In [31]:
```
import pandas as pd
from seaborn import regplot
import matplotlib.pyplot as plt
import scipy
from scipy.stats import pearsonr
import numpy as np
import seaborn as sns
import scipy.stats as sps
```

In [3]:
```
data = pd.read_csv("Desktop\exams.csv")
data.head(10)
```

Out[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	male	group A	high school	standard	completed	67	67	63
1	female	group D	some high school	free/reduced	none	40	59	55
2	male	group E	some college	free/reduced	none	59	60	50
3	male	group B	high school	standard	none	77	78	68
4	male	group E	associate's degree	standard	completed	78	73	68
5	female	group D	high school	standard	none	63	77	76
6	female	group A	bachelor's degree	standard	none	62	59	63
7	male	group E	some college	standard	completed	93	88	84
8	male	group D	high school	standard	none	63	56	65
9	male	group C	some college	free/reduced	none	47	42	45

In [6]:
```
data.info()
df=data.copy()
Gender = {'male': 0,'female': 1}
df.gender = [Gender[item] for item in df.gender]
df.head(4)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64 
 6   reading score                1000 non-null   int64 
 7   writing score                1000 non-null   int64 
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

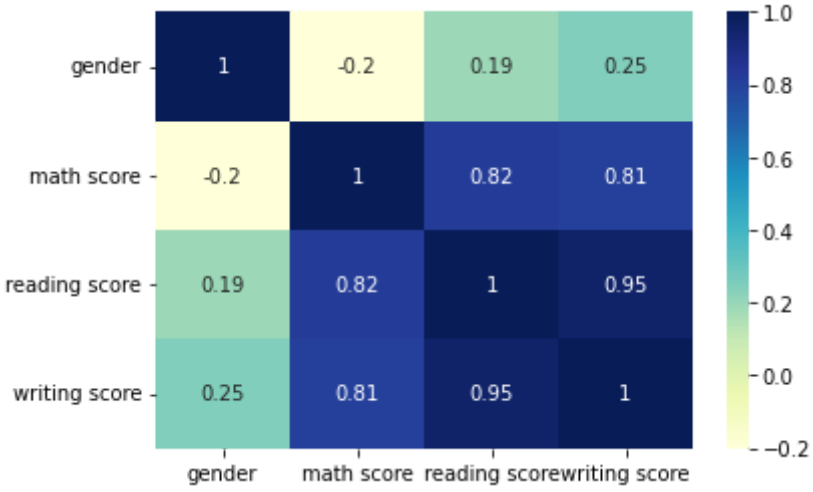
Out[6]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	0	group A	high school	standard	completed	67	67	63
1	1	group D	some high school	free/reduced	none	40	59	55
2	0	group E	some college	free/reduced	none	59	60	50
3	0	group B	high school	standard	none	77	78	68

In [12]:
```
print(df.corr())
dataplot = sns.heatmap(df.corr(), cmap="YlGnBu", annot=True)
```

```
                gender  math score  reading score  writing score
gender        1.000000   -0.200863       0.189389       0.246089
math score   -0.200863    1.000000       0.819398       0.805944
reading score 0.189389    0.819398       1.000000       0.954274
writing score 0.246089    0.805944       0.954274       1.000000
```



In [11]:
```
from scipy.stats import pearsonr
print('Assoc. - gender and writing score')
print(pearsonr(df['gender'], df['writing score']))

print('Assoc. - between reading score and writing score')
print(pearsonr(df['reading score'], df['writing score']))
```

```
Assoc. - gender and writing score
(0.24608898692830694, 2.9267319154280487e-15)
Assoc. - between reading score and writing score
(0.9542744344566843, 0.0)
```

In [25]:
```
data['math score'] = pd.cut(data['math score'] ,4,labels = ['Low','Medium','High','E
```

In [26]:
```
Zippedlist = list(zip(data['gender'],data['math score']))
GenderMathScore = pd.DataFrame(Zippedlist,columns=['gender','math score'])
GenderMathScorePivot = GenderMathScore.reset_index().groupby(['gender','math score']
GenderMathScorePivot
GenderMathScoreContigencyTable = GenderMathScorePivot.pivot(index='gender', columns=
GenderMathScoreContigencyTable
GenderMathScoreContigencyTable.fillna(0,inplace = True)
```

In [27]:

```
GenderMathScoreContigencyTable
```

Out[27]:

math score	Excellent	High	Low	Medium
gender				
female	86	236	18	143
male	135	278	2	102

In [32]:

```
chi2,p,dof,expected =sps.chi2_contingency(GenderMathScoreContigencyTable, correction
chi2,p,dof,expected
```

Out[32]:

```
(32.83934678312836,
 3.482021533987175e-07,
 3,
 array([[106.743, 248.262,   9.66 , 118.335],
        [114.257, 265.738,  10.34 , 126.665]]))
```

In [ ]: