Assignment

1) Replace the NaN values with correct value.

```
dataset.isnull().sum()
```

```
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
```

```
df.fillna(0,inplace = True)
df
```

	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

- Replaced null values of salary with zero because the students are not placed yet so not getting salary.

2) How many of them are not placed?

```
: (dataset["status"]=="Not Placed").sum()
```

```
: 67
```

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

```
: from scipy.stats import ttest_ind, chi2_contingency

# Make a copy of dataset
df = dataset.copy()

# Identify target
target = "status"   # (values: "Placed", "Not Placed")

# Split groups
placed = df[df[target] == "Placed"]
not_placed = df[df[target] == "Not Placed"]

print("===== Numerical Features (t-test) =====")
for col in numerical_cols:
    t_stat, p_val = ttest_ind(placed[col], not_placed[col])
    result = "Significant" if p_val < 0.05 else "Not Significant"
    print(f"{col}: p={p_val:.4f} → {result}")

print("\n===== Categorical Features (Chi-square) =====")
for col in categorical_cols:
    contingency = pd.crosstab(df[col], df[target])
    chi2, p, dof, expected = chi2_contingency(contingency)
    result = "Significant" if p < 0.05 else "Not Significant"
    print(f"{col}: p={p:.4f} → {result}")
```

```
===== Numerical Features (t-test) =====
ssc_p: p=0.0000 → Significant
hsc_p: p=0.0000 → Significant
degree_p: p=0.0000 → Significant
etest_p: p=0.0617 → Not Significant
mba_p: p=0.2614 → Not Significant

===== Categorical Features (Chi-square) =====
gender: p=0.2398 → Not Significant
ssc_b: p=0.6898 → Not Significant
hsc_b: p=0.9223 → Not Significant
hsc_s: p=0.5727 → Not Significant
degree_t: p=0.2266 → Not Significant
workex: p=0.0001 → Significant
specialisation: p=0.0004 → Significant
```

From the above ttest, it is noted that 10th,12th,degree are significant and from Chi-square test, work-experience and specialisation are significant
Thus we can make a conclusion that low academic performance at school level and degree marks,no work experience & some specialisation might be the reason for non-placement.

4)What kind of relation between salary and mba_p

```
dataset.corr()
```

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
sl_no	1.000000	-0.078155	-0.085711	-0.088281	0.063636	0.022327	0.002543
ssc_p	-0.078155	1.000000	0.511472	0.538404	0.261993	0.388478	0.538090
hsc_p	-0.085711	0.511472	1.000000	0.434206	0.245113	0.354823	0.452569
degree_p	-0.088281	0.538404	0.434206	1.000000	0.224470	0.402364	0.408371
etest_p	0.063636	0.261993	0.245113	0.224470	1.000000	0.218055	0.186988
mba_p	0.022327	0.388478	0.354823	0.402364	0.218055	1.000000	0.139823
salary	0.002543	0.538090	0.452569	0.408371	0.186988	0.139823	1.000000

Correlation between Mba_p and salary is about 13%. Positive Correlation. When the Mba marks increases salary also increases by 13%.

5)Which specialization is getting minimum salary?

```
dataset.describe().astype(int)
```

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
count	215	215	215	215	215	215	215
mean	108	67	66	66	72	62	198702
std	62	10	10	7	13	5	154780
min	1	40	37	50	50	51	0
25%	54	60	60	61	60	57	0
50%	108	67	65	66	71	62	240000
75%	161	75	73	72	83	66	282500
max	215	89	97	91	98	77	940000

```
dataset.loc[dataset["salary"].idxmin(),"specialisation"]
```

```
'Mkt&HR'
```

Marketing and Finance got the Minimum Salary of 200000

6)How many of them getting above 500000 salary?

```
count = (dataset["salary"]>500000).sum()
count
```

```
3
```

Only 3 Persons are getting above salary of 500000

7)Test the Analysis of Variance between etest_p and mba_p at signifance level 5%.(Make decision using Hypothesis Testing)

H0 : There is no difference between etest pass mark and mba pass mark

H1 : There is a difference between etest pass mark and mba pass mark

```
import scipy.stats as stats
stats.f_oneway(dataset["etest_p"],dataset["mba_p"])
```

```
F_onewayResult(statistic=98.64487057324706, pvalue=4.672547689133573e-21)
```

```
dataset.cov()
```

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
sl_no	3870.000000	-52.641355	-58.106028	-40.413645	52.556168	8.102336	2.449065e+04
ssc_p	-52.641355	117.228377	60.348373	42.897137	37.659225	24.535952	9.017549e+05
hsc_p	-58.106028	60.348373	118.755706	34.819820	35.461678	22.555846	7.633598e+05
degree_p	-40.413645	42.897137	34.819820	54.151103	21.929469	17.272020	4.651315e+05
etest_p	52.556168	37.659225	35.461678	21.929469	176.251018	16.886973	3.842344e+05
mba_p	8.102336	24.535952	22.555846	17.272020	16.886973	34.028376	1.262455e+05
salary	24490.654206	901754.893936	763359.777657	465131.504238	384234.419257	126245.485547	2.395714e+10

pvalue < 0.05 so rejecting Null testing(H0) since there is a difference between the two marks of etest and MBA of about 16%. So Accepting Alternative testing (H1)

8)Test the similarity between the degree_t(Sci&Tech) and specialisation(Mkt&HR) with respect to salary at significance level of 5%.(Make decision using Hypothesis Testing)

```
H0: There is no significant differences in salary between sci&Tech degree and MKT&HR specialisation.
H1: There is a significant differences in salary between sci&Tech degree and MKT&HR specialisation.
```

```
from scipy.stats import ttest_ind
degree = dataset[dataset["degree_t"]=="Sci&Tech"]["salary"]
spec = dataset[dataset["specialisation"]=="Mkt&HR"]["salary"]
ttest_ind(degree,spec)
```

```
Ttest_indResult(statistic=2.692041243555374, pvalue=0.007897969943471179)
```

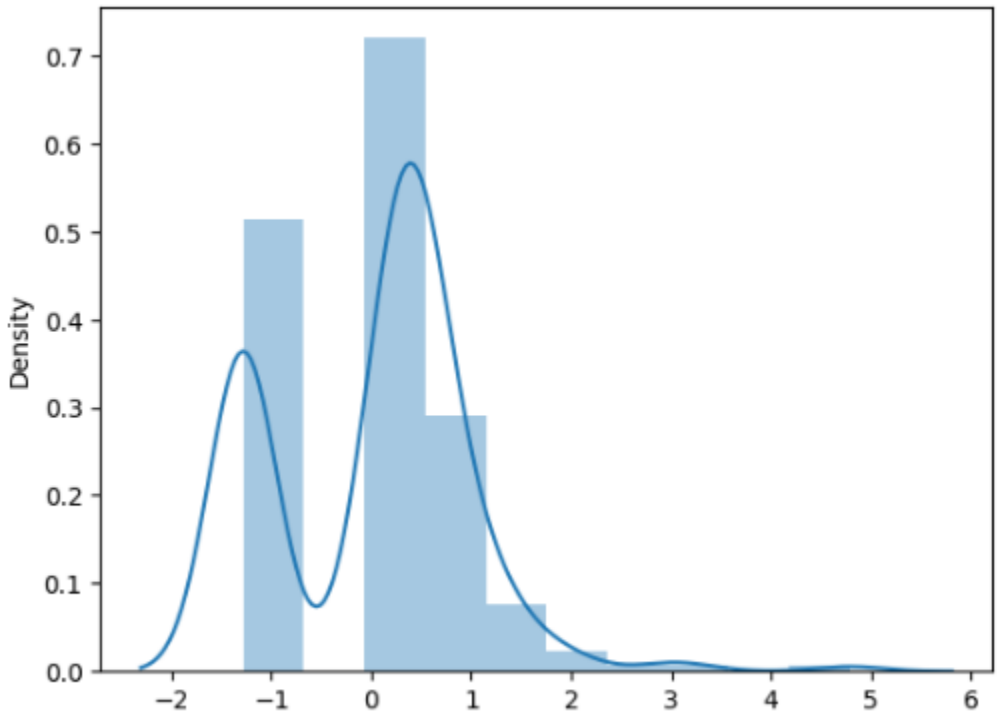
p-value<0.05 so rejecting Null Hypothesis and Accepting Alternate Hypothesis(H1). Thus there is a significant differences between the two groups

9)Convert the normal distribution to standard normal distribution for salary column

```
def SNDgraph(x):
    import seaborn as sns
    std = dataset["salary"].std()
    mean = dataset["salary"].mean()
    z_score = [((dataset["salary"] - mean)/std) ]
    print(z_score)
    sns.distplot(z_score,kde=True)
    sum(z_score)/len(z_score)
```

```
SNDgraph(dataset["salary"])
```

```
[0      0.460636
1      0.008384
2      0.331421
3     -1.283765
4      1.462051
         ...
210    1.300533
211    0.492940
212    0.622155
213    0.034227
214   -1.283765
Name: salary, Length: 215, dtype: float64]
```

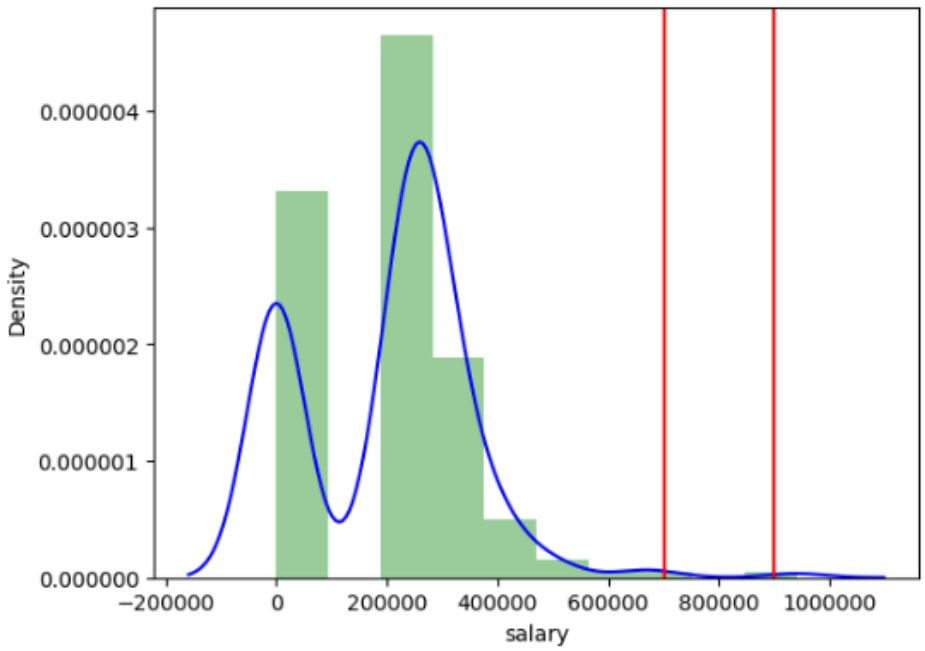


10)What is the probability Density Function of the salary range from 700000 to 900000?

```
def get_pdf(dataset,startrange,endrange):
    from matplotlib import pyplot
    from scipy.stats import norm
    import seaborn as sns
    ax = sns.distplot(dataset,kde = True,kde_kws = {'color':'blue'},color='Green')
    pyplot.axvline(startrange,color = 'Red')
    pyplot.axvline(endrange,color = 'Red')

    # generate a sample
    sample  = dataset

    # calculate parameters
    s_mean = sample.mean()
    s_std = sample.std()
    print('Mean = %.3f, Standard_deviation = %.3f' %(s_mean, s_std))

    # define the distribution
    dist = norm(s_mean,s_std)

    # probabilities sampling for startrange and endrange
    values = [value for value in range(startrange,endrange)]
    probabilities = [dist.pdf(value) for value in values]
    prob = sum(probabilities)
    print("The area between range ({},{}:{}".format(startrange,endrange,prob))
    return prob
```

```
get_pdf(dataset["salary"],700000,900000)
```

```
Mean = 198702.326, Standard_deviation = 154780.927
The area between range (700000,900000:0.0005973310593974868

0.0005973310593974868
```



11)Test the similarity between the degree_t(Sci&Tech)with respect to etest_p and mba_p at significance level of 5%.(Make decision using Hypothesis Testing)

```
Ho: There is no significant differences between sci&Tech degree of etest pass mark and mba pass mark
H1: There is a significant differences between sci&Tech degree of etest pass mark and mba pass mark
```

```
etest = dataset[dataset["degree_t"]=="Sci&Tech"]["etest_p"]
mba  = dataset[dataset["degree_t"]=="Sci&Tech"]["mba_p"]
from scipy.stats import ttest_ind
ttest_ind(etest,mba)
```

```
Ttest_indResult(statistic=4.532000225151251, pvalue=1.4289217003775636e-05)
```

```
pvalue<0.05 so rejecting null hypothesis and accepting Alternate hypothesis.
Thus we cant see any similarity for the sci&Tech degree of two marks.
```

12)Which parameter is highly correlated with salary?

```
dataset.drop("sl_no",axis = 1,inplace =True)
```

```
dataset.corr()
```

	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
ssc_p	1.000000	0.511472	0.538404	0.261993	0.388478	0.538090
hsc_p	0.511472	1.000000	0.434206	0.245113	0.354823	0.452569
degree_p	0.538404	0.434206	1.000000	0.224470	0.402364	0.408371
etest_p	0.261993	0.245113	0.224470	1.000000	0.218055	0.186988
mba_p	0.388478	0.354823	0.402364	0.218055	1.000000	0.139823
salary	0.538090	0.452569	0.408371	0.186988	0.139823	1.000000

10th mark is highly correlated with salary

13) plot any useful graph and explain it.

```
sns.pairplot(dataset)
```