(http://www.pieriandata.com)

Categorical Plots - Distribution within Categories

So far we've seen how to apply a statistical estimation (like mean or count) to categories and compare them to one another. Let's now explore how to visualize the distribution within categories. We already know about distplot() which allows to view the distribution of a single feature, now we will break down that same distribution per category.

Imports

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

The Data

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

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

Out[3]:

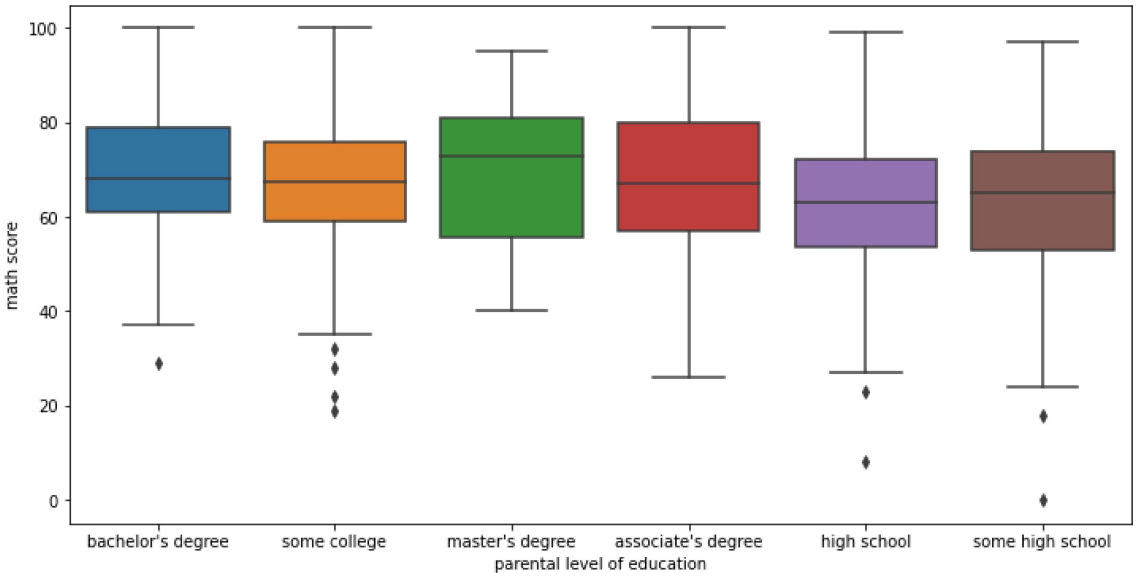
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group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

Boxplot

As described in the video, a boxplot display distribution through the use of quartiles and an IQR for outliers.

In [4]:
```
plt.figure(figsize=(12,6))
sns.boxplot(x='parental level of education',y='math score',data=df)
```

Out[4]:
```
<AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```

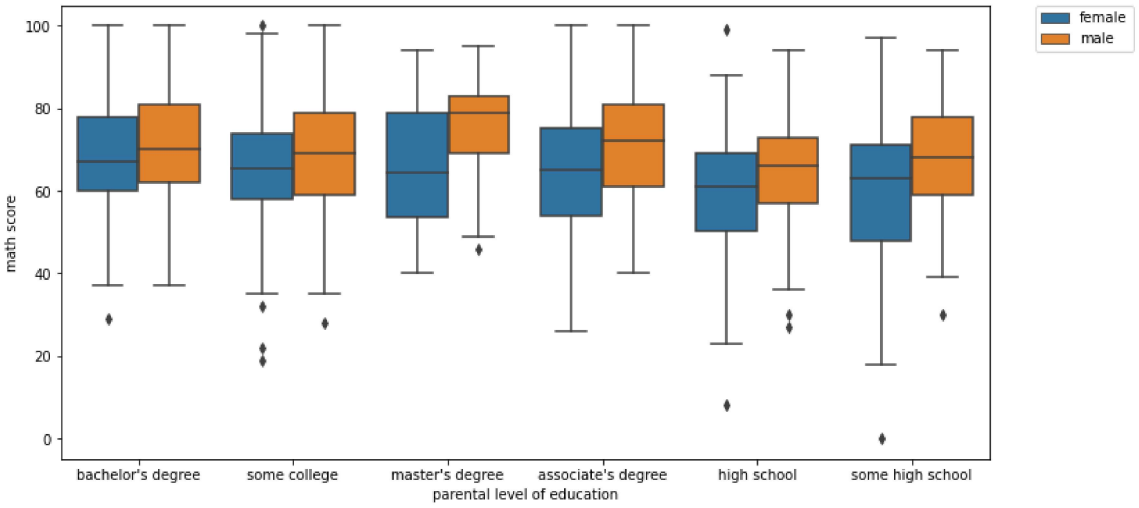


Adding hue for further segmentation

In [5]:
```
plt.figure(figsize=(12,6))
sns.boxplot(x='parental level of education',y='math score',data=df,hue='gen

# Optional move the legend outside
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

Out[5]:
```
<matplotlib.legend.Legend at 0x215583a7e08>
```

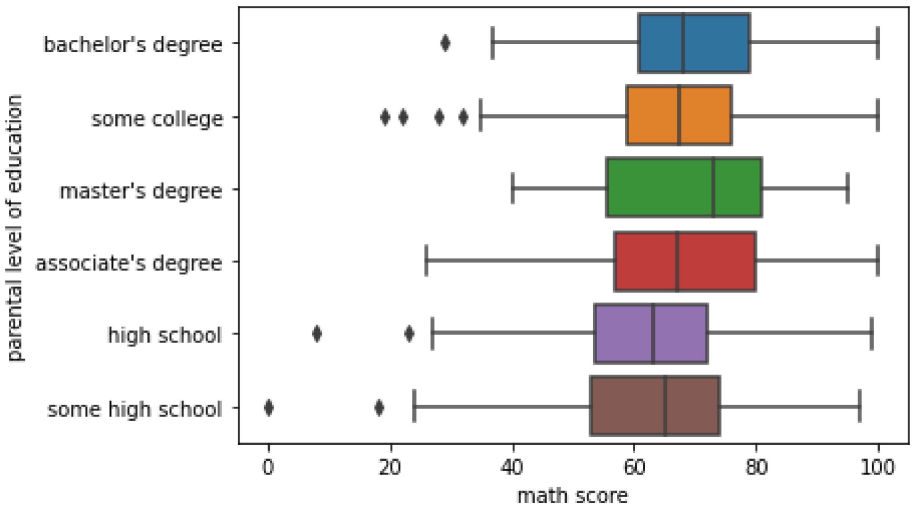


Boxplot Styling Parameters

Orientation

In [6]:
```
# NOTICE HOW WE HAVE TO SWITCH X AND Y FOR THE ORIENTATION TO MAKE SENSE!
sns.boxplot(x='math score',y='parental level of education',data=df,orient='
```

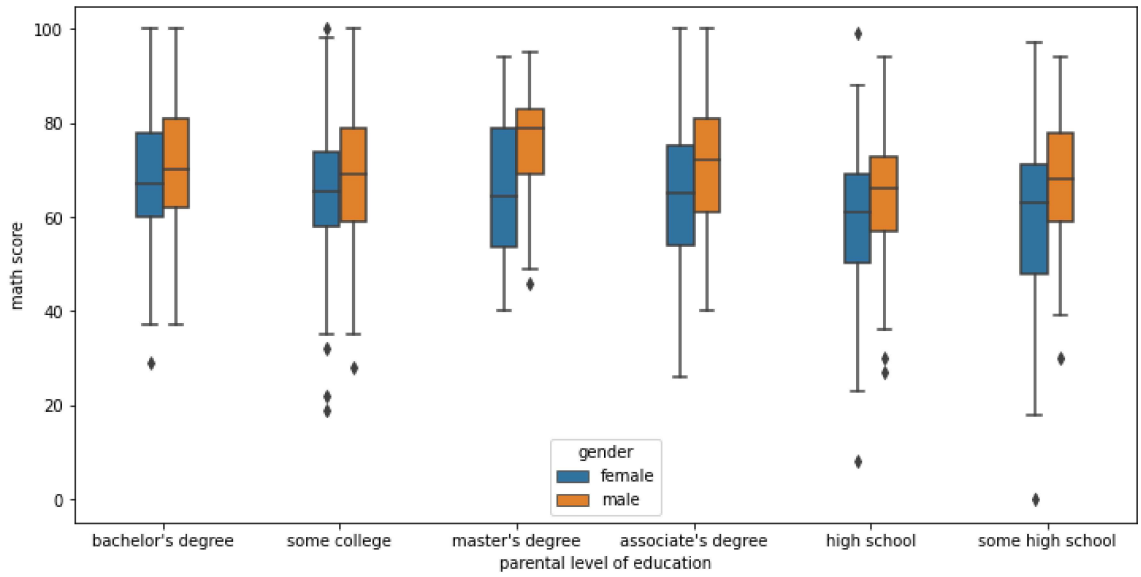
Out[6]:
```
<AxesSubplot:xlabel='math score', ylabel='parental level of education'>
```



Width

In [7]:
```
plt.figure(figsize=(12,6))
sns.boxplot(x='parental level of education',y='math score',data=df,hue='gen
```

Out[7]:
```
<AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```

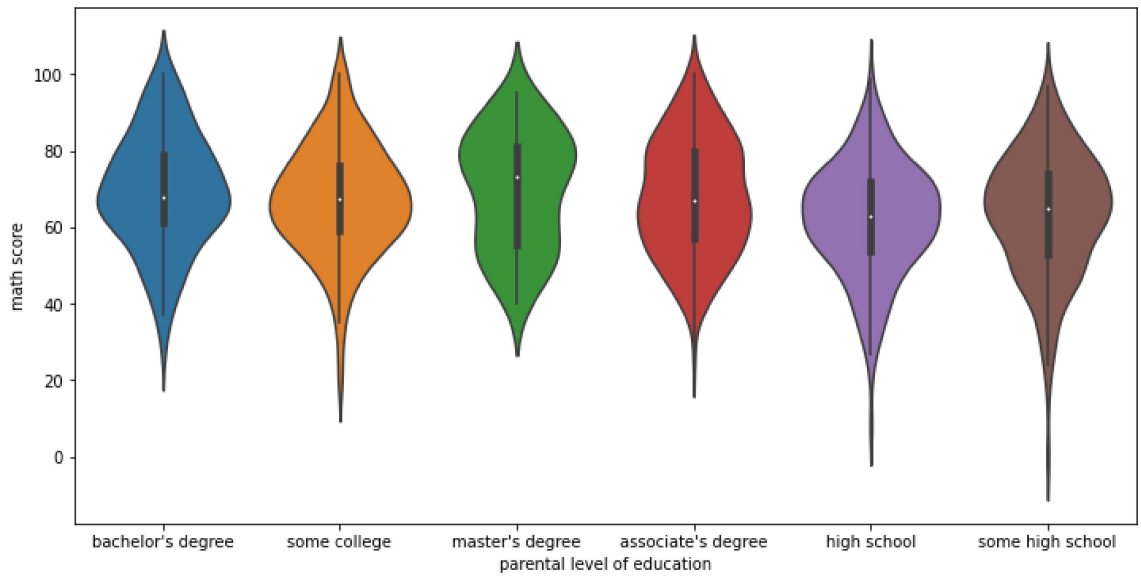


Violinplot

A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared. Unlike a box plot, in which all of the plot components correspond to actual datapoints, the violin plot features a kernel density estimation of the underlying distribution.

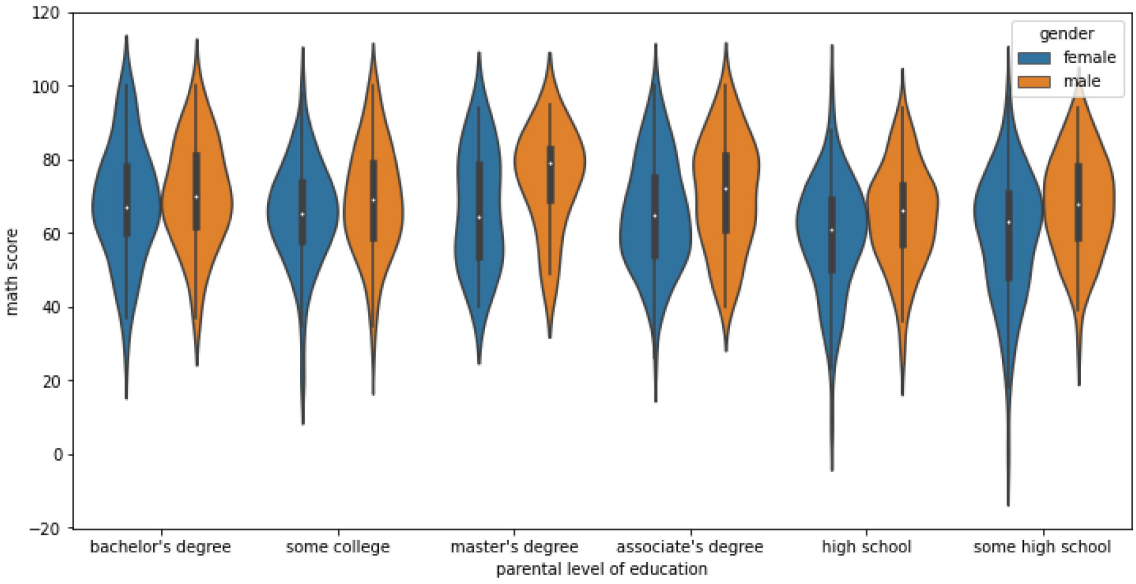
In [8]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df)
```

Out[8]:
```
<AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



In [9]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,hue='
```

Out[9]:
```
<AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



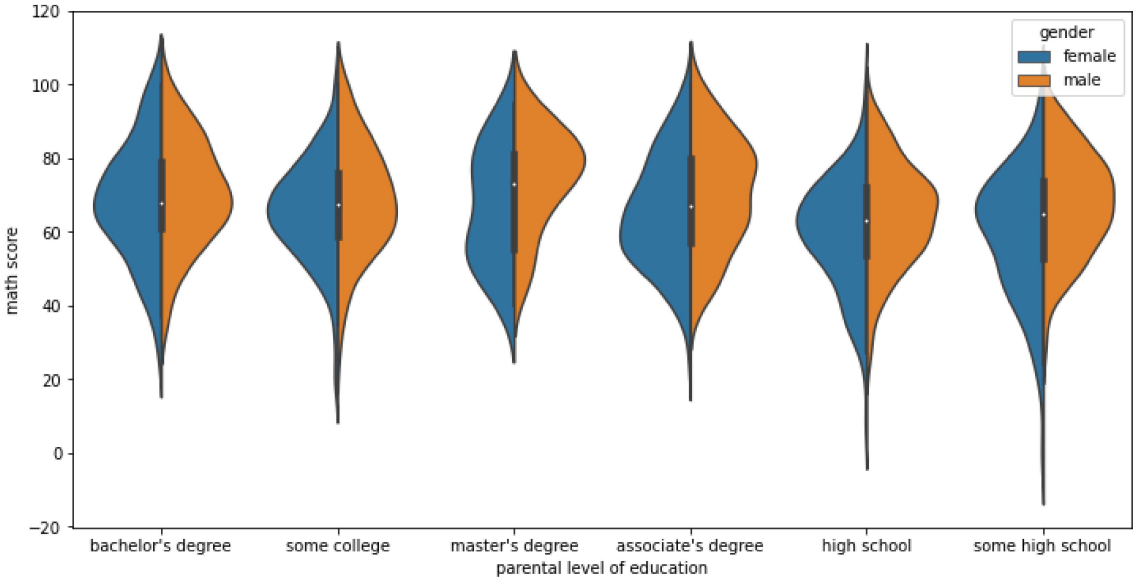
Violinplot Parameters

split

When using hue nesting with a variable that takes two levels, setting split to True will draw half of a violin for each level. This can make it easier to directly compare the distributions.

In [10]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,hue='
```

Out[10]:
```
<AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```

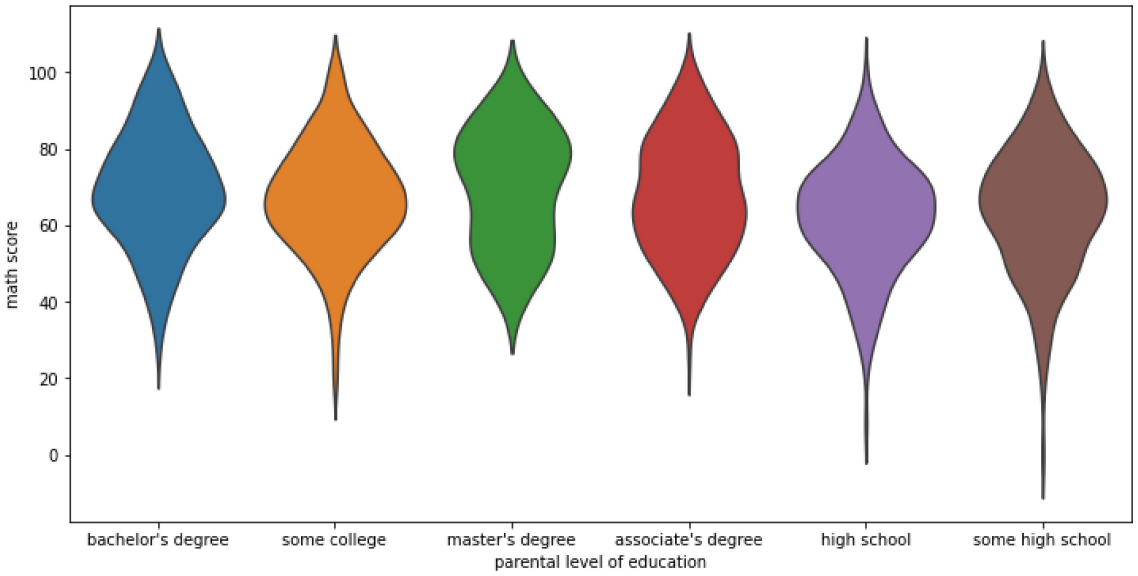


inner

Representation of the datapoints in the violin interior. If box, draw a miniature boxplot. If quartiles, draw the quartiles of the distribution. If point or stick, show each underlying

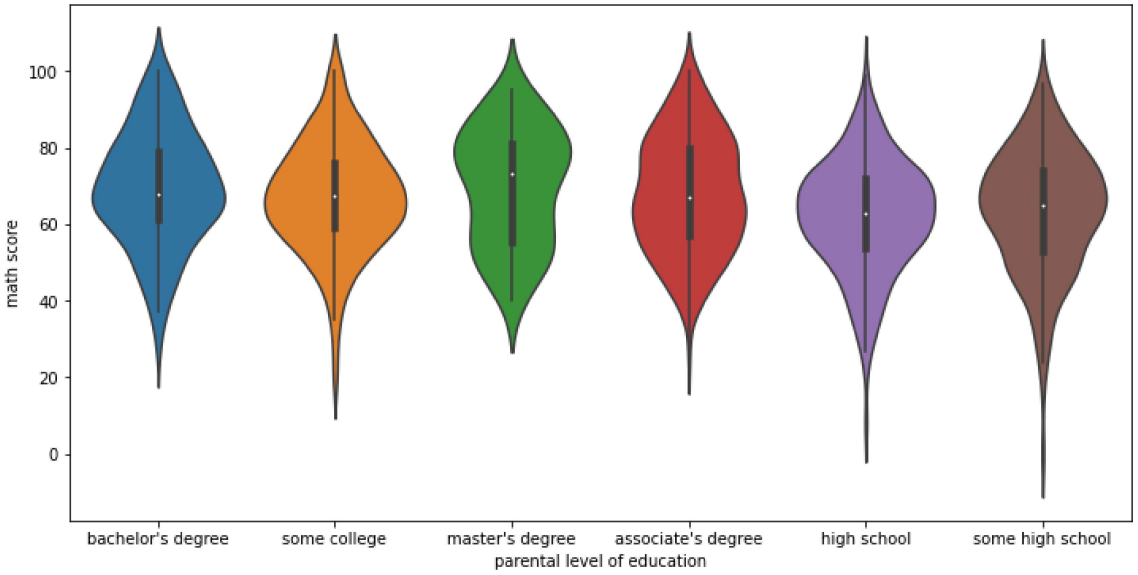
In [11]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,inner
```

Out[11]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>



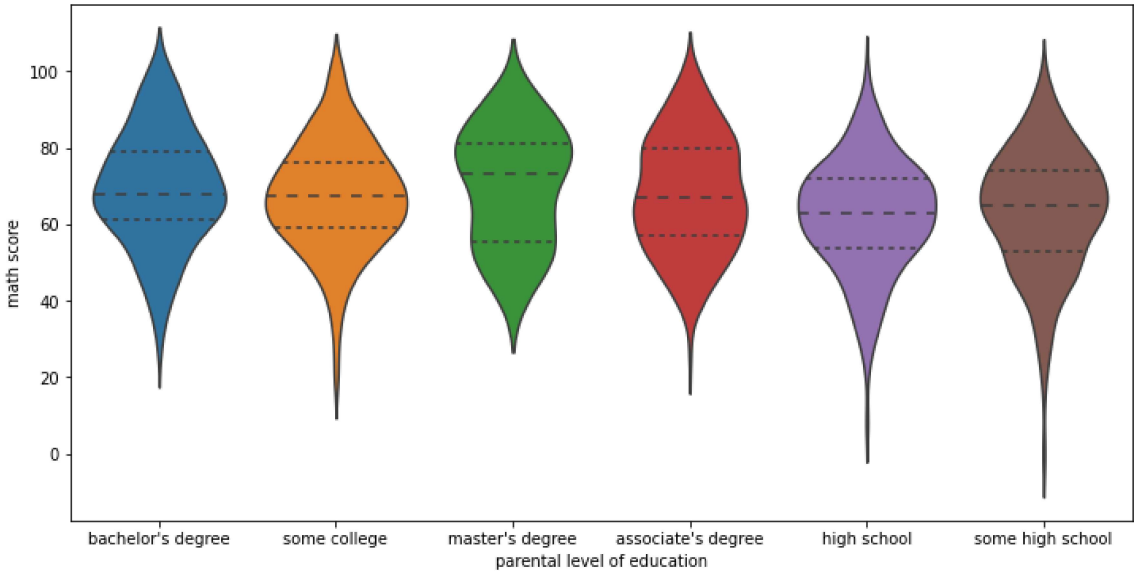
In [12]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,inner
```

Out[12]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>



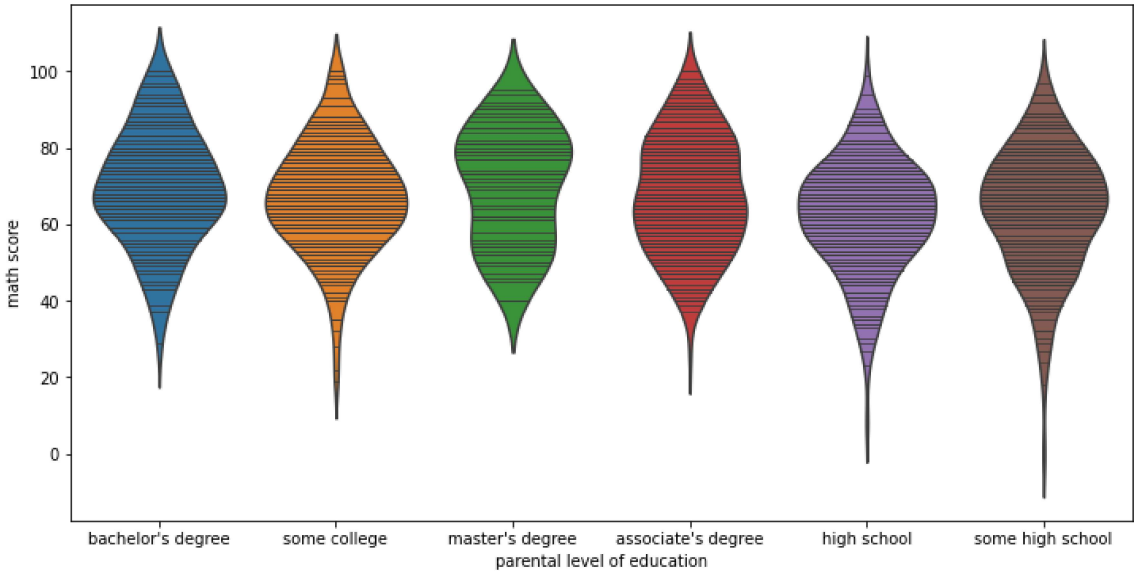
In [13]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,inner
```

Out[13]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>



In [14]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,inner
```

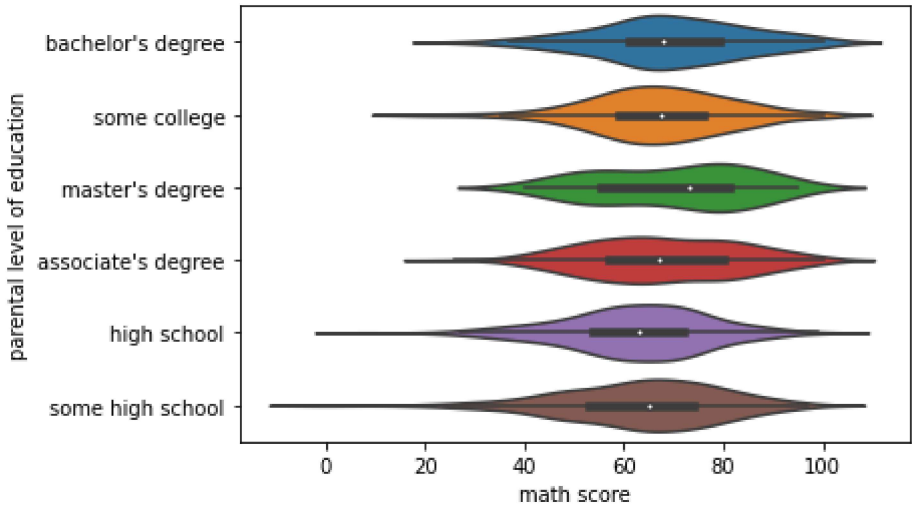
Out[14]: <AxesSubplot:xlabel='parental level of education', ylabel='math score'>



orientation

In [15]:
```
# Simply switch the continuous variable to y and the categorical to x
sns.violinplot(x='math score',y='parental level of education',data=df,)
```

Out[15]:
```
<AxesSubplot:xlabel='math score', ylabel='parental level of education'>
```

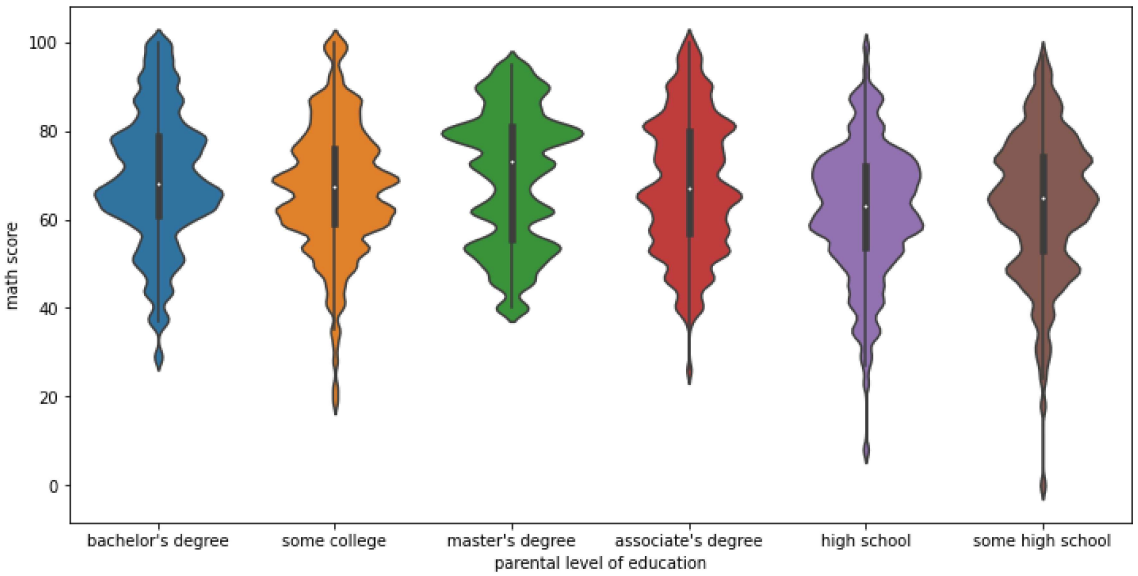


bandwidth

Similar to bandwidth argument for kdeplot

In [16]:
```
plt.figure(figsize=(12,6))
sns.violinplot(x='parental level of education',y='math score',data=df,bw=0.
```

Out[16]:
```
<AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```



Advanced Plots

We can use a boxenplot and swarmplot to achieve the same effect as the boxplot and violinplot, but with slightly more information included. Be careful when using these plots, as they often require you to educate the viewer with how the plot is actually constructed. Only use these if you are sure your audience will understand the visualization.

In [17]:
```
df.head()
```

Out[17]:

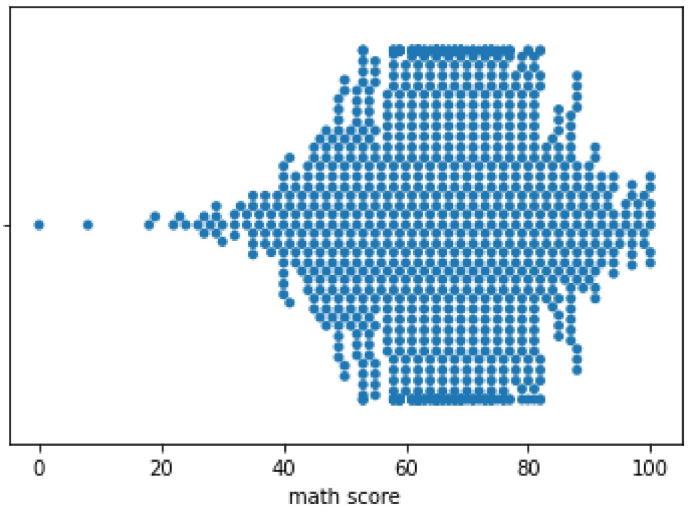
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group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

swarmplot

In [26]:
```
sns.swarmplot(x='math score',data=df)
```

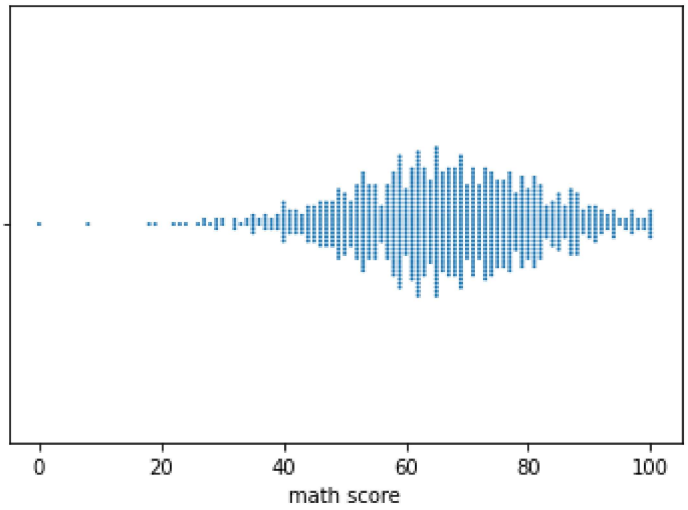
```
c:\users\marcial\anaconda3\envs\ml_master\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 15.8% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[26]:
```
<AxesSubplot:xlabel='math score'>
```



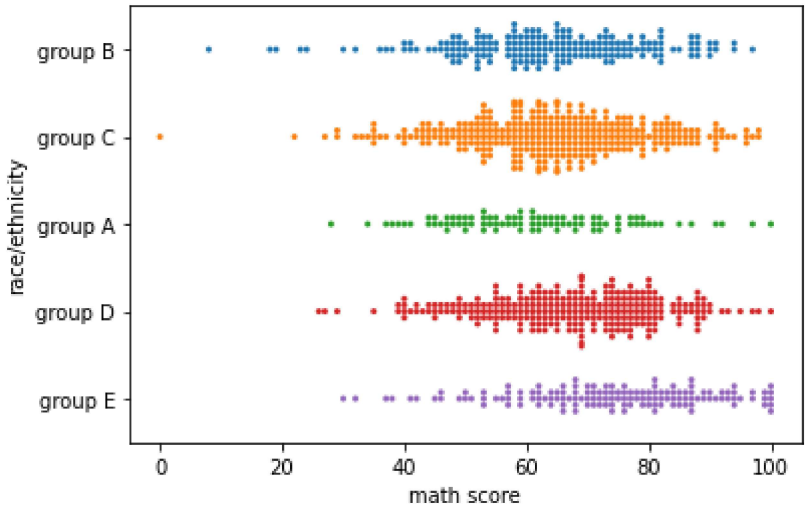
In [28]:
```
sns.swarmplot(x='math score',data=df,size=2)
```

Out[28]:
```
<AxesSubplot:xlabel='math score'>
```



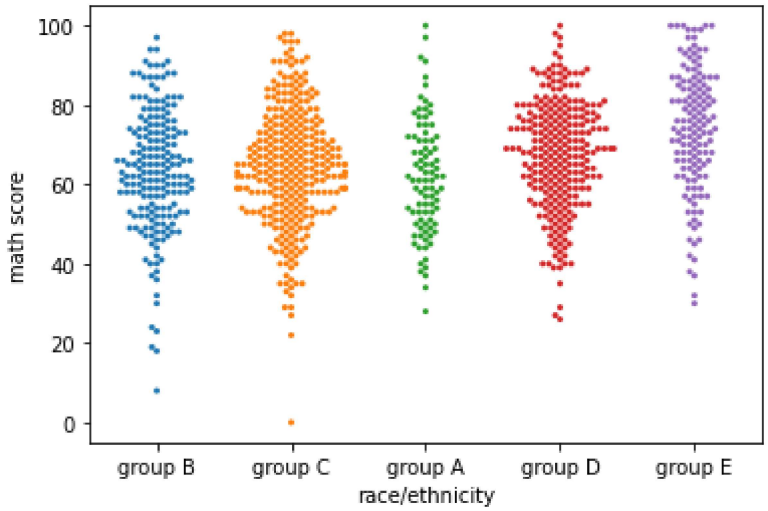
In [30]:
```
sns.swarmplot(x='math score',y='race/ethnicity',data=df,size=3)
```

Out[30]:
```
<AxesSubplot:xlabel='math score', ylabel='race/ethnicity'>
```



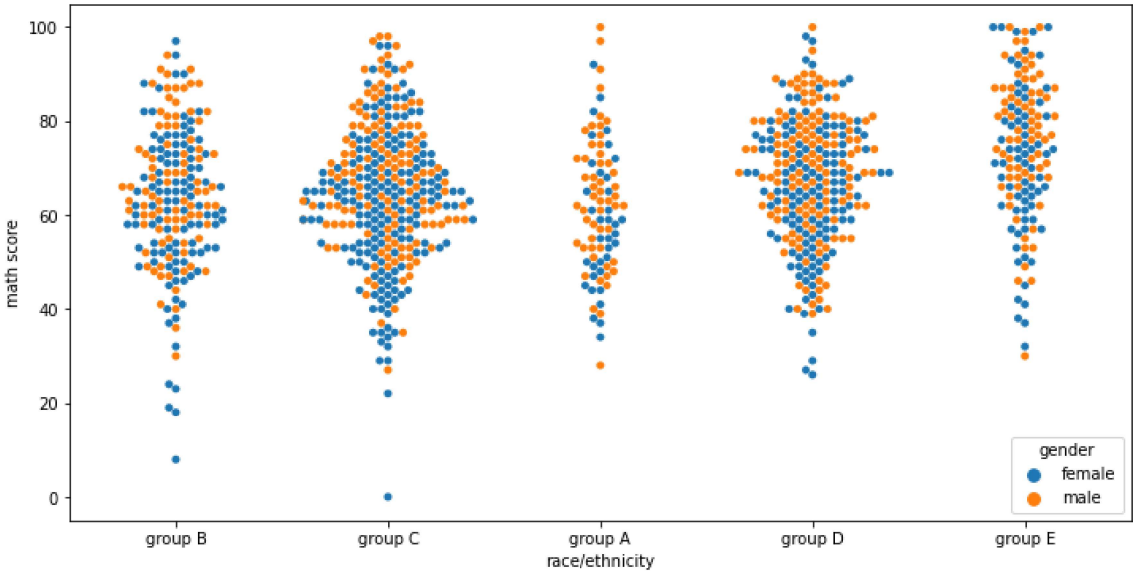
In [31]:
```
sns.swarmplot(x='race/ethnicity',y='math score',data=df,size=3)
```

Out[31]:
```
<AxesSubplot:xlabel='race/ethnicity', ylabel='math score'>
```



In [21]:
```
plt.figure(figsize=(12,6))
sns.swarmplot(x='race/ethnicity',y='math score',data=df,hue='gender')
```

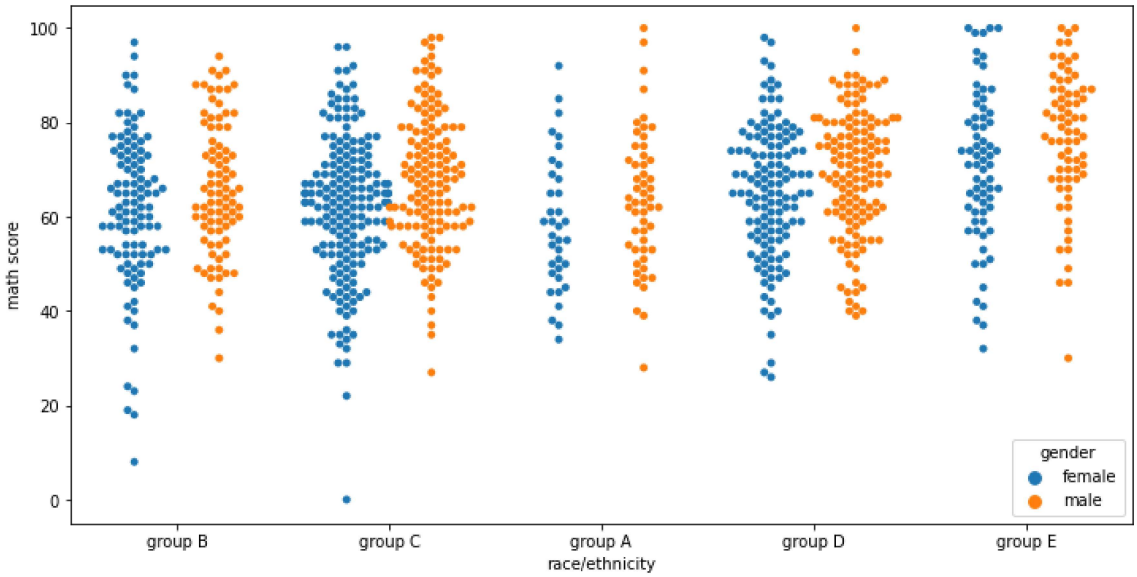
Out[21]:
```
<AxesSubplot:xlabel='race/ethnicity', ylabel='math score'>
```



In [22]:
```
plt.figure(figsize=(12,6))
sns.swarmplot(x='race/ethnicity',y='math score',data=df,hue='gender',dodge=
```

```
c:\users\marcial\anaconda3\envs\ml_master\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 6.7% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[22]:
```
<AxesSubplot:xlabel='race/ethnicity', ylabel='math score'>
```



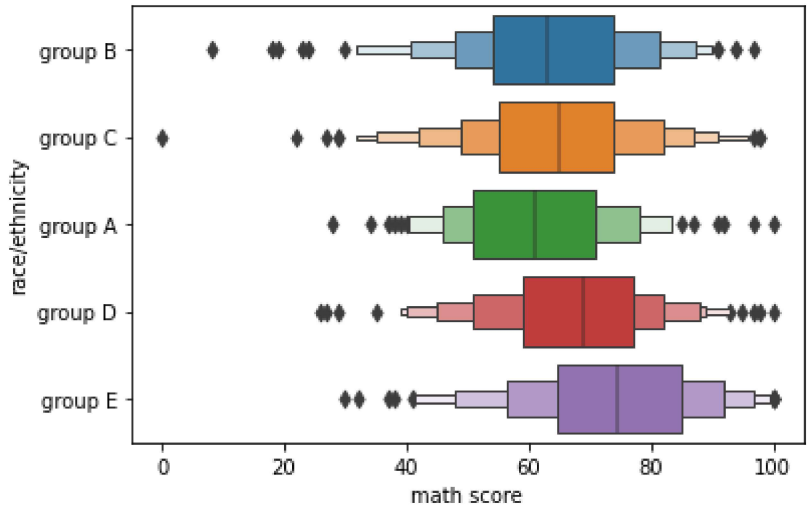
boxenplot (letter-value plot)

Official Paper on this plot: https://vita.had.co.nz/papers/letter-value-plot.html (https://vita.had.co.nz/papers/letter-value-plot.html)

This style of plot was originally named a “letter value” plot because it shows a large number of quantiles that are defined as “letter values”. It is similar to a box plot in plotting a nonparametric representation of a distribution in which all features correspond to actual observations. By plotting more quantiles, it provides more information about the shape of the distribution, particularly in the tails.

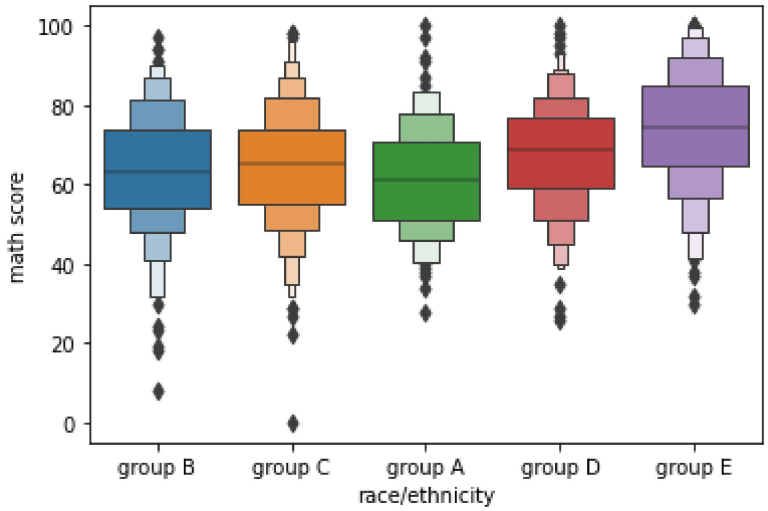
In [23]:
```
sns.boxenplot(x='math score',y='race/ethnicity',data=df)
```

Out[23]:
```
<AxesSubplot:xlabel='math score', ylabel='race/ethnicity'>
```



In [24]:
```
sns.boxenplot(x='race/ethnicity',y='math score',data=df)
```

Out[24]:
```
<AxesSubplot:xlabel='race/ethnicity', ylabel='math score'>
```



In [25]:
```
plt.figure(figsize=(12,6))
sns.boxenplot(x='race/ethnicity',y='math score',data=df,hue='gender')
```

Out[25]:
```
<AxesSubplot:xlabel='race/ethnicity', ylabel='math score'>
```