eda-1

March 20, 2025

[1]:

```
#https://www.kaggle.com/code/themlphdstudent/
 ↪campus-recruitment-eda-classification
import numpy as np
import pandas as pd
# data visualization
import matplotlib.pyplot as plt
import seaborn as sns





# machine learning
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn import preprocessing
```

[3]:

```
data = pd.read_csv('/content/Placement.csv')
```

[ ]:

```
data
```

[ ]:

```
     sl_no gender  ssc_p    ssc_b  hsc_p    hsc_b     hsc_s  degree_p  \
0        1      M  67.00   Others  91.00   Others  Commerce     58.00
1        2      M  79.33  Central  78.33   Others   Science     77.48
2        3      M  65.00  Central  68.00  Central      Arts     64.00
3        4      M  56.00  Central  52.00  Central   Science     52.00
4        5      M  85.80  Central  73.60  Central  Commerce     73.30
..     …    …    …      …    …      …       …       …
210    211      M  80.60   Others  82.00   Others  Commerce     77.60
211    212      M  58.00   Others  60.00   Others   Science     72.00
212    213      M  67.00   Others  67.00   Others  Commerce     73.00
```

```
213    214      F  74.00   Others  66.00   Others  Commerce     58.00
214    215      M  62.00  Central  58.00   Others   Science     53.00

      degree_t workex  etest_p specialisation  mba_p      status    salary
0     Sci&Tech     No     55.0         Mkt&HR  58.80      Placed  270000.0
1     Sci&Tech    Yes     86.5        Mkt&Fin  66.28      Placed  200000.0
2    Comm&Mgmt     No     75.0        Mkt&Fin  57.80      Placed  250000.0
3     Sci&Tech     No     66.0         Mkt&HR  59.43  Not Placed       NaN
4    Comm&Mgmt     No     96.8        Mkt&Fin  55.50      Placed  425000.0
..         …      …        …            …      …         …         …
210  Comm&Mgmt     No     91.0        Mkt&Fin  74.49      Placed  400000.0
211   Sci&Tech     No     74.0        Mkt&Fin  53.62      Placed  275000.0
212  Comm&Mgmt    Yes     59.0        Mkt&Fin  69.72      Placed  295000.0
213  Comm&Mgmt     No     70.0         Mkt&HR  60.23      Placed  204000.0
214  Comm&Mgmt     No     89.0         Mkt&HR  60.22  Not Placed       NaN

[215 rows x 15 columns]
```

[4]:

```
data.head()
```

[4]:

```
   sl_no gender  ssc_p    ssc_b  hsc_p    hsc_b     hsc_s  degree_p  \
0      1      M  67.00   Others  91.00   Others  Commerce     58.00
1      2      M  79.33  Central  78.33   Others   Science     77.48
2      3      M  65.00  Central  68.00  Central      Arts     64.00
3      4      M  56.00  Central  52.00  Central   Science     52.00
4      5      M  85.80  Central  73.60  Central  Commerce     73.30

    degree_t workex  etest_p specialisation  mba_p      status    salary
0   Sci&Tech     No     55.0         Mkt&HR  58.80      Placed  270000.0
1   Sci&Tech    Yes     86.5        Mkt&Fin  66.28      Placed  200000.0
2  Comm&Mgmt     No     75.0        Mkt&Fin  57.80      Placed  250000.0
3   Sci&Tech     No     66.0         Mkt&HR  59.43  Not Placed       NaN
4  Comm&Mgmt     No     96.8        Mkt&Fin  55.50      Placed  425000.0
```

[5]:

```
print(data.columns.values)
```

```
['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']
```

[6]:

```
print('='*50)
print("Describe data")
print('='*50)
print(data.describe())
```

```
==================================================
Describe data
==================================================
            sl_no       ssc_p       hsc_p    degree_p     etest_p       mba_p  \
count  215.000000  215.000000  215.000000  215.000000  215.000000  215.000000
mean   108.000000   67.303395   66.333163   66.370186   72.100558   62.278186
std     62.209324   10.827205   10.897509    7.358743   13.275956    5.833385
min      1.000000   40.890000   37.000000   50.000000   50.000000   51.210000
25%     54.500000   60.600000   60.900000   61.000000   60.000000   57.945000
50%    108.000000   67.000000   65.000000   66.000000   71.000000   62.000000
75%    161.500000   75.700000   73.000000   72.000000   83.500000   66.255000
max    215.000000   89.400000   97.700000   91.000000   98.000000   77.890000

              salary
count     148.000000
mean   288655.405405
std     93457.452420
min    200000.000000
25%    240000.000000
50%    265000.000000
75%    300000.000000
max    940000.000000
```

[7]:
```
#As it is clear that we don't need sl_no in training model or in EDA. Thus I am␣
 ↪dropping sl_n column. Rest of them I will keep as it is. After performing␣
 ↪EDA I will drop other if needed.
```

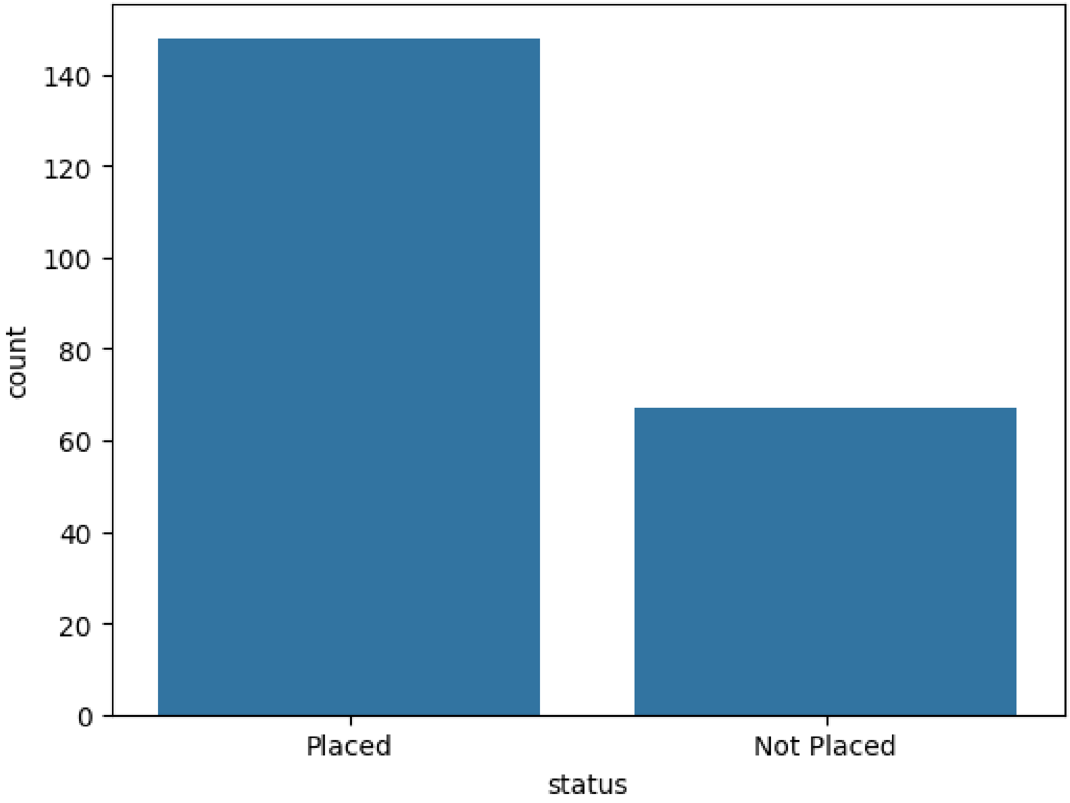
[8]:
```
data = data.drop(['sl_no'], axis=1)
```

[9]:
```
#Exploring important features
```

[9]:
```

```

[10]:
```
sns.countplot( data=data,x=data['status'])
```

[10]:
```
<Axes: xlabel='status', ylabel='count'>
```



[11]:
```
data['gender'].value_counts()
```

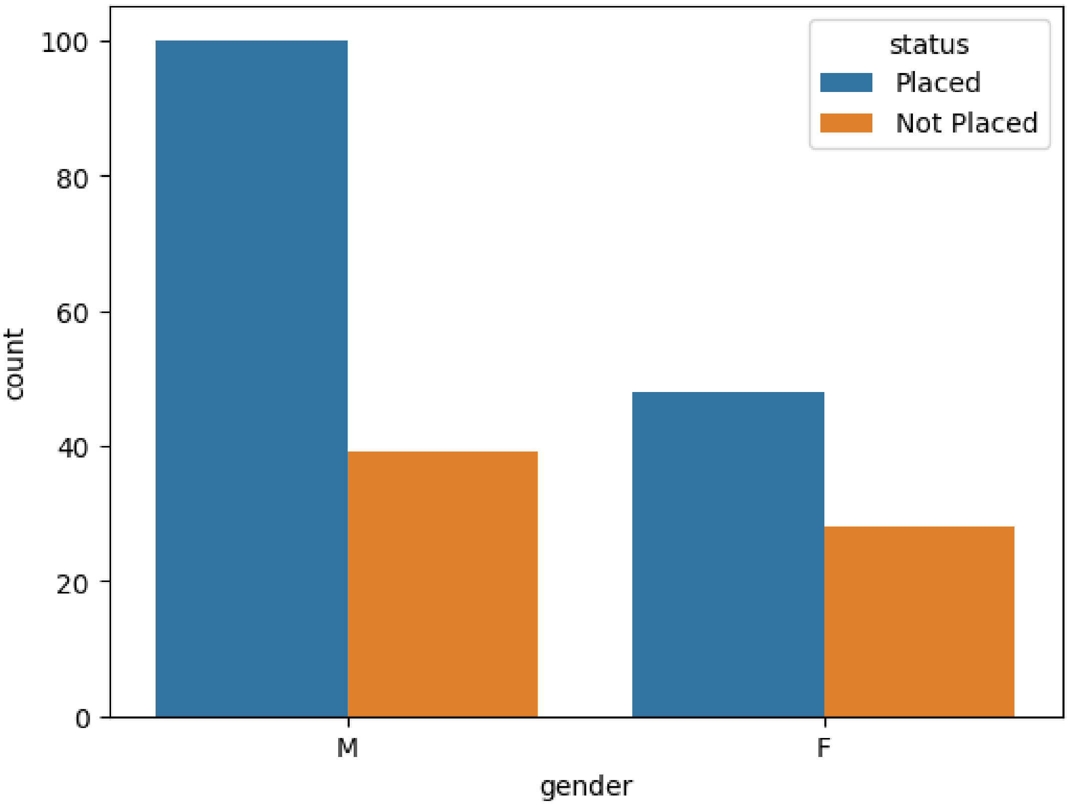
[11]:
```
gender
M    139
F     76
Name: count, dtype: int64
```

[12]:
```
df = pd.DataFrame(data.groupby(['gender','status'])['status'].count())
df
```

[12]:
```
                   status
gender status
F      Not Placed      28
       Placed          48
M      Not Placed      39
       Placed         100
```

[13]:
```
sns.countplot(x='gender', hue='status', data=data)
```

[13]:
```
<Axes: xlabel='gender', ylabel='count'>
```



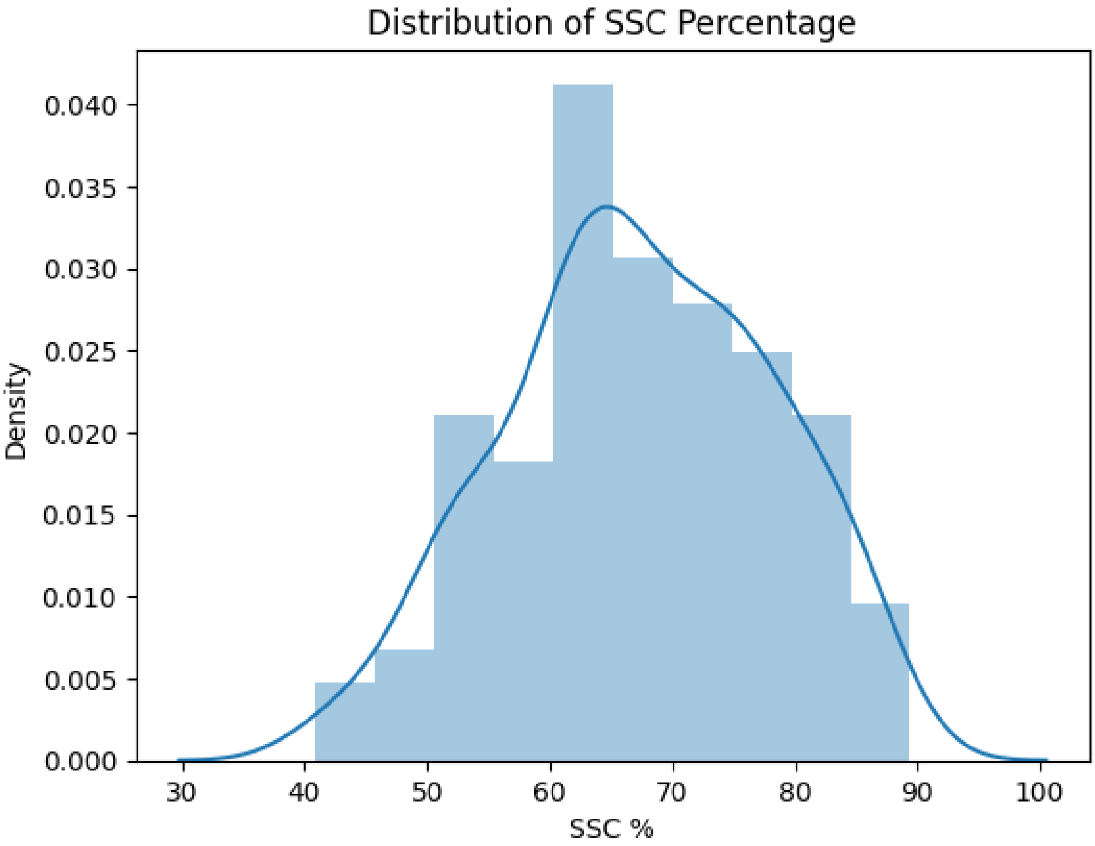
[14]:
```
#Conclusion: Male have high chances of getting placed compared to females.
```

[15]:
```
#SSC Percentage
```

[16]:
```
sns.distplot(data['ssc_p'])
plt.title('Distribution of SSC Percentage')
plt.xlabel('SSC %')
```

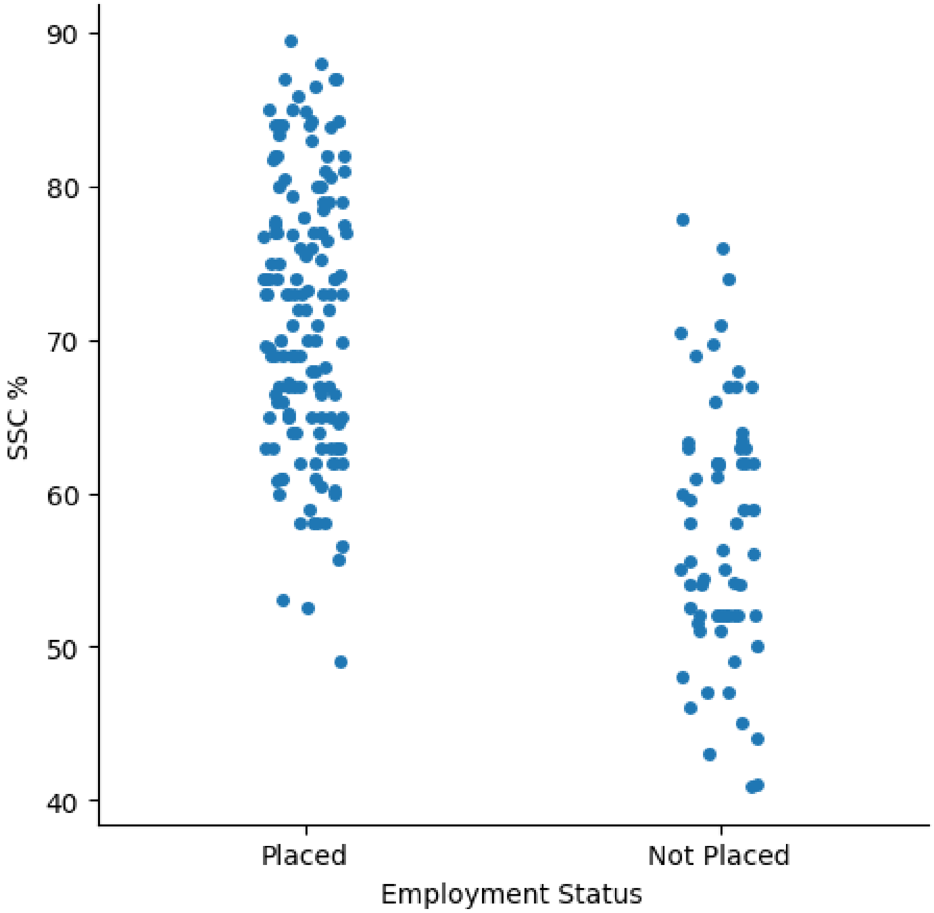
```
<ipython-input-16-f814d30203d6>:1: UserWarning:

`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(data['ssc_p'])
```

[16]:
```
Text(0.5, 0, 'SSC %')
```



```
[17]: sns.catplot(y='ssc_p', x='status', data=data)
      plt.xlabel('Employment Status')
      plt.ylabel('SSC %')
```

```
[17]: Text(30.375617283950618, 0.5, 'SSC %')
```



[18]:
```
data['ssc_b'].value_counts()
```

[18]:
```
ssc_b
Central    116
Others      99
Name: count, dtype: int64
```

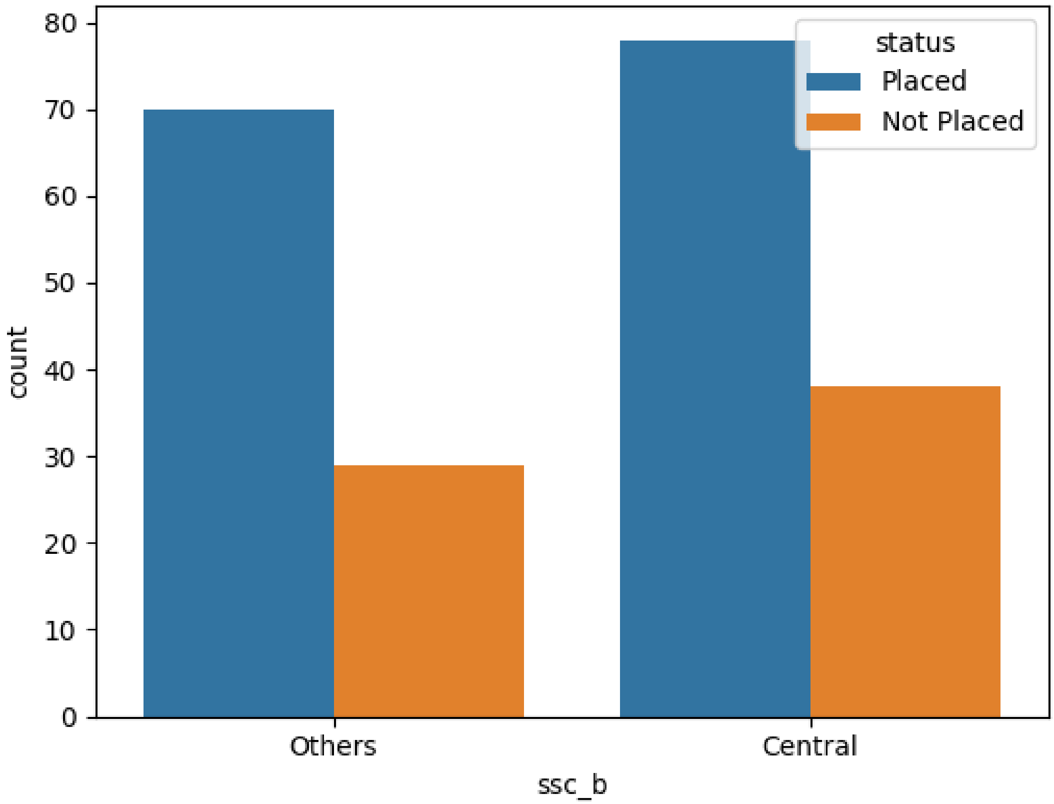
[19]:
```
df = pd.DataFrame(data.groupby(['ssc_b','status'])['status'].count())
df
```

[19]:
```
                    status
ssc_b   status
Central Not Placed      38
        Placed          78
Others  Not Placed      29
        Placed          70
```

[20]:
```
sns.countplot(x='ssc_b', hue='status', data=data)
```

[20]:

```
<Axes: xlabel='ssc_b', ylabel='count'>
```



[21]:

```
#conclusion: From the above analysis I can say that, SSC board is not important to recruiters when it come to hiring candidates. So I am not going to use this feature while training model.
```

[22]:

```
#HSC Percentage
```

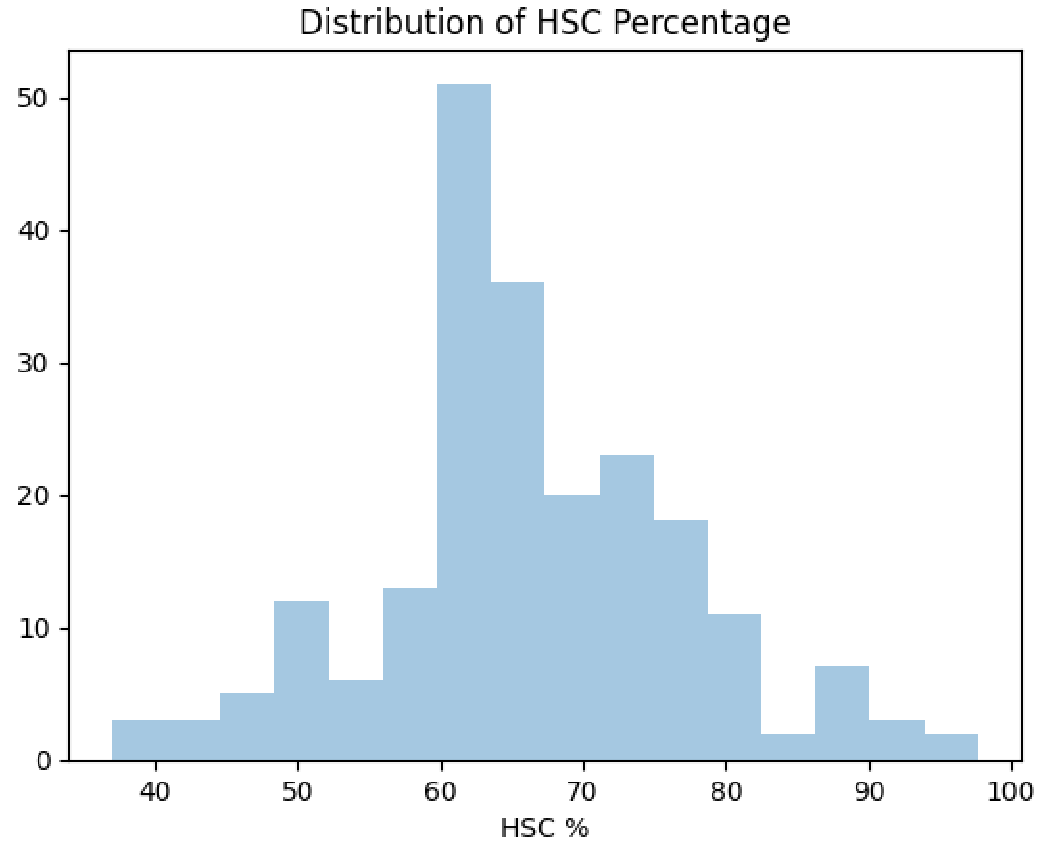
[23]:

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

```
<ipython-input-23-d466214de993>:1: 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
```

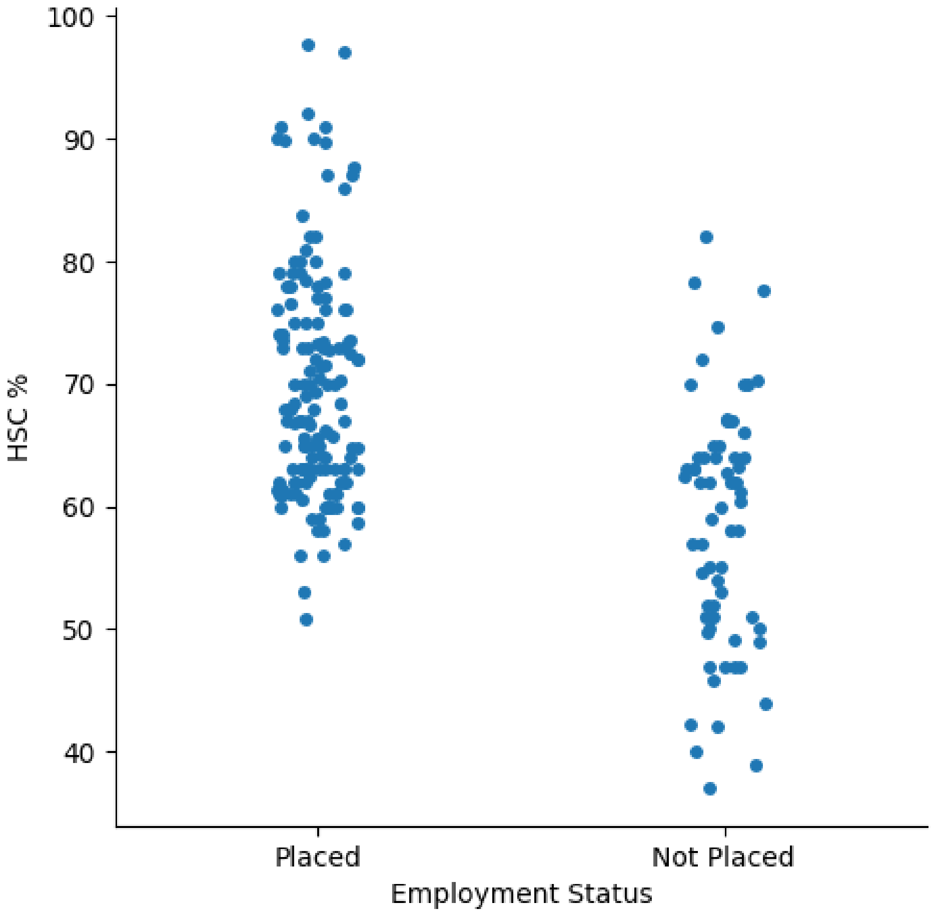
```
sns.distplot(data['hsc_p'], kde=False)
```

```
[23]: Text(0.5, 0, 'HSC %')
```



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

```
[24]: Text(30.71381172839505, 0.5, 'HSC %')
```



```
[25]: #Conclusion: HSC percentage are important features. As all placed students have␣
      ↪higher percentages.
```

```
[26]: #EDA for HSC Board
```

```
[27]: data['hsc_b'].value_counts()
```

```
[27]: hsc_b
      Others     131
      Central     84
      Name: count, dtype: int64
```

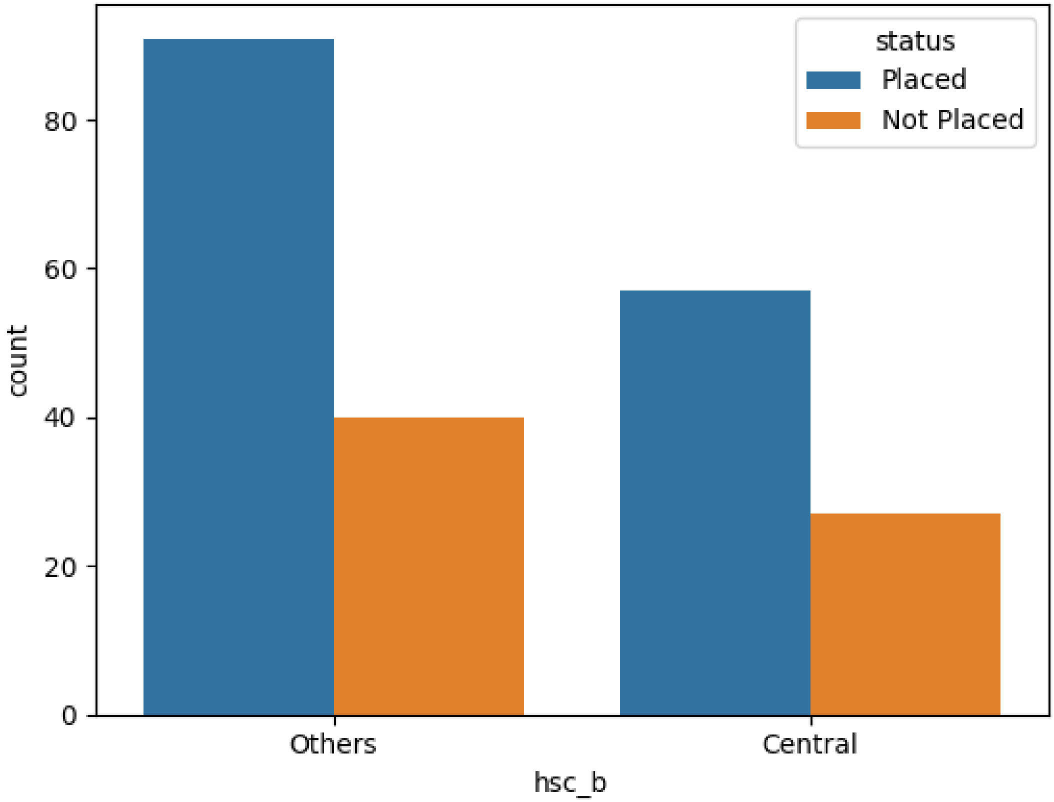
```
[28]: df = pd.DataFrame(data.groupby(['hsc_b','status'])['status'].count())
      df
```

```
[28]:                         status
      hsc_b   status
      Central Not Placed          27
```

```
        Placed          57
Others  Not Placed      40
        Placed          91
```

[29]:
```
sns.countplot(x='hsc_b', hue='status', data=data)
```

[29]:
```
<Axes: xlabel='hsc_b', ylabel='count'>
```



[30]:
```
#Conclusion: From the above analysis I can say that, hSC board is not important
↪to recruiters when it come to hiring candidates. So I am not going to use
↪this feature while training model.
```

[31]:
```
#EDA for HSC Specialisation
```

[32]:
```
data['hsc_s'].value_counts()
```

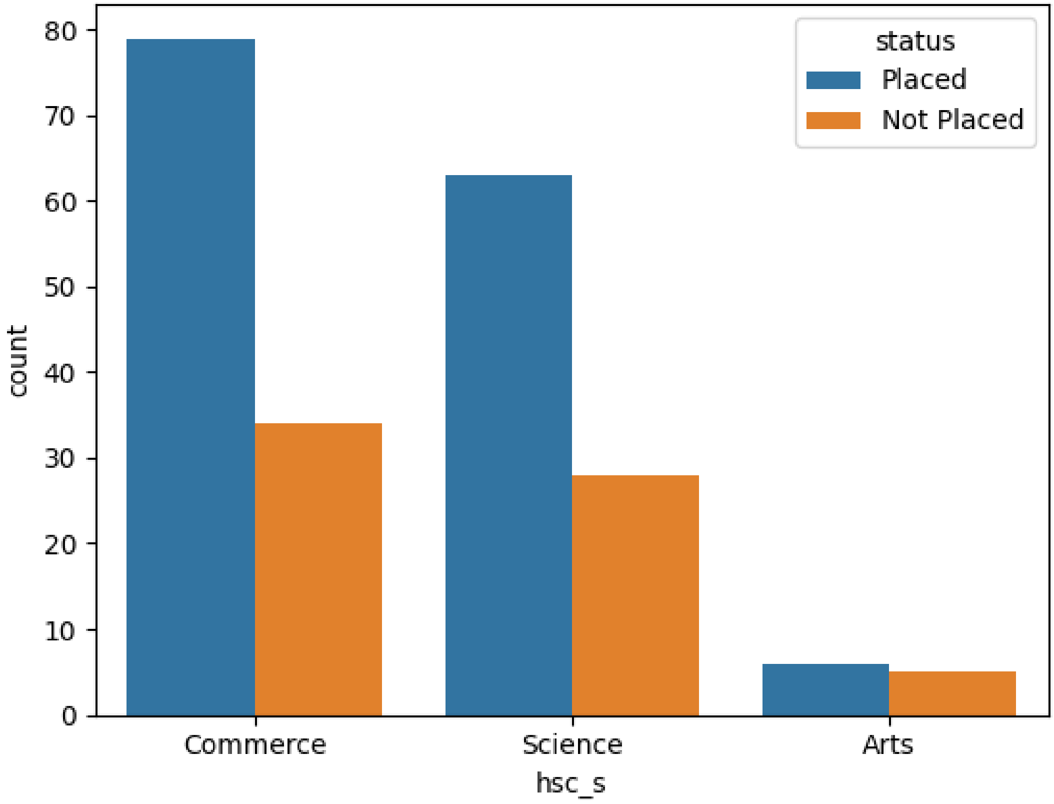
[32]:
```
hsc_s
Commerce    113
Science      91
Arts         11
Name: count, dtype: int64
```

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

```
[33]:                           status
      hsc_s    status
      Arts     Not Placed        5
               Placed            6
      Commerce Not Placed       34
               Placed           79
      Science  Not Placed       28
               Placed           63
```

```
[34]: sns.countplot(x='hsc_s', hue='status', data=data)
```

```
[34]: <Axes: xlabel='hsc_s', ylabel='count'>
```

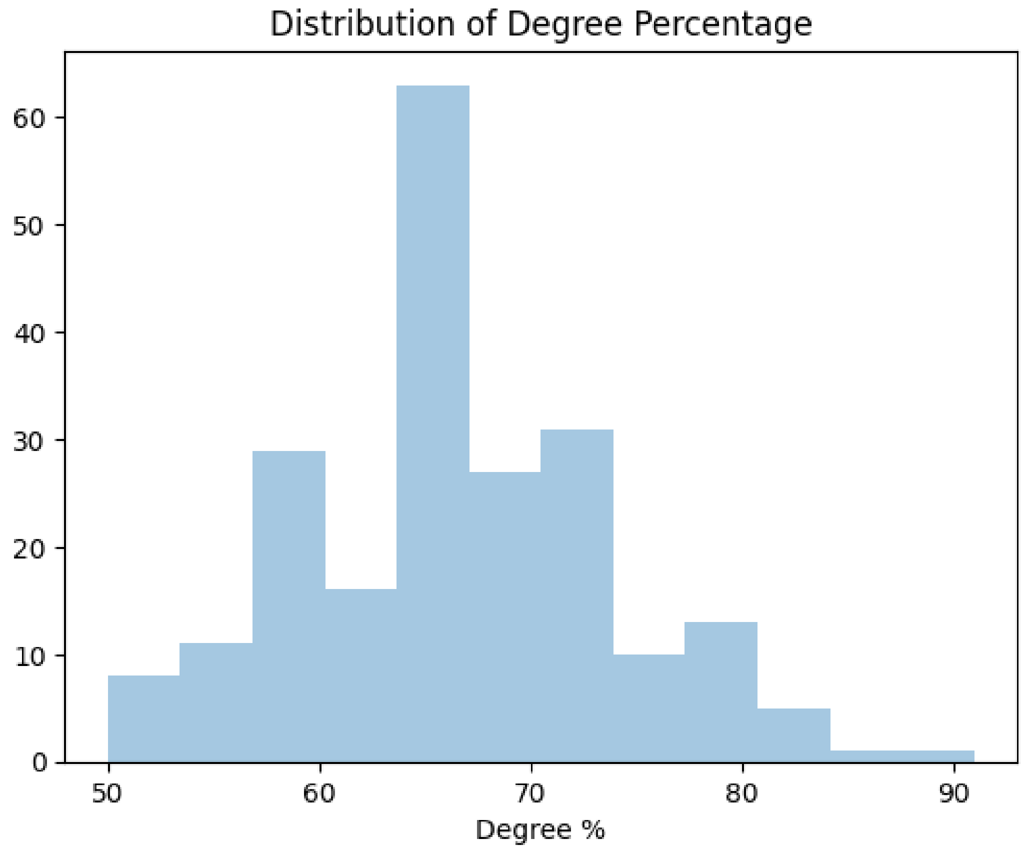


```
[35]: #Degree Percentage
```

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

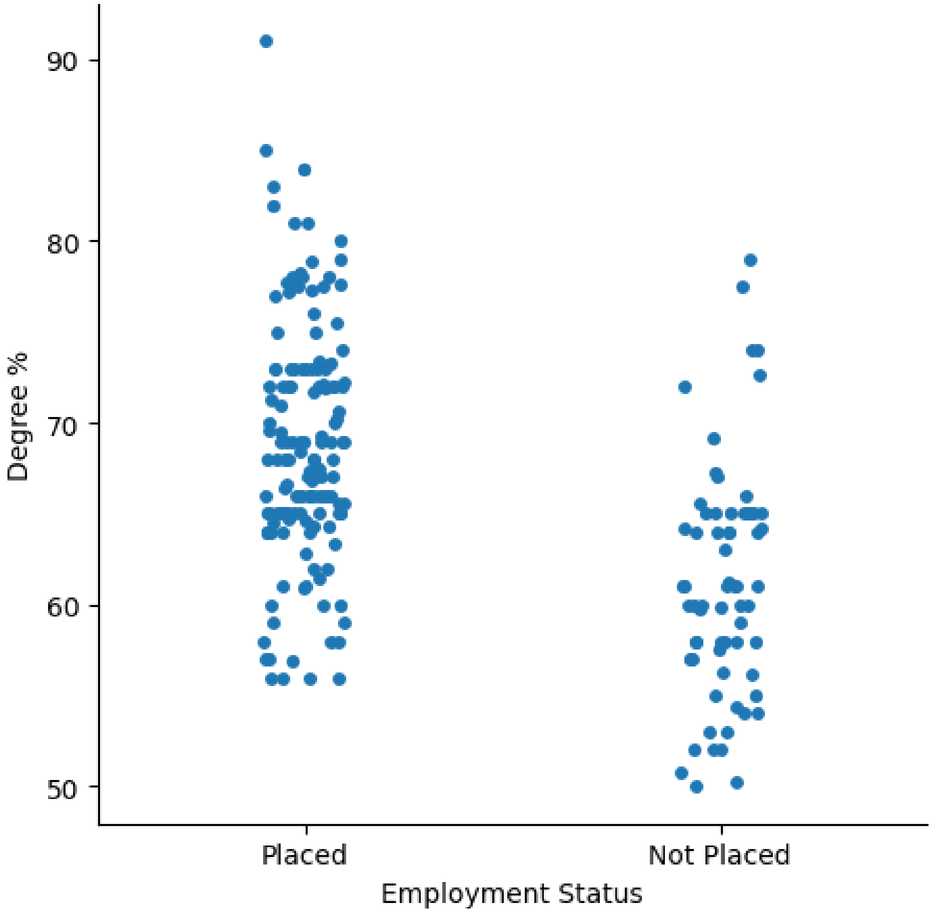
```
<ipython-input-36-2f9bcb03ee09>:1: UserWarning:

`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(data['degree_p'], kde=False)
```

[36]:
```
Text(0.5, 0, 'Degree %')
```



[37]:
```python
sns.catplot(y='degree_p', x='status', data=data)
plt.xlabel('Employment Status')
plt.ylabel('Degree %')
```

[37]:
```
Text(30.519367283950622, 0.5, 'Degree %')
```



[38]:
```
#conclusion: Like SSC and HSC percentages, Degree Percentages are also impotant␣
↪factor to get placed.
```

[39]:
```
#Work Experience
```

[40]:
```
data['workex'].value_counts()
```

[40]:
```
workex
No     141
Yes     74
Name: count, dtype: int64
```

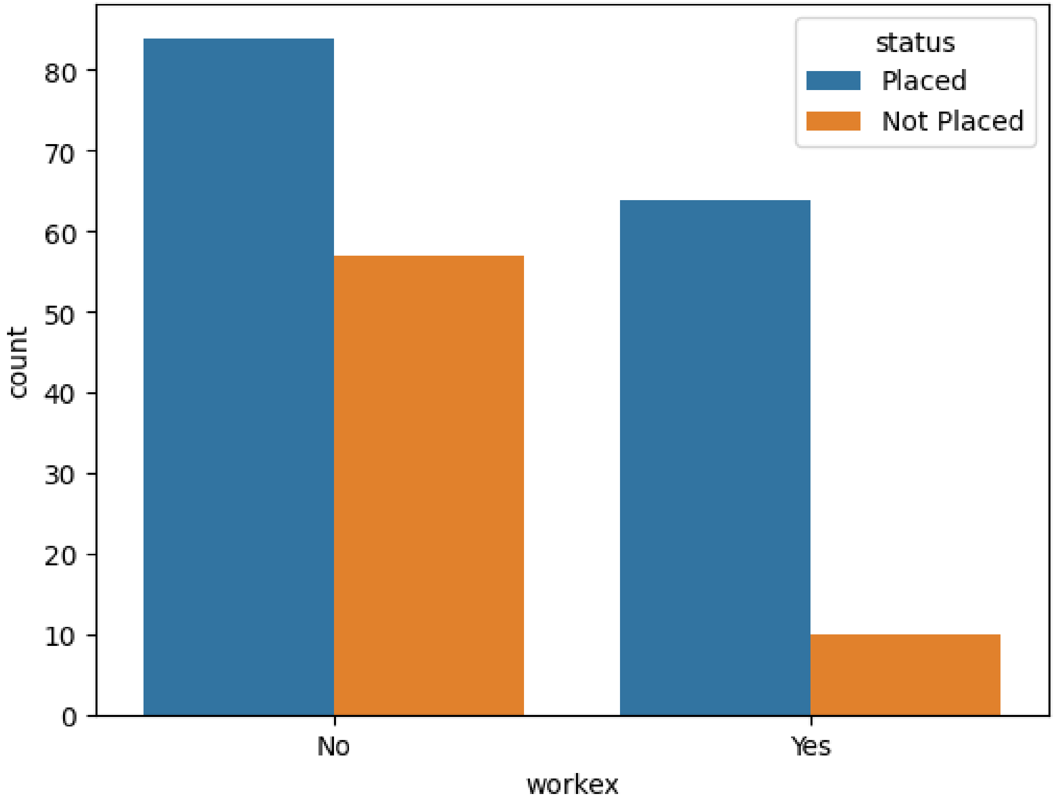
[41]:
```
df = pd.DataFrame(data.groupby(['workex','status'])['status'].count())
df
```

[41]:
```
                   status
workex status
No     Not Placed      57
```

```
        Placed          84
Yes     Not Placed      10
        Placed          64
```

[42]:
```
sns.countplot(x='workex', hue='status', data=data)
```

[42]:
```
<Axes: xlabel='workex', ylabel='count'>
```



[44]:
```
##Conclusion: It is clear that candidate with work experience have higher␣
↪chance of getting placed.
```

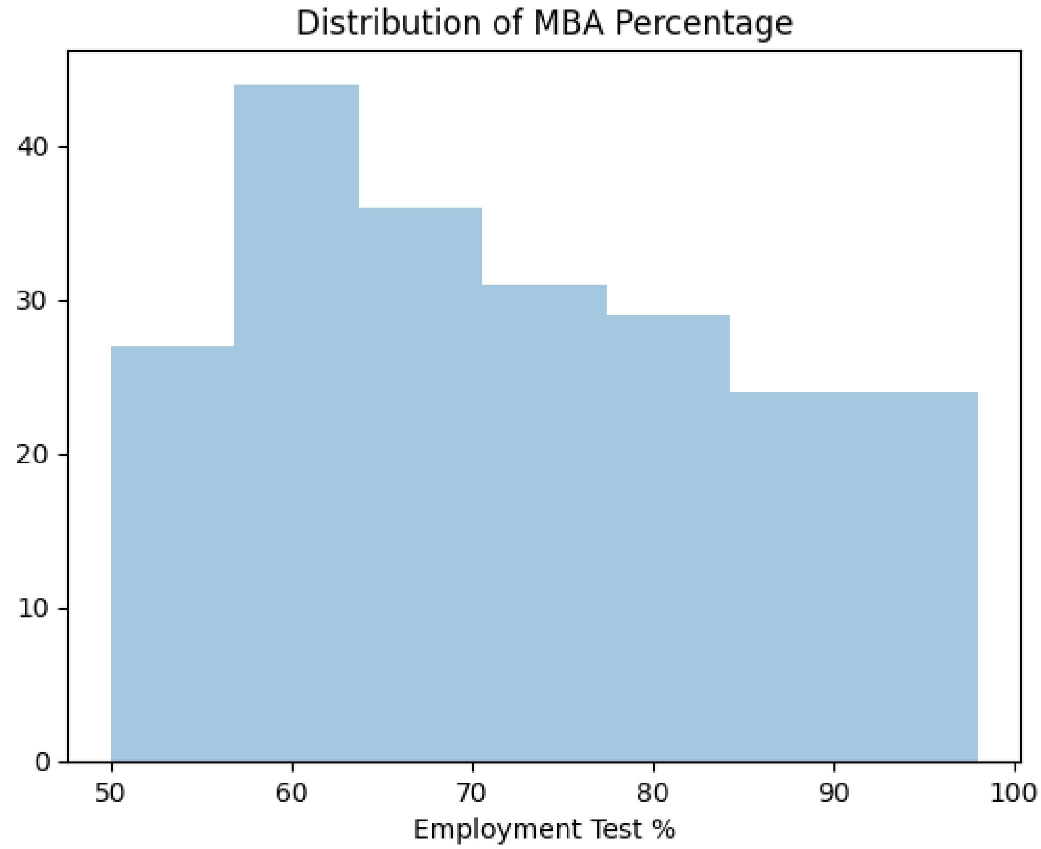
[43]:
```
## . Employment Test Percentage"
```

[45]:
```
sns.distplot(data['etest_p'], kde=False)
plt.title('Distribution of MBA Percentage')
plt.xlabel('Employment Test %')
```

```
<ipython-input-45-fb84975802b2>:1: UserWarning:

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

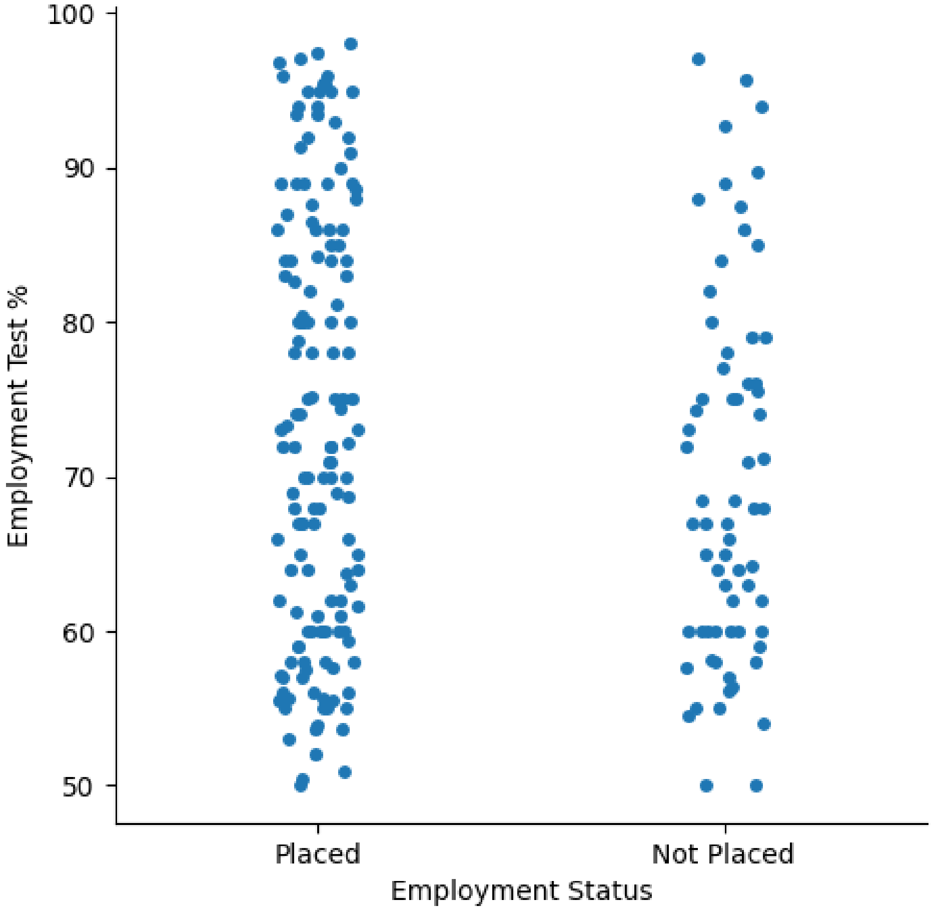
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

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

  sns.distplot(data['etest_p'], kde=False)
```

[45]:
```
Text(0.5, 0, 'Employment Test %')
```



[46]:
```python
sns.catplot(y='etest_p', x='status', data=data)
plt.xlabel('Employment Status')
plt.ylabel('Employment Test %')
```

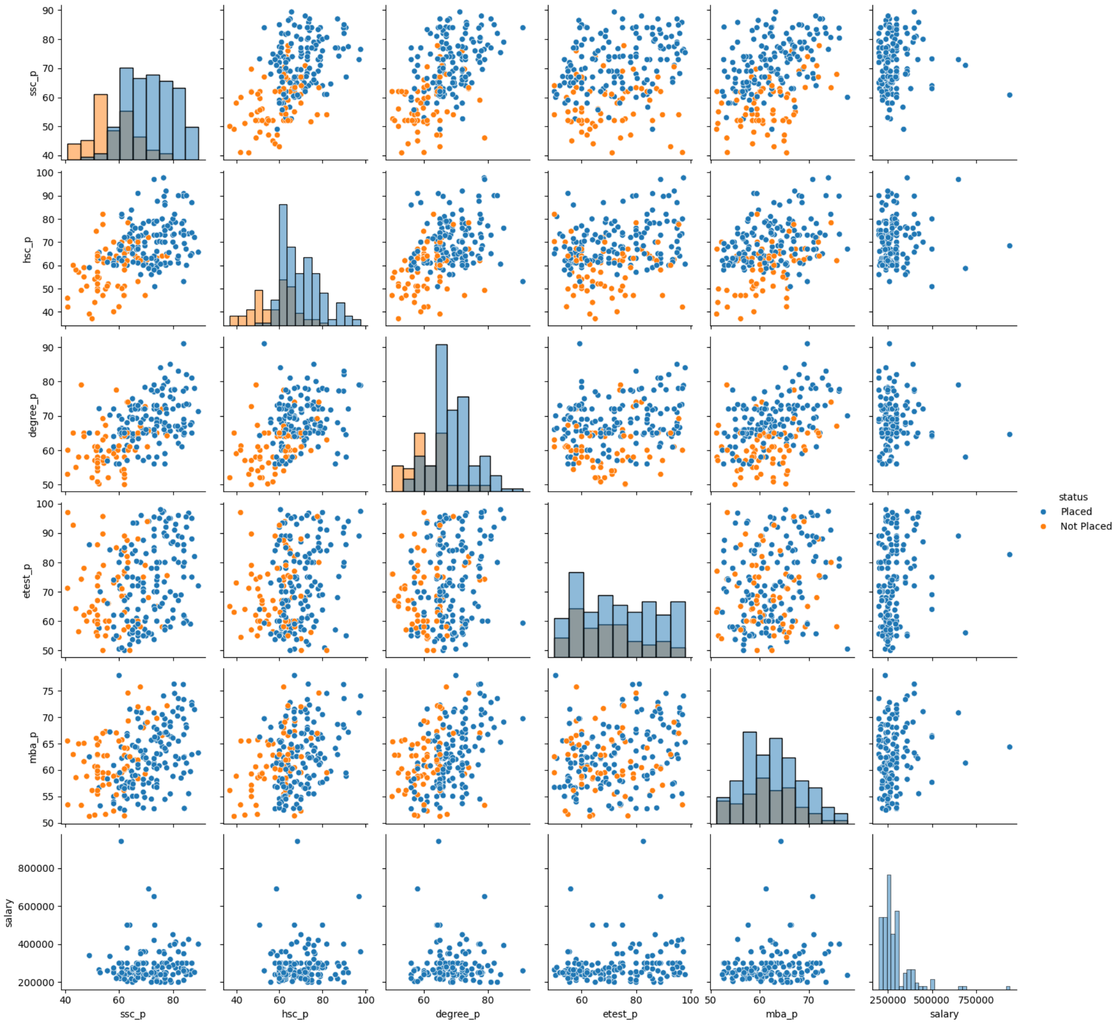
[46]:
```
Text(30.570061728395068, 0.5, 'Employment Test %')
```



[47]:
```
## Corelation between Features
```

[48]:
```
sns.pairplot(data=data[['ssc_p','hsc_p','degree_p', 'etest_p','mba_p','salary',␣
↪'status']], hue="status", diag_kind='hist')
```

[48]:
```
<seaborn.axisgrid.PairGrid at 0x7dc3d99a4c50>
```



```
[ ]: #Feature mapping
```

```
[49]: #Let's drop all unwanted columns as menstioned in above section.

      SSC Board
      HSC Board
      HSC Specialisation
      Degree Type
      Salary
```

```
  File "<ipython-input-49-7a0962a8de39>", line 3
    SSC Board
        ^

SyntaxError: invalid syntax
```

[50]:
```
data.drop(['ssc_b','hsc_b', 'hsc_s', 'degree_t', 'salary'], axis=1,␣
 ↪inplace=True)
```

[51]:
```
Let's map categorical feature to numeric one. Categorical features:

Gender : Gender feature have male and female values. I am going to map 0 for␣
 ↪male and 1 for female.
Work Experience : Work Experience feature have Yes and No values. I am going to␣
 ↪map 0 for No and 1 for Yes.
Status : Status feature have Not Placed and Placed values. Again for this␣
 ↪features I am mapping 0 for not placed and 1 for placed values.
Specialisation : Specialisation feature have two values Mkt&HR and Mkt&Fin. I␣
 ↪am going to map 0 to Mkt&HR and 1 to Mkt&Fin.
```

```
  File "<ipython-input-51-1a8a69a2b16c>", line 1
    Let's map categorical feature to numeric one. Categorical features:
       ^
SyntaxError: unterminated string literal (detected at line 1)
```

[52]:
```
data["gender"] = data.gender.map({"M":0,"F":1})
data["workex"] = data.workex.map({"No":0, "Yes":1})
data["status"] = data.status.map({"Not Placed":0, "Placed":1})
data["specialisation"] = data.specialisation.map({"Mkt&HR":0, "Mkt&Fin":1})
```

[53]:
```
data.columns
```

[53]:
```
Index(['gender', 'ssc_p', 'hsc_p', 'degree_p', 'workex', 'etest_p',
       'specialisation', 'mba_p', 'status'],
      dtype='object')
```

[54]:
```
data.head()
```

[54]:
```
   gender  ssc_p  hsc_p  degree_p  workex  etest_p  specialisation  mba_p  \
0       0  67.00  91.00     58.00       0     55.0               0  58.80
1       0  79.33  78.33     77.48       1     86.5               1  66.28
2       0  65.00  68.00     64.00       0     75.0               1  57.80
3       0  56.00  52.00     52.00       0     66.0               0  59.43
4       0  85.80  73.60     73.30       0     96.8               1  55.50

   status
0       1
1       1
2       1
3       0
4       1
```

```
[ ]:
```