**🎨 HƯỚNG DẪN VẼ BIỂU ĐỒ BẰNG MATPLOTLIB & SEABORN**

**🧱 I. Cấu trúc cơ bản của biểu đồ Matplotlib**

**1️⃣ Khởi tạo môi trường vẽ**

import matplotlib.pyplot as plt

import numpy as np

* matplotlib.pyplot là module chính để vẽ đồ thị 2D.
* Mọi hình vẽ trong matplotlib đều được hiển thị trong **“Figure”** (hình tổng) chứa **“Axes”** (vùng biểu đồ).

**2️⃣ Tạo dữ liệu mẫu**

x = np.linspace(0, 10, 100)

y = np.sin(x)

* np.linspace(0,10,100) → sinh 100 điểm đều nhau từ 0 đến 10.
* np.sin(x) → tính sin(x) tại các điểm đó.

**3️⃣ Vẽ biểu đồ đầu tiên (Line chart)**

plt.plot(x, y)

plt.show()

**Giải thích:**

* plt.plot(x,y) → vẽ đồ thị đường (line).
* plt.show() → hiển thị toàn bộ biểu đồ, bắt buộc phải có trong script.

**Kết quả:**  
Biểu đồ sóng sin đơn giản, nhưng mặc định (màu xanh, không tiêu đề, không nhãn trục).

**🖌 II. Làm biểu đồ đẹp và có ý nghĩa**

**1️⃣ Thêm tiêu đề và nhãn trục**

plt.plot(x, y)

plt.title("Đồ thị hàm sin / Sine function")

plt.xlabel("Giá trị x / X values")

plt.ylabel("Giá trị sin(x) / Y = sin(x)")

plt.show()

**Giải thích:**

* plt.title() → tiêu đề trên cùng.
* plt.xlabel() và plt.ylabel() → tên trục giúp người xem hiểu nội dung.

**2️⃣ Thay đổi kiểu đường và màu**

plt.plot(x, y, color='red', linestyle='--', linewidth=2, marker='o', markersize=4)

**Giải thích tham số:**

| **Tham số** | **Ý nghĩa** | **Gợi ý** |
| --- | --- | --- |
| color | Màu đường ('red', 'blue', '#FF8800') | Dùng mã HEX để tạo màu chuẩn |
| linestyle | Kiểu đường: '-', '--', ':', '-. ' | Dùng '--' cho đường phụ |
| linewidth | Độ dày đường | 1–3 là vừa |
| marker | Ký hiệu tại điểm dữ liệu | 'o', 's', '^', '\*' |
| markersize | Kích thước marker | 3–8 tùy độ dày biểu đồ |

**3️⃣ Bật lưới (grid)**

plt.grid(True, linestyle=':', alpha=0.6)

* True → bật lưới.
* linestyle=':' → chấm mảnh.
* alpha=0.6 → làm mờ (0–1), giúp nhìn mềm mại.

**4️⃣ Hiển thị nhiều đường trên cùng đồ thị**

y2 = np.cos(x)

plt.plot(x, y, label='sin(x)', color='blue')

plt.plot(x, y2, label='cos(x)', color='orange')

plt.legend()

plt.show()

**Giải thích:**

* label='...' → tên hiển thị trong chú thích.
* plt.legend() → bật chú thích (legend box).

**5️⃣ Cố định kích thước hình**

plt.figure(figsize=(8,5))

plt.plot(x, y)

plt.show()

* figsize=(8,5) → chiều rộng 8 inch, cao 5 inch.  
  => Giúp biểu đồ cân đối hơn trong báo cáo.

**6️⃣ Lưu ảnh chất lượng cao**

plt.savefig("sin\_chart.png", dpi=300, bbox\_inches='tight')

* dpi=300 → độ phân giải in ấn.
* bbox\_inches='tight' → cắt viền trắng thừa.  
  → Rất quan trọng khi làm **báo cáo học thuật**.

**📊 III. Các loại biểu đồ cơ bản (Matplotlib)**

**1️⃣ Histogram — Biểu đồ tần suất**

data = np.random.normal(170, 10, 200)

plt.hist(data, bins=15, color='skyblue', edgecolor='black', alpha=0.8)

plt.title("Phân phối chiều cao / Height distribution")

plt.xlabel("Chiều cao (cm)")

plt.ylabel("Tần suất xuất hiện")

plt.grid(alpha=0.3)

plt.show()

**Giải thích:**

* bins=15 → chia dữ liệu thành 15 khoảng.
* alpha=0.8 → độ trong suốt giúp nhẹ mắt.
* edgecolor='black' → thêm viền cột để rõ ràng.

👉 **Dùng histogram** để xem phân bố dữ liệu (đều, lệch trái/phải, chuẩn,...)

**2️⃣ Bar chart — Biểu đồ cột**

categories = ['A', 'B', 'C', 'D']

values = [5, 7, 3, 4]

plt.bar(categories, values, color='tomato', alpha=0.7)

plt.title("Biểu đồ cột / Bar chart")

plt.xlabel("Nhóm")

plt.ylabel("Giá trị")

plt.show()

* plt.bar() → cột đứng.
* plt.barh() → cột nằm ngang.

**3️⃣ Scatter plot — Biểu đồ phân tán**

x = np.random.rand(50)

y = 2\*x + np.random.randn(50)\*0.2

plt.scatter(x, y, color='green', s=60, alpha=0.6, edgecolors='black')

plt.title("Mối quan hệ giữa X và Y / Relationship between X and Y")

plt.xlabel("X")

plt.ylabel("Y")

plt.grid(True, linestyle='--', alpha=0.5)

plt.show()

* s=60 → kích thước điểm.
* edgecolors → viền ngoài điểm.
* Dùng để quan sát mối tương quan giữa hai biến.

**4️⃣ Boxplot — Biểu đồ hộp**

data = [np.random.normal(170, 10, 100), np.random.normal(160, 8, 100)]

plt.boxplot(data, patch\_artist=True, labels=['Nam', 'Nữ'], boxprops=dict(facecolor='lightblue'))

plt.title("So sánh phân phối / Boxplot comparison")

plt.ylabel("Chiều cao (cm)")

plt.show()

**Giải thích:**  
Boxplot hiển thị trung vị, tứ phân vị, và ngoại lệ.

* patch\_artist=True → tô màu hộp.
* boxprops=dict(facecolor='lightblue') → đặt màu cho hộp.

**5️⃣ Pie chart — Biểu đồ tròn**

sizes = [40, 35, 25]

labels = ['A', 'B', 'C']

plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90, colors=['#66b3ff','#99ff99','#ffcc99'])

plt.title("Tỷ lệ nhóm / Pie chart")

plt.show()

**Giải thích:**

* autopct='%1.1f%%' → hiển thị phần trăm.
* startangle=90 → bắt đầu từ góc 90° để cân đối.
* colors → danh sách màu thủ công.

**🌈 IV. Làm biểu đồ chuyên nghiệp hơn với Seaborn**

import seaborn as sns

Seaborn được xây trên Matplotlib, có giao diện thân thiện và màu mặc định đẹp hơn.

**1️⃣ Histogram với Seaborn**

sns.histplot(data=data, bins=15, kde=True, color='skyblue')

plt.title("Phân phối chiều cao (Seaborn)")

plt.show()

* kde=True → thêm đường mật độ chuẩn (đường cong mượt).
* color → màu histogram.  
  👉 Thích hợp để kiểm tra dữ liệu có chuẩn hay không.

**2️⃣ Scatter plot với Seaborn**

tips = sns.load\_dataset("tips")

sns.scatterplot(x='total\_bill', y='tip', hue='sex', style='smoker', data=tips)

plt.title("Mối quan hệ giữa tổng bill và tip")

plt.show()

**Giải thích:**

| **Tham số** | **Ý nghĩa** | **Ví dụ** |
| --- | --- | --- |
| x, y | Cột trục X và Y | 'total\_bill', 'tip' |
| hue | Phân biệt màu theo nhóm | 'sex' |
| style | Kiểu marker khác nhau | 'smoker' |
| data | Tên DataFrame chứa dữ liệu | tips |

**3️⃣ Boxplot và Violin plot**

sns.boxplot(x='day', y='tip', hue='sex', data=tips, palette='Set2')

sns.violinplot(x='day', y='tip', hue='sex', data=tips, split=True, inner='quartile')

plt.title("Phân phối tip theo ngày & giới tính")

plt.show()

* palette='Set2' → bộ màu có sẵn.
* split=True → chia đôi violin theo giới tính.
* inner='quartile' → hiển thị tứ phân vị.

**4️⃣ Pairplot — Vẽ nhiều biểu đồ tự động**

sns.pairplot(tips, hue='sex')

Tự động vẽ scatter + histogram giữa **mọi cặp biến số** → rất hữu ích để **khảo sát dữ liệu thô**.

**5️⃣ Heatmap — Bản đồ nhiệt (dễ ăn điểm đồ án)**

corr = tips.corr()

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

plt.title("Tương quan giữa các biến / Correlation Heatmap")

plt.show()

**Giải thích:**

* annot=True → hiển thị giá trị số trong từng ô.
* cmap → bảng màu ('coolwarm', 'viridis', 'YlGnBu').
* fmt='.2f' → hiển thị số với 2 chữ số sau dấu phẩy.

**⚙️ V. Mẹo làm biểu đồ đẹp hơn**

| **Hành động** | **Lệnh / Thông số** | **Giải thích** |
| --- | --- | --- |
| Giảm rối | alpha=0.7 | Làm màu mờ hơn để dễ nhìn |
| Dùng bảng màu chuẩn | sns.set\_palette('Set2') | Màu hài hòa, cân đối |
| Dùng plt.style.use('seaborn-v0\_8') | Áp dụng theme hiện đại | Dễ nhìn, cân đối |
| Dùng plt.tight\_layout() | Căn chỉnh chữ, tránh đè | Cần khi nhiều subplot |
| Đặt kích thước hợp lý | plt.figure(figsize=(8,5)) | Cân đối ngang/dọc |
| Xuất hình đẹp | plt.savefig('chart.png', dpi=300, bbox\_inches='tight') | In/đồ án chuyên nghiệp |

**🧩 VI. Tổng kết – Quy trình vẽ biểu đồ hoàn chỉnh**

| **Bước** | **Lệnh** | **Mục đích** |
| --- | --- | --- |
| 1️⃣ | plt.figure(figsize=(8,5)) | Tạo vùng vẽ |
| 2️⃣ | plt.plot() / sns.\*plot() | Vẽ dữ liệu |
| 3️⃣ | plt.title(), xlabel(), ylabel() | Thêm mô tả |
| 4️⃣ | plt.legend() | Hiển thị chú thích |
| 5️⃣ | plt.grid() | Thêm lưới |
| 6️⃣ | plt.tight\_layout() | Căn chỉnh layout |
| 7️⃣ | plt.show() hoặc plt.savefig() | Hiển thị / lưu ảnh |

**🎨 PHẦN 1 – BIỂU ĐỒ CƠ BẢN / BASIC PLOTS WITH MATPLOTLIB**

**🧩 1️⃣ Biểu đồ đường (Line Plot)**

import matplotlib.pyplot as plt

import numpy as np

x = np.linspace(0, 10, 100)

y = np.sin(x)

plt.plot(x, y, color='blue', linewidth=2, linestyle='-')

plt.title("Hàm sin / Sine function")

plt.xlabel("Giá trị x / X values")

plt.ylabel("Giá trị sin(x)")

plt.grid(True, linestyle=':', alpha=0.6)

plt.show()

**Giải thích:**

* np.linspace() tạo dữ liệu.
* plt.plot() vẽ đường liên tục.
* linewidth, linestyle, color làm đường rõ nét.
* grid() thêm lưới mảnh để đọc dễ hơn.

**Hình minh họa:** đường cong màu xanh nhạt, nền trắng, có lưới chấm nhẹ.

**🧩 2️⃣ Biểu đồ cột (Bar Chart)**

categories = ['A', 'B', 'C', 'D']

values = [5, 7, 3, 4]

plt.bar(categories, values, color='orange', edgecolor='black', alpha=0.8)

plt.title("Biểu đồ cột / Bar Chart")

plt.xlabel("Nhóm")

plt.ylabel("Giá trị")

plt.show()

**Giải thích:**

* plt.bar() tạo cột đứng.
* edgecolor thêm viền đen cho rõ ràng.
* alpha làm màu mờ nhẹ để không gắt.

**Hình minh họa:** 4 cột cam, viền đen, nền trắng, trục có nhãn “A–D”.

**🧩 3️⃣ Biểu đồ phân tán (Scatter Plot)**

x = np.random.rand(50)

y = 2\*x + np.random.randn(50)\*0.2

plt.scatter(x, y, color='green', s=70, alpha=0.6, edgecolors='black')

plt.title("Phân tán X–Y / Scatter plot")

plt.xlabel("X")

plt.ylabel("Y")

plt.grid(True, linestyle='--', alpha=0.5)

plt.show()

**Giải thích:**

* np.random.rand() sinh dữ liệu ngẫu nhiên.
* plt.scatter() vẽ điểm rời rạc.
* s=70 chỉnh kích thước điểm.
* edgecolors='black' thêm viền cho điểm.

**Hình minh họa:** các chấm xanh lá rải quanh đường chéo, có lưới mảnh.

**🧩 4️⃣ Biểu đồ tần suất (Histogram)**

data = np.random.normal(170, 10, 200)

plt.hist(data, bins=15, color='skyblue', edgecolor='black', alpha=0.8)

plt.title("Phân phối chiều cao / Height distribution")

plt.xlabel("Chiều cao (cm)")

plt.ylabel("Tần suất")

plt.grid(alpha=0.3)

plt.show()

**Giải thích:**

* np.random.normal() sinh 200 chiều cao chuẩn (μ = 170, σ = 10).
* plt.hist() chia 15 khoảng (bins).
* edgecolor + alpha giúp cột mềm và rõ.

**Hình minh họa:** cột xanh nhạt có dạng chuông, hơi cao ở giữa.

**🧩 5️⃣ Biểu đồ tròn (Pie Chart)**

sizes = [40, 35, 25]

labels = ['A', 'B', 'C']

plt.pie(sizes, labels=labels, autopct='%1.1f%%',

startangle=90, colors=['#66b3ff','#99ff99','#ffcc99'])

plt.title("Tỷ lệ thành phần / Pie chart")

plt.show()

**Giải thích:**

* sizes → phần trăm thực.
* labels → tên mỗi miếng.
* autopct → hiển thị phần trăm.
* startangle=90 → quay bánh cho cân đối.
* colors → bảng màu tự chọn.

**Hình minh họa:** bánh tròn với 3 miếng xanh – xanh lá – vàng, có tỷ lệ 40%, 35%, 25%.

# 🎨 **HƯỚNG DẪN CHUYÊN SÂU BIỂU ĐỒ THỐNG KÊ (Histogram / Boxplot / Bar chart)**

## 1️⃣ **Histogram – Biểu đồ tần suất**

### 🎯 Mục đích

Dùng để quan sát **phân bố dữ liệu** (chuẩn, lệch, đều,...).  
Thường áp dụng cho dữ liệu liên tục như chiều cao, điểm thi, doanh thu,...

### 📘 **Matplotlib**

import matplotlib.pyplot as plt

import numpy as np

data = np.random.normal(170, 10, 200) # sinh 200 giá trị ~ N(170, 10²)

plt.figure(figsize=(8,5))

plt.hist(data, bins=15, color='skyblue', edgecolor='black', alpha=0.8, density=True)

plt.title("Phân phối chiều cao / Height distribution")

plt.xlabel("Chiều cao (cm)")

plt.ylabel("Mật độ xác suất")

plt.grid(alpha=0.3)

plt.show()

**🔍 Giải thích từng dòng**

| **Lệnh** | **Giải thích** |
| --- | --- |
| np.random.normal(170, 10, 200) | Sinh dữ liệu có trung bình 170, σ = 10 |
| plt.hist(..., bins=15) | Chia dữ liệu thành 15 khoảng |
| color='skyblue' | Màu cột xanh nhạt dễ nhìn |
| edgecolor='black' | Thêm viền đen để phân biệt cột |
| alpha=0.8 | Làm cột trong nhẹ để dịu mắt |
| density=True | Biểu đồ chuẩn hóa thành mật độ (PDF), không phải tần suất thô |
| grid(alpha=0.3) | Lưới mờ giúp đọc trục dễ hơn |

**🖼 Hình minh họa:**  
Cột xanh nhạt dạng “chuông”, có lưới nhẹ → trực quan rõ ràng.

### 🌈 **Làm đẹp nâng cao**

Thêm đường cong chuẩn vào Histogram:

from scipy.stats import norm

x = np.linspace(130, 210, 100)

pdf = norm.pdf(x, loc=data.mean(), scale=data.std())

plt.hist(data, bins=15, color='skyblue', edgecolor='black', density=True, alpha=0.7)

plt.plot(x, pdf, color='red', linewidth=2, label='Normal PDF')

plt.legend()

plt.title("Histogram với đường chuẩn")

plt.show()

**👉 Giải thích:**

* norm.pdf() tính giá trị lý thuyết của hàm mật độ chuẩn.
* plt.plot(x, pdf) vẽ đường cong đỏ mượt lên trên Histogram → dễ so sánh thực tế với lý thuyết.

### 📘 **Seaborn**

import seaborn as sns

sns.histplot(data, bins=15, kde=True, color='skyblue')

plt.title("Histogram với KDE (Seaborn)")

plt.xlabel("Chiều cao (cm)")

plt.ylabel("Mật độ xác suất")

plt.show()

**Giải thích:**

* kde=True → thêm đường cong mượt (kernel density estimation).
* color='skyblue' → dùng tone dịu để biểu đồ chuyên nghiệp.

## 2️⃣ **Boxplot – Biểu đồ hộp**

### 🎯 Mục đích

Hiển thị **trung vị, tứ phân vị, ngoại lệ (outliers)**.  
Giúp so sánh phân phối giữa các nhóm (ví dụ: nam – nữ, ngày – đêm,...).

### 📘 **Matplotlib**

np.random.seed(0)

data1 = np.random.normal(170, 10, 100)

data2 = np.random.normal(160, 8, 100)

plt.figure(figsize=(6,5))

plt.boxplot([data1, data2],

tick\_labels=['Nam', 'Nữ'], # từ Matplotlib 3.9

patch\_artist=True, # tô màu hộp

boxprops=dict(facecolor='lightblue', color='navy'),

medianprops=dict(color='red', linewidth=2),

whiskerprops=dict(color='black'))

plt.title("So sánh chiều cao / Boxplot comparison")

plt.ylabel("Chiều cao (cm)")

plt.show()

**🔍 Giải thích**

| **Lệnh** | **Tác dụng** |
| --- | --- |
| tick\_labels=['Nam','Nữ'] | Tên nhóm trục X |
| patch\_artist=True | Cho phép tô màu hộp |
| boxprops | Màu và viền của hộp |
| medianprops | Đường trung vị (đỏ, to hơn) |
| whiskerprops | Đường râu (min–max) |

**🖼 Mô tả:**  
Hai hộp song song: Nam cao hơn Nữ, đường đỏ là trung vị, vài chấm ngoài = ngoại lệ.

### 📘 **Seaborn**

sns.boxplot(x="day", y="total\_bill", data=sns.load\_dataset("tips"),

palette="Set2", width=0.6)

plt.title("Boxplot tiền hóa đơn theo ngày / Tips dataset")

plt.show()

**Giải thích:**

* x, y là tên cột trong DataFrame.
* palette chọn bộ màu nhẹ (Set2, pastel, muted, ...).
* width thu hẹp hộp cho đẹp mắt.

**🖼 Mô tả:**  
4 hộp màu pastel tương ứng thứ 5 → CN, trung vị đỏ, ngoại lệ hiển thị bằng chấm.

### 🌈 **Làm đẹp nâng cao**

sns.set\_style('whitegrid')

sns.boxplot(x="day", y="tip", hue="sex", data=sns.load\_dataset("tips"),

palette="Set3", notch=True, fliersize=4)

plt.title("So sánh tip theo ngày và giới tính")

plt.legend(title='Giới tính')

plt.show()

**Giải thích:**

* hue='sex' → tách màu theo giới.
* notch=True → thêm khe tại trung vị → trực quan hơn.
* fliersize=4 → chỉnh kích thước điểm ngoại lệ.
* sns.set\_style('whitegrid') → nền sáng, có lưới mảnh.

## 3️⃣ **Bar Chart – Biểu đồ cột**

### 🎯 Mục đích

So sánh giá trị giữa các nhóm phân loại (rời rạc): ví dụ doanh thu theo tháng, số sinh viên từng khoa,...

### 📘 **Matplotlib**

categories = ['A', 'B', 'C', 'D']

values = [5, 7, 3, 4]

plt.figure(figsize=(7,5))

plt.bar(categories, values, color='lightcoral', edgecolor='black', alpha=0.8)

plt.title("Biểu đồ cột / Bar chart")

plt.xlabel("Nhóm")

plt.ylabel("Giá trị")

plt.grid(axis='y', linestyle=':', alpha=0.5)

plt.show()

**🔍 Giải thích**

| **Lệnh** | **Ý nghĩa** |
| --- | --- |
| plt.bar() | Cột thẳng đứng |
| color, edgecolor | Tô màu cột và viền |
| alpha=0.8 | Làm màu nhẹ |
| grid(axis='y') | Thêm lưới ngang, giúp đọc giá trị nhanh hơn |

**🖼 Mô tả:**  
4 cột màu đỏ nhạt, nền trắng, có đường lưới ngang nhẹ.

### 📘 **Seaborn**

data = sns.load\_dataset("tips")

sns.barplot(x="day", y="total\_bill", data=data, palette="Set2", ci=None)

plt.title("Trung bình tổng hóa đơn theo ngày / Barplot (Seaborn)")

plt.show()

**Giải thích:**

* ci=None → tắt cột lỗi (confidence interval).
* palette='Set2' → bảng màu đồng bộ với Seaborn.
* Tự động tính **trung bình** nếu nhiều giá trị trùng ngày.

### 🌈 **Làm đẹp nâng cao**

sns.barplot(x="day", y="tip", hue="sex", data=data, palette="pastel", edgecolor='black')

plt.title("Trung bình tiền tip theo giới và ngày")

plt.legend(title="Giới tính", loc="upper left")

plt.grid(axis='y', alpha=0.3)

plt.show()

**Giải thích:**

* hue='sex' → chia cột theo giới tính.
* edgecolor='black' → giúp tách các cột rõ ràng.
* loc='upper left' → đặt legend hợp lý.

## 💡 **Mẹo tổng hợp cho 3 loại biểu đồ**

| **Hành động** | **Lệnh / Thông số** | **Giải thích** |
| --- | --- | --- |
| Làm mịn màu | alpha=0.7 | Giảm độ chói |
| Dùng bảng màu nhã | palette='Set2' hoặc 'pastel' | Đẹp cho báo cáo |
| Thêm chú thích | plt.text(x, y, '...', ha='center') | Ghi số lên cột |
| Căn lề đẹp | plt.tight\_layout() | Tránh đè nhãn |
| Nền sáng | sns.set\_style('whitegrid') | Đẹp và dễ đọc |
| Kích thước hợp lý | plt.figure(figsize=(8,5)) | Giúp nhìn cân đối |

## ✅ **Tóm tắt nhanh: Khi nào dùng loại nào**

| **Biểu đồ** | **Dùng khi** | **Thư viện khuyên dùng** |
| --- | --- | --- |
| **Histogram** | Dữ liệu liên tục (phân bố) | plt.hist hoặc sns.histplot |
| **Boxplot** | So sánh phân phối, tìm ngoại lệ | sns.boxplot |
| **Bar Chart** | So sánh giá trị trung bình giữa nhóm | sns.barplot |

Tôi có thể làm **phần 2 mở rộng**:

“Cách kết hợp biểu đồ này với thống kê mô tả (mean, median, std) và thêm annotation trực tiếp vào hình”.  
Ví dụ: ghi giá trị trung bình ngay trên Boxplot, hoặc ghi phần trăm trên cột Histogram.

Bạn có muốn tôi làm tiếp phần đó không (gọi là “vẽ biểu đồ có chú thích thống kê & annotation chuyên nghiệp”)?

##### Bạn đã nói:

có

###### ChatGPT đã nói:

Rõ. Dưới đây là hướng dẫn **“Vẽ biểu đồ có chú thích thống kê & annotation chuyên nghiệp”**,  
song ngữ 🇻🇳 / 🇬🇧, tập trung vào **Histogram, Boxplot, và Bar chart**,  
với chú thích **giá trị trung bình, trung vị, phần trăm, ngoại lệ,…**.

# 🎯 **PHẦN MỞ RỘNG – CHÚ THÍCH & ANNOTATION TRONG BIỂU ĐỒ**

## 🧮 **1️⃣ Histogram có chú thích trung bình, độ lệch chuẩn, và giá trị**

### 📘 **Matplotlib**

import matplotlib.pyplot as plt

import numpy as np

from scipy.stats import norm

# Sinh dữ liệu

data = np.random.normal(170, 10, 200)

plt.figure(figsize=(8,5))

# Vẽ histogram

plt.hist(data, bins=15, color='lightblue', edgecolor='black', alpha=0.8, density=True)

# Tính toán thống kê

mean = data.mean()

std = data.std()

x = np.linspace(mean - 3\*std, mean + 3\*std, 100)

plt.plot(x, norm.pdf(x, mean, std), color='red', linewidth=2, label='Đường chuẩn')

# Thêm chú thích

plt.axvline(mean, color='green', linestyle='--', linewidth=2, label=f'Trung bình = {mean:.2f}')

plt.axvline(mean + std, color='gray', linestyle=':', linewidth=1)

plt.axvline(mean - std, color='gray', linestyle=':', linewidth=1)

plt.title("Histogram với đường chuẩn & trung bình")

plt.xlabel("Chiều cao (cm)")

plt.ylabel("Mật độ xác suất")

plt.legend()

plt.grid(alpha=0.3)

# Ghi text trực tiếp

plt.text(mean + 1, 0.03, f"σ = {std:.2f}", color='black', fontsize=10)

plt.tight\_layout()

plt.show()

**🔍 Giải thích**

| **Dòng** | **Chức năng** |
| --- | --- |
| plt.axvline() | Vẽ đường thẳng đứng chỉ ra giá trị trung bình và độ lệch chuẩn |
| plt.text() | Ghi chú trực tiếp lên đồ thị |
| norm.pdf() | Hàm mật độ chuẩn, vẽ đường cong đỏ |
| density=True | Giúp histogram hiển thị dưới dạng xác suất thay vì số lượng |

**🖼 Mô tả:**  
Cột màu xanh nhạt, đường cong đỏ mượt, đường đứt xanh lá ở trung bình, ghi rõ “σ = …”.

### 📘 **Seaborn**

import seaborn as sns

sns.histplot(data, bins=15, kde=True, color='skyblue')

plt.axvline(mean, color='green', linestyle='--', label=f"Mean = {mean:.2f}")

plt.text(mean + 2, 0.03, f"Std = {std:.2f}", color='black')

plt.legend()

plt.title("Histogram (Seaborn) với chú thích trung bình")

plt.show()

**Giải thích:**

* kde=True thêm đường mật độ mượt.
* axvline() và text() thêm chú thích rõ ràng ngay trên đồ thị.

## 📦 **2️⃣ Boxplot có chú thích trung vị và khoảng tứ phân vị (IQR)**

### 📘 **Matplotlib**

data = np.random.normal(170, 10, 100)

plt.figure(figsize=(5,5))

box = plt.boxplot(data, patch\_artist=True, boxprops=dict(facecolor='lightblue'))

plt.title("Boxplot có chú thích thống kê")

plt.ylabel("Chiều cao (cm)")

# Tính toán thống kê

median = np.median(data)

q1 = np.percentile(data, 25)

q3 = np.percentile(data, 75)

iqr = q3 - q1

# Thêm đường và text

plt.axhline(median, color='red', linestyle='--', label=f"Median = {median:.1f}")

plt.text(1.05, median, f"{median:.1f}", color='red', va='center')

plt.text(1.05, q3, f"Q3 = {q3:.1f}", color='blue', va='center')

plt.text(1.05, q1, f"Q1 = {q1:.1f}", color='blue', va='center')

plt.legend()

plt.tight\_layout()

plt.show()

**🔍 Giải thích**

| **Dòng** | **Chức năng** |
| --- | --- |
| np.percentile() | Tính phần trăm vị 25 và 75 |
| iqr = q3 - q1 | Khoảng tứ phân vị |
| axhline() | Đường ngang thể hiện median |
| text() | Ghi Q1, Q3, median bên cạnh hộp |

**🖼 Mô tả:**  
Một hộp xanh nhạt, đường đứt đỏ ngang qua median, có ghi “Q1 = …, Q3 = …”.

### 📘 **Seaborn**

tips = sns.load\_dataset("tips")

sns.boxplot(x="day", y="total\_bill", data=tips, palette="Set3", width=0.6)

plt.title("Boxplot với annotation trung bình")

# Tính trung bình theo ngày

group\_means = tips.groupby("day")["total\_bill"].mean()

# Vẽ chú thích trên từng hộp

for i, (day, val) in enumerate(group\_means.items()):

plt.text(i, val + 1, f"{val:.1f}", ha='center', color='black', fontsize=9)

plt.ylabel("Tổng hóa đơn ($)")

plt.show()

**Giải thích:**

* group\_means tính trung bình theo nhóm.
* for i, (day, val) → ghi chú trên từng hộp.
* ha='center' → căn giữa text với hộp.

**🖼 Mô tả:**  
4 hộp màu pastel, mỗi hộp có ghi trung bình (số) ngay trên nắp hộp.

## 🧱 **3️⃣ Bar chart có phần trăm & annotation trên cột**

### 📘 **Matplotlib**

categories = ['A', 'B', 'C', 'D']

values = [20, 35, 30, 15]

plt.figure(figsize=(7,5))

bars = plt.bar(categories, values, color='salmon', edgecolor='black')

plt.title("Biểu đồ cột với chú thích phần trăm")

plt.xlabel("Nhóm")

plt.ylabel("Giá trị (%)")

# Ghi chú phần trăm trên cột

for bar in bars:

yval = bar.get\_height()

plt.text(bar.get\_x() + bar.get\_width()/2, yval + 1, f"{yval:.0f}%",

ha='center', va='bottom', fontsize=10)

plt.ylim(0, max(values) + 10)

plt.tight\_layout()

plt.show()

**🔍 Giải thích**

| **Lệnh** | **Tác dụng** |
| --- | --- |
| bars = plt.bar(...) | Lưu đối tượng cột để chỉnh sau |
| bar.get\_height() | Lấy chiều cao mỗi cột |
| plt.text() | Ghi phần trăm ngay trên đỉnh |
| va='bottom' | Căn chữ ngay phía trên đầu cột |
| ylim() | Chừa khoảng trống để text không bị cắt |

**🖼 Mô tả:**  
4 cột đỏ nhạt, có phần trăm ghi rõ ràng trên đầu (20%, 35%, …).

### 📘 **Seaborn**

data = sns.load\_dataset("tips")

avg\_tip = data.groupby("day")["tip"].mean().reset\_index()

sns.barplot(x="day", y="tip", data=avg\_tip, palette="Set2", edgecolor="black")

plt.title("Bar chart với annotation trung bình tiền tip")

# Ghi chú trung bình

for i, val in enumerate(avg\_tip["tip"]):

plt.text(i, val + 0.05, f"{val:.2f}", ha='center', color='black')

plt.ylabel("Trung bình tip ($)")

plt.show()

**Giải thích:**

* reset\_index() giúp chuyển GroupBy về DataFrame.
* Ghi số trung bình trực tiếp trên cột → rõ ràng, đẹp mắt.

**🖼 Mô tả:**  
4 cột màu pastel, mỗi cột có số trung bình (như 2.80, 3.10) hiển thị ngay phía trên.

## 💡 **Mẹo làm biểu đồ có chú thích chuyên nghiệp**

| **Hành động** | **Cách làm** | **Mục đích** |
| --- | --- | --- |
| Giữ không gian cho chữ | plt.ylim(0, max(values)+padding) | Tránh chữ bị cắt |
| Dùng màu mềm | 'pastel', 'Set2', 'muted' | Giúp dễ nhìn khi thêm chữ |
| Căn giữa text | ha='center', va='bottom' | Đẹp và gọn |
| Tạo nền nhẹ | sns.set\_style('whitegrid') | Giúp text dễ đọc |
| Xuất ảnh đẹp | plt.savefig('plot.png', dpi=300, bbox\_inches='tight') | Đưa vào báo cáo |

## ✅ **Tổng kết phần mở rộng**

| **Loại biểu đồ** | **Dạng chú thích** | **Hàm chính** | **Mục đích** |
| --- | --- | --- | --- |
| Histogram | Mean, Std, PDF | axvline, text, plot | Hiển thị phân bố & độ lệch |
| Boxplot | Median, Q1, Q3 | percentile, text | Hiểu vị trí trung tâm & ngoại lệ |
| Bar chart | Giá trị, % | bar.get\_height, text | So sánh trực quan giữa nhóm |

# 📊 **PHẦN CUỐI – TỰ ĐỘNG TÍNH THỐNG KÊ & GẮN CHÚ THÍCH TRÊN BIỂU ĐỒ**

## 🧩 **1️⃣ Bar Chart – Trung bình và số lượng mẫu tự động**

### 🎯 Mục tiêu

Tính **trung bình và số mẫu** mỗi nhóm, vẽ bar chart hiển thị cả hai trên hình.

### 📘 **Seaborn + Pandas**

import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

# Dữ liệu mẫu

tips = sns.load\_dataset("tips")

# Tính thống kê: trung bình và số lượng mẫu

stats = tips.groupby("day")["total\_bill"].agg(['mean', 'count']).reset\_index()

# Vẽ biểu đồ

plt.figure(figsize=(7,5))

sns.barplot(x="day", y="mean", data=stats, palette="Set2", edgecolor='black')

# Thêm chú thích động

for i, row in stats.iterrows():

plt.text(i, row['mean'] + 0.5,

f"{row['mean']:.1f}\n(n={row['count']})",

ha='center', va='bottom', fontsize=10, color='black')

plt.title("Trung bình tổng hóa đơn & số mẫu mỗi ngày")

plt.xlabel("Ngày trong tuần")

plt.ylabel("Giá trị trung bình ($)")

plt.tight\_layout()

plt.show()

**🔍 Giải thích**

| **Lệnh** | **Tác dụng** |
| --- | --- |
| groupby("day") | Gom dữ liệu theo ngày |
| .agg(['mean','count']) | Tính trung bình và số lượng mẫu |
| reset\_index() | Đưa groupby về DataFrame thường |
| plt.text() | Ghi chú mean và (n=…) lên cột |
| \n | Xuống dòng giữa mean và n |

**🖼 Mô tả:**  
4 cột pastel (Thứ 5–CN), trên mỗi cột có ghi

22.8

(n=62)

## 🧩 **2️⃣ Boxplot – Hiển thị mean, median, std tự động**

### 📘 **Seaborn + NumPy**

import numpy as np

plt.figure(figsize=(7,5))

sns.boxplot(x="day", y="total\_bill", data=tips, palette="pastel")

# Tính thống kê từng nhóm

group\_stats = tips.groupby("day")["total\_bill"].agg(['mean', 'median', 'std'])

# Ghi chú tự động lên từng hộp

for i, (day, row) in enumerate(group\_stats.iterrows()):

plt.text(i, row['mean'] + 1, f"Mean={row['mean']:.1f}", ha='center', color='red', fontsize=9)

plt.text(i, row['median'] - 2, f"Med={row['median']:.1f}", ha='center', color='blue', fontsize=9)

plt.text(i, row['mean'] + 4, f"σ={row['std']:.1f}", ha='center', color='black', fontsize=8)

plt.title("Boxplot với chú thích Mean / Median / Std")

plt.ylabel("Tổng hóa đơn ($)")

plt.tight\_layout()

plt.show()

**🔍 Giải thích**

| **Dòng** | **Tác dụng** |
| --- | --- |
| .agg(['mean','median','std']) | Tính 3 thống kê tự động |
| enumerate() | Lấy chỉ số hộp để đặt chữ đúng vị trí |
| text() | Ghi chú mean (đỏ), median (xanh), std (đen) |
| ha='center' | Căn giữa với hộp dữ liệu |

**🖼 Mô tả:**  
Trên mỗi hộp pastel có 3 dòng nhỏ:  
Mean=24.2, Med=20.5, σ=10.3

## 🧩 **3️⃣ Histogram – Tự động thêm chú thích Mean, Median, Std**

### 📘 **Matplotlib + NumPy**

data = tips["total\_bill"]

mean = data.mean()

median = data.median()

std = data.std()

plt.figure(figsize=(8,5))

plt.hist(data, bins=15, color='lightblue', edgecolor='black', alpha=0.7)

plt.axvline(mean, color='red', linestyle='--', linewidth=2, label=f"Mean = {mean:.1f}")

plt.axvline(median, color='green', linestyle=':', linewidth=2, label=f"Median = {median:.1f}")

plt.axvline(mean + std, color='gray', linestyle='--', linewidth=1)

plt.axvline(mean - std, color='gray', linestyle='--', linewidth=1)

# Ghi text

plt.text(mean + 1, 5, f"Std = {std:.1f}", color='black', fontsize=10)

plt.title("Histogram với chú thích Mean / Median / Std")

plt.xlabel("Tổng hóa đơn ($)")

plt.ylabel("Số lượng mẫu")

plt.legend()

plt.tight\_layout()

plt.show()

**🔍 Giải thích**

| **Lệnh** | **Ý nghĩa** |
| --- | --- |
| axvline() | Đường dọc chỉ ra mean, median, std |
| label=f"Mean = {mean:.1f}" | Tự động format số 1 chữ thập phân |
| text() | Ghi giá trị std trực tiếp |
| legend() | Hiển thị chú thích màu |

**🖼 Mô tả:**  
Histogram xanh nhạt, đường đỏ (mean), đường xanh (median), vùng xám ±σ.

## 💡 **4️⃣ Mẹo chuyên nghiệp**

| **Tình huống** | **Kỹ thuật đề xuất** | **Giải thích** |
| --- | --- | --- |
| Văn bản chồng chéo | plt.text(..., fontsize=9, rotation=0) | Giảm cỡ, xoay chữ |
| Chữ bị đè viền | plt.tight\_layout() hoặc bbox\_inches='tight' | Căn viền hợp lý |
| Màu dễ nhìn | palette='Set2', 'pastel', 'husl' | Tăng độ tương phản dịu |
| Xuất ảnh sắc nét | plt.savefig("chart.png", dpi=300) | Dùng trong báo cáo |
| Ghi chú nhiều giá trị | f"{mean:.1f}\nσ={std:.1f}" | Xuống dòng dễ đọc |

## ✅ **Kết luận**

| **Biểu đồ** | **Chú thích tự động** | **Thư viện** | **Ứng dụng chính** |
| --- | --- | --- | --- |
| Bar Chart | Mean + Count | Seaborn + Pandas | So sánh nhóm |
| Boxplot | Mean + Median + Std | Seaborn + NumPy | So sánh phân phối |
| Histogram | Mean + Median + Std | Matplotlib | Quan sát phân bố |