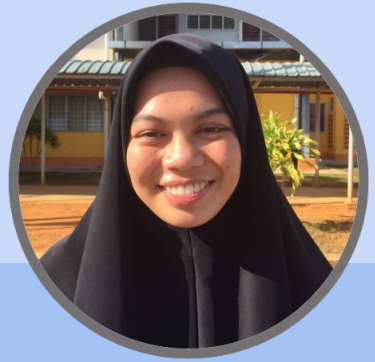
DIABETES DISEASE PREDICTION

SARAH BATRISYIA BINTI MORSHIDI (SD20008)
SITI NUR AISYAH BINTI ANUAR (SD20009)
NIK NUR AIN BINTI NIK JID (SD20022)
NIK NURUL SYUHADA BINTI MOHD ALI (SD20034)
MUHAMMAD ISYHRAF BIN AZMIN (SD20065)

GROUP MEMBERS



AISYAH

ISYHRAF

SYUHADA

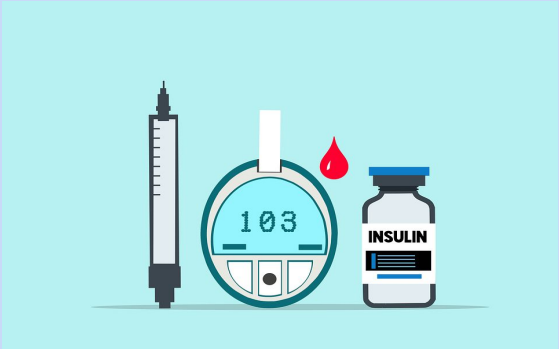
SARAH

NIK

SYNOPSIS

Description of the assignment

- Our group choose Diabetes Disease Detection 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.
- There are two types of diabetes which are Type 1 Diabetes and Type 2 Diabetes.
- TYPE 1 DIABETES : serious condition where your blood glucose (sugar) level is too high because your body cannot make a hormone called insulin.
- TYPE 2 DIABETES : a serious condition where the insulin your pancreas makes cannot work properly, or your pancreas cannot make enough insulin.



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.

Question to be answered

- 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?

Objectives

- 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.

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).

Packages Required

Packages	Packages Function
Pandas	- Read csv file - Data frame df - Data loc - Fillna - Desc data - Head()
Numpy	- Mean - Sort
Scipy	- Stats.iqr
Matplotlib.pyplot	- Plt.figure
Seaborn	- Sns.heatmap - Sns.countplot - Sns.lineplot - Sns.hisplot - Sns.scatterplot
Altair	- Alt.chart
Plotly.express	- Px.bar - Px.histogram

DATA PREPARATION

- IMPORT DATA
- FIND MISSING VALUES
- REPLACE MISSING VALUES USING MEAN
- DETECT AND REMOVE OUTLIER
- GENERATING PAIRWISE CORRELATION
- PREVIEW DATA USING SUBSET & GROUPBY
- MAKE DATA ANALYTICS USING AGGREGATION
- MAKE DATA VISUALIZATION

IMPORT DATA

```
Diabetes= pd.read_csv("diabetes.csv")
pd.set_option('display.max_rows', None)
#pd.set_option('display.max_columns', None)
Diabetes
```

	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

FIND NULL VALUES

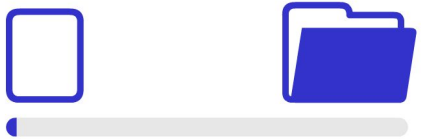
```
Diabetes.isnull()
```

	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

```
Diabetes.isna().sum()
```

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

DATA IMPUTATION



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

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

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

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


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

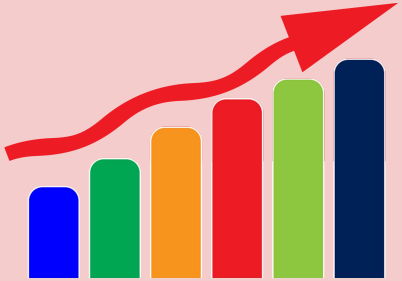
```
     Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
0              6  148.000000      72.000000      35.000000   79.799479
1              1   85.000000      66.000000      29.000000   79.799479
2              8  183.000000      64.000000      20.536458   79.799479
3              1   89.000000      66.000000      23.000000   94.000000
4              0  137.000000      40.000000      35.000000  168.000000
5              5  116.000000      74.000000      20.536458   79.799479
6              3   78.000000      50.000000      32.000000   88.000000
7             10  115.000000      69.105469      20.536458   79.799479
8              2  197.000000      70.000000      45.000000  543.000000
9              8  125.000000      96.000000      20.536458   79.799479
10             4  110.000000      92.000000      20.536458   79.799479
11            10  168.000000      74.000000      20.536458   79.799479
12            10  139.000000      80.000000      20.536458   79.799479
13             1  189.000000      60.000000      23.000000  846.000000
14             5  166.000000      72.000000      19.000000  175.000000
15             7  100.000000      69.105469      20.536458   79.799479
16             0  118.000000      84.000000      47.000000  230.000000
```

SUMMARY STATISTICS OF DATAFRAME

```
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



DETECT AND REMOVE OUTLIERS

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)
```

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
```

PAIRWISE CORRELATION OF THE DATASET

```
Diabetes.corr()
```

	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

```python
# 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
```

```python
# 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
```

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	

```python
age1=data_clean1.groupby('ageClass').sum('Pregnancies')
age1
```

ageClass	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
21-30	728	38312.683594	23669.320312	8598.208333	30960.716146	10723.340625	137.102	8480	63
31-40	682	15617.894531	9376.738281	3317.359375	12022.559896	4038.900000	61.377	4483	57
41-50	627	11511.894531	7112.421875	2470.458333	8566.166667	3180.600000	37.033	4114	51
51-60	234	4935.000000	2947.000000	873.729167	3694.585938	1121.192578	15.791	2023	18
61-70	79	2238.000000	1230.000000	369.901042	1487.192708	466.300000	6.425	1018	4

SUBSET 1

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

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

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

	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

```
notpregnant['Age'].mean()
```

```
29.193548387096776
```

```
notpregnant['Outcome'].sum()
```

```
31
```


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

	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

```
pregnant['Outcome'].sum()
```

```
211
```

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

	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

SUBSET 2

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

	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

EAT
SLEEP
INSULIN
REPEAT

```
nondiabetic.aggregate(['max'])
```

	Glucose	Insulin	Outcome
max	194.0	387.0	0

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

	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

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

	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

SUBSET 3

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

```
BloodPressure    72.137306
BMI              32.128876
dtype: float64
```

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

```
Sum of normal BMI that have diabetes:

7
```

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

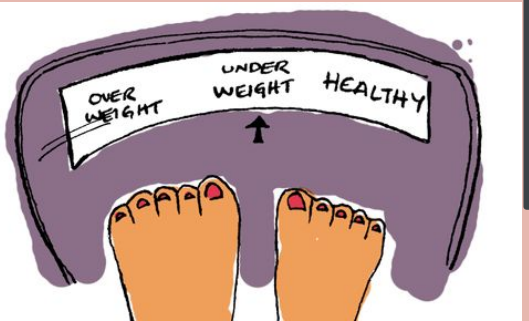
	SkinThickness	BMI	Outcome
0	35.000000	33.6	1
1	29.000000	26.6	0
2	20.536458	23.3	1
3	23.000000	28.1	0
5	20.536458	25.6	0

SUBSET 4

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

	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




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

	SkinThickness	BMI	Outcome
max	54.0	47.9	0
min	7.0	18.2	0

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

	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

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

	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

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

ageClass	DiabetesPedigreeFunction	Age	Outcome
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

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

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

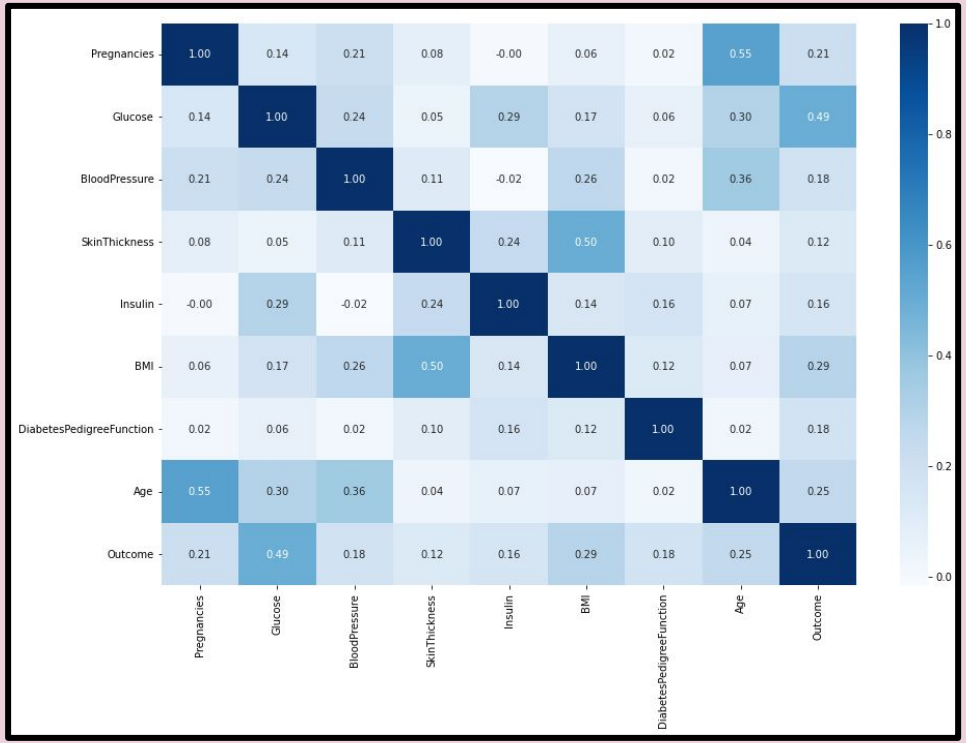
```
pedigree['DiabetesPedigreeFunction'].mean()
```

```
0.4807616580310883
```

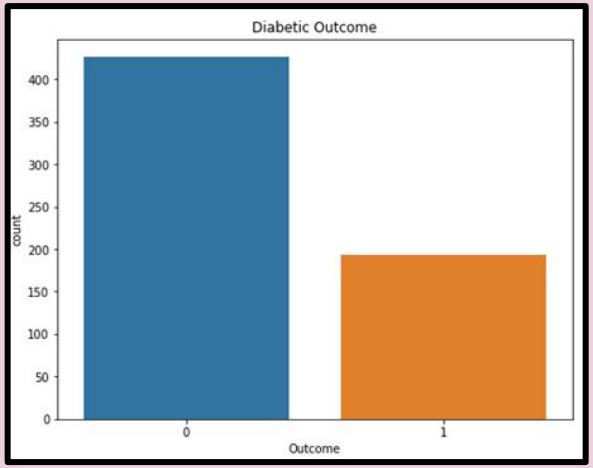
SUBSET 5

EXPLORATORY DATA ANALYSIS

I. *Visualization of pairwise correlation of the dataset*



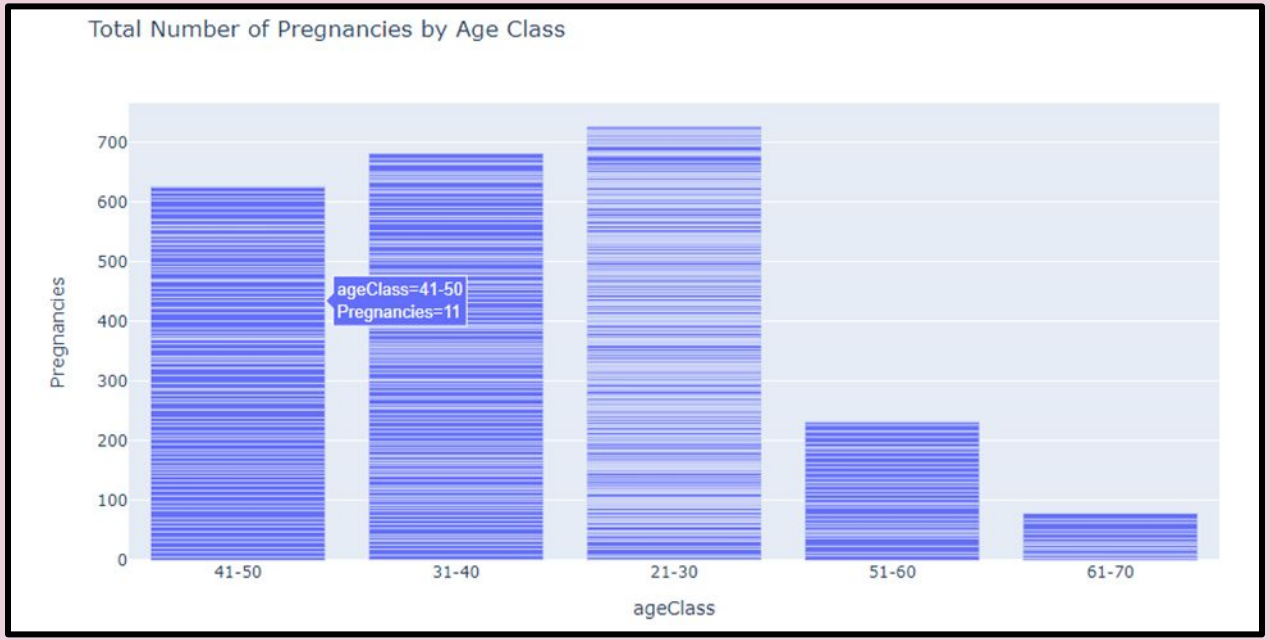
2. Diabetic outcome from dataset



- **Negative diabetes outweigh positive diabetes with 426 and 193 data respectively.**

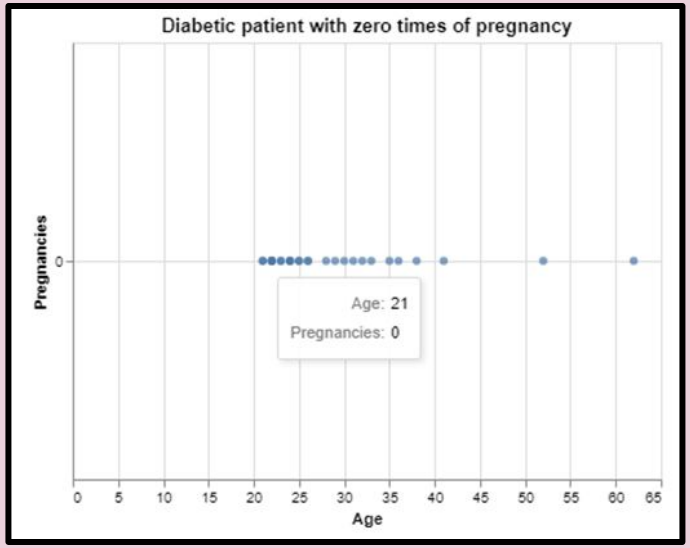
PREGNANCY FACTORS

3. Cumulative number of pregnancies by age class



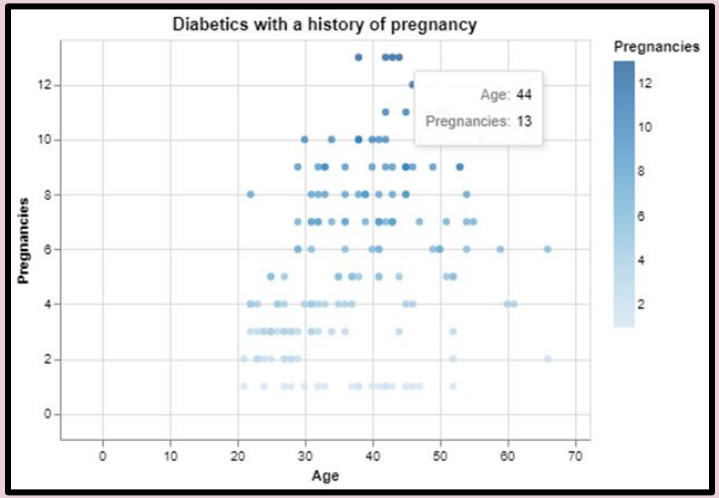
- **Highest total number of pregnancies is from age class 21-30 with 728 times of pregnancies.**
- **Lowest total number of pregnancies is from age class 61-70 with 79 times of pregnancies.**

4. Diabetics with zero times of pregnancy



- Youngest patient is 21 years old while the oldest is 62 years old.
- Patients mostly in the range of 21-30 years old.

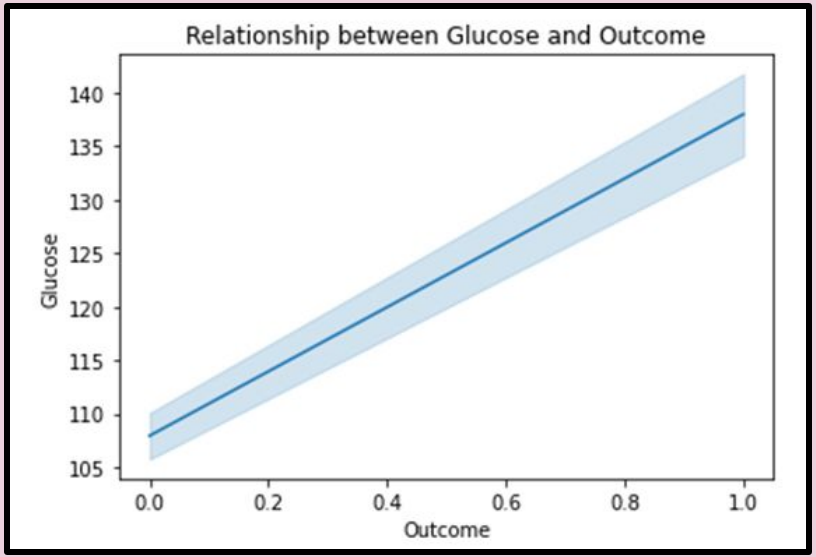
5. Diabetics with a history of pregnancy



- 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.

GLUCOSE FACTORS

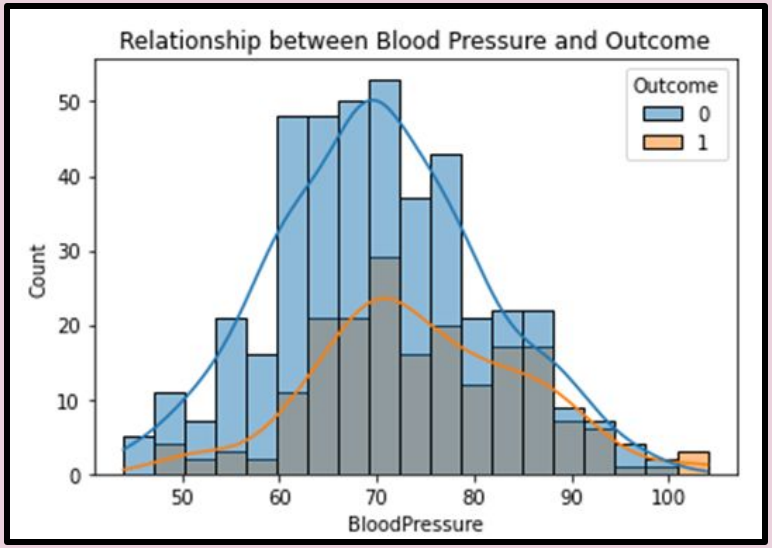
6. Relationship between glucose and outcome of diabetes



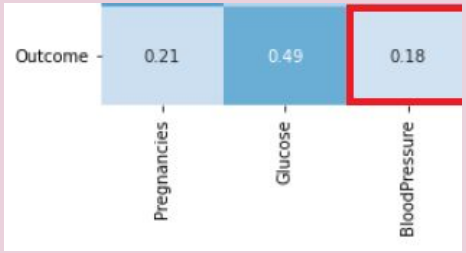
- *There is a linear relationship between glucose content and the outcomes of diabetes.*
- *The higher glucose level, the chances of getting diabetes are also high.*

BLOOD PRESSURE FACTORS

7. Average of blood pressure readings in dataset



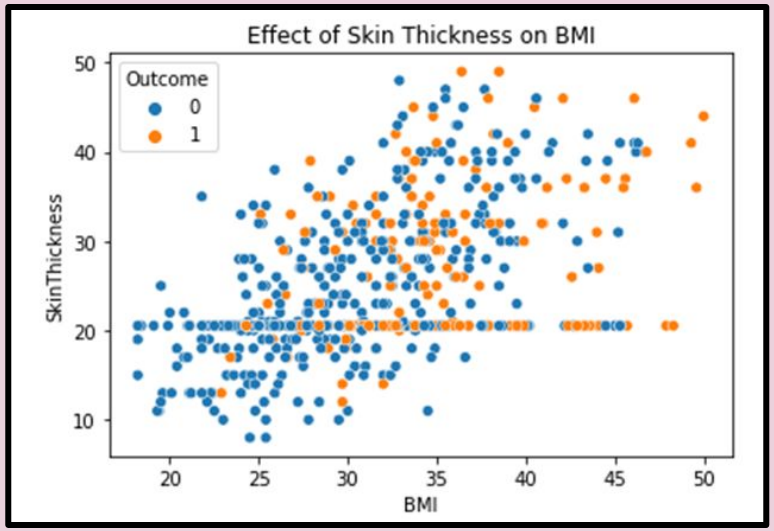
- ***There is weak relationship between blood pressure and outcome of diabetes.***



- ***Average blood pressure from dataset in the range of 60-90 mmHg.***

SKIN THICKNESS AND BMI FACTORS

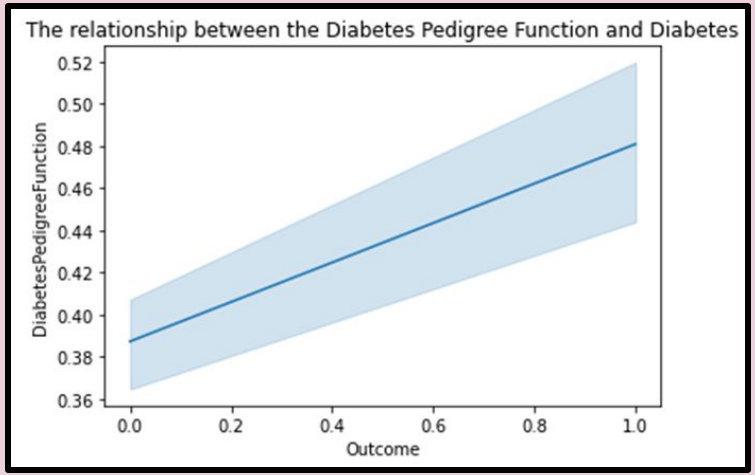
8. Effect of skin thickness on BMI



- Skin thickness has the strongest relation with BMI.
- As the skin thickness increases, the BMI also will increase.
- Positive diabetes outcomes are more concentrated in the obesity category (BMI >= 30)

PEDIGREE FACTORS

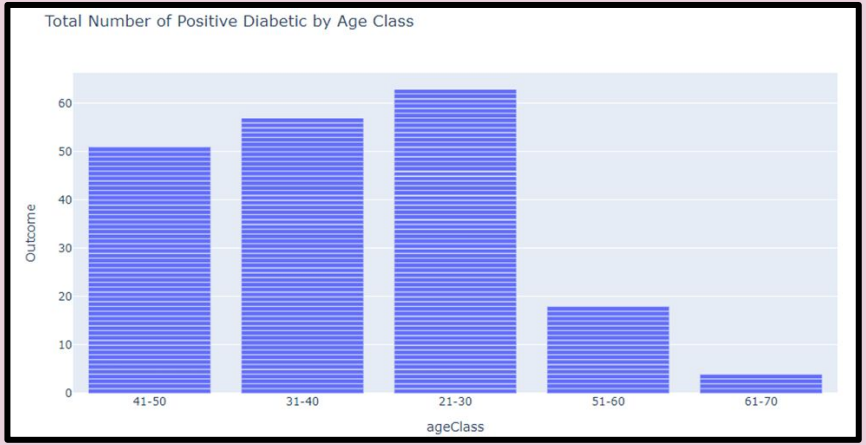
9. Relationship between diabetes pedigree and outcome of diabetes



- *There is a linear relationship between diabetes pedigree function and the outcome of diabetes.*
- *The higher the value of diabetes pedigree function, the chances of getting diabetes are also high.*

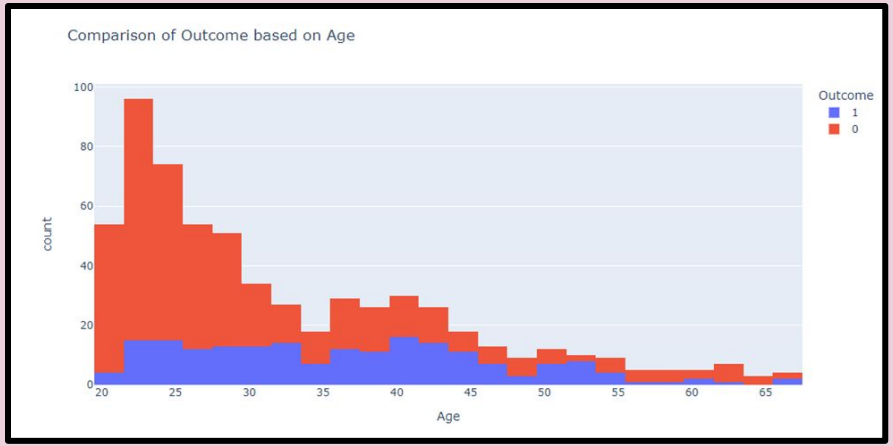
AGE FACTORS

10. Positive diabetes patients by age class

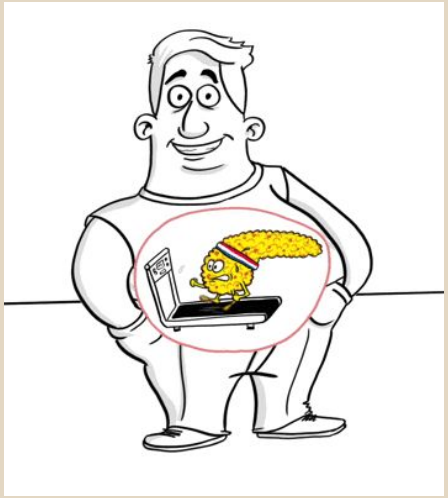


- Highest total number of patients is from age class 21-30 with 63 patients.
- Lowest total number of patients is from age class 61-70 with 4 patients.

11. Comparison of outcome based on age



- Highest age from negative diabetes are from age 22 to 23 years old with 81 patients.
- Highest age from positive diabetes are from age 40 to 41 years old with 16 patients.



SUMMARY

Highest number of diabetic patients :
age of **40-41 YEARS OLD**

Highest number of non-diabetic patients :
age **22–23 YEARS OLD**

↑ the value of the diabetes pedigree function and BMI, the greater the likelihood of that person developing diabetes.

Individuals aged **30-50 YEARS OLD** with a history of multiple pregnancies are <u>much more likely</u> to develop diabetes.

Glucose is **1** of the strongest factors that affect the outcome of positive diabetes.

THANK YOU