

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Department of Computer Science

# **Course Name: Introduction to Data Science**

**Midterm Project**

# **Prepared by-**

|  |  |
| --- | --- |
| Name | ID NUMBER |
| ZARIN TASNIM MIHIKA | 20-43790-2 |
| ABU SYED MOHAMMED SAFWAN | 21-44496-1 |

**Submitted To:**

TOHEDUL ISLAM

Assistant Professor

Faculty of Science Technology

American International University-Bangladesh

**About the dataset:**

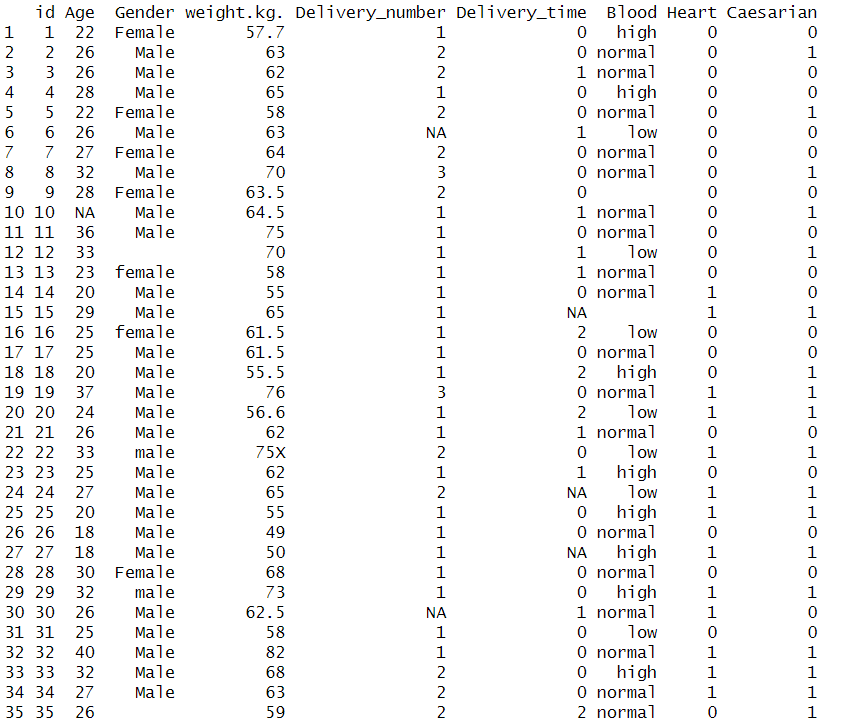
The dataset has details about different people, like their age, gender, weight, and whether they had a Caesarian delivery. It also includes information on their blood pressure and heart condition. However, the data needs some fixing because there are missing values and some mistakes in how the information is written. Once we clean up the data, we can look at it more closely to find out interesting things, like if there's a connection between age or weight and having a Caesarian delivery. We might also be able to predict what factors make a Caesarian delivery more likely. But we have to remember that the data might not be perfect, so we have to be careful with our conclusions.

**Import the data set as csv and print the data set:**

**Code:**



**Output:**



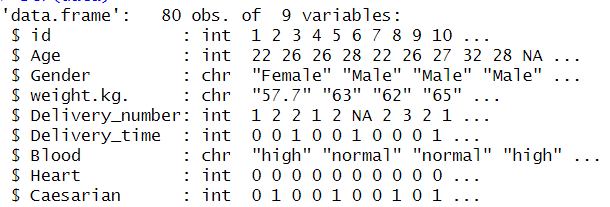
**Description:** Here is the code of importing the dataset as csv file. It is the output of the dataset which is imported in RStudio.

**Structure of Data:**

**Code:**



**Output:**



**Description:**

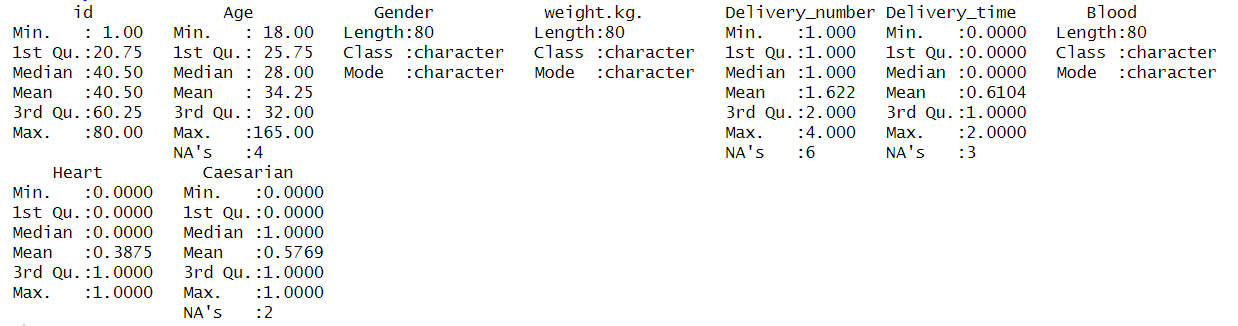
The 'data' data frame consists of 80 rows and 9 columns. The columns include information such as id, Age,Gender,weight.kg.,Delivery\_number, Delivery\_time, Blood, Heart, and Caesarian . It's important to note potential data type inconsistencies, such as 'weight.kg.' being represented as character instead of numeric and some missing values (NA) are present in the "age" column.

**Descriptive Statistics Using summary() Function:**

**Code:**



**Output:**

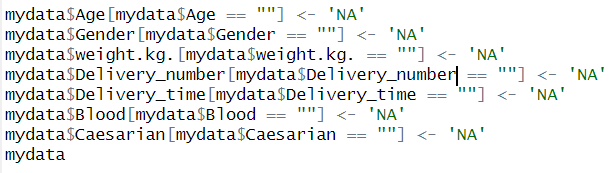


**Description:** Here using descriptive statistic, and the summary() function. In the output min, max, median, and mean are shown for every column.

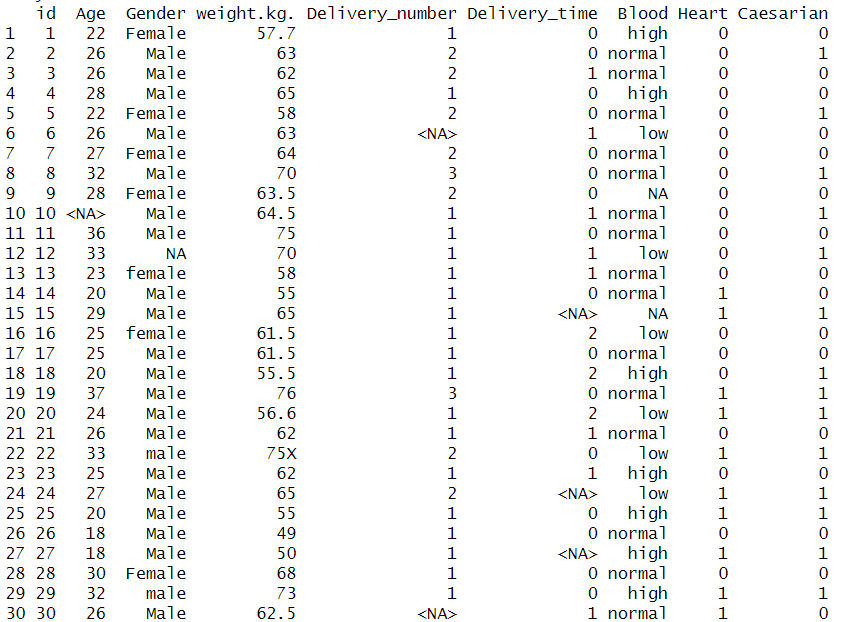
**Outlier Detection as a missing value:**

**Code:**





**Output:**

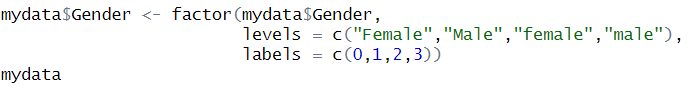


**Description:**

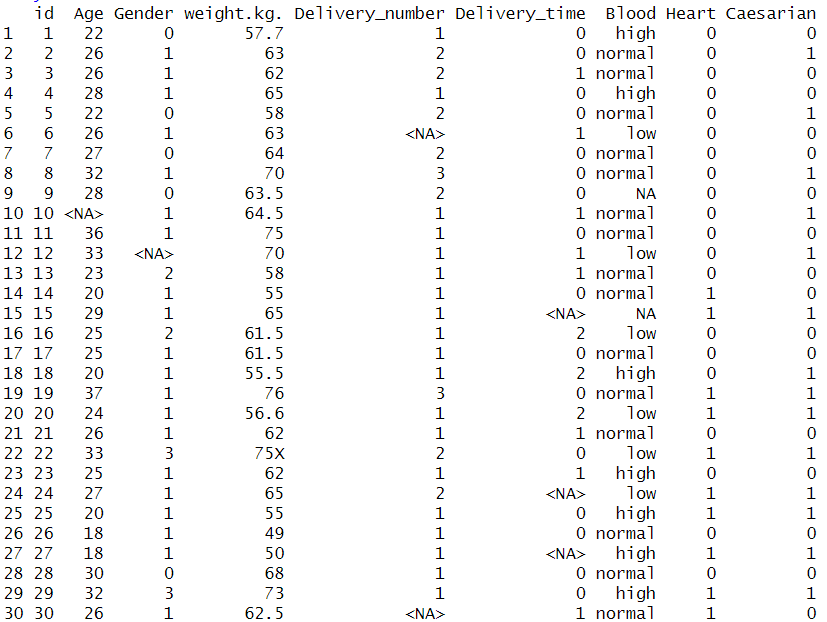
In this code, missing or empty values in the 'Age,' 'Gender,' 'weight.kg.,' 'Delivery\_number,' 'Delivery\_time,' 'Blood,' and 'Caesarian' columns with 'NA' in the 'mydata' dataset. This is a common data preprocessing step to handle missing or unknown values.

**Annotate Female as 0, Male as 1 , female as 2, male as 3 from “Gender” attribute:**

**Code:**



**Output:**

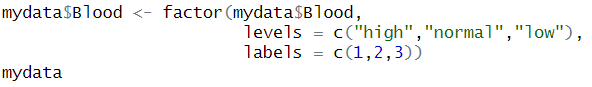


**Description:**

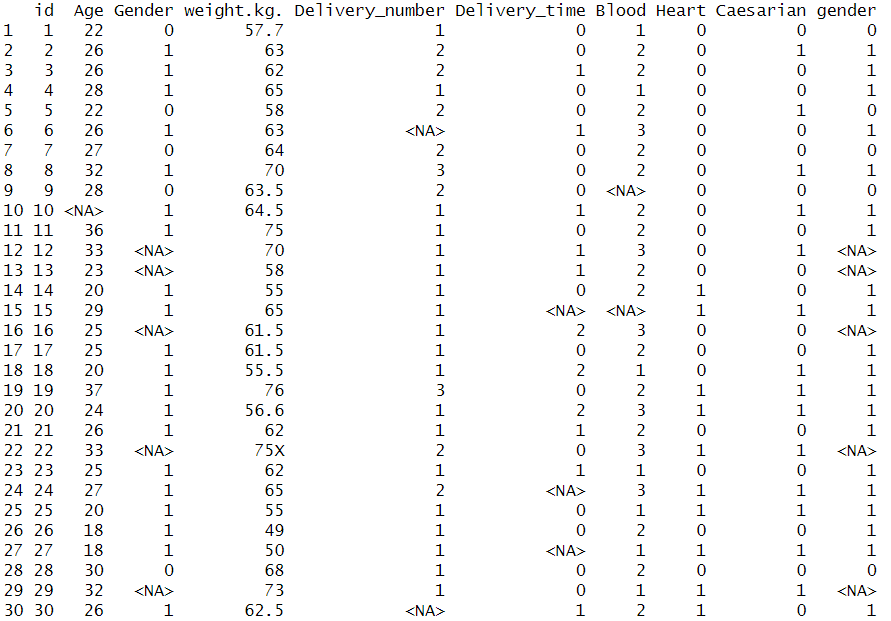
The "gender" column in the "mydata" dataset has been converted into a categorical variable of type "factor”. The levels "Female" and "Male" have been assigned numerical labels. It assigns the levels 0, 1, 2, and 3 to "Female," "Male," "female," and "male" categories, respectively.

**Annotate high as 1, normal as 2, low as 3 from “Blood” attribute:**

**Code:**



**Output:**



**Description:**

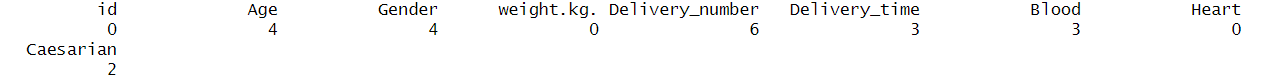
The 'Blood' column in the 'mydata' dataset into a factor variable, assigning numeric levels to different blood pressure categories. Specifically, it assigns the levels 1, 2, and 3 to "high," "normal," and "low" categories, respectively.

**Finding missing values:**

**Code:**



**Output:**



**Description:**

The code provides the count of missing values (NA) in each column of the "mydata" dataset. The resulting output provides a count of missing values for each variable, offering insights into the completeness of the dataset and potential areas that may require attention or imputation.

We will handle these missing values before performing further analysis on the dataset.

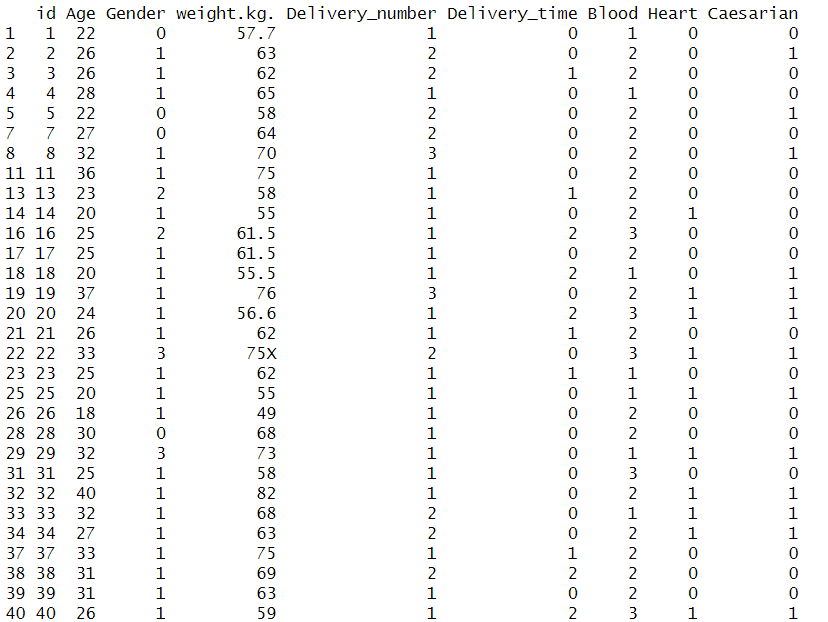
**HANDLING MISSING VALUES**

**Delete the rows with missing values:**

**Code:**



**Output:**



**Description:**

The "removedata" dataset is removing rows with missing values (NA) from the "mydata" dataset. This operation effectively eliminates rows containing incomplete data, resulting in a dataset with complete observations for further analysis or modelin

**'mydata' Dataset as 'meanData'**

**In meanData set we will replace all the missing values with mean value.**

**Code:**

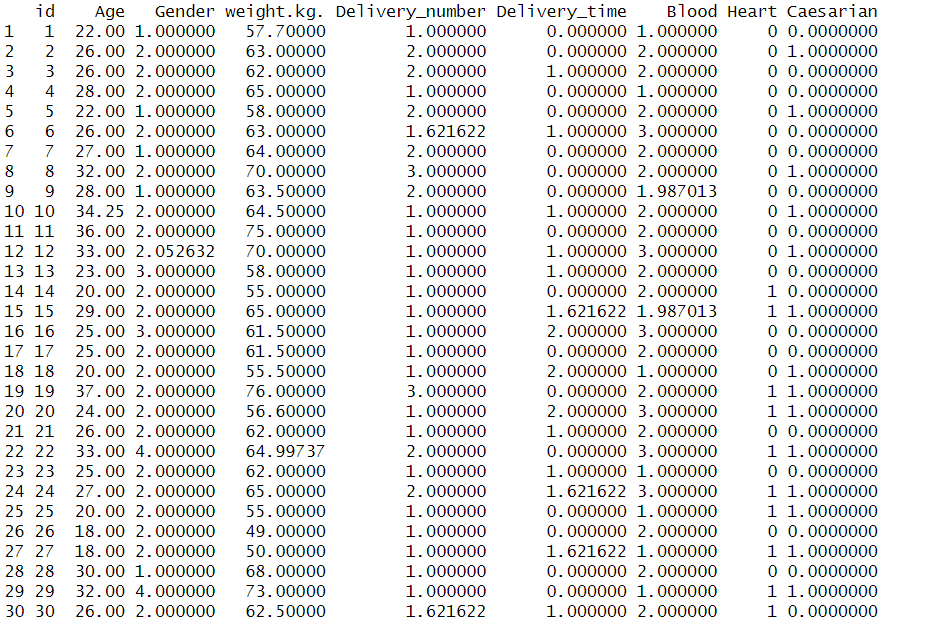


**Recovering Missing Values with Mean**

**Code:**



**Output:**



**Description:**

In this code, missing values in the "Gender," "Age," " **Weight**," " **Delivery Number** “ , “**Delivery Time” and “Blood”**columns of the "meanData" dataset are imputed using their respective column means. The dataset is first converted to numeric values, and then missing values are replaced with the calculated means. This process ensures that the dataset now contains imputed values for these columns, making it suitable for analysis.

**'mydata' Dataset as 'medianData'**

**In medianData set we will replace all the missing values with median value.**

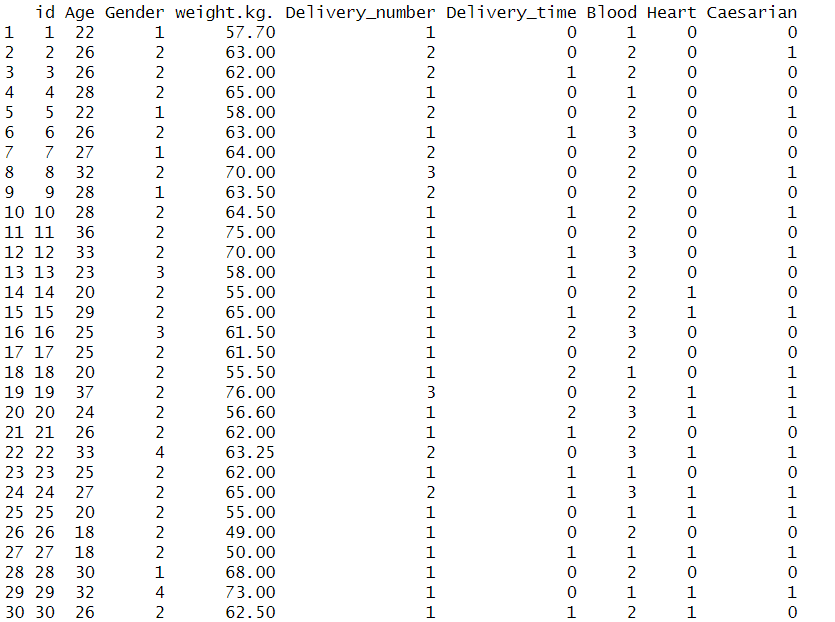


**Recovering Missing Values with Median**

**Code:**



**Output:**



**Description:**

The "medianData" dataset is being processed for imputing missing values in the "Gender" "Age," " **Weight**," " **Delivery Number** “ , “**Delivery Time” and “Blood** “columns using their respective column medians. First, the column is converted to numeric values using `as.numeric()`. Then, the variable is assigned the median value of the specific column, with missing values (NA) omitted during the calculation. This step is part of the process to impute missing values in the column using the median value.

**'mydata' Dataset as 'modeData'**

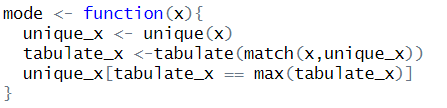
**In modeData set we will replace all the missing values with mode value.**

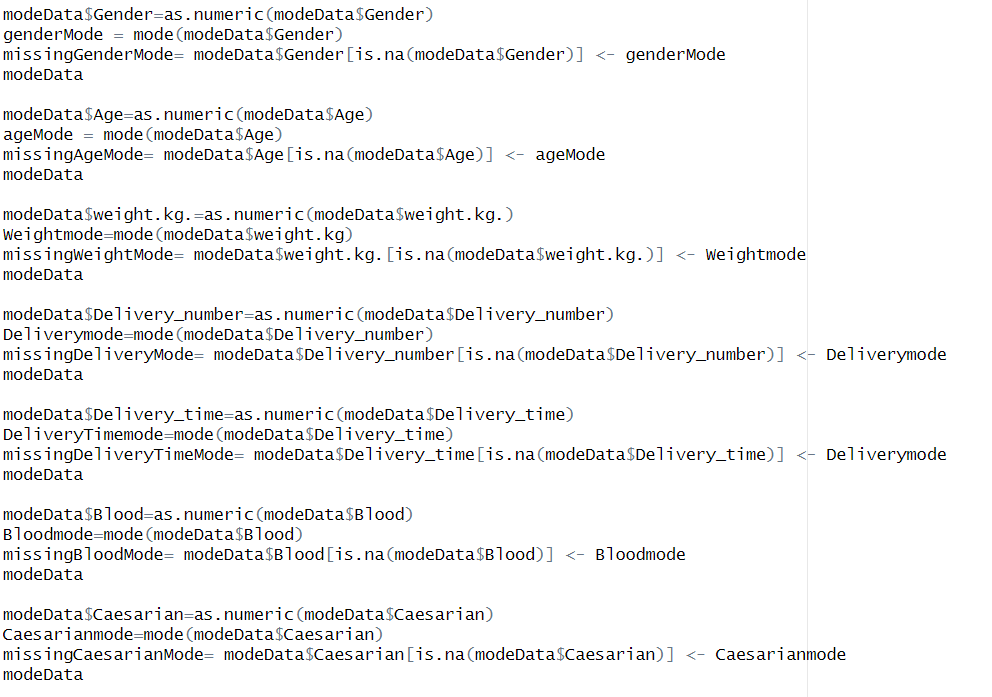
**Code:**



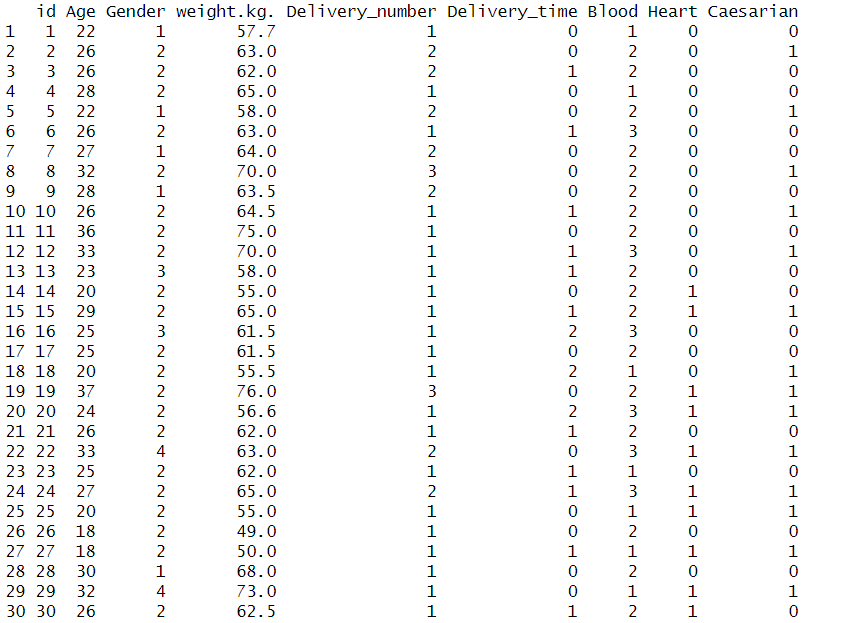
**Recovering missing values with Mode**

**Code:**





**Output:**

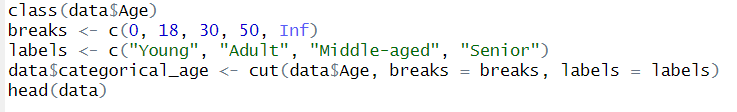


**Description:**

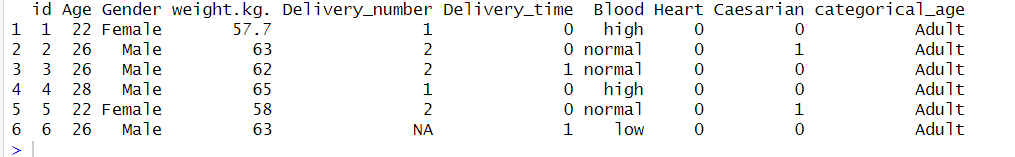
In this code, a custom "mode" function is defined to calculate the mode of a given vector. The code proceeds to apply this function to impute missing values in the "Gender “Age," " Weight," " Delivery Number“ , “Delivery Time” and “Blood “,columns within the "modeData" dataset. Mode represents the most frequently occurring value in each column. The code replaces missing values with the calculated modes, ensuring that the dataset is completer and more suitable for analysis.

**convert continuous or numeric attributes into categorical attributes.**

**Code:**

****

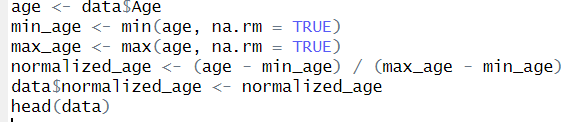
**Output:**

****

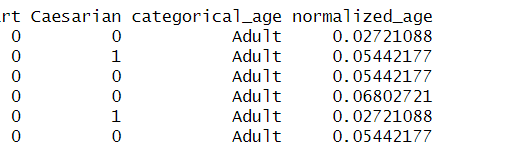
**Description**: This code snippet first checks the data type of the 'Age' column to ensure it is numeric. Then, it defines age intervals ('breaks') and corresponding labels for categorical age groups. Using the **cut()** function, the 'Age' column is categorized into discrete age groups based on the specified intervals and labels. The results are stored in a new column named 'categorical\_age', which is then displayed with the first few rows of the dataset using the **head()** function. This transformation simplifies the analysis by grouping ages into meaningful categories.

**The normalization method only for one attribute.**

**Code:**

****

**Output:**

****

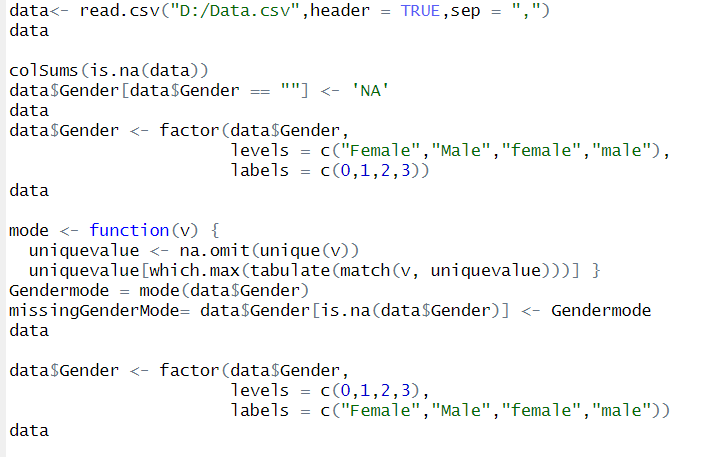
**Description**: In this code segment, the 'Age' attribute is normalized using Min-Max Scaling. Initially, the minimum and maximum values of 'Age' are computed, accounting for any missing values (NA). Subsequently, Min-Max Scaling is applied to standardize the 'Age' values within the range of 0 to 1. The dataset is then augmented by incorporating a new column labeled 'normalized\_age' to accommodate the normalized 'Age' values. Finally, the output exhibits the initial rows of the dataset, including the freshly appended normalized attribute.

This method ensures that the 'Age' attribute is harmonized into a normalized scale, enabling consistent comparisons and analyses.

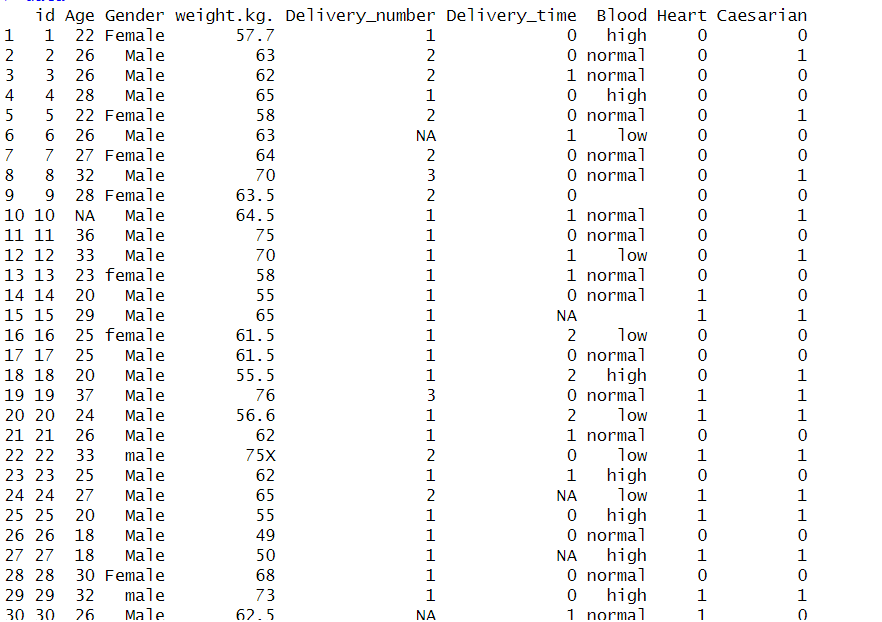
**Replacing categorical missing data by mode:**

(Since we can use mode both for numerical and categorical value at first, we replaced the missing values with numerical value. Now we will replace it with categorical value for Gender and Blood attribute).

**Code**

****

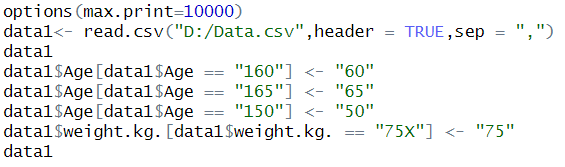
**Output:**

****

**Description:** The missing values in the "Gender" column of the "data" dataset are first identified and replaced with 'NA'. Then, the "Gender" column is converted into a factor variable with numeric values (1 for Male, 2 for Female,3 for male, 4 for female). The mode function is defined to calculate the mode for the "Gender" column. The missing values in the "Gender" column are imputed in the calculated mode, ensuring that the column is complete. Finally, the "Gender" column is converted back to a factor with appropriate labels (Male ,male, female and Female). This code prepares the "Gender" column for analysis by handling missing values and converting it to a factor variable with meaningful labels.

**Correcting Invalid Values**

**Code:**

****

**Output:**

****

****

****

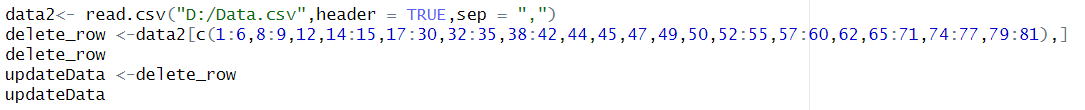
****

****

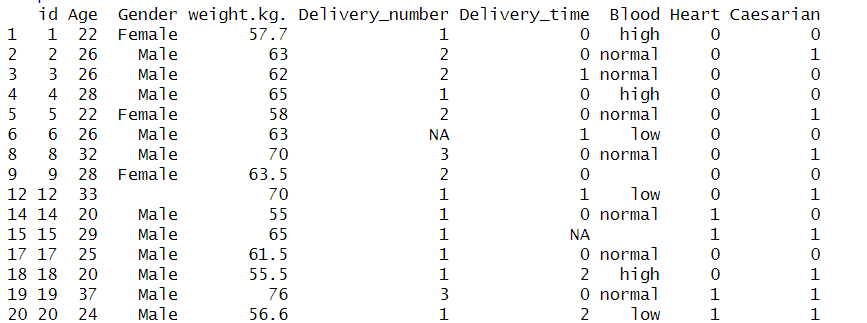
**Description:** The `max.print` option is set to display up to 10,000 rows. The "data1" dataset is loaded from "Data.csv." Subsequently, data cleaning is performed by replacing incorrect values in the "Age" and "Weight.kg" columns to ensure data consistency and accuracy. The first line replaces the value "160" in the "Age" column with "60,”. The second line replaces "165" in the "Age" column with "65," and the third line replaces "150" in the "Age" column with "50," ensuring data consistency. And the 4th line replaces "75X" in the "Weight.kg" column with "75," likely rectifying a data entry error.

**Deleting Invalid Values**

**Code:**



**Output:**



**Description:** This code performs data manipulation by removing specific rows from the dataset and creating a new variable ("updateData") with the modified data.

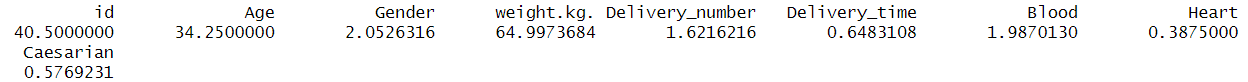
**Measure of Central Tendency**

**Calculate the mean for each attribute:**

**Code**



**Output:**



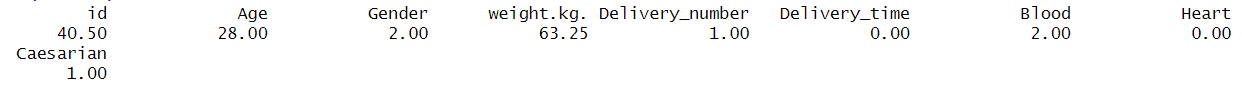
**Description:** The "means" variable contains the column meaning of the "meanData" dataset. It provides the average values for each column, making it useful for summarizing the central tendencies of the dataset.

**Calculate the median for each attribute using apply**

**Code:**



**Output:**



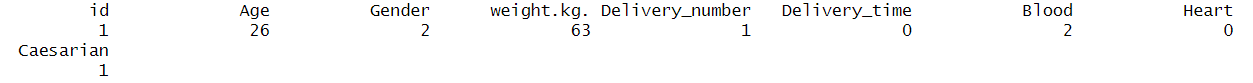
**Description:** The "medians" variable holds the median values calculated for each column of the "medianData" dataset. The "medians" variable provides the median values for each column in the "medianData" dataset. For instance, the median "age" is 28, and the median “weight.kg" is 63.25. This summary helps understand the central tendencies of the data, with medians representing the middle values in each column.

**Calculate the mode for each attribute using apply**

**Code:**



**Output:**



**Description:** In the "modes" variable, the most frequently occurring values for each column of the "modeData" dataset have been calculated. The result shows the modes for the columns, such as the mode "Gender" being "2" (likely representing a category), and "Age" being "26." This provides insight into the most common values within each column, which can be useful for understanding the data's central tendencies and frequent occurrences.

**Measure of Spread**

**'medianData' Dataset as 'newData'**



**Calculating the range**

**Code:**



**Output:**



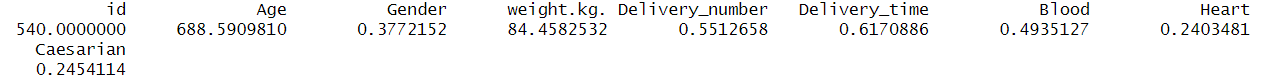
**Description:** In this code, a new dataset "numeric\_data" is created by selecting only the numeric columns from the "newData" dataset. The "Range" values for various numeric columns are calculated using the `range()` function. The output shows the minimum and maximum values for each column, providing insights into the data's range or spread: For instance, in the "Age" column, the range is from 18 to 165. In the "Weight.kg" column, the range is from 49 to 110. The "blood" column has a range from 1 to 3. This information is useful for understanding the variability and bounds of the data in each column.

**Calculating the variance**

**Code:**



**Output:**



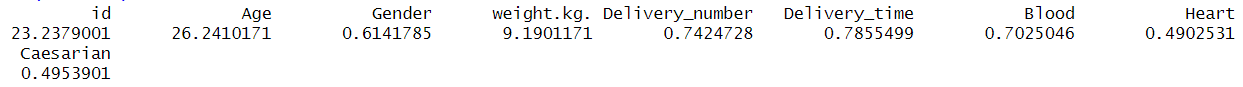
**Description:** In this code, the variance values for various numeric columns are calculated using the `var()` function and displayed. Variance quantifies the degree of data spread or variability in each column. For instance, the variance for "Age" is approximately 688.5909810.

**Calculating the standard deviation**

**Code:**



**Output:**



**Description:** In this line of code, the standard deviations (std\_devs) are calculated for each numeric column within the "numeric\_data" dataset. This step helps quantify the dispersion or variability of data in each column. These standard deviation values represent the degree of variability in each respective numeric column. Larger standard deviations typically indicate greater data variability, while smaller values suggest less variability or more consistent data.

**Histogram**

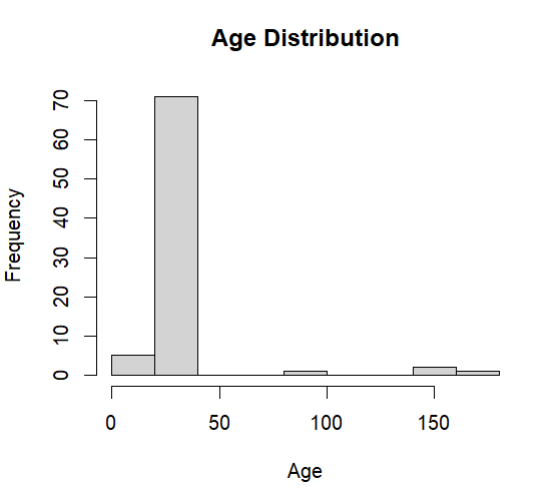
Histogram is suitable for continuous or numerical data. It displays the distribution and frequency of data values within specific bins or intervals. Here we used a dataset(modeData) where the missing values replaced with mode values.

**AGE:**

**Code:**

****

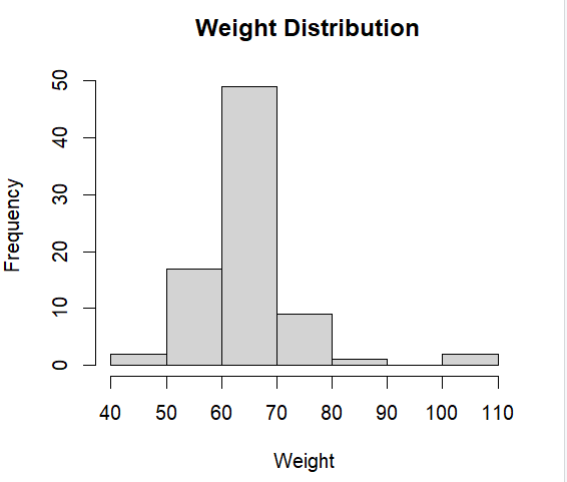
**Output:**

****

**Description:** In this code, a histogram is created to visualize the distribution of the "Age" attribute in the "modeData" dataset. The title of the histogram is set as "Age Distribution," with "Age" displayed on the x-axis and "Frequency" on the y-axis. This histogram provides a visual representation of the frequency of different age values in the dataset. From visualization we can see the age values in dataset are between 0 to 100.Aroud 70 age values are between 0 to 50 and rest of them are between 50 to 100. There are two/three values that are between 150 to 200.

**weight.kg.**

****

****

**Description:** Here we visualized Weight.kg distribution data. We used ylim to set the frequency range. From data distribution we can see most of the data values are between in 60 to 70 range. The range between 50 to 60 is the second highest and there is a negative value in our data

**Delivery number**

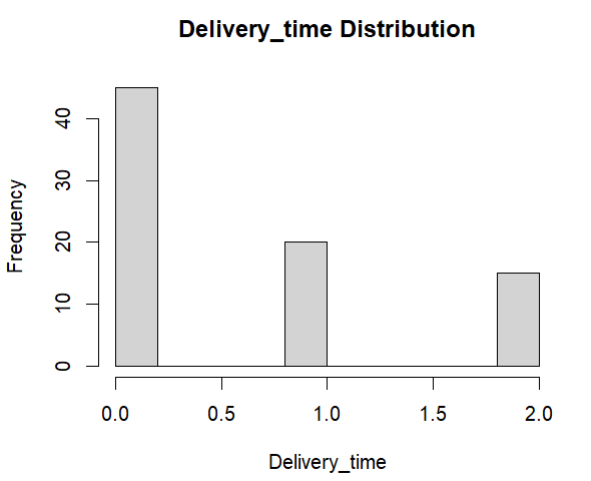
****

**Description :**The 'Delivery\_time' column of the dataset 'modeData'. Titled "Delivery\_time Distribution", the plot's x-axis is labeled as 'Delivery\_time', representing time values, while the y-axis denotes 'Frequency', indicating the count of occurrences. This visualization aids in understanding the frequency distribution of delivery times within the dataset.

**Delivery\_time**

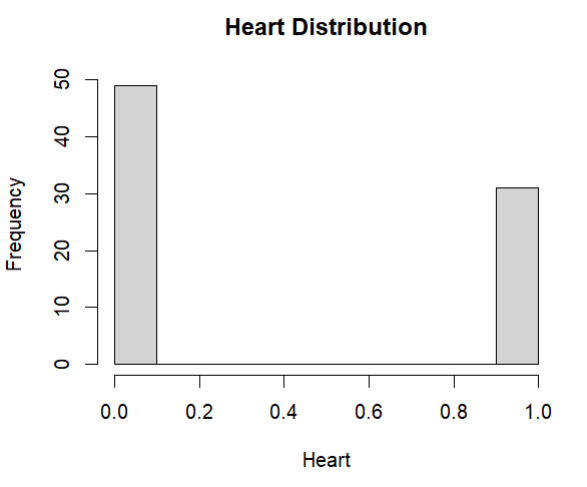
**Code:**

****

****

**Heart**

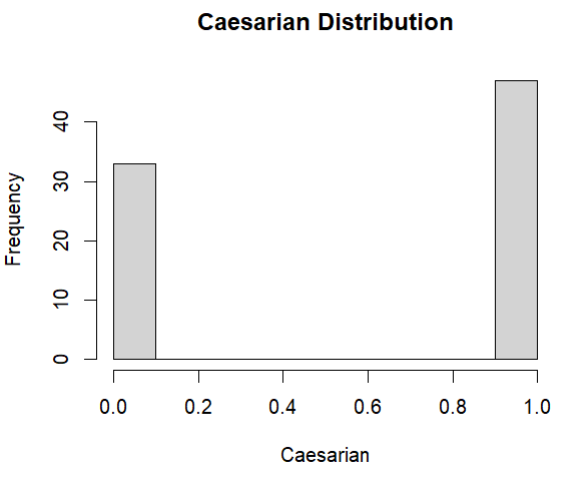
****

****

**Description :**The 'Heart' column of the dataset 'modeData'. The title of the histogram is "Heart Distribution", with the x-axis labeled as 'Heart' and the y-axis labeled as 'Frequency', denoting the count of occurrences. This visualization facilitates the analysis of the frequency distribution of heart-related data within the dataset.

**Caesarian**

****

****

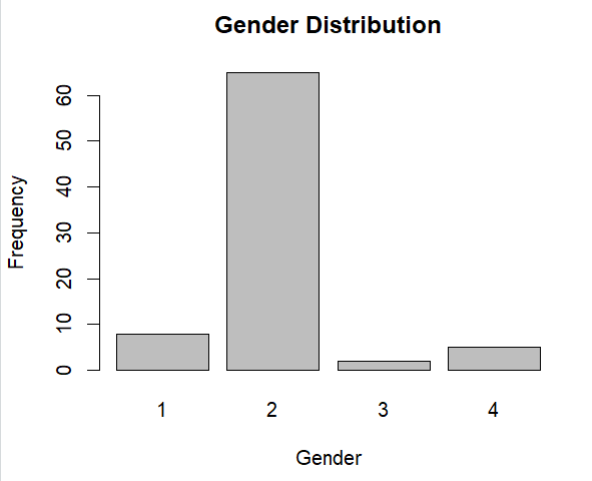
**Bar Graph**: Bar graph is suitable for categorical or nominal data. It represents the frequency or count different categories or groups. Mostly used for showing relationships between different categories. Here we used medianData set to replace all the null values with median value.

**GENDER:**

**Code:**

****

**Output:**

****

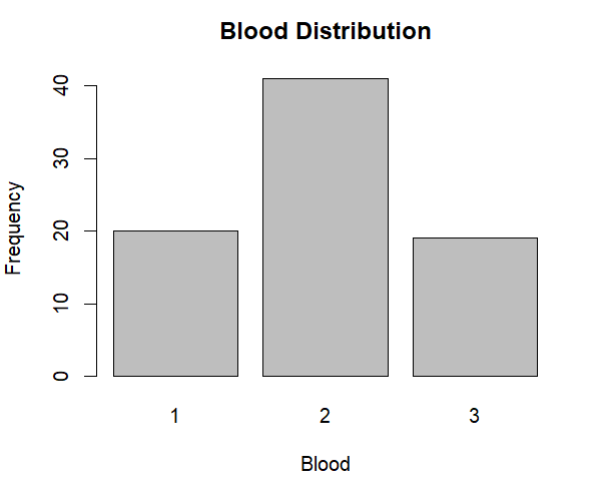
Description: barplot(….) function is used to create a bar plot. Table (medianData$Gender) creates a table that counts the frequency of each unique value in the mentioned variable. We unnoted female by 1 , male by 2,Female by 3 and Male by 4.

**Blood:**

**Code:**

****

**Output**

****

**Visualizing outliers via boxplot**

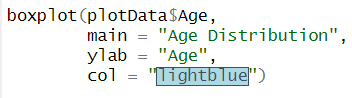
Boxplot is used for detecting outliers for numerical values in dataset. Here at first we applied boxplot for each data attribute and then removed outliers later where necessary.

**' medianData ' Dataset as ' plotData '(we are replacing missing values with median data and stored it into plotdata**

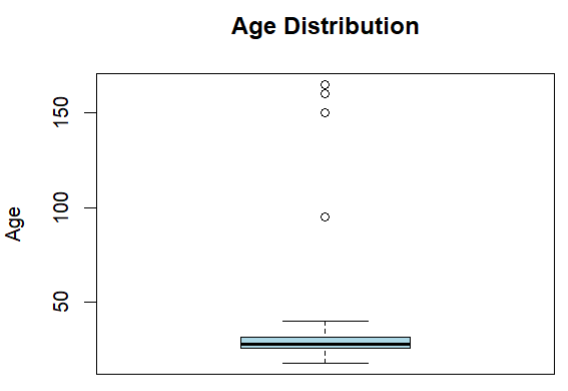
****

**Age:**

**Code:**

****

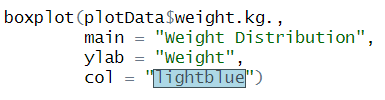
**Output:**

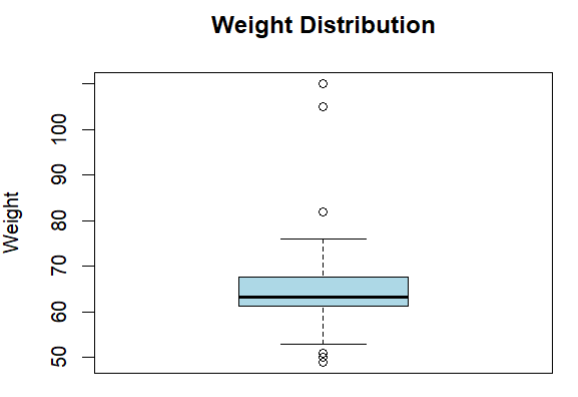
****

**Description:** plotData is used to create a box plot. plotData$age specifies the numeric variable age in the dataset. Main sets the title. Ylab sets the y-axis label.col=lightblue sets the color of the boxplot to light blue. We can see outliers here. There are two values which are out of the maximum range of boxplo

**Weight:**

**Code:**

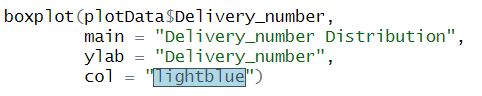




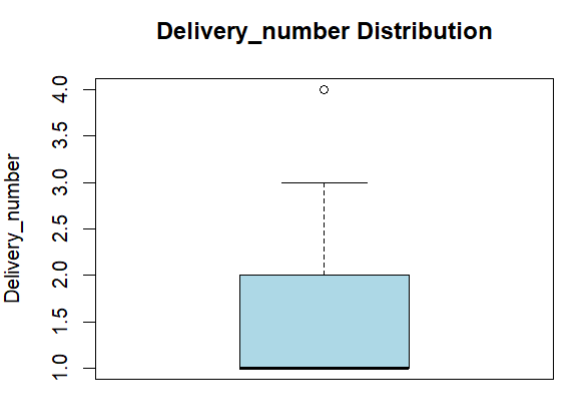
**Description:** In Weight.kg distribution there are lot of outliers. Most of the outliers are out of max range of boxplot. However there are also some values which are out of min range of box plot

**Delivery\_number:**

**Code**



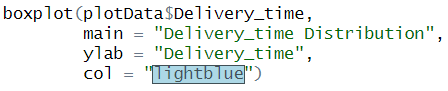
**Output:**



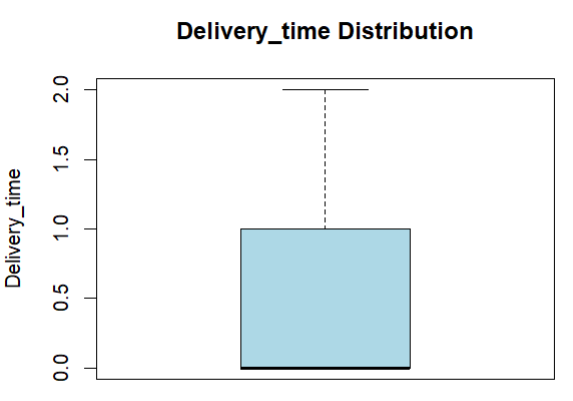
**Description:** In Delivery\_number distribution there are lot of outliers. Most of the outliers are out of max range of boxplot. On the other hand, there is only one value which is out of min range of box plot.

**Delivery\_time**

**Code:**



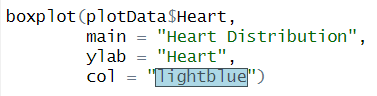
**Output:**



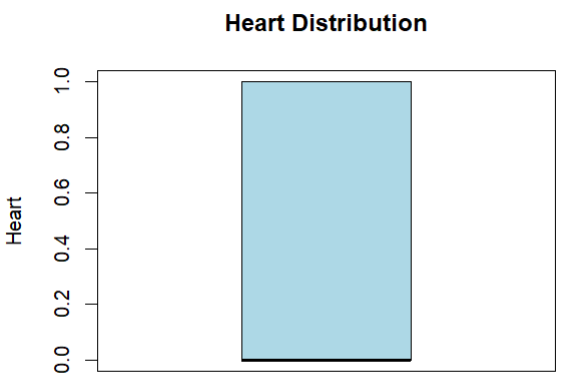
**Description:** the distribution of delivery times contained in the 'Delivery\_time' column of the dataset 'plotData'. The main title of the boxplot is "Delivery\_time Distribution", while the y-axis is labeled as 'Delivery\_time'. The boxplot is shaded in light blue color for better visualization. This visualization aids in comparing the distribution of delivery times and identifying any potential outliers within the dataset.

**Heart**

**Code:**



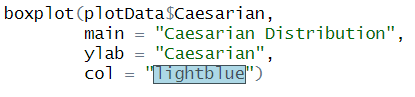
**Output:**



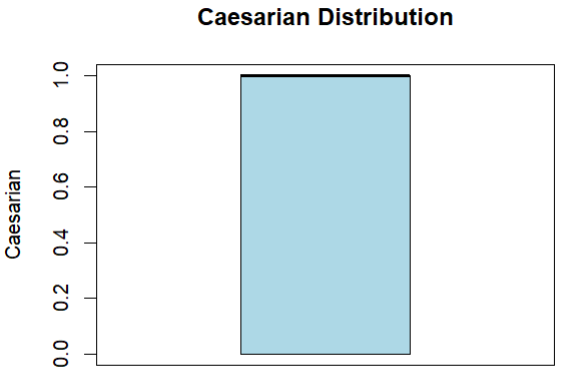
**Description:** "Heart Distribution", the plot's y-axis is labeled as 'Heart'. The boxplot is shaded in light blue for improved clarity. This visualization assists in analyzing the distribution and variability of heart-related data within the dataset.

**Caesarian**

**Code:**



**Output:**

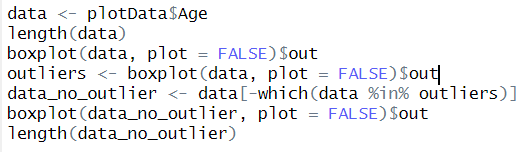


**Description:** The main title of the boxplot is "Caesarian Distribution", while the y-axis is labeled as 'Caesarian'. The boxplot is shaded in light blue color, enhancing the visualization. This visualization helps analyze the distribution of Caesarian data, including any potential outliers.

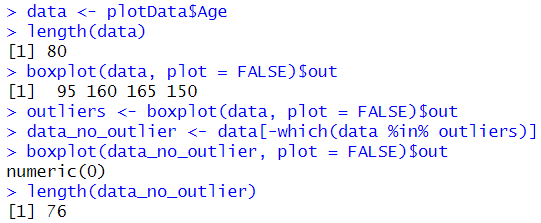
**Removing outliers**

**AGE:**

**Code:**

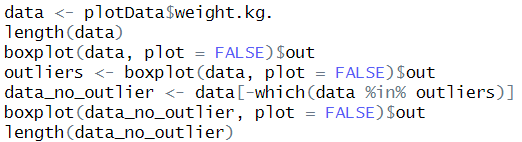


**Output:**

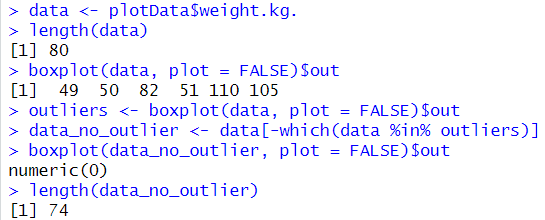


**Description:** The dataset "data" contains 80 values. By using a boxplot, in age column two outliers with values 95,160,165 and 150 are identified and stored in the "outliers" variable. These outliers are then removed from the "data" dataset to create a new dataset called "data\_no\_outlier." A second boxplot for "data\_no\_outlier" confirms the absence of outliers, and the length of the cleaned dataset is determined to indicating that the outliers have been successfully removed. This code showcases the process of outlier detection and data cleaning.

**Weight:**

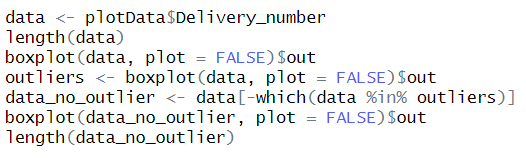


**Output:**

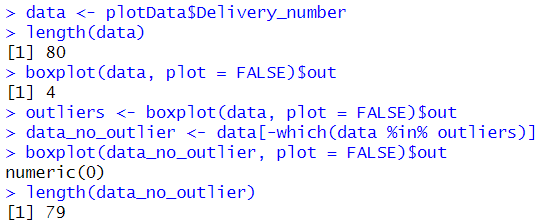


**Description:**A dataset containing weight data. Initially, the length of the dataset is calculated and printed. Then, a boxplot is generated to identify outliers, which are subsequently stored. The outliers are removed from the original dataset, resulting in a cleaned dataset.

**Delivery\_number:**



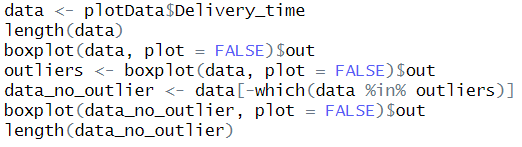
**Output:**



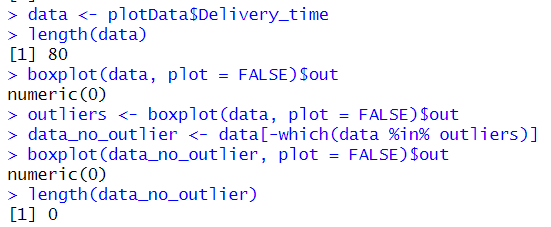
**Description:**Calculates the length of the dataset and then generates a boxplot to identify any outliers. The detected outlier(s) are stored, and subsequently removed from the original dataset. This process ensures the dataset is cleansed of outliers, providing a more accurate representation of the delivery number data.

**Delivery\_time:**

**Code:**



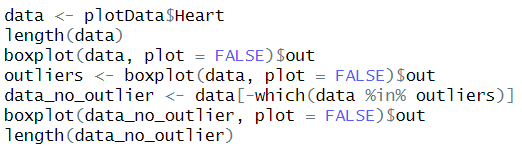
**Output:**



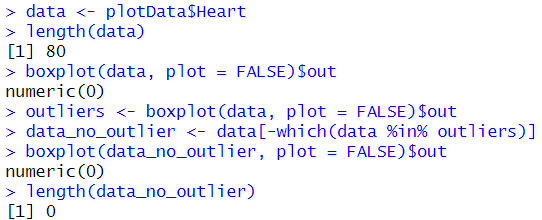
**Description:**The length of the 'Delivery\_time' dataset is calculated and printed, showing a total of 80 observations.A boxplot is generated for the 'Delivery\_time' data without displaying it, and any detected outliers are capturedThe original dataset is copied into 'data\_no\_outlier', as no outliers needed to be removed.Another boxplot is created for the cleaned dataset, again showing no outliers.

**Heart:**

**Code:**



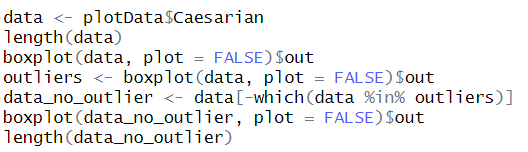
**Output:**



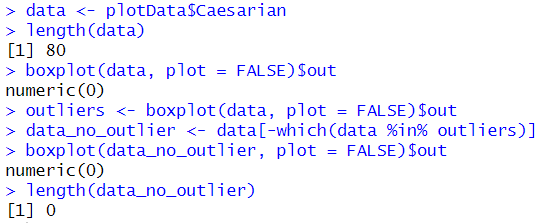
**Description:** Assigns the 'Heart' data from 'plotData' to the variable 'data’. Determines and presents the length of the 'Heart' dataset. Generates a boxplot of the 'Heart' data identifying and returning any outliers.

**Caesarian**

**Code:**



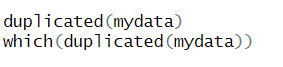
**Output:**



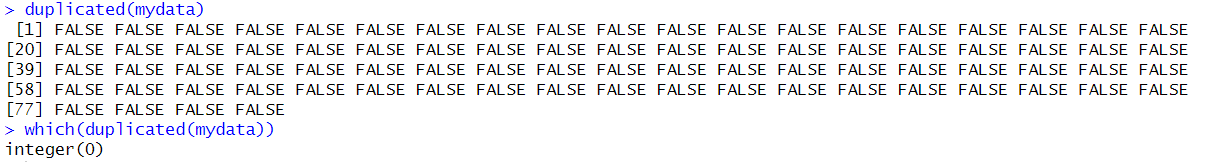
**Description:** Assigns the 'Caesarian' data from 'plotData' to the variable 'data ‘Computes the length of the 'Caesarian' dataset. Generates a boxplot of the 'Caesarian' data it, identifying and returning any outliers.

**Handling Duplicate Values**

**Code:**



**Output:**



**Description:** The duplicated() function is used to identify duplicate row in a dataset. The function returns a logical vector indicating whether each element is a duplicate of a previous element. There is no the duplicated row in the dataset.