Cơ Sở Lý Luận và Phương Pháp Luận Tạo Figure Bằng Python

Hà Minh Tuấn

Ngày 18 tháng 11 năm 2025

1 Giới Thiệu

Trong era của Big Data, khả năng truyền đạt thông tin phức tạp thông qua hình ảnh là một kỹ năng không thể thiếu. Theo triết lý thiết kế của Google, việc tạo visualizations không chỉ là một kỹ thuật lập trình mà còn là một quy trình khoa học có các nguyên tắc cơ bản rõ ràng.

2 Cơ Sở Lý Luận

2.1 Nguyên Tắc Truthfulness (Chân Thật)

Dữ liệu phải được biểu diễn một cách chính xác, không bao giờ sai lệch hoặc thao túng trục tọa độ để tạo ấn tượng sai lệch.

2.2 Nguyên Tắc Functionality (Chức Năng)

Mỗi phần tử trong figure phải có mục đích cụ thể. Decoration mà không mang lại giá trị thông tin phải bị loại bỏ.

2.3 Nguyên Tắc Beauty (Thẩm Mỹ)

Một visualization đẹp không chỉ giúp dễ hiểu mà còn khuyến khích người xem tương tác và khám phá dữ liệu.

3 Phương Pháp Luận: Quy Trình Tạo Figure

3.1 Bước 1: Xác Định Mục Tiêu

Trước khi viết code, phải trả lời các câu hỏi: Dữ liệu nào cần visualized? Ai là audience? Thông điệp chính là gì?

3.2 Bước 2: Chọn Loại Biểu Đồ Phù Hợp

- **Bar Chart**: So sánh giá trị trong các danh mục
- **Line Chart**: Theo dõi xu hướng theo thời gian
- **Scatter Plot**: Tìm mối tương quan giữa hai biến liên tục
- **Heatmap**: Hiển thị dữ liệu ma trận với mã màu
- **Box Plot**: Phân tích phân bố và outliers

3.3 Bước 3: Thiết Kế Visual Encoding

Visual encoding là quá trình chuyển đổi dữ liệu thành các thuộc tính hình ảnh (vị trí, kích thước, màu sắc, hình dạng).

3.4 Bước 4: Tối Ưu Hóa Readability

Đảm bảo labels, legends, và titles rõ ràng, dễ đọc.

4 Ví Dụ Thực Hành

4.1 Ví Dụ 1: Bar Chart Để So Sánh

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Bước 1: Chuẩn Bị Dữ Liệu
categories = ['Q1', 'Q2', 'Q3', 'Q4']
sales = [45000, 52000, 48000, 61000]
data = pd.DataFrame({'Quarter': categories, 'Sales': sales})

# Bước 2: Tạo Figure
fig, ax = plt.subplots(figsize=(10, 6))

# Bước 3: Visual Encoding
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728']
bars = ax.bar(data['Quarter'], data['Sales'],
              color=colors, edgecolor='black', linewidth=1.5)

# Bước 4: Tối Ưu Hóa Readability
ax.set_xlabel('Quý', fontsize=12, fontweight='bold')
ax.set_ylabel('Doanh Số Bán Hàng (VNĐ)', fontsize=12, fontweight='bold')
ax.set_title('Doanh Số Bán Hàng Theo Quý', fontsize=14, fontweight='bold')
ax.set_ylim(0, max(sales) * 1.1)
```

```
# Thêm giá trị lên đỉnh mỗi cột
for bar in bars:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width()/2., height,
            f'{int(height):,}',
            ha='center', va='bottom', fontsize=10)

ax.grid(axis='y', alpha=0.3, linestyle='--')
plt.tight_layout()
plt.show()
```

4.2 Ví Dụ 2: Line Chart Theo Dõi Xu Hướng

```
import matplotlib.pyplot as plt
import pandas as pd
from datetime import datetime, timedelta

# Bước 1: Tạo Dữ Liệu Chuỗi Thời Gian
dates = pd.date_range(start='2024-01-01', end='2024-12-31', freq='D')
values = np.cumsum(np.random.randn(len(dates))) + 100

data = pd.DataFrame({'Date': dates, 'Value': values})

# Bước 2: Tạo Figure
fig, ax = plt.subplots(figsize=(12, 6))

# Bước 3: Vẽ Line Chart
ax.plot(data['Date'], data['Value'],
        linewidth=2.5, color='#1f77b4', label='Giá trị')

# Thêm một Moving Average
data['MA'] = data['Value'].rolling(window=30).mean()
ax.plot(data['Date'], data['MA'],
        linewidth=2, color='#ff7f0e', linestyle='--', label='Moving Average (30
        ↪  ngày)')

# Bước 4: Tối Ưu Hóa
ax.set_xlabel('Ngày', fontsize=12, fontweight='bold')
ax.set_ylabel('Giá Trị', fontsize=12, fontweight='bold')
ax.set_title('Xu Hướng Giá Trị Theo Thời Gian', fontsize=14, fontweight='bold')
ax.legend(loc='best', fontsize=10)
ax.grid(True, alpha=0.3)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

4.3 Ví Dụ 3: Scatter Plot Tìm Tương Quan

```python
import matplotlib.pyplot as plt
from scipy.stats import pearsonr
import numpy as np

# Bước 1: Tạo Dữ Liệu Có Tương Quan
np.random.seed(42)
x = np.random.randn(100) * 10 + 50
y = x * 1.5 + np.random.randn(100) * 15 + 30

# Bước 2: Tính Hệ Số Tương Quan
correlation, p_value = pearsonr(x, y)

# Bước 3: Tạo Scatter Plot
fig, ax = plt.subplots(figsize=(10, 7))

scatter = ax.scatter(x, y, s=80, alpha=0.6, c=y,
                     cmap='viridis', edgecolors='black', linewidth=0.5)

# Thêm trend line
z = np.polyfit(x, y, 1)
p = np.poly1d(z)
ax.plot(x, p(x), "r--", linewidth=2, label='Trend Line')

# Bước 4: Tối Ưu Hóa
ax.set_xlabel('Biến X', fontsize=12, fontweight='bold')
ax.set_ylabel('Biến Y', fontsize=12, fontweight='bold')
ax.set_title(f'Scatter Plot với Tương Quan Pearson: {correlation:.3f}',
             fontsize=14, fontweight='bold')
ax.legend(fontsize=10)
ax.grid(True, alpha=0.3)

# Thêm colorbar
cbar = plt.colorbar(scatter, ax=ax)
cbar.set_label('Giá Trị Y', fontsize=10)

plt.tight_layout()
plt.show()
```

5 Best Practices Theo Google Design Philosophy

5.1 Tối Giản Hóa (Minimalism)

Loại bỏ tất cả các phần tử không cần thiết. Mỗi pixel phải có mục đích.

```
# Không nên: Quá nhiều decoration
ax.spines['top'].set_visible(True)
ax.spines['right'].set_visible(True)
ax.spines['left'].set_color('red')
ax.spines['bottom'].set_color('blue')

# Nên: Minimal design
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
```

5.2 Accessibility (Khả Năng Truy Cập)

Sử dụng colormap phù hợp với người mù màu.

```
# Sử dụng colormap thân thiện
plt.cm.viridis  # Phù hợp với mù màu đỏ-xanh
plt.cm.cividis  # Tối ưu hóa cho khả năng truy cập

# Tránh
plt.cm.jet  # Không tốt cho mù màu
```

5.3 Consistency (Nhất Quán)

Sử dụng cùng một bảng màu, font, và style trong toàn bộ project.

```
# Định Nghĩa Style Toàn Cầu
COLORS = ['#1f77b4', '#ff7f0e', '#2ca02c']
FONT_SIZE = 12
TITLE_SIZE = 14

plt.rcParams['font.size'] = FONT_SIZE
plt.rcParams['axes.titlesize'] = TITLE_SIZE
```

6 Kết Luận

Việc tạo figures bằng Python không phải chỉ là lập trình kỹ thuật mà còn là nghệ thuật truyền đạt thông tin. Bằng cách tuân theo các nguyên tắc của Google (Truthfulness, Functionality, Beauty) và phương pháp luận khoa học, chúng ta có thể tạo ra những visualizations vừa chính xác vừa hấp dẫn, giúp người xem hiểu rõ dữ liệu một cách hiệu quả.