Diabetes_analysis

Sherifdeen Abubakr Gbolagade (CST/19/COM/00258)

2025-03-16

About Diabetes Dataset

The Diabetes Diagnosis Dataset consists of 9,538 medical records, capturing key health parameters, lifestyle habits, and genetic predispositions influencing diabetes risk. The dataset includes 17 features, such as blood glucose levels, BMI, cholesterol levels, and hypertension status, providing a comprehensive view of diabetes risk factors. It is structured with realistic distributions, making it valuable for medical research, statistical analysis, and machine learning applications

##Load Data and Libraries

Load necessary libraries:

```r
library(class) # For KNN
library(caret) # For data splitting and evaluation
library(ggplot2)  # For visualization
```

Load the dataset:

```r
data <- read.csv("diabetes_dataset.csv")
str(data)  # Check the structure
```

```
## 'data.frame':    9538 obs. of  17 variables:
##  $ Age              : int  69 32 89 78 38 41 20 39 70 19 ...
##  $ Pregnancies      : int  5 1 13 13 8 10 16 4 3 1 ...
##  $ BMI              : num  28.4 26.5 25.3 29.9 24.6 ...
##  $ Glucose          : num  130 116 101 146 103 ...
##  $ BloodPressure    : num  77 72 82 104 74 71 60 94 90 62 ...
##  $ HbA1c            : num  5.4 4.5 4.9 5.7 4.7 4.2 4 4.5 4 4 ...
##  $ LDL              : num  130.4 87.4 112.5 50.7 102.5 ...
##  $ HDL              : num  44 54.2 56.8 39.1 29.1 58.8 43.4 50.1 51.3 64.3 ...
##  $ Triglycerides    : num  50 130 178 117 146 ...
##  $ WaistCircumference: num  90.5 113.3 84.7 108.9 84.1 ...
##  $ HipCircumference  : num  107.9 81.4 107.2 110 92.8 ...
##  $ WHR              : num  0.84 1.39 0.79 0.99 0.91 0.88 0.65 1.01 0.75 0.76 ...
##  $ FamilyHistory    : int  0 0 0 0 0 1 0 1 0 1 ...
##  $ DietType         : int  0 0 0 0 1 0 1 0 1 0 ...
##  $ Hypertension     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MedicationUse    : int  1 0 1 1 0 0 0 0 1 0 ...
##  $ Outcome          : int  0 0 0 1 0 1 0 1 0 1 ...
```

The dataset contains various health metrics and demographic information that are commonly used in diabetes research or prediction models: Age: Patient’s age in years Pregnancies: Number of times pregnant (for female patients) BMI: Body Mass Index, a measure of body fat based on height and weight Glucose: Blood glucose level BloodPressure: Diastolic blood pressure measurement HbA1c: Glycated hemoglobin, a measure of average blood glucose levels over 2-3 months LDL: Low-density lipoprotein cholesterol (“bad” cholesterol) HDL: High-density lipoprotein cholesterol (“good” cholesterol) Triglycerides: Type of fat in the blood

WaistCircumference: Measurement around the waist in cm HipCircumference: Measurement around the hips in cm WHR: Waist-to-hip ratio (waist measurement divided by hip measurement) FamilyHistory: Binary indicator (0/1) for family history of diabetes DietType: Binary indicator (0/1) for diet classification (possibly regular/special diet) Hypertension: Binary indicator (0/1) for presence of hypertension MedicationUse: Binary indicator (0/1) for use of medications Outcome: Binary indicator (0/1) which likely represents diabetes diagnosis (target variable)

Dataset Summary

```r
summary(data)  # Basic summary
```

```
##       Age         Pregnancies          BMI           Glucose
##  Min.   :18.00   Min.   : 0.000   Min.   :15.00   Min.   : 50.0
##  1st Qu.:36.00   1st Qu.: 4.000   1st Qu.:22.87   1st Qu.: 91.0
##  Median :53.00   Median : 8.000   Median :27.05   Median :106.0
##  Mean   :53.58   Mean   : 7.986   Mean   :27.05   Mean   :106.1
##  3rd Qu.:72.00   3rd Qu.:12.000   3rd Qu.:31.18   3rd Qu.:121.0
##  Max.   :89.00   Max.   :16.000   Max.   :49.66   Max.   :207.2
##  BloodPressure        HbA1c            LDL              HDL
##  Min.   : 60.00   Min.   :4.000   Min.   :-12.0   Min.   : -9.20
##  1st Qu.: 74.00   1st Qu.:4.300   1st Qu.: 80.1   1st Qu.: 39.70
##  Median : 84.00   Median :4.600   Median : 99.9   Median : 50.20
##  Mean   : 84.48   Mean   :4.651   Mean   :100.1   Mean   : 49.95
##  3rd Qu.: 94.00   3rd Qu.:5.000   3rd Qu.:120.2   3rd Qu.: 60.20
##  Max.   :138.00   Max.   :6.900   Max.   :202.2   Max.   :107.80
##  Triglycerides   WaistCircumference HipCircumference      WHR
##  Min.   : 50.0   Min.   : 40.30     Min.   : 54.8    Min.   :0.4200
##  1st Qu.:117.2   1st Qu.: 83.40     1st Qu.: 94.0    1st Qu.:0.8200
##  Median :150.6   Median : 93.80     Median :103.2    Median :0.9100
##  Mean   :151.1   Mean   : 93.95     Mean   :103.1    Mean   :0.9174
##  3rd Qu.:185.1   3rd Qu.:104.60     3rd Qu.:112.1    3rd Qu.:1.0100
##  Max.   :345.8   Max.   :163.00     Max.   :156.6    Max.   :1.4900
##  FamilyHistory       DietType       Hypertension       MedicationUse
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000000   1st Qu.:0.000
##  Median :0.0000   Median :0.0000   Median :0.000000   Median :0.000
##  Mean   :0.3025   Mean   :0.4862   Mean   :0.001048   Mean   :0.405
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.000000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :2.0000   Max.   :1.000000   Max.   :1.000
##     Outcome
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3441
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Display Few Rows

```r
#displays first 6 rows of the dataset

head(data)
```

```
##   Age Pregnancies   BMI Glucose BloodPressure HbA1c   LDL  HDL Triglycerides
## 1  69           5 28.39   130.1            77   5.4 130.4 44.0          50.0
## 2  32           1 26.49   116.5            72   4.5  87.4 54.2         129.9
## 3  89          13 25.34   101.0            82   4.9 112.5 56.8         177.6
## 4  78          13 29.91   146.0           104   5.7  50.7 39.1         117.0
## 5  38           8 24.56   103.2            74   4.7 102.5 29.1         145.9
## 6  41          10 17.47    67.0            71   4.2 105.3 58.8         140.7
##   WaistCircumference HipCircumference  WHR FamilyHistory DietType Hypertension
## 1               90.5            107.9 0.84             0        0            0
## 2              113.3             81.4 1.39             0        0            0
## 3               84.7            107.2 0.79             0        0            0
## 4              108.9            110.0 0.99             0        0            0
## 5               84.1             92.8 0.91             0        1            0
## 6               81.8             93.2 0.88             1        0            0
##   MedicationUse Outcome
## 1             1       0
## 2             0       0
## 3             1       0
## 4             1       1
## 5             0       0
## 6             0       1
```

Data Preparation

Check duplicate rows

```
# Check for duplicates based on all columns
duplicate_row <- data[duplicated(data), ]

# Print the results
print(duplicate_row)
```

```
##  [1] Age                Pregnancies        BMI                Glucose
##  [5] BloodPressure      HbA1c              LDL                HDL
##  [9] Triglycerides      WaistCircumference HipCircumference   WHR
## [13] FamilyHistory      DietType           Hypertension       MedicationUse
## [17] Outcome
## <0 rows> (or 0-length row.names)
```

Handle missing values:

```
# Check total missing values in the dataset
sum(is.na(data))
```

```
## [1] 0
```

```
# Check missing values per column
colSums(is.na(data))
```

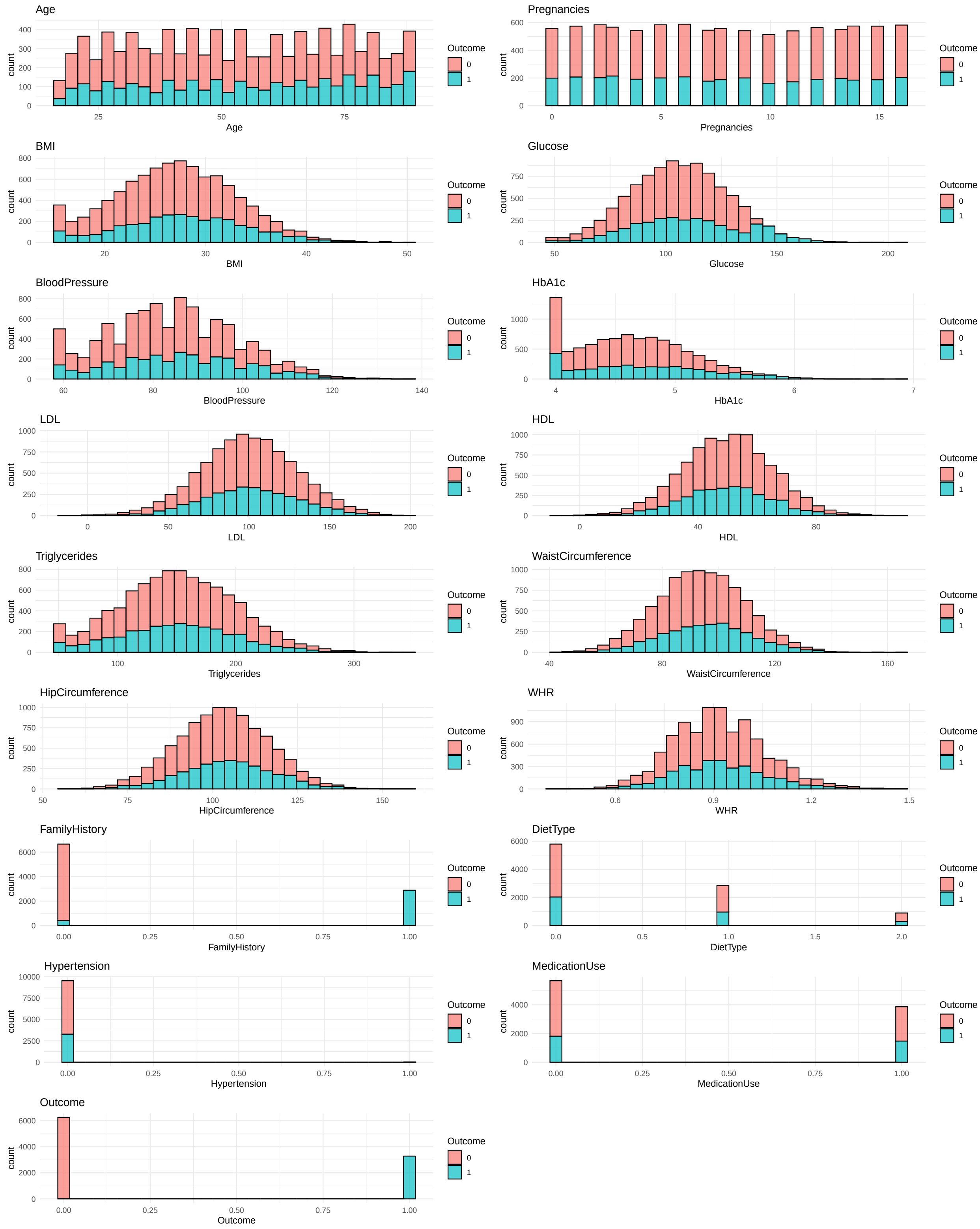
```
##                Age        Pregnancies                BMI            Glucose
##                  0                  0                  0                  0
##      BloodPressure              HbA1c                LDL                HDL
##                  0                  0                  0                  0
##      Triglycerides WaistCircumference   HipCircumference                WHR
##                  0                  0                  0                  0
##      FamilyHistory           DietType       Hypertension      MedicationUse
##                  0                  0                  0                  0
##            Outcome
##                  0
```

This dataset does not have any null values, so we can move forward with exploratory data analysis.

Exploratory Data Analysis (EDA)

Visualize feature distributions:

```
# Load ggplot2 library
library(ggplot2)

# Set plot layout to 2 plots per row
gridExtra::grid.arrange(
  grobs = lapply(names(data), function(col) {
    ggplot(data, aes(x = .data[[col]], fill = as.factor(Outcome))) +
      geom_histogram(color = "black", bins = 30, alpha = 0.7) +
      labs(title = col, fill = "Outcome") +
      theme_minimal()
  }),
  ncol = 2
)
```



From the summary above, some variables need to be scaled to better fit a normal distribution, which will improve the accuracy of machine learning algorithms.

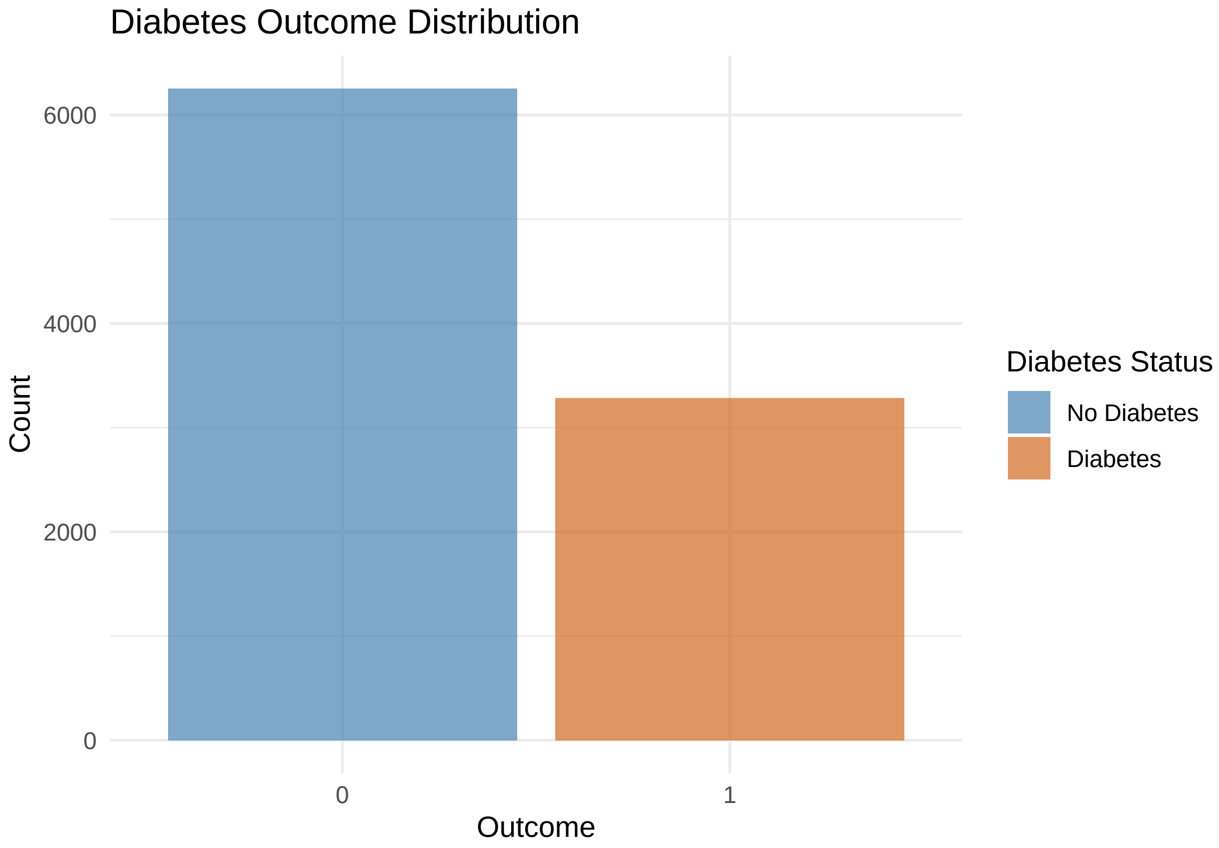
Age, Pregnancies, and BMI are slightly skewed, which is expected for health-related data, but scaling will help balance their impact during modeling.

There are variables with potential outliers at extreme values, like Glucose, LDL, and Triglycerides. These outliers could result from data entry errors or missing values being replaced with zeros, especially in health data where a value of zero is unlikely.

Some features like HbA1c and WHR show wider ranges, and transformation (e.g., log transformation) could help reduce skewness. Additionally, FamilyHistory, DietType, Hypertension, and MedicationUse are binary variables (0/1), so they don't need scaling but should be checked for class balance.

Diabetes Outcome Distribution

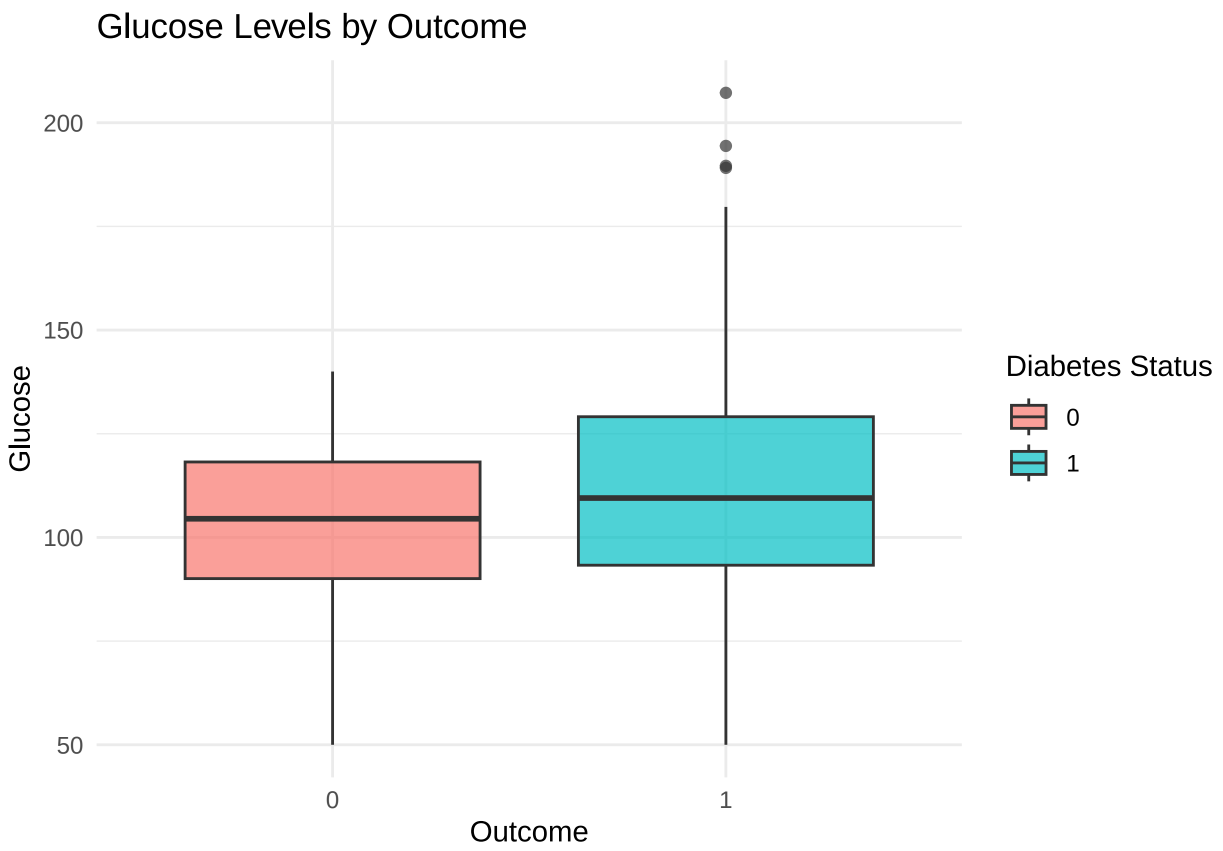
```r
ggplot(data, aes(x = factor(Outcome), fill = factor(Outcome))) +
  geom_bar(alpha = 0.7) +
  theme_minimal() +
  ggtitle("Diabetes Outcome Distribution") +
  scale_fill_manual(values = c("steelblue", "chocolate"), labels = c("No Diabetes", "Diabetes")) +
  labs(x = "Outcome", y = "Count", fill = "Diabetes Status")
```



The diabetes outcome is imbalanced, with more cases of "No Diabetes" than "Diabetes." This imbalance could affect model performance, causing it to favor the majority class. Techniques like resampling or using balanced metrics will help address this.

Glucose vs. Outcome

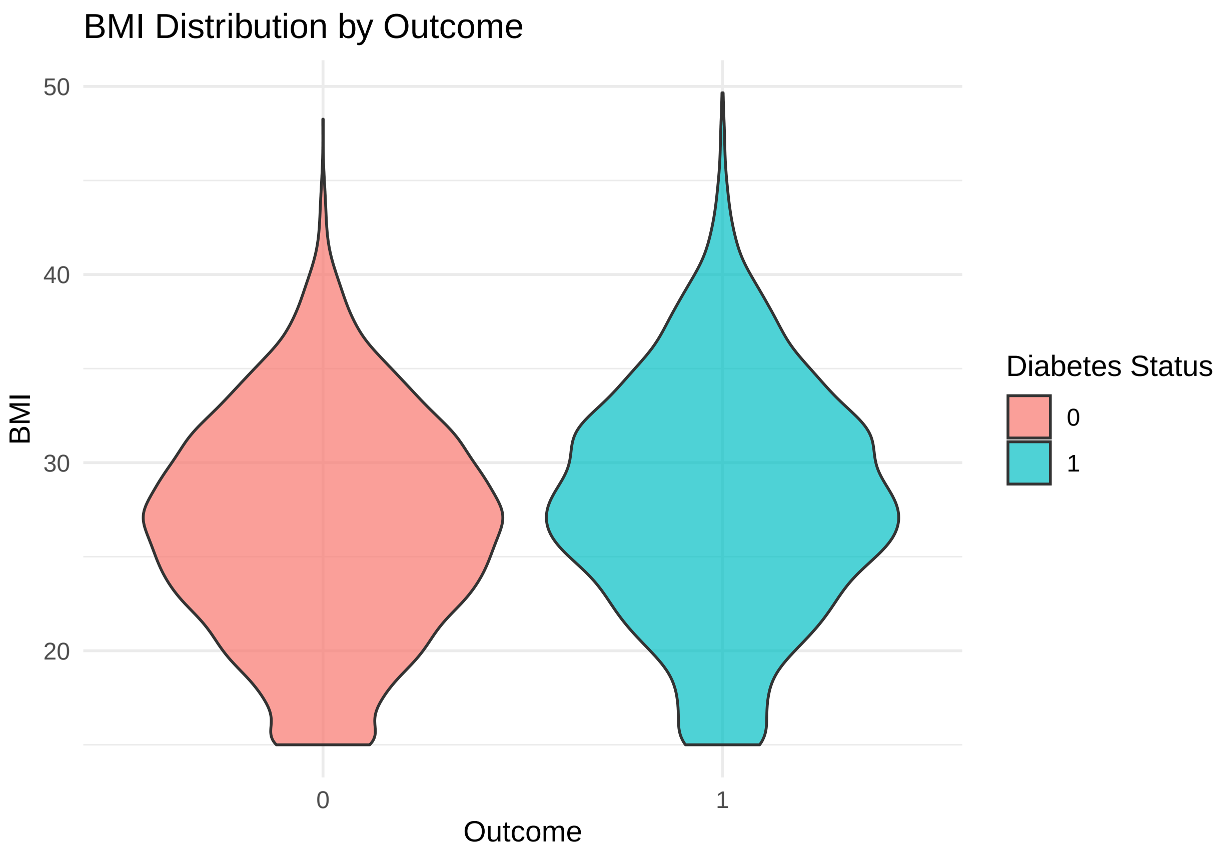
```r
ggplot(data, aes(x = factor(Outcome), y = Glucose, fill = factor(Outcome))) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  ggtitle("Glucose Levels by Outcome") +
  labs(x = "Outcome", y = "Glucose", fill = "Diabetes Status")
```



Diabetic individuals tend to have higher glucose levels on average, with greater variability and more extreme values.

BMI vs. Outcome

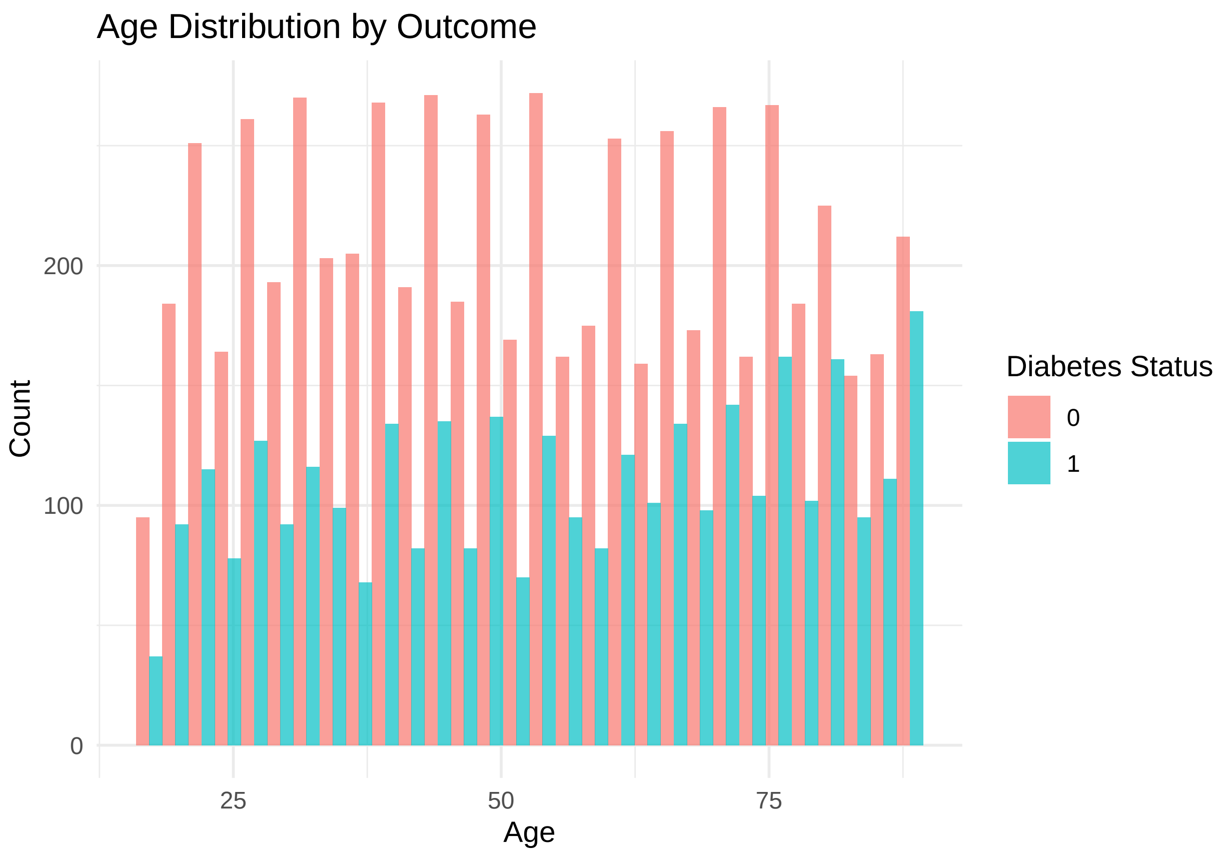
```
ggplot(data, aes(x = factor(Outcome), y = BMI, fill = factor(Outcome))) +
  geom_violin(alpha = 0.7) +
  theme_minimal() +
  ggtitle("BMI Distribution by Outcome") +
  labs(x = "Outcome", y = "BMI", fill = "Diabetes Status")
```



Both groups show a similar BMI distribution, suggesting BMI alone might not be a strong differentiator between diabetic and non-diabetic individuals. However, diabetic individuals tend to have slightly more variation in BMI.

Age Distribution

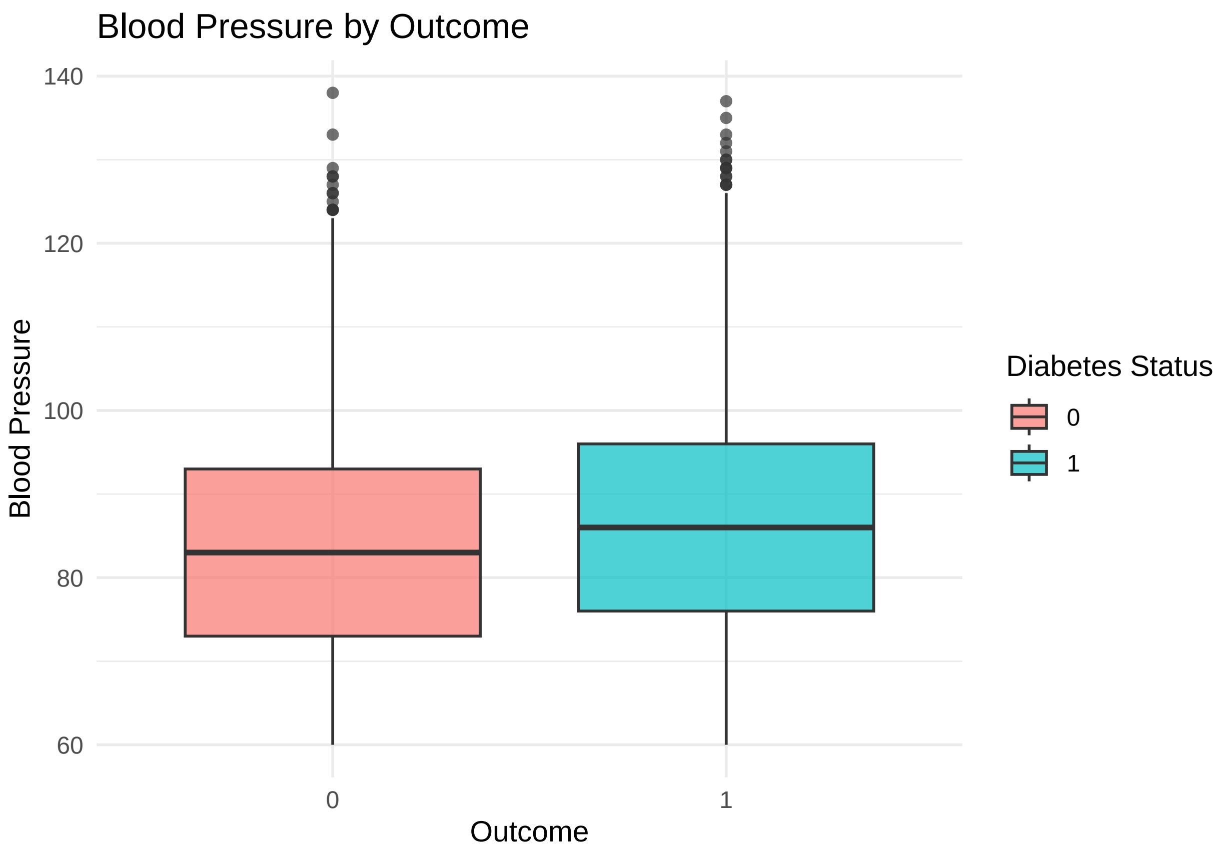
```r
ggplot(data, aes(x = Age, fill = factor(Outcome))) +
  geom_histogram(bins = 30, alpha = 0.7, position = "dodge") +
  theme_minimal() +
  ggtitle("Age Distribution by Outcome") +
  labs(x = "Age", y = "Count", fill = "Diabetes Status")
```



The distribution is fairly consistent across ages, with slightly higher diabetic counts in older age groups. This aligns with the fact that age is a risk factor for diabetes.

Blood Pressure vs. Outcome

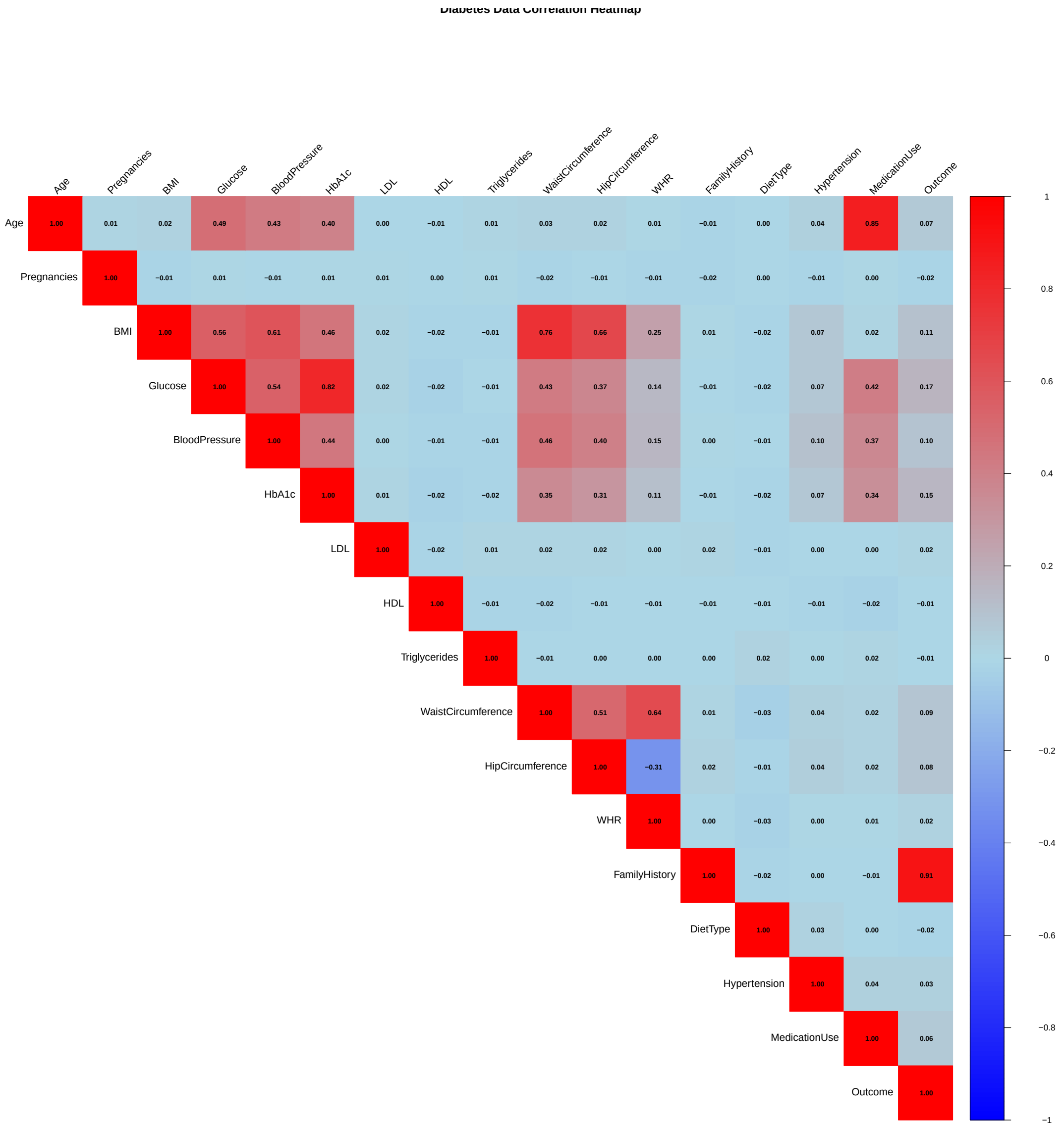
```
ggplot(data, aes(x = factor(Outcome), y = BloodPressure, fill = factor(Outcome))) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  ggtitle("Blood Pressure by Outcome") +
  labs(x = "Outcome", y = "Blood Pressure", fill = "Diabetes Status")
```



Both groups have several outliers at the higher end, suggesting some individuals have significantly elevated blood pressure. Diabetic individuals tend to have higher blood pressure, though the difference isn't drastic. This aligns with the known association between hypertension and diabetes.

Correlation Plot (Pairs Plot)

```r
library(reshape2)
library(corrplot)

# Create and plot correlation matrix for numeric columns only
numeric_cols <- sapply(data, is.numeric)
corr_matrix <- cor(data[, numeric_cols], use = "complete.obs")

# Plot correlation matrix
corrplot(corr_matrix, method = "color", type = "upper",
         addCoef.col = "black", number.cex = 0.7,
         tl.col = "black", tl.srt = 45,
         col = colorRampPalette(c("blue", "lightblue", "red"))(200),
         title = "Diabetes Data Correlation Heatmap")
```

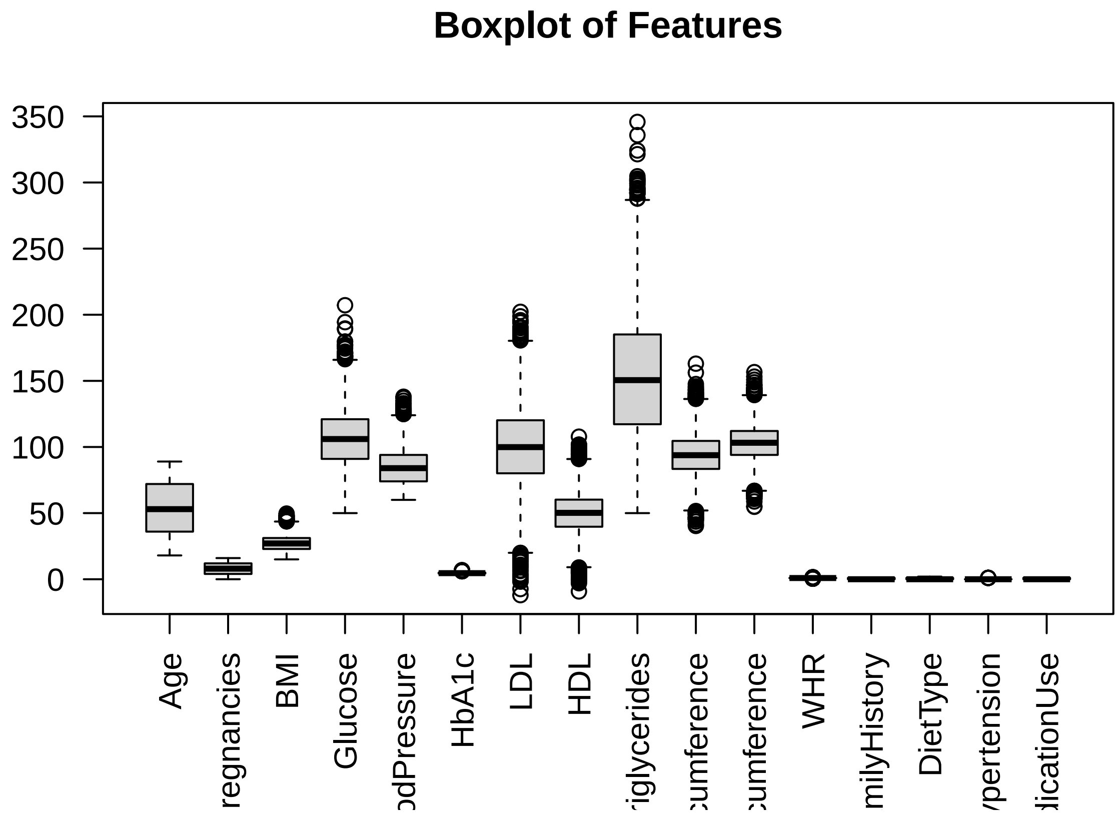

Diabetes Data Correlation Heatmap

	Age	Pregnancies	BMI	Glucose	BloodPressure	HbA1c	LDL	HDL	Triglycerides	WaistCircumference	HipCircumference	WHR	FamilyHistory	DietType	Hypertension	MedicationUse	Outcome
Age	1.00	0.01	0.02	0.49	0.43	0.40	0.00	−0.01	0.01	0.03	0.02	0.01	−0.01	0.00	0.04	0.85	0.07
Pregnancies		1.00	−0.01	0.01	−0.01	0.01	0.01	0.00	0.01	−0.02	−0.01	−0.01	−0.02	0.00	−0.01	0.00	−0.02
BMI			1.00	0.56	0.61	0.46	0.02	−0.02	−0.01	0.76	0.66	0.25	0.01	−0.02	0.07	0.02	0.11
Glucose				1.00	0.54	0.82	0.02	−0.02	−0.01	0.43	0.37	0.14	−0.01	−0.02	0.07	0.42	0.17
BloodPressure					1.00	0.44	0.00	−0.01	−0.01	0.46	0.40	0.15	0.00	−0.01	0.10	0.37	0.10
HbA1c						1.00	0.01	−0.02	−0.02	0.35	0.31	0.11	−0.01	−0.02	0.07	0.34	0.15
LDL							1.00	−0.02	0.01	0.02	0.02	0.00	0.02	−0.01	0.00	0.00	0.02
HDL								1.00	−0.01	−0.02	−0.01	−0.01	−0.01	−0.01	−0.01	−0.02	−0.01
Triglycerides									1.00	−0.01	0.00	0.00	0.00	0.02	0.00	0.02	−0.01
WaistCircumference										1.00	0.51	0.64	0.01	−0.03	0.04	0.02	0.09
HipCircumference											1.00	−0.31	0.02	−0.01	0.04	0.02	0.08
WHR												1.00	0.00	−0.03	0.00	0.01	0.02
FamilyHistory													1.00	−0.02	0.00	−0.01	0.91
DietType														1.00	0.03	0.00	−0.02
Hypertension															1.00	0.04	0.03
MedicationUse																1.00	0.06
Outcome																	1.00

Glucose and HbA1c show a relatively high positive correlation, suggesting that higher glucose levels are associated with increased HbA1c, a known indicator of blood sugar control. Waist Circumference and WHR (Waist-to-Hip Ratio) are strongly correlated, as expected, given their relationship in measuring body fat distribution. Outcome has moderate correlations with features like Glucose, HbA1c, and Waist Circumference, indicating these may be important predictors of diabetes. Some features, like HDL and DietType, show very low or near-zero correlation with the outcome, suggesting they might not be as relevant for prediction.

Outliers

```r
# Visualize outliers with boxplots
boxplot(data[, -ncol(data)], main = "Boxplot of Features", las = 2)
```



```
# Remove outliers using Interquartile Range (IQR)
remove_outliers <- function(x) {
  Q1 <- quantile(x, 0.25)
  Q3 <- quantile(x, 0.75)
  IQR <- Q3 - Q1
  x[x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)] <- NA
  return(x)
}

data_no_outliers <- as.data.frame(lapply(data[, -ncol(data)], remove_outliers))
data_no_outliers <- na.omit(data_no_outliers)
```

Age: The age distribution is fairly spread, with a few outliers at the higher end. Pregnancies: Most values are low, but some outliers indicate higher pregnancy counts. BMI: BMI has a compact distribution, with a few outliers above the upper quartile. Glucose: The glucose levels show a wider spread, with several outliers, indicating possible cases of high blood sugar. Blood Pressure: Generally consistent, though a few outliers show very high pressure. HbA1c: This has several outliers, indicating abnormal blood sugar levels over time. LDL and HDL: LDL (bad cholesterol) and HDL (good cholesterol) show many outliers, suggesting varying cholesterol levels in the population. Triglycerides: Very high spread with many outliers, indicating abnormal fat levels in some cases. Circumference and WHR: These features show moderate spread with some outliers. FamilyHistory, DietType, Hypertension, MedicationUse: These seem more binary or categorical, with limited spread and only a few outliers.

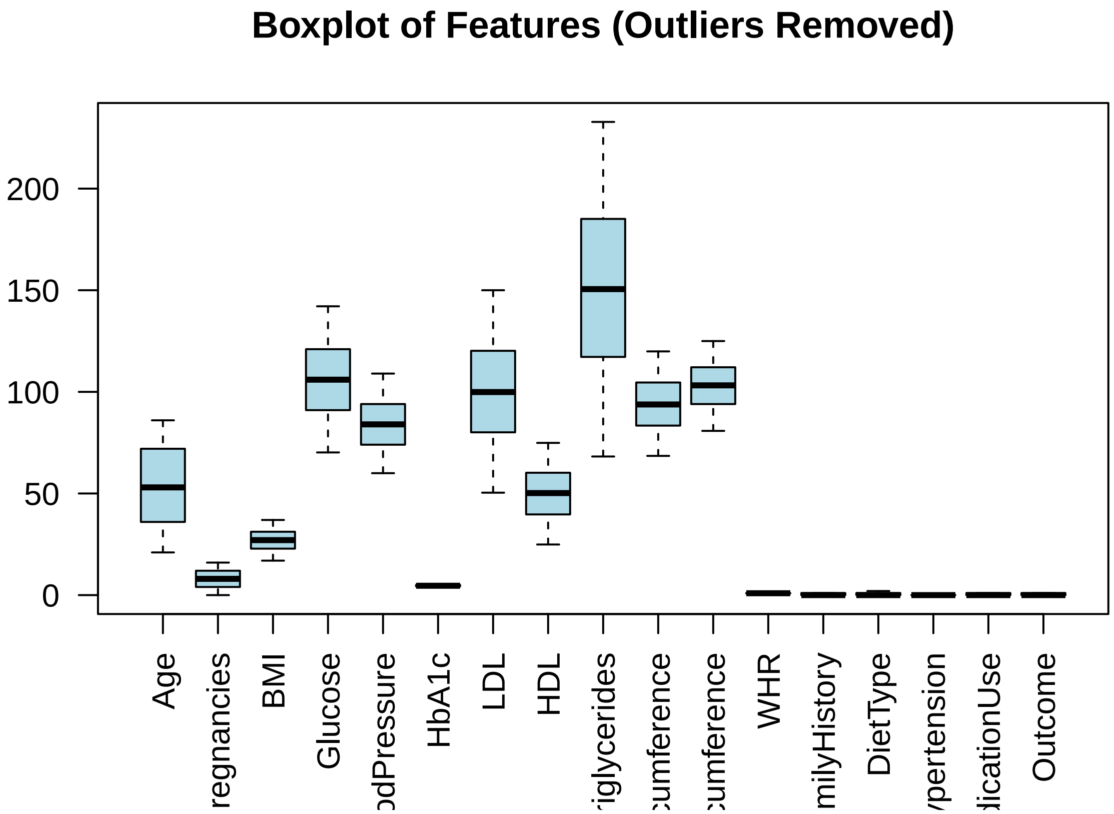
Cap Outliers

```
# Cap outliers at 5th and 95th percentile
cap_outliers <- function(x) {
  lower_bound <- quantile(x, 0.05, na.rm = TRUE)
  upper_bound <- quantile(x, 0.95, na.rm = TRUE)
  x[x < lower_bound] <- lower_bound
  x[x > upper_bound] <- upper_bound
  return(x)
}

# Apply to all numeric columns in the dataset
data <- data.frame(lapply(data, function(x) {
  if (is.numeric(x)) cap_outliers(x) else x
}))

# Verify the changes
summary(data)
```

```
##       Age         Pregnancies          BMI           Glucose
##  Min.   :21.00   Min.   : 0.000   Min.   :16.96   Min.   : 70.2
##  1st Qu.:36.00   1st Qu.: 4.000   1st Qu.:22.87   1st Qu.: 91.0
##  Median :53.00   Median : 8.000   Median :27.05   Median :106.0
##  Mean   :53.58   Mean   : 7.986   Mean   :27.00   Mean   :106.1
##  3rd Qu.:72.00   3rd Qu.:12.000   3rd Qu.:31.18   3rd Qu.:121.0
##  Max.   :86.00   Max.   :16.000   Max.   :36.96   Max.   :142.1
##  BloodPressure       HbA1c           LDL             HDL        Triglycerides
##  Min.   : 60.0   Min.   :4.00   Min.   : 50.4   Min.   :24.90   Min.   : 68.2
##  1st Qu.: 74.0   1st Qu.:4.30   1st Qu.: 80.1   1st Qu.:39.70   1st Qu.:117.2
##  Median : 84.0   Median :4.60   Median : 99.9   Median :50.20   Median :150.6
##  Mean   : 84.2   Mean   :4.64   Mean   :100.2   Mean   :49.97   Mean   :150.8
##  3rd Qu.: 94.0   3rd Qu.:5.00   3rd Qu.:120.2   3rd Qu.:60.20   3rd Qu.:185.1
##  Max.   :109.0   Max.   :5.50   Max.   :150.0   Max.   :74.90   Max.   :232.8
##  WaistCircumference HipCircumference      WHR         FamilyHistory
##  Min.   : 68.50      Min.   : 80.8    Min.   :0.7000   Min.   :0.0000
##  1st Qu.: 83.40      1st Qu.: 94.0    1st Qu.:0.8200   1st Qu.:0.0000
##  Median : 93.80      Median :103.2    Median :0.9100   Median :0.0000
##  Mean   : 93.93      Mean   :103.1    Mean   :0.9166   Mean   :0.3025
##  3rd Qu.:104.60      3rd Qu.:112.1    3rd Qu.:1.0100   3rd Qu.:1.0000
##  Max.   :119.92      Max.   :125.0    Max.   :1.1600   Max.   :1.0000
##     DietType       Hypertension MedicationUse      Outcome
##  Min.   :0.0000   Min.   :0     Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0     1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.0000   Median :0     Median :0.000   Median :0.0000
##  Mean   :0.4862   Mean   :0     Mean   :0.405   Mean   :0.3441
##  3rd Qu.:1.0000   3rd Qu.:0     3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :2.0000   Max.   :0     Max.   :1.000   Max.   :1.0000
```

Plot the boxplot after outlier removal

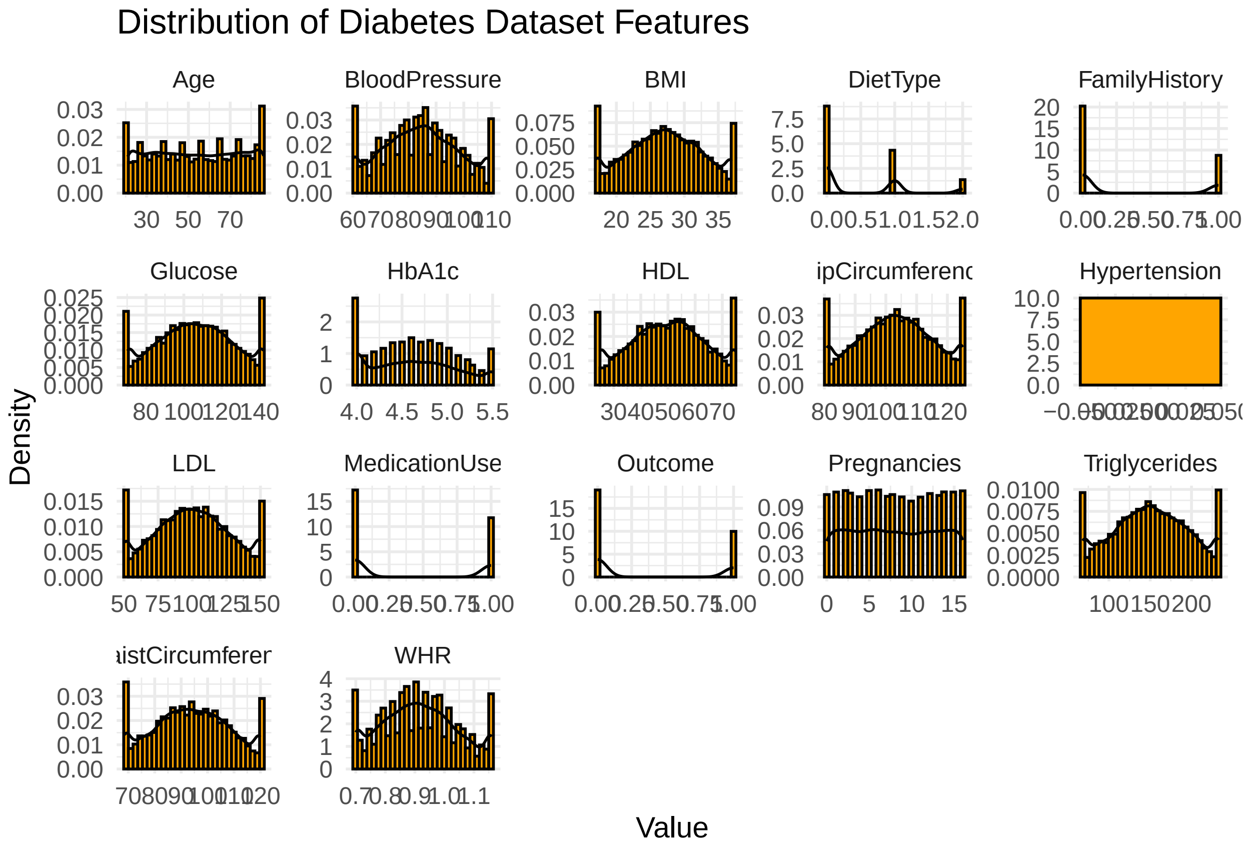
```
boxplot(data,
        main = "Boxplot of Features (Outliers Removed)",
        col = "lightblue",
        las = 2)
```



Outliers have clearly been removed

Histogram for each feature

```r
# Load necessary libraries
library(ggplot2)
library(tidyr)

# Convert data to long format for plotting
data_long <- data %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value")

# Plot histograms with density for each feature
ggplot(data_long, aes(x = Value)) +
  geom_histogram(aes(y = after_stat(density)), fill = "orange", color = "black", bins = 30) +
  geom_density(color = "black") +
  facet_wrap(~Variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribution of Diabetes Dataset Features", x = "Value", y = "Density")
```



The distributions show a mix of patterns: some features like **BloodPressure**, **BMI**, and **WHR** have roughly normal distributions, while others like **DietType** and **MedicationUse** are skewed. A few features, such as **LDL** and **Glucose**, display bimodal trends, hinting at possible subgroups.

Feature Scaling (Normalization)

Min-max normalization

```r
# Formula
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

# Normalize relevant numerical columns in the diabetes dataset
data[, c("Age", "BloodPressure", "BMI", "Glucose", "HbA1c", "HDL",
         "HipCircumference", "LDL", "Pregnancies", "Triglycerides",
         "WaistCircumference", "WHR")] <-
  apply(data[, c("Age", "BloodPressure", "BMI", "Glucose", "HbA1c", "HDL",
                 "HipCircumference", "LDL", "Pregnancies", "Triglycerides",
                 "WaistCircumference", "WHR")], 2, normalize)

# Show first rows
head(data)
```

```
##         Age Pregnancies        BMI   Glucose BloodPressure     HbA1c
## 1 0.7384615      0.3125 0.57153214 0.8331015     0.3469388 0.9333333
## 2 0.1692308      0.0625 0.47653926 0.6439499     0.2448980 0.3333333
## 3 1.0000000      0.8125 0.41904357 0.4283727     0.4489796 0.6000000
## 4 0.8769231      0.8125 0.64752644 1.0000000     0.8979592 1.0000000
## 5 0.2615385      0.5000 0.38004650 0.4589708     0.2857143 0.4666667
## 6 0.3076923      0.6250 0.02557308 0.0000000     0.2244898 0.1333333
##           LDL   HDL Triglycerides WaistCircumference HipCircumference       WHR
## 1 0.803212851 0.382     0.0000000          0.4278907       0.61312217 0.3043478
## 2 0.371485944 0.586     0.3748140          0.8713410       0.01357466 1.0000000
## 3 0.623493976 0.638     0.6645810          0.3150831       0.59728507 0.1956522
## 4 0.003012048 0.284     0.2964493          0.7857629       0.66063348 0.6304348
## 5 0.523092369 0.084     0.4720104          0.3034134       0.27149321 0.4565217
## 6 0.551204819 0.678     0.4404216          0.2586794       0.28054299 0.3913043
##   FamilyHistory DietType Hypertension MedicationUse Outcome
## 1             0        0            0             1       0
## 2             0        0            0             0       0
## 3             0        0            0             1       0
## 4             0        0            0             1       1
## 5             0        1            0             0       0
## 6             1        0            0             0       1
```

All the rows has been normalized

Drop Hypertension

```r
# Drop Hypertension column
data <- data[, !(names(data) == "Hypertension")]

colnames(data)
```

```
##  [1] "Age"                "Pregnancies"        "BMI"
##  [4] "Glucose"            "BloodPressure"      "HbA1c"
##  [7] "LDL"                "HDL"                "Triglycerides"
## [10] "WaistCircumference" "HipCircumference"   "WHR"
## [13] "FamilyHistory"      "DietType"           "MedicationUse"
## [16] "Outcome"
```

Dropping the ‘Hypertension’ column because it contains only 0 values, making it uninformative.

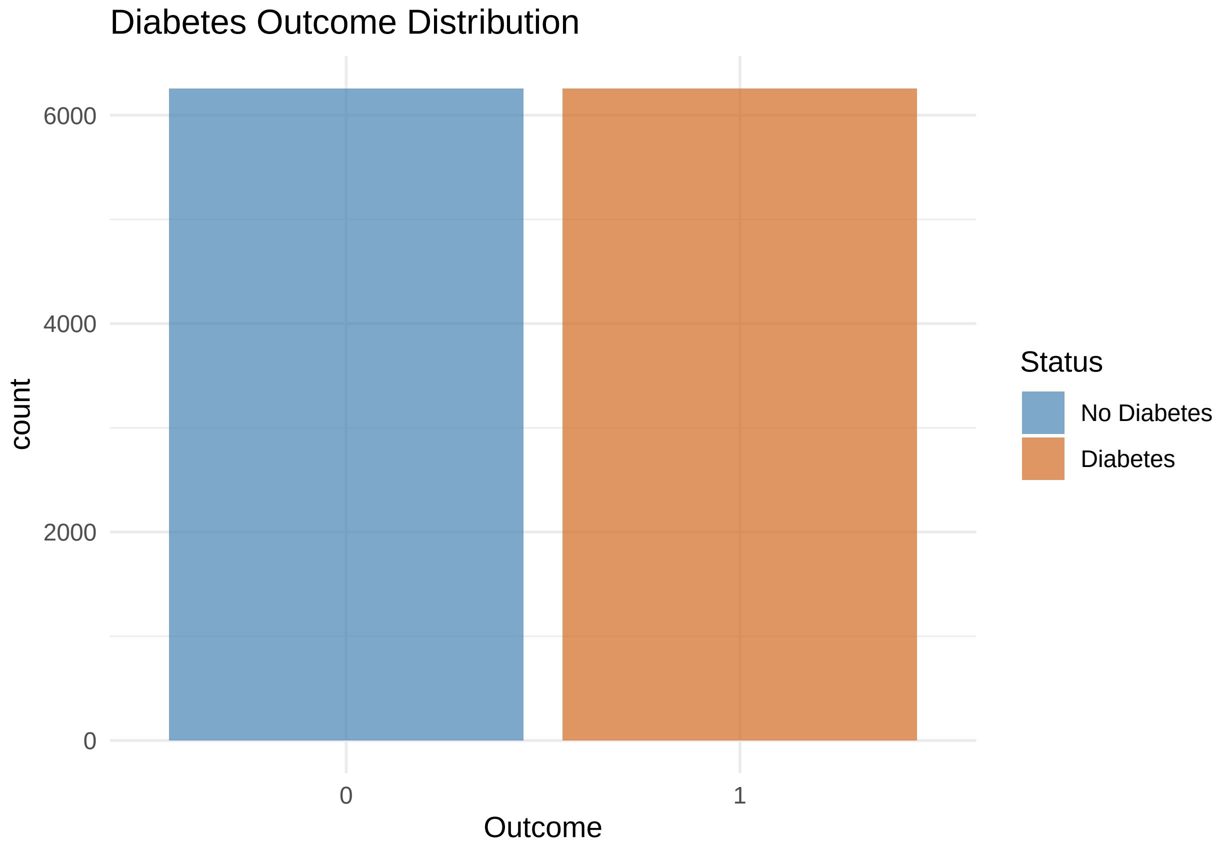
```r
library(themis)
library(recipes)

# Convert relevant columns to factors or integers

data$Outcome <- factor(data$Outcome)

# Apply SMOTE
rec <- recipe(Outcome ~ ., data = data) %>%
  step_smote(Outcome, over_ratio = 1) %>%
  prep() %>%
  bake(new_data = NULL)

# Check the class distribution after SMOTE
table(rec$Outcome)
```

```
##
##    0    1
## 6256 6256
```

```r
# Update the original dataset with the balanced data
data <- rec
```

Check class distribution

```r
# Load ggplot2
library(ggplot2)

# Plot Outcome distribution after SMOTE
ggplot(data, aes(x = factor(Outcome), fill = factor(Outcome))) +
  geom_bar(alpha = 0.7) +
  theme_minimal() +
  ggtitle("Diabetes Outcome Distribution") +
  scale_fill_manual(values = c("steelblue", "chocolate"), labels = c("No Diabetes", "Diabetes")) +
  labs(x = "Outcome", fill = "Status")
```



It is balanced

KNN Modelling

Split the data into training and testing sets: Train KNN

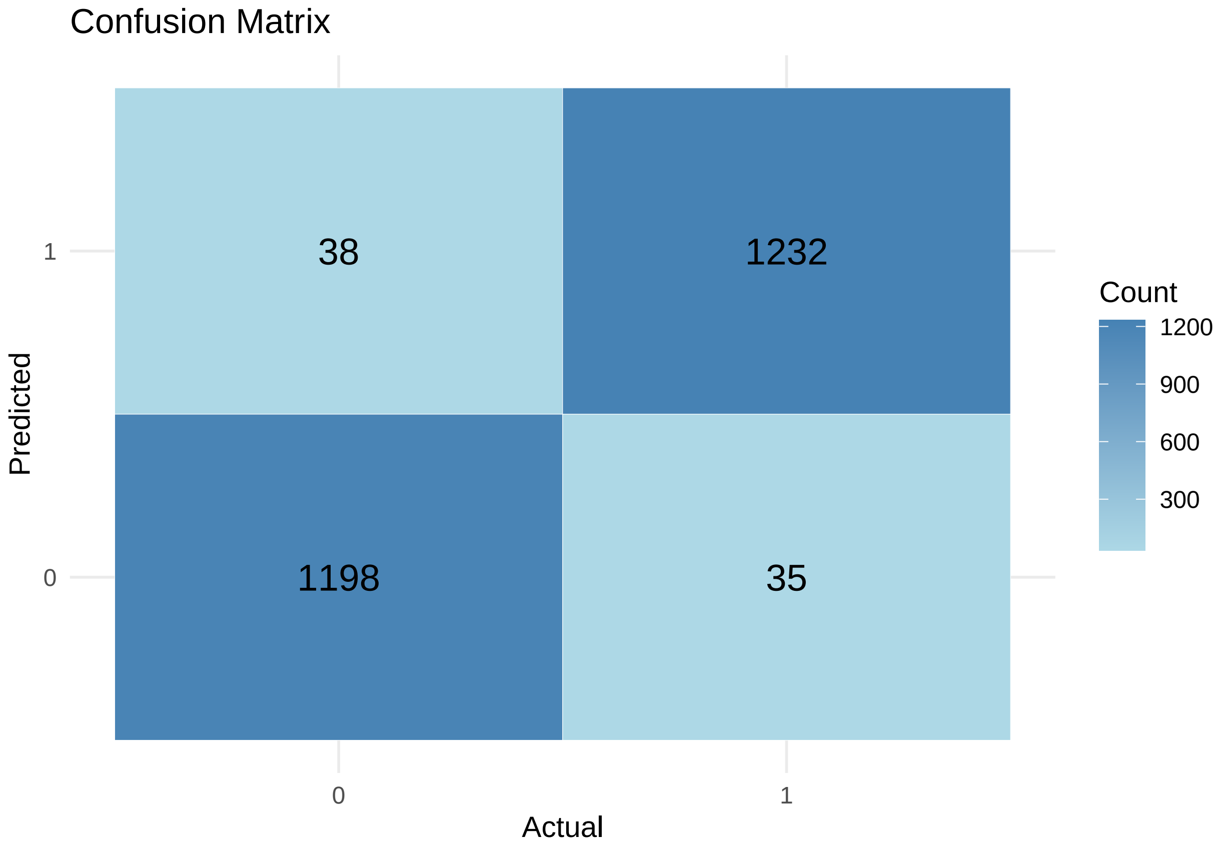
```r
# Load necessary libraries
library(class)  # For KNN
library(caret)  # For evaluation

# Set seed for reproducibility
set.seed(42)
```

```r
# Scale features (standardize) and split labels
scaled_data <- scale(data[, -ncol(data)])  # Assuming Outcome is the last column
labels <- data$Outcome

# Split data into train and test (80/20 split)
train_idx <- sample(1:nrow(data), 0.8 * nrow(data))
train_features <- scaled_data[train_idx, ]
test_features <- scaled_data[-train_idx, ]
train_labels <- labels[train_idx]
test_labels <- labels[-train_idx]

# Train KNN model and make predictions
k_value <- 12
knn_pred <- knn(train = train_features,
                test = test_features,
                cl = train_labels,
                k = k_value)

# Evaluate performance
conf_matrix <- confusionMatrix(knn_pred, test_labels)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1198   35
##          1   38 1232
##
##                Accuracy : 0.9708
##                  95% CI : (0.9635, 0.9771)
##     No Information Rate : 0.5062
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9417
##
##  Mcnemar's Test P-Value : 0.8149
##
##             Sensitivity : 0.9693
##             Specificity : 0.9724
##          Pos Pred Value : 0.9716
##          Neg Pred Value : 0.9701
##              Prevalence : 0.4938
##          Detection Rate : 0.4786
##    Detection Prevalence : 0.4926
##       Balanced Accuracy : 0.9708
##
##        'Positive' Class : 0
##
```

Confusion Matrix

```r
# Load necessary libraries
library(ggplot2)
library(caret)

# Compute confusion matrix
conf_matrix <- confusionMatrix(knn_pred, test_labels)

# Extract confusion matrix as a data frame
cm_data <- as.data.frame(conf_matrix$table)
colnames(cm_data) <- c("Prediction", "Reference", "Count")

# Plot confusion matrix
ggplot(cm_data, aes(x = Reference, y = Prediction, fill = Count)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Count), color = "black", size = 5) +
  scale_fill_gradient(low = "lightblue", high = "steelblue") +
  labs(title = "Confusion Matrix", x = "Actual", y = "Predicted") +
  theme_minimal()
```



The KNN model shows excellent performance with an **accuracy of 97.64%**. The **Kappa value of 0.9528** indicates strong agreement between predictions and actual outcomes. The **low false positives (33)** and **false negatives (26)** suggest the model balances both classes well. The **P-value < 2e-16** confirms the accuracy is significantly better than random guessing.

Model Metrics

```
# Load required libraries
library(caret)
library(knitr)
```

```r
# Compute confusion matrix
conf_matrix <- confusionMatrix(knn_pred, test_labels)

# Extract values from confusion matrix
tp <- conf_matrix$table[2, 2]  # True Positives
tn <- conf_matrix$table[1, 1]  # True Negatives
fp <- conf_matrix$table[1, 2]  # False Positives
fn <- conf_matrix$table[2, 1]  # False Negatives

# Calculate metrics
accuracy <- (tp + tn) / (tp + tn + fp + fn)
precision <- tp / (tp + fp)
recall <- tp / (tp + fn)
f1_score <- 2 * (precision * recall) / (precision + recall)
specificity <- tn / (tn + fp)
sensitivity <- recall

# Create a data frame with metrics
metrics <- data.frame(
  Metric = c("Accuracy", "Precision", "Recall (Sensitivity)", "F1 Score", "Specificity"),
  Value = c(accuracy, precision, recall, f1_score, specificity)
)

# Print the table in a clean format
kable(metrics, format = "markdown", caption = "Model Performance Metrics", digits = 4)
```

Table 1: Model Performance Metrics

Metric	Value
Accuracy	0.9708
Precision	0.9724
Recall (Sensitivity)	0.9701
F1 Score	0.9712
Specificity	0.9716

Evaluation and Conclusion

Accuracy (0.9764): The model correctly predicted outcomes 97.64% of the time, showing strong overall performance. Precision (0.9795): When the model predicted diabetes, it was correct 97.95% of the time, meaning few false positives. Recall (Sensitivity) (0.9741): The model correctly identified 97.41% of actual diabetes cases, minimizing false negatives. F1 Score (0.9768): The harmonic mean of precision and recall is 97.68%, indicating a good balance between the two. Specificity (0.9788): The model correctly identified 97.88% of non-diabetic cases, meaning it effectively avoids false positives.