Question1: What is the correlation between features in the dataset. and what two features have very strong correlation with the independent variable? Justify with reason

When two sets of data are strongly linked together we say they have a High Correlation .The word correlation (Co means together hence it is together relation) is Positive when the values increase together and correlation is Negative when one value decreases as the other increase. example : Set of Icecream sales vs Set of ice cream temperature.

Pandas(loc and iloc) and Numpy are the 2 features have very strong correlation with independent variable.

```
!pip install pandas
```

```
import pandas as pd
import numpy as np
```

```
data = {'Shanmukh': pd.Series([60,40,30,15]),
        'Pawan ': pd.Series([60,50,30,15])}
```

```
df = pd.DataFrame(data)
```

```
df
```

```
item = {'Shanmukh': pd.Series([60,40,30,15], index=['English', 'Maths', 'Kannada', 'Science']),
        'Pawan': pd.Series([60,50,30,15], index=['English', 'Maths', 'Kannada', 'Science' ])}
```

```
cart = pd.DataFrame(item)
cart
```

```
cart.iloc[[1,2,3]]
```

```
cart.loc[['Maths','Science']]
```

Question 2: Which feature has more outliers. Explain with a visualization.

Pandas/Data frames and some more are the features which have more outliers.

```
import matplotlib.pyplot as plt
import numpy as np



x=np.arange(0,10)
y=np.arange(10,20)


y
```

```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
x
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
##plotting using matplotlib

##plt scatter
plt.scatter(x,y,c='g')
plt.xlabel('no of students')
plt.ylabel('subject no')
plt.title('average students progress')
plt.savefig('Test.png')
plt.plot(x,y)
```

```
plt.bar(x, y, color = 'b')
plt.title('Bar graph')
```

```
data = 'roses', 'jasmine', 'lilly', 'hibiscus'
sizes = [150, 100, 50, 20]
plt.pie(sizes, labels=data)
```

```
data = np.array([1,2,3,4,5,6,7,8])
plt.boxplot(data,vert=True,patch_artist=True)
```

Question3: In which Age group Majority of people have diabetes? Make a visualization to validate your finding.

Link : https://raw.githubusercontent.com/plotly/datasets/master/diabetes.csv

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

```
df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/diabetes.csv')
```

```
df.head(3)
```

```
df.tail()
```

```
df.ndim
```

```
2
```

```
df.shape
```

```
(768, 9)
```

```
df.columns
```

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

```
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
```

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

```
0
```

```
df.describe()
```

```
sns.heatmap(df.isnull())
```

```
sns.countplot('Outcome',data=df)
```

```
sns.pairplot(df)
```

```
sns.boxplot(x="Pregnancies",y="Age",data=df,hue="Outcome")
```

```
sns.countplot(x="Pregnancies",data=df)
```

```
df['Glucose'].value_counts()
```

```
100    17
99     17
129    14
125    14
111    14
       ..
177     1
172     1
169     1
160     1
199     1
Name: Glucose, Length: 136, dtype: int64
```

```
base_color = sns.color_palette()[1]
gen_order = df['Glucose'].value_counts().index
sns.countplot(data = df, x = 'Glucose', color = base_color,
              order = gen_order)
```