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

%matplotlib inline
```

In [5]:
```python
students = pd.read_csv(r'C:\Users\mike1\Documents\Thinkful\StudentsPerformance.csv', low_memory = False)
students.head()
```

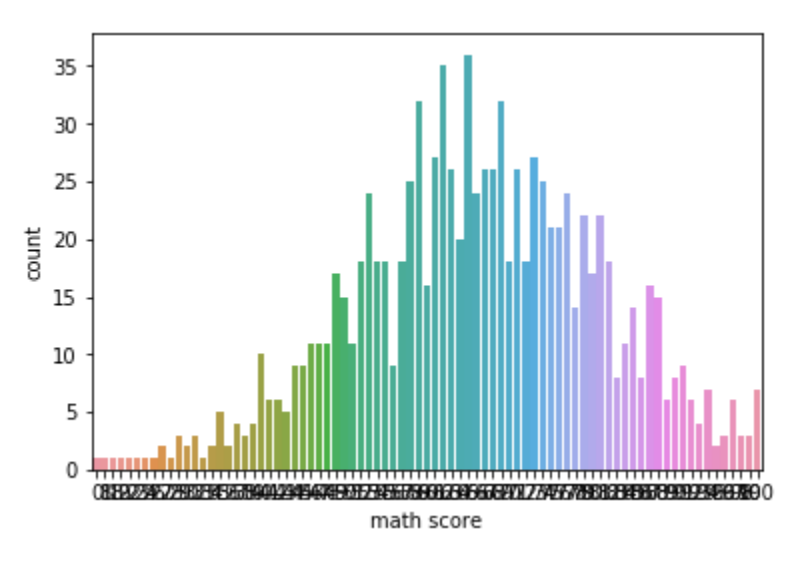
Out[5]:

	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

In [97]:
```python
##Choose one variable and plot that variable four different ways.
##Give the pros and cons of each plot you create. You can use variables from multiple datasets if you like.

sns.countplot(x = "math score", data = students)
```

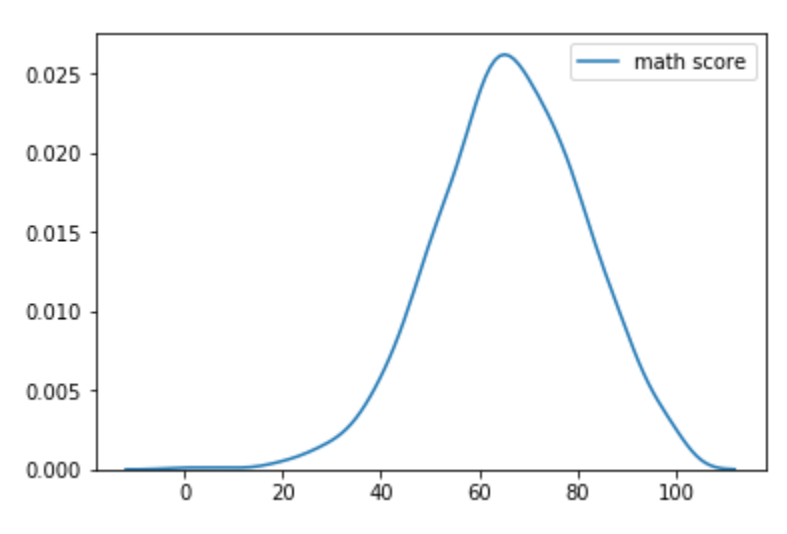
Out[97]: <matplotlib.axes._subplots.AxesSubplot at 0x17ea6e2a940>



From this plot we get a decent idea of the distribution of math scores but can't actually see the values so this is a poor visualization.

In [98]:
```python
sns.kdeplot(students["math score"])
```

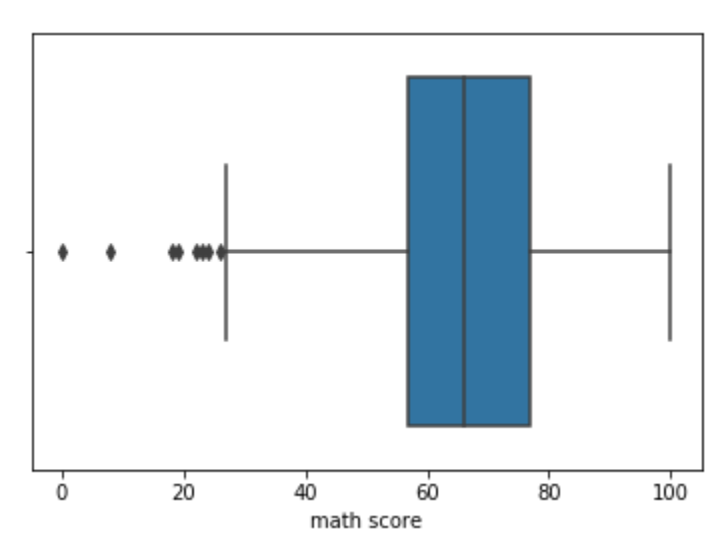
Out[98]: <matplotlib.axes._subplots.AxesSubplot at 0x17eb5280550>



This plot is much cleaner than the one above and we can clearly see the values now, however we don't get to visually see where the actual test results ended up, and we also can't visually see the outliers.

In [68]:
```python
sns.boxplot(x = "math score", data = students)
```

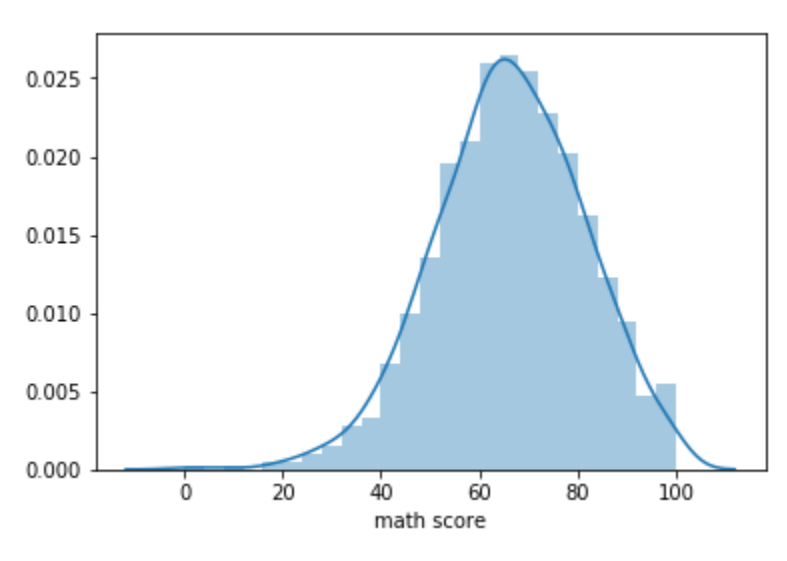
Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x17e8b64df98>



The boxplot is another good representation of the distribution of the data with the added benefit of showing the outliers.

In [69]:
```python
sns.distplot(students["math score"])
```

Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x17e8b904a58>

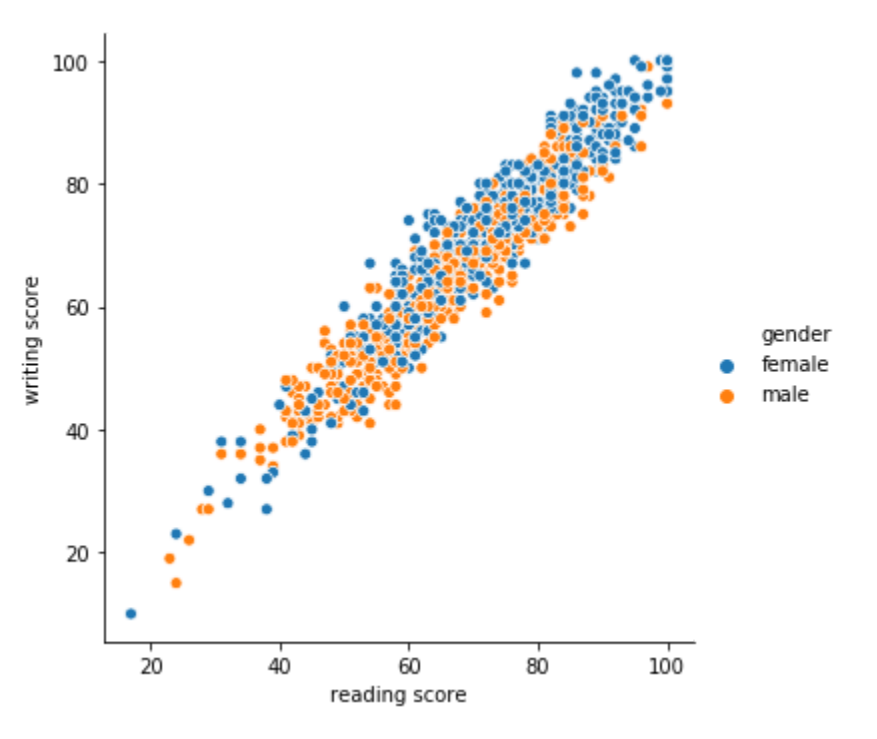


This plot gives a good representation of the distribution of the data however it doesn't show us anything more than that. It would be good to see the relationship between math score and other variables.

In [74]:
```python
##Choose two continuous variables, and plot them three different ways.
##Give the pros and cons of each plot you create. You can use variables from multiple datasets if you like.

sns.relplot(x = "reading score", y = "writing score", hue = "gender", data = students)
```

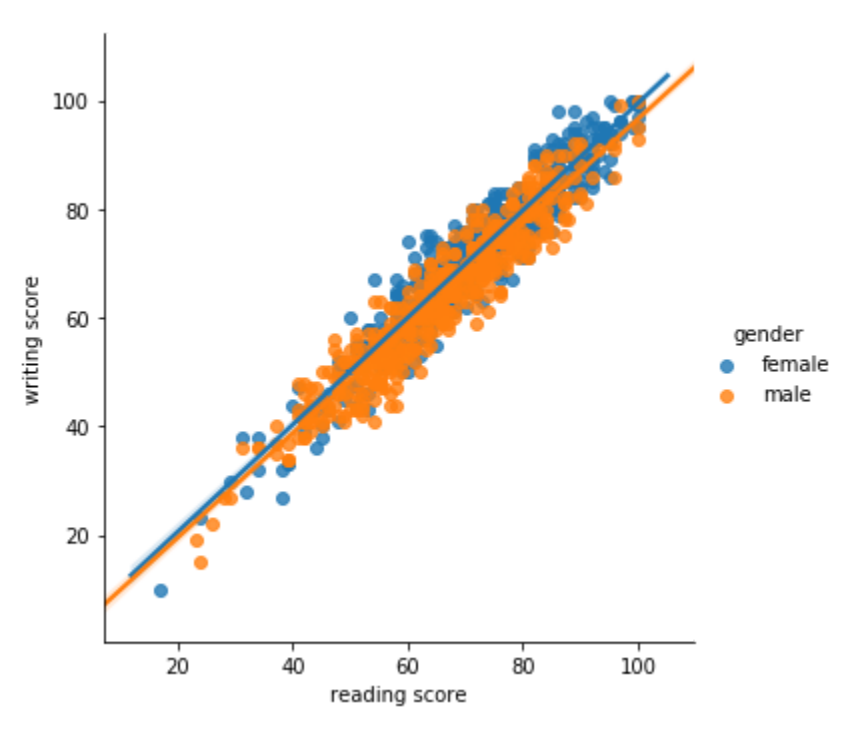
Out[74]: <seaborn.axisgrid.FacetGrid at 0x17e8cda0ef0>



This plot gives a good idea of the correlation between reading score and writing score however it would be nice to see the regression lines for both genders.

In [75]:
```python
sns.lmplot(x = "reading score", y = "writing score", hue = "gender", data = students)
```

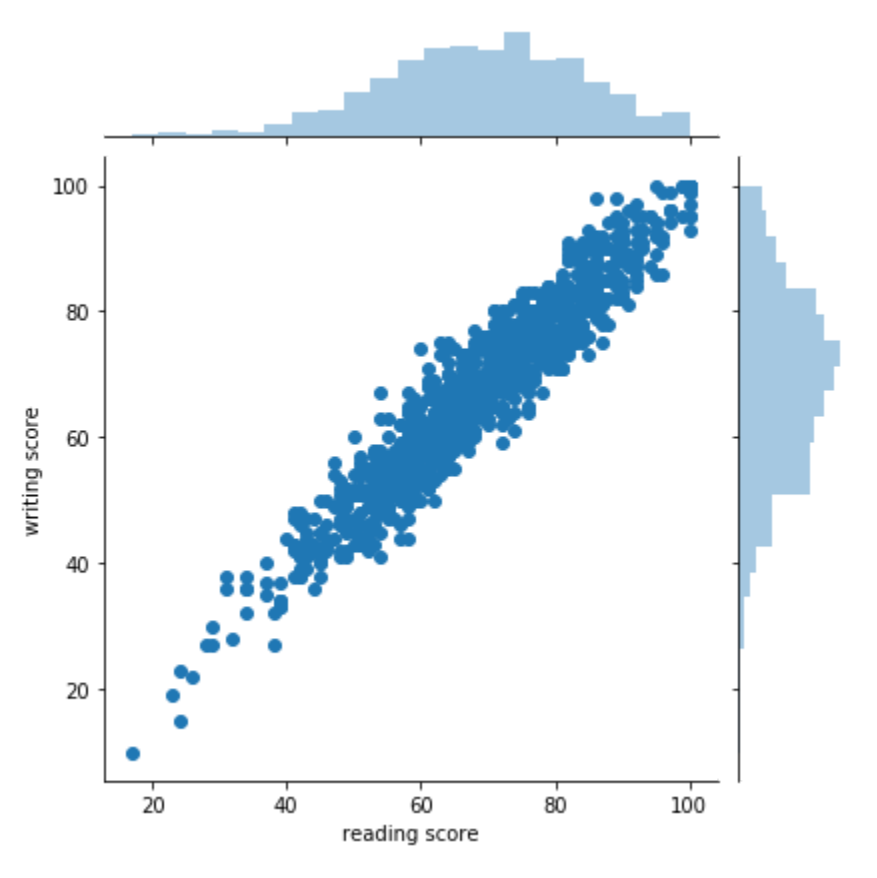
Out[75]: <seaborn.axisgrid.FacetGrid at 0x17e8bd32a58>



This plot we see the regression lines for both genders which is nice, but the actual points are not as clean as the plot above so visually it doesn't look as appealing.

In [73]:
```python
sns.jointplot(x = "reading score", y = "writing score", data = students)
```

Out[73]: <seaborn.axisgrid.JointGrid at 0x17e8b957b00>

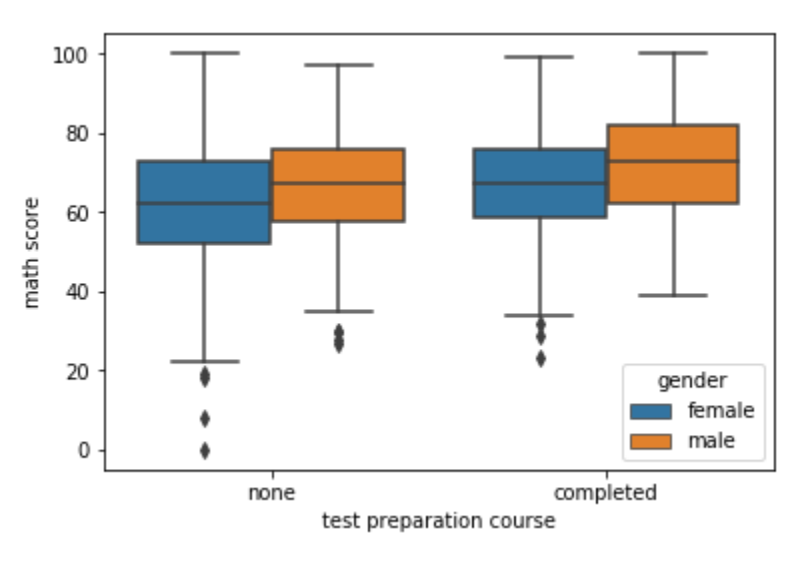


I like this plot because not only do we see how the two variables are correlated, but we also get the distribution of each variable. It would be better if we could break the data further like above and see for instance, the difference between genders.

In [76]:
```python
##Choose one continuous variable and one categorical variable, and plot them six different ways.
##Give the pros and cons of each plot you create. You can use variables from multiple datasets if you like.

sns.boxplot(x = "test preparation course", y = "math score", hue = "gender", data = students)
```

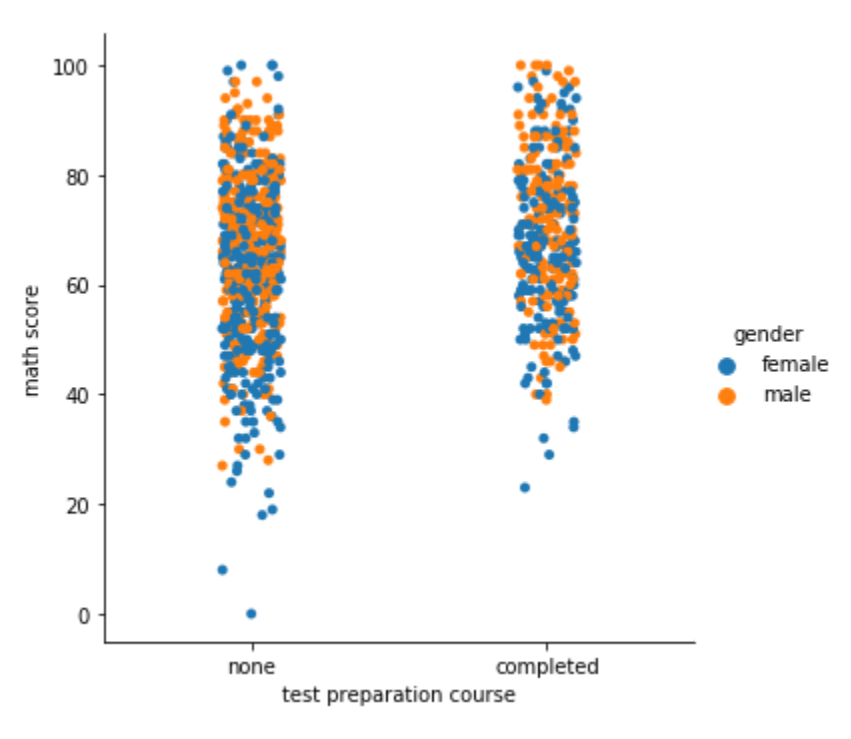
Out[76]: <matplotlib.axes._subplots.AxesSubplot at 0x17e8d475b38>



The boxplots give a good representation of the math scores based off gender and whether a test preparation course was completed.

In [77]:
```python
sns.catplot(x = "test preparation course", y = "math score", hue = "gender", kind = 'strip', data = students)
```

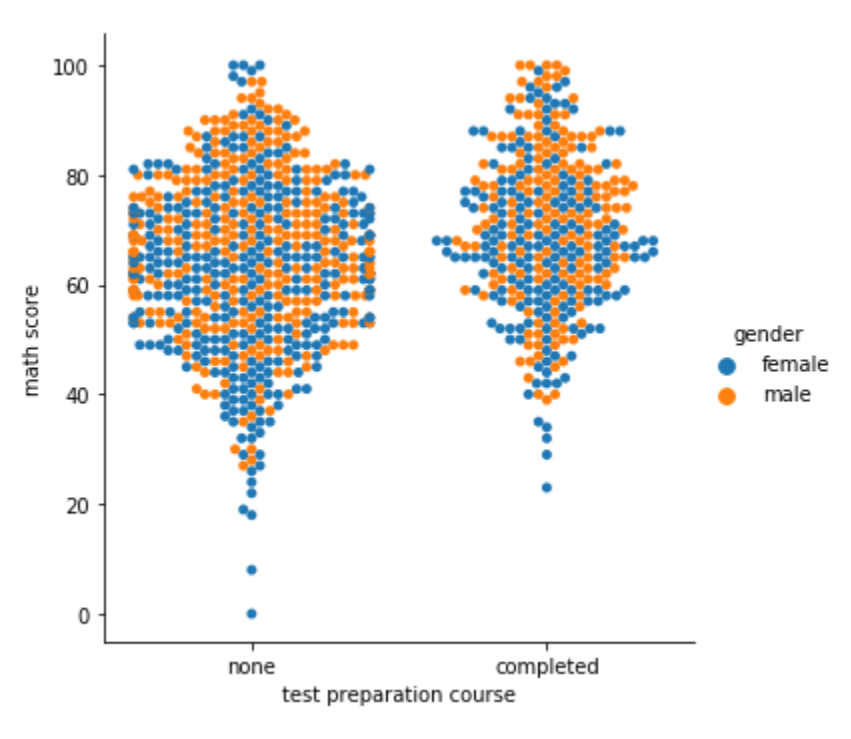
Out[77]: <seaborn.axisgrid.FacetGrid at 0x17e8d4a3e10>



This plot does a good job at breaking the math scores down based off of gender and whether a test preparation course was taken, however it is hard to see the distribution of the scores.

In [78]:
```python
sns.catplot(x = "test preparation course", y = "math score", hue = "gender", kind = 'swarm', data = students)
```

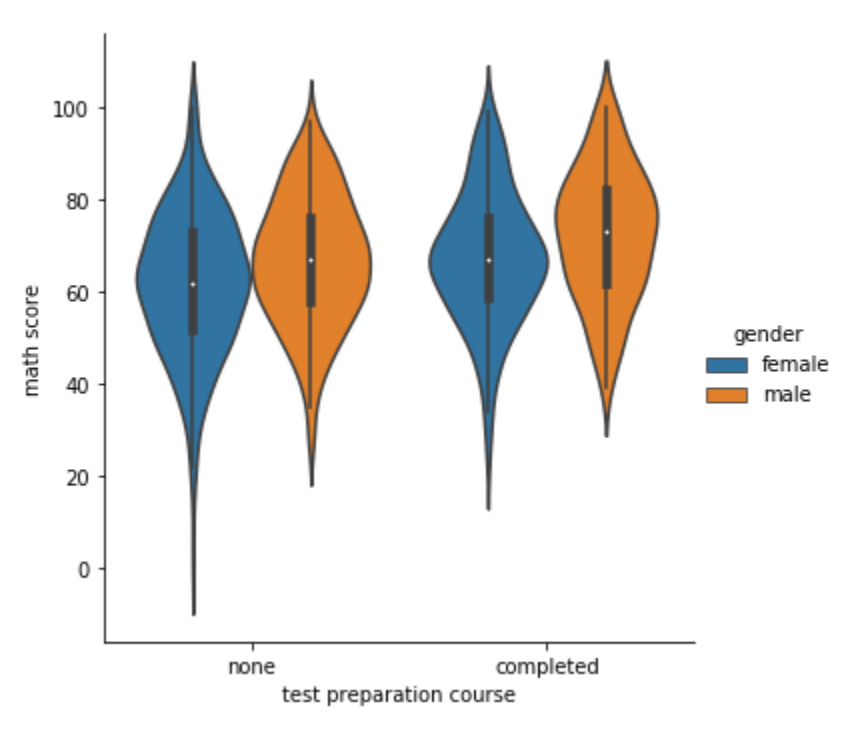
Out[78]: <seaborn.axisgrid.FacetGrid at 0x17e8d53e7b8>



This plot has all the advantages of the above plot with the added benefit of getting a glimpse at the distribution of the data. I don't find it as visually pleasing as the violin plots however.

In [79]:
```python
sns.catplot(x = "test preparation course", y = "math score", hue = "gender", kind = 'violin', data = students)
```

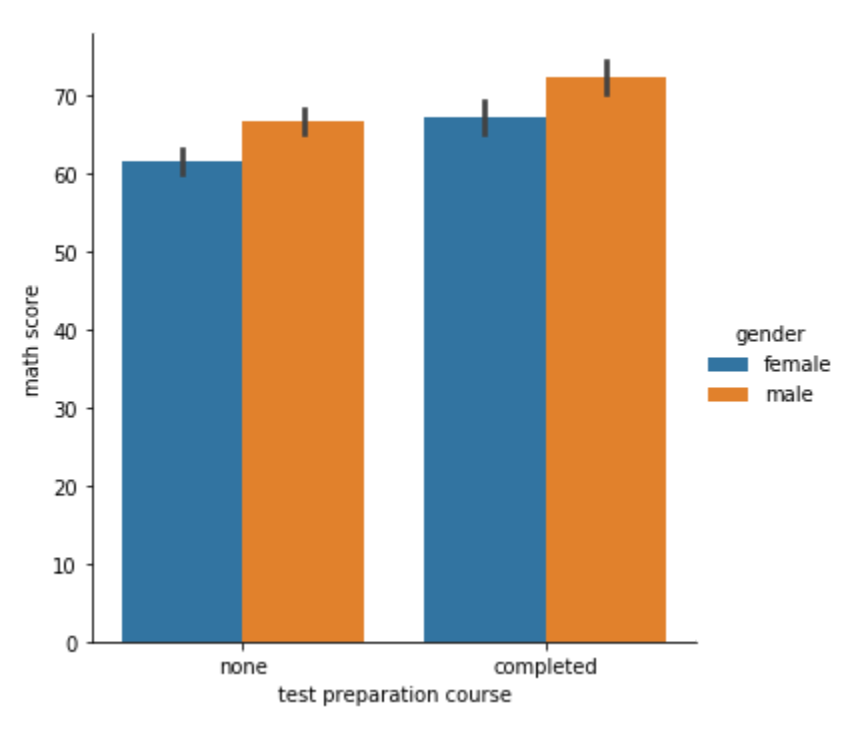
Out[79]: <seaborn.axisgrid.FacetGrid at 0x17e8d5a7da0>



I like this plot because it separates based off gender and you get a nice feeling for the distribution of math scores for the separate categories. The one drawback is we don't actually see the idividual outliers.

In [80]:
```python
sns.catplot(x = "test preparation course", y = "math score", hue = "gender", kind = 'bar', data = students)
```

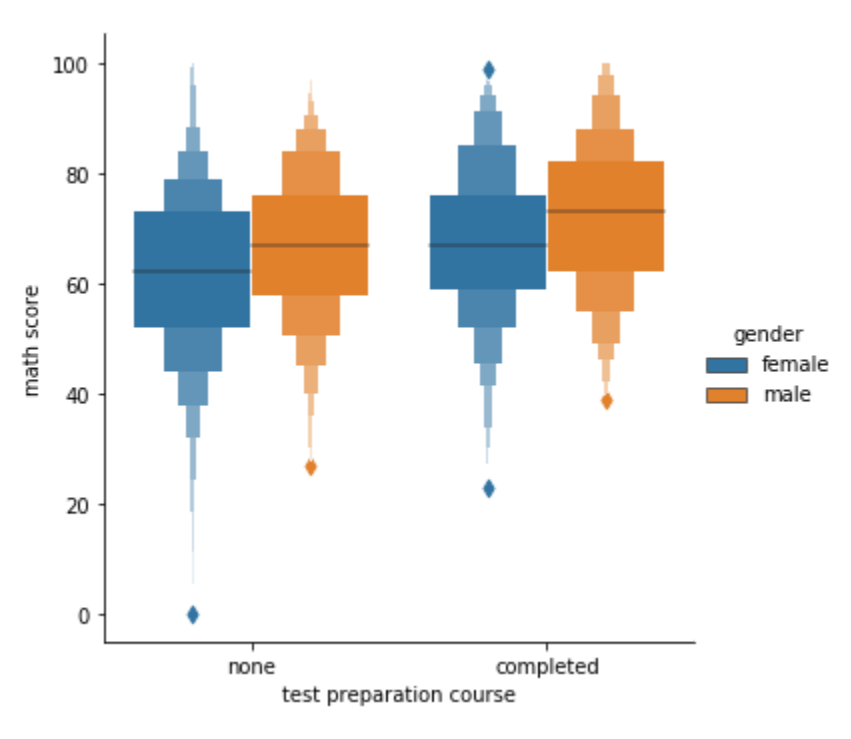
Out[80]: <seaborn.axisgrid.FacetGrid at 0x17e8d550f60>



This plot shows a the difference between the categories in terms of math scores but that is about it. We don't get a sense for the distribution of the data or the outliers or where any of the individual scores lie.

In [81]:
```python
sns.catplot(x = "test preparation course", y = "math score", hue = "gender", kind = 'boxen', data = students)
```

Out[81]: <seaborn.axisgrid.FacetGrid at 0x17e8d642f98>



I like this plot the most. I think it combines the best elements of the box plots and the violin plots. We see the outliers and we also get a good visual of the distributions and I feel it is the most asthetically pleasing of the plots.

In [ ]:
```python
##Find a dataset with at least four continuous variables and one categorical variable.
##Create one master plot that gives insight into the variables and their interrelationships, including:
##Probability distributions
##Bivariate relationships
##Whether the distributions or the relationships vary across groups
##Accompany your plot with a written description of what you see.
```

In [134]:
```python
shots = pd.read_csv(r'C:\Users\mike1\Documents\Thinkful\shot_logs.csv', low_memory = False)
shots.head()
```

Out[134]:

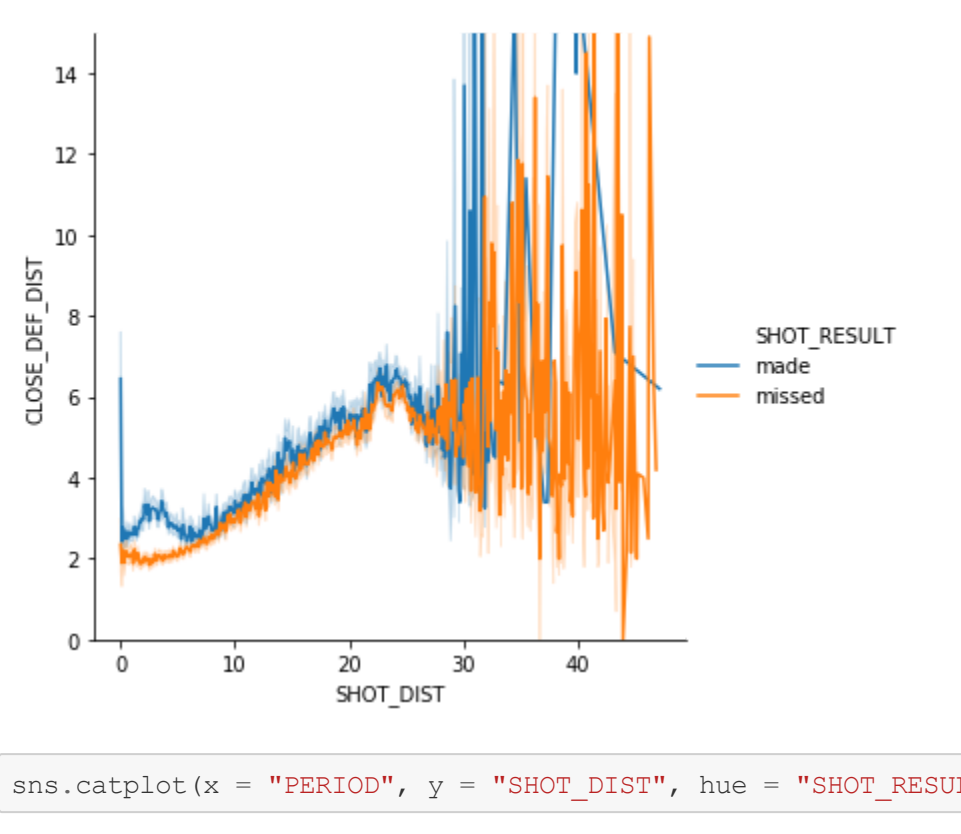
	GAME_ID	MATCHUP	LOCATION	W	FINAL_MARGIN	SHOT_NUMBER	PERIOD	GAME_CLOCK	SHOT_CLOCK	DRIBBLES	...	SHOT_DIST	PTS_TYPE	SI
0	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	1	1	1:09	10.8	2	...	7.7	2	
1	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	2	1	0:14	3.4	0	...	28.2	3	
2	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	3	1	0:00	NaN	3	...	10.1	2	
3	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	4	2	11:47	10.3	2	...	17.2	2	
4	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	5	2	10:34	10.9	2	...	3.7	2	

5 rows × 21 columns

In [146]:
```python
sns.relplot(x="SHOT_DIST", y="CLOSE_DEF_DIST",
            hue = "SHOT_RESULT", kind = "line",
            legend="full", data=shots)

plt.ylim(0, 15)
```

Out[146]: (0, 15)



In [147]:
```python
sns.catplot(x = "PERIOD", y = "SHOT_DIST", hue = "SHOT_RESULT", kind = 'violin', data = shots)
```

Out[147]: <seaborn.axisgrid.FacetGrid at 0x17eacede748>