- We can see that the `cp` and `thalach` variables are mildly positively correlated with `target` variable. So, I will analyze the interaction between these features and `target` variable.

Analysis of `target` and `cp` variable

Explore `cp` variable

- `cp` stands for chest pain type.
- First, I will check number of unique values in `cp` variable.

In [63]:
```
df['cp'].nunique()
```

Out[63]:
```
4
```

So, there are 4 unique values in `cp` variable. Hence, it is a categorical variable.

Now, I will view its frequency distribution as follows :

In [66]:
```
df['cp'].unique()
```

Out[66]:
```
array([3, 2, 1, 0], dtype=int64)
```

In [67]:
```
df['cp'].value_counts()
```

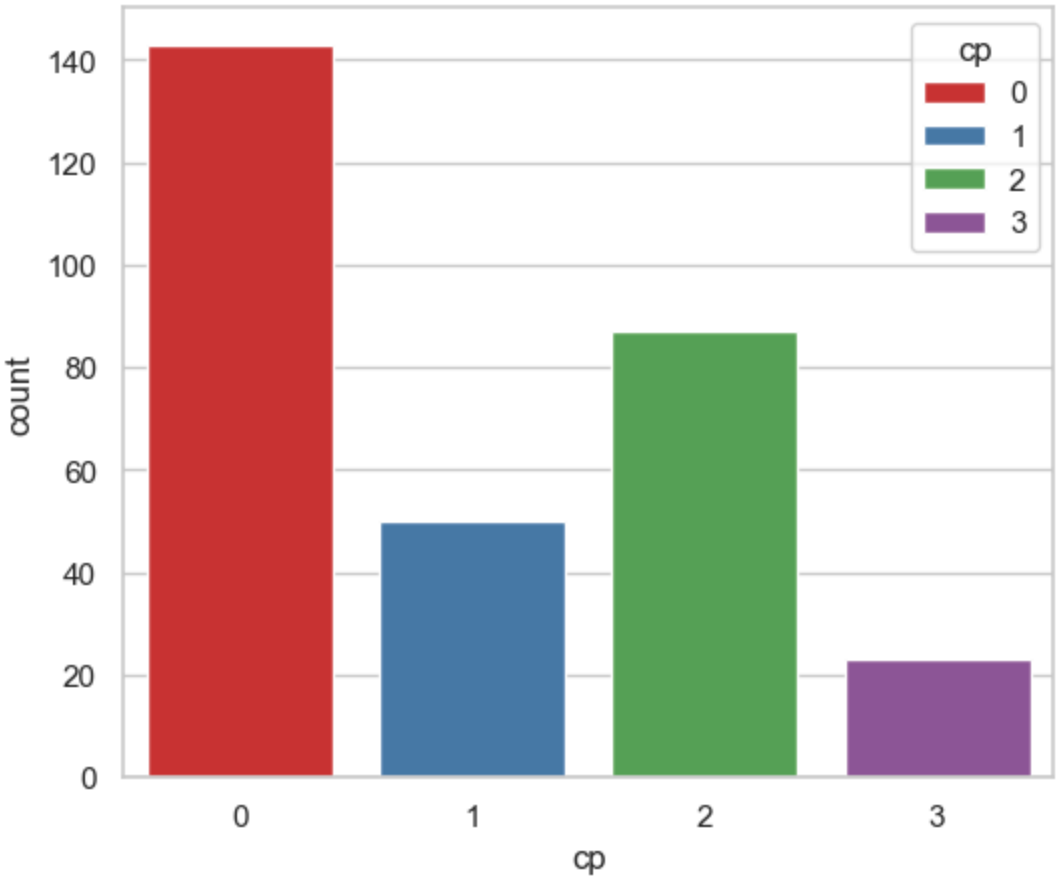
Out[67]:
```
cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
```

Comment

- It can be seen that `cp` is a categorical variable and it contains 4 types of values - 0, 1, 2 and 3.

#Visualize the frequency distribution of `cp` variable

In [70]:
```
f, ax= plt.subplots(figsize=(6,5))
ax = sns.countplot(x='cp',data=df,hue='cp',palette="Set1")
plt.show()
```



Frequency distribution of `target` variable wrt `cp`

In [72]:
```
df.groupby('cp')['target'].value_counts()
```

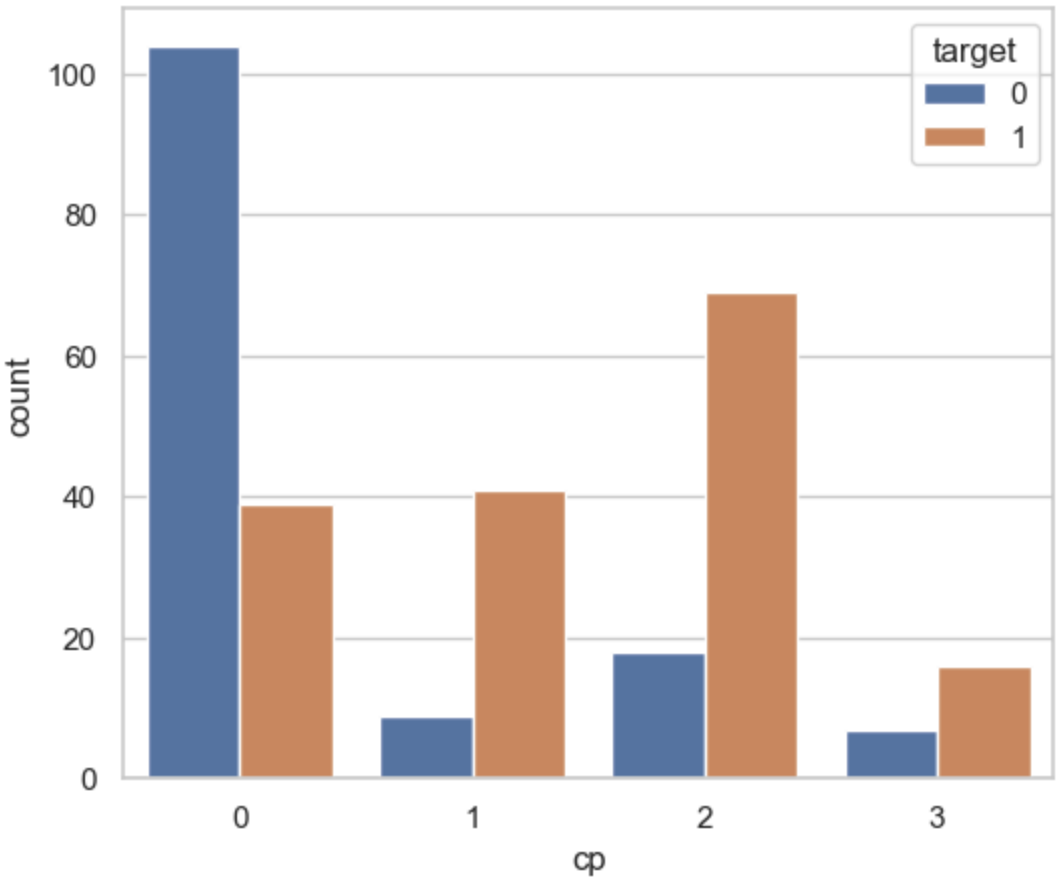
Out[72]:
```
cp  target
0   0         104
    1          39
1   1          41
    0           9
2   1          69
    0          18
3   1          16
    0           7
Name: count, dtype: int64
```

Comment

- `cp` variable contains four integer values 0, 1, 2 and 3.
- `target` variable contains two integer values 1 and 0 : (1 = Presence of heart disease; 0 = Absence of heart disease)
- So, the above analysis gives `target` variable values categorized into presence and absence of heart disease and groupby `cp` variable values.
- We can visualize this information below.

We can visualize the value counts of the `cp` variable wrt `target` as follows -

In [75]:
```
f,ax= plt.subplots(figsize=(6,5))
ax = sns.countplot(x='cp',hue='target',data=df)
plt.show()
```

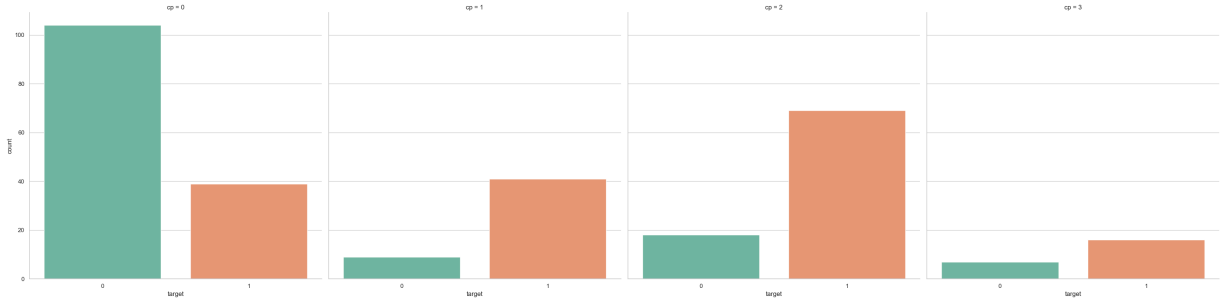


Interpretation

- We can see that the values of `target` variable are plotted wrt `cp`.
- `target` variable contains two integer values 1 and 0 : (1 = Presence of heart disease; 0 = Absence of heart disease)
- The above plot confirms our above findings,

Alternatively, we can visualize the same information as follows :

In [78]:
```
ax = sns.catplot(x='target', col='cp', data=df, kind= "count",height=8,aspect=1,pal
```



Analysis of `target` and `thalach` variable

Explore `thalach` variable

- `thalach` stands for maximum heart rate achieved.
- I will check number of unique values in `thalach` variable as follows :

In [81]:
```
df['thalach'].nunique()
```

Out[81]:
```
91
```

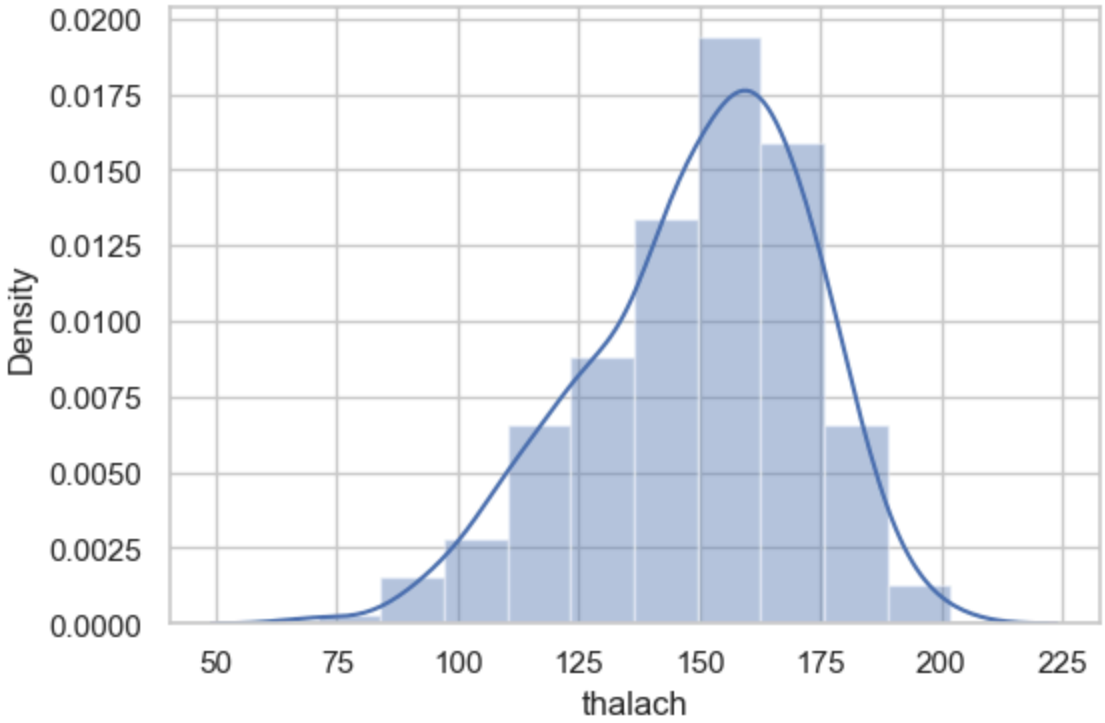
In [82]:
```
df['thalach'].unique()
```

Out[82]:
```
array([150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 171,
       144, 158, 114, 151, 161, 179, 137, 157, 123, 152, 168, 140, 188,
       125, 170, 165, 142, 180, 143, 182, 156, 115, 149, 146, 175, 186,
       185, 159, 130, 190, 132, 147, 154, 202, 166, 164, 184, 122, 169,
       138, 111, 145, 194, 131, 133, 155, 167, 192, 121,  96, 126, 105,
       181, 116, 108, 129, 120, 112, 128, 109, 113,  99, 177, 141, 136,
        97, 127, 103, 124,  88, 195, 106,  95, 117,  71, 118, 134,  90],
      dtype=int64)
```

- So, number of unique values in `thalach` variable is 91. Hence, it is numerical variable.
- I will visualize its frequency distribution of values as follows :

Visualize the frequency distribution of `thalach` variable

In [85]:
```
f, ax = plt.subplots(figsize=(6,4))
x = df['thalach']
ax = sns.distplot(x, bins=10)
plt.show()
```

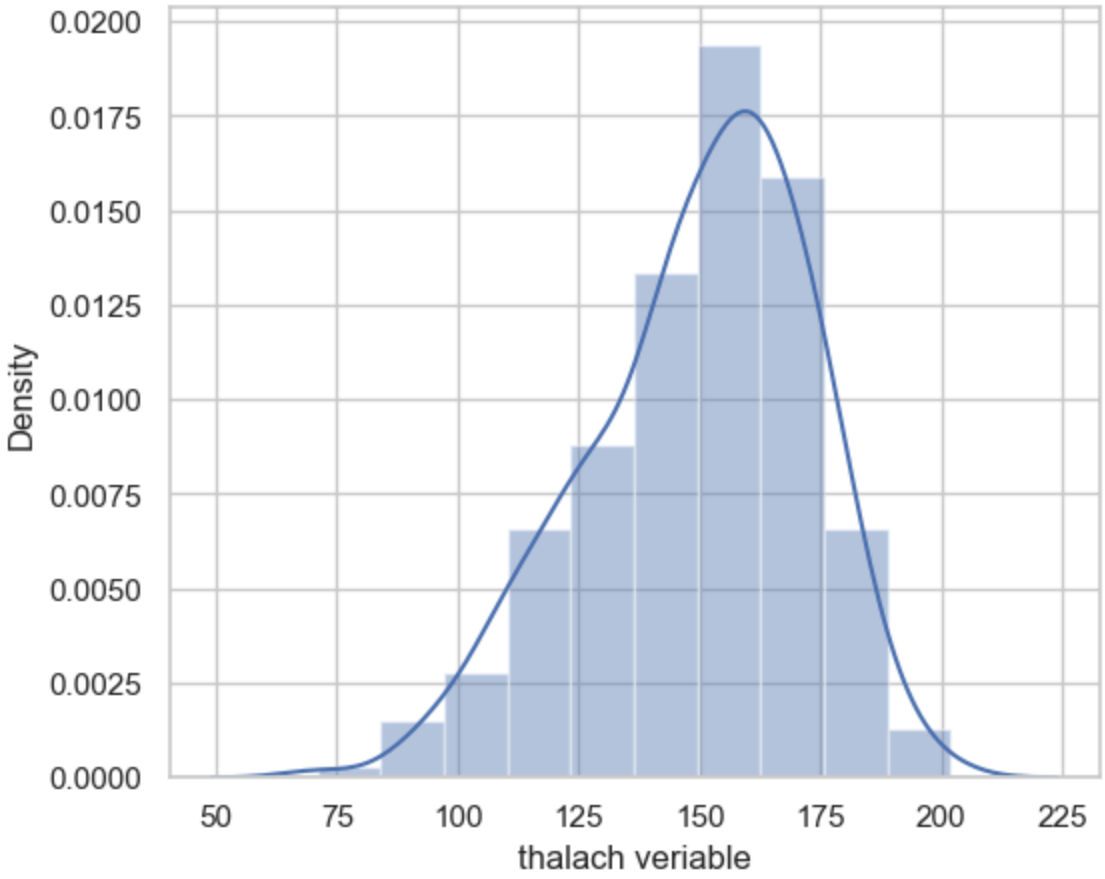


Comment

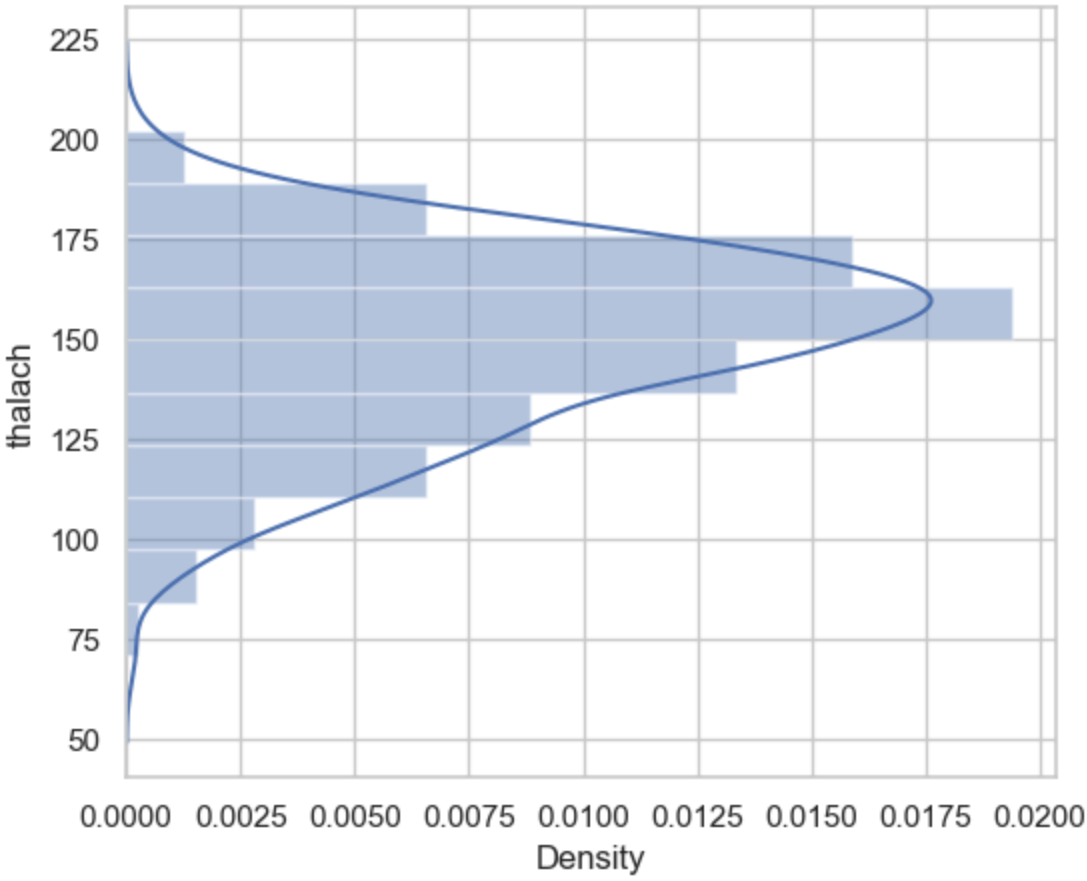
- We can see that the `thalach` variable is slightly negatively skewed.

We can use Pandas series object to get an informative axis label as follows :

In [88]:
```
f ,ax = plt.subplots(figsize=(6,5))
x = df['thalach']
x = pd.Series(x, name="thalach veriable")
ax = sns.distplot(x,bins=10)
plt.show()
```



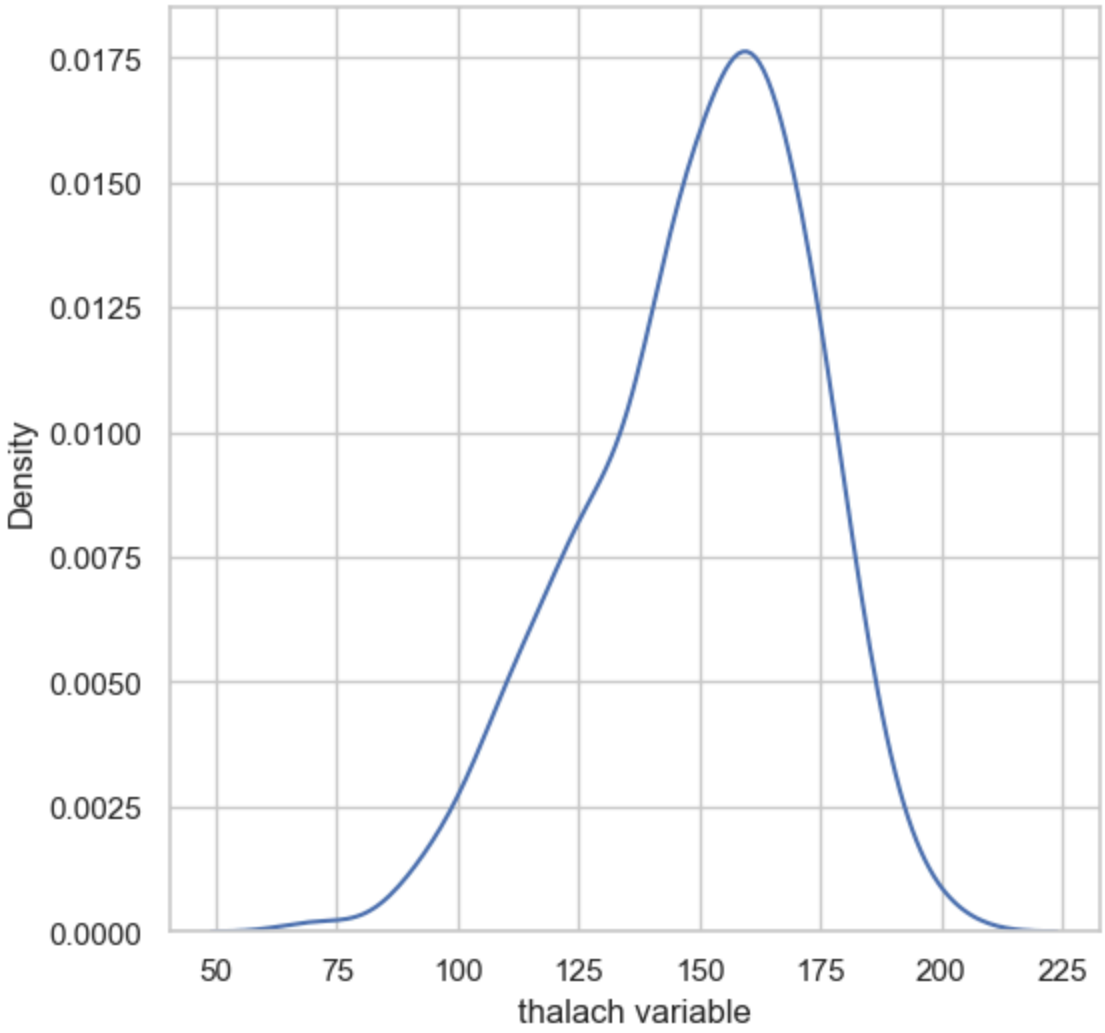
In [89]:
```
f, ax = plt.subplots(figsize=(6,5))
x = df['thalach']
ax = sns.distplot(x, bins=10, vertical=True)
plt.show()
```



Seaborn Kernel Density Estimation (KDE) Plot

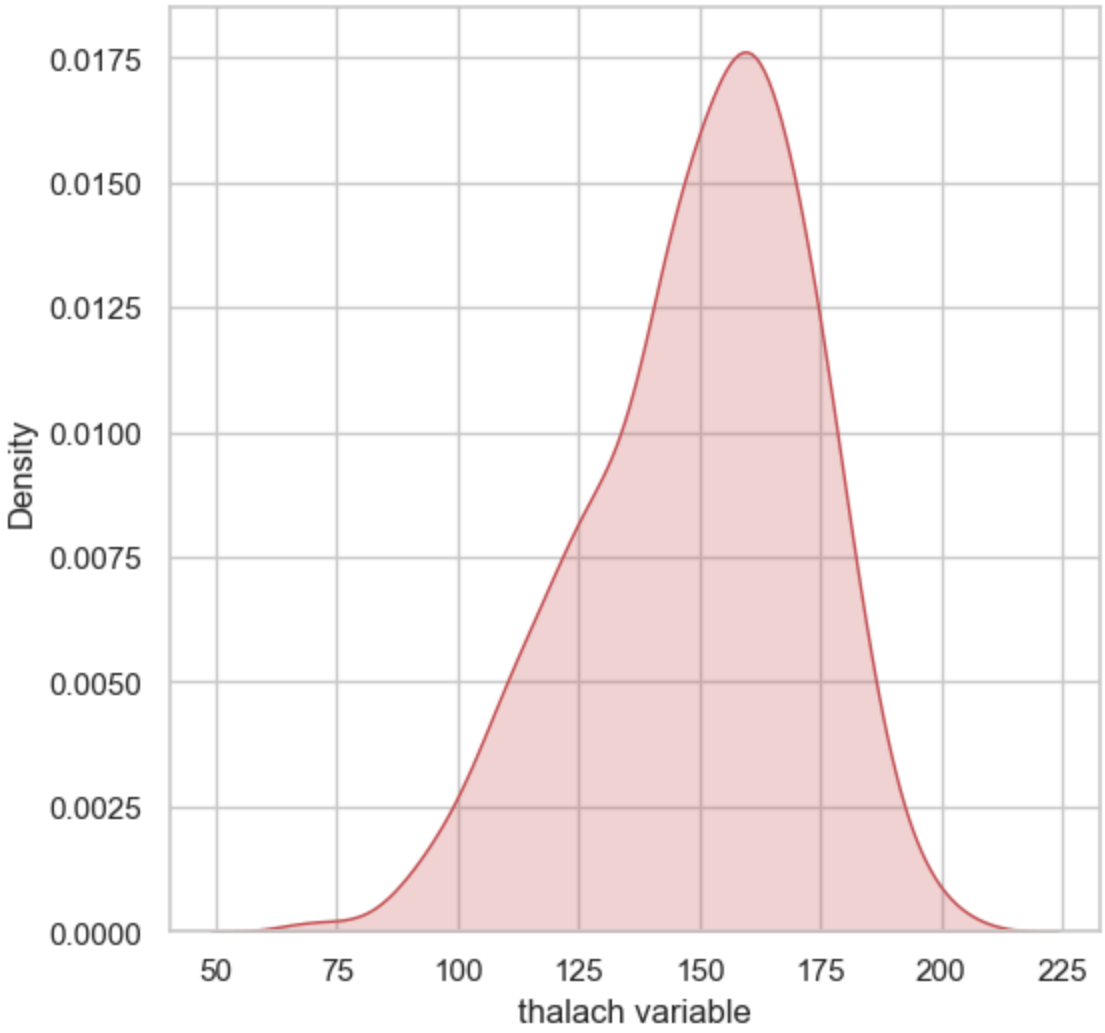
- The kernel density estimate (KDE) plot is a useful tool for plotting the shape of a distribution.
- The KDE plot plots the density of observations on one axis with height along the other axis.
- We can plot a KDE plot as follows :

In [91]:
```
f, ax = plt.subplots(figsize=(6,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.kdeplot(x)
plt.show()
```



We can shade under the density curve and use a different color as follows:

In [93]:
```
f, ax= plt.subplots(figsize=(6,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.kdeplot(x, shade=True, color='r')
plt.show()
```



Histogram

- A histograms represent the distribution of data by forming bins along the range of the data and then drawing bars to show the number of observations that fall in each bin.
- We can plot histogram as follows:
- thalach : maximum heart rate achieved

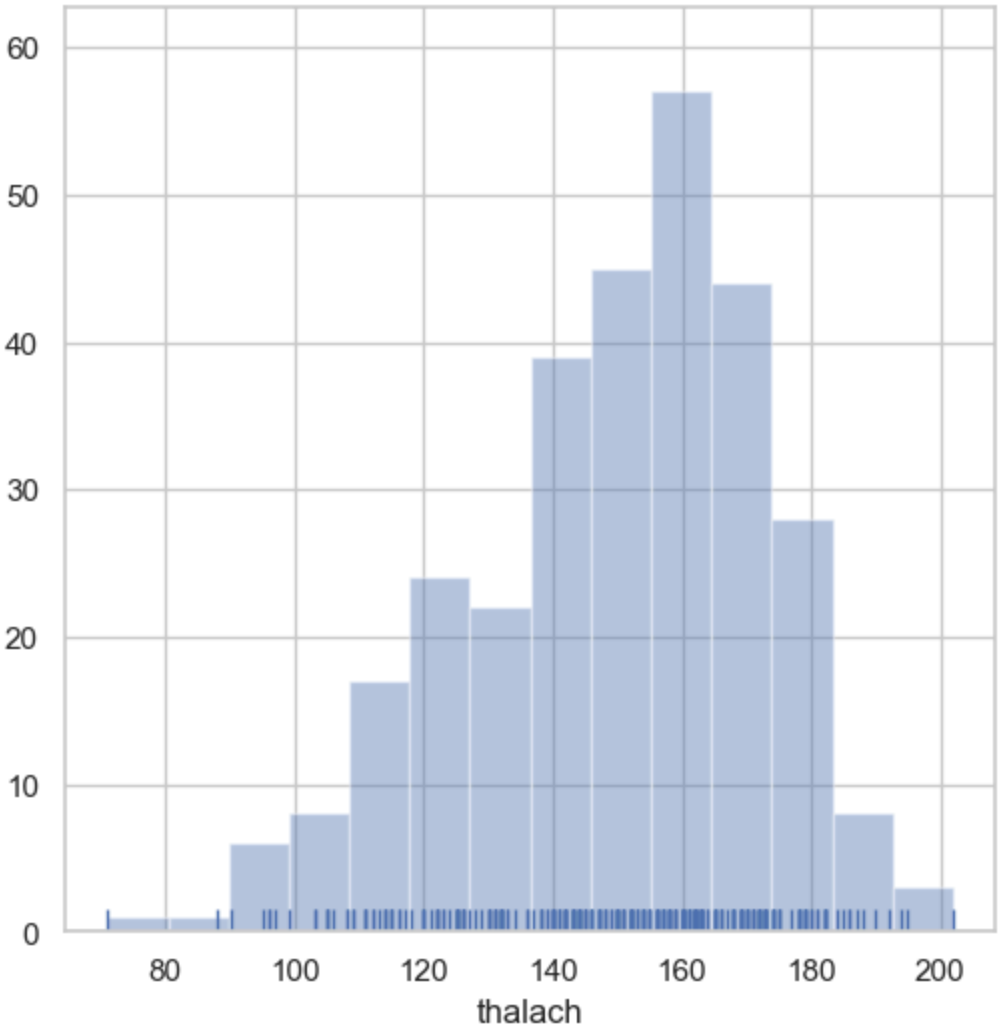
In [95]:
```
df['thalach'].unique()
```

Out[95]:
```
array([150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 171,
       144, 158, 114, 151, 161, 179, 137, 157, 123, 152, 168, 140, 188,
       125, 170, 165, 142, 180, 143, 182, 156, 115, 149, 146, 175, 186,
       185, 159, 130, 190, 132, 147, 154, 202, 166, 164, 184, 122, 169,
       138, 111, 145, 194, 131, 133, 155, 167, 192, 121,  96, 126, 105,
       181, 116, 108, 129, 120, 112, 128, 109, 113,  99, 177, 141, 136,
        97, 127, 103, 124,  88, 195, 106,  95, 117,  71, 118, 134,  90],
      dtype=int64)
```

In [96]:
```
df['thalach'].nunique()
```

Out[96]:
```
91
```

In [98]:
```
f,ax = plt.subplots(figsize=(6,6))
x = df['thalach']
ax = sns.distplot(x,kde=False,rug=True)
plt.show()
```



visualize frequency distribution of `thalach` veriable wrt `target`

In [100…
```
df.groupby('thalach')['target'].value_counts().head(50)
```

Out[100…

thalach	target	
71	0	1
88	0	1
90	0	1
95	0	1
96	0	1
	1	1
97	0	1
99	0	1
103	0	2
105	0	2
	1	1
106	0	1
108	0	2
109	0	2
111	0	2
	1	1
112	0	2
113	0	1
114	0	2
	1	1
115	1	2
	0	1
116	0	1
	1	1
117	0	1
118	0	1
120	0	3
121	1	1
122	1	3
	0	1
123	0	1
	1	1
124	0	1
125	0	5
	1	2
126	0	3
	1	1
127	0	1
128	0	1
129	0	1
130	0	3
	1	1
131	0	2
	1	2
132	0	6
	1	1
133	0	1
	1	1
134	0	1
136	0	2

Name: count, dtype: int64

- 71 (thalach) and 0 (target): There is 1 person with a heart rate of 71 and no heart disease.

- 96 (thalach):

target = 0: There is 1 person with no heart disease. target = 1: There is 1 person with heart disease.

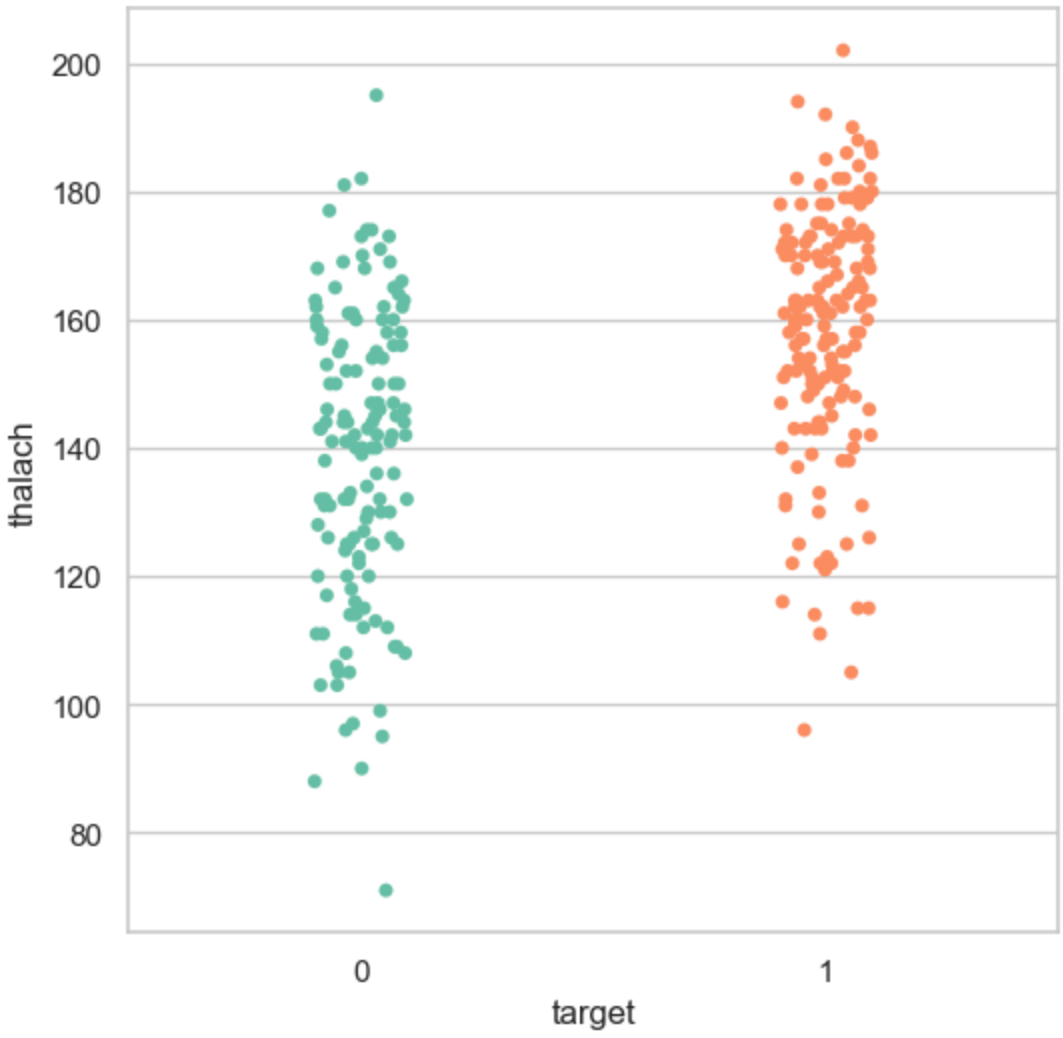
- 103 (thalach):

target = 0: There are 2 people with a heart rate of 103 and no heart disease.

- 105 (thalach):

target = 0: 2 people with no heart disease. target = 1: 1 person with heart disease.

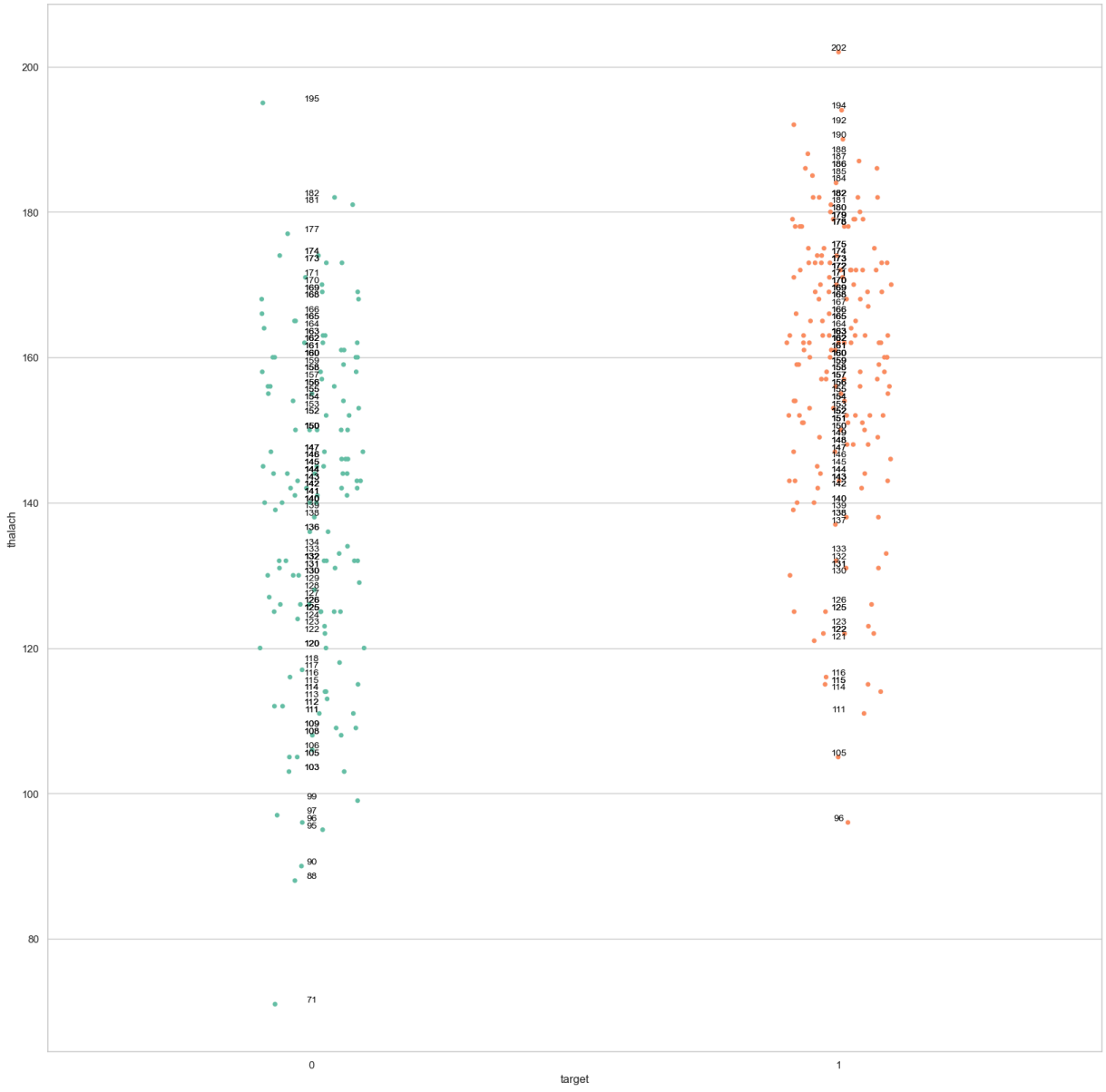
In [102…
```
f,ax = plt.subplots(figsize=(6,6))
sns.stripplot(x='target',y='thalach',data=df,palette='Set2')
plt.show()
```



In [ ]:
```
```

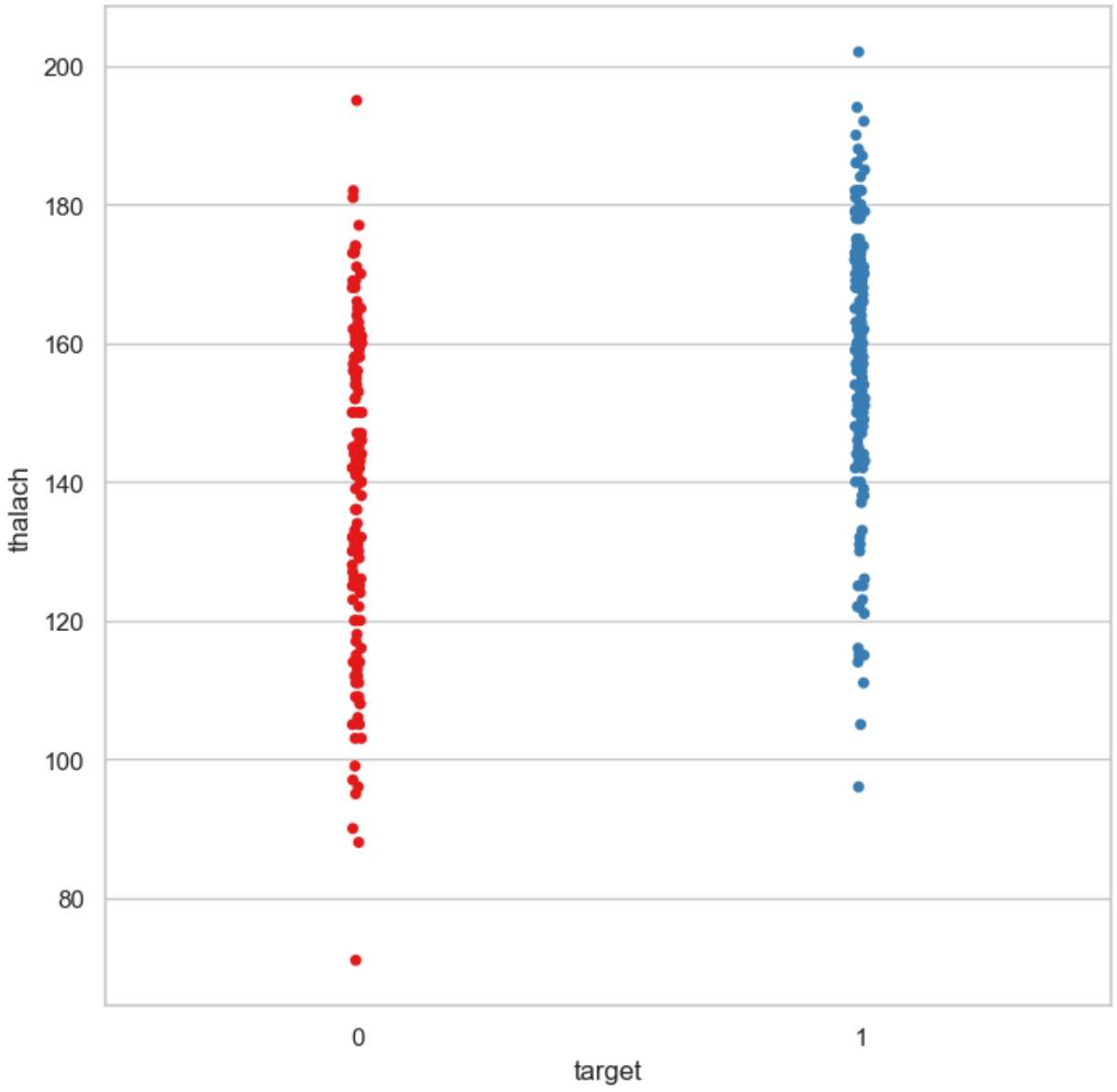
In [103…
```
f,ax = plt.subplots(figsize=(20,20))

sns.stripplot(x='target',y='thalach',data=df,palette='Set2')
```

```
for i in range(len(df)):
    ax.text(df['target'].iloc[i], df['thalach'].iloc[i],
            str(df['thalach'].iloc[i]),
            color='black', ha='center', va='bottom', fontsize=10)
plt.show()
```



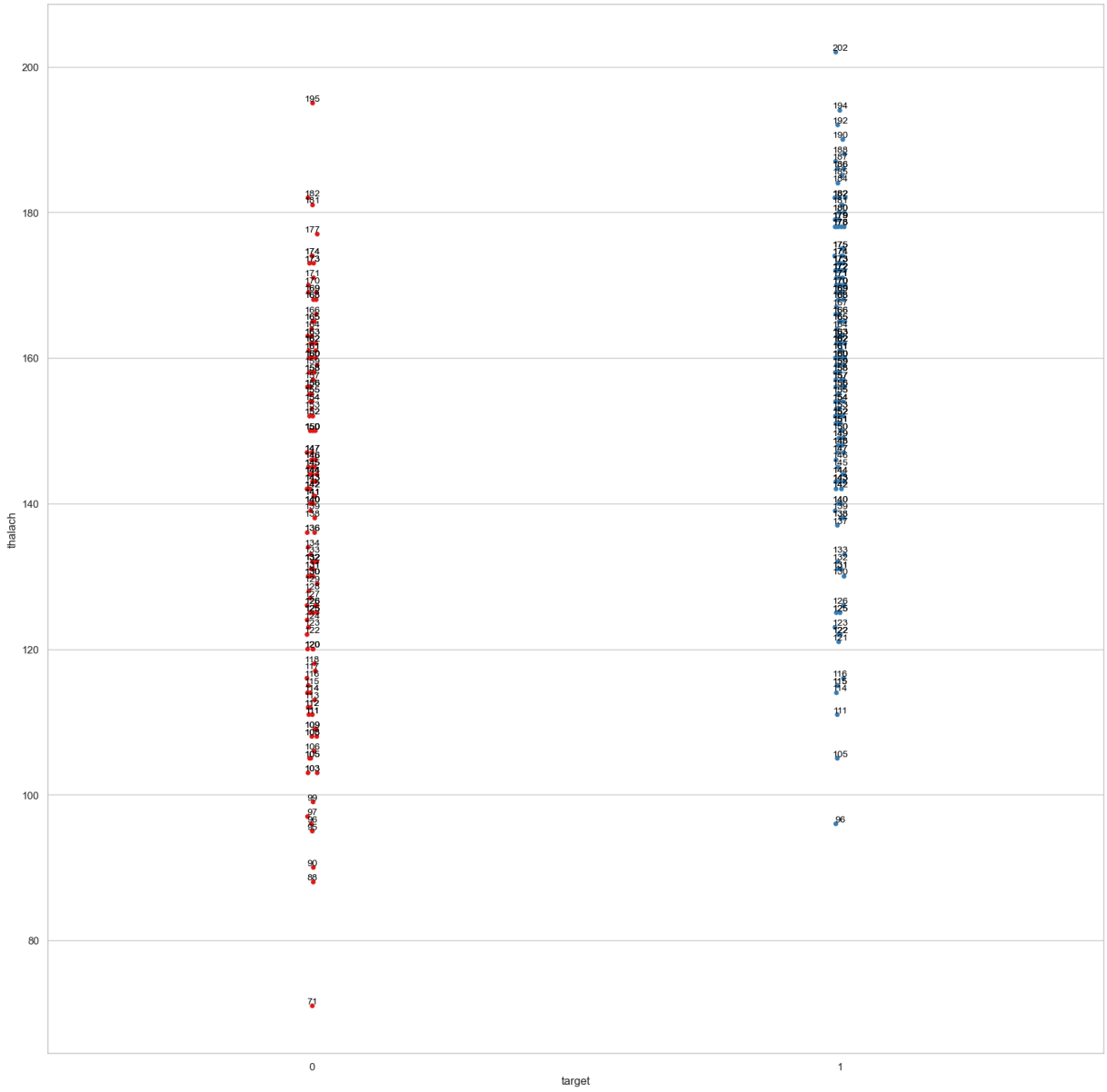
In [104…

```
f, ax= plt.subplots(figsize=(8,8))
sns.stripplot(x='target',y='thalach',data = df,palette='Set1',jitter=0.01)
plt.show()
```



In [105…

```
f, ax= plt.subplots(figsize=(20,20))
sns.stripplot(x='target',y='thalach',data = df,palette='Set1',jitter=0.01)
for i in range(len(df)):
    ax.text(df['target'].iloc[i], df['thalach'].iloc[i],
            str(df['thalach'].iloc[i]),
            color='black', ha='center', va='bottom', fontsize=10)
plt.show()
```



#Visualize distribution of `thalach` variable wrt `target` with boxplot

In [107…

```
f,ax =plt.subplots(figsize=(6,6))
sns.boxplot(x='target',y='thalach',data=df,palette='Set1')
plt.show()
```