**American International University-Bangladesh**

A picture containing text

Description automatically generated

**INTRODUCTION TO DATA SCIENCE**

*Mid Project*

**Submitted by:**

**Name:** Sinthia Maliha Mumu  
**Id:** 20-42242-1  
**Section:** D

**Submitted to:**

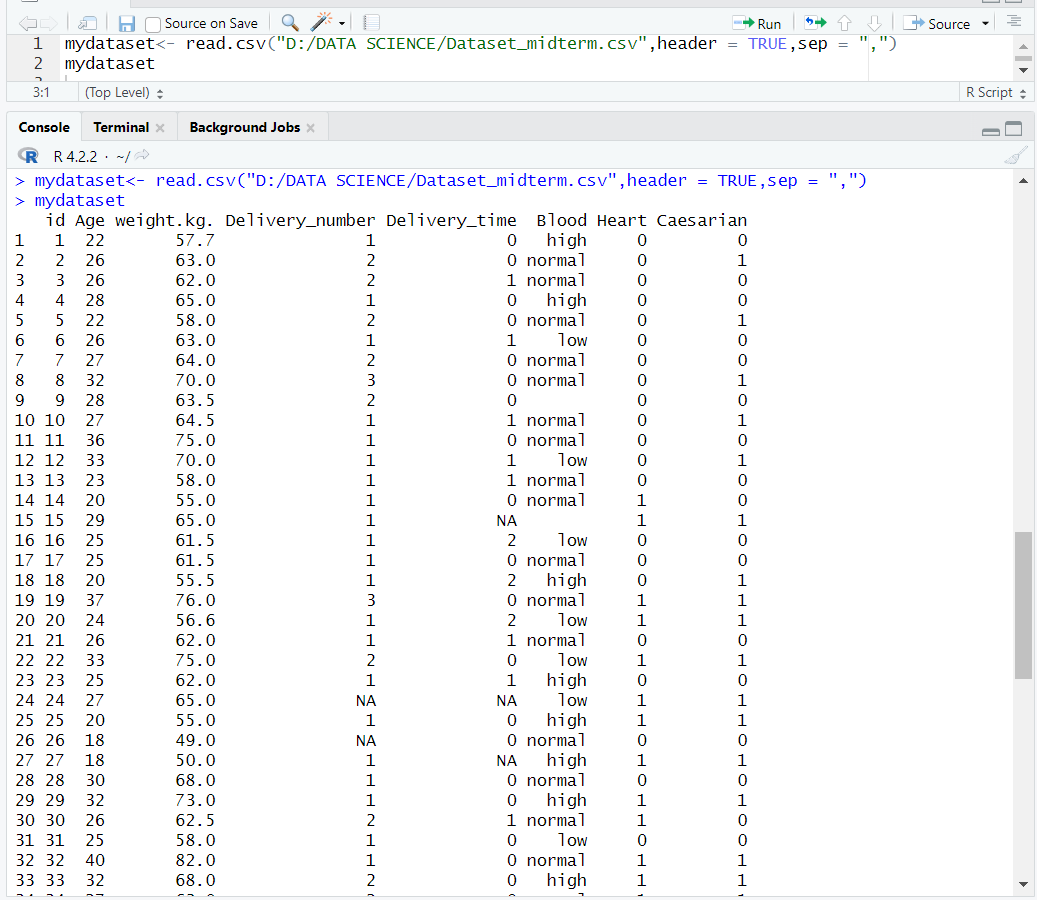
**TOHEDUL ISLAM**

**Project Overview:**

The aim of this project is to carry out data preparation and univariate exploration, which involves preparing raw data for analysis by cleaning it up and transforming it into usable information. Missing data is a typical problem in databases that can result from a column needing a value or data not being captured during collection. The data set examined in this project contains information concerning the results of caesarean sections for 80 pregnant women, along with the most significant features of delivery complications in the medical sector.

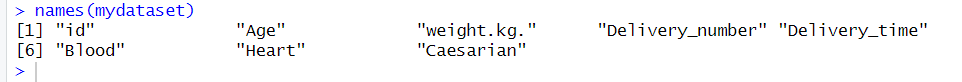
1. **To import the dataset and print**

mydataset<- read.csv("D:/DATA SCIENCE/Dataset\_midterm.csv",header = TRUE,sep = ",")

mydataset

1. **To see the column name of data set**

names(mydataset)



1. **To find the shape of the dataset**

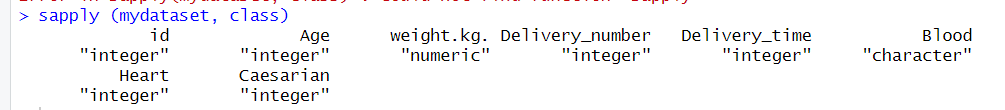
str(mydataset)

Text

Description automatically generated

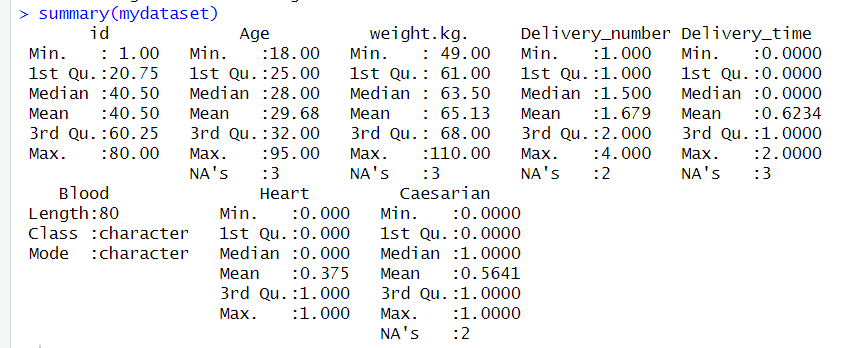
1. **To find the types of data for all attributes**

sapply (mydataset, class)



1. **Descriptive Statistics Using summary () Function**

summary(mydataset)



1. **Using dplyr deleting duplicate data**

library(dplyr)

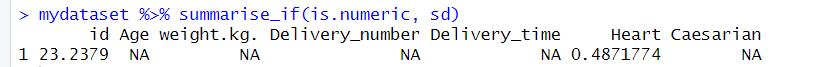
mydataset\_cleaned <- distinct(mydataset)

mydataset



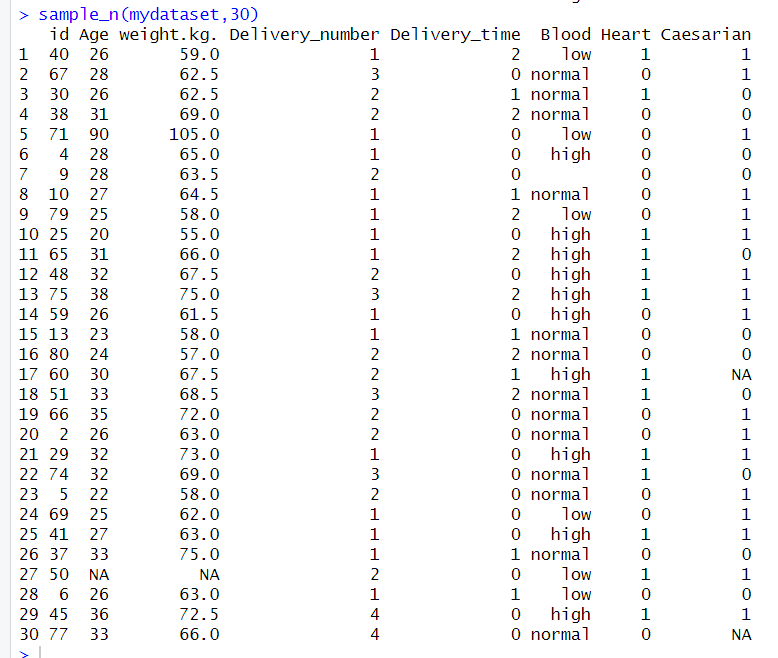
**Standard deviation of multiple columns in R using dplyr**

mydataset %>% summarise\_if(is.numeric, sd)



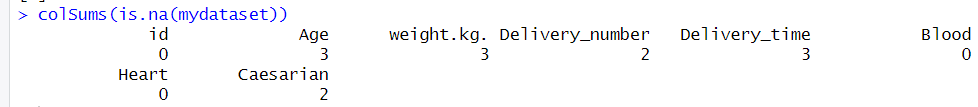
**Sampling Dataset**

sample\_n(mydataset,30)



1. **To find the missing values of all attributes**

colSums(is.na(mydataset))

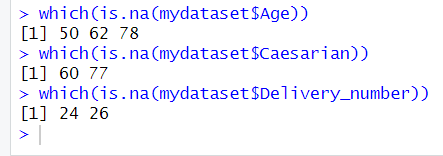


1. **Finding the specific row number of Null values.**

which(is.na(mydataset$Age))

which(is.na(mydataset$Caesarian))

which(is.na(mydataset$Delivery\_number))



1. **Annotate low as 0, high as 1 and normal as 2 from “Blood” attribute**

mydataset$Blood <- factor(mydataset$Blood, levels = c("low","high","normal"), labels = c(0,1,2))

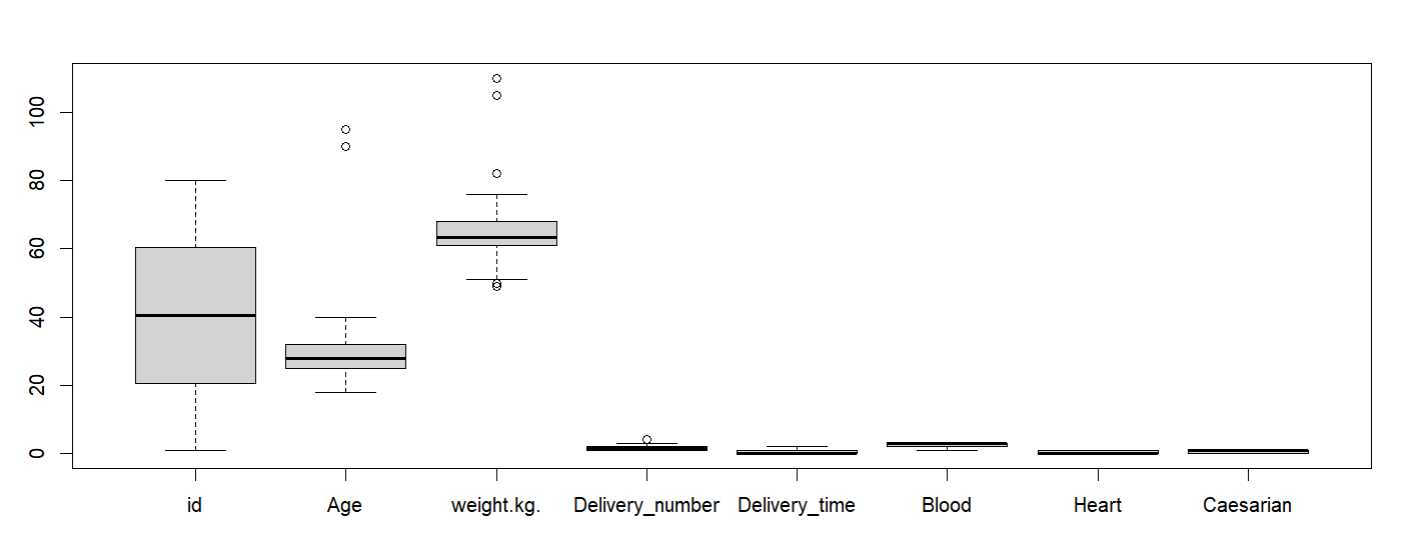
mydataset

**Table

Description automatically generated**

1. **Detecting outliers for finding noisy values.**

boxplot(mydataset)



1. **Converting outliers as missing values**

mydataset$Age[mydataset$Age > 60] = NA

mydataset$weight.kg.[mydataset$weight.kg. > 80] = NA

mydataset$Delivery\_number[mydataset$Delivery\_number > 3] = NA

Chart, box and whisker chart

Description automatically generated

mydataset

Table

Description automatically generated

**Missing Values**

1. **Deleting the rows with missing value**

mydatasetomit <- na.omit(mydataset)

mydatasetomit

Table

Description automatically generated

1. **Recovering missing values with mean value.**

mydatasetMean<- mydataset

mydatasetMean$weight.kg.[is.na(mydatasetMean$weight.kg.)]<-mean(mydatasetMean$weight.kg.,na.rm=TRUE)

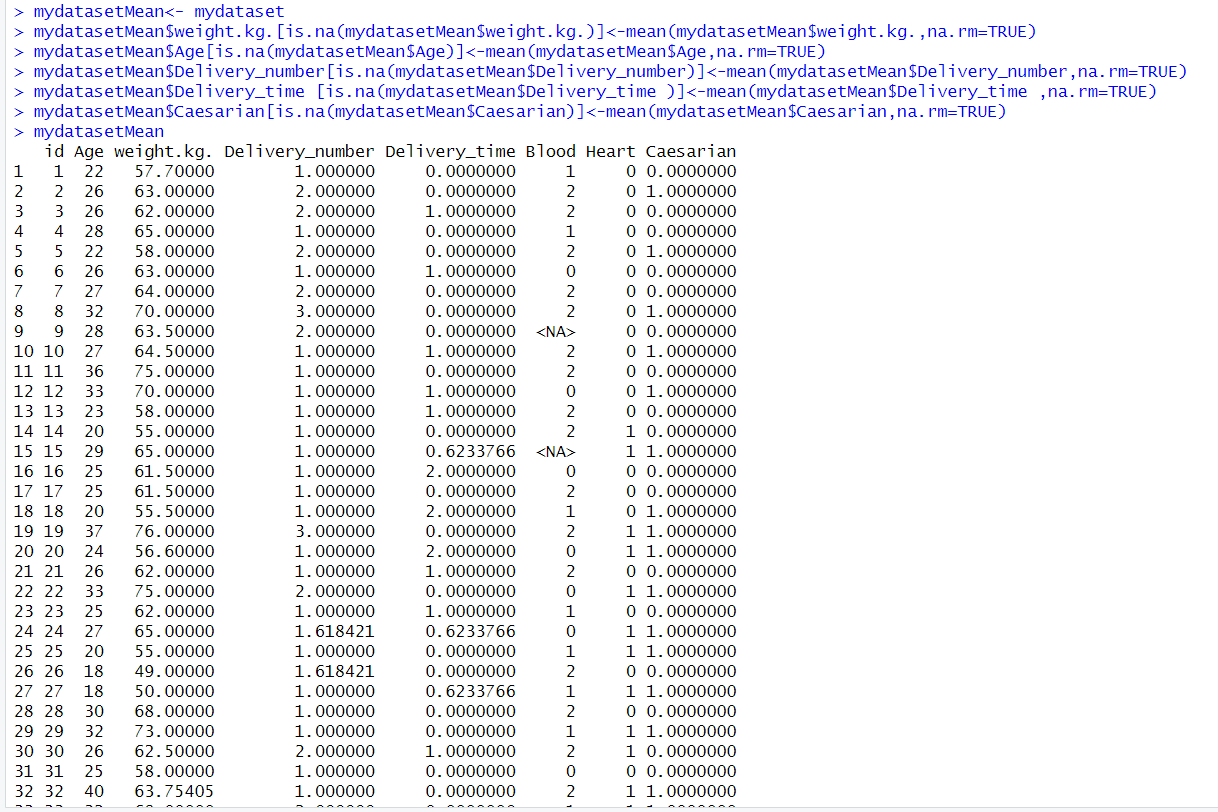
mydatasetMean$Age[is.na(mydatasetMean$Age)]<-mean(mydatasetMean$Age,na.rm=TRUE)

mydatasetMean$Delivery\_number[is.na(mydatasetMean$Delivery\_number)]<-mean(mydatasetMean$Delivery\_number,na.rm=TRUE)

mydatasetMean$Delivery\_time [is.na(mydatasetMean$Delivery\_time )]<-mean(mydatasetMean$Delivery\_time ,na.rm=TRUE)

mydatasetMean$Caesarian[is.na(mydatasetMean$Caesarian)]<-mean(mydatasetMean$Caesarian,na.rm=TRUE)

mydatasetMean



1. **Recovering missing values with median value.**

mydatasetMedian<- mydataset

mydatasetMedian$weight.kg.[is.na(mydatasetMedian$weight.kg.)]<-median(mydatasetMedian$weight.kg.,na.rm=TRUE)

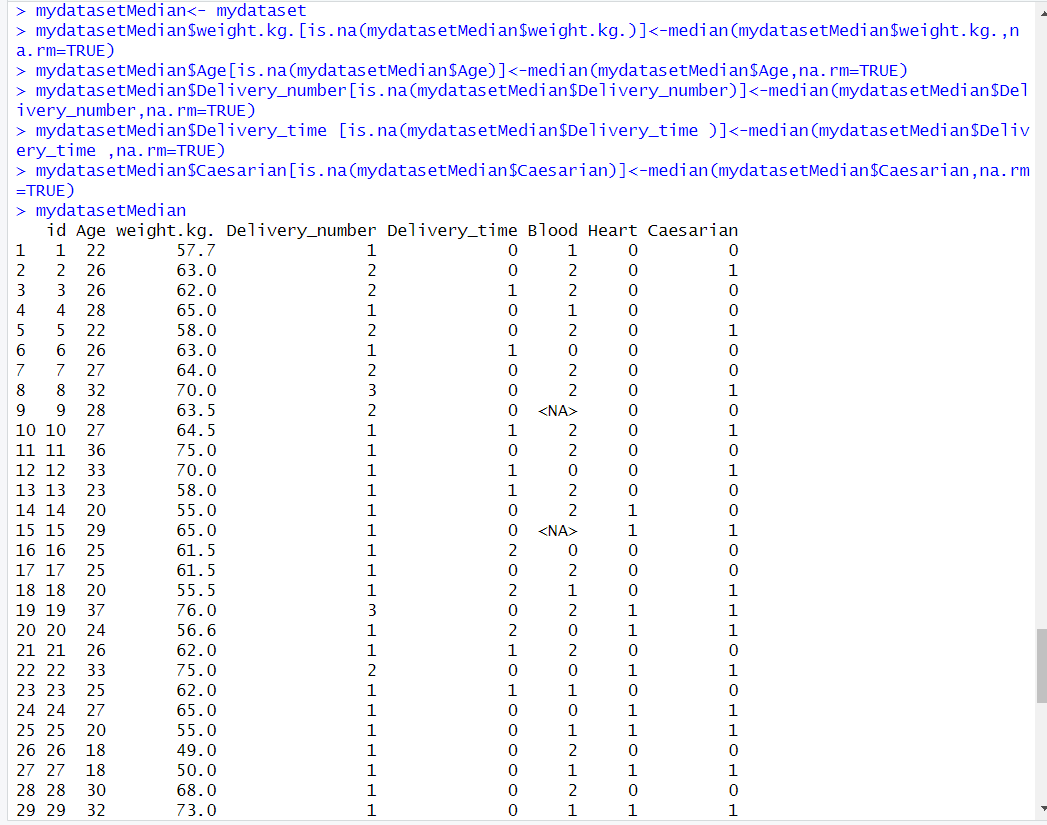
mydatasetMedian$Age[is.na(mydatasetMedian$Age)]<-median(mydatasetMedian$Age,na.rm=TRUE)

mydatasetMedian$Delivery\_number[is.na(mydatasetMedian$Delivery\_number)]<-median(mydatasetMedian$Delivery\_number,na.rm=TRUE)

mydatasetMedian$Delivery\_time [is.na(mydatasetMedian$Delivery\_time )]<-median(mydatasetMedian$Delivery\_time ,na.rm=TRUE)

mydatasetMedian$Caesarian[is.na(mydatasetMedian$Caesarian)]<-median(mydatasetMedian$Caesarian,na.rm=TRUE)

mydatasetMedian



1. **Recovering missing values with mode value.**

Age <- mydataset$Age

getAgeMode <- function(Age){

uniqv <- unique(Age)

uniqv[which.max(tabulate(match(Age, uniqv)))]

}

Age.mode <- getAgeMode(Age)

mydataset$Age[is.na(mydataset$Age)]<-Age.mode

mydataset

weight <- mydataset$weight.kg.

getweight.kg.Mode <- function(weight){

uniqv <- unique(weight)

uniqv[which.max(tabulate(match(weight, uniqv)))]

}

weight.mode <- getweight.kg.Mode(weight)

mydataset$weight.kg.[is.na(mydataset$weight.kg.)]<-weight.mode

mydataset

Delinum<- mydataset$Delivery\_number

getDelivery\_numberMode <- function(Delinum){

uniqv <- unique(Delinum)

uniqv[which.max(tabulate(match(Delinum, uniqv)))]

}

Delinum.mode <- getDelivery\_numberMode(Delinum)

mydataset$Delivery\_number[is.na(mydataset$Delivery\_number )]<-Delinum.mode

mydataset

Delitime<- mydataset$Delivery\_time

getDelivery\_timeMode <- function(Delitime){

uniqv <- unique(Delitime)

uniqv[which.max(tabulate(match(Delitime, uniqv)))]

}

Delitime.mode <- getDelivery\_numberMode(Delitime)

mydataset$Delivery\_time[is.na(mydataset$Delivery\_time )]<-Delitime.mode

mydataset

Caesar<- mydataset$Caesarian

getCaesarianMode <- function(Caesar){

uniqv <- unique(Caesar)

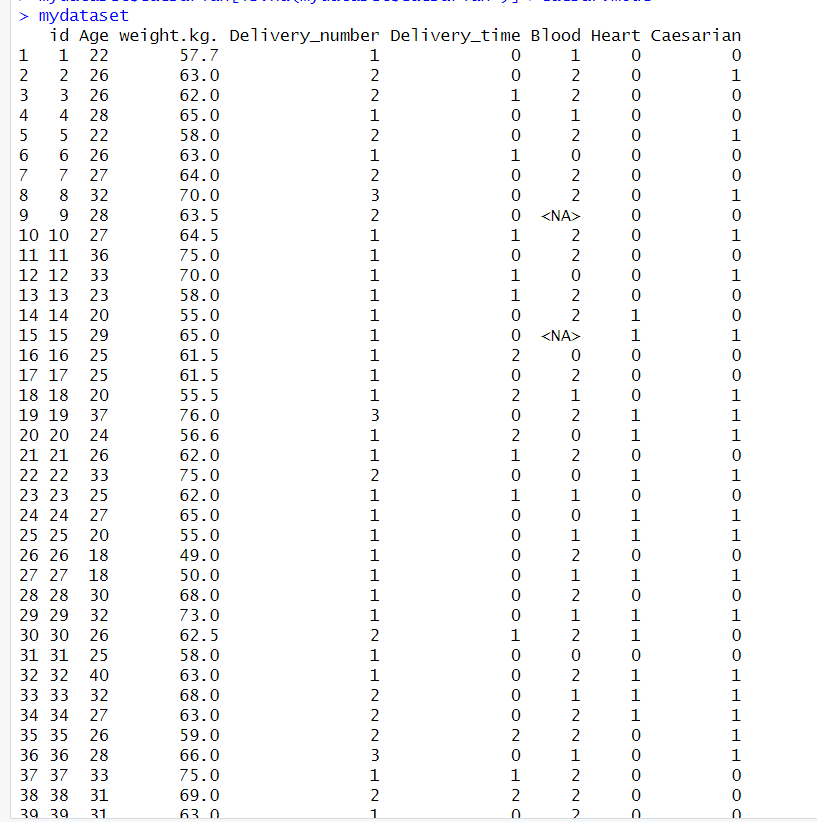
uniqv[which.max(tabulate(match(Caesar, uniqv)))]

**}**

Caesar.mode <- getCaesarianMode(Caesar)

mydataset$Caesarian[is.na(mydataset$Caesarian )]<-Caesar.mode

mydataset

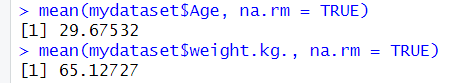


**Univariate Exploration:**

1. **Mean:**

mean(mydataset$Age, na.rm = TRUE)

mean(mydataset$weight.kg., na.rm = TRUE)



1. **Meadian:**

median(mydataset$Age, na.rm = TRUE)

median(mydataset$weight.kg., na.rm = TRUE)

A picture containing graphical user interface

Description automatically generated

1. **Mode:**

val <- as.numeric(mydataset$Age)

getAgeMode <- function(val){

uniqv <- unique(val)

uniqv[which.max(tabulate(match(val, uniqv)))]

}

Age.mode <- getAgeMode(val)

Age.mode

val <- as.numeric(mydataset$weight.kg.)

getweight.kg.Mode <- function(val){

uniqv <- unique(val)

uniqv[which.max(tabulate(match(val, uniqv)))]

}

weight.kg.mode <- getweight.kg.Mode(val)

weight.kg.mode

Text

Description automatically generated