Analysis of Diabetes in Pima Indian Women

Arefe Asadi

2025-08-18

1. Introduction

This report presents an analysis of a medical data set concerning diabetes among Pima Indian women.

We begin the analysis by loading the required R packages, including the dataset source and tools for data cleaning and visualization:

```
library(MASS)        # Contains the Pima.tr and Pima.tr2 datasets
library(tidyverse)   # Data wrangling and visualization
library(naniar)      # Missing data visualization
library(gtExtras)    # Enhanced table formatting
library(ggplot2)     # Visualization
library(gridExtra)   # Arrange multiple plots or tables in a grid layout
library(grid)        # Base system for creating and positioning graphical objects (grobs)
library(janitor)     # Data cleaning and frequency tables
library(dplyr)       # Data manipulation and wrangling
library(gt)          # For creating elegant and customizable tables
library(caret)       # For easy machine learning workflow
library(glmnet)      # For computing penalized regression
library(rpart)       # Decision Tree
library(rpart.plot)
library(caret)       # Evaluate the model
library(randomForest)
```

The MASS package provides the training dataset Pima.tr, which includes a randomly selected subset of 200 subjects, and the test dataset Pima.te, containing the remaining 332 subjects.

```
train.data <- Pima.tr

test.data <- Pima.te

dim(train.data)
```

```
## [1] 200   8
```

```
head(train.data, n = 5)
```

```
##   npreg glu bp skin  bmi   ped age type
## 1     5  86 68   28 30.2 0.364  24   No
## 2     7 195 70   33 25.1 0.163  55  Yes
## 3     5  77 82   41 35.8 0.156  35   No
## 4     0 165 76   43 47.9 0.259  26   No
## 5     0 107 60   25 26.4 0.133  23   No
```

The dataset includes seven predictor variables related to health measurements which are described as follows:

Missingness of Variables

variable	n_miss	pct_miss
npreg	0	0
glu	0	0
bp	0	0
skin	0	0
bmi	0	0
ped	0	0
age	0	0
type	0	0

- `npreg`: Number of pregnancies
- `glu`: Plasma glucose concentration in an oral glucose tolerance test
- `bp`: Diastolic blood pressure (mm Hg)
- `skin`: Triceps skin fold thickness (mm)
- `bmi`: Body mass index (weight in kg / height in m^2)
- `ped`: Diabetes pedigree function (a measure of genetic risk)
- `age`: Age in years
- `type`: Diabetes status (Yes = diabetic, No = non-diabetic), based on the diagnostic criteria established by the World Health Organization (WHO)

The goal of this analysis is to apply statistical learning methods to build predictive models that can classify new patients into diabetic or non-diabetic categories.

We first explore whether any of these variables contain missing values.

```
miss_var_summary(train.data)%>%
  gt() %>%
  gt_theme_guardian() %>%
  tab_header("Missingness of Variables")
```

According to the table, none of the variables contain missing values. Therefore, no imputation is required at this stage.

2. Univariate Analysis

To gain an initial understanding of the dataset, we analyze each variable separately, focusing on its distribution and key summary statistics.

Number of Pregnancies

```
preg_df <- tabyl(train.data$npreg) %>%
  rename(Number_of_Pregnancy = 'train.data$npreg', Count = 'n') %>%
  mutate(Percentage = paste0(round(percent * 100, 1), "%"))%>%
```

Frequnecy of Pregnancies

Number_of_Pregnancy	Count	Percentage
0	28	14%
1	45	22.5%
2	30	15%
3	19	9.5%
4	16	8%
5	11	5.5%
6	10	5%
7	12	6%
8	9	4.5%
9	7	3.5%
10	3	1.5%
11	1	0.5%
12	6	3%
13	1	0.5%
14	2	1%

```
  select(-percent)

preg_tbl <- preg_df %>%
  gt() %>%
  gt_theme_guardian()%>%
  tab_header("Frequnecy of Pregnancies")

preg_tbl
```

The table presents the distribution of the number of pregnancies among the study participants. The variable Number of Pregnancy ranges from 0 to 14.

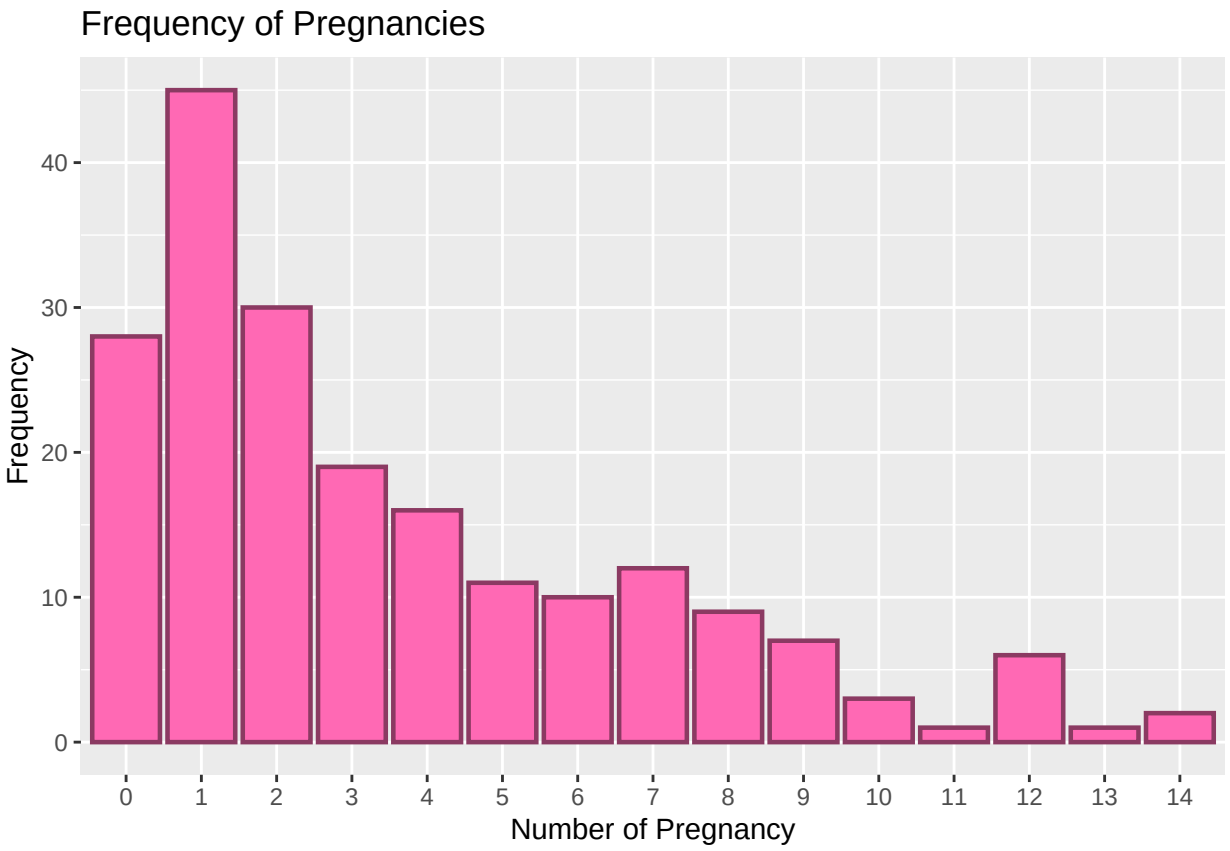
The most frequent category is 1 pregnancy, reported by 45 individuals (22.5%), followed by 2 pregnancies (15%) and 0 pregnancies (14%). As the number of pregnancies increases beyond 2, the frequencies decrease gradually.

This indicates a right-skewed distribution, where lower counts of pregnancies are more common in the sample. This skewness can be observed in the following bar chart:

```
ggplot(preg_df, aes(x = factor(Number_of_Pregnancy), y = Count))+
  geom_bar(stat = "identity", color = "hotpink4", fill = "hotpink",
           linewidth = 0.8)+
  labs(x = "Number of Pregnancy", y = "Frequency",
       title = "Frequency of Pregnancies")
```

Summary Statistics of Plasma Glocose

Statistic	Value
Min.	56.00
1st Qu.	100.00
Median	120.50
Mean	123.97
3rd Qu.	144.00
Max.	199.00



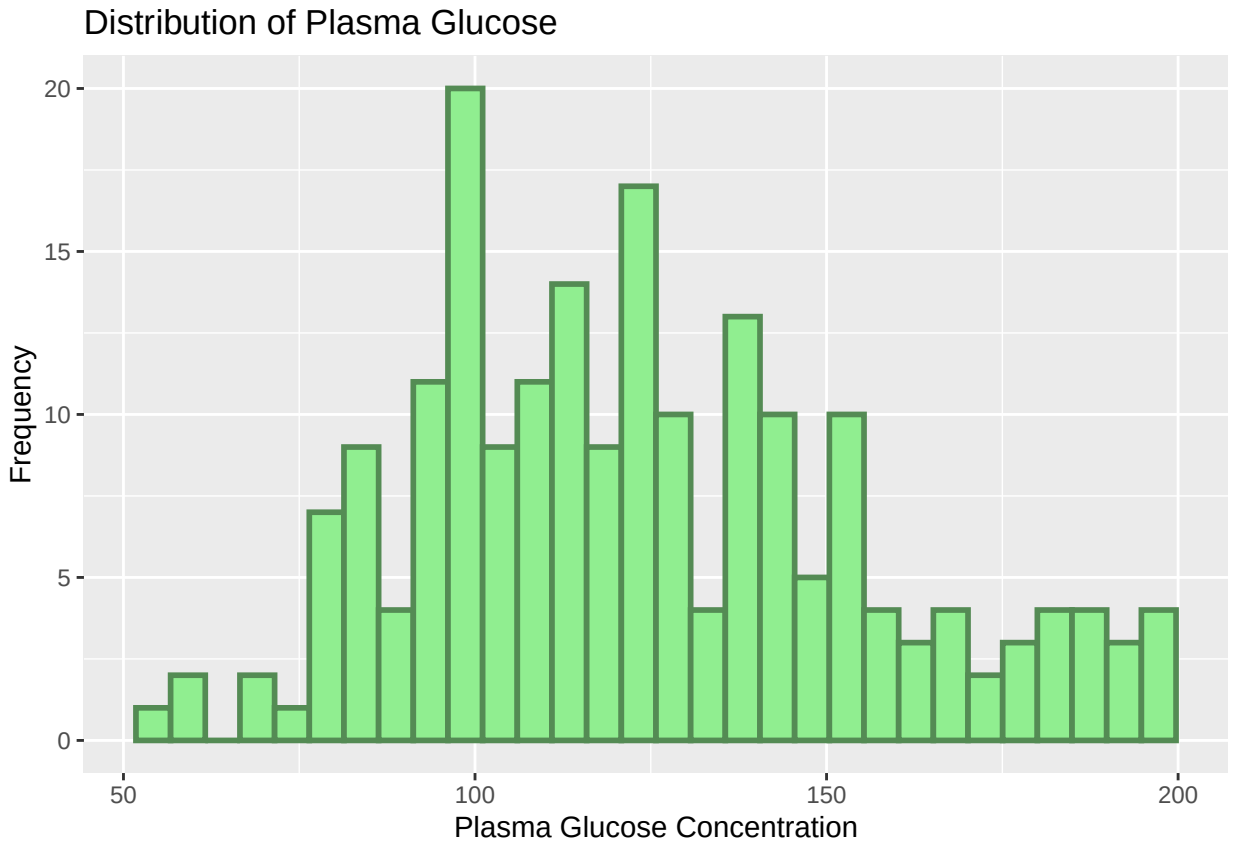
Plasma glucose concentration

```
summary(train.data$glu) %>%
  enframe(name = "Statistic", value = "Value") %>%
  gt() %>%
  gt_theme_guardian() %>%
  tab_header(title = "Summary Statistics of Plasma Glocose")
```

The table presents summary statistics for plasma glucose concentration among participants. The values range from 56 to 199, with a mean of 123.97 and a median of 120.5. The first and third quartiles are 100 and 144, respectively.

```
ggplot(train.data, aes(x = glu))+
  geom_histogram(color = "palegreen4", fill = "palegreen2",
```

```
                  linewidth = 1, bins = 30)+
   labs(x = "Plasma Glucose Concentration",
        y = "Frequency",
        title = "Distribution of Plasma Glucose")
```



The histogram shows that observations are concentrated between 90 and 150 with a gradual decrease in frequency for higher values.

Diastolic Blood Pressure (mm Hg)

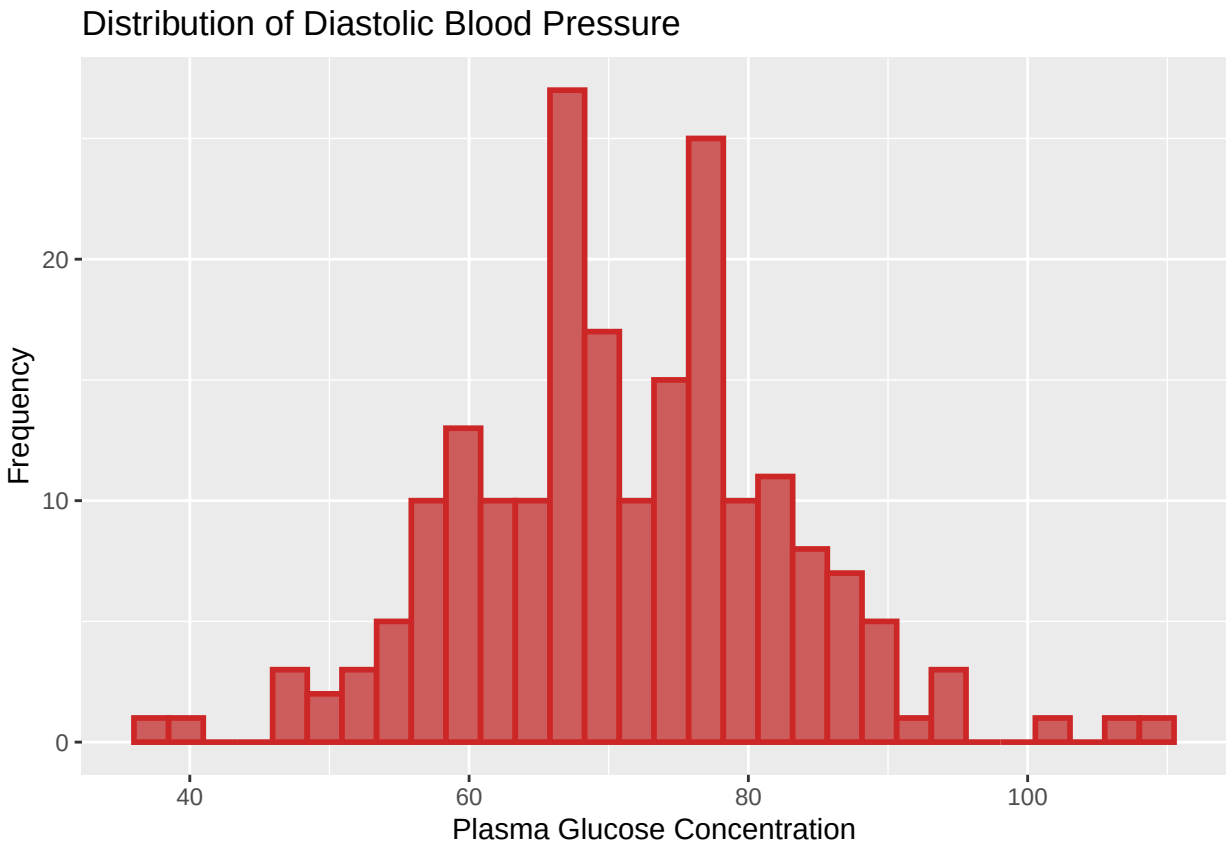
```
summary(train.data$bp) %>%
  enframe(name = "Statistic", value = "Value") %>%
  gt() %>%
  gt_theme_guardian() %>%
  tab_header(title = "Summary Statistics of Diastolic Blood Pressure")
```

The table demonstrates summary statistics for Diastolic Blood Pressure. Among the patients the lowest blood pressure is 38 while the highest is 110. The mean is approximately 71.3 mmHg, and the median is 70 mmHg, suggesting a roughly symmetric distribution.

```
ggplot(train.data, aes(x = bp))+
   geom_histogram(color = "firebrick3", fill = "indianred",
                  linewidth = 1, bins = 30)+
   labs(x = "Plasma Glucose Concentration",
        y = "Frequency",
        title = "Distribution of Diastolic Blood Pressure")
```

Summary Statistics of Diastolic Blood Pressure

Statistic	Value
Min.	38.00
1st Qu.	64.00
Median	70.00
Mean	71.26
3rd Qu.	78.00
Max.	110.00



The histogram shows that most of the values of blood pressure ranges from 60 to 80. A few potential outliers are observed on both ends, particularly below 50 mmHg and above 100 mmHg, which may indicate abnormal blood pressure levels in some individuals.

Triceps Skin Fold Thickness (mm)

```
summary(train.data$skin) %>%
  enframe(name = "Statistic", value = "Value") %>%
  gt() %>%
  gt_theme_guardian() %>%
  tab_header(title = "Summary Statistics of Triceps Skin Fold Thickness")
```

From the table above it can be seen that the minimum value of skin fold thickness is 7 mm. The interquartile range (IQR) spans from 20.75 to 36, capturing the middle 50% of the data. However the maximum value of

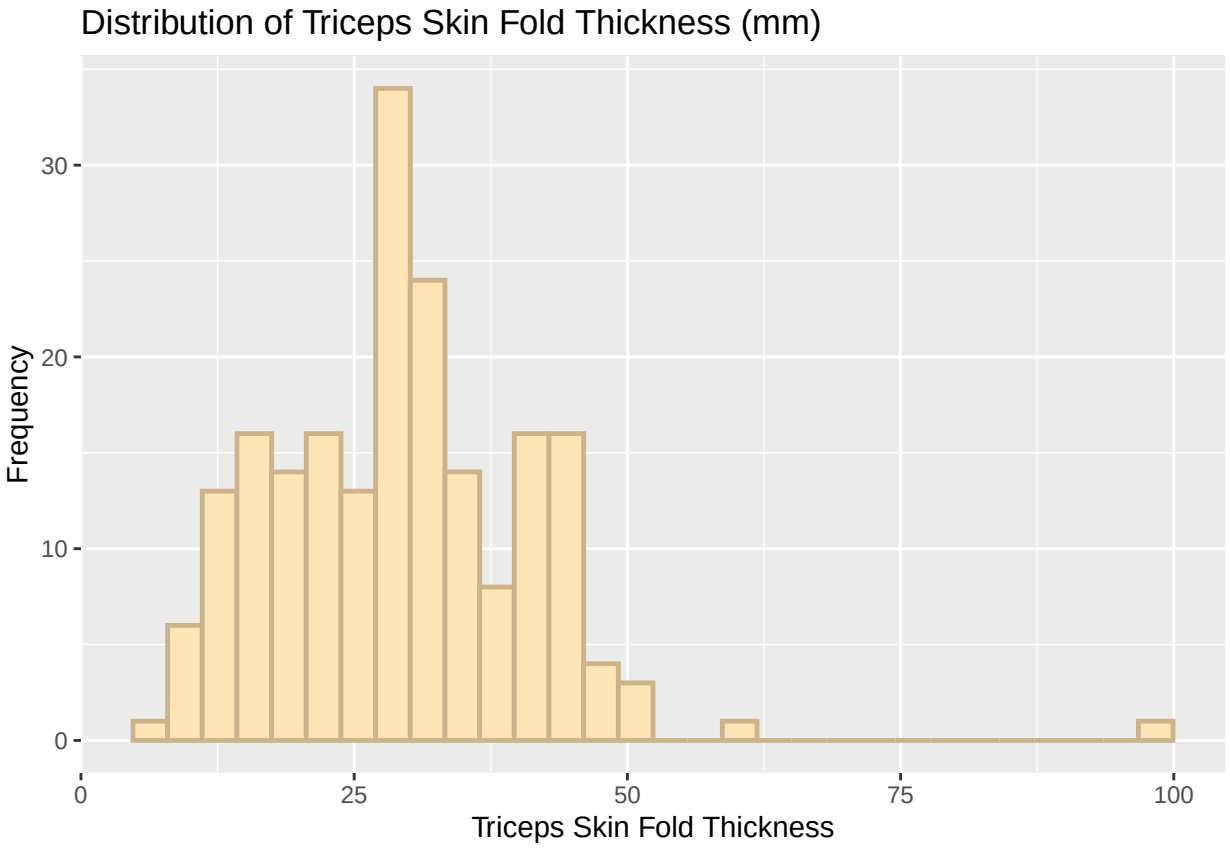
Summary Statistics of Triceps Skin Fold Thickness

Statistic	Value
Min.	7.000
1st Qu.	20.750
Median	29.000
Mean	29.215
3rd Qu.	36.000
Max.	99.000

99 mm is significantly higher than the third quartile and may represent a potential outlier. The histogram below shows a right-skewed distribution, indicating that most individuals have triceps skin fold thickness between 20 and 40 mm, while higher values are relatively rare.

```
ggplot(train.data, aes(x = skin))+
  geom_histogram(fill = "moccasin", color= "navajowhite3",
                 linewidth = 0.85)+
  labs(x = "Triceps Skin Fold Thickness",
       y = "Frequency",
       title = "Distribution of Triceps Skin Fold Thickness (mm)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Body Mass Index

Summary Statistics of BMI

Statistic	Value
Min.	18.200
1st Qu.	27.575
Median	32.800
Mean	32.310
3rd Qu.	36.500
Max.	47.900

```
  summary(train.data$bmi) %>%
    enframe(name = "Statistic", value = "Value") %>%
    gt() %>%
    gt_theme_guardian() %>%
    tab_header(title = "Summary Statistics of BMI")
```

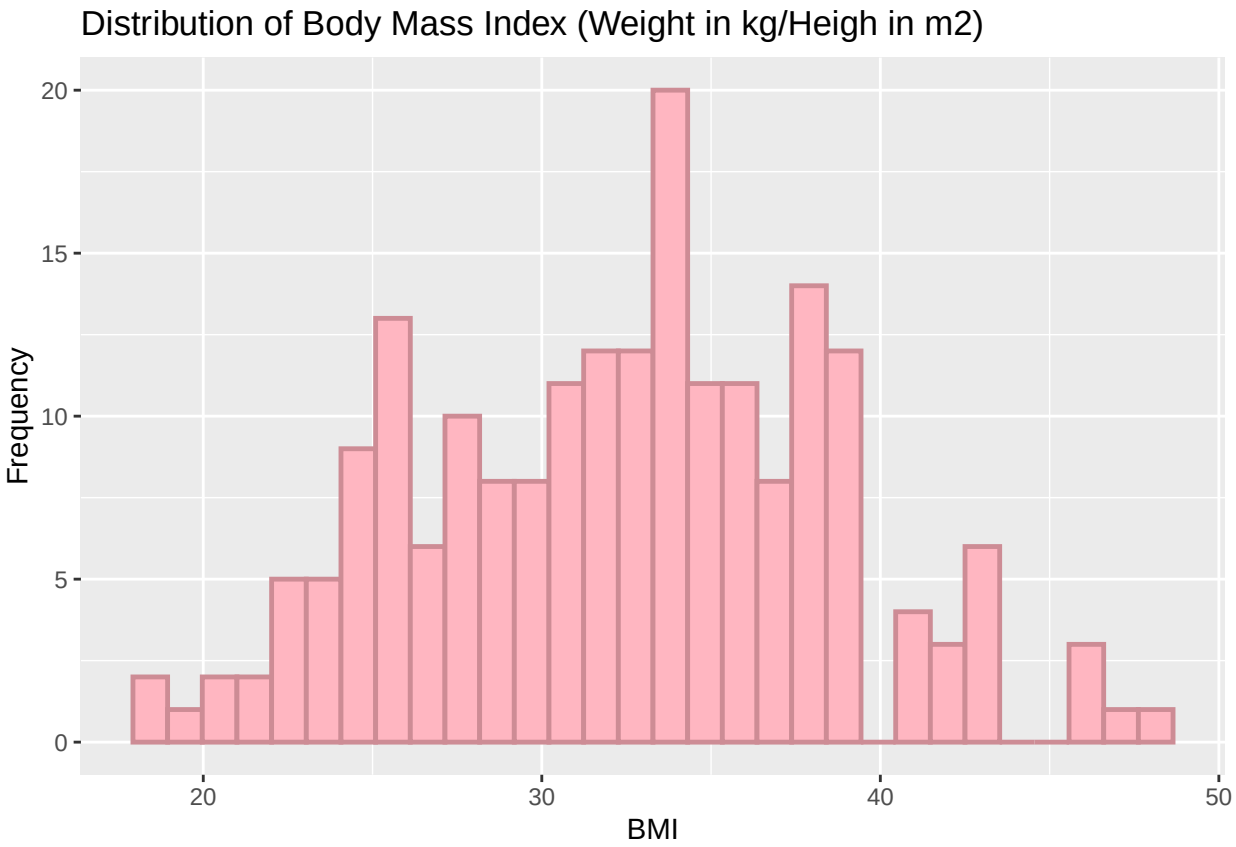
The BMI variable ranges from 18.2 to 47.9, with a mean of 32.31 and a median of 32.8. The interquartile range (IQR) extends from 27.575 to 36.5, indicating that the middle 50% of the observations fall within this interval.

The distribution is slightly right-skewed, with most values concentrated between 25 and 40.

```
ggplot(train.data, aes(x = bmi))+
  geom_histogram(color = "lightpink3", fill = "lightpink",
                 linewidth = 0.85, bins = 30)+
  labs(x = "BMI", y = "Frequency",
       title = "Distribution of Body Mass Index (Weight in kg/Heigh in m2)")
```

Summary Statistics of Diabetes Pedigree Function

Statistic	Value
Min.	0.085000
1st Qu.	0.253500
Median	0.372500
Mean	0.460765
3rd Qu.	0.616000
Max.	2.288000



The histogram shows a clear mode around the mid-30s, and a gradual decline in frequency for values above 40.

There are relatively few observations below 25 or above 45, suggesting the presence of some low- and high-end outliers, though their impact appears minimal.

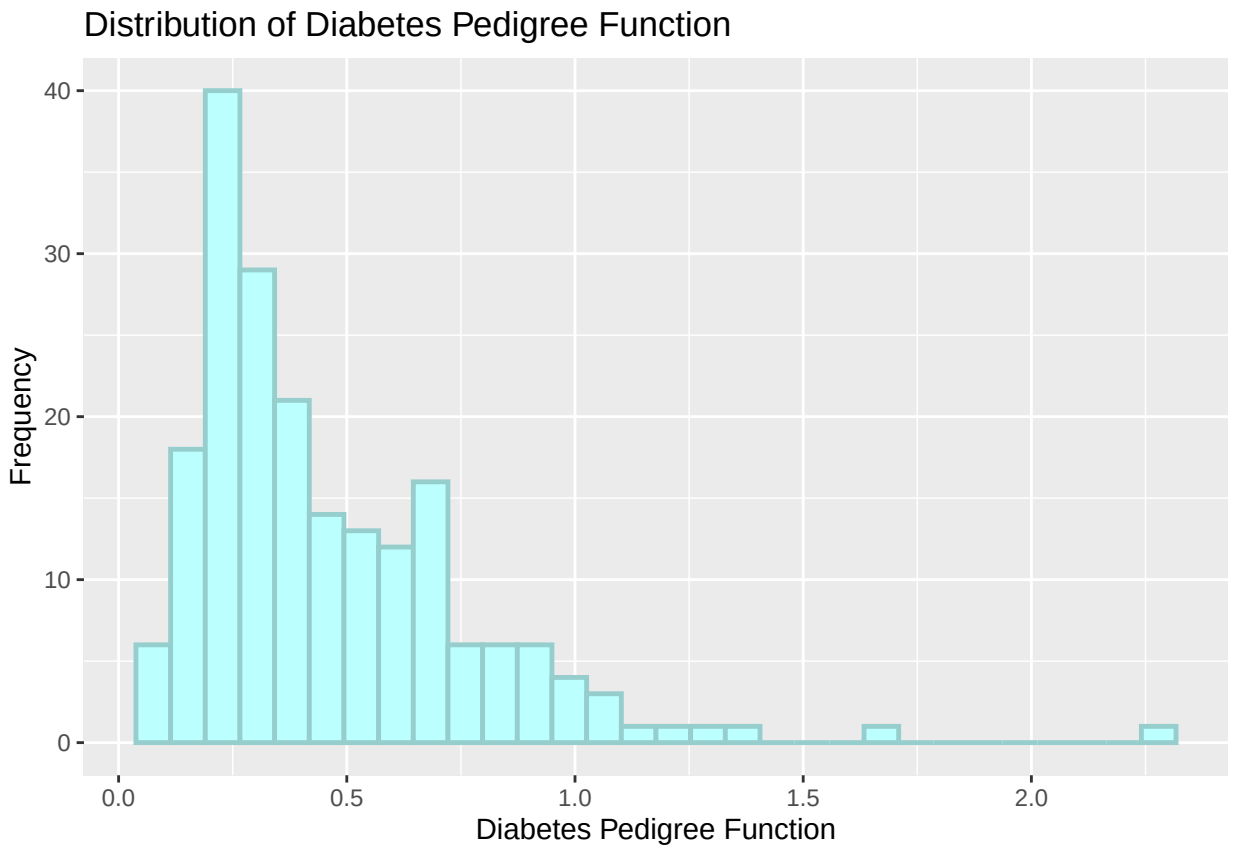
Diabetes Pedigree Function

```
summary(train.data$ped) %>%
  enframe(name = "Statistic", value = "Value") %>%
  gt() %>%
  gt_theme_guardian() %>%
  tab_header(title = "Summary Statistics of Diabetes Pedigree Function")
```

The summary statistics show that the Diabetes Pedigree Function ranges from 0.085 to 2.288, with a median

of 0.373 and a mean of 0.461. The interquartile range (0.254 to 0.616) suggests that most individuals fall within this interval.

```
ggplot(train.data, aes(x = ped))+
  geom_histogram(color = "paleturquoise3", fill = "paleturquoise1",
                 linewidth = 0.85, bins = 30)+
  labs(title = "Distribution of Diabetes Pedigree Function",
       x = "Diabetes Pedigree Function",
       y = "Frequency")
```



The histogram reveals a strong right-skewed distribution, indicating that higher values are rare and possibly outliers.

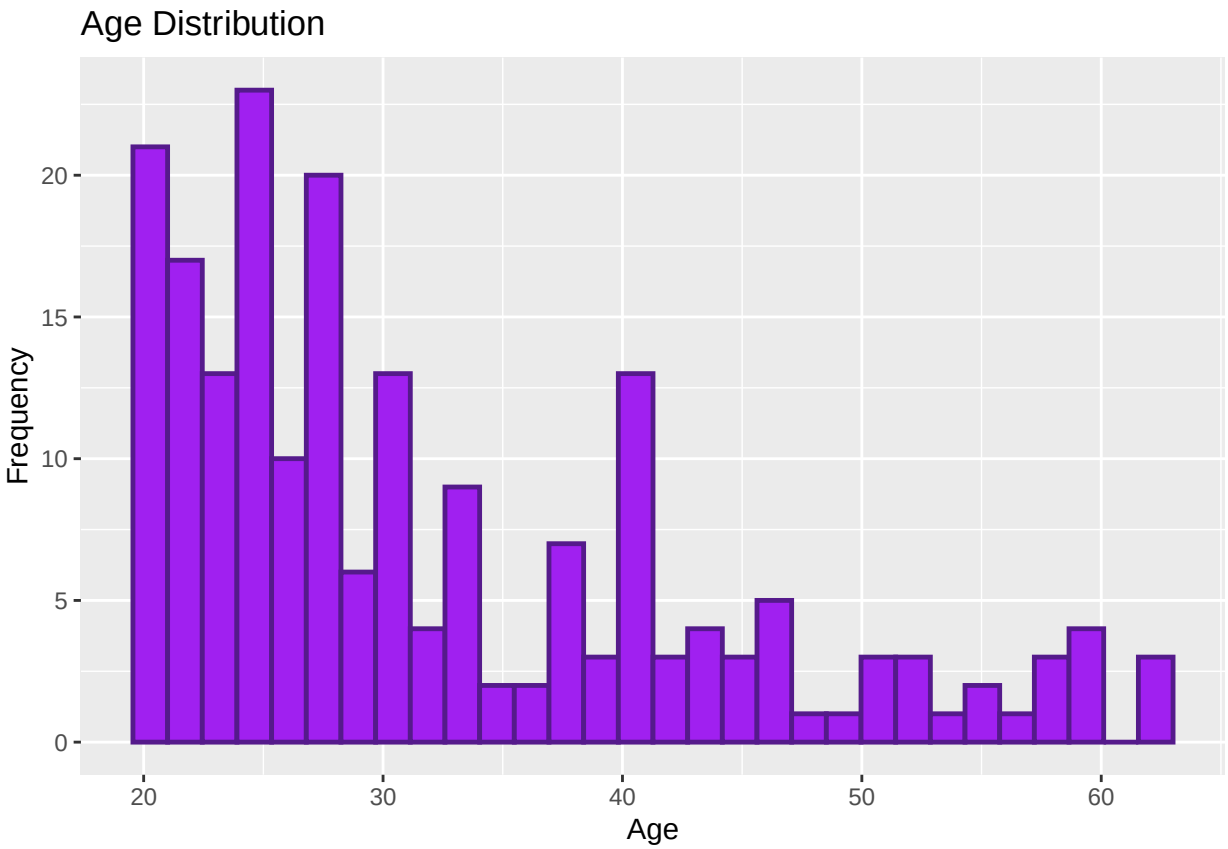
Age

```
  summary(train.data$age) %>%
    enframe(name = "Statistic", value = "Value") %>%
    gt() %>%
    gt_theme_guardian() %>%
    tab_header(title = "Summary Statistics of Age")
```

```
ggplot(train.data, aes(x = age))+
  geom_histogram(color = "purple4", fill = "purple",
                 linewidth = 0.85, bins = 30)+
  labs(title = "Age Distribution",
       x = "Age",
       y = "Frequency")
```

Summary Statistics of Age

Statistic	Value
Min.	21.00
1st Qu.	23.00
Median	28.00
Mean	32.11
3rd Qu.	39.25
Max.	63.00



The age of participants ranges from 21 to 63, with a median of 28. The distribution is clearly right-skewed, showing a concentration of younger individuals in the sample.

Diabetes status

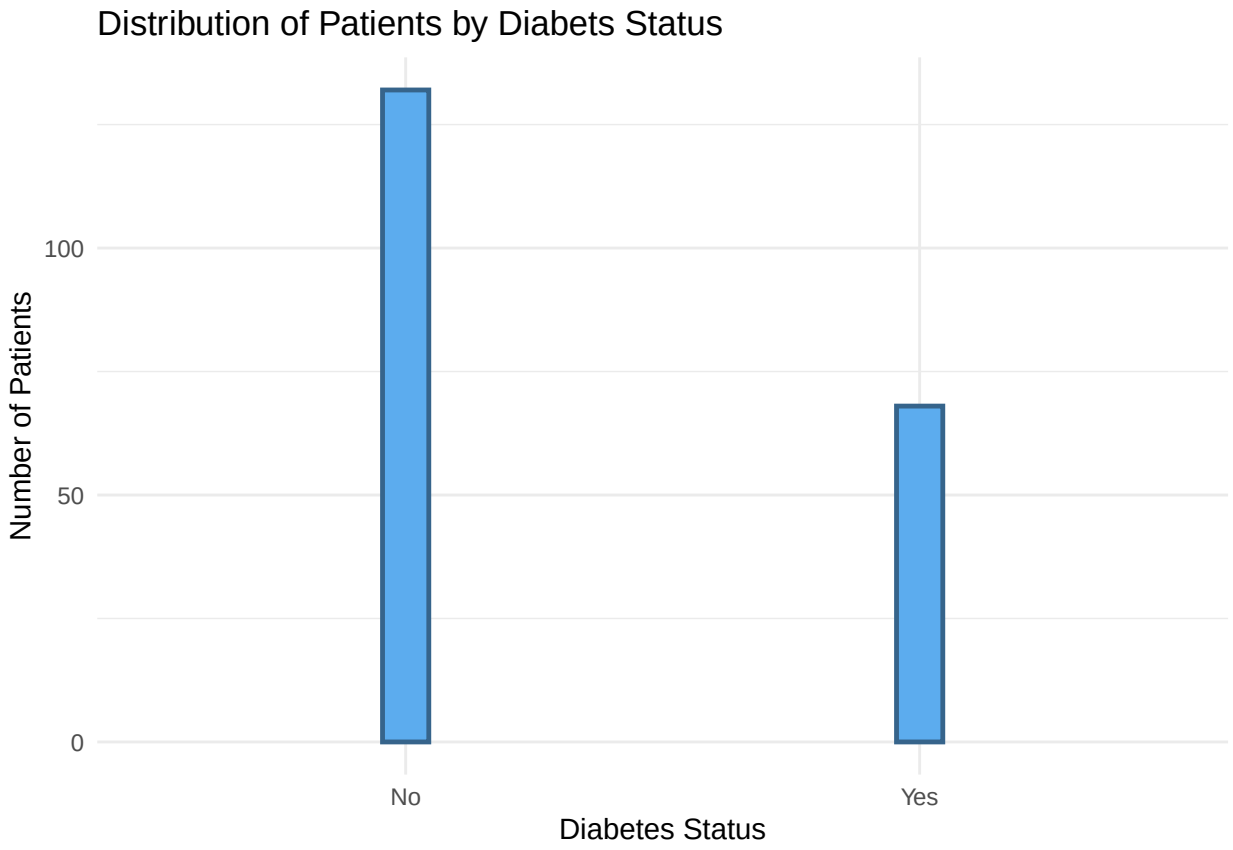
```
type.prob <- train.data$type %>%
  table() %>%
  prop.table() %>%
 as.data.frame()

colnames(type.prob) <- c("Status", "Percent")


type.prob %>%
```

Percentage Distribution of Diabetes Status

Status	Percent
No	0.66
Yes	0.34

```
  gt() %>%
  gt_theme_guardian() %>%
  tab_header(
    title = "Percentage Distribution of Diabetes Status"
  )
```

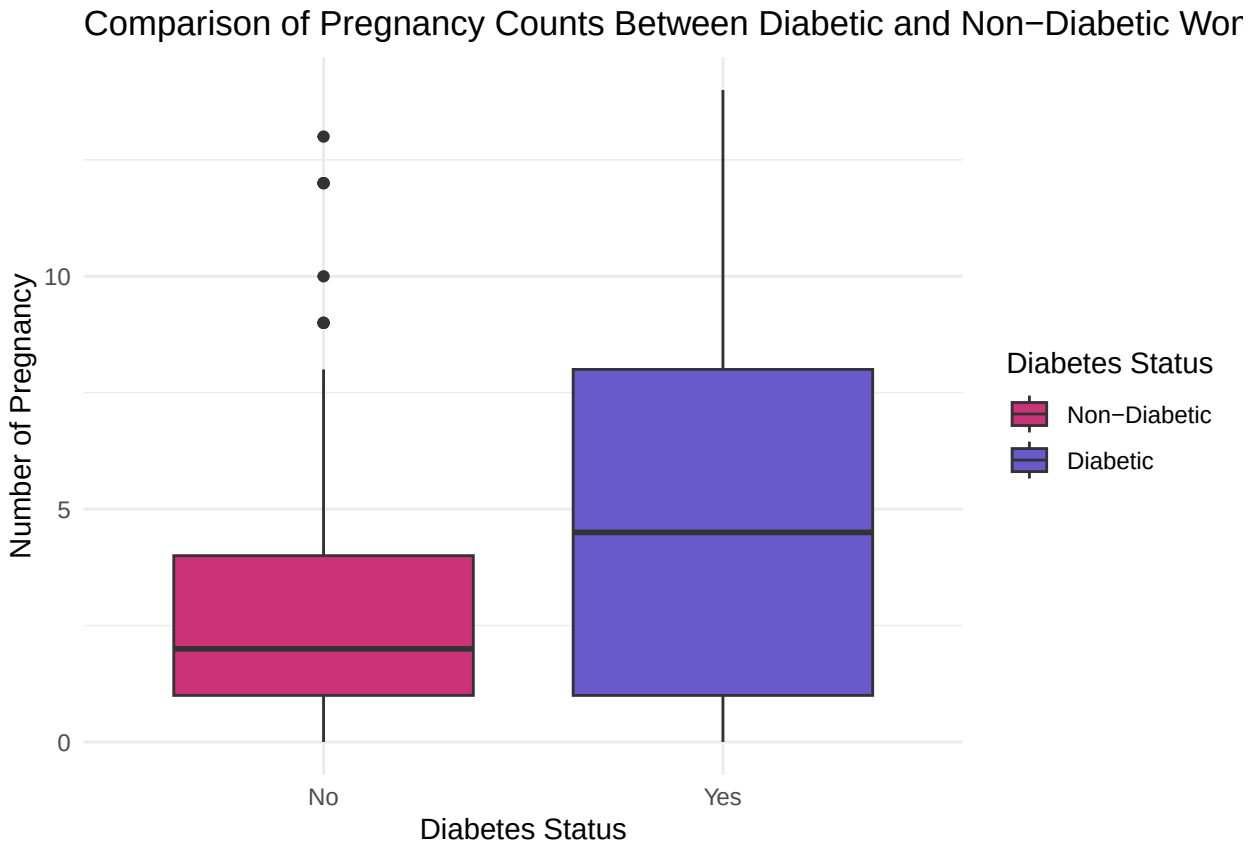
```
type <- train.data$type %>%
  table() %>%
  as.data.frame()

colnames(type) <- c("Status", "Freq")

ggplot(type, aes(x = Status, y = Freq))+
  geom_bar(stat = "identity", colour = "steelblue4", fill = "steelblue2",
           linewidth = 0.85,
           width = 0.09)+
  labs(x = "Diabetes Status",
       y = "Number of Patients",
       title = "Distribution of Patients by Diabets Status")+
  theme_minimal()
```



3. Bivariate Analysis

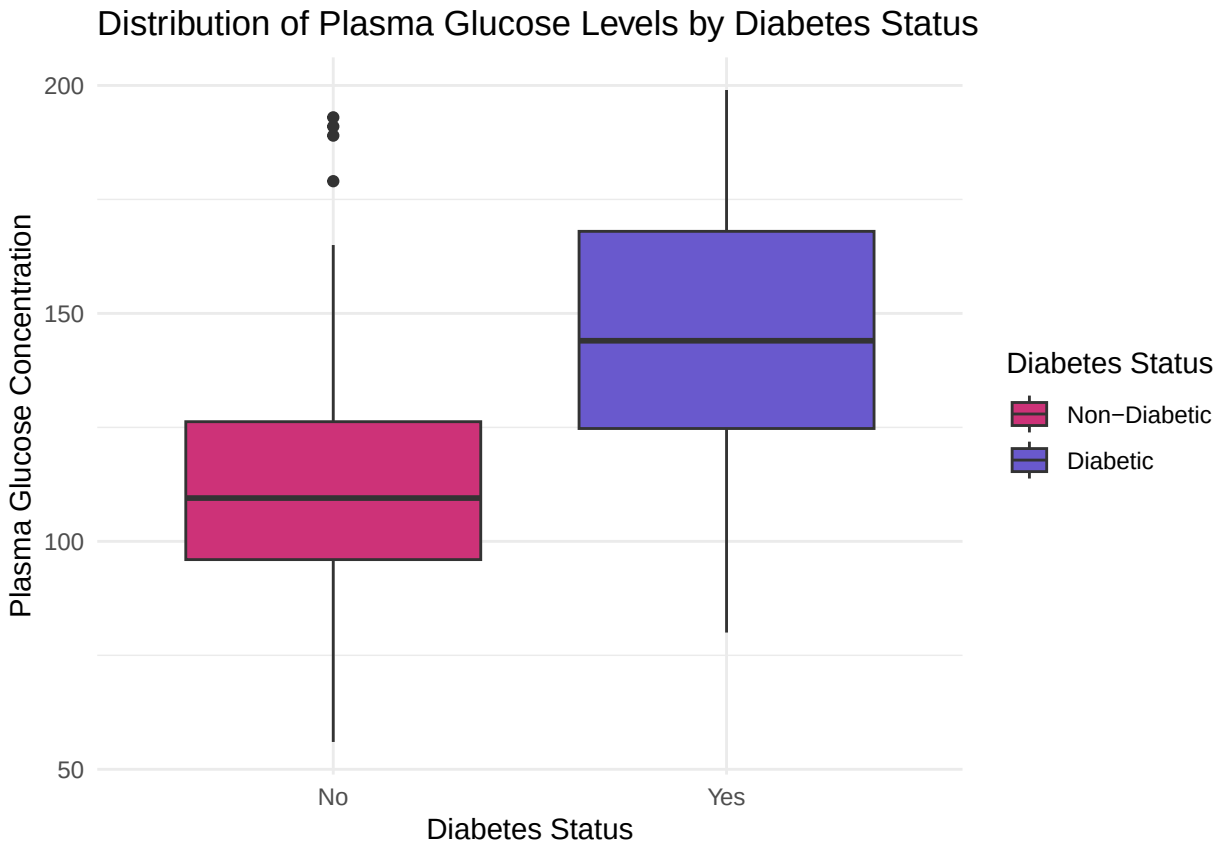
After the univariate analysis, we proceed with the bivariate analysis to explore the relationship between each variable and the response. For this purpose, boxplots were used to visualize how the distribution of the response variable varies across the categories or values of each explanatory variable.

```
ggplot(data = train.data, aes(x = type, y = npreg, fill = type))+
  geom_boxplot()+
  scale_fill_manual(
    name = "Diabetes Status",
    values = c("Yes" = "slateblue3", "No" = "violetred3"),
    labels = c("Yes" = "Diabetic", "No" = "Non-Diabetic"))+
  labs(title = "Comparison of Pregnancy Counts Between Diabetic and Non-Diabetic Women",
       x = "Diabetes Status",
       y = "Number of Pregnancy")+
  theme_minimal()
```



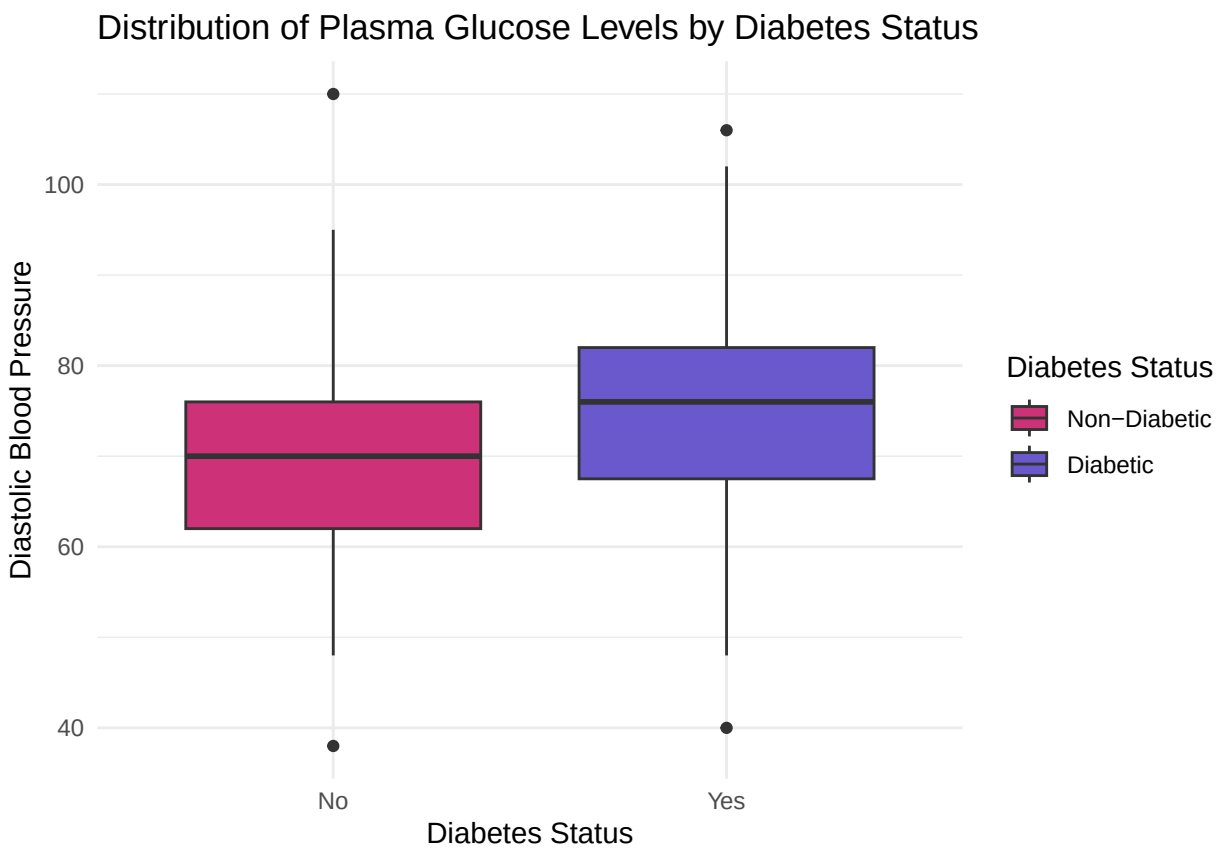
The first boxplot displays the distribution of the number of pregnancies between diabetic and non-diabetic women. It suggests that diabetic women tend to have a higher number of pregnancies compared to non-diabetic women, with a wider spread in the distribution.

```
ggplot(data = train.data, aes(x = type, y = glu, fill = type))+
  geom_boxplot()+
  scale_fill_manual(name = "Diabetes Status",
                    values = c("Yes" = "slateblue3", "No" = "violetred3"),
                    labels = c("Yes" = "Diabetic", "No" = "Non-Diabetic"))+
  labs(title = "Distribution of Plasma Glucose Levels by Diabetes Status",
       x = "Diabetes Status",
       y = "Plasma Glucose Concentration")+
  theme_minimal()
```



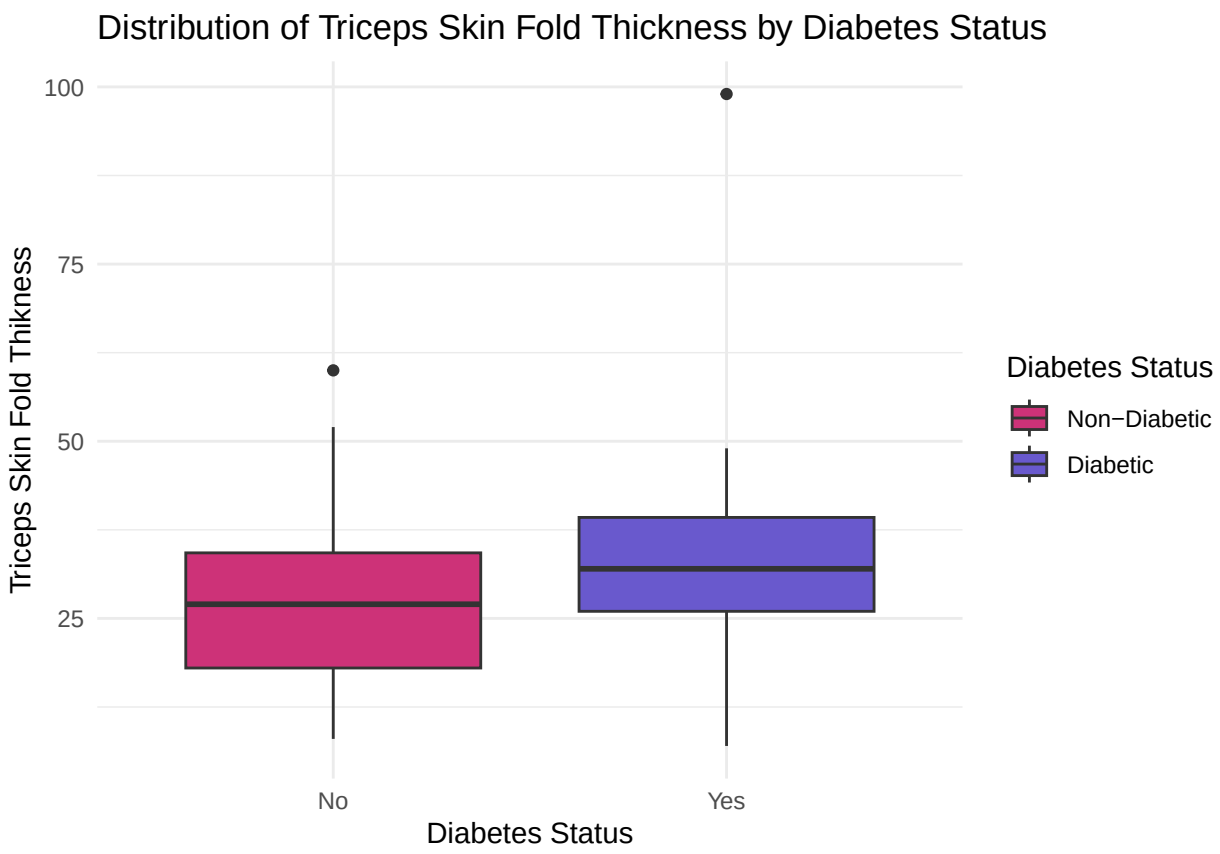
In addition, the distribution of plasma glucose concentration was examined by diabetes status. The boxplot shows that diabetic women generally present higher plasma glucose levels compared to non-diabetic women, with the median level clearly shifted upward.

```
ggplot(data = train.data, aes(x = type, y = bp, fill = type))+
  geom_boxplot()+
  scale_fill_manual(name = "Diabetes Status",
                    values = c("Yes" = "slateblue3", "No" = "violetred3"),
                    labels = c("Yes" = "Diabetic", "No" = "Non-Diabetic"))+
  labs(title = "Distribution of Plasma Glucose Levels by Diabetes Status",
       x = "Diabetes Status",
       y = "Diastolic Blood Pressure")+
  theme_minimal()
```



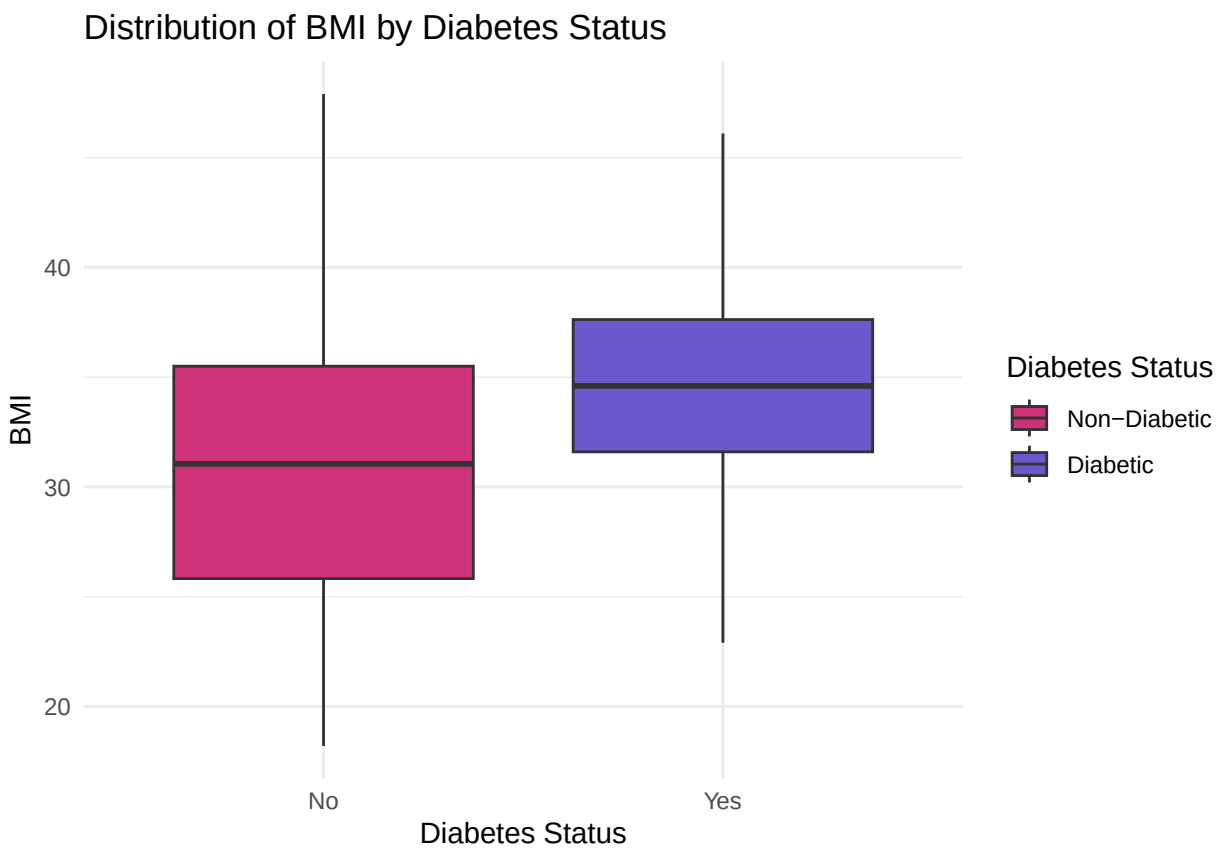
Furthermore, the distribution of diastolic blood pressure was examined by diabetes status. The boxplot suggests that diabetic women tend to have slightly higher values compared to non-diabetic women, although the difference is less pronounced than for pregnancies or plasma glucose levels.

```
ggplot(data = train.data, aes(x = type, y = skin, fill = type))+
  geom_boxplot()+
  scale_fill_manual(name = "Diabetes Status",
                    values = c("Yes" = "slateblue3", "No" = "violetred3"),
                    labels = c("Yes" = "Diabetic", "No" = "Non-Diabetic"))+
  labs(title = "Distribution of Triceps Skin Fold Thickness by Diabetes Status",
       x = "Diabetes Status",
       y = "Triceps Skin Fold Thikness")+
  theme_minimal()
```



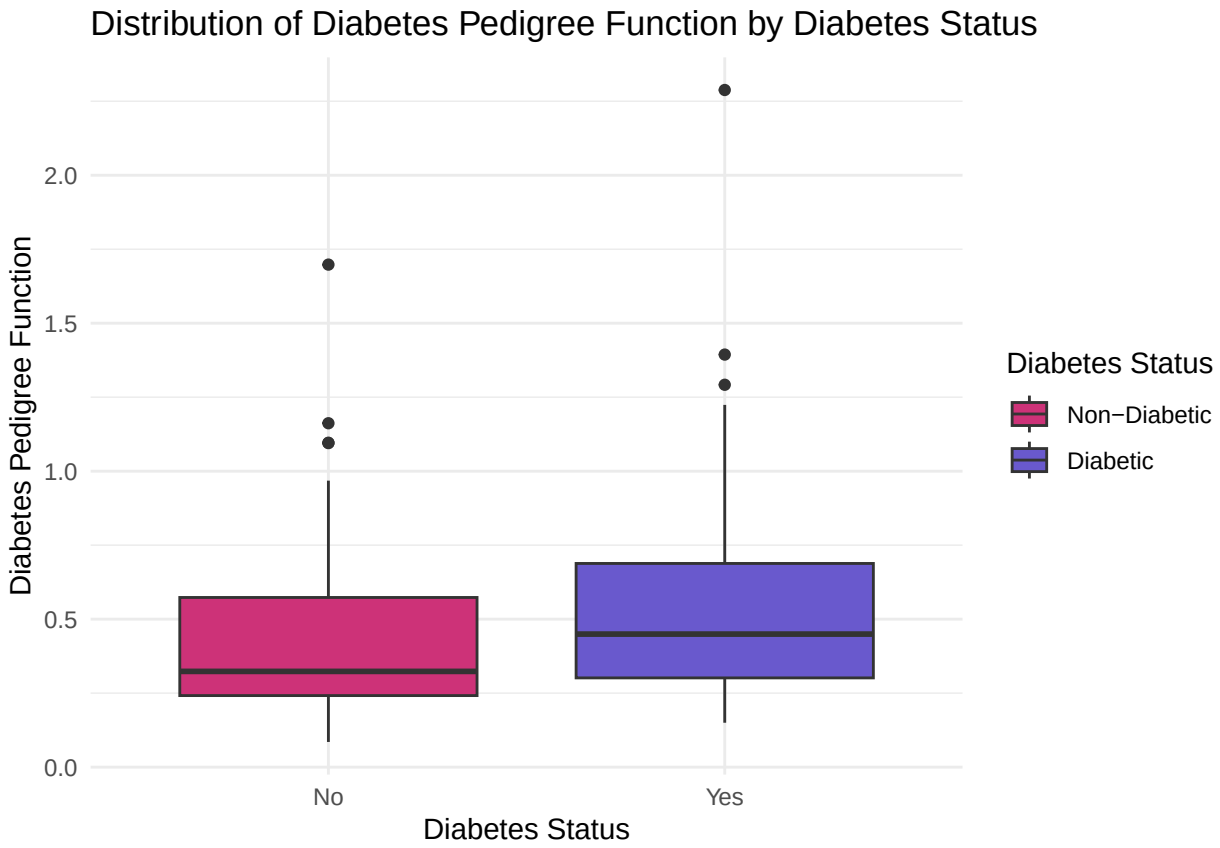
Moreover, the distribution of triceps skin fold thickness was analyzed according to diabetes status. The boxplot reveals that diabetic women generally show slightly higher skin fold thickness compared to non-diabetic women, although the overlap between the two groups remains considerable.

```
ggplot(data = train.data, aes(x = type, y = bmi, fill = type))+
  geom_boxplot()+
  scale_fill_manual(name = "Diabetes Status",
                    values = c("Yes" = "slateblue3", "No" = "violetred3"),
                    labels = c("Yes" = "Diabetic", "No" = "Non-Diabetic"))+
  labs(title = "Distribution of BMI by Diabetes Status",
       x = "Diabetes Status",
       y = "BMI")+
  theme_minimal()
```



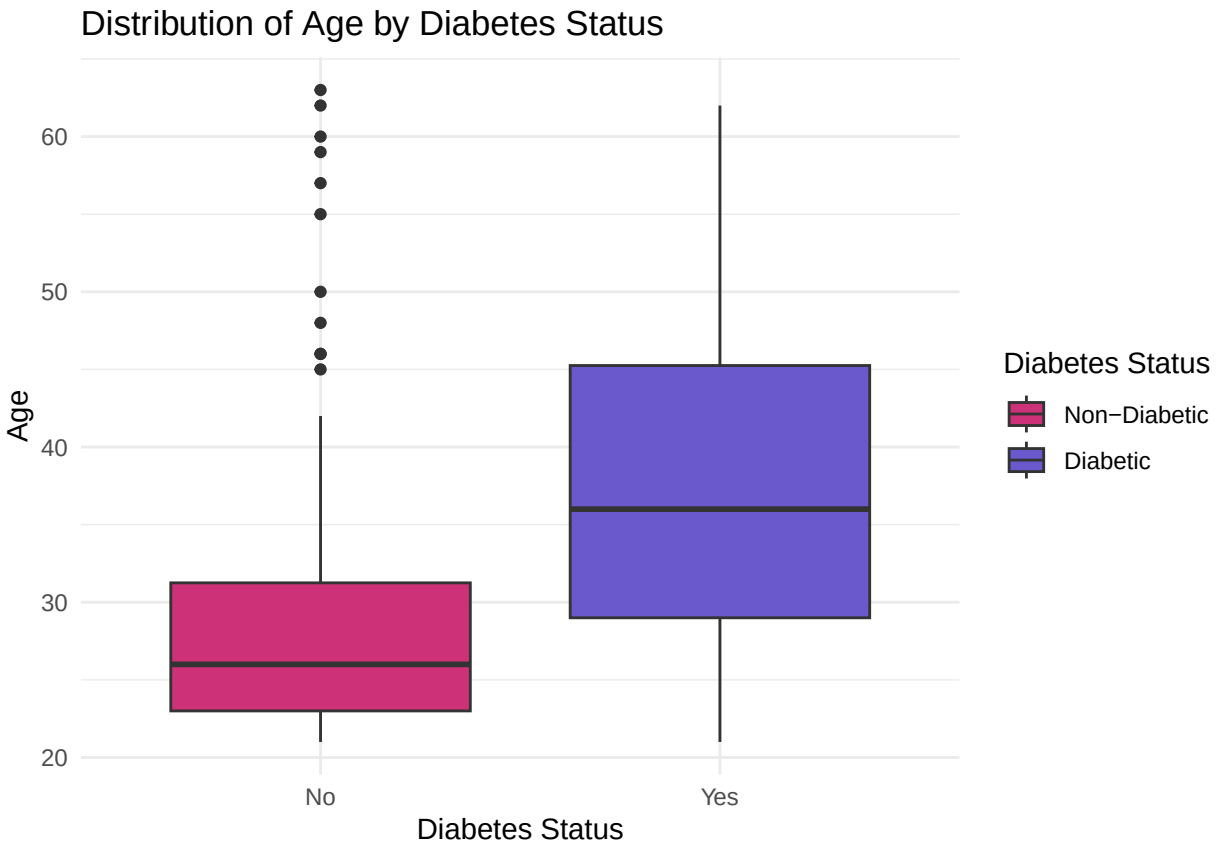
Similarly, the distribution of body mass index (BMI) was compared between diabetic and non-diabetic women. The boxplot shows that diabetic women generally have higher BMI values, with the median clearly shifted upward relative to the non-diabetic group.

```
ggplot(data = train.data, aes(x = type, y = ped, fill = type))+
  geom_boxplot()+
  scale_fill_manual(name = "Diabetes Status",
                    values = c("Yes" = "slateblue3", "No" = "violetred3"),
                    labels = c("Yes" = "Diabetic", "No" = "Non-Diabetic"))+
  labs(title = "Distribution of Diabetes Pedigree Function by Diabetes Status",
       x = "Diabetes Status",
       y = "Diabetes Pedigree Function")+
  theme_minimal()
```



In addition, the diabetes pedigree function was assessed across diabetes status. The boxplot indicates that diabetic women tend to have slightly higher values compared to non-diabetic women, although the two groups show a large degree of overlap.

```
ggplot(data = train.data, aes(x = type, y = age, fill = type))+
  geom_boxplot()+
  scale_fill_manual(name = "Diabetes Status",
                    values = c("Yes" = "slateblue3", "No" = "violetred3"),
                    labels = c("Yes" = "Diabetic", "No" = "Non-Diabetic"))+
  labs(title = "Distribution of Age by Diabetes Status",
       x = "Diabetes Status",
       y = "Age")+
  theme_minimal()
```



Finally, the distribution of age was examined by diabetes status. The boxplot shows that diabetic women tend to be older than non-diabetic women, with a noticeably higher median and a wider spread in the distribution.

4. Classification Models

After exploring the data, we now move to predictive modeling in order to classify diabetes status. Several models were trained on the training set and evaluated on the test set, including logistic regression, regularized regression methods (Lasso, Ridge, Elastic Net), and Decision Tree model. The goal is to assess the predictive performance of these approaches and compare their effectiveness.

4.1 Logistic Regression

Logistic regression is a baseline classification model that estimates the probability of diabetes status as a function of the explanatory variables.

```
logreg <- glm(type~ ., data = train.data, family = "binomial")
summary(logreg)
```

```
##
## Call:
## glm(formula = type ~ ., family = "binomial", data = train.data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.773062   1.770386  -5.520 3.38e-08 ***
## npreg        0.103183   0.064694   1.595  0.11073
## glu          0.032117   0.006787   4.732 2.22e-06 ***
```

```
## bp          -0.004768   0.018541  -0.257  0.79707
## skin        -0.001917   0.022500  -0.085  0.93211
## bmi          0.083624   0.042827   1.953  0.05087 .
## ped          1.820410   0.665514   2.735  0.00623 **
## age          0.041184   0.022091   1.864  0.06228 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 178.39  on 192  degrees of freedom
## AIC: 194.39
##
## Number of Fisher Scoring iterations: 5
```

Multivariable logistic regression to predict diabetes (Yes vs No) showed: higher glucose was strongly associated with diabetes (beta = 0.032; p < 0.001), BMI showed a borderline association (beta = 0.084; p = 0.05), the diabetes pedigree function (ped) was strongly associated (beta = 1.820; p = 0.006), and age was borderline (beta = 0.041; p = 0.062). Number of pregnancies, blood pressure, and skinfold thickness were not significant.

```
log_pred <- logreg %>% predict(test.data, type = "response")

log_class <- ifelse(log_pred > 0.5, "Yes", "No")

confusionMatrix(table(predictions = factor(log_class), Actual = test.data$type), positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##            Actual
## predictions  No Yes
##         No  200  43
##         Yes  23  66
##
##                Accuracy : 0.8012
##                  95% CI : (0.7542, 0.8428)
##     No Information Rate : 0.6717
##     P-Value [Acc > NIR] : 1.116e-07
##
##                   Kappa : 0.5271
##
##  Mcnemar's Test P-Value : 0.01935
##
##             Sensitivity : 0.6055
##             Specificity : 0.8969
##          Pos Pred Value : 0.7416
##          Neg Pred Value : 0.8230
##              Prevalence : 0.3283
##          Detection Rate : 0.1988
##    Detection Prevalence : 0.2681
##       Balanced Accuracy : 0.7512
##
##        'Positive' Class : Yes
##
```

The logistic regression model achieved an overall accuracy of about 80% (95% CI: 75%–84%). The specificity

was relatively high (0.90), indicating that the model performed well in correctly identifying non-diabetic women. However, the sensitivity was lower (0.60), suggesting that the model was less effective in detecting diabetic cases. The positive predictive value (0.74) and negative predictive value (0.82) further illustrate that predictions for non-diabetic women were more reliable than those for diabetic women. Overall, the logistic regression model provided a reasonable baseline performance, with stronger discrimination for non-diabetic than for diabetic cases.

We will use the R function 'glmnet()' [glmnet package] for computing penalized logistic regression.

The simplified format is as follow:

```
x.train <- model.matrix(type ~ ., train.data)[, -1]
y.train <- ifelse(train.data$type == "Yes", 1, 0)

glmnet(x.train, y.train, family = "binomial", alpha = 1, lambda = NULL)
```

'x.train': matrix of predictor variables

'y.train': the response or outcome variable, which is a binary variable.

'family': the response type. Use "binomial" for a binary outcome variable

'alpha': the elasticnet mixing parameter. Allowed values include:

"1": for lasso regression

"0": for ridge regression

a value between 0 and 1 (say 0.3) for elastic net regression.

lamba: a numeric value defining the amount of shrinkage. Should be specify by analyst.

4.2 Lasso Regression

Lasso regression extends logistic regression by adding an L1 penalty on the coefficients. This regularization shrinks some coefficients toward zero and performs automatic variable selection, which can improve prediction accuracy and interpretability, especially in the presence of many correlated predictors

```
# Finding the best lambda using cross validation
lambda.lasso = cv.glmnet(x.train, y.train, alpha = 1, family = "binomial")

lasso.model <- glmnet(x.train, y.train,
                      alpha = 1,
                      lambda = lambda.lasso$lambda.min,
                      family = "binomial")

coef(lasso.model)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -8.60788281
## npreg        0.08125343
## glu          0.02857505
## bp           .
## skin         .
## bmi          0.06500907
## ped          1.42106304
## age          0.03503423
```

The Lasso regression model performed automatic variable selection by shrinking the coefficients of non-informative predictors (such as blood pressure and skin fold thickness) to zero. The remaining predictors included number of pregnancies, glucose, BMI, pedigree function, and age.

```
x.test <- model.matrix(type~ ., test.data)[, -1]
y.test <- ifelse(test.data$type == "Yes", 1, 0)
probabilities <- lasso.model %>% predict(newx = x.test, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "Yes", "No")
predicted.classes <- factor(predicted.classes, levels = c("No", "Yes"))
actual.classes <- test.data$type
```

```
cm.lasso <- table(predictions = predicted.classes, Actual = actual.classes)
confusionMatrix(cm.lasso, positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##            Actual
## predictions  No Yes
##         No  200  43
##         Yes  23  66
##
##                Accuracy : 0.8012
##                  95% CI : (0.7542, 0.8428)
##     No Information Rate : 0.6717
##     P-Value [Acc > NIR] : 1.116e-07
##
##                   Kappa : 0.5271
##
##  Mcnemar's Test P-Value : 0.01935
##
##             Sensitivity : 0.6055
##             Specificity : 0.8969
##          Pos Pred Value : 0.7416
##          Neg Pred Value : 0.8230
##              Prevalence : 0.3283
##          Detection Rate : 0.1988
##    Detection Prevalence : 0.2681
##       Balanced Accuracy : 0.7512
##
##        'Positive' Class : Yes
##
```

The predictive performance on the test set was very similar to the standard logistic regression, with an accuracy of about 80%, a sensitivity of 0.61, and a specificity of 0.90. This shows that Lasso was able to simplify the model without loss of predictive performance.

4.3 Ridge Regression

Ridge regression applies an L2 penalty on the regression coefficients. Unlike Lasso, Ridge does not set coefficients exactly to zero, but instead shrinks them toward smaller values. This helps reduce overfitting and is particularly useful when predictors are highly correlated, leading to more stable estimates.

```
lambda.ridge <- cv.glmnet(x.train, y.train, alpha = 0, family = binomial)
ridge.model <- glmnet(x.train, y.train, alpha = 0, lambda = lambda.ridge$lambda.min, family = binomial)
coef(ridge.model)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -7.848450204
## npreg        0.081350442
## glu          0.023255674
## bp           0.003222154
## skin         0.006706758
## bmi          0.054497201
## ped          1.257061155
## age          0.032722743
```

```
probabilities.ridge <- ridge.model %>% predict(newx = x.test, type = "response")
predicted.classes.ridge <- ifelse(probabilities.ridge > 0.5, "Yes", "No")
predicted.classes.ridge <- factor(predicted.classes.ridge, levels = c("No", "Yes"))
xtab <- table(predictions = predicted.classes.ridge, Actual = test.data$type)
cm.ridge <- confusionMatrix(xtab, positive = "Yes")
cm.ridge
```

```
## Confusion Matrix and Statistics
##
##            Actual
## predictions  No Yes
##         No  202  47
##         Yes  21  62
##
##                Accuracy : 0.7952
##                  95% CI : (0.7477, 0.8373)
##     No Information Rate : 0.6717
##     P-Value [Acc > NIR] : 4.282e-07
##
##                   Kappa : 0.5055
##
##  Mcnemar's Test P-Value : 0.002432
##
##             Sensitivity : 0.5688
##             Specificity : 0.9058
##          Pos Pred Value : 0.7470
##          Neg Pred Value : 0.8112
##              Prevalence : 0.3283
##          Detection Rate : 0.1867
##    Detection Prevalence : 0.2500
##       Balanced Accuracy : 0.7373
##
##        'Positive' Class : Yes
##
```

The Ridge regression model yielded an overall accuracy of about 79.8%, with a sensitivity of 0.60 and a specificity of 0.90. These results are very similar to those obtained with standard logistic regression and Lasso regression. While Ridge helps address multicollinearity by shrinking coefficients toward smaller values, in this dataset with relatively few predictors, its predictive performance did not differ substantially from the other models.

4.4 Elastic Net Regression

```
alphas = seq(0, 1, 0.1)

results = data.frame()

set.seed(123)
for(a in alphas){
  cv.fit <- cv.glmnet(x.train, y.train, family = "binomial", alpha = a)
  results <- rbind(results,
                   data.frame(alpha = a,
                              lambda = cv.fit$lambda.min,
                              cvm = min(cv.fit$cvm)))
}

results[order(results$cvm)[1],]
```

```
##   alpha     lambda       cvm
## 5   0.4 0.01992527 0.9500162
```

```
elasticnet.model <- glmnet(x.train, y.train, family = "binomial", alpha = 0.4, lambda = 0.1992)

coef(elasticnet.model)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -3.24124705
## npreg        0.01061753
## glu          0.01191287
## bp           .
## skin         .
## bmi          0.01473139
## ped          0.12035919
## age          0.01542057
```

```
pro.elastic <- elasticnet.model %>% predict(newx = x.test,
                                            type = "response")
pred.elastic <- ifelse(pro.elastic > 0.5, "Yes", "No")
pred.elastic <- factor(pred.elastic, levels = c("No", "Yes"))
cm.elastic <- table(Predictions = pred.elastic, Actual = actual.classes)
confusionMatrix(cm.elastic, positive = "Yes")
```

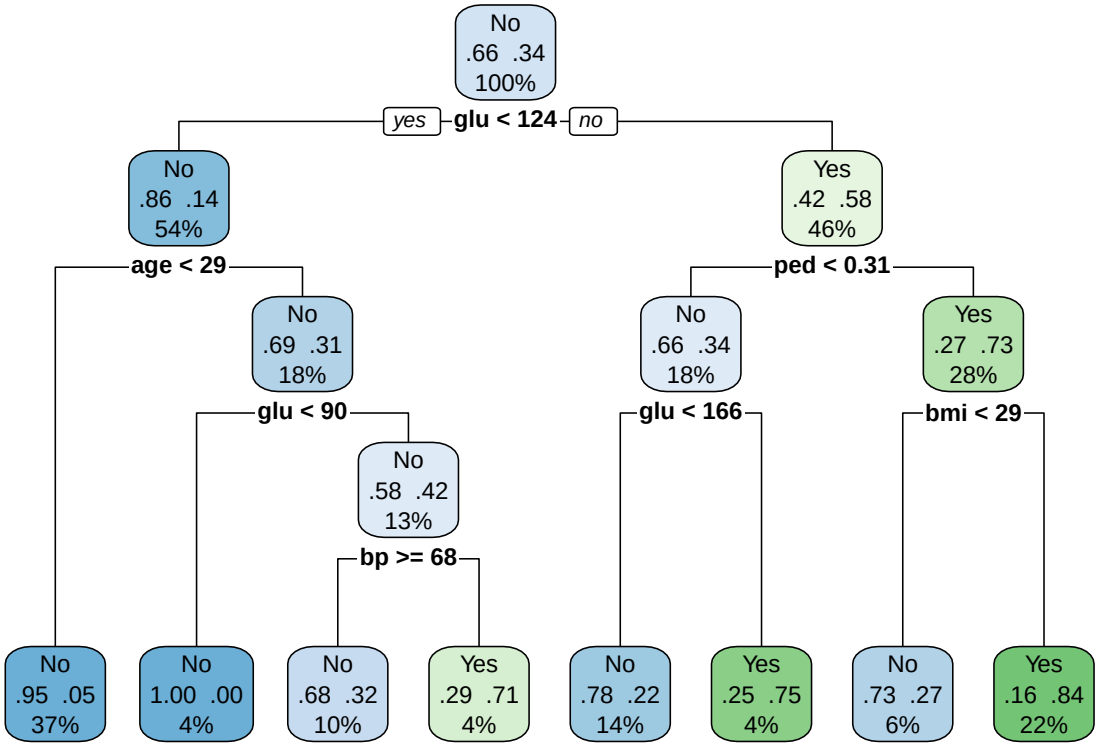
```
## Confusion Matrix and Statistics
##
##            Actual
## Predictions  No Yes
##         No  218  83
##         Yes   5  26
##
##                Accuracy : 0.7349
##                  95% CI : (0.684, 0.7816)
##     No Information Rate : 0.6717
##     P-Value [Acc > NIR] : 0.007503
##
##                   Kappa : 0.2645
##
```

```
##  Mcnemar's Test P-Value : 2.245e-16
##
##             Sensitivity : 0.23853
##             Specificity : 0.97758
##          Pos Pred Value : 0.83871
##          Neg Pred Value : 0.72425
##              Prevalence : 0.32831
##          Detection Rate : 0.07831
##    Detection Prevalence : 0.09337
##       Balanced Accuracy : 0.60806
##
##        'Positive' Class : Yes
##
```

The Elastic Net regression model achieved an accuracy of 73%, with very high specificity (0.98) but very low sensitivity (0.24). This indicates that while the model was highly effective at correctly classifying non-diabetic women, it failed to identify a large proportion of diabetic cases. In other words, the model tended to under-predict the positive class, which may limit its usefulness in clinical applications where sensitivity is crucial.

4.5 Decision Tree

```
pima_tree <- rpart(type ~ ., data = train.data, method = "class")
rpart.plot(pima_tree, type = 2, extra = 104)
```




```
predictions <- predict(pima_tree, test.data, type = "class")
cm.tree <- confusionMatrix(table(predictions = predictions, Actual = test.data$type), positive = "Yes")
cm.tree
```

```
## Confusion Matrix and Statistics
##
```

```
##            Actual
## predictions  No Yes
##         No  182  48
##         Yes  41  61
##
##                Accuracy : 0.7319
##                  95% CI : (0.6808, 0.7788)
##     No Information Rate : 0.6717
##     P-Value [Acc > NIR] : 0.01042
##
##                   Kappa : 0.382
##
##  Mcnemar's Test P-Value : 0.52478
##
##             Sensitivity : 0.5596
##             Specificity : 0.8161
##          Pos Pred Value : 0.5980
##          Neg Pred Value : 0.7913
##              Prevalence : 0.3283
##          Detection Rate : 0.1837
##    Detection Prevalence : 0.3072
##       Balanced Accuracy : 0.6879
##
##        'Positive' Class : Yes
##
```

The decision tree classifier achieved an accuracy of 73%, with a sensitivity of 0.56 and a specificity of 0.82. This indicates that the tree was moderately successful in identifying diabetic cases, although its overall performance was lower than that of logistic regression and its regularized variants. While decision trees provide interpretability through visualization, their predictive accuracy on this dataset was comparatively limited.

5. Conclusion

Overall, logistic regression and its regularized variants (Lasso and Ridge) provided the best balance between sensitivity and specificity, with an accuracy close to 80% and relatively stable performance across metrics. Lasso simplified the model by removing non-informative predictors without compromising predictive power, while Ridge stabilized coefficient estimates but did not substantially improve accuracy. In contrast, Elastic Net exhibited a strong bias toward the negative class, resulting in high specificity but poor sensitivity, which limits its usefulness when correctly identifying diabetic cases is critical. The decision tree offered moderate interpretability but showed lower overall performance compared to logistic-based approaches. Taken together, logistic regression and its penalized extensions appear to be the most appropriate models for this dataset, providing reliable and balanced predictive performance.