In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
dataset=pd.read_csv("Placement.csv")
```

In [3]:

```
dataset
```

Out[3]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No
...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No

215 rows × 15 columns

In [4]:

```
dataset.columns
```

Out[4]:

```
Index(['sl_no', 'gender', 'ssc_p', 'ssc_b', 'hsc_p', 'hsc_b', 'hsc_s',
       'degree_p', 'degree_t', 'workex', 'etest_p', 'specialisation', 'mba_
p',
       'status', 'salary'],
      dtype='object')
```

In [5]:

```
dataset["degree_p"].dtype
```

Out[5]:

```
dtype('float64')
```

In [6]:

```
quan=[]
qual=[]
for C_N in dataset.columns:
    if(dataset[C_N].dtype=="O"):
        qual.append(C_N)
    else:
        quan.append(C_N)
```

In [7]:

```
quan
```

Out[7]:

```
['sl_no', 'ssc_p', 'hsc_p', 'degree_p', 'etest_p', 'mba_p', 'salary']
```

In [8]:

```
qual
```

Out[8]:

```
['gender',
 'ssc_b',
 'hsc_b',
 'hsc_s',
 'degree_t',
 'workex',
 'specialisation',
 'status']
```

In [9]:

```
from UniAnalysis import Univariate
```

In [10]:

```
obj=Univariate()
```

In [11]:

```
Qual,Quan=obj.QuanQual(dataset)
```

In [12]:

```
Qual
```

Out[12]:

```
['gender',
 'ssc_b',
 'hsc_b',
 'hsc_s',
 'degree_t',
 'workex',
 'specialisation',
 'status']
```

In [13]:

```
Quan
```

Out[13]:

```
['sl_no', 'ssc_p', 'hsc_p', 'degree_p', 'etest_p', 'mba_p', 'salary']
```

In [14]:

```
dataset["degree_p"].mean()
```

Out[14]:

```
66.37018604651163
```

In [15]:

```
dataset["degree_p"].median()
```

Out[15]:

```
66.0
```

In [16]:

```
dataset["degree_p"].mode()
```

Out[16]:

```
0    65.0
dtype: float64
```

In [17]:

```
dataset["degree_p"].mode()[0]
```

Out[17]:

```
65.0
```

In [55]:

```
","IQR","IQR*1.5","Lesser Boundary","Greater Boundary","Min value","Max value","Outlier"])
```

In [56]:

```
UniTable
```

Out[56]:

	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
Mean	NaN	NaN	NaN	NaN	NaN	NaN
Median	NaN	NaN	NaN	NaN	NaN	NaN
Mode	NaN	NaN	NaN	NaN	NaN	NaN
25th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
50th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
75th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
90th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
99th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
100th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
IQR	NaN	NaN	NaN	NaN	NaN	NaN
IQR*1.5	NaN	NaN	NaN	NaN	NaN	NaN
Lesser Boundary	NaN	NaN	NaN	NaN	NaN	NaN
Greater Boundary	NaN	NaN	NaN	NaN	NaN	NaN
Min value	NaN	NaN	NaN	NaN	NaN	NaN
Max value	NaN	NaN	NaN	NaN	NaN	NaN
Outlier	NaN	NaN	NaN	NaN	NaN	NaN

In [57]:

```
for C_N in Quan[1:]:
    UniTable[C_N]["Mean"]=dataset[C_N].mean()
```

In [58]:

```
UniTable
```

Out[58]:

	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
Mean	67.3034	66.3332	66.3702	72.1006	62.2782	288655
Median	NaN	NaN	NaN	NaN	NaN	NaN
Mode	NaN	NaN	NaN	NaN	NaN	NaN
25th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
50th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
75th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
90th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
99th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
100th-Percentile	NaN	NaN	NaN	NaN	NaN	NaN
IQR	NaN	NaN	NaN	NaN	NaN	NaN
IQR*1.5	NaN	NaN	NaN	NaN	NaN	NaN
Lesser Boundary	NaN	NaN	NaN	NaN	NaN	NaN
Greater Boundary	NaN	NaN	NaN	NaN	NaN	NaN
Min value	NaN	NaN	NaN	NaN	NaN	NaN
Max value	NaN	NaN	NaN	NaN	NaN	NaN
Outlier	NaN	NaN	NaN	NaN	NaN	NaN

In [59]:

```
for C_N in Quan[1:]:
    UniTable[C_N]["Mean"]=dataset[C_N].mean()
    UniTable[C_N]["Median"]=dataset[C_N].median()
    UniTable[C_N]["Mode"]=dataset[C_N].mode()
```

In [81]:

```
dataset["salary"]=dataset["salary"].fillna(0)
```

In [82]:

```
UniTable
```

Out[82]:

	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
Mean	67.3034	66.3332	66.3702	72.1006	62.2782	198702
Median	67	65	66	71	62	240000
Mode	62	63	65	60	56.7	0
25th-Percentile	60.6	60.9	61	60	57.945	0
50th-Percentile	67	65	66	71	62	240000
75th-Percentile	75.7	73	72	83.5	66.255	282500
90th-Percentile	82	79	77.12	92	70.678	356000
99th-Percentile	87	91.86	83.86	97	76.1142	629000
100th-Percentile	89.4	97.7	91	98	77.89	940000
IQR	15.1	12.1	11	23.5	8.31	282500
IQR*1.5	22.65	18.15	16.5	35.25	12.465	423750
Lesser Boundary	37.95	42.75	44.5	24.75	45.48	-423750
Greater Boundary	98.35	91.15	88.5	118.75	78.72	706250
Min value	40.89	37	50	50	51.21	0
Max value	89.4	97.7	91	98	77.89	940000
Outlier	Not Present	Present	Present	Not Present	Not Present	Present

In [83]:

```
for C_N in Quan[1:]:
    UniTable[C_N]["Mean"]=dataset[C_N].mean()
    UniTable[C_N]["Median"]=dataset[C_N].median()
    UniTable[C_N]["Mode"]=dataset[C_N].mode()[0]
    UniTable[C_N]["25th-Percentile"]=np.percentile(dataset[C_N],25)
    UniTable[C_N]["50th-Percentile"]=np.percentile(dataset[C_N],50)
    UniTable[C_N]["75th-Percentile"]=np.percentile(dataset[C_N],75)
    UniTable[C_N]["90th-Percentile"]=np.percentile(dataset[C_N],90)
    UniTable[C_N]["99th-Percentile"]=np.percentile(dataset[C_N],99)
    UniTable[C_N]["100th-Percentile"]=np.percentile(dataset[C_N],100)
    UniTable[C_N]["IQR"]=UniTable[C_N]["75th-Percentile"]-UniTable[C_N]["25th-Percentile"]
    UniTable[C_N]["IQR*1.5"]=(UniTable[C_N]["IQR"])*1.5
    UniTable[C_N]["Lesser Boundary"]=UniTable[C_N]["25th-Percentile"]-UniTable[C_N]["IQR*1.
    UniTable[C_N]["Greater Boundary"]=UniTable[C_N]["75th-Percentile"]+UniTable[C_N]["IQR*1
    UniTable[C_N]["Min value"]=dataset[C_N].min()
    UniTable[C_N]["Max value"]=dataset[C_N].max()
    if(UniTable[C_N]["Lesser Boundary"]>UniTable[C_N]["Min value"] or UniTable[C_N]["Greate
        UniTable[C_N]["Outlier"]="Present"
    else:
        UniTable[C_N]["Outlier"]="Not Present"
```

In [84]:

```
UniTable
```

Out[84]:

	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
Mean	67.3034	66.3332	66.3702	72.1006	62.2782	198702
Median	67	65	66	71	62	240000
Mode	62	63	65	60	56.7	0
25th-Percentile	60.6	60.9	61	60	57.945	0
50th-Percentile	67	65	66	71	62	240000
75th-Percentile	75.7	73	72	83.5	66.255	282500
90th-Percentile	82	79	77.12	92	70.678	356000
99th-Percentile	87	91.86	83.86	97	76.1142	629000
100th-Percentile	89.4	97.7	91	98	77.89	940000
IQR	15.1	12.1	11	23.5	8.31	282500
IQR*1.5	22.65	18.15	16.5	35.25	12.465	423750
Lesser Boundary	37.95	42.75	44.5	24.75	45.48	-423750
Greater Boundary	98.35	91.15	88.5	118.75	78.72	706250
Min value	40.89	37	50	50	51.21	0
Max value	89.4	97.7	91	98	77.89	940000
Outlier	Not Present	Present	Present	Not Present	Not Present	Present

In [85]:

```
#here outlier Present does not make any big difference so no worries about Outliers row
np.percentile(dataset["degree_p"],25)
```

Out[85]:

```
61.0
```

In [95]:

```
dataset["ssc_p"].value_counts().sort_index()
```

Out[95]:

```
40.89    1
41.00    1
43.00    1
44.00    1
45.00    1
        ..
85.80    1
86.50    1
87.00    3
88.00    1
89.40    1
Name: ssc_p, Length: 103, dtype: int64
```

In [87]:

```
FreqTable=pd.DataFrame(columns=["UniqueValues","Frequency","Relative-Freq","Cumulative-Freq
```

In [88]:

```
FreqTable
```

Out[88]:

	UniqueValues	Frequency	Relative-Freq	Cumulative-Freq

In [108]:

```
FreqTable["UniqueValues"]=dataset["ssc_p"].value_counts().sort_index().index
FreqTable["Frequency"]=dataset["ssc_p"].value_counts().sort_index().values
FreqTable["Relative-Freq"]=(FreqTable["Frequency"]/(len(FreqTable)))*100
FreqTable["Cumulative-Freq"]=FreqTable["Relative-Freq"].cumsum()
```

In [109]:

```
FreqTable
```

Out[109]:

	UniqueValues	Frequency	Relative-Freq	Cumulative-Freq
0	40.89	1	0.970874	0.970874
1	41.00	1	0.970874	1.941748
2	43.00	1	0.970874	2.912621
3	44.00	1	0.970874	3.883495
4	45.00	1	0.970874	4.854369
...	...	...	...	...
98	85.80	1	0.970874	202.912621
99	86.50	1	0.970874	203.883495
100	87.00	3	2.912621	206.796117
101	88.00	1	0.970874	207.766990
102	89.40	1	0.970874	208.737864

103 rows × 4 columns

In [101]:

```
len(FreqTable)
```

Out[101]:

```
103
```

In [ ]:

```

```