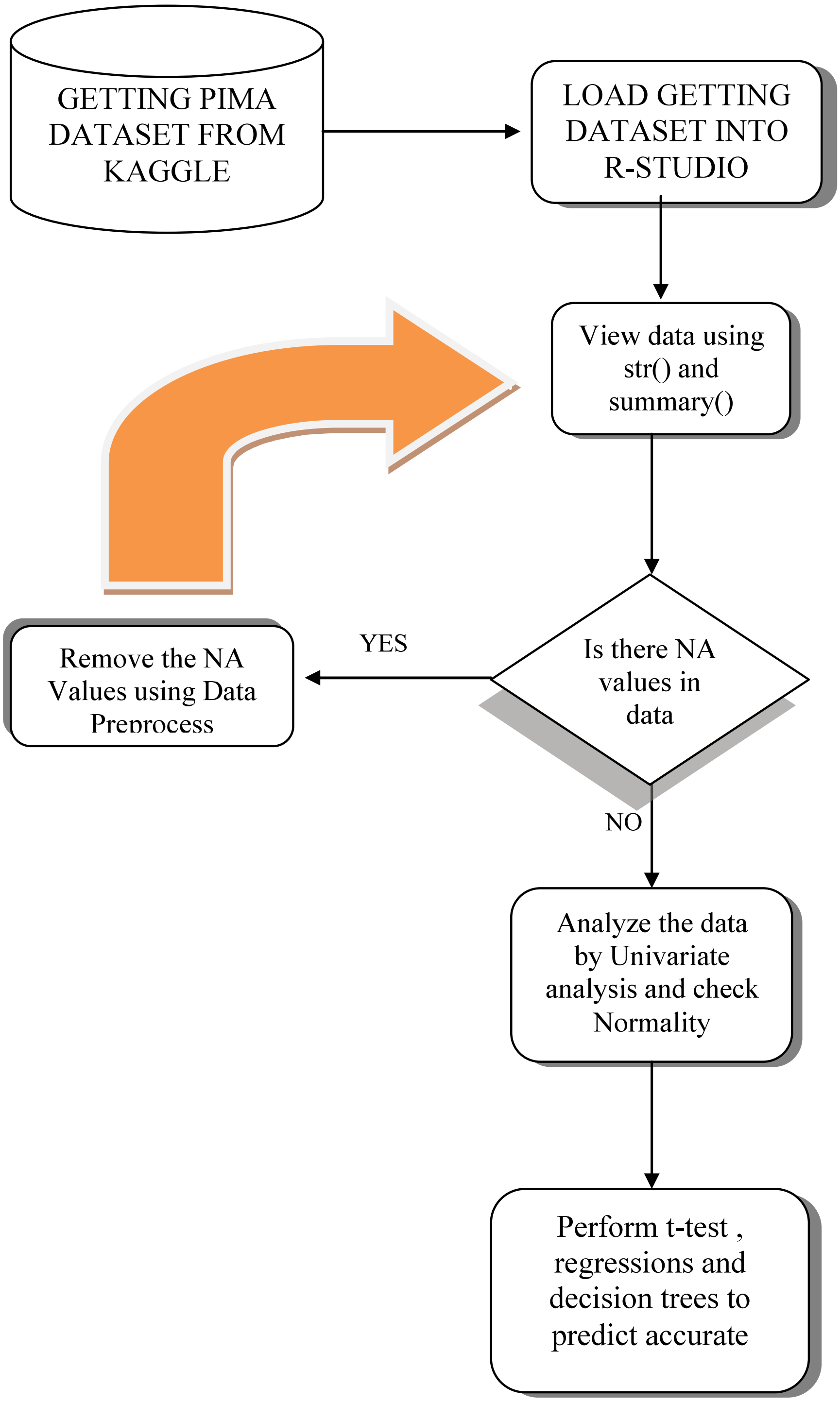
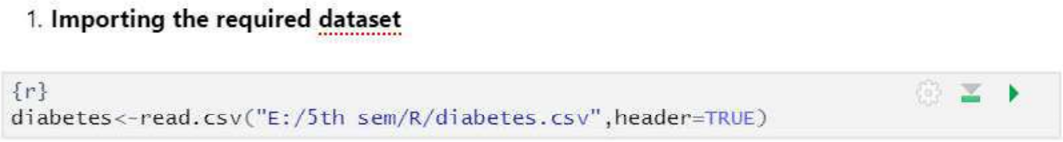
Design and Implementation

❖ Design


GETTING PIMA DATASET FROM KAGGLE
LOAD GETTING DATASET INTO R-STUDIO
View data using str() and summary()
Is there NA values in data
YES
Remove the NA Values using Data Preprocess
NO
Analyze the data by Univariate analysis and check Normality
Perform t-test , regressions and decision trees to predict accurate

❖ Implementation

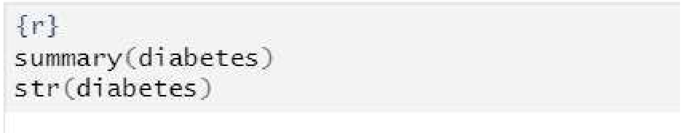
- **The Implementation of this project begins with loading dataset into the R-studio**



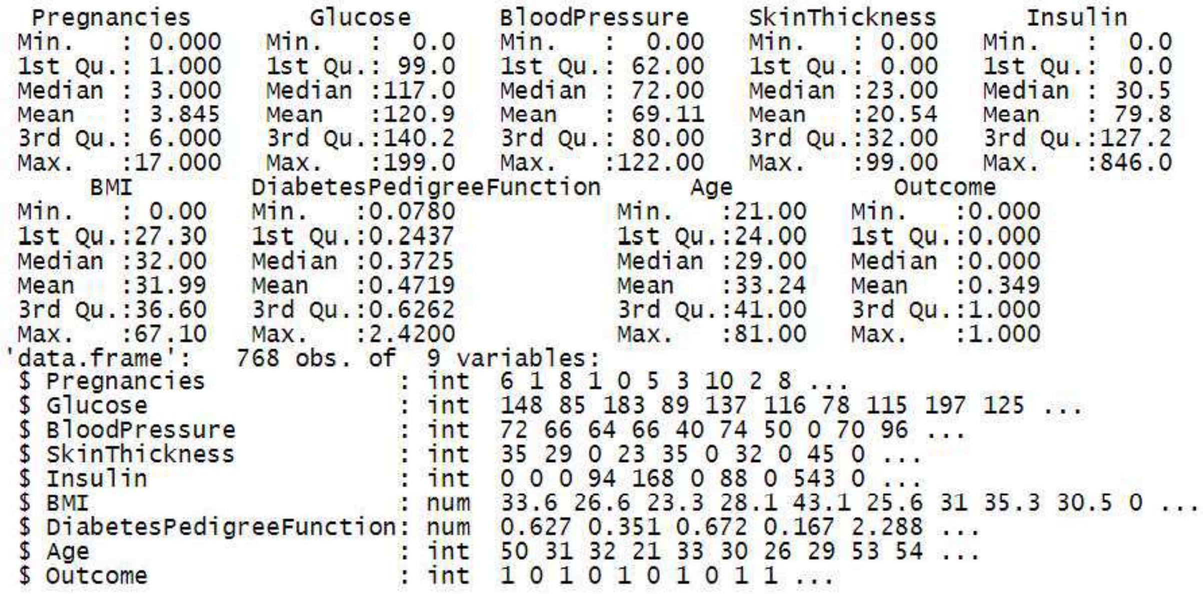
1. Importing the required dataset

```
{r}
diabetes<-read.csv("E:/5th sem/R/diabetes.csv",header=TRUE)
```

- **After the importing dataset into r-studio we going to view data using str() and summary() function**



```
{r}
summary(diabetes)
str(diabetes)
```



```
  Pregnancies        Glucose       BloodPressure     SkinThickness      Insulin    
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0  
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5  
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8  
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0  
      BMI        DiabetesPedigreeFunction      Age           Outcome     
 Min.   : 0.00   Min.   :0.0780           Min.   :21.00   Min.   :0.000  
 1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00   1st Qu.:0.000  
 Median :32.00   Median :0.3725           Median :29.00   Median :0.000  
 Mean   :31.99   Mean   :0.4719           Mean   :33.24   Mean   :0.349  
 3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00   3rd Qu.:1.000  
 Max.   :67.10   Max.   :2.4200           Max.   :81.00   Max.   :1.000  
'data.frame':	768 obs. of  9 variables:
 $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

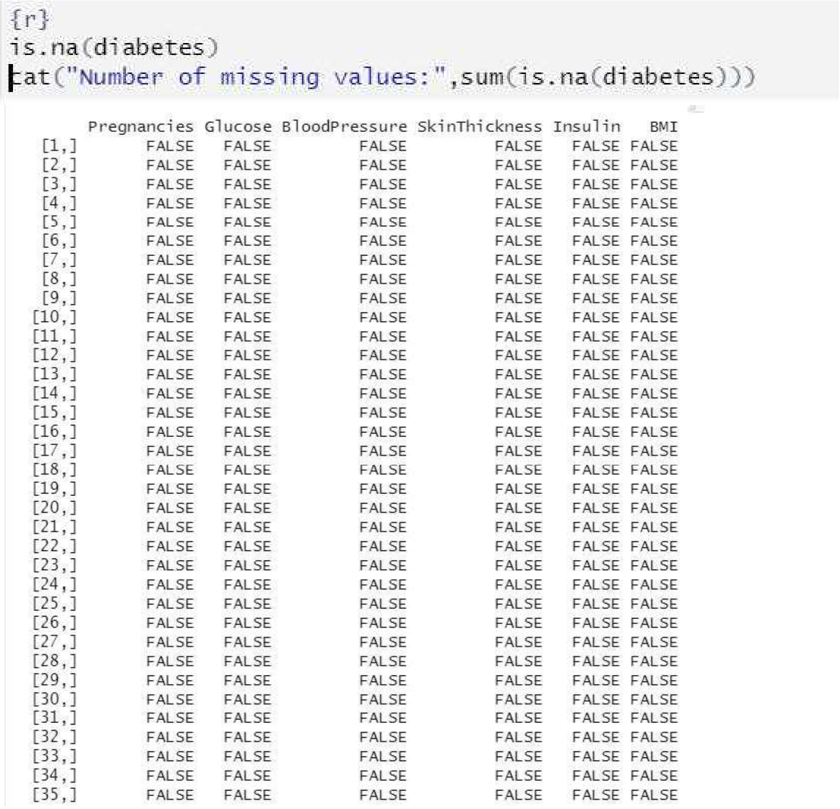
Summary() function provides the descriptive statistics of data includes,

- Mean , Median , Mode
- Minimum and maximum of variables
- Quadrants

Str() function can be used to ,

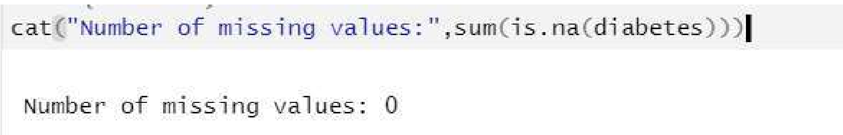
- display the structure of the data frame.
- including the number of rows and columns
- the names of the columns, and the classes of the variables in the data frame.

- **Looking for the missing values in data using is.na()**



```
{r}
is.na(diabetes)
cat("Number of missing values:",sum(is.na(diabetes)))
```

```
      Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
 [1,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [2,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [3,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [4,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [5,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [6,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [7,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [8,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [9,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[10,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[11,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[12,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[13,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[14,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[15,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[16,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[17,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[18,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[19,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[20,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[21,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[22,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[23,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[24,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[25,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[26,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[27,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[28,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[29,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[30,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[31,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[32,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[33,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[34,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[35,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
```

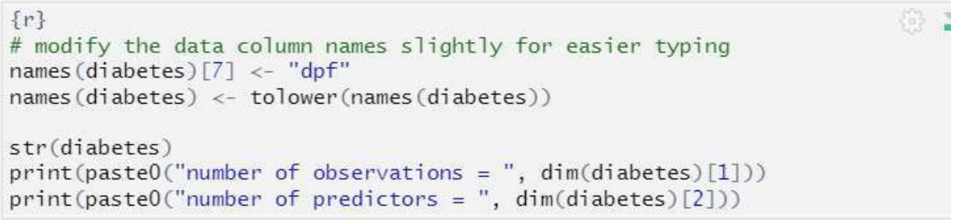


```
cat("Number of missing values:",sum(is.na(diabetes)))

 Number of missing values: 0
```

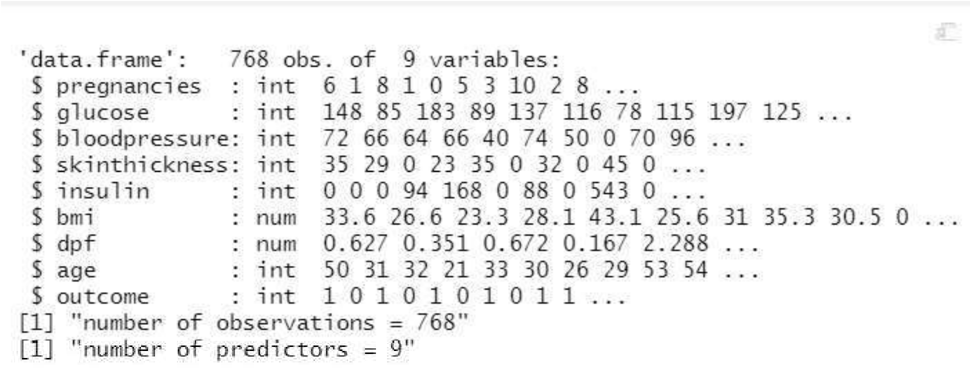
Here is no missing values in data. We can proceed to further steps.

- In our dataset there is a **Diabetes pedigree function** its not possible to analyse the variable using that large name so we going to change it to **dpf**



```
{r}
# modify the data column names slightly for easier typing
names(diabetes)[7] <- "dpf"
names(diabetes) <- tolower(names(diabetes))

str(diabetes)
print(paste0("number of observations = ", dim(diabetes)[1]))
print(paste0("number of predictors = ", dim(diabetes)[2]))
```



```
'data.frame':   768 obs. of  9 variables:
 $ pregnancies  : int  6 1 8 1 0 5 3 10 2 8 ...
 $ glucose      : int  148 85 183 89 137 116 78 115 197 125 ...
 $ bloodpressure: int  72 66 64 66 40 74 50 0 70 96 ...
 $ skinthickness: int  35 29 0 23 35 0 32 0 45 0 ...
 $ insulin      : int  0 0 0 94 168 0 88 0 543 0 ...
 $ bmi          : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ dpf          : num  0.627 0.351 0.672 0.167 2.288 ...
 $ age          : int  50 31 32 21 33 30 26 29 53 54 ...
 $ outcome      : int  1 0 1 0 1 0 1 0 1 1 ...
[1] "number of observations = 768"
[1] "number of predictors = 9"
```

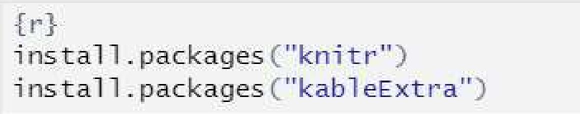
After undergoing data preprocessing like ,

- Checking for missing values
- Analyzing variables mean , max , min and quadrants.
- Changing column name

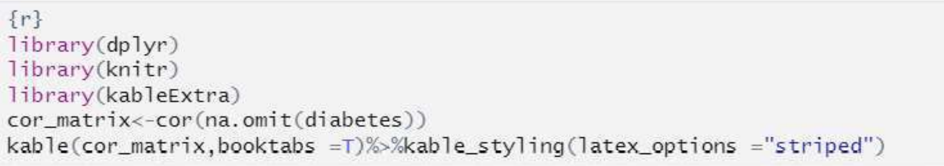
Moving to analyzing the data for predict the diabetes.

- **Needed Correlation table to know the relationships between the variables in the dataset**

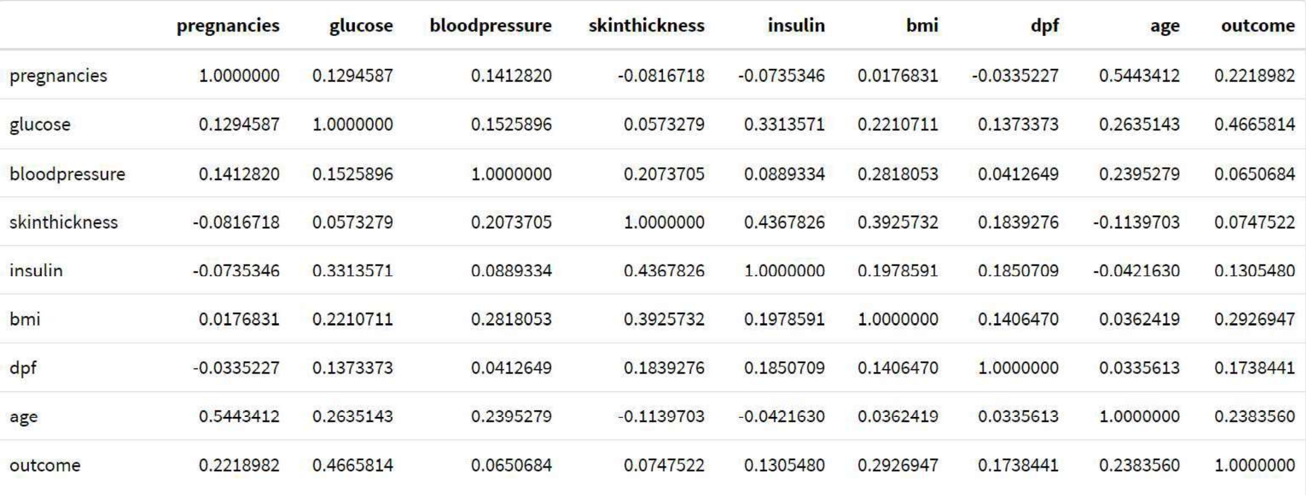
Required Packages



```
{r}
install.packages("knitr")
install.packages("kableExtra")
```



```
{r}
library(dplyr)
library(knitr)
library(kableExtra)
cor_matrix<-cor(na.omit(diabetes))
kable(cor_matrix,booktabs =T)%>%kable_styling(latex_options ="striped")
```



	pregnancies	glucose	bloodpressure	skinthickness	insulin	bmi	dpf	age	outcome
pregnancies	1.0000000	0.1294587	0.1412820	-0.0816718	-0.0735346	0.0176831	-0.0335227	0.5443412	0.2218982
glucose	0.1294587	1.0000000	0.1525896	0.0573279	0.3313571	0.2210711	0.1373373	0.2635143	0.4665814
bloodpressure	0.1412820	0.1525896	1.0000000	0.2073705	0.0889334	0.2818053	0.0412649	0.2395279	0.0650684
skinthickness	-0.0816718	0.0573279	0.2073705	1.0000000	0.4367826	0.3925732	0.1839276	-0.1139703	0.0747522
insulin	-0.0735346	0.3313571	0.0889334	0.4367826	1.0000000	0.1978591	0.1850709	-0.0421630	0.1305480
bmi	0.0176831	0.2210711	0.2818053	0.3925732	0.1978591	1.0000000	0.1406470	0.0362419	0.2926947
dpf	-0.0335227	0.1373373	0.0412649	0.1839276	0.1850709	0.1406470	1.0000000	0.0335613	0.1738441
age	0.5443412	0.2635143	0.2395279	-0.1139703	-0.0421630	0.0362419	0.0335613	1.0000000	0.2383560
outcome	0.2218982	0.4665814	0.0650684	0.0747522	0.1305480	0.2926947	0.1738441	0.2383560	1.0000000