In [ ]:

```
'''Heart Dataset Analysis

The goal of this dataset is to train a model so that it predicts
whether a person is likely to suffer from heart disease (whether the probability is above or below 50%);
however, in this exercise/project, we are simply going to observe and analyze the distribution of the data,
search for outliers and missing values, and assess the relationships between features.

REQUIRED:

check for outliers, missing values, and the trends and
relationships between different features of the dataset to gain a better understanding of the

available data and derive useful insights from it.

'''
```

In [ ]:

```
'''
*Loading and Understanding the Data

*Dealing with Outliers

*Plotting the Distributions and Relationships Between Specific Features

*Plotting Distributions and Relationships between Columns with Respect to the Target Column

*Plotting the Relationship between the Presence of Heart Disease and Maximum Recorded Heart Rate

*Plotting the Relationship between the Presence of Heart Disease and the Cholesterol Column

*Observing Correlations with a Heatmap

'''
```

In [424]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
Path = '/Users/tomisin/Dropbox/My Mac (Tomisins-MacBook-Pro.local)/Documents/MY WORKSPACE/Database/heart.csv
df = pd.read_csv(Path)
```

In [425]:
```
df=df.rename(columns={'age':'AGE','trestbps':'TRestBPS', 'sex':'SEX','restecg':'restECG', 'chol':'CHOL','tha
```

In [426]:
```
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 [427]:
```
1 df.describe()
```

Out[427]:

	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	303.000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000

In [428]:
```
#INFORMATION ABOUT THE DATASET

'''


*** df.info() shows the structure of this dataset is one that has 303 samples, a total of 14 variables
with all non-categorical features

*** df.describe() shows if the data in each variable normally distributed (gaussian curve) or not.
The data in this data set is generally skewed except for the CP feature.

*** There are no missing values in this dataset


'''
```

Out[428]:
```
'\n\n\n*** df.info() shows the structure of this dataset is one that has 303 samples, a total of 14 variables
\nwith all non-categorical features\n\n*** df.describe() shows if the data in each variable normally distribu
ted (gaussian curve) or not. \nThe data in this data set is generally skewed except for the CP feature.\n\n\n
\n\n'
```

In [429]:

```
'''There are no missing values in this dataset. df.info(), df.describe() tells me so'''


#Dealing with Outliers


'''There are certainly outliers in some of the features.'''
#In 'age', there is no outlier, minimum and maximum ages are 29 and 77 respectively.
#Data sample was taken amongst adults especially the older ones.

#In 'sex', there is/are no outlier, sex can either be 1 or 0 (male of female).

#In 'cp', there are no outliers, 4 data points in the density plot

#In 'trestbps', the presence of outliers indicating values between 170 till 200(max value)
#does not cause skewness in the data. The density plot using matplotlib appears quite normally distributed
#except for the hunches at these values which represent the outliers.

#In 'chol', presence of outliers indicating values between 350 till 564(max value).
#does not cause skewness in the data.
#The density plot using matplotlib appears quite normally distributed
#except for the hunches at these values which represent the outliers.

#In 'fbs', we have more zeros (250) than ones (53) making 1 an outlier,
#The ones should not be removed from this column in my opinion.
#The zeros and ones definitely infer something. Hopefully, the right readings were collected and reported.

#In 'restecg', presence of no outlier, 3 data points in the density plot

#In 'thalach', presence of an outliers indicating values between 71 and 120.
#Should not be removed assuming the right readings were collected and reported

#In 'exang', there are no outliers, 2 data points in the density plot.

#In 'oldpeak', presence of outliers between 4 and 6.2.

#In 'slope', there are no outliers, 3 data points in the density plot

#In 'ca', presence of outliers between 3 and 4.
#4 data points in the density plot.
#The outliers should not be removed assuming the right readings were collected and reported.
```

```
42
43 #In 'thal', presence of outlier representing the 0 input
44 #The zeros should not be removed assuming the right readings were collected and reported.
```

Out[429]: 'There are certainly outliers in some of the features.'

In [430]:

```

#UNIVARATE ANALYSIS

'''
*** For 'AGE' variable, the distribution shows participants from the ages 29-77.
It is likely that both young and old adults within this age range and prone to develop a heart disease
even though we do not have so many participants from ages (29-40).

*** For 'SEX' variable, there are more males than females.
Either of these genders can be prone to have a heart disease

*** For 'CP' variable, majority of the participants (246) have an angina-related chest pain.
I would want to think that angina is one of the symptom of a diseased heart.
This could also mean that some of those with non-angina chest pain may also have a diseased heart.

*** For 'TRestBPS' variable, A total of 243 participants have either
borderline high blood pressure or high blood pressure which already exceeds
the total number of participants with a diseased heart (165).
This could also mean that some of those with normal BP may also have a diseased heart
due to the iinfluence of another variable in this dataset.

*** For 'CHOL', a total of 252 participants have either
borderline high levels or high levels of cholesterol in their blood.
There is a likelihood that most of those with a diseased heart have excess cholesterol in their blood.
However, if some or most of the rest of the participants with normal cholesterol levels have a diseased hea
it may be due to the influence of another variable in this dataset.

*** For 'FBS', 258 out of the 303 participants have sugar in their blood.
Majority of reported diseased heart participants will have high amounts sugar in their blood.
However if majority of the remaining 45 participants have a diseased heart,
Depending on the proportion of those with a normal blood sugar levels who have a diseased heart
FBS may not be an important feature in the model design.

*** For 'restECG', we have like a 1:1 of those with a normal and an abnormal ECG.
Depending on the proportion of those with a normal ECG who have a diseased heart
restECG may not be an important feature in the model design.

*** For 'Thalach', an occurence of varying heart rates at its maximum ranging from 77-202.
HIgher maximum heart rates in a patient would mean the presence of Thalassemia disease.
Depending on the proportion of those with a normal Thalach who have a diseased heart
Thalach may not be an important feature in the model design.
```

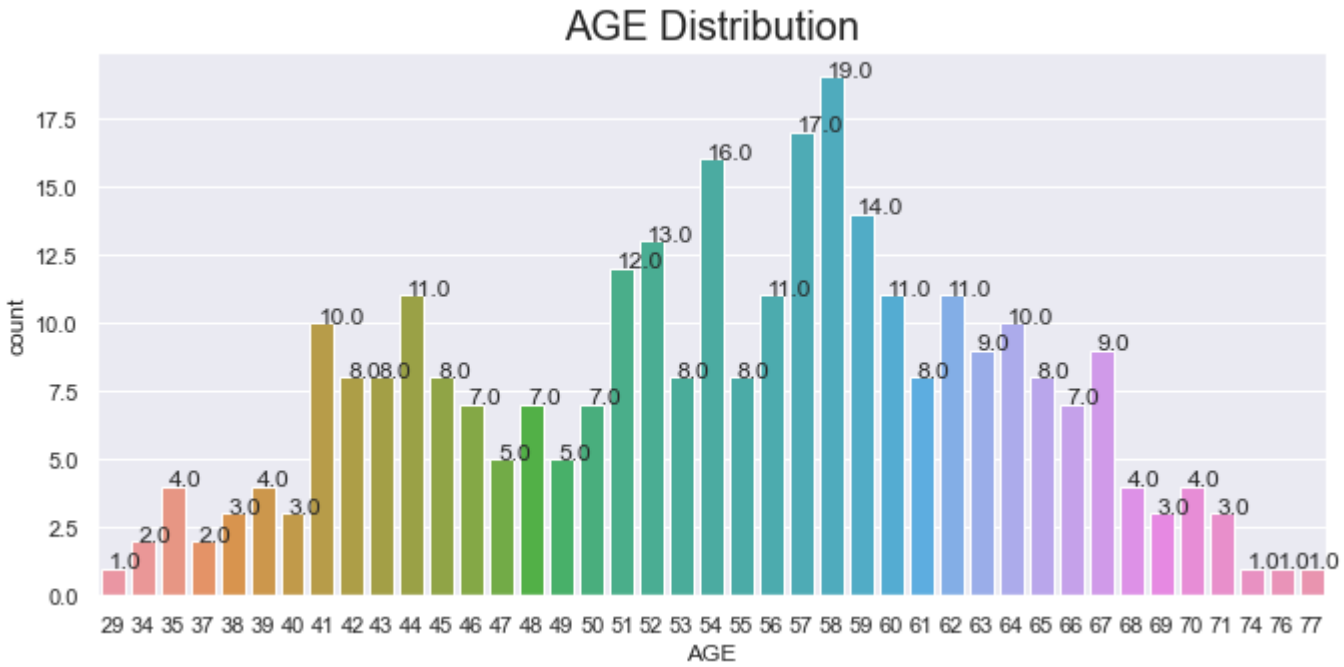
```
42
43 *** For 'EXANG', one of these two types of angina (non-induced and induced angina)
44 are seen in all the participants. Either of them can be a sign of a heart disease.
45 This variable may likely be an important one in model design.
46
47 *** For 'CA', 175 participants have no coloured artery, the rest have at least 1.
48 Depending on the proportion of those with a no colored artery who have a diseased heart
49 CA may not be an important feature in the model design.
50
51 *** For 'THAL', 283 people have fixed and reversible defect, the remaining 20 have no defect.
52 This would mean that some of those who have a fixed and reversible defect have a heart disease.
53
54 *** For 'TARGET', 165 have a heart disease and 138 do not. More like 1:!.
55 With most of the features having majority of the participant on the extremities (positive or negative)
56 May be difficult to decide which of them to keep or do away with
57
58
59
60
61
62 '''
```

Out[430]: "\n*** For 'AGE' variable, the distribution shows participants from the ages 29-77. \nIt is likely that both young and old adults within this age range and prone to develop a heart disease\neven though we do not have so many participants from ages (29-40).\n\n*** For 'SEX' variable, there are more males than females. \nEither of these genders can be prone to have a heart disease \n\n*** For 'CP' variable, majority of the participants (246) have an angina-related chest pain. \nI would want to think that angina is one of the symptom of a diseased heart. \nThis could also mean that some of those with non-angina chest pain may also have a diseased heart.\n\n*** For 'TRestBPS' variable, A total of 243 participants have either \nborderline high blood pressure or high blood pressure which already exceeds \nthe total number of participants with a diseased heart (165). \nThis could also mean that some of those with normal BP may also have a diseased heart \ndue to the iinfluence of another variable in this dataset.\n\n*** For 'CHOL', a total of 252 participants have either \nborderline high levels or high levels of cholesterol in their blood. \nThere is a likelihood that most of those with a diseased heart have excess cholesterol in their blood. \nHowever, if some or most of the rest of the participants with normal cholesterol levels have a diseased heart, \nit may be due to the influence of another variable in this dataset.\n\n*** For 'FBS', 258 out of the 303 participants have sugar in their blood. \nMajority of reported diseased heart participants will have high amounts sugar in their blood.\nHowever if majority of the remaining 45 participants have a diseased heart, \nDepending on the proportion of those with a normal blood sugar levels who have a diseased heart\nFBS may not be an important feature in the model design.\n\n*** For 'restECG', we have like a 1:1 of those with a normal and an abnormal ECG.\nDepending on the proportion of those with a normal ECG who have a diseased heart\nrestECG may not be an important feature in the model design.\n\n*** For 'Thalach', an occurence of varying heart rates at its maximum ranging from 77-202. \nHIgher max

```
imum heart rates in a patient would mean the presence of Thalassemia disease. \nDepending on the proportion o
f those with a normal Thalach who have a diseased heart\nThalach may not be an important feature in the model
design. \n\n*** For 'EXANG', one of these two types of angina (non-induced and induced angina) \nare seen in
all the participants. Either of them can be a sign of a heart disease. \nThis variable may likely be an impor
tant one in model design.\n\n*** For 'CA', 175 participants have no coloured artery, the rest have at least
1. \nDepending on the proportion of those with a no colored artery who have a diseased heart\nCA may not be a
n important feature in the model design.\n\n*** For 'THAL', 283 people have fixed and reversible defect, the
remaining 20 have no defect. \nThis would mean that some of those who have a fixed and reversible defect have
a heart disease.\n\n*** For 'TARGET', 165 have a heart disease and 138 do not. More like 1:!. \nWith most of
the features having majority of the participant on the extremities (positive or negative)\nMay be difficult t
o decide which of them to keep or do away with\n\n\n\n\n\n"
```

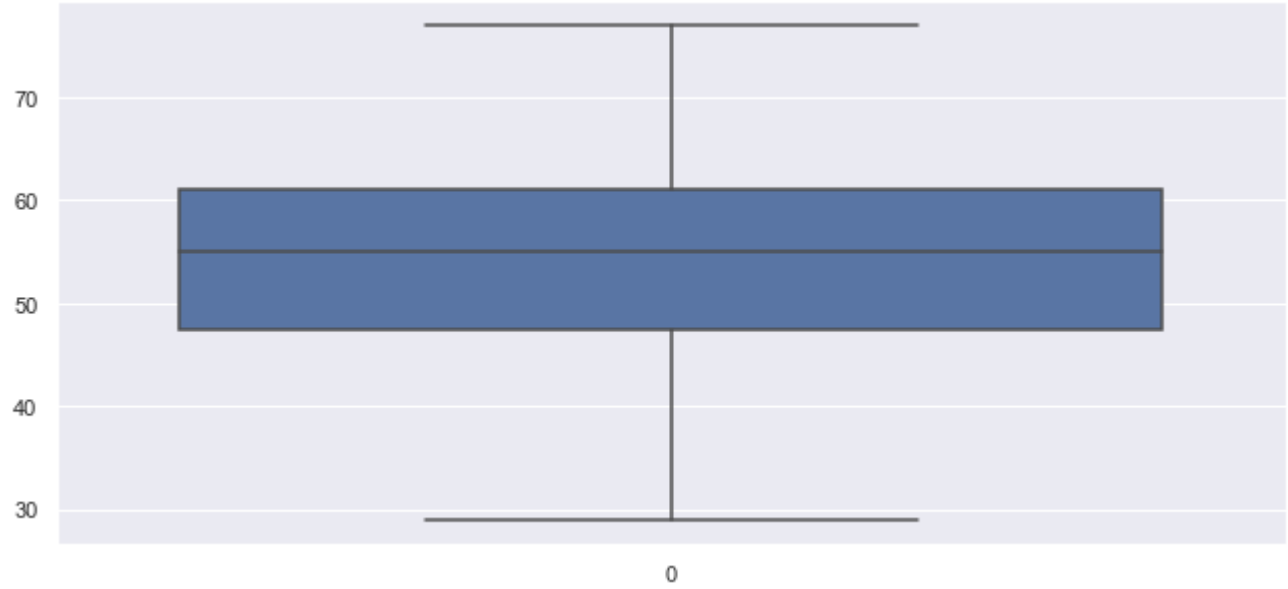
In [433]:

```
ax = sns.countplot(x="AGE", data=df)
ax.set_title('AGE Distribution', fontsize=20)

'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
sns.set(rc={'figure.figsize':(11,5)})
```



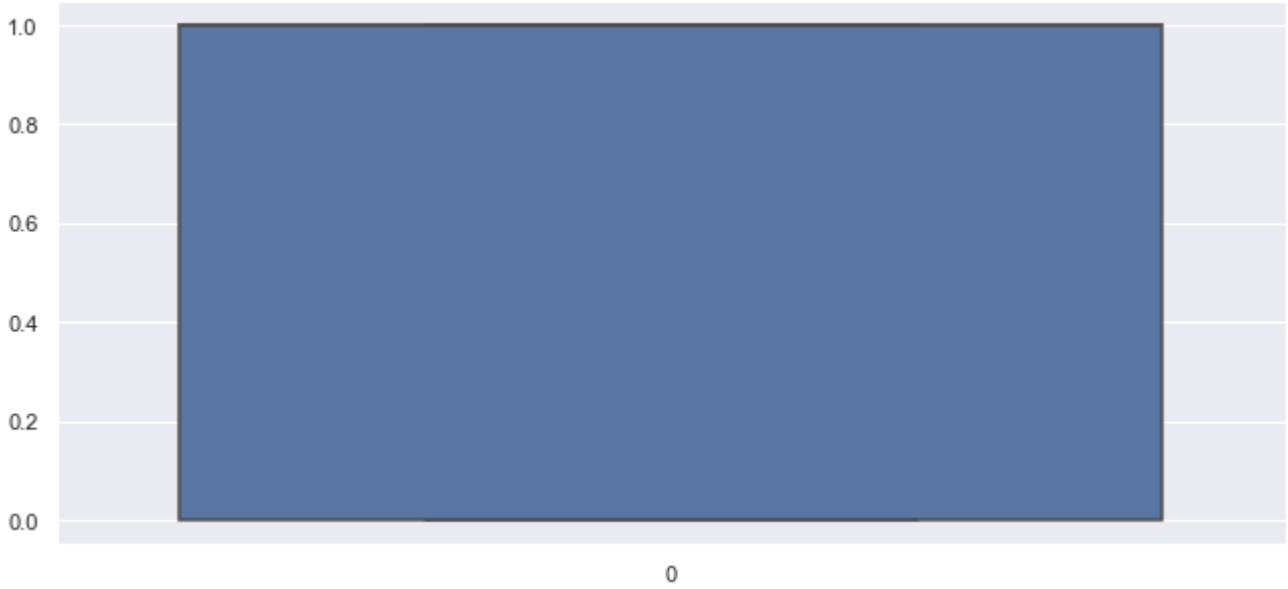
In [434]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['AGE'] )
```

Out[434]: <AxesSubplot:>



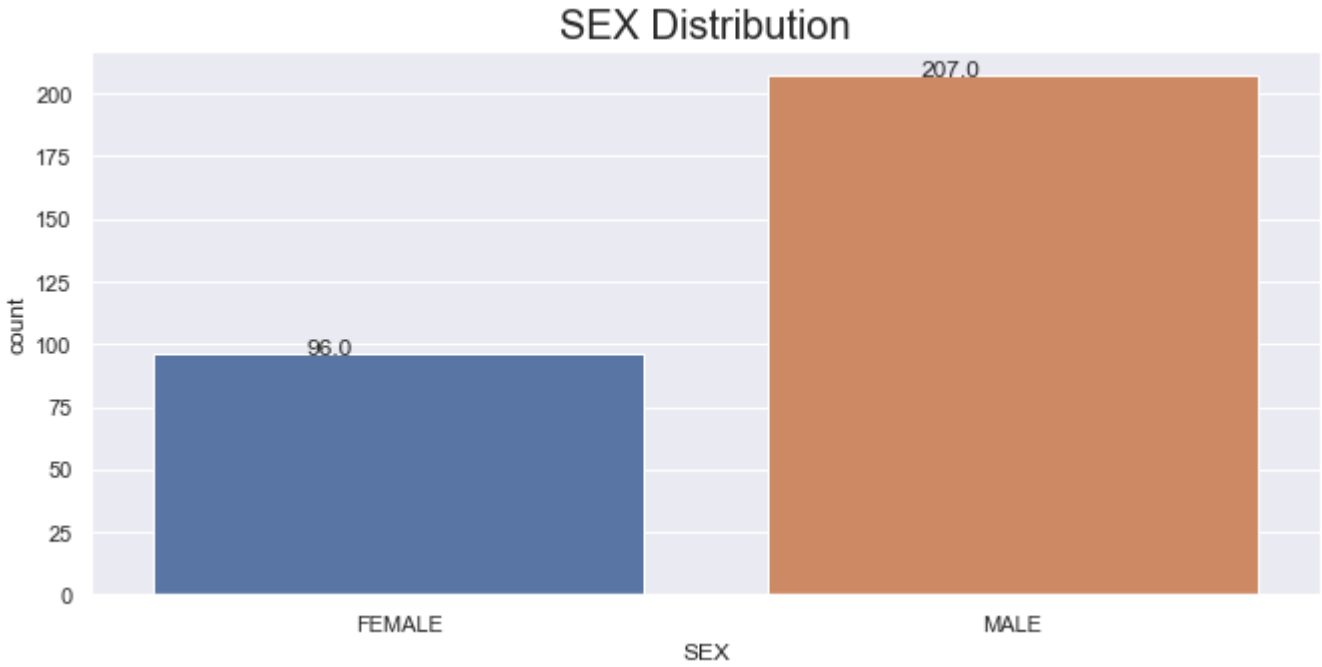
In [435]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['SEX'] )
```

Out[435]: <AxesSubplot:>



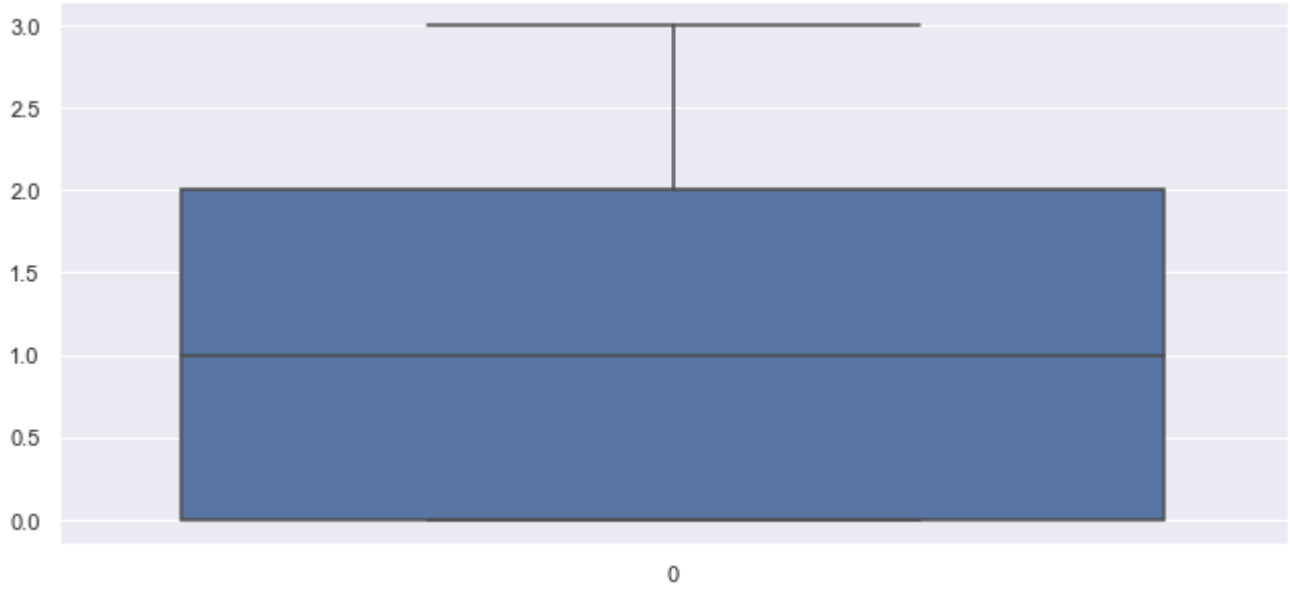
In [436]:

```
ax = sns.countplot(x="SEX", data=df)
ax.set_title('SEX Distribution', fontsize=20)
ax.set_xticklabels(['FEMALE','MALE'])
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```



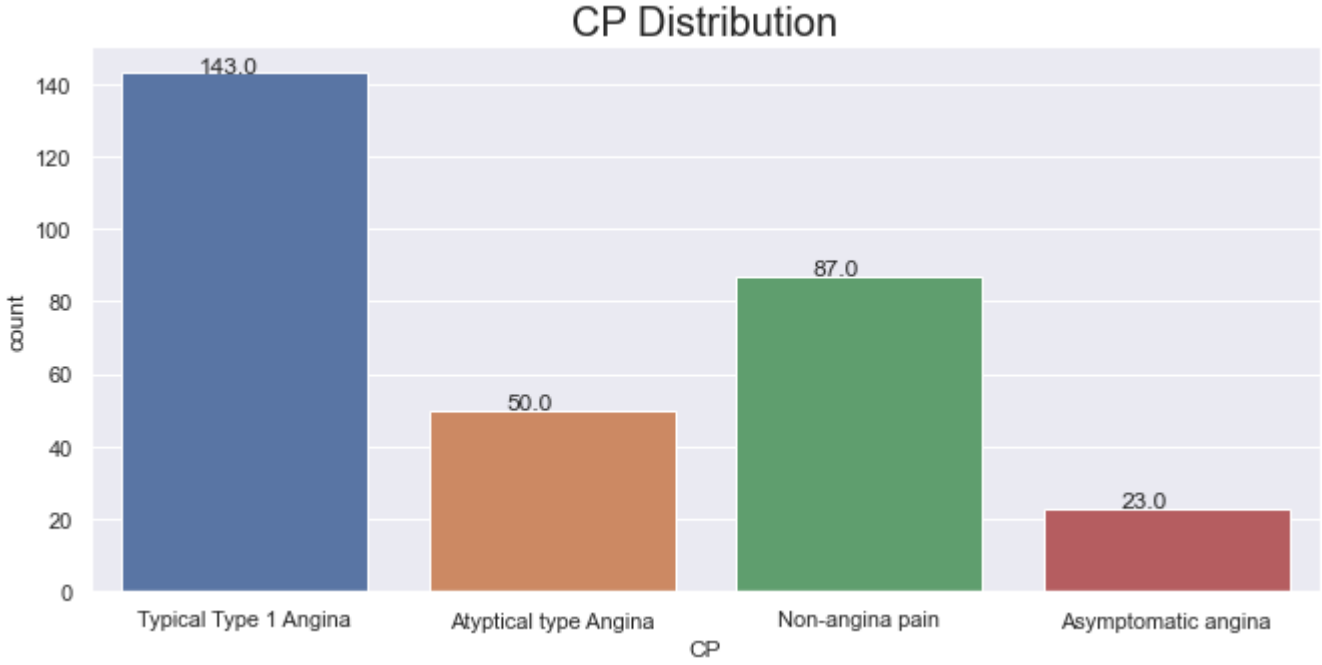
In [437]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['CP'])
```

Out[437]: <AxesSubplot:>



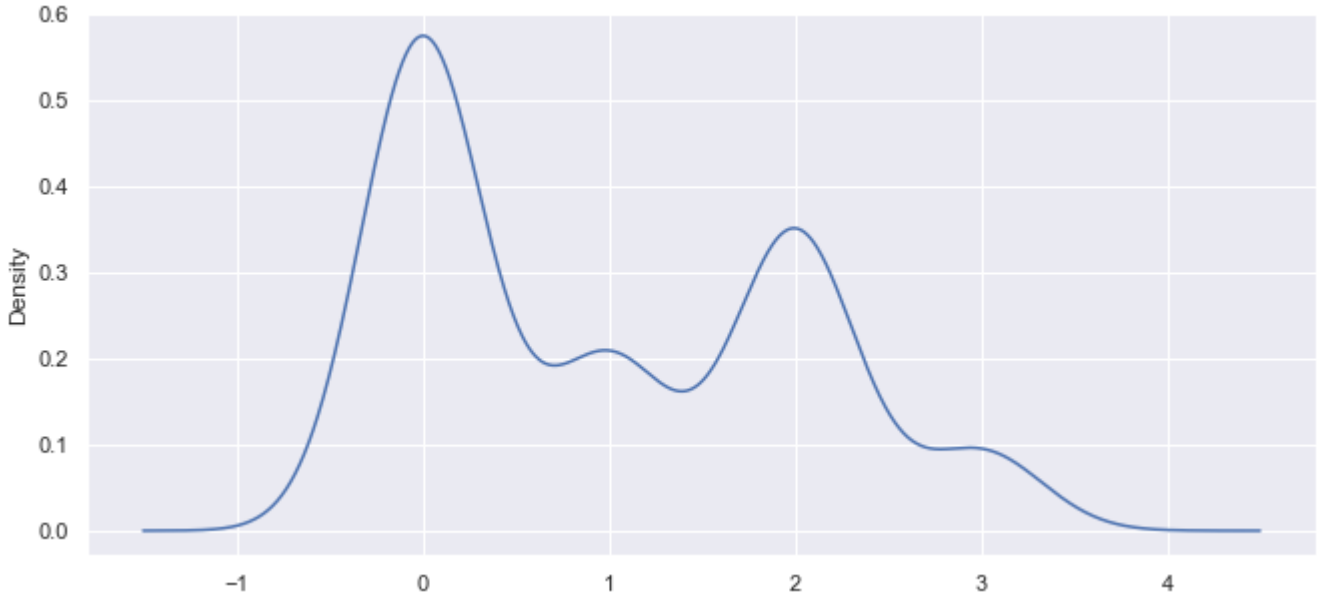
In [438]:

```
ax = sns.countplot(x="CP", data=df)
ax.set_title('CP Distribution', fontsize=20)
ax.set_xticklabels(['Typical Type 1 Angina','Atyptical type Angina','Non-angina pain','Asymptomatic angina'
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```



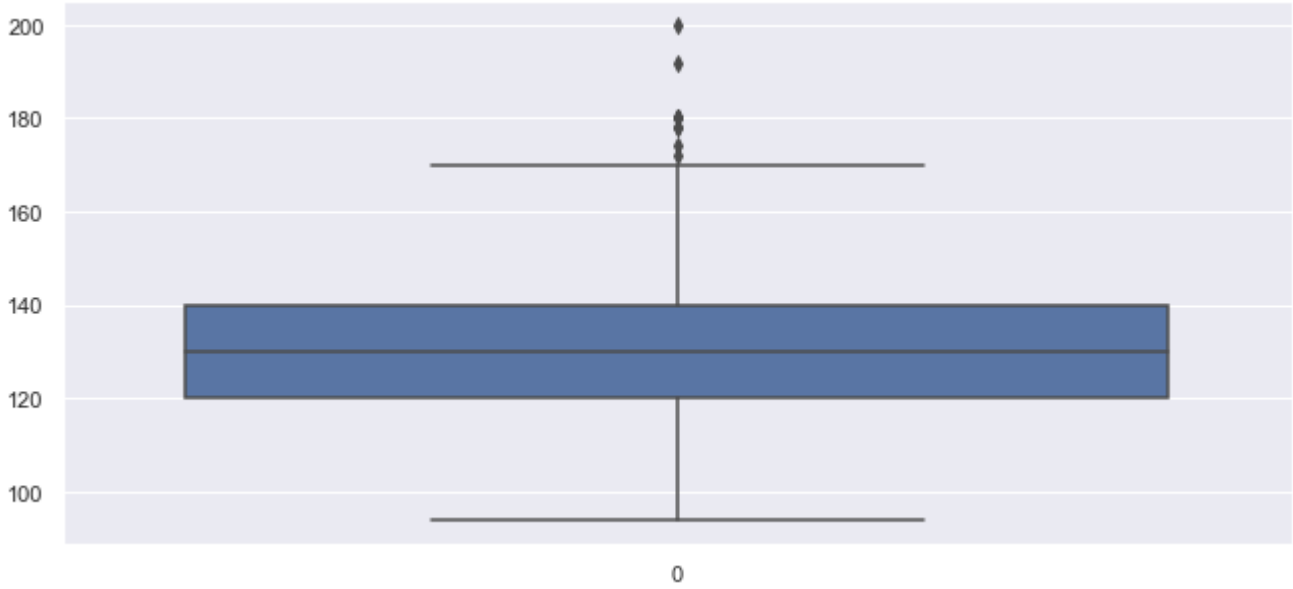
In [439]:
```
1 df['CP'].plot(kind = 'density')
```

Out[439]: <AxesSubplot:ylabel='Density'>



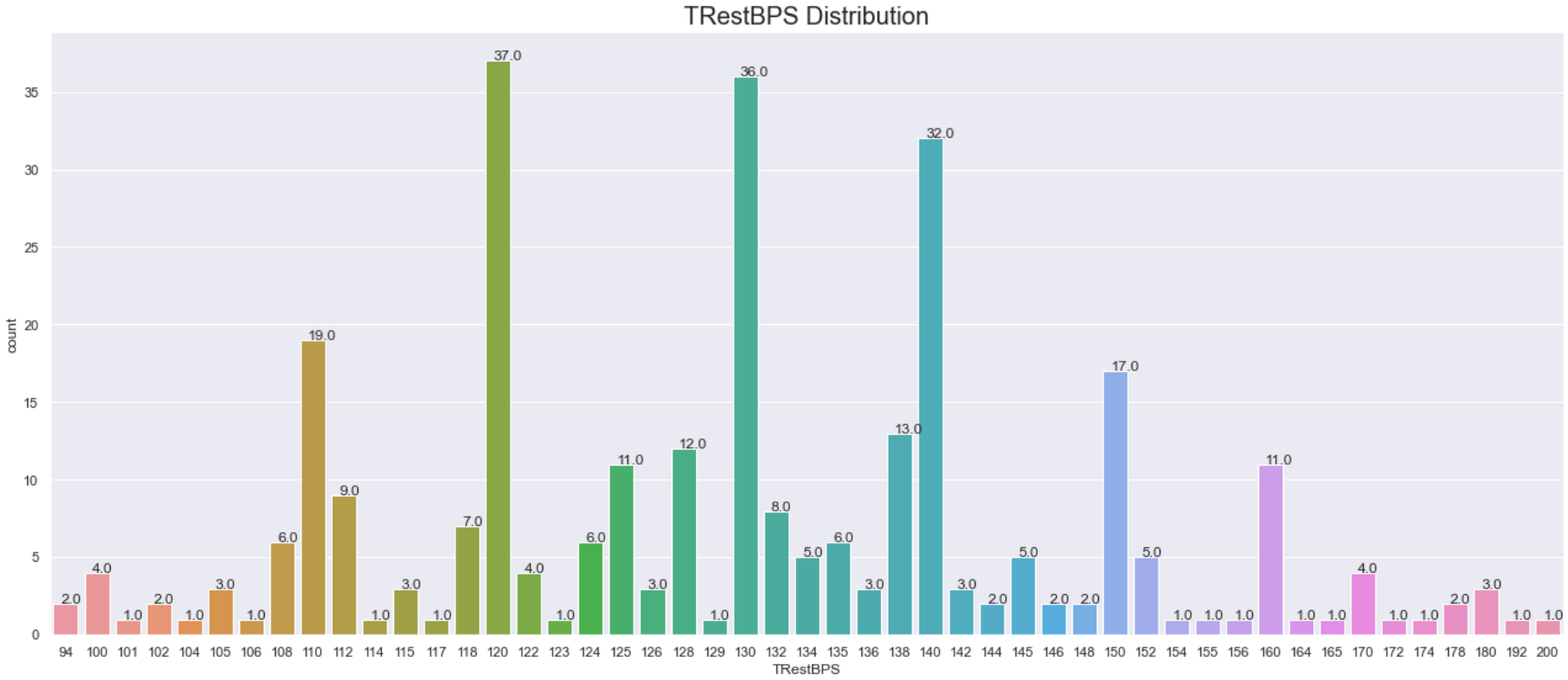
In [440]:
```
import seaborn as sb
sb.boxplot(data=df['TRestBPS'])


```

Out[440]: <AxesSubplot:>



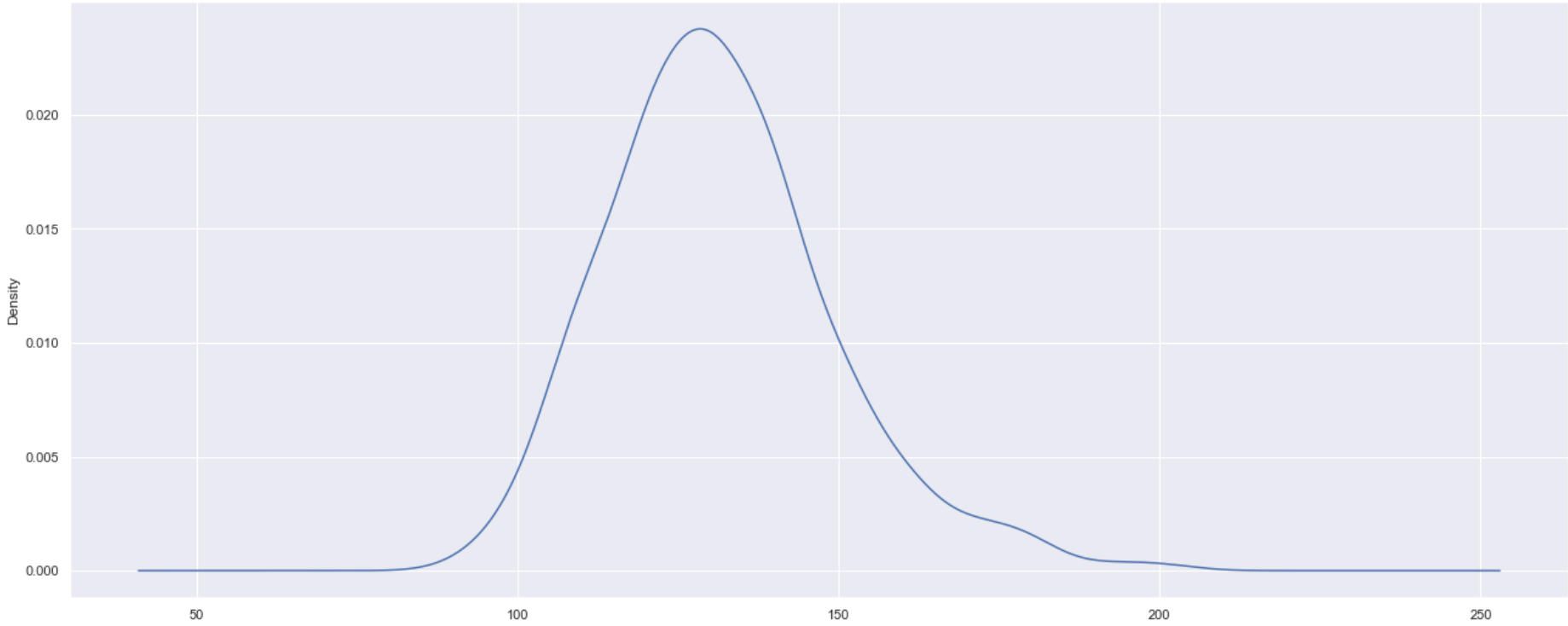
In [443]:

```
ax = sns.countplot(x="TRestBPS", data=df)
ax.set_title('TRestBPS Distribution', fontsize=20)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
sns.set(rc={'figure.figsize':(22,9)})

```



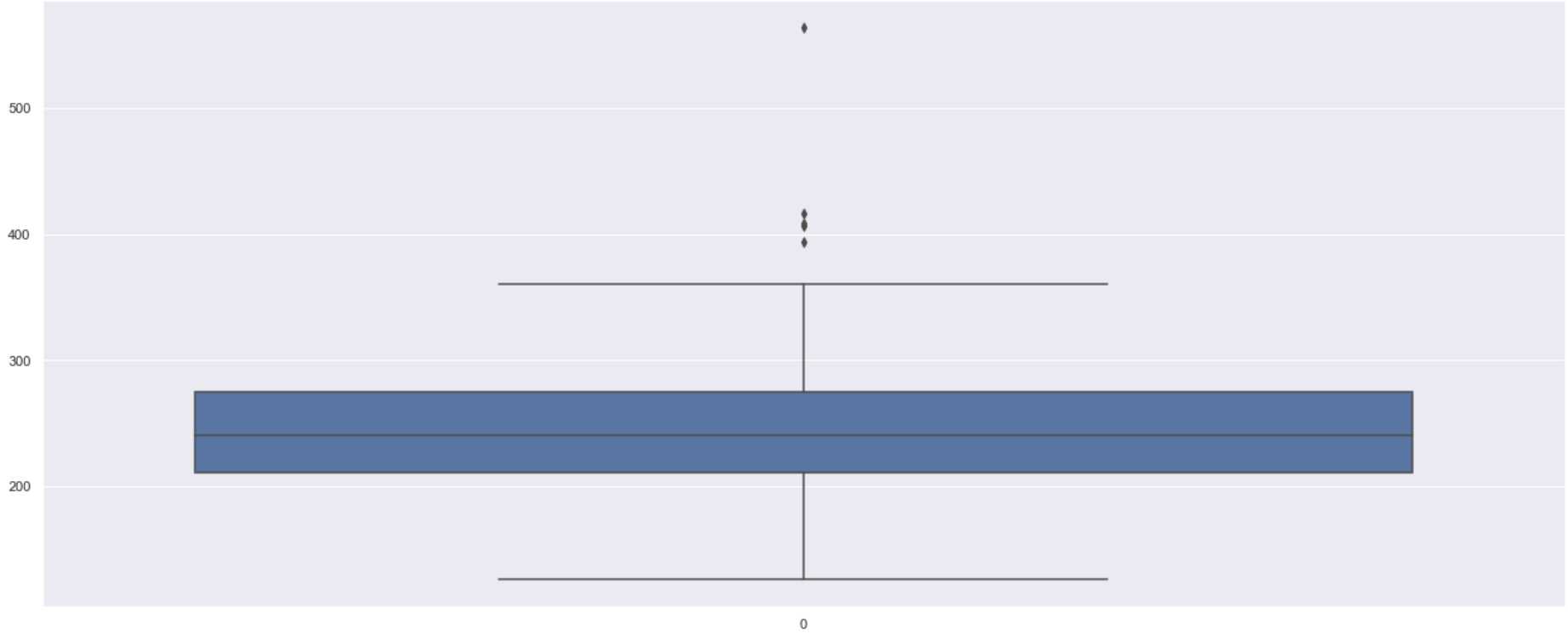
In [444]:
```
1 df['TRestBPS'].plot(kind = 'density')
```

Out[444]: <AxesSubplot:ylabel='Density'>



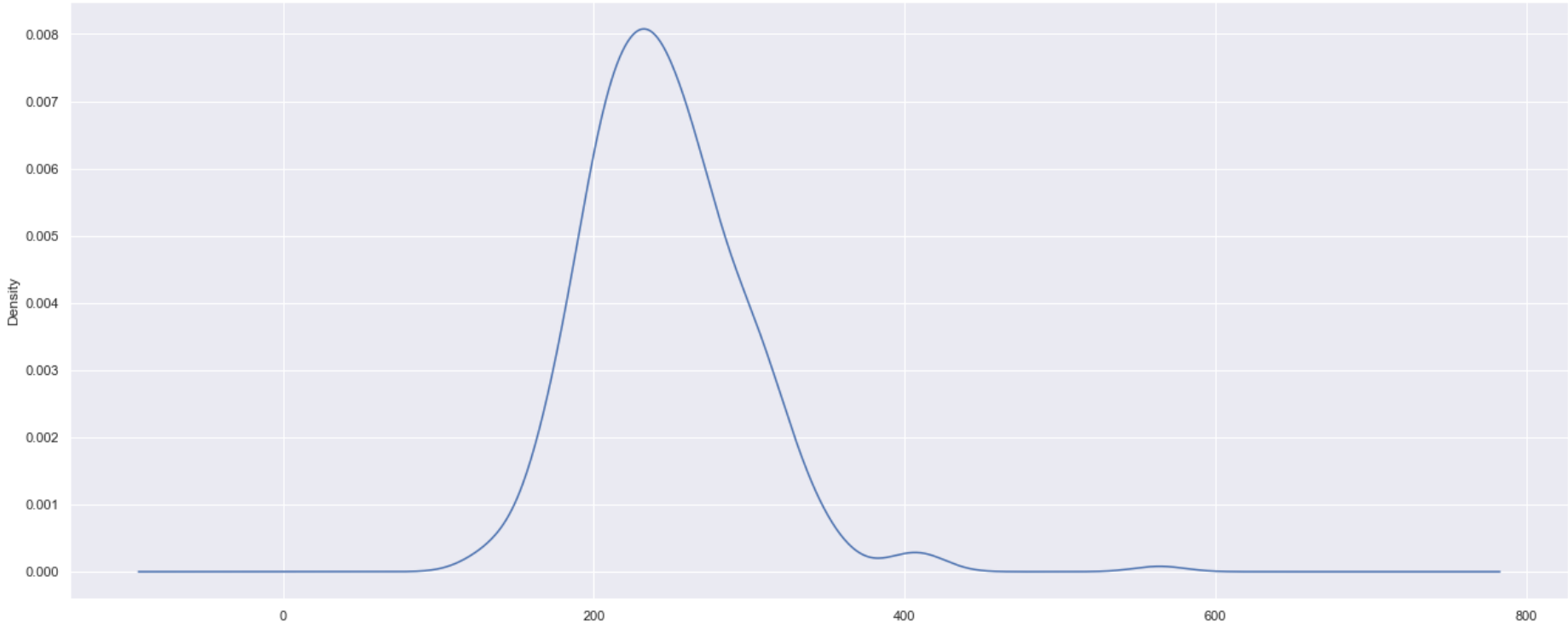
In [445]:
```
import seaborn as sb
sb.boxplot(data=df['CHOL'])


```

Out[445]: <AxesSubplot:>



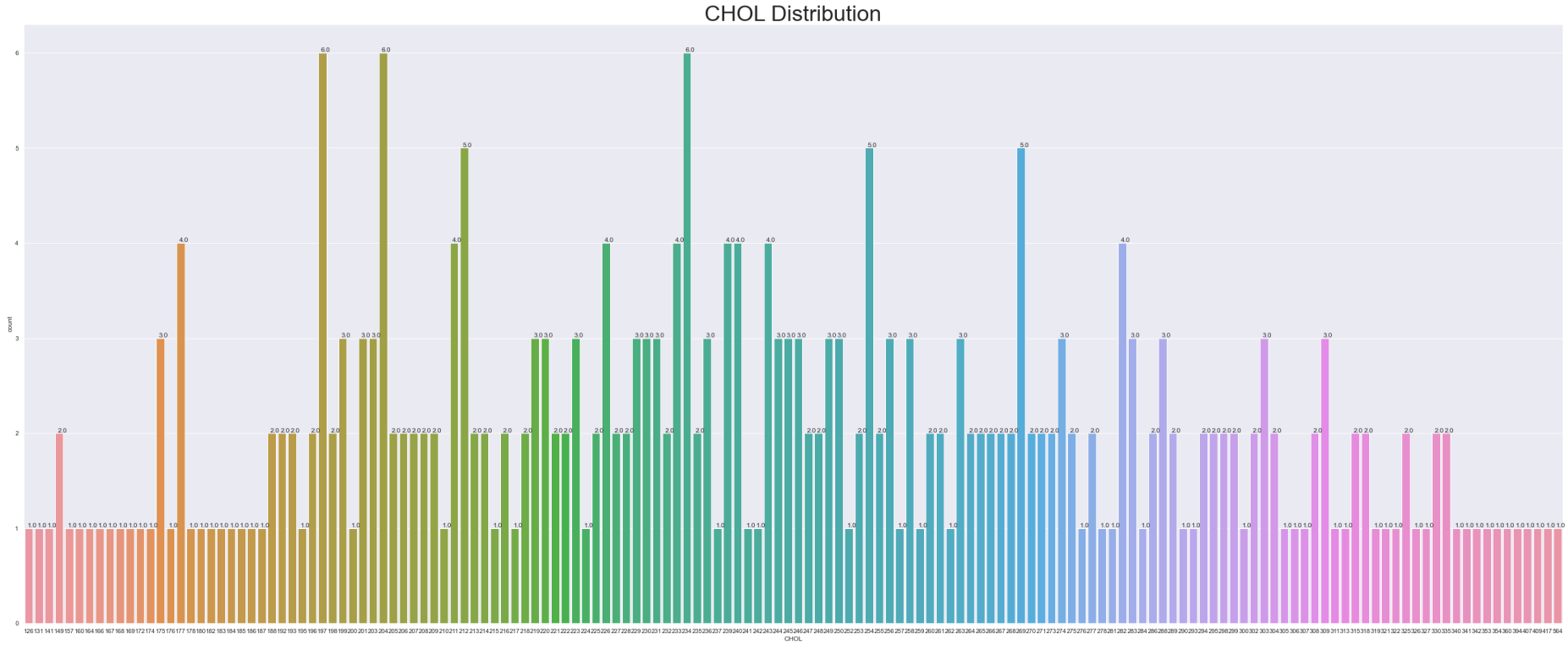
In [446]:
```
1 df['CHOL'].plot(kind = 'density')
```

Out[446]: <AxesSubplot:ylabel='Density'>



In [448]:

```
ax = sns.countplot(x="CHOL", data=df)
ax.set_title('CHOL Distribution', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
sns.set(rc={'figure.figsize':(50,20)})

```

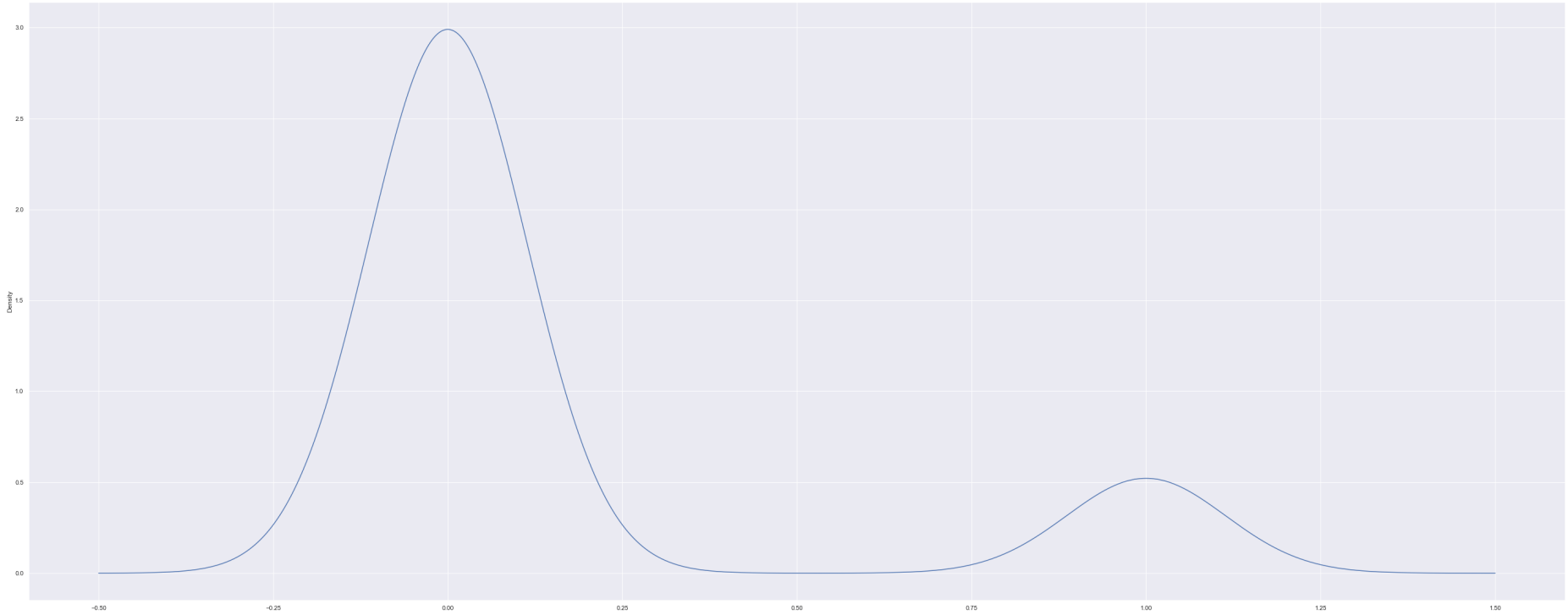


In [449]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['FBS'])
```

Out[449]: <AxesSubplot:>

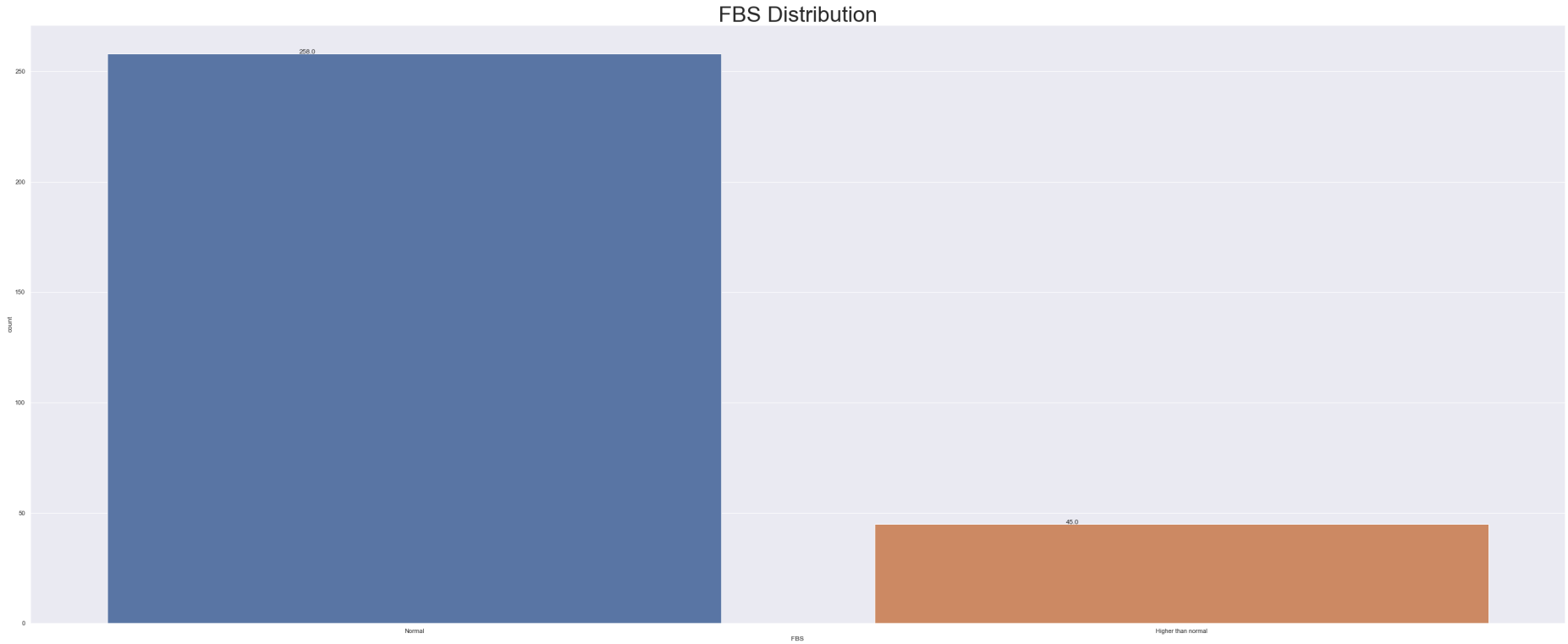
In [450]:
```
1 df['FBS'].plot(kind = 'density')
```

Out[450]: <AxesSubplot:ylabel='Density'>



In [455]:

```
ax = sns.countplot(x="FBS", data=df)
ax.set_title('FBS Distribution', fontsize=40)
ax.set_xticklabels(['Normal','Higher than normal'])
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```

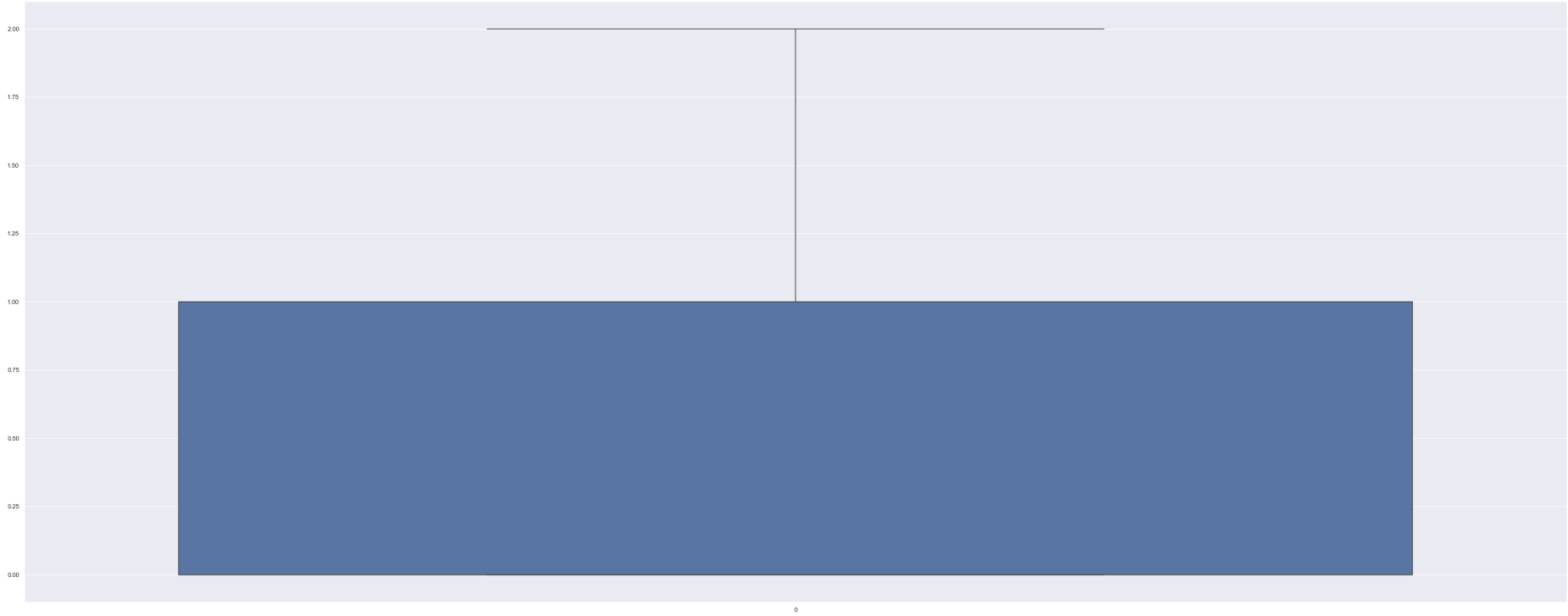


In [313]:
```
df.columns
```

Out[313]:
```
Index(['AGE', 'SEX', 'CP', 'TRestBPS', 'CHOL', 'FBS', 'restECG', 'Thalach',
       'EXANG', 'OldPeak', 'SLOPE', 'CA', 'THAL', 'TARGET'],
      dtype='object')
```

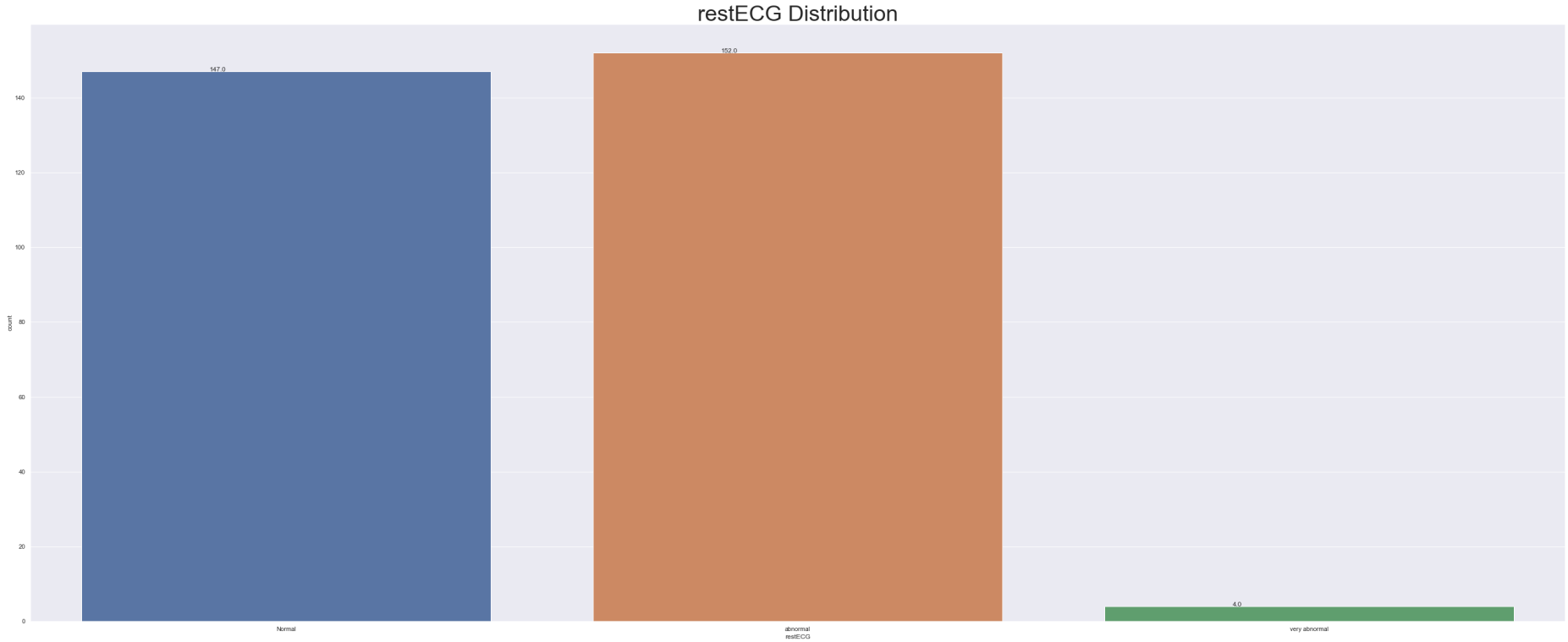
In [314]:
```
import seaborn as sb
sb.boxplot(data=df['restECG'])
```

Out[314]:
```
<AxesSubplot:>
```



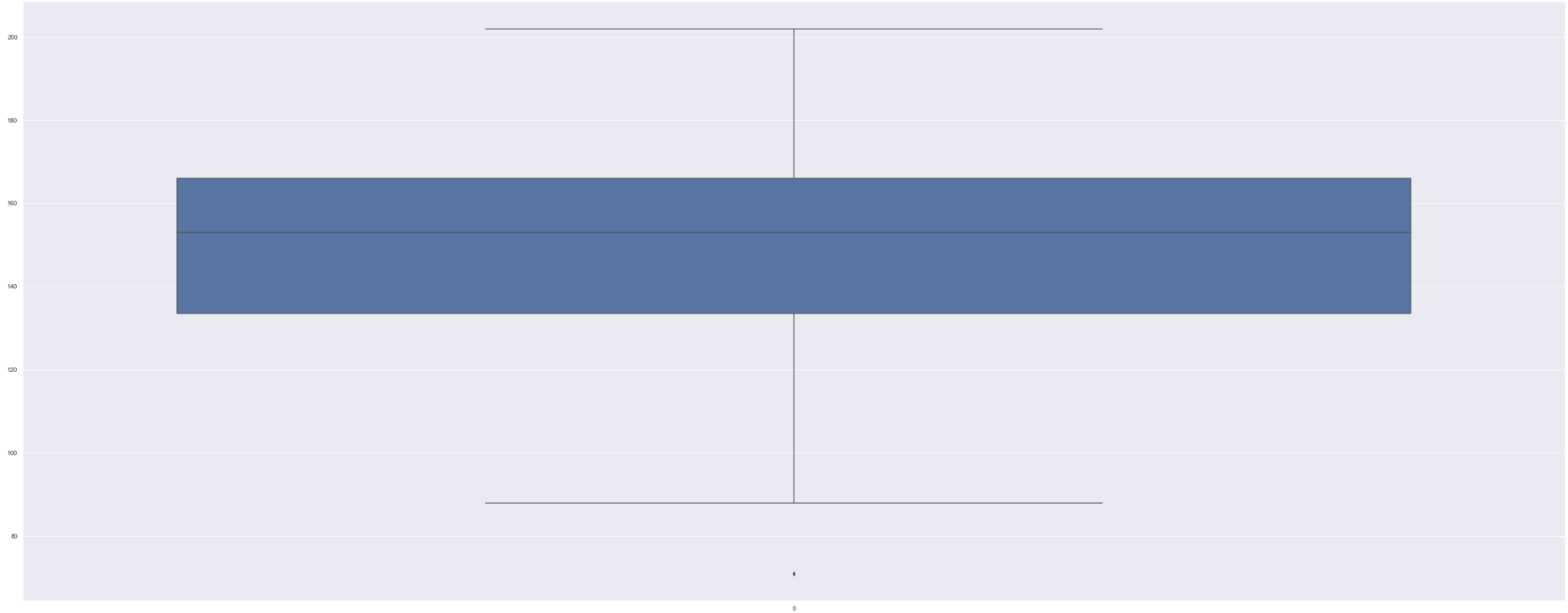
In [457]:

```
ax = sns.countplot(x="restECG", data=df)
ax.set_title('restECG Distribution', fontsize=40)
ax.set_xticklabels(['Normal','abnormal', 'very abnormal'])
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```



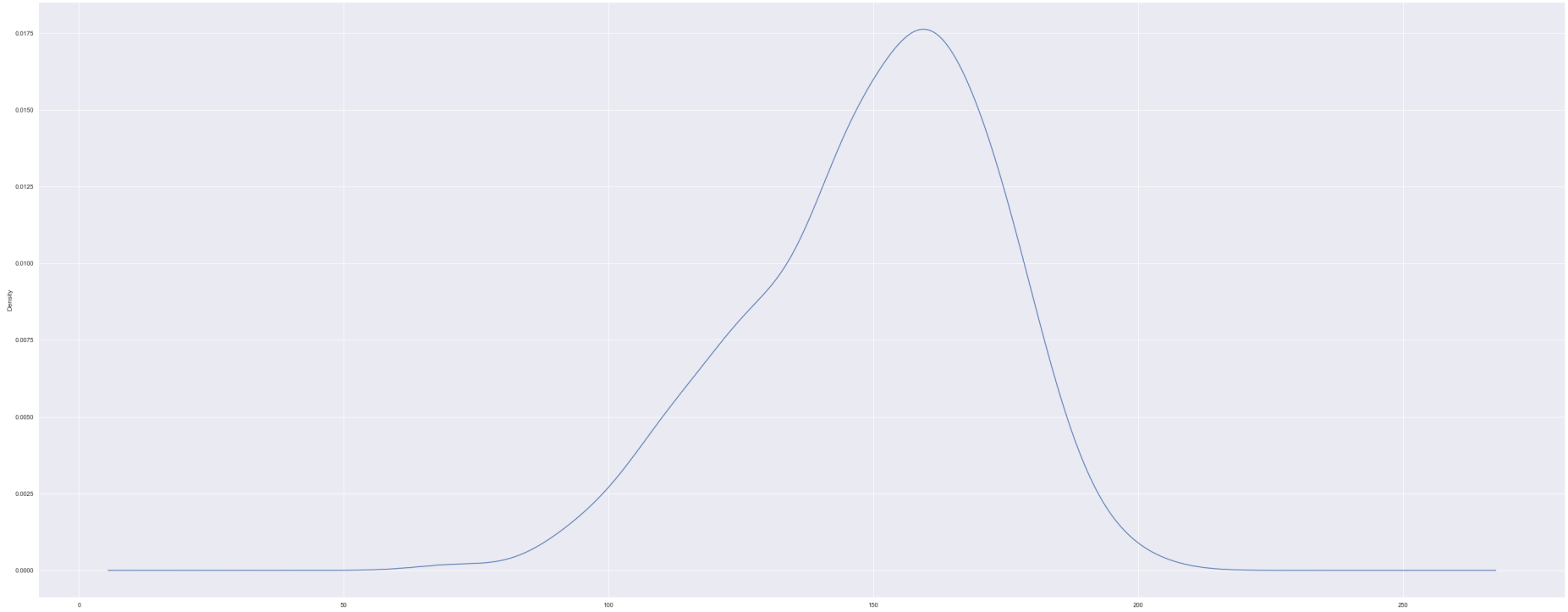
In [458]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['Thalach'])
```

Out[458]: <AxesSubplot:>



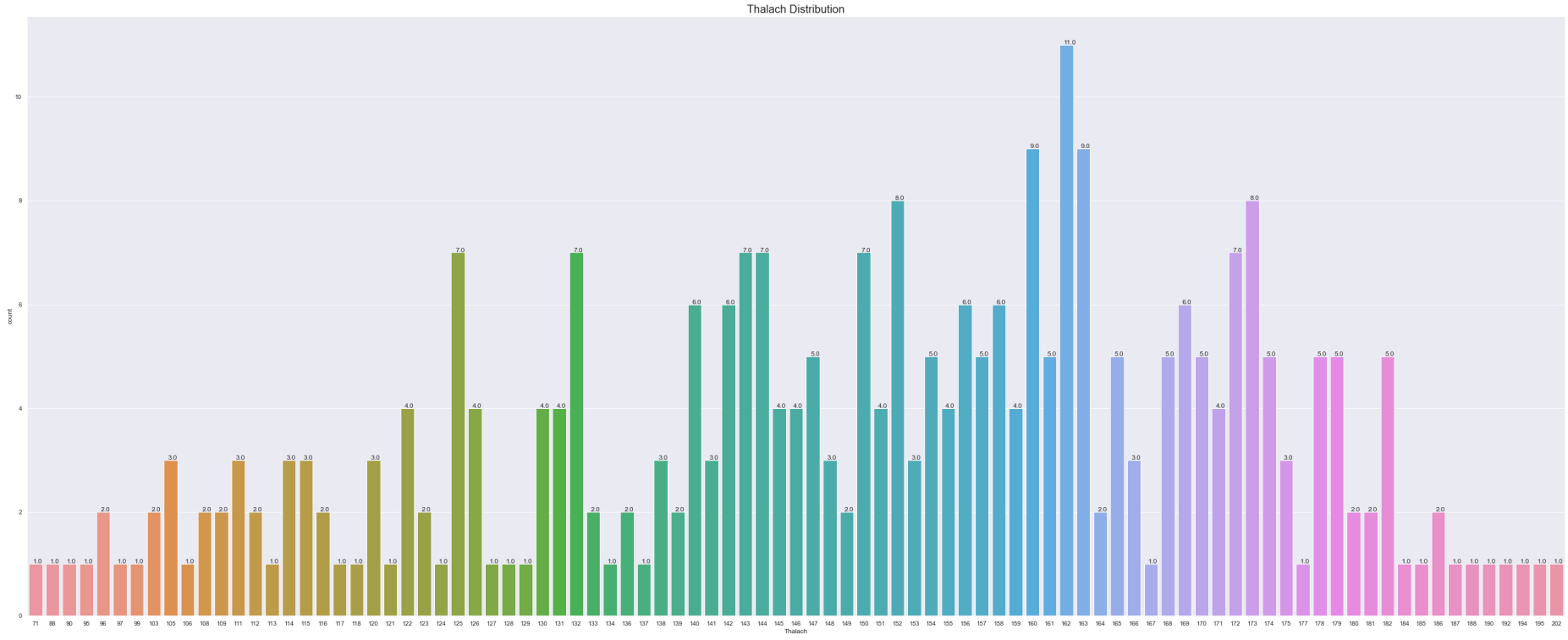
In [459]:
```
1 df['Thalach'].plot(kind = 'density')
```

Out[459]: <AxesSubplot:ylabel='Density'>



In [460]:

```
ax = sns.countplot(x="Thalach", data=df)
ax.set_title('Thalach Distribution', fontsize=20)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```

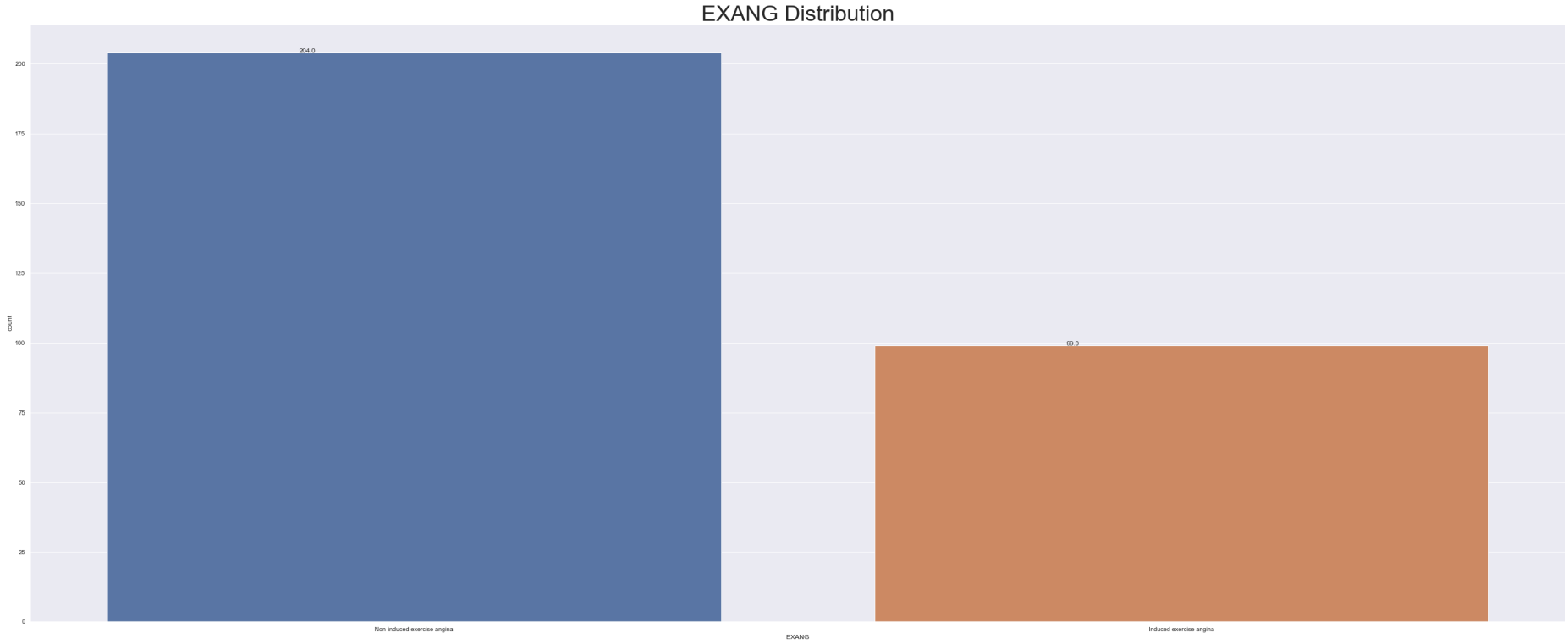


In [461]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['EXANG'])
```

Out[461]: <AxesSubplot:>

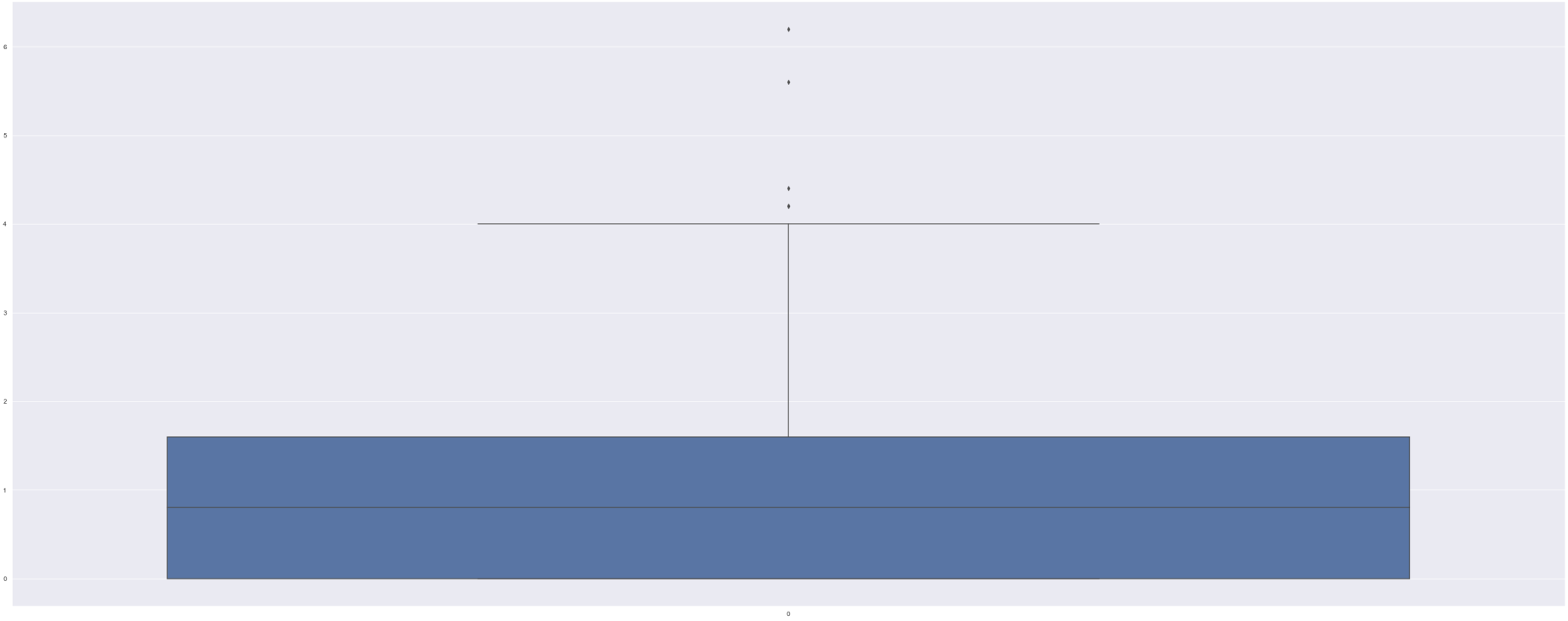
In [462]:

```
ax = sns.countplot(x="EXANG", data=df)
ax.set_title('EXANG Distribution', fontsize=40)
ax.set_xticklabels(['Non-induced exercise angina', 'Induced exercise angina'])
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```



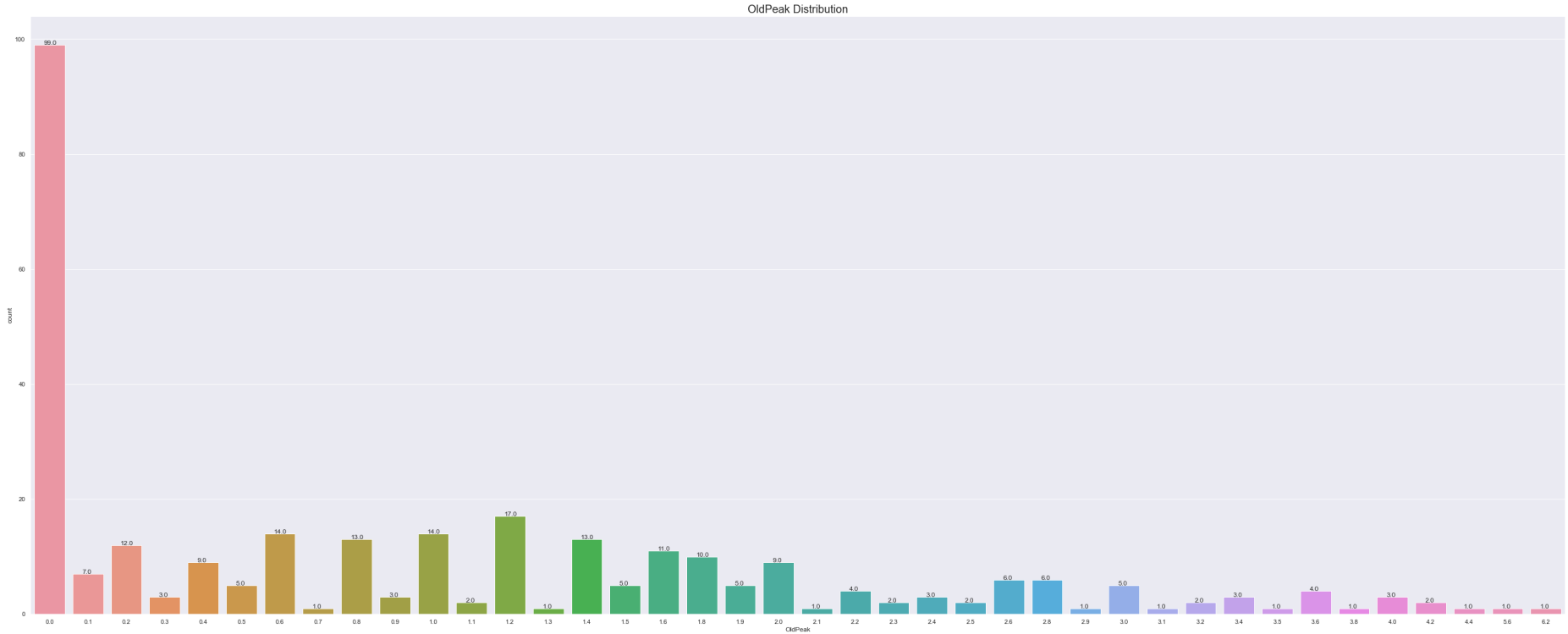
In [463]:
```
import seaborn as sb
sb.boxplot(data=df['OldPeak'])
```

Out[463]: <AxesSubplot:>



In [464]:

```
ax = sns.countplot(x="OldPeak", data=df)
ax.set_title('OldPeak Distribution', fontsize=20)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```



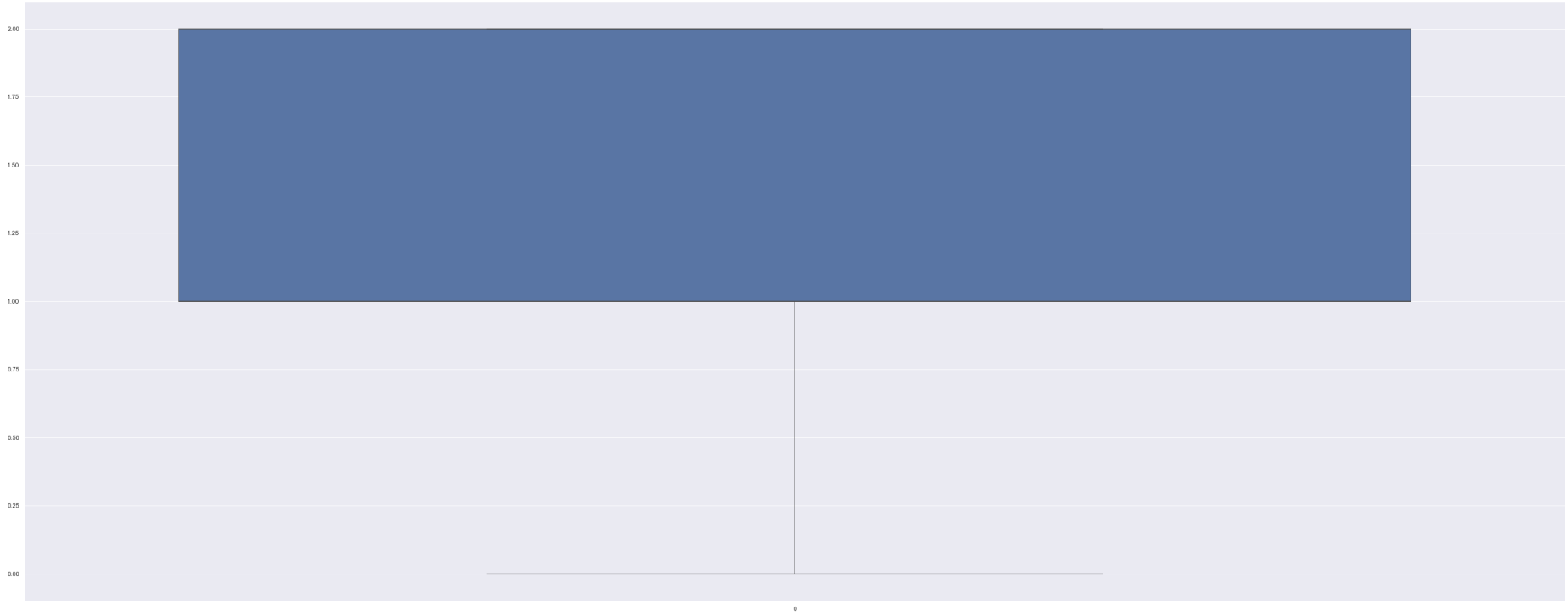
In [465]:
```
1 df.head()
```

Out[465]:

	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

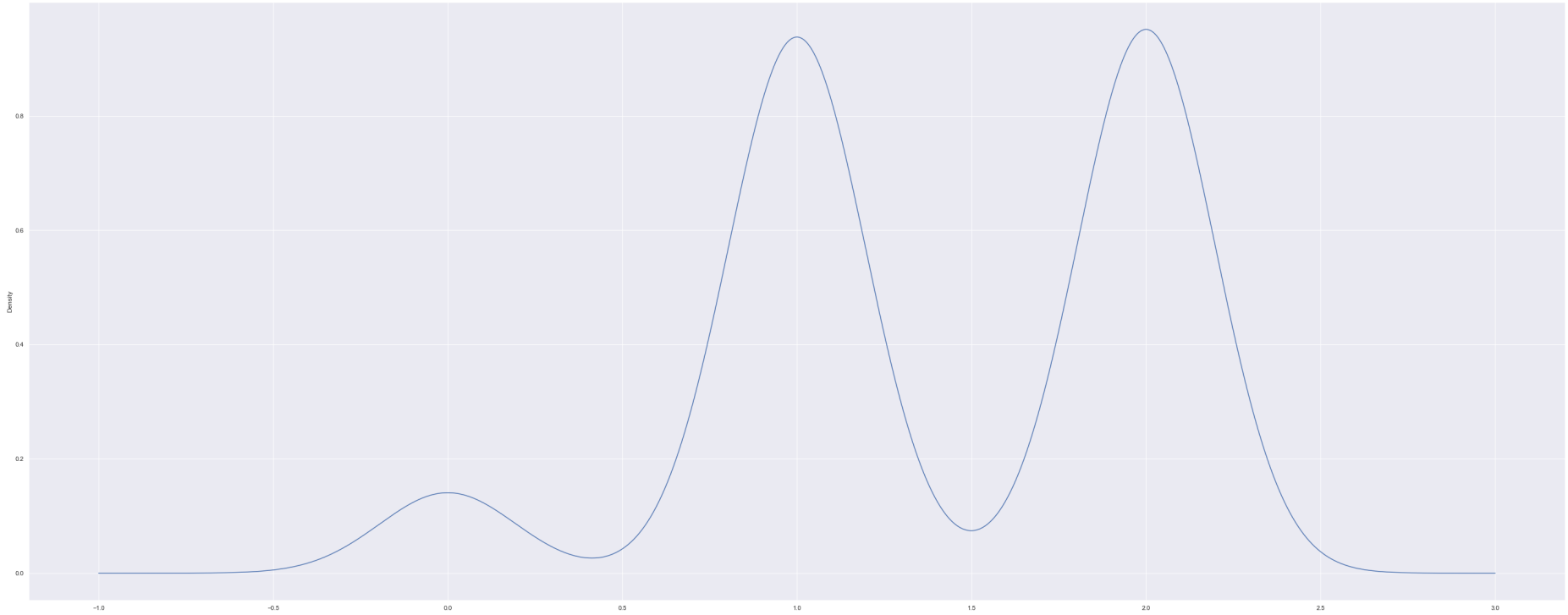
In [466]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['SLOPE'])
```

Out[466]: <AxesSubplot:>



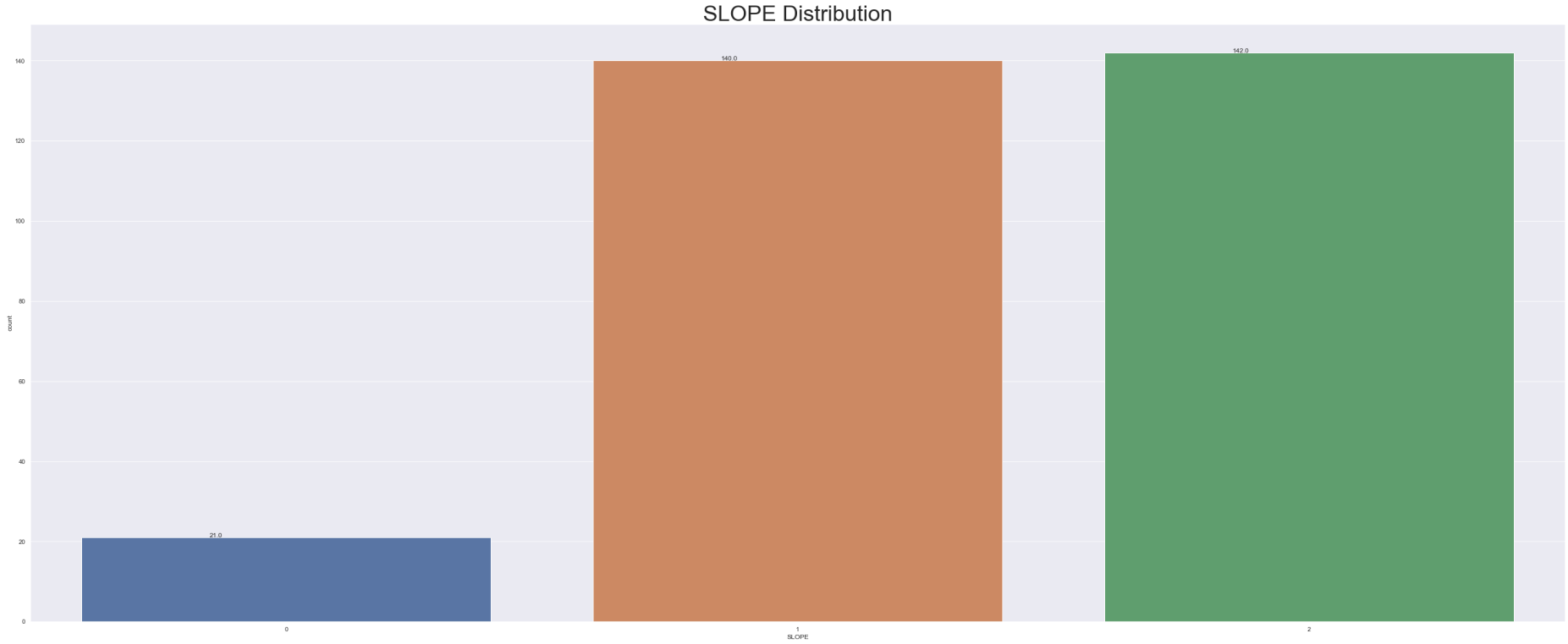
In [467]:
```
1 df['SLOPE'].plot(kind='density')
```

Out[467]: <AxesSubplot:ylabel='Density'>



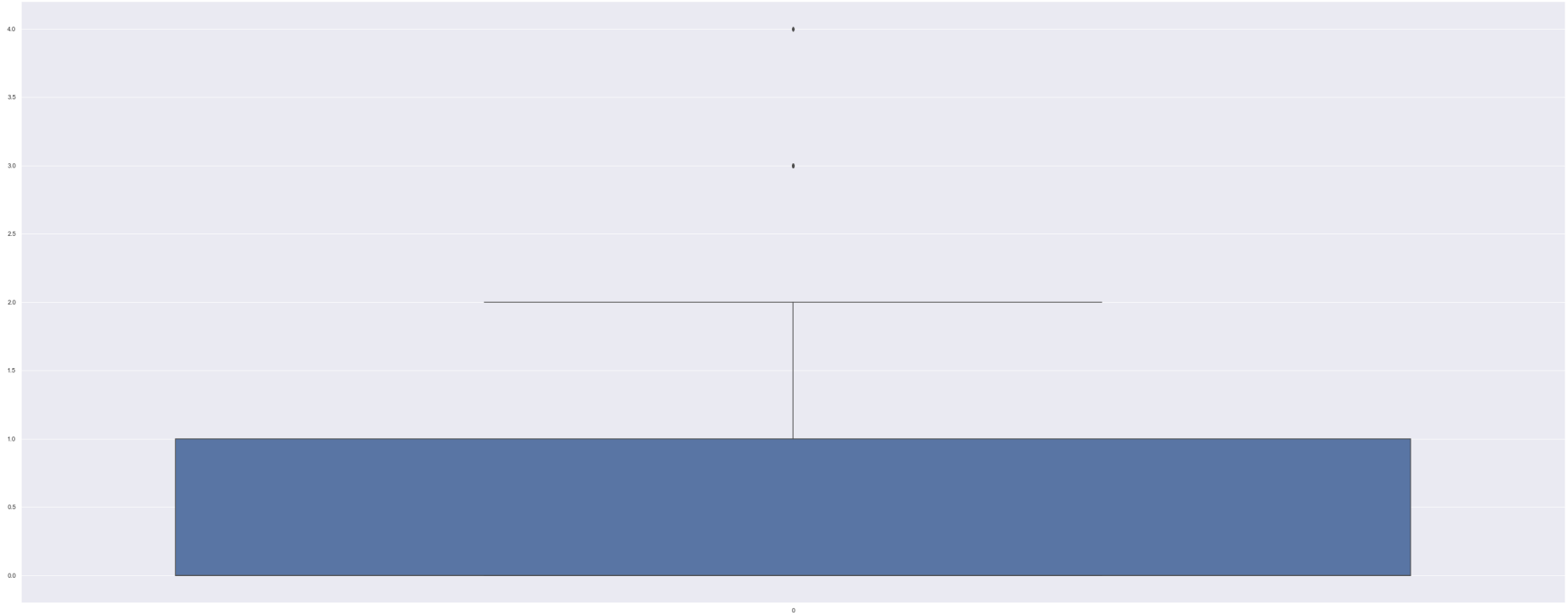
In [469]:

```
ax = sns.countplot(x="SLOPE", data=df)
ax.set_title('SLOPE Distribution', fontsize=40)
ax.set_xticklabels(['0','1','2'])
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```



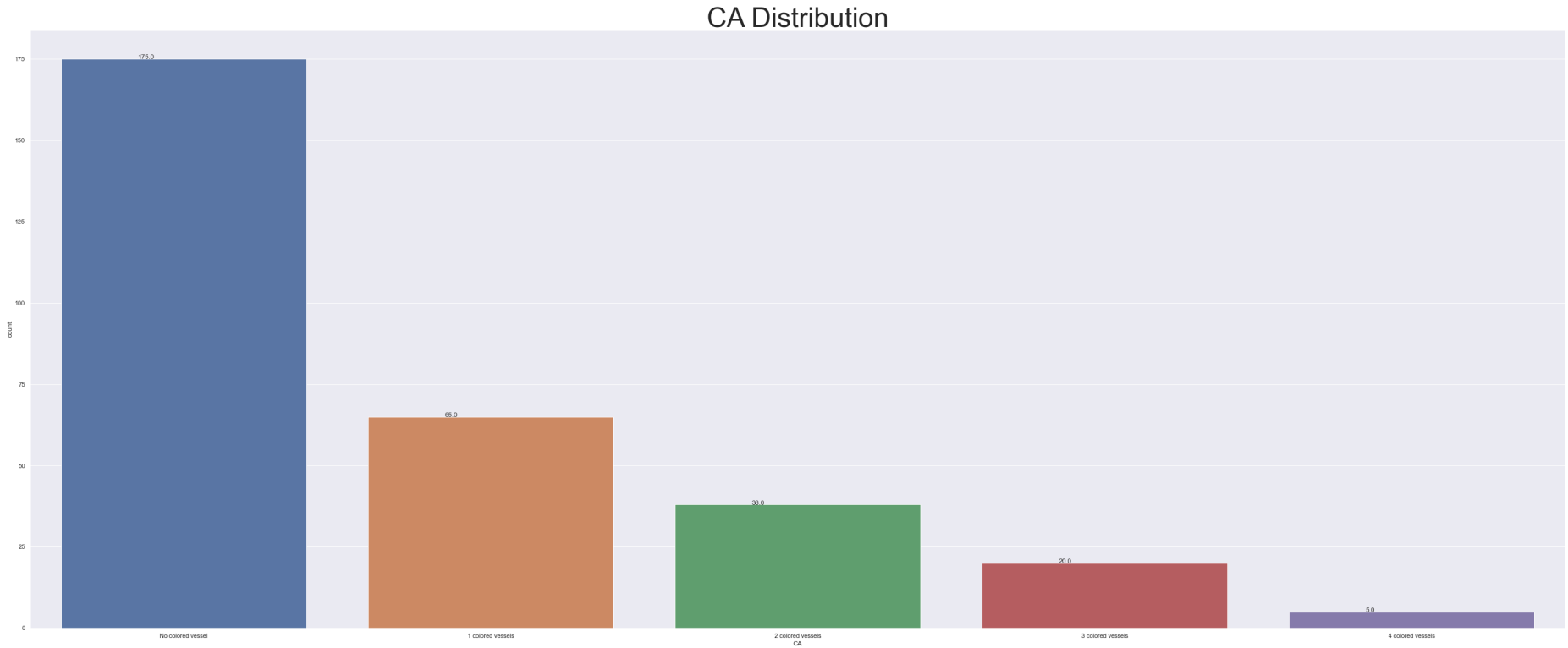
In [470]:
```
import seaborn as sb
sb.boxplot(data=df['CA'])
```

Out[470]: <AxesSubplot:>



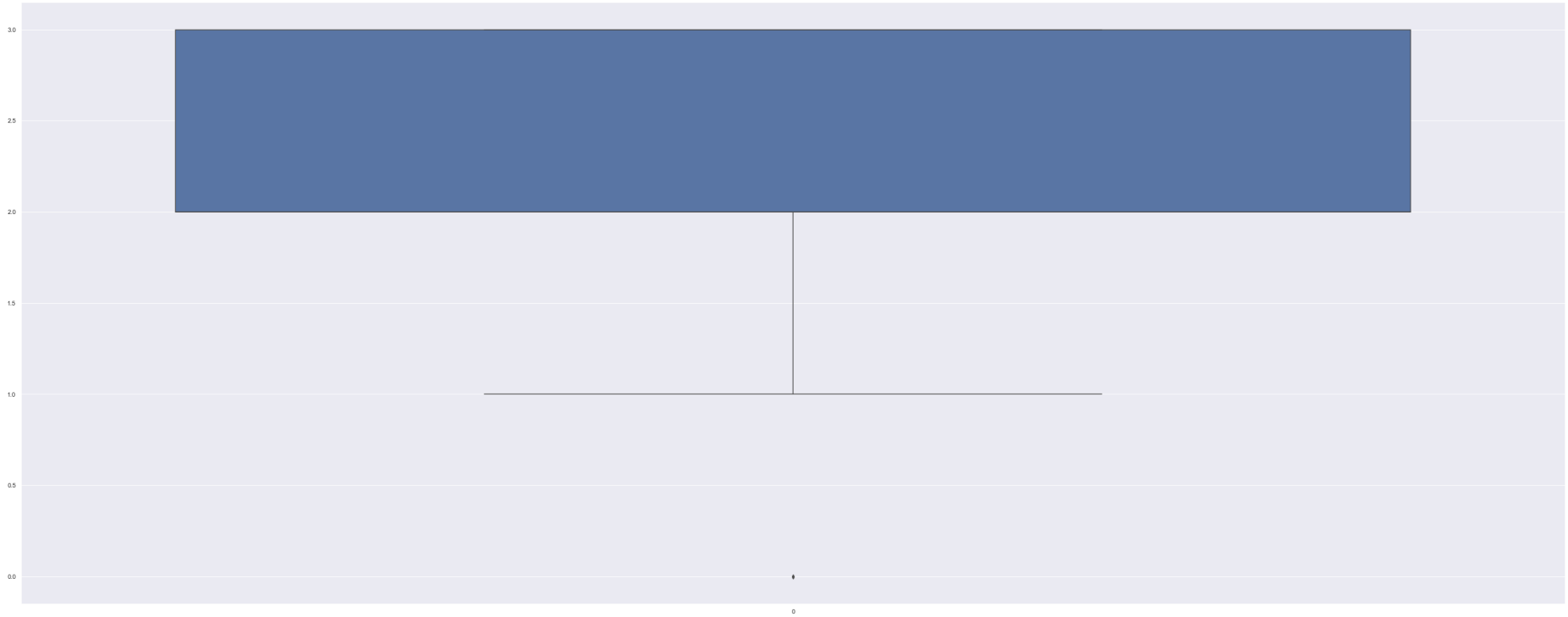
In [475]:

```
ax = sns.countplot(x="CA", data=df)
ax.set_title('CA Distribution', fontsize=50)
ax.set_xticklabels(['No colored vessel','1 colored vessels','2 colored vessels','3 colored vessels','4 colo
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



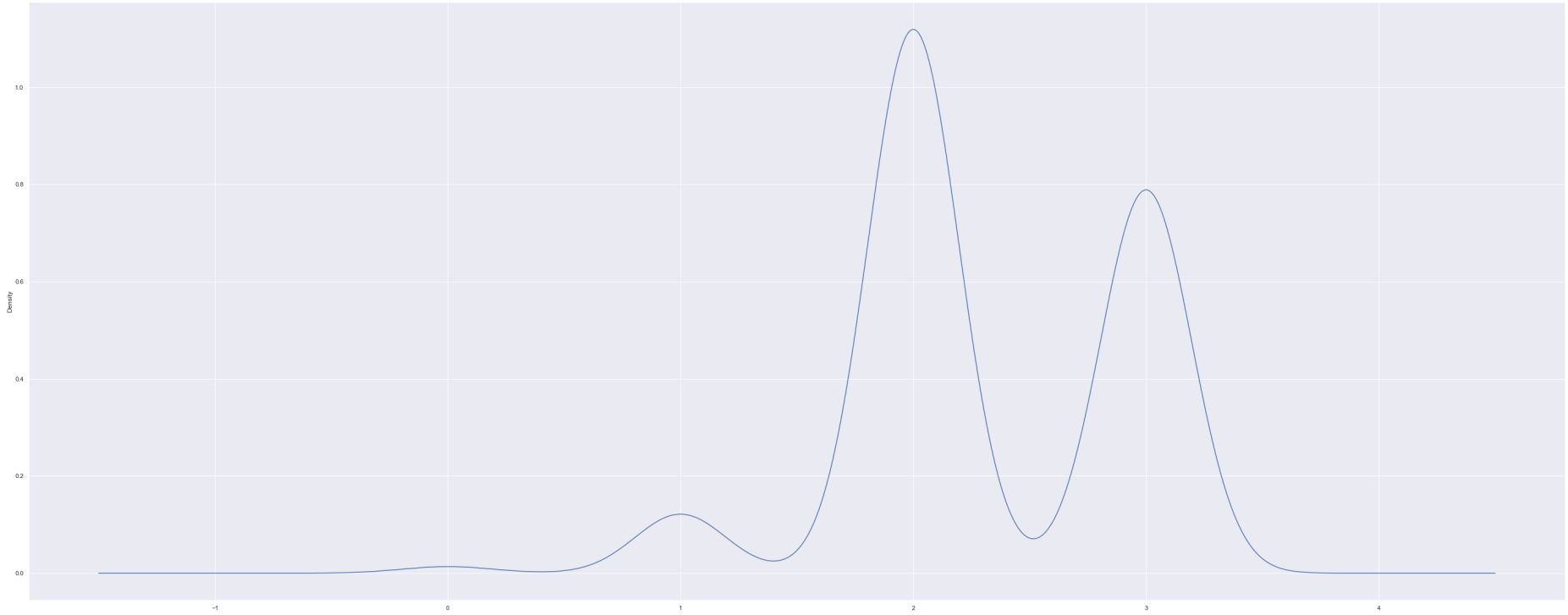
In [476]:
```
1 import seaborn as sb
2 sb.boxplot(data=df['THAL'])
```

Out[476]: <AxesSubplot:>



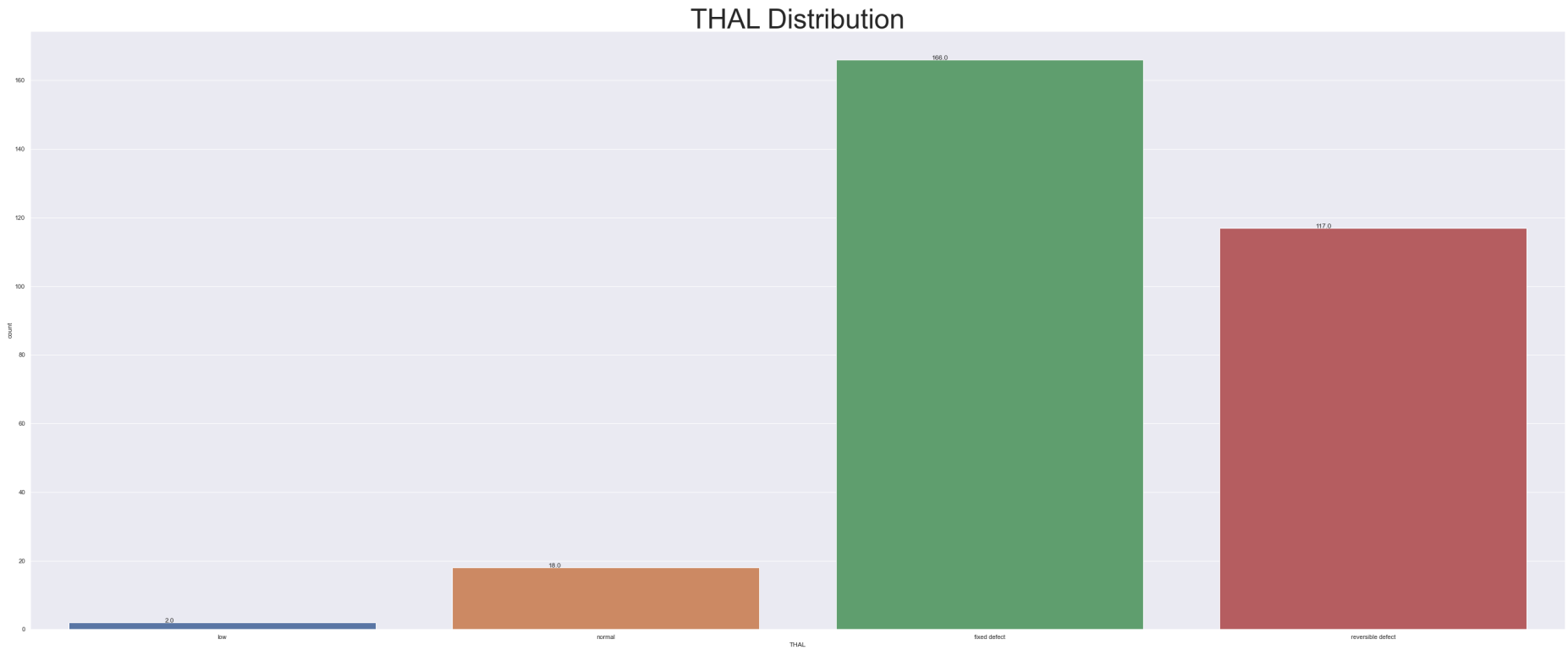
In [477]:
```
1 df['THAL'].plot(kind='density')
```

Out[477]: <AxesSubplot:ylabel='Density'>



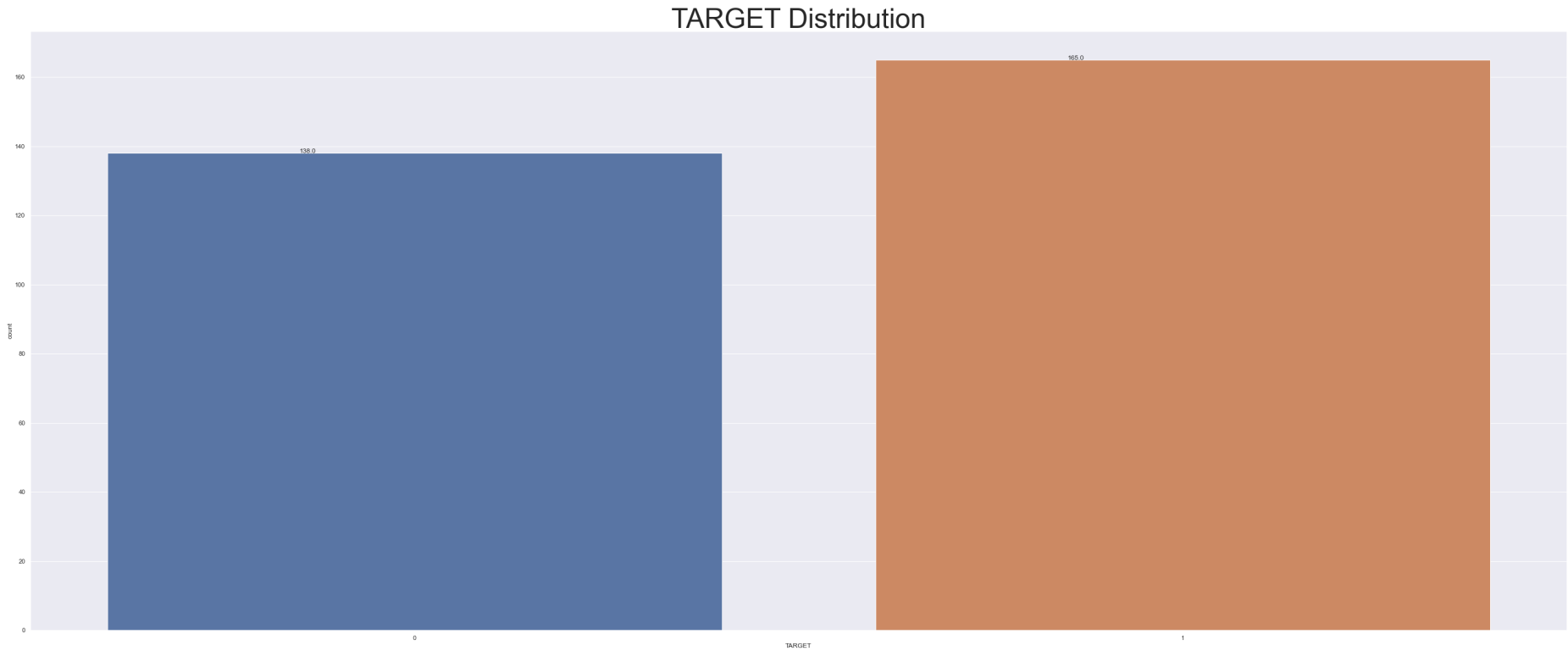
In [478]:

```
ax = sns.countplot(x="THAL", data=df)
ax.set_title('THAL Distribution', fontsize=50)
ax.set_xticklabels(['low', 'normal','fixed defect','reversible defect'])
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))

```



In [479]:

```
ax = sns.countplot(x="TARGET", data=df)
ax.set_title('TARGET Distribution', fontsize=50)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in ax.patches:
   ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))


```



In [480]:
```
1 df.head()
```

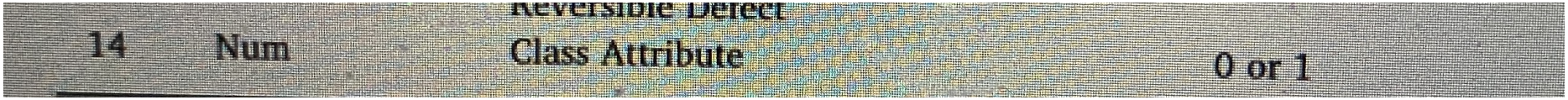
Out[480]:

	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

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

S.No	Attribute Name	Description	Range of Values
1	Age	Age of the person in years	29 to 79
2	Sex	Gender of the person [1: Male, 0: Female]	0, 1
3	Cp	Chest pain type [1-Typical Type 1 Angina 2- Atypical Type Angina 3-Non-angina pain 4-Asymptomatic)	1, 2, 3, 4
4	Trestbps	Resting Blood Pressure in mm Hg	94 to 200
5	Chol	Serum cholesterol in mg/dl	126 to 564
6	Fbs	Fasting Blood Sugar in mg/dl	0, 1
7	Restecg	Resting Electrocardiographic Results	0, 1, 2
8	Thalach	Maximum Heart Rate Achieved	71 to 202
9	Exang	Exercise Induced Angina	0, 1
10	OldPeak	ST depression induced by exercise relative to rest	1 to 3
11	Slope	Slope of the Peak Exercise ST segment	1, 2, 3
12	Ca	Number of major vessels colored by fluoroscopy	0 to 3
13	Thal	3 – Normal, 6 – Fixed Defect, 7 – Reversible Defect	3, 6, 7



1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

    The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

In [481]:
```
minAge = min(df['AGE'])
print(minAge)
maxAge = max(df['AGE'])
print(maxAge)
'''29-40 = 0
   41-52 = 1
   53-64 = 2
   64-77 = 3'''
```

```
29
77
```

Out[481]:
```
'29-40 = 0\n   41-52 = 1\n   53-64 = 2\n   64-77 = 3'
```

In [482]:
```
df.AGE.unique()
```

Out[482]:
```
array([63, 37, 41, 56, 57, 44, 52, 54, 48, 49, 64, 58, 50, 66, 43, 69, 59,
       42, 61, 40, 71, 51, 65, 53, 46, 45, 39, 47, 62, 34, 35, 29, 55, 60,
       67, 68, 74, 76, 70, 38, 77])
```

In [483]:
```
fill = (df.AGE == 29) | (df.AGE == 34)| (df.AGE == 35) | (df.AGE ==38)| (df.AGE ==39) | (df.AGE ==40) | (df
df.loc[fill, 'AGE'] = 0
```

In [484]:
```
fill = (df.AGE == 41) | (df.AGE == 42)| (df.AGE == 44)| (df.AGE == 43) | (df.AGE == 50) |(df.AGE ==45)| (df
df.loc[fill, 'AGE'] = 1
```

In [485]:
```
fill = (df.AGE == 53) | (df.AGE == 54)| (df.AGE == 55) | (df.AGE ==57)| (df.AGE ==59) |(df.AGE ==56) |(df.A(
df.loc[fill, 'AGE'] = 2
```

In [486]:
```
fill = (df.AGE == 65) | (df.AGE == 66)| (df.AGE == 67) | (df.AGE ==68)| (df.AGE ==69) | (df.AGE ==70)| (df.A
df.loc[fill, 'AGE'] = 3
```

In [487]:
```
minBP = min(df['TRestBPS'])
print(minBP)

maxBP = max(df['TRestBPS'])
print(maxBP)

'''90-119  = Normal
   120-139 = Borderline HBP
   140-200 = High BP'''
```

```
94
200
```

Out[487]:
```
'90-119  = Normal\n   120-139 = Borderline HBP\n   140-200 = High BP'
```

In [488]:
```
fill = (df.TRestBPS == 94) | (df.TRestBPS == 110) | (df.TRestBPS == 104) | (df.TRestBPS == 105)| (df.TRestBI
df.loc[fill, 'TRestBPS'] = 0
```

In [489]:
```
fill = (df.TRestBPS == 120) | (df.TRestBPS == 130) | (df.TRestBPS == 135) | (df.TRestBPS == 125) | (df.TRest
df.loc[fill, 'TRestBPS'] = 1
```

In [490]:
```
fill = (df.TRestBPS == 140) | (df.TRestBPS == 145) | (df.TRestBPS == 172) | (df.TRestBPS == 150) | (df.TRest
df.loc[fill, 'TRestBPS'] = 2
```

In [491]:
```
df.TRestBPS.unique()
```

Out[491]:
```
array([2, 1, 0])
```

In [ ]:
```

```

In [492]:
```
minChol = min(df['CHOL'])
print(minChol)
maxChol = max(df['CHOL'])
print(maxChol)

'''126-200 = Normal
   201-239 = Borderline high
   240-564 = high'''
```

```
126
564
```

Out[492]:
```
'126-200 = Normal\n   201-239 = Borderline high\n   240-564 = high'
```

In [493]:
```
fill = (df.CHOL == 126) | (df.CHOL == 192)| (df.CHOL == 199) | (df.CHOL ==168)| (df.CHOL ==175) | (df.CHOL =
df.loc[fill, 'CHOL'] = 0
```

In [494]:
```
== 203) |(df.CHOL == 206) |(df.CHOL == 209) |(df.CHOL == 217) |(df.CHOL == 218) |(df.CHOL == 224) |(df.CHOL == 2
OL'] = 1
```

In [495]:
```
fill = (df.CHOL == 241) | (df.CHOL == 242) | (df.CHOL == 244) | (df.CHOL == 294) | (df.CHOL == 246) | (df.CH
df.loc[fill, 'CHOL'] = 2
```

In [ ]:
```

```

In [496]:
```
sorted(df.CHOL.unique())
```

Out[496]:
```
[0, 1, 2]
```

In [497]:
```
minThalach = min(df['Thalach'])
print(minThalach)

maxThalach = max(df['Thalach'])
print(maxThalach)

'''71-100 = low Thalach
101-130 = normal Thalach
131-160 = borderline high Thalach
161-202 = high Thalach'''

```

```
71
202
```

Out[497]:
```
'71-100 = low Thalach\n101-130 = normal Thalach\n131-160 = borderline high Thalach\n161-202 = high Thalach'
```

In [498]:
```
fill = (df.Thalach == 71)|  (df.Thalach == 95)| (df.Thalach == 88)| (df.Thalach == 97)| (df.Thalach == 99)|
df.loc[fill, 'Thalach'] = 0
```

In [499]:
```
fill = (df.Thalach == 106)|  (df.Thalach == 103)| (df.Thalach == 127)| (df.Thalach == 105)| (df.Thalach ==
df.loc[fill, 'Thalach'] = 1
```

In [500]:
```
fill = (df.Thalach == 131)|  (df.Thalach == 132)| (df.Thalach == 133)| (df.Thalach == 134)| (df.Thalach ==
df.loc[fill, 'Thalach'] = 2
```

In [501]:
```
fill = (df.Thalach == 161)| (df.Thalach == 162)| (df.Thalach == 163)| (df.Thalach == 164)| (df.Thalach == 1
df.loc[fill,'Thalach'] = 3
```

In [502]:
```
sorted(df.Thalach.unique())
```

Out[502]:
```
[0, 1, 2, 3]
```

In [503]:
```
df.columns
```

Out[503]:
```
Index(['AGE', 'SEX', 'CP', 'TRestBPS', 'CHOL', 'FBS', 'restECG', 'Thalach',
       'EXANG', 'OldPeak', 'SLOPE', 'CA', 'THAL', 'TARGET'],
      dtype='object')
```

In [504]:

```
#BIVARATE ANALYSIS


'''I will be plotting distributions between these columns and the target column:
'AGE', 'SEX', 'CP', 'TRestBPS', 'CHOL', 'FBS', 'restECG', 'Thalach',
'EXANG', 'OldPeak', 'SLOPE', 'CA', 'THAL' '''

#For AGE,
'''we have a majority of the younger adults (A total of 84  out of 120 adults,
between 29 and 52; around 70%)
have a diseased heart as we don't expect whereas,
the reverse is the case for the older adults (A total of 81 out of 183;
between ages 53 and 77 around 44.2% ), have a diseased heart
Based on this outcome, heart disease is more common in younger adults.'''

#SEX
'''This study was carried out on fewer females than men but yet
there is a higher percentage of heart disease in females than in males which is very much significant.
This amounts to like 7 out of 9 (75% of the women) and 9 out of 20 (45% of the men) respectively
I believe there is a relationship between these two, we could infer that
females are likely more prone to a heart disease than males for whatever reason.
Studies have shown that women have smaller arteries than men, this could be a reason'''

#CP
'''The proportion of those with a diseased heart who also have one form of chest pain or the other
is higher than those with a healthy heart but it appears that
even those who do not have pain in their chest(are asymptomatic) and those who had pain not angina-induced
were considered to have a heart disease. Therefore, There is a relationship between these two columns'''

'''For TRestBPS,
we see that elevated blood pressure has no influence on heart disease development.
Even individuals who have normal blood pressure tend to have a higher proportion of heart disease developmer
Safe to say that high resting blood pressure has NO influence on heart disease development.'''


'''For CHOL,
There is almost an equal proportion of those who have a diseased heart
who also have normal or borderline high levels of blood cholesterol.
Even a 1:1 ratio of diseased and healthy heart in those who have higher blood cholesterol levels.
No increase in percentage of those with a diseased heart from normal to high cholesterol levels
and vice versa means no relationship between these two features'''
```

```
'''FBS
The number of those with high blood sugar way less than those with normal blood sugar.
For the relationship between FBS and target, proportion of those with a diseased heart
who also have a normal blood sugar level is even higher that those with high blood sugar levels
such that we cannot tell if individuals with higher than normal blood sugar levels
or with normal blood sugar levels will more likely have a diseased heart.
For me, I do not think there is a relatinship between these two
and having diabetes is not a yardstick for having a heart disease.'''

#For restECG,
'''Proportion of those with a diseased heart increases (from like 46.2% to 63%)
as we go from normal to abnormal restECG conditions.
If only we had more people with very abnormal ECG conditions, we may likely see an increase too
I think there is a relationship between these two columns
and restECG will be an important feature for the model design'''

#For Thalach,
'''Proportion of those with a diseased heart begins to increase with increase in their maximum heart rates.
I think there is a relationship between these two columns
and Thalach will be an important feature for the model design'''

#For EXANG,
'''There is a higher proportion (of those who have a diseased heart who also have non-induced exercise angi
than those with induced angina. Both type of anginas are not good and can be a sign of a failing heart.
I think there is a relationship between these two columns'''

#For SLOPE,
'''Proportion of those with a diseased heart begins to increase
with a much significant increase in their slope measurements(from flat-sloping to down-sloping).
I think there is a relationship between these two columns
and SLOPE will be an important feature for the model design'''


'''For CA feature, I expect a majority of those with 2, 3 or 4 coloured vessels
to have a diseased heart but the reverse is the case.
Even the majority of those with no coloured vessel at all have a diseased heart
I think there is no relationship between these two columns and will not be an import feature for the model'

#THAL,
'''There is a higher proportion (of those who have a diseased heart who also have fixed-defect thal)
than those with reversible defect thal. Both type of thal are not good and can be a sign of a failing heart
```

```
I think there is also a relationship between these two columns'''


'''The features in blue highlights all have a relationship with TARGET(outcome variable)'''
```

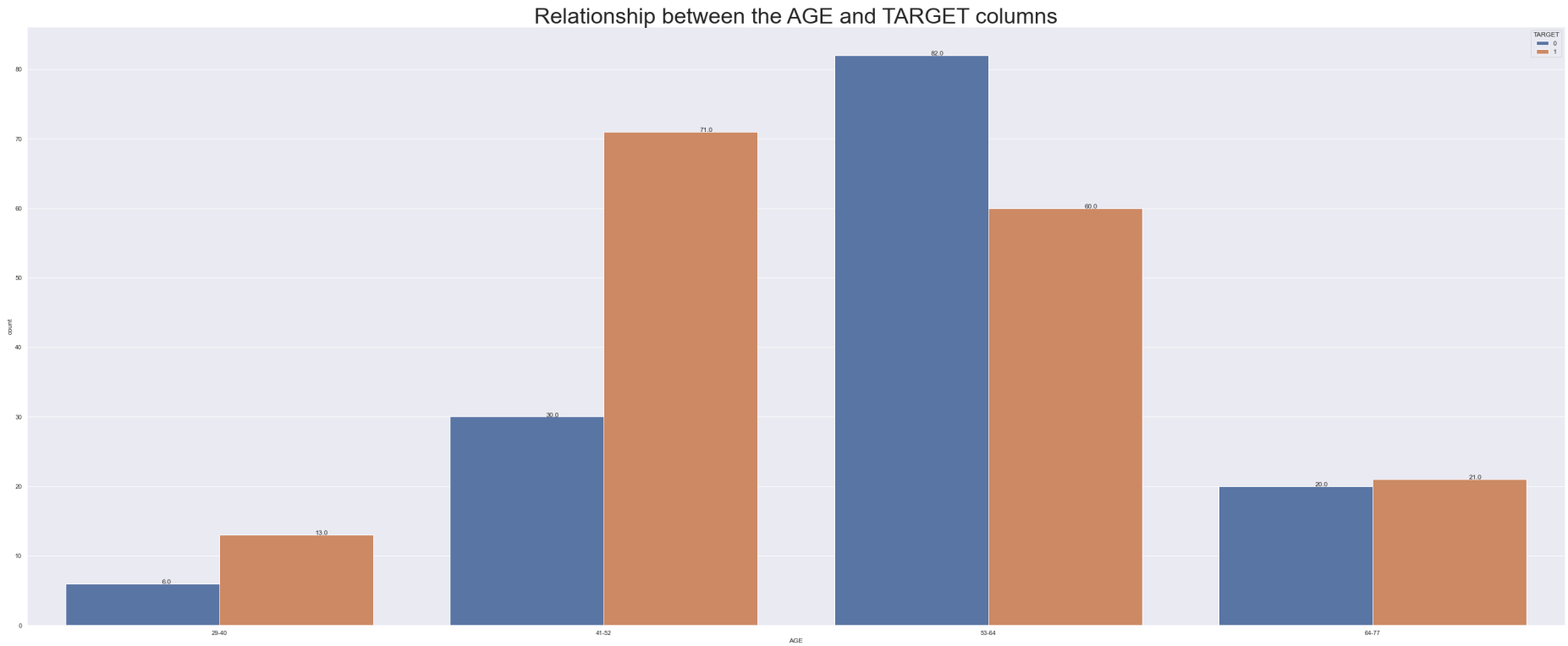
Out[504]: 'The features in blue highlights all have a relationship with TARGET(outcome variable)'

In [505]:
```
df['TARGET'].value_counts()
```

Out[505]:
```
1    165
0    138
Name: TARGET, dtype: int64
```

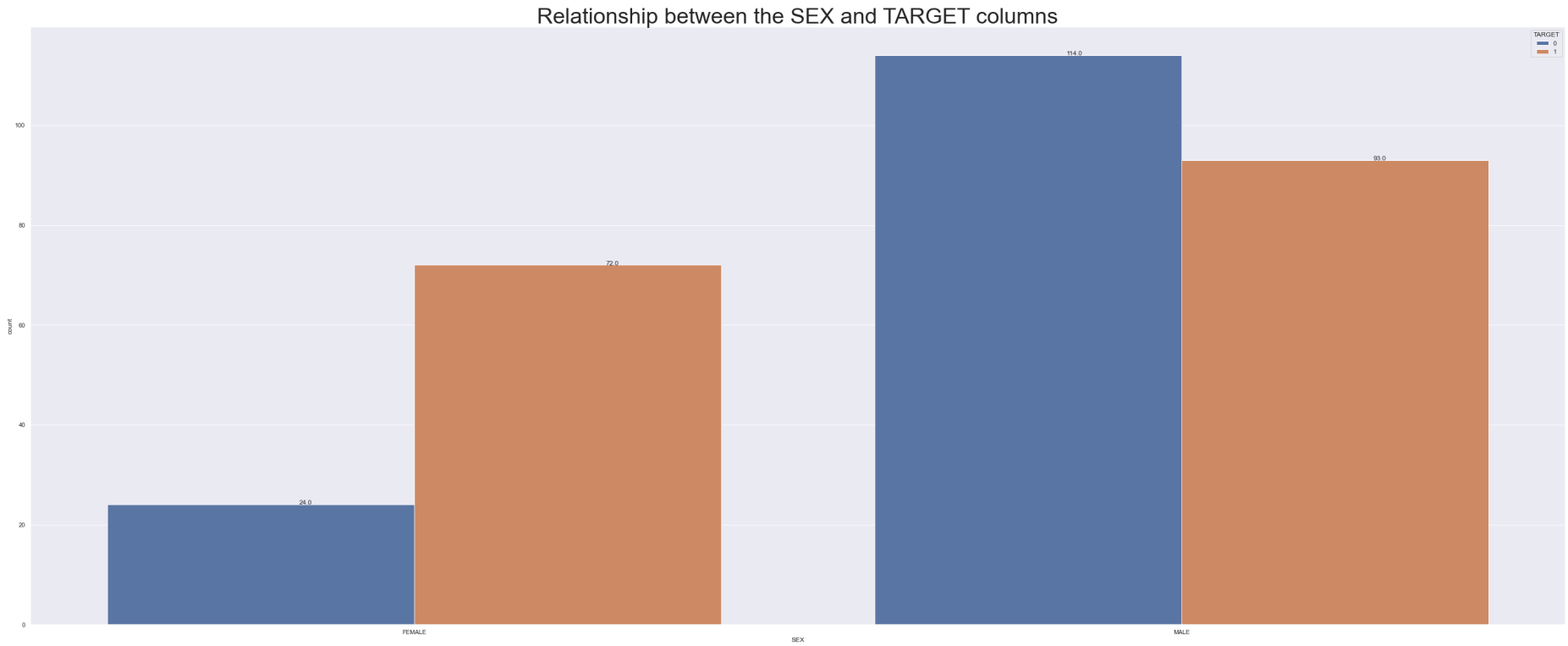
In [506]:

```
edu = sns.countplot(x='AGE', hue='TARGET', data=df)
edu.set_xticklabels(['29-40','41-52','53-64','64-77'])
edu.set_title('Relationship between the AGE and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



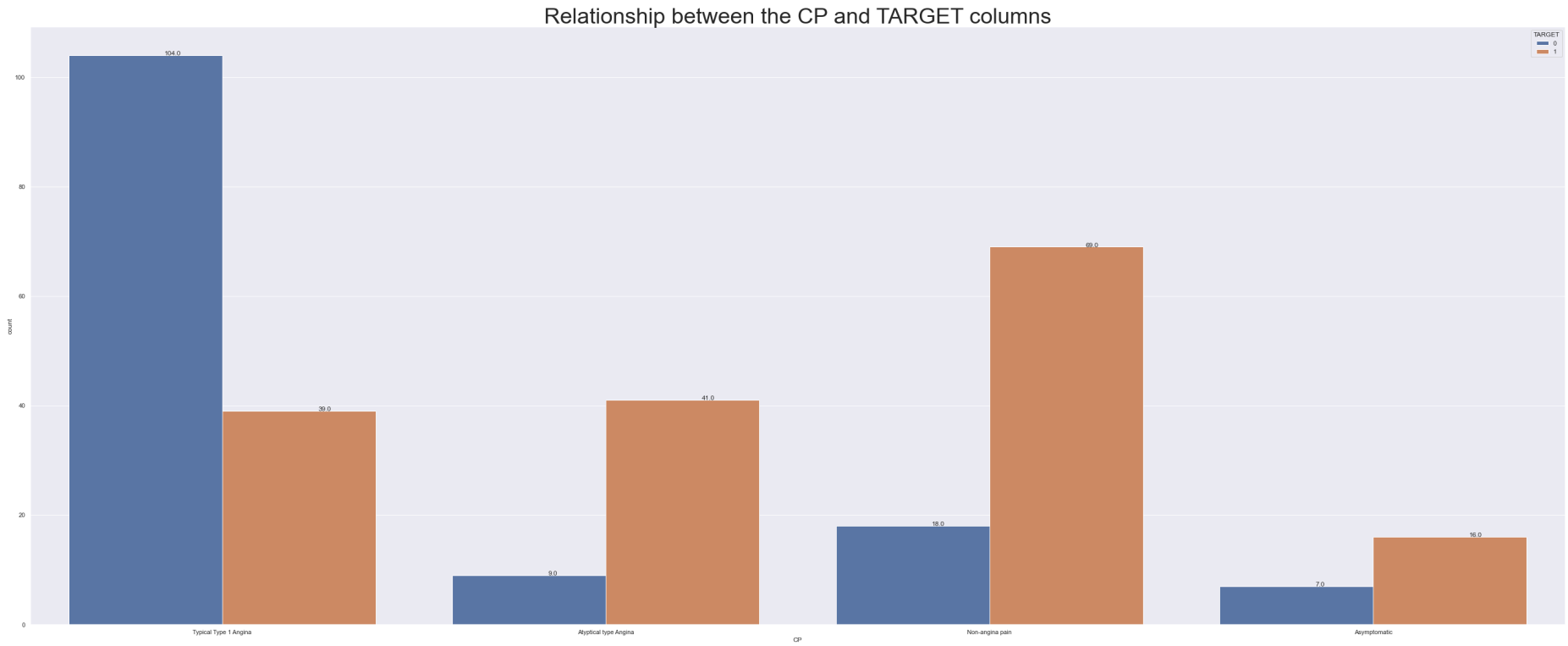
In [507]:

```
edu = sns.countplot(x='SEX', hue='TARGET', data=df)
edu.set_xticklabels(['FEMALE','MALE'])
edu.set_title('Relationship between the SEX and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



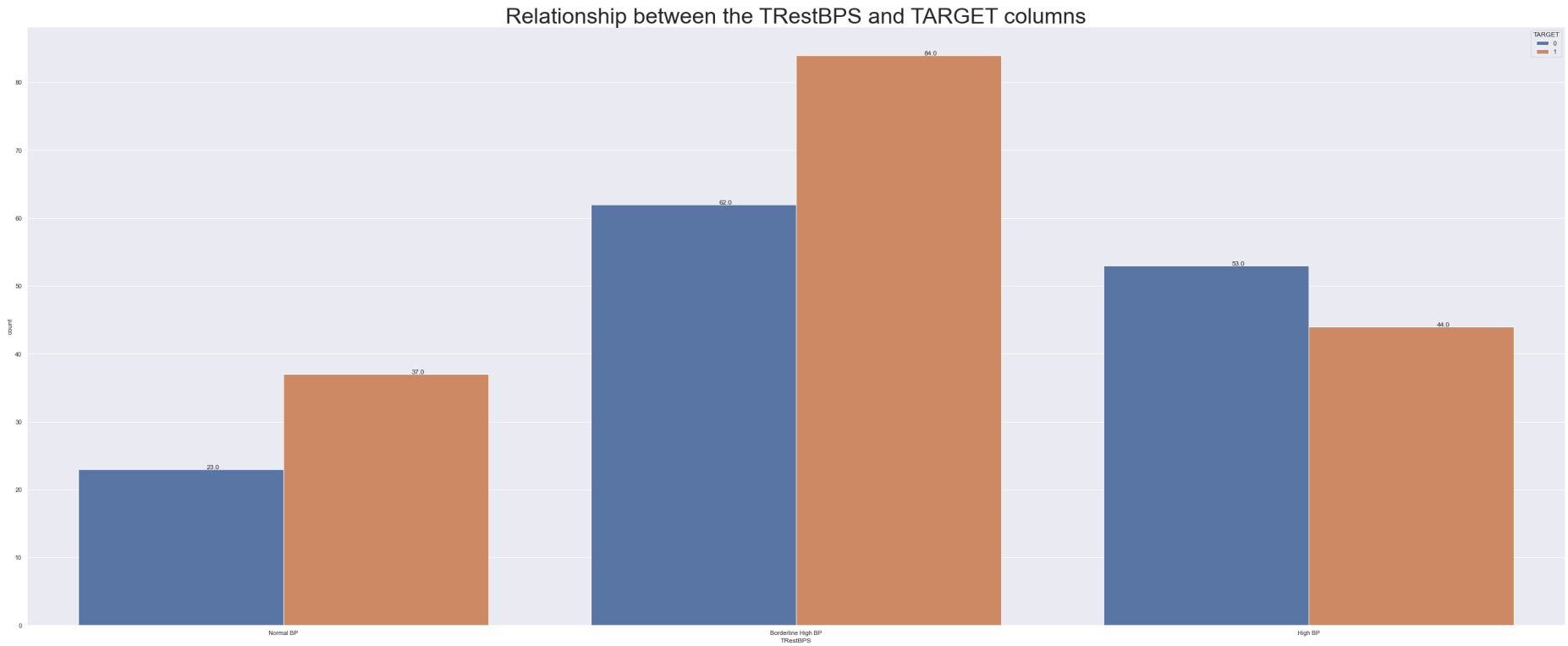
In [508]:

```
edu = sns.countplot(x='CP', hue='TARGET', data=df)
edu.set_xticklabels(['Typical Type 1 Angina','Atyptical type Angina','Non-angina pain','Asymptomatic'])
edu.set_title('Relationship between the CP and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



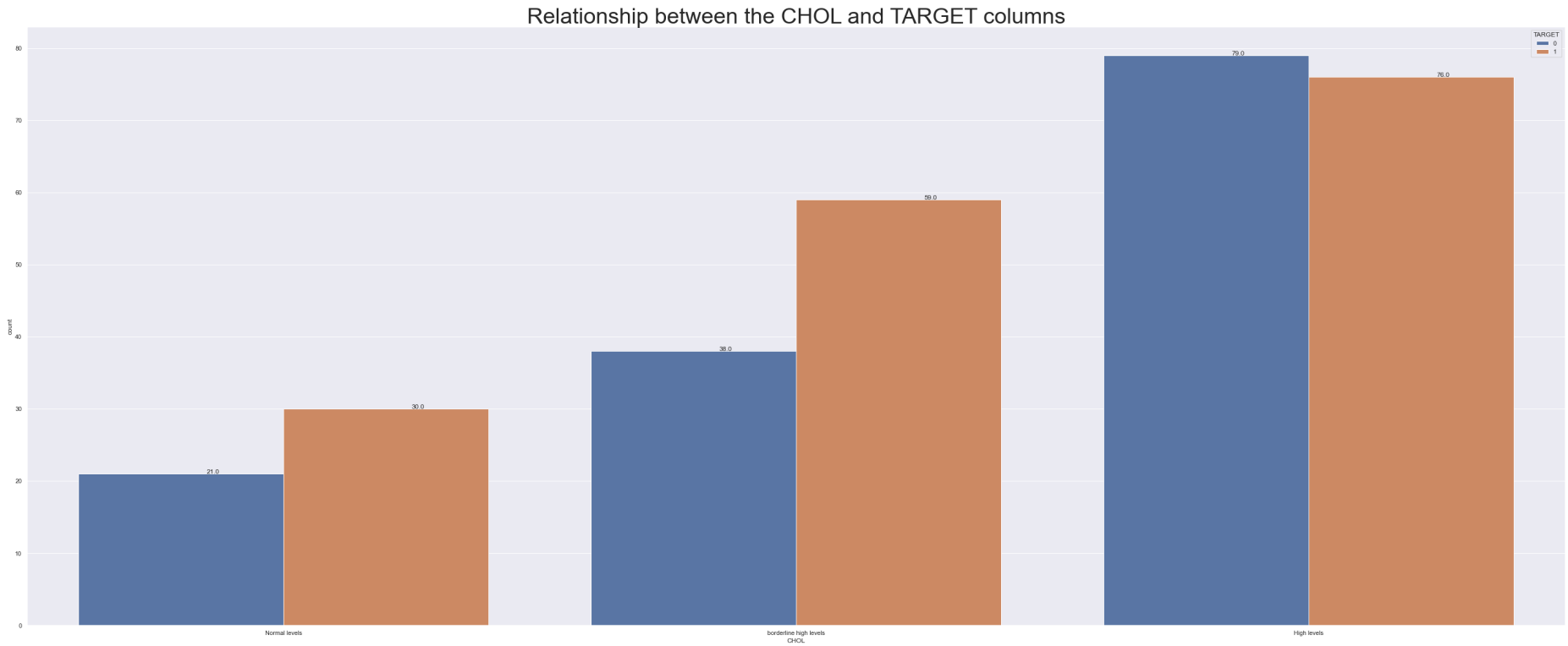
In [509]:

```
edu = sns.countplot(x='TRestBPS', hue='TARGET', data=df)
edu.set_xticklabels(['Normal BP','Borderline High BP','High BP'])
edu.set_title('Relationship between the TRestBPS and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



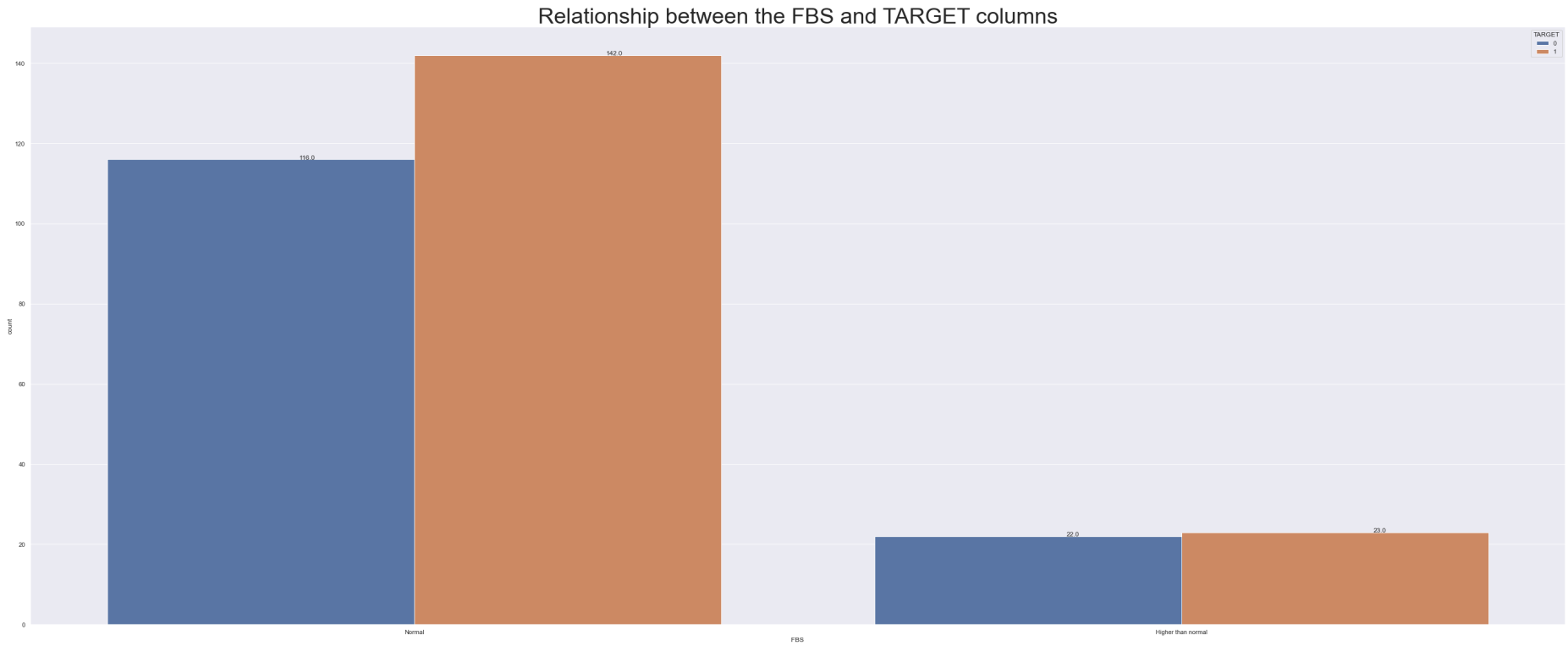
In [510]:

```
edu = sns.countplot(x='CHOL', hue='TARGET', data=df)
edu.set_xticklabels(['Normal levels','borderline high levels','High levels'])
edu.set_title('Relationship between the CHOL and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



In [511]:

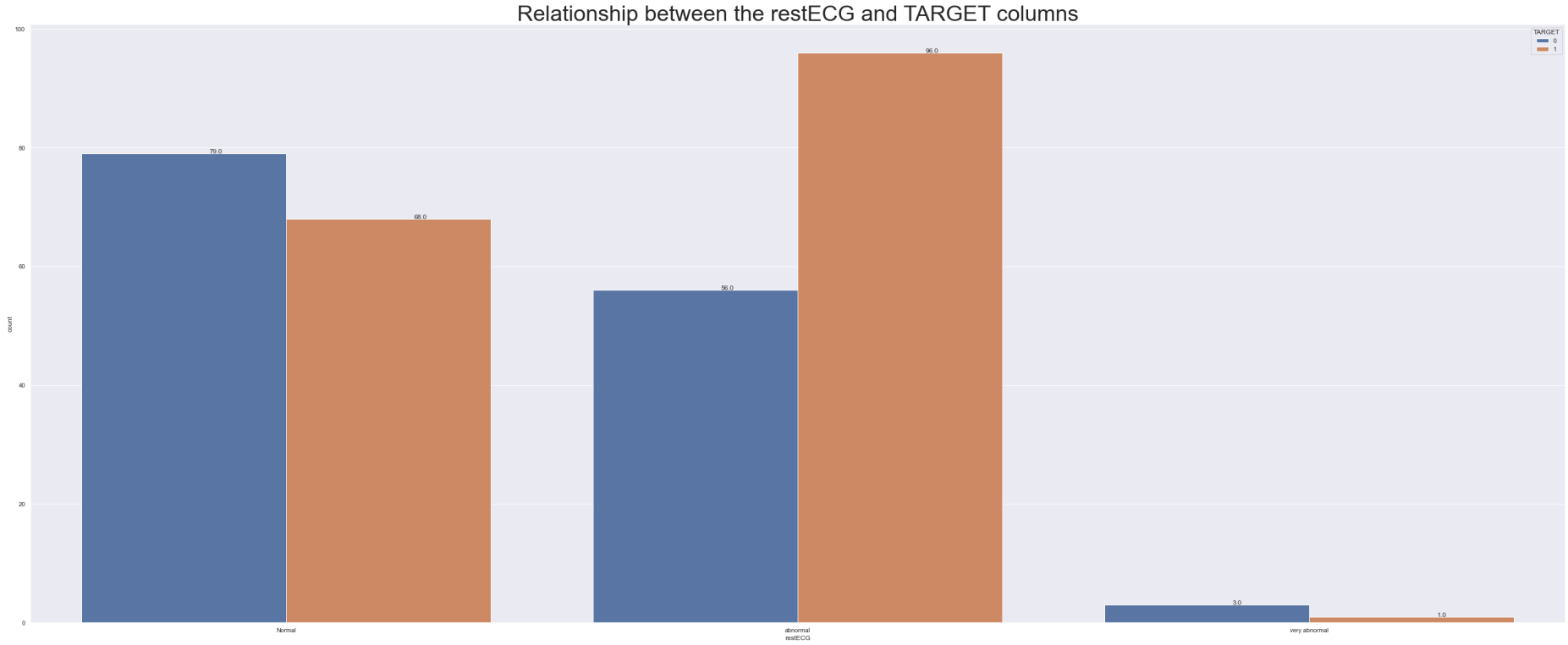
```
edu = sns.countplot(x='FBS', hue='TARGET', data=df)
edu.set_xticklabels(['Normal','Higher than normal'])
edu.set_title('Relationship between the FBS and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



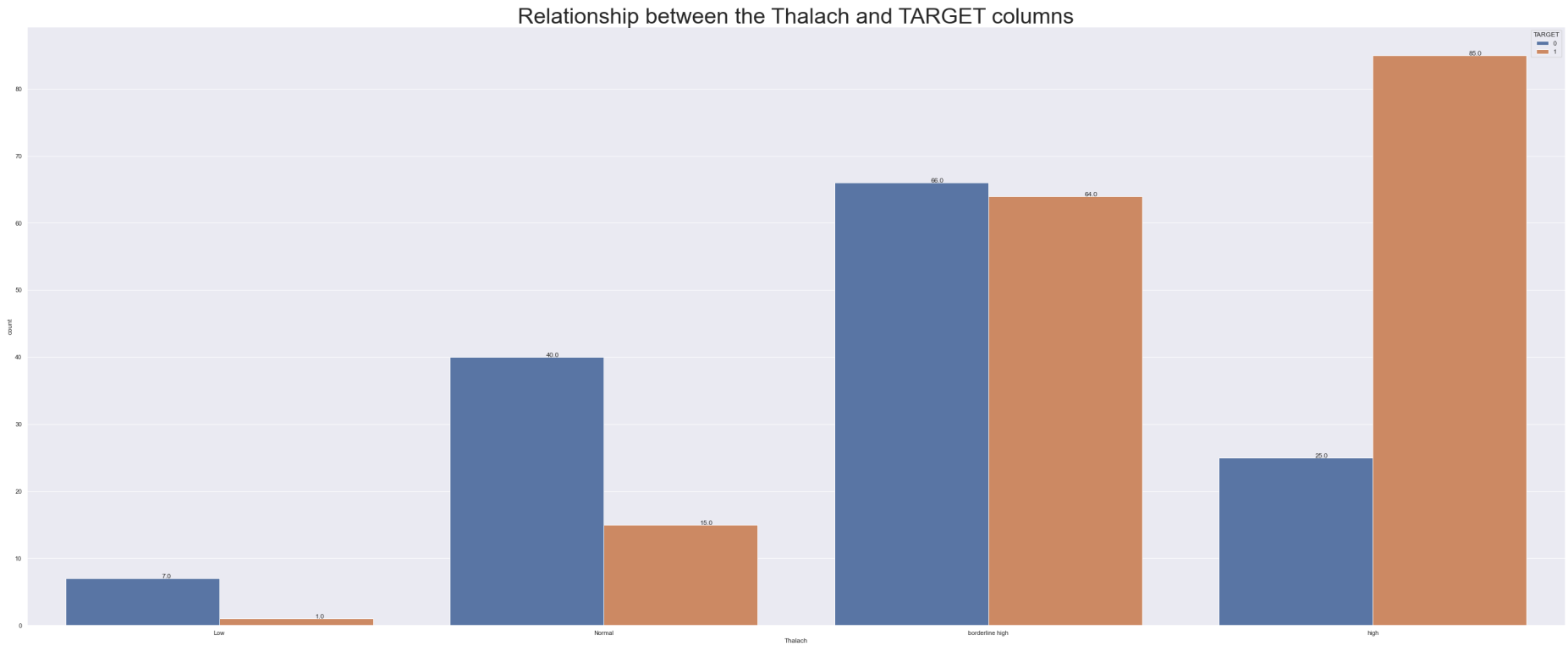
In [512]:
```
df['FBS'].value_counts()
```

Out[512]:
```
0    258
1     45
Name: FBS, dtype: int64
```

In [513]:
```
edu = sns.countplot(x='restECG', hue='TARGET', data=df)
edu.set_xticklabels(['Normal','abnormal', 'very abnormal'])
edu.set_title('Relationship between the restECG and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



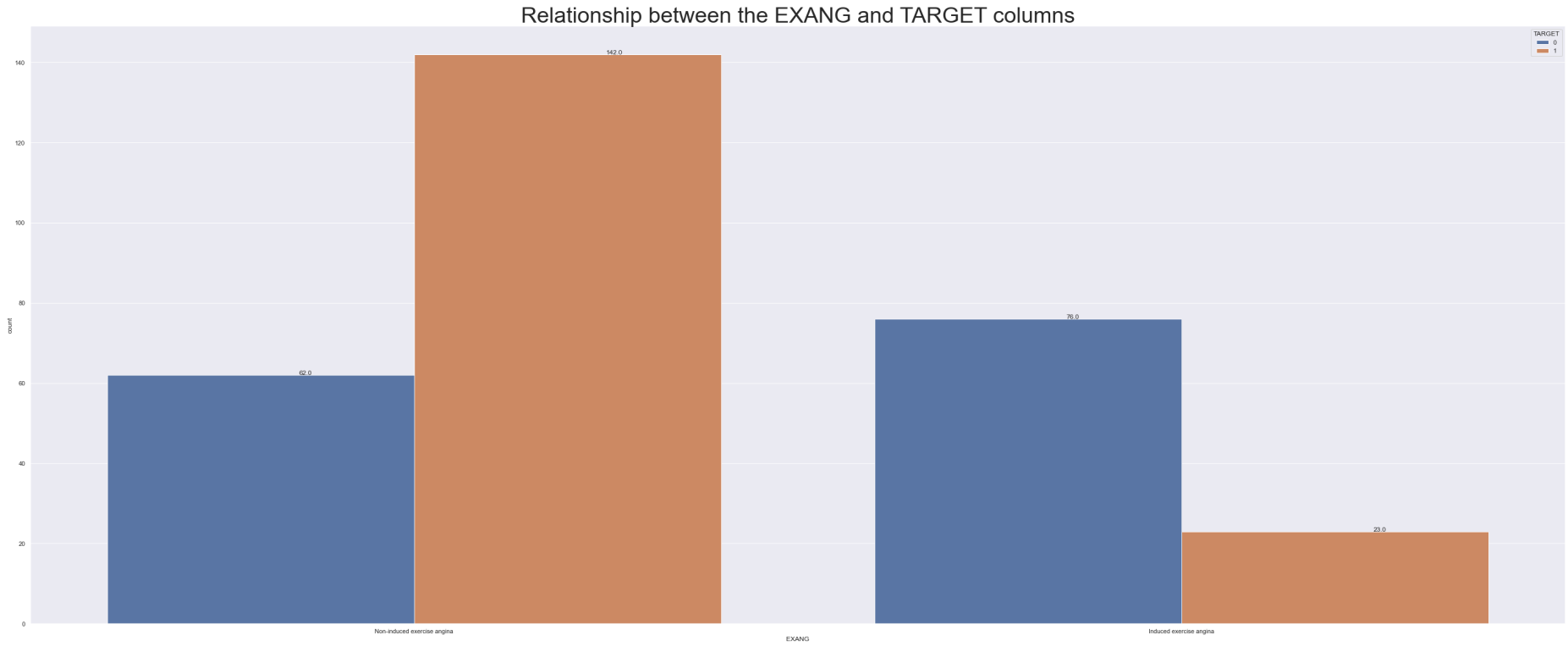
In [514]:
```
edu = sns.countplot(x='Thalach', hue='TARGET', data=df)
edu.set_xticklabels(['Low','Normal', 'borderline high','high'])
edu.set_title('Relationship between the Thalach and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



In [ ]:
```

```

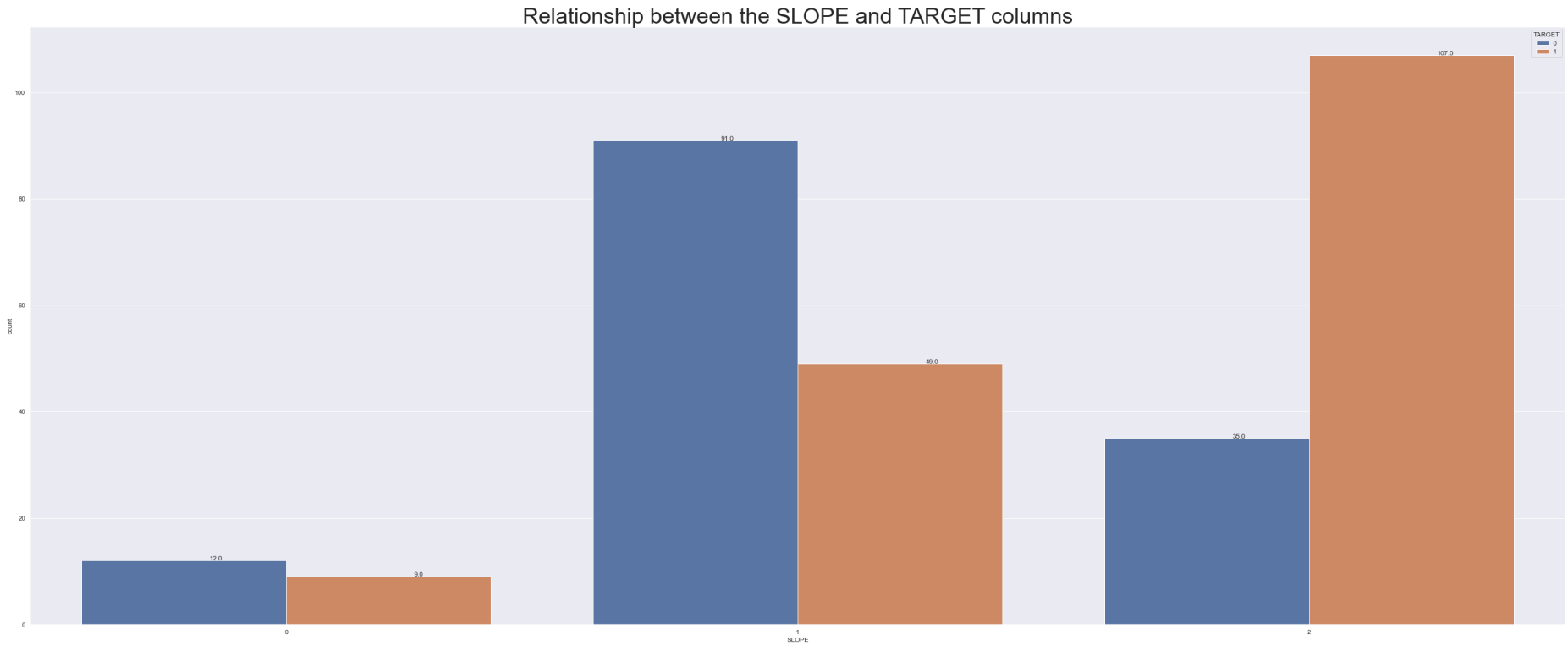
In [515]:

```
edu = sns.countplot(x='EXANG', hue='TARGET', data=df)
edu.set_xticklabels(['Non-induced exercise angina', 'Induced exercise angina'])
edu.set_title('Relationship between the EXANG and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



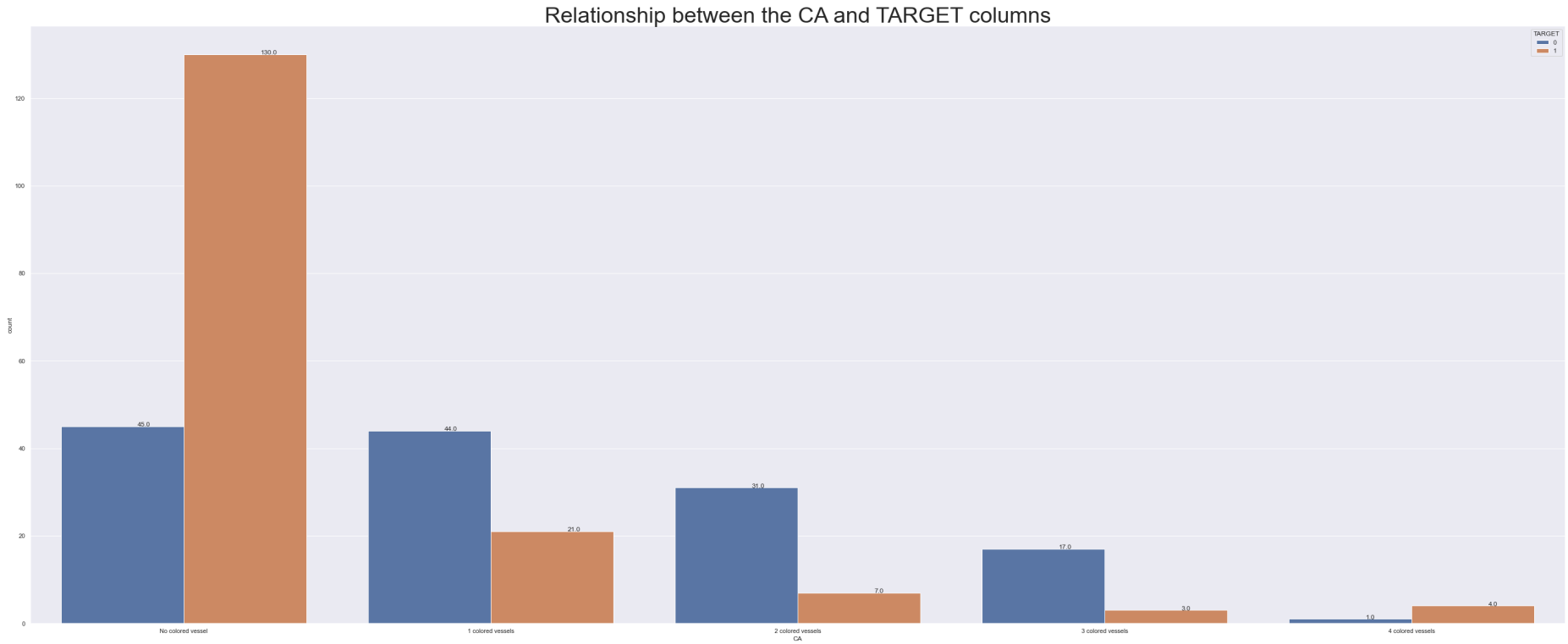
In [516]:

```
edu = sns.countplot(x='SLOPE', hue='TARGET', data=df)
edu.set_xticklabels(['0','1','2'])
edu.set_title('Relationship between the SLOPE and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



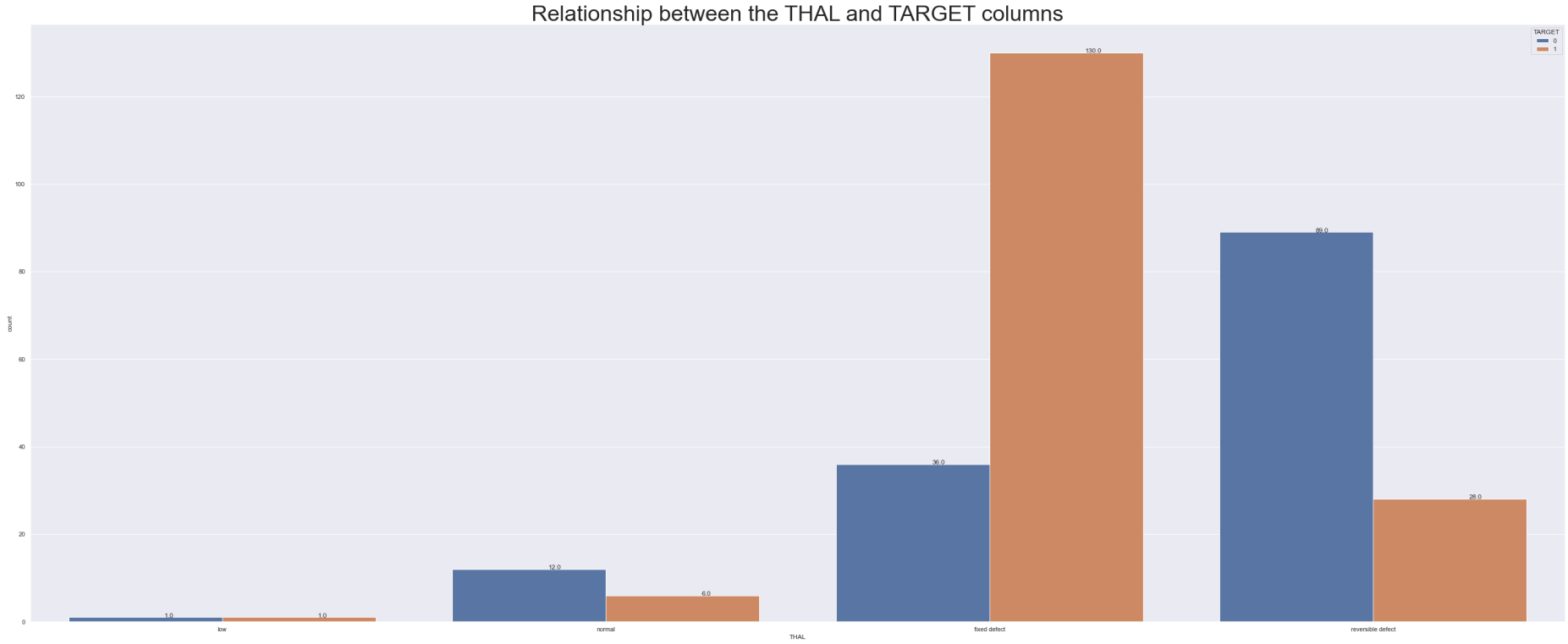
In [517]:

```
edu = sns.countplot(x='CA', hue='TARGET', data=df)
edu.set_xticklabels(['No colored vessel','1 colored vessels','2 colored vessels','3 colored vessels','4 colo
edu.set_title('Relationship between the CA and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



In [518]:

```
edu = sns.countplot(x='THAL', hue='TARGET', data=df)
edu.set_xticklabels(['low', 'normal','fixed defect','reversible defect'])
edu.set_title('Relationship between the THAL and TARGET columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```

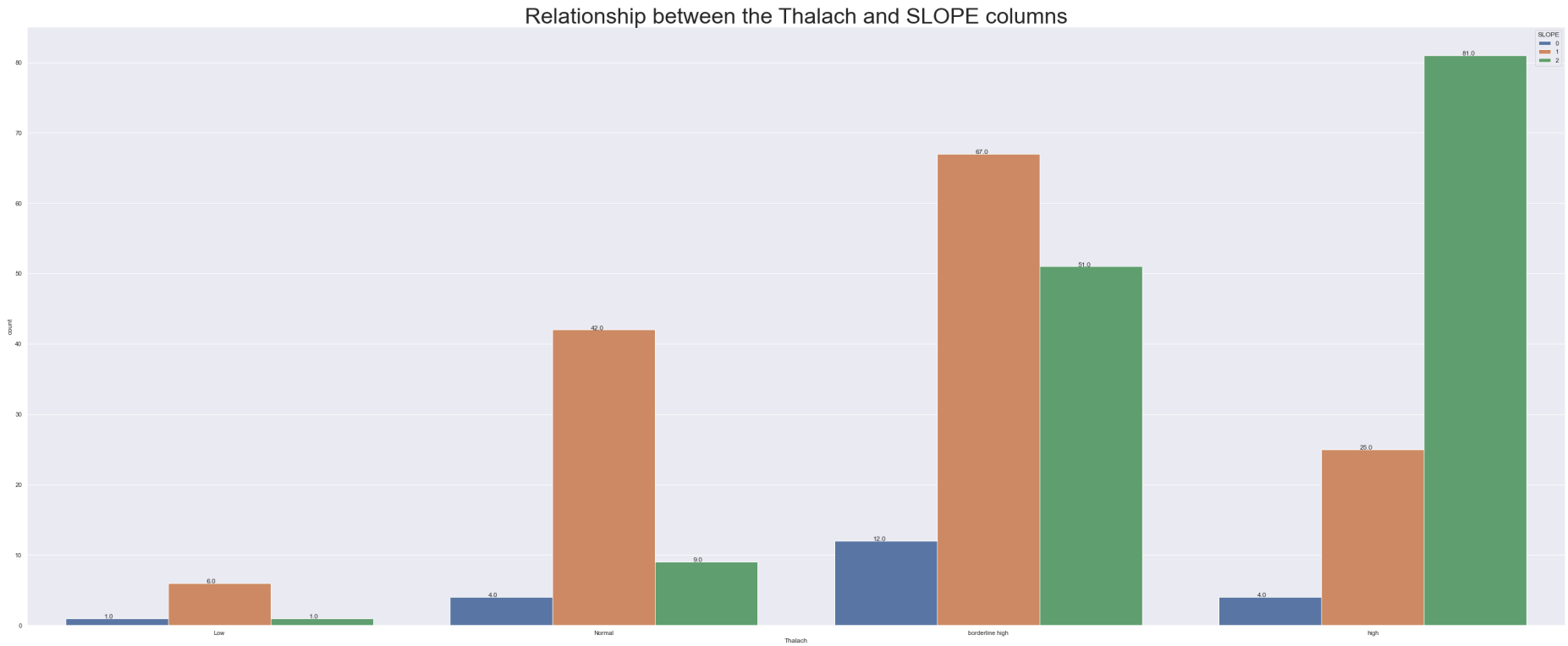


In [519]:
```
otting the Distributions and Relationships Between Specific Features
ots between "Thalach" and 'slope'
            "Thalach" and 'CP'
            'Age' and 'TRestBPS'
            'Age' and 'CA'
            'Thalach' and 'CHOL'
            'Thalach' and 'CA'
            'Thalach' and 'restECG'
The plots between specific variables I have made below all have a relationship as one variable increases,
 other also increases while for AGE and CA, 'Thalach' and 'CHOL' and 'Thalach' and 'CA' the reverse is the case
```

Out[519]:
```
"The plots between specific variables I have made below all have a relationship as one variable increases, \n
the other also increases while for AGE and CA, 'Thalach' and 'CHOL' and 'Thalach' and 'CA' the reverse is the
case "
```

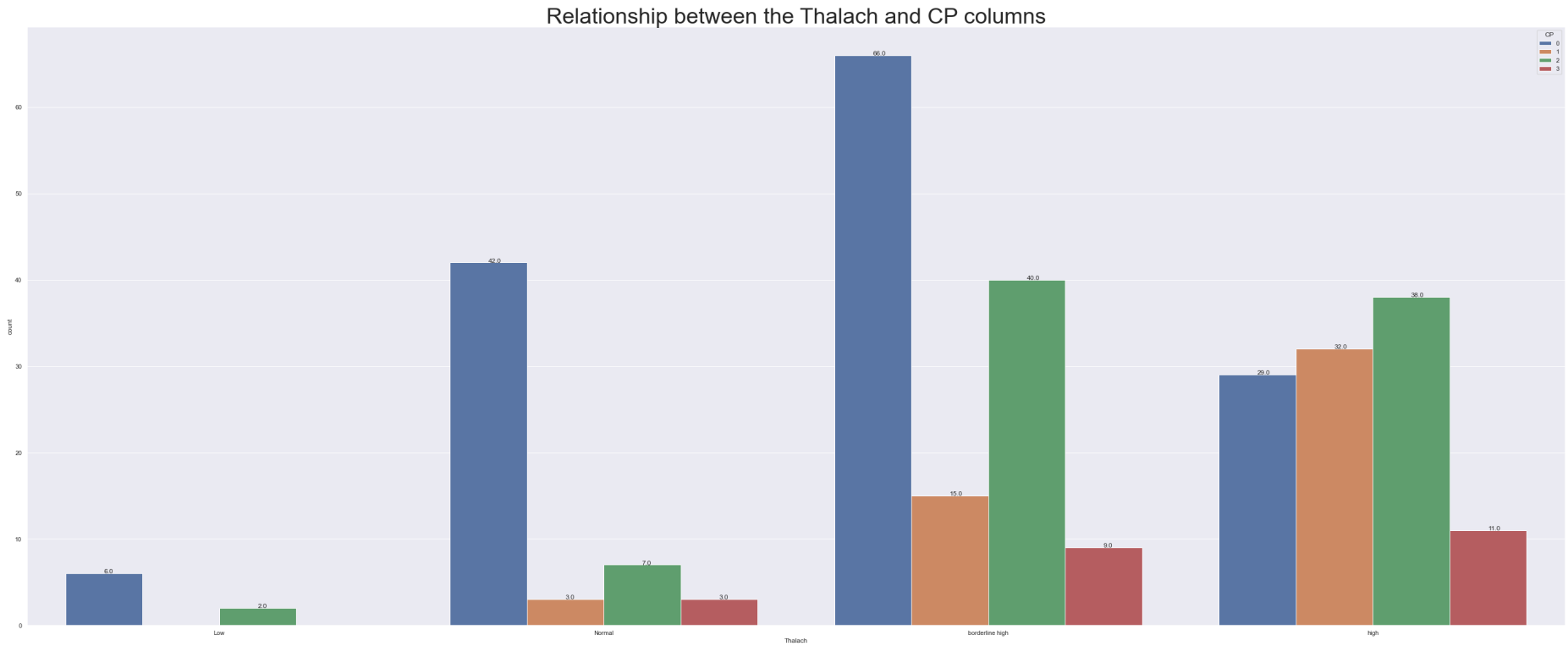
In [520]:

```
edu = sns.countplot(x='Thalach', hue='SLOPE', data=df)
edu.set_xticklabels(['Low','Normal', 'borderline high','high'])
edu.set_title('Relationship between the Thalach and SLOPE columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.10, p.get_height()+0.01))
```



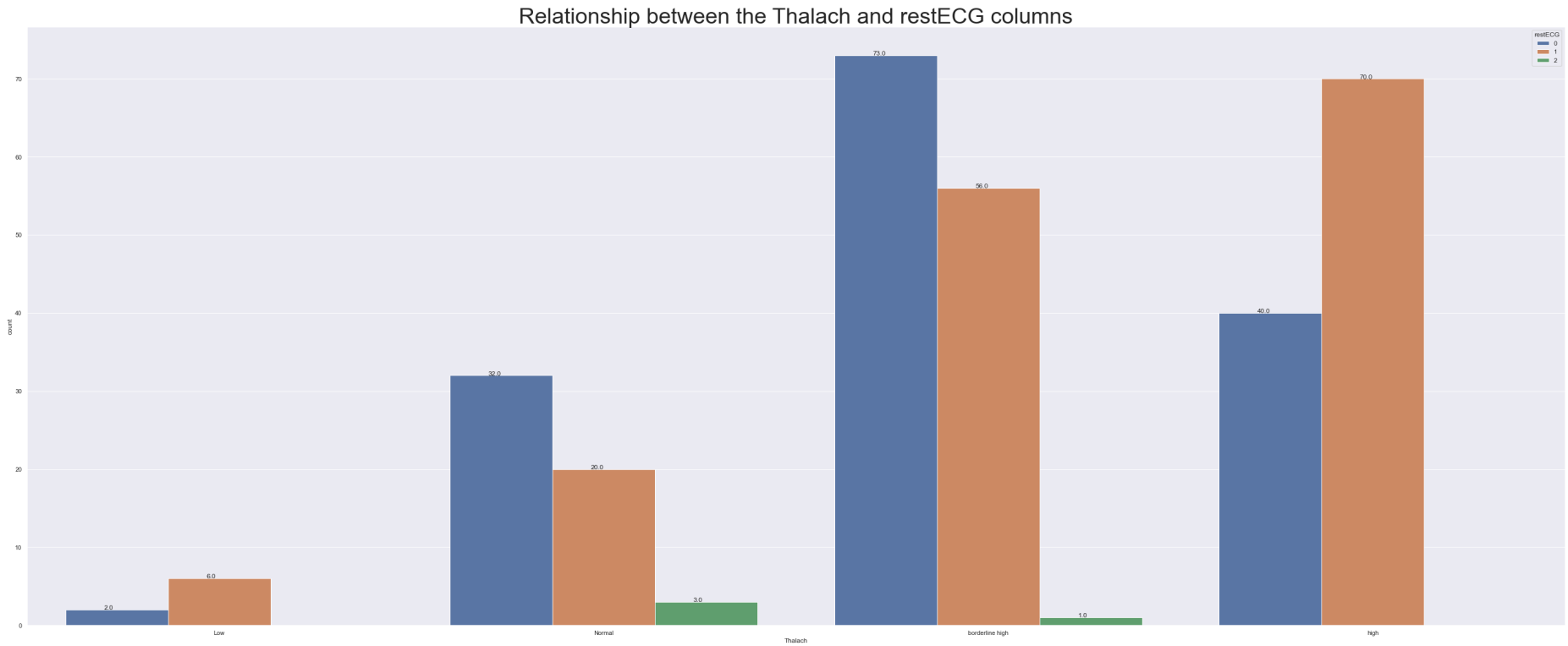
In [521]:

```
edu = sns.countplot(x='Thalach', hue='CP', data=df)
edu.set_xticklabels(['Low','Normal', 'borderline high','high'])
edu.set_title('Relationship between the Thalach and CP columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.10, p.get_height()+0.01))
```



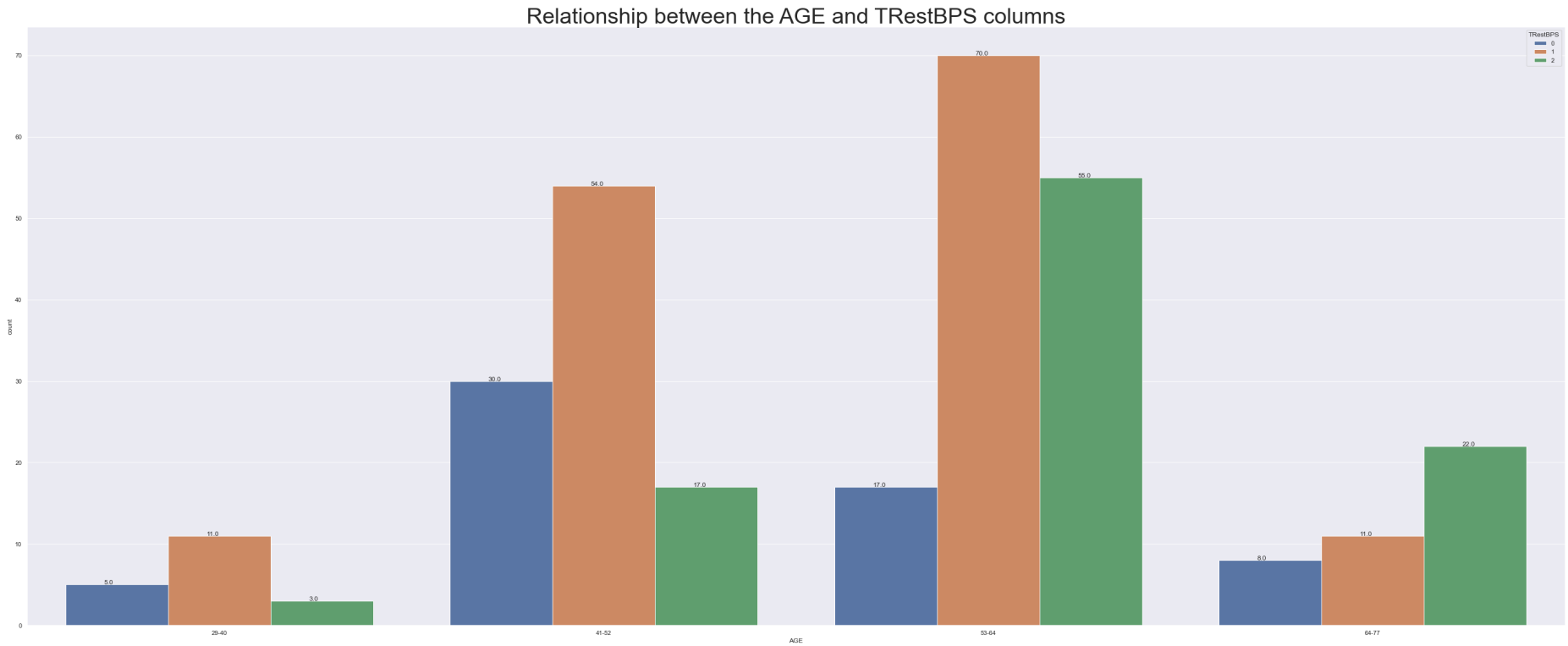
In [522]:

```
edu = sns.countplot(x='Thalach', hue='restECG', data=df)
edu.set_xticklabels(['Low','Normal', 'borderline high','high'])
edu.set_title('Relationship between the Thalach and restECG columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.10, p.get_height()+0.01))
```



In [523]:

```
edu = sns.countplot(x='AGE', hue='TRestBPS', data=df)
edu.set_xticklabels(['29-40','41-52','53-64','64-77'])
edu.set_title('Relationship between the AGE and TRestBPS columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.10, p.get_height()+0.01))
```

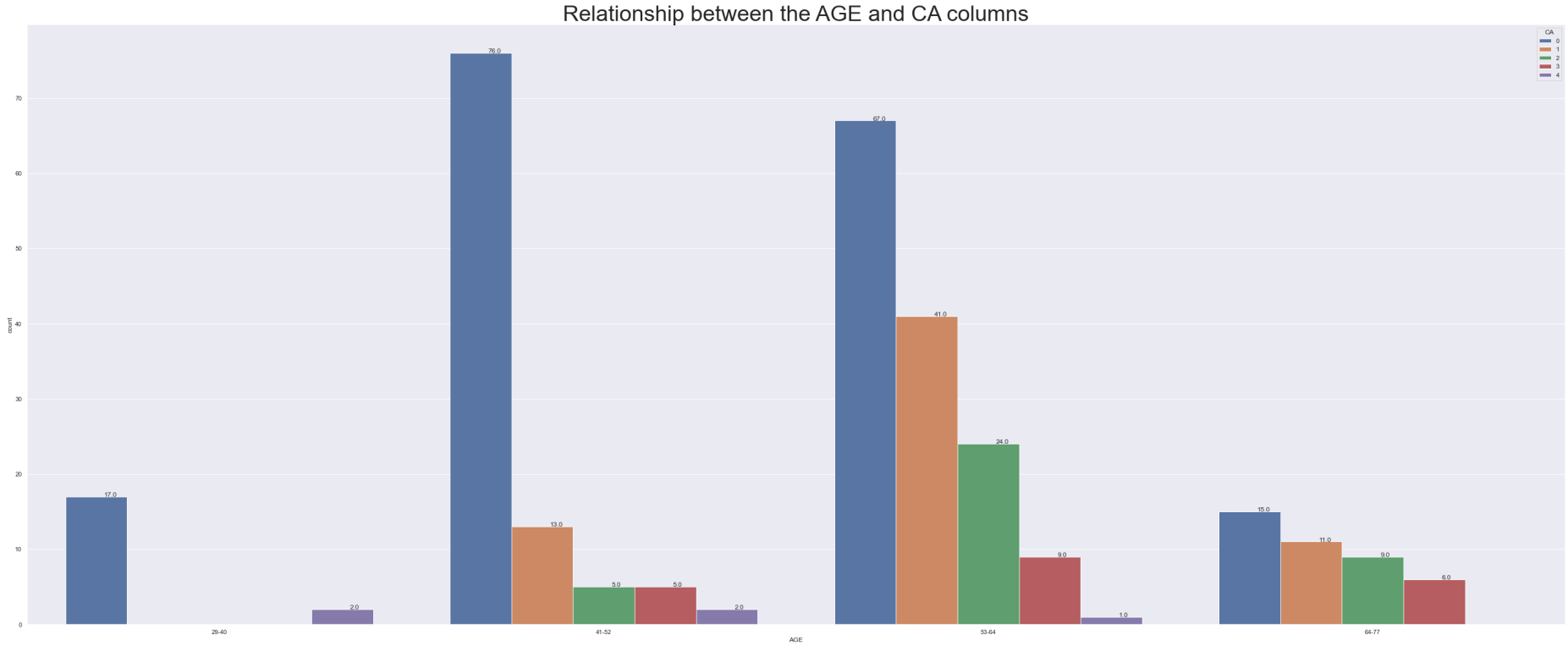


In [ ]:
```
1
```

In [524]:
```
1 df.FBS.unique()
```

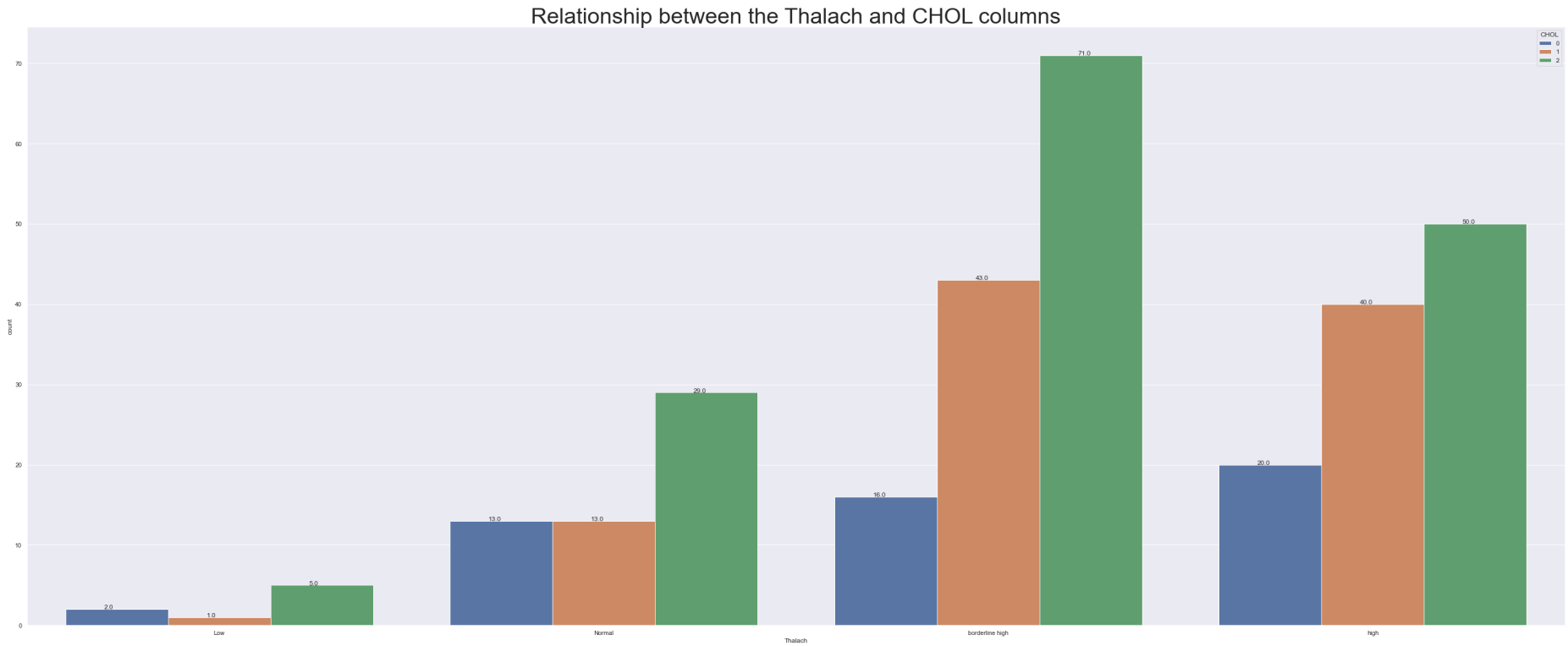
Out[524]:
```
array([1, 0])
```

In [525]:
```
1 edu = sns.countplot(x='AGE', hue='CA', data=df)
2 edu.set_xticklabels(['29-40','41-52','53-64','64-77'])
3 edu.set_title('Relationship between the AGE and CA columns', fontsize=40)
4 '''for p in ax.patches:
5         ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''
6
7 for p in edu.patches:
8    edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.10, p.get_height()+0.01))
```



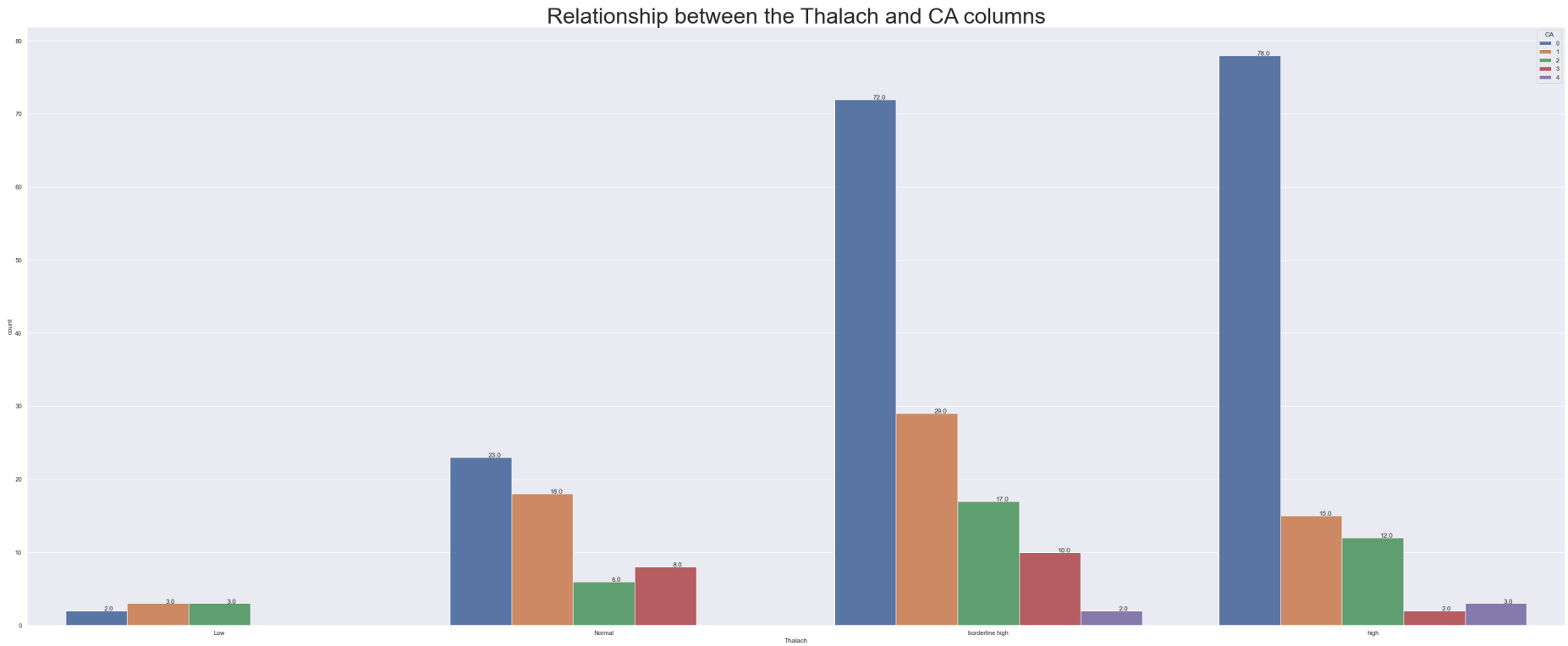
In [526]:

```
edu = sns.countplot(x='Thalach', hue='CHOL', data=df)
edu.set_xticklabels(['Low','Normal', 'borderline high','high'])
edu.set_title('Relationship between the Thalach and CHOL columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.10, p.get_height()+0.01))
```



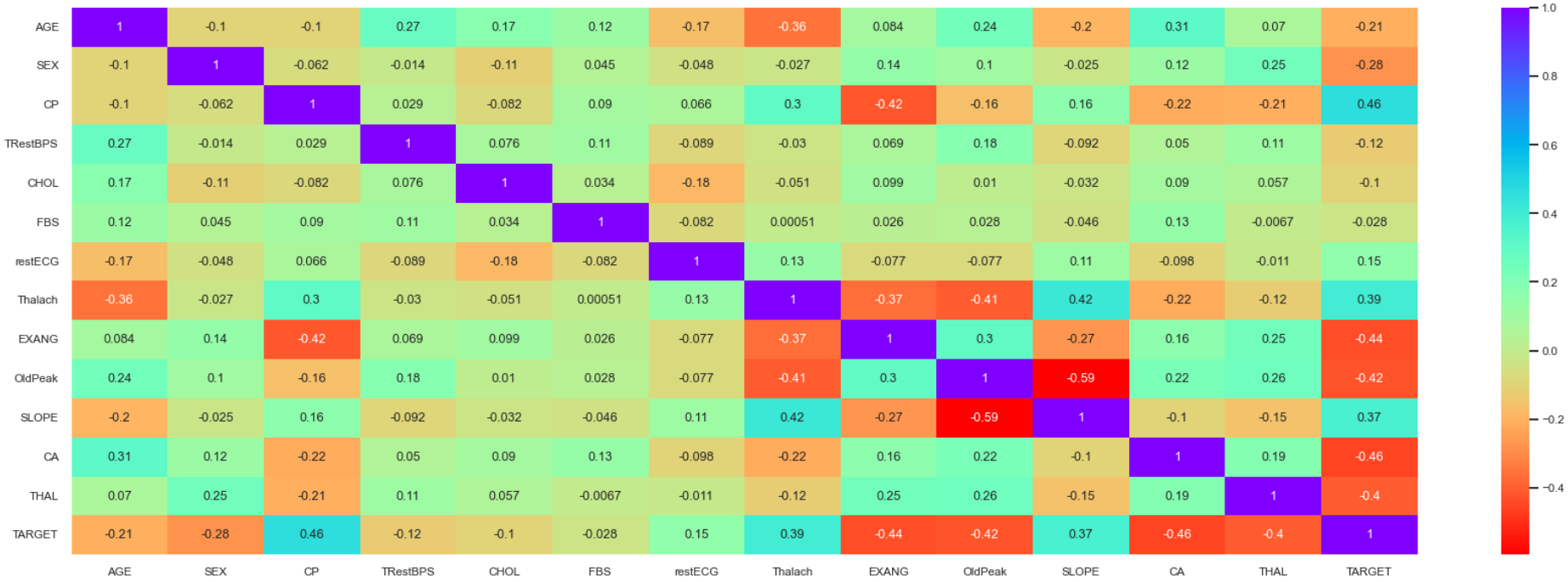
In [527]:

```
edu = sns.countplot(x='Thalach', hue='CA', data=df)
edu.set_xticklabels(['Low','Normal', 'borderline high','high'])
edu.set_title('Relationship between the Thalach and CA columns', fontsize=40)
'''for p in ax.patches:
        ax.annotate('%{:.1f}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))'''

for p in edu.patches:
   edu.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.10, p.get_height()+0.01))
```



In [528]:
```
#Evaluating the Correlation between Columns Using a Heatmap

#Plot the heatmap for all the columns in the DataFrame (other than the ID column) by using sns.heatmap and
#keep the figure size as 30,10 for better visibility:

sns.set(rc={'figure.figsize':(30,10)})
sns.set_context("talk", font_scale=0.7)


#Use Spearman as the method parameter to compute Spearman's rank correlation coefficient:

sns.heatmap(df.iloc[:,0:].corr(method='spearman'),cmap='rainbow_r', annot=True)
```

Out[528]: <AxesSubplot:>



In [529]:
```
#FINAL NOTES

'''A younger female adult (between ages 29-52) with chest pain (anginia), thalassemia disease and also with


Features having a strong relationship with the outcome(TARGET) according to their decreasing rank are:

'CP'- 0.46
'Thalach'- 0.43
'SLOPE'- 0.37
'restECG'- 0.15
'''
```

Out[529]:
```
"A younger female adult (between ages 29-52) with chest pain (anginia), thalassemia disease and also with an
abnormal resting ECG..... is at more risk of having a heart disease.\n\n\nFeatures having a
strong relationship with the outcome(TARGET) according to their decreasing rank are:\n\n'CP'- 0.46\n'Thalac
h'- 0.43\n'SLOPE'- 0.37\n'restECG'- 0.15\n"
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```