

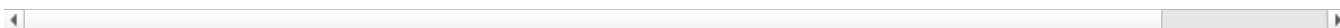
```
In [1]: import pandas as pd
```

```
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	st
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	PI
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	PI
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	PI
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	PI
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	PI
...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	PI
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	PI
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	PI
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	PI
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	PI

215 rows × 15 columns



```
In [3]: dataset.isnull().sum()
```

```
Out[3]: 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 [4]: dataset["salary"].fillna(dataset["salary"].median(),inplace=True)
dataset
```

C:\Users\SowmiGanesh\AppData\Local\Temp\ipykernel_10052\3922000602.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
dataset["salary"].fillna(dataset["salary"].median(),inplace=True)
```

Out[4]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	st
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	PI
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	PI
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	PI
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	PI
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	PI
...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	PI
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	PI
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	PI
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	PI
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	PI

215 rows × 15 columns

```
In [5]: def quanQual(dataset):
        quan=[]
        qual=[]
        for columnName in dataset.columns:
            if (dataset[columnName].dtypes=='O'):
                qual.append(columnName)
            else:
                quan.append(columnName)
        return quan,qual
```

```
In [6]: quan,qual=quanQual(dataset)
```

```
In [7]: dataset[quan]
```

Out[7]:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
0	1	67.00	91.00	58.00	55.0	58.80	270000.0
1	2	79.33	78.33	77.48	86.5	66.28	200000.0
2	3	65.00	68.00	64.00	75.0	57.80	250000.0
3	4	56.00	52.00	52.00	66.0	59.43	265000.0
4	5	85.80	73.60	73.30	96.8	55.50	425000.0
...
210	211	80.60	82.00	77.60	91.0	74.49	400000.0
211	212	58.00	60.00	72.00	74.0	53.62	275000.0
212	213	67.00	67.00	73.00	59.0	69.72	295000.0
213	214	74.00	66.00	58.00	70.0	60.23	204000.0
214	215	62.00	58.00	53.00	89.0	60.22	265000.0

215 rows × 7 columns

```
In [8]: import numpy as np
        from sklearn.impute import SimpleImputer
        imp= SimpleImputer(missing_values=np.nan, strategy="median")
        imp.fit(dataset[quan])
```

Out[8]:

SimpleImputer

SimpleImputer(strategy='median')

```
In [11]: # because of outliers present in the quality dataset, replace the nan values using median
import numpy as np
from sklearn.impute import SimpleImputer
imp= SimpleImputer(missing_values=np.nan, strategy="median")
imp.fit(dataset[quan])
df=imp.transform(dataset[quan])
df=pd.DataFrame(df,columns=quan)
```

```
In [12]: df=pd.DataFrame(df,columns=quan)
```

```
df
```

```
Out[12]:
```

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
0	1.0	67.00	91.00	58.00	55.0	58.80	270000.0
1	2.0	79.33	78.33	77.48	86.5	66.28	200000.0
2	3.0	65.00	68.00	64.00	75.0	57.80	250000.0
3	4.0	56.00	52.00	52.00	66.0	59.43	265000.0
4	5.0	85.80	73.60	73.30	96.8	55.50	425000.0
...
210	211.0	80.60	82.00	77.60	91.0	74.49	400000.0
211	212.0	58.00	60.00	72.00	74.0	53.62	275000.0
212	213.0	67.00	67.00	73.00	59.0	69.72	295000.0
213	214.0	74.00	66.00	58.00	70.0	60.23	204000.0
214	215.0	62.00	58.00	53.00	89.0	60.22	265000.0

215 rows × 7 columns

```
In [14]: cate=dataset[qual]
```

```
In [15]: df
```

```
Out[15]:
```

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
0	1.0	67.00	91.00	58.00	55.0	58.80	270000.0
1	2.0	79.33	78.33	77.48	86.5	66.28	200000.0
2	3.0	65.00	68.00	64.00	75.0	57.80	250000.0
3	4.0	56.00	52.00	52.00	66.0	59.43	265000.0
4	5.0	85.80	73.60	73.30	96.8	55.50	425000.0
...
210	211.0	80.60	82.00	77.60	91.0	74.49	400000.0
211	212.0	58.00	60.00	72.00	74.0	53.62	275000.0
212	213.0	67.00	67.00	73.00	59.0	69.72	295000.0
213	214.0	74.00	66.00	58.00	70.0	60.23	204000.0
214	215.0	62.00	58.00	53.00	89.0	60.22	265000.0

215 rows × 7 columns

```
In [16]: cate
```

```
Out[16]:
```

	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

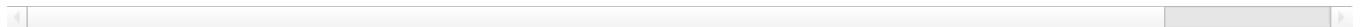
```
In [17]: two=[df,cate]
```

```
In [18]: preprocessed=pd.concat(two,axis=1)
```

```
In [19]: preprocessed
```

Out[19]:	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary	gender	ssc_b	hsc_b	hsc_s	degree_t	workex	special
0	1.0	67.00	91.00	58.00	55.0	58.80	270000.0	M	Others	Others	Commerce	Sci&Tech	No	1
1	2.0	79.33	78.33	77.48	86.5	66.28	200000.0	M	Central	Others	Science	Sci&Tech	Yes	1
2	3.0	65.00	68.00	64.00	75.0	57.80	250000.0	M	Central	Central	Arts	Comm&Mgmt	No	1
3	4.0	56.00	52.00	52.00	66.0	59.43	265000.0	M	Central	Central	Science	Sci&Tech	No	1
4	5.0	85.80	73.60	73.30	96.8	55.50	425000.0	M	Central	Central	Commerce	Comm&Mgmt	No	1
...
210	211.0	80.60	82.00	77.60	91.0	74.49	400000.0	M	Others	Others	Commerce	Comm&Mgmt	No	1
211	212.0	58.00	60.00	72.00	74.0	53.62	275000.0	M	Others	Others	Science	Sci&Tech	No	1
212	213.0	67.00	67.00	73.00	59.0	69.72	295000.0	M	Others	Others	Commerce	Comm&Mgmt	Yes	1
213	214.0	74.00	66.00	58.00	70.0	60.23	204000.0	F	Others	Others	Commerce	Comm&Mgmt	No	1
214	215.0	62.00	58.00	53.00	89.0	60.22	265000.0	M	Central	Others	Science	Comm&Mgmt	No	1

215 rows × 15 columns



```
In [21]: preprocessed.isnull().sum()
```

```
Out[21]: sl_no      0
         ssc_p      0
         hsc_p      0
         degree_p  0
         etest_p    0
         mba_p      0
         salary     0
         gender     0
         ssc_b      0
         hsc_b      0
         hsc_s      0
         degree_t   0
         workex     0
         specialisation 0
         status     0
         dtype: int64
```

```
In [22]: preprocessed.to_csv("Preplacementdata.csv",index=False)
         # after we preprocessed the data, save at csv file nan values cleared.
```

2) How many of them are not placed?

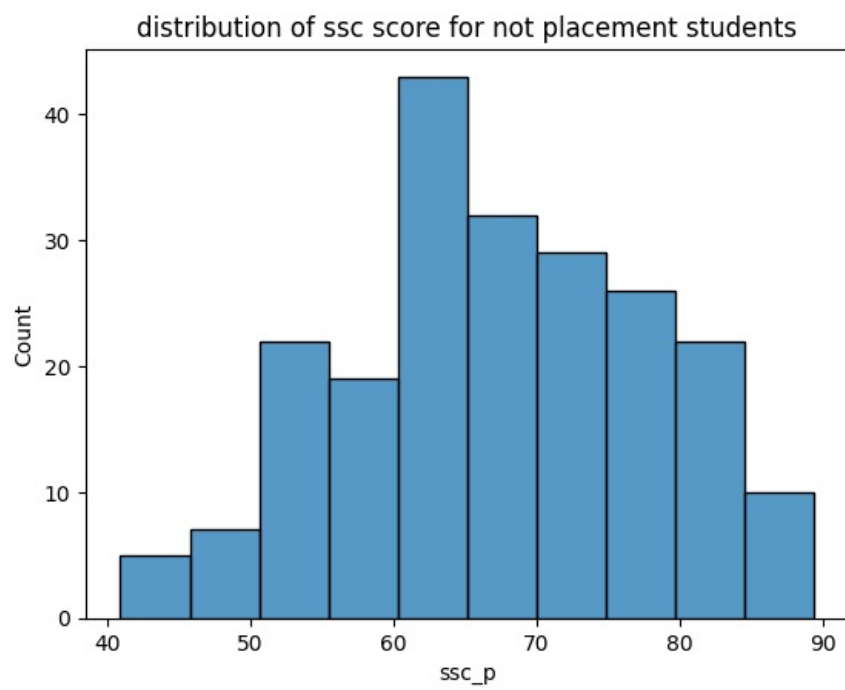
```
In [23]: dataset["status"].value_counts()
```

```
Out[23]: status
         Placed      148
         Not Placed    67
         Name: count, dtype: int64
```

67 of them are not placed

3) Find the reason for non placement from the dataset?

```
In [25]: import matplotlib.pyplot as plt
         import seaborn as sns
         sns.histplot(data=df,x="ssc_p")
         plt.title("distribution of ssc score for not placement students")
         plt.show()
```



using histograms the most of not placed students scored below average score.This is the one of the reason for not placed students.

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js