In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
dataset=pd.read_csv("Placement.csv")
dataset
```

Out[2]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | M | 67.00 | Others | 91.00 | Others | Commerce | 58.00 | Sci&Tech | No | 55.0 | Mkt&HR | 58.80 | Placed | 270000.0 |
| **1** | 2 | M | 79.33 | Central | 78.33 | Others | Science | 77.48 | Sci&Tech | Yes | 86.5 | Mkt&Fin | 66.28 | Placed | 200000.0 |
| **2** | 3 | M | 65.00 | Central | 68.00 | Central | Arts | 64.00 | Comm&Mgmt | No | 75.0 | Mkt&Fin | 57.80 | Placed | 250000.0 |
| **3** | 4 | M | 56.00 | Central | 52.00 | Central | Science | 52.00 | Sci&Tech | No | 66.0 | Mkt&HR | 59.43 | Not Placed | NaN |
| **4** | 5 | M | 85.80 | Central | 73.60 | Central | Commerce | 73.30 | Comm&Mgmt | No | 96.8 | Mkt&Fin | 55.50 | Placed | 425000.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **210** | 211 | M | 80.60 | Others | 82.00 | Others | Commerce | 77.60 | Comm&Mgmt | No | 91.0 | Mkt&Fin | 74.49 | Placed | 400000.0 |
| **211** | 212 | M | 58.00 | Others | 60.00 | Others | Science | 72.00 | Sci&Tech | No | 74.0 | Mkt&Fin | 53.62 | Placed | 275000.0 |
| **212** | 213 | M | 67.00 | Others | 67.00 | Others | Commerce | 73.00 | Comm&Mgmt | Yes | 59.0 | Mkt&Fin | 69.72 | Placed | 295000.0 |
| **213** | 214 | F | 74.00 | Others | 66.00 | Others | Commerce | 58.00 | Comm&Mgmt | No | 70.0 | Mkt&HR | 60.23 | Placed | 204000.0 |
| **214** | 215 | M | 62.00 | Central | 58.00 | Others | Science | 53.00 | Comm&Mgmt | No | 89.0 | Mkt&HR | 60.22 | Not Placed | NaN |

215 rows × 15 columns

In [3]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sl_no           215 non-null    int64  
 1   gender          215 non-null    object 
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object 
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object 
 6   hsc_s           215 non-null    object 
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object 
 9   workex          215 non-null    object 
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object 
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object 
 14  salary          148 non-null    float64
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

In [4]:
```
dataset.isnull().sum().T
#print(a,end='')
```

Out[4]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary           67
dtype: int64
```

In [5]:
```
dataset["salary"].fillna(0,inplace=True)
dataset
```

Out[5]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | M | 67.00 | Others | 91.00 | Others | Commerce | 58.00 | Sci&Tech | No | 55.0 | Mkt&HR | 58.80 | Placed | 270000.0 |
| **1** | 2 | M | 79.33 | Central | 78.33 | Others | Science | 77.48 | Sci&Tech | Yes | 86.5 | Mkt&Fin | 66.28 | Placed | 200000.0 |
| **2** | 3 | M | 65.00 | Central | 68.00 | Central | Arts | 64.00 | Comm&Mgmt | No | 75.0 | Mkt&Fin | 57.80 | Placed | 250000.0 |
| **3** | 4 | M | 56.00 | Central | 52.00 | Central | Science | 52.00 | Sci&Tech | No | 66.0 | Mkt&HR | 59.43 | Not Placed | 0.0 |
| **4** | 5 | M | 85.80 | Central | 73.60 | Central | Commerce | 73.30 | Comm&Mgmt | No | 96.8 | Mkt&Fin | 55.50 | Placed | 425000.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **210** | 211 | M | 80.60 | Others | 82.00 | Others | Commerce | 77.60 | Comm&Mgmt | No | 91.0 | Mkt&Fin | 74.49 | Placed | 400000.0 |
| **211** | 212 | M | 58.00 | Others | 60.00 | Others | Science | 72.00 | Sci&Tech | No | 74.0 | Mkt&Fin | 53.62 | Placed | 275000.0 |
| **212** | 213 | M | 67.00 | Others | 67.00 | Others | Commerce | 73.00 | Comm&Mgmt | Yes | 59.0 | Mkt&Fin | 69.72 | Placed | 295000.0 |
| **213** | 214 | F | 74.00 | Others | 66.00 | Others | Commerce | 58.00 | Comm&Mgmt | No | 70.0 | Mkt&HR | 60.23 | Placed | 204000.0 |
| **214** | 215 | M | 62.00 | Central | 58.00 | Others | Science | 53.00 | Comm&Mgmt | No | 89.0 | Mkt&HR | 60.22 | Not Placed | 0.0 |

215 rows × 15 columns

In [6]:
```
dataset.dropna(inplace=True)
dataset
```

Out[6]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | M | 67.00 | Others | 91.00 | Others | Commerce | 58.00 | Sci&Tech | No | 55.0 | Mkt&HR | 58.80 | Placed | 270000.0 |
| **1** | 2 | M | 79.33 | Central | 78.33 | Others | Science | 77.48 | Sci&Tech | Yes | 86.5 | Mkt&Fin | 66.28 | Placed | 200000.0 |
| **2** | 3 | M | 65.00 | Central | 68.00 | Central | Arts | 64.00 | Comm&Mgmt | No | 75.0 | Mkt&Fin | 57.80 | Placed | 250000.0 |
| **3** | 4 | M | 56.00 | Central | 52.00 | Central | Science | 52.00 | Sci&Tech | No | 66.0 | Mkt&HR | 59.43 | Not Placed | 0.0 |
| **4** | 5 | M | 85.80 | Central | 73.60 | Central | Commerce | 73.30 | Comm&Mgmt | No | 96.8 | Mkt&Fin | 55.50 | Placed | 425000.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **210** | 211 | M | 80.60 | Others | 82.00 | Others | Commerce | 77.60 | Comm&Mgmt | No | 91.0 | Mkt&Fin | 74.49 | Placed | 400000.0 |
| **211** | 212 | M | 58.00 | Others | 60.00 | Others | Science | 72.00 | Sci&Tech | No | 74.0 | Mkt&Fin | 53.62 | Placed | 275000.0 |
| **212** | 213 | M | 67.00 | Others | 67.00 | Others | Commerce | 73.00 | Comm&Mgmt | Yes | 59.0 | Mkt&Fin | 69.72 | Placed | 295000.0 |
| **213** | 214 | F | 74.00 | Others | 66.00 | Others | Commerce | 58.00 | Comm&Mgmt | No | 70.0 | Mkt&HR | 60.23 | Placed | 204000.0 |
| **214** | 215 | M | 62.00 | Central | 58.00 | Others | Science | 53.00 | Comm&Mgmt | No | 89.0 | Mkt&HR | 60.22 | Not Placed | 0.0 |

215 rows × 15 columns

In [7]:
```
dataset.isnull().sum()
```

Out[7]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary            0
dtype: int64
```

# Defining Quan & Qual for Outlier Removal

In [8]:
```
def quanQual(dataset):
    quan=[]
    qual=[]
    for columnName in dataset.columns:
        #print(columnName)
        if(dataset[columnName].dtype=='O'):
            #print("qual")
            qual.append(columnName)
        else:
            #print("quan")
            quan.append(columnName)
    return quan,qual
```

In [9]:
```
quan,qual=quanQual(dataset)
quan
```

Out[9]: ['sl_no', 'ssc_p', 'hsc_p', 'degree_p', 'etest_p', 'mba_p', 'salary']

In [10]:
```
dataset[qual]
```

Out[10]:

| | gender | ssc_b | hsc_b | hsc_s | degree_t | workex | specialisation | status |
|---|---|---|---|---|---|---|---|---|
| **0** | M | Others | Others | Commerce | Sci&Tech | No | Mkt&HR | Placed |
| **1** | M | Central | Others | Science | Sci&Tech | Yes | Mkt&Fin | Placed |
| **2** | M | Central | Central | Arts | Comm&Mgmt | No | Mkt&Fin | Placed |
| **3** | M | Central | Central | Science | Sci&Tech | No | Mkt&HR | Not Placed |
| **4** | M | Central | Central | Commerce | Comm&Mgmt | No | Mkt&Fin | Placed |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **210** | M | Others | Others | Commerce | Comm&Mgmt | No | Mkt&Fin | Placed |
| **211** | M | Others | Others | Science | Sci&Tech | No | Mkt&Fin | Placed |
| **212** | M | Others | Others | Commerce | Comm&Mgmt | Yes | Mkt&Fin | Placed |
| **213** | F | Others | Others | Commerce | Comm&Mgmt | No | Mkt&HR | Placed |
| **214** | M | Central | Others | Science | Comm&Mgmt | No | Mkt&HR | Not Placed |

215 rows × 8 columns

# Outlier Removal

In [11]:
```
descriptive=pd.DataFrame(index=["Mean","Median","Mode","Q1:25%","Q2:50%",
                                "Q3:75%","99%","Q4:100%","IQR","1.5rule","Lesser","Greater","Min","Max"])
for columnName in quan:
    descriptive[columnName]["Mean"]=dataset[columnName].mean()
    descriptive[columnName]["Median"]=dataset[columnName].median()
    descriptive[columnName]["Mode"]=dataset[columnName].mode()[0]
    descriptive[columnName]["Q1:25%"]=dataset.describe()[columnName]["25%"]
    descriptive[columnName]["Q2:50%"]=dataset.describe()[columnName]["50%"]
    descriptive[columnName]["Q3:75%"]=dataset.describe()[columnName]["75%"]
    descriptive[columnName]["99%"]=np.percentile(dataset[columnName],99)
    descriptive[columnName]["Q4:100%"]=dataset.describe()[columnName]["max"]
    descriptive[columnName]["IQR"]=descriptive[columnName]["Q3:75%"]-descriptive[columnName]["Q1:25%"]
    descriptive[columnName]["1.5rule"]=1.5*descriptive[columnName]["IQR"]
    descriptive[columnName]["Lesser"]=descriptive[columnName]["Q1:25%"]-descriptive[columnName]["1.5rul
    descriptive[columnName]["Greater"]=descriptive[columnName]["Q3:75%"]+descriptive[columnName]["1.5ru
    descriptive[columnName]["Min"]=dataset[columnName].min()
    descriptive[columnName]["Max"]=dataset[columnName].max()
```

In [12]:
```
lesser=[]
greater=[]

for columnName in quan:
    if(descriptive[columnName]["Lesser"]>descriptive[columnName]["Min"]):
        lesser.append(columnName)
    if(descriptive[columnName]["Greater"]<descriptive[columnName]["Q4:100%"]):
        greater.append(columnName)

```

In [13]:
```
lesser
```

Out[13]: ['hsc_p']

In [14]:
```
greater
```

Out[14]: ['hsc_p', 'degree_p', 'salary']

In [15]:
```
for column in lesser:
    dataset[column][dataset[column]<descriptive[column]["Lesser"]]=descriptive[column]["Lesser"]
for column in greater:
    dataset[column][dataset[column]>descriptive[column]["Greater"]]=descriptive[column]["Greater"]

```

```
/var/folders/07/ykgp85052b11h5kz22ghn8l40000gn/T/ipykernel_87072/3400726572.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  dataset[column][dataset[column]<descriptive[column]["Lesser"]]=descriptive[column]["Lesser"]
/var/folders/07/ykgp85052b11h5kz22ghn8l40000gn/T/ipykernel_87072/3400726572.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  dataset[column][dataset[column]>descriptive[column]["Greater"]]=descriptive[column]["Greater"]
```

In [16]:
```
descriptive=pd.DataFrame(index=["Mean","Median","Mode","Q1:25%","Q2:50%",
                                "Q3:75%","99%","Q4:100%","IQR","1.5rule","Lesser","Greater","Min","Max"])
for columnName in quan:
    descriptive[columnName]["Mean"]=dataset[columnName].mean()
    descriptive[columnName]["Median"]=dataset[columnName].median()
    descriptive[columnName]["Mode"]=dataset[columnName].mode()[0]
    descriptive[columnName]["Q1:25%"]=dataset.describe()[columnName]["25%"]
    descriptive[columnName]["Q2:50%"]=dataset.describe()[columnName]["50%"]
    descriptive[columnName]["Q3:75%"]=dataset.describe()[columnName]["75%"]
    descriptive[columnName]["99%"]=np.percentile(dataset[columnName],99)
    descriptive[columnName]["Q4:100%"]=dataset.describe()[columnName]["max"]
    descriptive[columnName]["IQR"]=descriptive[columnName]["Q3:75%"]-descriptive[columnName]["Q1:25%"]
    descriptive[columnName]["1.5rule"]=1.5*descriptive[columnName]["IQR"]
    descriptive[columnName]["Lesser"]=descriptive[columnName]["Q1:25%"]-descriptive[columnName]["1.5rul
    descriptive[columnName]["Greater"]=descriptive[columnName]["Q3:75%"]+descriptive[columnName]["1.5ru
    descriptive[columnName]["Min"]=dataset[columnName].min()
    descriptive[columnName]["Max"]=dataset[columnName].max()
```

In [17]:
```
descriptive
```

Out[17]:

| | sl_no | ssc_p | hsc_p | degree_p | etest_p | mba_p | salary |
|---|---|---|---|---|---|---|---|
| **Mean** | 108.0 | 67.303395 | 66.334744 | 66.358558 | 72.100558 | 62.278186 | 197615.116279 |
| **Median** | 108.0 | 67.0 | 65.0 | 66.0 | 71.0 | 62.0 | 240000.0 |
| **Mode** | 1 | 62.0 | 63.0 | 65.0 | 60.0 | 56.7 | 0.0 |
| **Q1:25%** | 54.5 | 60.6 | 60.9 | 61.0 | 60.0 | 57.945 | 0.0 |
| **Q2:50%** | 108.0 | 67.0 | 65.0 | 66.0 | 71.0 | 62.0 | 240000.0 |
| **Q3:75%** | 161.5 | 75.7 | 73.0 | 72.0 | 83.5 | 66.255 | 282500.0 |
| **99%** | 212.86 | 87.0 | 91.129 | 83.86 | 97.0 | 76.1142 | 629000.0 |
| **Q4:100%** | 215.0 | 89.4 | 91.15 | 88.5 | 98.0 | 77.89 | 706250.0 |
| **IQR** | 107.0 | 15.1 | 12.1 | 11.0 | 23.5 | 8.31 | 282500.0 |
| **1.5rule** | 160.5 | 22.65 | 18.15 | 16.5 | 35.25 | 12.465 | 423750.0 |
| **Lesser** | -106.0 | 37.95 | 42.75 | 44.5 | 24.75 | 45.48 | -423750.0 |
| **Greater** | 322.0 | 98.35 | 91.15 | 88.5 | 118.75 | 78.72 | 706250.0 |
| **Min** | 1 | 40.89 | 42.75 | 50.0 | 50.0 | 51.21 | 0.0 |
| **Max** | 215 | 89.4 | 91.15 | 88.5 | 98.0 | 77.89 | 706250.0 |

In [18]:
```
lesser=[]
greater=[]

for columnName in quan:
    if(descriptive[columnName]["Lesser"]>descriptive[columnName]["Min"]):
        lesser.append(columnName)
    if(descriptive[columnName]["Greater"]<descriptive[columnName]["Q4:100%"]):
        greater.append(columnName)
```

In [19]:
```
lesser
```

Out[19]: []

In [20]:
```
greater
```

Out[20]: []

In [21]:
```
dataset
```

Out[21]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | M | 67.00 | Others | 91.00 | Others | Commerce | 58.00 | Sci&Tech | No | 55.0 | Mkt&HR | 58.80 | Placed | 270000.0 |
| **1** | 2 | M | 79.33 | Central | 78.33 | Others | Science | 77.48 | Sci&Tech | Yes | 86.5 | Mkt&Fin | 66.28 | Placed | 200000.0 |
| **2** | 3 | M | 65.00 | Central | 68.00 | Central | Arts | 64.00 | Comm&Mgmt | No | 75.0 | Mkt&Fin | 57.80 | Placed | 250000.0 |
| **3** | 4 | M | 56.00 | Central | 52.00 | Central | Science | 52.00 | Sci&Tech | No | 66.0 | Mkt&HR | 59.43 | Not Placed | 0.0 |
| **4** | 5 | M | 85.80 | Central | 73.60 | Central | Commerce | 73.30 | Comm&Mgmt | No | 96.8 | Mkt&Fin | 55.50 | Placed | 425000.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **210** | 211 | M | 80.60 | Others | 82.00 | Others | Commerce | 77.60 | Comm&Mgmt | No | 91.0 | Mkt&Fin | 74.49 | Placed | 400000.0 |
| **211** | 212 | M | 58.00 | Others | 60.00 | Others | Science | 72.00 | Sci&Tech | No | 74.0 | Mkt&Fin | 53.62 | Placed | 275000.0 |
| **212** | 213 | M | 67.00 | Others | 67.00 | Others | Commerce | 73.00 | Comm&Mgmt | Yes | 59.0 | Mkt&Fin | 69.72 | Placed | 295000.0 |
| **213** | 214 | F | 74.00 | Others | 66.00 | Others | Commerce | 58.00 | Comm&Mgmt | No | 70.0 | Mkt&HR | 60.23 | Placed | 204000.0 |
| **214** | 215 | M | 62.00 | Central | 58.00 | Others | Science | 53.00 | Comm&Mgmt | No | 89.0 | Mkt&HR | 60.22 | Not Placed | 0.0 |

215 rows × 15 columns

## Saving the Preprocessed dataset as a New File

In [22]:
```
dataset.to_csv("PreProcessed_Placement.csv",index=False)
```

In [ ]:
```

```

## Alternative Codes for Learning

mode_value = dataset['rbc'].mode()[0] dataset['rbc'].fillna(mode_value, inplace=True) #dataset['pc'].fillna(modes, inplace=True)

mode_value = dataset['pc'].mode()[0] dataset['pc'].fillna(mode_value, inplace=True) #dataset['pc'].fillna(modes, inplace=True)