Phân tích vai trò và ứng dụng của trực quan hóa dữ liệu trong EDA

I. Vai trò của trực quan hóa dữ liệu trong phân tích và khám phá dữ liệu (EDA)

Trực quan hóa dữ liệu (Data Visualization) là bước quan trọng trong quy trình phân tích dữ liệu, đặc biệt trong giai đoạn khám phá dữ liệu (Exploratory Data Analysis – EDA).  
Việc biểu diễn dữ liệu bằng đồ thị giúp nhà phân tích nhận diện nhanh các xu hướng, mối quan hệ, và điểm bất thường (outliers) mà các phương pháp thống kê thuần túy khó thể hiện được.  
Trực quan hóa đóng vai trò là cầu nối giữa dữ liệu và hiểu biết, giúp:

Nắm bắt cấu trúc tổng thể và phân bố dữ liệu.

Phát hiện mối tương quan giữa các biến (dương hoặc âm).

Phát hiện sai lệch, dữ liệu thiếu hoặc ngoại lai.

Hỗ trợ xây dựng giả thuyết, chọn mô hình phù hợp.

Trình bày kết quả một cách trực quan, dễ hiểu cho người không chuyên về kỹ thuật.

Nhờ đó, trực quan hóa dữ liệu trở thành công cụ không thể thiếu trong quá trình phân tích và ra quyết định dựa trên dữ liệu.

II. Các loại biểu đồ phổ biến và ứng dụng

1. Histogram (biểu đồ tần suất)

Dùng để mô tả phân bố của một biến số liên tục.

Giúp xác định phân bố có chuẩn hay không, độ lệch (skewness), và có nhiều đỉnh (multimodal) hay không.

1. Boxplot (biểu đồ hộp)

Dùng để so sánh phân bố của dữ liệu giữa các nhóm.

Thể hiện giá trị trung vị, khoảng tứ phân vị (IQR) và điểm ngoại lai (outliers).

1. Scatter plot (biểu đồ phân tán)

Dùng để thể hiện mối quan hệ giữa hai biến số.

Giúp nhận diện mối tương quan tuyến tính hoặc phi tuyến, xu hướng và cụm điểm dữ liệu.

1. Bar chart (biểu đồ cột)

Dùng cho dữ liệu phân loại (categorical).

Giúp so sánh tần suất hoặc tổng giá trị giữa các nhóm.

1. Line chart (biểu đồ đường)

Phù hợp với dữ liệu theo thời gian (time series), giúp theo dõi xu hướng và biến động theo chu kỳ.

1. Heatmap (bản đồ nhiệt)

Dùng để hiển thị ma trận tương quan giữa các biến.

Màu sắc thể hiện cường độ của mối quan hệ.

III. Cách lựa chọn biểu đồ phù hợp với loại dữ liệu

|  |  |  |
| --- | --- | --- |
| Loại dữ liệu | Mục đích | Biểu đồ phù hợp |
| Biến số (numerical) | Phân bố, độ lệch, outliers | Histogram, Boxplot, Violin |
| Phân loại (categorical) | So sánh tần suất, tỷ lệ | Bar chart, Countplot |
| Số ↔ Số | Mối tương quan, xu hướng | Scatter plot, Line plot |
| Số ↔ Phân loại | So sánh trung bình, phân bố giữa nhóm | Boxplot, Violin |
| Dữ liệu thời gian | Xu hướng, mùa vụ | Line chart |
| Ma trận tương quan | Mối quan hệ nhiều biến | Heatmap |

Việc lựa chọn đúng loại biểu đồ sẽ giúp dữ liệu được thể hiện chính xác, trực quan và truyền tải thông tin hiệu quả nhất.

IV. So sánh giữa các thư viện trực quan hóa trong Python

|  |  |  |  |
| --- | --- | --- | --- |
| Thư viện | Ưu điểm | Hạn chế | Khi nên dùng |
| Matplotlib | Linh hoạt, mạnh mẽ, xuất ảnh chất lượng cao (PNG, PDF) | Cú pháp dài, khó tùy chỉnh thẩm mỹ | Khi cần biểu đồ tĩnh hoặc báo cáo in ấn |
| Seaborn | Dễ dùng, giao diện đẹp, tự động xử lý thống kê | Ít tương tác, tùy chỉnh sâu vẫn cần Matplotlib | Khi phân tích EDA nhanh, tạo biểu đồ đẹp cho báo cáo |
| Plotly | Tương tác cao, đẹp, có thể xuất HTML | File lớn, cần cài thêm kaleido để xuất ảnh | Khi cần biểu đồ tương tác, trình bày online |

V. Nguyên tắc thiết kế biểu đồ hiệu quả

1. Đơn giản – Dễ hiểu: Tránh quá nhiều màu, hiệu ứng rối mắt.
2. Chính xác: Không bóp méo tỷ lệ trục, không cắt trục làm sai lệch nhận thức.
3. Rõ ràng: Tiêu đề, nhãn trục, đơn vị phải đầy đủ.
4. Nhấn mạnh thông tin quan trọng: Dùng màu nổi bật, annotation hợp lý.
5. Nhất quán: Cùng loại dữ liệu → dùng cùng loại biểu đồ và thang màu.
6. Thân thiện thị giác: Dùng palette dễ nhìn, có độ tương phản phù hợp.

VI. Ví dụ tạo biểu đồ trong Python

1. Histogram (biểu đồ tần suất)

import matplotlib.pyplot as plt

import numpy as np

data = np.random.normal(50, 10, 1000)

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

plt.hist(data, bins=25, color='skyblue', edgecolor='black', alpha=0.8)

plt.title('Phân bố giá trị - Histogram')

plt.xlabel('Giá trị')

plt.ylabel('Tần suất')

plt.grid(alpha=0.3)

plt.tight\_layout()

plt.savefig('histogram.png', dpi=300)

plt.show()

2. Bar Chart (biểu đồ cột)

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

values = [230, 190, 320]

plt.bar(categories, values, color=['#2E86C1', '#AF7AC5', '#F1948A'])

plt.title('Doanh số theo sản phẩm')

plt.xlabel('Sản phẩm')

plt.ylabel('Số lượng bán')

plt.tight\_layout()

plt.savefig('bar\_chart.pdf')

plt.show()

3. Xuất biểu đồ ra các định dạng phổ biến

PNG (ảnh tĩnh): plt.savefig('bieudo.png', dpi=300)

PDF (ảnh vector): plt.savefig('bieudo.pdf')

HTML (tương tác - Plotly):

import plotly.express as px

fig = px.scatter(x=[1,2,3,4], y=[10,15,13,17], title='Biểu đồ tương tác')

fig.write\_html('bieudo\_tuong\_tac.html')