In [1]:
```
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from operator import add
import seaborn as sns
%matplotlib inline
```

In [2]:
```
data=pd.read_csv('dataready.csv')
```

In [3]:
```
data
```

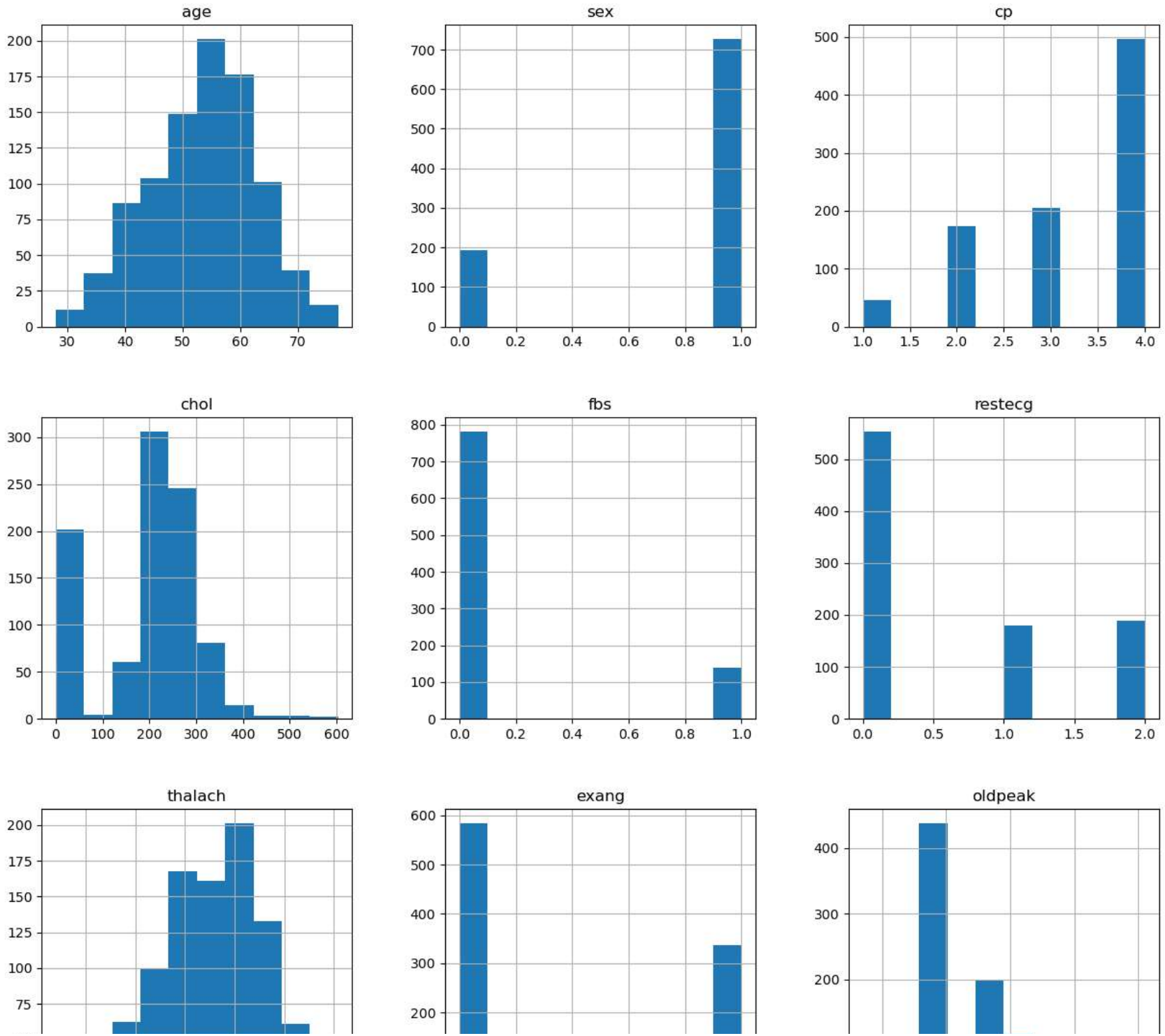
Out[3]:

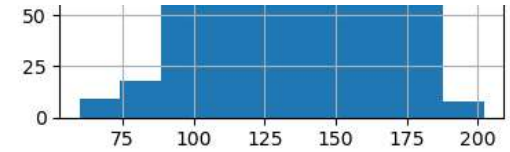
	age	sex	cp	chol	fbs	restecg	thalach	exang	oldpeak	ca	thal	target
0	63	1	1	233	1	2	150	0	2.3	0	6	0
1	67	1	4	286	0	2	108	1	1.5	3	3	1
2	67	1	4	229	0	2	129	1	2.6	2	7	1
3	37	1	3	250	0	0	187	0	3.5	0	3	0
4	41	0	2	204	0	2	172	0	1.4	0	3	0
...	...	...	...	...	...	...	...	...	...	...	...	...
915	52	1	4	331	0	0	94	1	2.5	0	3	1
916	54	0	3	294	0	1	100	1	0.0	0	3	1
917	56	1	4	342	1	0	150	1	3.0	0	3	1
918	58	0	2	393	0	0	110	1	1.0	0	7	1
919	65	1	4	275	0	1	115	1	1.0	0	3	1

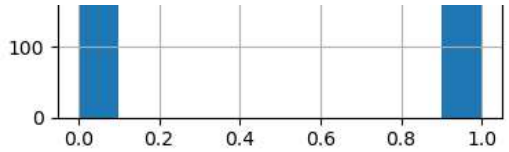
920 rows × 12 columns

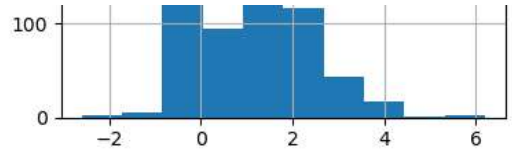
In [4]:

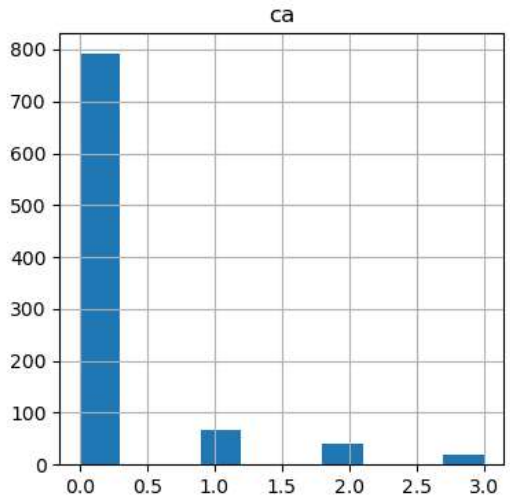
```
# plot histogram to see the distribution of the data
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
data.hist(ax = ax)
plt.show()
```

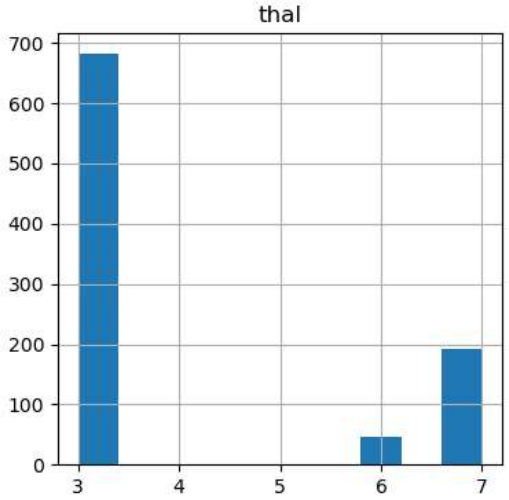

age
sex
cp
chol
fbs
restecg
thalach
exang
oldpeak

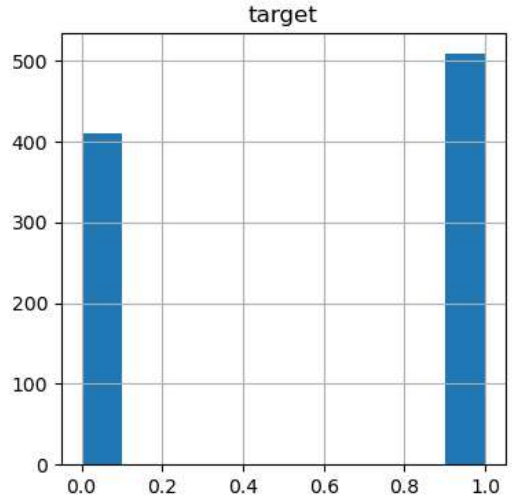

50
25
0
75
100
125
150
175
200


100
0
0.0
0.2
0.4
0.6
0.8
1.0

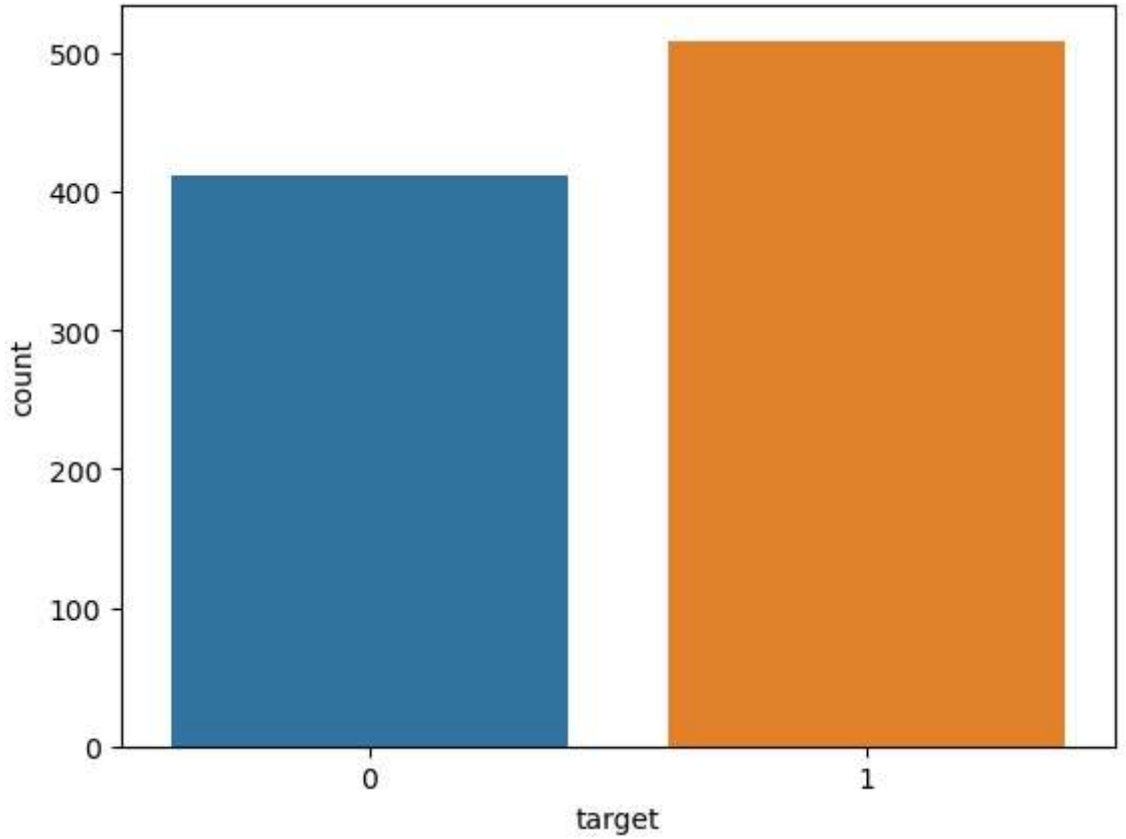

100
0
−2
0
2
4
6


ca
800
700
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0


thal
700
600
500
400
300
200
100
0
3
4
5
6
7

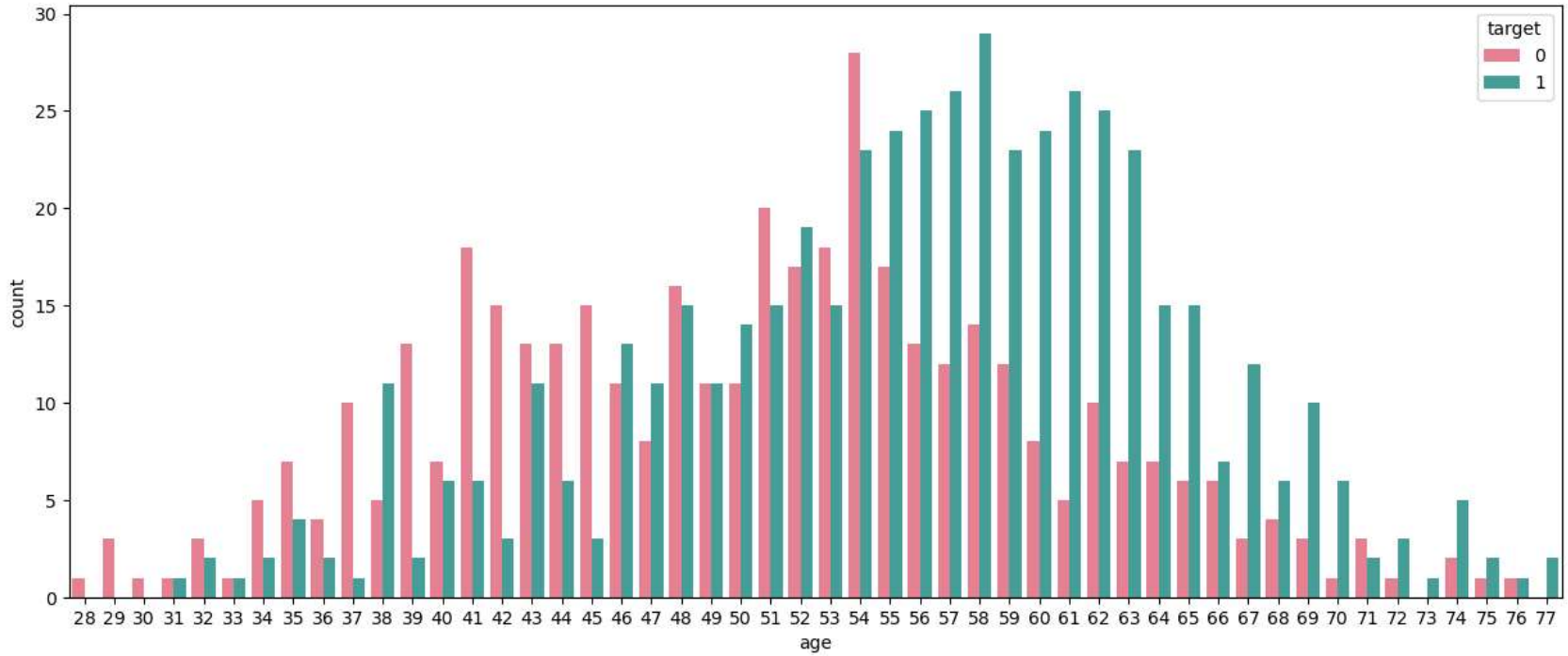

target
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0

In [5]:
```
sns.countplot(x='target',data=data)
plt.show()
cases = data.target.value_counts()
print(f"There are {cases[0]} patients without heart disease and {cases[1]} patients with the disease")
```




```
There are 411 patients without heart disease and 509 patients with the disease
```

In [6]:
```
plt.figure(figsize=(15,6))
sns.countplot(x='age',data = data, hue = 'target',palette='husl')
plt.show()
```



In [7]:
```
def stacked_barchart(data, title = None, ylabel = None, xlabel = None):
    default_colors = ['#008080', '#5f3c41', '#219AD8']
    # From raw value to percentage
    totals = data.sum(axis=1)
    bars = ((data.T / totals) * 100).T
    r = list(range(data.index.size))

    # Plot
    barWidth = 0.95
    names = data.index.tolist()
    bottom = [0] * bars.shape[0]

    # Create bars
    color_index = 0
    plots = []
    for bar in bars.columns:
        plots.append(plt.bar(r, bars[bar], bottom=bottom, color=default_colors[color_index], edgecolor
        bottom = list(map(add, bottom, bars[bar]))
        color_index = 0 if color_index >= len(default_colors) else color_index + 1

    # Custom x axis
    plt.title(title)
    plt.xticks(r, names)
    plt.xlabel(data.index.name if xlabel is None else xlabel)
    plt.ylabel(data.columns.name if ylabel is None else ylabel)
    ax = plt.gca()

    y_labels = ax.get_yticks()
    ax.set_yticklabels([str(y) + '%' for y in y_labels])

    flat_list = [item for sublist in data.T.values for item in sublist]
    for i, d in zip(ax.patches, flat_list):
        data_label = str(d) + " (" + str(round(i.get_height(), 2)) + "%)"
        ax.text(i.get_x() + 0.45, i.get_y() + 5, data_label, horizontalalignment='center', verticalali

    for item in ([ax.title]):
        item.set_fontsize(27)

    for item in ([ax.xaxis.label, ax.yaxis.label] + ax.get_xticklabels() + ax.get_yticklabels()):
        item.set_fontsize(24)

    legend = ax.legend(plots, bars.columns.tolist(), fancybox=True)
```

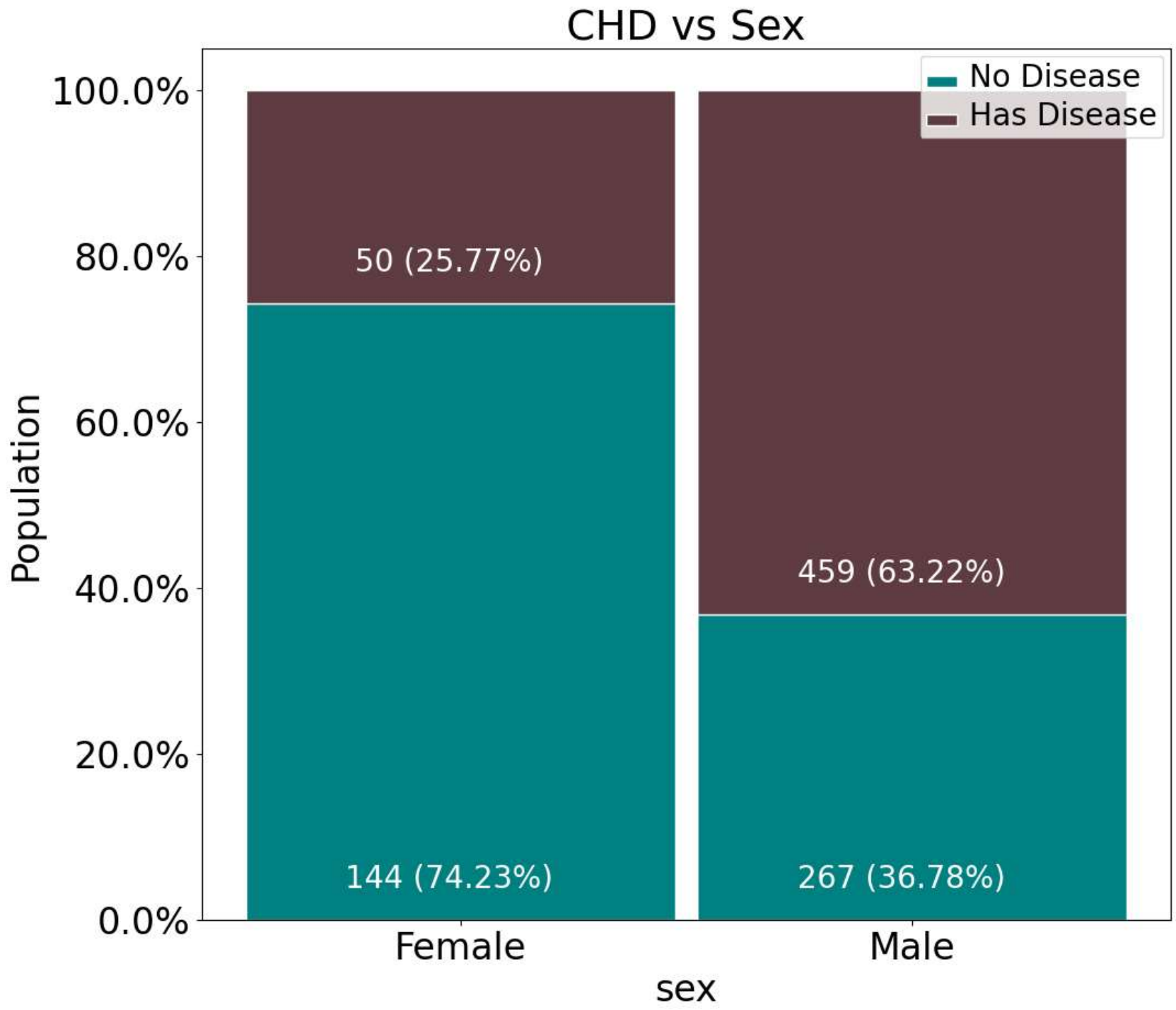
```
    plt.setp(legend.get_texts(), fontsize=20)
```

In [8]:

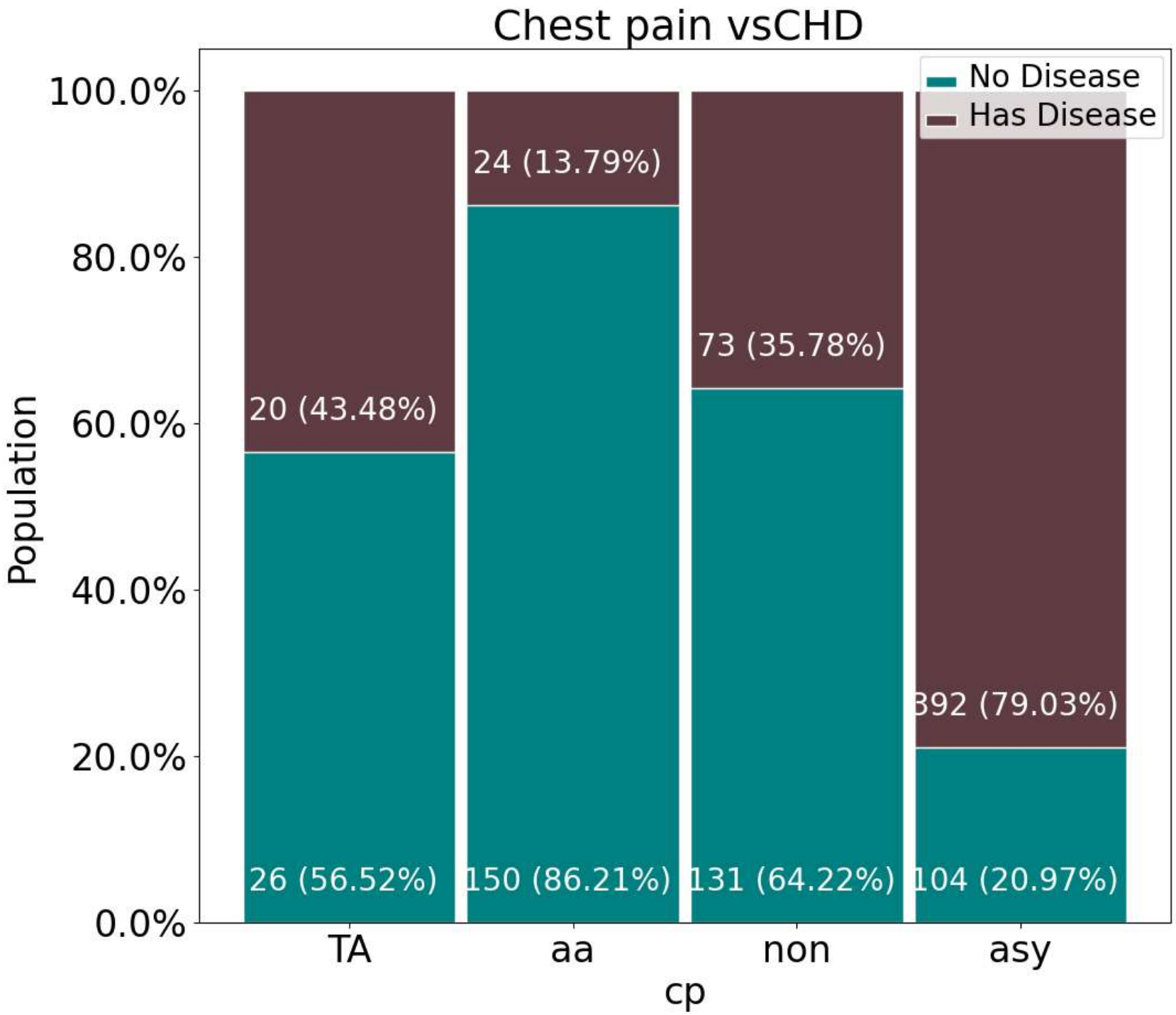
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2
```

```
<Figure size 2500x3500 with 0 Axes>
```

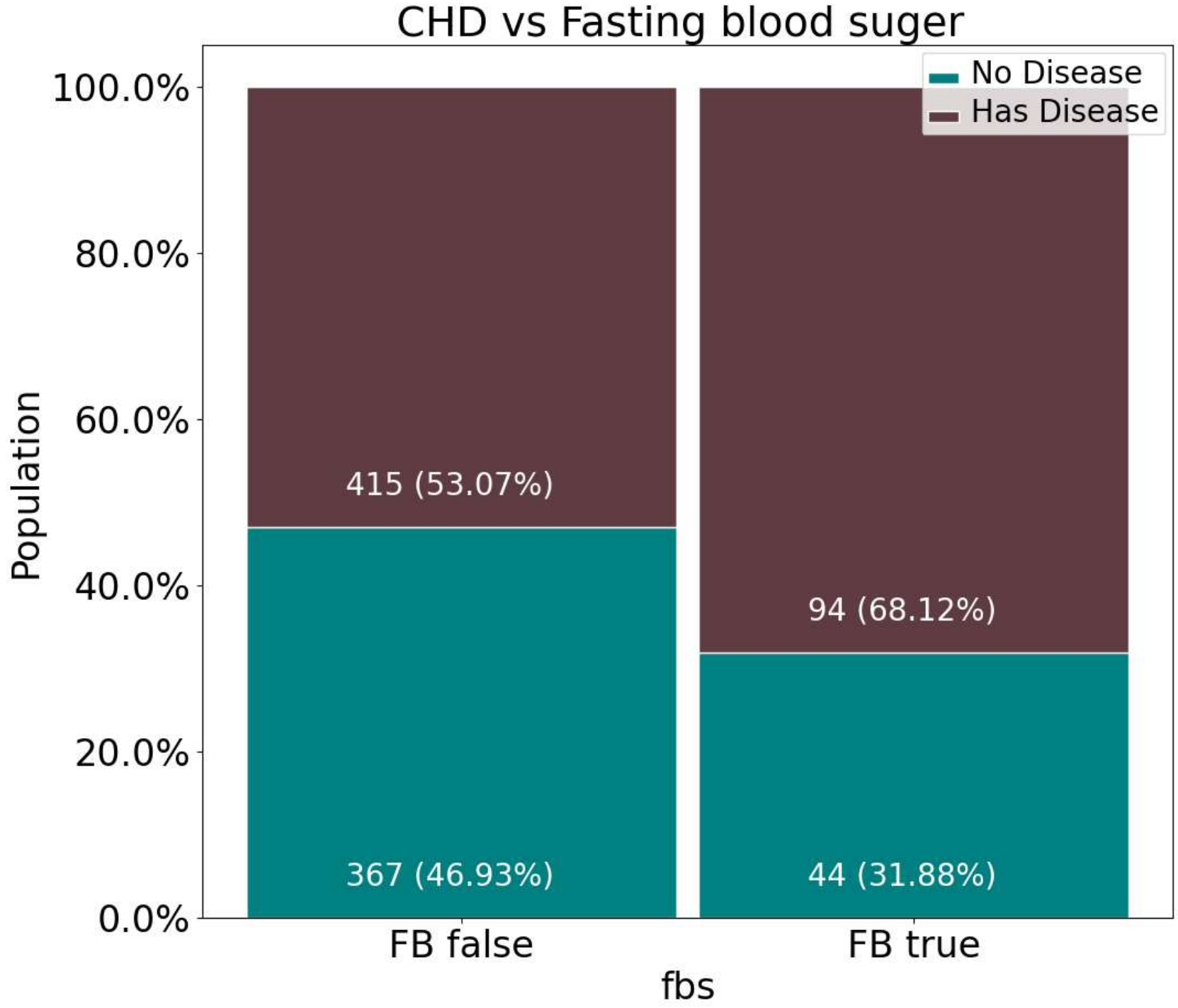
In [9]:
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2

#draw sex vs disease outcome
plt.subplot(grid_rows, grid_cols, 1)
temp = data[['sex','target']].groupby(['sex','target']).size().unstack('target')
temp.rename(index={0:'Female', 1:'Male'}, columns={0:'No Disease', 1:'Has Disease'}, inplace = True)
stacked_barchart(temp, title = 'CHD vs Sex', ylabel = 'Population')
```


CHD vs Sex
No Disease
Has Disease
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%
Population
50 (25.77%)
459 (63.22%)
144 (74.23%)
267 (36.78%)
Female
Male
sex

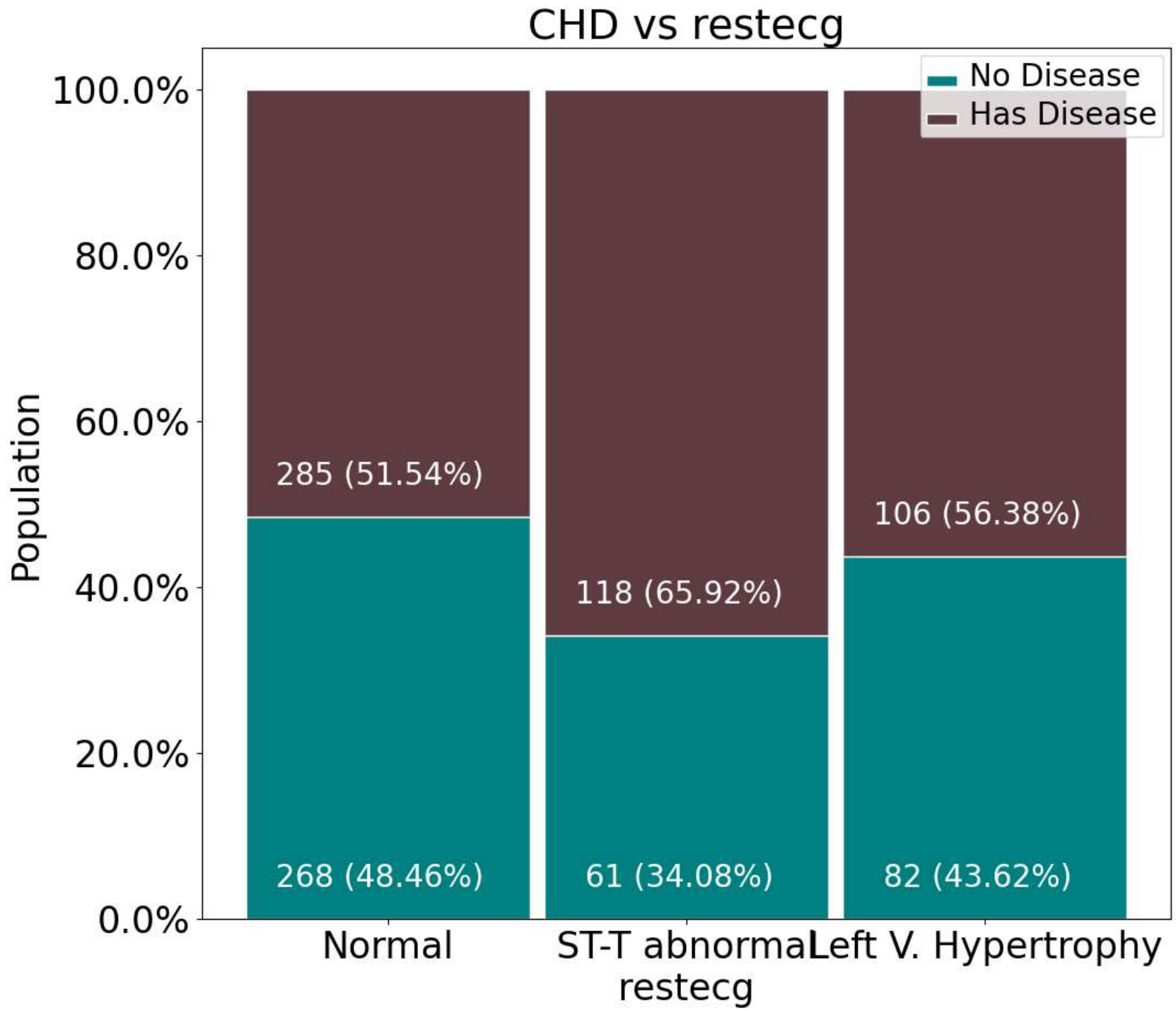
In [10]:
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2
#draw Chest pain  vs disease outcome
plt.subplot(grid_rows, grid_cols, 2)
temp = data[['cp','target']].groupby(['cp','target']).size().unstack('target')
temp.rename(index={1:'TA',2:'aa',3:'non',4:'asy'},
columns={0:'No Disease', 1:'Has Disease'}, inplace = True)
stacked_barchart(temp, title = 'Chest pain vsCHD ', ylabel = 'Population')
```


Chest pain vsCHD
No Disease
Has Disease
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%
Population
20 (43.48%)
24 (13.79%)
73 (35.78%)
392 (79.03%)
26 (56.52%)
150 (86.21%)
131 (64.22%)
104 (20.97%)
TA
aa
non
asy
cp

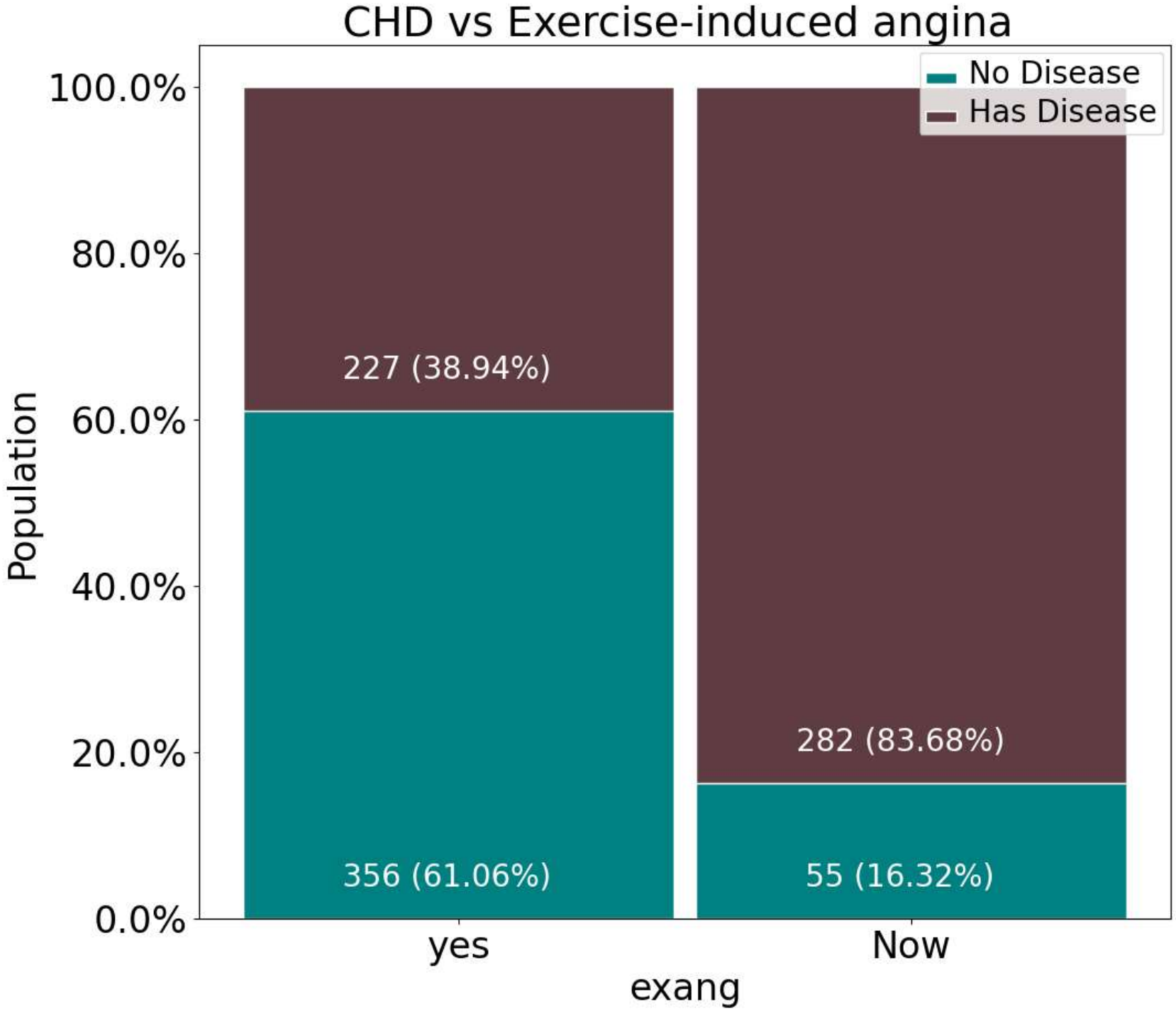
In [11]:
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2
#draw diabetes vs disease outcome
plt.subplot(grid_rows, grid_cols, 3)
temp = data[['fbs','target']].groupby(['fbs','target']).size().unstack('target')
temp.rename(index={0:'FB false', 1:'FB true'}, columns={0:'No Disease', 1:'Has Disease'}, inplace = Tr
stacked_barchart(temp, title = 'CHD vs Fasting blood suger ', ylabel = 'Population')
```


CHD vs Fasting blood suger
No Disease
Has Disease
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%
Population
415 (53.07%)
367 (46.93%)
94 (68.12%)
44 (31.88%)
FB false
FB true
fbs

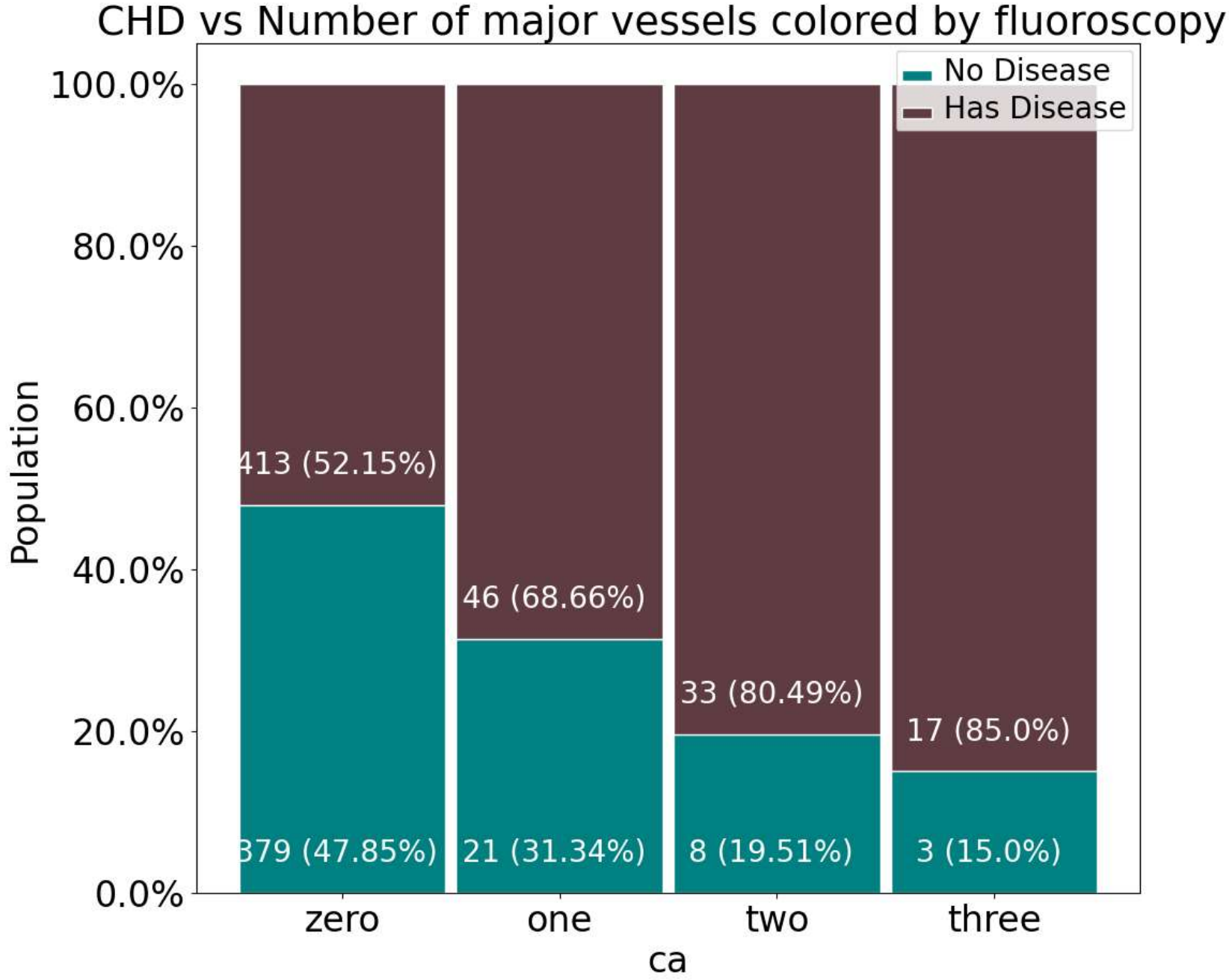
In [12]:
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2
#draw restecg vs disease outcome
plt.subplot(grid_rows, grid_cols, 4)
temp = data[['restecg','target']].groupby(['restecg','target']).size().unstack('target')
temp.rename(index={0:'Normal',1:'ST-T abnormal', 2:'Left V. Hypertrophy'}, columns={0:'No Disease', 1:
stacked_barchart(temp, title = 'CHD vs restecg', ylabel = 'Population')
```


CHD vs restecg
No Disease
Has Disease
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%
Population
285 (51.54%)
268 (48.46%)
118 (65.92%)
61 (34.08%)
106 (56.38%)
82 (43.62%)
Normal
ST-T abnorma
Left V. Hypertrophy
restecg

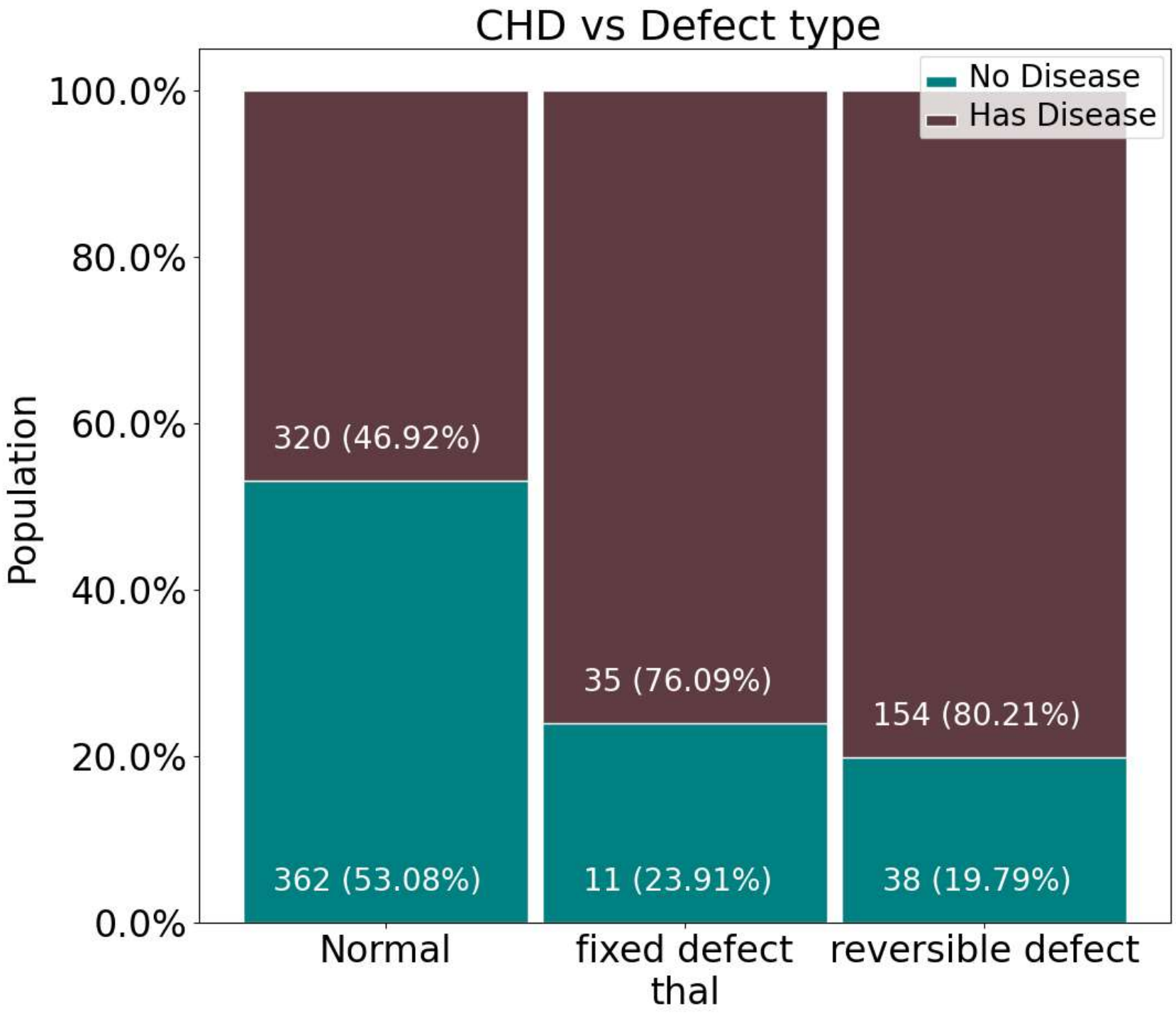
In [13]:
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2
#draw Exercise-induced angina  vs disease outcome
plt.subplot(grid_rows, grid_cols, 4)
temp = data[['exang','target']].groupby(['exang','target']).size().unstack('target')
temp.rename(index={0:'yes',1:'Now'}, columns={0:'No Disease', 1:'Has Disease'}, inplace = True)
stacked_barchart(temp, title = 'CHD vs Exercise-induced angina ', ylabel = 'Population')
```


CHD vs Exercise-induced angina
No Disease
Has Disease
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%
Population
227 (38.94%)
356 (61.06%)
282 (83.68%)
55 (16.32%)
yes
Now
exang

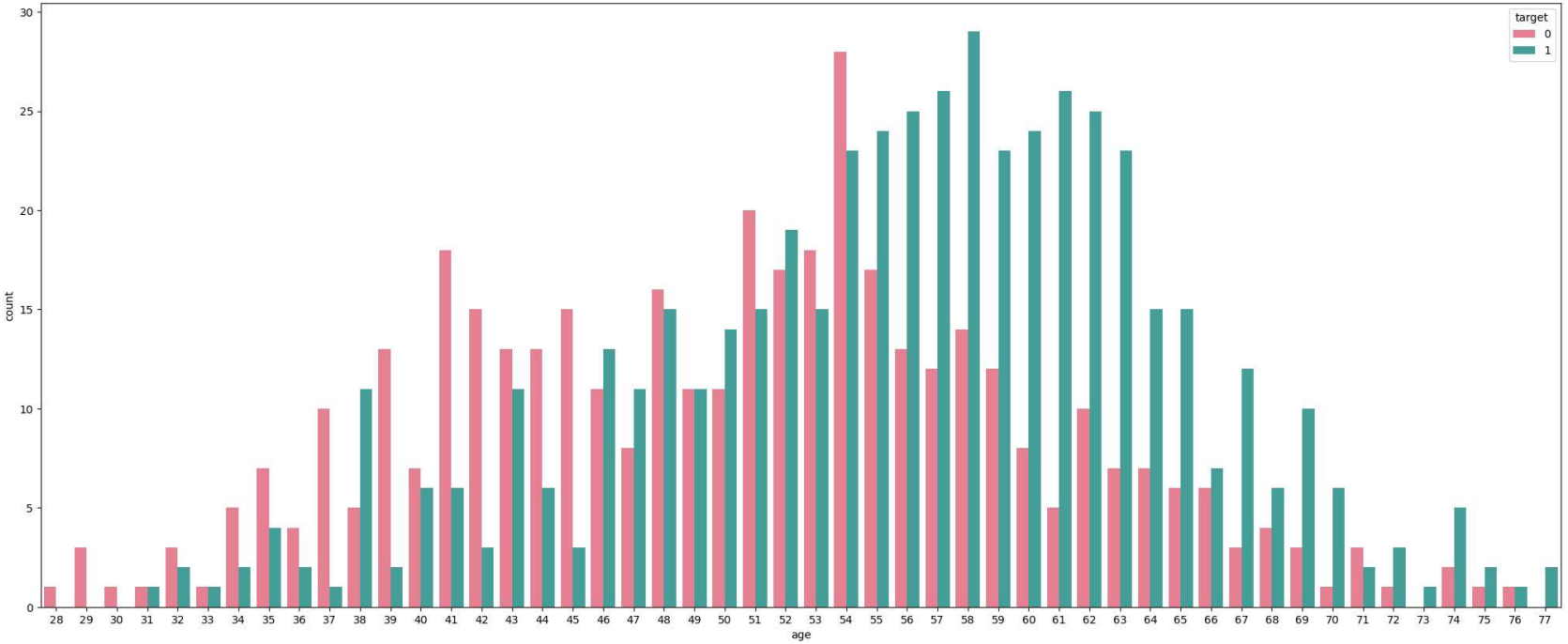
In [14]:
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2
#draw Number of major vessels colored by fluoroscopy  vs disease outcome
plt.subplot(grid_rows, grid_cols, 4)
temp = data[['ca','target']].groupby(['ca','target']).size().unstack('target')
temp.rename(index={0:'zero',1:'one',2:'two',3:'three'}, columns={0:'No Disease', 1:'Has Disease'}, inp
stacked_barchart(temp, title = 'CHD vs Number of major vessels colored by fluoroscopy ', ylabel = 'Pop
```


CHD vs Number of major vessels colored by fluoroscopy
No Disease
Has Disease
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%
Population
413 (52.15%)
379 (47.85%)
46 (68.66%)
21 (31.34%)
33 (80.49%)
8 (19.51%)
17 (85.0%)
3 (15.0%)
zero
one
two
three
ca

In [15]:
```
fig = plt.gcf()
fig.set_size_inches(25,35)
grid_rows = 3
grid_cols = 2
#draw Number of major vessels colored by fluoroscopy  vs disease outcome
plt.subplot(grid_rows, grid_cols, 4)
temp = data[['thal','target']].groupby(['thal','target']).size().unstack('target')
temp.rename(index={3:'Normal', 6:'fixed defect', 7:'reversible defect'}, columns={0:'No Disease', 1:'H
stacked_barchart(temp, title = 'CHD vs Defect type ', ylabel = 'Population')
```


CHD vs Defect type
No Disease
Has Disease
Population
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%
320 (46.92%)
362 (53.08%)
35 (76.09%)
11 (23.91%)
154 (80.21%)
38 (19.79%)
Normal
fixed defect
reversible defect
thal

In [16]:
```
plt.figure(figsize=(25,10))
sns.countplot(x='age',data = data, hue = 'target',palette='husl')
plt.show()
```



In [ ]:
```
```