Name :SHEIK PAREETH
Differentiate Quan and Qual using univariate

In [1]:

```
#import variable
import pandas as pd
```

In [2]:

```
#import and read the dataet in csv file
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 information like how many columns and rows are there
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
sl_no             215 non-null int64
gender            215 non-null object
ssc_p             215 non-null float64
ssc_b             215 non-null object
hsc_p             215 non-null float64
hsc_b             215 non-null object
hsc_s             215 non-null object
degree_p          215 non-null float64
degree_t          215 non-null object
workex            215 non-null object
etest_p           215 non-null float64
specialisation    215 non-null object
mba_p             215 non-null float64
status            215 non-null object
salary            148 non-null float64
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

In [5]:

```
#This is univariate so we take one column values (ssc_b)
dataset["ssc_b"].dtypes
```

Out[5]:

```
dtype('O')
```

In [6]:

```
#seggregate all column names only
dataset.columns
```

Out[6]:

```
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 [7]:

```
for columnName in dataset.columns:
    print(columnName)
```

```
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
```

In [8]:

```
#using for loop method for print Quan and Qual
for columnName in dataset.columns:
    print(columnName)
    if (dataset[columnName].dtypes=="O"):
        print("Qual")
    else:
        print("Quan")
```

```
sl_no
Quan
gender
Qual
ssc_p
Quan
ssc_b
Qual
hsc_p
Quan
hsc_b
Qual
hsc_s
Qual
degree_p
Quan
degree_t
Qual
workex
Qual
etest_p
Quan
specialisation
Qual
mba_p
Quan
status
Qual
salary
Quan
```

In [9]:

```
Quan=[]
```

In [10]:

```
Quan
```

Out[10]:

```
[]
```

In [11]:

```
Qual=[]
```

In [12]:

```
Qual
```

Out[12]:

```
[]
```

In [13]:

```
#imprt append because this is create list
for columnName in dataset.columns:
    print(columnName)
    if (dataset[columnName].dtypes=="O"):
        #print("Qual")
        Qual.append(columnName)
    else:
        #print("Quan")
         Quan.append(columnName)
```

```
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
```

In [14]:

```
#quan list
Quan
```

Out[14]:

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

In [15]:

```
#qual list
Qual
```

Out[15]:

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

In [16]:

```
#create function using return statement
def QuanQual():
    Quan=[]
    Qual=[]
    for columnName in dataset.columns:
        print(columnName)
        if (dataset[columnName].dtypes=="O"):
            #print("Qual")
            Qual.append(columnName)
        else:
            #print("Quan")
             Quan.append(columnName)
    return Quan,Qual
```

In [17]:

```
Quan,Qual=QuanQual()
```

```
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
```

In [18]:

```
Quan
```

Out[18]:

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

In [19]:

```
Qual
```

Out[19]:

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