SP24-IN-INFO-B518-20814

Project Presentation

Diabetes Prediction Dataset

A Comprehensive Dataset for Predicting Diabetes with Medical & Demographic Data

Group 5

Adarsh Viswanath
Amol Prakash
Keerthika Sunchu
Meghana Darla
Samantha Sanjeev

Luddy School of Informatics, Computing, and Engineering

Prof. Zeyana Hamid

Introduction

Diabetes mellitus is a global health concern characterized by insufficient insulin production or utilization, leading to metabolic imbalances and beta cell deterioration. Type 1 diabetes, commonly diagnosed in children and young adults, necessitates insulin therapy. Type 2 diabetes, often linked to insulin resistance, requires lifestyle modifications and medication. Gestational diabetes, emerging during pregnancy, affects both maternal and fetal well-being (Ganie et al., 2022).

Diabetes mellitus is influenced by a combination of factors including age, BMI, HbA1C levels, and blood glucose concentrations, with each playing a pivotal role in the development and management of this metabolic disorder (Fletcher et al., 2022).

Its implications stretch beyond health, impacting socio-economic realms due to severe complications. Amidst this crisis, machine learning tools show promise in healthcare, spanning disease prediction, diagnosis, and treatment. Recent research highlights a surge in utilizing machine learning for predicting type 2 diabetes mellitus, reflecting a growing interest in leveraging technology to tackle this urgent challenge (Ganie et al., 2022).

Problem Statement

Research question:

What is the impact of age, BMI, HbA1C levels, and blood glucose concentrations, on the likelihood of developing diabetes in a diverse patient population?

Hypothesis:

Null Hypothesis: There is no significant impact of age, BMI, HbA1C levels, and blood glucose concentrations, on the likelihood of developing diabetes in a diverse patient population.

Alternate Hypothesis: There is a significant impact of age, BMI, HbA1C levels, and blood glucose concentrations, on the likelihood of developing diabetes in a diverse patient population.

Dataset Description

Data source: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/code

- We first loaded the dataset.
- This dataset comprises 100,000 rows and 9 columns.
- The dependent variable is “Diabetes” which is being predicted, with values of 1 indicating the presence of diabetes and 0 indicating the absence of diabetes.
- Our dataset comprises of both categorical and numerical variables.

```r
library(readr)
diabetes_prediction_dataset <- read_csv("C:/Users/Meghana/Downloads/diabetes_prediction_dataset.csv")
View(diabetes_prediction_dataset)
```

```
Rows: 100000 Columns: 9— Column specification
```

```r
#checking the number of rows and columns
dim(diabetes_prediction_dataset)
```

```
[1] 100000      9
```

Categorical Variables	Numerical Variables
Gender	Age
Smoking History	BMI
Hypertension	HbA1c Level
Heart Disease	Blood Glucose Level
Diabetes	

Filter

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
1	Female	80.00	0	1	never	25.19	6.6	140	0
2	Female	54.00	0	0	No Info	27.32	6.6	80	0
3	Male	28.00	0	0	never	27.32	5.7	158	0
4	Female	36.00	0	0	current	23.45	5.0	155	0
5	Male	76.00	1	1	current	20.14	4.8	155	0
6	Female	20.00	0	0	never	27.32	6.6	85	0
7	Female	44.00	0	0	never	19.31	6.5	200	1
8	Female	79.00	0	0	No Info	23.86	5.7	85	0
9	Male	42.00	0	0	never	33.64	4.8	145	0
0	Female	32.00	0	0	never	27.32	5.0	100	0
1	Female	53.00	0	0	never	27.32	6.1	85	0
2	Female	54.00	0	0	former	54.70	6.0	100	0
3	Female	78.00	0	0	former	36.05	5.0	130	0
4	Female	67.00	0	0	never	25.69	5.8	200	0
5	Female	76.00	0	0	No Info	27.32	5.0	160	0
6	Male	78.00	0	0	No Info	27.32	6.6	126	0
7	Male	15.00	0	0	never	30.36	6.1	200	0
8	Female	42.00	0	0	never	24.48	5.7	158	0
9	Female	42.00	0	0	No Info	27.32	5.7	80	0

Showing 1 to 19 of 100,000 entries, 9 total columns

Exploratory Data Analysis

The rationale behind exploratory data analysis (EDA) is to identify potential issues such as outliers or missing values, and to generate hypotheses for more in-depth analysis.

- **Checking for null values –** Confirmed that our dataset has no null values.

```
#Counting null values for each variable
```{r}
null_values_in_age <- sum(is.na(diabetes_prediction_dataset$age))
print(null_values_in_age)
null_values_in_bmi <- sum(is.na(diabetes_prediction_dataset$bmi))
print(null_values_in_bmi)
null_values_in_gender <- sum(is.na(diabetes_prediction_dataset$gender))
print(null_values_in_gender)
null_values_in_hypertension <- sum(is.na(diabetes_prediction_dataset$hypertension))
print(null_values_in_hypertension)
null_values_in_heart_disease <- sum(is.na(diabetes_prediction_dataset$heart_disease))
print(null_values_in_heart_disease)
null_values_in_smoking_history <- sum(is.na(diabetes_prediction_dataset$smoking_history))
print(null_values_in_smoking_history)
null_values_in_HbA1c_level <- sum(is.na(diabetes_prediction_dataset$HbA1c_level))
print(null_values_in_HbA1c_level)
null_values_in_blood_glucose_level <- sum(is.na(diabetes_prediction_dataset$blood_glucose_level))
print(null_values_in_blood_glucose_level)
null_values_in_diabetes <- sum(is.na(diabetes_prediction_dataset$diabetes))
print(null_values_in_diabetes)
```
```
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
```

- **Summary statistics**: To understand its numerical characteristics providing a foundation for subsequent in-depth analyses.

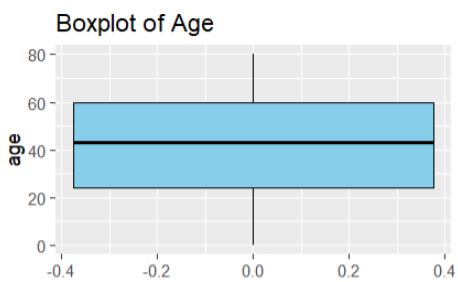
```
```{r}
# Summary of all the data attributes
summary(cleaned_dataset)

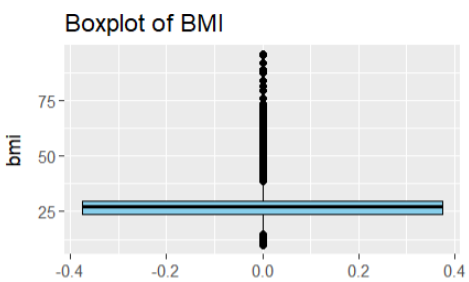
```
```

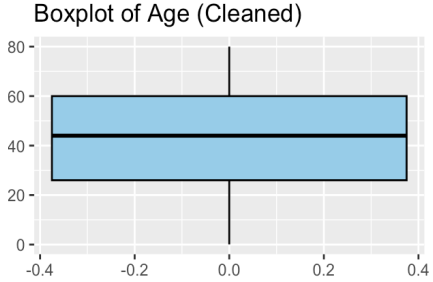
```
   gender             age         hypertension     heart_disease     smoking_history
Length:72087       Min.   : 0.08   Min.   :0.00000   Min.   :0.00000   Length:72087
Class :character   1st Qu.:26.00   1st Qu.:0.00000   1st Qu.:0.00000   Class :character
Mode  :character   Median :44.00   Median :0.00000   Median :0.00000   Mode  :character
                   Mean   :43.54   Mean   :0.06546   Mean   :0.03673
                   3rd Qu.:60.00   3rd Qu.:0.00000   3rd Qu.:0.00000
                   Max.   :80.00   Max.   :1.00000   Max.   :1.00000
      bmi         HbA1c_level    blood_glucose_level    diabetes
 Min.   :10.16   Min.   :3.500   Min.   : 80.0       Min.   :0.00000
 1st Qu.:25.15   1st Qu.:4.800   1st Qu.:100.0       1st Qu.:0.00000
 Median :27.32   Median :5.800   Median :140.0       Median :0.00000
 Mean   :26.47   Mean   :5.452   Mean   :134.4       Mean   :0.04827
 3rd Qu.:27.32   3rd Qu.:6.200   3rd Qu.:158.0       3rd Qu.:0.00000
 Max.   :31.50   Max.   :8.200   Max.   :240.0       Max.   :1.00000
```
```

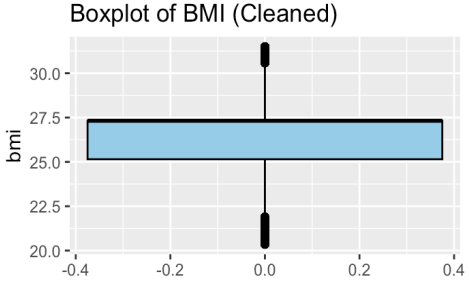
Detection of Outliers by using Box Plots Method

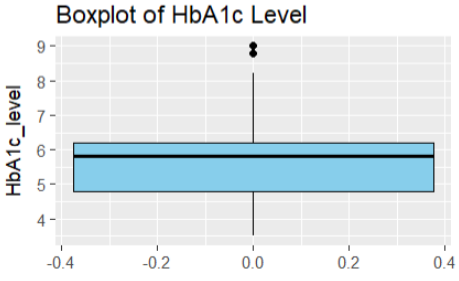
- We could detect the presence of outliers on three of our predictor variables. So, we chose to remove the outliers by using the IQR(Interquantile range).
- By comparing the boxplots before and after cleaning, we can visualize the impact of the data cleaning process on the distribution of the variables, particularly in terms of removing the effects of extreme outliers.

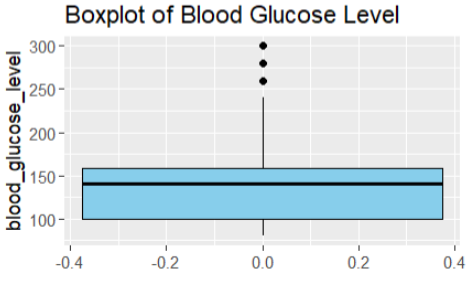


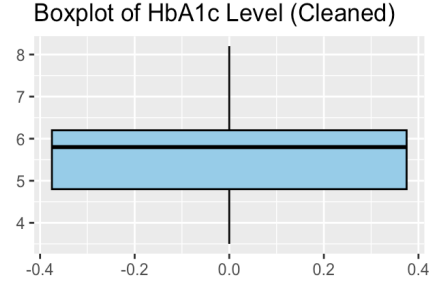


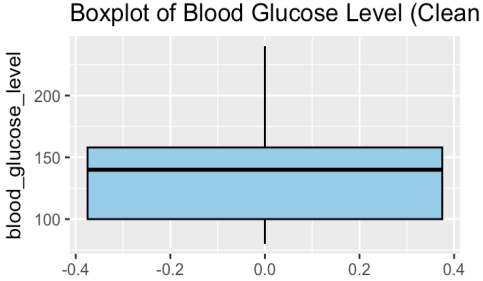






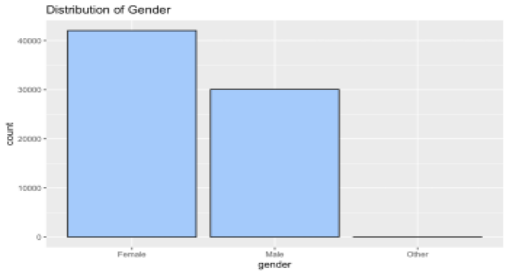


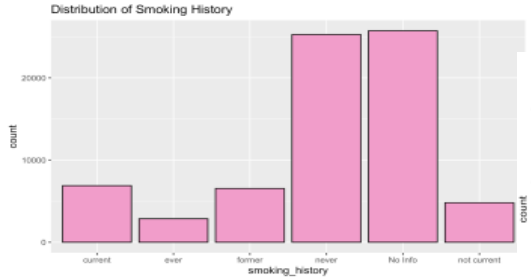


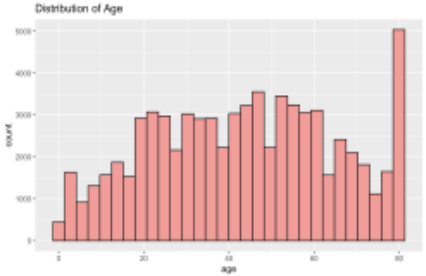


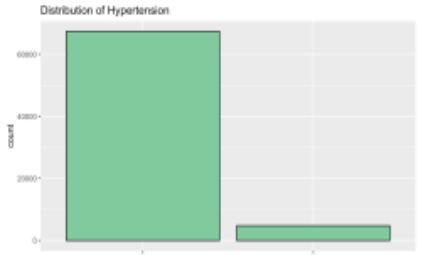
Data Visualization

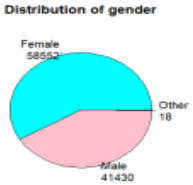
We visualized the distribution of all the variables using histograms and categorical variables using piecharts and bar graphs.

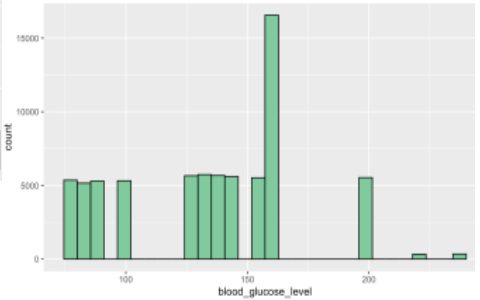


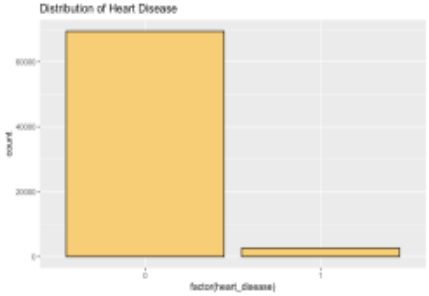


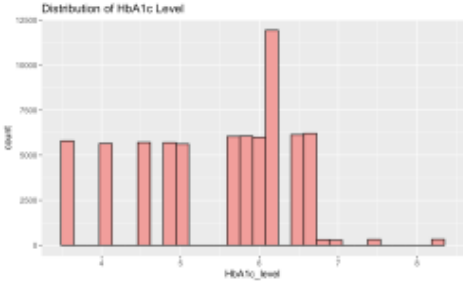


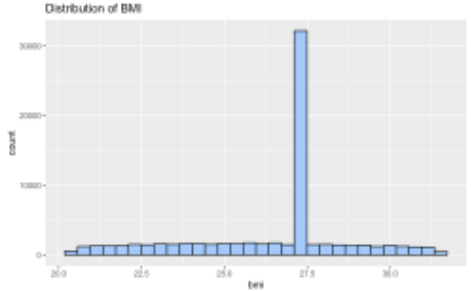












Logistic Regression

We performed logistic regression since our outcome was a binary outcome (that is, the presence or absence of diabetes) and we used it to predict its association with a set of predictor variables. We also estimated the odds ratio to represents how the likelihood of the outcome changes with a one-unit change in the predictor.

The impact of each of the variables on the outcome of diabetes revealed the following results:

- Age: Each year increases diabetes odds by 5.2% ($p < 2e-16$, AIC = 3147).
- BMI: 11% increase in diabetes odds per unit increase ($p < 2e-16$).
- HbA1c Level: Each unit raises diabetes odds by 231.6% ($p < 2e-16$).
- Blood Glucose Level: 2.8% increase per unit rise ($p < 2e-16$).

Overall Analysis: The p-value is less than 0.05 for all the variables, so we are rejecting the null hypothesis. So, there is indeed a significant association between the independent variables and diabetes in our dataset.

Logistic Regression

```
Call:
glm(formula = diabetes ~ age, family = binomial(), data = cleaned_dataset)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.6094158  0.0567152  -98.91   <2e-16 ***
age          0.0511086  0.0008952   57.09   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35743  on 90386  degrees of freedom
Residual deviance: 31467  on 90385  degrees of freedom
AIC: 31471

Number of Fisher Scoring iterations: 7

(Intercept)         age
0.003663209 1.052437211
```

```
Call:
glm(formula = diabetes ~ HbA1c_level, family = binomial(), data = cleaned_dataset)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -18.09420    0.24560  -73.67   <2e-16 ***
HbA1c_level   2.48140    0.03838   64.66   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35743  on 90386  degrees of freedom
Residual deviance: 26878  on 90385  degrees of freedom
AIC: 26882

Number of Fisher Scoring iterations: 8

  (Intercept)  HbA1c_level
1.386075e-08 1.195798e+01
```

```
Call:
glm(formula = diabetes ~ bmi, family = binomial(), data = cleaned_dataset)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.814424   0.098163  -69.42   <2e-16 ***
bmi          0.139252   0.003331   41.80   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35743  on 90386  degrees of freedom
Residual deviance: 33909  on 90385  degrees of freedom
AIC: 33913

Number of Fisher Scoring iterations: 6

(Intercept)         bmi
0.001097825 1.149413415
```

```
Call:
glm(formula = diabetes ~ blood_glucose_level, family = binomial(),
    data = cleaned_dataset)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         -7.3430456  0.0803470  -91.39   <2e-16 ***
blood_glucose_level  0.0292561  0.0004806   60.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35743  on 90386  degrees of freedom
Residual deviance: 31587  on 90385  degrees of freedom
AIC: 31591

Number of Fisher Scoring iterations: 6

        (Intercept) blood_glucose_level
       0.0006470768        1.0296883026
```

Logistic Regression

We also performed multivariate logistic regression to understand the effect of each predictor on the outcome while adjusting for the influence of other variables in the model.

Interpretation:

- The initial AIC value of 19847.2 is the baseline for the full model.
- We could interpret that all predictor variables are important for predicting diabetes, as removing any of them increases the AIC, which indicates a worse model fit.
- The variables with the largest impact on AIC when removed are HbA1c level and blood glucose level, suggesting they are the most important predictors of diabetes in the model.

Overall analysis: The p-value is less than 0.05 for all the variables, so we are rejecting the null hypothesis.
Older age, higher BMI, higher HbA1c level, and higher blood glucose level are associated with an increased risk of diabetes.

```
Start:  AIC=19847.2
diabetes ~ age + bmi + HbA1c_level + blood_glucose_level

                     Df Deviance   AIC
<none>                    19837 19847
- bmi                 1    20551 20559
- age                 1    22389 22397
- blood_glucose_level 1    22569 22577
- HbA1c_level         1    26710 26718

Call:
glm(formula = diabetes ~ age + bmi + HbA1c_level + blood_glucose_level,
    family = binomial(), data = cleaned_dataset)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -2.728e+01  3.361e-01  -81.17   <2e-16 ***
age                 5.140e-02  1.144e-03   44.91   <2e-16 ***
bmi                 1.198e-01  4.550e-03   26.33   <2e-16 ***
HbA1c_level         2.322e+00  4.105e-02   56.57   <2e-16 ***
blood_glucose_level 2.774e-02  5.835e-04   47.54   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35743  on 90386  degrees of freedom
Residual deviance: 19837  on 90382  degrees of freedom
AIC: 19847

Number of Fisher Scoring iterations: 8
```

Normality Testing - Histograms

- We could not perform Shapiro-Wilk test for assessing the normal distribution of our data since our sample size was greater than 5000.
- We then performed the normal curve visualization of our predictor variables using histograms to see the data symmetry.
- Interpretation: These histograms visually reveal that the data is not normally distributed.

```
#Normality Testing
```{r}
# Perform Shapiro-Wilk test on various columns
shapiro_age <- shapiro.test(cleaned_dataset$age)
shapiro_bmi <- shapiro.test(cleaned_dataset$bmi)
shapiro_hba1c <- shapiro.test(cleaned_dataset$HbA1c_level)
shapiro_glucose <- shapiro.test(cleaned_dataset$blood_glucose_level)

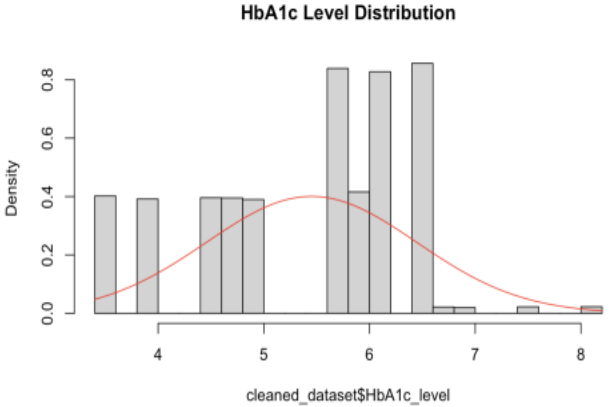
# Print the results
print("Shapiro-Wilk Test for Age:")
print(shapiro_age)

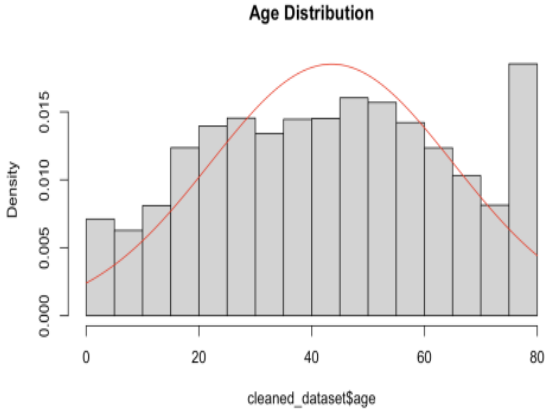
print("Shapiro-Wilk Test for BMI:")
print(shapiro_bmi)

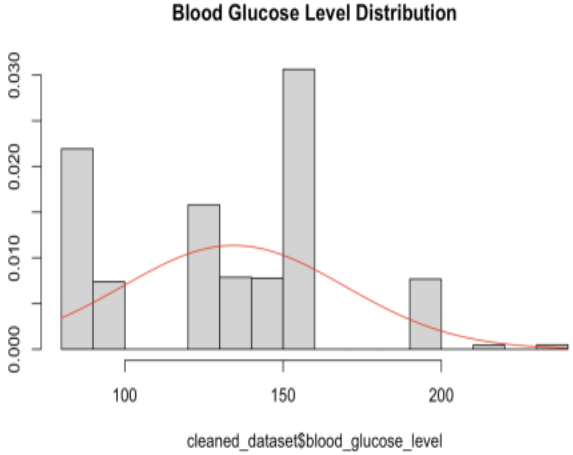
print("Shapiro-Wilk Test for HbA1c Level:")
print(shapiro_hba1c)

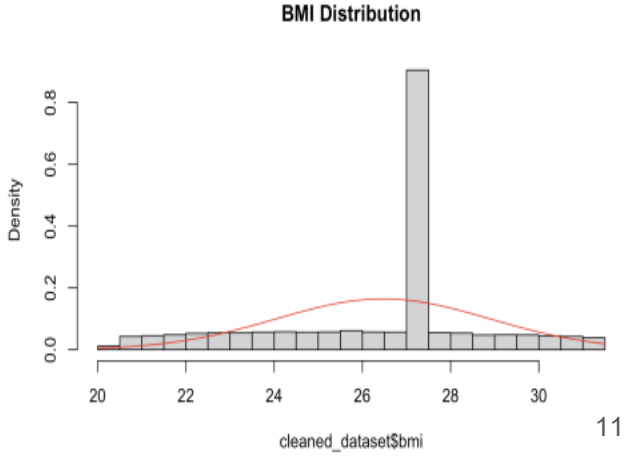
print("Shapiro-Wilk Test for Blood Glucose Level:")
print(shapiro_glucose)
```
```

```
Error in shapiro.test(cleaned_dataset$age) :
  sample size must be between 3 and 5000
```





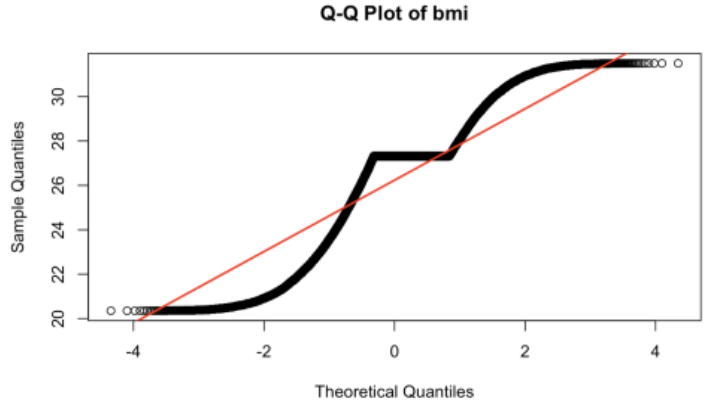


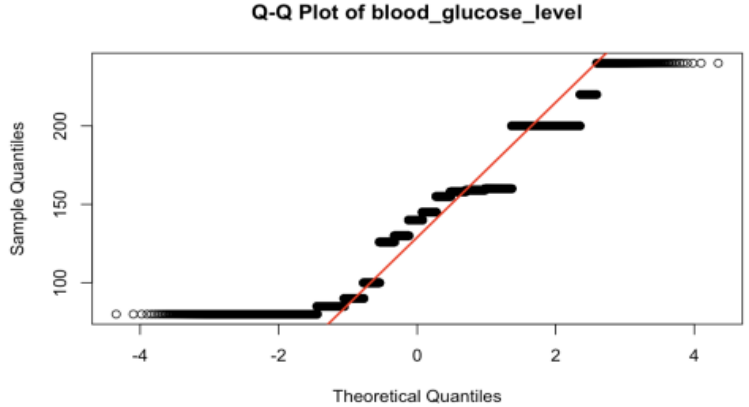


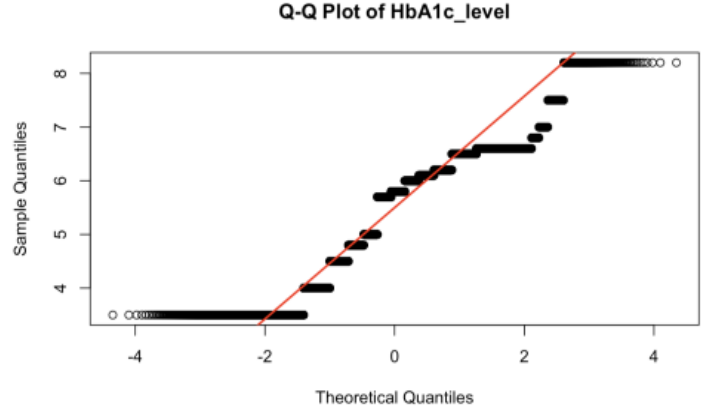
Normality Testing - Q-Q Plots

We also used Q-Q plots to visualize the normality of our data.

- Interpretation: The Q-Q plots substantiate that the data is not normally distributed for each variable.







Non - Parametric Test- Kruskal-Wallis Test

- We decided to perform non - parametric tests, since our data is not uniformly distributed.

- Interpretation: The p-values for all tests are extremely small (less than 2.2e-16), suggesting strong evidence against the null hypothesis.

- These results support the alternative hypothesis, indicating that age, BMI, HbA1c levels, and blood glucose concentrations significantly influence the likelihood of developing diabetes in our diverse patient population.

```
        Kruskal-Wallis rank sum test

data:  age by diabetes
Kruskal-Wallis chi-squared = 3963.6, df = 1, p-value < 2.2e-16
```

```
        Kruskal-Wallis rank sum test

data:  bmi by diabetes
Kruskal-Wallis chi-squared = 1639.3, df = 1, p-value < 2.2e-16
```

```
        Kruskal-Wallis rank sum test

data:  HbA1c_level by diabetes
Kruskal-Wallis chi-squared = 4941, df = 1, p-value < 2.2e-16
```

```
        Kruskal-Wallis rank sum test

data:  blood_glucose_level by diabetes
Kruskal-Wallis chi-squared = 2415.4, df = 1, p-value < 2.2e-16
```

Dunn's Test (Post-hoc test)

- Dunn's test is a non-parametric method used as a post-hoc test following a Kruskal-Wallis test to determine which groups differ from each other.
- Post-hoc analysis reveals significant differences in each variable between all pairs of diabetes levels ($p < 0.05$), supporting the alternate hypothesis.
- The results obtained from the analysis strongly support the alternate hypothesis, suggesting that there is indeed a significant impact of age, BMI, HbA1C levels, and blood glucose concentrations on the likelihood of developing diabetes in our diverse patient population.

```
        Kruskal-Wallis rank sum test

data:  age by diabetes
Kruskal-Wallis chi-squared = 3963.6, df = 1, p-value < 2.2e-16

Warning: Ties are present. z-quantiles were corrected for ties.
        Pairwise comparisons using Dunn's all-pairs test

data: cleaned_dataset$age and cleaned_dataset$diabetes

  0
1 <2e-16

P value adjustment method: holm
alternative hypothesis: two.sided
Warning: Ties are present. z-quantiles were corrected for ties.Warning: Ties are present. z-quantiles were corrected
for ties.Warning: Ties are present. z-quantiles were corrected for ties.
        Pairwise comparisons using Dunn's all-pairs test

data: cleaned_dataset$bmi and cleaned_dataset$diabetes

  0
1 <2e-16

P value adjustment method: holm
alternative hypothesis: two.sided

        Pairwise comparisons using Dunn's all-pairs test

data: cleaned_dataset$HbA1c_level and cleaned_dataset$diabetes

  0
1 <2e-16

P value adjustment method: holm
alternative hypothesis: two.sided

        Pairwise comparisons using Dunn's all-pairs test

data: cleaned_dataset$blood_glucose_level and cleaned_dataset$diabetes

  0
1 <2e-16

P value adjustment method: holm
alternative hypothesis: two.sided
```

Limitations

- We could not perform Shapiro-Wilk test for assessing the normal distribution of our data since our sample size was greater than 5000.

- Because the Kruskal-Wallis test ranks data rather than using the actual data values, some information about the magnitude of differences between groups is lost. This can lead to less precise estimates compared to parametric tests when the assumptions of those tests are met.

Conclusion

- Our statistical analysis using tests like logistic regression and Kruskal Wallis clearly shows that age, BMI, HbA1c levels, and blood glucose concentrations significantly impact the likelihood of developing diabetes in a diverse patient population ($p < 2e-16$).
- We found that each of these factors substantially increases diabetes risk, leading us to **reject the null hypothesis and accept the alternate hypothesis,** that is these factors indeed influence diabetes development. This underscores the importance of targeting these modifiable risk factors through personalized healthcare interventions and public health policies to effectively manage and prevent diabetes.

References

Fletcher, B., Gulanick, M., & Lamendola, C. (2002). Risk factors for type 2 diabetes mellitus. *The Journal of Cardiovascular Nursing*, *16*(2), 17–23. https://doi.org/10.1097/00005082-200201000-00003

Ganie, S. M., Malik, M. B., & Arif, T. (2022). Performance analysis and prediction of type 2 diabetes mellitus based on lifestyle data using machine learning approaches. *Journal of Diabetes and Metabolic Disorders, 21*(1), 339–352. https://doi.org/10.1007/s40200-022-00981-w

Appendix

Codes for loading and exploring the dataset

```{r}
library(readr)
diabetes_prediction_dataset <- read_csv("C:/Users/Meghana/Downloads/diabetes_prediction_dataset.csv")
View(diabetes_prediction_dataset)
```

```
Rows: 100000 Columns: 9── Column specification
```

```{r}
#checking the number of rows and columns
dim(diabetes_prediction_dataset)
```

```{r}
#Viewing the first few rows of the dataset
head(diabetes_prediction_dataset)
```

```{r}
column_names = names(diabetes_prediction_dataset)
print(column_names)
```

```{r}
#checking for datastructures of the dataset
str(diabetes_prediction_dataset)
```

Code for calculating the summary statistics of the dataset

```r
# Summary of all the data attributes
summary(diabetes_prediction_dataset)
```

Code for counting null values for each variable

```r
#Counting null values for each variable
null_values_in_age <- sum(is.na(diabetes_prediction_dataset$age))
print(null_values_in_age)
null_values_in_bmi <- sum(is.na(diabetes_prediction_dataset$bmi))
print(null_values_in_bmi)
null_values_in_gender <- sum(is.na(diabetes_prediction_dataset$gender))
print(null_values_in_gender)
null_values_in_hypertension <- sum(is.na(diabetes_prediction_dataset$hypertension))
print(null_values_in_hypertension)
null_values_in_heart_disease <- sum(is.na(diabetes_prediction_dataset$heart_disease))
print(null_values_in_heart_disease)
null_values_in_smoking_history <- sum(is.na(diabetes_prediction_dataset$smoking_history))
print(null_values_in_smoking_history)
null_values_in_HbA1c_level <- sum(is.na(diabetes_prediction_dataset$HbA1c_level))
print(null_values_in_HbA1c_level)
null_values_in_blood_glucose_level <- sum(is.na(diabetes_prediction_dataset$blood_glucose_level))
print(null_values_in_blood_glucose_level)
null_values_in_diabetes <- sum(is.na(diabetes_prediction_dataset$diabetes))
print(null_values_in_diabetes)
```

Code for checking outliers

```r
#Checking for outliers
```{r}
# Install patchwork package if not already installed
install.packages("patchwork")

# Load necessary libraries
library(ggplot2)
library(patchwork)

# Create boxplots for continuous variables
boxplot_age <- ggplot(diabetes_prediction_dataset, aes(y = age)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of Age")

boxplot_bmi <- ggplot(diabetes_prediction_dataset, aes(y = bmi)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of BMI")

boxplot_HbA1c <- ggplot(diabetes_prediction_dataset, aes(y = HbA1c_level)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of HbA1c Level")

boxplot_blood_glucose <- ggplot(diabetes_prediction_dataset, aes(y = blood_glucose_level)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of Blood Glucose Level")

# Arrange boxplots using patchwork
arranged_plots <- boxplot_age + boxplot_bmi + boxplot_HbA1c + boxplot_blood_glucose
arranged_plots <- arranged_plots + plot_layout(ncol = 2)

# Print the arranged plots
arranged_plots
```
```

Code for displaying boxplots with outliers

```
# Calculate and print outliers for each variable
outliers_age <- boxplot.stats(diabetes_prediction_dataset$age)$out
outliers_bmi <- boxplot.stats(diabetes_prediction_dataset$bmi)$out
outliers_HbA1c <- boxplot.stats(diabetes_prediction_dataset$HbA1c_level)$out
outliers_blood_glucose <- boxplot.stats(diabetes_prediction_dataset$blood_glucose_level)$out

print("Outliers in Age:")
print(outliers_age)

print("Outliers in BMI:")
print(outliers_bmi)

print("Outliers in HbA1c Level:")
print(outliers_HbA1c)

print("Outliers in Blood Glucose Level:")
print(outliers_blood_glucose)
```

Codes for removing the outliers

```r
#Removing the outliers
```{r}
# Load the dplyr package
library(dplyr)

# Calculate the IQR for HbA1c level
Q1_HbA1c <- quantile(diabetes_prediction_dataset$HbA1c_level, 0.25, na.rm = TRUE)  # First Quartile (25th
percentile)
Q3_HbA1c <- quantile(diabetes_prediction_dataset$HbA1c_level, 0.75, na.rm = TRUE)  # Third Quartile (75th
percentile)
IQR_HbA1c <- Q3_HbA1c - Q1_HbA1c  # Interquartile Range

# Define outlier thresholds
lower_bound_HbA1c <- Q1_HbA1c - 1.5 * IQR_HbA1c
upper_bound_HbA1c <- Q3_HbA1c + 1.5 * IQR_HbA1c

# Filter out outliers for HbA1c level
cleaned_dataset <- diabetes_prediction_dataset %>%
  filter(HbA1c_level >= lower_bound_HbA1c & HbA1c_level <= upper_bound_HbA1c)

# Calculate the IQR for blood glucose level
Q1_blood_glucose <- quantile(diabetes_prediction_dataset$blood_glucose_level, 0.25, na.rm = TRUE)  # First
Quartile (25th percentile)
Q3_blood_glucose <- quantile(diabetes_prediction_dataset$blood_glucose_level, 0.75, na.rm = TRUE)  # Third
Quartile (75th percentile)
IQR_blood_glucose <- Q3_blood_glucose - Q1_blood_glucose  # Interquartile Range

# Define outlier thresholds
lower_bound_blood_glucose <- Q1_blood_glucose - 1.5 * IQR_blood_glucose
upper_bound_blood_glucose <- Q3_blood_glucose + 1.5 * IQR_blood_glucose

# Filter out outliers for blood glucose level
cleaned_dataset <- cleaned_dataset %>%
  filter(blood_glucose_level >= lower_bound_blood_glucose & blood_glucose_level <= upper_bound_blood_glucose)
```
```

Contd. Code for removing outliers

```
# Calculate the IQR for BMI
Q1_bmi <- quantile(diabetes_prediction_dataset$bmi, 0.25, na.rm = TRUE)  # First Quartile (25th percentile)
Q3_bmi <- quantile(diabetes_prediction_dataset$bmi, 0.75, na.rm = TRUE)  # Third Quartile (75th percentile)
IQR_bmi <- Q3_bmi - Q1_bmi  # Interquartile Range

# Define outlier thresholds
lower_bound_bmi <- Q1_bmi - 1.5 * IQR_bmi
upper_bound_bmi <- Q3_bmi + 1.5 * IQR_bmi

# Filter out outliers for BMI
cleaned_dataset <- cleaned_dataset %>%
  filter(bmi >= lower_bound_bmi & bmi <= upper_bound_bmi)

# View the dimensions of the cleaned dataset to confirm rows are dropped
dim(cleaned_dataset)

```

Code for boxplots after removing outliers

```r
# Load necessary libraries
library(ggplot2)
library(patchwork)
# Re-create the boxplots for the cleaned continuous variables
boxplot_age_clean <- ggplot(cleaned_dataset, aes(y = age)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of Age (Cleaned)")

boxplot_bmi_clean <- ggplot(cleaned_dataset, aes(y = bmi)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of BMI (Cleaned)")

boxplot_HbA1c_clean <- ggplot(cleaned_dataset, aes(y = HbA1c_level)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of HbA1c Level (Cleaned)")

boxplot_blood_glucose_clean <- ggplot(cleaned_dataset, aes(y = blood_glucose_level)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot of Blood Glucose Level (Cleaned)")

# Arrange the cleaned boxplots using patchwork
arranged_plots_clean <- boxplot_age_clean + boxplot_bmi_clean + boxplot_HbA1c_clean + boxplot_blood_glucose_clean
arranged_plots_clean <- arranged_plots_clean + plot_layout(ncol = 2)

# Print the arranged plots
arranged_plots_clean
```

Codes for calculating summary statistics for cleaned dataset

```
#Calculating summary statistics for the cleaned dataset
```{r}
#checking the number of rows and columns
dim(cleaned_dataset)
```
```

```
```{r}
head(cleaned_dataset)

```
```

```
```{r}
# Summary of all the data attributes
summary(cleaned_dataset)

```
```

Codes for calculation frequency and percentage of each variable, along with central tendency and measures of dispersion

```
#Gender
```{r}
# Count or frequency of each gender
gender_frequency <- table(cleaned_dataset$gender)

# Displaying frequency of each gender
print("Frequency of each gender:")
print(gender_frequency)

# Percentage or proportion of each gender
gender_percentage <- prop.table(gender_frequency) * 100

# Displaying percentage or proportion of each gender
print("Percentage of each gender:")
print(gender_percentage)

```
```

```
#Hypertension
```{r}
# Count or frequency of hypertension
hypertension_frequency <- table(cleaned_dataset$hypertension)
# Percentage or proportion of hypertension
hypertension_percentage <- prop.table(hypertension_frequency) * 100

# Displaying frequency and percentage of hypertension
print("Frequency of Hypertension:")
print(hypertension_frequency)
print("Percentage of Hypertension:")
print(hypertension_percentage)
```
```

#Age

```{r}
# Measures of Central Tendency
mean_age <- mean(cleaned_dataset$age)
median_age <- median(cleaned_dataset$age)

# Displaying Measures of Central Tendency
print("Measures of Central Tendency:")
print(paste("Mean Age:", mean_age))
print(paste("Median Age:", median_age))

# Measures of Dispersion/Spread
range_age <- range(cleaned_dataset$age)
variance_age <- var(cleaned_dataset$age)
sd_age <- sd(cleaned_dataset$age)
iqr_age <- IQR(cleaned_dataset$age)

# Displaying Measures of Dispersion/Spread
print("Measures of Dispersion/Spread:")
print(paste("Range of Age:", range_age))
print(paste("Variance of Age:", variance_age))
print(paste("Standard Deviation of Age:", sd_age))
print(paste("Interquartile Range of Age:", iqr_age))

# Minimum and Maximum values
min_age <- min(cleaned_dataset$age)
max_age <- max(cleaned_dataset$age)

# Displaying Minimum and Maximum values
print("Minimum and Maximum Age:")
print(paste("Minimum Age:", min_age))
print(paste("Maximum Age:", max_age))

# Percentiles
percentiles_age <- quantile(cleaned_dataset$age, c(0.25, 0.5, 0.75))

print("Percentiles of Age:")
```

#Heart Disease

```{r}
# Count or frequency of heart disease
heart_disease_frequency <- table(cleaned_dataset$heart_disease)
# Percentage or proportion of heart disease
heart_disease_percentage <- prop.table(heart_disease_frequency) * 100

# Displaying frequency and percentage of heart disease
print("Frequency of Heart Disease:")
print(heart_disease_frequency)
print("Percentage of Heart Disease:")
print(heart_disease_percentage)
```

#Smoking History

```{r}
# Count or frequency of smoking history
smoking_history_frequency <- table(cleaned_dataset$smoking_history)
# Percentage or proportion of smoking history
smoking_history_percentage <- prop.table(smoking_history_frequency) * 100

# Displaying frequency and percentage of smoking history
print("Frequency of Smoking History:")
print(smoking_history_frequency)
print("Percentage of Smoking History:")
print(smoking_history_percentage)
```

```r
# Measures of Central Tendency for BMI
mean_bmi <- mean(cleaned_dataset$bmi)
median_bmi <- median(cleaned_dataset$bmi)

# Displaying Measures of Central Tendency for BMI
print("Measures of Central Tendency for BMI:")
print(paste("Mean BMI:", mean_bmi))
print(paste("Median BMI:", median_bmi))

# Measures of Dispersion/Spread for BMI
range_bmi <- range(cleaned_dataset$bmi)
variance_bmi <- var(cleaned_dataset$bmi)
sd_bmi <- sd(cleaned_dataset$bmi)
iqr_bmi <- IQR(cleaned_dataset$bmi)

# Displaying Measures of Dispersion/Spread for BMI
print("Measures of Dispersion/Spread for BMI:")
print(paste("Range of BMI:", range_bmi))
print(paste("Variance of BMI:", variance_bmi))
print(paste("Standard Deviation of BMI:", sd_bmi))
print(paste("Interquartile Range of BMI:", iqr_bmi))

# Minimum and Maximum values for BMI
min_bmi <- min(cleaned_dataset$bmi)
max_bmi <- max(cleaned_dataset$bmi)

# Displaying Minimum and Maximum values for BMI
print("Minimum and Maximum BMI:")
print(paste("Minimum BMI:", min_bmi))
print(paste("Maximum BMI:", max_bmi))

# Percentiles for BMI
percentiles_bmi <- quantile(cleaned_dataset$bmi, c(0.25, 0.5, 0.75))

# Displaying Percentiles for BMI
print("Percentiles of BMI:")
print(percentiles_bmi)
```

```r
# Measures of Central Tendency for HbA1c Level
mean_hba1c <- mean(cleaned_dataset$HbA1c_level)
median_hba1c <- median(cleaned_dataset$HbA1c_level)

# Displaying Measures of Central Tendency for HbA1c Level
print("Measures of Central Tendency for HbA1c Level:")
print(paste("Mean HbA1c Level:", mean_hba1c))
print(paste("Median HbA1c Level:", median_hba1c))

# Measures of Dispersion/Spread for HbA1c Level
range_hba1c <- range(cleaned_dataset$HbA1c_level)
variance_hba1c <- var(cleaned_dataset$HbA1c_level)
sd_hba1c <- sd(cleaned_dataset$HbA1c_level)
iqr_hba1c <- IQR(cleaned_dataset$HbA1c_level)

# Displaying Measures of Dispersion/Spread for HbA1c Level
print("Measures of Dispersion/Spread for HbA1c Level:")
print(paste("Range of HbA1c Level:", range_hba1c))
print(paste("Variance of HbA1c Level:", variance_hba1c))
print(paste("Standard Deviation of HbA1c Level:", sd_hba1c))
print(paste("Interquartile Range of HbA1c Level:", iqr_hba1c))

# Minimum and Maximum values for HbA1c Level
min_hba1c <- min(cleaned_dataset$HbA1c_level)
max_hba1c <- max(cleaned_dataset$HbA1c_level)

# Displaying Minimum and Maximum values for HbA1c Level
print("Minimum and Maximum HbA1c Level:")
print(paste("Minimum HbA1c Level:", min_hba1c))
print(paste("Maximum HbA1c Level:", max_hba1c))

# Percentiles for HbA1c Level
percentiles_hba1c <- quantile(cleaned_dataset$HbA1c_level, c(0.25, 0.5, 0.75))

# Displaying Percentiles for HbA1c Level
print("Percentiles of HbA1c Level:")
print(percentiles_hba1c)
```

```r
# Measures of Central Tendency for Blood Glucose Level
mean_blood_glucose <- mean(cleaned_dataset$blood_glucose_level)
median_blood_glucose <- median(cleaned_dataset$blood_glucose_level)

# Displaying Measures of Central Tendency for Blood Glucose Level
print("Measures of Central Tendency for Blood Glucose Level:")
print(paste("Mean Blood Glucose Level:", mean_blood_glucose))
print(paste("Median Blood Glucose Level:", median_blood_glucose))

# Measures of Dispersion/Spread for Blood Glucose Level
range_blood_glucose <- range(cleaned_dataset$blood_glucose_level)
variance_blood_glucose <- var(cleaned_dataset$blood_glucose_level)
sd_blood_glucose <- sd(cleaned_dataset$blood_glucose_level)
iqr_blood_glucose <- IQR(cleaned_dataset$blood_glucose_level)

# Displaying Measures of Dispersion/Spread for Blood Glucose Level
print("Measures of Dispersion/Spread for Blood Glucose Level:")
print(paste("Range of Blood Glucose Level:", range_blood_glucose))
print(paste("Variance of Blood Glucose Level:", variance_blood_glucose))
print(paste("Standard Deviation of Blood Glucose Level:", sd_blood_glucose))
print(paste("Interquartile Range of Blood Glucose Level:", iqr_blood_glucose))

# Minimum and Maximum values for Blood Glucose Level
min_blood_glucose <- min(cleaned_dataset$blood_glucose_level)
max_blood_glucose <- max(cleaned_dataset$blood_glucose_level)

# Displaying Minimum and Maximum values for Blood Glucose Level
print("Minimum and Maximum Blood Glucose Level:")
print(paste("Minimum Blood Glucose Level:", min_blood_glucose))
print(paste("Maximum Blood Glucose Level:", max_blood_glucose))

# Percentiles for Blood Glucose Level
percentiles_blood_glucose <- quantile(cleaned_dataset$blood_glucose_level, c(0.25, 0.5, 0.75))

# Displaying Percentiles for Blood Glucose Level
print("Percentiles of Blood Glucose Level:")
print(percentiles_blood_glucose)
```

#Diabetes

```r
# Count or frequency of diabetes
diabetes_frequency <- table(cleaned_dataset$diabetes)
print("Frequency of Diabetes:")
print(diabetes_frequency)

# Percentage or proportion of diabetes
diabetes_percentage <- prop.table(diabetes_frequency) * 100
print("Percentage of Diabetes:")
print(diabetes_percentage)
```

Codes for visualization

```
#Visualization for the cleaned dataset
```{r}
library(ggplot2)
# Gender (Categorical Variable)
ggplot(cleaned_dataset, aes(x = gender)) +
  geom_bar(fill = "#99CCFF", color = "black") +  # Custom color for bars
  labs(title = "Distribution of Gender")

# Age (Continuous Variable)
ggplot(cleaned_dataset, aes(x = age)) +
  geom_histogram(fill = "#FF9999", color = "black", bins = 30) +  # Custom color for bars
  labs(title = "Distribution of Age")

# Hypertension (Categorical Variable)
ggplot(cleaned_dataset, aes(x = factor(hypertension))) +
  geom_bar(fill = "#66CC99", color = "black") +  # Custom color for bars
  labs(title = "Distribution of Hypertension")

# Heart Disease (Categorical Variable)
ggplot(cleaned_dataset, aes(x = factor(heart_disease))) +
  geom_bar(fill = "#FFCC66", color = "black") +  # Custom color for bars
  labs(title = "Distribution of Heart Disease")

# Smoking History (Categorical Variable)
ggplot(cleaned_dataset, aes(x = smoking_history)) +
  geom_bar(fill = "#FF99CC", color = "black") +  # Custom color for bars
  labs(title = "Distribution of Smoking History")
```
```

Codes for visualization

```r
# Create a pie chart for gender
pie(table(diabetes_prediction_dataset$gender),
    main = "Distribution of gender",
    col = c("cyan", "pink"),
    labels = paste(names(table(diabetes_prediction_dataset$gender)), "\n",
table(diabetes_prediction_dataset$gender), sep = ""))
```

```r
# Create a pie chart for Smoking history
pie(table(cleaned_dataset$smoking_history),
    main = "Distribution of smoking history",
    col = c("lightblue", "pink", "yellow", "black", "cyan", "green"),
    labels = paste(names(table(cleaned_dataset$smoking_history)), "\n", table(cleaned_dataset$smoking_history),
sep = ""))
```

```r
# BMI (Continuous Variable)
ggplot(cleaned_dataset, aes(x = bmi)) +
  geom_histogram(fill = "#99CCFF", color = "black", bins = 30) +  # Custom color for bars
  labs(title = "Distribution of BMI")

# HbA1c_level (Continuous Variable)
ggplot(cleaned_dataset, aes(x = HbA1c_level)) +
  geom_histogram(fill = "#FF9999", color = "black", bins = 30) +  # Custom color for bars
  labs(title = "Distribution of HbA1c Level")

# Blood Glucose Level (Continuous Variable)
ggplot(cleaned_dataset, aes(x = blood_glucose_level)) +
  geom_histogram(fill = "#66CC99", color = "black", bins = 30) +  # Custom color for bars
  labs(title = "Distribution of Blood Glucose Level")
```

Codes for testing normality of the data

```
#Normality Testing
```{r}
# Perform Shapiro-Wilk test on various columns
shapiro_age <- shapiro.test(cleaned_dataset$age)
shapiro_bmi <- shapiro.test(cleaned_dataset$bmi)
shapiro_hba1c <- shapiro.test(cleaned_dataset$HbA1c_level)
shapiro_glucose <- shapiro.test(cleaned_dataset$blood_glucose_level)

# Print the results
print("Shapiro-Wilk Test for Age:")
print(shapiro_age)

print("Shapiro-Wilk Test for BMI:")
print(shapiro_bmi)

print("Shapiro-Wilk Test for HbA1c Level:")
print(shapiro_hba1c)

print("Shapiro-Wilk Test for Blood Glucose Level:")
print(shapiro_glucose)
```
```

Codes for Q-Q plots and histograms

```r
# Histogram for Age with Normal Curve
hist(cleaned_dataset$age, freq = FALSE, main = "Age Distribution")
curve(dnorm(x, mean = mean(cleaned_dataset$age), sd = sd(cleaned_dataset$age)), add = TRUE, col = "red")

# Histogram for BMI with Normal Curve
hist(cleaned_dataset$bmi, freq = FALSE, main = "BMI Distribution")
curve(dnorm(x, mean = mean(cleaned_dataset$bmi), sd = sd(cleaned_dataset$bmi)), add = TRUE, col = "red")

# Histogram for HbA1c Level with Normal Curve
hist(cleaned_dataset$HbA1c_level, freq = FALSE, main = "HbA1c Level Distribution")
curve(dnorm(x, mean = mean(cleaned_dataset$HbA1c_level), sd = sd(cleaned_dataset$HbA1c_level)), add = TRUE, col =
"red")

# Histogram for Blood Glucose Level with Normal Curve
hist(cleaned_dataset$blood_glucose_level, freq = FALSE, main = "Blood Glucose Level Distribution")
curve(dnorm(x, mean = mean(cleaned_dataset$blood_glucose_level), sd = sd(cleaned_dataset$blood_glucose_level)),
add = TRUE, col = "red")


```

```r
#Creating Q-Q plots and histograms with normal curve for assessing normality.
```{r}
# Q-Q Plot for Age
# Function to create Q-Q plots
create_qq_plot <- function(data, column) {
  qqnorm(data[[column]], main = paste("Q-Q Plot of", column))
  qqline(data[[column]], col = "red", lwd = 2)
}

# Create Q-Q plots for each continuous variable
create_qq_plot(cleaned_dataset, "age")
create_qq_plot(cleaned_dataset, "bmi")
create_qq_plot(cleaned_dataset, "HbA1c_level")
create_qq_plot(cleaned_dataset, "blood_glucose_level")

```
```

Codes for logistic regression model

```r
# Logistic regression with diabetes as the binary outcome and age as the predictor
logistic_model1 <- glm(diabetes ~ age, data = cleaned_dataset, family = binomial())

# Summary of the logistic model
summary(logistic_model1)

# Calculate the odds ratios from the model's coefficients
odds_ratios <- exp(coef(logistic_model1))

# Print the odds ratios
print(odds_ratios)
```

```r
# Logistic regression with diabetes as the binary outcome and bmi as the predictor
logistic_model6 <- glm(diabetes ~ bmi, data = cleaned_dataset, family = binomial())

# Summary of the logistic model
summary(logistic_model6)

# Calculate the odds ratios from the model's coefficients
odds_ratios <- exp(coef(logistic_model6))

# Print the odds ratios
print(odds_ratios)
```

```r
# Logistic regression with diabetes as the binary outcome and HbA1c as the predictor
logistic_model7 <- glm(diabetes ~ HbA1c_level, data = cleaned_dataset, family = binomial())

# Summary of the logistic model
summary(logistic_model7)

# Calculate the odds ratios from the model's coefficients
odds_ratios <- exp(coef(logistic_model7))

# Print the odds ratios
print(odds_ratios)

```

```r
# Logistic regression with diabetes as the binary outcome and blood glucose as the predictor
logistic_model8 <- glm(diabetes ~ blood_glucose_level, data = cleaned_dataset, family = binomial())

# Summary of the logistic model
summary(logistic_model8)

# Calculate the odds ratios from the model's coefficients
odds_ratios <- exp(coef(logistic_model8))

# Print the odds ratios
print(odds_ratios)

```

Multivariate logistic regression code

```r
# Multivariate Logistic Regression Model
full_model <- glm(diabetes ~ age + gender + hypertension + heart_disease + smoking_history + bmi + HbA1c_level +
blood_glucose_level,
                  family = binomial(), data = cleaned_dataset)


stepwise_model <- step(full_model, direction="both")

# Print the summary of the refined model
summary(stepwise_model)
```

Code for non-parametric testing (Kruskal Wallis)

```r
# Load necessary library
library(stats)

# Perform Kruskal-Wallis test for age across different levels of diabetes
kruskal_result_age <- kruskal.test(age ~ diabetes, data = cleaned_dataset)
print(kruskal_result_age)

# Perform Kruskal-Wallis test for BMI across different levels of diabetes
kruskal_result_bmi <- kruskal.test(bmi ~ diabetes, data = cleaned_dataset)
print(kruskal_result_bmi)

# Perform Kruskal-Wallis test for HbA1c level across different levels of diabetes
kruskal_result_HbA1c <- kruskal.test(HbA1c_level ~ diabetes, data = cleaned_dataset)
print(kruskal_result_HbA1c)

# Perform Kruskal-Wallis test for blood glucose level across different levels of diabetes
kruskal_result_blood_glucose <- kruskal.test(blood_glucose_level ~ diabetes, data = cleaned_dataset)
print(kruskal_result_blood_glucose)

kruskal_result_blood_glucose <- kruskal.test(smoking_history ~ diabetes, data = cleaned_dataset)
print(kruskal_result_blood_glucose)
```

Code for Dunn's post-hoc test

```r
# Install and load necessary library
install.packages("PMCMRplus")
library(PMCMRplus)

# Perform Kruskal-Wallis test for age across different levels of diabetes
kruskal_result_age <- kruskal.test(age ~ diabetes, data = cleaned_dataset)

# Check the result
print(kruskal_result_age)

# Perform post-hoc analysis for age using pairwise Wilcoxon rank sum tests with Bonferroni adjustment
posthoc_age <- PMCMRplus::kwAllPairsDunnTest(cleaned_dataset$age, cleaned_dataset$diabetes, method =
"bonferroni")

# Print the post-hoc analysis result
print(posthoc_age)

# Similarly, perform post-hoc analysis for other variables like BMI, HbA1c level, and blood glucose level
posthoc_bmi <- PMCMRplus::kwAllPairsDunnTest(cleaned_dataset$bmi, cleaned_dataset$diabetes, method =
"bonferroni")
posthoc_HbA1c <- PMCMRplus::kwAllPairsDunnTest(cleaned_dataset$HbA1c_level, cleaned_dataset$diabetes, method =
"bonferroni")
posthoc_blood_glucose <- PMCMRplus::kwAllPairsDunnTest(cleaned_dataset$blood_glucose_level,
cleaned_dataset$diabetes, method = "bonferroni")

# Print the post-hoc analysis results for BMI, HbA1c level, and blood glucose level
print(posthoc_bmi)
print(posthoc_HbA1c)
print(posthoc_blood_glucose)
```

Thank you!