoking))
> smokingMode <- names(sort(table(dataset$smoking[!is.na(dataset$smoking)]), decreasing = TRUE)[1])
> dataset$smoking[missingValues] <- smokingMode
> cat("Smoking Mode:", smokingMode)
Smoking Mode: 0> dataset$smoking
  [1] "0" "0" "1" "0" "0" "1" "0" "1" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "1" "0" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
 [41] "1" "1" "1" "0" "0" "1" "0" "1" "0" "0" "0" "0" "0" "0" "1" "0" "1" "1" "1" "1" "0" "0" "1" "0" "0" "1" "0" "1" "0" "0" "1" "1" "1" "1" "1" "1" "1" "0" "1" "0"
 [81] "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "1" "0" "1" "0" "0" "1" "1" "1" "1" "0" "0" "0" "0" "0" "0" "0"
[121] "1" "0" "0" "0" "0" "0" "0" "0" "1" "0" "1" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "1" "1" "0" "0" "0" "0" "0" "1" "1" "0" "0" "0" "1" "0" "1"
[161] "0" "1" "1" "0" "0" "0" "1" "0" "0" "0" "1" "1" "1" "0" "1" "1" "1" "0" "0" "1" "0" "1" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0"
>
```

First, NA values of smoking attribute were stored in missingValues using the which(is.na(dataset$ smoking)) method. Then table method returns data in a tabular format with variable name. dataset$ smoking [!is.na(dataset$ smoking)] was passed as parameter to tabulate only the smoking attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in smokingMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of smoking attribute were replaced by smokingMode.

❖ **Replacing NA values for time using Mode**

Code:

```
missingValues <- which(is.na(dataset$time))
timeMode <- names(sort(table(dataset$time[!is.na(dataset$time)]), decreasing = TRUE)[1])
dataset$time[missingValues] <- timeMode
cat("Time Mode:", timeMode)
dataset$time
```

Output:

```
> missingValues <- which(is.na(dataset$time))
> timeMode <- names(sort(table(dataset$time[!is.na(dataset$time)]), decreasing = TRUE)[1])
> dataset$time[missingValues] <- timeMode
> cat("Time Mode:", timeMode)
Time Mode: 10> dataset$time
  [1] "4"   "6"   "7"   "7"   "8"   "8"   "10"  "10"  "10"  "10"  "10"  "10"  "11"  "11"  "12"  "13"  "14"  "14"  "15"  "15"  "16"  "20"  "20"  "22"  "23"  "23"
 [27] "24"  "26"  "26"  "26"  "27"  "28"  "28"  "29"  "29"  "30"  "30"  "30"  "30"  "30"  "31"  "32"  "33"  "33"  "33"  "35"  "38"  "40"  "41"  "42"  "43"  "43"
 [53] "43"  "44"  "45"  "50"  "54"  "54"  "55"  "59"  "60"  "60"  "60"  "61"  "63"  "64"  "65"  "65"  "66"  "67"  "68"  "71"  "72"  "72"  "73"  "73"  "74"  "74"
 [79] "74"  "74"  "75"  "76"  "77"  "78"  "78"  "79"  "79"  "79"  "79"  "79"  "80"  "80"  "82"  "82"  "83"  "83"  "83"  "85"  "85"  "86"  "87"  "87"  "87"  "87"
[105] "87"  "88"  "88"  "88"  "88"  "88"  "90"  "90"  "90"  "90"  "91"  "91"  "94"  "94"  "94"  "95"  "95"  "95"  "95"  "95"  "96"  "97"  "100" "104" "104" "105"
[131] "106" "107" "107" "107" "107" "107" "107" "108" "108" "108" "109" "109" "109" "110" "111" "112" "112" "113" "113" "115" "115" "117" "118" "119" "120" "120"
[157] "120" "120" "121" "121" "121" "121" "123" "126" "129" "130" "134" "135" "140" "145" "145" "146" "146" "146" "146" "147" "147" "147" "147" "148" "150"
[183] "154" "162" "170" "171" "172" "172" "172" "174" "174" "174" "175" "180" "180" "180" "185" "186" "186"
>
```

First, NA values of time attribute were stored in missingValues using the which(is.na(dataset$ time)) method. Then table method returns data in a tabular format with variable name. dataset$ time[!is.na(dataset$time)] was passed as parameter to tabulate only the time attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in timeMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of time attribute were replaced by timeMode.

❖ **Replacing NA values for DEATH_EVENT using Mode**

Code:

```
missingValues <- which(is.na(dataset$DEATH_EVENT))

DEATH_EVENTMode <-
names(sort(table(dataset$DEATH_EVENT[!is.na(dataset$DEATH_EVENT)]), decreasing =
TRUE)[1])

dataset$DEATH_EVENT[missingValues] <- DEATH_EVENTMode

cat("DEATH_EVENT Mode:", DEATH_EVENTMode)

dataset$DEATH_EVENT
```

Output:

```
> missingValues <- which(is.na(dataset$DEATH_EVENT))
> DEATH_EVENTMode <- names(sort(table(dataset$DEATH_EVENT[!is.na(dataset$DEATH_EVENT)]), decreasing = TRUE)[1])
> dataset$DEATH_EVENT[missingValues] <- DEATH_EVENTMode
> cat("DEATH_EVENT Mode:", DEATH_EVENTMode)
DEATH_EVENT Mode: 0> dataset$DEATH_EVENT
  [1] "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" "1" "0" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" "0" "1"
 [41] "1" "1" "1" "0" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1" "0" "0" "1" "1" "1" "1" "0" "1" "0" "1" "1" "1" "1" "1" "0" "0" "1" "0" "1" "1" "0" "0" "0" "0"
 [81] "0" "0" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "1" "0" "0" "1" "0" "0" "0" "0" "0" "0" "1"
[121] "0" "0" "0" "0" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "1" "0" "0" "0" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0"
[161] "0" "0" "0" "1" "1" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "1" "1" "1" "1" "1" "1" "1" "0" "0" "0" "0" "0" "0" "1" "1" "0" "0" "0"
>
```

First, NA values of DEATH_EVENT attribute were stored in missingValues using the which(is.na(dataset$DEATH_EVENT)) method. Then table method returns data in a tabular format with variable name. dataset$DEATH_EVENT [!is.na(dataset$DEATH_EVENT time)] was passed as parameter to tabulate only the DEATH_EVENT attribute. After that decreasing = TRUE parameter was used to sort the result in decreasing order. Then the result was stored in DEATH_EVENTMode. cat method was used to print the mode (most frequent value) and lastly all the NA values of DEATH_EVENT attribute were replaced by DEATH_EVENTMode.

Replace by Average Value

❖ Replacing NA values for age using Mean

Code:

```
missingValues <- which(is.na(dataset$age))
ageMean <- floor(mean(dataset$age, na.rm = TRUE))
dataset$age[missingValues] <- ageMean
cat("Age Mean:", ageMean)
dataset$age
```

Output:

```
> missingValues <- which(is.na(dataset$age))
> ageMean <- floor(mean(dataset$age, na.rm = TRUE))
> dataset$age[missingValues] <- ageMean
> cat("Age Mean:", ageMean)
Age Mean: 63> dataset$age
  [1]  75.00  55.00  65.00  50.00  65.00  90.00  75.00  60.00  65.00  63.00  75.00  62.00  45.00  50.00  49.00  82.00  87.00  45.00  70.00  48.00  65.00  65.00  68.00
 [24]  53.00  75.00  80.00  95.00  70.00  63.00  82.00  94.00  85.00  50.00  50.00  65.00  63.00  90.00  82.00  60.00  60.00  70.00  50.00  70.00  72.00  60.00  50.00
 [47]  51.00  60.00  80.00  63.00  68.00  53.00  60.00  70.00  60.00  95.00  70.00  60.00  49.00  72.00  45.00  50.00  55.00  45.00  45.00  60.00  63.00  72.00  70.00
 [70]  65.00  41.00  58.00  85.00  65.00  69.00  60.00  70.00  42.00  75.00  55.00  70.00  67.00  60.00  79.00  59.00  51.00  55.00  65.00  44.00  57.00  70.00  60.00
 [93]  42.00  60.00  58.00  58.00  63.00  70.00  60.00  63.00  65.00  75.00  80.00  42.00  60.00  72.00  55.00  45.00  63.00  45.00  85.00  55.00  50.00  70.00  60.00
[116]  58.00  60.00  85.00  65.00  86.00  60.00  66.00  60.00  60.00  60.00  43.00  46.00  58.00  61.00  53.00  53.00  60.00  46.00  63.00  81.00  75.00  65.00  68.00
[139]  62.00  50.00  80.00  46.00  50.00 161.00  72.00  50.00  52.00  64.00  75.00  60.00  72.00  62.00  50.00  50.00  65.00  60.00  52.00  50.00  85.00  59.00  66.00
[162]  45.00  63.00  50.00  45.00  80.00  53.00  59.00  65.00  70.00  51.00  52.00  70.00  50.00  65.00  60.00  69.00  49.00  63.00  55.00  40.00  59.00  65.00  75.00
[185]  58.00 170.00  50.00  60.00  60.66  40.00  80.00  64.00  50.00  73.00  45.00  77.00  45.00  65.00  50.00
>
```

First, the NA values of the age attribute were stored in missingValues using which(is.na(dataset$age)). Then, the mean value of the age attribute was stored in a variable ageMean using the mean method. na.rm = TRUE is a parameter to ignore the NA values while calculating Mean. Floor was used to convert the result to an integer value. cat method was used to print the mean value and finally all the NA values in the age attribute were replaced by ageMean.

❖ **Replacing NA values for ejection_fraction using Mean**

Code:

```
missingValues <- which(is.na(dataset$ejection_fraction))
ejection_fractionMean <- floor(mean(dataset$ejection_fraction, na.rm = TRUE))
dataset$ejection_fraction[missingValues] <- ejection_fractionMean
cat("Ejection_fraction Mean:", ejection_fractionMean)
dataset$ejection_fraction
```

Output:

```
> missingValues <- which(is.na(dataset$ejection_fraction))
> ejection_fractionMean <- floor(mean(dataset$ejection_fraction, na.rm = TRUE))
> dataset$ejection_fraction[missingValues] <- ejection_fractionMean
> cat("Ejection_fraction Mean:", ejection_fractionMean)
Ejection_fraction Mean: 37> dataset$ejection_fraction
  [1] 20 38 20 20 20 40 15 60 65 35 38 25 30 38 30 50 38 14 25 55 25 30 35 60 30 38 40 45 38 30 38 45 35 30 50 35 50 50 30 38 20 30 45 50 60 38 25 38 20 30 25 20 62
 [54] 50 38 30 35 40 20 20 25 40 35 35 80 20 15 25 25 25 40 35 35 50 20 20 60 40 38 45 40 50 25 50 25 50 35 60 40 25 45 45 60 25 38 60 25 60 25 40 25 45 25 30 50 30
[107] 45 35 38 35 60 35 25 60 40 40 60 60 60 38 60 38 38 30 40 50 17 60 30 35 60 45 40 60 35 40 60 25 35 30 38 35 30 40 25 30 30 60 30 35 45 60 45 35 35 25 35 25 50
[160] 45 40 35 40 35 30 38 60 20 40 35 35 40 60 20 35 60 40 50 60 40 30 25 25 38 25 30 50 25 40 45 35 60 40 30 20 45 38 30 20
>
```

First, the NA values of the ejection_fraction attribute were stored in missingValues using which(is.na(ejection_fraction)). Then, the mean value of the ejection_fraction attribute was stored in a variable ejection_fractionMean using the mean method. na.rm = TRUE is a parameter to ignore the NA values while calculating Mean. Floor was used to convert the result to an integer value. cat method was used to print the mean value and finally all the NA values in the ejection_fraction attribute were replaced by ejection_fractionMean.

❖ **Replacing NA values for platelets using Mean**

Code:

```
missingValues <- which(is.na(dataset$platelets))
plateletsMean <- floor(mean(dataset$platelets, na.rm = TRUE))
dataset$platelets[missingValues] <- plateletsMean
cat("Platelets Mean:", plateletsMean)
dataset$platelets
```

Output:

```
> missingValues <- which(is.na(dataset$platelets))
> plateletsMean <- floor(mean(dataset$platelets, na.rm = TRUE))
> dataset$platelets[missingValues] <- plateletsMean
> cat("Platelets Mean:", plateletsMean)
Platelets Mean: 262560> dataset$platelets
  [1] 265000 263358 162000 210000 327000 204000 127000 454000 263358 388000 368000 262560 136000 276000 427000  47000 262000 166000 237000  87000 276000 297000 289000
 [24] 368000 263358 149000 262560 284000 153000 200000 263358 360000 319000 302000 188000 228000 226000 321000 305000 329000 263358 153000 185000 218000 262560 310000
 [47] 271000 451000 140000 395000 166000 418000 263358 351000 255000 461000 223000 216000 319000 254000 390000 216000 254000 385000 263358 119000 213000 274000 244000
 [70] 497000 374000 122000 243000 149000 262560 204000 317000 237000 283000 324000 293000 263358 196000 172000 302000 406000 173000 304000 235000 181000 249000 297000
 [93] 263358 210000 327000 219000 254000 255000 318000 262560 298000 263358 149000 226000 286000 621000 263000 226000 304000 850000 306000 228000 252000 351000 328000
[116] 164000 271000 507000 203000 263358 210000 162000 262560 127000 217000 237000 271000 300000 267000 227000 249000 250000 263358 295000 231000 263358 172000 305000
[139] 221000 211000 263358 348000 329000 229000 338000 266000 218000 242000 225000 228000 235000 244000 184000 263358 235000 194000 277000 262000 235000 362000 242000
[162] 174000 448000  75000 334000 192000 220000  70000 270000 305000 263358 325000 176000 189000 281000 337000 105000 132000 267000 279000 303000 221000 265000 224000
[185] 219000 389000 153000 365000 201000 275000 350000 309000 260000 160000 126000 223000 263358 259000 279000
>
```

First, the NA values of the platelets attribute were stored in missingValues using which(is.na(platelets)). Then, the mean value of the platelets attribute was stored in a variable plateletsMean using the mean method. na.rm = TRUE is a parameter to ignore the NA values while calculating Mean. Floor was used to convert the result to an integer value. cat method was used to print the mean value and finally all the NA values in the platelets attribute were replaced by plateletsMean.

❖ **Replacing NA values for serum_creatinine using Mean**

Code:

```
missingValues <- which(is.na(dataset$serum_creatinine))
serum_creatinineMean <- floor(mean(dataset$serum_creatinine, na.rm = TRUE))
dataset$serum_creatinine[missingValues] <- serum_creatinineMean
cat("Serum_creatinine:", serum_creatinineMean)
dataset$serum_creatinine
```

Output:

```
> missingValues <- which(is.na(dataset$serum_creatinine))
> serum_creatinineMean <- floor(mean(dataset$serum_creatinine, na.rm = TRUE))
> dataset$serum_creatinine[missingValues] <- serum_creatinineMean
> cat("Serum_creatinine:", serum_creatinineMean)
Serum_creatinine: 1> dataset$serum_creatinine
  [1] 1.90 1.10 1.30 1.90 2.70 2.10 1.20 1.10 1.50 9.40 4.00 0.90 1.10 1.10 1.00 1.30 0.90 0.80 1.00 1.90 1.30 1.60 0.90 0.80 1.83 1.90 1.00 1.30 5.80 1.20 1.83 3.00
 [33] 1.00 1.20 1.00 3.50 1.00 1.00 2.30 3.00 1.83 1.20 1.20 1.00 1.10 1.90 0.90 0.60 4.40 1.00 1.00 1.40 6.80 1.00 2.20 2.00 2.70 0.60 1.10 1.30 1.00 2.30 1.10 1.00
 [65] 1.18 2.90 1.30 1.00 1.20 1.83 0.80 0.90 1.00 1.30 1.20 0.70 0.80 1.20 0.60 0.90 1.70 1.18 2.50 1.80 1.00 0.70 1.10 0.80 0.70 1.10 0.80 1.00 1.18 1.70 0.70 1.00
 [97] 1.30 1.10 1.20 1.10 1.10 1.18 1.10 1.00 2.30 1.70 1.30 0.90 1.10 1.30 1.20 1.20 1.60 1.30 1.20 1.00 0.70 3.20 0.90 1.83 1.50 1.00 0.75 0.90 3.70 1.30 2.10 0.80
[129] 0.70 3.40 0.70 6.10 1.18 1.30 1.18 1.18 0.90 2.10 1.00 0.80 1.10 0.90 0.90 0.90 1.70 0.70 0.70 1.00 1.83 0.90 2.50 0.90 0.90 1.18 0.80 1.70 1.40 1.00 1.30 1.10
[161] 1.20 0.80 0.90 0.90 1.10 1.30 0.70 2.40 1.00 0.80 1.50 0.90 1.10 0.80 0.90 1.00 1.00 1.00 1.20 0.70 0.90 1.00 1.20 2.50 1.20 1.50 0.60 2.10 1.00 0.90 2.10 1.50
[193] 0.70 1.18 1.60 1.80 1.18 0.80 1.00
>
```

First, the NA values of the serum_creatinine attribute were stored in missingValues using which(is.na(serum_creatinine)). Then, the mean value of the serum_creatinine attribute was stored in a variable serum_creatinineMean using the mean method. na.rm = TRUE is a parameter to ignore the NA values while calculating Mean. Floor was used to convert the result to an integer value. cat method was used to print the mean value and finally all the NA values in the serum_creatinine attribute were replaced by serum_creatinineMean.

❖ **Replacing NA values for serum_sodium using Mean**

Code:

```
missingValues <- which(is.na(dataset$serum_sodium))
serum_sodiumMean <- floor(mean(dataset$serum_sodium, na.rm = TRUE))
dataset$serum_sodium[missingValues] <- serum_sodiumMean
cat("Serum_sodium:", serum_sodiumMean)
dataset$serum_sodium
```

Output:

```
> missingValues <- which(is.na(dataset$serum_sodium))
> serum_sodiumMean <- floor(mean(dataset$serum_sodium, na.rm = TRUE))
> dataset$serum_sodium[missingValues] <- serum_sodiumMean
> cat("Serum_sodium:", serum_sodiumMean)
Serum_sodium: 136> dataset$serum_sodium
  [1] 130 136 129 137 116 132 137 131 138 133 131 140 137 137 138 136 140 127 140 121 137 136 140 135 134 144 138 136 134 132 134 132 128 138 140 134 134 145 137 142
 [41] 134 136 139 134 142 135 130 138 133 140 138 139 146 134 132 132 138 138 136 136 139 131 139 145 137 127 136 140 142 135 140 139 132 137 134 139 140 140 131 140
 [81] 136 137 132 133 141 140 137 140 139 144 136 133 137 135 142 141 134 136 137 140 141 137 144 140 143 138 137 138 133 142 132 135 136 137 126 139 136 138 140 134
[121] 135 136 140 145 134 135 124 137 136 145 138 131 137 145 137 137 137 130 136 138 134 140 132 141 139 141 136 137 134 136 135 139 134 137 136 140 136 136 134 139
[161] 134 139 137 142 139 135 133 134 138 133 136 140 145 139 137 138 135 140 145 140 136 136 136 134 137 136 134 144 136 140 134 135 130 142 135 145 137 138 134
>
```

First, the NA values of the serum_sodium attribute were stored in missingValues using which(is.na(serum_sodium)). Then, the mean value of the serum_sodium attribute was stored in a variable serum_sodiumMean using the mean method. na.rm = TRUE is a parameter to ignore the NA values while calculating Mean. Floor was used to convert the result to an integer value. cat method was used to print the mean value and finally all the NA values in the serum_sodium attribute were replaced by serum_sodiumMean.

❖ **Replacing NA values for time using Mean**

Code:

```
missingValues <- which(is.na(dataset$time))
timeMean <- floor(mean(dataset$time, na.rm = TRUE))
dataset$time[missingValues] <- timeMean
cat("Time:", timeMean)
dataset$time
```

Output:

```
> missingValues <- which(is.na(dataset$time))
> timeMean <- floor(mean(dataset$time, na.rm = TRUE))
> dataset$time[missingValues] <- timeMean
> cat("Time:", timeMean)
Time: 84> dataset$time
  [1]   4   6   7   7   8   8  10  10  10  10  10  10  11  11  12  13  14  14  15  15  16  20  20  22  23  23  24  26  26  26  27  28  28  29  29  30  30  30  30  30
 [41]  31  32  33  33  33  35  38  40  41  42  43  43  43  44  45  50  54  54  55  59  60  60  60  61  63  64  65  65  66  67  68  71  72  72  73  73  74  74  74  74
 [81]  75  76  77  78  78  79  79  79  79  79  80  80  82  82  83  83  83  85  85  86  87  87  87  87  87  88  88  88  88  88  90  90  90  90  91  91  94  94  94  95
[121]  95  95  95  95  96  97 100 104 104 105 106 107 107 107 107 107 107 108 108 108 109 109 109 110 111 112 112 113 113 115 115 117 118 119 120 120 120 120 121 121
[161] 121 121 123 126 129 130 134 135 140 145 145 146 146 146 146 147 147 147 147 148 150 154 162 170 171 172 172 172 174 174 174 175 180 180 180 185 186 186
>
```

First, the NA values of the time attribute were stored in missingValues using which(is.na(time)). Then, the mean value of the time attribute was stored in a variable timeMean using the mean method. na.rm = TRUE is a parameter to ignore the NA values while calculating Mean. Floor was used to convert the result to an integer value. cat method was used to print the mean value and finally all the NA values in the time attribute were replaced by timeMean.

Replace by Median Value

❖ Replacing NA values for age using Median

Code:

```
missingValues <- which(is.na(dataset$age))
ageMedian <- floor(median(dataset$age, na.rm = TRUE))
dataset$age[missingValues] <- ageMedian
cat("Age Median:", ageMedian)
dataset$age
```

Output:

```
> missingValues <- which(is.na(dataset$age))
> ageMedian <- floor(median(dataset$age, na.rm = TRUE))
> dataset$age[missingValues] <- ageMedian
> cat("Age Median:", ageMedian)
Age Median: 60> dataset$age
  [1]  75.00  55.00  65.00  50.00  65.00  90.00  75.00  60.00  65.00  63.00  75.00  62.00  45.00  50.00  49.00  82.00  87.00  45.00  70.00  48.00  65.00  65.00  68.00
 [24]  53.00  75.00  80.00  95.00  70.00  63.00  82.00  94.00  85.00  50.00  50.00  65.00  63.00  90.00  82.00  60.00  60.00  70.00  50.00  70.00  72.00  60.00  50.00
 [47]  51.00  60.00  80.00  63.00  68.00  53.00  60.00  70.00  60.00  95.00  70.00  60.00  49.00  72.00  45.00  50.00  55.00  45.00  45.00  60.00  63.00  72.00  70.00
 [70]  65.00  41.00  58.00  85.00  65.00  69.00  60.00  70.00  42.00  75.00  55.00  70.00  67.00  60.00  79.00  59.00  51.00  55.00  65.00  44.00  57.00  70.00  60.00
 [93]  42.00  60.00  58.00  58.00  63.00  70.00  60.00  63.00  65.00  75.00  80.00  42.00  60.00  72.00  55.00  45.00  63.00  45.00  85.00  55.00  50.00  70.00  60.00
[116]  58.00  60.00  85.00  65.00  86.00  60.00  66.00  60.00  60.00  60.00  43.00  46.00  58.00  61.00  53.00  53.00  60.00  46.00  63.00  81.00  75.00  65.00  68.00
[139]  62.00  50.00  80.00  46.00  50.00 161.00  72.00  50.00  52.00  64.00  75.00  60.00  72.00  62.00  50.00  50.00  65.00  60.00  52.00  50.00  85.00  59.00  66.00
[162]  45.00  63.00  50.00  45.00  80.00  53.00  59.00  65.00  70.00  51.00  52.00  70.00  50.00  65.00  60.00  69.00  49.00  63.00  55.00  40.00  59.00  65.00  75.00
[185]  58.00 170.00  50.00  60.00  60.66  40.00  80.00  64.00  50.00  73.00  45.00  77.00  45.00  65.00  50.00
>
```

First, the NA values of the age attribute were stored in missingValues using which(is.na(age)). Then, the median value of the age attribute was stored in a variable ageMedian using the median method. na.rm = TRUE is a parameter to ignore the NA values while calculating Median. Floor was used to convert the result to an integer value. cat method was used to print the median value and finally all the NA values in the age attribute were replaced by ageMedian.

❖ **Replacing NA values for ejection_fraction using Median**

Code:

```
missingValues <- which(is.na(dataset$ejection_fraction))
ejection_fractionMedian <- floor(median(dataset$ejection_fraction, na.rm = TRUE))
dataset$ejection_fraction[missingValues] <- ejection_fractionMedian
cat("Ejection_fraction:", ejection_fractionMedian)
dataset$ejection_fraction
```

Output:

```
> missingValues <- which(is.na(dataset$ejection_fraction))
> ejection_fractionMedian <- floor(median(dataset$ejection_fraction, na.rm = TRUE))
> dataset$ejection_fraction[missingValues] <- ejection_fractionMedian
> cat("Ejection_fraction:", ejection_fractionMedian)
Ejection_fraction: 38> dataset$ejection_fraction
  [1] 20 38 20 20 20 40 15 60 65 35 38 25 30 38 30 50 38 14 25 55 25 30 35 60 30 38 40 45 38 30 38 45 35 30 50 35 50 50 30 38 20 30 45 50 60 38 25 38 20 30 25 20 62
 [54] 50 38 30 35 40 20 20 25 40 35 35 80 20 15 25 25 25 40 35 35 50 20 20 60 40 38 45 40 50 25 50 25 50 35 60 40 25 45 45 60 25 38 60 25 60 25 40 25 45 25 30 50 30
[107] 45 35 38 35 60 35 25 60 40 40 60 60 60 38 60 38 38 30 40 50 17 60 30 35 60 45 40 60 35 40 60 25 35 30 38 35 30 40 25 30 30 60 30 35 45 60 45 35 35 25 35 25 50
[160] 45 40 35 40 35 30 38 60 20 40 35 35 40 60 20 35 60 40 50 60 40 30 25 25 38 25 30 50 25 40 45 35 60 40 30 20 45 38 30 20
>
```

First, the NA values of the ejection_fraction attribute were stored in missingValues using which(is.na(ejection_fraction)). Then, the median value of the ejection_fraction attribute was stored in a variable ejection_fraction Median using the median method. na.rm = TRUE is a parameter to ignore the NA values while calculating Median. Floor was used to convert the result to an integer value. cat method was used to print the median value and finally all the NA values in the ejection_fraction attribute were replaced by age ejection_fractionMedian.

❖ **Replacing NA values for platelets using Median**

Code:

```
missingValues <- which(is.na(dataset$platelets))
plateletsMedian <- floor(median(dataset$platelets, na.rm = TRUE))
dataset$platelets[missingValues] <- plateletsMedian
cat("Platelets:", plateletsMedian)
dataset$platelets
```

Output:

```
> missingValues <- which(is.na(dataset$platelets))
> plateletsMedian <- floor(median(dataset$platelets, na.rm = TRUE))
> dataset$platelets[missingValues] <- plateletsMedian
> cat("Platelets:", plateletsMedian)
Platelets: 262560> dataset$platelets
  [1] 265000 263358 162000 210000 327000 204000 127000 454000 263358 388000 368000 262560 136000 276000 427000  47000 262000 166000 237000  87000 276000 297000 289000
 [24] 368000 263358 149000 262560 284000 153000 200000 263358 360000 319000 302000 188000 228000 226000 321000 305000 329000 263358 153000 185000 218000 262560 310000
 [47] 271000 451000 140000 395000 166000 418000 263358 351000 255000 461000 223000 216000 319000 254000 390000 216000 254000 385000 263358 119000 213000 274000 244000
 [70] 497000 374000 122000 243000 149000 262560 204000 317000 237000 283000 324000 293000 263358 196000 172000 302000 406000 173000 304000 235000 181000 249000 297000
 [93] 263358 210000 327000 219000 254000 255000 318000 262560 298000 263358 149000 226000 286000 621000 263000 226000 304000 850000 306000 228000 252000 351000 328000
[116] 164000 271000 507000 203000 263358 210000 162000 262560 127000 217000 237000 271000 300000 267000 227000 249000 250000 263358 295000 231000 263358 172000 305000
[139] 221000 211000 263358 348000 329000 229000 338000 266000 218000 242000 225000 228000 235000 244000 184000 263358 235000 194000 277000 262000 235000 362000 242000
[162] 174000 448000  75000 334000 192000 220000  70000 270000 305000 263358 325000 176000 189000 281000 337000 105000 132000 267000 279000 303000 221000 265000 224000
[185] 219000 389000 153000 365000 201000 275000 350000 309000 260000 160000 126000 223000 263358 259000 279000
>
```

First, the NA values of the platelets attribute were stored in missingValues using which(is.na(platelets)). Then, the median value of the platelets attribute was stored in a variable plateletsMedian using the median method. na.rm = TRUE is a parameter to ignore the NA values while calculating Median. Floor was used to convert the result to an integer value. cat method was used to print the median value and finally all the NA values in the platelets attribute were replaced by plateletsMedian.

❖ **Replacing NA values for serum_creatinine using Median**

Code:

```
missingValues <- which(is.na(dataset$serum_creatinine))
serum_creatinineMedian <- floor(median(dataset$serum_creatinine, na.rm = TRUE))
dataset$serum_creatinine[missingValues] <- serum_creatinineMedian
cat("Serum_creatinine:", serum_creatinineMedian)
dataset$serum_creatinine
```

Output:

```
> missingValues <- which(is.na(dataset$serum_creatinine))
> serum_creatinineMedian <- floor(median(dataset$serum_creatinine, na.rm = TRUE))
> dataset$serum_creatinine[missingValues] <- serum_creatinineMedian
> cat("Serum_creatinine:", serum_creatinineMedian)
Serum_creatinine: 1> dataset$serum_creatinine
  [1] 1.90 1.10 1.30 1.90 2.70 2.10 1.20 1.10 1.50 9.40 4.00 0.90 1.10 1.10 1.00 1.30 0.90 0.80 1.00 1.90 1.30 1.60 0.90 0.80 1.83 1.90 1.00 1.30 5.80 1.20 1.83 3.00
 [33] 1.00 1.20 1.00 3.50 1.00 1.00 2.30 3.00 1.83 1.20 1.20 1.00 1.10 1.90 0.90 0.60 4.40 1.00 1.00 1.40 6.80 1.00 2.20 2.00 2.70 0.60 1.10 1.30 1.00 2.30 1.10 1.00
 [65] 1.18 2.90 1.30 1.00 1.20 1.83 0.80 0.90 1.00 1.30 1.20 0.70 0.80 1.20 0.60 0.90 1.70 1.18 2.50 1.80 1.00 0.70 1.10 0.80 0.70 1.10 0.80 1.00 1.18 1.70 0.70 1.00
 [97] 1.30 1.10 1.20 1.10 1.10 1.18 1.10 1.00 2.30 1.70 1.30 0.90 1.10 1.30 1.20 1.20 1.60 1.30 1.20 1.00 0.70 3.20 0.90 1.83 1.50 1.00 0.75 0.90 3.70 1.30 2.10 0.80
[129] 0.70 3.40 0.70 6.10 1.18 1.30 1.18 1.18 0.90 2.10 1.00 0.80 1.10 0.90 0.90 0.90 1.70 0.70 0.70 1.00 1.83 0.90 2.50 0.90 0.90 1.18 0.80 1.70 1.40 1.00 1.30 1.10
[161] 1.20 0.80 0.90 0.90 1.10 1.30 0.70 2.40 1.00 0.80 1.50 0.90 1.10 0.80 0.90 1.00 1.00 1.00 1.20 0.70 0.90 1.00 1.20 2.50 1.20 1.50 0.60 2.10 1.00 0.90 2.10 1.50
[193] 0.70 1.18 1.60 1.80 1.18 0.80 1.00
>
```

First, the NA values of the serum_creatinine attribute were stored in missingValues using which(is.na(serum_creatinine)). Then, the median value of the serum_creatinine attribute was stored in a variable serum_creatinineMedian using the median method. na.rm = TRUE is a parameter to ignore the NA values while calculating Median. Floor was used to convert the result to an integer value. cat method was used to print the median value and finally all the NA values in the serum_creatinine attribute were replaced by serum_creatinineMedian.

❖ **Replacing NA values for serum_sodium using Median**

Code:

```
missingValues <- which(is.na(dataset$serum_sodium))
serum_sodiumMedian <- floor(median(dataset$serum_sodium, na.rm = TRUE))
dataset$serum_sodium[missingValues] <- serum_sodiumMedian
cat("Serum_sodium:", serum_sodiumMedian)
dataset$serum_sodium
```

Output:

```
> missingValues <- which(is.na(dataset$serum_sodium))
> serum_sodiumMedian <- floor(median(dataset$serum_sodium, na.rm = TRUE))
> dataset$serum_sodium[missingValues] <- serum_sodiumMedian
> cat("Serum_sodium:", serum_sodiumMedian)
Serum_sodium: 137> dataset$serum_sodium
  [1] 130 136 129 137 116 132 137 131 138 133 131 140 137 137 138 136 140 127 140 121 137 136 140 135 134 144 138 136 134 132 134 132 128 138 140 134 134 145 137 142
 [41] 134 136 139 134 142 135 130 138 133 140 138 139 146 134 132 132 138 138 136 136 139 131 139 145 137 127 136 140 142 135 140 139 132 137 134 139 140 140 131 140
 [81] 136 137 132 133 141 140 137 140 139 144 136 133 137 135 142 141 134 136 137 140 141 137 144 140 143 138 137 138 133 142 132 135 136 137 126 139 136 138 140 134
[121] 135 136 140 145 134 135 124 137 136 145 138 131 137 145 137 137 137 130 136 138 134 140 132 141 139 141 136 137 134 136 135 139 134 137 136 140 136 136 134 139
[161] 134 139 137 142 139 135 133 134 138 133 136 140 145 139 137 138 135 140 145 140 136 136 136 134 137 136 134 144 136 140 134 135 130 142 135 145 137 138 134
>
```

First, the NA values of the serum_sodium attribute were stored in missingValues using which(is.na(serum_sodium)). Then, the median value of the serum_sodium attribute was stored in a variable serum_sodiumMedian using the median method. na.rm = TRUE is a parameter to ignore the NA values while calculating Median. Floor was used to convert the result to an integer value. cat method was used to print the median value and finally all the NA values in the serum_sodium attribute were replaced by serum_sodiumMedian.

❖ **Replacing NA values for time using Median**

Code:

```
missingValues <- which(is.na(dataset$time))
timeMedian <- floor(median(dataset$time, na.rm = TRUE))
dataset$time[missingValues] <- timeMedian
cat("Time:", timeMedian)
dataset$time
```

Output:

```
> missingValues <- which(is.na(dataset$time))
> timeMedian <- floor(median(dataset$time, na.rm = TRUE))
> dataset$time[missingValues] <- timeMedian
> cat("Time:", timeMedian)
Time: 86> dataset$time
  [1]   4   6   7   7   8   8  10  10  10  10  10  10  11  11  12  13  14  14  15  15  16  20  20  22  23  23  24  26  26  26  27  28  28  29  29  30  30  30  30  30
 [41]  31  32  33  33  33  35  38  40  41  42  43  43  43  44  45  50  54  54  55  59  60  60  60  61  63  64  65  65  66  67  68  71  72  72  73  73  74  74  74  74
 [81]  75  76  77  78  78  79  79  79  79  79  80  80  82  82  83  83  83  85  85  86  87  87  87  87  87  88  88  88  88  88  90  90  90  90  91  91  94  94  94  95
[121]  95  95  95  95  96  97 100 104 104 105 106 107 107 107 107 107 107 107 108 108 108 109 109 109 110 111 112 112 113 113 115 115 117 118 119 120 120 120 120 121 121
[161] 121 121 123 126 129 130 134 135 140 145 145 146 146 146 146 146 147 147 147 147 148 150 154 162 170 171 172 172 172 174 174 174 175 180 180 180 185 186 186
>
```

First, the NA values of the time attribute were stored in missingValues using which(is.na(time)). Then, the median value of the time attribute was stored in a variable timeMedian using the median method. na.rm = TRUE is a parameter to ignore the NA values while calculating Median. Floor was used to convert the result to an integer value. cat method was used to print the median value and finally all the NA values in the time attribute were replaced by timeMedian.

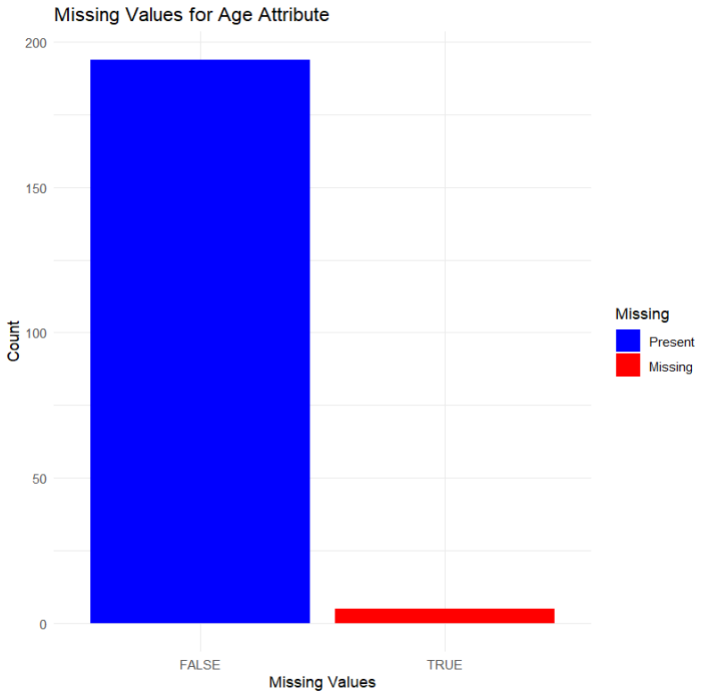
SHOW MISSING VALUES ON GRAPH

❖ Show Missing Values on Graph for age

Code:

```
dataset$age_missingvalue <- ifelse(is.na(dataset$age), TRUE, FALSE)

missing_counts <- table(dataset$age_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count =
as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) + geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Age Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



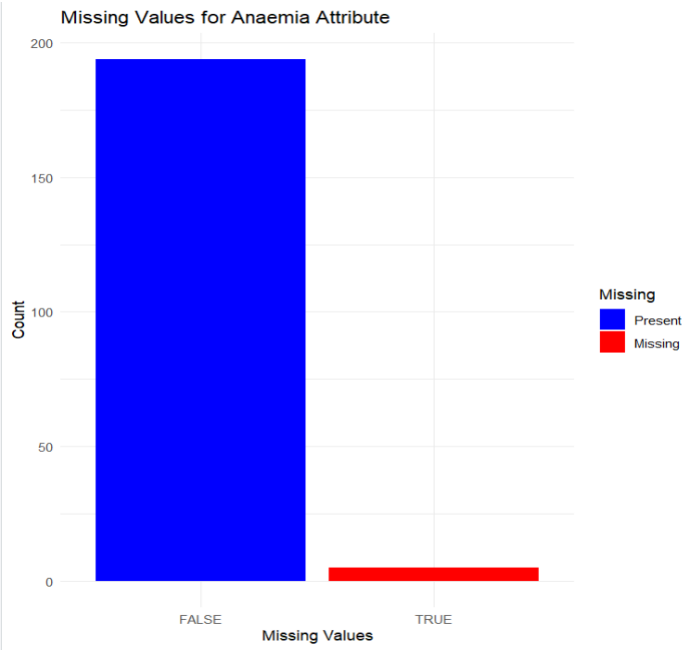
First, a binary indicator is created for missing age values using ifelse(is.na(dataset$age), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing age values is counted using table (dataset$age_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of age values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background.

❖ **Show Missing Values on Graph for anaemia**

Code:

```
dataset$anaemia_missingvalue <- ifelse(is.na(dataset$anaemia), TRUE, FALSE)

missing_counts <- table(dataset$anaemia_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count
=as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Anaemia Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



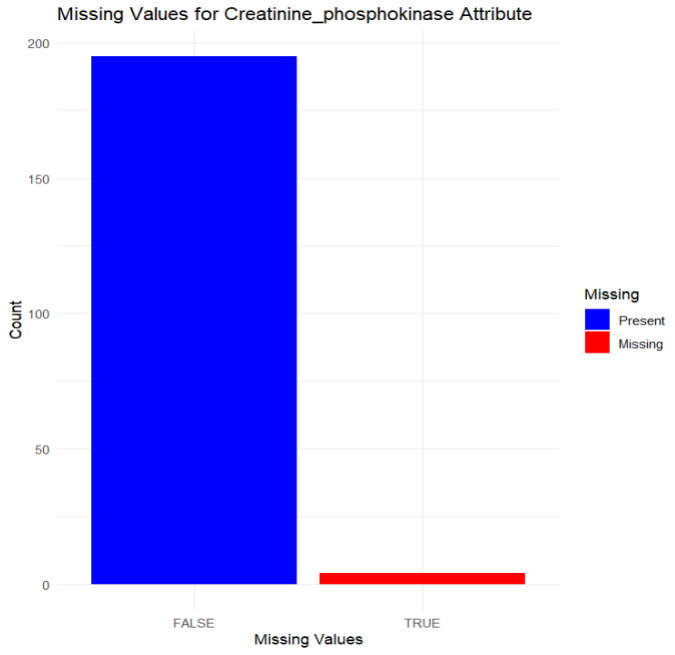
First, a binary indicator is created for missing anaemia values using ifelse(is.na(dataset$anaemia), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing anaemia values is counted using table (dataset$anaemia_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of anaemia values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background.

❖ **Show Missing Values on Graph for creatinine_phosphokinase**

Code:

```
dataset$creatinine_phosphokinase_missingvalue <-
ifelse(is.na(dataset$creatinine_phosphokinase), TRUE, FALSE)

missing_counts <- table(dataset$creatinine_phosphokinase_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count =
as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Creatinine_phosphokinase Attribute", x = "Missing Values", y =
"Count") +theme_minimal()
```

Output:



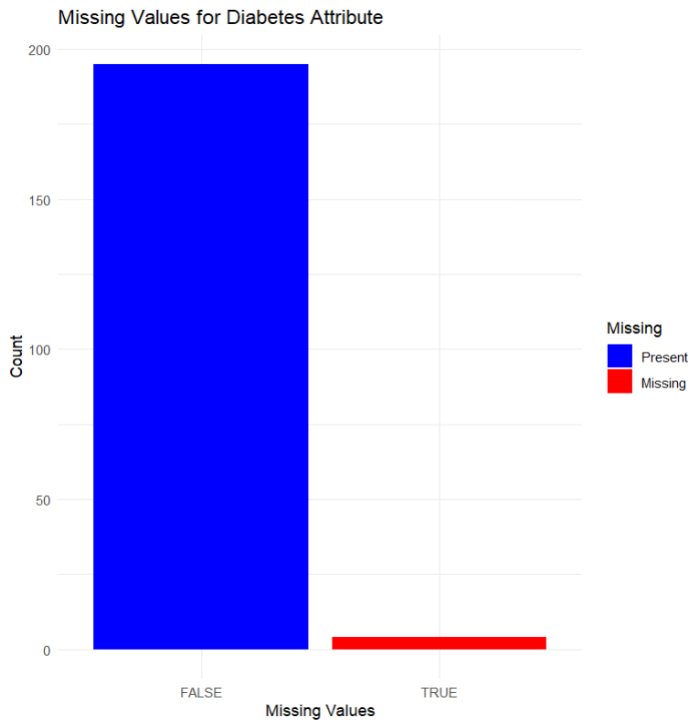
First, a binary indicator is created for missing creatinine_phosphokinase values using ifelse(is.na(dataset$creatinine_phosphokinase), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing creatinine_phosphokinase values is counted using table (dataset$creatinine_phosphokinase_ missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of creatinine_phosphokinase values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background.

❖ **Show Missing Values on Graph for diabetes**

Code:

```
dataset$diabetes_missingvalue <- ifelse(is.na(dataset$diabetes), TRUE, FALSE)

missing_counts <- table(dataset$diabetes_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count
=as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Diabetes Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



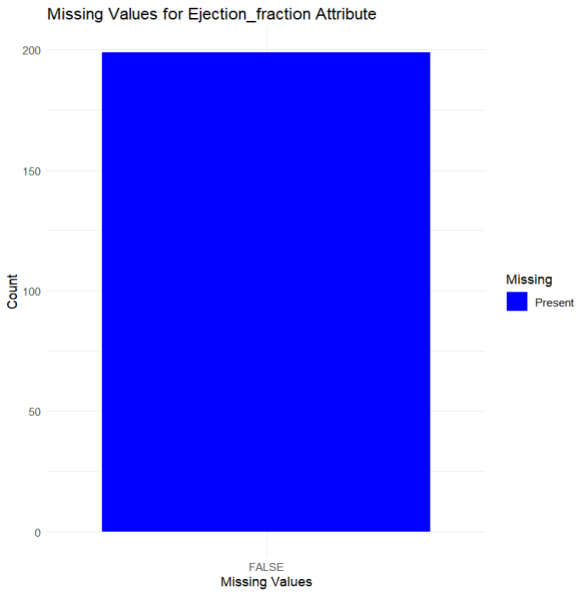
First, a binary indicator is created for missing diabetes values using ifelse(is.na(dataset$diabetes), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing diabetes values is counted using table (dataset$diabetes_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of diabetes values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background.

❖ **Show Missing Values on Graph for ejection_fraction**

Code:

```
dataset$ejection_fraction_missingvalue <- ifelse(is.na(dataset$ejection_fraction), TRUE, FALSE)

missing_counts <- table(dataset$ejection_fraction_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count =as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat = "identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing")) +labs(title = "Missing Values for Ejection_fraction Attribute", x = "Missing Values", y = "Count") +theme_minimal()
```

Output:



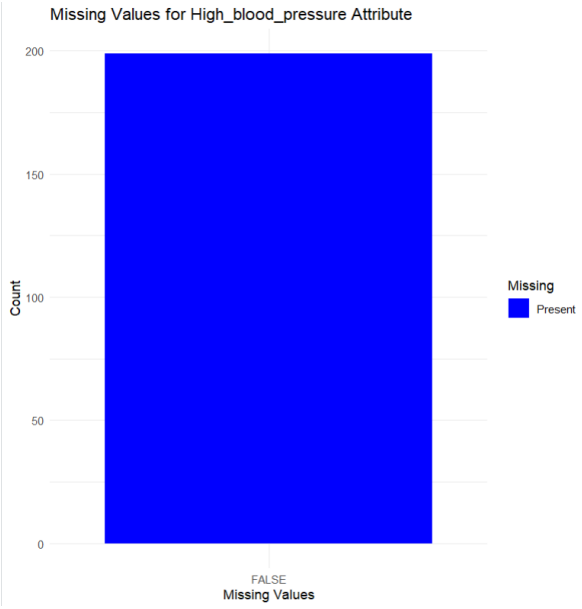
First, a binary indicator is created for missing ejection_fraction values using ifelse(is.na(dataset$ ejection_fraction), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing ejection_fraction values is counted using table (dataset$ejection_fraction _missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of ejection_fraction values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background. It can be seen from the graph that there is no missing values in ejection_fraction attribute.

❖ **Show Missing Values on Graph for high_blood_pressure**

Code:

```
dataset$high_blood_pressure_missingvalue <- ifelse(is.na(dataset$high_blood_pressure), TRUE, FALSE)

missing_counts <- table(dataset$high_blood_pressure_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count =as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat = "identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing")) +labs(title = "Missing Values for High_blood_pressure Attribute", x = "Missing Values", y ="Count") +theme_minimal()
```

Output:



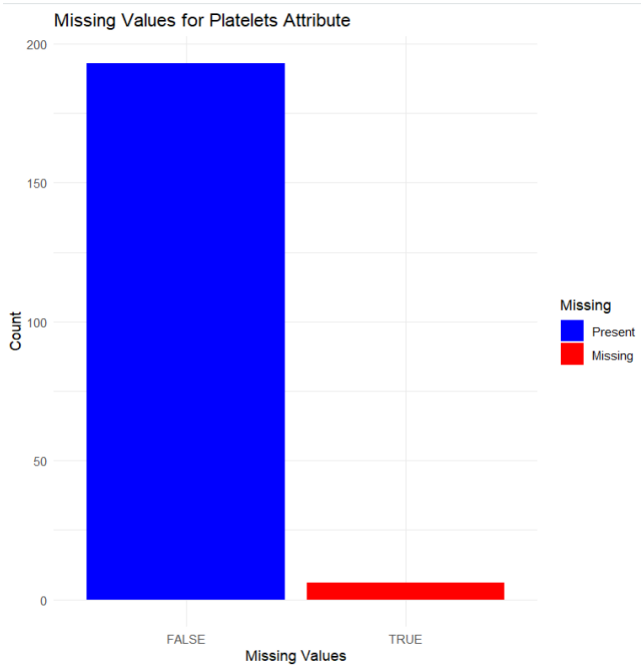
First, a binary indicator is created for missing high_blood_pressure values using ifelse(is.na(dataset$high_blood_pressure), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing high_blood_pressure values is counted using table (dataset$high_blood_pressure_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of high_blood_pressure values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background. It can be seen from the graph that there is no missing values in high_blood_pressure attribute.

❖ **Show Missing Values on Graph for platelets**

Code:

```
dataset$platelets_missingvalue <- ifelse(is.na(dataset$platelets), TRUE, FALSE)

missing_counts <- table(dataset$platelets_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count
=as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Platelets Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



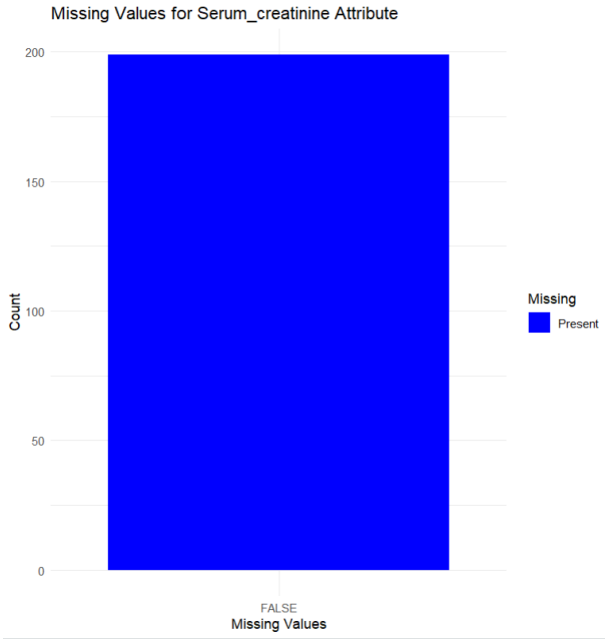
First, a binary indicator is created for missing platelets values using ifelse(is.na(dataset$ platelets), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing platelets values is counted using table (dataset$ platelets_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of platelets values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background.

❖ **Show Missing Values on Graph for serum_creatinine**

Code:

```
dataset$serum_creatinine_missingvalue <- ifelse(is.na(dataset$serum_creatinine), TRUE, FALSE)

missing_counts <- table(dataset$serum_creatinine_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count =as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat = "identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing")) +labs(title = "Missing Values for Serum_creatinine Attribute", x = "Missing Values", y ="Count") +theme_minimal()
```

Output:



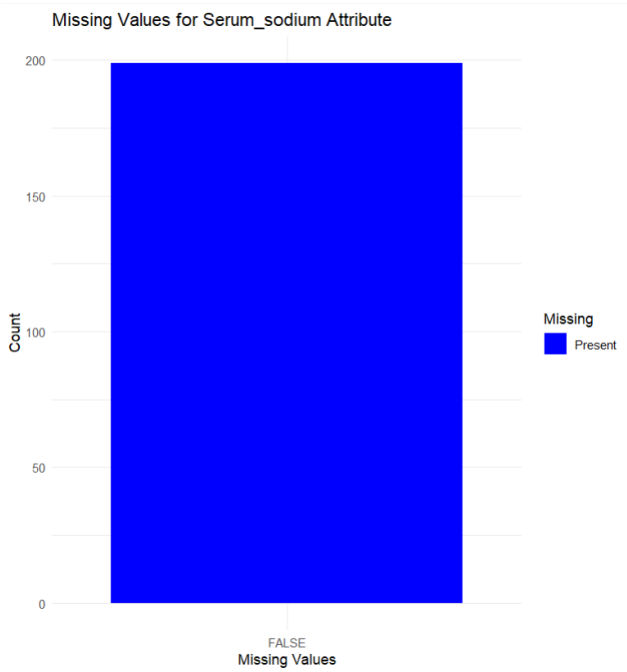
First, a binary indicator is created for missing serum_creatinine values using ifelse(is.na(dataset$ serum_creatinine), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing serum_creatinine values is counted using table (dataset$ serum_creatinine_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of serum_creatinine values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background. It can be seen from the graph that there is no missing values in serum_creatinine attribute.

❖ **Show Missing Values on Graph for serum_sodium**

Code:

```
dataset$serum_sodium_missingvalue <- ifelse(is.na(dataset$serum_sodium), TRUE, FALSE)

missing_counts <- table(dataset$serum_sodium_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count
=as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Serum_sodium Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



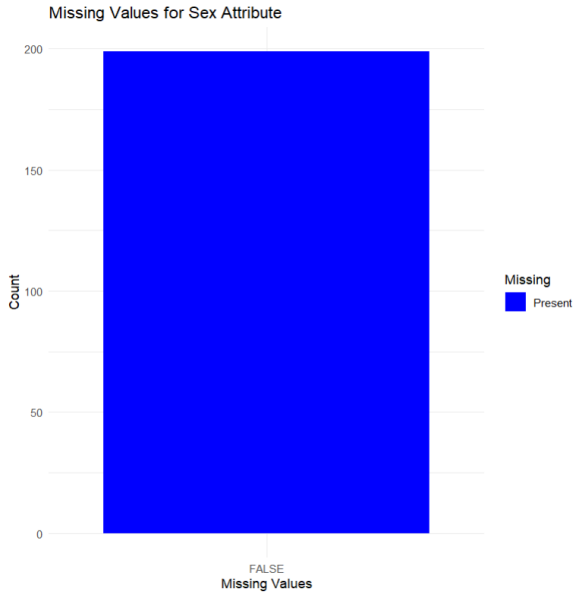
First, a binary indicator is created for missing serum_sodium values using ifelse(is.na(dataset$ serum_sodium), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing serum_sodium values is counted using table (dataset$serum_sodium_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of serum_sodium values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background. It can be seen from the graph that there is no missing values in serum_sodium attribute.

❖ **Show Missing Values on Graph for sex**

Code:

```
dataset$sex_missingvalue <- ifelse(is.na(dataset$sex), TRUE, FALSE)

missing_counts <- table(dataset$sex_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count
=as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Sex Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



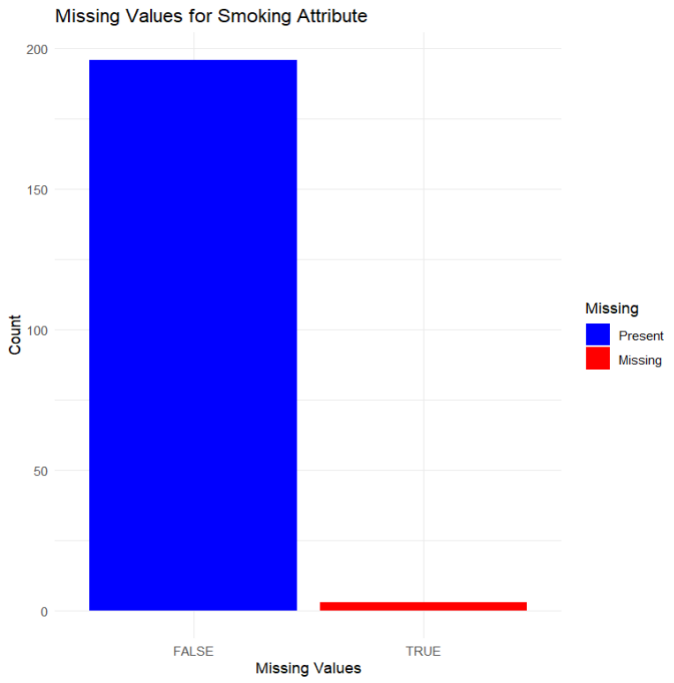
First, a binary indicator is created for missing sex values using ifelse(is.na(dataset$sex), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing sex values is counted using table (dataset$sex_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of sex values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background. It can be seen from the graph that there is no missing values in sex attribute.

❖ **Show Missing Values on Graph for smoking**

Code:

```
dataset$smoking_missingvalue <- ifelse(is.na(dataset$smoking), TRUE, FALSE)

missing_counts <- table(dataset$smoking_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count
=as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Smoking Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



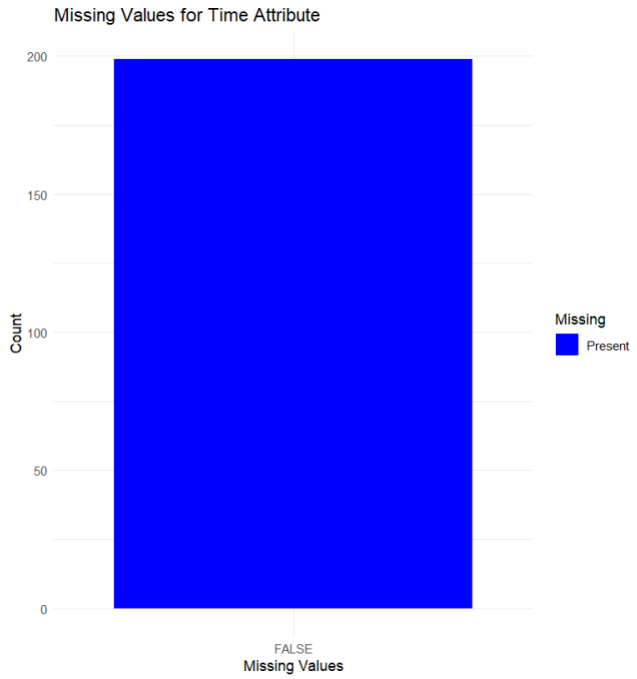
First, a binary indicator is created for missing smoking values using ifelse(is.na(dataset$smoking), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing smoking values is counted using table (dataset$smoking_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of smoking values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background.

❖ **Show Missing Values on Graph for time**

Code:

```
dataset$time_missingvalue <- ifelse(is.na(dataset$time), TRUE, FALSE)

missing_counts <- table(dataset$time_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count
=as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat =
"identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing"))
+labs(title = "Missing Values for Time Attribute", x = "Missing Values", y = "Count")
+theme_minimal()
```

Output:



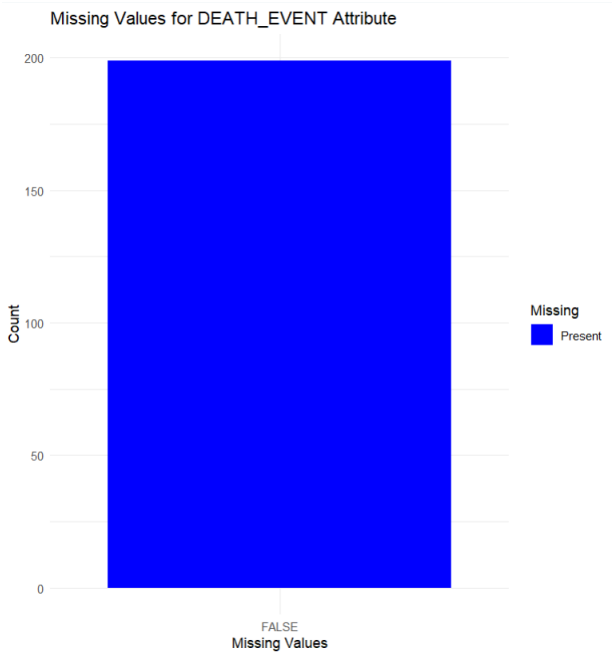
First, a binary indicator is created for missing time values using ifelse(is.na(dataset$time), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing time values is counted using table (dataset$time_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of time values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background. It can be seen from the graph that there is no missing values in time attribute.

❖ **Show Missing Values on Graph for DEATH_EVENT**

Code:

```
dataset$DEATH_EVENT_missingvalue <- ifelse(is.na(dataset$DEATH_EVENT), TRUE, FALSE)

missing_counts <- table(dataset$DEATH_EVENT_missingvalue)

ggplot(data = data.frame(Missing = names(missing_counts), Count =as.numeric(missing_counts)), aes(x = Missing, y = Count, fill = Missing)) +geom_bar(stat = "identity") +scale_fill_manual(values = c("blue", "red"), labels = c("Present", "Missing")) +labs(title = "Missing Values for DEATH_EVENT Attribute", x = "Missing Values", y ="Count") +theme_minimal()
```

Output:



First, a binary indicator is created for missing time values using ifelse(is.na(dataset$ DEATH_EVENT), TRUE, FALSE). TRUE means missing values and FALSE means present values. Then the number of missing and non-missing DEATH_EVENT values is counted using table (dataset$ DEATH_EVENT_missingvalue). Then a bar plot is created using ggplot, where the x-axis represents the presence or absence of DEATH_EVENT values (present or missing), and the y-axis represents the count of instances for each category. Here theme_minimal() is used to remove background gridline and grey background. It can be seen from the graph that there is no missing values in DEATH_EVENT attribute.

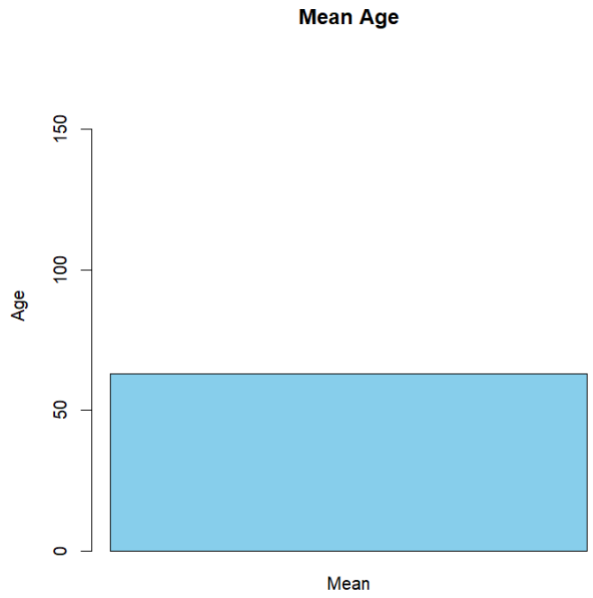
MEAN ON GRAPH

❖ **Show Mean for age on a Graph**

Code:

```
missingValues <- which(is.na(dataset$age))

ageMean <- floor(mean(dataset$age, na.rm = TRUE))

dataset$age[missingValues] <- ageMean

cat("Age Mean:", ageMean)

dataset$age

barplot(ageMean, main = "Mean Age", ylab = "Age", names.arg = "Mean", col = "skyblue", ylim = c(0, max(dataset$age) + 5))
```

Output:



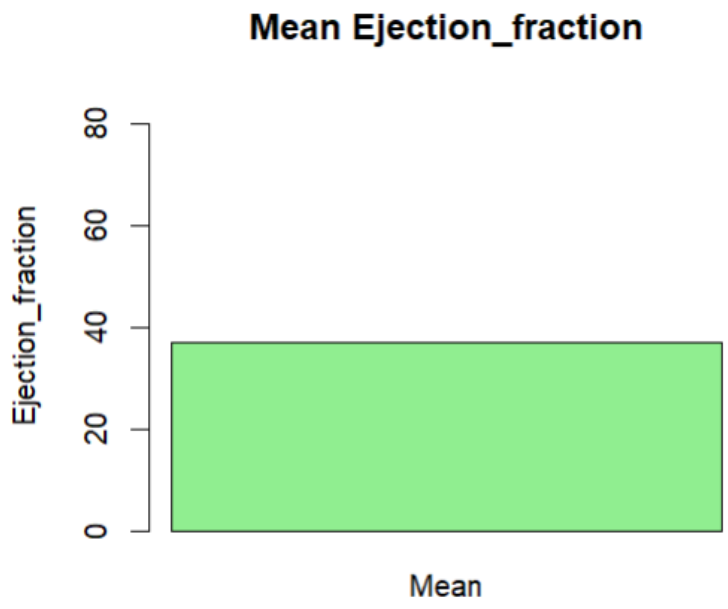
First, missing values of age attribute were replaced by ageMean. Then barplot is used to display the mean age value in the graph. ylim = c(0, max(dataset$age) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Mean for ejection_fraction on a Graph**

Code:

```
missingValues <- which(is.na(dataset$ejection_fraction))
ejection_fractionMean <- floor(mean(dataset$ejection_fraction, na.rm = TRUE))
dataset$ejection_fraction[missingValues] <- ejection_fractionMean
cat("Ejection_fraction Mean:", ejection_fractionMean)
dataset$ejection_fraction
barplot(ejection_fractionMean, main = "Mean Ejection_fraction", ylab = "Ejection_fraction", names.arg = "Mean", col = "lightgreen", ylim = c(0, max(dataset$ejection_fraction) + 5))
```

Output:



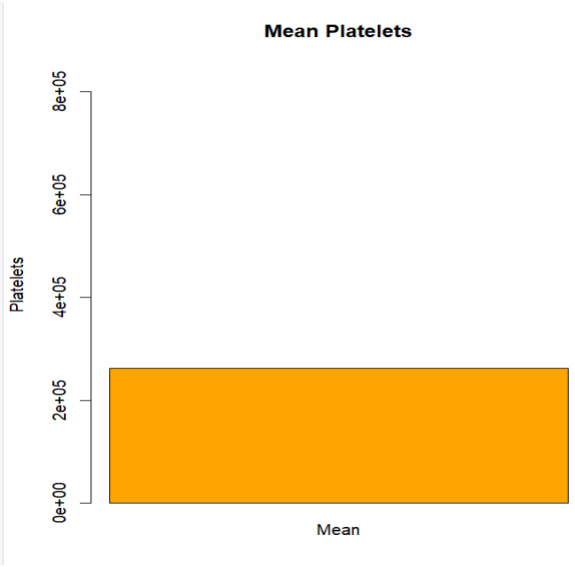
First, missing values of ejection_fraction attribute were replaced by ejection_fractionMean. Then barplot is used to display the mean ejection_fraction value in the graph. ylim = c(0, max(dataset$ ejection_fraction) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Mean for platelets on a Graph**

Code:

```
missingValues <- which(is.na(dataset$platelets))

plateletsMean <- floor(mean(dataset$platelets, na.rm = TRUE))

dataset$platelets[missingValues] <- plateletsMean

cat("Platelets Mean:", plateletsMean)

dataset$platelets

barplot(plateletsMean, main = "Mean Platelets", ylab = "Platelets", names.arg = "Mean", col = "orange", ylim = c(0, max(dataset$platelets) + 5))
```

Output:



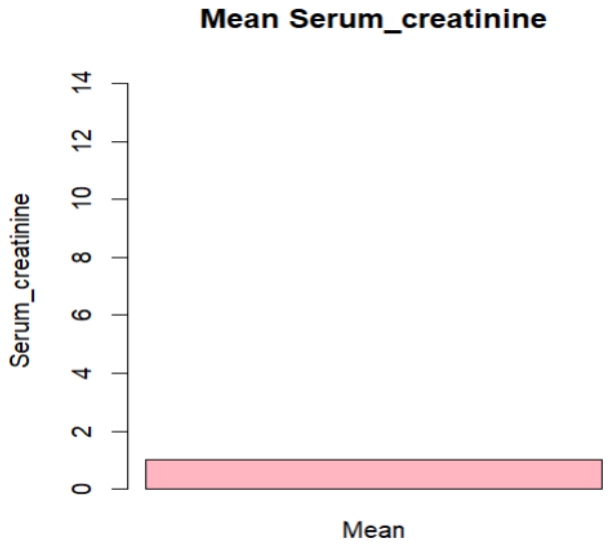
First, missing values of platelets attribute were replaced by plateletsMean. Then barplot is used to display the mean platelets value in the graph. ylim = c(0, max(dataset$platelets) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Mean for serum_creatinine on a Graph**

Code:

```
missingValues <- which(is.na(dataset$serum_creatinine))

serum_creatinineMean <- floor(mean(dataset$serum_creatinine, na.rm = TRUE))

dataset$serum_creatinine[missingValues] <- serum_creatinineMean

cat("Serum_creatinine Mean:", serum_creatinineMean)

dataset$serum_creatinine

barplot(serum_creatinineMean, main = "Mean Serum_creatinine", ylab = "Serum_creatinine",
names.arg = "Mean", col = "lightpink", ylim = c(0, max(dataset$serum_creatinine) + 5))
```

Output:



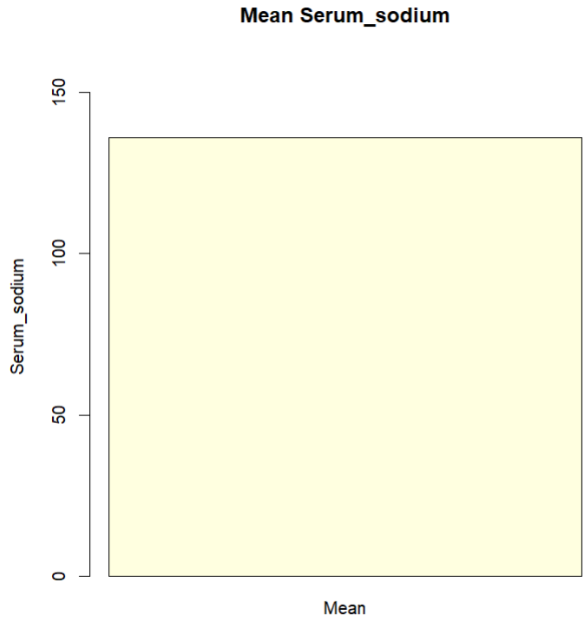
First, missing values of serum_creatinine attribute were replaced by serum_creatinineMean. Then barplot is used to display the mean serum_creatinine value in the graph. ylim = c(0, max(dataset$ serum_creatinine) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Mean for serum_sodium on a Graph**

Code:

```
missingValues <- which(is.na(dataset$serum_sodium))

serum_sodiumMean <- floor(mean(dataset$serum_sodium, na.rm = TRUE))

dataset$serum_sodium[missingValues] <- serum_sodiumMean

cat("Serum_sodium Mean:", serum_sodiumMean)

dataset$serum_sodium

barplot(serum_sodiumMean, main = "Mean Serum_sodium", ylab = "Serum_sodium", names.arg = "Mean", col = "lightyellow", ylim = c(0, max(dataset$serum_sodium) + 15))
```

Output:



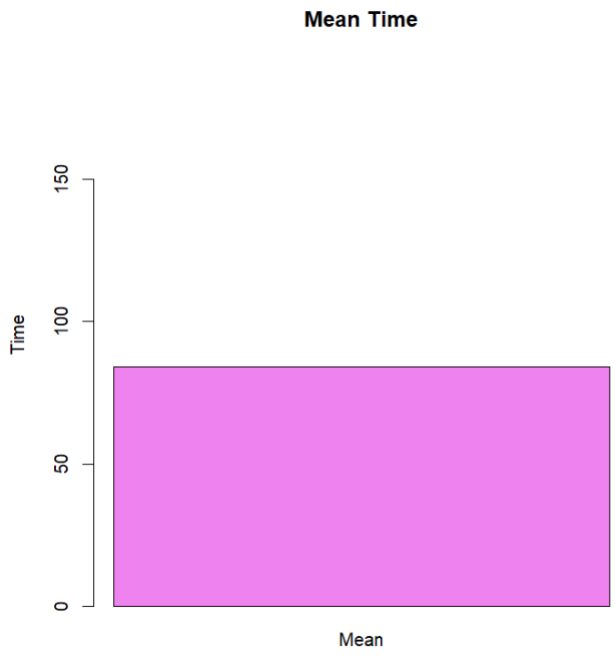
First, missing values of serum_sodium attribute were replaced by serum_sodiumMean. Then barplot is used to display the mean serum_sodium value in the graph. ylim = c(0, max(dataset$ serum_sodium) + 15) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Mean for time on a Graph**

Code:

```
missingValues <- which(is.na(dataset$time))

timeMean <- floor(mean(dataset$time, na.rm = TRUE))

dataset$time[missingValues] <- timeMean

cat("Time Mean:", timeMean)

dataset$time

barplot(timeMean, main = "Mean Time", ylab = "Time", names.arg = "Mean", col = "violet",
ylim = c(0, max(dataset$time) + 5))
```

Output:



First, missing values of time attribute were replaced by timeMean. Then barplot is used to display the mean time value in the graph. ylim = c(0, max(dataset$time) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

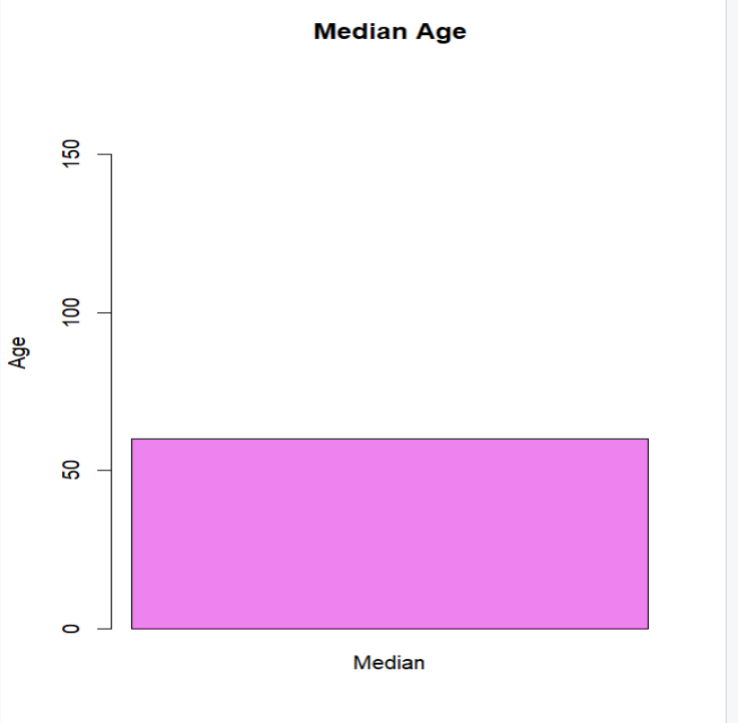
MEDIAN ON GRAPH

❖ **Show Median for age on a Graph**

Code:

```
missingValues <- which(is.na(dataset$age))

ageMedian <- floor(median(dataset$age, na.rm = TRUE))

dataset$age[missingValues] <- ageMedian

cat("Age Median:", ageMedian)

dataset$age

barplot(ageMedian, main = "Median Age", ylab = "Age", names.arg = "Median", col = "violet", ylim = c(0, max(dataset$age) + 5))
```

Output:



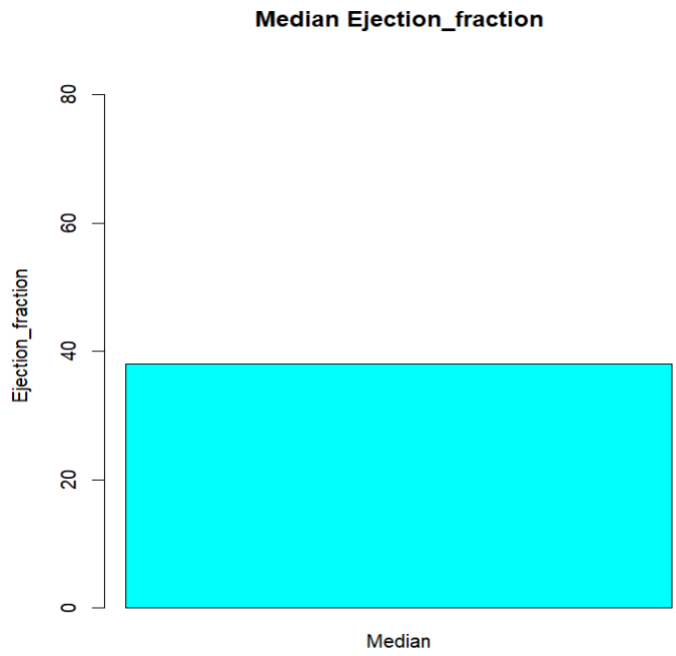
First, missing values of age attribute were replaced by ageMedian. Then barplot is used to display the median age value in the graph. ylim = c(0, max(dataset$age) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Median for ejection_fraction on a Graph**

Code:

```
missingValues <- which(is.na(dataset$ejection_fraction))

ejection_fractionMedian <- floor(median(dataset$ejection_fraction, na.rm = TRUE))

dataset$ejection_fraction[missingValues] <- ejection_fractionMedian

cat("Ejection_fraction Median:", ejection_fractionMedian)

dataset$ejection_fraction

barplot(ejection_fractionMedian, main = "Median Ejection_fraction", ylab = "Ejection_fraction",
names.arg = "Median", col = "cyan", ylim = c(0, max(dataset$ejection_fraction) + 5))
```

Output:



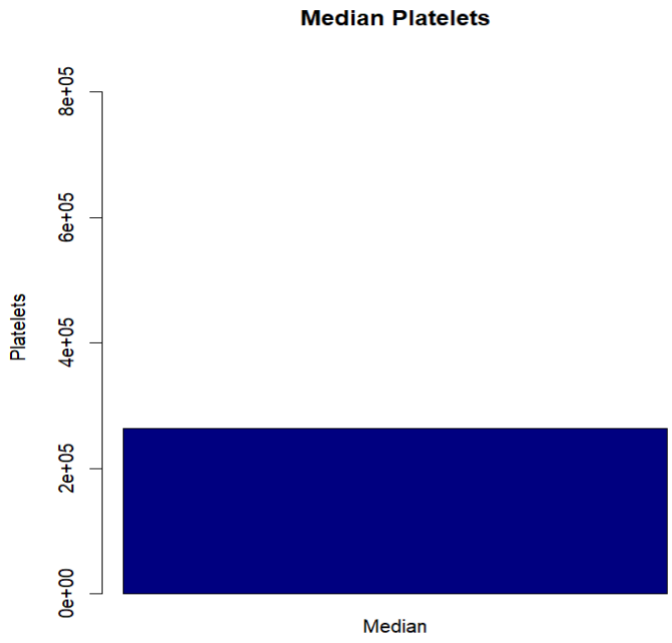
First, missing values of ejection_fraction attribute were replaced by ejection_fractionMedian. Then barplot is used to display the median ejection_fraction value in the graph. ylim = c(0, max(dataset$ejection_fraction) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Median for platelets on a Graph**

Code:

```
missingValues <- which(is.na(dataset$platelets))

plateletsMedian <- floor(median(dataset$platelets, na.rm = TRUE))

dataset$platelets[missingValues] <- plateletsMedian

cat("Platelets Median:", plateletsMedian)

dataset$platelets

barplot(plateletsMedian, main = "Median Platelets", ylab = "Platelets", names.arg = "Median", col = "navyblue", ylim = c(0, max(dataset$platelets) + 5))
```

Output:



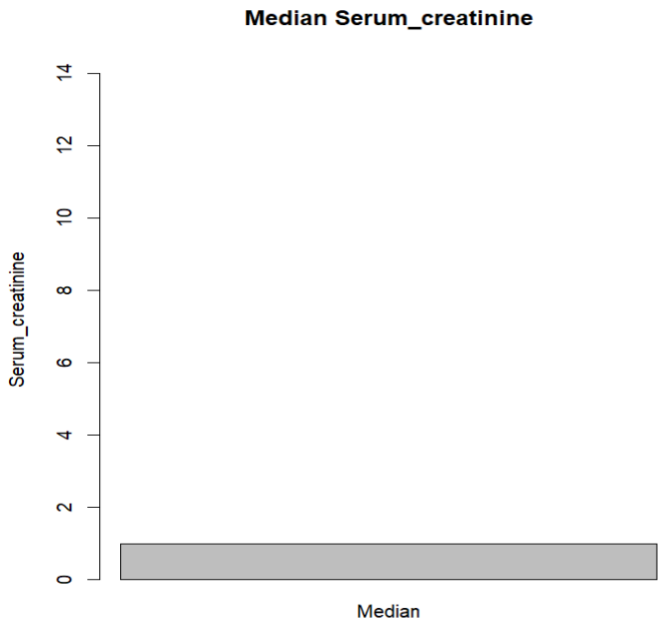
First, missing values of platelets attribute were replaced by plateletsMedian. Then barplot is used to display the median platelets value in the graph. ylim = c(0, max(dataset$platelets) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Median for serum_creatinine on a Graph**

Code:

```
missingValues <- which(is.na(dataset$serum_creatinine))

serum_creatinineMedian <- floor(median(dataset$serum_creatinine, na.rm = TRUE))

dataset$serum_creatinine[missingValues] <- serum_creatinineMedian

cat("Serum_creatinine Median:", serum_creatinineMedian)

dataset$serum_creatinine

barplot(serum_creatinineMedian, main = "Median Serum_creatinine", ylab = "Serum_creatinine",
names.arg = "Median", col = "grey", ylim = c(0, max(dataset$serum_creatinine) + 5))
```

Output:



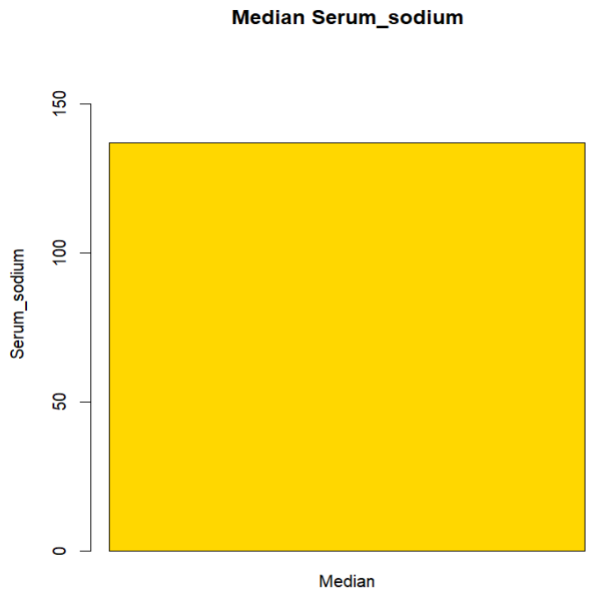
First, missing values of serum_creatinine attribute were replaced by serum_creatinineMedian. Then barplot is used to display the median serum_creatinine value in the graph. ylim = c(0, max(dataset$serum_creatinine) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Median for serum_sodium on a Graph**

Code:

```
missingValues <- which(is.na(dataset$serum_sodium))

serum_sodiumMedian <- floor(median(dataset$serum_sodium, na.rm = TRUE))

dataset$serum_sodium[missingValues] <- serum_sodiumMedian

cat("Serum_sodium Median:", serum_sodiumMedian)

dataset$serum_sodium

barplot(serum_sodiumMedian, main = "Median Serum_sodium", ylab = "Serum_sodium", names.arg = "Median", col = "gold", ylim = c(0, max(dataset$serum_sodium) + 20))
```

Output:



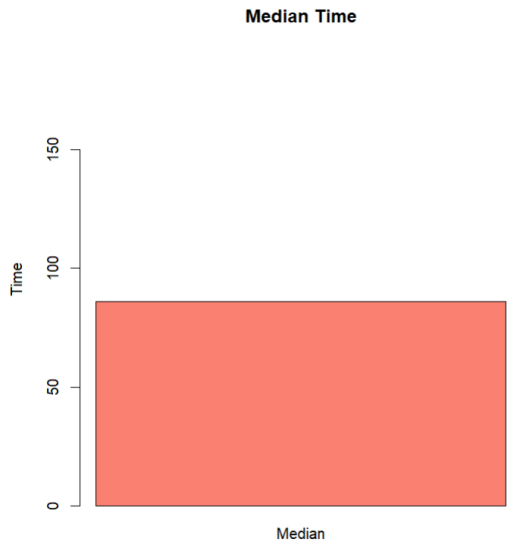
First, missing values of serum_ sodium attribute were replaced by serum_ sodiumMedian. Then barplot is used to display the median serum_sodium value in the graph. ylim = c(0, max(dataset$serum_sodium) + 20) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

❖ **Show Median for time on a Graph**

Code:

```
missingValues <- which(is.na(dataset$time))

timeMedian <- floor(median(dataset$time, na.rm = TRUE))

dataset$time[missingValues] <- timeMedian

cat("Time Median:", timeMedian)

dataset$time

barplot(timeMedian, main = "Median Time", ylab = "Time", names.arg = "Median", col = "salmon", ylim = c(0, max(dataset$time) + 5))
```

Output:



First, missing values of time attribute were replaced by timeMedian. Then barplot is used to display the median time value in the graph. ylim = c(0, max(dataset$time) + 5) parameter is used to leave some extra space at the top of the plot, making it visually more appealing and easier to interpret.

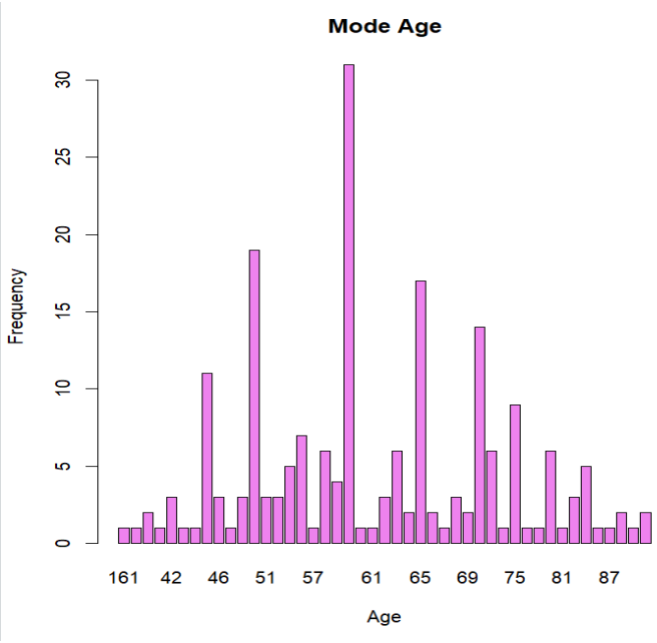
MODE ON GRAPH

❖ **Show Mode for age on a Graph**

Code:

```
missingValues <- which(is.na(dataset$age))
ageMode <- names(sort(table(dataset$age[!is.na(dataset$age)]), decreasing = TRUE)[1])
dataset$age[missingValues] <- ageMode
cat("Age Mode:", ageMode)
dataset$age
barplot(table(dataset$age), main = "Mode Age", xlab="Age", ylab = "Frequency",col = "violet")
```

Output:



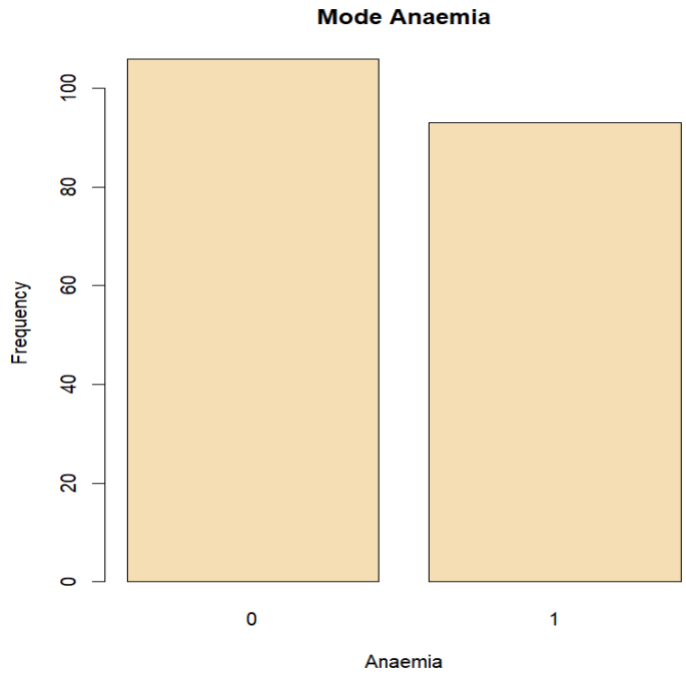
First, missing values of age attribute were replaced by ageMode. Then barplot is used to display the mode age value in the graph.

❖ **Show Mode for anaemia on a Graph**

Code:

```
missingValues <- which(is.na(dataset$anaemia))

anaemiaMode <- names(sort(table(dataset$anaemia[!is.na(dataset$anaemia)]), decreasing = TRUE)[1])

dataset$anaemia[missingValues] <- anaemiaMode

cat("Anaemia Mode:", anaemiaMode)

dataset$anaemia

barplot(table(dataset$anaemia), main = "Mode Anaemia", xlab="Anaemia", ylab = "Frequency",col = "wheat")
```

Output:



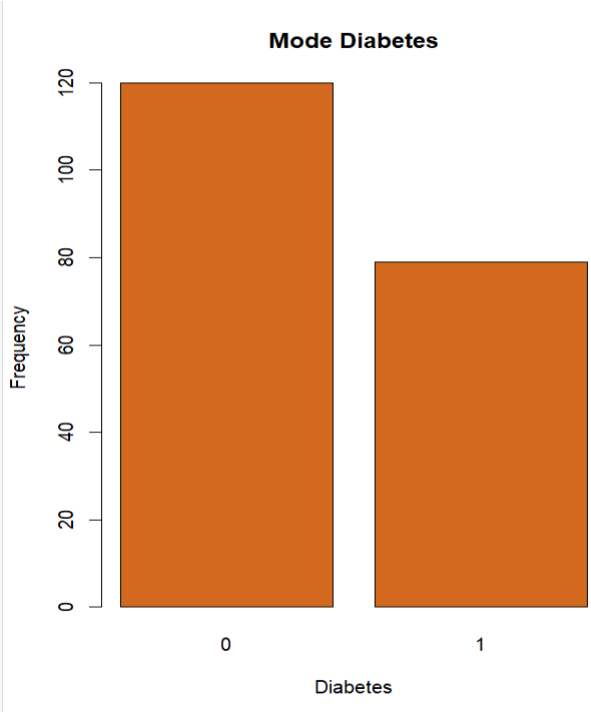
First, missing values of anaemia attribute were replaced by anaemiaMode. Then barplot is used to display the mode anaemia value in the graph.

❖ **Show Mode for diabetes on a Graph**

Code:

```
missingValues <- which(is.na(dataset$diabetes))

diabetesMode <- names(sort(table(dataset$diabetes[!is.na(dataset$diabetes)]), decreasing = TRUE)[1])

dataset$diabetes[missingValues] <- diabetesMode

cat("Diabetes Mode:", diabetesMode)

dataset$diabetes

barplot(table(dataset$diabetes), main = "Mode Diabetes", xlab="Diabetes", ylab = "Frequency",col = "chocolate")
```

Output:



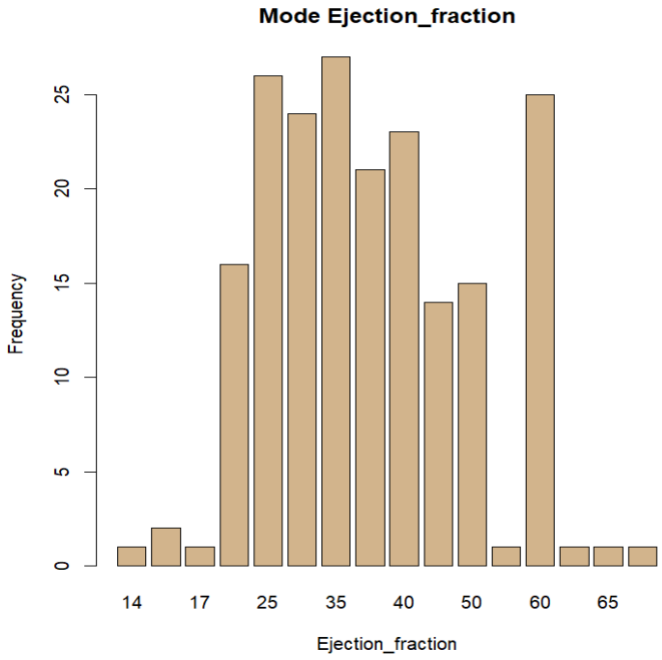
First, missing values of diabetes attribute were replaced by diabetesMode. Then barplot is used to display the mode diabetes value in the graph.

❖ **Show Mode for ejection_fraction on a Graph**

Code:

```
missingValues <- which(is.na(dataset$ejection_fraction))

ejection_fractionMode <-
names(sort(table(dataset$ejection_fraction[!is.na(dataset$ejection_fraction)]), decreasing =
TRUE)[1])

dataset$ejection_fraction[missingValues] <- ejection_fractionMode

cat("Ejection_fraction Mode:", ejection_fractionMode)

dataset$ejection_fraction

barplot(table(dataset$ejection_fraction), main = "Mode Ejection_fraction",
xlab="Ejection_fraction", ylab = "Frequency",col = "tan")
```

Output:



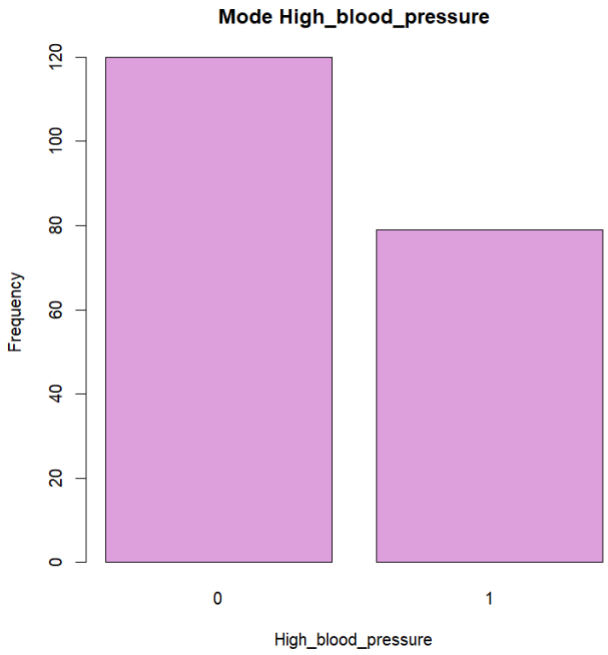
First, missing values of ejection_fraction attribute were replaced by ejection_fractionMode. Then barplot is used to display the mode ejection_fraction value in the graph.

❖ **Show Mode for high_blood_pressure on a Graph**

Code:

```
missingValues <- which(is.na(dataset$high_blood_pressure))

high_blood_pressureMode <-
names(sort(table(dataset$high_blood_pressure[!is.na(dataset$high_blood_pressure)]), decreasing
= TRUE)[1])

dataset$high_blood_pressure[missingValues] <- high_blood_pressureMode

cat("High_blood_pressure Mode:", high_blood_pressureMode)

dataset$high_blood_pressure

barplot(table(dataset$high_blood_pressure), main = "Mode High_blood_pressure",
xlab="High_blood_pressure", ylab = "Frequency",col = "plum")
```

Output:



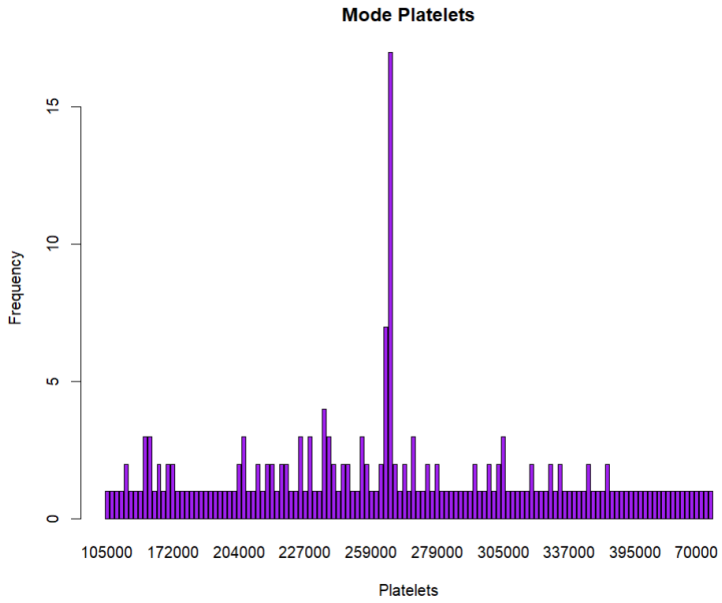
First, missing values of high_blood_pressure attribute were replaced by high_blood_pressureMode. Then barplot is used to display the mode high_blood_pressure value in the graph.

❖ **Show Mode for platelets on a Graph**

Code:

```
missingValues <- which(is.na(dataset$platelets))

plateletsMode <- names(sort(table(dataset$platelets[!is.na(dataset$platelets)]), decreasing = TRUE)[1])

dataset$platelets[missingValues] <- plateletsMode

cat("Platelets Mode:", plateletsMode)

dataset$platelets

barplot(table(dataset$platelets), main = "Mode Platelets", xlab="Platelets", ylab = "Frequency",col = "purple")
```

Output:



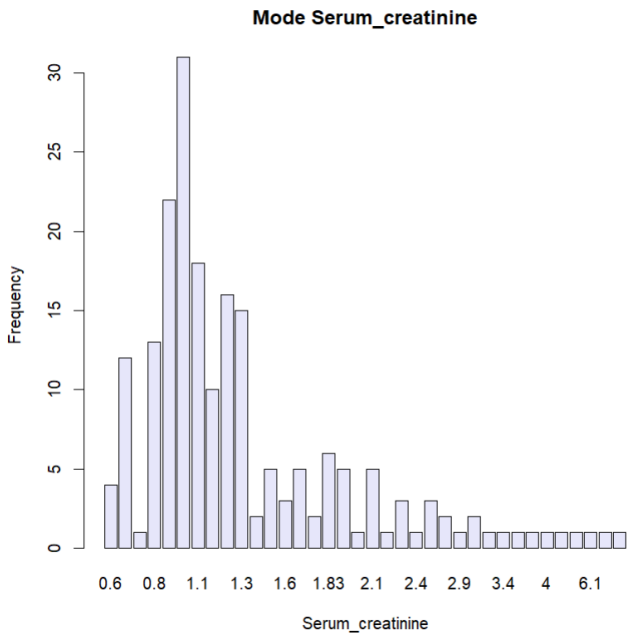
First, missing values of platelets attribute were replaced by plateletsMode. Then barplot is used to display the mode platelets value in the graph.

❖ **Show Mode for serum_creatinine on a Graph**

Code:

```
missingValues <- which(is.na(dataset$serum_creatinine))

serum_creatinineMode <-
names(sort(table(dataset$serum_creatinine[!is.na(dataset$serum_creatinine)]), decreasing =
TRUE)[1])

dataset$serum_creatinine[missingValues] <- serum_creatinineMode

cat("Serum_creatinine Mode:", serum_creatinineMode)

dataset$serum_creatinine

barplot(table(dataset$serum_creatinine), main = "Mode Serum_creatinine",
xlab="Serum_creatinine", ylab = "Frequency",col = "lavender")
```

Output:



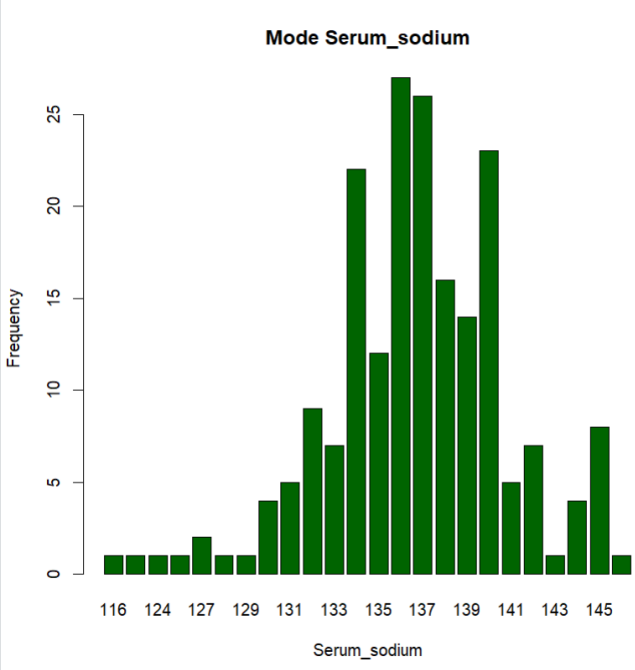
First, missing values of serum_creatinine attribute were replaced by serum_creatinineMode. Then barplot is used to display the mode serum_creatinine value in the graph.

❖ **Show Mode for serum_sodium on a Graph**

Code:

```
missingValues <- which(is.na(dataset$serum_sodium))

serum_sodiumMode <- names(sort(table(dataset$serum_sodium[!is.na(dataset$serum_sodium)]), decreasing = TRUE)[1])

dataset$serum_sodium[missingValues] <- serum_sodiumMode

cat("Serum_sodium Mode:", serum_sodiumMode)

dataset$serum_sodium

barplot(table(dataset$serum_sodium), main = "Mode Serum_sodium", xlab="Serum_sodium", ylab = "Frequency",col = "darkgreen")
```

Output:



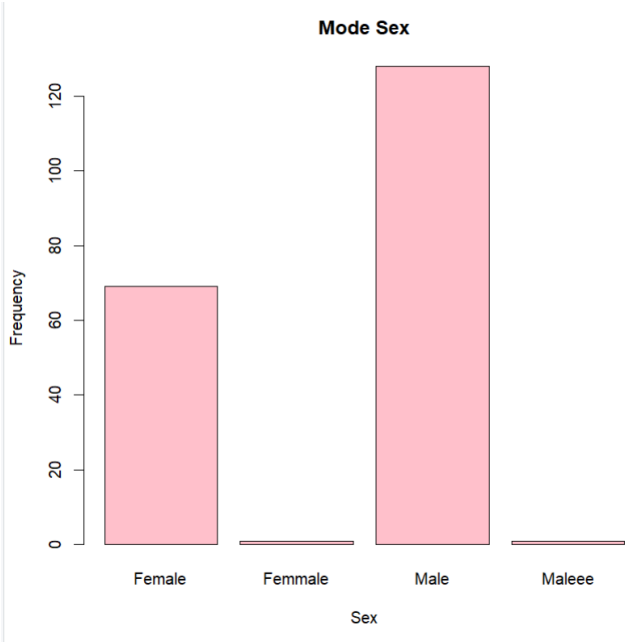
First, missing values of serum_sodium attribute were replaced by serum_sodiumMode. Then barplot is used to display the mode serum_sodium value in the graph.

❖ **Show Mode for sex on a Graph**

Code:

```
missingValues <- which(is.na(dataset$sex))
sexMode <- names(sort(table(dataset$sex[!is.na(dataset$sex)]), decreasing = TRUE)[1])
dataset$sex[missingValues] <- sexMode
cat("Sex Mode:", sexMode)
dataset$sex
barplot(table(dataset$sex), main = "Mode Sex", xlab="Sex", ylab = "Frequency",col = "pink")
```

Output:



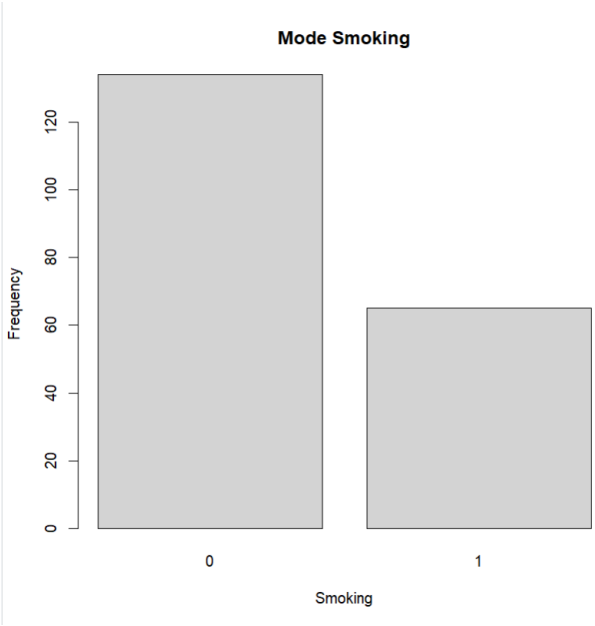
First, missing values of sex attribute were replaced by sexMode. Then barplot is used to display the mode sex value in the graph.

❖ **Show Mode for smoking on a Graph**

Code:

```
missingValues <- which(is.na(dataset$smoking))

smokingMode <- names(sort(table(dataset$smoking[!is.na(dataset$smoking)]), decreasing = TRUE)[1])

dataset$smoking[missingValues] <- smokingMode

cat("Smoking Mode:", smokingMode)

dataset$smoking

barplot(table(dataset$smoking), main = "Mode Smoking", xlab="Smoking", ylab = "Frequency",col = "lightgrey")
```

Output:



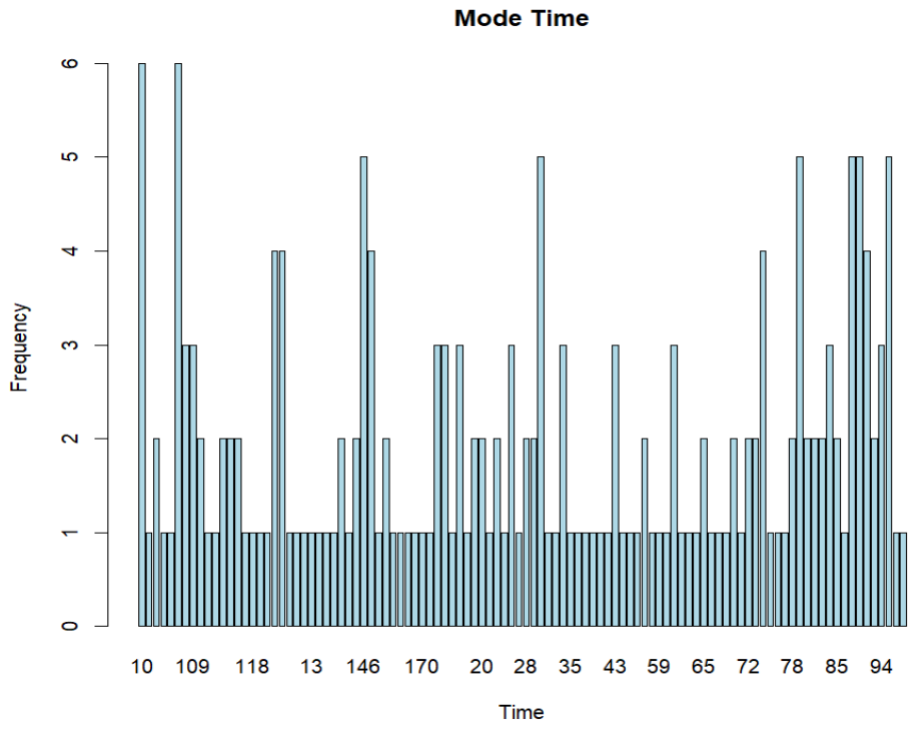
First, missing values of smoking attribute were replaced by smokingMode. Then barplot is used to display the mode smoking value in the graph.

❖ **Show Mode for time on a Graph**

Code:

```
missingValues <- which(is.na(dataset$time))

timeMode <- names(sort(table(dataset$time[!is.na(dataset$time)]), decreasing = TRUE)[1])

dataset$time[missingValues] <- timeMode

cat("Time Mode:", timeMode)

dataset$time

barplot(table(dataset$time), main = "Mode Time", xlab="Time", ylab = "Frequency",col = "lightblue")
```

Output:



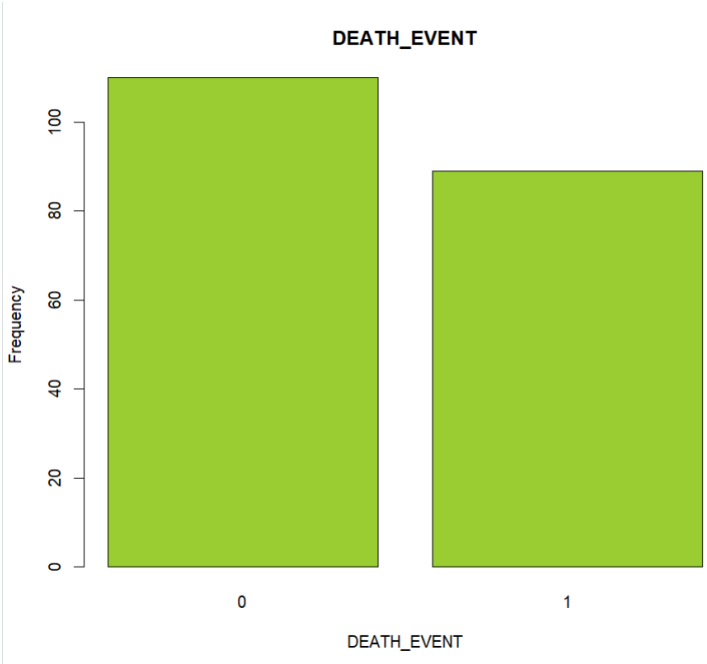
First, missing values of time attribute were replaced by timeMode. Then barplot is used to display the mode time value in the graph.

❖ **Show Mode for DEATH_EVENT on a Graph**

Code:

```
missingValues <- which(is.na(dataset$DEATH_EVENT))

DEATH_EVENTMode <-
names(sort(table(dataset$DEATH_EVENT[!is.na(dataset$DEATH_EVENT)]), decreasing =
TRUE)[1])

dataset$DEATH_EVENT[missingValues] <- DEATH_EVENTMode

cat("DEATH_EVENT:", DEATH_EVENTMode)

dataset$DEATH_EVENT

barplot(table(dataset$DEATH_EVENT), main = "DEATH_EVENT", xlab="DEATH_EVENT",
ylab = "Frequency",col = "yellowgreen")
```

Output:



First, missing values of DEATH_EVENT attribute were replaced by DEATH_EVENTMode. Then barplot is used to display the mode DEATH_EVENT value in the graph.

CONVERT NUMERICAL ATTRIBUTES

INTO CATEGORICAL ATTRIBUTES & VICE VERSA

❖ **Convert Numerical age into Categorical age**

Code:

```
dataset1 <- na.omit(dataset)

age_categories <- cut(dataset1$age, breaks = c(0, 30, 50, 70, Inf), labels = c("Young", "Middle-aged", "Old", "Very Old"))

dataset1$age <- age_categories

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> age_categories <- cut(dataset1$age, breaks = c(0, 30, 50, 70, Inf), labels = c("Young", "Middle-aged", "Old", "Very Old"))
> dataset1$age <- age_categories
> dataset1
           age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1     Very Old       0                     582        0                20                   1    265000            1.90          130    Male       0    4           1
2          Old       0                    7861        0                38                   0    263358            1.10          136    Male       0    6           1
3          Old       0                     146        0                20                   0    162000            1.30          129    Male       1    7           1
4  Middle-aged       1                     111        0                20                   0    210000            1.90          137    Male       0    7           1
5          Old       1                     160        1                20                   0    327000            2.70          116  Female       0    8           1
6     Very Old       1                      47        0                40                   1    204000            2.10          132    Male       1    8           1
9          Old       0                     157        0                65                   0    263358            1.50          138  Female       0   10           1
11    Very Old       1                      81        0                38                   1    368000            4.00          131    Male       1   10           1
13 Middle-aged       1                     981        0                30                   0    136000            1.10          137    Male       0   11           1
14 Middle-aged       1                     168        0                38                   1    276000            1.10          137    Male       0   11           1
15 Middle-aged       1                      80        0                30                   1    427000            1.00          138  Female       0   12           0
16    Very Old       1                     379        0                50                   0     47000            1.30          136    Male       0   13           1
19         Old       1                     125        0                25                   1    237000            1.00          140  Female       0   15           1
20 Middle-aged       1                     582        1                55                   0     87000            1.90          121  Female       0   15           1
21         Old       1                      52        0                25                   1    276000            1.30          137  Female       0   16           0
23         Old       1                     220        0                35                   1    289000            0.90          140    Male       1   20           1
24         Old       0                      63        1                60                   0    368000            0.80          135    Male       0   22           0
25    Very Old       0                     582        1                30                   1    263358            1.83          134  Female       0   23           1
26    Very Old       0                     148        1                38                   0    149000            1.90          144    Male       1   23           1
28         Old       0                     122        1                45                   1    284000            1.30          136    Male       1   26           1
30    Very Old       0                      70        1                30                   0    200000            1.20          132    Male       1   26           1
31    Very Old       0                     582        1                38                   1    263358            1.83          134    Male       0   27           1
32    Very Old       0                      23        0                45                   0    360000            3.00          132    Male       0   28           1
33 Middle-aged       1                     249        1                35                   1    319000            1.00          128 Femmale       0   28           1
34 Middle-aged       1                     159        1                30                   0    302000            1.20          138  Female       0   29           0
35         Old       0                      94        1                50                   1    188000            1.00          140    Male       0   29           1
38    Very Old       1                     855        1                50                   1    321000            1.00          145  Female       0   30           1
39         Old       0                    2656        1                30                   0    305000            2.30          137    Male       0   30           0
41         Old       0                     582        0                20                   1    263358            1.83          134    Male       1   31           1
42 Middle-aged       0                     124        1                30                   1    153000            1.20          136  Female       1   32           1
43         Old       0                     571        1                45                   1    185000            1.20          139    Male       1   33           1
46 Middle-aged       0                     582        1                38                   0    310000            1.90          135    Male       1   35           1
47         Old       0                    1380        0                25                   1    271000            0.90          130    Male       0   38           1
49    Very Old       1                     553        0                20                   1    140000            4.40          133    Male       0   41           1
52         Old       1                      91        0                20                   1    418000            1.40          139  Female       0   43           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of age is converted into categorical values. Here age is categorized using breaks = c(0, 30, 50, 70, Inf) range and labeled using labels = c("Young", "Middle-aged", "Old", "Very Old").

❖ **Convert Numerical anaemia into Categorical anaemia**

Code:

```
dataset1 <- na.omit(dataset)
dataset1$anaemia<-factor(dataset1$anaemia,levels =c(0,1),labels = c("No","Yes"))
dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> dataset1$anaemia<-factor(dataset1$anaemia,levels =c(0,1),labels = c("No","Yes"))
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75      No                     582        0                20                   1    265000            1.90           130    Male       0    4           1
2   55      No                    7861        0                38                   0    263358            1.10           136    Male       0    6           1
3   65      No                     146        0                20                   0    162000            1.30           129    Male       1    7           1
4   50     Yes                     111        0                20                   0    210000            1.90           137    Male       0    7           1
5   65     Yes                     160        1                20                   0    327000            2.70           116  Female       0    8           1
6   90     Yes                      47        0                40                   1    204000            2.10           132    Male       1    8           1
9   65      No                     157        0                65                   0    263358            1.50           138  Female       0   10           1
11  75     Yes                      81        0                38                   1    368000            4.00           131    Male       1   10           1
13  45     Yes                     981        0                30                   0    136000            1.10           137    Male       0   11           1
14  50     Yes                     168        0                38                   1    276000            1.10           137    Male       0   11           1
15  49     Yes                      80        0                30                   1    427000            1.00           138  Female       0   12           0
16  82     Yes                     379        0                50                   0     47000            1.30           136    Male       0   13           1
19  70     Yes                     125        0                25                   1    237000            1.00           140  Female       0   15           1
20  48     Yes                     582        1                55                   0     87000            1.90           121  Female       0   15           1
21  65     Yes                      52        0                25                   1    276000            1.30           137  Female       0   16           0
23  68     Yes                     220        0                35                   1    289000            0.90           140    Male       1   20           1
24  53      No                      63        1                60                   0    368000            0.80           135    Male       0   22           0
25  75      No                     582        1                30                   1    263358            1.83           134  Female       0   23           1
26  80      No                     148        1                38                   0    149000            1.90           144    Male       1   23           1
28  70      No                     122        1                45                   1    284000            1.30           136    Male       1   26           1
30  82      No                      70        1                30                   0    200000            1.20           132    Male       1   26           1
31  94      No                     582        1                38                   1    263358            1.83           134    Male       0   27           1
32  85      No                      23        0                45                   0    360000            3.00           132    Male       0   28           1
33  50     Yes                     249        1                35                   1    319000            1.00           128  Femmale      0   28           1
34  50     Yes                     159        1                30                   0    302000            1.20           138  Female       0   29           0
35  65      No                      94        1                50                   1    188000            1.00           140    Male       0   29           1
38  82     Yes                     855        1                50                   1    321000            1.00           145  Female       0   30           1
39  60      No                    2656        1                30                   0    305000            2.30           137    Male       0   30           0
41  70      No                     582        0                20                   1    263358            1.83           134    Male       1   31           1
42  50      No                     124        1                30                   1    153000            1.20           136  Female       1   32           1
43  70      No                     571        1                45                   1    185000            1.20           139    Male       1   33           1
46  50      No                     582        1                38                   0    310000            1.90           135    Male       1   35           1
47  51      No                    1380        0                25                   1    271000            0.90           130    Male       0   38           1
49  80     Yes                     553        0                20                   1    140000            4.40           133    Male       0   41           1
52  53     Yes                      91        0                20                   1    418000            1.40           139  Female       0   43           1
53  60      No                    3964        1                62                   0    263358            6.80           146  Female       0   43           1
54  70     Yes                      69        1                50                   1    351000            1.00           134  Female       0   44           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of anaemia is converted into categorical values. Here 0 = “No” (doesn’t have anaemia) and 1 = “Yes”(has anaemia).

❖ **Convert Numerical diabetes into Categorical diabetes**

Code:

```
dataset1 <- na.omit(dataset)
dataset1$diabetes<-factor(dataset1$diabetes,levels =c(0,1),labels = c("No","Yes"))
dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> dataset1$diabetes<-factor(dataset1$diabetes,levels =c(0,1),labels = c("No","Yes"))
> dataset1
    age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1    75       0                     582       No                20                    1    265000            1.90           130    Male       0    4           1
2    55       0                    7861       No                38                    0    263358            1.10           136    Male       0    6           1
3    65       0                     146       No                20                    0    162000            1.30           129    Male       1    7           1
4    50       1                     111       No                20                    0    210000            1.90           137    Male       0    7           1
5    65       1                     160      Yes                20                    0    327000            2.70           116  Female       0    8           1
6    90       1                      47       No                40                    1    204000            2.10           132    Male       1    8           1
9    65       0                     157       No                65                    0    263358            1.50           138  Female       0   10           1
11   75       1                      81       No                38                    1    368000            4.00           131    Male       1   10           1
13   45       1                     981       No                30                    0    136000            1.10           137    Male       0   11           1
14   50       1                     168       No                38                    1    276000            1.10           137    Male       0   11           1
15   49       1                      80       No                30                    1    427000            1.00           138  Female       0   12           0
16   82       1                     379       No                50                    0     47000            1.30           136    Male       0   13           1
19   70       1                     125       No                25                    1    237000            1.00           140  Female       0   15           1
20   48       1                     582      Yes                55                    0     87000            1.90           121  Female       0   15           1
21   65       1                      52       No                25                    1    276000            1.30           137  Female       0   16           0
23   68       1                     220       No                35                    1    289000            0.90           140    Male       1   20           1
24   53       0                      63      Yes                60                    0    368000            0.80           135    Male       0   22           0
25   75       0                     582      Yes                30                    1    263358            1.83           134  Female       0   23           1
26   80       0                     148      Yes                38                    0    149000            1.90           144    Male       1   23           1
28   70       0                     122      Yes                45                    1    284000            1.30           136    Male       1   26           1
30   82       0                      70      Yes                30                    0    200000            1.20           132    Male       1   26           1
31   94       0                     582      Yes                38                    1    263358            1.83           134    Male       0   27           1
32   85       0                      23       No                45                    0    360000            3.00           132    Male       0   28           1
33   50       1                     249      Yes                35                    1    319000            1.00           128  Femmale      0   28           1
34   50       1                     159      Yes                30                    0    302000            1.20           138  Female       0   29           0
35   65       0                      94      Yes                50                    1    188000            1.00           140    Male       0   29           1
38   82       1                     855      Yes                50                    1    321000            1.00           145  Female       0   30           1
39   60       0                    2656      Yes                30                    0    305000            2.30           137    Male       0   30           0
41   70       0                     582       No                20                    1    263358            1.83           134    Male       1   31           1
42   50       0                     124      Yes                30                    1    153000            1.20           136  Female       1   32           1
43   70       0                     571      Yes                45                    1    185000            1.20           139    Male       1   33           1
46   50       0                     582      Yes                38                    0    310000            1.90           135    Male       1   35           1
47   51       0                    1380       No                25                    1    271000            0.90           130    Male       0   38           1
49   80       1                     553       No                20                    1    140000            4.40           133    Male       0   41           1
52   53       1                      91       No                20                    1    418000            1.40           139  Female       0   43           1
53   60       0                    3964      Yes                62                    0    263358            6.80           146  Female       0   43           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of diabetes is converted into categorical values. Here 0 = “No” (doesn’t have diabetes) and 1 = “Yes”(has diabetes).

❖ **Convert Numerical ejection_fraction into Categorical ejection_fraction**

Code:

```
dataset1 <- na.omit(dataset)

ejection_fraction_categories <- cut(dataset1$ejection_fraction, breaks = c(0, 40, 55, 70, Inf), labels = c("Possible Heart Failure", "Low Function", "Normal Function", "High Function"))

dataset1$ejection_fraction <- ejection_fraction_categories

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> ejection_fraction_categories <- cut(dataset1$ejection_fraction, breaks = c(0, 40, 55, 70, Inf), labels = c("Possible Heart Failure", "Low Function",
"Normal Function", "High Function"))
> dataset1$ejection_fraction <- ejection_fraction_categories
> dataset1
   age anaemia creatinine_phosphokinase diabetes      ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75       0                     582        0 Possible Heart Failure                   1    265000            1.90          130    Male       0    4           1
2   55       0                    7861        0 Possible Heart Failure                   0    263358            1.10          136    Male       0    6           1
3   65       0                     146        0 Possible Heart Failure                   0    162000            1.30          129    Male       1    7           1
4   50       1                     111        0 Possible Heart Failure                   0    210000            1.90          137    Male       0    7           1
5   65       1                     160        1 Possible Heart Failure                   0    327000            2.70          116  Female       0    8           1
6   90       1                      47        0 Possible Heart Failure                   1    204000            2.10          132    Male       1    8           1
9   65       0                     157        0        Normal Function                   0    263358            1.50          138  Female       0   10           1
11  75       1                      81        0 Possible Heart Failure                   1    368000            4.00          131    Male       1   10           1
13  45       1                     981        0 Possible Heart Failure                   0    136000            1.10          137    Male       0   11           1
14  50       1                     168        0 Possible Heart Failure                   1    276000            1.10          137    Male       0   11           1
15  49       1                      80        0 Possible Heart Failure                   1    427000            1.00          138  Female       0   12           0
16  82       1                     379        0          Low Function                    0     47000            1.30          136    Male       0   13           1
19  70       1                     125        0 Possible Heart Failure                   1    237000            1.00          140  Female       0   15           1
20  48       1                     582        1          Low Function                    0     87000            1.90          121  Female       0   15           1
21  65       1                      52        0 Possible Heart Failure                   1    276000            1.30          137  Female       0   16           0
23  68       1                     220        0 Possible Heart Failure                   1    289000            0.90          140    Male       1   20           1
24  53       0                      63        1        Normal Function                   0    368000            0.80          135    Male       0   22           0
25  75       0                     582        1 Possible Heart Failure                   1    263358            1.83          134  Female       0   23           1
26  80       0                     148        1 Possible Heart Failure                   0    149000            1.90          144    Male       1   23           1
28  70       0                     122        1          Low Function                    1    284000            1.30          136    Male       1   26           1
30  82       0                      70        1 Possible Heart Failure                   0    200000            1.20          132    Male       1   26           1
31  94       0                     582        1 Possible Heart Failure                   1    263358            1.83          134    Male       0   27           1
32  85       0                      23        0          Low Function                    0    360000            3.00          132    Male       0   28           1
33  50       1                     249        1 Possible Heart Failure                   1    319000            1.00          128 Femmale       0   28           1
34  50       1                     159        1 Possible Heart Failure                   0    302000            1.20          138  Female       0   29           0
35  65       0                      94        1          Low Function                    1    188000            1.00          140    Male       0   29           1
38  82       1                     855        1          Low Function                    1    321000            1.00          145  Female       0   30           1
39  60       0                    2656        1 Possible Heart Failure                   0    305000            2.30          137    Male       0   30           0
41  70       0                     582        0 Possible Heart Failure                   1    263358            1.83          134    Male       1   31           1
42  50       0                     124        1 Possible Heart Failure                   1    153000            1.20          136  Female       1   32           1
43  70       0                     571        1          Low Function                    1    185000            1.20          139    Male       1   33           1
46  50       0                     582        1 Possible Heart Failure                   0    310000            1.90          135    Male       1   35           1
47  51       0                    1380        0 Possible Heart Failure                   1    271000            0.90          130    Male       0   38           1
49  80       1                     553        0 Possible Heart Failure                   1    140000            4.40          133    Male       0   41           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of ejection_fraction is converted into categorical values. Here ejection_fraction is categorized using breaks = c(0, 40, 55, 70, Inf) range and labeled using labels = c("Possible Heart Failure", "Low Function", "Normal Function", "High Function").

❖ **Convert Numerical high_blood_pressure into Categorical high_blood_pressure**

Code:

```
dataset1 <- na.omit(dataset)

dataset1$high_blood_pressure<-factor(dataset1$high_blood_pressure,levels =c(0,1),labels = c("No","Yes"))

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> dataset1$high_blood_pressure<-factor(dataset1$high_blood_pressure,levels =c(0,1),labels = c("No","Yes"))
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75       0                     582        0                20                Yes    265000            1.90           130    Male       0    4           1
2   55       0                    7861        0                38                 No    263358            1.10           136    Male       0    6           1
3   65       0                     146        0                20                 No    162000            1.30           129    Male       1    7           1
4   50       1                     111        0                20                 No    210000            1.90           137    Male       0    7           1
5   65       1                     160        1                20                 No    327000            2.70           116  Female       0    8           1
6   90       1                      47        0                40                Yes    204000            2.10           132    Male       1    8           1
9   65       0                     157        0                65                 No    263358            1.50           138  Female       0   10           1
11  75       1                      81        0                38                Yes    368000            4.00           131    Male       1   10           1
13  45       1                     981        0                30                 No    136000            1.10           137    Male       0   11           1
14  50       1                     168        0                38                Yes    276000            1.10           137    Male       0   11           1
15  49       1                      80        0                30                Yes    427000            1.00           138  Female       0   12           0
16  82       1                     379        0                50                 No     47000            1.30           136    Male       0   13           1
19  70       1                     125        0                25                Yes    237000            1.00           140  Female       0   15           1
20  48       1                     582        1                55                 No     87000            1.90           121  Female       0   15           1
21  65       1                      52        0                25                Yes    276000            1.30           137  Female       0   16           0
23  68       1                     220        0                35                Yes    289000            0.90           140    Male       1   20           1
24  53       0                      63        1                60                 No    368000            0.80           135    Male       0   22           0
25  75       0                     582        1                30                Yes    263358            1.83           134  Female       0   23           1
26  80       0                     148        1                38                 No    149000            1.90           144    Male       1   23           1
28  70       0                     122        1                45                Yes    284000            1.30           136    Male       1   26           1
30  82       0                      70        1                30                 No    200000            1.20           132    Male       1   26           1
31  94       0                     582        1                38                Yes    263358            1.83           134    Male       0   27           1
32  85       0                      23        0                45                 No    360000            3.00           132    Male       0   28           1
33  50       1                     249        1                35                Yes    319000            1.00           128  Femmale       0   28           1
34  50       1                     159        1                30                 No    302000            1.20           138  Female       0   29           0
35  65       0                      94        1                50                Yes    188000            1.00           140    Male       0   29           1
38  82       1                     855        1                50                Yes    321000            1.00           145  Female       0   30           1
39  60       0                    2656        1                30                 No    305000            2.30           137    Male       0   30           0
41  70       0                     582        0                20                Yes    263358            1.83           134    Male       1   31           1
42  50       0                     124        1                30                Yes    153000            1.20           136  Female       1   32           1
43  70       0                     571        1                45                Yes    185000            1.20           139    Male       1   33           1
46  50       0                     582        1                38                 No    310000            1.90           135    Male       1   35           1
47  51       0                    1380        0                25                Yes    271000            0.90           130    Male       0   38           1
49  80       1                     553        0                20                Yes    140000            4.40           133    Male       0   41           1
52  53       1                      91        0                20                Yes    418000            1.40           139  Female       0   43           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of high_blood_pressure is converted into categorical values. Here 0 = “No” (doesn’t have high_blood_pressure) and 1 = “Yes”(has high_blood_pressure).

❖ **Convert Numerical platelets into Categorical platelets**

Code:

```
dataset1 <- na.omit(dataset)

platelets_categories <- cut(dataset1$platelets, breaks = c(0, 150000, 450000, Inf), labels = c("Low", "Normal", "High"))

dataset1$platelets <- platelets_categories

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> platelets_categories <- cut(dataset1$platelets, breaks = c(0, 150000, 450000, Inf), labels = c("Low", "Normal", "High"))
> dataset1$platelets <- platelets_categories
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75       0                     582        0                20                   1    Normal            1.90          130    Male       0    4           1
2   55       0                    7861        0                38                   0    Normal            1.10          136    Male       0    6           1
3   65       0                     146        0                20                   0    Normal            1.30          129    Male       1    7           1
4   50       1                     111        0                20                   0    Normal            1.90          137    Male       0    7           1
5   65       1                     160        1                20                   0    Normal            2.70          116  Female       0    8           1
6   90       1                      47        0                40                   1    Normal            2.10          132    Male       1    8           1
9   65       0                     157        0                65                   0    Normal            1.50          138  Female       0   10           1
11  75       1                      81        0                38                   1    Normal            4.00          131    Male       1   10           1
13  45       1                     981        0                30                   0       Low            1.10          137    Male       0   11           1
14  50       1                     168        0                38                   1    Normal            1.10          137    Male       0   11           1
15  49       1                      80        0                30                   1    Normal            1.00          138  Female       0   12           0
16  82       1                     379        0                50                   0       Low            1.30          136    Male       0   13           1
19  70       1                     125        0                25                   1    Normal            1.00          140  Female       0   15           1
20  48       1                     582        1                55                   0       Low            1.90          121  Female       0   15           1
21  65       1                      52        0                25                   1    Normal            1.30          137  Female       0   16           0
23  68       1                     220        0                35                   1    Normal            0.90          140    Male       1   20           1
24  53       0                      63        1                60                   0    Normal            0.80          135    Male       0   22           0
25  75       0                     582        1                30                   1    Normal            1.83          134  Female       0   23           1
26  80       0                     148        1                38                   0       Low            1.90          144    Male       1   23           1
28  70       0                     122        1                45                   1    Normal            1.30          136    Male       1   26           1
30  82       0                      70        1                30                   0    Normal            1.20          132    Male       1   26           1
31  94       0                     582        1                38                   1    Normal            1.83          134    Male       0   27           1
32  85       0                      23        0                45                   0    Normal            3.00          132    Male       0   28           1
33  50       1                     249        1                35                   1    Normal            1.00          128 Femmale       0   28           1
34  50       1                     159        1                30                   0    Normal            1.20          138  Female       0   29           0
35  65       0                      94        1                50                   1    Normal            1.00          140    Male       0   29           1
38  82       1                     855        1                50                   1    Normal            1.00          145  Female       0   30           1
39  60       0                    2656        1                30                   0    Normal            2.30          137    Male       0   30           0
41  70       0                     582        0                20                   1    Normal            1.83          134    Male       1   31           1
42  50       0                     124        1                30                   1    Normal            1.20          136  Female       1   32           1
43  70       0                     571        1                45                   1    Normal            1.20          139    Male       1   33           1
46  50       0                     582        1                38                   0    Normal            1.90          135    Male       1   35           1
47  51       0                    1380        0                25                   1    Normal            0.90          130    Male       0   38           1
49  80       1                     553        0                20                   1       Low            4.40          133    Male       0   41           1
52  53       1                      91        0                20                   1    Normal            1.40          139  Female       0   43           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of platelets is converted into categorical values. Here platelets is categorized using breaks = c(0, 150000, 450000, Inf) range and labeled using labels = c("Low", "Normal", "High").

❖ **Convert Numerical serum_creatinine into Categorical serum_creatinine**

Code:

```
dataset1 <- na.omit(dataset)

serum_creatinine_categories <- cut(dataset1$serum_creatinine, breaks = c(0, 0.7, 1.3, Inf), labels = c("Low", "Normal", "High"))

dataset1$serum_creatinine <- serum_creatinine_categories

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> serum_creatinine_categories <- cut(dataset1$serum_creatinine, breaks = c(0, 0.7, 1.3, Inf), labels = c("Low", "Normal", "High"))
> dataset1$serum_creatinine <- serum_creatinine_categories
> dataset1
    age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking  time DEATH_EVENT
1    75       0                     582        0                20                  1    265000             High           130    Male       0     4           1
2    55       0                    7861        0                38                  0    263358           Normal           136    Male       0     6           1
3    65       0                     146        0                20                  0    162000           Normal           129    Male       1     7           1
4    50       1                     111        0                20                  0    210000             High           137    Male       0     7           1
5    65       1                     160        1                20                  0    327000             High           116  Female       0     8           1
6    90       1                      47        0                40                  1    204000             High           132    Male       1     8           1
9    65       0                     157        0                65                  0    263358             High           138  Female       0    10           1
11   75       1                      81        0                38                  1    368000             High           131    Male       1    10           1
13   45       1                     981        0                30                  0    136000           Normal           137    Male       0    11           1
14   50       1                     168        0                38                  1    276000           Normal           137    Male       0    11           1
15   49       1                      80        0                30                  1    427000           Normal           138  Female       0    12           0
16   82       1                     379        0                50                  0     47000           Normal           136    Male       0    13           1
19   70       1                     125        0                25                  1    237000           Normal           140  Female       0    15           1
20   48       1                     582        1                55                  0     87000             High           121  Female       0    15           1
21   65       1                      52        0                25                  1    276000           Normal           137  Female       0    16           0
23   68       1                     220        0                35                  1    289000           Normal           140    Male       1    20           1
24   53       0                      63        1                60                  0    368000           Normal           135    Male       0    22           0
25   75       0                     582        1                30                  1    263358             High           134  Female       0    23           1
26   80       0                     148        1                38                  0    149000             High           144    Male       1    23           1
28   70       0                     122        1                45                  1    284000           Normal           136    Male       1    26           1
30   82       0                      70        1                30                  0    200000           Normal           132    Male       1    26           1
31   94       0                     582        1                38                  1    263358             High           134    Male       0    27           1
32   85       0                      23        0                45                  0    360000             High           132    Male       0    28           1
33   50       1                     249        1                35                  1    319000           Normal           128 Femmale       0    28           1
34   50       1                     159        1                30                  0    302000           Normal           138  Female       0    29           0
35   65       0                      94        1                50                  1    188000           Normal           140    Male       0    29           1
38   82       1                     855        1                50                  1    321000           Normal           145  Female       0    30           1
39   60       0                    2656        1                30                  0    305000             High           137    Male       0    30           0
41   70       0                     582        0                20                  1    263358             High           134    Male       1    31           1
42   50       0                     124        1                30                  1    153000           Normal           136  Female       1    32           1
43   70       0                     571        1                45                  1    185000           Normal           139    Male       1    33           1
46   50       0                     582        1                38                  0    310000             High           135    Male       1    35           1
47   51       0                    1380        0                25                  1    271000           Normal           130    Male       0    38           1
49   80       1                     553        0                20                  1    140000             High           133    Male       0    41           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of serum_creatinine is converted into categorical values. Here serum_creatinine is categorized using breaks = c(0, 0.7, 1.3, Inf) range and labeled using labels = c("Low", "Normal", "High").

❖ **Convert Numerical serum_sodium into Categorical serum_sodium**

Code:

```
dataset1 <- na.omit(dataset)

serum_sodium_categories <- cut(dataset1$serum_sodium, breaks = c(0, 135, 145, Inf), labels = c("Low", "Normal", "High"))

dataset1$serum_sodium <- serum_sodium_categories

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> serum_sodium_categories <- cut(dataset1$serum_sodium, breaks = c(0, 135, 145, Inf), labels = c("Low", "Normal", "High"))
> dataset1$serum_sodium <- serum_sodium_categories
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75       0                     582        0                20                   1    265000            1.90          Low    Male       0    4           1
2   55       0                    7861        0                38                   0    263358            1.10       Normal    Male       0    6           1
3   65       0                     146        0                20                   0    162000            1.30          Low    Male       1    7           1
4   50       1                     111        0                20                   0    210000            1.90       Normal    Male       0    7           1
5   65       1                     160        1                20                   0    327000            2.70          Low  Female       0    8           1
6   90       1                      47        0                40                   1    204000            2.10          Low    Male       1    8           1
9   65       0                     157        0                65                   0    263358            1.50       Normal  Female       0   10           1
11  75       1                      81        0                38                   1    368000            4.00          Low    Male       1   10           1
13  45       1                     981        0                30                   0    136000            1.10       Normal    Male       0   11           1
14  50       1                     168        0                38                   1    276000            1.10       Normal    Male       0   11           1
15  49       1                      80        0                30                   1    427000            1.00       Normal  Female       0   12           0
16  82       1                     379        0                50                   0     47000            1.30       Normal    Male       0   13           1
19  70       1                     125        0                25                   1    237000            1.00       Normal  Female       0   15           1
20  48       1                     582        1                55                   0     87000            1.90          Low  Female       0   15           1
21  65       1                      52        0                25                   1    276000            1.30       Normal  Female       0   16           0
23  68       1                     220        0                35                   1    289000            0.90       Normal    Male       1   20           1
24  53       0                      63        1                60                   0    368000            0.80          Low    Male       0   22           0
25  75       0                     582        1                30                   1    263358            1.83          Low  Female       0   23           1
26  80       0                     148        1                38                   0    149000            1.90       Normal    Male       1   23           1
28  70       0                     122        1                45                   1    284000            1.30       Normal    Male       1   26           1
30  82       0                      70        1                30                   0    200000            1.20          Low    Male       1   26           1
31  94       0                     582        1                38                   1    263358            1.83          Low    Male       0   27           1
32  85       0                      23        0                45                   0    360000            3.00          Low    Male       0   28           1
33  50       1                     249        1                35                   1    319000            1.00          Low Femmale       0   28           1
34  50       1                     159        1                30                   0    302000            1.20       Normal  Female       0   29           0
35  65       0                      94        1                50                   1    188000            1.00       Normal    Male       0   29           1
38  82       1                     855        1                50                   1    321000            1.00       Normal  Female       0   30           1
39  60       0                    2656        1                30                   0    305000            2.30       Normal    Male       0   30           0
41  70       0                     582        0                20                   1    263358            1.83          Low    Male       1   31           1
42  50       0                     124        1                30                   1    153000            1.20       Normal  Female       1   32           1
43  70       0                     571        1                45                   1    185000            1.20       Normal    Male       1   33           1
46  50       0                     582        1                38                   0    310000            1.90          Low    Male       1   35           1
47  51       0                    1380        0                25                   1    271000            0.90          Low    Male       0   38           1
49  80       1                     553        0                20                   1    140000            4.40          Low    Male       0   41           1
52  53       1                      91        0                20                   1    418000            1.40       Normal  Female       0   43           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of serum_sodium is converted into categorical values. Here serum_sodium is categorized using breaks = c(0, 135, 145, Inf) range and labeled using labels = c("Low", "Normal", "High").

❖ **Convert Categorical sex into Numerical sex**

Code:

```
dataset1 <- na.omit(dataset)
dataset1$sex<-factor(dataset1$sex,levels =c("Male","Female"),labels = c(1,2))
dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> dataset1$sex<-factor(dataset1$sex,levels =c("Male","Female"),labels = c(1,2))
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium  sex smoking time  DEATH_EVENT
1   75       0                     582        0                20                   1    265000            1.90          130    1       0    4           1
2   55       0                    7861        0                38                   0    263358            1.10          136    1       0    6           1
3   65       0                     146        0                20                   0    162000            1.30          129    1       1    7           1
4   50       1                     111        0                20                   0    210000            1.90          137    1       0    7           1
5   65       1                     160        1                20                   0    327000            2.70          116    2       0    8           1
6   90       1                      47        0                40                   1    204000            2.10          132    1       1    8           1
9   65       0                     157        0                65                   0    263358            1.50          138    2       0   10           1
11  75       1                      81        0                38                   1    368000            4.00          131    1       1   10           1
13  45       1                     981        0                30                   0    136000            1.10          137    1       0   11           1
14  50       1                     168        0                38                   1    276000            1.10          137    1       0   11           1
15  49       1                      80        0                30                   1    427000            1.00          138    2       0   12           0
16  82       1                     379        0                50                   0     47000            1.30          136    1       0   13           1
19  70       1                     125        0                25                   1    237000            1.00          140    2       0   15           1
20  48       1                     582        1                55                   0     87000            1.90          121    2       0   15           1
21  65       1                      52        0                25                   1    276000            1.30          137    2       0   16           0
23  68       1                     220        0                35                   1    289000            0.90          140    1       1   20           1
24  53       0                      63        1                60                   0    368000            0.80          135    1       0   22           0
25  75       0                     582        1                30                   1    263358            1.83          134    2       0   23           1
26  80       0                     148        1                38                   0    149000            1.90          144    1       1   23           1
28  70       0                     122        1                45                   1    284000            1.30          136    1       1   26           1
30  82       0                      70        1                30                   0    200000            1.20          132    1       1   26           1
31  94       0                     582        1                38                   1    263358            1.83          134    1       0   27           1
32  85       0                      23        0                45                   0    360000            3.00          132    1       0   28           1
33  50       1                     249        1                35                   1    319000            1.00          128 <NA>       0   28           1
34  50       1                     159        1                30                   0    302000            1.20          138    2       0   29           0
35  65       0                      94        1                50                   1    188000            1.00          140    1       0   29           1
38  82       1                     855        1                50                   1    321000            1.00          145    2       0   30           1
39  60       0                    2656        1                30                   0    305000            2.30          137    1       0   30           0
41  70       0                     582        0                20                   1    263358            1.83          134    1       1   31           1
42  50       0                     124        1                30                   1    153000            1.20          136    2       1   32           1
43  70       0                     571        1                45                   1    185000            1.20          139    1       1   33           1
46  50       0                     582        1                38                   0    310000            1.90          135    1       1   35           1
47  51       0                    1380        0                25                   1    271000            0.90          130    1       0   38           1
49  80       1                     553        0                20                   1    140000            4.40          133    1       0   41           1
52  53       1                      91        0                20                   1    418000            1.40          139    2       0   43           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, categorical values of sex is converted into numerical values. Here Male = 1 and Female = 2. The <NA> values in the output are invalid values.

❖ **Convert Numerical smoking into Categorical smoking**

Code:

```
dataset1 <- na.omit(dataset)
dataset1$smoking<-factor(dataset1$smoking,levels =c(0,1),labels = c("No","Yes"))
dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> dataset1$smoking<-factor(dataset1$smoking,levels =c(0,1),labels = c("No","Yes"))
> dataset1
    age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1    75       0                     582        0                20                   1    265000            1.90           130    Male      No    4           1
2    55       0                    7861        0                38                   0    263358            1.10           136    Male      No    6           1
3    65       0                     146        0                20                   0    162000            1.30           129    Male     Yes    7           1
4    50       1                     111        0                20                   0    210000            1.90           137    Male      No    7           1
5    65       1                     160        1                20                   0    327000            2.70           116  Female      No    8           1
6    90       1                      47        0                40                   1    204000            2.10           132    Male     Yes    8           1
9    65       0                     157        0                65                   0    263358            1.50           138  Female      No   10           1
11   75       1                      81        0                38                   1    368000            4.00           131    Male     Yes   10           1
13   45       1                     981        0                30                   0    136000            1.10           137    Male      No   11           1
14   50       1                     168        0                38                   1    276000            1.10           137    Male      No   11           1
15   49       1                      80        0                30                   1    427000            1.00           138  Female      No   12           0
16   82       1                     379        0                50                   0     47000            1.30           136    Male      No   13           1
19   70       1                     125        0                25                   1    237000            1.00           140  Female      No   15           1
20   48       1                     582        1                55                   0     87000            1.90           121  Female      No   15           1
21   65       1                      52        0                25                   1    276000            1.30           137  Female      No   16           0
23   68       1                     220        0                35                   1    289000            0.90           140    Male     Yes   20           1
24   53       0                      63        1                60                   0    368000            0.80           135    Male      No   22           0
25   75       0                     582        1                30                   1    263358            1.83           134  Female      No   23           1
26   80       0                     148        1                38                   0    149000            1.90           144    Male     Yes   23           1
28   70       0                     122        1                45                   1    284000            1.30           136    Male     Yes   26           1
30   82       0                      70        1                30                   0    200000            1.20           132    Male     Yes   26           1
31   94       0                     582        1                38                   1    263358            1.83           134    Male      No   27           1
32   85       0                      23        0                45                   0    360000            3.00           132    Male      No   28           1
33   50       1                     249        1                35                   1    319000            1.00           128  Femmale     No   28           1
34   50       1                     159        1                30                   0    302000            1.20           138  Female      No   29           0
35   65       0                      94        1                50                   1    188000            1.00           140    Male      No   29           1
38   82       1                     855        1                50                   1    321000            1.00           145  Female      No   30           1
39   60       0                    2656        1                30                   0    305000            2.30           137    Male      No   30           0
41   70       0                     582        0                20                   1    263358            1.83           134    Male     Yes   31           1
42   50       0                     124        1                30                   1    153000            1.20           136  Female     Yes   32           1
43   70       0                     571        1                45                   1    185000            1.20           139    Male     Yes   33           1
46   50       0                     582        1                38                   0    310000            1.90           135    Male     Yes   35           1
47   51       0                    1380        0                25                   1    271000            0.90           130    Male      No   38           1
49   80       1                     553        0                20                   1    140000            4.40           133    Male      No   41           1
52   53       1                      91        0                20                   1    418000            1.40           139  Female      No   43           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of smoking is converted into categorical values. Here 0 = “No” (doesn’t smoke) and 1 = “Yes”(smoke).

❖ **Convert Numerical DEATH_EVENT into Categorical DEATH_EVENT**

Code:

```
dataset1 <- na.omit(dataset)

dataset1$DEATH_EVENT<-factor(dataset1$DEATH_EVENT,levels =c(0,1),labels =
c("No","Yes"))

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> dataset1$DEATH_EVENT<-factor(dataset1$DEATH_EVENT,levels =c(0,1),labels = c("No","Yes"))
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75       0                     582        0                20                   1    265000            1.90          130    Male       0    4         Yes
2   55       0                    7861        0                38                   0    263358            1.10          136    Male       0    6         Yes
3   65       0                     146        0                20                   0    162000            1.30          129    Male       1    7         Yes
4   50       1                     111        0                20                   0    210000            1.90          137    Male       0    7         Yes
5   65       1                     160        1                20                   0    327000            2.70          116  Female       0    8         Yes
6   90       1                      47        0                40                   1    204000            2.10          132    Male       1    8         Yes
9   65       0                     157        0                65                   0    263358            1.50          138  Female       0   10         Yes
11  75       1                      81        0                38                   1    368000            4.00          131    Male       1   10         Yes
13  45       1                     981        0                30                   0    136000            1.10          137    Male       0   11         Yes
14  50       1                     168        0                38                   1    276000            1.10          137    Male       0   11         Yes
15  49       1                      80        0                30                   1    427000            1.00          138  Female       0   12          No
16  82       1                     379        0                50                   0     47000            1.30          136    Male       0   13         Yes
19  70       1                     125        0                25                   1    237000            1.00          140  Female       0   15         Yes
20  48       1                     582        1                55                   0     87000            1.90          121  Female       0   15         Yes
21  65       1                      52        0                25                   1    276000            1.30          137  Female       0   16          No
23  68       1                     220        0                35                   1    289000            0.90          140    Male       1   20         Yes
24  53       0                      63        1                60                   0    368000            0.80          135    Male       0   22          No
25  75       0                     582        1                30                   1    263358            1.83          134  Female       0   23         Yes
26  80       0                     148        1                38                   0    149000            1.90          144    Male       1   23         Yes
28  70       0                     122        1                45                   1    284000            1.30          136    Male       1   26         Yes
30  82       0                      70        1                30                   0    200000            1.20          132    Male       1   26         Yes
31  94       0                     582        1                38                   1    263358            1.83          134    Male       0   27         Yes
32  85       0                      23        0                45                   0    360000            3.00          132    Male       0   28         Yes
33  50       1                     249        1                35                   1    319000            1.00          128 Femmale       0   28         Yes
34  50       1                     159        1                30                   0    302000            1.20          138  Female       0   29          No
35  65       0                      94        1                50                   1    188000            1.00          140    Male       0   29         Yes
38  82       1                     855        1                50                   1    321000            1.00          145  Female       0   30         Yes
39  60       0                    2656        1                30                   0    305000            2.30          137    Male       0   30          No
41  70       0                     582        0                20                   1    263358            1.83          134    Male       1   31         Yes
42  50       0                     124        1                30                   1    153000            1.20          136  Female       1   32         Yes
43  70       0                     571        1                45                   1    185000            1.20          139    Male       1   33         Yes
46  50       0                     582        1                38                   0    310000            1.90          135    Male       1   35         Yes
47  51       0                    1380        0                25                   1    271000            0.90          130    Male       0   38         Yes
49  80       1                     553        0                20                   1    140000            4.40          133    Male       0   41         Yes
52  53       1                      91        0                20                   1    418000            1.40          139  Female       0   43         Yes
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then, numerical values of DEATH_EVENT is converted into categorical values. Here 0 = “No” (no DEATH_EVENT) and 1 = “Yes”(DEATH_EVENT happened).

NORMALIZATION METHOD (MIN MAX NORMALIZATION)

❖ Apply Normalization Method for age Attribute

Code:

```
dataset1 <- na.omit(dataset)

numerical_age <- as.numeric(dataset1$age)

normalized_age <- (numerical_age - min(numerical_age)) / (max(numerical_age) - min(numerical_age))

print(min(numerical_age))

print(max(numerical_age))

dataset1$age <- normalized_age

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> numerical_age <- as.numeric(dataset1$age)
> normalized_age <- (numerical_age - min(numerical_age)) / (max(numerical_age) - min(numerical_age))
> print(min(numerical_age))
[1] 40
> print(max(numerical_age))
[1] 170
> dataset1$age <- normalized_age
> dataset1
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
1	0.269230769	0	582	0	20	1	265000	1.90	130	Male	0	4	1
2	0.115384615	0	7861	0	38	0	263358	1.10	136	Male	0	6	1
3	0.192307692	0	146	0	20	0	162000	1.30	129	Male	1	7	1
4	0.076923077	1	111	0	20	0	210000	1.90	137	Male	0	7	1
5	0.192307692	1	160	1	20	0	327000	2.70	116	Female	0	8	1
6	0.384615385	1	47	0	40	1	204000	2.10	132	Male	1	8	1
9	0.192307692	0	157	0	65	0	263358	1.50	138	Female	0	10	1
11	0.269230769	1	81	0	38	1	368000	4.00	131	Male	1	10	1
13	0.038461538	1	981	0	30	0	136000	1.10	137	Male	0	11	1
14	0.076923077	1	168	0	38	1	276000	1.10	137	Male	0	11	1
15	0.069230769	1	80	0	30	1	427000	1.00	138	Female	0	12	0
16	0.323076923	1	379	0	50	0	47000	1.30	136	Male	0	13	1
19	0.230769231	1	125	0	25	1	237000	1.00	140	Female	0	15	1
20	0.061538462	1	582	1	55	0	87000	1.90	121	Female	0	15	1
21	0.192307692	1	52	0	25	1	276000	1.30	137	Female	0	16	0
23	0.215384615	1	220	0	35	1	289000	0.90	140	Male	1	20	1
24	0.100000000	0	63	1	60	0	368000	0.80	135	Male	0	22	0
25	0.269230769	0	582	1	30	1	263358	1.83	134	Female	0	23	1
26	0.307692308	0	148	1	38	0	149000	1.90	144	Male	1	23	1
28	0.230769231	0	122	1	45	1	284000	1.30	136	Male	1	26	1
30	0.323076923	0	70	1	30	0	200000	1.20	132	Male	1	26	1
31	0.415384615	0	582	1	38	1	263358	1.83	134	Male	0	27	1
32	0.346153846	0	23	0	45	0	360000	3.00	132	Male	0	28	1
33	0.076923077	1	249	1	35	1	319000	1.00	128	Femmale	0	28	1
34	0.076923077	1	159	1	30	0	302000	1.20	138	Female	0	29	0
35	0.192307692	0	94	1	50	1	188000	1.00	140	Male	0	29	1
38	0.323076923	1	855	1	50	1	321000	1.00	145	Female	0	30	1
39	0.153846154	0	2656	1	30	0	305000	2.30	137	Male	0	30	0
41	0.230769231	0	582	0	20	1	263358	1.83	134	Male	1	31	1
42	0.076923077	0	124	1	30	1	153000	1.20	136	Female	1	32	1

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then the age values are stored in numerical_age. After that, values of age are normalized using (numerical_age - min(numerical_age)) / (max(numerical_age) - min(numerical_age). Here the min value of age attribute is 40 and max value of age attribute is 170. Finally, the values of age is replaced with the normalized value and the dataset1 is printed.

❖ **Apply Normalization Method for ejection_fraction Attribute**

Code:

```
dataset1 <- na.omit(dataset)

numerical_ejection_fraction <- as.numeric(dataset1$ejection_fraction)

normalized_ejection_fraction <- (numerical_ejection_fraction - min(numerical_ejection_fraction)) / (max(numerical_ejection_fraction) - min(numerical_ejection_fraction))

print(min(numerical_ejection_fraction))

print(max(numerical_ejection_fraction))

dataset1$ejection_fraction <- normalized_ejection_fraction

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> numerical_ejection_fraction <- as.numeric(dataset1$ejection_fraction)
> normalized_ejection_fraction <- (numerical_ejection_fraction - min(numerical_ejection_fraction)) / (max(numerical_ejection_fraction) - min(numerical
_ejection_fraction))
> print(min(numerical_ejection_fraction))
[1] 17
> print(max(numerical_ejection_fraction))
[1] 80
> dataset1$ejection_fraction <- normalized_ejection_fraction
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking  time DEATH_EVENT
1   75       0                     582        0        0.04761905                   1    265000            1.90          130    Male       0     4           1
2   55       0                    7861        0        0.33333333                   0    263358            1.10          136    Male       0     6           1
3   65       0                     146        0        0.04761905                   0    162000            1.30          129    Male       1     7           1
4   50       1                     111        0        0.04761905                   0    210000            1.90          137    Male       0     7           1
5   65       1                     160        1        0.04761905                   0    327000            2.70          116  Female       0     8           1
6   90       1                      47        0        0.36507937                   1    204000            2.10          132    Male       1     8           1
9   65       0                     157        0        0.76190476                   0    263358            1.50          138  Female       0    10           1
11  75       1                      81        0        0.33333333                   1    368000            4.00          131    Male       1    10           1
13  45       1                     981        0        0.20634921                   0    136000            1.10          137    Male       0    11           1
14  50       1                     168        0        0.33333333                   1    276000            1.10          137    Male       0    11           1
15  49       1                      80        0        0.20634921                   1    427000            1.00          138  Female       0    12           0
16  82       1                     379        0        0.52380952                   0     47000            1.30          136    Male       0    13           1
19  70       1                     125        0        0.12698413                   1    237000            1.00          140  Female       0    15           1
20  48       1                     582        1        0.60317460                   0     87000            1.90          121  Female       0    15           1
21  65       1                      52        0        0.12698413                   1    276000            1.30          137  Female       0    16           0
23  68       1                     220        0        0.28571429                   1    289000            0.90          140    Male       1    20           1
24  53       0                      63        1        0.68253968                   0    368000            0.80          135    Male       0    22           0
25  75       0                     582        1        0.20634921                   1    263358            1.83          134  Female       0    23           1
26  80       0                     148        1        0.33333333                   0    149000            1.90          144    Male       1    23           1
28  70       0                     122        1        0.44444444                   1    284000            1.30          136    Male       1    26           1
30  82       0                      70        1        0.20634921                   0    200000            1.20          132    Male       1    26           1
31  94       0                     582        1        0.33333333                   1    263358            1.83          134    Male       0    27           1
32  85       0                      23        0        0.44444444                   0    360000            3.00          132    Male       0    28           1
33  50       1                     249        1        0.28571429                   1    319000            1.00          128 Femmale       0    28           1
34  50       1                     159        1        0.20634921                   0    302000            1.20          138  Female       0    29           0
35  65       0                      94        1        0.52380952                   1    188000            1.00          140    Male       0    29           1
38  82       1                     855        1        0.52380952                   1    321000            1.00          145  Female       0    30           1
39  60       0                    2656        1        0.20634921                   0    305000            2.30          137    Male       0    30           0
41  70       0                     582        0        0.04761905                   1    263358            1.83          134    Male       1    31           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then the ejection_fraction values are stored in numerical_ ejection_fraction. After that, values of ejection_fraction are normalized using (numerical_ ejection_fraction - min(numerical_ ejection_fraction)) / (max(numerical_ ejection_fraction) - min(numerical_ ejection_fraction). Here the min value of ejection_fraction attribute is 17 and max value of ejection_fraction attribute is 80. Finally, the values of ejection_fraction is replaced with the normalized value and the dataset1 is printed.

❖ Apply Normalization Method for platelets Attribute

Code:

```
dataset1 <- na.omit(dataset)

numerical_platelets <- as.numeric(dataset1$platelets)

normalized_platelets <- (numerical_platelets - min(numerical_platelets)) /
(max(numerical_platelets) - min(numerical_platelets))

print(min(numerical_platelets))

print(max(numerical_platelets))

dataset1$platelets <- normalized_platelets

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> numerical_platelets <- as.numeric(dataset1$platelets)
> normalized_platelets <- (numerical_platelets - min(numerical_platelets)) / (max(numerical_platelets) - min(numerical_platelets))
> print(min(numerical_platelets))
[1] 47000
> print(max(numerical_platelets))
[1] 850000
> dataset1$platelets <- normalized_platelets
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure  platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75       0                     582        0                20                   1 0.27148194            1.90          130    Male       0    4           1
2   55       0                    7861        0                38                   0 0.26943715            1.10          136    Male       0    6           1
3   65       0                     146        0                20                   0 0.14321295            1.30          129    Male       1    7           1
4   50       1                     111        0                20                   0 0.20298879            1.90          137    Male       0    7           1
5   65       1                     160        1                20                   0 0.34869240            2.70          116  Female       0    8           1
6   90       1                      47        0                40                   1 0.19551681            2.10          132    Male       1    8           1
9   65       0                     157        0                65                   0 0.26943715            1.50          138  Female       0   10           1
11  75       1                      81        0                38                   1 0.39975093            4.00          131    Male       1   10           1
13  45       1                     981        0                30                   0 0.11083437            1.10          137    Male       0   11           1
14  50       1                     168        0                38                   1 0.28518057            1.10          137    Male       0   11           1
15  49       1                      80        0                30                   1 0.47322540            1.00          138  Female       0   12           0
16  82       1                     379        0                50                   0 0.00000000            1.30          136    Male       0   13           1
19  70       1                     125        0                25                   1 0.23661270            1.00          140  Female       0   15           1
20  48       1                     582        1                55                   0 0.04981320            1.90          121  Female       0   15           1
21  65       1                      52        0                25                   1 0.28518057            1.30          137  Female       0   16           0
23  68       1                     220        0                35                   1 0.30136986            0.90          140    Male       1   20           1
24  53       0                      63        1                60                   0 0.39975093            0.80          135    Male       0   22           0
25  75       0                     582        1                30                   1 0.26943715            1.83          134  Female       0   23           1
26  80       0                     148        1                38                   0 0.12702366            1.90          144    Male       1   23           1
28  70       0                     122        1                45                   1 0.29514321            1.30          136    Male       1   26           1
30  82       0                      70        1                30                   0 0.19053549            1.20          132    Male       1   26           1
31  94       0                     582        1                38                   1 0.26943715            1.83          134    Male       0   27           1
32  85       0                      23        0                45                   0 0.38978829            3.00          132    Male       0   28           1
33  50       1                     249        1                35                   1 0.33872976            1.00          128 Femmale       0   28           1
34  50       1                     159        1                30                   0 0.31755915            1.20          138  Female       0   29           0
35  65       0                      94        1                50                   1 0.17559153            1.00          140    Male       0   29           1
38  82       1                     855        1                50                   1 0.34122042            1.00          145  Female       0   30           1
39  60       0                    2656        1                30                   0 0.32129514            2.30          137    Male       0   30           0
41  70       0                     582        0                20                   1 0.26943715            1.83          134    Male       1   31           1
42  50       0                     124        1                30                   1 0.13200498            1.20          136  Female       1   32           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then the platelets values are stored in numerical_ platelets. After that, values of platelets are normalized using (numerical_ platelets - min(numerical_platelets)) / (max(numerical_platelets) - min(numerical_ platelets). Here the min value of platelets attribute is 47000 and max value of platelets attribute is 850000. Finally, the values of platelets is replaced with the normalized value and the dataset1 is printed.

❖ **Apply Normalization Method for serum_creatinine Attribute**

Code:

```
dataset1 <- na.omit(dataset)

numerical_serum_creatinine <- as.numeric(dataset1$serum_creatinine)

normalized_serum_creatinine <- (numerical_serum_creatinine - min(numerical_serum_creatinine)) / (max(numerical_serum_creatinine) - min(numerical_serum_creatinine))

print(min(numerical_serum_creatinine))

print(max(numerical_serum_creatinine))

dataset1$serum_creatinine <- normalized_serum_creatinine

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> numerical_serum_creatinine <- as.numeric(dataset1$serum_creatinine)
> normalized_serum_creatinine <- (numerical_serum_creatinine - min(numerical_serum_creatinine)) / (max(numerical_serum_creatinine) - min(numerical_serum_creatinine))
> print(min(numerical_serum_creatinine))
[1] 0.6
> print(max(numerical_serum_creatinine))
[1] 6.8
> dataset1$serum_creatinine <- normalized_serum_creatinine
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking time DEATH_EVENT
1   75       0                     582        0                20                   1    265000       0.20967742          130    Male       0    4           1
2   55       0                    7861        0                38                   0    263358       0.08064516          136    Male       0    6           1
3   65       0                     146        0                20                   0    162000       0.11290323          129    Male       1    7           1
4   50       1                     111        0                20                   0    210000       0.20967742          137    Male       0    7           1
5   65       1                     160        1                20                   0    327000       0.33870968          116  Female       0    8           1
6   90       1                      47        0                40                   1    204000       0.24193548          132    Male       1    8           1
9   65       0                     157        0                65                   0    263358       0.14516129          138  Female       0   10           1
11  75       1                      81        0                38                   1    368000       0.54838710          131    Male       1   10           1
13  45       1                     981        0                30                   0    136000       0.08064516          137    Male       0   11           1
14  50       1                     168        0                38                   1    276000       0.08064516          137    Male       0   11           1
15  49       1                      80        0                30                   1    427000       0.06451613          138  Female       0   12           0
16  82       1                     379        0                50                   0     47000       0.11290323          136    Male       0   13           1
19  70       1                     125        0                25                   1    237000       0.06451613          140  Female       0   15           1
20  48       1                     582        1                55                   0     87000       0.20967742          121  Female       0   15           1
21  65       1                      52        0                25                   1    276000       0.11290323          137  Female       0   16           0
23  68       1                     220        0                35                   1    289000       0.04838710          140    Male       1   20           1
24  53       0                      63        1                60                   0    368000       0.03225806          135    Male       0   22           0
25  75       0                     582        1                30                   1    263358       0.19838710          134  Female       0   23           1
26  80       0                     148        1                38                   0    149000       0.20967742          144    Male       1   23           1
28  70       0                     122        1                45                   1    284000       0.11290323          136    Male       1   26           1
30  82       0                      70        1                30                   0    200000       0.09677419          132    Male       1   26           1
31  94       0                     582        1                38                   1    263358       0.19838710          134    Male       0   27           1
32  85       0                      23        0                45                   0    360000       0.38709677          132    Male       0   28           1
33  50       1                     249        1                35                   1    319000       0.06451613          128 Femmale       0   28           1
34  50       1                     159        1                30                   0    302000       0.09677419          138  Female       0   29           0
35  65       0                      94        1                50                   1    188000       0.06451613          140    Male       0   29           1
38  82       1                     855        1                50                   1    321000       0.06451613          145  Female       0   30           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then the serum_creatinine values are stored in numerical_ serum_creatinine. After that, values of serum_creatinine are normalized using (numerical_ serum_creatinine - min(numerical_ serum_creatinine)) / (max(numerical_ serum_creatinine) - min(numerical_ serum_creatinine). Here the min value of serum_creatinine attribute is 0.6 and max value of serum_creatinine attribute is 6.8. Finally, the values of serum_creatinine is replaced with the normalized value and the dataset1 is printed.

❖ **Apply Normalization Method for serum_sodium Attribute**

Code:

```
dataset1 <- na.omit(dataset)

numerical_serum_sodium <- as.numeric(dataset1$serum_sodium)

normalized_serum_sodium <- (numerical_serum_sodium - min(numerical_serum_sodium)) / (max(numerical_serum_sodium) - min(numerical_serum_sodium))

print(min(numerical_serum_sodium))

print(max(numerical_serum_sodium))

dataset1$serum_sodium <- normalized_serum_sodium

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> numerical_serum_sodium <- as.numeric(dataset1$serum_sodium)
> normalized_serum_sodium <- (numerical_serum_sodium - min(numerical_serum_sodium)) / (max(numerical_serum_sodium) - min(numerical_serum_sodium))
> print(min(numerical_serum_sodium))
[1] 116
> print(max(numerical_serum_sodium))
[1] 146
> dataset1$serum_sodium <- normalized_serum_sodium
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium    sex smoking time DEATH_EVENT
1   75       0                     582        0                20                   1    265000            1.90    0.4666667   Male       0    4           1
2   55       0                    7861        0                38                   0    263358            1.10    0.6666667   Male       0    6           1
3   65       0                     146        0                20                   0    162000            1.30    0.4333333   Male       1    7           1
4   50       1                     111        0                20                   0    210000            1.90    0.7000000   Male       0    7           1
5   65       1                     160        1                20                   0    327000            2.70    0.0000000 Female       0    8           1
6   90       1                      47        0                40                   1    204000            2.10    0.5333333   Male       1    8           1
9   65       0                     157        0                65                   0    263358            1.50    0.7333333 Female       0   10           1
11  75       1                      81        0                38                   1    368000            4.00    0.5000000   Male       1   10           1
13  45       1                     981        0                30                   0    136000            1.10    0.7000000   Male       0   11           1
14  50       1                     168        0                38                   1    276000            1.10    0.7000000   Male       0   11           1
15  49       1                      80        0                30                   1    427000            1.00    0.7333333 Female       0   12           0
16  82       1                     379        0                50                   0     47000            1.30    0.6666667   Male       0   13           1
19  70       1                     125        0                25                   1    237000            1.00    0.8000000 Female       0   15           1
20  48       1                     582        1                55                   0     87000            1.90    0.1666667 Female       0   15           1
21  65       1                      52        0                25                   1    276000            1.30    0.7000000 Female       0   16           0
23  68       1                     220        0                35                   1    289000            0.90    0.8000000   Male       1   20           1
24  53       0                      63        1                60                   0    368000            0.80    0.6333333   Male       0   22           0
25  75       0                     582        1                30                   1    263358            1.83    0.6000000 Female       0   23           1
26  80       0                     148        1                38                   0    149000            1.90    0.9333333   Male       1   23           1
28  70       0                     122        1                45                   1    284000            1.30    0.6666667   Male       1   26           1
30  82       0                      70        1                30                   0    200000            1.20    0.5333333   Male       1   26           1
31  94       0                     582        1                38                   1    263358            1.83    0.6000000   Male       0   27           1
32  85       0                      23        0                45                   0    360000            3.00    0.5333333   Male       0   28           1
33  50       1                     249        1                35                   1    319000            1.00    0.4000000 Femmale       0   28           1
34  50       1                     159        1                30                   0    302000            1.20    0.7333333 Female       0   29           0
35  65       0                      94        1                50                   1    188000            1.00    0.8000000   Male       0   29           1
38  82       1                     855        1                50                   1    321000            1.00    0.9666667 Female       0   30           1
39  60       0                    2656        1                30                   0    305000            2.30    0.7000000   Male       0   30           0
41  70       0                     582        0                20                   1    263358            1.83    0.6000000   Male       1   31           1
42  50       0                     124        1                30                   1    153000            1.20    0.6666667 Female       1   32           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then the serum_sodium values are stored in numerical_serum_sodium. After that, values of serum_sodium are normalized using (numerical_serum_sodium - min(numerical_serum_sodium)) / (max(numerical_ serum_sodium) - min(numerical_serum_sodium). Here the min value of serum_sodium attribute is 116 and max value of serum_sodium attribute is 146. Finally, the values of serum_sodium is replaced with the normalized value and the dataset1 is printed.

❖ **Apply Normalization Method for time Attribute**

Code:

```
dataset1 <- na.omit(dataset)

numerical_time <- as.numeric(dataset1$time)

normalized_time <- (numerical_time - min(numerical_time)) / (max(numerical_time) - min(numerical_time))

print(min(numerical_time))

print(max(numerical_time))

dataset1$time <- normalized_time

dataset1
```

Output:

```
> dataset1 <- na.omit(dataset)
> numerical_time <- as.numeric(dataset1$time)
> normalized_time <- (numerical_time - min(numerical_time)) / (max(numerical_time) - min(numerical_time))
> print(min(numerical_time))
[1] 4
> print(max(numerical_time))
[1] 186
> dataset1$time <- normalized_time
> dataset1
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium     sex smoking       time DEATH_EVENT
1   75       0                     582        0                20                   1    265000            1.90          130    Male       0 0.00000000           1
2   55       0                    7861        0                38                   0    263358            1.10          136    Male       0 0.01098901           1
3   65       0                     146        0                20                   0    162000            1.30          129    Male       1 0.01648352           1
4   50       1                     111        0                20                   0    210000            1.90          137    Male       0 0.01648352           1
5   65       1                     160        1                20                   0    327000            2.70          116  Female       0 0.02197802           1
6   90       1                      47        0                40                   1    204000            2.10          132    Male       1 0.02197802           1
9   65       0                     157        0                65                   0    263358            1.50          138  Female       0 0.03296703           1
11  75       1                      81        0                38                   1    368000            4.00          131    Male       1 0.03296703           1
13  45       1                     981        0                30                   0    136000            1.10          137    Male       0 0.03846154           1
14  50       1                     168        0                38                   1    276000            1.10          137    Male       0 0.03846154           1
15  49       1                      80        0                30                   1    427000            1.00          138  Female       0 0.04395604           0
16  82       1                     379        0                50                   0     47000            1.30          136    Male       0 0.04945055           1
19  70       1                     125        0                25                   1    237000            1.00          140  Female       0 0.06043956           1
20  48       1                     582        1                55                   0     87000            1.90          121  Female       0 0.06043956           1
21  65       1                      52        0                25                   1    276000            1.30          137  Female       0 0.06593407           0
23  68       1                     220        0                35                   1    289000            0.90          140    Male       1 0.08791209           1
24  53       0                      63        1                60                   0    368000            0.80          135    Male       0 0.09890110           0
25  75       0                     582        1                30                   1    263358            1.83          134  Female       0 0.10439560           1
26  80       0                     148        1                38                   0    149000            1.90          144    Male       1 0.10439560           1
28  70       0                     122        1                45                   1    284000            1.30          136    Male       1 0.12087912           1
30  82       0                      70        1                30                   0    200000            1.20          132    Male       1 0.12087912           1
31  94       0                     582        1                38                   1    263358            1.83          134    Male       0 0.12637363           1
32  85       0                      23        0                45                   0    360000            3.00          132    Male       0 0.13186813           1
33  50       1                     249        1                35                   1    319000            1.00          128 Femmale       0 0.13186813           1
34  50       1                     159        1                30                   0    302000            1.20          138  Female       0 0.13736264           0
35  65       0                      94        1                50                   1    188000            1.00          140    Male       0 0.13736264           1
38  82       1                     855        1                50                   1    321000            1.00          145  Female       0 0.14285714           1
39  60       0                    2656        1                30                   0    305000            2.30          137    Male       0 0.14285714           0
41  70       0                     582        0                20                   1    263358            1.83          134    Male       1 0.14835165           1
42  50       0                     124        1                30                   1    153000            1.20          136  Female       1 0.15384615           1
43  70       0                     571        1                45                   1    185000            1.20          139    Male       1 0.15934066           1
```

At first, all instances containing at least one NA values are discarded using the na.omit function. The output was saved in dataset1, after the removal of the NA values. Then the time values are stored in numerical_time. After that, values of time are normalized using (numerical_time - min(numerical_ time)) / (max(numerical_ time) - min(numerical_ time). Here the min value of time attribute is 4 and max value of time attribute is 186. Finally, the values of time is replaced with the normalized value and the dataset1 is printed.

OUTLIERS

❖ **Replacing age Outliers using Mean Value**

Code:

```
ageBoxplot <- boxplot(dataset$age,main=" Age Distribution ", ylab="age",col="lightgreen")
outliers <- ageBoxplot$out
cat("Outliers are", outliers)
ageMean <- mean(dataset$age, na.rm = TRUE)
outlierPositions <- match(outliers, dataset$age)
dataset$age[outlierPositions] <- as.integer (ageMean)
dataset
```

Output:

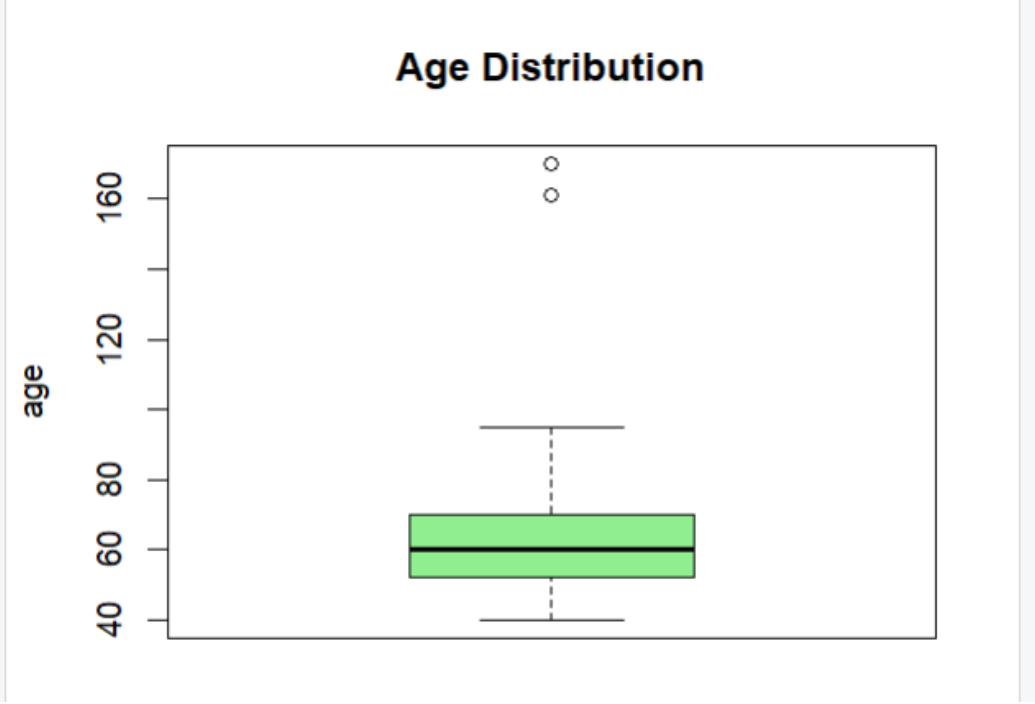


Fig: Age Distribution

Here the values belong to outside of the box are Outliers.

```
> ageBoxplot <- boxplot(dataset$age,main=" Age Distribution ", ylab="age",col="lightgreen")
> outliers <- ageBoxplot$out
> cat("Outliers are", outliers)
Outliers are 161 170> ageMean <- mean(dataset$age, na.rm = TRUE)
> outlierPositions <- match(outliers, dataset$age)
> dataset$age[outlierPositions] <- as.integer (ageMean)
> dataset
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
1	75.00	0	582	0	20	1	265000	1.90	130	Male	0	4	1
2	55.00	0	7861	0	38	0	263358	1.10	136	Male	0	6	1
3	65.00	0	146	0	20	0	162000	1.30	129	Male	1	7	1
4	50.00	1	111	0	20	0	210000	1.90	137	Male	0	7	1
5	65.00	1	160	1	20	0	327000	2.70	116	Female	0	8	1
6	90.00	1	47	0	40	1	204000	2.10	132	Male	1	8	1
7	75.00	1	246	NA	15	0	127000	1.20	137	Male	0	10	1
8	60.00	NA	315	1	60	0	454000	1.10	131	Male	1	10	1
9	65.00	0	157	0	65	0	263358	1.50	138	Female	0	10	1
10	NA	1	123	0	35	1	388000	9.40	133	Maleee	1	10	1
11	75.00	1	81	0	38	1	368000	4.00	131	Male	1	10	1
12	62.00	0	231	0	25	1	NA	0.90	140	Male	NA	10	1
13	45.00	1	981	0	30	0	136000	1.10	137	Male	0	11	1
14	50.00	1	168	0	38	1	276000	1.10	137	Male	0	11	1
15	49.00	1	80	0	30	1	427000	1.00	138	Female	0	12	0
16	82.00	1	379	0	50	0	47000	1.30	136	Male	0	13	1
17	87.00	1	NA	0	38	0	262000	0.90	140	Male	0	14	1
18	45.00	0	582	0	14	0	166000	0.80	127	Male	NA	14	1
19	70.00	1	125	0	25	1	237000	1.00	140	Female	0	15	1
20	48.00	1	582	1	55	0	87000	1.90	121	Female	0	15	1
21	65.00	1	52	0	25	1	276000	1.30	137	Female	0	16	0
22	65.00	NA	128	1	30	1	297000	1.60	136	Female	0	20	1
23	68.00	1	220	0	35	1	289000	0.90	140	Male	1	20	1
24	53.00	0	63	1	60	0	368000	0.80	135	Male	0	22	0
25	75.00	0	582	1	30	1	263358	1.83	134	Female	0	23	1
26	80.00	0	148	1	38	0	149000	1.90	144	Male	1	23	1

Showing 1 to 27 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
27	95.00	1	112	0	40	1	NA	1.00	138	Female	0	24	1
28	70.00	0	122	1	45	1	284000	1.30	136	Male	1	26	1
29	NA	1	60	0	38	0	153000	5.80	134	Male	0	26	1
30	82.00	0	70	1	30	0	200000	1.20	132	Male	1	26	1
31	94.00	0	582	1	38	1	263358	1.83	134	Male	0	27	1
32	85.00	0	23	0	45	0	360000	3.00	132	Male	0	28	1
33	50.00	1	249	1	35	1	319000	1.00	128	Femmale	0	28	1
34	50.00	1	159	1	30	0	302000	1.20	138	Female	0	29	0
35	65.00	0	94	1	50	1	188000	1.00	140	Male	0	29	1
36	NA	NA	582	1	35	0	228000	3.50	134	Male	0	30	1
37	90.00	1	60	NA	50	0	226000	1.00	134	Male	0	30	1
38	82.00	1	855	1	50	1	321000	1.00	145	Female	0	30	1
39	60.00	0	2656	1	30	0	305000	2.30	137	Male	0	30	0
40	60.00	NA	235	1	38	0	329000	3.00	142	Female	0	30	1
41	70.00	0	582	0	20	1	263358	1.83	134	Male	1	31	1
42	50.00	0	124	1	30	1	153000	1.20	136	Female	1	32	1
43	70.00	0	571	1	45	1	185000	1.20	139	Male	1	33	1
44	72.00	0	127	1	50	1	218000	1.00	134	Male	NA	33	0
45	60.00	1	588	1	60	0	NA	1.10	142	Female	0	33	1
46	50.00	0	582	1	38	0	310000	1.90	135	Male	1	35	1
47	51.00	0	1380	0	25	1	271000	0.90	130	Male	0	38	1
48	60.00	NA	582	1	38	1	451000	0.60	138	Male	1	40	1
49	80.00	1	553	0	20	1	140000	4.40	133	Male	0	41	1
50	NA	1	129	0	30	0	395000	1.00	140	Female	0	42	1
51	68.00	1	NA	0	25	1	166000	1.00	138	Male	0	43	1
52	53.00	1	91	0	20	1	418000	1.40	139	Female	0	43	1

Showing 27 to 53 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
53	60.00	0	3964	1	62	0	263358	6.80	146	Female	0	43	1
54	70.00	1	69	1	50	1	351000	1.00	134	Female	0	44	1
55	60.00	1	260	1	38	0	255000	2.20	132	Female	1	45	1
56	95.00	1	371	0	30	0	461000	2.00	132	Male	0	50	1
57	70.00	1	75	0	35	0	223000	2.70	138	Male	1	54	0
58	60.00	1	607	0	40	0	216000	0.60	138	Male	1	54	0
59	49.00	0	789	0	20	1	319000	1.10	136	Male	1	55	1
60	72.00	0	364	1	20	1	254000	1.30	136	Male	1	59	1
61	45.00	0	7702	1	25	1	390000	1.00	139	Male	0	60	1
62	50.00	0	318	0	40	1	216000	2.30	131	Female	0	60	1
63	55.00	0	109	0	35	0	254000	1.10	139	Male	1	60	0
64	45.00	0	*NA*	0	35	0	385000	1.00	145	Male	0	61	1
65	45.00	0	582	0	80	0	263358	1.18	137	Female	0	63	0
66	60.00	0	68	0	20	0	119000	2.90	127	Male	1	64	1
67	*NA*	1	250	1	15	0	213000	1.30	136	Female	0	65	1
68	72.00	1	110	0	25	0	274000	1.00	140	Male	1	65	1
69	70.00	0	161	0	25	0	244000	1.20	142	Female	0	66	1
70	65.00	0	113	1	25	0	497000	1.83	135	Male	0	67	1
71	41.00	0	148	0	40	0	374000	0.80	140	Male	1	68	0
72	58.00	0	582	1	35	0	122000	0.90	139	Male	1	71	0
73	85.00	0	5882	0	35	0	243000	1.00	132	Male	1	72	1
74	65.00	0	224	1	50	0	149000	1.30	137	Male	1	72	0
75	69.00	0	582	0	20	0	*NA*	1.20	134	Male	1	73	1
76	60.00	1	47	0	20	0	204000	0.70	139	Male	1	73	1
77	70.00	0	92	0	60	1	317000	0.80	140	Female	1	74	0
78	42.00	0	102	1	40	0	237000	1.20	140	Male	0	74	0

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
79	75.00	1	203	1	38	1	283000	0.60	131	Male	1	74	0
80	55.00	0	336	0	45	1	324000	0.90	140	Female	0	74	0
81	70.00	0	69	0	40	0	293000	1.70	136	Female	0	75	0
82	67.00	0	582	0	50	0	263358	1.18	137	Male	1	76	0
83	60.00	1	76	1	25	0	196000	2.50	132	Female	0	77	1
84	79.00	1	55X	0	50	1	172000	1.80	133	Male	0	78	0
85	59.00	1	280	1	25	1	302000	1.00	141	Female	0	78	1
86	51.00	0	78	0	50	0	406000	0.70	140	Male	0	79	0
87	55.00	0	*NA*	0	35	1	173000	1.10	137	Male	0	79	0
88	65.00	1	68	1	60	1	304000	0.80	140	Male	0	79	0
89	44.00	0	84	1	40	1	235000	0.70	139	Male	0	79	0
90	57.00	1	115	0	25	1	181000	1.10	144	Male	0	79	0
91	70.00	0	66	1	45	0	249000	0.80	136	Male	1	80	0
92	60.00	0	897	1	45	0	297000	1.00	133	Male	0	80	0
93	42.00	0	582	0	60	0	263358	1.18	137	Female	0	82	0
94	60.00	1	154	0	25	0	210000	1.70	135	Male	0	82	1
95	58.00	0	144	1	38	1	327000	0.70	142	Female	0	83	0
96	58.00	1	133	0	60	1	219000	1.00	141	Male	0	83	0
97	63.00	1	514	1	25	1	254000	1.30	134	Male	0	83	0
98	70.00	1	59	0	60	0	255000	1.10	136	Female	0	85	0
99	60.00	1	156	1	25	1	318000	1.20	137	Female	0	85	0
100	63.00	1	61	1	40	0	*NA*	1.10	140	Female	0	86	0
101	65.00	1	305	0	25	0	298000	1.10	141	Male	0	87	0
102	75.00	0	582	0	45	1	263358	1.18	137	Male	0	87	0
103	80.00	0	898	0	25	0	149000	1.10	144	Male	1	87	0
104	42.00	0	5209	0	30	0	226000	1.00	140	Male	1	87	0

Showing 79 to 105 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
105	60.00	0	53	0	50	1	286000	2.30	143	Female	0	87	0
106	72.00	1	328	0	30	1	621000	1.70	138	Female	1	88	1
107	55.00	0	748	0	45	0	263000	1.30	137	Male	0	88	0
108	45.00	1	1876	1	35	0	226000	0.90	138	Male	0	88	0
109	63.00	0	936	0	38	0	304000	1.10	133	Male	1	88	0
110	45.00	0	292	1	35	0	850000	1.30	142	Male	1	88	0
111	85.00	0	129	0	60	0	306000	1.20	132	Male	1	90	1
112	55.00	0	60	0	35	0	228000	1.20	135	Male	1	90	0
113	50.00	0	369	1	25	0	252000	1.60	136	Male	0	90	0
114	70.00	1	143	0	60	0	351000	1.30	137	Female	0	90	1
115	60.00	1	754	1	40	1	328000	1.20	126	Male	0	91	0
116	58.00	1	400	0	40	0	164000	1.00	139	Female	0	91	0
117	60.00	1	96	1	60	1	271000	0.70	136	Female	0	94	0
118	85.00	1	102	0	60	0	507000	3.20	138	Female	0	94	0
119	65.00	1	113	1	60	1	203000	0.90	140	Female	0	94	0
120	86.00	0	582	0	38	0	263358	1.83	134	Female	0	95	1
121	60.00	1	737	0	60	1	210000	1.50	135	Male	1	95	0
122	66.00	1	68	1	38	1	162000	1.00	136	Female	0	95	0
123	60.00	0	96	1	38	0	*NA*	0.75	140	Female	0	95	0
124	60.00	1	582	0	30	1	127000	0.90	145	Female	0	95	0
125	60.00	0	582	0	40	0	217000	3.70	134	Male	0	96	1
126	43.00	1	358	0	50	0	237000	1.30	135	Female	0	97	0
127	46.00	0	168	1	17	1	271000	2.10	124	Female	0	100	1
128	58.00	1	200	*NA*	60	0	300000	0.80	137	Female	0	104	0
129	61.00	0	248	0	30	1	267000	0.70	136	Male	1	104	0
130	53.00	1	270	1	35	0	227000	3.40	145	Male	0	105	0

Showing 105 to 130 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
131	53.00	1	1808	0	60	1	249000	0.70	138	Male	1	106	0
132	60.00	1	1082	1	45	0	250000	6.10	131	Male	0	107	0
133	46.00	0	719	0	40	1	263358	1.18	137	Female	0	107	0
134	63.00	0	193	0	60	1	295000	1.30	145	Male	1	107	0
135	81.00	0	4540	0	35	0	231000	1.18	137	Male	1	107	0
136	75.00	0	582	0	40	0	263358	1.18	137	Male	0	107	0
137	65.00	1	59	1	60	0	172000	0.90	137	Female	0	107	0
138	68.00	1	646	0	25	0	305000	2.10	130	Male	0	108	0
139	62.00	0	281	1	35	0	221000	1.00	136	Female	0	108	0
140	50.00	0	1548	0	30	1	211000	0.80	138	Male	0	108	0
141	80.00	0	805	0	38	0	263358	1.10	134	Male	0	109	1
142	46.00	1	291	0	35	0	348000	0.90	140	Female	0	109	0
143	50.00	0	482	1	30	0	329000	0.90	132	Female	0	109	0
144	63.00	1	84	0	40	1	229000	0.90	141	Female	0	110	0
145	72.00	1	943	0	25	1	338000	1.70	139	Male	1	111	1
146	50.00	0	185	0	30	0	266000	0.70	141	Male	1	112	0
147	52.00	0	132	0	30	0	218000	0.70	136	Male	1	112	0
148	64.00	0	1610	0	60	0	242000	1.00	137	Male	0	113	0
149	75.00	1	582	0	30	0	225000	1.83	134	Male	0	113	1
150	60.00	0	2261	0	35	1	228000	0.90	136	Male	0	115	0
151	72.00	0	233	0	45	1	235000	2.50	135	Female	0	115	1
152	62.00	0	30	1	60	1	244000	0.90	139	Male	0	117	0
153	50.00	0	115	0	45	1	184000	0.90	134	Male	1	118	0
154	50.00	0	1846	1	35	0	263358	1.18	137	Male	1	119	0
155	65.00	1	335	0	35	1	235000	0.80	136	Female	0	120	0
156	60.00	1	231	1	25	0	194000	1.70	140	Male	0	120	0

Showing 131 to 157 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
157	52.00	1	58	0	35	0	277000	1.40	136	Female	0	120	0
158	50.00	0	250	0	25	0	262000	1.00	136	Male	1	120	0
159	85.00	1	910	0	50	0	235000	1.30	134	Male	0	121	0
160	59.00	1	129	0	45	1	362000	1.10	139	Male	1	121	0
161	66.00	1	72	0	40	1	242000	1.20	134	Male	0	121	0
162	45.00	1	130	0	35	0	174000	0.80	139	Male	1	121	0
163	63.00	1	582	0	40	0	448000	0.90	137	Male	1	123	0
164	50.00	1	2334	NA	35	0	75000	0.90	142	Female	0	126	1
165	45.00	0	2442	1	30	0	334000	1.10	139	Male	0	129	1
166	80.00	0	776	1	38	1	192000	1.30	135	Female	0	130	1
167	53.00	0	196	0	60	0	220000	0.70	133	Male	1	134	0
168	59.00	0	66	1	20	0	70000	2.40	134	Male	0	135	1
169	65.00	0	582	1	40	0	270000	1.00	138	Female	0	140	0
170	70.00	0	835	0	35	1	305000	0.80	133	Female	0	145	0
171	51.00	1	582	1	35	0	263358	1.50	136	Male	1	145	0
172	52.00	0	3966	0	40	0	325000	0.90	140	Male	1	146	0
173	70.00	1	171	0	60	1	176000	1.10	145	Male	1	146	0
174	50.00	1	115	0	20	0	189000	0.80	139	Male	0	146	0
175	65.00	0	198	1	35	1	281000	0.90	137	Male	1	146	0
176	60.00	1	95	0	60	0	337000	1.00	138	Male	1	146	0
177	69.00	0	1419	0	40	0	105000	1.00	135	Male	1	147	0
178	49.00	1	69	0	50	0	132000	1.00	140	Female	0	147	0
179	63.00	1	122	1	60	0	267000	1.20	145	Male	0	147	0
180	55.00	0	835	0	40	0	279000	0.70	140	Male	1	147	0
181	40.00	0	478	1	30	0	303000	0.90	136	Male	0	148	0
182	59.00	1	176	1	25	0	221000	1.00	136	Male	1	150	1

Showing 157 to 183 of 199 entries, 13 total columns

183	65.00	0	395	1	25	0	265000	1.20	136	Male	1	154	1
184	75.00	0	99	0	38	1	224000	2.50	134	Male	0	162	1
185	58.00	1	145	0	25	0	219000	1.20	137	Male	1	170	1
186	63.00	1	104	1	30	0	389000	1.50	136	Male	0	171	1
187	50.00	0	582	0	50	0	153000	0.60	134	Female	0	172	1
188	60.00	0	1896	1	25	0	365000	2.10	144	Female	0	172	1
189	60.66	1	151	1	40	1	201000	1.00	136	Female	0	172	0
190	40.00	0	244	0	45	1	275000	0.90	140	Female	0	174	0
191	80.00	0	582	1	35	0	350000	2.10	134	Male	0	174	0
192	64.00	1	62	0	60	0	309000	1.50	135	Female	0	174	0
193	50.00	1	121	1	40	0	260000	0.70	130	Male	0	175	0
194	73.00	1	231	1	30	0	160000	1.18	142	Male	1	180	0
195	45.00	0	582	0	20	1	126000	1.60	135	Male	0	180	1
196	77.00	1	418	0	45	0	223000	1.80	145	Male	0	180	1
197	45.00	0	582	1	38	1	263358	1.18	137	Female	0	185	0
198	65.00	0	167	0	30	0	259000	0.80	138	Female	0	186	0
199	50.00	1	582	1	20	1	279000	1.00	134	Female	0	186	0

Showing 174 to 199 of 199 entries, 13 total columns

First, age attribute was boxplotted from the dataset to detect the outliers and saved its instance as ageBoxplot. Then, the outliers were extracted using ageBoxplot$out. After that, cat method was used to print the outliers. The mean value of the age attribute was stored in a variable ageMean using the mean method. na.rm = TRUE parameter was passed as an argument so that the NA values are ignored while calculating mean. After that, the positions of the outliers were stored in outlierPositions using the match method which returns the outlier positions in the age attribute. Then the mean value 63 was converted to integer and replaced all the outliers in the age attribute. Finally, the dataset was printed.

❖ **Replacing age Outliers using Median Value**

Code:

```
ageBoxplot <- boxplot(dataset$age,main=" Age Distribution ", ylab="age",col="lightblue")
outliers <- ageBoxplot$out
cat("Outliers are", outliers)
ageMedian <- median(dataset$age, na.rm = TRUE)
outlierPositions <- match(outliers, dataset$age)
dataset$age[outlierPositions] <- as.integer (ageMedian)
dataset
```

Output:

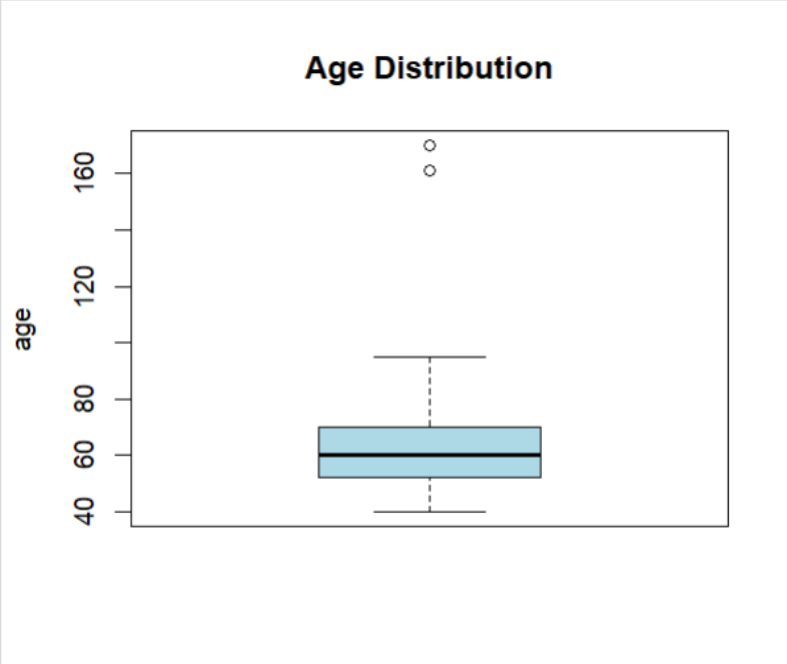


Fig: Age Distribution

Here the values belong to outside of the box are Outliers.

```
> ageBoxplot <- boxplot(dataset$age,main=" Age Distribution ", ylab="age",col="lightblue")
> outliers <- ageBoxplot$out
> cat("Outliers are", outliers)
Outliers are 161 170> ageMedian <- median(dataset$age, na.rm = TRUE)
> outlierPositions <- match(outliers, dataset$age)
> dataset$age[outlierPositions] <- as.integer (ageMedian)
> dataset
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
1	75.00	0	582	0	20	1	265000	1.90	130	Male	0	4	1
2	55.00	0	7861	0	38	0	263358	1.10	136	Male	0	6	1
3	65.00	0	146	0	20	0	162000	1.30	129	Male	1	7	1
4	50.00	1	111	0	20	0	210000	1.90	137	Male	0	7	1
5	65.00	1	160	1	20	0	327000	2.70	116	Female	0	8	1
6	90.00	1	47	0	40	1	204000	2.10	132	Male	1	8	1
7	75.00	1	246	NA	15	0	127000	1.20	137	Male	0	10	1
8	60.00	NA	315	1	60	0	454000	1.10	131	Male	1	10	1
9	65.00	0	157	0	65	0	263358	1.50	138	Female	0	10	1
10	NA	1	123	0	35	1	388000	9.40	133	Maleee	1	10	1
11	75.00	1	81	0	38	1	368000	4.00	131	Male	1	10	1
12	62.00	0	231	0	25	1	NA	0.90	140	Male	NA	10	1
13	45.00	1	981	0	30	0	136000	1.10	137	Male	0	11	1
14	50.00	1	168	0	38	1	276000	1.10	137	Male	0	11	1
15	49.00	1	80	0	30	1	427000	1.00	138	Female	0	12	0
16	82.00	1	379	0	50	0	47000	1.30	136	Male	0	13	1
17	87.00	1	NA	0	38	0	262000	0.90	140	Male	0	14	1
18	45.00	0	582	0	14	0	166000	0.80	127	Male	NA	14	1
19	70.00	1	125	0	25	1	237000	1.00	140	Female	0	15	1
20	48.00	1	582	1	55	0	87000	1.90	121	Female	0	15	1
21	65.00	1	52	0	25	1	276000	1.30	137	Female	0	16	0
22	65.00	NA	128	1	30	1	297000	1.60	136	Female	0	20	1
23	68.00	1	220	0	35	1	289000	0.90	140	Male	1	20	1
24	53.00	0	63	1	60	0	368000	0.80	135	Male	0	22	0
25	75.00	0	582	1	30	1	263358	1.83	134	Female	0	23	1
26	80.00	0	148	1	38	0	149000	1.90	144	Male	1	23	1

Showing 1 to 27 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
27	95.00	1	112	0	40	1	NA	1.00	138	Female	0	24	1
28	70.00	0	122	1	45	1	284000	1.30	136	Male	1	26	1
29	NA	1	60	0	38	0	153000	5.80	134	Male	0	26	1
30	82.00	0	70	1	30	0	200000	1.20	132	Male	1	26	1
31	94.00	0	582	1	38	1	263358	1.83	134	Male	0	27	1
32	85.00	0	23	0	45	0	360000	3.00	132	Male	0	28	1
33	50.00	1	249	1	35	1	319000	1.00	128	Femmale	0	28	1
34	50.00	1	159	1	30	0	302000	1.20	138	Female	0	29	0
35	65.00	0	94	1	50	1	188000	1.00	140	Male	0	29	1
36	NA	NA	582	1	35	0	228000	3.50	134	Male	0	30	1
37	90.00	1	60	NA	50	0	226000	1.00	134	Male	0	30	1
38	82.00	1	855	1	50	1	321000	1.00	145	Female	0	30	1
39	60.00	0	2656	1	30	0	305000	2.30	137	Male	0	30	0
40	60.00	NA	235	1	38	0	329000	3.00	142	Female	0	30	1
41	70.00	0	582	0	20	1	263358	1.83	134	Male	1	31	1
42	50.00	0	124	1	30	1	153000	1.20	136	Female	1	32	1
43	70.00	0	571	1	45	1	185000	1.20	139	Male	1	33	1
44	72.00	0	127	1	50	1	218000	1.00	134	Male	NA	33	0
45	60.00	1	588	1	60	0	NA	1.10	142	Female	0	33	1
46	50.00	0	582	1	38	0	310000	1.90	135	Male	1	35	1
47	51.00	0	1380	0	25	1	271000	0.90	130	Male	0	38	1
48	60.00	NA	582	1	38	1	451000	0.60	138	Male	1	40	1
49	80.00	1	553	0	20	1	140000	4.40	133	Male	0	41	1
50	NA	1	129	0	30	0	395000	1.00	140	Female	0	42	1
51	68.00	1	NA	0	25	1	166000	1.00	138	Male	0	43	1
52	53.00	1	91	0	20	1	418000	1.40	139	Female	0	43	1

Showing 27 to 53 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
53	60.00	0	3964	1	62	0	263358	6.80	146	Female	0	43	1
54	70.00	1	69	1	50	1	351000	1.00	134	Female	0	44	1
55	60.00	1	260	1	38	0	255000	2.20	132	Female	1	45	1
56	95.00	1	371	0	30	0	461000	2.00	132	Male	0	50	1
57	70.00	1	75	0	35	0	223000	2.70	138	Male	1	54	0
58	60.00	1	607	0	40	0	216000	0.60	138	Male	1	54	0
59	49.00	0	789	0	20	1	319000	1.10	136	Male	1	55	1
60	72.00	0	364	1	20	1	254000	1.30	136	Male	1	59	1
61	45.00	0	7702	1	25	1	390000	1.00	139	Male	0	60	1
62	50.00	0	318	0	40	1	216000	2.30	131	Female	0	60	1
63	55.00	0	109	0	35	0	254000	1.10	139	Male	1	60	0
64	45.00	0	*NA*	0	35	0	385000	1.00	145	Male	0	61	1
65	45.00	0	582	0	80	0	263358	1.18	137	Female	0	63	0
66	60.00	0	68	0	20	0	119000	2.90	127	Male	1	64	1
67	*NA*	1	250	1	15	0	213000	1.30	136	Female	0	65	1
68	72.00	1	110	0	25	0	274000	1.00	140	Male	1	65	1
69	70.00	0	161	0	25	0	244000	1.20	142	Female	0	66	1
70	65.00	0	113	1	25	0	497000	1.83	135	Male	0	67	1
71	41.00	0	148	0	40	0	374000	0.80	140	Male	1	68	0
72	58.00	0	582	1	35	0	122000	0.90	139	Male	1	71	0
73	85.00	0	5882	0	35	0	243000	1.00	132	Male	1	72	1
74	65.00	0	224	1	50	0	149000	1.30	137	Male	1	72	0
75	69.00	0	582	0	20	0	*NA*	1.20	134	Male	1	73	1
76	60.00	1	47	0	20	0	204000	0.70	139	Male	1	73	1
77	70.00	0	92	0	60	1	317000	0.80	140	Female	1	74	0
78	42.00	0	102	1	40	0	237000	1.20	140	Male	0	74	0

Showing 53 to 79 of 199 entries. 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
79	75.00	1	203	1	38	1	283000	0.60	131	Male	1	74	0
80	55.00	0	336	0	45	1	324000	0.90	140	Female	0	74	0
81	70.00	0	69	0	40	0	293000	1.70	136	Female	0	75	0
82	67.00	0	582	0	50	0	263358	1.18	137	Male	1	76	0
83	60.00	1	76	1	25	0	196000	2.50	132	Female	0	77	1
84	79.00	1	55X	0	50	1	172000	1.80	133	Male	0	78	0
85	59.00	1	280	1	25	1	302000	1.00	141	Female	0	78	1
86	51.00	0	78	0	50	0	406000	0.70	140	Male	0	79	0
87	55.00	0	*NA*	0	35	1	173000	1.10	137	Male	0	79	0
88	65.00	1	68	1	60	1	304000	0.80	140	Male	0	79	0
89	44.00	0	84	1	40	1	235000	0.70	139	Male	0	79	0
90	57.00	1	115	0	25	1	181000	1.10	144	Male	0	79	0
91	70.00	0	66	1	45	0	249000	0.80	136	Male	1	80	0
92	60.00	0	897	1	45	0	297000	1.00	133	Male	0	80	0
93	42.00	0	582	0	60	0	263358	1.18	137	Female	0	82	0
94	60.00	1	154	0	25	0	210000	1.70	135	Male	0	82	1
95	58.00	0	144	1	38	1	327000	0.70	142	Female	0	83	0
96	58.00	1	133	0	60	1	219000	1.00	141	Male	0	83	0
97	63.00	1	514	1	25	1	254000	1.30	134	Male	0	83	0
98	70.00	1	59	0	60	0	255000	1.10	136	Female	0	85	0
99	60.00	1	156	1	25	1	318000	1.20	137	Female	0	85	0
100	63.00	1	61	1	40	0	*NA*	1.10	140	Female	0	86	0
101	65.00	1	305	0	25	0	298000	1.10	141	Male	0	87	0
102	75.00	0	582	0	45	1	263358	1.18	137	Male	0	87	0
103	80.00	0	898	0	25	0	149000	1.10	144	Male	1	87	0
104	42.00	0	5209	0	30	0	226000	1.00	140	Male	1	87	0

Showing 79 to 104 of 199 entries. 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
105	60.00	0	53	0	50	1	286000	2.30	143	Female	0	87	0
106	72.00	1	328	0	30	1	621000	1.70	138	Female	1	88	1
107	55.00	0	748	0	45	0	263000	1.30	137	Male	0	88	0
108	45.00	1	1876	1	35	0	226000	0.90	138	Male	0	88	0
109	63.00	0	936	0	38	0	304000	1.10	133	Male	1	88	0
110	45.00	0	292	1	35	0	850000	1.30	142	Male	1	88	0
111	85.00	0	129	0	60	0	306000	1.20	132	Male	1	90	1
112	55.00	0	60	0	35	0	228000	1.20	135	Male	1	90	0
113	50.00	0	369	1	25	0	252000	1.60	136	Male	0	90	0
114	70.00	1	143	0	60	0	351000	1.30	137	Female	0	90	1
115	60.00	1	754	1	40	1	328000	1.20	126	Male	0	91	0
116	58.00	1	400	0	40	0	164000	1.00	139	Female	0	91	0
117	60.00	1	96	1	60	1	271000	0.70	136	Female	0	94	0
118	85.00	1	102	0	60	0	507000	3.20	138	Female	0	94	0
119	65.00	1	113	1	60	1	203000	0.90	140	Female	0	94	0
120	86.00	0	582	0	38	0	263358	1.83	134	Female	0	95	1
121	60.00	1	737	0	60	1	210000	1.50	135	Male	1	95	0
122	66.00	1	68	1	38	1	162000	1.00	136	Female	0	95	0
123	60.00	0	96	1	38	0	*NA*	0.75	140	Female	0	95	0
124	60.00	1	582	0	30	1	127000	0.90	145	Female	0	95	0
125	60.00	0	582	0	40	0	217000	3.70	134	Male	0	96	1
126	43.00	1	358	0	50	0	237000	1.30	135	Female	0	97	0
127	46.00	0	168	1	17	1	271000	2.10	124	Female	0	100	1
128	58.00	1	200	*NA*	60	0	300000	0.80	137	Female	0	104	0
129	61.00	0	248	0	30	1	267000	0.70	136	Male	1	104	0
130	53.00	1	270	1	35	0	227000	3.40	145	Male	0	105	0

Showing 105 to 131 of 199 entries. 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
131	53.00	1	1808	0	60	1	249000	0.70	138	Male	1	106	0
132	60.00	1	1082	1	45	0	250000	6.10	131	Male	0	107	0
133	46.00	0	719	0	40	1	263358	1.18	137	Female	0	107	0
134	63.00	0	193	0	60	1	295000	1.30	145	Male	1	107	0
135	81.00	0	4540	0	35	0	231000	1.18	137	Male	1	107	0
136	75.00	0	582	0	40	0	263358	1.18	137	Male	0	107	0
137	65.00	1	59	1	60	0	172000	0.90	137	Female	0	107	0
138	68.00	1	646	0	25	0	305000	2.10	130	Male	0	108	0
139	62.00	0	281	1	35	0	221000	1.00	136	Female	0	108	0
140	50.00	0	1548	0	30	1	211000	0.80	138	Male	0	108	0
141	80.00	0	805	0	38	0	263358	1.10	134	Male	0	109	1
142	46.00	1	291	0	35	0	348000	0.90	140	Female	0	109	0
143	50.00	0	482	1	30	0	329000	0.90	132	Female	0	109	0
144	60.00	1	84	0	40	1	229000	0.90	141	Female	0	110	0
145	72.00	1	943	0	25	1	338000	1.70	139	Male	1	111	1
146	50.00	0	185	0	30	0	266000	0.70	141	Male	1	112	0
147	52.00	0	132	0	30	0	218000	0.70	136	Male	1	112	0
148	64.00	0	1610	0	60	0	242000	1.00	137	Male	0	113	0
149	75.00	1	582	0	30	0	225000	1.83	134	Male	0	113	1
150	60.00	0	2261	0	35	1	228000	0.90	136	Male	0	115	0
151	72.00	0	233	0	45	1	235000	2.50	135	Female	0	115	1
152	62.00	0	30	1	60	1	244000	0.90	139	Male	0	117	0
153	50.00	0	115	0	45	1	184000	0.90	134	Male	1	118	0
154	50.00	0	1846	1	35	0	263358	1.18	137	Male	1	119	0
155	65.00	1	335	0	35	1	235000	0.80	136	Female	0	120	0
156	60.00	1	231	1	25	0	194000	1.70	140	Male	0	120	0

Showing 131 to 157 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
157	52.00	1	58	0	35	0	277000	1.40	136	Female	0	120	0
158	50.00	0	250	0	25	0	262000	1.00	136	Male	1	120	0
159	85.00	1	910	0	50	0	235000	1.30	134	Male	0	121	0
160	59.00	1	129	0	45	1	362000	1.10	139	Male	1	121	0
161	66.00	1	72	0	40	1	242000	1.20	134	Male	0	121	0
162	45.00	1	130	0	35	0	174000	0.80	139	Male	1	121	0
163	63.00	1	582	0	40	0	448000	0.90	137	Male	1	123	0
164	50.00	1	2334	NA	35	0	75000	0.90	142	Female	0	126	1
165	45.00	0	2442	1	30	0	334000	1.10	139	Male	0	129	1
166	80.00	0	776	1	38	1	192000	1.30	135	Female	0	130	1
167	53.00	0	196	0	60	0	220000	0.70	133	Male	1	134	0
168	59.00	0	66	1	20	0	70000	2.40	134	Male	0	135	1
169	65.00	0	582	1	40	0	270000	1.00	138	Female	0	140	0
170	70.00	0	835	0	35	1	305000	0.80	133	Female	0	145	0
171	51.00	1	582	1	35	0	263358	1.50	136	Male	1	145	0
172	52.00	0	3966	0	40	0	325000	0.90	140	Male	1	146	0
173	70.00	1	171	0	60	1	176000	1.10	145	Male	1	146	0
174	50.00	1	115	0	20	0	189000	0.80	139	Male	0	146	0
175	65.00	0	198	1	35	1	281000	0.90	137	Male	1	146	0
176	60.00	1	95	0	60	0	337000	1.00	138	Male	1	146	0
177	69.00	0	1419	0	40	0	105000	1.00	135	Male	1	147	0
178	49.00	1	69	0	50	0	132000	1.00	140	Female	0	147	0
179	63.00	1	122	1	60	0	267000	1.20	145	Male	0	147	0
180	55.00	0	835	0	40	0	279000	0.70	140	Male	1	147	0
181	40.00	0	478	1	30	0	303000	0.90	136	Male	0	148	0
182	59.00	1	176	1	25	0	221000	1.00	136	Male	1	150	1

Showing 157 to 183 of 199 entries, 13 total columns

183	65.00	0	395	1	25	0	265000	1.20	136	Male	1	154	1
184	75.00	0	99	0	38	1	224000	2.50	134	Male	0	162	1
185	58.00	1	145	0	25	0	219000	1.20	137	Male	1	170	1
186	60.00	1	104	1	30	0	389000	1.50	136	Male	0	171	1
187	50.00	0	582	0	50	0	153000	0.60	134	Female	0	172	1
188	60.00	0	1896	1	25	0	365000	2.10	144	Female	0	172	1
189	60.66	1	151	1	40	1	201000	1.00	136	Female	0	172	0
190	40.00	0	244	0	45	1	275000	0.90	140	Female	0	174	0
191	80.00	0	582	1	35	0	350000	2.10	134	Male	0	174	0
192	64.00	1	62	0	60	0	309000	1.50	135	Female	0	174	0
193	50.00	1	121	1	40	0	260000	0.70	130	Male	0	175	0
194	73.00	1	231	1	30	0	160000	1.18	142	Male	1	180	0
195	45.00	0	582	0	20	1	126000	1.60	135	Male	0	180	1
196	77.00	1	418	0	45	0	223000	1.80	145	Male	0	180	1
197	45.00	0	582	1	38	1	263358	1.18	137	Female	0	185	0
198	65.00	0	167	0	30	0	259000	0.80	138	Female	0	186	0
199	50.00	1	582	1	20	1	279000	1.00	134	Female	0	186	0

Showing 174 to 199 of 199 entries, 13 total columns

First, age attribute was boxplotted from the dataset to detect the outliers and saved its instance as ageBoxplot. Then, the outliers were extracted using ageBoxplot$out. After that, cat method was used to print the outliers. The median value of the age attribute was stored in a variable ageMedian using the median method. na.rm = TRUE parameter was passed as an argument so that the NA values are ignored while calculating median. After that, the positions of the outliers were stored in outlierPositions using the match method which returns the outlier positions in the age attribute. Then the median value 60 was converted to integer and replaced all the outliers in the age attribute. Finally, the dataset was printed.

❖ **Replacing age Outliers using Mode Value**

Code:

```
ageBoxplot <- boxplot(dataset$age,main=" Age Distribution ", ylab="age",col="orange")
outliers <- ageBoxplot$out
cat("Outliers are", outliers)
ageMode <- strtoi(names(sort(table(dataset$age[!is.na(dataset$age)]), decreasing = TRUE)[1]))
outlierPositions <- match(outliers, dataset$age)
dataset$age[outlierPositions] <- as.integer (ageMode)
```

Output:

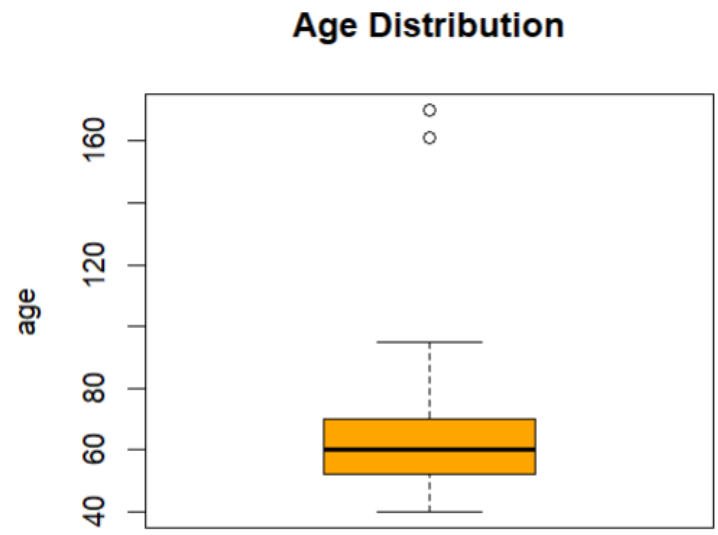


Fig: Age Distribution

Here the values belong to outside of the box are Outliers.

```
> ageBoxplot <- boxplot(dataset$age,main=" Age Distribution ", ylab="age",col="orange")
> outliers <- ageBoxplot$out
> cat("Outliers are", outliers)
Outliers are 161 170> ageMode <- strtoi(names(sort(table(dataset$age[!is.na(dataset$age)]), decreasing = TRUE)[1]))
> outlierPositions <- match(outliers, dataset$age)
> dataset$age[outlierPositions] <- as.integer (ageMode)
> View(dataset)
>
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
1	75.00	0	582	0	20	1	265000	1.90	130	Male	0	4	1
2	55.00	0	7861	0	38	0	263358	1.10	136	Male	0	6	1
3	65.00	0	146	0	20	0	162000	1.30	129	Male	1	7	1
4	50.00	1	111	0	20	0	210000	1.90	137	Male	0	7	1
5	65.00	1	160	1	20	0	327000	2.70	116	Female	0	8	1
6	90.00	1	47	0	40	1	204000	2.10	132	Male	1	8	1
7	75.00	1	246	NA	15	0	127000	1.20	137	Male	0	10	1
8	60.00	NA	315	1	60	0	454000	1.10	131	Male	1	10	1
9	65.00	0	157	0	65	0	263358	1.50	138	Female	0	10	1
10	NA	1	123	0	35	1	388000	9.40	133	Maleee	1	10	1
11	75.00	1	81	0	38	1	368000	4.00	131	Male	1	10	1
12	62.00	0	231	0	25	1	NA	0.90	140	Male	NA	10	1
13	45.00	1	981	0	30	0	136000	1.10	137	Male	0	11	1
14	50.00	1	168	0	38	1	276000	1.10	137	Male	0	11	1
15	49.00	1	80	0	30	1	427000	1.00	138	Female	0	12	0
16	82.00	1	379	0	50	0	47000	1.30	136	Male	0	13	1
17	87.00	1	NA	0	38	0	262000	0.90	140	Male	0	14	1
18	45.00	0	582	0	14	0	166000	0.80	127	Male	NA	14	1
19	70.00	1	125	0	25	1	237000	1.00	140	Female	0	15	1
20	48.00	1	582	1	55	0	87000	1.90	121	Female	0	15	1
21	65.00	1	52	0	25	1	276000	1.30	137	Female	0	16	0
22	65.00	NA	128	1	30	1	297000	1.60	136	Female	0	20	1
23	68.00	1	220	0	35	1	289000	0.90	140	Male	1	20	1
24	53.00	0	63	1	60	0	368000	0.80	135	Male	0	22	0
25	75.00	0	582	1	30	1	263358	1.83	134	Female	0	23	1
26	80.00	0	148	1	38	0	149000	1.90	144	Male	1	23	1

Showing 1 to 27 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
27	95.00	1	112	0	40	1	NA	1.00	138	Female	0	24	1
28	70.00	0	122	1	45	1	284000	1.30	136	Male	1	26	1
29	NA	1	60	0	38	0	153000	5.80	134	Male	0	26	1
30	82.00	0	70	1	30	0	200000	1.20	132	Male	1	26	1
31	94.00	0	582	1	38	1	263358	1.83	134	Male	0	27	1
32	85.00	0	23	0	45	0	360000	3.00	132	Male	0	28	1
33	50.00	1	249	1	35	1	319000	1.00	128	Femmale	0	28	1
34	50.00	1	159	1	30	0	302000	1.20	138	Female	0	29	0
35	65.00	0	94	1	50	1	188000	1.00	140	Male	0	29	1
36	NA	NA	582	1	35	0	228000	3.50	134	Male	0	30	1
37	90.00	1	60	NA	50	0	226000	1.00	134	Male	0	30	1
38	82.00	1	855	1	50	1	321000	1.00	145	Female	0	30	1
39	60.00	0	2656	1	30	0	305000	2.30	137	Male	0	30	0
40	60.00	NA	235	1	38	0	329000	3.00	142	Female	0	30	1
41	70.00	0	582	0	20	1	263358	1.83	134	Male	1	31	1
42	50.00	0	124	1	30	1	153000	1.20	136	Female	1	32	1
43	70.00	0	571	1	45	1	185000	1.20	139	Male	1	33	1
44	72.00	0	127	1	50	1	218000	1.00	134	Male	NA	33	0
45	60.00	1	588	1	60	0	NA	1.10	142	Female	0	33	1
46	50.00	0	582	1	38	0	310000	1.90	135	Male	1	35	1
47	51.00	0	1380	0	25	1	271000	0.90	130	Male	0	38	1
48	60.00	NA	582	1	38	1	451000	0.60	138	Male	1	40	1
49	80.00	1	553	0	20	1	140000	4.40	133	Male	0	41	1
50	NA	1	129	0	30	0	395000	1.00	140	Female	0	42	1
51	68.00	1	NA	0	25	1	166000	1.00	138	Male	0	43	1
52	53.00	1	91	0	20	1	418000	1.40	139	Female	0	43	1

Showing 27 to 53 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
53	60.00	0	3964	1	62	0	263358	6.80	146	Female	0	43	1
54	70.00	1	69	1	50	1	351000	1.00	134	Female	0	44	1
55	60.00	1	260	1	38	0	255000	2.20	132	Female	1	45	1
56	95.00	1	371	0	30	0	461000	2.00	132	Male	0	50	1
57	70.00	1	75	0	35	0	223000	2.70	138	Male	1	54	0
58	60.00	1	607	0	40	0	216000	0.60	138	Male	1	54	0
59	49.00	0	789	0	20	1	319000	1.10	136	Male	1	55	1
60	72.00	0	364	1	20	1	254000	1.30	136	Male	1	59	1
61	45.00	0	7702	1	25	1	390000	1.00	139	Male	0	60	1
62	50.00	0	318	0	40	1	216000	2.30	131	Female	0	60	1
63	55.00	0	109	0	35	0	254000	1.10	139	Male	1	60	0
64	45.00	0	*NA*	0	35	0	385000	1.00	145	Male	0	61	1
65	45.00	0	582	0	80	0	263358	1.18	137	Female	0	63	0
66	60.00	0	68	0	20	0	119000	2.90	127	Male	1	64	1
67	*NA*	1	250	1	15	0	213000	1.30	136	Female	0	65	1
68	72.00	1	110	0	25	0	274000	1.00	140	Male	1	65	1
69	70.00	0	161	0	25	0	244000	1.20	142	Female	0	66	1
70	65.00	0	113	1	25	0	497000	1.83	135	Male	0	67	1
71	41.00	0	148	0	40	0	374000	0.80	140	Male	1	68	0
72	58.00	0	582	1	35	0	122000	0.90	139	Male	1	71	0
73	85.00	0	5882	0	35	0	243000	1.00	132	Male	1	72	1
74	65.00	0	224	1	50	0	149000	1.30	137	Male	1	72	0
75	69.00	0	582	0	20	0	*NA*	1.20	134	Male	1	73	1
76	60.00	1	47	0	20	0	204000	0.70	139	Male	1	73	1
77	70.00	0	92	0	60	1	317000	0.80	140	Female	1	74	0
78	42.00	0	102	1	40	0	237000	1.20	140	Male	0	74	0

Showing 53 to 79 of 199 entries. 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
79	75.00	1	203	1	38	1	283000	0.60	131	Male	1	74	0
80	55.00	0	336	0	45	1	324000	0.90	140	Female	0	74	0
81	70.00	0	69	0	40	0	293000	1.70	136	Female	0	75	0
82	67.00	0	582	0	50	0	263358	1.18	137	Male	1	76	0
83	60.00	1	76	1	25	0	196000	2.50	132	Female	0	77	1
84	79.00	1	55X	0	50	1	172000	1.80	133	Male	0	78	0
85	59.00	1	280	1	25	1	302000	1.00	141	Female	0	78	1
86	51.00	0	78	0	50	0	406000	0.70	140	Male	0	79	0
87	55.00	0	*NA*	0	35	1	173000	1.10	137	Male	0	79	0
88	65.00	1	68	1	60	1	304000	0.80	140	Male	0	79	0
89	44.00	0	84	1	40	1	235000	0.70	139	Male	0	79	0
90	57.00	1	115	0	25	1	181000	1.10	144	Male	0	79	0
91	70.00	0	66	1	45	0	249000	0.80	136	Male	1	80	0
92	60.00	0	897	1	45	0	297000	1.00	133	Male	0	80	0
93	42.00	0	582	0	60	0	263358	1.18	137	Female	0	82	0
94	60.00	1	154	0	25	0	210000	1.70	135	Male	0	82	1
95	58.00	0	144	1	38	1	327000	0.70	142	Female	0	83	0
96	58.00	1	133	0	60	1	219000	1.00	141	Male	0	83	0
97	63.00	1	514	1	25	1	254000	1.30	134	Male	0	83	0
98	70.00	1	59	0	60	0	255000	1.10	136	Female	0	85	0
99	60.00	1	156	1	25	1	318000	1.20	137	Female	0	85	0
100	63.00	1	61	1	40	0	*NA*	1.10	140	Female	0	86	0
101	65.00	1	305	0	25	0	298000	1.10	141	Male	0	87	0
102	75.00	0	582	0	45	1	263358	1.18	137	Male	0	87	0
103	80.00	0	898	0	25	0	149000	1.10	144	Male	1	87	0
104	42.00	0	5209	0	30	0	226000	1.00	140	Male	1	87	0

Showing 79 to 104 of 199 entries. 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
105	60.00	0	53	0	50	1	286000	2.30	143	Female	0	87	0
106	72.00	1	328	0	30	1	621000	1.70	138	Female	1	88	1
107	55.00	0	748	0	45	0	263000	1.30	137	Male	0	88	0
108	45.00	1	1876	1	35	0	226000	0.90	138	Male	0	88	0
109	63.00	0	936	0	38	0	304000	1.10	133	Male	1	88	0
110	45.00	0	292	1	35	0	850000	1.30	142	Male	1	88	0
111	85.00	0	129	0	60	0	306000	1.20	132	Male	1	90	1
112	55.00	0	60	0	35	0	228000	1.20	135	Male	1	90	0
113	50.00	0	369	1	25	0	252000	1.60	136	Male	0	90	0
114	70.00	1	143	0	60	0	351000	1.30	137	Female	0	90	1
115	60.00	1	754	1	40	1	328000	1.20	126	Male	0	91	0
116	58.00	1	400	0	40	0	164000	1.00	139	Female	0	91	0
117	60.00	1	96	1	60	1	271000	0.70	136	Female	0	94	0
118	85.00	1	102	0	60	0	507000	3.20	138	Female	0	94	0
119	65.00	1	113	1	60	1	203000	0.90	140	Female	0	94	0
120	86.00	0	582	0	38	0	263358	1.83	134	Female	0	95	1
121	60.00	1	737	0	60	1	210000	1.50	135	Male	1	95	0
122	66.00	1	68	1	38	1	162000	1.00	136	Female	0	95	0
123	60.00	0	96	1	38	0	*NA*	0.75	140	Female	0	95	0
124	60.00	1	582	0	30	1	127000	0.90	145	Female	0	95	0
125	60.00	0	582	0	40	0	217000	3.70	134	Male	0	96	1
126	43.00	1	358	0	50	0	237000	1.30	135	Female	0	97	0
127	46.00	0	168	1	17	1	271000	2.10	124	Female	0	100	1
128	58.00	1	200	*NA*	60	0	300000	0.80	137	Female	0	104	0
129	61.00	0	248	0	30	1	267000	0.70	136	Male	1	104	0
130	53.00	1	270	1	35	0	227000	3.40	145	Male	0	105	0

Showing 105 to 131 of 199 entries. 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
131	53.00	1	1808	0	60	1	249000	0.70	138	Male	1	106	0
132	60.00	1	1082	1	45	0	250000	6.10	131	Male	0	107	0
133	46.00	0	719	0	40	1	263358	1.18	137	Female	0	107	0
134	63.00	0	193	0	60	1	295000	1.30	145	Male	1	107	0
135	81.00	0	4540	0	35	0	231000	1.18	137	Male	1	107	0
136	75.00	0	582	0	40	0	263358	1.18	137	Male	0	107	0
137	65.00	1	59	1	60	0	172000	0.90	137	Female	0	107	0
138	68.00	1	646	0	25	0	305000	2.10	130	Male	0	108	0
139	62.00	0	281	1	35	0	221000	1.00	136	Female	0	108	0
140	50.00	0	1548	0	30	1	211000	0.80	138	Male	0	108	0
141	80.00	0	805	0	38	0	263358	1.10	134	Male	0	109	1
142	46.00	1	291	0	35	0	348000	0.90	140	Female	0	109	0
143	50.00	0	482	1	30	0	329000	0.90	132	Female	0	109	0
144	60.00	1	84	0	40	1	229000	0.90	141	Female	0	110	0
145	72.00	1	943	0	25	1	338000	1.70	139	Male	1	111	1
146	50.00	0	185	0	30	0	266000	0.70	141	Male	1	112	0
147	52.00	0	132	0	30	0	218000	0.70	136	Male	1	112	0
148	64.00	0	1610	0	60	0	242000	1.00	137	Male	0	113	0
149	75.00	1	582	0	30	0	225000	1.83	134	Male	0	113	1
150	60.00	0	2261	0	35	1	228000	0.90	136	Male	0	115	0
151	72.00	0	233	0	45	1	235000	2.50	135	Female	0	115	1
152	62.00	0	30	1	60	1	244000	0.90	139	Male	0	117	0
153	50.00	0	115	0	45	1	184000	0.90	134	Male	1	118	0
154	50.00	0	1846	1	35	0	263358	1.18	137	Male	1	119	0
155	65.00	1	335	0	35	1	235000	0.80	136	Female	0	120	0
156	60.00	1	231	1	25	0	194000	1.70	140	Male	0	120	0

Showing 131 to 157 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
157	52.00	1	58	0	35	0	277000	1.40	136	Female	0	120	0
158	50.00	0	250	0	25	0	262000	1.00	136	Male	1	120	0
159	85.00	1	910	0	50	0	235000	1.30	134	Male	0	121	0
160	59.00	1	129	0	45	1	362000	1.10	139	Male	1	121	0
161	66.00	1	72	0	40	1	242000	1.20	134	Male	0	121	0
162	45.00	1	130	0	35	0	174000	0.80	139	Male	1	121	0
163	63.00	1	582	0	40	0	448000	0.90	137	Male	1	123	0
164	50.00	1	2334	NA	35	0	75000	0.90	142	Female	0	126	1
165	45.00	0	2442	1	30	0	334000	1.10	139	Male	0	129	1
166	80.00	0	776	1	38	1	192000	1.30	135	Female	0	130	1
167	53.00	0	196	0	60	0	220000	0.70	133	Male	1	134	0
168	59.00	0	66	1	20	0	70000	2.40	134	Male	0	135	1
169	65.00	0	582	1	40	0	270000	1.00	138	Female	0	140	0
170	70.00	0	835	0	35	1	305000	0.80	133	Female	0	145	0
171	51.00	1	582	1	35	0	263358	1.50	136	Male	1	145	0
172	52.00	0	3966	0	40	0	325000	0.90	140	Male	1	146	0
173	70.00	1	171	0	60	1	176000	1.10	145	Male	1	146	0
174	50.00	1	115	0	20	0	189000	0.80	139	Male	0	146	0
175	65.00	0	198	1	35	1	281000	0.90	137	Male	1	146	0
176	60.00	1	95	0	60	0	337000	1.00	138	Male	1	146	0
177	69.00	0	1419	0	40	0	105000	1.00	135	Male	1	147	0
178	49.00	1	69	0	50	0	132000	1.00	140	Female	0	147	0
179	63.00	1	122	1	60	0	267000	1.20	145	Male	0	147	0
180	55.00	0	835	0	40	0	279000	0.70	140	Male	1	147	0
181	40.00	0	478	1	30	0	303000	0.90	136	Male	0	148	0
182	59.00	1	176	1	25	0	221000	1.00	136	Male	1	150	1

Showing 157 to 183 of 199 entries, 13 total columns

183	65.00	0	395	1	25	0	265000	1.20	136	Male	1	154	1
184	75.00	0	99	0	38	1	224000	2.50	134	Male	0	162	1
185	58.00	1	145	0	25	0	219000	1.20	137	Male	1	170	1
186	60.00	1	104	1	30	0	389000	1.50	136	Male	0	171	1
187	50.00	0	582	0	50	0	153000	0.60	134	Female	0	172	1
188	60.00	0	1896	1	25	0	365000	2.10	144	Female	0	172	1
189	60.66	1	151	1	40	1	201000	1.00	136	Female	0	172	0
190	40.00	0	244	0	45	1	275000	0.90	140	Female	0	174	0
191	80.00	0	582	1	35	0	350000	2.10	134	Male	0	174	0
192	64.00	1	62	0	60	0	309000	1.50	135	Female	0	174	0
193	50.00	1	121	1	40	0	260000	0.70	130	Male	0	175	0
194	73.00	1	231	1	30	0	160000	1.18	142	Male	1	180	0
195	45.00	0	582	0	20	1	126000	1.60	135	Male	0	180	1
196	77.00	1	418	0	45	0	223000	1.80	145	Male	0	180	1
197	45.00	0	582	1	38	1	263358	1.18	137	Female	0	185	0
198	65.00	0	167	0	30	0	259000	0.80	138	Female	0	186	0
199	50.00	1	582	1	20	1	279000	1.00	134	Female	0	186	0

Showing 174 to 199 of 199 entries, 13 total columns

First, age attribute was boxplotted from the dataset to find the outliers and saved its instance as ageBoxplot. Then, the outliers were extracted using ageBoxplot$out. After that, cat method was used to print the outliers. Then table method was used which returns a categorical representation of data with variable name. dataset$age[!is.na(dataset$age)] was passed as a parameter to tabulate only the age attribute where the values are not NA. After that the result was sorted using the sort method. decreasing = TRUE parameter was used to sort the result in a decreasing order. Then the names method returned the name and the result was stored in ageMode. After that, the positions of the outliers were stored in outlierPositions using the match method which returns the outlier positions in the age attribute. Then the mode value 60 was converted to integer and replaced all the outliers in the age attribute.

INVALID VALUES

❖ **Recover Invalid Values for creatinine_phosphokinase**

Code:

```
dataset$creatinine_phosphokinase <- as.numeric(dataset$creatinine_phosphokinase)
median_value <- median(dataset$creatinine_phosphokinase, na.rm = TRUE)
dataset$creatinine_phosphokinase[is.na(dataset$creatinine_phosphokinase)] <- median_value
print(dataset)
```

Output:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
1	75.00	0	582	0	20	1	265000	1.90	130	Male	0	4	1
2	55.00	0	7861	0	38	0	263358	1.10	136	Male	0	6	1
3	65.00	0	146	0	20	0	162000	1.30	129	Male	1	7	1
4	50.00	1	111	0	20	0	210000	1.90	137	Male	0	7	1
5	65.00	1	160	1	20	0	327000	2.70	116	Female	0	8	1
6	90.00	1	47	0	40	1	204000	2.10	132	Male	1	8	1
7	75.00	1	246	NA	15	0	127000	1.20	137	Male	0	10	1
8	60.00	NA	315	1	60	0	454000	1.10	131	Male	1	10	1
9	65.00	0	157	0	65	0	263358	1.50	138	Female	0	10	1
10	NA	1	123	0	35	1	388000	9.40	133	Maleee	1	10	1
11	75.00	1	81	0	38	1	368000	4.00	131	Male	1	10	1
12	62.00	0	231	0	25	1	NA	0.90	140	Male	NA	10	1
13	45.00	1	981	0	30	0	136000	1.10	137	Male	0	11	1
14	50.00	1	168	0	38	1	276000	1.10	137	Male	0	11	1
15	49.00	1	80	0	30	1	427000	1.00	138	Female	0	12	0
16	82.00	1	379	0	50	0	47000	1.30	136	Male	0	13	1
17	87.00	1	245	0	38	0	262000	0.90	140	Male	0	14	1
18	45.00	0	582	0	14	0	166000	0.80	127	Male	NA	14	1
19	70.00	1	125	0	25	1	237000	1.00	140	Female	0	15	1
20	48.00	1	582	1	55	0	87000	1.90	121	Female	0	15	1
21	65.00	1	52	0	25	1	276000	1.30	137	Female	0	16	0
22	65.00	NA	128	1	30	1	297000	1.60	136	Female	0	20	1
23	68.00	1	220	0	35	1	289000	0.90	140	Male	1	20	1
24	53.00	0	63	1	60	0	368000	0.80	135	Male	0	22	0
25	75.00	0	582	1	30	1	263358	1.83	134	Female	0	23	1
26	80.00	0	148	1	38	0	149000	1.90	144	Male	1	23	1

Showing 1 to 27 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
27	95.00	1	112	0	40	1	NA	1.00	138	Female	0	24	1
28	70.00	0	122	1	45	1	284000	1.30	136	Male	1	26	1
29	NA	1	60	0	38	0	153000	5.80	134	Male	0	26	1
30	82.00	0	70	1	30	0	200000	1.20	132	Male	1	26	1
31	94.00	0	582	1	38	1	263358	1.83	134	Male	0	27	1
32	85.00	0	23	0	45	0	360000	3.00	132	Male	0	28	1
33	50.00	1	249	1	35	1	319000	1.00	128	Femmale	0	28	1
34	50.00	1	159	1	30	0	302000	1.20	138	Female	0	29	0
35	65.00	0	94	1	50	1	188000	1.00	140	Male	0	29	1
36	NA	NA	582	1	35	0	228000	3.50	134	Male	0	30	1
37	90.00	1	60	NA	50	0	226000	1.00	134	Male	0	30	1
38	82.00	1	855	1	50	1	321000	1.00	145	Female	0	30	1
39	60.00	0	2656	1	30	0	305000	2.30	137	Male	0	30	0
40	60.00	NA	235	1	38	0	329000	3.00	142	Female	0	30	1
41	70.00	0	582	0	20	1	263358	1.83	134	Male	1	31	1
42	50.00	0	124	1	30	1	153000	1.20	136	Female	1	32	1
43	70.00	0	571	1	45	1	185000	1.20	139	Male	1	33	1
44	72.00	0	127	1	50	1	218000	1.00	134	Male	NA	33	0
45	60.00	1	588	1	60	0	NA	1.10	142	Female	0	33	1
46	50.00	0	582	1	38	0	310000	1.90	135	Male	1	35	1
47	51.00	0	1380	0	25	1	271000	0.90	130	Male	0	38	1
48	60.00	NA	582	1	38	1	451000	0.60	138	Male	1	40	1
49	80.00	1	553	0	20	1	140000	4.40	133	Male	0	41	1
50	NA	1	129	0	30	0	395000	1.00	140	Female	0	42	1
51	68.00	1	245	0	25	1	166000	1.00	138	Male	0	43	1
52	53.00	1	91	0	20	1	418000	1.40	139	Female	0	43	1

Showing 27 to 53 of 199 entries, 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
53	60.00	0	3964	1	62	0	263358	6.80	146	Female	0	43	1
54	70.00	1	69	1	50	1	351000	1.00	134	Female	0	44	1
55	60.00	1	260	1	38	0	255000	2.20	132	Female	1	45	1
56	95.00	1	371	0	30	0	461000	2.00	132	Male	0	50	1
57	70.00	1	75	0	35	0	223000	2.70	138	Male	1	54	0
58	60.00	1	607	0	40	0	216000	0.60	138	Male	1	54	0
59	49.00	0	789	0	20	1	319000	1.10	136	Male	1	55	1
60	72.00	0	364	1	20	1	254000	1.30	136	Male	1	59	1
61	45.00	0	7702	1	25	1	390000	1.00	139	Male	0	60	1
62	50.00	0	318	0	40	1	216000	2.30	131	Female	0	60	1
63	55.00	0	109	0	35	0	254000	1.10	139	Male	1	60	0
64	45.00	0	245	0	35	0	385000	1.00	145	Male	0	61	1
65	45.00	0	582	0	80	0	263358	1.18	137	Female	0	63	0
66	60.00	0	68	0	20	0	119000	2.90	127	Male	1	64	1
67	NA	1	250	1	15	0	213000	1.30	136	Female	0	65	1
68	72.00	1	110	0	25	0	274000	1.00	140	Male	1	65	1
69	70.00	0	161	0	25	0	244000	1.20	142	Female	0	66	1
70	65.00	0	113	1	25	0	497000	1.83	135	Male	0	67	1
71	41.00	0	148	0	40	0	374000	0.80	140	Male	1	68	0
72	58.00	0	582	1	35	0	122000	0.90	139	Male	1	71	0
73	85.00	0	5882	0	35	0	243000	1.00	132	Male	1	72	1
74	65.00	0	224	1	50	0	149000	1.30	137	Male	1	72	0
75	69.00	0	582	0	20	0	NA	1.20	134	Male	1	73	1
76	60.00	1	47	0	20	0	204000	0.70	139	Male	1	73	1
77	70.00	0	92	0	60	1	317000	0.80	140	Female	1	74	0
78	42.00	0	102	1	40	0	237000	1.20	140	Male	0	74	0

Showing 53 to 79 of 199 entries. 13 total columns

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
79	75.00	1	203	1	38	1	283000	0.60	131	Male	1	74	0
80	55.00	0	336	0	45	1	324000	0.90	140	Female	0	74	0
81	70.00	0	69	0	40	0	293000	1.70	136	Female	0	75	0
82	67.00	0	582	0	50	0	263358	1.18	137	Male	1	76	0
83	60.00	1	76	1	25	0	196000	2.50	132	Female	0	77	1
84	79.00	1	245	0	50	1	172000	1.80	133	Male	0	78	0
85	59.00	1	280	1	25	1	302000	1.00	141	Female	0	78	1
86	51.00	0	78	0	50	0	406000	0.70	140	Male	0	79	0
87	55.00	0	245	0	35	1	173000	1.10	137	Male	0	79	0
88	65.00	1	68	1	60	1	304000	0.80	140	Male	0	79	0
89	44.00	0	84	1	40	1	235000	0.70	139	Male	0	79	0
90	57.00	1	115	0	25	1	181000	1.10	144	Male	0	79	0
91	70.00	0	66	1	45	0	249000	0.80	136	Male	1	80	0
92	60.00	0	897	1	45	0	297000	1.00	133	Male	0	80	0
93	42.00	0	582	0	60	0	263358	1.18	137	Female	0	82	0
94	60.00	1	154	0	25	0	210000	1.70	135	Male	0	82	1
95	58.00	0	144	1	38	1	327000	0.70	142	Female	0	83	0
96	58.00	1	133	0	60	1	219000	1.00	141	Male	0	83	0
97	63.00	1	514	1	25	1	254000	1.30	134	Male	0	83	0
98	70.00	1	59	0	60	0	255000	1.10	136	Female	0	85	0
99	60.00	1	156	1	25	1	318000	1.20	137	Female	0	85	0
100	63.00	1	61	1	40	0	NA	1.10	140	Female	0	86	0
101	65.00	1	305	0	25	0	298000	1.10	141	Male	0	87	0
102	75.00	0	582	0	45	1	263358	1.18	137	Male	0	87	0
103	80.00	0	898	0	25	0	149000	1.10	144	Male	1	87	0
104	42.00	0	5209	0	30	0	226000	1.00	140	Male	1	87	0

Showing 79 to 104 of 199 entries. 13 total columns

At first, the creatinine_phosphokinase column is converted into numeric data type using as.numeric(dataset$creatinine_phosphokinase. Then invalid values and missing values were replaced with the correct values using median (dataset$creatinine_phosphokinase, na.rm = TRUE. Finally, printed the updated dataset with invalid and missing values replaced by the median. In the updated dataset, 55X (invalid value) was replaced by the median value of 245.

❖ **Recover Invalid Values for sex**

Code:

```
invalid_values <- !dataset$sex %in% c("Male", "Female")

dataset$sex[invalid_values] <- ifelse(dataset$sex[invalid_values] == "Maleee", "Male", "Female")

print(dataset)
```

Output:

```
> invalid_values <- !dataset$sex %in% c("Male", "Female")
> dataset$sex[invalid_values] <- ifelse(dataset$sex[invalid_values] == "Maleee", "Male", "Female")
> print(dataset)
   age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium    sex smoking time DEATH_EVENT
1   75       0                     582        0                20                   1    265000            1.90          130   Male       0    4           1
2   55       0                    7861        0                38                   0    263358            1.10          136   Male       0    6           1
3   65       0                     146        0                20                   0    162000            1.30          129   Male       1    7           1
4   50       1                     111        0                20                   0    210000            1.90          137   Male       0    7           1
5   65       1                     160        1                20                   0    327000            2.70          116 Female       0    8           1
6   90       1                      47        0                40                   1    204000            2.10          132   Male       1    8           1
7   75       1                     246       NA                15                   0    127000            1.20          137   Male       0   10           1
8   60      NA                     315        1                60                   0    454000            1.10          131   Male       1   10           1
9   65       0                     157        0                65                   0    263358            1.50          138 Female       0   10           1
10  NA       1                     123        0                35                   1    388000            9.40          133   Male       1   10           1
11  75       1                      81        0                38                   1    368000            4.00          131   Male       1   10           1
12  62       0                     231        0                25                   1        NA            0.90          140   Male      NA   10           1
13  45       1                     981        0                30                   0    136000            1.10          137   Male       0   11           1
14  50       1                     168        0                38                   1    276000            1.10          137   Male       0   11           1
15  49       1                      80        0                30                   1    427000            1.00          138 Female       0   12           0
16  82       1                     379        0                50                   0     47000            1.30          136   Male       0   13           1
17  87       1                    <NA>        0                38                   0    262000            0.90          140   Male       0   14           1
18  45       0                     582        0                14                   0    166000            0.80          127   Male      NA   14           1
19  70       1                     125        0                25                   1    237000            1.00          140 Female       0   15           1
20  48       1                     582        1                55                   0     87000            1.90          121 Female       0   15           1
21  65       1                      52        0                25                   1    276000            1.30          137 Female       0   16           0
22  65      NA                     128        1                30                   1    297000            1.60          136 Female       0   20           1
23  68       1                     220        0                35                   1    289000            0.90          140   Male       1   20           1
24  53       0                      63        1                60                   0    368000            0.80          135   Male       0   22           0
25  75       0                     582        1                30                   1    263358            1.83          134 Female       0   23           1
26  80       0                     148        1                38                   0    149000            1.90          144   Male       1   23           1
27  95       1                     112        0                40                   1        NA            1.00          138 Female       0   24           1
28  70       0                     122        1                45                   1    284000            1.30          136   Male       1   26           1
29  NA       1                      60        0                38                   0    153000            5.80          134   Male       0   26           1
30  82       0                      70        1                30                   0    200000            1.20          132   Male       1   26           1
31  94       0                     582        1                38                   1    263358            1.83          134   Male       0   27           1
32  85       0                      23        0                45                   0    360000            3.00          132   Male       0   28           1
33  50       1                     249        1                35                   1    319000            1.00          128 Female       0   28           1
34  50       1                     159        1                30                   0    302000            1.20          138 Female       0   29           0
35  65       0                      94        1                50                   1    188000            1.00          140   Male       0   29           1
36  NA      NA                     582        1                35                   0    228000            3.50          134   Male       0   30           1
```

First, invalid values were identified using !dataset$sex %in% c("Male", "Female"). Then invalid values were replaced with the correct values using ifelse(dataset$sex[invalid_values] == "Maleee", "Male", "Female"). Finally, printed the updated values.