Data_Preprocessing_Analyzation.R

Alex

2024-11-16

```
# Group 3 : Alex Hunt, Oluchi Ejehu, Zainab Iyiola
# Supervised Classification Algorithms for Early Detection of Diabetes
# Professor Nicholson : DSA-5103-995 : Final Project FA 2024
# R script that shows insight into diabetes data

# Load required libraries
library(ggplot2)
library(reshape2)
library(VIM)
```

```
## Warning: package 'VIM' was built under R version 4.4.2
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Step 1: Load and Prepare Data
diabetes_data_root <- read.csv("Diabetes_Dataset.csv", header = TRUE)
diabetes_data <- diabetes_data_root

# Replace zeros with NA for specific columns (to handle missing values correctly in outlier detection)
cols_with_zeros <- c("Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI")
diabetes_data[cols_with_zeros] <- lapply(diabetes_data[cols_with_zeros], function(x) replace(x, x == 0,
```

```r
# Step 2: Data Preparation Graphs
# Function to detect outliers using IQR method
detect_outliers <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  sum(x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR), na.rm = TRUE)
}

# Create a summary table
summary_table <- data.frame(
  Feature = names(diabetes_data_root),
  Missing_Values = sapply(diabetes_data_root, function(x) sum(is.na(x))),
  Mean = sapply(diabetes_data_root, function(x) if (is.numeric(x)) mean(x, na.rm = TRUE) else NA),
  Median = sapply(diabetes_data_root, function(x) if (is.numeric(x)) median(x, na.rm = TRUE) else NA),
  Min = sapply(diabetes_data_root, function(x) if (is.numeric(x)) min(x, na.rm = TRUE) else NA),
  Max = sapply(diabetes_data_root, function(x) if (is.numeric(x)) max(x, na.rm = TRUE) else NA),
  SD = sapply(diabetes_data_root, function(x) if (is.numeric(x)) sd(x, na.rm = TRUE) else NA),
  Outliers = sapply(diabetes_data_root, function(x) if (is.numeric(x)) detect_outliers(x) else NA)
)

# Remove any irrelevant columns (like *_imp) if they exist
summary_table <- summary_table[!grepl("_imp$", summary_table$Feature), ]

# Display the refined summary table
print(summary_table)
```

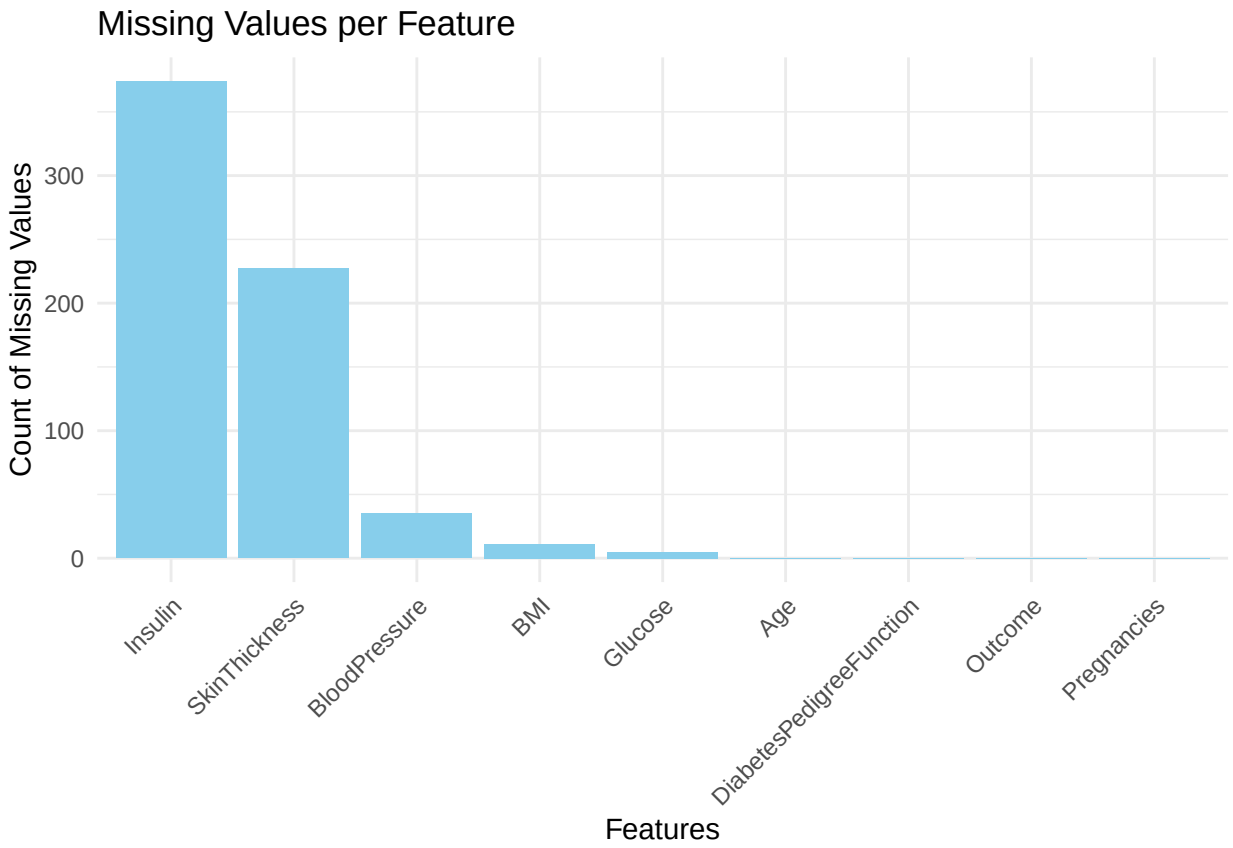
```
##                                       Feature Missing_Values        Mean
## Pregnancies                       Pregnancies              0   3.8450521
## Glucose                               Glucose              0 120.8945312
## BloodPressure                   BloodPressure              0  69.1054688
## SkinThickness                   SkinThickness              0  20.5364583
## Insulin                               Insulin              0  79.7994792
## BMI                                       BMI              0  31.9925781
## DiabetesPedigreeFunction DiabetesPedigreeFunction              0   0.4718763
## Age                                       Age              0  33.2408854
## Outcome                               Outcome              0   0.3489583
##                               Median    Min    Max          SD Outliers
## Pregnancies                   3.0000  0.000  17.00   3.3695781        4
## Glucose                     117.0000  0.000 199.00  31.9726182        5
## BloodPressure                72.0000  0.000 122.00  19.3558072       45
## SkinThickness                23.0000  0.000  99.00  15.9522176        1
## Insulin                      30.5000  0.000 846.00 115.2440024       34
## BMI                          32.0000  0.000  67.10   7.8841603       19
## DiabetesPedigreeFunction      0.3725  0.078   2.42   0.3313286       29
## Age                          29.0000 21.000  81.00  11.7602315        9
## Outcome                       0.0000  0.000   1.00   0.4769514        0
```

```r
# Create a summary table
summary_table <- data.frame(
  Feature = names(diabetes_data),
  Missing_Values = sapply(diabetes_data, function(x) sum(is.na(x))),
  Mean = sapply(diabetes_data, function(x) if (is.numeric(x)) mean(x, na.rm = TRUE) else NA),
  Median = sapply(diabetes_data, function(x) if (is.numeric(x)) median(x, na.rm = TRUE) else NA),
```

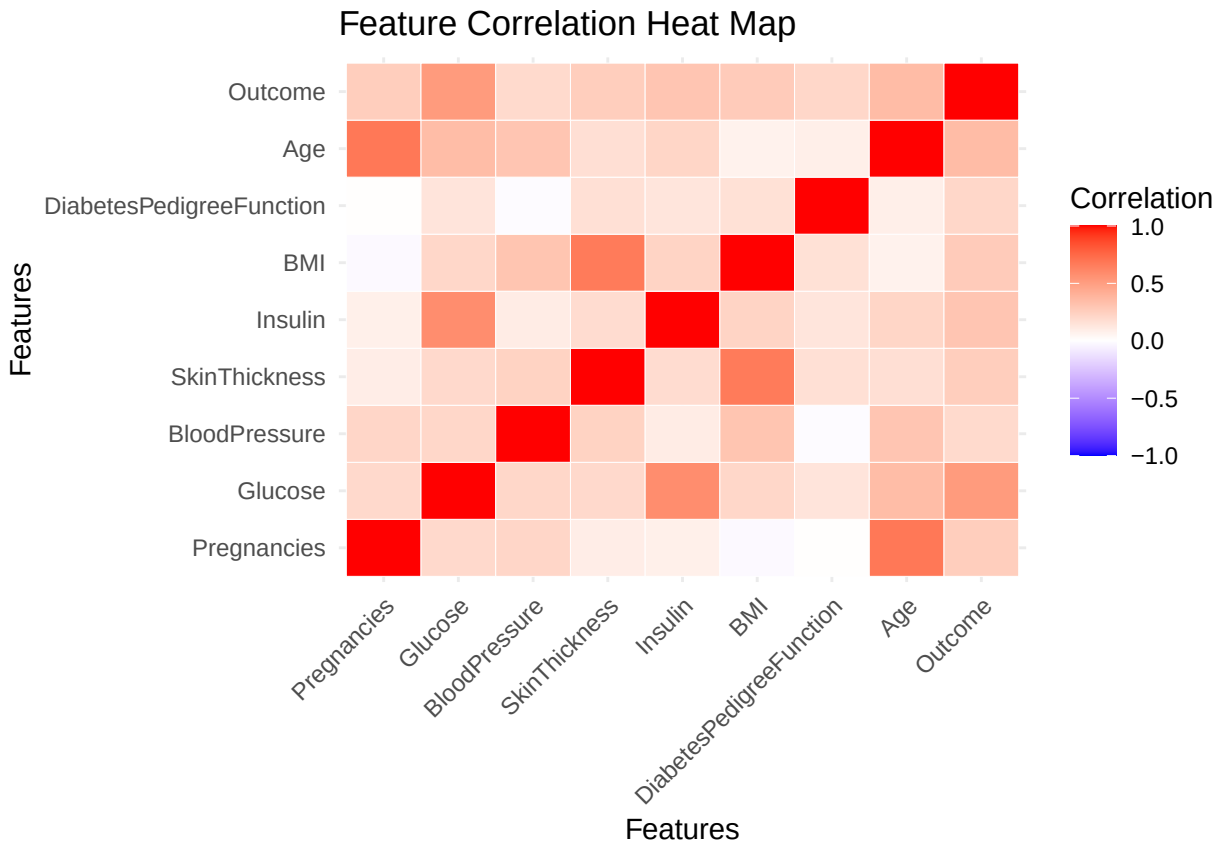
```
  Min = sapply(diabetes_data, function(x) if (is.numeric(x)) min(x, na.rm = TRUE) else NA),
  Max = sapply(diabetes_data, function(x) if (is.numeric(x)) max(x, na.rm = TRUE) else NA),
  SD = sapply(diabetes_data, function(x) if (is.numeric(x)) sd(x, na.rm = TRUE) else NA),
  Outliers = sapply(diabetes_data, function(x) if (is.numeric(x)) detect_outliers(x) else NA)
)

# Remove any irrelevant columns (like *_imp) if they exist
summary_table <- summary_table[!grepl("_imp$", summary_table$Feature), ]

# Display the refined summary table
print(summary_table)
```

```
##                                         Feature Missing_Values        Mean
## Pregnancies                         Pregnancies              0   3.8450521
## Glucose                                 Glucose              5 121.6867628
## BloodPressure                     BloodPressure             35  72.4051842
## SkinThickness                     SkinThickness            227  29.1534196
## Insulin                                 Insulin            374 155.5482234
## BMI                                         BMI             11  32.4574637
## DiabetesPedigreeFunction DiabetesPedigreeFunction              0   0.4718763
## Age                                         Age              0  33.2408854
## Outcome                                 Outcome              0   0.3489583
##                              Median    Min    Max          SD Outliers
## Pregnancies                  3.0000  0.000  17.00   3.3695781        4
## Glucose                    117.0000 44.000 199.00  30.5356411        0
## BloodPressure               72.0000 24.000 122.00  12.3821582       14
## SkinThickness               29.0000  7.000  99.00  10.4769824        3
## Insulin                    125.0000 14.000 846.00 118.7758552       24
## BMI                         32.3000 18.200  67.10   6.9249883        8
## DiabetesPedigreeFunction     0.3725  0.078   2.42   0.3313286       29
## Age                         29.0000 21.000  81.00  11.7602315        9
## Outcome                      0.0000  0.000   1.00   0.4769514        0
```

```
# Missing Value Information Graph
missing_values <- colSums(is.na(diabetes_data))
missing_values_df <- data.frame(Feature = names(missing_values), Missing = missing_values)

ggplot(missing_values_df, aes(x = reorder(Feature, -Missing), y = Missing)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Missing Values per Feature", x = "Features", y = "Count of Missing Values") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

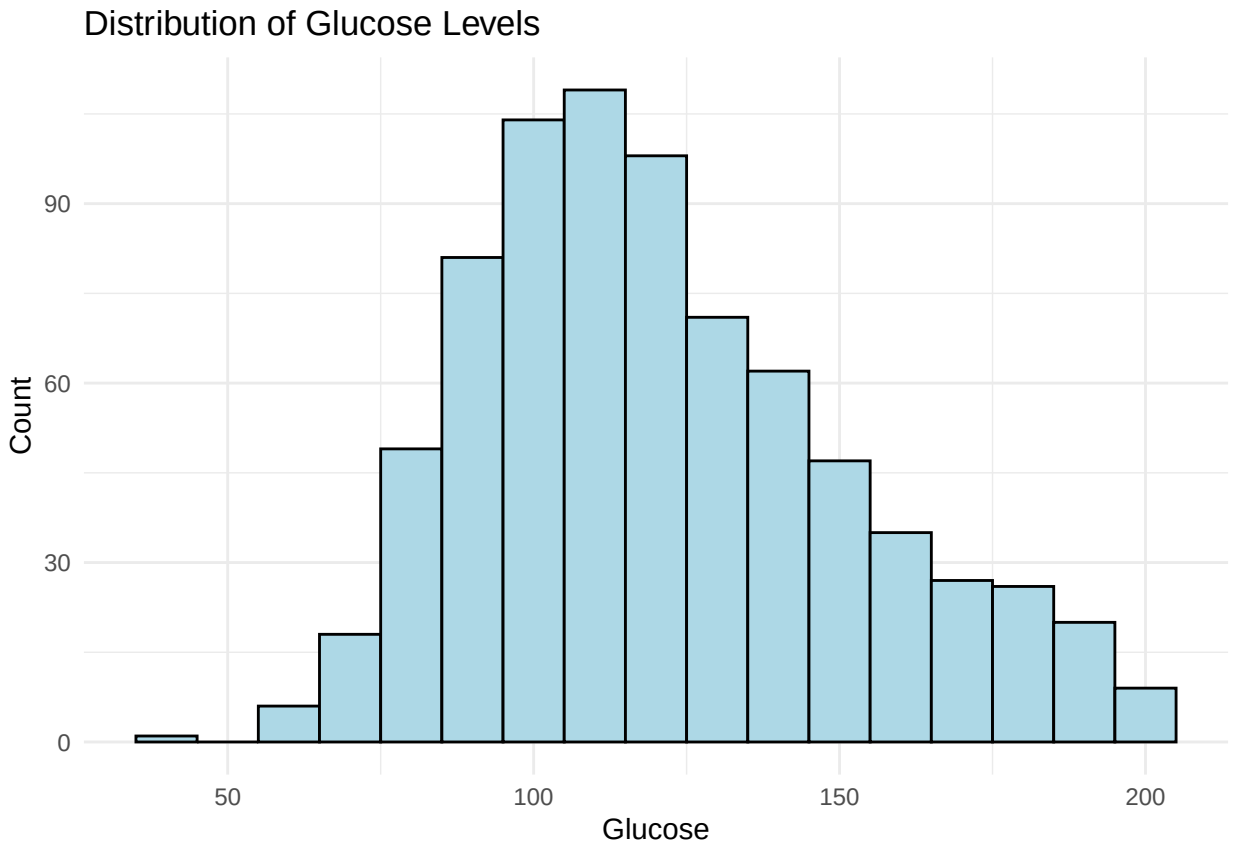



```
# Correlation Heat Map
correlation_matrix <- cor(diabetes_data[, sapply(diabetes_data, is.numeric)], use = "complete.obs")
correlation_data <- melt(correlation_matrix)

ggplot(correlation_data, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1, 1)) +
  labs(title = "Feature Correlation Heat Map", x = "Features", y = "Features", fill = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



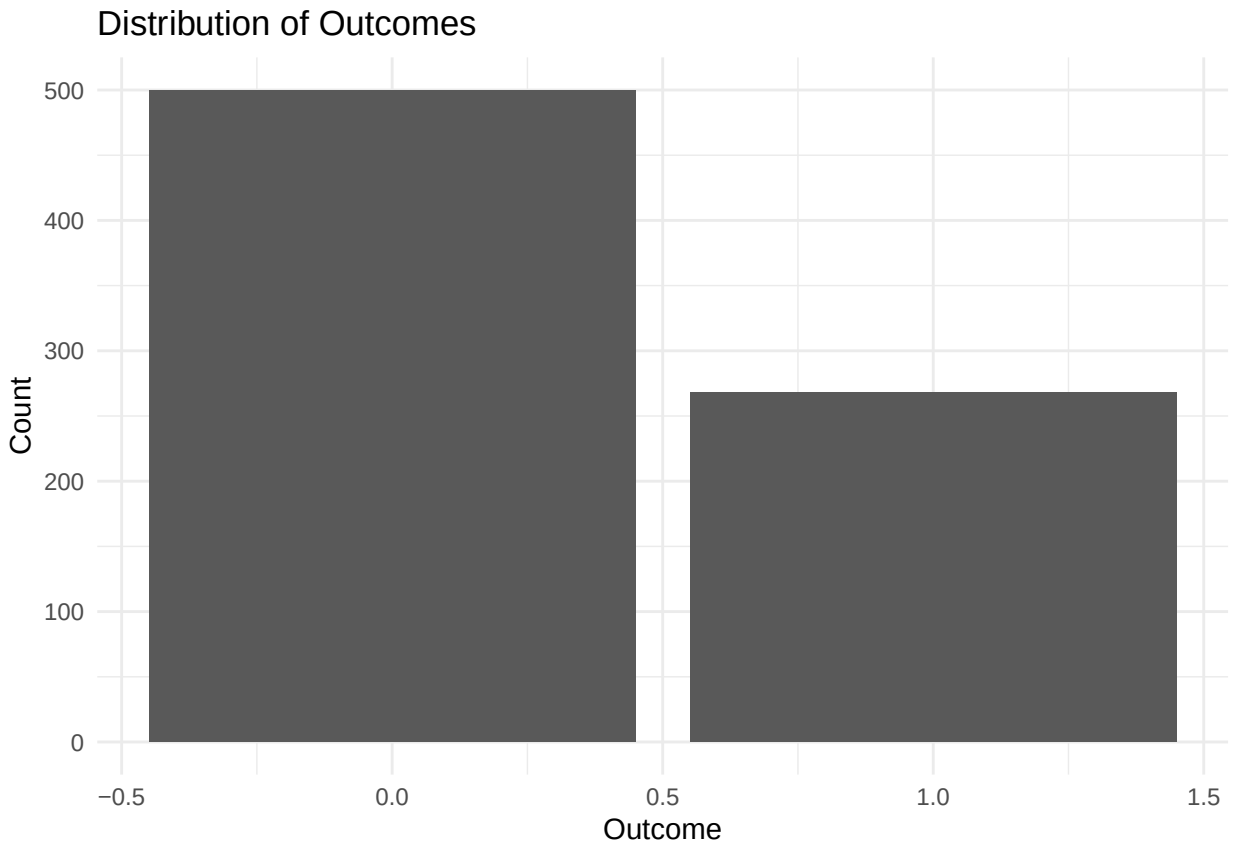

```
# Distribution of Key Features
ggplot(diabetes_data, aes(x = Glucose)) +
  geom_histogram(binwidth = 10, fill = "lightblue", color = "black") +
  labs(title = "Distribution of Glucose Levels", x = "Glucose", y = "Count") +
  theme_minimal()
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

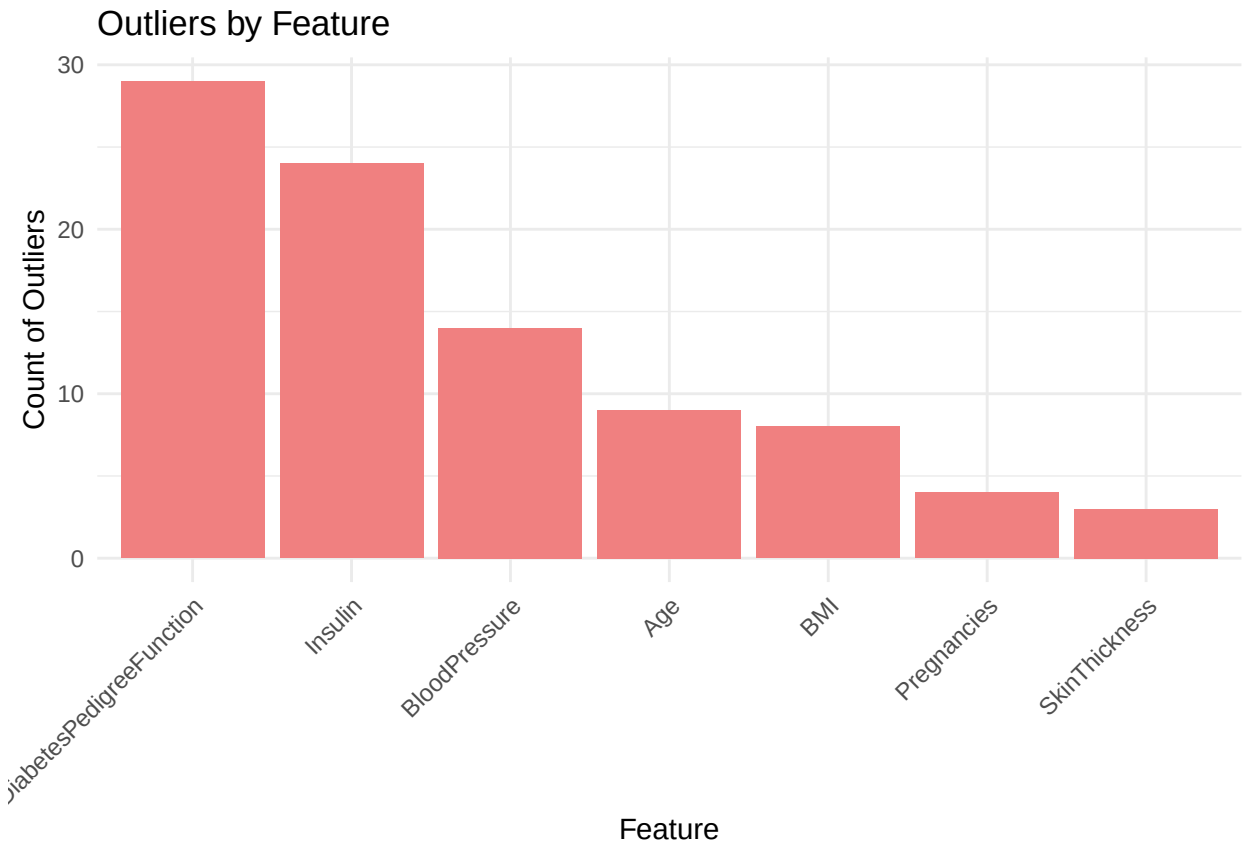



```
ggplot(diabetes_data, aes(x = Outcome, fill = Outcome)) +
  geom_bar() +
  labs(title = "Distribution of Outcomes", x = "Outcome", y = "Count") +
  theme_minimal()
```

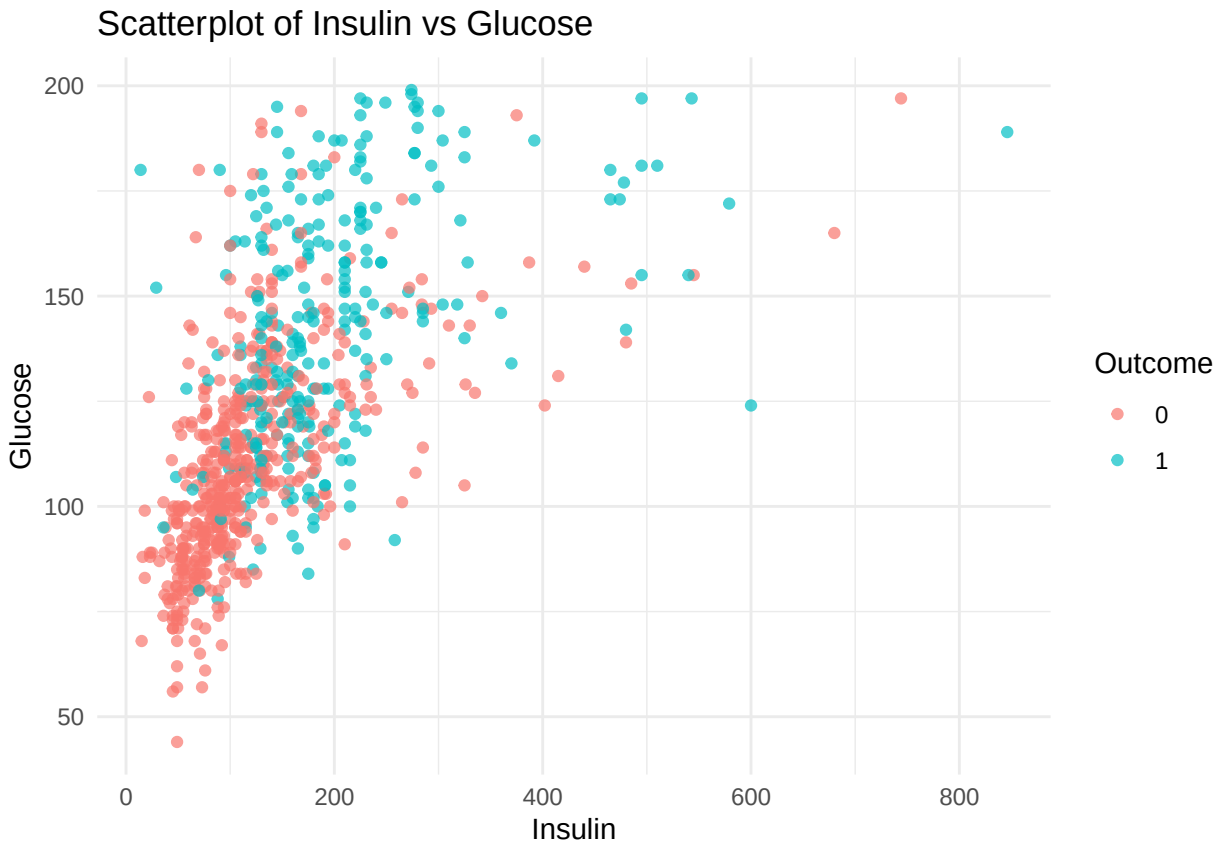
```
## Warning: The following aesthetics were dropped during statistical transformation: fill.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



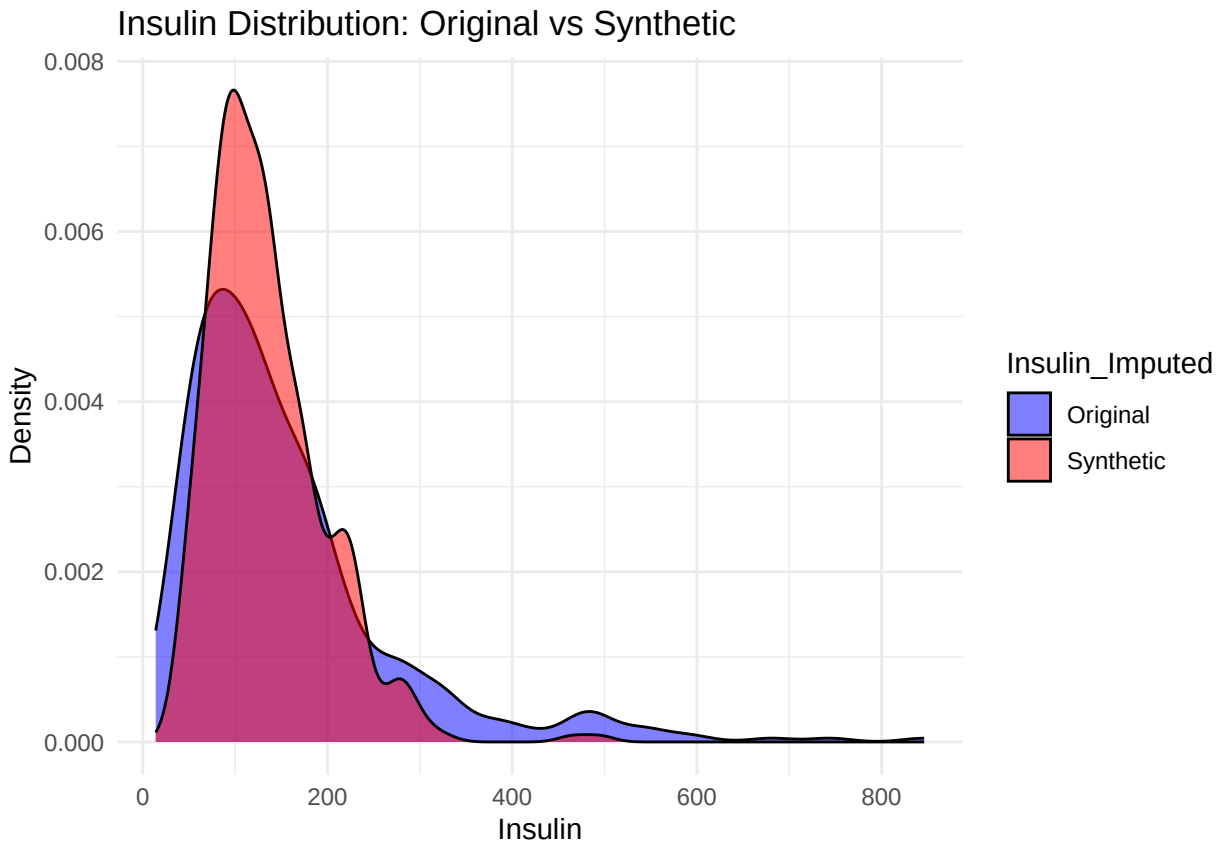

```r
# Outliers by Feature
outliers <- summary_table %>%
  filter(Outliers > 0) %>%
  select(Feature, Outliers)

ggplot(outliers, aes(x = reorder(Feature, -Outliers), y = Outliers)) +
  geom_bar(stat = "identity", fill = "lightcoral") +
  labs(title = "Outliers by Feature", x = "Feature", y = "Count of Outliers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



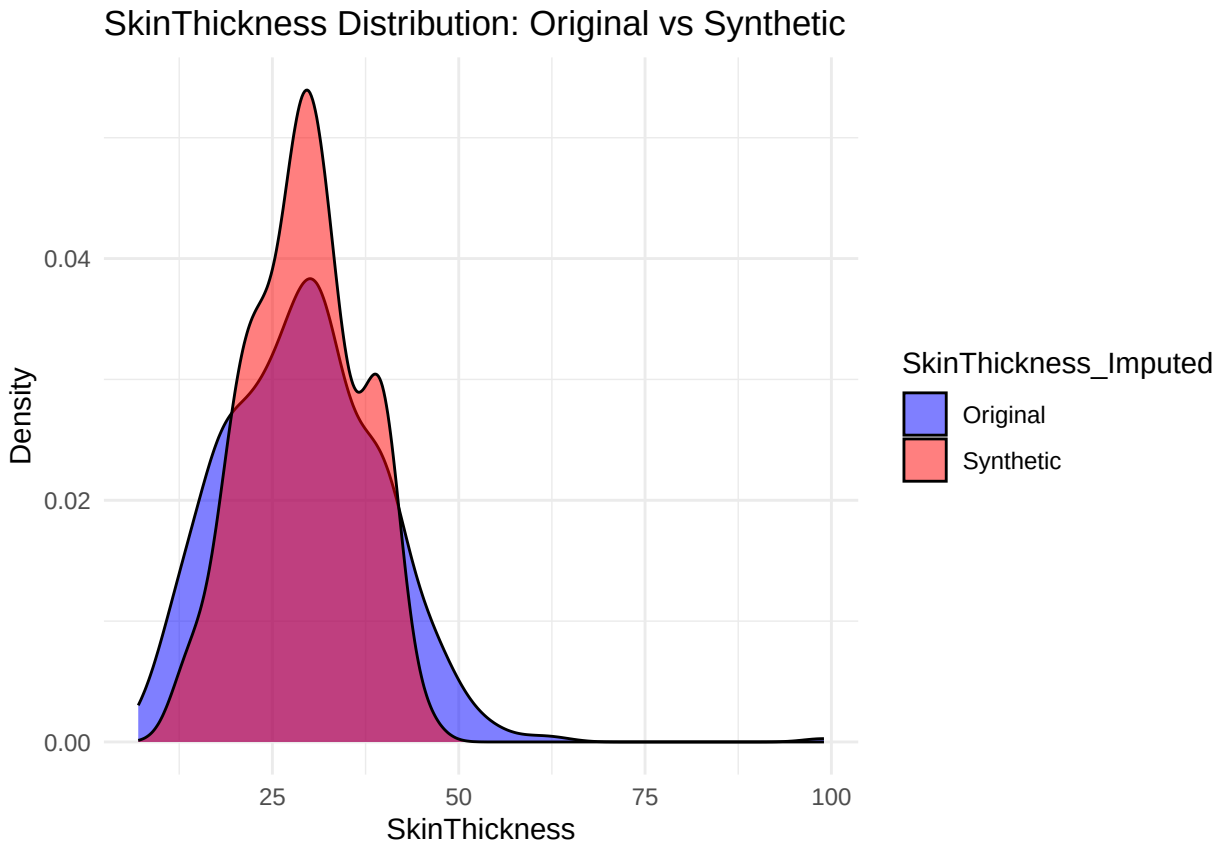

```r
# Step 2: Label Original and Imputed Data
# Create a dataset with Original vs. Synthetic categories for all features
density_data <- diabetes_data %>%
  mutate(
    Insulin_Imputed = ifelse(is.na(Insulin), "Synthetic", "Original"),
    SkinThickness_Imputed = ifelse(is.na(SkinThickness), "Synthetic", "Original"),
    Glucose_Imputed = ifelse(is.na(Glucose), "Synthetic", "Original"),
    BloodPressure_Imputed = ifelse(is.na(BloodPressure), "Synthetic", "Original"),
    BMI_Imputed = ifelse(is.na(BMI), "Synthetic", "Original")
  )

# Step 3: Impute Missing Values using kNN
synthetic_density_data <- kNN(density_data, variable = c("Insulin", "SkinThickness","BloodPressure","BM
save(synthetic_density_data, file = "synthetic_density_data.RData")

# Scatterplot Across Predictors
ggplot(synthetic_density_data, aes(x = Insulin, y = Glucose, color = as.factor(Outcome))) +
  geom_point(alpha = 0.7) +
  labs(title = "Scatterplot of Insulin vs Glucose", x = "Insulin", y = "Glucose", color = "Outcome") +
  theme_minimal()
```



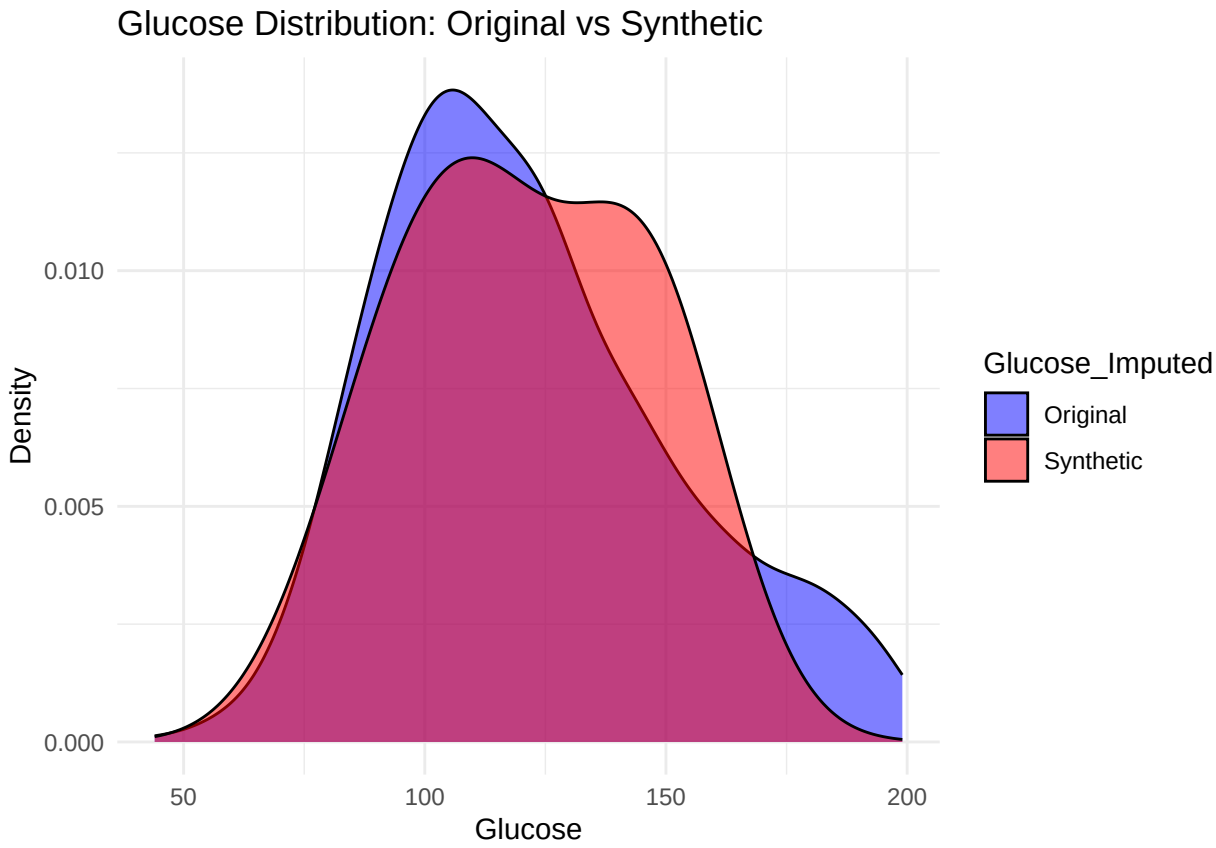

```
# Density plot for Insulin
ggplot(synthetic_density_data, aes(x = Insulin, fill = Insulin_Imputed)) +
  geom_density(alpha = 0.5, na.rm = TRUE) +
  labs(title = "Insulin Distribution: Original vs Synthetic", x = "Insulin", y = "Density") +
  scale_fill_manual(values = c("Original" = "blue", "Synthetic" = "red")) +
  theme_minimal()
```



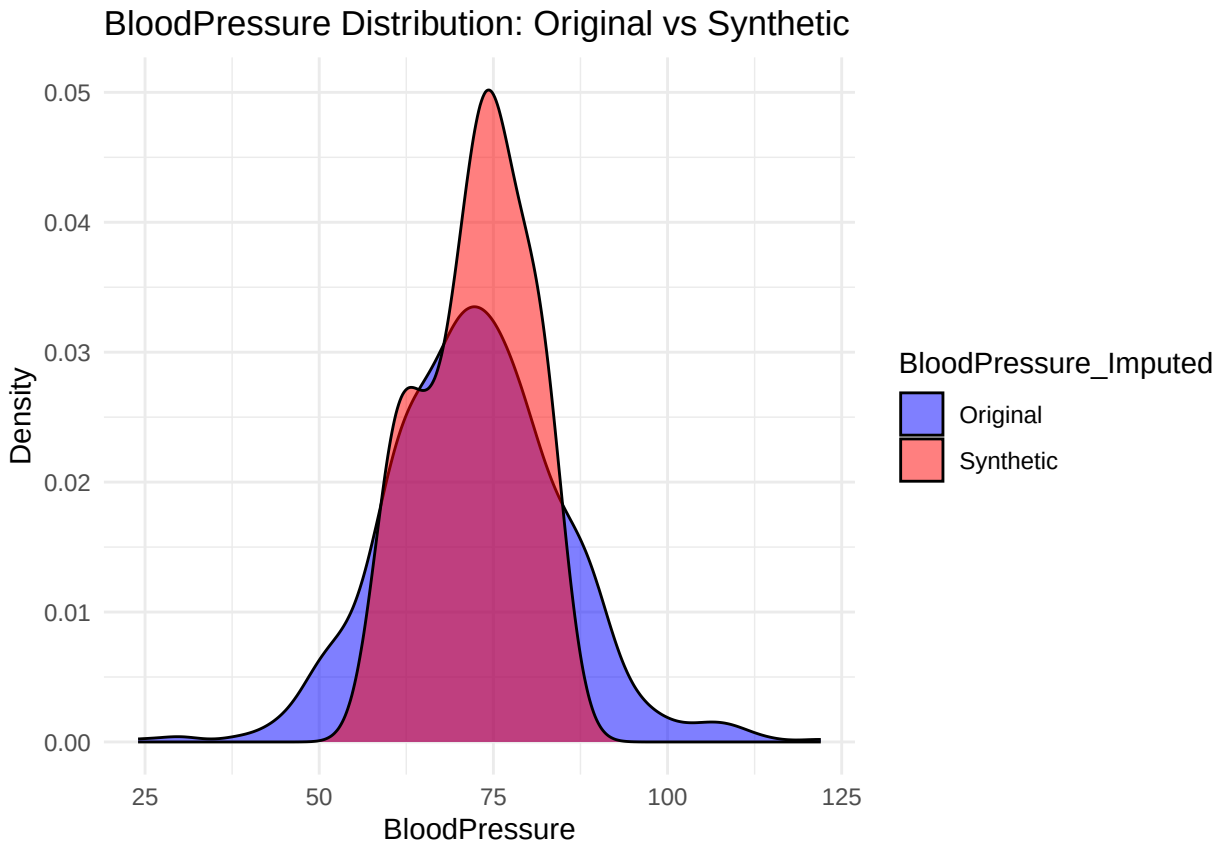

```r
# Density plot for SkinThickness
ggplot(synthetic_density_data, aes(x = SkinThickness, fill = SkinThickness_Imputed)) +
  geom_density(alpha = 0.5, na.rm = TRUE) +
  labs(title = "SkinThickness Distribution: Original vs Synthetic", x = "SkinThickness", y = "Density") +
  scale_fill_manual(values = c("Original" = "blue", "Synthetic" = "red")) +
  theme_minimal()
```



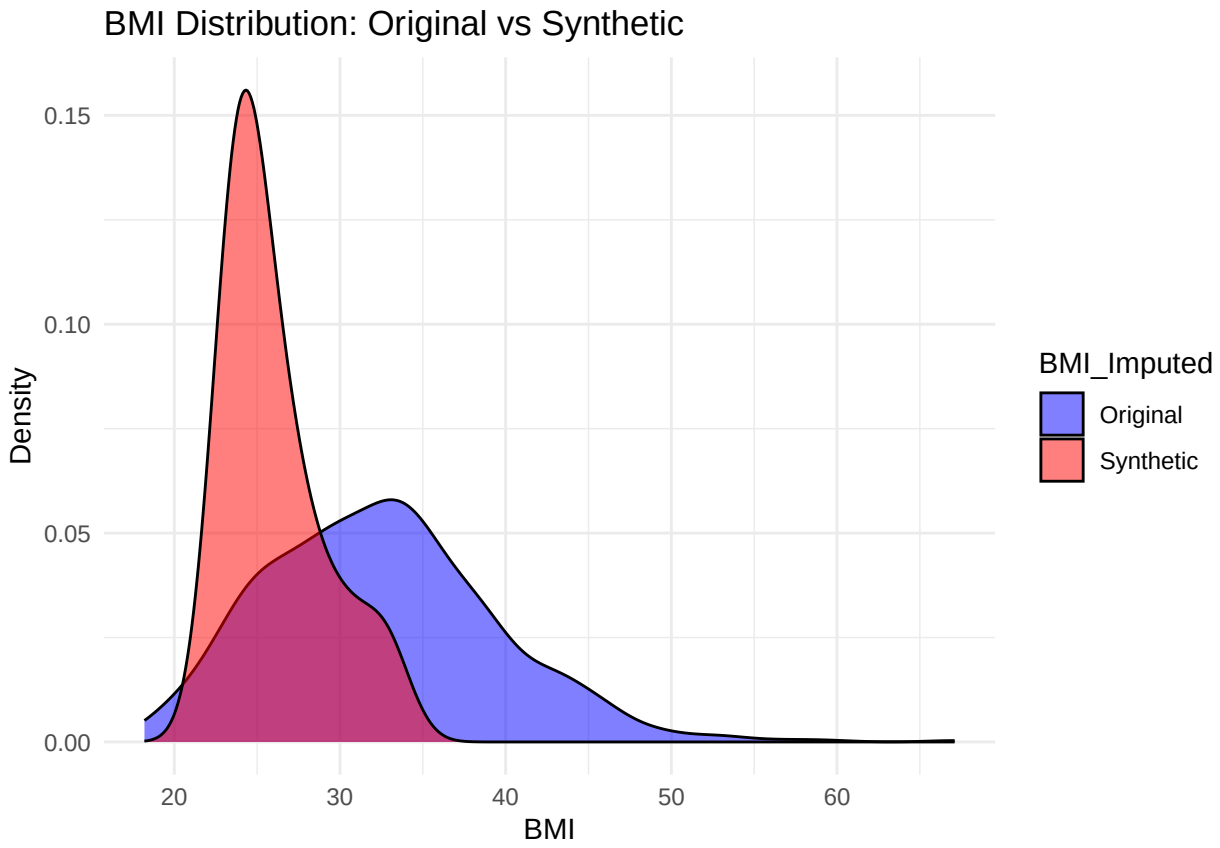

```r
# Density plot for Glucose
ggplot(synthetic_density_data, aes(x = Glucose, fill = Glucose_Imputed)) +
  geom_density(alpha = 0.5, na.rm = TRUE) +
  labs(title = "Glucose Distribution: Original vs Synthetic", x = "Glucose", y = "Density") +
  scale_fill_manual(values = c("Original" = "blue", "Synthetic" = "red")) +
  theme_minimal()
```




```r
# Density plot for BloodPressure
ggplot(synthetic_density_data, aes(x = BloodPressure, fill = BloodPressure_Imputed)) +
  geom_density(alpha = 0.5, na.rm = TRUE) +
  labs(title = "BloodPressure Distribution: Original vs Synthetic", x = "BloodPressure", y = "Density") +
  scale_fill_manual(values = c("Original" = "blue", "Synthetic" = "red")) +
  theme_minimal()
```




```r
# Density plot for BMI
ggplot(synthetic_density_data, aes(x = BMI, fill = BMI_Imputed)) +
  geom_density(alpha = 0.5, na.rm = TRUE) +
  labs(title = "BMI Distribution: Original vs Synthetic", x = "BMI", y = "Density") +
  scale_fill_manual(values = c("Original" = "blue", "Synthetic" = "red")) +
  theme_minimal()
```




```r
# Create a summary table
summary_table <- data.frame(
  Feature = names(synthetic_density_data),
  Missing_Values = sapply(synthetic_density_data, function(x) sum(is.na(x))),
  Mean = sapply(synthetic_density_data, function(x) if (is.numeric(x)) mean(x, na.rm = TRUE) else NA),
  Median = sapply(synthetic_density_data, function(x) if (is.numeric(x)) median(x, na.rm = TRUE) else NA
  Min = sapply(synthetic_density_data, function(x) if (is.numeric(x)) min(x, na.rm = TRUE) else NA),
  Max = sapply(synthetic_density_data, function(x) if (is.numeric(x)) max(x, na.rm = TRUE) else NA),
  SD = sapply(synthetic_density_data, function(x) if (is.numeric(x)) sd(x, na.rm = TRUE) else NA),
  Outliers = sapply(synthetic_density_data, function(x) if (is.numeric(x)) detect_outliers(x) else NA)
)

# Remove any irrelevant columns (like *_imp) if they exist
summary_table <- summary_table[!grepl("_imp$", summary_table$Feature), ]
summary_table <- summary_table[!grepl("_Imputed$", summary_table$Feature), ]

# Display the refined summary table
print(summary_table)
```

```
##                                                 Feature Missing_Values        Mean
## Pregnancies                                 Pregnancies              0   3.8450521
## Glucose                                         Glucose              0 121.6809896
## BloodPressure                             BloodPressure              0  72.4114583
## SkinThickness                             SkinThickness              0  29.1796875
## Insulin                                         Insulin              0 145.3255208
## BMI                                                 BMI              0  32.3654948
## DiabetesPedigreeFunction DiabetesPedigreeFunction              0   0.4718763
```

```
## Age                                      Age              0  33.2408854
## Outcome                                  Outcome          0   0.3489583
##                             Median    Min    Max         SD Outliers
## Pregnancies                  3.0000  0.000  17.00  3.3695781        4
## Glucose                    117.0000 44.000 199.00 30.4895913        0
## BloodPressure               72.0000 24.000 122.00 12.1932494       14
## SkinThickness               29.0000  7.000  99.00  9.6577146        3
## Insulin                    125.0000 14.000 846.00 96.0310998       38
## BMI                         32.0000 18.200  67.10  6.9260440        8
## DiabetesPedigreeFunction     0.3725  0.078   2.42  0.3313286       29
## Age                         29.0000 21.000  81.00 11.7602315        9
## Outcome                      0.0000  0.000   1.00  0.4769514        0
```