Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Bài 4: Trực quan hóa dữ liệu với Matplotlib và Seaborn

Phòng Lập Trình

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

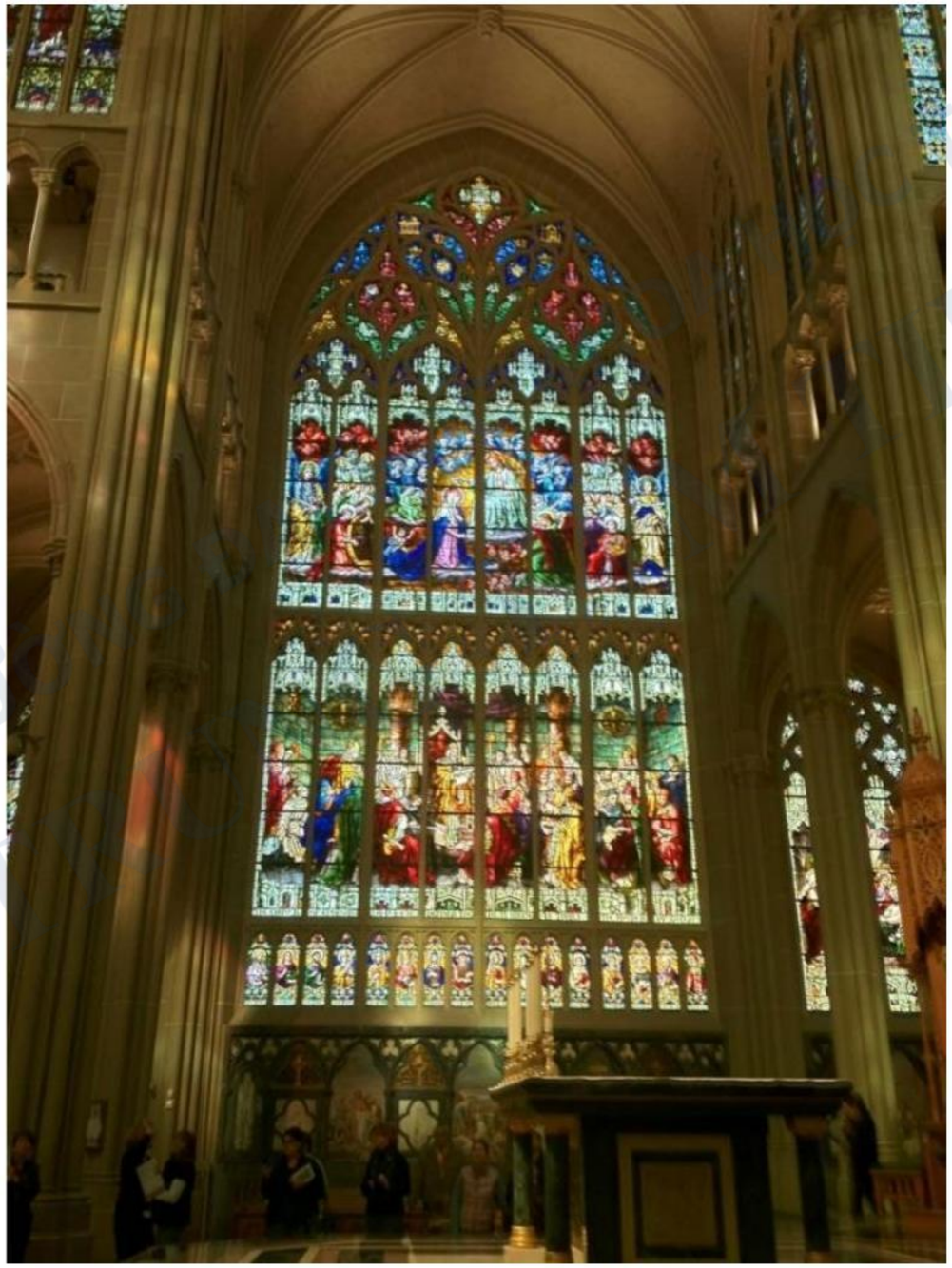
2023

Nội dung

1. Tổng quan
2. Trực quan hóa dữ liệu với Matplotlib
3. Trực quan hóa dữ liệu với Seaborn
4. Phân tích trực quan đa biến với lưới
5. Trực quan hóa dữ liệu văn bản

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Nội dung



Nội dung

	I		II		III		IV	
	X	Y	X	Y	X	Y	X	Y
	10	8.04	10	9.14	10	7.46	8	6.58
	8	6.95	8	8.14	8	6.77	8	5.76
	13	7.58	13	8.74	13	12.74	8	7.71
	9	8.81	9	8.77	9	7.11	8	8.84
	11	8.33	11	9.26	11	7.81	8	8.47
	14	9.96	14	8.1	14	8.84	8	7.04
	6	7.24	6	6.13	6	6.08	8	5.25
	4	4.26	4	3.1	4	5.38	19	12.5
	12	10.84	12	9.13	12	8.15	8	5.56
	7	4.82	7	7.26	7	6.42	8	7.91
	5	5.68	5	4.74	5	5.73	8	6.88
MEAN	9.00	7.5	9.00	7.5	9.00	7.5	9.00	7.5
STD	3.32	2.03	3.32	2.03	3.32	2.03	3.32	2.03
CORR	0.82		0.82		0.82		0.82	
LIN REG	y = 3.00 + 0.500x		y = 3.00 + 0.500x		y = 3.00 + 0.500x		y = 3.00 + 0.500x	

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Nội dung

- **Vai trò của trực quan hóa dữ liệu**
  - Giúp người xem thấy dữ liệu một cách trực quan và dễ hiểu.
  - Cung cấp cho người xem bức tranh toàn cảnh về dữ liệu.
  - Giúp dễ dàng phát hiện xu hướng, nhóm mẫu đặc trưng, các điểm ngoại lệ,...
  - Giúp phân tích, giải quyết vấn đề hiệu quả.
  - Giúp kể câu chuyện về dữ liệu.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

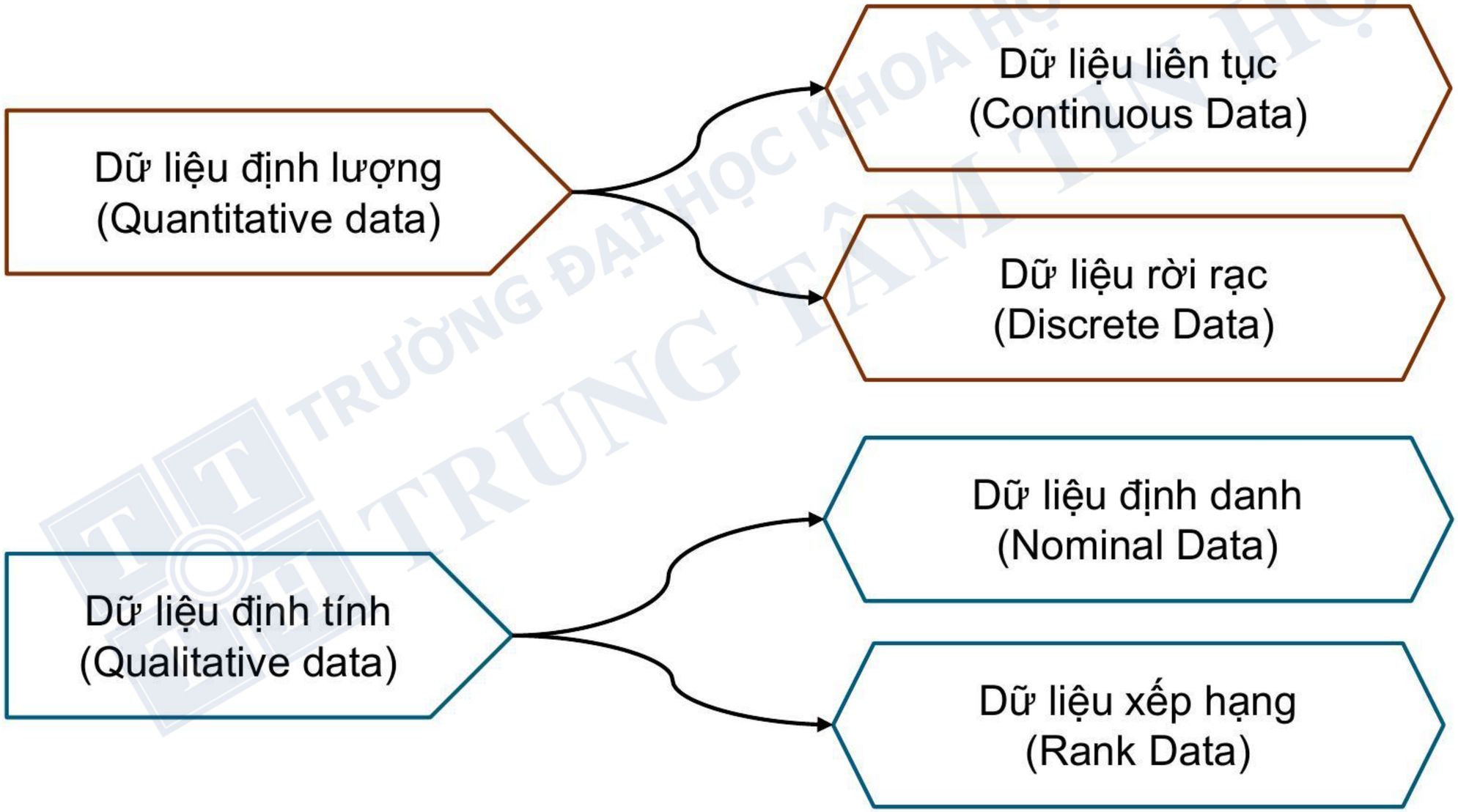
Nội dung

- **Các thành phần chung của biểu đồ**
  - Loại biểu đồ (bar, line, scatter, histogram, pie…)
  - Dữ liệu trên các trục (Axis data ranges)
  - Nhãn trên trục (Axis labels)
  - Ghi chú (Legend)
  - Các chú thích (Annotations)

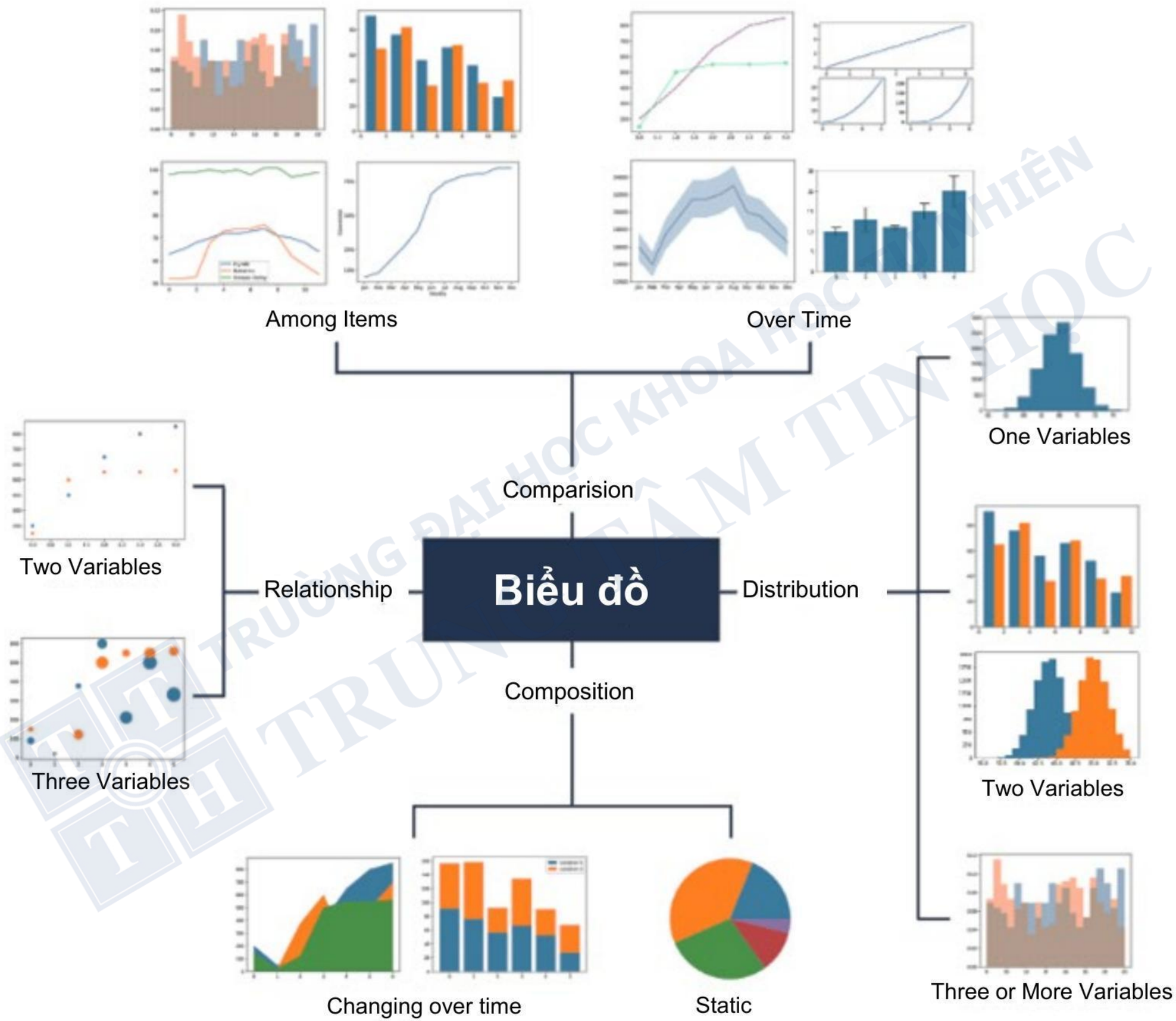
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Tổng quan

- Tính chất dữ liệu giúp chọn biểu đồ trực quan hóa dữ liệu phù hợp
- Phân loại dữ liệu



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Tổng quan

Một số thư viện trực quan hóa dữ liệu trên Python

matplotlib

seaborn

plotly

...

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Nội dung

1. Tổng quan
2. Trực quan hóa dữ liệu với Matplotlib
3. Trực quan hóa dữ liệu với Seaborn
4. Phân tích trực quan đa biến với lưới
5. Trực quan hóa dữ liệu văn bản

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Matplotlib**
  - Thư viện trực quan hóa dữ liệu phổ biến của Python
  - Sử dụng để vẽ biểu đồ trong phân tích khám phá dữ liệu (EDA)
  - Dữ liệu nguồn là dữ liệu dạng danh sách: list, numpy array, set, dictionary, series,...
  - Có thể tùy chỉnh line style, font, các thuộc tính trên trục…
  - Là thư viện trực quan hóa dữ liệu thường được dùng trong Data Science, Machine Learning.
- **Cài đặt**

```
pip install matplotlib
```

- **Sử dụng**

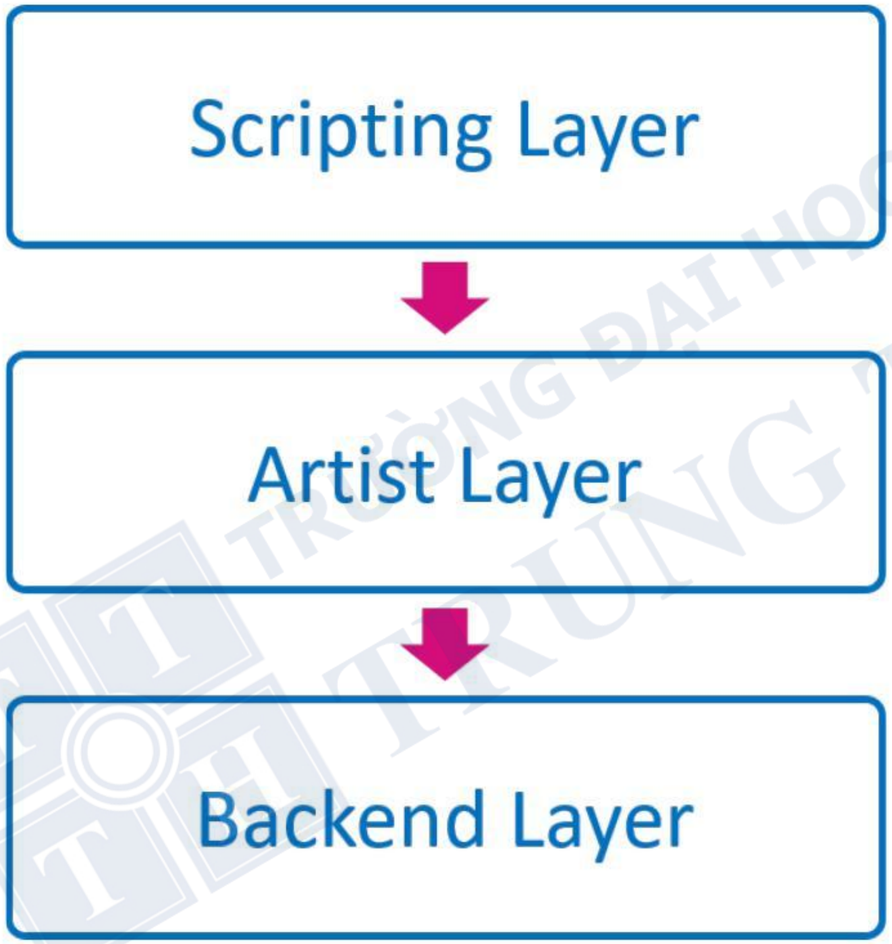
```
import matplotlib.pyplot as plt
```

https://matplotlib.org/

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

❑ Kiến trúc Matplotlib



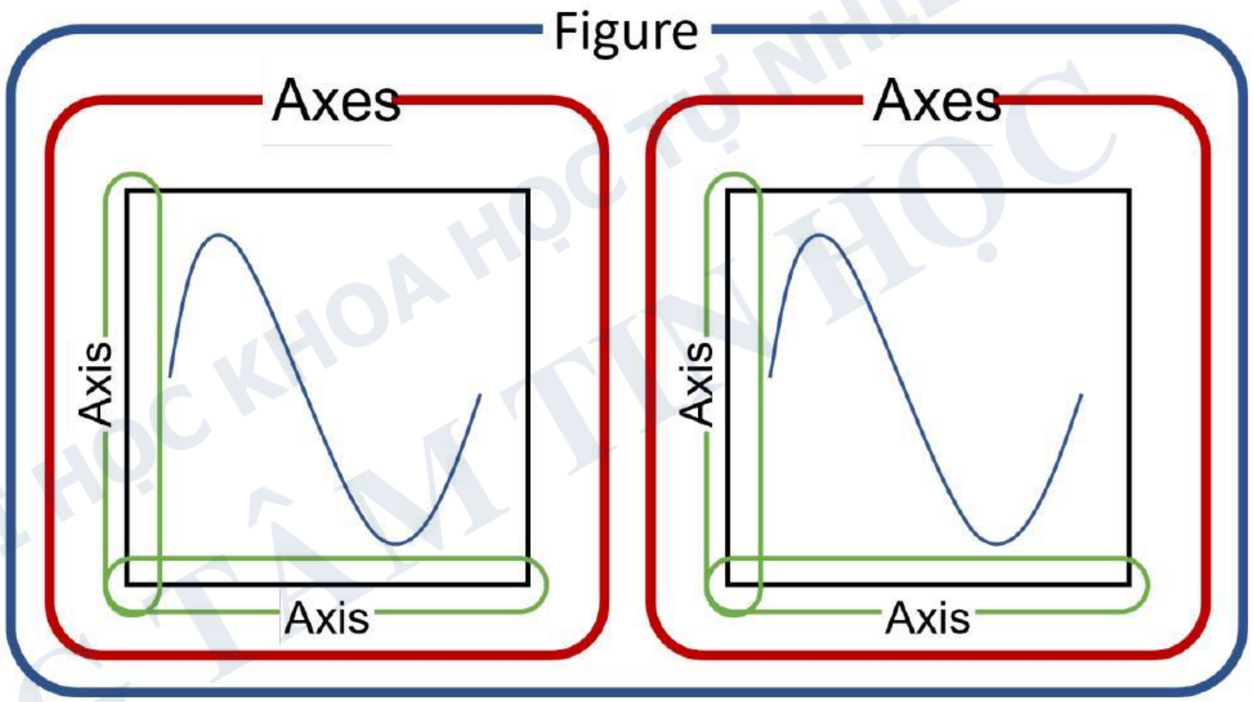
- Gọi trực tiếp các lệnh qua **pyplot module**
- Thực hiện các thay đổi trên Figure và Axes hiện hành

- Sử dụng các biến đối tượng một cách tường minh
- Mỗi biến sẽ giúp thay đổi một số thuộc tính tương ứng của biểu đồ

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Cấu trúc của biểu đồ**
  - Cây phân cấp đối tượng
  - Figure và Axes là 2 đối tượng chính
  - Figure: không gian thể hiện, cho phép trên đó có nhiều biểu đồ.
  - Axes: biểu đồ thật sự với các thông tin: trục x-y, ghi chú,...
  - Tạo figure, axes




```
fig = plt.figure()
ax = plt.axes()
```

Trực quan hóa dữ liệu với Matplotlib

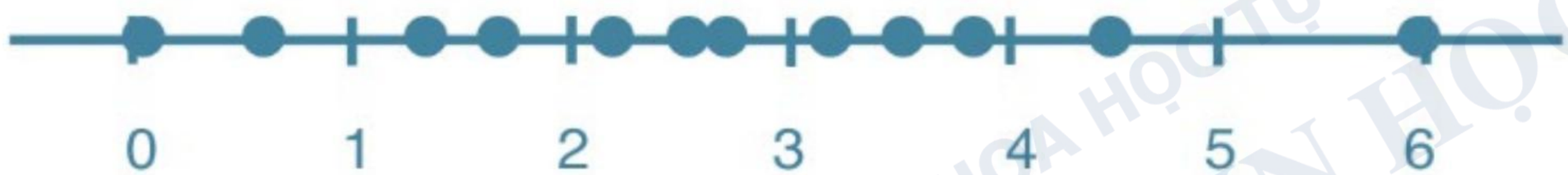
```python
import matplotlib.pyplot as plt
```

Thay đổi kích thước biểu đồ	`plt.figure(figsize=(<width>,<height>))`
Tạo tiêu đề cho biểu đồ	`plt.title(<title>)`
Tạo tiêu đề cho trục x, y	`plt.xlabel(<title>) /` `plt.ylabel(<title>)`
Điều chỉnh vị trí chú thích	`plt.legend([location])`
Hiển thị biểu đồ	`plt.show()`

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- Histogram



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Histogram**
  - Dùng để **phân tích đơn biến**
  - Là dạng biểu đồ biểu diễn dạng phân phối tần suất (frequency distribution table) của dữ liệu định lượng.
  - Cho biết:
    - Kiểu phân bố dữ liệu
    - Độ rộng của dữ liệu
    - Dữ liệu có bị lệch không ?
    - Dữ liệu có giá trị outlier không ?
  - bins: Số lượng khoảng chia của dữ liệu (t*hay đổi giá trị bins sẽ ảnh hưởng đến hình dạng của biểu đồ)*

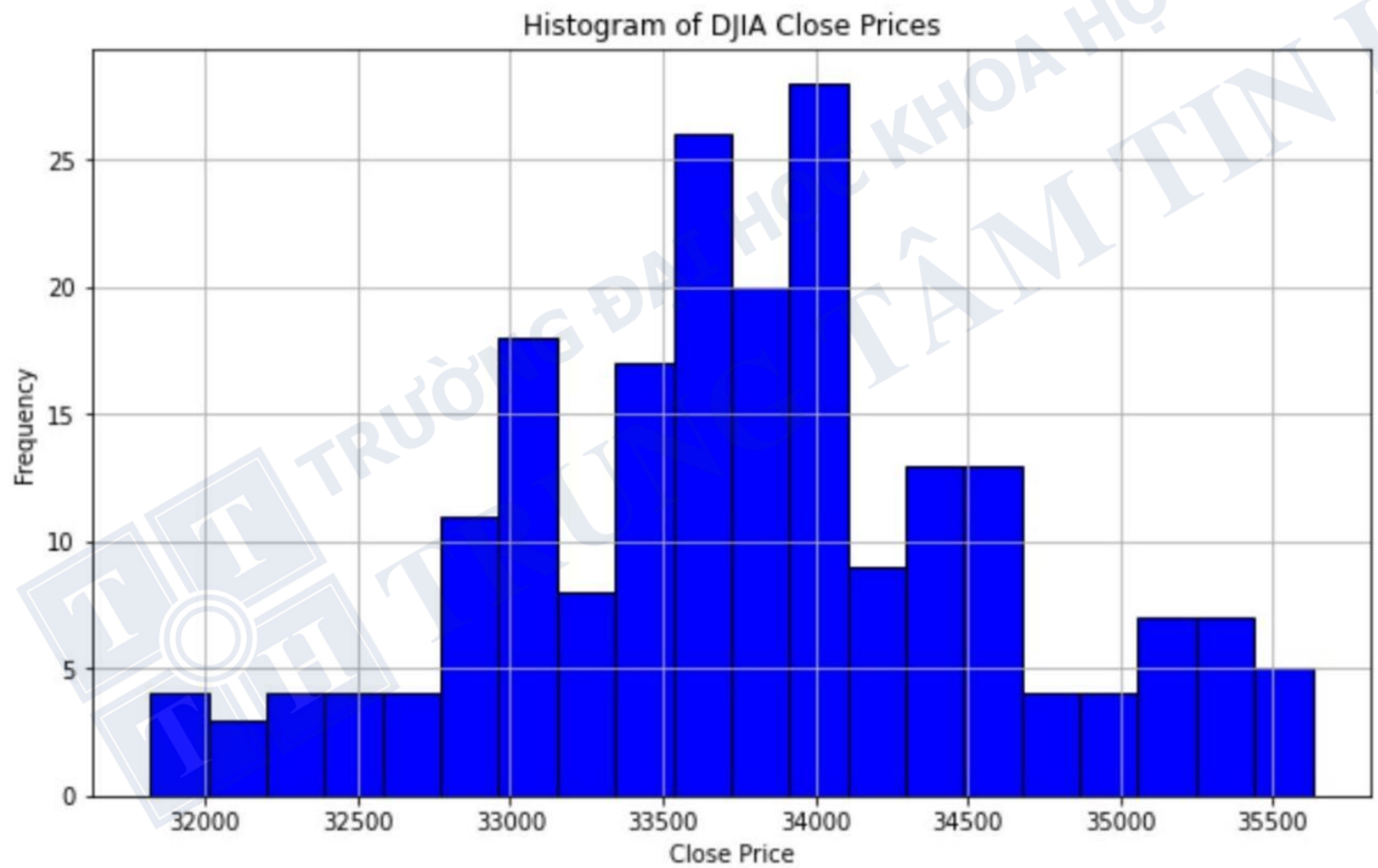
Ví dụ: Hãy quan sát và nhận xét đồ thị histogram về phổ điểm thi tốt nghiệp THPT năm 2023 trên báo Tuổi Trẻ.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Histogram** `plt.hist(x,[bins])`

```
plt.hist(djia['Close'],bins=20,color='blue')
```

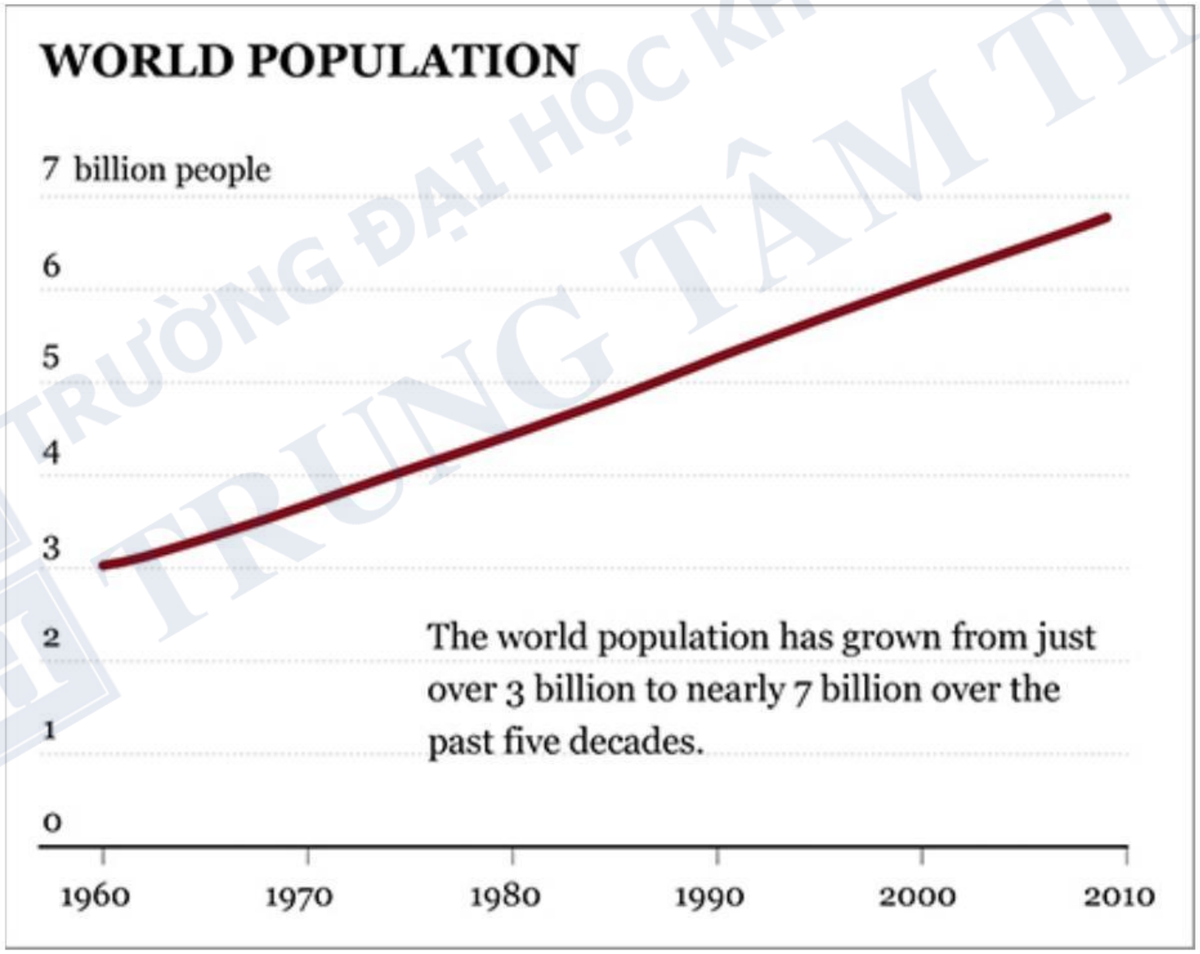


HV thử thay đổi giá trị `bins` khác nhau và quan sát kết quả.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Line plot**
  - Thường được dùng để phân tích xu hướng của dữ liệu theo thời gian
  - Ví dụ: hiển thị chỉ số chứng khoán theo ngày, số lượng vé máy bay bán hàng ngày, số lượng sản phẩm tiêu thụ theo quý ...



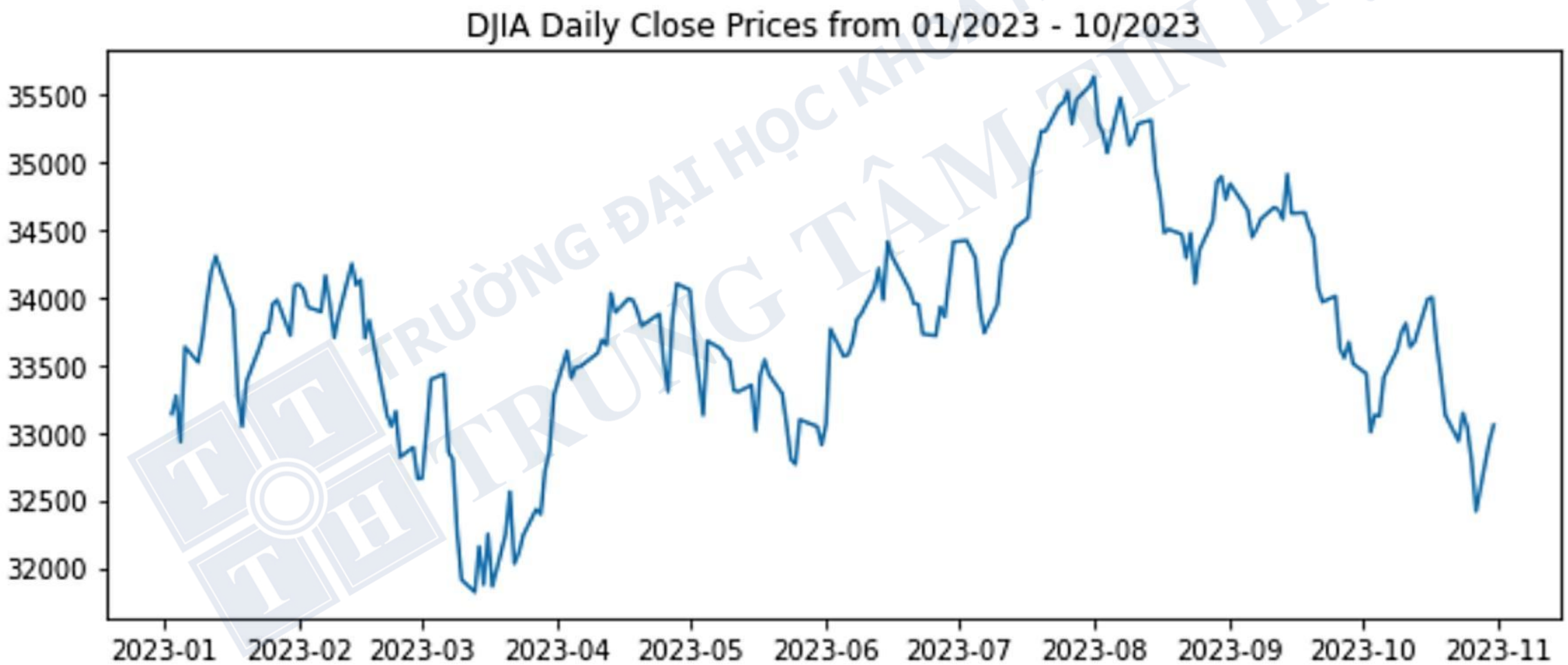
Visualize This: The FlowingData Guide to Design, Visualization, and Statistics

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Line plot** `plt.plot(x, y)`

```
plt.plot(djia['Date'], djia['Close'])
```

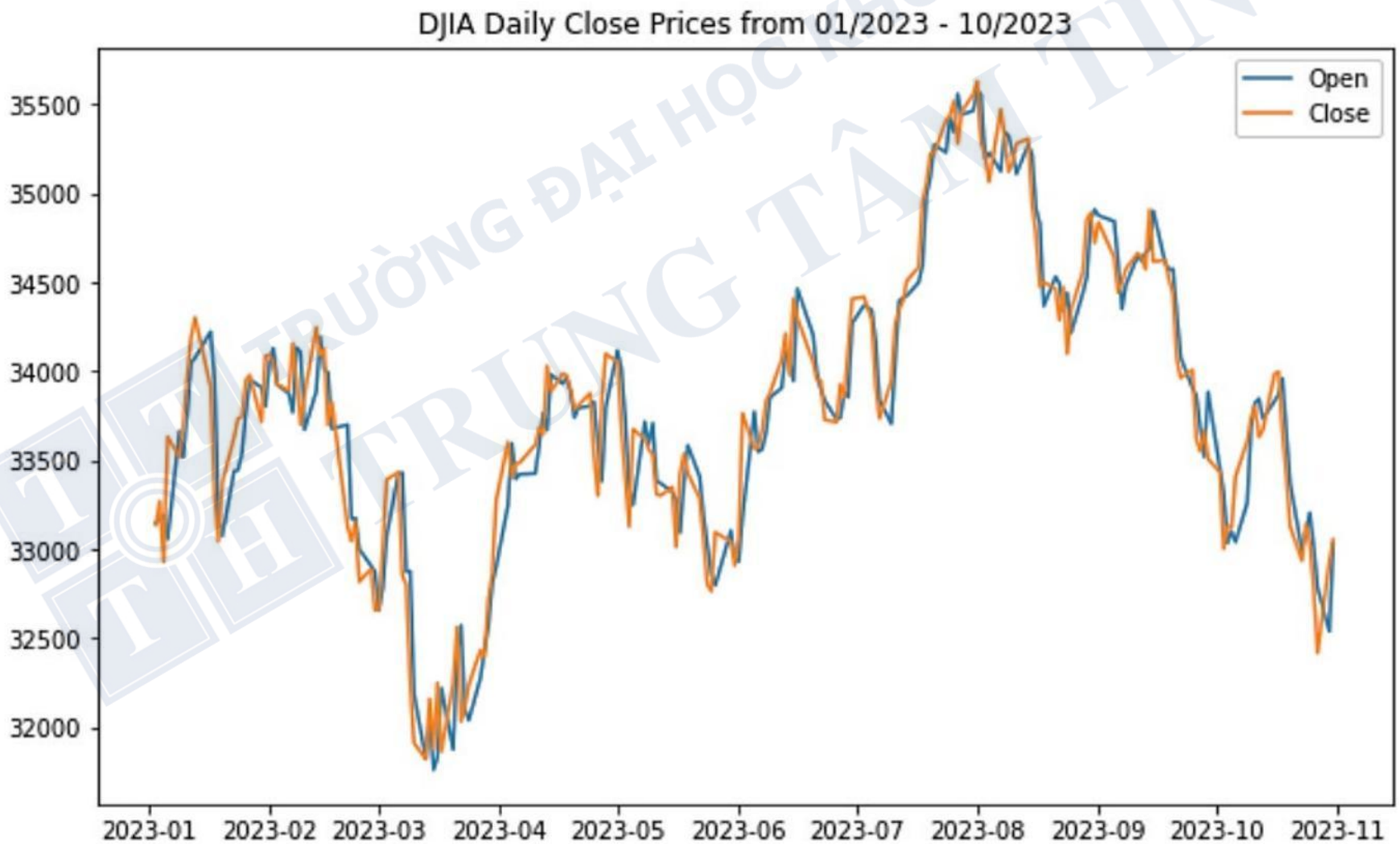


TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Line plot**
  - Vẽ nhiều biểu đồ Line trên cùng biểu đồ

```
plt.plot(djia['Date'], djia['Open'], label='Open')
plt.plot(djia['Date'], djia['Close'], label='Close')
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

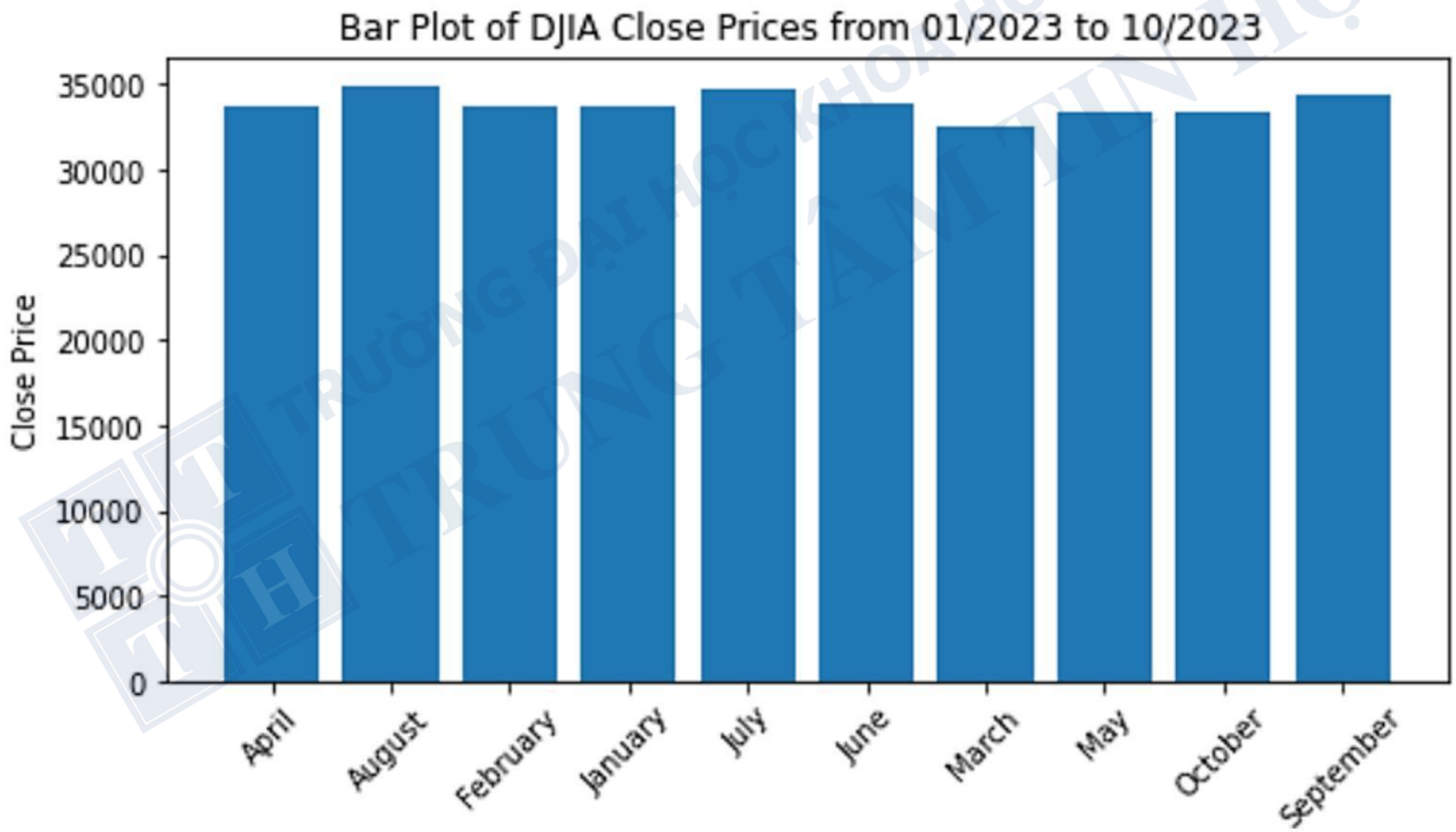
- **Barplot**
  - Là dạng biểu đồ biểu diễn dữ liệu dưới dạng các khối chữ nhật, giúp so sánh giá trị tính toán giữa các biến phân loại khác nhau.
  - Một trục là biến phân loại, một trục là giá trị tính toán của biến phân loại tương ứng.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Barplot** `plt.bar(x, height)`

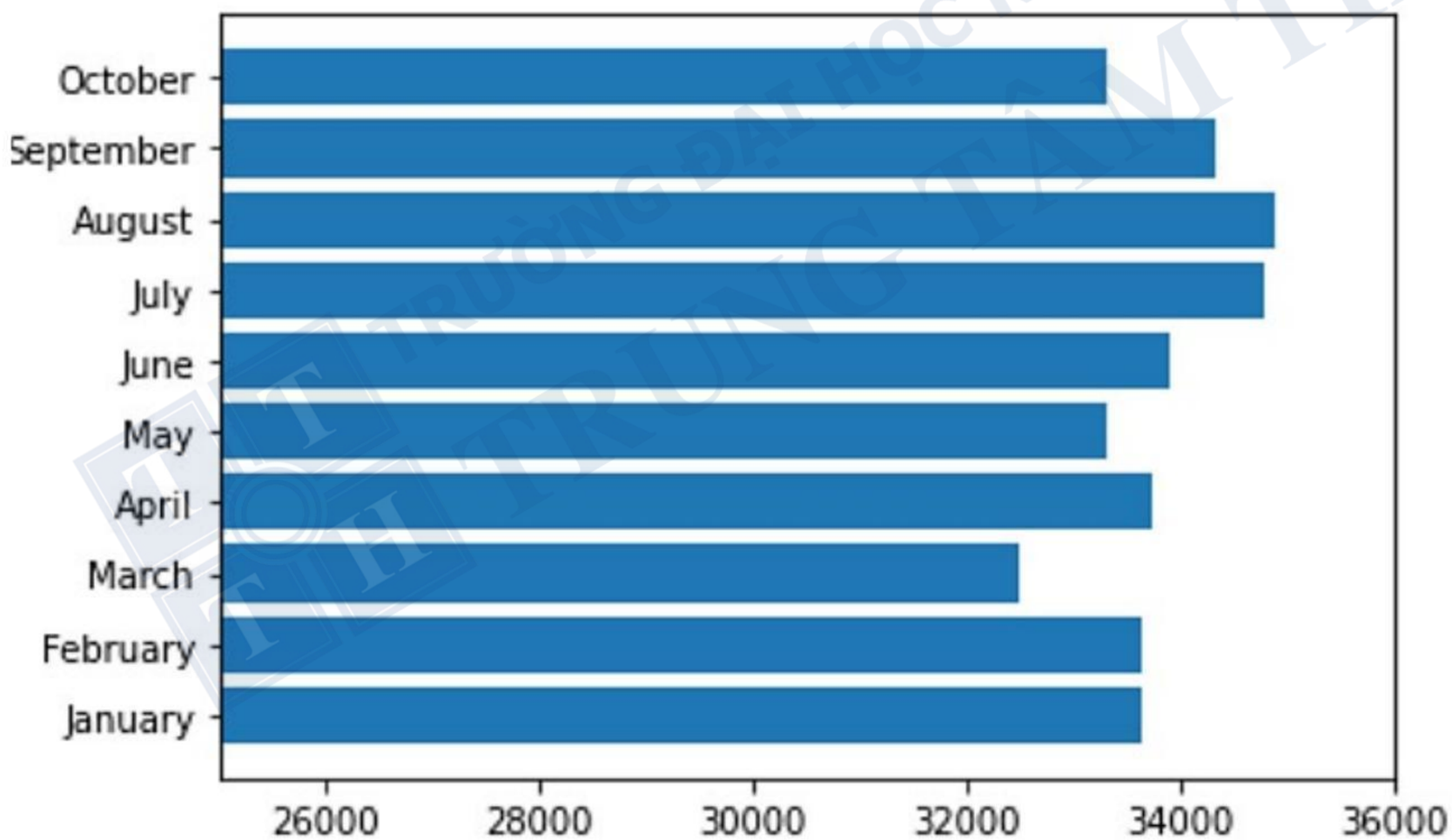
```
plt.bar(djia_monthly_mean['Month'], height = djia_monthly_mean['Close'])
```



Trực quan hóa dữ liệu với Matplotlib

- **Barplot xoay ngang** `plt.barh(y, width)`

```
plt.barh(djia_monthly_mean['Month'], width = djia_monthly_mean['Close'])
plt.xlim(25000,36000)
plt.show()
```

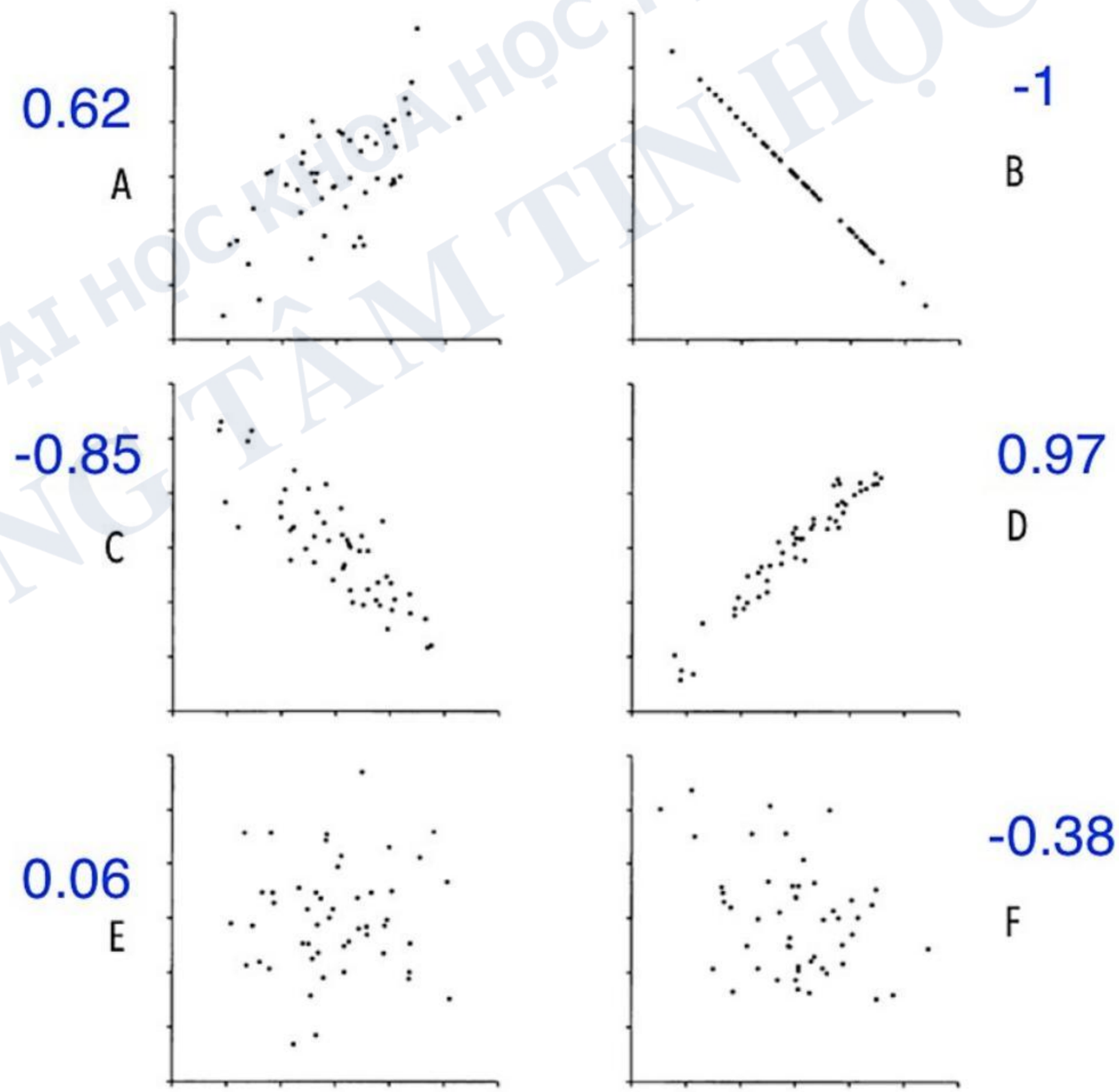


TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Scatter plot**
  - Hiển thị mối quan hệ giữa 2 biến liên tục (Correlation).

r value	Correlation
0.0 – 0.2	Tương quan yếu
0.3 – 0.6	Tương quan vừa
0.7 – 1.0	Tương quan mạnh

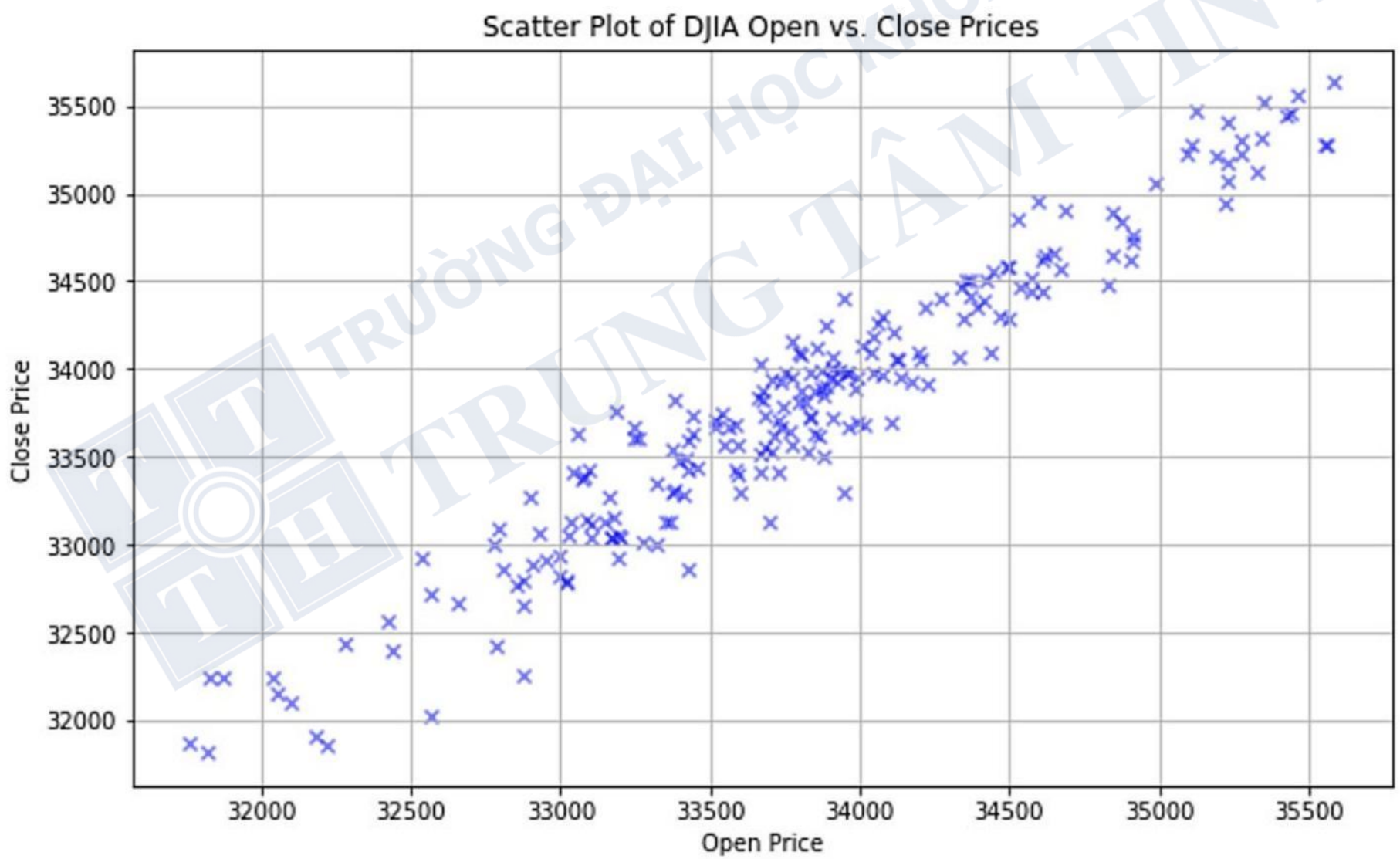


TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

- **Scatter plot** `plt.scatter(x, y,[<format>])`

```
plt.scatter(djia['Open'], djia['Close'],
            alpha=0.5,color='blue',marker='x')
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Matplotlib

Thay đổi giá trị đơn vị và góc hiển thị trục x	`plt.xticks(rotation=<degree>)`
Thay đổi giá trị đơn vị và góc hiển thị trục y	`plt.yticks(rotation=<degree>)`
Xác định khoảng giá trị trục x	`plt.xlim([<min>,<max>])`
Xác định khoảng giá trị trục y	`plt.ylim([<min>,<max>])`
Tắt hiển thị trục x, y	`plt.axis('off')`

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Nội dung

1. Tổng quan
2. Trực quan hóa dữ liệu với Matplotlib
3. Trực quan hóa dữ liệu với Seaborn
4. Phân tích trực quan đa biến với lưới
5. Trực quan hóa dữ liệu văn bản - Word Cloud

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Seaborn**
  - Được xây dựng dựa trên thư viện Matplotlib.
  - Là phần bổ sung, không phải thay thế Matplotlib.
  - Tích hợp chặt với các cấu trúc dữ liệu Dataframe của Pandas.
  - Có sẵn nhiều bộ dữ liệu mẫu, đã được xử lý (iris, tips,...)
  - Có nhiều dạng biểu đồ, nhiều mẫu định dạng sẵn, đẹp mắt.
  - Giúp trực quan hóa dữ liệu đơn biến, hai hay nhiều biến một cách dễ dàng.
- **Cài đặt**

```
pip install seaborn
```

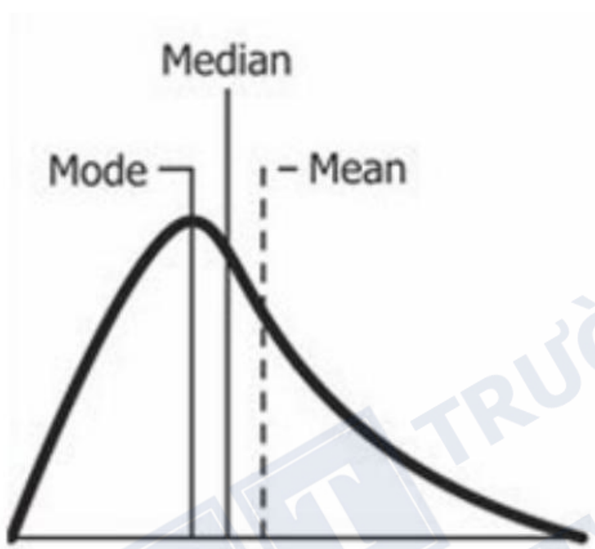
- **Sử dụng**

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

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

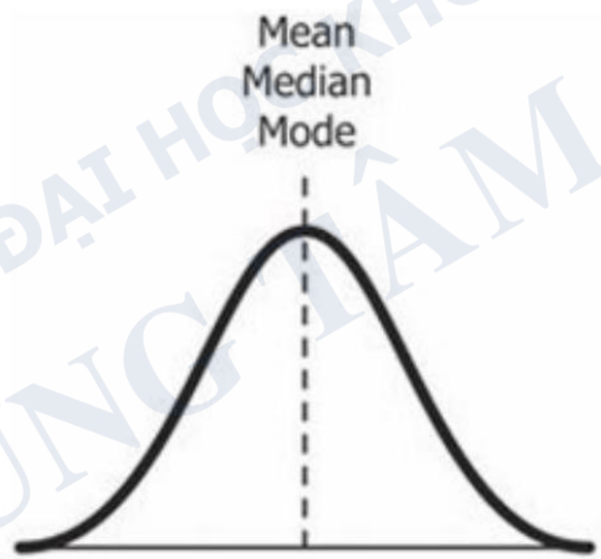
Trực quan hóa dữ liệu với Seaborn

- **Density plot - Đồ thị hàm mật độ xác suất**
  - Tương tự như biểu đồ Histogram nhưng là một phiên bản được làm mịn của biểu đồ histogram, thường dùng phương pháp KDE để làm mịn.



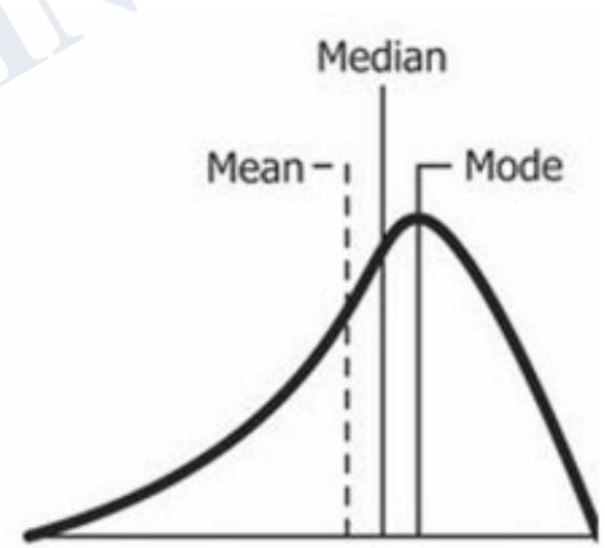
mean > median > mode

Lệch phải



mean = median = mode

Đối xứng



mean < median < mode

Lệch trái

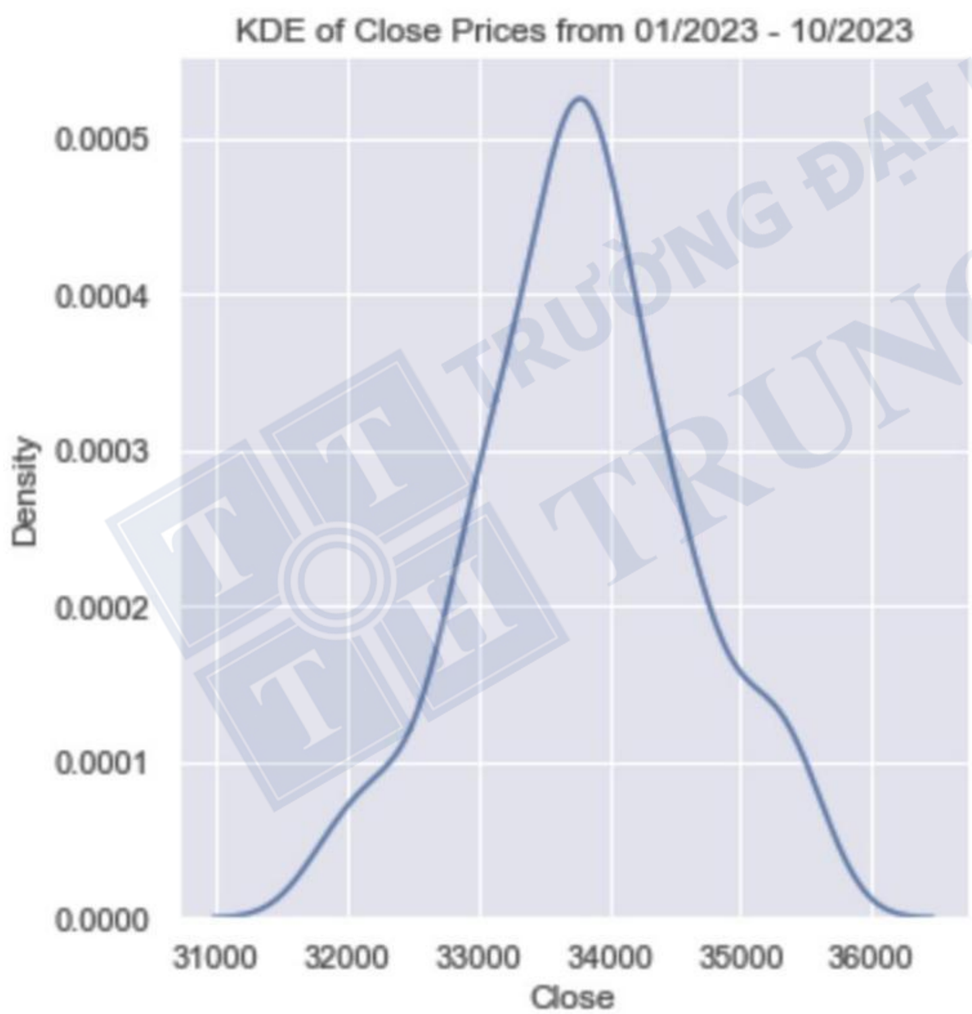
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TT TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Density plot - Đồ thị hàm mật độ xác suất**

```
sns.displot(data = <dataframe>,x=<column_name>,
            kind=<value>,...)
```

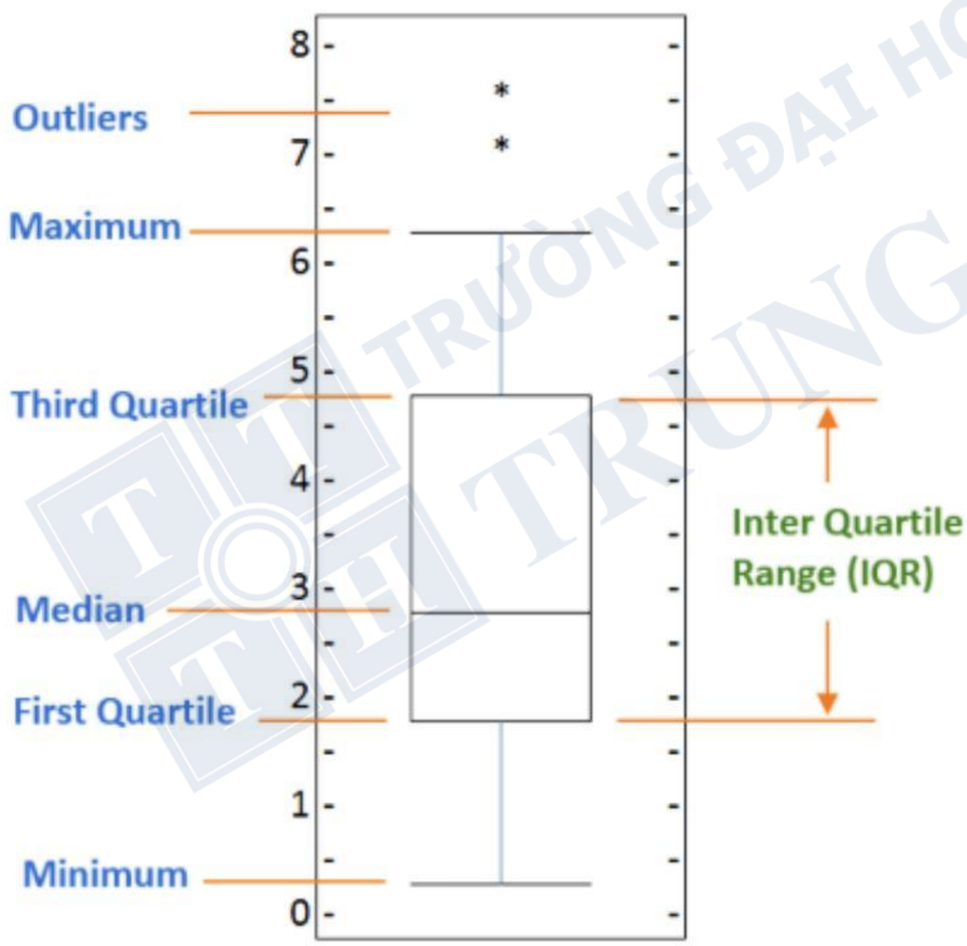
```
sns.displot(data=djia, x='Close', kind='kde')
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Box plot**
  - Là dạng biểu đồ trên đó mô tả phân bố giá trị theo các khoảng tứ phân vị.
  - Có thể dùng trong **phân tích đơn biến** (phân tích riêng biến liên tục) hoặc kết hợp phân tích biến phân loại – biến liên tục.



Lower Outlier < Q1 – (1.5 x IQR)

Higher Outlier > Q3 + (1.5 x IQR)

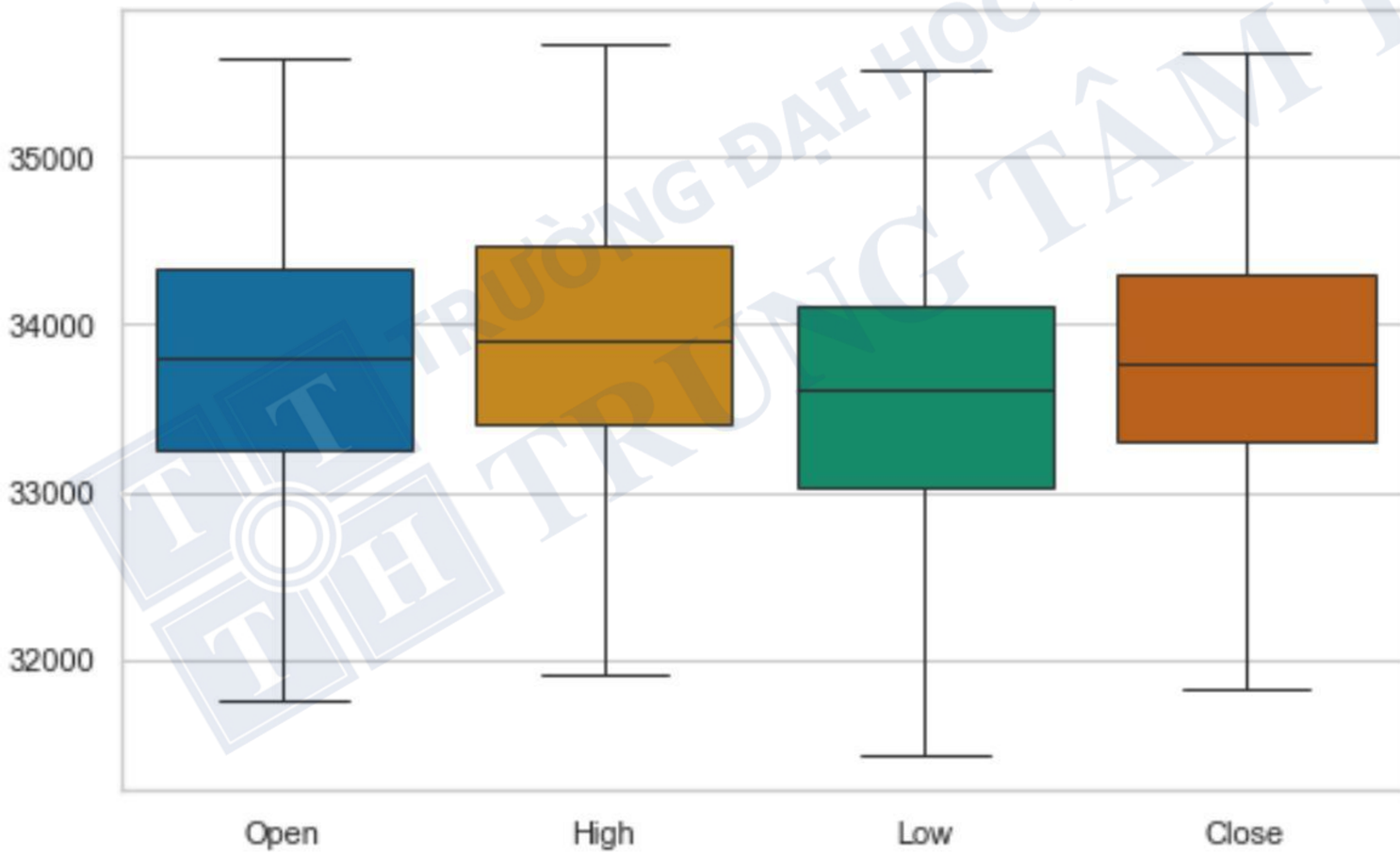
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Box plot**

```
sns.boxplot(<dataframe>,x=<column_name>,
            y=<column_name>,...)
```

```
sns.boxplot(data=djia.iloc[:,:-2])
```



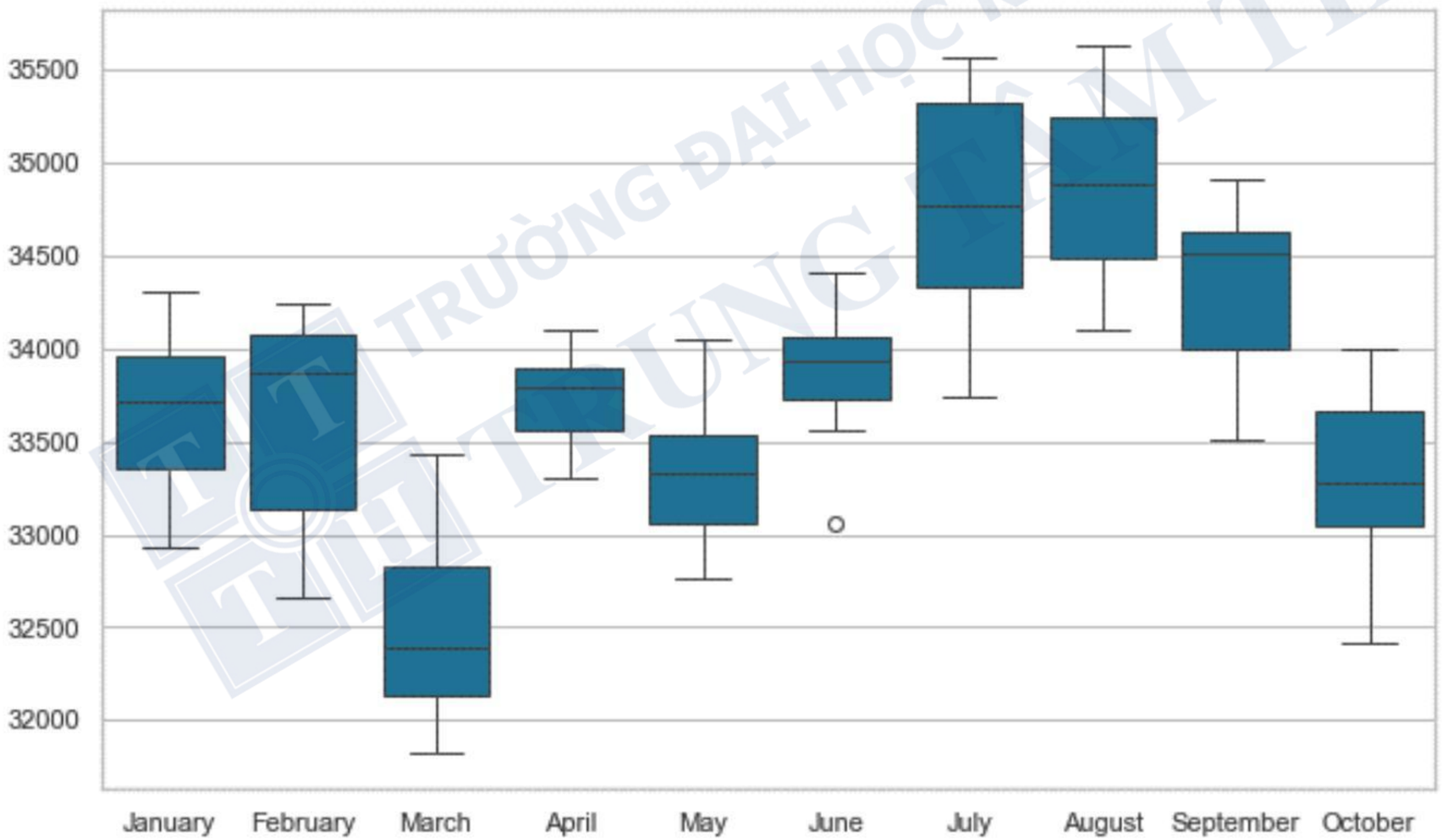
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Box plot**

```
sns.boxplot(<dataframe>,x=<column_name>,
            y=<column_name>,...)
```

```
sns.boxplot(data=djia,x='Month',y='Close')
```

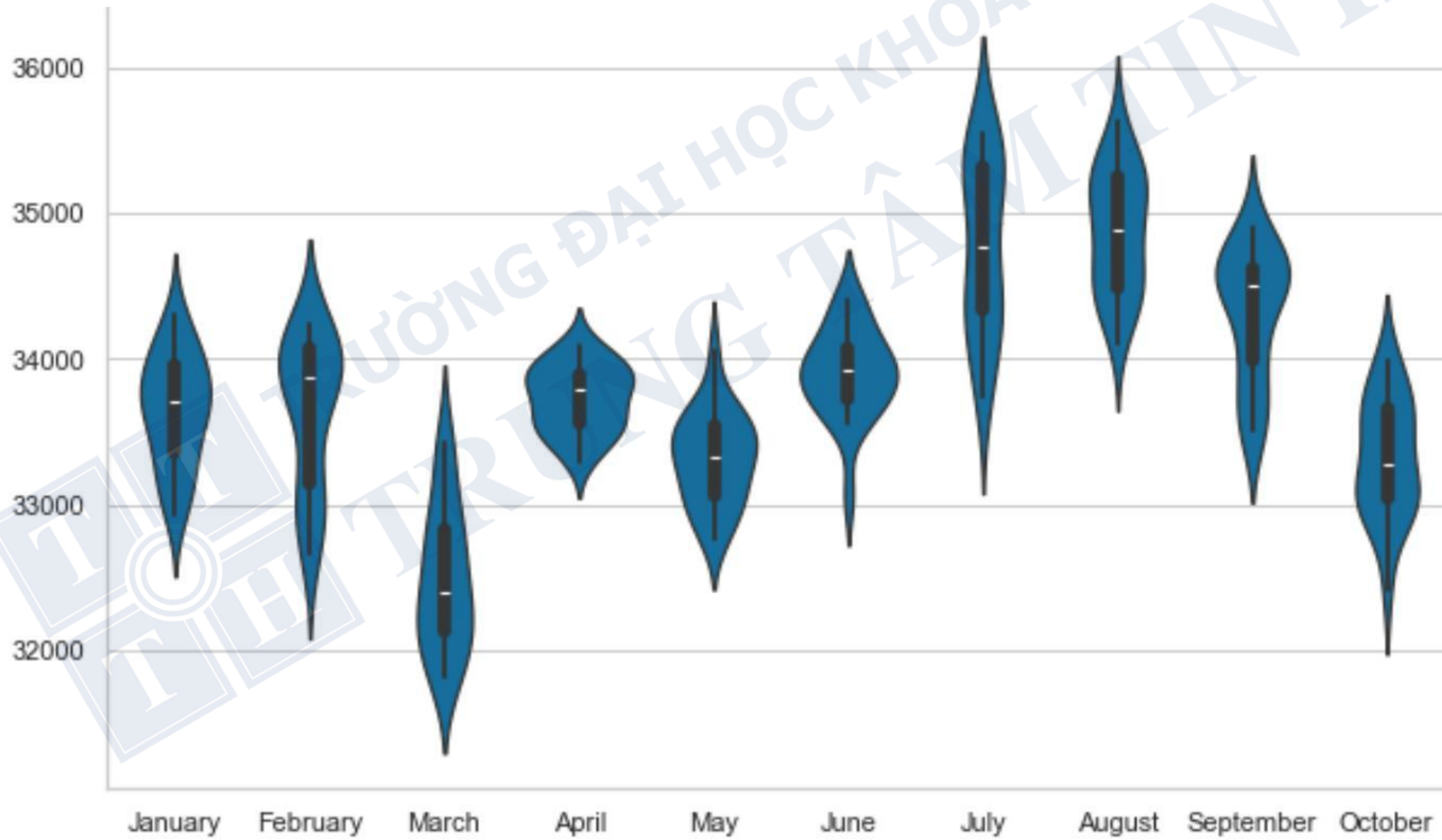


TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TT TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- Các hàm hiển thị phân phối dữ liệu tương tự Box plot: Violin plot, Swarm plot, Strip plot

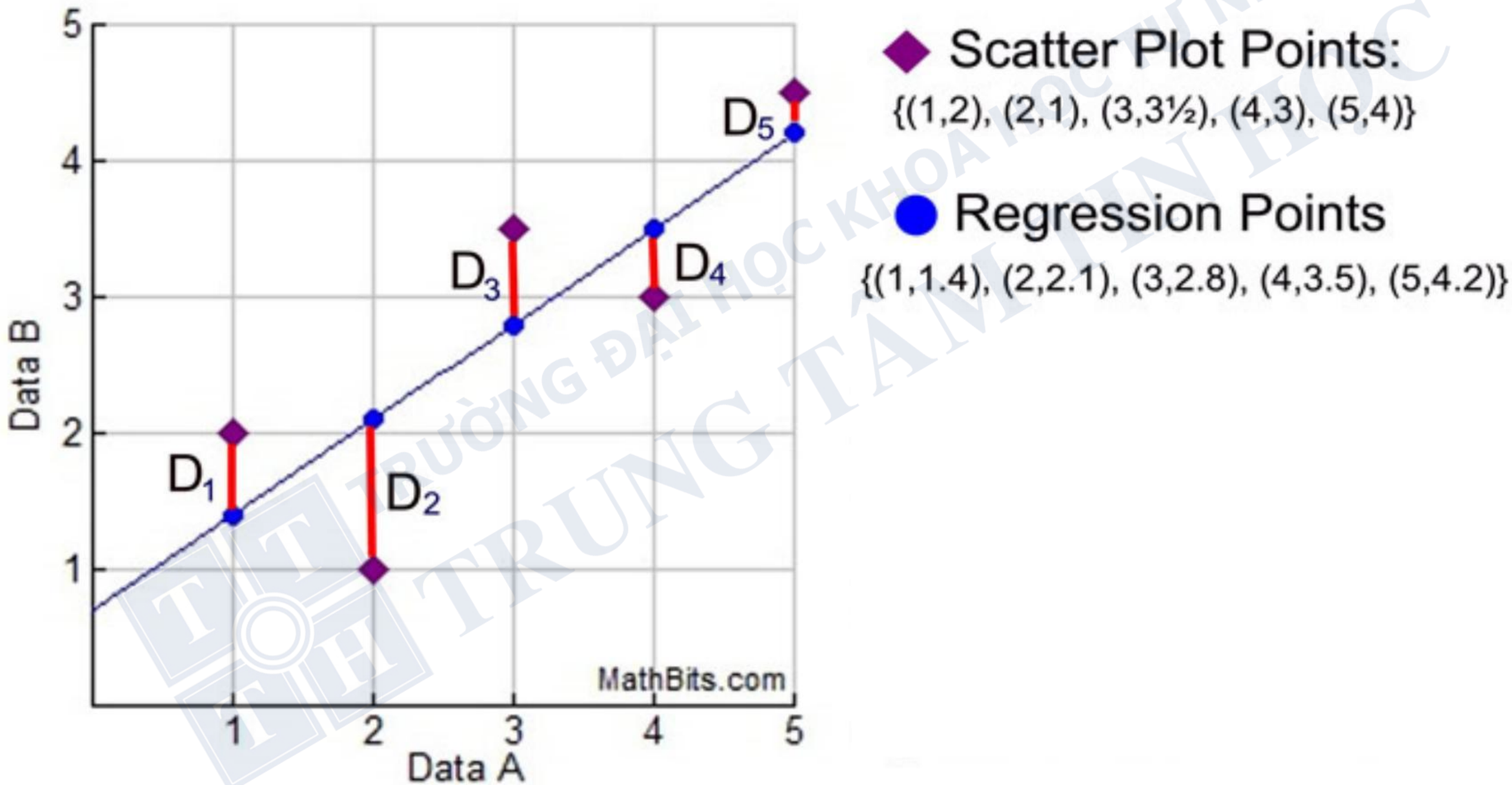
```
sns.violinplot(data=djia, x='Month', y='Close')
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Regression plot - Residual plot**



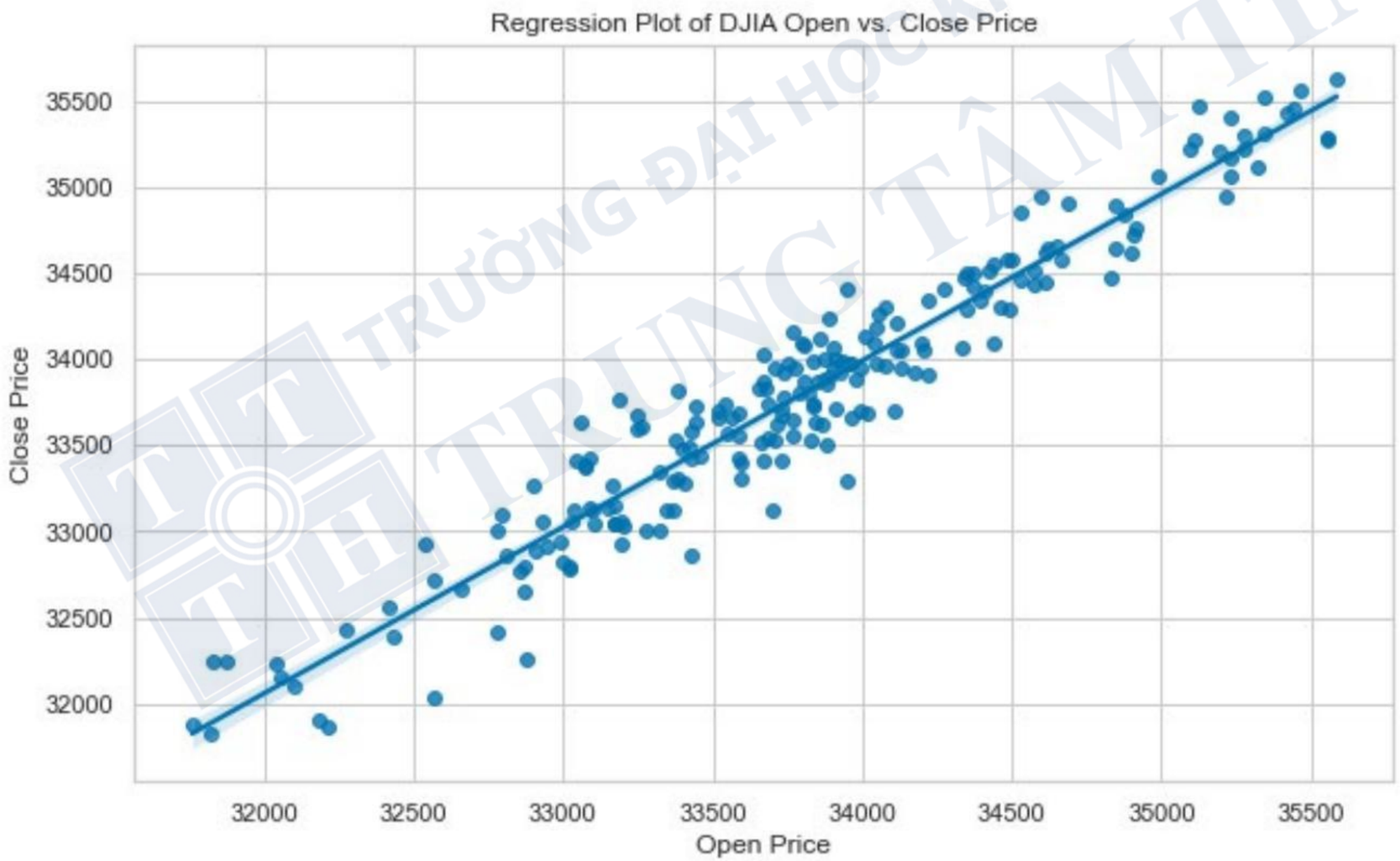
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Regression plot - Residual plot**

```
sns.regplot(<dataframe>,x=<column_name>,
            y=<column_name>,...)
```

```
sns.regplot(djia, x='Open', y='Close')
```



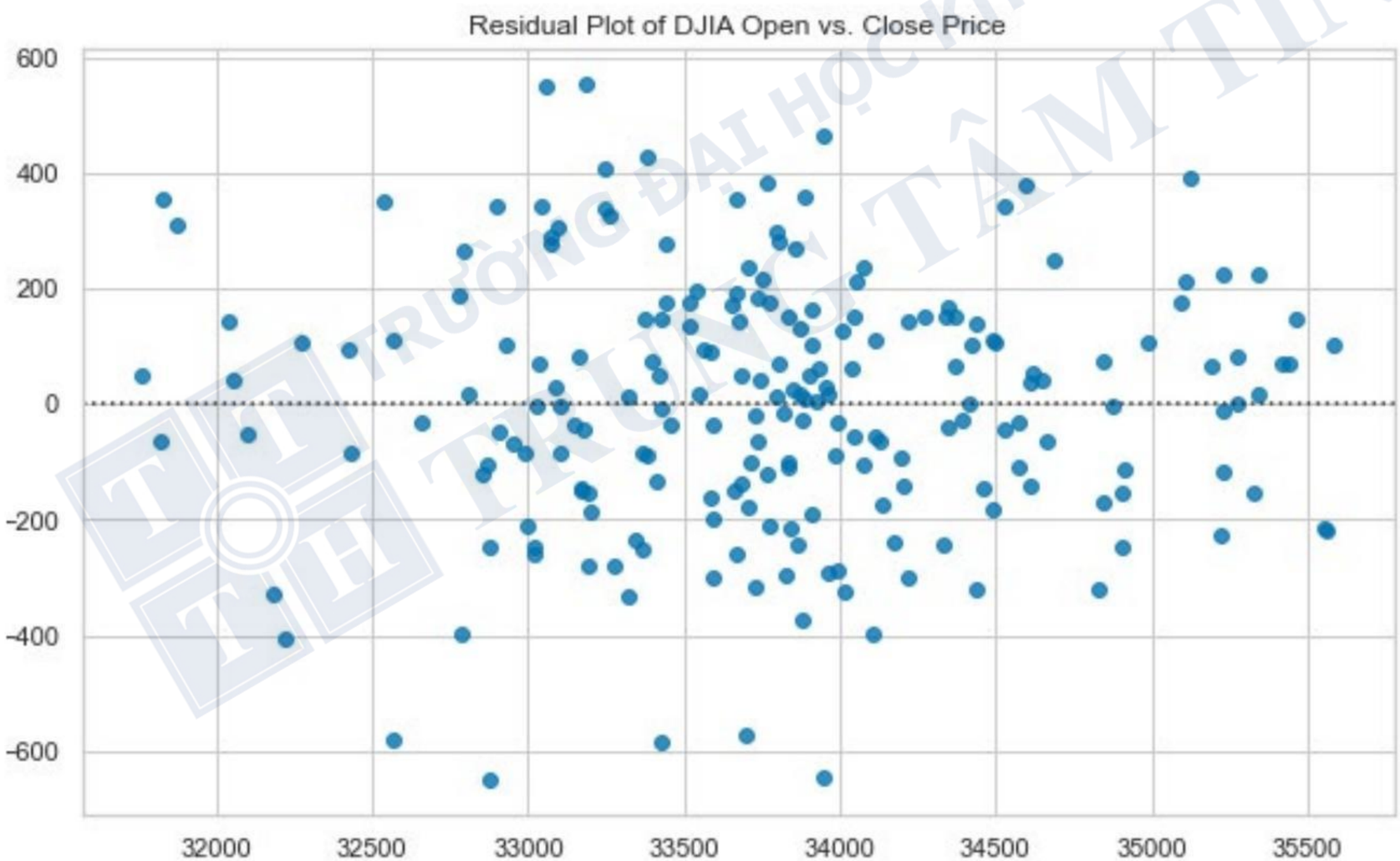
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Regression plot - Residual plot**

```
sns.residplot(<dataframe>,x=<column_name>,
              y=<column_name>,...)
```

```
sns.residplot(djia, x='Open', y='Close')
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Bar plot**

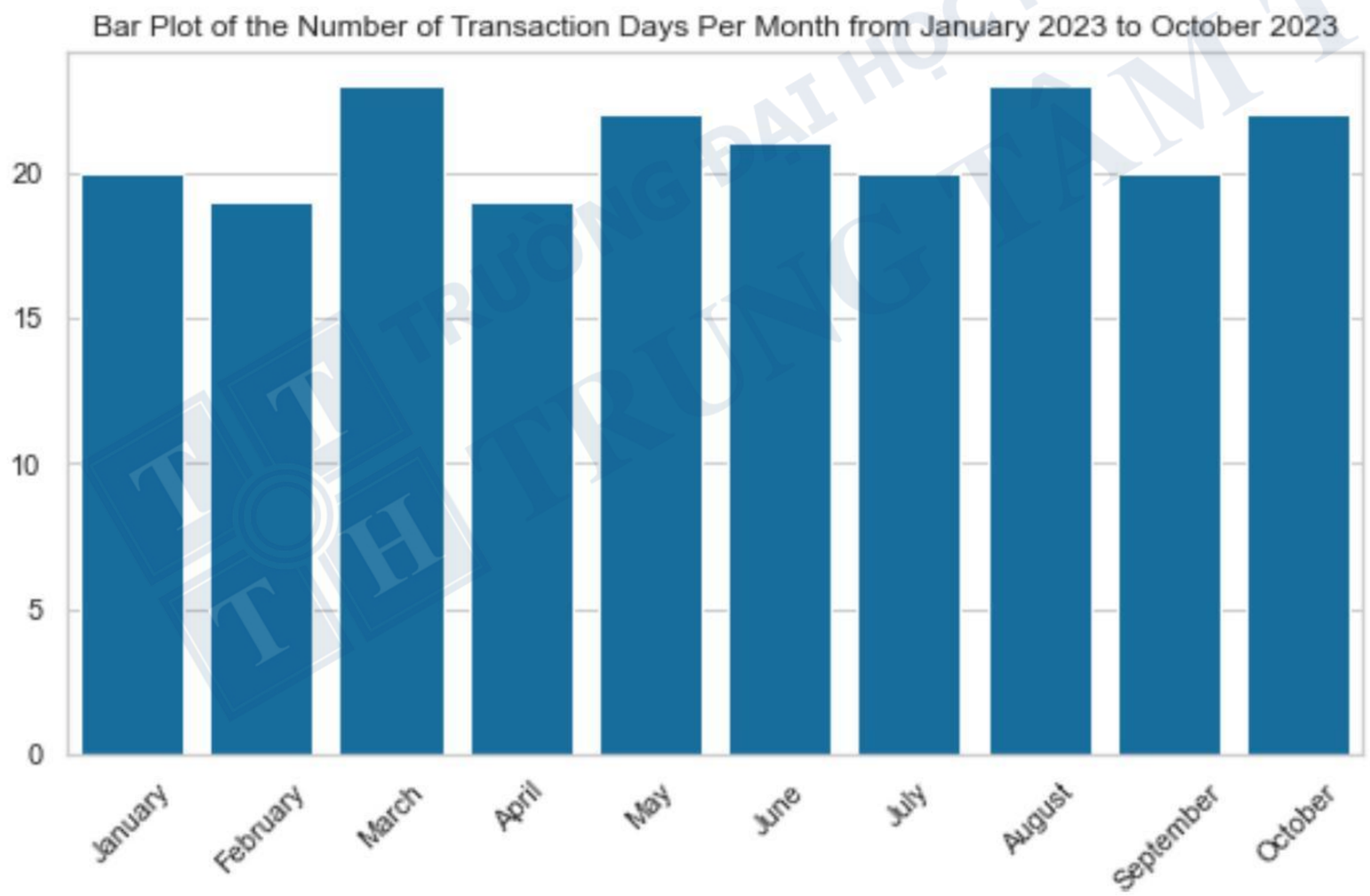
```
sns.barplot(<dataframe>,x=<column_name>,
            y=<column_name>,...)
```

```
sns.barplot(djia, y = 'Close', x='Month', errorbar=None)
```

Trực quan hóa dữ liệu với Seaborn

- **Count plot**
  - Dùng để hiển thị số lượng phần tử đếm được trong biến phân loại
  - Cú pháp: `sns.count plot(<dataframe>,x=<column_name>,...)`

```
sns.countplot(djia,  x='Month')
```

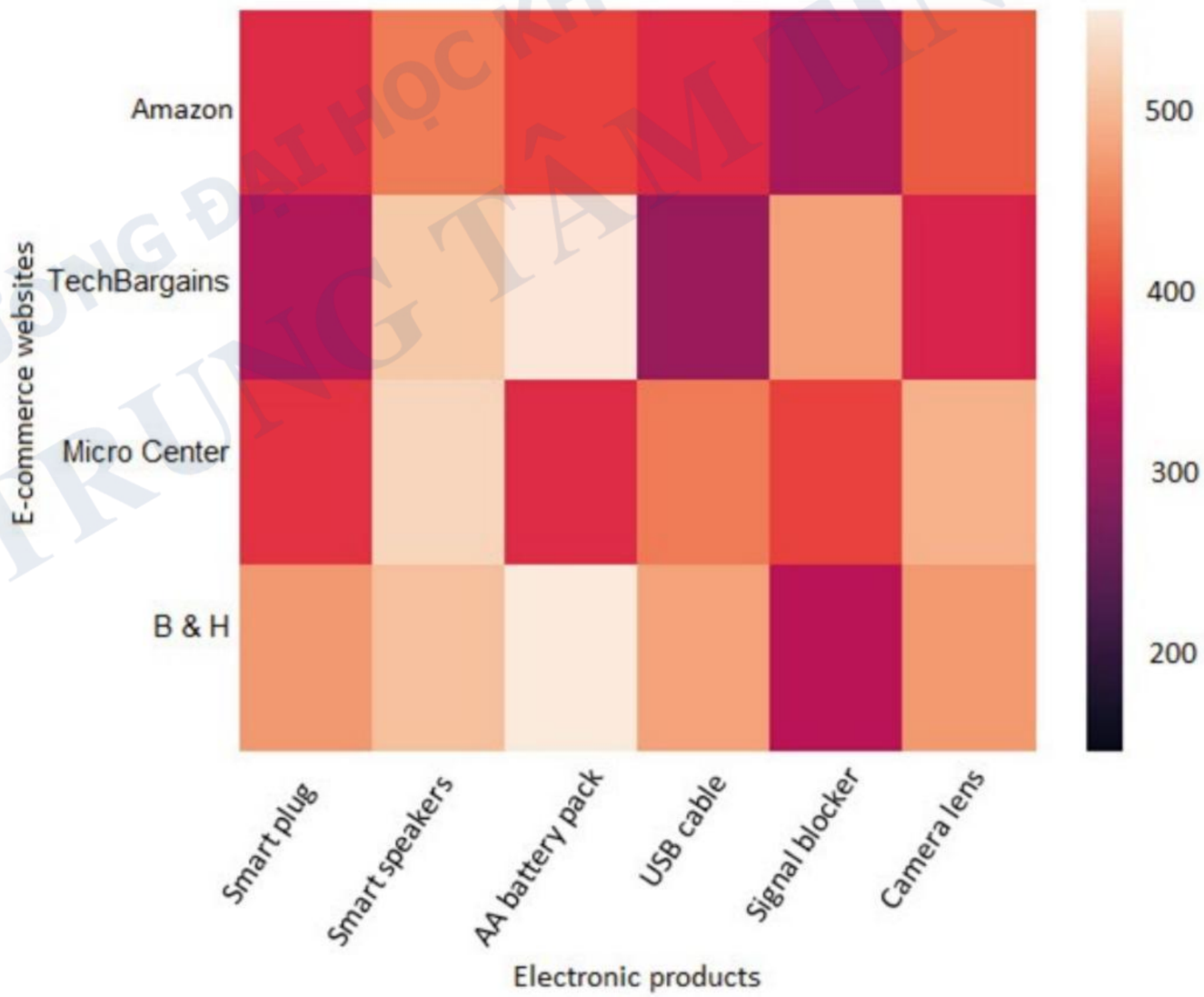


TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Heatmap**
  - Dùng để biểu diễn sự phân phối, mật độ, hoặc mức độ tần suất của các biến thông qua màu sắc trong một ma trận.
  - Có thể được dùng để tóm tắt giá trị tính toán giữa hai biến phân loại.

Hãy quan sát và nhận xét biểu đồ bên

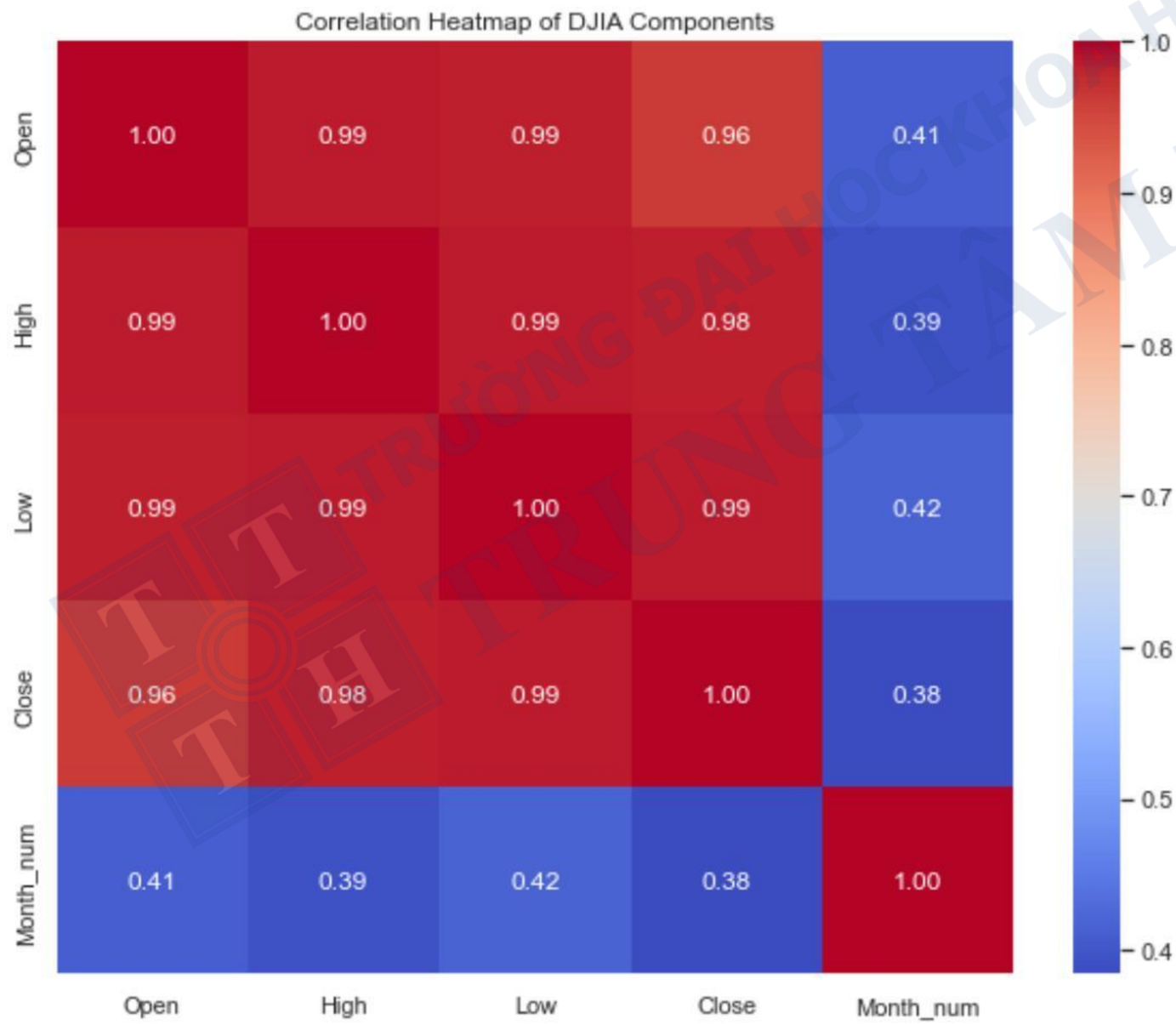


TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Heatmap** `sns.heatmap(<data>,...)`

```
heatmap = sns.heatmap(djia.corr(), annot=True,
                      fmt=".2f", cmap='coolwarm', cbar=True)
```



Trực quan hóa dữ liệu với Seaborn

- **Định dạng biểu đồ theo các mẫu định dạng sẵn (style)**

```
sns.set_style(<style name>)
```

- **darkgrid**: Nền tối với lưới, sử dụng khi cần so sánh dữ liệu và muốn có những đường lưới để dễ dàng tham chiếu giá trị.
- **whitegrid:** Tương tự như **darkgrid** nhưng với nền trắng.
- **dark:** Nền tối không có lưới, mang lại cho người xem cảm giác đơn giản, tập trung hoàn toàn vào dữ liệu mà không có lưới làm phân tâm.
- **white:** Tương tự như **dark** nhưng với nền trắng.
- **ticks:** Là dạng style **white** nhưng có thêm dấu ticks ở các trục, giúp việc đọc và hiểu giá trị trên trục dễ dàng hơn.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu với Seaborn

- **Định dạng màu sắc trong biểu đồ theo Palette**
  - Palette: bảng màu đã được sắp xếp các màu với nhau một cách hợp lý, có chủ đích.
  - Qualitative Colors (Màu định tính): dùng trong biểu đồ với dữ liệu là biến phân loại, với mỗi màu đại diện cho một giá trị trong biến phân loại.
  - Sequential Colors (Màu tuần tự): để biểu diễn dữ liệu có thứ tự với dải màu sắc chuyển từ đậm sang nhạt.
  - Diverging colors (Màu phân kỳ): được sử dụng khi muốn người xem chú ý đến giá trị cao nhất và thấp nhất của dữ liệu. Ví dụ: nhiệt độ, lợi nhuận, hệ số tương quan,...
  - Tùy chỉnh palette màu:

```
sns.set_palette(<palette name>)
```

Nội dung

1. Tổng quan
2. Trực quan hóa dữ liệu với Matplotlib
3. Trực quan hóa dữ liệu với Seaborn
4. **Phân tích trực quan đa biến với lưới**
5. Trực quan hóa dữ liệu văn bản

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

- **Ý nghĩa**
  - Cho phép người xem nhanh chóng có được nhiều góc nhìn khác nhau trên cùng tập dữ liệu.
  - Giúp phát hiện mô hình, xu hướng và ngoại lệ một cách dễ dàng.
  - Giúp phân tích, khám phá dữ liệu một cách toàn diện để xác định những biến quan trọng, cần phân tích thêm.
  - Sử dụng lưới biểu đồ giúp tận dụng tối đa không gian trực quan trên màn hình, cho phép người xem nắm bắt thông tin từ nhiều biểu đồ mà không cần phải di chuyển nhiều trên trang.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

- **Các hàm đã tích hợp sẵn tính năng cho phép phân tích đa biến với lưới:**
  - Là các hàm có tham số col, row và hue.
  - Không được sử dụng cho phương thức map của FacetGrid.
  - **relplot(), displot(), catplot()**, và **lmplot()** đều sử dụng Facetgrid bên trong khi vẽ lưới.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

- **FacetGrid**
  - Cho phép tạo lưới giúp dễ dàng phân tích các mối quan hệ giữa các biến.
  - Cho phép người dùng thể hiện nhiều loại biểu đồ trên lưới.
  - FacetGrid có thể được vẽ với tối đa 3 tham số: **row**, **col** và **hue**
  - Các bước thực hiện:
    - Tạo FacetGrid
    - Sử dụng hàm map() để vẽ biểu đồ tại các khu vực trong lưới
  - Các hàm có thể sử dụng trong hàm map(): sns.histplot, sns.kdeplot, sns.boxplot, sns.violinplot, sns.barplot, sns.countplot, sns.stripplot, sns.swarmplot, sns.scatterplot, plt.scatter, plt.hist, plt.plot...

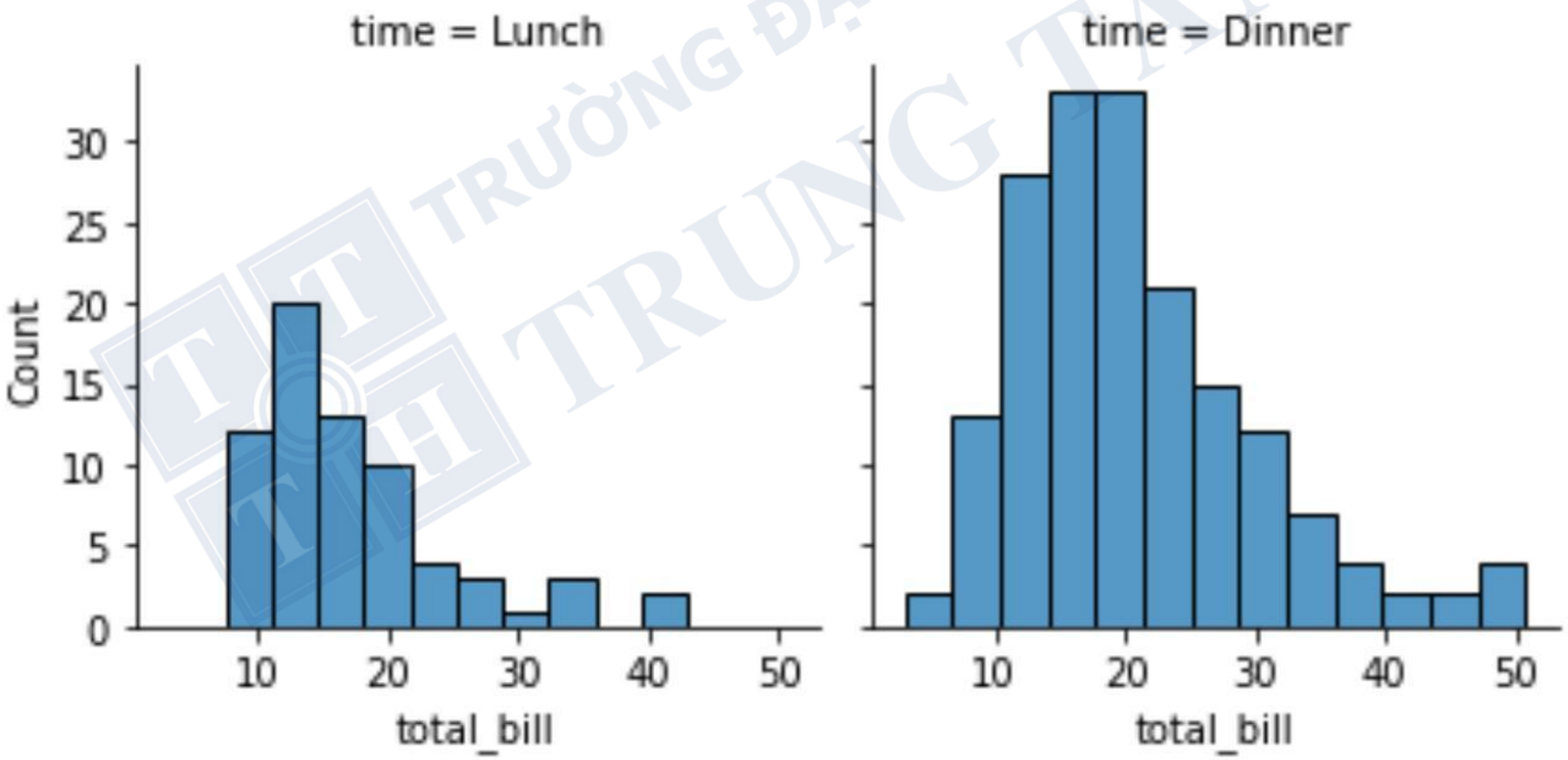
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

- **FacetGrid**

```
#Khởi tạo grid, tách tập dữ liệu theo "time"
g = sns.FacetGrid(tips, col="time")

#Sử dụng hàm map() để vẽ đồ thị trên các ô trong lưới
g.map(sns.histplot, "total_bill")
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

- **PairGrid**
  - Cung cấp nhanh chóng cái nhìn tổng quan về các mối quan hệ giữa các cặp biến số trong tập dữ liệu.
  - PairGrid cho phép thể hiện nhiều loại biểu đồ trên lưới (FacetGrid chỉ cho phép thể hiện cùng loại biểu đồ trên lưới)

```
#Định nghĩa lưới
g = sns.PairGrid(tips[['total_bill','tip','sex']],hue="sex")

#Vẽ biểu đồ ở đường chéo trên, đường chéo dưới và trên đường chéo
g.map_upper(sns.scatterplot)
g.map_lower(sns.kdeplot)
g.map_diag(sns.histplot)
g.add_legend()
```

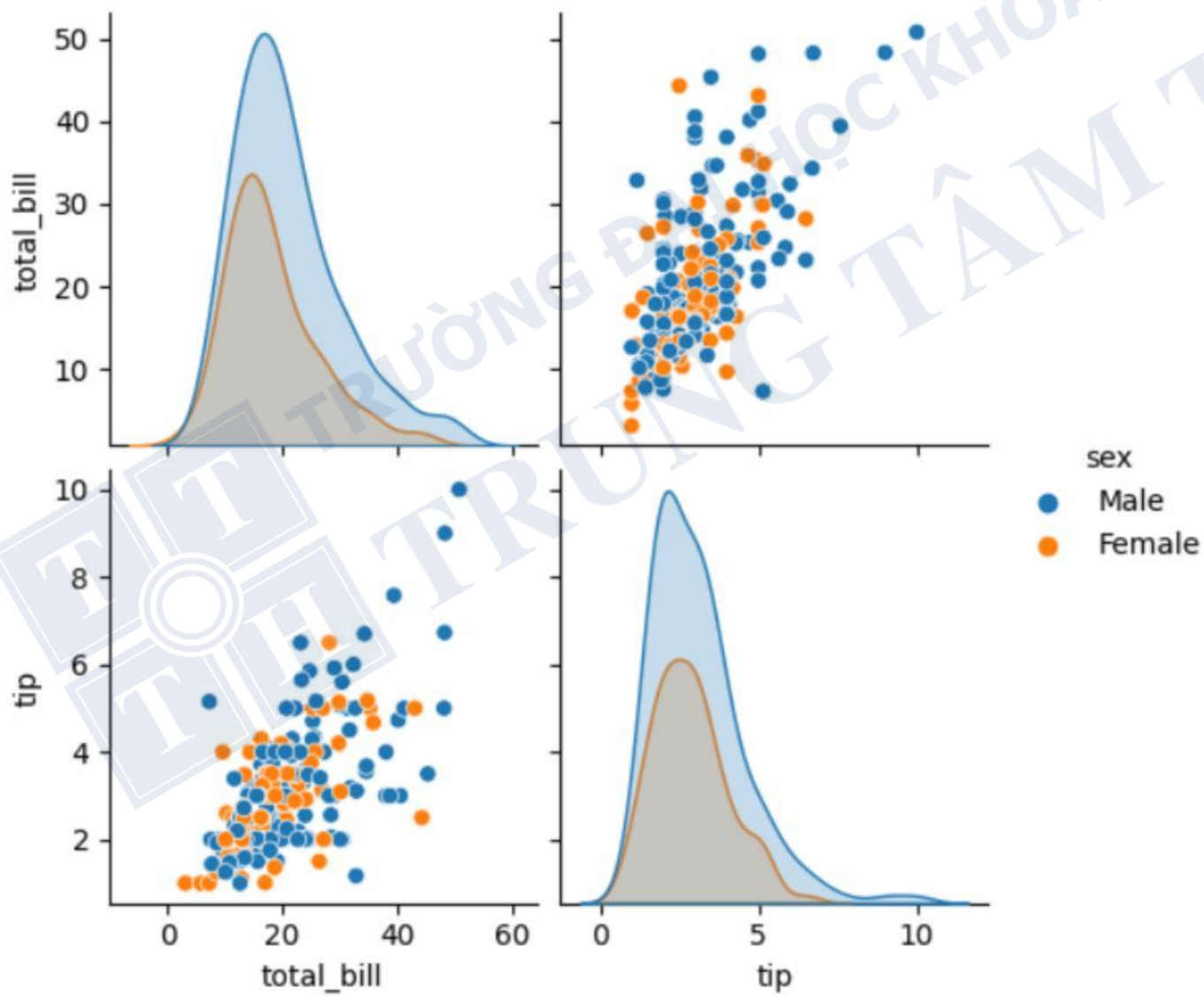
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

- **pairplot** https://seaborn.pydata.org/generated/seaborn.pairplot.html
  - pairplot() là một phiên bản thuận tiện, rút gọn của PairGrid

```
sns.pairplot(tips[['total_bill','tip','sex']], hue="sex")
```

```
<seaborn.axisgrid.PairGrid at 0x17f51bec970>
```



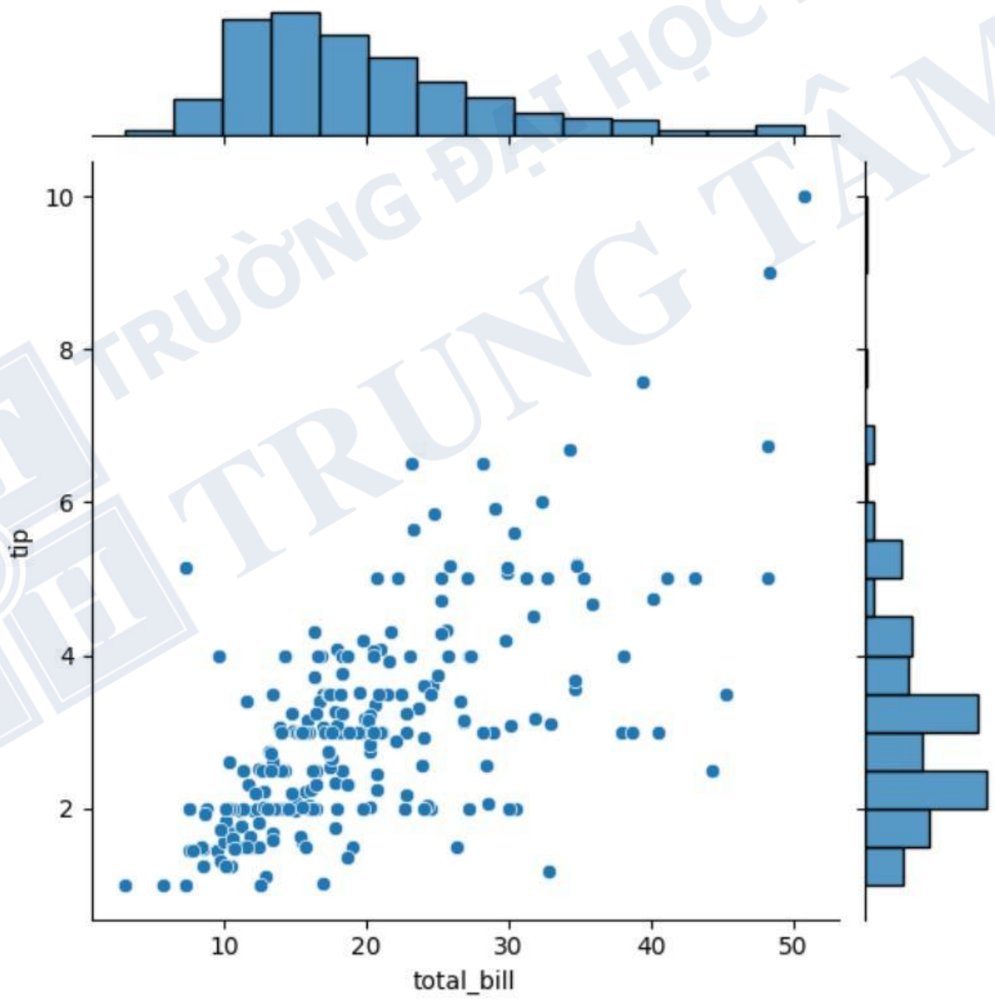
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

JointGrid

- JointGrid cho phép kết hợp các biểu đồ đơn biến (histogram, rug plot, kde) với các biểu đồ 2 biến (scatter, regression).

```
g=sns.JointGrid(data=tips, x='total_bill', y='tip')
g.plot(sns.scatterplot, sns.histplot)
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Phân tích trực quan đa biến với lưới

- Sử dụng subplots của Matplotlib để linh động tùy chỉnh biểu đồ trên lưới

```
# Khởi tạo subplots
fig, axes = plt.subplots(2, 2, figsize=(10, 8))

# Vẽ biểu đồ trên từng axes
sns.histplot(tips["total_bill"], ax=axes[0, 0])
sns.boxplot(x="day", y="total_bill", data=tips, ax=axes[0, 1])
sns.scatterplot(x="total_bill", y="tip", data=tips, ax=axes[1, 0])
sns.countplot(x="day", data=tips, ax=axes[1, 1])

# Tùy chỉnh layout và tiêu đề
plt.tight_layout()
plt.suptitle("Multiple Seaborn Plots", y = 1.05)

# Hiển thị figure
plt.show()
```

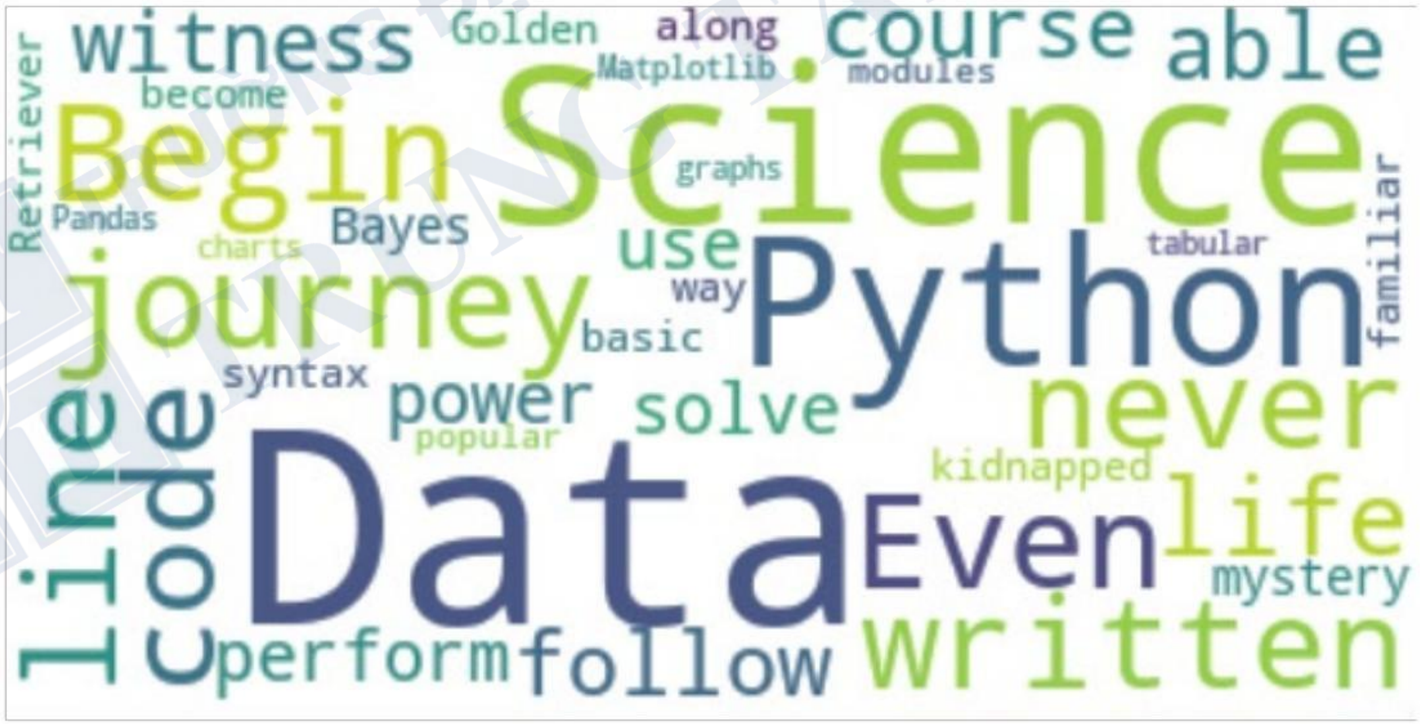
Nội dung

1. Tổng quan
2. Trực quan hóa dữ liệu với Matplotlib
3. Trực quan hóa dữ liệu với Seaborn
4. Phân tích trực quan đa biến với lưới
5. Trực quan hóa dữ liệu văn bản

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu văn bản

- **Word Cloud**
  - Hiển thị trực quan dữ liệu văn bản theo các từ khóa.
  - Giúp người xem nhanh chóng phát hiện các thuật ngữ nổi bật nhất.
  - Tần suất xuất hiện của từ khóa trong dữ liệu văn bản càng nhiều, thì kích thước từ khóa càng lớn và ở vị trí trung tâm hơn khi thể hiện trong word clouds.



Trực quan hóa dữ liệu văn bản

- **Sử dụng thư viện wordcloud để vẽ Word Cloud**
  - Cài đặt

```
pip install wordcloud
```

  - Các bước thực hiện:
    - Thu thập dữ liệu văn bản (text)
    - Tạo wordcloud image : `WordCloud().generate(text)`
    - Hiển thị: `plt.imshow()`

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu văn bản

- Hiển thị WordCloud với định dạng mặc định

```
from wordcloud import WordCloud, STOPWORDS
```

```
wordcloud = WordCloud().generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu văn bản

- Loại bỏ stop word, các từ không quan trọng và tùy chỉnh hiển thị

```
stopwords = set(STOPWORDS)
#Tùy chỉnh word cloud
wc = WordCloud(
    background_color='white',
    max_words=30,
    stopwords=stopwords
)

#Cập nhật, bổ sung stop word
list_of_words = ['time','well','didn','today','anyone','day']
stopwords.update(list_of_words)

#Tạo word cloud
wc.generate(text)

# Hiển thị wordcloud
plt.figure(figsize=(10, 12))
plt.imshow(wc, interpolation='bilinear')
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Trực quan hóa dữ liệu văn bản

- Tạo mask cho word cloud

```
import numpy as np
from PIL import Image
```

```
wc_mask = np.array(Image.open(r'data\heart.jpg'))
```

```
plt.imshow(wc_mask, interpolation='bilinear')
```

```
#Tùy chỉnh word cloud
wc = WordCloud(background_color='white',
               max_words=50, mask=wc_mask,
               stopwords=stopwords)
#Tạo word cloud
wc.generate(text)
# Hiển thị word cloud
plt.figure(figsize=(10, 12))
plt.imshow(wc1, interpolation='bilinear')
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC