

HEART DISEASE PREDICTION

```
In [1]: ▶ import numpy as np # Linear algebra
import pandas as pd
```

```
In [2]: ▶ import os
for dirname, _, filenames in os.walk(r'C:\Users\yogay\OneDrive\Desktop\Yogita_Yadav\Data Science\17th\EDA'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
```

C:\Users\yogay\OneDrive\Desktop\Yogita_Yadav\Data Science\17th\EDA\extensive-analysis-visualization-with-python.ipynb
C:\Users\yogay\OneDrive\Desktop\Yogita_Yadav\Data Science\17th\EDA\heart.csv

```
In [3]: ▶ import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st
%matplotlib inline

sns.set(style="whitegrid")
```

```
In [4]: ▶ import warnings
warnings.filterwarnings('ignore')
```

```
In [5]: ▶ df = pd.read_csv(r'C:\Users\yogay\OneDrive\Desktop\Yogita_Yadav\Data Science\17th\EDA\heart.csv')
```

In [6]: `df`

Out[6]:

	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
...
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 × 14 columns

In [7]: `print('The shape of the dataset : ', df.shape)`

The shape of the dataset : (303, 14)

In [8]: `df.head()`

Out[8]:

	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 [9]: `df.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: 33.3 KB
```

```
In [10]: df.dtypes
```

```
Out[10]: age          int64
sex            int64
cp             int64
trestbps       int64
chol           int64
fbs            int64
restecg        int64
thalach        int64
exang          int64
oldpeak        float64
slope          int64
ca             int64
thal           int64
target         int64
dtype: object
```

```
In [11]: df.describe()
```

```
Out[11]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000

```
In [12]: df.columns
```

```
Out[12]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',  
              'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],  
              dtype='object')
```

Univariate Analysis

```
In [13]: df['target'].nunique()
```

```
Out[13]: 2
```

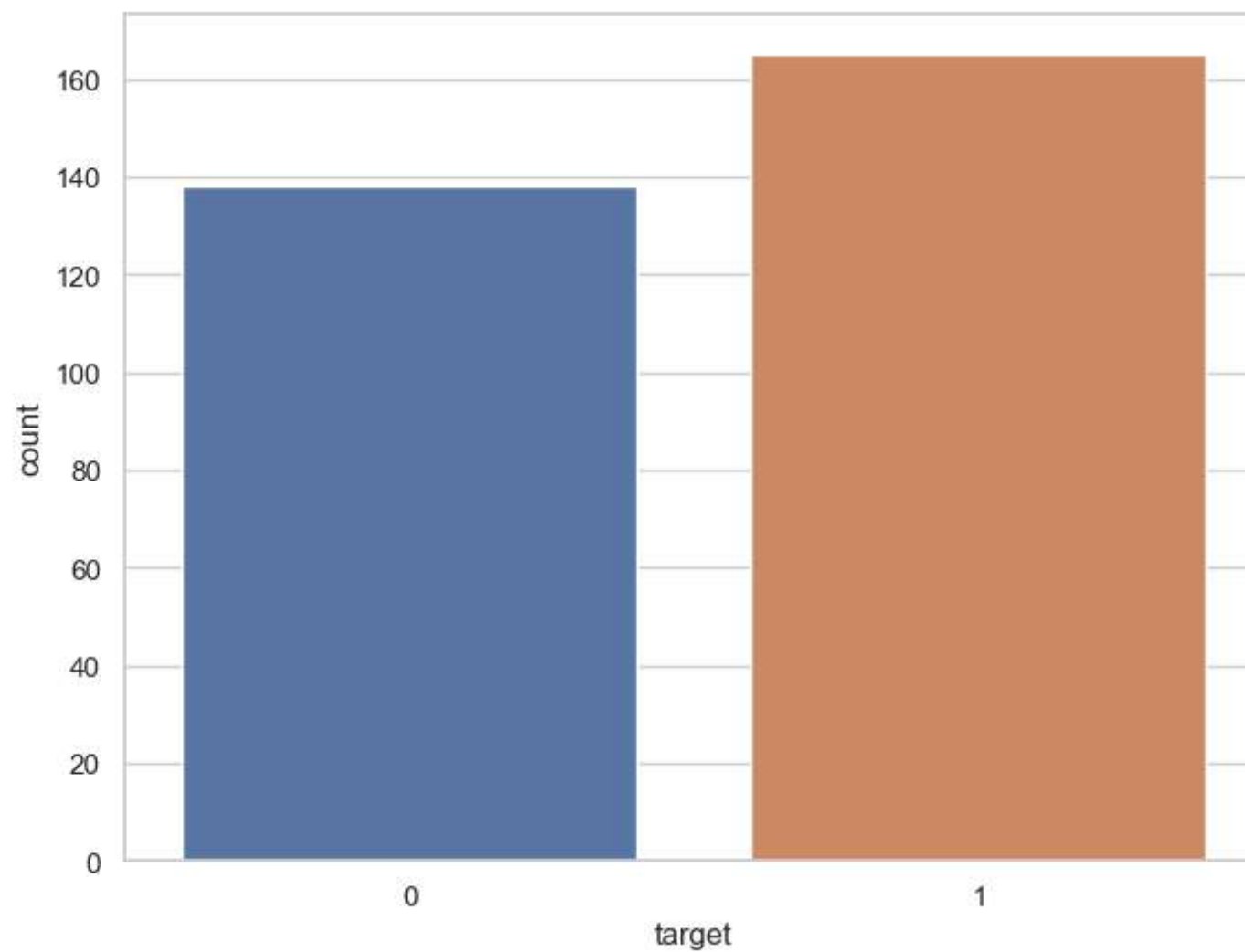
```
In [14]: df['target'].unique()
```

```
Out[14]: array([1, 0], dtype=int64)
```

```
In [15]: df['target'].value_counts()
```

```
Out[15]: 1    165  
         0    138  
         Name: target, dtype: int64
```

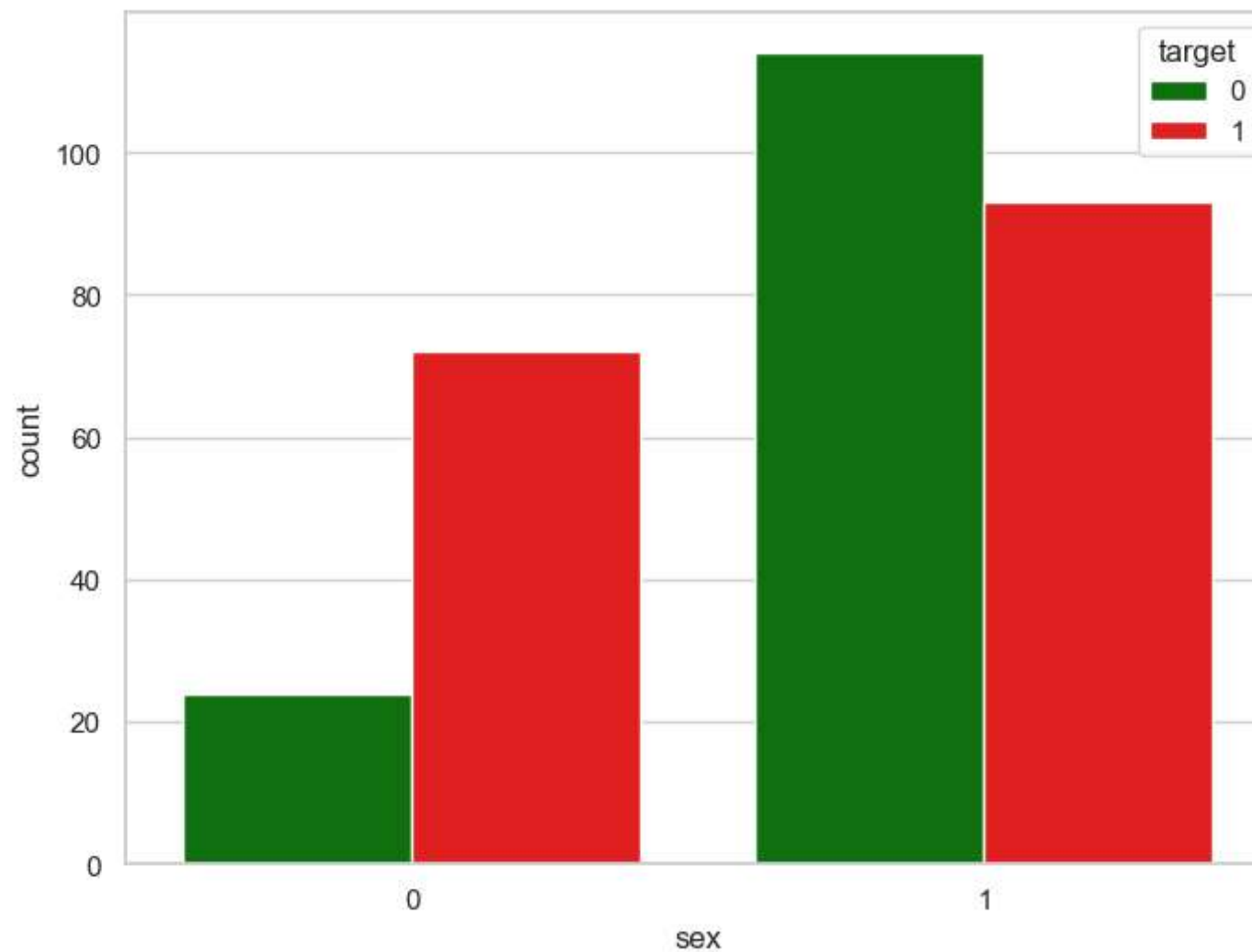
```
In [16]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="target", data=df)  
plt.show()
```



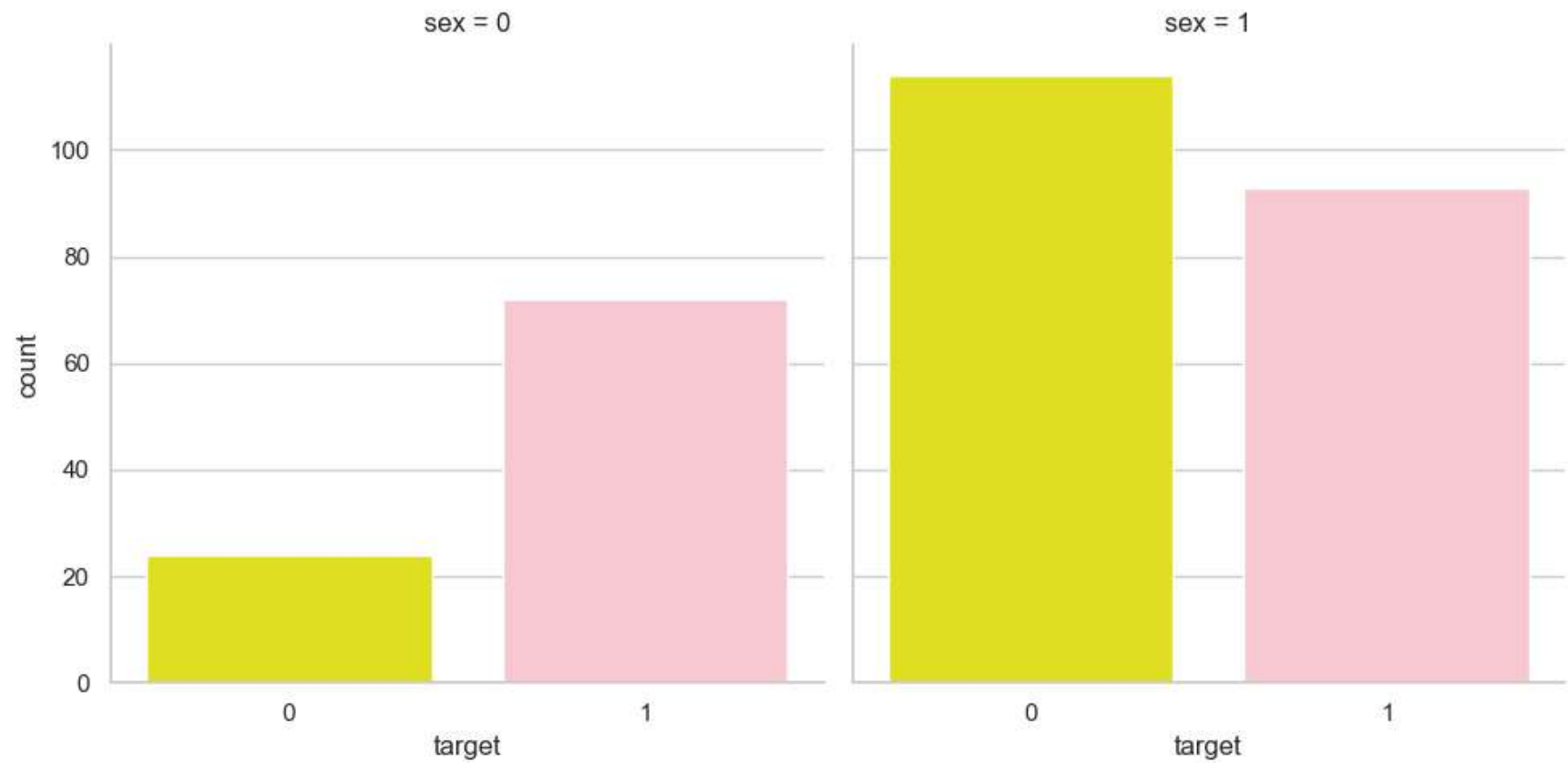
```
In [17]: df.groupby('sex')['target'].value_counts()
```

```
Out[17]: sex  target
0      1      72
        0      24
1      0     114
        1      93
Name: target, dtype: int64
```

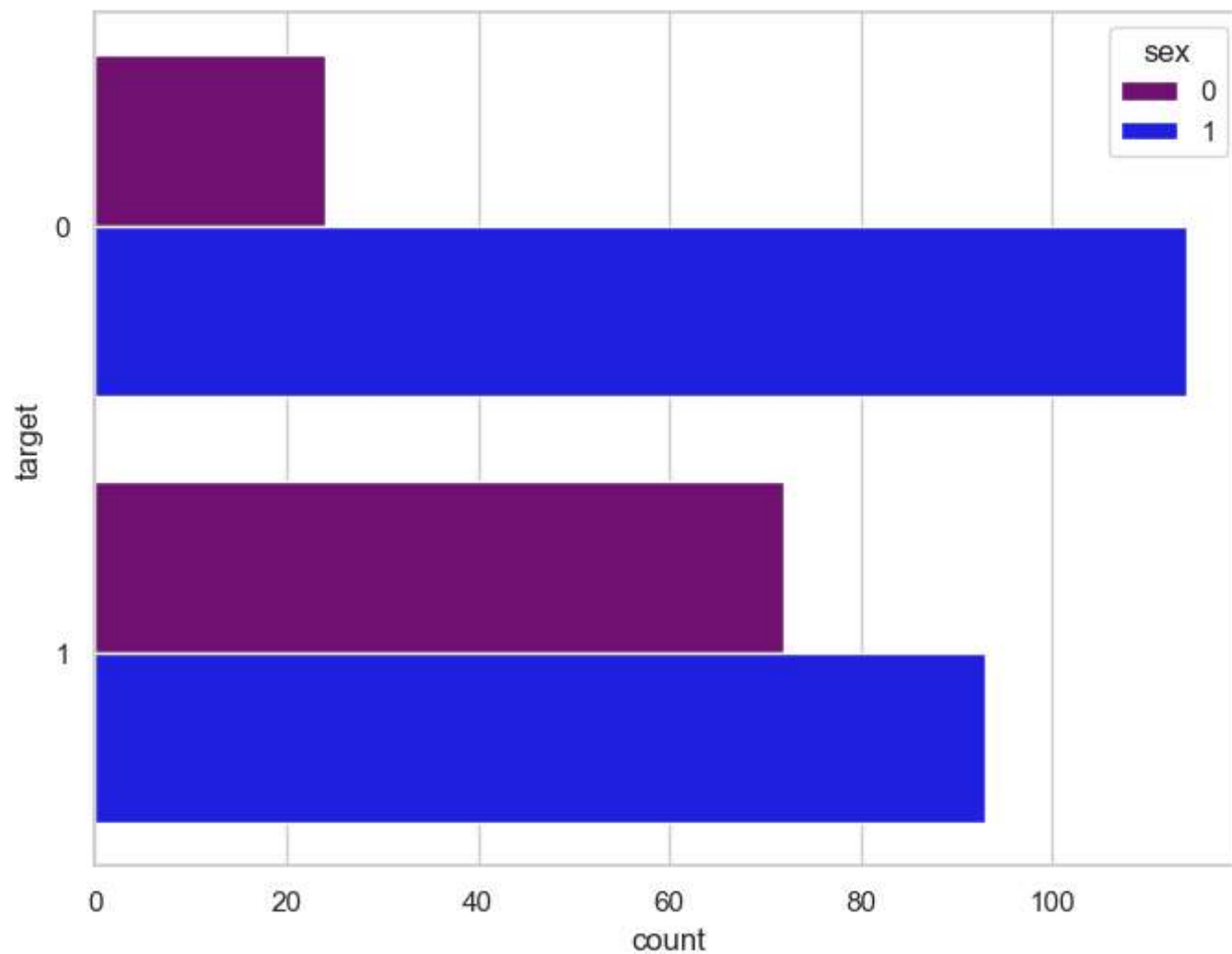
```
In [18]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="sex", hue="target", data=df, palette={'red', 'green'})  
plt.show()
```



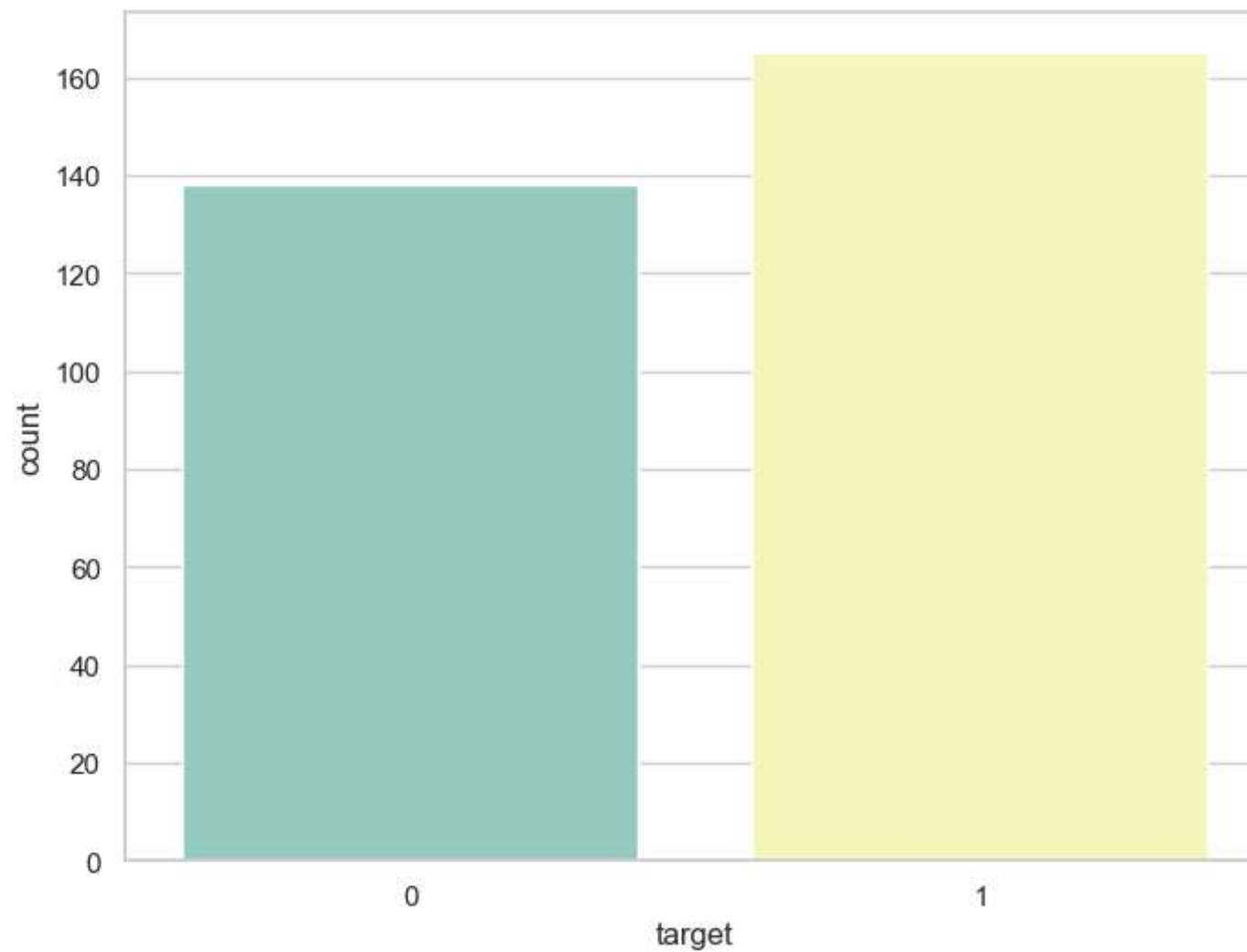

```
In [19]: ax = sns.catplot(x="target", col="sex", data=df, kind="count", height=5, aspect=1,palette={'pink','yellow'})
```



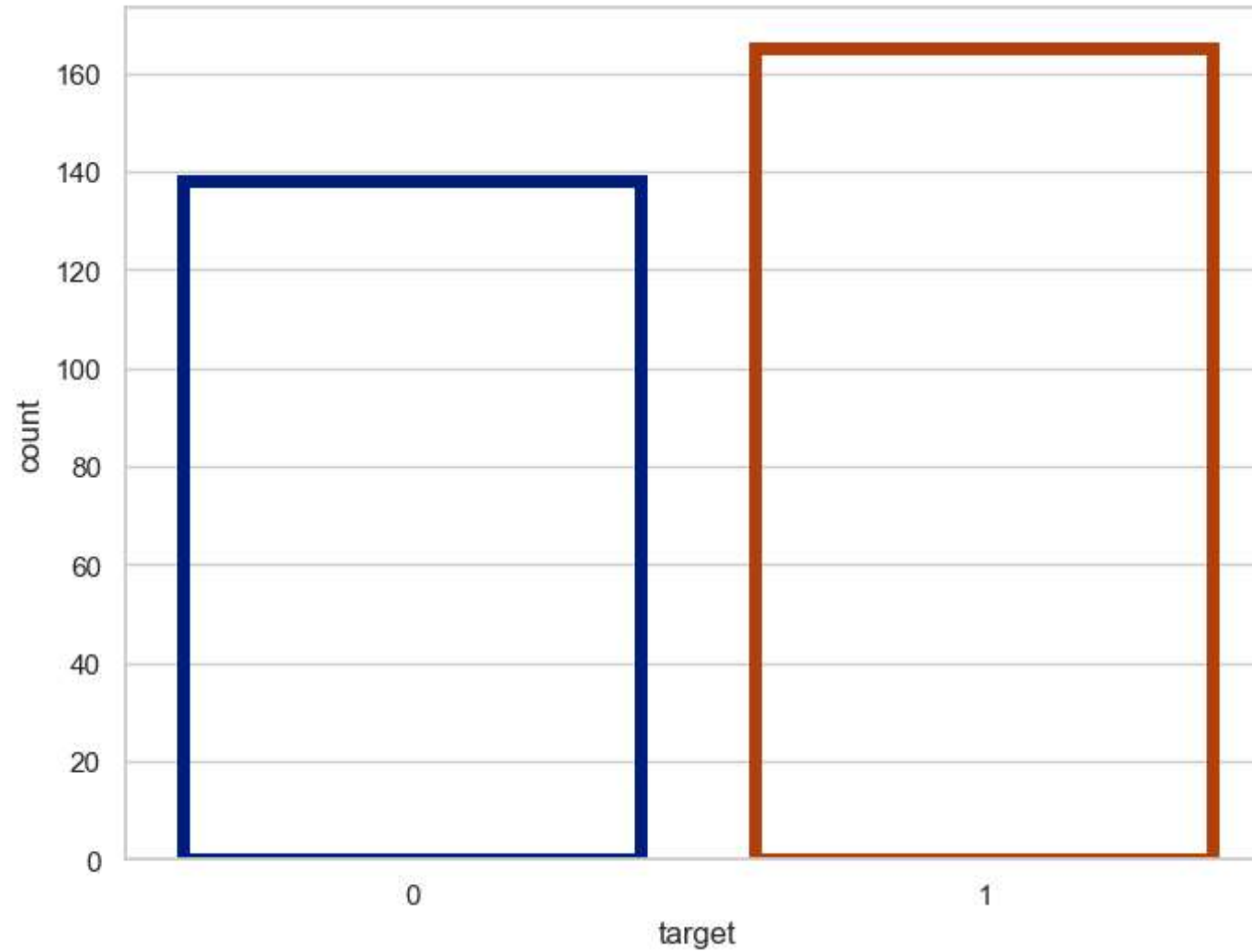
```
In [20]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(y="target", hue="sex", data=df, palette={'blue', 'purple'})  
plt.show()
```



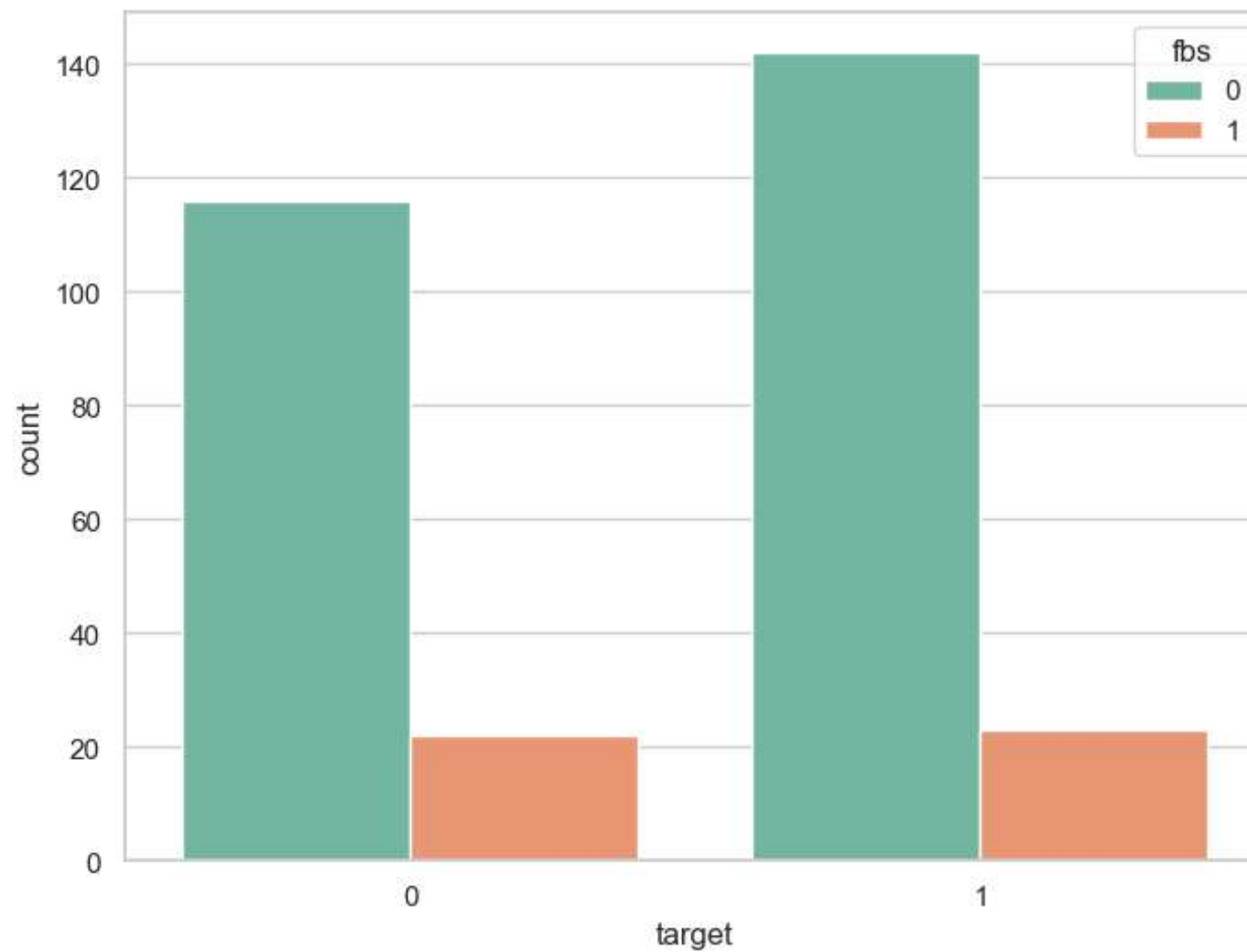
```
In [21]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="target", data=df, palette="Set3")  
plt.show()
```



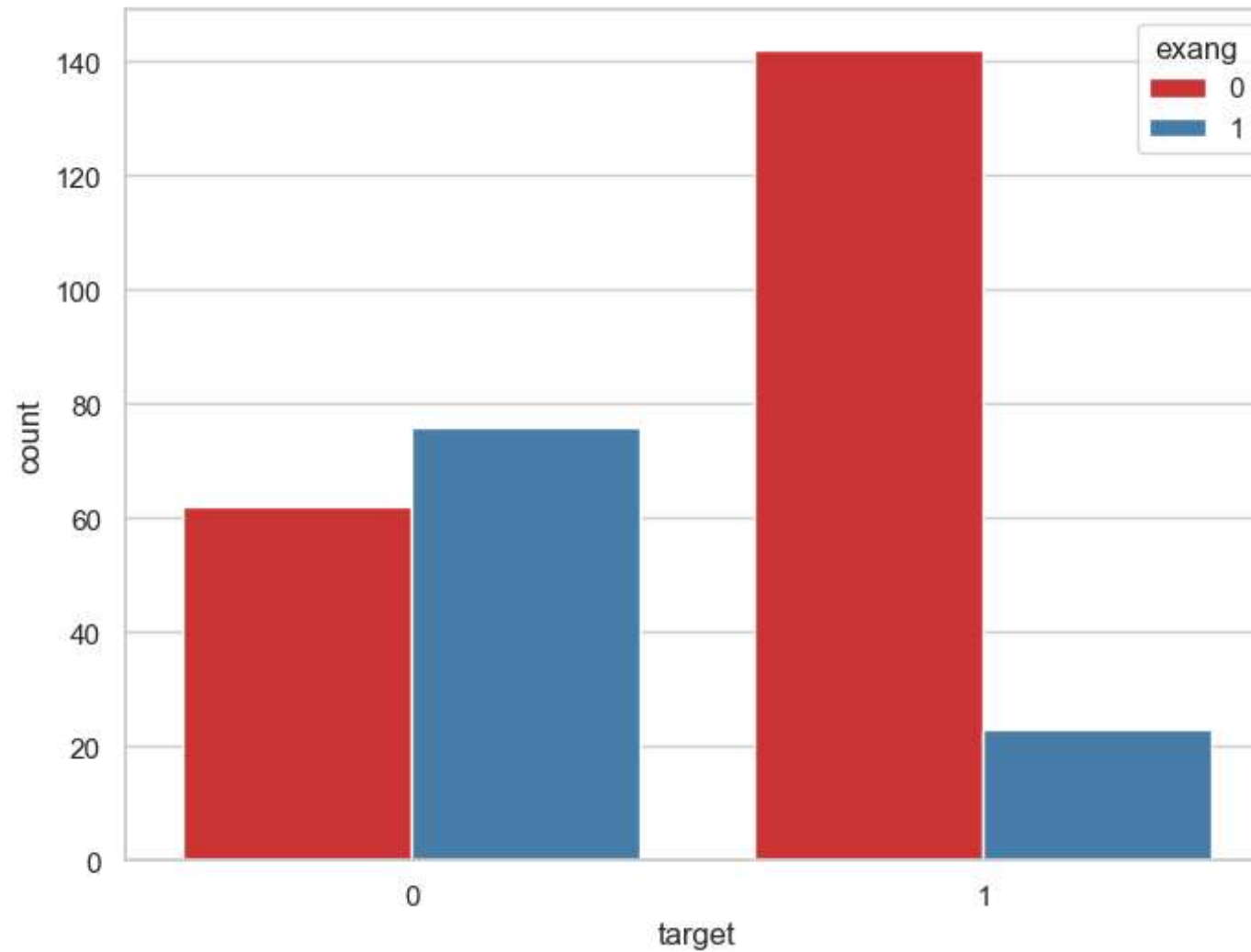
```
In [22]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="target", data=df, facecolor=(0, 0, 0, 0), linewidth=5, edgecolor=sns.color_palette("dark",  
plt.show())
```



```
In [24]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="target", hue="fbs", data=df,palette="Set2")  
plt.show()
```



```
In [25]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="target", hue="exang", data=df, palette="Set1")  
plt.show()
```



Bivariate Analysis

```
In [26]: correlation = df.corr()
```

```
In [27]: correlation['target'].sort_values(ascending=False)
```

```
Out[27]: target      1.000000  
cp          0.433798  
thalach     0.421741  
slope       0.345877  
restecg     0.137230  
fbs         -0.028046  
chol        -0.085239  
trestbps    -0.144931  
age         -0.225439  
sex         -0.280937  
thal        -0.344029  
ca          -0.391724  
oldpeak     -0.430696  
exang       -0.436757  
Name: target, dtype: float64
```

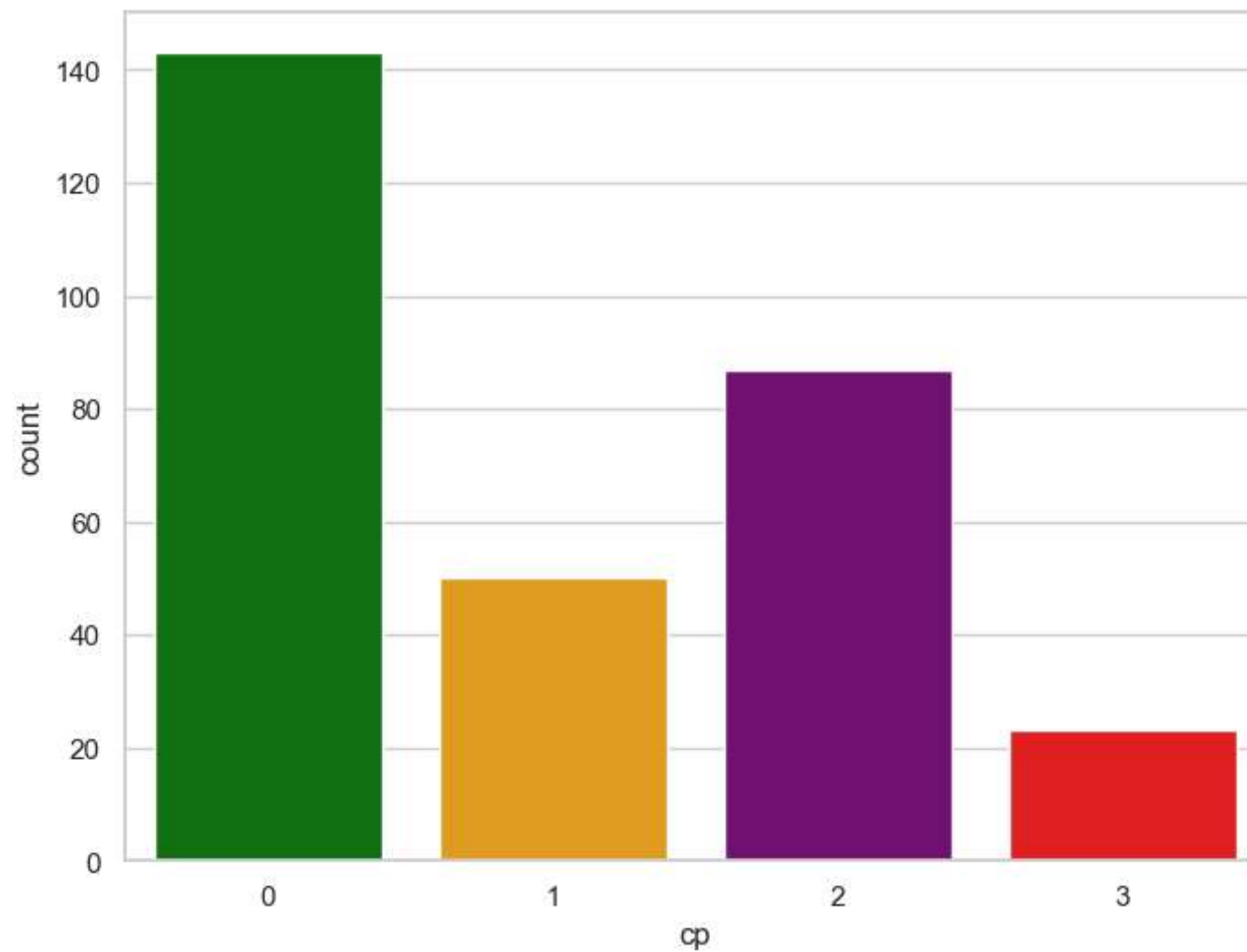
```
In [28]: df['cp'].nunique()
```

```
Out[28]: 4
```

```
In [29]: df['cp'].value_counts()
```

```
Out[29]: 0      143  
2       87  
1       50  
3       23  
Name: cp, dtype: int64
```

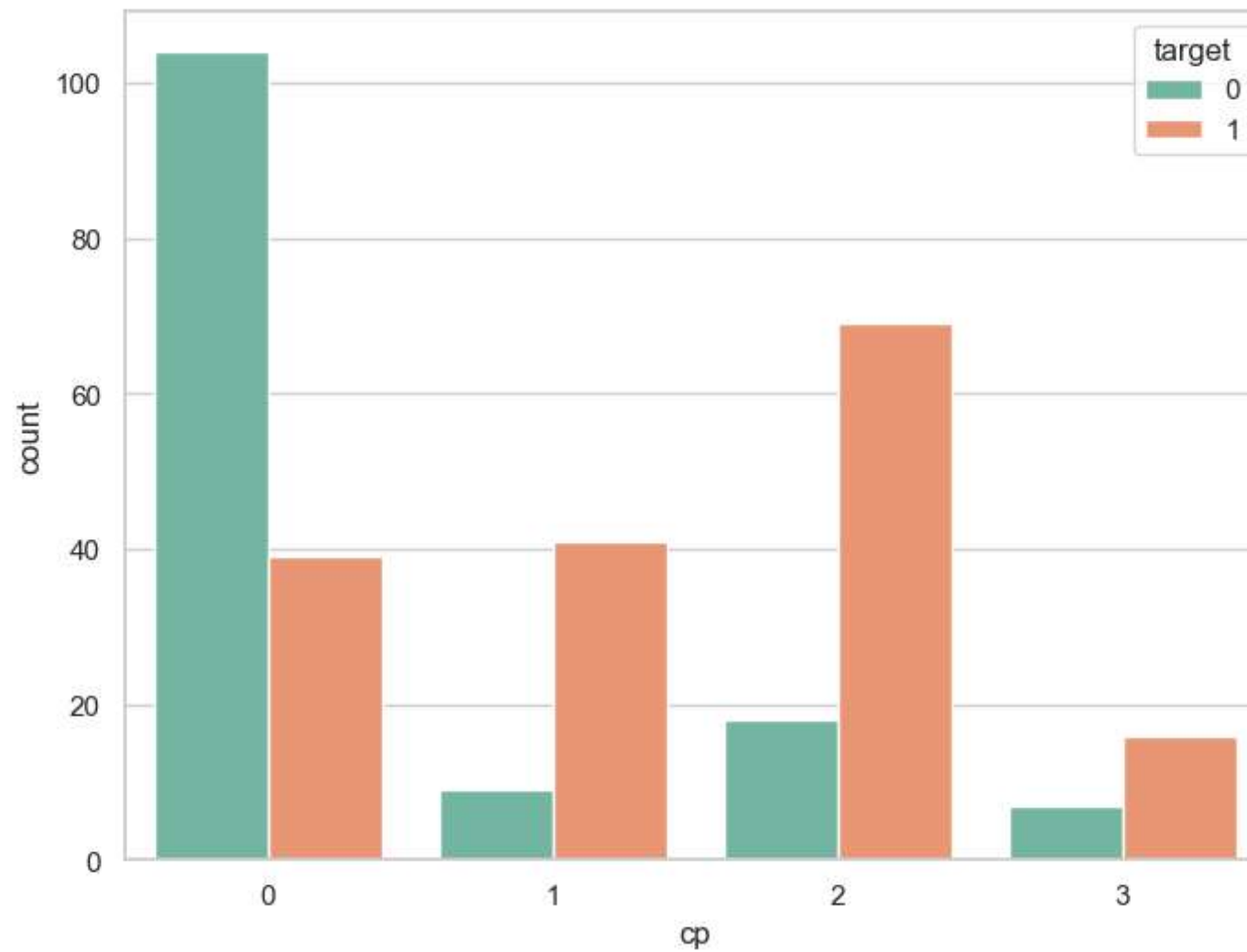
```
In [30]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="cp", data=df,palette={'red','green','purple','orange'})  
plt.show()
```



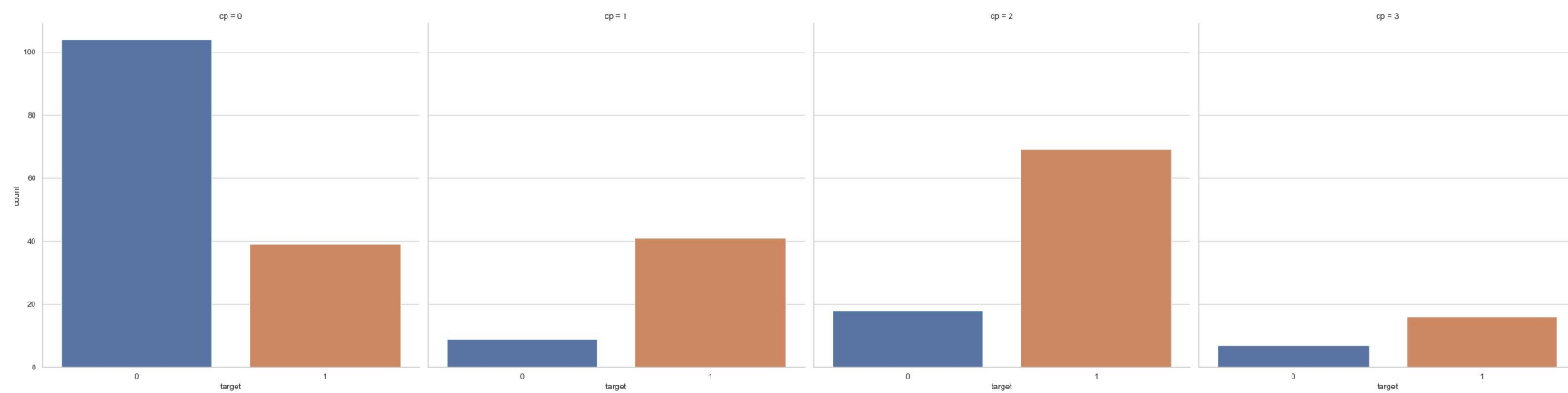

```
In [31]: df.groupby('cp')['target'].value_counts()
```

```
Out[31]: cp  target
0    0      104
      1       39
1    1       41
      0        9
2    1       69
      0       18
3    1       16
      0        7
Name: target, dtype: int64
```

```
In [33]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.countplot(x="cp", hue="target", data=df,palette="Set2")  
plt.show()
```



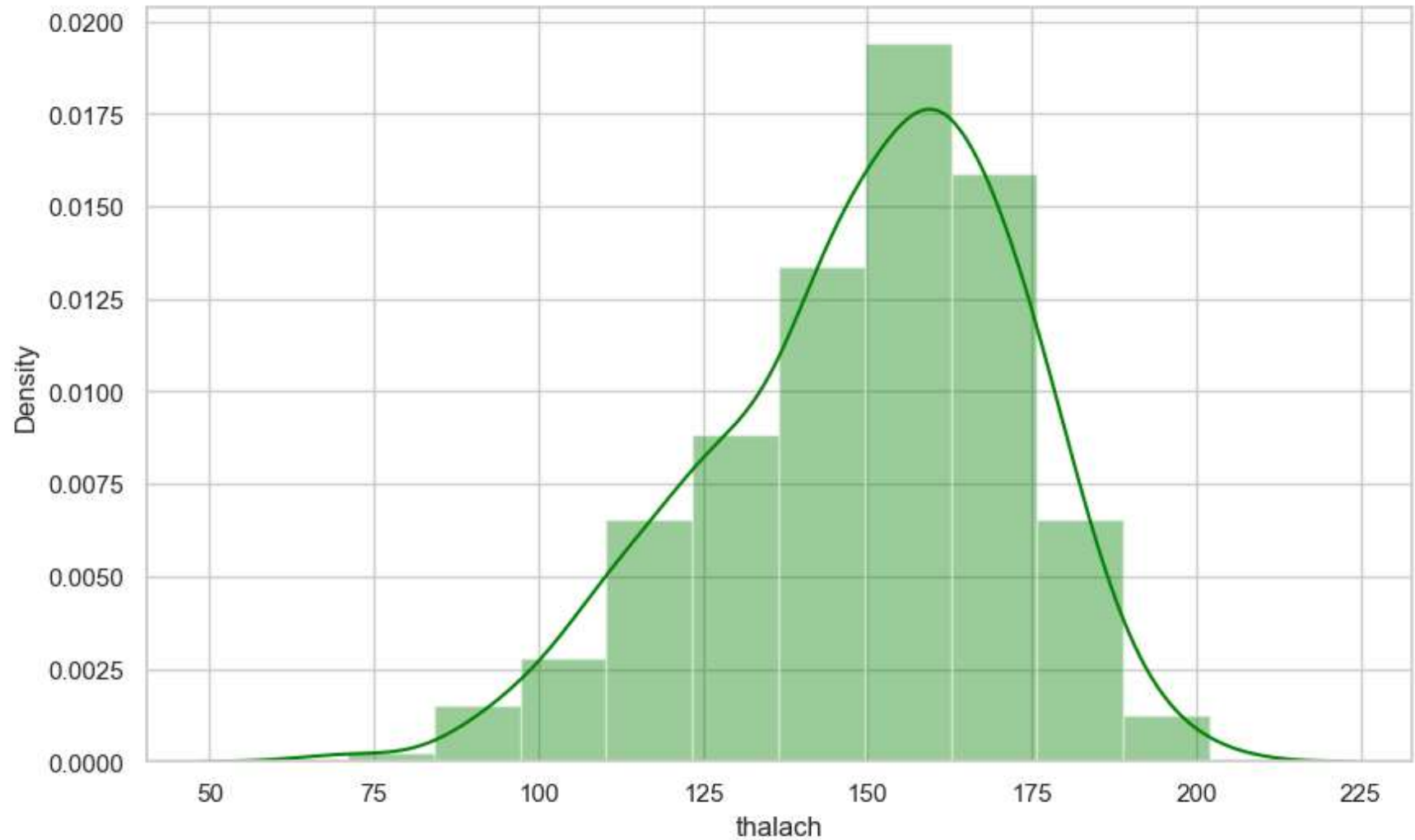
```
In [34]: ax = sns.catplot(x="target", col="cp", data=df, kind="count", height=8, aspect=1)
```



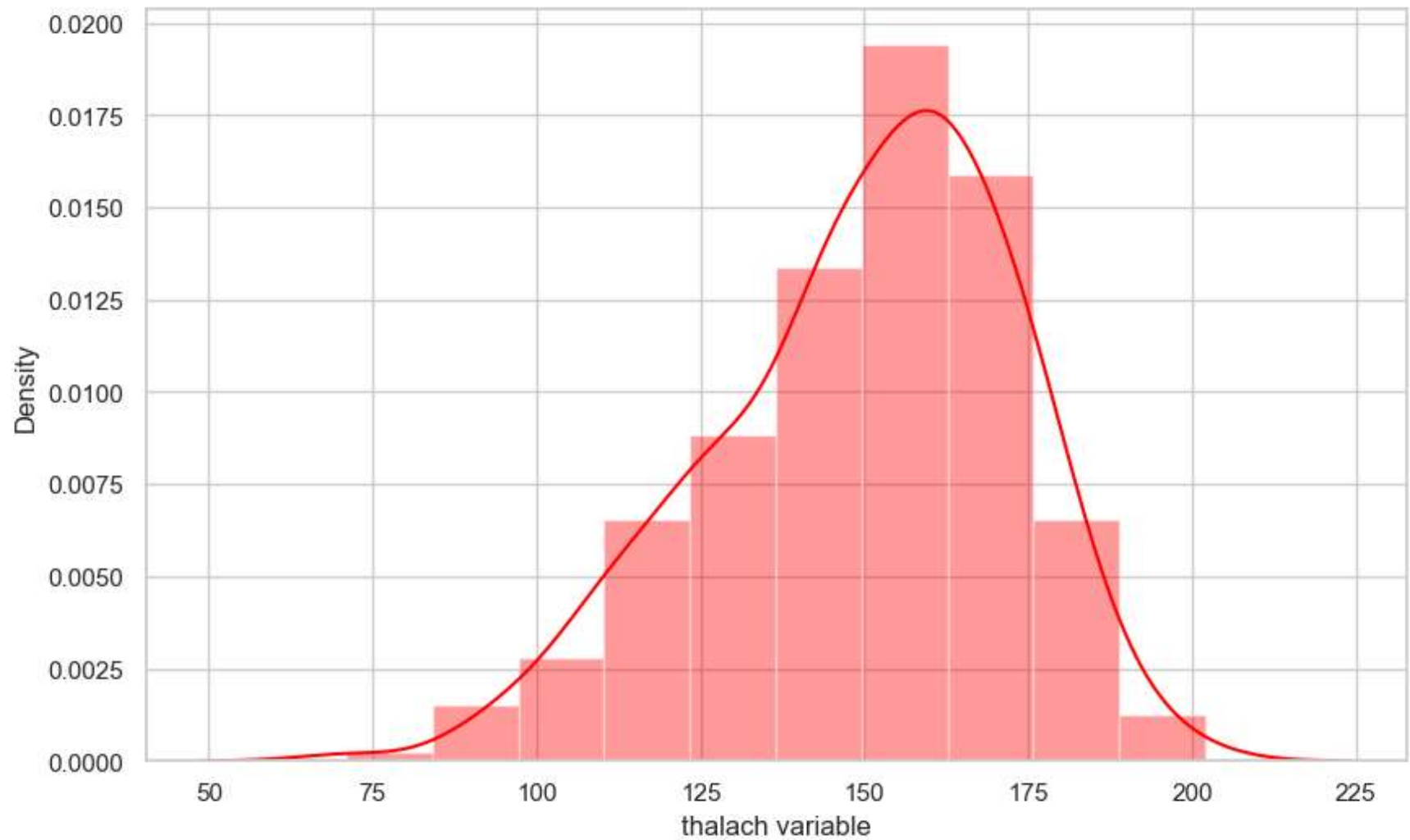
```
In [35]: df['thalach'].nunique()
```

Out[35]: 91

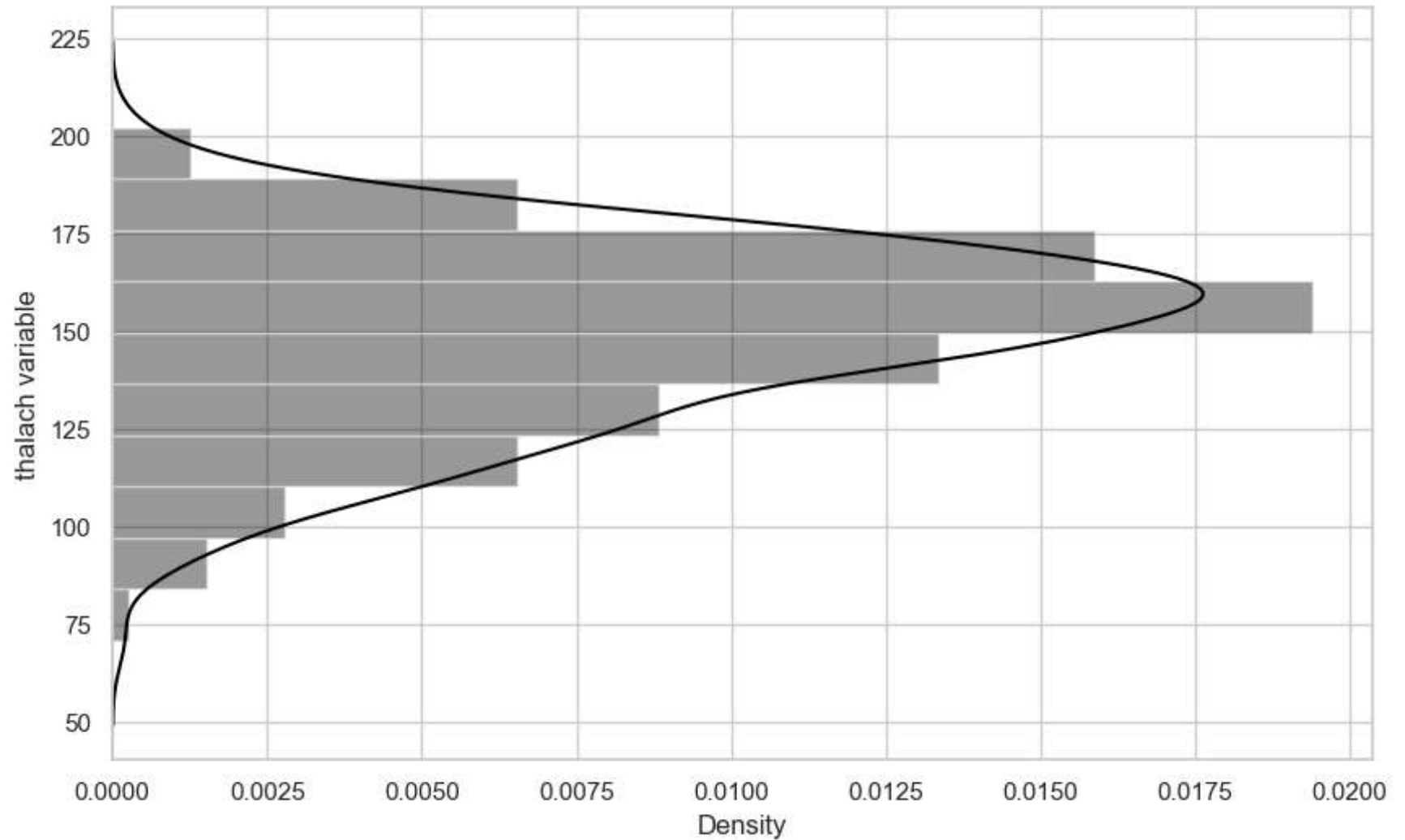
```
In [36]: f, ax = plt.subplots(figsize=(10,6))  
x = df['thalach']  
ax = sns.distplot(x, bins=10,color='green')  
plt.show()
```



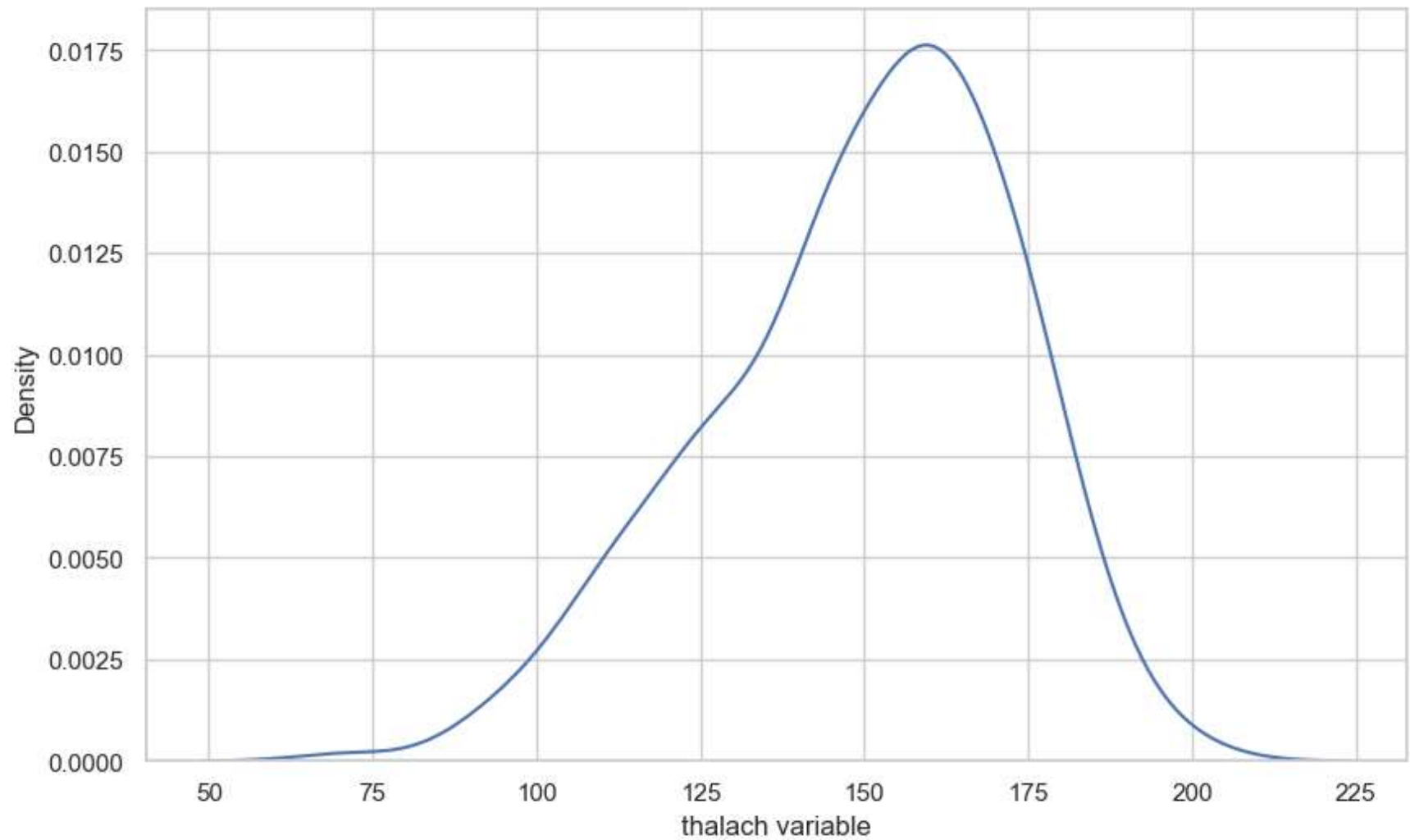
```
In [37]: f, ax = plt.subplots(figsize=(10,6))  
x = df['thalach']  
x = pd.Series(x, name="thalach variable")  
ax = sns.distplot(x, bins=10,color='red')  
plt.show()
```



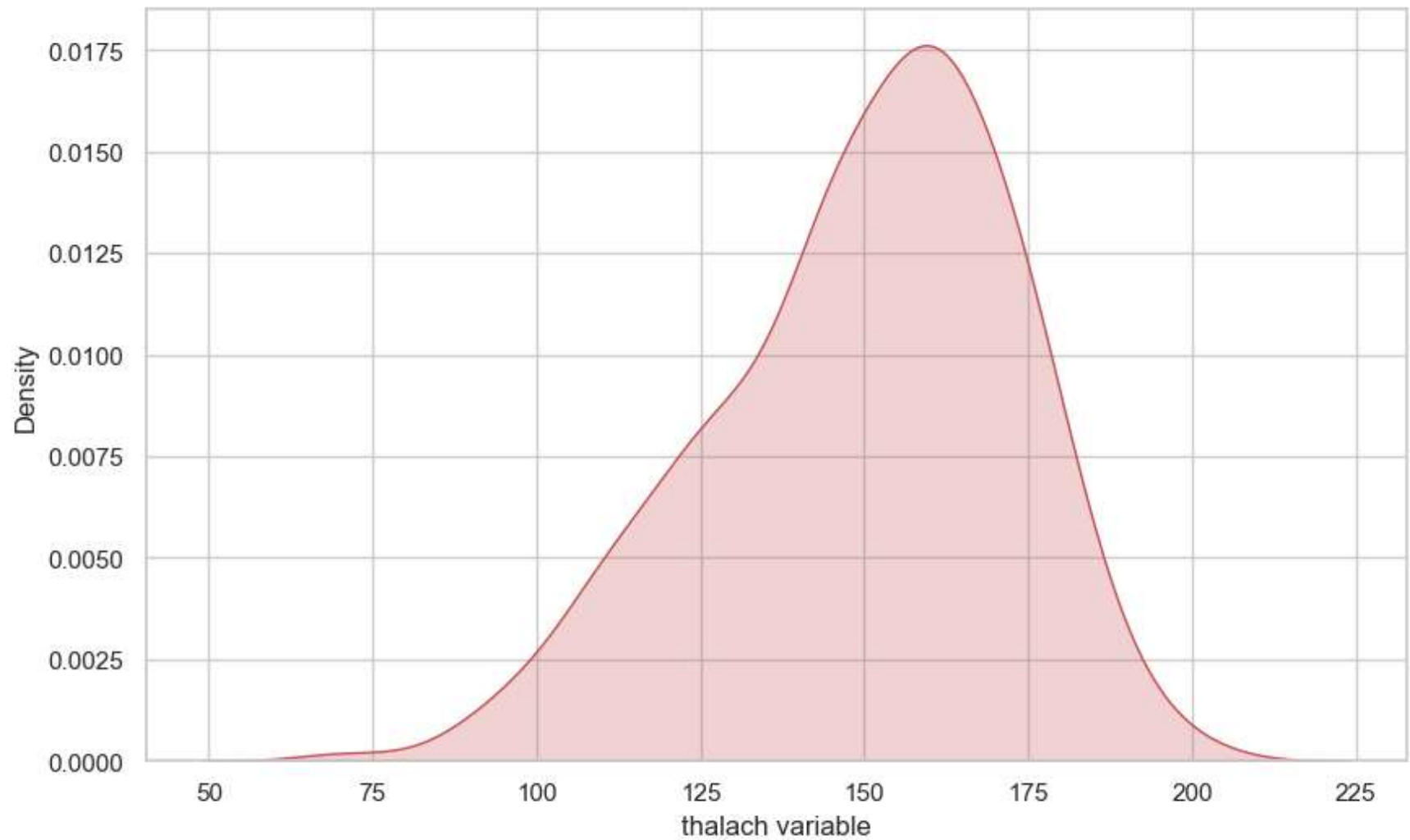
```
In [38]: f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.distplot(x, bins=10,color='black',vertical=True)
plt.show()
```



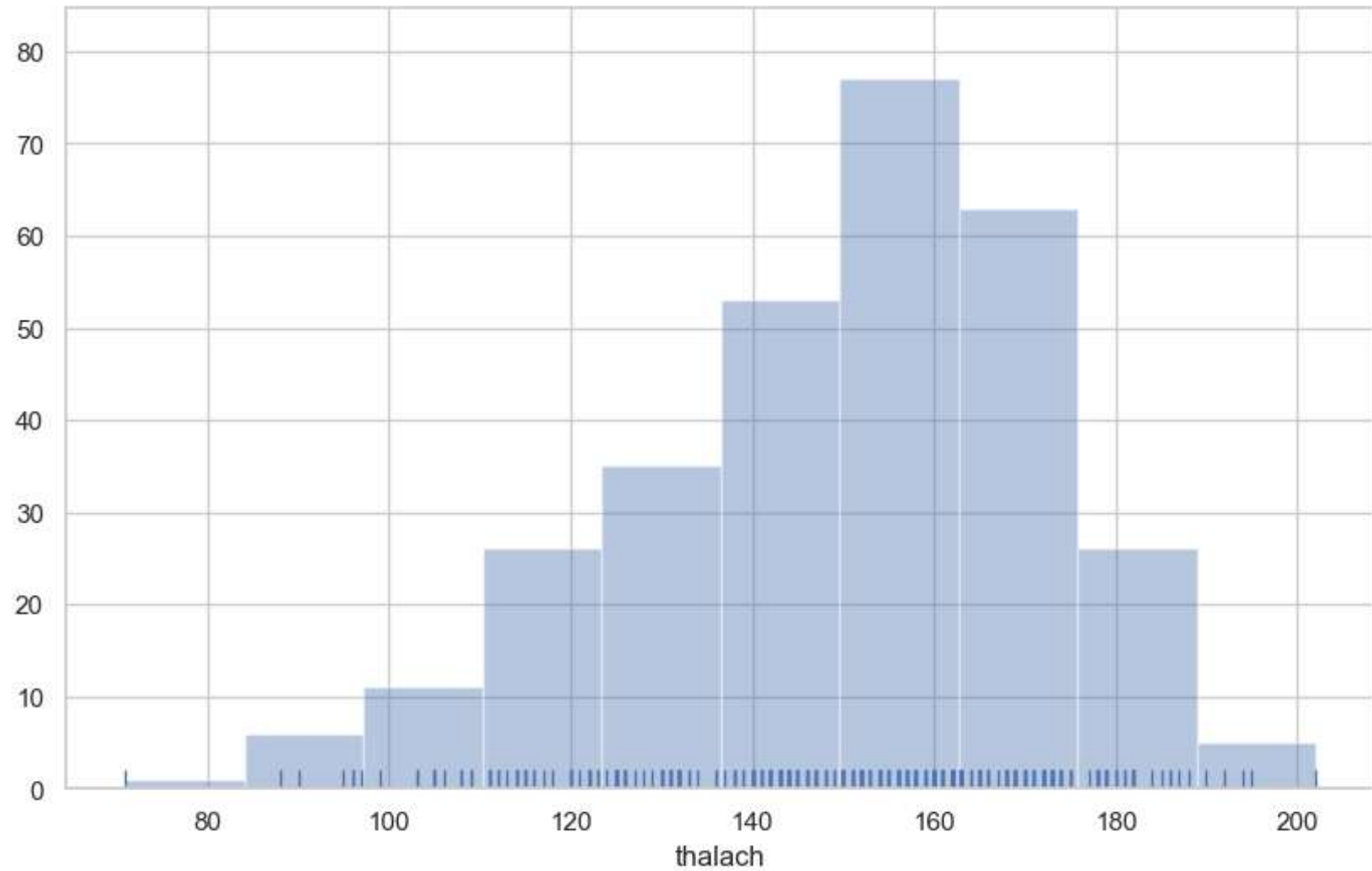
```
In [39]: f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.kdeplot(x)
plt.show()
```



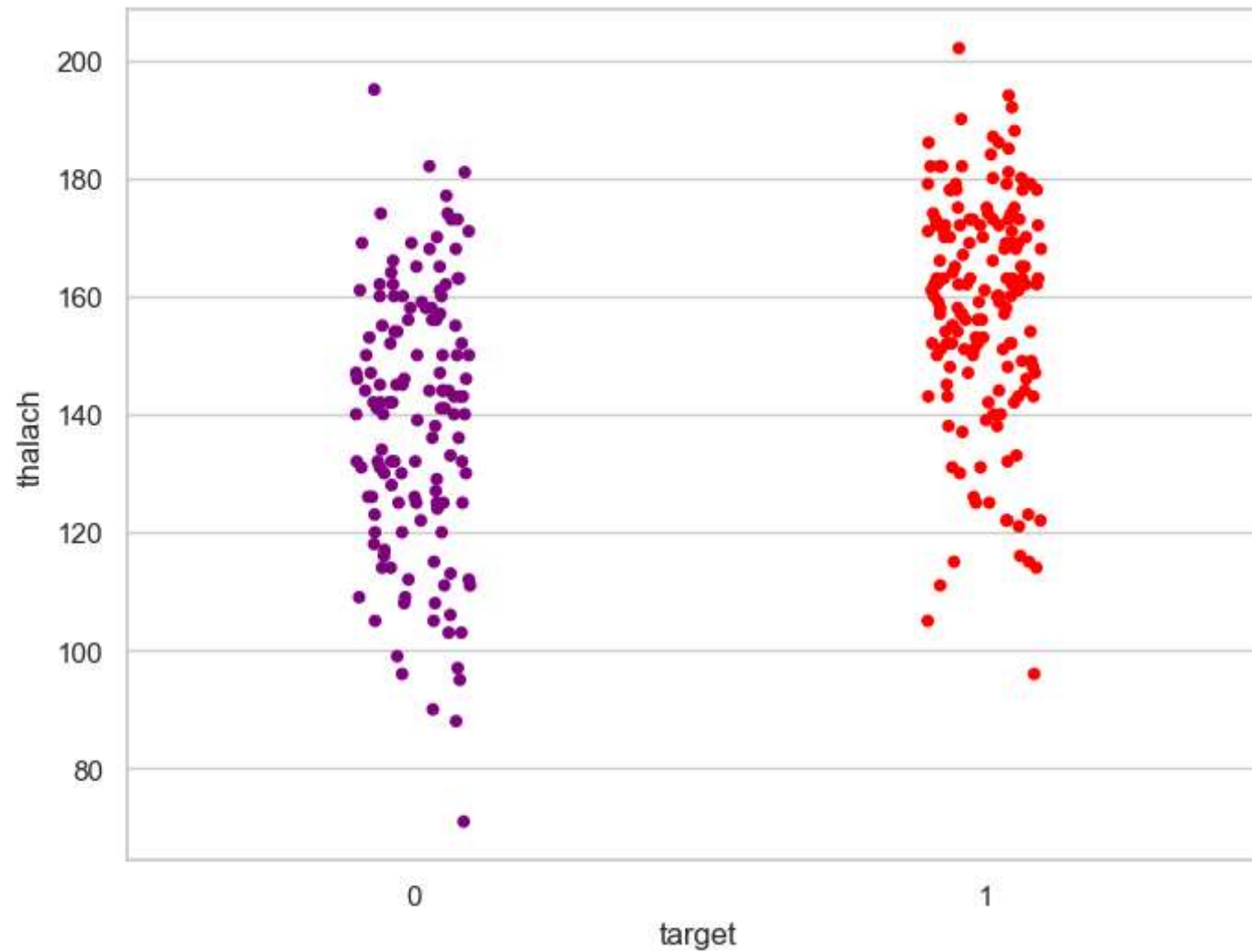
```
In [40]: f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.kdeplot(x, shade=True, color='r')
plt.show()
```




```
In [41]: f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
ax = sns.distplot(x, kde=False, rug=True, bins=10)
plt.show()
```

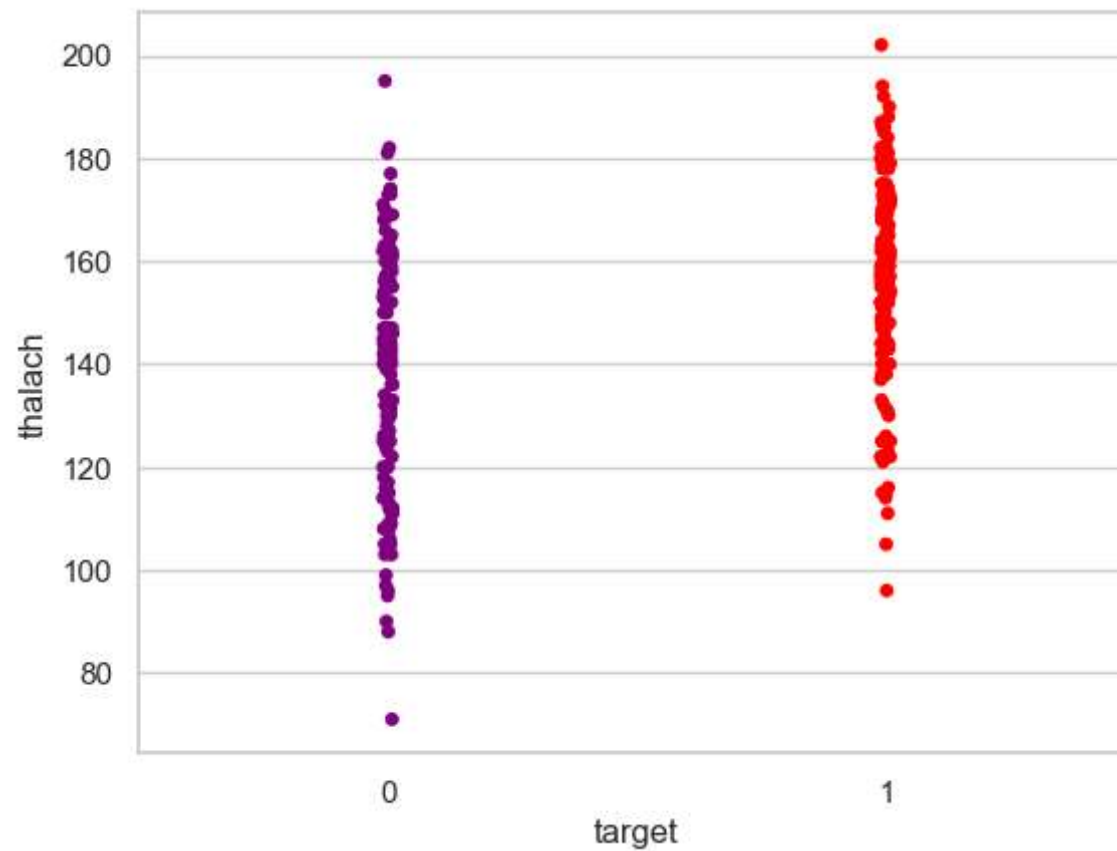


```
In [42]: f, ax = plt.subplots(figsize=(8, 6))  
sns.stripplot(x="target", y="thalach", data=df,palette={'red','purple'})  
plt.show()
```

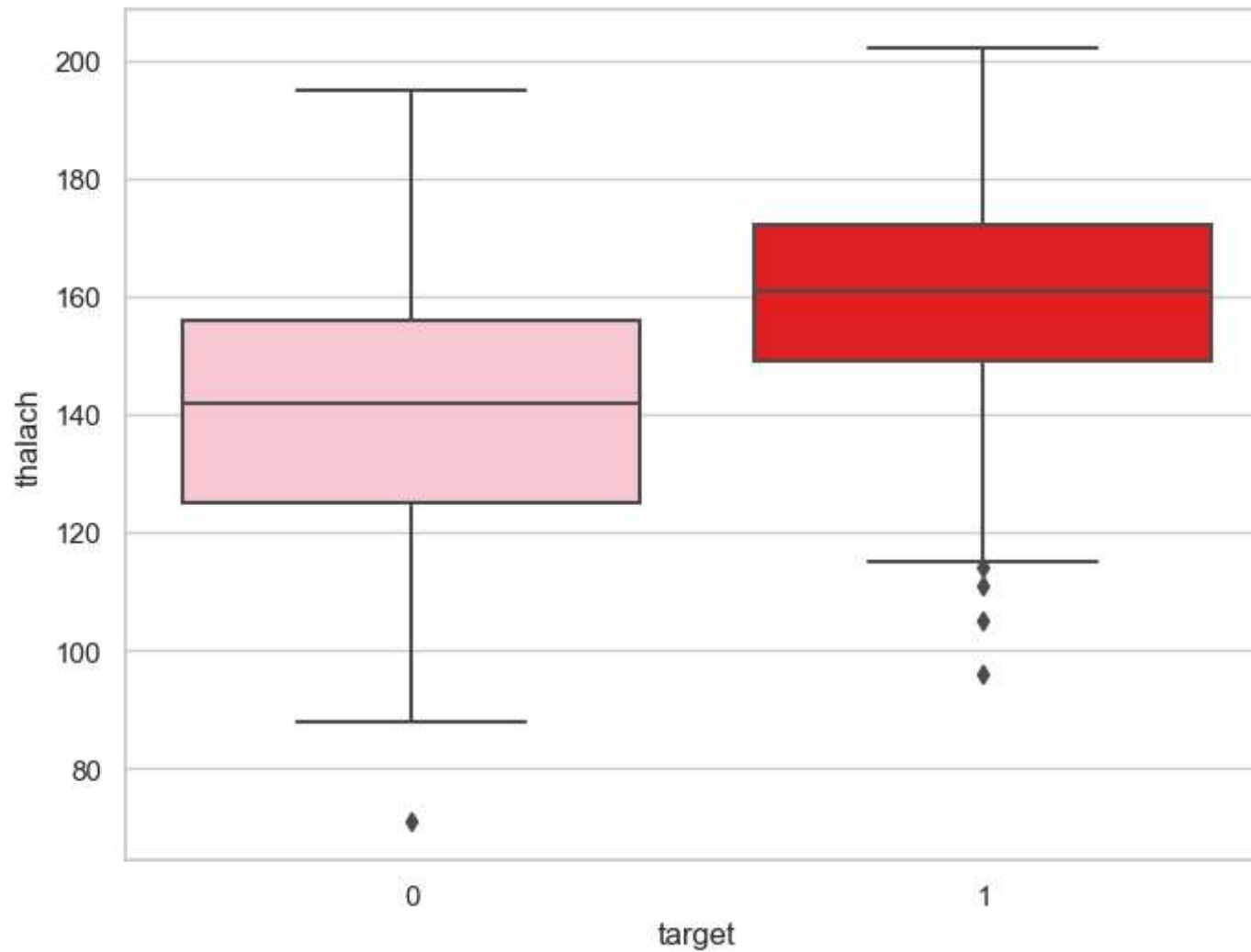


```
In [43]: ▶ sns.stripplot(x="target", y="thalach", data=df,palette={'red','purple'},jitter=0.01)
```

```
Out[43]: <Axes: xlabel='target', ylabel='thalach'>
```

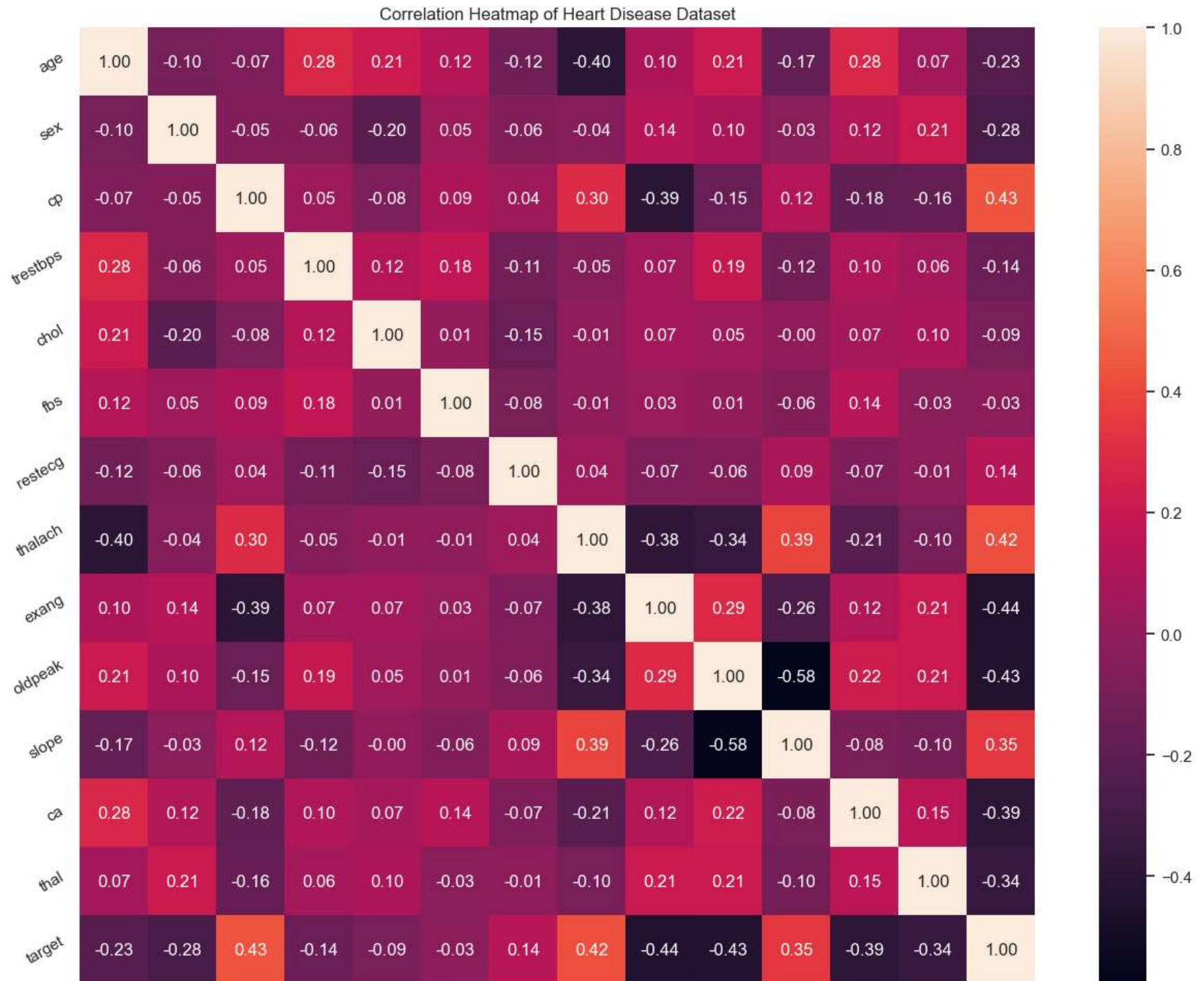


```
In [44]: f, ax = plt.subplots(figsize=(8, 6))  
sns.boxplot(x="target", y="thalach", data=df,palette={'pink','red'})  
plt.show()
```



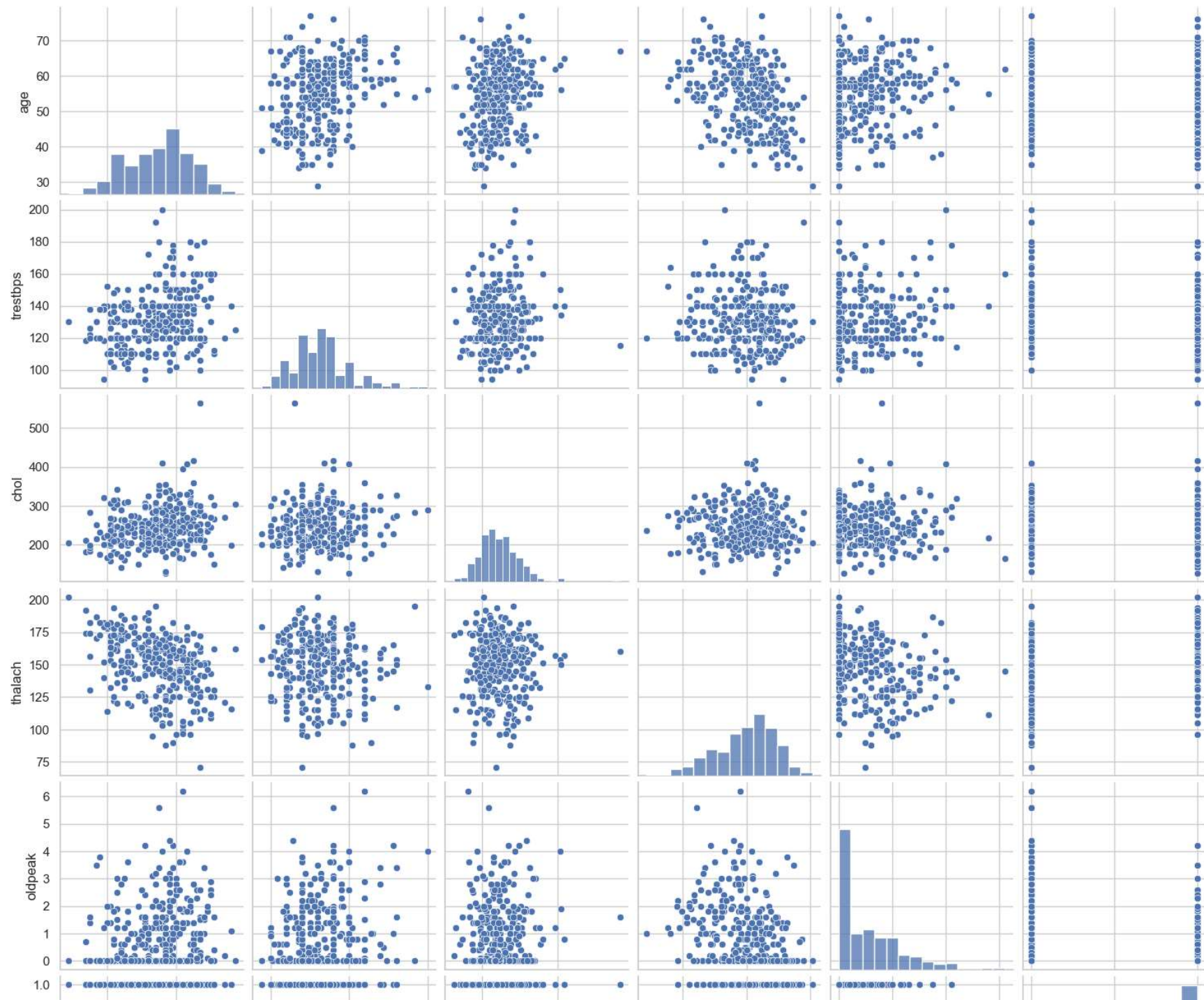
Multivariate Analysis

```
In [45]: ▶ plt.figure(figsize=(16,12))
plt.title('Correlation Heatmap of Heart Disease Dataset')
a = sns.heatmap(correlation, square=True, annot=True, fmt='.2f', linecolor='white')
a.set_xticklabels(a.get_xticklabels(), rotation=90)
a.set_yticklabels(a.get_yticklabels(), rotation=30)
plt.show()
```

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


```
In [46]: ▶ num_var = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'target' ]  
sns.pairplot(df[num_var], kind='scatter', diag_kind='hist')  
plt.show()
```

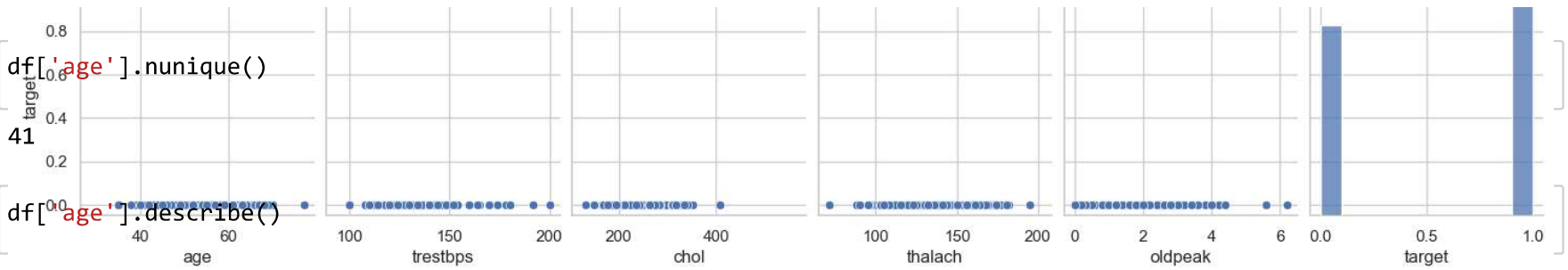



```
In [47]: df['age'].nunique()
```

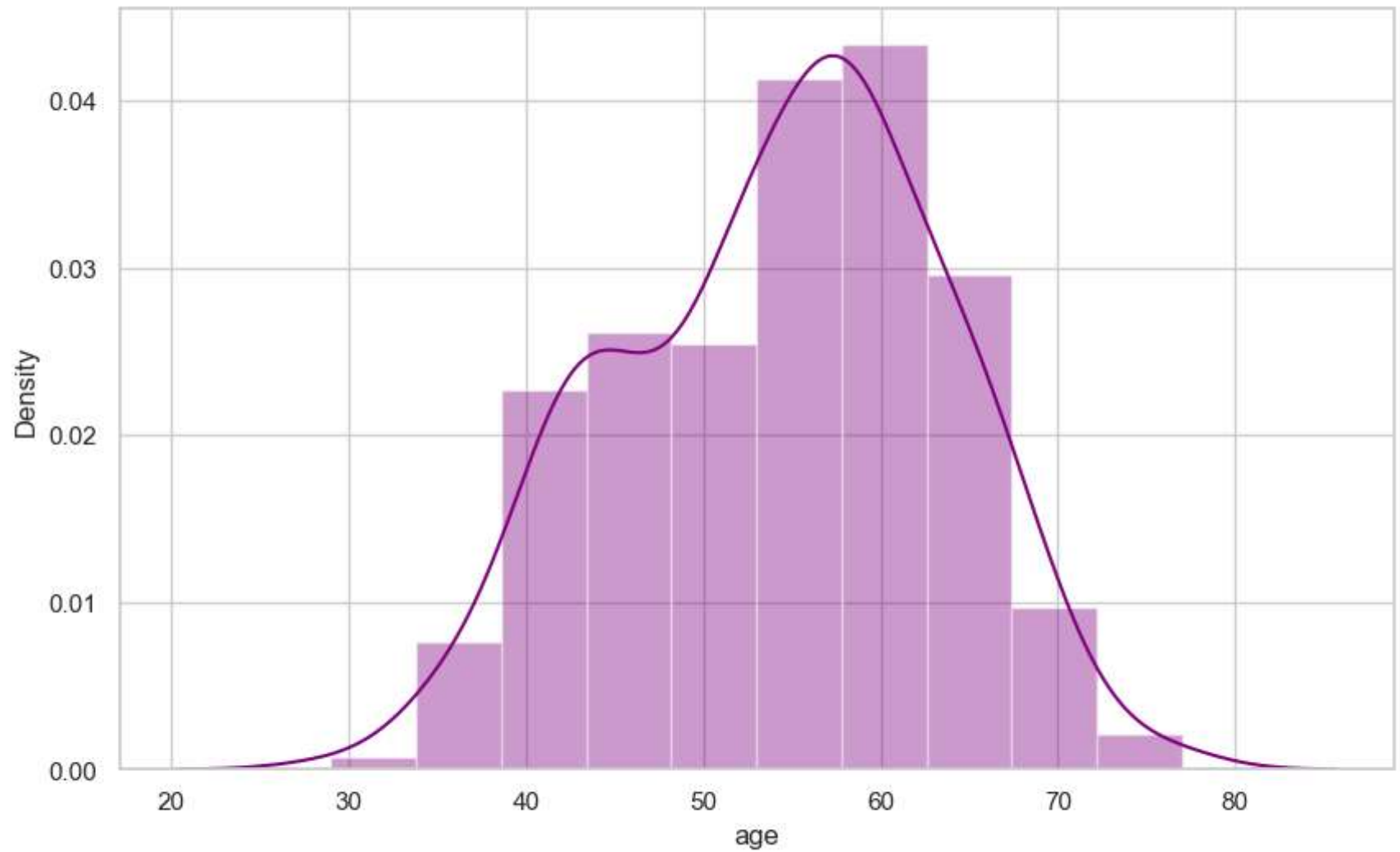
```
Out[47]: 41
```

```
In [48]: df['age'].describe()
```

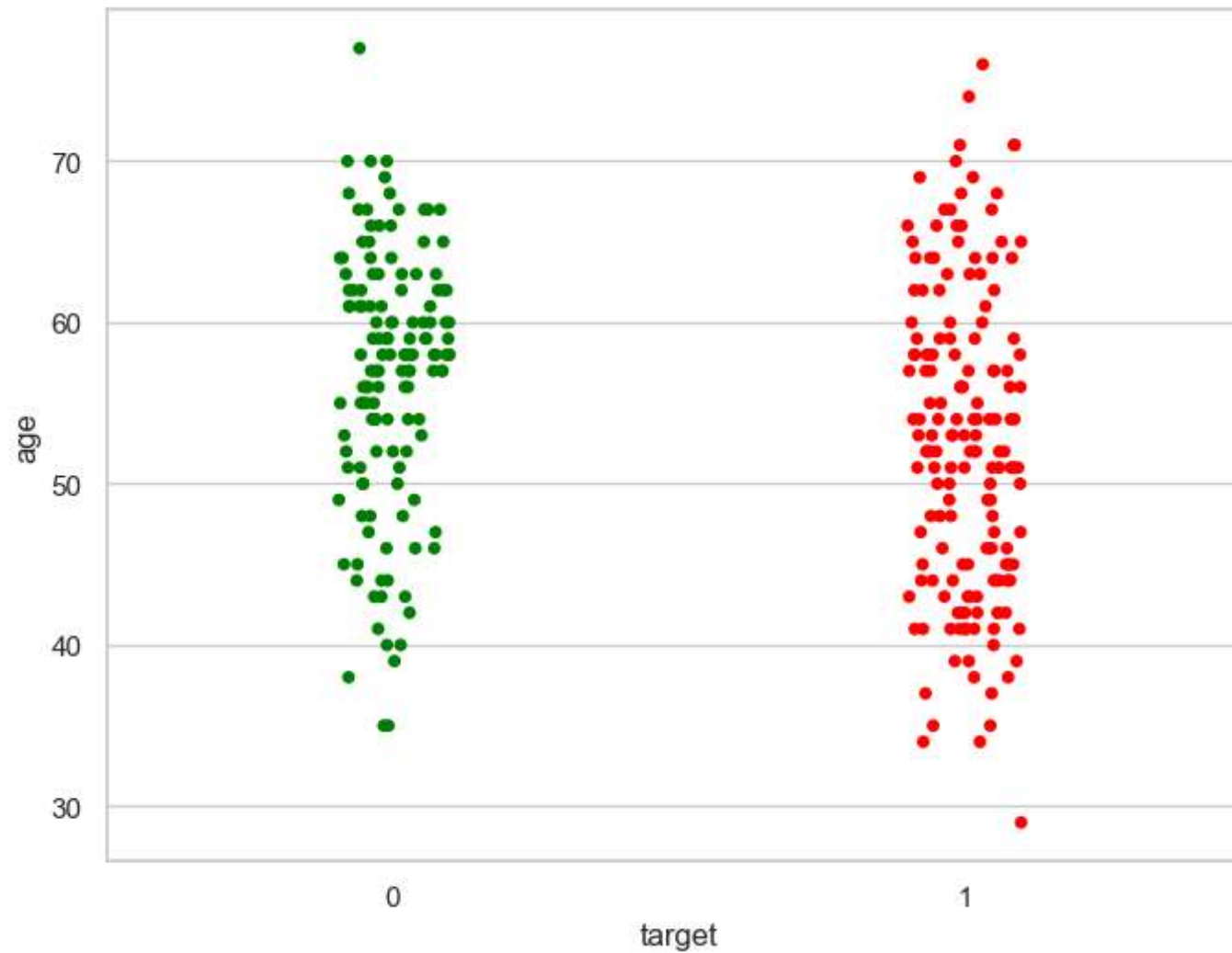
```
Out[48]: count    303.000000  
mean       54.366337  
std        9.082101  
min        29.000000  
25%        47.500000  
50%        55.000000  
75%        61.000000  
max        77.000000  
Name: age, dtype: float64
```



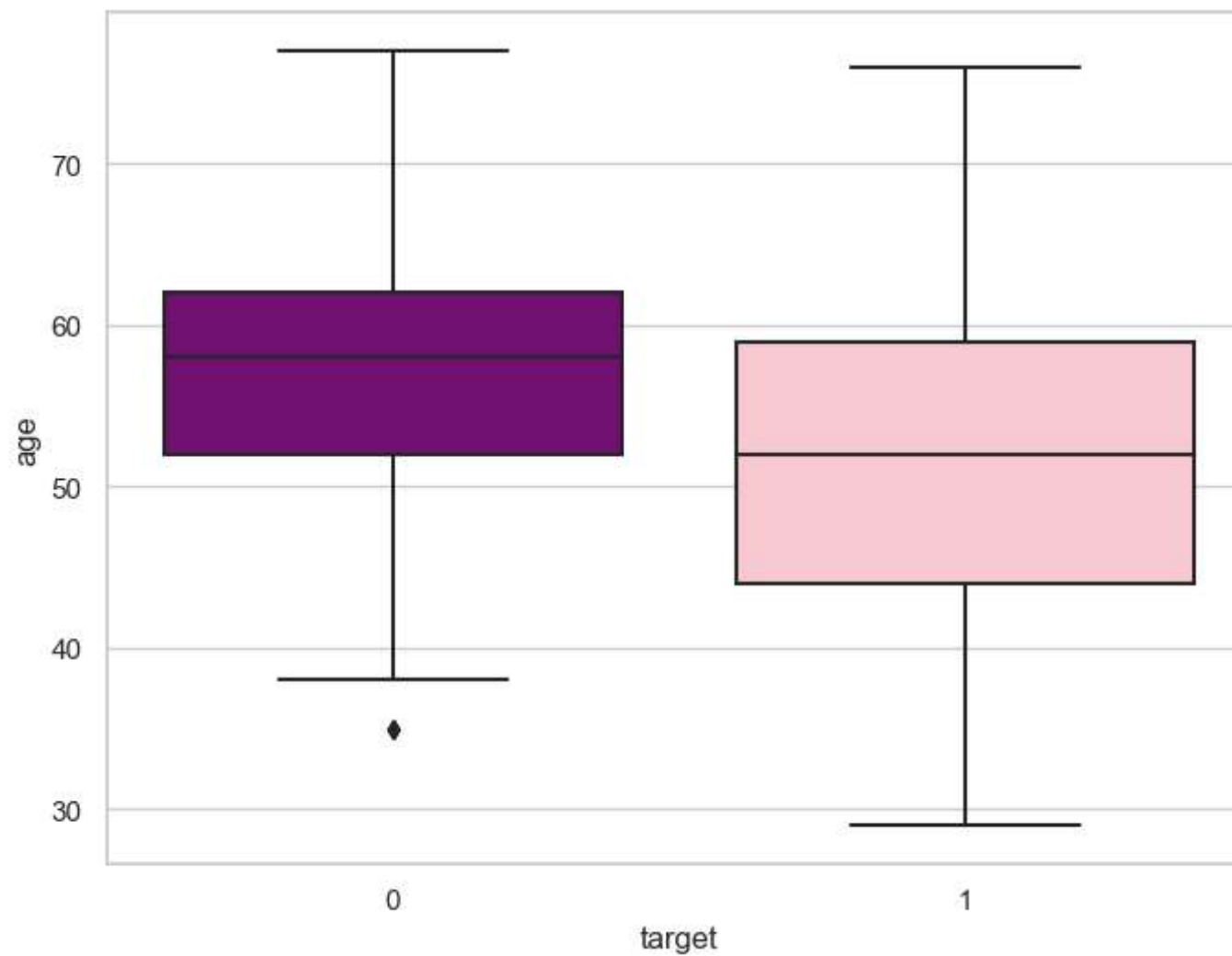
```
In [51]: f, ax = plt.subplots(figsize=(10,6))  
x = df['age']  
ax = sns.distplot(x, bins=10,color="purple")  
plt.show()
```



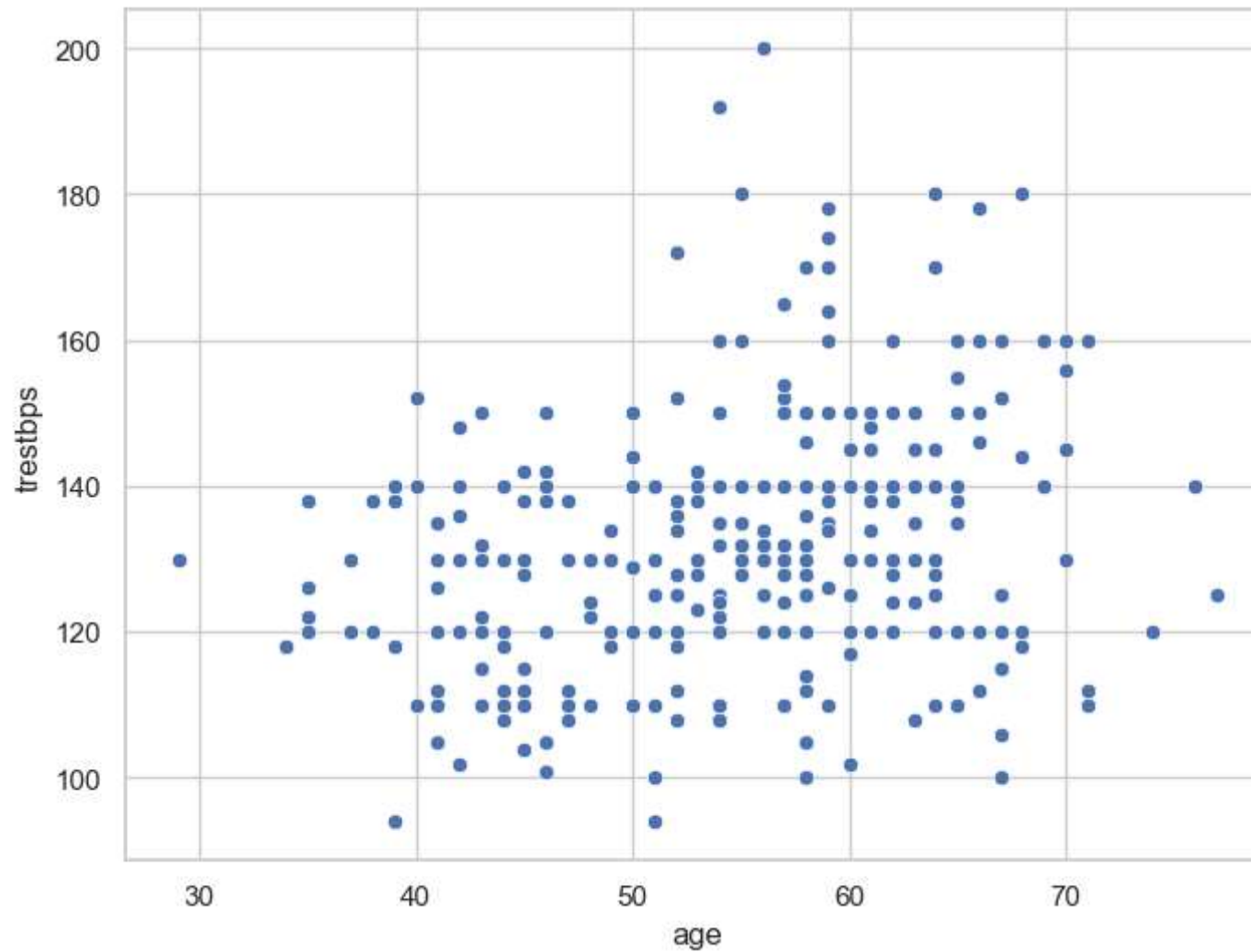
```
In [53]: f, ax = plt.subplots(figsize=(8, 6))  
sns.stripplot(x="target", y="age", data=df,palette={'red', 'green'})  
plt.show()
```



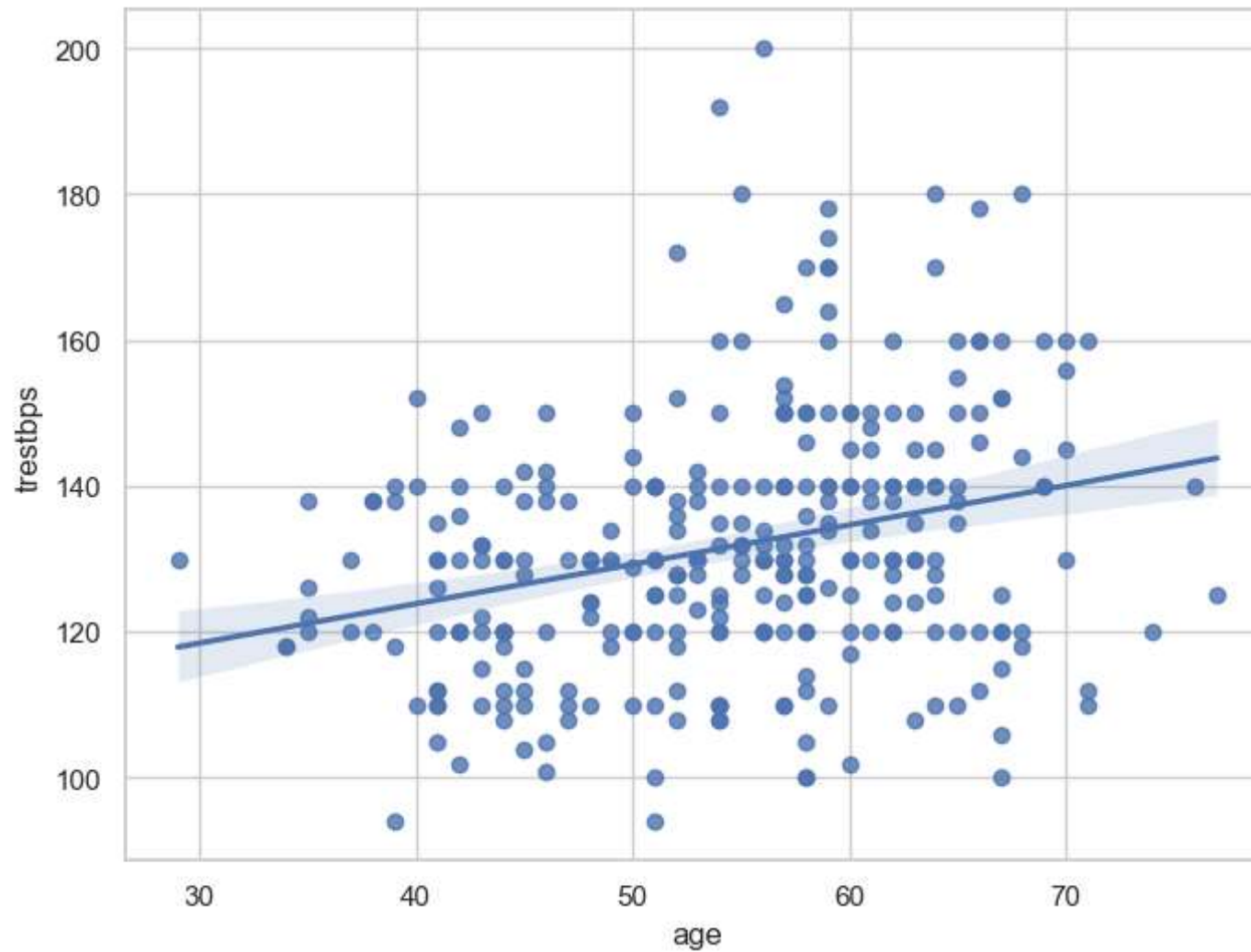
```
In [55]: f, ax = plt.subplots(figsize=(8, 6))  
sns.boxplot(x="target", y="age", data=df,palette={'purple','pink'})  
plt.show()
```



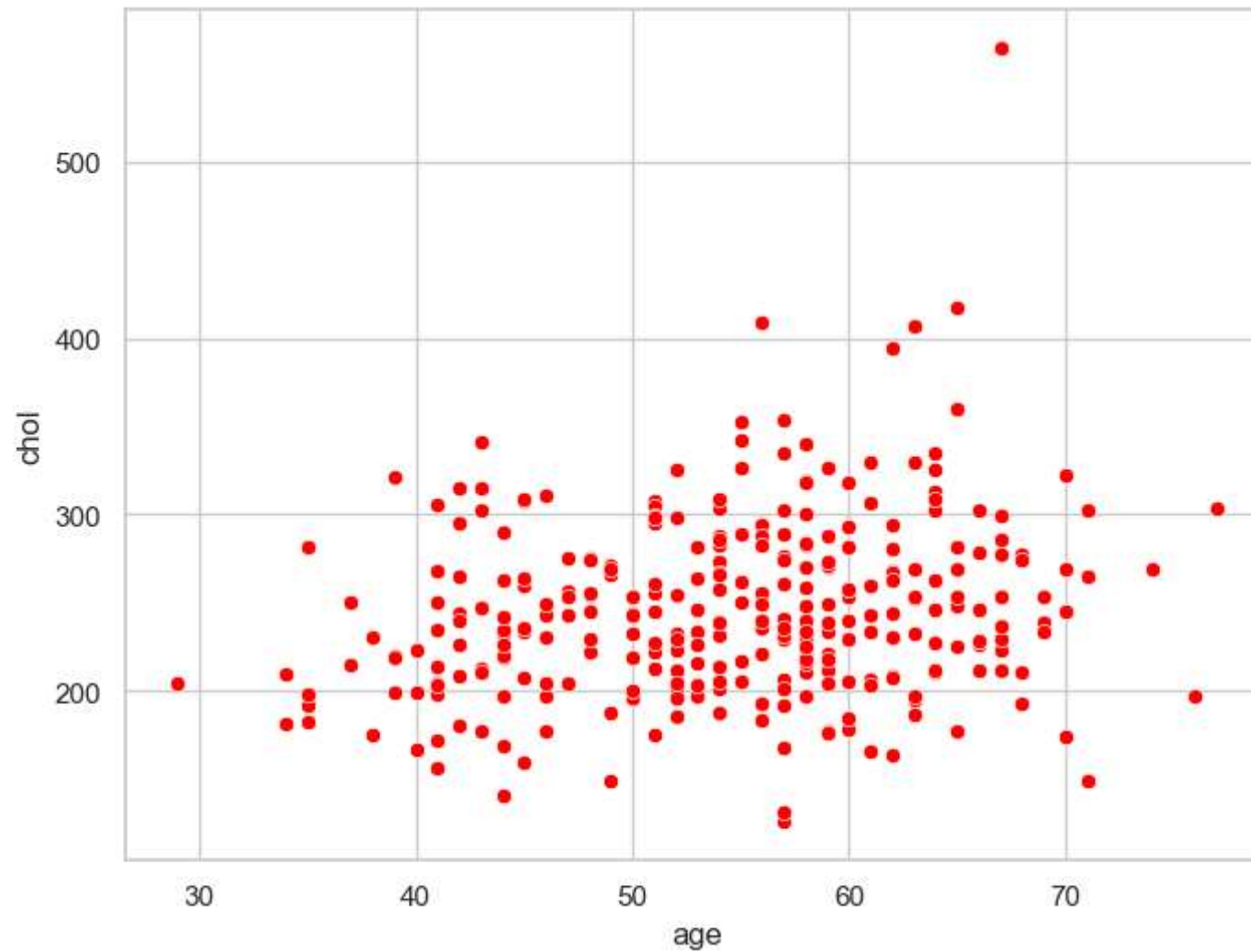
```
In [56]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.scatterplot(x="age", y="trestbps", data=df)  
plt.show()
```



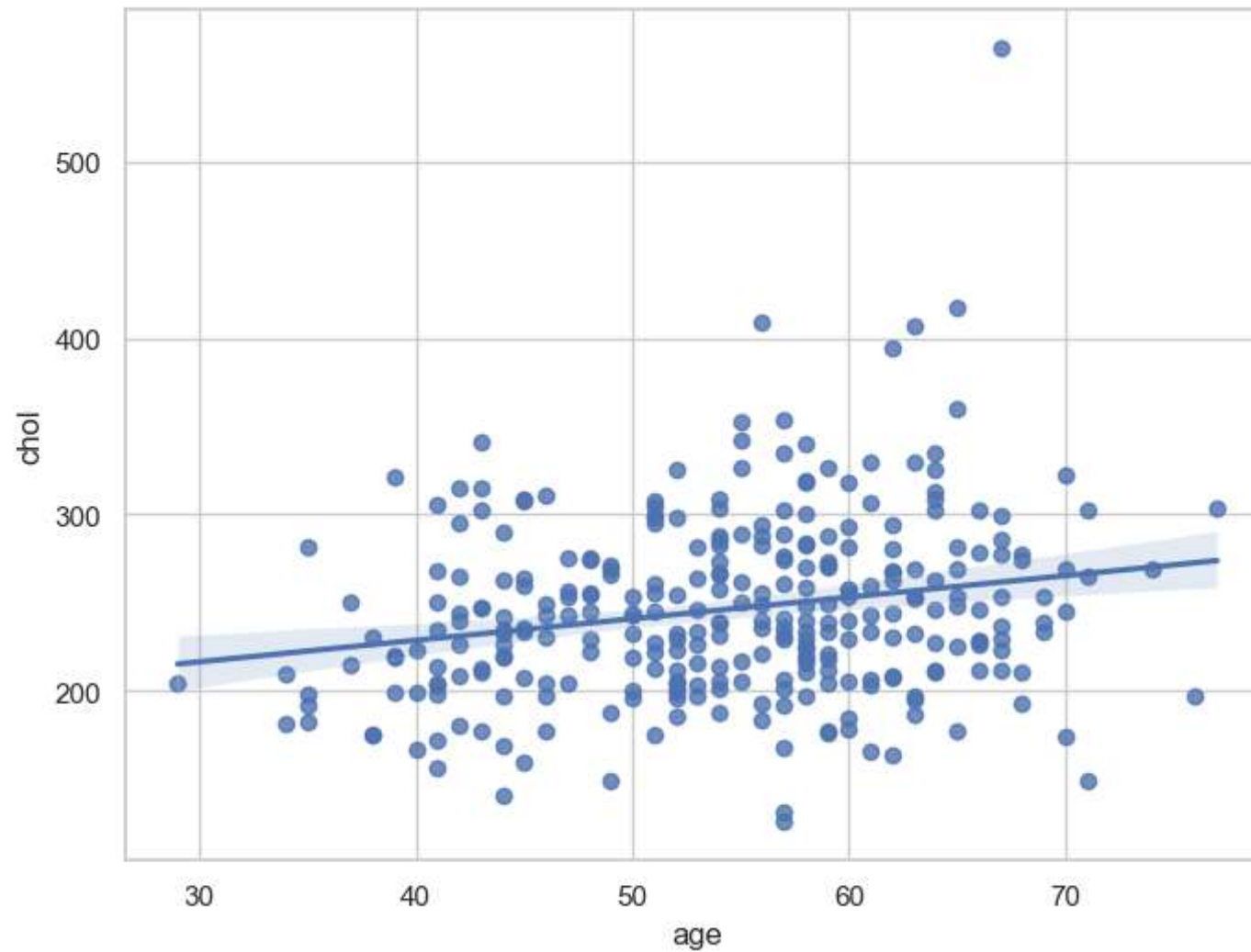

```
In [57]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.regplot(x="age", y="trestbps", data=df)  
plt.show()
```



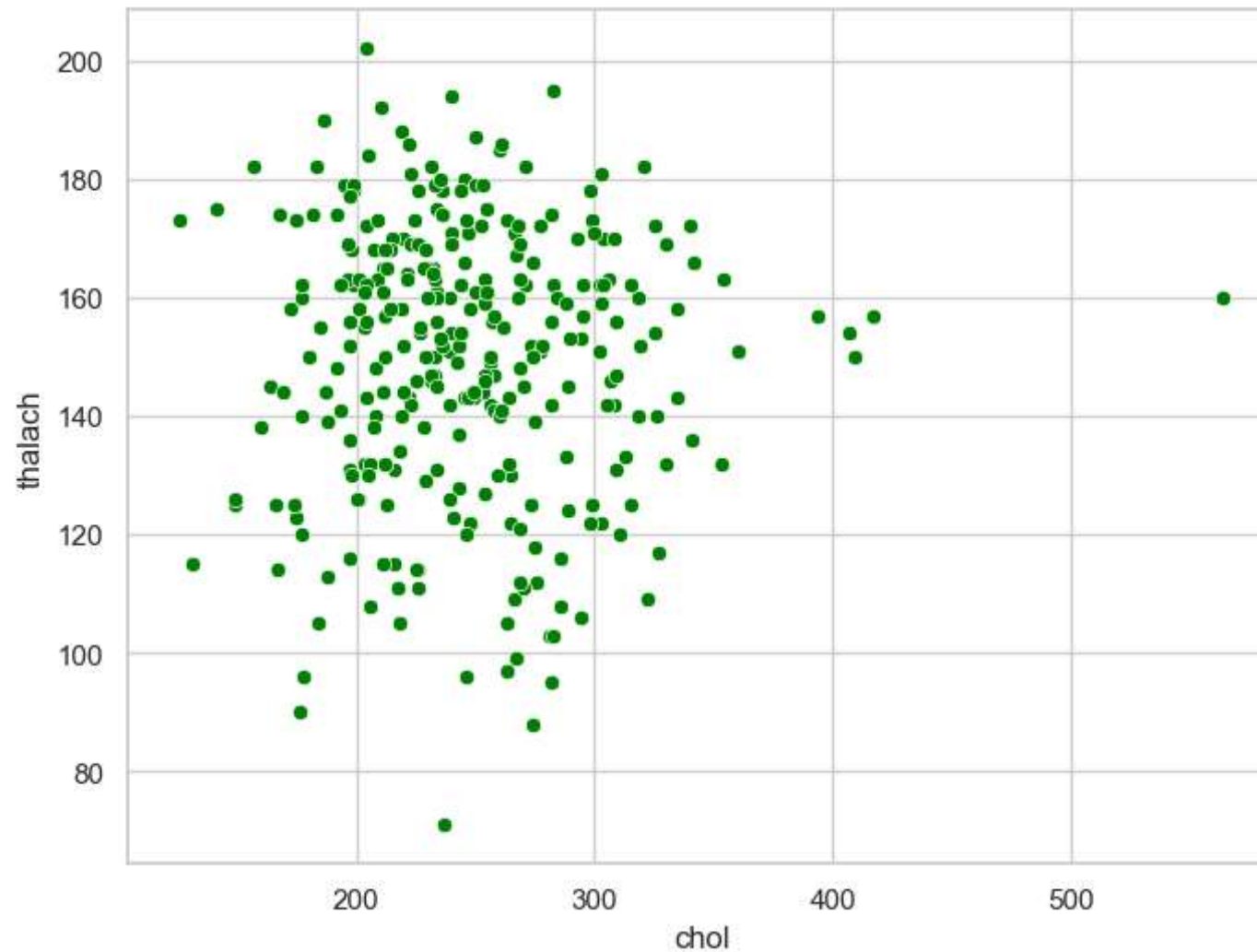
```
In [58]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.scatterplot(x="age", y="chol", data=df,color='red')  
plt.show()
```



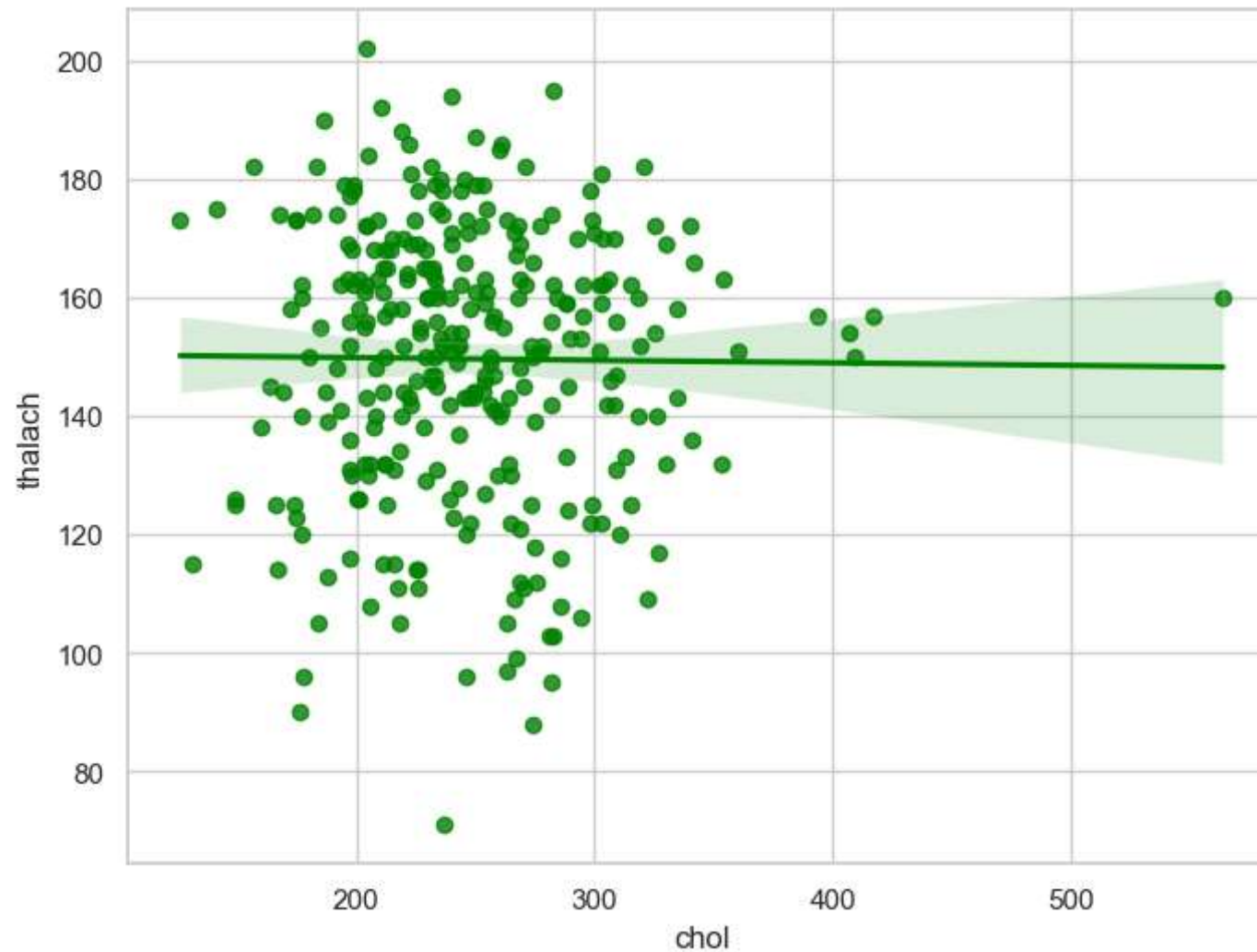
```
In [59]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.regplot(x="age", y="chol", data=df)  
plt.show()
```



```
In [60]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.scatterplot(x="chol", y = "thalach", data=df,color='green')  
plt.show()
```



```
In [61]: f, ax = plt.subplots(figsize=(8, 6))  
ax = sns.regplot(x="chol", y = "thalach", data=df,color='green')  
plt.show()
```



In [62]:  *# check for missing values*

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

Out[62]:

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 [63]:  *#assert that there are no missing values in the dataframe*

```
assert pd.notnull(df).all().all()
```

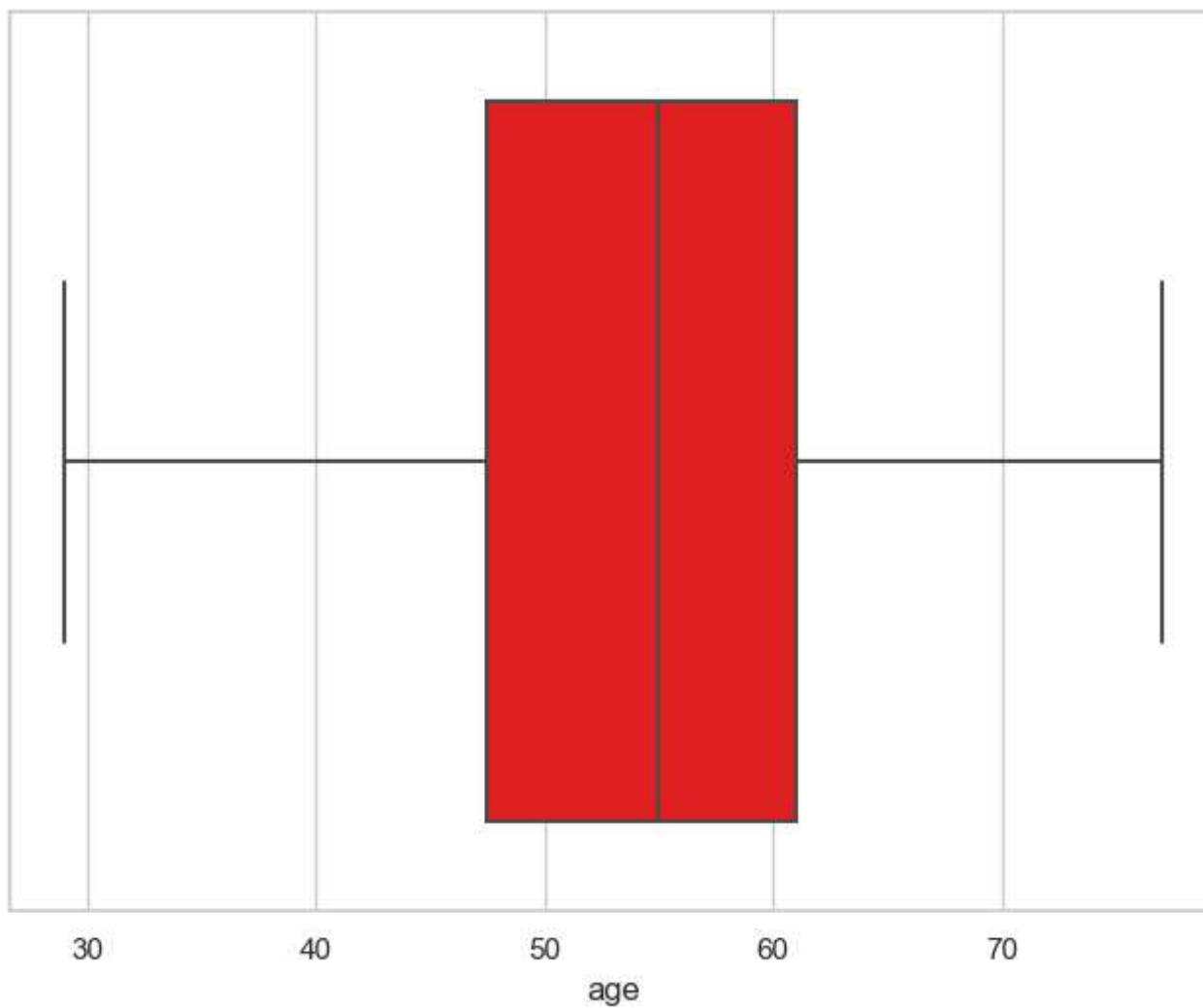
In [64]:  *#assert all values are greater than or equal to 0*

```
assert (df >= 0).all().all()
```

```
In [65]: df['age'].describe()
```

```
Out[65]: count    303.000000  
         mean     54.366337  
         std      9.082101  
         min     29.000000  
         25%     47.500000  
         50%     55.000000  
         75%     61.000000  
         max     77.000000  
         Name: age, dtype: float64
```

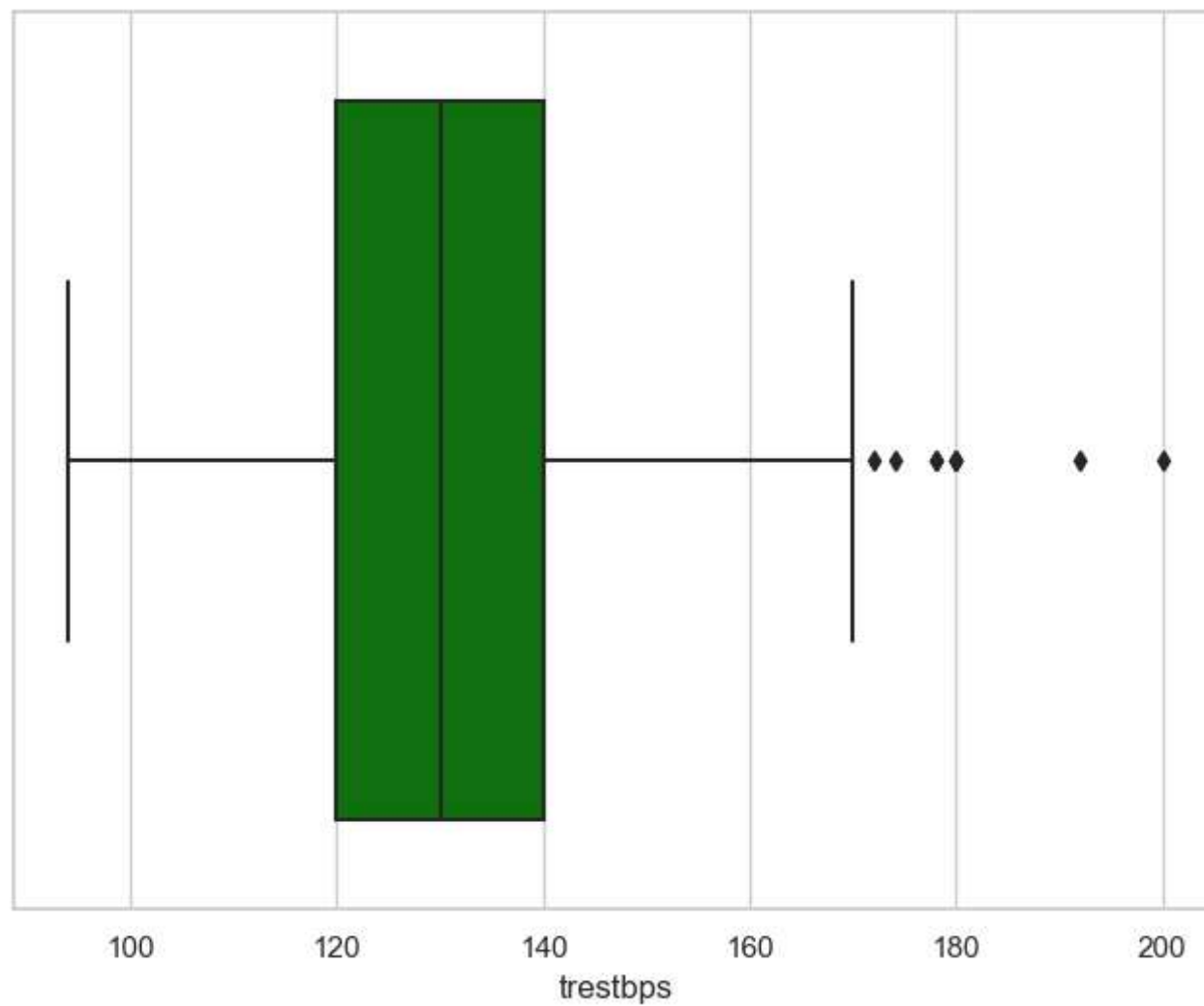
```
In [66]: f, ax = plt.subplots(figsize=(8, 6))  
sns.boxplot(x=df["age"],color='red')  
plt.show()
```




```
In [67]: df['trestbps'].describe()
```

```
Out[67]: count    303.000000  
         mean     131.623762  
         std      17.538143  
         min      94.000000  
         25%     120.000000  
         50%     130.000000  
         75%     140.000000  
         max     200.000000  
         Name: trestbps, dtype: float64
```

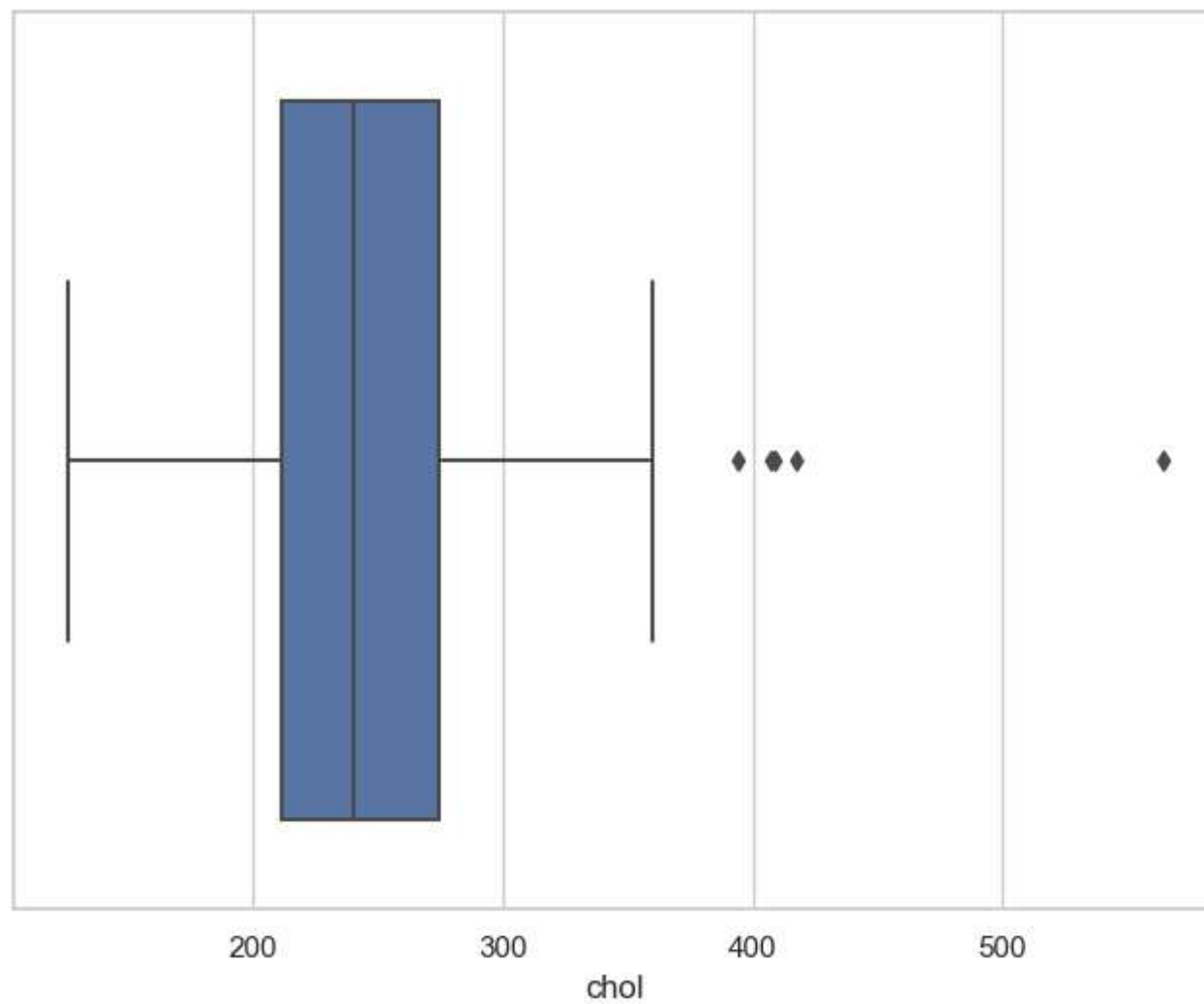
```
In [68]: f, ax = plt.subplots(figsize=(8, 6))  
sns.boxplot(x=df["trestbps"],color='green')  
plt.show()
```



```
In [69]: df['chol'].describe()
```

```
Out[69]: count    303.000000  
         mean     246.264026  
         std      51.830751  
         min     126.000000  
         25%     211.000000  
         50%     240.000000  
         75%     274.500000  
         max     564.000000  
         Name: chol, dtype: float64
```

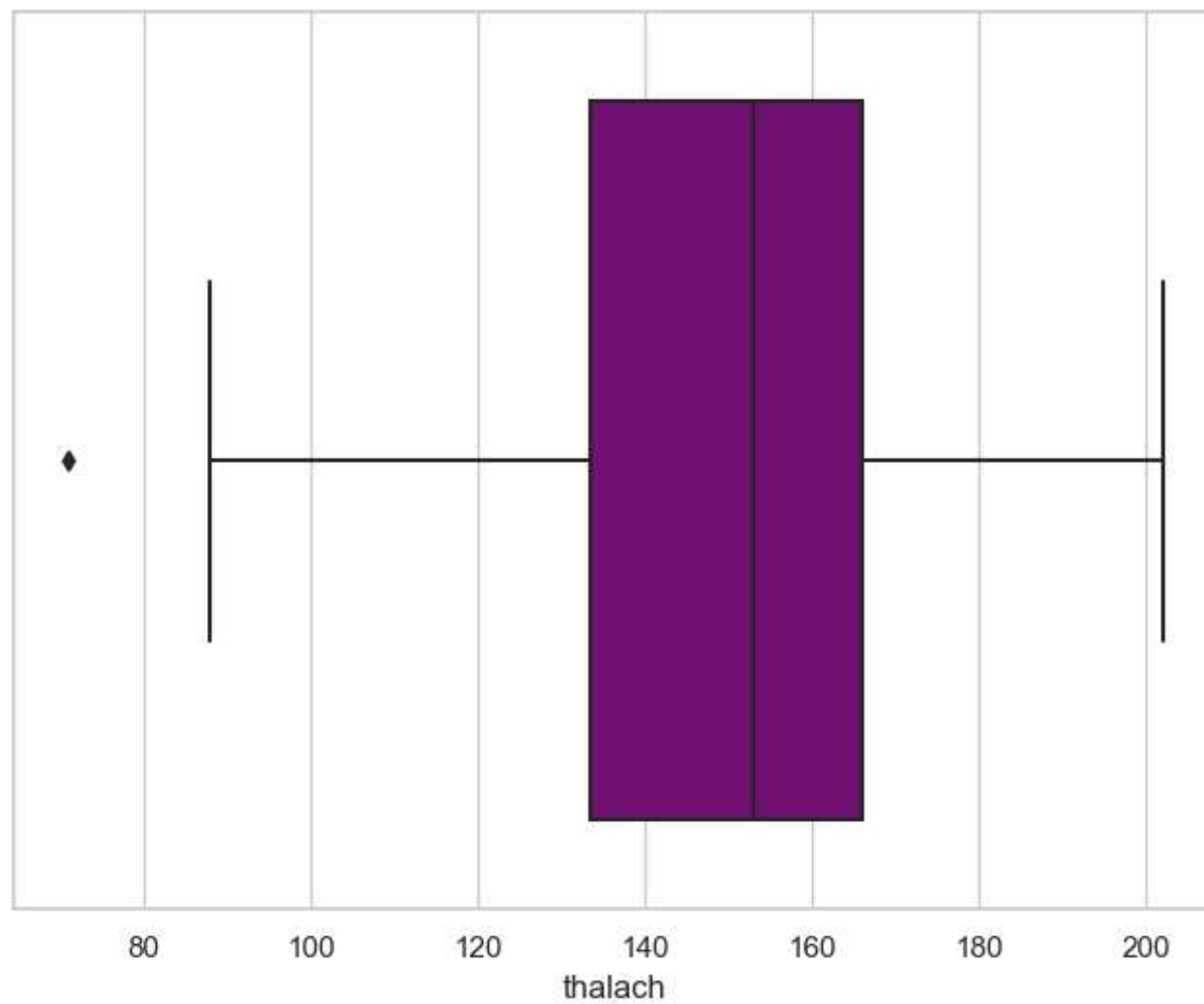
```
In [70]: f, ax = plt.subplots(figsize=(8, 6))  
sns.boxplot(x=df["chol"])  
plt.show()
```



```
In [71]: df['thalach'].describe()
```

```
Out[71]: count    303.000000  
         mean     149.646865  
         std      22.905161  
         min       71.000000  
         25%     133.500000  
         50%     153.000000  
         75%     166.000000  
         max     202.000000  
         Name: thalach, dtype: float64
```

```
In [72]: f, ax = plt.subplots(figsize=(8, 6))  
sns.boxplot(x=df["thalach"],color='purple')  
plt.show()
```



```
In [73]: df['oldpeak'].describe()
```

```
Out[73]: count    303.000000  
         mean      1.039604  
         std       1.161075  
         min       0.000000  
         25%       0.000000  
         50%       0.800000  
         75%       1.600000  
         max       6.200000  
         Name: oldpeak, dtype: float64
```

```
In [74]: f, ax = plt.subplots(figsize=(8, 6))  
sns.boxplot(x=df["oldpeak"],color='yellow')  
plt.show()
```

