**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày …. . tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày …. . tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Lời nói đầu, Em xin cảm ơn cô Hà Thị Thúy Vi đã hỗ trợ, hướng dẫn tận tình chúng em trong thời gian em làm thực tập cơ sở ngành , những kiến thức mà cô đã dạy chúng em sẽ là hành trang quý báo trên con đường học vấn và phát triển sự nghiệp tương lai rộng mở của chúng em. Cô đã luôn kiên nhẫn, nhiệt tình trong việc truyền đạt kiến thức và kinh nghiệm quý báo, giúp chúng em vượt qua những khó khăn và thử thách trong quá trình học tập và nghiên cứu.

Những lời khuyên, góp ý của cô không chỉ là kim chỉ nam cho sự phát triển của đồ án môn học này mà còn là nguồn động viên, khích lệ tinh thần lớn lao cho chúng em.

Chúng em xin hứa sẽ tiếp tục nỗ lực không ngừng để không phụ lòng cô đã dành cho chúng em.

Xin chân thành cảm ơn Cô.

Trà Vinh, 29 Tháng 12 năm 2024

**Trần Ngọc Hành**

**MỤC LỤC**

[MỞ ĐẦU 7](#_Toc187299465)

[CHƯƠNG 1: TỔNG QUAN 9](#_Toc187299466)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 10](#_Toc187299467)

[2.1. Cơ sở lý thuyết 10](#_Toc187299468)

[2.1.1. Giới thiệu về trực quan hóa dữ liệu 10](#_Toc187299469)

[2.1.2. Khái quát về Jupyter Book 10](#_Toc187299470)

[2.2. Công cụ và công nghệ sử dụng 17](#_Toc187299471)

[2.2.1. Jupyter Book 17](#_Toc187299472)

[2.2.2. Ngôn ngữ lập trình Python 17](#_Toc187299473)

[2.2.3. Các công cụ hỗ trợ 18](#_Toc187299474)

[2.3. Phần mềm và tài nguyên sử dụng 20](#_Toc187299475)

[2.3.1. Phần mềm 20](#_Toc187299476)

[2.3.2. Tài nguyên học thuật 20](#_Toc187299477)

[2.4. Mô hình tổ chức nội dung sách 21](#_Toc187299478)

[2.4.1. Cấu trúc tài liệu 21](#_Toc187299479)

[2.4.2. Phân tích cách sử dụng Jupyter Book 24](#_Toc187299480)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 25](#_Toc187299481)

[3.1. Chuẩn bị công cụ và môi trường làm việc 25](#_Toc187299482)

[3.1.1. Cài đặt Python 25](#_Toc187299483)

[3.1.2. Cài đặt Jupyter Book: 25](#_Toc187299484)

[3.1.3. Cài đặt các thư viện hỗ trợ (Matplotlib, Seaborn, Plotly): 26](#_Toc187299485)

[3.1.4. Cài đặt Anaconda để quản lý môi trường làm việc: 26](#_Toc187299486)

[3.1.5. Cài đặt Visual Studio Code để soạn thảo nội dung sách: 27](#_Toc187299487)

[3.2. Xây dựng cấu trúc và nội dung sách 27](#_Toc187299488)

[3.3. Tích hợp mã nguồn và trực quan hóa dữ liệu 29](#_Toc187299489)

[3.4. Xây dựng và xuất bản sách 35](#_Toc187299490)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 36](#_Toc187299491)

[4.1. Kết quả đạt được 36](#_Toc187299492)

[4.2. Đánh giá hiệu năng 36](#_Toc187299493)

[4.3. Trải nghiệm người dùng 37](#_Toc187299494)

[4.4. Giao diện chức năng nghiên cứu 37](#_Toc187299495)

[4.5. Những khó khăn và cách khắc phục 37](#_Toc187299496)

[4.6. Một số nội dung của tài liệu đã biên soạn 37](#_Toc187299497)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 38](#_Toc187299498)

[5.1. Kết luận 38](#_Toc187299499)

[5.2. Hướng phát triển 38](#_Toc187299500)

[DANH MỤC TÀI LIỆU THAM KHẢO 39](#_Toc187299501)

[PHỤ LỤC 40](#_Toc187299502)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1 Biểu đồ cột 11](#_Toc156252437)

[Hình 2.2 Biểu đồ đường 12](#_Toc156252437)

[Hình 2.3 Biểu đồ tròn 12](#_Toc156252437)

[Hình 2.4 Histogram 13](#_Toc156252437)

[Hình 2.5 Box Plot 13](#_Toc156252437)

[Hình 2.6 Violin plot 14](#_Toc156252437)

[Hình 2.7 Scatter plot 14](#_Toc156252437)

[Hình 2.8 Bubble chart 15](#_Toc156252437)

[Hình 2.9 Heatmap 15](#_Toc156252437)

[Hình 2.10 Treemap 16](#_Toc156252437)

[Hình 2.11 Network graph 16](#_Toc156252437)

[Hình 2.12 Geographic map 17](#_Toc156252437)

[Hình 2.13 Tiêu đề cấp 6 18](#_Toc156252438)

[Hình 2.14 Danh sách không có thứ tự 18](#_Toc156252438)

[Hình 2.15 Danh sách có thứ tự 18](#_Toc156252438)

[Hình 2.16 Liên kết hình ảnh 19](#_Toc156252438)

[Hình 2.17 Chèn mã inline 19](#_Toc156252438)

[Hình 2.18 Chèn khối mã 19](#_Toc156252438)

[Hình 2.19 Liên kết chéo 19](#_Toc156252438)

[Hình 2.20 Cấu trúc tài liệu 23](#_Toc156252438)

[Hình 3.1 Logo Python 25](#_Toc156252442)

[Hình 3.2 Logo Jupyter Book 26](#_Toc156252442)

[Hình 3.3 Logo Anaconda 26](#_Toc156252443)

[Hình 3.4 Logo Visual Studio Code 27](#_Toc156252444)

[Hình 3.5 Thư mục dự án 27](#_Toc156252445)

[Hình 3.6 \_config.yml cấu hình tài liệu 28](#_Toc156252446)

[Hình 3.7 \_toc.yml Cấu trúc nội dung 28](#_Toc156252447)

[Hình 3.8 Biểu đồ đường Matplotlib 29](#_Toc156252448)

[Hình 3.9 Tạo bố cục nhiều biểu đồ 30](#_Toc156252449)

[Hình 3.10 Biểu đồ cột Seaborn 31](#_Toc156252449)

[Hình 3.11 Biểu đồ ghép 32](#_Toc156252449)

[Hình 3.12 Biểu đồ tương tác với plotly 33](#_Toc156252449)

[Hình 3.13 Biểu đồ bản đồ 34](#_Toc156252449)

[Hình 3.14 Xuất sang định sang PDF 35](#_Toc156252449)

**TÓM TẮT ĐỒ ÁN**

**Vấn đề nghiên cứu:**

Hiện nay, việc sử dụng các công cụ trực quan hóa dữ liệu đã trở thành một yêu cầu quan trọng trong lĩnh vực phân tích và khoa học dữ liệu. Python, với các thư viện mạnh mẽ như Matplotlib, Seaborn, và Plotly, là một trong những ngôn ngữ phổ biến nhất để xây dựng các biểu đồ và minh họa dữ liệu.

Tuy nhiên, tài liệu hướng dẫn chi tiết, có cấu trúc rõ ràng để hỗ trợ người học từ cơ bản đến nâng cao chưa thực sự phong phú. Do đó, đề tài tập trung vào việc:

Nghiên cứu Jupyter Book - một công cụ mạnh mẽ để biên soạn, tổ chức và xuất bản tài liệu trực tuyến.

Xây dựng tài liệu hướng dẫn lập trình trực quan hóa dữ liệu bằng Python, tích hợp lý thuyết và ví dụ thực tế.

**Các hướng tiếp cận:**

Nghiên cứu công cụ Jupyter Book: Tìm hiểu cấu trúc, tính năng, cách biên tập tài liệu, và các định dạng xuất bản như HTML và PDF.

Phân tích các thư viện trực quan hóa dữ liệu: Khám phá Matplotlib, Seaborn, và Plotly, tập trung vào khả năng tạo biểu đồ, tùy chỉnh, và tương tác.

Xây dựng nội dung tài liệu: Chia tài liệu thành các chương rõ ràng, từ cài đặt môi trường, giới thiệu dữ liệu, đến các kỹ thuật trực quan hóa nâng cao.

**Cách giải quyết vấn đề:**

Cấu trúc nội dung và tài liệu được biên soạn thành các chương cụ thể, bao gồm:

* Giới thiệu về lập trình trực quan hóa dữ liệu.
* Cài đặt môi trường và công cụ Python.
* Trình bày các thư viện Matplotlib, Seaborn, và Plotly.
* Ví dụ thực tiễn và các kỹ thuật nâng cao.
* Tích hợp trực quan hóa vào bài tập thực tế.

Sử dụng Jupyter Book:

Xây dựng tài liệu sử dụng định dạng Markdown và Jupyter Notebook.

Tích hợp hình ảnh, mã nguồn Python, và kết quả trực quan.

Xuất bản tài liệu dưới dạng website và file PDF để phục vụ học tập và tham khảo.

**Một số kết quả đạt được:**

Tài liệu hướng dẫn hoàn chỉnh: Tài liệu gồm 10 chương với các nội dung từ cơ bản đến nâng cao, cung cấp các ví dụ minh họa chi tiết về trực quan hóa dữ liệu.

Ứng dụng Jupyter Book: Đã biên soạn và xuất bản thành công tài liệu dưới dạng website trực tuyến và PDF, hỗ trợ người học dễ dàng truy cập.

Hiệu quả thực tiễn: Người học nắm vững cách sử dụng các thư viện Matplotlib, Seaborn, và Plotly để trực quan hóa dữ liệu và Tăng khả năng áp dụng vào các dự án thực tế trong phân tích dữ liệu và khoa học dữ liệu

# MỞ ĐẦU

**Lí do chọn đề tài:**

Tầm quan trọng của trực quan hóa dữ liệu trong thời đại số: Trong kỷ nguyên dữ liệu, khả năng trực quan hóa thông tin từ dữ liệu một cách hiệu quả đóng vai trò quan trọng trong việc hỗ trợ ra quyết định. Trực quan hóa không chỉ giúp truyền tải thông tin phức tạp thành hình ảnh dễ hiểu mà còn giúp người dùng khám phá, phân tích và tương tác với dữ liệu.

Sự phổ biến và sức mạnh của Python trong phân tích dữ liệu: Python là ngôn ngữ lập trình phổ biến nhờ cú pháp đơn giản và hệ sinh thái thư viện mạnh mẽ, đặc biệt trong lĩnh vực phân tích và trực quan hóa dữ liệu. Các thư viện như Matplotlib, Seaborn, và Plotly cung cấp khả năng tạo biểu đồ từ cơ bản đến nâng cao, đáp ứng nhu cầu từ người mới học đến chuyên gia.

Thiếu tài liệu hướng dẫn có cấu trúc: Mặc dù có nhiều tài liệu hướng dẫn trực tuyến, chúng thường rời rạc, thiếu tính hệ thống và không phù hợp với những người mới học hoặc cần tài liệu để giảng dạy. Việc biên soạn một tài liệu chi tiết, kết hợp lý thuyết và thực hành, đồng thời có tính tương tác là cần thiết để giúp người học tiếp cận hiệu quả hơn với lập trình trực quan hóa dữ liệu.

Ưu điểm của Jupyter Book trong giáo dục và nghiên cứu: Jupyter Book là một công cụ mạnh mẽ để xây dựng tài liệu giảng dạy và sách điện tử với khả năng tích hợp lý thuyết, mã nguồn và kết quả thực thi. Đây là nền tảng lý tưởng để phát triển tài liệu hướng dẫn lập trình, đặc biệt là trong lĩnh vực trực quan hóa dữ liệu.

Ứng dụng thực tế và tiềm năng phát triển: Đề tài không chỉ phù hợp với việc học và giảng dạy mà còn có tiềm năng ứng dụng vào nhiều lĩnh vực như khoa học dữ liệu, kinh tế, môi trường và xã hội. Việc sử dụng Jupyter Book để xuất bản tài liệu giúp tài liệu dễ dàng tiếp cận hơn, từ đó hỗ trợ cộng đồng người học và nhà nghiên cứu.

**Mục đích chọn đề tài:**

Biên soạn tài liệu hướng dẫn lập trình trực quan hóa dữ liệu trên Python một cách hệ thống và dễ tiếp cận.

Nghiên cứu và ứng dụng Jupyter Book để tạo sách điện tử tương tác, tích hợp lý thuyết và thực hành.

Hỗ trợ người học và giảng viên trong việc học tập và giảng dạy lập trình trực quan hóa dữ liệu.

Phát triển tài liệu giáo dục chất lượng cao, ứng dụng trong phân tích dữ liệu thực tiễn.

**Đối tượng nghiên cứu:**

Jupyter Book: Cách cài đặt, cấu hình và sử dụng Jupyter Book để biên soạn tài liệu học tập, Tích hợp nội dung lý thuyết, mã nguồn Python, và biểu đồ trực quan vào sách điện tử tương tác.

Các thư viện trực quan hóa dữ liệu trên Python:

Matplotlib: Thư viện cơ bản để tạo biểu đồ và đồ họa dữ liệu.

Seaborn: Công cụ mở rộng trực quan hóa với các biểu đồ nâng cao.

Plotly: Thư viện hỗ trợ biểu đồ tương tác mạnh mẽ và linh hoạt.

Người học lập trình và phân tích dữ liệu: Sinh viên, nhà nghiên cứu, và người làm việc trong lĩnh vực khoa học dữ liệu cần tài liệu hướng dẫn rõ ràng, dễ hiểu, người mới học lập trình Python hoặc muốn tìm hiểu trực quan hóa dữ liệu để ứng dụng vào công việc.

**Phạm vi nghiên cứu:**

Tập trung vào các thư viện trực quan hóa dữ liệu Python: Matplotlib, Seaborn, và Plotly.

Sử dụng Jupyter Book để biên soạn tài liệu học tập tương tác.

Ứng dụng cho việc học tập, giảng dạy và phân tích dữ liệu cơ bản đến trung cấp.

# TỔNG QUAN

Trong thời đại dữ liệu, trực quan hóa thông tin là công cụ quan trọng giúp phân tích và trình bày dữ liệu một cách hiệu quả. Python với các thư viện mạnh mẽ như Matplotlib, Seaborn, và Plotly đã trở thành ngôn ngữ phổ biến trong lập trình trực quan hóa dữ liệu. Tuy nhiên, việc thiếu tài liệu hướng dẫn có hệ thống và dễ tiếp cận đã gây khó khăn cho người học và giảng viên.

Jupyter Book là công cụ hiện đại cho phép xây dựng tài liệu học tập dưới dạng sách điện tử tương tác. Với khả năng tích hợp lý thuyết, mã nguồn, và kết quả thực thi, Jupyter Book hỗ trợ hiệu quả cho việc biên soạn tài liệu lập trình, đặc biệt là trong lĩnh vực trực quan hóa dữ liệu.

Đề tài này tập trung vào việc tìm hiểu Jupyter Book và các thư viện trực quan hóa dữ liệu trên Python, sau đó biên soạn một tài liệu hướng dẫn có cấu trúc rõ ràng, tích hợp lý thuyết và thực hành. Nội dung tài liệu bao gồm từ kiến thức cơ bản đến ứng dụng thực tế, phục vụ cho sinh viên, giảng viên, và người làm việc trong lĩnh vực phân tích dữ liệu.

Đối tượng và ứng dụng:

Đối tượng: Người học lập trình, sinh viên, giảng viên, và người làm việc với dữ liệu.

Ứng dụng: Hỗ trợ giảng dạy và học tập trực quan hóa dữ liệu, áp dụng trong phân tích dữ liệu các lĩnh vực như kinh doanh, khoa học, và công nghệ.

Ý nghĩa thực tiễn: Đề tài không chỉ giúp người học tiếp cận dễ dàng hơn với lập trình trực quan hóa dữ liệu mà còn đóng góp tài liệu giáo dục chất lượng cao, phù hợp với xu hướng sử dụng tài liệu điện tử và học tập tương tác trong thời đại số.

# NGHIÊN CỨU LÝ THUYẾT

## Cơ sở lý thuyết

### Giới thiệu về trực quan hóa dữ liệu

Trực quan hóa dữ liệu đóng vai trò quan trọng trong phân tích và truyền tải thông tin. Cụ thể:

Hỗ trợ phân tích dữ liệu: Giúp làm rõ các mối quan hệ, xu hướng và mẫu dữ liệu trong các tập dữ liệu lớn. Ví dụ: Biểu đồ đường giúp phát hiện xu hướng theo thời gian, biểu đồ phân tán thể hiện mối tương quan giữa hai biến.

Truyền tải thông tin hiệu quả: Cung cấp hình ảnh trực quan dễ hiểu, hỗ trợ việc trình bày dữ liệu phức tạp trong báo cáo hoặc bài thuyết trình.

Hỗ trợ ra quyết định: Biểu đồ và đồ thị giúp các nhà quản lý hoặc nhà khoa học dữ liệu đưa ra quyết định nhanh chóng và chính xác hơn.

Các nguyên tắc cơ bản:

Màu sắc: Chọn màu sắc dễ phân biệt, hài hòa và phù hợp với đối tượng mục tiêu. Tránh sử dụng quá nhiều màu gây rối.

Bố cục: Đảm bảo sự rõ ràng, trực quan, tập trung vào thông tin chính. Tiêu đề, trục và chú thích cần sắp xếp hợp lý để dễ đọc.

Loại biểu đồ: Chọn loại biểu đồ phù hợp với mục đích sử dụng:

Biểu đồ thanh: So sánh các nhóm dữ liệu.

Biểu đồ tròn: Minh họa tỷ lệ phần trăm.

Biểu đồ phân tán: Tìm mối tương quan giữa hai biến.

### Khái quát về Jupyter Book

**Tổng quan về Jupyter Book**

Jupyter Book là công cụ mã nguồn mở giúp tạo tài liệu đa phương tiện kết hợp giữa:

Nội dung văn bản: Viết bằng Markdown để trình bày lý thuyết và khái niệm.

Mã lệnh: Sử dụng Jupyter Notebook để trình bày và thực thi mã Python.

Hình ảnh và biểu đồ: Minh họa dữ liệu trực quan.

Jupyter Book đặc biệt hữu ích trong biên soạn tài liệu học thuật, sách giáo khoa và hướng dẫn lập trình.

**Lợi ích của Jupyter Book**

Kết hợp lý thuyết và thực hành: Tích hợp lý thuyết với mã lệnh thực thi trực tiếp, giúp người học hiểu và thực hành dễ dàng.

Khả năng tương tác cao: Cho phép người đọc chạy và chỉnh sửa mã trực tiếp trên tài liệu.

Tổ chức nội dung rõ ràng: Sử dụng Markdown và Notebook để xây dựng cấu trúc chương mục cụ thể.

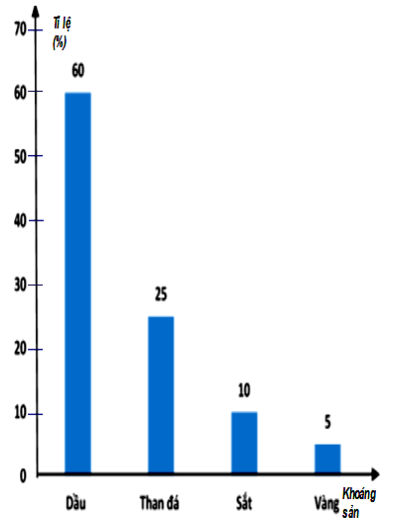
Xuất bản đa dạng: Hỗ trợ xuất bản dưới dạng website, PDF, hoặc ebook.

Tích hợp với hệ sinh thái Python: Hỗ trợ các thư viện như Matplotlib, Seaborn, Plotly để minh họa dữ liệu trực quan.

Trực quan hóa dữ liệu là một công cụ mạnh mẽ giúp chúng ta chuyển đổi dữ liệu thô thành những hình ảnh dễ hiểu, từ đó rút ra những insights sâu sắc. Có rất nhiều kỹ thuật trực quan hóa khác nhau, mỗi kỹ thuật phù hợp với một loại dữ liệu và câu hỏi nghiên cứu khác nhau.

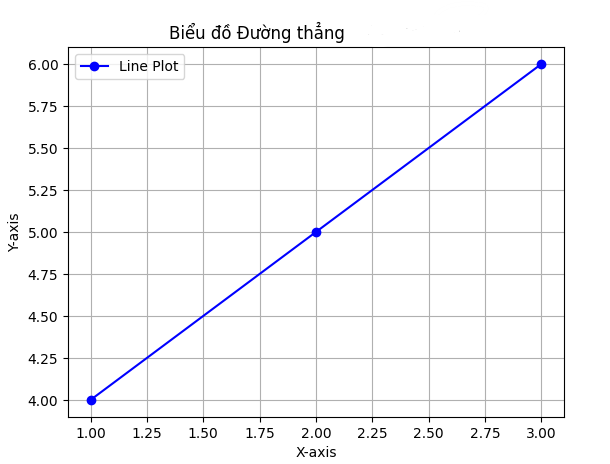
**Biểu đồ cơ bản:**

**Biểu đồ cột**: Sử dụng các cột để biểu diễn dữ liệu theo danh mục. Thường dùng để so sánh các giá trị giữa các danh mục.

****

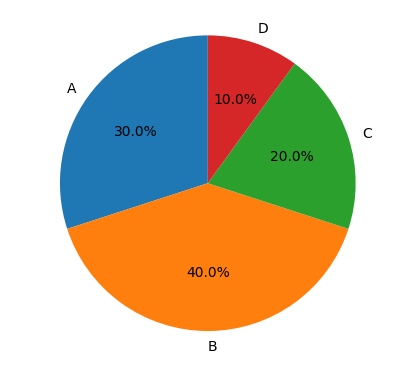
Hình 2.1: Biểu đồ cột

**Biểu đồ đường:** Sử dụng đường để thể hiện sự thay đổi của dữ liệu theo thời gian hoặc một biến số liên tục khác. Thường dùng để thể hiện xu hướng.



Hình 2.2: Biểu đồ đường

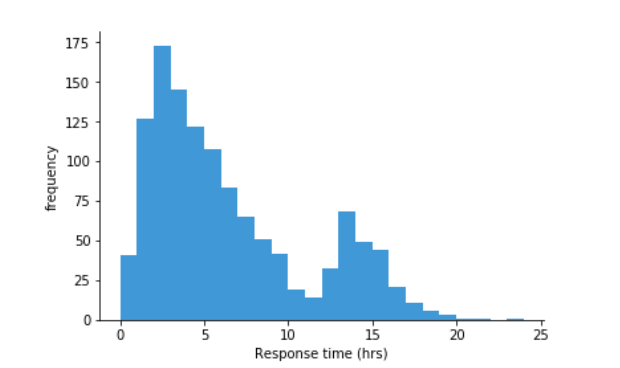
**Biểu đồ tròn:** Sử dụng các phần của một hình tròn để biểu diễn tỷ lệ phần trăm của các danh mục trong một tổng thể.



Hình 2.3: Biểu đồ tròn

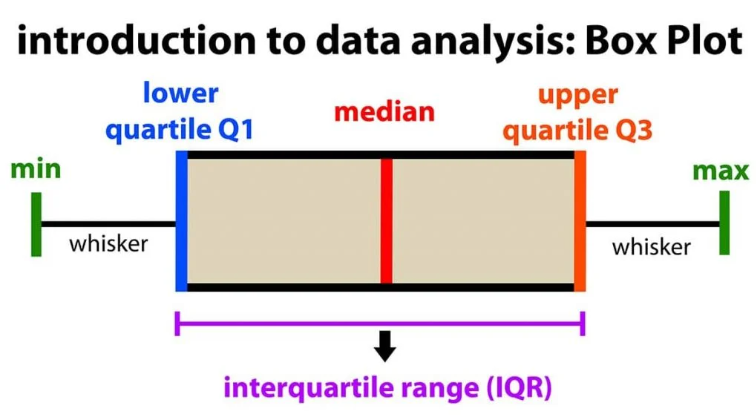
**Biểu đồ phân bố:**

**Histogram**: Sử dụng các hình chữ nhật để biểu diễn tần suất của các giá trị trong một khoảng dữ liệu. Thường dùng để xem xét phân phối của dữ liệu.



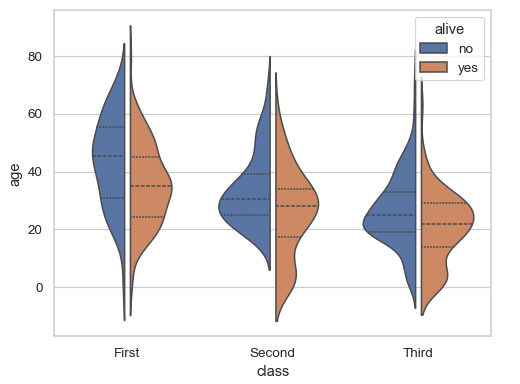
Hình 2.4: Histogram

**Box plot:** Sử dụng một hộp và các đường để biểu diễn phân bố của dữ liệu, bao gồm các giá trị ngoại lệ.



Hình 2.5: Box Plot

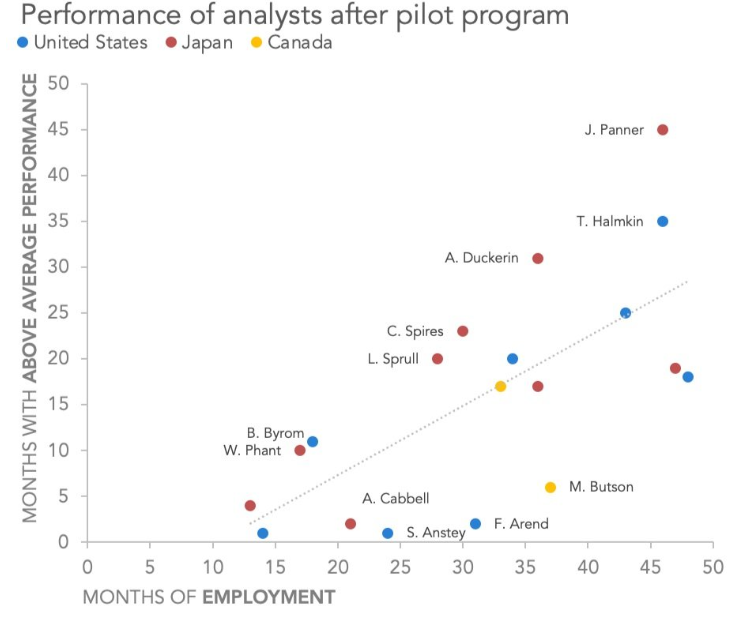
**Violin plot:** Tương tự như box plot nhưng sử dụng mật độ để thể hiện hình dạng phân bố.



Hình 2.6: Violin plot

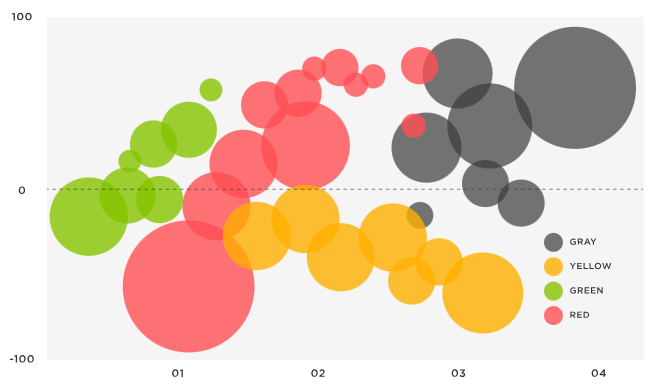
**Biểu đồ mối quan hệ:**

**Scatter plot:** Sử dụng các điểm để biểu diễn mối quan hệ giữa hai biến số liên tục

****

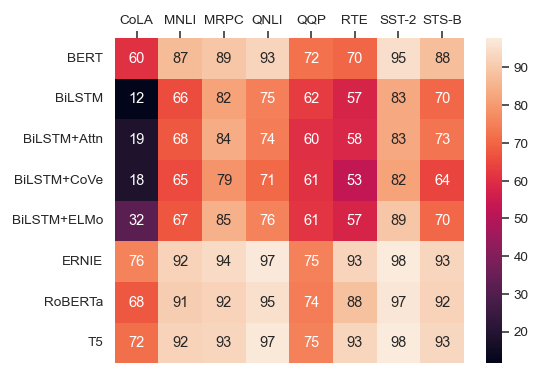
Hình 2.7: Scatter plot

**Bubble chart:** Giống như scatter plot nhưng kích thước của các điểm biểu diễn một biến thứ ba.



Hình 2.8: Bubble chart

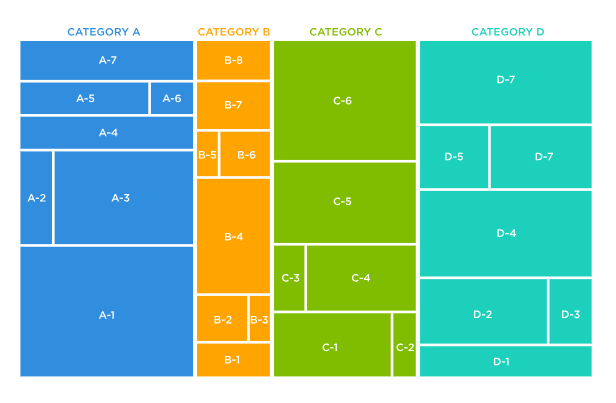
**Heatmap:** Sử dụng màu sắc để biểu diễn giá trị của các ô trong một ma trận. Thường dùng để thể hiện mối quan hệ giữa hai biến số phân loại.



Hình 2.9: Heatmap

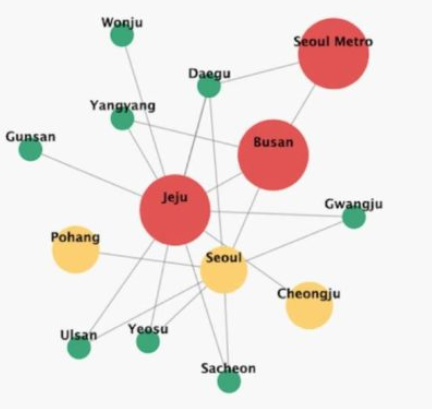
**Các kỹ thuật khác:**

**Treemap:** Sử dụng các hình chữ nhật có kích thước khác nhau để biểu diễn dữ liệu phân cấp.



Hình 2.10: Treemap

**Network graph:** Sử dụng các nút và cạnh để biểu diễn mối quan hệ giữa các đối tượng.



Hình 2.11: Network graph

**Geographic map:** Sử dụng bản đồ để hiển thị dữ liệu địa lý.



Hình 2.12: Geographic map

## Công cụ và công nghệ sử dụng

### Jupyter Book

Phiên bản sử dụng và lý do lựa chọn: Phiên bản mới nhất của Jupyter Book được chọn nhằm đảm bảo tích hợp tốt nhất với các công cụ và thư viện Python, đồng thời hỗ trợ xuất bản tài liệu dưới dạng website, PDF, hoặc ebook.

Cách cấu hình: Jupyter Book được cấu hình để hỗ trợ Markdown và Jupyter Notebook, giúp kết hợp nội dung văn bản, mã lệnh và biểu đồ trực quan trong một tài liệu duy nhất. Các bước cấu hình bao gồm cài đặt thư viện cần thiết, thiết lập tệp cấu hình \_config.yml, và tổ chức cấu trúc thư mục.

### Ngôn ngữ lập trình Python

**Các thư viện chính:**

Matplotlib: Tạo biểu đồ cơ bản như biểu đồ cột, đường, hoặc hình tròn.

Seaborn: Hỗ trợ tạo các biểu đồ nâng cao, dễ dàng tùy chỉnh với giao diện trực quan.

Plotly: Cung cấp khả năng tạo biểu đồ tương tác như biểu đồ 3D, bản đồ nhiệt, hoặc biểu đồ phân tán động.

Tích hợp Python với Jupyter Notebook:Python được sử dụng kết hợp với Jupyter Notebook để trình bày lý thuyết cùng mã lệnh thực hành, cho phép người học chạy mã trực tiếp và xem kết quả tức thì.

### Các công cụ hỗ trợ

Markdown: Ngôn ngữ đơn giản để viết tài liệu lý thuyết với cấu trúc rõ ràng và dễ đọc.

**Cú pháp cơ bản của Markdown:**

Tiêu đề: Sử dụng # để tạo tiêu đề các cấp từ 1 đến 6

A screenshot of a computer

Description automatically generated

Hình 2.13: Tiêu đề 6 cấp

**Định dạng văn bản**:

*In nghiêng*: \*in nghiêng\* hoặc \_in nghiêng\_

**In đậm**: \*\*in đậm\*\*

***In nghiêng và đậm***: \*\*\*in nghiêng và đậm\*\*\*

~~gạch ngang~~

**Danh sách**:

Danh sách không thứ tự:

A close-up of a white background

Description automatically generated

Hình 2.14: Danh sách không có thứ tự

Danh sách có thứ tự:

A close-up of a number

Description automatically generated

Hình 2.15: Danh sách có thứ tự

**Liên kết và hình ảnh:**

A close-up of a code

Description automatically generated

Hình 2.16: Liên kết hình ảnh

**Code Block:**

Bạn có thể sử dụng khối mã với Markdown:

Chèn mã inline:



Hình 2.17: Chèn mã inline

Chèn khối mã:

A close up of a logo

Description automatically generated

Hình 2.18: chèn khối mã

**Liên kết chéo (Cross-referencing) :**

Tạo liên kết đến các phần hoặc tài liệu khác trong Jupyter Book:

**A computer code with many colorful text

Description automatically generated with medium confidence**

Hình 2.19: Liên kết chéo

Jupyter Notebook: Công cụ chính để thực thi mã lệnh Python, minh họa kết quả và tích hợp nội dung trực quan.

GitHub/Git: Hỗ trợ quản lý phiên bản, theo dõi lịch sử chỉnh sửa và xuất bản tài liệu trực tuyến.

Visual Studio Code/JupyterLab: Các môi trường soạn thảo và phát triển hiệu quả, tích hợp mạnh mẽ với Jupyter Notebook và các công cụ khác.

Anaconda: là một công cụ phổ biến giúp quản lý các môi trường Python và thư viện, cung cấp các công cụ như Jupyter Notebook. Việc sử dụng Anaconda giúp đơn giản hóa quá trình cài đặt và quản lý các thư viện cần thiết cho dự án, bao gồm Matplotlib, Seaborn, Plotly và các thư viện khác.

## Phần mềm và tài nguyên sử dụng

### Phần mềm

Python (phiên bản cụ thể): Phiên bản Python được sử dụng trong dự án này là Python 3.x, với các thư viện chính như Matplotlib, Seaborn, Plotly, và Pandas. Việc sử dụng Python 3.x đảm bảo tính tương thích với các thư viện và công cụ mới nhất, đồng thời đáp ứng yêu cầu về hiệu suất và tính năng của các công cụ phân tích và trực quan hóa dữ liệu.

Jupyter Book và các plugin mở rộng: Jupyter Book là công cụ chính để biên soạn tài liệu, cho phép kết hợp nội dung văn bản, mã lệnh Python, và biểu đồ trực quan. Các plugin mở rộng của Jupyter Book có thể được sử dụng để hỗ trợ tính năng xuất bản, tương tác hoặc định dạng tài liệu.

jupyter-book: Plugin chính để tạo và xuất bản tài liệu từ Markdown và Jupyter Notebook.

sphinx-book-theme: Plugin giúp tạo giao diện đẹp và dễ dàng điều hướng cho sách.

nbsphinx: Plugin hỗ trợ tích hợp Jupyter Notebook vào tài liệu Sphinx.

### Tài nguyên học thuật

**Các sách và tài liệu tham khảo:**

"Python Data Science Handbook" của Jake VanderPlas: Cung cấp nền tảng vững chắc về các thư viện phân tích và trực quan hóa dữ liệu bằng Python, rất hữu ích cho việc hiểu sâu về cách sử dụng Matplotlib, Seaborn, và Plotly.

"Data Visualization with Python and Matplotlib" của Kyran Dale: Tài liệu chi tiết về cách sử dụng Matplotlib trong việc trực quan hóa dữ liệu.

"Interactive Data Visualization with Python" của Abhishek Kumar: Hướng dẫn chi tiết về cách tạo biểu đồ tương tác với Plotly.

**Nguồn tài nguyên mã nguồn mở từ cộng đồng:**

GitHub Repositories: Các mã nguồn mở trên GitHub, như kho lưu trữ ví dụ về trực quan hóa dữ liệu, mẫu Jupyter Book hoặc Jupyter Notebooks, giúp người học và người phát triển có thể tham khảo, học hỏi và đóng góp.

Kaggle Datasets: Kaggle cung cấp nhiều bộ dữ liệu mở, có thể sử dụng trong các ví dụ minh họa trực quan hóa dữ liệu. Đây là nguồn tài nguyên phong phú để thực hành và thử nghiệm.

Plotly, Seaborn, và Matplotlib Documentation: Tài liệu chính thức từ các thư viện này sẽ cung cấp các hướng dẫn chi tiết về cách sử dụng các thư viện trong việc tạo ra các biểu đồ và đồ thị trực quan.

Jupyter Community: Các diễn đàn và nhóm trực tuyến nơi người dùng có thể thảo luận, trao đổi về các vấn đề và giải pháp khi làm việc với Jupyter Book và các công cụ liên quan.

## Mô hình tổ chức nội dung sách

### Cấu trúc tài liệu

Cuốn sách được tổ chức thành nhiều chương, mỗi chương có mục tiêu rõ ràng và nội dung chi tiết, với mục đích giúp người đọc hiểu và thực hành trực quan hóa dữ liệu bằng Python thông qua các công cụ như Jupyter Book. Dưới đây là cấu trúc của cuốn sách:

**Chapter 1 (Giới thiệu)**

Mục tiêu: Cung cấp cái nhìn tổng quan về trực quan hóa dữ liệu, giải thích tầm quan trọng của việc sử dụng biểu đồ và đồ thị trong việc phân tích và truyền tải thông tin.

Nội dung: Khái niệm cơ bản về trực quan hóa dữ liệu và vai trò của trực quan hóa trong việc hỗ trợ ra quyết định, nhận diện xu hướng và làm rõ các mối quan hệ trong dữ liệu giới thiệu về các loại biểu đồ cơ bản và nâng cao, cùng với những nguyên tắc cơ bản khi thiết kế trực quan.

**Chapter 2 (Hướng dẫn cài đặt)**

Mục tiêu: Hướng dẫn cách cài đặt môi trường và công cụ cần thiết để làm việc với Python và Jupyter Book.

Nội dung: Cài đặt Python, Jupyter Notebook, và các thư viện cần thiết như Matplotlib, Seaborn, Plotly và hướng dẫn cài đặt Jupyter Book và cách cấu hình môi trường làm việc và giới thiệu các công cụ hỗ trợ như Visual Studio Code và GitHub để quản lý phiên bản tài liệu.

**Chapter 3 (Giới thiệu về Dữ liệu)**

Mục tiêu: Giới thiệu về các loại dữ liệu phổ biến trong phân tích dữ liệu và cách chuẩn bị dữ liệu để thực hiện trực quan hóa.

Nội dung: Các loại dữ liệu: Dữ liệu số, dữ liệu phân loại, dữ liệu chuỗi thời gian, v.v…. Cách làm sạch và xử lý dữ liệu trước khi đưa vào trực quan hóa và giới thiệu về các bộ dữ liệu mẫu và cách sử dụng chúng trong các bài thực hành.

**Chapter 4 (Thống kê mô tả)**

Mục tiêu: Hướng dẫn các khái niệm cơ bản trong thống kê mô tả và cách sử dụng chúng để phân tích dữ liệu.

Nội dung: Các phép đo cơ bản: Trung bình, trung vị, phương sai, độ lệch chuẩn và có nhiều cách tính toán và diễn giải các chỉ số thống kê mô tả sử dụng trực quan hóa để minh họa các khái niệm thống kê mô tả.

**Chapter 5 (Trực quan hóa dữ liệu với Matplotlib)**

Mục tiêu: Giới thiệu về thư viện Matplotlib và cách tạo các biểu đồ cơ bản trong Python.

Nội dung: Cách sử dụng Matplotlib để tạo biểu đồ thanh, đường, hình tròn, biểu đồ phân tán và cách tùy chỉnh biểu đồ: Đổi màu sắc, kiểu đường, tiêu đề, trục, v.v….. Giới thiệu các tính năng nâng cao của Matplotlib như biểu đồ con (subplot) và đồ thị 3D.

**Chapter 6 (Trực quan hóa dữ liệu với Seaborn)**

Mục tiêu: Hướng dẫn sử dụng thư viện Seaborn để tạo các biểu đồ nâng cao và đẹp mắt.

Nội dung: Tạo biểu đồ phân phối, biểu đồ hộp (box plot), biểu đồ nhiệt (heatmap) và Cách sử dụng các tính năng của Seaborn để tùy chỉnh màu sắc và kiểu dáng biểu đồ và có các kỹ thuật trực quan hóa nâng cao như vẽ các mối quan hệ giữa các biến (pair plots, scatter plots).

**Chapter 7 (Trực quan hóa dữ liệu với Plotly)**

Mục tiêu: Giới thiệu về thư viện Plotly và cách tạo các biểu đồ tương tác.

Nội dung: Cách tạo biểu đồ tương tác như biểu đồ phân tán động, biểu đồ 3D, và bản đồ nhiệt (heatmap) và tích hợp các tính năng tương tác như zoom, hover thông tin, và thay đổi kiểu dáng và giới thiệu về Plotly Dash để xây dựng ứng dụng trực quan hóa dữ liệu tương tác.

**Chapter 8 (Kỹ thuật trực quan hóa dữ liệu nâng cao)**

Mục Tiêu: Cung cấp các kỹ thuật trực quan hóa nâng cao để giải quyết các bài toán phức tạp trong dữ liệu.

Nội dung: Các biểu đồ động, biểu đồ phân loại, và biểu đồ mạng và sử dụng các thư viện nâng cao như Plotly, Bokeh để tạo các biểu đồ tùy chỉnh áp dụng các kỹ thuật trực quan hóa vào các bài toán thực tế như phân tích dữ liệu lớn và mô hình học máy.

**Chapter 9 (Trực quan hóa trong các dự án)**

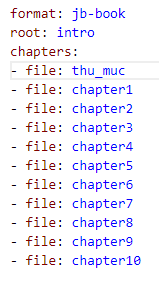
Mục tiêu: Hướng dẫn cách áp dụng các kỹ thuật trực quan hóa dữ liệu vào các dự án thực tế.

Nội dung: Cách sử dụng trực quan hóa để hỗ trợ quyết định trong các dự án thực tế (ví dụ: phân tích kinh doanh, nghiên cứu thị trường) và Giới thiệu cách tích hợp trực quan hóa dữ liệu vào các báo cáo, thuyết trình, hoặc ứng dụng web và có các ví dụ thực tế về sử dụng trực quan hóa trong các dự án.

**Chapter 10 (Tài liệu tham khảo)**

Mục tiêu: Cung cấp các nguồn tài liệu tham khảo hữu ích cho người học và người nghiên cứu.

Nội dung: Các sách, bài báo, và nghiên cứu liên quan đến trực quan hóa dữ liệu và Python và các tài nguyên học trực tuyến, khóa học, và diễn đàn giúp người học tiếp tục mở rộng kiến thức và các tài nguyên mã nguồn mở từ cộng đồng giúp người học nâng cao kỹ năng lập trình và trực quan hóa dữ liệu.



Hình 2.20: Cấu trúc tài liệu

### Phân tích cách sử dụng Jupyter Book

Tổ Chức File: Markdown (.md) và Notebook (.ipynb): Cuốn sách sử dụng Markdown (.md) cho phần lý thuyết và Jupyter Notebook (.ipynb) cho các ví dụ mã lệnh thực hành và trực quan hóa dữ liệu. Mỗi chương có thể bao gồm cả hai loại file này

Markdown (.md): Dùng để viết nội dung lý thuyết, giải thích khái niệm, hướng dẫn từng bước, và cung cấp tài liệu tham khảo.

Jupyter Notebook (.ipynb): Dùng để triển khai mã lệnh Python, thực hiện các phép toán, và tạo biểu đồ trực quan. Người đọc có thể chạy mã trực tiếp trong Jupyter Notebook và nhìn thấy kết quả tức thì.

Tích Hợp Mã Nguồn, Đồ Thị, và Hình Ảnh Vào Tài Liệu:

Mã nguồn: Các đoạn mã Python sẽ được tích hợp trực tiếp vào trong các notebook, giúp người học không chỉ đọc lý thuyết mà còn thực hành ngay lập tức. Những ví dụ mã nguồn này giúp giải thích các khái niệm lý thuyết và thực hành tạo các biểu đồ.

Đồ thị và hình ảnh: Hình ảnh trực quan, bao gồm các biểu đồ, đồ thị, và ảnh minh họa, sẽ được nhúng vào tài liệu để làm rõ các khái niệm và kết quả của mã lệnh.

# HIỆN THỰC HÓA NGHIÊN CỨU

## Chuẩn bị công cụ và môi trường làm việc

### Cài đặt Python

**Cài đặt Python từ** <https://www.python.org>

Chạy tệp tải về và làm theo các bước hướng dẫn để cài đặt Python trên máy tính của bạn. Khi cài đặt, nhớ chọn tùy chọn **Add Python to PATH** để dễ dàng sử dụng Python từ dòng lệnh.

Kiểm tra cài đặt Python Mở terminal:

**python –version**



Hình 3.1: Logo Python

Chú Thích: Python là ngôn ngữ lập trình chính được sử dụng trong dự án. Việc cài đặt Python đảm bảo môi trường cần thiết để chạy mã lệnh và thực hiện các bài tập lập trình.

### Cài đặt Jupyter Book:

Sau khi cài đặt Python, mở terminal trên Visual studio code và cài đặt Jupyter Book bằng pip:



Kiểm tra cài đặt Jupyter Book bằng cách chạy lệnh sau:



****

Hình 3.2: Logo Jupyter Book

Chú thích: Jupyter Book là công cụ dùng để biên soạn và xuất bản sách, hỗ trợ tích hợp nội dung lý thuyết, mã nguồn, và trực quan hóa dữ liệu trong cùng một tài liệu.

### Cài đặt các thư viện hỗ trợ (Matplotlib, Seaborn, Plotly):







Chú thích: Các thư viện này cung cấp công cụ mạnh mẽ để trực quan hóa dữ liệu, tạo biểu đồ và đồ thị từ cơ bản đến nâng cao.

### Cài đặt Anaconda để quản lý môi trường làm việc:

Tải và cài đặt Anaconda từ <https://www.anaconda.com>

Tạo môi trường mới với Python 3.10:

**conda create --name book\_env python=3.10**

**conda activate book\_env**



Hình 3.3: Logo Anaconda

Chú thích: Anaconda giúp quản lý môi trường Python, cài đặt thư viện, và hỗ trợ làm việc với Jupyter Notebook một cách dễ dàng, hiệu quả.

### Cài đặt Visual Studio Code để soạn thảo nội dung sách:

Tải Visual Studio Code từ <https://code.visualstudio.com>

Cài đặt các tiện ích mở rộng như Python và Markdown All-in-One.

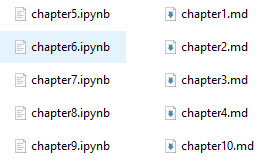


Hình 3.4: Logo Visual Studio Code

Chú thích: Visual Studio Code là môi trường phát triển tích hợp (IDE) hỗ trợ việc soạn thảo nội dung, quản lý mã nguồn, và chỉnh sửa tệp Markdown hoặc Notebook thuận tiện.

## Xây dựng cấu trúc và nội dung sách

Tạo thư mục dự án Jupyter Book



Hình 3.5: Thư mục dự án

Thư mục dự án tổ chức nội dung Jupyter Book một cách rõ ràng, hỗ trợ biên dịch sách, dễ dàng bảo trì, mở rộng, và tạo điều kiện thuận lợi cho cộng tác hiệu quả.

Tạo tệp \_config.yml để cấu hình sách:

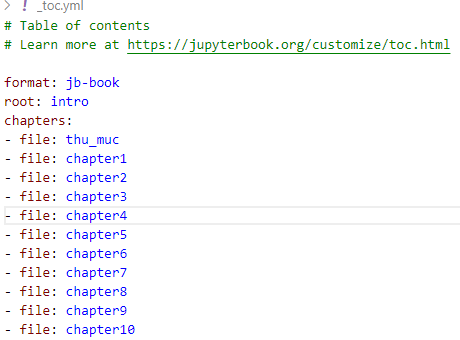
A screenshot of a computer program

Description automatically generated

Hình 3.6: \_config.yml cấu hình sách

Tệp \_config.yml cấu hình các thiết lập chính cho Jupyter Book, bao gồm tiêu đề, tác giả, mục lục, giao diện, tính năng mở rộng, và định dạng xuất bản, giúp kiểm soát cách sách hiển thị và hoạt động.

Tạo tệp \_toc.yml để định nghĩa cấu trúc nội dung.



Hình 3.7: \_toc.yml Cấu trúc nội dung

Tệp \_toc.yml xác định cấu trúc nội dung Jupyter Book, bao gồm thứ tự các chương, mục, và các tệp liên kết để tổ chức sách một cách hợp lý.

## Tích hợp mã nguồn và trực quan hóa dữ liệu

**Tạo biểu đồ với Matplotlib, Seaborn, và Plotly:**

Matplotlib: Biểu đồ đường thẳng, tĩnh, cơ bản.



A graph with a line graph

Description automatically generated

Hình 3.8: Biểu đồ đường Matplotlib

Chú thích: Biểu đồ đường Matplotlib hướng dẫn cách tạo biểu đồ đường cơ bản và tùy chỉnh trong Matplotlib.

Tạo bố cục nhiều biểu đồ **(Subplots)**



A group of graphs and charts

Description automatically generated

Hình 3.9: Tạo bố cục nhiều biểu đồ

Seaborn: Biểu đồ cột, trực quan và màu sắc dễ nhìn.

A screenshot of a computer code

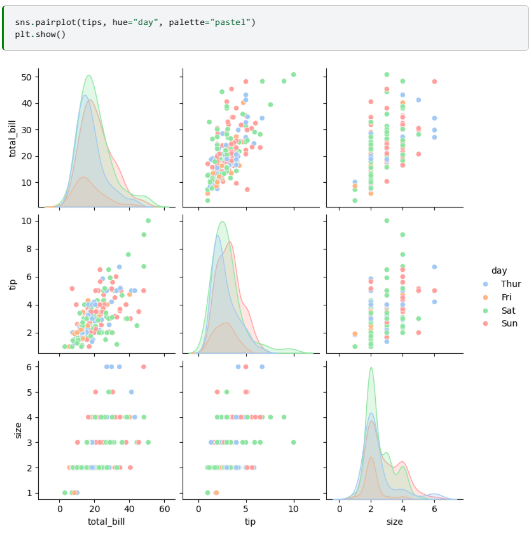
Description automatically generated

A graph of different colored squares

Description automatically generated

Hình 3.10: Biểu đồ cột Seaborn

Chú thích: Biểu đồ cột Seaborn minh họa cách tạo và tùy chỉnh biểu đồ cột sử dụng Seaborn, với dữ liệu trực quan hơn.

 Biểu đồ ghép (Pair Plot) Hiển thị mối quan hệ giữa tất cả các cặp biến.

Hình 3.11:  Biểu đồ ghép

Chú thích: Mỗi ô trong biểu đồ đại diện cho một cặp biến số. Đường chéo chính chứa các biểu đồ phân phối, còn các ô khác là các biểu đồ tán xạ giữa hai biến.

Plotly: Biểu đồ đường tương tác, cho phép zoom, pan, và xuất dữ liệu.:

A screenshot of a computer program

Description automatically generated

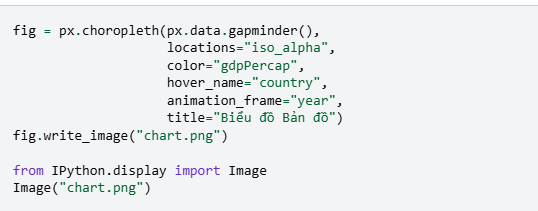
A graph with different colored dots

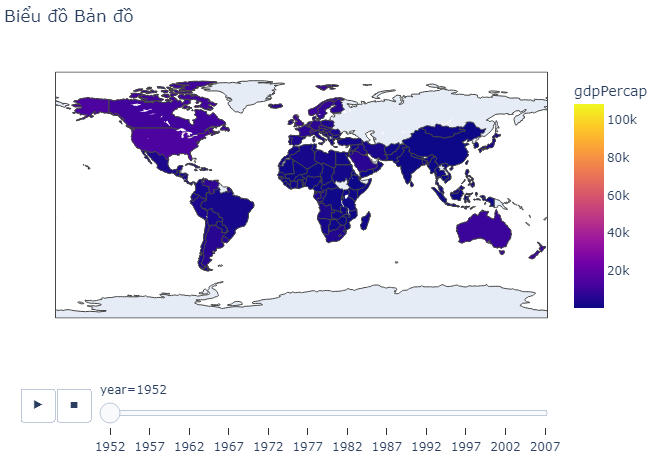
Description automatically generated

Hình 3.11: Biểu đồ tương tác Plotly

Chú Thích: Biểu đồ tương tác Plotly : Tạo các biểu đồ tương tác như biểu đồ đường, biểu đồ cột trong Plotly.

Biểu đồ bản đồ (Choropleth Map):





Hình 3.12: Biểu đồ bản đồ

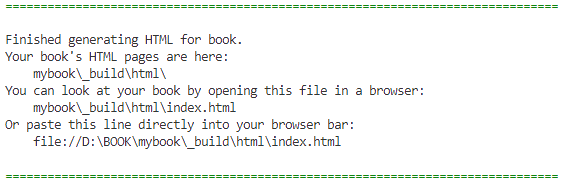
Chú thích**:** px.choropleth(): Tạo biểu đồ bản đồ, trong đó các quốc gia được tô màu theo GDP.

## Xây dựng và xuất bản sách

Sử dụng lệnh jupyter-book build

**jupyter-book build my-book/**

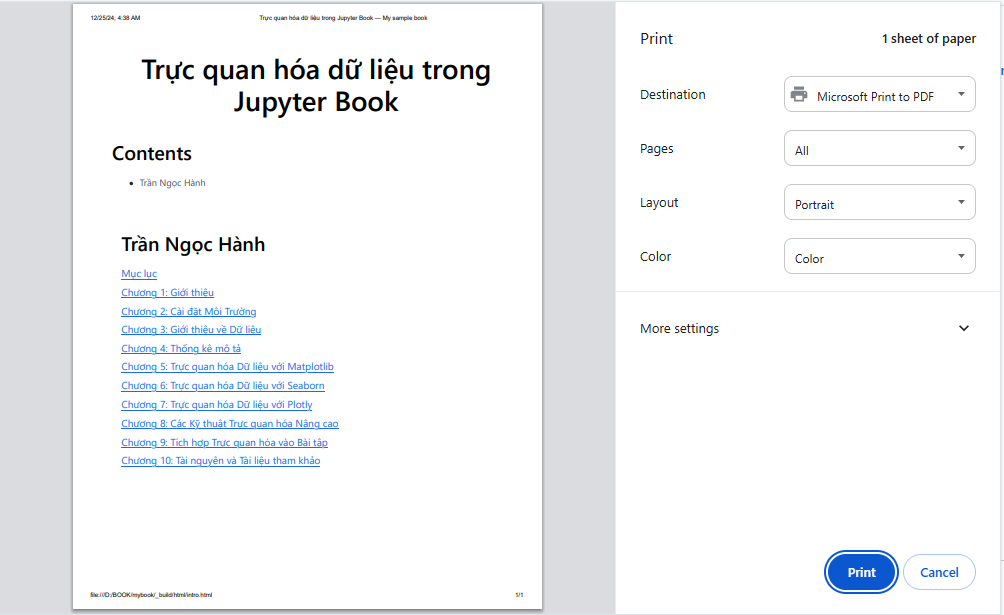
Kiểm tra nội dung sách ở thư mục \_build/html.



Hình 3.13: Kiểm tra nội dung sách

Chú thích: Kiểm tra nội dung sách: Đánh giá và chỉnh sửa nội dung hoàn chỉnh của sách trước khi xuất bản.

Xuất bản sách sang định dạng PDF**:**



Hình 3.14: Xuất bản sách sang định dạng PDF

Chú thích: Xuất bản sách sang định dạng PDF giúp dễ dàng phân phối, bảo toàn định dạng nội dung, hỗ trợ in ấn và lưu trữ lâu dài.

# KẾT QUẢ NGHIÊN CỨU

## Kết quả đạt được

Tạo tài liệu hướng dẫn hoàn chỉnh: Đã biên soạn và xuất bản một tài liệu hướng dẫn chi tiết, bao gồm các chương mục từ lý thuyết cơ bản đến thực hành nâng cao và tài liệu được biên soạn dưới dạng một Jupyter Book, cung cấp giao diện trực quan và tích hợp các ví dụ mã lệnh thực thi trực tiếp.

Cấu trúc nội dung chặt chẽ, logic: Các chương được tổ chức rõ ràng, bắt đầu từ khái niệm cơ bản (cách cài đặt Python, giới thiệu các thư viện) đến các kỹ thuật trực quan hóa phức tạp (biểu đồ động, tích hợp Dashboards) và Tài liệu hỗ trợ cả người mới bắt đầu và những người có kinh nghiệm trong lập trình dữ liệu.

Thực hiện trực quan hóa dữ liệu bằng Python Đã tạo ra các biểu đồ mẫu minh họa cho từng thư viện:

Matplotlib: Biểu đồ đường, biểu đồ cột, biểu đồ phân tán.

Seaborn: Biểu đồ hộp, heatmap, biểu đồ violin.

Plotly: Biểu đồ tương tác, biểu đồ 3D.

Bokeh: Biểu đồ tương tác nâng cao.

Tích hợp các ví dụ với dữ liệu thực tế để người học dễ hình dung và thực hành.

Hướng dẫn xuất bản và chia sẻ Jupyter Book: Đã hướng dẫn chi tiết cách xuất bản Jupyter Book lên các nền tảng như GitHub Pages và xuất tài liệu dưới dạng HTML hoặc PDF và Đảm bảo tài liệu có thể dễ dàng truy cập và chia sẻ với cộng đồng.

## Đánh giá hiệu năng

Hiệu năng của tài liệu: Tài liệu được tối ưu hóa để tải nhanh trên trình duyệt nhờ việc sử dụng định dạng HTML và người dùng có thể chạy các đoạn mã trực tiếp trong tài liệu mà không cần cài đặt thêm công cụ, nhờ tích hợp với Binder.

Hiệu năng của các thư viện trực quan hóa đã thử nghiệm các thư viện trực quan hóa trên tập dữ liệu lớn và đánh giá thời gian xử lý, tốc độ hiển thị biểu đồ:

Matplotlib và Seaborn: Tốc độ nhanh với dữ liệu nhỏ và trung bình.

Plotly: Cung cấp các biểu đồ tương tác mạnh mẽ, tuy nhiên tốc độ giảm khi xử lý dữ liệu lớn.

Bokeh: Hiệu suất tốt, đặc biệt khi sử dụng với ứng dụng web.

## Trải nghiệm người dùng

Dễ sử dụng: Giao diện của Jupyter Book thân thiện, người dùng có thể dễ dàng điều hướng giữa các chương mục và các đoạn mã được viết rõ ràng, kèm theo giải thích chi tiết, giúp người học dễ dàng tiếp cận.

Phản hồi từ thử nghiệm thực tế: Trong quá trình thử nghiệm với một nhóm sinh viên và lập trình viên, tài liệu nhận được phản hồi tích cực về tính trực quan và dễ hiểu và người dùng đánh giá cao phần ví dụ minh họa thực tế và hướng dẫn từng bước cụ thể.

## Giao diện chức năng nghiên cứu

Giao diện của Jupyter Book: trang chủ hiển thị tổng quan nội dung của tài liệu, bao gồm mục lục và hướng dẫn sử dụng.

Chương mục chi tiết: Các trang nội dung được trình bày với tiêu đề rõ ràng, tích hợp hình ảnh và biểu đồ minh họa.

Chạy mã trực tiếp: Mỗi ví dụ mã đều có thể chạy trực tiếp thông qua Binder, cho phép người dùng tương tác với mã mà không cần cài đặt.

Biểu đồ minh họa: Tích hợp nhiều biểu đồ tương tác giúp người dùng dễ dàng tùy chỉnh và khám phá dữ liệu.

Xuất bản và chia sẻ: Tài liệu được triển khai trên GitHub Pages, với các liên kết dễ truy cập và hỗ trợ tải xuống ở định dạng PDF.

## Những khó khăn và cách khắc phục

Khó khăn:

Việc tối ưu hóa hiệu năng khi sử dụng dữ liệu lớn với các thư viện trực quan hóa.

Một số tính năng của Jupyter Book đòi hỏi cấu hình phức tạp, đặc biệt là khi xuất bản trên GitHub Pages

Cách khắc phục:

Sử dụng các kỹ thuật giảm kích thước dữ liệu và tối ưu hóa mã nguồn.

Tham khảo tài liệu chính thức của Jupyter Book và các diễn đàn hỗ trợ để giải quyết các vấn đề kỹ thuật.

## Một số nội dung của tài liệu đã biên soạn

Tham khảo phần phụ lục

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

**Mục tiêu đạt được:** Tài liệu hướng dẫn lập trình trực quan hóa dữ liệu bằng Python đã giúp người học hiểu và thực hành các kỹ thuật trực quan hóa dữ liệu với các thư viện như Matplotlib, Seaborn, và Plotly. Các khái niệm từ cơ bản đến nâng cao đã được giải thích rõ ràng và minh họa qua các ví dụ thực tế

**Hiệu quả:** Qua quá trình triển khai, tài liệu đã hỗ trợ người học nắm vững các kỹ thuật trực quan hóa và áp dụng chúng vào các dự án thực tế. Phản hồi tích cực từ người học cho thấy tài liệu là công cụ hữu ích cho việc học và thực hành.

**Đóng góp**: Tài liệu góp phần nâng cao khả năng sử dụng các công cụ Python trong việc phân tích và trực quan hóa dữ liệu, hỗ trợ việc đưa ra quyết định trong nghiên cứu và công việc.

## Hướng phát triển

**Mở rộng ví dụ thực tế:** Trong tương lai, tài liệu có thể bổ sung thêm các bài toán thực tế, như phân tích dữ liệu lớn hoặc trực quan hóa dữ liệu từ các nguồn không gian (ví dụ: dữ liệu không gian địa lý).

**Cập nhật công nghệ mới**: Liên tục cập nhật các thư viện và công cụ mới trong Python để tài liệu luôn phản ánh các xu hướng mới trong lĩnh vực trực quan hóa dữ liệu, như việc tích hợp với các công cụ AI và machine learning.

**Phát triển phiên bản nâng cao:** Cung cấp thêm các kỹ thuật nâng cao trong việc trực quan hóa dữ liệu, như trực quan hóa dữ liệu động, 3D hoặc trực quan hóa dữ liệu phức tạp.

**Tăng tính tương tác:** Phát triển thêm các ví dụ về biểu đồ tương tác (interactive visualizations) để người học có thể trực tiếp thao tác và trải nghiệm dữ liệu.

# DANH MỤC TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | P. Jupyter, Jupyter Book Documentation, Hoa Kỳ: (tổ chức phi lợi nhuận) và cộng đồng mã nguồn mở., 2018. |
| [2] | P. Jupyter, Jupyter Book Tutorial, Hoa Kỳ: tài liệu này đã được phát triển song song với Jupyter Book., 2018. |
| [3] | P. Jupyter, Creating Interactive Data Science Content, Hoa Kỳ: tài liệu này đã được xuất bản để hướng dẫn người dùng về việc tạo nội dung tương tác trong Jupyter Book., 2020. |
| [4] | R. Python, Data Visualization with Matplotlib and Seaborn, Hoa Kỳ: bài viết này được xuất bản trên Real Python., 2020. |
| [5] | Plotly, Plotly Python Documentation, Hoa Kỳ: Tài liệu chính thức của Plotly liên tục được cập nhật, nhưng Plotly bản ổn định đầu tiên được phát hành vào năm 2013., 2013. |
| [6] | T. D. Science, Jupyter Book for Data Science, Hoa Kỳ: bài viết này được xuất bản trên Towards Data Science., 2021. |

# 

# PHỤ LỤC