Heart Disease Data Analysis Paper

By  
 Ritaprava Dutta

Data Source: Kaggle  
Data Format: Comma Separated Value  
Technology Used: Python   
IDE Used: Spyder   
Special Thanks to NumPy, Pandas, Matplotlib and Seaborn Communities

* Python has a wide range of development community. And thankfully so it provides very useful packages. First job is to import all the necessary packages.

**In[0]:**

# import the packages   
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns

* Now let’s import the data and get some overview of it.

**In[1]:**

# import the data  
data = pd.read\_csv('heart.csv')  
print(data.info())  
print(data.head())

**Out[1]:**

RangeIndex: 303 entries, 0 to 302

Data columns (total 14 columns):

age 303 non-null int64

sex 303 non-null int64

cp 303 non-null int64

trestbps 303 non-null int64

chol 303 non-null int64

fbs 303 non-null int64

restecg 303 non-null int64

thalach 303 non-null int64

exang 303 non-null int64

oldpeak 303 non-null float64

slope 303 non-null int64

ca 303 non-null int64

thal 303 non-null int64

target 303 non-null int64

dtypes: float64(1), int64(13)

memory usage: 33.2 KB

age sex cp trestbps chol fbs ... exang oldpeak slope ca thal target

0 63 1 3 145 233 1 ... 0 2.3 0 0 1 1

1 37 1 2 130 250 0 ... 0 3.5 0 0 2 1

2 41 0 1 130 204 0 ... 0 1.4 2 0 2 1

3 56 1 1 120 236 0 ... 0 0.8 2 0 2 1

4 57 0 0 120 354 0 ... 1 0.6 2 0 2 1

[5 rows x 14 columns]  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Examining the age column. It represents age of every person in the dataset.

**In[2]:**

#Analyze The age data  
age\_data = data['age']  
print("Total Age Data| Count:"+str(len(age\_data)))  
print("Minimum age:"+str(min(age\_data)))  
print("Maximum age:"+str(max(age\_data)))  
print("Average age:"+str(np.mean(age\_data)))  
print("Median age:"+str(np.median(age\_data)))

sorted\_age = np.sort(age\_data)  
plt.figure(0)  
plt.hist(age\_data, bins = 8)  
plt.title('Total Age Data')  
plt.xlabel('Age')  
plt.ylabel('No. of Person')  
plt.show()

true\_age\_data = data[data.target==1]['age']  
print("True Age Data| Count:"+str(len(true\_age\_data)))  
print("Minimum age:"+str(min(true\_age\_data)))  
print("Maximum age:"+str(max(true\_age\_data)))  
print("Average age:"+str(np.mean(true\_age\_data)))  
print("Median age:"+str(np.median(true\_age\_data)))

sorted\_age = np.sort(true\_age\_data)  
plt.figure(1)  
plt.hist(true\_age\_data, bins = 8)  
plt.title('True Age Data')  
plt.xlabel('Age')  
plt.ylabel('No. of Person')  
plt.show()

**Out[2]:**

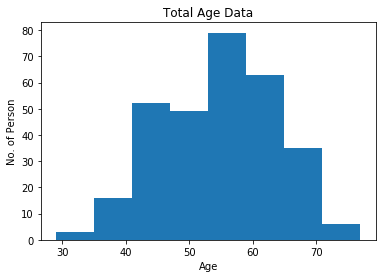
Total Age Data| Count:303

Minimum age:29

Maximum age:77

Average age:54.366336633663366

Median age:55.0



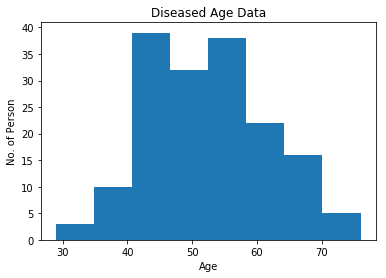
True Age Data| Count:165

Minimum age:29

Maximum age:76

Average age:52.4969696969697

Median age:52.0



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Next comes sex column. This one is binary data where 1 represents Male and 0 represents Female

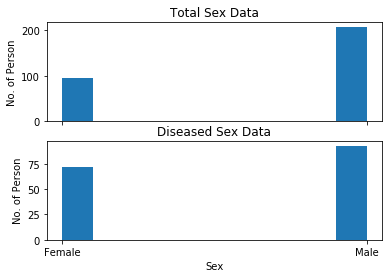
**In[3]:**

#Analyze sex data  
sex\_data = data['sex']  
print("Total Sex Data and Diseased Sex Data")  
plt.figure(2)  
plt.subplot(2,1,1)  
plt.hist(sex\_data)  
plt.title('Total Sex Data')  
plt.ylabel('No. of Person')  
plt.xticks([0,1],['', ''])

true\_sex\_data = data[data.target==1]['sex']  
plt.subplot(2,1,2)  
plt.hist(true\_sex\_data)  
plt.title('Diseased Sex Data')  
plt.xlabel('Sex')  
plt.ylabel('No. of Person')  
plt.xticks([0,1],['Female', 'Male'])  
plt.show()

Out[3]:

Total Sex Data and Diseased Sex Data



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Let’s move on to cp column which signifies chest pain. This is a categorical data that ranges between 0 to 3.

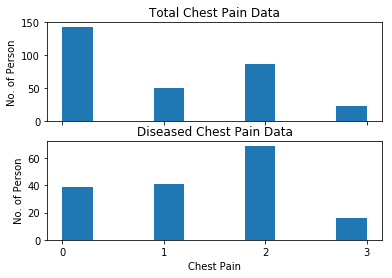
**In[3]:**

#Analyze cp data   
cp\_data = data['cp']  
print("Total Chest Pain Data and Diseased Chest Data")  
plt.figure(3)  
plt.subplot(2,1,1)  
plt.hist(cp\_data)  
plt.title('Total Chest Pain Data')  
plt.ylabel('No. of Person')  
plt.xticks([0,1, 2, 3],['', '', '', ''])

true\_cp\_data = data[data.target==1]['cp']  
plt.subplot(2,1,2)  
plt.hist(true\_cp\_data)  
plt.title('Diseased Chest Pain Data')  
plt.xlabel('Chest Pain')  
plt.ylabel('No. of Person')  
plt.xticks([0, 1, 2, 3])

plt.show()

**Out[3]:**

Total Chest Pain Data and Diseased Chest Data  


* Resting blood pressure given as trestbp in the data is next to be observed.
* As previous data we will make a histogram first. Then also make a ECDF( Empirical Cumulative Distribution Function) for the same do get some more visual EDA. For that purpose let’s define a function named ecdf.

**In[4]:**

def ecdf(data):

"""Compute ECDF for a one-dimensional array of measurements."""  
# Number of data points: n  
n = len(data)

# x-data for the ECDF: x  
x = np.sort(data)

# y-data for the ECDF: y  
y = np.arange(1, n+1) / n  
  
return x, y

**In[5]:**

#Analyze cp data

#Plotting a histogram  
rbp\_data = data['trestbps']  
print("Total Resting Blood Pressure Data")  
print("Minimum RBP:"+str(min(rbp\_data)))  
print("Maximum RBP:"+str(max(rbp\_data)))  
print("Average RBP:"+str(np.mean(rbp\_data)))  
print("Median RBP:"+str(np.median(rbp\_data)))

plt.figure(4)  
plt.hist(bp\_data, bins=8)  
plt.title('Total Resting Blood Pressure Data')  
plt.ylabel('No. of Person')  
plt.xlabel('Blood Pressure')  
plt.show()

true\_rbp\_data = data[data.target==1]['trestbps']  
print("Diseased Resting Blood Pressure Data")  
print("Minimum RBP:"+str(min(true\_rbp\_data)))  
print("Maximum RBP:"+str(max(true\_rbp\_data)))  
print("Average RBP:"+str(np.mean(true\_rbp\_data)))  
print("Median RBP:"+str(np.median(true\_rbp\_data)))

plt.figure(5)  
plt.hist(true\_rbp\_data, bins=8)  
plt.title('Diseased Resting Blood Pressure Data')  
plt.xlabel('Blood Pressure')  
plt.ylabel('No. of Person')  
plt.show()

**Out[5]:**

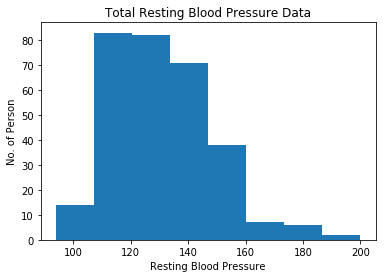
Total Resting Blood Pressure Data

Minimum RBP:94

Maximum RBP:200

Average RBP:131.62376237623764

Median RBP:130.0



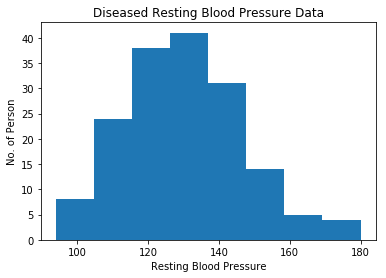
Diseased Resting Blood Pressure Data

Minimum RBP:94

Maximum RBP:180

Average RBP:129.3030303030303

Median RBP:130.0



* The list is followed by cholesterol.

**In[6]:**

#Cholesterol analysis

chol\_data = data['chol']  
print("Total Cholesterol Data")  
print("Minimum Cholesterol:"+str(min(chol\_data)))print("Maximum Cholesterol:"+str(max(chol\_data)))  
print("Average Cholesterol:"+str(np.mean(chol\_data)))  
print("Median Cholesterol:"+str(np.median(chol\_data)))plt.figure(7)  
plt.hist(rbp\_data, bins=8)  
plt.title('Total Cholesterol Data')  
plt.ylabel('No. of Person')  
plt.xlabel('Cholesterol')  
plt.show()

true\_chol\_data = data[data.target==1]['chol']  
print("Diseased Cholesterol Data")  
print("Minimum Cholesterol:"+str(min(true\_chol\_data)))  
print("Maximum Cholesterol:"+str(max(true\_chol\_data)))  
print("Average Cholesterol:"+str(np.mean(true\_chol\_data)))  
print("Median Cholesterol:"+str(np.median(true\_chol\_data)))  
plt.figure(8)  
plt.hist(true\_rbp\_data, bins=8)  
plt.title('Diseased Cholesterol Data')  
plt.xlabel('Cholesterol')  
plt.ylabel('No. of Person')  
plt.show()

false\_chol\_data = data[data.target==0]['chol']  
chol\_com\_data = [true\_chol\_data, false\_chol\_data]  
plt.figure(8)  
plt.boxplot(chol\_com\_data)  
plt.xticks([1,2], [True, False])  
plt.ylabel("Cholesterol")  
plt.show()

**Out[6]:**

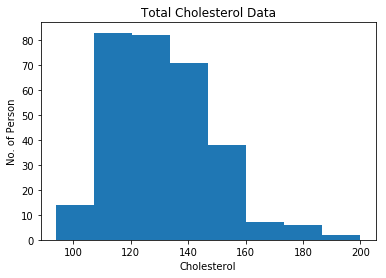
Total Cholesterol Data

Minimum Cholesterol:126

Maximum Cholesterol:564

Average Cholesterol:246.26402640264027

Median Cholesterol:240.0



Diseased Cholesterol Data

Minimum Cholesterol:126

Maximum Cholesterol:564

Average Cholesterol:242.23030303030302

Median Cholesterol:234.0

