BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SÀI GÒN

TRƯỜNG ĐẠI HỌC SÀI GÒN

Khoa Công nghệ thông tin

BÀI THU HOẠCH

MÔN:

PHÂN TÍCH DỮ LIỆU

Lớp	**:**	**DCT121C5**
Giảng Viên	**:**	**PGS. TS. Nguyễn Tuấn Đăng**
Sinh viên thực hiện (MSSV)	**:**	**Vũ Huy Hoàng(3121411079)**
		Nguyễn Minh Trí (3121411212)
		Nguyễn Hoàng Tiến (3121411206)
Năm học	**:**	**2023-20224 / HK2**

Thành phố Hồ Chí Minh, tháng 12 năm 2023.

LỜI MỞ ĐẦU

Chào thầy và các bạn!

Trong bối cảnh thời đại số hóa ngày càng phát triển, dữ liệu đã trở thành một phần không thể thiếu trong mọi lĩnh vực, từ kinh doanh, khoa học đến giáo dục và y tế. Việc phân tích và hiểu biết dữ liệu giúp chúng ta đưa ra quyết định chính xác hơn, tối ưu hóa quy trình và đạt được hiệu suất tốt nhất trong công việc.

Trong bài thu hoạch này, chúng ta sẽ cùng nhau khám phá và phân tích một tập dữ liệu thực tế để trả lời một số câu hỏi quan trọng. Chúng ta sẽ sử dụng các công cụ và kỹ thuật phân tích dữ liệu như biểu đồ hộp, heatmap và pairplot để hiểu rõ hơn về mối quan hệ giữa các biến số và tìm ra các mẫu phân bố dữ liệu.

Mục tiêu của bài thu hoạch này là giúp quý vị và các bạn:

1. Hiểu rõ hơn về quá trình phân tích dữ liệu và các công cụ thống kê cơ bản.
2. Áp dụng kiến thức để phân tích một tập dữ liệu thực tế và đưa ra những nhận định, phán đoán hợp lý.
3. Trang bị cho mình những kỹ năng và kiến thức cần thiết để đối mặt và xử lý các tập dữ liệu phức tạp trong tương lai.

Chúng ta sẽ bắt đầu với việc tải và khám phá tập dữ liệu, sau đó tiến hành phân tích và rút ra các kết luận quan trọng. Hãy cùng nhau khám phá và học hỏi từ bài thu hoạch này!

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Khoa Công nghệ thông tin, Trường đại học Sài Gòn đã tạo điều kiện thuận lợi cho chúng em học tập và hoàn thành báo cáo thu hoạch môn học này. Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Tuấn Đăng đã dày công truyền đạt kiến thức cho chúng em.

Chúng em đã cố gắng vận dụng những kiến thức đã học được trong học kỳ qua để hoàn thành đồ án này. Nhưng do kiến thức hạn chế và không có nhiều kinh nghiệm thực tiễn nên khó tránh khỏi những thiếu sót trong quá trình làm bài và trình bày. Chúng em rất mong nhận được sự góp ý của thầy để đồ án môn học của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

NHẬN XÉT, ĐÁNH GIÁ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:.....(Bằng chữ: .....)

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ

CHƯƠNG I:GIỚI THIỆU BÀI THU HOẠCH

3. Giới thiệu về Data Visualization

Data visualization là quá trình biểu diễn dữ liệu và thông tin bằng các đồ họa để hiểu và truyền đạt ý nghĩa của chúng một cách hiệu quả hơn. Điều này thường được thực hiện thông qua việc sử dụng biểu đồ, bản đồ, đồ thị và các phương tiện trực quan khác để biểu hiện mối quan hệ, xu hướng và mẫu số học trong dữ liệu.

Mục tiêu chính của data visualization là giúp con người hiểu và phân tích dữ liệu một cách nhanh chóng và dễ dàng hơn, từ đó có thể đưa ra những quyết định thông minh và dự đoán xu hướng trong tương lai. Điều này đặc biệt quan trọng trong thời đại số hóa ngày nay khi lượng dữ liệu sản sinh ra liên tục tăng lên.

Công cụ và kỹ thuật data visualization đa dạng, bao gồm các phần mềm chuyên dụng như Tableau, Power BI, matplotlib, seaborn, ggplot2, và nhiều công nghệ khác. Các biểu đồ phổ biến bao gồm biểu đồ cột, biểu đồ đường, biểu đồ tròn, bản đồ choropleth, scatter plot, histogram, và nhiều loại khác.

Data visualization không chỉ là công cụ hữu ích trong lĩnh vực khoa học dữ liệu và phân tích dữ liệu, mà còn có ứng dụng rộng rãi trong các lĩnh vực như kinh doanh, y tế, marketing, tài chính, và chính trị. Bằng cách sử dụng data visualization, người dùng có thể khám phá, hiểu và chia sẻ thông tin một cách sâu sắc và hấp dẫn hơn.

4. Giới thiệu về dữ liệu

Students data là tập dữ liệu chứa thông tin về hiệu suất học tập của học sinh trung học môn toán, bao gồm các điểm số và thông tin dân số học. Dữ liệu được thu thập từ ba trường trung học tại Hoa Kỳ.

Dữ liệu bao gồm có các cột sau:

• **Gender**: Giới tính của học sinh (nam/nữ)

• **Race/ethnicity**: Dân tộc hoặc dân tộc của học sinh (Á, Châu Phi - Mỹ, Hispanic, v.v.)

• **Parental level of education**: Trình độ giáo dục cao nhất đạt được bởi phụ huynh hoặc người giám hộ của học sinh

• **Lunch**: Học sinh có nhận bữa trưa miễn phí hoặc giảm giá không (có/không)

• **Test preparation course**: Học sinh đã hoàn thành khóa học chuẩn bị thi không (có/không)

• **Math score**: Điểm số của học sinh trên bài kiểm tra toán chuẩn hóa

• **Reading score**: Điểm số của học sinh trên bài kiểm tra đọc chuẩn hóa

• **Writing score**: Điểm số của học sinh trên bài kiểm tra viết chuẩn hóa

Tập dữ liệu này có thể được sử dụng cho các câu hỏi nghiên cứu liên quan đến giáo dục, như khảo sát ảnh hưởng của trình độ giáo dục của phụ huynh hoặc khóa học chuẩn bị thi đến hiệu suất học tập của học sinh. Nó cũng có thể được sử dụng để phát triển các mô hình học máy để dự đoán hiệu suất học tập của học sinh dựa trên các yếu tố dân số học và khác.

CHƯƠNG II:GIỚI THIỆU DỮ LIỆU

Phân tích tổng quan

```
import findspark

findspark.init()
import pyspark
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from pyspark.sql import *
sns.set(color_codes=True)
path="C:\\Users\\TIEN NGUYEN\\Downloads\\students.csv"
data = pd.read_csv(path)
data.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Hình 1.1 : Top 5 dòng đầu của dữ liệu file students.csv

```
data.info()
```

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

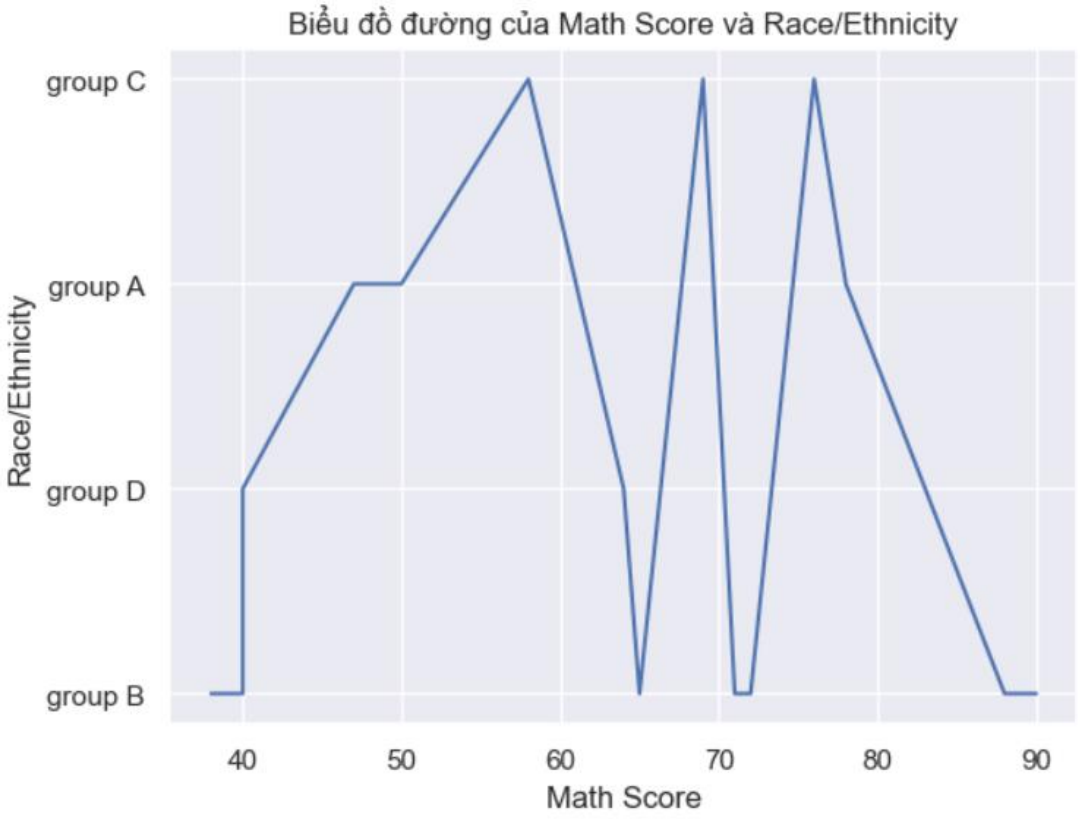
Hình 1.2 : Các thuộc tính của file student.csv

```
data.shape
```

```
: (1000, 8)
```

1. Line plot

```
# Giới hạn dữ liệu
limited_data = data.head(15)
sorted_data = limited_data.sort_values(by='math score')

# Tạo dữ liệu cho trục x và trục y từ dữ liệu đã được sắp xếp
race_ethnicity = sorted_data['race/ethnicity']
math_score = sorted_data['math score']

plt.plot(math_score,race_ethnicity,label='linear')  # Plot data from column_a on x-axis and column_b on y-axis
plt.title('Title of Plot')  # Add a title
plt.xlabel('Math Score')  # Add label for x-axis
plt.ylabel('Race/Ethnicity')  # Add label for y-axis
plt.show()  # Show the plot
```

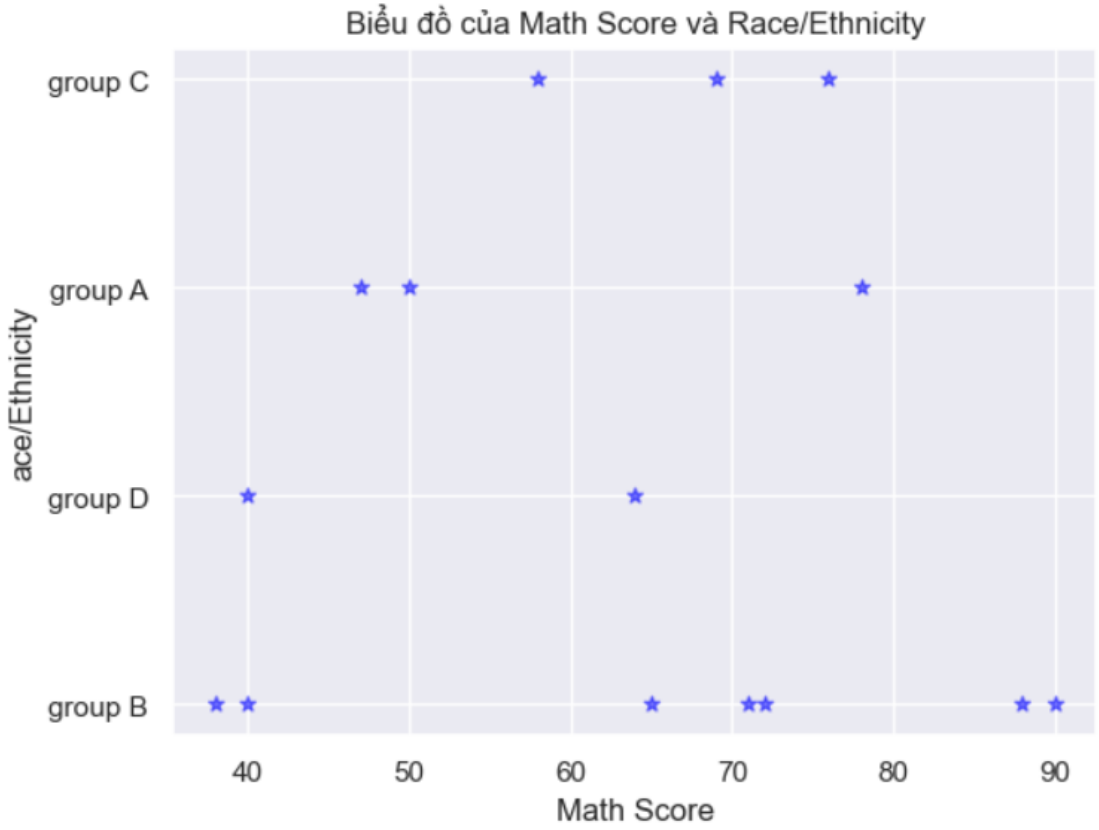


Hình 1.3 : Line Plot của Math Score và Race/Ethnicity

*Đặc diểm của hàm plot()
- *math_score* : là giá trị trên trục x
- *race_ethnicity* : là giá trị trên trục y
- *label :* nhãn dán của đường vẽ

2. Scatter plot

```
plt.scatter(math_score,race_ethnicity,c='blue', marker='*',alpha=0.5)  # Plot data from column_a on x-axis and column_b on y-axis
plt.title('Title of Plot')  # Add a title
plt.xlabel('X-axis Label')  # Add label for x-axis
plt.ylabel('Y-axis Label')  # Add label for y-axis
plt.show()  # Show the plot
```



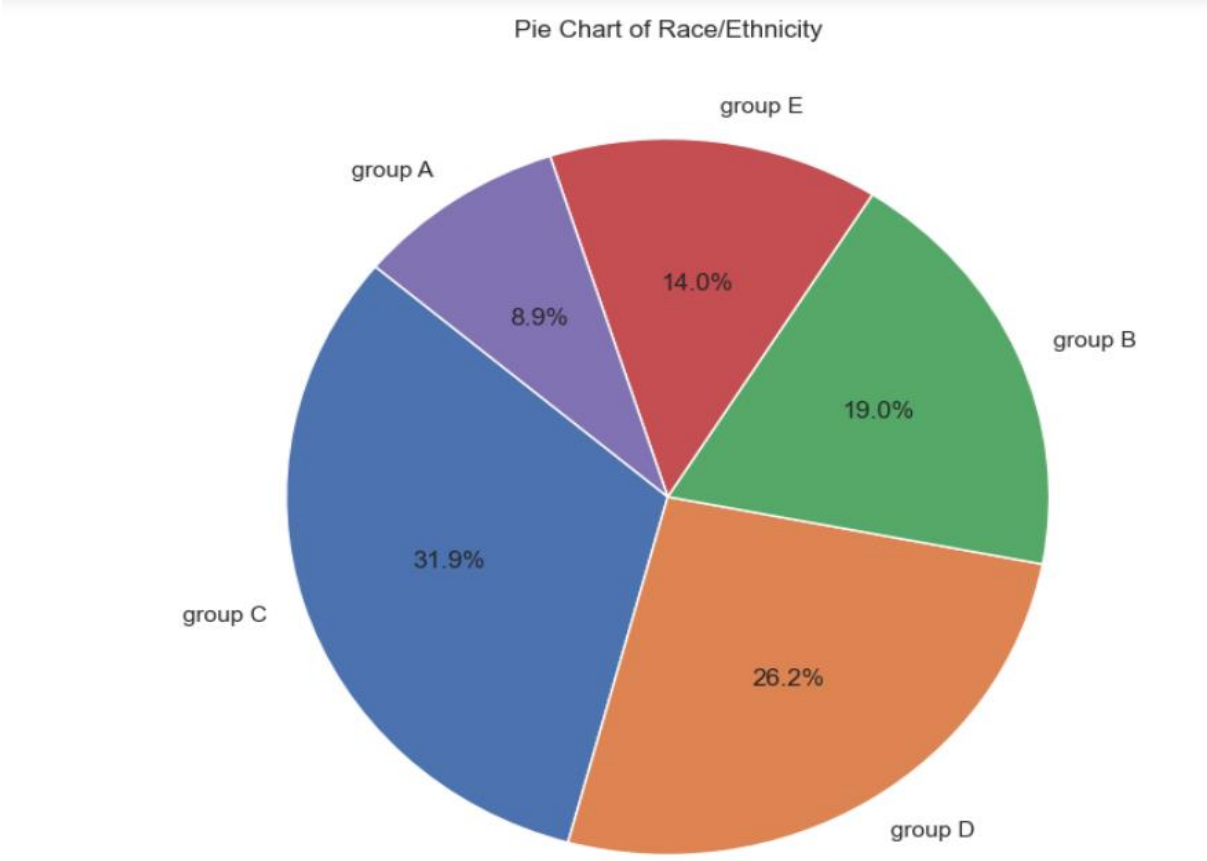
Hình 1.4 : Scatter Plot của Math Score và Race/Ethnicity

*Đặc điểm của hàm Scatter()

- *math_score* : là giá trị trên trục x
- *race_ethnicity* : là giá trị trên trục y
- *c ='blue'* : là màu của kí tự
- *marker='*'* : là kí tự
- *alpha=**0.5*** **:** Độ mờ của đường vẽ

3. Pie plot

```
ethnicity_counts = data['race/ethnicity'].value_counts()
# Lấy các nhãn và kích thước tương ứng
labels = ethnicity_counts.index
sizes = ethnicity_counts.values
# Tạo biểu đồ pie
plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
# Đặt tiêu đề
plt.title('Pie Chart of Race/Ethnicity')
# Hiển thị biểu đồ
plt.show()
```



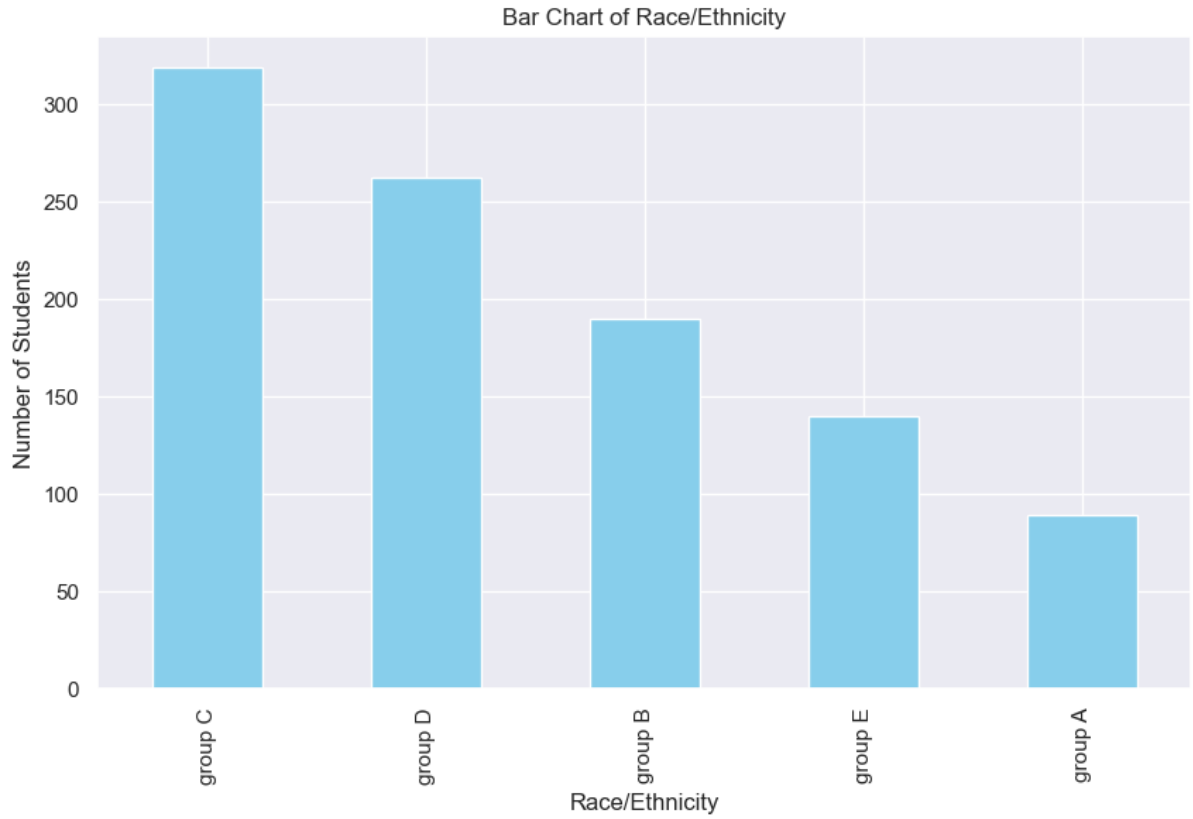
Hình 1.5 : Pie plot của Race/Ethnicity

*Đặc điểm của hàm pie()

- *sizes* : lấy các số lượng trong ethnicity_counts
- *labels=labels* : lấy các nhãn trong ethnicity_counts
- *autopct='%1.1f%%* : chỉ số thập phân thêm dấu %
- *startangle=**140*** **:** bắt đầu vẽ ở góc 140 độ

4. Bar plot

```
plt.figure(figsize=(10, 6))
ethnicity_counts.plot(kind='bar', color='skyblue')

# Đặt tiêu đề và nhãn trục
plt.title('Bar Chart of Race/Ethnicity')
plt.xlabel('Race/Ethnicity')
plt.ylabel('Number of Students')

# Hiển thị biểu đồ
plt.show()
```



Hình 1.6 : Bar plot của Race/Ethnicity

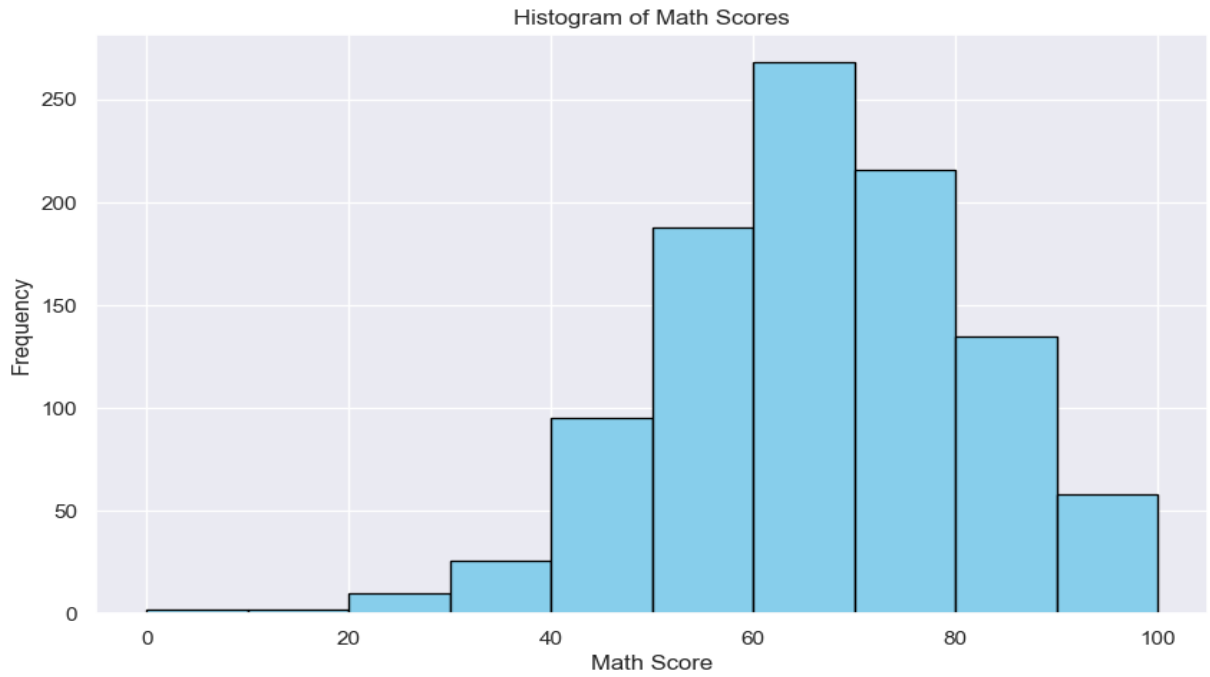
*Đặc điểm của hàm .plot(kind='bar', color='skyblue')

- *kind = 'bar'* : vẽ biểu độ cột bar chart

- *color='skyblue'* : màu là 'skublue'

5. Histogram plot

```
math_scores = data['math score']

plt.figure(figsize=(10, 6))
plt.hist(math_scores, bins=10, color='skyblue', edgecolor='black')

# Đặt tiêu đề và nhãn trục
plt.title('Histogram of Math Scores')
plt.xlabel('Math Score')
plt.ylabel('Frequency')

# Hiển thị biểu đồ
plt.show()
```



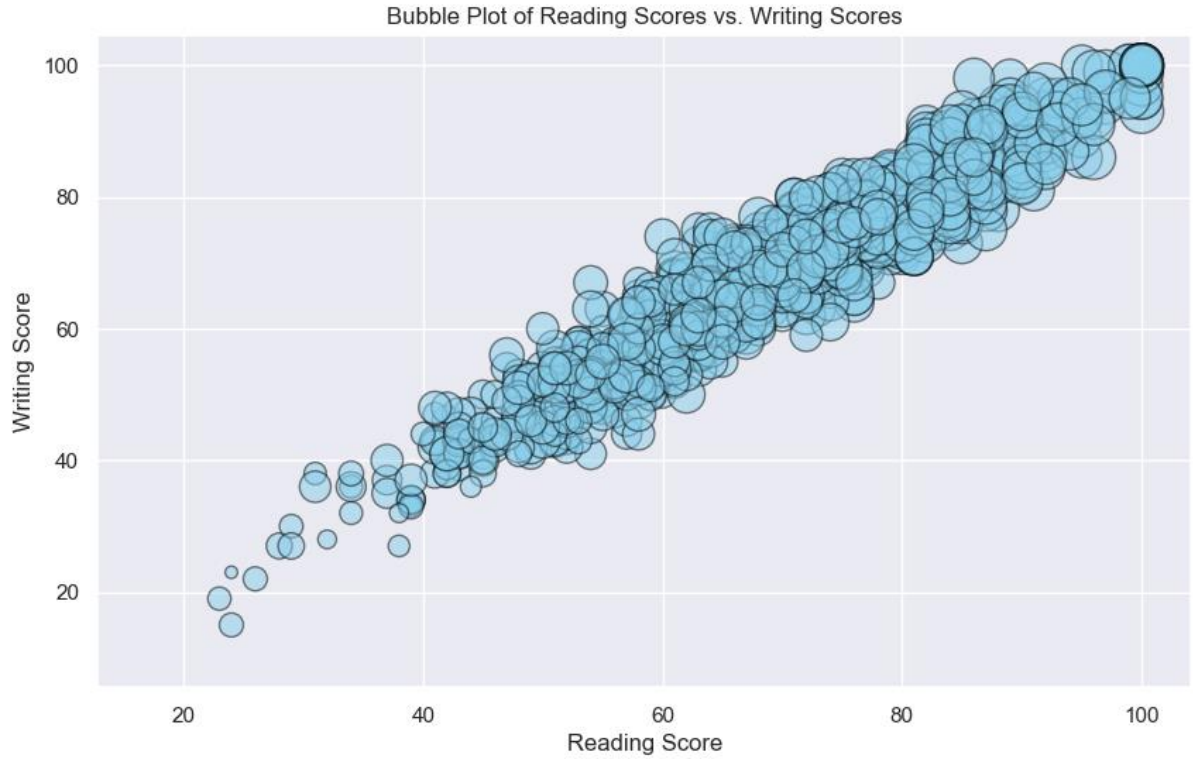
Hình 1.7 : Histogram plot của Math Scores

*Đặc điểm của hist()

- *math_scores* : lấy dữ liệu từ cột data['math score']
- *color='skyblue'* : màu là skyblue
- *edgecolor='black'* : đường viền là màu black
- *bins=**10*** **:** vẽ thánh 10 cột

6. Bubble plot

```
reading_scores = data['reading score']
writing_scores = data['writing score']

# Tính toán kích thước bong bóng dựa trên điểm math
sizes = math_scores * 5  # Để tạo sự khác biệt rõ ràng giữa các bong bóng, bạn có thể nhân điểm math với một hằng số
# Vẽ biểu đồ bubble plot
plt.figure(figsize=(10, 6))
plt.scatter(reading_scores, writing_scores, s=sizes, alpha=0.5, c='skyblue', edgecolor='black')
# Đặt tiêu đề và nhãn trục
plt.title('Bubble Plot of Reading Scores vs. Writing Scores')
plt.xlabel('Reading Score')
plt.ylabel('Writing Score')
# Hiển thị biểu đồ
plt.show()
```



Hình 1.8 : Bubble plot của Reading Scores và Writing Scores

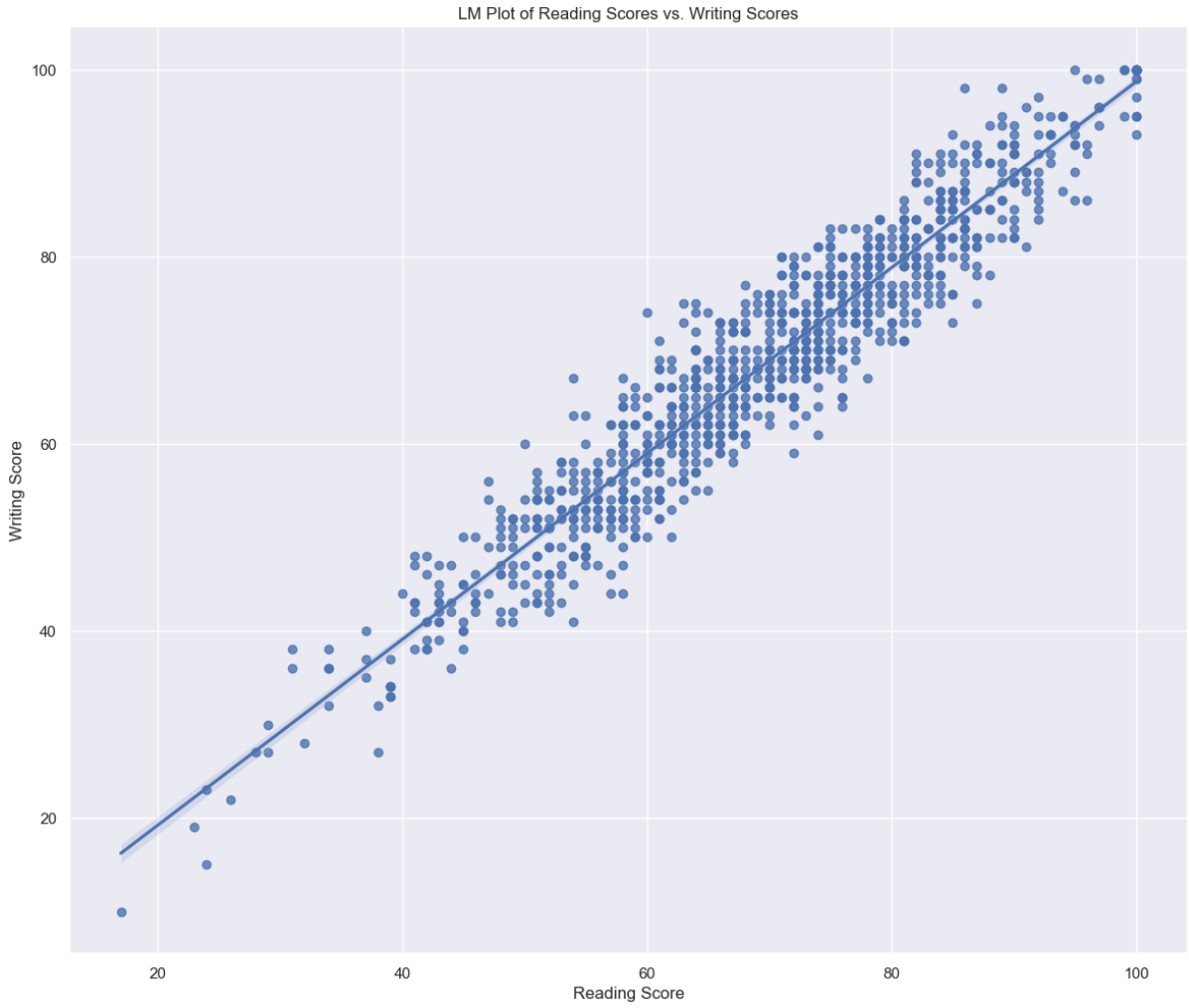
*Đặc điểm của hàm Scatter()

- reading_scores: là giá trị trên trục x, lấy từ cột data['reading score']
- writing_scores: là giá trị trên trục y, lấy từ cột data['writing score']

- s=sizes: kích thước của bong bóng
- *color='skyblue'* : màu là skyblue
- *edgecolor='black'* : đường viền là màu black

7. Lm plots

```
sns.lmplot(x='reading score', y='writing score', data=data, height=10, aspect=1.2)

# Đặt tiêu đề và nhãn trục
plt.title('LM Plot of Reading Scores vs. Writing Scores')
plt.xlabel('Reading Score')
plt.ylabel('Writing Score')

# Hiển thị biểu đồ
plt.show()
```



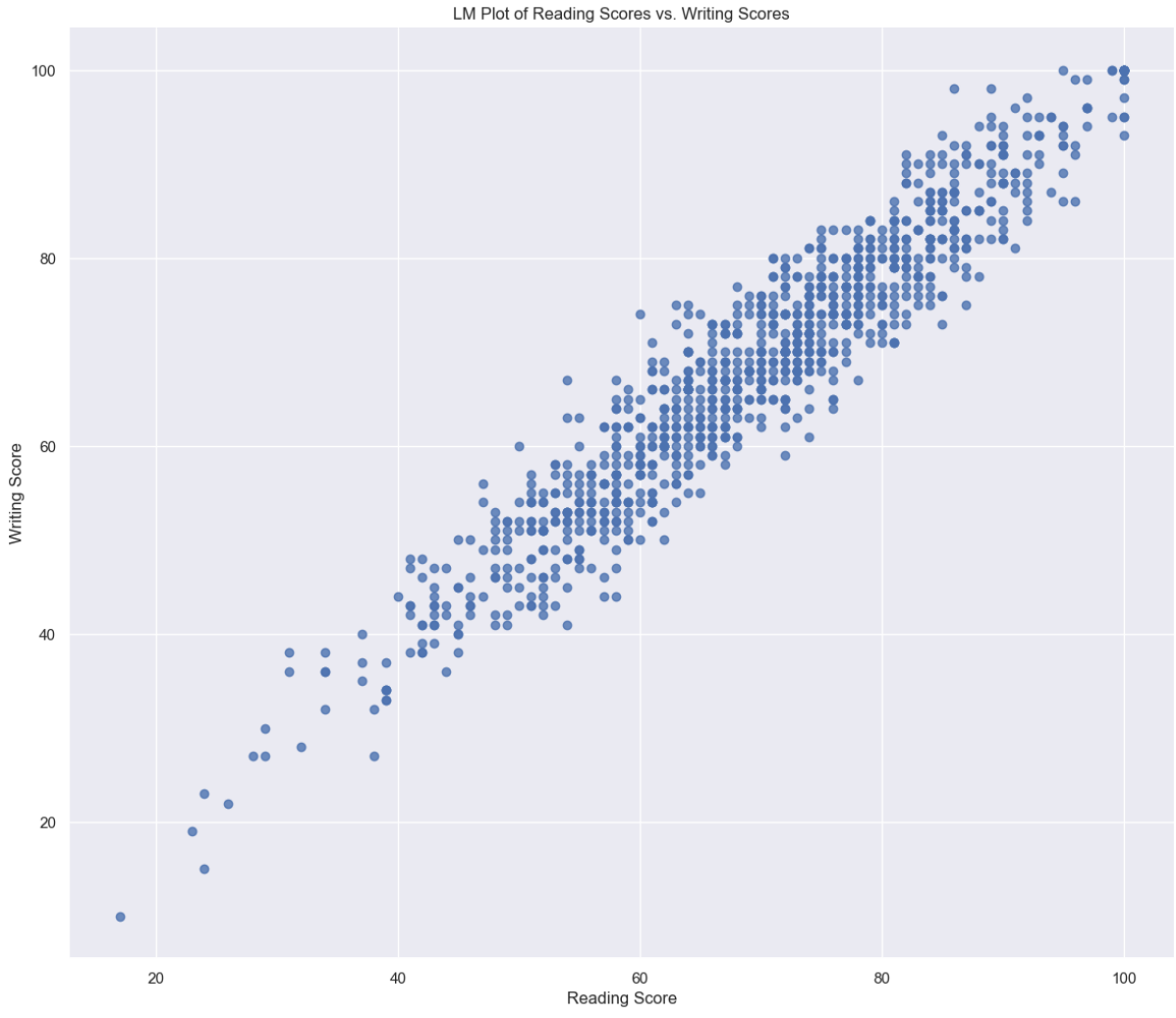
Hình 1.9 : LM plot của Reading Scores và Writing Scores

*Đặc điểm của hàm lmplot()

- *x='reading score'* : lấy dữ liệu từ cột reading score
- *y='writing score'* : lấy dữ liệu từ cột writing score
- *height=**10*** **: chiều cao của biểu độ**
- *aspect=**1.2*** **: tỉ lệ giữa chiều rộng và chiều cao**

8. lm plots (fit_reg=Flase)

```
sns.lmplot(x='reading score', y='writing score', data=data, height=10,
aspect=1.2,fit_reg=False)

# Đặt tiêu đề và nhãn trục
plt.title('LM Plot of Reading Scores vs. Writing Scores')
plt.xlabel('Reading Score')
plt.ylabel('Writing Score')

# Hiển thị biểu đồ
plt.show()
```



Hình 1.10 : LM plot của Reading Scores và Writing Scores

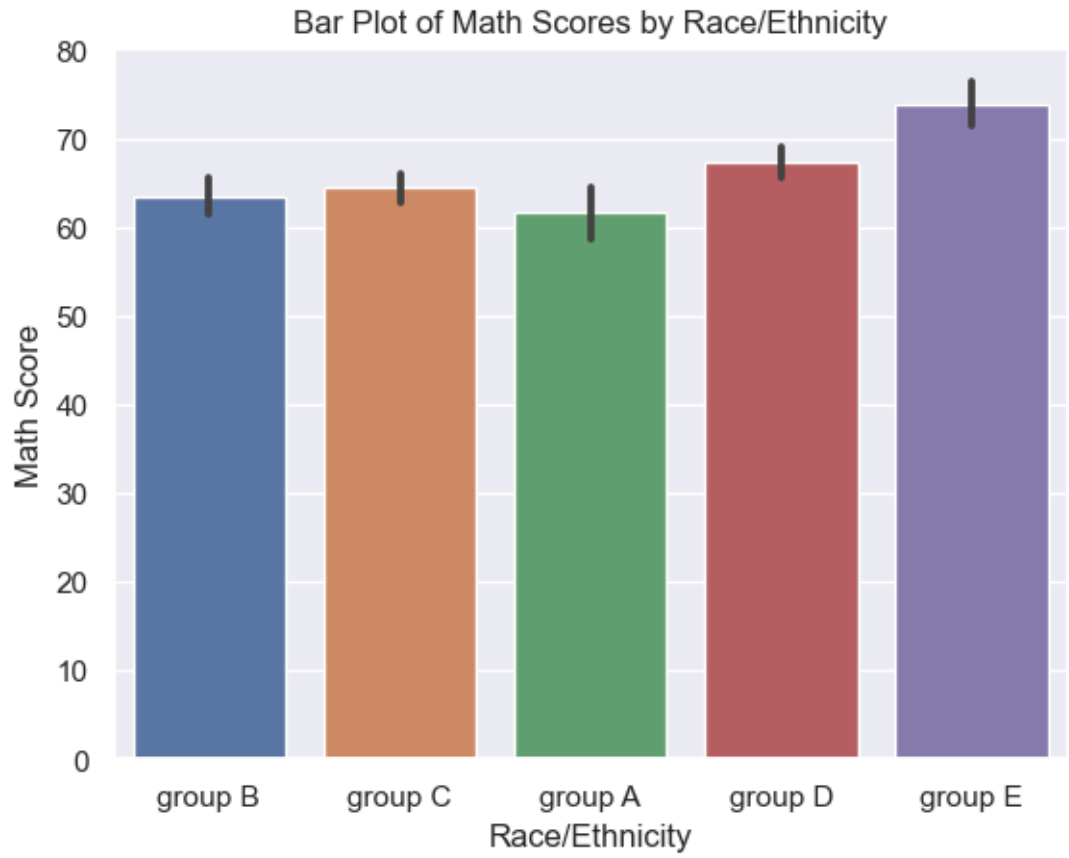
9. lm plots(hue='gender')

```
sns.lmplot(x='reading score', y='writing score', data=data, height=10,
aspect=1.2,fit_reg=False,hue='gender')
# Đặt tiêu đề và nhãn trục
plt.title('LM Plot of Reading Scores vs. Writing Scores')
plt.xlabel('Reading Score')
plt.ylabel('Writing Score')

# Hiển thị biểu đồ
plt.show()
```

Hình 1.11 : LM plot của Reading Scores và Writing Scores

10. Bar plots

```
sns.barplot(x='race/ethnicity', y='math score', data=data)

# Đặt tiêu đề và nhãn trục
plt.title('Bar Plot of Math Scores by Race/Ethnicity')
plt.xlabel('Race/Ethnicity')
plt.ylabel('Math Score')

# Hiển thị biểu đồ
plt.show()
```



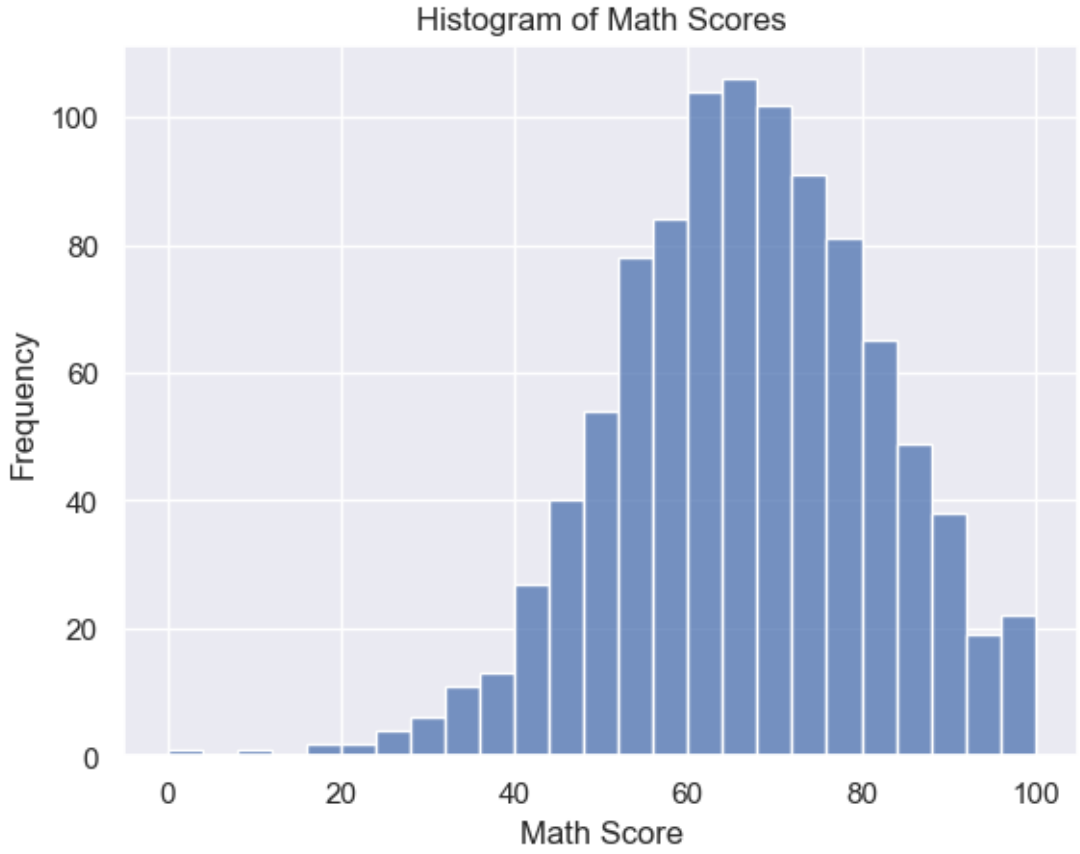
Hình 1.12 : Bar plot của Math Scores bởi Race/Ethnicity

*Đặc điểm của hàm barplot()

- *x='race/ethnicity'* : dữ liệu trục x
- *y='math score'* : dữ liệu trục y
- *data=data*

11. Histogram plots

```
sns.histplot(data['math score'], kde=False)

# Đặt tiêu đề và nhãn trục
plt.title('Histogram of Math Scores')
plt.xlabel('Math Score')
plt.ylabel('Frequency')

# Hiển thị biểu đồ
plt.show()
```



Hình 1.13 : Histogram plot của Math Scores

*Đặc điểm hàm hisplot()

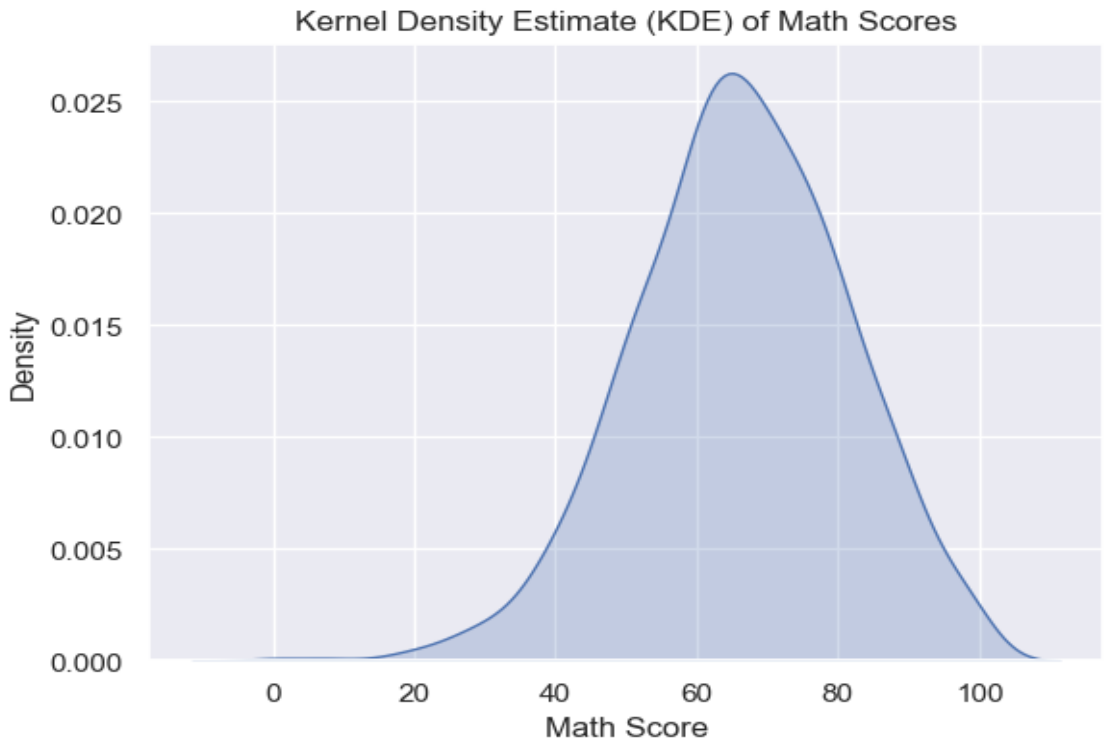
- *data['math score'],* : lấy dữ liệu từ cột math score

- *kde=False* : đường ước lượng mật độ xác suất sẽ được hiển thị trên biểu đồ, cùng với các cột dạng cột cơ bản của biểu đồ barplot.

+Kde để chỉ định xem có muốn hiển thị đường ước lượng mật độ xác suất (KDE - Kernel Density Estimation) trên biểu đồ barplot hay không.

12.KDE Plot

```
sns.kdeplot(data['math score'], shade=True)

# Đặt tiêu đề và nhãn trục
plt.title('Kernel Density Estimate (KDE) of Math Scores')
plt.xlabel('Math Score')
plt.ylabel('Density')

# Hiển thị biểu đồ
plt.show()
```



Hình 1.14 : KDE plot của Math Scores

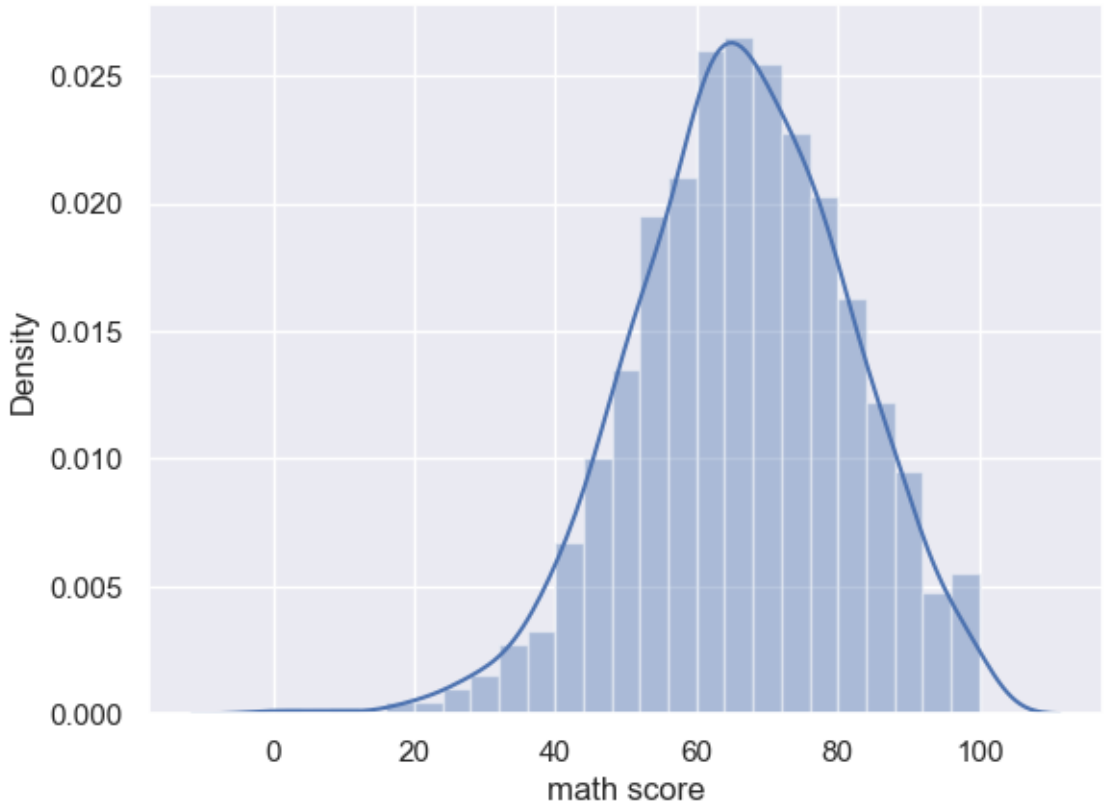
*Đặc điểm hàm kdeplot()

- *data['math score'] :* lấy dữ liệu từ cột math score
- *shade = True* : dưới đường ước lượng mật độ xác suất sẽ được tô màu

Shade để chỉ định xem có muốn tô màu (shading) dưới đường ước lượng mật độ xác suất (KDE - Kernel Density Estimation) trên biểu đồ KDE hay không.

13.Distribution plots

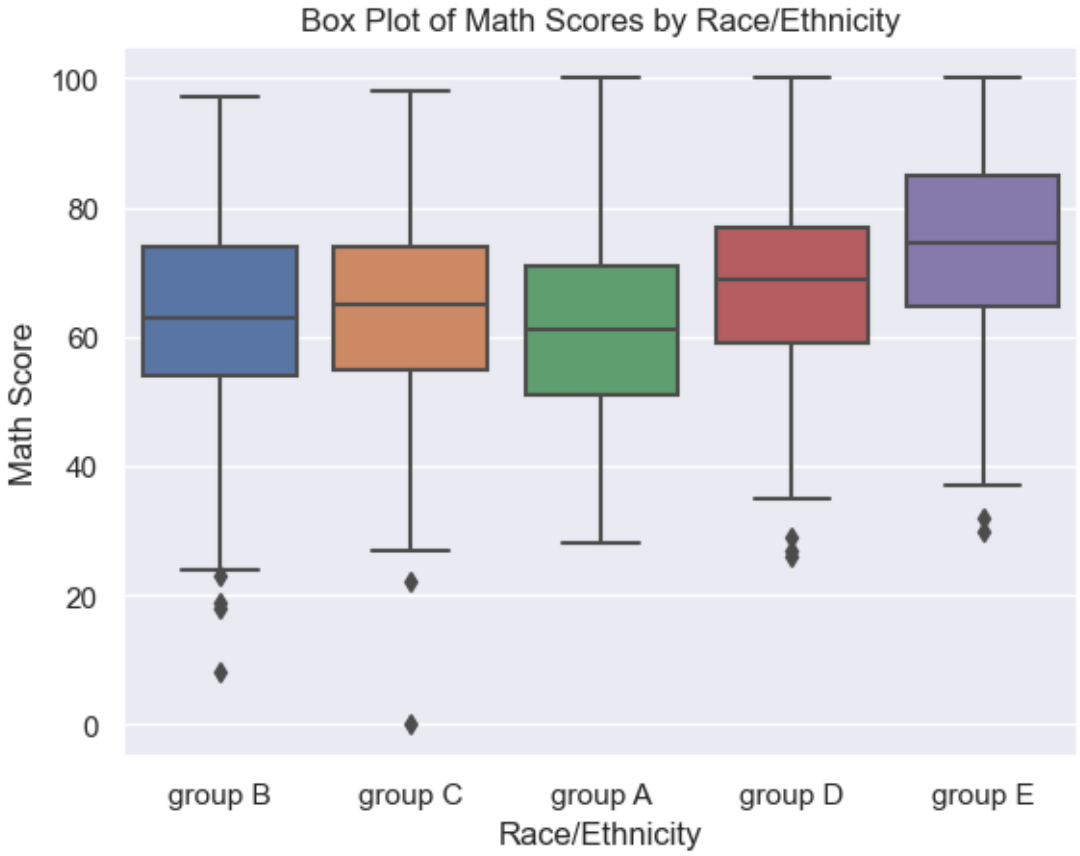
```
sns.distplot(data['math score'])

# Hiển thị biểu đồ
plt.show()
```



Hình 1.15 : Distribute plot của Math Scores

14. Box plots

```
# Vẽ biểu đồ box plot
sns.boxplot(x='race/ethnicity', y='math score', data=data)

# Đặt tiêu đề và nhãn trục
plt.title('Box Plot of Math Scores by Race/Ethnicity')
plt.xlabel('Race/Ethnicity')
plt.ylabel('Math Score')

# Hiển thị biểu đồ
plt.show()
```



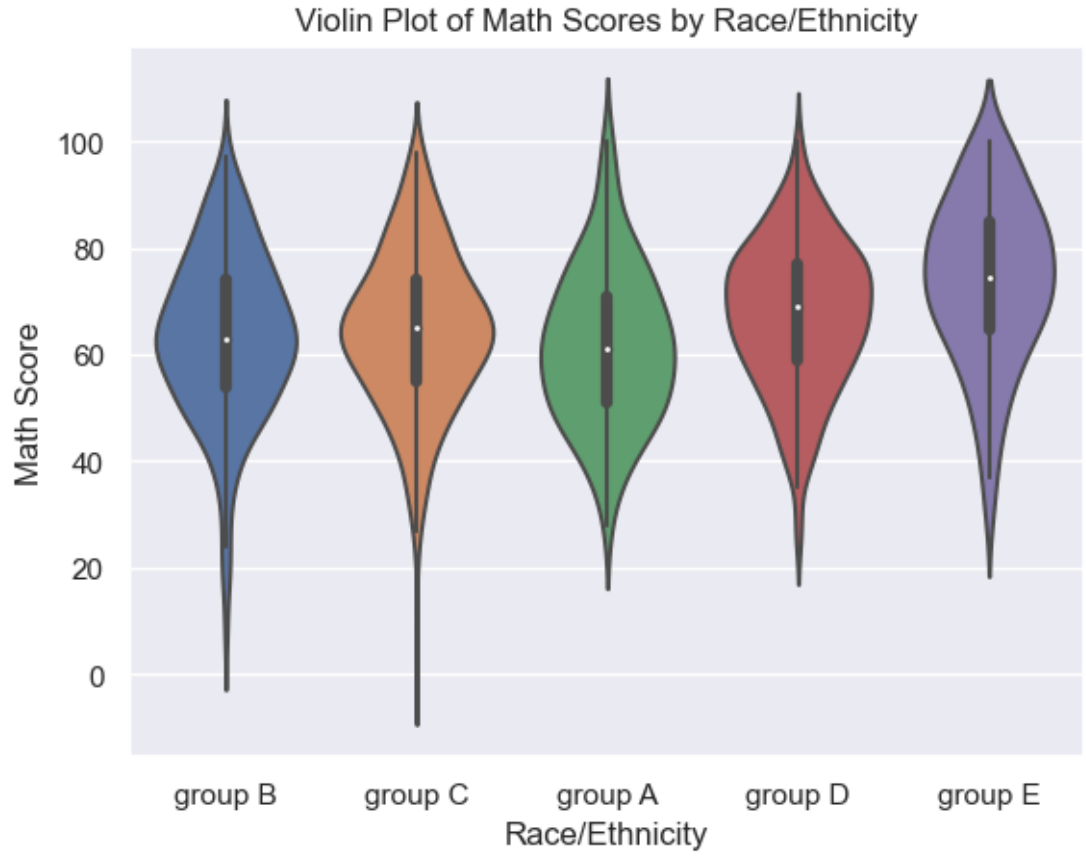
Hình 1.16 : Box plot của Math Scores bởi Race/Ethnicity

*Đặc điểm hàm boxplot()

- *x='race/ethnicity'* : dữ liệu trục x
- *y='math score'* : dữ liệu trục y
- *data = data :* xác định DataFrame chứa dữ liệu bạn muốn sử dụng cho biểu đồ.

15. Violin plots

```
# Vẽ biểu đồ violin plot
sns.violinplot(x='race/ethnicity', y='math score', data=data)

# Đặt tiêu đề và nhãn trục
plt.title('Violin Plot of Math Scores by Race/Ethnicity')
plt.xlabel('Race/Ethnicity')
plt.ylabel('Math Score')

# Hiển thị biểu đồ
plt.show()
```



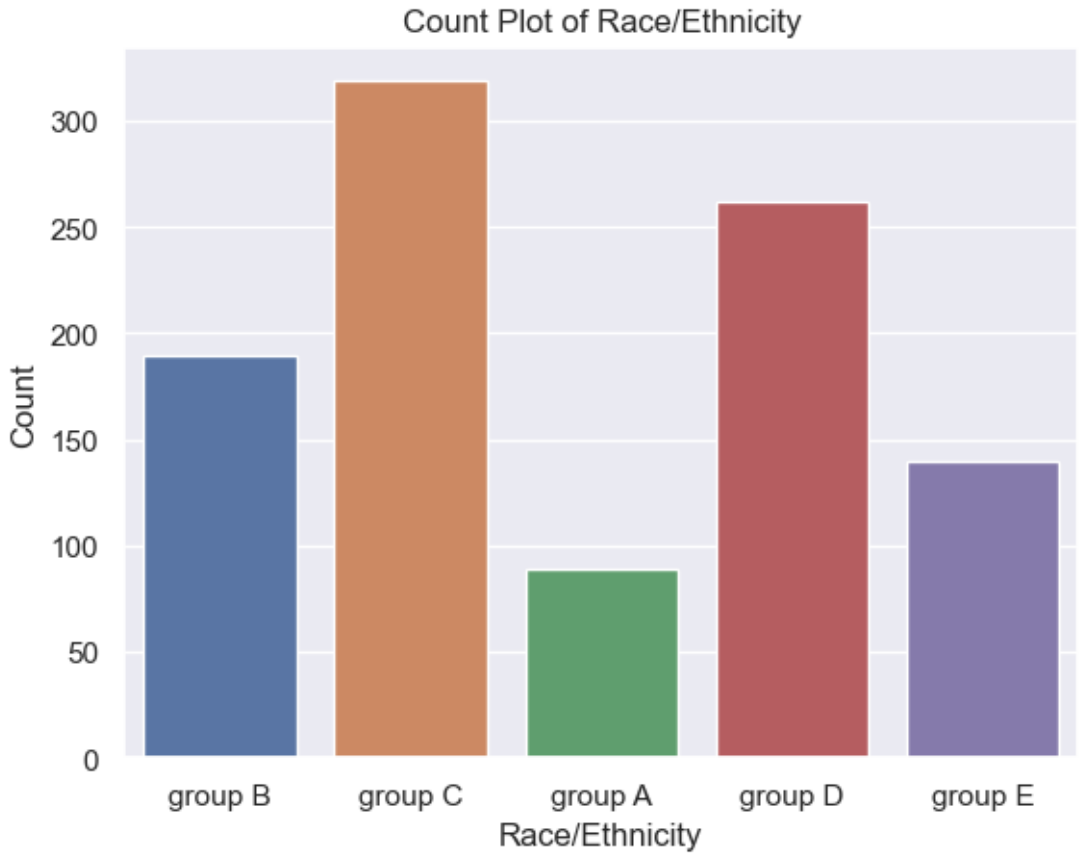
Hình 1.17 : Violin plot của Math Scores bởi Race/Ethnicity

*Đặc điểm hàm violinplot()

- *x='race/ethnicity'* : dữ liệu trục x
- *y='math score'* : dữ liệu trục y
- *data = data :* xác định DataFrame chứa dữ liệu bạn muốn sử dụng cho biểu đồ.

16. Count plots

```
sns.countplot(x='race/ethnicity', data=data)

# Đặt tiêu đề và nhãn trục
plt.title('Count Plot of Race/Ethnicity')
plt.xlabel('Race/Ethnicity')
plt.ylabel('Count')

# Hiển thị biểu đồ
plt.show()
```



Hình 1.18 : Count plot của Race/Ethnicity

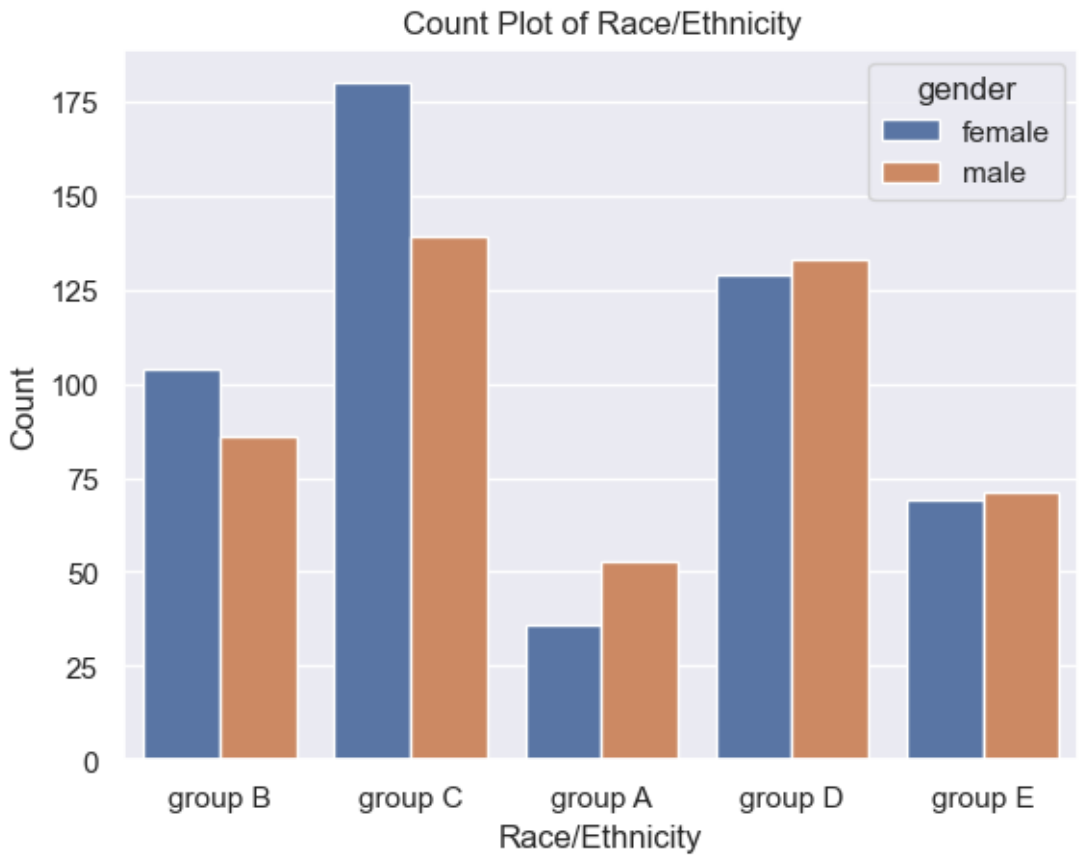
*Đặc điểm hàm countplot()

- *x='race/ethnicity'* : dữ liệu trục x

- *data = data :* xác định DataFrame chứa dữ liệu bạn muốn sử dụng cho biểu đồ.

17.Count Plot(hue=gender)

```
sns.countplot(x='race/ethnicity', data=data,hue='gender')

# Đặt tiêu đề và nhãn trục
plt.title('Count Plot of Race/Ethnicity')
plt.xlabel('Race/Ethnicity')
plt.ylabel('Count')

# Hiển thị biểu đồ
plt.show()
```



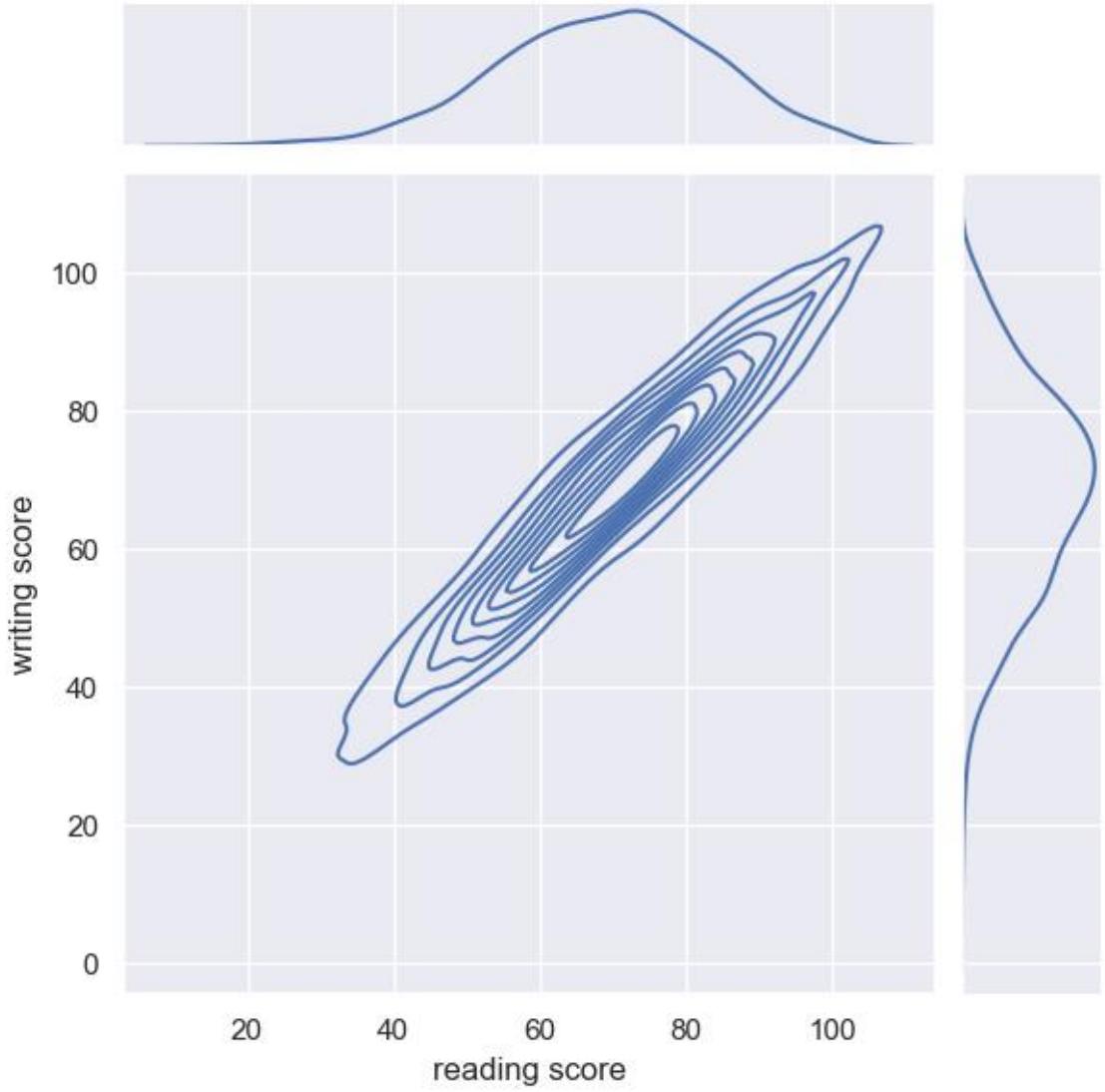
Hình 1.19 : Count plot của Race/Ethnicity

*Đặc điểm hàm countplot()

- *x='race/ethnicity'* : dữ liệu trục x
- *data = data :* xác định DataFrame chứa dữ liệu bạn muốn sử dụng cho biểu đồ.
- *hue='gender'* : biểu đồ hộp sẽ được vẽ cho mỗi nhóm phân loại

18.Joint plots

```
sns.jointplot(x='reading score', y='writing score', data=data, kind='kde')

# Hiển thị biểu đồ
plt.show()
plt.show()
```



Hình 1.20 : Joint plot của Writing Scores và Reading Score

*Đặc điểm hàm jointplot ()

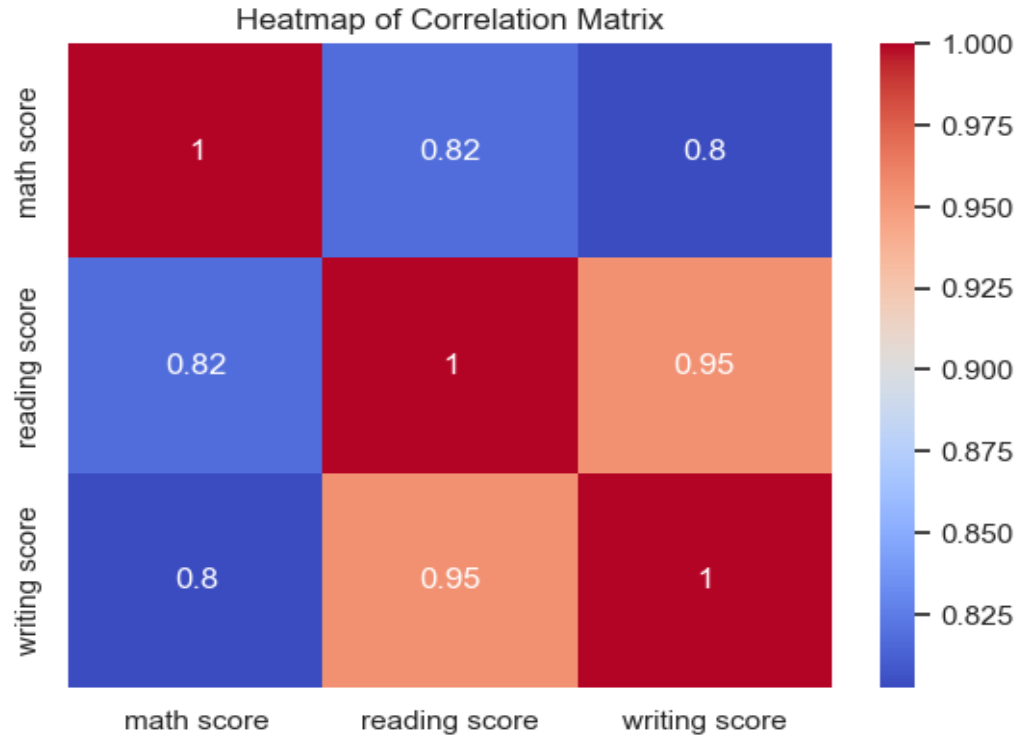
- *x='reading score'* : dữ liệu trục x
- *y='writing score'* : dữ liệu trục y
- *data = data :* xác định DataFrame chứa dữ liệu bạn muốn sử dụng cho biểu đồ.

+Kde để chỉ định xem có muốn hiển thị đường ước lượng mật độ xác suất (KDE

- Kernel Density Estimation) trên biểu đồ barplot hay không.

19. Heatmaps

```
numeric_data = data.select_dtypes(include=['float64', 'int64']) # Lọc các cột số
correlation_matrix = numeric_data.corr() # Tính toán ma trận tương quan


# Vẽ biểu đồ heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')

# Đặt tiêu đề
plt.title('Heatmap of Correlation Matrix')

# Hiển thị biểu đồ
plt.show()
```



Hình 1.21 : Heatmap của Correlation Matrix

*Đặc điểm của hàm heatmap()

- *correlation_matrix :* là ma trận tương quan hoặc ma trận dữ liệu bạn muốn hiển thị dưới dạng heatmap.

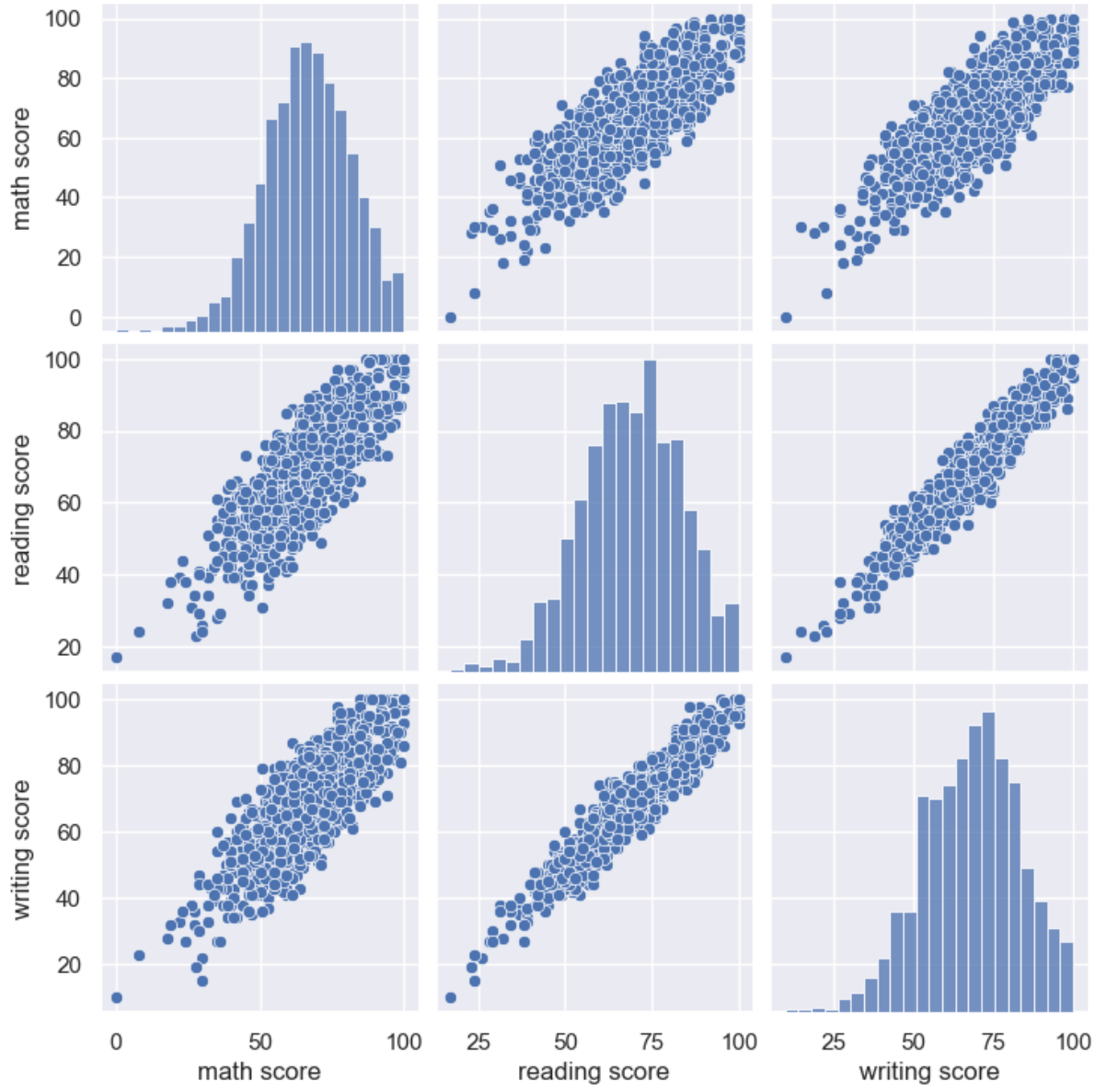
- *annot=True* : xác định xem có muốn hiển thị giá trị của từng ô trên heatmap hay không

- *cmap='coolwarm'* : xác định màu sắc của heatmap.

20. **Pair plots**

sns.pairplot(data)

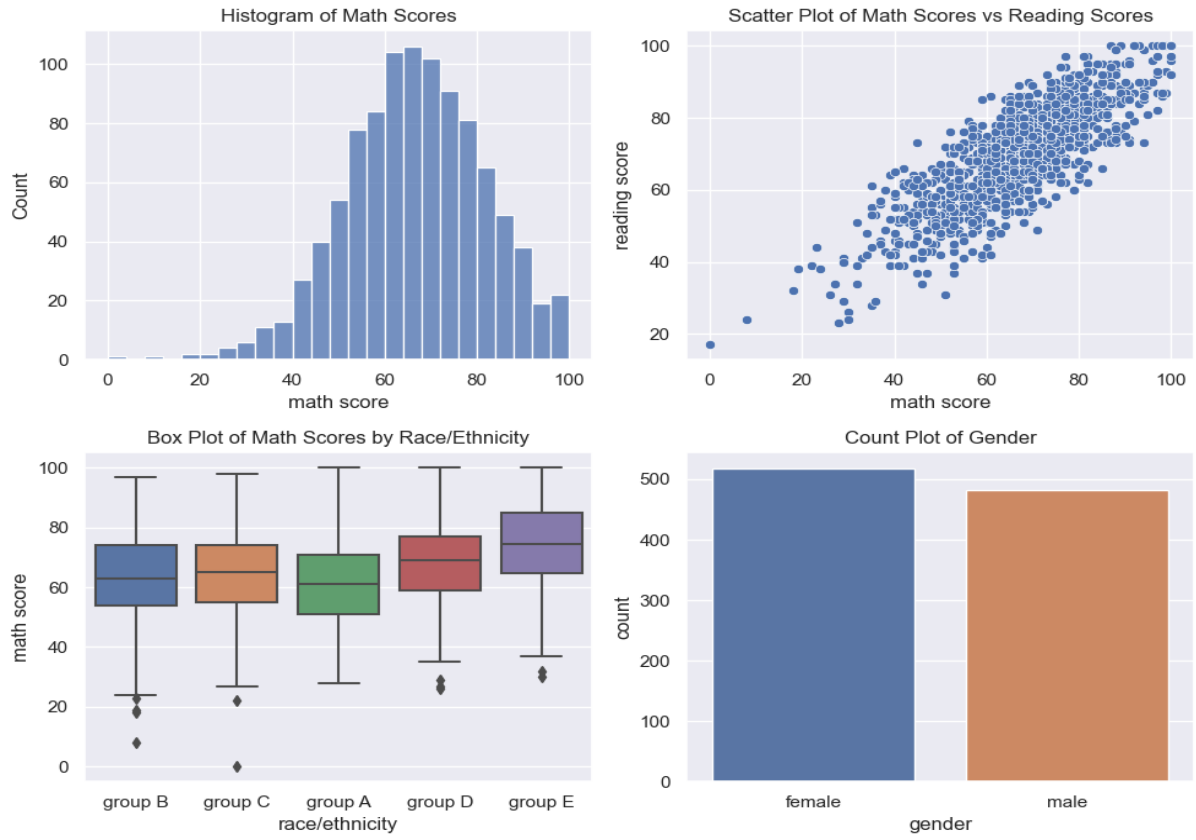
- Hàm pairplot(data), Đối số này xác định DataFrame bạn muốn sử dụng để vẽ biểu đồ pairplot.



Hình 1.22: Pair plot

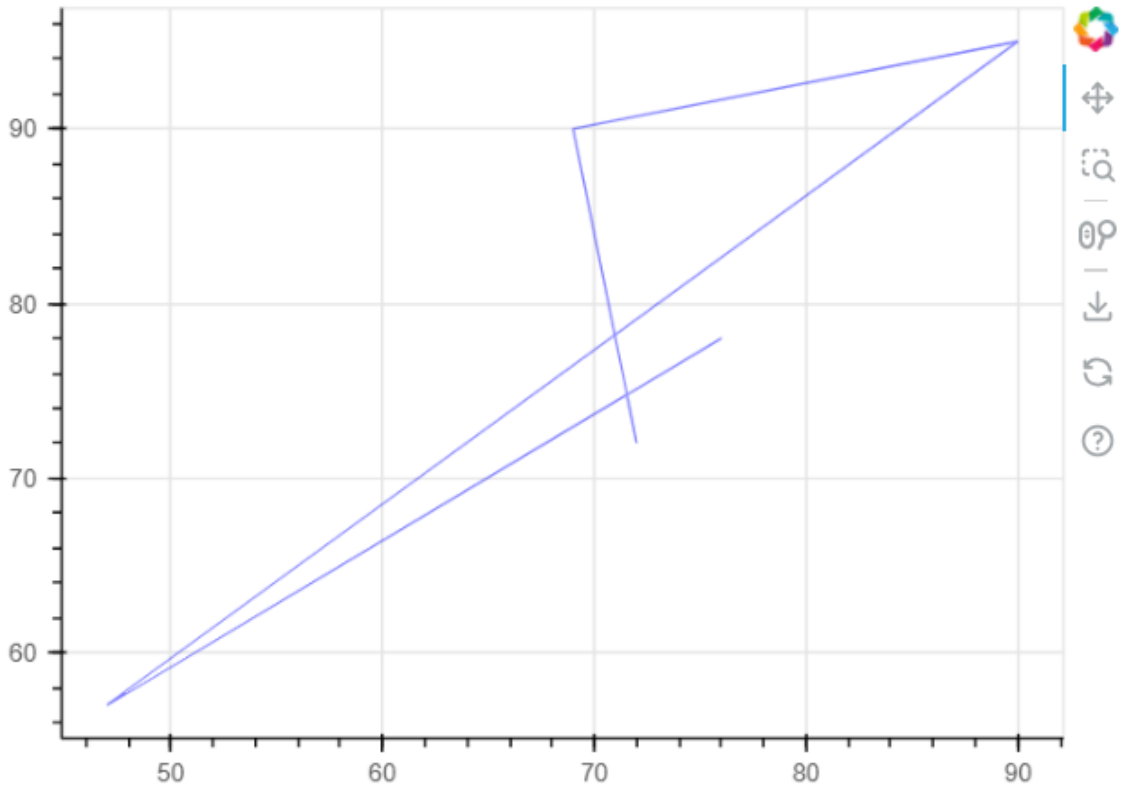
21. Tight layout

```
# Tạo một figure và các axes (subplot)
fig, axes = plt.subplots(2, 2, figsize=(10, 8))  # Tạo một lưới 2x2 subplot

# Biểu đồ 1: Biểu đồ histogram
sns.histplot(data['math score'], ax=axes[0, 0])
axes[0, 0].set_title('Histogram of Math Scores')

# Biểu đồ 2: Biểu đồ scatter plot
sns.scatterplot(x='math score', y='reading score', data=data, ax=axes[0, 1])
axes[0, 1].set_title('Scatter Plot of Math Scores vs Reading Scores')

# Biểu đồ 3: Biểu đồ box plot
sns.boxplot(x='race/ethnicity', y='math score', data=data, ax=axes[1, 0])
axes[1, 0].set_title('Box Plot of Math Scores by Race/Ethnicity')

# Biểu đồ 4: Biểu đồ count plot
sns.countplot(x='gender', data=data, ax=axes[1, 1])
axes[1, 1].set_title('Count Plot of Gender')

# Tăng khoảng cách giữa các subplot
plt.tight_layout()

# Hiển thị biểu đồ
plt.show()
```



22. Glyphs

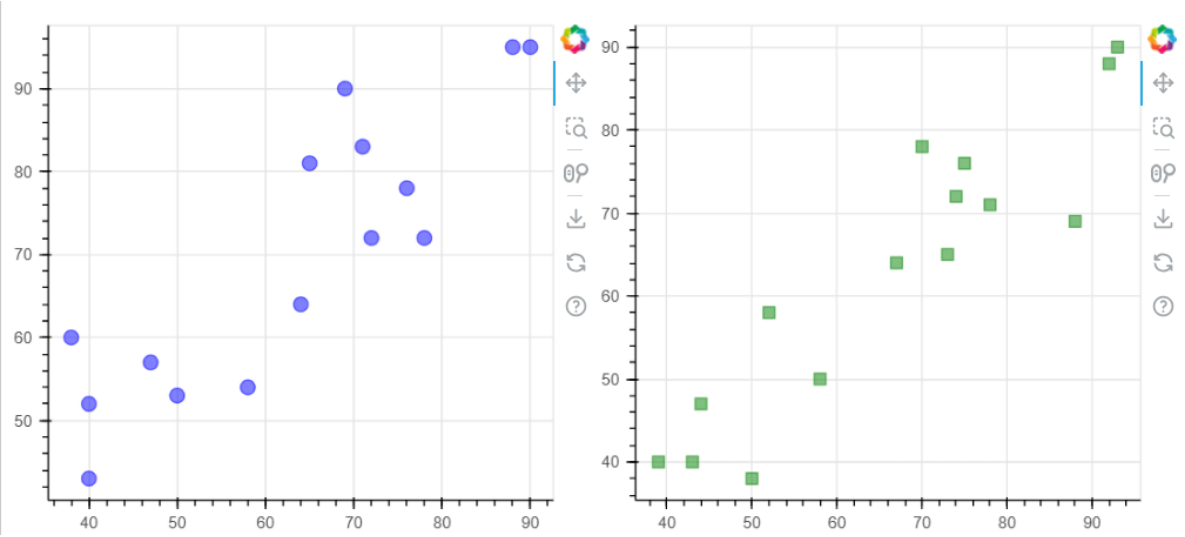
```
# Import the required modules
from bokeh.plotting import figure, output_notebook, show
from bokeh.layouts import *

data1=data.head(5)

# Lấy các cột chứa dữ liệu của các biểu tượng
x_values = data1['math score'] # Thay 'x_column' bằng tên cột chứa dữ liệu x
y_values = data1['reading score'] # Thay 'y_column' bằng tên cột chứa dữ liệu y


# Tạo một đối tượng figure
p = figure(width=500, height=350)

output_notebook()

# Tạo glyphs
p.line(x=x_values, y=y_values, line_width = 1, color='blue', alpha=0.5)

# Hiển thị đồ thị
show(p)
```



Hình 2.1:Biểu đồ Glyphs

23.Layouts(row)

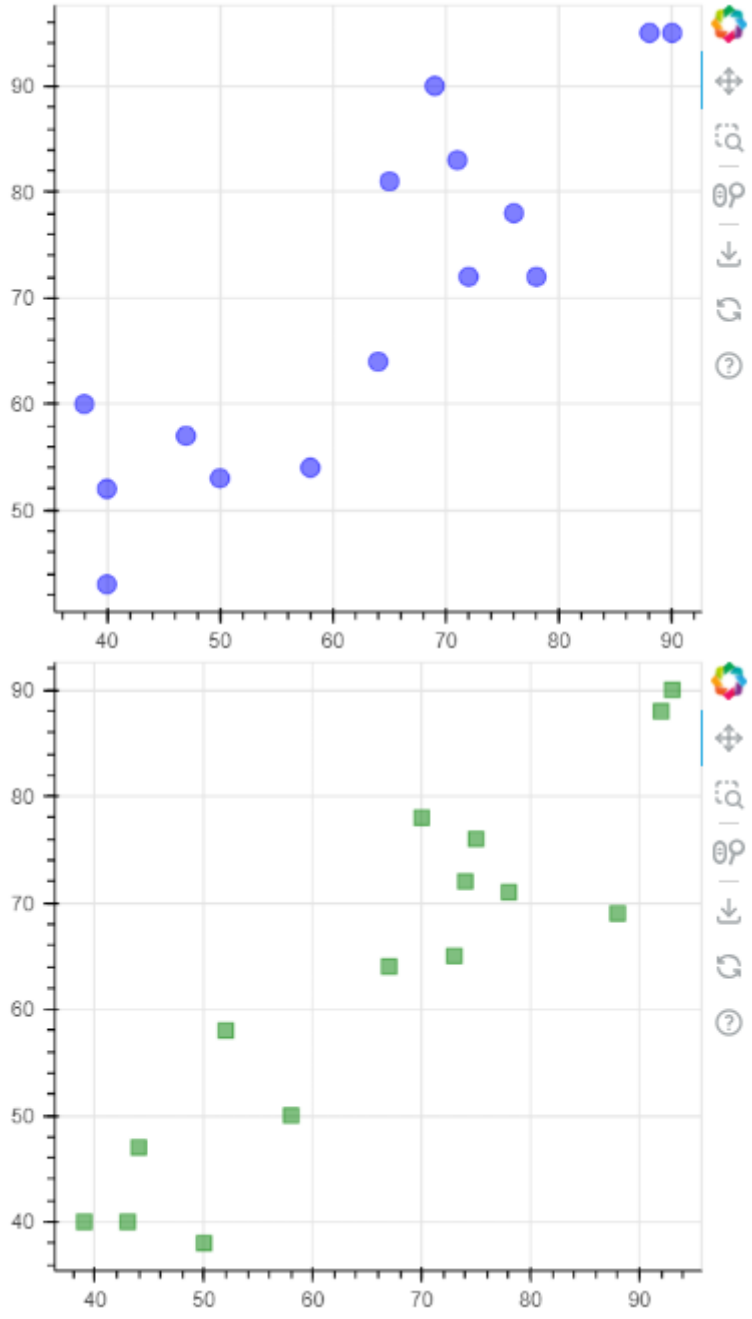
```
data2=data.head(15)
# Lấy dữ liệu từ các cột của DataFrame
x_values1 = data2['math score']
y_values1 = data2['reading score']
x_values2 = data2['writing score']
y_values2 = data2['math score']

# Output đồ thị tới notebook
output_notebook()

# Tạo đối tượng figure cho biểu đồ 1
p1 = figure(width=400, height=350)
p1.circle(x=x_values1, y=y_values1, size=10, color='blue', alpha=0.5)

# Tạo đối tượng figure cho biểu đồ 2
p2 = figure(width=400, height=350)
p2.square(x=x_values2, y=y_values2, size=8, color='green', alpha=0.5)

# Tạo layout row
layout = row(p1, p2)

# Hiển thị layout
show(layout)
```



Hình 2.2:Biểu đồ Layout row

24. Layouts(column)

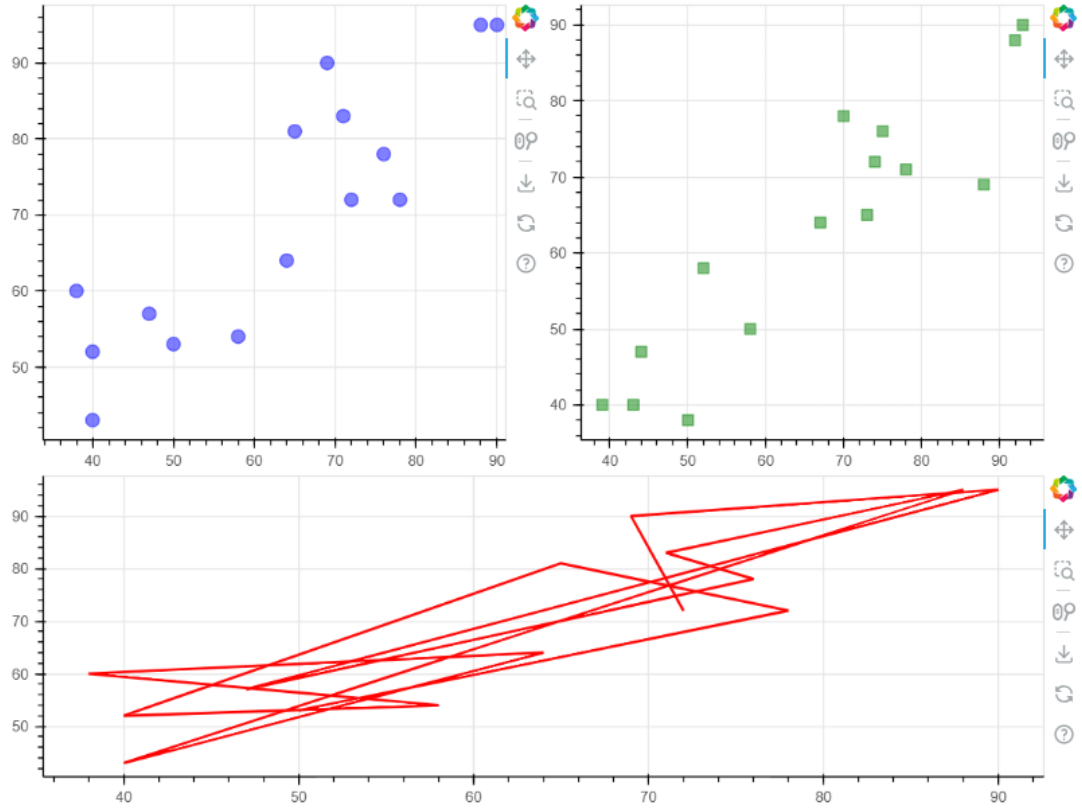
```
# Tạo layout column
layout = column(p1, p2)

# Hiển thị layout
show(layout)
```



Hình 2.3:Biểu đồ Layout column

25. Layouts(Nested)

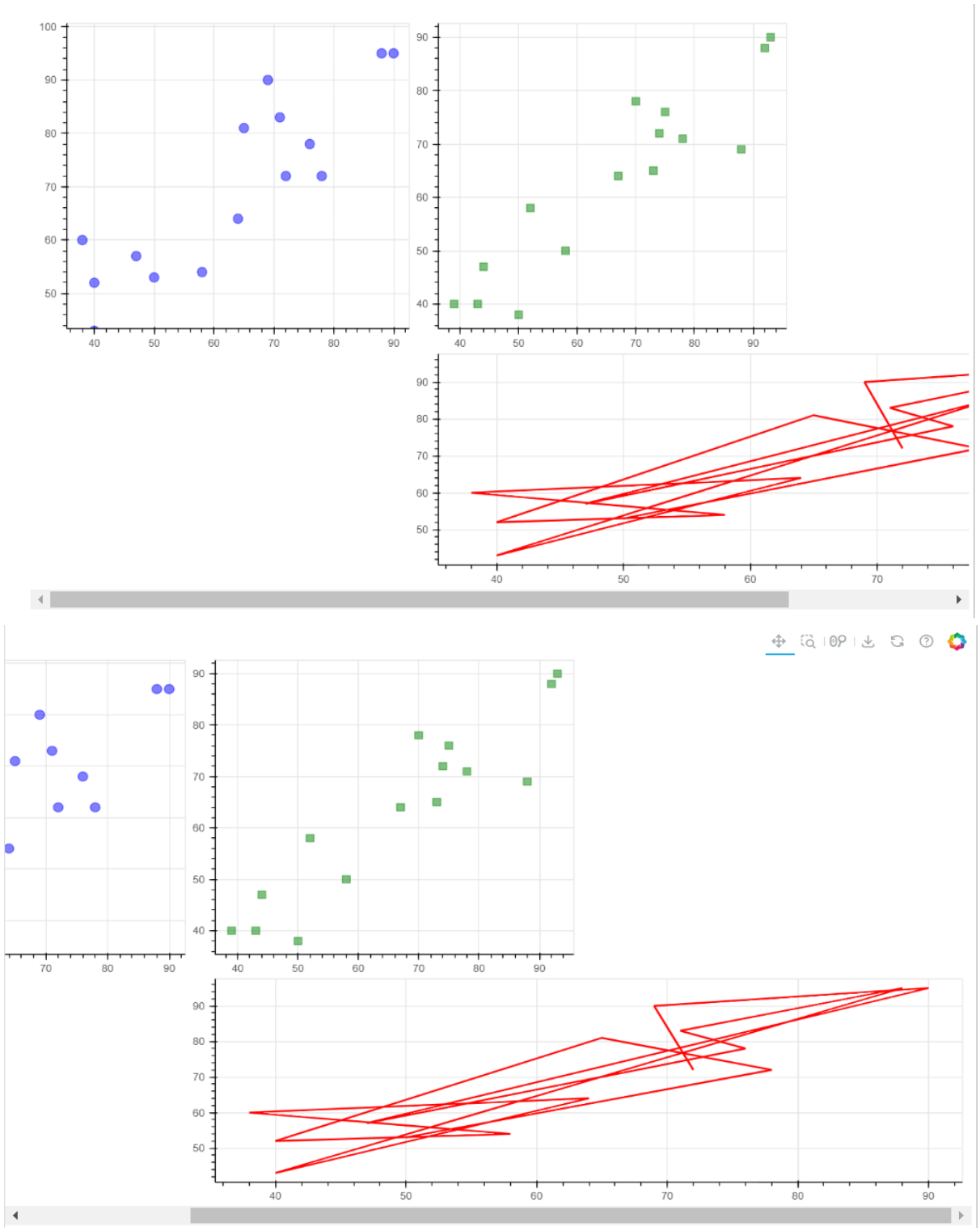
```
# Tạo layout row bao gồm p1 và p2
row_layout = row(p1, p2)

# Tạo đối tượng figure cho biểu đồ 3
p3 = figure(width=800, height=250)
p3.line(x_values1, y_values1, line_width=2, color='red')

# Tạo layout column bao gồm row_layout và p3
column_layout = column(row_layout, p3)

# Hiển thị layout
show(column_layout)
```



Hình 2.4:Biểu đồ Nested layout

26.**Layout(Grid)**

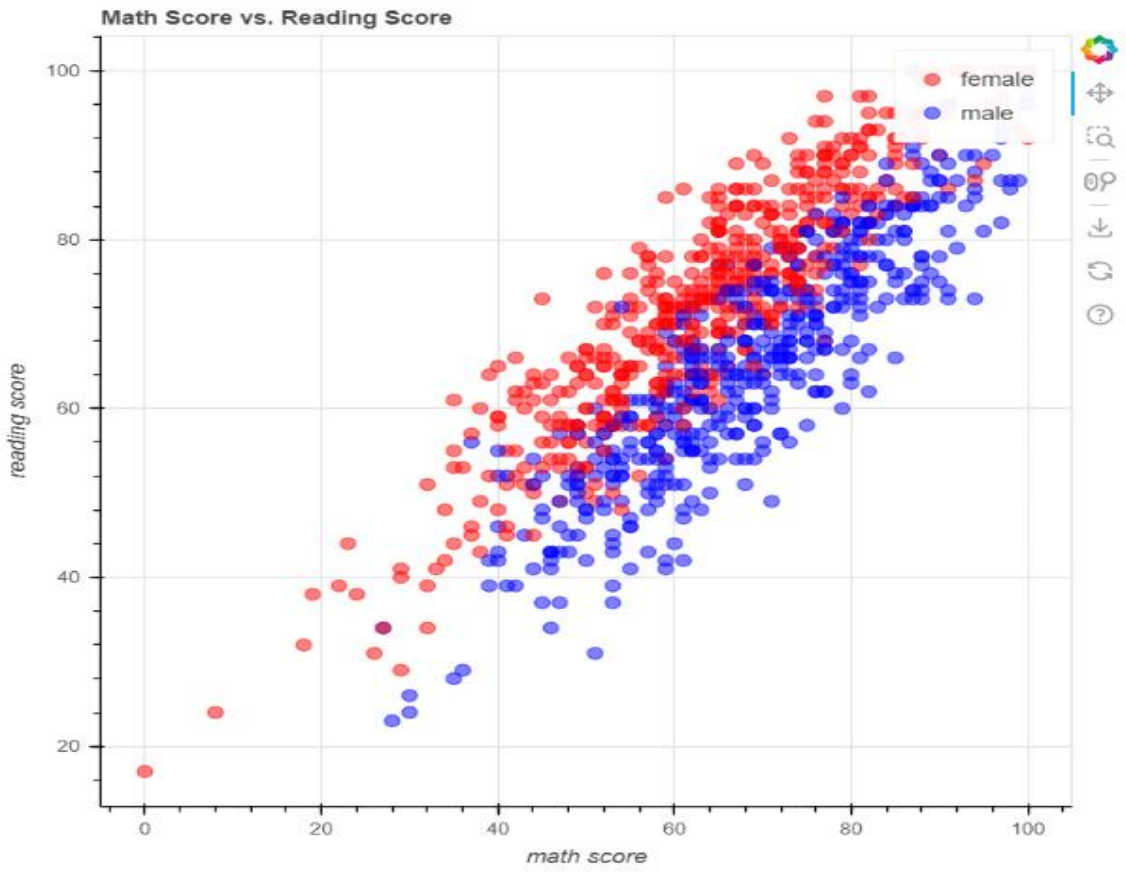
```
output_notebook()
grid_layout = gridplot([[p1, p2], [None,p3]])
# Show the plot
show(grid_layout)
```



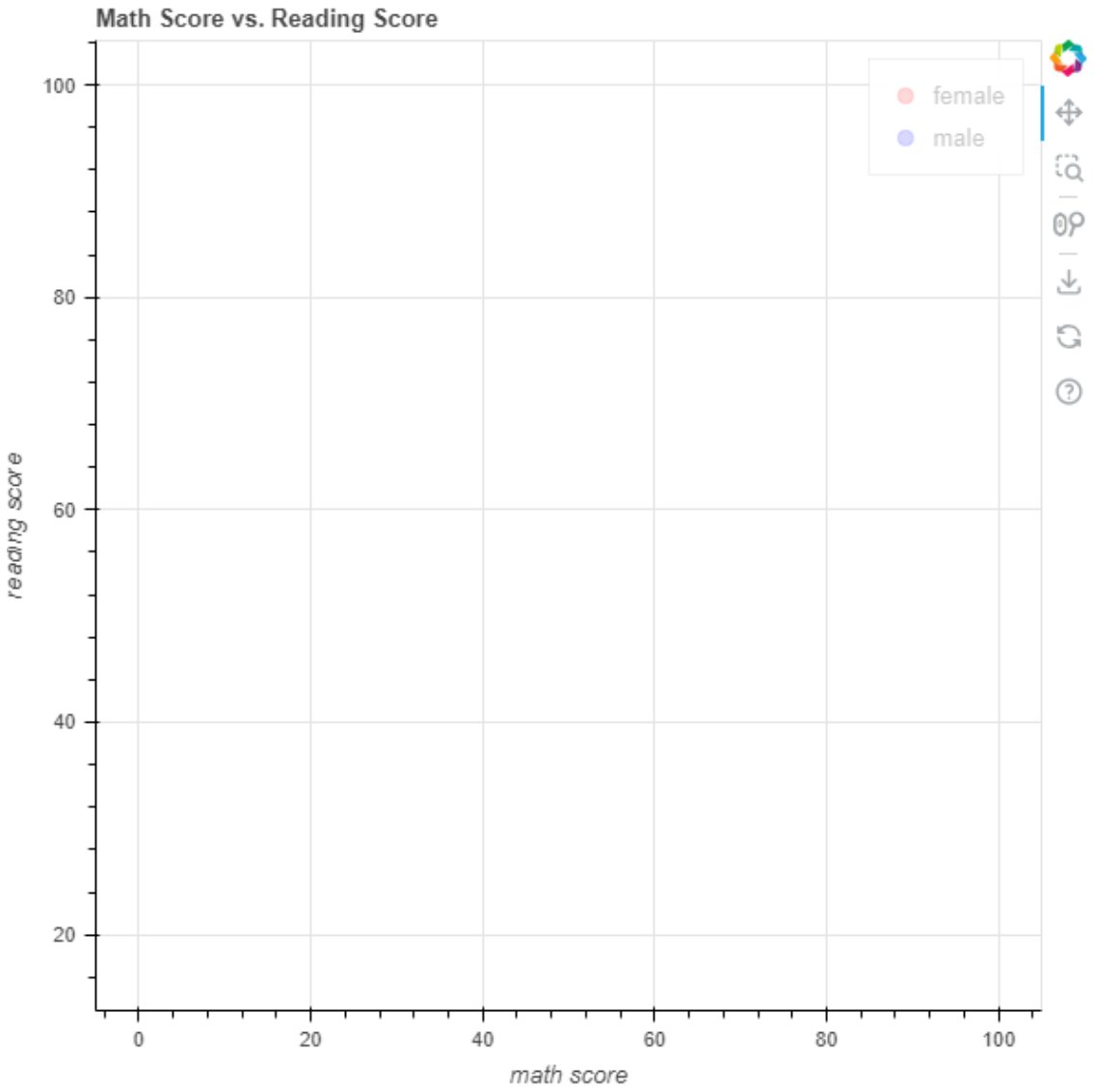
Hình 2.5:Biểu đồ Grid Layout

27.Hide click policy

```
# Output vào notebook
output_notebook()

# Tạo một đối tượng figure
p = figure(title="Math Score vs. Reading Score", x_axis_label='math score',
y_axis_label='reading score')

# Tạo một mức ánh xạ màu sắc phân loại
color_mapper = CategoricalColorMapper(factors=['male', 'female'], palette=['blue',
'red'])

# Tạo scatter plot với màu sắc phân loại
p.circle('math score', 'reading score', size=8, color={'field': 'gender', 'transform':
color_mapper}, alpha=0.5, legend_field='gender', source=data)


# Ẩn chính sách click cho legend
p.legend.click_policy = "hide"

# Hiển thị biểu đồ
show(p)
```



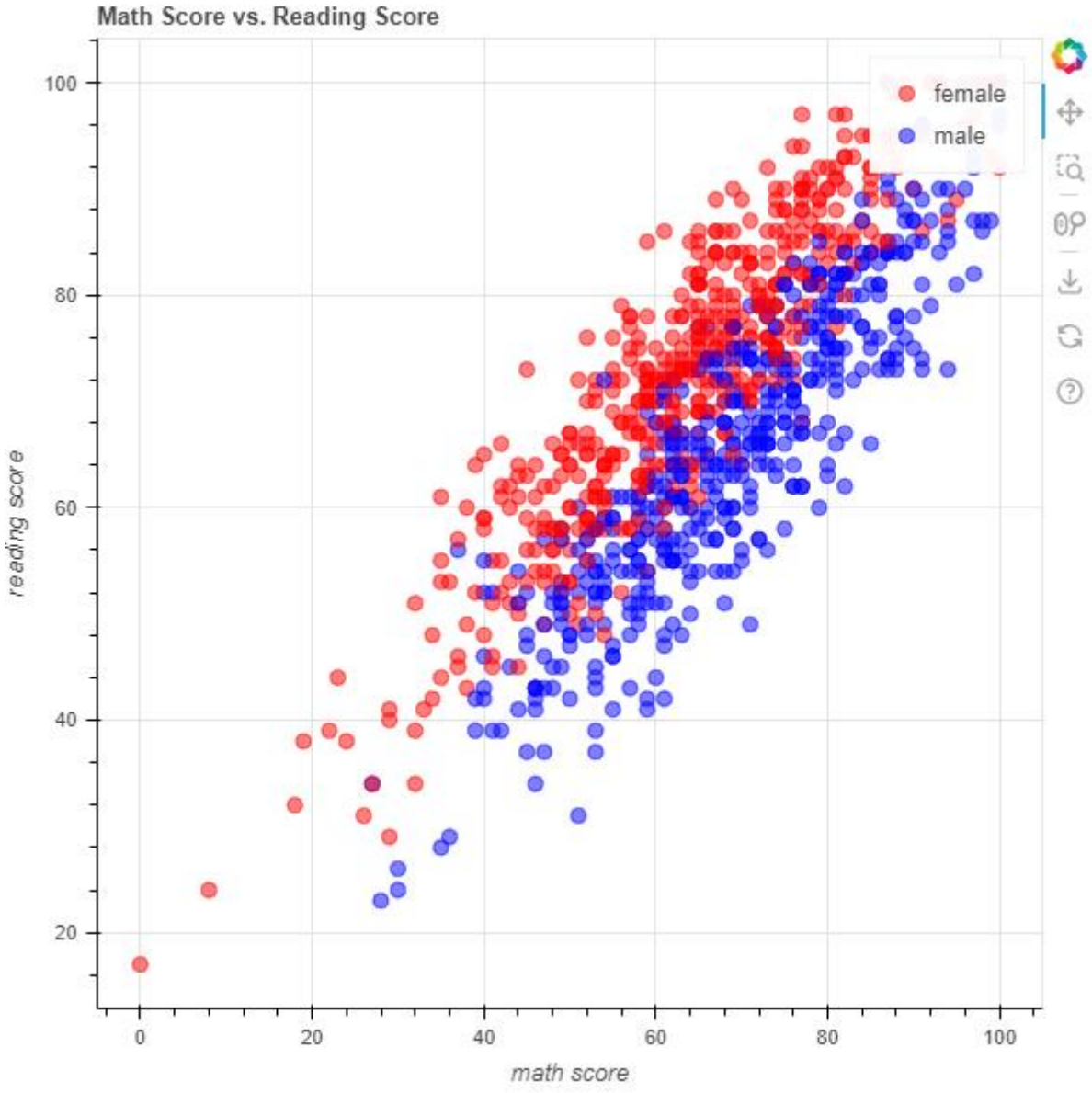
Hình 2.6:Biểu đồ khi chưa hide



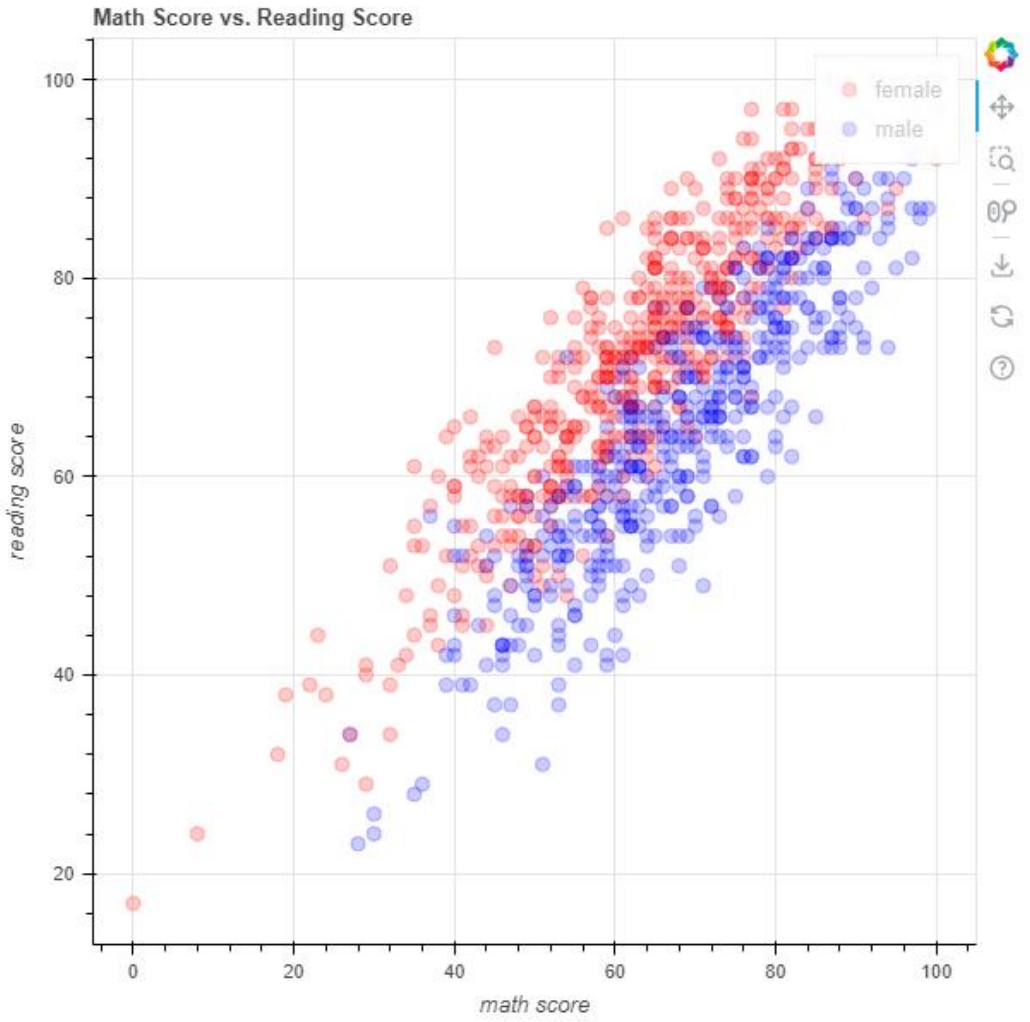
Hình 2.7:Biểu đồ sau khi hide

28. Mute click policy

```
# Tắt chính sách click cho legend
p.legend.click_policy = "mute"

# Hiển thị biểu đồ
show(p)
```



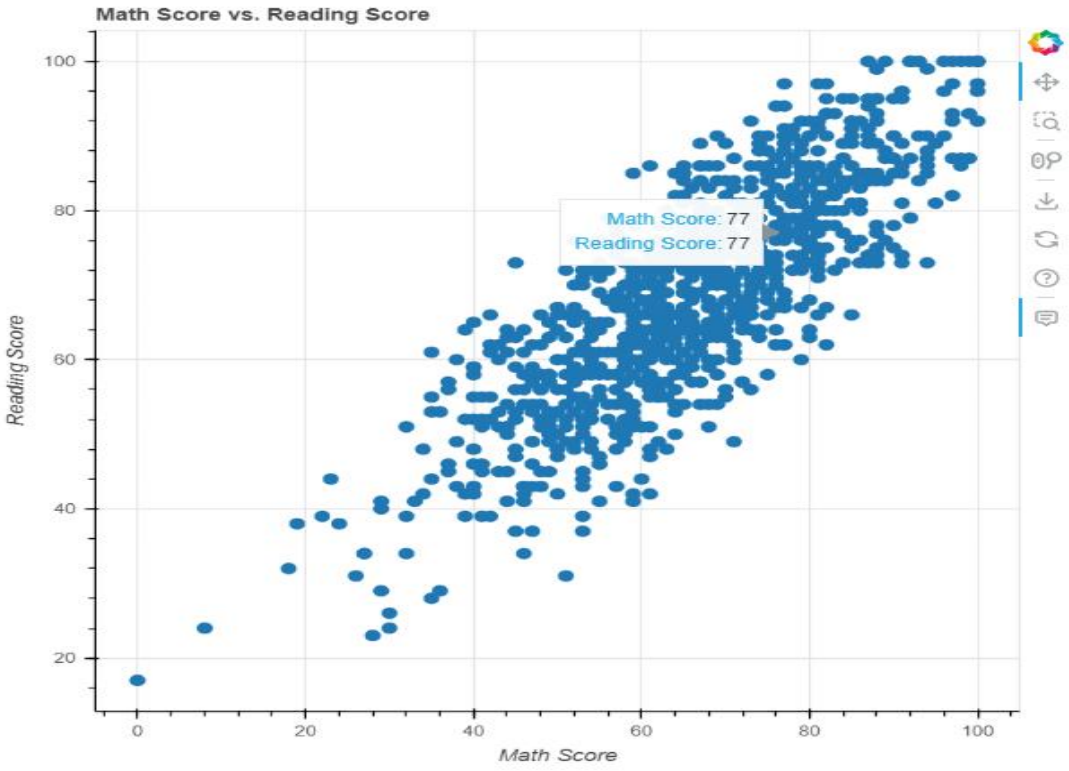
Hình 2.8:Biểu đồ trước khi mute



Hình 2.9:Biểu đồ sau khi mute

29.Hover tool

```
from bokeh.models import *  # Import HoverTool from bokeh.models
# Output vào notebook
output_notebook()

# Tạo một đối tượng figure
p = figure(title="Math Score vs. Reading Score", x_axis_label='Math Score',
y_axis_label='Reading Score')

# Tạo scatter plot
p.circle('math score', 'reading score', size=8, source=data)

# Thêm công cụ di chuột
hover = HoverTool()
hover.tooltips = [
    ("Math Score", "@{math score}"),
    ("Reading Score", "@{reading score}"),
    # Thêm thông tin khác tùy ý
]
p.add_tools(hover)

# Hiển thị biểu đồ
show(p)
```



Hình 2.10:Biểu đồ sau khi hover

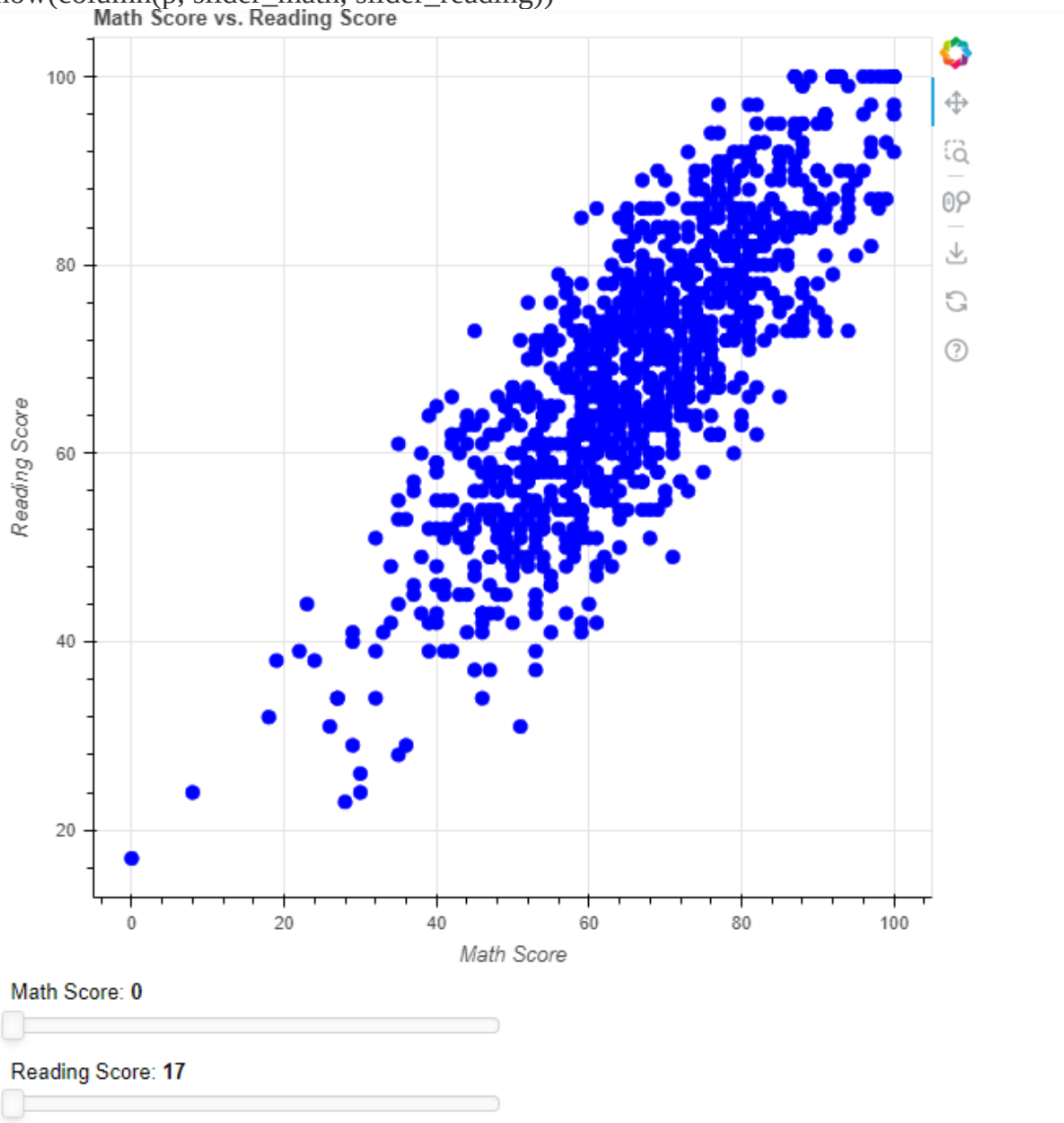
30. Tab panel

```
from bokeh.plotting import figure, output_notebook, show
from bokeh.models.widgets import Tabs, Panel
import pandas as pd

# Assume 'data' is your DataFrame containing math, reading, and writing scores

# Output to notebook
output_notebook()

# Create figures for each tab
p1 = figure(title="Math Score vs. Reading Score", x_axis_label='Math Score',
y_axis_label='Reading Score')
p1.circle('math score', 'reading score', size=8, source=data)

p2 = figure(title="Math Score vs. Writing Score", x_axis_label='Math Score',
y_axis_label='Writing Score')
p2.circle('math score', 'writing score', size=8, source=data)

# Create panels for each tab
tab1 = Panel(child=p1, title="Math vs. Reading")
tab2 = Panel(child=p2, title="Math vs. Writing")

# Create tabs and add panels to them
tabs = Tabs(tabs=[tab1, tab2])

# Show the tab panel
show(tabs)
```

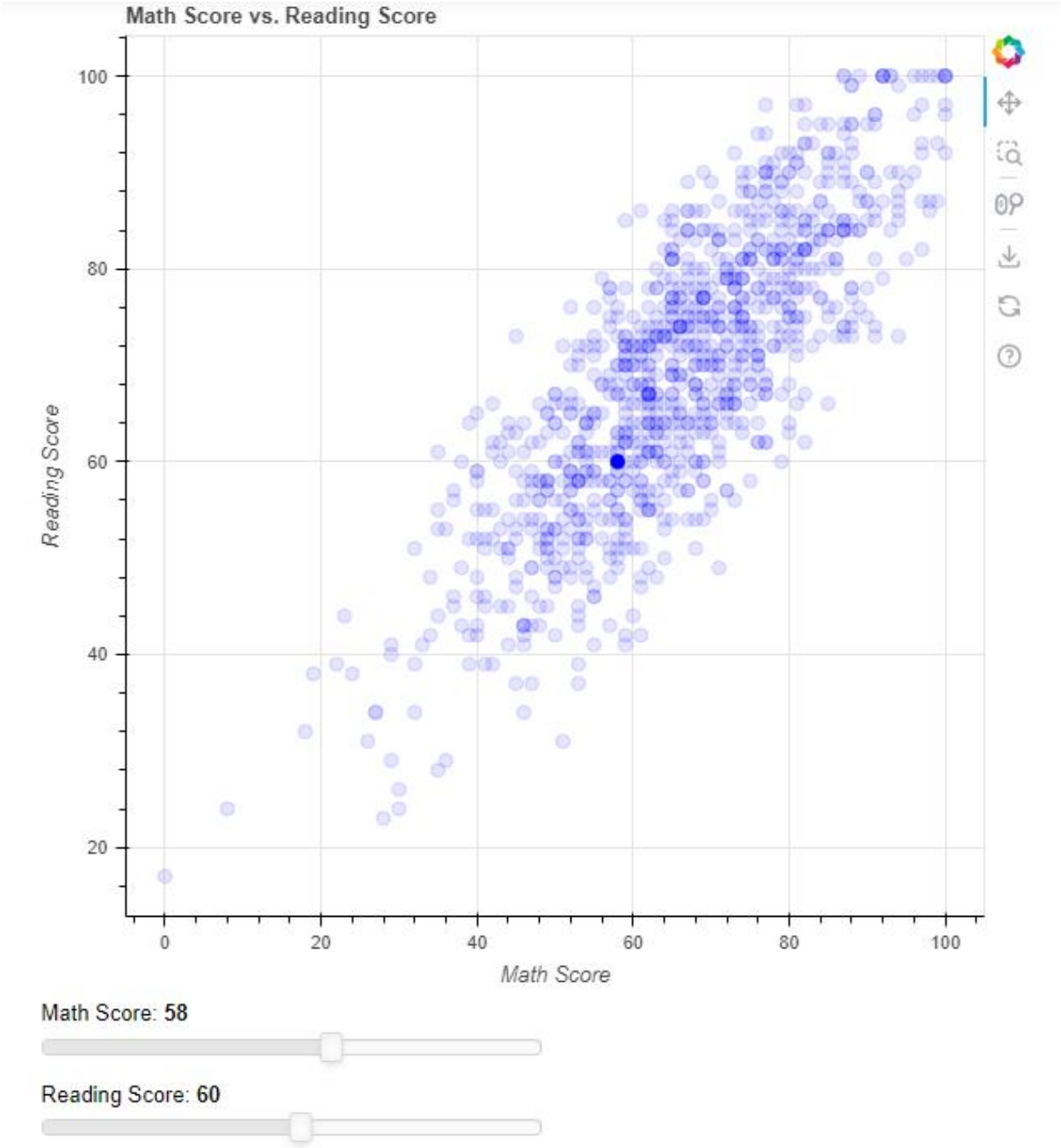
31. Slider

```
# Output vào notebook
output_notebook()

# Tạo một đối tượng ColumnDataSource từ DataFrame
source = ColumnDataSource(data)

# Tạo một đối tượng figure
p = figure(title="Math Score vs. Reading Score", x_axis_label='Math Score',
y_axis_label='Reading Score')

# Vẽ scatter plot cho reading score
p.circle(x='math score', y='reading score', size=8, color='blue', source=source)


# Tạo slider cho biến "math score"
slider_math = Slider(start=data['math score'].min(), end=data['math score'].max(),
value=data['math score'].min(), step=1, title="Math Score")

# Tạo slider cho biến "reading score"
slider_reading = Slider(start=data['reading score'].min(), end=data['reading
score'].max(), value=data['reading score'].min(), step=1, title="Reading Score")

# Tạo JavaScript callback để cập nhật dữ liệu khi slider thay đổi
callback = CustomJS(args=dict(source=source, slider_math=slider_math,
slider_reading=slider_reading), code="""
    const data = source.data;
    const math_score = slider_math.value;
    const reading_score = slider_reading.value;
    const math_scores = data['math score'];
    const reading_scores = data['reading score'];

    // Tạo mảng mới chứa chỉ số của các điểm phù hợp
    const indices = [];
    for (let i = 0; i < math_scores.length; i++) {
        if (math_scores[i] == math_score && reading_scores[i] == reading_score) {
            indices.push(i);
        }
    }

    // Cập nhật dữ liệu của source chỉ với các điểm phù hợp
    source.selected.indices = indices;
    source.change.emit();
""")
slider_math.js_on_change('value', callback)
slider_reading.js_on_change('value', callback)
```

```
# Hiển thị biểu đồ và slider
show(column(p, slider_math, slider_reading))
```



Hình 2.11:Biểu đồ trước khi lọc



Hình 2.12:Biểu đồ sau khi lọc

CHƯƠNG III:KẾT LUẬN

Trong quá trình khám phá và sử dụng các thư viện trực quan hóa dữ liệu như Matplotlib, Seaborn và Bokeh, chúng ta đã nắm vững các công cụ và kỹ thuật cơ bản để biểu diễn và phân tích dữ liệu một cách mạnh mẽ và linh hoạt. Dưới đây là những điểm chính mà chúng ta đã tìm hiểu:

1. **Matplotlib**:

    - Là một thư viện trực quan hóa dữ liệu cơ bản và mạnh mẽ, cho phép chúng ta tạo ra nhiều loại biểu đồ phức tạp và tùy chỉnh chúng đến mức cao.
    - Dễ dàng tích hợp với các thư viện khác và hỗ trợ đa dạng các định dạng đầu ra.

2. **Seaborn**:

    - Tích hợp trên nền tảng Matplotlib và cung cấp các biểu đồ trực quan hóa dữ liệu cao cấp và dễ đọc.
    - Hỗ trợ nhanh chóng cho việc hiển thị các mối quan hệ phức tạp giữa các biến và phân tích dữ liệu nhanh chóng.

3. **Bokeh**:

    - Là một thư viện trực quan hóa dữ liệu mạnh mẽ cho việc tạo ra các ứng dụng web và biểu đồ tương tác.
    - Cho phép chúng ta tạo ra các biểu đồ tương tác và ứng dụng web mà không cần phải có kiến thức về lập trình web.

Từ việc sử dụng và khám phá các thư viện này, chúng ta có thể thấy rằng việc trực quan hóa dữ liệu không chỉ là cách đơn giản để hiển thị thông tin mà còn là một công cụ mạnh mẽ để khám phá, phân tích và truyền đạt thông tin một cách hiệu quả. Các thư viện này đều có những đặc điểm và ưu điểm riêng, giúp

chúng ta linh hoạt trong việc chọn lựa và áp dụng vào các tình huống và mục tiêu phân tích khác nhau.

Cuối cùng, việc nắm vững các kỹ thuật và công cụ trực quan hóa dữ liệu này sẽ giúp chúng ta không chỉ nâng cao kỹ năng phân tích dữ liệu mà còn mở ra nhiều cơ hội mới trong lĩnh vực phân tích dữ liệu và khoa học dữ liệu.

TÀI LIỆU THAM KHẢO

1. **Python Data Analysis Third Edition**
2. Python-Data-Analysis-Third-Edition/Chapter05/Ch5.ipynb at master · PacktPublishing/Python-Data-Analysis-Third-Edition · GitHub
3. **https://www.kaggle.com/datasets/rkiattisak/student-performance-in-mathematics**