heartvisualizeaf

May 1, 2025

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

[3]:
```
df=pd.read_csv('heart1.csv')
```

[4]:
```
df
```

[4]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
..	…	…	..	…	…	…	…	…	…	…	…	…	…	…
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

[303 rows x 14 columns]

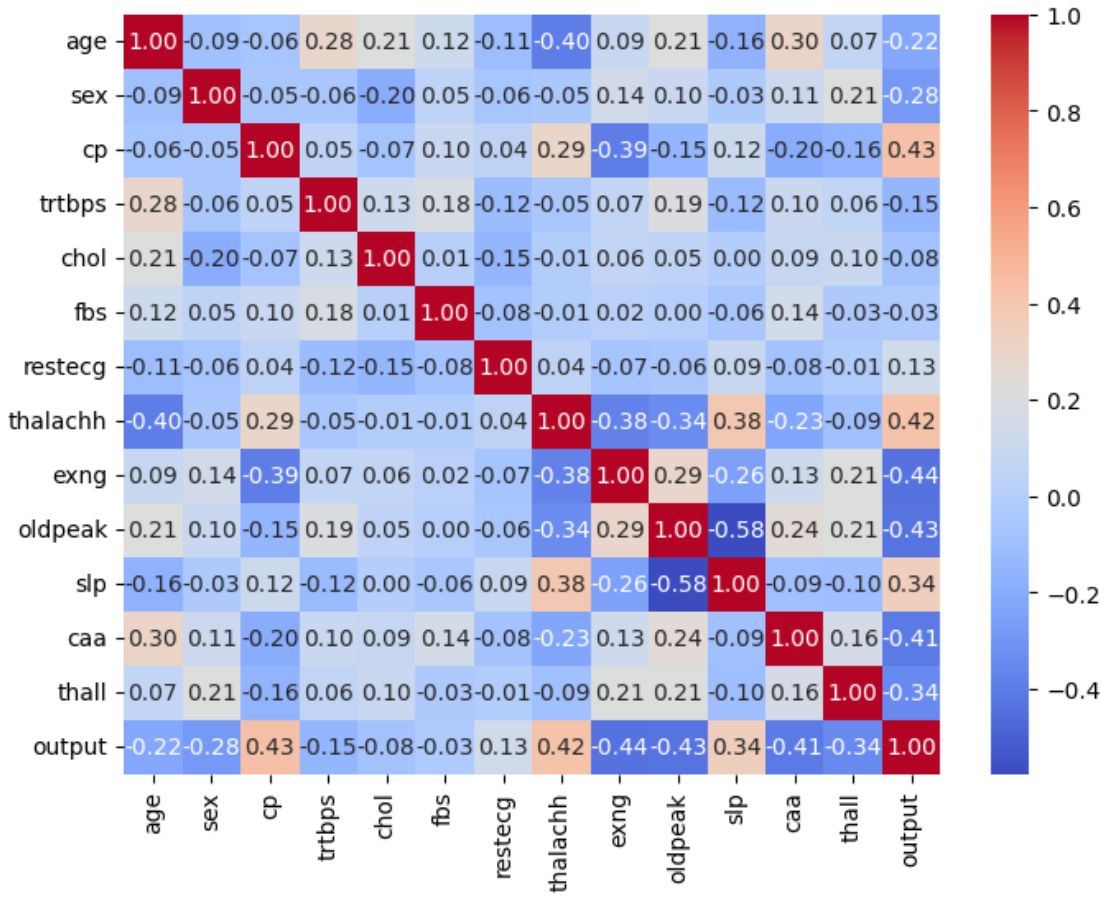
[5]:
```
df=df.drop_duplicates()
```

[6]:
```
df.isna().sum()
```

[6]:
```
age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype: int64
```

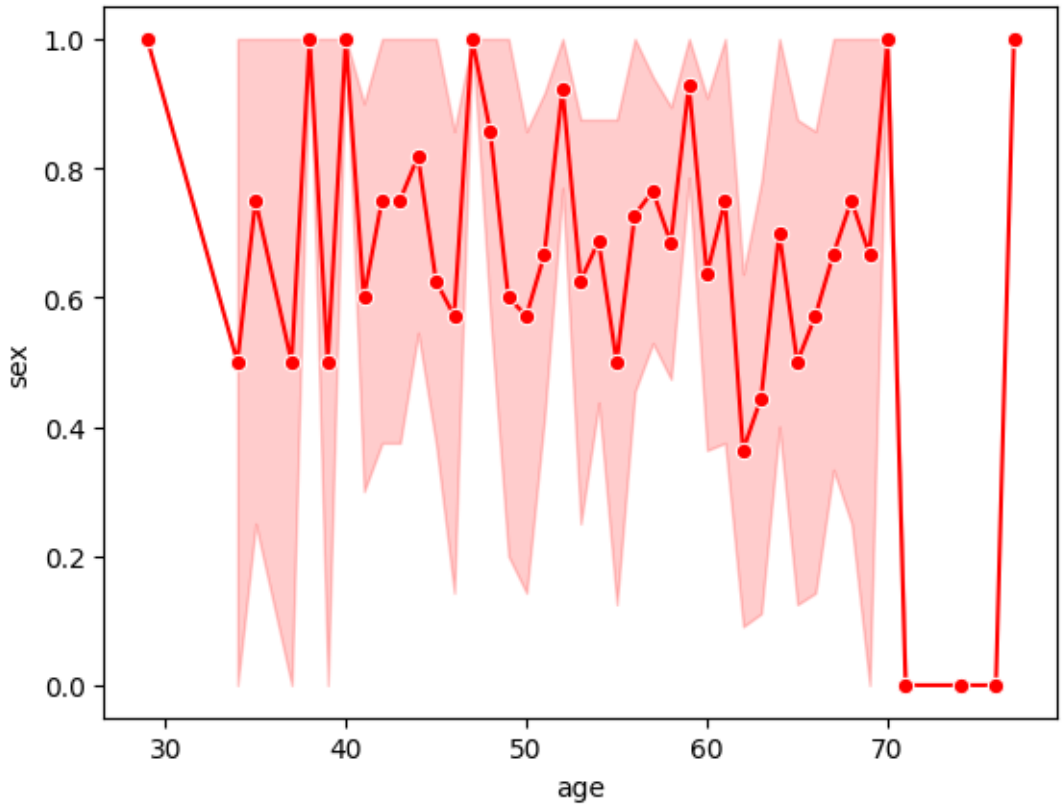
[7]:
```
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(numeric_only=True),annot=True,cmap="coolwarm",fmt='.2f')
```

[7]:
```
<Axes: >
```



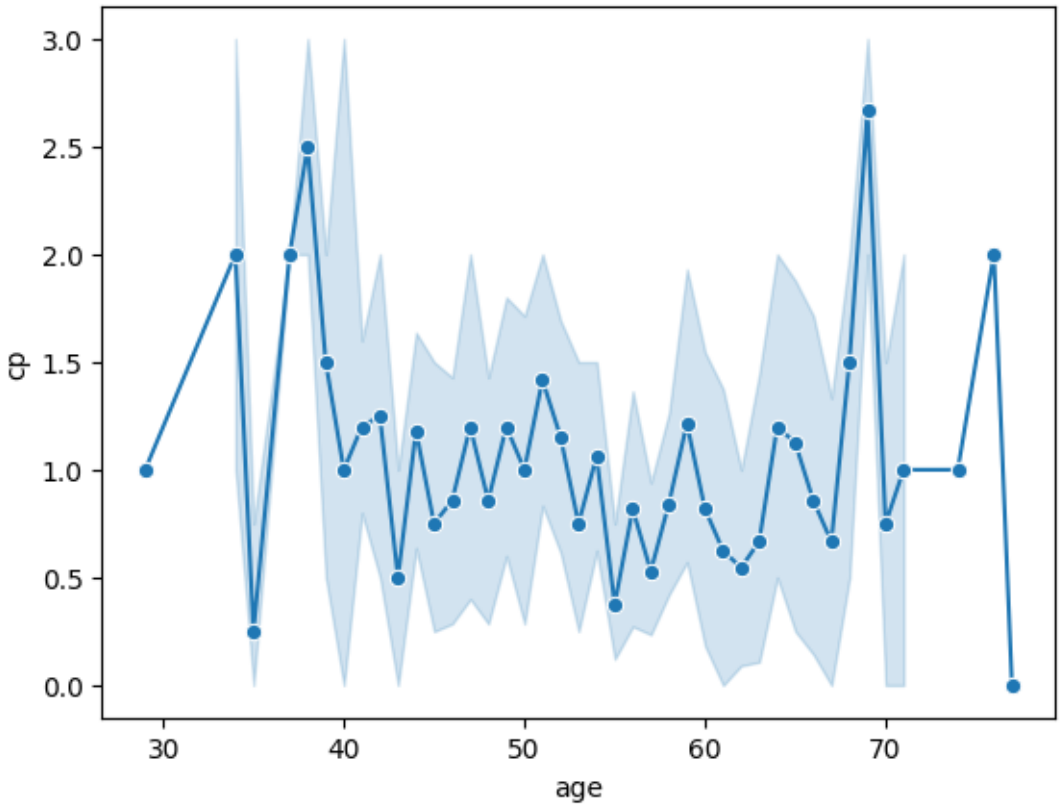
[8]:
```
sns.lineplot(df.sort_values(by='age'),x='age',y='sex',marker='o',color='red')
```

[8]: <Axes: xlabel='age', ylabel='sex'>



[9]:
```
sns.lineplot(df,x='age',y='cp' ,marker='o')
```

[9]:
```
<Axes: xlabel='age', ylabel='cp'>
```

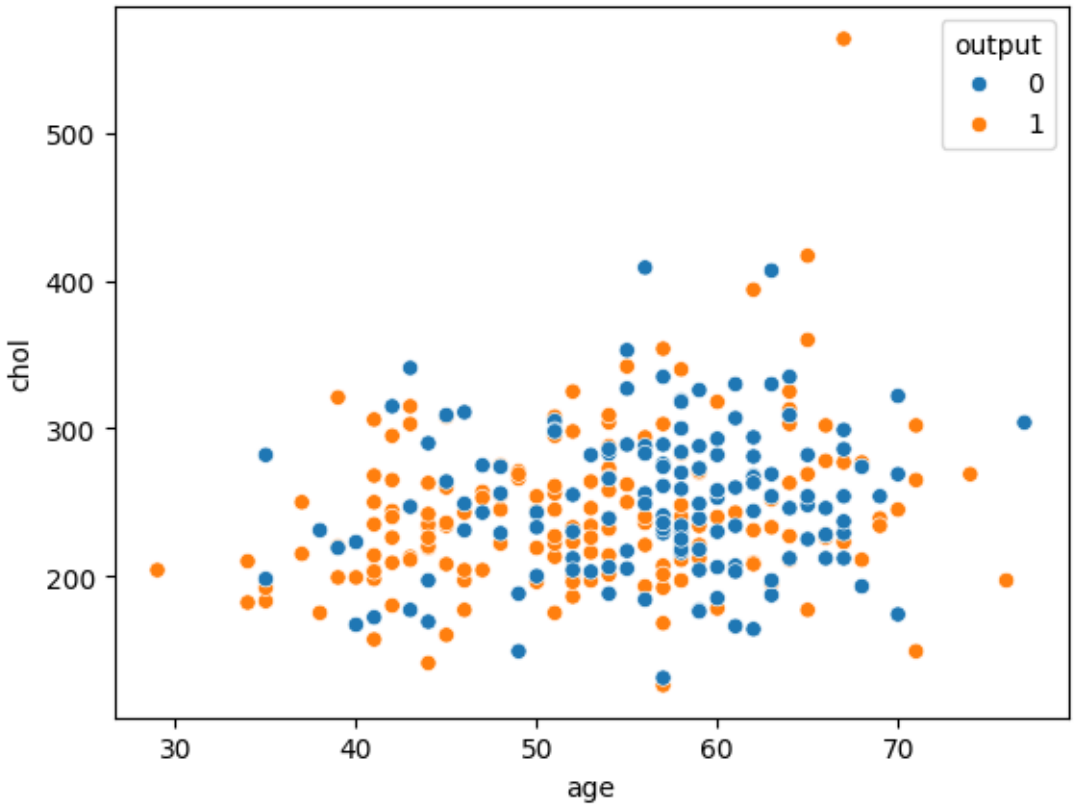


[10]:
```
sns.scatterplot(df,x='age',y='chol',hue='output')
df.head()
```

[10]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	\
0	63	1	3	145	233	1	0	150	0	2.3	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	
3	56	1	1	120	236	0	1	178	0	0.8	2	
4	57	0	0	120	354	0	1	163	1	0.6	2	

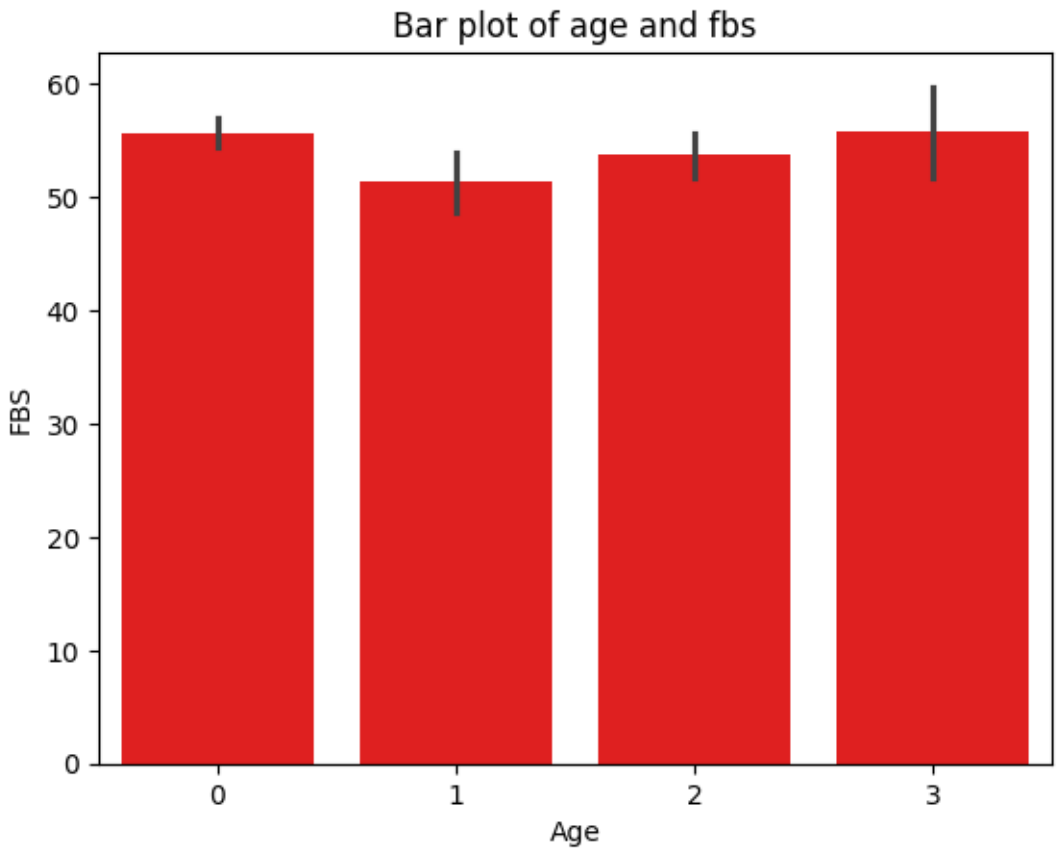
	caa	thall	output
0	0	1	1
1	0	2	1
2	0	2	1
3	0	2	1
4	0	2	1



[ ]:

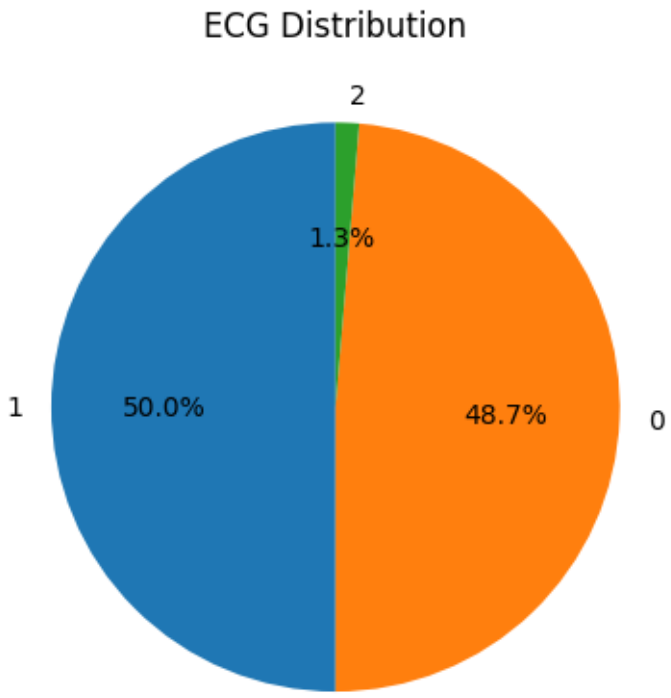
[17]:
```
sns.barplot(df,x='cp',y='age',color='red')
plt.title("Bar plot of age and fbs")
plt.xlabel("Age")
plt.ylabel("FBS")
```

[17]: Text(0, 0.5, 'FBS')



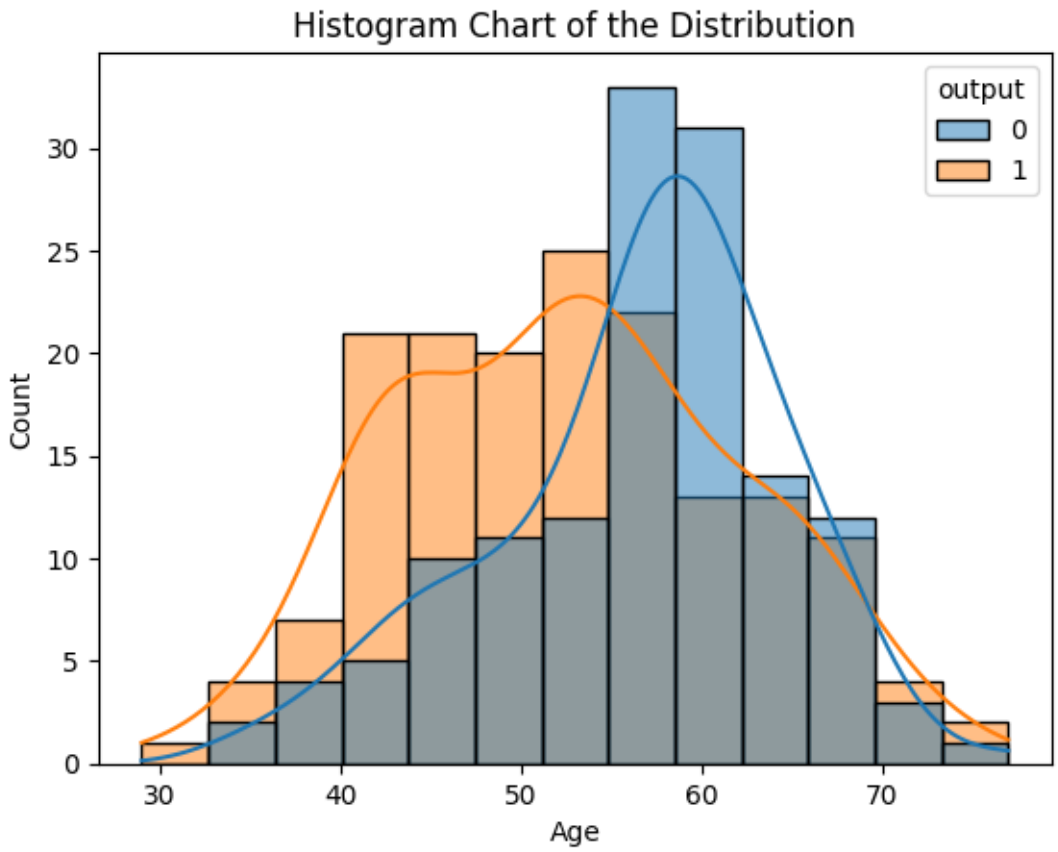
[25]:
```
gender_count=df['restecg'].value_counts()
labels=['Yes','No']
plt.pie(gender_count,labels=gender_count.index,autopct='%1.1f%%',startangle=90)
plt.title("ECG Distribution")
```

[25]: Text(0.5, 1.0, 'ECG Distribution')



[13]:
```
sns.histplot(data=df,x='age',kde=True,hue='output')
plt.title("Histogram Chart of the Distribution")
plt.xlabel('Age')
plt.ylabel('Count')
```

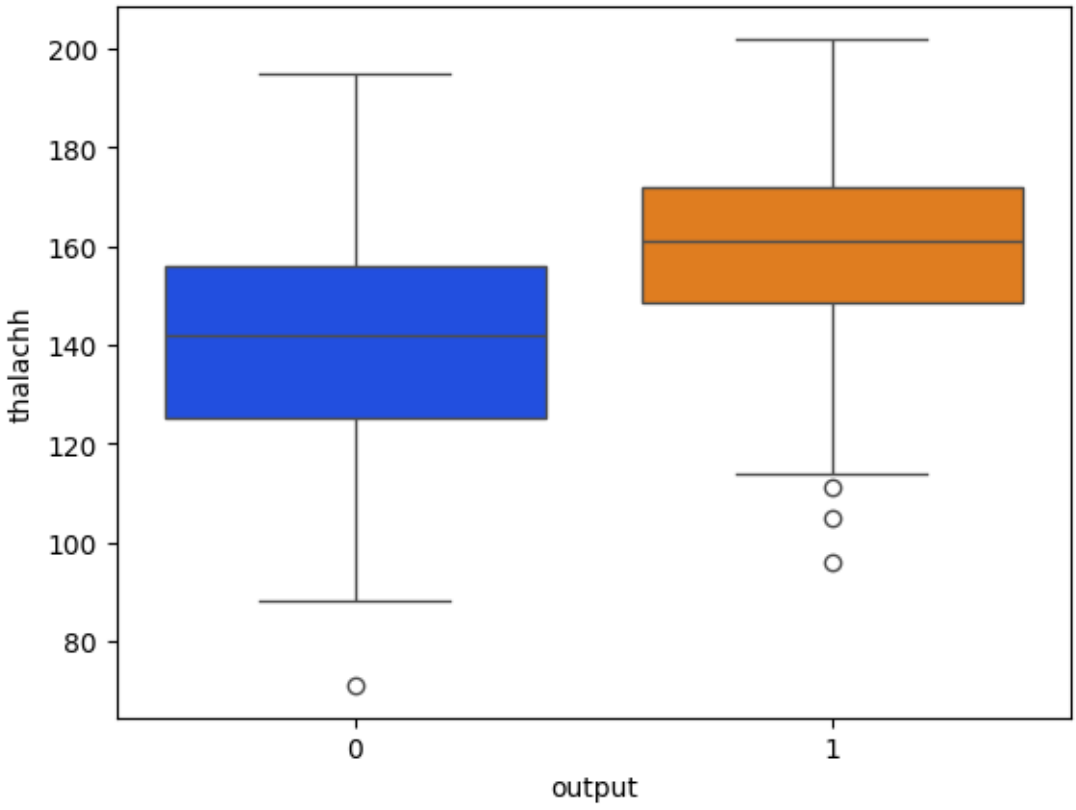
[13]:
```
Text(0, 0.5, 'Count')
```



[14]:
```
sns.boxplot(data=df, x='output',y='thalachh',palette='bright')
```

```
C:\Users\AMOL\AppData\Local\Temp\ipykernel_1496\1892671067.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(data=df, x='output',y='thalachh',palette='bright')
```

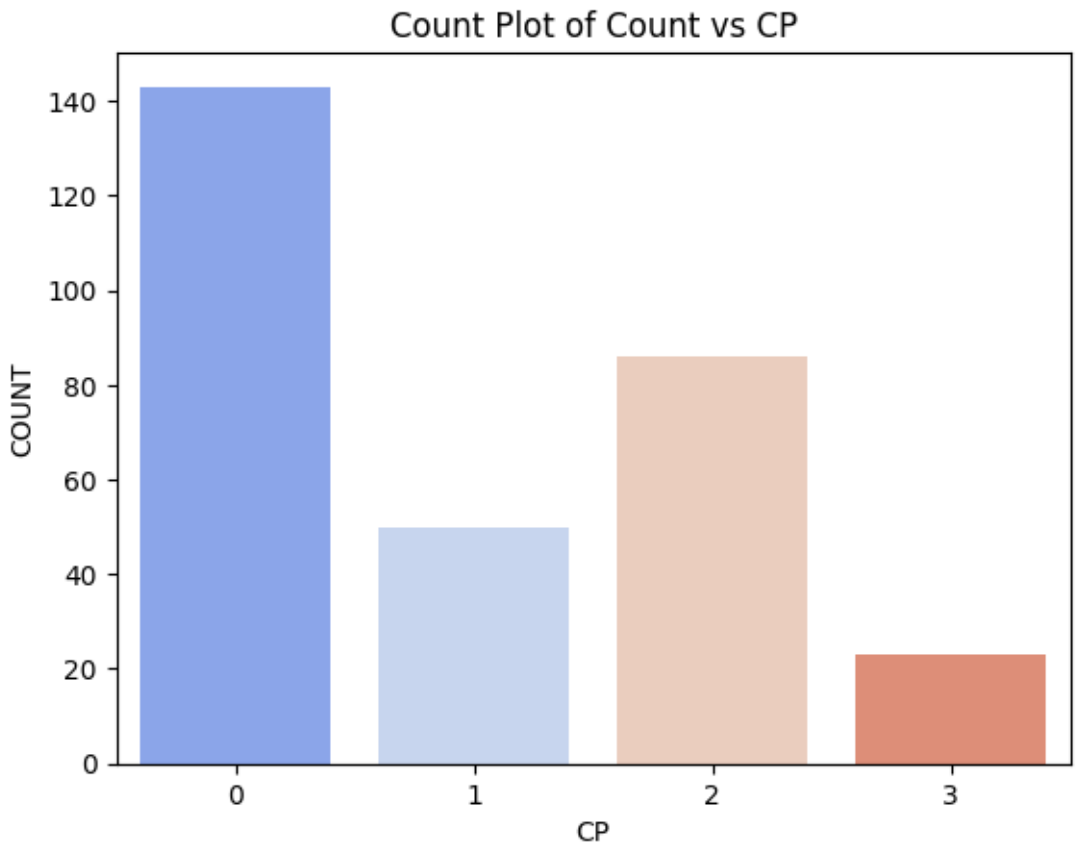
[14]:
```
<Axes: xlabel='output', ylabel='thalachh'>
```



[15]:
```
sns.countplot(df,x=df['cp'],palette='coolwarm')
plt.title('Count Plot of Count vs CP')
plt.xlabel("CP")
plt.ylabel("COUNT")
```

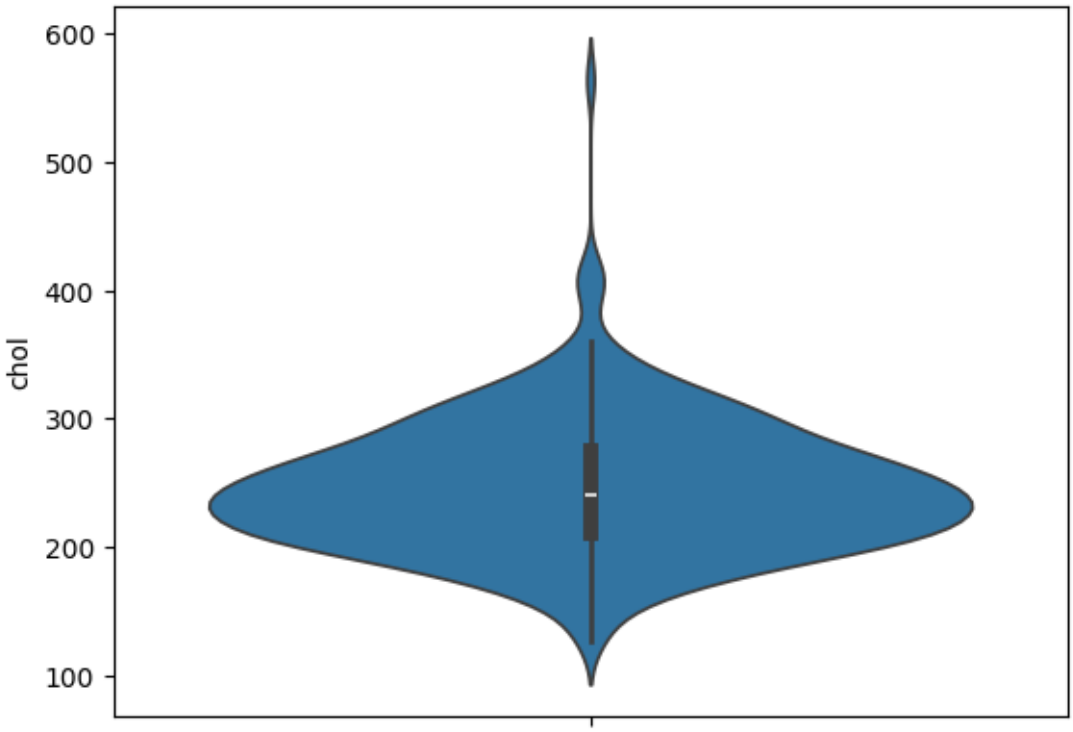
```
C:\Users\AMOL\AppData\Local\Temp\ipykernel_1496\302248787.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(df,x=df['cp'],palette='coolwarm')
```

[15]:
```
Text(0, 0.5, 'COUNT')
```



[16]:
```
sns.violinplot(df,y='chol')
```

[16]:
```
<Axes: ylabel='chol'>
```


600
500
400
300
200
100
chol