

Scatter plots

```
In [38]: import pandas as pd  
import seaborn as sns  
import numpy as np
```

```
In [2]: df = pd.read_csv("dm_office_sales.csv")
```

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

```
Out[3]:
```

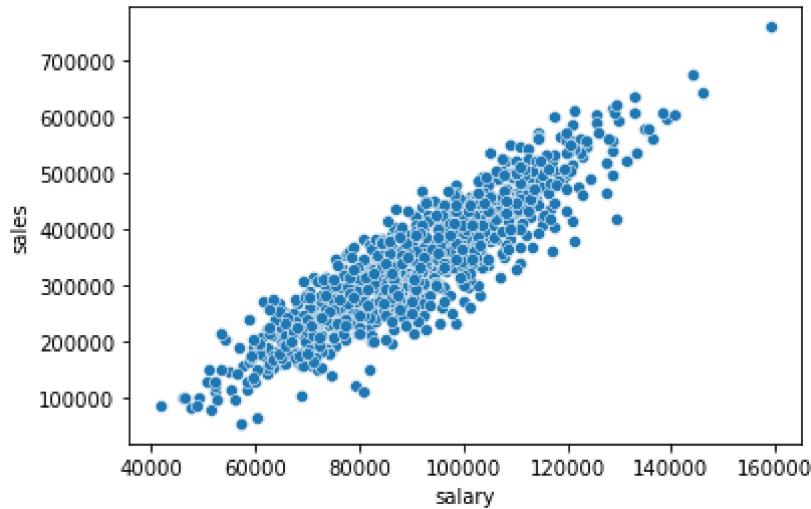
	division	level of education	training level	work experience	salary	sales
0	printers	some college	2	6	91684	372302
1	printers	associate's degree	2	10	119679	495660
2	peripherals	high school	0	9	82045	320453
3	office supplies	associate's degree	2	5	92949	377148
4	office supplies	high school	1	5	71280	312802

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 6 columns):  
 #   Column           Non-Null Count  Dtype     
 ---    
 0   division        1000 non-null    object    
 1   level of education 1000 non-null    object    
 2   training level   1000 non-null    int64    
 3   work experience  1000 non-null    int64    
 4   salary          1000 non-null    int64    
 5   sales           1000 non-null    int64    
 dtypes: int64(4), object(2)  
 memory usage: 47.0+ KB
```

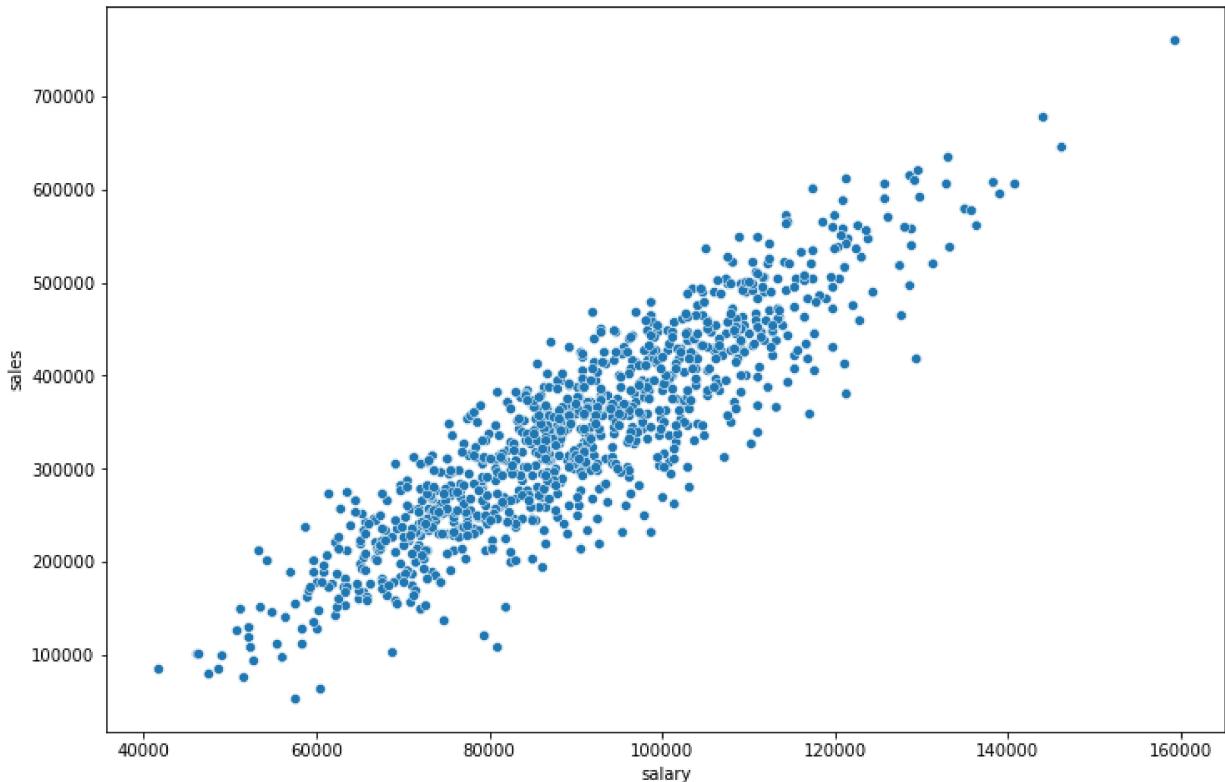
```
In [5]: sns.scatterplot(x='salary',y='sales',data=df)
```

```
Out[5]: <AxesSubplot:xlabel='salary', ylabel='sales'>
```

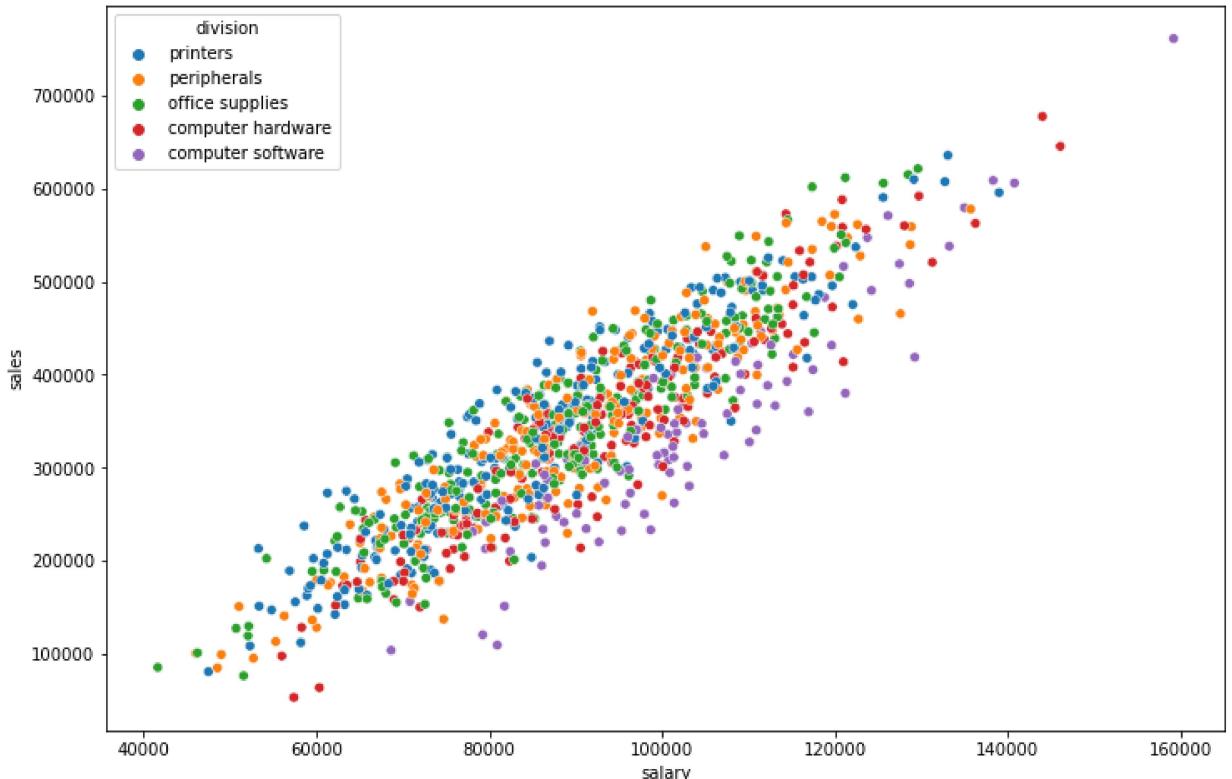


```
In [6]: import matplotlib.pyplot as plt
```

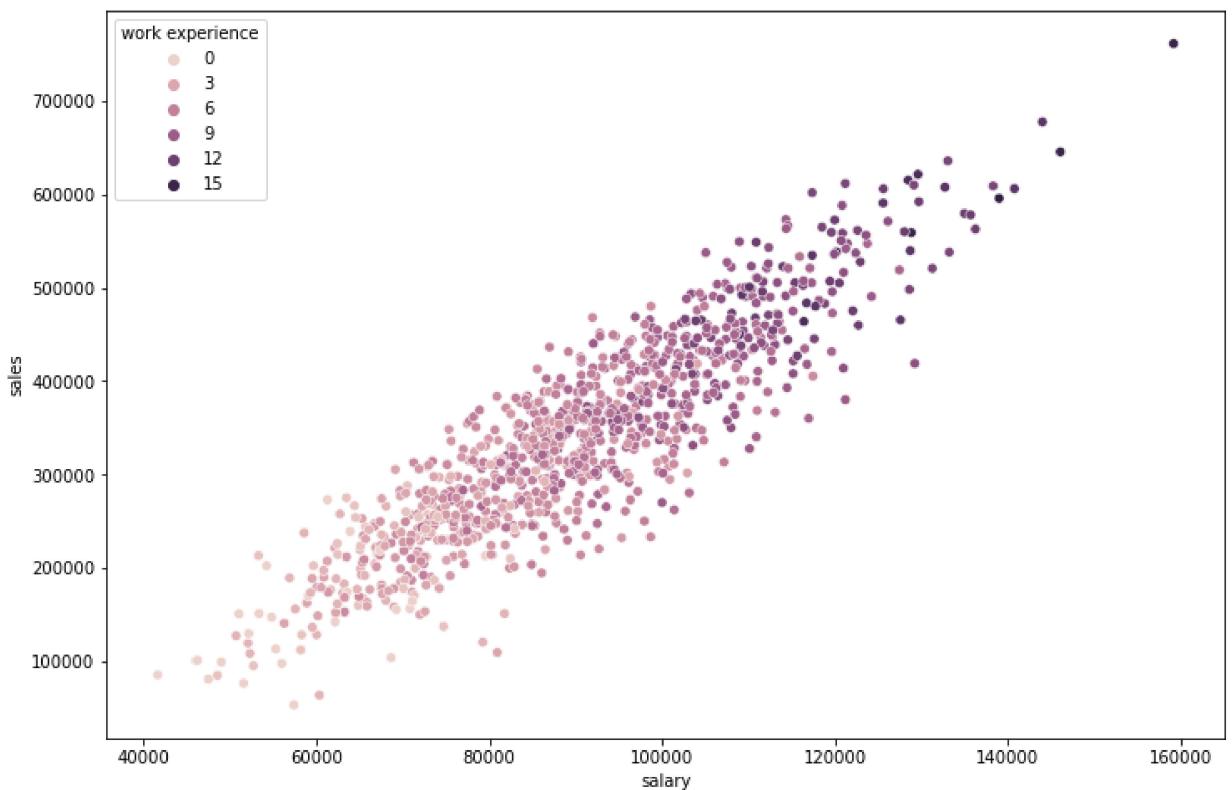
```
In [7]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',data=df)
plt.show()
```



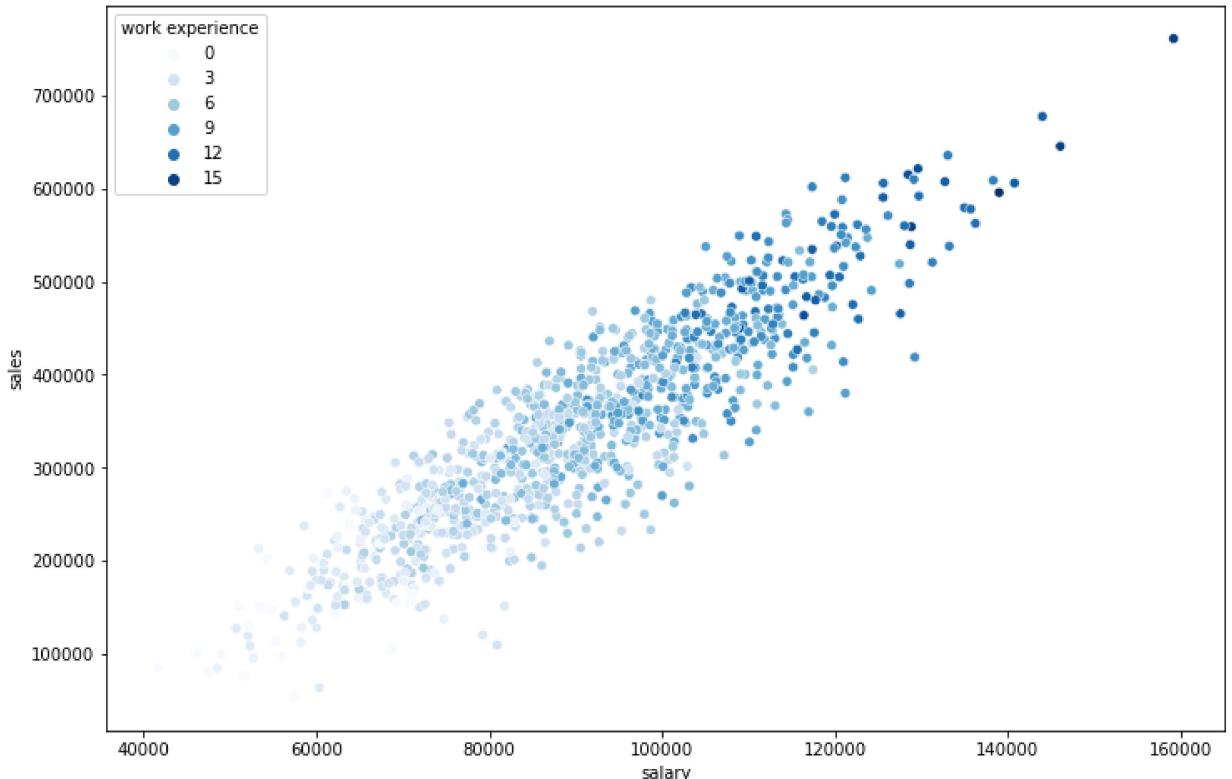
```
In [8]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',hue='division',data=df)
plt.show()
```



```
In [9]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',hue= 'work experience',data=df)
plt.show()
```

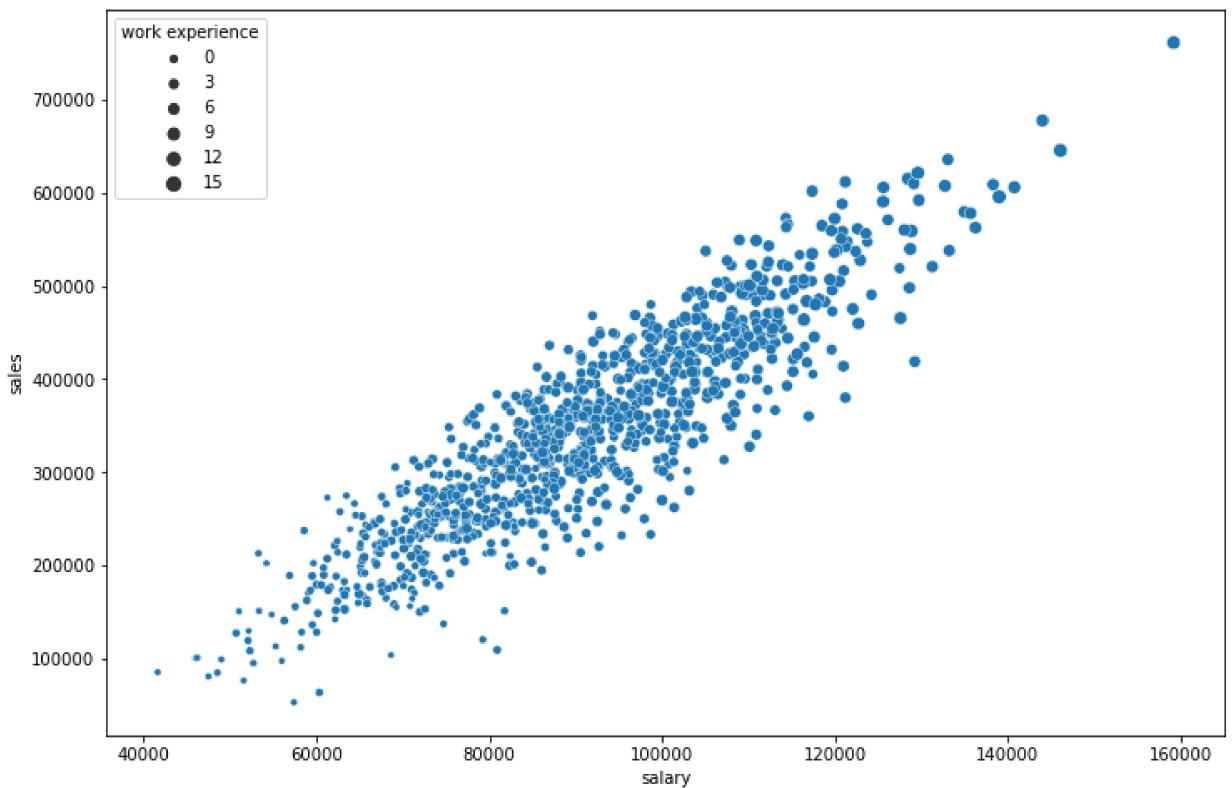


```
In [10]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',hue= 'work experience',data=df,palette='Blues')
plt.show()
```

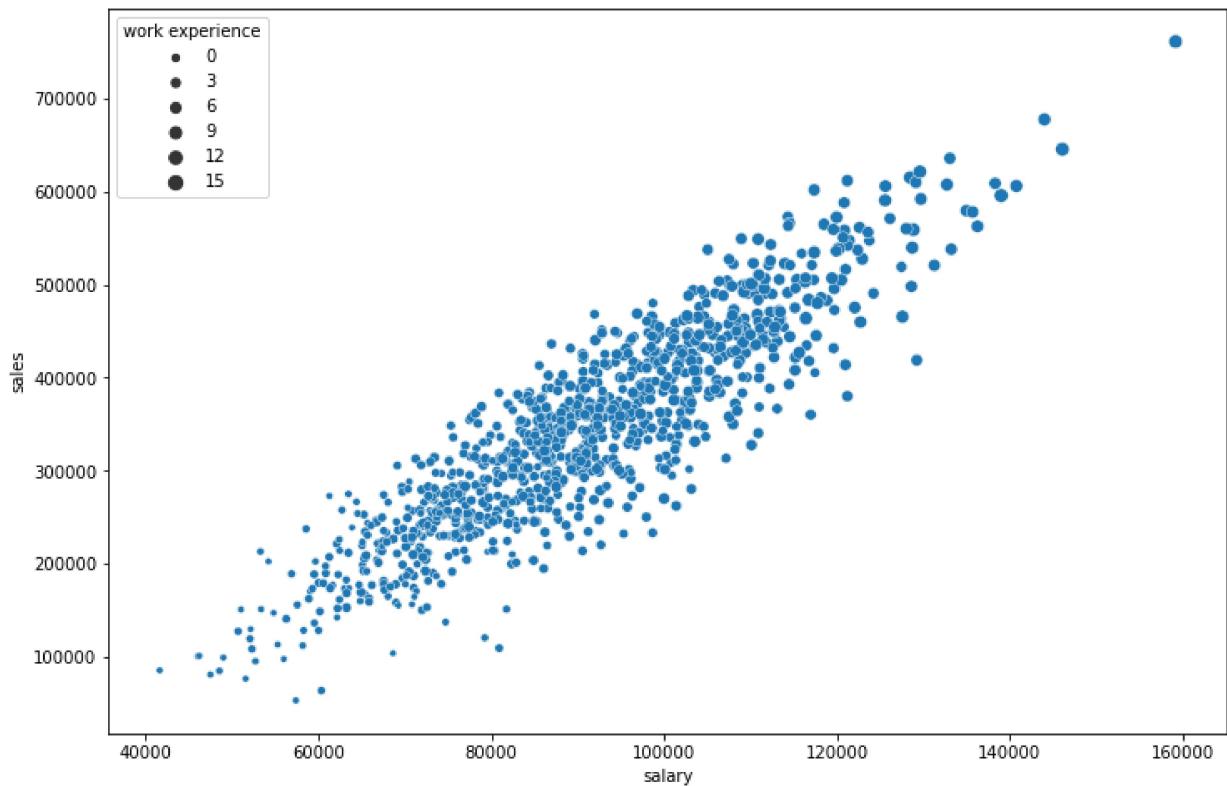


size()

```
In [11]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',size= 'work experience',data=df)
plt.show()
```

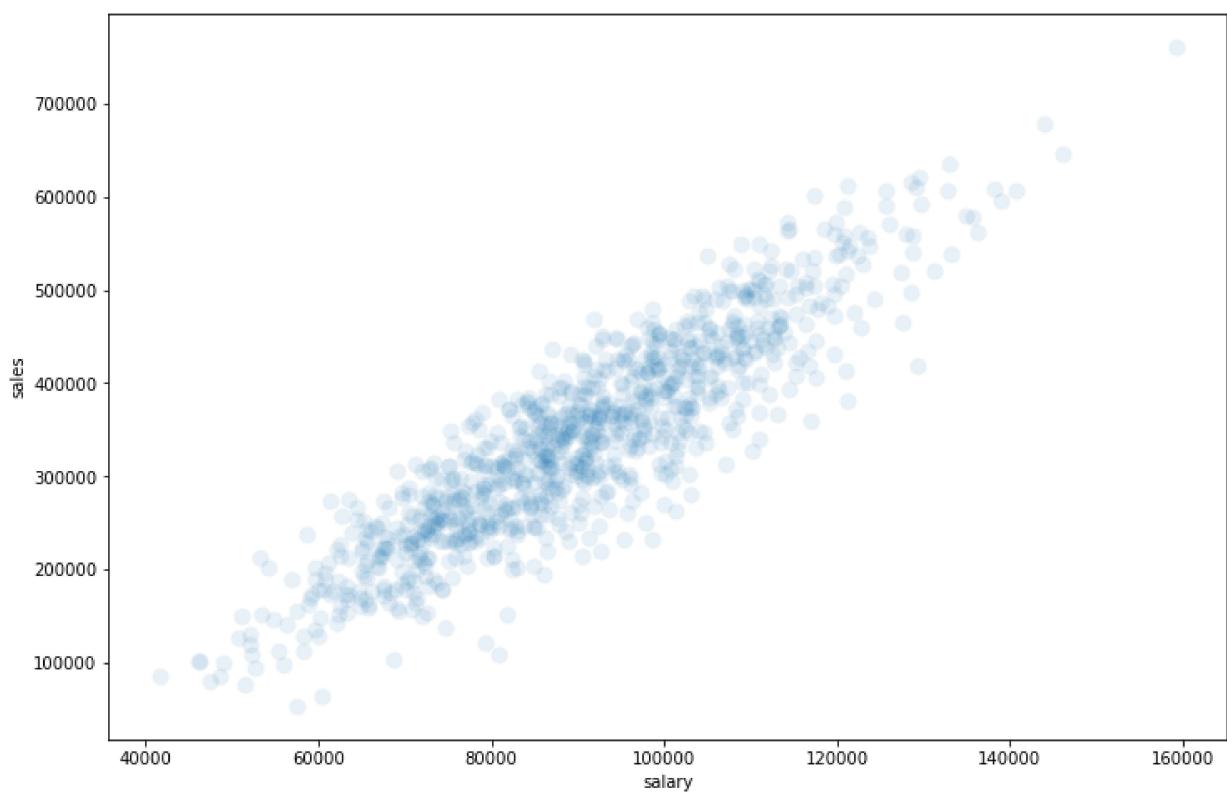


```
In [12]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',data=df,size='work experience')
plt.show()
```



```
In [13]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',data=df,s=100,linewidth=0.5,alpha=0.1 )
```

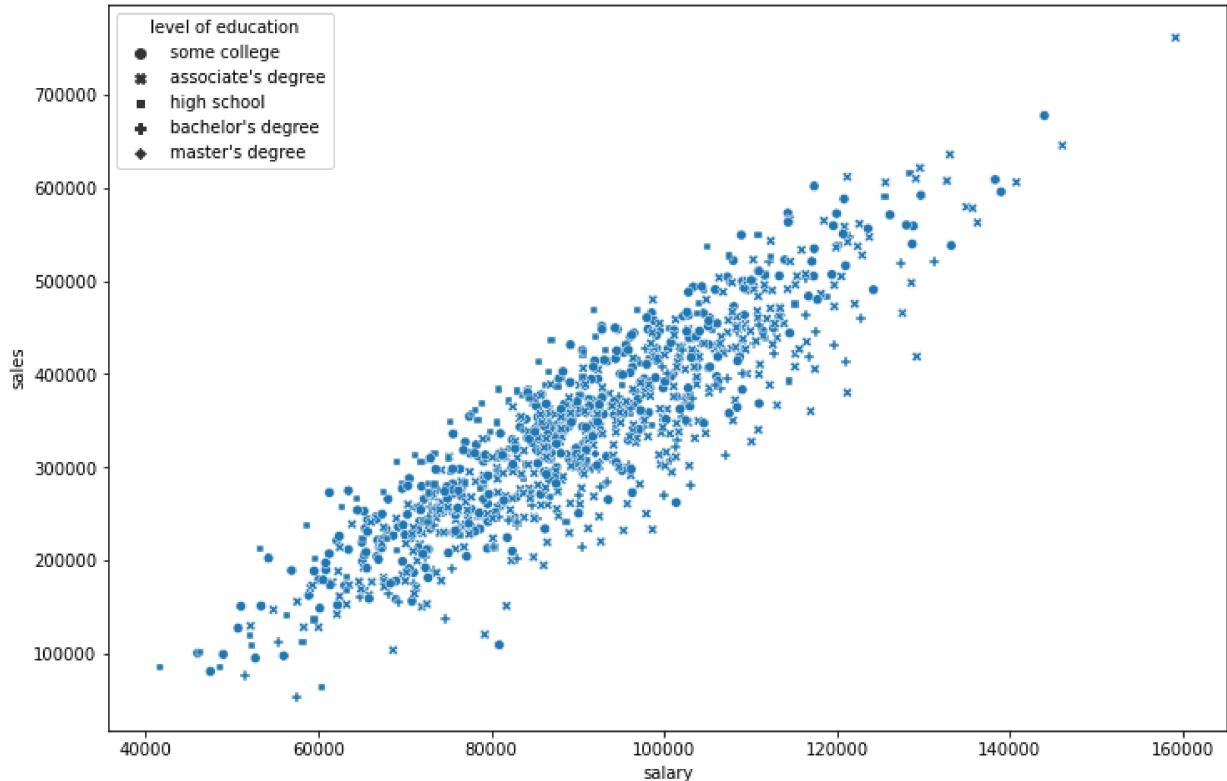
```
Out[13]: <AxesSubplot:xlabel='salary', ylabel='sales'>
```



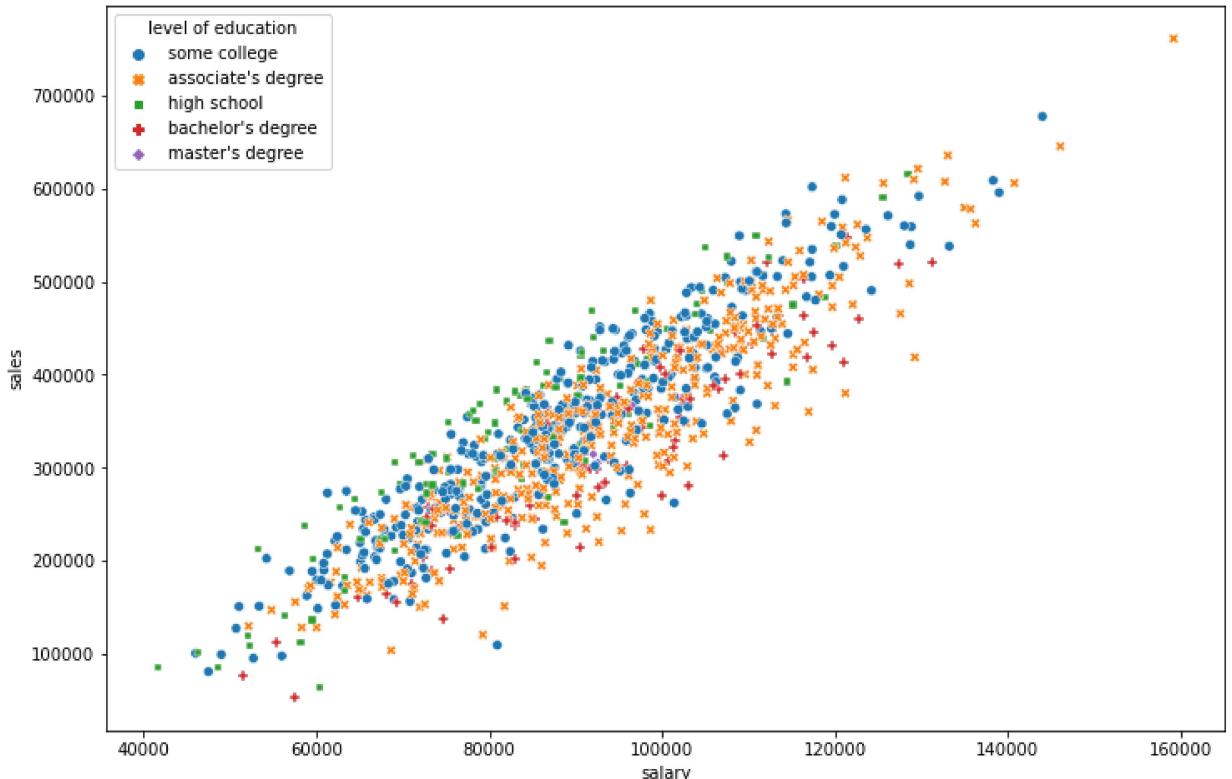
Style()

```
In [14]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',data=df,style='level of education')
```

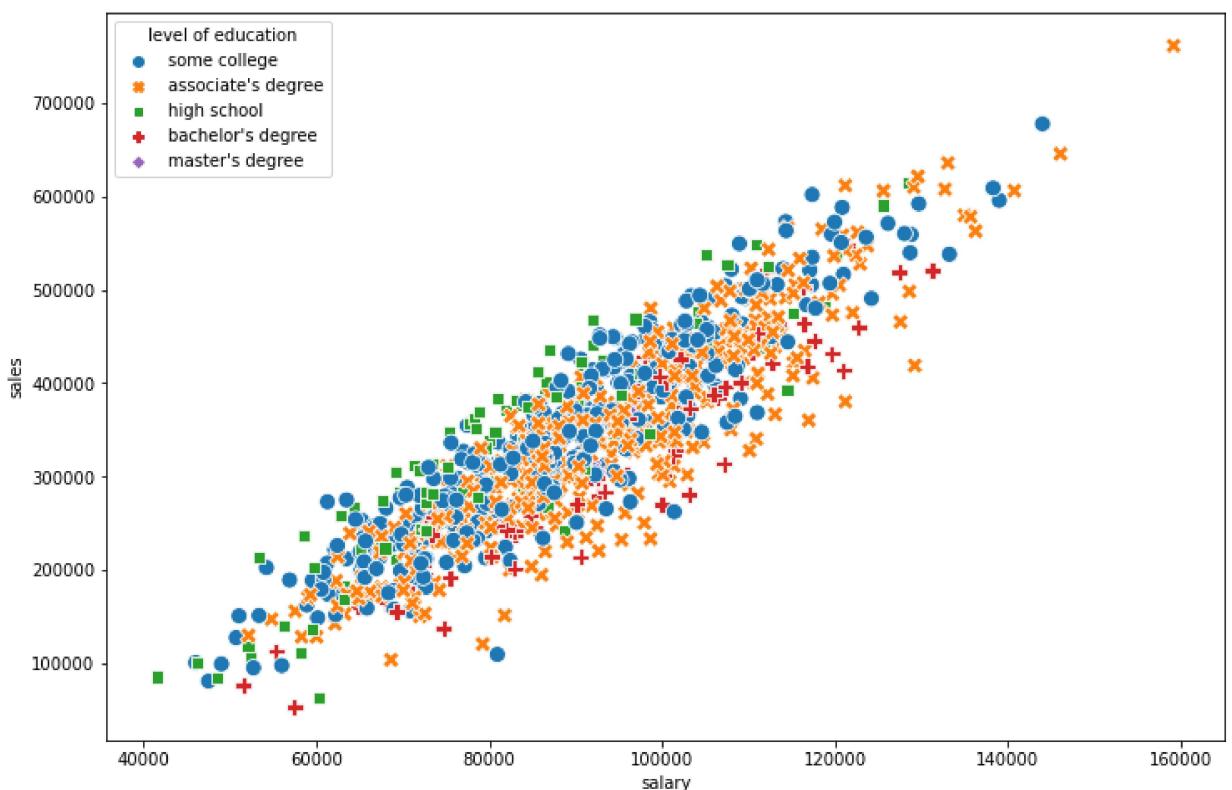
```
Out[14]: <AxesSubplot:xlabel='salary', ylabel='sales'>
```



```
In [15]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',hue= 'level of education',data=df,style='level of education')
plt.show()
```



```
In [16]: plt.figure(figsize=(12,8))
sns.scatterplot(x='salary',y='sales',hue= 'level of education',data=df,style='level of
plt.show()
```

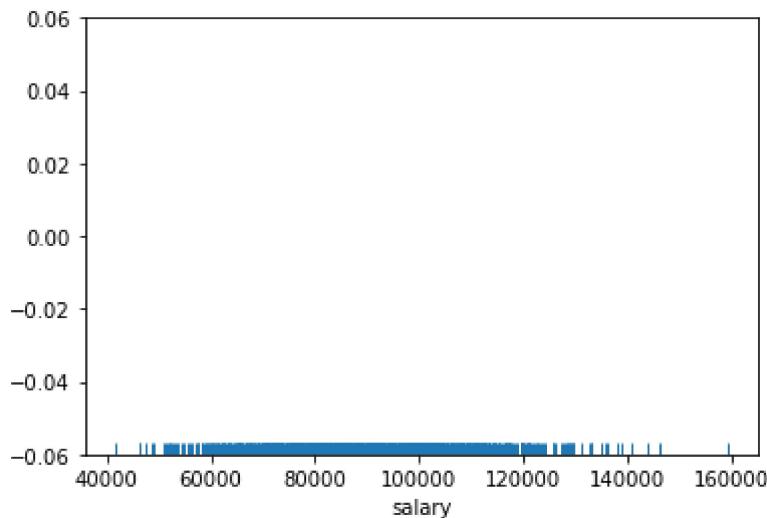


```
In [17]: import pandas as pd
```

Rugplot

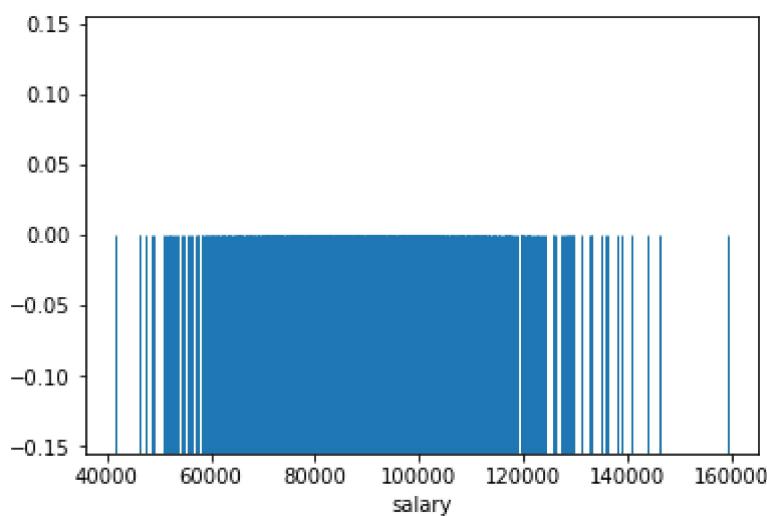
```
In [18]: sns.rugplot(x='salary', data=df)
```

```
Out[18]: <AxesSubplot:xlabel='salary'>
```



```
In [19]: sns.rugplot(x='salary', data=df, height=0.5)
```

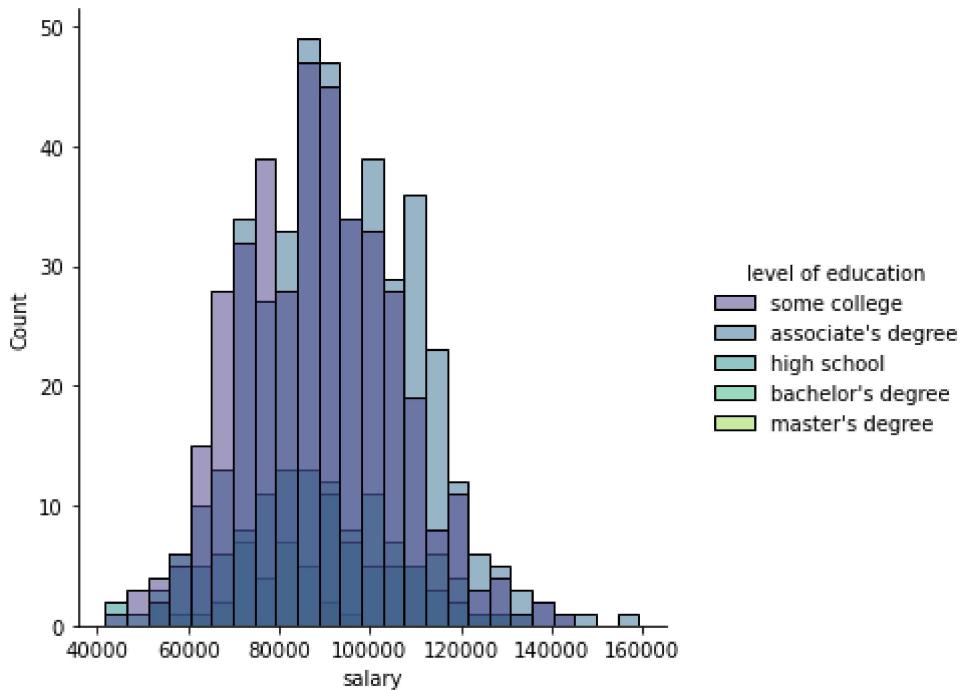
```
Out[19]: <AxesSubplot:xlabel='salary'>
```



distplot and histplot

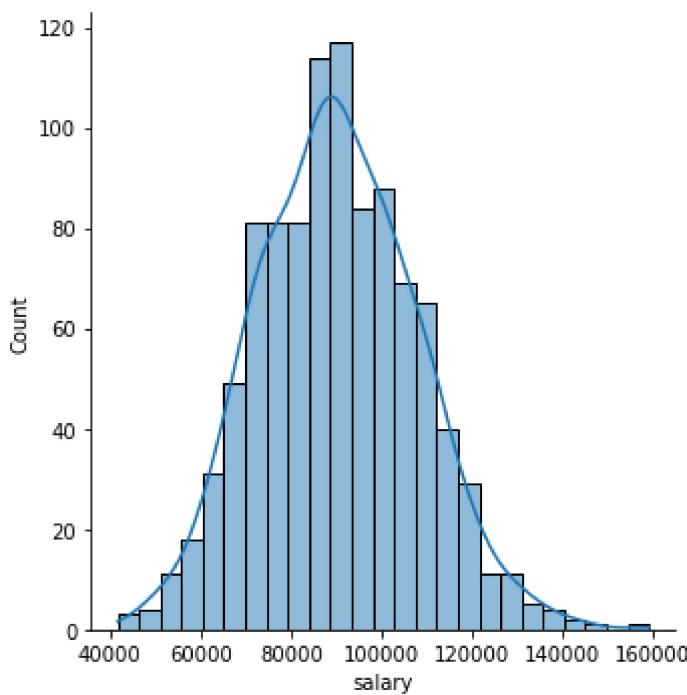
```
In [20]: sns.distplot(data=df, x='salary', hue='level of education', palette='viridis')
```

```
Out[20]: <seaborn.axisgrid.FacetGrid at 0x1fd5be31460>
```



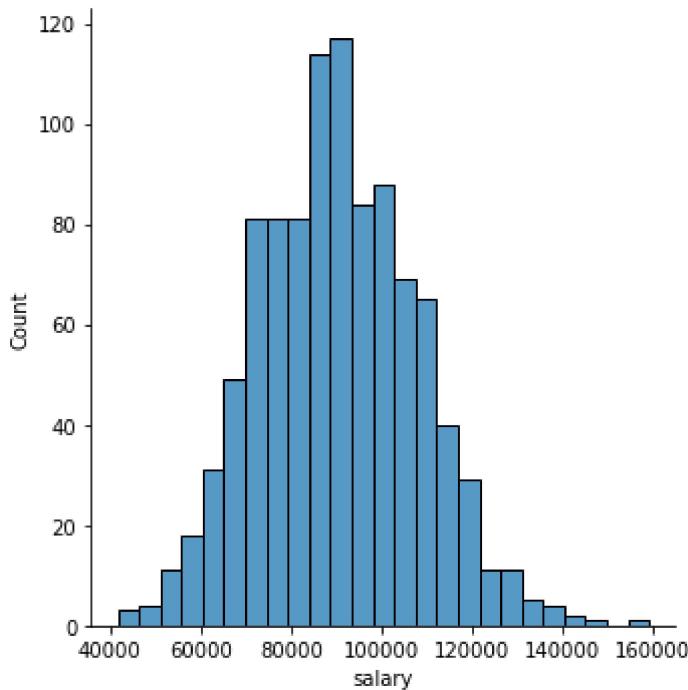
```
In [21]: sns.displot(data=df,x='salary',kde=True)
```

```
Out[21]: <seaborn.axisgrid.FacetGrid at 0x1fd5bd81ee0>
```



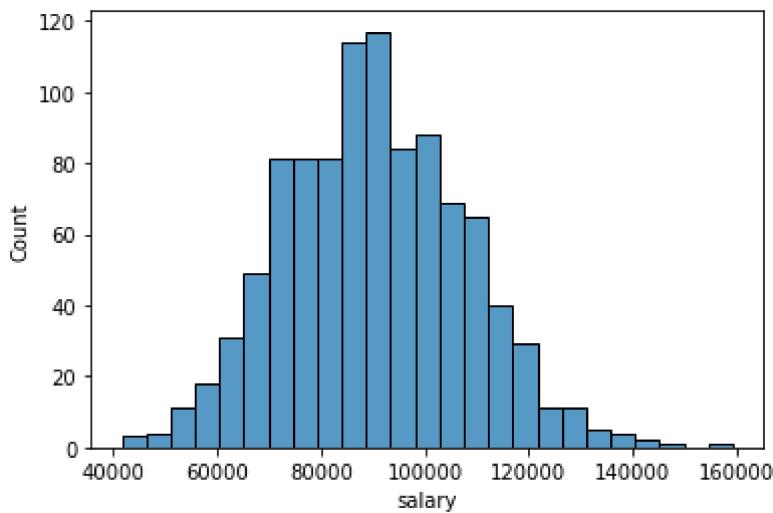
```
In [22]: sns.displot(data=df,x='salary')
```

```
Out[22]: <seaborn.axisgrid.FacetGrid at 0x1fd5bd81fd0>
```



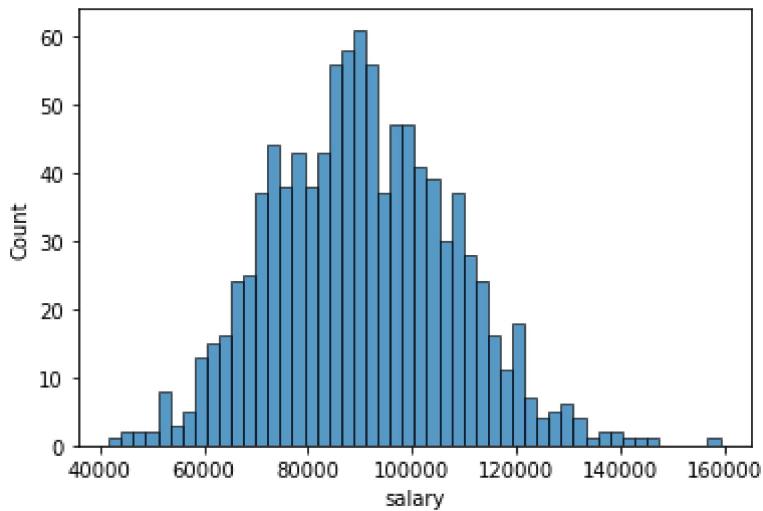
```
In [23]: sns.histplot(data=df,x='salary')
```

```
Out[23]: <AxesSubplot:xlabel='salary', ylabel='Count'>
```



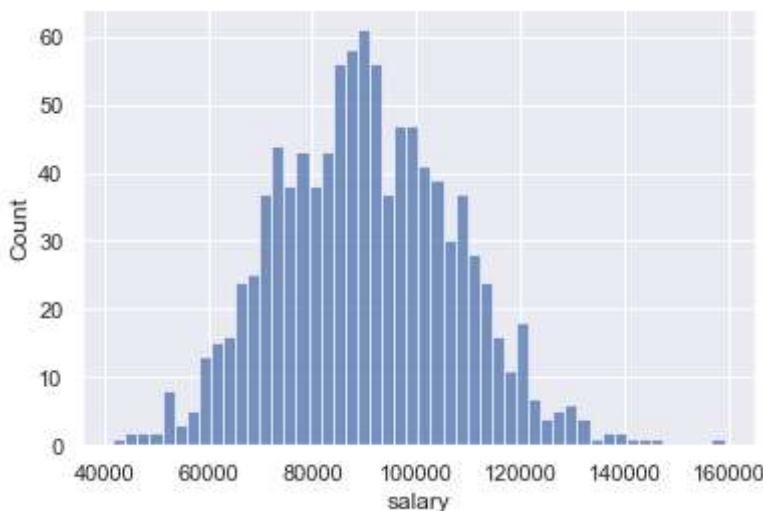
```
In [24]: sns.histplot(data=df,x='salary',bins=50)
```

```
Out[24]: <AxesSubplot:xlabel='salary', ylabel='Count'>
```



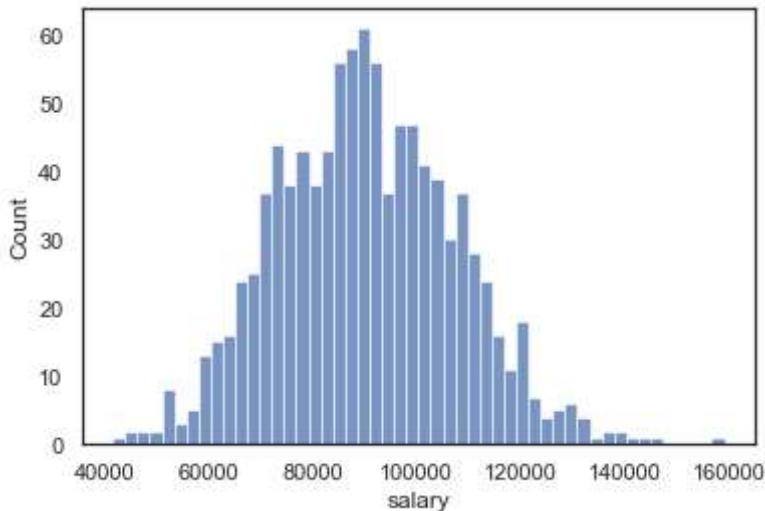
```
In [25]: sns.set(style='darkgrid')
sns.histplot(data=df,x='salary',bins=50)
```

```
Out[25]: <AxesSubplot:xlabel='salary', ylabel='Count'>
```



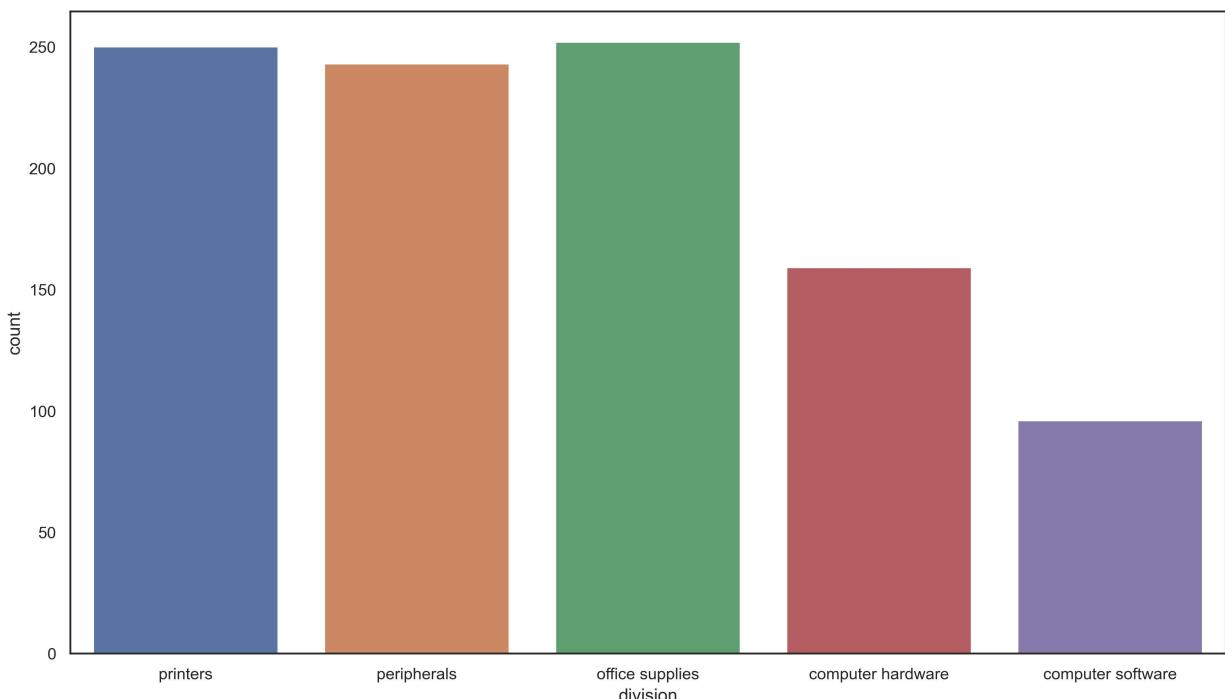
```
In [26]: sns.set(style='white')
sns.histplot(data=df,x='salary',bins=50)
```

```
Out[26]: <AxesSubplot:xlabel='salary', ylabel='Count'>
```



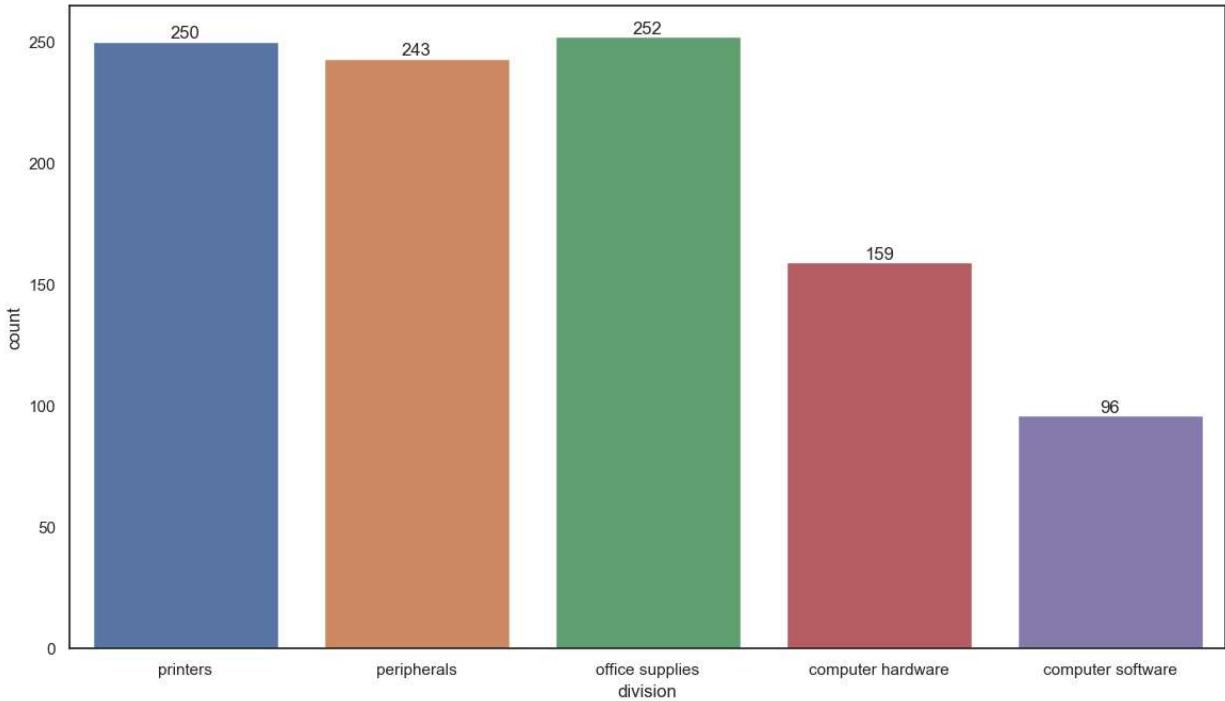
```
In [28]: plt.figure(figsize=(14,8),dpi=300)
sns.countplot(x='division',data=df)
```

```
Out[28]: <AxesSubplot:xlabel='division', ylabel='count'>
```



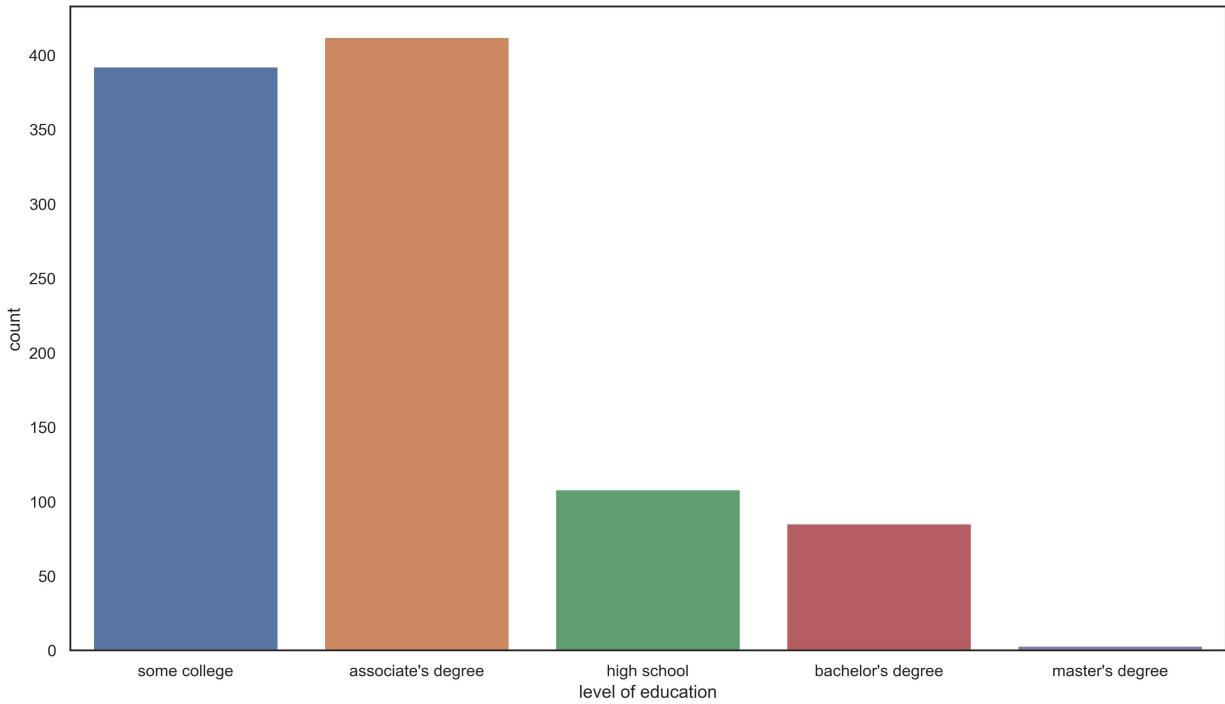
```
In [29]: plt.figure(figsize=(14,8),dpi=100)
ax=sns.countplot(x='division',data=df)
ax.bar_label(ax.containers[0])
```

```
Out[29]: [Text(0, 0, '250'),
Text(0, 0, '243'),
Text(0, 0, '252'),
Text(0, 0, '159'),
Text(0, 0, '96')]
```



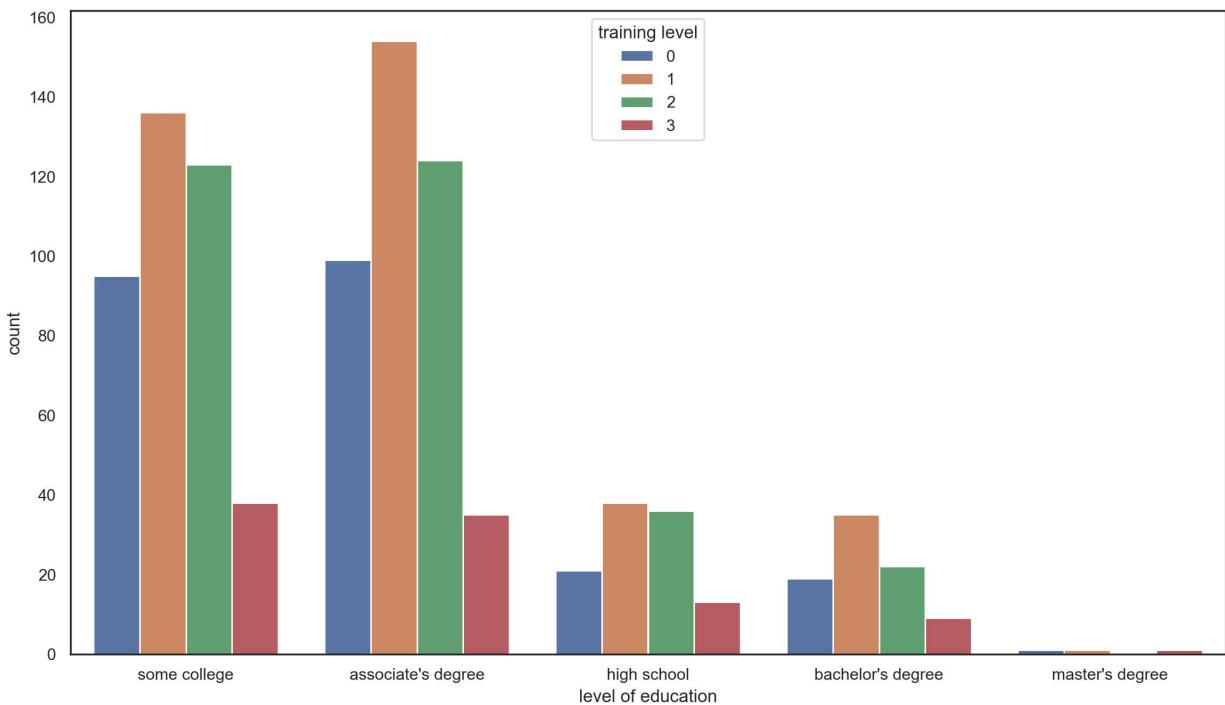
```
In [30]: plt.figure(figsize=(14,8),dpi=300)
sns.countplot(x='level of education',data=df)
```

```
Out[30]: <AxesSubplot:xlabel='level of education', ylabel='count'>
```



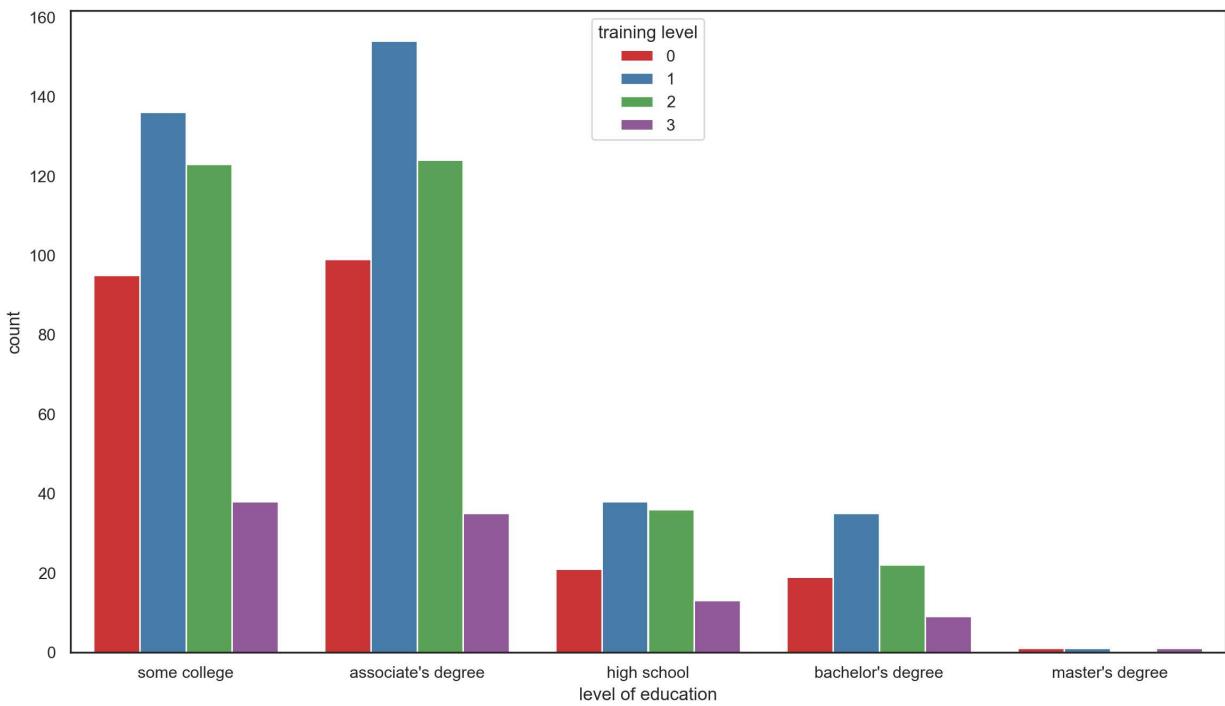
```
In [31]: plt.figure(figsize=(14,8),dpi=200)
sns.countplot(x='level of education',data=df,hue='training level')
```

```
Out[31]: <AxesSubplot:xlabel='level of education', ylabel='count'>
```

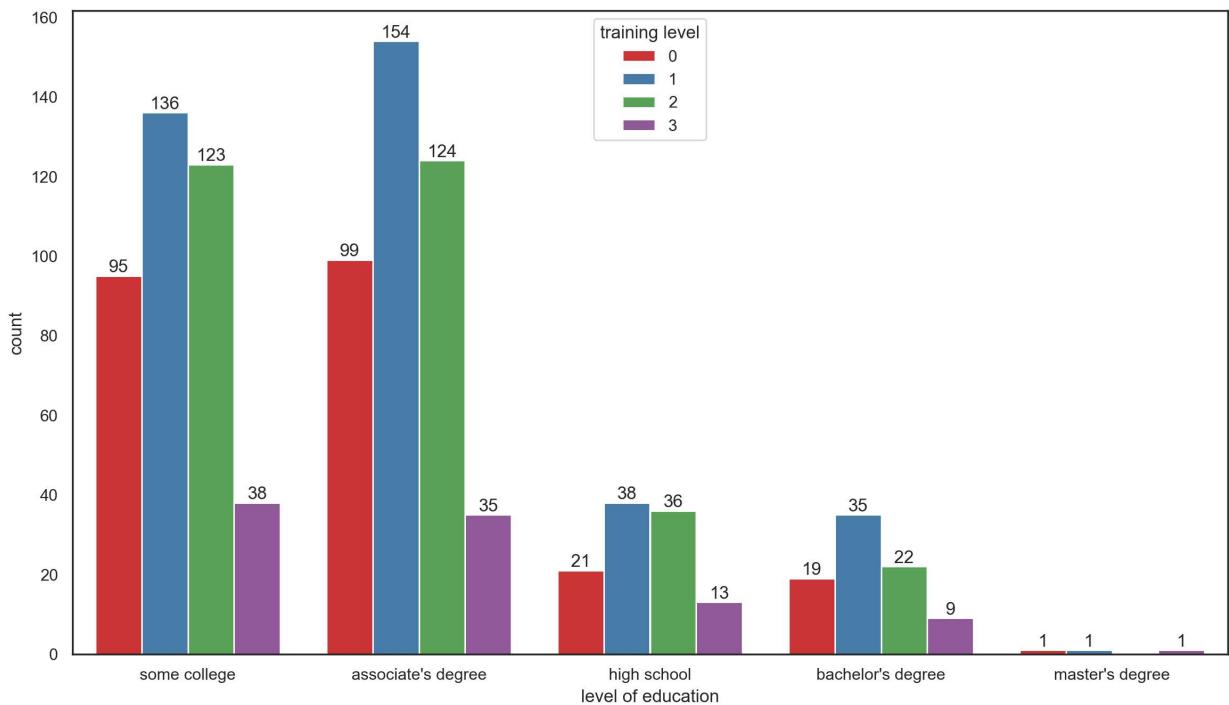


```
In [32]: plt.figure(figsize=(14,8),dpi=200)
sns.countplot(x='level of education',data=df,hue='training level',palette='Set1')
```

Out[32]: <AxesSubplot:xlabel='level of education', ylabel='count'>



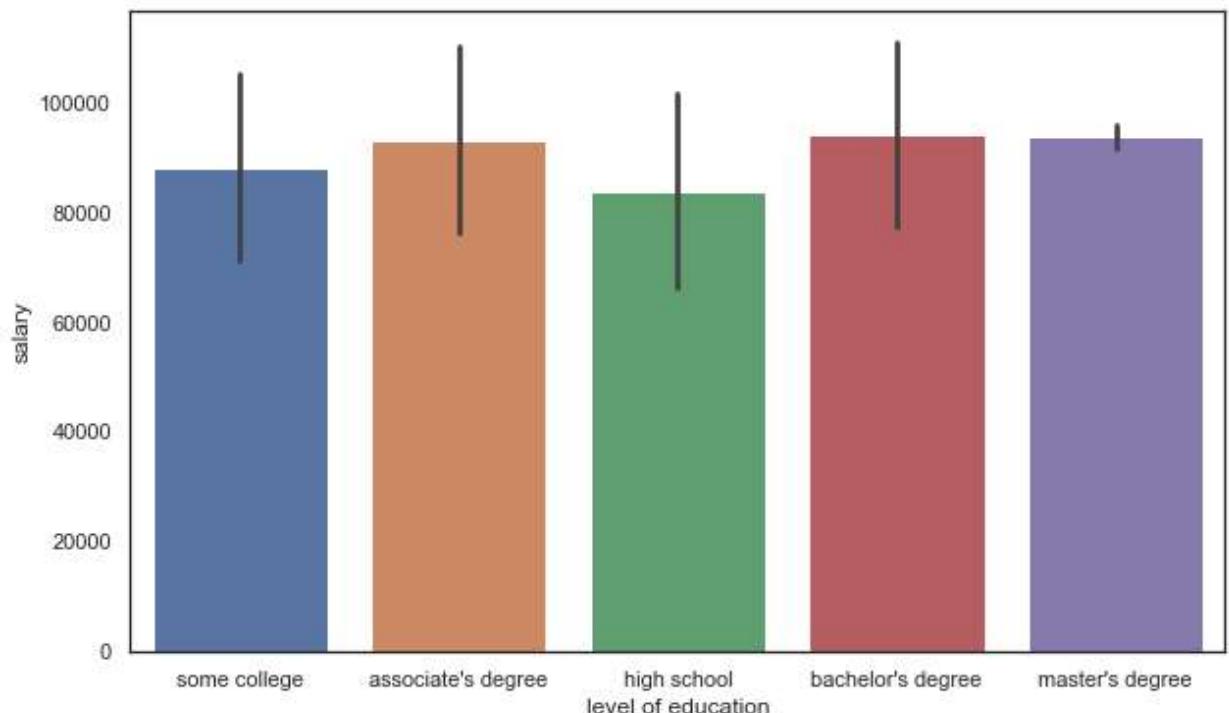
```
In [33]: plt.figure(figsize=(14,8),dpi=200)
ax=sns.countplot(x='level of education',data=df,hue='training level',palette='Set1')
for container in ax.containers:
    ax.bar_label(container)
```



barplot()

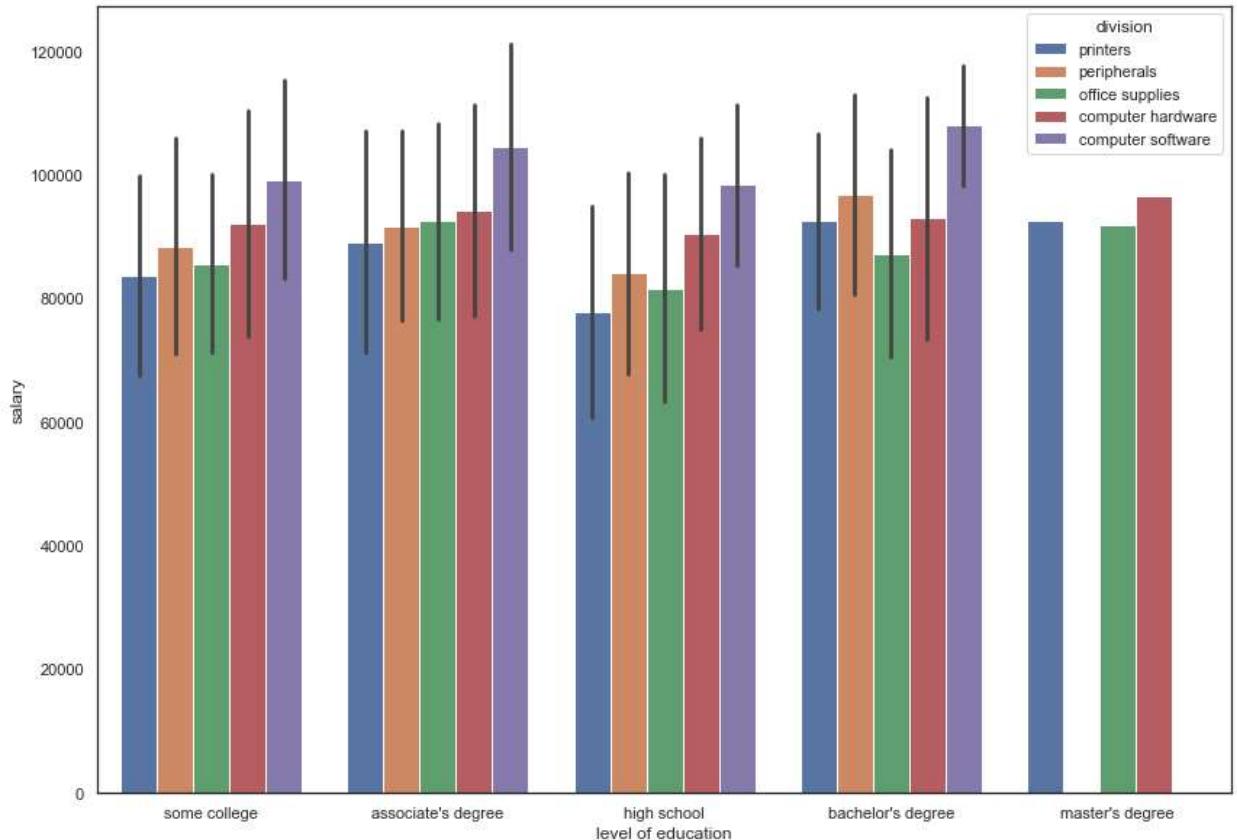
```
In [39]: plt.figure(figsize=(10,6))
sns.barplot(x='level of education',y='salary',data=df,estimator=np.mean,ci='sd')
```

```
Out[39]: <AxesSubplot:xlabel='level of education', ylabel='salary'>
```



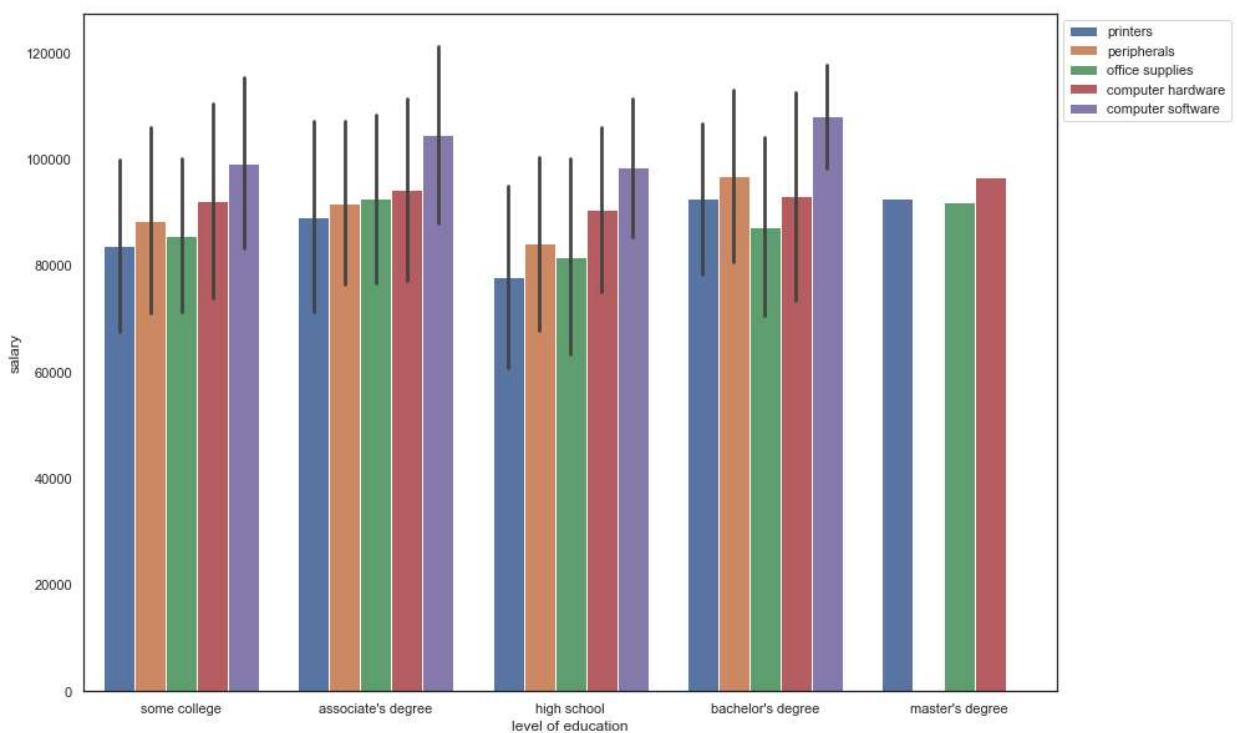
```
In [42]: plt.figure(figsize=(14,10))
sns.barplot(x='level of education',y='salary',data=df,estimator=np.mean,ci='sd',hue='c')
```

```
Out[42]: <AxesSubplot:xlabel='level of education', ylabel='salary'>
```



```
In [45]: plt.figure(figsize=(14,10))
sns.barplot(x='level of education',y='salary',data=df,estimator=np.mean,ci='sd',hue='division')
plt.legend(bbox_to_anchor=(1,1))
```

Out[45]: <matplotlib.legend.Legend at 0x1fd64333df0>



```
In [46]: df= pd.read_csv("StudentsPerformance.csv")
```

```
In [47]: df.head()
```

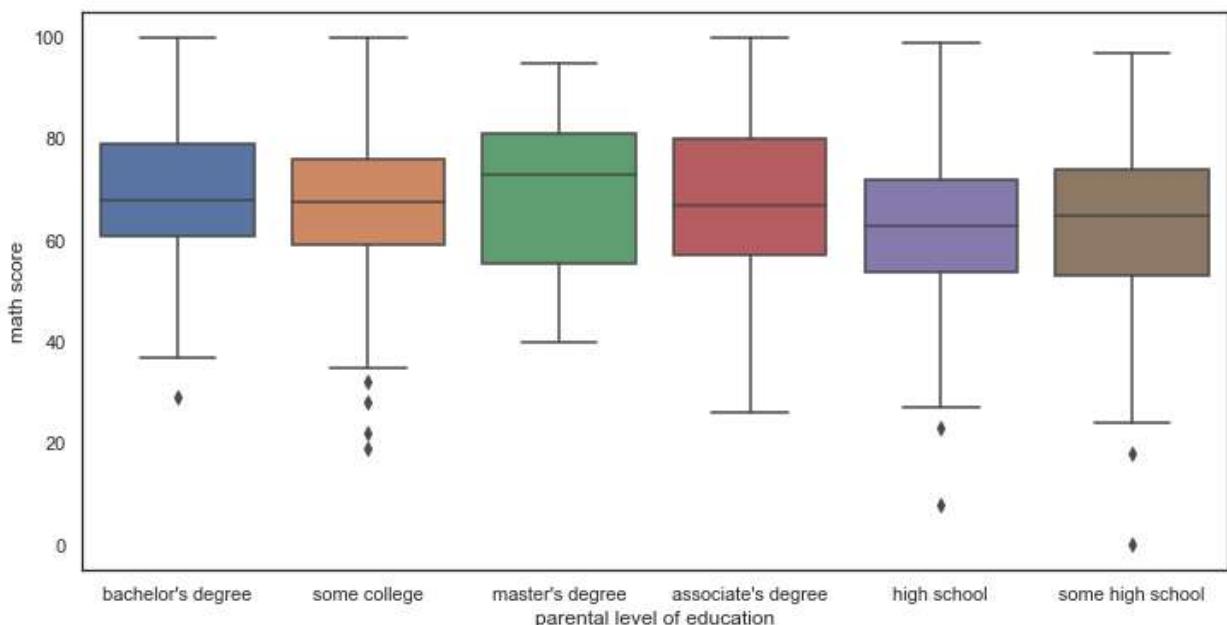
Out[47]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Boxplot

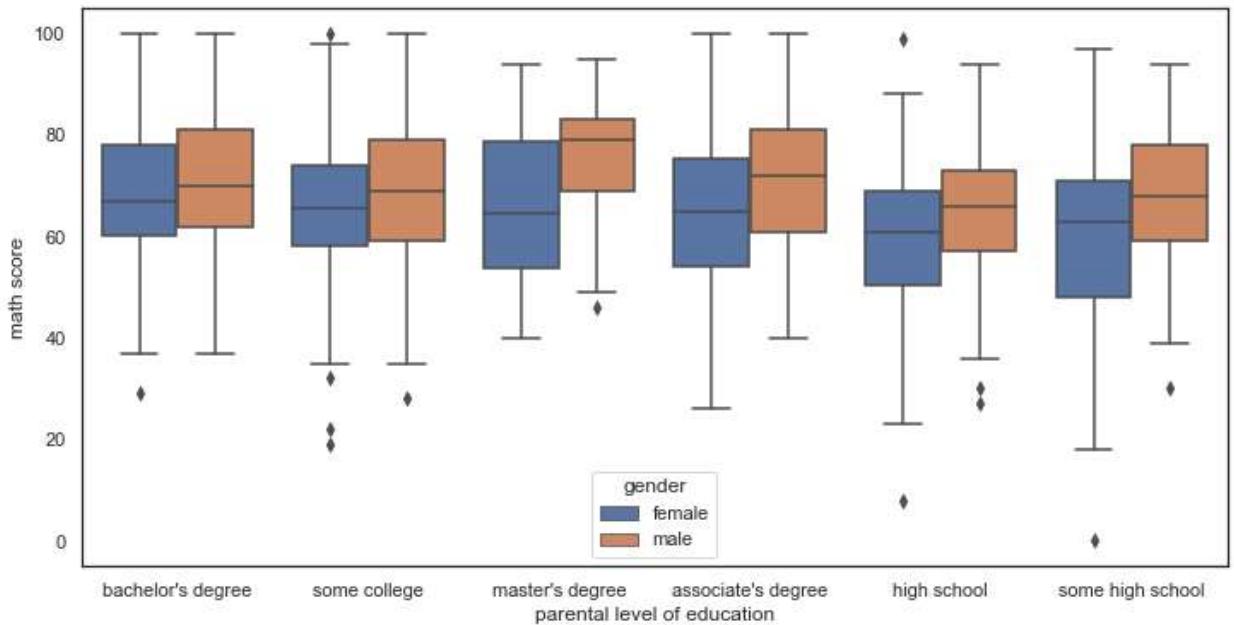
```
In [48]: plt.figure(figsize=(12,6))
sns.boxplot(x='parental level of education',y='math score',data=df)
```

Out[48]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>



```
In [49]: plt.figure(figsize=(12,6))
sns.boxplot(x='parental level of education',y='math score',data=df,hue='gender')
```

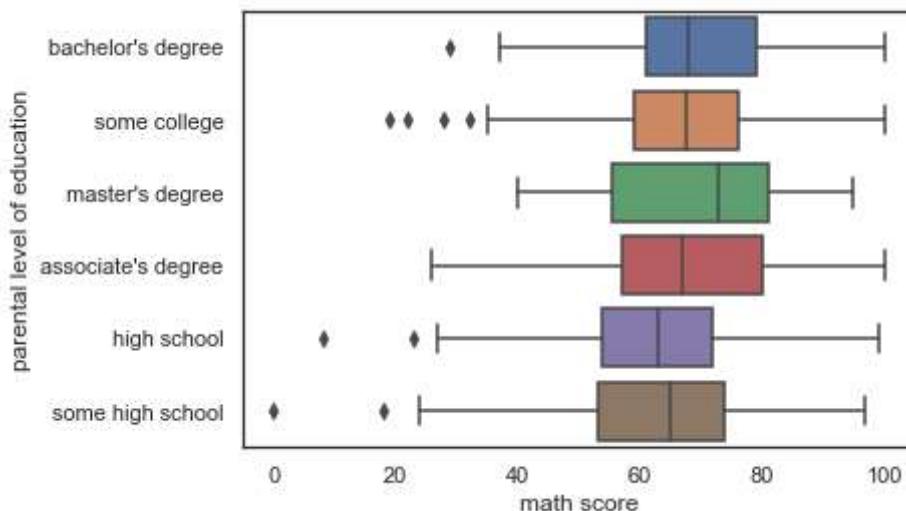
Out[49]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>



Boxplot styling parameter

```
In [54]: sns.boxplot(y='parental level of education',x='math score',data=df,orient='h')
```

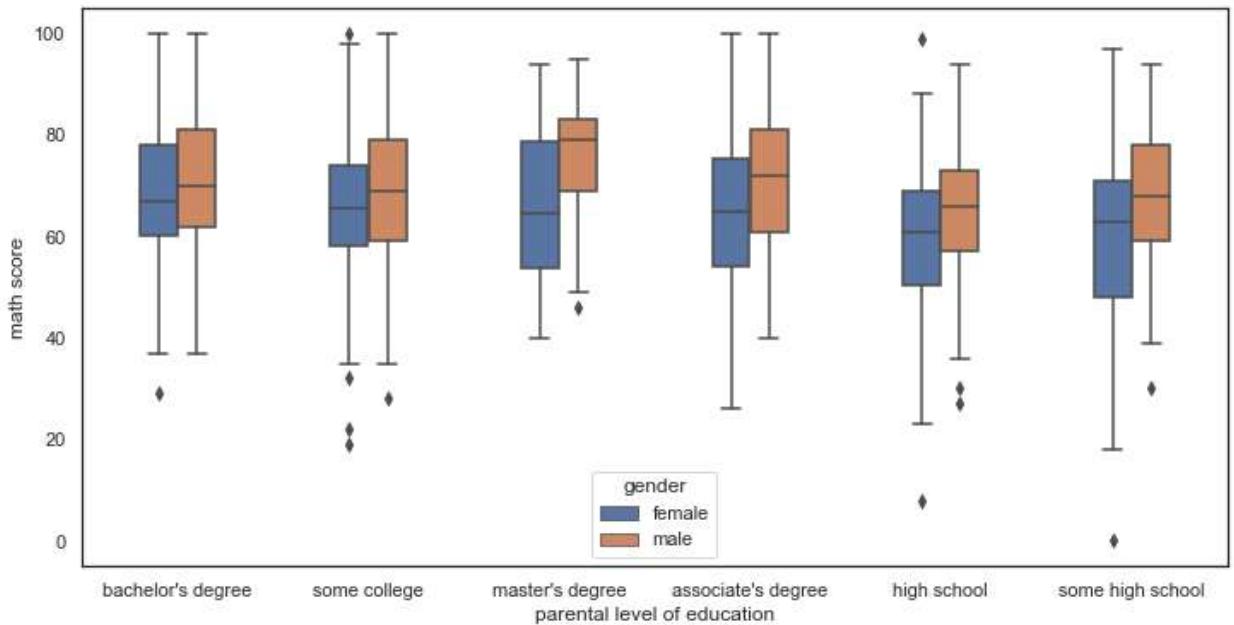
```
Out[54]: <AxesSubplot:xlabel='math score', ylabel='parental level of education'>
```



Width

```
In [58]: plt.figure(figsize=(12,6))
sns.boxplot(x='parental level of education',y='math score',data=df,hue='gender',width=)
```

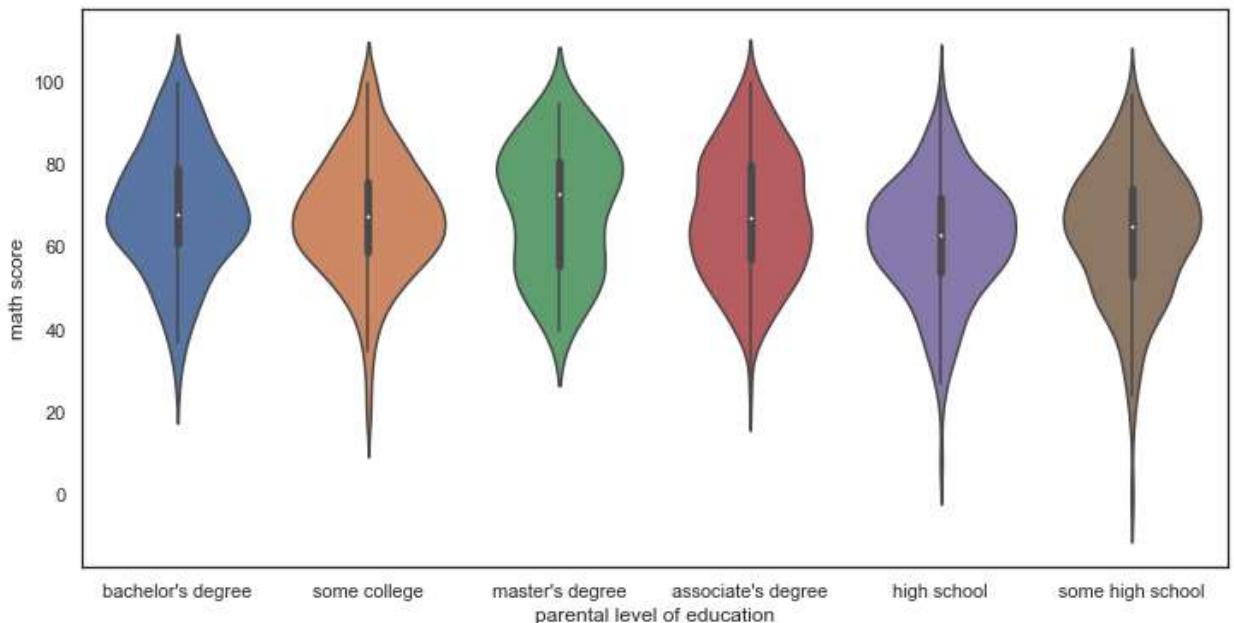
```
Out[58]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



violinplot

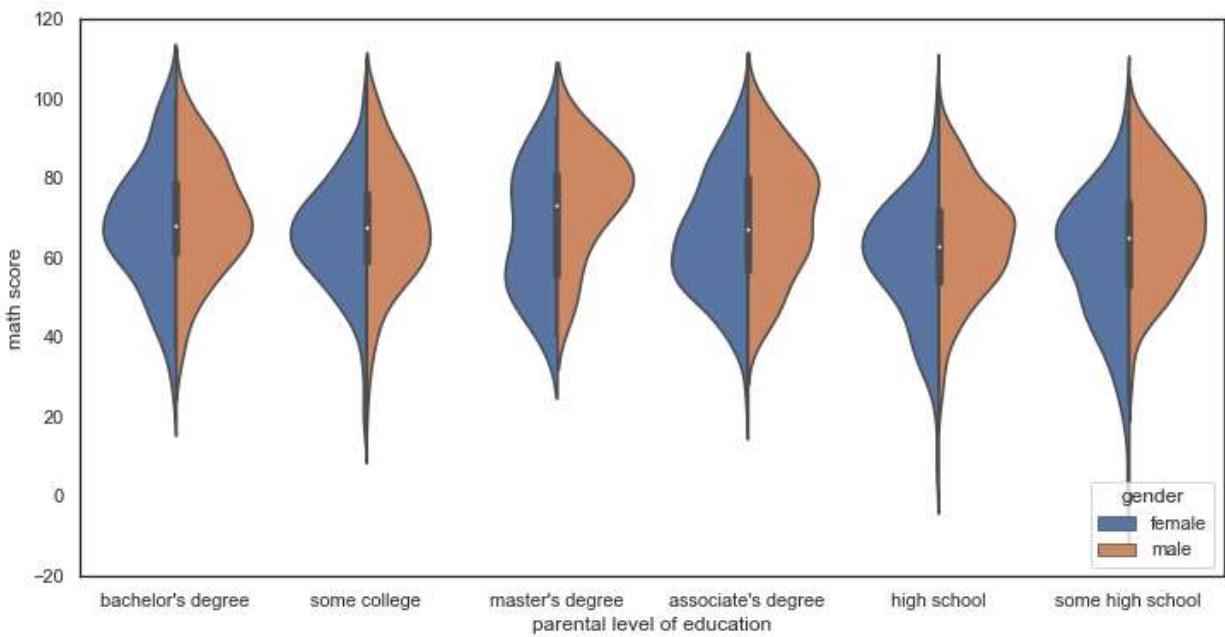
```
In [60]: plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df)
```

```
Out[60]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



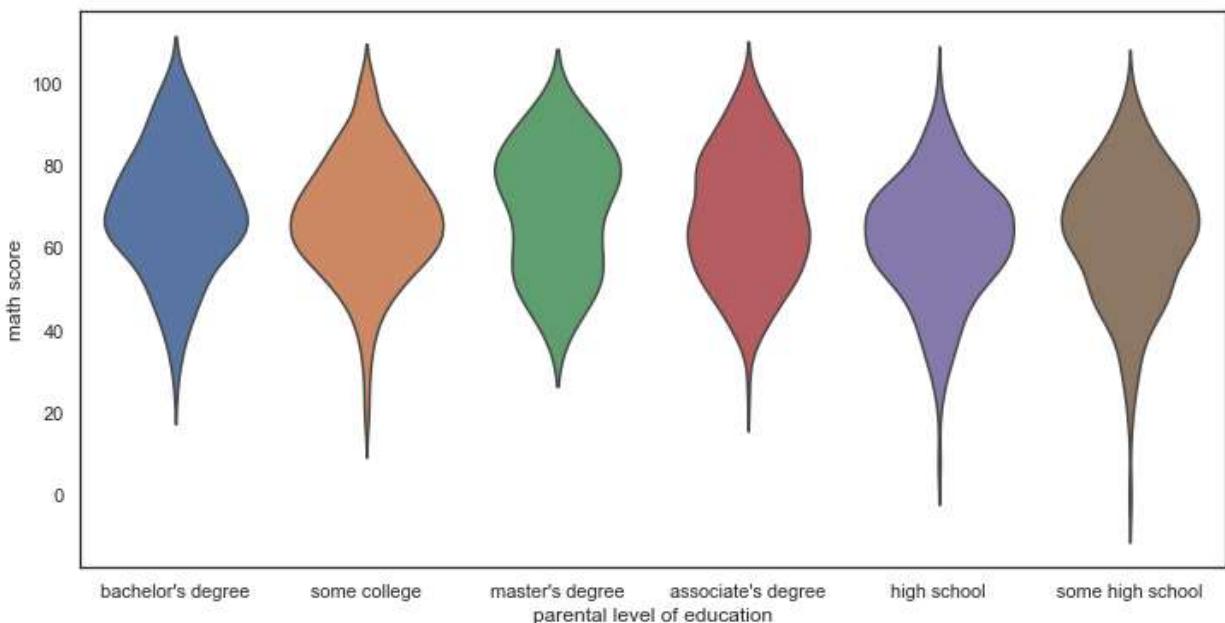
```
In [61]: plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,hue='gender',sp]
```

```
Out[61]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



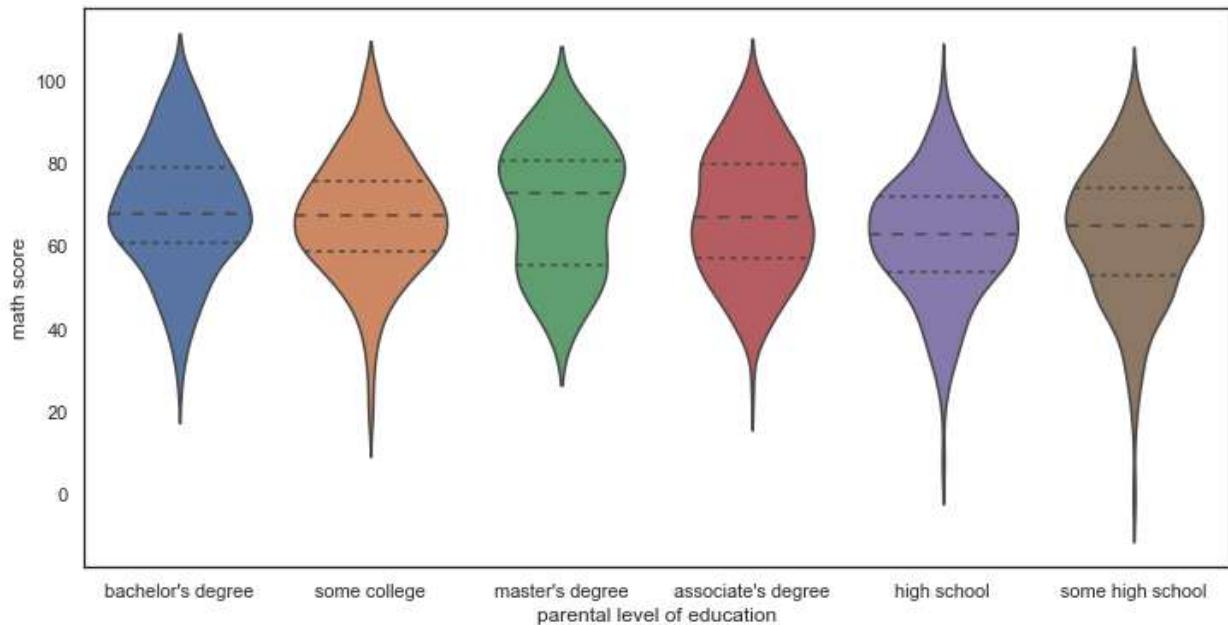
```
In [63]: plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,inner=None)
```

```
Out[63]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



```
In [65]: plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,inner='quartile')
```

```
Out[65]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



swarmplot

In []: