PISA_Data_Explanatory_Analysis_II

November 25, 2022

1 Part II - (Explanatory Data Visualization of the 2012 PISA DATA)

1.1 by (Oluwashina Dedenuola)

1.2 Investigation Overview

The overall goal of this presentation is to present key insights into the analysis carried out on the PISA 2012 Dataset.

Four main visualizations—two from the bivariate exploration and two from the multivariate exploration—will be my main focus.

I'll concentrate on the bivariate count plot, correlation plot, and regression plots that illustrated the relationship between two variables and their distribution according to gender. I was curious to find out if students who did well in one subject had an impact in another. The Multivariate relationship between the student scores will then be evaluated.

I'll stick with the exploratory analysis's chosen visualizations.

1.3 Dataset Overview

The dataset has Six Hundred and Thirt Six(636) columns and Four Hundred and Eighty Five Thousand, Four Hundred and Ninety Columns with data types which consists of floats, integers and objects. I selected just 29 columns of interest from the entire dataset. I also removed rows with null values prior to establishing relationships between variables.

```
In [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
        import warnings
        warnings.simplefilter("ignore")
```

```
In [2]: # load in the dataset into a pandas dataframe
        pisa_data = pd.read_csv('pisa2012.csv',encoding='unicode_escape')
```

```
In [3]: pisa_data = pisa_data.copy()
```

```
In [4]: pisa_data = pisa_data[['STIDSTD','AGE','ST03Q02','ST04Q01','ICTRES','INTMAT','OCOD1','OC
        'PV1MATH','PV2MATH','PV3MATH','PV4MATH','PV5MATH','PV1READ','PV2READ','PV3READ', 'PV4REA

In [5]: #IN ORDER TO OBTAIN THE AVERAGE SCORE OF THE DIFFERENT SUBJECTS, I'LL ADD THEIR SCORES T

        pisa_data['Std Maths Score'] = (pisa_data.loc[:, ['PV1MATH', 'PV2MATH','PV3MATH','PV4MAT
        pisa_data['Std Reading Score'] = (pisa_data.loc[:, ['PV1READ', 'PV2READ','PV3READ','PV4R
        pisa_data['Std Science Score'] = (pisa_data.loc[:, ['PV1SCIE', 'PV2SCIE','PV3SCIE','PV4S

In [6]: #I'll like to rename the other columns heading for better understanding of what the colu
        pisa_data.rename(columns = {'ST04Q01':'Gender','ICTRES':'ICT_resources','ST03Q02':'Birth
```

The focus of the PISA project is on students' subjects and scores as well as any potential relationships between the various data in the dataset because the project's goal is to understand how well students have learned and understood their curriculum. Additionally, I am interested in identifying any potential linear relationships between the participants' performance and their gender distribution as determined by their scores.

1.4 (Visualization 1)

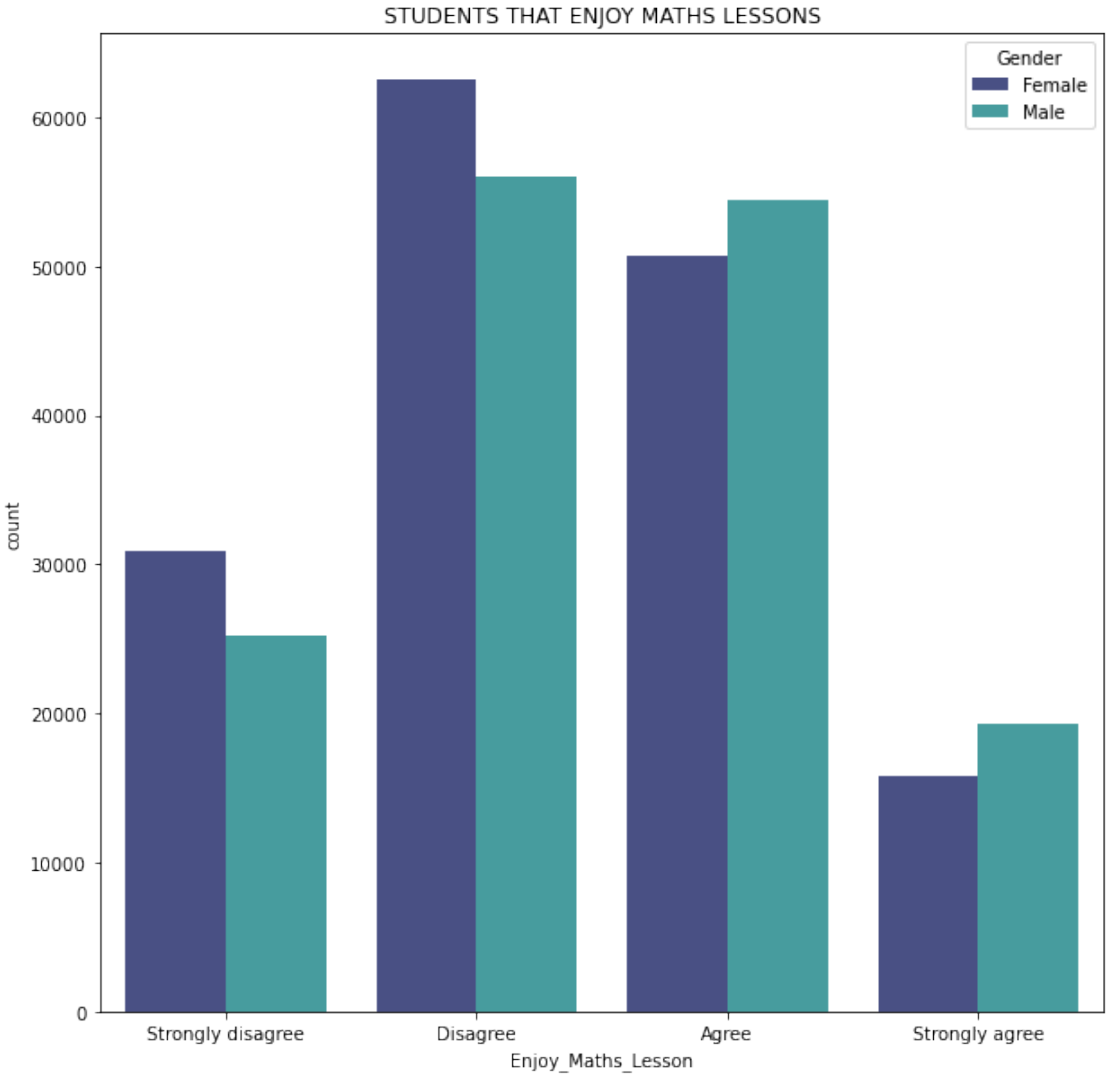
To represent the gender distribution of the students who enjoy math lessons, I will plot a countplot.

In general, the majority of students don't think they enjoy math lessons, and the majority of them are female. More female than male disagree with the statement that they enjoy math.

```
In [7]: O_Maths_dict = {'Enjoy_Maths_Lesson':['Strongly disagree', 'Disagree', 'Agree', 'Strongl


        for maths in O_Maths_dict:
            ordered_math = pd.api.types.CategoricalDtype(ordered = True,
                                                         categories = O_Maths_dict[maths])
            pisa_data[maths] = pisa_data[maths].astype(ordered_math)

In [8]: plt.figure(figsize=(10,10))
        sb.countplot(data = pisa_data, x = "Enjoy_Maths_Lesson", palette="mako", hue = 'Gender')
        plt.title ("STUDENTS THAT ENJOY MATHS LESSONS")
        plt.show ()
```



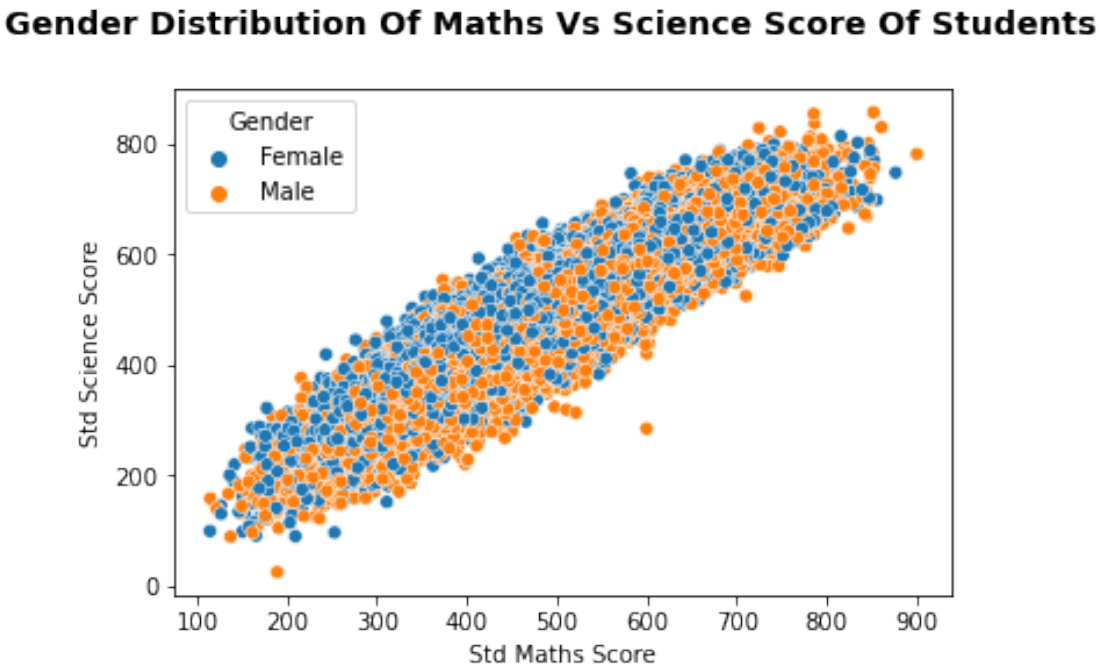

```
In [9]: #Dropping more rows that have null values for uniformity sake.
        pisa_data = pisa_data.dropna(subset=['Enjoy_Reading_Maths','Enjoy_Maths','Interest_in_Ma
```

1.5 (Visualization 2)

To represent the gender distribution of the students and the correlation between their Math and Science Scores, I will create a scatterplot.

The outcome below demonstrates that both genders performed similarly, with reading and math scores showing a strong positive correlation. The scatter plot also reveals that a male received the highest scores in both math and science.

```
In [10]: sb.scatterplot(data=pisa_data, x="Std Maths Score", y="Std Science Score", hue="Gender"
         plt.suptitle("GENDER DISTRIBUTION OF MATHS VS SCIENCE SCORE OF STUDENTS".title(), y = 1
```

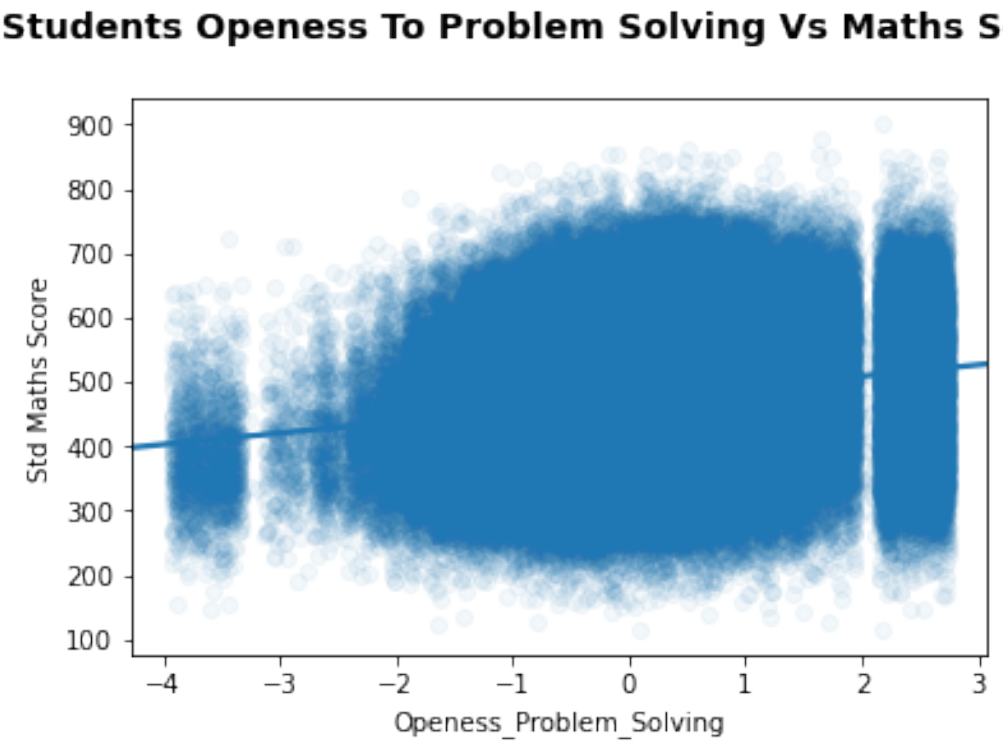


1.6 (Visualization 3)

I'll use a lmplot to illustrate the connection between students' math interest and math test scores.

The regplot suggests a positive weak regression line, which suggests that some students who are open to problem-solving have a tendency to perform very well on their math assessments.

```
In [11]: sb.regplot(data = pisa_data, x = 'Openess_Problem_Solving', y = 'Std Maths Score', trun
         plt.suptitle("STUDENTS OPENESS TO PROBLEM SOLVING VS MATHS SCORE".title(), y = 1, fonts
```

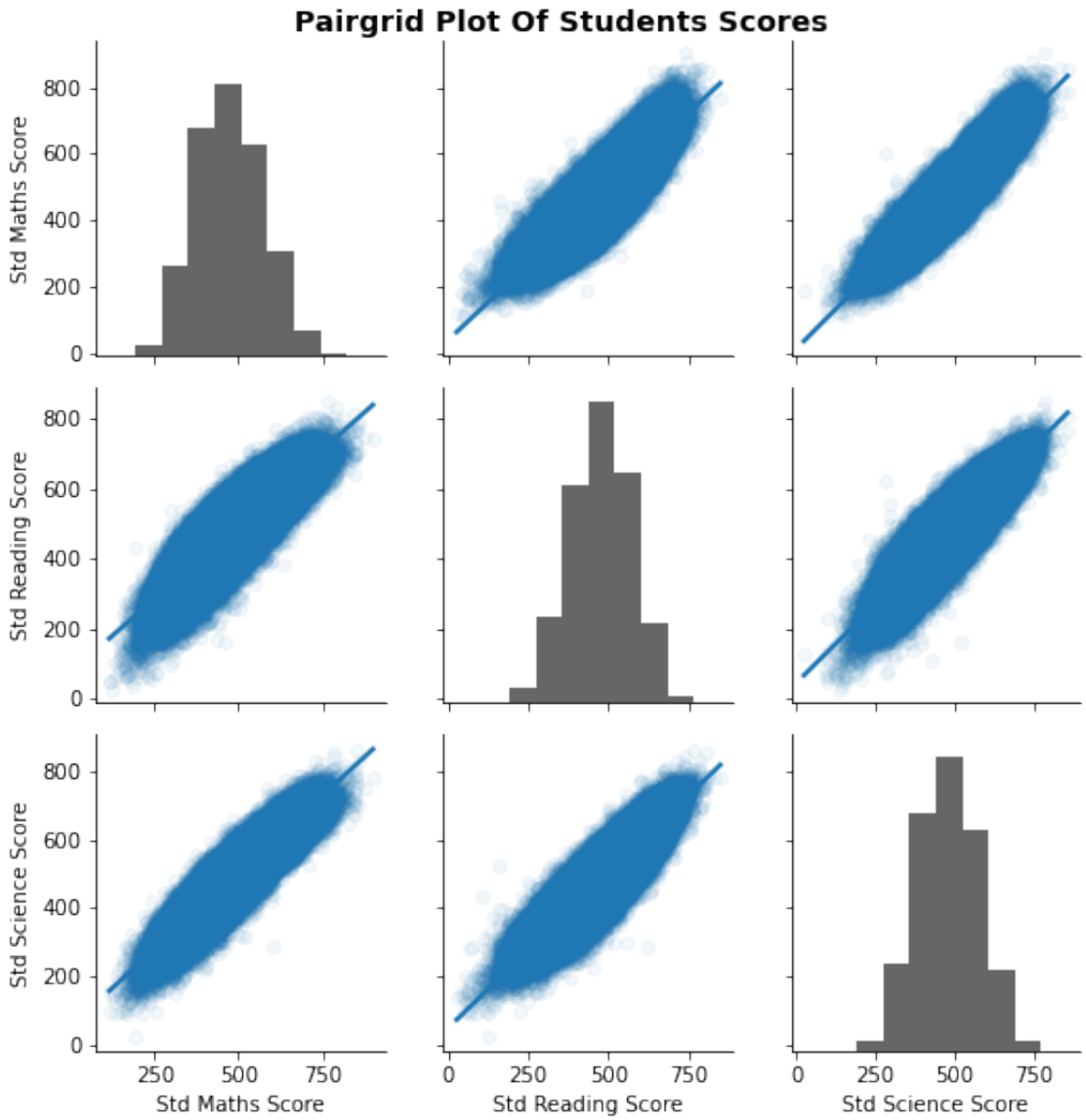


1.7 (Visualization 4)

To express the correlation between the student scores and further substantiate the findings of the bivariate analysis, I will plot a Pairgrid.

The student test scores in math, science, and reading were further validated by the Pairgrid plot, which also expressed the relationship between three quantitative variables. While the histogram shows an 800 peak across all student scores, the scatter plot demonstrates positive correlation across the student scores. A symmetric short-tailed distribution across all student scores is also visible in the histogram.

```
In [12]: g = sb.PairGrid(data = pisa_data, vars = ['Std Maths Score', 'Std Reading Score', 'Std
         g.map_diag(plt.hist, color=".4")
         g.map_offdiag(sb.regplot, x_jitter = 0.3, scatter_kws={'alpha':1/20});
         plt.suptitle("PAIRGRID PLOT OF STUDENTS SCORES".title(), y = 1, fontsize = 14, weight =
```



Generate Slideshow: Once you're ready to generate your slideshow, use the `jupyter nbconvert` command to generate the HTML slide show. . From the terminal or command line, use the following expression.

```
In [ ]: !jupyter nbconvert 'PISA_Data_Explanatory_Analysis_II.ipynb' --to slides --post serve --
```

```
[NbConvertApp] Converting notebook PISA_Data_Explanatory_Analysis_II.ipynb to slides
[NbConvertApp] Writing 573674 bytes to PISA_Data_Explanatory_Analysis_II.slides.html
[NbConvertApp] Redirecting reveal.js requests to https://cdnjs.cloudflare.com/ajax/libs/reveal.j
Serving your slides at http://127.0.0.1:8000/PISA_Data_Explanatory_Analysis_II.slides.html
Use Control-C to stop this server
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: x-www-browser: not found
```

```
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: firefox: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: iceweasel: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: seamonkey: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: mozilla: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: epiphany: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: konqueror: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: chromium-browser: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: google-chrome: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: www-browser: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: links2: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: elinks: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: links: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: lynx: not found
/usr/bin/xdg-open: 778: /usr/bin/xdg-open: w3m: not found
xdg-open: no method available for opening 'http://127.0.0.1:8000/PISA_Data_Explanatory_Analysis_
```

> This should open a tab in your web browser where you can scroll through your presentation. Sub-slides can be accessed by pressing 'down' when viewing its parent slide. Make sure you remove all of the quote-formatted guide notes like this one before you finish your presentation! At last, you can stop the Kernel.