THỰC HÀNH: PHÂN TÍCH KHÁM PHÁ DỮ LIỆU

# 1. THỐNG KÊ MÔ TẢ

# 1.1. Thống kê mô tả là gì? Nó khác gì với thống kê suy luận (inferential statistics)?

* Thống kê mô tả (Descriptive Statistics): Là các phương pháp dùng để thu thập, tóm tắt, trình bày và mô tả các đặc điểm chính của một tập hợp dữ liệu. Mục tiêu là để hiểu rõ về dữ liệu đang có (ví dụ: điểm trung bình, mức độ phân tán). Các chỉ số này chỉ cung cấp thông tin về mẫu dữ liệu hiện tại mà không đưa ra bất kỳ kết luận nào về tổng thể lớn hơn.
* Thống kê suy luận (Inferential Statistics): Là các phương pháp sử dụng dữ liệu từ một mẫu để suy luận, dự đoán, ước lượng hoặc kiểm định giả thuyết về một tổng thể lớn hơn mà mẫu đó đại diện.

|  |  |  |
| --- | --- | --- |
| Đặc điểm | Thống kê mô tả | Thống kê suy luận |
| Mục đích | Mô tả và tóm tắt dữ liệu. | Rút ra kết luận về tổng thể từ mẫu. |
| Phạm vi | Chỉ tập trung vào dữ liệu đã thu thập. | Mở rộng kết quả từ mẫu ra tổng thể. |
| Công cụ | Trung bình, trung vị, độ lệch chuẩn, biểu đồ. | Kiểm định t, ANOVA, hồi quy, ước lượng khoảng tin cậy. |

## 1.2. Các thước đo thống kê mô tả chính (ví dụ: trung bình, trung vị, phương sai, độ lệch chuẩn) được sử dụng để làm gì? Trong trường hợp nào thì nên dùng trung vị thay vì trung bình?

**Các thước đo thống kê mô tả chính và ý nghĩa**

* Trung bình (Mean): Giá trị trung bình cộng. Dùng khi dữ liệu có phân phối đối xứng (phân phối chuẩn) và không có giá trị ngoại lai (outliers) đáng kể. Dùng để mô tả xu hướng trung tâm của dữ liệu.
* Trung vị (Median): Giá trị nằm giữa tập dữ liệu đã sắp xếp (chia dữ liệu thành hai nửa bằng nhau). Dùng để biểu diễn giá trị trung tâm ổn định, ít bị ảnh hưởng bởi ngoại lệ.
* Phương sai (Variance): Đo mức độ phân tán của dữ liệu so với trung bình. Giá trị càng lớn → dữ liệu càng biến động.
* Độ lệch chuẩn (Standard Deviation, SD): Căn bậc hai của phương sai, cho biết dữ liệu thường lệch trung bình bao nhiêu đơn vị. Rất phổ biến vì giữ cùng đơn vị đo với dữ liệu gốc, dễ hiểu hơn phương sai.

**Khi nào dùng trung vị thay vì trung bình?**

Nên dùng trung vị trong các trường hợp:

* **Dữ liệu có ngoại lệ**: Ví dụ thu nhập, giá nhà… vài giá trị cực lớn/lớn bất thường sẽ làm trung bình bị lệch. Trung vị sẽ phản ánh trung tâm thực tế tốt hơn.
* **Dữ liệu phân bố lệch**: Ví dụ thời gian hoàn thành một tác vụ, điểm thi có nhiều học sinh điểm thấp nhưng vài người điểm cực cao. Trung vị cho thấy vị trí trung tâm hợp lý hơn.

## 1.3. Làm thế nào để xác định phân bố của một tập dữ liệu? Các loại phân bố phổ biến là gì (ví dụ: phân bố chuẩn, lệch trái, lệch phải)?

**Cách xác định phân bố của tập dữ liệu**

* Trực quan hóa dữ liệu: Histogram, Boxplot, Density plot.
* Các chỉ số thống kê: Mean và Median, Skewness.

**Các loại phân bố phổ biến**

* Phân bố chuẩn (Normal distribution)
* Phân bố lệch phải (Right-skewed / Positive skew)
* Phân bố lệch trái (Left-skewed / Negative skew)
* Phân bố đều (Uniform distribution)
* Phân bố nhọn hoặc bẹt (Kurtosis)

## 1.4. Độ lệch chuẩn và phạm vi (range) có ý nghĩa gì trong việc đánh giá sự phân tán của dữ liệu?

Phạm vi (Range) cho biết dữ liệu trải dài từ giá trị nhỏ nhất đến lớn nhất, giúp ta thấy được khoảng biến động tổng quát. Tuy nhiên, nó rất nhạy cảm với giá trị ngoại lệ, nên thường chỉ dùng để có cái nhìn sơ bộ. Độ lệch chuẩn (Standard Deviation) cho biết mức độ phân tán trung bình của các điểm dữ liệu so với giá trị trung bình, phản ánh rõ ràng hơn sự dao động của toàn bộ tập dữ liệu. Nếu SD nhỏ, dữ liệu tập trung quanh trung bình; nếu SD lớn, dữ liệu phân tán rộng.

## 1.5. Sự khác biệt giữa các thước đo như Q1, Q2, Q3 trong biểu đồ hộp (boxplot) là gì?

* **Q1 (First Quartile – tứ phân vị thứ nhất)**:  
  Là giá trị tại **25% dữ liệu nhỏ nhất**. Nó cho biết ranh giới giữa nhóm dữ liệu thấp nhất và phần còn lại.
* **Q2 (Second Quartile – tứ phân vị thứ hai, hay Median)**:  
  Là **trung vị** – giá trị nằm chính giữa tập dữ liệu (50%). Nó chia dữ liệu thành hai nửa bằng nhau.
* **Q3 (Third Quartile – tứ phân vị thứ ba)**:  
  Là giá trị tại **75% dữ liệu nhỏ nhất**. Nó cho biết ranh giới giữa nhóm dữ liệu cao nhất và phần còn lại.

## 1.6. Làm thế nào để xử lý giá trị thiếu (missing values) trước khi tính toán các chỉ số thống kê mô tả?

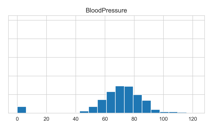
* **Xóa dữ liệu**: nếu số lượng giá trị thiếu ít, có thể xóa hàng chứa missing hoặc xóa cột nếu cột đó có quá nhiều giá trị trống.
* **Thay thế bằng thống kê đơn giản**: điền giá trị thiếu bằng trung bình (mean), trung vị (median) hoặc mode (giá trị xuất hiện nhiều nhất) tùy thuộc vào loại dữ liệu và phân bố.
* **Sử dụng kỹ thuật nâng cao**: dự đoán giá trị thiếu bằng mô hình (hồi quy, KNN, random forest) hoặc dùng multiple imputation để tạo nhiều bộ dữ liệu giả định rồi tổng hợp kết quả.
* **Bỏ qua khi tính toán**: một số công cụ cho phép bỏ qua missing values trong quá trình tính toán.

## 1.7. Bạn có thể giải thích cách đọc và diễn giải một biểu đồ histogram hoặc boxplot từ dữ liệu thực tế không?

Histogram là biểu đồ tần suất, chia dữ liệu thành các khoảng (bins).

Khi nhìn histogram, ta chú ý:

* **Hình dạng phân bố**:
  + Đối xứng, hình chuông → dữ liệu gần chuẩn.
  + Lệch phải (đuôi dài bên phải) → nhiều giá trị nhỏ, một số giá trị rất lớn.
  + Lệch trái (đuôi dài bên trái) → nhiều giá trị lớn, một số giá trị rất nhỏ.
* **Đỉnh (mode)**: có 1 đỉnh → phân bố đơn đỉnh, nhiều đỉnh → có thể dữ liệu gồm nhiều nhóm khác nhau.
* **Độ phân tán**: histogram trải rộng → dữ liệu biến động lớn, tập trung hẹp → biến động nhỏ.
* **Outlier**: thường hiện ra như những cột tách biệt ở hai đầu.



**Như biểu đồ ở trên, dữ liệu tương đối chuẩn nhưng, có ngoại lệ ở giá trị 0, phân bố đơn đỉnh, phân tán không rộng.**

## 1.8. Khi gặp một tập dữ liệu có giá trị ngoại lai (outliers), bạn sẽ xử lý chúng như thế nào trước khi thực hiện thống kê mô tả?

Khi gặp dữ liệu có giá trị ngoại lai (outliers), ta cần xử lý để tránh làm sai lệch kết quả thống kê mô tả. Các bước thường dùng gồm:

* Xác định outlier bằng boxplot, z-score hoặc dựa trên ngữ cảnh thực tế.
* Kiểm tra nguyên nhân: nếu do nhập sai hay lỗi đo lường thì loại bỏ; nếu là giá trị hiếm nhưng hợp lý thì cân nhắc giữ lại.
* Loại bỏ dữ liệu: áp dụng khi số lượng outlier ít và chắc chắn là sai.
* Chuyển đổi dữ liệu: dùng log, căn bậc hai hoặc winsorization để giảm ảnh hưởng của outlier.
* Dùng thước đo bền vững: thay mean và standard deviation bằng median và IQR, vốn ít nhạy cảm với ngoại lai.

# 2. XỬ LÝ VÀ TRỰC QUAN HÓA DỮ LIỆU

## 2.1. Trực quan hóa dữ liệu có vai trò gì trong phân tích dữ liệu? Tại sao nó quan trọng trong khám phá dữ liệu (EDA)?

Trực quan hóa dữ liệu có vai trò giúp ta nhìn thấy xu hướng, mô hình và mối quan hệ trong dữ liệu một cách trực quan mà các con số thuần túy khó thể hiện được. Trong quá trình khám phá dữ liệu (EDA), trực quan hóa đặc biệt quan trọng vì nó hỗ trợ phát hiện dữ liệu thiếu, ngoại lệ, phân bố bất thường và đưa ra giả thuyết ban đầu cho phân tích sâu hơn.

## 2.2. Các loại biểu đồ phổ biến (như histogram, scatter plot, boxplot, bar chart) được sử dụng trong các trường hợp nào?

Histogram thường dùng để quan sát phân bố của dữ liệu số; scatter plot để tìm hiểu mối quan hệ giữa hai biến liên tục; boxplot để phát hiện ngoại lệ và đánh giá độ phân tán; bar chart để so sánh tần suất hoặc giá trị giữa các nhóm dữ liệu phân loại.

## 2.3. Làm thế nào để chọn loại biểu đồ phù hợp với đặc điểm của dữ liệu (ví dụ: dữ liệu phân loại, dữ liệu số, dữ liệu thời gian)?

Việc chọn biểu đồ phụ thuộc vào loại dữ liệu: với dữ liệu phân loại thường dùng bar chart hoặc pie chart; với dữ liệu số thường dùng histogram, boxplot, scatter plot; với dữ liệu theo thời gian thì line chart là phổ biến để quan sát xu hướng theo chuỗi thời gian.

## 2.4. Sự khác biệt giữa các thư viện trực quan hóa trong Python như Matplotlib, Seaborn và Plotly là gì?

Matplotlib là thư viện cơ bản, linh hoạt nhưng cú pháp hơi dài; Seaborn xây dựng trên Matplotlib, cung cấp biểu đồ đẹp sẵn và dễ dùng cho thống kê; Plotly hỗ trợ biểu đồ tương tác, phù hợp cho dashboard hoặc khi cần trực quan hóa động.

## 2.5. Những nguyên tắc thiết kế nào cần tuân thủ để tạo ra một biểu đồ trực quan hóa dễ hiểu và hiệu quả?

Khi thiết kế biểu đồ, cần đảm bảo rõ ràng và dễ đọc: chọn loại biểu đồ phù hợp, đặt tiêu đề và nhãn trục đầy đủ, không dùng quá nhiều màu sắc gây rối, và tập trung vào thông tin chính cần truyền tải.

## 2.6. Làm thế nào để tạo một biểu đồ đơn giản như histogram hoặc bar chart bằng Matplotlib? Bạn có thể chia sẻ đoạn code mẫu không?

import matplotlib.pyplot as plt

data = [1, 2, 2, 3, 3, 3, 4, 4, 5]

plt.hist(data, bins=5)

plt.title("Histogram Example")

plt.show()

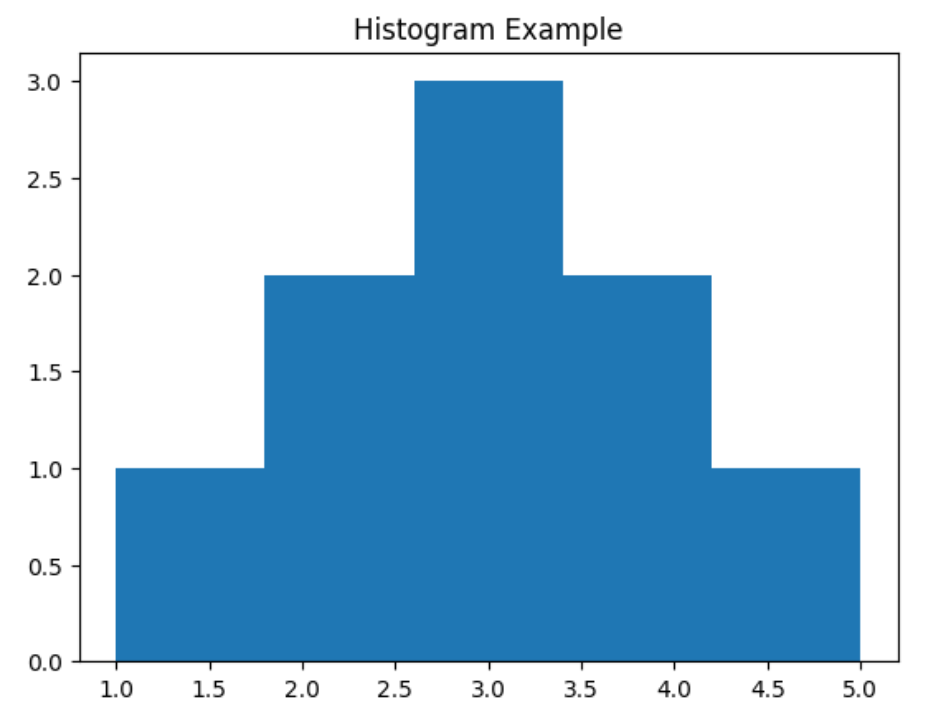
categories = ['A', 'B', 'C']

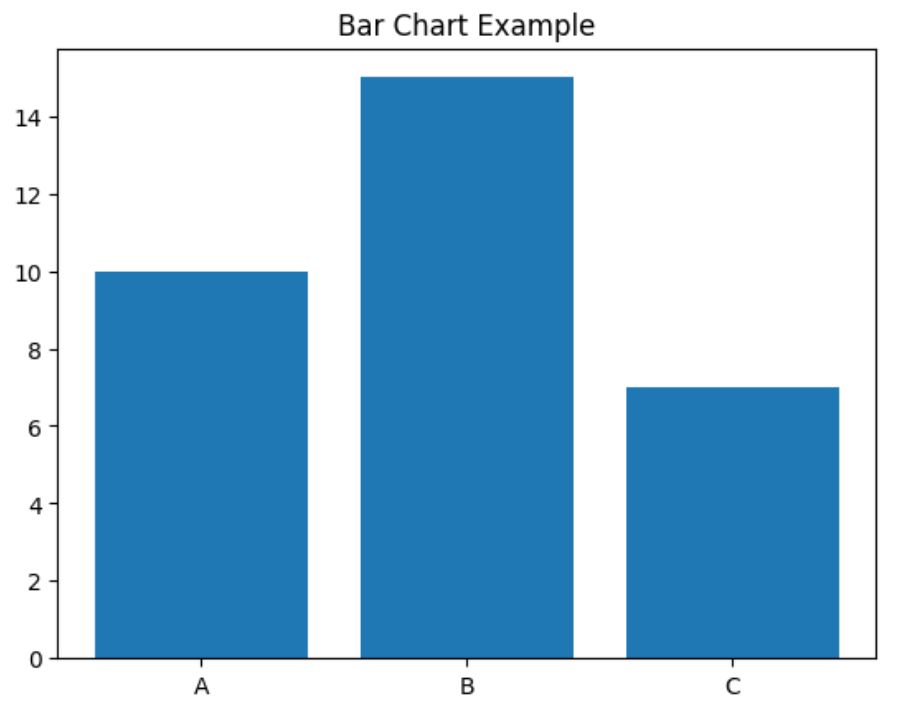
values = [10, 15, 7]

plt.bar(categories, values)

plt.title("Bar Chart Example")

plt.show()





## 2.7. Làm thế nào để xuất biểu đồ từ Python ra các định dạng như PNG, PDF hoặc HTML để sử dụng trong báo cáo?

Để xuất biểu đồ, Matplotlib hỗ trợ **plt.savefig("tenfile.png")** hoặc các định dạng khác như **plt.savefig("tenfile.pdf").** Với Plotly, có thể xuất HTML tương tác bằng **fig.write\_html("tenfile.html")**.

# 3. PHÂN TÍCH ĐƠN BIẾN VÀ HAI BIẾN

## 3.1. Phân tích đơn biến (univariate analysis) là gì? Nó khác gì với phân tích hai biến (bivariate analysis) trong khám phá dữ liệu?

Phân tích đơn biến (Univariate Analysis) là bước phân tích tập trung vào một biến duy nhất để mô tả đặc điểm của nó. Mục tiêu là hiểu được phân bố, xu hướng trung tâm và mức độ phân tán của biến đó. Ví dụ: kiểm tra điểm thi của sinh viên bằng biểu đồ histogram, tính trung bình, trung vị, độ lệch chuẩn,…

Phân tích hai biến (Bivariate Analysis) thì xem xét mối quan hệ giữa hai biến cùng lúc, để tìm hiểu chúng có liên hệ hay phụ thuộc lẫn nhau không. Ví dụ: phân tích mối quan hệ giữa số giờ học và điểm thi, dùng scatter plot hoặc tính hệ số tương quan.

**Sự khác biệt**:

* Đơn biến: chỉ mô tả đặc điểm của một biến, không xét đến các biến khác.
* Hai biến: tập trung vào mối quan hệ giữa hai biến, có thể là giữa hai biến định lượng, hai biến định tính, hoặc một biến định tính và một biến định lượng.

## 3.2. Các thước đo thống kê nào thường được sử dụng trong phân tích đơn biến (ví dụ: trung bình, trung vị, mode, độ lệch chuẩn)?

* Trung bình (Mean): Giá trị đại diện của dữ liệu.
* Trung vị (Median): Điểm chính giữa khi sắp xếp dữ liệu.
* Mode: Giá trị xuất hiện nhiều nhất.
* Độ lệch chuẩn (Standard Deviation): Mức độ phân tán của dữ liệu.
* Min, Max, Quartile, IQR: Giúp hiểu phạm vi và sự phân bố.

## 3.3. Trong phân tích hai biến, làm thế nào để xác định mối quan hệ giữa hai biến (ví dụ: tương quan, nhân quả)?

* Tính hệ số tương quan (Correlation, như Pearson hoặc Spearman).
* Quan sát trực quan qua scatter plot hoặc heatmap.
* Kiểm định thống kê (ví dụ: Chi-square test cho biến phân loại).

## 3.4. Sự khác biệt giữa tương quan (correlation) và hiệp biến (covariance) trong phân tích hai biến là gì?

* Covariance: Đo lường hai biến thay đổi cùng nhau thế nào (cùng tăng, cùng giảm). Đơn vị phụ thuộc vào dữ liệu.
* Correlation: Phiên bản chuẩn hóa của covariance, giá trị từ -1 → 1, dễ so sánh, không phụ thuộc đơn vị.

## 3.5. Khi nào nên sử dụng biểu đồ trực quan hóa trong phân tích đơn biến so với phân tích hai biến?

* Đơn biến: Khi muốn xem phân bố, tần suất, đặc điểm của 1 biến (histogram, bar chart, boxplot).
* Hai biến: Khi muốn xem quan hệ giữa 2 biến (scatter plot, heatmap, boxplot theo nhóm).

## 3.6. Đoạn code mẫu để tạo biểu đồ scatter plot hoặc heatmap để phân tích mối quan hệ giữa hai biến?

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

# ví dụ data

df = pd.DataFrame({

"x": [1,2,3,4,5,6],

"y": [2,4,5,4,5,7]

})

plt.scatter(df["x"], df["y"])

plt.xlabel("X")

plt.ylabel("Y")

plt.title("Scatter plot")

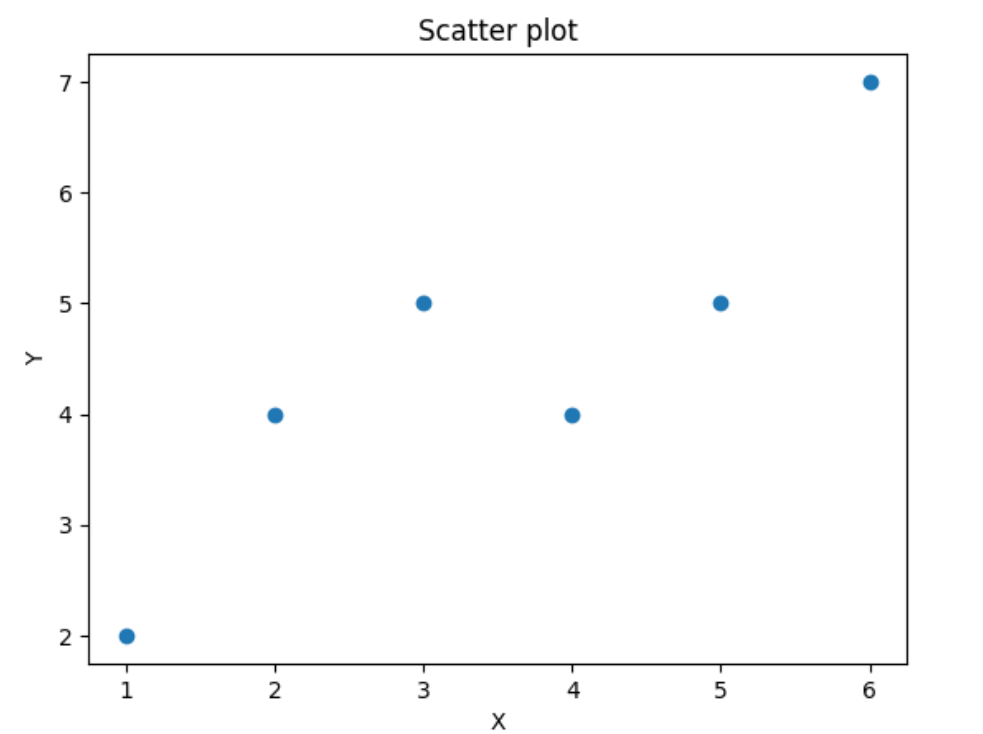
plt.show()

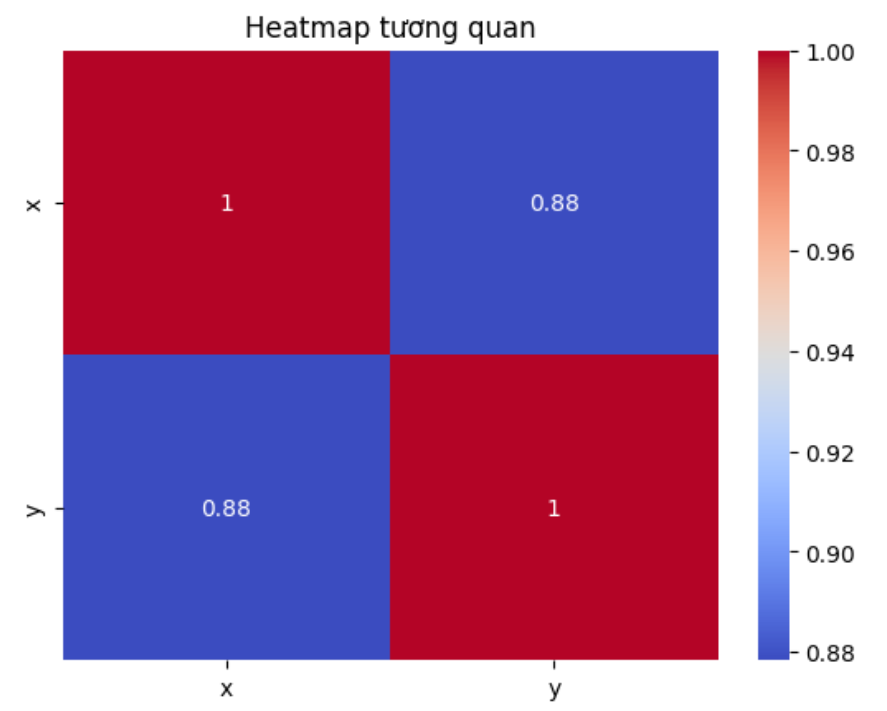
corr = df.corr()

sns.heatmap(corr, annot=True, cmap="coolwarm")

plt.title("Heatmap tương quan")

plt.show()





## 3.7. Làm thế nào để trực quan hóa mối quan hệ giữa một biến số và một biến phân loại bằng biểu đồ boxplot hoặc violin plot trong Python?

import seaborn as sns

# giả sử df có biến 'Category' (phân loại) và 'Value' (số)

sns.boxplot(x="Category", y="Value", data=df)

plt.title("Boxplot")

plt.show()

sns.violinplot(x="Category", y="Value", data=df)

plt.title("Violin plot")

plt.show()