

```
In [ ]: #HEART DISEASE ANALYSIS
```

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

```
In [3]: file = ("C:/Users/pc/Videos/python/heart.csv")
data=pd.read_csv(file)
```

```
In [5]: data.head()
```

Out[5]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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

```
In [6]: data.describe()
```

Out[6]:

	age	sex	cp	trestbps	chol	fbs	restecg	thal
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.00
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.64
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.90
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.00
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.50
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.00
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.00
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.00

```
In [7]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   age         303 non-null   int64
 1   sex         303 non-null   int64
 2   cp          303 non-null   int64
 3   trestbps    303 non-null   int64
 4   chol        303 non-null   int64
 5   fbs         303 non-null   int64
 6   restecg     303 non-null   int64
 7   thalach     303 non-null   int64
 8   exang       303 non-null   int64
 9   oldpeak     303 non-null   float64
10   slope       303 non-null   int64
11   ca          303 non-null   int64
12   thal        303 non-null   int64
13   target      303 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

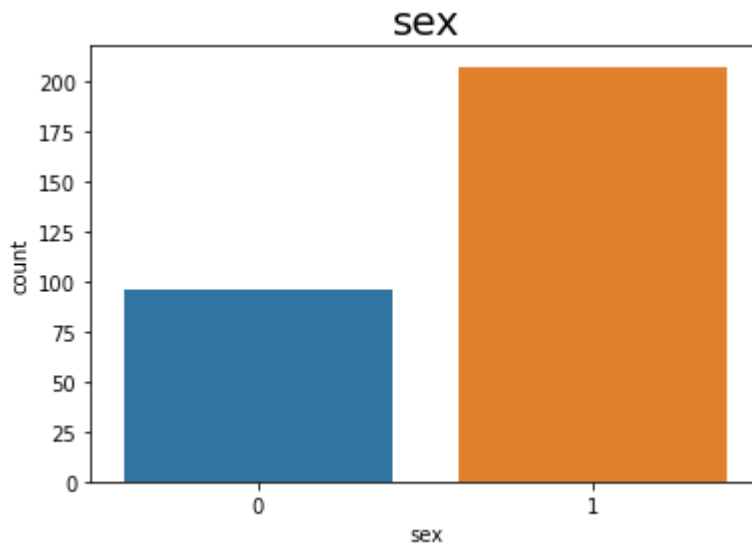
```
In [8]: data.isnull().sum()
```

```
Out[8]: 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
```

```
In [17]: %matplotlib inline
sns.countplot(data['sex'])
plt.title('sex', fontsize = 20)
plt.show()
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```



```
In [20]: print(data['sex'].value_counts())
print()
print(data['sex'].value_counts(normalize=True))
```

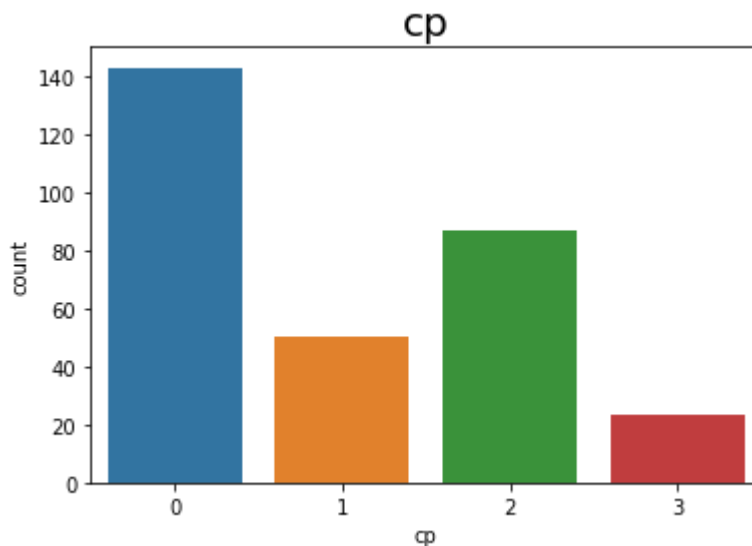
```
1    207
0     96
Name: sex, dtype: int64
```

```
1    0.683168
0    0.316832
Name: sex, dtype: float64
```

```
In [12]: %matplotlib inline
sns.countplot(data['cp'])
plt.title('cp', fontsize = 20)
plt.show()
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



```
In [21]: print(data['cp'].value_counts())
print()
print(data['cp'].value_counts(normalize=True))
```

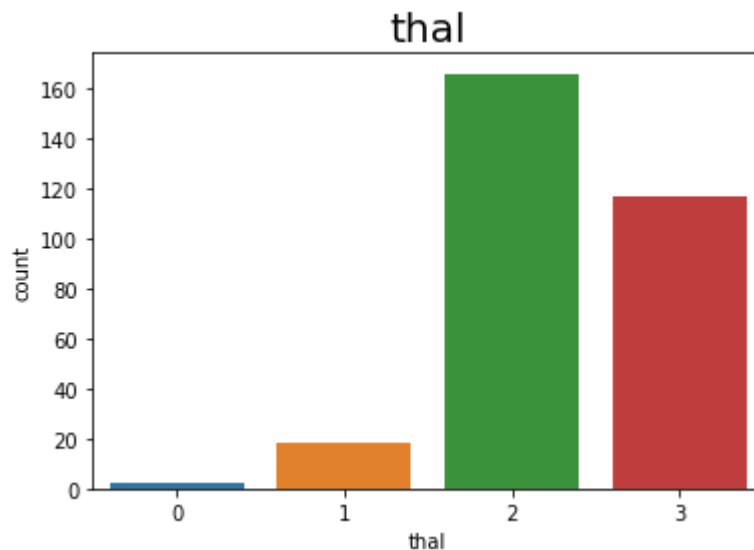
```
0    143
2     87
1     50
3     23
Name: cp, dtype: int64
```

```
0    0.471947
2    0.287129
1    0.165017
3    0.075908
Name: cp, dtype: float64
```

```
In [14]: %matplotlib inline
sns.countplot(data['thal'])
plt.title('thal', fontsize = 20)
plt.show()
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



```
In [22]: print(data['thal'].value_counts())
print()
print(data['thal'].value_counts(normalize=True))
```

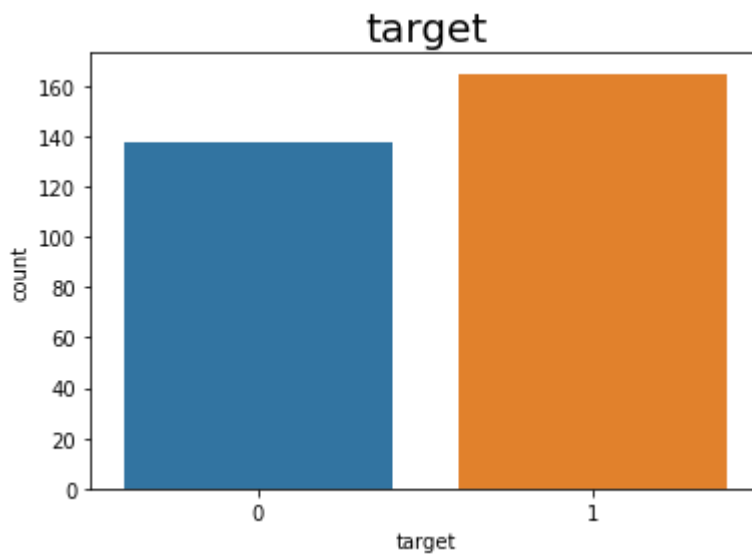
```
2    166
3    117
1     18
0      2
Name: thal, dtype: int64
```

```
2    0.547855
3    0.386139
1    0.059406
0    0.006601
Name: thal, dtype: float64
```

```
In [15]: %matplotlib inline
sns.countplot(data['target'])
plt.title('target', fontsize = 20)
plt.show()
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```



```
In [23]: print(data['target'].value_counts())
print()
print(data['target'].value_counts(normalize=True))
```

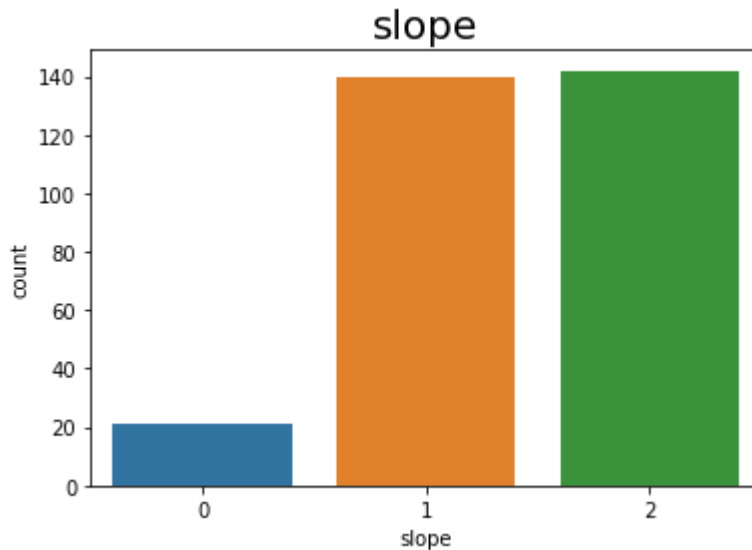
```
1    165
0    138
Name: target, dtype: int64

1    0.544554
0    0.455446
Name: target, dtype: float64
```

```
In [19]: %matplotlib inline
sns.countplot(data['slope'])
plt.title('slope', fontsize = 20)
plt.show()
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



```
In [24]: print(data['slope'].value_counts())
print()
print(data['slope'].value_counts(normalize=True))
```

```
2    142
1    140
0     21
Name: slope, dtype: int64
```

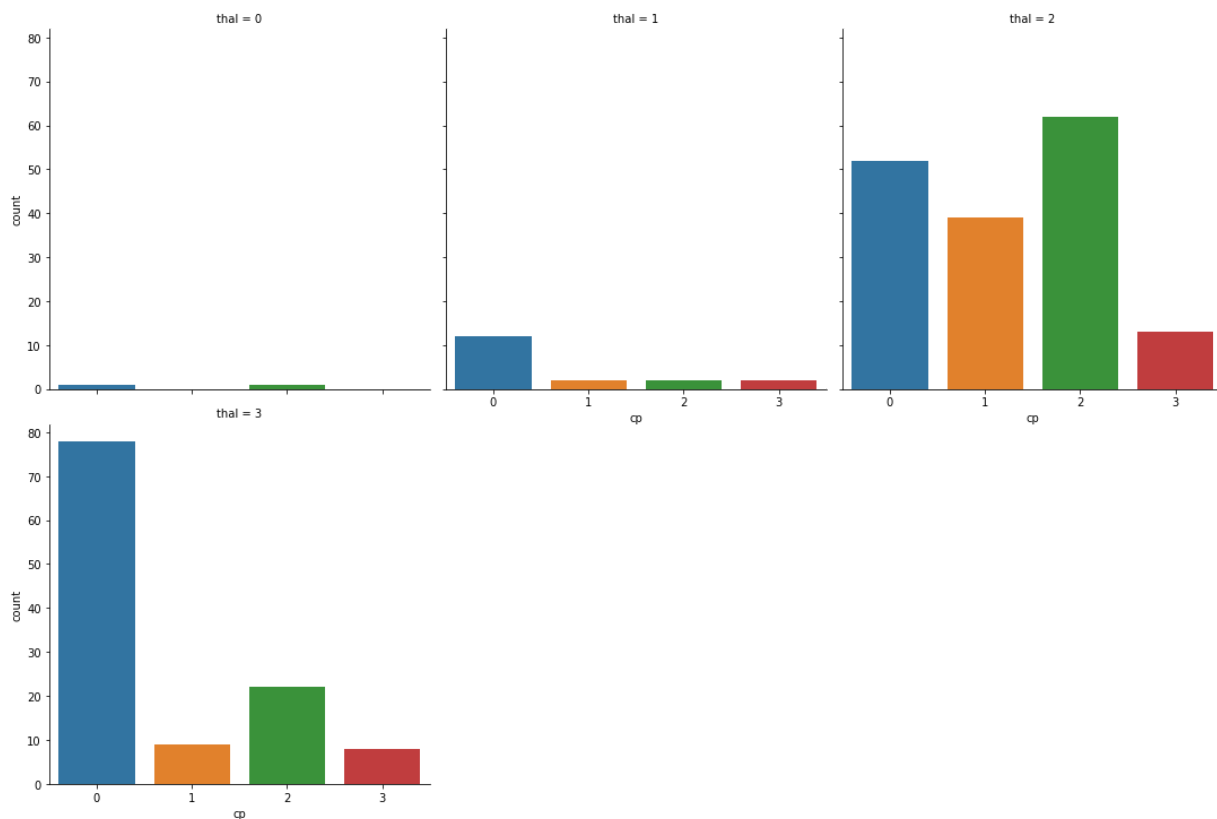
```
2    0.468647
1    0.462046
0    0.069307
Name: slope, dtype: float64
```

```
In [ ]: #BIVARIATE ANALYSIS
```

```
In [25]: g = sns.catplot("cp", col="thal", col_wrap=3, data=data, kind="count", height=5, a  
plt.show()
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

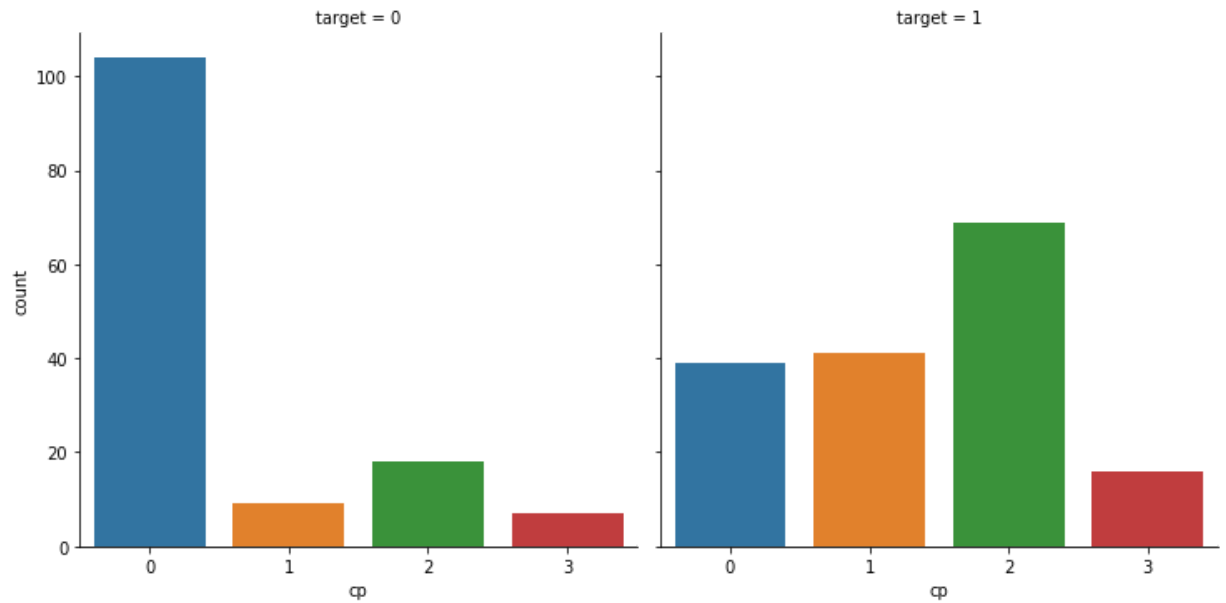
```
warnings.warn(
```




```
In [26]: g = sns.catplot("cp", col="target", col_wrap=3, data=data, kind="count", height=5,  
plt.show())
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

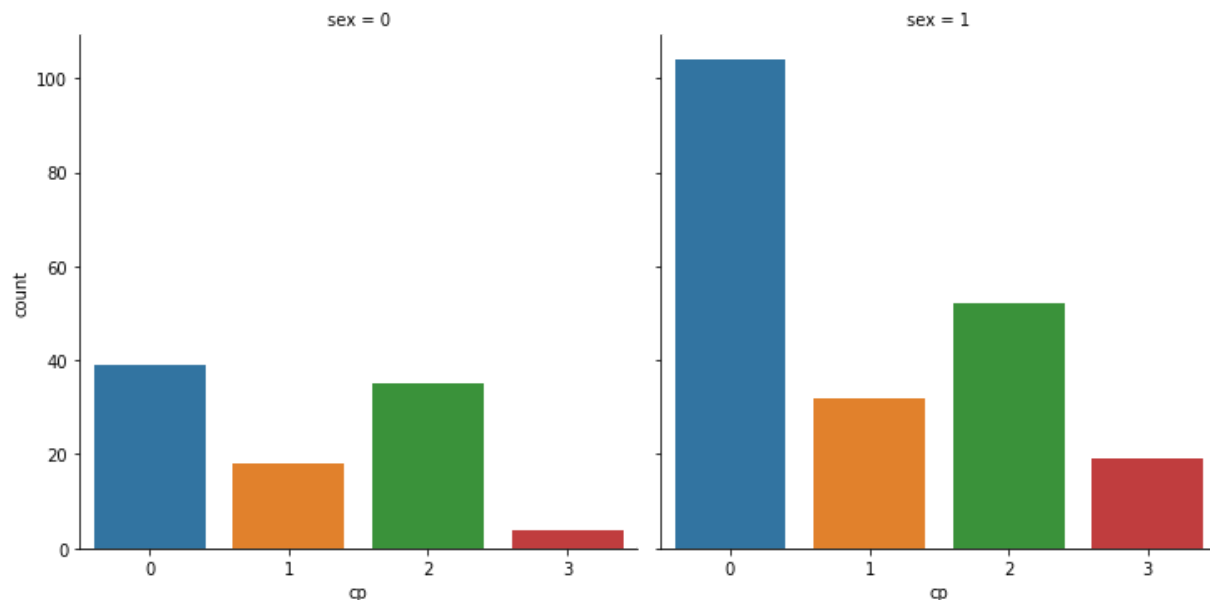
warnings.warn(



```
In [27]: g = sns.catplot("cp", col="sex", col_wrap=3, data=data, kind="count", height=5, as
plt.show())
```

C:\Users\pc\Downloads\anaconda\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 argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

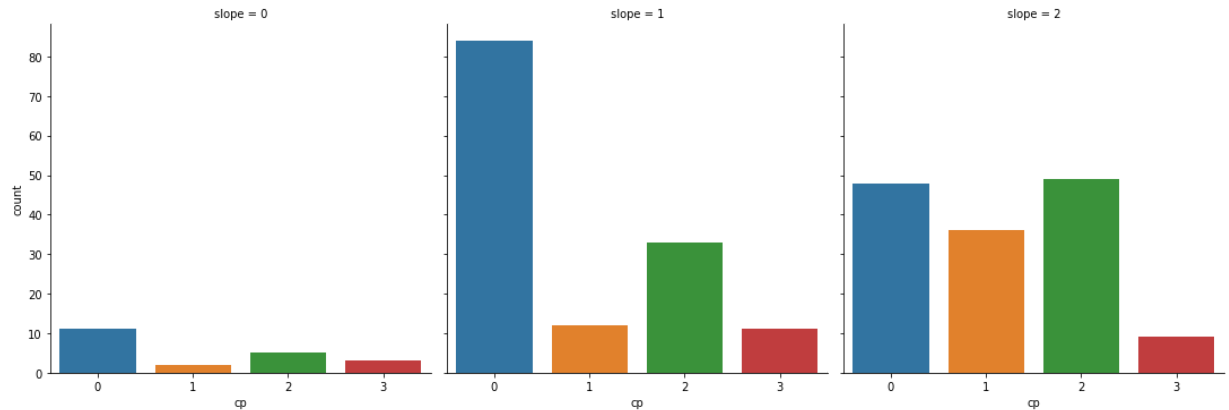
```
warnings.warn(
```



```
In [28]: g = sns.catplot("cp", col="slope", col_wrap=3, data=data, kind="count", height=5,
plt.show())
```

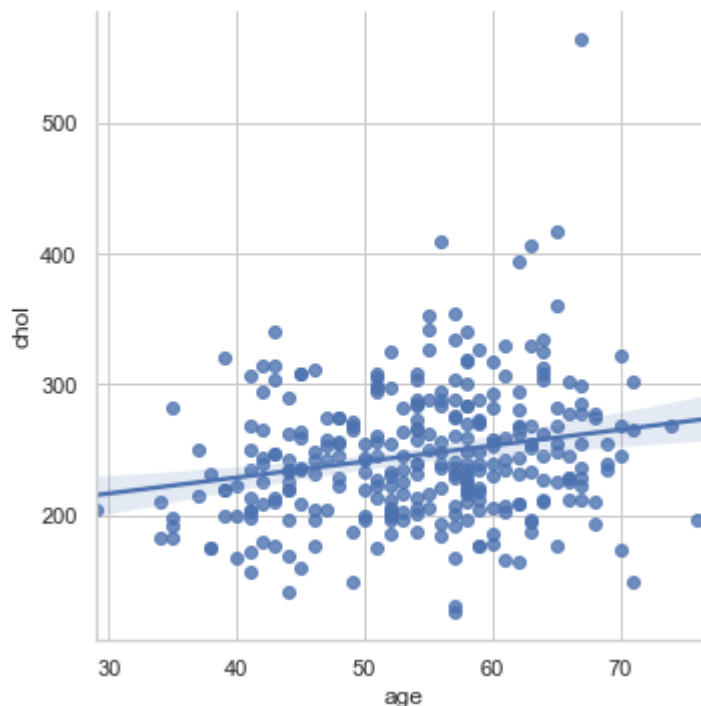
C:\Users\pc\Downloads\anaconda\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



```
In [35]: #CORRELATION
```

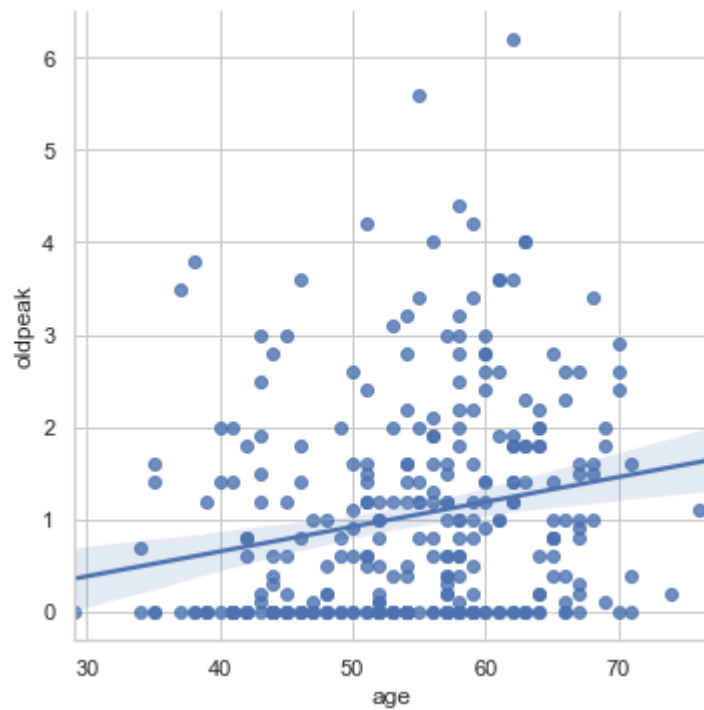
```
sns.set(style="whitegrid")
ax = sns.lmplot(x="age", y="chol", data=data)
```



```
In [ ]: #from the graph, we can see that the reaction was much at the middle.
```

In [37]: *#CORRELATION*

```
sns.set(style="whitegrid")  
ax = sns.lmplot(x="age", y="oldpeak" , data=data)
```



In []: *oldpeak is much compared to the other parts of the graph. Meaning that the correlation*