NAME: PASALA NEELIMA

REGNO:24MDT1064

EXPERIMENT :01 & 02

Task : introduction and
datapreprocessing and
visualization.

```
array=[80,85,90,95,100,105,110,115,120,125]
print(array)
```

```
[80, 85, 90, 95, 100, 105, 110, 115, 120, 125]
```

```
a=[[1,2,3,4],
[5,6,7,8],
[9,10,11,12]]
print("a=",a)
print(a[0])
print(a[0][3])
```

```
a= [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
 [1, 2, 3, 4]
 4
```

```
import pandas as pd d = {'col1':[1,2,3,4,7,8,5],
'col2':[4,5,6,9,5,7,8],'col3':[7,8,12,1,11,4,5]} df = pd.DataFrame(data=d)

print(df)
```

```
   col1  col2  col3 0
  1     4     7
  1     2     5     8
  2     3     6    12
   3     4     9                                                           4
   4     7     5    115     8     7                                        4
   6     5     8                                                           4
```

```
shape = df.shape
print(shape)
```

```
(7, 3)
```

```
num_of_rows = len(df)
print(f"the number of rows is {num_of_rows}.")
```

```
the number of rows is 7.
```

```
shape = df.shape
num_of_rows=df.shape[0]
print(num_of_rows)
```

```
7
```

```
num_of_rows=df.index.size
print(num_of_rows)
```

```
7
```

```
num_coloums=len(df.columns)
print(num_coloums)
```

```
3
```

```
num_of_cols=df.shape[1]
print(num_of_cols)
```

```
3
```

```
import numpy as np matrix =
np.arange(100).reshape(10, 10)
print(matrix)
```

```
[[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]
 [20 21 22 23 24 25 26 27 28 29]
 [30 31 32 33 34 35 36 37 38 39]
 [40 41 42 43 44 45 46 47 48 49]
 [50 51 52 53 54 55 56 57 58 59]
 [60 61 62 63 64 65 66 67 68 69]
 [70 71 72 73 74 75 76 77 78 79]
 [80 81 82 83 84 85 86 87 88 89]
 [90 91 92 93 94 95 96 97 98 99]]
```

```
display(df.head())
```

	col1	col2	col3
0	1	4	7
1	2	5	8
2	3	6	12
3	4	9	1
4	7	5	11

```
display(df.head(3))
```

	col1	col2	col3
0	1	4	7
1	2	5	8
2	3	6	12

```
print(list(df.columns))
```

```
['col1', 'col2', 'col3']
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
 RangeIndex: 7 entries, 0 to 6 Data
 columns  (total  3  columns):    #
 Column  Non-Null Count  Dtype
 ---  ------  --------------  -----
0   col1    7 non-null      int64
1   col2    7 non-null      int64 2   col3    7 non-null
    int64 dtypes: int64(3) memory usage: 296.0 bytes
```

```
df.mean()
```

	0
col1	4.285714
col2	6.285714
col3	6.857143

dtype: float64

```
df.median()
```

	0
col1	4.0
col2	6.0
col3	7.0

```
dtype: float64
```

```
df.var()
```

```
              0
col1   6.571429
col2   3.238095
col3  15.142857

dtype: float64
```

```
df.std()
```

```
             0
col1  2.563480 col2
      1.799471 col3
      3.891382

dtype: float64
```

```
df.min()
```

```
        0
col1  1 col2
      4

 col3   1

dtype: int64
```

```
df.max()
```

```
        0
col1 8

col2 9

col3 12

dtype: int64
```

```
df.describe()
```

	col1	col2	col3
count	7.000000	7.000000	7.000000
mean	4.285714	6.285714	6.857143
std	2.563480	1.799471	3.891382
min	1.000000	4.000000	1.000000
25%	2.500000	5.000000	4.500000
50%	4.000000	6.000000	7.000000
75%	6.000000	7.500000	9.500000
max	8.000000	9.000000	12.000000

```
df.isnull()
```

```
   col1  col2  col3
```

```
0   False   False   False
1   False   False   False
2   False   False   False
3   False   False   False
4   False   False   False
5   False   False   False
6   False   False   False
```

```
import numpy as np
import pandas as pd

d = {'ala':[1,2,3,4,np.nan,8,5], 'python':[4,5,6,np.nan,9,5,7], 'maths':[7,8,12,np.nan,1,11,4]}
df2 = pd.DataFrame(data=d) print(df2)
```

```
   ala  python  maths 0
 1.0     4.0    7.0
1  2.0     5.0    8.0
2  3.0     6.0   12.0
3  4.0     NaN    NaN
4  NaN     9.0    1.0
5  8.0     5.0   11.0
6  5.0     7.0    4.0
```

```
student_names=['neelima','ragav','harshi']
df.index=Student_names print(df)
```

```
---------------------------------------------------------------------------NameError
 Traceback (most recent call last)
<ipython-input-26-6698914c1ec8> in <cell line: 2>()
      1 student_names=['neelima','ragav','harshi']
----> 2 df.index=Student_names
      3 print(df)

NameError: name 'Student_names' is not defined
```

```
student_names = ['neelima', 'ragav', 'harshi'] + list(df.index[3:])
df.index = student_names print(df)
```

```
            col1   col2   col3
neelima      1      4      7
ragav        2      5      8
harshi       3      6     12
3          4      9      1
 4           7      5     11
 5           8      7      4
 6           5      8      5
```

```
missing_values=df2.isnull()
print(missing_values)
print("count total NaN at each column in dataframe:\n",df2.isnull().sum())
print("count total NaN in a dataframe")
```

```
    ala  python  maths 0
False   False  False
1   False   False  False
2   False   False  False
3   False    True   True
4   True   False  False
5   False   False  False6  False   False  False count total NaN at each column in
    dataframe:  ala       1 python     1 maths     1 dtype: int64
count total NaN in a dataframe
```

```
Start coding or generate with AI.
```

lab2 07-1-2025 `Start coding or`

```
generate with AI.
```

NAME: PASALA NEELIMA

REGNO : 24MDT1064 SLOT:

LU5,U6

EXPERIMENT NO : 02 DATE:07-01-2025

```
import pandas as pd
d = {'mark1':[2,3,4,5,6,7,8],'mark2':[4,5,6,7,8,9,10],'mark3':[6,7,8,9,10,11,12]}
df = pd.DataFrame(data=d) print(df)
```

```
   mark1  mark2  mark3 0
   2      4      6
   1      3      5      7
   2      4      6      8
   3      5      7      9
   4      6      8     10
   5      7      9     11
   6      8     10     12
```

```
d = {'mark1':[2,3,4,5,6,7,8],'mark2':[4,5,6,7,8,9,10],'mark3':[6,7,8,9,10,11,12]}
df = pd.DataFrame(data=d) print(df[['mark2', 'mark3']])
```

```
   mark2  mark3 0
   4      6
   1      5      7
   2      6      8
   3      7      9
   4      8     10
   5      9     11
   6      10     12
```

```
import pandas as pd import
scipy import numpy as np
import seaborn as sns import
matplotlib.pyplot as plt
```

```
df = pd.read_csv("/content/diabetes.csv")
print(df.head())
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

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

```
                          0
```

Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

dtype: int64

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

	0
Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

dtype: int64

```
df.describe()
```

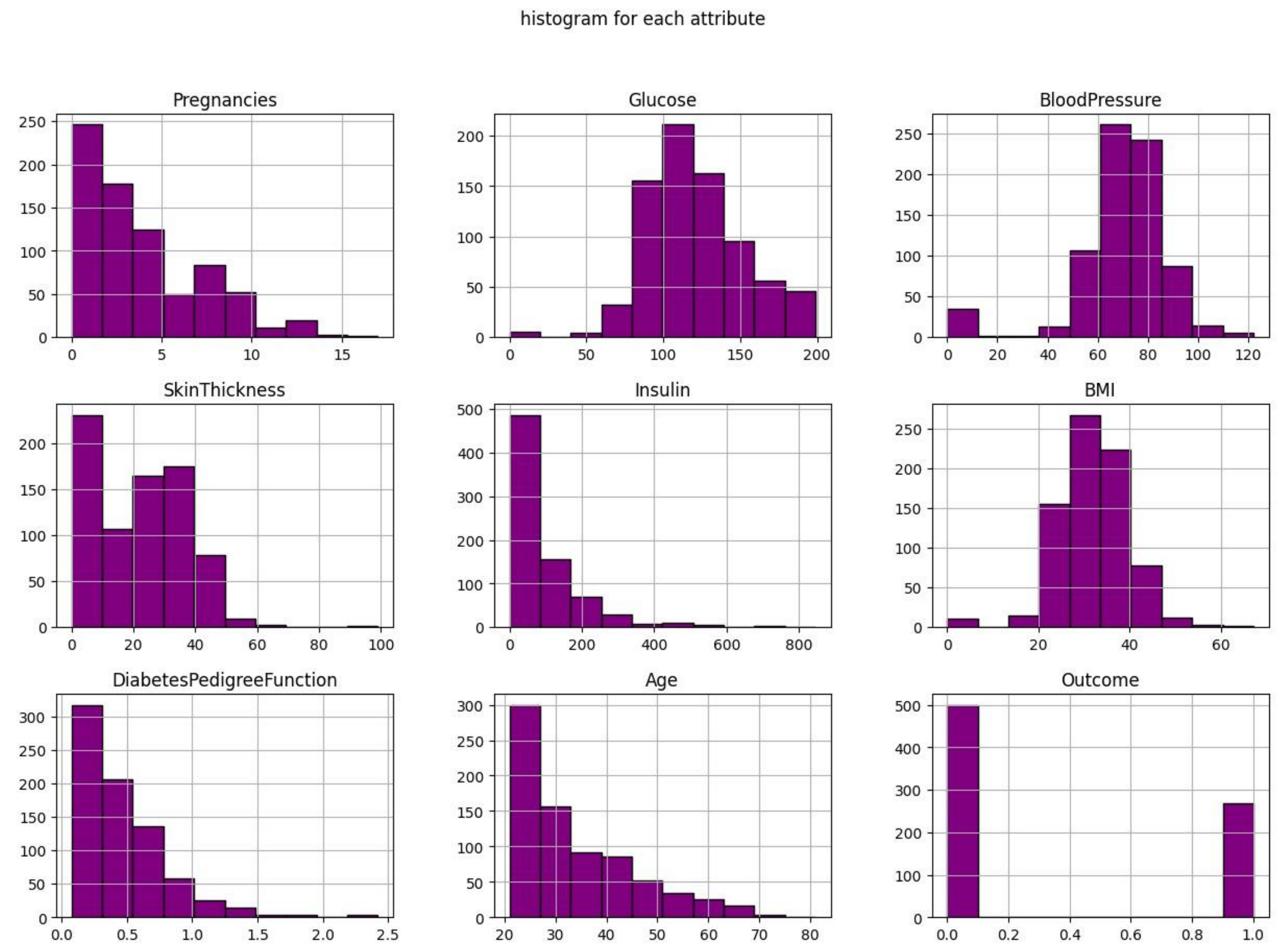
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00

```
df.hist(bins=10Glucose ,figsize=(15,10)768.0,color=120.894531'purple',edgecolor=31.972618'black'0.000)
99.00000 117.0000 140.25000 199.00 plt.suptitle("histogram for each attribute")
```

```
plt.show() BloodPressure    768.0    69.105469    19.355807    0.000    62.00000    72.0000    80.00000    122.00
```


histogram for each attribute
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome