

# # Heart Disease Prediction In Healthcare System.

Importing Packages

In [1]:

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

Importing Dataset

In [2]:

```
data=pd.read_csv("Raw dataset of heart disease.csv")
```

In [3]:

```
data.head()
```

Out[3]:

|   | age | sex    | chest pain type | resting blood pressure | serum cholestoral | fasting blood sugar | electrocardiographic results | resting maximum heart rate achieved | exercise induced angina |
|---|-----|--------|-----------------|------------------------|-------------------|---------------------|------------------------------|-------------------------------------|-------------------------|
| 0 | 52  | Male   | typical angina  | 125                    | 212               | False               | having ST-T                  | 168                                 | No                      |
| 1 | 53  | Male   | typical angina  | 140                    | 203               | True                | normal                       | 155                                 | Yes                     |
| 2 | 70  | Male   | typical angina  | 145                    | 174               | False               | having ST-T                  | 125                                 | Yes                     |
| 3 | 61  | Male   | typical angina  | 148                    | 203               | False               | having ST-T                  | 161                                 | No                      |
| 4 | 62  | Female | typical angina  | 138                    | 294               | True                | having ST-T                  | 106                                 | No                      |



In [4]:

```
data.describe()
```

Out[4]:

|              | age         | resting blood pressure | serum cholestoral | maximum heart rate achieved | oldpeak     | colored by flourosopy |
|--------------|-------------|------------------------|-------------------|-----------------------------|-------------|-----------------------|
| <b>count</b> | 1025.000000 | 1025.000000            | 1025.000000       | 1025.000000                 | 1025.000000 | 1025.000000           |
| <b>mean</b>  | 54.434146   | 131.611707             | 246.000000        | 149.114146                  | 1.071512    | 0.754146              |
| <b>std</b>   | 9.072290    | 17.516718              | 51.59251          | 23.005724                   | 1.175053    | 1.030798              |
| <b>min</b>   | 29.000000   | 94.000000              | 126.00000         | 71.000000                   | 0.000000    | 0.000000              |
| <b>25%</b>   | 48.000000   | 120.000000             | 211.00000         | 132.000000                  | 0.000000    | 0.000000              |
| <b>50%</b>   | 56.000000   | 130.000000             | 240.00000         | 152.000000                  | 0.800000    | 0.000000              |
| <b>75%</b>   | 61.000000   | 140.000000             | 275.00000         | 166.000000                  | 1.800000    | 1.000000              |
| <b>max</b>   | 77.000000   | 200.000000             | 564.00000         | 202.000000                  | 6.200000    | 4.000000              |

In [5]:

```
data.shape
```

Out[5]:

```
(1025, 14)
```

In [6]:

```
data.columns
```

Out[6]:

```
Index(['age', 'sex', 'chest pain type', 'resting blood pressure',
       'serum cholestoral', 'fasting blood sugar',
       'resting electrocardiographic results ', 'maximum heart rate achieve
d',
       'exercise induced angina', 'oldpeak', ' slope of the peak',
       'colored by flourosopy', 'thal', 'target'],
      dtype='object')
```

In [7]:

```
data.nunique()
```

Out[7]:

```
age                41
sex                2
chest pain type    4
resting blood pressure    49
serum cholestoral    152
fasting blood sugar    2
resting electrocardiographic results    3
maximum heart rate achieved    91
exercise induced angina    2
oldpeak            40
  slope of the peak    3
colored by flourosopy    5
thal                3
target             2
dtype: int64
```

In [8]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   age                                  1025 non-null   int64
1   sex                                  1025 non-null   object
2   chest pain type                      1025 non-null   object
3   resting blood pressure               1025 non-null   int64
4   serum cholestoral                   1025 non-null   int64
5   fasting blood sugar                 1025 non-null   bool
6   resting electrocardiographic results 1025 non-null   object
7   maximum heart rate achieved          1025 non-null   int64
8   exercise induced angina             1025 non-null   object
9   oldpeak                             1025 non-null   float64
10  slope of the peak                    1025 non-null   object
11  colored by flourosopy                1025 non-null   int64
12  thal                                 1018 non-null   object
13  target                               1025 non-null   object
dtypes: bool(1), float64(1), int64(5), object(7)
memory usage: 77.1+ KB
```

Data Pre-processing

In [9]:

```
data.isnull().sum()
```

Out[9]:

```
age                0
sex                0
chest pain type    0
resting blood pressure    0
serum cholestoral    0
fasting blood sugar    0
resting electrocardiographic results    0
maximum heart rate achieved    0
exercise induced angina    0
oldpeak           0
  slope of the peak    0
colored by flourosopy    0
thal              7
target           0
dtype: int64
```

In [ ]:

In [10]:

```
data['sex'].unique()
```

Out[10]:

```
array(['Male', 'Female'], dtype=object)
```

In [11]:

```
data['sex']=data['sex'].map({"Male":1,
                             "Female":0})
```

In [ ]:

In [12]:

```
data['chest pain type'].unique()
```

Out[12]:

```
array(['typical angina', 'atypical angina', 'non-anginal pain',
      'asymptomatic'], dtype=object)
```

In [13]:

```
data['chest pain type']=data['chest pain type'].map({"typical angina":0,"atypical angina":1,
                                                    "asymptomatic":3})
```

In [ ]:

In [14]:

```
data['fasting blood sugar'].unique()
```

Out[14]:

```
array([False,  True])
```

In [15]:

```
data['fasting blood sugar']=data['fasting blood sugar'].map({False:0,True:1})
```

In [ ]:

In [16]:

```
data['resting electrocardiographic results '].unique()
```

Out[16]:

```
array(['having ST-T', 'normal', 'hypertrophy'], dtype=object)
```

In [17]:

```
data['resting electrocardiographic results ']=data['resting electrocardiographic results ']
```

In [ ]:

In [18]:

```
data['exercise induced angina'].unique()
```

Out[18]:

```
array(['No', 'Yes'], dtype=object)
```

In [19]:

```
data['exercise induced angina']=data['exercise induced angina'].map({"No":0,"Yes":1})
```

In [ ]:

In [20]:

```
data[' slope of the peak'].unique()
```

Out[20]:

```
array(['downsloping', 'upsloping', 'flat'], dtype=object)
```

In [21]:

```
data[' slope of the peak']=data[' slope of the peak'].map({"upsloping":0,"flat":1,"downslop
```

In [ ]:

In [22]:

```
data[' slope of the peak'].unique()
```

Out[22]:

```
array([2, 0, 1], dtype=int64)
```

In [23]:

```
data['thal'].isnull().sum()
```

Out[23]:

```
7
```

In [ ]:

In [ ]:

In [24]:

```
data['thal']=data['thal'].map({"normal":1,"fixed defect":2,"reversable defect":3})
```

In [25]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column                                  Non-Null Count  Dtype
---  -
 0   age                                    1025 non-null   int64
 1   sex                                    1025 non-null   int64
 2   chest pain type                        1025 non-null   int64
 3   resting blood pressure                1025 non-null   int64
 4   serum cholestoral                    1025 non-null   int64
 5   fasting blood sugar                   1025 non-null   int64
 6   resting electrocardiographic results  1025 non-null   int64
 7   maximum heart rate achieved          1025 non-null   int64
 8   exercise induced angina              1025 non-null   int64
 9   oldpeak                              1025 non-null   float64
10   slope of the peak                    1025 non-null   int64
11   colored by flourosopy                 1025 non-null   int64
12   thal                                  1018 non-null   float64
13   target                                1025 non-null   object
dtypes: float64(2), int64(11), object(1)
memory usage: 108.2+ KB
```

In [26]:

```
data['thal'].isnull().sum()
```

Out[26]:

7

In [27]:

```
data['thal']=data['thal'].replace(np.nan,0)
```

In [28]:

```
data['thal'].isnull().sum()
```

Out[28]:

0

In [29]:

```
data['thal']=data['thal'].astype(np.int64)
```

In [30]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column                                  Non-Null Count  Dtype  
---  -
 0   age                                    1025 non-null   int64  
 1   sex                                    1025 non-null   int64  
 2   chest pain type                        1025 non-null   int64  
 3   resting blood pressure                1025 non-null   int64  
 4   serum cholestoral                     1025 non-null   int64  
 5   fasting blood sugar                   1025 non-null   int64  
 6   resting electrocardiographic results  1025 non-null   int64  
 7   maximum heart rate achieved           1025 non-null   int64  
 8   exercise induced angina               1025 non-null   int64  
 9   oldpeak                              1025 non-null   float64 
10   slope of the peak                     1025 non-null   int64  
11   colored by flourosopy                 1025 non-null   int64  
12   thal                                  1025 non-null   int64  
13   target                                1025 non-null   object  
dtypes: float64(1), int64(12), object(1)
memory usage: 108.2+ KB
```

In [ ]:

In [ ]:

In [31]:

```
data['target'].unique()
```

Out[31]:

```
array(['No', 'Yes'], dtype=object)
```

In [32]:

```
data['target']=data['target'].map({"Yes":1,"No":0})
```

In [ ]:



In [33]:

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   age                                   1025 non-null   int64
1   sex                                   1025 non-null   int64
2   chest pain type                       1025 non-null   int64
3   resting blood pressure                1025 non-null   int64
4   serum cholestoral                    1025 non-null   int64
5   fasting blood sugar                   1025 non-null   int64
6   resting electrocardiographic results 1025 non-null   int64
7   maximum heart rate achieved           1025 non-null   int64
8   exercise induced angina              1025 non-null   int64
9   oldpeak                              1025 non-null   float64
10  slope of the peak                    1025 non-null   int64
11  colored by flourosopy                1025 non-null   int64
12  thal                                  1025 non-null   int64
13  target                               1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

In [ ]:

In [34]:

data.columns

Out[34]:

```
Index(['age', 'sex', 'chest pain type', 'resting blood pressure',
      'serum cholestoral', 'fasting blood sugar',
      'resting electrocardiographic results ', 'maximum heart rate achieve
d',
      'exercise induced angina', 'oldpeak', ' slope of the peak',
      'colored by flourosopy', 'thal', 'target'],
      dtype='object')
```

In [ ]:

In [36]:

```
data.rename(columns={'chest pain type':'cp','resting blood pressure':'trestbps','serum chol
'fasting blood sugar':'fbs','resting electrocardiographic results ':'r
'maximum heart rate achieved':'thalach','exercise induced angina':'exan
'colored by flourosopy':'ca'},inplace=True)
```

In [ ]:

Exporting the .csv file which have numerical data after data preprocessing.

In [37]:

```
#data.to_csv("Heart Disease.csv")
```

In [ ]:

Data Analysis

In [ ]:

In [38]:

```
df=pd.read_csv("Heart Disease.csv")
```

In [39]:

```
df.head()
```

Out[39]:

|   | Unnamed: 0 | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca |
|---|------------|-----|-----|----|----------|------|-----|---------|---------|-------|---------|-------|----|
| 0 | 0          | 52  | 1   | 0  | 125      | 212  | 0   | 1       | 168     | 0     | 1.0     | 2     | 2  |
| 1 | 1          | 53  | 1   | 0  | 140      | 203  | 1   | 0       | 155     | 1     | 3.1     | 0     | 0  |
| 2 | 2          | 70  | 1   | 0  | 145      | 174  | 0   | 1       | 125     | 1     | 2.6     | 0     | 0  |
| 3 | 3          | 61  | 1   | 0  | 148      | 203  | 0   | 1       | 161     | 0     | 0.0     | 2     | 1  |
| 4 | 4          | 62  | 0   | 0  | 138      | 294  | 1   | 1       | 106     | 0     | 1.9     | 1     | 3  |

In [40]:

```
df.shape
```

Out[40]:

(1025, 15)

In [41]:

```
df1= df.drop(columns=['Unnamed: 0'])
```

In [42]:

```
df1.shape
```

Out[42]:

(1025, 14)

In [43]:

```
df1.nunique()
```

Out[43]:

```
age          41
sex           2
cp            4
trestbps     49
chol        152
fbs           2
restecg       3
thalach       91
exang         2
oldpeak       40
slope         3
ca            5
thal          4
target        2
dtype: int64
```

In [44]:

```
df1.info()
```

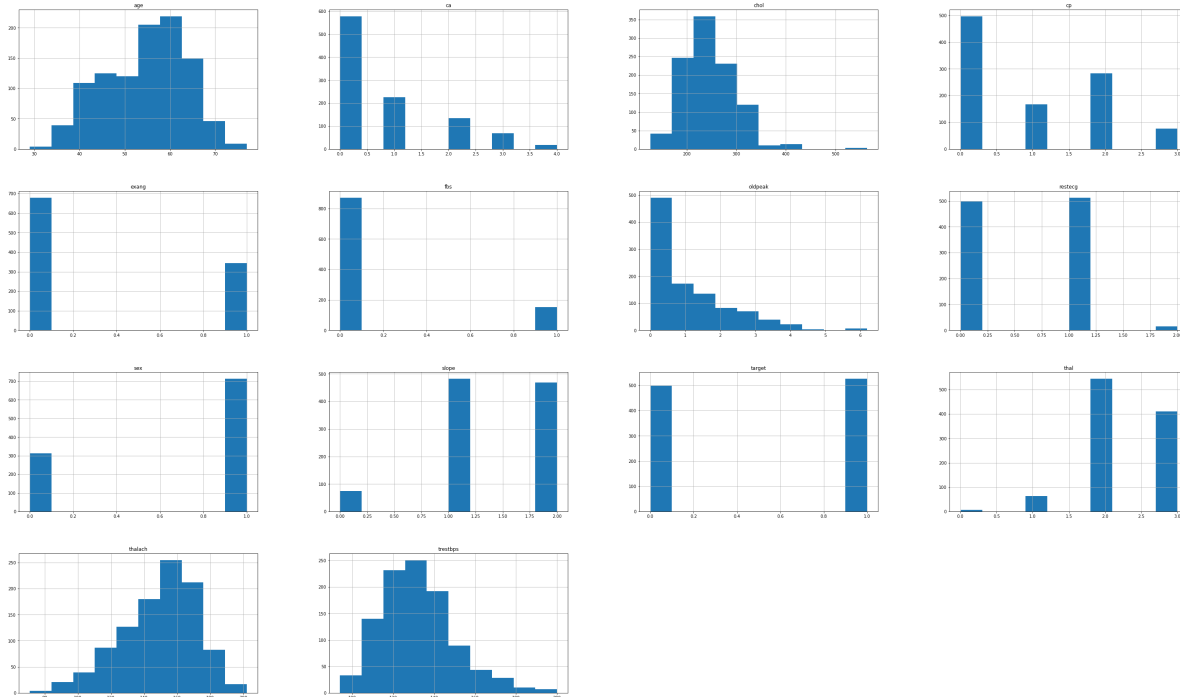
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         1025 non-null  int64
1   sex         1025 non-null  int64
2   cp          1025 non-null  int64
3   trestbps    1025 non-null  int64
4   chol        1025 non-null  int64
5   fbs         1025 non-null  int64
6   restecg     1025 non-null  int64
7   thalach     1025 non-null  int64
8   exang       1025 non-null  int64
9   oldpeak     1025 non-null  float64
10  slope       1025 non-null  int64
11  ca          1025 non-null  int64
12  thal        1025 non-null  int64
13  target      1025 non-null  int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

In [45]:

```
df1.hist(figsize=(50,30))
```

Out[45]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0EE3ECB0>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x0142F4B0>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x016B9CB0>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x016E04D0>],
      [<matplotlib.axes._subplots.AxesSubplot object at 0x016F7CD0>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x0171D4F0>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x01736D30>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x0175A4F0>],
      [<matplotlib.axes._subplots.AxesSubplot object at 0x0175A510>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x01773D90>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x017B7D30>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x017E0550>],
      [<matplotlib.axes._subplots.AxesSubplot object at 0x017FBD50>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x01822570>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x0183CD70>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x01862590>]],
      dtype=object)
```



In [57]:

```
sns.distplot(data['age'],bins=10,color='#3CB371',label="Age")  
plt.legend()  
plt.title("Histogram of Age")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages  
\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated fu  
nction and will be removed in a future version. Please adapt your code to us  
e either `displot` (a figure-level function with similar flexibility) or `hi  
stplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

Out[57]:

Text(0.5, 1.0, 'Histogram of Age')



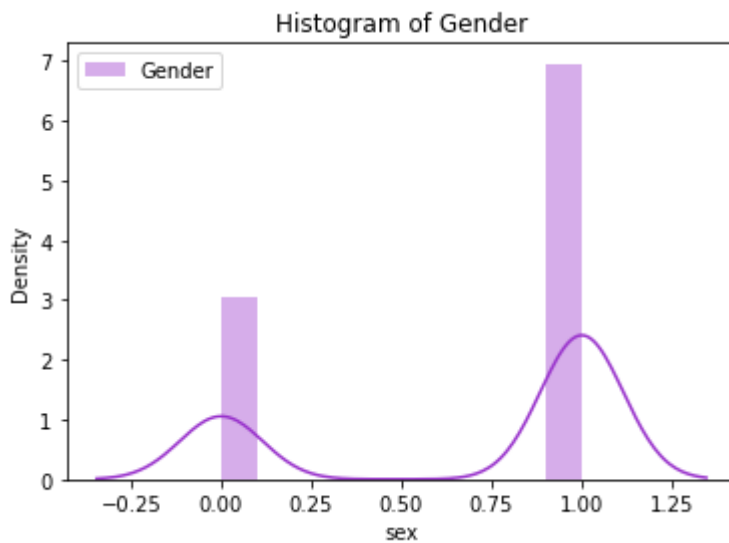
In [76]:

```
sns.distplot(data['sex'],bins=10,color='#9932CC',label="Gender")  
plt.legend()  
plt.title("Histogram of Gender")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages  
\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated fu  
nction and will be removed in a future version. Please adapt your code to us  
e either `displot` (a figure-level function with similar flexibility) or `hi  
stplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

Out[76]:

Text(0.5, 1.0, 'Histogram of Gender')



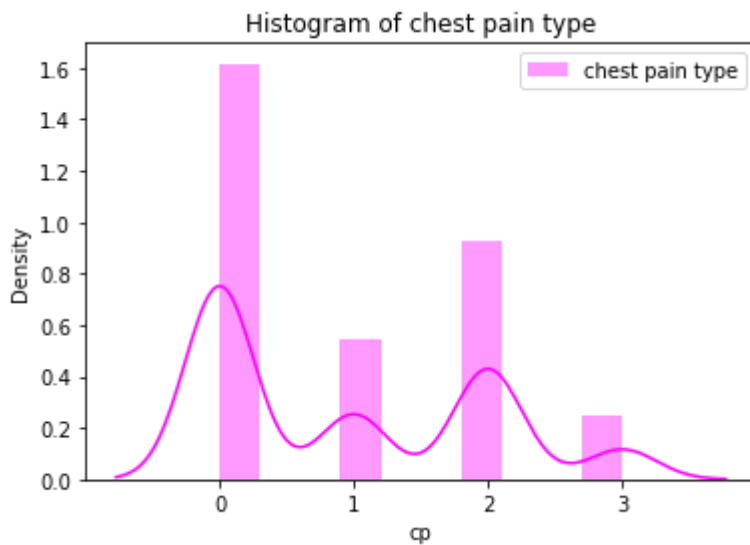
In [75]:

```
sns.distplot(data['cp'],bins=10,color='#FF00FF',label="chest pain type")
plt.legend()
plt.title("Histogram of chest pain type")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is 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[75]:

Text(0.5, 1.0, 'Histogram of chest pain type')



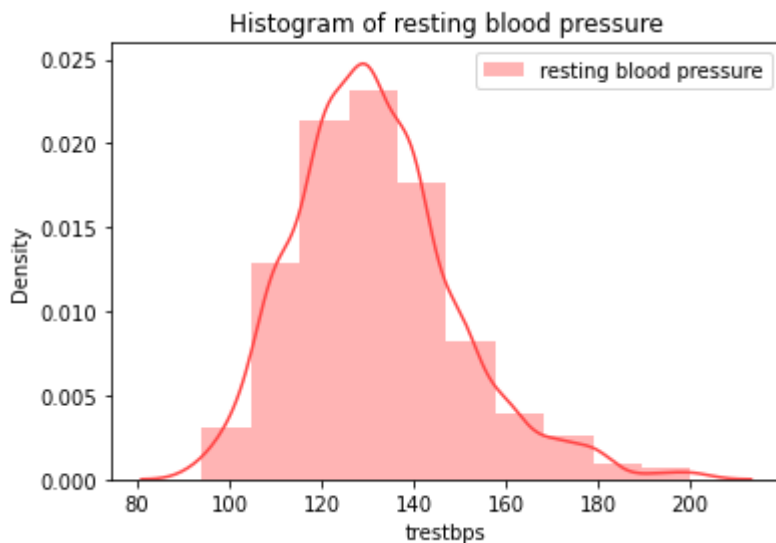
In [77]:

```
sns.distplot(data['trestbps'],bins=10,color='#FF4444',label="resting blood pressure")
plt.legend()
plt.title("Histogram of resting blood pressure")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is 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[77]:

Text(0.5, 1.0, 'Histogram of resting blood pressure')





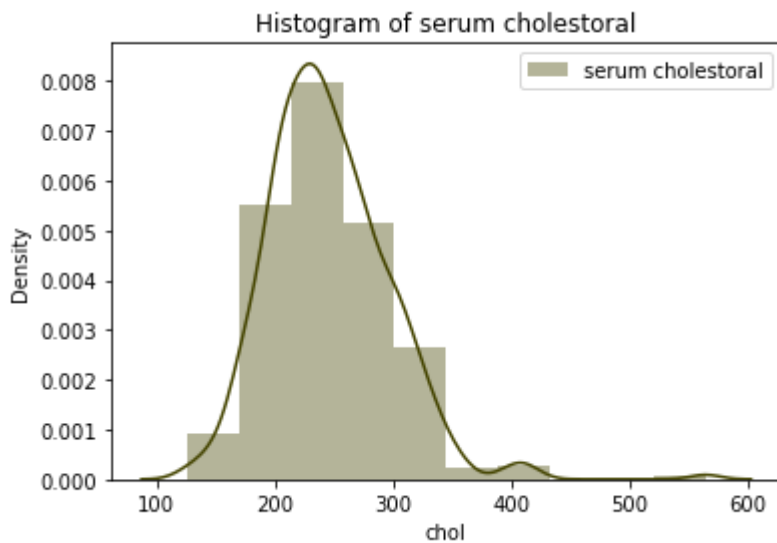
In [79]:

```
sns.distplot(data['chol'],bins=10,color='#444400',label="serum cholestoral")  
plt.legend()  
plt.title("Histogram of serum cholestoral")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages  
\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated fu  
nction and will be removed in a future version. Please adapt your code to us  
e either `displot` (a figure-level function with similar flexibility) or `hi  
stplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

Out[79]:

Text(0.5, 1.0, 'Histogram of serum cholestoral')

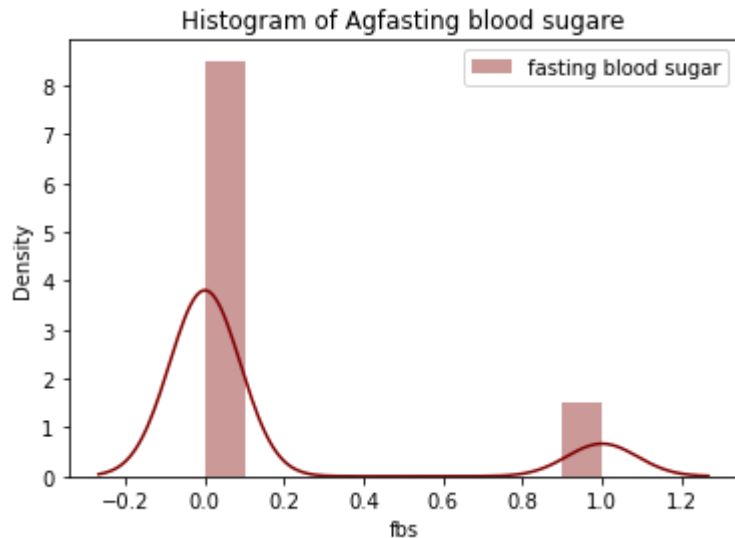


In [82]:

```
sns.distplot(data['fbs'],bins=10,color='#800000',label="fasting blood sugar")
plt.legend()
plt.title("Histogram of Agfasting blood sugare")
```

Out[82]:

```
Text(0.5, 1.0, 'Histogram of Agfasting blood sugare')
```



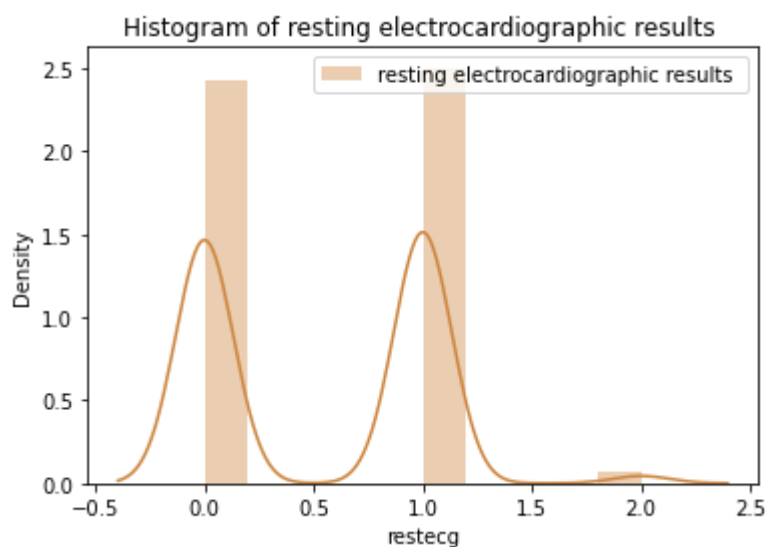
In [83]:

```
sns.distplot(data['restecg'],bins=10,color='#CD853F',label="resting electrocardiographic re
plt.legend()
plt.title("Histogram of resting electrocardiographic results ")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages  
 \seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated fu  
 nction and will be removed in a future version. Please adapt your code to us  
 e either `displot` (a figure-level function with similar flexibility) or `hi  
 stplot` (an axes-level function for histograms).  
 warnings.warn(msg, FutureWarning)

Out[83]:

```
Text(0.5, 1.0, 'Histogram of resting electrocardiographic results ')
```



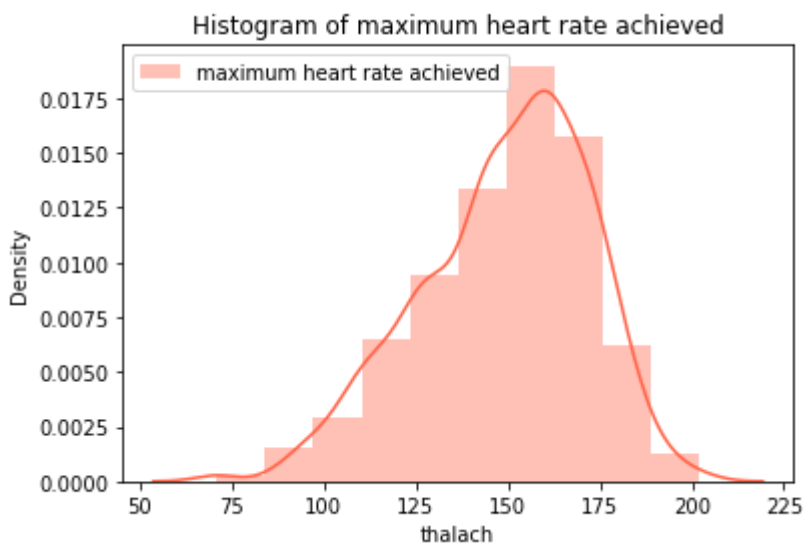
In [85]:

```
sns.distplot(data['thalach'],bins=10,color='#FF6347',label="maximum heart rate achieved")
plt.legend()
plt.title("Histogram of maximum heart rate achieved")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is 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[85]:

Text(0.5, 1.0, 'Histogram of maximum heart rate achieved')



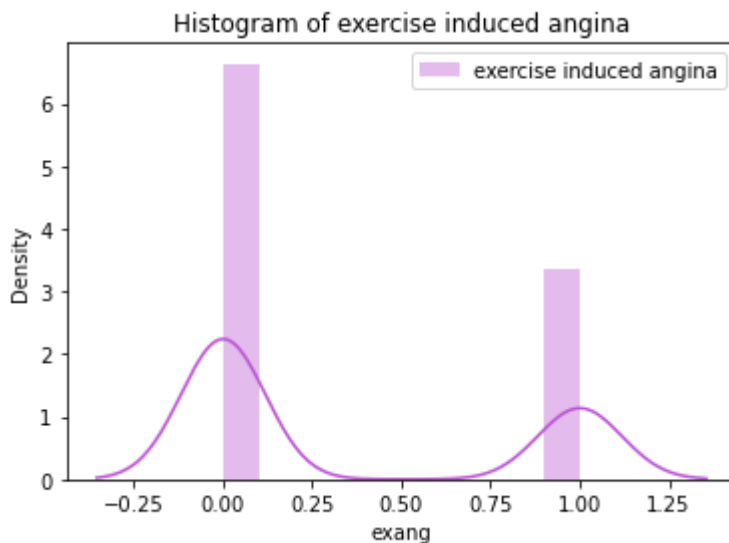
In [86]:

```
sns.distplot(data['exang'],bins=10,color='#BA55D3',label="exercise induced angina")
plt.legend()
plt.title("Histogram of exercise induced angina")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is 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[86]:

Text(0.5, 1.0, 'Histogram of exercise induced angina')



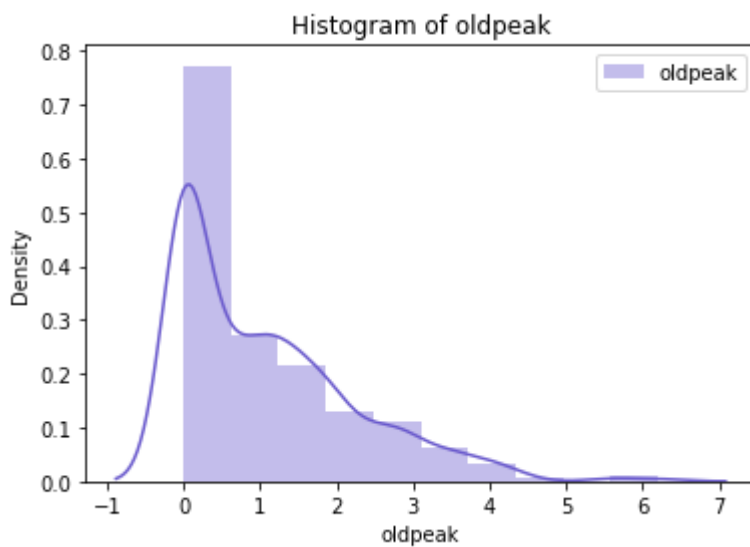
In [87]:

```
sns.distplot(data['oldpeak'],bins=10,color='#6A5ACD',label="oldpeak")  
plt.legend()  
plt.title("Histogram of oldpeak")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages  
\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated fu  
nction and will be removed in a future version. Please adapt your code to us  
e either `displot` (a figure-level function with similar flexibility) or `hi  
stplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

Out[87]:

Text(0.5, 1.0, 'Histogram of oldpeak')



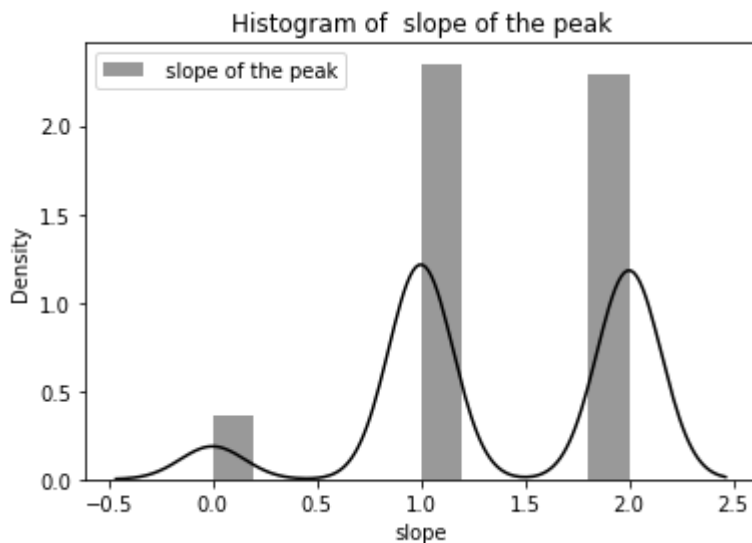
In [88]:

```
sns.distplot(data['slope'],bins=10,color='#000000',label=" slope of the peak")  
plt.legend()  
plt.title("Histogram of  slope of the peak")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages  
\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated fu  
nction and will be removed in a future version. Please adapt your code to us  
e either `displot` (a figure-level function with similar flexibility) or `hi  
stplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

Out[88]:

Text(0.5, 1.0, 'Histogram of slope of the peak')



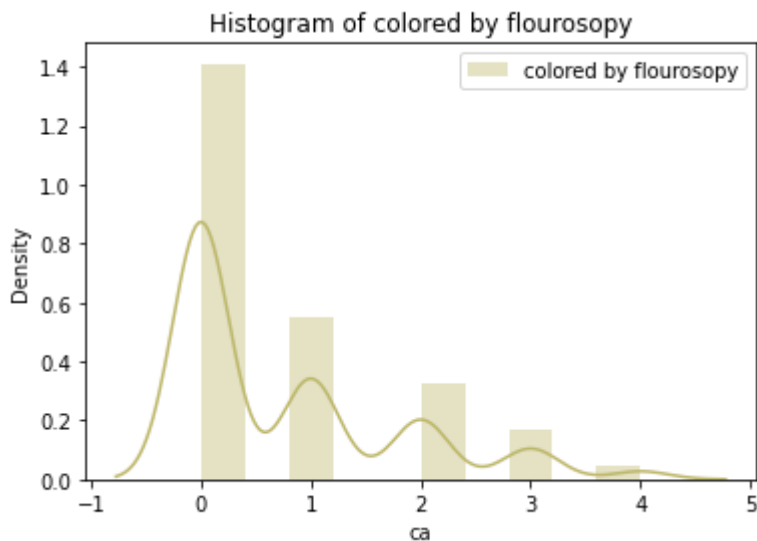
In [89]:

```
sns.distplot(data['ca'],bins=10,color='#BDB76B',label="colored by flourosopy")
plt.legend()
plt.title("Histogram of colored by flourosopy")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is 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[89]:

Text(0.5, 1.0, 'Histogram of colored by flourosopy')



In [72]:

```
sns.distplot(data['thal'],bins=10,color='#3CB371',label="thal")
plt.legend()
plt.title("Histogram of thal")
```

File "<ipython-input-72-1e0d8dbcea16>", line 3  
plt.title("Histogram of thal")

**SyntaxError:** unexpected EOF while parsing

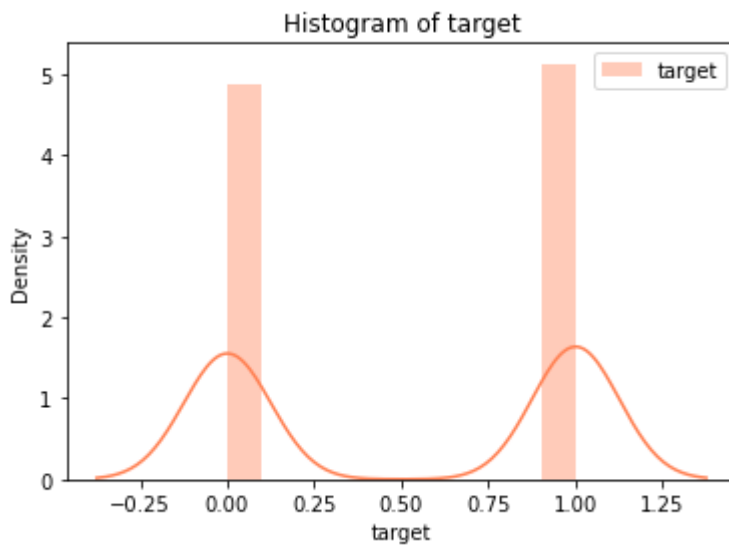
In [90]:

```
sns.distplot(data['target'],bins=10,color='#FF7F50',label="target")  
plt.legend()  
plt.title("Histogram of target")
```

c:\users\lovebro\appdata\local\programs\python\python37-32\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is 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[90]:

Text(0.5, 1.0, 'Histogram of target')



In [ ]:



In [91]:

df1.corr()

Out[91]:

|          | age       | sex       | cp        | trestbps  | chol      | fbs       | restecg   | thalach   |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| age      | 1.000000  | -0.103240 | -0.071966 | 0.271121  | 0.219823  | 0.121243  | -0.132696 | -0.390227 |
| sex      | -0.103240 | 1.000000  | -0.041119 | -0.078974 | -0.198258 | 0.027200  | -0.055117 | -0.049365 |
| cp       | -0.071966 | -0.041119 | 1.000000  | 0.038177  | -0.081641 | 0.079294  | 0.043581  | 0.306839  |
| trestbps | 0.271121  | -0.078974 | 0.038177  | 1.000000  | 0.127977  | 0.181767  | -0.123794 | -0.039264 |
| chol     | 0.219823  | -0.198258 | -0.081641 | 0.127977  | 1.000000  | 0.026917  | -0.147410 | -0.021772 |
| fbs      | 0.121243  | 0.027200  | 0.079294  | 0.181767  | 0.026917  | 1.000000  | -0.104051 | -0.008866 |
| restecg  | -0.132696 | -0.055117 | 0.043581  | -0.123794 | -0.147410 | -0.104051 | 1.000000  | 0.048411  |
| thalach  | -0.390227 | -0.049365 | 0.306839  | -0.039264 | -0.021772 | -0.008866 | 0.048411  | 1.000000  |
| exang    | 0.088163  | 0.139157  | -0.401513 | 0.061197  | 0.067382  | 0.049261  | -0.065606 | -0.380281 |
| oldpeak  | 0.208137  | 0.084687  | -0.174733 | 0.187434  | 0.064880  | 0.010859  | -0.050114 | -0.349796 |
| slope    | -0.169105 | -0.026666 | 0.131633  | -0.120445 | -0.014248 | -0.061902 | 0.086086  | 0.395308  |
| ca       | 0.271551  | 0.111729  | -0.176206 | 0.104554  | 0.074259  | 0.137156  | -0.078072 | -0.207888 |
| thal     | 0.072297  | 0.198424  | -0.163341 | 0.059276  | 0.100244  | -0.042177 | -0.020504 | -0.098068 |
| target   | -0.229324 | -0.279501 | 0.434854  | -0.138772 | -0.099966 | -0.041164 | 0.134468  | 0.422895  |

In [92]:

# Heatmap

In [94]:

```

cormat = df1.corr()
top_cor_features = cormat.index
plt.figure(figsize=(20,20))
# Plot Heat Map
g = sns.heatmap(df1[top_cor_features].corr(), annot=True, cmap="coolwarm")

```

