Including Packages

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#importing all the packages
```

Importing Dataset

```
placementdataset =pd.read_csv("/content/Placement_Data_Full_Class.csv")
#importing dataset
```

```
placementdataset.shape
#215 rows and 15 columns
```

```
(215, 15)
```

```
placementdataset.info()
#information about the data
```

```
<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 
                        ll    float64
                        ll    object 
      _                 ll    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
```

Saved successfully! ×

```
placementdataset.head()
#first five entries
```

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	speci
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	

```
placementdataset.tail()
#last five rows
```

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	spe
210	211	M	80.6	Others	82.0	Others	Commerce	77.6	Comm&Mgmt	No	91.0	
211	212	M	58.0	Others	60.0	Others	Science	72.0	Sci&Tech	No	74.0	
212	213	M	67.0	Others	67.0	Others	Commerce	73.0	Comm&Mgmt	Yes	59.0	
213	214	F	74.0	Others	66.0	Others	Commerce	58.0	Comm&Mgmt	No	70.0	
214	215	M	62.0	Central	58.0	Others	Science	53.0	Comm&Mgmt	No	89.0	

```
print(placementdataset.describe(include='all'))
#
# **'include'** is the argument which is used to pass necessary information regarding what columns need to be considered for summarizing.
```

```
             sl_no gender       ssc_p    ssc_b       hsc_p    hsc_b     hsc_s  \
count   215.000000    215  215.000000      215  215.000000      215       215
unique         NaN      2         NaN        2         NaN        2         3
top            NaN      M         NaN  Central         NaN   Others  Commerce
freq           NaN    139         NaN      116         NaN      131       113
mean    108.000000    NaN   67.303395      NaN   66.333163      NaN       NaN
std      62.209324    NaN   10.827205      NaN   10.897509      NaN       NaN
min       1.000000    NaN   40.890000      NaN   37.000000      NaN       NaN
25%      54.500000    NaN   60.600000      NaN   60.900000      NaN       NaN
50%     108.000000    NaN   67.000000      NaN   65.000000      NaN       NaN
75%     161.500000    NaN   75.700000      NaN   73.000000      NaN       NaN
max     215.000000    NaN   89.400000      NaN   97.700000      NaN       NaN

          degree_p   degree_t workex     etest_p specialisation       mba_p  \
count   215.000000        215    215  215.000000            215  215.000000
unique         NaN          3      2         NaN              2         NaN
top            NaN  Comm&Mgmt     No         NaN        Mkt&Fin         NaN
freq           NaN        145    141         NaN            120         NaN
mean     66.370186        NaN    NaN   72.100558            NaN   62.278186
std       7.358743        NaN    NaN   13.275956            NaN    5.833385
min      50.000000        NaN    NaN   50.000000            NaN   51.210000
25%      61.000000        NaN    NaN   60.000000            NaN   57.945000
50%      66.000000        NaN    NaN   71.000000            NaN   62.000000
75%      72.000000        NaN    NaN   83.500000            NaN   66.255000
max      91.000000        NaN    NaN   98.000000            NaN   77.890000

        status         salary
count      215     148.000000
unique       2            NaN
top     Placed            NaN
freq       148            NaN
mean       NaN  288655.405405
std        NaN   93457.452420
min        NaN  200000.000000
25%        NaN  240000.000000
50%        NaN  265000.000000
75%        NaN  300000.000000
```

Saved successfully!

```
placementdataset['salary'].describe()
#shows min max mean etc
#The describe() method returns description of the data in the DataFrame.

#If the DataFrame contains numerical data, the description contains these information for each column:

#count - The number of not-empty values.
#mean - The average (mean) value.
#std - The standard deviation.
#min - the minimum value.
#25% - The 25% percentile*.
#50% - The 50% percentile*.
#75% - The 75% percentile*.
#max - the maximum value.

#*Percentile meaning: how many of the values are less than the given percentile
```

```
count       148.000000
mean     288655.405405
std       93457.452420
min      200000.000000
25%      240000.000000
50%      265000.000000
75%      300000.000000
max      940000.000000
Name: salary, dtype: float64
```

```
for cols in placementdataset.columns:
  print(cols)
  #displays all the column names
  #using iteration for doing this
```

```
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
```

```
print(placementdataset['workex'].unique())
print(placementdataset['specialisation'].unique())
print(placementdataset['status'].unique())
```

```
['No' 'Yes']
['Mkt&HR' 'Mkt&Fin']
['Placed' 'Not Placed']
```

```
placementdataset['salary'].max()
```

```
940000.0
```

Cleaning the dataset

```
placementdataset.isnull().sum()
#handling missing data
#Returns the sum of the values for the requested axis
#the salary column has null values for students that arent placed
#If you want to simply exclude the missing values, then use the dropna function along with the axis argument.
### **Check for Missing Values**
#To make detecting missing values easier (and across different array dtypes), Pandas provides the **isnull()** and **notnull()** functior
```

```
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
```

Saved successfully! ×

```
mba_p              0
status             0
salary            67
dtype: int64
```

```
placementdataset['salary'].fillna(value=0, inplace=True)
#replacing empty columns with value 0
#Pandas provides various methods for cleaning the missing values. The fillna function can “fill in” NA values with non-null data in a cou
```

```
placementdataset.isnull().sum()
#no more empty columns
```

```
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
```

```
placementdataset.groupby('degree_t')['status'].value_counts()
#syntax:obj.groupby('key')
#Which degree has how many placed and not placed students
#We are using groupby operation for cobining the results by performing aggregation i.e computing a summary statistic
```

```
degree_t   status
Comm&Mgmt  Placed        102
           Not Placed     43
Others     Not Placed      6
           Placed          5
Sci&Tech   Placed         41
           Not Placed     18
Name: status, dtype: int64
```

```
#Does Employability test percentage conducted by college matters?
groups = placementdataset.groupby(['status', pd.cut(placementdataset.etest_p, [40,60,80, 100])])
```

```
#display bin count grouped by team
groups.size().unstack()
```

etest_p	(40, 60]	(60, 80]	(80, 100]
status			
Not Placed	21	34	12
Placed	38	61	49

```
#Does High school percentage matters?
groups = placementdataset.groupby(['status', pd.cut(placementdataset.hsc_p, [10, 50, 70, 100])])

#display bin count grouped by team
groups.size().unstack()
#Inference
#Yes , high school mark matter we can see that student with (mark > 70) have 91% chance of getting placed
#similarly people with high school percentage less 50% have no chance of getting placed
```

hsc_p	(10, 50]	(50, 70]	(70, 100]
status			
Not Placed	16	45	6
Placed	0	84	64

```
placementdataset[(placementdataset['status']=='Placed')].sort_values(by = 'salary',ascending = False).reset_index().head(1)
#student that received highest salary
#There are two kinds of sorting available in Pandas. They are −

#* By label
#* By Actual Value
                                      ending parameter, the order of the sorting can be controlled.
                                       sorting by values.

#It accepts a '**by**' argument which will use the column name of the DataFrame with which the values are to be sorted.
```

Saved successfully!

	index	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	stat
0	119	120	M	60.8	Central	68.4	Central	Commerce	64.6	Comm&Mgmt	Yes	82.66	Mkt&Fin	64.34	Plac

```
placementdataset[(placementdataset['degree_t']=='Sci&Tech')&(placementdataset['status']=='Placed')].sort_values(by = 'salary',ascending =
#highest salary in science field
```

	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	
0	150	151	M	71.0	Central	58.66	Central	Science	58.0	Sci&Tech	Yes	56.0	Mkt&Fin	61.3	Placed	69

```
placementdataset[(placementdataset['degree_t']=='Comm&Mgmt')&(placementdataset['status']=='Placed')].sort_values(by = 'salary',ascending
#highest salary in commerce and management field
```

	index	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	stat
0	119	120	M	60.8	Central	68.4	Central	Commerce	64.6	Comm&Mgmt	Yes	82.66	Mkt&Fin	64.34	Plac

```
placementdataset[(placementdataset['salary']>placementdataset['salary'].mean())].sort_values(by = 'salary',ascending = False).reset_inde>
#students with salary more than the avg salary
```

	index	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	119	120	M	60.80	Central	68.40	Central	Commerce	64.60	Comm&Mgmt	Yes	82.66	Mkt&Fin	64.34	Pl

```
placementdataset[(placementdataset['degree_t']=='Comm&Mgmt')&(placementdataset['salary']>placementdataset['salary'].mean())].sort_values(
# students with salary more than avg salary Comm&Mgmt dep
```

	index	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	119	120	M	60.8	Central	68.4	Central	Commerce	64.6	Comm&Mgmt	Yes	82.66	Mkt&Fin	64.34	Pl
1	177	178	F	73.0	Central	97.0	Others	Commerce	79.0	Comm&Mgmt	Yes	89.00	Mkt&Fin	70.81	Pl
2	4	5	M	85.8	Central	73.6	Central	Commerce	73.3	Comm&Mgmt	No	96.80	Mkt&Fin	55.50	Pl
3	95	96	M	73.0	Central	78.0	Others	Commerce	65.0	Comm&Mgmt	Yes	95.46	Mkt&Fin	62.16	Pl
4	210	211	M	80.6	Others	82.0	Others	Commerce	77.6	Comm&Mgmt	No	91.00	Mkt&Fin	74.49	Pl
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
97	15	16	F	65.0	Central	75.0	Central	Commerce	69.0	Comm&Mgmt	Yes	72.00	Mkt&Fin	64.66	Pl
98	107	108	M	82.0	Others	90.0	Others	Commerce	83.0	Comm&Mgmt	No	80.00	Mkt&HR	73.52	Pl
99	135	136	F	72.0	Central	56.0	Others	Science	69.0	Comm&Mgmt	No	55.60	Mkt&HR	65.63	Pl
100	50	51	F	75.2	Central	73.2	Central	Science	68.4	Comm&Mgmt	No	65.00	Mkt&HR	62.98	Pl
101	44	45	F	77.0	Others	73.0	Others	Commerce	81.0	Comm&Mgmt	Yes	89.00	Mkt&Fin	69.70	Pl

102 rows × 16 columns

```
placementdataset[(placementdataset['degree_t']=='Sci&Tech')&(placementdataset['salary']>placementdataset['salary'].mean())].sort_values(b
```

Saved successfully! ×

	index	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	stat
0	150	151	M	71.00	Central	58.66	Central	Science	58.00	Sci&Tech	Yes	56.00	Mkt&Fin	61.30	Plac
1	77	78	M	64.00	Others	80.00	Others	Science	65.00	Sci&Tech	Yes	69.00	Mkt&Fin	57.65	Plac
2	163	164	M	63.00	Others	67.00	Others	Science	64.00	Sci&Tech	No	75.00	Mkt&Fin	66.46	Plac
3	174	175	M	73.24	Others	50.83	Others	Science	64.27	Sci&Tech	Yes	64.00	Mkt&Fin	66.23	Plac
4	53	54	M	80.00	Others	70.00	Others	Science	72.00	Sci&Tech	No	87.00	Mkt&HR	71.04	Plac
5	39	40	M	81.00	Others	68.00	Others	Science	64.00	Sci&Tech	No	93.00	Mkt&Fin	62.56	Plac
6	145	146	M	89.40	Others	65.66	Others	Science	71.25	Sci&Tech	No	72.00	Mkt&HR	63.23	Plac
7	128	129	M	80.40	Central	73.40	Central	Science	77.72	Sci&Tech	Yes	81.20	Mkt&HR	76.26	Plac
8	24	25	M	76.50	Others	97.70	Others	Science	78.86	Sci&Tech	No	97.40	Mkt&Fin	74.01	Plac
9	70	71	M	82.00	Others	61.00	Others	Science	62.00	Sci&Tech	No	89.00	Mkt&Fin	65.45	Plac
10	22	23	F	69.80	Others	60.80	Others	Science	72.23	Sci&Tech	No	55.53	Mkt&HR	68.81	Plac
12	153	154	M	49.00	Others	59.00	Others	Science	65.00	Sci&Tech	Yes	86.00	Mkt&Fin	62.48	Plac

Calculatting number of male and female students

```
len(placementdataset.gender.unique())
placementdataset['gender'].value_counts()
#number of male and female students
```

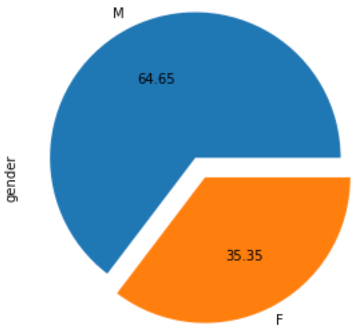
```
M    139
F     76
Name: gender, dtype: int64
```

```
plt.figure(figsize·=·(15,5))
```

Saved successfully!

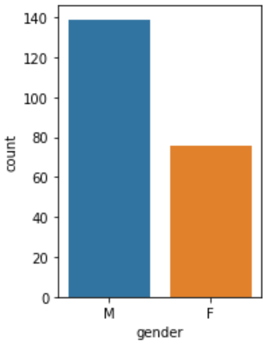
```
nts().plot(kind·=·'pie',autopct·=·'%.2f',explode=explode)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b43f40be0>
```



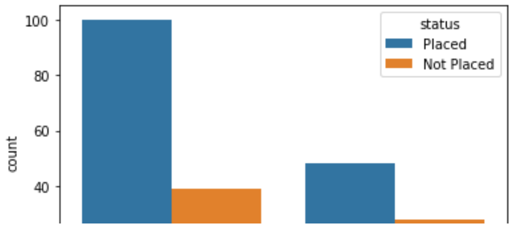
```
plt.subplot(122)
sns.countplot(data = placementdataset, x = 'gender')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b43d91940>
```



```
sns.countplot(x="gender",data=placementdataset,hue="status")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b440554f0>
```

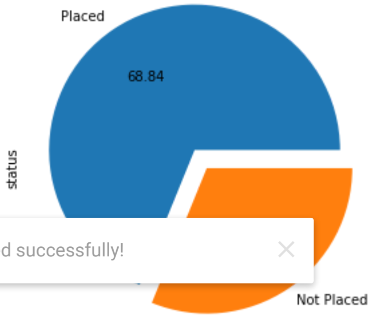


```
len(placementdataset.status.unique())
placementdataset['status'].value_counts()
```

```
Placed        148
Not Placed     67
Name: status, dtype: int64
```

```
plt.figure(figsize = (15,5))
explode = [0.15,0]
plt.subplot(121)
placementdataset['status'].value_counts().plot(kind ='pie',autopct = '%.2f',explode=explode)
```

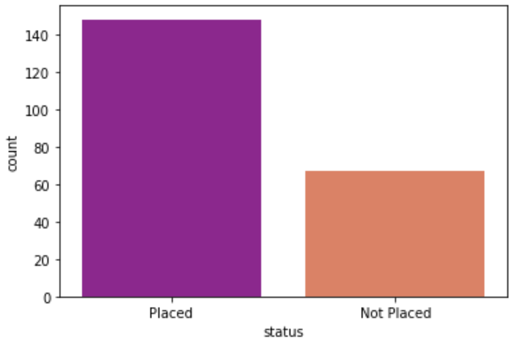
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b3faf8850>
```



Saved successfully!

```
sns.countplot(placementdataset['status'],palette='plasma')
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. F
  warnings.warn(
```



```
placementdataset.groupby('degree_t')['status'].value_counts()
```

```
degree_t   status
Comm&Mgmt  Placed        102
           Not Placed     43
Others     Not Placed      6
           Placed          5
Sci&Tech   Placed         41
           Not Placed     18
Name: status, dtype: int64
```

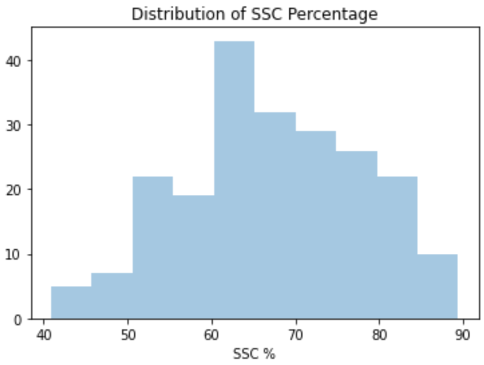
```
placementdataset.groupby('specialisation')['status'].value_counts()
```

```
specialisation  status
Mkt&Fin         Placed        95
                Not Placed    25
Mkt&HR          Placed        53
                Not Placed    42
Name: status, dtype: int64
```

SSC percentage

```
sns.distplot(placementdataset['ssc_p'], kde=False)
plt.title('Distribution of SSC Percentage')
plt.xlabel('SSC %')
```

```
    /usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
      warnings.warn(msg, FutureWarning)
    Text(0.5, 0, 'SSC %')
```



```
placementdataset['ssc_b'].value_counts()
#Students are usually from 2 types of Secondary Education (10th Grade) boards

#Central ~ 54%
#Others ~ 46%
#Also the gender count is more in Central board.
#Students from a Central Board in Secondary Education have slightly higher chance of getting placed.
```

Saved successfully!

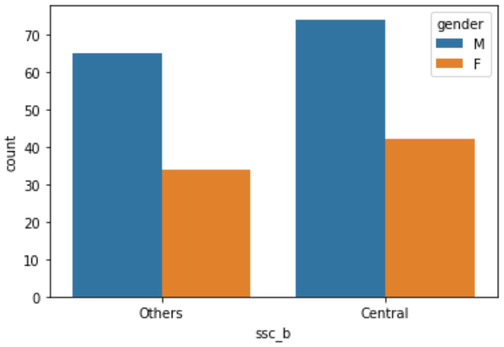
```
    Others      99
    Name: ssc_b, dtype: int64
```

```
df = pd.DataFrame(placementdataset.groupby(['ssc_b','status'])['status'].count())
df
```

		status
ssc_b	**status**	
Central	**Not Placed**	38
	Placed	78
Others	**Not Placed**	29
	Placed	70

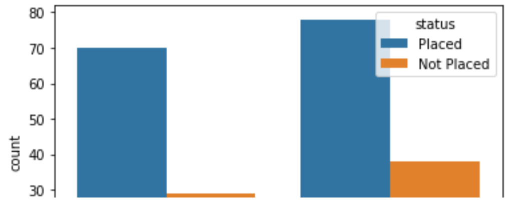
```
sns.countplot(x='ssc_b', hue='gender', data=placementdataset)
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f2b43c3f910>
```



```
sns.countplot(x='ssc_b', hue='status', data=placementdataset)
#From the above analysis I can say that, SSC board is not important to recruiters when it come to hiring candidates
```

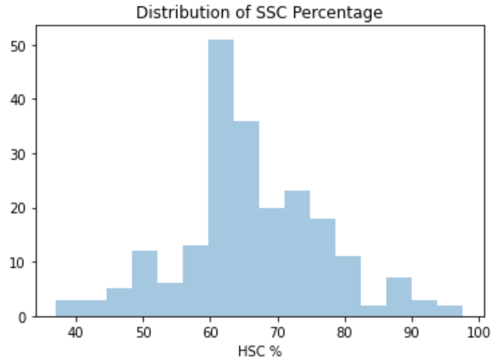
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b43ba7910>
```



HSC Percentage

```
sns.distplot(placementdataset['hsc_p'], kde=False)
plt.title('Distribution of SSC Percentage')
plt.xlabel('HSC %')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
Text(0.5, 0, 'HSC %')
```



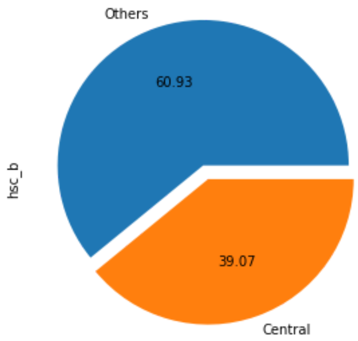
Saved successfully!

```
placementdataset['hsc_b'].value_counts()
```

```
Others     131
Central     84
Name: hsc_b, dtype: int64
```

```
plt.figure(figsize = (20,5))
explode = [0.1,0]
plt.subplot(131)
placementdataset['hsc_b'].value_counts().plot(kind ='pie',autopct = '%.2f',explode=explode)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b43b00ac0>
```

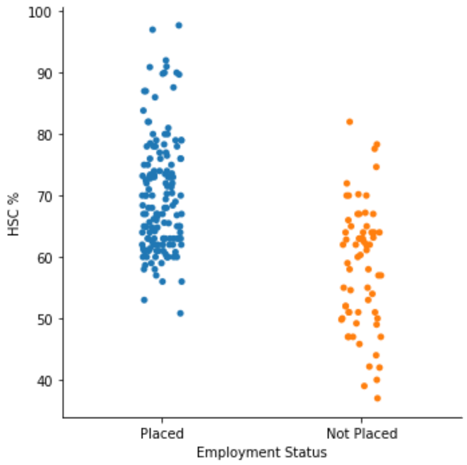


```
df = pd.DataFrame(placementdataset.groupby(['hsc_b','status'])['status'].count())
df
#Here also we see 2 boards

#Central ~ 61%
#Others ~ 39%
#In 12th Grade Male students strength is almost the double in Others Boards

#Students from Others Board during their 12th Grade have a higher chance of getting placed.
```

		status
hsc_b	status	
Central	Not Placed	27
	Placed	57

```
sns.catplot(y='hsc_p', x='status', data=placementdataset)
plt.xlabel('Employment Status')
plt.ylabel('HSC %')
```

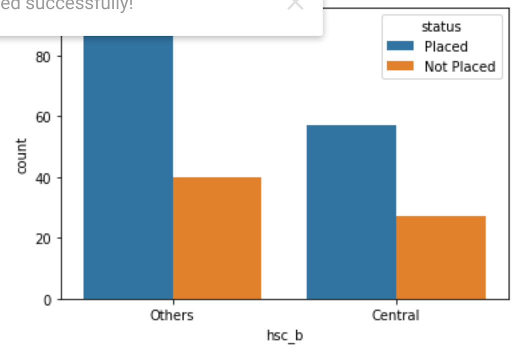
```
    Text(3.799999999999997, 0.5, 'HSC %')
```



```
sns.countplot(x='hsc_b', hue='status', data=placementdataset)
```

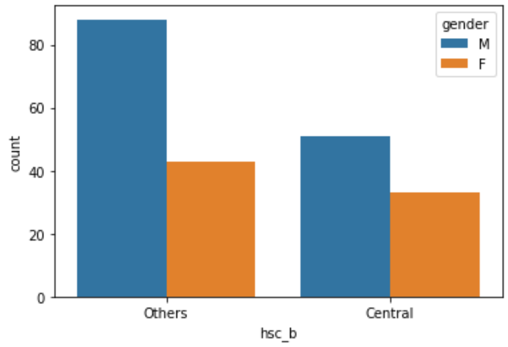
Saved successfully!

```
Subplot at 0x7f2b47072a30>
```



```
sns.countplot(x='hsc_b', hue='gender', data=placementdataset)
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f2b44766250>
```



HSC Specialization

```
placementdataset['hsc_s'].value_counts()
```

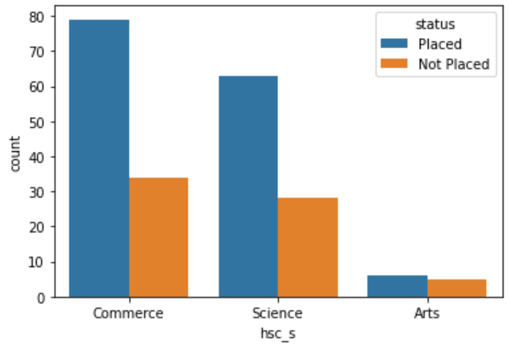
```
    Commerce    113
    Science      91
    Arts         11
    Name: hsc_s, dtype: int64
```

```
df = pd.DataFrame(placementdataset.groupby(['hsc_s','status'])['status'].count())
df
```

		status
hsc_s	**status**	
Arts	**Not Placed**	5
	Placed	6
Commerce	**Not Placed**	34
	Placed	79
Science	**Not Placed**	28
	Placed	63

```
sns.countplot(x='hsc_s', hue='status', data=placementdataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b476d2610>
```

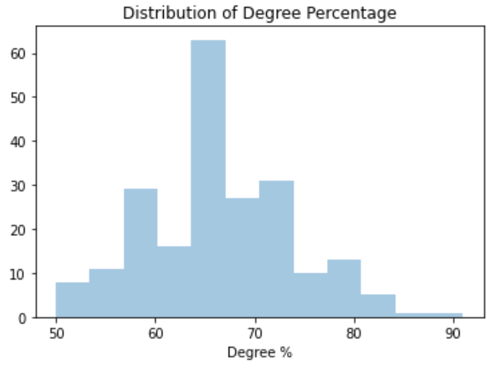


Degree percentage

Saved successfully!

```
sns.distplot(placementdataset['degree_p'], kde=False)
plt.title('Distribution of Degree Percentage')
plt.xlabel('Degree %')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
Text(0.5, 0, 'Degree %')
```



```
sns.catplot(y='degree_p', x='status', data=placementdataset)
plt.xlabel('Employment Status')
plt.ylabel('Degree %')
```

```
Text(10.049999999999997, 0.5, 'Degree %')
```

Degree type

```
placementdataset['degree_t'].value_counts()
```

```
Comm&Mgmt    145
Sci&Tech      59
Others        11
Name: degree_t, dtype: int64
```

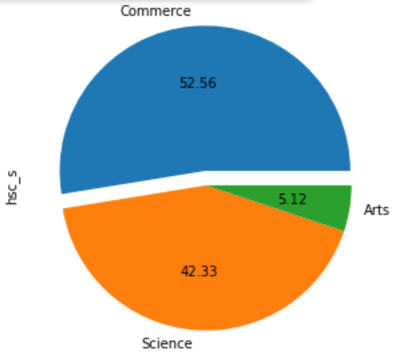
```
df = pd.DataFrame(placementdataset.groupby(['degree_t','status'])['status'].count())
df
```

		status
degree_t	**status**	
Comm&Mgmt	**Not Placed**	43
	Placed	102
Others	**Not Placed**	6
	Placed	5
Sci&Tech	**Not Placed**	18
	Placed	41

```
plt.figure(figsize = (20,5))
explode = [0.1,0,0]
plt.subplot(131)
                                       ts().plot(kind ='pie',autopct = '%.2f',explode=explode)
```

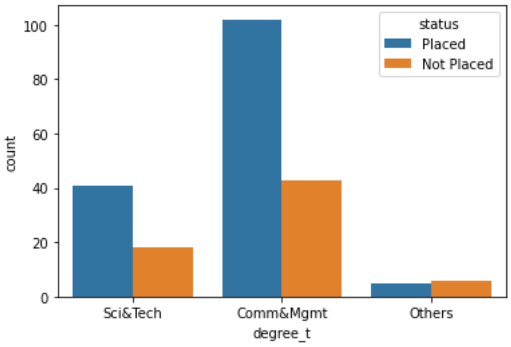
Saved successfully!

```
Subplot at 0x7f2b440e2460>
```



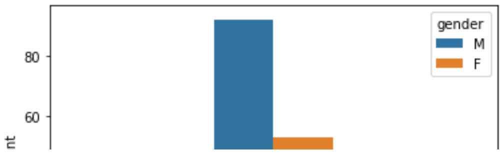
```
sns.countplot(x='degree_t', hue='status', data=placementdataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b446becd0>
```



```
sns.countplot(x='degree_t', hue='gender', data=placementdataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b445f5e50>
```



Work Experience

```
placementdataset['workex'].value_counts()
```

```
No     141
Yes     74
Name: workex, dtype: int64
```

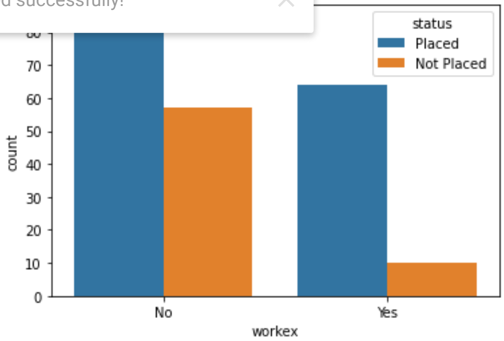
```
df = pd.DataFrame(placementdataset.groupby(['workex','status'])['status'].count())
df
```

		status
workex	status	
No	Not Placed	57
	Placed	84
Yes	Not Placed	10
	Placed	64

```
sns.countplot(x='workex', hue='status', data=placementdataset)
#It is clear that candidate with work experience have higher chance of getting placed
```

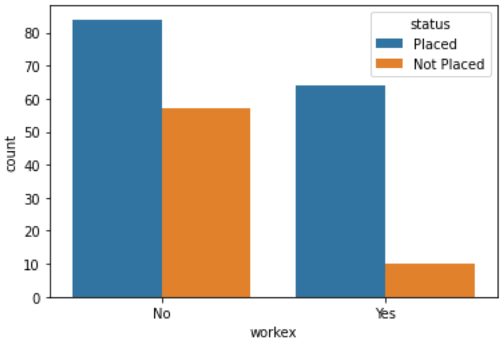
Saved successfully!

```
Subplot at 0x7f2b43b11790>
```



```
sns.countplot(x='workex', hue='status', data=placementdataset)
```

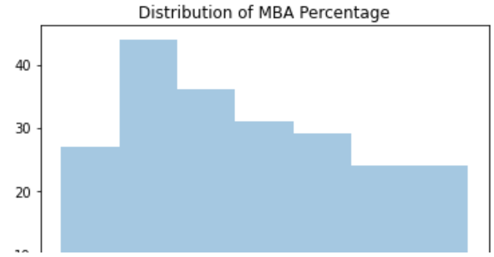
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b44541b50>
```



Employement test percentage

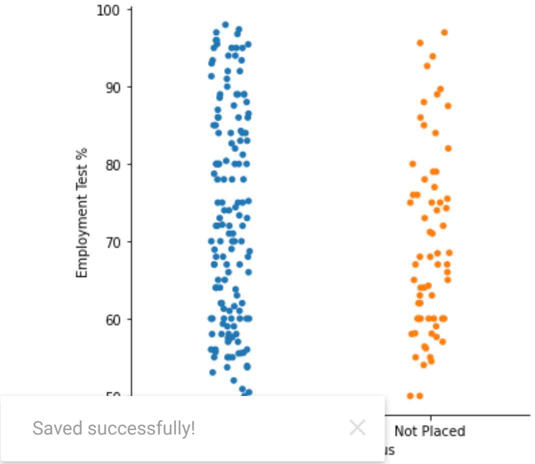
```
sns.distplot(placementdataset['etest_p'], kde=False)
plt.title('Distribution of MBA Percentage')
plt.xlabel('Employment Test %')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
Text(0.5, 0, 'Employment Test %')
```



```
sns.catplot(y='etest_p', x='status', data=placementdataset)
plt.xlabel('Employment Status')
plt.ylabel('Employment Test %')
```

```
Text(3.7999999999999997, 0.5, 'Employment Test %')
```



MBA Specialization

```
placementdataset['specialisation'].value_counts()
```

```
Mkt&Fin    120
Mkt&HR      95
Name: specialisation, dtype: int64
```

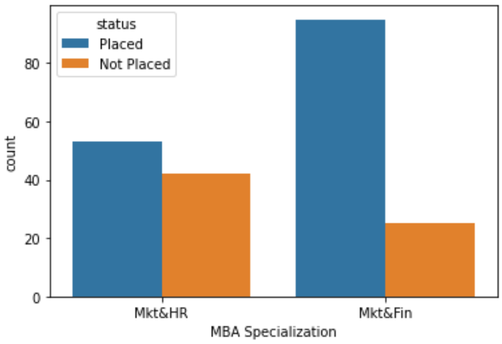
```
df = pd.DataFrame(placementdataset.groupby(['specialisation','status'])['status'].count())
df
```

		status
specialisation	**status**	
Mkt&Fin	**Not Placed**	25
	Placed	95
Mkt&HR	**Not Placed**	42
	Placed	53

```
placementdataset.groupby(['specialisation'])['status'].count().plot(kind = 'bar',color = 'orange')
plt.show()
#which specialization has maximum placed students
```

```
sns.countplot(x='specialisation', hue='status', data=placementdataset)
plt.xlabel('MBA Specialization')
```

```
Text(0.5, 0, 'MBA Specialization')
```



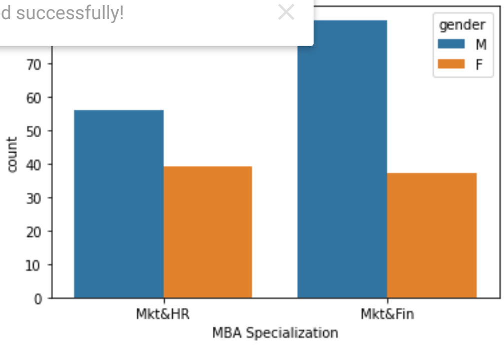
```
df=placementdataset.groupby('specialisation')['mba_p'].mean()
df=pd.DataFrame(df).rename(columns={'mba_p': 'avg. mba %'}).reset_index()
df
```

	specialisation	avg. mba %
0	Mkt&Fin	62.825667
1	Mkt&HR	61.586632

```
sns.countplot(x='specialisation', hue='gender', data=placementdataset)
plt.xlabel('MBA Specialization')
```

```
n')
```

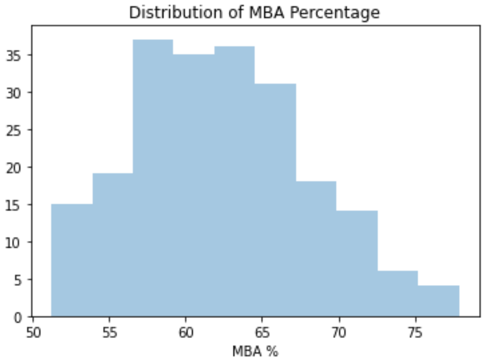
Saved successfully!



MBA Percentage

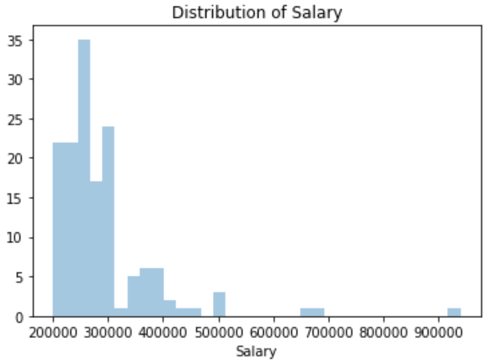
```
sns.distplot(placementdataset['mba_p'], kde=False)
plt.title('Distribution of MBA Percentage')
plt.xlabel('MBA %')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
Text(0.5, 0, 'MBA %')
```



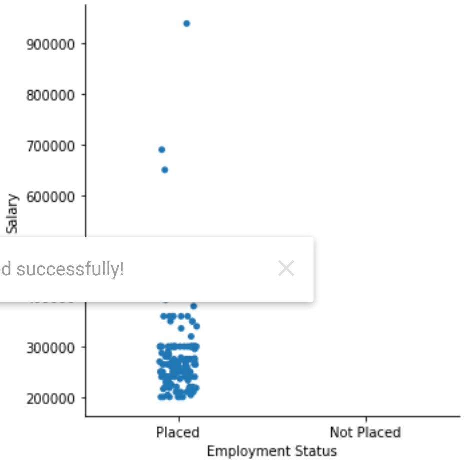
```
sns.distplot(placementdataset['salary'], kde=False)
plt.title('Distribution of Salary')
plt.xlabel('Salary')
#most important factor salary
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
Text(0.5, 0, 'Salary')
```



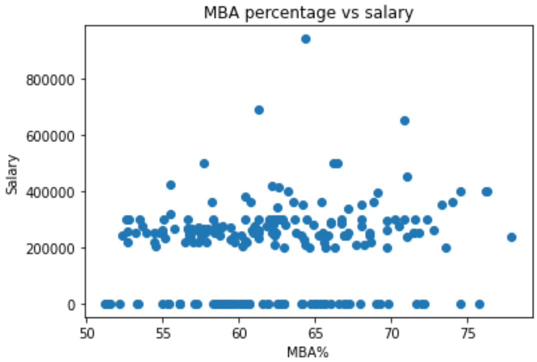
```
sns.catplot(y='salary', x='status', data=placementdataset)
plt.xlabel('Employment Status')
plt.ylabel('Salary')
```

```
Text(-14.950000000000003, 0.5, 'Salary')
```



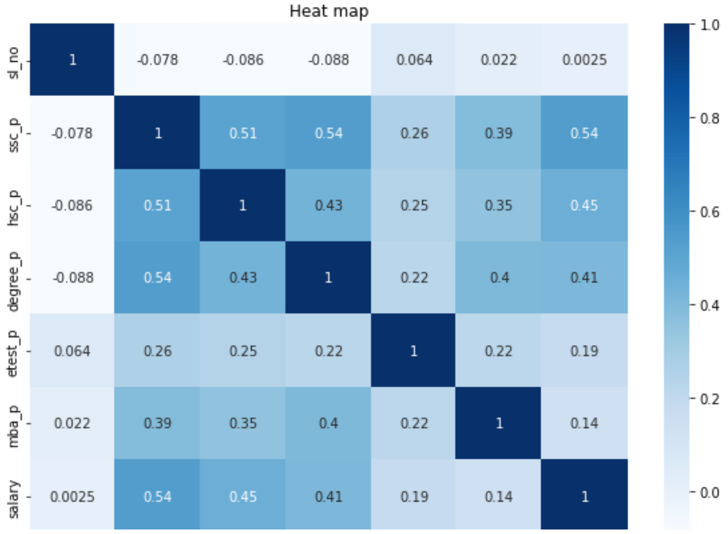
Saved successfully!

```
fig = plt.scatter(data=placementdataset, x='mba_p', y='salary')
plt.xlabel("MBA%")
plt.ylabel("Salary")
plt.title("MBA percentage vs salary")
plt.show()
#how much dependancy between mba percentage and salary
```



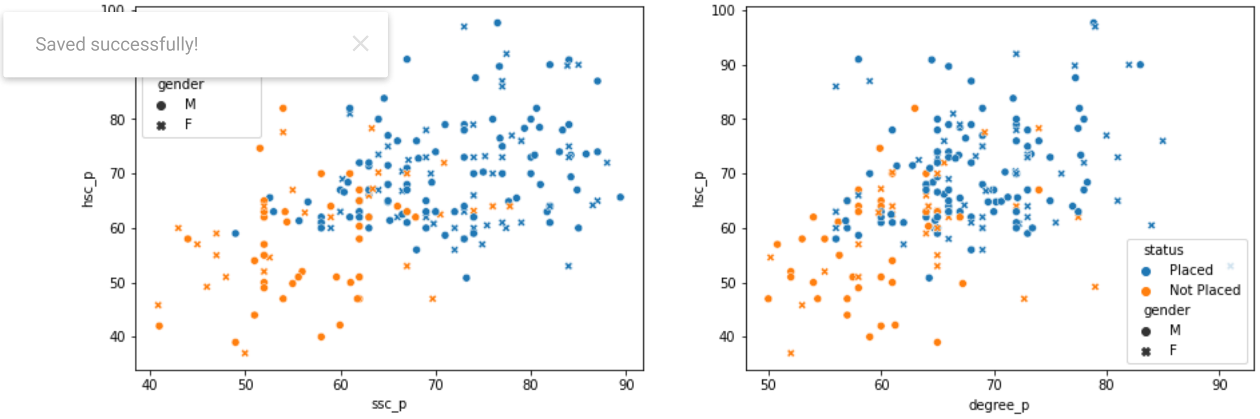
```
# calculates correlations between columns in  the dataframe.
corr = placementdataset.corr()
f, ax = plt.subplots(figsize=(10, 7))
cmap = sns.diverging_palette(220, 20)
sns.heatmap(placementdataset.corr(), annot = True, cmap = 'Blues').set_title('Heat map')

#Since ssc_p has some amount of correlation with hsc_p and degree_p. It also tells us student who scored more in 10th Grade also scored h

#We don't see any high correlation between other features.

#Since ssc_p , hsc_p and degree_p have high correlation let's check their relation with placement status
```

```
Text(0.5, 1.0, 'Heat map')
```



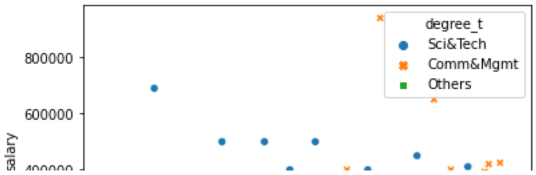
```
fig = plt.figure(figsize = (15,5))
plt.subplot(121)
sns.scatterplot(data =placementdataset, x ='ssc_p' , y='hsc_p',hue = 'status',style = 'gender')
plt.subplot(122)
sns.scatterplot(data =placementdataset, x ='degree_p' , y='hsc_p',hue = 'status',style = 'gender')

#It's clear that usually students who have a high percentage in both 10th , 12th and degree have a high chance of getting placed.

#Since the campus placement is done in a MBA college let's check which specialisation gives more salary.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b43b111f0>
```



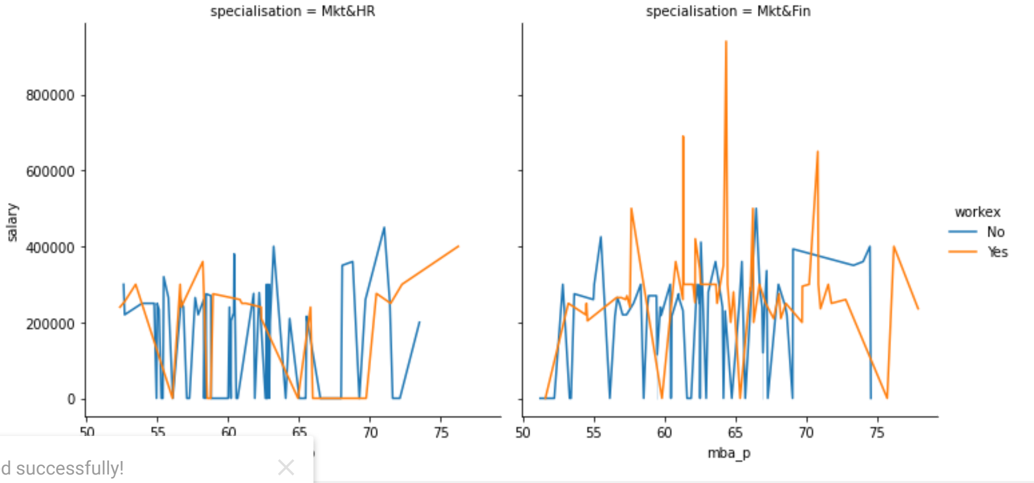
```
sns.scatterplot("etest_p","salary",data=placementdataset,hue="degree_t",style="degree_t")
sns.relplot("mba_p","salary",data=placementdataset,hue="specialisation")
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y
  warnings.warn(
<seaborn.axisgrid.FacetGrid at 0x7f2b41f57580>
```



```
sns.relplot("mba_p","salary",data=placementdataset,kind="line",col="specialisation",hue="workex")
```

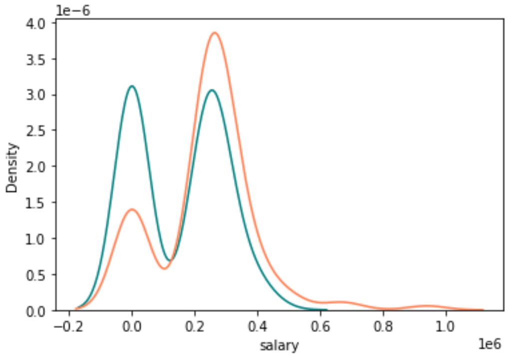
```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y
  warnings.warn(
<seaborn.axisgrid.FacetGrid at 0x7f2b41f4f9a0>
```



Saved successfully!

```
sns.distplot(placementdataset[placementdataset['specialisation'] == 'Mkt&HR']['salary'], color = 'teal',hist= False)
sns.distplot(placementdataset[placementdataset['specialisation'] == 'Mkt&Fin']['salary'],color = 'coral',hist =False)

#Students who have Marketing and Finance specialisation have the highest salary of all.
#Students who have Marketing and HR specialisation have less salary as compared to Marketing and Finance.
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b41dd5cd0>
```

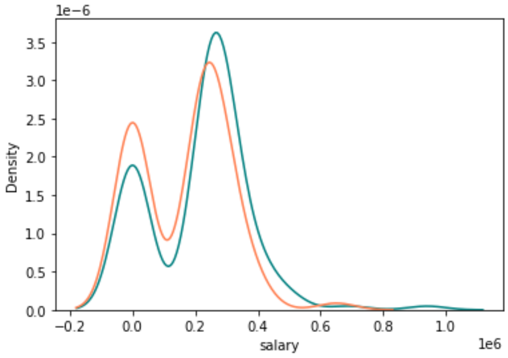


```
plt.figure(figsize=(7,7))
sns.boxplot(x='gender', y='salary', hue='hsc_s', data=placementdataset, palette=['blue', 'green', 'red'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b3fd4ed90>
```

```
sns.distplot(placementdataset[placementdataset['gender'] == 'M']['salary'], color = 'teal',hist= False)
sns.distplot(placementdataset[placementdataset['gender'] == 'F']['salary'],color = 'coral',hist =False)
 #A Male student grabs the highest package nevertheless Female salary also catches up but it's not the highest.
```

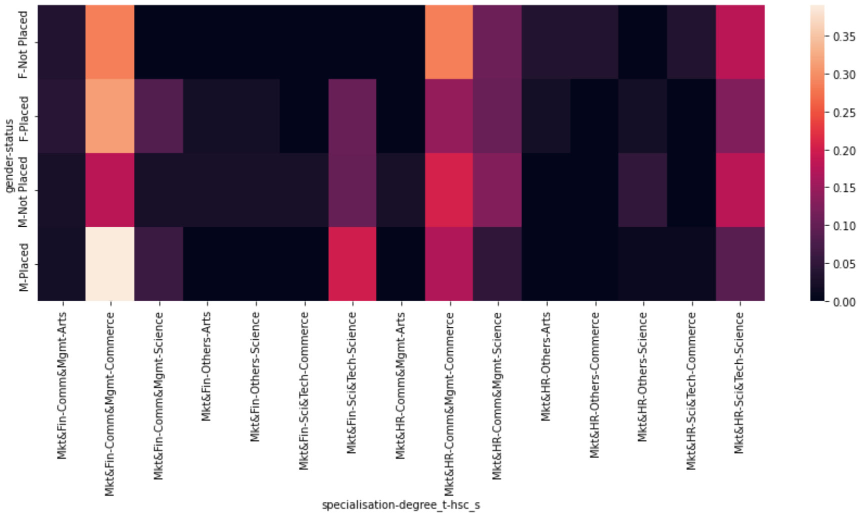
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will b
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b3fc909a0>
```



Saved successfully!

```
#Finally let's see what path a student must follow to have higher chance of getting placed.
plt.figure(figsize = (15,5))
map= pd.crosstab([placementdataset['gender'] , placementdataset['status']],[placementdataset['specialisation'],placementdataset['degree_t
sns.heatmap(map)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b3fce6c40>
```



```
groups = placementdataset.groupby(['status', pd.cut(placementdataset.hsc_p, [10, 50, 70, 100])])
groups.size().unstack()

#does high school percentage matter
#Yes , high school mark matter we can see that student with (mark > 70) have 91% chance of getting placed
#similarly people with high school percentage less 50% have no chance of getting placed
```

hsc_p	(10, 50]	(50, 70]	(70, 100]
status			
Not Placed	16	45	6
Placed	0	84	64

✓ 0s completed at 20:09 ● ×

Saved successfully! ×