In [1]: In [3]: Out[3]:	<pre>import pandas as pd import matplotlib.pyplot as plt import seaborn as sns  data = pd.read_csv('Heart Disease.csv') data.head(10)  age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target  0 52 1 0 125 212 0 1 168 0 1.0 2 2 3 0</pre>
	1       53       1       0       140       203       1       0       155       1       3.1       0       0       3       0         2       70       1       0       145       174       0       1       125       1       2.6       0       0       3       0         3       61       1       0       148       203       0       1       161       0       0.0       2       1       3       0         4       62       0       0       138       294       1       1       106       0       1.9       1       3       2       0         5       58       0       0       100       248       0       0       122       0       1.0       1       0       2       1         6       58       1       0       160       289       0       0       145       1       0.8       1       1       3       0         8       46       1       0       120       249       0       0       144       0       0.8       2       0       3       0
(fasting bloo definite left v	9 54 1 0 122 286 0 0 116 1 3.2 1 2 2 0  st pain type (4 values) value 0:typeical angin Value 1:atypical angin value 2:non-anginal pain value 3:asymptomatic trestdps: resting blood pressure(in mm hg on admission to the hospital) chol:serum cholestoral in mg/dl fbs: d sugar>120 mg/dl)(1 = true;0 = false) restecg:resting eletrocardiographic results value 0:normal value 1:having ST-T wave abnormality(T wave inversions and/or ST elevation or depression of >0.05mV) value 2:showing probable or rentricular hypertrophy by Estes criteria thalach:maximum heart rate achieved exang:exercise induced angina(1=yes;0 = no) oldpeak = ST depression induced by exercise ST segment value 1:upsloping value 2:flat value ng ca:number of major vessel(0-3)colored by flourosopy thal: 3 = normal;6 = fixed defect;7 = reversable defect target: 0:less chance of heart attack, 1=mare chance of heart attack
In [4]: Out[4]:	data tail()    Age   Sex   Cp   Trestbps   Chol   Fbs   Festecy   Chalach   Evang   Oldpeak   Slope   Ca   Chalach   Evang   Chol   Cho
In [5]:	Find shape of our Dataset(Number of Rows and Columns)  print("Number of Rows", data.shape[0]) print("Number of Colommns", data.shape[1])  Number of Rows 1025 Number of Colommns 14  Get information about our Dataset like Toatal Number Row, Total Number of colomns, Datatype of Each
In [6]:	data.info()
In [8]: Out[8]:	13 target 1025 non-null int64 dtypes: float64(1), int64(13) memory usage: 112.2 kB  Check Null values in the Dataset  data.isnull().sum()  age 0 sex 0 cp 0 trestbps 0 chol 0 fbs 0 restecg 0 thalach 0 exang 0
In [9]:	oldpeak 0 slope 0 ca 0 thal 0 target 0 dtype: int64   Check for Duplicate Date and drop them   data_dup = data.duplicated().any() print(data_dup)
In [10]: In [12]: Out[12]:	<pre>True  data = data.drop_duplicates()  data.shape (302, 14)</pre>
In [13]: Out[13]:	Count   302.00000   302.0000
In [17]:	Draw correlation Matrix  plt.figure(figsize=(17,7)) sns.heatmap(data.corr(),annot=True)
Out[17]:	Age - 1
In [18]: Out[18]:	How many people have Heart disease, and how many don't have heart disesse in this dataset?    data.columns   Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 't
In [19]: Out[19]: In [26]:	<pre>'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'], dtype='object')  data['target'].value_counts()  1    164 0    138 Name: target, dtype: int64  sns.countplot(data['target'])</pre>
Out[26]:	C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional a rgument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.  **AxesSubplot:Xlabel='target', ylabel='count'>  160 140 120 100 100 100 100 100 100 100 100 10
	Find count of male & female in this Dataset
In [27]: Out[27]: In [28]: Out[28]:	<pre>data.columns Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',</pre>
In [31]:	Name: sex, dtype: int64  sns.countplot(data['sex']) plt.xticks([0,1],['female','male']) plt.show()  C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional a rgument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.  200- 175- 150-
	Find genber distribution according to the Target varible
<pre>In [32]: Out[32]: In [36]:</pre>	<pre>data.columns Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',</pre>
	Chook Ago distribution in the Dataset
In [40]:	<pre>check Age distribution in the Dataset  sns.distplot(data['age']) plt.show()  c:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt y our code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).     warnings.warn(msg, FutureWarning)  ous</pre>
	0.04 - 0.03 - 0.02 - 0.01 - 0.00 - 20 30 40 50 60 70 80
chest pain ty	Check chest Pain type <pre> pe(4 values) value 0:typical angina value 1:atypical angina value 2:non-anginal pain value 3:non-anginal pain  sns.countplot(data['cp']) plt.xticks([0,1,2,3],["typical angina","atypical angina","non-anginal pain","anginal pain"]) plt.xticks(rotation=75) plt.show()  C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional a rgument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  warnings.warn(</pre>
	show The cheet pair distribution as per target veriable
In [45]: Out[45]: In [47]:	show The chest pain distribution as per target variable  data.columns  Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'], dtype='object')  sns.countplot(x="cp", hue="target", data=data)
[].	plt.legend(labels=["no-disease", "disease"]) plt.show()  no-disease disease  quad and the series of
In [48]:	Show fasting blood sugar distribution according to target variable  sns.countplot(x="cp", hue="target", data=data) plt.legend(labels=["no-disease", "disease"]) plt.show()
	no-disease disease disease p p p p p p p p p p p p p p p p p p p
In [50]: Out[50]:	Check Resting blood pressure distribution  data.columns  Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'], dtype='object')
In [51]: Out[51]:	<pre>data['trestbps'].hist() </pre> <pre> 70 60</pre>
	show distribution of serum cholesterol
<pre>In [54]: Out[54]: In [55]: Out[55]:</pre>	<pre>data.columns Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',</pre>
Out[55]:	AxesSubplot:>
<pre>In [57]: Out[57]: In [60]:</pre>	<pre>plot continuous variables  data.columns  Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',</pre>
<pre>In [61]: Out[61]: In [62]: Out[62]: In [67]:</pre>	<pre>cate_val.append(column) else:     cont_val.append(column)  cate_val ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']  cont_val ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']  data.hist(cont_val,figsize=(15,6))</pre>
	age trestbps  60 40 20 40 40 40 40 40 40 40 40 40 40 40 40 40
	30 40 50 chol 60 70  100 120 14thalach 160 180 200  75  50  200 300 oldpeak 400 500  80 100 120 140 160 180 200
In [ ]:	