```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')


df=pd.read_csv('heart.csv')
df
```

	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
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.isnull().sum()
```

	0
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

```
df.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
```

```
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.

```
df.corr()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
age	1.000000	-0.103240	-0.071966	0.271121	0.219823	0.121243	-0.132696	-0.390227	0.088163	0.208137	-0.169105	0.271551	0.072
sex	-0.103240	1.000000	-0.041119	-0.078974	-0.198258	0.027200	-0.055117	-0.049365	0.139157	0.084687	-0.026666	0.111729	0.198
cp	-0.071966	-0.041119	1.000000	0.038177	-0.081641	0.079294	0.043581	0.306839	-0.401513	-0.174733	0.131633	-0.176206	-0.163
trestbps	0.271121	-0.078974	0.038177	1.000000	0.127977	0.181767	-0.123794	-0.039264	0.061197	0.187434	-0.120445	0.104554	0.059
chol	0.219823	-0.198258	-0.081641	0.127977	1.000000	0.026917	-0.147410	-0.021772	0.067382	0.064880	-0.014248	0.074259	0.100
fbs	0.121243	0.027200	0.079294	0.181767	0.026917	1.000000	-0.104051	-0.008866	0.049261	0.010859	-0.061902	0.137156	-0.042
restecg	-0.132696	-0.055117	0.043581	-0.123794	-0.147410	-0.104051	1.000000	0.048411	-0.065606	-0.050114	0.086086	-0.078072	-0.020
thalach	-0.390227	-0.049365	0.306839	-0.039264	-0.021772	-0.008866	0.048411	1.000000	-0.380281	-0.349796	0.395308	-0.207888	-0.098
exang	0.088163	0.139157	-0.401513	0.061197	0.067382	0.049261	-0.065606	-0.380281	1.000000	0.310844	-0.267335	0.107849	0.197
oldpeak	0.208137	0.084687	-0.174733	0.187434	0.064880	0.010859	-0.050114	-0.349796	0.310844	1.000000	-0.575189	0.221816	0.202
slope	-0.169105	-0.026666	0.131633	-0.120445	-0.014248	-0.061902	0.086086	0.395308	-0.267335	-0.575189	1.000000	-0.073440	-0.094
ca	0.271551	0.111729	-0.176206	0.104554	0.074259	0.137156	-0.078072	-0.207888	0.107849	0.221816	-0.073440	1.000000	0.149
thal	0.072297	0.198424	-0.163341	0.059276	0.100244	-0.042177	-0.020504	-0.098068	0.197201	0.202672	-0.094090	0.149014	1.000
target	-0.229324	-0.279501	0.434854	-0.138772	-0.099966	-0.041164	0.134468	0.422895	-0.438029	-0.438441	0.345512	-0.382085	-0.337

Develop at least 7 specific questions, such as:

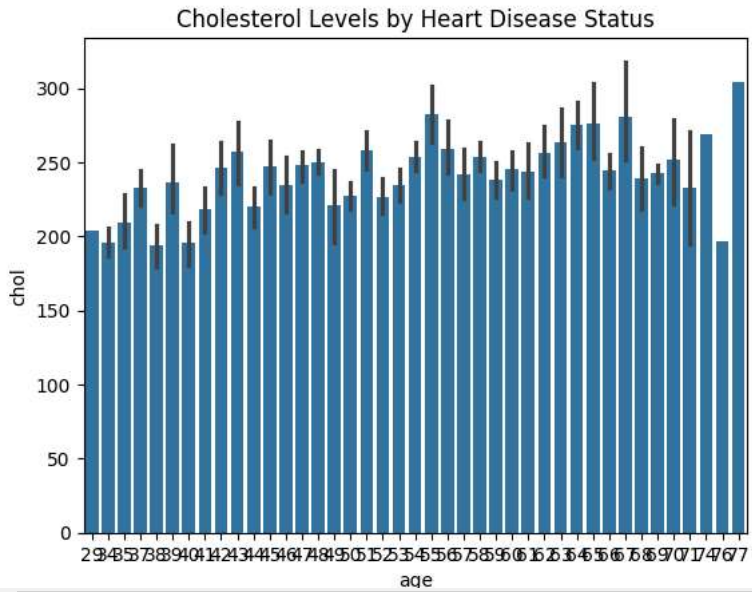
1. What is the average age of patients diagnosed with heart disease?
2. How does cholesterol level correlate with heart disease?
3. Is there a significant difference in Cp prevalence between men and women?
4. Which age group has the highest risk of heart disease(Cp)?
5. How do blood pressure levels vary among Age?
6. Can smoking status or diabetes be predictors of heart disease?
7. What is the most common combination of risk factors in heart disease patients?

```
heart_disease_patients = df[df['age'] == 1]
average_age = heart_disease_patients['age'].mean()
```

```
print(f"The average age of patients diagnosed with heart disease is {average_age:.2f} years.")
```

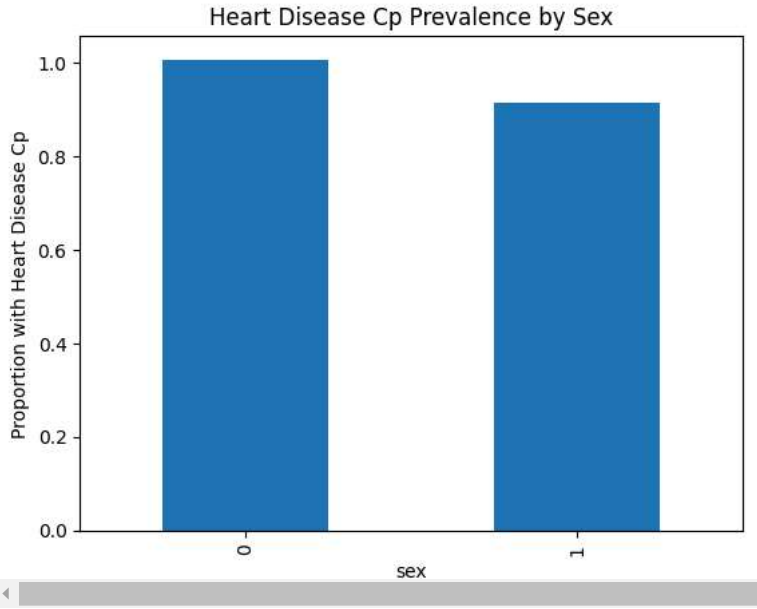
```
The average age of patients diagnosed with heart disease is nan years.
```

```
sns.barplot(x='age', y='chol', data=df)
plt.title("Cholesterol Levels by Heart Disease Status")
plt.show()
```



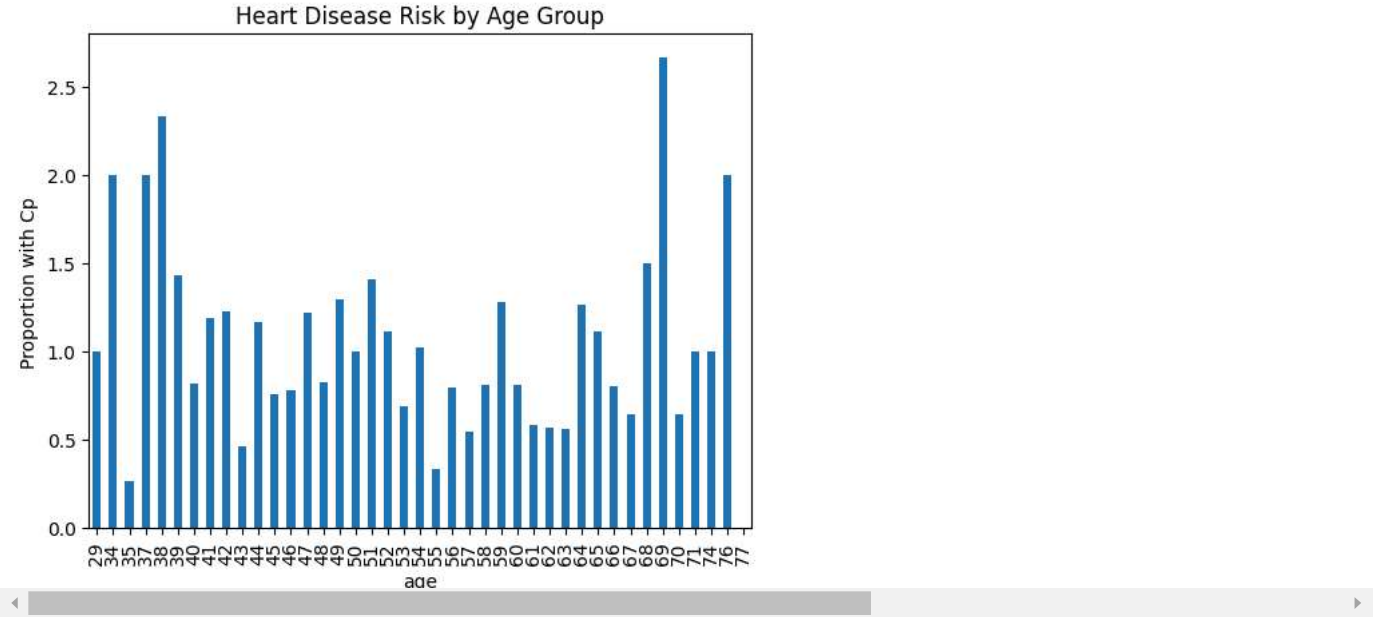

```
gender_heart_disease = df.groupby('sex')['cp'].mean()
print(gender_heart_disease)

gender_heart_disease.plot(kind='bar', title='Heart Disease Cp Prevalence by Sex')
plt.ylabel('Proportion with Heart Disease Cp')
plt.show()
```

```
sex
0    1.006410
1    0.914446
Name: cp, dtype: float64
```

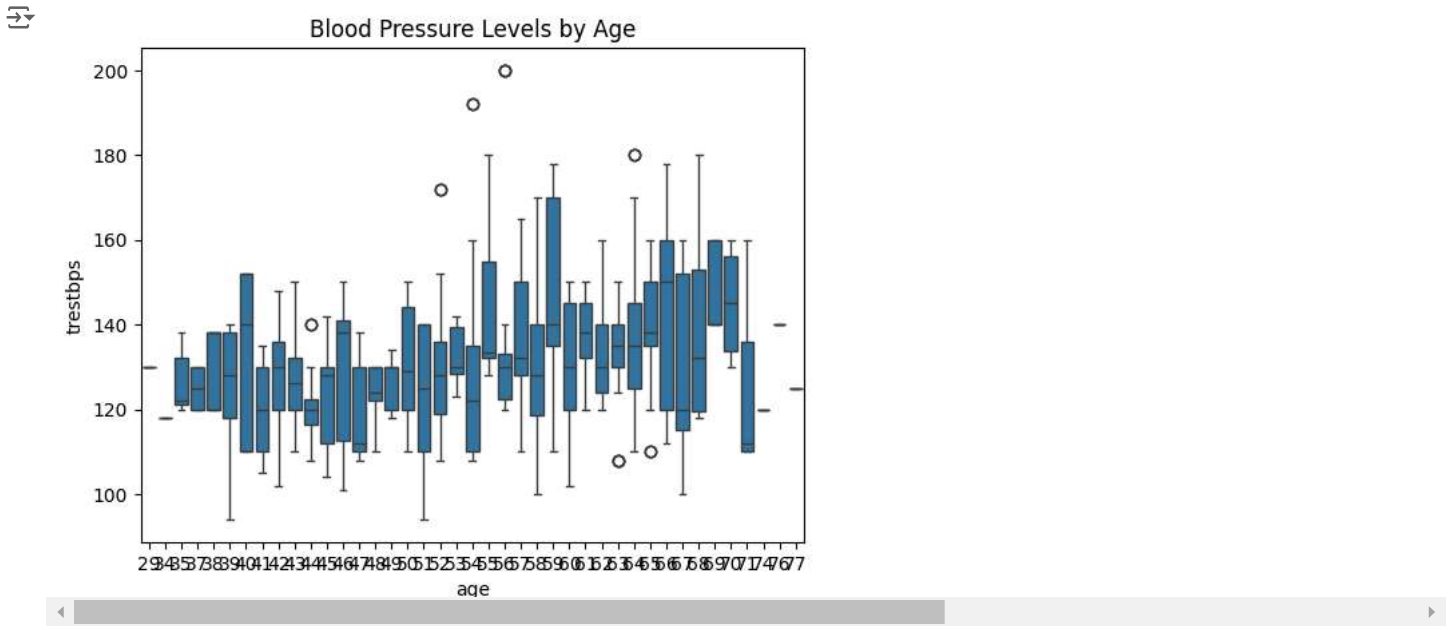



```
age_group_risk = df.groupby('age')['cp'].mean()
print(age_group_risk)
age_group_risk.plot(kind='bar', title='Heart Disease Risk by Age Group')
plt.ylabel('Proportion with Cp')
plt.show()
```

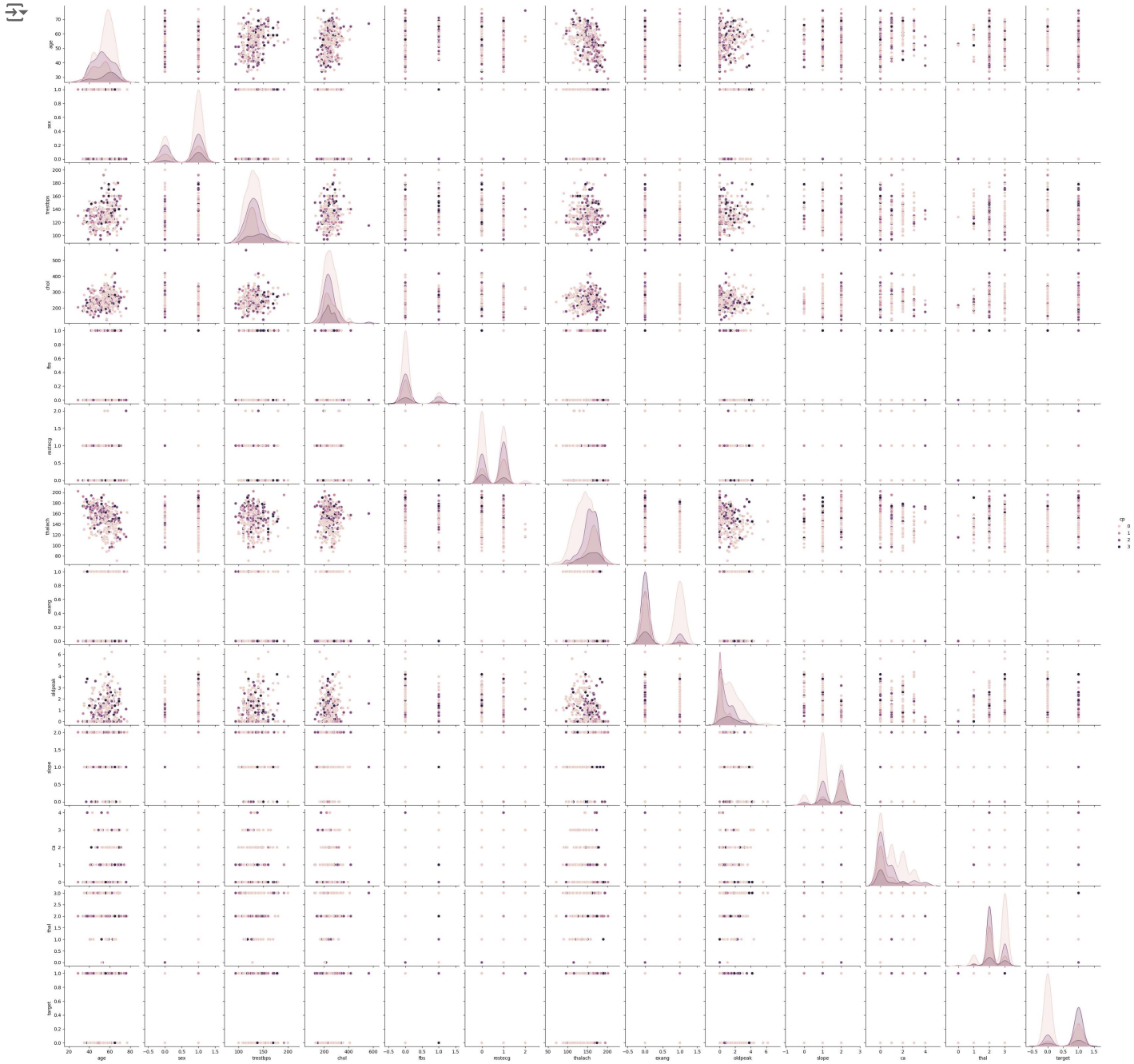
```
age
29    1.000000
34    2.000000
35    0.266667
37    2.000000
38    2.333333
39    1.428571
40    0.818182
41    1.187500
42    1.230769
43    0.461538
44    1.166667
45    0.760000
46    0.782609
47    1.222222
48    0.826087
49    1.294118
50    1.000000
51    1.410256
52    1.116279
53    0.692308
54    1.018868
55    0.333333
56    0.794872
57    0.543860
58    0.808824
59    1.282609
60    0.810811
61    0.580645
62    0.567568
63    0.562500
64    1.264706
65    1.111111
66    0.800000
67    0.645161
68    1.500000
69    2.666667
70    0.642857
71    1.000000
74    1.000000
76    2.000000
77    0.000000
Name: cp, dtype: float64
```




```
sns.boxplot(x='age', y='trestbps', data=df)
plt.title('Blood Pressure Levels by Age')
plt.show()
```




```
sns.pairplot(df, hue='cp')
plt.show()
```



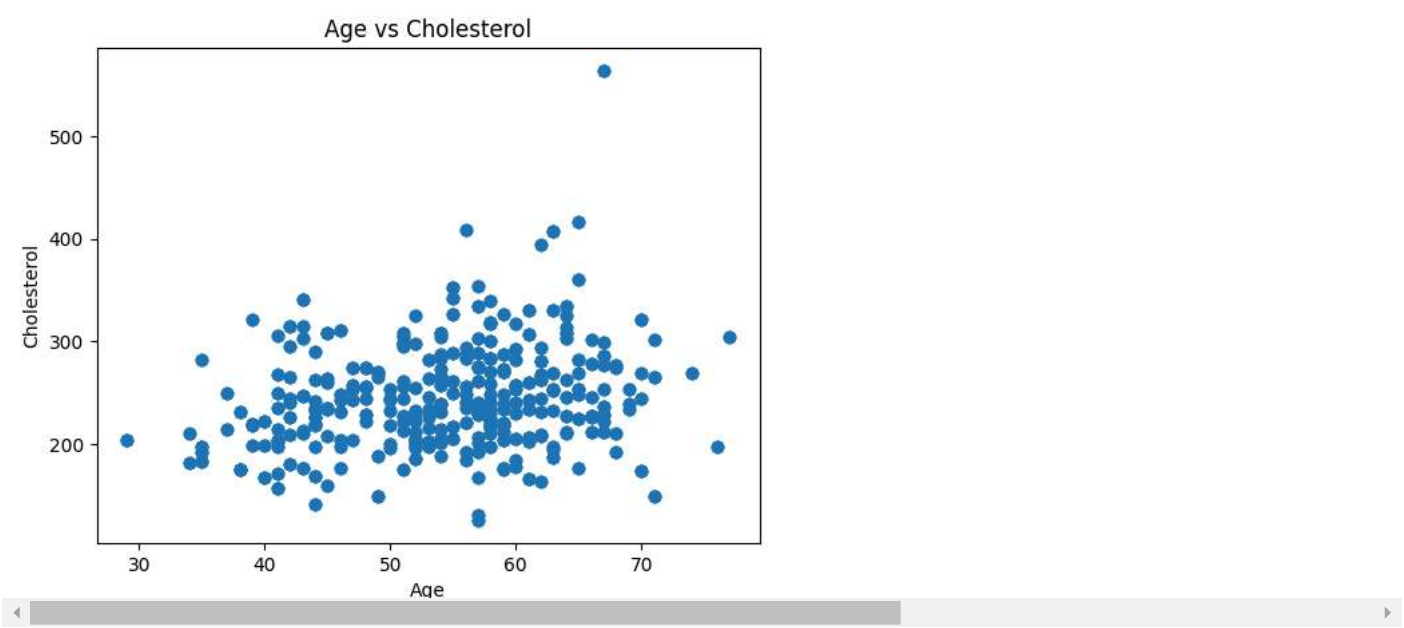

```
common_combinations = df[['cp','trestbps','chol','fbs','restecg','thalach','exang','oldpeak','slope','ca','thal']].value_counts()
print("Most common risk factor combinations:")
```

```
print(common_combinations)
```

```
Most common risk factor combinations:
cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal
2   138       175   0    1        173      0      0.0      2      4   2       8
0   100       234   0    1        156      0      0.1      2      1   3       4
    150       225   0    0        114      0      1.0      1      3   3       4
              407   0    0        154      0      4.0      1      3   3       4
    152       223   0    1        181      0      0.0      2      0   3       4
                                                                            ..
    120       267   0    1        99       1      1.8      1      2   3       3
1   126       306   0    1        163      0      0.0      2      0   2       3
    128       205   1    1        184      0      0.0      2      0   2       3
              208   1    0        140      0      0.0      2      0   2       3
3   178       270   0    0        145      0      4.2      0      0   3       3
Name: count, Length: 302, dtype: int64
```

Visualization

```
plt.scatter(df['age'], df['chol'])
plt.xlabel('Age')
plt.ylabel('Cholesterol')
plt.title('Age vs Cholesterol')
plt.show()
```




```
sns.heatmap(df.corr(), annot=True)
plt.title('Correlation Heatmap')
plt.show()
```