```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        import seaborn as sns
        %matplotlib inline
        import numpy as np
```

```
In [2]: df = pd.read_csv('Data.csv')
        df = df.drop(['sl_no','ssc_b','hsc_b'], axis=1)
        df
```

Out[2]:

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	M	56.00	52.00	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	M	85.80	73.60	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0
...	...	...	...	...	...	...	...	...	...	...	...	...
210	M	80.60	82.00	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0
211	M	58.00	60.00	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0
212	M	67.00	67.00	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0
213	F	74.00	66.00	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0
214	M	62.00	58.00	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN

215 rows × 12 columns

```
In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   gender          215 non-null    object 
 1   ssc_p           215 non-null    float64
 2   hsc_p           215 non-null    float64
 3   hsc_s           215 non-null    object 
 4   degree_p        215 non-null    float64
 5   degree_t        215 non-null    object 
 6   workex          215 non-null    object 
 7   etest_p         215 non-null    float64
 8   specialisation  215 non-null    object 
 9   mba_p           215 non-null    float64
 10  status          215 non-null    object 
 11  salary          148 non-null    float64
dtypes: float64(6), object(6)
memory usage: 20.3+ KB
```

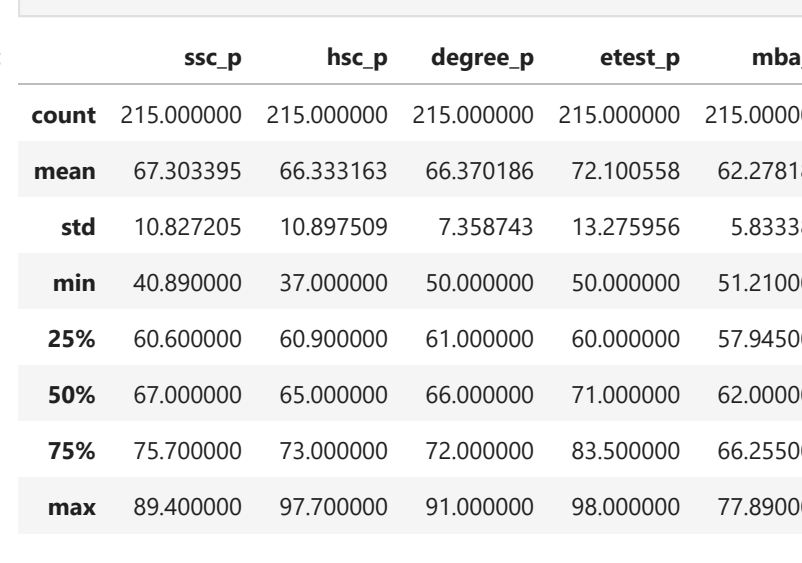
```
In [4]: df1 = df.groupby('status').get_group('Not Placed')
        df1.info()
        #All null salary values is connected to the status, if your status is not placed, your salary is null.

<class 'pandas.core.frame.DataFrame'>
Int64Index: 67 entries, 3 to 214
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   gender          67 non-null     object 
 1   ssc_p           67 non-null     float64
 2   hsc_p           67 non-null     float64
 3   hsc_s           67 non-null     object 
 4   degree_p        67 non-null     float64
 5   degree_t        67 non-null     object 
 6   workex          67 non-null     object 
 7   etest_p         67 non-null     float64
 8   specialisation  67 non-null     object 
 9   mba_p           67 non-null     float64
 10  status          67 non-null     object 
 11  salary          0 non-null      float64
dtypes: float64(6), object(6)
memory usage: 6.8+ KB
```

```
In [5]: df.describe()
```

Out[5]:

	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
count	215.000000	215.000000	215.000000	215.000000	215.000000	148.000000
mean	67.303395	66.333163	66.370186	72.100558	62.278186	288655.405405
std	10.827205	10.897509	7.358743	13.275956	5.833385	93457.452420
min	40.890000	37.000000	50.000000	50.000000	51.210000	200000.000000
25%	60.600000	60.900000	61.000000	60.000000	57.945000	240000.000000
50%	67.000000	65.000000	66.000000	71.000000	62.000000	265000.000000
75%	75.700000	73.000000	72.000000	83.500000	66.255000	300000.000000
max	89.400000	97.700000	91.000000	98.000000	77.890000	940000.000000

```
In [6]: sns.distplot(df.ssc_p,kde=False)

C:\Program Files\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Out[6]: <AxesSubplot:xlabel='ssc_p'>
```




```
In [7]: bins=[0,55,70,80,100]
        group=['Low','Average','High','Very High']
        df['ssc_p_withgroups']=pd.cut(df['ssc_p'],bins,labels=group)
```

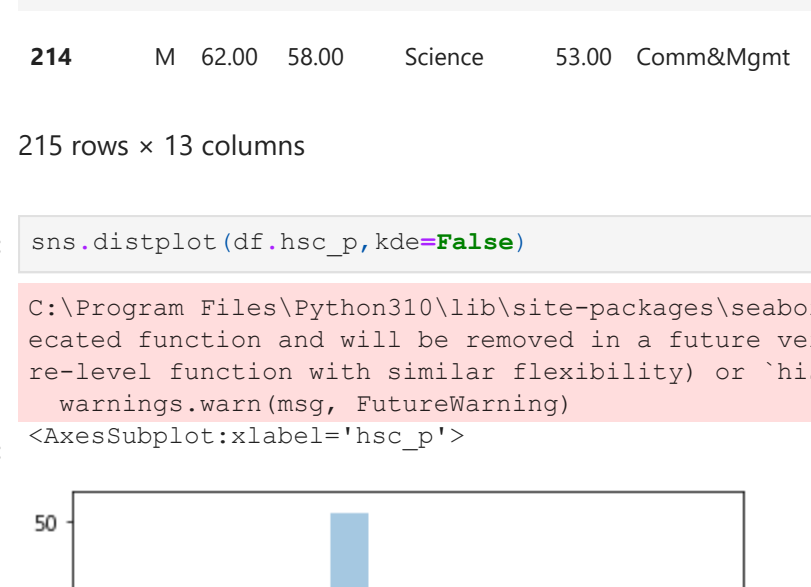
```
In [8]: df
```

Out[8]:

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary	ssc_p_withgroups
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0	Average
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0	High
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0	Average
3	M	56.00	52.00	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN	Average
4	M	85.80	73.60	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0	Very High
...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	M	80.60	82.00	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0	Very High
211	M	58.00	60.00	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0	Average
212	M	67.00	67.00	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0	Average
213	F	74.00	66.00	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0	High
214	M	62.00	58.00	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN	Average

215 rows × 13 columns

```
In [9]: sns.distplot(df.hsc_p,kde=False)

C:\Program Files\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Out[9]: <AxesSubplot:xlabel='hsc_p'>
```




```
In [10]: bins=[0,55,70,85,100]
         group=['Low','Average','High','Very High']
         df['hsc_p_withgroups']=pd.cut(df['hsc_p'],bins,labels=group)
```

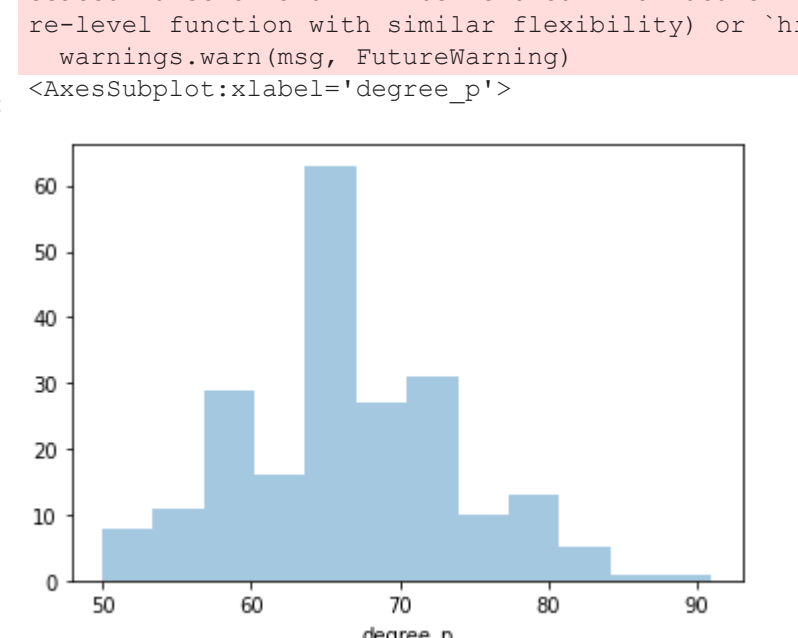
```
In [11]: df
```

Out[11]:

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary	ssc_p_withgroups	hs
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0	Average	
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0	High	
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0	Average	
3	M	56.00	52.00	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN	Average	
4	M	85.80	73.60	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0	Very High	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210	M	80.60	82.00	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0	Very High	
211	M	58.00	60.00	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0	Average	
212	M	67.00	67.00	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0	Average	
213	F	74.00	66.00	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0	High	
214	M	62.00	58.00	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN	Average	

215 rows × 14 columns

```
In [12]: sns.distplot(df.degree_p,kde=False)

C:\Program Files\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Out[12]: <AxesSubplot:xlabel='degree_p'>
```




```
In [13]: bins=[0,60,70,80,100]
         group=['Low','Average','High','Very High']
         df['degree_p_withgroups']=pd.cut(df['degree_p'],bins,labels=group)
```

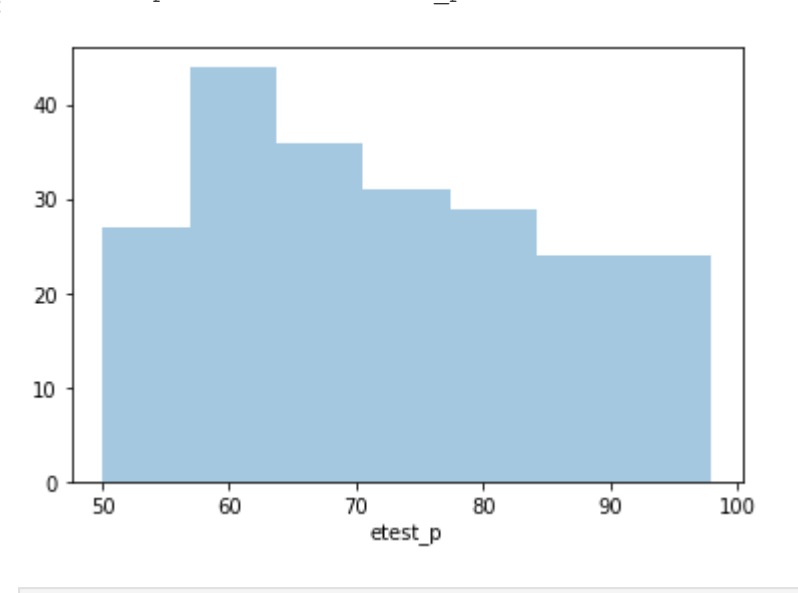
```
In [14]: df
```

Out[14]:

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary	ssc_p_withgroups	hs
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0	Average	
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0	High	
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0	Average	
3	M	56.00	52.00	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN	Average	
4	M	85.80	73.60	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0	Very High	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210	M	80.60	82.00	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0	Very High	
211	M	58.00	60.00	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0	Average	
212	M	67.00	67.00	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0	Average	
213	F	74.00	66.00	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0	High	
214	M	62.00	58.00	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN	Average	

215 rows × 15 columns

```
In [15]: sns.distplot(df.etest_p,kde=False)

C:\Program Files\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Out[15]: <AxesSubplot:xlabel='etest_p'>
```




```
In [16]: bins=[0,60,70,90,100]
         group=['Low','Average','High','Very High']
         df['etest_p_withgroups']=pd.cut(df['etest_p'],bins,labels=group)
```

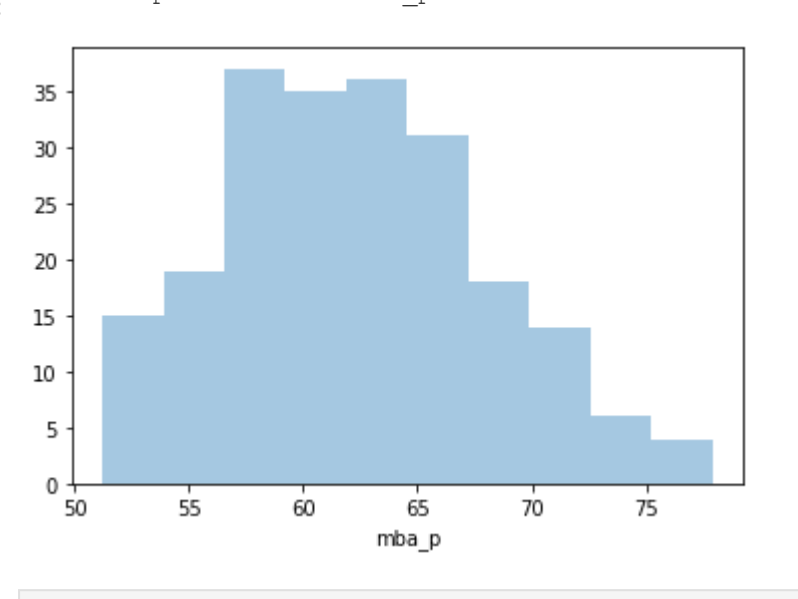
```
In [17]: df
```

Out[17]:

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary	ssc_p_withgroups	hs
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0	Average	
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0	High	
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0	Average	
3	M	56.00	52.00	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN	Average	
4	M	85.80	73.60	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0	Very High	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210	M	80.60	82.00	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0	Very High	
211	M	58.00	60.00	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0	Average	
212	M	67.00	67.00	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0	Average	
213	F	74.00	66.00	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0	High	
214	M	62.00	58.00	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN	Average	

215 rows × 16 columns

```
In [18]: sns.distplot(df.mba_p,kde=False)

C:\Program Files\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Out[18]: <AxesSubplot:xlabel='mba_p'>
```




```
In [19]: bins=[0,57,65,70,100]
         group=['Low','Average','High','Very High']
         df['mba_p_withgroups']=pd.cut(df['mba_p'],bins,labels=group)
```

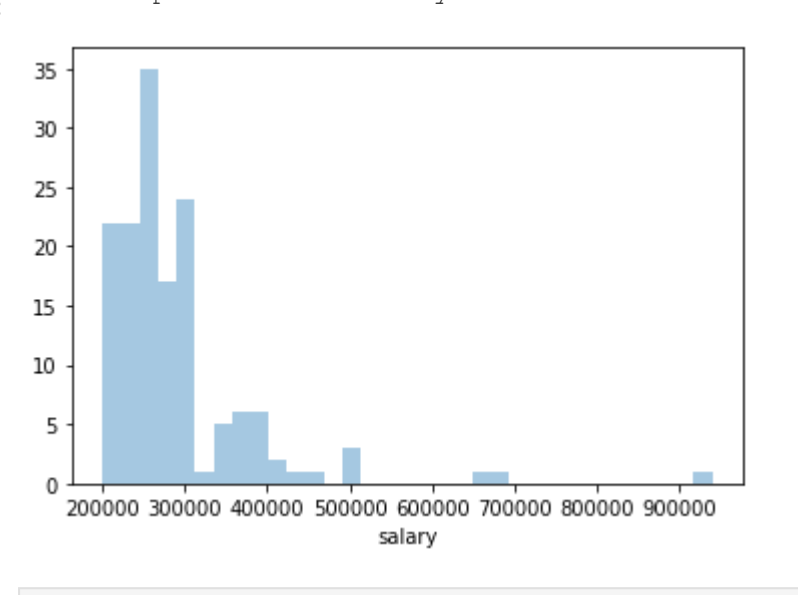
```
In [20]: df
```

Out[20]:

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary	ssc_p_withgroups	hs
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0	Average	
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0	High	
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0	Average	
3	M	56.00	52.00	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN	Average	
4	M	85.80	73.60	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0	Very High	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210	M	80.60	82.00	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0	Very High	
211	M	58.00	60.00	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0	Average	
212	M	67.00	67.00	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0	Average	
213	F	74.00	66.00	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0	High	
214	M	62.00	58.00	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN	Average	

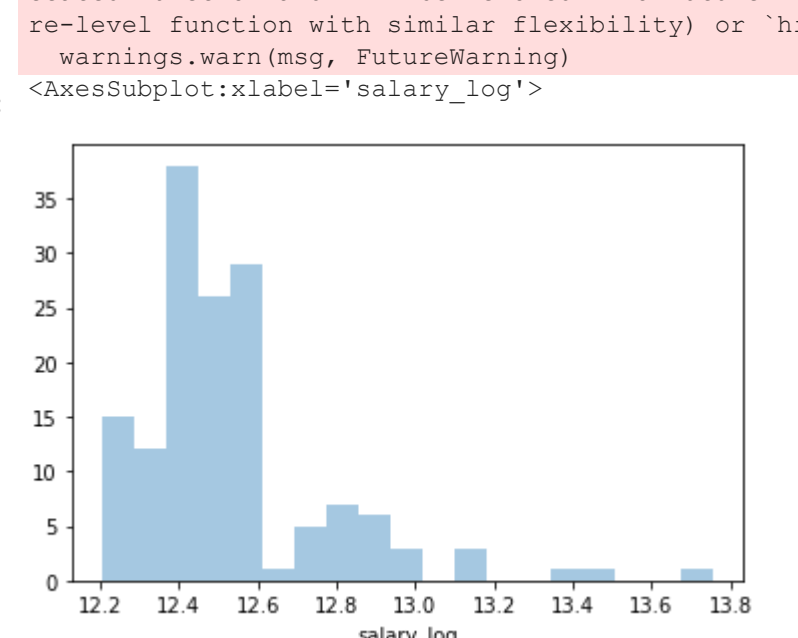
215 rows × 17 columns

```
In [21]: sns.distplot(df.salary,kde=False)

C:\Program Files\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Out[21]: <AxesSubplot:xlabel='salary'>
```




```
In [22]: df['salary_log'] =np.log(df['salary'])
```

```
In [23]: sns.distplot(df.salary_log,kde=False)

C:\Program Files\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Out[23]: <AxesSubplot:xlabel='salary_log'>
```




```
In [24]: df["salary_log"]=df["salary_log"].fillna(0, inplace = False)
         bins=[-1,12,12.38,12.61,12.99,100]
         group=['Not Hired','Low','Average','High','Very High']
         df['salary_with_groups']=pd.cut(df['salary_log'],bins,labels=group)
```

- Not hired = 0
- Low = 200000 - 240000
- Average = 240000 - 300000
- High = 300000 - 440000
- Very High = More Than 440000

```
In [25]: df['salary_with_groups'].value_counts()

Out[25]: Average      69
         Not Hired    67
         High         43
         Low          29
         Very High     7
         Name: salary_with_groups, dtype: int64
```

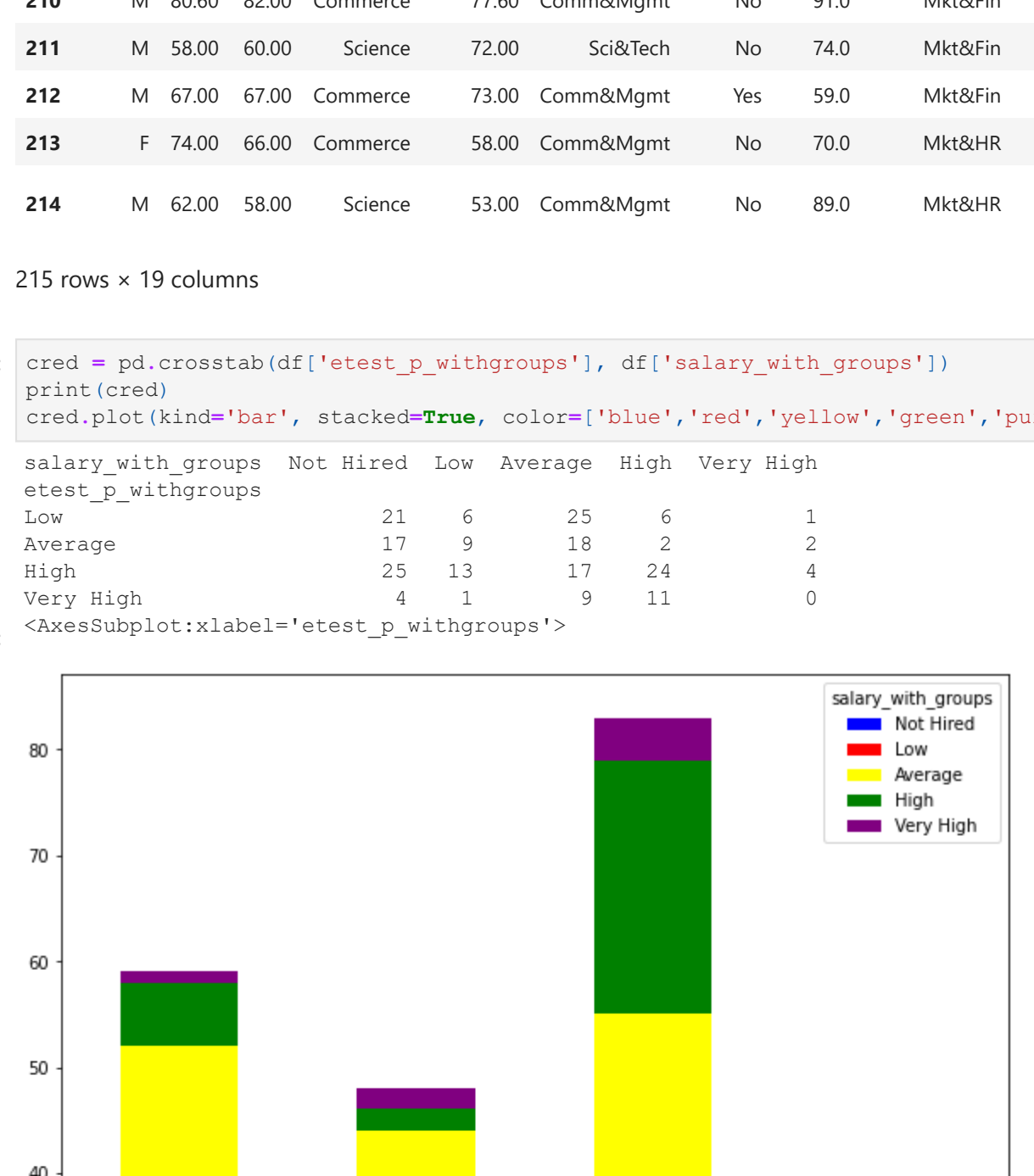
```
In [26]: df
```

Out[26]:

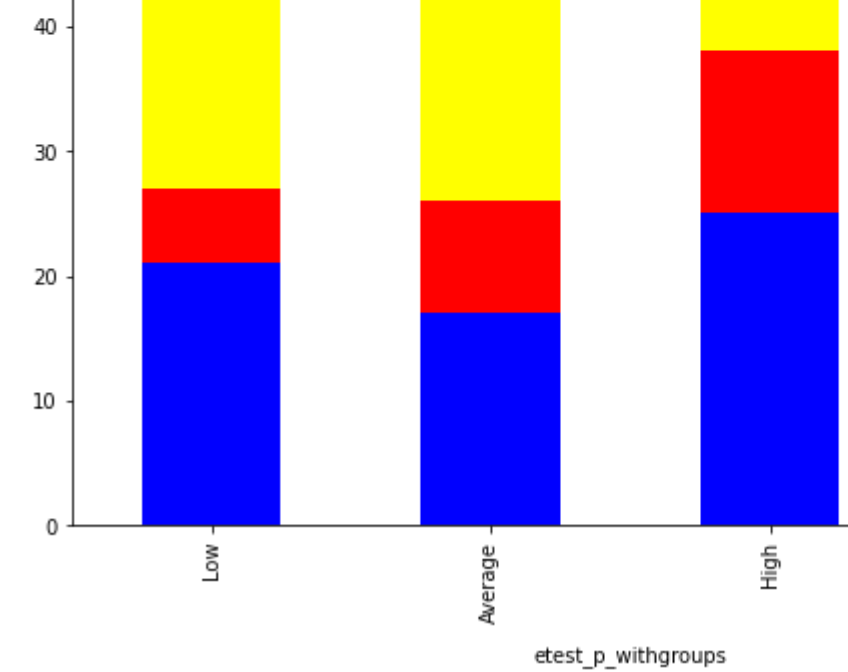
	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary	ssc_p_withgroups	hs
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0	Average	
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0	High	
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0	Average	
3	M	56.00	52.00	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN	Average	
4	M	85.80	73.60	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0	Very High	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210	M	80.60	82.00	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0	Very High	
211	M	58.00	60.00	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0	Average	
212	M	67.00	67.00	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0	Average	
213	F	74.00	66.00	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0	High	
214	M	62.00	58.00	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN	Average	

215 rows × 19 columns

```
In [27]: cred = pd.crosstab(df['etest_p_withgroups'], df['salary_with_groups'])
         print(cred)
         cred.plot(kind='bar', stacked=True, color=['blue','red','yellow','green','purple'],figsize=(10,10), grid=False)

salary_with_groups  Not Hired  Low  Average  High  Very High
etest_p_withgroups                                          
Low                        21    6       25     6          1
Average                    17    9       18     2          2
High                       25   13       17    24          4
Very High                   4    1        9    11          0
Out[27]: <AxesSubplot:xlabel='etest_p_withgroups'>
```



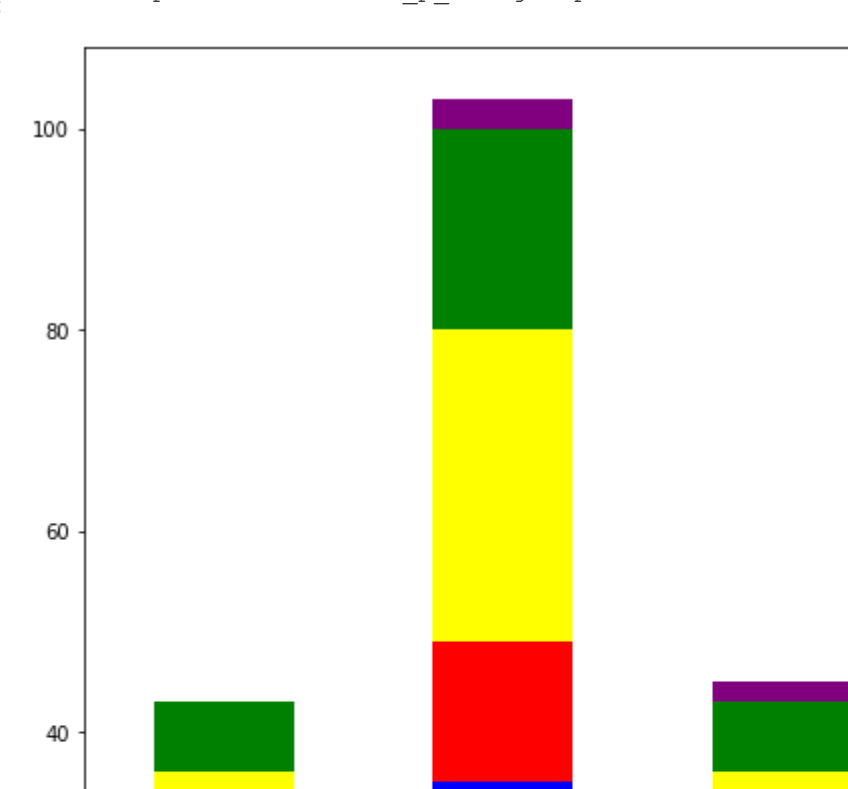

```
In [28]: sns.boxplot(x = 'specialisation', y = 'salary', data = df)

Out[28]: <AxesSubplot:xlabel='specialisation', ylabel='salary'>
```




```
In [29]: cred = pd.crosstab(df['specialisation'], df['salary_with_groups'])
         print(cred)

salary_with_groups  Not Hired  Low  Average  High  Very High
specialisation                                              
Mkt&Fin                    25   18       41    30          6
Mkt&HR                     42   11       28    13          1
```

```
In [30]: cred = pd.crosstab(df['mba_p_withgroups'], df['salary_with_groups'])
         print(cred)
         cred.plot(kind='bar', stacked=True, color=['blue','red','yellow','green','purple'],figsize=(10,10), grid=False)

salary_with_groups  Not Hired  Low  Average  High  Very High
mba_p_withgroups                                            
Low                        13    5       18     7          0
Average                    35   14       31    20          3
High                       14    7       15     7          2
Very High                   5    3        5     9          2
Out[30]: <AxesSubplot:xlabel='mba_p_withgroups'>
```


```
In [42]: sns.boxplot(x = 'workex', y = 'salary', data = df)

Out[42]: <AxesSubplot:xlabel='workex', ylabel='salary'>
```




```
In [43]: cred = pd.crosstab(df['workex'], df['salary_with_groups'])
         print(cred)

salary_with_groups  Not Hired  Low  Average  High  Very High
workex                                                      
No                         57   19       39    24          2
Yes                        10   10       30    19          5
```

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:
```

```
In [32]: train = df.drop(['salary','salary_log','ssc_p','hsc_p','degree_p','etest_p','mba_p','salary','salary_log'],axis
         x = train.drop('salary_with_groups',1)
         y = train.salary_with_groups

C:\Users\Emir Tonoğlu\AppData\Local\Temp\ipykernel_4004\2536793472.py:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only.
  x = train.drop('salary_with_groups',1)
```

```
In [33]: x = pd.get_dummies(x)
```

```
In [34]: from sklearn.model_selection import train_test_split
         x_train, x_cv, y_train, y_cv = train_test_split(x,y, test_size=0.3)
```

```
In [35]: from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import accuracy_score
         model = LogisticRegression(random_state = 1)
         model.fit(x_train, y_train)

Out[35]: LogisticRegression(random_state=1)
```

```
In [36]: pred_cv = model.predict(x_cv)
         accuracy_score(y_cv,pred_cv)

Out[36]: 0.6
```

```
In [37]: from sklearn.metrics import confusion_matrix
         cnf=confusion_matrix(y_cv, pred_cv)
         cnf

Out[37]: array([[14,  4,  2,  0,  0],
                [ 6,  5,  3,  0,  0],
                [ 4,  1,  1,  0,  0],
                [ 0,  0,  0, 19,  0],
                [ 4,  1,  1,  0,  0]], dtype=int64)
```

```
In [38]: from sklearn import tree
         model = tree.DecisionTreeClassifier(random_state=1)
         model.fit(x_train,y_train)

Out[38]: DecisionTreeClassifier(random_state=1)
```

```
In [39]: pred_cv = model.predict(x_cv)
         accuracy_score(y_cv,pred_cv)

Out[39]: 0.5692307692307692
```

```
In [40]: from sklearn.ensemble import RandomForestClassifier
         model = RandomForestClassifier(random_state=1, max_depth=10)
         model.fit(x_train,y_train)

Out[40]: RandomForestClassifier(max_depth=10, random_state=1)
```

```
In [41]: pred_cv = model.predict(x_cv)
         accuracy_score(y_cv,pred_cv)

Out[41]: 0.5384615384615384
```

```
In [ ]:
```