Diabetes in Youth and Adults in India

The Project aims at finding the number of youth and adults suffering from diabetes based on various parameters such as Age group, Alcohol consumption, Gender, Genetic Risk Score and The type of diabetes they are suffering from according to the age group.

The whole data set is divided into two age groups, i.e., 15-20 and 20-25. The aim of the project is to analyse how these all are related to the nature of diabetes people are suffering from.

Vanshik Narula

vanshiknarula03@gmail.com

DIABETES IN YOUTH AND ADULTS IN INDIA ANALYSIS

Step 1: Import all the libraries

Code:

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

Step 2: Read the Dataset

Code & Output:

```
df=pd.read_csv("C:\\Users\\MY PC\\Downloads\\diabetes_young_adults_india.csv")
```

```
df.head()
```

	ID	Age	Gender	Region	Family_Income	Family_History_Diabetes	Parent_Diabetes_Type	Genetic_Risk_Score	BMI	Physical_Activity_Level	...	Smoking
0	1	21	Male	North	2209393	No	None	6	31.4	Sedentary	...	Yes
1	2	18	Female	Central	387650	No	None	5	24.4	Active	...	No
2	3	25	Male	North	383333	No	None	6	20.0	Moderate	...	No
3	4	22	Male	Northeast	2443733	No	None	4	39.8	Moderate	...	No
4	5	19	Male	Central	1449463	No	None	4	19.2	Moderate	...	No

5 rows × 22 columns

Step 3: Checking Data Info

Code & Output:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 22 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   ID                       100000 non-null  int64
 1   Age                      100000 non-null  int64
 2   Gender                   100000 non-null  object
 3   Region                   100000 non-null  object
 4   Family_Income            100000 non-null  int64
 5   Family_History_Diabetes  100000 non-null  object
 6   Parent_Diabetes_Type     100000 non-null  object
 7   Genetic_Risk_Score       100000 non-null  int64
 8   BMI                      100000 non-null  float64
 9   Physical_Activity_Level  100000 non-null  object
 10  Dietary_Habits           100000 non-null  object
 11  Fast_Food_Intake         100000 non-null  int64
 12  Smoking                  100000 non-null  object
 13  Alcohol_Consumption      100000 non-null  object
 14  Fasting_Blood_Sugar      100000 non-null  float64
 15  HbA1c                    100000 non-null  float64
 16  Cholesterol_Level        100000 non-null  float64
 17  Prediabetes              100000 non-null  object
 18  Diabetes_Type            100000 non-null  object
 19  Sleep_Hours              100000 non-null  float64
 20  Stress_Level             100000 non-null  int64
 21  Screen_Time              100000 non-null  float64
dtypes: float64(6), int64(6), object(10)
memory usage: 16.8+ MB
```

Step 4: Checking the null values

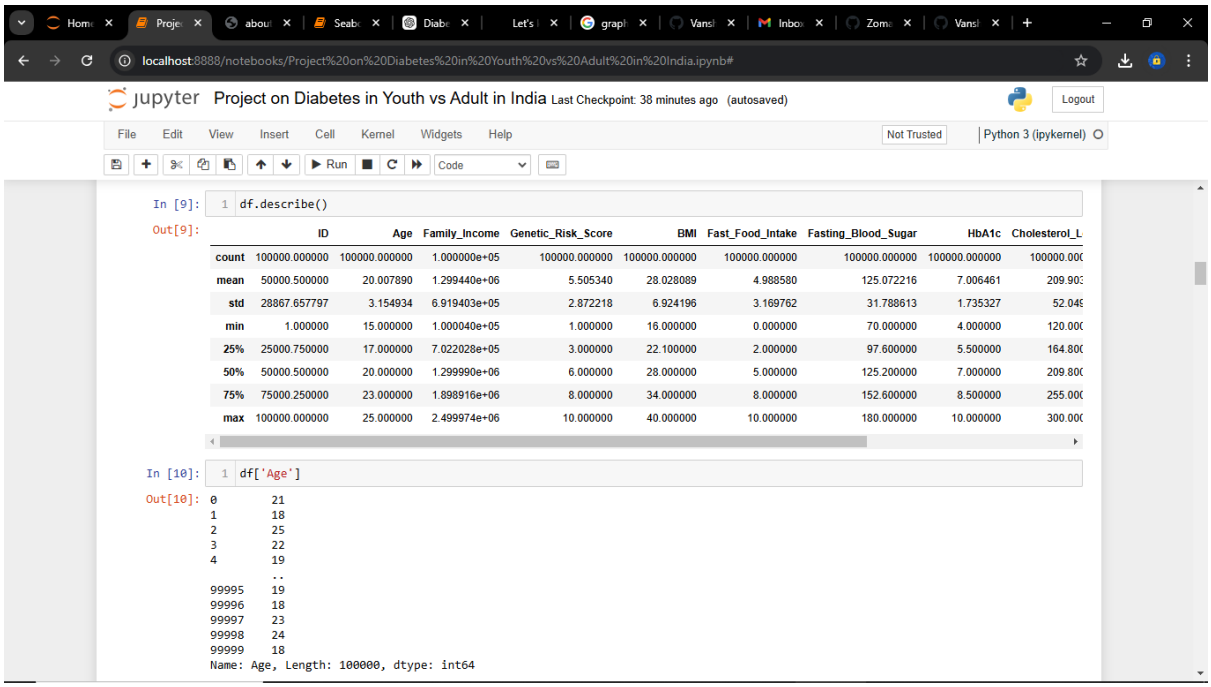
Code & Output:

```
1 df.isna().sum()
```

```
ID                         0
Age                        0
Gender                     0
Region                     0
Family_Income              0
Family_History_Diabetes    0
Parent_Diabetes_Type       0
Genetic_Risk_Score         0
BMI                        0
Physical_Activity_Level    0
Dietary_Habits             0
Fast_Food_Intake           0
Smoking                    0
Alcohol_Consumption        0
Fasting_Blood_Sugar        0
HbA1c                      0
Cholesterol_Level          0
Prediabetes                0
Diabetes_Type              0
Sleep_Hours                0
Stress_Level               0
Screen_Time                0
dtype: int64
```

Step 5: Descriptive Analysis Of dataset

Code & Output:



Step 6: Making a Column of Age groups:

Code & Output:

```
bins = [15,20,26]  # Define the bin edges
labels = ['15-20', '20-26']
```

```
grouped = df.groupby('Age_Group')

age_group_summary = grouped.size()

print(age_group_summary)
```

```
Age_Group
15-20    45258
20-26    54742
dtype: int64
```

Step 7: Converting diabetes_type column in 0,1 for further analysis

Code & Output:

```
# Filter data for diabetic patients
diabetic_data = df[df['Diabetes_Type'] != 'None']
```

```
df['Has_Diabetes'] = df['Diabetes_Type'].apply(lambda x: 1 if x != 'None' else 0)
df.head()
```

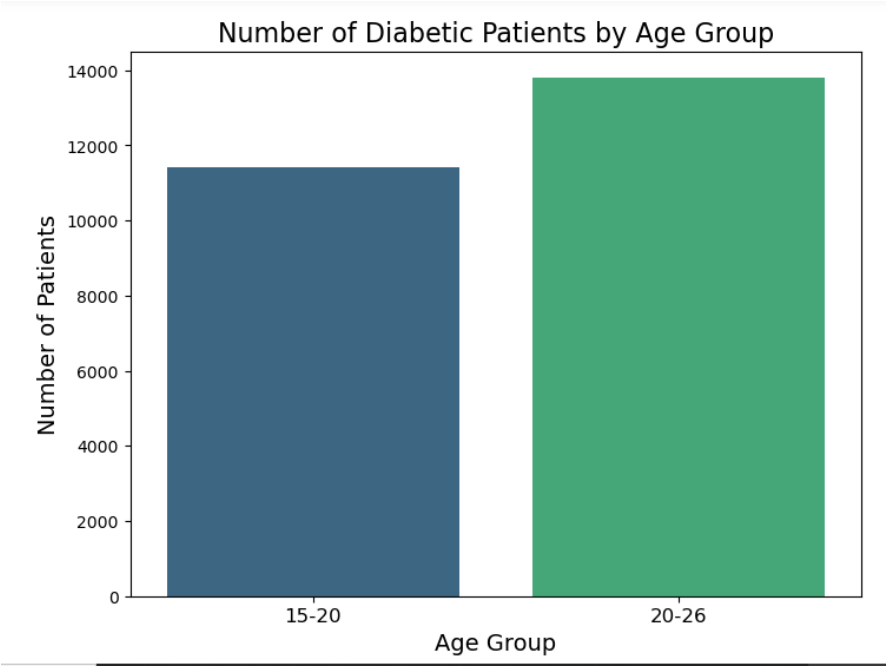
	ID	Age	Gender	Region	Family_Income	Family_History_Diabetes	Parent_Diabetes_Type	Genetic_Risk_Score	BMI	Physical_Activity_Level	...	Fasting_Bl
0	1	21	Male	North	2209393	No	None	6	31.4	Sedentary	...	
1	2	18	Female	Central	387650	No	None	5	24.4	Active	...	
2	3	25	Male	North	383333	No	None	6	20.0	Moderate	...	
3	4	22	Male	Northeast	2443733	No	None	4	39.8	Moderate	...	
4	5	19	Male	Central	1449463	No	None	4	19.2	Moderate	...	

5 rows × 24 columns

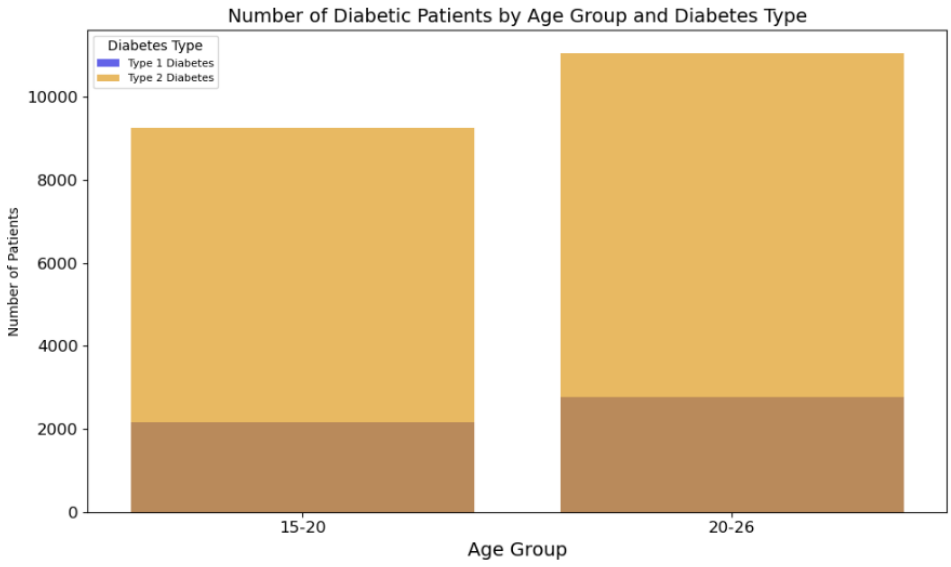
Cleaning is Completed

DATA VISUALISATION

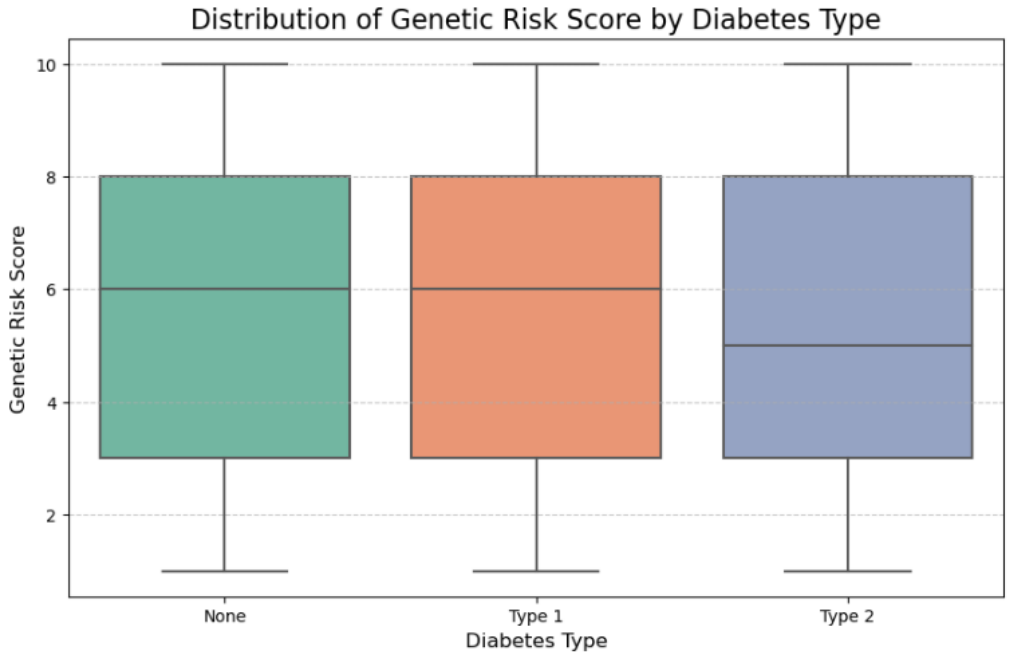
1. Age Group Wise Diabetic Patients.



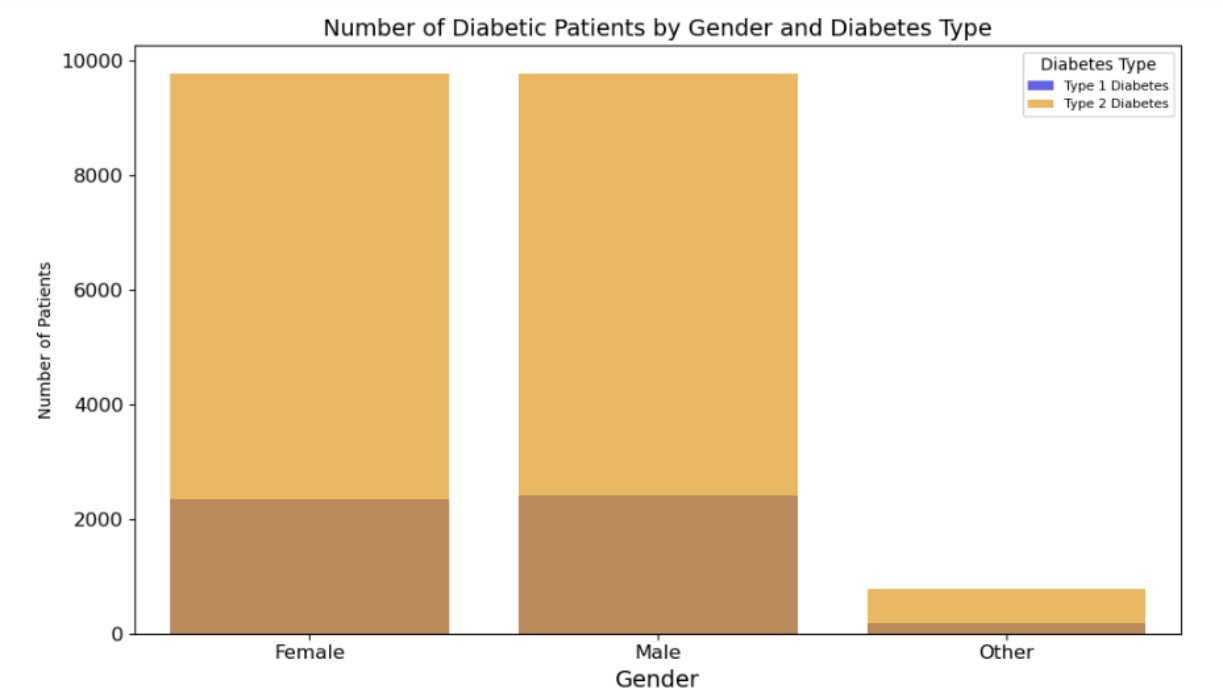
2. Age Group Wise Diabetic Type of Patients



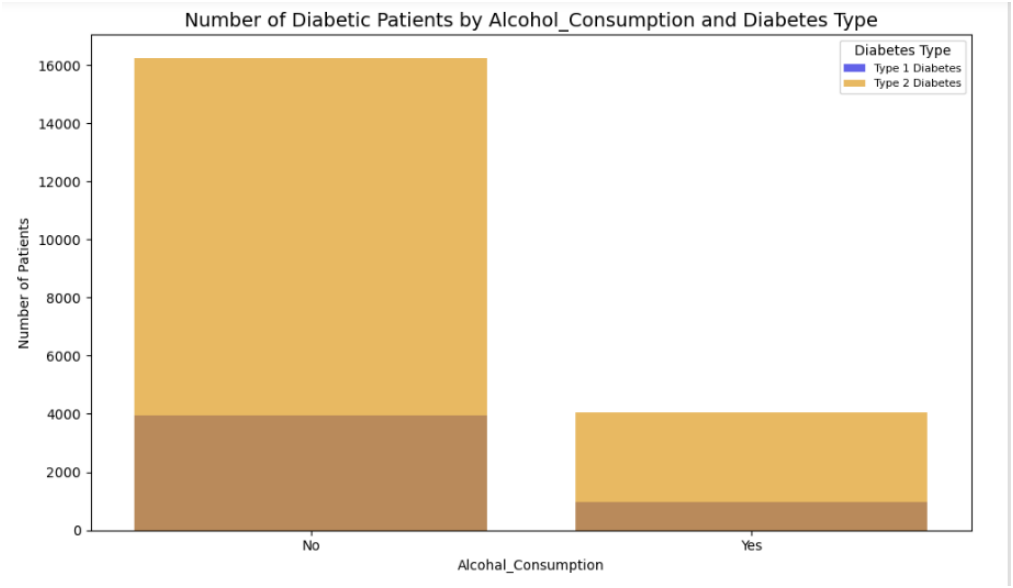
3. Boxplot for Comparison of Genetic Risk Score with Diabetes Type:



4. Distribution of Diabetic Patients With respect to Gender



5. Diabetic Patients with respect to alcohol consumption



DASHBOARD