```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno # To visualize missing value
import plotly.graph_objects as go # To Generate Graphs
import plotly.express as px # To Generate box plot for statistical representatio
%matplotlib inline
df = pd.read_csv('./heart.csv')
df.nunique()
```

```
age          41
sex           2
cp            4
trestbps     49
chol        152
fbs           2
restecg       3
thalach      91
exang         2
oldpeak      40
slope         3
ca            5
thal          4
target        2
dtype: int64
```

```
df.loc[df["ca"]==4,'ca'] = np.NaN
```

```
df.loc[df["thal"]==0,'thal'] = np.NaN
```

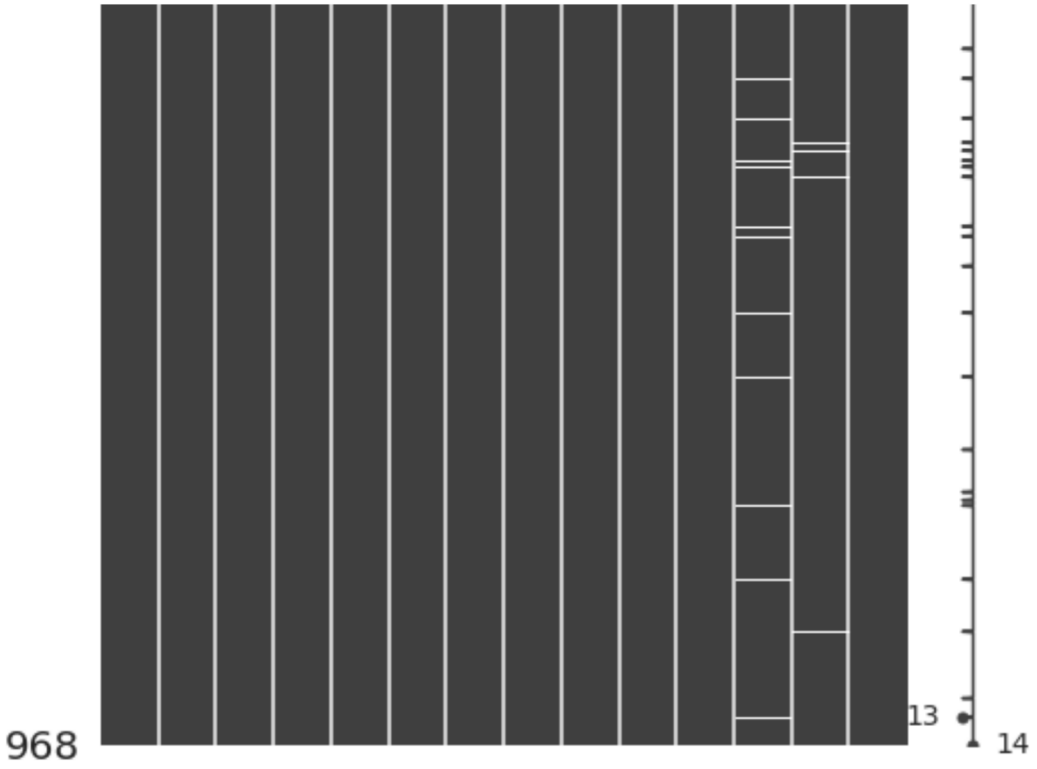
```
df.isnull().sum()
```

```
age          0
sex          0
cp           0
trestbps     0
chol         0
fbs          0
restecg      0
thalach      0
exang        0
oldpeak      0
slope        0
ca          18
thal         7
target       0
dtype: int64
```

```
msno.matrix(df, figsize=(8,8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f35735f4050>
```

✓ 0s completed at 2:37 PM

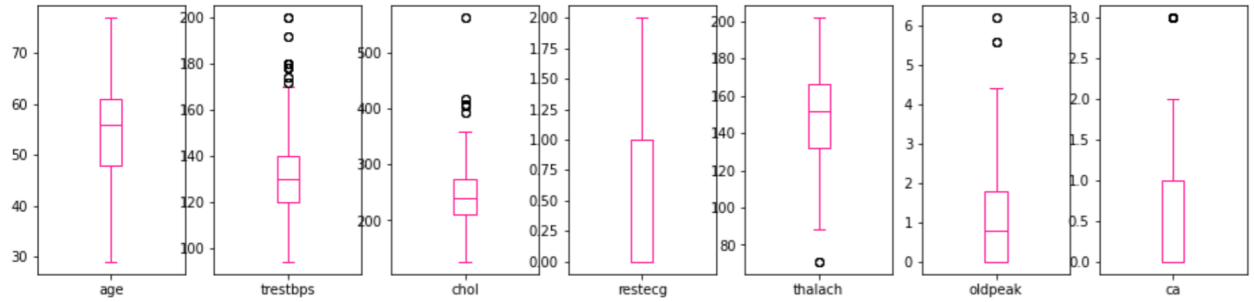



```
df.fillna(df.median())
df.isnull().sum()
```

```
age          0
sex          0
cp           0
trestbps     0
chol         0
fbs          0
restecg      0
thalach      0
exang        0
oldpeak      0
slope        0
ca          18
thal         7
target       0
dtype: int64
```

```
duplicate = df.duplicated().sum()
if duplicate:
  print(f"Duplicated items are {duplicate}")
```

```
Duplicated items are 723
```




```
# define continuous variable & plot
continous_features = ['age','trestbps','chol','thalach','oldpeak']
def outliers(df_out, drop = False):
    for each_feature in df_out.columns:
        feature_data = df_out[each_feature]
        Q1 = np.percentile(feature_data, 25.) # 25th percentile of the data of t
        Q3 = np.percentile(feature_data, 75.) # 75th percentile of the data of t
        IQR = Q3-Q1 #Interquartile Range
        outlier_step = IQR * 1.5 #That's we were talking about above
        outliers = feature_data[~((feature_data >= Q1 - outlier_step) & (feature
        if not drop:
            print('For the feature {}, No of Outliers is {}'.format(each_feature
        if drop:
            df.drop(outliers, inplace = True, errors = 'ignore')
            print('Outliers from {} feature removed'.format(each_feature))
outliers(df[continous_features])
```

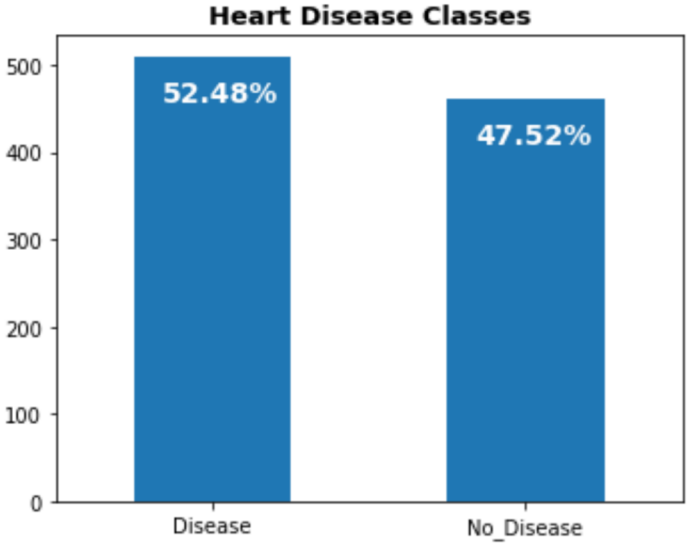
```
For the feature age, No of Outliers is 0
For the feature trestbps, No of Outliers is 30
For the feature chol, No of Outliers is 16
For the feature thalach, No of Outliers is 4
For the feature oldpeak, No of Outliers is 7
```

```
outliers(df[continous_features],drop=True)
```

```
Outliers from age feature removed
Outliers from trestbps feature removed
```

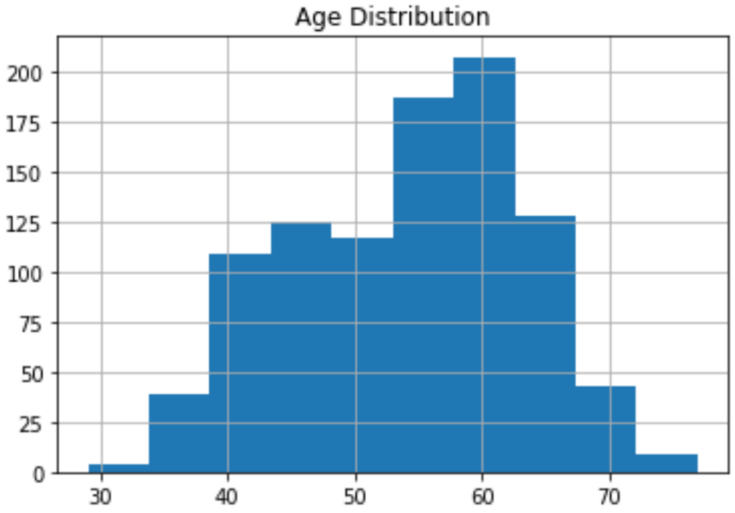
```
        str(round((i.get_height()/total)*100, 2))+'%', fontsize=14,
            color='white', weight = 'bold')

plt.tight_layout()
```

```
Disease       508
No_disease    460
Name: target, dtype: int64
```

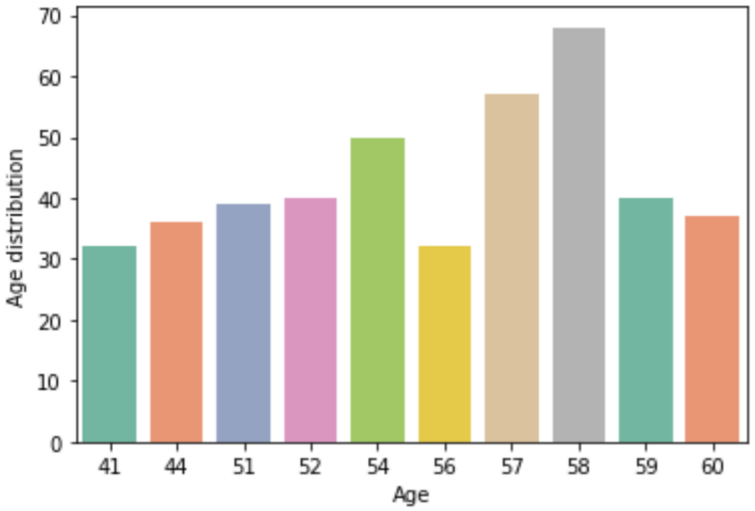



```
# print(df.age.value_counts())
df['age'].hist().plot(kind='bar')
plt.title("Age Distribution")
```

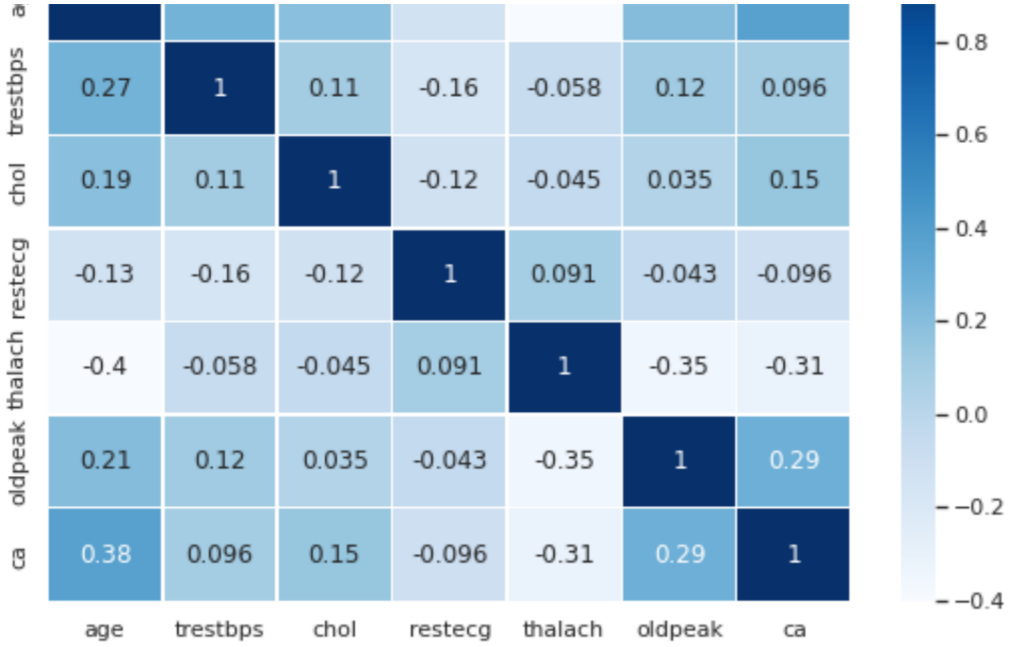
```
Text(0.5, 1.0, 'Age Distribution')
```



```
41    32
56    32
Name: age, dtype: int64
Text(0, 0.5, 'Age distribution')
```




```
fig, ax = plt.subplots(figsize=(8,4))
name = df['cp']
ax = sns.countplot(x='cp', hue='target', data=df, palette='Set2')
ax.set_title("Chest Pain Distribution according to Target", fontsize = 13, weigh
ax.set_xticklabels (name, rotation = 0)

totals = []
for i in ax.patches:
    totals.append(i.get_height())
total = sum(totals)
```


a
trestbps
chol
restecg
thalach
oldpeak
ca
0.27 1 0.11 -0.16 -0.058 0.12 0.096
0.19 0.11 1 -0.12 -0.045 0.035 0.15
-0.13 -0.16 -0.12 1 0.091 -0.043 -0.096
-0.4 -0.058 -0.045 0.091 1 -0.35 -0.31
0.21 0.12 0.035 -0.043 -0.35 1 0.29
0.38 0.096 0.15 -0.096 -0.31 0.29 1
age
trestbps
chol
restecg
thalach
oldpeak
ca
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4