Part_I_notebook

June 26, 2022

1 Part I - PISA 2012 Data Exploration

1.1 by Jaclyn Tobin

1.2 Introduction

PISA Data: PISA is a survey of students' skills and knowledge as they approach the end of compulsory education. This survey examines how well students have learned the school curriculum, how well prepared they are for life beyond school. Around 510,000 students in 65 economies took part in the PISA 2012 assessment of reading, mathematics and science.

1.3 Preliminary Wrangling

```
In [2]: # import all packages and set plots to be embedded inline
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

        %matplotlib inline
```

```
In [5]: df=pd.read_csv('pisa2012.csv', encoding = "ISO-8859-1")
```

```
/opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2785: DtypeWarning: Colu
  interactivity=interactivity, compiler=compiler, result=result)
```

```
In [6]: df.shape
```

```
Out[6]: (485490, 636)
```

```
In [7]: #create dataframe with selected columns
        df1=df[['ISCEDL','STIDSTD','CNT','ST04Q01','ST08Q01','ST09Q01','ST13Q01','ST17Q01','ST15
```

```
In [9]: #save data I will use to smaller csv file
        df1.to_csv('my_pisa.csv', index= False)
```

```
In [3]: df2=pd.read_csv('my_pisa.csv')
```

```
In [4]: !pip install seaborn --upgrade
```

```
Collecting seaborn
  Downloading https://files.pythonhosted.org/packages/10/5b/0479d7d845b5ba410ca702ffcd7f2cd95a14
    100% || 296kB 18.7MB/s ta 0:00:01
Collecting numpy>=1.15 (from seaborn)
  Downloading https://files.pythonhosted.org/packages/45/b2/6c7545bb7a38754d63048c7696804a0d9473
    100% || 13.4MB 2.5MB/s eta 0:00:01   17% |                          | 2.4MB 27.7MB/s eta 0:0
Requirement already satisfied, skipping upgrade: scipy>=1.0 in /opt/conda/lib/python3.6/site-pac
Collecting matplotlib>=2.2 (from seaborn)
  Downloading https://files.pythonhosted.org/packages/09/03/b7b30fa81cb687d1178e085d0f01111ceaea
    100% || 11.5MB 3.3MB/s eta 0:00:01   42% |                  | 4.9MB 25.4MB/s eta 0:00:01
Requirement already satisfied, skipping upgrade: pandas>=0.23 in /opt/conda/lib/python3.6/site-p
Collecting pillow>=6.2.0 (from matplotlib>=2.2->seaborn)
  Downloading https://files.pythonhosted.org/packages/7d/2a/2fc11b54e2742db06297f7fa7f420a0e3069
    100% || 49.4MB 578kB/s ta 0:00:011  4% |                              | 2.4MB 25.0MB/s eta 0
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in /opt/conda/lib/python3.
Collecting kiwisolver>=1.0.1 (from matplotlib>=2.2->seaborn)
  Downloading https://files.pythonhosted.org/packages/a7/1b/cbd8ae738719b5f41592a12057ef5442e2ed
    100% || 1.1MB 12.3MB/s ta 0:00:01
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /op
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /opt/conda/lib/python3.6/site-p
Requirement already satisfied, skipping upgrade: pytz>=2011k in /opt/conda/lib/python3.6/site-pa
Requirement already satisfied, skipping upgrade: six>=1.5 in /opt/conda/lib/python3.6/site-packa
Building wheels for collected packages: pillow
  Running setup.py bdist_wheel for pillow ... done
  Stored in directory: /root/.cache/pip/wheels/a7/69/9a/bba9fca6782340f88dbc378893095722a663cbc6
Successfully built pillow
tensorflow 1.3.0 requires tensorflow-tensorboard<0.2.0,>=0.1.0, which is not installed.
scikit-image 0.14.2 has requirement dask[array]>=1.0.0, but you'll have dask 0.16.1 which is inc
Installing collected packages: numpy, pillow, kiwisolver, matplotlib, seaborn
  Found existing installation: numpy 1.12.1
    Uninstalling numpy-1.12.1:
      Successfully uninstalled numpy-1.12.1
  Found existing installation: Pillow 5.2.0
    Uninstalling Pillow-5.2.0:
      Successfully uninstalled Pillow-5.2.0
  Found existing installation: matplotlib 2.1.0
    Uninstalling matplotlib-2.1.0:
      Successfully uninstalled matplotlib-2.1.0
  Found existing installation: seaborn 0.8.1
    Uninstalling seaborn-0.8.1:
      Successfully uninstalled seaborn-0.8.1
Successfully installed kiwisolver-1.3.1 matplotlib-3.3.4 numpy-1.19.5 pillow-8.4.0 seaborn-0.11.
```

```
In [14]: print(df2.shape)
         print(df2.dtypes)
         print(df2.head())
```

```
(485490, 20)
ISCED_LEVEL     object
ID               int64
COUNTRY         object
GENDER          object
LATE            object
SKIP            object
M_EDU           object
F_EDU           object
M_JOB           object
F_JOB           object
STUDY_AREA      object
COMPUTER        object
INTERNET        object
TEXTBOOKS       object
CHESS           object
PROGRAM         object
MATH           float64
READING        float64
SCIENCE        float64
WEALTH         float64
dtype: object
     ISCED_LEVEL  ID  COUNTRY  GENDER               LATE    SKIP  \
0  ISCED level 3   1  Albania  Female              None    None
1  ISCED level 3   2  Albania  Female  One or two times    None
2  ISCED level 2   3  Albania  Female              None    None
3  ISCED level 2   4  Albania  Female              None    None
4  ISCED level 2   5  Albania  Female  One or two times    None

                                 M_EDU                  F_EDU  \
0                     <ISCED level 3A>       <ISCED level 3A>
1                     <ISCED level 3A>       <ISCED level 3A>
2                 <ISCED level 3B, 3C>       <ISCED level 3A>
3                 <ISCED level 3B, 3C>       <ISCED level 3A>
4  She did not complete <ISCED level 1>   <ISCED level 3B, 3C>

                              M_JOB                         F_JOB  \
0  Other (e.g. home duties, retired)    Working part-time <for pay>
1        Working full-time <for pay>   Working full-time <for pay>
2        Working full-time <for pay>   Working full-time <for pay>
3        Working full-time <for pay>   Working full-time <for pay>
4         Working part-time <for pay>   Working part-time <for pay>

  STUDY_AREA COMPUTER INTERNET TEXTBOOKS            CHESS          PROGRAM  \
0        Yes       No       No       Yes  Never or rarely  Never or rarely
1        Yes      Yes      Yes       Yes  Never or rarely  Never or rarely
2        Yes      Yes      Yes       Yes  Never or rarely  Never or rarely
3        Yes      Yes      Yes       Yes              NaN              NaN
```

```
4          No       Yes       Yes       Yes                NaN         Sometimes

       MATH   READING   SCIENCE  WEALTH
0  406.8469  249.5762  341.7009   -2.92
1  486.1427  406.2936  548.9929    0.69
2  533.2684  401.2100  499.6643   -0.23
3  412.2215  547.3630  438.6796   -1.17
4  381.9209  311.7707  361.5628   -1.17
```

1.3.1 What is the structure of your dataset?

The original dataset had 636 columns and 485438 rows. After selecting the columns I want to work with I know have a dataframe with only 20 columns and around 485438 rows. This represents over 480000 students who took the PISA survey in 2012.

1.3.2 What is/are the main feature(s) of interest in your dataset?

The main feature of this dataset is a measure of these students performance in math, reading and science. Beyond these scores there are many different features of each students life, habits, and outlooks.

1.3.3 What features in the dataset do you think will help support your investigation into your feature(s) of interest?

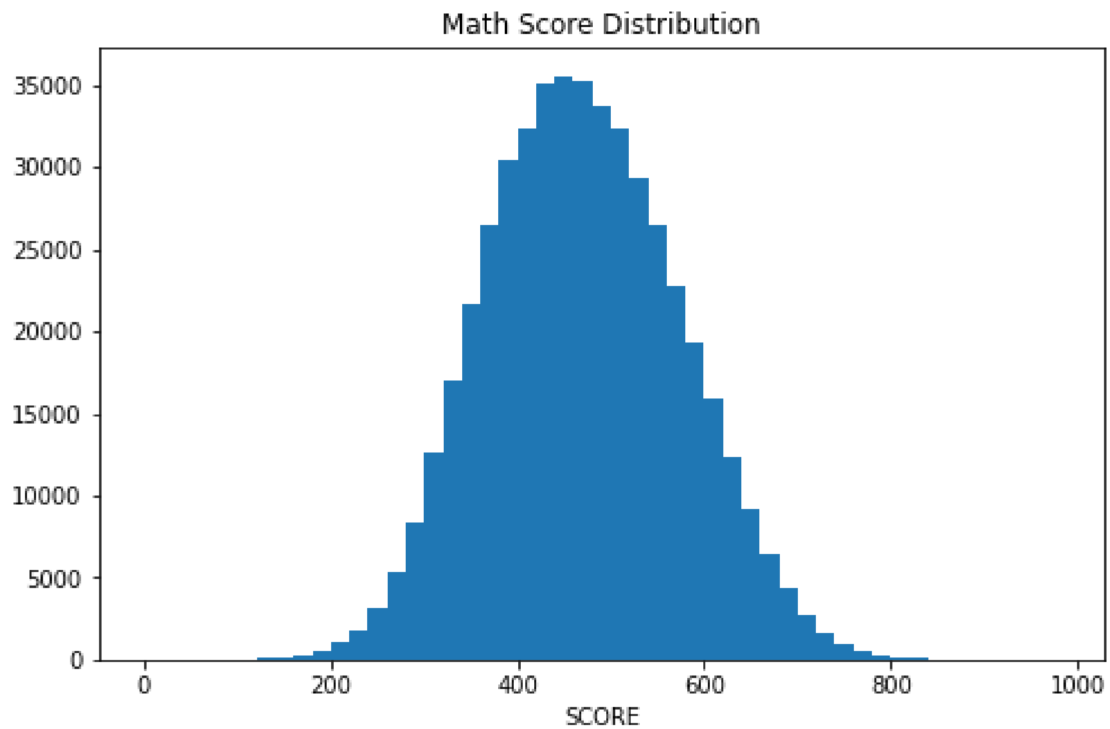
For my invesitgation into this dataset I am going to look for correlations between multiple features and test scores for each student. This may include how the following correlate with each students math, reading, and science scores: * Family wealth * Gender * Country * If the student is late or skips classes * Mother and Father's education levels * Mother and Father's employment status

1.4 Univariate Exploration

1.4.1 Question #1

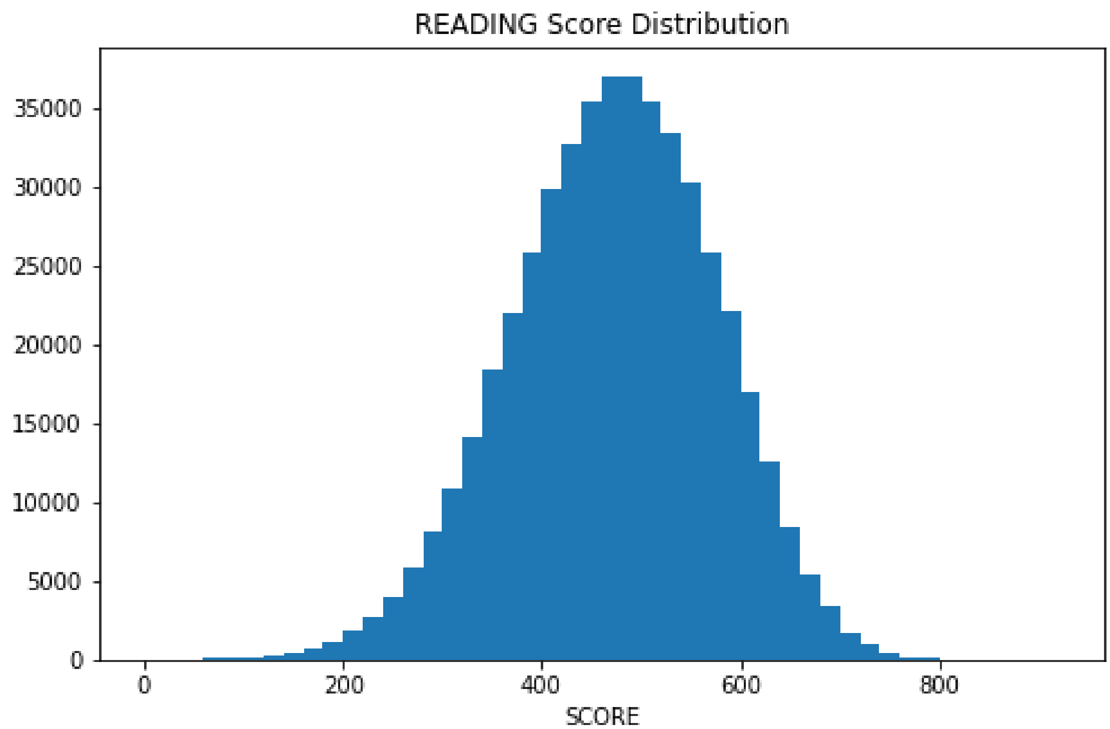
I first want to look at the distrubaution of the students math, reading and science test scores.

```
In [3]: #viz for math scores
        binsize = 20
        bins = np.arange(0, df2['MATH'].max()+binsize, binsize)

        plt.figure(figsize=[8, 5])
        plt.hist(data = df2, x = 'MATH', bins = bins)
        plt.xlabel('SCORE')
        plt.title('Math Score Distribution')
        plt.show()

        print(df2.MATH.describe())
```



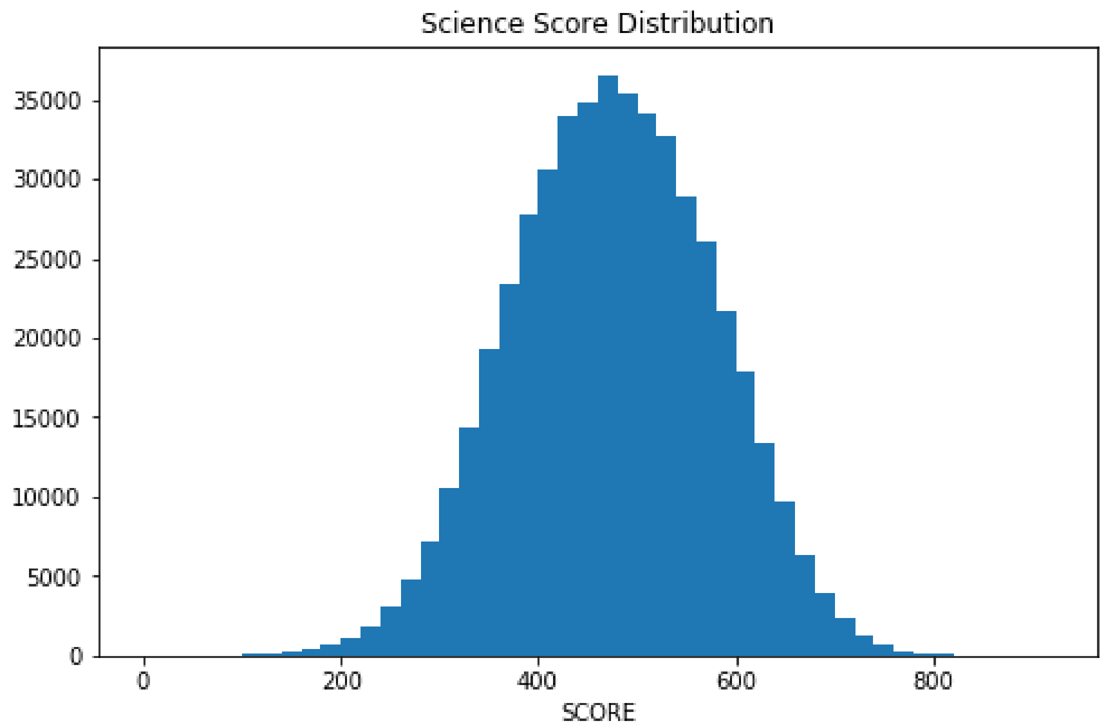

```
count    485490.000000
mean        469.621653
std         103.265391
min          19.792800
25%         395.318600
50%         466.201900
75%         541.057800
max         962.229300
Name: MATH, dtype: float64
```

```
In [73]: #viz for reading scores
         binsize = 20
         bins = np.arange(0, df2['READING'].max()+binsize, binsize)

         plt.figure(figsize=[8, 5])
         plt.hist(data = df2, x = 'READING', bins = bins)
         plt.xlabel('SCORE')
         plt.title('READING Score Distribution')
         plt.show()

         print(df2.READING.describe())
```




```
count    485490.000000
mean        472.004640
std         102.505523
min           0.083400
25%         403.600700
50%         475.455000
75%         544.502500
max         904.802600
Name: READING, dtype: float64
```

```
In [9]: #viz for science scores
        binsize = 20
        bins = np.arange(0, df2['SCIENCE'].max()+binsize, binsize)

        plt.figure(figsize=[8, 5])
        plt.hist(data = df2, x = 'SCIENCE', bins = bins)
        plt.xlabel('SCORE')
        plt.title('Science Score Distribution')
        plt.show()

        print(df2.SCIENCE.describe())
```




```
count    485490.000000
mean        475.769824
std         101.464426
min           2.648300
25%         404.457300
50%         475.699400
75%         547.780700
max         903.338300
Name: SCIENCE, dtype: float64
```

1.4.2 Observations

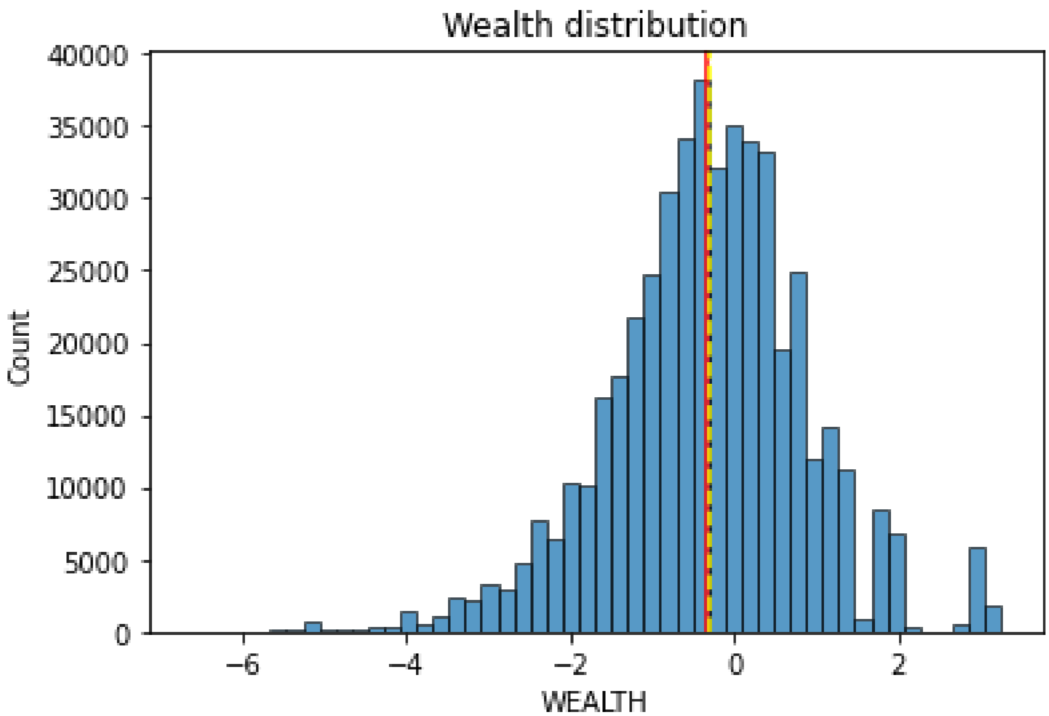
The math, reading and science test scores all seem to have a normal distribution. The median test scores are similar in math (469), reading (472), and science (475) as well.

1.4.3 Question #2

What does the wealth distibution look like?

```
In [4]: #viz for wealth

        import seaborn as sns
```

```
sns.histplot(data=df2, x="WEALTH", bins=50).set(title="Wealth distribution")
plt.axvline(x=df2.WEALTH.mean(),
            color='red')
plt.axvline(x=df2.WEALTH.median(),
            color='yellow',
            ls='--',)
plt.show()

print(df2.WEALTH.describe())
```




```
count    479597.00000
mean         -0.33701
std           1.21530
min          -6.65000
25%          -1.04000
50%          -0.30000
75%           0.43000
max           3.25000
Name: WEALTH, dtype: float64
```

1.4.4 Observations

Wealth is normally distributed. Although the histagram looks slightly left skewed, the mean and median are very close to each other, indicating a normal distribution.

1.4.5 Question #3

Which countries had the most and least participation in the survey?

```
In [11]: #viz for countries

         #I need to reassign some states that were listed separately
         states={'Florida (USA)':'United States of America','Connecticut (USA)':'United States o
         df2['COUNTRY'] = df2['COUNTRY'].replace(states)

         large=df2.COUNTRY.value_counts().nlargest(10).index.tolist()
         small=df2.COUNTRY.value_counts().nsmallest(10).index.tolist()

         fig, ax = plt.subplots(nrows=2, figsize = [8,8], constrained_layout=True)
         default_color = sns.color_palette()[0]
         sns.countplot(data = df2[df2.COUNTRY.isin(small)], x = 'COUNTRY', order= small, color =
         sns.countplot(data = df2[df2.COUNTRY.isin(large)], x = 'COUNTRY', order= large, color =
         for ax in fig.axes:
             plt.sca(ax)
             plt.xticks(rotation=45)
```

```
        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-11-bf7a039c7eed> in <module>()
          8 small=df2.COUNTRY.value_counts().nsmallest(10).index.tolist()
          9
    ---> 10 fig, ax = plt.subplots(nrows=2, figsize = [8,8], constrained_layout=True)
         11 default_color = sns.color_palette()[0]
         12 sns.countplot(data = df2[df2.COUNTRY.isin(small)], x = 'COUNTRY', order= small, colo


        /opt/conda/lib/python3.6/site-packages/matplotlib/pyplot.py in subplots(nrows, ncols, sh
       1177     subplot
       1178     """
    -> 1179     fig = figure(**fig_kw)
       1180     axs = fig.subplots(nrows=nrows, ncols=ncols, sharex=sharex, sharey=sharey,
       1181                        squeeze=squeeze, subplot_kw=subplot_kw,


        /opt/conda/lib/python3.6/site-packages/matplotlib/pyplot.py in figure(num, figsize, dpi,
        532                                         frameon=frameon,
        533                                         FigureClass=FigureClass,
    --> 534                                         **kwargs)
        535
        536         if figLabel:
```

```
/opt/conda/lib/python3.6/site-packages/matplotlib/backend_bases.py in new_figure_manager
167         from matplotlib.figure import Figure
168         fig_cls = kwargs.pop('FigureClass', Figure)
--> 169         fig = fig_cls(*args, **kwargs)
170         return cls.new_figure_manager_given_figure(num, fig)
171


TypeError: __init__() got an unexpected keyword argument 'constrained_layout'
```

In [16]:
```python
print(df2.COUNTRY.value_counts().nsmallest(10))
print(df2.COUNTRY.value_counts().nlargest(10))
```

```
Liechtenstein                293
Perm(Russian Federation)    1761
Iceland                     3508
New Zealand                 4291
Latvia                      4306
Tunisia                     4407
Netherlands                 4460
Costa Rica                  4602
Poland                      4607
France                      4613
Name: COUNTRY, dtype: int64
Mexico                  33806
Italy                   31073
Spain                   25313
Canada                  21544
Brazil                  19204
Australia               14481
United Kingdom          12659
United Arab Emirates    11500
Switzerland             11229
Qatar                   10966
Name: COUNTRY, dtype: int64
```

1.4.6 Observations

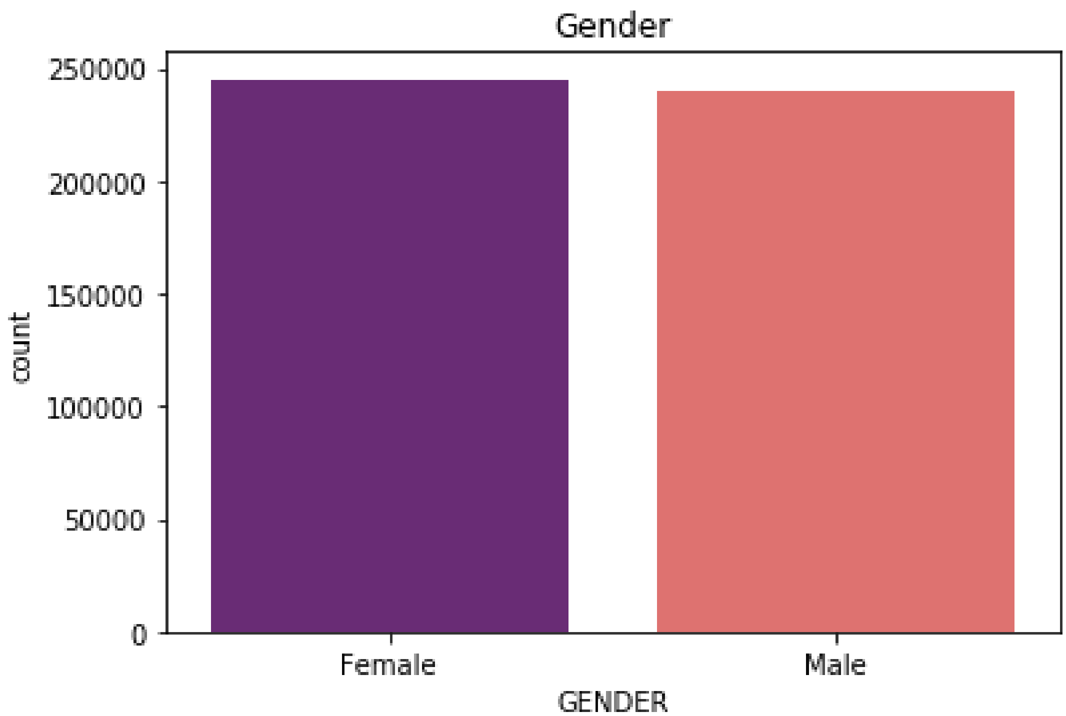
Mexico, Italy Spain, and Canada all had over 20,000 student participates. Iceland, Russia, and Liechtenshtein each had under 4000 student participants.

1.4.7 Question #4

Did more females or males take this survey?

```
In [17]: #viz for gender
         print(df2.GENDER.value_counts())
         sns.countplot(x = 'GENDER', data = df2, palette = 'magma')
         plt.title('Gender')
         plt.show()
```

```
Female    245064
Male      240426
Name: GENDER, dtype: int64
```



1.4.8 Obeservation

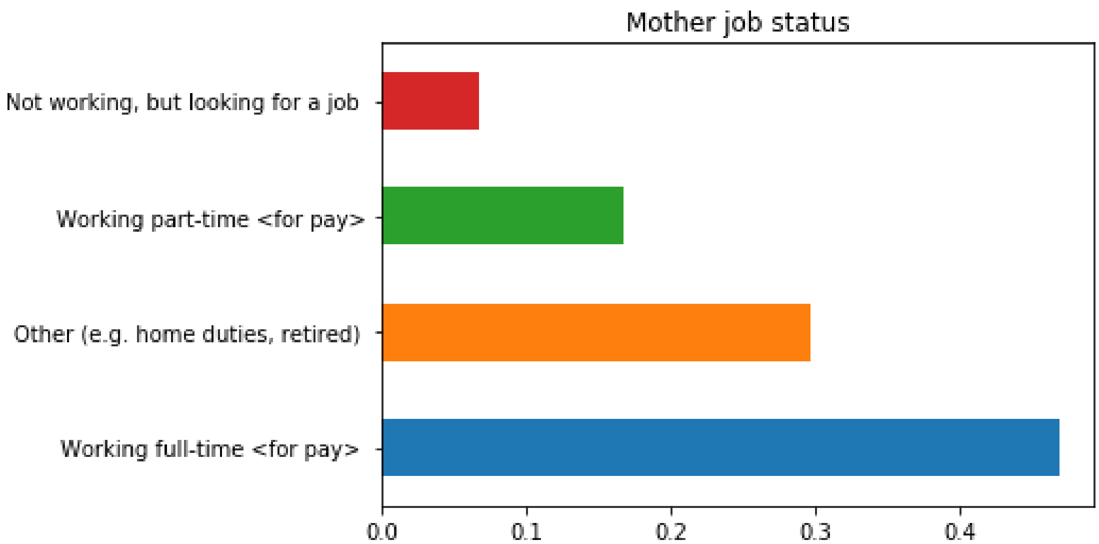
According to the above visulization there appears to be about the same amount of male and female survey participants.

1.4.9 Question #5

What does the mother/father job status distributions look like?

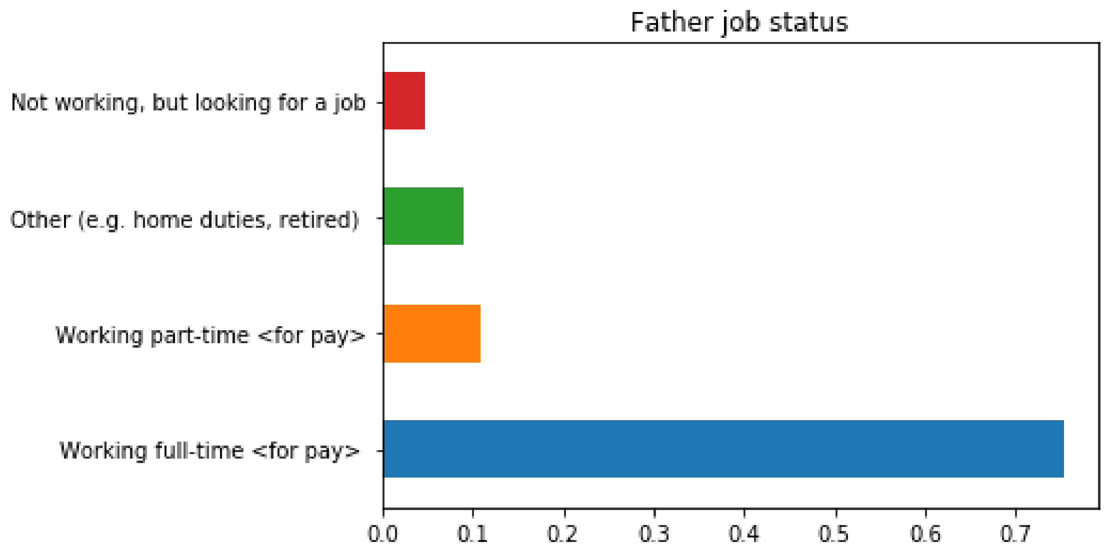
```
In [36]: #mother job status viz
         df2.M_JOB.value_counts(normalize= True).plot(kind='barh').set(title='Mother job status'
         print(df2.M_JOB.value_counts(normalize= True))
```

```
Working full-time <for pay>           0.468401
Other (e.g. home duties, retired)     0.296827
Working part-time <for pay>           0.167262
Not working, but looking for a job    0.067510
Name: M_JOB, dtype: float64
```




```
In [37]: #father job status viz
         df2.F_JOB.value_counts(normalize= True).plot(kind='barh').set(title='Father job status'
         print(df2.F_JOB.value_counts(normalize= True))
```

```
Working full-time <for pay>          0.752524
Working part-time <for pay>          0.109663
Other (e.g. home duties, retired)    0.089914
Not working, but looking for a job   0.047899
Name: F_JOB, dtype: float64
```



1.4.10 Observation

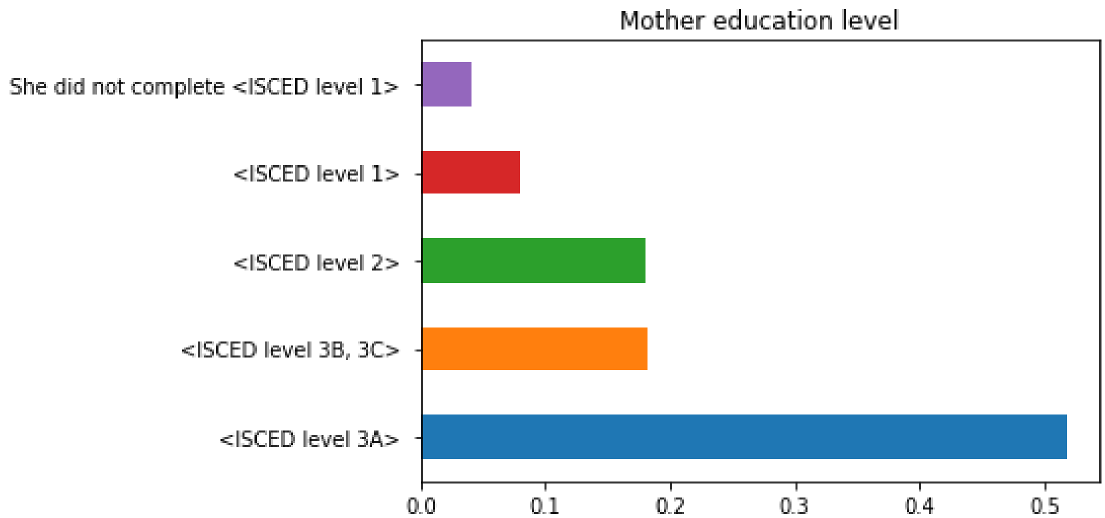
There is a large difference in the proportion of fathers working full time compared to mothers (75% fathers/47% mothers). Another large difference is the proportion of 'Others (home-duties, retired) 30% mothers/ 9% fathers.

1.4.11 Question #6

Is there a large difference in the education level of mothers and fathers?

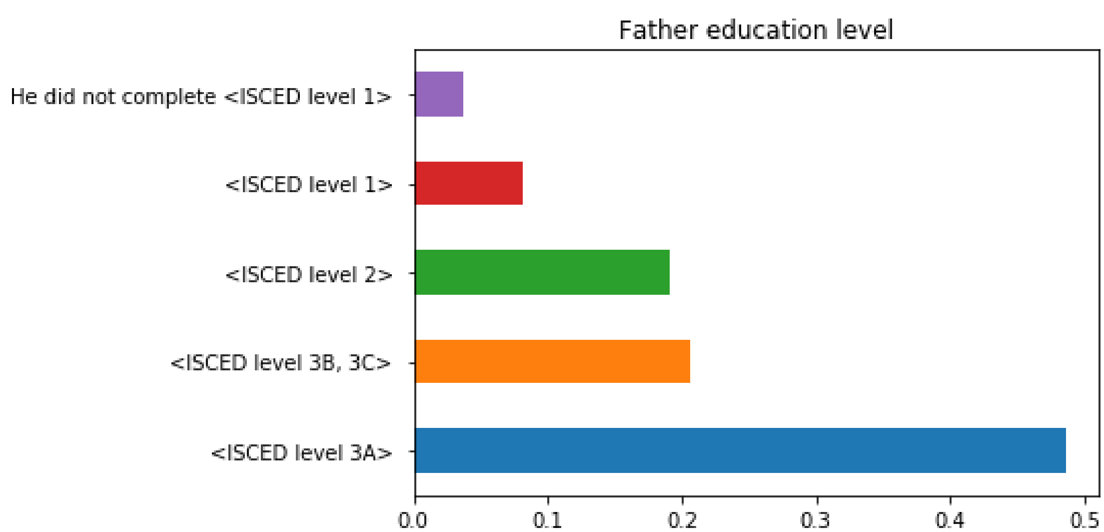
```
In [30]: #mother education level viz
         df2.M_EDU.value_counts().plot(kind='barh').set(title='Mother education level')
         print(df2.M_EDU.value_counts(normalize= True))
```

```
<ISCED level 3A>                        0.517476
<ISCED level 3B, 3C>                    0.181336
<ISCED level 2>                         0.180388
<ISCED level 1>                         0.079820
She did not complete <ISCED level 1>    0.040980
Name: M_EDU, dtype: float64
```




```
In [38]: #father education level viz
         df2.F_EDU.value_counts(normalize= True).plot(kind='barh').set(title='Father education l
         print(df2.F_EDU.value_counts(normalize= True))
```

```
<ISCED level 3A>                        0.485673
<ISCED level 3B, 3C>                    0.205700
<ISCED level 2>                         0.190247
<ISCED level 1>                         0.081076
He did not complete <ISCED level 1>     0.037303
Name: F_EDU, dtype: float64
```



1.4.12 Observation

There does not seem to be much of a difference between education levels for students mothers and fathers.

1.4.13 Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

The student test score distribution for all three exams followed a normal distribution. I thought it was unusal that wealth also followed a normal distribution so I accessed the literature on PISA 2012 and confirmed they had transformed this data on purpose to follow a normal distribution.

1.4.14 Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

I thought finding the best and worst performing countries was very interesting. However, when I first looked at the worst performing countries there were individual states listed. Because of this I had to tidy the data so I could compare countries to countries. I also normalized the count on parent job status and education level. Without normalizing the count, it was hard to look from one viz to another and compare.

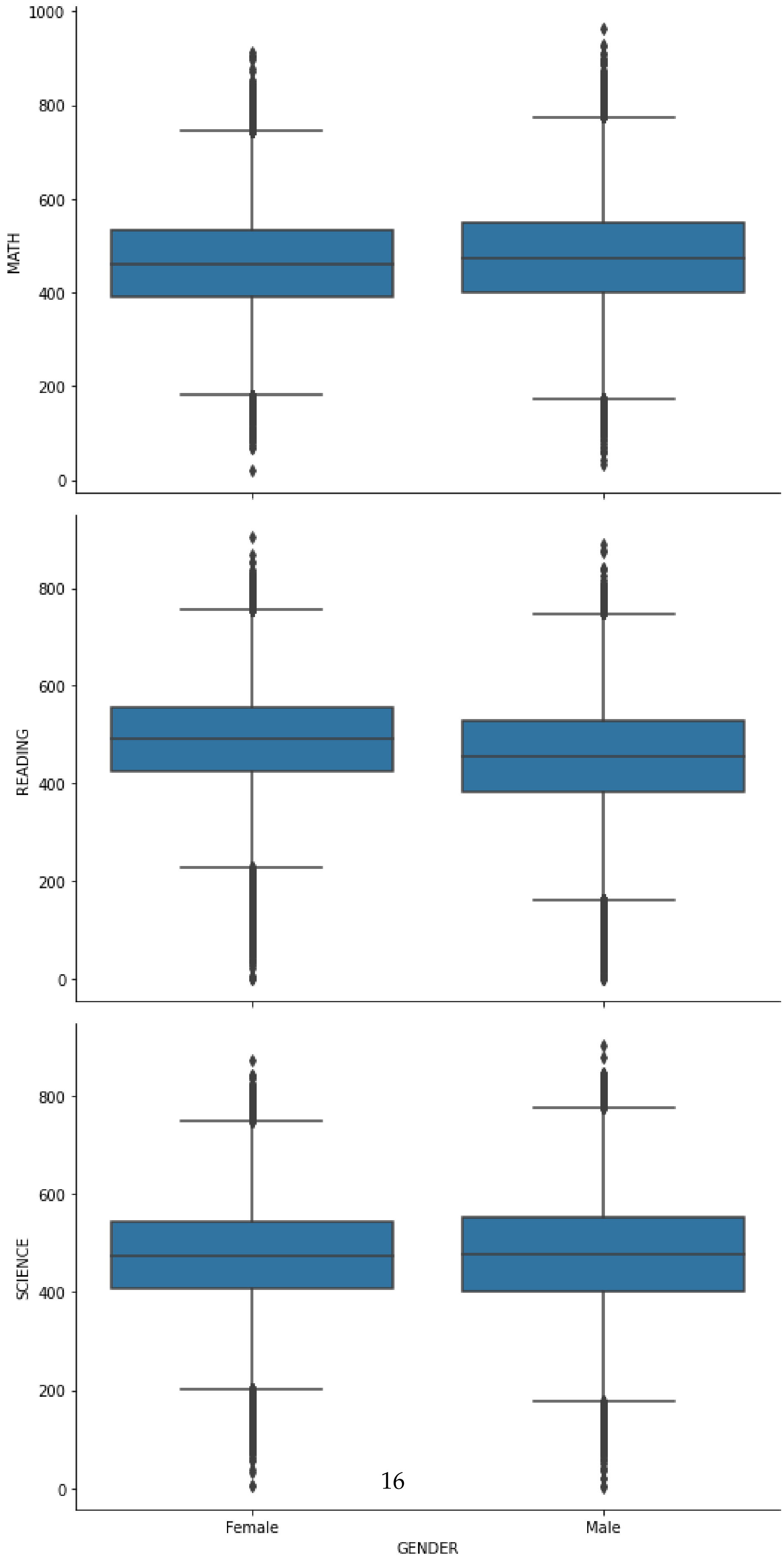
1.5 Bivariate Exploration

1.5.1 Question #1

Does gender change the test score distribution?

```
In [133]: #viz for gender vs test scores
          def boxgrid(x, y, **kwargs):
              default_color = sb.color_palette()[0]
              sb.boxplot(x=x, y=y, color=default_color)

          plt.figure(figsize = [10, 10])
          g = sb.PairGrid(data = df2, y_vars = ['MATH','READING','SCIENCE'], x_vars = ['GENDER']
                          height = 5, aspect = 1.5)
          g.map(boxgrid)
          plt.show();

          print(df2.MATH.groupby(df2['GENDER']).mean())
          print(df2.READING.groupby(df2['GENDER']).mean())
          print(df2.SCIENCE.groupby(df2['GENDER']).mean())
```

```
<Figure size 720x720 with 0 Axes>
```



1000
800
600
400
200
0
MATH
READING
SCIENCE
Female
Male
GENDER

```
GENDER
Female    464.033534
Male      475.317572
Name: MATH, dtype: float64
GENDER
Female    489.701508
Male      453.966386
Name: READING, dtype: float64
GENDER
Female    475.332517
Male      476.215567
Name: SCIENCE, dtype: float64
```

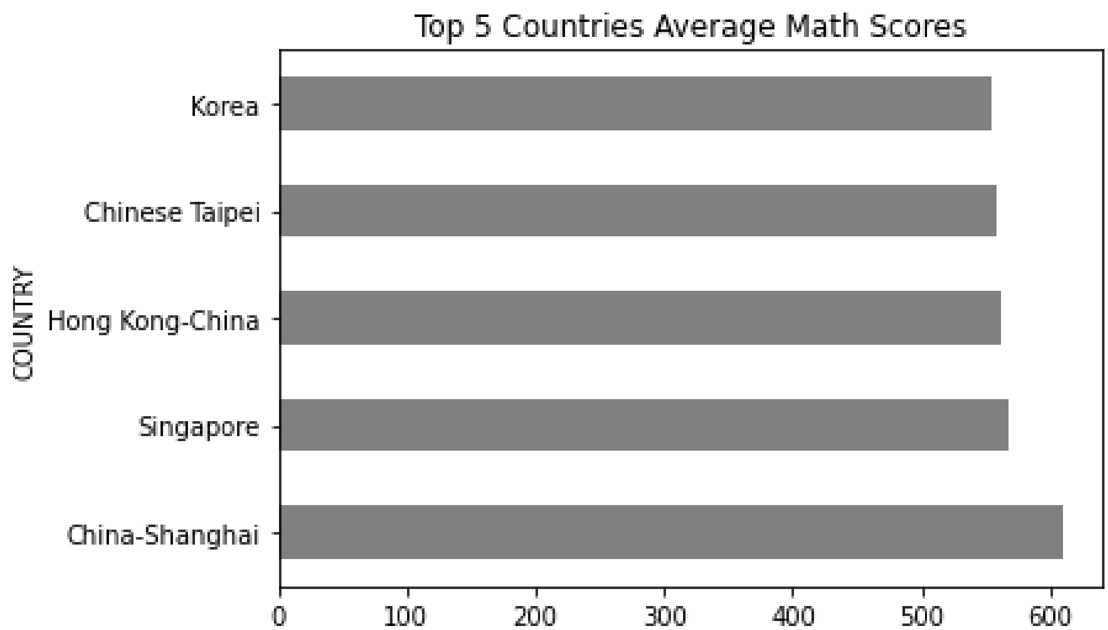
1.5.2 Observations

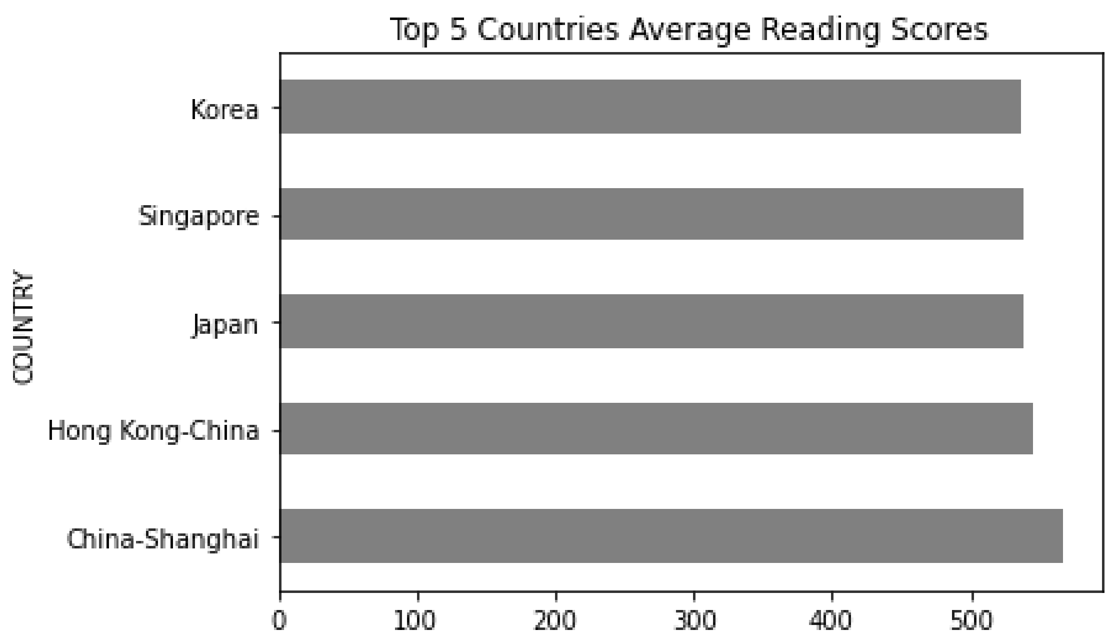
There seems to be very little difference in test scores between genders in science. In reading it is clear females perform better and males performed slightly better in math.

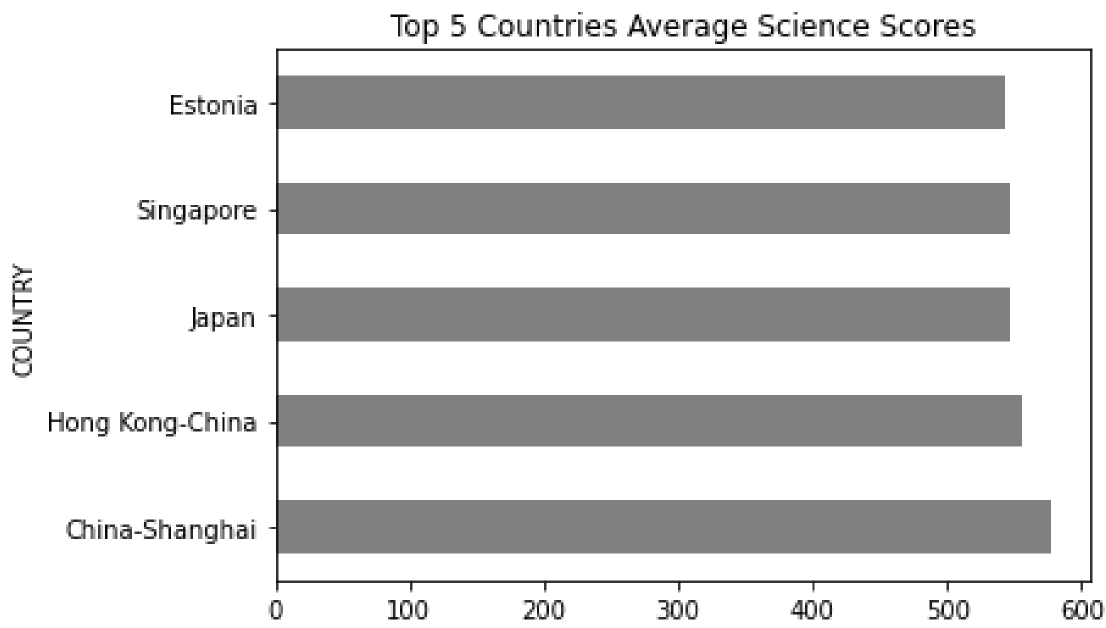
1.5.3 Question #2

What countries have the best and worst test scores?

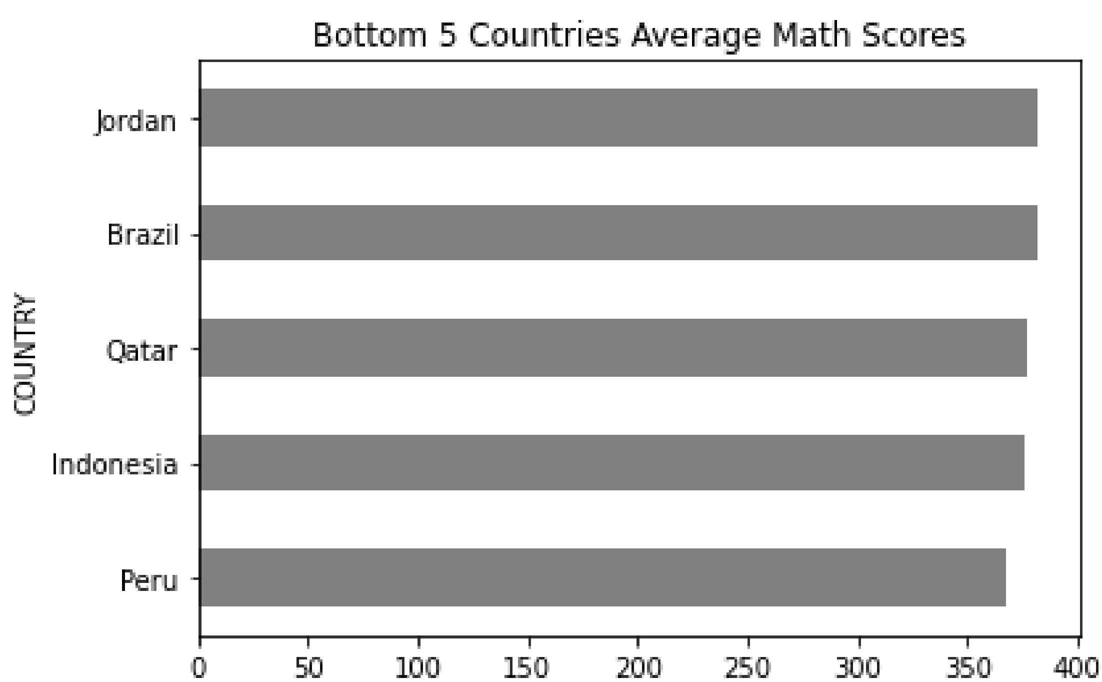
```
In [37]: #viz for largest average test scores
         df2.MATH.groupby(df2['COUNTRY']).mean().nlargest(5).plot(kind='barh', color= 'grey').se
         plt.show()
         df2.READING.groupby(df2['COUNTRY']).mean().nlargest(5).plot(kind='barh', color= 'grey')
         plt.show()
         df2.SCIENCE.groupby(df2['COUNTRY']).mean().nlargest(5).plot(kind='barh', color= 'grey')
         plt.show()
```

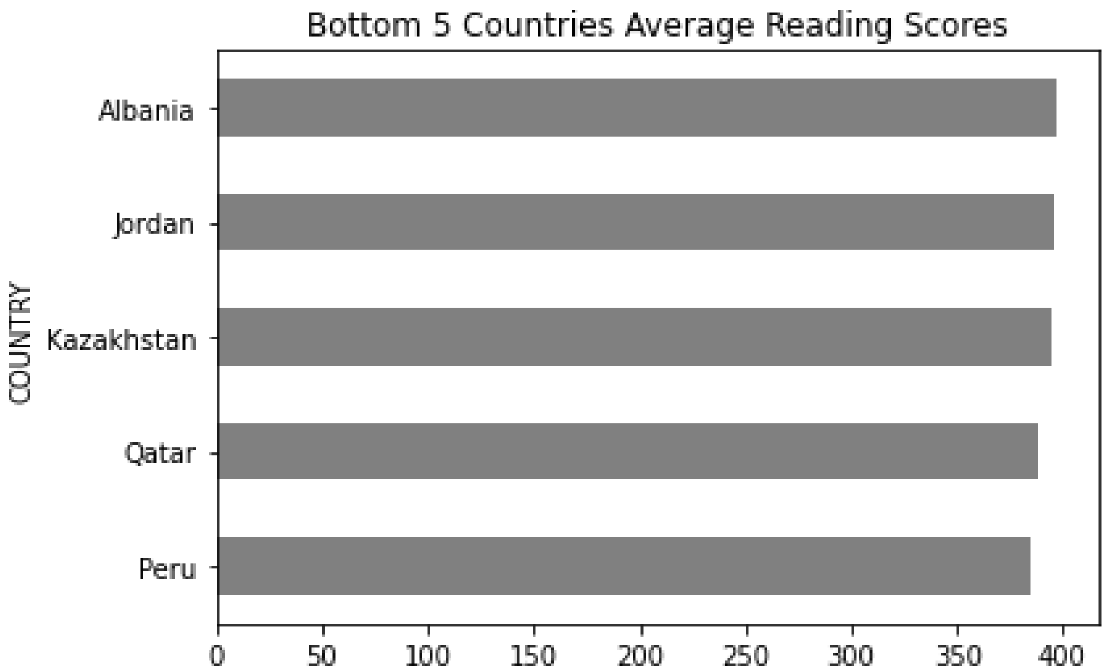

Top 5 Countries Average Math Scores
COUNTRY
Korea
Chinese Taipei
Hong Kong-China
Singapore
China-Shanghai
0
100
200
300
400
500
600

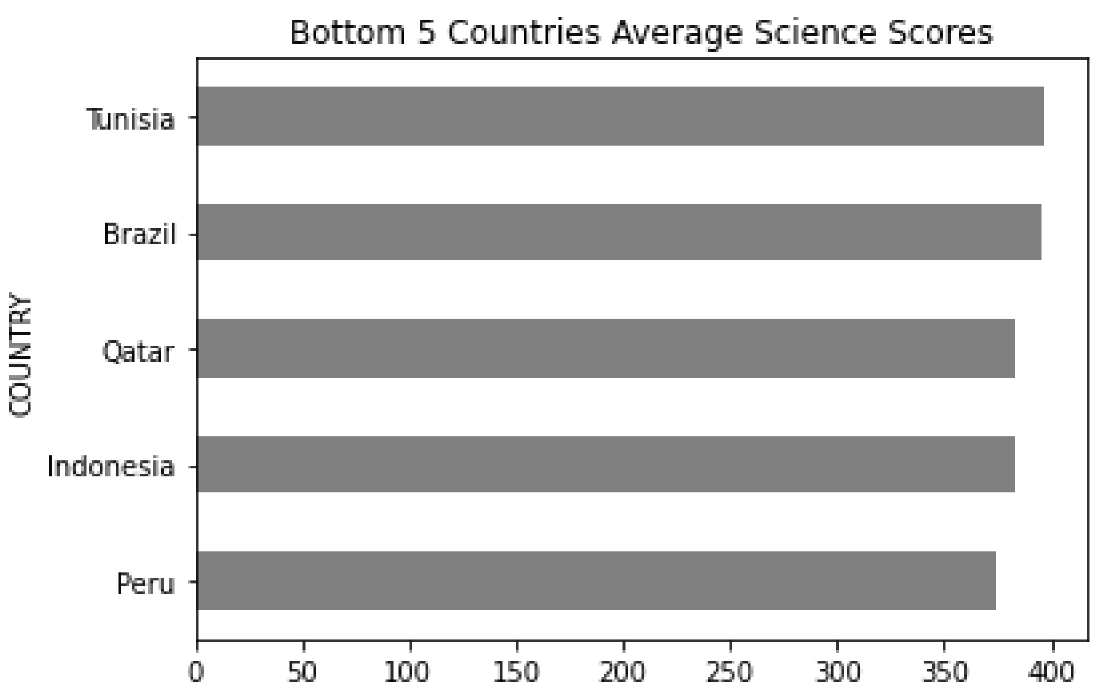

Top 5 Countries Average Reading Scores
COUNTRY
Korea
Singapore
Japan
Hong Kong-China
China-Shanghai
0
100
200
300
400
500




```
In [41]: #viz for lowest average test scores
         df2.MATH.groupby(df2['COUNTRY']).mean().nsmallest(5).plot(kind='barh', color= 'grey').s
         plt.show()
         df2.READING.groupby(df2['COUNTRY']).mean().nsmallest(5).plot(kind='barh', color= 'grey'
         plt.show()
         df2.SCIENCE.groupby(df2['COUNTRY']).mean().nsmallest(5).plot(kind='barh', color= 'grey'
         plt.show()
```




Bottom 5 Countries Average Reading Scores
COUNTRY
Albania
Jordan
Kazakhstan
Qatar
Peru
0
50
100
150
200
250
300
350
400


Bottom 5 Countries Average Science Scores
COUNTRY
Tunisia
Brazil
Qatar
Indonesia
Peru
0
50
100
150
200
250
300
350
400

1.5.4 Observations

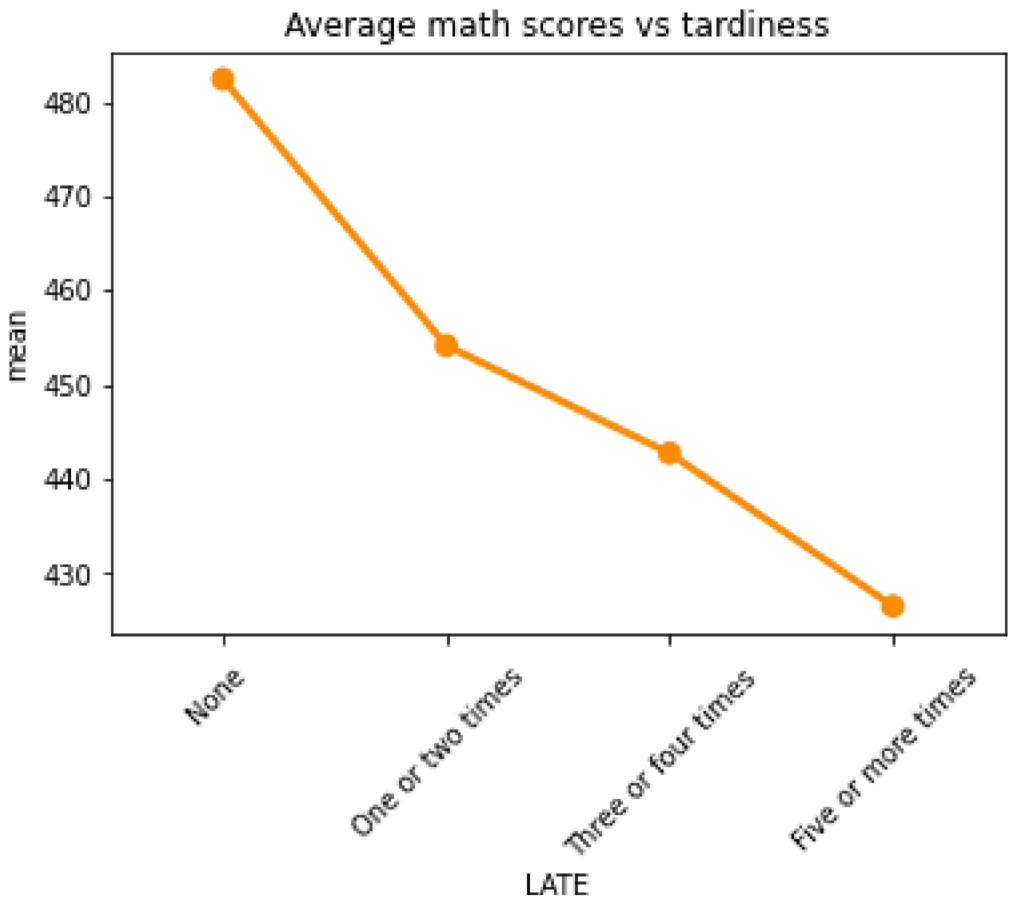
China-shanghai students scored the highest overall in all three exams; math, reading and science. China-hong-kong, Japan and Singapore were also in the top 5 for all three sections.

Peru students scored the lowest overall in all three exams. Qatar also fell in the bottom 5 in all three exams.

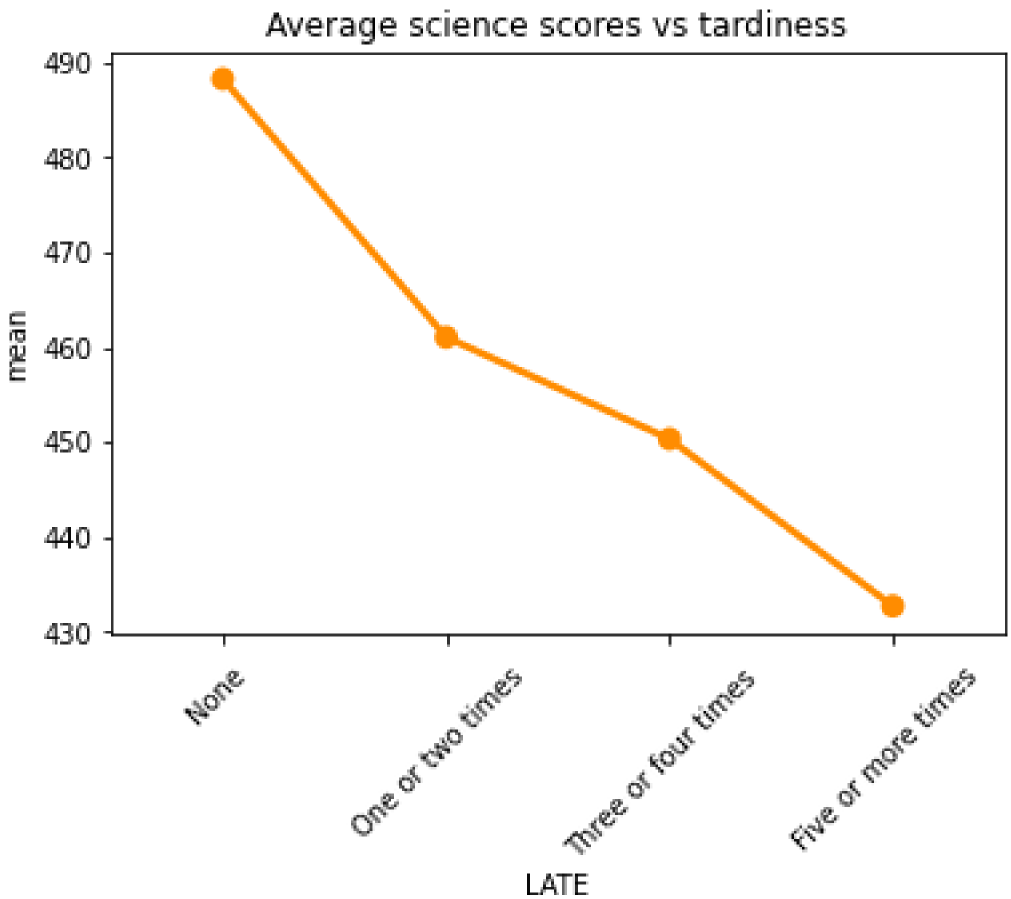
1.5.5 Question #3

Do test scores change if the student is tardy or skips classes?

```
In [63]: #viz for tardiness
         a=df2.MATH.groupby(df2.LATE).mean().reset_index(name='mean').sort_values(by='mean',asce
         b=df2.READING.groupby(df2.LATE).mean().reset_index(name='mean').sort_values(by='mean',a
         c=df2.SCIENCE.groupby(df2.LATE).mean().reset_index(name='mean').sort_values(by='mean',a

         sns.pointplot(x='LATE',y='mean',data=a,color='darkorange').set(title="Average math scor
         plt.xticks(rotation=45)
         plt.show()
         sns.pointplot(x='LATE',y='mean',data=b,color='darkorange').set(title="Average reading s
         plt.xticks(rotation=45)
         plt.show()
         sns.pointplot(x='LATE',y='mean',data=c,color='darkorange').set(title="Average science s
         plt.xticks(rotation=45)
         plt.show()
```


Average math scores vs tardiness
mean
480
470
460
450
440
430
None
One or two times
Three or four times
Five or more times
LATE


Average reading scores vs tardiness
480
470
460
450
440
430
420
mean
None
One or two times
Three or four times
Five or more times
LATE



In [71]:
```
#viz for skipping
d=df2.MATH.groupby(df2.SKIP).mean().reset_index(name='mean').sort_values(by='mean',asce
e=df2.READING.groupby(df2.SKIP).mean().reset_index(name='mean').sort_values(by='mean',a
f=df2.SCIENCE.groupby(df2.SKIP).mean().reset_index(name='mean').sort_values(by='mean',a


fig, ax = plt.subplots(figsize=(18,7))
c=sns.pointplot(data = d, x='SKIP', y='mean', color="b",
                label='math')
d=sns.pointplot(data = e, x='SKIP', y='mean', color="r",
                label='reading')
r=sns.pointplot(data = f, x='SKIP', y='mean', color="g",
                label='science')
ax.set_title('Average test scores vs skipping class', fontsize=22, y=1.015)
ax.set_xlabel('skipping', labelpad=16)
ax.set_ylabel('mean test score', labelpad=16)
ax.legend()

t=plt.xticks(rotation=45)
```

```
No handles with labels found to put in legend.
```

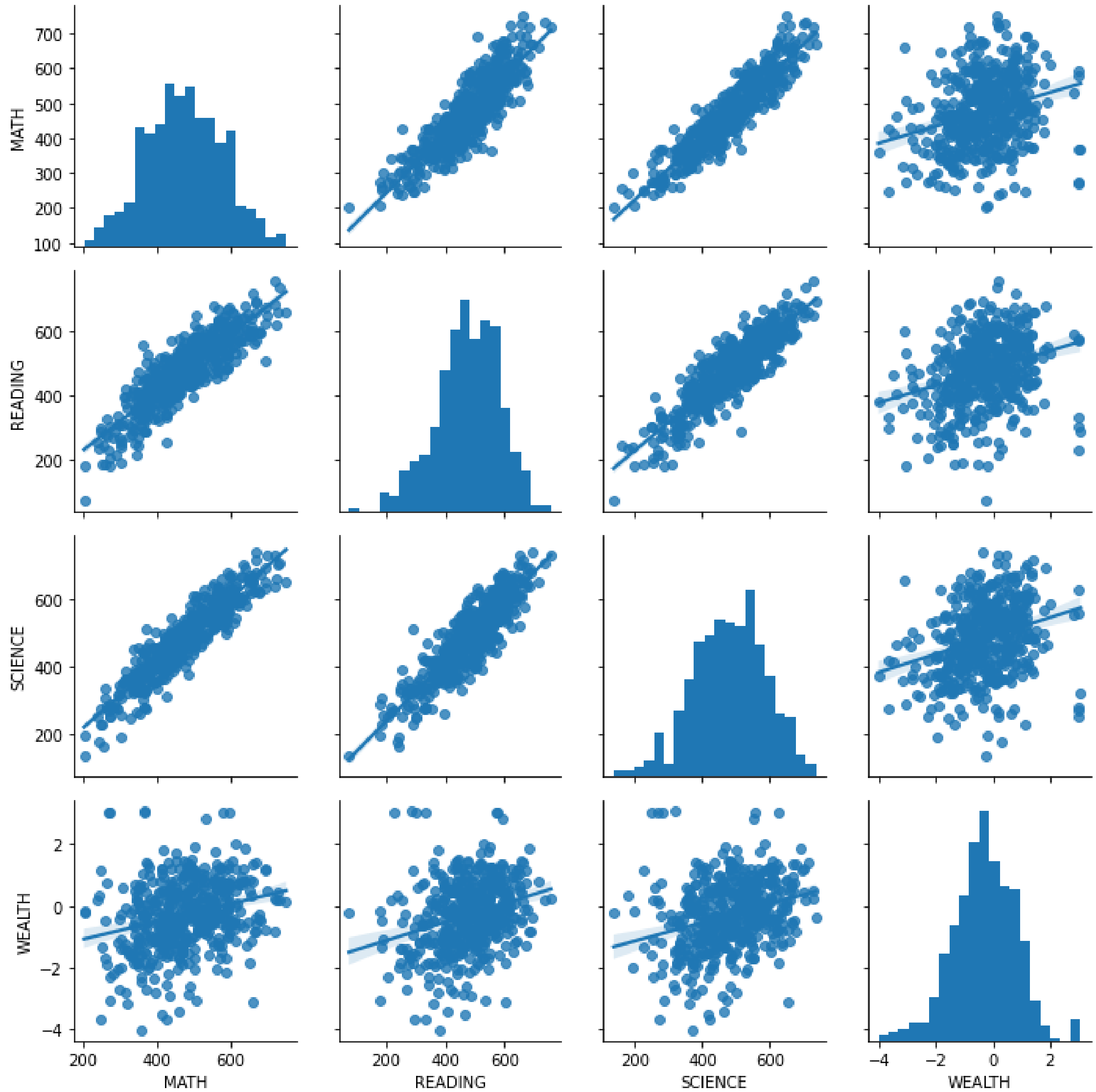
1.5.6 Observations

There is a very pronouced negative correlation between being tardy/late or skipping classes and test score dropping. The more frequently the student skips or is late the lower their test scores in all three exams.

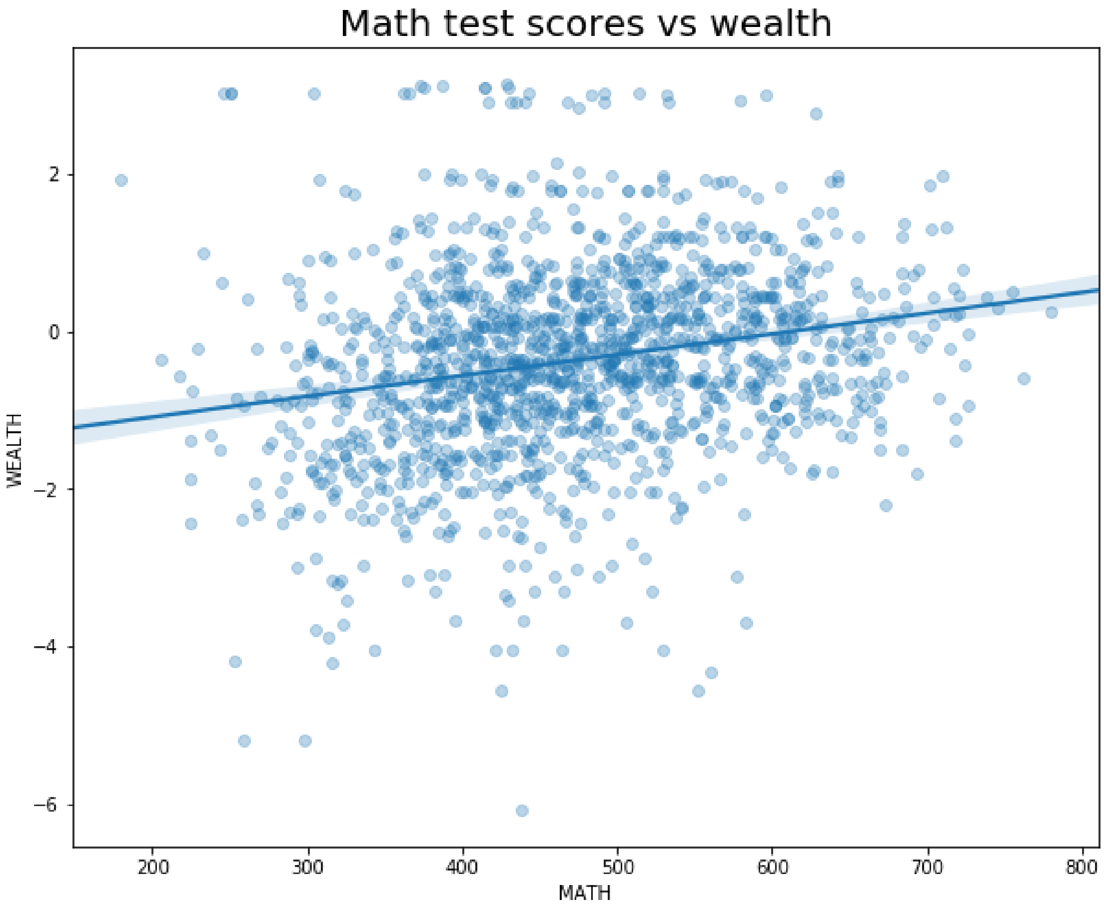
1.5.7 Question #4

Does wealth affect test scores?

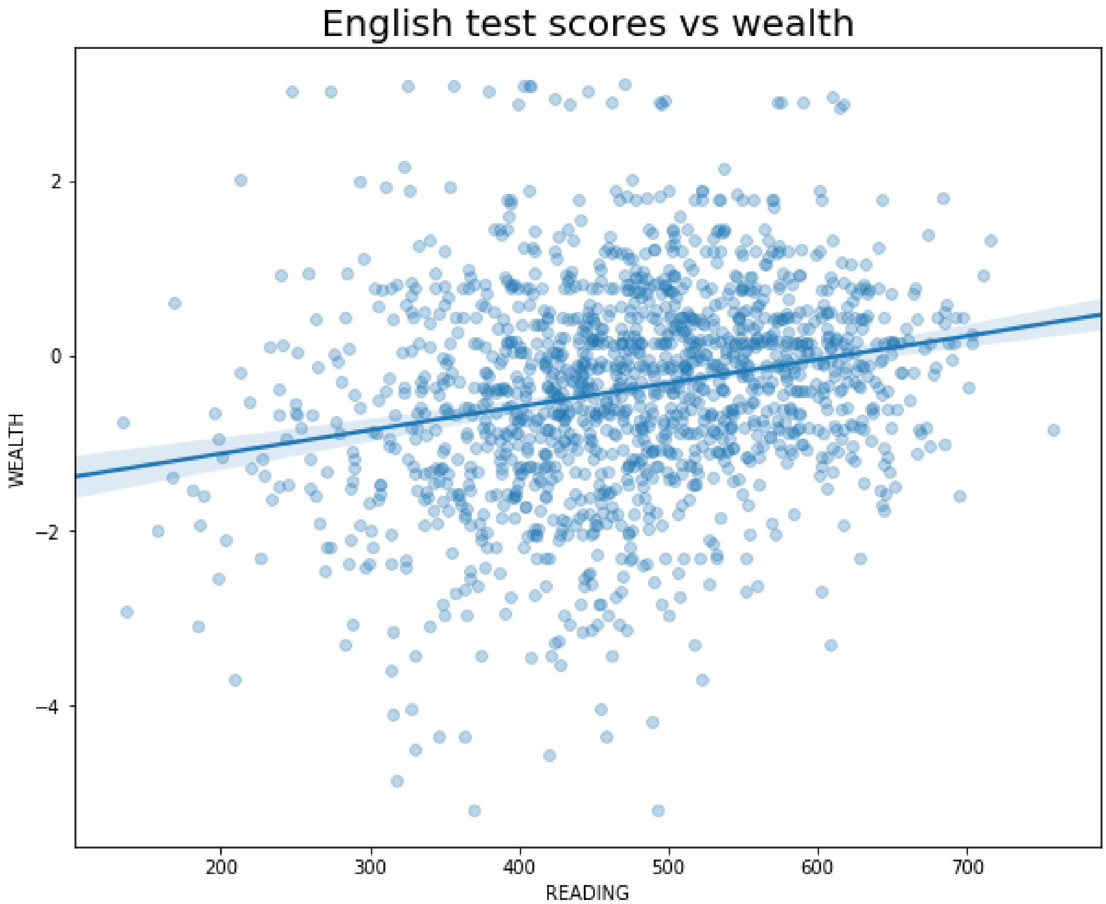
```
In [10]: sample = df2.sample(n=500, replace = False)
         numeric_var=df2[['MATH','READING','SCIENCE', 'WEALTH']]
         g = sns.PairGrid(data = sample, vars = numeric_var)
         g = g.map_diag(plt.hist, bins = 20);
         g.map_offdiag(sns.regplot)

Out[10]: <seaborn.axisgrid.PairGrid at 0x7f94a10ec240>
```



In [13]:
```
#viz math test scores vs wealth
g=df2.sample(n=1500)
plt.figure(figsize = (10,8))
sns.regplot(g.MATH, g.WEALTH, scatter_kws={'alpha':0.3})
plt.title('Math test scores vs wealth', fontsize = 20)
plt.show()
```




```
In [15]: #viz reading test scores vs wealth
         g=df2.sample(n=1500)
         plt.figure(figsize = (10,8))
         sns.regplot(g.READING, g.WEALTH, scatter_kws={'alpha':0.3})
         plt.title('English test scores vs wealth', fontsize = 20)
         plt.show()
```




```
In [16]: #viz science test scores vs wealth
         g=df2.sample(n=1500)
         plt.figure(figsize = (10,8))
         sns.regplot(g.SCIENCE, g.WEALTH, scatter_kws={'alpha':0.3})
         plt.title('Science test scores vs wealth', fontsize = 20)
         plt.show()
```

1.5.8 Observations

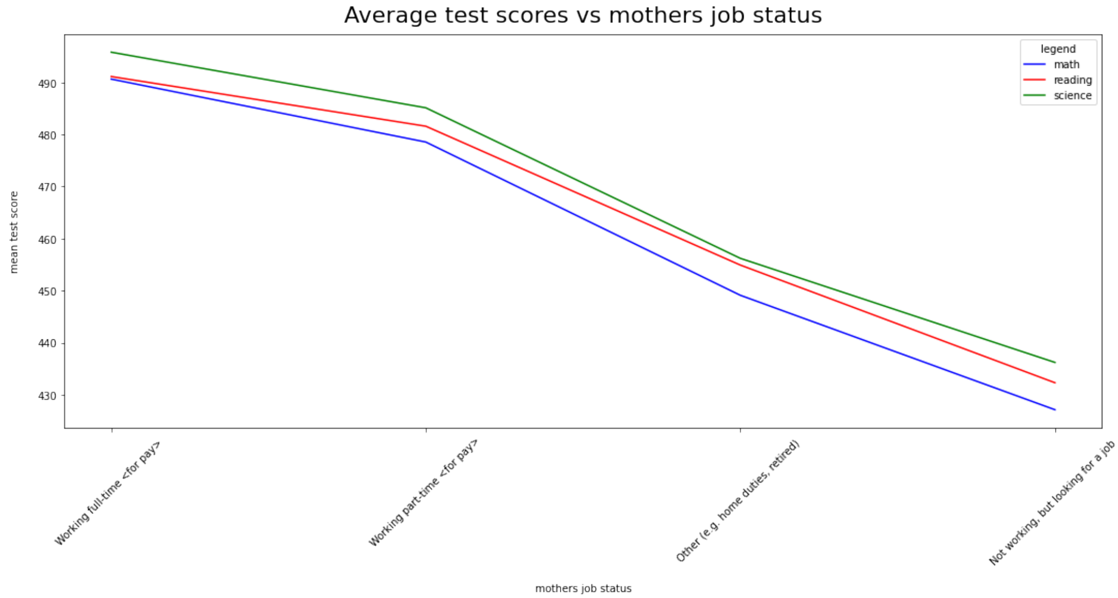
According to the three visulizations above, there seems to be a positive correlation between wealth and test scores.

1.5.9 Question #5

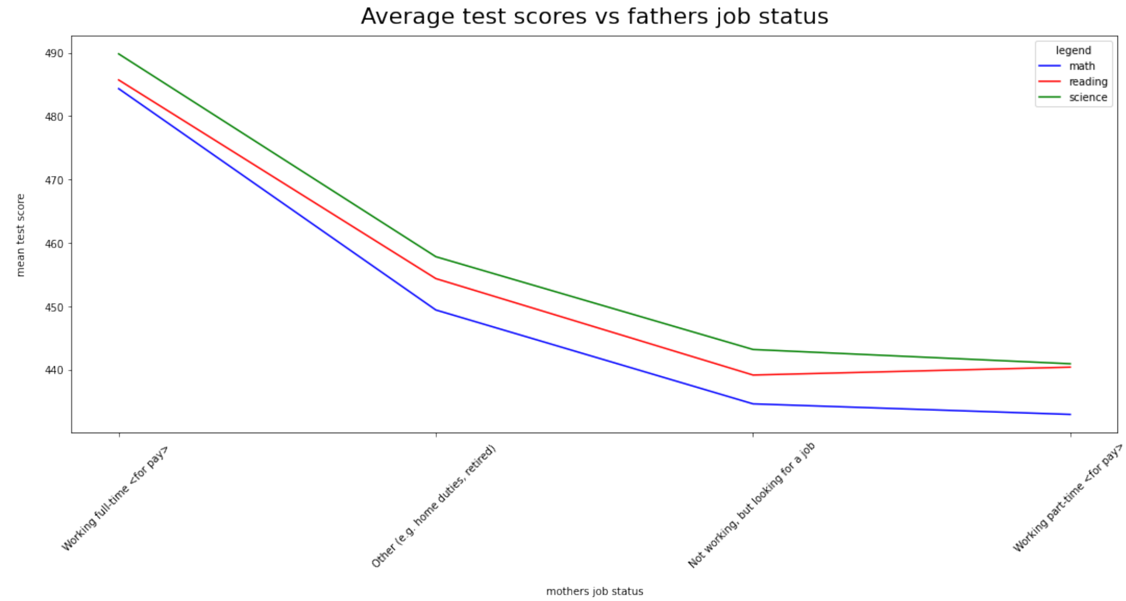
Is there a difference in test scores if the students mother works full time or ′other (home duties, retired)′?

```
In [75]: d=df2.MATH.groupby(df2.M_JOB).mean().reset_index(name='mean').sort_values(by='mean',asc
         e=df2.READING.groupby(df2.M_JOB).mean().reset_index(name='mean').sort_values(by='mean',
         f=df2.SCIENCE.groupby(df2.M_JOB).mean().reset_index(name='mean').sort_values(by='mean',


         fig, ax = plt.subplots(figsize=(18,7))
         c=sns.lineplot(data = d, x='M_JOB', y='mean', color="b",
                         label='math')
         d=sns.lineplot(data = e, x='M_JOB', y='mean', color="r",
```

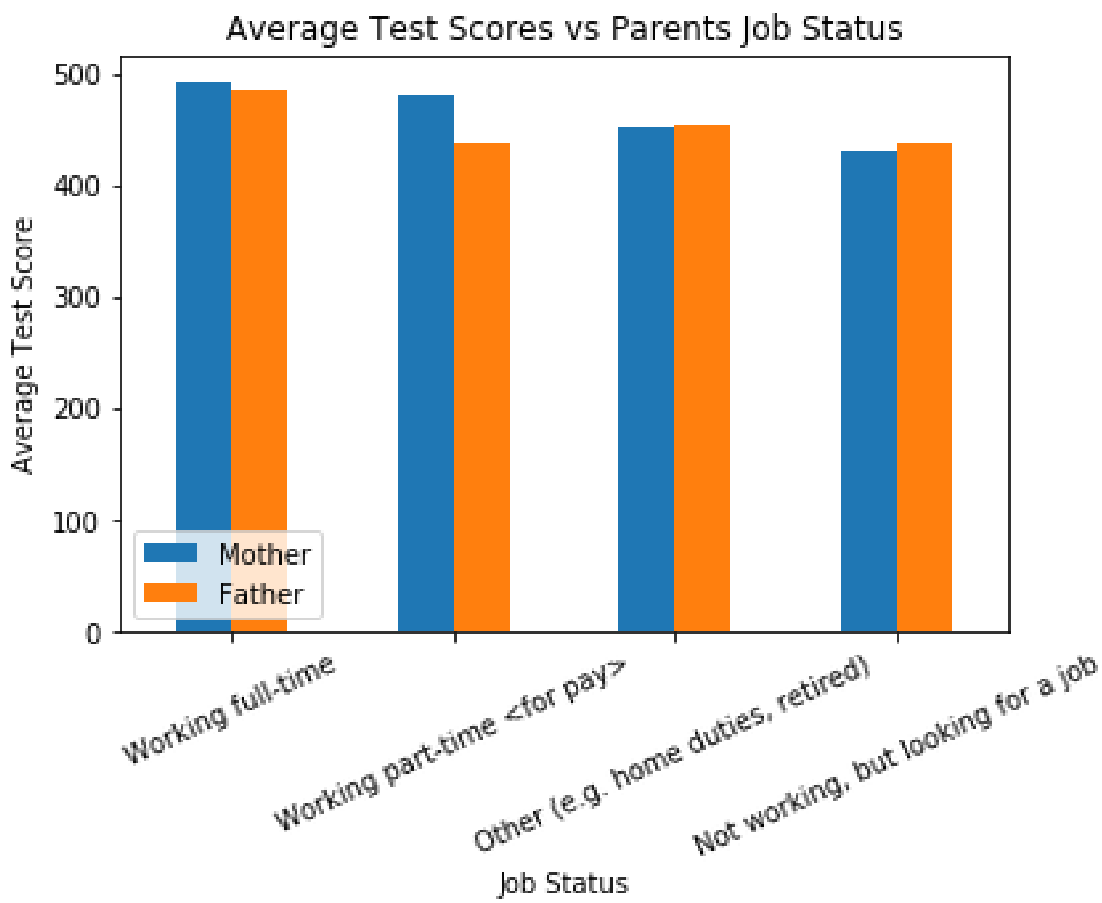
```
                label='reading')
r=sns.lineplot(data = f, x='M_JOB', y='mean', color="g",
                label='science')
ax.set_title('Average test scores vs mothers job status', fontsize=22, y=1.015)
ax.set_xlabel('mothers job status', labelpad=16)
ax.set_ylabel('mean test score', labelpad=16)
ax.legend(title='legend')

t=plt.xticks(rotation=45)
```




```
In [5]: d=df2.MATH.groupby(df2.F_JOB).mean().reset_index(name='mean').sort_values(by='mean',asce
        e=df2.READING.groupby(df2.F_JOB).mean().reset_index(name='mean').sort_values(by='mean',a
        f=df2.SCIENCE.groupby(df2.F_JOB).mean().reset_index(name='mean').sort_values(by='mean',a


        fig, ax = plt.subplots(figsize=(18,7))
        c=sns.lineplot(data = d, x='F_JOB', y='mean', color="b",
                        label='math')
        d=sns.lineplot(data = e, x='F_JOB', y='mean', color="r",
                        label='reading')
        r=sns.lineplot(data = f, x='F_JOB', y='mean', color="g",
                        label='science')
        ax.set_title('Average test scores vs fathers job status', fontsize=22, y=1.015)
        ax.set_xlabel('mothers job status', labelpad=16)
        ax.set_ylabel('mean test score', labelpad=16)
        ax.legend(title='legend')

        t=plt.xticks(rotation=45)
```




```
In [4]: jobs=["Working full-time", "Working part-time <for pay>", "Other (e.g. home duties, reti
        parent={
            "Mother":[492.037652,481.261803,452.905892,431.369983],
            "Father":[486.128335,437.629457,453.400981,438.522076],

        }

        df=pd.DataFrame(parent,index=jobs)

        df.plot(kind="bar",stacked=False,figsize=(6,4))
        plt.legend(loc="lower left")
        plt.xticks(rotation=25)
        plt.title('Average Test Scores vs Parents Job Status')
        plt.xlabel('Job Status')
        plt.ylabel('Average Test Score')
        plt.show()
```



1.5.10 Observations

Students have the highest test score if the mother is working full time. There is a slight drop when they work part time. There is a more pronouced test score drop if the mother is not working or other. There is a large drop in score if the father is not working full time.

1.5.11 Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

I was somewhat suprised gender did not correlate with more of a difference in test scores. There was a larger difference in reading scores but still not what I had expected.It was interesting how dominate chinese students were in all test scores and the top 5 countries were almost the same in each exam. There was more variability in the bottom 5 countries. As expected test scores dropped when students were tardy or skipped classes with more of a drop when skipping classes. There is also evidence that the more wealth a student has the better their test scores may be.

1.5.12 Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

I thought is was very interesting that students test score dropped dramatically if their mother was not working full or part time. I did not expect there to be such a strong correlation.

1.6 Multivariate Exploration

1.6.1 Question #1

Do wealth and gender have an effect on test scores?

```
In [13]: #create a single test score mean column
         #convert data type of test scores
         df2[['MATH','READING','SCIENCE']] = df2[['MATH','READING','SCIENCE']].astype(int)
         df2['mean']=df2['MATH']+df2['READING']+df2['SCIENCE']
         df2['mean']=(df2['mean']/3)
```

```
In [14]: #create bins(low,medium,high) for wealth to easier interpret wealth levels
         df2['wealth_bins']=pd.qcut(x=df2['WEALTH'],q=[0,.33,.67,1],labels=['low','medium','high
```

```
In [8]: #Average test scores vs wealth and gender
        d=df2.groupby(['wealth_bins','GENDER'])['mean'].mean().reset_index(name='mean')

        fig, ax = plt.subplots(figsize=(10,5))
        c=sns.barplot(data = d, x='wealth_bins', y='mean', hue= d.GENDER, palette="rocket")

        ax.set_title('Average test scores vs wealth and gender', fontsize=22, y=1.015)
        ax.set_xlabel('wealth status', labelpad=16)
        ax.set_ylabel('mean test score', labelpad=16)
        ax.legend(title='legend')


        t=plt.xticks(rotation=45)
```

1.6.2 Observation

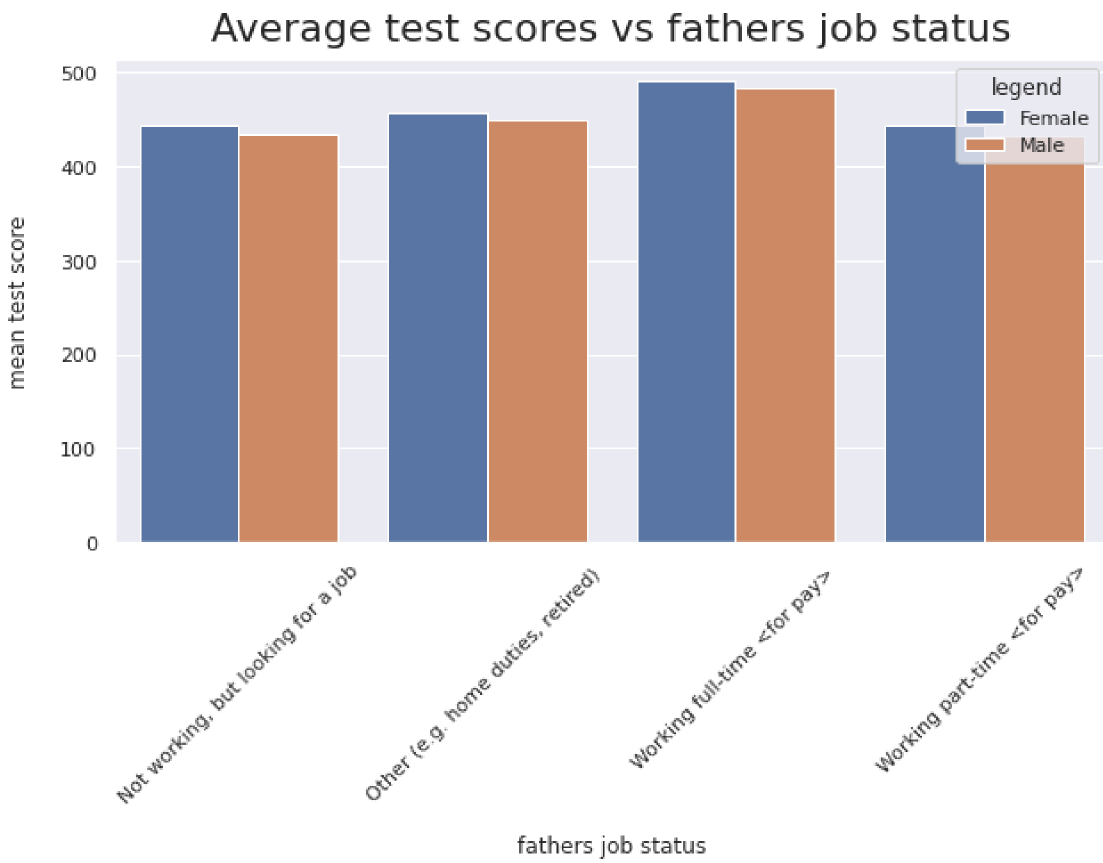
This viz shows the average test score in each wealth bin is higher for females than males in the same bin. As the wealth bins increase (low-high) the test scores for both genders also increase.

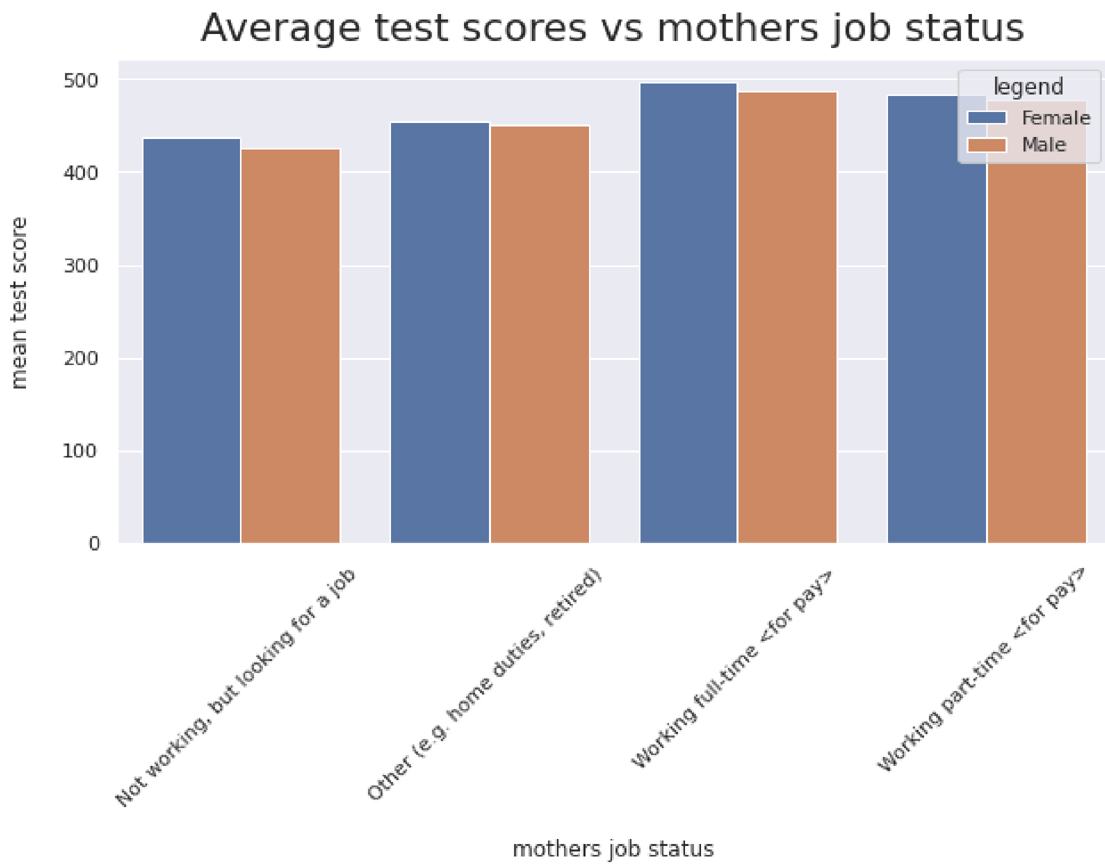
1.6.3 Question #2

Do parents job status AND students gender have a correlation to test scores?

In [111]:
```
#Average test scores vs fathers job status and gender
d=df2.groupby(['F_JOB','GENDER'])['mean'].mean().reset_index(name='means')

fig, ax = plt.subplots(figsize=(10,5))
c=sns.barplot(data = d, x='F_JOB', y='means', hue= d.GENDER)

ax.set_title('Average test scores vs fathers job status', fontsize=22, y=1.015)
ax.set_xlabel('fathers job status', labelpad=16)
ax.set_ylabel('mean test score', labelpad=16)
ax.legend(title='legend')

t=plt.xticks(rotation=45)

#Average test scores vs mothers job status and gender
```

```
e=df2.groupby(['M_JOB','GENDER'])['mean'].mean().reset_index(name='means')

fig, ax = plt.subplots(figsize=(10,5))
c=sns.barplot(data = e, x='M_JOB', y='means', hue= e.GENDER)

ax.set_title('Average test scores vs mothers job status and gender', fontsize=22, y=1.
ax.set_xlabel('mothers job status', labelpad=16)
ax.set_ylabel('mean test score', labelpad=16)
ax.legend(title='legend')

t=plt.xticks(rotation=45)
```





1.6.4 Observation

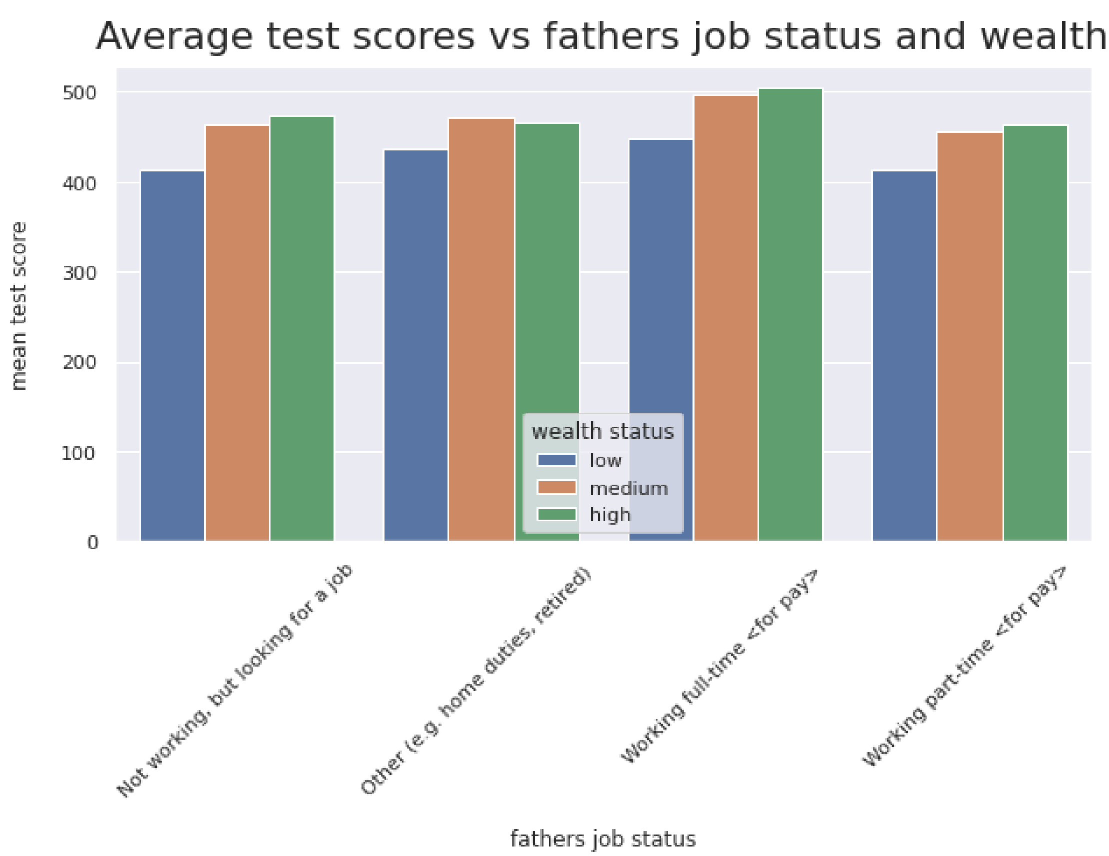
The highest test scores are correlated with both parents working full time. However, there is only very suble difference in test scores if the mother works part time compared to a more noticable test score drop if the father works part time or not at all. Gender test score differences do not seem to change much as parents job status changes.

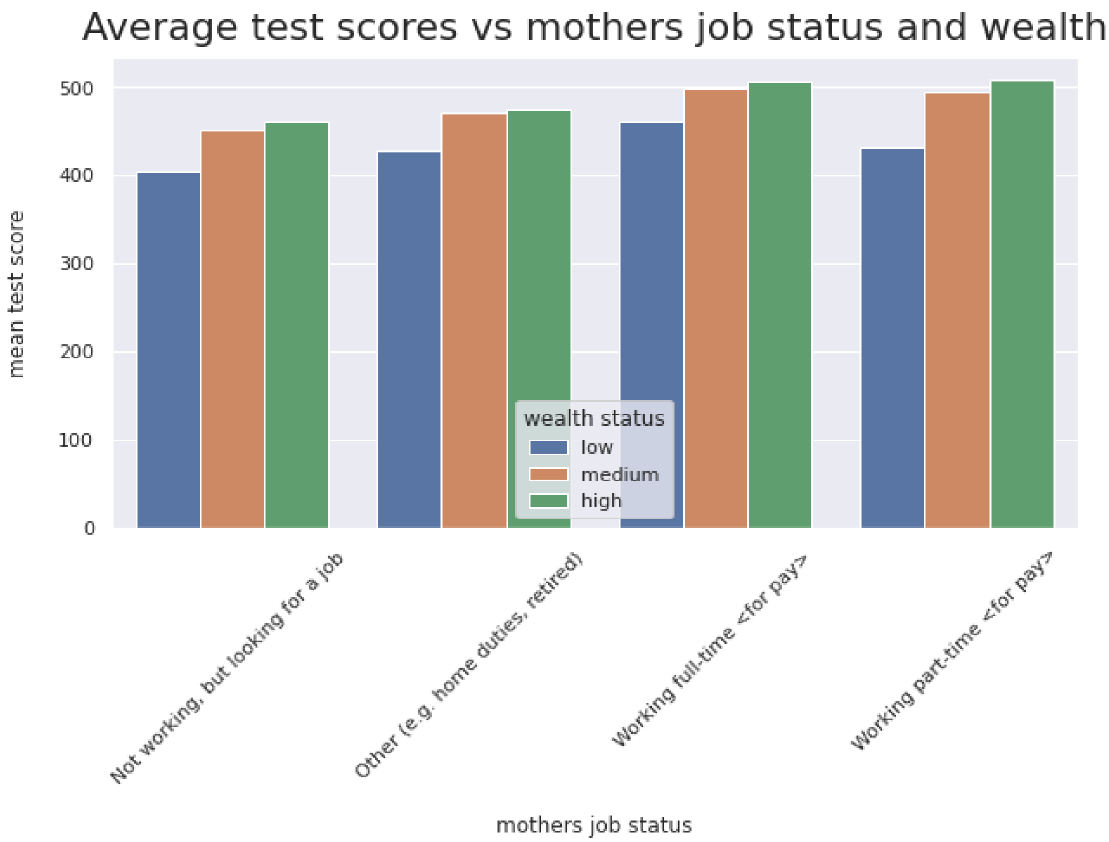
1.6.5 Question #3

Do parents job status AND wealth status correlate with students test scores?

```
In [112]: #test scores vs job status and wealth- father

          d=df2.groupby(['F_JOB','wealth_bins'])['mean'].mean().reset_index(name='mean')

          fig, ax = plt.subplots(figsize=(10,5))
          c=sns.barplot(data = d, x='F_JOB', y='mean', hue= d.wealth_bins)

          ax.set_title('Average test scores vs fathers job status and wealth', fontsize=22, y=1.
          ax.set_xlabel('fathers job status', labelpad=16)
```

```
ax.set_ylabel('mean test score', labelpad=16)
ax.legend(title='wealth status')

t=plt.xticks(rotation=45)

#test scores vs job status and wealth- mother

e=df2.groupby(['M_JOB','wealth_bins'])['mean'].mean().reset_index(name='mean')

fig, ax = plt.subplots(figsize=(10,5))
c=sns.barplot(data = e, x='M_JOB', y='mean', hue= e.wealth_bins)

ax.set_title('Average test scores vs mothers job status and wealth', fontsize=22, y=1.
ax.set_xlabel('mothers job status', labelpad=16)
ax.set_ylabel('mean test score', labelpad=16)
ax.legend(title='wealth status')

t=plt.xticks(rotation=45)
```





1.6.6 Observation

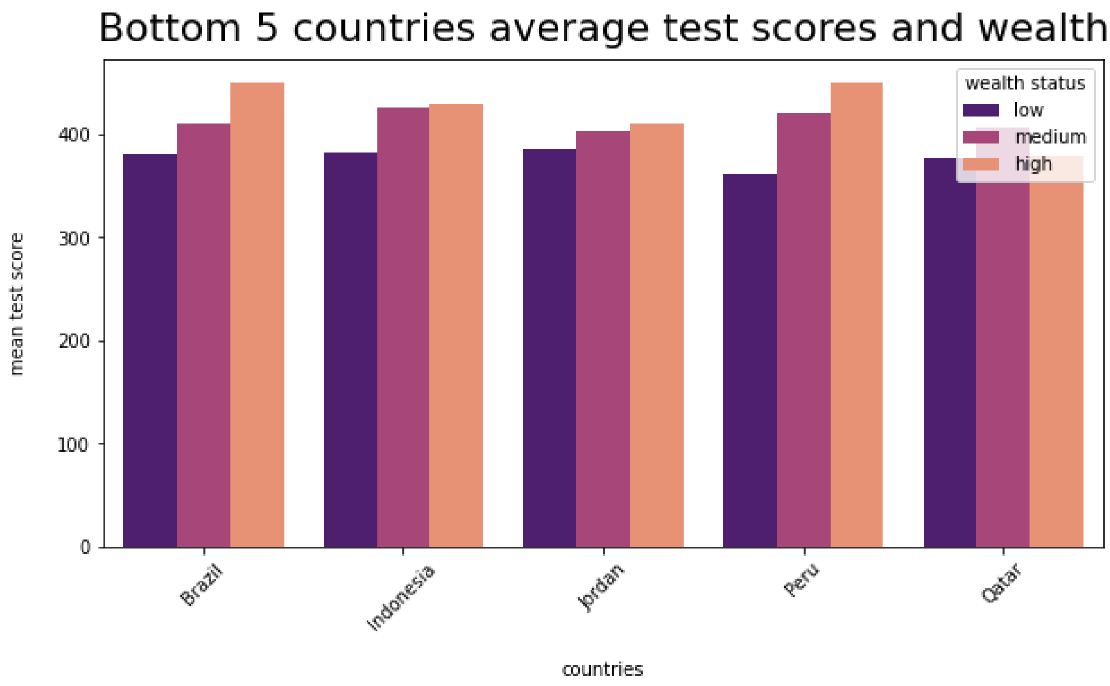
The highest average test scores correlate with high-wealth fathers who work full time and high-wealth mothers who work full or part time. The lowest average test scores in each job category belong to those in the low-wealth status. Low-wealth fathers working part-time or not working and low-wealth mothers who are not working have the lowest average test scores.

1.6.7 Question #4

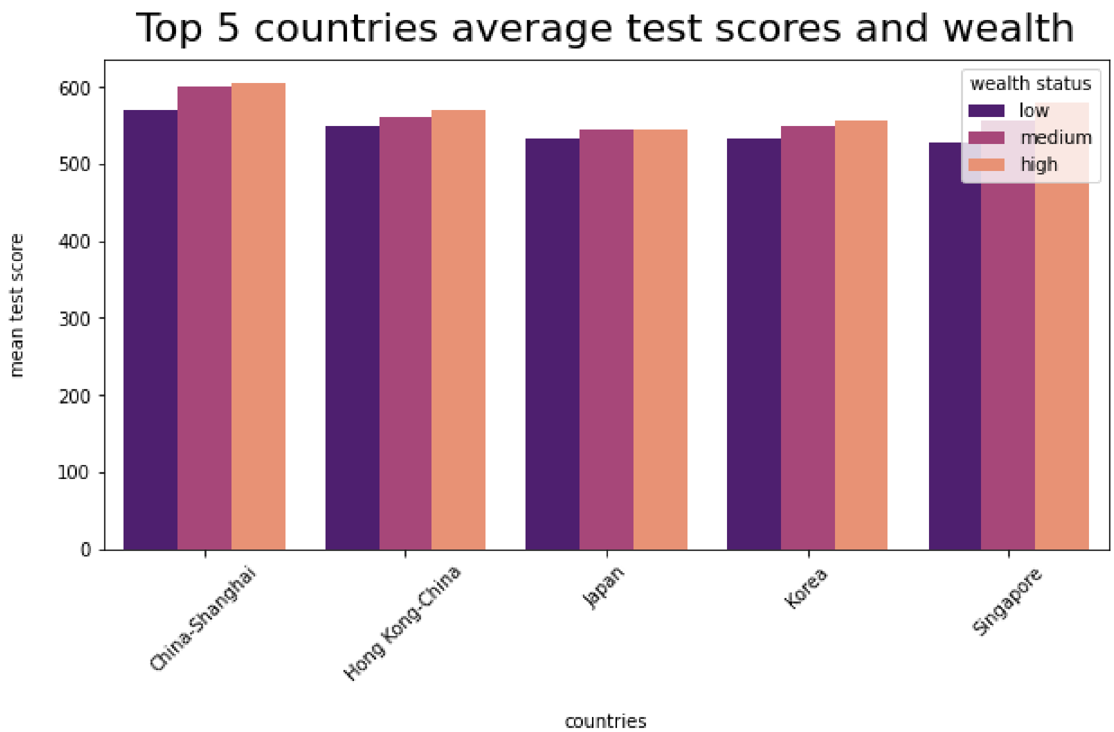
Is there correlation between the students country, wealth, and test scores?

In [51]:
```
#Bottom 5 countries average test scores and wealth
k=df2['mean'].groupby(df2['COUNTRY']).mean().nsmallest(5)
m=df2.wealth_bins.groupby(df2['COUNTRY']).value_counts(normalize= True)


e=df2.groupby(['COUNTRY','wealth_bins'])['mean'].mean().reset_index(name='mean')


fig, ax = plt.subplots(figsize=(10,5))
```

```
c=sns.barplot(data = e[e.COUNTRY.isin(k.index.tolist())], x='COUNTRY', y='mean', hue= e
ax.set_title('Bottom 5 countries average test scores and wealth', fontsize=22, y=1.015)
ax.set_xlabel('countries', labelpad=16)
ax.set_ylabel('mean test score', labelpad=16)
ax.legend(title='wealth status')

t=plt.xticks(rotation=45)
```



In [52]:
```
#Top 5 countries average test scores and wealth
k=df2['mean'].groupby(df2['COUNTRY']).mean().nlargest(5)

e=df2.groupby(['COUNTRY','wealth_bins'])['mean'].mean().reset_index(name='mean')


fig, ax = plt.subplots(figsize=(10,5))
c=sns.barplot(data = e[e.COUNTRY.isin(k.index.tolist())], x='COUNTRY', y='mean', hue= e
ax.set_title('Top 5 countries average test scores and wealth', fontsize=22, y=1.015)
ax.set_xlabel('countries', labelpad=16)
ax.set_ylabel('mean test score', labelpad=16)
ax.legend(title='wealth status')

t=plt.xticks(rotation=45)
```




```
In [45]: #Highest wealth countries
         df2[df2['wealth_bins']== "high"].groupby(['COUNTRY']).agg({'WEALTH':'mean'}).reset_inde
```

Out[45]:

	COUNTRY	WEALTH
47	Qatar	1.624669
60	United Arab Emirates	1.500011
7	Canada	1.146040
62	United States of America	1.094543
8	Chile	1.083732

```
In [48]: #Lowest wealth counties
         df2[df2['wealth_bins']== "high"].groupby(['COUNTRY']).agg({'WEALTH':'mean'}).reset_inde
```

Out[48]:

	COUNTRY	WEALTH
15	Denmark	0.617632
13	Croatia	0.600048
64	Vietnam	0.590355
9	China-Shanghai	0.562357
31	Korea	0.425931

1.6.8 Observation

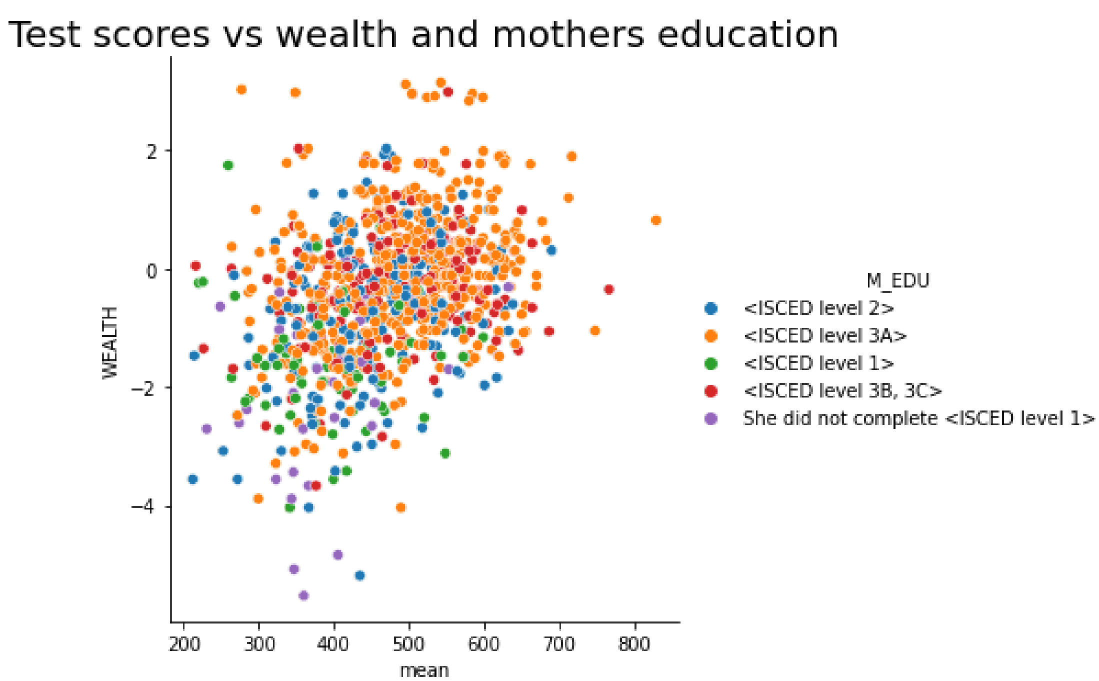
The visulizations for top and bottom countries for average test scores and wealth surprised me. Based on what I see, my takeaway is that wealth less of an impact in the countries with the highest average scores. Wealth has more of an impact on test scores in the bottom 5 countries. I quickly looked at the highest wealth countries and lowest wealth: in our dataset, Qatar is the wealthiest country but bottom 5 in average test scores. China-Shanghai has the highest average wealth but is the 2nd poorest country.

1.6.9 Question #5

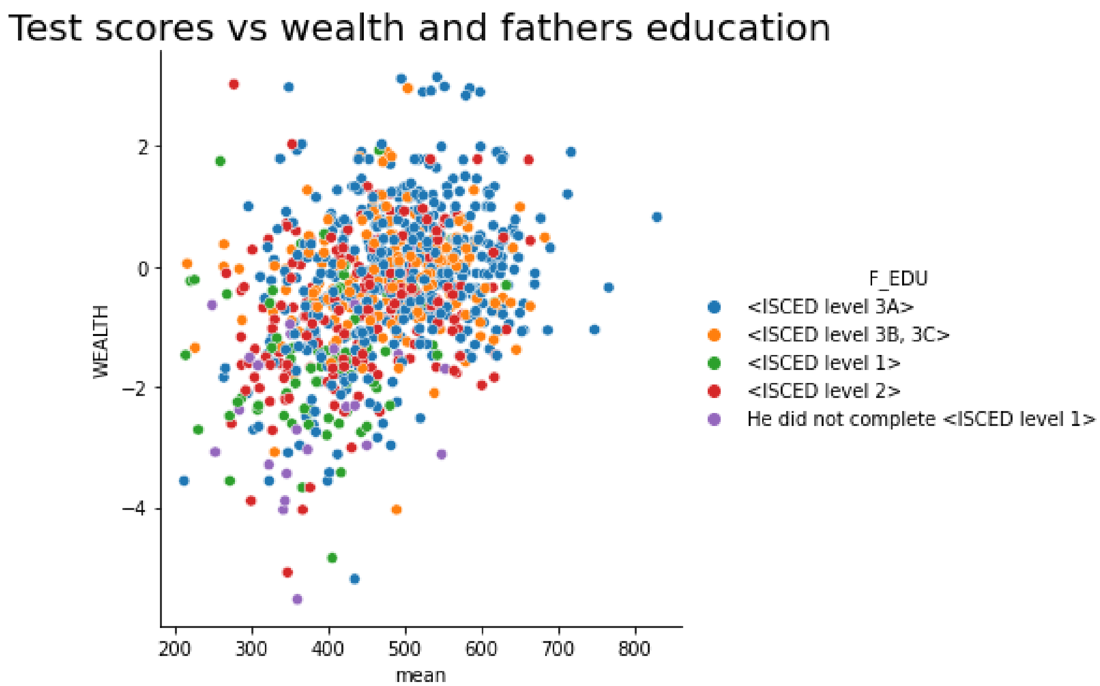
I never did take a close look at the parents education level compared to test scores. Do parents education level and wealth status have any correlation to students test scores?

```
In [49]: #Test scores vs wealth and mothers education
         g=df2.sample(n=1000)
         c=sns.relplot(g['mean'], g.WEALTH, hue= g.M_EDU)
         plt.title('Test scores vs wealth and mothers education', fontsize = 20)
         plt.show()

         #Test scores vs wealth and fathers education
         c=sns.relplot(g['mean'], g.WEALTH, hue= g.F_EDU)
         plt.title('Test scores vs wealth and fathers education', fontsize = 20)
         plt.show()
```

```
/opt/conda/lib/python3.6/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the follow
  FutureWarning
```



```
/opt/conda/lib/python3.6/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the follow
  FutureWarning
```



1.6.10 Observation

There is a higher concentration of 'did not complete level 1' and 'level 1' in the low test score, low wealth section of this viz. This indicates students test scores suffer with the combination of low weath and less educated parents. It shows the opposite to be true as well; high wealth and highly educated parents have positive effects on test scores.

1.6.11 Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

In my investigation I found that greater wealth consistantly has a positive correlation with test scores. Low wealth clearly has a negative correlation on test scores. My last visulization showed that the higher the parents education level and wealth level correlated with higher test scores. Parents working full time and in the high wealth bin also had students with the highest scores. Female students had overall slightly higher scores than their male counterparts across wealth and parent job status.

1.6.12 Were there any interesting or surprising interactions between features?

I was surprised that there was a drop in test scores if the father worked anything but full time. To where the mother could work full time or part time with very little change in scores. I had also thought mothers without a job may provide more help to their students resulting in higher scores but non- working mothers had the lowest student test scores. Also,Qatar is the wealthiest country but bottom 5 in average test scores. China-Shanghai has the highest average wealth but is the 2nd poorest country.

```
In [5]: df2.to_csv('my_pisa1.csv', index= False)
```

```
In [ ]:
```