AMERICAN INTERNATIONAL UNIVERSITY OF BANGLADESH

2025

Data Cleaning, Exploration, and Statistical Analysis of Health Dataset in R

This project involves a comprehensive analysis of a health dataset using R. It covers data import, missing value handling, outlier treatment, feature engineering, data normalization, duplication removal, invalid data correction, data balancing, and visualization. Statistical summaries including mean, median, mode, range, IQR, and variance are explored and visualized across gender and glucose levels to understand data distribution and trends

MD ADLUL ISLAM
adlulislam95@gmail.com

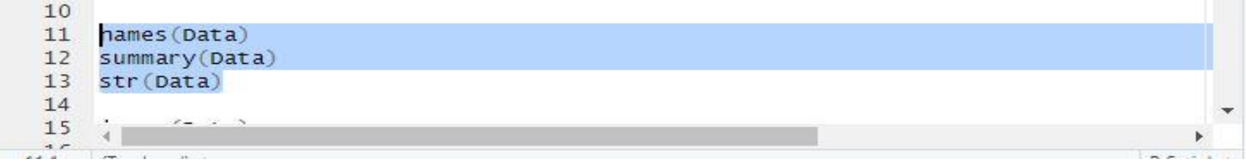
CALL ME
+880 1706 144 654

Heart Disease Classification Dataset:

The dataset examines the relationship between heart attack occurrences and various demographic and medical factors. Key attributes include age, gender, heart rate (impulse), systolic and diastolic blood pressure (high pressure and low pressure), and blood sugar level (glucose). The outcome variable indicates the presence or absence of a heart attack. This dataset facilitates analyzing how physiological and lifestyle-related factors influence cardiovascular health, identifies heart attack risk patterns, and informs preventive healthcare strategies.

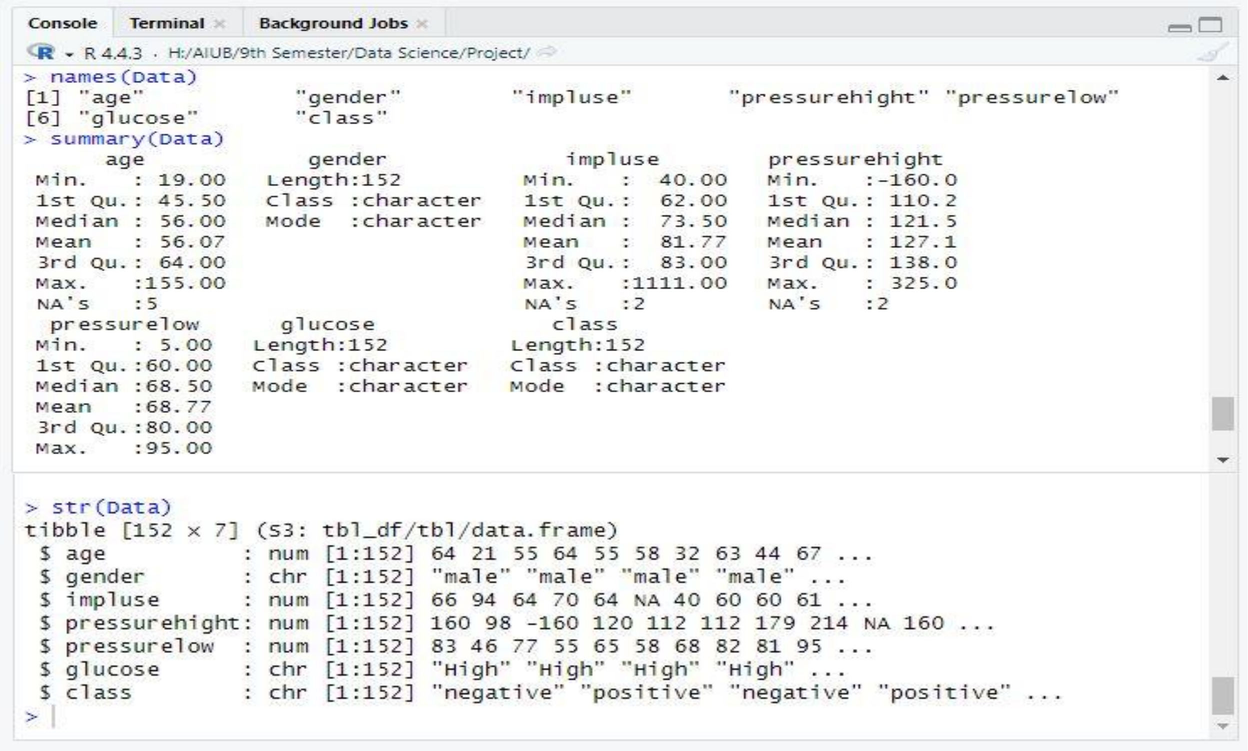
- **Description:** This is the summary of the dataset

Code:



```
names(Data)
summary(Data)
str(Data)
```

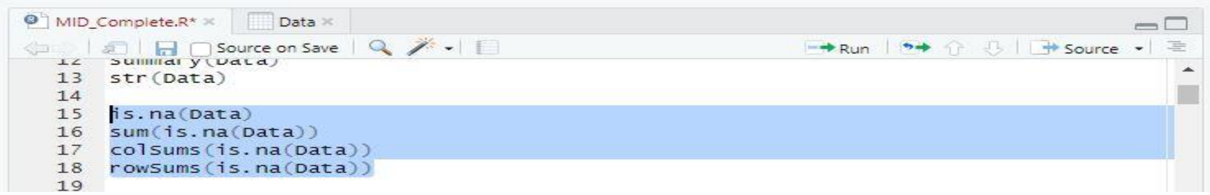
Output:



```
> names(Data)
[1] "age"            "gender"         "impluse"        "pressurehight" "pressurelow"
[6] "glucose"        "class"
> summary(Data)
      age            gender              impluse          pressurehight
 Min.   : 19.00   Length:152         Min.   :  40.00   Min.   :-160.0
 1st Qu.: 45.50   Class :character   1st Qu.:  62.00   1st Qu.: 110.2
 Median : 56.00   Mode  :character   Median :  73.50   Median : 121.5
 Mean   : 56.07                      Mean   :  81.77   Mean   : 127.1
 3rd Qu.: 64.00                      3rd Qu.:  83.00   3rd Qu.: 138.0
 Max.   :155.00                      Max.   :1111.00   Max.   : 325.0
 NA's   :5                           NA's   :2         NA's   :2
  pressurelow      glucose             class
 Min.   : 5.00   Length:152         Length:152
 1st Qu.:60.00   Class :character   Class :character
 Median :68.50   Mode  :character   Mode  :character
 Mean   :68.77
 3rd Qu.:80.00
 Max.   :95.00

> str(Data)
tibble [152 × 7] (S3: tbl_df/tbl/data.frame)
 $ age          : num [1:152] 64 21 55 64 55 58 32 63 44 67 ...
 $ gender       : chr [1:152] "male" "male" "male" "male" ...
 $ impluse      : num [1:152] 66 94 64 70 64 NA 40 60 60 61 ...
 $ pressurehight: num [1:152] 160 98 -160 120 112 112 179 214 NA 160 ...
 $ pressurelow  : num [1:152] 83 46 77 55 65 58 68 82 81 95 ...
 $ glucose      : chr [1:152] "High" "High" "High" "High" ...
 $ class        : chr [1:152] "negative" "positive" "negative" "positive" ...
>
```

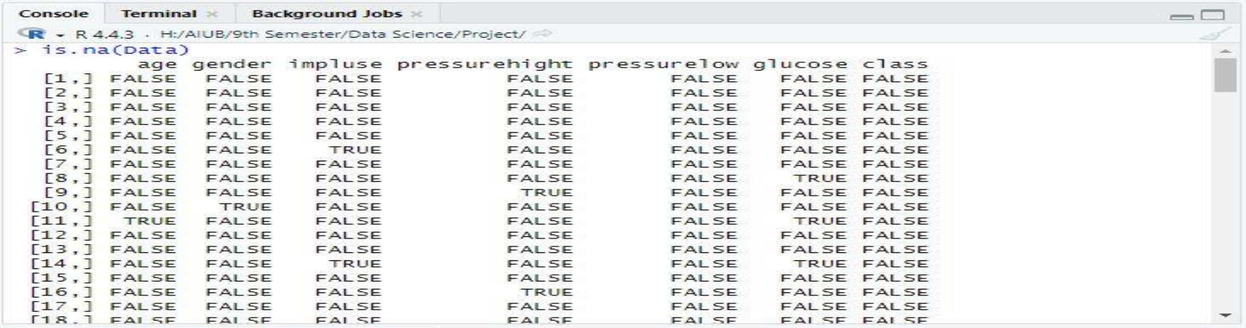
- **Description:** To show the values that are missing from the dataset

Code:

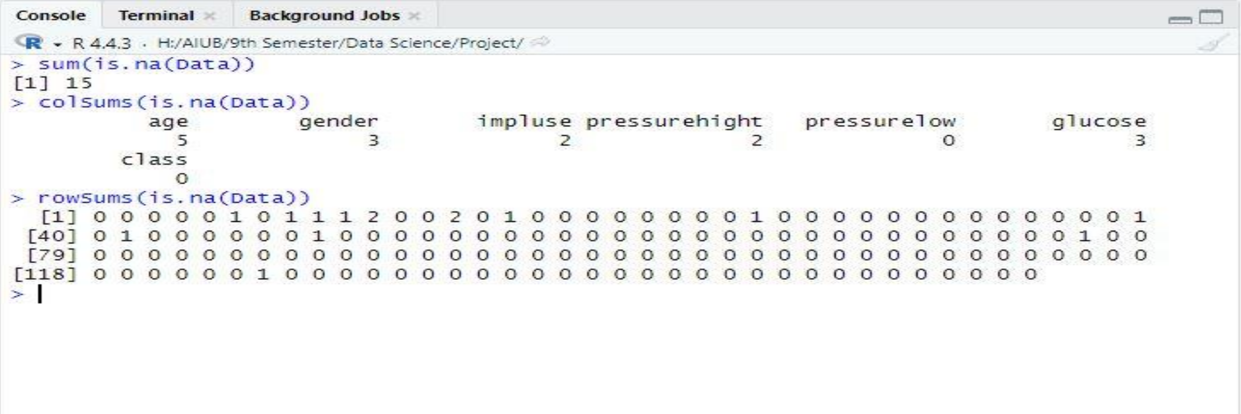


```
str(Data)

is.na(Data)
sum(is.na(Data))
colSums(is.na(Data))
rowSums(is.na(Data))
```

Output:



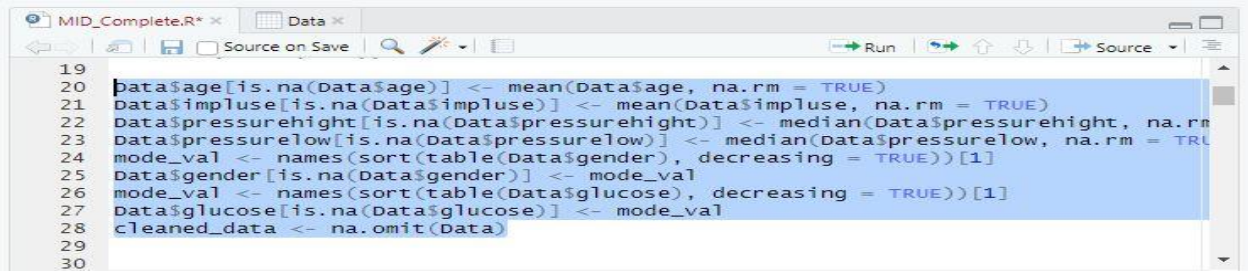
```
> is.na(Data)
        age gender impluse pressurehight pressurelow glucose class
  [1,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
  [2,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
  [3,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
  [4,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
  [5,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
  [6,] FALSE  FALSE    TRUE         FALSE       FALSE   FALSE FALSE
  [7,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
  [8,] FALSE  FALSE   FALSE         FALSE       FALSE    TRUE FALSE
  [9,] FALSE  FALSE   FALSE          TRUE       FALSE   FALSE FALSE
 [10,] FALSE   TRUE   FALSE         FALSE       FALSE   FALSE FALSE
 [11,]  TRUE  FALSE   FALSE         FALSE       FALSE    TRUE FALSE
 [12,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
 [13,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
 [14,] FALSE  FALSE    TRUE         FALSE       FALSE    TRUE FALSE
 [15,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
 [16,] FALSE  FALSE   FALSE          TRUE       FALSE   FALSE FALSE
 [17,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
 [18,] FALSE  FALSE   FALSE         FALSE       FALSE   FALSE FALSE
```



```
> sum(is.na(Data))
[1] 15
> colSums(is.na(Data))
          age        gender       impluse pressurehight   pressurelow       glucose 
            5             3             2             2             0             3 
        class 
            0 
> rowSums(is.na(Data))
  [1] 0 0 0 0 0 1 0 1 1 1 2 0 0 2 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
 [40] 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
 [79] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[118] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> |
```

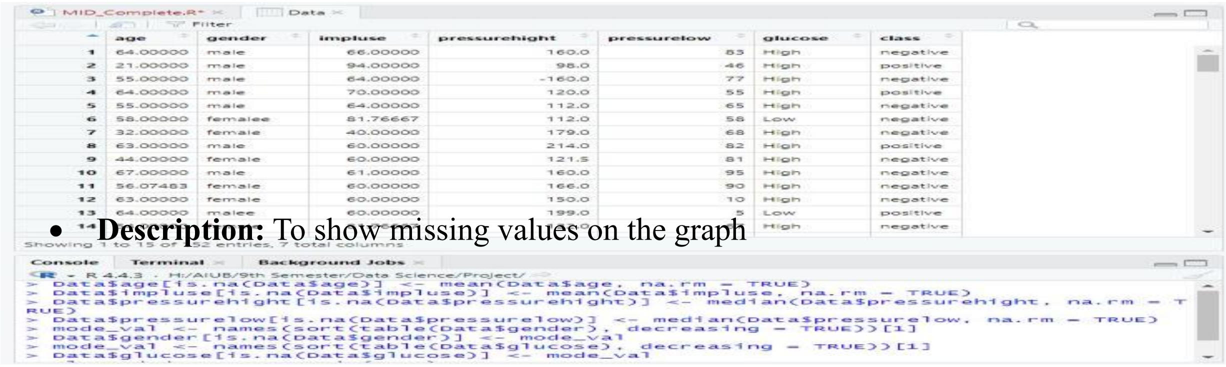
- **Description:** To handle the values that are missing

Code:



```
Data$age[is.na(Data$age)] <- mean(Data$age, na.rm = TRUE)
Data$impluse[is.na(Data$impluse)] <- mean(Data$impluse, na.rm = TRUE)
Data$pressurehight[is.na(Data$pressurehight)] <- median(Data$pressurehight, na.rm
Data$pressurelow[is.na(Data$pressurelow)] <- median(Data$pressurelow, na.rm = TRU
mode_val <- names(sort(table(Data$gender), decreasing = TRUE))[1]
Data$gender[is.na(Data$gender)] <- mode_val
mode_val <- names(sort(table(Data$glucose), decreasing = TRUE))[1]
Data$glucose[is.na(Data$glucose)] <- mode_val
cleaned_data <- na.omit(Data)
```

Output:



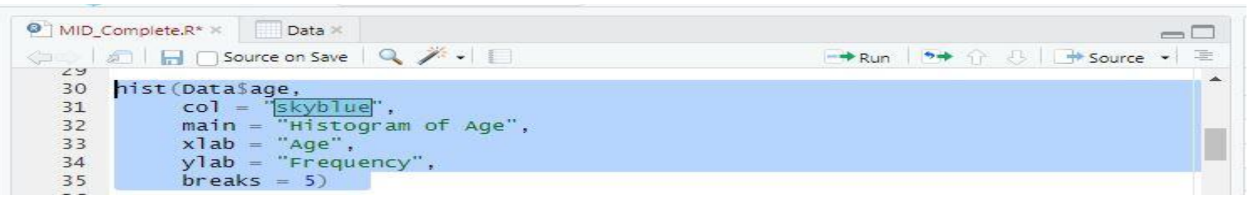
	age	gender	impluse	pressurehight	pressurelow	glucose	class
1	64.00000	male	66.00000	160.0	83	High	negative
2	21.00000	male	94.00000	98.0	46	High	positive
3	55.00000	male	64.00000	-160.0	77	High	negative
4	64.00000	male	70.00000	120.0	55	High	positive
5	55.00000	male	64.00000	112.0	65	High	negative
6	58.00000	femalee	81.76667	112.0	58	Low	negative
7	32.00000	female	40.00000	179.0	68	High	negative
8	63.00000	male	60.00000	214.0	82	High	positive
9	44.00000	female	60.00000	121.5	81	High	negative
10	67.00000	male	61.00000	160.0	95	High	negative
11	56.07483	female	60.00000	166.0	90	High	negative
12	63.00000	female	60.00000	150.0	10	High	negative
13	64.00000	malee	60.00000	199.0	5	Low	positive
14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	High	negative

Showing 1 to 15 of 152 entries, 7 total columns

```
> Data$age[is.na(Data$age)] <- mean(Data$age, na.rm = TRUE)
> Data$impluse[is.na(Data$impluse)] <- mean(Data$impluse, na.rm = TRUE)
> Data$pressurehight[is.na(Data$pressurehight)] <- median(Data$pressurehight, na.rm = T
RUE)
> Data$pressurelow[is.na(Data$pressurelow)] <- median(Data$pressurelow, na.rm = TRUE)
> mode_val <- names(sort(table(Data$gender), decreasing = TRUE))[1]
> Data$gender[is.na(Data$gender)] <- mode_val
> mode_val <- names(sort(table(Data$glucose), decreasing = TRUE))[1]
> Data$glucose[is.na(Data$glucose)] <- mode_val
```

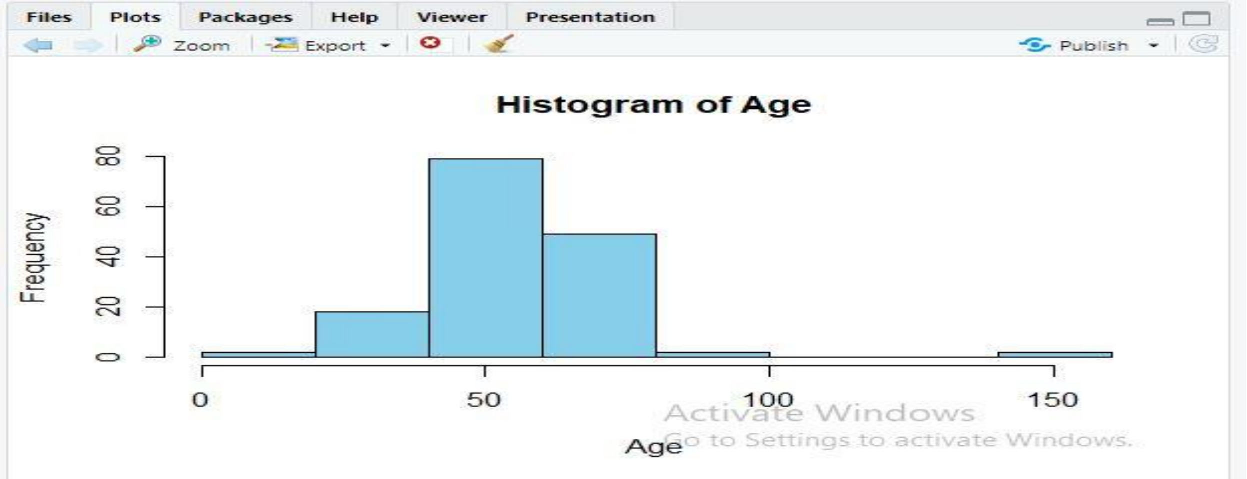
- **Description:** To show missing values on the graph

Code:



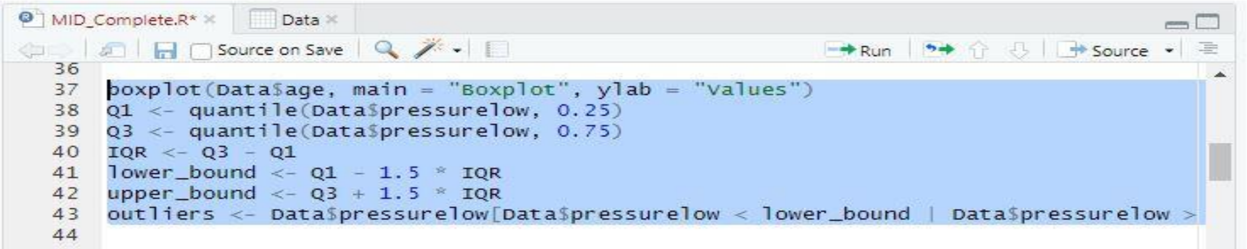
```
hist(Data$age,
     col = "skyblue",
     main = "Histogram of Age",
     xlab = "Age",
     ylab = "Frequency",
     breaks = 5)
```

Output:



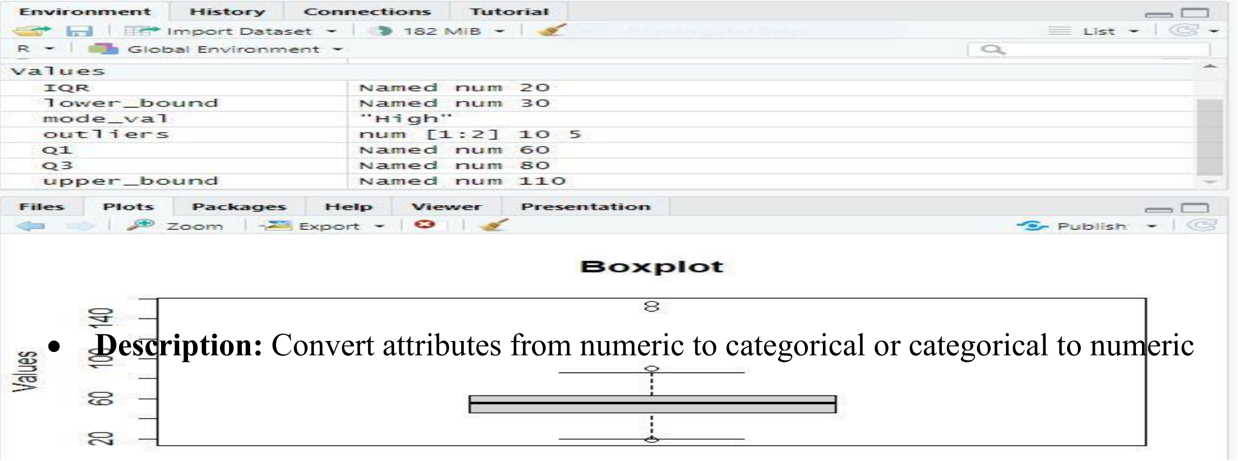
Description: To show outliers in the dataset and handle the outliers

Code:



```
boxplot(Data$age, main = "Boxplot", ylab = "Values")
Q1 <- quantile(Data$pressurelow, 0.25)
Q3 <- quantile(Data$pressurelow, 0.75)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
outliers <- Data$pressurelow[Data$pressurelow < lower_bound | Data$pressurelow >
```

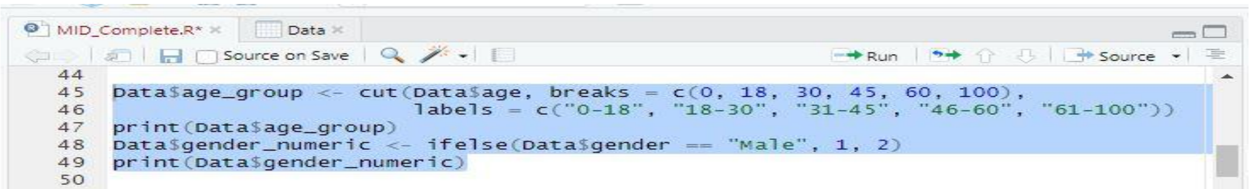
Output:



```
Values
  IQR                  Named num 20
  lower_bound          Named num 30
  mode_val             "High"
  outliers             num [1:2] 10 5
  Q1                   Named num 60
  Q3                   Named num 80
  upper_bound          Named num 110
```

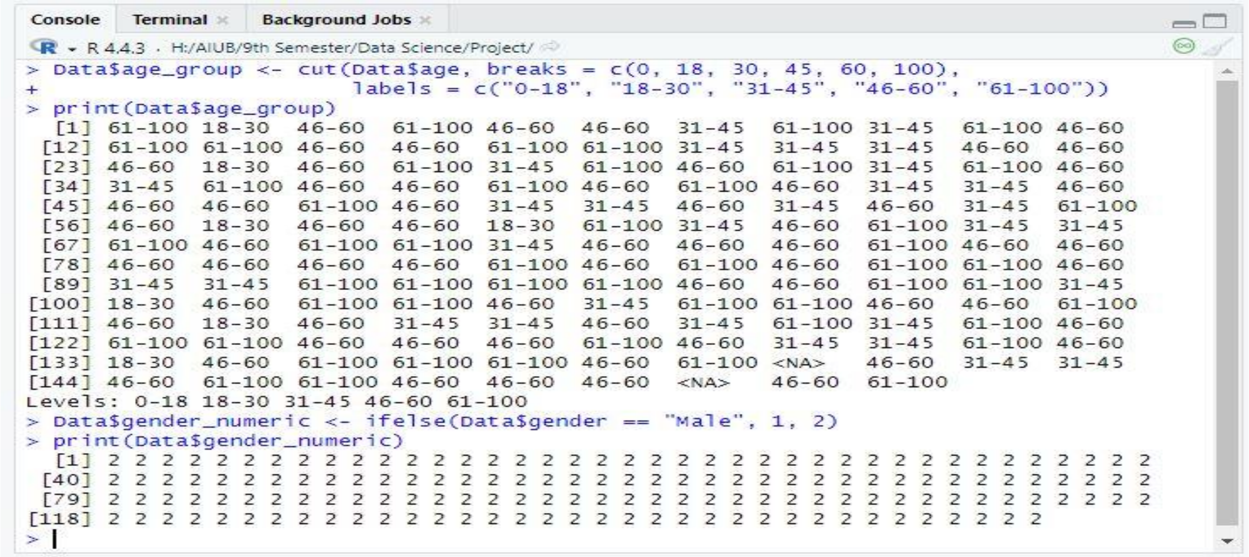
- **Description:** Convert attributes from numeric to categorical or categorical to numeric

Code:



```
Data$age_group <- cut(Data$age, breaks = c(0, 18, 30, 45, 60, 100),
                      labels = c("0-18", "18-30", "31-45", "46-60", "61-100"))
print(Data$age_group)
Data$gender_numeric <- ifelse(Data$gender == "Male", 1, 2)
print(Data$gender_numeric)
```

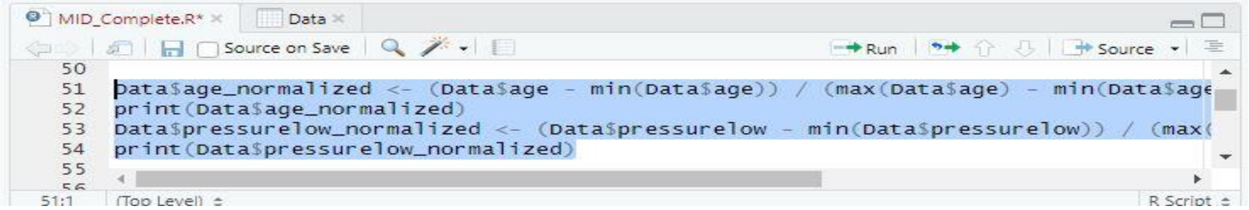
Output:



```
> Data$age_group <- cut(Data$age, breaks = c(0, 18, 30, 45, 60, 100),
+                       labels = c("0-18", "18-30", "31-45", "46-60", "61-100"))
> print(Data$age_group)
  [1] 61-100 18-30  46-60  61-100 46-60  46-60  31-45  61-100 31-45  61-100 46-60
 [12] 61-100 61-100 46-60  46-60  61-100 61-100 31-45  31-45  31-45  46-60  46-60
 [23] 46-60  18-30  46-60  61-100 31-45  61-100 46-60  61-100 31-45  61-100 46-60
 [34] 31-45  61-100 46-60  46-60  61-100 46-60  61-100 46-60  31-45  31-45  46-60
 [45] 46-60  46-60  61-100 46-60  31-45  31-45  46-60  31-45  46-60  31-45  61-100
 [56] 46-60  18-30  46-60  46-60  18-30  61-100 31-45  46-60  61-100 31-45  31-45
 [67] 61-100 46-60  61-100 61-100 31-45  46-60  46-60  46-60  61-100 46-60  46-60
 [78] 46-60  46-60  46-60  46-60  61-100 46-60  61-100 46-60  61-100 61-100 46-60
 [89] 31-45  31-45  61-100 61-100 61-100 61-100 46-60  46-60  61-100 61-100 31-45
[100] 18-30  46-60  61-100 61-100 46-60  31-45  61-100 61-100 46-60  46-60  61-100
[111] 46-60  18-30  46-60  31-45  31-45  46-60  31-45  61-100 31-45  61-100 46-60
[122] 61-100 61-100 46-60  46-60  46-60  61-100 46-60  31-45  31-45  61-100 46-60
[133] 18-30  46-60  61-100 61-100 61-100 46-60  61-100 <NA>   46-60  31-45  31-45
[144] 46-60  61-100 61-100 46-60  46-60  46-60  <NA>   46-60  61-100
Levels: 0-18 18-30 31-45 46-60 61-100
> Data$gender_numeric <- ifelse(Data$gender == "Male", 1, 2)
> print(Data$gender_numeric)
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [40] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [79] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[118] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
> |
```

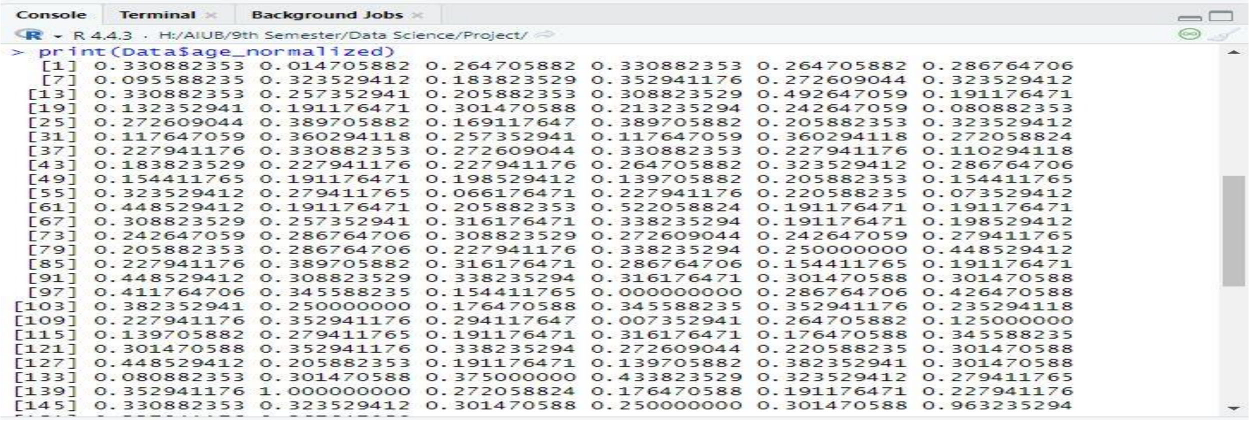
- **Description:** Apply the normalization method for any continuous attribute

Code:



```
Data$age_normalized <- (Data$age - min(Data$age)) / (max(Data$age) - min(Data$age
print(Data$age_normalized)
Data$pressurelow_normalized <- (Data$pressurelow - min(Data$pressurelow)) / (max(
print(Data$pressurelow_normalized)
```

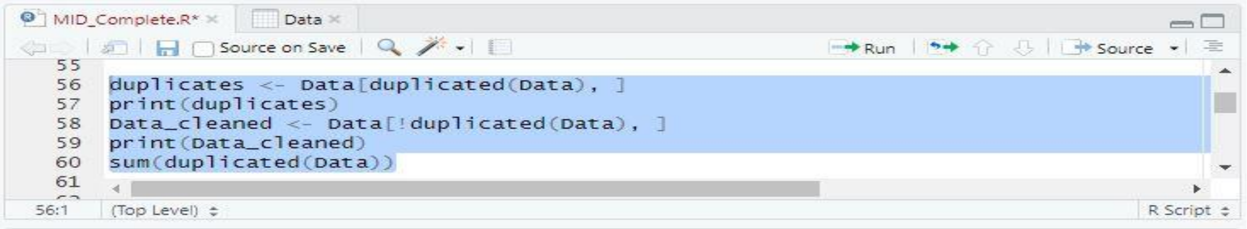
Output:



```
> print(Data$age_normalized)
  [1] 0.330882353 0.014705882 0.264705882 0.330882353 0.264705882 0.286764706
  [7] 0.095588235 0.323529412 0.183823529 0.352941176 0.272609044 0.323529412
 [13] 0.330882353 0.257352941 0.205882353 0.308823529 0.492647059 0.191176471
 [19] 0.132352941 0.191176471 0.301470588 0.213235294 0.242647059 0.080882353
 [25] 0.272609044 0.389705882 0.169117647 0.389705882 0.205882353 0.323529412
 [31] 0.117647059 0.360294118 0.257352941 0.117647059 0.360294118 0.272058824
 [37] 0.227941176 0.330882353 0.272609044 0.330882353 0.227941176 0.110294118
 [43] 0.183823529 0.227941176 0.227941176 0.264705882 0.323529412 0.286764706
 [49] 0.154411765 0.191176471 0.198529412 0.139705882 0.205882353 0.154411765
 [55] 0.323529412 0.279411765 0.066176471 0.227941176 0.220588235 0.073529412
 [61] 0.448529412 0.191176471 0.205882353 0.522058824 0.191176471 0.191176471
 [67] 0.308823529 0.257352941 0.316176471 0.338235294 0.191176471 0.198529412
 [73] 0.242647059 0.286764706 0.308823529 0.272609044 0.242647059 0.279411765
 [79] 0.205882353 0.286764706 0.227941176 0.338235294 0.250000000 0.448529412
 [85] 0.227941176 0.389705882 0.316176471 0.286764706 0.154411765 0.191176471
 [91] 0.448529412 0.308823529 0.338235294 0.316176471 0.301470588 0.301470588
 [97] 0.411764706 0.345588235 0.154411765 0.000000000 0.286764706 0.426470588
[103] 0.382352941 0.250000000 0.176470588 0.345588235 0.352941176 0.235294118
[109] 0.227941176 0.352941176 0.294117647 0.007352941 0.264705882 0.125000000
[115] 0.139705882 0.279411765 0.191176471 0.316176471 0.176470588 0.345588235
[121] 0.301470588 0.352941176 0.338235294 0.272609044 0.220588235 0.301470588
[127] 0.448529412 0.205882353 0.191176471 0.139705882 0.382352941 0.301470588
[133] 0.080882353 0.301470588 0.375000000 0.433823529 0.323529412 0.279411765
[139] 0.352941176 1.000000000 0.272058824 0.176470588 0.191176471 0.227941176
[145] 0.330882353 0.323529412 0.301470588 0.250000000 0.301470588 0.963235294
```

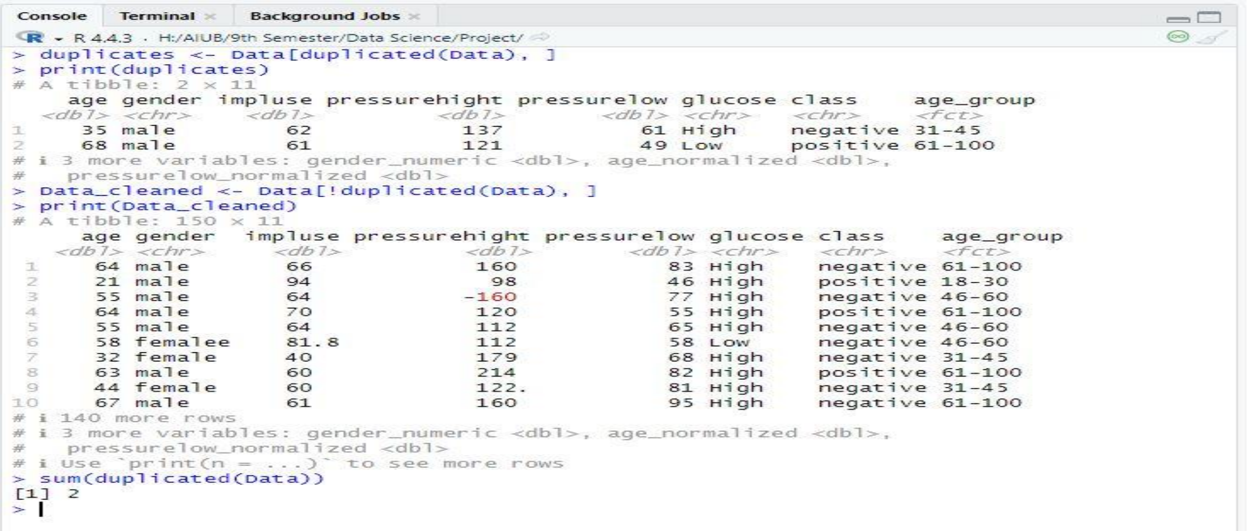
- **Description:** To find and remove duplicate values

Code:



```
duplicates <- Data[duplicated(Data), ]
print(duplicates)
Data_cleaned <- Data[!duplicated(Data), ]
print(Data_cleaned)
sum(duplicated(Data))
```

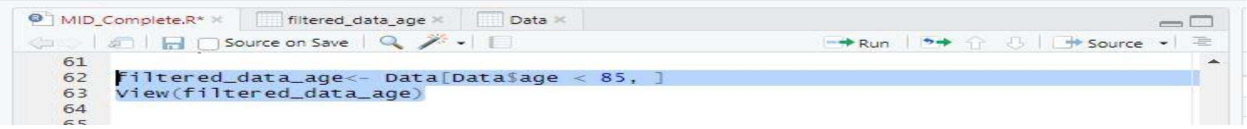
Output:



```
> duplicates <- Data[duplicated(Data), ]
> print(duplicates)
# A tibble: 2 × 11
     age gender impluse pressurehight pressurelow glucose class    age_group
   <dbl> <chr>    <dbl>         <dbl>       <dbl> <chr>   <chr>    <fct>
1     35 male        62           137          61 High    negative 31-45
2     68 male        61           121          49 Low     positive 61-100
# ℹ 3 more variables: gender_numeric <dbl>, age_normalized <dbl>,
#   pressurelow_normalized <dbl>
> Data_cleaned <- Data[!duplicated(Data), ]
> print(Data_cleaned)
# A tibble: 150 × 11
     age gender   impluse pressurehight pressurelow glucose class    age_group
   <dbl> <chr>      <dbl>         <dbl>       <dbl> <chr>   <chr>    <fct>
 1    64 male        66             160          83 High    negative 61-100
 2    21 male        94              98          46 High    positive 18-30
 3    55 male        64            -160          77 High    negative 46-60
 4    64 male        70             120          55 High    positive 61-100
 5    55 male        64             112          65 High    negative 46-60
 6    58 femalee     81.8           112          58 Low     negative 46-60
 7    32 female      40             179          68 High    negative 31-45
 8    63 male        60             214          82 High    positive 61-100
 9    44 female      60             122.         81 High    negative 31-45
10    67 male        61             160          95 High    negative 61-100
# ℹ 140 more rows
# ℹ 3 more variables: gender_numeric <dbl>, age_normalized <dbl>,
#   pressurelow_normalized <dbl>
# ℹ Use `print(n = ...)` to see more rows
> sum(duplicated(Data))
[1] 2
> 
```

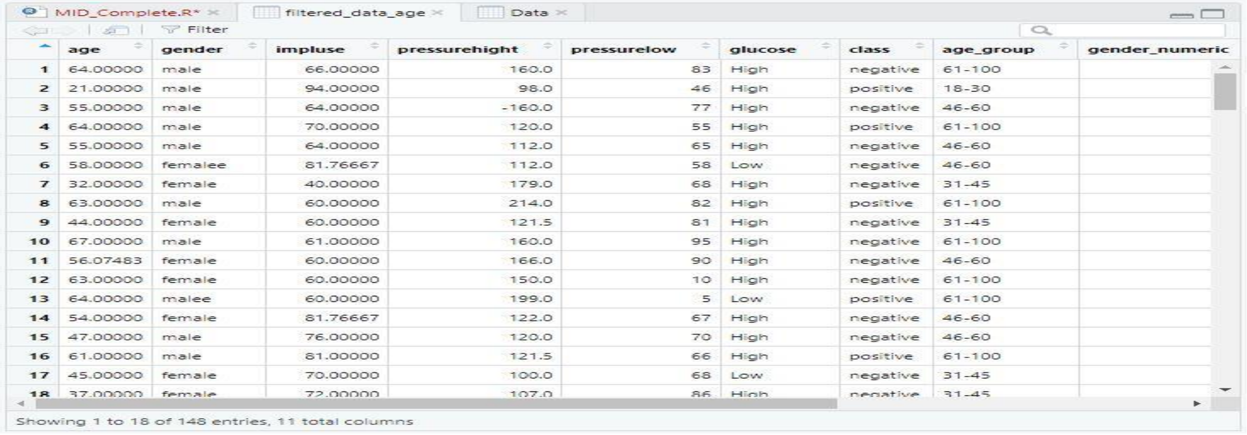
- **Description:** Apply filtering methods to filter the data.

Code:



```
Filtered_data_age<- Data[Data$age < 85, ]
View(filtered_data_age)
```

Output:

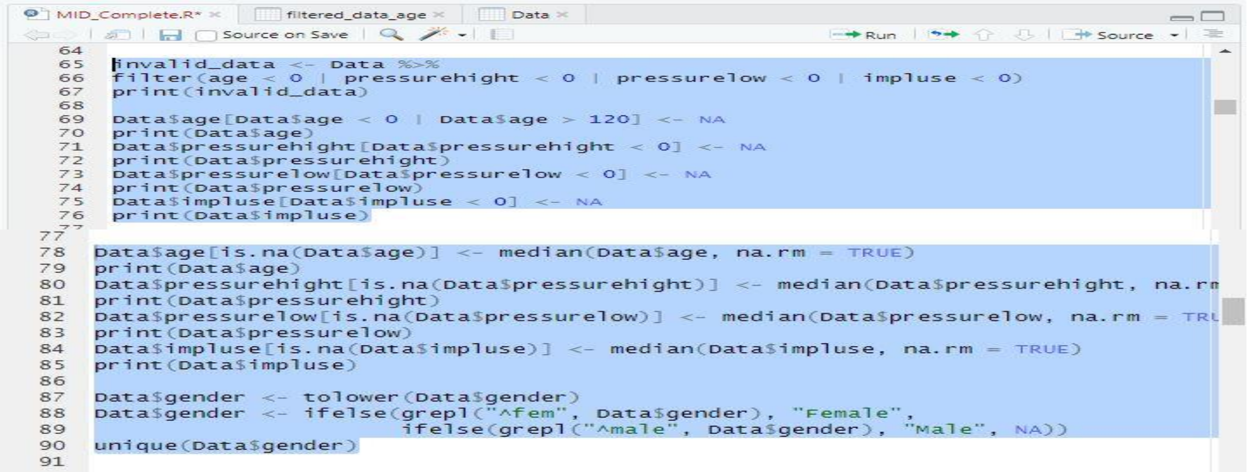


	age	gender	impluse	pressurehight	pressurelow	glucose	class	age_group	gender_numeric
1	64.00000	male	66.00000	160.0	83	High	negative	61-100	
2	21.00000	male	94.00000	98.0	46	High	positive	18-30	
3	55.00000	male	64.00000	-160.0	77	High	negative	46-60	
4	64.00000	male	70.00000	120.0	55	High	positive	61-100	
5	55.00000	male	64.00000	112.0	65	High	negative	46-60	
6	58.00000	femalee	81.76667	112.0	58	Low	negative	46-60	
7	32.00000	female	40.00000	179.0	68	High	negative	31-45	
8	63.00000	male	60.00000	214.0	82	High	positive	61-100	
9	44.00000	female	60.00000	121.5	81	High	negative	31-45	
10	67.00000	male	61.00000	160.0	95	High	negative	61-100	
11	56.07483	female	60.00000	166.0	90	High	negative	46-60	
12	63.00000	female	60.00000	150.0	10	High	negative	61-100	
13	64.00000	malee	60.00000	199.0	5	Low	positive	61-100	
14	54.00000	female	81.76667	122.0	67	High	negative	46-60	
15	47.00000	male	76.00000	120.0	70	High	negative	46-60	
16	61.00000	male	81.00000	121.5	66	High	positive	61-100	
17	45.00000	female	70.00000	100.0	68	Low	negative	31-45	
18	37.00000	female	72.00000	107.0	86	High	negative	31-45	

Showing 1 to 18 of 148 entries, 11 total columns

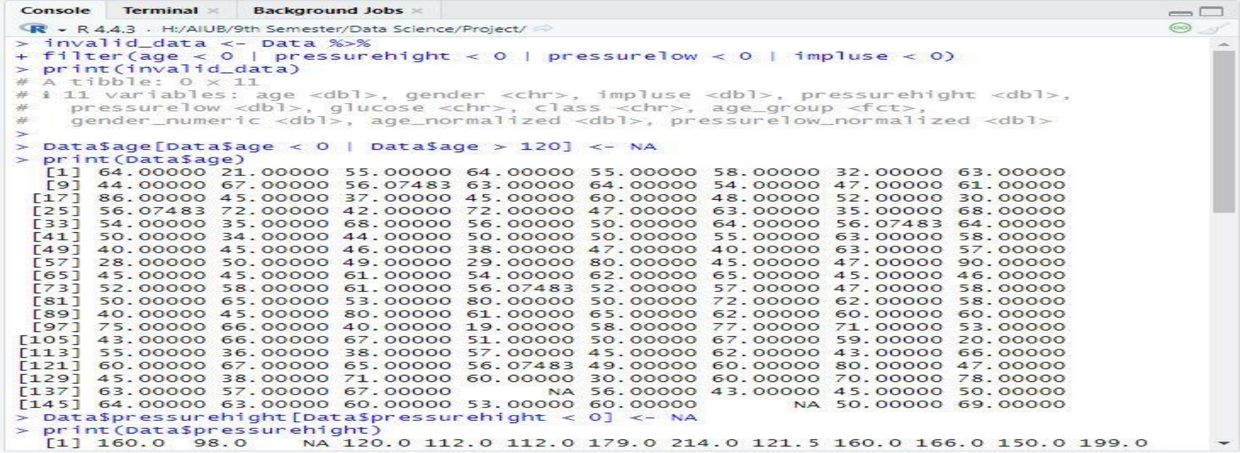
- **Description:** Detect invalid data in the data set and handle those values

Code:

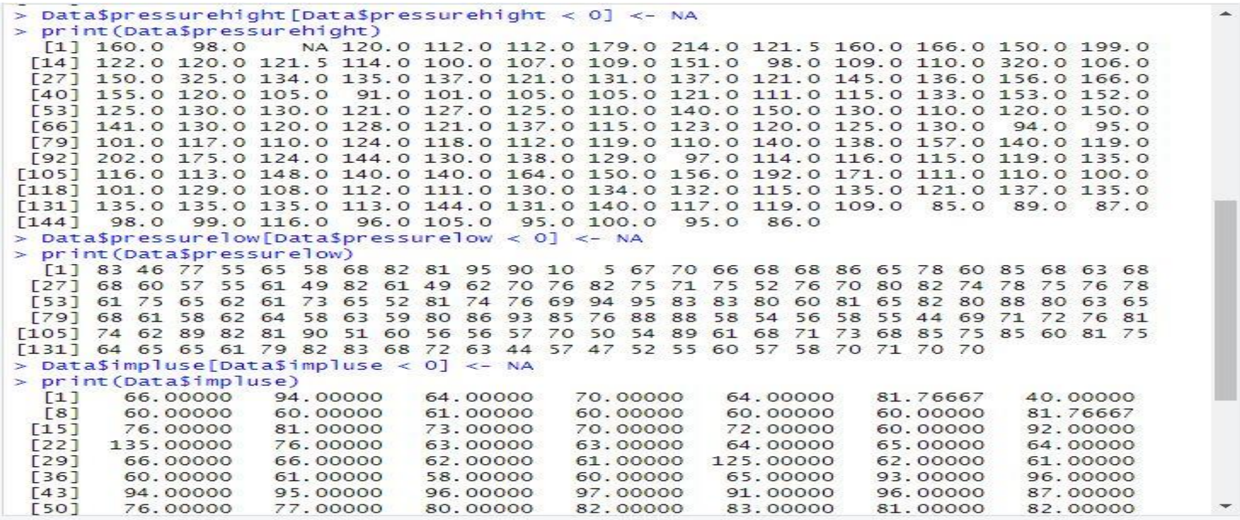


```
invalid_data <- Data %>%
filter(age < 0 | pressurehight < 0 | impluse < 0)
print(invalid_data)

Data$age[Data$age < 0 | Data$age > 120] <- NA
print(Data$age)
Data$pressurehight[Data$pressurehight < 0] <- NA
print(Data$pressurehight)
Data$pressurelow[Data$pressurelow < 0] <- NA
print(Data$pressurelow)
Data$impluse[Data$impluse < 0] <- NA
print(Data$impluse)

Data$age[is.na(Data$age)] <- median(Data$age, na.rm = TRUE)
print(Data$age)
Data$pressurehight[is.na(Data$pressurehight)] <- median(Data$pressurehight, na.rm
print(Data$pressurehight)
Data$pressurelow[is.na(Data$pressurelow)] <- median(Data$pressurelow, na.rm = TRU
print(Data$pressurelow)
Data$impluse[is.na(Data$impluse)] <- median(Data$impluse, na.rm = TRUE)
print(Data$impluse)

Data$gender <- tolower(Data$gender)
Data$gender <- ifelse(grepl("^fem", Data$gender), "Female",
                      ifelse(grepl("^male", Data$gender), "Male", NA))
unique(Data$gender)
```

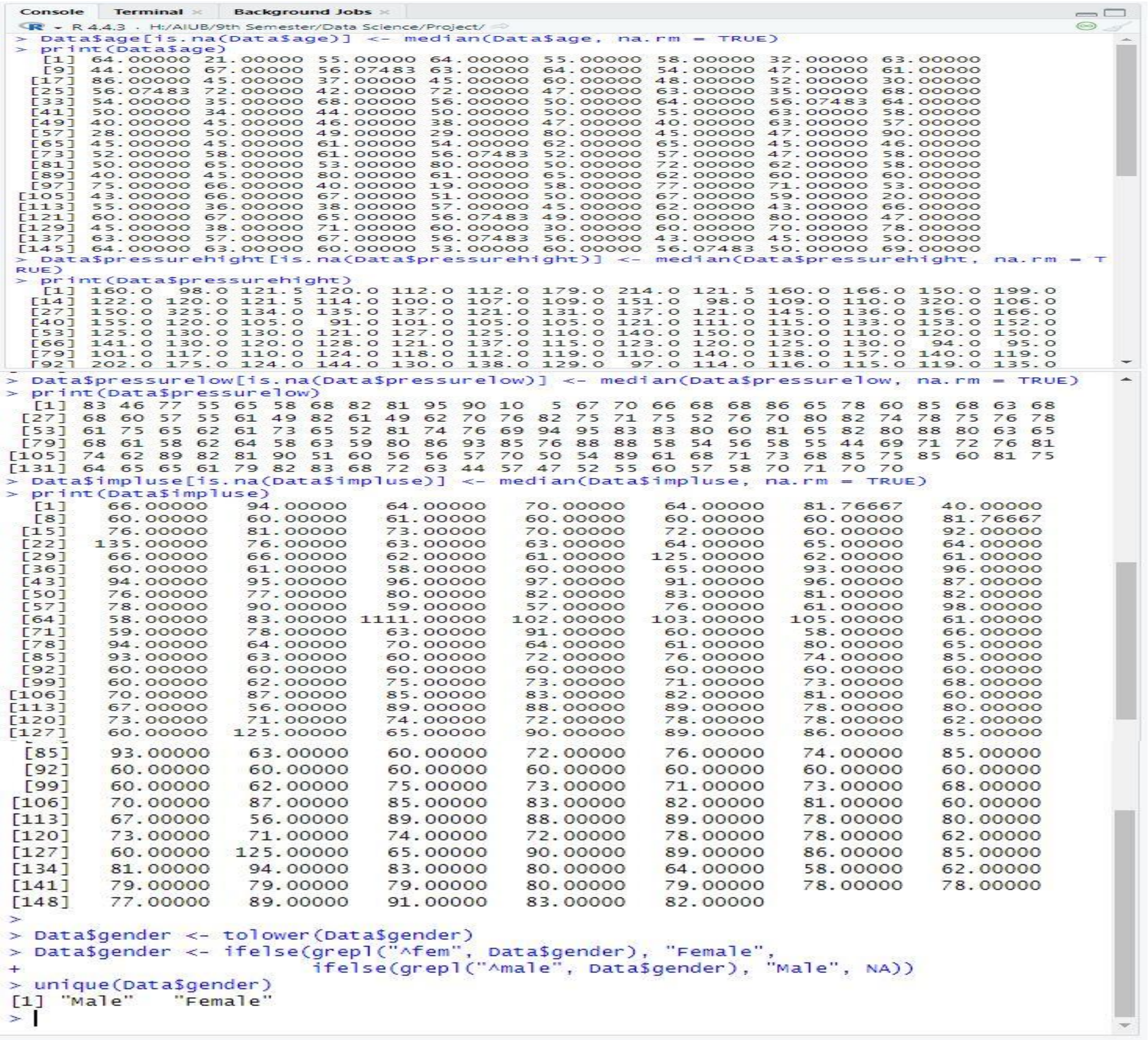
Output:



```
> invalid_data <- Data %>%
+ filter(age < 0 | pressurehight < 0 | impluse < 0)
> print(invalid_data)
# A tibble: 0 × 11
# ℹ 11 variables: age <dbl>, gender <chr>, impluse <dbl>, pressurehight <dbl>,
#   pressurelow <dbl>, glucose <chr>, class <chr>, age_group <fct>,
#   gender_numeric <dbl>, age_normalized <dbl>, pressurelow_normalized <dbl>
>
> Data$age[Data$age < 0 | Data$age > 120] <- NA
> print(Data$age)
  [1] 64.00000 21.00000 55.00000 64.00000 55.00000 58.00000 32.00000 63.00000
  [9] 44.00000 67.00000 56.07483 63.00000 64.00000 54.00000 47.00000 61.00000
 [17] 86.00000 45.00000 37.00000 45.00000 60.00000 48.00000 52.00000 30.00000
 [25] 56.07483 72.00000 42.00000 72.00000 47.00000 63.00000 35.00000 68.00000
 [33] 54.00000 35.00000 68.00000 56.00000 50.00000 64.00000 56.07483 64.00000
 [41] 50.00000 34.00000 44.00000 50.00000 50.00000 55.00000 63.00000 58.00000
 [49] 40.00000 45.00000 46.00000 38.00000 47.00000 40.00000 63.00000 57.00000
 [57] 28.00000 50.00000 49.00000 29.00000 80.00000 45.00000 47.00000 90.00000
 [65] 45.00000 45.00000 61.00000 54.00000 62.00000 65.00000 45.00000 46.00000
 [73] 52.00000 58.00000 61.00000 56.07483 52.00000 57.00000 47.00000 58.00000
 [81] 50.00000 65.00000 53.00000 80.00000 50.00000 72.00000 62.00000 58.00000
 [89] 40.00000 45.00000 80.00000 61.00000 65.00000 62.00000 60.00000 60.00000
 [97] 75.00000 66.00000 40.00000 19.00000 58.00000 77.00000 71.00000 53.00000
[105] 43.00000 66.00000 67.00000 51.00000 50.00000 67.00000 59.00000 20.00000
[113] 55.00000 36.00000 38.00000 57.00000 45.00000 62.00000 43.00000 66.00000
[121] 60.00000 67.00000 65.00000 56.07483 49.00000 60.00000 80.00000 47.00000
[129] 45.00000 38.00000 71.00000 60.00000 30.00000 60.00000 70.00000 78.00000
[137] 63.00000 57.00000 67.00000       NA 56.00000 43.00000 45.00000 50.00000
[145] 64.00000 63.00000 60.00000 53.00000 60.00000       NA 50.00000 69.00000
> Data$pressurehight[Data$pressurehight < 0] <- NA
> print(Data$pressurehight)
  [1] 160.0  98.0    NA 120.0 112.0 112.0 179.0 214.0 121.5 160.0 166.0 150.0 199.0
```



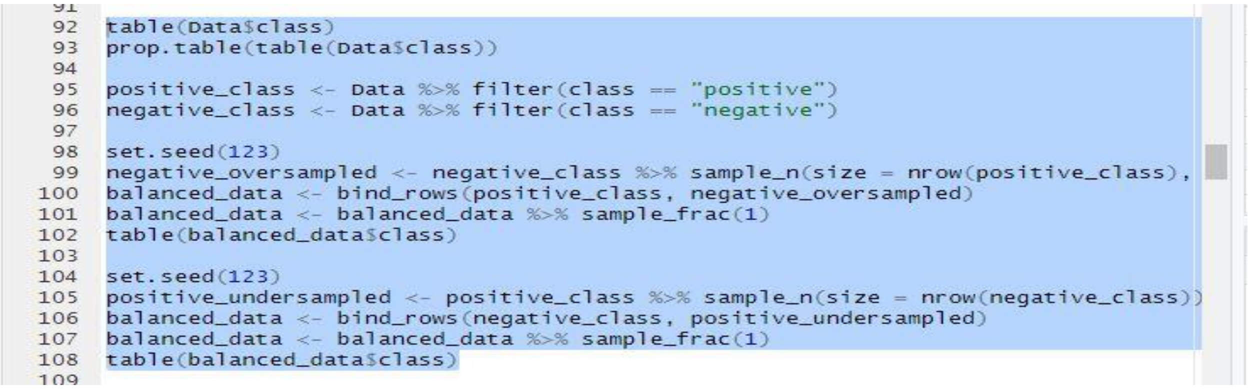
```
> Data$pressurehight[Data$pressurehight < 0] <- NA
> print(Data$pressurehight)
  [1] 160.0  98.0    NA 120.0 112.0 112.0 179.0 214.0 121.5 160.0 166.0 150.0 199.0
 [14] 122.0 120.0 121.5 114.0 100.0 107.0 109.0 151.0  98.0 109.0 110.0 320.0 106.0
 [27] 150.0 325.0 134.0 135.0 137.0 121.0 131.0 137.0 121.0 145.0 136.0 156.0 166.0
 [40] 155.0 120.0 105.0  91.0 101.0 105.0 105.0 121.0 111.0 115.0 133.0 153.0 152.0
 [53] 125.0 130.0 130.0 121.0 127.0 125.0 110.0 140.0 150.0 130.0 110.0 120.0 150.0
 [66] 141.0 130.0 120.0 128.0 121.0 137.0 115.0 123.0 120.0 125.0 130.0  94.0  95.0
 [79] 101.0 117.0 110.0 124.0 118.0 112.0 119.0 110.0 140.0 138.0 157.0 140.0 119.0
 [92] 202.0 175.0 124.0 144.0 130.0 138.0 129.0  97.0 114.0 116.0 115.0 119.0 135.0
[105] 116.0 113.0 148.0 140.0 140.0 164.0 150.0 156.0 192.0 171.0 111.0 110.0 100.0
[118] 101.0 129.0 108.0 112.0 111.0 130.0 134.0 132.0 115.0 135.0 121.0 137.0 135.0
[131] 135.0 135.0 135.0 113.0 144.0 131.0 140.0 117.0 119.0 109.0  85.0  89.0  87.0
[144]  98.0  99.0 116.0  96.0 105.0  95.0 100.0  95.0  86.0
> Data$pressurelow[Data$pressurelow < 0] <- NA
> print(Data$pressurelow)
  [1] 83 46 77 55 65 58 68 82 81 95 90 10  5 67 70 66 68 68 86 65 78 60 85 68 63 68
 [27] 68 60 57 55 61 49 82 61 49 62 70 76 82 75 71 75 52 76 70 80 82 74 78 75 76 78
 [53] 61 75 65 62 61 73 65 52 81 74 76 69 94 95 83 83 80 60 81 65 82 80 88 80 63 65
 [79] 68 61 58 62 64 58 63 59 80 86 93 85 76 88 88 58 54 56 58 55 44 69 71 72 76 81
[105] 74 62 89 82 81 90 51 60 56 56 57 70 50 54 89 61 68 71 73 68 85 75 85 60 81 75
[131] 64 65 65 61 79 82 83 68 72 63 44 57 47 52 55 60 57 58 70 71 70 70
> Data$impluse[Data$impluse < 0] <- NA
> print(Data$impluse)
  [1]  66.00000  94.00000  64.00000  70.00000  64.00000  81.76667  40.00000
  [8]  60.00000  60.00000  61.00000  60.00000  60.00000  60.00000  81.76667
 [15]  76.00000  81.00000  73.00000  70.00000  72.00000  60.00000  92.00000
 [22] 135.00000  76.00000  63.00000  63.00000  64.00000  65.00000  64.00000
 [29]  66.00000  66.00000  62.00000  61.00000 125.00000  62.00000  61.00000
 [36]  60.00000  61.00000  58.00000  60.00000  65.00000  93.00000  96.00000
 [43]  94.00000  95.00000  96.00000  97.00000  91.00000  96.00000  87.00000
 [50]  76.00000  77.00000  80.00000  82.00000  83.00000  81.00000  82.00000
```



```
> Data$age[is.na(Data$age)] <- median(Data$age, na.rm = TRUE)
> print(Data$age)
  [1] 64.00000 21.00000 55.00000 64.00000 55.00000 58.00000 32.00000 63.00000
  [9] 44.00000 67.00000 56.07483 63.00000 64.00000 54.00000 47.00000 61.00000
 [17] 86.00000 45.00000 37.00000 45.00000 60.00000 48.00000 52.00000 30.00000
 [25] 56.07483 72.00000 42.00000 72.00000 47.00000 63.00000 35.00000 68.00000
 [33] 54.00000 35.00000 68.00000 56.00000 50.00000 64.00000 56.07483 64.00000
 [41] 50.00000 34.00000 44.00000 50.00000 50.00000 55.00000 63.00000 58.00000
 [49] 40.00000 45.00000 46.00000 38.00000 47.00000 40.00000 63.00000 57.00000
 [57] 28.00000 50.00000 49.00000 29.00000 80.00000 45.00000 47.00000 90.00000
 [65] 45.00000 45.00000 61.00000 54.00000 62.00000 65.00000 45.00000 46.00000
 [73] 52.00000 58.00000 61.00000 56.07483 52.00000 57.00000 47.00000 58.00000
 [81] 50.00000 65.00000 53.00000 80.00000 50.00000 72.00000 62.00000 58.00000
 [89] 40.00000 45.00000 80.00000 61.00000 65.00000 62.00000 60.00000 60.00000
 [97] 75.00000 66.00000 40.00000 19.00000 58.00000 77.00000 71.00000 53.00000
[105] 43.00000 66.00000 67.00000 51.00000 50.00000 67.00000 59.00000 20.00000
[113] 55.00000 36.00000 38.00000 57.00000 45.00000 62.00000 43.00000 66.00000
[121] 60.00000 67.00000 65.00000 56.07483 49.00000 60.00000 80.00000 47.00000
[129] 45.00000 38.00000 71.00000 60.00000 30.00000 60.00000 70.00000 78.00000
[137] 63.00000 57.00000 67.00000 56.07483 56.00000 43.00000 45.00000 50.00000
[145] 64.00000 63.00000 60.00000 53.00000 60.00000 56.07483 50.00000 69.00000
> Data$pressurehight[is.na(Data$pressurehight)] <- median(Data$pressurehight, na.rm = T
RUE)
> print(Data$pressurehight)
  [1] 160.0  98.0 121.5 120.0 112.0 112.0 179.0 214.0 121.5 160.0 166.0 150.0 199.0
 [14] 122.0 120.0 121.5 114.0 100.0 107.0 109.0 151.0  98.0 109.0 110.0 320.0 106.0
 [27] 150.0 325.0 134.0 135.0 137.0 121.0 131.0 137.0 121.0 145.0 136.0 156.0 166.0
 [40] 155.0 120.0 105.0  91.0 101.0 105.0 105.0 121.0 111.0 115.0 133.0 153.0 152.0
 [53] 125.0 130.0 130.0 121.0 127.0 125.0 110.0 140.0 150.0 130.0 110.0 120.0 150.0
 [66] 141.0 130.0 120.0 128.0 121.0 137.0 115.0 123.0 120.0 125.0 130.0  94.0  95.0
 [79] 101.0 117.0 110.0 124.0 118.0 112.0 119.0 110.0 140.0 138.0 157.0 140.0 119.0
 [92] 202.0 175.0 124.0 144.0 130.0 138.0 129.0  97.0 114.0 116.0 115.0 119.0 135.0
> Data$pressurelow[is.na(Data$pressurelow)] <- median(Data$pressurelow, na.rm = TRUE)
> print(Data$pressurelow)
  [1] 83 46 77 55 65 58 68 82 81 95 90 10  5 67 70 66 68 68 86 65 78 60 85 68 63 68
 [27] 68 60 57 55 61 49 82 61 49 62 70 76 82 75 71 75 52 76 70 80 82 74 78 75 76 78
 [53] 61 75 65 62 61 73 65 52 81 74 76 69 94 95 83 83 80 60 81 65 82 80 88 80 63 65
 [79] 68 61 58 62 64 58 63 59 80 86 93 85 76 88 88 58 54 56 58 55 44 69 71 72 76 81
[105] 74 62 89 82 81 90 51 60 56 56 57 70 50 54 89 61 68 71 73 68 85 75 85 60 81 75
[131] 64 65 65 61 79 82 83 68 72 63 44 57 47 52 55 60 57 58 70 71 70 70
> Data$impluse[is.na(Data$impluse)] <- median(Data$impluse, na.rm = TRUE)
> print(Data$impluse)
  [1]    66.00000    94.00000    64.00000    70.00000    64.00000    81.76667    40.00000
  [8]    60.00000    60.00000    61.00000    60.00000    60.00000    60.00000    81.76667
 [15]    76.00000    81.00000    73.00000    70.00000    72.00000    60.00000    92.00000
 [22]   135.00000    76.00000    63.00000    63.00000    64.00000    65.00000    64.00000
 [29]    66.00000    66.00000    62.00000    61.00000   125.00000    62.00000    61.00000
 [36]    60.00000    61.00000    58.00000    60.00000    65.00000    93.00000    96.00000
 [43]    94.00000    95.00000    96.00000    97.00000    91.00000    96.00000    87.00000
 [50]    76.00000    77.00000    80.00000    82.00000    83.00000    81.00000    82.00000
 [57]    78.00000    90.00000    59.00000    57.00000    76.00000    61.00000    98.00000
 [64]    58.00000    83.00000  1111.00000   102.00000   103.00000   105.00000    61.00000
 [71]    59.00000    78.00000    63.00000    91.00000    60.00000    58.00000    66.00000
 [78]    94.00000    64.00000    70.00000    64.00000    61.00000    80.00000    65.00000
 [85]    93.00000    63.00000    60.00000    72.00000    76.00000    74.00000    85.00000
 [92]    60.00000    60.00000    60.00000    60.00000    60.00000    60.00000    60.00000
 [99]    60.00000    62.00000    75.00000    73.00000    71.00000    73.00000    68.00000
[106]    70.00000    87.00000    85.00000    83.00000    82.00000    81.00000    60.00000
[113]    67.00000    56.00000    89.00000    88.00000    89.00000    78.00000    80.00000
[120]    73.00000    71.00000    74.00000    72.00000    78.00000    78.00000    62.00000
[127]    60.00000   125.00000    65.00000    90.00000    89.00000    86.00000    85.00000
 [85]    93.00000    63.00000    60.00000    72.00000    76.00000    74.00000    85.00000
 [92]    60.00000    60.00000    60.00000    60.00000    60.00000    60.00000    60.00000
 [99]    60.00000    62.00000    75.00000    73.00000    71.00000    73.00000    68.00000
[106]    70.00000    87.00000    85.00000    83.00000    82.00000    81.00000    60.00000
[113]    67.00000    56.00000    89.00000    88.00000    89.00000    78.00000    80.00000
[120]    73.00000    71.00000    74.00000    72.00000    78.00000    78.00000    62.00000
[127]    60.00000   125.00000    65.00000    90.00000    89.00000    86.00000    85.00000
[134]    81.00000    94.00000    83.00000    80.00000    64.00000    58.00000    62.00000
[141]    79.00000    79.00000    79.00000    80.00000    79.00000    78.00000    78.00000
[148]    77.00000    89.00000    91.00000    83.00000    82.00000
>
> Data$gender <- tolower(Data$gender)
> Data$gender <- ifelse(grepl("^fem", Data$gender), "Female",
+                       ifelse(grepl("^male", Data$gender), "Male", NA))
> unique(Data$gender)
[1] "Male"   "Female"
> 
```

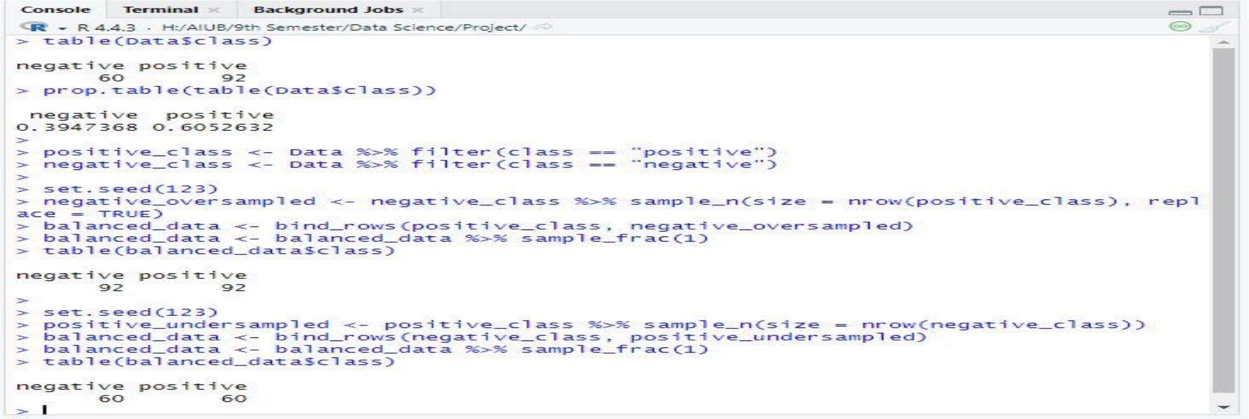
- **Description:** Convert the imbalanced data set into the balanced data set

Code:



```r
table(Data$class)
prop.table(table(Data$class))

positive_class <- Data %>% filter(class == "positive")
negative_class <- Data %>% filter(class == "negative")

set.seed(123)
negative_oversampled <- negative_class %>% sample_n(size = nrow(positive_class),
balanced_data <- bind_rows(positive_class, negative_oversampled)
balanced_data <- balanced_data %>% sample_frac(1)
table(balanced_data$class)

set.seed(123)
positive_undersampled <- positive_class %>% sample_n(size = nrow(negative_class))
balanced_data <- bind_rows(negative_class, positive_undersampled)
balanced_data <- balanced_data %>% sample_frac(1)
table(balanced_data$class)
```

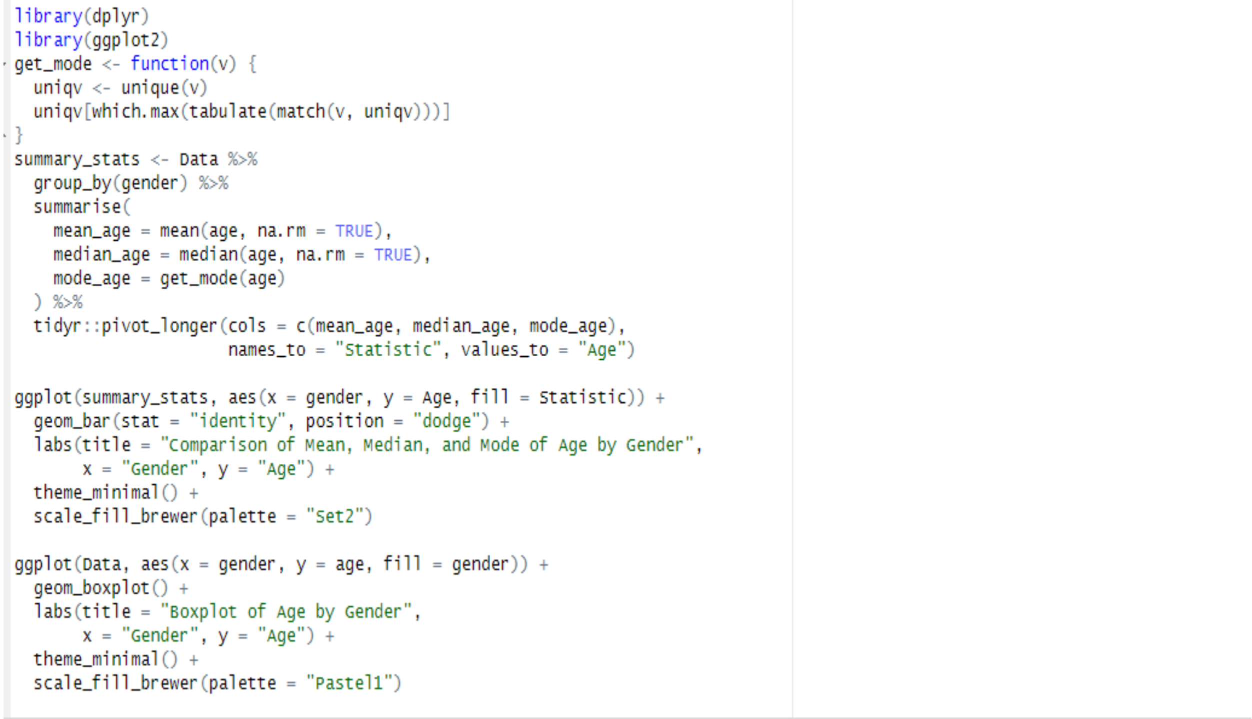
Output:



```
> table(Data$class)

negative positive 
      60       92 
> prop.table(table(Data$class))

 negative  positive 
0.3947368 0.6052632 
> 
> positive_class <- Data %>% filter(class == "positive")
> negative_class <- Data %>% filter(class == "negative")
> 
> set.seed(123)
> negative_oversampled <- negative_class %>% sample_n(size = nrow(positive_class), repl
ace = TRUE)
> balanced_data <- bind_rows(positive_class, negative_oversampled)
> balanced_data <- balanced_data %>% sample_frac(1)
> table(balanced_data$class)

negative positive 
      92       92 
> 
> set.seed(123)
> positive_undersampled <- positive_class %>% sample_n(size = nrow(negative_class))
> balanced_data <- bind_rows(negative_class, positive_undersampled)
> balanced_data <- balanced_data %>% sample_frac(1)
> table(balanced_data$class)

negative positive 
      60       60 
> 
```

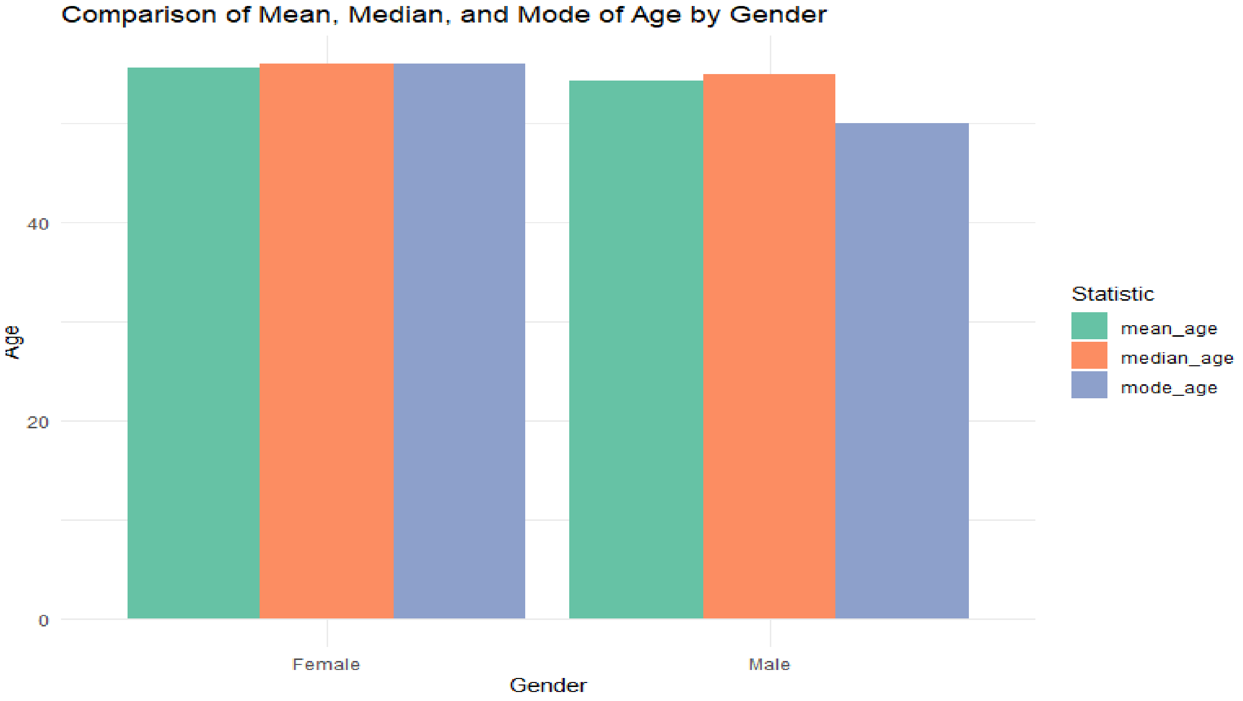
- **Description:** Split the dataset for Training and Testing

Code:



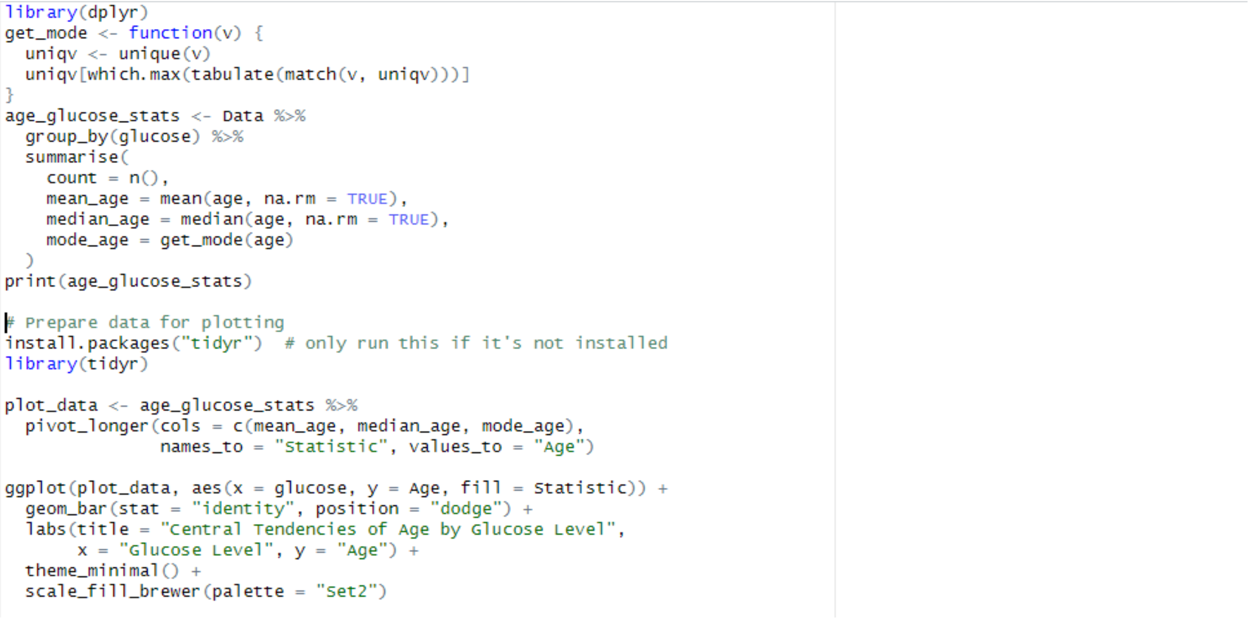
```r
library(dplyr)
library(ggplot2)
get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
summary_stats <- Data %>%
  group_by(gender) %>%
  summarise(
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    mode_age = get_mode(age)
  ) %>%
  tidyr::pivot_longer(cols = c(mean_age, median_age, mode_age),
                      names_to = "Statistic", values_to = "Age")

ggplot(summary_stats, aes(x = gender, y = Age, fill = Statistic)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Comparison of Mean, Median, and Mode of Age by Gender",
       x = "Gender", y = "Age") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")

ggplot(Data, aes(x = gender, y = age, fill = gender)) +
  geom_boxplot() +
  labs(title = "Boxplot of Age by Gender",
       x = "Gender", y = "Age") +
  theme_minimal() +
  scale_fill_brewer(palette = "Pastel1")
```

Output:



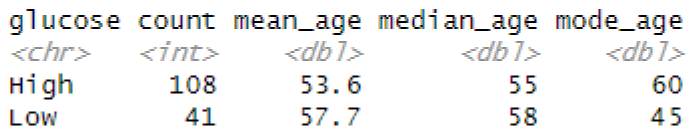
- **Description:** Comparing the central tendency of age across different gender groups using the mean, median, and mode.

Code:

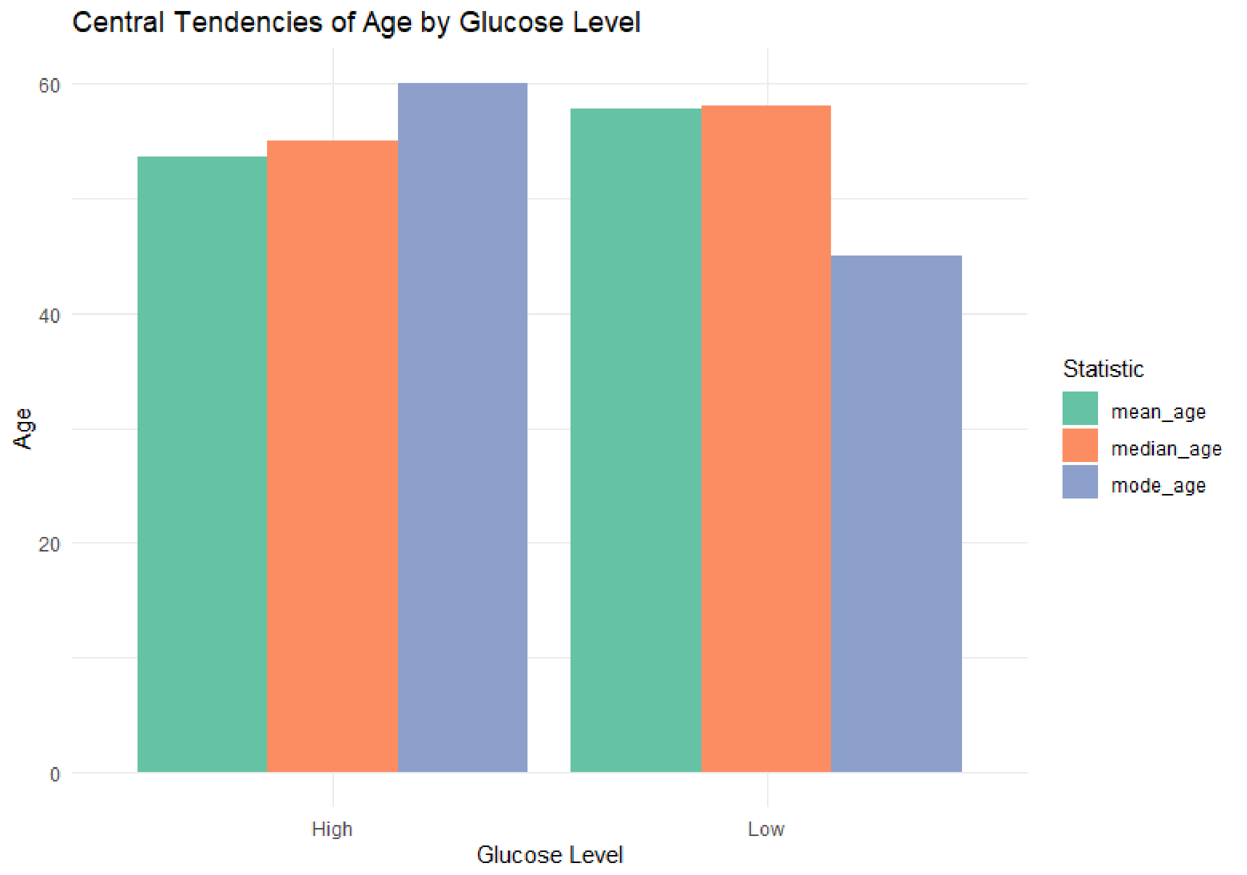


```
library(dplyr)
get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
age_glucose_stats <- Data %>%
  group_by(glucose) %>%
  summarise(
    count = n(),
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    mode_age = get_mode(age)
  )
print(age_glucose_stats)

# Prepare data for plotting
install.packages("tidyr")  # only run this if it's not installed
library(tidyr)

plot_data <- age_glucose_stats %>%
  pivot_longer(cols = c(mean_age, median_age, mode_age),
               names_to = "Statistic", values_to = "Age")

ggplot(plot_data, aes(x = glucose, y = Age, fill = Statistic)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Central Tendencies of Age by Glucose Level",
       x = "Glucose Level", y = "Age") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```

Output:

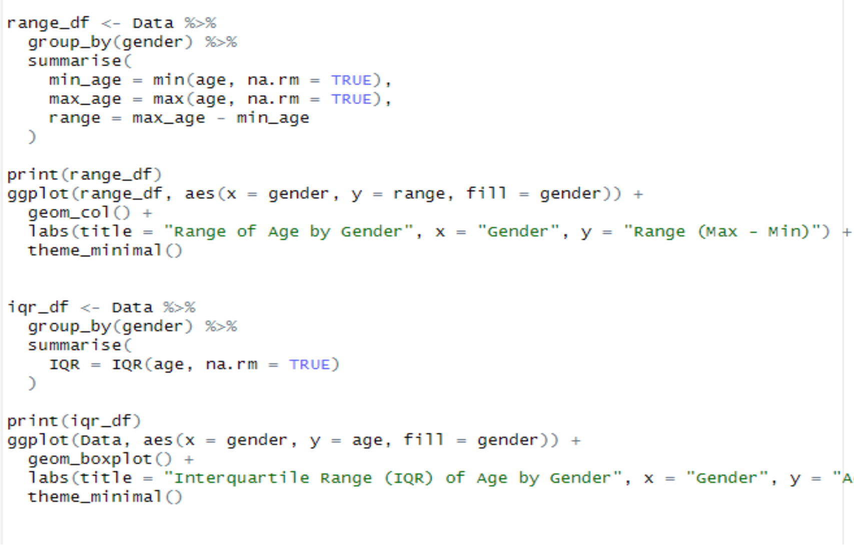


```
glucose count mean_age median_age mode_age
<chr>   <int>    <dbl>      <dbl>    <dbl>
High      108     53.6         55       60
Low        41     57.7         58       45
```

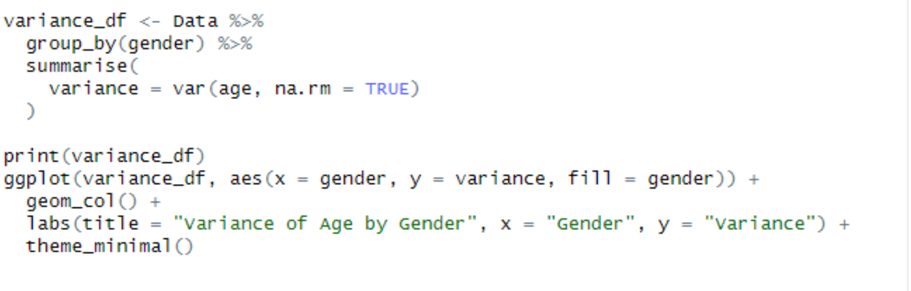


Description: Comparing the age’s central tendency across glucose levels using the mean, median, and mode

Code:

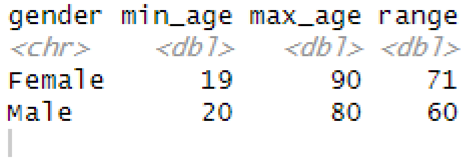


```
range_df <- Data %>%
  group_by(gender) %>%
  summarise(
    min_age = min(age, na.rm = TRUE),
    max_age = max(age, na.rm = TRUE),
    range = max_age - min_age
  )

print(range_df)
ggplot(range_df, aes(x = gender, y = range, fill = gender)) +
  geom_col() +
  labs(title = "Range of Age by Gender", x = "Gender", y = "Range (Max - Min)") +
  theme_minimal()


iqr_df <- Data %>%
  group_by(gender) %>%
  summarise(
    IQR = IQR(age, na.rm = TRUE)
  )

print(iqr_df)
ggplot(Data, aes(x = gender, y = age, fill = gender)) +
  geom_boxplot() +
  labs(title = "Interquartile Range (IQR) of Age by Gender", x = "Gender", y = "Age") +
  theme_minimal()
```

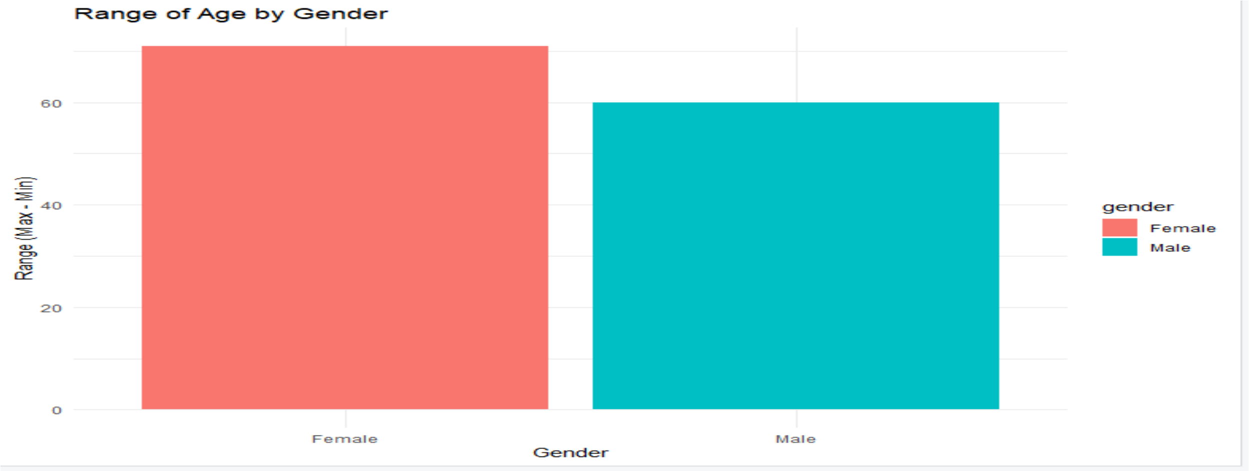


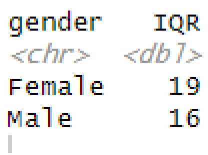
```
variance_df <- Data %>%
  group_by(gender) %>%
  summarise(
    variance = var(age, na.rm = TRUE)
  )

print(variance_df)
ggplot(variance_df, aes(x = gender, y = variance, fill = gender)) +
  geom_col() +
  labs(title = "Variance of Age by Gender", x = "Gender", y = "Variance") +
  theme_minimal()
```

Output:

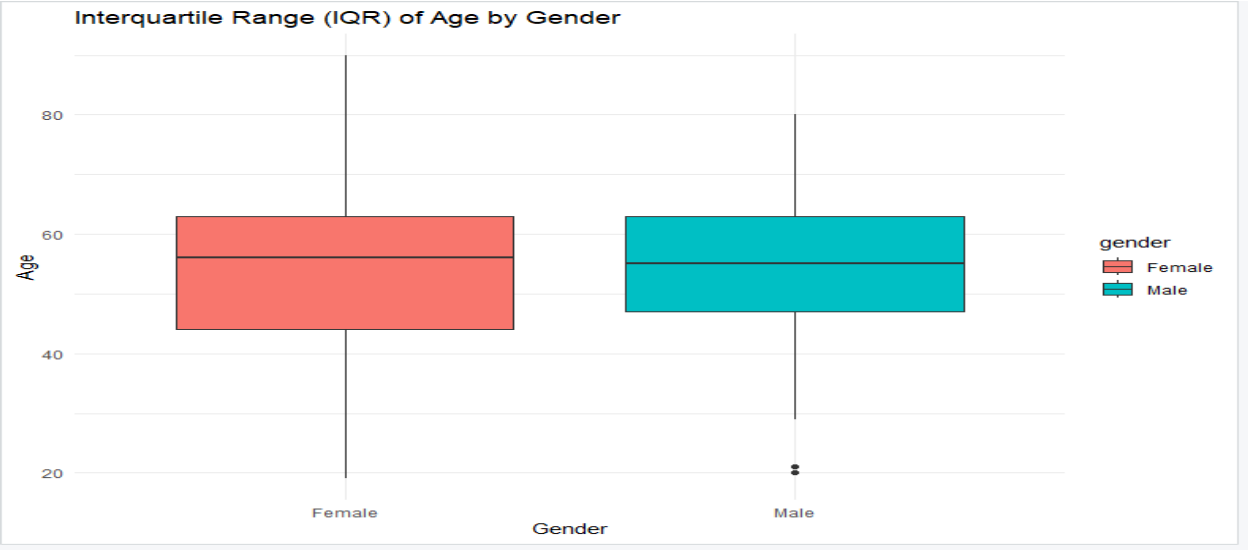


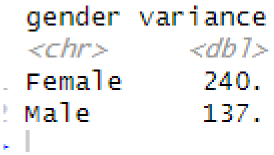
```
gender min_age max_age range
<chr>    <dbl>   <dbl> <dbl>
Female      19      90    71
Male        20      80    60
```



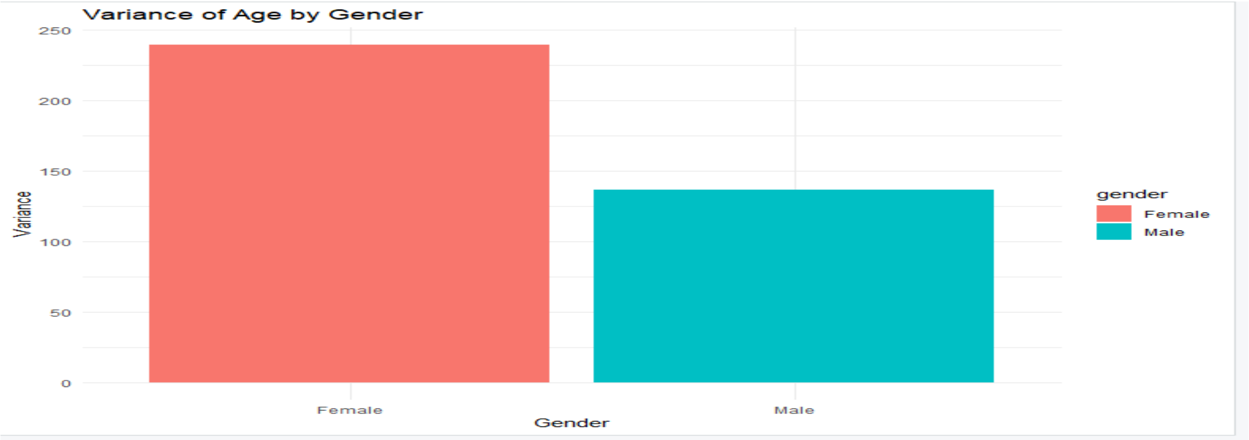


```
gender    IQR
<chr>   <dbl>
Female     19
Male       16
```





```
gender variance
<chr>     <dbl>
Female     240.
Male       137.
```



Description: Comparing the spread of Age across different groups of gender using the Range, IQR, and Variance