In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
data = pd.read_csv('diabetes.csv')
data.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	O
0	6	148	72	35	0	33.6	0.627	50	
1	1	85	66	29	0	26.6	0.351	31	
2	8	183	64	0	0	23.3	0.672	32	
3	1	89	66	23	94	28.1	0.167	21	
4	0	137	40	35	168	43.1	2.288	33	

Class Distribution: (class value 1 is interpreted as "tested positive for diabetes")

In [3]:
```
data.shape
```

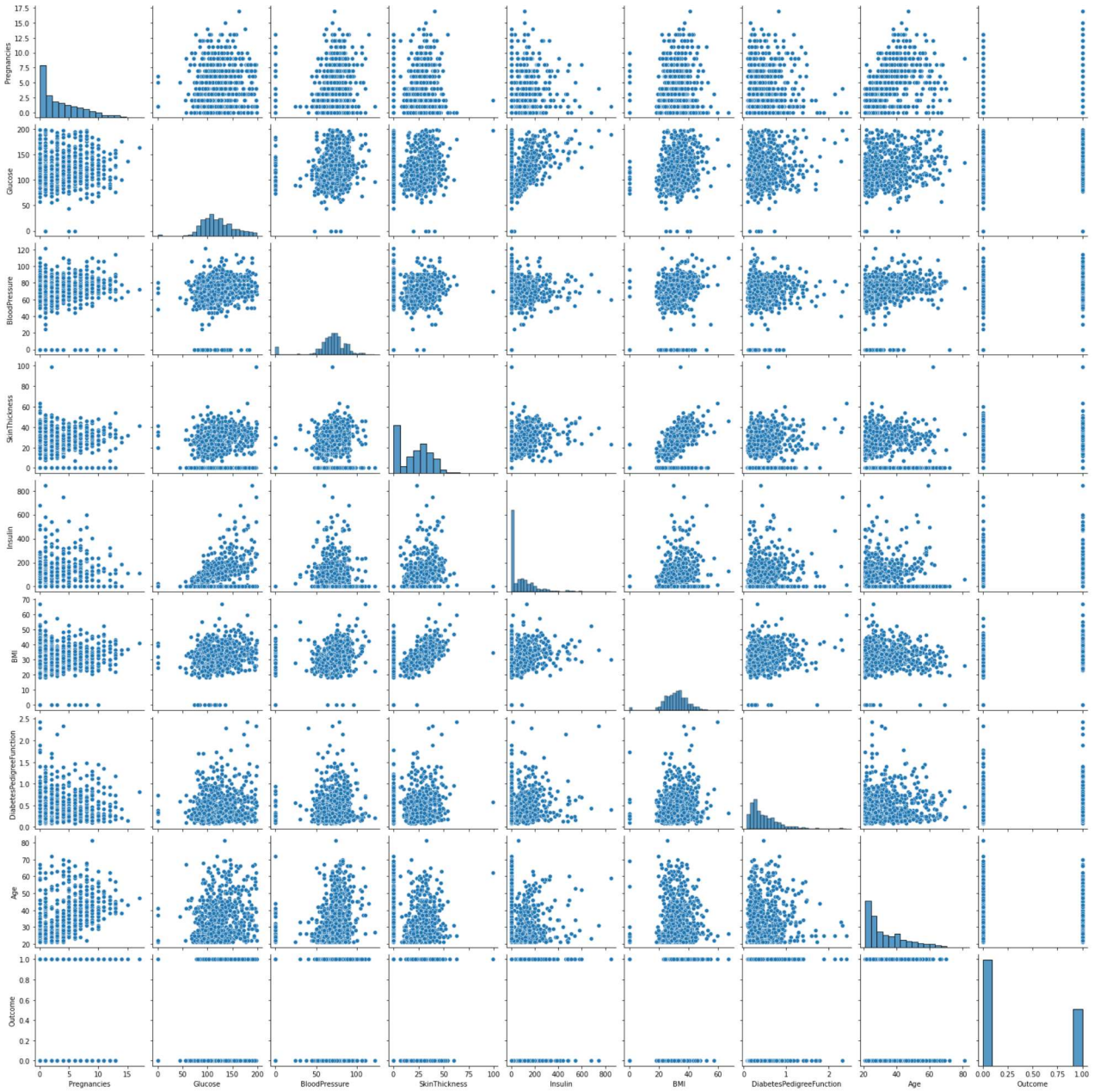
Out[3]:
```
(768, 9)
```

In [4]:
```
data.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
```

In [5]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.pairplotplot(data)
```

Out[5]:
```
<seaborn.axisgrid.PairGrid at 0x1c67840f400>
```

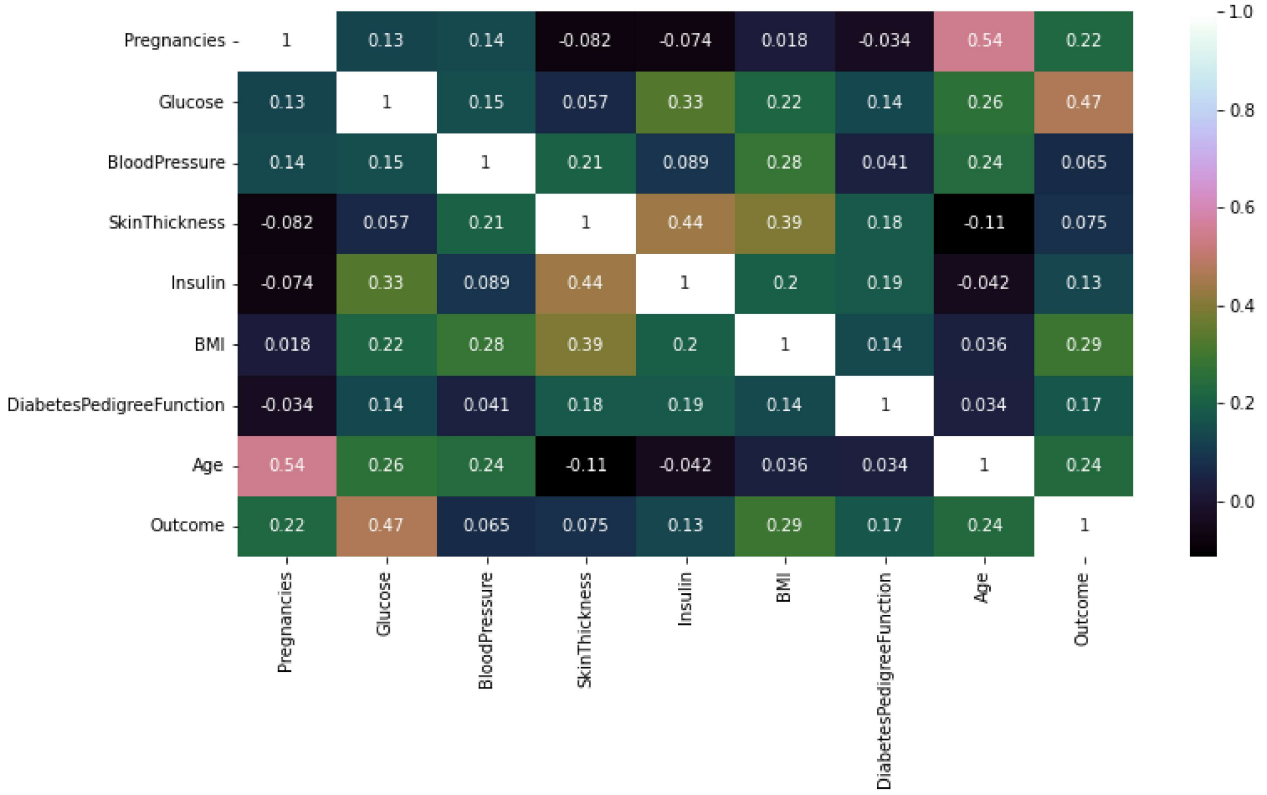


In [7]:

```
plt.figure(figsize=(12,6))
sns.heatmap(data.corr(),annot=True,cmap='cubehelix')
```

Out[7]:

```
<AxesSubplot:>
```



In [8]:
```
type(data.corr())
```

Out[8]:
```
pandas.core.frame.DataFrame
```

In [9]:
```
data.corr()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Dia
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	

In [15]:
```
data.loc[:, data.corr()['Outcome'] > 0.4]
```

Out[15]:

	Glucose	Outcome
0	148	1

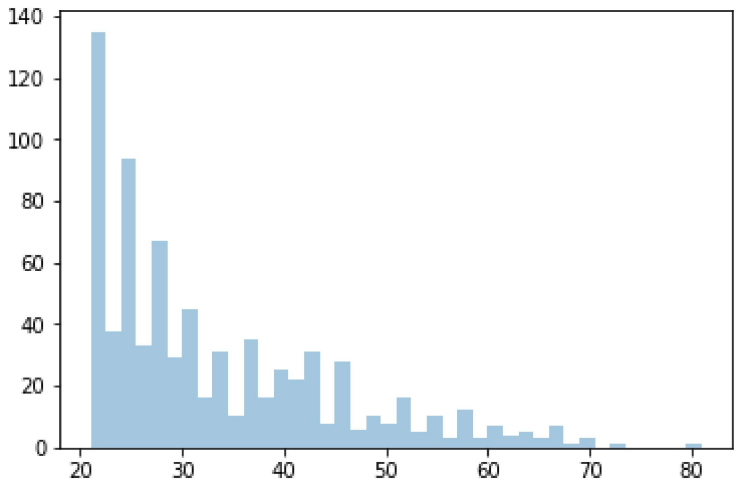
	Glucose	Outcome
1	85	0
2	183	1
3	89	0
4	137	1
...	...	...
763	101	0
764	122	0
765	121	0
766	126	1
767	93	0

768 rows × 2 columns

In [18]:
```
sns.distplot(x=data['Age'],kde=False,bins=40)
```

```
E:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` i
s a deprecated function and will be removed in a future version. Please adapt your code
to use either `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

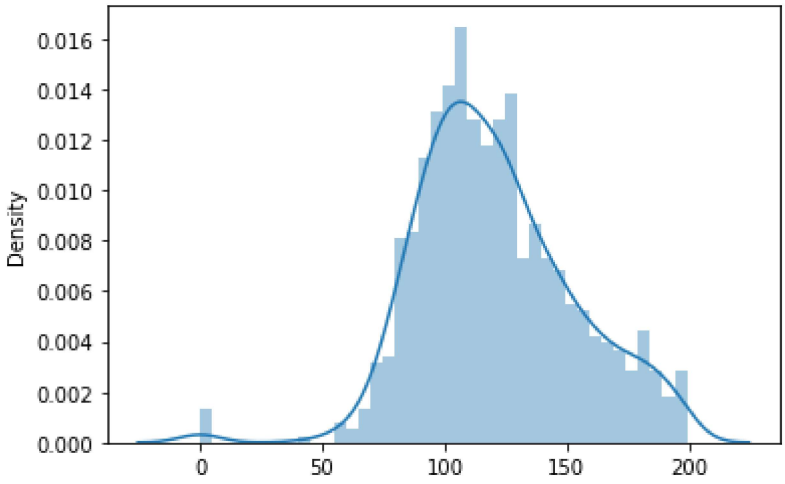
Out[18]:
```
<AxesSubplot:>
```



In [21]:
```
sns.distplot(x=data['Glucose'],bins=40)
```

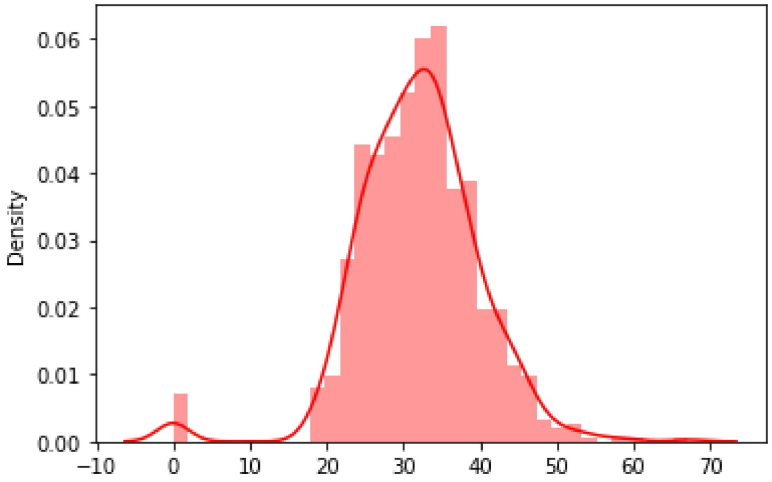
```
E:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` i
s a deprecated function and will be removed in a future version. Please adapt your code
to use either `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[21]:
```
<AxesSubplot:ylabel='Density'>
```



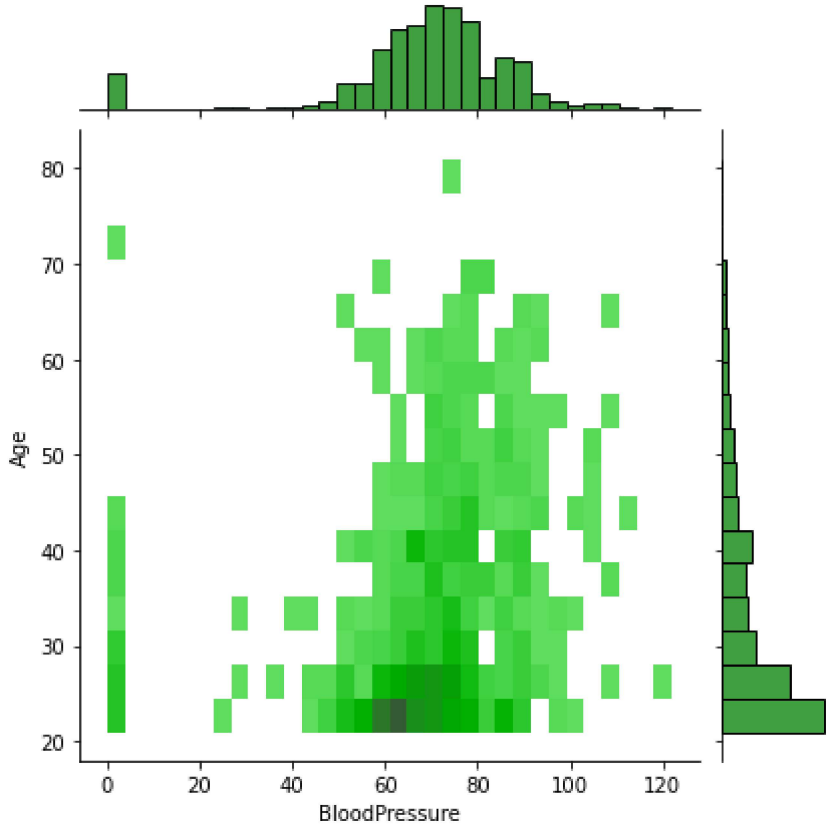
In [20]:
```
sns.distplot(x=data['BMI'],color='red')
```

Out[20]:
```
<AxesSubplot:ylabel='Density'>
```



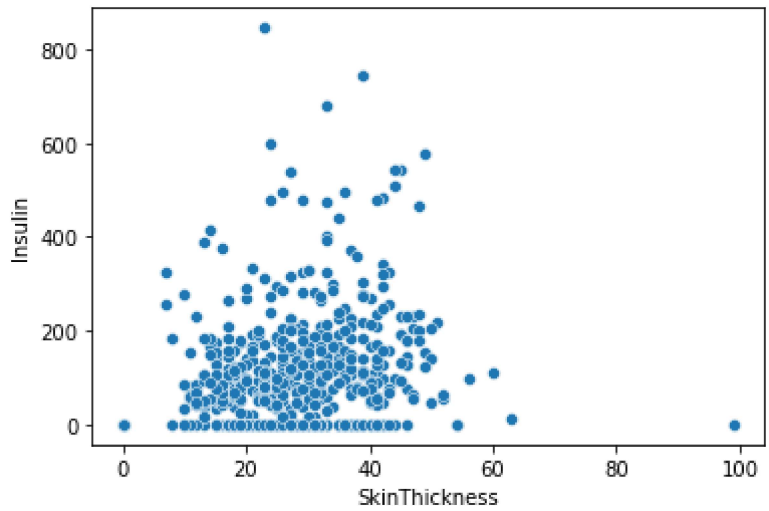
In [22]:
```
sns.jointplot(x=data['BloodPressure'],y=data['Age'],kind='hist',color='green')
```

Out[22]:
```
<seaborn.axisgrid.JointGrid at 0x1c602a4d2b0>
```



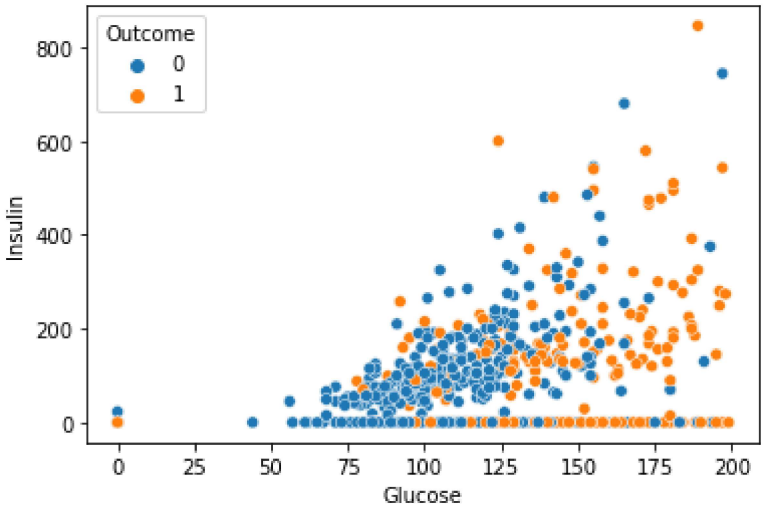
In [23]:
```
sns.scatterplot(x=data['SkinThickness'],y=data['Insulin'])
```

Out[23]:
```
<AxesSubplot:xlabel='SkinThickness', ylabel='Insulin'>
```



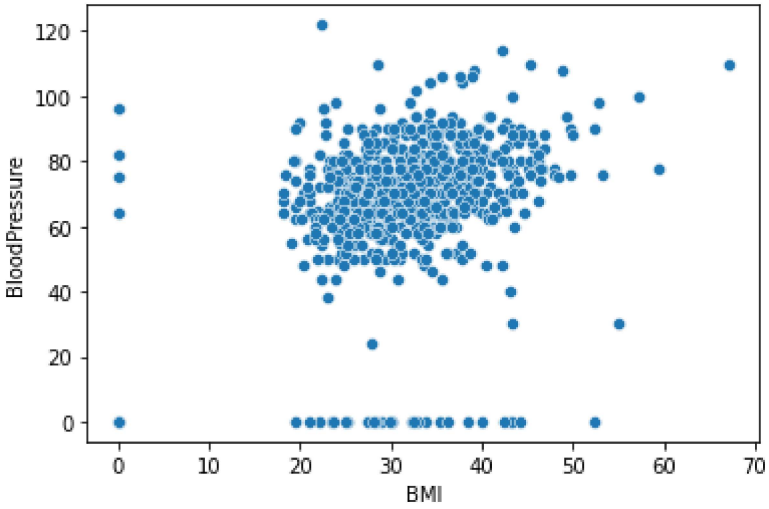
In [24]:
```
sns.scatterplot(x=data['Glucose'],y=data['Insulin'],hue=data['Outcome'])
```

Out[24]:
```
<AxesSubplot:xlabel='Glucose', ylabel='Insulin'>
```



In [25]:
```
sns.scatterplot(x=data['BMI'],y=data['BloodPressure'])
```

Out[25]:
```
<AxesSubplot:xlabel='BMI', ylabel='BloodPressure'>
```



In [27]:
```
X = data.iloc[:,:-1].values
y = data.iloc[:,-1].values
```

In [29]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0
```

In [30]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [31]:
```
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.naive_bayes import GaussianNB
```

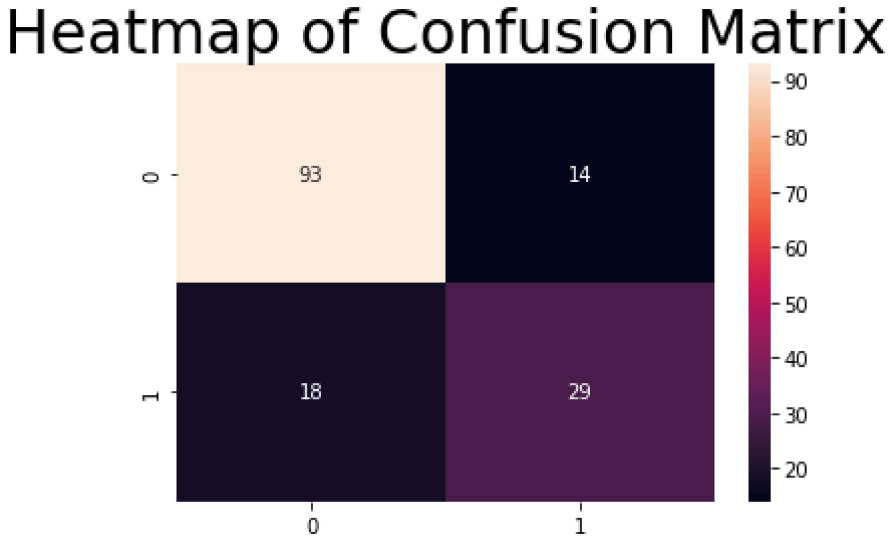
In [32]:
```
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Out[32]:
```
GaussianNB()
```

In [33]:
```
y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
print(accuracy_score(y_test, y_pred))
```

```
[[93 14]
 [18 29]]
0.7922077922077922
```

In [34]:
```
plt.title('Heatmap of Confusion Matrix', fontsize = 30)
sns.heatmap(cm, annot = True)
plt.show()
```



In [ ]: