

Inferential Data Analysis Answers

```
In [3]: import pandas as pd
```

```
In [4]: dataset = pd.read_csv("Placement.csv")
```

1) Replace the NaN values with correct value. And justify why you have chosen the same.

```
In [6]: dataset.isna().sum()
```

```
Out[6]: 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 [7]: dataset["salary"].fillna(0,inplace=True)
```

```
In [8]: dataset.isna().sum()
```

```
Out[8]: 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
```

2)How many of them are not placed?

```
In [10]: Not_Placed = dataset[dataset['status']=="Not Placed"]
No_of_Not_Placed = len(Not_Placed.index)
print(No_of_Not_Placed, "Students are Not Placed")
```

67 Students are Not Placed

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

```
In [12]: # The Students Those who scored less in ssc and hsc examinations haven't placed
```

4)What kind of relation between salary and mba_p

```
In [14]: quan = [col for col in dataset.columns if dataset[col].dtype != 'O']
```

```
In [15]: dataset[quan].corr()
```

```
Out[15]:
```

	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

```
In [16]: # Salary and MBA_p has low Degree Positive Correlation with a value of 0.13
```

5)Which specialization is getting minimum salary?

```
In [18]: Mkt_HR = dataset[dataset['specialisation']=="Mkt&HR"]['salary']
Mkt_Fin = dataset[dataset['specialisation']=="Mkt&Fin"]['salary']

Mkt_HR_Sal = Mkt_HR.mean()
Mkt_Fin_Sal = Mkt_Fin.mean()

if(Mkt_HR_Sal < Mkt_Fin_Sal):
    difference = Mkt_Fin_Sal - Mkt_HR_Sal
    print("Marketing and Finance gets higher salary than Marketing and HR wi
else:
    difference = Mkt_HR_Sal - Mkt_Fin_Sal
    print("Marketing and HR gets higher salary than Marketing and Finance wi
```

Marketing and Finance gets higher salary than Marketing and HR with ₹ 85749.56140350876

6)How many of them getting above 500000 salary?

```
In [20]: list = 0
for sal in dataset['salary']:
    if (sal < 500000):
        list+=1

print(list,"Of the Placed students getting the salary above 500000")
```

209 Of the Placed students getting the salary above 500000

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

```
In [22]: import scipy.stats as stats
stats.f_oneway(dataset['etest_p'],dataset['mba_p'])
print('There is a Significant Differeance between Enterance test and MBA Pas
```

There is a Significant Differeance between Enterance test and MBA Pass mark

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)

```
In [24]: Sci_Tech = dataset[dataset['degree_t']=="Sci&Tech"]['salary']
Mkt_HR = dataset[dataset['specialisation']=="Mkt&HR"]['salary']

from scipy import stats
print(stats.ttest_ind(Sci_Tech, Mkt_HR))

print("Accept Alternate Hypothesis since p value is less than 0.05. Threfore
```

TtestResult(statistic=2.692041243555374, pvalue=0.007897969943471179, df=152.0)

Accept Alternate Hypothesis since p value is less than 0.05. Threfore, there is a similarity between the degree_t(Sci&Tech) and specialisation(Mkt&HR) wi th respect to salary

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

```
In [26]: def stdNBgraph(dataset):  
        # Coverted to standard Normal Distribution  
        import seaborn as sns  
        mean=dataset.mean()  
        std=dataset.std()  
  
        values=[i for i in dataset]  
  
        z_score=[((j-mean)/std) for j in values]  
  
        sns.distplot(z_score,kde=True)  
  
        sum(z_score)/len(z_score)  
        #z_score.std()  
  
        stdNBgraph(dataset["salary"])
```

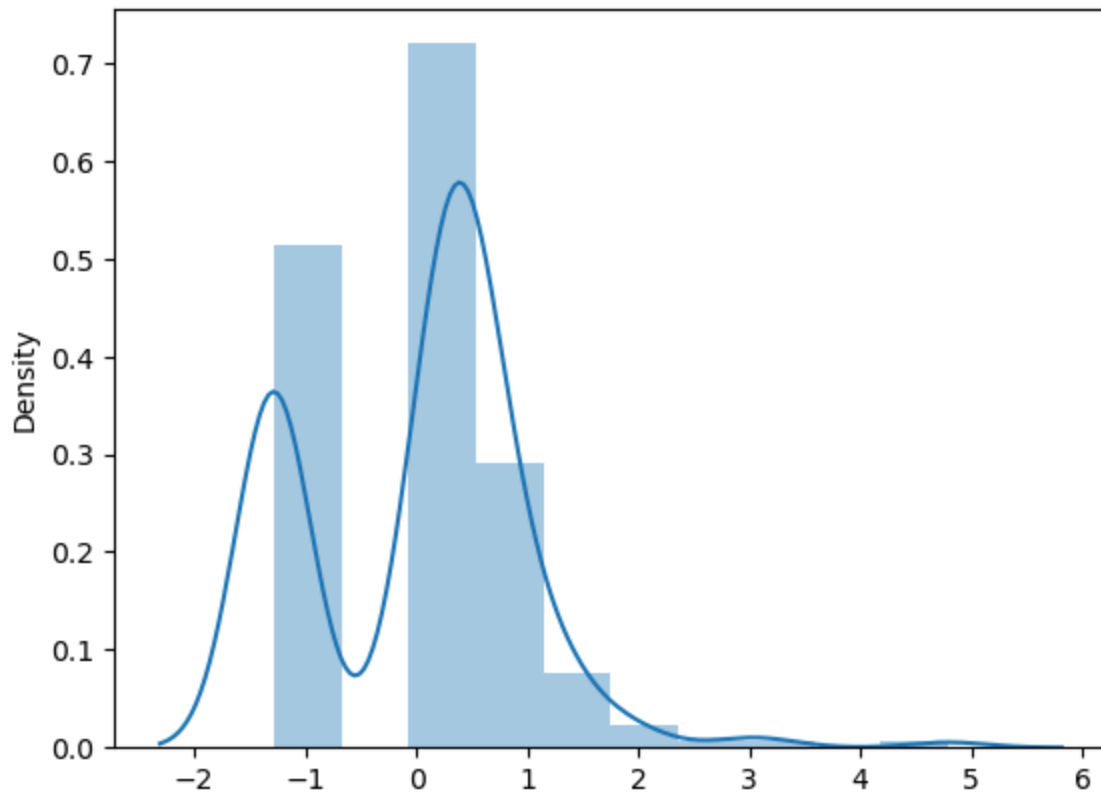
C:\Users\BANISH\AppData\Local\Temp\ipykernel_9572\3511215854.py:11: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(z_score,kde=True)  
C:\Users\BANISH\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: Future  
Warning: use_inf_as_na option is deprecated and will be removed in a future  
version. Convert inf values to NaN before operating instead.  
  with pd.option_context('mode.use_inf_as_na', True):
```



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

```
In [28]: def get_pdf_probability(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
    sample_mean = sample.mean()
    sample_std = sample.std()
    print('Mean=%.3f, Standard Deviation=%.3f' % (sample_mean, sample_std))
    # define the distribution
    dist = norm(sample_mean, sample_std)

    # sample probabilities for a range of outcomes
    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,sum(
    return prob

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

C:\Users\BANISH\AppData\Local\Temp\ipykernel_9572\1876602292.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

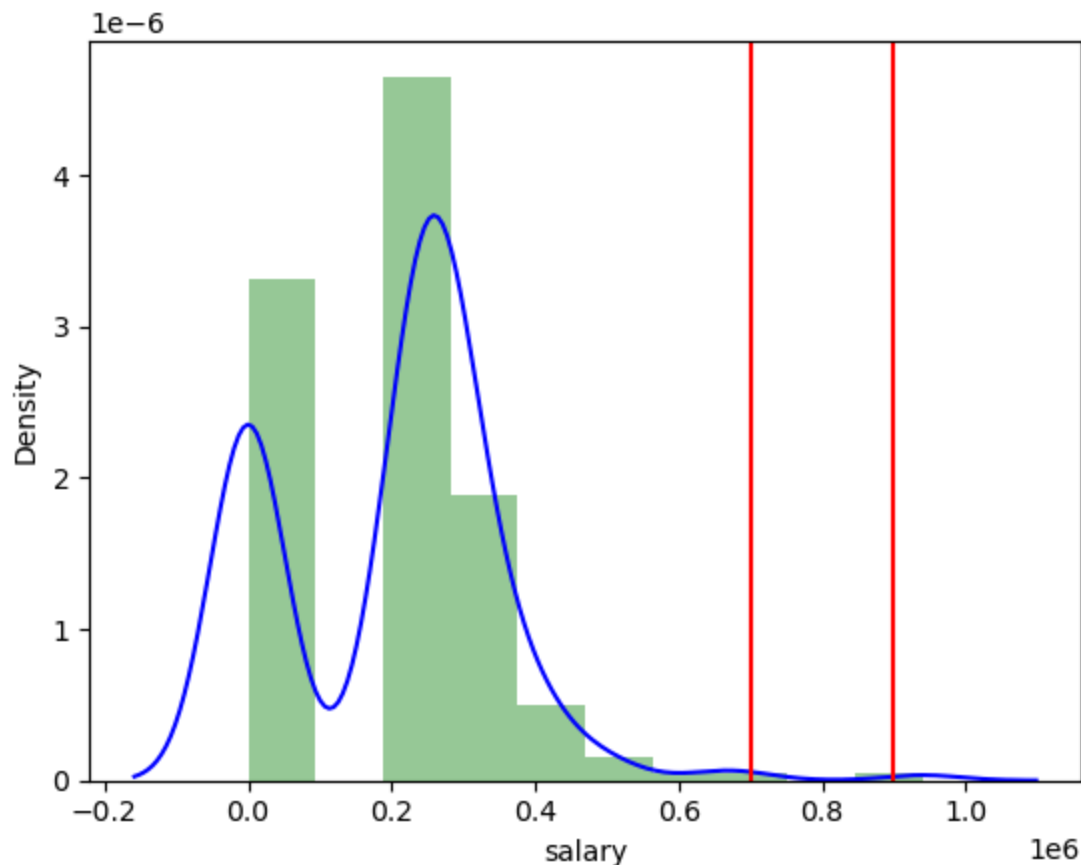
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
ax = sns.distplot(dataset,kde=True,kde_kws={'color':'blue'},color='Green')
C:\Users\BANISH\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: Future
Warning: use_inf_as_na option is deprecated and will be removed in a future
version. Convert inf values to NaN before operating instead.
```

```
with pd.option_context('mode.use_inf_as_na', True):
Mean=198702.326, Standard Deviation=154780.927
The area between range(700000,900000):0.0005973310593974901
```

Out[28]: 0.0005973310593974901



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)

```
In [30]: Etest = dataset[dataset['degree_t']=="Sci&Tech"]['etest_p']
MBA = dataset[dataset['degree_t']=="Sci&Tech"]['mba_p']

from scipy import stats
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js t, MBA))

```
print("Accept Null Hypothesis since p value is Greater than 0.05. Therefore,
```

```
TtestResult(statistic=4.532000225151251, pvalue=1.4289217003775636e-05, df=16.0)
```

Accept Null Hypothesis since p value is Greater than 0.05. Therefore, there is a significant Difference between the degree_t(Sci&Tech) with respect to etest_p and mba_p at significance level of 5%

12) Which parameter is highly correlated with salary?

```
In [32]: dataset[quan].corr()
```

```
Out[32]:
```

	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

```
In [33]: # SSC_P is Higly correlated with salary with 53%
```

13. plot any useful graph and explain it.

```
In [35]: import seaborn as sns
sns.distplot(dataset["ssc_p"])
```

C:\Users\BANISH\AppData\Local\Temp\ipykernel_9572\3471076516.py:2: UserWarning:

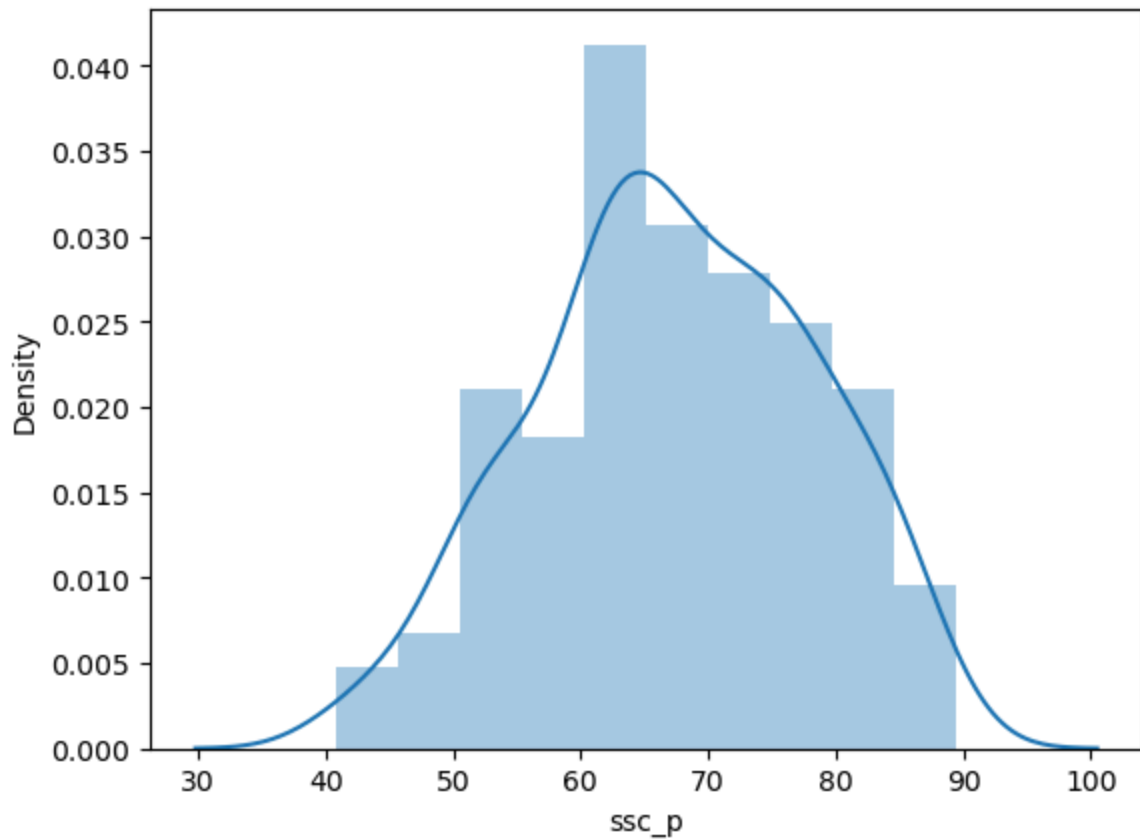
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(dataset["ssc_p"])
C:\Users\BANISH\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: Future
Warning: use_inf_as_na option is deprecated and will be removed in a future
version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
Out[35]: <Axes: xlabel='ssc_p', ylabel='Density'>
```



Name : ***Banish J***

Phone Number : +91 9444333914

E Mail : mail@banish.in

Website : <https://www.banish.in/>

Prepared by [Banish J](#) Mentored by [Ramisha Rani Mam](#)