

Heart diseases analysis

May 14, 2024

```
[2]: ##import the all library
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[3]: ## import the data set
data = pd.read_csv("C:/Users/Vikas/Desktop/heart.csv")
```

```
[4]: data.head(6)
```

```
[4]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	\
0	52	1	0	125	212	0	1	168	0	1.0	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	
4	62	0	0	138	294	1	1	106	0	1.9	1	
5	58	0	0	100	248	0	0	122	0	1.0	1	

	ca	thal	target
0	2	3	0
1	0	3	0
2	0	3	0
3	1	3	0
4	3	2	0
5	0	2	1

```
[5]: data.columns
```

```
[5]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
        dtype='object')
```

```
[6]: data.shape
```

```
[6]: (1025, 14)
```

```
[7]: data_dup = data.drop_duplicates().any()
      print(data_dup)
```

```
age      True
sex      True
cp       True
trestbps True
chol     True
fbs      True
restecg  True
thalach  True
exang    True
oldpeak  True
slope    True
ca       True
thal     True
target   True
dtype: bool
```

```
[8]: data.describe()
```

```
[8]:
```

	age	sex	cp	trestbps	chol \
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.000000
std	9.072290	0.460373	1.029641	17.516718	51.59251
min	29.000000	0.000000	0.000000	94.000000	126.000000
25%	48.000000	0.000000	0.000000	120.000000	211.000000
50%	56.000000	1.000000	1.000000	130.000000	240.000000
75%	61.000000	1.000000	2.000000	140.000000	275.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000

	fbs	restecg	thalach	exang	oldpeak \
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	0.149268	0.529756	149.114146	0.336585	1.071512
std	0.356527	0.527878	23.005724	0.472772	1.175053
min	0.000000	0.000000	71.000000	0.000000	0.000000
25%	0.000000	0.000000	132.000000	0.000000	0.000000
50%	0.000000	1.000000	152.000000	0.000000	0.800000
75%	0.000000	1.000000	166.000000	1.000000	1.800000
max	1.000000	2.000000	202.000000	1.000000	6.200000

	slope	ca	thal	target
count	1025.000000	1025.000000	1025.000000	1025.000000
mean	1.385366	0.754146	2.323902	0.513171
std	0.617755	1.030798	0.620660	0.500070
min	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	2.000000	0.000000
50%	1.000000	0.000000	2.000000	1.000000

75%	2.000000	1.000000	3.000000	1.000000
max	2.000000	4.000000	3.000000	1.000000

```
[9]: data.shape
```

```
[9]: (1025, 14)
```

```
[ ]:
```

```
[10]: 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   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
```

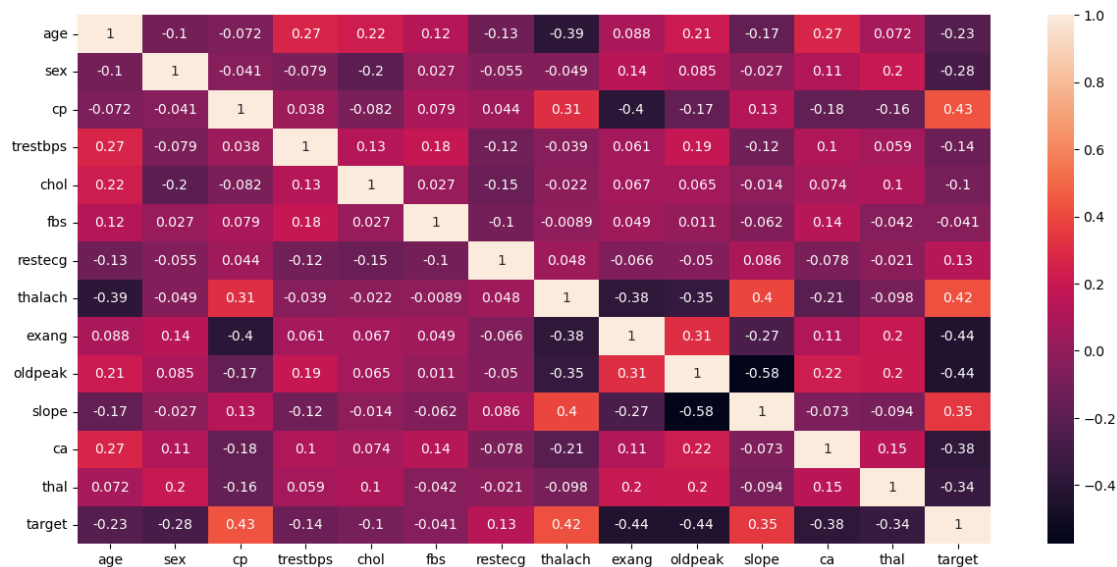
```
[11]: data.isnull().sum()
```

```
[11]: age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
```

```
target      0
dtype: int64
```

```
[12]: plt.figure(figsize=(15,7))
      sns.heatmap(data.corr(),annot=True)
```

```
[12]: <Axes: >
```



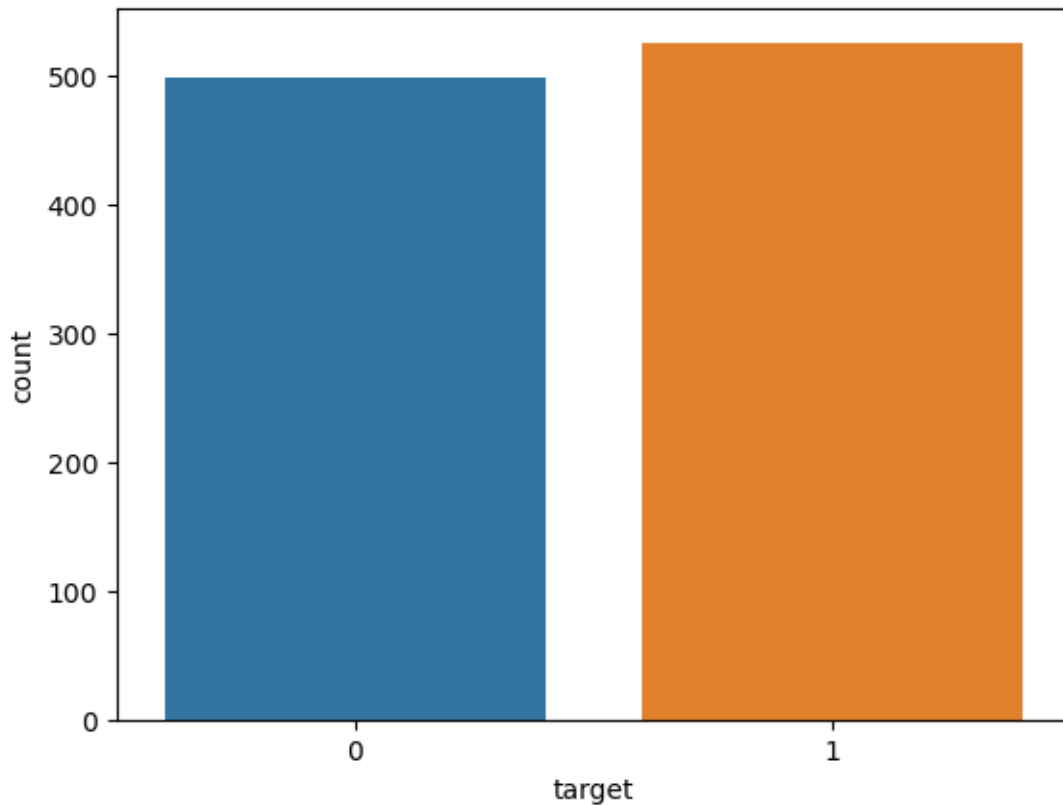
```
[13]: ## how many peoples have heart diseases and how many people dont have heart
      ↪disease.
      data.columns
```

```
[13]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
          'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
          dtype='object')
```

```
[14]: data['target'].value_counts()
```

```
[14]: 1    526
      0    499
      Name: target, dtype: int64
```

```
[19]: sns.countplot(x='target',data=data)
      plt.show()
```



0.1 above mentioned visualisation we how many peoples are heart deases and how many peoples not heart deseases

```
[25]: ## find the count of male and female in this datasets.
data.columns
```

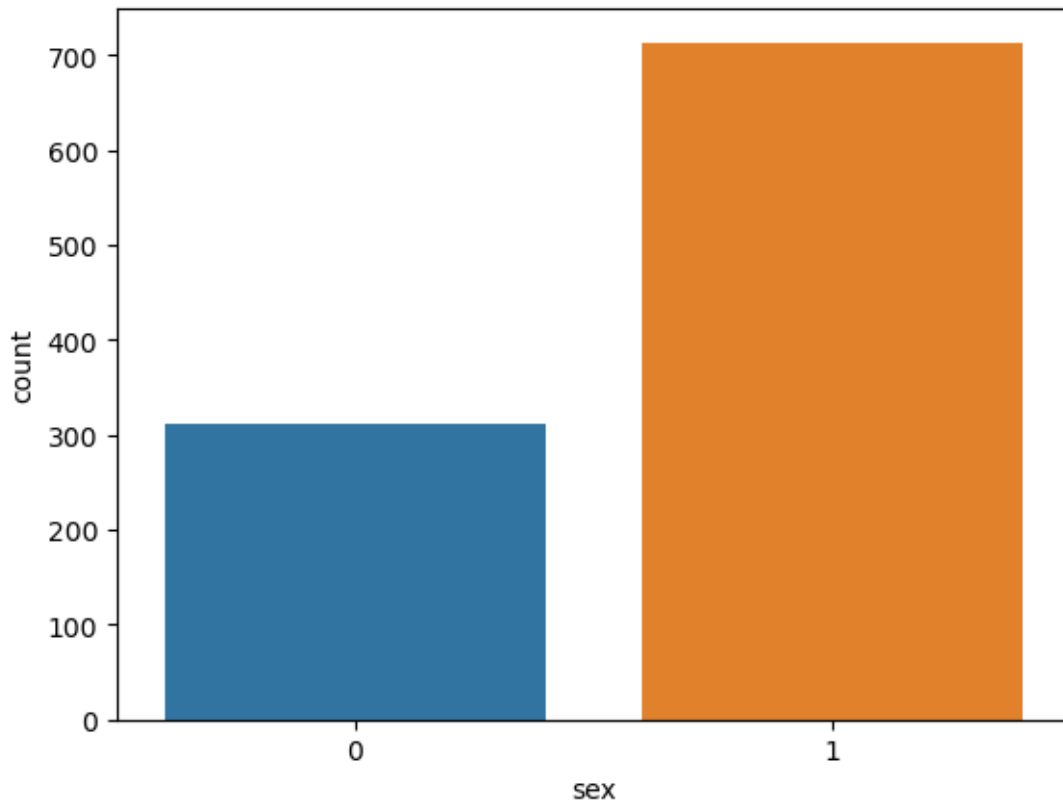
```
[25]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
        dtype='object')
```

```
[27]: data['sex'].value_counts()
```

```
[27]: 1    713
      0    312
      Name: sex, dtype: int64
```

```
[32]: sns.countplot(x='sex',data=data)
```

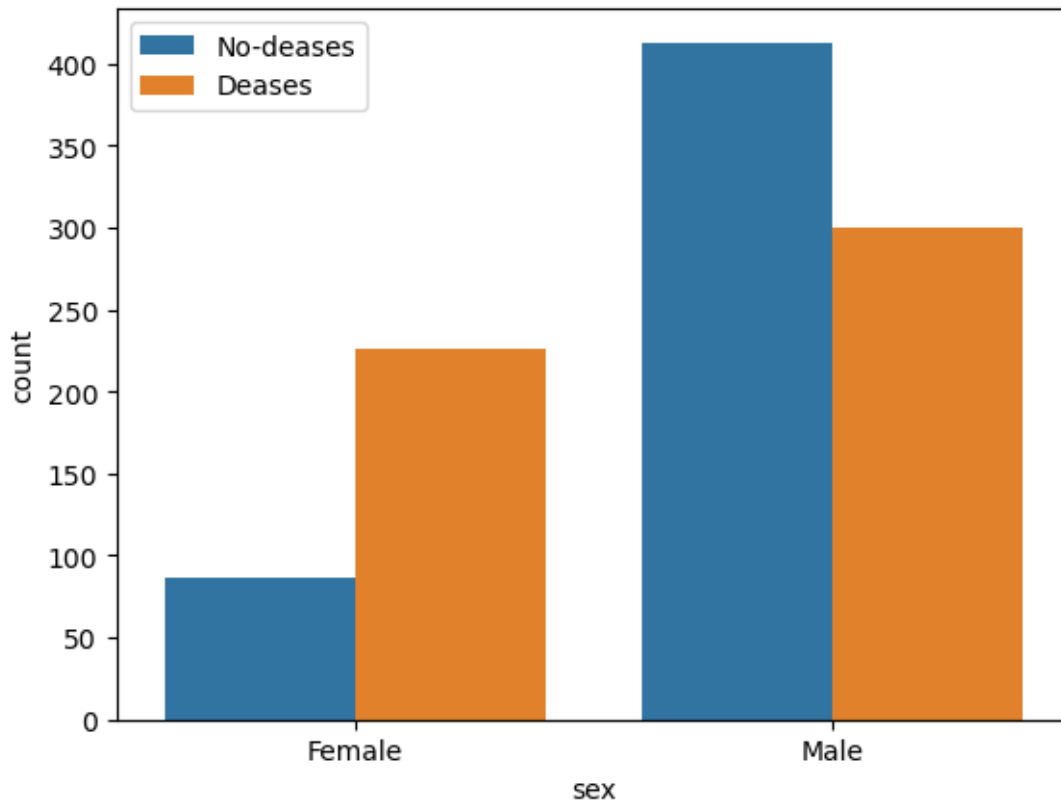
```
[32]: <Axes: xlabel='sex', ylabel='count'>
```



```
[33]: ## find the gender destribution according to the target variable.  
data.columns
```

```
[33]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',  
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],  
        dtype='object')
```

```
[36]: sns.countplot(x='sex',hue='target',data=data)  
plt.xticks([1,0],['Male','Female'])  
plt.legend(labels=['No-deases','Deases'])  
plt.show()
```



```
[37]: ## check the age distribution in datasets
data.columns
```

```
[37]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
        dtype='object')
```

```
[40]: sns.distplot(data['age'],bins=30)
plt.show()
```

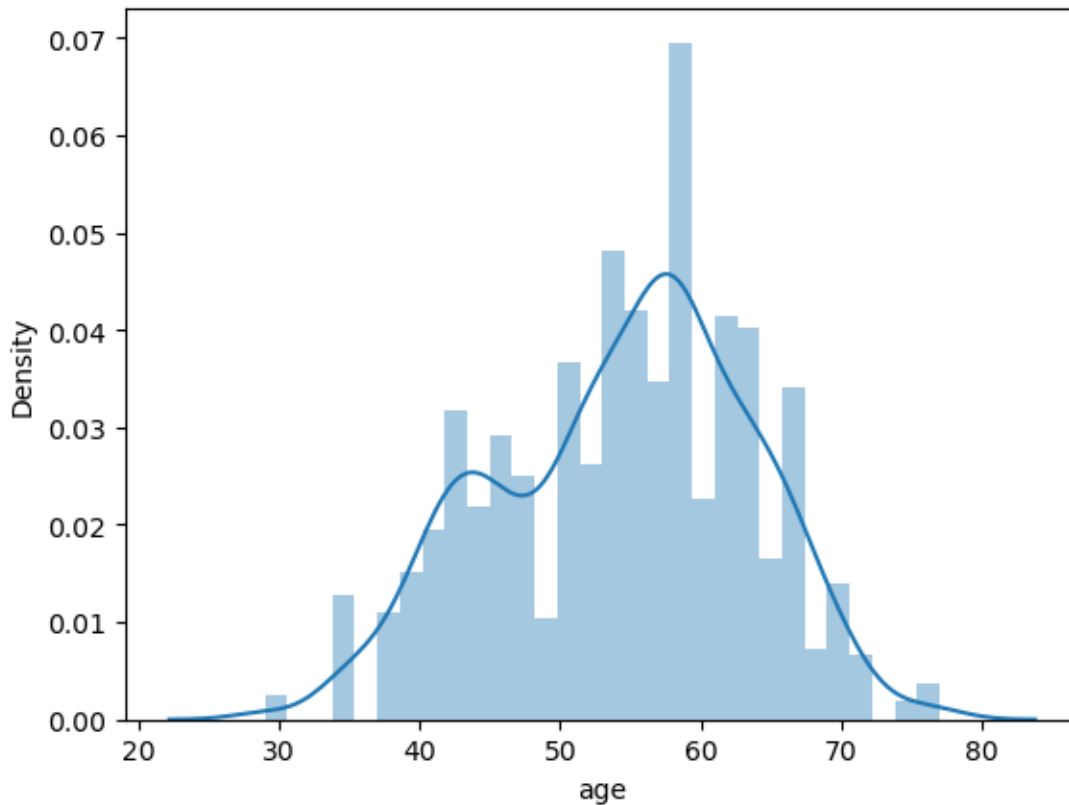
C:\Users\Vikas\AppData\Local\Temp\ipykernel_26776\3186824094.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data['age'],bins=30)
```



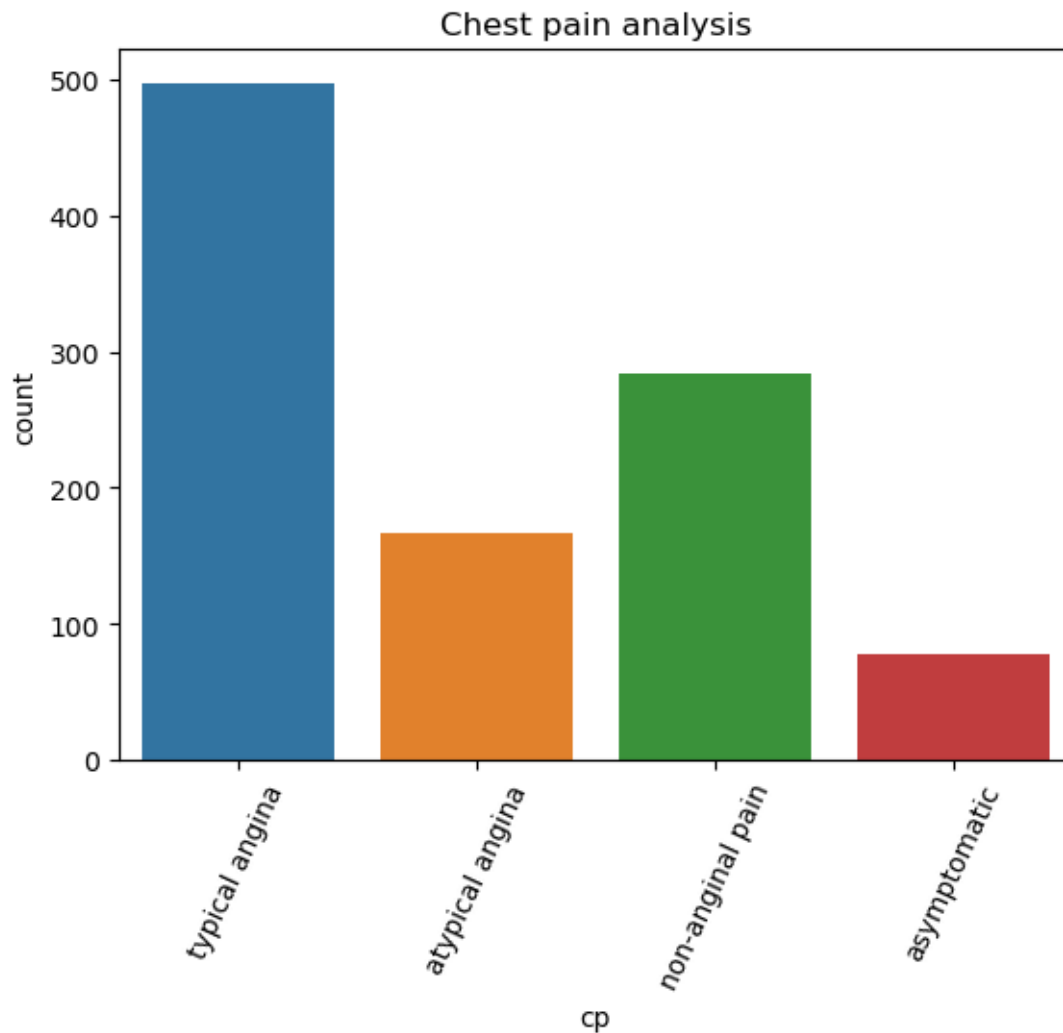
##check the chest pain type, ##value 0: typical angina ##value 1: atypical angina ##value 2: non-anginal pain ##value 3: asymptomatic

```
[42]: data.columns
```

```
[42]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
          'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
          dtype='object')
```

```
[50]: sns.countplot(x='cp',data=data)
plt.xticks([0,1,2,3],["typical angina","atypical angina","non-anginal_
↪pain","asymptomatic"])
plt.xticks(rotation=65)
plt.title("Chest pain analysis")
plt.show
```

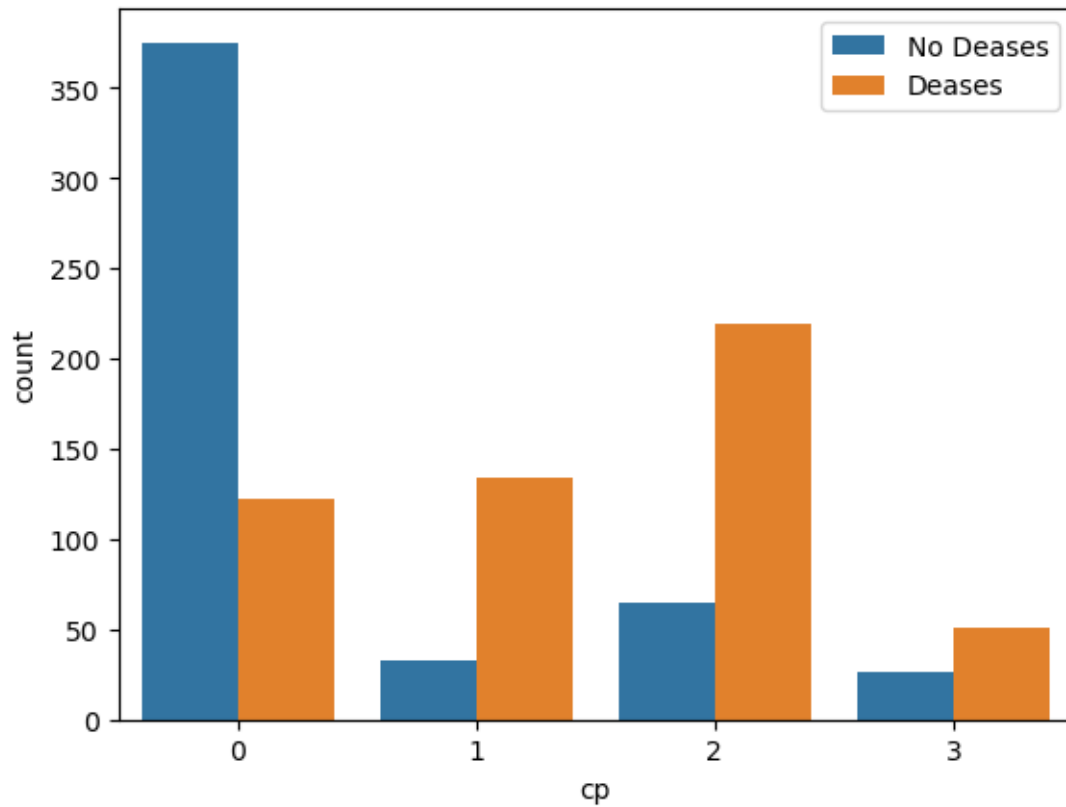
```
[50]: <function matplotlib.pyplot.show(close=None, block=None)>
```

```
[51]: ## show the chest pain destribution as per the target value,
      data.columns
```

```
[51]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
        dtype='object')
```

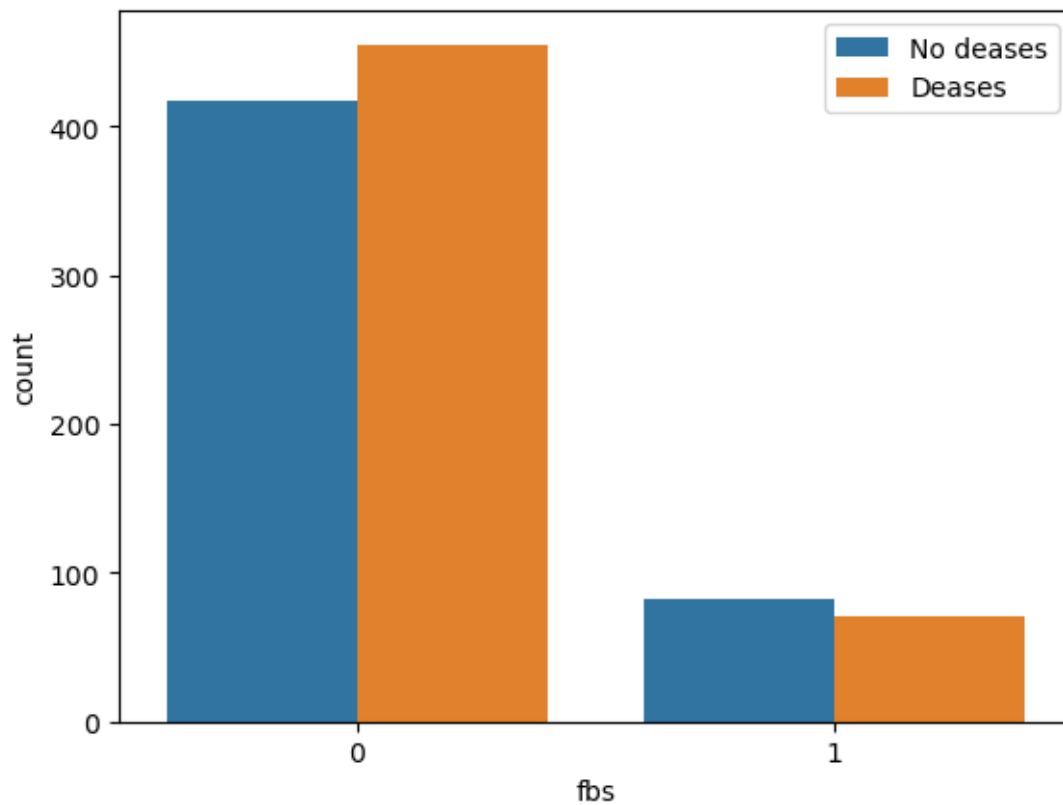
```
[54]: sns.countplot(x="cp",hue="target",data=data)
      plt.legend(labels=["No Deases","Deases"])
      plt.show()
```



```
[55]: ## Show the fasting blood suger destribution according to the target variable,  
data.columns
```

```
[55]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',  
          'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],  
          dtype='object')
```

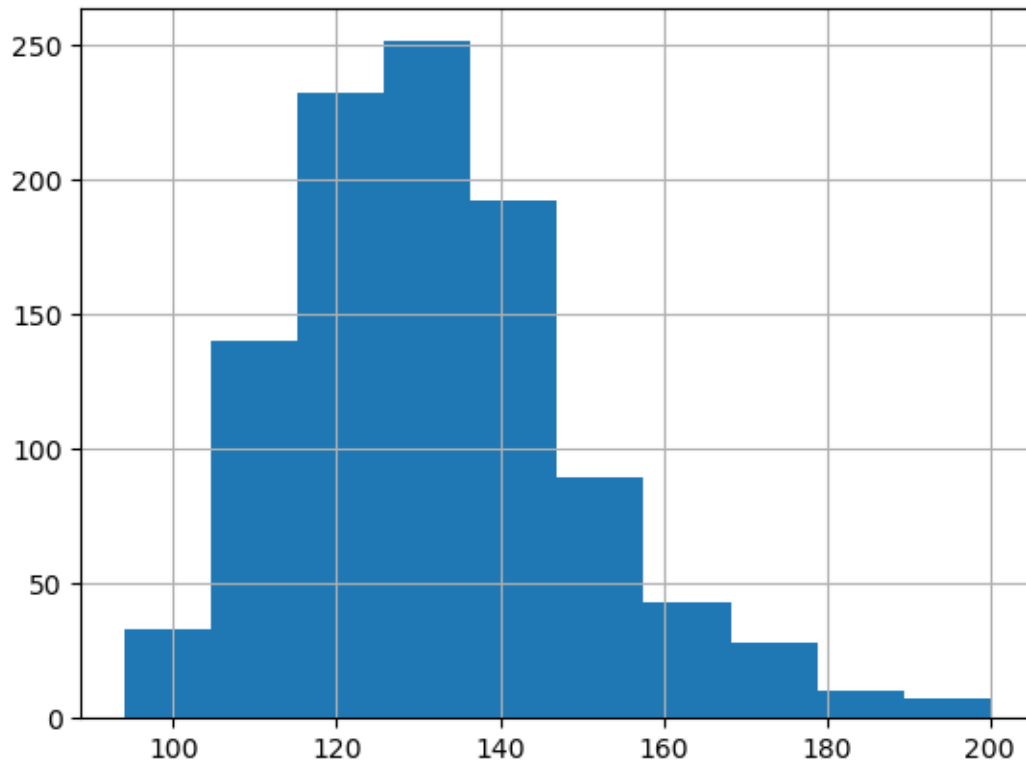
```
[57]: sns.countplot(x="fbs",hue="target",data=data)  
plt.legend(labels=["No deases","Deases"])  
plt.show()
```



```
[59]: ## Check the resting blood pressure Distribution,  
data.columns
```

```
[59]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',  
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],  
        dtype='object')
```

```
[64]: data['trestbps'].hist()  
plt.show()
```



```
[65]: ## C0mpare resting blood pressure as per the sex column,
      data.columns
```

```
[65]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
           'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
          dtype='object')
```

```
[69]: g = sns.FacetGrid(data,hue='sex',aspect=4)
      g.map(sns.kdeplot,'trestbps',shade=True)
      plt.legend(labels=['Male','Female'])
      plt.show()
```

C:\Users\Vikas\anaconda3\lib\site-packages\seaborn\axisgrid.py:848:
FutureWarning:

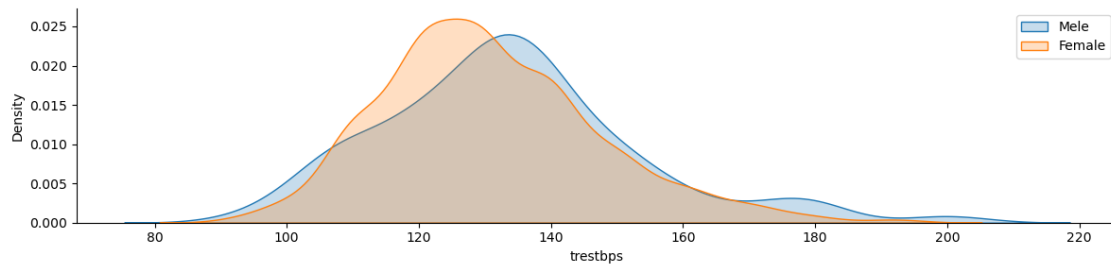
`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

```
func(*plot_args, **plot_kwargs)
C:\Users\Vikas\anaconda3\lib\site-packages\seaborn\axisgrid.py:848:  
FutureWarning:
```

`shade` is now deprecated in favor of `fill`; setting `fill=True`.

This will become an error in seaborn v0.14.0; please update your code.

```
func(*plot_args, **plot_kwargs)
```

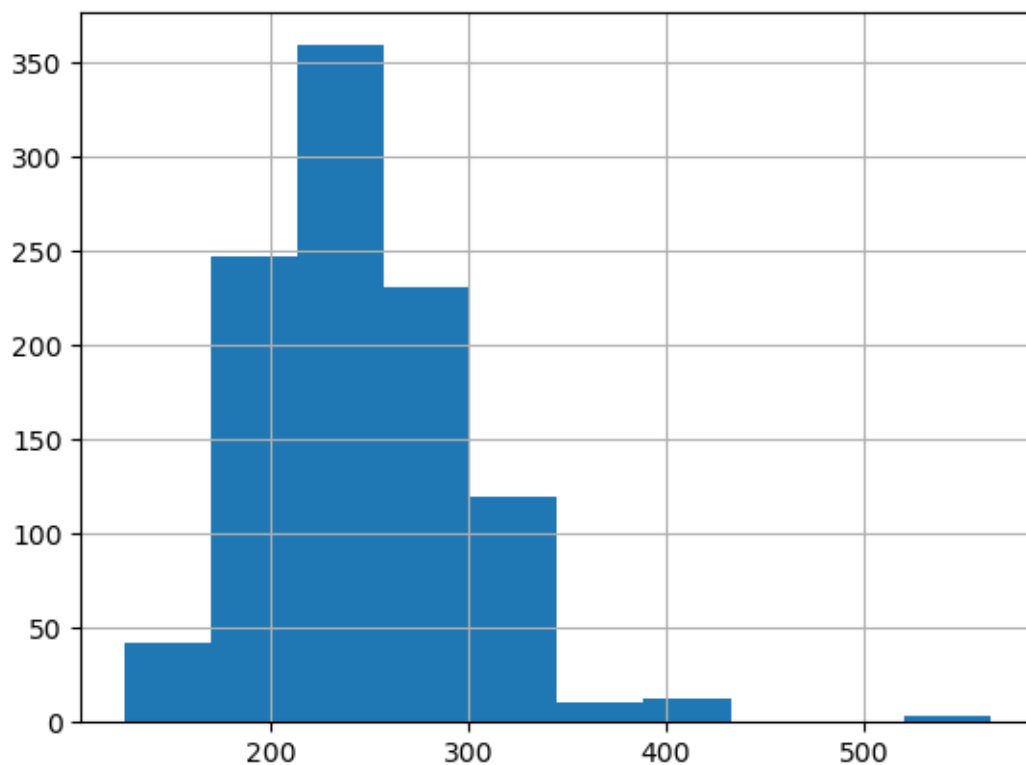


```
[70]: ## show the destribution of serum cholestorol.  
data.columns
```

```
[70]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',  
         'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],  
        dtype='object')
```

```
[71]: data['chol'].hist()
```

```
[71]: <Axes: >
```



```
[72]: ## plot the continues variables.  
data.columns
```

```
[72]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',  
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],  
        dtype='object')
```

```
[74]: cat_val=[]  
cont_val=[]  
for column in data.columns:  
    if data[column].nunique() <=10:  
        cat_val.append(column)  
    else:  
        cont_val.append(column)
```

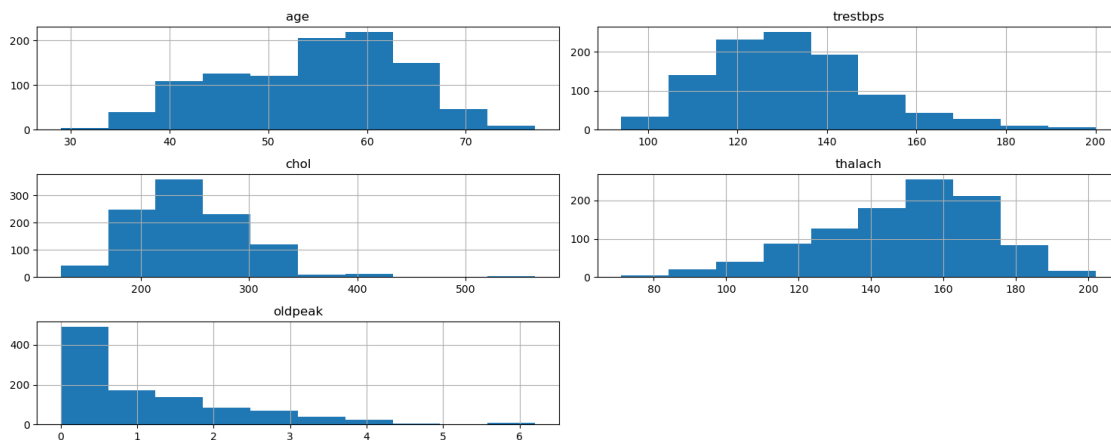
```
[76]: cat_val
```

```
[76]: ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```

```
[77]: cont_val
```

```
[77]: ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
```

```
[84]: data.hist(cont_val,figsize=(15,6))  
plt.tight_layout()  
plt.show()
```



```
[ ]:
```