Atividade1- Jose-Luiz-Vilas-Boas

November 3, 2021

[5]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statistics as sts
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[6]:

```
df = pd.read_csv('heart.csv')
```

[7]:

```
display(df)
```

```
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0      52    1   0       125   212    0        1      168      0      1.0
1      53    1   0       140   203    1        0      155      1      3.1
2      70    1   0       145   174    0        1      125      1      2.6
3      61    1   0       148   203    0        1      161      0      0.0
4      62    0   0       138   294    1        1      106      0      1.9
...   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
1020   59    1   1       140   221    0        1      164      1      0.0
1021   60    1   0       125   258    0        0      141      1      2.8
1022   47    1   0       110   275    0        0      118      1      1.0
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4

      slope  ca  thal  target
0         2   2     3       0
1         0   0     3       0
2         0   0     3       0
3         2   1     3       0
4         1   3     2       0
...     ...  ..   ...     ...
1020      2   0     2       1
1021      1   1     3       0
1022      1   1     2       0
1023      2   0     2       1
1024      1   1     3       0
```

```
[1025 rows x 14 columns]
```

[8]:

```
#verificar valores nulos
df.isnull().sum()
```

[8]:

```
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
```

[14]:

```
#verificar quais atributos são numéricos e valores nulos
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

[19]:

```
#Resumo
df.describe().T
```

[19]:

```
          count        mean        std    min    25%    50%    75%    max
age      1025.0   54.434146   9.072290   29.0   48.0   56.0   61.0   77.0
sex      1025.0    0.695610   0.460373    0.0    0.0    1.0    1.0    1.0
cp       1025.0    0.942439   1.029641    0.0    0.0    1.0    2.0    3.0
trestbps 1025.0  131.611707  17.516718   94.0  120.0  130.0  140.0  200.0
chol     1025.0  246.000000  51.592510  126.0  211.0  240.0  275.0  564.0
fbs      1025.0    0.149268   0.356527    0.0    0.0    0.0    0.0    1.0
restecg  1025.0    0.529756   0.527878    0.0    0.0    1.0    1.0    2.0
thalach  1025.0  149.114146  23.005724   71.0  132.0  152.0  166.0  202.0
exang    1025.0    0.336585   0.472772    0.0    0.0    0.0    1.0    1.0
oldpeak  1025.0    1.071512   1.175053    0.0    0.0    0.8    1.8    6.2
slope    1025.0    1.385366   0.617755    0.0    1.0    1.0    2.0    2.0
ca       1025.0    0.754146   1.030798    0.0    0.0    0.0    1.0    4.0
thal     1025.0    2.323902   0.620660    0.0    2.0    2.0    3.0    3.0
target   1025.0    0.513171   0.500070    0.0    0.0    1.0    1.0    1.0
```

0.0.1 Verificando a variável fbs

[16]:

```
group = df.groupby(['fbs']).size()
group
```

[16]:

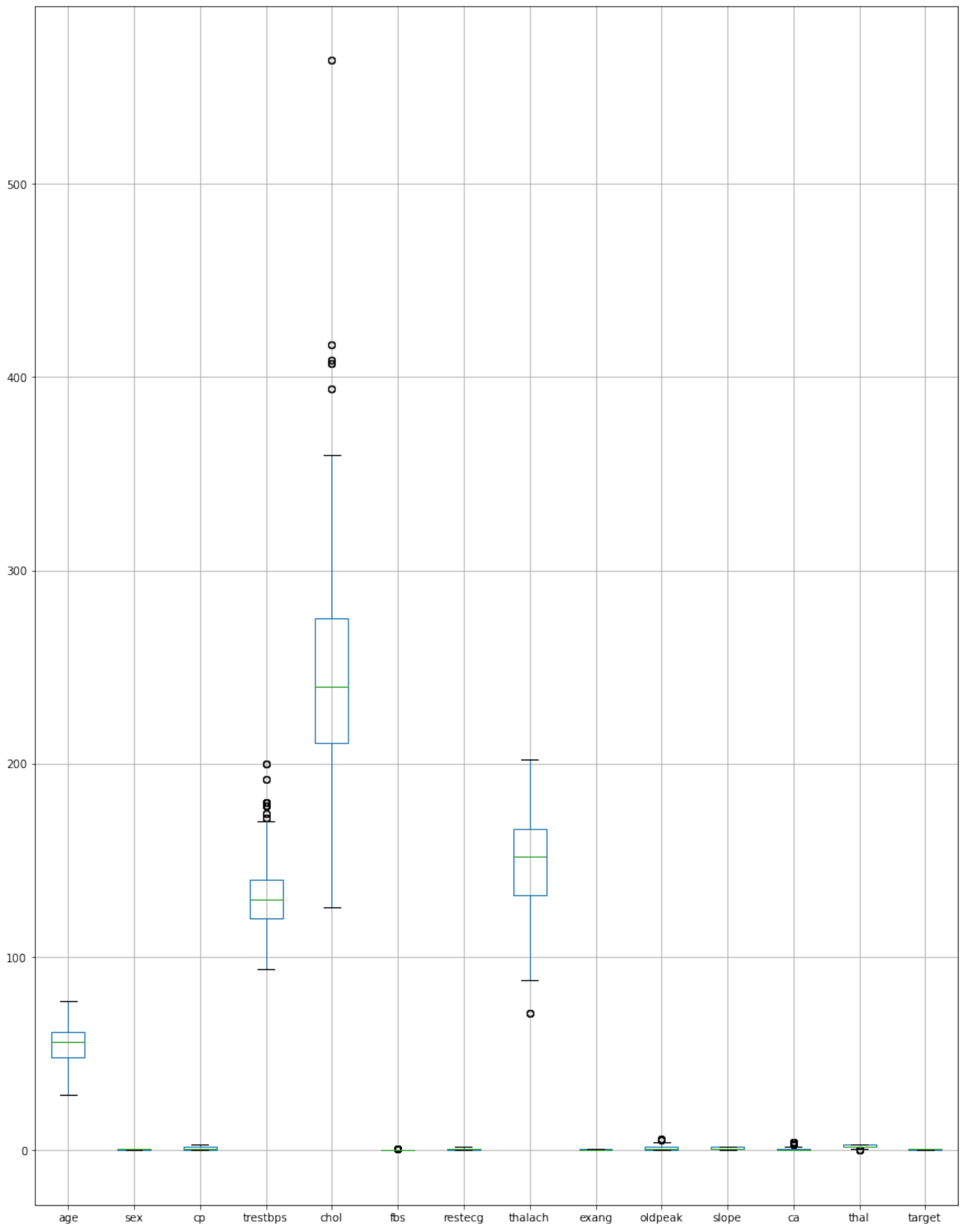
```
fbs
0    872
1    153
dtype: int64
```

Analisando a variável FastingBS, embora seja numérica possui caracterśiticas categórias, portanto, não será trabalhada.

0.0.2 Apresentar a média, moda,variância e o desvio padrão de todos os atributos contínuos

[20]:

```
#Gerando o boxplot com p pandas de todos os campos
boxplot = df.boxplot(grid = 'False',figsize = (15,20))
```



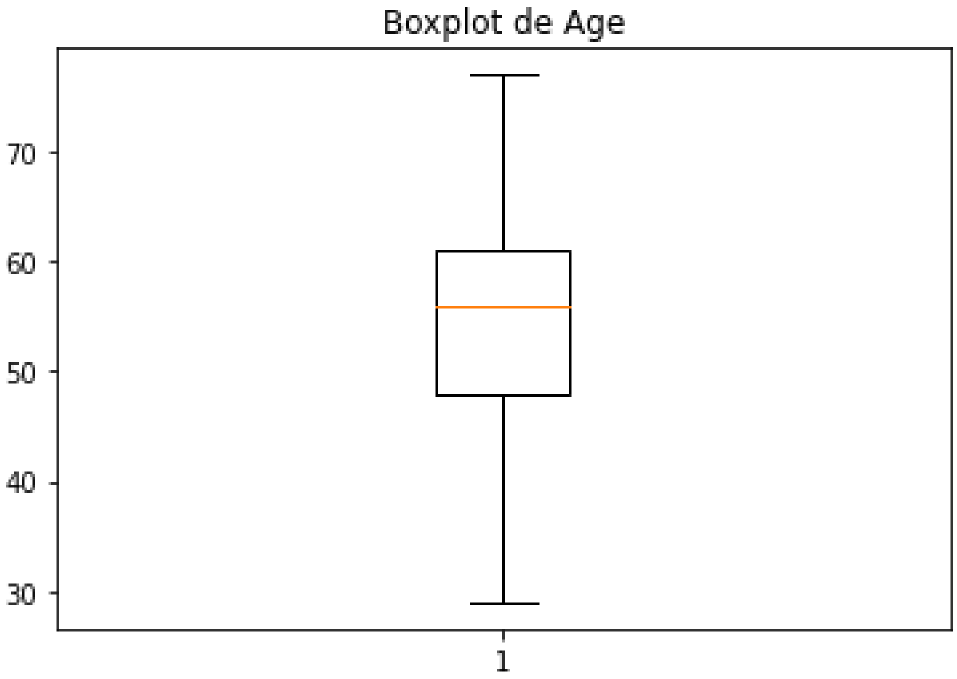
age

```
[5]: print('Média = ',round(df['age'].mean(),2))
     print('Moda = ',sts.mode(df['age']))
     print('Mediana = ',round(df['age'].median(),2))
     print('Variância = ',round(df['age'].var(),2))
     print('Desvio Padrão = ',round(df['age'].std(),2))
```

```
Média =  54.43
Moda =  58
Mediana =  56.0
Variância =  82.31
Desvio Padrão =  9.07
```

[6]:

```
#Gerando o boxplot
diamante = dict(markerfacecolor='r', marker='D')
fig1, ax1 = plt.subplots()
ax1.set_title('Boxplot de Age')
ax1.boxplot(df['age'],flierprops=diamante)
plt.show()
```



trestbps

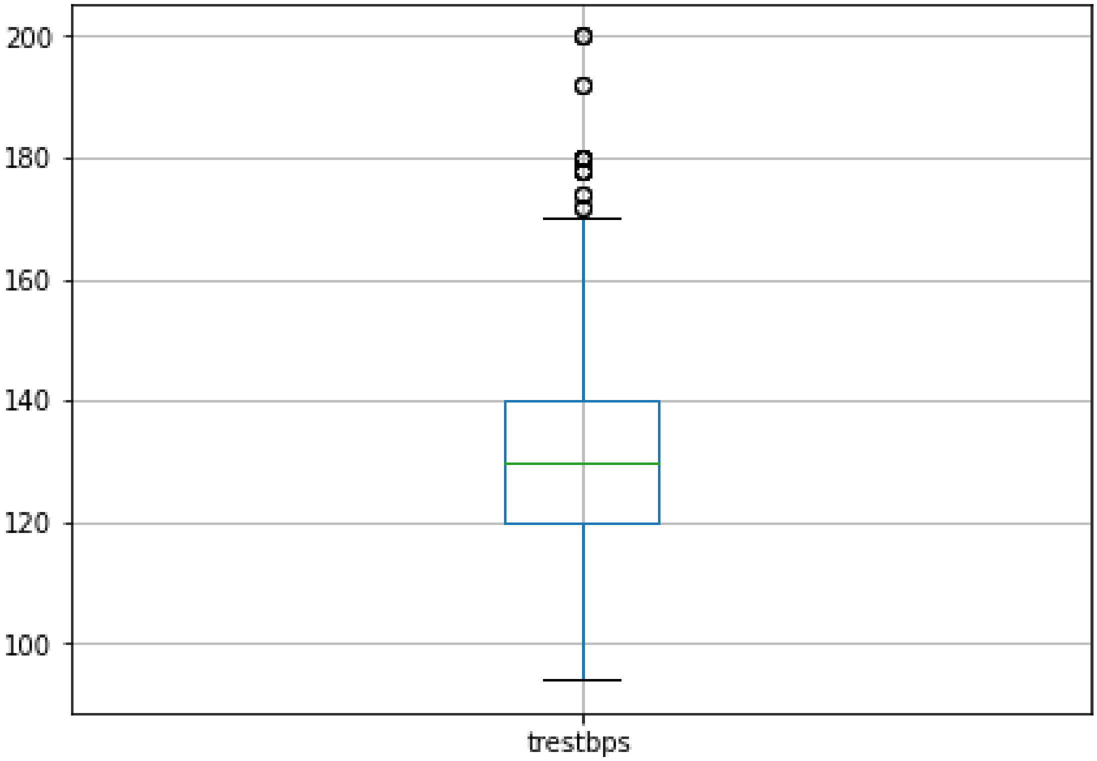
[7]:

```
print('Média = ',round(df['trestbps'].mean(),2))
print('Moda = ',sts.mode(df['trestbps']))
print('Mediana = ',round(df['trestbps'].median(),2))
print('Variância = ',round(df['trestbps'].var(),2))
print('Desvio Padrão = ',round(df['trestbps'].std(),2))
```

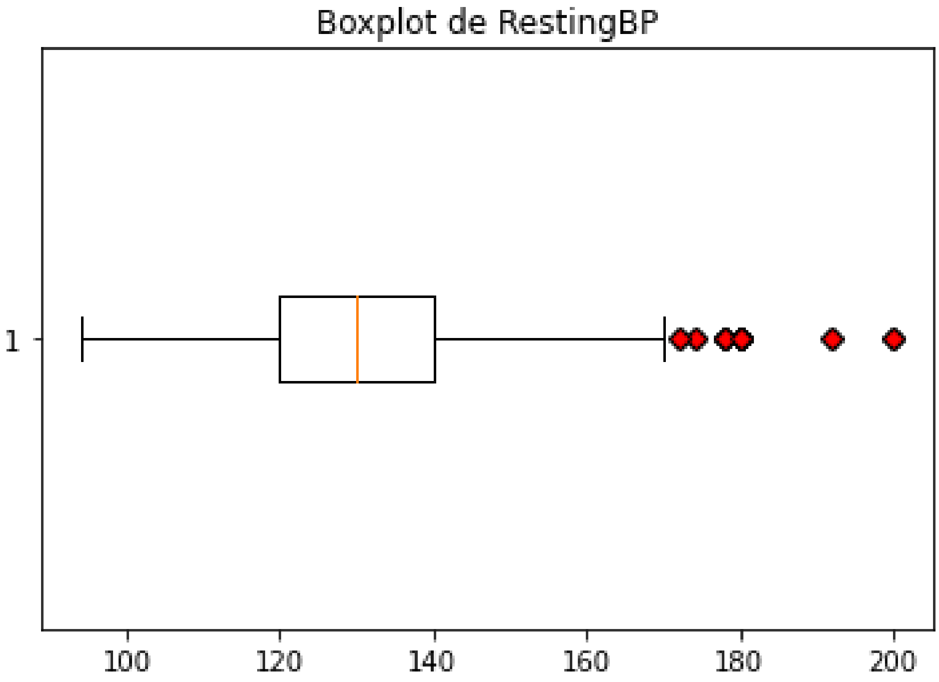
```
Média =  131.61
Moda =  120
```

```
Mediana =  130.0
Variância =  306.84
Desvio Padrão =  17.52
```

[27]:
```
#Blox plot de trestbps
boxplot = df.boxplot(column = 'trestbps',figsize = (7,5))
```



[8]:
```
#Gerando o boxplot
fig2, ax2 = plt.subplots()
ax2.set_title('Boxplot de RestingBP')
ax2.boxplot(df['trestbps'],vert=False,flierprops=diamante)
plt.show()
```



chol

[9]:

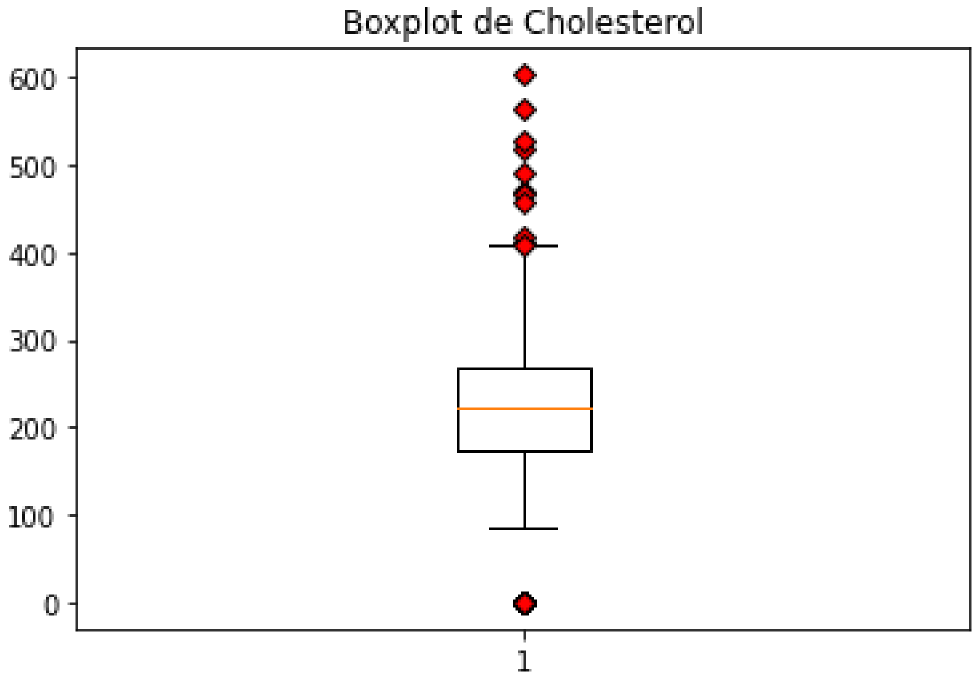
```
print('Média = ',round(df['chol'].mean(),2))
print('Moda = ',sts.mode(df['chol']))
print('Mediana = ',round(df['chol'].median(),2))
print('Variância = ',round(df['chol'].var(),2))
print('Desvio Padrão = ',round(df['chol'].std(),2))
```

```
Média =  246.0
Moda =  204
Mediana =  240.0
Variância =  2661.79
Desvio Padrão =  51.59
```

[10]:

```
#fig3, ax3 = plt.subplots()
#ax3.set_title('Boxplot de Cholesterol')
#ax3.boxplot(df['Cholesterol'], flierprops= diamante)
#plt.show()

fig3,ax3 = plt.subplots()
ax3.set_title('Boxplot de Cholesterol')
ax3.boxplot(df['Cholesterol'], flierprops = diamante)
plt.show()
```

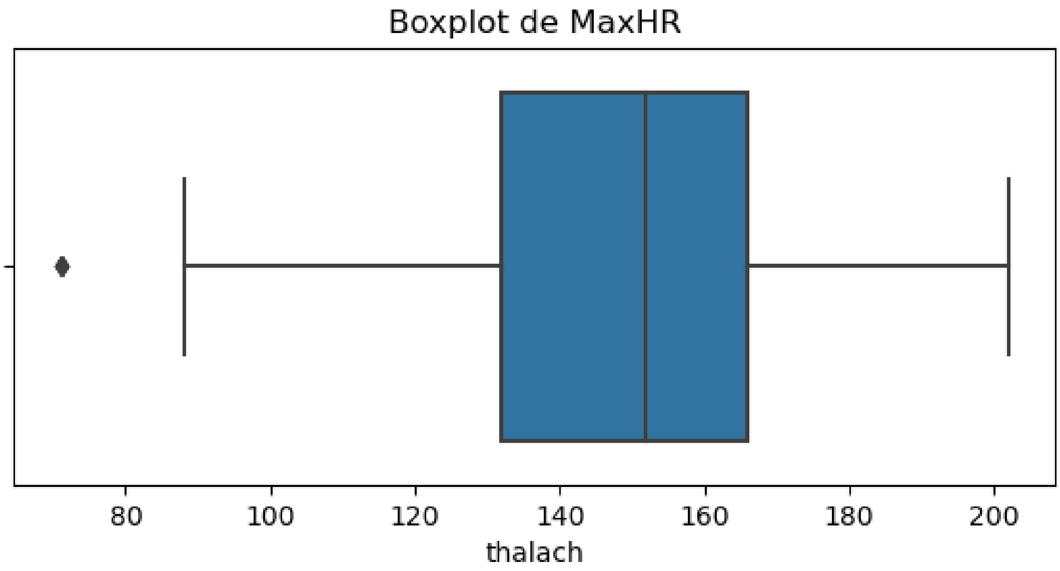


thalach

```
[12]: print('Média = ',round(df['thalach'].mean(),2))
      print('Moda = ',sts.mode(df['thalach']))
      print('Mediana = ',round(df['thalach'].median(),2))
      print('Variância = ',round(df['thalach'].var(),2))
      print('Desvio Padrão = ',round(df['thalach'].std(),2))
```

```
Média =  149.11
Moda =  162
Mediana =  152.0
Variância =  529.26
Desvio Padrão =  23.01
```

```
[13]: plt.figure(figsize=(7,3),dpi=100)
      ax = sns.boxplot(df['thalach']).set_title('Boxplot de MaxHR')
```

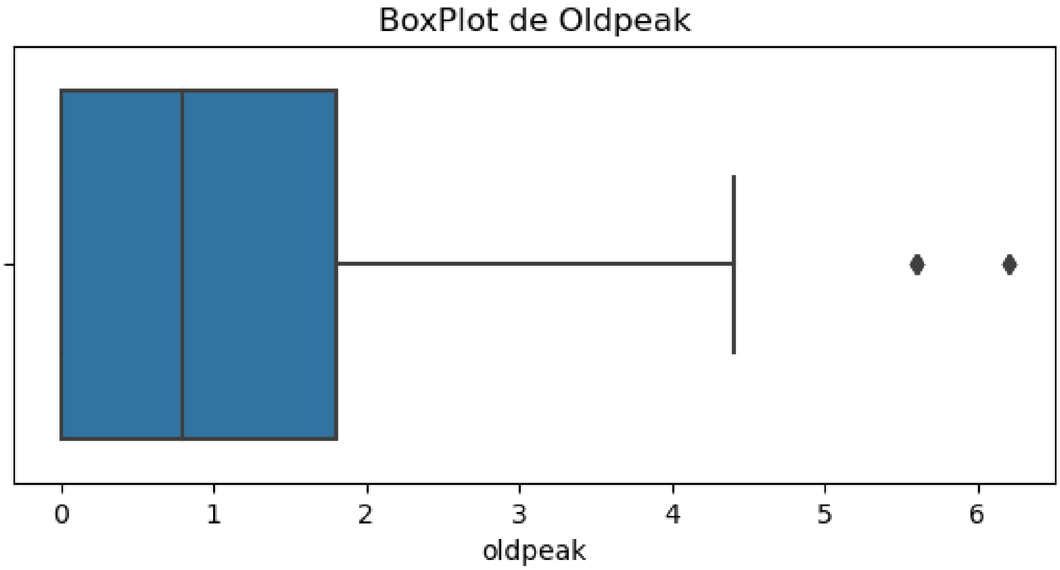


oldpeak

```
[19]: print('Média = ',round(df['oldpeak'].mean(),2))
      print('Moda = ',df['oldpeak'].mode())
      print('Mediana = ',round(df['oldpeak'].median(),2))
      print('Variância = ',round(df['oldpeak'].var(),2))
      print('Desvio Padrão = ',round(df['oldpeak'].std(),2))
```

```
Média =  1.07
Moda =  0    0.0
dtype: float64
Mediana =  0.8
Variância =  1.38
Desvio Padrão =  1.18
```

```
[20]: plt.figure(figsize = (7,3),dpi=100)
      ax = sns.boxplot(df['oldpeak']).set_title('BoxPlot de Oldpeak')
```

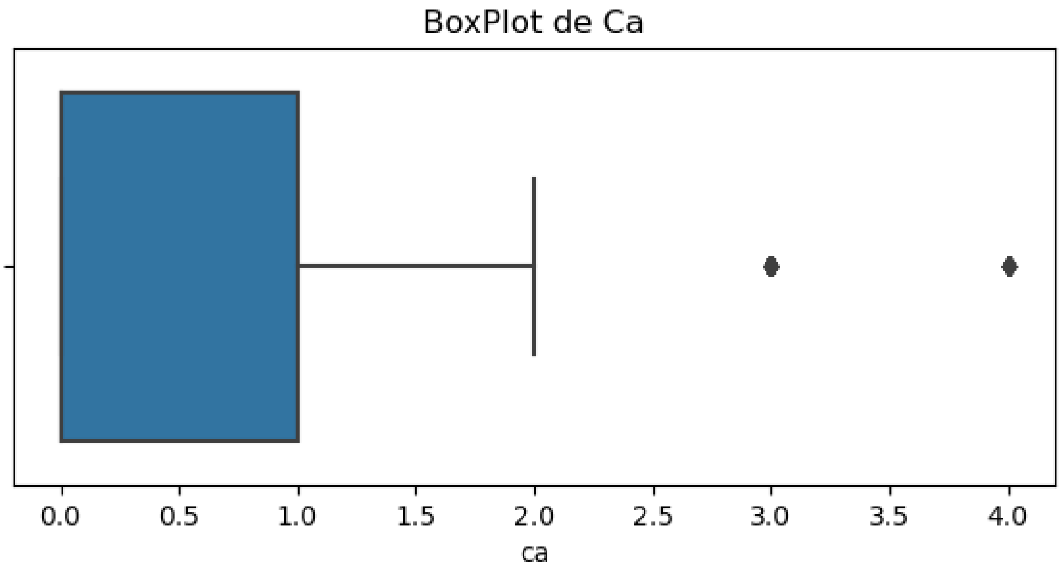


ca

[27]:
```
print('Média = ',round(df['ca'].mean(),2))
print('Moda = ',df['ca'].mode())
print('Mediana = ',round(df['ca'].median(),2))
print('Variância = ',round(df['ca'].var(),2))
print('Desvio Padrão = ',round(df['ca'].std(),2))
```

```
Média =  0.75
Moda =  0    0
dtype: int64
Mediana =  0.0
Variância =  1.06
Desvio Padrão =  1.03
```

[28]:
```
plt.figure(figsize = (7,3),dpi=100)
ax = sns.boxplot(df['ca']).set_title('BoxPlot de Ca')
```

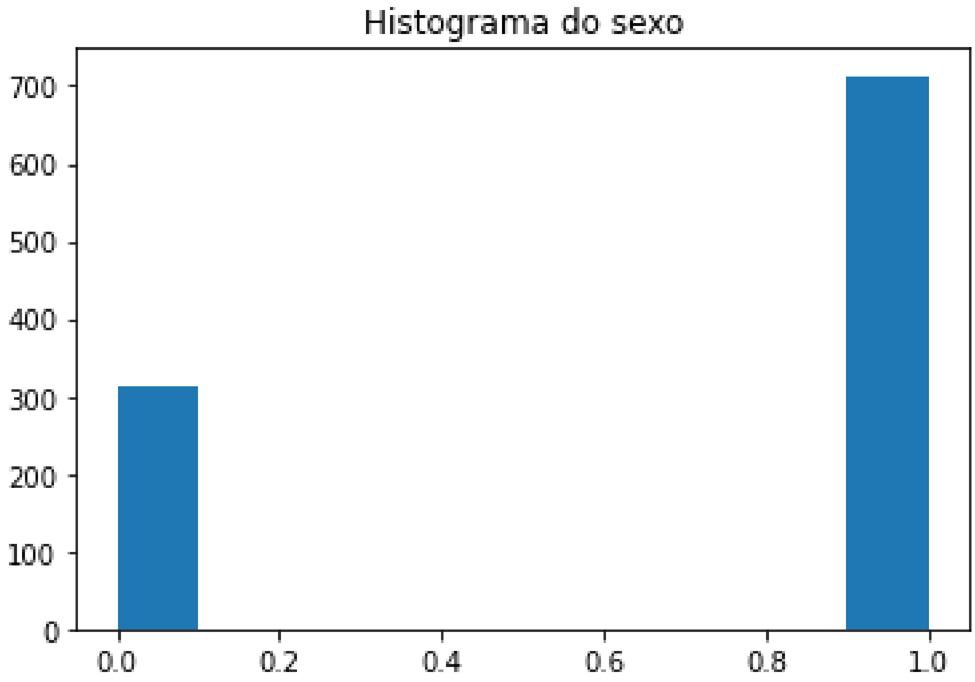


0.0.3 Criandos histogramas para os atributos discretos

sex

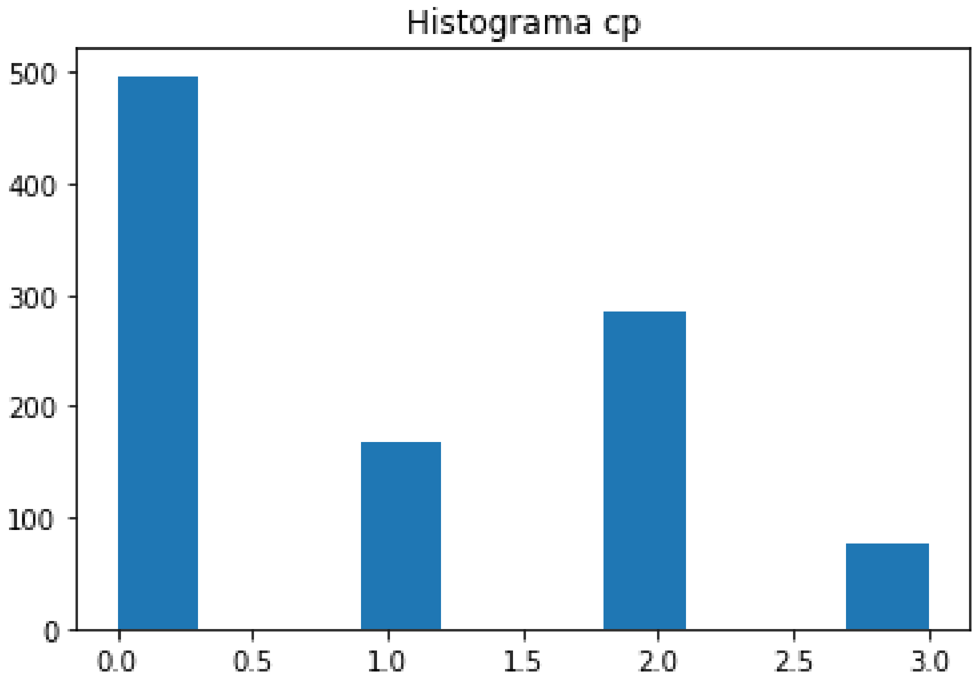
[37]:

```
plt.hist(df['sex'], histtype = 'stepfilled', rwidth = 0.8)
plt.title("Histograma do sexo")
plt.show()
```



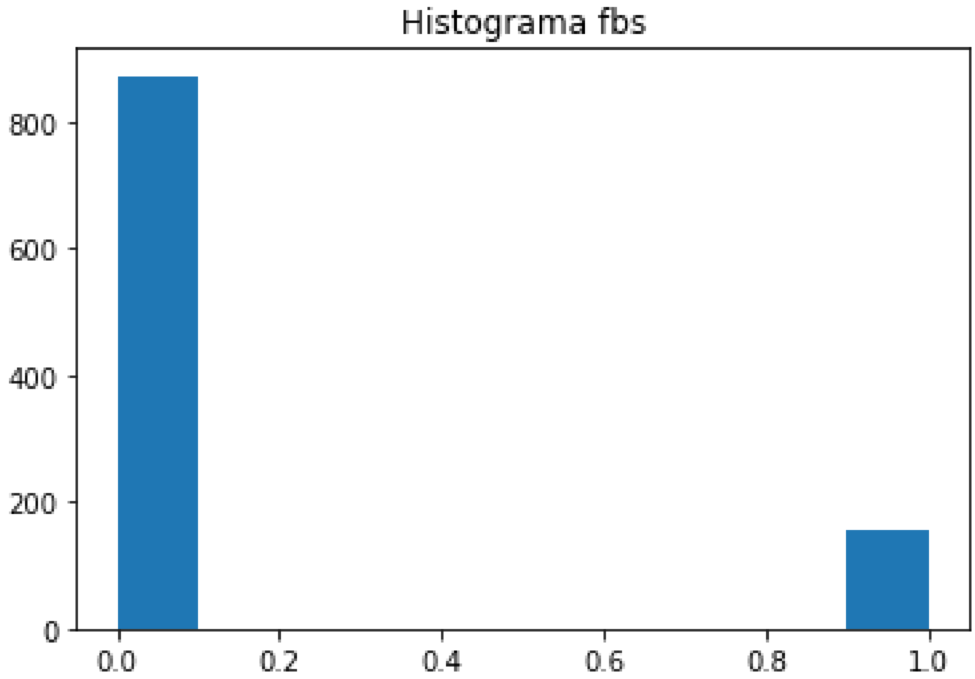
cp

[38]:
```
plt.hist(df['cp'], histtype = 'stepfilled', rwidth = 0.8)
plt.title("Histograma cp")
plt.show()
```



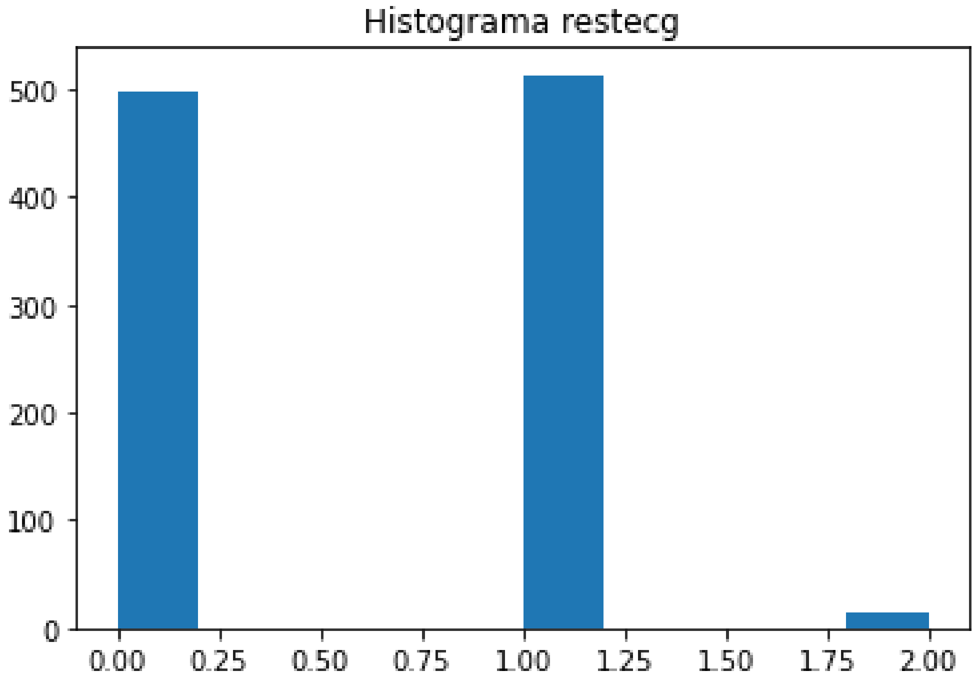
fbs

[39]:
```
plt.hist(df['fbs'], histtype = 'stepfilled', rwidth = 0.8)
plt.title("Histograma fbs")
plt.show()
```



restecg

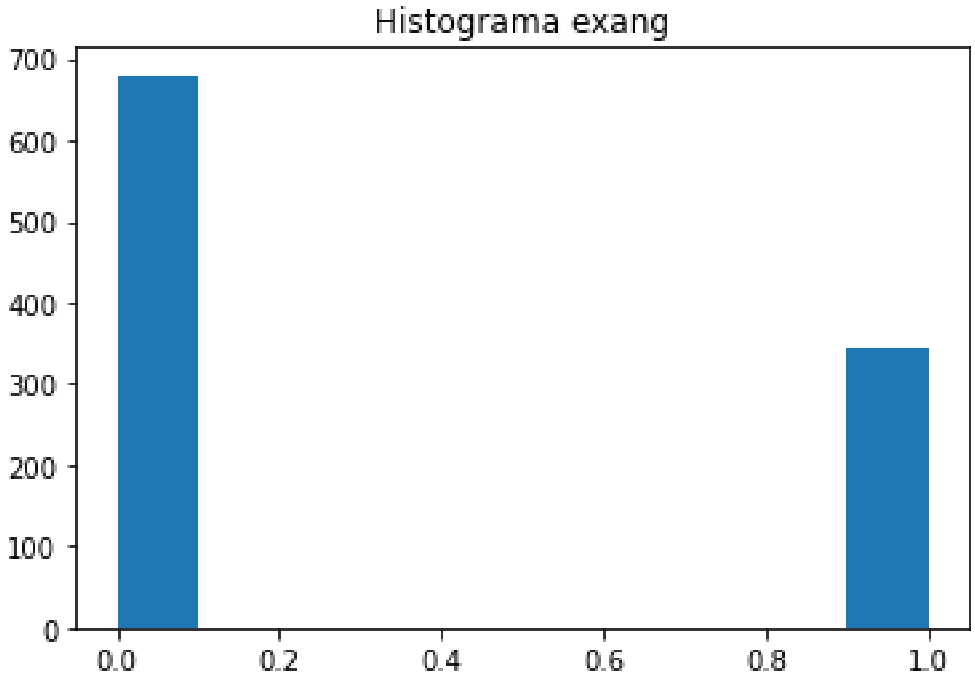
```
[40]: plt.hist(df['restecg'], histtype = 'stepfilled', rwidth = 0.8)
      plt.title("Histograma restecg")
      plt.show()
```



exang

[41]:

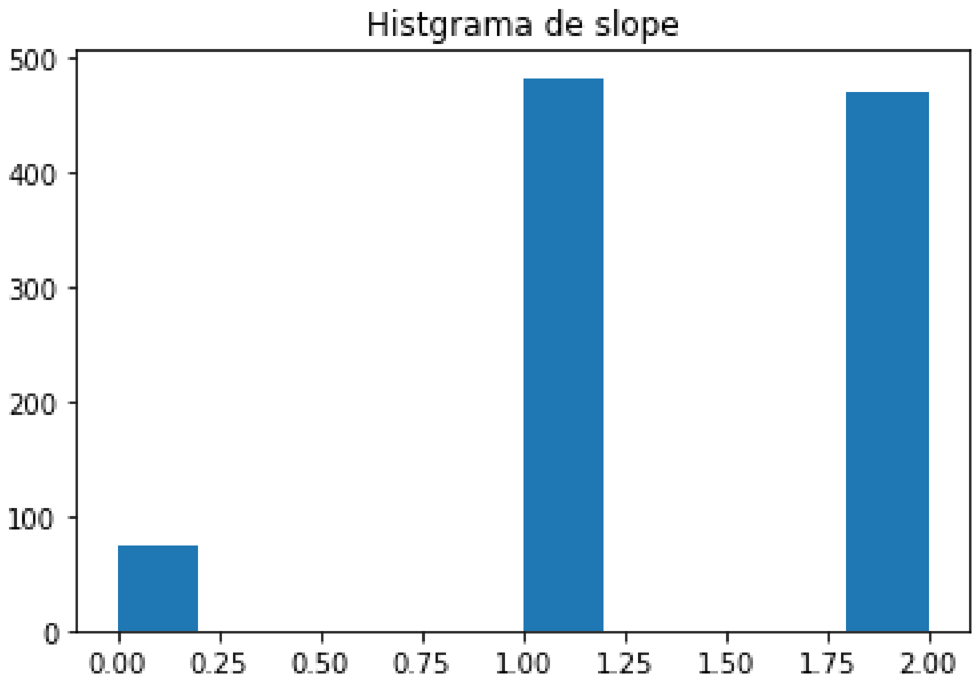
```
plt.hist(df['exang'], histtype = 'stepfilled', rwidth = 0.8)
plt.title("Histograma exang")
plt.show()
```



slope

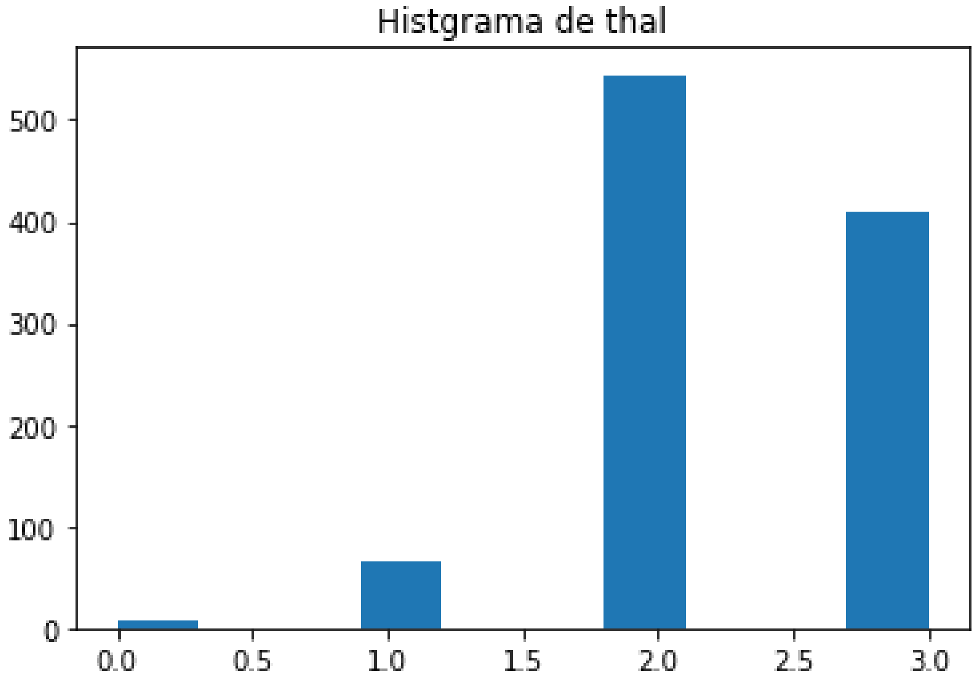
[31]:

```
plt.hist(df['slope'], histtype = 'stepfilled', rwidth = 0.8)
plt.title("Histgrama de slope")
plt.show()
```



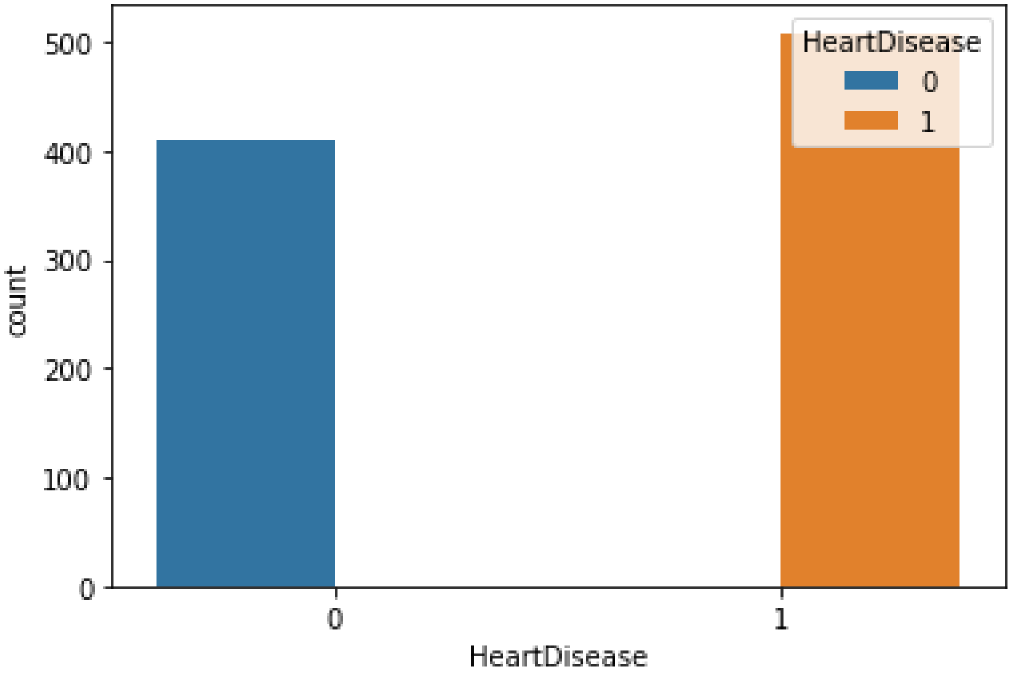
thal

```
[42]: plt.hist(df['thal'], histtype = 'stepfilled', rwidth = 0.8)
      plt.title("Histgrama de thal")
      plt.show()
```



HeartDisease

```
[80]: ax = sns.countplot(x = df['HeartDisease'], hue = df['HeartDisease'],  data =␣
      ↪df['HeartDisease'])
```



```
[9]: #USar o dataprep.
     from dataprep.eda import create_report
     from dataprep.eda import plot
```

```
[10]: create_report(df).show()
```

```
  0%|          | 0/1737 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

```
[ ]:
```