Practical 5

Aditi Kharche

Div: C (C2 Batch)

Roll No.:328

PRN NO.: 202201060050

Select any one real-life dataset. Perform data analysis. Identify 10 grains for a given dataset. Develop an interactive dashboard using the matplotlib/Seaborn library. (Use any 10 different graphs with proper titles, legends, axis names, etc. to map identified grains

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

df = pd.read_csv('/content/diabetes.csv')

# Display the first few rows of the dataset
print(df.head())

# Get statistical summary of the dataset
print(df.describe())

# Check the correlation between variables
print(df.corr())
```

Output:

```
frequency  Pregnancies  Glucose  blood pressure  Skin Thickness
insulin  \
0          1           85       66              29                0
26.6
1          8          183       64               0                0
23.3
2          1           89       66              23               94
28.1
3          0          137       40              35              168
43.1
4          5          116       74               0                0
25.6


     BMI   Age  outcome
```

```
0  0.351   31        0

1  0.672   32        1

2  0.167   21        0

3  2.288   33        1

4  0.201   30        0
```

```
       frequency  Pregnancies    Glucose  blood pressure  Skin
Thickness  \

count  20.000000     20.00000  20.000000       20.000000
20.000000

mean    4.350000    128.30000  62.500000       18.100000
127.900000

std     3.558163     35.08651  26.729641       18.087216
215.584371

min     0.000000     78.00000   0.000000        0.000000
0.000000

25%     1.000000    106.00000  57.500000        0.000000
0.000000

50%     3.500000    117.00000  70.000000       21.000000
41.500000

75%     7.250000    145.75000  75.500000       32.750000
169.750000

max    10.000000    197.00000  96.000000       47.000000
846.000000
```

```
         insulin        BMI        Age    outcome

count  20.000000  20.000000  20.000000  20.000000

mean   31.235000   0.515500  36.300000   0.600000

std     9.819491   0.514888  11.420665   0.502625

min     0.000000   0.134000  21.000000   0.000000

25%    26.975000   0.198500  30.000000   0.000000

50%    30.300000   0.374500  32.000000   1.000000

75%    37.700000   0.560000  38.250000   1.000000

max    45.800000   2.288000  59.000000   1.000000
```

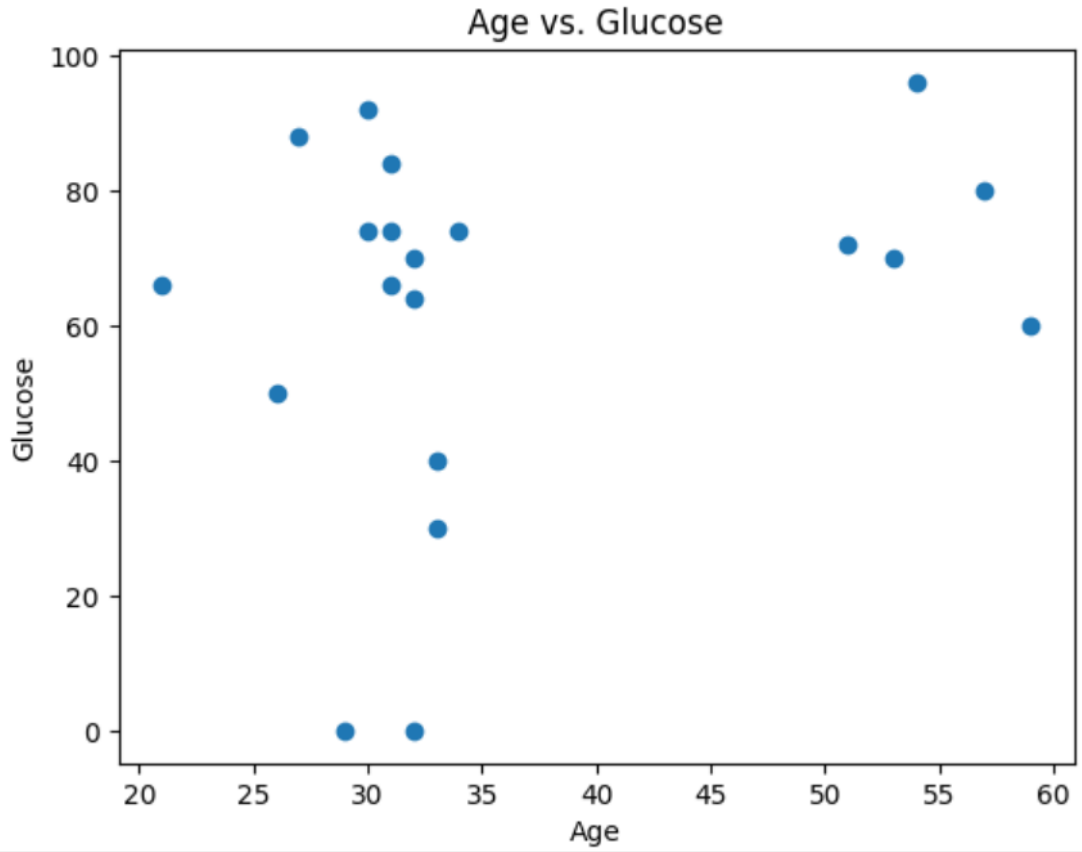
```
                frequency  Pregnancies   Glucose  blood pressure  \

frequency        1.000000     0.171963 -0.086051       -0.840456

Pregnancies      0.171963     1.000000  0.201862       -0.016554

Glucose         -0.086051     0.201862  1.000000        0.054541
```

blood pressure	-0.840456	-0.016554	0.054541	1.000000
Skin Thickness	-0.485660	0.572709	0.073333	0.516354
insulin	-0.403474	-0.082799	-0.264520	0.478119
BMI	-0.052472	0.203137	-0.025878	0.115443
Age	0.177310	0.653732	0.239822	-0.072004
outcome	-0.005886	0.427968	0.015670	0.079893

	Skin Thickness	insulin	BMI	Age	outcome
frequency	-0.485660	-0.403474	-0.052472	0.177310	-0.005886
Pregnancies	0.572709	-0.082799	0.203137	0.653732	0.427968
Glucose	0.073333	-0.264520	-0.025878	0.239822	0.015670
blood pressure	0.516354	0.478119	0.115443	-0.072004	0.079893
Skin Thickness	1.000000	0.147611	-0.006341	0.509929	0.296872
insulin	0.147611	1.000000	0.245479	-0.435906	-0.138843
BMI	-0.006341	0.245479	1.000000	0.146527	0.152935
Age	0.509929	-0.435906	0.146527	1.000000	0.297068
outcome	0.296872	-0.138843	0.152935	0.297068	1.000000

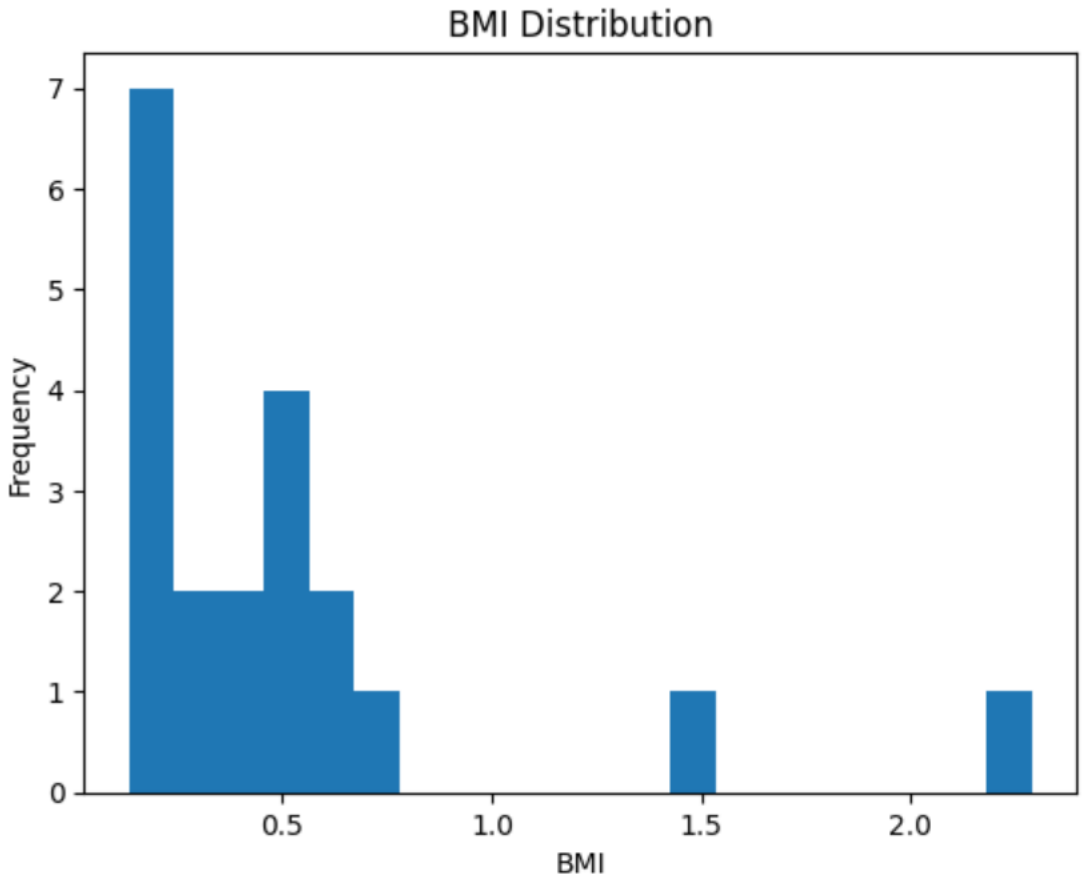
```
# Scatter plot
plt.scatter(df['Age'], df['Glucose'])
plt.title('Age vs. Glucose')
plt.xlabel('Age')
plt.ylabel('Glucose')
plt.show()
```

Output:




```
# Histogram
plt.hist(df['BMI'], bins=20)
plt.title('BMI Distribution')
plt.xlabel('BMI')
plt.ylabel('Frequency')
plt.show()
```

Output:

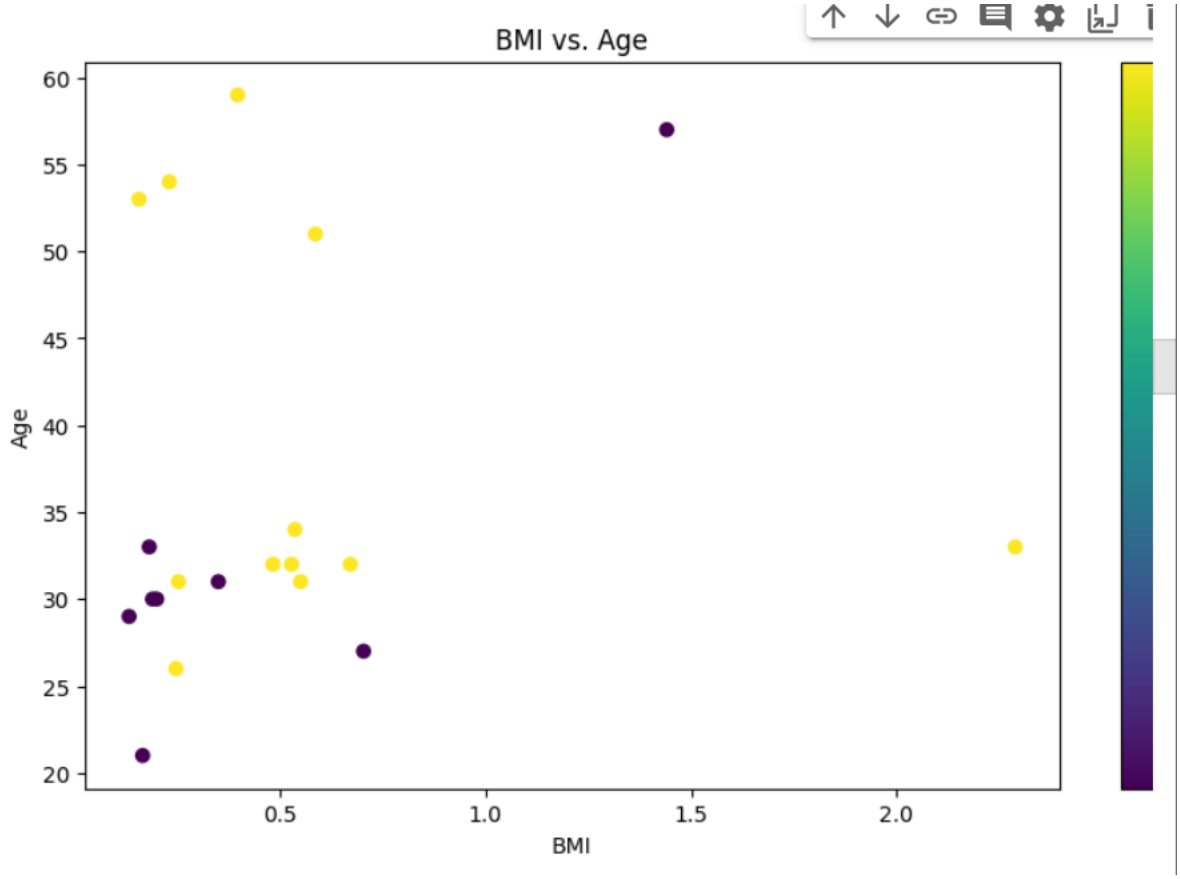



```
plt.figure(figsize=(10, 6))
sns.lineplot(x='Age', y='Glucose', data=df, hue='outcome')
plt.title('Glucose Levels by Age')
plt.xlabel('Age')
plt.ylabel('Glucose')
plt.legend(title='outcome')
plt.show()
```

Output:

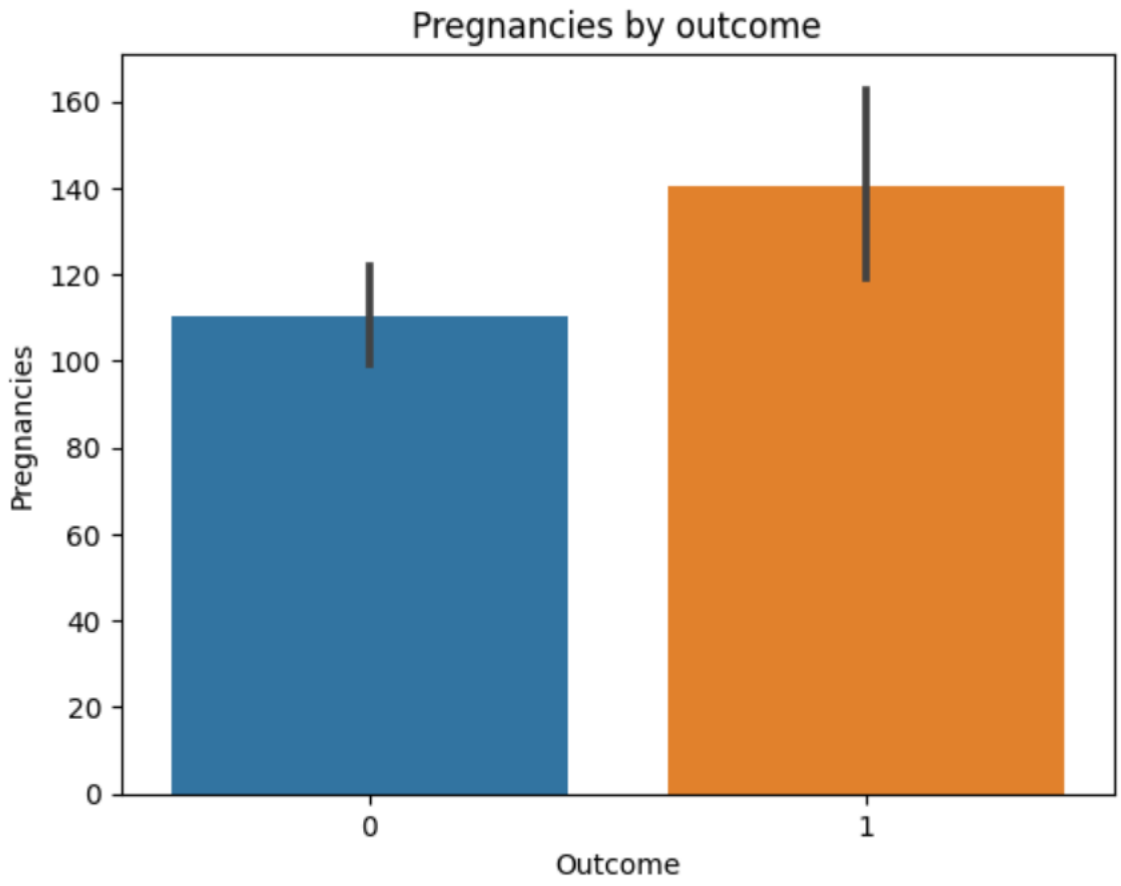

```
plt.figure(figsize=(10, 6))
plt.scatter(df['BMI'], df['Age'], c=df['outcome'], cmap='viridis')
plt.title('BMI vs. Age')
plt.xlabel('BMI')
plt.ylabel('Age')
plt.colorbar(label='outcome')
plt.show()
```

Output:



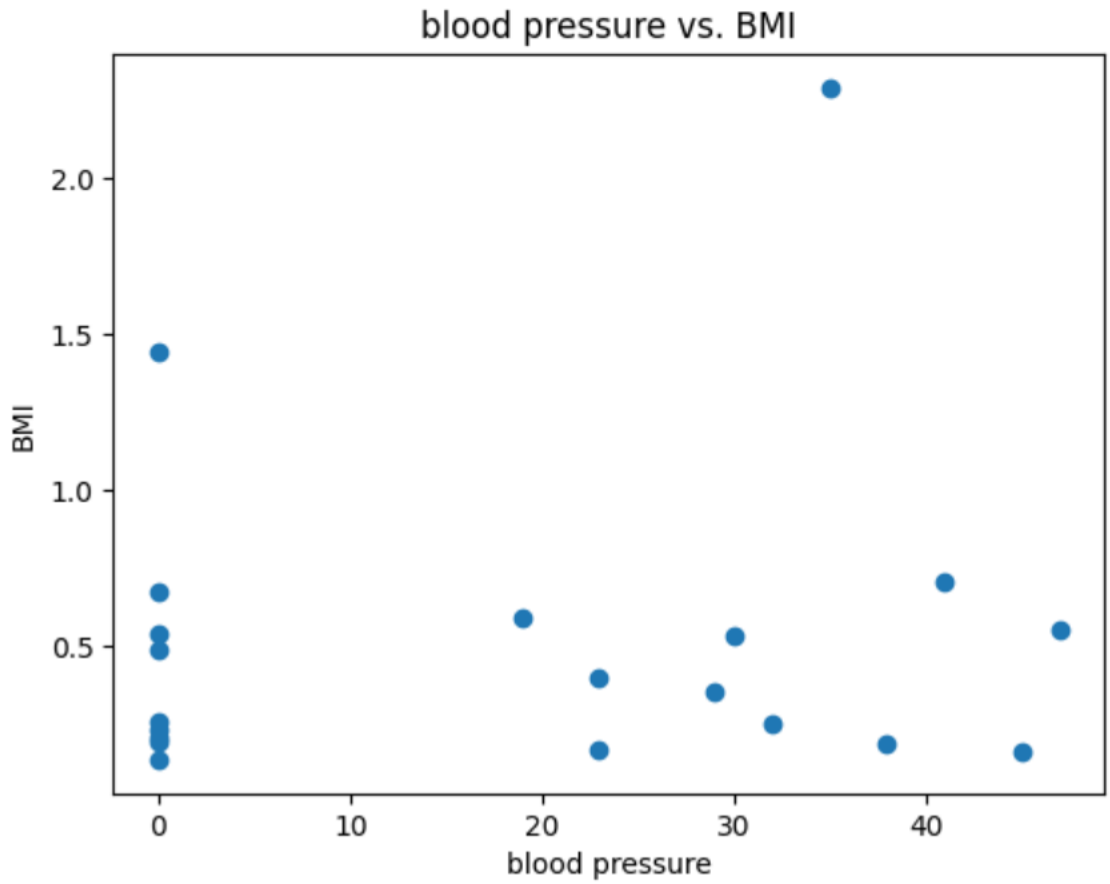

```
# Bar plot
sns.barplot(x='outcome', y='Pregnancies', data=df)
plt.title('Pregnancies by outcome')
plt.xlabel('Outcome')
plt.ylabel('Pregnancies')
plt.show()
```

Output:



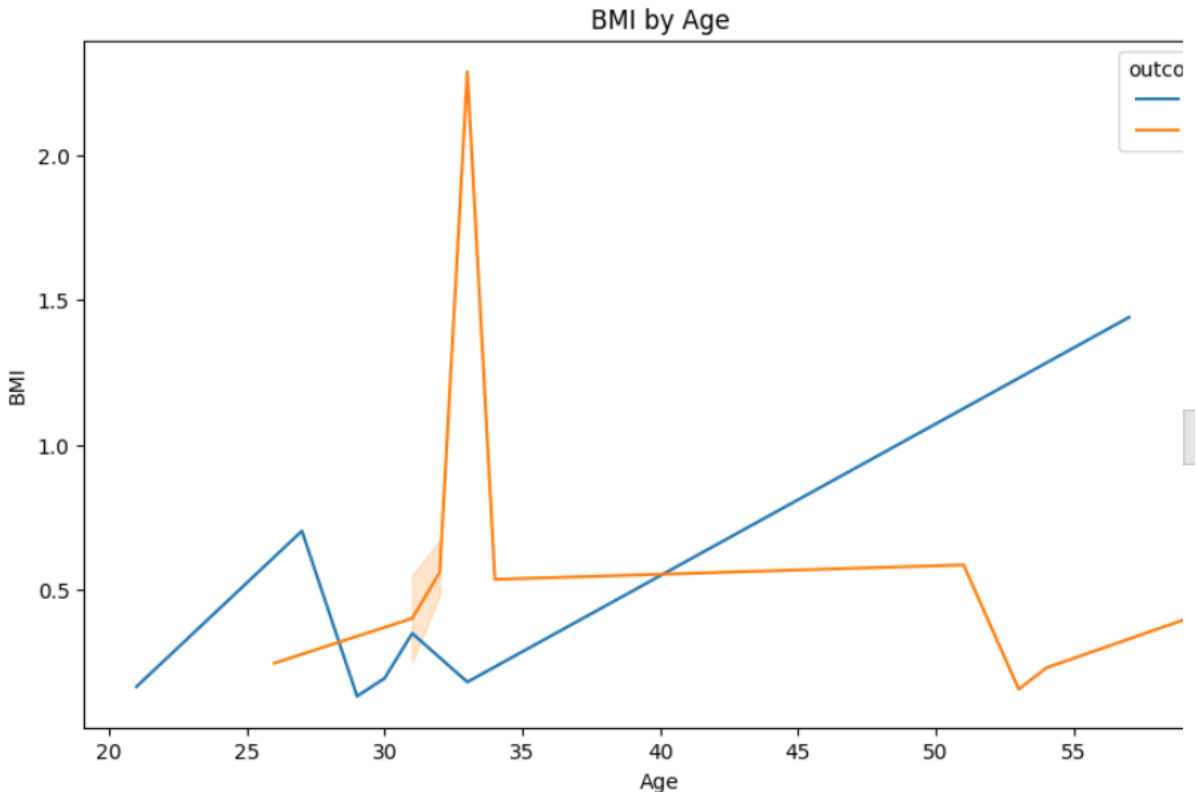

```
# Scatter plot
plt.scatter(df['blood pressure'], df['BMI'])
plt.title('blood pressure vs. BMI')
plt.xlabel('blood pressure')
plt.ylabel('BMI')
plt.show()
```

Output:



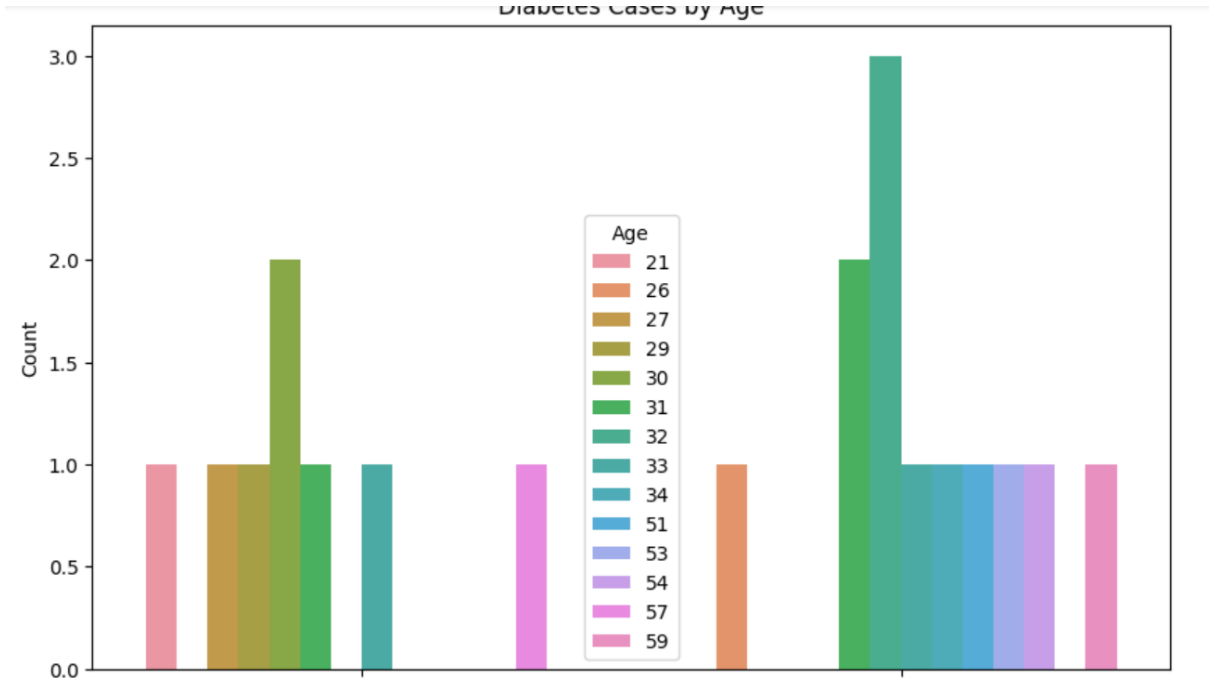

```
plt.figure(figsize=(10, 6))
sns.lineplot(x='Age', y='BMI', data=df, hue='outcome')
plt.title('BMI by Age')
plt.xlabel('Age')
plt.ylabel('BMI')
plt.legend(title='outcome')
plt.show()
```

Output:



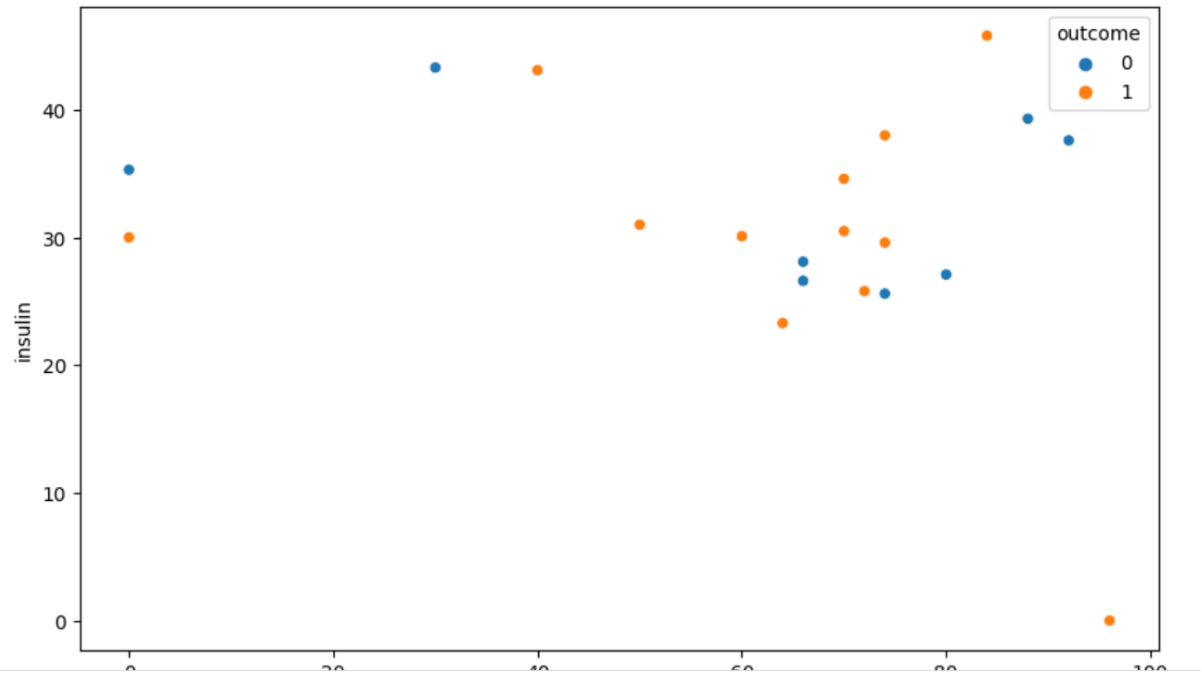

```
plt.figure(figsize=(10, 6))
sns.countplot(x='outcome', hue='Age', data=df)
plt.title('Diabetes Cases by Age')
plt.xlabel('outcome')
plt.ylabel('Count')
plt.legend(title='Age')
plt.show()
```

Output:



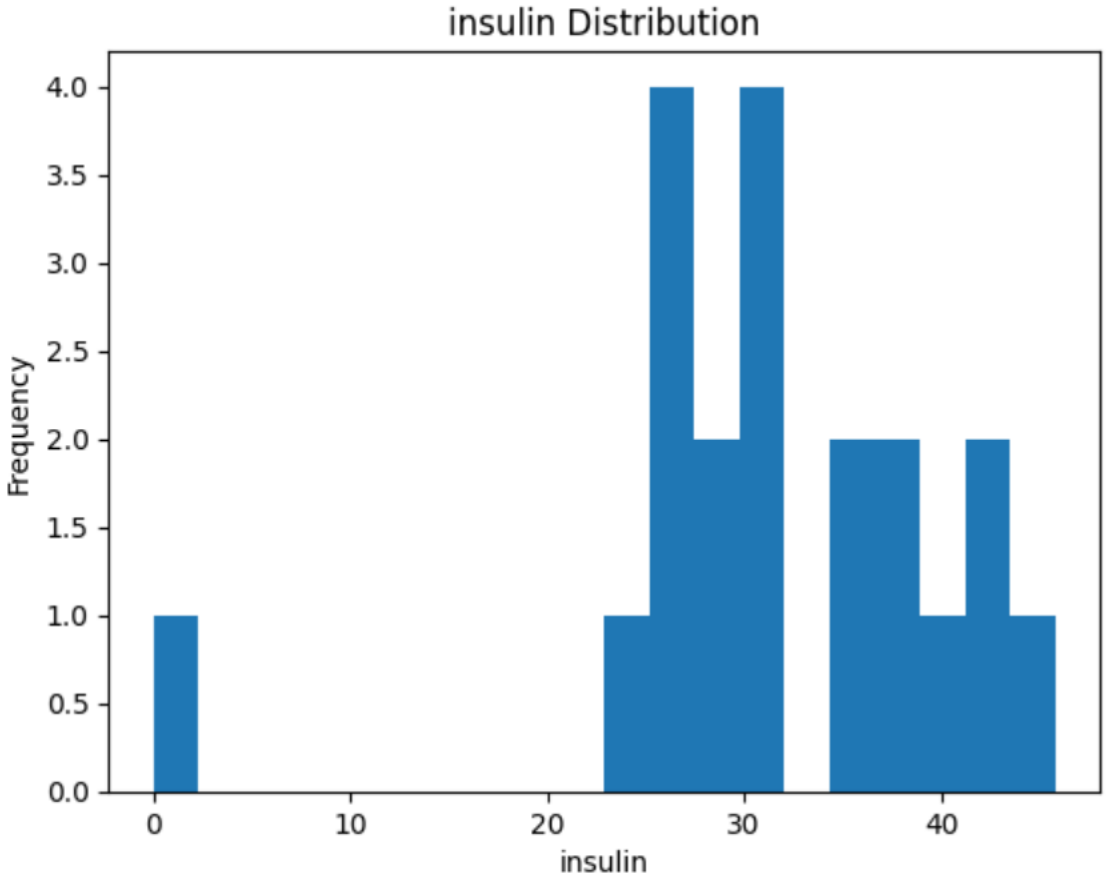
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Glucose', y='insulin', data=df, hue='outcome')
plt.title('Glucose vs. insulin')
plt.xlabel('Glucose')
plt.ylabel('insulin')
plt.legend(title='outcome')
plt.show()
```

Output:



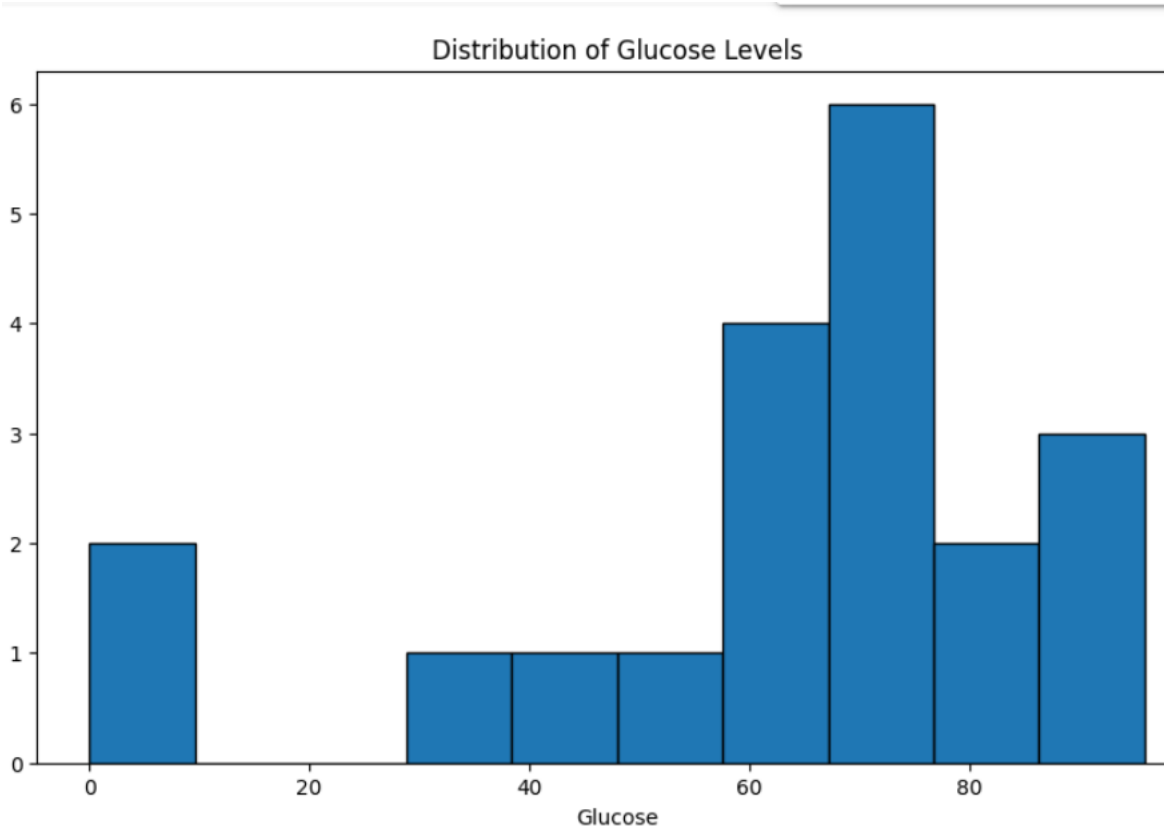

```
# Histogram
plt.hist(df['insulin'], bins=20)
plt.title('insulin Distribution')
plt.xlabel('insulin')
plt.ylabel('Frequency')
plt.show()
```

Output:




```
plt.figure(figsize=(10, 6))
plt.hist(df['Glucose'], bins=10, edgecolor='black')
plt.title('Distribution of Glucose Levels')
plt.xlabel('Glucose')
plt.ylabel('Count')
plt.show()
```

Output:



DATASET:

A	B	C	D	E	F	G	H	I
frequency	Pregnanci	Glucose	blood pre	Skin Thick	insulin	BMI	Age	outcome
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0