PROJECT 1: DIABETES PATIENTS ANALYSIS

About Dataset

The diabetes dataset is originally from the National Institute of Diabetes and Kidney Diseases. The main objective and aim of the dataset is to diagnostically predict whether a patient has diabetes or not based on certain key features in the dataset. In particular, all patients here are females of at least 21 years old of Pima Indian heritage 2.

Features :

Pregnancies: Number of times pregnant.

Glucose: Plasma glucose concentration.

Blood Pressure: Diastolic blood pressure.

Skin Thickness: Triceps skinfold thickness.

Insulin: 2-Hour serum insulin.

BMI(Body Mass Index): A function that represents the likelihood of diabetes based on family history.

Age: Age in years.

Outcomes: A binary variable indicating the presence or absence of diabetes(0 - no diabetes, 1 - diabetes).

Data Loading

```
# Importing required libraries for analysis and visulization

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


# Importing file from a system

from google.colab import files
uploaded = files.upload()
```

Choose Files diabetes.csv

- **diabetes.csv**(text/csv) - 23875 bytes, last modified: 10/16/2023 - 100% done

```
Saving diabetes.csv to diabetes.csv
```

```
# Reading the file

df = pd.read_csv("diabetes.csv")
df.head(5)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

Data Exploration

```
# Check the rows and Columns

df.shape
```

```
(768, 9)
```

```
# Display the column names in the Dataframe

df.columns
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
# Display information about the Dataframe

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Checking the Missing Values

```
# Check Null values in a tabular form

df.isnull().sum()
```

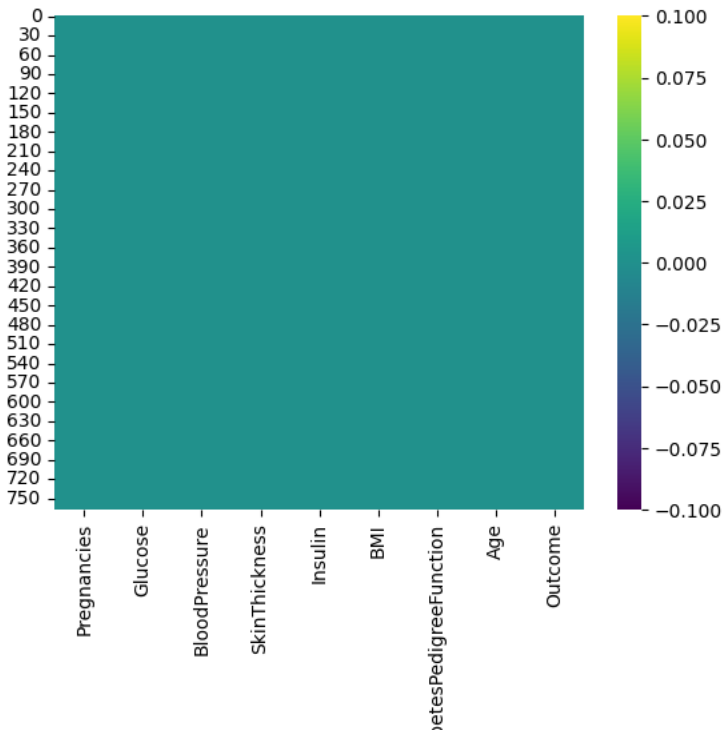
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

```
# Generate summary statitics for the numerical columns in the Dataframe

df.describe().transpose()
```

	count	mean	std	min	25%	50%	
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	1
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	1
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	

```
# Create a heatmap to visualize missing values (NULL) in the Dataframe

sns.heatmap(df.isnull(),cmap="viridis")
```

```
<Axes: >
```




```
# Calculate the correlation matrix for the Dataframe columns
# To understand the relationships between variables

correlation = df.corr()
print(correlation)
```

```
                          Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                  1.000000  0.129459       0.141282      -0.081672
Glucose                      0.129459  1.000000       0.152590       0.057328
BloodPressure                0.141282  0.152590       1.000000       0.207371
SkinThickness               -0.081672  0.057328       0.207371       1.000000
Insulin                     -0.073535  0.331357       0.088933       0.436783
BMI                          0.017683  0.221071       0.281805       0.392573
DiabetesPedigreeFunction    -0.033523  0.137337       0.041265       0.183928
Age                          0.544341  0.263514       0.239528      -0.113970
Outcome                      0.221898  0.466581       0.065068       0.074752

                           Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies              -0.073535  0.017683                 -0.033523
Glucose                   0.331357  0.221071                  0.137337
BloodPressure             0.088933  0.281805                  0.041265
SkinThickness             0.436783  0.392573                  0.183928
Insulin                   1.000000  0.197859                  0.185071
BMI                       0.197859  1.000000                  0.140647
DiabetesPedigreeFunction  0.185071  0.140647                  1.000000
Age                      -0.042163  0.036242                  0.033561
Outcome                   0.130548  0.292695                  0.173844

                               Age   Outcome
Pregnancies               0.544341  0.221898
Glucose                   0.263514  0.466581
BloodPressure             0.239528  0.065068
SkinThickness            -0.113970  0.074752
Insulin                  -0.042163  0.130548
BMI                       0.036242  0.292695
DiabetesPedigreeFunction  0.033561  0.173844
Age                       1.000000  0.238356
Outcome                   0.238356  1.000000
```

CORRELATION MATRIX

1. Glucose and Age have moderate positive correlations with Outcome, indicating that higher glucose levels and older age are associated with a higher likelihood of diabetes.
2. BMI and Insulin also show some positive correlation with Outcome, though weaker that Glucose and Age.
3. Most other variables have relatively low correlations with Outcome.

As shown in the plot above, there are no null values present.

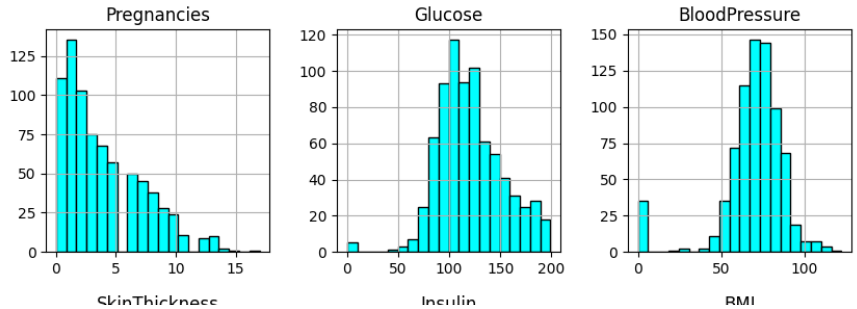
```
# Check the heatmap correlation of the features

sns.heatmap(df.corr(), annot = True, cmap = "YlGnBu")
```

```
<Axes: >
```

DATA VISUALIZATION

Distribution of data using appropriate plots.

```
# Create histogram for each column in the Dataframe with specified parameters

df.hist(bins = 20,figsize=(10,10),color='cyan',edgecolor='k',alpha=1,lw=1)
plt.show()
```




```
# Calculate skewness for the dataframe

def skewness(data):
  '''This function calcultes the skewness value
  of each column
  parameter: data=dataframe
  return: Series with skewness values for each column'''
  return data.skew()

skewness(df)
```

```
Pregnancies                 0.901674
Glucose                     0.173754
BloodPressure              -1.843608
SkinThickness               0.109372
Insulin                     2.272251
BMI                        -0.428982
DiabetesPedigreeFunction    1.919911
Age                         1.129597
Outcome                     0.635017
dtype: float64
```


By calling defined function, i calculate skewness of each column of dataframe.

If skewness is between **-0.5 to 0.5 then data are fairly symmetrical**

If skewness is between **-1 to -0.5 & 0.5 to 1 then data are moderatelu skewed**.

If skewness is between **lt -1 or gt 1 then data are highly skewed.**

To find the outliers present in a given dataset using box plot method

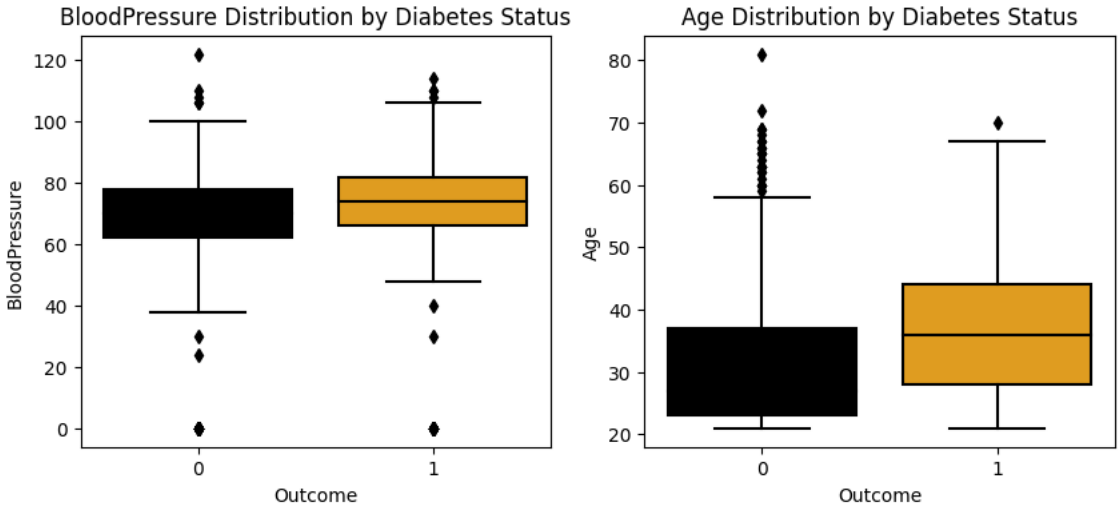
```
c_palette = {0: 'black', 1: 'orange'}

# Create a figure with two subplots
fig, axes = plt.subplots(1, 2, figsize=(10, 4))

# Plot the first boxplot on the left subplot
sns.boxplot(x="Outcome", y="Pregnancies", data=df, palette=c_palette, ax=axes[0])
axes[0].set_title("Diabetes by Pregnancies")

# Plot the second boxplot on the right subplot
sns.boxplot(x="Outcome", y="Glucose", data=df, palette=c_palette, ax=axes[1])
axes[1].set_title("Glucose Distribution by Diabetes Status")

# Display the plots
plt.show()
```

This box plot illustrates the distribution of the number of pregnancies for diabetic(1) and non-diabetic(0) patients.

The non-diabetic group(0) exhibits outliers

Diabetic patients tend to have higher glucose levels compared to non-diabetic patient, there are some exceptioal case with elevated glucose levels.

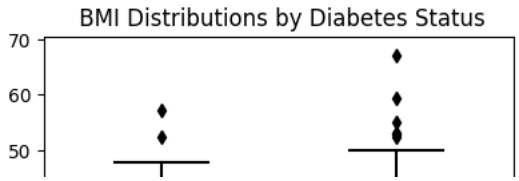
```
c_palette = {0:"black", 1:"orange"}

# Create a figure with two subplots
fig, axes = plt.subplots(1, 2, figsize=(10, 4))

# Plot the first boxplot for "BloodPressure" on the left subplot
sns.boxplot(x="Outcome", y="BloodPressure", data=df, palette=c_palette, ax=axes[0])
axes[0].set_title("BloodPressure Distribution by Diabetes Status")

# Plot the second boxplot for "Age" on the right subplot
sns.boxplot(x="Outcome", y="Age", data=df, palette=c_palette, ax=axes[1])
axes[1].set_title("Age Distribution by Diabetes Status")

# Display the plots
plt.show()
```

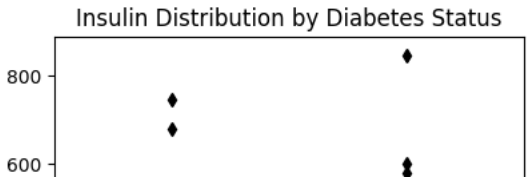


Diabetic individuals (orange) tend to have a slightly higher blood pressure compared to non-diabetic individuals (black).

Diabetic individuals (orange) are generally younger, while non-diabetic individuals (black) span a wider age range.

```
c_palette = {0: 'black', 1: 'orange'}

# Create a figure with two subplots
fig, axes = plt.subplots(1, 2, figsize=(10,4))

# Plot the boxplot for "BMI" on the left subplot
sns.boxplot(x="Outcome", y="BMI", data=df, palette=c_palette, ax=axes[0])
axes[0].set_title(" BMI Distributions by Diabetes Status")

# Plot the boxplot for "Insulin" on the right subplot
sns.boxplot(x="Outcome", y="Insulin", data=df, palette=c_palette, ax=axes[1])
axes[1].set_title("Insulin Distribution by Diabetes Status")

# Display the plots
plt.show()
```





There's a noticeable difference in BMI distribution between the two groups, with **diabetic individuals having higher BMI on average.**

Diabetic individuals show more variation in insulin levels(orange) that non-diabetics(black).

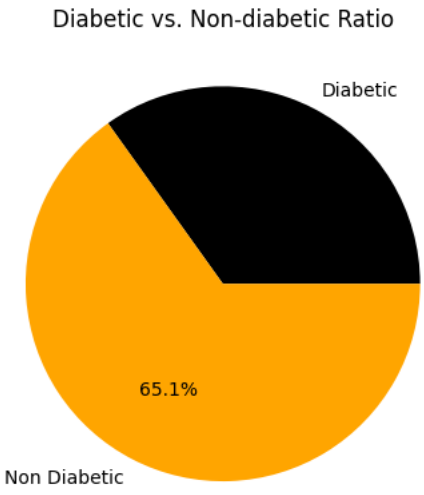
Glucose: The Glucose column does not show significant outliers, and it has a relatively symmetrical distribution.

Outcome: The "Outcome" column is a binary variable and is not visualized in a box plot.

And rest columns shows a skewed distributions with some outliers

Distribution Of Outcome

```
# Calculate the number of diabetic individuals (outcoe-1) and non-diabetic individuals (outcome-0)
diabetic = len(df[df["Outcome"]==1])
non_diabetic = len(df[df["Outcome"]==0])

# Create a counts of diabetic and non-diabetic individuals
count = (diabetic, non_diabetic)

# Define labels for the pie chart, representing the categories
labels = ("Diabetic", "Non Diabetic")

plt.pie(count, labels=labels, autopct="%1.1f%%", colors={"orange", "black"})

# Set a title for the pie chart
plt.title("Diabetic vs. Non-diabetic Ratio")

plt.show()
```



Importing Libraries for Machine Learning Model

```
# Import necessary libraries and suppress warnings

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

import warnings
warnings.filterwarnings('ignore')


# Splitting Data into training and testing sets

x = df.drop('Outcome', axis=1)
y = df['Outcome']
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

```
# Creating and Training a Logistic Regression Model
model = LogisticRegression()
model.fit(x_train,y_train)
```

```
LogisticRegression()
```

```
# Making Predictions and Printing the results
prediction = model.predict(x_test)
print(prediction)
```

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

```
# Calculating and Printing Model Accuracy
accuracy = accuracy_score(prediction,y_test)
print(accuracy)
```

```
0.7662337662337663
```

Conclusion

The Diabetic Patient's data analysis project identified key factors associated with diabetes, including age, BMI, Blood pressure, and insulin levels. A logistic regression model achieved an accuracy of appproximately 77% in the predicting diabetes. The analysis highlights the need for further exploration and data preprocessing to support effective strategies for diabetes prevention and management."