VISVESVARAYATECHNOLOGICALUNIVERSITY
JNANASANGAMA, BELGAUM-590014

ADDITIONAL ACTIVITY-1

FOUNDATIONS OF DATA SCIENCE

INTRODUCTION TO R PROGRAMMING

Team member:

Yashaswini MB	**1SI19CS139**
Yashaswini GR	**1SI19CS143**
Shruthi N	**1SI19CS145**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belgaum, Recognized by AICTE and Accredited by NBA, New Delhi)

2021-2022

ACTIVITY 1

INTRODUCTION

R programming language is a open source programming language that has been widely used across the world. It is the language that can also help businesses analyze vast amount of information quickly and effectively.

2 Million Plus User across the world. And the number of user is steadily growing. R has set its foothold in the Data Science industry and owing to its massive repository, it has become the most sought after programming language in the world.

WHAT IS R PRORAMMING?

R is an open source programming language used for statistical computing. It is one of the most popular programming languages today. R was inspired by S+, it is similar to the S programming language.

FEATURES OF R

- **Open-source**

R is an open-source programming language. This means that it is free of cost and requires no license.

- **Comprehensive Language**

R is a comprehensive programming language, meaning that it provides services for statistical modeling as well as for software development. R is the primary language for Data Science as well as for developing web applications.

- **Provides a Wide Array of Packages**

R is most widely used because of its wide availability of libraries. R has CRAN, which is a repository holding more than 10,0000 packages.

- **Possesses a Number of Graphical Libraries**

The most important feature of R that sets it apart from other programming languages of Data Science is its massive collection of graphical libraries like ggplot2, plotly, etc. that are capable of making aesthetic and quality visualizations.

- **Performs Fast Calculations**

Through R, you can perform a wide variety of complex operations on vectors, arrays, data frames and other data objects of varying sizes. Furthermore, all these operations operate at a lightning speed.

CONCEPTS IN R

Variables

Variables can be created using the “<-” (assignment) operator.
Variables are case-sensitive. Var1 and var1 are considered different variables.
The class function can be used on variables to determine the data type of the variable which is classified in three major types – character, numeric and logical.

```
myString <- "Hello, World!"
```

If Else Statement

An if statement can be followed by an optional else statement which executes when the boolean expression is false.

Syntax

```
if(boolean_expression)
{ // statement(s) will execute if the boolean expression is true. }
else
{ // statement(s) will execute if the boolean expression is false. }
```

If the Boolean expression evaluates to be true, then the if block of code will be executed, otherwise else block of code will be executed.

While Loop

The While loop executes the same code again and again until a stop condition is met

Syntax

```
while (test_expression) {
 statements
 }
```

Function Definition

A function is a set of statements organized together to perform a specific task.

An R function is created by using the keyword function.

syntax

```
function_name <- function(arg_1, arg_2, ...){
Function body
 }
```

Paste function

Paste function in R is used to concatenate Vectors by converting them into character.It also used to concatenate the two string values by separating with delimiters sepecified by keyword sep. Example

```
paste('one',2,'three',4,'five',sep=',')
1] "one,2 ,three,4,five"
```

Floor function

The R floor method is one of the R Math functions, which is to return the largest integer value. That is not greater than (less than) or equal to a specific number or an expression.

Example

```
print(floor(2.6))
[1]  2
```

Caret symbol

It is the exponent operator. calculate a base number raised to the power of exponent number.

Example

```
print(2^3)
[1]  8
```

PROGRAM TO CHECK WHETHER GIVEN NUMBER IS ARMSTRONG OR NOT

```
isArmstrong <- function(x){
 sum <- 0
 y <- x
 while(y > 0){
  digit <- y %% 10
  sum <- sum + (digit^3)
  y <- floor(y/10)
 }
 if(x == sum){
  print(paste(x,' is an Armstrong Number '))
 } else {
  print(paste(x,'is not an Armstrong Number'))
 }
}
# take input from the user
num = as.integer(readline(prompt="Enter a number: "))
isArmstrong (num)
```

Explanation:

To check whether the number is armstrong or not, we have defined a function isArmstrong with x as the argument here x is the number which we have to check if it is Armstrong or not.

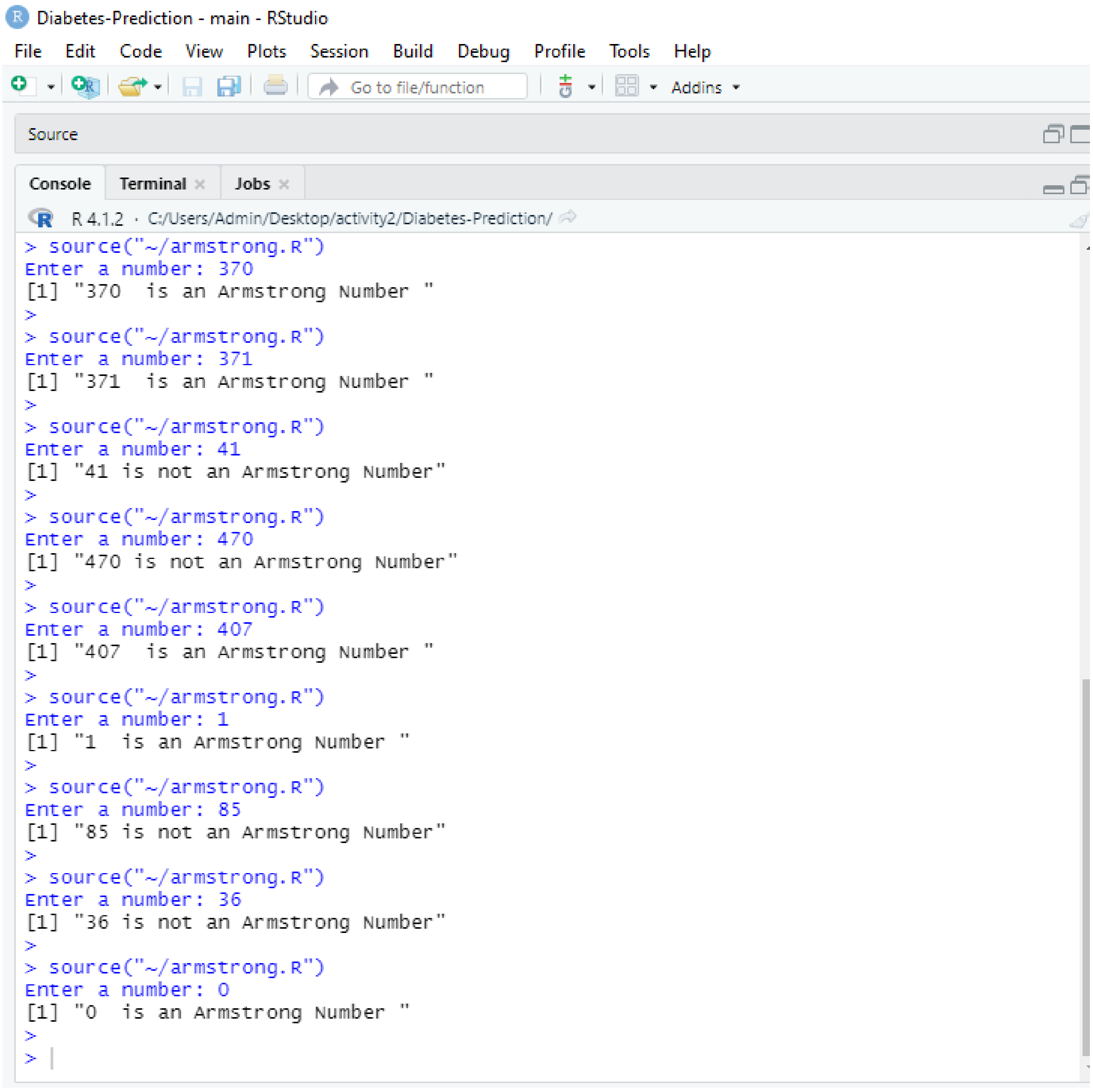
We need to calculate the sum of the cube of each digit. So, we initialize the sum to 0 and obtain each digit number by using the **modulus operator** %%.

The remainder of a number when it is divided by 10 is the last digit of that number. We take the cubes using **exponent operator**.

Finally, we compare the sum with the original number and conclude that it is Armstrong number if they are equal.

To read the input from user the **prompt** argument is chosen to display an appropriate message for the user. Use **readline()** function to take input from the user (terminal). And we convert the input, which is a character vector into integer using the function **as.integer()**.

OUTPUT


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/

```
> source("~/armstrong.R")
Enter a number: 370
[1] "370  is an Armstrong Number "
>
> source("~/armstrong.R")
Enter a number: 371
[1] "371  is an Armstrong Number "
>
> source("~/armstrong.R")
Enter a number: 41
[1] "41 is not an Armstrong Number"
>
> source("~/armstrong.R")
Enter a number: 470
[1] "470 is not an Armstrong Number"
>
> source("~/armstrong.R")
Enter a number: 407
[1] "407  is an Armstrong Number "
>
> source("~/armstrong.R")
Enter a number: 1
[1] "1  is an Armstrong Number "
>
> source("~/armstrong.R")
Enter a number: 85
[1] "85 is not an Armstrong Number"
>
> source("~/armstrong.R")
Enter a number: 36
[1] "36 is not an Armstrong Number"
>
> source("~/armstrong.R")
Enter a number: 0
[1] "0  is an Armstrong Number "
>
> 
```

ACTIVITY 2

INTRODUCTION

Diabetes:

Diabetes is a standout amongst the most well-known non-transmittable diseases in the world. It is assessed to be the seventh leading cause for death. Diabetes causes a large number of deaths each year and a large number of people living with the disease do not realize their health condition early enough.

According to a study by the World Health Organization (WHO), this number will have raised to 552 million by 2030, denote that one in 10 grownups will have diabetes by 2030 if no serious act is taken. In 2014, the worldwide frequency of diabetes was projected to be 9 % among adults aged 18+ years

Diabetes is a disease caused due to the increase level of blood glucose. Diabetes is a chronic disease with the potential to cause a worldwide health care crisis.

Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. However, early prediction of diabetes is quite challenging task for medical practitioners due to complex interdependence on various factors as diabetes affects human organs such as kidney, eye, heart, nerves, foot etc. Data science methods have the potential to benefit other scientific fields by shedding new light on common questions.

One such task is to help make predictions on medical data. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience.

OBSERVATION

The number of people with diabetes has risen from 108 million in 1980 to 422 million in 2014, with the global prevalence of diabetes among adults over 18 years of age rising from 4.7% in 1980 to 8.5% in 2014. By 2040, 642 million adults (1 in 10 adults) are expected to have diabetes. Also, 46.5% of those with diabetes have not been diagnosed. In order to reduce the number of deaths attributable to diabetes, it is essential that methods and techniques that will aid in early diagnosis of diabetes be devised, because a large number of deaths in diabetic patients are due to late diagnosis.

PROBLEM STATEMENT

The improved model for early diabetes prediction by integrating Logistic Regression and Decision Tree techniques.

APPROACHES

- **Logistic Regression**

Logistic regression is a classification model in machine learning, extensively used in clinical analysis. It uses probabilistic estimations which helps in understanding the relationship between the dependent variable and one or more independent variables. Diabetes, being one of the most common diseases around the world, when detected early, may prevent the progression of the disease and avoid other complications. In this work, we design a prediction model, that predicts whether a patient has diabetes, based on certain diagnostic measurements included in the dataset, and explore various techniques to boost the performance and accuracy.

- **Decision Tree**

Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

DATASET

The data set used for the purpose of this study is Pima Indians Diabetes Database of National Institute of Diabetes and Digestive and Kidney Diseases. This diabetes database is a collection of medical diagnostic reports of 768 examples from a population living near Phoenix, Arizona, USA. The samples consist of examples with 8 attribute values and one of the two possible outcomes, namely whether the patient is tested positive for diabetes (indicated by output one) or not (indicated by zero).

Attribute Information:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

DESCRIPTIVE STATISTICS

IMPORTING LIBRARIES:

Libraries for Box Plot and Histogram:

library(ggplot2)

library(dplyr)

library(gridExtra)

library(corrplot)

Libraraies for Class Distribution:

library(car)

library(caret)

library(class)

library(dplyr)

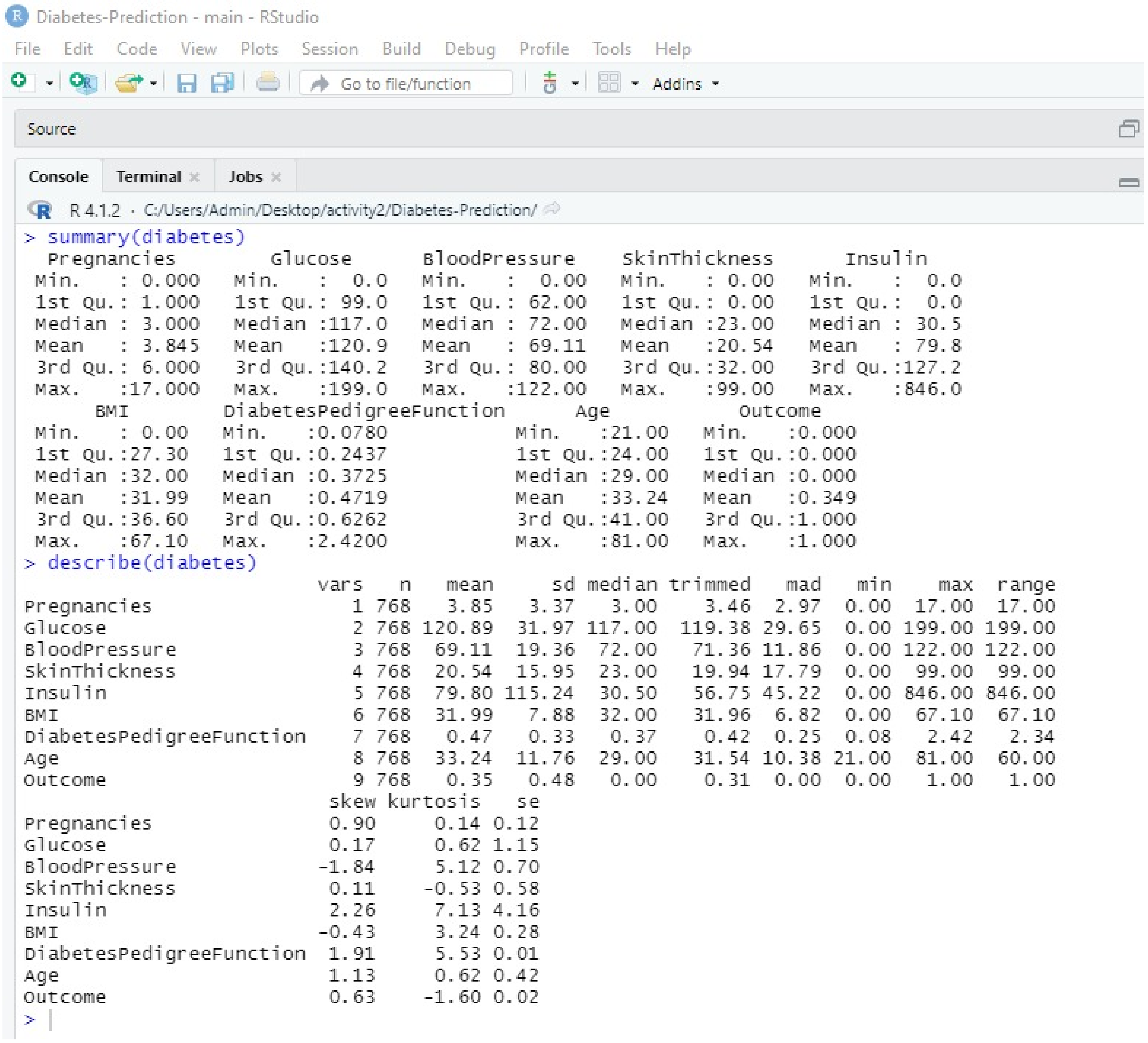
library(ggplot2)

library(tidyr)

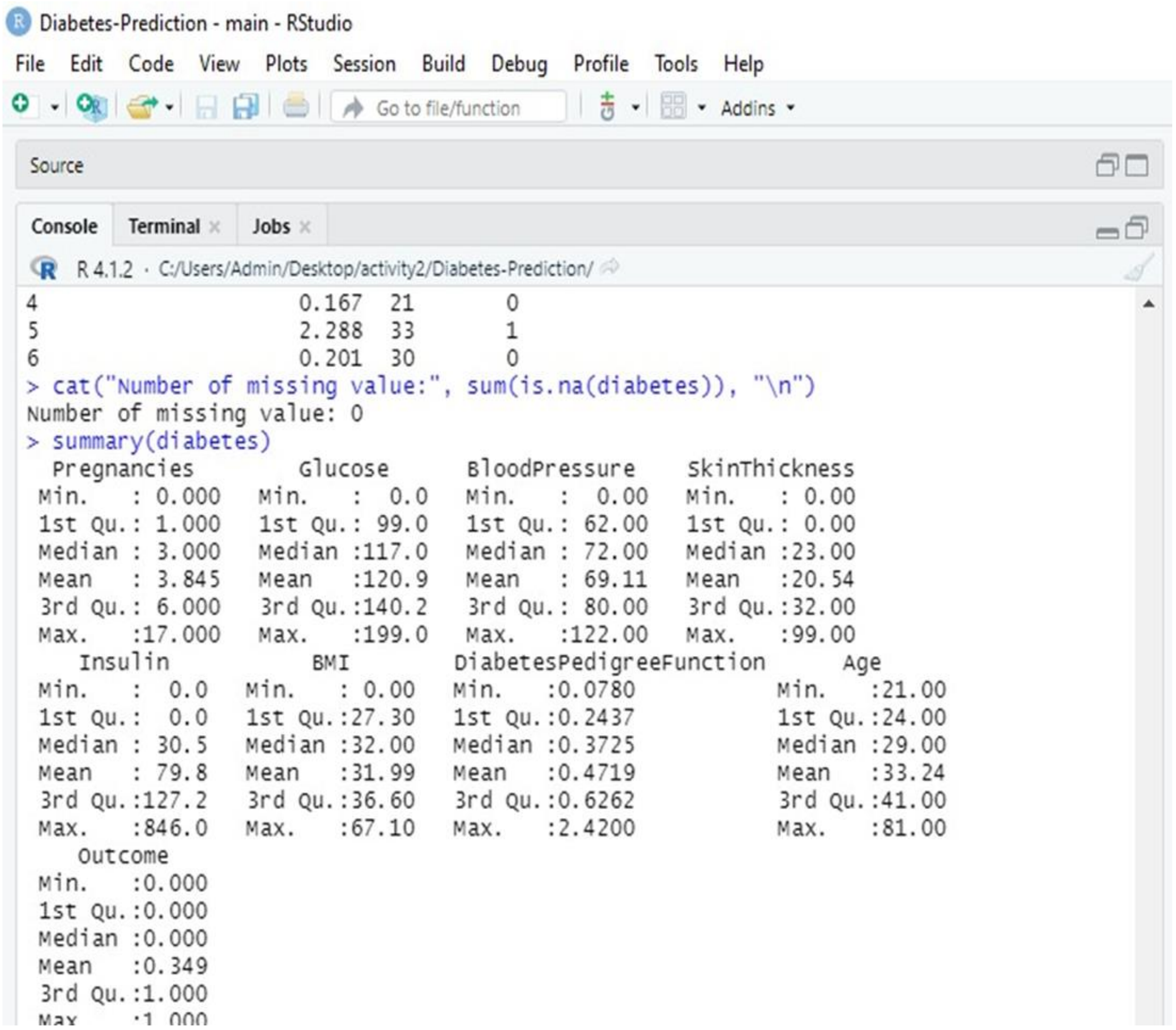
library(tidyverse)

library(performance)

DATA SUMMARY

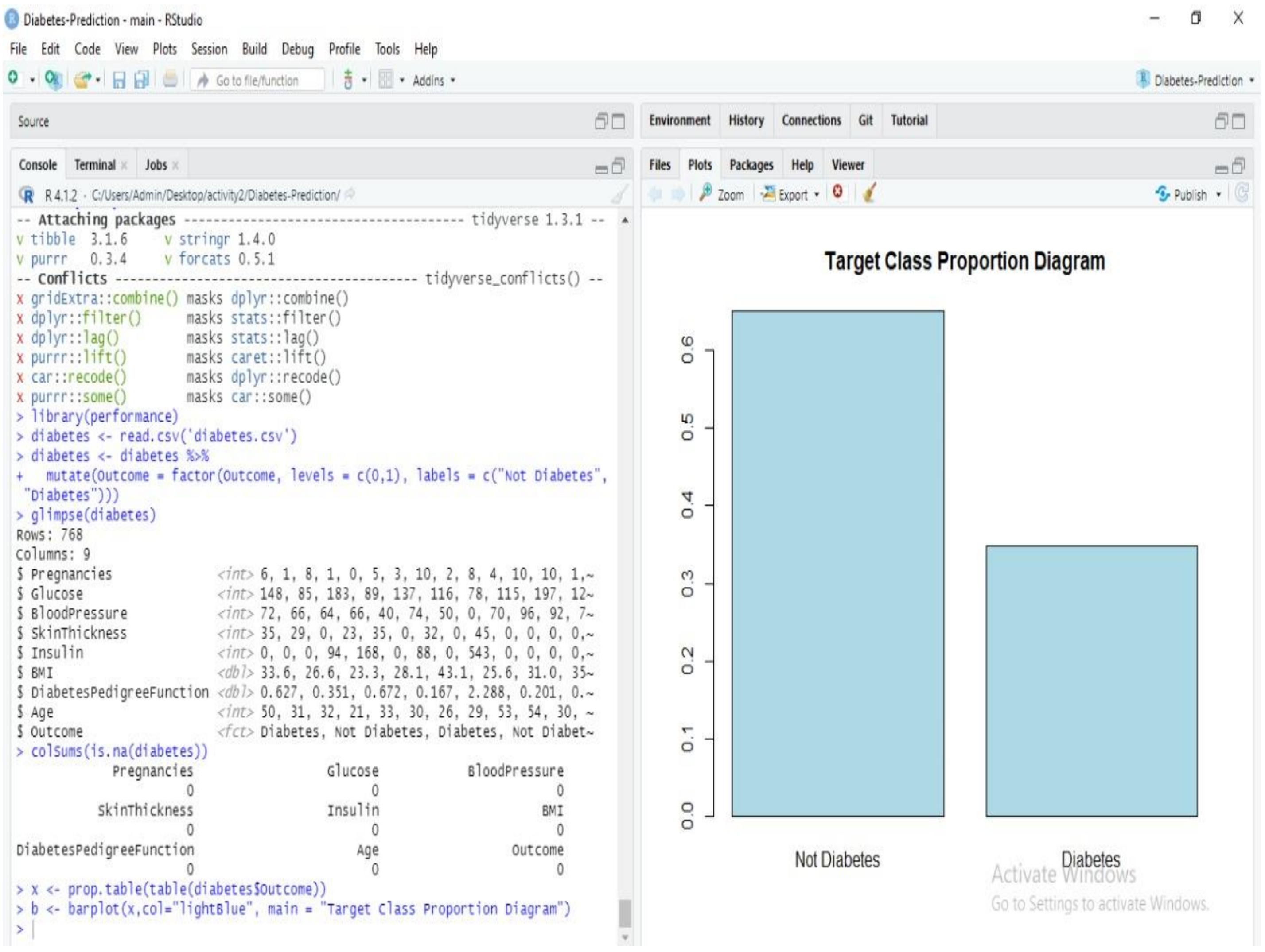

Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console Terminal Jobs
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/

```
> summary(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
> describe(diabetes)
                         vars   n   mean     sd median trimmed   mad   min    max  range
Pregnancies                 1 768   3.85   3.37   3.00    3.46  2.97  0.00  17.00  17.00
Glucose                     2 768 120.89  31.97 117.00  119.38 29.65  0.00 199.00 199.00
BloodPressure               3 768  69.11  19.36  72.00   71.36 11.86  0.00 122.00 122.00
SkinThickness               4 768  20.54  15.95  23.00   19.94 17.79  0.00  99.00  99.00
Insulin                     5 768  79.80 115.24  30.50   56.75 45.22  0.00 846.00 846.00
BMI                         6 768  31.99   7.88  32.00   31.96  6.82  0.00  67.10  67.10
DiabetesPedigreeFunction    7 768   0.47   0.33   0.37    0.42  0.25  0.08   2.42   2.34
Age                         8 768  33.24  11.76  29.00   31.54 10.38 21.00  81.00  60.00
Outcome                     9 768   0.35   0.48   0.00    0.31  0.00  0.00   1.00   1.00
                          skew kurtosis   se
Pregnancies               0.90     0.14 0.12
Glucose                   0.17     0.62 1.15
BloodPressure            -1.84     5.12 0.70
SkinThickness             0.11    -0.53 0.58
Insulin                   2.26     7.13 4.16
BMI                      -0.43     3.24 0.28
DiabetesPedigreeFunction  1.91     5.53 0.01
Age                       1.13     0.62 0.42
Outcome                   0.63    -1.60 0.02
> 
```


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/

```
4                    0.167  21       0
5                    2.288  33       1
6                    0.201  30       0
> cat("Number of missing value:", sum(is.na(diabetes)), "\n")
Number of missing value: 0
> summary(diabetes)
  Pregnancies        Glucose      BloodPressure    SkinThickness
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
    Insulin           BMI        DiabetesPedigreeFunction      Age
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
 Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
 Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
    Outcome
 Min.   :0.000
 1st Qu.:0.000
 Median :0.000
 Mean   :0.349
 3rd Qu.:1.000
 Max.   :1.000
```

CLASS DISTRIBUTION

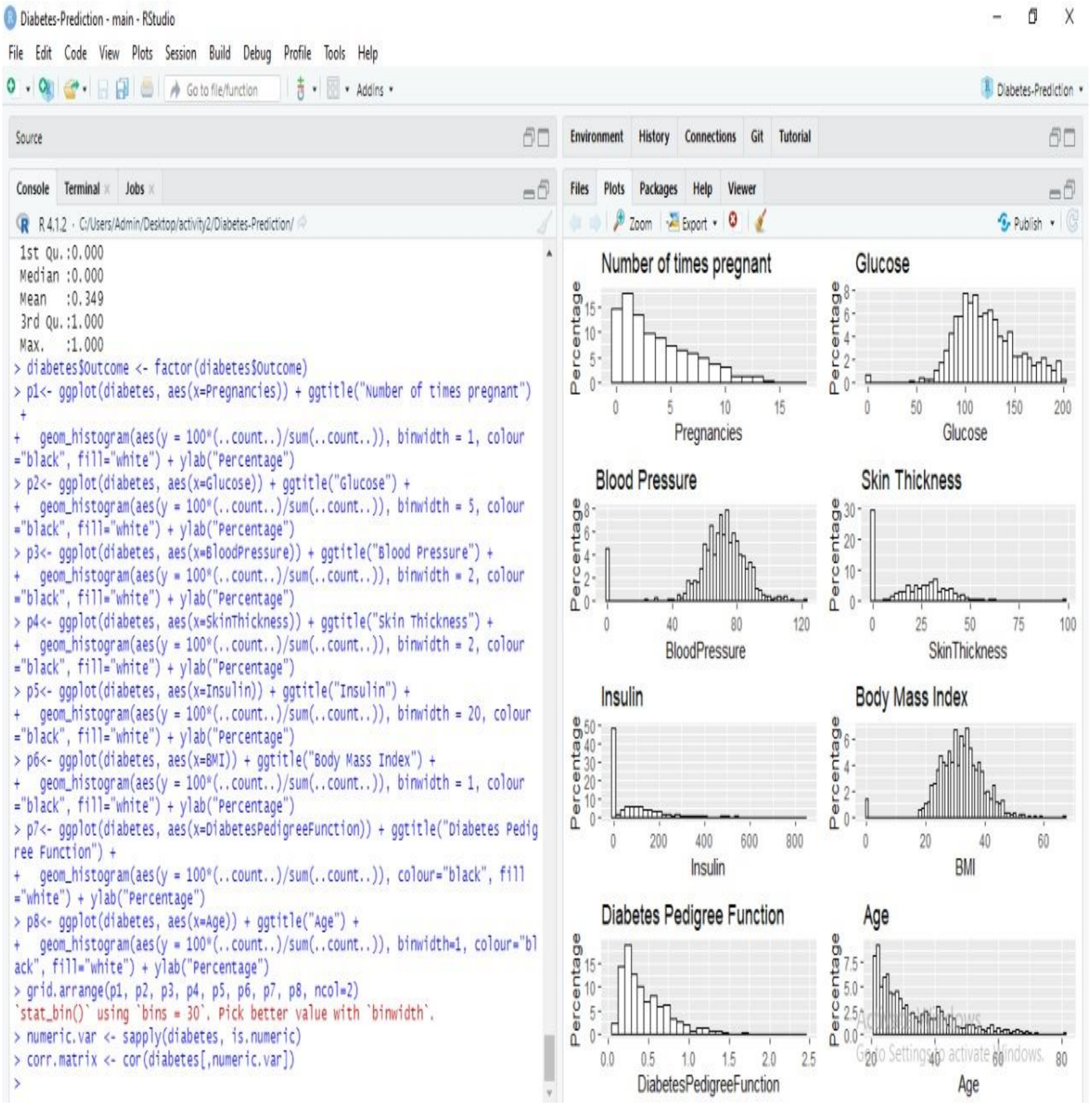


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Diabetes-Prediction
Source
Environment History Connections Git Tutorial
Console Terminal Jobs
Files Plots Packages Help Viewer
Zoom Export
Publish
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/
Target Class Proportion Diagram
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Not Diabetes
Diabetes
Activate Windows
Go to Settings to activate Windows.

```
-- Attaching packages ------------------------------------ tidyverse 1.3.1 --
v tibble  3.1.6     v stringr 1.4.0
v purrr   0.3.4     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x gridExtra::combine() masks dplyr::combine()
x dplyr::filter()      masks stats::filter()
x dplyr::lag()         masks stats::lag()
x purrr::lift()        masks caret::lift()
x car::recode()        masks dplyr::recode()
x purrr::some()        masks car::some()
> library(performance)
> diabetes <- read.csv('diabetes.csv')
> diabetes <- diabetes %>%
+   mutate(Outcome = factor(Outcome, levels = c(0,1), labels = c("Not Diabetes",
 "Diabetes")))
> glimpse(diabetes)
Rows: 768
Columns: 9
$ Pregnancies              <int> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1,~
$ Glucose                  <int> 148, 85, 183, 89, 137, 116, 78, 115, 197, 12~
$ BloodPressure            <int> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96, 92, 7~
$ SkinThickness            <int> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0, 0,~
$ Insulin                  <int> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0, 0,~
$ BMI                      <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, 35~
$ DiabetesPedigreeFunction <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.~
$ Age                      <int> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30, ~
$ Outcome                  <fct> Diabetes, Not Diabetes, Diabetes, Not Diabet~
> colSums(is.na(diabetes))
             Pregnancies                  Glucose            BloodPressure 
                       0                        0                        0 
           SkinThickness                  Insulin                      BMI 
                       0                        0                        0 
DiabetesPedigreeFunction                      Age                  Outcome 
                       0                        0                        0 
> x <- prop.table(table(diabetes$Outcome))
> b <- barplot(x,col="lightBlue", main = "Target Class Proportion Diagram")
> 
```

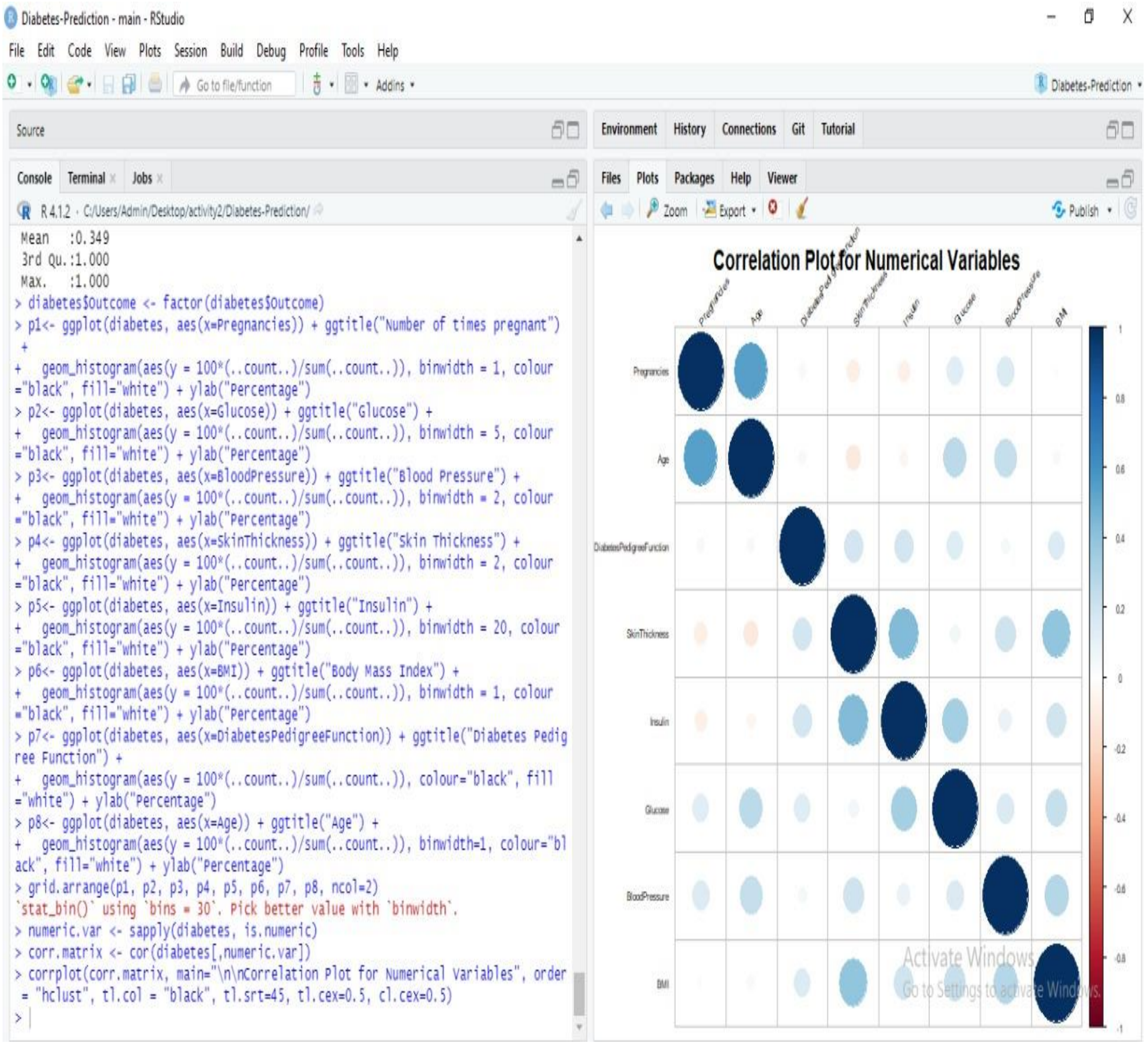
DATA VISUALISATION

Histogram Plot:


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Diabetes-Prediction
Source
Environment History Connections Git Tutorial
Console Terminal Jobs
Files Plots Packages Help Viewer
Zoom Export Publish
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/
Number of times pregnant
Percentage
Pregnancies
Glucose
Blood Pressure
BloodPressure
Skin Thickness
SkinThickness
Insulin
Body Mass Index
BMI
Diabetes Pedigree Function
DiabetesPedigreeFunction
Age

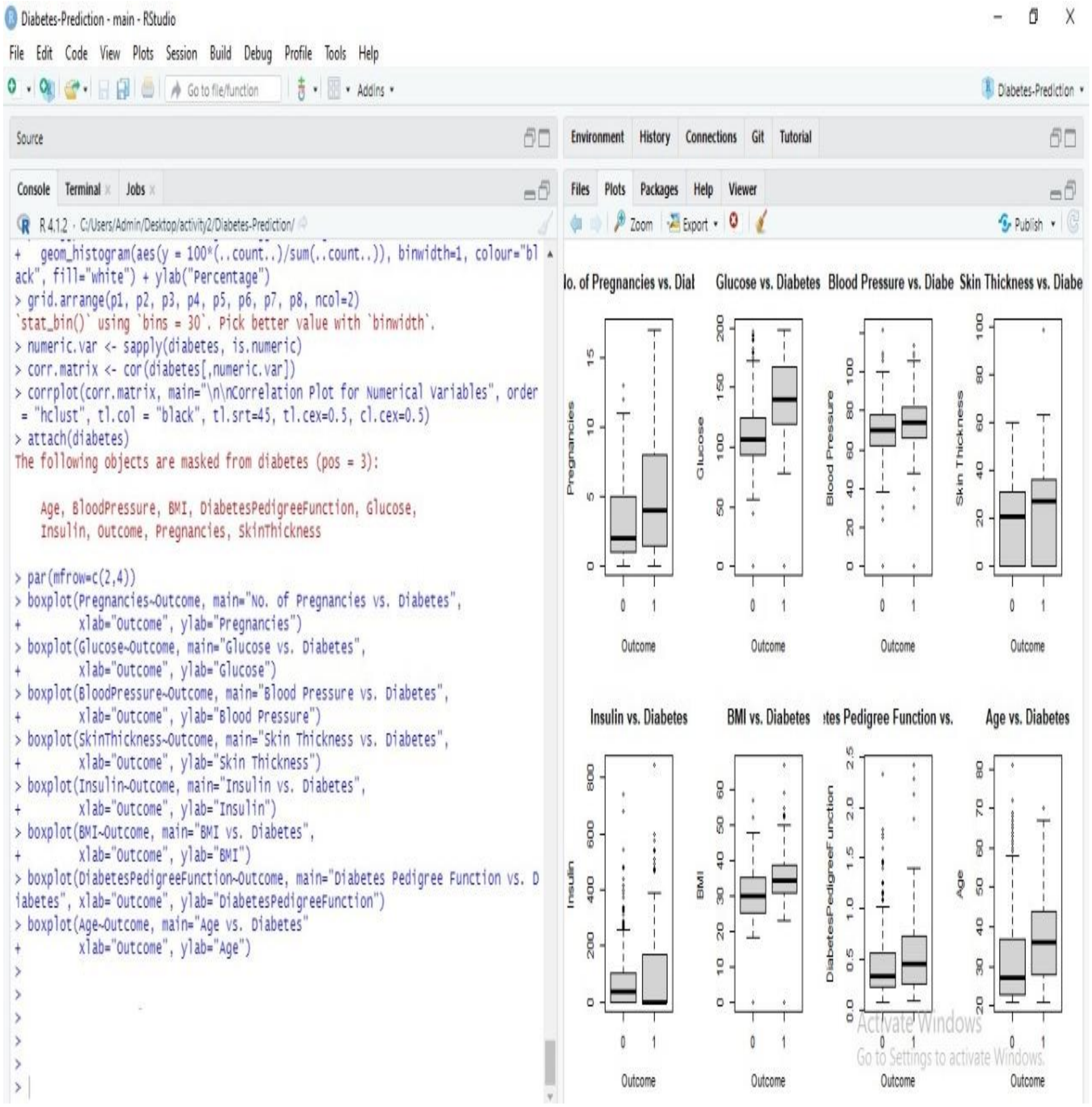
```
 1st Qu.:0.000
 Median :0.000
 Mean   :0.349
 3rd Qu.:1.000
 Max.   :1.000
> diabetes$Outcome <- factor(diabetes$Outcome)
> p1<- ggplot(diabetes, aes(x=Pregnancies)) + ggtitle("Number of times pregnant")
 +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 1, colour
="black", fill="white") + ylab("Percentage")
> p2<- ggplot(diabetes, aes(x=Glucose)) + ggtitle("Glucose") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 5, colour
="black", fill="white") + ylab("Percentage")
> p3<- ggplot(diabetes, aes(x=BloodPressure)) + ggtitle("Blood Pressure") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 2, colour
="black", fill="white") + ylab("Percentage")
> p4<- ggplot(diabetes, aes(x=SkinThickness)) + ggtitle("Skin Thickness") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 2, colour
="black", fill="white") + ylab("Percentage")
> p5<- ggplot(diabetes, aes(x=Insulin)) + ggtitle("Insulin") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 20, colour
="black", fill="white") + ylab("Percentage")
> p6<- ggplot(diabetes, aes(x=BMI)) + ggtitle("Body Mass Index") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 1, colour
="black", fill="white") + ylab("Percentage")
> p7<- ggplot(diabetes, aes(x=DiabetesPedigreeFunction)) + ggtitle("Diabetes Pedig
ree Function") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), colour="black", fill
="white") + ylab("Percentage")
> p8<- ggplot(diabetes, aes(x=Age)) + ggtitle("Age") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth=1, colour="bl
ack", fill="white") + ylab("Percentage")
> grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, ncol=2)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> numeric.var <- sapply(diabetes, is.numeric)
> corr.matrix <- cor(diabetes[,numeric.var])
>
```

Correlation Plot:


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Diabetes-Prediction
Source
Environment History Connections Git Tutorial
Console Terminal Jobs
Files Plots Packages Help Viewer
Zoom Export Publish
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/
Correlation Plot for Numerical Variables
Pregnancies
Age
DiabetesPedigreeFunction
SkinThickness
Insulin
Glucose
BloodPressure
BMI
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Activate Windows
Go to Settings to activate Windows.

```
 Mean   :0.349
 3rd Qu.:1.000
 Max.   :1.000
> diabetes$Outcome <- factor(diabetes$Outcome)
> p1<- ggplot(diabetes, aes(x=Pregnancies)) + ggtitle("Number of times pregnant")
 +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 1, colour
="black", fill="white") + ylab("Percentage")
> p2<- ggplot(diabetes, aes(x=Glucose)) + ggtitle("Glucose") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 5, colour
="black", fill="white") + ylab("Percentage")
> p3<- ggplot(diabetes, aes(x=BloodPressure)) + ggtitle("Blood Pressure") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 2, colour
="black", fill="white") + ylab("Percentage")
> p4<- ggplot(diabetes, aes(x=SkinThickness)) + ggtitle("Skin Thickness") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 2, colour
="black", fill="white") + ylab("Percentage")
> p5<- ggplot(diabetes, aes(x=Insulin)) + ggtitle("Insulin") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 20, colour
="black", fill="white") + ylab("Percentage")
> p6<- ggplot(diabetes, aes(x=BMI)) + ggtitle("Body Mass Index") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 1, colour
="black", fill="white") + ylab("Percentage")
> p7<- ggplot(diabetes, aes(x=DiabetesPedigreeFunction)) + ggtitle("Diabetes Pedig
ree Function") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), colour="black", fill
="white") + ylab("Percentage")
> p8<- ggplot(diabetes, aes(x=Age)) + ggtitle("Age") +
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth=1, colour="bl
ack", fill="white") + ylab("Percentage")
> grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, ncol=2)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> numeric.var <- sapply(diabetes, is.numeric)
> corr.matrix <- cor(diabetes[,numeric.var])
> corrplot(corr.matrix, main="\n\nCorrelation Plot for Numerical Variables", order
 = "hclust", tl.col = "black", tl.srt=45, tl.cex=0.5, cl.cex=0.5)
> 
```

Box and Whisker Plot:


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Diabetes-Prediction
Source
Environment History Connections Git Tutorial
Console Terminal Jobs
Files Plots Packages Help Viewer
Zoom Export
Publish
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/
lo. of Pregnancies vs. Diał
Glucose vs. Diabetes
Blood Pressure vs. Diabe
Skin Thickness vs. Diabe
Pregnancies
Glucose
Blood Pressure
Skin Thickness
Insulin vs. Diabetes
BMI vs. Diabetes
ıtes Pedigree Function vs.
Age vs. Diabetes
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
Activate Windows
Go to Settings to activate Windows.

```
+   geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth=1, colour="bl
ack", fill="white") + ylab("Percentage")
> grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, ncol=2)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> numeric.var <- sapply(diabetes, is.numeric)
> corr.matrix <- cor(diabetes[,numeric.var])
> corrplot(corr.matrix, main="\n\nCorrelation Plot for Numerical Variables", order
 = "hclust", tl.col = "black", tl.srt=45, tl.cex=0.5, cl.cex=0.5)
> attach(diabetes)
The following objects are masked from diabetes (pos = 3):

    Age, BloodPressure, BMI, DiabetesPedigreeFunction, Glucose,
    Insulin, Outcome, Pregnancies, SkinThickness

> par(mfrow=c(2,4))
> boxplot(Pregnancies~Outcome, main="No. of Pregnancies vs. Diabetes",
+         xlab="Outcome", ylab="Pregnancies")
> boxplot(Glucose~Outcome, main="Glucose vs. Diabetes",
+         xlab="Outcome", ylab="Glucose")
> boxplot(BloodPressure~Outcome, main="Blood Pressure vs. Diabetes",
+         xlab="Outcome", ylab="Blood Pressure")
> boxplot(SkinThickness~Outcome, main="Skin Thickness vs. Diabetes",
+         xlab="Outcome", ylab="Skin Thickness")
> boxplot(Insulin~Outcome, main="Insulin vs. Diabetes",
+         xlab="Outcome", ylab="Insulin")
> boxplot(BMI~Outcome, main="BMI vs. Diabetes",
+         xlab="Outcome", ylab="BMI")
> boxplot(DiabetesPedigreeFunction~Outcome, main="Diabetes Pedigree Function vs. D
iabetes", xlab="Outcome", ylab="DiabetesPedigreeFunction")
> boxplot(Age~Outcome, main="Age vs. Diabetes"
+         xlab="Outcome", ylab="Age")
>
>
>
>
>
> |
```

PREDICTION ON DATA

Logistic regression:

we use a training data set containing a random sample of 70% of the observation to perform a Logistic Regression with “Diabetes” as the response and the remains variables as predictors.

Algorithm of Logistic Regression in R:

x <- cbind(x_train,y_train)

# Train the model using the training sets and check score

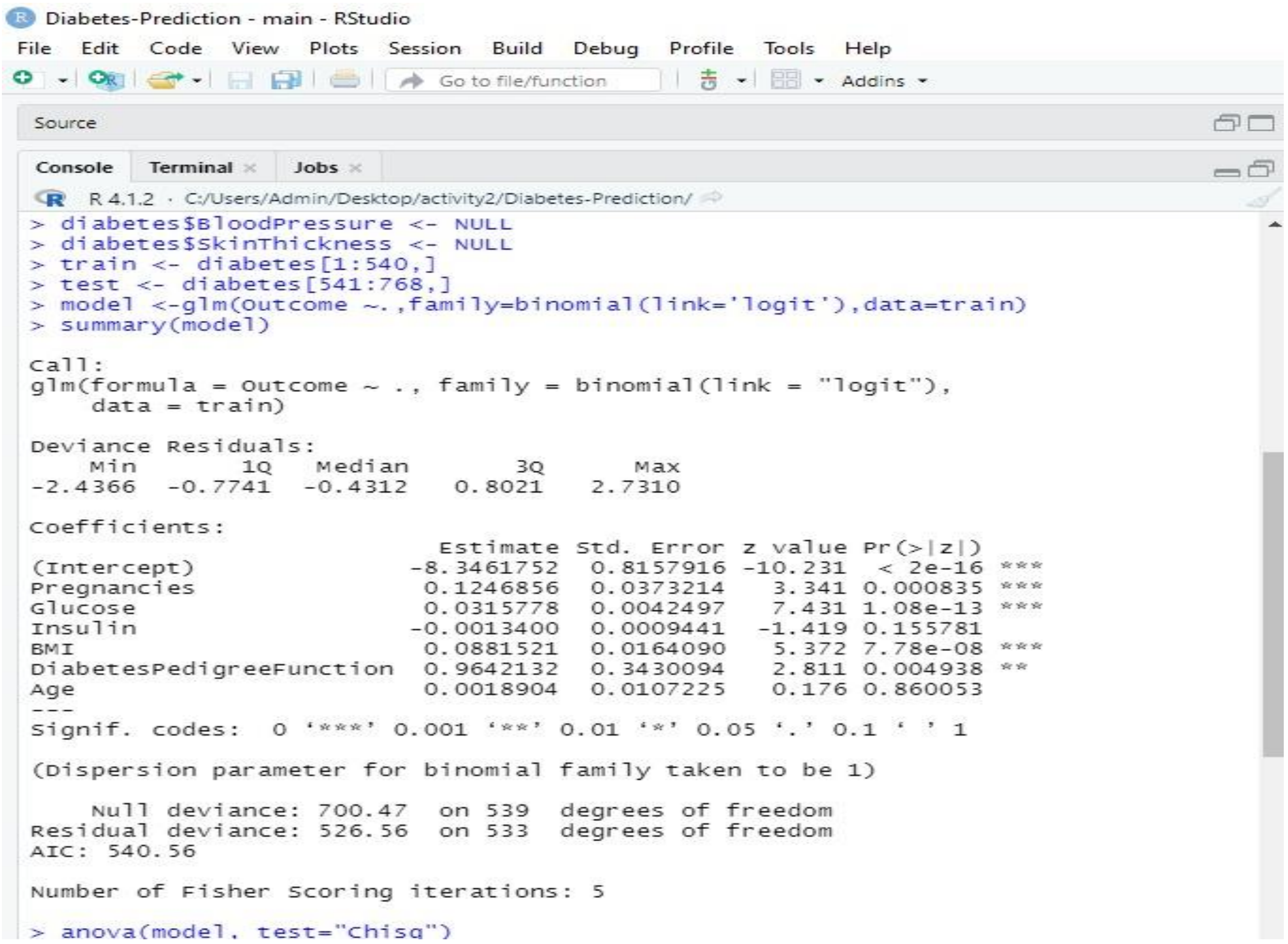
model <- glm(train ~ ., data = x,family='binomial')

summary(logistic)

#Predict Output

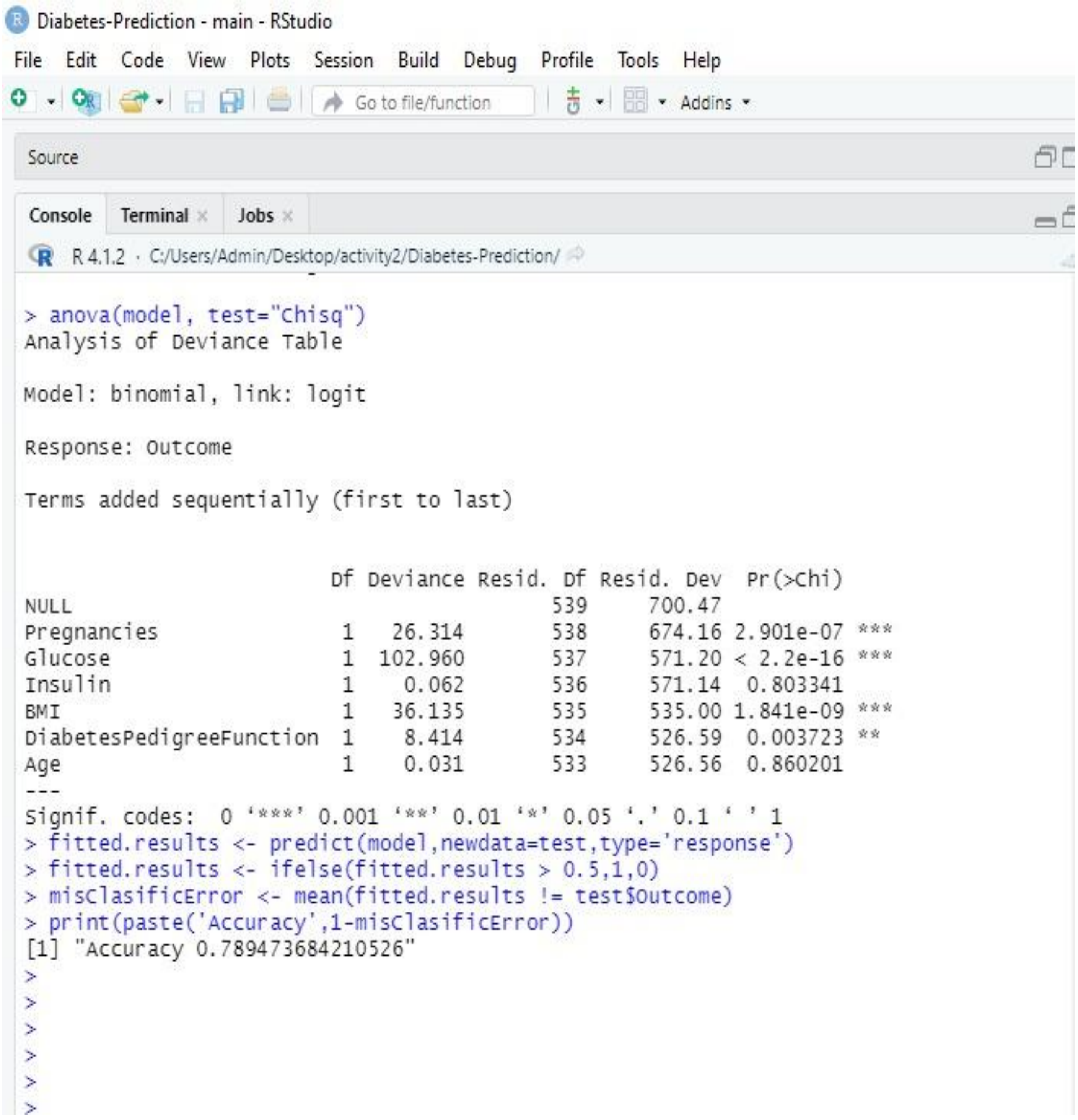
predicted= predict(logistic,test)

Output:


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console Terminal Jobs
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/

```
> diabetes$BloodPressure <- NULL
> diabetes$SkinThickness <- NULL
> train <- diabetes[1:540,]
> test <- diabetes[541:768,]
> model <-glm(Outcome ~.,family=binomial(link='logit'),data=train)
> summary(model)

Call:
glm(formula = Outcome ~ ., family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4366  -0.7741  -0.4312   0.8021   2.7310

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -8.3461752  0.8157916 -10.231  < 2e-16 ***
Pregnancies              0.1246856  0.0373214   3.341 0.000835 ***
Glucose                  0.0315778  0.0042497   7.431 1.08e-13 ***
Insulin                 -0.0013400  0.0009441  -1.419 0.155781
BMI                      0.0881521  0.0164090   5.372 7.78e-08 ***
DiabetesPedigreeFunction 0.9642132  0.3430094   2.811 0.004938 **
Age                      0.0018904  0.0107225   0.176 0.860053
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 700.47  on 539  degrees of freedom
Residual deviance: 526.56  on 533  degrees of freedom
AIC: 540.56

Number of Fisher Scoring iterations: 5

> anova(model, test="Chisq")
```

The result shows that the variables Triceps_Skin, Serum_Insulin and Age are not statiscally significance. In other words, the p_values is greather than 0.01. Therefore they will be removed.


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/

```
> anova(model, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: Outcome

Terms added sequentially (first to last)


                         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
NULL                                       539     700.47              
Pregnancies               1   26.314       538     674.16 2.901e-07 ***
Glucose                   1  102.960       537     571.20 < 2.2e-16 ***
Insulin                   1    0.062       536     571.14  0.803341    
BMI                       1   36.135       535     535.00 1.841e-09 ***
DiabetesPedigreeFunction  1    8.414       534     526.59  0.003723 ** 
Age                       1    0.031       533     526.56  0.860201    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fitted.results <- predict(model,newdata=test,type='response')
> fitted.results <- ifelse(fitted.results > 0.5,1,0)
> misClasificError <- mean(fitted.results != test$Outcome)
> print(paste('Accuracy',1-misClasificError))
[1] "Accuracy 0.789473684210526"
>
>
>
>
>
>
```

Accuracy of Logistic Regression is 78.94%

Decision Tree:

Now we present the Decision Trees algorithm

Algorithm of Decision Tree In R:

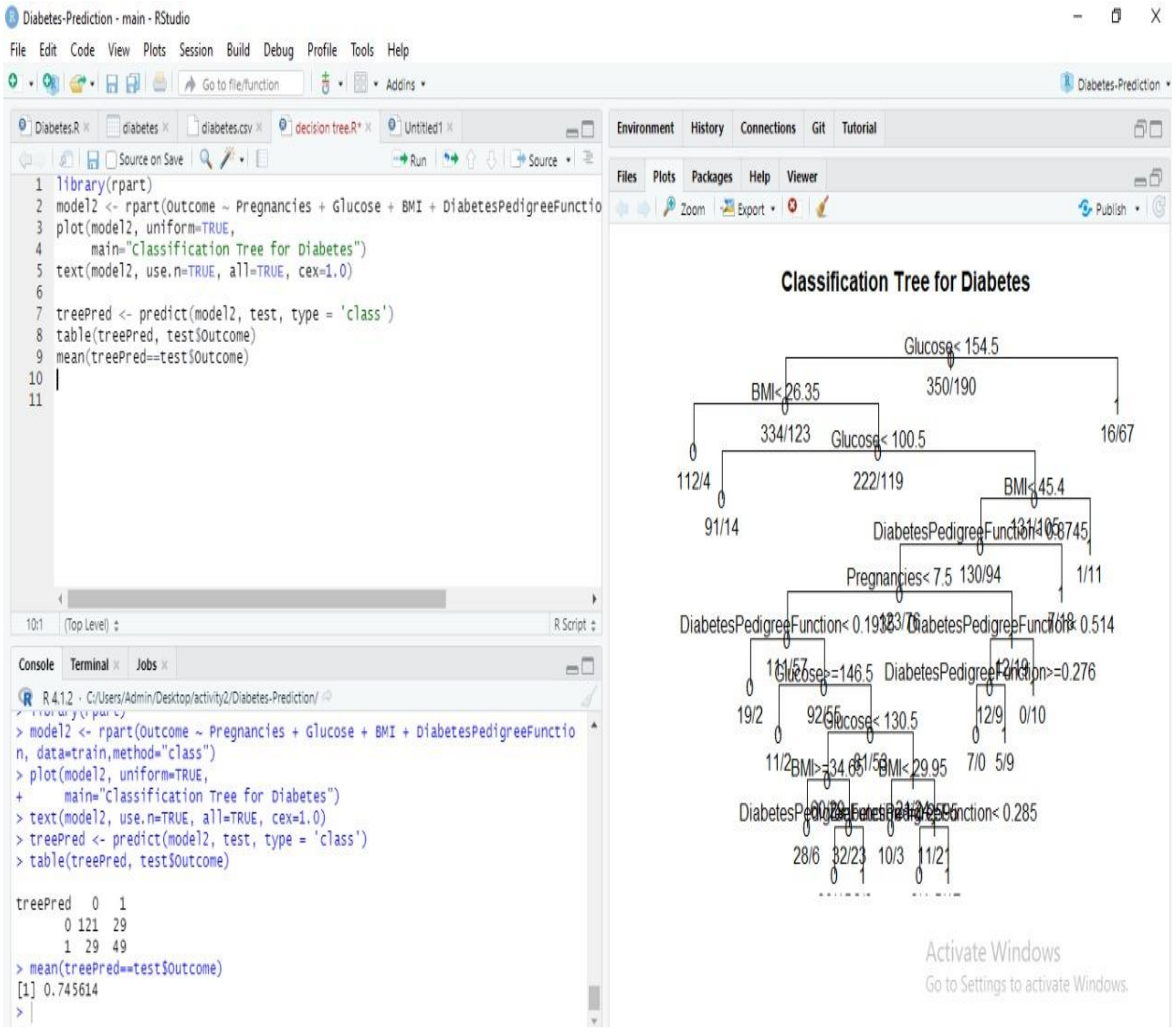
```
library(rpart)

x <- cbind(x_train,y_train)

# grow tree

fit <- rpart(y_train ~ ., data = x,method="class")

summary(fit)

#Predict Output

predicted= predict(fit,x_test)
```

Output:

The results display the split criterion (e.g. Plasma_Glucose < 154.5), the number of observations in that branch, the deviance, the overall prediction for the branch (Yes or No), and the fraction of observations in that branch that take on values of Yes and No. Branches that lead to terminal nodes are indicated using asterisks.

Now we plot of the tree, and interpret the results.

“Diabetes” appears to be Plasma_Glucose, since the first branch split criterion (e.g. Plasma_Glucose <154.5).


Diabetes-Prediction - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Diabetes.R
diabetes
diabetes.csv
decision tree.R*
Untitled1
Source on Save
Run
Source
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer
Zoom Export
Publish
Console Terminal Jobs
R 4.1.2 · C:/Users/Admin/Desktop/activity2/Diabetes-Prediction/
Classification Tree for Diabetes
Glucose< 154.5
350/190
BMI< 26.35
334/123
16/67
Glucose< 100.5
112/4
222/119
91/14
BMI< 45.4
DiabetesPedigreeFunction< 0.8745
131/105
Pregnancies< 7.5
130/94
1/11
DiabetesPedigreeFunction< 0.1935
123/76
DiabetesPedigreeFunction< 0.514
7/18
Glucose>=146.5
DiabetesPedigreeFunction>=0.276
19/2
92/55
Glucose< 130.5
12/9
0/10
11/2
BMI>=34.65
81/53
BMI< 29.95
7/0
5/9
DiabetesPedigreeFunction< 0.285
28/6
32/23
10/3
11/21
Activate Windows
Go to Settings to activate Windows.

```
library(rpart)
model2 <- rpart(Outcome ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunctio
plot(model2, uniform=TRUE,
     main="Classification Tree for Diabetes")
text(model2, use.n=TRUE, all=TRUE, cex=1.0)

treePred <- predict(model2, test, type = 'class')
table(treePred, test$Outcome)
mean(treePred==test$Outcome)
```

```
> model2 <- rpart(Outcome ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunctio
n, data=train,method="class")
> plot(model2, uniform=TRUE,
+      main="Classification Tree for Diabetes")
> text(model2, use.n=TRUE, all=TRUE, cex=1.0)
> treePred <- predict(model2, test, type = 'class')
> table(treePred, test$Outcome)

treePred   0   1
       0 121  29
       1  29  49
> mean(treePred==test$Outcome)
[1] 0.745614
>
```

The test error rate is 30%. In other words, the accuracy is 70%.