

Pisa Data Exploration

June 3, 2020

1 Pisa Data Exploration

1.1 by Samir Attyani

1.2 Preliminary Wrangling

PISA program overview

The Programme for International Student Assessment (PISA) is an international assessment of the skills and knowledge of 15-year-old students; in addition, it provides information about a range of factors that contribute to the success of students, schools, and education systems. PISA is a collaborative effort among member countries of the Organisation for Economic Co-operation and Development (OECD).

PISA covers three domains — reading, mathematics, and science. Although each assessment includes questions from all three domains, the focus shifts. In 2000, the emphasis was on reading, with mathematics and science as minor domains. In 2003, mathematics was the major domain, and in 2006, it was science. In 2009, the focus was again reading, and in 2012, mathematics. In the assessment of 2015, the focus was science once again. The repetition of the assessments at regular intervals yields timely data that can be compared internationally and over time.

As PISA is an international assessment, it measures skills that are generally recognized as key outcomes of the educational process. Rather than testing on facts, the assessment focuses on young people near the end of compulsory schooling and their ability to use their knowledge and skills to meet real-life challenges.

Participation

International participation in PISA has grown steadily — from 32 countries/economies in 2000 to 41 in 2003, 57 in 2006, 65 in 2009, and 65 in 2012. In the latest PISA cycle, in 2015, there were 72 countries/economies participating. Canada has participated in PISA since its inception, through a partnership among CMEC, Employment and Social Development Canada, and Statistics Canada.

All 10 provinces have participated in each assessment. Approximately 20,000 Canadian students from about 1,000 schools have taken part in each PISA assessment in either English or French. Schools and the students within schools are selected randomly for participation. This large sample size allows results to be reported for each province, as well as for both the French- and English-language school systems in Nova Scotia, New Brunswick, Quebec, Ontario, Manitoba, Alberta, and British Columbia. Currently, Yukon, Northwest Territories, and Nunavut do not participate in PISA, nor do Indigenous students from band-operated schools.

The results are valid only on the pan-Canadian and provincial levels. No results are attributed to individual schools or students. PISA does not assess individual student achievement.

The assessment

In addition to two hours of direct assessment of reading, mathematics, and science, students in Canada complete a background questionnaire about themselves and their homes, about information and communication technology, and about their school experiences, work activities, and relationships with others. School principals complete a separate questionnaire.

In order to determine the content of the assessment, experts from OECD Member countries developed definitions for each domain, which guided the preparation of the testing instruments:

Reading literacy: The capacity to understand, use, and reflect on written texts in order to achieve one's goals and potential, develop knowledge, and participate in society. Mathematics literacy: The capacity to identify, understand, and engage in mathematics, and make well-founded judgments about the role that mathematics plays in the private, occupational, and social lives of constructive, concerned, and reflective citizens. Scientific literacy: The capacity to use scientific knowledge, identify questions, and draw evidence-based conclusions in order to understand and help make decisions about the natural world and the changes made to it through human activity.

One other domain was included in the PISA 2015 cycle, financial literacy, which was administered as an option in some countries, including some Canadian provinces. PISA 2015 was completely computerized.

The benefits of PISA

Canada invests significant public resources in the provision of elementary and secondary education, and Canadians are concerned about the quality of education provided by schools. The skills acquired by the end of secondary school are the essential foundation for further learning and for meeting the social and economic challenges of the future. PISA examines the level of achievement of 15-year-olds, providing an indication of the knowledge and skills they have acquired and their preparedness for continuing their studies or entering the workforce.

Results from PISA are valuable to educators, governments, social-policy analysts, and advocacy groups. Comparative information helps in the evaluation of the effectiveness of existing programs and practices, as well as in the understanding of the influences of socioeconomic and other factors on educational success.

PISA results

On an international level, Canada has performed very well in all of the PISA assessments. For example, in the 2018 assessment, Canadian 15-year-olds placed well above the OECD average and were among the top performers in reading. Of the 79 countries and economies participating in the assessment, only three— Beijing, Shanghai, Jiangsu, Zhejiang (B-S-J-Z) (China), Singapore, and Macao (China)—outperformed Canada.

source: [https://www.cmec.ca/251/Programme_for_International_Student_Assessment_\(PISA\).html](https://www.cmec.ca/251/Programme_for_International_Student_Assessment_(PISA).html)

```
[1]: # 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 sb

%matplotlib inline
```

Load in your dataset and describe its properties through the questions below. Try and motivate your exploration goals through this section.

```
[2]: # failed to read file as utf-8. changed to ISO-8859-1 instead.
df_pisa=pd.read_csv('pisa2012.csv', encoding = "ISO-8859-1")
```

```
C:\Users\saatt\AppData\Local\Continuum\anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (15,16,17,
21,22,23,24,25,26,30,31,36,37,45,65,123,155,156,157,158,159,160,161,162,163,164,
165,166,167,168,169,170,171,284,285,286,287,288,289,290,291,292,293,294,295,296,
297,298,299,300,301,302,303,307,308,309,310,311,312,313,314,315,316,317,318,319,
320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,
340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,376,377,
378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,
398,399,400,401,402,403,475) have mixed types.Specify dtype option on import or
set low_memory=False.
    interactivity=interactivity, compiler=compiler, result=result)
```

```
[280]: # read dataset dictionary
df_dict = pd.read_csv('pisadict2012.csv',encoding = "ISO-8859-1")
```

```
[281]: df_pisa.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Columns: 636 entries, Unnamed: 0 to VER_STU
dtypes: float64(250), int64(18), object(368)
memory usage: 2.3+ GB
```

```
[283]: df_dict.head()
```

```
[283]:
```

	CNT	Country code 3-character
0	SUBNATIO	Adjudicated sub-region code 7-digit code (3-di...
1	STRATUM	Stratum ID 7-character (cnt + region ID + orig...
2	OECD	OECD country
3	NC	National Centre 6-digit Code
4	SCHOOLID	School ID 7-digit (region ID + stratum ID + 3-...

```
[6]: df_pisa.shape, df_dict.shape
```

```
[6]: ((485490, 636), (635, 3))
```

```
[7]: df_pisa['ST04Q01'].value_counts()
```

```
[7]: Female      245064
      Male       240426
      Name: ST04Q01, dtype: int64
```

```
[8]: # Age counts
      df_pisa['AGE'].value_counts()
```

```
[8]: 15.58      42762
      15.67      42353
      15.75      41664
      15.83      41402
      15.92      41084
      16.00      41049
      15.42      40437
      15.50      40291
      16.08      39313
      16.17      38356
      15.33      28354
      16.25      26139
      15.25      11986
      16.33      10183
      15.17         1
      Name: AGE, dtype: int64
```

```
[9]: df_pisa['CNT'].value_counts()
```

```
[9]: Mexico                33806
      Italy                31073
      Spain               25313
      Canada              21544
      Brazil              19204
      ...
      Florida (USA)        1896
      Perm(Russian Federation) 1761
      Massachusetts (USA)  1723
      Connecticut (USA)    1697
      Liechtenstein         293
      Name: CNT, Length: 68, dtype: int64
```

```
[10]: # Unique countries
      df_pisa['CNT'].nunique()
```

```
[10]: 68
```

```
[11]: df_pisa.dtypes
```

```
[11]: Unnamed: 0      int64
      CNT          object
      SUBNATIO      int64
      STRATUM       object
      OECD          object
      ...
      W_FSTR80      float64
      WVARSTRR      int64
      VAR_UNIT      int64
      SENWGT_STU    float64
      VER_STU       object
      Length: 636, dtype: object
```

```
[12]: df_pisa.head()
```

```
[12]: Unnamed: 0      CNT  SUBNATIO  STRATUM      OECD      NC  SCHOOLID  \
0          1  Albania    80000  ALB0006  Non-OECD  Albania      1
1          2  Albania    80000  ALB0006  Non-OECD  Albania      1
2          3  Albania    80000  ALB0006  Non-OECD  Albania      1
3          4  Albania    80000  ALB0006  Non-OECD  Albania      1
4          5  Albania    80000  ALB0006  Non-OECD  Albania      1

      STIDSTD  ST01Q01  ST02Q01  ...  W_FSTR75  W_FSTR76  W_FSTR77  W_FSTR78  \
0          1         10        1.0  ...   13.7954   13.9235   13.1249   13.1249
1          2         10        1.0  ...   13.7954   13.9235   13.1249   13.1249
2          3          9        1.0  ...   12.7307   12.7307   12.7307   12.7307
3          4          9        1.0  ...   12.7307   12.7307   12.7307   12.7307
4          5          9        1.0  ...   12.7307   12.7307   12.7307   12.7307

      W_FSTR79  W_FSTR80  WVARSTRR  VAR_UNIT  SENWGT_STU  VER_STU
0      4.3389   13.0829         19         1      0.2098  22NOV13
1      4.3389   13.0829         19         1      0.2098  22NOV13
2      4.2436   12.7307         19         1      0.1999  22NOV13
3      4.2436   12.7307         19         1      0.1999  22NOV13
4      4.2436   12.7307         19         1      0.1999  22NOV13
```

```
[5 rows x 636 columns]
```

```
[14]: df_pisa.NC.value_counts()
```

```
[14]: Mexico          33806
      Italy          31073
      Spain          25313
      Canada         21544
      Brazil         19204
      ...
      New Zealand     4291
```

```

Iceland          3508
United Kingdom (Scotland)  2945
Perm (Russian Federation)  1761
Liechtenstein    293
Name: NC, Length: 66, dtype: int64

```

```
[15]: df_pisa.info(verbose=True, null_counts=True)
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Data columns (total 636 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Unnamed: 0      485490 non-null  int64
1   CNT             485490 non-null  object
2   SUBNATIO       485490 non-null  int64
3   STRATUM        485490 non-null  object
4   OECD           485490 non-null  object
5   NC             485490 non-null  object
6   SCHOOLID       485490 non-null  int64
7   STIDSTD        485490 non-null  int64
8   ST01Q01        485490 non-null  int64
9   ST02Q01        485438 non-null  float64
10  ST03Q01        485490 non-null  int64
11  ST03Q02        485490 non-null  int64
12  ST04Q01        485490 non-null  object
13  ST05Q01        476166 non-null  object
14  ST06Q01        457994 non-null  float64
15  ST07Q01        436690 non-null  object
16  ST07Q02        431278 non-null  object
17  ST07Q03        305687 non-null  object
18  ST08Q01        479143 non-null  object
19  ST09Q01        479131 non-null  object
20  ST115Q01       479269 non-null  float64
21  ST11Q01        460559 non-null  object
22  ST11Q02        441036 non-null  object
23  ST11Q03        400076 non-null  object
24  ST11Q04        390768 non-null  object
25  ST11Q05        348180 non-null  object
26  ST11Q06        337638 non-null  object
27  ST13Q01        457979 non-null  object
28  ST14Q01        390481 non-null  object
29  ST14Q02        407641 non-null  object
30  ST14Q03        382441 non-null  object
31  ST14Q04        304215 non-null  object
32  ST15Q01        467751 non-null  object
33  ST17Q01        443261 non-null  object

```

34	ST18Q01	371415	non-null	object
35	ST18Q02	387796	non-null	object
36	ST18Q03	362834	non-null	object
37	ST18Q04	292093	non-null	object
38	ST19Q01	451410	non-null	object
39	ST20Q01	476363	non-null	object
40	ST20Q02	472518	non-null	object
41	ST20Q03	469141	non-null	object
42	ST21Q01	32728	non-null	float64
43	ST25Q01	465496	non-null	object
44	ST26Q01	473079	non-null	object
45	ST26Q02	469693	non-null	object
46	ST26Q03	472020	non-null	object
47	ST26Q04	473877	non-null	object
48	ST26Q05	463178	non-null	object
49	ST26Q06	473182	non-null	object
50	ST26Q07	465860	non-null	object
51	ST26Q08	467094	non-null	object
52	ST26Q09	467249	non-null	object
53	ST26Q10	471242	non-null	object
54	ST26Q11	463566	non-null	object
55	ST26Q12	474039	non-null	object
56	ST26Q13	469115	non-null	object
57	ST26Q14	474076	non-null	object
58	ST26Q15	485490	non-null	int64
59	ST26Q16	485490	non-null	int64
60	ST26Q17	485490	non-null	int64
61	ST27Q01	477079	non-null	object
62	ST27Q02	476548	non-null	object
63	ST27Q03	473459	non-null	object
64	ST27Q04	472499	non-null	object
65	ST27Q05	469643	non-null	object
66	ST28Q01	473765	non-null	object
67	ST29Q01	315911	non-null	object
68	ST29Q02	315473	non-null	object
69	ST29Q03	314928	non-null	object
70	ST29Q04	314737	non-null	object
71	ST29Q05	315231	non-null	object
72	ST29Q06	314746	non-null	object
73	ST29Q07	315066	non-null	object
74	ST29Q08	315232	non-null	object
75	ST35Q01	315860	non-null	object
76	ST35Q02	315315	non-null	object
77	ST35Q03	314873	non-null	object
78	ST35Q04	315160	non-null	object
79	ST35Q05	314843	non-null	object
80	ST35Q06	313389	non-null	object
81	ST37Q01	314644	non-null	object

82	ST37Q02	314624	non-null	object
83	ST37Q03	313883	non-null	object
84	ST37Q04	313416	non-null	object
85	ST37Q05	313970	non-null	object
86	ST37Q06	313678	non-null	object
87	ST37Q07	314070	non-null	object
88	ST37Q08	314112	non-null	object
89	ST42Q01	313855	non-null	object
90	ST42Q02	313502	non-null	object
91	ST42Q03	312176	non-null	object
92	ST42Q04	311980	non-null	object
93	ST42Q05	312624	non-null	object
94	ST42Q06	312327	non-null	object
95	ST42Q07	312583	non-null	object
96	ST42Q08	312456	non-null	object
97	ST42Q09	312223	non-null	object
98	ST42Q10	312853	non-null	object
99	ST43Q01	314971	non-null	object
100	ST43Q02	314182	non-null	object
101	ST43Q03	313494	non-null	object
102	ST43Q04	313420	non-null	object
103	ST43Q05	313228	non-null	object
104	ST43Q06	313470	non-null	object
105	ST44Q01	314119	non-null	object
106	ST44Q03	313405	non-null	object
107	ST44Q04	312645	non-null	object
108	ST44Q05	312996	non-null	object
109	ST44Q07	312970	non-null	object
110	ST44Q08	313374	non-null	object
111	ST46Q01	313898	non-null	object
112	ST46Q02	313567	non-null	object
113	ST46Q03	312994	non-null	object
114	ST46Q04	312997	non-null	object
115	ST46Q05	313043	non-null	object
116	ST46Q06	312900	non-null	object
117	ST46Q07	312854	non-null	object
118	ST46Q08	312989	non-null	object
119	ST46Q09	313040	non-null	object
120	ST48Q01	294410	non-null	object
121	ST48Q02	289827	non-null	object
122	ST48Q03	298479	non-null	object
123	ST48Q04	267716	non-null	object
124	ST48Q05	287992	non-null	object
125	ST49Q01	313495	non-null	object
126	ST49Q02	313025	non-null	object
127	ST49Q03	312168	non-null	object
128	ST49Q04	312378	non-null	object
129	ST49Q05	312582	non-null	object

130	ST49Q06	312571	non-null	object
131	ST49Q07	312425	non-null	object
132	ST49Q09	312752	non-null	object
133	ST53Q01	309947	non-null	object
134	ST53Q02	309880	non-null	object
135	ST53Q03	309272	non-null	object
136	ST53Q04	308931	non-null	object
137	ST55Q01	307761	non-null	object
138	ST55Q02	308171	non-null	object
139	ST55Q03	306090	non-null	object
140	ST55Q04	304130	non-null	object
141	ST57Q01	301367	non-null	float64
142	ST57Q02	269808	non-null	float64
143	ST57Q03	283813	non-null	float64
144	ST57Q04	279657	non-null	float64
145	ST57Q05	289502	non-null	float64
146	ST57Q06	289428	non-null	float64
147	ST61Q01	312799	non-null	object
148	ST61Q02	312284	non-null	object
149	ST61Q03	311616	non-null	object
150	ST61Q04	310304	non-null	object
151	ST61Q05	311698	non-null	object
152	ST61Q06	311376	non-null	object
153	ST61Q07	311797	non-null	object
154	ST61Q08	311498	non-null	object
155	ST61Q09	309084	non-null	object
156	ST62Q01	306484	non-null	object
157	ST62Q02	307481	non-null	object
158	ST62Q03	306602	non-null	object
159	ST62Q04	306319	non-null	object
160	ST62Q06	306733	non-null	object
161	ST62Q07	306627	non-null	object
162	ST62Q08	306640	non-null	object
163	ST62Q09	307479	non-null	object
164	ST62Q10	306316	non-null	object
165	ST62Q11	305550	non-null	object
166	ST62Q12	306327	non-null	object
167	ST62Q13	306158	non-null	object
168	ST62Q15	306297	non-null	object
169	ST62Q16	306406	non-null	object
170	ST62Q17	306784	non-null	object
171	ST62Q19	307729	non-null	object
172	ST69Q01	299618	non-null	float64
173	ST69Q02	298601	non-null	float64
174	ST69Q03	291943	non-null	float64
175	ST70Q01	296878	non-null	float64
176	ST70Q02	298339	non-null	float64
177	ST70Q03	289068	non-null	float64

178	ST71Q01	255665	non-null	float64
179	ST72Q01	294163	non-null	float64
180	ST73Q01	309601	non-null	object
181	ST73Q02	308965	non-null	object
182	ST74Q01	309845	non-null	object
183	ST74Q02	309303	non-null	object
184	ST75Q01	309289	non-null	object
185	ST75Q02	308663	non-null	object
186	ST76Q01	308980	non-null	object
187	ST76Q02	308489	non-null	object
188	ST77Q01	315248	non-null	object
189	ST77Q02	314913	non-null	object
190	ST77Q04	314368	non-null	object
191	ST77Q05	314827	non-null	object
192	ST77Q06	314807	non-null	object
193	ST79Q01	314909	non-null	object
194	ST79Q02	314328	non-null	object
195	ST79Q03	313955	non-null	object
196	ST79Q04	313906	non-null	object
197	ST79Q05	313637	non-null	object
198	ST79Q06	313875	non-null	object
199	ST79Q07	314093	non-null	object
200	ST79Q08	314201	non-null	object
201	ST79Q10	313979	non-null	object
202	ST79Q11	313782	non-null	object
203	ST79Q12	313472	non-null	object
204	ST79Q15	313846	non-null	object
205	ST79Q17	314039	non-null	object
206	ST80Q01	314171	non-null	object
207	ST80Q04	313521	non-null	object
208	ST80Q05	312593	non-null	object
209	ST80Q06	312490	non-null	object
210	ST80Q07	312521	non-null	object
211	ST80Q08	312591	non-null	object
212	ST80Q09	311814	non-null	object
213	ST80Q10	312305	non-null	object
214	ST80Q11	312865	non-null	object
215	ST81Q01	313982	non-null	object
216	ST81Q02	313546	non-null	object
217	ST81Q03	312716	non-null	object
218	ST81Q04	312994	non-null	object
219	ST81Q05	313436	non-null	object
220	ST82Q01	311690	non-null	object
221	ST82Q02	311243	non-null	object
222	ST82Q03	310986	non-null	object
223	ST83Q01	313505	non-null	object
224	ST83Q02	313112	non-null	object
225	ST83Q03	312943	non-null	object

226	ST83Q04	312945	non-null	object
227	ST84Q01	310981	non-null	object
228	ST84Q02	311399	non-null	object
229	ST84Q03	310713	non-null	object
230	ST85Q01	312474	non-null	object
231	ST85Q02	312120	non-null	object
232	ST85Q03	311832	non-null	object
233	ST85Q04	311727	non-null	object
234	ST86Q01	313223	non-null	object
235	ST86Q02	312591	non-null	object
236	ST86Q03	312188	non-null	object
237	ST86Q04	312294	non-null	object
238	ST86Q05	311904	non-null	object
239	ST87Q01	311776	non-null	object
240	ST87Q02	312138	non-null	object
241	ST87Q03	310821	non-null	object
242	ST87Q04	310998	non-null	object
243	ST87Q05	310587	non-null	object
244	ST87Q06	310952	non-null	object
245	ST87Q07	310281	non-null	object
246	ST87Q08	310735	non-null	object
247	ST87Q09	311101	non-null	object
248	ST88Q01	311250	non-null	object
249	ST88Q02	310964	non-null	object
250	ST88Q03	310980	non-null	object
251	ST88Q04	311371	non-null	object
252	ST89Q02	311522	non-null	object
253	ST89Q03	311233	non-null	object
254	ST89Q04	311243	non-null	object
255	ST89Q05	311138	non-null	object
256	ST91Q01	311430	non-null	object
257	ST91Q02	310396	non-null	object
258	ST91Q03	309826	non-null	object
259	ST91Q04	309398	non-null	object
260	ST91Q05	309610	non-null	object
261	ST91Q06	309656	non-null	object
262	ST93Q01	312856	non-null	object
263	ST93Q03	312140	non-null	object
264	ST93Q04	311311	non-null	object
265	ST93Q06	312270	non-null	object
266	ST93Q07	312259	non-null	object
267	ST94Q05	312404	non-null	object
268	ST94Q06	312185	non-null	object
269	ST94Q09	311413	non-null	object
270	ST94Q10	311747	non-null	object
271	ST94Q14	312001	non-null	object
272	ST96Q01	311381	non-null	object
273	ST96Q02	311460	non-null	object

274	ST96Q03	311078	non-null	object
275	ST96Q05	311319	non-null	object
276	ST101Q01	311290	non-null	float64
277	ST101Q02	310906	non-null	float64
278	ST101Q03	310321	non-null	float64
279	ST101Q05	310655	non-null	float64
280	ST104Q01	310449	non-null	float64
281	ST104Q04	309969	non-null	float64
282	ST104Q05	310366	non-null	float64
283	ST104Q06	310156	non-null	float64
284	IC01Q01	296977	non-null	object
285	IC01Q02	297068	non-null	object
286	IC01Q03	295602	non-null	object
287	IC01Q04	297305	non-null	object
288	IC01Q05	296587	non-null	object
289	IC01Q06	294773	non-null	object
290	IC01Q07	296116	non-null	object
291	IC01Q08	297109	non-null	object
292	IC01Q09	296855	non-null	object
293	IC01Q10	297451	non-null	object
294	IC01Q11	295118	non-null	object
295	IC02Q01	296975	non-null	object
296	IC02Q02	295618	non-null	object
297	IC02Q03	294625	non-null	object
298	IC02Q04	296944	non-null	object
299	IC02Q05	296167	non-null	object
300	IC02Q06	295830	non-null	object
301	IC02Q07	294249	non-null	object
302	IC03Q01	293216	non-null	object
303	IC04Q01	296305	non-null	object
304	IC05Q01	485490	non-null	int64
305	IC06Q01	485490	non-null	int64
306	IC07Q01	485490	non-null	int64
307	IC08Q01	294123	non-null	object
308	IC08Q02	293646	non-null	object
309	IC08Q03	293162	non-null	object
310	IC08Q04	293249	non-null	object
311	IC08Q05	293822	non-null	object
312	IC08Q06	293744	non-null	object
313	IC08Q07	293570	non-null	object
314	IC08Q08	293053	non-null	object
315	IC08Q09	293496	non-null	object
316	IC08Q11	293431	non-null	object
317	IC09Q01	292880	non-null	object
318	IC09Q02	292463	non-null	object
319	IC09Q03	291964	non-null	object
320	IC09Q04	292024	non-null	object
321	IC09Q05	291721	non-null	object

322	IC09Q06	291982	non-null	object
323	IC09Q07	292051	non-null	object
324	IC10Q01	291811	non-null	object
325	IC10Q02	291025	non-null	object
326	IC10Q03	290262	non-null	object
327	IC10Q04	290907	non-null	object
328	IC10Q05	291025	non-null	object
329	IC10Q06	290268	non-null	object
330	IC10Q07	290565	non-null	object
331	IC10Q08	290770	non-null	object
332	IC10Q09	290815	non-null	object
333	IC11Q01	289894	non-null	object
334	IC11Q02	289427	non-null	object
335	IC11Q03	288868	non-null	object
336	IC11Q04	289085	non-null	object
337	IC11Q05	289082	non-null	object
338	IC11Q06	289269	non-null	object
339	IC11Q07	289277	non-null	object
340	IC22Q01	290487	non-null	object
341	IC22Q02	290000	non-null	object
342	IC22Q04	289671	non-null	object
343	IC22Q06	289239	non-null	object
344	IC22Q07	288871	non-null	object
345	IC22Q08	288959	non-null	object
346	EC01Q01	156043	non-null	object
347	EC02Q01	156008	non-null	object
348	EC03Q01	160533	non-null	object
349	EC03Q02	165310	non-null	object
350	EC03Q03	165197	non-null	object
351	EC03Q04	165164	non-null	object
352	EC03Q05	164984	non-null	object
353	EC03Q06	164935	non-null	object
354	EC03Q07	165136	non-null	object
355	EC03Q08	164921	non-null	object
356	EC03Q09	164992	non-null	object
357	EC03Q10	132539	non-null	object
358	EC04Q01A	169730	non-null	float64
359	EC04Q01B	169765	non-null	float64
360	EC04Q01C	169779	non-null	float64
361	EC04Q02A	169783	non-null	float64
362	EC04Q02B	169784	non-null	float64
363	EC04Q02C	169798	non-null	float64
364	EC04Q03A	169796	non-null	float64
365	EC04Q03B	169786	non-null	float64
366	EC04Q03C	169799	non-null	float64
367	EC04Q04A	169655	non-null	float64
368	EC04Q04B	169641	non-null	float64
369	EC04Q04C	169656	non-null	float64

370	EC04Q05A	169716	non-null	float64
371	EC04Q05B	169716	non-null	float64
372	EC04Q05C	169725	non-null	float64
373	EC04Q06A	169643	non-null	float64
374	EC04Q06B	169640	non-null	float64
375	EC04Q06C	169636	non-null	float64
376	EC05Q01	129658	non-null	object
377	EC06Q01	40345	non-null	object
378	EC07Q01	44012	non-null	object
379	EC07Q02	43219	non-null	object
380	EC07Q03	42277	non-null	object
381	EC07Q04	42832	non-null	object
382	EC07Q05	42864	non-null	object
383	EC08Q01	43633	non-null	object
384	EC08Q02	43393	non-null	object
385	EC08Q03	43489	non-null	object
386	EC08Q04	43330	non-null	object
387	EC09Q03	118588	non-null	object
388	EC10Q01	43293	non-null	object
389	EC11Q02	118637	non-null	object
390	EC11Q03	118659	non-null	object
391	EC12Q01	42909	non-null	object
392	ST22Q01	40721	non-null	object
393	ST23Q01	13730	non-null	object
394	ST23Q02	13512	non-null	object
395	ST23Q03	13497	non-null	object
396	ST23Q04	13411	non-null	object
397	ST23Q05	13450	non-null	object
398	ST23Q06	13373	non-null	object
399	ST23Q07	13411	non-null	object
400	ST23Q08	13382	non-null	object
401	ST24Q01	13457	non-null	object
402	ST24Q02	13351	non-null	object
403	ST24Q03	13281	non-null	object
404	CLCUSE1	412337	non-null	object
405	CLCUSE301	485490	non-null	int64
406	CLCUSE302	485490	non-null	int64
407	DEFFORT	485490	non-null	int64
408	QUESTID	485490	non-null	object
409	BOOKID	485490	non-null	object
410	EASY	485490	non-null	object
411	AGE	485374	non-null	float64
412	GRADE	484617	non-null	float64
413	PROGN	485490	non-null	object
414	ANXMAT	314764	non-null	float64
415	ATSCHL	312584	non-null	float64
416	ATTLNACT	311675	non-null	float64
417	BELONG	313399	non-null	float64

418	BFMJ2	416150	non-null	float64
419	BMMJ1	364814	non-null	float64
420	CLSMAN	312708	non-null	float64
421	COBN_F	481825	non-null	object
422	COBN_M	481843	non-null	object
423	COBN_S	481836	non-null	object
424	COGACT	314557	non-null	float64
425	CULTDIST	13380	non-null	float64
426	CULTPOS	471357	non-null	float64
427	DISCLIMA	314777	non-null	float64
428	ENTUSE	295195	non-null	float64
429	ESCS	473648	non-null	float64
430	EXAPPLM	313279	non-null	float64
431	EXPUREM	312602	non-null	float64
432	FAILMAT	314448	non-null	float64
433	FAMCON	310304	non-null	float64
434	FAMCONC	308442	non-null	float64
435	FAMSTRUC	429058	non-null	float64
436	FISCED	452903	non-null	object
437	HEDRES	477772	non-null	float64
438	HERITCUL	13496	non-null	float64
439	HISCED	473091	non-null	object
440	HISEI	450621	non-null	float64
441	HOMEPOS	479807	non-null	float64
442	HOMSCH	293194	non-null	float64
443	HOSTCUL	13598	non-null	float64
444	ICTATTNEG	289744	non-null	float64
445	ICTATTPOS	290490	non-null	float64
446	ICTHOME	298740	non-null	float64
447	ICTRES	477754	non-null	float64
448	ICTSCH	297995	non-null	float64
449	IMMIG	471793	non-null	object
450	INFOCAR	165792	non-null	float64
451	INFOJOB1	83305	non-null	float64
452	INFOJOB2	83305	non-null	float64
453	INSTMOT	316322	non-null	float64
454	INTMAT	316708	non-null	float64
455	ISCEDD	485438	non-null	object
456	ISCEDL	485438	non-null	object
457	ISCEDO	485438	non-null	object
458	LANGCOMM	44094	non-null	float64
459	LANGN	481765	non-null	object
460	LANGRPPD	43137	non-null	float64
461	LMINS	282866	non-null	float64
462	MATBEH	313847	non-null	float64
463	MATHEFF	315948	non-null	float64
464	MATINTFC	301360	non-null	float64
465	MATWKETH	314501	non-null	float64

466	MISCED	467085	non-null	object
467	MMINS	283303	non-null	float64
468	MTSUP	313599	non-null	float64
469	OCOD1	483887	non-null	object
470	OCOD2	482936	non-null	object
471	OPENPS	312766	non-null	float64
472	OUTHOURS	308799	non-null	float64
473	PARED	473091	non-null	float64
474	PERSEV	313172	non-null	float64
475	REPEAT	461117	non-null	object
476	SCMAT	314607	non-null	float64
477	SMINS	270914	non-null	float64
478	STUDREL	313860	non-null	float64
479	SUBNORM	316323	non-null	float64
480	TCHBEHFA	314678	non-null	float64
481	TCHBEHSO	315114	non-null	float64
482	TCHBEHTD	315519	non-null	float64
483	TEACHSUP	316371	non-null	float64
484	TESTLANG	484697	non-null	object
485	TIMEINT	297074	non-null	float64
486	USEMATH	290260	non-null	float64
487	USESCH	292585	non-null	float64
488	WEALTH	479597	non-null	float64
489	ANCATSCHL	306835	non-null	float64
490	ANCATTLNACT	306487	non-null	float64
491	ANCBELONG	307640	non-null	float64
492	ANCCLSMAN	308467	non-null	float64
493	ANCCOGACT	308150	non-null	float64
494	ANCINSTMOT	155221	non-null	float64
495	ANCINTMAT	155280	non-null	float64
496	ANCMATWKETH	153879	non-null	float64
497	ANCMTSUP	308631	non-null	float64
498	ANCSCMAT	306948	non-null	float64
499	ANCSTUDREL	308058	non-null	float64
500	ANCSUBNORM	155233	non-null	float64
501	PV1MATH	485490	non-null	float64
502	PV2MATH	485490	non-null	float64
503	PV3MATH	485490	non-null	float64
504	PV4MATH	485490	non-null	float64
505	PV5MATH	485490	non-null	float64
506	PV1MACC	473031	non-null	float64
507	PV2MACC	473031	non-null	float64
508	PV3MACC	473031	non-null	float64
509	PV4MACC	473031	non-null	float64
510	PV5MACC	473031	non-null	float64
511	PV1MACQ	473031	non-null	float64
512	PV2MACQ	473031	non-null	float64
513	PV3MACQ	473031	non-null	float64

514	PV4MACQ	473031	non-null	float64
515	PV5MACQ	473031	non-null	float64
516	PV1MACS	473031	non-null	float64
517	PV2MACS	473031	non-null	float64
518	PV3MACS	473031	non-null	float64
519	PV4MACS	473031	non-null	float64
520	PV5MACS	473031	non-null	float64
521	PV1MACU	473031	non-null	float64
522	PV2MACU	473031	non-null	float64
523	PV3MACU	473031	non-null	float64
524	PV4MACU	473031	non-null	float64
525	PV5MACU	473031	non-null	float64
526	PV1MAPE	471439	non-null	float64
527	PV2MAPE	471439	non-null	float64
528	PV3MAPE	471439	non-null	float64
529	PV4MAPE	471439	non-null	float64
530	PV5MAPE	471439	non-null	float64
531	PV1MAPF	471439	non-null	float64
532	PV2MAPF	471439	non-null	float64
533	PV3MAPF	471439	non-null	float64
534	PV4MAPF	471439	non-null	float64
535	PV5MAPF	471439	non-null	float64
536	PV1MAPI	471439	non-null	float64
537	PV2MAPI	471439	non-null	float64
538	PV3MAPI	471439	non-null	float64
539	PV4MAPI	471439	non-null	float64
540	PV5MAPI	471439	non-null	float64
541	PV1READ	485490	non-null	float64
542	PV2READ	485490	non-null	float64
543	PV3READ	485490	non-null	float64
544	PV4READ	485490	non-null	float64
545	PV5READ	485490	non-null	float64
546	PV1SCIE	485490	non-null	float64
547	PV2SCIE	485490	non-null	float64
548	PV3SCIE	485490	non-null	float64
549	PV4SCIE	485490	non-null	float64
550	PV5SCIE	485490	non-null	float64
551	W_FSTUWT	485490	non-null	float64
552	W_FSTR1	485490	non-null	float64
553	W_FSTR2	485490	non-null	float64
554	W_FSTR3	485490	non-null	float64
555	W_FSTR4	485490	non-null	float64
556	W_FSTR5	485490	non-null	float64
557	W_FSTR6	485490	non-null	float64
558	W_FSTR7	485490	non-null	float64
559	W_FSTR8	485490	non-null	float64
560	W_FSTR9	485490	non-null	float64
561	W_FSTR10	485490	non-null	float64

562	W_FSTR11	485490	non-null	float64
563	W_FSTR12	485490	non-null	float64
564	W_FSTR13	485490	non-null	float64
565	W_FSTR14	485490	non-null	float64
566	W_FSTR15	485490	non-null	float64
567	W_FSTR16	485490	non-null	float64
568	W_FSTR17	485490	non-null	float64
569	W_FSTR18	485490	non-null	float64
570	W_FSTR19	485490	non-null	float64
571	W_FSTR20	485490	non-null	float64
572	W_FSTR21	485490	non-null	float64
573	W_FSTR22	485490	non-null	float64
574	W_FSTR23	485490	non-null	float64
575	W_FSTR24	485490	non-null	float64
576	W_FSTR25	485490	non-null	float64
577	W_FSTR26	485490	non-null	float64
578	W_FSTR27	485490	non-null	float64
579	W_FSTR28	485490	non-null	float64
580	W_FSTR29	485490	non-null	float64
581	W_FSTR30	485490	non-null	float64
582	W_FSTR31	485490	non-null	float64
583	W_FSTR32	485490	non-null	float64
584	W_FSTR33	485490	non-null	float64
585	W_FSTR34	485490	non-null	float64
586	W_FSTR35	485490	non-null	float64
587	W_FSTR36	485490	non-null	float64
588	W_FSTR37	485490	non-null	float64
589	W_FSTR38	485490	non-null	float64
590	W_FSTR39	485490	non-null	float64
591	W_FSTR40	485490	non-null	float64
592	W_FSTR41	485490	non-null	float64
593	W_FSTR42	485490	non-null	float64
594	W_FSTR43	485490	non-null	float64
595	W_FSTR44	485490	non-null	float64
596	W_FSTR45	485490	non-null	float64
597	W_FSTR46	485490	non-null	float64
598	W_FSTR47	485490	non-null	float64
599	W_FSTR48	485490	non-null	float64
600	W_FSTR49	485490	non-null	float64
601	W_FSTR50	485490	non-null	float64
602	W_FSTR51	485490	non-null	float64
603	W_FSTR52	485490	non-null	float64
604	W_FSTR53	485490	non-null	float64
605	W_FSTR54	485490	non-null	float64
606	W_FSTR55	485490	non-null	float64
607	W_FSTR56	485490	non-null	float64
608	W_FSTR57	485490	non-null	float64
609	W_FSTR58	485490	non-null	float64

610	W_FSTR59	485490	non-null	float64
611	W_FSTR60	485490	non-null	float64
612	W_FSTR61	485490	non-null	float64
613	W_FSTR62	485490	non-null	float64
614	W_FSTR63	485490	non-null	float64
615	W_FSTR64	485490	non-null	float64
616	W_FSTR65	485490	non-null	float64
617	W_FSTR66	485490	non-null	float64
618	W_FSTR67	485490	non-null	float64
619	W_FSTR68	485490	non-null	float64
620	W_FSTR69	485490	non-null	float64
621	W_FSTR70	485490	non-null	float64
622	W_FSTR71	485490	non-null	float64
623	W_FSTR72	485490	non-null	float64
624	W_FSTR73	485490	non-null	float64
625	W_FSTR74	485490	non-null	float64
626	W_FSTR75	485490	non-null	float64
627	W_FSTR76	485490	non-null	float64
628	W_FSTR77	485490	non-null	float64
629	W_FSTR78	485490	non-null	float64
630	W_FSTR79	485490	non-null	float64
631	W_FSTR80	485490	non-null	float64
632	WVARSTRR	485490	non-null	int64
633	VAR_UNIT	485490	non-null	int64
634	SENWGT_STU	485490	non-null	float64
635	VER_STU	485490	non-null	object

dtypes: float64(250), int64(18), object(368)
memory usage: 2.3+ GB

1.3 Quick data analysis

1. Dataset has 485490 rows, and 635 columns
2. this is a huge dataset, we need to filter out columns based on question we need to answer. since the focus of PISA is math, science and reading. we need to get columns with such important details or have impact. such as:
 - math/reading/science score
 - internet
 - mother, father education
 - gender
 - mother still available and at home
 - father still available and at home
 - mother highest schooling
 - father highest schooling
 - Math scores (5 items)
 - reading scores (5 items)
 - science scores (5 items)
3. Ration of females to male also 50/50 (more female)

4. students coming from 68 countries (67 to be exact, USA represented by two records Connecticut & Massachusetts. Liechtenstein with total count of 293, while mixco comes first with total rows of 33806. Wonder why Mixco gets the lion share?

[]:

1.4 Data wrangling

During my analysis i will try answering the following quetions:

1. How will student from various countries perform in math, reading, and science cosidering mother, father education, work, availability of internet, and textbox?
2. Are there countries that perform better than others?
3. compare results based on gender?

```
[95]: #Rename some of the columns to be more intiuitive and useful
cols=['CNT', 'STIDSTD', 'AGE', 'ST04Q01', 'ST11Q01',
      ↪ 'ST11Q02', 'ST13Q01', 'ST15Q01', 'ST17Q01', 'ST19Q01', 'ST26Q06', 'ST26Q10', 'PV1MATH', 'PV2MATH', 'PV3MATH', 'PV4MATH']
df_pisa_clean = pd.read_csv('pisa2012.csv', usecols=cols, encoding =
      ↪ "ISO-8859-1")
```

```
C:\Users\saatt\AppData\Local\Continuum\anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (21,22)
have mixed types.Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)
```

[96]: df_pisa_clean.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Data columns (total 27 columns):
#   Column      Non-Null Count  Dtype
---  -
0   CNT          485490 non-null  object
1   STIDSTD      485490 non-null  int64
2   ST04Q01     485490 non-null  object
3   ST11Q01     460559 non-null  object
4   ST11Q02     441036 non-null  object
5   ST13Q01     457979 non-null  object
6   ST15Q01     467751 non-null  object
7   ST17Q01     443261 non-null  object
8   ST19Q01     451410 non-null  object
9   ST26Q06     473182 non-null  object
10  ST26Q10     471242 non-null  object
11  AGE          485374 non-null  float64
12  PV1MATH     485490 non-null  float64
13  PV2MATH     485490 non-null  float64
14  PV3MATH     485490 non-null  float64
15  PV4MATH     485490 non-null  float64
```

```

16 PV5MATH 485490 non-null float64
17 PV1READ 485490 non-null float64
18 PV2READ 485490 non-null float64
19 PV3READ 485490 non-null float64
20 PV4READ 485490 non-null float64
21 PV5READ 485490 non-null float64
22 PV1SCIE 485490 non-null float64
23 PV2SCIE 485490 non-null float64
24 PV3SCIE 485490 non-null float64
25 PV4SCIE 485490 non-null float64
26 PV5SCIE 485490 non-null float64
dtypes: float64(16), int64(1), object(10)
memory usage: 100.0+ MB

```

1.5 fill columns with nan values as below

```

[97]: #fillna with 'unknown' for nan values of mother at home: ST11Q01 & father at_
      ↪home ST11Q02
      #mother at home with unknown
df_pisa_clean['ST11Q01'].fillna('unknown', inplace=True)
      # fillna father at home with unknown
df_pisa_clean['ST11Q02'].fillna('unknown', inplace=True)
      #mother highest schooling
df_pisa_clean['ST13Q01'].fillna('unknown', inplace=True)
      #mother current job with unknown
df_pisa_clean['ST15Q01'].fillna('unknown', inplace=True)
      #father highest schooling with unknown
df_pisa_clean['ST17Q01'].fillna('unknown', inplace=True)
      #father current job with unknown
df_pisa_clean['ST19Q01'].fillna('unknown', inplace=True)
      #internet with unknown
df_pisa_clean['ST26Q06'].fillna('unknown', inplace=True)
      #testbook with unknown
df_pisa_clean['ST26Q10'].fillna('unknown', inplace=True)
      #age with mean age
mean=df_pisa_clean['AGE'].mean()
df_pisa_clean['AGE'].fillna(mean, inplace=True)

```

```

[98]: df_pisa_clean.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Data columns (total 27 columns):
#   Column      Non-Null Count  Dtype
---  -
0   CNT         485490 non-null  object
1   STIDSTD     485490 non-null  int64
2   ST04Q01     485490 non-null  object

```

```

3  ST11Q01  485490 non-null  object
4  ST11Q02  485490 non-null  object
5  ST13Q01  485490 non-null  object
6  ST15Q01  485490 non-null  object
7  ST17Q01  485490 non-null  object
8  ST19Q01  485490 non-null  object
9  ST26Q06  485490 non-null  object
10 ST26Q10  485490 non-null  object
11 AGE      485490 non-null  float64
12 PV1MATH  485490 non-null  float64
13 PV2MATH  485490 non-null  float64
14 PV3MATH  485490 non-null  float64
15 PV4MATH  485490 non-null  float64
16 PV5MATH  485490 non-null  float64
17 PV1READ  485490 non-null  float64
18 PV2READ  485490 non-null  float64
19 PV3READ  485490 non-null  float64
20 PV4READ  485490 non-null  float64
21 PV5READ  485490 non-null  float64
22 PV1SCIE  485490 non-null  float64
23 PV2SCIE  485490 non-null  float64
24 PV3SCIE  485490 non-null  float64
25 PV4SCIE  485490 non-null  float64
26 PV5SCIE  485490 non-null  float64
dtypes: float64(16), int64(1), object(10)
memory usage: 100.0+ MB

```

```
[99]: df_pisa_clean['ST11Q02'].value_counts()
```

```

[99]: Yes          372161
      No           68875
      unknown      44454
      Name: ST11Q02, dtype: int64

```

1.6 Rename columns to be more intuitive and easy to handle

```

[100]: df_pisa_clean.rename(columns={'CNT': 'country', 'STIDSTD': 'student_id', 'AGE':
    ↪ 'age', 'ST04Q01': 'gender',
                                     'ST11Q01': 'mother_at_home', 'ST11Q02':
    ↪ 'father_at_home',
                                     'ST13Q01': 'mother_grade', 'ST15Q01':
    ↪ 'mother_job', 'ST17Q01': 'father_grade',
                                     'ST19Q01': 'father_job', 'ST26Q06':
    ↪ 'internet', 'ST26Q10': 'textbook'}, inplace=True)

```

```
[101]: df_pisa_clean.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Data columns (total 27 columns):
#   Column                Non-Null Count  Dtype
---  -
0   country                485490 non-null object
1   student_id             485490 non-null int64
2   gender                 485490 non-null object
3   mother_at_home         485490 non-null object
4   father_at_home         485490 non-null object
5   mother_grade           485490 non-null object
6   mother_job             485490 non-null object
7   father_grade           485490 non-null object
8   father_job             485490 non-null object
9   internet               485490 non-null object
10  textbook               485490 non-null object
11  age                    485490 non-null float64
12  PV1MATH                485490 non-null float64
13  PV2MATH                485490 non-null float64
14  PV3MATH                485490 non-null float64
15  PV4MATH                485490 non-null float64
16  PV5MATH                485490 non-null float64
17  PV1READ                485490 non-null float64
18  PV2READ                485490 non-null float64
19  PV3READ                485490 non-null float64
20  PV4READ                485490 non-null float64
21  PV5READ                485490 non-null float64
22  PV1SCIE                485490 non-null float64
23  PV2SCIE                485490 non-null float64
24  PV3SCIE                485490 non-null float64
25  PV4SCIE                485490 non-null float64
26  PV5SCIE                485490 non-null float64
dtypes: float64(16), int64(1), object(10)
memory usage: 100.0+ MB

```

```

[102]: # calculate mean value for math, reading, science
# add a new column for total of means
df_pisa_clean['math']=(df_pisa_clean['PV1MATH']+df_pisa_clean['PV2MATH']+df_pisa_clean['PV3MAT
↪5
df_pisa_clean['reading']=(df_pisa_clean['PV1READ']+df_pisa_clean['PV2READ']+df_pisa_clean['PV3
↪5
df_pisa_clean['science']=(df_pisa_clean['PV1SCIE']+df_pisa_clean['PV2SCIE']+df_pisa_clean['PV3
↪5
df_pisa_clean['total']=(df_pisa_clean['math']+df_pisa_clean['reading']+df_pisa_clean['science']
↪3

```

```
[103]: # since we are used the average value for each of Math, reading and science, no
        ↪need to keep the old columns.
df_pisa_clean.drop(columns=['PV1MATH', 'PV2MATH', 'PV3MATH', 'PV4MATH', 'PV5MATH',
                           'PV1READ', 'PV2READ', 'PV3READ', 'PV4READ', 'PV5READ',
                           ↪
                           'PV1SCIE', 'PV2SCIE', 'PV3SCIE', 'PV4SCIE', 'PV5SCIE'], inplace=True)
```

```
[104]: df_pisa_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   country               485490 non-null  object
1   student_id           485490 non-null  int64
2   gender               485490 non-null  object
3   mother_at_home       485490 non-null  object
4   father_at_home       485490 non-null  object
5   mother_grade         485490 non-null  object
6   mother_job           485490 non-null  object
7   father_grade         485490 non-null  object
8   father_job           485490 non-null  object
9   internet             485490 non-null  object
10  textbook             485490 non-null  object
11  age                  485490 non-null  float64
12  math                 485490 non-null  float64
13  reading              485490 non-null  float64
14  science              485490 non-null  float64
15  total                485490 non-null  float64
dtypes: float64(5), int64(1), object(10)
memory usage: 59.3+ MB
```

```
[105]: df_pisa_clean['mother_grade'].value_counts()
```

```
[105]: <ISCED level 3A>                236993
       <ISCED level 3B, 3C>         83048
       <ISCED level 2>              82614
       <ISCED level 1>              36556
       unknown                     27511
       She did not complete <ISCED level 1> 18768
       Name: mother_grade, dtype: int64
```

```
[106]: df_pisa_clean['father_grade'].value_counts()
```

```
[106]: <ISCED level 3A>                215280
       <ISCED level 3B, 3C>         91179
```



```

<ISCED level 2>                84329
unknown                        42229
<ISCED level 1>                35938
He did not complete <ISCED level 1> 16535
Name: father_grade, dtype: int64

```

```

[107]: # replace values to be less wordy
df_pisa_clean['mother_grade'].replace({'<ISCED level 3A> ':'ISCED 3A', '<ISCED_
↳level 3B, 3C> ':'ISCED 3B-3C',
                                     '<ISCED level 2> ':'ISCED 2', '<ISCED_
↳level 1> ':'ISCED 1',
                                     'She did not complete <ISCED level 1> ':'
↳incomplete ISCED 1'}, inplace=True)

df_pisa_clean['father_grade'].replace({'<ISCED level 3A> ':'ISCED 3A', '<ISCED_
↳level 3B, 3C> ':'ISCED 3B-3C',
                                     '<ISCED level 2> ':'ISCED 2', '<ISCED_
↳level 1> ':'ISCED 1',
                                     'He did not complete <ISCED level 1> ':'
↳incomplete ISCED 1'}, inplace=True)

```

```

[108]: df_pisa_clean['mother_grade'].value_counts()

```

```

[108]: ISCED 3A                236993
ISCED 3B-3C                83048
ISCED 2                    82614
ISCED 1                    36556
unknown                    27511
incomplete ISCED 1        18768
Name: mother_grade, dtype: int64

```

```

[109]: df_pisa_clean['father_grade'].value_counts()

```

```

[109]: ISCED 3A                215280
ISCED 3B-3C                91179
ISCED 2                    84329
unknown                    42229
ISCED 1                    35938
incomplete ISCED 1        16535
Name: father_grade, dtype: int64

```

```

[110]: df_pisa_clean['internet'].value_counts()

```

```

[110]: Yes                402040
No                71142
unknown          12308
Name: internet, dtype: int64

```

```
[111]: df_pisa_clean['textbook'].value_counts()
```

```
[111]: Yes          389408
      No           81834
      unknown      14248
      Name: textbook, dtype: int64
```

```
[112]: df_pisa_clean['father_job'].value_counts()
```

```
[112]: Working full-time <for pay>          339697
      Working part-time <for pay>         49503
      Other (e.g. home duties, retired)   40588
      unknown                             34080
      Not working, but looking for a job  21622
      Name: father_job, dtype: int64
```

```
[113]: df_pisa_clean['mother_job'].value_counts()
```

```
[113]: Working full-time <for pay>          219095
      Other (e.g. home duties, retired)   138841
      Working part-time <for pay>         78237
      Not working, but looking for a job  31578
      unknown                             17739
      Name: mother_job, dtype: int64
```

```
[114]: # replace values to be less wordy
df_pisa_clean['mother_job'].replace({'Working full-time <for pay> ' :␣
    ↳ 'full-time',
                                     'Other (e.g. home duties, retired) ' :␣
    ↳ 'others',
                                     'Working part-time <for pay>' :␣
    ↳ 'part-time',
                                     'Not working, but looking for a job ' :␣
    ↳ 'unemployed'}, inplace=True)

df_pisa_clean['father_job'].replace({'Working full-time <for pay> ' :␣
    ↳ 'full-time',
                                     'Other (e.g. home duties, retired) ' :␣
    ↳ 'others',
                                     'Working part-time <for pay>' :␣
    ↳ 'part-time',
                                     'Not working, but looking for a job' :␣
    ↳ 'unemployed'}, inplace=True)
```

```
[115]: df_pisa_clean['mother_job'].value_counts()
```

```
[115]: full-time      219095
      others        138841
      part-time     78237
      unemployed    31578
      unknown       17739
      Name: mother_job, dtype: int64
```

```
[116]: df_pisa_clean['father_job'].value_counts()
```

```
[116]: full-time      339697
      part-time     49503
      others        40588
      unknown       34080
      unemployed    21622
      Name: father_job, dtype: int64
```

```
[117]: # grades is an ordinal category, assuming unknown as the lowest, and ISCED
      ↪ 3B-3C is the max
      grades_category=['unknown','incomplete ISCED 1','ISCED 1','ISCED 2','ISCED
      ↪ 3A','ISCED 3B-3C',]
      grades=pd.api.types.CategoricalDtype(ordered=True,categories=grades_category)
      df_pisa_clean['mother_grade']=df_pisa_clean['mother_grade'].astype(grades)
      df_pisa_clean['father_grade']=df_pisa_clean['father_grade'].astype(grades)
```

```
[118]: # job category is more a nominal datatype, there is not sequence we can use to
      ↪ say this is higher than the other.

      job_category=['unknown', 'unemployed','part-time', 'full-time', 'others']
      jobs=pd.api.types.CategoricalDtype(ordered=False,categories=job_category)
      df_pisa_clean['mother_job']=df_pisa_clean['mother_job'].astype(jobs)
      df_pisa_clean['father_job']=df_pisa_clean['father_job'].astype(jobs)
```

```
[119]: # modify type to category for internet, textbook, father_at_home, mother_at_home

      df_pisa_clean = df_pisa_clean.astype({'internet':'category', 'textbook':
      ↪ 'category',
      ↪ 'gender':'category','mother_at_home':
      ↪ 'category','father_at_home':'category'})
```

```
[120]: df_pisa_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   country                485490 non-null object
```

```

1  student_id      485490 non-null  int64
2  gender          485490 non-null  category
3  mother_at_home  485490 non-null  category
4  father_at_home  485490 non-null  category
5  mother_grade    485490 non-null  category
6  mother_job      485490 non-null  category
7  father_grade    485490 non-null  category
8  father_job      485490 non-null  category
9  internet        485490 non-null  category
10 textbook        485490 non-null  category
11 age            485490 non-null  float64
12 math           485490 non-null  float64
13 reading        485490 non-null  float64
14 science        485490 non-null  float64
15 total          485490 non-null  float64
dtypes: category(9), float64(5), int64(1), object(1)
memory usage: 30.1+ MB

```

```
[121]: df_pisa_clean.head()
```

```

[121]:   country  student_id  gender  mother_at_home  father_at_home  \
0  Albania         1  Female                Yes                Yes
1  Albania         2  Female                Yes                Yes
2  Albania         3  Female                Yes                Yes
3  Albania         4  Female                Yes                Yes
4  Albania         5  Female                Yes                Yes

      mother_grade  mother_job  father_grade  father_job  internet  textbook  \
0      ISCED 3A      others      ISCED 3A  part-time        No        Yes
1      ISCED 3A  full-time      ISCED 3A  full-time        Yes        Yes
2      ISCED 3B-3C  full-time      ISCED 3A  full-time        Yes        Yes
3      ISCED 3B-3C  full-time      ISCED 3A  full-time        Yes        Yes
4  incomplete ISCED 1  part-time  ISCED 3B-3C  part-time        Yes        Yes

      age      math      reading      science      total
0  16.17  366.18634  261.01424  371.91348  333.038020
1  16.17  470.56396  384.68832  478.12382  444.458700
2  15.58  505.53824  405.18154  486.60946  465.776413
3  15.67  449.45476  477.46376  453.97240  460.296973
4  15.50  385.50398  256.01010  367.15778  336.223953

```

```
[378]: df_pisa_clean.to_csv('df_pisa_clean.csv', index=False)
```

1.6.1 What is the structure of your dataset?

the data structure is very wide, has consideration for many many cases and factors such as job of the father, job for mother, every aspect of living details like if child got a separate room, type of electronic devices, and so much more. Such details seems to have

purposes beyond the objectives of the project, and need deep analysis for optimal use, and probably other aspect which i am not aware of.

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

Main aspect of my interest is math, science, and reading. this is where i will focus my analysis on.

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

Math, science, and reading each has five separate scores. i have used the mean for each as a separate column. this new value will be used by focus my analysis on.

1.7 Univariate Exploration

In this section, investigate distributions of individual variables. If you see unusual points or outliers, take a deeper look to clean things up and prepare yourself to look at relationships between variables.

Make sure that, after every plot or related series of plots, that you include a Markdown cell with comments about what you observed, and what you plan on investigating next.

```
[122]: df_pisa_clean[['math', 'reading', 'science', 'total']].describe()
```

```
[122]:
```

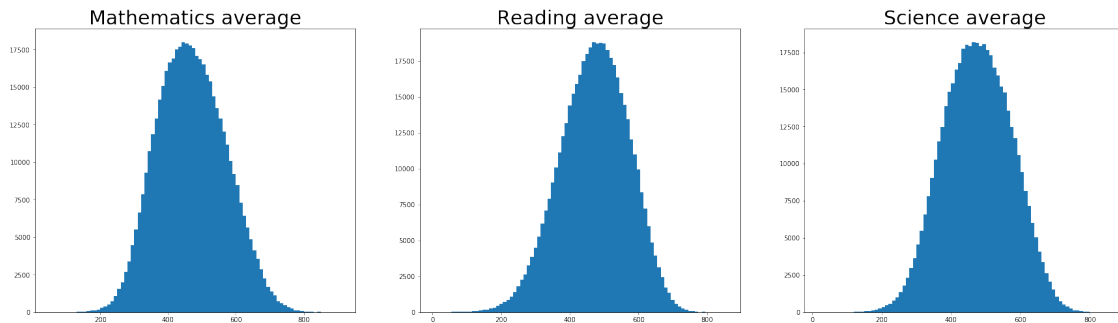
	math	reading	science	total
count	485490.000000	485490.000000	485490.000000	485490.000000
mean	469.651234	472.006964	475.808094	472.488764
std	100.786610	98.863310	97.998470	96.036271
min	54.767080	6.445400	25.158540	77.114593
25%	396.019620	405.044200	405.762800	403.992595
50%	465.734520	475.477980	475.512860	472.046460
75%	540.123060	542.831195	546.381920	541.455700
max	903.107960	849.359740	857.832900	826.592027

```
[123]: math_bins=np.arange(50,df_pisa_clean['math'].max()+10,10)
reading_bins=np.arange(5,df_pisa_clean['reading'].max()+10,10)
science_bins=np.arange(20,df_pisa_clean['science'].max()+10,10)

plt.figure(figsize=(30,8))
plt.subplot(1, 3, 1)
plt.hist(df_pisa_clean['math'],bins=math_bins);
plt.title('Mathematics average',fontsize=30)

plt.subplot(1, 3, 2)
plt.hist(df_pisa_clean['reading'],bins=reading_bins);
plt.title('Reading average',fontsize=30)
```

```
plt.subplot(1, 3, 3)
plt.hist(df_pisa_clean['science'],bins=science_bins);
plt.title('Science average',fontsize=30);
```



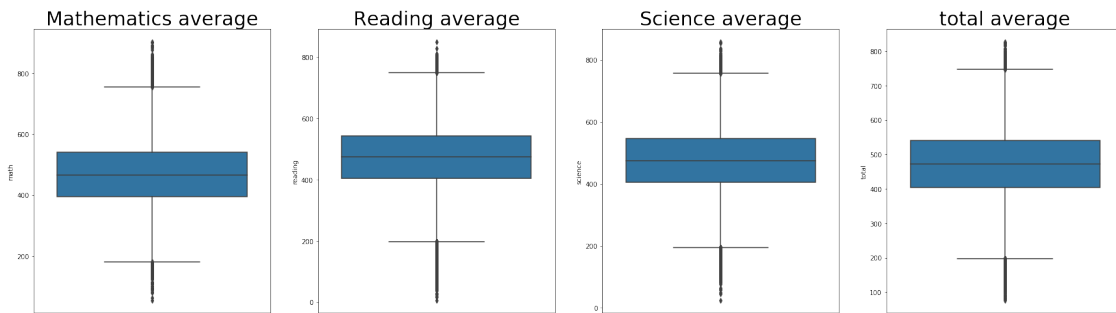
1. All seems like normal ditribution
2. further investigation needs to be done considering country, and gender as well as other categorical variables like internet, mother/father grade, job, etc..

```
[124]: plt.figure(figsize=(30,8))
plt.subplot(1, 4, 1)
sb.boxplot(y=df_pisa_clean['math'])
plt.title('Mathematics average',fontsize=30)

plt.subplot(1, 4, 2)
sb.boxplot(y=df_pisa_clean['reading'])
plt.title('Reading average',fontsize=30)

plt.subplot(1, 4, 3)
sb.boxplot(y=df_pisa_clean['science'])
plt.title('Science average',fontsize=30);

plt.subplot(1, 4, 4)
sb.boxplot(y=df_pisa_clean['total'])
plt.title('total average',fontsize=30);
```

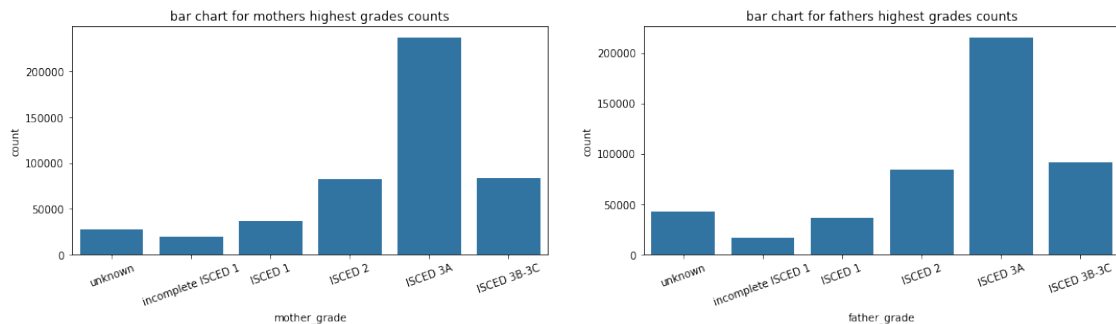


```
[125]: color=sb.color_palette()[0]
plt.figure(figsize=(18,4))

plt.subplot(1, 2, 1)
sb.countplot(data=df_pisa_clean,x='mother_grade',color=color)
plt.title('bar chart for mothers highest grades counts')
plt.xticks(rotation=20)

plt.subplot(1, 2, 2)
sb.countplot(data=df_pisa_clean,x='father_grade',color=color)
plt.title('bar chart for fathers highest grades counts')
plt.xticks(rotation=20)
```

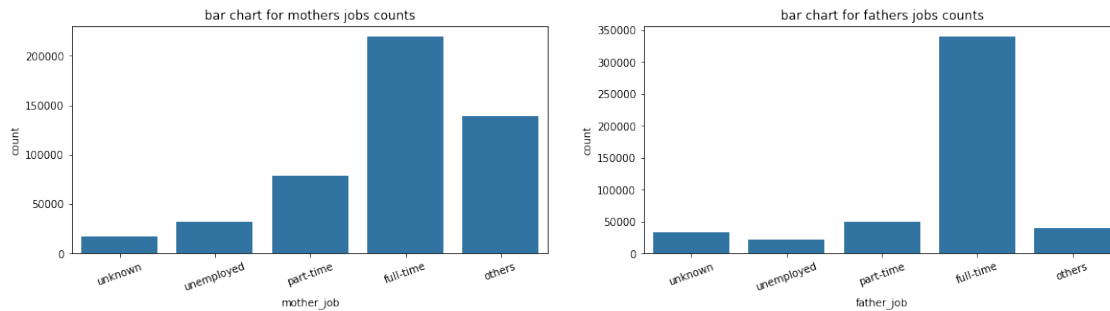
[125]: (array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)



```
[126]: color=sb.color_palette()[0]
plt.figure(figsize=(18,4))

plt.subplot(1, 2, 1)
sb.countplot(data=df_pisa_clean,x='mother_job',color=color);
plt.title('bar chart for mothers jobs counts');
plt.xticks(rotation=20);
```

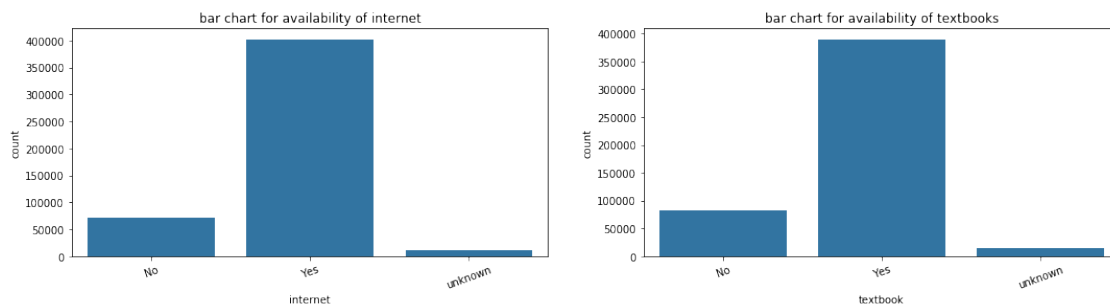
```
plt.subplot(1, 2, 2)
sb.countplot(data=df_pisa_clean,x='father_job',color=color);
plt.title('bar chart for fathers jobs counts');
plt.xticks(rotation=20);
```



```
[127]: color=sb.color_palette()[0]
plt.figure(figsize=(18,4))

plt.subplot(1, 2, 1)
sb.countplot(data=df_pisa_clean,x='internet',color=color);
plt.title('bar chart for availability of internet');
plt.xticks(rotation=20);

plt.subplot(1, 2, 2)
sb.countplot(data=df_pisa_clean,x='textbook',color=color);
plt.title('bar chart for availability of textbooks');
plt.xticks(rotation=20);
```



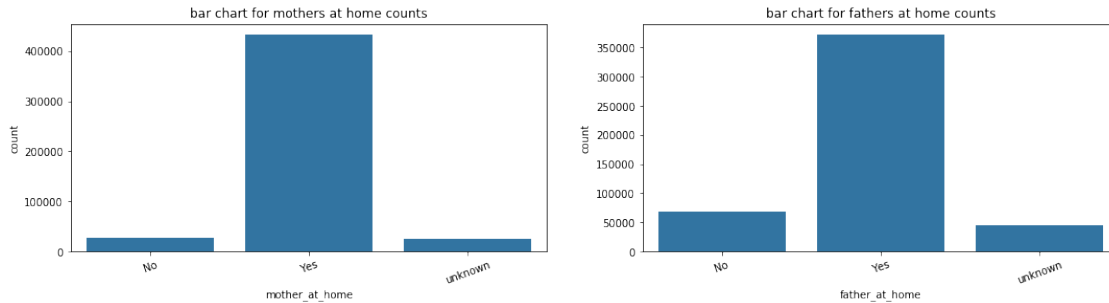
```
[128]: color=sb.color_palette()[0]
plt.figure(figsize=(18,4))

plt.subplot(1, 2, 1)
sb.countplot(data=df_pisa_clean,x='mother_at_home',color=color);
plt.title('bar chart for mothers at home counts');
```



```
plt.xticks(rotation=20);

plt.subplot(1, 2, 2)
sb.countplot(data=df_pisa_clean,x='father_at_home',color=color);
plt.title('bar chart for fathers at home counts');
plt.xticks(rotation=20);
```



1. using boxplot we notice big range of outliers, this is also clear when we used the describe method for the three columns (math, reading, and science)
2. math average seems to have the lowest median
3. conclusion, we need to investigate data using bivariate exploration on country, gender, age, and other variables

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

using the average value along gives normal distribution, but there are many outliers which indicates variations based on other factors

1.7.2 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 used the mean for the three most important features math, reading, science to analyze the data

1.8 Bivariate Exploration

In this section, investigate relationships between pairs of variables in your data. Make sure the variables that you cover here have been introduced in some fashion in the previous section (univariate exploration).

check for math, reading, science average based on countries to find any relation between top performers and countries

```
[180]: df_pisa_countries = df_pisa_clean.
        ↳groupby('country')[['math','reading','science','total']].mean()
```

```
[184]: df_pisa_countries.sort_values(by='total', ascending=False).head(10)
```

```
[184]:
```

	math	reading	science	total
country				
China-Shanghai	611.438933	568.629233	579.556540	586.541569
Hong Kong-China	561.052123	544.521735	554.986433	553.520097
Singapore	568.546974	537.742138	546.822920	551.037344
Korea	553.752034	535.805221	537.831998	542.463084
Japan	535.925248	537.722484	546.413455	540.020395
Chinese Taipei	558.312010	522.185472	522.356935	534.284806
Liechtenstein	538.886608	518.275718	527.598522	528.253616
Estonia	522.340803	518.208090	543.241849	527.930247
Poland	520.522589	520.763584	528.245169	523.177114
Macao-China	538.319791	509.095969	520.690411	522.702057

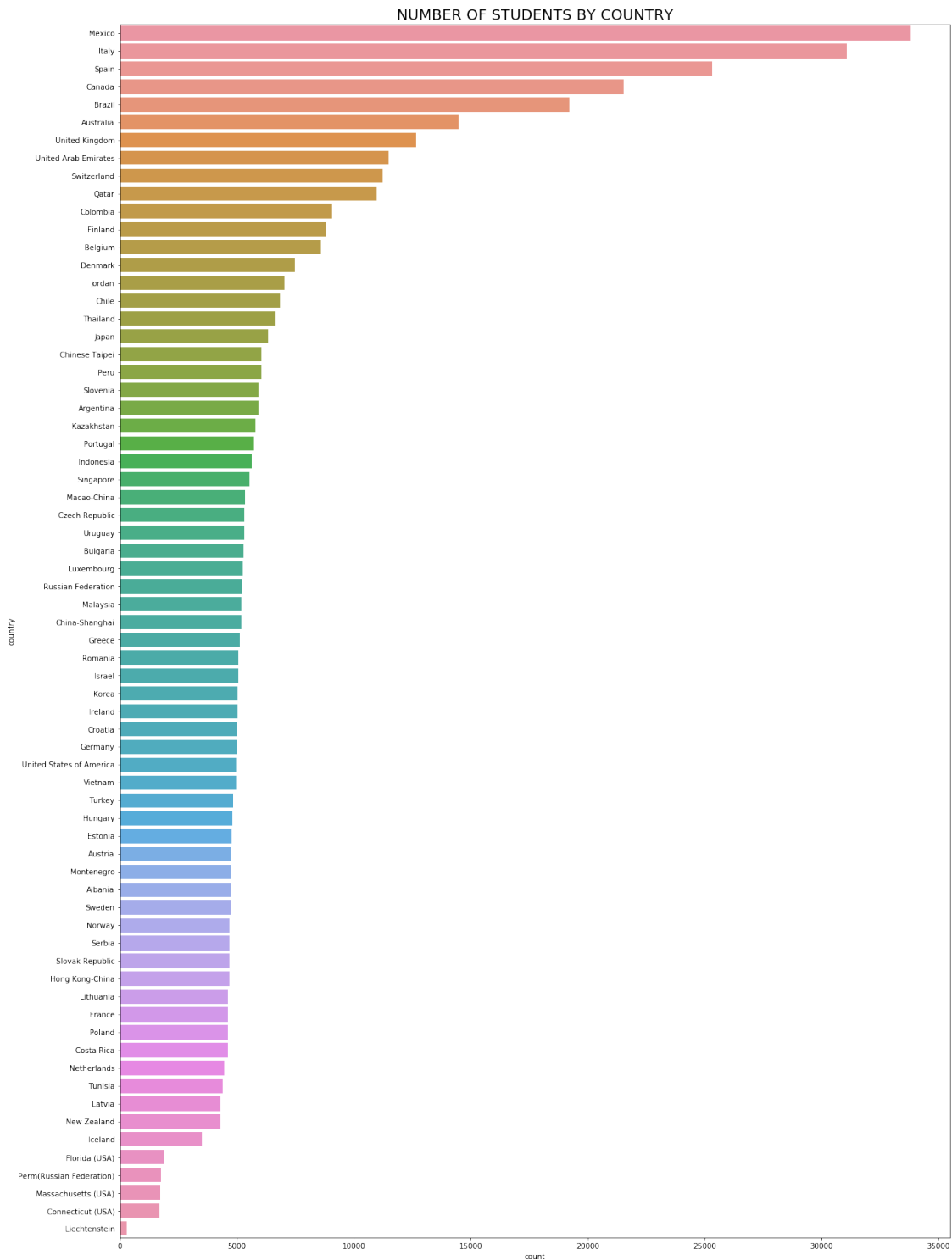
```
[185]: df_pisa_countries.sort_values(by='total', ascending=False).tail(10)
```

```
[185]:
```

	math	reading	science	total
country				
Montenegro	406.728296	419.983824	408.197858	411.636659
Argentina	395.635711	403.596060	410.478404	403.236725
Colombia	385.972409	414.221547	408.862431	403.018796
Albania	394.878912	396.250245	398.916529	396.681895
Tunisia	387.434260	403.614273	397.831316	396.293283
Jordan	382.739077	396.514701	404.795878	394.683219
Brazil	382.547146	400.421704	395.513221	392.827357
Indonesia	375.621968	397.114815	382.744804	385.160529
Qatar	376.339232	387.407142	383.531664	382.426012
Peru	367.859676	384.453116	373.440303	375.251032

1. Around 7 out of 10 top performer countries are in Far east, remaining 3 in Europe
2. low performing countries range from south america
3. more investigation on total average

```
[226]: order= df_pisa_clean['country'].value_counts().index
plt.figure(figsize=[20,30])
sb.countplot(data=df_pisa_clean,y='country',order=order)
plt.title('Number of students per country',size=20);
```



```
[206]: df_pisa_gender = df_pisa_clean.
        ↳groupby('gender')[['math','reading','science','total']].mean().reset_index()
df_pisa_gender
```

```
[206]:
```

	gender	math	reading	science	total
0	Female	464.060962	489.719710	475.348653	476.376442
1	Male	475.349347	453.952526	476.276398	468.526090

```
[386]: 489.719710-453.95252
```

```
[386]: 35.767190000000003
```

```
[383]: df_pisa_gender_melt
```

```
[383]:
```

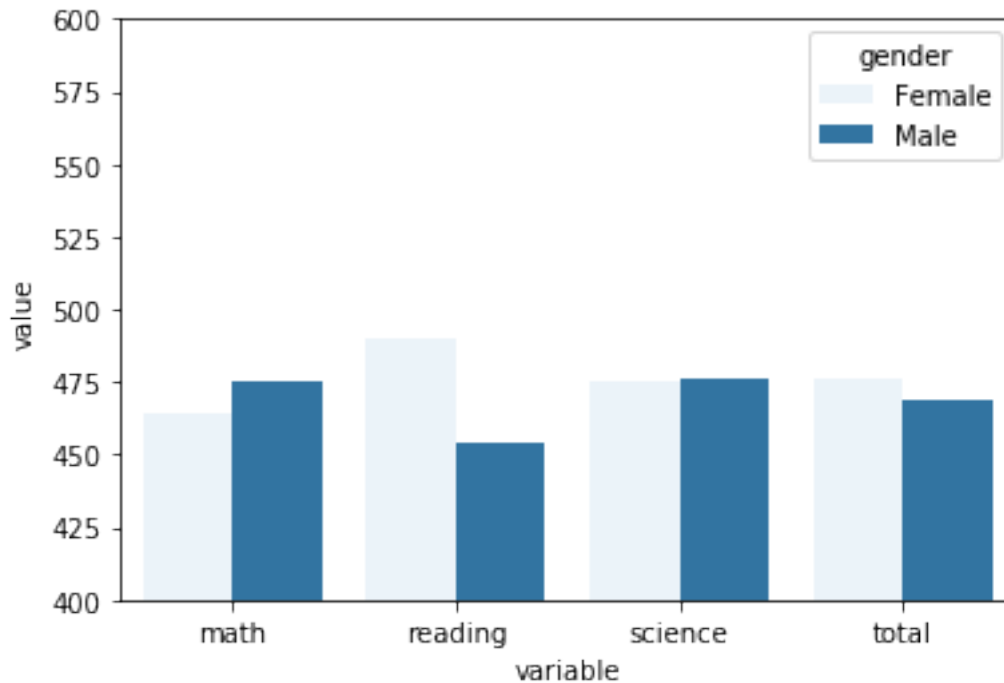
	gender	variable	value
0	Female	math	464.060962
1	Male	math	475.349347
2	Female	reading	489.719710
3	Male	reading	453.952526
4	Female	science	475.348653
5	Male	science	476.276398
6	Female	total	476.376442
7	Male	total	468.526090

```
[382]: df_pisa_gender_melt[(df_pisa_gender_melt['variable']=='reading')&
↳ (df_pisa_gender_melt['variable']=='Female')]
```

```
[382]: Empty DataFrame
Columns: [gender, variable, value]
Index: []
```

```
[208]: sb.barplot(data=df_pisa_gender_melt, x="variable",y="value",hue="gender",
↳ color=sb.color_palette()[0])
plt.ylim(400, 600)
```

```
[208]: (400, 600)
```



```
[205]: df_pisa_internet = df_pisa_clean.
        ↳groupby('internet')[['math','reading','science','total']].mean().
        ↳reset_index()
df_pisa_internet
```

```
[205]:   internet    math    reading    science    total
0      No  400.018756  408.247075  409.686776  405.984202
1     Yes  484.814701  486.403139  490.411081  487.209640
2  unknown  376.823281  370.298152  380.993168  376.038200
```

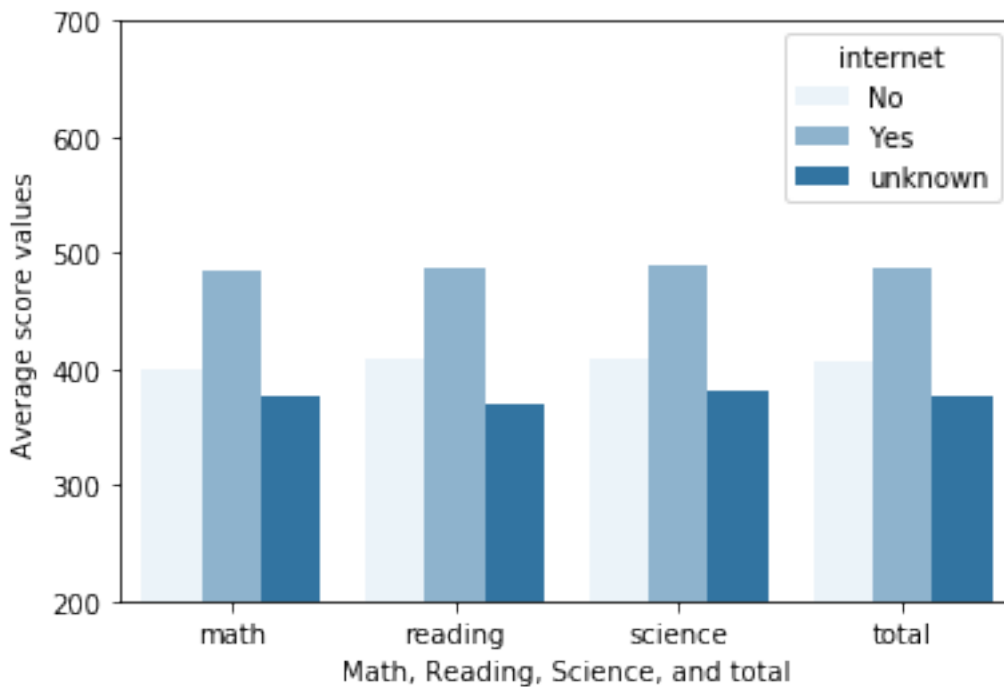
```
[210]: df_pisa_internet_melt = pd.melt(df_pisa_internet, id_vars=['internet'])
df_pisa_internet_melt
```

```
[210]:   internet variable    value
0      No    math  400.018756
1     Yes    math  484.814701
2  unknown    math  376.823281
3      No  reading  408.247075
4     Yes  reading  486.403139
5  unknown  reading  370.298152
6      No  science  409.686776
7     Yes  science  490.411081
8  unknown  science  380.993168
9      No   total  405.984202
```

```
10      Yes      total  487.209640
11  unknown      total  376.038200
```

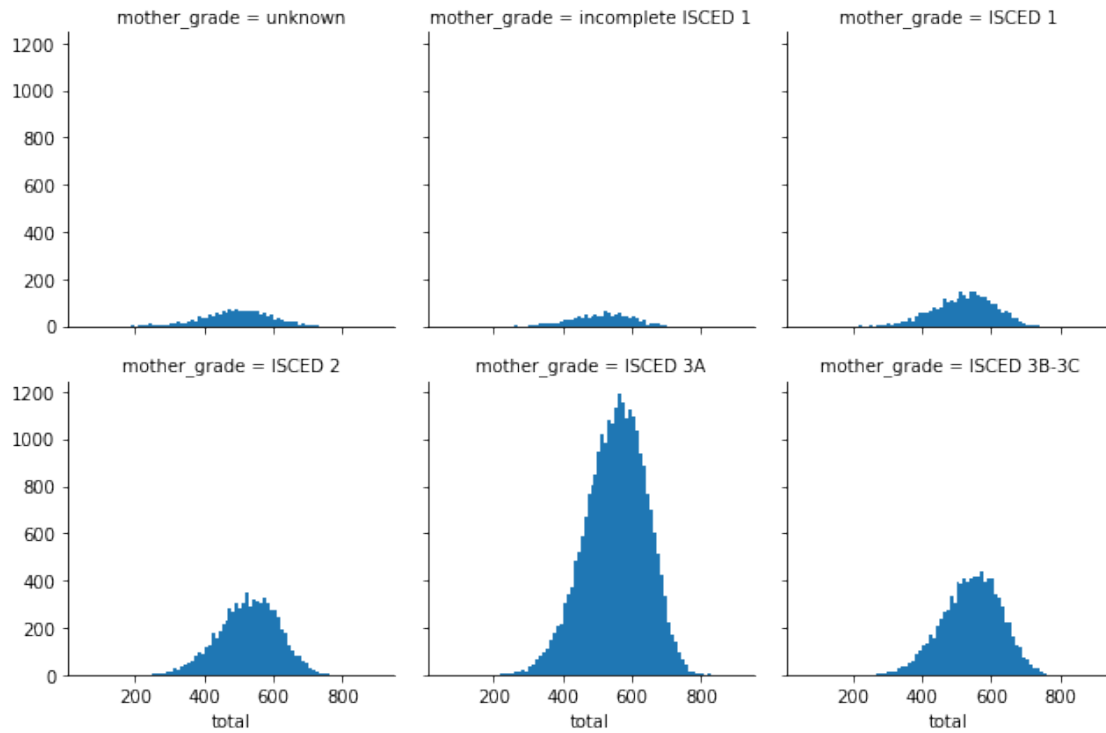
```
[224]: sb.barplot(data=df_pisa_internet_melt,
↳x="variable",y="value",hue="internet",color=sb.color_palette()[0])
plt.ylim(200,700)
plt.xlabel('Math, Reading, Science, and total')
plt.ylabel('Average score values')
```

```
[224]: Text(0, 0.5, 'Average score values')
```



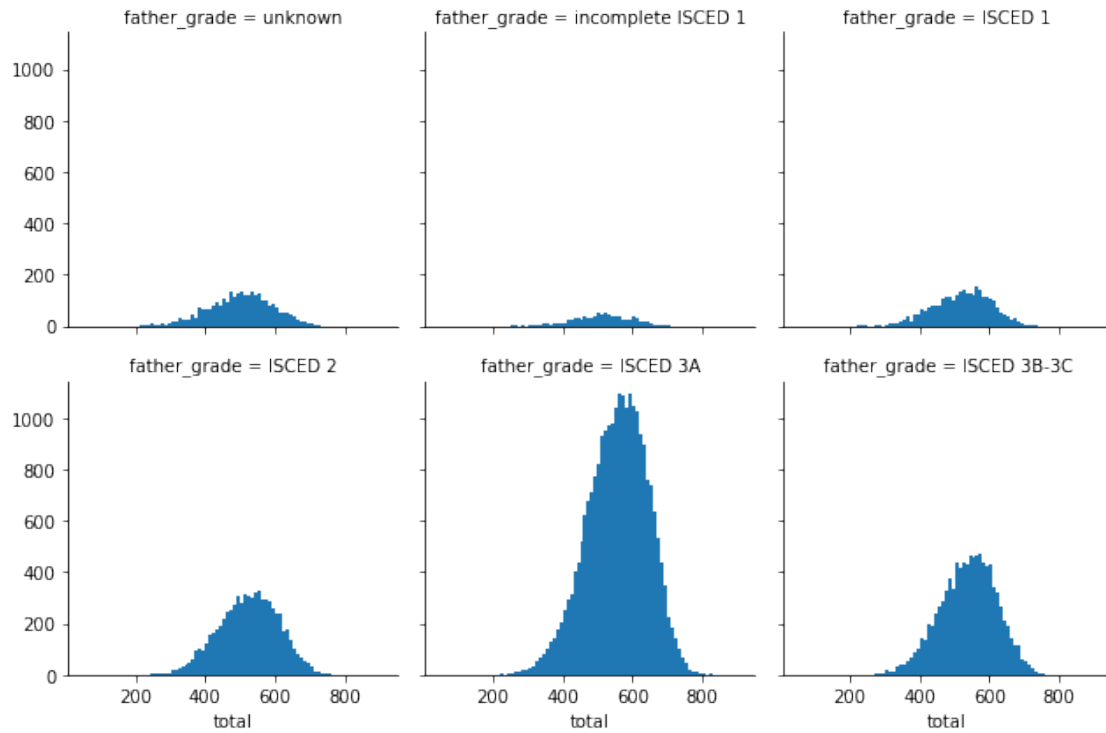
```
[151]: g = sb.FacetGrid(data = df_top_countries, col =
↳'mother_grade',col_wrap=3,sharey=True)
g.map(plt.hist, "total",bins=math_bins)
```

```
[151]: <seaborn.axisgrid.FacetGrid at 0x1f3394b58c8>
```



```
[152]: g = sb.FacetGrid(data = df_top_countries, col = 'father_grade',
↳ 'father_grade', col_wrap=3, sharey=True)
g.map(plt.hist, "total", bins=math_bins)
```

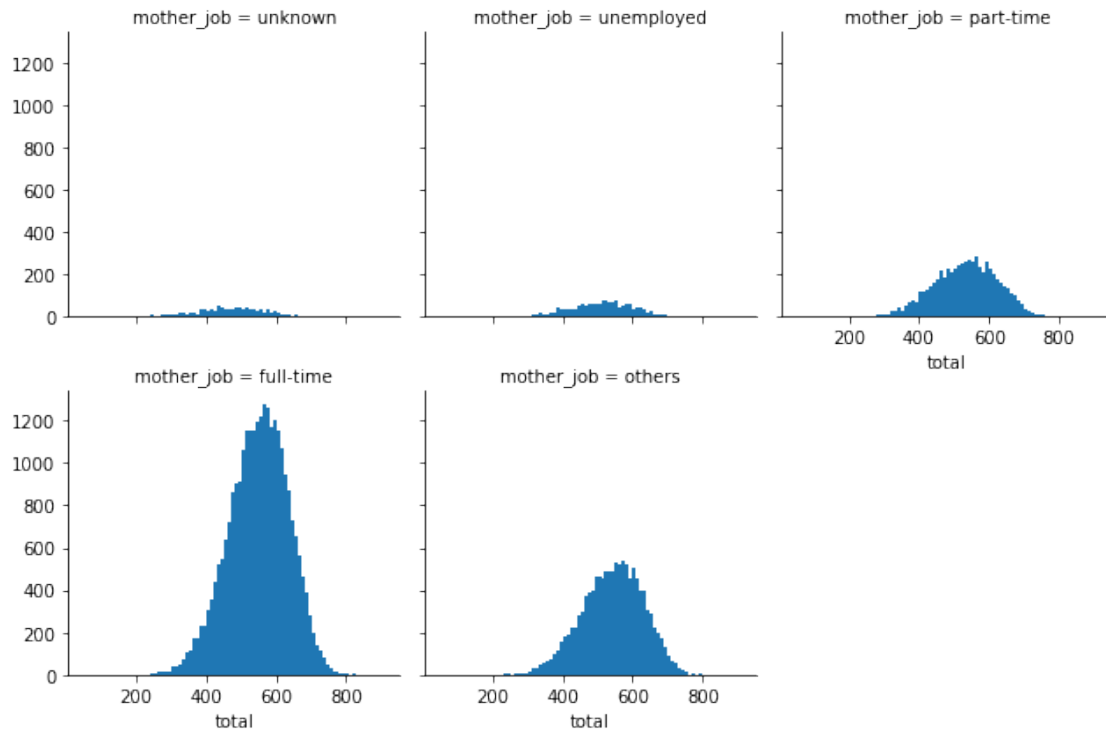
```
[152]: <seaborn.axisgrid.FacetGrid at 0x1f2b67cf0c8>
```



[]:

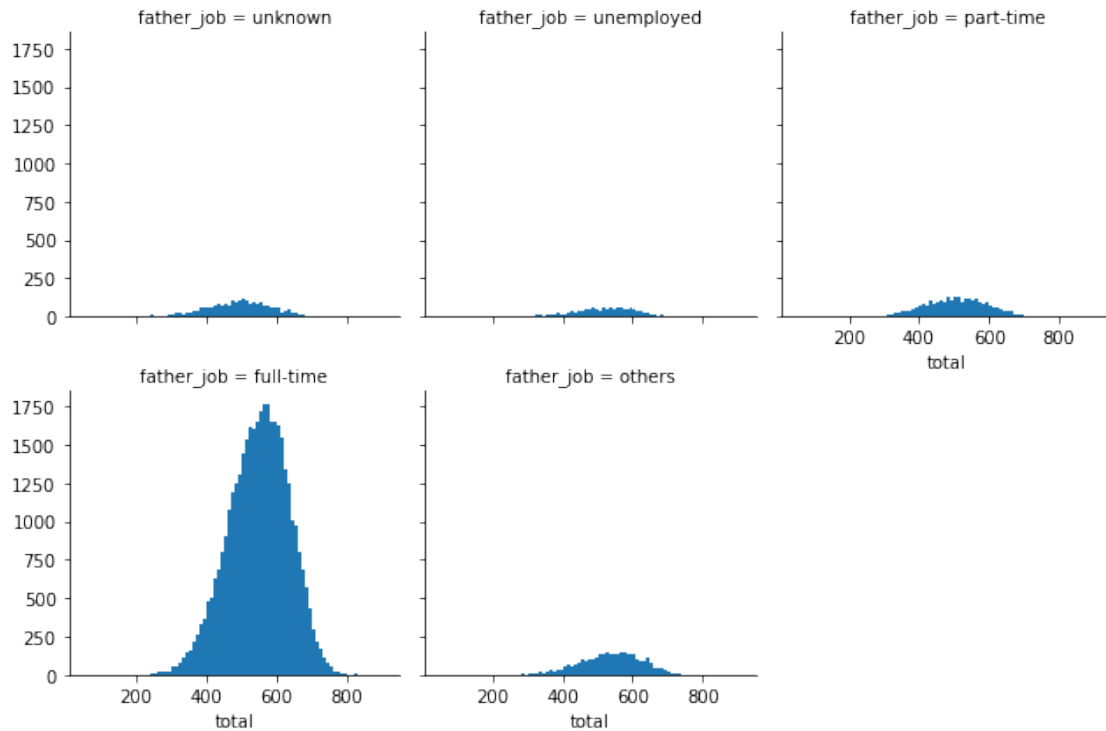
```
[153]: g = sb.FacetGrid(data = df_top_countries, col = 'mother_job', col_wrap=3, sharey=True)
g.map(plt.hist, "total", bins=math_bins)
```

[153]: <seaborn.axisgrid.FacetGrid at 0x1f3343d7348>



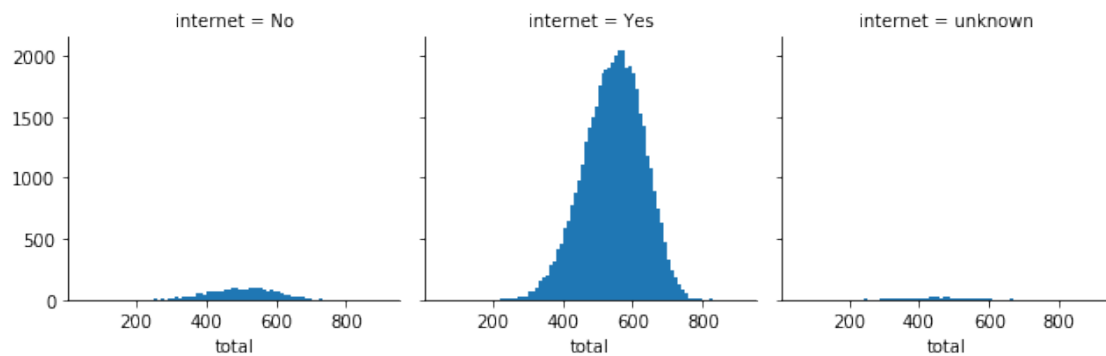
```
[154]: g = sb.FacetGrid(data = df_top_countries, col = 'father_job', col_wrap=3, sharey=True)
g.map(plt.hist, "total", bins=math_bins)
```

```
[154]: <seaborn.axisgrid.FacetGrid at 0x1f338199f88>
```



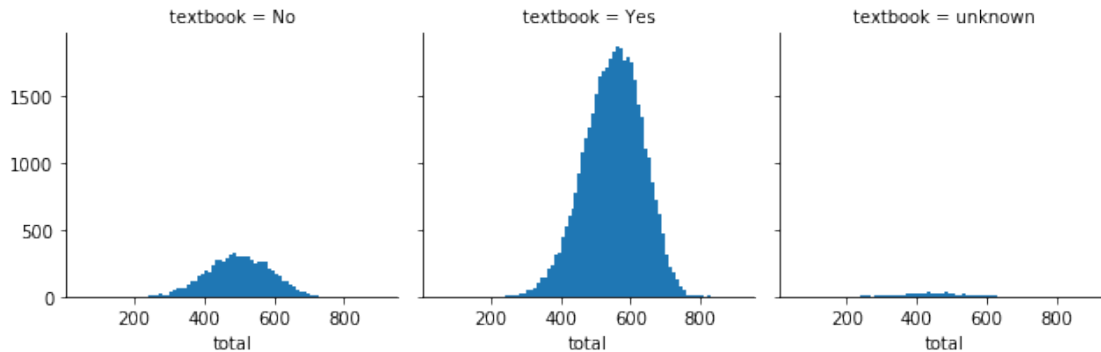
```
[155]: g = sb.FacetGrid(data = df_top_countries, col = "internet",
    ↪ 'internet', col_wrap=3, sharey=True)
g.map(plt.hist, "total", bins=math_bins)
```

```
[155]: <seaborn.axisgrid.FacetGrid at 0x1f3393f5f88>
```



```
[156]: g = sb.FacetGrid(data = df_top_countries, col = "textbook",
    ↪ 'textbook', col_wrap=3, sharey=True)
g.map(plt.hist, "total", bins=math_bins)
```

[156]: <seaborn.axisgrid.FacetGrid at 0x1f339586dc8>



1. Data shows that most of the students have parents (mother / father) that finished 'ISCED level 3A' level
2. same for full time working mom / dads
3. most people have internet, and textbook

1.8.1 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?

1. the internet had a huge impact on the scores compared with student who didn't have access to internet

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

1. Females in general had better scores specially in reading
2. Around 9 out of 10 top performer countries are in Far east, remaining 3 in Europe
3. low performing countries range from south america

1.9 Multivariate Exploration

Create plots of three or more variables to investigate your data even further. Make sure that your investigations are justified, and follow from your work in the previous sections.

```
[238]: df_pisa_clean_copy=df_pisa_clean.copy()
```

```
[240]: df_pisa_clean_copy.drop(columns=['total'], inplace=True)
```

```
[242]: df_pisa_clean_melt = df_pisa_clean_copy.
        ↪melt(id_vars=['country','student_id','gender','mother_at_home','father_at_home','mother_gra
        ↪
        ↪
        ↪'father_job','mother_job','internet','textbook','age'],
```

```
var_name='material',value_name='score')
```

```
[243]: df_pisa_clean_melt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1456470 entries, 0 to 1456469
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   country                1456470 non-null object
1   student_id             1456470 non-null int64
2   gender                 1456470 non-null category
3   mother_at_home         1456470 non-null category
4   father_at_home         1456470 non-null category
5   mother_grade           1456470 non-null category
6   father_grade           1456470 non-null category
7   father_job             1456470 non-null category
8   mother_job             1456470 non-null category
9   internet               1456470 non-null category
10  textbook               1456470 non-null category
11  age                    1456470 non-null float64
12  material               1456470 non-null object
13  score                  1456470 non-null float64
dtypes: category(9), float64(2), int64(1), object(2)
memory usage: 68.1+ MB
```

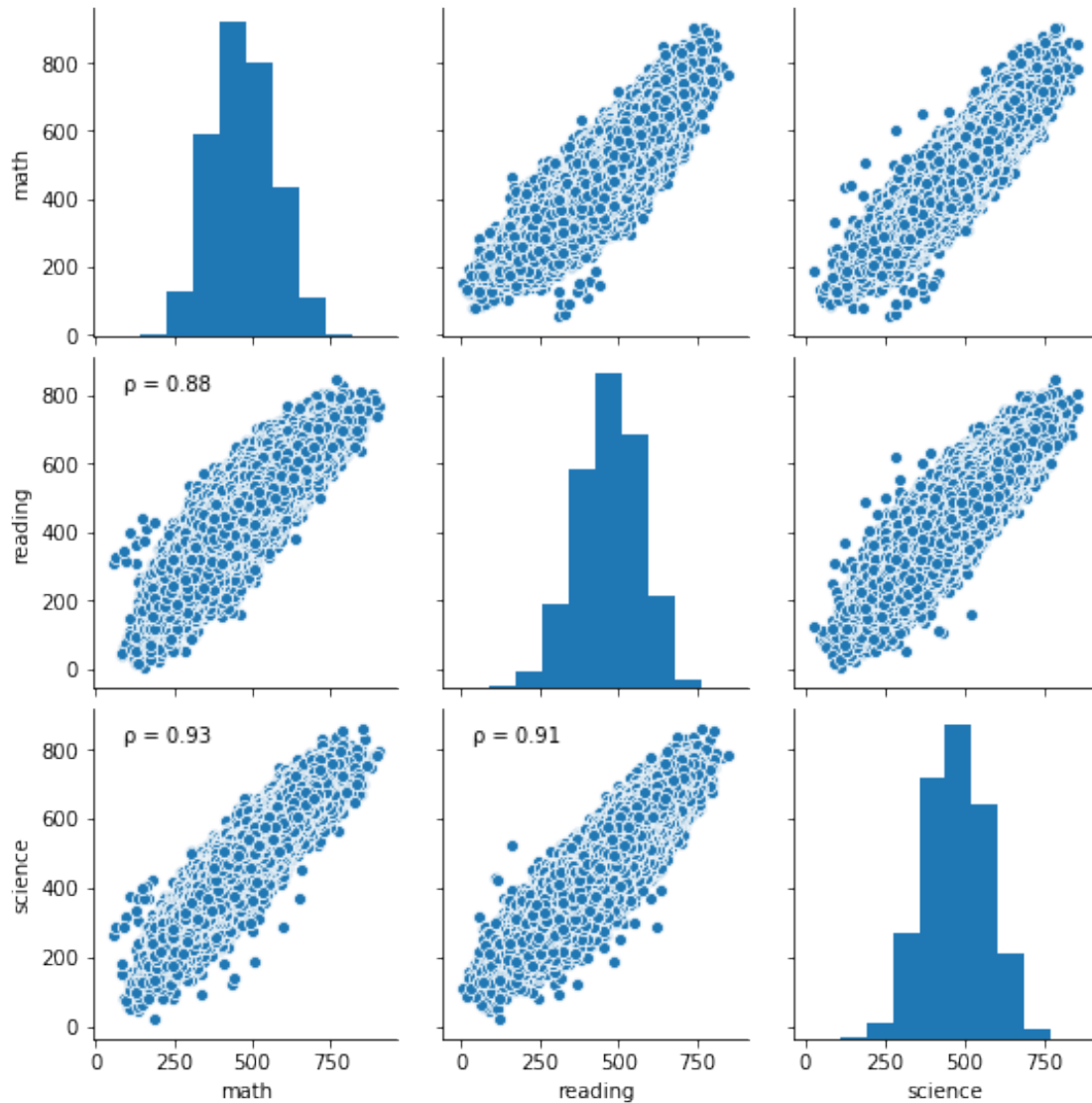
```
[330]: #source: https://stackoverflow.com/questions/30942577/seaborn-correlation-coefficient-on-pairgrid

from scipy.stats import pearsonr

def corrfunc(x,y, ax=None, **kws):
    """Plot the correlation coefficient in the top left hand corner of a plot.
    ↪ """
    r, _ = pearsonr(x, y)
    ax = ax or plt.gca()
    # Unicode for lowercase rho ( )
    rho = '\u03C1'
    ax.annotate(f'{rho} = {r:.2f}', xy=(.1, .9), xycoords=ax.transAxes)

g = sb.pairplot(data = df_pisa_clean, vars=["math", "reading", "science"]);
g.map_lower(corrfunc)
```

```
[330]: <seaborn.axisgrid.PairGrid at 0x1f346541948>
```



we see strong correlation for all plots, highest between math and science.

1.9.1 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?

The dataset used for analysis limited capabilities to make extensive relations for multivariate exploration. Most of the details is already clear in the single / bivariate. The correlation diagram indicates strong relation between scores in math, reading, science and score. Highest between math and science.

1.9.2 Were there any interesting or surprising interactions between features?

Most surprising is the impact on having internet on the students' scores.