```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [3]: df=pd.read_csv(r"C:\\Users\\prach\\Downloads\\Heart Disease data.csv")
```

```
In [4]: df
```

Out[4]:

	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

```
In [5]: import seaborn as sns
```

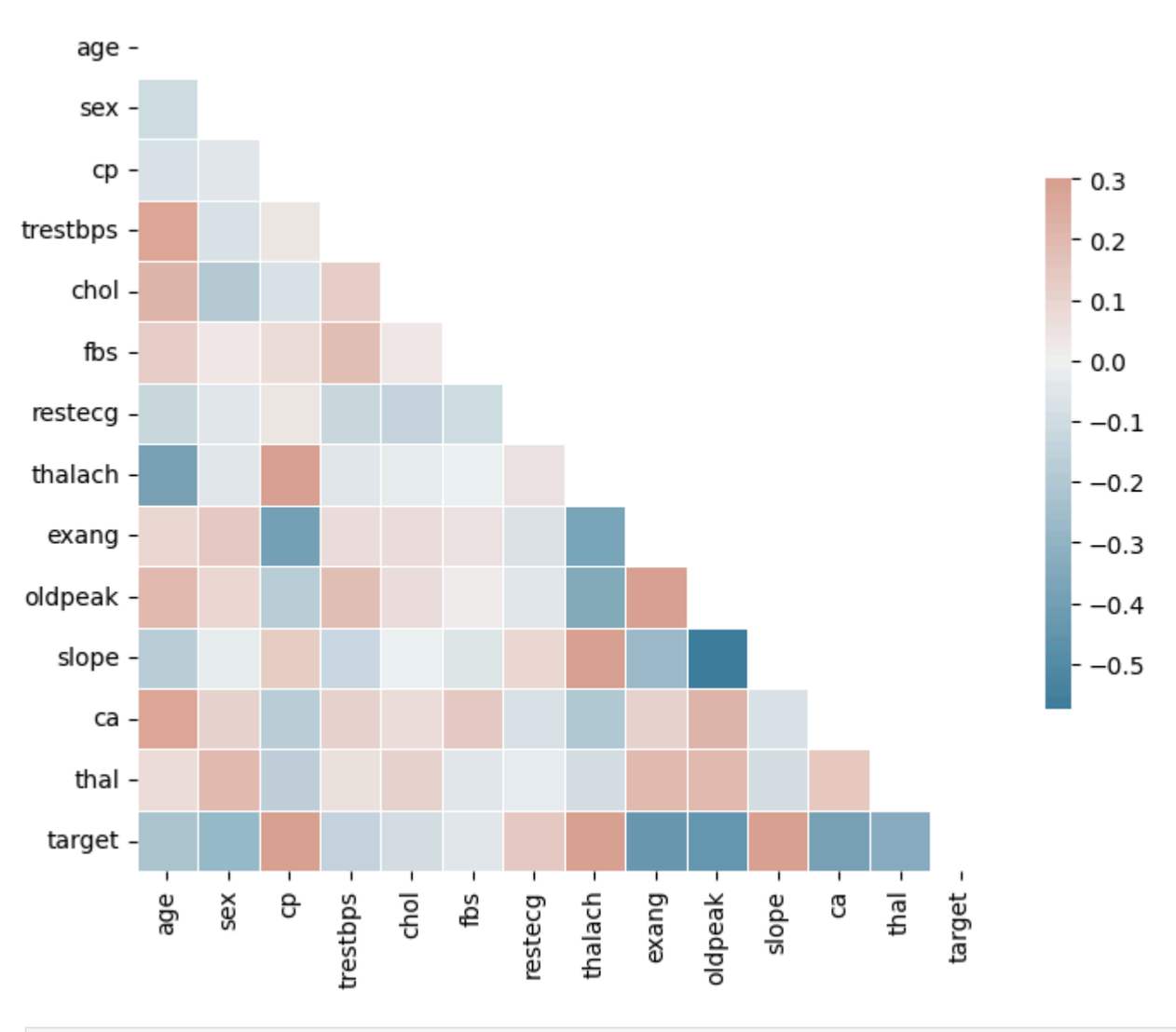
```
In [6]: print(df.isna().any())
```

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

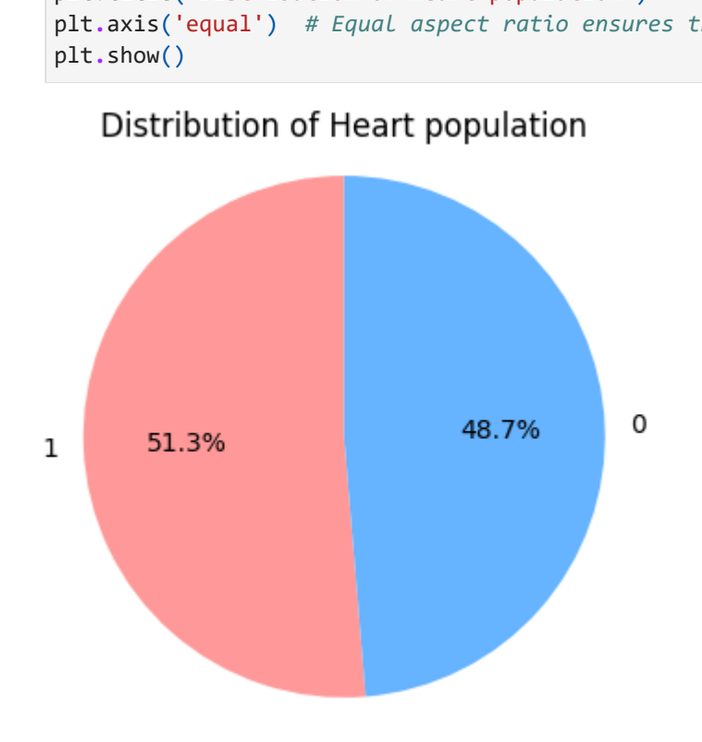
```
In [7]: print(df.info())
```

```
<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
None
```

```
In [10]: corr_matrix = df.corr()
         mask = np.triu(np.ones_like(corr_matrix, dtype=bool))
         f, ax = plt.subplots(figsize=(8, 8))
         cmap = sns.diverging_palette(230, 20, as_cmap=True)
         sns.heatmap(corr_matrix, mask=mask, cmap=cmap, vmax=.3, center=0,
                     square=True, linewidths=.5, cbar_kws={"shrink": .5})

         plt.show()
```




```
In [13]: target_counts = df['target'].value_counts()

         # Plot pie chart
         plt.figure(figsize=(4, 4))
         plt.pie(target_counts, labels=target_counts.index, autopct='%1.1f%%', startangle=90, colors=['#ff9999','#66b3ff'])
         plt.title('Distribution of Heart population')
         plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
         plt.show()
```



So out of 1025, 526 are having Heart Disease where as 499 not having heart disease

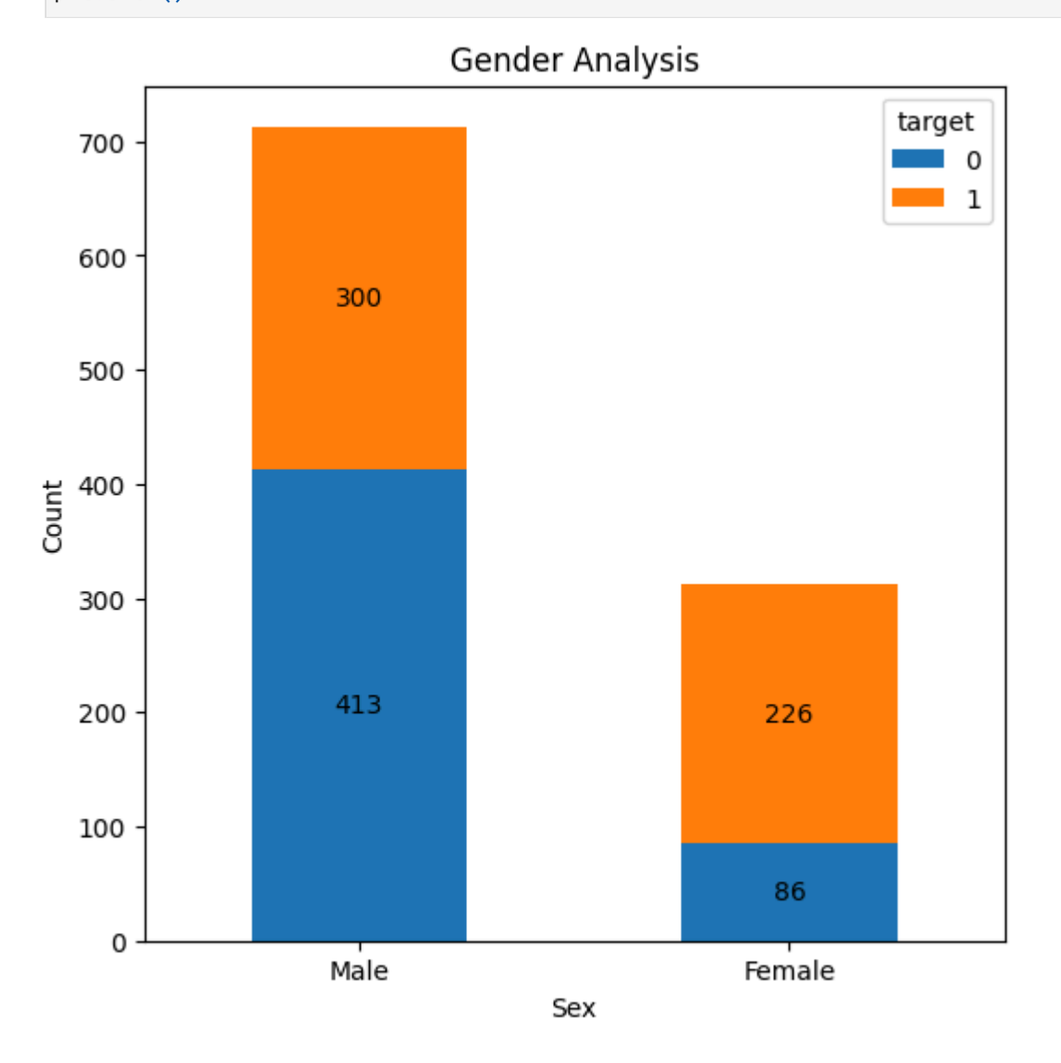
```
In [14]: df['sex'] = pd.Categorical(df['sex'], categories=[1, 0], ordered=True)
         df['sex'] = df['sex'].cat.rename_categories(['Male', 'Female'])
```

```
In [15]: df.head(3)
```

Out[15]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	Male	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	Male	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	Male	0	145	174	0	1	125	1	2.6	0	0	3	0

```
In [20]: fig, ax = plt.subplots(figsize=(6, 6))
         ct = pd.crosstab(df['sex'], df['target'])
         ct.plot(kind='bar', stacked=True, ax=ax)
         for container in ax.containers:
             ax.bar_label(container, label_type='center')
         ax.set_title('Gender Analysis')
         ax.set_xlabel('Sex')
         ax.set_ylabel('Count')

         plt.xticks(rotation=0)
         plt.show()
```



Out of 713 males 300 having heart disease where as out of 312 females 226 females have heart disease. A higher proportion of females (72.44%) have heart disease compared to males (42.08%). This suggests that, in this sample, females are more likely to have heart disease than males, despite there being fewer females in the total sample.

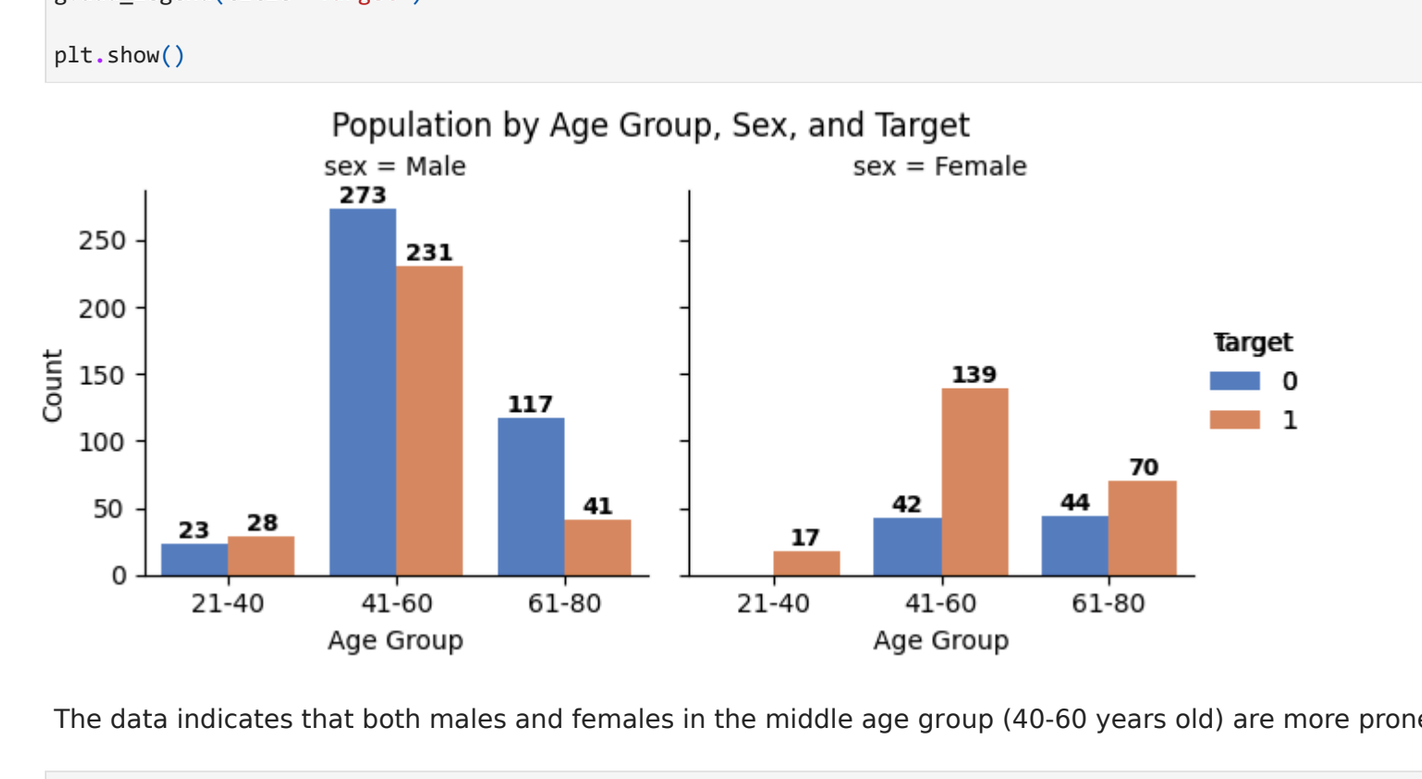
```
In [30]: bins = [20, 40, 60, 80]
         labels = ['21-40', '41-60', '61-80']
         df['Age Group'] = pd.cut(df['age'], bins=bins, labels=labels, right=True)
```

```
In [31]: df.head(3)
```

Out[31]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Age Group
0	52	Male	0	125	212	0	1	168	0	1.0	2	2	3	0	41-60
1	53	Male	0	140	203	1	0	155	1	3.1	0	0	3	0	41-60
2	70	Male	0	145	174	0	1	125	1	2.6	0	0	3	0	61-80

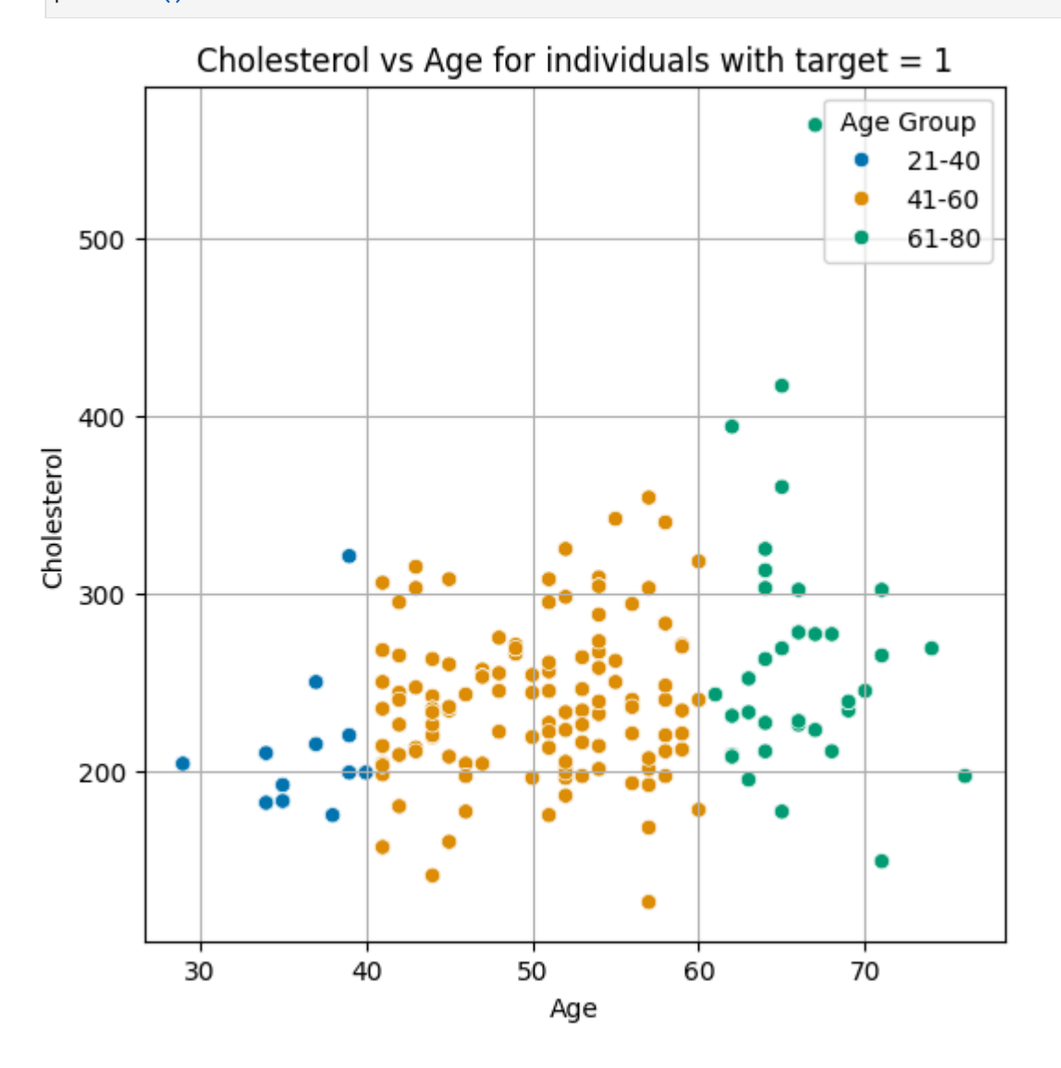
```
In [32]: g = sns.catplot(
             data=df,
             x='Age Group',
             hue='target',
             col='sex',
             kind='count',
             height=3,
             aspect=1,
             dodge=True,
             palette='muted'
         )
         for ax in g.axes.flat:
             for c in ax.containers:
                 labels = [int(v.get_height()) if v.get_height() > 0 else '' for v in c]
                 ax.bar_label(c, labels=labels, label_type='edge', fontsize=9, fontweight='bold')
         g.set_axis_labels("Age Group", "Count")
         g.fig.suptitle("Population by Age Group, Sex, and Target", y=1.03)
         g.add_legend(title="Target")

         plt.show()
```



The data indicates that both males and females in the middle age group (40-60 years old) are more prone to heart disease compared to other age groups. May require more targeted preventive measures and screenings.

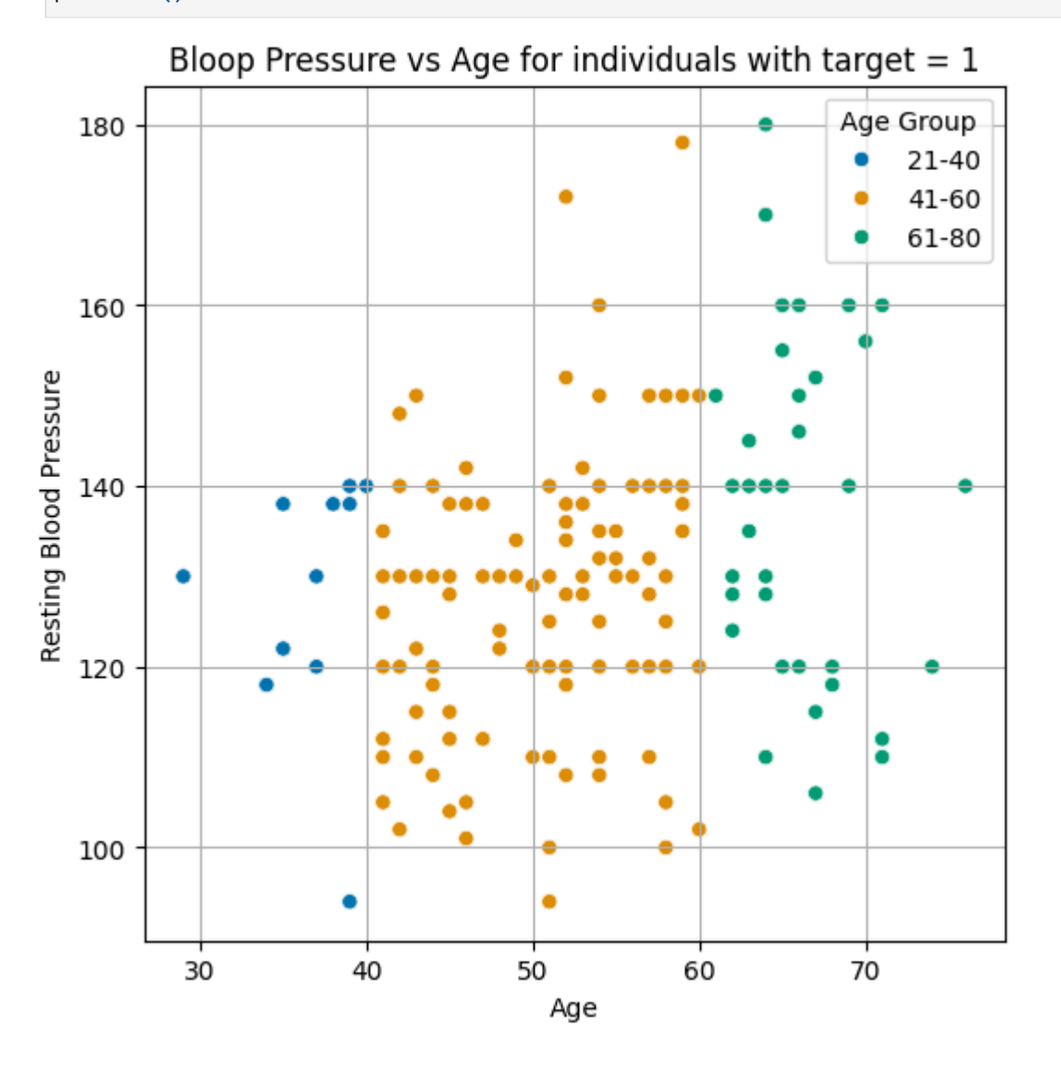
```
In [33]: df_target_1 = df[df['target'] == 1]
```

```
In [37]: plt.figure(figsize=(6, 6))
         sns.scatterplot(data=df_target_1, x='age', y='chol', hue='Age Group', palette='colorblind')
         plt.title('Cholesterol vs Age for individuals with target = 1')
         plt.xlabel('Age')
         plt.ylabel('Cholesterol')
         plt.grid(True)
         plt.legend(title='Age Group')
         plt.show()
```



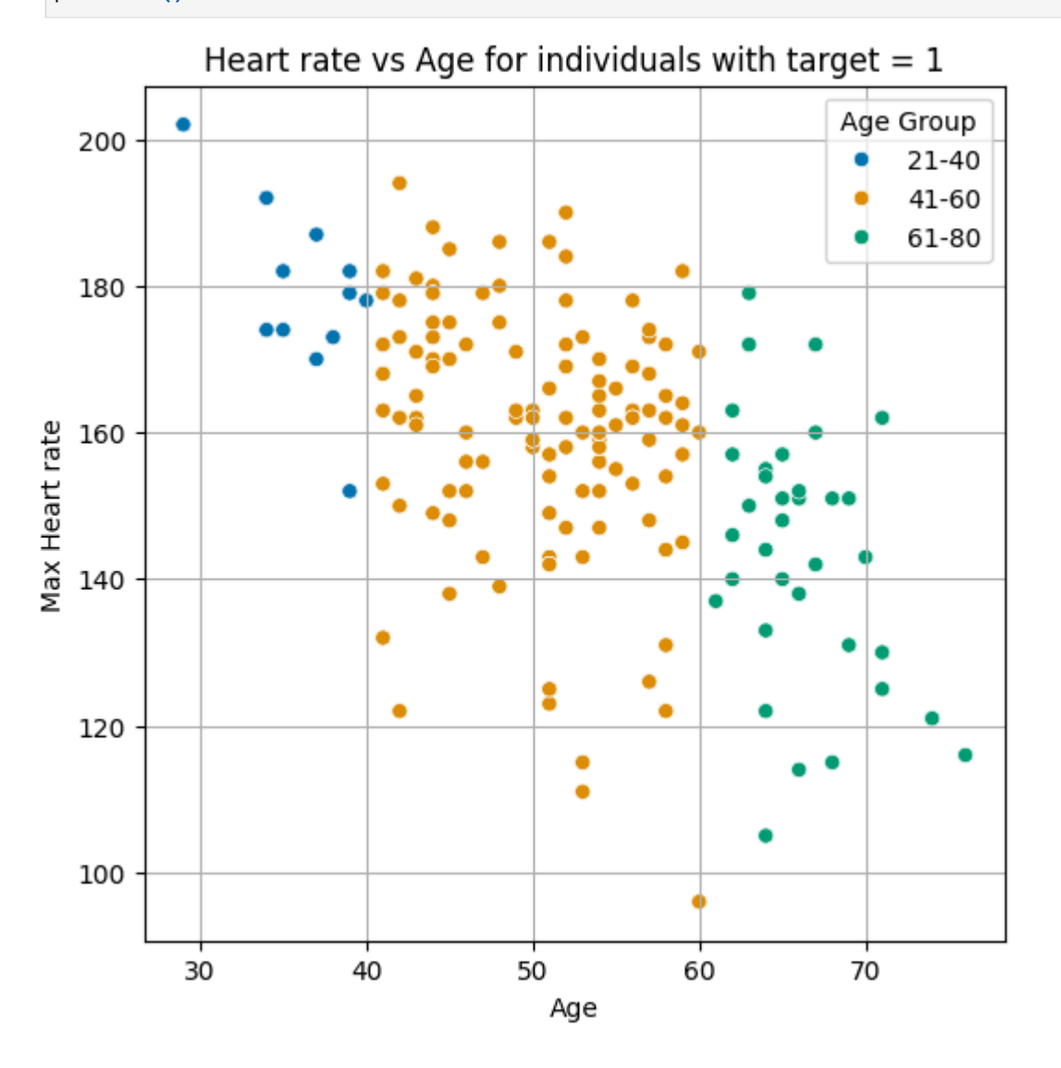
The data indicates that elevated cholesterol levels (greater than 200 mg/dL) are a significant risk factor for heart disease across all age groups. This aligns with existing medical knowledge that high cholesterol contributes to the buildup of plaques in the arteries, increasing the risk of cardiovascular events.

```
In [38]: plt.figure(figsize=(6, 6))
         sns.scatterplot(data=df_target_1, x='age', y='trestbps', hue='Age Group', palette='colorblind')
         plt.title('Bloop Pressure vs Age for individuals with target = 1')
         plt.xlabel('Age')
         plt.ylabel('Resting Blood Pressure')
         plt.grid(True)
         plt.legend(title='Age Group')
         plt.show()
```



This suggests a strong association between elevated resting blood pressure and the presence of heart disease across different age groups. Elevated resting blood pressure could be a significant risk factor or indicator for heart disease.

```
In [40]: plt.figure(figsize=(6, 6))
         sns.scatterplot(data=df_target_1, x='age', y='thalach', hue='Age Group', palette='colorblind')
         plt.title('Heart rate vs Age for individuals with target = 1')
         plt.xlabel('Age')
         plt.ylabel('Max Heart rate')
         plt.grid(True)
         plt.legend(title='Age Group')
         plt.show()
```



Tachycardia is defined as a resting heart rate exceeding 100 beats per minute (bpm). If most patients with heart disease in your dataset have a heart rate over 120 bpm, it indicates a significant prevalence of tachycardia among these patients.

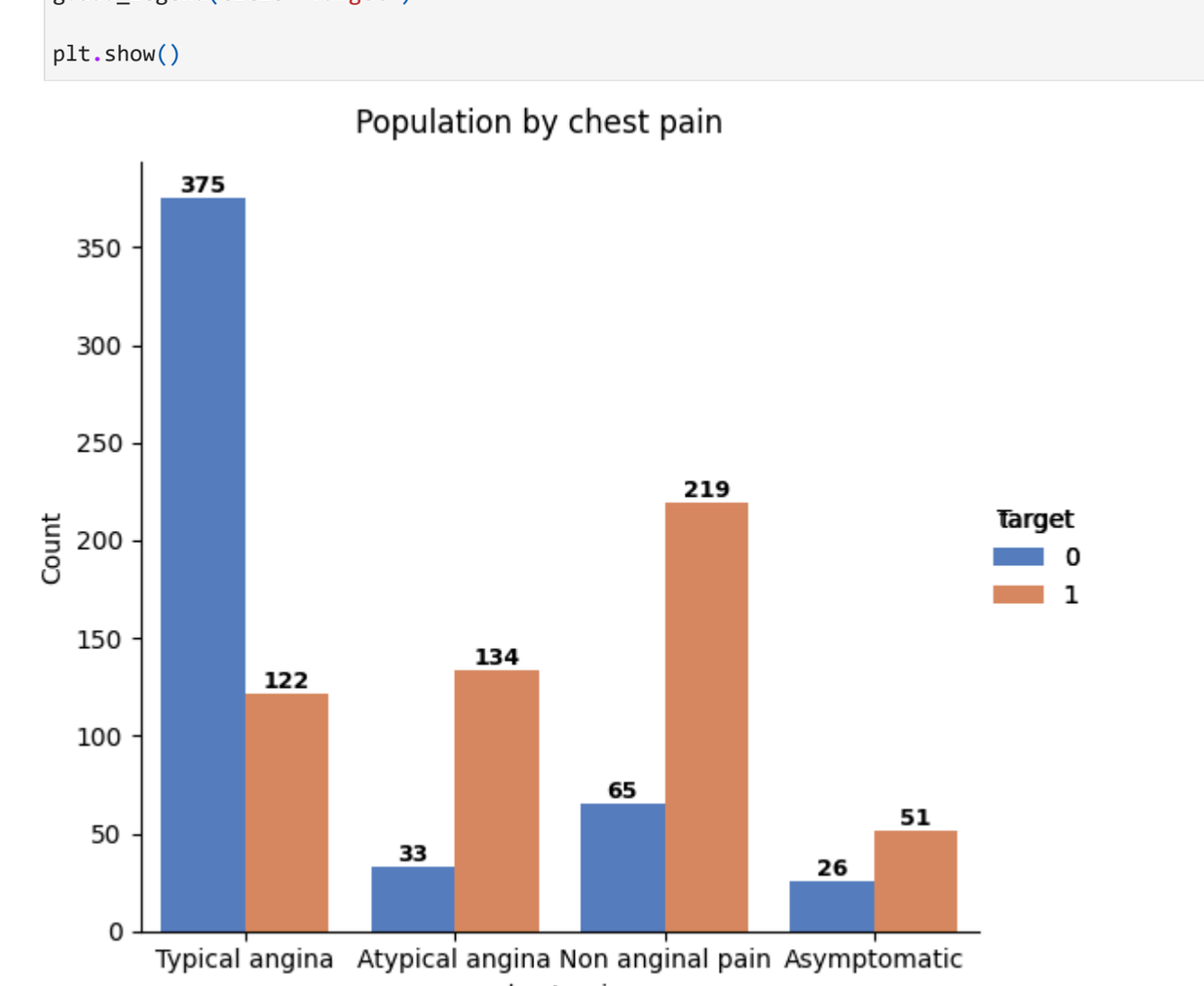
```
In [43]: df['cp'] = pd.Categorical(df['cp'], categories=[0,1,2,3], ordered=True)
         df['cp'] = df['cp'].cat.rename_categories(['Typical angina','Atypical angina','Non anginal pain','Asymptomatic'])
```

```
In [44]: df.head(3)
```

Out[44]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Age Group
0	52	Male	Typical angina	125	212	0	1	168	0	1.0	2	2	3	0	41-60
1	53	Male	Typical angina	140	203	1	0	155	1	3.1	0	0	3	0	41-60
2	70	Male	Typical angina	145	174	0	1	125	1	2.6	0	0	3	0	61-80

```
In [50]: g = sns.catplot(
             data=df,
             x='cp',
             hue='target',
             kind='count',
             height= 5,
             aspect=1,
             dodge=True,
             palette='muted'
         )
         for ax in g.axes.flat:
             for c in ax.containers:
                 labels = [int(v.get_height()) if v.get_height() > 0 else '' for v in c]
                 ax.bar_label(c, labels=labels, label_type='edge', fontsize=9, fontweight='bold')
         g.set_axis_labels("chest pain", "Count")
         g.fig.suptitle("Population by chest pain", y=1.03)
         g.add_legend(title="Target")

         plt.show()
```

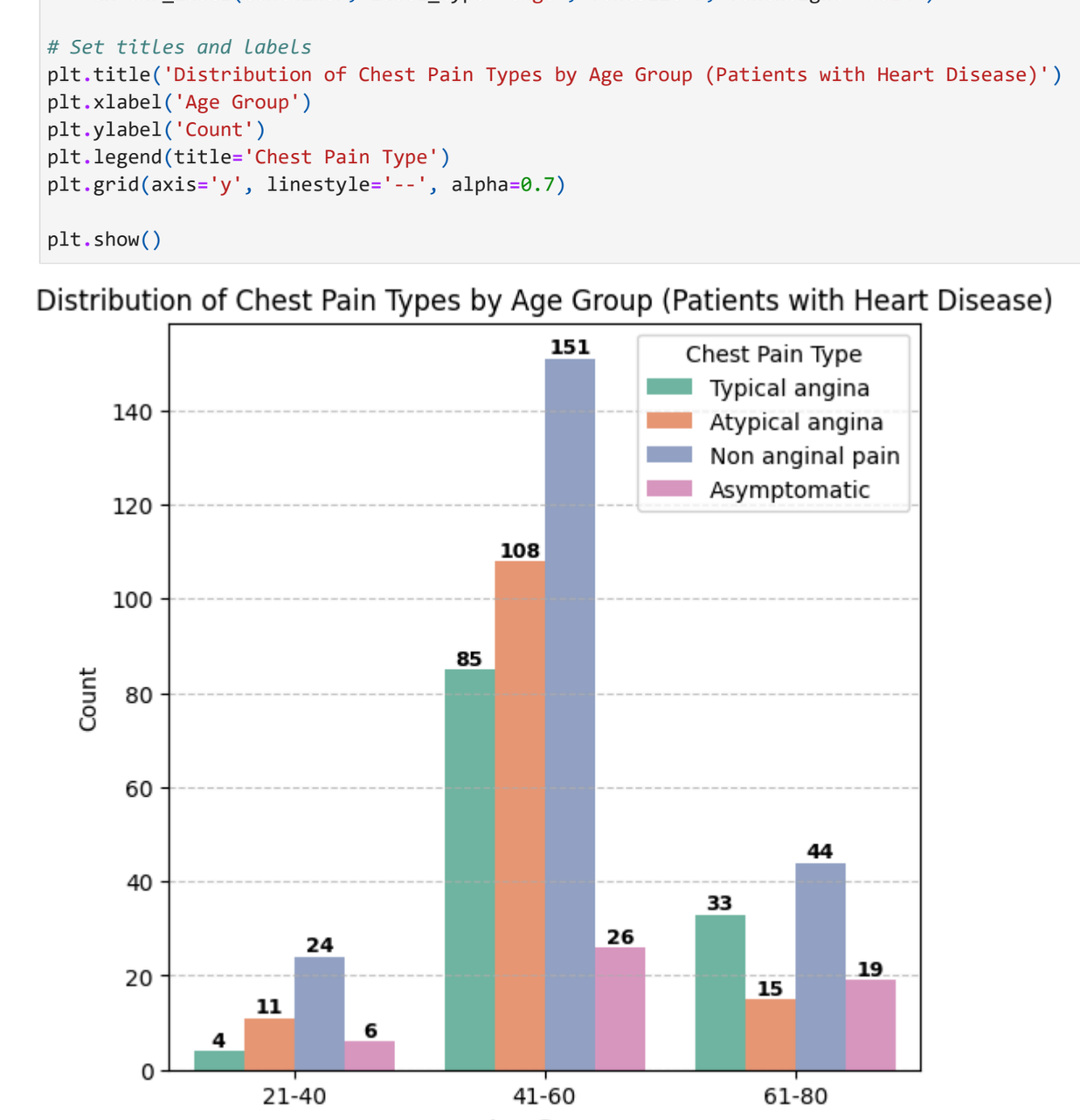


The data suggests that individuals experiencing non-anginal chest pain are more likely to develop heart disease. This finding highlights the importance of not dismissing non-anginal pain as benign. Among those with angina, patients with atypical anginal pain appear to have a higher prevalence of heart disease, which might not always present with the classic symptoms of heart disease, can sometimes be misleading. This underscores the need for comprehensive diagnostic evaluations

```
In [57]: df_target_1['cp'] = pd.Categorical(df_target_1['cp'], categories=[0,1,2,3], ordered=True)
         df_target_1['cp'] = df_target_1['cp'].cat.rename_categories(['Typical angina','Atypical angina','Non anginal pain','Asymptomatic'])
```

```
C:\Users\prach\AppData\Local\Temp\ipykernel_27660\3044753552.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_target_1['cp'] = pd.Categorical(df_target_1['cp'], categories=[0,1,2,3], ordered=True)
C:\Users\prach\AppData\Local\Temp\ipykernel_27660\3044753552.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_target_1['cp'] = df_target_1['cp'].cat.rename_categories(['Typical angina','Atypical angina','Non anginal pain','Asymptomatic'])
```

```
In [58]: plt.figure(figsize=(6, 6))

         ax = sns.countplot(data=df_target_1, x='Age Group', hue='cp', palette='Set2')

         # Adding count labels on the bars
         for container in ax.containers:
             ax.bar_label(container, label_type='edge', fontsize=9, fontweight='bold')

         # Set titles and labels
         plt.title('Distribution of Chest Pain Types by Age Group (Patients with Heart Disease)')
         plt.xlabel('Age Group')
         plt.ylabel('Count')
         plt.legend(title='Chest Pain Type')
         plt.grid(axis='y', linestyle='--', alpha=0.7)

         plt.show()
```



The high presence of non-angina heart disease across all age groups suggests that non-anginal heart disease is independent of age. Implying that similar preventive measures and treatments might be applicable across different age groups for non-anginal heart disease. In senior age group Typical angina is more prevalent. This form of angina is usually characterized by classic symptoms such as chest pain or discomfort, which are often triggered by exertion or emotional stress. The presence of typical angina in seniors makes diagnosis based on symptomatic presentation more straightforward

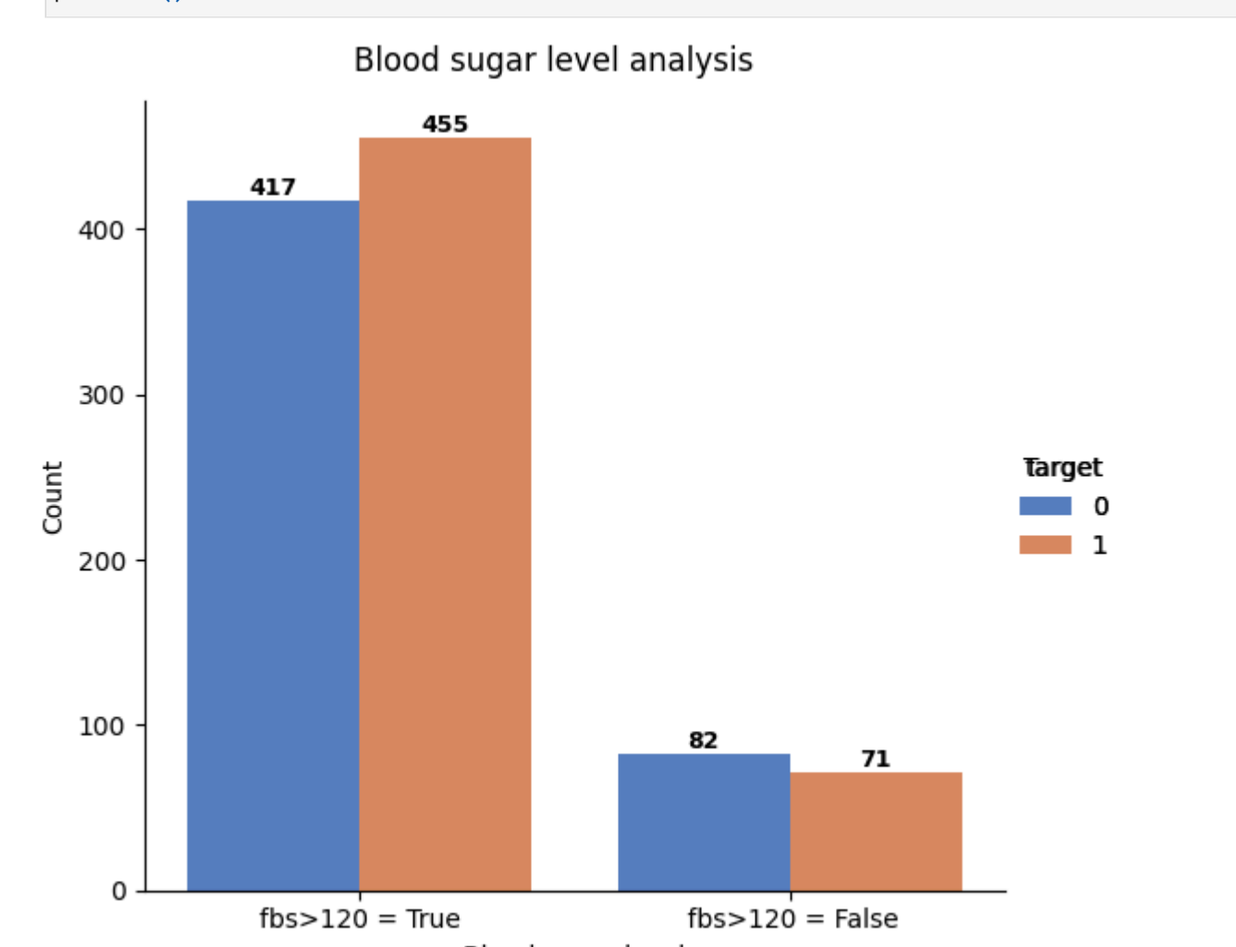
```
In [59]: df['fbs'] = pd.Categorical(df['fbs'], categories=[0,1], ordered=True)
         df['fbs'] = df['fbs'].cat.rename_categories(['fbs>120 = True',' fbs>120 = False'])
```

```
In [60]: df.head(3)
```

Out[60]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Age Group
0	52	Male	Typical angina	125	212	fbs>120 = True	1	168	0	1.0	2	2	3	0	41-60
1	53	Male	Typical angina	140	203	fbs>120 = False	0	155	1	3.1	0	0	3	0	41-60
2	70	Male	Typical angina	145	174	fbs>120 = True	1	125	1	2.6	0	0	3	0	61-80

```
In [61]: g = sns.catplot(
             data=df,
             x='fbs',
             hue='target',
             kind='count',
             height= 5,
             aspect=1,
             dodge=True,
             palette='muted'
         )
         for ax in g.axes.flat:
             for c in ax.containers:
                 labels = [int(v.get_height()) if v.get_height() > 0 else '' for v in c]
                 ax.bar_label(c, labels=labels, label_type='edge', fontsize=9, fontweight='bold')
         g.set_axis_labels("Blood sugar level", "Count")
         g.fig.suptitle("Blood sugar level analysis", y=1.03)
         g.add_legend(title="Target")

         plt.show()
```

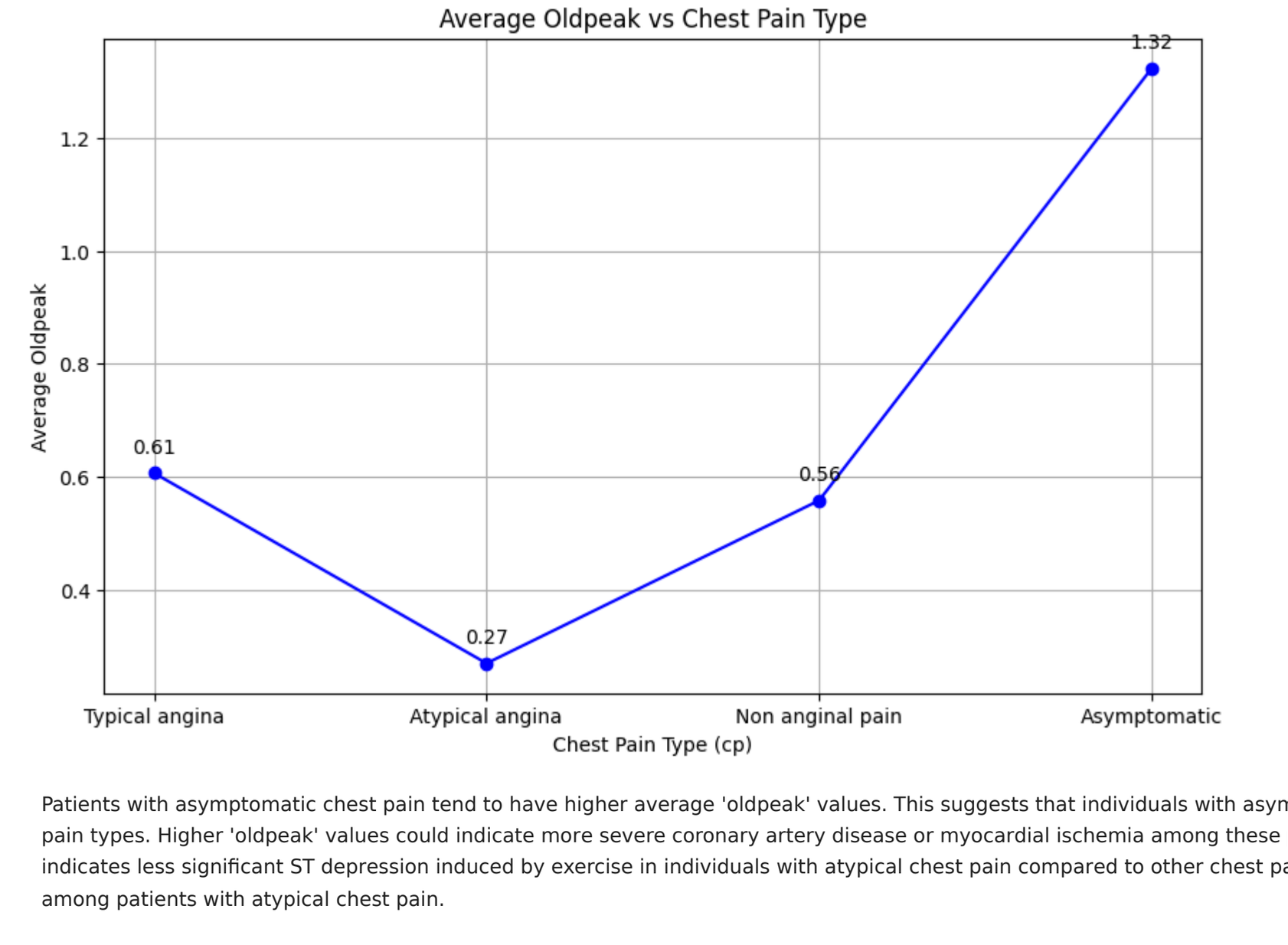


Only about 13.5% (71 out of 526) of the individuals have a fasting blood sugar level above 120 mg/dL , indicating that high fasting blood sugar levels are relatively uncommon in this population

```
In [62]: summary = df_target_1.groupby('cp')['oldpeak'].mean().reset_index()

         print(summary)
```

```
                 cp   oldpeak
0    Typical angina  0.606557
1   Atypical angina  0.269403
2  Non anginal pain  0.557991
3      Asymptomatic  1.323529
```

```
In [63]: plt.figure(figsize=(10, 6))
         plt.plot(summary['cp'], summary['oldpeak'], marker='o', color='blue', linestyle='-')

         # Add labels with y-coordinates on each point
         for i, txt in enumerate(summary['oldpeak']):
             plt.annotate(f'{txt:.2f}', (summary['cp'][i], summary['oldpeak'][i]), textcoords="offset points", xytext=(0,10), ha='center')

         # Set titles and labels
         plt.title('Average Oldpeak vs Chest Pain Type')
         plt.xlabel('Chest Pain Type (cp)')
         plt.ylabel('Average Oldpeak')
         plt.grid(True)

         plt.show()
```



Patients with asymptomatic chest pain tend to have higher average 'oldpeak' values. This suggests that individuals with asymptomatic chest pain may experience more significant ST depression induced by exercise relative to other chest pain types. Higher 'oldpeak' values could indicate more severe coronary artery disease or myocardial ischemia among these patients.On the other hand, patients with atypical chest pain exhibit the lowest average 'oldpeak' values. This indicates less significant ST depression induced by exercise in individuals with atypical chest pain compared to other chest pain types. Lower 'oldpeak' values may suggest relatively milder coronary artery disease or myocardial ischemia among patients with atypical chest pain.