اونيۏرسيتي مليسيا ڤهڠ

UNIVERSITI MALAYSIA PAHANG

BSD2333 DATA WRANGLING

2021/2022 SEMESTER II

GROUP'S NAME: KRUSKAL WALLIS

TITLE:

DIABETES DISEASE DETECTION

PREPARED FOR

DR MOHD KHAIRUL BAZLI BIN MOHD AZIZ

PREPARED BY

MATRIC ID	NAME	SECTION
SD20008	SARAH BATRISYIA BINTI MORSHIDI	01G
SD20009	SITI NUR AISYAH BINTI ANUAR	
SD20022	NIK NUR AIN BINTI NIK JID	
SD20034	NIK NURUL SYUHADA BINTI MOHD ALI	
SD20065	MUHAMMAD ISYHRAF BIN AZMIN	

SUBMISSION DATE

27TH MAY 2022

GROUP MEMBERS PHOTOS :

TABLE OF CONTENTS

1.0 SYNOPSIS

1.1 Description of the assignment

This assignment requires us to explore a real dataset and apply our data wrangling knowledge on real data. We are required to search a complete set of data that has at least 5 or more integer attributes and must be at least 1000 data. Our group consists of 5 members and we choose Diabetes Disease as our topic for this assignment. Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a hormone that regulates blood sugar. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels. There are two types of diabetes which are Type 1 Diabetes and Type 2 Diabetes. Type 1 Diabetes is a serious condition where your blood glucose (sugar) level is too high because your body cannot make a hormone called insulin. This happens because your body attacks the cells in your pancreas that make the insulin, meaning you cannot produce any at all. While Type 2 Diabetes is a serious condition where the insulin your pancreas makes cannot work properly, or your pancreas cannot make enough insulin. This means your blood glucose (sugar) levels keep rising. When you have type 2 diabetes, your body still breaks down carbohydrate from your food and drink and turns it into glucose. The pancreas then responds to this by releasing insulin. But because this insulin cannot work properly, your blood sugar levels keep rising. In this assignment we will discuss the relationship of the variables attainable in the dataset with the diabetes disease.

1.2 Problem to be solved

In 2019, an estimated 1.4 million new cases of diabetes were diagnosed among people ages 18 and older. Seeing how this disease keeps on increasing by year, it is starting to be a concerning issue around the world. We chose this dataset to help in identifying which factor leads to this diabetes disease and which factor is the high factor that can increase the amount of this disease. In identifying these factors, we hoped that we are able to detect factors related to diabetes much quicker so that it might help the health system.

1.3 Question to be answered

There are several question to be answered in this assignment, those questions are:

- What is the average age of a person that suffers from diabetes?
- What is the major factor that could lead to diabetes?
- What effect does the measurement of BMI have on predicting the likelihood of someone getting diabetes?

1.4 Objectives

Diabetes Mellitus (DM), a chronic metabolic condition, is one of the most pressing public health issues of the twenty-first century. DM is defined by high blood glucose levels, which are mostly caused by insufficient insulin synthesis or the body's inability to respond to insulin. According to the most recent data from the International Diabetes Federation (IDF,https://www.diabetesatlas.org/), there are 425 million diabetics worldwide, with one in every 11 adults diagnosed with the disease. The number of individuals with diabetes is anticipated to rise to 629 million by 2045. Below are the objectives for the diabetes dataset:

- To detect factors related to diabetes much quicker.
- To improve the quality of life among diabetic people.
- To make better decisions in diabetes medical care.
- To identify the age range of people that suffer diabetes.

1.5 Data Description

Data Variable	Data Description
Pregnancies	The number of times a patient got pregnant.
Glucose	Plasma glucose concentration is 2 hours in an oral glucose tolerance test.
Blood Pressure	Diastolic blood pressure test of the patients(mm Hg).

Skin Thickness	Triceps skin fold thickness of non diabetic and diabetic patients(mm).
Insulin	2-Hour serum insulin intake of the patients in (mu U/ml).
BMI	Body mass index of the patients(weight in kg/(height in m)^2).
Diabetes Pedigree Function	Diabetes pedigree function (a function which scores likelihood of diabetes based on family history).
Age	Age (years of patients).
Outcome	Class variable (0 if non diabetic and 1 if diabetic).

Table of data description 1.5.1

2.0 PACKAGES REQUIRED

2.1 Pandas

- Read csv file - In order to read the csv file and obtain the data, pandas packages are needed.
- Data frame df - Load the data frame to cleaning and deal with the related rows and columns.
- Data loc - To replace the 0 values for Skin Thickness, BMI and Blood Pressure with a mean for each variable.
- Fillna - Replace the missing value with a suitable or appropriate value.
- Desc data - Looking for outliers for related variables and the outliers will be removed once we detect the outliers.
- Head() - Used to return the first 5 rows on a selected column that we use for data cleaning.

2.2 Numpy

- Mean - The function that we use to replace with 0 values.
- Sort - Sort related columns and rows.

2.3 Scipy

- stats.iqr - Statistic function to find an interquartile range based on a selected variable.

2.4 Matplotlib.pyplot

- plt.figure - Use to change and control the size of the figure.

2.5 Seaborn

- sns.heatmap - Used to see the correlation between variables and blood pressure based on age.
- sns.countplot - Used to count the number of outcomes from datasets.
- sns.lineplot - Used to visualize a line graph for specific variables.
- sns.hisplot - Used to visualize the blood pressure in the dataset.

- sns.scatterplot - Used to visualize the effect of skin thickness on BMI.

2.6 Altair

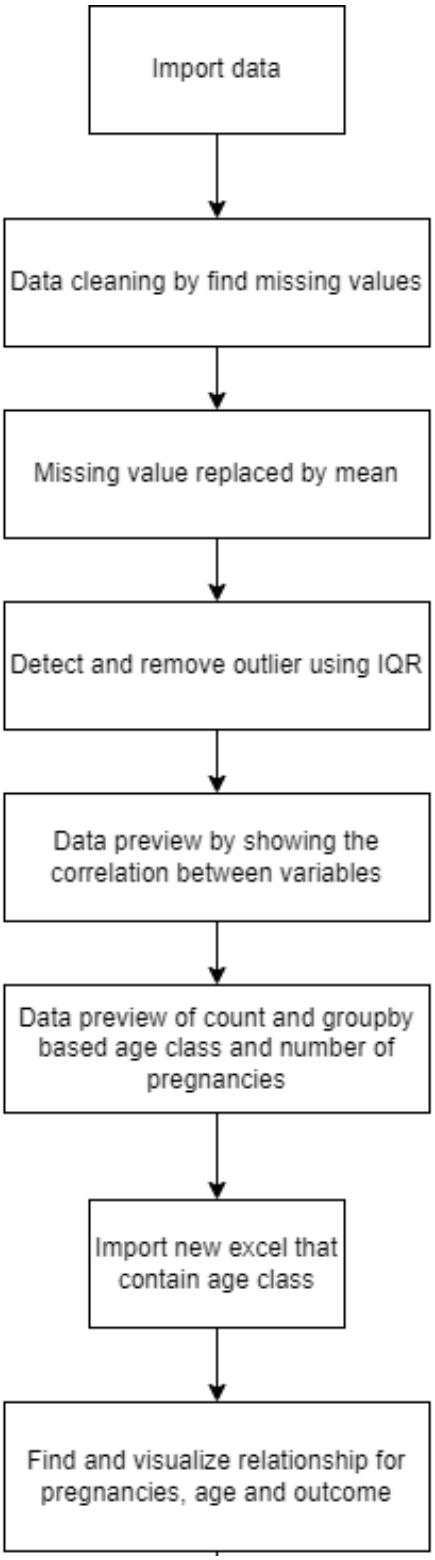
- alt.chart - To visualize the diabetic patient with zero times of pregnancy, diabetics with a history of pregnancy in interactive visualization.

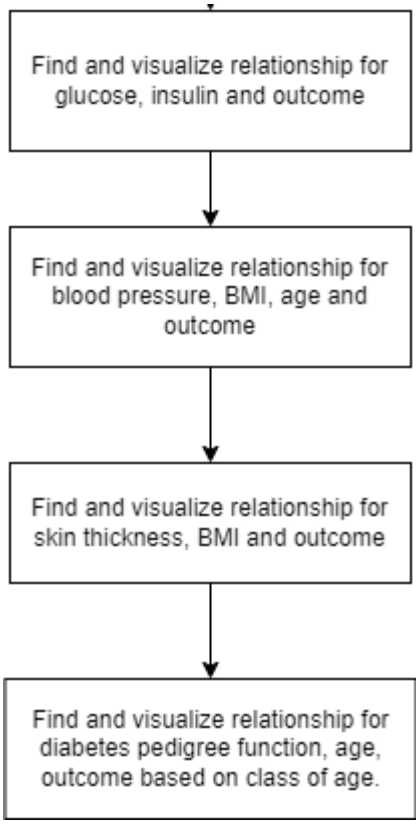
2.7 Plotly.express

- px.bar - interactive bar graph for number of pregnancies by age class and total number of positive diabetic by age class.
- px.histogram - interactive histogram for comparison of outcome based on age.

3.0 DATA PREPARATION

FLOWCHART DATA PREPARATION




Find and visualize relationship for glucose, insulin and outcome
Find and visualize relationship for blood pressure, BMI, age and outcome
Find and visualize relationship for skin thickness, BMI and outcome
Find and visualize relationship for diabetes pedigree function, age, outcome based on class of age.

3.1 Data Import

The first step in preparing the data is to load the dataset that we have chosen into the jupyter notebook. We used the pandas module to load the data by creating a data frame for the csv file. The csv file must be in the same directory as the python file in order to load it. Otherwise, the location of the file needs to be specified in the coding. The output of the coding will show the contents of the csv file that we have loaded.

In [145]:
```
import pandas as pd
import numpy as np
Diabetes= pd.read_csv("diabetes.csv")
#pd.set_option('display.max_rows', None)
#pd.set_option('display.max_columns', None)
Diabetes
```

Out[145]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1
10	4	110	92	0	0	37.6	0.191	30	0
11	10	168	74	0	0	38.0	0.537	34	1

3.2 Data Cleaning

Next, we are required to quickly analyze and understand the data that has been loaded. After understanding the contents of the dataset, we can start doing the data cleaning. The diagram below shows the coding to identify any data that might be a null value. If the output is false, then there is null value but if the output is true, then it is a null value.

```
In [146]: Diabetes.isnull()
```

Out[146]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False	False

Since the dataset is huge, we might not be able to check each row. Thus, to make it easier we checked the total number of null values in the dataset. The diagram below shows that there are no null values in this dataset.

```
In [162]: Diabetes.isna().sum()

Out[162]: Pregnancies                 0
          Glucose                     0
          BloodPressure               0
          SkinThickness               0
          Insulin                     0
          BMI                         0
          DiabetesPedigreeFunction    0
          Age                         0
          Outcome                     0
          dtype: int64
```

Although there are no null values, some of the variables like blood pressure, skin thickness, BMI and insulin have some values that are equal to zero. In general, we know that a human being could never have a blood pressure, skin thickness, BMI , glucose and insulin that are zero. Thus, we replaced the zeroes with the mean of each variable.

```
In [67]: meanBloodPressure = Diabetes['BloodPressure'].mean(skipna=True)
         Diabetes.loc[Diabetes.BloodPressure == 0,  'BloodPressure'] = meanBloodPressure

         meanSkinThickness = Diabetes['SkinThickness'].mean(skipna=True)
         Diabetes.loc[Diabetes.SkinThickness == 0,  'SkinThickness'] = meanSkinThickness

         meanBMI = Diabetes['BMI'].mean(skipna=True)
         Diabetes.loc[Diabetes.BMI == 0,  'BMI'] = meanBMI

         meanInsulin = Diabetes['Insulin'].mean(skipna=True)
         Diabetes.loc[Diabetes.Insulin == 0, 'Insulin'] = meanInsulin

         meanGlucose = Diabetes['Glucose'].mean(skipna=True)
         Diabetes.loc[Diabetes.Glucose == 0, 'Glucose'] = meanGlucose

         #Replacing the 0 value with mean where we decide to choose columns BloodPressure,SkinThickness,BMI & Insulin
         print(Diabetes)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin   BMI  \
0              6    148.0           72.0      35.000000   79.799479  33.6
1              1     85.0           66.0      29.000000   79.799479  26.6
2              8    183.0           64.0      20.536458   79.799479  23.3
3              1     89.0           66.0      23.000000   94.000000  28.1
4              0    137.0           40.0      35.000000  168.000000  43.1
..           ...      ...            ...            ...         ...   ...
763           10    101.0           76.0      48.000000  180.000000  32.9
764            2    122.0           70.0      27.000000   79.799479  36.8
765            5    121.0           72.0      23.000000  112.000000  26.2
766            1    126.0           60.0      20.536458   79.799479  30.1
767            1     93.0           70.0      31.000000   79.799479  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
```

The next process of the data cleaning is to detect the outliers. We used the describe function to show the basic statistical details of the dataset. We can compare each variable's maximum, minimum and the mean values whether there exists a significant gap between those values.

```
In [68]: Diabetes.describe()
```

Out[68]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.681605	72.254807	26.606479	118.660163	32.450805	0.471876	33.240885	0.348958
std	3.369578	30.436016	12.115932	9.631241	93.080358	6.875374	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.750000	64.000000	20.536458	79.799479	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	79.799479	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In order to get an accurate view of the outliers, we use the interquartile range method to detect the dataset's outliers. The interquartile range approach calculates a lower and upper value using the 5-th and 95th-percentiles, with all values lower than the lower value and all values greater than the upper value designated as outliers.

```
In [112]: # IQR
Q1 = np.percentile(Diabetes, 25,
                   interpolation = 'midpoint')

Q3 = np.percentile(Diabetes, 75,
                   interpolation = 'midpoint')
IQR = Q3 - Q1

# Above Upper bound
upper = Diabetes >= (Q3+1.5*IQR)

print("Upper bound:",upper)
print(np.where(upper))

# Below Lower bound
lower = Diabetes <= (Q1-1.5*IQR)
print("Lower bound:", lower)
print(np.where(lower))
```

```
Upper bound:      Pregnancies  Glucose  BloodPressure  SkinThickness  Insuli
n    BMI  \
0          False    False          False          False    False  False
1          False    False          False          False    False  False
2          False    False          False          False    False  False
3          False    False          False          False    False  False
4          False    False          False          False    False  False
5          False    False          False          False    False  False
```

After detecting the outlier using the interquartile range, we chose to trim all the outliers using the interquartile range method. These outliers were removed to ensure the quality of the data before we start analyzing it.

```
In [113]: #find Q1, Q3, and interquartile range for each column
Q1 = Diabetes.quantile(q=.25)
Q3 = Diabetes.quantile(q=.75)
IQR = Diabetes.apply(stats.iqr)

#only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
data_clean = Diabetes[~((Diabetes < (Q1-1.5*IQR)) | (Diabetes > (Q3+1.5*IQR))).a

#find how many rows are left in the dataframe
newshape = data_clean.shape
oldshape = Diabetes.shape

print("Old shape: ", oldshape)
print("New shape: ", newshape)
```

```
Old shape:  (768, 9)
New shape:  (619, 9)
```

3.3 Data Preview

After we have cleaned the data, we start by observing each variable's correlation. We want to see whether there exists a significant correlation between those variables in order to have a better understanding regarding each column's relationships. 1 indicates a strong positive relationship while -1 indicates a strong negative relationship. If the result of correlation is equal to 0 then it has no relationship at all.

```
In [135]: Diabetes.corr()
```

Out[135]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.127964	0.208984	0.013376	-0.018082	0.021546	-0.033523	0.544341	0.221898
Glucose	0.127964	1.000000	0.219666	0.160766	0.396597	0.231478	0.137106	0.266600	0.492908
BloodPressure	0.208984	0.219666	1.000000	0.134155	0.010926	0.281231	0.000371	0.326740	0.162986
SkinThickness	0.013376	0.160766	0.134155	1.000000	0.240361	0.535703	0.154961	0.026423	0.175026
Insulin	-0.018082	0.396597	0.010926	0.240361	1.000000	0.189856	0.157806	0.038652	0.179185
BMI	0.021546	0.231478	0.281231	0.535703	0.189856	1.000000	0.153508	0.025748	0.312254
DiabetesPedigreeFunction	-0.033523	0.137106	0.000371	0.154961	0.157806	0.153508	1.000000	0.033561	0.173844
Age	0.544341	0.266600	0.326740	0.026423	0.038652	0.025748	0.033561	1.000000	0.238356
Outcome	0.221898	0.492908	0.162986	0.175026	0.179185	0.312254	0.173844	0.238356	1.000000

Next, we grouped the column age by range (21-30), (31-40), (41-50), (51-60) and (61-70) to analyze other attributes according to the age class.

```
In [12]: # grouping age by range (21-30), (31-40), (41-50), (51-60), (61-70)

labels = ["{0} - {1}".format(i, i+9) for i in range(21, 71, 10)]
category = pd.cut(data_clean['Age'], np.arange(20, 71, 10),
            include_lowest=True, right=False,
            labels=labels)
ageClass = data_clean['Age'].groupby(category).agg(['count'])
print(ageClass)
```

```
         count
Age
21 - 30    329
31 - 40    134
41 - 50     96
51 - 60     41
61 - 70     19
```

We then converted the dataset according to their age range and placed it in an excel. We imported the new excel that has been cleaned and grouped the data by the number of pregnancies according to their age class.

In [13]:
```
# import new excel that containt ageClass
data_clean1 = pd.read_excel("data_clean.xlsx")

#group data by ageClass follow by pregnancies
age=data_clean1.groupby(['ageClass', 'Pregnancies'])
age.first()  #print value in each group
```

Out[13]:

ageClass	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPe
21-30	0	105.0	64.000000	41.000000	142.000000	41.500000	
	1	89.0	66.000000	23.000000	94.000000	28.100000	
	2	90.0	68.000000	42.000000	79.799479	38.200000	
	3	78.0	50.000000	32.000000	88.000000	31.000000	
	4	110.0	92.000000	20.536458	79.799479	37.600000	

From the output above, we used the groupby function for age class according to the summation of the pregnancies. This table shows all summation of the variables based on the age range.

In [14]:
```
age1=data_clean1.groupby('ageClass').sum('Pregnancies')
age1
```

Out[14]:

ageClass	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	D
21-30	728	38312.683594	23669.320312	8598.208333	30960.716146	10723.340625	
31-40	682	15617.894531	9376.738281	3317.359375	12022.559896	4038.900000	
41-50	627	11511.894531	7112.421875	2470.458333	8566.166667	3180.600000	
51-60	234	4935.000000	2947.000000	873.729167	3694.585938	1121.192578	
61-70	79	2238.000000	1230.000000	369.901042	1487.192708	466.300000	

Based on the correlation output that we have obtained, we have formed multiple subsets according to their coefficient correlation. The subset allows us to access only a certain part of the data frame. This can make our analytical process much easier since we have limited the variables according to their coefficient correlation.

```
In [48]: subset1 = data_clean[['Pregnancies','Age','Outcome']]
         subset1.head()
```

Out[48]:

	Pregnancies	Age	Outcome
0	6	50	1
1	1	31	0
2	8	32	1
3	1	21	0
5	5	30	0

```
In [51]: subset2 = data_clean[['Glucose','Insulin','Outcome']]
         subset2.head()
```

Out[51]:

	Glucose	Insulin	Outcome
0	148.0	79.799479	1
1	85.0	79.799479	0
2	183.0	79.799479	1
3	89.0	94.000000	0
5	116.0	79.799479	0

```
In [56]: subset3 = data_clean[['BloodPressure','BMI','Age','Outcome']]
         subset3.head()
```

Out[56]:

	BloodPressure	BMI	Age	Outcome
0	72.0	33.6	50	1
1	66.0	26.6	31	0
2	64.0	23.3	32	1
3	66.0	28.1	21	0
5	74.0	25.6	30	0

```
In [65]: subset4 = data_clean[['SkinThickness','BMI','Outcome']]
         subset4.head()
```

Out[65]:

	SkinThickness	BMI	Outcome
0	35.000000	33.6	1
1	29.000000	26.6	0
2	20.536458	23.3	1
3	23.000000	28.1	0
5	20.536458	25.6	0

```
In [37]: subset5 = data_clean1[['DiabetesPedigreeFunction','Age','Outcome', 'ageClass']]
         subset5.head()
```

Out[37]:

	DiabetesPedigreeFunction	Age	Outcome	ageClass
0	0.627	50	1	41-50
1	0.351	31	0	31-40
2	0.672	32	1	31-40
3	0.167	21	0	21-30
4	0.201	30	0	21-30

Based on the subset1 that we have formed, we wanted to see the age of people that suffer diabetes and have never got pregnant.

```
In [49]: notpregnant = subset1[(data_clean['Pregnancies']==0) & (data_clean['Outcome']== 1)]
         notpregnant
```

Out[49]:

	Pregnancies	Age	Outcome
16	0	31	1
66	0	38	1
78	0	26	1
109	0	24	1
124	0	23	1
129	0	62	1
164	0	32	1
213	0	24	1
237	0	23	1
266	0	25	1
280	0	28	1
291	0	25	1

We also made a table for people that have diabetes and have been pregnant.

```
In [120]: pregnant = subset1[(data_clean['Pregnancies']>=1) & (data_clean['Outcome']== 1)]
          pregnant
```

Out[120]:

	Pregnancies	Age	BMI	Outcome
0	6	50	33.600000	1
2	8	32	23.300000	1
6	3	26	31.000000	1
9	8	54	31.992578	1
11	10	34	38.000000	1
...	...	...	...	...
754	8	45	32.400000	1
755	1	37	36.500000	1
759	6	66	35.500000	1
761	9	43	44.000000	1
766	1	47	30.100000	1

211 rows × 4 columns

Next, we formed a table that are non-diebetic outcomes based on the subset2.

```
In [95]: nondiabetic = subset2[(data_clean['Outcome']==0)]
         nondiabetic
```

Out[95]:

	Glucose	Insulin	Outcome
1	85.0	79.799479	0
3	89.0	94.000000	0
5	116.0	79.799479	0
7	115.0	79.799479	0
10	110.0	79.799479	0
...	...	...	...
762	89.0	79.799479	0
763	101.0	180.000000	0
764	122.0	79.799479	0
765	121.0	112.000000	0
767	93.0	79.799479	0

476 rows × 3 columns

The diagram below shows the data for blood pressure and age with a condition that all those data result in diabetes and have a normal BMI from the subset3.

```
In [76]: normalBMI = subset3[(data_clean['BMI']>=18.5) & (data_clean['BMI']<=24.9) &
                              (data_clean['Outcome']==1)]
         normalBMI
```

Out[76]:

	BloodPressure	BMI	Age	Outcome
2	64.0	23.3	32	1
93	72.0	23.8	60	1
197	62.0	22.9	23	1
319	78.0	23.5	59	1
646	74.0	23.4	33	1
676	86.0	24.8	53	1
749	62.0	24.3	50	1

From the subset4, we chose to display the 'Skin Thickness' and 'BMI' data that result with non-diabetic. This lets us go through the range of data skin thickness and BMI that does not have diabetes

```
In [121]: st_nondiabetic = subset4[(data_clean['Outcome']==0)]
          st_nondiabetic
```

Out[121]:

	SkinThickness	BMI	Outcome
1	29.000000	26.6	0
3	23.000000	28.1	0
5	20.536458	25.6	0
7	20.536458	35.3	0
10	20.536458	37.6	0
...	...	...	...
762	20.536458	22.5	0
763	48.000000	32.9	0
764	27.000000	36.8	0
765	23.000000	26.2	0
767	31.000000	30.4	0

476 rows × 3 columns

We observed the diabetes pedigree function with an outcome that resulted in diabetes from the subset5.

```
In [85]: pedigree = subset5[(data_clean['Outcome']==1)]
         pedigree
```

Out[85]:

	DiabetesPedigreeFunction	Outcome
0	0.627	1
2	0.672	1
6	0.248	1
9	0.232	1
11	0.537	1
...	...	...
755	1.057	1
757	0.258	1
759	0.278	1
761	0.403	1
766	0.349	1

242 rows × 2 columns

Lastly, the summation of the subset 5 according to their age class.

```
In [41]: # number of diabetic patients by age class
         pedigree1=pedigree.groupby('ageClass').sum('Outcome')
         pedigree1
```

Out[41]:

	DiabetesPedigreeFunction	Age	Outcome
ageClass			
21-30	29.778	1599	63
31-40	29.827	1992	57
41-50	22.847	2250	51
51-60	7.972	963	18
61-70	2.363	255	4

3.4 Data Description

The total number of women who have never gotten pregnant and have diabetes is 31 where most of their ages are around 29 years old based on the age mean shown in the figure below (figure 3.4.1).

```
In [149]: notpregnant['Age'].mean()
Out[149]: 29.193548387096776

In [150]: notpregnant['Outcome'].sum()
Out[150]: 31
```

(Figure 3.4.1)

The figure below (figure 3.4.2) shows the total number of women that have been pregnant and also have diabetes are 211.

```
In [153]: pregnant['Outcome'].sum()
Out[153]: 211
```

(Figure 3.4.2)

The max glucose level is 194 for non-diabetic outcome. The maximum glucose level is at a normal range however the max insulin level exceeds the normal range of insulin levels which is 387.0. Although the insulin level exceeds the normal range, the diagram below *(figure 3.4.3)* shows that in some cases it does not fall under the diabetic outcome.

```
In [144]: nondiabetic.aggregate(['max'])
Out[144]:
```

	Glucose	Insulin	Outcome
max	194.0	387.0	0

(Figure 3.4.3)

The average blood pressure and BMI for the diabetic outcome are stated below (figure 3.4.4) which are 72.1273 and 32.1289 respectively. Based on the output, the average blood pressure of a diabetic outcome is still within the normal range. However, the average for BMI exceeds the normal range. The total number of diabetic cases that have normal BMI is 7. We can see that the BMI plays a crucial part in detecting diabetes.

```
In [147]: avgbp_bmi = subset3[['BloodPressure','BMI']].mean()
          avgbp_bmi

Out[147]: BloodPressure    72.137306
          BMI              32.128876
          dtype: float64

In [148]: print('Sum of normal BMI that have diabetes:')
          normalBMI['Outcome'].sum()

          Sum of normal BMI that have diabetes:

Out[148]: 7
```

(Figure 3.4.4)

For non-diabetic outcome, the max skin thickness outcome is 54.0 while the minimum skin thickness is 7.0, the diagram below *(figure 3.4.5)* shows that in some cases it does not fall under the diabetic outcome. In diabetic individuals, skin thickness is influenced by BMI and anatomical location. Skin thickness plays a big role in needle length selection for intradermal or subcutaneous injections, especially for auto-injectors.

```
In [151]: st_nondiabetic.aggregate(['max','min'])
```

Out[151]:

	SkinThickness	BMI	Outcome
max	54.0	47.9	0
min	7.0	18.2	0

(Figure 3.4.5)

Based on the output shown in *figure 3.4.6*, the maximum Diabetes Pedigree Function is from the age range of 61-70 years old, which is at age 66, while the minimum is from the age range of 21-30 years old, which is at age 21. The greater the number of the function, the more likely you are to get diabetes. Diabetes Pedigree Function is 0.4808 on average in diabetic patients.

```
In [39]: pedigree.aggregate(['max', 'min'])
```

Out[39]:

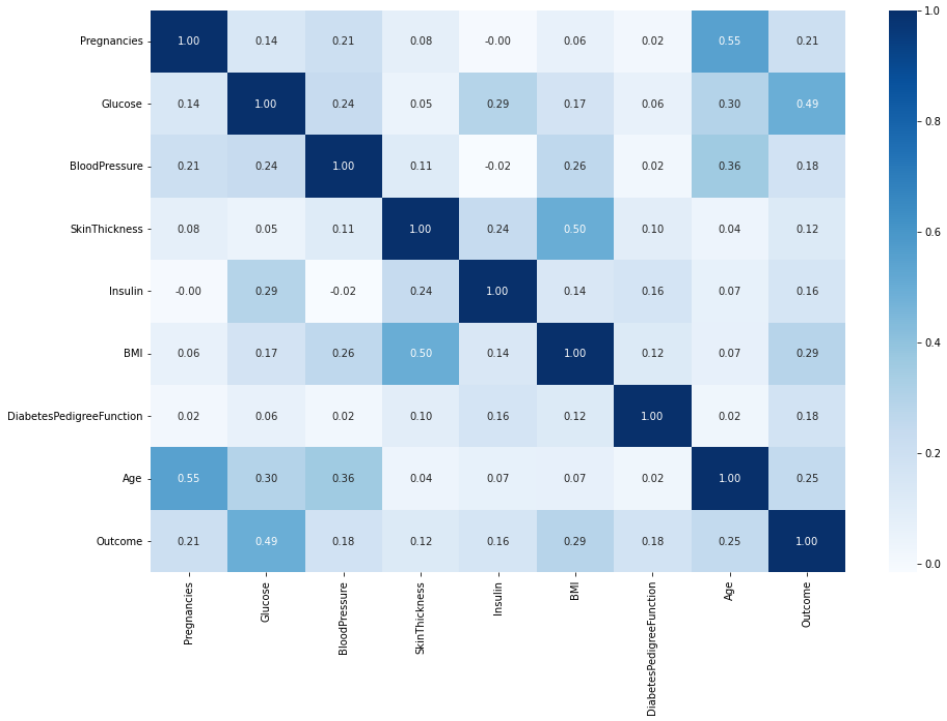
	DiabetesPedigreeFunction	Age	Outcome	ageClass
max	1.191	66	1	61-70
min	0.088	21	1	21-30

```
In [40]: pedigree['DiabetesPedigreeFunction'].mean()
Out[40]: 0.4807616580310883
```

(Figure 3.4.6)

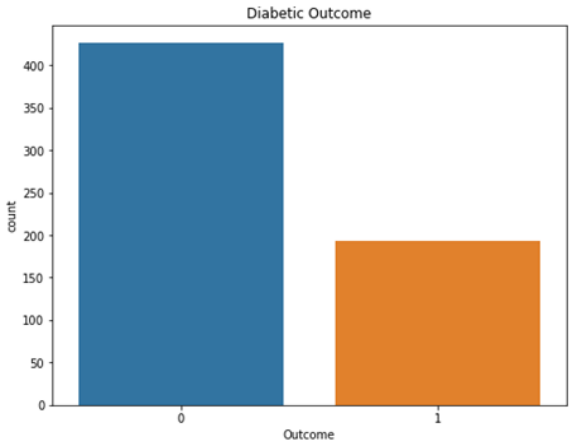
4.0 EXPLORATORY DATA ANALYSIS

1. Create a heatmap for all variables to see the correlation between variables so that we can do the analysis based on the strongest relation between variables.



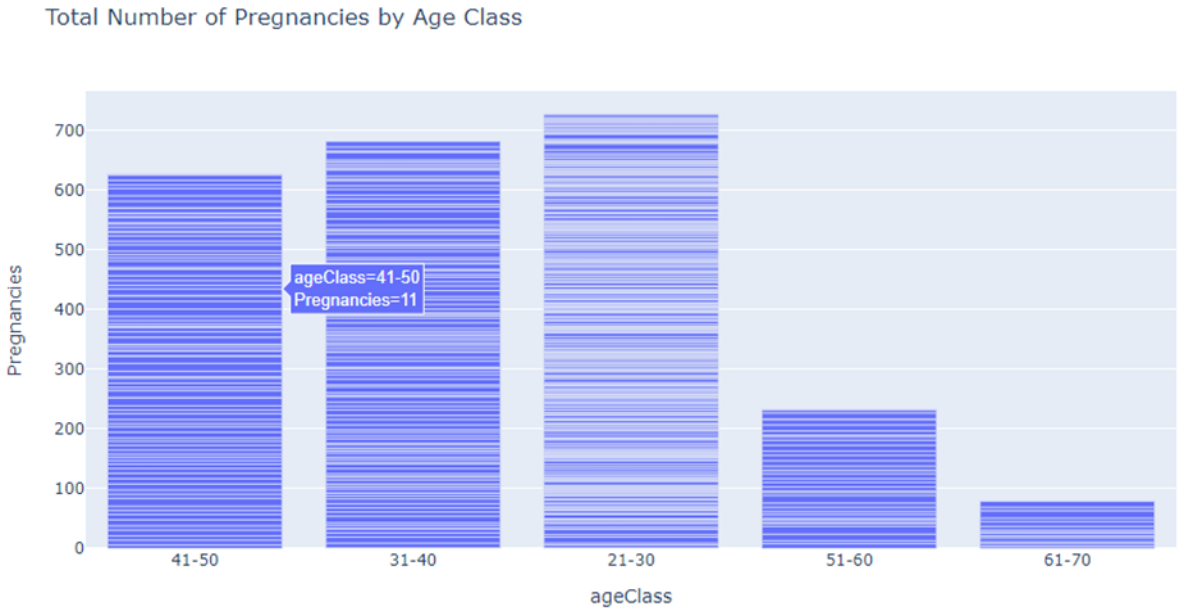
As we can see from the visualization above, there is a strong relationship between Age and Pregnancies, BMI and SkinThickness, Outcome and Glucose, Age and BloodPressure, and so on. Hence, in order to do the analysis, these strong relationships between variables must be taken into consideration.

2. Diabetic outcome in dataset using bar graph.



After the cleaning process, we only left with 619 data. Outcome = 1 indicates that the person has positive diabetes. Meanwhile, Outcome = 0 indicates that the person has negative diabetes. As we can see from the bar graph above, from 619 data, negative outcomes outweigh positive outcomes with 426 and 193 data respectively.

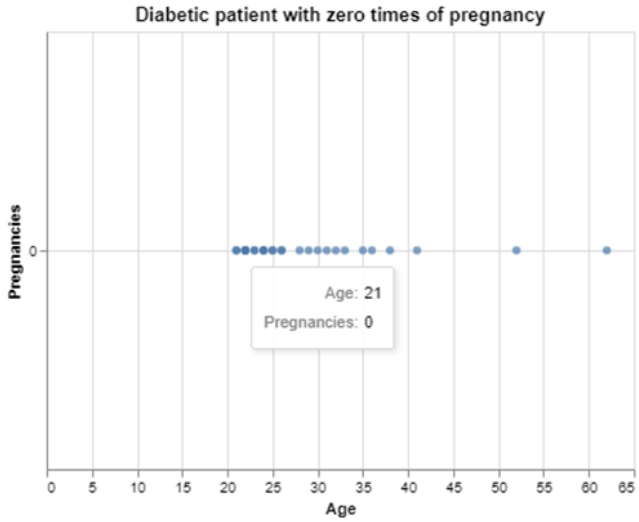
3. Cumulative number of pregnancies by age class using bar graph.



From the graph, the highest total number of pregnancies is from age class 21-30 with 728 times of pregnancies. Meanwhile the lowest total number of pregnancies is from age class 61-70 with 79 times of pregnancies only. This happens because the number of individu from age class 21-30 is higher than the number of individu from age class 61-70 which is 329 and 19 persons respectively.

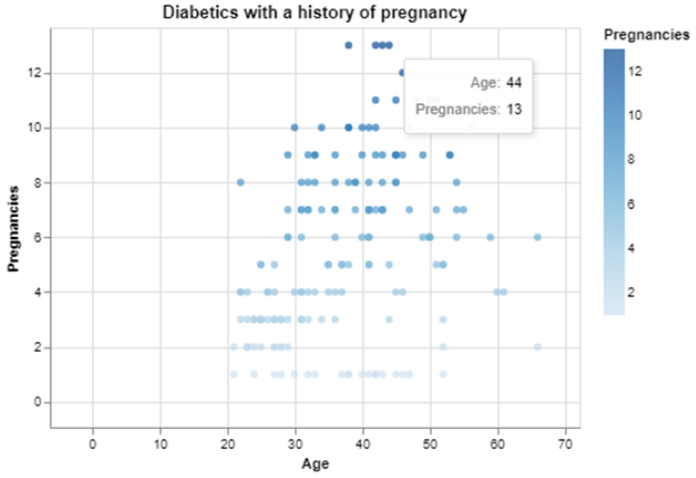
However, if we look at it in terms of individual pregnancies, the 41-50 age class has a higher individual pregnancy compared to the 21-30 age class. This is because the blue lines in the 41-50 age class are more concentrated than the 21-30 age class.

4. Diabetic patient with zero times of pregnancy using interactive scatter plot.



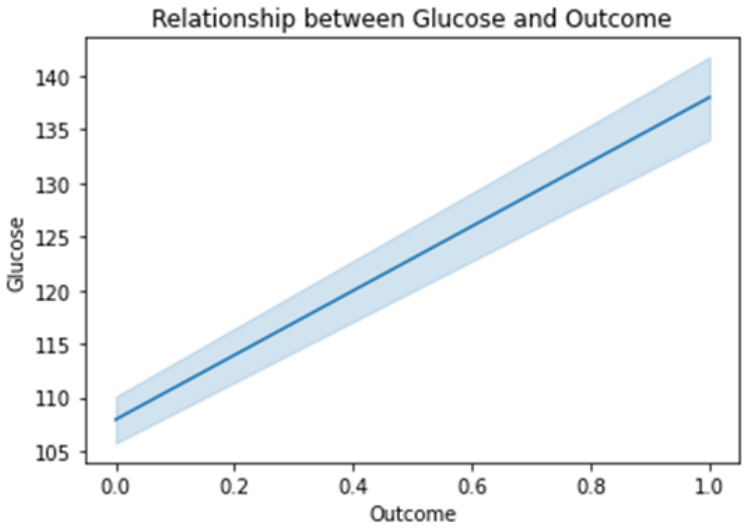
In this chart, we visualize the positive diabetes patients with zero times of pregnancy. We can assume Pregnancies = 0 means the person is either a single woman or has never been pregnant. The youngest diabetic patient with zero times of pregnancy in the data is 21 years old while the oldest is 62 years old. Patients in this category are mostly in the range of 21-30 years old.

5. Diabetics with a history of pregnancy using interactive scatter plot.



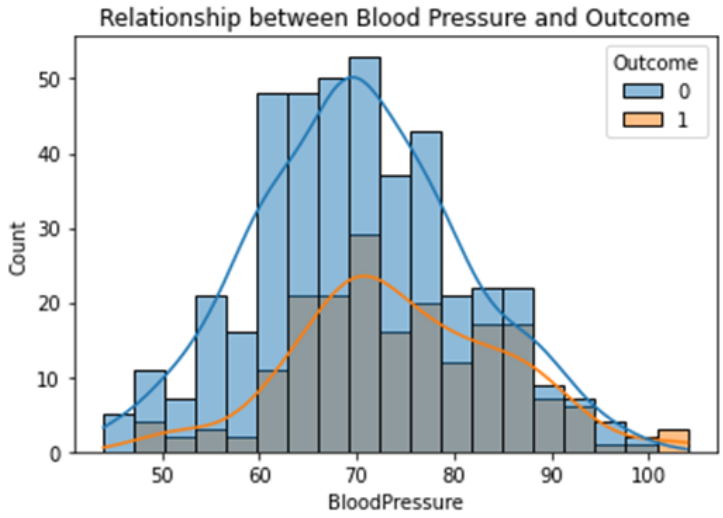
In this chart, we visualize the positive diabetes patients with a history of pregnancy. Many data concentrated in the age range of 30-50 years old with the number of individual pregnancies in the range of four to ten times. This shows that individuals aged 30-50 years old with many pregnancies are much more likely to get diabetes.

6. Relationship between glucose and outcome of diabetes using line chart.



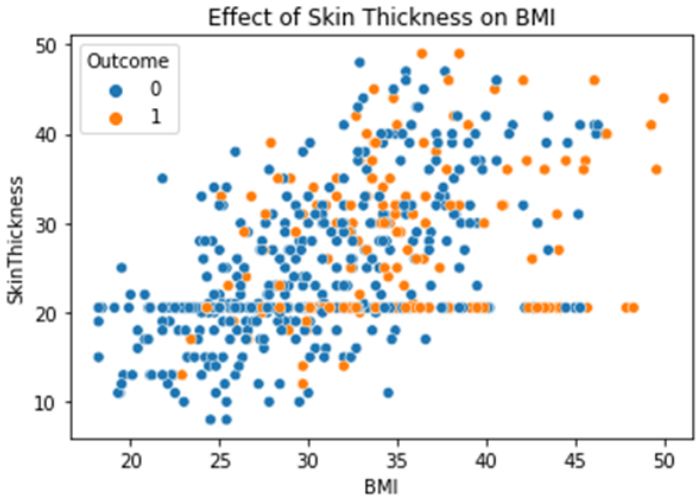
From the chart, there is a linear relationship between glucose content in individuals and the outcome of diabetes. Linear relationships are applied where one factor relies on another factor. In this situation, outcome of positive diabetes relies on the glucose content in body. The more glucose content in someone body, the chances of that person getting diabetes are also high.

7. Relationship between blood pressure and outcome of diabetes using multiple histograms.



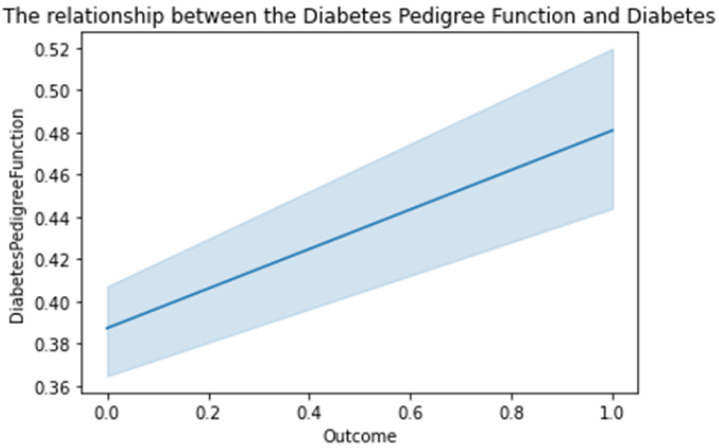
As we know, Outcome = 1 indicates that the person has positive diabetes. Meanwhile, Outcome = 0 indicates that the person has negative diabetes. Regardless of the outcome of diabetes, both outcomes show that the blood pressure of individuals from this dataset are mostly in the range of 60-90 mmHg.

8. Effect of skin thickness on BMI using scatter plot.



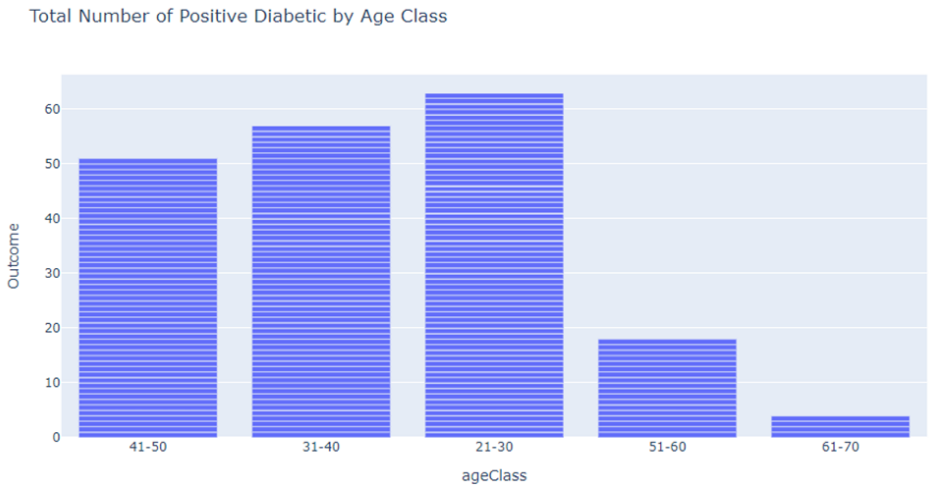
With the range of age between 21-70 years old from this dataset, the normal BMI for adults is in between 18.5 to 24.9 for ideal weight category and 25 to 29.9 for overweight weight category. Even Though it falls under the overweight category, it is still an average BMI that can be considered normal for adults. From the scatter plot above, we can see that most of the data fall under obesity BMI which is large or equal to 30. This happens because, as the skin thickness increases the weight of someone also increases and will affect their BMI. This shows that BMI is dependent on skin thickness as one of the factors that affect the BMI value.

9. Relationship between the diabetes pedigree function and diabetes using line charts.



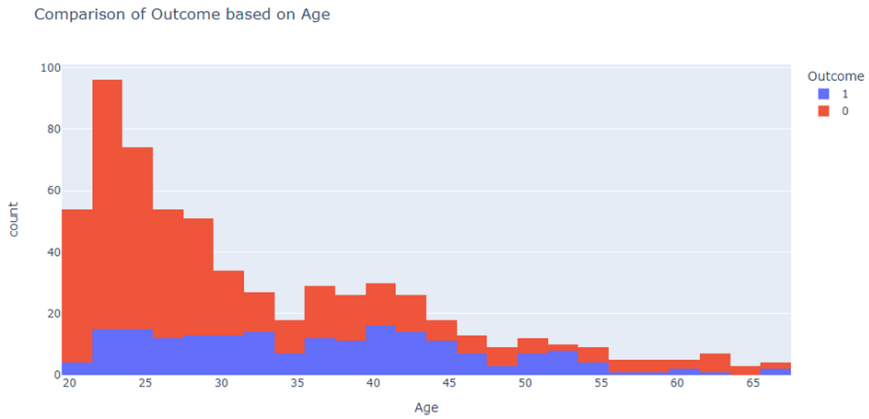
In this chart, there is a linear relationship between diabetes pedigree function and the outcome of the diabetes. The higher the value of diabetes pedigree function, the chances of that person getting diabetes are also high.

10. Total number of positive diabetic patients by age class using bar graph.



From the graph above, the highest cumulative total of positive diabetes patients are in the age class of 21-30 with 63 patients. Meanwhile the lowest cumulative total of positive diabetes are in the age class of 61-70 with 4 patients only.

11. Comparison of diabetes outcome based on age using multiple histograms graph.



From the graph above, the highest number of ages from negative diabetes are in the range of age within 22 to 23 years old and the lowest are in the range of age within 64 to 65 years old with 81 and 3 persons respectively.

Meanwhile, for positive diabetes disease, the highest number of age are in the range of age within 40 to 41 years old and the lowest is also in the range of age within 64 to 65 years old with 16 and 0 persons respectively.

5.0 SUMMARY

Float and integer attributes that have been used to complete the analysis are Pregnancies, Glucose, Blood Pressure, Skin Thickness, BMI, Diabetes Pedigree Function, Age, and Outcome. The used of this attributes are:

- Pregnancies - To see the relationship between the times of pregnancies for individuals and the chance of them to get diabetes. As we can see from the analysis, the more often a person gets pregnant, the higher their chances of getting diabetes.

- Glucose - Glucose is one of the strongest factors that affect the outcome of positive diabetes. This is because glucose is a type of sugar that we get from food and will be used by our body as energy. The excess glucose level that does not transform into energy will cause diabetes.

- Blood Pressure - From the correlations between variables that has been shown using heatmap, there is a weak relationship between blood pressure and outcome of diabetes. However, the use of this integer for the analysis is to see the average of blood pressure readings by all individuals involved in this study.

- Skin Thickness - Skin thickness has the weakest relationship with the outcome of positive diabetes. However, the use of this integer for the analysis is to see the effect of skin thickness on BMI as it has the strongest relation with the BMI. The thicker a person's skin is, the greater its effect on their BMI.

- BMI - To see the relationship between BMI and the outcome in terms of how many individuals get diabetes even with normal BMI because it is one of the strong factors that affect the outcome of positive diabetes. If the individual has a higher BMI than normal BMI, it will increase the risk of being diagnosed with diabetes mellitus complications.

- Diabetes Pedigree Function - To determine the minimum and maximum number of diabetes pedigree functions, as well as the likelihood of individuals developing diabetes. Diabetes pedigree function is defined by diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient, so the higher the diabetes pedigree function, the higher the chances of individuals getting diabetes.

- Age - To see the early and late age at which individuals get diabetes and in what range of age most and least individuals get diabetes.

- Outcome - This is to determine whether the individuals get diabetes or not. If the individual gets diabetes, the outcome is 1, but if not, the outcome will be 0.

From the results obtained, we can conclude that the highest number of pregnancies is from the age range of 21–30 years old. The highest number of non-diabetic patients is at age 22–23, with a total of 81 patients, while the lowest is at age 64–65, with only three non-diabetic patients. Then, the highest total of diabetic patients is at age 40–41, which has a total of 16 patients, while the lowest is at age 64–65, where there are no diabetic patients. Individual pregnancies are higher in the 41-50 age group compared to the 21-30 age group. Furthermore, individuals aged 30-50 years old with a history of multiple pregnancies are much more likely to develop diabetes. Then it was proven that the more glucose in your body, the more likely you are to develop diabetes. Other than that, the higher the value of the diabetes pedigree function, the greater the likelihood of that person developing diabetes.

6.0 REFERENCES

1. Better Health Channel. (2021, October 17). Diabetes and insulin - Better Health Channel. Better Health. Retrieved May 15, 2022, from https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/diabetes-and-insulin
2. Bhandari, P. (2022, May 20). How to find and remove outliers. Scribbr. https://www.scribbr.com/statistics/outliers/
3. Collier, A., Patrick, A. W., Bell, D., Matthews, D. M., Maclntyre, C. C. A., Ewing, D. J., & Clarke, B. F. (1989). Relationship of Skin Thickness to Duration of Diabetes, Glycemic Control, and Diabetic Complications in Male IDDM Patients. Diabetes Care, 12(5), 309–312. https://doi.org/10.2337/diacare.12.5.309
4. Joshi, G. (2022, January 4). Diabetes Prediction using Machine Learning — Python. Medium. Retrieved May 15, 2022, from https://medium.com/geekculture/diabetes-prediction-using-machine-learning-python-23fc98125d8
5. S. (2019, November 12). What's the definition of Diabetes Pedigree Function? · Issue #26 · susanli2016/Machine-Learning-with-Python. GitHub. Retrieved May 8, 2022, from https://github.com/susanli2016/Machine-Learning-with-Python/issues/26
6. Vieira, G. (2020, March 19). Diabetes and High Blood Pressure: What is the relationship? Diabetes Strong. Retrieved May 15, 2022, from https://diabetesstrong.com/diabetes-and-high-blood-pressure/

Google drive link for the dataset:

https://drive.google.com/drive/folders/1BG4Cw_EvqS3tYIxQ_znYtd7js5gen0fr?usp=sharing