Chương 6: Trực quan hóa Dữ liệu với Seaborn

Contents

Trực quan hóa Dữ liệu với Seaborn

Seaborn là một thư viện trực quan hóa dữ liệu mạnh mẽ, được xây dựng trên Matplotlib. Thư viện này cung cấp các biểu đồ nâng cao và giao diện đơn giản để làm việc với dữ liệu dạng bảng, đặc biệt khi tích hợp với pandas.

6.1 Giới thiệu về Seaborn

6.1.1 Cài đặt Seaborn

Để cài đặt Seaborn, sử dụng lệnh:

```
pip install seaborn
```

6.1.2 Nhập thư viện và dữ liệu mẫu

Seaborn đi kèm với nhiều tập dữ liệu mẫu (datasets). Bạn có thể tải và sử dụng chúng để thực hành.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Tải dữ liệu mẫu
tips = sns.load_dataset("tips")
print(tips.head())
```

```
import plotly.io as pio
pio.renderers.default = 'vscode'
```

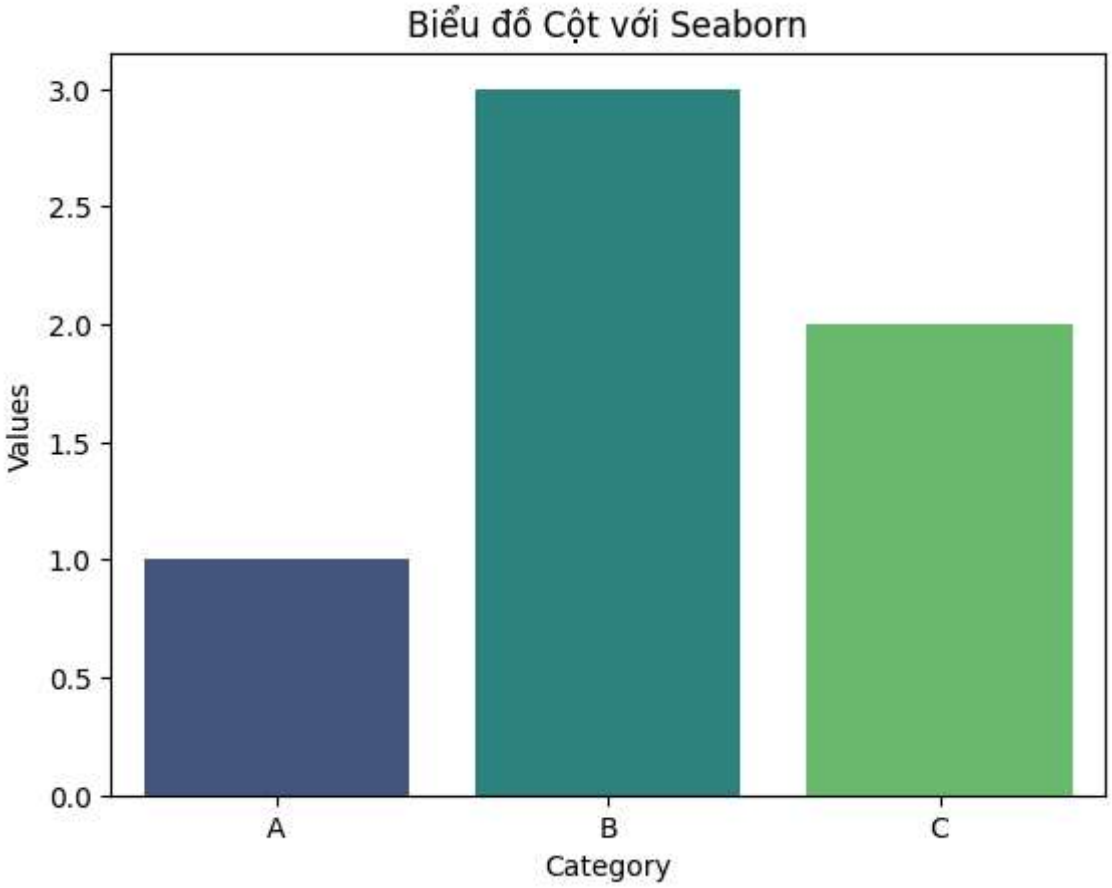
6.2 Các biểu đồ cơ bản với Seaborn

6.2.1 Biểu đồ cột và phân phối Biểu đồ cột:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import warnings

warnings.filterwarnings("ignore", category=FutureWarning)

data = {
    "Category": ["A", "B", "C"],
    "Values": [1, 3, 2]
}
df = pd.DataFrame(data)
sns.barplot(data=df, x="Category", y="Values", palette="viridis")
plt.title('Biểu đồ Cột với Seaborn')
plt.show()
```

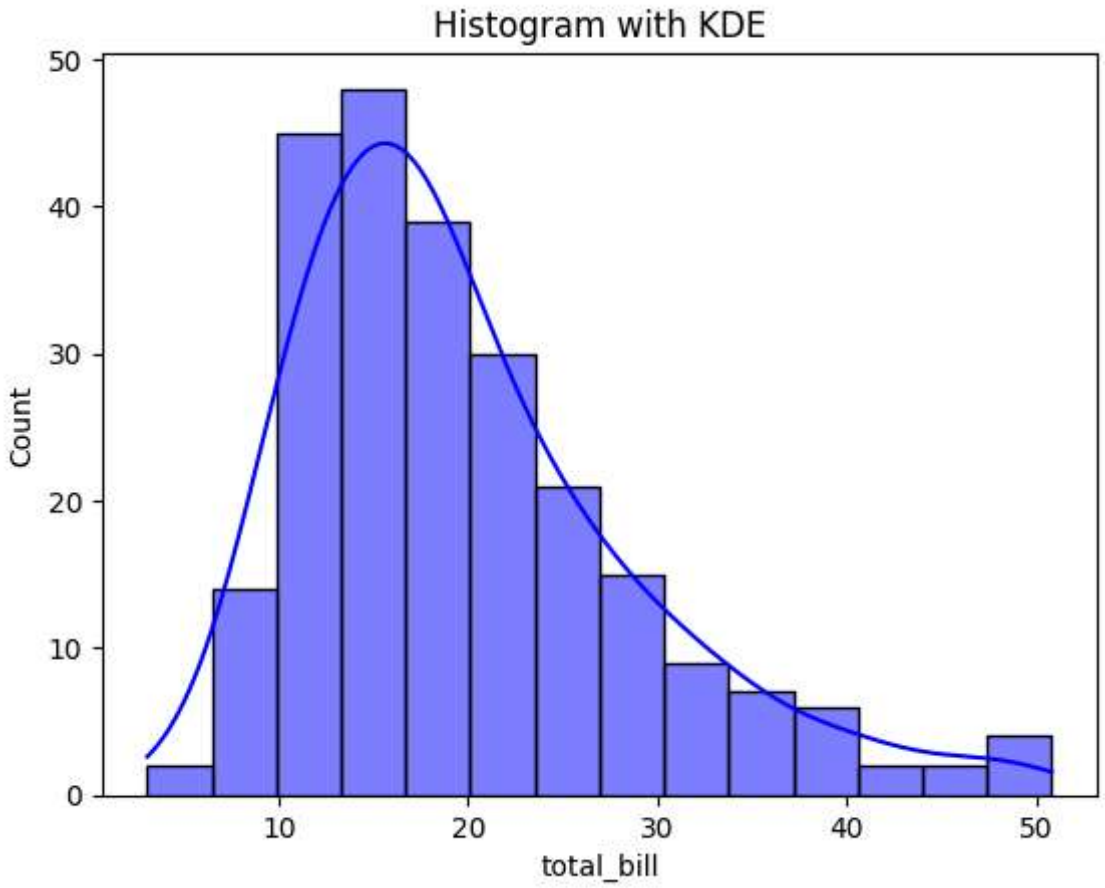


Biểu đồ phân phối:

```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")

sns.histplot(tips['total_bill'], kde=True, color='blue')
plt.title("Histogram with KDE")
plt.show()
```

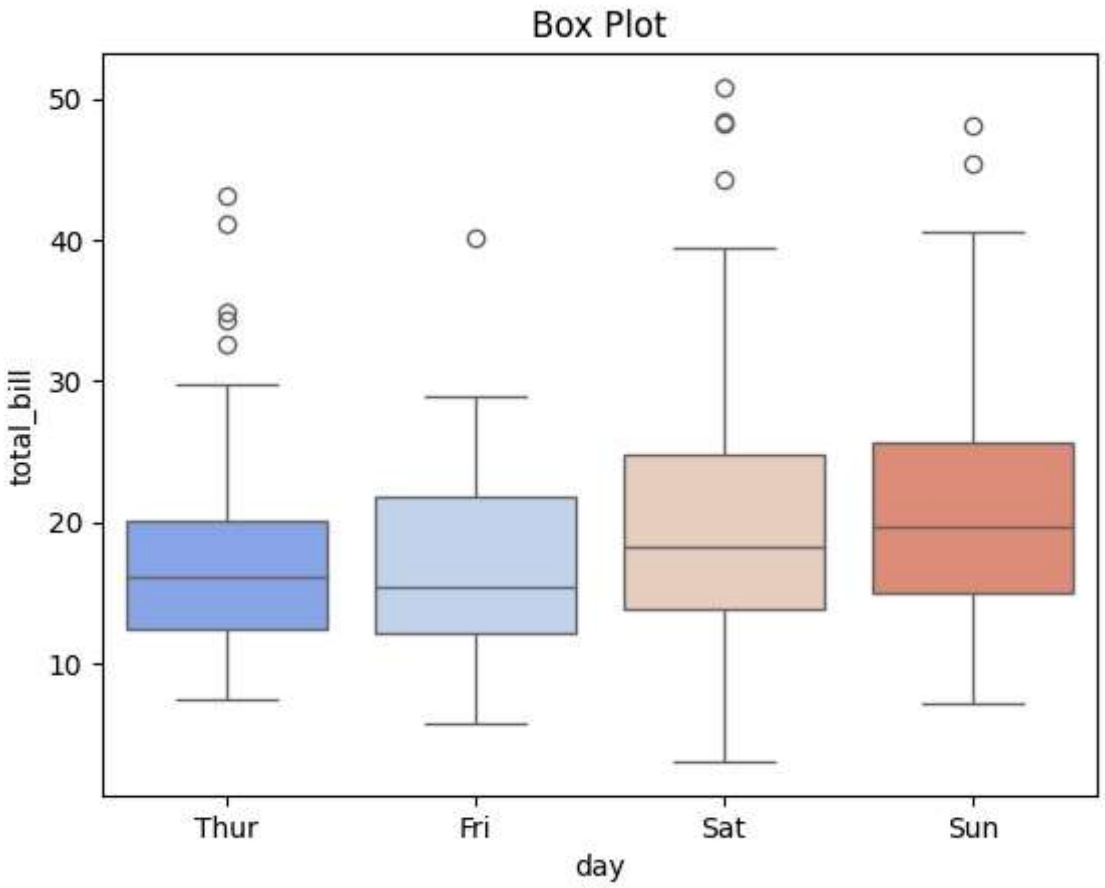


Chú thích:

- Biểu đồ hiển thị phân phối hóa đơn total_bill trong tập dữ liệu. Đường KDE cho thấy mật độ xác suất của dữ liệu.

6.2.2 Biểu đồ hộp (Box Plot) Dùng để hiển thị phân phối và phát hiện giá trị ngoại lai.

```
sns.boxplot(x="day", y="total_bill", data=tips, hue="day", palette="coolwarm", dodge=
plt.legend([], [], frameon=False)  # Ẩn legend nếu không cần
plt.title("Box Plot")
plt.show()
```

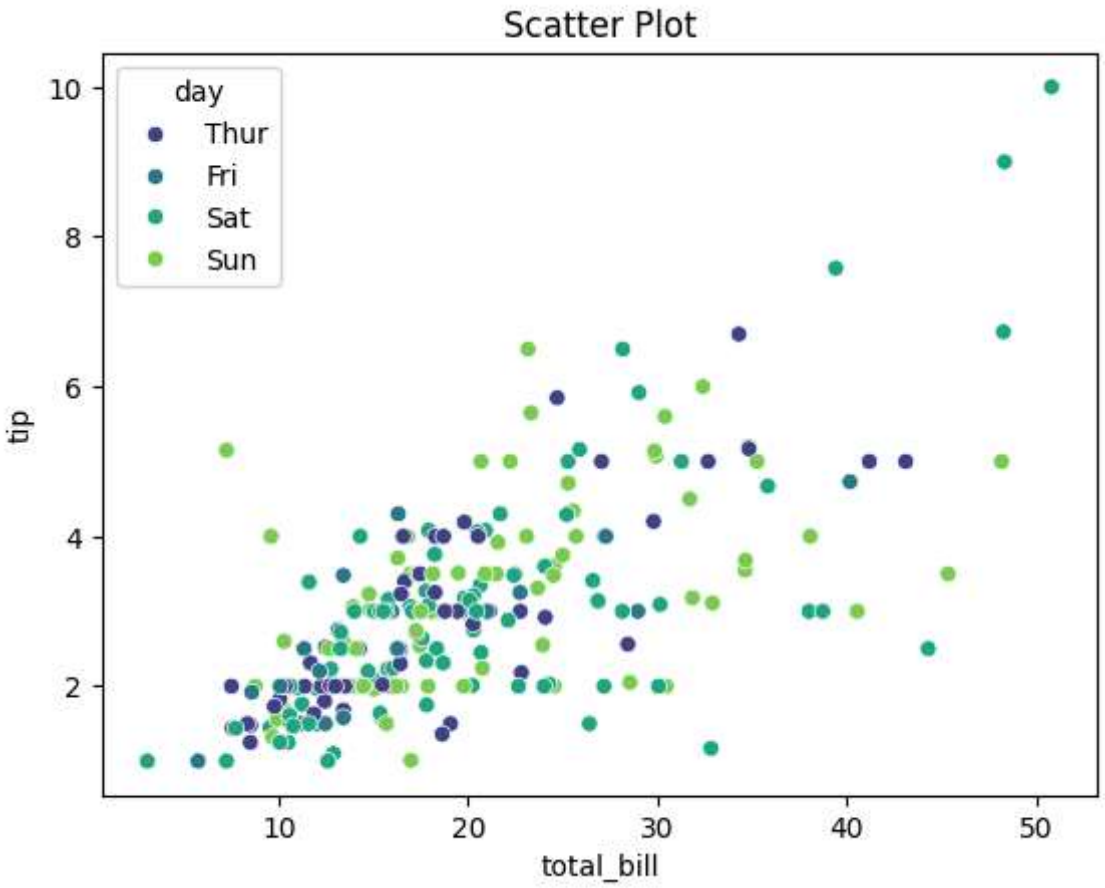


Chú thích:

- Biểu đồ cho thấy phân phối giá trị total_bill theo các ngày trong tuần (day). Các giá trị ngoài dải (outliers) được biểu thị bằng các dấu chấm.

6.2.3 Biểu đồ tán xạ (Scatter Plot) Dùng để hiển thị mối quan hệ giữa hai biến.

```
sns.scatterplot(x="total_bill", y="tip", hue="day", data=tips, palette="viridis")
plt.title("Scatter Plot")
plt.show()
```

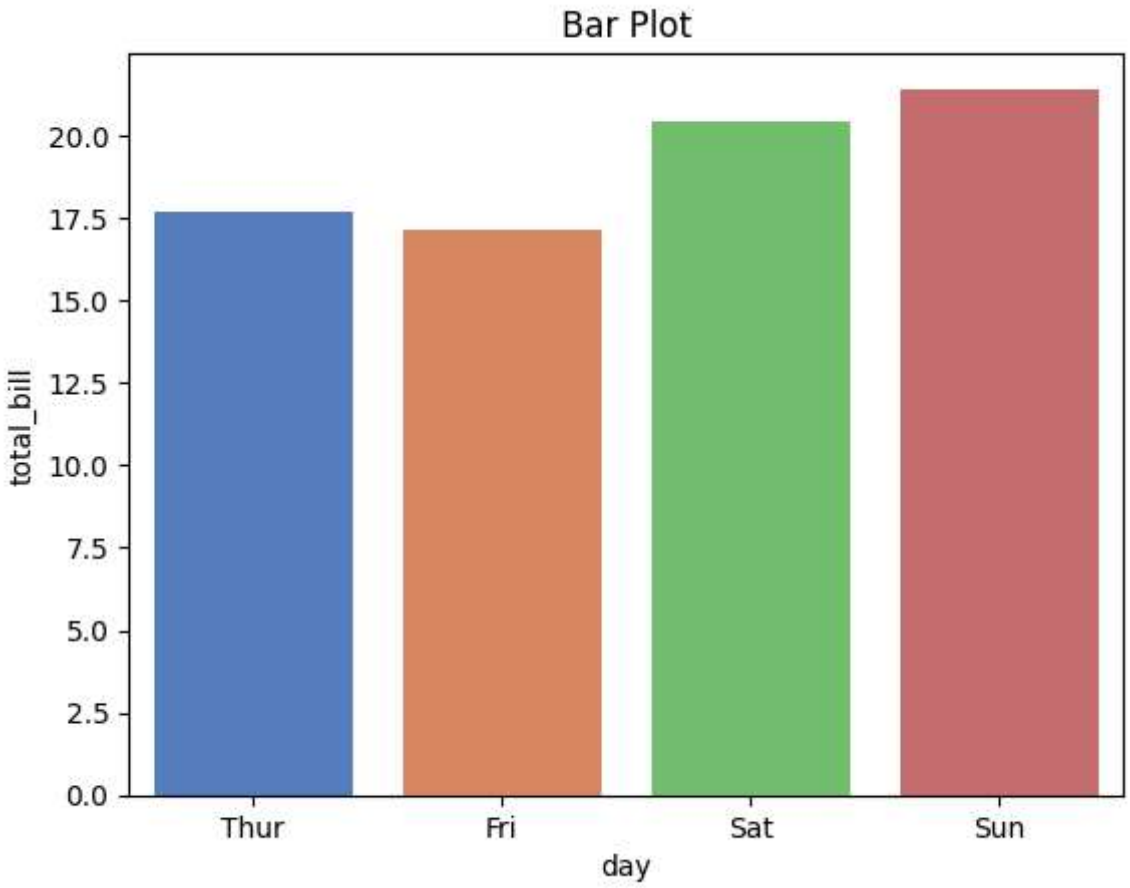


Chú thích:

- Biểu đồ này cho thấy mối quan hệ giữa tổng hóa đơn (total_bill) và tiền tip (tip) theo từng ngày. Các màu sắc khác nhau đại diện cho các ngày trong tuần.

6.2.4 Biểu đồ thanh (Bar Plot) Dùng để hiển thị giá trị trung bình của một biến theo nhóm.

```
sns.barplot(x="day", y="total_bill", data=tips, errorbar=None, palette="muted", hue="
plt.title("Bar Plot")
plt.show()
```



Chú thích:

- Biểu đồ này cho thấy tổng hóa đơn trung bình (total_bill) theo từng ngày trong tuần. Các thanh đại diện cho giá trị trung bình của từng nhóm.

6.3 Các biểu đồ nâng cao

6.3.1 Biểu đồ ma trận tương quan (Heatmap) Dùng để trực quan hóa mối tương quan giữa các biến số.

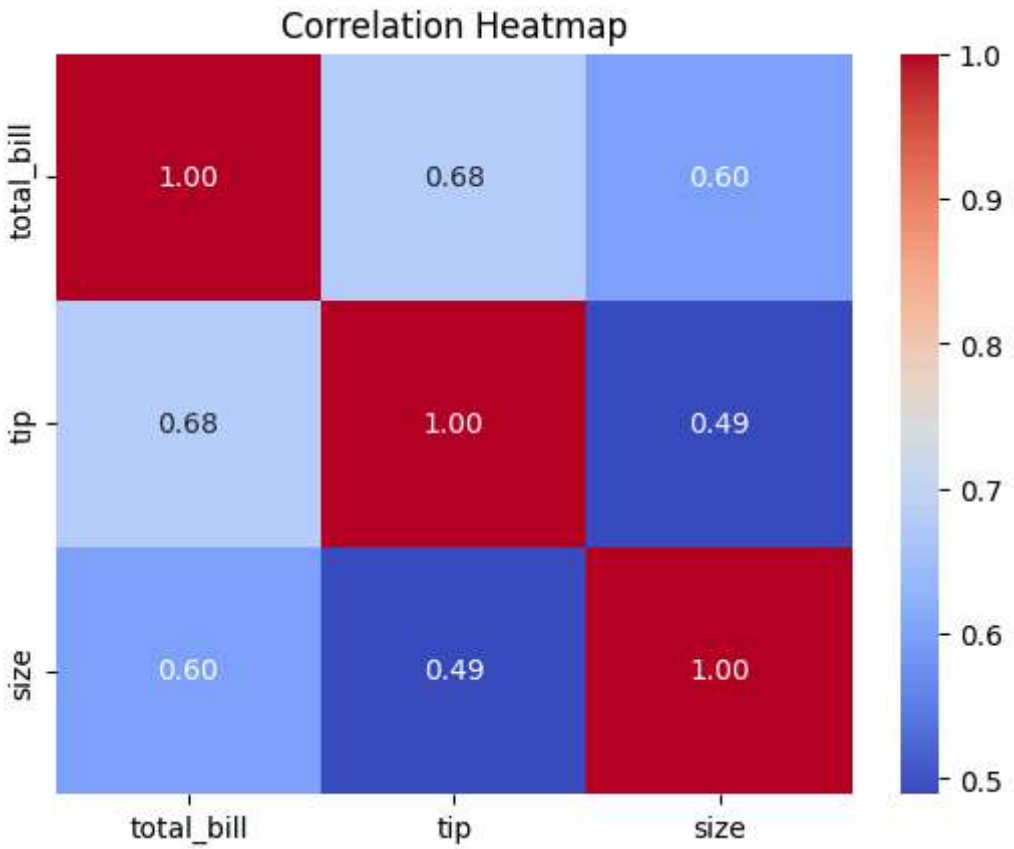
```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")

numeric_tips = tips.select_dtypes(include='number')

correlation = numeric_tips.corr()

sns.heatmap(correlation, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

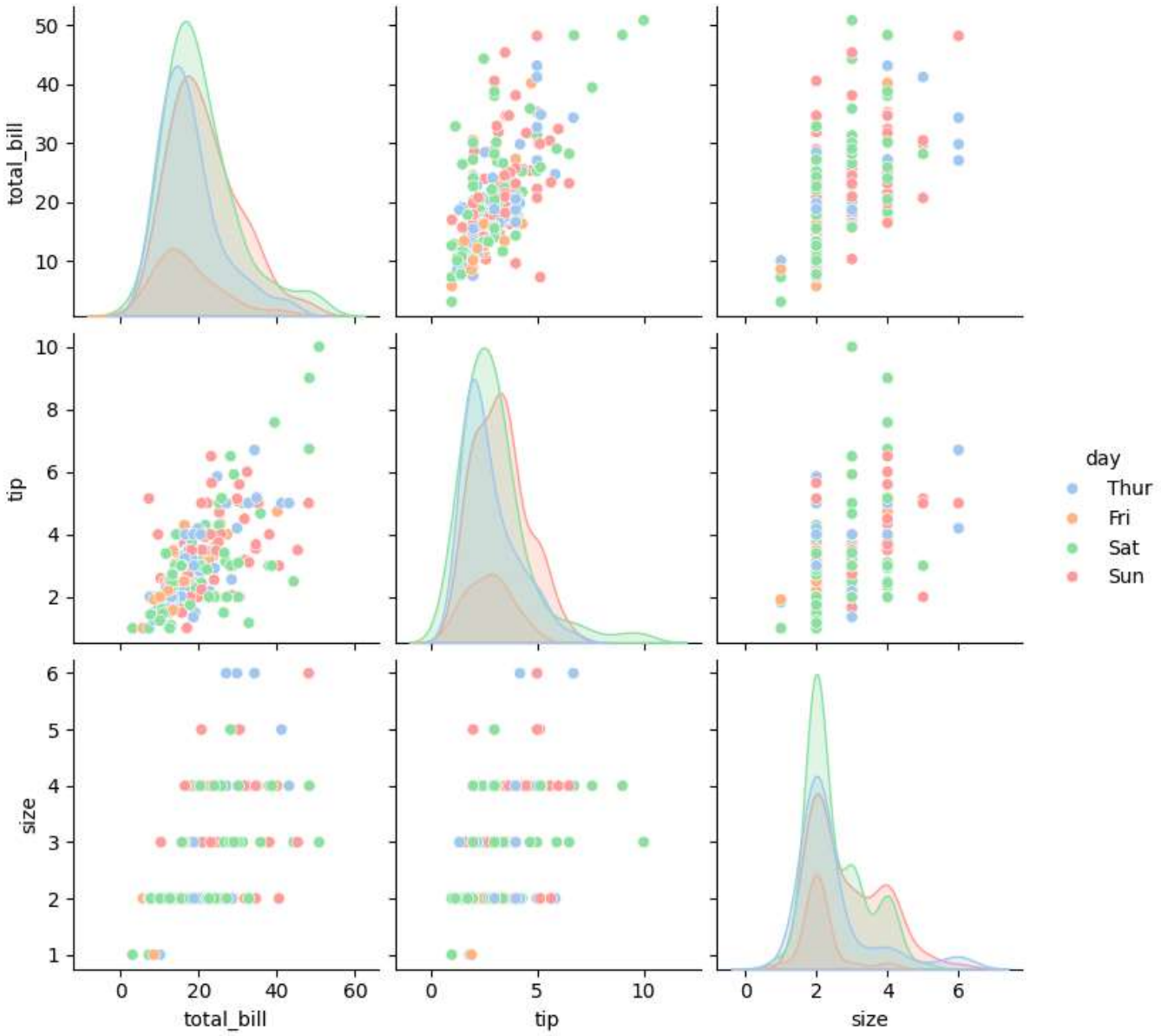


Chú thích:

- Biểu đồ này hiển thị mối quan hệ tuyến tính giữa các cặp biến số trong dữ liệu. Giá trị gần 1 hoặc -1 cho thấy mối tương quan mạnh, trong khi giá trị gần 0 cho thấy mối tương quan yếu.

6.3.2 Biểu đồ ghép (Pair Plot) Hiển thị mối quan hệ giữa tất cả các cặp biến.

```
sns.pairplot(tips, hue="day", palette="pastel")
plt.show()
```

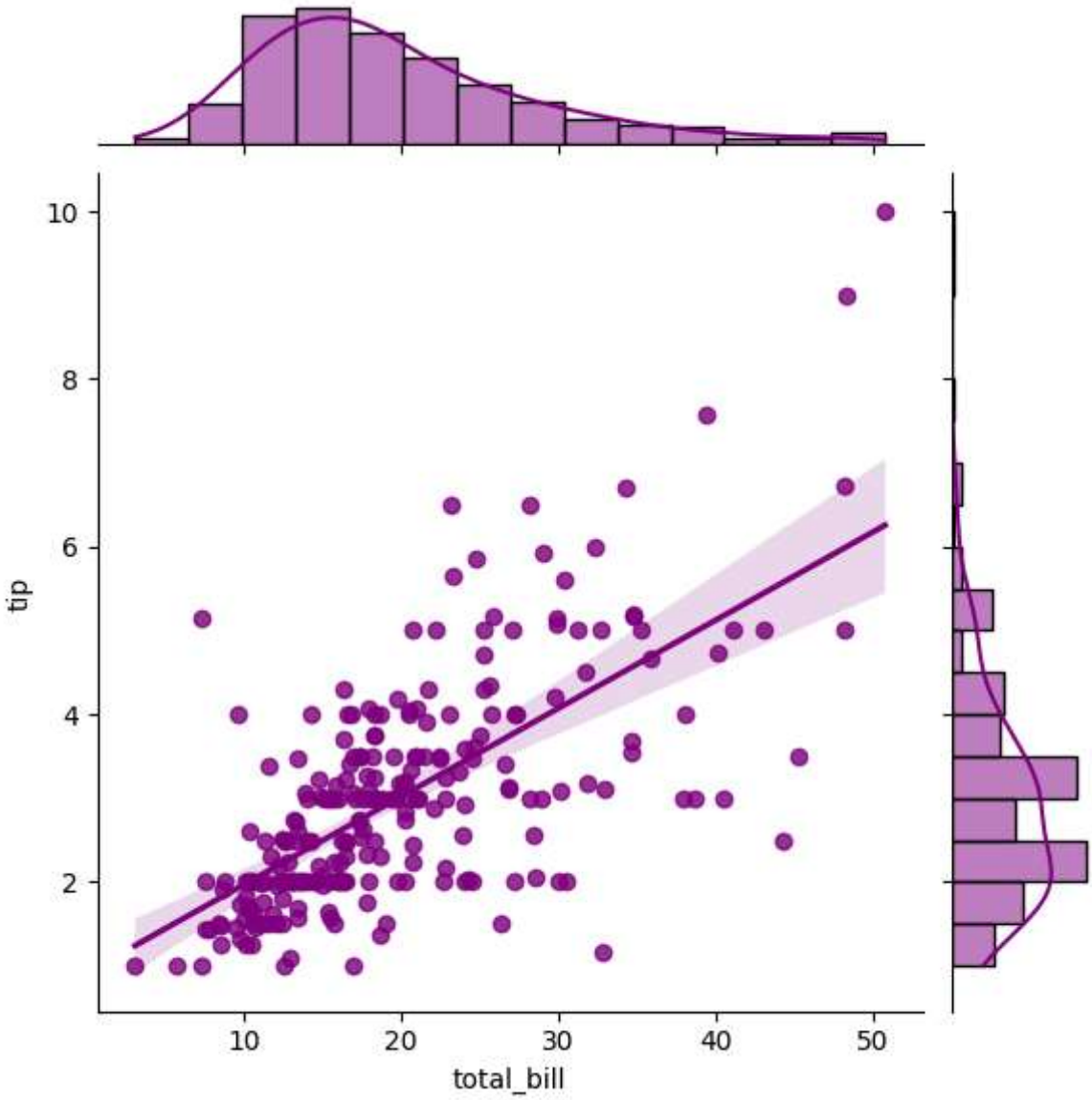


Chú thích:

- Mỗi ô trong biểu đồ đại diện cho một cặp biến số. Đường chéo chính chứa các biểu đồ phân phối, còn các ô khác là các biểu đồ tán xạ giữa hai biến.

6.3.3 Biểu đồ phân phối dạng khớp (Joint Plot) Hiển thị mối quan hệ giữa hai biến cùng với biểu đồ phân phối của chúng.

```
sns.jointplot(x="total_bill", y="tip", data=tips, kind="reg", color="purple")
plt.show()
```



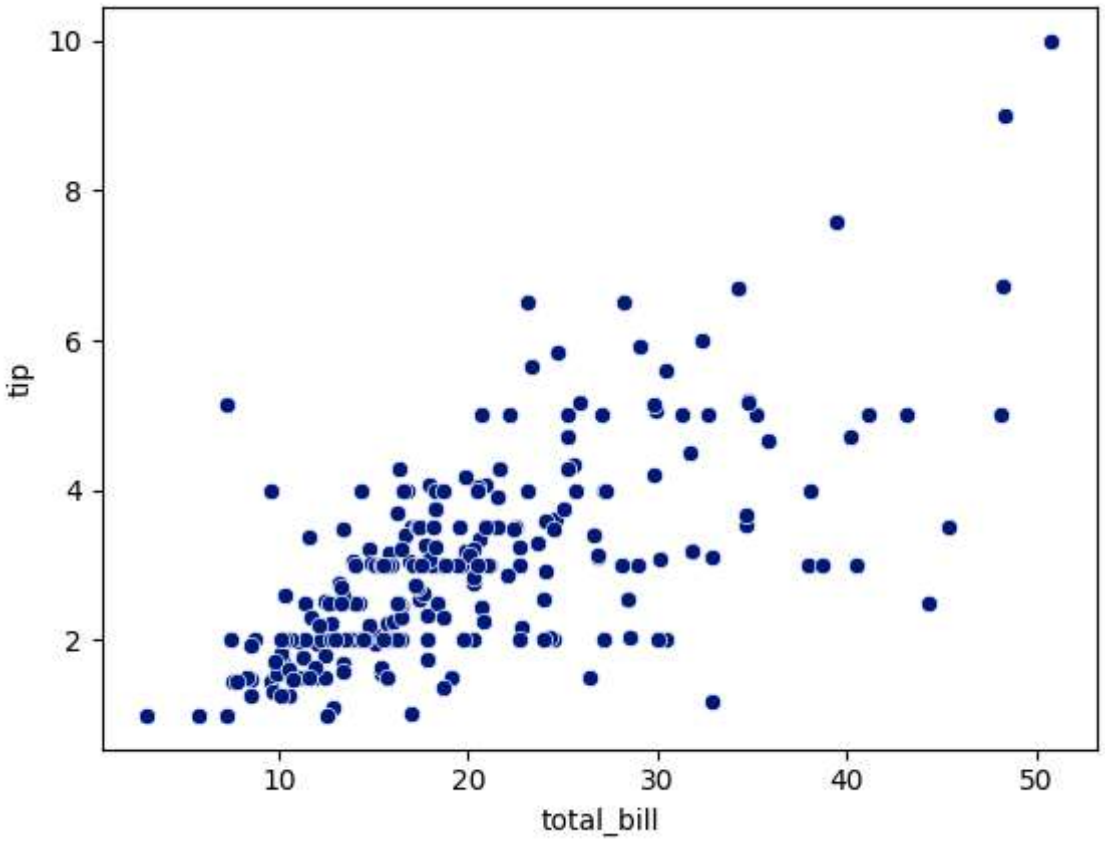
Chú thích:

- Biểu đồ này không chỉ hiển thị mối quan hệ giữa hai biến mà còn hiển thị các phân phối biên (marginal distributions). Đường hồi quy (regression) chỉ ra xu hướng tổng quát.

6.4 Tùy chỉnh và cài đặt giao diện

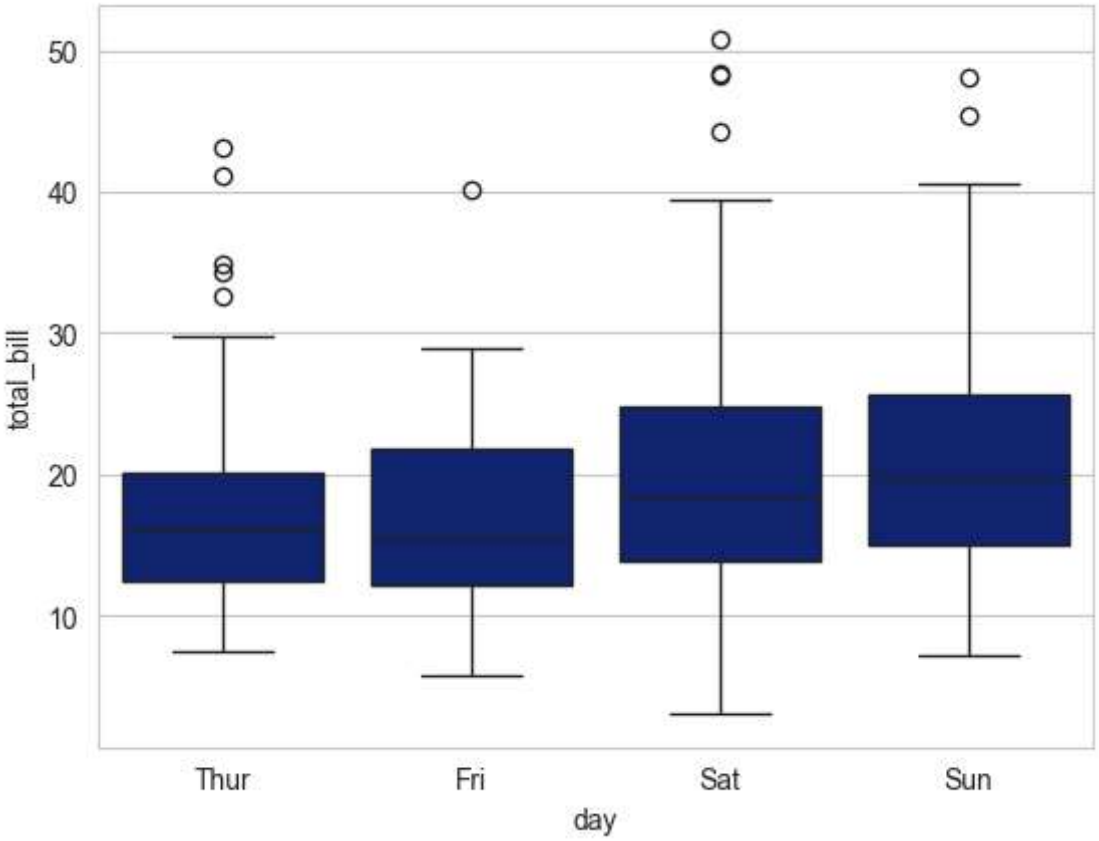
6.4.1 Tùy chỉnh màu sắc Seaborn cung cấp các bảng màu sẵn có.

```
sns.set_palette("dark")
sns.scatterplot(x="total_bill", y="tip", data=tips)
plt.show()
```



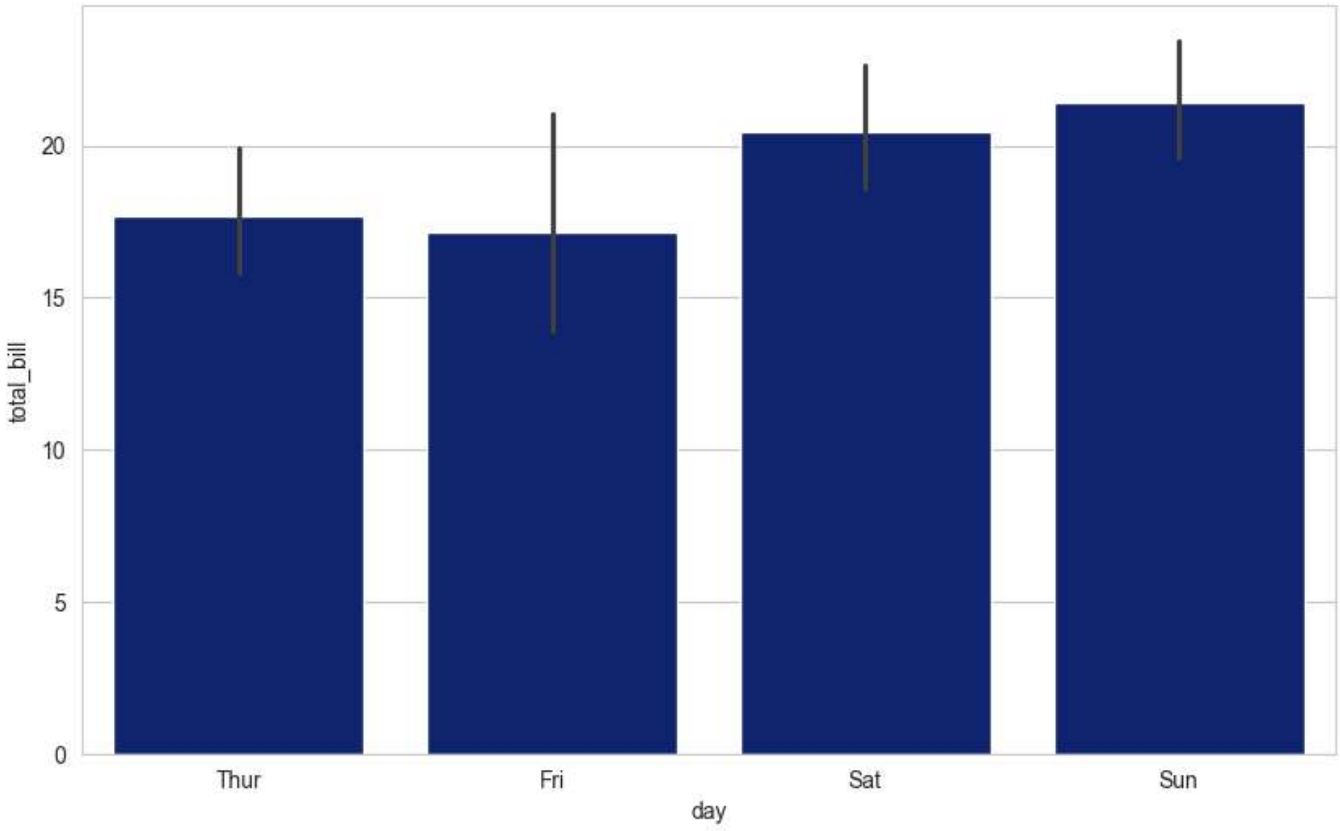
6.4.2 Thay đổi phong cách Bạn có thể thay đổi phong cách để biểu đồ trông chuyên nghiệp hơn.

```
sns.set_style("whitegrid")
sns.boxplot(x="day", y="total_bill", data=tips)
plt.show()
```



6.4.3 Điều chỉnh kích thước biểu đồ

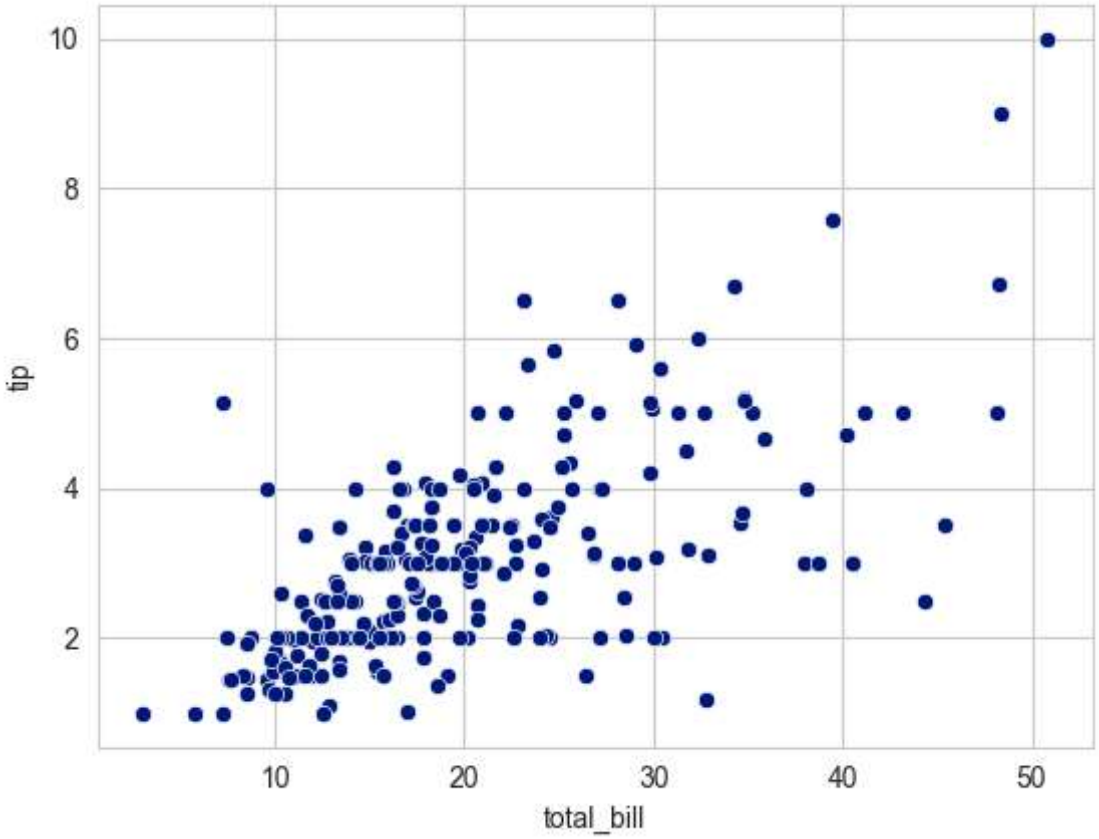
```
plt.figure(figsize=(10, 6))
sns.barplot(x="day", y="total_bill", data=tips)
plt.show()
```



6.5 Lưu biểu đồ

Seaborn hoạt động trên nền Matplotlib, nên bạn có thể lưu biểu đồ giống như Matplotlib.

```
sns.scatterplot(x="total_bill", y="tip", data=tips)
plt.savefig("scatter_plot.png", dpi=300, bbox_inches="tight")
plt.show()
```



6.6 Thực hành

Bài tập 1: Sử dụng tập dữ liệu tips để tạo:

- Biểu đồ phân phối với sns.histplot.
- Biểu đồ hộp nhóm theo day.

Bài tập 2: Tải dữ liệu mẫu iris và tạo:

- Biểu đồ ghép (Pair Plot) với sns.pairplot.
- Biểu đồ nhiệt mối tương quan giữa các biến.

Bài tập 3: Với một tập dữ liệu của riêng bạn, hãy:

- Tạo biểu đồ thanh hiển thị giá trị trung bình của một biến theo nhóm.
- Tùy chỉnh màu sắc và phong cách biểu đồ.

6.7 Tổng kết chương 6

Nội dung:Trong chương 6 đã cung cấp cách sử dụng Seaborn để trực quan hóa dữ liệu một cách đơn giản nhưng chuyên nghiệp. Trong chương 7 bạn sẽ học cách làm việc với Plotly để tạo các biểu đồ tương tác mạnh mẽ.