APPENDIX

Project on Numpy:

numpy

```
import numpy as np
import pandas as pd
import matplotlib as mlt
import matplotlib.pyplot as plt

#np.loadtxt is another method
hd = np.genfromtxt('E:\DATA ASSOCIATE\POHD PROJECT\data_file\S1Data
(2).csv',delimiter= ',',skip_header=True,dtype = int)
print(hd)
```

```
[[    97      0      0 ...      1 237000    358]
 [   180      0      1 ...      1 160000    231]
 [    31      1      1 ...      1 263358    582]
 ...
 [    14      1      1 ...      0 166000    582]
 [    80      0      1 ...      1 297000    897]
 [    16      0      0 ...      1 276000     52]]
```

```
hd
```

```
array([[    97,      0,      0, ...,      1, 237000,    358],
       [   180,      0,      1, ...,      1, 160000,    231],
       [    31,      1,      1, ...,      1, 263358,    582],
       ...,
       [    14,      1,      1, ...,      0, 166000,    582],
       [    80,      0,      1, ...,      1, 297000,    897],
       [    16,      0,      0, ...,      1, 276000,     52]])
```

```
print(hd.shape)
print(hd.ndim)
```

```
(299, 13)
2
```

```
#accessing the rows
print(hd[0])
```

```
[    97      0      0      0      0      0      1     43     50    135
      1 237000    358]
```

```
print(hd[0::2])
```

```
[[    97      0      0 ...      1 237000    358]
 [    31      1      1 ...      1 263358    582]
 [   113      0      1 ...      1 242000   1610]
 ...
 [   250      0      0 ...      1 543000    582]
```

```
 [    14      1      1 ...      0 166000    582]
 [    16      0      0 ...      1 276000     52]]
```

```
#accessing the columns
hd[:,1:3]
```

```
array([[0, 0],
       [0, 1],
       [1, 1],
       [0, 1],
       [0, 1],
       [1, 1],
       [0, 1],
       [1, 1],
       [0, 1],
       [0, 1],
       [0, 1],
       [0, 1],
       [1, 1],
       [1, 1],
       [0, 1],
       [0, 1],
       [1, 1],
       [1, 1],
       [0, 1],
       [0, 1],
       [0, 1],
       [0, 1],
       [1, 1],
       [0, 0],
       [0, 1],
       [1, 1],
       [0, 1],
       [0, 0],
       [1, 1],
       [0, 1],
       [1, 1],
       [0, 1],
       [0, 1],
       [0, 0],
       [0, 0],
       [0, 1],
       [0, 1],
       [0, 1],
       [0, 1],
       [1, 0],
       [0, 0],
       [1, 1],
       [1, 0],
       [0, 1],
```

```
[1, 1],
[1, 1],
[1, 0],
[1, 1],
[1, 0],
[1, 1],
[1, 0],
[1, 0],
[1, 0],
[0, 0],
[0, 1],
[0, 0],
[0, 0],
[0, 0],
[0, 1],
[0, 0],
[1, 1],
[0, 0],
[0, 1],
[1, 1],
[0, 0],
[1, 1],
[0, 1],
[1, 1],
[0, 1],
[0, 1],
[1, 0],
[0, 1],
[1, 1],
[0, 0],
[1, 0],
[0, 1],
[0, 1],
[1, 1],
[1, 1],
[0, 1],
[0, 1],
[1, 0],
[1, 0],
[1, 1],
[1, 1],
[1, 0],
[0, 0],
[1, 0],
[0, 1],
[0, 0],
[0, 0],
[0, 1],
[0, 0],
```

```
[0, 0],
[1, 1],
[0, 0],
[0, 0],
[1, 0],
[1, 0],
[0, 1],
[0, 1],
[1, 0],
[1, 1],
[0, 1],
[0, 1],
[1, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[1, 1],
[0, 1],
[1, 1],
[0, 1],
[1, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[1, 1],
[0, 1],
[1, 1],
[1, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
```

```
[0, 1],
[0, 1],
[0, 1],
[1, 1],
[1, 1],
[1, 1],
[0, 1],
[0, 1],
[0, 1],
[1, 1],
[0, 1],
[0, 1],
[0, 1],
[1, 1],
[0, 1],
[1, 1],
[1, 1],
[1, 1],
[1, 1],
[0, 1],
[0, 1],
[1, 1],
[0, 0],
[0, 1],
[0, 1],
[1, 1],
[0, 1],
[0, 1],
[0, 0],
[0, 0],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[1, 0],
[1, 1],
[1, 1],
[0, 1],
[0, 1],
[0, 1],
[1, 1],
[0, 1],
[1, 1],
[0, 1],
[1, 1],
[1, 1],
[1, 1],
[0, 1],
[0, 1],
```

```
[0, 0],
[1, 0],
[1, 1],
[0, 0],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 1],
[0, 0],
[0, 0],
[0, 1],
[0, 0],
[1, 1],
[0, 0],
[0, 1],
[0, 0],
[0, 0],
[0, 0],
[0, 0],
[1, 0],
[0, 0],
[0, 0],
[0, 0],
[1, 0],
[1, 1],
[0, 0],
[0, 0],
[0, 1],
[1, 1],
[1, 0],
[1, 0],
[1, 0],
[1, 0],
[0, 0],
[1, 0],
[0, 1],
[1, 0],
[0, 0],
[0, 1],
[1, 0],
[0, 0],
[0, 1],
```

```
[0, 1],
[0, 0],
[0, 1],
[0, 1],
[1, 0],
[0, 0],
[0, 0],
[1, 0],
[0, 0],
[0, 1],
[0, 1],
[0, 0],
[0, 1],
[1, 1],
[0, 0],
[0, 0],
[1, 1],
[1, 1],
[0, 1],
[0, 0],
[0, 0],
[0, 1],
[0, 0],
[0, 0],
[0, 0],
[0, 0],
[1, 0],
[1, 0],
[0, 1],
[0, 0],
[0, 0],
[0, 0],
[0, 0],
[0, 0],
[1, 1],
[0, 0],
[0, 1],
[0, 0],
[0, 0],
[1, 0],
[0, 0],
[0, 1],
[0, 1],
[0, 0],
[1, 1],
[0, 1],
[0, 1],
[0, 0],
[0, 0],
```

```
       [0, 0],
       [0, 1],
       [1, 0],
       [1, 0],
       [0, 1],
       [0, 0],
       [0, 0],
       [1, 1],
       [0, 1],
       [0, 0]])
```

```
b=hd[:2,1:3]#array is mutable
print(b)
b[0,0]
```

```
[[0 0]
 [0 1]]

0
```

```
#b[0,0]=2
```

```
print(hd)
```

```
[[    97      0      0 ...      1 237000    358]
 [   180      0      1 ...      1 160000    231]
 [    31      1      1 ...      1 263358    582]
 ...
 [    14      1      1 ...      0 166000    582]
 [    80      0      1 ...      1 297000    897]
 [    16      0      0 ...      1 276000     52]]
```

```
b[0,0]=0
```

```
#print(c[[0,1,2],[0,1,0]])
#print(c[0,0],b[1,1],b[2,0])
c=print(hd[[0,1,2,3,4,5,6,7,8,9,10,11,12],[0,1,2,3,4,5,6,7,8,9,10,11,12]]
)
```

```
[    97      0      1      0      0      0      0     94     45    144
      1 266000    379]
```

```
d=np.array([0,1,2,3,4,5,6,7,8,9,10,11,12])
print(hd[np.arange(13),d])
hd[np.arange(13),d]+=10
print(hd)
```

```
[    97      0      1      0      0      0      0     94     45    144
      1 266000    379]
[[   107      0      0 ...      1 237000    358]
 [   180     10      1 ...      1 160000    231]
```

```
 [    31      1     11 ...      1 263358    582]
 ...
 [    14      1      1 ...      0 166000    582]
 [    80      0      1 ...      1 297000    897]
 [    16      0      0 ...      1 276000     52]]
```

```
s = np.array([[10,20,30,40,50,60,70,80,90,15,25,35,45]])
p= hd*s
print(p)
```

```
[[    1070        0        0 ...       25  8295000    16110]
 [    1800      200       30 ...       25  5600000    10395]
 [     310       20      330 ...       25  9217530    26190]
 ...
 [     140       20       30 ...        0  5810000    26190]
 [     800        0       30 ...       25 10395000    40365]
 [     160        0        0 ...       25  9660000     2340]]
```

```
p= hd+s
print(p)
```

```
[[   117     20     30 ...     26 237035    403]
 [   190     30     31 ...     26 160035    276]
 [    41     21     41 ...     26 263393    627]
 ...
 [    24     21     31 ...     25 166035    627]
 [    90     20     31 ...     26 297035    942]
 [    26     20     30 ...     26 276035     97]]
```

```
p= hd-s
print(d)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12]
```

```
p= hd/s
print(p)
```

```
[[1.07000000e+01 0.00000000e+00 0.00000000e+00 ... 4.00000000e-02
  6.77142857e+03 7.95555556e+00]
 [1.80000000e+01 5.00000000e-01 3.33333333e-02 ... 4.00000000e-02
  4.57142857e+03 5.13333333e+00]
 [3.10000000e+00 5.00000000e-02 3.66666667e-01 ... 4.00000000e-02
  7.52451429e+03 1.29333333e+01]
 ...
 [1.40000000e+00 5.00000000e-02 3.33333333e-02 ... 0.00000000e+00
  4.74285714e+03 1.29333333e+01]
 [8.00000000e+00 0.00000000e+00 3.33333333e-02 ... 4.00000000e-02
  8.48571429e+03 1.99333333e+01]
 [1.60000000e+00 0.00000000e+00 0.00000000e+00 ... 4.00000000e-02
  7.88571429e+03 1.15555556e+00]]
```

```
np.ndim(p)
```

```
2
```

```
print(s)
np.ndim(s)
```

```
[[10 20 30 40 50 60 70 80 90 15 25 35 45]]
```

```
2
```

```
#mathematical operations
```

```
print(np.sqrt(hd,p))
```

```
[[ 10.34408043   0.           0.         ...   1.         486.82645779
   18.92088793]
 [ 13.41640786   3.16227766   1.         ...   1.         400.
   15.19868415]
 [  5.56776436   1.           3.31662479 ...   1.         513.18417746
   24.12467616]
 ...
 [  3.74165739   1.           1.         ...   0.         407.43097575
   24.12467616]
 [  8.94427191   0.           1.         ...   1.         544.97706374
   29.94995826]
 [  4.           0.           0.         ...   1.         525.35702146
    7.21110255]]
```

```
m= hd-p
print(m)
```

```
[[9.66559196e+01 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  2.36513174e+05 3.39079112e+02]
 [1.66583592e+02 6.83772234e+00 0.00000000e+00 ... 0.00000000e+00
  1.59600000e+05 2.15801316e+02]
 [2.54322356e+01 0.00000000e+00 7.68337521e+00 ... 0.00000000e+00
  2.62844816e+05 5.57875324e+02]
 ...
 [1.02583426e+01 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  1.65592569e+05 5.57875324e+02]
 [7.10557281e+01 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  2.96455023e+05 8.67050042e+02]
 [1.20000000e+01 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  2.75474643e+05 4.47888974e+01]]
```

```
m= hd+p
print(m)
```

```
[[1.17344080e+02 0.00000000e+00 0.00000000e+00 ... 2.00000000e+00
  2.37486826e+05 3.76920888e+02]
 [1.93416408e+02 1.31622777e+01 2.00000000e+00 ... 2.00000000e+00
```

```
  1.60400000e+05 2.46198684e+02]
 [3.65677644e+01 2.00000000e+00 1.43166248e+01 ... 2.00000000e+00
  2.63871184e+05 6.06124676e+02]
 ...
 [1.77416574e+01 2.00000000e+00 2.00000000e+00 ... 0.00000000e+00
  1.66407431e+05 6.06124676e+02]
 [8.89442719e+01 0.00000000e+00 2.00000000e+00 ... 2.00000000e+00
  2.97544977e+05 9.26949958e+02]
 [2.00000000e+01 0.00000000e+00 0.00000000e+00 ... 2.00000000e+00
  2.76525357e+05 5.92111026e+01]]
```

```
m= hd*p
print(m)
```

```
[[1.10681661e+03 0.00000000e+00 0.00000000e+00 ... 1.00000000e+00
  1.15377870e+08 6.77367788e+03]
 [2.41495342e+03 3.16227766e+01 1.00000000e+00 ... 1.00000000e+00
  6.40000000e+07 3.51089604e+03]
 [1.72600695e+02 1.00000000e+00 3.64828727e+01 ... 1.00000000e+00
  1.35151159e+08 1.40405615e+04]
 ...
 [5.23832034e+01 1.00000000e+00 1.00000000e+00 ... 0.00000000e+00
  6.76335420e+07 1.40405615e+04]
 [7.15541753e+02 0.00000000e+00 1.00000000e+00 ... 1.00000000e+00
  1.61858188e+08 2.68651126e+04]
 [6.40000000e+01 0.00000000e+00 0.00000000e+00 ... 1.00000000e+00
  1.44998538e+08 3.74977333e+02]]
```

```
m= hd/p
print(m)
```

```
[[ 10.34408043          nan          nan ...   1.         486.82645779
   18.92088793]
 [ 13.41640786   3.16227766   1.         ...   1.         400.
   15.19868415]
 [  5.56776436   1.           3.31662479 ...   1.         513.18417746
   24.12467616]
 ...
 [  3.74165739   1.           1.         ...          nan 407.43097575
   24.12467616]
 [  8.94427191          nan   1.         ...   1.         544.97706374
   29.94995826]
 [  4.                  nan          nan ...   1.         525.35702146
    7.21110255]]
```

```
<ipython-input-55-ad3e85116629>:1: RuntimeWarning: invalid value
encountered in true_divide
  m= hd/p
```

```
print(hd.T)
```

```
[[   107    180     31 ...     14     80     16]
 [     0     10      1 ...      1      0      0]
 [     0      1     11 ...      1      1      0]
 ...
 [     1      1      1 ...      0      1      1]
 [237000 160000 263358 ... 166000 297000 276000]
 [   358    231    582 ...    582    897     52]]
```

```
print(hd)
```

```
[[   107      0      0 ...      1 237000    358]
 [   180     10      1 ...      1 160000    231]
 [    31      1     11 ...      1 263358    582]
 ...
 [    14      1      1 ...      0 166000    582]
 [    80      0      1 ...      1 297000    897]
 [    16      0      0 ...      1 276000     52]]
```

```
#broadcasting
x= np.array([1,1,1,1,1,1,1,1,1,1,1,1])
y= np.empty_like(hd)
for i in range(299):
    y[i,:] = hd[i,:] + x
print(y)
```

```
[[   108      1      1 ...      2 237001    359]
 [   181     11      2 ...      2 160001    232]
 [    32      2     12 ...      2 263359    583]
 ...
 [    15      2      2 ...      1 166001    583]
 [    81      1      2 ...      2 297001    898]
 [    17      1      1 ...      2 276001     53]]
```

Project data on Pandas & matplotlib

```
import pandas as pd
import matplotlib as mlt
import matplotlib.pyplot as plt
#from matplot import pyplot as plt
import numpy as np
```

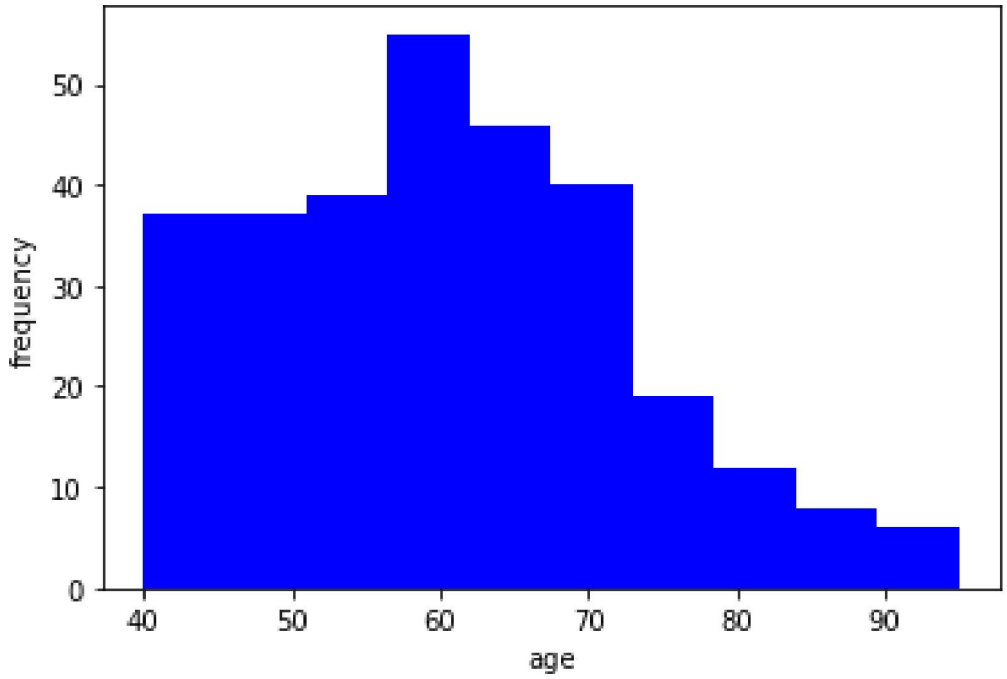
```
phdf = pd.read_csv('E:\DATA ASSOCIATE\POHD PROJECT\data_file\S1Data
(2).csv')
```

```
print(phdf)
```

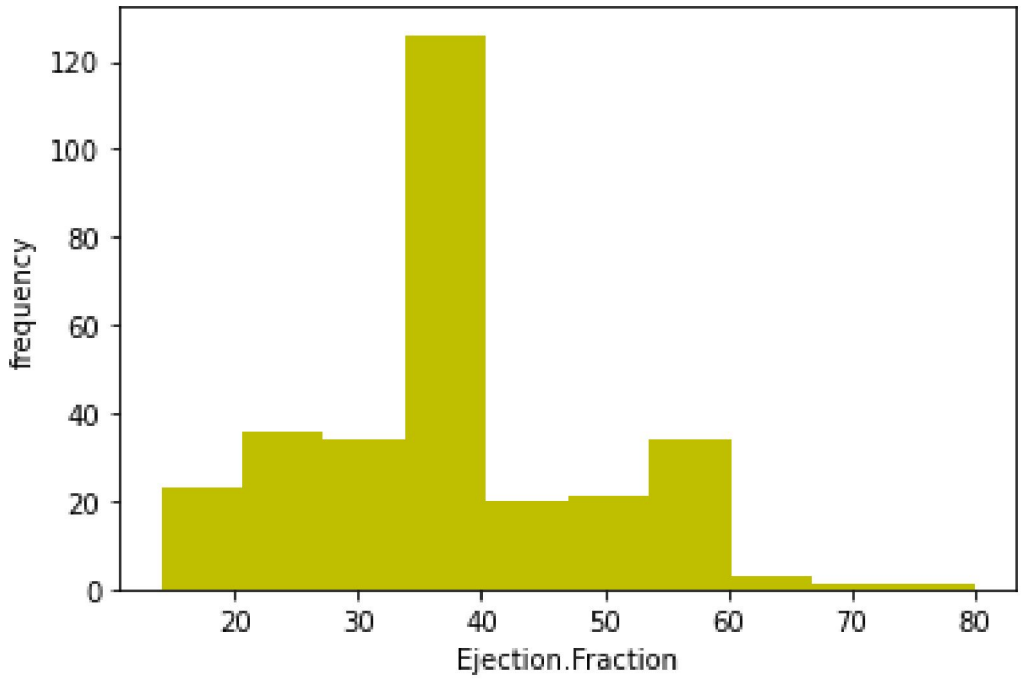
```
      Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
0    43.0       0         0        1                 50
0
```

```
1    73.0       1         1        1                30
0
2    70.0       1         0        0                20
1
3    65.0       1         0        1                25
0
4    64.0       1         0        0                60
0
..    ...     ...       ...      ...               ...
...
294  45.0       0         1        0                55
0
295  51.0       0         1        0                40
0
296  45.0       1         0        0                14
0
297  60.0       1         1        0                45
0
298  65.0       0         0        1                25
1

     creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium
\
0                         358  237000.00              1.30           135
1                         231  160000.00              1.18           142
2                         582  263358.03              1.83           134
3                         305  298000.00              1.10           141
4                        1610  242000.00              1.00           137
..                        ...        ...               ...           ...
294                       582  543000.00              1.00           132
295                       582  221000.00              0.90           134
296                       582  166000.00              0.80           127
297                       897  297000.00              1.00           133
298                        52  276000.00              1.30           137

     Smoking  TIME  DEATH_EVENT
0          0    97            0
1          1   180            0
2          1    31            1
3          0    87            0
4          0   113            0
..       ...   ...          ...
294        0   250            0
295        0   244            0
296        0    14            1
297        0    80            0
298        0    16            0

[299 rows x 13 columns]
```

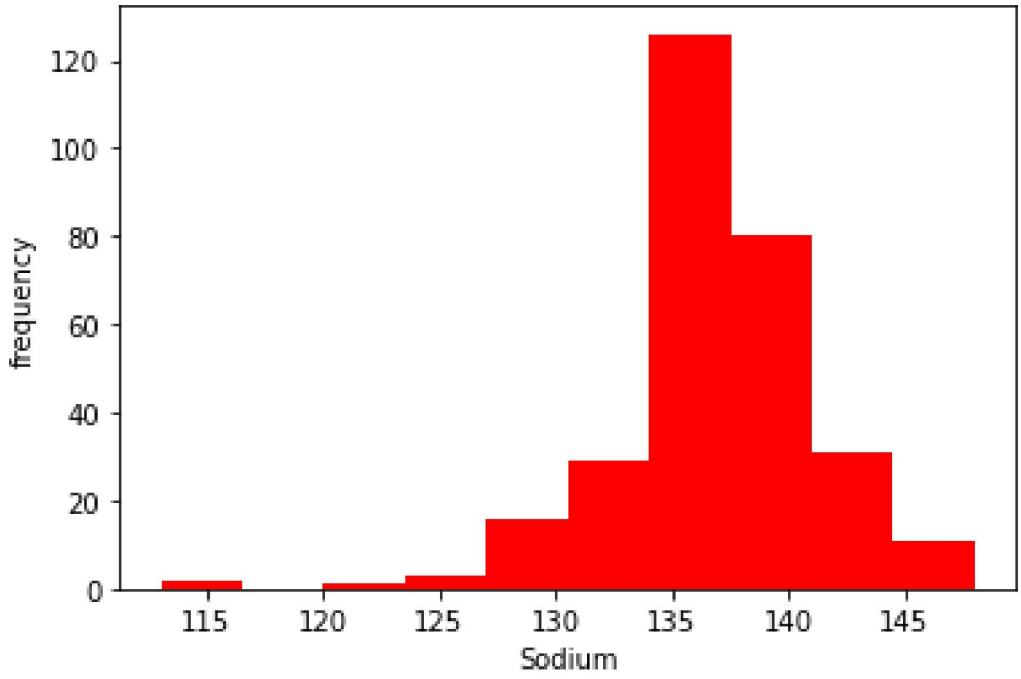
```
#Histogram

plt.hist(phdf.Age,bins= 10,color = 'b')
plt.xlabel('age')
plt.ylabel('frequency')
plt.show()
```



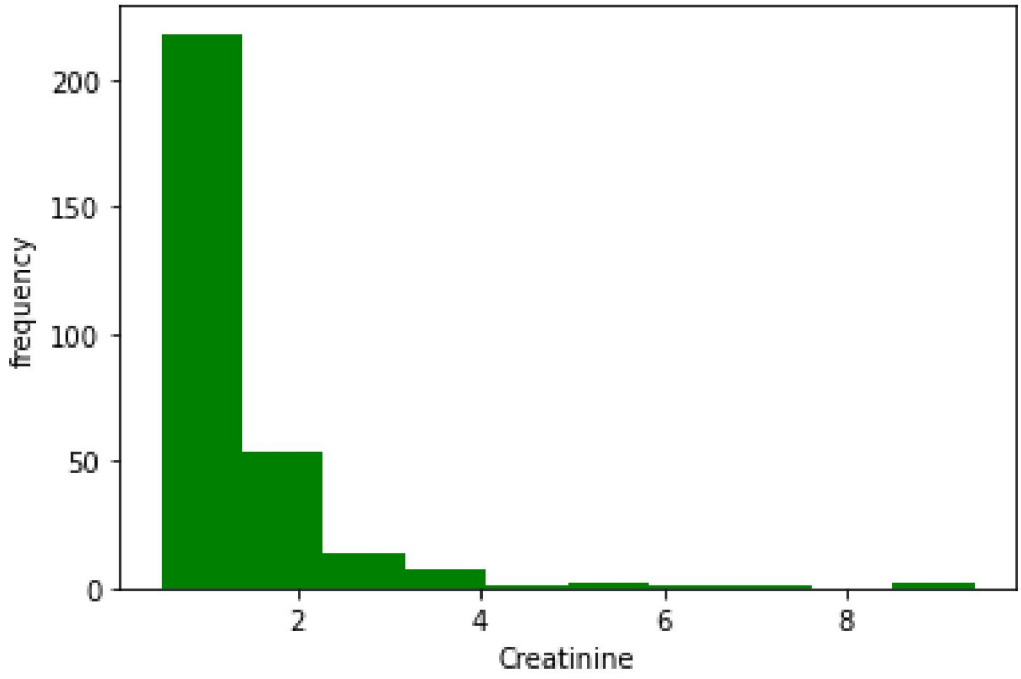

```
plt.hist(phdf.Ejection_Fraction,bins= 10,color = 'y')
plt.xlabel('Ejection.Fraction')
plt.ylabel('frequency')
plt.show()
```



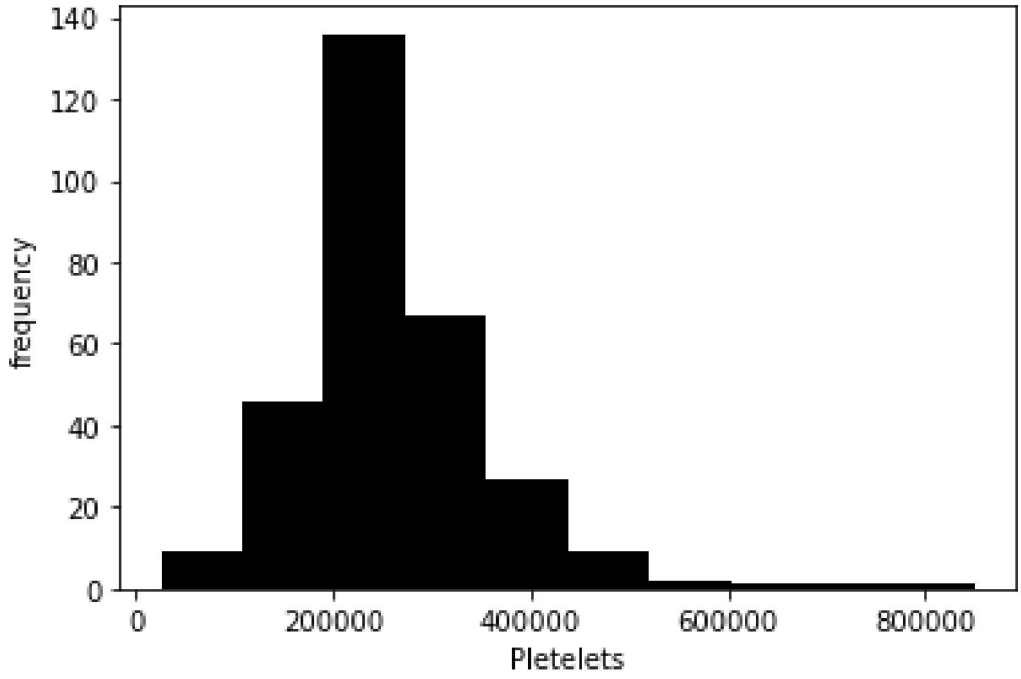
```
plt.hist(phdf.Sodium  ,bins= 10,color = 'r')
plt.xlabel('Sodium')
plt.ylabel('frequency')
plt.show()
```



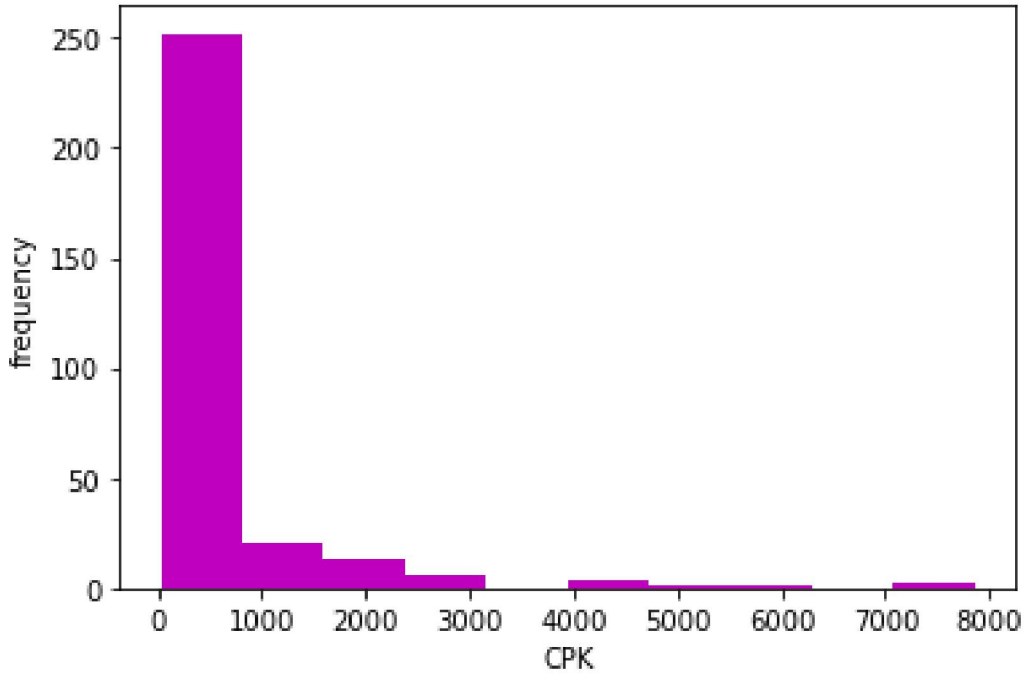

```
plt.hist(phdf.Creatinine  ,bins= 10,color = 'g')
plt.xlabel('Creatinine')
plt.ylabel('frequency')
plt.show()
```



```
plt.hist(phdf.Pletelets   ,bins= 10,color = 'k')
plt.xlabel('Pletelets')
plt.ylabel('frequency')
plt.show()
```

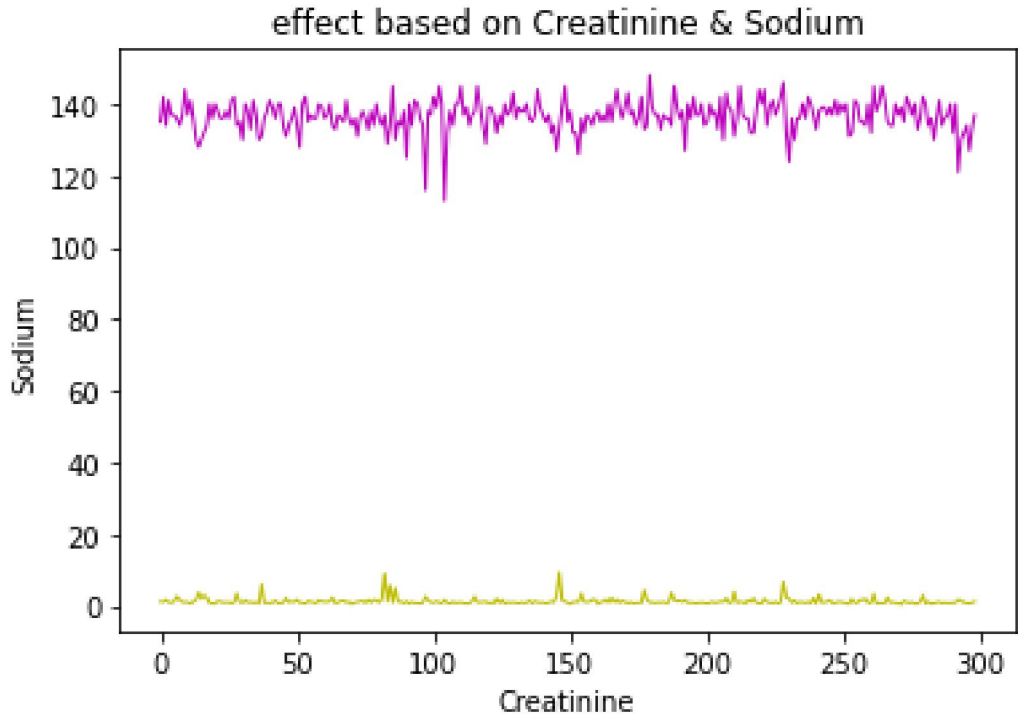



```
plt.hist(phdf.CPK,bins= 10,color = 'm')
plt.xlabel('CPK')
plt.ylabel('frequency')
plt.show()
```

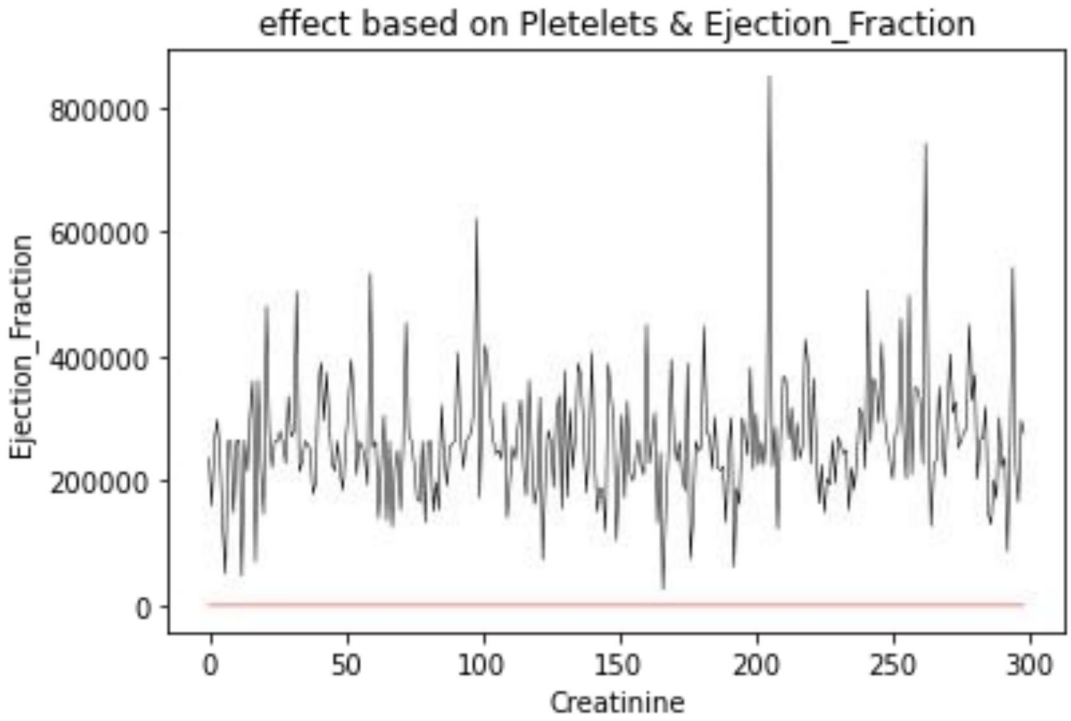



```
#line chart
```

```
plt.plot(phdf.Creatinine,'y',phdf.Sodium,'m',linewidth= 1)
plt.title('effect based on Creatinine & Sodium')
plt.xlabel('Creatinine')
plt.ylabel('Sodium')
plt.show()
```

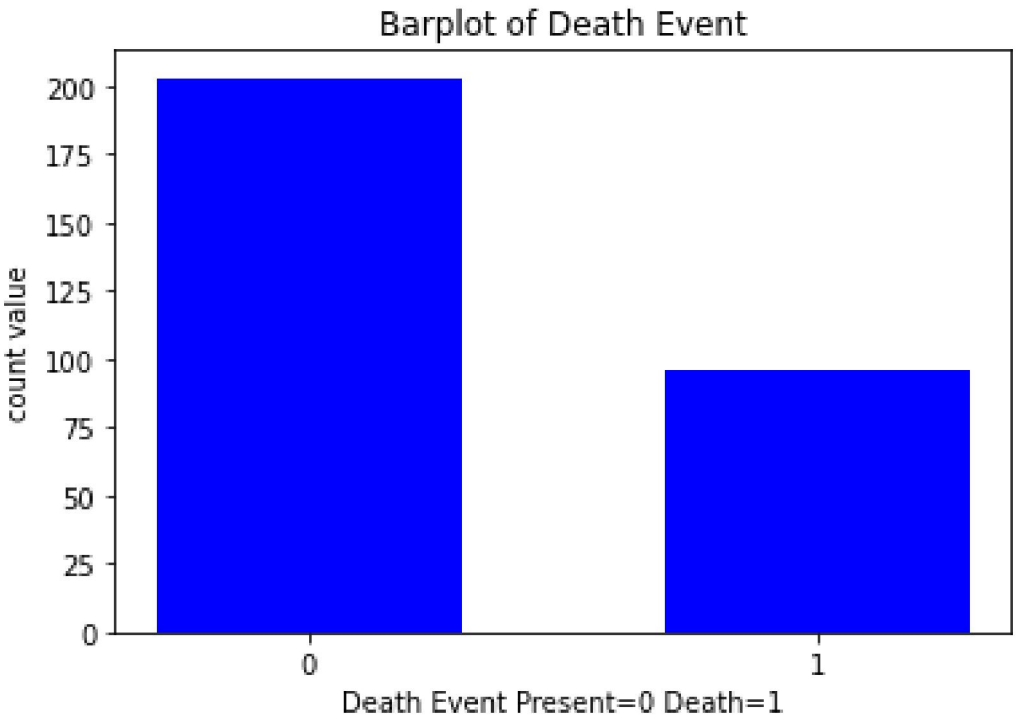



```
plt.plot(phdf.Pletelets,'k',phdf.Ejection_Fraction,'r',linewidth= 0.5)
plt.title('effect based on Pletelets & Ejection_Fraction')
plt.xlabel('Creatinine')
plt.ylabel('Ejection_Fraction')
plt.show()
```



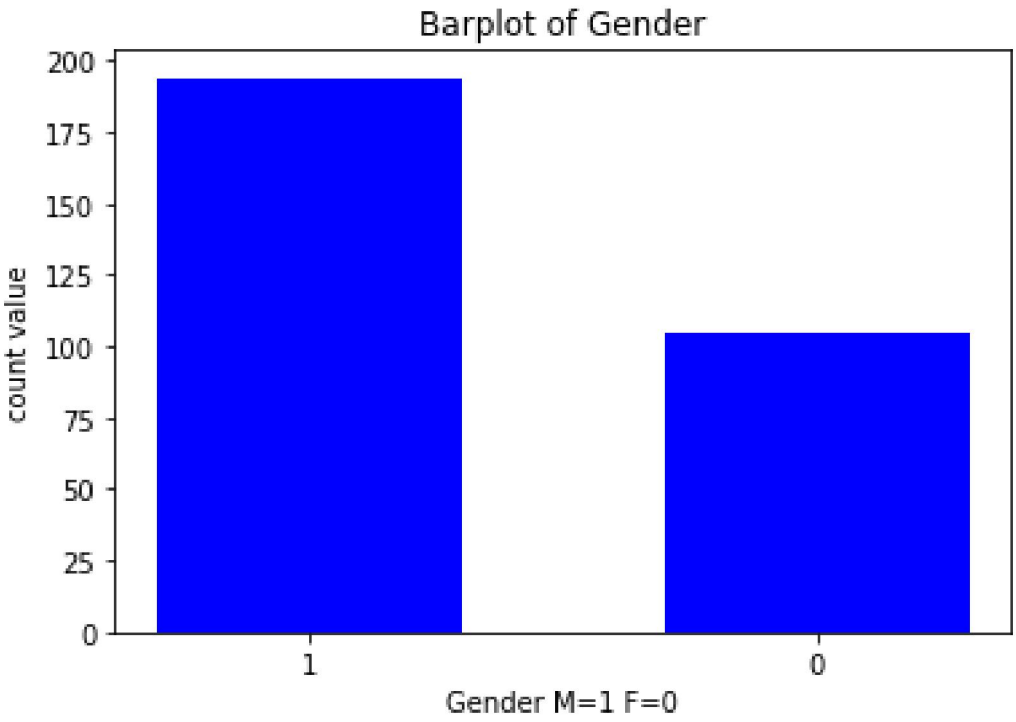

```
#bar chart

a=phdf["Event"].value_counts()
c=["0","1"]
plt.bar(c,a,color="b",width=0.6)
plt.xlabel("Death Event Present=0 Death=1")
plt.ylabel("count value")
plt.title("Barplot of Death Event")
```

```
Text(0.5, 1.0, 'Barplot of Death Event')
```



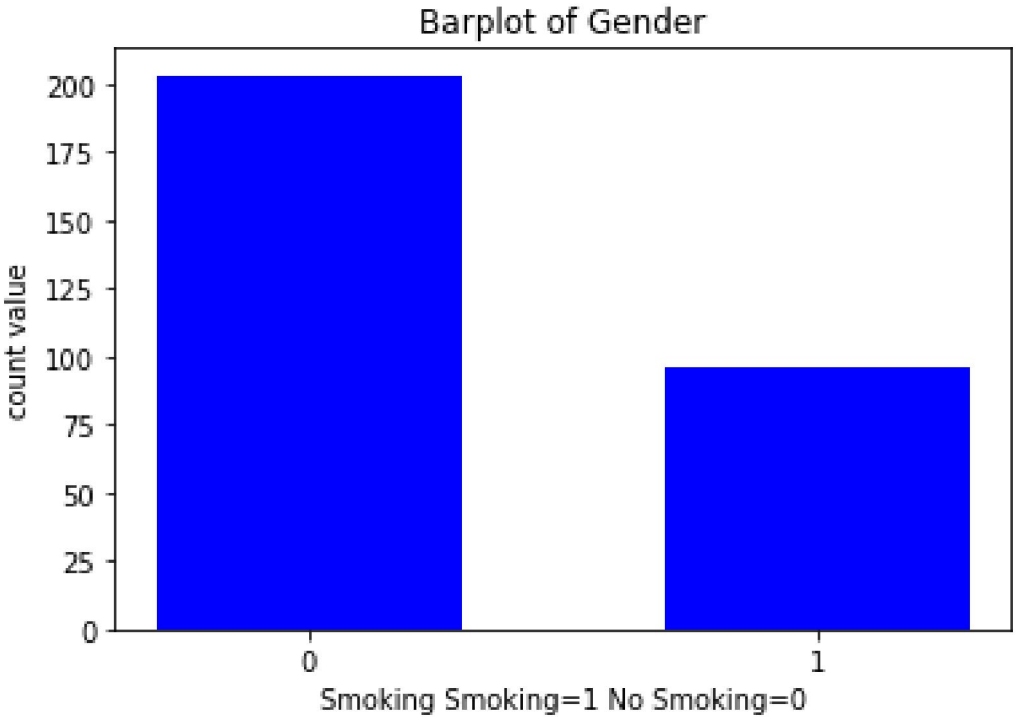

```
a=phdf["Gender"].value_counts()
c=["1","0"]
plt.bar(c,a,color="b",width=0.6)
plt.xlabel("Gender M=1 F=0")
plt.ylabel("count value")
plt.title("Barplot of Gender")
```

```
Text(0.5, 1.0, 'Barplot of Gender')
```



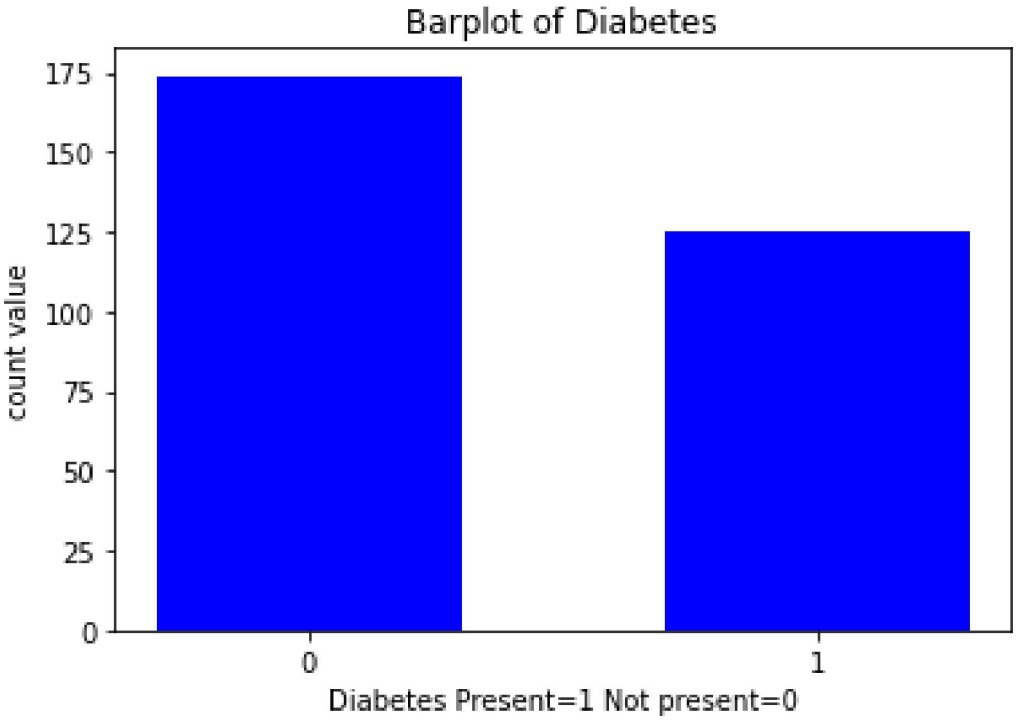
```
a=phdf["Smoking"].value_counts()
c=["0","1"]
plt.bar(c,a,color="b",width=0.6)
plt.xlabel("Smoking Smoking=1 No Smoking=0")
plt.ylabel("count value")
plt.title("Barplot of Gender")
print(a)
```

```
0    203
1     96
Name: Smoking, dtype: int64
```



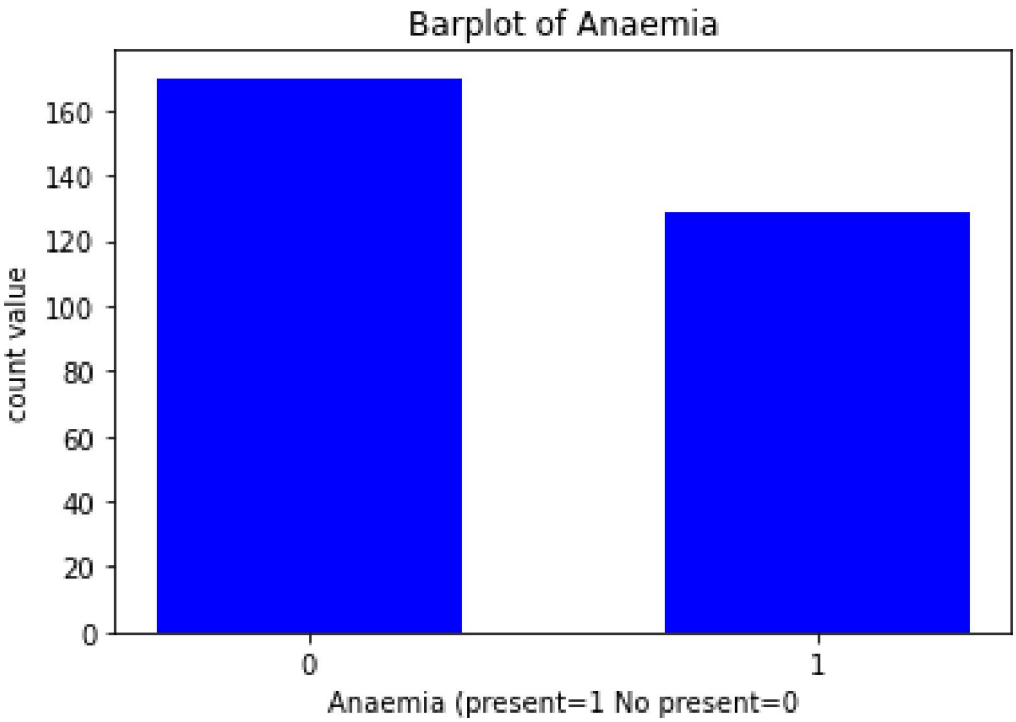

```
a=phdf["Diabetes"].value_counts()
c=["0","1"]
plt.bar(c,a,color="b",width=0.6)
plt.xlabel("Diabetes Present=1 Not present=0")
plt.ylabel("count value")
plt.title("Barplot of Diabetes")
```

```
Text(0.5, 1.0, 'Barplot of Diabetes')
```




```
a=phdf["Anaemia"].value_counts()
c=["0","1"]
plt.bar(c,a,color="b",width=0.6)
plt.xlabel("Anaemia (present=1 No present=0")
plt.ylabel("count value")
plt.title("Barplot of Anaemia")
```

```
Text(0.5, 1.0, 'Barplot of Anaemia')
```



pandas

```
print(type(phdf))
phdf.head()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
   TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
0    97      0       0        0         0   0        1  43.0
1   180      0       1        1         1   0        1  73.0
2    31      1       1        1         0   1        0  70.0
3    87      0       1        0         0   0        1  65.0
4   113      0       1        0         0   0        0  64.0

   Ejection_Fraction  Sodium  Creatinine  Pletelets   CPK
0                 50     135        1.30  237000.00   358
1                 30     142        1.18  160000.00   231
2                 20     134        1.83  263358.03   582
3                 25     141        1.10  298000.00   305
4                 60     137        1.00  242000.00  1610
```

```
phdf.tail()
```

```
     TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
294   250      0       0        0         1   0        0  45.0
295   244      0       0        0         1   0        0  51.0
296    14      1       1        0         0   0        0  45.0
297    80      0       1        0         1   0        0  60.0
298    16      0       0        0         0   1        1  65.0

     Ejection_Fraction  Sodium  Creatinine  Pletelets  CPK
294                 55     132         1.0   543000.0  582
295                 40     134         0.9   221000.0  582
296                 14     127         0.8   166000.0  582
297                 45     133         1.0   297000.0  897
298                 25     137         1.3   276000.0   52
```

```
phdf.shape
```

```
(299, 13)
```

```
phdf.columns.tolist()
```

```
['TIME',
 'Event',
 'Gender',
 'Smoking',
 'Diabetes',
 'BP',
 'Anaemia',
 'Age',
 'Ejection_Fraction',
 'Sodium',
```

```
 'Creatinine',
 'Pletelets',
 'CPK']

phdf.describe()
```

```
             TIME      Event      Gender    Smoking    Diabetes
BP  \
count  299.000000  299.00000  299.000000  299.00000  299.000000
299.000000
mean   130.260870    0.32107    0.648829    0.32107    0.418060
0.351171
std     77.614208    0.46767    0.478136    0.46767    0.494067
0.478136
min      4.000000    0.00000    0.000000    0.00000    0.000000
0.000000
25%     73.000000    0.00000    0.000000    0.00000    0.000000
0.000000
50%    115.000000    0.00000    1.000000    0.00000    0.000000
0.000000
75%    203.000000    1.00000    1.000000    1.00000    1.000000
1.000000
max    285.000000    1.00000    1.000000    1.00000    1.000000
1.000000

          Anaemia         Age  Ejection_Fraction      Sodium  Creatinine
\
count  299.000000  299.000000         299.000000  299.000000   299.00000
mean     0.431438   60.833893          38.083612  136.625418     1.39388
std      0.496107   11.894809          11.834841    4.412477     1.03451
min      0.000000   40.000000          14.000000  113.000000     0.50000
25%      0.000000   51.000000          30.000000  134.000000     0.90000
50%      0.000000   60.000000          38.000000  137.000000     1.10000
75%      1.000000   70.000000          45.000000  140.000000     1.40000
max      1.000000   95.000000          80.000000  148.000000     9.40000

           Pletelets          CPK
count     299.000000   299.000000
mean   263358.029264   581.839465
std     97804.236869   970.287881
min     25100.000000    23.000000
25%    212500.000000   116.500000
50%    262000.000000   250.000000
75%    303500.000000   582.000000
max    850000.000000  7861.000000
```

```
print(phdf.max())
print(phdf.min())
```

```
TIME                    285.0
Event                     1.0
Gender                    1.0
Smoking                   1.0
Diabetes                  1.0
BP                        1.0
Anaemia                   1.0
Age                      95.0
Ejection_Fraction        80.0
Sodium                  148.0
Creatinine                9.4
Pletelets            850000.0
CPK                    7861.0
dtype: float64
TIME                      4.0
Event                     0.0
Gender                    0.0
Smoking                   0.0
Diabetes                  0.0
BP                        0.0
Anaemia                   0.0
Age                      40.0
Ejection_Fraction        14.0
Sodium                  113.0
Creatinine                0.5
Pletelets            25100.0
CPK                      23.0
dtype: float64
```

```
phdf["Pletelets"].max()
```

```
850000.0
```

```
phdf["Ejection_Fraction"].mean()
```

```
38.08361204013378
```

```
phdf["Pletelets"].argmax()
```

```
205
```

```
phdf["Event"].value_counts()
```

```
0    203
1     96
Name: Event, dtype: int64
```

```
phdf.iloc[[109]]
```

```
     TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
109   206      0       1        0         0   1        1  55.0
```

```
     Ejection_Fraction  Sodium  Creatinine  Pletelets   CPK
109                 35     140         1.0   141000.0  2794
```

```
phdf.iloc[[109]]["Event"]
```

```
109    0
Name: Event, dtype: int64
```

```
phdf.iloc[:3,:3]
```

```
   TIME  Event  Gender
0    97      0       0
1   180      0       1
2    31      1       1
```

```
phdf.loc[:3]
```

```
   TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
0    97      0       0        0         0   0        1  43.0
1   180      0       1        1         1   0        1  73.0
2    31      1       1        1         0   1        0  70.0
3    87      0       1        0         0   0        1  65.0

   Ejection_Fraction  Sodium  Creatinine  Pletelets  CPK
0                 50     135        1.30  237000.00  358
1                 30     142        1.18  160000.00  231
2                 20     134        1.83  263358.03  582
3                 25     141        1.10  298000.00  305
```

```
phdf.loc[109,["Diabetes","Ejection_Fraction"]]
```

```
Diabetes             0.0
Ejection_Fraction    35.0
Name: 109, dtype: float64
```

```
#At() is really useful wheneever you know the row label and the column
label of the particular value that you want to get.
```

```
phdf.at[1,"Pletelets"]
```

```
160000.0
```

```
# Sorting
z=phdf.sort_values("Age").head()
z
```

```
     TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
224   187      0       0        0         0   0        1  40.0
185   148      0       1        0         1   0        0  40.0
111   209      0       1        0         0   0        1  40.0
71    212      0       1        1         0   0        0  40.0
290   244      0       1        0         1   0        0  40.0
```

```
     Ejection_Fraction  Sodium  Creatinine  Pletelets  CPK
224                 40     141         0.8   226000.0  101
185                 30     136         0.9   303000.0  478
111                 35     137         0.9   255000.0  129
71                  35     136         1.1   255000.0   90
290                 35     132         1.0   222000.0  582
```

```
a=phdf.groupby("Age")
a.get_group(50)
```

```
     TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
10    119      0       1        1         1   0        0  50.0
11    112      0       1        1         0   0        0  50.0
18    240      0       1        1         0   0        1  50.0
20    192      0       1        1         0   1        0  50.0
26    250      0       1        0         0   0        1  50.0
36    175      0       1        0         1   0        1  50.0
46     60      1       0        0         0   1        0  50.0
51     28      1       0        0         1   1        1  50.0
70    172      1       0        0         0   0        0  50.0
80    120      0       1        1         0   0        0  50.0
96    186      0       0        0         1   1        1  50.0
120   146      0       1        0         0   0        1  50.0
159     7      1       1        0         0   0        1  50.0
163    35      1       1        1         1   0        0  50.0
169   285      0       1        1         0   0        0  50.0
173   108      0       1        0         0   1        0  50.0
174   118      0       1        1         0   1        0  50.0
176   126      1       0        0         1   0        1  50.0
182    11      1       1        0         0   1        1  50.0
183   215      0       1        0         0   1        0  50.0
190    90      0       1        0         1   0        0  50.0
191    29      0       0        0         1   0        1  50.0
211   187      0       0        0         1   0        1  50.0
214   246      0       0        0         1   0        1  50.0
234    32      1       0        1         1   1        0  50.0
260   109      0       0        0         1   0        0  50.0
271   214      0       0        0         0   1        0  50.0
```

```
     Ejection_Fraction  Sodium  Creatinine  Pletelets   CPK
10                  35     137        1.18  263358.03  1846
11                  30     141        0.70  266000.00   185
18                  35     140        0.90  362000.00   298
20                  62     140        0.80  147000.00   582
26                  40     141        0.80  279000.00    54
36                  40     130        0.70  260000.00   121
46                  40     131        2.30  216000.00   318
51                  35     128        1.00  319000.00   249
70                  50     134        0.60  153000.00   582
80                  25     136        1.00  262000.00   250
```

```
96                  20     134        1.00  279000.00   582
120                 20     139        0.80  189000.00   115
159                 20     137        1.90  210000.00   111
163                 38     135        1.90  310000.00   582
169                 45     136        1.60  395000.00   196
173                 30     138        0.80  211000.00  1548
174                 45     134        0.90  184000.00   115
176                 35     142        0.90   75000.00  2334
182                 38     137        1.10  276000.00   168
183                 45     133        1.00  274000.00   245
190                 25     136        1.60  252000.00   369
191                 30     138        1.20  302000.00   159
211                 45     136        1.00  362000.00   167
214                 30     136        0.70  232000.00  1051
234                 30     136        1.20  153000.00   124
260                 30     132        0.90  329000.00   482
271                 30     139        0.50  404000.00  2522
```

```
# # Filtering Rows Conditionally
phdf[phdf["Age"]>90]
```

```
     TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
7      27      1       1        0         1   1        0  94.0
39     24      1       0        0         0   1        1  95.0
253    50      1       1        0         0   0        1  95.0

     Ejection_Fraction  Sodium  Creatinine  Pletelets  CPK
7                   38     134        1.83  263358.03  582
39                  40     138        1.00  196000.00  112
253                 30     132        2.00  461000.00  371
```

```
phdf[(phdf["Age"]>90) & (phdf["Diabetes"]==1)]
```

```
   TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
7    27      1       1        0         1   1        0  94.0

   Ejection_Fraction  Sodium  Creatinine  Pletelets  CPK
7                 38     134        1.83  263358.03  582
```

```
# # Grouping
```

```
phdf.groupby("Age")["Event"].value_counts().head(5)
```

```
Age   Event
40.0  0        7
41.0  0        1
42.0  0        6
      1        1
43.0  0        1
Name: Event, dtype: int64
```

```
phdf.values
```

```
array([[9.7000000e+01, 0.0000000e+00, 0.0000000e+00, ..., 1.3000000e+00,
        2.3700000e+05, 3.5800000e+02],
       [1.8000000e+02, 0.0000000e+00, 1.0000000e+00, ..., 1.1800000e+00,
        1.6000000e+05, 2.3100000e+02],
       [3.1000000e+01, 1.0000000e+00, 1.0000000e+00, ..., 1.8300000e+00,
        2.6335803e+05, 5.8200000e+02],
       ...,
       [1.4000000e+01, 1.0000000e+00, 1.0000000e+00, ..., 8.0000000e-01,
        1.6600000e+05, 5.8200000e+02],
       [8.0000000e+01, 0.0000000e+00, 1.0000000e+00, ..., 1.0000000e+00,
        2.9700000e+05, 8.9700000e+02],
       [1.6000000e+01, 0.0000000e+00, 0.0000000e+00, ..., 1.3000000e+00,
        2.7600000e+05, 5.2000000e+01]])
```

```
phdf.values[96][5]
```

```
1.0
```

```
phdf.loc[50]["Event"]
```

```
1.0
```

```
# # Dataframe Iteration
```

```
for index, rows in phdf.loc[:10,["Age"]].iterrows():
    print(index)
    print(rows)
```

```
0
Age    43.0
Name: 0, dtype: float64
1
Age    73.0
Name: 1, dtype: float64
2
Age    70.0
Name: 2, dtype: float64
3
Age    65.0
Name: 3, dtype: float64
4
Age    64.0
Name: 4, dtype: float64
5
Age    75.0
Name: 5, dtype: float64
6
Age    70.0
Name: 6, dtype: float64
7
Age    94.0
```

```
Name: 7, dtype: float64
8
Age    75.0
Name: 8, dtype: float64
9
Age    80.0
Name: 9, dtype: float64
10
Age    50.0
Name: 10, dtype: float64
```

```
# # Extracting Rows and Columns

phdf[["Age","Event"]].head(5)
```

```
    Age  Event
0  43.0      0
1  73.0      0
2  70.0      1
3  65.0      0
4  64.0      0
```

```
phdf.loc[:,["Pletelets","Smoking"]].head()
```

```
   Pletelets  Smoking
0  237000.00        0
1  160000.00        1
2  263358.03        1
3  298000.00        0
4  242000.00        0
```

```
phdf[0:3]
```

```
   TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
0    97      0       0        0         0   0        1  43.0
1   180      0       1        1         1   0        1  73.0
2    31      1       1        1         0   1        0  70.0

   Ejection_Fraction  Sodium  Creatinine  Pletelets  CPK
0                 50     135        1.30  237000.00  358
1                 30     142        1.18  160000.00  231
2                 20     134        1.83  263358.03  582
```

```
phdf.iloc[99:105,0:4]
```

```
     TIME  Event  Gender  Smoking
99     79      0       1        0
100   216      0       1        1
101    43      1       0        0
102    61      1       1        0
103    79      0       1        0
104   186      0       1        1
```

```
phdf.loc[:3,["Age","Diabetes","Event"]]
```

```
    Age  Diabetes  Event
0  43.0         0      0
1  73.0         1      0
2  70.0         0      1
3  65.0         0      0
```

```
# #  Data cleaning
phdf.isnull().sum()
```

```
TIME                  0
Event                 0
Gender                0
Smoking               0
Diabetes              0
BP                    0
Anaemia               0
Age                   0
Ejection_Fraction     0
Sodium                0
Creatinine            0
Pletelets             0
CPK                   0
dtype: int64
```

```
phdf.dropna()
```

```
     TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
0      97      0       0        0         0   0        1  43.0
1     180      0       1        1         1   0        1  73.0
2      31      1       1        1         0   1        0  70.0
3      87      0       1        0         0   0        1  65.0
4     113      0       1        0         0   0        0  64.0
..    ...    ...     ...      ...       ...  ..      ...   ...
294   250      0       0        0         1   0        0  45.0
295   244      0       0        0         1   0        0  51.0
296    14      1       1        0         0   0        0  45.0
297    80      0       1        0         1   0        0  60.0
298    16      0       0        0         0   1        1  65.0

     Ejection_Fraction  Sodium  Creatinine  Pletelets   CPK
0                   50     135        1.30  237000.00   358
1                   30     142        1.18  160000.00   231
2                   20     134        1.83  263358.03   582
3                   25     141        1.10  298000.00   305
4                   60     137        1.00  242000.00  1610
..                 ...     ...         ...        ...   ...
294                 55     132        1.00  543000.00   582
295                 40     134        0.90  221000.00   582
296                 14     127        0.80  166000.00   582
```

```
297                45     133        1.00  297000.00   897
298                25     137        1.30  276000.00    52

[299 rows x 13 columns]
```

```
phdf.fillna(10)
```

```
     TIME  Event  Gender  Smoking  Diabetes  BP  Anaemia   Age  \
0      97      0       0        0         0   0        1  43.0
1     180      0       1        1         1   0        1  73.0
2      31      1       1        1         0   1        0  70.0
3      87      0       1        0         0   0        1  65.0
4     113      0       1        0         0   0        0  64.0
..    ...    ...     ...      ...       ...  ..      ...   ...
294   250      0       0        0         1   0        0  45.0
295   244      0       0        0         1   0        0  51.0
296    14      1       1        0         0   0        0  45.0
297    80      0       1        0         1   0        0  60.0
298    16      0       0        0         0   1        1  65.0

     Ejection_Fraction  Sodium  Creatinine  Pletelets   CPK
0                   50     135        1.30  237000.00   358
1                   30     142        1.18  160000.00   231
2                   20     134        1.83  263358.03   582
3                   25     141        1.10  298000.00   305
4                   60     137        1.00  242000.00  1610
..                 ...     ...         ...        ...   ...
294                 55     132        1.00  543000.00   582
295                 40     134        0.90  221000.00   582
296                 14     127        0.80  166000.00   582
297                 45     133        1.00  297000.00   897
298                 25     137        1.30  276000.00    52

[299 rows x 13 columns]
```

Chi- Square Test

```
import pandas as pd
import numpy as np
import scipy.stats as stats

data = pd.read_csv('heart_failure.csv')

data.columns

Index(['Age', 'Gender', 'Diabetes', 'Anaemia', 'Ejection_Fraction',
       'High_Blood_Pressure', 'creatinine_phosphokinase', 'Pletelets',
       'Serum_Creatinine', 'Serum_Sodium', 'Smoking', 'TIME',
'DEATH_EVENT'],
      dtype='object')

data_table_smoke=pd.crosstab(data['Smoking'],data['DEATH_EVENT'])
print(data_table_smoke)

DEATH_EVENT    0   1
Smoking
0            137  66
1             66  30

#observed value
obs_value_smoke= data_table_smoke.values
print('observed values:-\n',obs_value_smoke)

observed values:-
 [[137  66]
 [ 66  30]]

val_1= stats.chi2_contingency(obs_value_smoke)
val_1

(0.007331473567119502,
 0.9317652998235507,
 1,
 array([[137.82274247,  65.17725753],
        [ 65.17725753,  30.82274247]]))

Expected_value_smoke= val_1[3]

Expected_value_smoke

array([[137.82274247,  65.17725753],
       [ 65.17725753,  30.82274247]])

no_of_rows =  len(obs_value_smoke[0:2,0])
no_of_columns= len(obs_value_smoke[0,0:2])
ddof=(no_of_rows-1)*(no_of_columns-1)
print('Degree of Freedom:-',ddof)
alpha = 0.05
```

```
Degree of Freedom:- 1
```

```
from scipy.stats import chi2
chi_square= sum([(o-e)**2./e for o,e in
zip(obs_value_smoke,Expected_value_smoke)])
chi_square_statistc= chi_square[0]+chi_square[1]
print(chi_square_statistc)
```

```
0.047643851312819833
```

```
print('chi-square statistic :-',chi_square_statistc)
```

```
chi-square statistic :- 0.047643851312819833
```

```
critical_value= chi2.ppf(q=1-alpha,df=ddof)
print('critical_value: ', critical_value)
```

```
critical_value:  3.841458820694124
```

```
#p-value
p_value_1 = chi2.cdf(x=chi_square_statistc,df=ddof)
print('p-value_1: ',p_value_1)
print('significance level: ',alpha)
print('Degree fo freedom: ',ddof)
```

```
p-value_1:  0.17278492618676242
significance level:  0.05
Degree fo freedom:  1
```

```
p_value= 1-p_value_1
print(p_value)
```

```
0.8272150738132376
```

```
if chi_square_statistc>=critical_value:
    print('Rejcet Ho ,There is  relationship between 2 categorical
varivales')
else:
    print('cannot Rejcet Ho ,There is no relationship between 2
categorical varivales')
```

```
if p_value<=alpha:
    print('Rejcet Ho ,There is no relationship between 2 categorical
varivales')
else:
    print('cannot Rejcet Ho ,There is  relationship between 2 categorical
varivales')
```

```
cannot Rejcet Ho ,There is no relationship between 2 categorical
varivales
cannot Rejcet Ho ,There is  relationship between 2 categorical varivales
```

Logistic Regression

```
import numpy as np
import pandas as pd

data = pd.read_csv('heart_failure.csv')

data
```

```
      Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
0    43.0       0         0        1                 50
0
1    73.0       1         1        1                 30
0
2    70.0       1         0        0                 20
1
3    65.0       1         0        1                 25
0
4    64.0       1         0        0                 60
0
..    ...     ...       ...      ...                ...
...
294  45.0       0         1        0                 55
0
295  51.0       0         1        0                 40
0
296  45.0       1         0        0                 14
0
297  60.0       1         1        0                 45
0
298  65.0       0         0        1                 25
1

     creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium
\
0                         358  237000.00              1.30           135
1                         231  160000.00              1.18           142
2                         582  263358.03              1.83           134
3                         305  298000.00              1.10           141
4                        1610  242000.00              1.00           137
..                        ...        ...               ...           ...
294                       582  543000.00              1.00           132
295                       582  221000.00              0.90           134
296                       582  166000.00              0.80           127
297                       897  297000.00              1.00           133
298                        52  276000.00              1.30           137

     Smoking  TIME  DEATH_EVENT
0          0    97            0
1          1   180            0
```

```
2          1    31            1
3          0    87            0
4          0   113            0
..       ...   ...          ...
294        0   250            0
295        0   244            0
296        0    14            1
297        0    80            0
298        0    16            0

[299 rows x 13 columns]
```

```
count_0, count_1 = data.DEATH_EVENT.value_counts()
```

```
count_0
```

```
203
```

```
count_1
```

```
96
```

```
data_count0 = data[data["DEATH_EVENT"] == 0]
```

```
data_count1 = data[data["DEATH_EVENT"] == 1]
```

```
data_count0
```

```
      Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
0    43.0       0         0        1                 50
0
1    73.0       1         1        1                 30
0
3    65.0       1         0        1                 25
0
4    64.0       1         0        0                 60
0
6    70.0       1         0        0                 40
0
..    ...     ...       ...      ...                ...
...
293  44.0       1         1        0                 30
1
294  45.0       0         1        0                 55
0
295  51.0       0         1        0                 40
0
297  60.0       1         1        0                 45
0
298  65.0       0         0        1                 25
1
```

```
     creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium
\
0                         358  237000.00              1.30           135
1                         231  160000.00              1.18           142
3                         305  298000.00              1.10           141
4                        1610  242000.00              1.00           137
6                         582   51000.00              2.70           136
..                        ...        ...               ...           ...
293                       582  263358.03              1.60           130
294                       582  543000.00              1.00           132
295                       582  221000.00              0.90           134
297                       897  297000.00              1.00           133
298                        52  276000.00              1.30           137

     Smoking  TIME  DEATH_EVENT
0          0    97            0
1          1   180            0
3          0    87            0
4          0   113            0
6          1   250            0
..       ...   ...          ...
293        1   244            0
294        0   250            0
295        0   244            0
297        0    80            0
298        0    16            0

[203 rows x 13 columns]

data_count1

      Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
2    70.0       1         0        0                 20
1
5    75.0       1         0        1                 15
0
7    94.0       1         1        0                 38
1
12   82.0       1         0        1                 50
0
13   75.0       1         0        0                 20
1
..    ...     ...       ...      ...                ...
...
279  60.0       0         1        0                 38
0
284  49.0       1         0        0                 20
1
```

```
291  70.0       0         0        1                 25
1
292  48.0       0         1        1                 55
0
296  45.0       1         0        0                 14
0

     creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium
\
2                         582  263358.03              1.83           134
5                         246  127000.00              1.20           137
7                         582  263358.03              1.83           134
12                        379   47000.00              1.30           136
13                        582  265000.00              1.90           130
..                        ...        ...               ...           ...
279                       235  329000.00              3.00           142
284                       789  319000.00              1.10           136
291                       125  237000.00              1.00           140
292                       582   87000.00              1.90           121
296                       582  166000.00              0.80           127

     Smoking  TIME  DEATH_EVENT
2          1    31            1
5          0    10            1
7          0    27            1
12         0    13            1
13         0     4            1
..       ...   ...          ...
279        0    30            1
284        1    55            1
291        0    15            1
292        0    15            1
296        0    14            1

[96 rows x 13 columns]
```

```
data_count_over = data_count1.sample(count_0,replace= True)
```

```
data_count_over.shape
```

```
(203, 13)
```

```
new_data = pd.concat([data_count0, data_count_over], axis= 0)
```

```
new_data
```

```
      Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
0    43.0       0         0        1                 50
0
1    73.0       1         1        1                 30
```

```
0
3    65.0       1         0        1                 25
0
4    64.0       1         0        0                 60
0
6    70.0       1         0        0                 40
0
..    ...     ...       ...      ...                ...
...
177  80.0       1         0        1                 20
1
145  60.0       1         0        0                 20
0
232  48.0       0         1        1                 30
1
284  49.0       1         0        0                 20
1
98   72.0       0         0        1                 30
1

     creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium
\
0                         358   237000.0              1.30           135
1                         231   160000.0              1.18           142
3                         305   298000.0              1.10           141
4                        1610   242000.0              1.00           137
6                         582    51000.0              2.70           136
..                        ...        ...               ...           ...
177                       553   140000.0              4.40           133
145                        68   119000.0              2.90           127
232                       131   244000.0              1.60           130
284                       789   319000.0              1.10           136
98                        328   621000.0              1.70           138

     Smoking  TIME  DEATH_EVENT
0          0    97            0
1          1   180            0
3          0    87            0
4          0   113            0
6          1   250            0
..       ...   ...          ...
177        0    41            1
145        1    64            1
232        0   193            1
284        1    55            1
98         1    88            1

[406 rows x 13 columns]

new_data.shape
```

```
(406, 13)
```

```
new_data["DEATH_EVENT"].value_counts()
```

```
0    203
1    203
Name: DEATH_EVENT, dtype: int64
```

```
x = new_data.iloc[:,0:-1]
```

```
x
```

```
      Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
0    43.0       0         0        1                 50
0
1    73.0       1         1        1                 30
0
3    65.0       1         0        1                 25
0
4    64.0       1         0        0                 60
0
6    70.0       1         0        0                 40
0
..    ...     ...       ...      ...                ...
...
177  80.0       1         0        1                 20
1
145  60.0       1         0        0                 20
0
232  48.0       0         1        1                 30
1
284  49.0       1         0        0                 20
1
98   72.0       0         0        1                 30
1

     creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium
\
0                         358   237000.0              1.30           135
1                         231   160000.0              1.18           142
3                         305   298000.0              1.10           141
4                        1610   242000.0              1.00           137
6                         582    51000.0              2.70           136
..                        ...        ...               ...           ...
177                       553   140000.0              4.40           133
145                        68   119000.0              2.90           127
232                       131   244000.0              1.60           130
284                       789   319000.0              1.10           136
98                        328   621000.0              1.70           138
```

```
     Smoking  TIME
0          0    97
1          1   180
3          0    87
4          0   113
6          1   250
..       ...   ...
177        0    41
145        1    64
232        0   193
284        1    55
98         1    88

[406 rows x 12 columns]
```

```
y = new_data.iloc[:,-1]
```

```
y
```

```
0      0
1      0
3      0
4      0
6      0
      ..
177    1
145    1
232    1
284    1
98     1
Name: DEATH_EVENT, Length: 406, dtype: int64
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x, y,test_size=0.2,
random_state= True)
```

```
from sklearn.linear_model import LogisticRegression
```

```
clf = LogisticRegression()
```

```
clf.fit(x_train, y_train)
```

```
LogisticRegression()
```

```
predict = clf.predict(x_test)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix,
precision_score,classification_report
```

```
confusion_matrix(y_test,predict)
```

```
array([[32,  6],
       [11, 33]], dtype=int64)
```

```
accuracy_score(y_test, predict)*100
```

```
79.26829268292683
```

```
precision_score(y_test, predict)*100
```

```
84.61538461538461
```

```
a=classification_report(y_test,predict)
print(a)
```

```
              precision    recall  f1-score   support

           0       0.74      0.84      0.79        38
           1       0.85      0.75      0.80        44

    accuracy                           0.79        82
   macro avg       0.80      0.80      0.79        82
weighted avg       0.80      0.79      0.79        82
```

```
import statsmodels.api as sm
```

```
logit_model= sm.Logit(y_train,sm.add_constant(x_train))
logit_model
results= logit_model.fit()
stats1=results.summary()
stats2=results.summary2()
print(stats1)
print(stats2)
```

```
C:\Users\OMKAR\anaconda3\lib\site-
packages\statsmodels\tsa\tsatools.py:142: FutureWarning: In a future
version of pandas all arguments of concat except for the argument 'objs'
will be keyword-only
  x = pd.concat(x[::order], 1)
```

```
Optimization terminated successfully.
         Current function value: 0.431284
         Iterations 7
                          Logit Regression Results
==========================================================================
=====
Dep. Variable:            DEATH_EVENT   No. Observations:
324
Model:                          Logit   Df Residuals:
311
Method:                           MLE   Df Model:
12
```

```
Date:                Thu, 16 Dec 2021   Pseudo R-squ.:
0.3776
Time:                        18:50:51   Log-Likelihood:                -
139.74
converged:                       True   LL-Null:                       -
224.52
Covariance Type:            nonrobust   LLR p-value:
5.828e-30
==============================================================================
===================
                               coef    std err          z      P>|z|
[0.025      0.975]
------------------------------------------------------------------------------
-------------------
const                       15.3336      5.621      2.728      0.006
4.316      26.351
Age                          0.0321      0.014      2.276      0.023
0.004       0.060
Gender                      -0.6946      0.362     -1.916      0.055
-1.405       0.016
Diabetes                    -0.3230      0.317     -1.019      0.308
-0.944       0.298
Anaemia                     -0.1661      0.318     -0.522      0.601
-0.789       0.457
Ejection_Fraction           -0.0642      0.014     -4.619      0.000
-0.092      -0.037
High_Blood_Pressure         -0.1686      0.320     -0.526      0.599
-0.797       0.459
creatinine_phosphokinase     0.0002      0.000      1.530      0.126   -
6.45e-05       0.001
Pletelets                -1.387e-06   1.48e-06     -0.936      0.349   -
4.29e-06    1.52e-06
Serum_Creatinine             0.6053      0.163      3.713      0.000
0.286       0.925
Serum_Sodium                -0.0926      0.039     -2.382      0.017
-0.169      -0.016
Smoking                     -0.3433      0.367     -0.937      0.349
-1.062       0.375
TIME                        -0.0198      0.003     -7.392      0.000
-0.025      -0.015
==============================================================================
===================
                           Results: Logit
================================================================
Model:                 Logit              Pseudo R-squared:   0.378
Dependent Variable:    DEATH_EVENT        AIC:                305.4718
Date:                  2021-12-16 18:50   BIC:                354.6215
No. Observations:      324                Log-Likelihood:     -139.74
Df Model:              12                 LL-Null:            -224.52
```

```
Df Residuals:          311                LLR p-value:        5.8279e-30
Converged:             1.0000             Scale:              1.0000
No. Iterations:        7.0000
------------------------------------------------------------------------
                          Coef.  Std.Err.    z    P>|z|   [0.025  0.975]
------------------------------------------------------------------------
const                   15.3336   5.6212  2.7278 0.0064  4.3162 26.3509
Age                      0.0321   0.0141  2.2760 0.0228  0.0045  0.0597
Gender                  -0.6946   0.3625 -1.9161 0.0553 -1.4051  0.0159
Diabetes                -0.3230   0.3169 -1.0192 0.3081 -0.9441  0.2981
Anaemia                 -0.1661   0.3181 -0.5223 0.6015 -0.7895  0.4573
Ejection_Fraction       -0.0642   0.0139 -4.6190 0.0000 -0.0915 -0.0370
High_Blood_Pressure     -0.1686   0.3204 -0.5261 0.5988 -0.7967  0.4595
creatinine_phosphokinase 0.0002   0.0001  1.5298 0.1261 -0.0001  0.0005
Pletelets               -0.0000   0.0000 -0.9359 0.3493 -0.0000  0.0000
Serum_Creatinine         0.6053   0.1630  3.7128 0.0002  0.2858  0.9248
Serum_Sodium            -0.0926   0.0389 -2.3815 0.0172 -0.1688 -0.0164
Smoking                 -0.3433   0.3666 -0.9365 0.3490 -1.0619  0.3752
TIME                    -0.0198   0.0027 -7.3916 0.0000 -0.0250 -0.0145
========================================================================
```

Decision Tree

```
x_train.shape
```

```
(324, 12)
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
clf2 = DecisionTreeClassifier()
```

```
clf2.fit(x_train,y_train)
```

```
DecisionTreeClassifier()
```

```
predict = clf2.predict(x_test)
```

```
predict
```

```
array([1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0], dtype=int64)
```

```
confusion_matrix(y_test,predict)
```

```
array([[35,  3],
       [ 4, 40]], dtype=int64)
```

```
accuracy_score(y_test, predict)*100
```

```
91.46341463414635
```

Heat Map for Continuous Data

```
select= ['Age', 'Ejection_Fraction', 'creatinine_phosphokinase',
'Pletelets', 'Serum_Creatinine', 'Serum_Sodium', 'TIME']
z=(data[select])
```

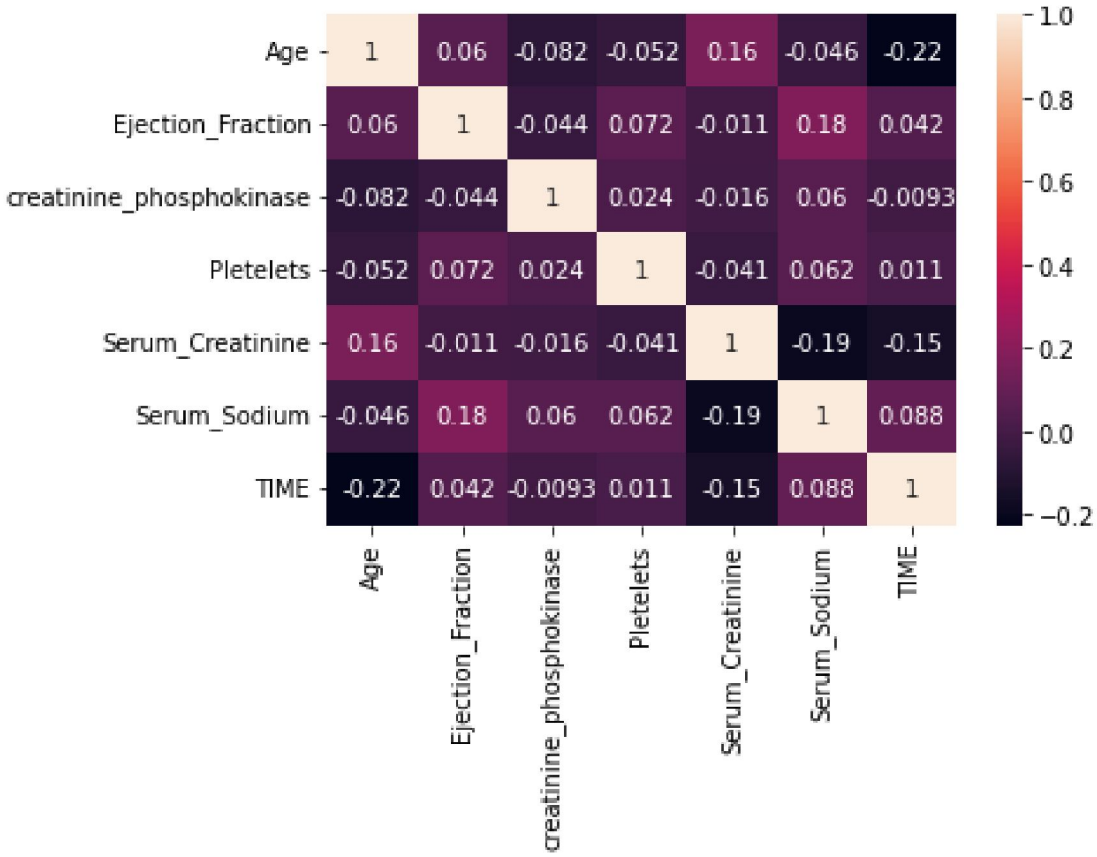
```
select[0]
```

```
'Age'
```

```
import seaborn as sb
```

```
sb.heatmap(z.corr(),annot= True)
```

```
<AxesSubplot:>
```



Random Forest & cross validation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data=pd.read_csv("Heart_failure_csv.csv")
data.head()
```

```
    Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
0  43.0       0         0        1                 50
0
1  73.0       1         1        1                 30
0
2  70.0       1         0        0                 20
1
3  65.0       1         0        1                 25
0
4  64.0       1         0        0                 60
0

   creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium  \
0                       358  237000.00              1.30           135
1                       231  160000.00              1.18           142
2                       582  263358.03              1.83           134
3                       305  298000.00              1.10           141
4                      1610  242000.00              1.00           137

   Smoking  TIME  DEATH_EVENT
0        0    97            0
1        1   180            0
2        1    31            1
3        0    87            0
4        0   113            0
```

```
data.shape
```

```
(299, 13)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       299 non-null    float64
 1   Gender                    299 non-null    int64
 2   Diabetes                  299 non-null    int64
 3   Anaemia                   299 non-null    int64
```

```
 4   Ejection_Fraction         299 non-null    int64
 5   High_Blood_Pressure       299 non-null    int64
 6   creatinine_phosphokinase  299 non-null    int64
 7   Pletelets                 299 non-null    float64
 8   Serum_Creatinine          299 non-null    float64
 9   Serum_Sodium              299 non-null    int64
 10  Smoking                   299 non-null    int64
 11  TIME                      299 non-null    int64
 12  DEATH_EVENT               299 non-null    int64
dtypes: float64(3), int64(10)
memory usage: 30.5 KB
```

```
data.describe()
```

```
              Age      Gender    Diabetes     Anaemia  Ejection_Fraction
\
count  299.000000  299.000000  299.000000  299.000000         299.000000
mean    60.833893    0.648829    0.418060    0.431438          38.083612
std     11.894809    0.478136    0.494067    0.496107          11.834841
min     40.000000    0.000000    0.000000    0.000000          14.000000
25%     51.000000    0.000000    0.000000    0.000000          30.000000
50%     60.000000    1.000000    0.000000    0.000000          38.000000
75%     70.000000    1.000000    1.000000    1.000000          45.000000
max     95.000000    1.000000    1.000000    1.000000          80.000000

       High_Blood_Pressure  creatinine_phosphokinase      Pletelets  \
count           299.000000                299.000000     299.000000
mean              0.351171                581.839465  263358.029264
std               0.478136                970.287881   97804.236869
min               0.000000                 23.000000   25100.000000
25%               0.000000                116.500000  212500.000000
50%               0.000000                250.000000  262000.000000
75%               1.000000                582.000000  303500.000000
max               1.000000               7861.000000  850000.000000

       Serum_Creatinine  Serum_Sodium    Smoking        TIME  DEATH_EVENT
count         299.00000    299.000000  299.00000  299.000000    299.00000
mean            1.39388    136.625418    0.32107  130.260870      0.32107
std             1.03451      4.412477    0.46767   77.614208      0.46767
min             0.50000    113.000000    0.00000    4.000000      0.00000
25%             0.90000    134.000000    0.00000   73.000000      0.00000
50%             1.10000    137.000000    0.00000  115.000000      0.00000
75%             1.40000    140.000000    1.00000  203.000000      1.00000
max             9.40000    148.000000    1.00000  285.000000      1.00000
```

```
data.DEATH_EVENT.value_counts()
```

```
0    203
1     96
Name: DEATH_EVENT, dtype: int64
```

```
count_class_0,count_class_1=data.DEATH_EVENT.value_counts()

data_count_0=data[data["DEATH_EVENT"]==0]
data_count_1=data[data["DEATH_EVENT"]==1]

#Oversampling method
data_count_over=data_count_1.sample(count_class_0,replace=True)

newdata=pd.concat([data_count_0,data_count_over],axis=0)
newdata
```

```
      Age  Gender  Diabetes  Anaemia  Ejection_Fraction
High_Blood_Pressure  \
0    43.0       0         0        1                 50
0
1    73.0       1         1        1                 30
0
3    65.0       1         0        1                 25
0
4    64.0       1         0        0                 60
0
6    70.0       1         0        0                 40
0
..    ...     ...       ...      ...                ...
...
105  85.0       1         0        0                 35
0
47   80.0       1         0        0                 38
0
184  58.0       1         0        1                 25
0
291  70.0       0         0        1                 25
1
237  70.0       0         0        0                 25
0

     creatinine_phosphokinase  Pletelets  Serum_Creatinine  Serum_Sodium
\
0                         358  237000.00              1.30           135
1                         231  160000.00              1.18           142
3                         305  298000.00              1.10           141
4                        1610  242000.00              1.00           137
6                         582   51000.00              2.70           136
..                        ...        ...               ...           ...
105                      5882  243000.00              1.00           132
47                        805  263358.03              1.10           134
184                       145  219000.00              1.20           137
291                       125  237000.00              1.00           140
237                       161  244000.00              1.20           142
```

```
     Smoking  TIME  DEATH_EVENT
0          0    97            0
1          1   180            0
3          0    87            0
4          0   113            0
6          1   250            0
..       ...   ...          ...
105        1    72            1
47         0   109            1
184        1   170            1
291        0    15            1
237        0    66            1

[406 rows x 13 columns]
```

```
newdata.shape
```

```
(406, 13)
```

```
newdata.DEATH_EVENT.value_counts()
```

```
1    203
0    203
Name: DEATH_EVENT, dtype: int64
```

```
x=newdata.iloc[:,0:-1]
y=newdata.iloc[:,-1]
```

```
x.shape
```

```
(406, 12)
```

```
y.shape
```

```
(406,)
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_st
ate=True)
```

```
x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

```
((324, 12), (324,), (82, 12), (82,))
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
clf2=RandomForestClassifier(n_estimators=100)
```

```
clf2.fit(x_train,y_train)
```

```
RandomForestClassifier()
```

```
predi=clf2.predict(x_test)
```

```
from sklearn.metrics import
accuracy_score,confusion_matrix,r2_score,precision_score,classification_r
eport,recall_score

confusion_matrix(y_test,predi)

array([[38,  0],
       [ 1, 43]], dtype=int64)

accuracy_score(y_test,predi)*100

98.78048780487805

print(classification_report(y_test,predi))

              precision    recall  f1-score   support

           0       0.97      1.00      0.99        38
           1       1.00      0.98      0.99        44

    accuracy                           0.99        82
   macro avg       0.99      0.99      0.99        82
weighted avg       0.99      0.99      0.99        82


precision_score(y_test,predi)

1.0

recall_score(y_test,predi)

0.9772727272727273

## cross validation

from sklearn.model_selection import cross_val_score

print(cross_val_score(clf2, x, y, cv=10,scoring ='accuracy').mean())

0.9285975609756096
```