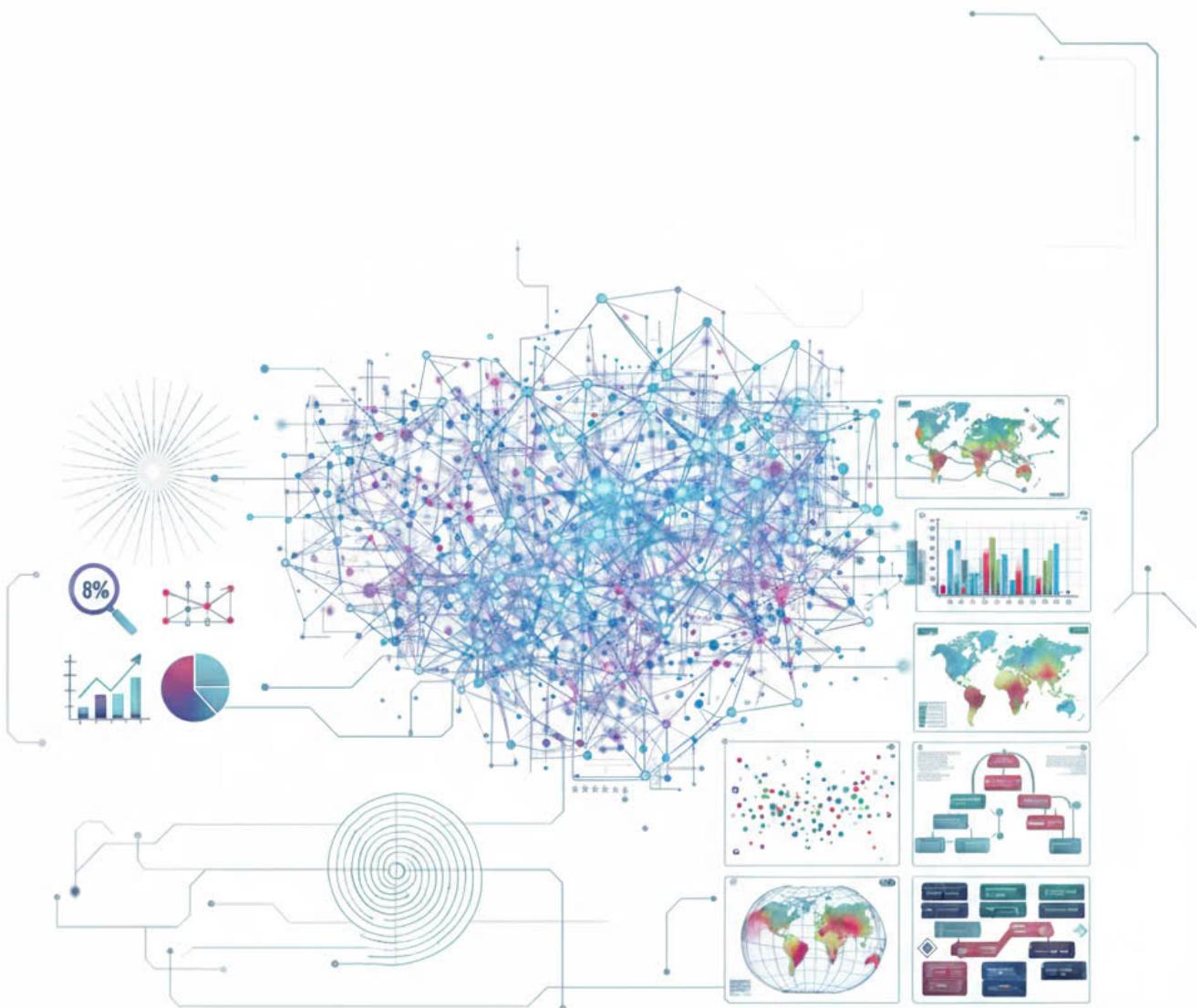


BÀI GIẢNG

TRỰC QUAN HOÁ DỮ LIỆU



PGS. TS. TRẦN VĂN LONG

**BÀI GIẢNG
TRỰC QUAN HÓA DỮ LIỆU**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
HÀ NỘI - 2025**

Lời nói đầu

“Dữ liệu là tài nguyên mới!” – câu nói này giờ đã quá quen thuộc. Dữ liệu không chỉ dồi dào mà đã trở thành tài sản chiến lược của mọi tổ chức và cá nhân. Mọi hành động, mọi cuộc trò chuyện của chúng ta đều đang bị theo dõi và lưu trữ. Chỉ vài ngày sau khi bạn nhắc đến một món đồ, quảng cáo về nó đã xuất hiện trên điện thoại con bạn. thậm chí công ty bảo hiểm còn gửi email khuyên bạn mua thêm gói bảo vệ vì phát hiện bạn hay chạy quá tốc độ qua Google Maps! Ý nghĩa thực sự của những câu nói này là dữ liệu đã trở nên vô cùng phong phú, có giá trị ngày càng tăng, và là nguồn tài nguyên thiết yếu giúp các chính phủ, doanh nghiệp, tổ chức và cá nhân đưa ra quyết định quan trọng. Tôi thích nói rằng dữ liệu đã trở thành một tài sản chiến lược cho các thương hiệu và tổ chức.

Khả năng quan sát và trích xuất kiến thức từ những tập dữ liệu khổng lồ đang trở thành một kỹ năng chuyên môn quan trọng, thậm chí là kỹ năng bắt buộc để tồn tại và phát triển trong thời đại ngày nay. Việc biến dữ liệu thành những hiểu biết sâu sắc, khám phá có ý nghĩa (insight) có thể hành động không còn là kỹ năng chuyên sâu của một nhóm nhỏ chuyên gia nữa, mà đã trở thành yêu cầu cơ bản đối với hầu hết mọi người.

Cuốn sách này tập trung vào môn học cốt lõi: Trực quan hóa và khám phá dữ liệu (EDA-Exploratory Data Analysis) là bước đầu tiên và quan trọng nhất trong hành trình phân tích dữ liệu. Đây là giai đoạn chúng ta làm quen, làm sạch, mô tả, trực quan hóa và “lắng nghe” câu chuyện mà dữ liệu muôn kể. Cuốn sách được viết dưới dạng hướng dẫn thực hành gần gũi, đầy ví dụ và bài tập thực tế, phù hợp cho sinh viên, người mới bắt đầu cũng như các bạn đã làm việc với dữ liệu muôn nâng cao kỹ năng.

Mục tiêu lớn nhất của cuốn sách không chỉ giúp bạn làm được với trực quan hóa và khám phá dữ liệu, mà còn giúp bạn thích thú và tự tin khi khám phá dữ liệu bởi trực quan hóa không phải công việc khô khan chỉ vẽ các hình ảnh mà là một cuộc phiêu lưu đầy bất ngờ.

Hà Nội, 12/2025

Tác giả

Mục lục

Lời nói đầu	i
Mục lục	vii
Danh sách hình vẽ	xi
Danh sách bảng	xiii
Chương 1 GIỚI THIỆU VỀ PHÂN TÍCH KHÁM PHÁ DỮ LIỆU	1
1.1 Khoa học dữ liệu -(Data science)	2
1.1.1 Khoa học dữ liệu	3
1.1.2 Dữ liệu Lớn (Big Data)	4
1.1.3 Phân tích dữ liệu mô tả (Data Analysis) và Phân tích dữ liệu (Data Analytics)	5
1.1.4 Khai phá dữ liệu (Data mining) và Khám phá tri thức trong sơ sở dữ liệu (KDD)	6
1.2 Các nghề nghiệp trong Khoa học dữ liệu	8
1.2.1 Kỹ năng Cần thiết	9
1.2.2 Các Nghề nghiệp Phổ biến trong Khoa học Dữ liệu	9
1.3 Quy trình làm việc trong Khoa học dữ liệu	12
1.4 Dữ liệu	16
1.4.1 Dữ liệu dạng bảng	17
1.4.2 Phân loại kiểu dữ liệu	18
1.4.3 Phân loại theo bản chất	19
1.4.4 Phân loại theo loại biến	20
1.4.5 Tính biến đổi theo thời gian (Time Variability)	20
1.4.6 Phân loại theo số chiều (Dimensionality)	22
1.4.7 Phân loại theo quyền sở hữu (Ownership)	22
1.5 Chuẩn bị dữ liệu	23
1.5.1 Lấy Mẫu (Sampling)	24
1.5.2 Xử lý Giá trị Thiếu (Missing Values)	24

1.5.3 Chuẩn hóa (Normalization)	24
1.6 Dữ liệu	25
1.6.1 Forest Fires Dataset (Bộ dữ liệu cháy rừng)	25
1.6.2 Mammographic Mass Dataset (Bộ dữ liệu khối u tuyến vú)	26
1.6.3 Gapminder Dataset (Bộ dữ liệu Gapminder)	26
1.6.4 Daily Delhi Climate	27
1.6.5 Auto-mpg dataset	27
1.6.6 IMDb Movie Reviews Corpus (Tập dữ liệu Đánh giá Phim IMDb) . .	27
1.6.7 Zachary's Karate Club Dataset (Bộ dữ liệu Câu lạc bộ Karate của Zachary)	27
1.7 Bài tập	27
Chương 2 PHÂN TÍCH MÔ TẢ	31
2.1 Phân phối	31
2.1.1 Phân phối tần số và tần suất	31
2.1.2 Bảng Tần suất (Frequency Tables)	32
2.1.3 Hình dạng Phân phối (Shapes of Distributions)	34
2.1.4 Bảng Ngẫu nhiên (Contingency Tables)	35
2.2 Các số đo đặc trưng	38
2.2.1 Số đo xu hướng trung tâm	38
2.2.2 Các số đo biến thiên (Variability Measures)	40
2.2.3 Các số đo hình dạng (Measures of Shape)	41
2.3 Các độ đo mối quan hệ	45
2.3.1 Hiệp phương sai (Covariance)	45
2.3.2 Hệ số tương quan (Correlation)	46
2.3.3 So sánh các phép đo tương quan	51
2.3.4 Hồi quy tuyến tính đơn	51
2.4 BÀI TẬP	52
Chương 3 CÁC NGUYÊN TẮC TRỰC QUAN HÓA DỮ LIỆU	55
3.1 Quá trình nhận thức thị giác	55
3.1.1 Xử lý tiền chú ý (Preattentive Processing)	59
3.1.2 Các nguyên tắc Gestalt	62
3.1.3 Các nguyên tắc của Tufte	66
3.1.4 Các nguyên tắc sử dụng màu sắc	74
3.2 Các nguyên tắc thiết kế cho trực quan hóa dữ liệu	83

3.2.1	Bảng (Tables)	84
3.2.2	Đồ thị (Graphs)	85
3.3	Bài tập	87
Chương 4	PHƯƠNG PHÁP TRỰC QUAN HÓA DỮ LIỆU	89
4.1	Phân phối (Distributions)	90
4.1.1	Histogram (Biểu đồ Tần suất)	90
4.1.2	Boxplot (Biểu đồ Hộp và Râu - Box and Whisker Plot)	91
4.1.3	Violin Plot (Biểu đồ Violin)	93
4.2	Các Mối liên hệ (Associations)	94
4.2.1	Scatter Plot (Biểu đồ phân tán)	95
4.2.2	Bubble Chart (Biểu đồ Bong bóng)	96
4.2.3	Scatterplot Matrix Plot (Ma trận biểu đồ phân tán)	97
4.2.4	Heatmaps và Correlograms (Bản đồ nhiệt và Biểu đồ tương quan) .	100
4.3	Số lượng (Amounts)	101
4.3.1	Bar Chart (Biểu đồ thanh)	102
4.3.2	Radar Chart (Biểu đồ Radar)	103
4.4	Tỷ lệ (Proportions)	106
4.4.1	Pie Chart (Biểu đồ Tròn)	106
4.4.2	Doughnut Chart (Biểu đồ Vòng)	107
4.4.3	Treemap (Biểu đồ Hình cây)	109
4.5	Biểu đồ đường và dòng	112
4.5.1	Line Chart (Biểu đồ đường)	112
4.5.2	Sankey Chart (Biểu đồ Sankey)	113
4.5.3	Gantt Chart (Biểu đồ Gantt)	115
4.6	Dữ liệu Địa lý Không gian (Geospatial)	116
4.6.1	Choropleth Map (Bản đồ Phân vùng theo màu)	117
4.6.2	Bubble Map (Bản đồ Bong bóng)	119
4.7	Bài tập	120
Chương 5	MỘT SỐ LOẠI DỮ LIỆU ĐẶC BIỆT	123
5.1	Chuỗi Thời Gian (Time Series)	123
5.1.1	Các Loại và Đặc Điểm của Chuỗi Thời Gian	123
5.1.2	Mục tiêu Phân tích Khám phá Chuỗi Thời Gian (EDA)	124
5.1.3	Trực quan hóa Dữ liệu Chuỗi Thời Gian	125
5.1.4	Trung bình trượt và phân rã theo mùa	129

5.2	Dữ Liệu Văn Bản và Tài Liệu (Text and Document Data)	132
5.2.1	Mục tiêu Phân tích Khám phá Dữ liệu Văn bản và Tài liệu	133
5.2.2	Cấu trúc hóa văn bản (Text Structuring)	133
5.2.3	Phân tích Mô tả Dữ liệu Văn bản và Tài liệu	136
5.2.4	Trực quan hóa Dữ liệu Văn bản và Tài liệu	136
5.3	Cây và Mạng (Trees and Networks)	143
5.3.1	Các Khái niệm của Lý thuyết đồ thị (Concepts of Graph Theory) . .	144
5.3.2	Mục tiêu phân tích khám phá Cây và Mạng (EDA)	144
5.3.3	Phân tích Mô tả đối với Cây (Descriptive Analysis for Trees)	148
5.3.4	Trực quan hóa Cây (Visualizing Trees)	149
5.3.5	Phân tích Mô tả đối với Mạng (Descriptive Analysis for Networks) .	153
5.3.6	Trực quan hóa mạng (Visualizing Networks)	156
5.4	Dữ liệu nhiều chiều	158
5.4.1	Hệ toạ độ song song (Parallel coordinates)	159
5.4.2	Hệ toạ độ hình sao (Star coordinates)	162
5.4.3	Phương pháp chiếu xuyên tâm (Radviz)	165
5.5	Bài tập	168
Chương 6	KỂ CHUYỆN BẰNG DỮ LIỆU VÀ THIẾT KẾ BẢNG ĐIỀU KHIỂN	171
6.1	Kể chuyện bằng Dữ liệu (Data Storytelling)	171
6.1.1	Các Bước Thiết kế Câu chuyện Dữ liệu	173
6.2	Thiết kế Bảng điều khiển (Dashboard Design)	175
6.2.1	Chọn các Trực quan hóa Thích hợp (Selecting Appropriate Visualizations)	176
6.2.2	Thiết kế Bộ cục Bảng điều khiển (Designing the Dashboard Layout)	176
6.2.3	Chọn Phối màu (Choosing a Color Scheme)	177
6.2.4	Áp dụng Tính tương tác (Applying Interactivity)	177
6.2.5	Bảo mật và Kiểm soát Truy cập (Security and Access Control) . . .	178
6.2.6	Kiểm tra và Lặp lại (Test and Iterate)	178
6.3	Nghiên cứu Tình huống 1: GAPMINDER DATASET	179
6.3.1	Kể chuyện bằng Dữ liệu (Data Storytelling)	179
6.3.2	Thiết kế Bảng điều khiển (Dashboard Design)	179
6.4	Nghiên cứu tình huống 2: SUPERSTORE SALES DATASET	181
6.4.1	Kể chuyện bằng Dữ liệu (Data Storytelling)	182
6.4.2	Thiết kế Bảng điều khiển (Dashboard Design)	183

6.5 Bài tập	184
Tài liệu tham khảo	188

Danh sách hình vẽ

1.1	Sơ đồ lặp trong nghiên cứu.	3
1.2	Các yếu tố chính của Khoa học dữ liệu.	4
1.3	Các kỹ năng cần thiết cho nhà khoa học dữ liệu.	10
1.4	Vị trí nghề nghiệp trong khoa học dữ liệu và các kỹ năng.	13
1.5	Quy trình làm việc trong Khoa học dữ liệu.	14
1.6	Từ dữ liệu đến thông tin và đến tri thức.	16
1.7	Phân loại theo biến của dữ liệu.	21
1.8	Phân loại các dạng dữ liệu.	23
2.1	Biểu đồ hình tròn.	33
2.2	Biểu đồ phân phối.	36
2.3	Biểu đồ phân phối với trung bình, trung vị, mode.	44
2.4	Các dạng biểu đồ về phân phối với độ nhọn khác nhau.	45
3.1	Bộ tứ dữ liệu Anscombe.	57
3.2	Bộ dữ liệu Datasaurus Dozen.	58
3.3	Nguyên lý liên tục.	63
3.4	Nguyên lý đóng kín.	64
3.5	Nguyên lý tương đồng.	64
3.6	Nguyên lý tương đồng.	65
3.7	Nguyên lý đối xứng.	66
3.8	Nguyên lý hình nền.	66
3.9	Nguyên lý đồng bộ.	67
3.10	Hệ số nói dối.	68
3.11	Tỷ lệ mực dữ liệu.	69
3.12	Mô tả Bản đồ Minard về Chiến dịch Nga năm 1812.	71
3.13	Biểu đồ của John Snow về dịch tả ở Soho, London năm 1854.	73

3.14 Không gian màu RGB.	75
3.15 Hệ thống thị giác của chúng ta tự động hiểu các kênh độ chói (luminance) và độ bão hòa (saturation) theo thứ tự, nhưng kênh sắc độ (hue) thì không.	76
3.16 Sự phân loại bản đồ màu (colormap categorization) phản ánh một phần các loại dữ liệu: phân loại (categorical) so với có thứ tự (ordered), và tuần tự (sequential) cùng phân kỳ (diverging) nằm trong nhóm có thứ tự.	78
3.17 Bảng màu cho dữ liệu tuần tự.	80
3.18 Bảng màu cho dữ liệu phân kỳ.	81
3.19 Bảng màu cho dữ liệu có chu kỳ.	82
3.20 Bảng màu cho dữ liệu định tính.	83
3.21 Các thành phần chính của một bảng dữ liệu.	84
3.22 Các thành phần chính của một đồ thị.	86
 4.1 Histogram và Boxplot.	93
4.2 Violin plot và Boxplot.	95
4.3 Scatter plot và Bubble chart.	98
4.4 Scatter matrix plot.	99
4.5 Heatmap và Correlograms.	101
4.6 Biểu đồ thanh.	104
4.7 Radar Chart.	105
4.8 Pie Chart.	108
4.9 Donut Chart.	109
4.10 Treemap.	111
4.11 Biểu đồ đường.	114
4.12 Bản đồ.	118
 5.1 Trực quan hóa cuối thời gian.	127
5.2 Trực quan hóa cuối thời gian.	128
5.3 Trực quan hóa cuối thời gian.	128
5.4 Trung bình trượt.	130
5.5 Phân tích chuỗi thời gian.	132
5.6 Word cloud và Bar chart các từ phổ biến nhất.	139
5.7 Tag cloud với bi-grams và tri-grams.	141
5.8 Categorial heatmap.	143
5.9 Mạng có trọng số.	146
5.10 Cây.	148

5.11	Trực quan hoá Cây.	151
5.12	Trực quan hoá Mạng.	158
5.13	Trực quan hoá Mạng bằng ma trận kè.	159
5.14	Hệ toạ độ song song.	162
5.15	Hệ toạ độ hình sao.	165
5.16	Hệ toạ độ xuyên tâm (phương pháp Radviz).	168
6.1	Mối quan hệ giữa dữ liệu, tường thuật và hình ảnh để tạo nên một câu chuyện xoay quanh dữ liệu có sẵn và bối cảnh của nó.	172
6.2	Quá trình thiết kế về kể chuyện dữ liệu.	173
6.3	Quá trình thiết kế bảng điều khiển.	176
6.4	Sơ đồ thiết kế bảng điều khiển.	180
6.5	Bảng điều khiển cho dữ liệu Gapminder.	181
6.6	Sơ đồ thiết kế bảng điều khiển.	184
6.7	Bảng điều khiển cho dữ liệu Superstore Sales.	184

Danh sách bảng

1.1	Bôn cấp độ của Data Analytics	6
1.2	So sánh Data Analysis và Data Analytics	7
1.3	So sánh Data Analytics và Data Science	8
1.4	So sánh các loại Dữ liệu	19
3.1	Tóm tắt các nguyên tắc Gestalt	68

Chương 1

GIỚI THIỆU VỀ PHÂN TÍCH KHÁM PHÁ DỮ LIỆU

Bối cảnh và tầm quan trọng của dữ liệu

Sự thay đổi của thế giới: Cho đến gần đây, các nhà quản lý doanh nghiệp đưa ra hầu hết các quyết định chỉ dựa trên kiến thức và kinh nghiệm. Tuy nhiên, thế giới ngày nay năng động hơn nhiều, và người dùng và người tiêu dùng trở nên thiếu kiên nhẫn, kết nối và khắt khe hơn. Sự thay đổi trong hành vi này và cạnh tranh thị trường gia tăng đang buộc các công ty phải thích ứng.

Dữ liệu là tài sản chiến lược: Dữ liệu đã trở nên dồi dào, vô cùng quý giá và cần thiết để cung cấp năng lượng cho nhiều khía cạnh của xã hội hiện đại, tương tự như dầu mỏ hoặc tiền tệ. Khả năng biến dữ liệu thành thông tin chi tiết có ý nghĩa là một kỹ năng chuyên môn và là một điều cần thiết để hiểu thế giới xung quanh.

Khái niệm phân tích khám phá dữ liệu (Exploratory Data Analysis-EDA): Định nghĩa của phân tích khám phá dữ liệu:

- Phân tích khám phá dữ liệu bao gồm bất kỳ phương pháp phân tích dữ liệu nào không bao gồm việc lập mô hình hoặc suy luận.
- Trong phân tích khám phá dữ liệu, dữ liệu chỉ đơn thuần là được thao tác, tóm tắt và trực quan hóa (trình bày).
- Trọng tâm của phân tích khám phá dữ liệu là hiểu, mô tả đặc điểm, tóm tắt và trực quan hóa dữ liệu

Mục tiêu của phân tích khám phá dữ liệu: Mục tiêu chung của phân tích khám phá dữ liệu là hiểu các đặc điểm và xu hướng của dữ liệu, cũng như trích xuất các chỉ số và hình ảnh trực quan. Các mục tiêu cụ thể bao gồm:

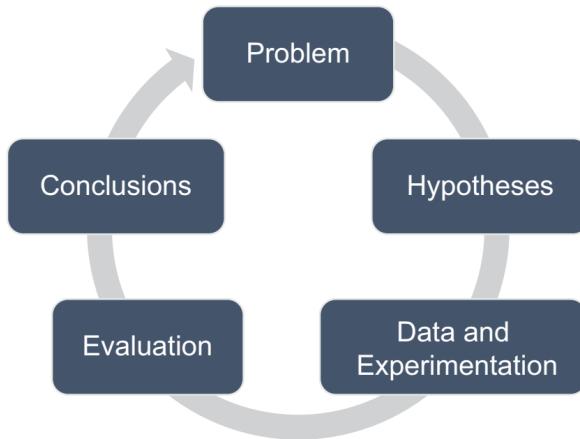
- Hiểu được sự phân bố và cấu trúc của dữ liệu.
- Tóm tắt đặc điểm dữ liệu.
- Trích xuất thông tin chi tiết và chỉ số từ dữ liệu.
- Để xác định sự liên quan và/hoặc lựa chọn các biến.
- Để hình dung các mối quan hệ tiềm năng giữa các biến.
- Để xác định những điểm bất thường.
- Cho phép áp dụng và/hoặc lựa chọn các phương pháp dựa trên học tập.

1.1 Khoa học dữ liệu -(Data science)

Khoa học dữ liệu là sự kết hợp giữa dữ liệu, như một nguyên liệu thô, và khoa học, cụ thể hơn là phương pháp khoa học, như một quy trình. Điều này có nghĩa là khoa học dữ liệu sẽ lấy dữ liệu làm đầu vào và, bằng cách sử dụng phương pháp khoa học, sẽ tạo ra một đầu ra nhất định. Phương pháp khoa học là một quy trình có hệ thống nhằm tìm ra câu trả lời hoặc giải pháp cho một vấn đề nhất định, bắt đầu từ một hoặc nhiều giả thuyết cho trước. Quy trình này bao gồm một tập hợp các bước cụ thể (Hình ??):

1. Bắt đầu bằng một quan sát hoặc một vấn đề cần giải quyết;
2. Xác định các câu hỏi cần trả lời hoặc các giả thuyết cần xác thực;
3. Thu thập dữ liệu và thực hiện các thí nghiệm dựa trên dữ liệu và giả thuyết;
4. Đánh giá (phân tích) kết quả; và
5. Rút ra kết luận

Các bước này được lặp lại cho đến khi đạt được kết quả mong muốn. Điểm đầu tiên cần lưu ý ở đây là, khác với mô hình tính toán (lập trình) tiêu chuẩn, kết quả của quá trình khoa học dữ liệu không được biết trước. Thay vào đó, nó được thu thập bằng cách áp dụng một quy trình lặp đi lặp lại trên dữ liệu có sẵn. Ngược lại, trong phương pháp



Hình 1.1: Sơ đồ lặp trong nghiên cứu.

lập trình chuẩn, kết quả đã được biết trước, và việc lập trình chỉ thành công khi đạt được kết quả như vậy. Ví dụ, một chương trình kiểm soát hàng tồn kho trong kho phải thêm một mặt hàng nếu nó đang vào kho và trừ một mặt hàng nếu nó đang ra khỏi kho. Đây là một nhiệm vụ đơn giản và có thể dự đoán được. Ngược lại, một chương trình học cách phát hiện bất thường chỉ có thể được đánh giá hiệu suất sau khi mô hình được tạo và thử nghiệm trên một số dữ liệu nhất định, và nó có thể đạt được kết quả mong muốn hoặc không. Trong trường hợp này, kết quả sẽ chỉ được biết sau khi giải pháp được thiết kế, triển khai và thử nghiệm.

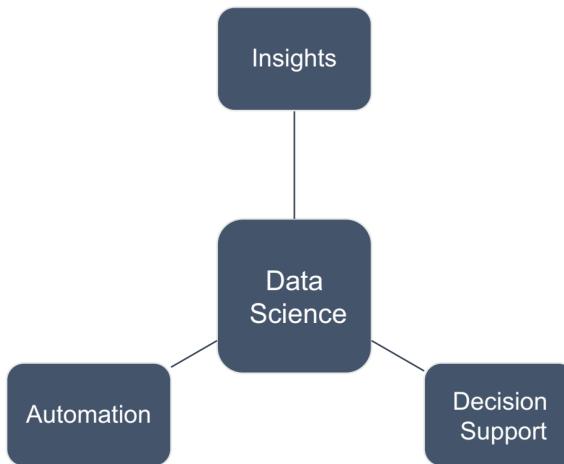
1.1.1 Khoa học dữ liệu

Khoa học dữ liệu được định nghĩa là sự kết hợp giữa dữ liệu (nguyên liệu thô) và khoa học (cụ thể hơn là phương pháp khoa học) như một quy trình. Khoa học dữ liệu sẽ lấy dữ liệu làm đầu vào và sử dụng phương pháp khoa học (một quy trình có hệ thống để tìm câu trả lời hoặc giải pháp cho một vấn đề) để tạo ra một đầu ra nhất định. Các ứng dụng tính toán (mục đích sử dụng chính) được tạo ra từ một dự án khoa học dữ liệu có thể là một trong ba loại sau:

1. Thông tin chi tiết (Insights): Nhằm mục đích trích xuất một số kiến thức chung để giúp chúng ta hiểu rõ hơn về dữ liệu và cấu trúc của nó. Điều này có thể được cung cấp thông qua việc xây dựng một bảng điều khiển (dashboard) với các kiến thức và/hoặc các Chỉ số Hiệu suất Chính (KPIs) được trích xuất.
2. Hệ thống Hỗ trợ Ra quyết định (Decision Support System - DSS): Thiết kế các công

cụ hỗ trợ quá trình ra quyết định, cung cấp các câu trả lời hoặc giải pháp thử nghiệm dựa trên dữ liệu có sẵn để người ra quyết định có thể đưa ra quyết định nhanh hơn và sáng suốt hơn.

3. Tự động hóa (Automation): Thay thế sự can thiệp của con người bằng một thuật toán hoặc máy móc, có nghĩa là sự can thiệp của con người không còn cần thiết cho một hành động hoặc quyết định nào đó. Tự động hóa được xem là một bước tiến xa hơn so với DSS



Hình 1.2: Các yếu tố chính của Khoa học dữ liệu.

1.1.2 Dữ liệu Lớn (Big Data)

Khác với khoa học dữ liệu, Dữ liệu Lớn là lĩnh vực xử lý việc lưu trữ, xử lý và phân tích các tập dữ liệu lớn. Dữ liệu lớn được đặc trưng bởi 3V:

- Velocity (Vận tốc)
- Variety (Đa dạng)
- Volume (Khối lượng)

Các nguồn tài liệu cũng lưu ý rằng đã có thêm các yếu tố bổ sung vào 3V, bao gồm tính

- Veracity (Xác thực),

- Value (Giá trị), và
- Variability (Khả biến).

1.1.3 Phân tích dữ liệu mô tả (Data Analysis) và Phân tích dữ liệu (Data Analytics)

Phân tích dữ liệu mô tả (Data Analysis): Phân tích dữ liệu mô tả là một khái niệm hạn chế hơn. Phân tích dữ liệu mô tả các bước được sử dụng để tiền xử lý dữ liệu (ví dụ: chuyển đổi và lập mô hình dữ liệu), trích xuất thống kê mô tả cơ bản, thông tin chi tiết, chỉ số và trực quan hóa dữ liệu, với các báo cáo và bảng điều khiển là sản phẩm chính. Data Analysis là một tập hợp con của Data Analytics và bản chất là một phương pháp tiếp cận mô tả và trực quan hóa, không yêu cầu học từ dữ liệu (không sử dụng thuật toán học máy).

Phân tích dữ liệu mô tả: “Tôi đang **hiểu** dữ liệu như thế nào?”

Ví dụ: “Doanh thu quý này giảm 15% so với quý trước” hoặc “80% khách hàng đến từ TP.HCM”

Phân tích dữ liệu (Data Analytics): Là một thuật ngữ rộng, bao gồm các khái niệm, ứng dụng và công cụ để phát triển các giải pháp dựa trên dữ liệu. Nó bao gồm tất cả các bước cần thiết để thu thập, chuẩn bị và phân tích dữ liệu nhằm cung cấp các giải pháp (thông tin chi tiết, hỗ trợ quyết định và tự động hóa). Tài liệu giả định rằng Data Analytics thường bao gồm một số loại thuật toán học máy trong quy trình của nó.

Phân tích dữ liệu: “Tôi dùng dữ liệu để **ra quyết định kinh doanh và tạo lợi thế cạnh tranh** như thế nào?”

Ví dụ: “Nếu tăng ngân sách quảng cáo Facebook 20%, doanh thu sẽ tăng thêm 1,2 tỷ với xác suất 87%” hoặc “Khách hàng nhóm A có nguy cơ rời bỏ chiếm 72% dẫn đến gửi ưu đãi ngay”

Bốn cấp độ của Data Analytics:

- Descriptive-Phân tích mô tả: “Đã xảy ra chuyện gì?”
- Diagnostic-Phân tích chuẩn đoán: “Tại sao lại xảy ra?”
- Predictive-Phân tích dự đoán: “Sắp tới sẽ ra sao?”
- Prescriptive-Phân tích chỉ định: “Ta phải làm gì đây?”

Bảng 1.1: Bốn cấp độ của Data Analytics

Tiêu chí	Descriptive	Diagnostic	Predictive	Prescriptive
Cấp độ	1. Thấp	2. Trung bình	3. Cao	4. Rất cao
Tiếng Anh	Descriptive Analytics	Diagnostic Analytics	Predictive Analytics	Prescriptive Analytics
Tiếng Việt	Phân tích mô tả	Phân tích chẩn đoán	Phân tích dự đoán	Phân tích chỉ định
Câu hỏi	Điều gì đã xảy ra?	Tại sao nó xảy ra?	Điều gì sẽ xảy ra?	Chúng ta nên làm gì?
Mục đích	Tóm tắt hiện tại & quá khứ	Tìm nguyên nhân gốc rễ	Dự báo xu hướng, xác suất	Đề xuất hành động tối ưu
Phương pháp	Thông kê cơ bản, dashboard, KPI	Phân tích tương quan	Machine Learning, time-series, AI	Optimization, simulation, recommendation
Ví dụ	Doanh thu tháng 10 đạt 15 tỷ	Doanh thu giảm do mất khách miền Bắc	72% khách sẽ churn trong 30 ngày	Gửi ưu đãi 20% + tăng ngân sách quảng cáo 1,5 tỷ

1.1.4 Khai phá dữ liệu (Data mining) và Khám phá tri thức trong cơ sở dữ liệu (KDD)

Data Mining (Khai phá Dữ liệu) : Là một khái niệm được giới thiệu như một phép ẩn dụ cho quá trình khai thác khoáng sản quý giá, nơi một mỏ được khám phá bằng các công cụ và phương pháp thích hợp để thu được vật liệu có giá trị.

Knowledge Discovery in Databases (KDD - Khám phá tri thức trong cơ sở dữ liệu): Là một quá trình rộng hơn bao gồm toàn bộ quy trình, từ khôi phục dữ liệu đến phân tích kinh doanh, chuẩn bị dữ liệu, phân tích dữ liệu và khai thác, và cuối cùng là tích hợp giải pháp.

Tóm lại, khoa học dữ liệu (Data Science) là tên của toàn bộ lĩnh vực. Khám phá Tri thức trong Cơ sở Dữ liệu tương ứng với quá trình sử dụng khoa học dữ liệu để đạt được các mục tiêu của nó. Phân tích dữ liệu mô tả (Data Analysis), Khai phá dữ liệu (Data Mining), và Phân tích dữ liệu (Data Analytics) là các quy trình nằm trong Khám phá tri thức trong cơ sở dữ liệu.

Bảng 1.2: So sánh Data Analysis và Data Analytics

Tiêu chí	Data Analysis (Phân tích dữ liệu mô tả)	Data Analytics (Phân tích dữ liệu)
Phạm vi	Hẹp hơn, là một giai đoạn/bước cụ thể trong quy trình xử lý dữ liệu	Rộng hơn, là toàn bộ lĩnh vực bao gồm nhiều giai đoạn và mục tiêu khác nhau
Mục tiêu chính	Hiểu dữ liệu hiện tại (Điều gì đã xảy ra? Đang xảy ra gì?)	Hỗ trợ ra quyết định kinh doanh, dự đoán và tối ưu (Tại sao xảy ra? Sẽ xảy ra gì? Nên làm gì?)
Các loại chính	<ul style="list-style-type: none"> • Descriptive Analysis (Mô tả) • Diagnostic Analysis (Chuẩn đoán) 	<ul style="list-style-type: none"> • Descriptive (Mô tả) • Diagnostic (Chuẩn đoán) • Predictive (Dự đoán) • Prescriptive (Chỉ định)
Độ phức tạp	Thấp → Trung bình (thống kê cơ bản, EDA, báo cáo)	Trung bình → Rất cao (machine learning, AI, automation)
Người thực hiện	Data Analyst, Business Analyst	Data Analyst, Data Scientist, Data Engineer, Analytics Engineer, BI Analyst...
Kết quả đầu ra điển hình	Báo cáo, dashboard mô tả, bảng tóm tắt, biểu đồ EDA	Dự báo doanh thu, mô hình churn, hệ thống gợi ý, phân khúc khách hàng tự động, đề xuất hành động tối ưu
Công cụ phổ biến	Excel, SQL, Power BI, Tableau, Python (pandas, matplotlib, seaborn)	Python/R (scikit-learn, TensorFlow), Spark, Snowflake, Databricks, AutoML,

Bảng 1.3: So sánh Data Analytics và Data Science

Tiêu chí	Data Analytics	Data Science
Phạm vi	Tập trung khai thác dữ liệu hiện có để hỗ trợ ra quyết định kinh doanh	Lĩnh vực rộng, kết hợp toán học, thống kê, lập trình và kiến thức chuyên môn để tạo ra tri thức và sản phẩm mới từ dữ liệu
Mục tiêu chính	Trả lời các câu hỏi kinh doanh: Điều gì đã xảy ra? Tại sao? Sẽ ra sao? Nên làm gì?	Phát hiện mẫu hình mới, xây dựng mô hình dự đoán chính xác cao, tạo sản phẩm AI/ML
Câu hỏi điển hình	“Doanh thu miền Nam giảm vì sao?” “Nhóm khách hàng nào đáng đầu tư quý nhất?”	“Làm sao dự đoán chính xác khách sẽ mua gì trong 7 ngày tới?” “Xây hệ thống phát hiện gian lận realtime như thế nào?”
Độ sâu kỹ thuật	Trung bình → Cao (SQL, Excel, Power BI, Tableau, Python/R cơ bản)	Rất cao (toán nâng cao, machine learning/deep learning, big data, software engineering)
Kết quả đầu ra	Báo cáo, dashboard, insight kinh doanh, đề xuất hành động	Mô hình ML/DL, hệ thống AI, sản phẩm dữ liệu (recommendation engine, fraud detection, chatbot, xe tự lái...)
Vai trò phổ biến	Data Analyst, Business Analyst, BI Analyst, Analytics Manager	Data Scientist, Machine Learning Engineer, AI Engineer, Research Scientist
Thời gian tạo giá trị	Nhanh (vài ngày đến vài tuần)	Thường chậm hơn (vài tháng đến vài năm nếu làm sản phẩm AI)
Ví dụ thực tế tại Việt Nam	Tiki/Shopee tối ưu khuyến mãi từ hành vi khách; ngân hàng làm dashboard KPI chi nhánh	VNPAY phát hiện gian lận realtime; VinAI nghiên cứu xe tự lái; FPT AI phát triển chatbot tiếng Việt
Tóm tắt một câu	Dùng dữ liệu để cải thiện quyết định kinh doanh hiện tại	Dùng khoa học và công nghệ để tạo ra tri thức mới và sản phẩm thông minh từ dữ liệu

1.2 Các nghề nghiệp trong Khoa học dữ liệu

Khoảng năm 2010, khi khoa học dữ liệu bắt đầu nhận được sự chú ý ngày càng tăng, “nhà khoa học dữ liệu” (data scientist) được coi là nghề nghiệp duy nhất trong lĩnh vực này. Tuy nhiên, do phạm vi kỹ năng cần thiết cho vị trí này quá rộng—bao gồm cả kỹ năng cứng (technical skills) và kỹ năng mềm (soft skills)—nên theo thời gian, nhiều chuyên môn hóa đã xuất hiện để dễ dàng xây dựng và tổ chức các đội khoa học dữ liệu hơn.

Các nghề nghiệp phổ biến nhất hiện nay trong khoa học dữ liệu là nhà phân tích

nghiệp vụ (business analyst), nhà khoa học dữ liệu (data scientist), nhà phân tích dữ liệu (data analyst), kỹ sư dữ liệu (data engineer), kỹ sư học máy (machine learning engineer), và Giám đốc Dữ liệu (Chief Data Officer - CDO)

1.2.1 Kỹ năng Cần thiết

Kỹ năng Cần thiết cho Nhà Khoa học Dữ liệu (Vai trò Nền tảng) Ban đầu, nhà khoa học dữ liệu được kỳ vọng phải sở hữu nhiều kỹ năng cứng và mềm phức tạp:

Kỹ năng Cứng (Hard Skills)

1. Toán học và Thống kê: Cần thiết cho phân tích và tổng hợp các giải pháp dựa trên dữ liệu. Kiến thức yêu cầu bao gồm học máy (machine learning), học sâu (deep learning), mô hình thống kê, xác suất, đại số tuyến tính, và lập trình toán học.

2. Máy tính (Computing): Khoa học dữ liệu là một phương pháp tiếp cận mang tính toán học nội tại (intrinsically computational). Yêu cầu phải thành thạo lập trình, cơ sở dữ liệu, kỹ thuật phần mềm, tính toán hiệu năng cao (high-performance computing), xử lý song song và phân tán, thiết kế bảng điều khiển (dashboard design), và quản lý dự án.

3. Kinh doanh (Business): Cần thiết để sử dụng dữ liệu nhằm trích xuất kiến thức (insights), thiết kế Hệ thống Hỗ trợ Ra quyết định (DSS), hoặc tự động hóa hệ thống. Cần có hiểu biết chung về quản lý, tài chính, vận hành, và các lĩnh vực quan trọng khác của doanh nghiệp mục tiêu.

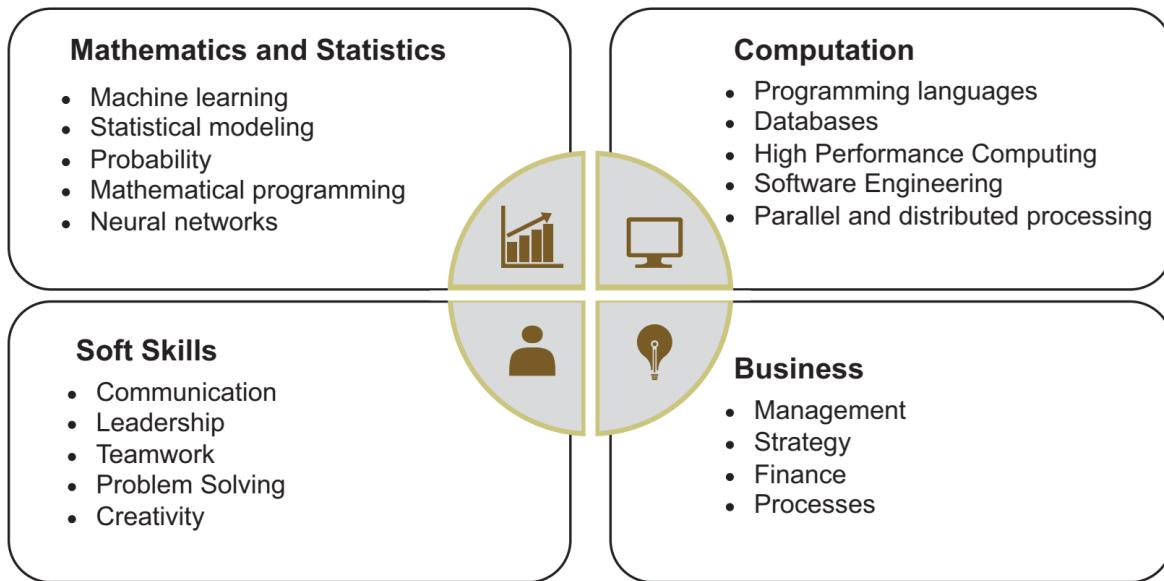
Kỹ năng Mềm (Soft Skills)

1. Kỹ năng Nội tại (Intrapersonal): Khả năng phục hồi (resilience), làm việc dưới áp lực, và trí tuệ cảm xúc.

2. Kỹ năng Tương tác (Interpersonal): Làm việc trong các nhóm đa ngành, yêu cầu kỹ năng giao tiếp, kể chuyện bằng dữ liệu (storytelling), lãnh đạo, làm việc nhóm, giải quyết vấn đề, và sáng tạo.

1.2.2 Các Nghề nghiệp Phổ biến trong Khoa học Dữ liệu

Ngay cả khi chỉ xét đến các kỹ năng cứng, việc tìm kiếm tất cả chúng ở một người vẫn rất khó khăn. Qua nhiều năm, nhiều chuyên ngành khác nhau đã xuất hiện, và vẫn



Hình 1.3: Các kỹ năng cần thiết cho nhà khoa học dữ liệu.

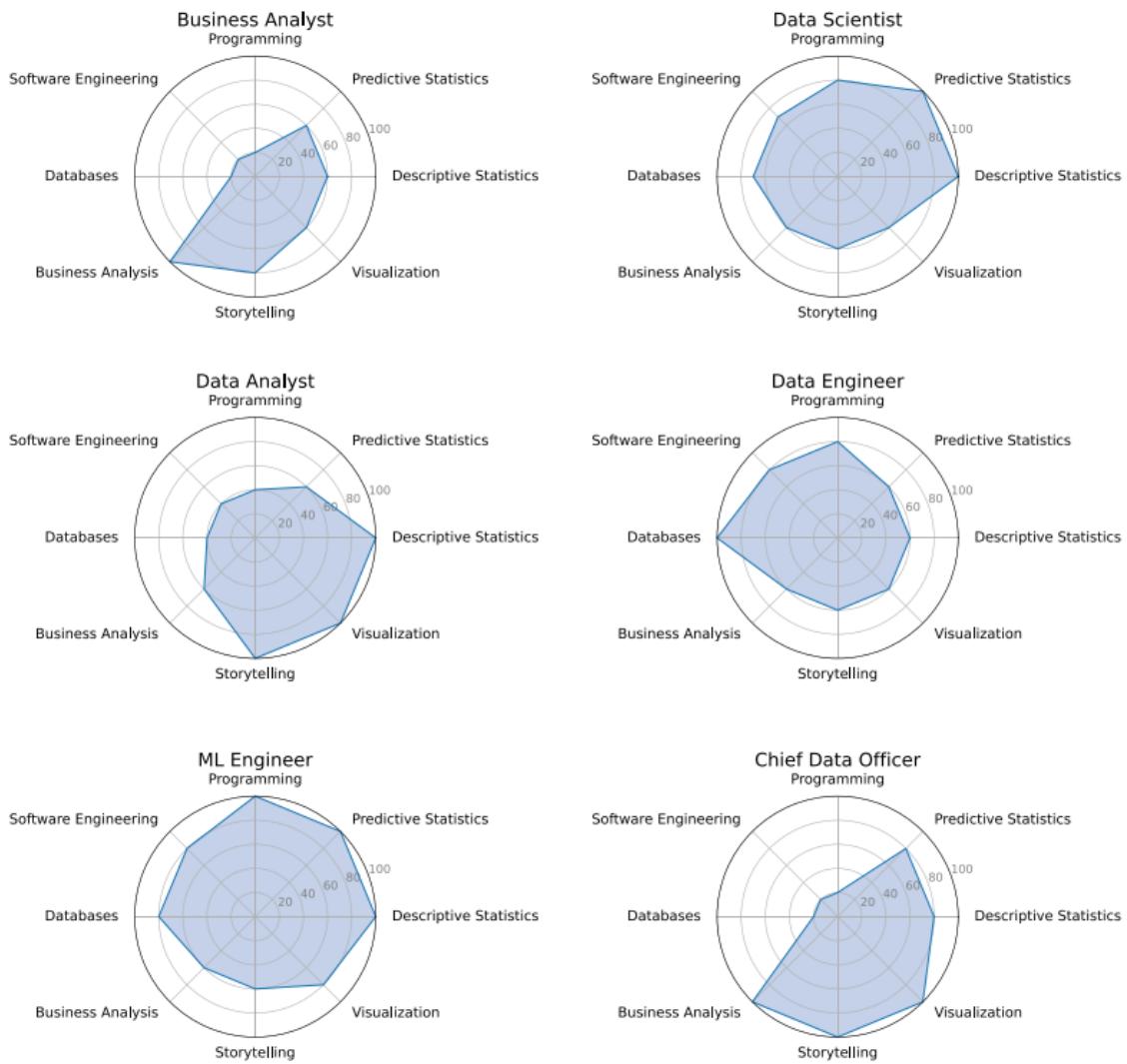
đang tiếp tục xuất hiện, với mục tiêu xây dựng và tổ chức các nhóm khoa học dữ liệu dễ dàng hơn theo các lộ trình nghề nghiệp rõ ràng và bền vững. Cho đến nay, các nghề nghiệp phổ biến nhất trong khoa học dữ liệu là phân tích kinh doanh, khoa học dữ liệu, phân tích dữ liệu, kỹ sư dữ liệu, học máy, Kỹ sư công nghệ thông tin và Giám đốc Dữ liệu (CDO). Những vị trí này sẽ được mô tả trong các phần sau, nhấn mạnh vai trò chính, các kỹ năng cần thiết và bằng cấp đại học điển hình mà các chuyên gia này đạt được.

Nghề nghiệp	Vai trò Chính (Main Roles)	Kỹ năng/Nền tảng Cần thiết
Nhà Phân tích Nghề vụ (Business Analyst)	<ul style="list-style-type: none"> Hiểu vấn đề từ góc độ kinh doanh. Phân tích yêu cầu và chuẩn bị các trường hợp sử dụng. Chuyển đổi tiềm năng phân tích của dữ liệu thành kết quả kinh doanh tiềm năng. Cấu trúc hóa các giải pháp dựa trên dữ liệu. 	Đóng vai trò là giao diện chiến lược giữa kinh doanh và khoa học dữ liệu.
Nhà Khoa học Dữ liệu (Data Scientist)	<ul style="list-style-type: none"> Xác định dữ liệu phù hợp nhất để phân tích. Lập kế hoạch, thiết kế, phát triển và áp dụng các thuật toán khoa học dữ liệu. Phân tích kết quả, dịch các yêu cầu kinh doanh sang phân tích. 	Kiến thức chuyên sâu về Máy tính (lập trình, thuật toán, học máy, học sâu) và Toán học/Thống kê (mô hình thống kê, xác suất). Nền tảng phổ biến: toán học, thống kê, kỹ thuật.
Nhà Phân tích Dữ liệu (Data Analyst)	<ul style="list-style-type: none"> Trích xuất dữ liệu thô từ nhiều nguồn. Áp dụng phần mềm cụ thể để tìm kiếm các thông tin chi tiết có thể hành động được (actionable insights). Tạo ra các hình ảnh trực quan (visualizations) và diễn giải. Cung cấp kết quả thành báo cáo và/hoặc bảng điều khiển (dashboards). 	Sử dụng các công cụ thao tác dữ liệu (như SQL, bảng tính) và các công cụ thiết kế bảng điều khiển (như Power BI, Tableau). Thường là vai trò khởi đầu trong khoa học dữ liệu.

Nghề nghiệp	Vai trò Chính (Main Roles)	Kỹ năng/Nền tảng Cần thiết
Kỹ sư Dữ liệu (Data Engineer)	<ul style="list-style-type: none"> Quản lý quy trình ETL (Extract, Transform, Load). Xây dựng và duy trì đường dẫn dữ liệu (data pipelines). Biến đổi dữ liệu thô thành định dạng có thể sử dụng được. 	Thành thạo hệ thống quản lý dữ liệu, cấu trúc dữ liệu, kỹ thuật phần mềm, và lập trình. Nền tảng điển hình là các lĩnh vực công nghệ như kỹ thuật và CNTT.
Kỹ sư Học máy (ML Engineer)	<ul style="list-style-type: none"> Nghiên cứu thuật toán ML. Thiết kế, phát triển, thử nghiệm, và xác thực các hệ thống học máy. 	Là sự kết hợp giữa kỹ sư phần mềm và nhà khoa học dữ liệu. Nền tảng: toán học, kỹ thuật, CNTT.
Giám đốc Dữ liệu (Chief Data Officer - CDO)	<ul style="list-style-type: none"> Quản trị dữ liệu (Data governance). Xem dữ liệu là tài sản chiến lược. Xác định ưu tiên và quản lý các nhóm dữ liệu. Xây dựng và hướng dẫn văn hóa dữ liệu của công ty. 	Vị trí quản lý lãnh đạo, yêu cầu kỹ năng quản lý và lãnh đạo mạnh mẽ. Nền tảng có thể là kỹ thuật, CNTT, hoặc kinh doanh.

1.3 Quy trình làm việc trong Khoa học dữ liệu

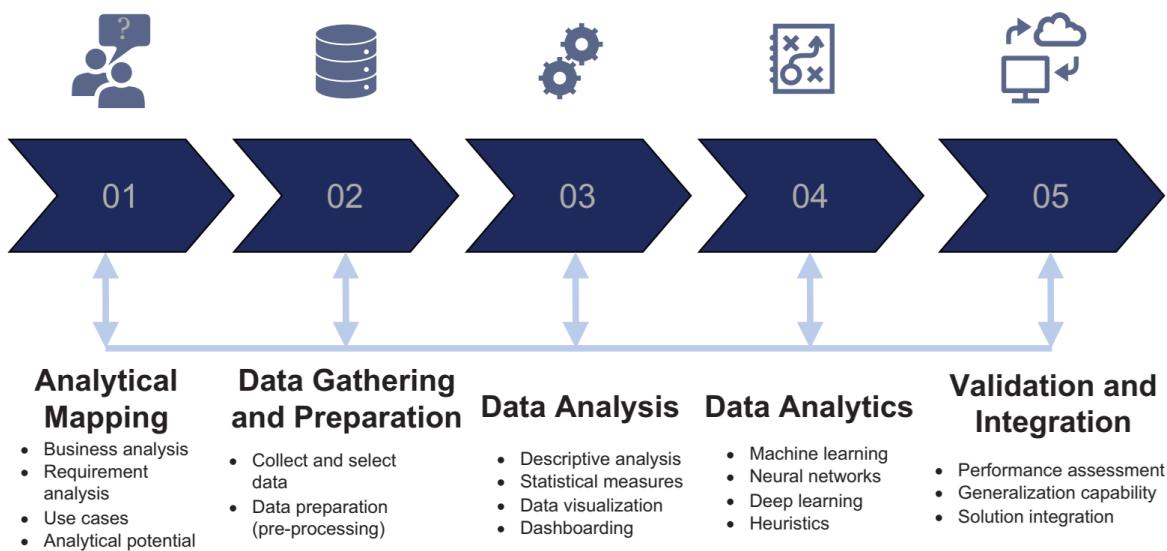
Quy trình làm việc trong Khoa học Dữ liệu (The Data Science Workflow), là chuỗi các bước cần thiết để thực hiện một dự án khoa học dữ liệu từ giai đoạn lập kế hoạch đến



Hình 1.4: Vị trí nghề nghiệp trong khoa học dữ liệu và các kỹ năng.

triển khai sản phẩm. Quy trình này không phải là một chuỗi tuyến tính nghiêm ngặt, mà là một quá trình lặp đi lặp lại (iterative) và tương tác (interactive), cho phép di chuyển tiến hoặc lùi giữa các bước tùy thuộc vào kết quả và quan sát đạt được.

Dưới đây là tóm tắt các bước chính trong Quy trình làm việc Khoa học Dữ liệu (theo Hình ??)



Hình 1.5: Quy trình làm việc trong Khoa học dữ liệu.

Bước 1: Lập bản đồ Phân tích (Analytical Mapping) Đây là bước khởi đầu của quy trình làm việc, thường do Nhà Phân tích Nghiệp vụ (Business Analyst) thực hiện cùng với Giám đốc Dữ liệu (CDO).

- Mục tiêu: Hiểu rõ các quy trình kinh doanh, nhu cầu, mục tiêu chính và dữ liệu sẵn có của doanh nghiệp.
- Hoạt động:
 - Thực hiện phân tích yêu cầu (Requirement analysis), đây là một giai đoạn quan trọng để thu thập, phân tích và ghi lại nhu cầu và kỳ vọng của các bên liên quan.
 - Thiết kế các trường hợp sử dụng (Use cases).
 - Xác định cách dữ liệu có thể được sử dụng để cải thiện kết quả, giảm chi phí và tổn thất.

- Thiết kế các giải pháp sẽ được xây dựng.

Bước này là cần thiết vì nó sẽ hướng dẫn toàn bộ quy trình làm việc còn lại, từ thu thập dữ liệu đến xác thực mô hình.

Bước 2: Thu thập và Chuẩn bị Dữ liệu (Data Gathering and Preparation) Bước này liên quan đến việc xử lý dữ liệu để chúng sẵn sàng cho việc phân tích.

- Hoạt động: Thu thập và lựa chọn dữ liệu.
- Tiền xử lý dữ liệu (Data preparation/pre-processing): Đây là một giai đoạn quan trọng, nơi các phương pháp khác nhau được áp dụng để chuẩn bị dữ liệu cho mục đích phân tích.

Vai trò kỹ sư dữ liệu (Data Engineer) chủ yếu tập trung vào quy trình ETL (Extract, Transform, and Load): Truy cập và cung cấp dữ liệu (Extract), áp dụng các phương pháp chuẩn bị (như chọn biến, chuẩn hóa dữ liệu, xử lý giá trị thiếu) (Transform), và cung cấp dữ liệu đã chuẩn bị vào kho dữ liệu (Load).

Bước 3: Phân tích Dữ liệu (Data Analysis) Đây là chủ đề chính của cuốn sách này. Mặc dù thuật ngữ phân tích dữ liệu có thể có phạm vi rộng hơn, trong bối cảnh EDA, nó bao gồm ba nhiệm vụ cốt lõi:

- Thực hiện phân tích mô tả trên dữ liệu để tóm tắt chúng, khám phá các mẫu, xu hướng, điểm bất thường (anomalies) và kiểm tra giả thuyết.
- Trực quan hóa dữ liệu và các chỉ số chính bằng các biểu diễn đồ họa.
- Tạo các tường thuật xoay quanh dữ liệu (data storytelling) và xây dựng các bảng điều khiển (dashboards).

Các nhiệm vụ này chủ yếu là trách nhiệm của Nhà Phân tích Dữ liệu (Data Analyst), nhưng đôi khi cũng được thực hiện bởi Nhà Khoa học Dữ liệu và/hoặc Kỹ sư Học máy.

Bước 4: Phân tích Dữ liệu Nâng cao (Data Analytics) Bước này liên quan đến việc áp dụng các thuật toán và mô hình phức tạp.

- Hoạt động: Sử dụng Học máy (Machine learning), Mạng thần kinh (Neural networks), Học sâu (Deep learning), và Heuristics.

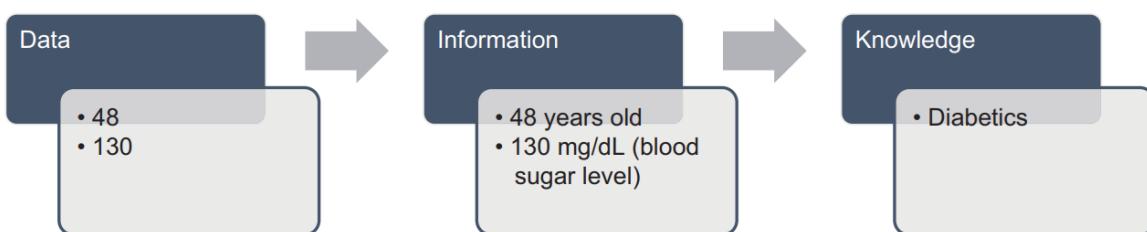
Bước 5: Xác thực và Tích hợp (Validation and Integration) Đây là bước kết thúc quy trình làm việc, đảm bảo chất lượng và khả năng ứng dụng của giải pháp.,

- Hoạt động: Đánh giá hiệu suất, kiểm tra khả năng khái quát hóa, và tích hợp giải pháp.
- Mục đích: Đảm bảo rằng các mục tiêu của dự án được đáp ứng và các mô hình hoạt động hiệu quả

1.4 Dữ liệu

Về bản chất, dữ liệu (data) là mọi thứ có thể được sử dụng, di chuyển, xử lý hoặc dịch để mang một ý nghĩa nào đó. Ví dụ về dữ liệu bao gồm một con số, một từ, một hình ảnh, một văn bản, một biểu đồ, và một âm thanh. Về mặt tính toán, bất cứ thứ gì có thể được lưu trữ và/hoặc xử lý đều là một dạng dữ liệu.

- Data (Dữ liệu): Là bất kỳ thông tin nào xuất hiện một cách độc lập, không có ngữ cảnh.
- Information (Thông tin): Dữ liệu cần có một ngữ cảnh để mang lại ý nghĩa, và lúc đó nó được gọi là thông tin. Ví dụ: số "48" (dữ liệu) trở thành "48 tuổi" (thông tin) khi có ngữ cảnh.
- Knowledge (Tri thức): Là kết quả của việc sử dụng, giải thích và xử lý thông tin để đưa ra quyết định hoặc trích xuất thông tin chi tiết (insights) từ dữ liệu.



Hình 1.6: Từ dữ liệu đến thông tin và đến tri thức.

1.4.1 Dữ liệu dạng bảng

Mặc dù dữ liệu có thể tồn tại ở nhiều định dạng khác nhau (ký tự, tín hiệu, video, âm thanh, v.v.), tài liệu lưu ý rằng dạng số ít của data là datum, và hai thuật ngữ này được sử dụng thay thế cho nhau trong văn bản. Tất cả dữ liệu thường có thể được chia thành hai phần chính:

- Objects (Đối tượng): Còn được gọi là instances, items, observations, hoặc patterns. Mỗi đối tượng đại diện cho một mẫu dữ liệu (datum) với các biến của nó. Ví dụ, một chiếc điện thoại di động là một đối tượng.
- Variables (Biến): Còn được gọi là features, attributes, hoặc characteristics. Đây là các mô tả về đối tượng, ví dụ: dung lượng bộ nhớ, kích thước màn hình, loại bộ xử lý của điện thoại.

Dạng bảng (Tabular Format): Tổ chức điển hình của một bộ dữ liệu có cấu trúc là nơi mỗi hàng tương ứng với một đối tượng và mỗi cột tương ứng với một biến. Đây sẽ là định dạng chuẩn được sử dụng trong cuốn sách này.

Vai trò của Biến trong phân tích Trong một nhiệm vụ phân tích, các biến có thể được chia thành:

- Biến độc lập (Independent variables): Là những biến có thể được thao tác, kiểm soát và đo lường bằng thực nghiệm, và chúng gây ảnh hưởng đến giá trị của biến phụ thuộc.
- Biến phụ thuộc (Dependent variable): Là biến được quan sát hoặc tính toán, bị ảnh hưởng bởi các biến độc lập. Ví dụ, Chỉ số Khối cơ thể (*BMI*) là biến phụ thuộc, trong khi cân nặng (*w*) và chiều cao (*h*) là các biến độc lập.

Dữ liệu dạng từ điển (Data Dictionary) Khi tổ chức một bộ dữ liệu để thực hiện bất kỳ loại phân tích nào, việc xây dựng một từ điển dữ liệu (data dictionary) là hữu ích.

- Từ điển dữ liệu là một cấu trúc mô tả tất cả các biến trong bộ dữ liệu.
- Nó bao gồm định nghĩa (ý nghĩa) của các biến và các thông tin cần thiết khác cho việc phân tích, chẳng hạn như miền giá trị (domain) của chúng.

1.4.2 Phân loại kiểu dữ liệu

Sự đa dạng lớn của dữ liệu cho phép chúng ta phân loại chúng dưới nhiều hình thức khác nhau, tùy thuộc vào đặc điểm, ứng dụng và phương pháp cụ thể. Dữ liệu có thể được phân loại dựa trên: cấu trúc, bản chất, loại, tính biến đổi theo thời gian, số chiều (dimension) và quyền sở hữu (ownership).

Phân loại theo Cấu trúc (Structure) Cấu trúc đề cập đến cách dữ liệu được tổ chức. Dữ liệu có thể được phân loại thành ba loại cấu trúc chính: có cấu trúc, bán cấu trúc, và không cấu trúc.

- Structured Data (Dữ liệu có cấu trúc):
 - Là loại dữ liệu có một mô hình dữ liệu (data model) được xác định rõ ràng, mô tả các đối tượng, mối quan hệ và thuộc tính của chúng.
 - Việc truy cập, lưu trữ và phân tích dữ liệu có cấu trúc dễ dàng hơn so với hai loại còn lại.
 - Dữ liệu này thường được trình bày dưới dạng bảng (table), nơi mỗi hàng là một đối tượng và mỗi cột là một biến (features).

Ví dụ: Một bảng mô tả các thông số của một bộ ô tô, bao gồm giá mua, mức bảo trì, số cửa, v.v., và kết quả cuối cùng cho biết liệu nó có chấp nhận được để mua hay không.

• Semi-structured Data (Dữ liệu bán cấu trúc): Dữ liệu bán cấu trúc (semi-structured data) là loại dữ liệu không tuân theo cấu trúc bảng cố định như dữ liệu có cấu trúc (cơ sở dữ liệu quan hệ), nhưng vẫn có chứa các thẻ, đánh dấu hoặc cấu trúc nhất định giúp nhận diện và tổ chức thông tin.

Ví dụ: File dữ liệu định dạng JSON có kiểu từ điển hoặc định dạng XML

```
json
{
  "ten": "Nguyễn Văn A",
  "tuoi": 25,
  "dia_chi": {
    "thanh_pho": "Hà Nội",
    "quan": "Cầu Giấy"
  },
  "so_thich": [ "đọc sách", "du lịch", "lập trình" ]
}
```

XML

```
<nhanvien>
  <ten>Trần Thị B</ten>
  <tuoi>30</tuoi>
  <ky_nang>Python, Java</ky_nang>
</nhanvien>
```

- Unstructured Data (Dữ liệu không cấu trúc): Dữ liệu không có cấu trúc (unstructured data) là dữ liệu không có mô hình dữ liệu cố định, không sắp xếp theo bảng, hàng, cột hay lược đồ rõ ràng, nên máy tính khó xử lý trực tiếp mà không cần công cụ đặc biệt.

Ví dụ: Một số dạng phổ biến dữ liệu không có cấu trúc

- Văn bản tự do: bài viết, email nội dung thân thư, báo cáo Word/PDF, bình luận mạng xã hội
- Hình ảnh: ảnh chụp, ảnh y tế (X-quang, MRI), ảnh vệ tinh
- Video: video YouTube, camera giám sát, phim
- Âm thanh: file ghi âm cuộc gọi, podcast, nhạc
- Tin nhắn chat: Zalo, Messenger, Telegram (chủ yếu là văn bản tự do + emoji + file đính kèm)
- Trang web/HTML: nội dung bài viết, blog, tin tức (có thẻ HTML nhưng nội dung chính là văn bản tự do)
- File log thô: không có định dạng chuẩn, chỉ là chuỗi text dài

Bảng 1.4: So sánh các loại Dữ liệu

Loại Dữ liệu	Mô hình Dữ liệu	Siêu dữ liệu	Ví dụ
Có cấu trúc	Có	Không	Bảng, bảng tính, CSDL quan hệ
Bán cấu trúc	Không	Có	E-mail, tệp XML, JSON
Không cấu trúc	Không	Không	Hình ảnh, âm thanh, văn bản

1.4.3 Phân loại theo bản chất

Dựa trên bản chất, dữ liệu được chia thành định lượng (Quantitative) và định tính (Qualitative):

- Dữ liệu định lượng: Liên quan đến số lượng, những thứ có thể được đo lường hoặc đếm, và được biểu thị bằng số (ví dụ: khoảng cách, "bao nhiêu").
- Dữ liệu định tính: Được biểu thị bằng tên, ký hiệu, hoặc biến phân loại (ví dụ: "tốt đến mức nào")

Điều quan trọng cần lưu ý là một số giá trị số có thể được coi là biến định tính nếu chúng được sử dụng làm mã định danh duy nhất (ID), chẳng hạn như Mã số An sinh xã hội (SSN). Trong bộ dữ liệu đánh giá ô tô, các biến như "Doors" (Số cửa) và "Persons" (Số người) được coi là biến phân loại mặc dù giá trị của chúng là số.

1.4.4 Phân loại theo loại biến

Tập trung vào loại biến, dữ liệu được chia thành hai loại chính: số (numerical) và phân loại (categorical). Dữ liệu phân loại là dữ liệu định tính, được nhận dạng bằng các ký hiệu hoặc nhãn.

Dữ liệu phân loại được chia thành ba loại con:

- Binary (Nhị phân): Chỉ có thể nhận một trong hai giá trị, ví dụ: 0, 1.
- Nominal (Danh nghĩa): Giá trị có các ký hiệu hoặc nhãn riêng biệt nhưng không có thứ tự cụ thể (ví dụ: tình trạng hôn nhân).
- Ordinal (Thứ bậc): Giá trị có một trật tự cụ thể của các danh mục (ví dụ: high, medium, low), nhưng không nhất thiết phải có khái niệm rõ ràng về khoảng cách giữa các giá trị.

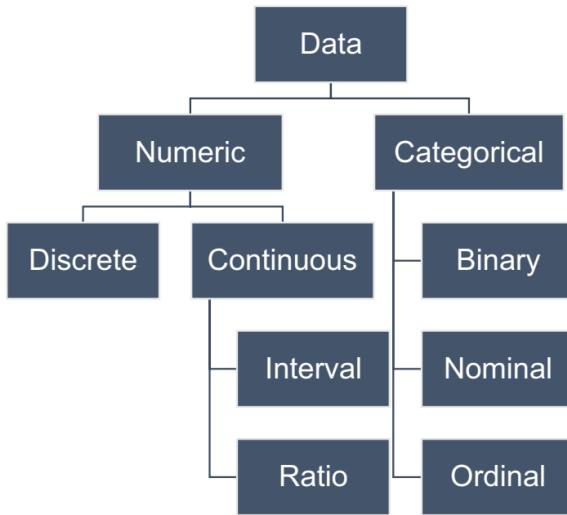
1.4.5 Tính biến đổi theo thời gian (Time Variability)

Tính biến đổi theo thời gian (còn gọi là thang thời gian hoặc tính chất tĩnh/stationary nature) cho phép dữ liệu được phân loại thành hai loại chính: tĩnh và động.

Dữ liệu Tĩnh (Static Data) Dữ liệu tĩnh là dữ liệu cố định theo nghĩa là chúng không thay đổi theo thời gian.

Ví dụ: Các biến liên quan đến một số loại đá, chẳng hạn như trọng lượng, kích thước, màu sắc của chúng, về nguyên tắc là tĩnh vì các giá trị này không thay đổi theo thời gian.

Dữ liệu tĩnh truyền thống đôi khi được gọi là dữ liệu theo lô (batch data), nghĩa là toàn bộ bộ dữ liệu được lưu trữ và sẵn có để phân tích.



Hình 1.7: Phân loại theo biến của dữ liệu.

Dữ liệu Động (Dynamic Data) Dữ liệu động (còn gọi là non-stationary) là những dữ liệu được cập nhật định kỳ, tức là thay đổi theo thời gian.

Ví dụ Giá trị cổ phiếu trên thị trường chứng khoán là một biến động (time-varying variable), vì chúng cực kỳ dễ biến động và thay đổi liên tục tùy thuộc vào áp lực mua và bán trong giờ hoạt động của thị trường.

Mặc dù có nhiều loại dữ liệu động khác nhau, nhưng trọng tâm của tài liệu này là hai loại dữ liệu biến đổi theo thời gian đặc biệt quan trọng cho phân tích dữ liệu:

a. Dữ liệu Chuỗi thời gian (Time-series data - TS)

- **Khái niệm:** Chuỗi thời gian là tập hợp các quan sát được lập chỉ mục hoặc thu được từ các phép đo tuần tự được thực hiện tại các thời điểm cách đều nhau và tuân theo một tốc độ lấy mẫu nhất định.
- **Đặc điểm:** Khi được vẽ trên biểu đồ, chuỗi thời gian luôn có một trong các trục (thường là trục x) đại diện cho thời gian.
- **Vai trò của thời gian:** Phân tích chuỗi thời gian luôn phải coi thời gian là một biến quan trọng trong phân tích, nhằm mục đích hiển thị cách một hoặc nhiều biến biến đổi (hành xử) theo thời gian.
- **Ví dụ:** Giá cổ phiếu theo thời gian (luôn bị ảnh hưởng bởi giá trước đó), nhịp tim, nhiệt độ, và lâng suât.

b. Luồng Dữ liệu (Data Streams)

- Khái niệm: Luồng dữ liệu có thể được định nghĩa là chuỗi (có tiềm năng) không giới hạn của các đối tượng dữ liệu được tạo ra liên tục.
- Vai trò của thời gian: Luồng dữ liệu là bất kỳ dòng dữ liệu nào, và thang thời gian hoặc khoảng thời gian không có liên quan lớn.
- Ví dụ: Luồng video và âm thanh, nhật ký web, lưu lượng mạng máy tính, tin nhắn mạng xã hội, và các chủ đề thịnh hành (trending topics - TTs) trên mạng xã hội. Sự khác biệt cốt lõi là ở vai trò của thời gian: Luồng dữ liệu có thể không được phân tích như một chuỗi thời gian nếu thời gian không phải là một đặc trưng quan trọng. Ví dụ, các chủ đề thịnh hành trên X (Twitter cũ) thay đổi liên tục, nhưng thang thời gian thường không ảnh hưởng hoặc không liên quan đến việc phân tích các TTs mới.

Tóm lại: Dữ liệu được phân loại thành tĩnh (cố định) và động (thay đổi). Trong dữ liệu động, Chuỗi thời gian là dữ liệu có sự phụ thuộc không thể tránh khỏi vào thời gian, trong khi Luồng dữ liệu là dòng dữ liệu liên tục nhưng sự phụ thuộc vào thời gian có thể không quan trọng.

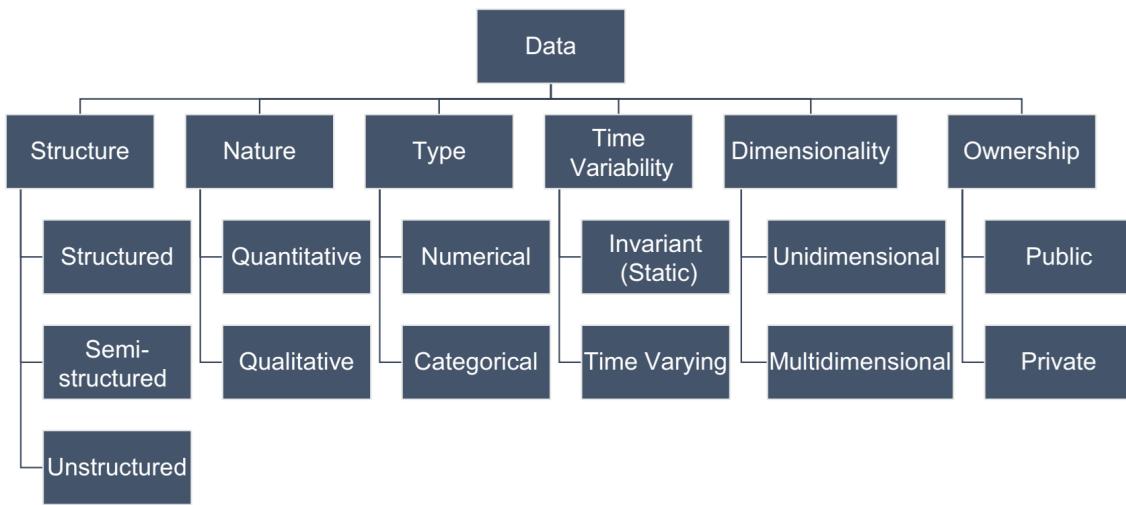
1.4.6 Phân loại theo số chiều (Dimensionality)

Số chiều của một bộ dữ liệu được xác định bằng số lượng các biến hoặc thuộc tính (attributes) có trong tập dữ liệu. Các biến có giá trị duy nhất cho mỗi đối tượng (chẳng hạn như ID xe hơi hoặc tên người) được sử dụng để nhận dạng và thường không được tính vào phân tích. Các biến được sử dụng làm nhãn để xác định một nhóm hoặc lớp đối tượng cụ thể (ví dụ: biến "Class" trong Table 2.1) thường được phân tích riêng và không được tính là một chiều của dữ liệu

1.4.7 Phân loại theo quyền sở hữu (Ownership)

Dữ liệu có thể được phân loại là riêng tư (private) hoặc công khai (public):

- Dữ liệu Riêng tư: Là dữ liệu thuộc về và được tạo ra bởi cá nhân hoặc doanh nghiệp của bạn.
- Dữ liệu Công khai: Là dữ liệu được tạo ra bởi bên thứ ba nhưng có giá trị hoặc liên quan đến bạn hoặc doanh nghiệp của bạn, chẳng hạn như dữ liệu trực tuyến, bao gồm dữ liệu mạng xã hội.



Hình 1.8: Phân loại các dạng dữ liệu.

1.5 Chuẩn bị dữ liệu

Quá trình làm sạch và xử lý dữ liệu thô (raw data) để chúng có thể được sử dụng hiệu quả trong phân tích. Dữ liệu thô là dữ liệu nguồn hoặc dữ liệu sơ cấp được nhập ban đầu vào cơ sở dữ liệu bởi người vận hành, cảm biến hoặc bất kỳ thiết bị nào. Tổng quan và Các vấn đề của dữ liệu thô.

Dữ liệu thô thường chứa một số vấn đề khiến việc phân tích trở nên khó khăn và dễ mắc lỗi và hiểu sai. Các vấn đề chính bao gồm:

- Data Overload (Quá tải dữ liệu): Số lượng đối tượng hoặc biến quá mức.
- Incompleteness (Không đầy đủ): Thiếu đối tượng, giá trị, hoặc biến.
- Inconsistency (Không nhất quán): Vi phạm miền giá trị (domain violations) và sự không khớp.
- Noise (Nhiều).

Quá trình chuẩn bị dữ liệu là cần thiết để đảm bảo dữ liệu phù hợp cho việc khám phá hoặc thiết kế giải pháp cho các vấn đề đang xét.

Các kỹ thuật chuẩn bị dữ liệu quan trọng:

1.5.1 Lấy Mẫu (Sampling)

Lấy mẫu là một bước quan trọng trong chuẩn bị dữ liệu. Lấy mẫu bao gồm các phương pháp khác nhau, ví dụ như được minh họa bằng Bộ dữ liệu Mammographic.

1.5.2 Xử lý Giá trị Thiếu (Missing Values)

Một tập dữ liệu được coi là không đầy đủ (incomplete) khi có các đối tượng, biến hoặc giá trị bị thiếu. Để xác định một đối tượng bị thiếu, cần phải nhận biết sự tồn tại của nó. Có một số cách để xử lý giá trị bị thiếu, bao gồm:

- **Bỏ qua Đối tượng (Ignore the Object):** Cách đơn giản và phổ biến nhất là loại bỏ đối tượng có giá trị bị thiếu khỏi tập dữ liệu. Tuy nhiên, cần lưu ý rằng việc loại bỏ một đối tượng đồng nghĩa với việc loại bỏ thông tin có sẵn trong tất cả các biến khác của đối tượng đó. Nếu chỉ thực hiện phân tích mô tả trên một biến duy nhất, việc này có thể không gây ảnh hưởng lớn.
- **Điền giá trị bằng hằng số toàn cục (Inputting using a global constant).**
- **Điền giá trị dựa trên sự tương đồng của đối tượng (Inputting based on object similarity).**
- **Sử dụng số đo xu hướng trung tâm của Biến (Central Tendency Measure of the Variable):** Phương pháp phổ biến này tính toán một trong các số đo xu hướng trung tâm của biến (ví dụ: trung bình, mode, trung vị) và sử dụng giá trị này để thay thế giá trị bị thiếu.
- **Sử dụng số đo xu hướng trung tâm của Biến Lớp (Central Tendency Measure of the Class Variable):** Phương pháp này tương tự như trên, nhưng sử dụng nhãn lớp hoặc nhóm của đối tượng (nếu có) để tính toán số đo xu hướng trung tâm của riêng lớp đó, sau đó thay thế giá trị bị thiếu.

1.5.3 Chuẩn hóa (Normalization)

Mục đích của chuẩn hóa là tiêu chuẩn hóa các phạm vi của biến.

- **Chuẩn hóa Min-Max:** Đây là phương pháp chuẩn hóa đặt các giá trị của biến vào một phạm vi giữa một giá trị tối thiểu (min) và tối đa (max).
- **Chuẩn hóa Z-Score (z-score):** Là một phương pháp chuẩn hóa khác.

Chuẩn bị dữ liệu, bao gồm lấy mẫu, xử lý giá trị bị thiếu và chuẩn hóa, là bước đóng góp vào nền tảng của quá trình phân tích dữ liệu và ra quyết định thành công

1.6 Dữ liệu

Một số bộ dữ liệu dùng trong bài giảng.

1.6.1 Forest Fires Dataset (Bộ dữ liệu cháy rừng)

Bộ dữ liệu **Forest Fires** được sử dụng rộng rãi để minh họa các khái niệm phân tích mô tả như hình dạng phân phối, các biện pháp xu hướng trung tâm và biến thiên.

Các biến chính:

- Fine Fuel Moisture Code (FFMC) (khoảng [18.7, 96.20]),
- Duff Moisture Code (DMC) (khoảng [1.1, 291.3]),
- Drought Code (DC),
- chỉ số lan truyền ban đầu (ISI),
- nhiệt độ (temp),
- độ ẩm (RH),
- gió (wind),
- lượng mưa (rain),
- bao gồm các biến phân loại như tháng (month) và ngày (day).

Câu hỏi mẫu và giả thuyết về tập dữ liệu cháy rừng:

Câu hỏi: Những yếu tố chính nào ảnh hưởng đến sự xuất hiện và lan rộng của cháy rừng?

Giả thuyết: Nhiệt độ cao hơn và độ ẩm thấp hơn góp phần làm tăng các vụ cháy rừng.

1.6.2 Mammographic Mass Dataset (Bộ dữ liệu khối u tuyến vú)

Dữ liệu này được sử dụng để thí nghiệm với các hệ số tương quan cho dữ liệu phân loại và có sẵn qua kho lưu trữ UCI. Bộ dữ liệu này liên quan đến chụp nhũ ảnh, là hình ảnh X-quang vú được sử dụng để sàng lọc ung thư vú.

Các biến chính:

- BI-RADS,
- Age (Tuổi),
- Shape (Hình dạng),
- Margin (Rìa),
- Density (Mật độ), và
- Severity (Mức độ nghiêm trọng).

Câu hỏi mẫu và giả thuyết về tập dữ liệu chụp nhũ ảnh:

Câu hỏi: Những đặc điểm quan trọng nào liên quan đến khả năng mắc ung thư vú? ung thư dựa trên dữ liệu chụp nhũ ảnh?

Giả thuyết: Các mẫu dày đặc trong chụp nhũ ảnh có mối tương quan tích cực với sự gia tăng nguy cơ ung thư vú.

1.6.3 Gapminder Dataset (Bộ dữ liệu Gapminder)

Dữ liệu quốc gia trên toàn thế giới, kéo dài từ năm 1998 đến năm 2018, với 3,675 đối tượng (objects).

Các biến chính:

- Bao gồm country (quốc gia),
- continent (châu lục),
- year (năm),
- life_exp (tuổi thọ),
- hdi_index (chỉ số phát triển con người - HDI),

- co2_consump (lượng khí thải CO₂ trên mỗi người),
- gdp (Tổng sản phẩm quốc nội trên mỗi người), và
- services (tỷ lệ phần trăm người lao động trong lĩnh vực dịch vụ).

1.6.4 Daily Delhi Climate

Được sử dụng để minh họa các khái niệm phân tích chuỗi thời gian (Time Series), bao gồm phân tích mô tả, trung bình trượt và phân rã chuỗi.

1.6.5 Auto-mpg dataset

Dữ liệu có cấu trúc điển hình nơi mỗi hàng là một đối tượng và mỗi cột là một biến.

1.6.6 IMDb Movie Reviews Corpus (Tập dữ liệu Đánh giá Phim IMDb)

Đây là tập dữ liệu văn bản/tài liệu (text/document data) có sẵn trong bộ công cụ NLTK (Natural Language Toolkit). Chứa các bài đánh giá phim, thường được phân loại là tiêu cực (negative) hoặc tích cực (positive).

1.6.7 Zachary's Karate Club Dataset (Bộ dữ liệu Câu lạc bộ Karate của Zachary)

Đây là bộ dữ liệu mạng (Network). Mô tả một mạng xã hội, thường được biểu diễn bằng ma trận kề (Adjacency Matrix). Được sử dụng để minh họa cấu trúc và trực quan hóa đồ thị.

1.7 Bài tập

Chủ đề và câu hỏi nghiên cứu

Bài 1.1. Bạn vừa được một công ty (hoặc quyết định thành lập một công ty khởi nghiệp) tuyển dụng vào làm việc trên mạng xã hội. Phân tích dữ liệu truyền thông liên quan đến TV. Vấn đề bạn cần giải quyết là:

Ngày nay mọi người sử dụng phương tiện truyền thông xã hội để nói về TV và tương tác với các diễn viên, nhà làm phim, TV các trạm, v.v. Tương tác này có thể được định

lượng và chuyển đổi thành chỉ số khán giả, và cũng đủ điều kiện để tất cả các bên liên quan biết người tiêu dùng nghĩ gì về chương trình, diễn viên, sản phẩm, dịch vụ, v.v. Việc định lượng và định tính dữ liệu truyền thông xã hội liên quan đến TV tạo ra cái mà bây giờ được gọi là Phân tích TV Xã hội. Nhiệm vụ của bạn là thiết kế bảng điều khiển phân tích để định lượng và đánh giá chất lượng truyền hình xã hội dữ liệu. Thảo luận theo nhóm về những gì bạn sẽ đưa vào bảng thông tin này và lý do.

Bài 1.2. Thảo luận về các nguyên tắc và triết lý cơ bản hướng dẫn lĩnh vực khoa học dữ liệu. Một số cân nhắc về mặt đạo đức trong khoa học dữ liệu là gì?

Bài 1.3. Nghiên cứu sự hiện diện của thiên kiến trong dữ liệu và cách nó có thể ảnh hưởng đến việc ra quyết định trong các lĩnh vực khác nhau. Thảo luận về những thách thức trong việc giảm thiểu thiên kiến trong thuật toán và những tác động về mặt đạo đức của dữ liệu thiên kiến.

Bài 1.4. Xem xét những thách thức và tầm quan trọng của việc tạo ra các hệ thống AI minh bạch và dễ hiểu. Thảo luận về sự đánh đổi giữa tính phức tạp và khả năng diễn giải trong các mô hình học máy, cũng như ý nghĩa của chúng đối với niềm tin và trách nhiệm giải trình.

Bài 1.5. Mục 1.2 giới thiệu các nghề nghiệp chính trong khoa học dữ liệu, nhấn mạnh các hoạt động chính, kiến thức cần thiết và nền tảng của chúng. Mô tả ít nhất một nghề nghiệp khác trong khoa học dữ liệu chưa được đề cập trong tài liệu và nhấn mạnh ba khía cạnh tương tự được sử dụng trong tài liệu.

Bài 1.6. Phần 1.4 trình bày tổng quan về một số cột mốc trong lịch sử phát triển của AI. Làm việc theo nhóm, sinh viên vẽ một dòng thời gian tóm tắt về những phát triển chính trong khoa học dữ liệu mà bạn cho là quan trọng.

Bài 1.7. Khoa học dữ liệu là việc trích xuất giá trị từ dữ liệu, bằng cách rút ra những hiểu biết sâu sắc, xây dựng các công cụ hỗ trợ quyết định, hoặc tự động hóa các hệ thống và quy trình. Theo thời gian, rất nhiều kho dữ liệu đã được phát triển, không chỉ để chia sẻ dữ liệu cho mục đích thử nghiệm hoặc cạnh tranh, như được cung cấp bởi Kho dữ liệu Học máy UCI và Kaggle, mà còn để cung cấp thông tin một cách minh bạch cho cộng đồng, như trường hợp của dữ liệu chính phủ. Hãy cung cấp danh sách ba kho dữ liệu trực tuyến có thể được sử dụng để phân tích dữ liệu và giải thích lý do tại sao các kho này- các ries đã được chọn.

Bài 1.8. Thảo luận về các cân nhắc đạo đức liên quan đến quyền sở hữu dữ liệu. Ai nên sở hữu dữ liệu, đặc biệt là khi dữ liệu được thu thập từ công chúng? Chủ sở hữu dữ liệu có trách nhiệm gì?

Khám phá các khía cạnh triết học của việc lấy mẫu trong quá trình chuẩn bị dữ liệu. Thảo luận về vai trò của việc lấy mẫu trong việc định hình nhận thức của chúng ta về tổng thể, khả năng sai lệch và ý nghĩa triết học của việc khai quát hóa từ một tập hợp con lên toàn bộ quần thể.

Bài 1.9. Dữ liệu thị trường chứng khoán có thể là chuỗi thời gian hoặc luồng dữ liệu, và sự khác biệt nằm ở cách dữ liệu được quan sát và xử lý. Giải thích khi nào dữ liệu thị trường chứng khoán được xem là chuỗi thời gian và khi nào được xem là luồng dữ liệu.

Bài 1.10. Chuẩn bị một từ điển dữ liệu cho các tập dữ liệu sau: Chụp nhũ ảnh, Cháy rừng, Auto MPG, Gapminder và NaturalEarth_lowres.

Bài 1.11. Bản chất của việc chuẩn bị dữ liệu, bao gồm lấy mẫu, xử lý giá trị bị thiếu và chuẩn hóa, ảnh hưởng như thế nào đến độ tin cậy và độ chính xác của thông tin chi tiết thu được từ phân tích dữ liệu?

Bài 1.12. Các khía cạnh khác nhau của việc mô tả và chuẩn bị dữ liệu được đề cập trong chương này đóng góp như thế nào vào nền tảng của quá trình phân tích dữ liệu và ra quyết định thành công?

Bài 1.13. Chọn hai tập dữ liệu từ sách (Mục 2.4) và phân loại dữ liệu dựa trên các khía cạnh được đề cập trong chương: cấu trúc, bản chất, loại, biến thiên theo thời gian, chiều và quyền sở hữu. Giải thích cách phân loại của bạn.

Bài tập tính toán

Bài 1.14. Đối với tập dữ liệu Gapminder, hãy sử dụng thư viện Pandas để xác định và đếm các giá trị bị thiếu trong biến "gdp". Triển khai hai hoặc nhiều kỹ thuật tính toán giá trị bị thiếu đã được thảo luận trong chương và so sánh tác động của chúng lên các số liệu của tập dữ liệu.

Bài 1.15. Chuẩn hóa tất cả bốn thuộc tính của tập dữ liệu Iris bằng các phương pháp được trình bày trong chương và so sánh phạm vi chuẩn hóa của từng phương pháp.

Bài 1.16. Chọn một tập dữ liệu và mô phỏng quy trình chuẩn bị dữ liệu. Áp dụng lấy mẫu, xử lý các giá trị bị thiếu và thực hiện chuẩn hóa trên tập dữ liệu. Ghi lại từng bước và giải thích cách chúng đóng góp vào quy trình phân tích.

Chương 2

PHÂN TÍCH MÔ TẢ

Phân tích Mô tả (Descriptive Analysis) là một trong những giai đoạn phân tích quan trọng nhất trong Khoa học dữ liệu. Mục tiêu chính của chương này là cung cấp các giá trị số nhằm hiểu rõ cấu trúc bộ dữ liệu, các biến của nó và đặc điểm hóa của chúng. Thông tin này cho phép tóm tắt dữ liệu và trích xuất những hiểu biết có giá trị để hỗ trợ quá trình ra quyết định.

2.1 Phân phối

2.1.1 Phân phối tần số và tần suất

Phân phối tần suất là một danh sách chỉ rõ tất cả các giá trị, loại (categories) hoặc khoảng (intervals) có thể có của một biến nhất định, và đồng thời định lượng tần suất (frequency) của chúng (tức là số lần mỗi giá trị xảy ra).

Phân phối tần suất là một công cụ hữu ích để xác định phạm vi giá trị của một biến và để xây dựng các biểu đồ giúp chúng ta hiểu và đặc trưng hóa dữ liệu.

Phân phối tần suất thường được hình dung bằng cách sử dụng bảng tần suất (frequency table) hoặc bằng cách vẽ các biểu đồ cụ thể, tùy thuộc vào loại biến. Hình dung Phân phối theo Loại Biến:

- Biến liên tục (Continuous variables): Phân phối tần suất của chúng có thể được quan sát bằng cách sử dụng biểu đồ histogram.
- Biến danh nghĩa (Nominal variables): Phân phối tần suất của chúng thường được quan sát bằng cách sử dụng biểu đồ tròn (pie chart) hoặc biểu đồ cột (bar chart).

2.1.2 Bảng Tần suất (Frequency Tables)

Bảng tần suất tóm tắt các giá trị của biến phân loại (categorical variables) và bao gồm ba loại tần suất chính:

- Tần suất Tuyệt đối (Absolute Frequency): Là số lần đếm (count) của mỗi giá trị.
- Tần suất Tương đối (Relative Frequency): Được tính bằng cách chia tần suất tuyệt đối cho tổng số đối tượng (mẫu) và thường được biểu thị bằng phần trăm (sẽ cộng lại thành 100% nếu tính theo phần trăm).
- Tần suất Tích lũy (Cumulative Frequency): Được tính bằng cách cộng lặp đi lặp lại giá trị tần suất hiện tại với giá trị trước đó.

Ví dụ 2.1. Ví dụ Minh họa: • Tài liệu minh họa bằng cách sử dụng biến "Shape" (Hình dạng) từ Bộ dữ liệu Khối u Tuyển vú (Mammographic Dataset).

- Biến "Shape" có thể nhận các giá trị như "Irregular", "Round", "Oval", và "Lobular", đồng thời có cả giá trị bị thiếu được biểu thị bằng dấu chấm hỏi "?". Các giá trị bị thiếu này cũng được bao gồm trong bảng tần suất.
- Mục này cũng đề cập đến một tập lệnh Python (Code 3.2) để tính toán và in bảng tần suất cho một biến số (numeric variable), cho phép thiết lập số lượng bins (thanh) cũng như giới hạn dưới và trên của biểu đồ histogram.

```
[1]: # Determining the frequency distribution, frequency table and
      # pie chart
      # of variable 'Shape' in the Mammographic dataset

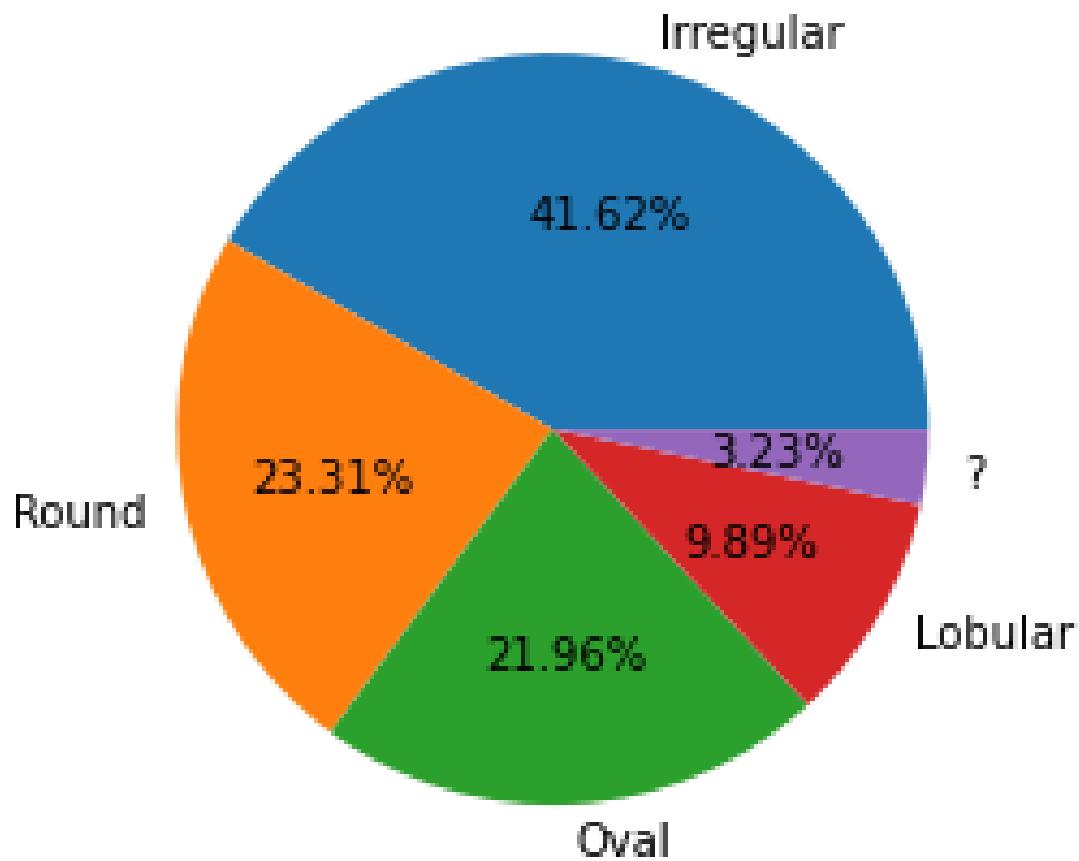
import pandas as pd
import matplotlib.pyplot as plt

# Loading dataset1
# https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass
dmammo = pd.read_csv('mammographic_masses_nominal.csv')

SShape = pd.Series(dammo['Shape'])
ftable = SShape.value_counts() # Generate the frequency table
rftable = ftable/len(SShape)*100 # Relative frequency
cftable = ftable.cumsum()/len(SShape)*100 # Cumulative
      #frequency
df = pd.DataFrame({'Frequency':ftable.to_list(),
                   'Relative Frequency':rftable.to_list(),
```

```
'Cumulative Frequency': cftable.to_list()}  
print(df)  
fig, figtable = plt.subplots()  
figtable.pie(ftable.to_list(), labels=ftable.index.to_list(),  
             autopct='%.2f%%') # From Matplotlib
```

	Frequency	Relative Frequency	Cumulative Frequency
0	400	41.623309	41.623309
1	224	23.309053	64.932362
2	211	21.956296	86.888658
3	95	9.885536	96.774194
4	31	3.225806	100.000000



Hình 2.1: Biểu đồ hình tròn.

2.1.3 Hình dạng Phân phối (Shapes of Distributions)

Phân phối có thể có các hình dạng khác nhau, cho biết phạm vi và mẫu phân phối của dữ liệu.

- Histogram: Biểu đồ histogram đặc biệt quan trọng để phân tích phân phối dữ liệu vì chúng cho phép trực quan hóa mẫu phân phối tổng thể, hình dạng, trung tâm và độ phân tán (spread).
- Giá trị ngoại lệ (Outliers): Histogram cũng cho phép xác định các giá trị ngoại lệ, là những giá trị hoặc phạm vi hiếm khi xảy ra hoặc khác biệt đáng kể so với những giá trị khác.
- Đường cong Mật độ (Density Curve): Thông thường, mẫu của một biến có nhiều giá trị được hiển thị bằng một đường cong trơn gọi là đường cong mật độ, vai trò của nó là mô tả mẫu tổng thể của phân phối. Đường cong mật độ được tạo ra bằng ước tính mật độ hạt nhân (kernel density estimation).
- Đặc tính của Đường cong Mật độ: Một đường cong mật độ liên quan đến một biến định lượng thường có ba thuộc tính: (i) nó là giá trị không âm; (ii) diện tích dưới đường cong là 1; và (iii) diện tích dưới đường cong giữa hai giá trị đại diện cho tỷ lệ các quan sát rơi vào phạm vi đó.
- Độ phân tán (Spread): Việc quan sát độ phân tán là quan trọng.

Ví dụ 2.2. Ví dụ Minh họa: Ví dụ, các biến "RH" và "DMC" (từ Bộ dữ liệu Cháy rừng) có độ phân tán lớn hơn các biến "FFMC" và "ISI", điều này cho thấy sự xuất hiện của "FFMC" và "ISI" tập trung hơn xung quanh các giá trị cụ thể. • Mục 3.2 sẽ trình bày các phép đo tóm tắt giúp xác định hình dạng phân phối bằng số, bao gồm trung bình, độ biến thiên (độ phân tán), sự hiện diện của giá trị ngoại lệ và mức độ dẹt hoặc nhọn.

```
[3] : # Determining the frequency distribution, frequency table and
      # histogram
      # of continuous variables in the Forest Fire dataset

      import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      import seaborn as sns

      # Loading dataset2
      # https://archive.ics.uci.edu/ml/datasets/forest+fires
```

```

dforest = pd.read_csv('forestfires.csv')

var = 'temp' # Choose the target variable
SShape = pd.Series(dforest[var])
nbins = 10
inflimit = 0; suplimit = max(SShape)
ampl = (suplimit - inflimit)/nbins

# Define the range of the variable and bin size
fbins = np.arange(0,suplimit+ampl,ampl)

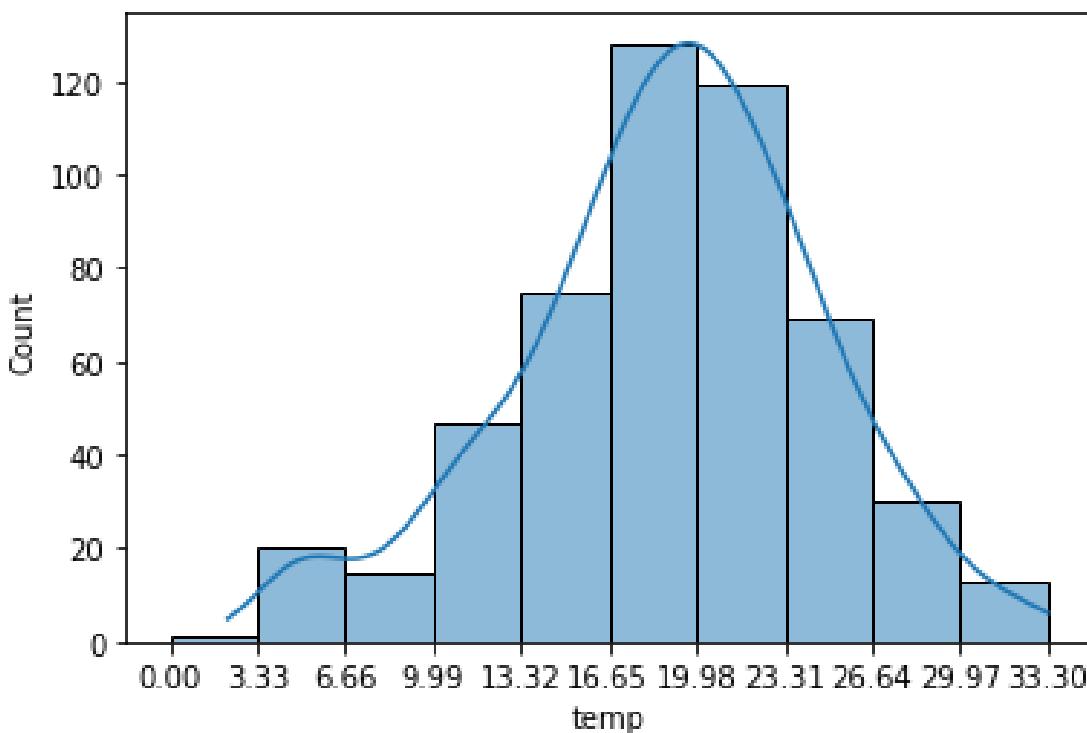
# The pandas.cut function groups the data into bins and counts
# the frequency
ftable = pd.cut(SShape,fbins).value_counts() # Absolute
# frequency
rftable = ftable/len(SShape)*100 # Relative frequency
cftable = ftable.cumsum()/len(SShape)*100 # Cumulative
# frequency
df = pd.DataFrame({'Bins':ftable.index.to_list(),
                   'Frequency':ftable.to_list(),
                   'Relative Frequency':rftable.to_list(),
                   'Cumulative Frequency':cftable.to_list()})
print(df)
plt.xticks(fbins)
sns.histplot(dforest,x=var,bins=fbins, kde = 2)

```

	Bins	Frequency	Relative Frequency	Cumulative Frequency
0	(16.65, 19.98]	128	24.758221	24.758221
1	(19.98, 23.31]	119	23.017408	47.775629
2	(13.32, 16.65]	75	14.506770	62.282398
3	(23.31, 26.64]	69	13.346228	75.628627
4	(9.99, 13.32]	47	9.090909	84.719536
5	(26.64, 29.97]	30	5.802708	90.522244
6	(3.33, 6.66]	20	3.868472	94.390716
7	(6.66, 9.99]	15	2.901354	97.292070
8	(29.97, 33.3]	13	2.514507	99.806576
9	(0.0, 3.33]	1	0.193424	100.000000

2.1.4 Bảng Ngẫu nhiên (Contingency Tables)

Bảng ngẫu nhiên là một công cụ hữu ích để tóm tắt mối quan hệ giữa hai biến phân loại. Nó còn được gọi là bảng tần suất hai chiều (two-way frequency table) hoặc cross-



Hình 2.2: Biểu đồ phân phối.

tabulation. Mỗi quan hệ giữa phân phối tần suất của hai biến được trình bày trong một bảng, với một biến ở hàng và biến kia ở cột.

Ví dụ 2.3. Ví dụ: Phân tích cặp biến "Shape" (Hình dạng) và "Severity" (Mức độ nghiêm trọng) của Bộ dữ liệu Khối u Tuyến vú (Mammographic Dataset). Kết quả cho thấy hình dạng khối u "Irregular" (Không đều) là một chỉ báo thường xuyên của khối u ác tính (malignant tumor)

```
[8]: # Generate Contingency Tables for the Mammographic Dataset

import pandas as pd

url = "https://archive.ics.uci.edu/ml/
       -machine-learning-databases/mammographic-masses/
       -mammographic_masses.data"
cols = ['BI-RADS', 'Age', 'Shape', 'Margin', 'Density',
        'Severity']
dmammo = pd.read_csv(url, names=cols, na_values='?')

# Remove rows with missing values
```

```
dmammo.dropna(inplace=True)

# Print the contingency tables
var = ['Shape', 'Margin', 'Density']
print('**Contingency Tables**')
for i in var:
    CT = pd.crosstab(dammo[i], dammo['Severity'])
    print('Variables', i, 'and Severity:\n', CT)
```

```
**Contingency Tables**
Variables Shape and Severity:
  Severity      0      1
Shape
  1.0        158     32
  2.0        149     31
  3.0        39      42
  4.0        81     298
Variables Margin and Severity:
  Severity      0      1
Margin
  1.0        282     38
  2.0         8      15
  3.0        39      67
  4.0        77     177
  5.0        21     106
Variables Density and Severity:
  Severity      0      1
Density
  1.0          6      5
  2.0        38     18
  3.0       379     376
  4.0          4      4
```

2.2 Các số đo đặc trưng

2.2.1 Số đo xu hướng trung tâm

Các phép đo này xác định giá trị tiêu biểu hoặc trung bình của phân phối.

- **Trung bình (Mean) (μ hoặc \bar{x}):** Tổng của tất cả các giá trị chia cho số lượng giá trị. Trung bình **rất nhạy cảm** với các giá trị ngoại lai (outliers).

$$\text{Trung bình mẫu: } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \text{Trung bình tập chính: } \mu = \frac{1}{N} \sum_{i=1}^N x_i. \quad (2.1)$$

- **Trung vị (Median):** Giá trị nằm ở giữa khi dữ liệu được sắp xếp. Trung vị **ít bị ảnh hưởng bởi outliers** hơn trung bình và là một phép đo mạnh hơn].
- **Mode:** Giá trị xuất hiện lớn nhất.
- **Midpoint:** Giá trị nằm giữa giá trị lớn nhất (x_L) và nhỏ nhất (x_l) của phân phối, tính bằng $\text{midpoint} = \frac{x_L + x_l}{2}$.
- **Trung bình với bảng tần suất:** Giá trị mỗi khoảng thay bởi điểm giữa x_i và nhân với giá trị tần suất tương ứng f_i .

$$\bar{x} = \sum_{i=1}^n f_i \cdot x_i.$$

- **Trung bình có Trọng số (Weighted Average):** Gán trọng số (w_i) cho các giá trị để phản ánh mức độ quan trọng khác nhau. Công thức:

$$\bar{x} = \frac{\sum_{i=1}^n w_i \cdot x_i}{\sum_{i=1}^n w_i}.$$

- **Trung bình nhân (Geometric Mean):** Sử dụng tích của các giá trị thay vì tổng. Hữu ích cho các mối quan hệ nhân hoặc tỷ lệ.

$$\bar{x} = \sqrt[n]{x_1 x_2 \dots x_n}.$$

- **Trung bình điều hoà (Harmonic mean):**

$$\bar{x} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}.$$

- **Trung bình Cắt (Trimmed Mean):** Loại bỏ một tỷ lệ phần trăm nhất định ($r\%$) các giá trị cực đoan từ cả hai đầu của dữ liệu đã sắp xếp để giảm độ nhạy với outliers.

$$\bar{x} = \frac{1}{n_t} \sum_{i=1}^{n_t} x_i.$$

```
[13]: import numpy as np
import scipy.stats as spy

# https://archive.ics.uci.edu/ml/datasets/forest+fires
dforest = pd.read_csv('forestfires.csv')

var = 'FFMC'
weights = np.random.randn(len(dforest[var]))
wavg = np.average(dforest[var], weights=weights)
gavg = spy.gmean(dforest[var]) # From Scipy library
havg = spy.hmean(dforest[var]) # From Scipy library
tavg = spy.trim_mean(dforest[var], 0.05) # 5% trim

print('Weighted average of variable FFMC: {:.2f}'.format(wavg))
print('Geometric mean of variable FFMC: {:.2f}'.format(gavg))
print('Harmonic mean of variable FFMC: {:.2f}'.format(havg))
print('Trimmed mean of variable FFMC: {:.2f}'.format(tavg))

var = 'temp'
weights = np.random.randn(len(dforest[var]))
wavg = np.average(dforest[var], weights=weights)
gavg = spy.gmean(dforest[var]) # From Scipy library
havg = spy.hmean(dforest[var]) # From Scipy library
tavg = spy.trim_mean(dforest[var], 0.05) # 5% trim

print('\nWeighted average of variable temp: {:.2f}'.
      format(wavg))
print('Geometric mean of variable temp: {:.2f}'.format(gavg))
print('Harmonic mean of variable temp: {:.2f}'.format(havg))
print('Trimmed mean of variable temp: {:.2f}'.format(tavg))
```

Weighted average of variable FFMC: 78.89
 Geometric mean of variable FFMC: 90.37
 Harmonic mean of variable FFMC: 89.82
 Trimmed mean of variable FFMC: 91.27

Weighted average of variable temp: 21.43
 Geometric mean of variable temp: 17.74
 Harmonic mean of variable temp: 16.08
 Trimmed mean of variable temp: 19.01

2.2.2 Các số đo biến thiên (Variability Measures)

Các phép đo này chỉ số mức độ trải rộng (spread) của dữ liệu.

- **Miền giá trị (Range):** Khoảng cách giữa giá trị lớn nhất và nhỏ nhất.

$$R = X_L - x_l.$$

- **Khoảng Tứ phân vị (Interquartile Range - IQR):** Đo lường sự phân tán ở nửa trung tâm của phân phối. Được tính bằng:

$$IQR = Q_3 - Q_1.$$

IQR **ít nhạy cảm với outliers** và hữu ích cho các phân phối bị lệch.

- **Khoảng bán tứ phân vị (Semi-Interquartile Range - sIQR):** Đo lường sự phân tán ở nửa trung tâm của phân phối. Được tính bằng:

$$sIQR = \frac{Q_3 - Q_1}{2}.$$

- **Phương sai (Variance) và Độ lệch chuẩn (Standard Deviation):** Đo lường mức độ gần của các giá trị dữ liệu so với giá trị trung bình.

$$\text{Phương sai: } s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \text{Độ lệch: } s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}. \quad (2.2)$$

- **Hệ số biến thiên (Coefficient of Variance-CV):** Là tỷ lệ phần trăm giữ độ lệch tiêu chuẩn và giá trị trung bình

$$CV = \frac{s}{\bar{x}} \cdot 100\%.$$

```
[ ]: import numpy as np
```

```
# https://archive.ics.uci.edu/ml/datasets/forest+fires
dforest = pd.read_csv('forestfires.csv')
```

```

var = 'FFMC'
drange = np.max(dforest[var]) - np.min(dforest[var])
Q1, Q3 = np.percentile(dforest[var], [25, 75])
IQR = Q3 - Q1
sIQR = IQR / 2
dvar = np.var(dforest[var])
dstd = np.std(dforest[var])
CV = dstd / np.mean(dforest[var]) * 100

print('*Variability Measures*')
print('Range of variable FFMC: {:.2f}'.format(drange))
print('IQR of variable FFMC: {:.2f}'.format(IQR))
print('sIQR of variable FFMC: {:.2f}'.format(sIQR))
print('Variance of variable FFMC: {:.2f}'.format(dvar))
print('Standard deviation of variable FFMC: {:.2f}'.format(dstd))
print('Variation coefficient of variable FFMC: {:.2f}'.format(CV))

```

2.2.3 Các số đo hình dạng (Measures of Shape)

Các phép đo định lượng hình dạng của phân phối, thường là các mô men chuẩn hóa (standardized moments).

Moment bậc k được cho bởi công thức

$$m_k = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^k.$$

- **Độ xiên (Skewness):** Đo lường sự bất đối称 (lack of symmetry) của phân phối.

Độ xiên Fischer-Pearson cho bởi công thức

$$\gamma = \frac{m_3}{\sqrt{m_2^3}}.$$

Độ xiên Pearson cho bởi công thức

$$\gamma = \frac{\bar{x} - \text{mode}(x)}{s}.$$

- **Xiên dương (Right-skewed):** Đầu dài hơn về bên phải. Thường có Trung bình > Trung vị > Mode.

- **Xiên âm (Left-skewed):** Đuôi dài hơn về bên trái. Thường có Trung bình < Trung vị < Mode.
- **Độ nhọn (Kurtosis):** Đo lường độ nhọn (peakedness) hoặc độ phẳng (flatness) của phân phối và được xác định bởi công thức

$$\kappa = \frac{m_4}{s^4} - 3.$$

```
[2]: # Skewness and Skewed distributions
# Generate random data with a right-skewed distribution

import statistics as st
import numpy as np
from scipy.stats import skew
import seaborn as sns
import matplotlib.pyplot as plt

data = np.random.beta(a=1, b=5, size=1000)    # Beta distribution
mean = st.mean(data)
median = st.median(data)
midpoint = (max(data) + min(data)) / 2    # Calculate the midpoint
print('Mean, median, and midpoint: {:.2f} {:.2f} {:.2f}'
      .format(mean, median, midpoint))
print('Skewness (Fischer-Pearson Coefficient): {:.2f}'
      .format(skew(data)))
print('Skewness (First Skewness Coefficient): {:.2f}'
      .format((mean - midpoint) / np.std(data)))
sns.histplot(data, bins='auto', kde=2)
plt.axvline(x=mean, color='r', linestyle='--', label='Mean')
plt.axvline(x=median, color='g', linestyle='-', label='Median')
plt.axvline(x=midpoint, color='b', linestyle=':', 
            label='Midpoint')
plt.legend()
plt.show()
```

Mean, median, and midpoint: 0.16 0.13 0.41
 Skewness (Fischer-Pearson Coefficient): 1.28
 Skewness (First Skewness Coefficient): -1.76

```
[3]: # Skewness and Skewed distributions
# Generate random data with a left-skewed distribution
```

```

import statistics as st
import numpy as np
from scipy.stats import skew
import seaborn as sns
import matplotlib.pyplot as plt

data_neg = np.random.beta(a=5, b=1, size=1000) # Beta
distribution
mean = st.mean(data_neg)
median = st.median(data_neg)
midpoint = (max(data_neg) + min(data_neg)) / 2 # Calculate the
midpoint
print('Mean, median, and midpoint: {:.2f} {:.2f} {:.2f}'
      .format(mean, median, midpoint))
print('Skewness (Fischer-Pearson Coefficient): {:.2f}'
      .format(skew(data_neg)))
print('Skewness (First Skewness Coefficient): {:.2f}'
      .format((mean - midpoint) / np.std(data_neg)))
sns.histplot(data_neg, bins='auto', kde=2)
plt.axvline(x=mean, color='r', linestyle='--', label='Mean')
plt.axvline(x=median, color='g', linestyle='-', label='Median')
plt.axvline(x=midpoint, color='b', linestyle=':',
            label='Midpoint')
plt.legend()
plt.show()

```

Mean, median, and midpoint: 0.84 0.88 0.60
 Skewness (Fischer-Pearson Coefficient): -1.25
 Skewness (First Skewness Coefficient): 1.65

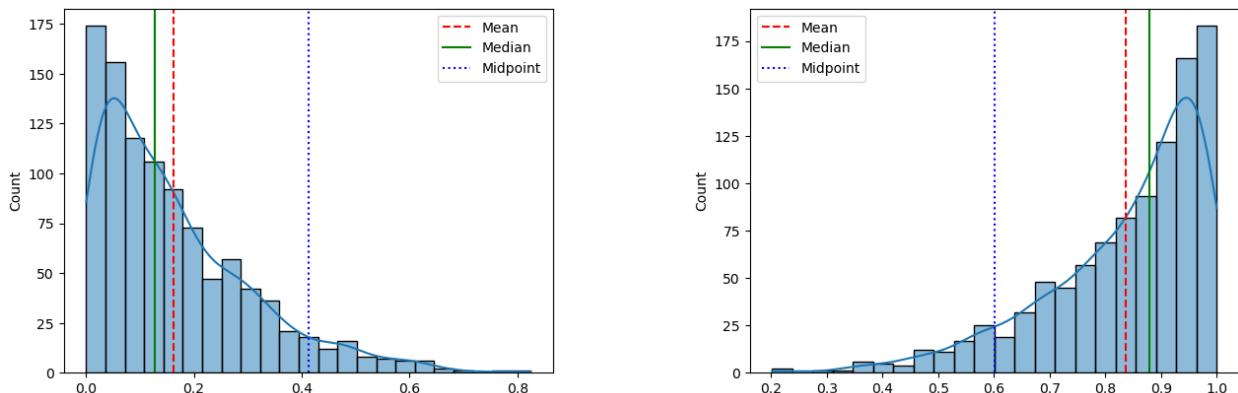
```

[4]: # Kurtosis: mesokurtic, platykurtic and leptokurtic
      distributions

import numpy as np
from scipy.stats import norm, laplace, semicircular
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as spy

# Normal distribution (Mesokurtic)
dnorm = norm.rvs(size=10000)
print(type(dnorm))

```



Hình 2.3: Biểu đồ phân phối với trung bình, trung vị, mode.

```

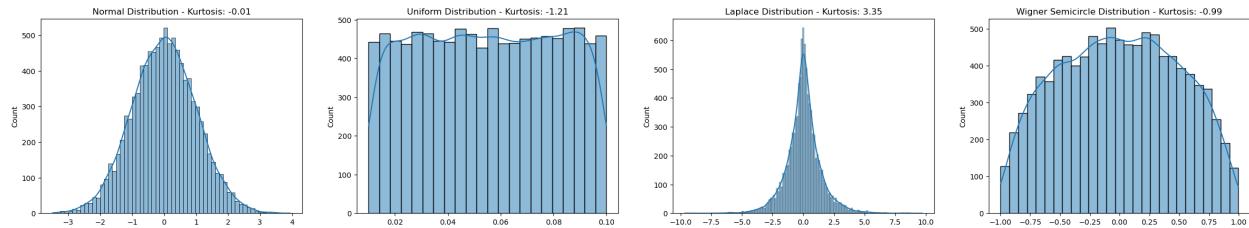
k = spy.kurtosis(dnorm)
print('Kurtosis: {:.2f}'.format(k))
sns.histplot(dnorm,bins='auto', kde = 2)
plt.title(f"Normal Distribution - Kurtosis: {k:.2f}")
plt.show()

# Uniform distribution (Platykurtic)
dunif = np.random.uniform(0.01, 0.10, 10000)
k = spy.kurtosis(dunif)
print('Kurtosis: {:.2f}'.format(k))
sns.histplot(dunif,bins='auto', kde = 2)
plt.title(f"Uniform Distribution - Kurtosis: {k:.2f}")
plt.show()

# Laplace distribution (Leptokurtic)
dlap = laplace.rvs(loc=0, scale=1, size=10000)
k = spy.kurtosis(dlap)
sns.histplot(dlap, bins='auto', kde = 2)
plt.title(f"Laplace Distribution - Kurtosis: {k:.2f}")
plt.show()

# Wigner semicircle distribution
dwigner = semicircular.rvs(size=10000)
k = spy.kurtosis(dwigner)
sns.histplot(dwigner, bins='auto', kde = 2)
plt.title(f"Wigner Semicircle Distribution - Kurtosis: {k:.2f}")
plt.show()

```



Hình 2.4: Các dạng biểu đồ về phân phối với độ nhọn khác nhau.

2.3 Các độ đo mối quan hệ

Các độ đo mối quan hệ tập trung vào việc định lượng **mối quan hệ và sự phụ thuộc lẫn nhau** giữa hai hoặc nhiều biến số, đo lường **cường độ (strength)** và **hướng (direction)** của mối liên hệ đó.

Các Khía cạnh Cần Xem xét của Phép đo Mối liên hệ Khi chọn phép đo, cần xem xét ba khía cạnh chính:

- Loại Phân phối:** Xác định xem phép đo là **tham số (parametric)** (giả định dữ liệu phân phối chuẩn) hay **phi tham số (non-parametric)**.
- Phạm vi Mối liên hệ (Range of Association):** Hầu hết các phép đo tương quan nằm trong phạm vi $[-1, 1]$.
 - $[-1, 1]$: Tương quan tuyến tính (ví dụ: PCC, SRCC, KRCC, PBCC).
 - $[0, 1]$: Chỉ đo cường độ, không đo hướng (ví dụ: Cramer's V).
 - $+1$: Tương quan dương hoàn hảo (cả hai biến cùng tăng hoặc cùng giảm).
 - -1 : Tương quan âm hoàn hảo (một biến tăng, biến kia giảm).
 - 0 : Không có tương quan.
- Thời gian Biến (Variable Time):** Liên quan đến dữ liệu chuỗi thời gian (Time series data).

2.3.1 Hiệp phương sai (Covariance)

Hiệp phương sai đo lường **mức độ thay đổi** của một biến so với biến kia, nhằm mô tả các mối quan hệ giữa các biến.

Hiệp phương sai Cho n cặp quan sát (x_i, y_i) , hiệp phương sai cho bởi công thức

$$\text{cov}(x, y) = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}).$$

Ma trận hiệp phương sai Cho n quan sát với p biến $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$, ma trận hiệp phương sai cho bởi ma trận

$$\Sigma = \left(\text{cov}(x[:, i], x[:, j]) \right)_{p \times p}$$

2.3.2 Hệ số tương quan (Correlation)

Tương quan đo lường **sự phụ thuộc** và cung cấp một mối quan hệ mang tính **dự đoán** giữa các biến.

Hệ số tương quan cho dữ liệu số (Numerical Data)

- **Pearson Correlation Coefficient (PCC):** Là phép đo **tham số**. Được sử dụng cho dữ liệu **liên tục** (Continuous) có phân phối chuẩn và mối quan hệ tuyến tính. Hệ số tương quan Pearson cho bởi công thức:

$$PCC = \rho(x, y) = \frac{\text{cov}(x, y)}{\sqrt{\text{cov}(x, x)\text{cov}(y, y)}}.$$

Ma trận hệ số tương quan xác định là ma trận của hệ số tương quan của tất cả các cặp biến.

- **Spearman Rank Correlation Coefficient (SRCC):** Là phép đo **phi tham số**. Được sử dụng cho dữ liệu **thứ bậc** (Ordinal) hoặc **liên tục** (Continuous) với mối quan hệ đơn điệu (monotonic).

$$SRCC = r_s = \rho(R(x), R(y)) = \frac{\text{cov}(R(x), R(y))}{\sqrt{\text{cov}(R(x), R(x))\text{cov}(R(y), R(y))}}.$$

Công thức trên thường biến đổi thành dạng dễ sử dụng

$$SRCC = r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)},$$

trong đó $d_i = R(x_i) - R(y_i)$ là hiệu thứ hạng của x_i và y_i .

- **Kendall's τ (KRCC):** Là phép đo **phi tham số**. Được sử dụng cho các biến được xếp hạng (ranked variables), đặc biệt khi khoảng cách giữa các biến không thể đo lường. Một cặp gọi là đồng nhất nếu $(x_i - x_j)(y_i - y_j) > 0$ và không đồng nhất nếu $(x_i - x_j)(y_i - y_j) < 0$. Hệ số tương quan hạng Kendall được cho bởi công thức

$$\text{KRCC} = \tau = \frac{\text{Số cặp đồng nhất} - \text{Số cặp không đồng nhất}}{\text{Tổng số cặp}}.$$

Hệ số tương quan cho dữ liệu phân loại (Categorical Data)

- **Chi-square (χ^2)** Là phép đo **phi tham số**. Được sử dụng khi để đánh giá sự khác biệt giữa tần suất quan sát được O (Observed frequency) và tần suất kỳ vọng E (Expected frequency). Công thức tính cho bởi biểu thức

$$\chi^2 = \sum \frac{(O - E)^2}{E}.$$

Phạm vi là $[0, \infty)$.

- Nếu χ^2 nhỏ (tiến về 0): Điều này có nghĩa là tần suất quan sát (O) rất gần với tần suất kỳ vọng (E). Hay nói cách khác, dữ liệu mẫu của bạn phù hợp với Giả thuyết Không (H_0).
- Nếu χ^2 lớn: Điều này có nghĩa là có sự khác biệt đáng kể giữa tần suất quan sát (O) và tần suất kỳ vọng (E). Dữ liệu mẫu không phù hợp với Giả thuyết Không (H_0), dẫn đến việc bác bỏ H_0 .
- **Hệ số phi Cramer (V)** Là phép đo **phi tham số** về **chỉ cường độ** của mối liên hệ giữa hai biến phân loại không phải nhị phân. Dùng cho các biến phân loại có **nhiều hơn hai cấp độ**.

Hệ số Φ được cho bởi công thức

$$\Phi = \sqrt{\frac{\chi^2}{n}},$$

với χ^2 là giá trị thống kê χ^2 , và n là kích thước mẫu. Công thức này thường được sử dụng với bảng 2×2 .

Hệ số Cramer V là mở rộng của hệ số Φ được sử dụng khi bảng tần suất chéo có kích thước lớn hơn 2×2 cho bởi công thức

$$V = \sqrt{\frac{\chi^2}{n(\min(r, c) - 1)}},$$

trong đó χ^2 là giá trị thông kê χ^2 , n là kích thước mẫu, và $\min(r, c)$ là giá trị nhỏ nhất giữa số hàng r và số cột c của bảng tần số chéo.

- $V \approx 0$: Liên hệ rất yếu hoặc không có,
- $V \approx 1$: Liên hệ rất mạnh (hoàn hảo).

- **Hệ số số tương quan PBCC (Point-Biserial Correlation Coefficient):** Là phép đo tham số. Đo cường độ và hướng của mối quan hệ giữa biến nhị phân (binary) và biến liên tục (continuous).

Cho x là biến liên tục và y là biến nhị phân nhận hai giá trị $\{0, 1\}$. Ký hiệu

- M_1 : Giá trị trung bình của x đối với nhóm có $y = 1$,
- M_0 : Giá trị trung bình của x đối với nhóm có $y = 0$,
- s_n : Độ lệch chuẩn của biến x (trên toàn bộ mẫu),
- n_1 : Số lượng quan sát trong nhóm $y = 1$,
- n_0 : Số lượng quan sát trong nhóm $y = 0$,
- n : Tổng cõi mẫu ($n = n_0 + n_1$).

Công thức tính

$$r_{pb} = \frac{M_1 - M_0}{s_n} \sqrt{\frac{n_0 n_1}{n^2}}.$$

Công thức này về cơ bản đo lường sự khác biệt về giá trị trung bình của biến liên tục X giữa hai nhóm. Phạm vi của hệ số r_{pb} là $[-1, 1]$.

```
[9]: # Calculate Correlation Coefficients: PCC, SRCC and KRCC
# for the numerical variables of the Forest Fires dataset

import pandas as pd

# Read the data into a pandas dataframe
dforest = pd.read_csv("forestfires.csv")
pd.options.display.float_format = '{:.2f}'.format

# Calculate PCC, SRCC, KRCC
print('**Forest Fires Dataset: PCC, SRCC, KRCC**')
PCC = dforest.corr(method='pearson')
print('Pearson Correlation Coefficient (PCC)\n', PCC)
SRCC = dforest.corr(method='spearman')
print('\nSpearman Rank Correlation Coefficient (SRCC)\n', SRCC)
```

```
KRCC = dforest.corr(method='kendall')
print('Kramers Rank Correlation Coefficient (KRCC)')

**Forest Fires Dataset: PCC, SRCC, KRCC**
Pearson Correlation Coefficient (PCC)
```

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
X	1.00	0.54	-0.02	-0.05	-0.09	0.01	-0.05	0.09	0.02	0.07	0.06
Y	0.54	1.00	-0.05	0.01	-0.10	-0.02	-0.02	0.06	-0.02	0.03	0.04
FFMC	-0.02	-0.05	1.00	0.38	0.33	0.53	0.43	-0.30	-0.03	0.06	0.04
DMC	-0.05	0.01	0.38	1.00	0.68	0.31	0.47	0.07	-0.11	0.07	0.07
DC	-0.09	-0.10	0.33	0.68	1.00	0.23	0.50	-0.04	-0.20	0.04	0.05
ISI	0.01	-0.02	0.53	0.31	0.23	1.00	0.39	-0.13	0.11	0.07	0.01
temp	-0.05	-0.02	0.43	0.47	0.50	0.39	1.00	-0.53	-0.23	0.07	0.10
RH	0.09	0.06	-0.30	0.07	-0.04	-0.13	-0.53	1.00	0.07	0.10	-0.08
wind	0.02	-0.02	-0.03	-0.11	-0.20	0.11	-0.23	0.07	1.00	0.06	0.01
rain	0.07	0.03	0.06	0.07	0.04	0.07	0.07	0.10	0.06	1.00	-0.01
area	0.06	0.04	0.04	0.07	0.05	0.01	0.10	-0.08	0.01	-0.01	1.00

Spearman Rank Correlation Coefficient (SRCC)

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
X	1.00	0.49	-0.06	-0.08	-0.07	-0.01	-0.05	0.07	0.03	0.11	0.06
Y	0.49	1.00	-0.01	0.00	-0.11	-0.01	-0.04	0.05	-0.01	0.08	0.05
FFMC	-0.06	-0.01	1.00	0.51	0.26	0.78	0.59	-0.32	-0.04	0.10	0.03
DMC	-0.08	0.00	0.51	1.00	0.56	0.43	0.50	0.03	-0.11	0.12	0.07
DC	-0.07	-0.11	0.26	0.56	1.00	0.10	0.31	0.03	-0.21	0.01	0.06
ISI	-0.01	-0.01	0.78	0.43	0.10	1.00	0.42	-0.18	0.14	0.12	0.01
temp	-0.05	-0.04	0.59	0.50	0.31	0.42	1.00	-0.52	-0.18	0.03	0.08
RH	0.07	0.05	-0.32	0.03	0.03	-0.18	-0.52	1.00	0.04	0.18	-0.02
wind	0.03	-0.01	-0.04	-0.11	-0.21	0.14	-0.18	0.04	1.00	0.12	0.05
rain	0.11	0.08	0.10	0.12	0.01	0.12	0.03	0.18	0.12	1.00	-0.06
area	0.06	0.05	0.03	0.07	0.06	0.01	0.08	-0.02	0.05	-0.06	1.00

Kramers Rank Correlation Coefficient (KRCC)

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
X	1.00	0.40	-0.04	-0.06	-0.05	-0.01	-0.04	0.05	0.02	0.09	0.05
Y	0.40	1.00	-0.01	0.00	-0.08	-0.01	-0.03	0.04	-0.01	0.07	0.04
FFMC	-0.04	-0.01	1.00	0.37	0.17	0.62	0.44	-0.22	-0.02	0.08	0.02
DMC	-0.06	0.00	0.37	1.00	0.44	0.30	0.36	0.02	-0.08	0.10	0.05
DC	-0.05	-0.08	0.17	0.44	1.00	0.06	0.19	0.02	-0.14	0.01	0.04
ISI	-0.01	-0.01	0.62	0.30	0.06	1.00	0.29	-0.13	0.10	0.10	0.01
temp	-0.04	-0.03	0.44	0.36	0.19	0.29	1.00	-0.38	-0.13	0.02	0.06
RH	0.05	0.04	-0.22	0.02	0.02	-0.13	-0.38	1.00	0.02	0.15	-0.02
wind	0.02	-0.01	-0.02	-0.08	-0.14	0.10	-0.13	0.02	1.00	0.10	0.04
rain	0.09	0.07	0.08	0.10	0.01	0.10	0.02	0.15	0.10	1.00	-0.06
area	0.05	0.04	0.02	0.05	0.04	0.01	0.06	-0.02	0.04	-0.06	1.00

```
[10]: # Calculate Correlation Coefficients: Chi-square, Cramer's V and
      # Point Biserial
      # for the categorical variables of the Mammographic dataset

from scipy import stats
from scipy.stats import pointbiserialr

# Read the data from URL into a pandas dataframe
url = "https://archive.ics.uci.edu/ml/
       -machine-learning-databases/mammographic-masses/
       -mammographic_masses.data"
cols = ['BI-RADS', 'Age', 'Shape', 'Margin', 'Density',
        'Severity']
dmammo = pd.read_csv(url, names=cols, na_values='?')
dmammo.dropna(inplace=True) # Remove rows with missing values

# Chi-square and Cramer's V
cvars = ['Shape', 'Margin', 'Density', 'Severity'] # 
      #Categorical variables
chis = pd.DataFrame()
phi = pd.DataFrame()
for var1 in cvars:
    for var2 in cvars:
        if var1 != var2:
            chi2, p, dof, ex = stats.chi2_contingency(pd.
              crosstab(dmammo[var1], dmammo[var2]))
            chis.loc[var1, var2] = chi2
            phi.loc[var1, var2] = np.sqrt(chi2 / (dmammo.
              shape[0] * (min(ex.shape) - 1)))
print('\n***Mammographic Dataset: Chi-Square, Crammers V,
      PBCC**\n')
print('Chi-Square Correlation Coefficient (chi^2)\n',chis)
print('\nKramers V Correlation Coefficient (phi)\n',phi)

# Point Biserial Correlation between Age and Severity
PBCC, pval = pointbiserialr(dmammo['Severity'], dmammo['Age'])
print('\nPBCC between Age and Severity: {:.2f}'.format(PBCC))
```

***Mammographic Dataset: Chi-Square, Crammers V, PBCC**

Chi-Square Correlation Coefficient (chi^2)

	Margin	Density	Severity	Shape
Shape	525.12	20.78	284.81	NaN
Margin	NaN	17.79	291.39	525.12
Density	17.79	NaN	6.56	20.78
Severity	291.39	6.56	NaN	284.81

Cramers V Correlation Coefficient (phi)

	Margin	Density	Severity	Shape
Shape	0.46	0.09	0.59	NaN
Margin	NaN	0.08	0.59	0.46
Density	0.08	NaN	0.09	0.09
Severity	0.59	0.09	NaN	0.59

PBCC between Age and Severity: 0.46

2.3.3 So sánh các phép đo tương quan

Các hệ số tương quan được so sánh dựa trên các thuộc tính như tính tham số, phạm vi và loại biến mà chúng xử lý:

Measure	Parametric	Range	Types of Variables
PCC	✓	[-1, 1]	Continuous, Normally distributed data
SRCC	✗	[-1, 1]	Ordinal or continuous
KRCC	✗	[-1, 1]	Ordinal, Ranked variables
χ^2	✗	[0, ∞)	Categorical, Frequency or counts
V	✗	[0, 1]	Categorical (more than two levels)
PBCC	✓	[-1, 1]	Continuous with Binary

2.3.4 Hồi quy tuyến tính đơn

Hồi quy tuyến tính giả định một biến là **biến độc lập** (x) và biến kia là **biến phụ thuộc** (y).

- **Dạng tiêu chuẩn** của hồi quy tuyến tính là:

$$y = a + b \cdot x$$

- Trong đó:

- a : là **hệ số chặn (intercept)**, tương ứng với giá trị mà đường thẳng cắt trục y .
 - b : là **độ dốc (slope)** của đường thẳng. Độ dốc này cho biết **loại tương quan** giữa các biến.
 - y : là biến phụ thuộc (dependent variable).
 - x : là biến độc lập (independent variable).
- Đường hồi quy là đường thẳng mô tả gần đúng nhất phân phối dữ liệu.
 - Các phép đo như tương quan (Correlation) cung cấp một mối quan hệ mang tính **dự đoán** giữa hai biến.
 - **Phân tích hồi quy (Regression analysis)** là quá trình tìm kiếm một đường cong (curve) mô tả gần đúng nhất (fits) dữ liệu, nhằm tóm tắt mối quan hệ giữa các biến.
 - Nếu đường cong này là **một đường thẳng (line)**, quá trình đó được gọi là **Hồi quy tuyến tính (Linear Regression)**.

2.4 BÀI TẬP

Chủ đề và câu hỏi nghiên cứu

Bài 2.1. Thảo luận về ý nghĩa của các hình dạng phân phối khác nhau và việc sử dụng liên tục bảng dữ liệu của cơ quan. Những khía cạnh này ảnh hưởng như thế nào đến việc hiểu dữ liệu của chúng ta?

Bài 2.2. Khám phá ý nghĩa của các phép đo xu hướng trung tâm và độ biến thiên. Các phép đo này giúp chúng ta hiểu các giá trị "diễn hình" và độ phân tán của dữ liệu như thế nào?

Bài 2.3. Thảo luận về những lý do đằng sau sự phổ biến của phân phối chuẩn trong tự nhiên khoa học xã hội và khoa học thực vật. Tại sao nó được coi là phân phối "mặc định" đối với nhiều người hiện tượng?

Bài 2.4. Thảo luận về việc sử dụng hồi quy tuyến tính để hiểu mối quan hệ giữa các biến ables. Liệu mối tương quan có ngụ ý quan hệ nhân quả không?

Bài tập tính toán

Bài 2.5. Tạo ba tập dữ liệu khác nhau:

- Sinh 1000 số ngẫu nhiên phân phối đều giữa 0 và 1.
- Sinh 1000 số ngẫu nhiên tuân theo phân phối chuẩn với trung bình là 50 và độ lệch chuẩn là 10.
- Sinh 1000 số ngẫu nhiên theo phân phối lệch, chẳng hạn như phân phối gamma với tham số hình dạng 2

Vẽ biểu đồ histogram cho từng tập dữ liệu để trực quan hóa hình dạng của chúng. So sánh và thống nhất theo dõi hình dạng của các phân phối.

Bài 2.6. Đối với biến “temp” của tập dữ liệu Cháy rừng, hãy đưa các giá trị trung bình hình học và điều hòa vào biểu đồ và thảo luận về vị trí của chúng khi so sánh với các biện pháp xu hướng trung tâm khác.

Bài 2.7. Tính hệ số lệch Pearson thứ hai và so sánh với hai hệ số lệch khác được trình bày (hệ số lệch Fischer-Pearson và hệ số lệch thứ nhất).

Bài 2.8. Tạo một tập dữ liệu gồm 1.000 số ngẫu nhiên theo phân phối chuẩn với giá trị trung bình là 60 và độ lệch chuẩn là 15. Vẽ biểu đồ hình chữ nhật để trực quan hóa phân phối. Tính phần trăm dữ liệu trong phạm vi một, hai và ba độ lệch chuẩn so với giá trị trung bình.

Nghiên cứu tình huống Phần này trình bày ba nghiên cứu điển hình, trong đó các nhiệm vụ sau đây phải được thực hiện:

- Trình bày từ điển dữ liệu.
- Thực hiện phân tích mô tả dữ liệu, bao gồm phân phối của từng biến (sử dụng biểu đồ), các biện pháp tóm tắt và hình dạng của phân phối.
- Tạo bảng dự phòng cho các biến phân loại.
- Tính toán các biện pháp liên kết (phương sai và tương quan) giữa các biến.
- Vẽ đường hồi quy tuyến tính trên biểu đồ với các cặp biến (biểu đồ phân tán) và quan sát loại tương quan.

- Giải thích kết quả và đưa ra hiểu biết sâu sắc.

Bài 2.9. NGHIÊN CỨU TRƯỜNG HỢP 1 Khoa học sức khỏe – Bộ dữ liệu bệnh tim mạch

<https://www.kaggle.com/datasets/jocelyndumlao/cardiovascular-disease-dataset>

Mô tả tập dữ liệu Kaggle: Tập dữ liệu này chứa thông tin về bệnh nhân tim mạch từ một bệnh viện đa khoa ở Ấn Độ. Nó cung cấp quyền truy cập vào dữ liệu có giá trị của 1.000 cá nhân. Ngoài ra, bao gồm 12 đặc điểm chính thường liên quan đến bệnh tim. Bộ dữ liệu này có thể được sử dụng để phát triển các phương pháp phát hiện sớm và xây dựng các mô hình học máy dự đoán bệnh tim.

Mục tiêu: Hiểu được sự phân bố của các số liệu sức khỏe này trong quần thể bệnh nhân. Xác định bất kỳ mối liên hệ nào giữa chúng và xây dựng các mô hình hồi quy tuyến tính đơn giản giữa các cặp biến.

Bài 2.10. NGHIÊN CỨU TRƯỜNG HỢP 2 Kinh doanh – Bộ dữ liệu bán hàng siêu thị

<https://www.kaggle.com/datasets/rohitsahoo/sales-forecasting>

Mô tả tập dữ liệu: Tập dữ liệu bán lẻ của một siêu thị toàn cầu trong 4 năm. Tập dữ liệu chứa thông tin về ID đơn hàng, dữ liệu đơn hàng, ngày và phương thức giao hàng, phân khúc, quốc gia và thành phố.

Mục tiêu: Hiểu được xu hướng bán hàng và xác định bất kỳ mô hình hoặc mối liên hệ nào.

Bài 2.11. NGHIÊN CỨU TRƯỜNG HỢP 3 Công nghệ – Phân tích dữ liệu hành vi người dùng Netflix

<https://www.kaggle.com/datasets/arnavsmayan/netflix-userbase-dataset>

Mô tả tập dữ liệu Kaggle: Tập dữ liệu này cung cấp một mẫu dữ liệu người dùng Netflix mô phỏng, cung cấp thông tin chi tiết về đăng ký, doanh thu, chi tiết tài khoản và hoạt động của người dùng. Mỗi bản ghi đại diện cho một người dùng duy nhất được xác định bằng ID người dùng ẩn danh. Nó bao gồm các chi tiết như phụ loại đăng ký (Cơ bản, Tiêu chuẩn hoặc Cao cấp), doanh thu đăng ký hàng tháng, ngày tham gia, ngày thanh toán gần nhất và vị trí của người dùng. Các tính năng bổ sung giúp làm rõ hành vi của người dùng, chẳng hạn như loại thiết bị ưa thích (Smart TV, Di động, v.v.) và trạng thái tài khoản hiện tại (đang hoạt động hoặc không hoạt động). Cần lưu ý rằng đây là tập dữ liệu tổng hợp và không phản ánh thông tin người dùng Netflix thực tế.

Mục tiêu: Hiểu hành vi của người dùng, xác định bất kỳ mô hình hoặc mối liên kết nào và dự đoán mức độ tương tác của người dùng.

Chương 3

CÁC NGUYÊN TẮC TRỰC QUAN HÓA DỮ LIỆU

Trực quan hóa dữ liệu (data visualization) là một lĩnh vực quan trọng, dựa trên cách hệ thống thị giác của con người hấp thụ và diễn giải thông tin. Các hình vẽ, minh họa đã được xã hội loài người sử dụng như những phương tiện hiệu quả để truyền tải thông tin quan trọng. Trực quan hóa dữ liệu là một lĩnh vực được phát triển để nghiên cứu các kỹ thuật này nhằm giao tiếp thông tin.

Trong Chương 3 đã trình bày các phép đo tóm tắt (summary measures) để đặc trưng hóa dữ liệu (như giá trị điển hình, độ biến thiên, hình dạng và mối liên hệ giữa các biến). Tuy nhiên, những thông tin số học này thường không đủ để có được kiến thức đầy đủ về phân phối dữ liệu.

Sự thật các tập dữ liệu có sự phân phối rất khác nhau nhưng lại có thể có các phép đo tóm tắt **giống hệt hoặc rất giống nhau**. Trong những trường hợp như vậy, **trực quan hóa đóng vai trò then chốt** trong việc bổ sung cho phân tích để hiểu dữ liệu.

3.1 Quá trình nhận thức thị giác

Trong phần này ta tập trung vào cách hệ thống thị giác của con người hấp thụ và diễn giải thông tin, cung cấp cơ sở nhận thức để thiết kế trực quan hóa dữ liệu hiệu quả.

Phần này chứng minh sự **bất cập** của việc chỉ dựa vào các phép đo tóm tắt (summary measures) dữ liệu bởi các số đo đặc trưng.

Ví dụ 3.1 (Bộ tứ dữ liệu Anscombe). Tập dữ liệu này, được Francis Anscombe tạo ra năm 1973, bao gồm bốn bộ dữ liệu có các **thống kê tóm tắt giống hệt nhau** (như trung bình,

độ lệch chuẩn và tương quan) nhưng có **phân phôi và hình dạng khác nhau** khi được trực quan hóa. Điều này nhấn mạnh rằng các phép đo tóm tắt không đủ và có thể gây hiểu lầm nếu không có trực quan hóa dữ liệu.

```
[5]: # Print the Anscombe's Quartet table with the summary measures
      -(mean, std, corr, linear)
# regression) for each dataset and plot the scatterplots of each
      -of the four datasets

import seaborn as sns
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

# Uses the Seaborn library to load the Anscombe's data
danscombe = sns.load_dataset("anscombe")
pd.options.display.float_format = "{:.2f}".format

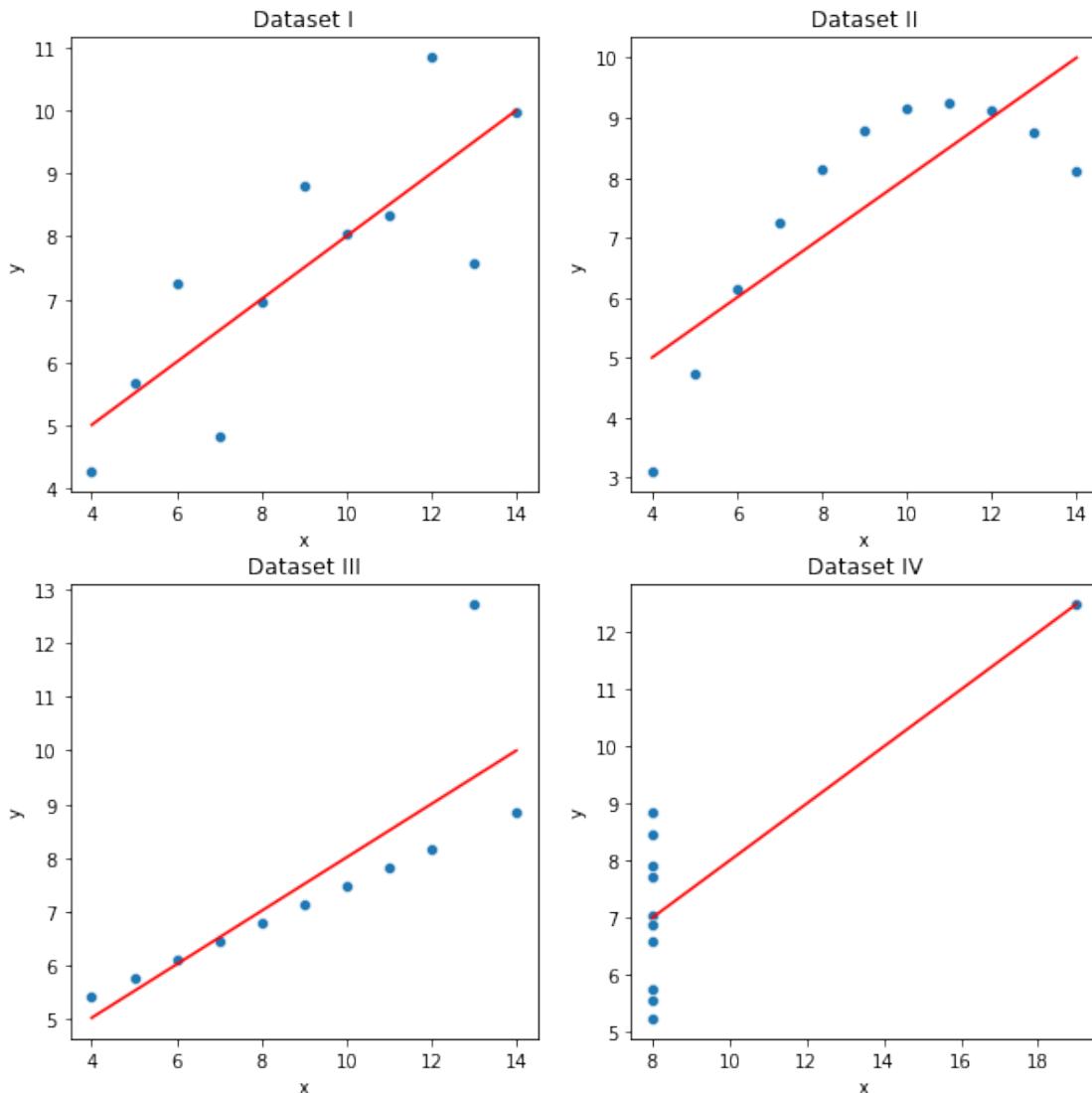
# Print the data and summary measures for each dataset
for dataset in ["I", "II", "III", "IV"]:
    df_subset = danscombe[danscombe.dataset == dataset]
    print(f"Dataset {dataset}\n{df_subset}")
    print(f"Summary Measures for Dataset {dataset}:")
    print(f"Mean of x: {np.mean(df_subset.x):.2f}")
    print(f"Mean of y: {np.mean(df_subset.y):.2f}")
    print(f"Std of x: {np.std(df_subset.x):.2f}")
    print(f"Std of y: {np.std(df_subset.y):.2f}")
    print(f"Correlation between x and y: {np.corrcoef(df_subset.
        -x, df_subset.y)[0,1]:.2f}")
    model = sm.OLS(df_subset.y, sm.add_constant(df_subset.x)).fit()
    print(f"Linear regression model: y = {model.params[0]:.2f} +
        -{model.params[1]:.2f}x\n")

# Plot the scatterplots and regression lines for each dataset
fig, axes = plt.subplots(2, 2, figsize=(10, 10))
for i, dataset in enumerate(["I", "II", "III", "IV"]):
    df_subset = danscombe[danscombe.dataset == dataset]
    x = df_subset.x; y = df_subset.y
    model = sm.OLS(y, sm.add_constant(x)).fit()
    y_pred = model.predict(sm.add_constant(x))
```

```

sns.scatterplot(x=x, y=y, ax=axes[i//2, i%2])
sns.lineplot(x=x, y=y_pred, color="red", ax=axes[i//2, i%2])
axes[i//2, i%2].set_title(f"Dataset {dataset}")
plt.show()

```



Hình 3.1: Bộ tứ dữ liệu Anscombe.

Ví dụ 3.2 (Datasaurus Dozen). Tương tự như Anscombe's Quartet, tập hợp này bao gồm 13 tập dữ liệu có hình dạng khác nhau nhưng có các phép đo tóm tắt **thực tế là giống nhau**

Summary Measures for Dataset:

Mean of x: 54.26

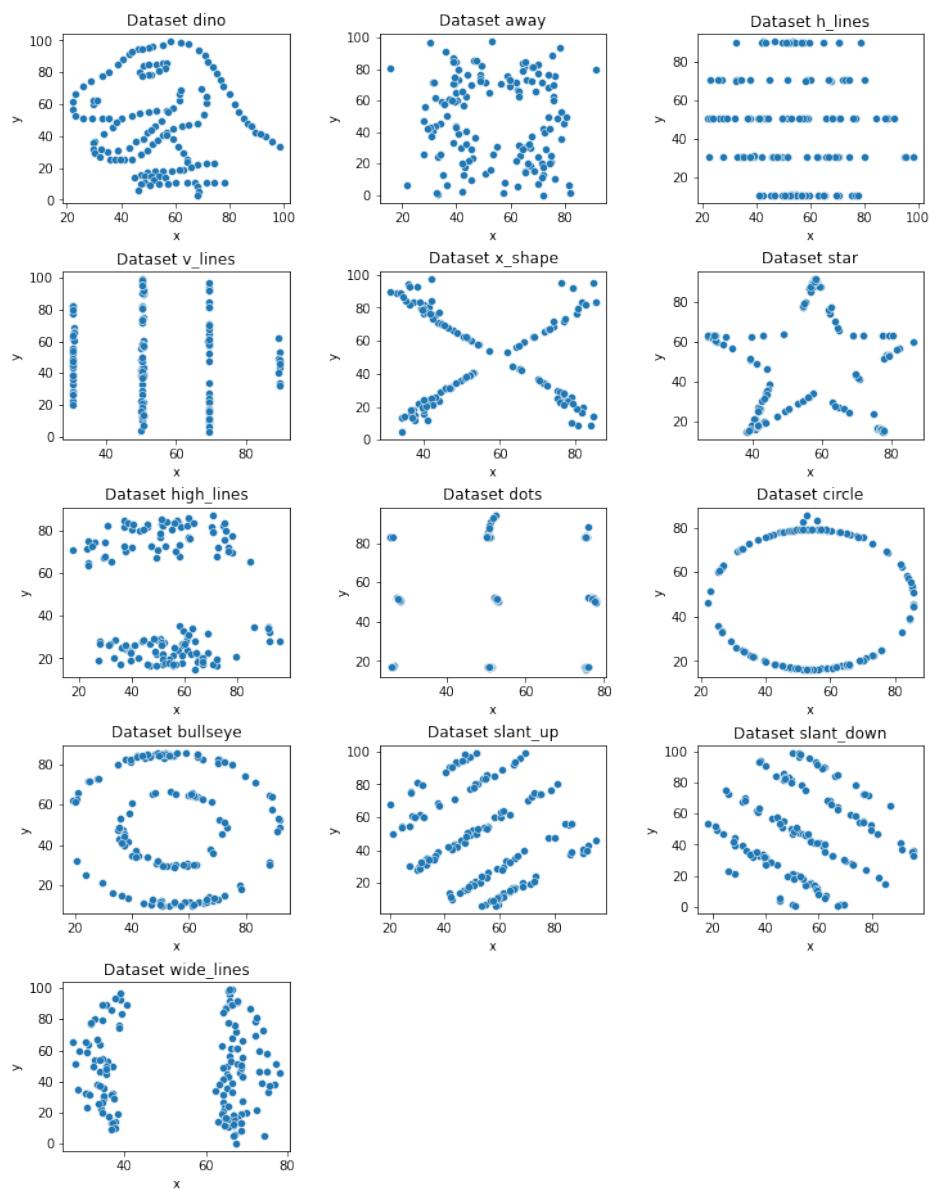
Mean of y: 47.83

Std of x: 16.71

Std of y: 26.84

Correlation between x and y: -0.06

Linear regression model: $y = 53.45 + -0.10x$

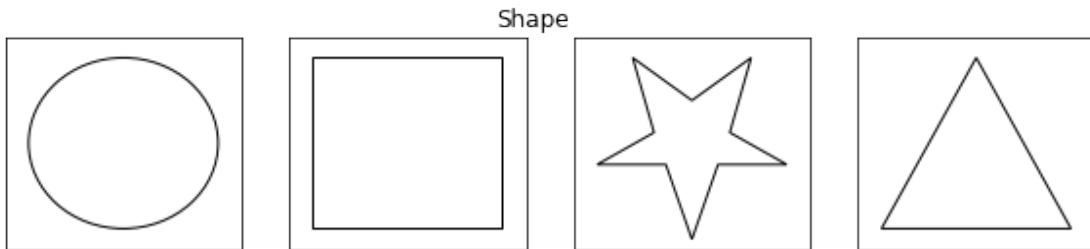


Hình 3.2: Bộ dữ liệu Datasaurus Dozen.

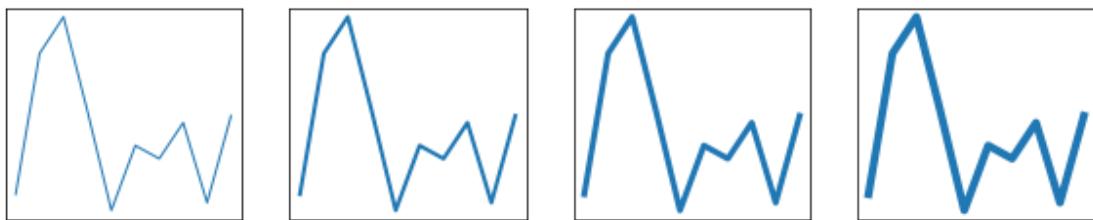
3.1.1 Xử lý tiền chú ý (Preattentive Processing)

Xử lý tiền chú ý là quá trình xử lý nhận thức **tự động và nhanh chóng** các đặc điểm thị giác xảy ra **trước khi sự chú ý** được tập trung vào một đối tượng hoặc vị trí cụ thể.

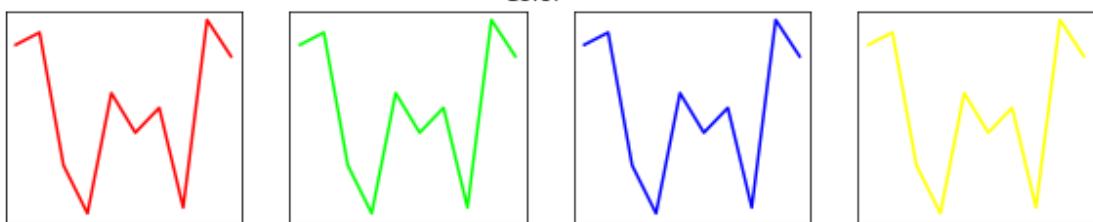
- **Tốc độ:** Các tác vụ có thể được thực hiện trên màn hình đa yếu tố trong vòng **200–250 mili giây** được coi là tiền chú ý.
- **Mục đích:** Bằng cách hiểu các đặc điểm này, nhà phân tích có thể tạo ra các thiết kế trực quan hóa hiệu quả để truyền tải thông tin **hiệu quả và chính xác hơn**.
- **Các Đặc điểm Tiền chú ý Chính:** Đây là các thuộc tính thị giác cơ bản được xử lý tự động, bao gồm:
 1. **Hình dạng (Shape)**
 2. **Màu sắc (Color)**
 3. **Định hướng (Orientation)**
 4. **Kích thước (Size)**
 5. Độ rộng đường kẻ (Line Width)
 6. Dấu hiệu (Markings)
 7. Vị trí (Position)
 8. Chiều sâu 3D (3D Depth Cues)
 9. Độ dài (Length)
 10. Độ cong (Curvature)
 11. Mật độ (Density)
 12. Tính khép kín (Closure)
 13. **Kết cấu (Texture):** Được xử lý tiền chú ý và có thể giúp phân biệt giữa các điểm dữ liệu.



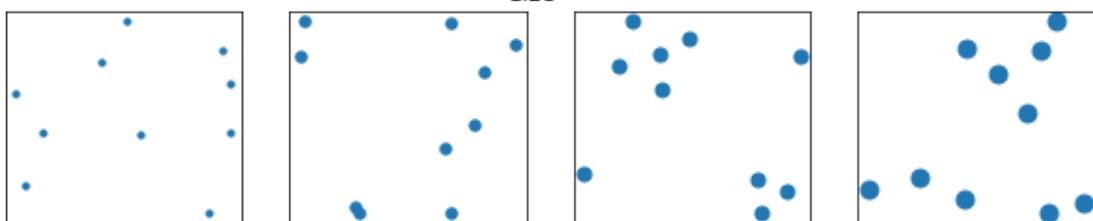
Line Width



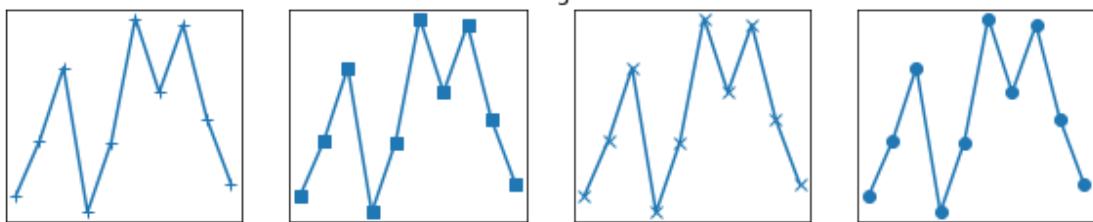
Color



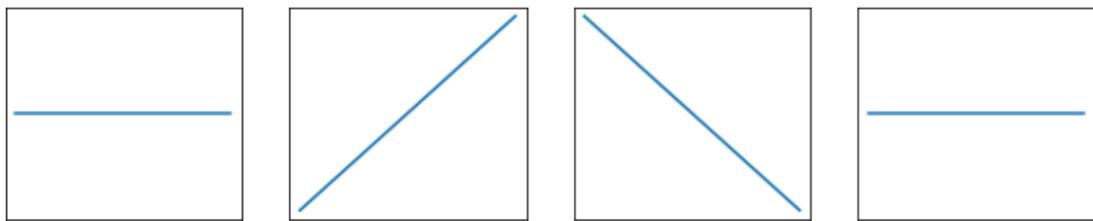
Size

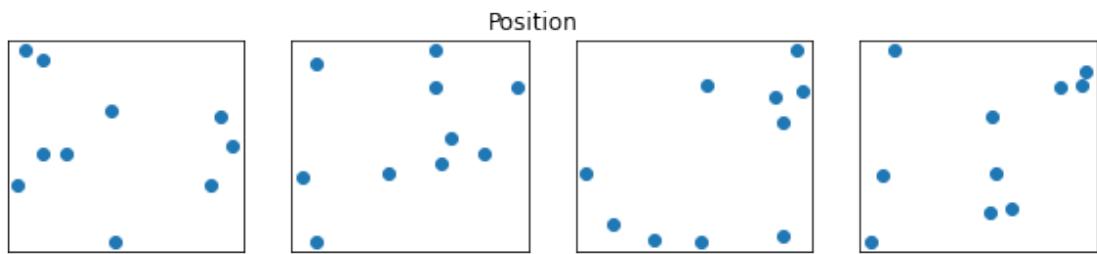


Markings

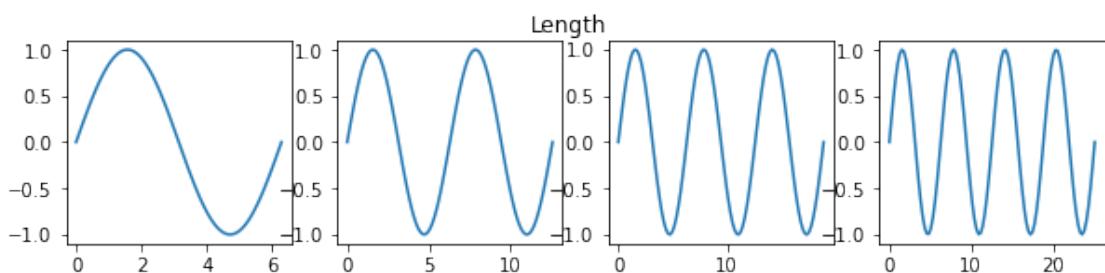
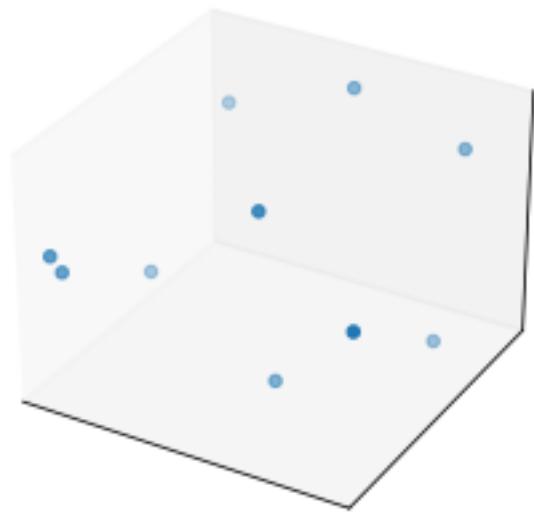


Orientation

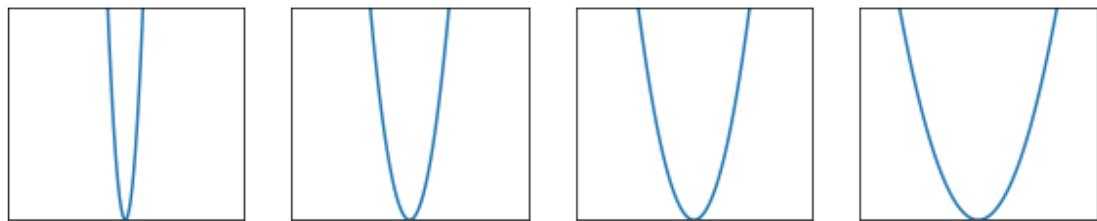


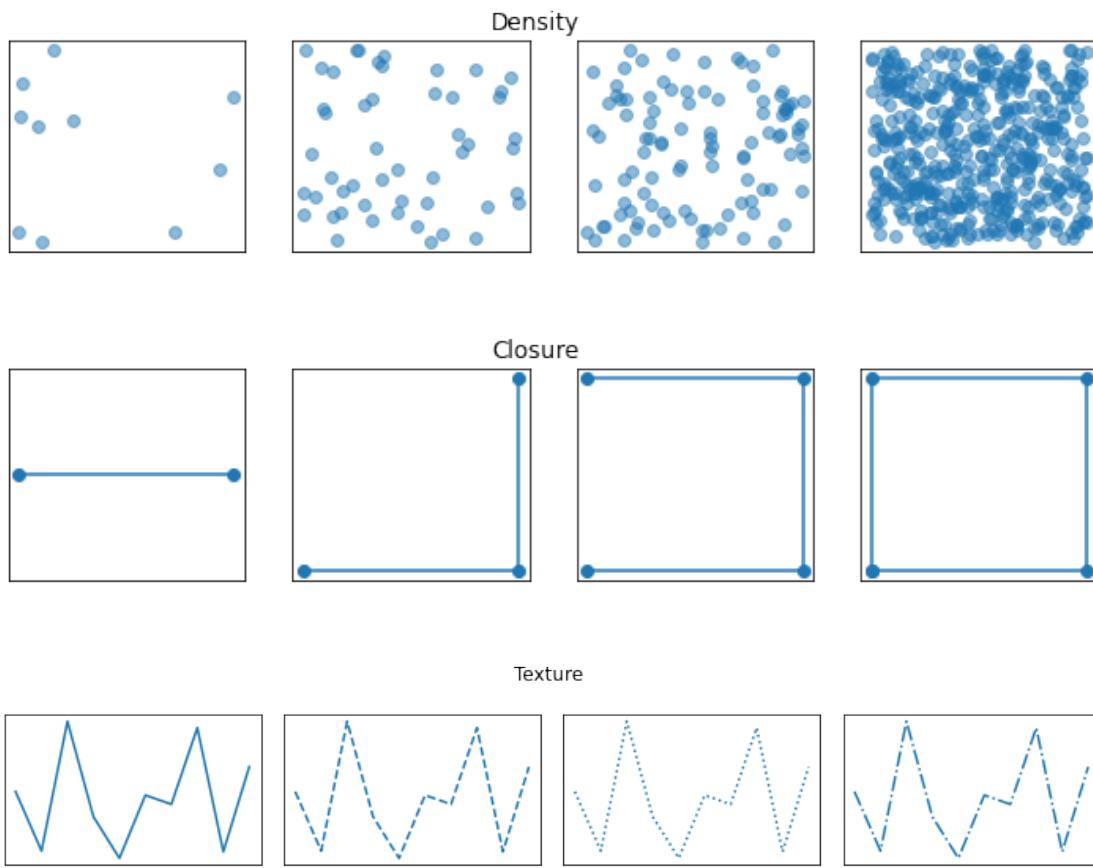


3D



Curvature





3.1.2 Các nguyên tắc Gestalt

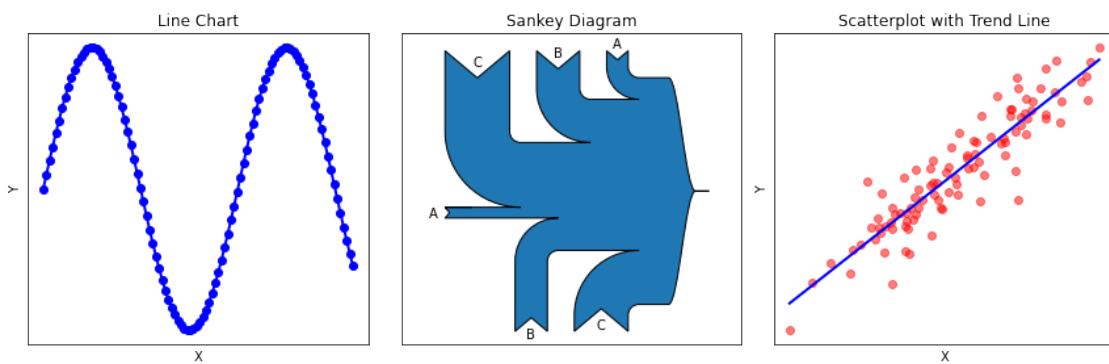
Các nguyên tắc Gestalt là những hướng dẫn cho nhận thức thị giác, giải thích cách bộ não chúng ta tự nhiên **tổ chức và nhóm các yếu tố thị giác** thành các mảng có ý nghĩa. Các nguyên tắc này có cách tiếp cận **toàn diện và không phụ gia** (holistic and non-additive), và là các nguyên tắc thiết kế quan trọng để tạo ra các biểu đồ trực quan hóa dữ liệu hiệu quả.

Phần này mô tả bảy nguyên tắc Gestalt chính liên quan đến trực quan hóa dữ liệu.

Nguyên lý liên tục (Principle of Continuity)

- **Mô tả:** Các đối tượng được sắp xếp theo cách trơn tru, liên tục có nhiều khả năng được nhận thức là một đối tượng duy nhất, ngay cả khi mảng của chúng bị gián đoạn.
- **Ứng dụng trong Trực quan hóa:**

- **Biểu đồ đường (Line chart):** Kết nối các điểm dữ liệu để thể hiện xu hướng liên tục theo thời gian.
- **Biểu đồ Sankey (Sankey diagram):** Sử dụng một loạt mũi tên để hiển thị luồng dữ liệu thông qua một hệ thống.
- **Biểu đồ phân tán (Scatterplot):** Có thể đi kèm với một đường xu hướng (trend line) để chỉ ra một kết nối liên tục.



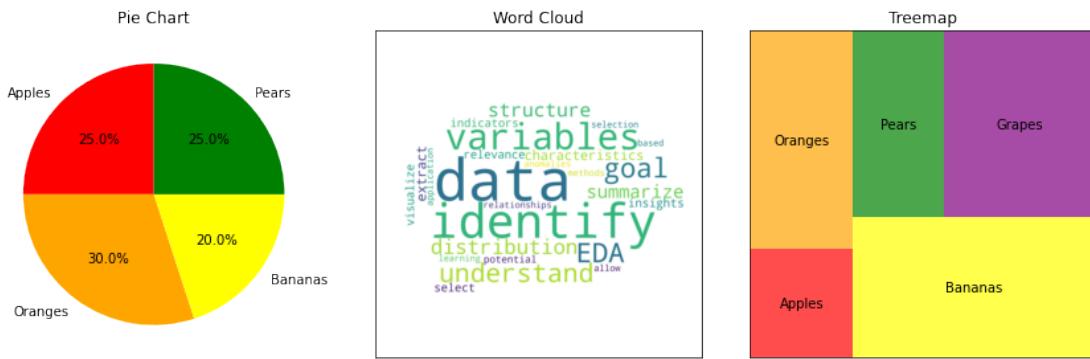
Hình 3.3: Nguyên lý liên tục.

Nguyên lý khép kín (Principle of Closure)

- **Mô tả:** Khả năng tự động hoàn thành một đối tượng hoặc hình dạng bị thiếu một phần.
- **Ứng dụng trong Trực quan hóa:**
 - **Biểu đồ tròn (Pie chart):** Các phần tử được vẽ trong một ranh giới để tạo thành một hình dạng đóng kín.
 - **Treemap.**
 - **Word cloud (Đám mây từ):** Các từ được vẽ trong một mặt nạ (mask) để tạo thành một hình bóng đóng kín.

Nguyên lý tương đồng (Principle of Proximity)

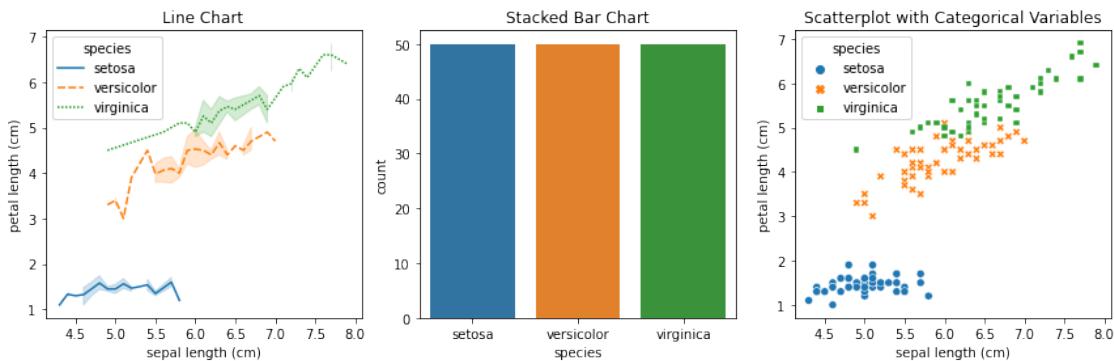
- **Mô tả:** Các đối tượng gần nhau có xu hướng được nhận thức là một nhóm hoặc một mảng.



Hình 3.4: Nguyên lý đóng kín.

- **Ứng dụng trong Trực quan hóa:**

- **Heatmap:** Các ô liền kề trong ma trận có màu tương tự truyền tải cảm giác tổ chức và mối quan hệ.
- **Biểu đồ phân tán (Scatterplot):** Các giá trị dữ liệu tương tự được đặt gần nhau.
- **Bảng (Tables):** Sử dụng khoảng trắng để nhóm dữ liệu thuộc về nhau và phân tách chúng khỏi các dữ liệu khác.

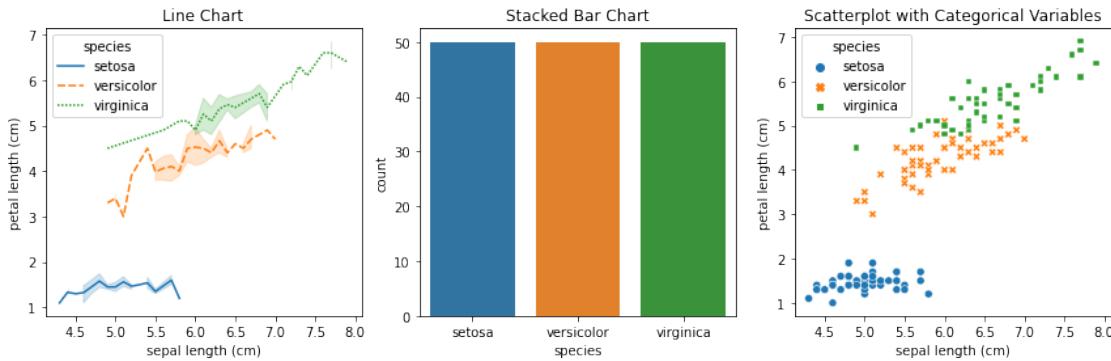


Hình 3.5: Nguyên lý tương đồng.

Nguyên lý tương tự (Principle of Similarity)

- **Mô tả:** Các đối tượng chia sẻ các đặc điểm tương tự, như **màu sắc** hoặc **hình thức (shape)**, có xu hướng được nhận thức là một nhóm hoặc một mẫu.
- **Ứng dụng trong Trực quan hóa:**

- Biểu đồ đường và biểu đồ phân tán:** Các đường cong hoặc điểm dữ liệu có xu hướng và sự gần gũi tương tự được nhóm lại.
- Biểu đồ thanh (Bar chart):** Mẫu hoặc màu sắc của các thanh cho thấy chúng thuộc cùng một nhóm hoặc danh mục.



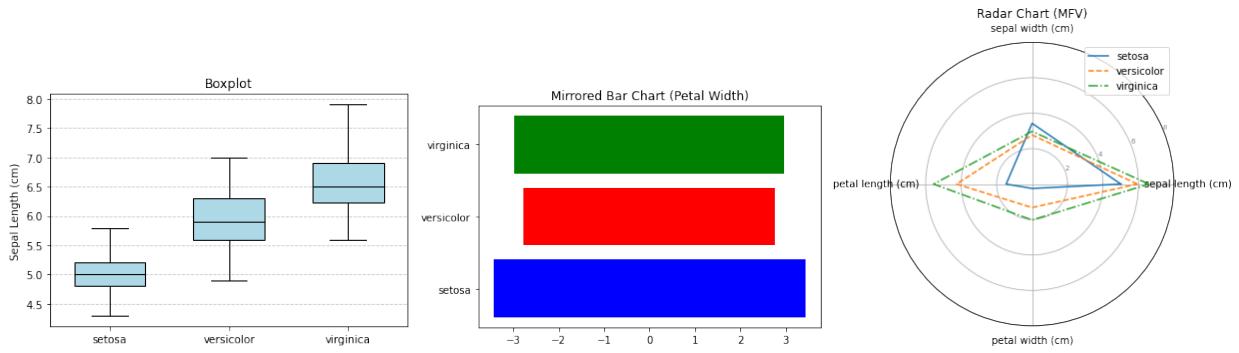
Hình 3.6: Nguyên lý tương đồng.

Nguyên lý đối xứng (Principle of Symmetry)

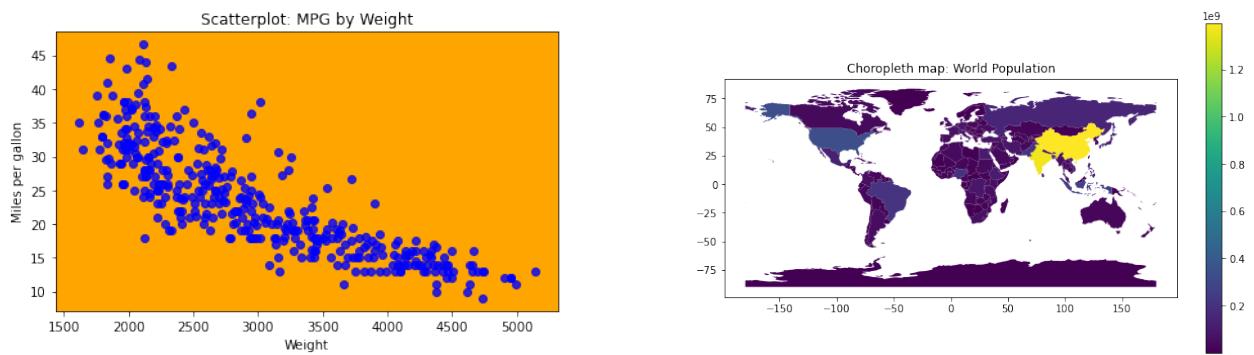
- Mô tả:** Các đối tượng đối xứng (symmetrical) hoặc có vẻ ngoài cân bằng có xu hướng được nhận thức là một nhóm hoặc một mẫu.
- Ứng dụng trong Trực quan hóa:**
 - Boxplot:** Với các hộp được đặt đối xứng xung quanh giá trị trung vị (Q_2).
 - Biểu đồ mạng nhện (Radar chart).**
 - Biểu đồ thanh gương (Mirrored bar chart).**

Nguyên lý hình nền (Principle of Figure-Ground)

- Mô tả:** Giúp phân tách đối tượng chính (hình) khỏi bối cảnh (nền).
- Minh họa:** Được minh họa bằng các biểu đồ như biểu đồ phân tán, biểu đồ bong bóng và bản đồ choropleth.



Hình 3.7: Nguyên lý đối xứng.



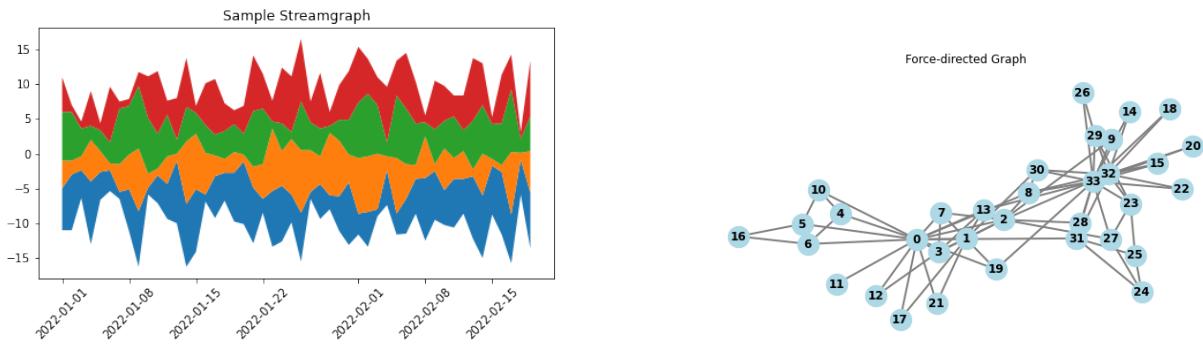
Hình 3.8: Nguyên lý hình nền.

Nguyên lý đồng bộ (Principle of Common Fate)

- **Mô tả:** Các đối tượng **di chuyển cùng nhau** hoặc thay đổi một cách tương tự có xu hướng được nhận thức là một nhóm hoặc một mẫu.
- **Ứng dụng trong Trực quan hóa:** Các đồ thị phải thể hiện một loại hoặc một cảm giác chuyển động. Ví dụ bao gồm:
 - **Motion chart (Biểu đồ chuyển động).**
 - **Streamgraph (Biểu đồ dòng).**
 - **Force-directed graph.**

3.1.3 Các nguyên tắc của Tufte

Edward Tufte – nhà thống kê và chuyên gia trực quan hóa dữ liệu hàng đầu thế giới – đã đưa ra một hệ thống các nguyên lý thiết kế đồ họa thông tin rất sâu sắc, chủ yếu



Hình 3.9: Nguyên lý đồng bộ.

được trình bày trong các cuốn sách nổi tiếng của ông: *The Visual Display of Quantitative Information*, *Envisioning Information*, *Visual Explanations*, và *Beautiful Evidence*.

Sáu nguyên tắc Tính toàn vẹn đồ họa (Graphical Integrity)

1. Trình bày số liệu phải tỷ lệ chính xác với giá trị thực

Không cắt trực tung, không dùng hiệu ứng 3D làm méo tỷ lệ, không phóng đại bằng diện tích hay thể tích.

2. Sử dụng nhãn rõ ràng, chi tiết và trung thực

Ghi đầy đủ nguồn dữ liệu, đơn vị đo, thời gian, chú thích.

3. Lượng mực dùng để thể hiện dữ liệu phải tỷ lệ với lượng thông tin số

Tối đa hoá Data-Ink Ratio, loại bỏ hoàn toàn chartjunk.

4. Không thay đổi tỷ lệ trong cùng một biểu đồ

Trục ngang và trục dọc phải thống nhất từ đầu đến cuối.

5. Tối đa hoá mật độ thông tin (Data Density)

Đưa càng nhiều dữ liệu có ý nghĩa vào càng ít diện tích càng tốt.

6. Hệ số đối trả (Lie Factor) phải gần bằng 1

$$\text{Lie Factor} = \frac{\text{kích thước thay đổi trong đồ họa}}{\text{kích thước thay đổi thực tế trong dữ liệu}}$$

Giá trị lý tưởng: $0.95 \leq \text{Lie Factor} \leq 1.05$.

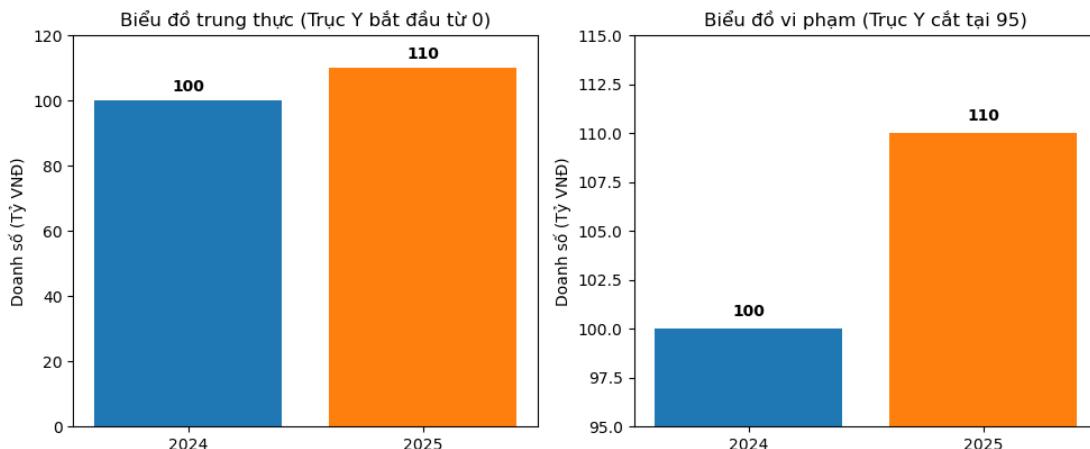
Ví dụ 3.3 (Lie Factor).

Ví dụ 3.4 (Data-Ink ratio).

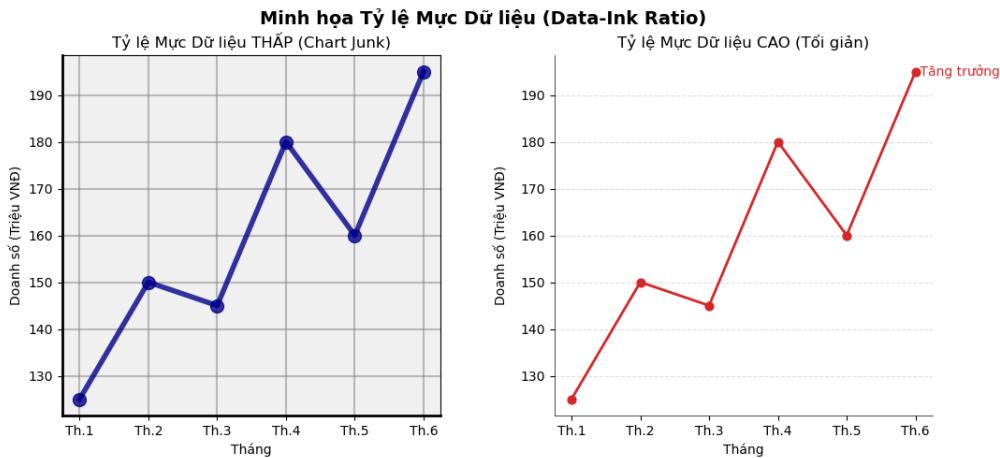
Bảng 3.1: Tóm tắt các nguyên tắc Gestalt

Nguyên lý	Mô tả
Continuity (liên tục)	Các đối tượng được sắp xếp theo cách trơn tru, liên tục có nhiều khả năng được nhận thức là một đối tượng duy nhất, ngay cả khi mẫu của chúng bị gián đoạn.
Closure (khép kín)	Nguyên tắc này liên quan đến khả năng của chúng ta trong việc hoàn thành (đóng kín) một đối tượng hoặc một hình dạng bị thiếu một phần.
Proximity (gần gũi)	Nguyên tắc này đề xuất rằng các đối tượng gần nhau có xu hướng được nhận thức là một nhóm hoặc một mẫu.
Similarity (tương tự)	Nguyên tắc này đề xuất rằng các đối tượng chia sẻ các đặc điểm tương tự, chẳng hạn như màu sắc hoặc hình thức (shape), có xu hướng được nhận thức là một nhóm hoặc một mẫu.
Symmetry (đối xứng)	Các đối tượng đối xứng, hoặc có vẻ ngoài cân bằng, có xu hướng được nhận thức là một nhóm hoặc một mẫu.
Figure-Ground (hình nền)	Nguyên tắc này giúp phân tách đối tượng chính (hình) khỏi bối cảnh (nền).
Common Fate (tương đồng)	Nguyên tắc này đề xuất rằng các đối tượng di chuyển cùng nhau hoặc thay đổi một cách tương tự có xu hướng được nhận thức là một nhóm hoặc một mẫu.

Minh họa nguyên lý tính toàn vẹn đồ họa (Graphical Integrity)



Hình 3.10: Hệ số nói dối.



Hình 3.11: Tỷ lệ mực dữ liệu.

Sự xuất sắc đồ họa (Graphical Excellence)

Sự xuất sắc đồ họa (Graphical excellence) là việc truyền đạt những ý tưởng phức tạp một cách rõ ràng, chính xác và hiệu quả nhất.

Các tiêu chí để đạt được Graphical Excellence:

1. Trình bày thật nhiều dữ liệu trong không gian nhỏ nhất có thể
2. Khuyến khích người xem so sánh các phần dữ liệu khác nhau
3. Thể hiện dữ liệu trung thực, không bóp méo sự thật
4. Tích hợp chặt chẽ số liệu – chữ viết – hình ảnh
5. Loại bỏ hoàn toàn *chartjunk*
6. Có mục đích rõ ràng: mô tả, khám phá, bảng biểu hay trang trí

Các nguyên lý thiết kế quan trọng khác

Ví dụ 3.5. Bản đồ nổi tiếng của Charles Minard về chiến dịch Nga năm 1812 của Napoleon là một sơ đồ luồng dữ liệu (*flow map*) kết hợp nhiều chiều dữ liệu một cách bậc thầy để mô tả sự mất mát sinh mạng thảm khốc chỉ trong một đồ họa tinh duy nhất.

Bản đồ hiển thị sáu loại dữ liệu chính:

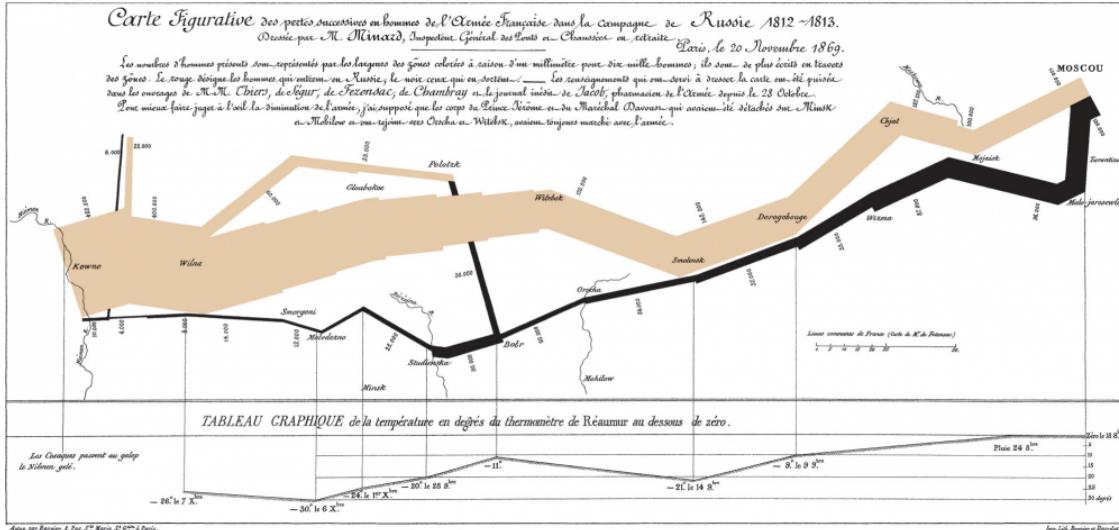
- **Số lượng quân:** Chiều rộng của đường đại diện cho sự di chuyển của quân đội tỷ lệ thuận với số lượng binh lính tại bất kỳ điểm nào. Đường này mỏng đi một cách

Nguyên lý	Mô tả	Ví dụ
Data-Ink Ratio	Tối đa hoá tỉ lệ mực dùng cho dữ liệu / tổng mực	Xóa lưới, viền thừa, bóng 3D
Chartjunk	Mọi yếu tố trang trí không mang thông tin	Biểu đồ bánh 3D, cột hình người
Small Multiples	Lặp lại cùng một thiết kế nhỏ để so sánh nhiều biến	50 biểu đồ thời tiết nhỏ của một thành phố
Layering & Separation	Phân tầng thông tin bằng màu, độ dày nét, vị trí để não dẽ tách lớp	Biểu đồ Minard về Napoleon
Micro/Macro Readings	Cho phép đọc chi tiết (micro) và tổng quan (macro) trong cùng một biểu đồ	Sparkline + số liệu lớn
Color for Information	Chỉ dùng màu khi cần phân biệt dữ liệu, không dùng để trang trí	Đánh dấu bất thường bằng màu đỏ
1 + 1 = 3 effect	Tránh tạo ảo giác thị giác do chồng chéo lưới, hoa văn	Không dùng lưới đậm chồng lên nhau

đáng kinh ngạc từ lực lượng ban đầu hơn 400.000 người xuống chỉ còn khoảng 10.000 người sống sót khi quay về.

- **Hướng di chuyển:** Hành trình tiến đến Moscow được thể hiện bằng dải màu sáng (thường là màu be hoặc cam), trong khi cuộc rút lui tàn khốc là dải màu tối tương phản (đen), cho thấy dòng chảy của quân đội.
- **Vị trí địa lý:** Các đường được vẽ trên một bản đồ tối giản, hiển thị kinh độ và vĩ độ, cũng như các thành phố và con sông cụ thể mà quân đội đã đi qua, chẳng hạn như Kaunas, Minsk, Smolensk và sông Berezina.
- **Khoảng cách đã đi:** Chiều dài của các đường và bản đồ bên dưới cung cấp cảm giác về khoảng cách rộng lớn đã được bao phủ trong chiến dịch.
- **Ngày tháng:** Các ngày quan trọng được đánh dấu dọc theo con đường, cho phép người xem tương quan quy mô quân đội và vị trí với các thời điểm cụ thể.
- **Nhiệt độ:** Một biểu đồ riêng biệt ở dưới cùng của bản đồ chính được liên kết trực tiếp với hành trình trở về, vẽ biểu đồ nhiệt độ cực thấp, đóng băng (xuống tới -30 độ Réaumur hay -37,5 °C) đã tàn phá quân đội rút lui trong mùa đông khắc nghiệt của Nga.

Sức mạnh của bản đồ nằm ở khả năng kể câu chuyện bi thảm về tiến trình của chiến dịch và sự đau khổ của con người với sự hùng hồn tàn bạo, tích hợp tất cả các biến số này vào một câu chuyện trực quan mạch lạc duy nhất.



Hình 3.12: Mô tả Bản đồ Minard về Chiến dịch Nga năm 1812.

Ứng dụng các nguyên lý của Tufte trong biểu đồ.

1. Tỷ lệ Mức Dữ liệu Cao (*High Data-Ink Ratio*)

- **Tối giản:** Biểu đồ gần như chỉ là mực dữ liệu. Minard loại bỏ các chi tiết bản đồ không cần thiết (như đường biên giới hành chính, địa hình chi tiết) và chỉ giữ lại đường bờ sông để định vị.
 - **Mục tiêu:** Mọi nét mực đều phục vụ cho việc kể câu chuyện về số lượng quân đội, vị trí, và thời gian. Không có *Chart Junk* nào làm xao nhãng khỏi thông điệp chính.

2. Tích hợp Văn bản và Đồ hoa

- **Chú thích tích hợp:** Thay vì đặt chú thích và số liệu ở xa, Minard tích hợp số lượng quân đội (ví dụ: "422,000 người", "100,000 người") ngay trên đường đi tại các điểm quan trọng.
 - **Dòng nhiệt độ:** Biểu đồ nhiệt độ ở phía dưới được căn chỉnh theo thời gian và vị trí trên bản đồ phía trên, cho phép người xem ngay lập tức thấy mối tương quan giữa sự suy giảm quân số và cái lạnh thảm khốc.

3. Thể hiện Đa chiều (*Multivariate Data*)

- **Tính đa chiều:** Đây là đỉnh cao của thiết kế dữ liệu. Biểu đồ Minard đã thành công kết hợp **năm biến số** vào một không gian 2 chiều duy nhất mà vẫn giữ được sự rõ ràng tuyệt đối.
- **Nguyên lý:** Nó chứng minh rằng đồ họa có thể vượt qua giới hạn của hai chiều màn hình để truyền tải thông tin đa chiều phức tạp.

4. Tính toàn vẹn của Đồ họa (*Graphical Integrity*)

- **Tính tỷ lệ:** Chiều rộng của đường trên biểu đồ được đo đạc cẩn thận để **tỷ lệ chính xác** với số lượng quân đội. Khi số lượng quân giảm 50% (ví dụ: từ 422,000 xuống 200,000), chiều rộng đường cũng giảm chính xác 50%.
- **Tính trung thực:** Biểu đồ trung thực kể về **mức độ khủng khiếp** của thảm họa, không che giấu hay phóng đại.

5. Kể chuyện (*Narrative Graphics*)

- **Sự trôi chảy:** Toàn bộ biểu đồ được thiết kế để dễ dàng theo dõi dòng chảy của câu chuyện: từ sự khởi đầu hùng mạnh, đến sự chia tách và hợp nhất của các đơn vị, và cuối cùng là sự tan rã bi thảm.
- **Mục tiêu:** Biểu đồ không chỉ là tập hợp số liệu mà là một **phân tích lịch sử** được mã hóa bằng thị giác, khiến nó trở thành một ví dụ mẫu mực về cách kể chuyện bằng dữ liệu.

Ví dụ 3.6 (John Snow). Biểu đồ của John Snow về dịch tả ở Soho, London năm 1854 (thường gọi là "Cholera Map" hoặc "Snow's Dot Map") là một trong những biểu đồ quan trọng nhất trong lịch sử khoa học và trực quan hóa dữ liệu.

- **Dữ liệu được trực quan hóa:** Vị trí của các **ca tử vong do dịch tả** và vị trí của các **máy bơm nước công cộng** (nguồn cung cấp nước).
- **Mục tiêu:** Chứng minh giả thuyết rằng dịch tả lây lan qua nước bị ô nhiễm (thay vì qua không khí như quan điểm phổ biến lúc bấy giờ).

Các Nguyên lý Tufte được áp dụng



Hình 3.13: Biểu đồ của John Snow về dịch tả ở Soho, London năm 1854.

1. Tích hợp Văn bản và Đồ họa (*Integration of Text and Graphic*)

- **Tính tích hợp:** Bản đồ John Snow không tách rời các ghi chú phân tích ra khỏi dữ liệu. Các vị trí tử vong và vị trí máy bơm được đánh dấu trực tiếp trên bản đồ nền, cho phép người xem ngay lập tức thấy mối quan hệ không gian.
- **Vị trí tử vong:** Các thanh đen nhỏ được xếp chồng lên nhau tại các địa chỉ cụ thể để thể hiện số lượng ca tử vong. Số lượng thanh chính là dữ liệu, được đặt chính xác tại vị trí địa lý của nó.
- **Mục tiêu:** Giúp người xem so sánh trực tiếp dữ liệu bệnh tật với dữ liệu cơ sở hạ tầng (máy bơm nước).

2. Tính toàn vẹn của Đồ họa (*Graphical Integrity*)

- **Tính tỷ lệ:** Bản đồ thể hiện chính xác vị trí địa lý của các con phố và các ca tử vong. Không có sự bóp méo nào về mặt không gian.
- **Sự rõ ràng:** Trực quan hóa **tạo ra bằng chứng** bằng cách cho thấy rõ ràng các ca tử vong tập trung thành một cụm xung quanh **máy bơm nước Broad Street** (tâm điểm của ổ dịch).

3. Tối đa hóa Mật độ Dữ liệu (*Maximize Data Density*)

- Bản đồ chứa đựng một lượng lớn thông tin trên một diện tích nhỏ: Vị trí của **hang trăm ca tử vong** (chi tiết và cụ thể), bố cục của **toàn bộ khu phố** và **vị trí của tất cả các máy bơm nước**.
- Sự cô đọng này cho phép người xem thực hiện **phân tích không gian (spatial analysis)** chỉ bằng mắt thường.

4. Tỷ lệ Mực Dữ liệu Cao (*High Data-Ink Ratio*)

- **Thiết kế tối giản:** Bản đồ chỉ sử dụng các nét mực cần thiết: phác thảo đường phố, các dấu chấm/thanh đại diện cho cái chết, và các ký hiệu cho máy bơm.
- **Loại bỏ Chart Junk:** Không có màu sắc rực rỡ, hiệu ứng trang trí hay đường lưỡi khòn cần thiết. Sự tập trung hoàn toàn là vào mối quan hệ giữa **Cái chết** (Dữ liệu) và **Máy bơm** (Nguyên nhân giả định).

Bản đồ John Snow là một ví dụ hoàn hảo về cách một đồ thị không chỉ là một công cụ trình bày mà còn là một **công cụ phân tích và khám phá**, đã thay đổi nhận thức khoa học và cứu sống hàng ngàn người.

3.1.4 Các nguyên tắc sử dụng màu sắc

Phần này ta nghiên cứu việc lập bản đồ màu và các kênh phi không gian khác trong thiết kế mã hóa thị giác, bao gồm:

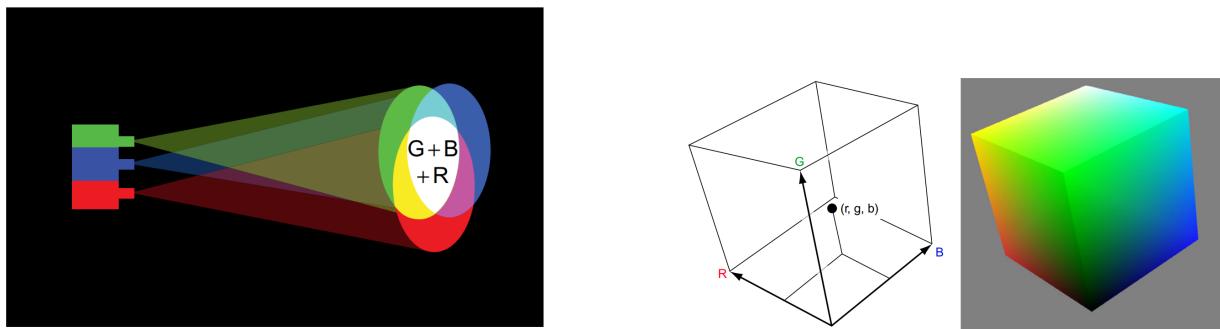
- **Màu sắc:** Được hiểu thông qua ba kênh riêng biệt: độ chói (luminance), sắc độ (hue) và độ bão hòa (saturation).

- Kênh Độ lớn (Magnitude Channels): Kích thước (size), góc (angle) và độ cong (curvature).
- Kênh Nhận dạng (Identity Channels): Hình dạng (shape) và chuyển động (motion).

Không gian màu

Mắt người có ba loại tế bào nón nhạy cảm với các bước sóng khác nhau của ánh sáng nhìn thấy. Hệ thống thị giác xử lý các tín hiệu này thành ba kênh màu đối lập: đỏ-xanh lá cây, xanh dương-vàng, và đen-trắng (kênh độ chói). Kênh độ chói truyền tải thông tin cạnh (edge information) với độ phân giải cao, trong khi các kênh sắc độ (chromaticity) có độ phân giải thấp hơn. Thiếu hụt màu sắc (thường gọi là mù màu), chủ yếu là đỏ-xanh lá cây, ảnh hưởng đến khoảng 8% nam giới.

Không gian màu (Color Spaces) Không gian màu là không gian ba chiều. Hệ thống RGB (Red, Green, Blue) tiện lợi cho tính toán nhưng không khớp tốt với cơ chế thị giác con người vì các trục không phải là kênh tách biệt.



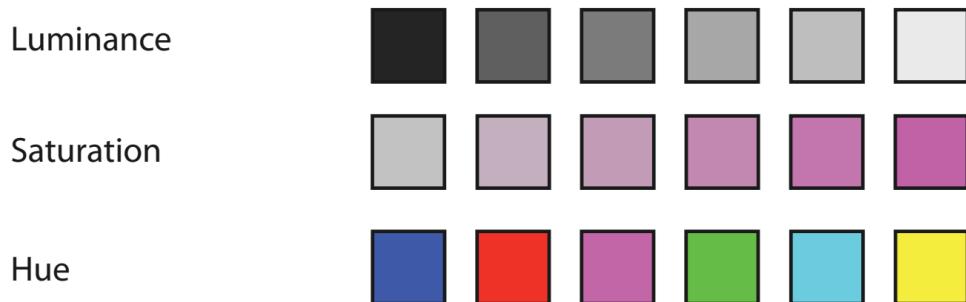
Hình 3.14: Không gian màu RGB.

Không gian HSL (Hue, Saturation, Lightness) hoặc HSV (Hue, Saturation, Value) trực quan hơn cho các nhà thiết kế: Hue là màu thuần khiết, Saturation là lượng màu trắng được pha trộn, và Lightness/Value là lượng màu đen được pha trộn. Tuy nhiên, HSL chỉ là giả cảm nhận vì độ sáng (Lightness L) không tương ứng chính xác với độ chói mà chúng ta cảm nhận được.

- Độ chói (Luminance): Là kênh độ lớn (magnitude channel), phù hợp cho dữ liệu có thứ tự (ordered data). Độ chói là cách duy nhất chúng ta có thể phân giải chi tiết

mịn và nhìn thấy các cạnh sắc nét; do đó, văn bản không thể đọc được nếu thiếu độ tương phản độ chói (đề xuất tối thiểu 3:1, lý tưởng là 10:1).

- Độ bão hòa (Saturation): Là kênh độ lớn, cũng phù hợp cho dữ liệu có thứ tự. Độ chính xác thấp cho các vùng không liền kề và chỉ nên dùng khoảng ba bin phân biệt. Màu có độ bão hòa cao (sáng) nên dùng cho các vùng nhỏ (ví dụ: đường kẻ); màu bão hòa thấp (pastel) nên dùng cho các vùng lớn (ví dụ: nền).
- Sắc độ (Hue): Là kênh nhận dạng (identity channel), cực kỳ hiệu quả cho dữ liệu phân loại (categorical data) và phân nhóm,. Sắc độ không có thứ tự cảm nhận mặc định (unlike luminance/saturation).
- Độ trong suốt là một kênh thứ tư liên quan đến màu sắc, tương tác mạnh với các kênh màu khác và không nên được sử dụng độc lập. Nó thường được sử dụng cho dữ liệu nhị phân (chỉ hai bước) hoặc để phân biệt các lớp chồng lên nhau.



Hình 3.15: Hệ thống thị giác của chúng ta tự động hiểu các kênh độ chói (luminance) và độ bão hòa (saturation) theo thứ tự, nhưng kênh sắc độ (hue) thì không.

Công thức chuyển đổi RGB sang HSL, HSV Giả sử giá trị RGB nằm trong khoảng [0, 255].

$$R' = \frac{R}{255}, G' = \frac{G}{255}, B' = \frac{B}{255}.$$

$$C_{\max} = \max(R', G', B'), C_{\min} = \min(R', G', B'), \Delta = C_{\max} - C_{\min}.$$

Hue (H) – Sắc màu (độ)

$$H = \begin{cases} 0^\circ & \text{nếu } \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \mod 6 \right) & \text{nếu } C_{\max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & \text{nếu } C_{\max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & \text{nếu } C_{\max} = B' \end{cases}$$

Lightness (L) – Độ sáng (%)

$$L = \frac{C_{\max} + C_{\min}}{2} \times 100\%$$

Saturation (S) – Độ bão hòa (%)

$$S = \begin{cases} 0\% & \text{nếu } C_{\max} = 0 \\ \frac{\Delta}{C_{\max}} \times 100\% & \text{ngược lại} \end{cases}$$

Value (V) – Độ sáng (%)

$$V = C_{\max} \times 100\%$$

Bảng màu

Bản đồ màu (colormap) là sự ánh xạ giữa màu sắc và giá trị dữ liệu.

Bản đồ màu phân loại (Categorical Colormaps) Sử dụng màu sắc để mã hóa các danh mục.

Số lượng màu có thể phân biệt được cho các vùng nhỏ bị tách biệt là hạn chế, thường là từ sáu đến mười hai bin.

Để tránh vấn đề phân biệt màu sắc và tính nổi bật, nên điều chỉnh độ bão hòa phù hợp với loại dấu (mark type) (ví dụ: độ bão hòa cao cho các dấu chấm nhỏ, độ bão hòa thấp cho các khu vực lớn).

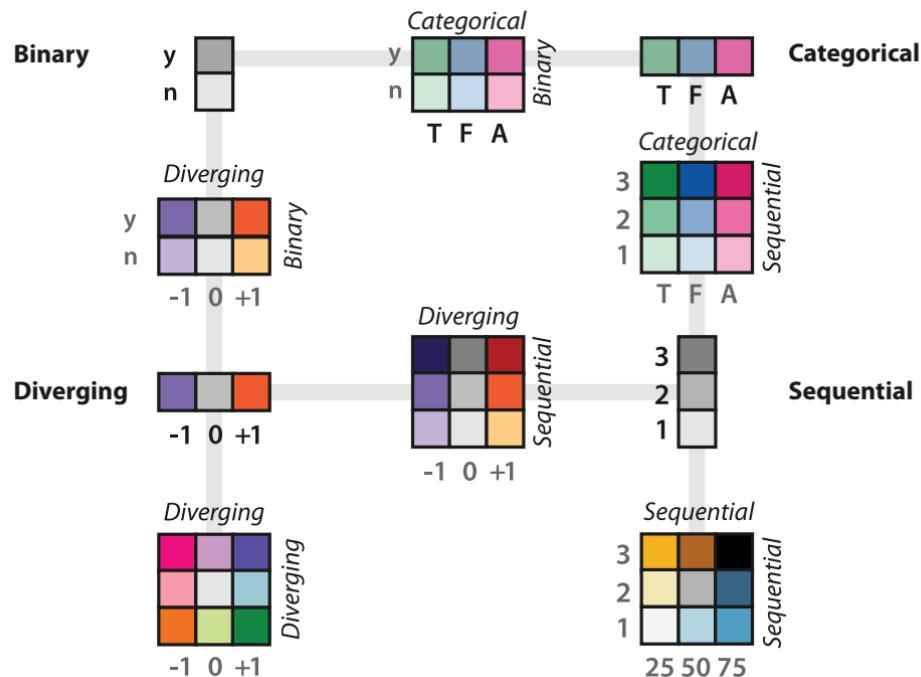
Bản đồ màu có thứ tự (Ordered Colormaps) Phù hợp cho dữ liệu định lượng hoặc có thứ tự. Nên sử dụng các kênh độ lớn như độ chói và độ bão hòa vì sắc độ không có thứ tự cảm nhận mặc định.

Bản đồ màu tuần tự (Sequential) Thể hiện sự tiến triển từ giá trị tối thiểu đến tối đa.

Bản đồ màu phân kỳ (Diverging) Có hai sắc độ ở hai đầu và một màu trung tính (ví dụ: trắng, xám) ở điểm giữa (thường là điểm 0).

Bản đồ màu cầu vồng (Rainbow Colormaps) Chúng là một lựa chọn mặc định không may trong nhiều phần mềm.

- Không khớp về tính biểu đạt: Sử dụng kênh nhận dạng (hue) để biểu thị thứ tự.
- Không tuyến tính về mặt cảm nhận: Các bước thay đổi giá trị bằng nhau không được mắt cảm nhận như nhau (non-perceptually linear).
- Không khớp về độ chính xác: Kênh sắc độ không thể hiện chi tiết mịn (fine detail) tốt như kênh độ chói.



Hình 3.16: Sự phân loại bản đồ màu (colormap categorization) phản ánh một phần các loại dữ liệu: phân loại (categorical) so với có thứ tự (ordered), và tuần tự (sequential) cùng phân kỳ (diverging) nằm trong nhóm có thứ tự.

Giải pháp cho dữ liệu có thứ tự: Thiết kế bản đồ màu có độ chói tăng đơn điệu (monotonically increasing luminance), trong đó các sắc độ được sắp xếp theo độ chói từ thấp nhất đến cao nhất. Điều này cung cấp thứ tự cảm nhận và khả năng phân biệt chi tiết tốt hơn.

Bản đồ màu hai biến (Bivariate Colormaps)

- Mã hóa hai thuộc tính riêng biệt cùng một lúc.
- Cách sử dụng an toàn nhất là khi một trong hai thuộc tính là nhị phân (chỉ có hai cấp độ).
- Chúng khó giải thích khi cả hai thuộc tính đều có nhiều cấp độ.

Thiết kế an toàn cho người mù màu (Colorblind-Safe)

- Cần xem xét vấn đề thiếu hụt màu đỏ-xanh lá cây, ảnh hưởng đến 8% nam giới.
- Chiến lược an toàn nhất là tránh chỉ sử dụng kênh sắc độ (hue) để mã hóa thông tin; thay vào đó, hãy thay đổi độ chói hoặc độ bão hòa để đảm bảo sự tương phản. Nên sử dụng các công cụ mô phỏng để kiểm tra thiết kế.

Các kênh thị giác khác Các kênh thị giác phi không gian quan trọng khác bao gồm:

- Kênh kích thước: Phù hợp cho dữ liệu có thứ tự. Độ chính xác của việc đánh giá độ dài là cao nhất, tiếp theo là diện tích; còn thể tích (3D) thì rất không chính xác.,
- Kênh Góc/Hướng: Phù hợp cho dữ liệu có thứ tự, có độ chính xác cao hơn diện tích.
- Kênh hình dạng: Kênh nhận dạng, có thể có hàng chục hoặc hàng trăm bin phân biệt nếu điểm dữ liệu đủ lớn.
- Kênh chuyển động: Kênh nhận dạng, có tính nổi bật cao và thường được phân loại là nhị phân (di chuyển hoặc không di chuyển), chủ yếu dùng để làm nổi bật (highlighting),,
- Kênh độ cong: Độ chính xác thấp, phù hợp với dữ liệu có thứ tự nhưng chỉ nên dùng khoảng hai hoặc ba bin phân biệt

```
[1]: import matplotlib.pyplot as plt
print(plt.colormaps())
```

```
[2]: import matplotlib.pyplot as plt
import numpy as np
gradient = np.linspace(0, 1, 256)
```

```

gradient = np.vstack((gradient, gradient))
sequential_colormaps = ['viridis', 'plasma', 'inferno', 'magma',
-'cividis', 'turbo']
n = len(sequential_colormaps)
fig, axs = plt.subplots(n, 1, figsize=(10, n*0.35),
- facecolor='white')
for i, cmap_name in enumerate(sequential_colormaps):
    if cmap_name == '':
        axs[i].axis('off')
    continue
    ax = axs[i]
    ax.imshow(gradient, aspect='auto', cmap=cmap_name)
    ax.text(0, 0.5, cmap_name, va='center', ha='left',
- fontsize=12,
        fontfamily='consolas', weight='bold', color='black',
        bbox=dict(facecolor='white', alpha=0.8,
- edgecolor='none', pad=2))
    ax.set_axis_off()
plt.show()

```



Hình 3.17: Bảng màu cho dữ liệu tuần tự.

```

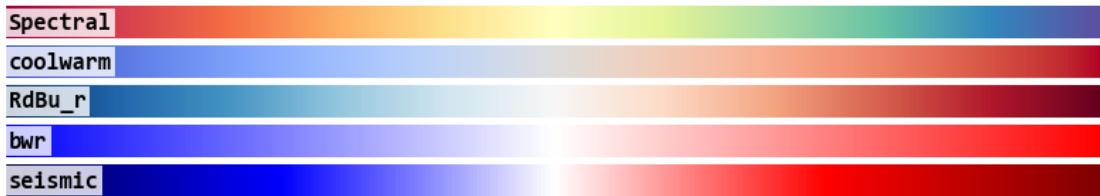
[3]: import matplotlib.pyplot as plt
import numpy as np
gradient = np.linspace(0, 1, 256)
gradient = np.vstack((gradient, gradient))
diverging_colormaps =
-['Spectral', 'coolwarm', 'RdBu_r', 'bwr', 'seismic']
n = len(diverging_colormaps)
fig, axs = plt.subplots(n, 1, figsize=(10, n*0.35),
- facecolor='white')
for i, cmap_name in enumerate(diverging_colormaps):
    if cmap_name == '':
        axs[i].axis('off')

```

```

continue
ax = axs[i]
ax.imshow(gradient, aspect='auto', cmap=cmap_name)
ax.text(0, 0.5, cmap_name, va='center', ha='left',
    fontsize=12,
        fontfamily='consolas', weight='bold', color='black',
        bbox=dict(facecolor='white', alpha=0.8,
    edgecolor='none', pad=2))
ax.set_axis_off()
plt.show()

```



Hình 3.18: Bảng màu cho dữ liệu phân kỳ.

```

[4]: import matplotlib.pyplot as plt
import numpy as np
gradient = np.linspace(0, 1, 256)
gradient = np.vstack((gradient, gradient))
cyclic_colormaps = ['twilight', 'twilight_shifted', 'hsv']
n = len(cyclic_colormaps)
fig, axs = plt.subplots(n, 1, figsize=(10, n*0.35),
    facecolor='white')
for i, cmap_name in enumerate(cyclic_colormaps):
    if cmap_name == '':
        axs[i].axis('off')
    continue
ax = axs[i]
ax.imshow(gradient, aspect='auto', cmap=cmap_name)
ax.text(0, 0.5, cmap_name, va='center', ha='left',
    fontsize=12,
        fontfamily='consolas', weight='bold', color='black',
        bbox=dict(facecolor='white', alpha=0.8,
    edgecolor='none', pad=2))
ax.set_axis_off()
plt.show()

```



Hình 3.19: Bảng màu cho dữ liệu có chu kỳ.

```
[5]: import matplotlib.pyplot as plt
import numpy as np
qualitative_colormaps = [
    'tab10',           # 10 màu - chuẩn khoa học, mặc định cho
    ↪pandas/seaborn
    'tab20',           # 20 màu
    'tab20b',          # 20 màu khác (nâu, xám...)
    'tab20c',          # 20 màu thứ 3 (nhạt hơn)
    '',
    'Set1',            # 9 màu mạnh
    'Set2',            # 8 màu pastel nhẹ nhàng
    'Set3',            # 12 màu nhạt
    'Pastel1',         # 9 màu pastel đậm
    'Pastel2',         # 8 màu pastel nhẹ
    'Paired',          # 12 màu đối (rất hay dùng)
    'Accent',          # 8 màu nổi bật
    'Dark2'            # 8 màu tối, đẹp cho nền trắng
]
cmaps = [c for c in qualitative_colormaps if c.strip() != '']
n = len(cmaps)
fig = plt.figure(figsize=(11, n * 0.6), facecolor='white',
    ↪dpi=150)

for i, cmap_name in enumerate(cmaps):
    cmap = plt.get_cmap(cmap_name)
    n_colors = cmap.N
    for j in range(n_colors):
        ax = fig.add_axes([0.1 + j*0.07, 0.96 - i*0.08, 0.06, 0.
    ↪06])
        ax.imshow([[cmap(j)]], aspect='auto')
        ax.set_axis_off()
    if cmap_name in ['tab10', 'Paired', 'Set2', 'Dark2',
    ↪'Colorblind']:
        stt = ' ' if cmap_name == 'tab10' else ' '
        print(stt, cmap_name)
```

```

fw = 'bold'
col = '#c41e3a'

else:
    stt = ''
    fw = 'medium'
    col = 'black'
    fig.text(0.1 + n_colors*0.07 + 0.02, 0.96 - i*0.08 + 0.03,
              f'{stt}{cmap_name} ({n_colors} màu)',
              va='center', ha='left', fontsize=14, fontweight=fw,
              color=col,
              fontfamily='Consolas')
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.axis('off')
plt.show()

```



Hình 3.20: Bảng màu cho dữ liệu định tính.

3.2 Các nguyên tắc thiết kế cho trực quan hóa dữ liệu

Trong phần này tập trung vào các hướng dẫn thực tế để tạo ra các bảng (tables) và đồ thị (graphs) hiệu quả, xây dựng trên nền tảng của xử lý thị giác và các nguyên tắc Gestalt đã được thảo luận trước đó, và cung cấp các nguyên tắc thiết kế cụ thể cho từng loại.

3.2.1 Bảng (Tables)

Bảng là một trong những phương pháp trực quan hóa dữ liệu phổ biến nhất, thường là điểm khởi đầu cho việc phân tích dữ liệu có cấu trúc. Trong bảng, hàng thường tương ứng với đối tượng, và cột tương ứng với các biến. Bảng tận dụng các nguyên tắc Gestalt về **Tính gần gũi** (Proximity) và **Tính liên tục** (Continuity) để tổ chức dữ liệu.

Car ID	Buying	Persons	Lug_boot	Safety	Class
1	vhigh	2	small	low	unacc
2	vhigh	2		med	unacc
3	vhigh	2		high	unacc
4	vhigh	2		med	unacc

Hình 3.21: Các thành phần chính của một bảng dữ liệu.

1. Khi nào nên sử dụng Bảng

Bảng đặc biệt hữu ích khi cần:

- Xác định giá trị cụ thể với độ chính xác cao.
- So sánh các cặp giá trị liên quan.
- Thực hiện **tra cứu và so sánh từng cặp**.
- So sánh các đối tượng dựa trên nhiều đặc điểm với **các đơn vị đo lường khác nhau**.
- Trình bày cả dữ liệu đơn vị và dữ liệu tóm tắt trên một màn hình duy nhất.

2. Các nguyên tắc thiết kế cho Bảng

Các hướng dẫn thiết kế nhằm chống lại sự **rối mắt (clutter)** của bảng:

- a. **Sử dụng đường kẻ tối thiểu:** Chỉ sử dụng đường kẻ làm khung bảng và để tách hàng tiêu đề.
- b. **Không sử dụng đường kẻ dọc** và không sử dụng đường kẻ giữa các hàng dữ liệu trong toàn bộ bảng.
- c. **Làm nổi bật tiêu đề** (headers) so với phần thân: Bằng cách sử dụng chữ **in đậm** hoặc đường kẻ.
- d. **Hướng dẫn người đọc bằng khoảng trắng:** Khoảng trắng được sử dụng giữa các hàng và cột để **nhóm dữ liệu liên quan lại với nhau** (dựa trên Tính gần gũi).
- e. **Loại bỏ việc lặp lại đơn vị:** Các đơn vị đo lường nên được trình bày trong tiêu đề, phụ đề, hoặc tiêu đề cột.
- f. **Nhấn mạnh các giá trị cụ thể:** Có thể làm nổi bật các giá trị quan trọng (ví dụ: hiệu suất tốt nhất) bằng cách sử dụng chữ in đậm, tô bóng màu xám hoặc màu sắc, khoanh tròn, hoặc các dấu hiệu khác.
- g. **Nhóm dữ liệu tương tự và làm nổi bật chúng:** Các nhóm có thể được làm nổi bật bằng cách tăng khoảng cách giữa các hàng hoặc sử dụng một số màu hoặc sơ đồ tô bóng (việc tô bóng cần phải tinh tế).

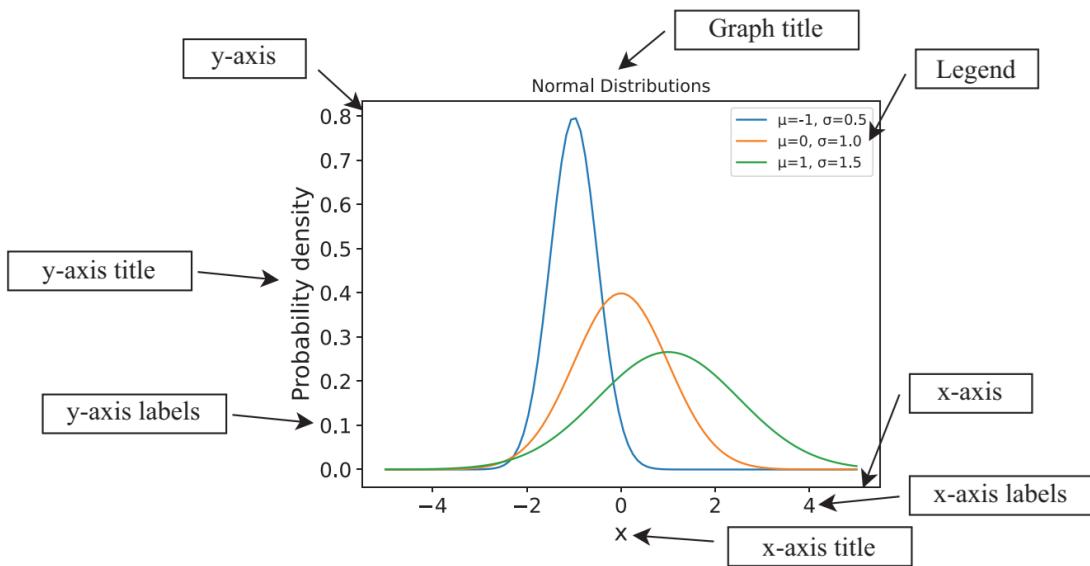
3.2.2 Đồ thị (Graphs)

Đồ thị hiển thị các giá trị bằng cách sử dụng các đối tượng thị giác trong khu vực giới hạn bởi các trục. Đồ thị cho phép chúng ta thực hiện **so sánh tương đối và xấp xỉ** các giá trị.

1. Khi nào nên sử dụng Đồ thị

Đồ thị đặc biệt hữu ích khi cần:

- a. **Kiểm tra một tập hợp các giá trị định lượng để tìm ra hình dạng, mẫu (pattern), hoặc xu hướng chung** của nó.
- b. Chỉ cần một đơn vị đo lường duy nhất để so sánh các giá trị khác nhau.
- c. Tìm kiếm **các mẫu cụ thể**, chẳng hạn như các giá trị bất thường và các cụm.
- d. **Khối lượng dữ liệu quá lớn** đối với một bảng nhưng có thể được tóm tắt trong một đồ thị.



Hình 3.22: Các thành phần chính của một đồ thị.

2. Các lựa chọn thiết kế Đồ họa (Graphical Design Choices)

Thiết kế trực quan hóa đồ họa thành công đòi hỏi sự kết hợp hiệu quả giữa nội dung, ngữ cảnh, xây dựng và thiết kế.

A. Lựa chọn Hình thức Đồ họa (Choice of Graphical Form) Bước này phụ thuộc vào **loại dữ liệu có sẵn** (ví dụ: liên tục, phân loại) và **thông tin cần truyền tải** (ví dụ: phân phối, mối liên hệ, tỷ lệ).

B. Tùy chọn Hiển thị Đồ họa (Graphical Display Options)

- **Thang đo (Scales):** Đối với các biến phân loại, thang đo nên phản ánh một thứ tự thông tin mong muốn. Đối với các biến liên tục, cần xác định điểm cuối, phân chia và các dấu mốc.
- **Chú thích, Chú giải và Chú thích bổ sung:**
 - **Chú giải (Legends)** mô tả các ký hiệu hoặc màu sắc, và nên được đặt **trực tiếp trong cốt truyện** thay vì tách biệt.
 - **Chú thích bổ sung (Annotations)** dùng để giải thích các quan sát cụ thể, nhưng thường **không được khuyến nghị** vì chúng có xu hướng gây rối mắt.

- **Màu sắc (Colors):** Việc lựa chọn màu sắc là khó khăn và cần xem xét các yếu tố như **mù màu**, các **liên kết cụ thể** của màu sắc (ví dụ: màu đỏ cho bị cấm), và sự khác biệt giữa màu in và màu trên màn hình.

3. Các Hướng dẫn Bổ sung cho Trực quan hóa Đồ thị

Các hướng dẫn thiết kế đồ họa tốt hơn bao gồm:

- a. **Hiển thị dữ liệu:** Chỉ làm nổi bật các giá trị có tầm quan trọng trung tâm đối với câu chuyện đang được kể.
- b. **Giảm sự lộn xộn (Reduce the clutter):** Tập trung vào các yếu tố thị giác cần thiết và đủ để truyền tải một thông điệp cụ thể, và **tiết kiệm** trong việc sử dụng màu sắc, độ chính xác của số, nhãn, chú thích.
- c. **Tích hợp đồ họa và văn bản:** Tiêu đề, nhãn, chú giải và chú thích thường quan trọng như chính biểu đồ; ví dụ, dữ liệu nên được gắn nhãn trực tiếp thay vì sử dụng chú giải.
- d. **Tránh "biểu đồ spaghetti" (spaghetti chart):** Việc sử dụng quá nhiều màu sắc, biểu tượng, đường, v.v., làm cho việc đọc và hiểu biểu đồ trở nên phức tạp hơn.
- e. **Sự thành công của trực quan hóa đồ họa** đòi hỏi sự kết hợp hiệu quả của nội dung, ngữ cảnh, xây dựng và thiết kế.

3.3 Bài tập

Chủ đề và câu hỏi nghiên cứu

Bài 3.1. Thảo luận về ý nghĩa của xử lý hình ảnh trong trực quan hóa dữ liệu. Nhận thức ảnh hưởng đến sự hiểu biết của chúng ta về dữ liệu như thế nào?

Bài 3.2. Thảo luận về vai trò của quá trình xử lý tiền chú ý trong trực quan hóa dữ liệu. Khả năng xử lý một số đặc tính thị giác mà không cần suy nghĩ có ý thức của não bộ ảnh hưởng như thế nào đến hiệu quả của trực quan hóa dữ liệu?

Bài 3.3. Nghiên cứu ứng dụng các nguyên tắc Gestalt trong trực quan hóa dữ liệu và thảo luận về cách Những nguyên tắc này hướng dẫn việc thiết kế hình ảnh trực quan hiệu quả.

Bài 3.4. Thảo luận về ưu điểm và nhược điểm của bảng và đồ thị trong trực quan hóa dữ liệu. Khi nào thì sử dụng cái này thích hợp hơn cái kia?

Bài 3.5. Thảo luận về cách cấu trúc và cách trình bày dữ liệu dạng bảng có thể tác động đến tính rõ ràng, tính minh bạch và khả năng hiểu sai, làm dấy lên câu hỏi về việc truyền đạt dữ liệu có trách nhiệm.

Bài 3.6. Khám phá tiềm năng kể chuyện của biểu đồ trong trực quan hóa dữ liệu. Thảo luận về cách biểu diễn đồ họa kể chuyện về dữ liệu và xem xét ý nghĩa của việc sử dụng tường thuật trực quan để truyền tải thông tin, bao gồm các câu hỏi về biểu diễn, định kiến và bản chất của việc kể chuyện.

Bài 3.7. Dựa trên Hướng dẫn trực quan hóa bảng, hãy vẽ một bảng vi phạm một số nguyên tắc thiết kế này, sau đó vẽ lại bảng đó và sửa các vi phạm trước đó. Mô tả những lợi ích trực quan của bảng đã hiệu chỉnh

Bài 3.8. Dựa trên Hướng dẫn Hiển thị Biểu đồ, hãy vẽ một biểu đồ vi phạm một số nguyên tắc thiết kế biểu đồ này, sau đó vẽ lại biểu đồ đó và sửa các lỗi đã nêu. Mô tả lợi ích trực quan của biểu đồ đã sửa.

Bài tập tính toán

Bài 3.9. Bộ tứ Anscombe là một ví dụ nổi tiếng về những hạn chế của thống kê tóm tắt và tầm quan trọng của phân tích dữ liệu trực quan. Dựa trên các đặc điểm trực quan của từng biểu đồ, hãy thảo luận về cách các điểm dữ liệu tự phân bố và tác động của việc loại bỏ các giá trị ngoại lai khỏi dữ liệu.

Bài 3.10. Đối với tập dữ liệu Iris, hãy tạo biểu đồ phân tán của bất kỳ hai biến nào (ví dụ: chiều dài cánh hoa so với chiều rộng lá dài) và áp dụng các nguyên tắc Gestalt như tính liên tục và tính khép kín để cải thiện khả năng tương tác diễn giải. Hãy cân nhắc việc thêm các đường xu hướng hoặc làm nổi bật các cụm để làm rõ các mô hình cơ bản.

Bài 3.11. Thiết kế hình ảnh trực quan tận dụng quá trình xử lý tiền chú ý. Giải thích các lựa chọn thiết kế của bạn và cách chúng phù hợp với các nguyên tắc của quá trình xử lý tiền chú ý.

Bài 3.12. Chọn một nguyên lý Gestalt và tạo một hình ảnh trực quan hóa dữ liệu minh họa cho nguyên lý này. Thảo luận lý do bạn chọn nguyên lý này và cách nó được thể hiện trong hình ảnh trực quan hóa của bạn.

Chương 4

PHƯƠNG PHÁP TRỰC QUAN HOÁ DỮ LIỆU

Chương này tập trung vào việc trình bày dữ liệu dưới định dạng hình ảnh hoặc đồ họa, sử dụng các loại biểu đồ, đồ thị, bản đồ và các yếu tố thị giác khác.

Những cách khả thi để truyền đạt thông tin như vậy. Việc truyền đạt này đòi hỏi phải sử dụng các kỹ thuật trực quan hóa và kể chuyện dữ liệu được lên kế hoạch kỹ lưỡng, những kỹ thuật này sẽ định hướng cho hoạt động âm thanh của bạn. thông qua thông điệp mà bạn muốn truyền tải. Hình ảnh hóa dữ liệu, còn được gọi là datavis hay còn gọi là hình ảnh hóa dữ liệu, là việc thể hiện dữ liệu dưới dạng hình ảnh hoặc đồ họa, sử dụng biểu đồ, đồ thị, bản đồ hoặc các yếu tố trực quan khác. Hình ảnh hóa dữ liệu quan trọng vì nhiều lý do:

- **Khám phá xu hướng trong dữ liệu:** các mô hình dữ liệu có thể không hiển thị trực tiếp với chúng ta, nhưng một hình ảnh trực quan phù hợp cho phép chúng ta xác định các mô hình và xu hướng trong dữ liệu, chẳng hạn như tương quan, giá trị cực đại hoặc cực tiểu, độ lệch và giá trị ngoại lai. Ví dụ, biểu đồ histogram cho phép chúng ta xác định định dạng của một phân phối, từ đó đưa ra kết luận về độ phẳng, độ đuôi, giá trị ngoại lai, v.v.
- **Làm cho dữ liệu phức tạp dễ hiểu:** các biểu diễn dữ liệu dạng bảng hoặc văn bản có thể khó diễn giải và hình ảnh hóa có thể đơn giản hóa sự phức tạp đó bằng cách Gửi dữ liệu dưới dạng đồ họa. Ví dụ, biểu đồ phân tán cho phép chúng ta quan sát mối tương quan giữa các biến và sự hiện diện của các cụm đối tượng.
- **Cung cấp góc nhìn về dữ liệu:** Hình ảnh trực quan có thể mang đến một số góc nhìn chưa rõ ràng, bao gồm góc nhìn đa chiều hoặc góc nhìn theo thời gian về dữ

liệu, cho phép chúng ta so sánh các biến, tìm mối quan hệ và hiểu bối cảnh. Ví dụ: biểu đồ đường cho phép chúng ta quan sát tính thời vụ và các xu hướng khác trong dữ liệu.

- **Tiết kiệm thời gian phân tích:** biểu diễn dữ liệu trực quan có thể cho phép chúng ta phát hiện các giá trị ngoại lai, tìm cụm, xác định xu hướng, v.v., tất cả mà không cần phải chạy các thuật toán học phức tạp.
- **Cho phép chúng tôi kể một câu chuyện (kể chuyện dữ liệu):** trực quan hóa dữ liệu là một công cụ mạnh mẽ để kể chuyện, vì nó cho phép trình bày dữ liệu trong một câu chuyện hấp dẫn, giúp truyền đạt hiểu biết và phát hiện tới khán giả dễ dàng hơn nhiều.

4.1 Phân phối (Distributions)

Phần này giới thiệu các phương pháp đồ họa được sử dụng để khám phá và trực quan hóa các mẫu và xu hướng trong một biến dữ liệu đã cho. Bằng cách xem xét phân phối dữ liệu, người ta có thể xác định các đặc điểm quan trọng của dữ liệu như: tổng quan mẫu, trung bình, phân tán, hình dạng, ngoại lai, median, mode, độ lệch, độ nhọn.

Mô tả ba loại biểu đồ trực quan hóa dữ liệu chính được sử dụng để phân tích các phân phối: **Histogram**, **Boxplot** và **Violin Plot**.

4.1.1 Histogram (Biểu đồ Tần suất)

- **Mục đích:** Khám phá và trực quan hóa sự phân phối của một biến.
- **Kiểu dữ liệu phổ biến:** dữ liệu liên tục.
- **Điễn giải:** biểu đồ phân chia phạm vi dữ liệu thành các ngăn và trình bày tần suất
- Tần suất xuất hiện (số lượng) của mỗi bin theo chiều cao của nó, cung cấp cái nhìn tổng quan về phân phối của biến. Do đó, việc diễn giải biểu đồ histogram bao gồm việc đánh giá mô hình tổng thể, tâm, độ phân tán, hình dạng, giá trị ngoại lai, khoảng trống, chế độ, độ lệch và độ nhọn của nó.

4.1.2 Boxplot (Biểu đồ Hộp và Râu - Box and Whisker Plot)

- **Mục đích:** để khám phá và hình dung sự phân phối của một biến hoặc để so sánh sự phân phối của các biến khác nhau, sử dụng tóm tắt năm số, là phần mở rộng của tứ phân vị để bao gồm các giá trị nhỏ nhất và lớn nhất của biến: min, Q_1 , Q_2 , Q_3 , max.
- **Kiểu dữ liệu phổ biến:** dữ liệu liên tục.
- **Điễn giải:** Biểu đồ hộp là một biểu đồ thể hiện giá trị nhỏ nhất, tứ phân vị thứ nhất (Q_1), trung vị (Q_2), tứ phân vị thứ ba (Q_3) và giá trị lớn nhất của một biến, cùng với các giá trị ngoại lai. Do đó, biểu đồ hộp cho phép chúng ta quan sát phân phối dữ liệu, hình dạng của nó và sự hiện diện của các giá trị ngoại lai.
- **Ví dụ về ứng dụng:** để trực quan hóa sự phân bố của bất kỳ loại biến (liên tục) nào, chẳng hạn như chiều cao, khoảng cách, cân nặng, nhiệt độ, huyết áp, điểm thi, mức thu nhập, v.v.

Một đồ thị được gọi là biểu đồ hộp, hay biểu đồ hộp và râu, được thiết kế bằng cách sử dụng một hộp trong khoảng liên tứ phân vị (IQR), một đường ngang trên trung vị và râu đi từ hộp đến các giá trị tối thiểu và tối đa. Biểu đồ hộp làm nổi bật các tứ phân vị trong hộp và các giá trị tối thiểu/tối đa ở các cực của râu (*whiskers*).

Biểu đồ hộp là một công cụ mạnh mẽ để xác định các giá trị ngoại lai trong một biến duy nhất. Các giá trị lớn hơn $Q_3 + \gamma IQR$ hoặc nhỏ hơn $Q_1 - \gamma IQR$, trong đó γ là bội số IQR được xác định trước, là được coi là các bất thường và được vẽ bên ngoài râu. Hầu hết các gói phần mềm sử dụng $\gamma = 1,5$ làm giá trị chuẩn để xác định các giá trị ngoại lệ. Các giá trị được coi là **bất thường** (*anomalies*) hoặc **ngoại lai** (*outliers*) được vẽ bên ngoài râu, thường được hiển thị dưới dạng các chấm nhỏ.

Hình 4.1 hiển thị năm trường hợp ngoại lệ dưới dạng các chấm nhỏ được vẽ phía trên và phía dưới râu, tức là các giá trị cực đại và cực tiểu trong biểu đồ hộp. Biểu đồ tần suất cũng hiển thị ngưỡng ngoại lệ trên và dưới dưới dạng các đường thẳng đứng đứt nét.

```
[8]: # Box plot and histogram for a normal distribution

import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as spy
import seaborn as sns
```

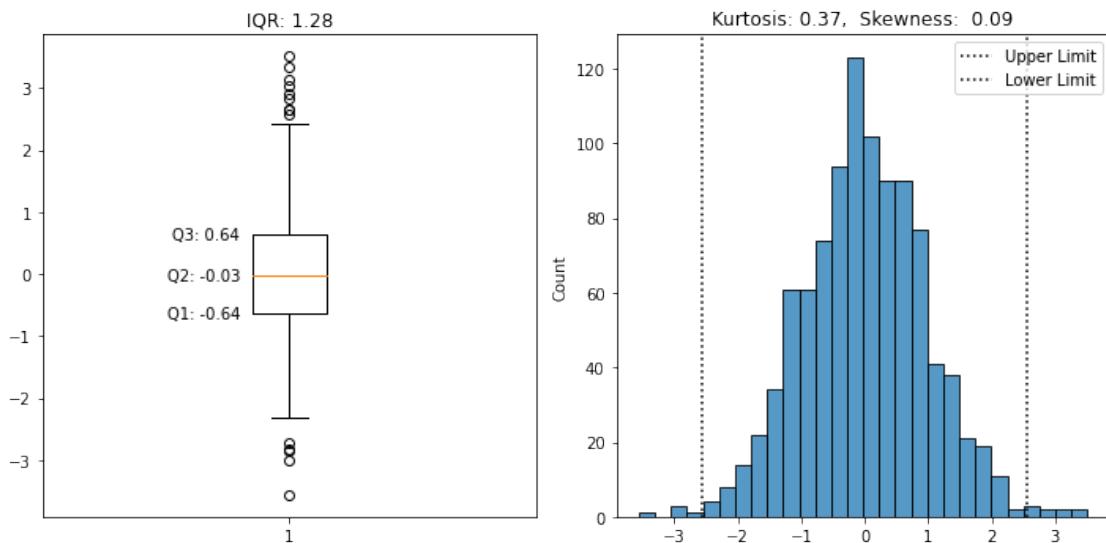
```
# Generate the random sample data
np.random.seed()
data = np.random.normal(loc=0, scale=1, size=1000)

# Calculate the summary measures
q1, q2, q3 = np.percentile(data, [25, 50, 75])
iqr = q3 - q1
upper_whisker = q3 + 1.5*iqr
lower_whisker = q1 - 1.5*iqr
max_val = np.max(data)
min_val = np.min(data)
midpoint = (max_val+min_val)/2
k = spy.kurtosis(data)
s = spy.skew(data)

# Plot the boxplot and print the values (Q1, Q2, Q3) on the
# first subplot
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(10,5))
bp = ax1.boxplot(data)
ax1.set_title(f"IQR: {iqr:.2f}")
ax1.text(0.9, q1, f'Q1: {q1:.2f}', ha='right', va='center')
ax1.text(0.9, q2, f'Q2: {q2:.2f}', ha='right', va='center')
ax1.text(0.9, q3, f'Q3: {q3:.2f}', ha='right', va='center')

# Plot the histogram on the second subplot
sns.histplot(data, bins='auto', ax=ax2)
plt.axvline(x=upper_whisker, color='k', linestyle=':', 
            label='Upper Limit')
plt.axvline(x=lower_whisker, color='k', linestyle=':', 
            label='Lower Limit')
ax2.set_ylabel('Count')
plt.title(f"Kurtosis: {k:.2f}, Skewness: {s: .2f}")

# Display the plots
plt.tight_layout()
plt.legend(); plt.show()
```



Hình 4.1: Histogram và Boxplot.

4.1.3 Violin Plot (Biểu đồ Violin)

Violin Plot được giới thiệu là một trong những phương pháp trực quan hóa phân phối dữ liệu.

- **Mục đích:** để khám phá và trực quan hóa sự phân bố của một biến hoặc để so sánh sự phân bố của các biến khác nhau, bằng cách kết hợp biểu đồ hộp với biểu đồ mật độ hạt nhân.
- **Kiểu dữ liệu phổ biến:** dữ liệu liên tục.
- **Điễn giải:** các khu vực rộng hơn trong biểu đồ violin biểu thị mật độ điểm dữ liệu lớn hơn, các vùng có diện tích nhỏ có nghĩa là ít điểm dữ liệu hơn, và các giá trị ngoại lai xuất hiện bên ngoài thân violin. Do đó, biểu đồ violin cho phép chúng ta quan sát phân bố dữ liệu, hình dạng của nó và sự hiện diện của các giá trị ngoại lai.
- **Ví dụ về ứng dụng:** để trực quan hóa sự phân bố của bất kỳ loại biến (liên tục) nào, chẳng hạn như chiều cao, khoảng cách, cân nặng, nhiệt độ, huyết áp, điểm thi, mức thu nhập, v.v.

```
[26]: # Comparing the Box and Violin plots for the Sepal length of
# the Iris dataset grouped by plant species
```

```
import seaborn as sns
```

```

import matplotlib.pyplot as plt
from sklearn.datasets import load_iris

diris = load_iris() # Load the Iris dataset from Scikit-learn

# Boxplot with the data points
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(12, 6))
sns.boxplot(x=diris.target, y=diris.data[:,0], ax=axs[0],
             sym='', width=0.6,
             boxprops=dict(edgecolor='black'),
             whiskerprops=dict(color='black', linestyle='-' ),
             medianprops=dict(color='black'),
             capprops=dict(color='black', linestyle='-' ))
sns.stripplot(x=diris.target, y=diris.data[:,0], ax=axs[0],
               color='black')
axs[0].set_xticklabels(diris.target_names)
axs[0].set_ylabel('Sepal Length (cm)')
axs[0].set_ylim([3.6, 8.6])
axs[0].set_title('Boxplot with Data Points')

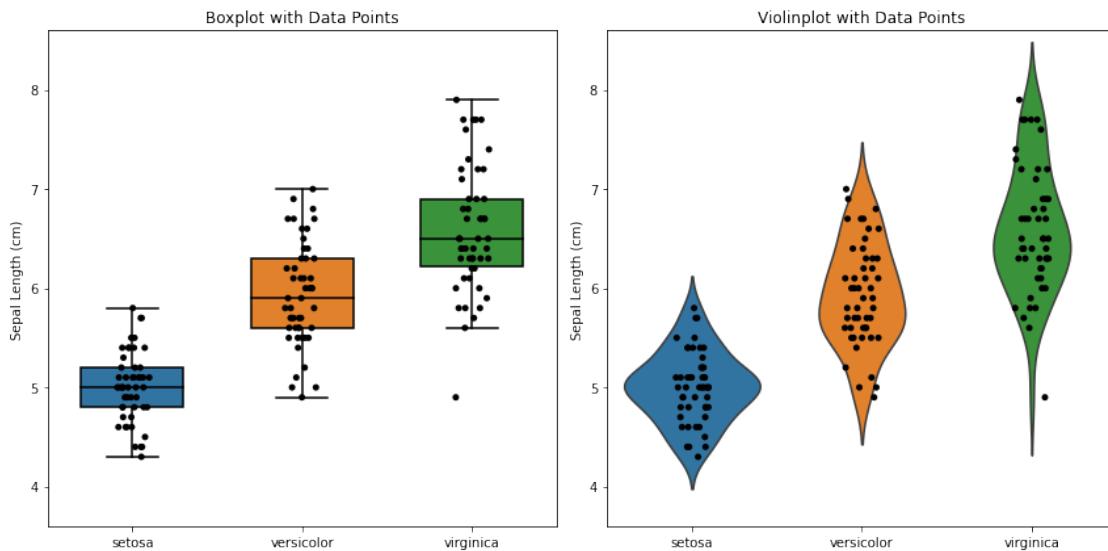
# Violinplot with the data points
sns.violinplot(x=diris.target, y=diris.data[:,0], ax=axs[1],
                 inner=None)
sns.stripplot(x=diris.target, y=diris.data[:,0], ax=axs[1],
               jitter=True, color='black')
axs[1].set_xticklabels(diris.target_names)
axs[1].set_ylabel('Sepal Length (cm)')
axs[1].set_ylim([3.6, 8.6])
axs[1].set_title('Violinplot with Data Points')

# Show the plot
plt.tight_layout()
plt.show()

```

4.2 Các Mối liên hệ (Associations)

Trong nhiều tình huống, cần thiết phải tạo ra các đồ thị cho phép chúng ta phân tích **mối quan hệ giữa hai hoặc nhiều biến**. Đôi khi các đồ thị này chỉ hiển thị giá trị của biến này liên quan đến biến khác, và đôi khi chúng hiển thị một thước đo được tính toán cho cả hai biến, ví dụ, **mối tương quan** của chúng.



Hình 4.2: Violin plot và Boxplot.

Các đồ thị mối liên hệ được trình bày trong mục này là **scatterplot** (biểu đồ phân tán), **bubble chart** (biểu đồ bong bóng), **scatterplot matrix** (ma trận biểu đồ phân tán), và **heatmaps** (bản đồ nhiệt).

4.2.1 Scatter Plot (Biểu đồ phân tán)

- **Mục đích:** Khám phá và trực quan hóa mối quan hệ giữa các biến.
- **Kiểu dữ liệu phổ biến:** dữ liệu liên tục trên cả hai trục và dữ liệu phân loại theo màu sắc.
- **Điễn giải:** phân tích loại và cường độ của mối quan hệ giữa hai biến Ví dụ, nó có thể xác định xem có bất kỳ loại tương quan nào (tương quan dương yếu/mạnh, tương quan âm hoặc không có tương quan) giữa các biến hay không, và xác định xu hướng, mô hình và thay đổi trong dữ liệu. Vì nó biểu diễn một biến so với một biến khác, nó cũng là một công cụ hữu ích để xác định các cụm (nhóm) dữ liệu và các giá trị ngoại lai. Một biến thứ ba, thường là biến phân loại, có thể được đưa vào biểu đồ phân tán bằng cách sử dụng các màu khác nhau cho các chấm.
- **Ví dụ ứng dụng:** Biểu đồ phân tán có thể được sử dụng để trực quan hóa mối quan hệ giữa:
 - Giá cổ phiếu và các chỉ số tài chính.

- Các phép đo sinh lý và chất lượng cuộc sống.
- Mối quan hệ giữa biểu hiện gen và bệnh tật.
- Các chỉ số kinh tế xã hội và trình độ học vấn.
- Hành vi và sở thích của khách hàng.

4.2.2 Bubble Chart (Biểu đồ Bong bóng)

- **Mục đích:** để hình dung mối quan hệ (liên kết) giữa ba hoặc bốn biến số.
- **Kiểu dữ liệu phổ biến:** dữ liệu liên tục trên cả hai trục và kích thước bong bóng, và dữ liệu phân loại dữ liệu trong màu sắc.
- **Điễn giải:** Biểu đồ bong bóng là phần mở rộng của biểu đồ phân tán với các chấm (bong bóng) có kích thước khác nhau nên cách diễn giải của nó tuân theo cách diễn giải của biểu đồ phân tán, thêm kích thước chấm làm giá trị của biến thứ ba hoặc thứ tư.
- **Ví dụ về ứng dụng:** để hình dung mối quan hệ giữa ba hoặc bốn biến, như giá cả so với chất lượng và thị phần của một sản phẩm nhất định; mối quan hệ giữa thu nhập, trình độ học vấn và độ tuổi; mức tiêu thụ nhiên liệu so với khả năng tăng tốc, số lượng xi-lanh và mã lực của một chiếc xe; và tuổi thọ so với chỉ số phát triển con người, lục địa và mức tiêu thụ CO₂, v.v.

```
[28]: # Bubble charts with four variables for the Gapminder and Auto
      MPG datasets

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
dgapminder = pd.read_csv('gapminder_data_graphs.csv')

# Filter out missing values in 'life_exp', 'hdi_index', and
# 'co2_consumption'
dgapminder = dgapminder.dropna(subset=['life_exp', 'hdi_index',
                                         'co2_consump'])

# Set plot features
sns.set_style("whitegrid")
```

```

fig, axs = plt.subplots(nrows=2, ncols=1, figsize=(10, 12))

# Create a bubble chart for the Gapminder dataset
sns.scatterplot(data=dgapminder, x="hdi_index", y="life_exp",
                 hue="continent",
                 size="co2_consump", sizes=(20, 500), alpha=0.7,
                 ax=axs[0])
axs[0].set_xlabel("HDI Index", fontsize=12)
axs[0].set_ylabel("Life Expectancy", fontsize=12)
axs[0].set_title("Life Expectancy vs HDI Index by Continent
                  (Bubble size is CO2 Consumption)",
                  fontsize=14)
axs[0].legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

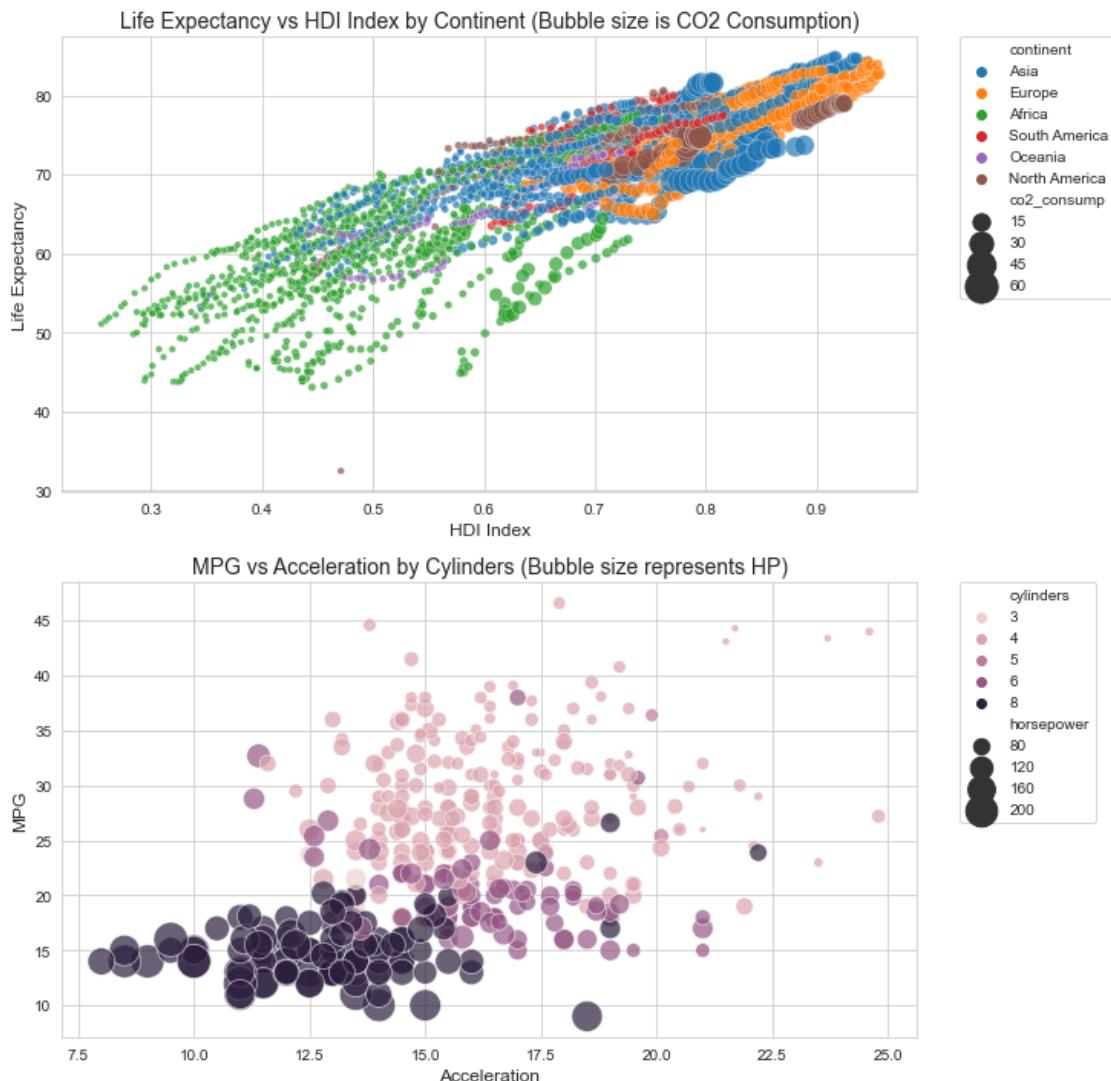
# Create a bubble chart for the Auto MPG dataset
dmpg = pd.read_csv('mpg.csv')
sns.scatterplot(data=dmpg, x="acceleration", y="mpg",
                 hue="cylinders",
                 size="horsepower", sizes=(20, 500), alpha=0.7,
                 ax=axs[1])
axs[1].set_xlabel("Acceleration", fontsize=12)
axs[1].set_ylabel("MPG", fontsize=12)
axs[1].set_title("MPG vs Acceleration by Cylinders (Bubble size
                  represents HP)",
                  fontsize=14)
axs[1].legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

# Show the plot
plt.show()

```

4.2.3 Scatterplot Matrix Plot (Ma trận biểu đồ phân tán)

- **Mục đích:** để khám phá và hình dung mối quan hệ (liên kết) giữa nhiều biến cùng một lúc.
- **Kiểu dữ liệu phổ biến:** dữ liệu liên tục.
- **Điễn giải:** Trong ma trận phân tán, các biểu đồ đường chéo thường là biểu đồ histogram thể hiện phân phối của từng biến, và các biểu đồ lệch đường chéo là biểu đồ phân tán từng cặp thể hiện mối quan hệ giữa từng cặp biến. Do đó, ma trận này có thể được sử dụng để phân tích loại và cường độ mối quan hệ giữa



Hình 4.3: Scatter plot và Bubble chart.

nhiều biến, ví dụ, để xác định xem có bất kỳ loại tương quan nào (tương quan dương yếu/mạnh, tương quan âm hoặc không có tương quan) giữa các biến, cũng như để phát hiện xu hướng, mô hình và thay đổi trong dữ liệu. Vì ma trận này biểu diễn tất cả các biến so với tất cả các biến, nên nó cũng là một công cụ hữu ích để xác định các cụm (nhóm) dữ liệu và các giá trị ngoại lai.

- **Ví dụ về ứng dụng:** để hình dung mối quan hệ giữa giá cổ phiếu và các chỉ số tài chính, phép đo sinh lý và chất lượng cuộc sống, các chỉ số kinh tế xã hội và trình độ học vấn, v.v.

```
[41]: # Scatterplot matrix for the Iris dataset

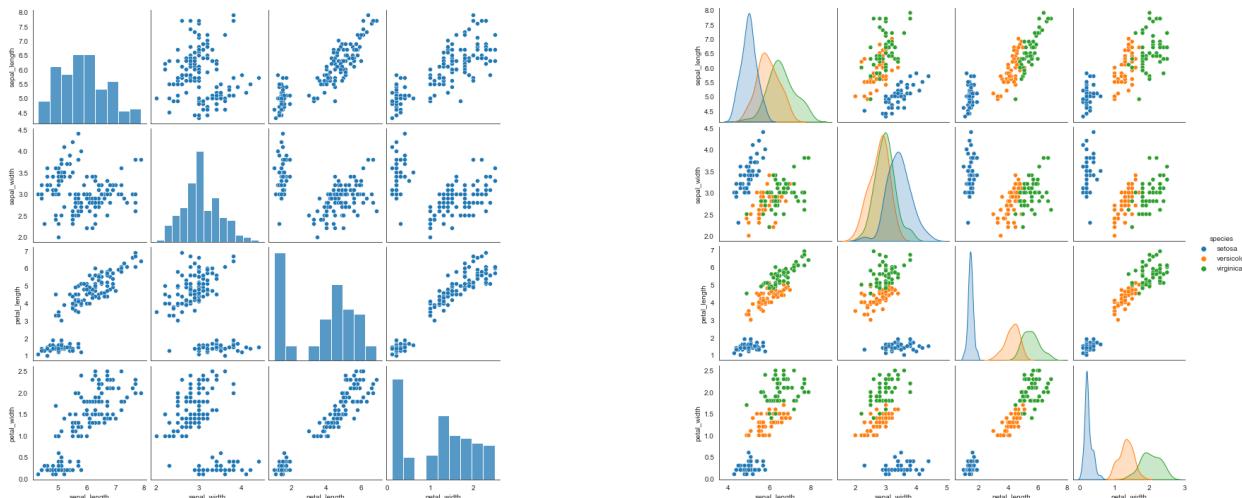
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris

# Load the Iris dataset from scikit-learn
iris = load_iris()

# Convert the dataset to a Pandas DataFrame
diris = sns.load_dataset('iris')

# Pairplot
sns.set_style("white")
sns.pairplot(diris) # Distributions
sns.pairplot(diris, hue='species') # Kernel density estimate
# → (KDE)

# Show the plot
plt.show()
```



Hình 4.4: Scatter matrix plot.

4.2.4 Heatmaps và Correlograms (Bản đồ nhiệt và Biểu đồ tương quan)

- **Mục đích:** sử dụng màu sắc để khám phá và hình dung độ lớn của các biến số. Nếu các độ lớn này là mối tương quan giữa các biến trong một tập dữ liệu cho thấy loại và cường độ mối quan hệ của chúng, thì bản đồ nhiệt được gọi là biểu đồ tương quan.
- **Kiểu dữ liệu phổ biến:** dữ liệu số, dữ liệu rời rạc hoặc dữ liệu liên tục.
- **Điễn giải:** Chúng ta thường sử dụng một dải màu để biểu diễn các giá trị của dữ liệu, trong đó màu tối hơn biểu thị giá trị cao hơn và màu sáng hơn biểu thị giá trị thấp hơn. Nó có thể được sử dụng để phân tích loại (hướng) và cường độ của mối quan hệ giữa các biến, ví dụ, để xác định xem có bất kỳ loại tương quan nào (yếu/mạnh) (tương quan dương, âm hoặc không có tương quan) giữa các biến. Tương quan dương chỉ ra rằng hai biến có xu hướng tăng hoặc giảm cùng nhau, trong khi tương quan âm mối quan hệ chỉ ra rằng khi một biến tăng, biến kia có xu hướng giảm.
- **Ví dụ về ứng dụng:** để hình dung mối quan hệ giữa giá cổ phiếu và tài chính, các chỉ số xã hội, các phép đo sinh lý và chất lượng cuộc sống, mối quan hệ của các gen biểu hiện và bệnh tật, chỉ số kinh tế xã hội và trình độ học vấn, khách hàng hành vi và sở thích, v.v.

```
[47]: # Correlation heatmaps (correlograms) for the Iris and Forest
      -Fires datasets

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
import pandas as pd
import numpy as np

# Load the Iris dataset from scikit-learn
iris = load_iris()
diris = sns.load_dataset('iris')

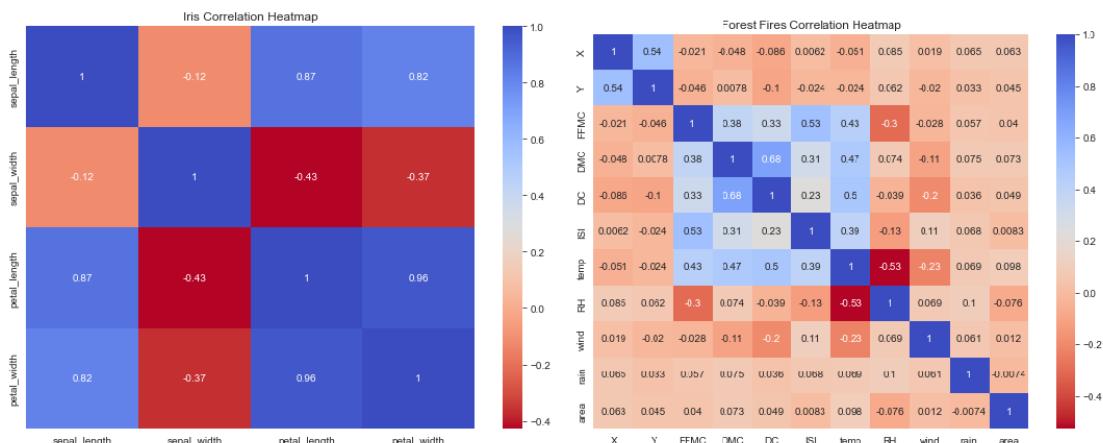
# Load the Forest Fires dataset
# https://archive.ics.uci.edu/ml/datasets/forest+fires
dforest = pd.read_csv('forestfires.csv')
```

```
# Create a subplot grid with 2 rows and 1 column
fig, axs = plt.subplots(2, 1, figsize=(8, 12))

# Correlation heatmap for Iris dataset
corr_iris = diris.corr()
cmap_inverted = sns.color_palette("coolwarm", as_cmap=True)
cmap_inverted = cmap_inverted.reversed()
sns.heatmap(corr_iris, annot=True, cmap=cmap_inverted,
            ax=axs[0])
axs[0].set_title('Iris Correlation Heatmap')

# Correlation heatmap for the Forest Fires dataset
corr_forest = dforest.corr()
sns.heatmap(corr_forest, annot=True, cmap=cmap_inverted,
            ax=axs[1])
axs[1].set_title('Forest Fires Correlation Heatmap')

# Show the plots
plt.tight_layout()
plt.show()
```



Hình 4.5: Heatmap và Correlograms.

4.3 Số lượng (Amounts)

Trong nhiều tình huống, mục tiêu là trực quan hóa **giá trị của một biến cụ thể**, ví dụ: kích thước của một mặt hàng, giá của một cổ phiếu, hoặc số lượng huy chương trong

một giải vô địch. Trong tất cả các trường hợp này, có những danh mục cụ thể đang được nghiên cứu (như kích thước, giá cả, giải thưởng).

Những tình huống này được gọi là **số lượng** (*amounts*) vì mục đích là trực quan hóa các giá trị. Hai loại đồ thị sẽ được xem xét trong danh mục này là **biểu đồ thanh** (*bar charts*) và **biểu đồ mạng nhện** (*radar charts*).

4.3.1 Bar Chart (Biểu đồ thanh)

- **Mục đích:** để trực quan hóa dữ liệu có thể được nhóm thành các danh mục hoặc nhóm riêng biệt và để so sánh quy mô của các biến hoặc danh mục khác nhau.
- **Kiểu dữ liệu phổ biến:** dữ liệu phân loại hoặc dữ liệu rời rạc.
- **Điều giải:** phân tích quy mô tương đối của các loại hoặc nhóm khác nhau, xác định xu hướng, mô hình và thay đổi trong dữ liệu và so sánh nhiều biến số, các khả năng hoặc chuỗi dữ liệu.
- **Ví dụ về ứng dụng:** để trực quan hóa mức độ bán hàng của các danh mục sản phẩm khác nhau, kết quả khảo sát, dữ liệu nhân khẩu học, dữ liệu tài chính, v.v.

Biểu đồ thanh tương tự như biểu đồ đường, ngoại trừ việc mỗi điểm dữ liệu được thay thế bằng một hình chữ nhật có chiều cao tỷ lệ thuận với giá trị. Hình chữ nhật thường được căn giữa vào thuộc tính không gian của dữ liệu và chiều rộng của nó thường đồng đều. Khi các giá trị mang tính phân loại hoặc rời rạc và không thể hiển thị theo chuỗi, biểu đồ thanh có thể là một lựa chọn thay thế phù hợp cho biểu đồ đường.

Đặc điểm thị giác: Chiều dài (*length*) là một thuộc tính tiền chú ý (*preattentive feature*) thường được sử dụng trong biểu đồ thanh; thanh dài hơn có thể biểu thị các giá trị hoặc số lượng lớn hơn.

So sánh với Biểu đồ tròn: Biểu đồ thanh hiển thị **tần suất tương đối** của từng danh mục biến, giúp người đọc nhận biết sự khác biệt một cách chính xác hơn so với biểu đồ tròn, bởi vì nhận thức của con người dễ dàng xử lý khoảng cách (chiều cao thanh) hơn là diện tích (lát cắt tròn).

```
[271]: # Create the bar charts for Severity and Shape vs Severity
# of the mammographic dataset

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

```

# Load the dataset into a pandas DataFrame
dmammo = pd.read_csv('mammographic_masses_nominal.csv')

# 1. Calculate and plot the bar chart for 'Severity'
counts = dmammo['Severity'].value_counts()
counts = counts.sort_index()
plt.bar(counts.index, counts.values) # Plot the bar chart
plt.title('Bar chart: Severity count')
plt.ylabel('Count')
plt.show()

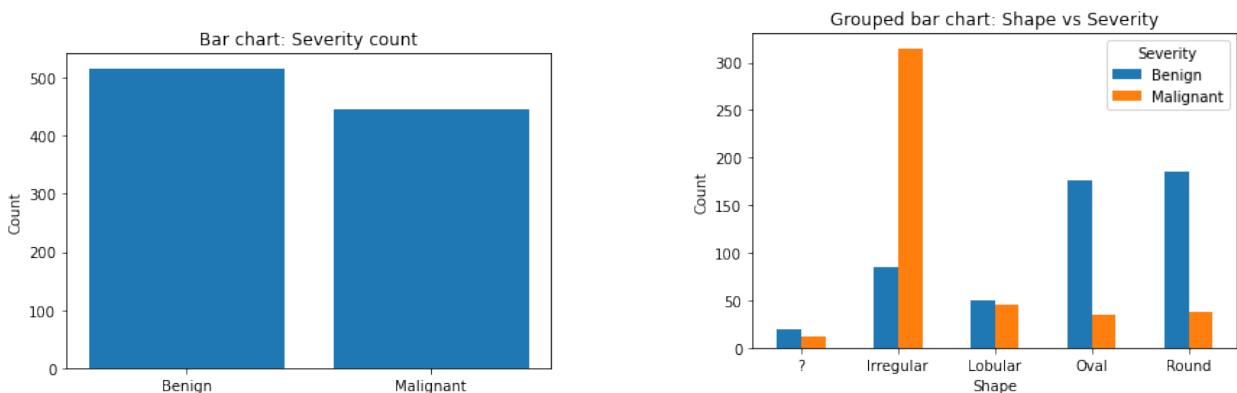
# 2. Plot a bar chart for 'Shape' in relation to 'Severity'
dmammo.groupby(['Shape', 'Severity']).size().unstack() .
    plot(kind='bar', rot=0)
plt.title('Grouped bar chart: Shape vs Severity')
plt.ylabel('Count')
plt.show()

# 3. Calculate and plot the bar chart for 'Shape' & 'Severity' vs
# 'Severity'
counts = dmammo.groupby(['Shape', 'Severity']).size() .
    reset_index(name='count')
plt.bar(range(len(counts)), counts['count']) # Plot the bar
    chart
plt.title('Bar chart: (Shape & Severity) vs Count')
plt.xlabel('& '.join([var.replace('_', ' ') .title() for var in
    ['Shape', 'Severity']] ))
plt.ylabel('Count')
plt.xticks(range(len(counts)), [', ' .join(map(str, tpl)) for tpl
    in
        counts[['Shape', 'Severity']] .
    to_records(index=False)])
plt.xticks(rotation=90)
plt.show()

```

4.3.2 Radar Chart (Biểu đồ Radar)

- **Mục đích:** để so sánh và trực quan hóa nhiều chuỗi biến liên tục trong một biểu đồ bán kính.
- **Kiểu dữ liệu phổ biến:** dữ liệu liên tục.



Hình 4.6: Biểu đồ thanh.

- **Điễn giải:** Biểu đồ radar vẽ một hoặc nhiều chuỗi giá trị có tính đến các biến số trên một đồ thị xuyên tâm, tạo thành một đa giác. Trong biểu đồ radar, các giá trị lớn hơn được thể hiện xa hơn tâm radar, cho phép so sánh các chuỗi khác nhau về biên độ. Hơn nữa, hình dạng và kích thước của đa giác có thể được sử dụng để so sánh các chuỗi khác nhau.
- **Ví dụ về các ứng dụng:** để so sánh các thương hiệu xem xét các tính năng của chúng, cổ phiếu liên quan đến các chỉ số (cơ bản) của chúng, học sinh liên quan đến điểm số của họ ở mỗi môn học, động vật liên quan đến đặc điểm của chúng, v.v.

```
[48]: # Radar Chart for the Iris dataset available at the Scikitlearn
       library

import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import load_iris

diris = load_iris() # Load the Iris dataset from Scikitlearn

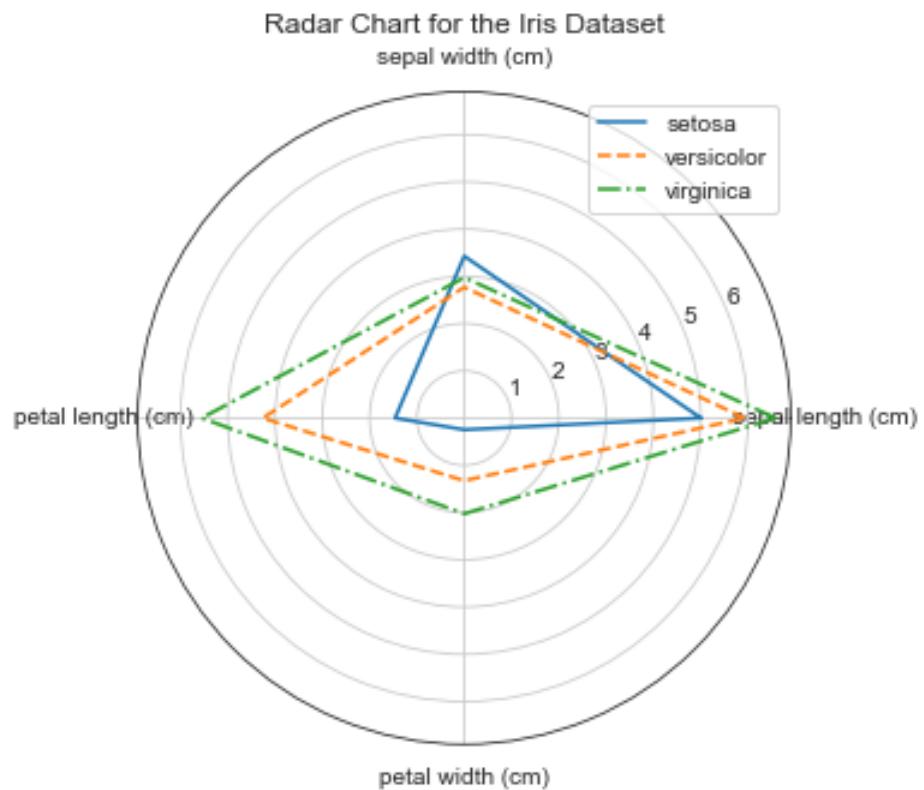
# Setup the radar chart figure
fig = plt.figure(figsize=(10, 5))
axs = fig.add_subplot(polar=True)

# Calculate the angles for the radar chart
angles = np.linspace(0, 2*np.pi, len(diris.feature_names),
                     endpoint=False)
angles = np.concatenate((angles, [angles[0]]))
```

```
# Linestyle for each Iris class
line = ['-', '--', '-.']

# For each Iris class, plot the mean values of its features as a
# line
for i in range(3):
    values = diris.data[diris.target==i].mean(axis=0)
    values = np.concatenate((values, [values[0]]))
    axs.plot(angles, values, label=diris.target_names[i],
             ls=line[i])
plt.xticks(angles[:-1], diris.feature_names)
axs.legend()
axs.set_title('Radar Chart for the Iris Dataset')

# Display the plot
plt.show()
```



Hình 4.7: Radar Chart.

4.4 Tỷ lệ (Proportions)

Mục tiêu của trực quan hóa tỷ lệ là điều tra cách một nhóm hoặc một phần có thể được chia thành các nhóm con đại diện cho **các phần của một tổng thể**, hay còn gọi là **tỷ lệ**.

Các phương pháp trực quan hóa tỷ lệ được xem xét trong mục này là **biểu đồ tròn** (*pie charts*), **biểu đồ vòng** (*doughnut charts*), và **biểu đồ hình cây** (*treemaps*).

4.4.1 Pie Chart (Biểu đồ Tròn)

- **Mục đích:** khám phá và hình dung sự phân bố của một biến có thể được chia thành một số ít các danh mục khác nhau và loại trừ lẫn nhau, trong đó mỗi danh mục là một phần (lát cắt) của tổng thể.
- **Kiểu dữ liệu phổ biến:** dữ liệu phân loại.
- **Điễn giải:** mỗi lát cắt biểu đồ hình tròn đại diện cho một danh mục trong tổng thể và lát cắt càng lớn thì giá trị (tỷ lệ) của nó càng lớn.
- **Ví dụ về ứng dụng:** để trực quan hóa tỷ lệ của bất kỳ biến phân loại nào, chẳng hạn như màu sắc trong một bức tranh, khách hàng mua từng loại sản phẩm, tình trạng hôn nhân ở một khu vực nhất định, v.v.

Biểu đồ tròn và biểu đồ thanh được nhóm là các kỹ thuật trực quan hóa cho thấy khả năng của chúng ta trong việc xác định chính xác hơn sự khác biệt về đóng góp của từng danh mục.

So sánh với Biểu đồ thanh: Nhận thức của con người xử lý **khoảng cách dễ dàng hơn** so với diện tích (lát cắt tròn), do đó, sự khác biệt về chiều cao trong biểu đồ thanh được nhận thấy chính xác hơn so với sự khác biệt giữa các lát cắt tròn.

```
[36]: # Pie Chart and Grouped Bar Chart for variables 'Margin' of the
# Mammographic dataset, and 'Day' of the Forest Fires dataset
# Code with percentage values for the grouped bar chart

import pandas as pd
import matplotlib.pyplot as plt

# Load the datasets
dmammo = pd.read_csv('mammographic_masses_nominal.csv',
                     na_values=['?'])
```

```

dforest = pd.read_csv('forestfires.csv')

# Create a figure with two subplots in each row
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))

# Mammographic dataset - Pie chart
mm = dmammo['Margin'].value_counts()
mm.plot(kind='pie', autopct='%1.1f%%', startangle=90,
        ax=axs[0,0], fontsize=12)
axs[0,0].set_title('Margin', fontsize=14); axs[0,0].set_ylabel('')

# Mammographic dataset - Grouped bar chart
mm = dmammo['Margin'].value_counts(normalize=True) * 100
mm.plot(kind='bar', ax=axs[0,1], color="#1f77b4", fontsize=12)
axs[0,1].grid(False); axs[0,1].legend().remove()
axs[0,1].set_ylabel('Percentage', fontsize=12)

# Forest Fires dataset - Pie chart
custom_order = ['sun', 'sat', 'fri', 'thu', 'wed', 'tue', 'mon']
ff = dforest['day'].value_counts()
ff = ff[custom_order]
ff.plot(kind='pie', autopct='%1.1f%%', startangle=90,
        ax=axs[1,0], fontsize=12)
axs[1,0].set_title('Day', fontsize=14); axs[1,0].set_ylabel('')

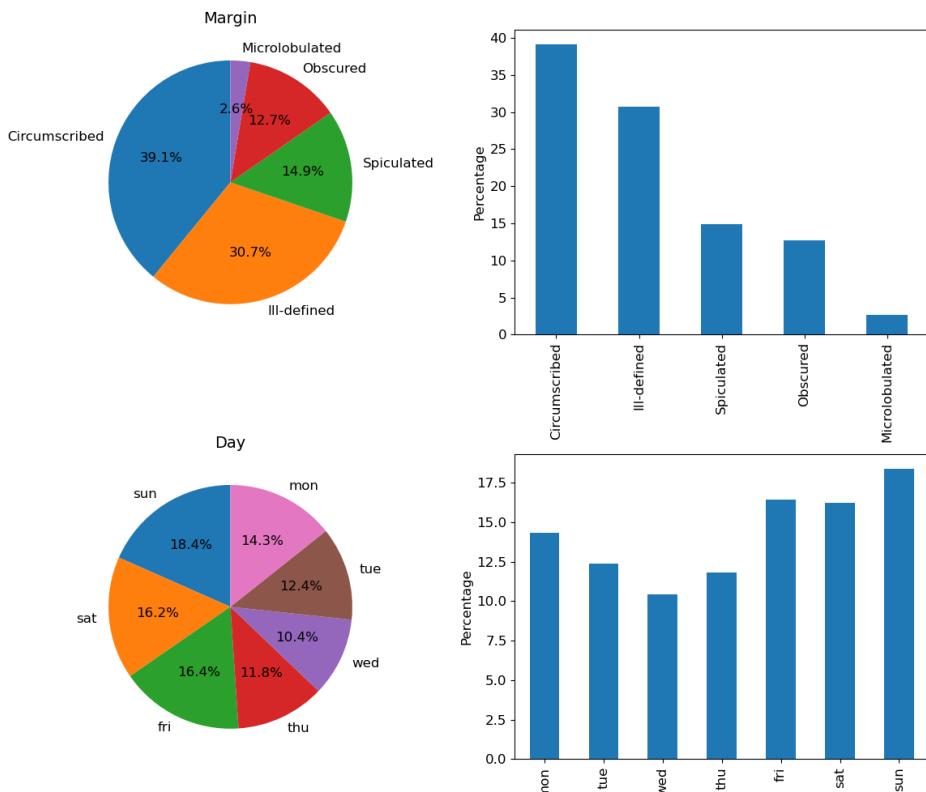
# Forest Fires dataset - Grouped bar chart
custom_order = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun']
ff = dforest['day'].value_counts(normalize=True) * 100
ff = ff[custom_order]
ff.plot(kind='bar', ax=axs[1,1], color="#1f77b4", fontsize=12)
axs[1,1].grid(False); axs[1,1].legend().remove()
axs[1,1].set_ylabel('Percentage', fontsize=12)

plt.tight_layout()
plt.show()

```

4.4.2 Doughnut Chart (Biểu đồ Vòng)

- Biểu đồ vòng là một **biến thể của biểu đồ tròn** trong đó tâm của hình tròn bị loại bỏ, tạo thành một biểu đồ dạng vòng.



Hình 4.8: Pie Chart.

- Do đó, **mục đích, loại dữ liệu, cách diễn giải** và các ví dụ ứng dụng của biểu đồ vòng đều **giống như biểu đồ tròn**.

```
[1]: # Doughnut chart for variables 'Margin' of the Mammographic
# dataset,
# and 'Day' of the Forest Fires dataset
# Code with percentage values for the grouped bar chart

import pandas as pd
import matplotlib.pyplot as plt

# Load the datasets
dmammo = pd.read_csv('mammographic_masses_nominal.csv',
na_values=['?'])
dforest = pd.read_csv('forestfires.csv')

# Create a figure with two subplots in each row
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(12, 10))
```

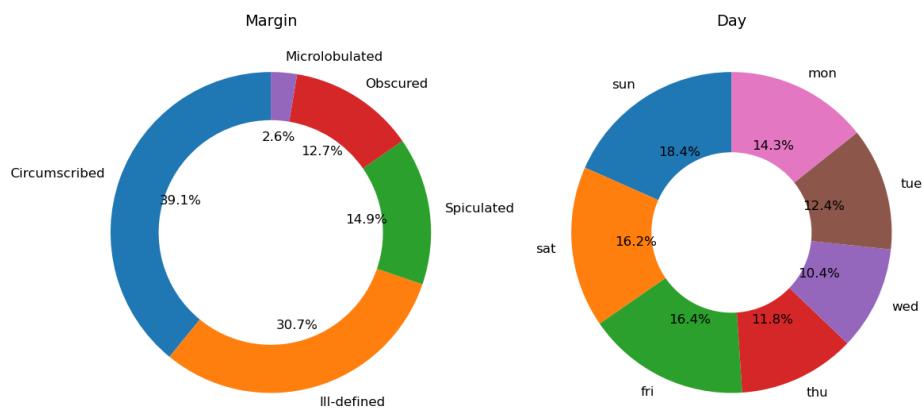
```

# Mammographic dataset - Doughnut chart
mm = dmammo['Margin'].value_counts()
mm.plot(kind='pie', autopct='%1.1f%%', startangle=90, ax=axs[0],
         wedgeprops={'width': 0.3}, fontsize=12)
axs[0].set_title('Margin', fontsize=14); axs[0].set_ylabel('')

# Forest Fires dataset - Doughnut chart
custom_order = ['sun', 'sat', 'fri', 'thu', 'wed', 'tue', 'mon']
ff = dforest['day'].value_counts()
ff = ff[custom_order]
ff.plot(kind='pie', autopct='%1.1f%%', startangle=90, ax=axs[1],
         wedgeprops={'width': 0.5}, fontsize=12)
axs[1].set_title('Day', fontsize=14); axs[1].set_ylabel('')

plt.tight_layout()
plt.show()

```



Hình 4.9: Donut Chart.

4.4.3 Treemap (Biểu đồ Hình cây)

- **Mục đích:** để khám phá và trực quan hóa dữ liệu phân cấp trong các kiểu lồng nhau và tương hỗ. Các hình chữ nhật độc quyền có kích thước khác nhau, trong đó mỗi loại là một phần (lát cắt) của tổng thể. (*hierarchical data*) trong các hình chữ nhật lồng nhau và loại trừ lẫn nhau với các kích thước khác nhau.
- **Kiểu dữ liệu phổ biến:** dữ liệu có cấu trúc phân cấp và các danh mục có thể chia thành các danh mục con.

- **Điễn giải:** mỗi hình chữ nhật đại diện cho một phạm trù trong tổng thể, và hình chữ nhật càng lớn thì giá trị (tỷ lệ) của nó càng lớn. Màu sắc có thể được sử dụng để biểu thị các giá trị hoặc biến khác.
- **Ví dụ về các ứng dụng:** để trực quan hóa tỷ lệ các biến phân cấp, chẳng hạn như các châu lục, quốc gia và dân số; tháng, ngày, lượng mưa, v.v.

```
[34]: # Treemap for the Gapminder dataset having the countries as the
      # rectangles,
      # their GDP in the slice sizes, and the continents in the colors

import pandas as pd
import squarify
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

# Load the Gapminder dataset
dgapminder = pd.read_csv('gapminder_data_graphs.csv')

# Define a color palette for each continent
color_palette = {'Asia': '#Ffff00',
                 'Europe': '#ff7f0e',
                 'Africa': '#2ca02c',
                 'North America': '#d62728',
                 'South America': '#1f77b4',
                 'Oceania': '#9467bd' }

# Calculate total GDP per continent
gdp_per_continent = dgapminder.groupby('continent')['gdp'].sum()

# Calculate the percentage of GDP by country in each continent
gdp_perc_by_country = dgapminder.
    ↪groupby(['continent', 'country'])['gdp'].sum() /
    ↪gdp_per_continent

# Assign colors to each rectangle based on the corresponding
# continent
colors = [color_palette[c] for c in gdp_perc_by_country.index.
    ↪get_level_values(0) ]

# Create custom legend
```

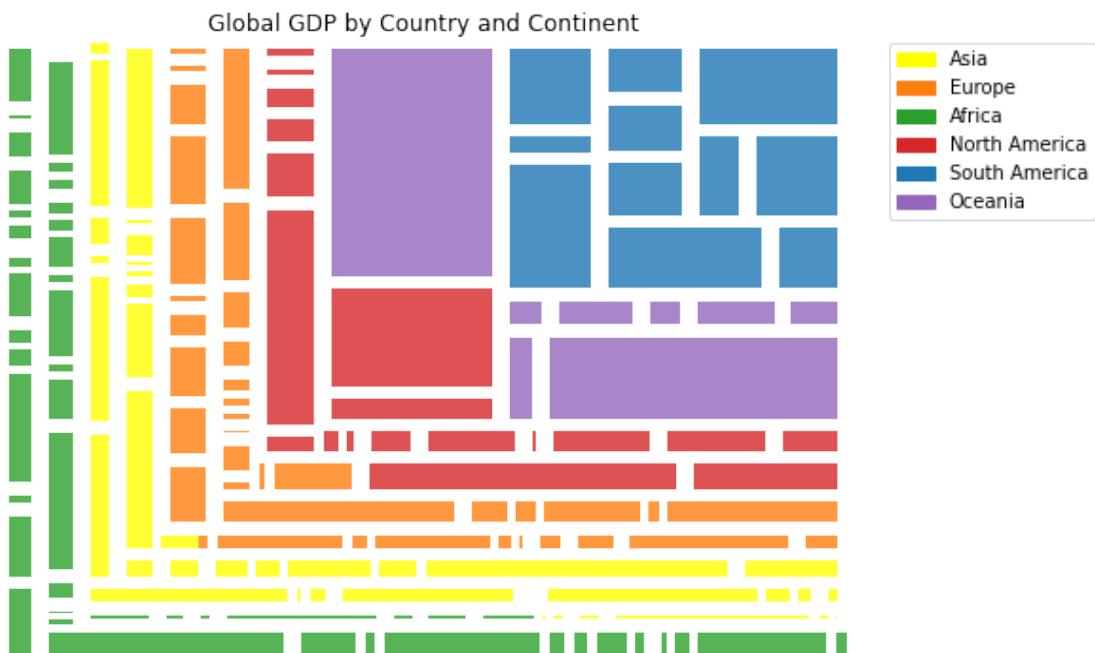
```

handles = [mpatches.Patch(color=color_palette[continent],
                           label=continent) for continent in color_palette.keys()]

# Plot the tree map
plt.figure(figsize=(8, 6))
# squarify.plot(sizes=gdp_perc_by_country,
label=gdp_perc_by_country.index.get_level_values(1),
#           alpha=.8, pad=True, color=colors)
squarify.plot(sizes=gdp_perc_by_country, alpha=.8, pad=True,
              color=colors)
plt.axis('off')
plt.legend(handles=handles, bbox_to_anchor=(1.05, 1), loc=2,
           borderaxespad=0.)
plt.title('Global GDP by Country and Continent')
plt.show()

# Print the tree structure
print(gdp_perc_by_country)

```



Hình 4.10: Treemap.

4.5 Biểu đồ đường và dòng

Có một lượng lớn dữ liệu cho thấy **sự thay đổi của các biến theo thời gian** (*evolution*). Ví dụ, dữ liệu về số lượng người trong một thành phố, thu nhập trung bình, tuổi tác, sản xuất công nghiệp, hoặc giá cổ phiếu đều thay đổi theo thời gian. Dữ liệu được thu thập theo thời gian là một trong những loại phổ biến nhất và đóng vai trò quan trọng trong việc tìm hiểu cách một biến nhất định phát triển.

Khác biệt với trường hợp tiến hóa, nơi các biến thay đổi theo thời gian, trong trường hợp **lưu lượng** (*flow*), ý tưởng là điều tra **luồng dữ liệu** qua các giai đoạn khác nhau của một quá trình chuyển đổi. Ví dụ về quá trình chuyển đổi bao gồm từ cấp trung học lên đại học, rồi thạc sĩ và tiến sĩ, hoặc từ các vị trí vận hành lên quản lý.

Mục này xem xét **biểu đồ đường** là kỹ thuật chính để nghiên cứu sự thay đổi của dữ liệu theo thời gian (tiến hóa), và **biểu đồ Sankey** cùng **biểu đồ Gantt** là các công cụ để trực quan hóa luồng dữ liệu.

4.5.1 Line Chart (Biểu đồ đường)

- Mục đích:** Để hiển thị các mẫu, xu hướng, thay đổi và bất thường trong dữ liệu trên thời gian hoặc một biến số tăng dần (số lượng).
- Kiểu dữ liệu phổ biến:** chuỗi một chiều liên tục.
- Điều giải:** xác định xu hướng, những thay đổi về hướng hoặc cường độ của xu hướng, tìm kiếm các điểm tăng đột biến hoặc giảm đột ngột, xác định các mô hình hoặc chu kỳ lặp lại, v.v.
- Ví dụ về ứng dụng:** để trực quan hóa giá cổ phiếu, dữ liệu thời tiết, xu hướng dân số, sự lây lan của dịch bệnh, v.v.

Biểu đồ đường hữu ích trong việc xác định các mô hình và xu hướng trong một chuỗi dữ liệu đơn biến một chiều, tức là dữ liệu liên tục theo thời gian với một giá trị duy nhất cho mỗi mục dữ liệu. Chúng ánh xạ dữ liệu chuỗi (ví dụ: thời gian) vào một chiều, thường là trục x, và giá trị dữ liệu vào một chiều khác, thường là trục y, tạo thành một đường thẳng; hoặc vào màu của một dấu hiệu, hoặc vùng dọc theo trục không gian, tạo thành một thanh. Dữ liệu được điều chỉnh kích thước để nằm trong giới hạn của thuộc tính hiển thị.

```
[11]: # Line charts for Life Expectancy, CO2 Consumption, and HDI
      # vs the year for the Gapminder dataset

import pandas as pd
import matplotlib.pyplot as plt

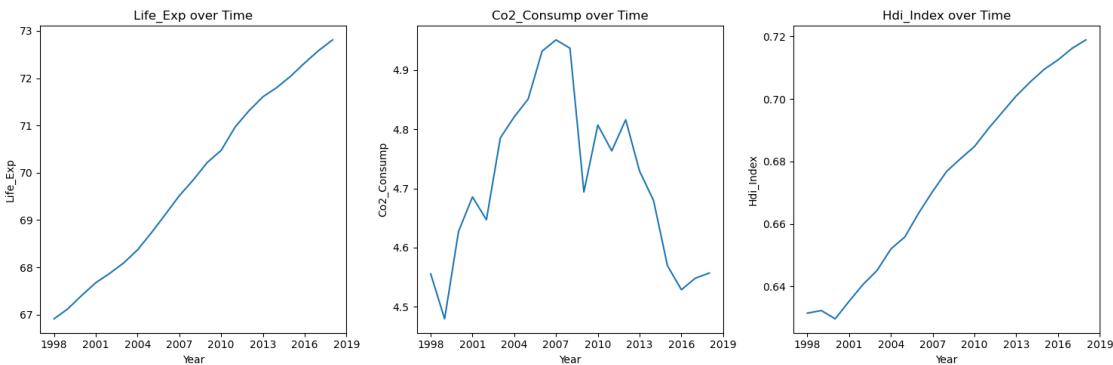
# Load the dataset, select the target variables and group by
# year
df = pd.read_csv('gapminder_data_graphs.csv')
dgapminder = df[['life_exp', 'co2_consump', 'hdi_index',
                  'year']]
dby = dgapminder.groupby('year').mean() # dby: data by year

# Loop through a list of tuples and plot each variable in a
# separate subplot
fig, axes = plt.subplots(ncols=3, figsize=(15,5))
vars_and_indices = [('life_exp', 0), ('co2_consump', 1),
                    ('hdi_index', 2)]
for var, i in vars_and_indices:
    axes[i].plot(dby.index, dby[var])
    #axes[i].bar(dby.index, dby[var])
    axes[i].set_title(var.replace('-', ' ').title() + ' over
                      Time')
    axes[i].set_xlabel('Year')
    axes[i].set_ylabel(var.replace('-', ' ').title())
    axes[i].xaxis.set_major_locator(plt.
        MaxNLocator(integer=True))

# Adjust the spacing between the subplots and show the plots
plt.tight_layout()
plt.show()
```

4.5.2 Sankey Chart (Biểu đồ Sankey)

- **Mục đích:** Để so sánh và trực quan hóa luồng dữ liệu qua các giai đoạn hoặc quá trình chuyển đổi khác nhau của một quy trình.
- **Các loại dữ liệu phổ biến:** dữ liệu phân loại và dữ liệu số (thứ tự và liên tục).
- **Điển giải:** Biểu đồ Sankey bao gồm các nút (thường được biểu thị bằng hình chữ



Hình 4.11: Biểu đồ đường.

nhật hoặc văn bản) và các luồng (thường được biểu thị bằng mũi tên hoặc cung tròn) tương ứng với dữ liệu và lượng dữ liệu được di chuyển qua luồng. Luồng càng lớn (rộng) thì lượng dữ liệu được di chuyển từ nút này sang nút khác càng lớn, hoặc mức độ quan trọng của dữ liệu càng cao. Do đó, độ rộng của luồng cho phép chúng ta quan sát các điểm nghẽn và đường dẫn của các luồng cao hơn.

- **Ví dụ về ứng dụng:** dòng năng lượng trong lưới điện, dòng người qua các sân bay hoặc quốc gia, dòng hàng hóa và dịch vụ trong chuỗi cung ứng, dòng xe trên đường cao tốc, dòng sinh viên qua các năm, v.v.

Biểu đồ Sankey là một loại đồ thị cho phép trực quan hóa luồng dữ liệu, trong đó các mục được biểu diễn dưới dạng **các nút** (*nodes*) và luồng được biểu diễn bằng **các cung** (*arcs*) có độ rộng khác nhau.

```
[2]: # Sankey diagram with the flow of an academic career progression

import plotly.graph_objects as go

# Define the nodes of the Sankey diagram
nodes = dict(
    type='sankey',
    node=dict(
        pad=15,
        thickness=20,
        line=dict(color='black', width=0.5),
        label=["Undergraduate", "M.Sc.", "Ph.D.", "Postdoc",
              "Faculty", "Industry"],
        color=[ "#3D9970", "#FF851B", "#FFDC00", "#7FDBFF",
              "#FF7F0E", "#D62728"]
```

```

),
# Define the links between the nodes
link=dict(source=[0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4,
-5, 5],
          target=[1, 2, 4, 5, 2, 4, 5, 3, 4, 5, 4, 5, 4, 5, 5, 4,
-value=[20,10,10,60,40,40,20,40,40,20,70,30,80,20,80,20])))

# Define the layout of the Sankey diagram
layout = dict(title="Flow of Academic Career
Progression", font=dict(size=10))

# Create the figure
fig = go.Figure(data=[nodes], layout=layout)

# Show the figure
fig.show()

```

4.5.3 Gantt Chart (Biểu đồ Gantt)

- **Mục đích:** Để hình dung và khám phá dòng thời gian của các nhiệm vụ, thời lượng của chúng, và sự phụ thuộc theo thời gian.
- **Loại dữ liệu phổ biến:**
 - a. Các nhiệm vụ thường được mô tả bằng **biến danh nghĩa**.
 - b. Ngày bắt đầu, ngày kết thúc và thời lượng được mô tả bằng **biến số liên quan đến thời gian** (giờ, ngày, tháng, năm, v.v.).
 - c. Các biến định tính khác như sự phụ thuộc, các cột mốc, và tài nguyên có thể được thêm vào.
- **Điễn giải:** các thanh ngang biểu diễn các nhiệm vụ, độ dài của chúng tương ứng với thời lượng của nhiệm vụ, vị trí ban đầu của chúng là thời điểm bắt đầu của nhiệm vụ và vị trí kết thúc của chúng là thời điểm kết thúc của nhiệm vụ. Các đường nối hoặc mũi tên có thể được sử dụng để thể hiện sự phụ thuộc giữa các nhiệm vụ, và các thanh tổng hợp biểu diễn các nhiệm vụ con.
- **Ví dụ về các ứng dụng:** chủ yếu được sử dụng để trực quan hóa lịch trình dự án cho người quản lý dự án Mục đích quản lý. Nó cho phép theo dõi tiến độ, xác định các điểm nghẽn và sự chậm trễ, phân bổ nguồn lực và ra quyết định.

```
[3]: # Gantt Chart to show the career evolution for the stages in the
      # previous Sankey Chart

import plotly.express as px
import pandas as pd

# Define the tasks and their start/end dates
tasks = ["Undergraduate", "M.Sc.", "Ph.D.", "Postdoc",
         "Faculty", "Industry"]
start_dates =
    ['2022-08', '2026-08', '2027-08', '2030-08', '2032-08', '2042-08']
end_dates =
    ['2026-05', '2027-05', '2030-05', '2032-05', '2042-05', '2057-05']

# Create a Dataframe with the task data
data = {'Career Stage': tasks, 'Start': start_dates, 'Finish':
        end_dates}
df = pd.DataFrame(data)

# Create the Gantt chart
fig = px.timeline(df, x_start='Start', x_end='Finish', y='Career
      # Stage')

# Customize the Gantt chart appearance
fig.update_layout(title='Gantt Chart of Career Evolution',
                  xaxis_title='Year',
                  yaxis_title='Career Stage')

# Show the Gantt chart
fig.show()
```

4.6 Dữ liệu Địa lý Không gian (Geospatial)

Các phương pháp trực quan hóa dữ liệu địa lý không gian phù hợp để vẽ đồ thị các biến liên quan đến **các vị trí trong thế giới vật lý**.

Mục này xem xét **Bản đồ phân vùng theo màu** (*Choropleth*) và **Bản đồ bong bóng** (*Bubble maps*) là những công cụ quan trọng để trực quan hóa dữ liệu địa lý không gian.

4.6.1 Choropleth Map (Bản đồ Phân vùng theo màu)

- **Mục đích:** Khám phá và trực quan hóa **dữ liệu không gian** (*spatial data*) trong bản đồ khu vực.
- **Loại dữ liệu phổ biến:** Dữ liệu không gian, tức là dữ liệu chứa các giá trị hoặc danh mục khác nhau được liên kết với các khu vực (địa lý) cụ thể.
- **Giải thích:** Bản đồ này bao gồm việc **tô màu hoặc tô bóng các khu vực** sao cho màu tối hơn thường biểu thị các giá trị cao hơn, trong khi màu sáng hơn thường biểu thị các giá trị nhỏ hơn.
- **Các ví dụ ứng dụng:** Trực quan hóa dữ liệu không gian như mật độ dân số, sự lây lan dịch bệnh, tiêu thụ điện, dữ liệu thời tiết, sở thích chính trị, tỷ lệ việc làm, các chỉ số kinh tế (ví dụ: GDP và HDI), v.v..
- **Phân loại:** Bản đồ phân vùng theo màu được phân loại thành:
 - a. **Bản đồ Choropleth phân lớp** (*Classed choropleth maps*): Biến số được **rời rạc hóa** (*discretized*), và bản đồ sử dụng một tập hợp màu sắc hữu hạn để đại diện cho các khoảng lớp của biến. Ví dụ, việc phân loại các quốc gia theo kích thước dân số thành các khoảng xác định giúp dễ dàng quan sát sự khác biệt hơn.
 - b. **Bản đồ Choropleth không phân lớp** (*Unclassed choropleth maps*): Sử dụng **thang đo màu liên tục** để đại diện cho các giá trị của biến. Trong trường hợp này, việc phân biệt (ước tính) kích thước dân số của hầu hết các quốc gia có thể khó khăn do dân số ở châu Á (Ấn Độ và Trung Quốc) lớn hơn nhiều so với các quốc gia khác.
- **Tính tương tác:** Phương pháp `choropleth()` trong Plotly cho phép vẽ một bản đồ tương tác với một biến động (dynamic variable), ví dụ như hiển thị sự tiêu thụ CO_2 theo thời gian. Bằng cách di chuyển con trỏ qua bản đồ, người dùng có thể thấy thông tin về năm, quốc gia và mức tiêu thụ CO_2 .

```
[28]: # Choropleth map and its variations for the Geopandas
# naturalearth_lowres dataset

import geopandas as gpd
import matplotlib.pyplot as plt

# Load the dataset
```

```

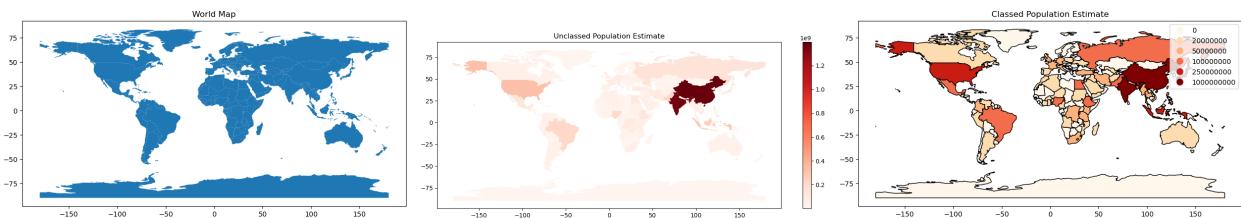
world = gpd.read_file(gpd.datasets.
    ↪get_path('naturalearth_lowres'))

# Plot a basic map
world.plot(figsize=(10, 6))
plt.title("World Map")
plt.show()

# Plot a choropleth map
world.plot(column='pop_est', cmap='Reds', legend=True,
    ↪figsize=(13, 5))
plt.title("Unclassed Population Estimate")
plt.show()

# Choropleth map with the classed data
# Define breaks for the population estimate data
breaks = [0, 20000000, 50000000, 100000000, 250000000,
    ↪1000000000, np.inf]
# Assign each country to a class based on its population
estimate
world['pop_class'] = pd.cut(world['pop_est'], bins=breaks,
    ↪labels=breaks[:-1],
                           include_lowest=True, right=False)
# Plot a choropleth map with the classed data
world.plot(column='pop_class', cmap='OrRd', legend=True,
    ↪figsize=(10, 6),
    edgecolor='black')
plt.title("Classed Population Estimate")
plt.show()

```



Hình 4.12: Bản đồ.

4.6.2 Bubble Map (Bản đồ Bong bóng)

- **Mục đích:** Để khám phá và trực quan hóa dữ liệu không gian được biểu thị bằng các bong bóng hoặc hình tròn có kích thước khác nhau trên bản đồ khu vực.
- **Kiểu dữ liệu phổ biến:** dữ liệu không gian, tức là dữ liệu chứa các giá trị hoặc danh mục khác nhau các khu vực liên quan đến các vùng (địa lý) cụ thể.
- **Điễn giải:** bản đồ bong bóng sử dụng các chấm (bong bóng) có kích thước khác nhau trên bản đồ để biểu thị giá trị của biến.
- **Ví dụ về ứng dụng:** để trực quan hóa dữ liệu không gian, chẳng hạn như mật độ dân số, sự lây lan của dịch bệnh, mức tiêu thụ điện, dữ liệu thời tiết, tỷ lệ việc làm, chỉ số kinh tế (ví dụ: GDP và HDI), v.v.

Tính tương tác: Tương tự như phương pháp `choropleth()` trong Plotly, phương pháp `scatter_geo()` được sử dụng để tạo bản đồ bong bóng cũng mang tính **tương tác**. Người dùng có thể di chuyển chuột qua bản đồ để kiểm tra thông tin liên quan đến từng bong bóng, và có thể cuộn để phóng to hoặc thu nhỏ bản đồ.

```
[4]: # Animated Choropleth map for a merged dataset
# of Gapminder and naturalearth_lowres

import pandas as pd
import geopandas as gpd
import plotly.express as px

# Load the Gapminder dataset
dgapminder = pd.read_csv('gapminder_data_graphs.csv')
dgapminder['gdp'] = dgapminder['gdp'].fillna(0)

# Load a world shapefile
dworld = gpd.read_file(gpd.datasets.
    get_path('naturalearth_lowres'))

# Merge the Gapminder dataset with the world shapefile
dmerged = dworld.merge(dgapminder, left_on='name',
    right_on='country', how='inner')
fig = px.choropleth(dgapminder, locations='country',
    locationmode='country names',
    color='co2_consump', animation_frame='year',
```

```

color_continuous_scale=px.colors.sequential.
    -Plasma_r)
fig.update_layout(title={
    'text': 'Choropleth Map of CO2 Consumption Animated by
    -Year',
    'font': {'size': 16}})
fig.show()

```

4.7 Bài tập

Chủ đề và câu hỏi nghiên cứu

Bài 4.1. Khám phá các khía cạnh của việc trực quan hóa mối liên hệ bằng biểu đồ phân tán, biểu đồ bong bóng, ma trận phân tán và bản đồ nhiệt. Những hình ảnh trực quan này giúp chúng ta hiểu mối quan hệ giữa các biến như thế nào?

Bài 4.2. Thảo luận về ý nghĩa của việc trực quan hóa số lượng bằng biểu đồ thanh và biểu đồ radar. Những hình ảnh trực quan này giúp chúng ta hiểu được quy mô của các loại hoặc biến khác nhau như thế nào?

Bài 4.3. Khám phá tính tỷ lệ trong biểu đồ hình tròn. Thảo luận về cách bản chất hình tròn của biểu đồ hình tròn có thể ảnh hưởng đến nhận thức về tỷ lệ và xem xét các câu hỏi về hiệu quả và độ chính xác của việc biểu diễn tỷ lệ trong định dạng trực quan này.

Bài 4.4. Khám phá ý nghĩa của việc biểu diễn dữ liệu bằng biểu đồ histogram. Thảo luận về cách lựa chọn kích thước và khoảng cách bin có thể ảnh hưởng đến nhận thức về các mẫu, đặt ra câu hỏi về tính khách quan và chủ quan của việc biểu diễn dữ liệu.

Bài tập tính toán

Bài 4.5. Đối với tập dữ liệu Cháy rừng, hãy vẽ biểu đồ phân phối tần suất bằng thang logarit. Điều gì xảy ra với hình dạng của phân phối?

Bài 4.6. Đối với tập dữ liệu Auto MPG, hãy tạo biểu đồ hộp để so sánh phân phối (trung tâm, độ lan tỏa, giá trị ngoại lai) của từng biến liên tục, khám phá việc sử dụng biểu đồ hộp có khía để so sánh so sánh các trung vị hiệu quả hơn và điều tra sự hiện diện của các giá trị ngoại lai và tác động tiềm ẩn của chúng.

Bài 4.7. Đối với tập dữ liệu Gapminder, hãy tạo biểu đồ violin để trực quan hóa phân phối và mật độ tuổi thọ, chỉ số HDI và mức tiêu thụ CO2 trên toàn cầu các sắc thái. So sánh biểu đồ violin với biểu đồ hộp và thảo luận về những hiểu biết bổ sung cung cấp.

Bài 4.8. Đối với tập dữ liệu Daily Delhi Climate Train, hãy thực hiện:

- Hình dung xu hướng của nhiều thông số thời tiết khác nhau (ví dụ: nhiệt độ, độ ẩm, áp suất) theo thời gian bằng biểu đồ đường.
- Phân tích các mô hình hàng ngày, theo mùa hoặc hàng năm.
- Khám phá sự phân bố nhiệt độ, độ ẩm và các biến số khác bằng cách sử dụng histogram hoặc biểu đồ hộp, và xác định các giá trị ngoại lai tiềm ẩn hoặc các hiện tượng thời tiết cực đoan.
- Tạo biểu đồ phân tán để nghiên cứu mối quan hệ giữa các thông số thời tiết khác nhau và xem liệu có mối tương quan giữa nhiệt độ và độ ẩm hoặc tốc độ gió hay không.

Bài 4.9. Đối với tập dữ liệu Naturalearth_lowres, hãy thực hiện:

- Tạo bản đồ choropleth để trực quan hóa nhiều số liệu thống kê khác nhau trên toàn thế giới, chẳng hạn như mật độ dân số, diện tích đất hoặc các chỉ số kinh tế của các quốc gia khác nhau.
- Sử dụng bản đồ bong bóng để biểu diễn dân số của các thành phố lớn hoặc GDP của các thành phố khác nhau. các quốc gia cụ thể khi đặt chúng trên bản đồ thế giới.
- Khám phá việc kết hợp bản đồ choropleth với các hình ảnh trực quan khác (ví dụ: biểu đồ phân tán) để hiển thị các điểm dữ liệu bổ sung trên bản đồ địa lý.

Nghiên cứu tình huống Chương 3 trình bày ba nghiên cứu điển hình tập trung vào phân tích mô tả. Đối với các nghiên cứu điển hình tương tự, hãy thực hiện:

Bài 4.10. Kiểm tra các giá trị bị thiếu và quyết định chiến lược xử lý phù hợp (ví dụ: imput-sự di dời, sự loại bỏ).

Bài 4.11. Phân tích các kiểu dữ liệu của từng biến và trực quan hóa phân phối của chúng bằng cách sử dụng histogram, biểu đồ hộp hoặc các kỹ thuật khác.

Bài 4.12. Kiểm tra sự hiện diện của các giá trị ngoại lai và quyết định xem có nên loại bỏ, chuyển đổi hay giữ lại hay không. dựa trên tác động của chúng đối với quá trình phân tích.

Bài 4.13. Sử dụng biểu đồ phân tán, ma trận tương quan hoặc bản đồ nhiệt để khám phá các mối quan hệ tiềm năng giữa các tính năng khác nhau trong tập dữ liệu.

Bài 4.14. So sánh các biến số giữa các nhóm khác nhau (ví dụ: sự hiện diện của bệnh so với sự vắng mặt của bệnh trong ô tô) dữ liệu mạch máu, danh mục sản phẩm trong dữ liệu siêu thị, loại đăng ký trong dữ liệu Netflix) bằng biểu đồ thanh, biểu đồ hộp hoặc biểu đồ violin.

Bài 4.15. Tạo hình ảnh trực quan giàu thông tin phù hợp với loại dữ liệu và mục tiêu phân tích. Sử dụng các thư viện như Matplotlib, Seaborn hoặc Plotly để tạo hình ảnh trực quan tương tác.

Chương 5

MỘT SỐ LOẠI DỮ LIỆU ĐẶC BIỆT

Trong khoa học dữ liệu, việc khám phá và phân tích các bộ dữ liệu đa dạng thúc đẩy sự hiểu biết sâu sắc và việc ra quyết định. Chương này tập trung vào ba loại dữ liệu đặc biệt: **chuỗi thời gian** (có yếu tố thời gian), **văn bản/tài liệu** (bản cấu trúc hoặc phi cấu trúc), **cây/mạng lưới** (quan hệ được xác định trước giữa các đối tượng), và **dữ liệu nhiều chiều**.

5.1 Chuỗi Thời Gian (Time Series)

Một chuỗi các điểm dữ liệu hoặc quan sát được đo lường qua thời gian được gọi là **chuỗi thời gian (TS)**. Thang thời gian có thể là bất kỳ bậc độ lớn nào, như mili giây, giây, giờ, ngày, tuần, tháng, năm, hoặc thậm chí các khoảng thời gian dài hơn. Đặc điểm phân biệt của chuỗi thời gian là chúng **được lập chỉ mục theo thời gian**. Đặc điểm này khiến chúng trở thành một loại dữ liệu đặc biệt, vì nó cho phép phân tích quá khứ và dự đoán tương lai của một biến nhất định. Các khái niệm được khám phá trong mục này sẽ sử dụng Bộ dữ liệu Khí hậu hàng ngày Delhi (*Daily Delhi Climate Data*).

5.1.1 Các Loại và Đặc Điểm của Chuỗi Thời Gian

Chuỗi thời gian là một trong những loại dữ liệu phổ biến nhất hiện có và có một số danh mục đại diện:

- **Kinh tế (Economy):** Dữ liệu liên quan đến tỷ lệ lạm phát, tỷ lệ việc làm và thất nghiệp, giá cổ phiếu, chỉ số thị trường chứng khoán, GDP, lãi suất, cán cân thương mại, v.v.

- **Vật lý (Physical):** Dữ liệu liên quan đến các khoa học vật lý (ví dụ: thiên văn học, khí tượng, khí hậu học, v.v.).
- **Tiếp thị (Marketing):** Hầu hết dữ liệu tiếp thị liên quan đến các chỉ số hiệu suất cụ thể như tỷ lệ chuyển đổi, doanh thu bán hàng, tỷ lệ nhấp chuột, v.v.
- **Nhân khẩu học (Demography):** Dữ liệu liên quan đến đặc điểm dân số như tuổi, giới tính, thu nhập, trình độ học vấn, **tuổi thọ**, Chỉ số Phát triển Con người (HDI), GDP, v.v.
- **Kiểm soát Quy trình (Process Control):** Dữ liệu liên quan đến các phép đo của một hệ thống hoặc quy trình theo thời gian, như số lượng sản phẩm được sản xuất.

5.1.2 Mục tiêu Phân tích Khám phá Chuỗi Thời Gian (EDA)

Mặc dù có rất nhiều phân tích có thể được thực hiện trên dữ liệu TS, nhưng hầu hết nghiên cứu trong lĩnh vực này là **dự đoán** (*predictive*); tức là, mục tiêu là sử dụng dữ liệu quá khứ, và đôi khi là thông tin khác, để dự đoán giá trị tương lai của chuỗi.

- **Mô tả:** Ngay cả các phân tích mô tả đơn giản được trình bày ở đây cũng cho phép nhà phân tích có được những hiểu biết quan trọng về chuỗi dữ liệu, chẳng hạn như quan sát xu hướng, chu kỳ, tính theo mùa và các mô hình hoặc đặc điểm khác trong dữ liệu.
- **Giải thích:** nhằm mục đích giải thích nguyên nhân hoặc các yếu tố dẫn đến hiện tượng quan sát được. Các thuật ngữ và mối quan hệ giữa các biến thể của một biến và các biến khác.
- **Dự đoán:** liên quan đến việc ước tính hoặc dự báo các giá trị hoặc mô hình tương lai trong chuỗi thời gian. Nó có giá trị thực tiễn đáng kể vì nó cho phép chúng ta tránh tổn thất, tăng lợi nhuận và lập kế hoạch cũng như đưa ra quyết định nói chung.
- **Kiểm soát:** bao gồm việc thực hiện một hành động cụ thể trong chuỗi thời gian, chẳng hạn như khi kết quả mong muốn xảy ra. Trong hầu hết các trường hợp, việc thiết kế các chiến lược kiểm soát chuỗi thời gian sẽ yêu cầu áp dụng trước các phương pháp mô tả, giải thích và dự đoán.

Ở đây chúng ta sẽ tập trung vào phân tích mô tả dữ liệu chuỗi thời gian bằng các phương pháp được đề cập trước. đã gửi trong các chương trước và sẽ bổ sung thêm hai kỹ thuật phân tích cụ thể: trung bình di chuyển độ tuổi và phân tích chuỗi. Cả hai đều có mục đích cung cấp cái nhìn tổng quan về xu hướng của chuỗi.

5.1.3 Trực quan hóa Dữ liệu Chuỗi Thời Gian

Trong phần này, trọng tâm là phân tích mô tả dữ liệu chuỗi thời gian bằng các phương pháp đã được trình bày trong các chương trước, và bổ sung hai kỹ thuật phân tích cụ thể: **trung bình trượt** (*moving averages*) và **phân rã chuỗi** (*series decomposition*). Cả hai đều phục vụ mục đích có được cái nhìn tổng quan về **xu hướng** của chuỗi.

Các loại biểu đồ sau được sử dụng cho phân tích chuỗi thời gian:

- **Biểu đồ Đường (Line Charts):** Được sử dụng để trực quan hóa chung về **hình dạng, xu hướng, tính mùa vụ** (*seasonality*) và **tính chu kỳ** (*cyclicity*) của chuỗi thời gian.
- **Boxplot (Biểu đồ Hộp):** Được sử dụng để hiển thị **phân phối** các giá trị qua các khoảng thời gian.
- **Ma trận Biểu đồ Phân tán (Scatterplot Matrices):** Được sử dụng để điều tra phân phối tần suất và **mối liên hệ** giữa các giá trị chuỗi.
- **Heatmaps (Bản đồ Nhiệt):** Được sử dụng để tính toán và trực quan hóa **tương quan** trong dữ liệu chuỗi thời gian.

```
[8]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Line charts for the Daily Delhi Climate Train Data
df = pd.read_csv('DailyDelhiClimateTrain.csv')
dDelhi = df[['meantemp', 'humidity', 'wind_speed',
             'meanpressure', 'date']]
fig, axes = plt.subplots(nrows=4, figsize=(10, 15))
vars_and_indices = [('meantemp', 0), ('humidity', 1),
                     ('wind_speed', 2), ('meanpressure', 3)]
for var, i in vars_and_indices:
    axes[i].plot(dDelhi['date'], dDelhi[var], linestyle='--')
    axes[i].set_title(var.replace('-', ' ').title())
```

```

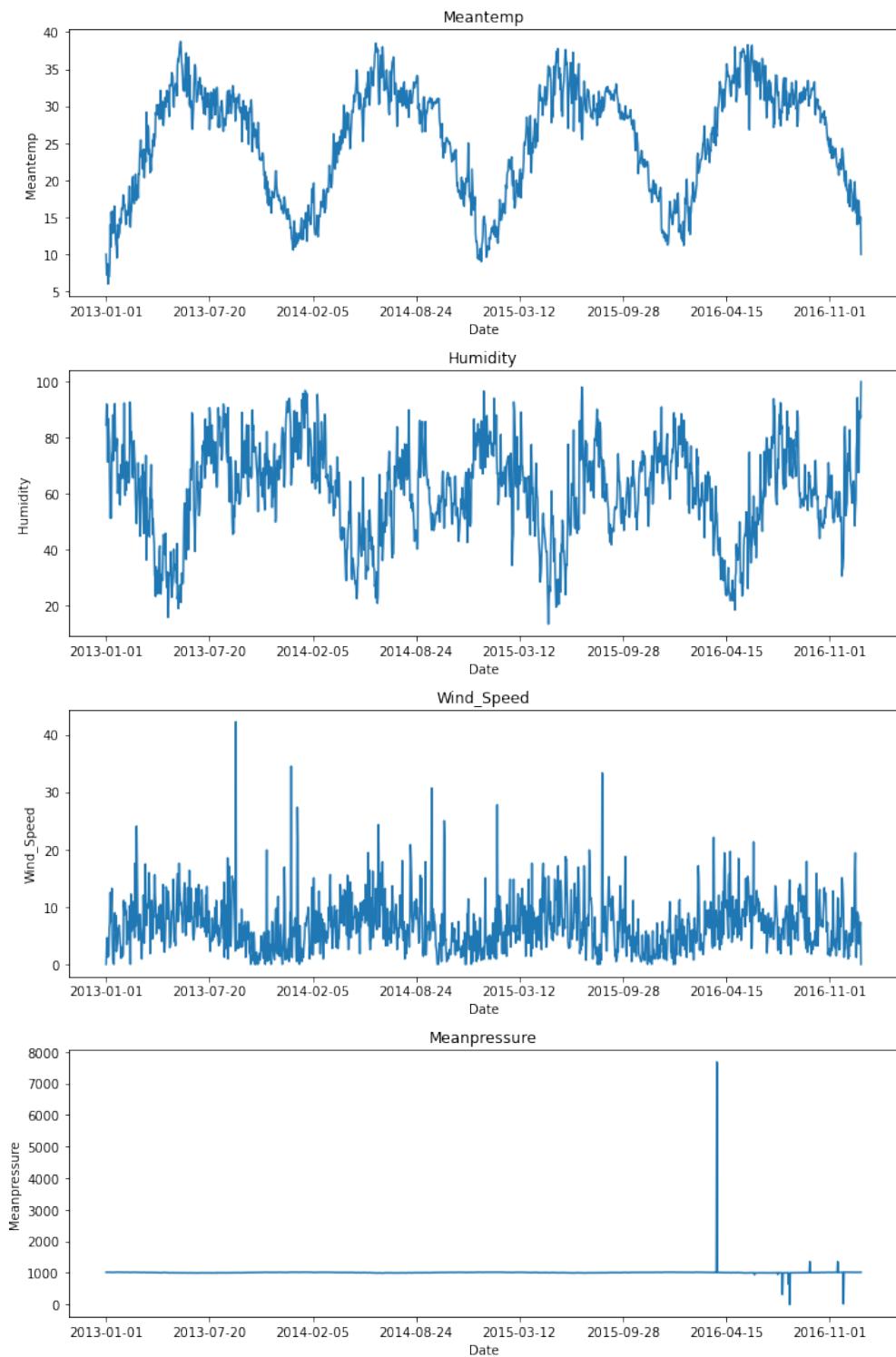
axes[i].set_xlabel('Date')
axes[i].set_ylabel(var.replace('-', ' ').title())
axes[i].xaxis.set_major_locator(plt.
    ~MaxNLocator(integer=True))
plt.tight_layout()
plt.show()

# Box plots for the Daily Delhi Climate Train Data
dDelhi = pd.read_csv('DailyDelhiClimateTrain.csv')
variables = ['meantemp', 'humidity', 'wind_speed',
    ~'meanpressure']
dDelhi_normalized = (dDelhi[variables] - dDelhi[variables].
    ~min()) / (dDelhi[variables].max() - dDelhi[variables].min())
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
axes[0].boxplot(dDelhi[variables].values)
axes[0].set_title('Boxplot - Original Data')
axes[0].set_ylabel('Value')
axes[0].set_xticklabels(variables, rotation=45)
axes[1].boxplot(dDelhi_normalized.values)
axes[1].set_title('Boxplot - Normalized Data')
axes[1].set_ylabel('Value')
axes[1].set_xticklabels(variables, rotation=45)
plt.subplots_adjust(wspace=0.4)
plt.show()

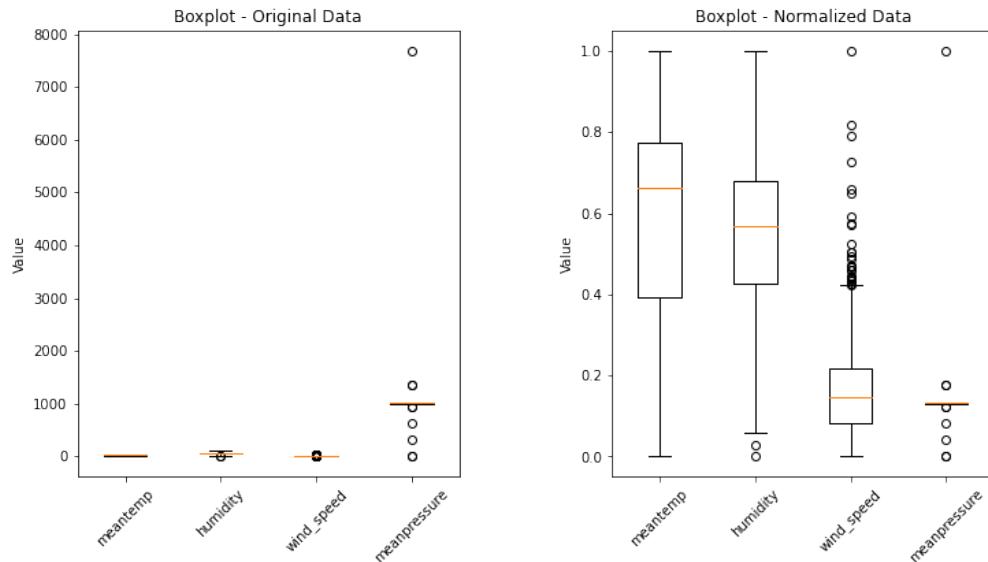
# Scatterplot matrix and Heatmap for the Daily Delhi Climate
~Train Data
dDelhi = pd.read_csv('DailyDelhiClimateTrain.csv')
variables = ['meantemp', 'humidity', 'wind_speed',
    ~'meanpressure']
sns.set(style='ticks')
sns.pairplot(dDelhi[variables])
plt.suptitle('Scatterplot Matrix', y=1.02)
plt.show()
corr_matrix = dDelhi[variables].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Heatmap - Correlation')
plt.show()

```

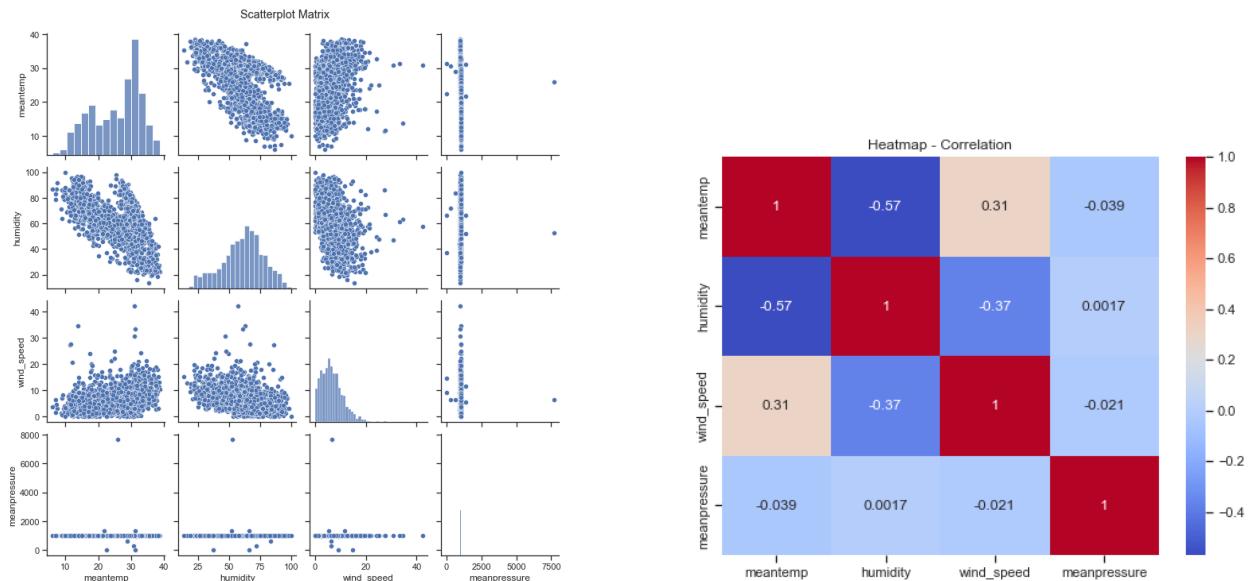
Ví dụ 5.1 (Ví dụ về Biểu đồ Đường (Dữ liệu Khí hậu Delhi)). • **Nhiệt độ trung bình (meantemp):** Có hành vi **chu kỳ và theo mùa** theo thời gian, với giá trị tối đa vào



Hình 5.1: Trực quan hóa chuỗi thời gian.



Hình 5.2: Trực quan hoá cuối thời gian.



Hình 5.3: Trực quan hoá cuối thời gian.

khoảng tháng 6 và 7, và tối thiểu vào khoảng tháng 12 và tháng 1.

- **Độ ẩm (humidity):** Theo mô hình tương tự, nhưng **ngược pha** (*reverse phase*); tức là, khi nhiệt độ cao hơn, độ ẩm thấp hơn, và ngược lại. Tính mùa vụ của độ ẩm không rõ ràng bằng nhiệt độ.
- **Áp suất trung bình (meanpressure):** Gần như **giá trị không đổi**, với một số giá trị

ngoại lai lớn (đỉnh và đáy).

5.1.4 Trung bình trượt và phân rã theo mùa

Trung bình Trượt (*Moving Average*)

- Mục đích:** Trung bình trượt, hay trung bình động (*rolling averages*), là phương pháp thường được sử dụng để **làm phẳng** các biến động của chuỗi TS và **tiết lộ xu hướng** hoặc mô hình cơ bản trong dữ liệu.
- Cách tính:** Kỹ thuật này bao gồm việc tính trung bình một số giá trị chuỗi thời gian lân cận được chỉ định cho mỗi điểm dữ liệu.
- Hiệu quả:** Phương pháp này làm phẳng hiệu quả các biến động ngắn hạn và giúp nắm bắt xu hướng cơ bản.
- Kích thước cửa sổ (Window Size):** Chiều dài của trung bình trượt thường được chọn để giảm thiểu tác động của các hiệu ứng theo mùa. **Giá trị cửa sổ lớn hơn** sẽ tạo ra một đường cong **mượt mà hơn** và dễ dàng trực quan hóa xu hướng chuỗi hơn.

```
[51]: # Moving averages for variables 'meantemp' and 'humidity'
# of the Daily Delhi Climate Train Data

import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset and extract the variables
df = pd.read_csv('DailyDelhiClimateTrain.csv')
dDelhi = df[['meantemp', 'humidity', 'date']]

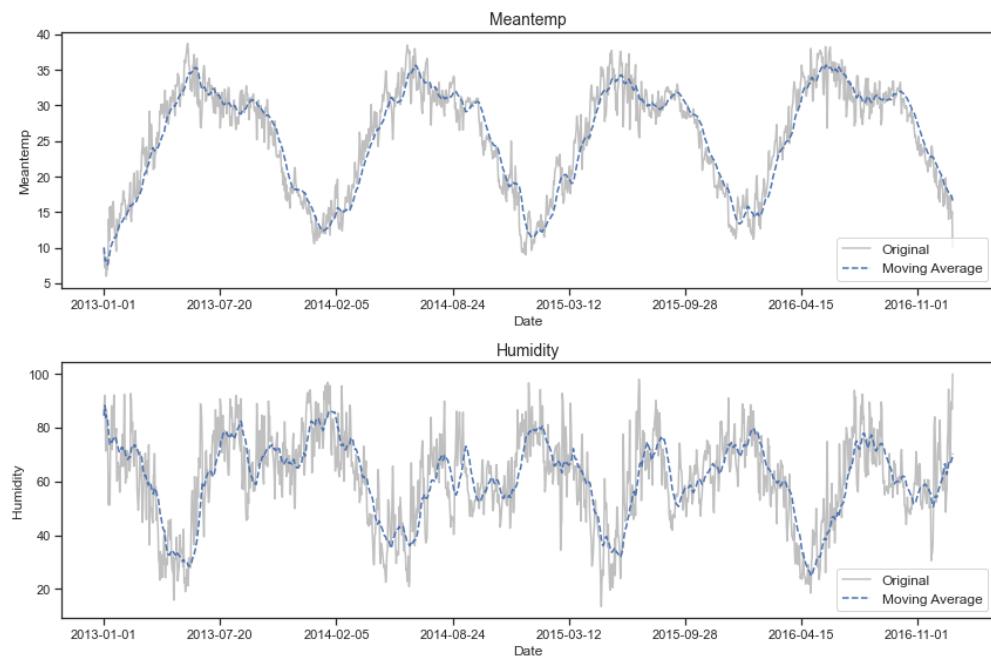
# Create a single figure with subplots stacked vertically
fig, axes = plt.subplots(nrows=2, figsize=(12, 8))
vars_and_indices = [('meantemp', 0), ('humidity', 1)]
for var, i in vars_and_indices:
    axes[i].plot(dDelhi['date'], dDelhi[var], linestyle='--',
                  color='gray',
                  alpha=.5, label='Original')
    axes[i].plot(dDelhi['date'], dDelhi[var].rolling(window=20,
min_periods=1).mean(),
                  linestyle='--', label='Moving Average')
```

```

    axes[i].set_title(var.replace('-', ' ').title(),
                       fontsize=14)
    axes[i].set_xlabel('Date', fontsize=12)
    axes[i].set_ylabel(var.replace('-', ' ').title(),
                       fontsize=12)
    axes[i].xaxis.set_major_locator(plt.
        MaxNLocator(integer=True))
    axes[i].legend(fontsize=12)

# Adjust the layout and display the plot
plt.tight_layout()
plt.show()

```



Hình 5.4: Trung bình trượt.

Phân tích(*Decomposition*)

Một chuỗi thời gian có thể được coi là bao gồm ba thành phần chính:

- **Xu hướng (*Trend*):** Tương ứng với xu hướng hoặc chuyển động **dài hạn** của chuỗi thời gian (tăng, giảm, không đổi, hoặc chu kỳ).
- **Theo mùa (*Seasonal*):** Nhằm xác định các **mẫu lặp đi lặp lại** hoặc chu kỳ trong các khoảng thời gian cụ thể (như ngày, tuần, tháng)

- **Phân dư (Residual):** Còn được gọi là nhiễu, lỗi, hoặc ngẫu nhiên, là thành phần **bất thường** còn lại sau khi các thành phần khác đã được xác định và loại bỏ.

Trung bình trượt có thể được sử dụng để xác định và loại bỏ phần xu hướng của chuỗi. Nếu sử dụng **phương pháp cộng** (*additive method*), việc trừ thành phần xu hướng khỏi chuỗi gốc sẽ còn lại các thành phần theo mùa và phân dư. Phân rã chuỗi thời gian có thể được thực hiện bằng hàm `seasonal_decompose()` của thư viện Statsmodels.

```
[9]: # Time series decomposition of the 'meantemp' variable
# of the Daily Delhi Climate Train data

import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

# Load the dataset
dDelhi = pd.read_csv('DailyDelhiClimateTrain.csv')

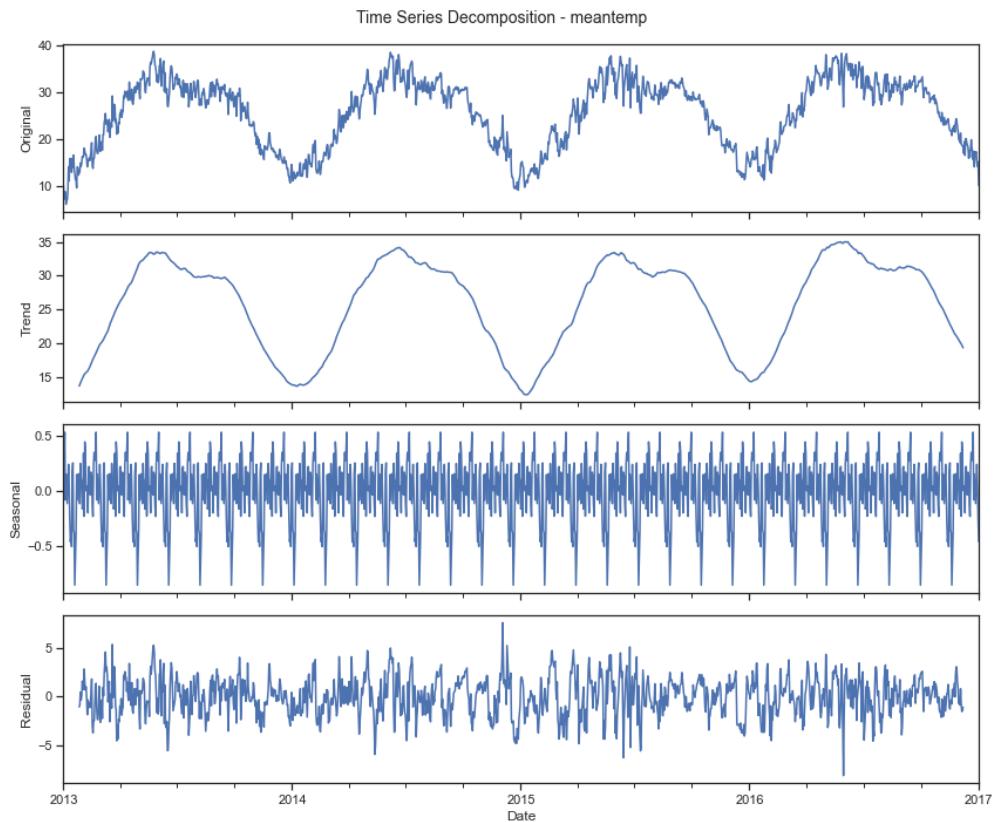
# Extract the variable and set the date column as the index
variable = 'meantemp'
dDelhi['date'] = pd.to_datetime(dDelhi['date'])
dDelhi.set_index('date', inplace=True)

# Perform seasonal decomposition
decomposition = seasonal_decompose(dDelhi[variable],
                                   model='additive', period = 50)

# Plot the original, trend, seasonal, and residual components
fig, axes = plt.subplots(4, 1, figsize=(12, 10), sharex=True)
dDelhi[variable].plot(ax=axes[0])
axes[0].set_ylabel('Original')
decomposition.trend.plot(ax=axes[1])
axes[1].set_ylabel('Trend')
decomposition.seasonal.plot(ax=axes[2])
axes[2].set_ylabel('Seasonal')
decomposition.resid.plot(ax=axes[3])
axes[3].set_ylabel('Residual')

plt.xlabel('Date')
plt.suptitle('Time Series Decomposition - {}'.format(variable),
             fontsize=14)
```

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



Hình 5.5: Phân tích chuỗi thời gian.

5.2 Dữ Liệu Văn Bản và Tài Liệu (Text and Document Data)

Số lượng dữ liệu văn bản và tài liệu được tạo ra và lưu trữ đã tăng lên đáng kể trong những năm gần đây, chủ yếu do việc sử dụng Internet, thiết bị di động, và chuyển đổi kỹ thuật số. Văn bản và tài liệu là các loại dữ liệu đặc biệt được sử dụng để truyền tải thông tin liên quan đến ngôn ngữ tự nhiên hoặc quy trình giao tiếp.

- **Dữ liệu Văn bản (Text data)** được coi là **văn bản thuần túy (plain text)**, tức là một chuỗi các ký tự hoặc từ, ví dụ như e-mail, đánh giá sản phẩm, tiểu luận, tweets, v.v..

- **Dữ liệu Tài liệu (Document data)** bổ sung các thuộc tính hoặc thông tin vào dữ liệu văn bản, chẳng hạn như **siêu dữ liệu (metadata)**, định dạng, và phương tiện truyền thông.

Dữ liệu văn bản thường không được tổ chức xung quanh một mô hình dữ liệu hoặc cấu trúc dữ liệu cụ thể, và do đó, chúng là dữ liệu **phi cấu trúc** hoặc **bán cấu trúc**, tùy thuộc vào sự tồn tại của thông tin bổ sung (ví dụ: siêu dữ liệu) để mô tả các đối tượng.

Hai lĩnh vực phân tích chính xoay quanh văn bản và tài liệu là **khai thác văn bản (text mining)** và **xử lý ngôn ngữ tự nhiên (natural language processing - NLP)**. Khai thác văn bản là một lĩnh vực điều tra đa ngành nhằm mục đích trích xuất các quy luật và mô hình từ các văn bản ngôn ngữ tự nhiên, tương tự như khai thác dữ liệu trong các tập dữ liệu có cấu trúc.

Phân tích khám phá (EDA) của văn bản đòi hỏi các phương pháp và quy trình cụ thể, vì văn bản là dữ liệu **định tính**. EDA của văn bản sử dụng các khái niệm và kỹ thuật từ khai thác văn bản và NLP, cung cấp các thước đo số học và phương pháp trực quan hóa để hiểu cấu trúc văn bản và đặc điểm của nó.

5.2.1 Mục tiêu Phân tích Khám phá Dữ liệu Văn bản và Tài liệu

Mặc dù các thước đo định lượng chung (như trung bình, mode, và phương sai) có thể được tính toán, chúng cần được đặt trong ngữ cảnh của văn bản.

Các mục tiêu của trực quan hóa văn bản bao gồm:

- Trích xuất thông tin về **mức độ liên quan hoặc tần suất** của từ, câu hoặc văn bản.
- **Phát hiện hoặc nhận dạng** các chủ đề (*topics*) và thực thể (*entities*).
- **Nhận dạng các mối quan hệ** giữa các từ, câu và văn bản.

Các thước đo định lượng hữu ích cho việc tóm tắt văn bản (*text summarization*) bao gồm: số lượng ký tự trung bình trên mỗi từ, số lượng từ (duy nhất), kích thước từ vựng, và độ dài từ trung bình.

5.2.2 Cấu trúc hóa văn bản (Text Structuring)

Tính chất bán cấu trúc hoặc phi cấu trúc của dữ liệu văn bản đòi hỏi phải có **tiền xử lý** để cấu trúc hóa dữ liệu phục vụ cho phân tích. Các hình thức cấu trúc hóa phổ biến dựa trên **biểu diễn từ vựng (lexical representations)** hoặc **biểu diễn cú pháp (syntactic representations)**.

Biểu diễn Từ vựng (Lexical Representation)

- **Khái niệm:** Phương pháp này cấu trúc dữ liệu bằng cách nhấn mạnh các từ hoặc thuật ngữ trong văn bản, biến chúng thành các **thực thể nguyên tử** gọi là **tokens**. Mỗi văn bản hoặc tài liệu là một đối tượng, và các token là các biến của đối tượng đó.
- **Mô hình Không gian Vector (Vector Space Model):** Phương pháp biểu diễn văn bản này được biết đến là mô hình không gian vector. Mỗi văn bản được đại diện dưới dạng một **vector đặc trưng**, trong đó mỗi đặc trưng (biến) tương ứng với một thuật ngữ (token) duy nhất trong tập hợp các thuật ngữ.
- **Kích thước:** Tập hợp tất cả các từ duy nhất từ tất cả các tài liệu được gọi là **corpus**. Kích thước của vector đặc trưng bằng số lượng token duy nhất, điều này có thể dẫn đến **tính chiều cao** (*large dimensionality*).
- **Trọng số:** Giá trị (*weight*) của một thuật ngữ trong vector đặc trưng tương ứng với **mức độ liên quan** (*relevance*) của nó trong tài liệu. Phương pháp này được gọi là từ vựng vì nó chia văn bản thành token mà không tính đến cú pháp hay ngữ pháp.

Các bước chính để thực hiện biểu diễn từ vựng cho tài liệu bao gồm:

- a. **Tokenization (Mã hóa):** Chuyển đổi văn bản thành các tập hợp từ (tokens) được sử dụng để xây dựng các vector đặc trưng. Việc phân tách có thể xảy ra thông qua các dấu phân cách như khoảng trắng, dấu câu (ví dụ: ";", ",", ".") và các ký tự đặc biệt.
- b. **Stop words removal (Loại bỏ từ dừng):** Từ dừng (*stopword*) là các từ được lọc bỏ vì chúng không liên quan đến mục đích phân tích và có thể được xem là nhiễu, làm khó khăn cho việc xác định tầm quan trọng của các từ trong tài liệu.
- c. **Stemming/Lemmatization:** Giảm các từ về dạng gốc của chúng. Ví dụ, thuật toán PorterStemmer là một trong những thuật toán phổ biến nhất được sử dụng trong NLP để giảm từ về gốc của chúng (*stems*).
- d. **Vector generation (Vector hóa):** Tính toán mức độ liên quan hoặc trọng số ($w_{i,j}$) của mỗi thuật ngữ trong ma trận đại diện cho tập hợp các tài liệu. Tập hợp toàn bộ các từ từ tất cả các tài liệu được gọi là **dictionary** hoặc **vocabulary**.

Việc tạo vector đặc trưng có thể xem xét nhiều phép biến đổi đặc trưng, chẳng hạn như:

- **Binary (Nhi phân):** Chèn "0" hoặc "1" vào ma trận tùy theo sự hiện diện hoặc vắng mặt của một thuật ngữ nhất định trong tài liệu.
- **Absolute frequency (Tần suất tuyệt đối):** Chèn tần suất xuất hiện của một thuật ngữ (token) trong tài liệu.

$$w_{ij} = \text{số lần xuất hiện từ thứ } j \text{ trong văn bản thứ } i.$$

- **Tần suất tương đối (Relative Frequency):** Tính tần suất tương đối xuất hiện của từ trong văn bản

$$\text{TF}_{i,j} = \frac{w_{ij}}{\sum_k w_{ik}}.$$

- **TF-IDF (Term Frequency–Inverse Document Frequency):** Là một phép tính trọng số phổ biến. IDF (Inverse Document Frequency) là số chuẩn hoá logarit của từ j trong số các văn bản:

$$\text{IDF}_j = \log \frac{N}{|\{d_i \in D : j \in d_i\}|},$$

trong đó N là tổng số văn bản, $|\{d_i \in D : j \in d_i\}|$ là số văn bản chứa token j .

TF-IDF xác định bởi công thức:

$$\text{TF-IDF}_{ij} = \text{TF}_{ij} * \text{IDF}_j.$$

Biểu diễn này nhằm tăng mức độ liên quan của các thuật ngữ có tính phân biệt (những thuật ngữ xuất hiện trong ít tài liệu hơn).

Biểu diễn Cú pháp (Syntactic Representation)

Biểu diễn cú pháp sử dụng một bộ **thẻ** (*tags*) được xác định trước để đại diện cho các phạm trù từ. Các phương pháp nổi tiếng là:

- **Linguistic Inquiry and Word Count (LIWC):** Là phần mềm được thiết kế để phân tích các đặc điểm **ngôn ngữ và tâm lý** của văn bản bằng cách tính đến các danh mục từ và kích thước ngôn ngữ được xác định trước.
- **Part-of-Speech (PoS) Tagging:** Gán các **phạm trù ngữ pháp** (*grammatical categories*) cho các từ (ví dụ: danh từ, tính từ 'JJ', đại từ, giới từ, v.v.), chỉ ra chức năng cú pháp của chúng.

Mặc dù tương tự, LIWC tập trung vào nội dung **tâm lý và cảm xúc**, trong khi PoS Tagging nhằm xác định **chức năng và phạm trù ngữ pháp** của các từ [20]. Cả hai đều có thể được sử dụng để cấu trúc văn bản cho các ứng dụng khai thác văn bản và NLP.

5.2.3 Phân tích Mô tả Dữ liệu Văn bản và Tài liệu

Ngoài các thước đo thống kê mô tả tiêu chuẩn, các thước đo sau đây là phù hợp cho văn bản:

- **Total number of words** (Word Count).
- **Number of distinct words** (Unique Word Count).
- **Vocabulary size** (Kích thước từ vựng)].
- **Average word length** (Độ dài từ trung bình).
- **Most common words** (Các từ phổ biến nhất) và **frequency distribution of words** (phân phối tần suất từ).
- **Sentence count** (Số lượng câu) và **average sentence length** (độ dài câu trung bình).
- **Stop word count** (Số lượng từ dừng).

Các thống kê mô tả này dựa trên việc đếm ký tự, từ hoặc câu trong tập dữ liệu văn bản.

Các thống kê đặc biệt khác cho văn bản bao gồm:

- **PoS Tagging distribution:** Phân phối tần suất của các loại ngữ pháp khác nhau (ví dụ: 'NN' cho danh từ, 'JJ' cho tính từ).
- **Readability measures (Thước đo dễ đọc):** Ước tính **mức độ khó đọc** của văn bản, thường được sử dụng để đánh giá cấp lớp (US grade level) cần thiết để hiểu văn bản [21]. Các chỉ số như **Flesch-Kincaid Grade Level (FKGL)** và **Automated Readability Index (ARI)** được sử dụng].
- **Co-occurrence matrix (Ma trận đồng xuất hiện):** Cho phép phân tích sự đồng xuất hiện của các từ.

5.2.4 Trực quan hóa Dữ liệu Văn bản và Tài liệu

Trực quan hóa văn bản là thuật ngữ được sử dụng cho các phương pháp tập trung vào dữ liệu văn bản thô, khác với các cách tiếp cận trực quan hóa kết quả của các thuật toán khai thác văn bản và NLP.

Các phương pháp trực quan hóa chính được xem xét bao gồm:

- **Word Clouds (Đám mây Từ):** Hữu ích để trực quan hóa **tần suất** hoặc **mức độ liên quan** của các từ.
- **Frequency Distribution (Phân phối Tần suất):** Có thể được biểu diễn bằng biểu đồ thanh (*bar chart*) cho các từ có tần suất cao nhất.
- **n-Grams:** Là chuỗi liên tục của n từ (ví dụ: bi-grams cho $n = 2$, tri-grams cho $n = 3$). Chúng có thể được trực quan hóa bằng các đám mây từ, sử dụng dấu gạch dưới “_” giữa các từ để chỉ ra rằng chúng thuộc cùng một chuỗi từ.
- **Categorical Heatmaps (Bản đồ Nhiệt Phân loại):** Có thể được sử dụng để trực quan hóa các **danh mục từ** (thể ngữ pháp). Một trực biến diễn các từ và trực kia biểu diễn các thể, và màu sắc đại diện cho tần suất tuyệt đối của các từ đó.
- **Dependency Parse Trees (Cây Phân tích Phụ thuộc):**
 - **Mục đích:** Khám phá và trực quan hóa **cấu trúc cú pháp phân cấp** của một câu, cho thấy mối quan hệ cú pháp giữa các từ.
 - **Cấu trúc:** Bao gồm **các nút** (đại diện cho từ), **các cạnh có hướng** (đại diện cho mối quan hệ giữa các từ), và **các nhãn phụ thuộc** (nhãn cạnh đại diện cho mối quan hệ ngữ pháp).
 - **Ứng dụng:** Được sử dụng trong nhiều phân tích dữ liệu văn bản, chẳng hạn như phân tích cú pháp, trích xuất thông tin, nhận dạng thực thể có tên, và dịch máy.

```
[13]: # Code to generate a Tag Cloud and a Frequency Distribution
# of the words in the IMDb corpus
```

```
import nltk
from nltk.corpus import movie_reviews
from nltk.probability import FreqDist
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Download the IMDb dataset and stopwords corpus
nltk.download('movie_reviews')
nltk.download('stopwords')

# Load the movie reviews dataset
```

```
documents = [ (movie_reviews.raw(fileid), category)
              for category in movie_reviews.categories()
              for fileid in movie_reviews.fileids(category) ]

# Concatenate all the reviews into a single text
all_reviews_text = " ".join([text for text, _ in documents])

# Tokenization and Preprocessing
tokens = word_tokenize(all_reviews_text.lower())
tokens = [token for token in tokens if token.isalpha()]
filtered_tokens = [token for token in tokens if token not in
                    stopwords.words('english')]

# Calculate word frequency
word_frequency = FreqDist(filtered_tokens)

# Generate the tag cloud
wordcloud = WordCloud(width=800, height=400,
                       background_color='white').
                       generate_from_frequencies(word_frequency)

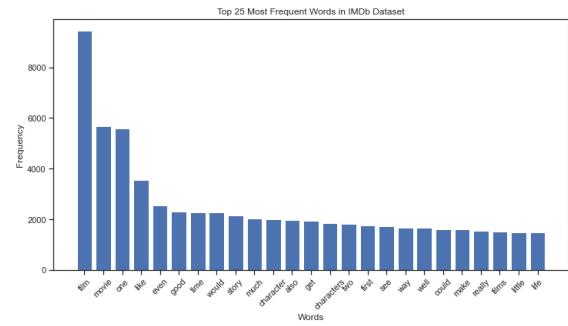
# Display the tag cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Tag Cloud for IMDb Dataset')
plt.show()

# Sort the word_frequency dictionary by frequency in descending
# order
sorted_word_frequency = dict(sorted(word_frequency.items(),
                                       key=lambda item: item[1], reverse=True))

# Get the first words and their frequencies
top_25_words = list(sorted_word_frequency.keys())[:25]
top_25_frequencies = list(sorted_word_frequency.values())[:25]

# Plot the bar chart for the first 20 words
plt.figure(figsize=(12, 6))
plt.bar(top_25_words, top_25_frequencies)
plt.title('Top 25 Most Frequent Words in IMDb Dataset')
plt.xlabel('Words')
```

```
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```



Hình 5.6: Word cloud và Bar chart các từ phổ biến nhất.

```
[14]: # Code to generate the bi-gram and the tri-gram
# for the IMDb corpus in NLTK

import nltk
from nltk.corpus import movie_reviews
from nltk.util import ngrams
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Download the IMDb dataset and stopwords corpus
nltk.download('movie_reviews')
nltk.download('stopwords')
nltk.download('punkt')

# Load the movie reviews dataset
documents = [(movie_reviews.raw(fileid), category)
              for category in movie_reviews.categories()
              for fileid in movie_reviews.fileids(category)]

# Select only 10% of the documents
num_documents = int(0.01 * len(documents))
documents = documents[:num_documents]

# Concatenate all the reviews into a single text
all_reviews_text = " ".join([text for text, _ in documents])
```

```
# Tokenization and Preprocessing
tokens = word_tokenize(all_reviews_text.lower())
tokens = [token for token in tokens if token.isalpha()]
filtered_tokens = [token for token in tokens if token not in
stopwords.words('english')]

# Generate 2-grams
bi_grams = list(ngrams(filtered_tokens, 2))
bi_gram_freq_dist = nltk.FreqDist([" ".join(gram) for gram in
bi_grams])

# Generate 3-grams
tri_grams = list(ngrams(filtered_tokens, 3))
tri_gram_freq_dist = nltk.FreqDist([" ".join(gram) for gram in
tri_grams])

# Generate the tag cloud for 2-grams
bi_gram_cloud = WordCloud(width=800, height=400,
background_color='white').
generate_from_frequencies(bi_gram_freq_dist)

# Generate the tag cloud for 3-grams
tri_gram_cloud = WordCloud(width=800, height=400,
background_color='white').
generate_from_frequencies(tri_gram_freq_dist)

# Display the tag clouds
plt.figure(figsize=(10, 5))
plt.imshow(bi_gram_cloud, interpolation='bilinear')
plt.axis('off')
plt.title('2-Gram Tag Cloud for IMDb Dataset')
plt.show()

plt.figure(figsize=(10, 5))
plt.imshow(tri_gram_cloud, interpolation='bilinear')
plt.axis('off')
plt.title('3-Gram Tag Cloud for IMDb Dataset')
plt.show()
```



Hình 5.7: Tag cloud với bi-grams và tri-grams.

```
[15]: # Code to generate the heatmap for POS Tags
# for the IMDb corpus

import nltk
from nltk.corpus import movie_reviews
from nltk.probability import FreqDist
import seaborn as sns
import matplotlib.pyplot as plt

# Download the necessary resources
nltk.download('movie_reviews')
nltk.download('stopwords')

# Load the IMDb movie reviews dataset
documents = [(movie_reviews.raw(fileid), category)
              for category in movie_reviews.categories()
              for fileid in movie_reviews.fileids(category)]

# Tokenization and Preprocessing
def preprocess_document(document):
    tokens = nltk.word_tokenize(document.lower())
    stop_words = set(nltk.corpus.stopwords.words('english'))
    tokens = [token for token in tokens if token.isalpha() and
              token not in stop_words]
    return tokens

# Perform POS tagging and calculate word frequency in each tag
# category
def calculate_word_frequency_with_pos(documents):
    word_frequency = {}
    all_tags = set()
```

```
for document in documents:
    tokens = preprocess_document(document[0])
    tagged_tokens = nltk.pos_tag(tokens)
    for word, pos_tag in tagged_tokens:
        if word not in word_frequency:
            word_frequency[word] = {}
        if pos_tag not in word_frequency[word]:
            word_frequency[word][pos_tag] = 0
        word_frequency[word][pos_tag] += 1
        all_tags.add(pos_tag)

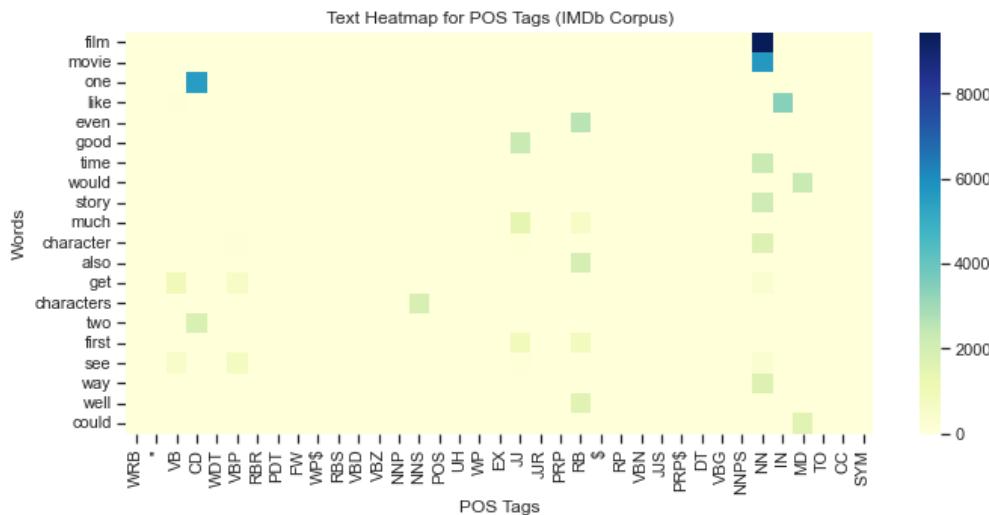
# Sort words by frequency and select the top 20
sorted_words = sorted(word_frequency.items(), key=lambda
    item: sum(item[1].values()), reverse=True)
top_20_words = [word for word, _ in sorted_words[:20]]

# Create the matrix
all_words = list(top_20_words)
matrix = []
for word in all_words:
    row = [word_frequency[word].get(tag, 0) for tag in
    all_tags]
    matrix.append(row)
return matrix, list(all_words), list(all_tags)

# Calculate word frequency with POS tags
matrix, words, tag_categories =
    calculate_word_frequency_with_pos(documents)

# Convert the matrix to a DataFrame for Seaborn heatmap
import pandas as pd
df = pd.DataFrame(matrix, index=words, columns=tag_categories)

# Plot the heatmap
plt.figure(figsize=(10, 5))
sns.heatmap(df, cmap='YlGnBu', annot=False, fmt='d', cbar=True)
plt.title('Text Heatmap for POS Tags (IMDb Corpus)')
plt.xlabel('POS Tags')
plt.ylabel('Words')
plt.tight_layout()
plt.show()
```



Hình 5.8: Categorical heatmap.

5.3 Cây và Mạng (Trees and Networks)

Cho đến phần này của sách, hầu hết các phương pháp được trình bày tập trung vào việc phân tích các biến đơn lẻ hoặc cặp biến độc lập. Tuy nhiên, có một lượng lớn **dữ liệu quan hệ** (*relational data*) được tạo ra, có thể dẫn đến những kiến thức hữu ích và sâu sắc. Dữ liệu quan hệ là dữ liệu (các đối tượng) bằng cách nào đó được **kết nối** với nhau. Các ví dụ bao gồm các nhóm người được kết nối qua mạng xã hội, các thành phố được kết nối bằng đường sá, hoặc những người được kết nối qua cây gia phả.

Lý thuyết Đồ thị (*Graph theory*) cung cấp một khuôn khổ chính thức để nghiên cứu các mối quan hệ giữa các mục, cho phép xác định các **nút** (hoặc đỉnh) và định nghĩa các **cạnh** (hoặc liên kết) giữa chúng. Các mối quan hệ này có thể có các thuộc tính như sức mạnh và hướng, dẫn đến việc phân tích dữ liệu quan hệ phức tạp.

Cây (*Trees*), ngược lại, là một trường hợp đặc biệt của đồ thị, đóng vai trò quan trọng trong các **cấu trúc phân cấp** (*hierarchical structures*). Chúng có các thuộc tính xác định, chẳng hạn như **một đường dẫn duy nhất** giữa mọi cặp nút và **không có chu trình** (*absence of cycles*). Cuối cùng, **Mạng lưới** (*Networks*) là các cấu trúc tổng quát hơn, có thể thể hiện các thuộc tính khác nhau, chẳng hạn như tính định hướng, chu trình và tính liên thông. Mục này tập trung vào phân tích khám phá (EDA) của cây và mạng lưới, xem xét các khái niệm từ lý thuyết đồ thị, kỹ thuật phân tích mô tả và các phương pháp trực quan hóa.

5.3.1 Các Khái niệm của Lý thuyết đồ thị (Concepts of Graph Theory)

- **Đồ thị (G):** Là một khuôn khổ được sử dụng để nghiên cứu mối quan hệ giữa các mục. Đồ thị G được định nghĩa là $G = (V, E)$, trong đó $V = \{v_1, v_2, \dots, v_N\}$ là tập hợp N đối tượng (nút hoặc đỉnh), và $E = \{e_1, e_2, \dots, e_m\}$ là tập hợp m cạnh.
- **Bậc (Degree):** Là số lượng cạnh nối với một đỉnh. Hai đỉnh được gọi là **kề nhau** (*adjacent*) nếu và chỉ nếu chúng được kết nối bởi cùng một cạnh.
- **Đồ thị Có trọng số (Weighted Graph):** Các khoảng cách hoặc chi phí được thêm vào các cạnh, đóng vai trò là trọng số.
- **Đồ thị Liên thông (Connected Graph):** Nếu tất cả các nút trong đồ thị được kết nối.
- **Cây (Tree):** Là một loại đồ thị đặc biệt, **không có chu trình** (*no cycles*), có một **đường dẫn duy nhất** giữa mỗi cặp nút, và **liên thông**. Cây tổng quát hóa khái niệm về mối quan hệ phân cấp.
- **Polytree:** Là cây có hướng được thêm vào các cạnh.
- **Đường đi (Path):** Một **walk** (*bước đi*) là một chuỗi luân phiên hữu hạn của các nút và cạnh. Một walk mở trong đó mỗi nút chỉ xuất hiện một lần trong chuỗi được gọi là **đường đi**.
- **Độ dài đường đi (Path Length):** Là số lượng cạnh (hoặc tổng trọng số của các cạnh) trong đường đi.

5.3.2 Mục tiêu phân tích khám phá Cây và Mạng (EDA)

- **Cây (Trees):** EDA lý tưởng để nghiên cứu **dữ liệu phân cấp** (*hierarchical data*), nơi có mối quan hệ cha-con rõ ràng giữa các phần tử.
- **Mạng lưới (Networks):** EDA phù hợp để khám phá **dữ liệu được kết nối liên thông** (*interconnected data*), nơi các phần tử có thể có nhiều mối quan hệ với nhau.
- **Mục tiêu chung:** Đạt được cái nhìn sâu sắc về cấu trúc và mối quan hệ, xác định các mẫu, xu hướng và dị thường, tóm tắt các đặc điểm chính, và chuẩn bị dữ liệu cho các phân tích sâu hơn (ví dụ: mô hình hóa và học máy).

- **Các thước đo đối với Cây:** EDA sẽ tập trung vào các thước đo như **chiều cao** (*height*), **bậc** (*degree*), **hệ số phân nhánh** (*branching factor*), **đường kính** (*diameter*), và **cấp độ** (*level*).
- **Các thước đo đối với Mạng lưới:** Bao gồm **hệ số gom cụm** (*clustering coefficient*), **mật độ** (*density*), **đường kính**, và các **thước đo độ trung tâm** (*centrality measures*) khác nhau.

```
[19]: import networkx as nx
import matplotlib.pyplot as plt

# Create a graph
G = nx.Graph()

# Add nodes (cities)
cities = ["Miami", "Orlando", "Tampa", "Jacksonville",
          "Tallahassee", "Fort Lauderdale", "Fort Myers"]

G.add_nodes_from(cities)

# Add edges (connections between cities) with distances
edges = [
    ("Miami", "Orlando", {"distance": 230}),
    ("Miami", "Tampa", {"distance": 280}),
    ("Miami", "Fort Lauderdale", {"distance": 30}),
    ("Orlando", "Tampa", {"distance": 85}),
    ("Tampa", "Jacksonville", {"distance": 200}),
    ("Jacksonville", "Tallahassee", {"distance": 160}),
    ("Tallahassee", "Orlando", {"distance": 270}),
    ("Tallahassee", "Tampa", {"distance": 230}),
    ("Fort Myers", "Tampa", {"distance": 140}),
    ("Fort Myers", "Miami", {"distance": 150}), # Distance
    ↪between Fort Myers and Miami
]

G.add_edges_from(edges)

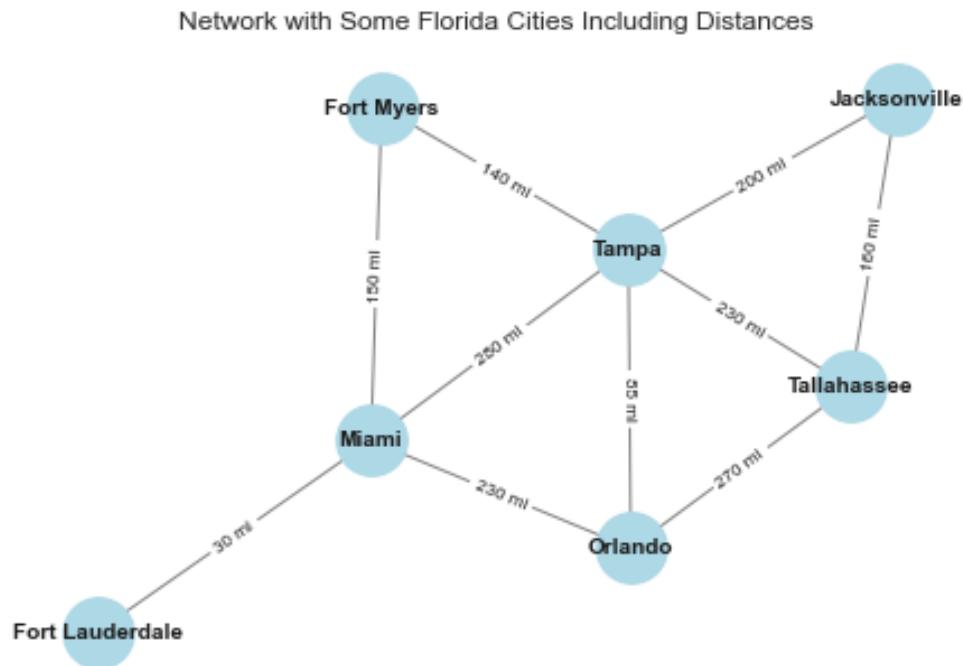
# Plot the network with distances as edge labels
pos = nx.spring_layout(G) # Positioning algorithm
nx.draw(G, pos, with_labels=True, node_size=1000,
        node_color='lightblue', font_size=10, font_weight='bold',
        edge_color='gray', arrowsize=20)
```

```

# Add edge labels (distances)
edge_labels = {(u, v): f"{attr['distance']} mi" for u, v, attr
               in G.edges(data=True)}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels,
                             font_size=8)

# Display the plot
plt.title("Network with Some Florida Cities Including
           Distances")
plt.show()

```



Hình 5.9: Mạng có trọng số.

```

[20]: # Script to generate a directed and weighted budget tree
import matplotlib.pyplot as plt
import networkx as nx

# Create a directed graph
G = nx.DiGraph()

# Add nodes (expense categories)

```

```

nodes = ["Gross Salary",
         "Expenses", "House", "Learn", "Leisure", "Market", "Utils",
         "College", "Others", "Power", "W&W", "Rental", "Maint", "Savings",
         "Invest", "Retire"]

# Add edges (expense relationships) with values
edges = [
    ("Gross Salary", "Expenses", {'value': 5000}),
    ("Gross Salary", "Savings", {'value': 1000}),
    ("Expenses", "House", {'value': 1500}),
    ("Expenses", "Learn", {'value': 1600}),
    ("Expenses", "Leisure", {'value': 600}),
    ("Expenses", "Market", {'value': 900}),
    ("Expenses", "Utils", {'value': 400}),
    ("Learn", "College", {'value': 1200}),
    ("Learn", "Others", {'value': 400}),
    ("Utils", "Power", {'value': 250}),
    ("Utils", "W&W", {'value': 150}),
    ("House", "Rental", {'value': 1200}),
    ("House", "Maint", {'value': 300}),
    ("Savings", "Invest", {'value': 400}),
    ("Savings", "Retire", {'value': 600}),
]

G.add_nodes_from(nodes)
G.add_edges_from(edges)

# Manually set node positions
node_positions = {
    "Gross Salary": (11, 12), "Expenses": (7, 8),
    "Savings": (19, 8), "House": (1, 4),
    "Learn": (7, 4), "Leisure": (10, 4),
    "Market": (13, 4), "Utils": (4, 4),
    "College": (8, 0), "Others": (12, 0),
    "Power": (2, 0), "W&W": (5, 0),
    "Rental": (-4, 0), "Maint": (-1, 0),
    "Invest": (17, 4), "Retire": (21, 4),
}

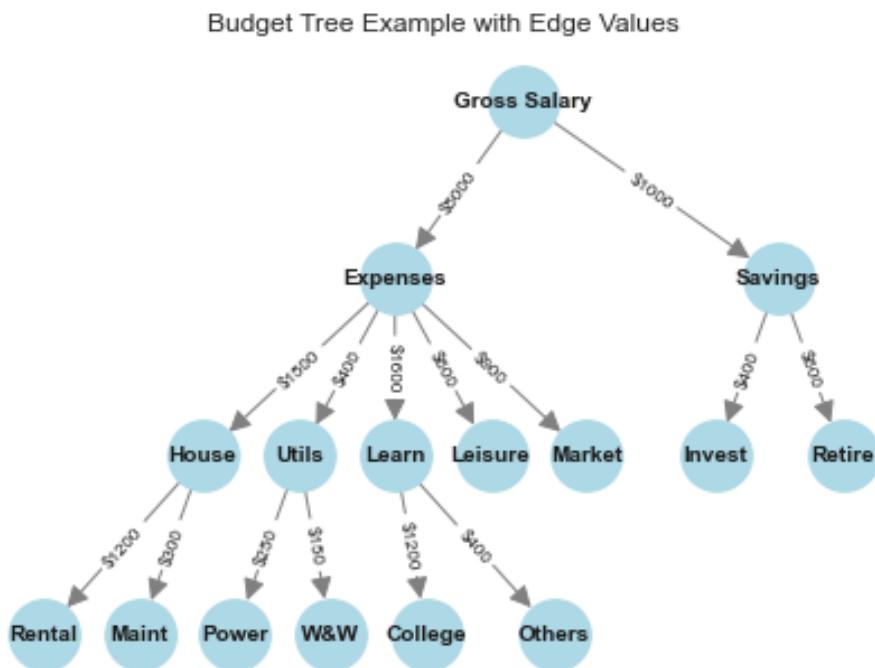
# Get edge labels from edge attributes
edge_labels = { (u, v): f"${attr['value']}"} for u, v, attr in G.
    edges(data=True) }
```

```

# Plot the expense tree with manual positions and edge labels
nx.draw(G, pos=node_positions, with_labels=True, node_size=1000,
        node_color='lightblue',
        font_size=10, font_weight='bold', edge_color='gray',
        arrowsize=20)
nx.draw_networkx_edge_labels(G, pos=node_positions,
                             edge_labels=edge_labels, font_size=8)

# Display the plot
plt.title("Budget Tree Example with Edge Values")
#plt.title("Budget Tree Example")
plt.axis("off")
plt.show()

```



Hình 5.10: Cây.

5.3.3 Phân tích Mô tả đối với Cây (Descriptive Analysis for Trees)

- **Cấu trúc Cây:** Một số cây là **có gốc** (*rooted*), bắt đầu từ một nút gốc (*root node*) và phân nhánh ra, kết thúc ở các nút lá (*leaf nodes*). Nút mà từ đó một nút khác xuất

phát được gọi là **nút cha** (*parent node*), và nút xuất phát là **nút con** (*child node*). Cây thường được biểu diễn ngược, với gốc ở trên cùng và lá ở phía dưới.

- **Các Thước đo Chính:**

- **Số lượng nút:** Số lượng nút (nodes) của cây.
- **Số lượng Cạnh:** Số lượng cạnh của cây
- **Chiều cao của cây:** Chiều cao đo đường đi dài nhất từ nút gốc đến nút lá.
- **Bậc vào (In-Degree)** và **Bậc ra (Out-Degree)** của mỗi nút.
- **Hệ số phân nhánh (Branching factor):** trung bình của số bậc ra của các nút.
- **Đường kính cây (Tree Diameter):** Đường kính đo đường đi dài nhất giữa bất kỳ hai nút nào.
- **Độ sâu (Level):** Độ sâu từ nút gốc .

5.3.4 Trực quan hóa Cây (Visualizing Trees)

Cây có thể được trực quan hóa bằng các thuật toán **không lấp đầy không gian** (*non-space-filling*) hoặc **lấp đầy không gian** (*space-filling*).

Phương pháp Không Lấp đầy Không gian (Non-Space-Filling Methods)

Các phương pháp này rất phổ biến, thường được sử dụng trong khoa học xã hội và máy tính. Chúng có thể bao gồm định dạng cấu trúc (sử dụng thut lề, tương tự như thư mục máy tính) hoặc biểu diễn trực quan, nơi mỗi nút (hình chữ nhật) đại diện cho một phần tử và các kết nối liên kết cha với con.

```
[26]: # Code to generate partial genealogic trees of Queen Elizabeth
      ↵II
      # With internal functions

import matplotlib.pyplot as plt

# Define the family members and relationships
family_tree = {
    "Queen Elizabeth II": ["Prince Charles", "Princess Anne",
                           "Prince Andrew", "Prince Edward"],
    "Prince Philip": ["Prince Charles", "Princess Anne", "Prince
                      Andrew", "Prince Edward"],
```

```

"Prince Charles": ["Prince William", "Prince Henry"],
"Princess Anne": [],
"Prince Andrew": [],
"Prince Edward": [],
"Prince William": ["Prince George", "Princess Charlotte",
"Prince Louis"],
}

def plot_genealogy(node, depth=0):
    print(" " * depth + "|_" + node)
    if node in family_tree:
        for child in family_tree[node]:
            plot_genealogy(child, depth + 1)

if __name__ == "__main__":
    print("Genealogical Tree:")
    plot_genealogy("Queen Elizabeth II")

    # Plot using Matplotlib
    fig, ax = plt.subplots(figsize=(8, 6))
    ax.set_xlim(0, 10)
    ax.set_ylim(0, 6)

    def plot_tree(node, x, y, level=0):
        ax.text(x, y, node, ha='center', va='center',
                bbox=dict(facecolor='lightblue',
                           edgecolor='gray', boxstyle='round,pad=0.3'))
        if node in family_tree:
            num_children = len(family_tree[node])
            child_spacing = 18 / (num_children + 6)
            for i, child in enumerate(family_tree[node]):
                child_x = x + (i - (num_children - 1) / 3) *
                child_spacing
                child_y = y - 1
                ax.plot([x, child_x], [y, child_y],
                        color='gray')
                plot_tree(child, child_x, child_y, level + 1)

    plot_tree("Queen Elizabeth II", 5, 5.5)
    plt.title("Genealogical Tree Visualization")
    plt.axis("off")
    plt.show()

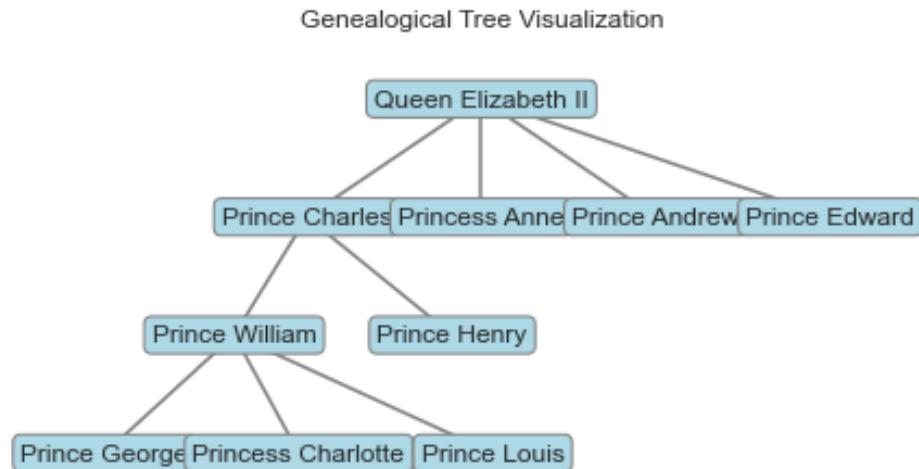
```

Genealogical Tree:

```

|_Queen Elizabeth II
 |_Prince Charles
   |_Prince William
     |_Prince George
     |_Princess Charlotte
     |_Prince Louis
   |_Prince Henry
 |_Princess Anne
 |_Prince Andrew
 |_Prince Edward

```



Hình 5.11: Trực quan hóa Cây.

Phương pháp Lấp đầy Không gian (Space-Filling Methods)

Các phương pháp này sử dụng sự sắp xếp cạnh nhau (*juxtapositioning*) để tạo ra một biểu diễn của cấu trúc phân cấp, tối đa hóa khu vực hiển thị.

- **Treemap (Biểu đồ Hình cây):** Trực quan hóa dữ liệu phân cấp trong các hình chữ nhật lồng nhau và loại trừ lẫn nhau, với kích thước khác nhau. Treemap có thể được coi là phiên bản vuông của biểu đồ tròn. Kích thước của hình chữ nhật tương ứng với **mức độ liên quan** (tỷ lệ) của chúng trong tổng thể.
- **Sunburst Chart (Biểu đồ Vòng mặt trời):** Hiển thị nút gốc (ví dụ: "Gross Salary") ở **vòng trung tâm**, với mỗi cấp độ phân cấp tiếp theo được vẽ đồng tâm hướng ra ngoài. Biểu đồ sử dụng màu sắc để phân biệt các nhánh và hiển thị giá trị tuyệt đối cũng như tương đối của các nút bên trong mỗi lát cắt.

[23]: # Treemap example with the Budget Tree synthetic data

```
import plotly.express as px

# Define data for the treemap
data = {
    'labels': ['Gross
Salary', 'Expenses', 'Savings', 'House', 'Utils', 'Learn', 'Leisure', 'Market'
Invest', 'Retire', 'Rental', 'Maint', 'Power', 'W&W', 'College', 'Others'],
    'parents': ['', 'Gross Salary', 'Gross Salary', 'Expenses',
'Expenses', 'Expenses',
'Expenses', 'Expenses', 'Savings', 'Savings',
House', 'House',
'Utils', 'Utils', 'Learn', 'Learn'],
    'values': [6000, 5000, 1000, 1500, 400, 1600, 600, 900, 400,
600, 1200, 300, 250,
150, 1200, 400]
}

# Create a treemap
fig = px.treemap(data, names='labels', parents='parents', values='values', branchval

# Update font size of labels
fig.update_traces(textinfo='label+percent entry+value') # Add
labels parameter here
fig.update_layout(title_x=.5)

# Show the chart
fig.show()
```

```
[25]: # Sunburst example with the Budget Tree synthetic data

import plotly.graph_objects as go

# Define data for the sunburst chart
labels = ['Gross
Salary', 'Expenses', 'Savings', 'House', 'Utils', 'Learn', 'Leisure', 'Market'
Invest', 'Retire', 'Rental', 'Maint', 'Power', 'W&W', 'College', 'Others']
parents = ['', 'Gross Salary', 'Gross
Salary', 'Expenses', 'Expenses', 'Expenses',
'Expenses', 'Savings', 'Savings', 'House', 'House', 'Utils', 'Utils', 'Learn',
values = [6000, 5000, 1000, 1500, 400, 1600, 600, 900, 400, 600,
1200, 300, 250, 150, 1200, 400]

# Create a sunburst chart
fig = go.Figure(go.
    Sunburst(labels=labels, parents=parents, values=values, branchvalues="total"))

# Set the title
fig.update_traces(textinfo='label+percent entry+value')
fig.update_layout(title="Sunburst Chart Example", title_x=.5)

# Show the chart
fig.show()
```

5.3.5 Phân tích Mô tả đối với Mạng (Descriptive Analysis for Networks)

Mạng là dạng tổng quát hóa của cây. Các thước đo mô tả mạng lưới bao gồm:

Các thước đo chung

- **Số lượng nút (N) và Cạnh (E):** Tổng số nút và cạnh cấu thành mạng.
- **Hệ số gom cụm bộ (C_i):** Đo lường mức độ mà các nút lân cận của một nút đã cho liên kết với nhau, phản ánh sự hình thành cộng đồng hoặc cụm. Được định nghĩa là:

$$C_i = \frac{2L_i}{k_i(k_i - 1)}$$

trong đó L_i là số cạnh giữa k_i nút lân cận của nút i .

- **Hệ số Gom cụm Trung bình ($\langle C \rangle$):** Là giá trị trung bình của hệ số gom cụm cục bộ của tất cả các nút trong mạng lưới.

$$\langle C \rangle = \frac{1}{N} \sum_{i=1}^N C_i.$$

- **Mật độ Mạng lưới (Network Density):** Tỷ lệ giữa số lượng cạnh thực tế trong mạng lưới (E) và số lượng cạnh tối đa có thể có (E_{max}):

$$d = E/E_{max}$$

- **Độ dài Đường đi Ngắn nhất (Shortest path length):** Số lượng cạnh nhỏ nhất (hoặc tổng trọng số nhỏ nhất cho mạng có trọng số) cần thiết để đi từ nút này sang nút khác.
- **Đường kính Mạng lưới (Network diameter):** Giá trị của khoảng cách dài nhất trong mạng lưới.
- **Hệ số Hỗn hợp/Đồng loại (Assortativity coefficient):** Định lượng mức độ mà các nút có đặc điểm **tương tự** có xu hướng kết nối (hệ số dương) hoặc các nút có đặc điểm **không tương đồng** có xu hướng kết nối (hệ số âm).

Các thước đo độ trung tâm với mạng (Centrality Measures)

- **Độ trung tâm giữa (Betweenness centrality - $C_B(e)$):** Đo lường mức độ một nút hoạt động như một cầu nối, bằng cách tính tổng số đường đi ngắn nhất giữa hai nút bất kỳ đi qua cạnh e . Nút có độ trung tâm giữa cao tạo điều kiện truyền thông tin giữa các phần khác nhau của mạng lưới.

$$C_B(e) = \sum_{s \neq t \neq e \in E} \frac{\sigma(s, t|e)}{\sigma(s, t)},$$

trong đó $\sigma(s, t|e)$ là tổng số đường đi ngắn nhất từ s đến t và đi qua e , và $\sigma(s, t)$ là số đường đi ngắn nhất giữa s và t .

- **Độ trung tâm gần (Closeness centrality - $C_{cl}(e)$):** Giả định rằng độ trung tâm là thước đo "sự gần gũi" của một nút với nhiều nút khác.

$$C_{cl}(e) = \frac{1}{\sum_{u \in E} d(e, u)},$$

trong đó $d(e, u)$ số cạnh của đường ngắn nhất giữa e và u .

- **Độ Trung tâm Vector riêng (Eigenvector centrality):** Tính toán độ trung tâm của một nút dựa trên độ trung tâm của các nút lân cận của nó. Nút có độ trung tâm gần cao có vị trí chiến lược để giao tiếp và lan truyền thông tin.

$$C_{Gi}(e) = \alpha \sum_{\{u,e\} \in G} C_{Gi}(u),$$

trong đó $C_{Gi}(e)$ là nghiệm của bài toán tìm véc-tơ riêng.

```
[26]: # Code to calculate Descriptive Statistics of the
# Zachary's Karate Club Social Network

import networkx as nx

# Load the Zachary's Karate Club dataset
G = nx.karate_club_graph()

# Network Data Statistics
print("Is the graph a tree?", nx.is_tree(G))
print("Number of nodes:", G.number_of_nodes())
print("Number of edges:", G.number_of_edges())
print("Is the graph directed?", G.is_directed())
print("Is the graph connected?", nx.is_connected(G))
print("Average clustering coefficient:", nx.
      average_clustering(G))
print("Average shortest path length:", nx.
      average_shortest_path_length(G))
print("Number of connected components:", nx.
      number_connected_components(G))
print("Density:", nx.density(G))
print("Maximum degree:", max(dict(G.degree()).values()))
print("Minimum degree:", min(dict(G.degree()).values()))
print("Average degree:", sum(dict(G.degree()).values()) / G.
      number_of_nodes())
print("Assortativity coefficient:", nx.assortativity.
      degree_assortativity_coefficient(G))
print("Degree centrality:")
for node, centrality in nx.degree_centrality(G).items():
    print(f"Node {node}: {centrality:.4f}")
print("Betweenness centrality:")
for node, centrality in nx.betweenness_centrality(G).items():
    print(f"Node {node}: {centrality:.4f}")
print("Closeness centrality:")
```

```

for node, centrality in nx.closeness_centrality(G).items():
    print(f"Node {node}: {centrality:.4f}")
print("Eigenvector centrality:")
for node, centrality in nx.eigenvector_centrality(G).items():
    print(f"Node {node}: {centrality:.4f}")

```

5.3.6 Trực quan hóa mạng (Visualizing Networks)

Trực quan hóa mạng lưới thường phức tạp do sự kết hợp lớn các khả năng (có trọng số, có hướng, có chu trình, v.v.). Các phương pháp chính là **biểu đồ liên kết nút** (*node-link diagram*) với các bố cục khác nhau và hiển thị ma trận.

- **Bố cục Spring layout** (*Spring layout*): Định vị các nút bằng cách mô phỏng biểu diễn hướng lực, nhằm **tối ưu hóa sự phân bố** của các nút trong không gian để trực quan hóa tốt hơn.
- **Bố cục tròn** (*Circular layout*): Vẽ mạng lưới theo định dạng tròn.
- **Bố cục Vỏ** (*Shell layout*): Định vị các nút trong các vòng tròn đồng tâm.

Các bố cục này thường được tạo ra bằng thư viện NetworkX. Có thể tùy chỉnh vị trí nút để nhấn mạnh các đặc điểm mong muốn, ví dụ như trải rộng các nút có bậc cao nhất để làm nổi bật chúng.

```
[28]: # Code to Visualize the Zachary's Karate Club Social Network in
      ↵different layouts

import networkx as nx
import matplotlib.pyplot as plt

# Load the Zachary's Karate Club dataset
G = nx.karate_club_graph()

# Plot using spring layout
pos_spring = nx.spring_layout(G, seed=4)
plt.figure(figsize=(10, 25))

plt.subplot(411)
nx.draw_networkx(G, pos_spring, node_color='lightblue',
                  font_size=8, edge_color='gray')
plt.title("Spring Layout")
```

```
# Plot using circular layout
pos_circular = nx.circular_layout(G)
plt.subplot(412)
nx.draw_networkx(G, pos_circular, node_color='lightblue',
                  font_size=8, edge_color='gray')
plt.title("Circular Layout")

# Plot using shell Layout
shell_layout = [list(range(0, 17)), list(range(17, 34))]
pos_shell = nx.shell_layout(G, nlist=shell_layout)
plt.subplot(413)
nx.draw_networkx(G, pos_shell, node_color='lightblue',
                  font_size=8, edge_color='gray')
plt.title("Shell Layout")

# Plot using spring layout with custom positions
pos_custom = nx.spring_layout(G, seed=4, iterations=200)
custom_positions = {0: (0.0, 1.5), 1: (.5, 1.5), 2: (0.5, -1.5),
                     33: (0.0, -1.5)}
pos_custom.update(custom_positions)
plt.subplot(414)
nx.draw_networkx(G, pos_custom, node_color='lightblue',
                  font_size=8, edge_color='gray')
plt.title("Custom Spring Layout")

plt.show()
```

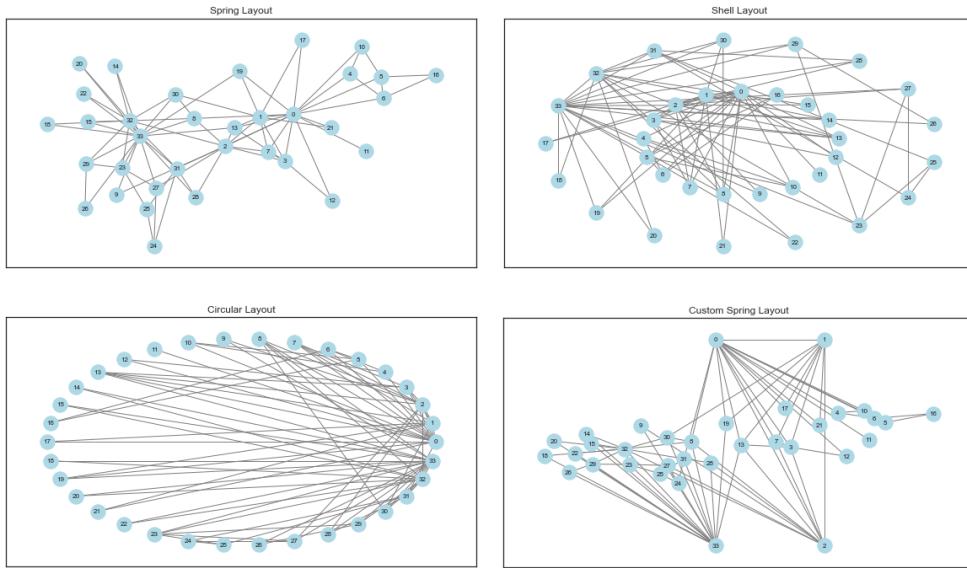
```
[27]: # Code to plot the heatmap for the Zachary's data

import networkx as nx
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Zachary's Karate Club dataset
G = nx.karate_club_graph()

# Plot the social network
pos = nx.spring_layout(G, seed=4)    # Positioning algorithm with
                                         # fixed seed for reproducibility

# Heatmap
```



Hình 5.12: Trực quan hóa Mạng.

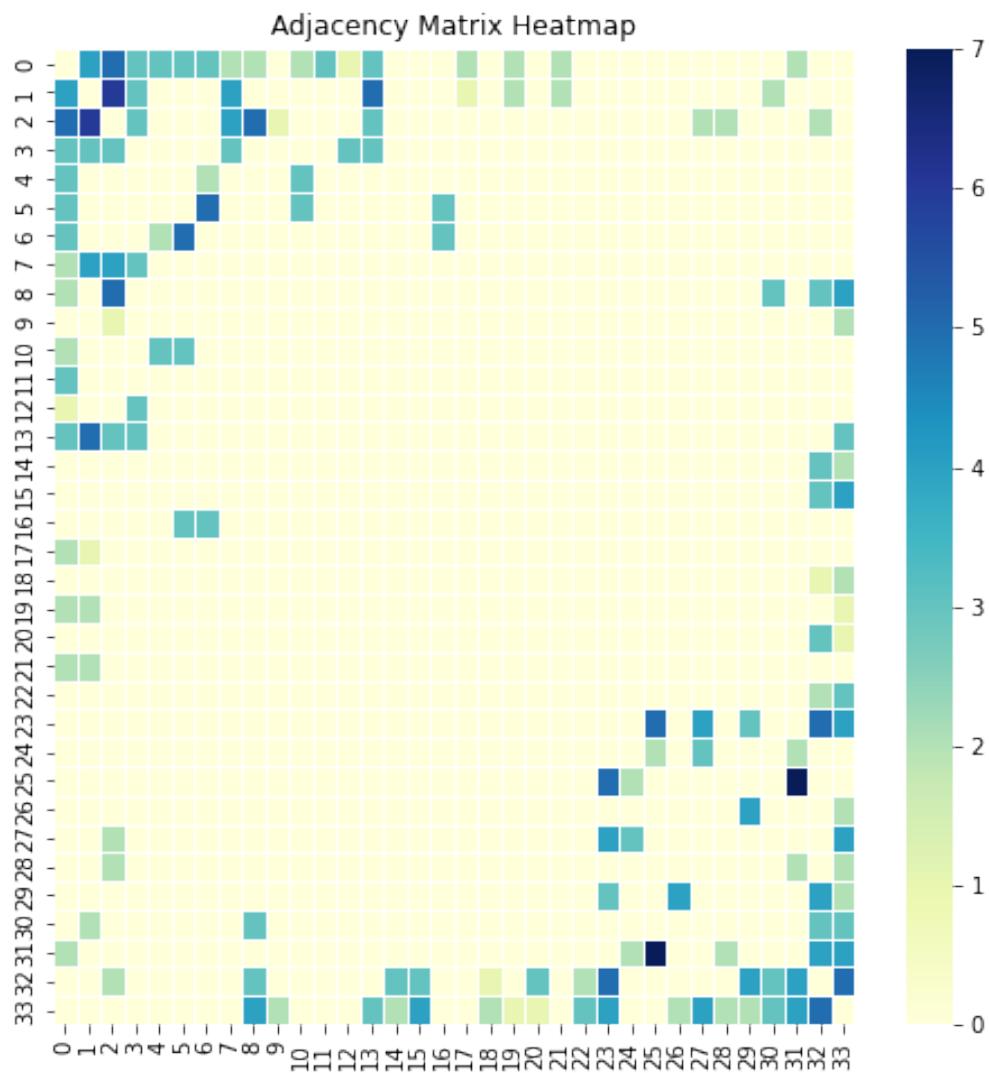
```
adjacency_matrix = nx.to_numpy_matrix(G)
plt.figure(figsize=(8, 8))
sns.heatmap(adjacency_matrix, cmap='YlGnBu', linewidths=0.5,
            annot=False)
plt.title("Adjacency Matrix Heatmap")
plt.show()
```

5.4 Dữ liệu nhiều chiều

Dữ liệu nhiều chiều (multivariate/high-dimensional data) là dữ liệu mà mỗi quan sát có từ 5 biến trở lên, thường lên đến hàng trăm–hàng nghìn biến (single-cell genomics, văn bản embedding, sensor IoT, tài chính, ...).

Con người chỉ cảm nhận tự nhiên không gian 3 chiều và khoảng 7-10 kênh thị giác. Do đó, mọi phương pháp trực quan hóa nhiều chiều đều phải thực hiện một trong ba chiến lược chính:

- Giảm số chiều xuống 2–3 chiều (dimensionality reduction)
- Chia nhỏ thành nhiều view 2D và liên kết chúng (multiple coordinated views)
- Dùng các kênh thị giác thay thế (glyphs, pixel-oriented, parallel coordinates, ...)



Hình 5.13: Trực quan hoá Mạng bằng ma trận kế.

5.4.1 Hệ toạ độ song song (Parallel coordinates)

1. Ý tưởng cốt lõi Phương pháp **Hệ Tọa độ Song song** là một kỹ thuật trực quan hóa dữ liệu đa chiều (p chiều) bằng cách ánh xạ dữ liệu lên một mặt phẳng $2D$. Kỹ thuật này được đề xuất bởi Alfred Inselberg.

- Mỗi chiều (biên) $i \in \{1, 2, \dots, p\}$ được đại diện bởi một trục số thẳng đứng V_i .
- Tất cả p trục V_1, V_2, \dots, V_p được đặt song song và cách đều nhau trên mặt phẳng $2D$.
- Mỗi quan sát (điểm dữ liệu) $\mathbf{P} = (x_1, x_2, \dots, x_p)$ trong không gian p -chiều được biểu

diễn bằng một **đường gấp khúc** duy nhất, cắt mỗi trục V_i tại vị trí tương ứng với giá trị x_i .

Chuẩn hóa Dữ liệu (Normalization) Do các biến có thể có các đơn vị và phạm vi khác nhau, dữ liệu cần được chuẩn hóa (ví dụ: Min-Max scaling) về phạm vi thống nhất $[0, 1]$ để đảm bảo tính so sánh và cân bằng thị giác:

$$x'_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

Trong đó x'_i là giá trị đã chuẩn hóa của chiều i .

Mã hóa Điểm Dữ liệu Điểm dữ liệu P được biểu diễn như sau:

- Giả sử các trục V_1, \dots, V_p nằm cách đều nhau đọc theo trục ngang (trục hoành) tại các vị trí t_1, t_2, \dots, t_N .
- Điểm dữ liệu P được vẽ bằng một đường nối các điểm (t_i, x'_i) cho $i = 1$ đến p .

Điền giải và Khám phá Mối quan hệ Mối quan hệ giữa hai chiều kề nhau (V_i và V_{i+1}) được suy luận từ hình dạng của các đường giữa hai trục đó.

Phân loại Mối quan hệ

- **Tương quan Dương (Thuận):** Nếu các đường giữa V_i và V_{i+1} có xu hướng **song song** và không giao nhau. Khi x'_i tăng, x'_{i+1} cũng tăng.
- **Tương quan Âm (Nghịch):** Nếu các đường giữa V_i và V_{i+1} có xu hướng **cắt nhau** tại một điểm nằm giữa hai trục. Khi x'_i tăng, x'_{i+1} có xu hướng giảm.
- **Không tương quan:** Các đường giao nhau lộn xộn, không theo một mẫu hình rõ ràng.

Phát hiện Cụm và Ngoại lai

- **Cụm (Clusters):** Được biểu thị bằng các **dải hẹp** của các đường đi cùng nhau xuyên qua nhiều trục kề nhau.
- **Ngoại lai (Outliers):** Được biểu thị bằng các đường **riêng rẽ** hoặc có hình dạng khác biệt rõ rệt so với đại đa số các đường khác.

Hạn chế

- **Quá tải (Overplotting):** Với số lượng lớn điểm dữ liệu, các đường bị chồng lên nhau dày đặc, gây khó khăn cho việc nhận diện mẫu hình.
- **Độ nhạy Thứ tự Trục:** Thứ tự sắp xếp các trục ảnh hưởng lớn đến các mối quan hệ được nhìn thấy. Việc tìm kiếm thứ tự tối ưu là một thách thức.

Các kỹ thuật hiện đại làm Parallel Coordinates trở thành “vua tương tác” 2025

- Axis reordering (kéo thả trục)
- Brushing & linking
- Edge bundling / curve smoothing
- Opacity + density coloring
- Angular brushing
- Axis inversion (đảo trục)

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from pandas.plotting import parallel_coordinates
from sklearn.preprocessing import MinMaxScaler
import numpy as np

iris = load_iris()
iris_df = pd.DataFrame(data=iris.data, columns=iris.
    feature_names)
iris_df['species'] = iris.target_names[iris.target]
feature_cols = iris.feature_names
# Khởi tạo MinMaxScaler
scaler = MinMaxScaler()

iris_scaled_data = scaler.fit_transform(iris_df[feature_cols])
iris_scaled_df = pd.DataFrame(iris_scaled_data,
    columns=feature_cols)

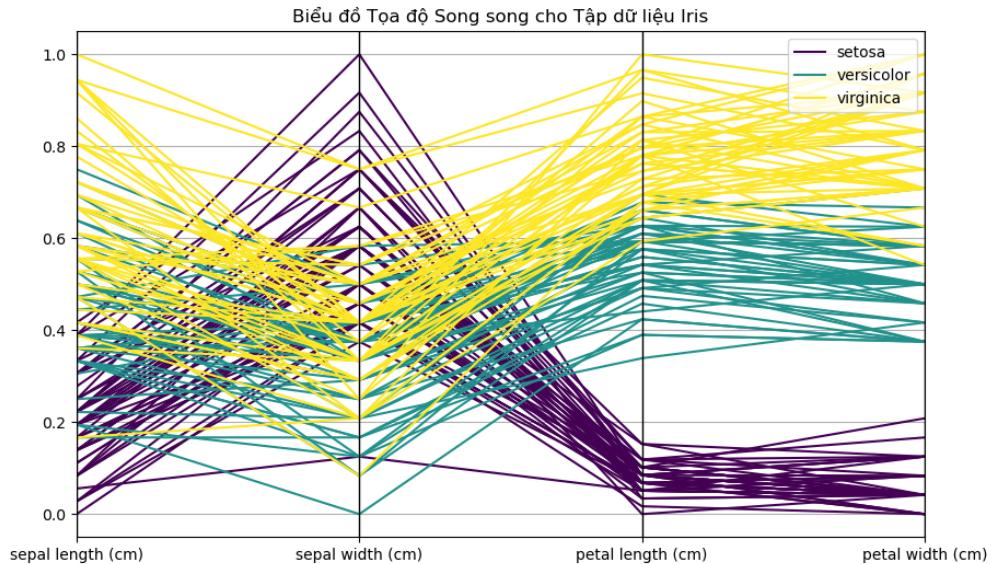
iris_scaled_df['species'] = iris_df['species']
```

```

plt.figure(figsize=(10, 6))

parallel_coordinates(iris_scaled_df, 'species',
                     colormap=plt.cm.get_cmap('viridis'))
plt.title('Biểu đồ Tọa độ Song song cho Tập dữ liệu Iris')
plt.grid(True)
plt.show()

```



Hình 5.14: Hệ tọa độ song song.

5.4.2 Hệ tọa độ hình sao (Star coordinates)

Star Coordinates (Eser Kandogan, 2000–2001) là một kỹ thuật trực quan hoá dữ liệu nhiều chiều **trực tiếp** (direct visualization), **không giảm chiều**, rất dễ hiểu và đặc biệt mạnh khi có **tương tác**. Đến năm 2025, phương pháp này vẫn được dùng rộng rãi trong Tableau, Spotfire, Orange, Power BI Custom Visuals và nhiều dashboard khám phá dữ liệu.

Ý tưởng cốt lõi

- Mỗi biến (chiều) được biểu diễn bằng một trục đơn vị tỏa ra từ tâm theo dạng hình sao (góc đều quanh vòng tròn 360°).

- Mỗi **quan sát** (một dòng dữ liệu) được ánh xạ thành một **điểm 2D** bằng cách cộng vectơ có hướng theo từng trục, độ dài tỉ lệ với giá trị đã chuẩn hoá của quan sát đó trên chiều tương ứng.
- Kết quả: tất cả các điểm dữ liệu được hiển thị đồng thời trong một không gian 2D duy nhất mà không mất thông tin.

Cho dữ liệu n quan sát, p chiều: $\mathbf{X} \in \mathbb{R}^{n \times p}$ đã chuẩn hoá về $[0, 1]$ hoặc $[-1, 1]$.

- a. Tạo p vectơ đơn vị đều quanh gốc:

$$\mathbf{C}_i = (\cos \theta_i, \sin \theta_i), \quad \theta_i = \frac{2\pi(i-1)}{p}, \quad i = 1, \dots, p$$

- b. Toạ độ 2D của quan sát thứ k :

$$\mathbf{p}_k = \sum_{j=1}^p x_{kj} \cdot \mathbf{C}_j \quad (= \mathbf{x}_k \cdot \mathbf{C})$$

Phiên bản tương tác hiện đại – **điểm mạnh nhất** của Star Coordinates

- Người dùng có thể **kéo thả, xoay, co giãn, thay đổi thứ tự** từng trục bằng chuột.
- Khi thay đổi một trục → toàn bộ đám điểm di chuyển ngay lập tức → giúp nhanh chóng phát hiện:
 - Nhóm (cluster)
 - Tương quan giữa các chiều
 - Outlier
 - Cách sắp xếp trục tối ưu để tách nhóm rõ nhất

```
[2]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
from sklearn.preprocessing import MinMaxScaler

def plot_star_coordinates(df_scaled, class_col, features):
    N = len(features)
    angles = np.linspace(0, 2 * np.pi, N, endpoint=False)
    x_coords = np.sum(df_scaled[features].values * np.
                      cos(angles), axis=1)
```

```

y_coords = np.sum(df_scaled[features].values * np.
sin(angles), axis=1)

fig, ax = plt.subplots(figsize=(10, 10))
ax.set_title('Biểu đồ Tọa độ Hình sao cho Dữ liệu Wine (Đã
Chuẩn hóa)',

fontsize=14, fontweight='bold')

vector_x = np.cos(angles)
vector_y = np.sin(angles)

for i in range(N):
    ax.plot([0, vector_x[i]], [0, vector_y[i]],
color='gray')

circle = plt.Circle((0, 0), 1, color='lightgray',
fill=False)
ax.add_artist(circle)

unique_classes = df_scaled[class_col].unique()
colors = plt.cm.get_cmap('rainbow', len(unique_classes))

for i, cls in enumerate(unique_classes):
    subset = df_scaled[df_scaled[class_col] == cls]
    x_subset = np.sum(subset[features].values * np.
cos(angles), axis=1)
    y_subset = np.sum(subset[features].values * np.
sin(angles), axis=1)

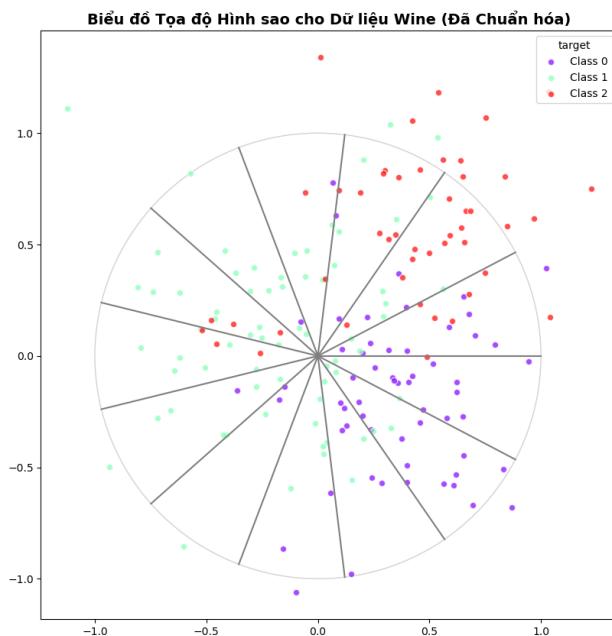
    ax.scatter(x_subset, y_subset,
               color=colors(i),
               label=f'Class {cls}',
               alpha=0.7, edgecolors='w')

ax.set_aspect('equal', adjustable='box') # Giữ tỉ lệ trục 1:
# 1
ax.legend(title=class_col, loc='upper right')
plt.show()

wine = load_wine()
wine_df = pd.DataFrame(data=wine.data, columns=wine.
feature_names)

```

```
wine_df['target'] = wine.target
feature_cols = wine.feature_names
scaler = MinMaxScaler()
wine_scaled_data = scaler.fit_transform(wine_df[feature_cols])
wine_scaled_df = pd.DataFrame(wine_scaled_data,
    columns=feature_cols)
wine_scaled_df['target'] = wine_df['target']
plot_star_coordinates(wine_scaled_df, 'target', feature_cols)
```



Hình 5.15: Hệ tọa độ hình sao.

5.4.3 Phương pháp chiêu xuyên tâm (Radviz)

Radviz là một kỹ thuật trực quan hóa nhằm ánh xạ dữ liệu p -chiều $\mathbf{D} = \{P_1, P_2, \dots, P_n\}$ thành các điểm trên không gian 2D.

Mô hình Lực Lò xo (Spring Model)

- Mỗi chiều (biến) $i \in \{1, 2, \dots, p\}$ được đặt tại một điểm neo A_i nằm đều trên chu vi của một vòng tròn đơn vị (hoặc bán kính R).
- Mỗi điểm dữ liệu P_j được xem như một vật thể được gắn với N điểm neo A_i bằng các lò xo tương ứng.

- Lực kéo của lò xo từ \mathbf{A}_i đến P_j tỷ lệ thuận với giá trị chuẩn hóa của chiều i của điểm dữ liệu P_j .

Chuẩn hóa Dữ liệu Trước tiên, dữ liệu cần được chuẩn hóa (thường là Min-Max scaling) về phạm vi $[0, 1]$:

$$x_{i,j} = \frac{x_{i,j} - \min(x_i)}{\max(x_i) - \min(x_i)}$$

Trong đó:

- $x_{i,j}$: Giá trị gốc của chiều i cho điểm dữ liệu P_j .
- $x_{i,j}$: Giá trị chuẩn hóa (lực kéo) của chiều i cho điểm P_j .

Xác định Vị trí các Điểm Neo Các điểm neo \mathbf{A}_i được đặt đều trên một vòng tròn bán kính R :

$$\mathbf{A}_i = \left(R \cos\left(\frac{2\pi(i-1)}{N}\right), R \sin\left(\frac{2\pi(i-1)}{N}\right) \right)$$

Thường $R = 1$ cho vòng tròn đơn vị.

Tính toán Vị trí Điểm Dữ liệu Vị trí cuối cùng $\mathbf{P}_j = (x_j, y_j)$ của điểm dữ liệu P_j là điểm cân bằng của tất cả các lực kéo từ các lò xo. Vị trí này được tính bằng công thức trung bình có trọng số của vị trí các điểm neo:

$$\mathbf{P}_j = \frac{\sum_{i=1}^p x_{i,j} \cdot \mathbf{A}_i}{\sum_{i=1}^p x_{i,j}}$$

Công thức Tọa độ Chi tiết Tọa độ x và y của điểm \mathbf{P}_j :

$$x_j = \frac{\sum_{i=1}^p x_{i,j} \cdot A_{xi}}{\sum_{i=1}^p x_{i,j}}$$

$$y_j = \frac{\sum_{i=1}^p x_{i,j} \cdot A_{yi}}{\sum_{i=1}^p x_{i,j}}$$

Trong đó (x_{A_i}, y_{A_i}) là tọa độ của điểm neo \mathbf{A}_i .

Diễn giải Trực quan

- Nếu \mathbf{P}_j nằm gần $\mathbf{A}_i = (Ax_i, Ay_i)$: Chiều i có giá trị $x_{i,j} \approx 1$ (rất cao).
- Nếu \mathbf{P}_j nằm gần tâm $(0,0)$: Giá trị $x_{i,j}$ phân bố thấp hoặc trung bình, tạo ra sự cân bằng lực kéo.
- \mathbf{P}_j nằm giữa \mathbf{A}_i và \mathbf{A}_k : Chiều i và k có giá trị cao, các chiều khác có giá trị thấp.

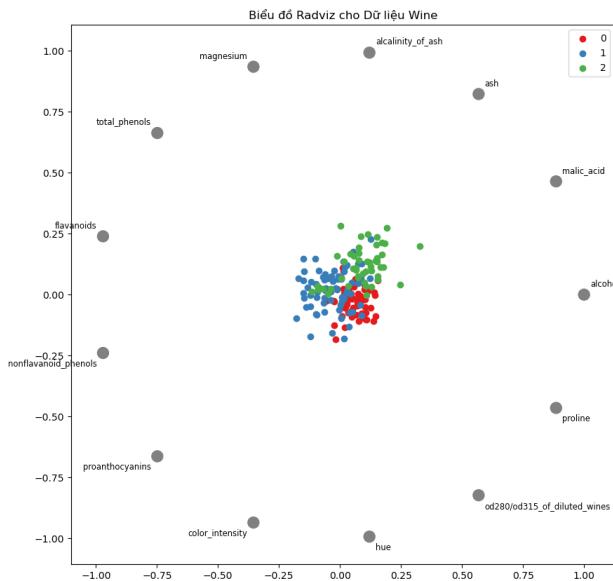
Các kỹ thuật tương tác mạnh nhất dành riêng cho RadViz

- Kéo thả Dimensional Anchors (mỏ neo). Kéo bất kỳ mỏ neo nào trên vòng tròn → thay đổi góc và thứ tự chiều ngay lập tức
- Axis Reordering bằng drag-and-drop. Kéo mỏ neo ra rồi thả vào vị trí mới → thay đổi thứ tự chiều (rất quan trọng vì RadViz phụ thuộc mạnh thứ tự).
- Inversion (đảo trực). Click đúp mỏ neo → đảo giá trị ($1 - x$).
- Animated Transition. Điểm di chuyển mượt khi thay đổi thứ tự mỏ neo.

```
[3]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
from sklearn.preprocessing import MinMaxScaler
from pandas.plotting import radviz

wine = load_wine()
wine_df = pd.DataFrame(data=wine.data, columns=wine.
    ↪feature_names)
wine_df['Class'] = wine.target
feature_cols = wine.feature_names
scaler = MinMaxScaler()
wine_scaled_data = scaler.fit_transform(wine_df[feature_cols])
wine_scaled_df = pd.DataFrame(wine_scaled_data,
    ↪columns=feature_cols)
wine_scaled_df['Class'] = wine_df['Class']

plt.figure(figsize=(10, 10))
radviz(wine_scaled_df, 'Class', color=plt.cm.get_cmap('Set1').
    ↪colors[:3])
plt.title('Biểu đồ Radviz cho Dữ liệu Wine')
plt.show()
```



Hình 5.16: Hệ toạ độ xuyên tâm (phương pháp Radviz).

5.5 Bài tập

Chủ đề và câu hỏi nghiên cứu

Bài 5.1. Quá trình cấu trúc văn bản ảnh hưởng như thế nào đến kết quả phân tích dữ liệu khám phá văn bản và tài liệu?

Bài 5.2. Những thách thức trong việc tiến hành phân tích mô tả cho đồ thị và mạng lưới là gì? và chúng có thể được giải quyết như thế nào?

Bài 5.3. Thảo luận về những cân nhắc về mặt đạo đức liên quan đến việc thể hiện các loại dữ liệu khác nhau, bao gồm các sai lệch tiềm ẩn trong hình ảnh hóa và phân tích mô tả, cũng như tác động đến việc ra quyết định.

Bài 5.4. Thảo luận về những hàm ý triết học của việc dự đoán các sự kiện trong tương lai dựa trên lịch sử dữ liệu chuỗi thời gian, đặt câu hỏi về khái niệm nhân quả và những vấn đề đạo đức liên quan đến việc sử dụng thông tin quá khứ để dự đoán tương lai.

Bài tập tính toán

Bài 5.5. Đối với Bộ dữ liệu Đánh giá Phim IMDb, hãy khám phá các kỹ thuật như đám mây từ hoặc phân tích tần suất để hình dung các từ hoặc cụm từ phổ biến nhất được sử dụng trong các bài đánh giá phim tích cực so với tiêu cực. Tạo biểu đồ phân tán hoặc

biểu đồ thanh để khám phá mối quan hệ giữa xếp hạng phim và cảm nhận của bài đánh giá hoặc độ dài từ.

Bài 5.6. Đối với Bộ dữ liệu Câu lạc bộ Karate của Zachary, hãy sử dụng biểu đồ mạng để trực quan hóa bối cảnh xã hội Kết nối giữa các thành viên câu lạc bộ karate, làm nổi bật các cộng đồng khác nhau dựa trên sự chia tách câu lạc bộ cuối cùng. Áp dụng thuật toán để xác định các cộng đồng trong mạng và phân tích các đặc điểm của chúng. Khám phá các thuộc tính như quy mô cộng đồng hoặc số lượng cạnh trong và giữa các cộng đồng. Điều tra các lý do tiềm ẩn cho sự chia tách câu lạc bộ dựa trên cấu trúc mạng.

Bài 5.7. Dữ liệu Mạng lưới vòng tròn xã hội Facebook1 bao gồm các vòng tròn từ Facebook col- được thực hiện thông qua khảo sát. Dữ liệu được ẩn danh và bao gồm các đặc điểm nút (hồ sơ), vòng tròn và mạng lưới bản ngã. Đối với tập dữ liệu này, hãy thực hiện:

- a. Phân tích cấu trúc mạng, bao gồm số lượng nút và cạnh, phân phối bậc và hệ số phân cụm.
- b. Xác định các cộng đồng và các nút có ảnh hưởng bằng thuật toán phát hiện cộng đồng Nhịp điệu và thước đo độ tập trung.

Bài 5.8. Sử dụng bộ dữ liệu Iris và viết các phương pháp tương tác biểu diễn dữ liệu nhiều chiều đối với các phương pháp: Hệ toạ độ song song, hệ toạ độ hình sao, và Radviz.

Chương 6

KẾ CHUYỆN BẰNG DỮ LIỆU VÀ THIẾT KẾ BẢNG ĐIỀU KHIỂN

Chương này bao gồm các chủ đề nhằm nâng cao khả năng của một nhà phân tích dữ liệu, đặc biệt là trong lĩnh vực phân tích mô tả, kể chuyện bằng dữ liệu (data storytelling), và thiết kế bảng điều khiển (dashboard design).

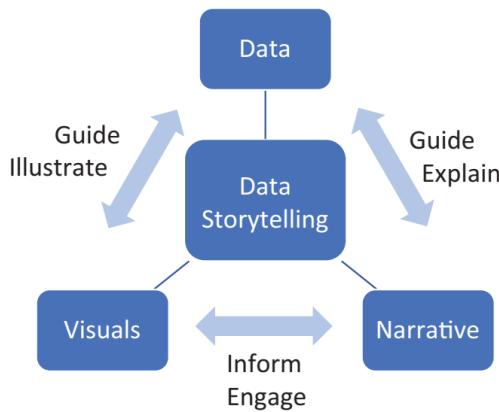
6.1 Kể chuyện bằng Dữ liệu (Data Storytelling)

Kể chuyện bằng dữ liệu là một phương pháp mạnh mẽ để truyền đạt dữ liệu và phân tích. Phương pháp này làm cho toàn bộ quá trình trở nên dễ hiểu, giúp thu hút và duy trì sự chú ý của khán giả, làm nổi bật các hiểu biết sâu sắc (*insights*), thúc đẩy hành động, tạo điều kiện ghi nhớ và thậm chí gợi lên cảm xúc.

Trong khi phân tích mô tả và trực quan hóa dữ liệu cung cấp các khía cạnh phân tích và thị giác của việc khám phá dữ liệu, thì kể chuyện bằng dữ liệu tiến thêm một bước bằng cách tích hợp các yếu tố này vào một **câu chuyện hấp dẫn** (*compelling narrative*). Câu chuyện này tạo được sự đồng cảm với khán giả, thúc đẩy sự hiểu biết, ra quyết định và hành động tốt hơn.

Kể chuyện bằng dữ liệu liên quan đến sự kết hợp hiệu quả của **dữ liệu, câu chuyện (narrative) và hình ảnh trực quan (visuals)**.

- **Dữ liệu (Data)** là cốt lõi của bất kỳ câu chuyện dữ liệu nào, là thông tin thô mà bạn muốn truyền tải. Mọi câu chuyện dữ liệu nên bắt nguồn từ dữ liệu; tức là các hiểu biết sâu sắc và thông điệp được truyền đạt phải được **trích xuất từ, hỗ trợ bởi và hướng dẫn bởi dữ liệu**.



Hình 6.1: Mối quan hệ giữa dữ liệu, tường thuật và hình ảnh để tạo nên một câu chuyện xoay quanh dữ liệu có sẵn và bối cảnh của nó.

- **Câu chuyện (Narrative)** là cốt truyện hoặc chuỗi sự kiện mang lại ý nghĩa cho dữ liệu. Nó giúp thông báo và thu hút khán giả thông qua dữ liệu, làm nổi bật các điểm chính và cung cấp **ngữ cảnh**. Ngữ cảnh rất quan trọng vì nó cung cấp nền tảng giúp diễn giải dữ liệu.
- **Hình ảnh trực quan (Visuals)** là các biểu đồ, đồ thị, bảng và các yếu tố hình ảnh khác được sử dụng để đại diện và minh họa dữ liệu cùng các mối quan hệ của nó. Hình ảnh trực quan giúp làm cho dữ liệu phức tạp trở nên dễ hiểu hơn và làm nổi bật các xu hướng hoặc mô hình.

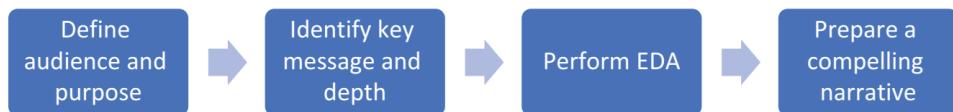
Sự tương tác giữa dữ liệu, câu chuyện và hình ảnh trực quan được tóm tắt như sau: Dữ liệu và Câu chuyện dùng để **Hướng dẫn (Guide)** và **Giải thích (Explain)**; Dữ liệu và Hình ảnh trực quan dùng để **Hướng dẫn (Guide)** và **Minh họa (Illustrate)**; Hình ảnh trực quan và Câu chuyện dùng để **Thông báo (Inform)** và **Thu hút (Engage)**.

Kể chuyện bằng dữ liệu hiệu quả có thể làm cho dữ liệu phức tạp trở nên **dễ hiểu và dễ ghi nhớ** hơn. Trong phương pháp được trình bày trong tài liệu này, kể chuyện bằng dữ liệu sẽ đóng vai trò là **hướng dẫn và khuôn khổ cho việc thiết kế bảng điều khiển (dashboard)**. Điều này có nghĩa là quy trình kể chuyện sẽ cho phép xây dựng một bảng điều khiển mà khi được khám phá, sẽ dẫn người dùng đến những hiểu biết sâu sắc, kết luận, quyết định và hành động của riêng họ.

6.1.1 Các Bước Thiết kế Câu chuyện Dữ liệu

Các bước sau đây được sử dụng để xây dựng một câu chuyện hiệu quả xoay quanh dữ liệu:

- a. Xác định đối tượng và mục đích;
- b. Xác định thông điệp chính và độ sâu thông tin;
- c. Thực hiện phân tích dữ liệu khám phá (EDA);
- d. Chuẩn bị một câu chuyện hấp dẫn (*narrative*).



Hình 6.2: Quá trình thiết kế về kể chuyện dữ liệu.

Xác định Đối tượng và Mục đích (*Define the Audience and Purpose*)

Xác định đối tượng và mục đích là một bước **cơ bản** trong việc tạo ra một câu chuyện dữ liệu hiệu quả.

- **Đối tượng:** Cần xác định nhóm cụ thể hoặc các cá nhân bạn muốn giao tiếp. Điều này bao gồm xem xét các yếu tố như nhân khẩu học, sở thích và **mức độ kiến thức** của họ. Việc hiểu các khía cạnh này giúp điều chỉnh câu chuyện để **tạo sự đồng cảm** với khán giả, làm cho nó hấp dẫn và liên quan hơn đến nhu cầu của họ.
- **Mục đích:** Cần làm rõ các mục tiêu mà bạn muốn đạt được thông qua câu chuyện dữ liệu của mình. Mục đích có thể là để **thông báo, thuyết phục, giáo dục hoặc truyền cảm hứng hành động**.

Tóm lại, bước ban đầu này thiết lập nền tảng để xây dựng một câu chuyện dữ liệu không chỉ truyền đạt hiệu quả mà còn kết nối với đối tượng mục tiêu ở mức độ ý nghĩa.

Xác định Thông điệp Chính và Độ sâu (*Identify the Key Message and Depth*)

Đây là bước tiếp theo trong việc giao tiếp hiệu quả thông qua dữ liệu.

- **Thông điệp Chính (Key Message):** Đóng vai trò là **chủ đề trung tâm** hoặc hiểu biết sâu sắc mà bạn muốn truyền đạt. Đó là điều cốt lõi mà bạn muốn khán giả nắm bắt được từ dữ liệu được trình bày. Thông điệp này phải **rõ ràng, súc tích và phù hợp** với mục đích giao tiếp của bạn.
- **Độ sâu Thông tin (Depth of Information):** Để tăng cường tác động của câu chuyện dữ liệu, việc hiểu **độ sâu thông tin** cần thiết để hỗ trợ thông điệp chính là rất quan trọng. Điều này bao gồm việc đánh giá mức độ chi tiết, ngữ cảnh và dữ liệu hỗ trợ cần thiết để cung cấp sự hiểu biết toàn diện về thông điệp chính.

Bằng cách xác định thông điệp chính và hiểu độ sâu của thông tin, bạn thiết lập nền tảng để tạo ra một câu chuyện dữ liệu tập trung và hấp dẫn, truyền đạt hiệu quả các hiểu biết sâu sắc dự định.

Thực hiện Phân tích Dữ liệu Khám phá (*Perform the Exploratory Data Analysis - EDA*)

Đây là bước thứ ba trong quy trình kể chuyện bằng dữ liệu. EDA trong bối cảnh này nhằm trả lời các câu hỏi như:

- Bạn muốn trả lời những câu hỏi nào bằng dữ liệu?
- Những loại mối quan hệ nào tồn tại trong dữ liệu?
- Đâu là các kỹ thuật tốt nhất để hiển thị dữ liệu?

Chuẩn bị một Câu chuyện Hấp dẫn (*Prepare a Compelling Narrative*)

Việc chuẩn bị một câu chuyện hấp dẫn và **ngữ cảnh hóa dữ liệu** là yếu tố thiết yếu trong quá trình soạn thảo một bài thuyết trình hoặc báo cáo có sức thuyết phục và tác động.

- **Cấu trúc Câu chuyện:** Cần xây dựng một cấu trúc câu chuyện dẫn dắt khán giả đi qua các hiểu biết sâu sắc và thông tin.
- **Thành phần:** Cấu trúc này thường bao gồm **phần giới thiệu** (để thiết lập bối cảnh), một **cốt truyện được xác định rõ ràng** (để trình bày dữ liệu một cách mạch lạc), và một **kết luận** (để gắn kết các hiểu biết sâu sắc chính lại với nhau).

- **Dòng chảy:** Câu chuyện phải tuân theo **một dòng chảy hợp lý**, thu hút sự chú ý của khán giả và duy trì sự tham gia của họ.

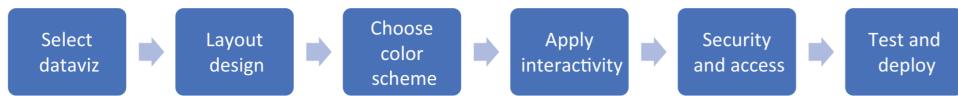
6.2 Thiết kế Bảng điều khiển (Dashboard Design)

Bảng điều khiển (Dashboard) là một **biểu diễn trực quan** của thông tin quan trọng cần thiết để hoàn thành một hoặc nhiều mục tiêu. Nó được trình bày dưới dạng hợp nhất và có tổ chức trên một màn hình duy nhất để dễ dàng và nhanh chóng theo dõi. Bảng điều khiển được thiết kế để cung cấp một cái nhìn tổng quan khách quan về tình trạng hoặc hiệu suất của một hệ thống, quy trình hoặc doanh nghiệp.

Trong khi kể chuyện bằng dữ liệu là phần cốt truyện (*narrative*), thì **thiết kế bảng điều khiển** là quy trình mà theo đó kết quả của phân tích dữ liệu khám phá (EDA) được xây dựng và trình bày một cách trực quan và **thẩm mỹ**.

- Bảng điều khiển đóng vai trò là **giao diện trực quan** chuyển đổi quy trình phân tích dữ liệu thành các màn hình hiển thị trực quan và linh hoạt.
- Chúng hoạt động như một trung tâm điều khiển, cung cấp ảnh chụp nhanh về các chỉ số chính (*key metrics*), xu hướng và các chỉ số hiệu suất.
- Mục tiêu là cho phép người dùng ra quyết định sáng suốt và hành động bằng cách nắm bắt nhanh chóng các thông tin và xu hướng thiết yếu.
- Bảng điều khiển có thể là **tĩnh** hoặc **tương tác** và thường có thể được tùy chỉnh để đáp ứng nhu cầu cụ thể của người dùng.
- Các tính năng phổ biến bao gồm khả năng điều hướng qua thông tin chi tiết, tìm hiểu sâu, **đặt bộ lọc** và nhận cập nhật thời gian thực.
- Bảng điều khiển còn tạo điều kiện cho sự **hợp tác** bằng cách cung cấp một nền tảng tập trung.

Quá trình thiết kế bảng điều khiển theo một luồng công việc, bao gồm các bước: Chọn Trực quan hóa Phù hợp, Thiết kế Bố cục, Chọn Phối màu, Áp dụng Tính tương tác, Bảo mật và Kiểm soát Truy cập, và Kiểm tra và Lặp lại.



Hình 6.3: Quá trình thiết kế bảng điều khiển.

6.2.1 Chọn các Trực quan hóa Thích hợp (Selecting Appropriate Visualizations)

Việc lựa chọn các trực quan hóa được đưa vào bảng điều khiển phụ thuộc vào dữ liệu có sẵn và **câu chuyện cần kể**.

- Các phương pháp trực quan hóa đã được khám phá trong sách được tổ chức dựa trên khả năng hiển thị **phân phối, mối liên hệ, số lượng, tỷ lệ, sự tiến hóa và lưu lượng, và dữ liệu địa lý không gian**.
- Khi lựa chọn, cần xem xét **bản chất của dữ liệu** (danh mục, số, chuỗi thời gian, văn bản, mạng lưới), vì các loại dữ liệu khác nhau yêu cầu trực quan hóa khác nhau.
- Cần xem xét mức độ quen thuộc của khán giả với các loại trực quan hóa nhất định [10], [11].
- Trực quan hóa phải **đơn giản và dễ hiểu**, tránh lộn xộn và chọn định dạng giảm thiểu sự nhầm lẫn.
- Một quy tắc kinh nghiệm đơn giản là xem xét **ba câu hỏi hướng dẫn EDA**: những câu hỏi nào cần trả lời bằng dữ liệu, những mối quan hệ nào cần quan sát, và các phương pháp tốt nhất để hiển thị các biến và mối quan hệ của chúng.

6.2.2 Thiết kế Bố cục Bảng điều khiển (Designing the Dashboard Layout)

Thiết kế bố cục là bước **quan trọng** để đảm bảo câu chuyện được trình bày một cách rõ ràng, hiệu quả và thân thiện với người dùng.

- Một bảng điều khiển hấp dẫn về mặt thị giác không chỉ dễ nhìn mà còn giúp người dùng **hiểu và diễn giải dữ liệu dễ dàng hơn**.
- Tính thẩm mỹ có thể bị ảnh hưởng bởi việc sử dụng **phối màu nhất quán và dễ chịu**, phông chữ thích hợp và khoảng cách.

- Bộ cục nên được tổ chức theo **dòng chảy thông tin hợp lý** (ví dụ: từ trái sang phải, từ trên xuống dưới).
- Thông tin quan trọng nên được đặt ở **các vị trí nổi bật**.
- Dữ liệu tương tự hoặc có liên quan nên được **nhóm lại**.
- Kích thước của mỗi biểu đồ hoặc widget nên **phản ánh tầm quan trọng** hoặc sự phức tạp của dữ liệu nó đại diện, hoặc tính thẩm mỹ chung của bảng điều khiển.
- Cần sử dụng không gian một cách hiệu quả để tách các phần khác nhau và **tránh sự lộn xộn** (*cluttered*) hoặc các yếu tố trang trí không cần thiết.

6.2.3 Chọn Phối màu (Choosing a Color Scheme)

Chọn phối màu là một khía cạnh quan trọng để thiết kế bảng điều khiển hấp dẫn về mặt thị giác và dễ hiểu, nhằm **tăng cường khả năng đọc dữ liệu và kể chuyện**. Việc lựa chọn và triển khai phối màu một cách chu đáo, dựa trên các nguyên tắc, sẽ tạo ra một bảng điều khiển vừa có tính thẩm mỹ vừa truyền đạt thông tin hiệu quả.

6.2.4 Áp dụng Tính tương tác (Applying Interactivity)

Mục đích của việc áp dụng tính tương tác là **nâng cao trải nghiệm người dùng** và cung cấp khả năng kiểm soát đối với quá trình khám phá dữ liệu.

- **Bộ lọc (Filters):** Cho phép người dùng **giới hạn dữ liệu** được hiển thị. Các loại bộ lọc có thể là danh mục (ví dụ: theo loại sản phẩm), số (ví dụ: theo số lượng bán hàng), hoặc thời gian (ví dụ: theo ngày). Bộ lọc thường được trình bày dưới dạng menu thả xuồng, hộp kiểm hoặc thanh trượt.
- **Tooltip:** Bảng điều khiển có thể bao gồm các chú giải công cụ (*tooltips*) xuất hiện khi di chuột qua các điểm dữ liệu, cung cấp thông tin chi tiết bổ sung (ví dụ: giá trị hoặc nhãn chính xác) mà không làm lộn xộn hình ảnh trực quan chính.
- Tính tương tác thúc đẩy trải nghiệm **thu hút và năng động** (*dynamic*), khuyến khích người dùng tham gia tích cực vào quá trình phân tích.

6.2.5 Bảo mật và Kiểm soát Truy cập (Security and Access Control)

Bảo mật và kiểm soát truy cập là các thành phần **quan trọng** của bất kỳ hệ thống thông tin nào, nhằm bảo vệ dữ liệu nhạy cảm và đảm bảo rằng chỉ những người được ủy quyền mới có quyền truy cập.

- Việc triển khai các biện pháp bảo mật thích hợp bao gồm sự kết hợp của các biện pháp **kỹ thuật, thủ tục và tổ chức** để bảo vệ khỏi truy cập trái phép và vi phạm dữ liệu.
- Mặc dù các biện pháp kỹ thuật chi tiết (như bảo vệ dữ liệu, cơ chế xác thực) **nằm ngoài phạm vi** của tài liệu này, nhưng vai trò quan trọng của chúng trong việc duy trì an ninh cho bảng điều khiển là được công nhận.

6.2.6 Kiểm tra và Lặp lại (Test and Iterate)

Kiểm tra và lặp lại là các quy trình **quan trọng** liên quan đến việc đánh giá, liên tục tinh chỉnh và cải thiện thiết kế, chức năng và khả năng sử dụng của bảng điều khiển.

- Thông qua kiểm tra, nhà thiết kế thu thập **phản hồi có giá trị từ người dùng** để xác định và giải quyết các vấn đề.
- Cách tiếp cận lặp lại này đảm bảo bảng điều khiển cuối cùng **phù hợp chặt chẽ với nhu cầu và mục tiêu của người dùng**.
- Việc kiểm tra và lặp lại phải bao gồm:
 - Kiểm tra từng hình ảnh trực quan về chức năng và độ chính xác.
 - Kiểm tra bảng điều khiển với người dùng tiềm năng và điều chỉnh dựa trên phản hồi.
 - Đảm bảo khả năng **phản hồi** (*responsiveness*) trên các thiết bị và kích thước màn hình khác nhau.
 - Tối ưu hóa hiệu suất để **thời gian tải nhanh**.
 - Cung cấp tài liệu và đào tạo cho người dùng.
 - Cập nhật bảng điều khiển thường xuyên để phản ánh những thay đổi trong dữ liệu và nhu cầu người dùng.

6.3 Nghiên cứu Tình huống 1: GAPMINDER DATASET

Bộ dữ liệu Gapminder cung cấp một cái nhìn tổng quan về **xu hướng phát triển toàn cầu**. Bộ dữ liệu này tập trung vào các chỉ số xã hội, kinh tế và sức khỏe trên khắp các quốc gia theo thời gian [1]. Xuyên suốt cuốn sách này, bộ dữ liệu Gapminder đã được sử dụng để trình bày và thảo luận về nhiều khái niệm và hình ảnh trực quan khác nhau, bao gồm biểu đồ bong bóng (*bubble charts*), biểu đồ hình cây (*tree maps*), biểu đồ đường (*line charts*) và bản đồ phân vùng theo màu (*choropleth maps*).

Mục tiêu của nghiên cứu tình huống ban đầu này là cung cấp một **phân tích đơn giản về sự phát triển toàn cầu** dựa trên bộ dữ liệu Gapminder. Bảng điều khiển (*dashboard*) sẽ cung cấp các hiểu biết sâu sắc về mối quan hệ giữa **Chỉ số Phát triển Con người (HDI)**, **tuổi thọ (life expectancy)**, và **mức tiêu thụ CO₂** ở mỗi châu lục. Người dùng sẽ có khả năng chọn châu lục và quốc gia muốn điều tra, và bảng điều khiển cũng sẽ cung cấp các hiểu biết sâu sắc về các quốc gia có **GDP cao nhất** ở mỗi châu lục.

6.3.1 Kể chuyện bằng Dữ liệu (Data Storytelling)

Bước đầu tiên trong quy trình thiết kế bảng điều khiển là **kể chuyện bằng dữ liệu**. (Nội dung chi tiết cho bước này không được cung cấp thêm trong nguồn.)

6.3.2 Thiết kế Bảng điều khiển (Dashboard Design)

Quá trình thiết kế bảng điều khiển bao gồm các bước sau:

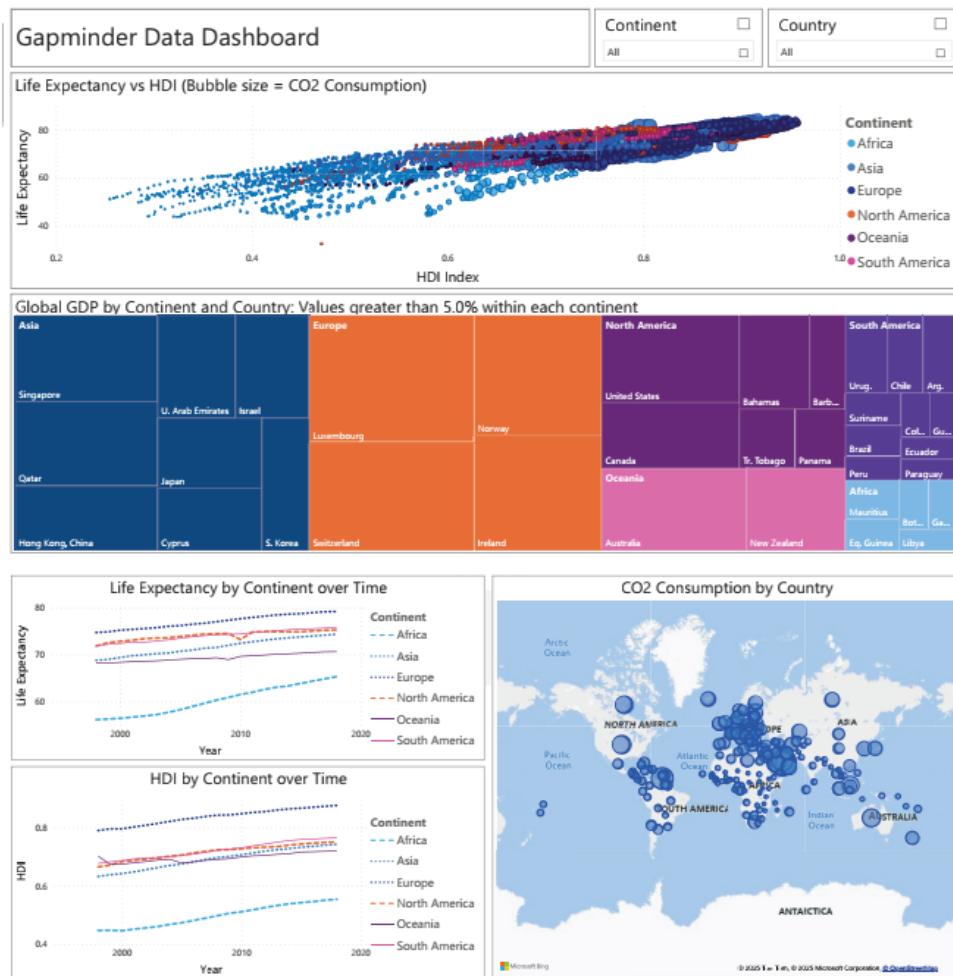
a. **Chọn Trực quan hóa Phù hợp:** Dựa trên mục tiêu đã xác định, các phương pháp trực quan hóa sau đã được chọn để xây dựng bảng điều khiển cuối cùng:

- Một **biểu đồ bong bóng** (*bubble chart*) cho mối quan hệ giữa tuổi thọ và HDI, trong đó các châu lục được phân biệt bằng màu sắc khác nhau, và kích thước của bong bóng biểu thị mức tiêu thụ CO₂.
- Một **biểu đồ hình cây** (*treetemap*) thể hiện GDP theo quốc gia và châu lục.
- Hai **biểu đồ đường** (*line charts*), một cho tuổi thọ và một cho HDI, cả hai đều thể hiện sự tiến hóa theo thời gian.
- Ngoài ra, bảng điều khiển còn có **các hộp kết hợp** (*combo boxes*) để người dùng chọn châu lục và quốc gia cần phân tích.

- b. **Thiết kế Bô cục Bảng điều khiển:** Bô cục được sử dụng để xây dựng bảng điều khiển được mô tả trong sơ đồ khung (*wireframe*).
- c. **Chọn Phối màu:** Trong nghiên cứu tình huống này, bảng điều khiển được phát triển bằng Python, sử dụng thư viện **Dash**. Thư viện Dash được thiết kế đặc biệt để xây dựng các ứng dụng web phân tích và giao diện trực quan hóa dữ liệu tương tác. Để so sánh với một bảng điều khiển có áp dụng phối màu cụ thể (sẽ được thực hiện trong nghiên cứu tình huống thứ hai), nghiên cứu này **không áp dụng phối màu nào** và sẽ sử dụng các phối màu tiêu chuẩn của từng biểu đồ đã chọn.
- d. **Các Bước Khác:** (Các bước 4, 5 và 6 tương tự như nghiên cứu tình huống khác)
 - Bảo mật và kiểm soát truy cập (*Security and Access Control*).
 - Kiểm tra và lặp lại (*Test and Iterate*).



Hình 6.4: Sơ đồ thiết kế bảng điều khiển.



Hình 6.5: Bảng điều khiển cho dữ liệu Gapminder.

6.4 Nghiên cứu tình huống 2: SUPERSTORE SALES DATASET

Bộ dữ liệu Bán hàng Siêu thị (*Superstore Sales Dataset*) chứa dữ liệu bán lẻ của một siêu thị toàn cầu trong **bốn năm**. Bộ dữ liệu này có 18 thuộc tính, bao gồm ID hàng, ID đơn hàng, ngày đặt hàng, ngày và phương thức vận chuyển, phân khúc, quốc gia, thành phố, v.v..

Mục tiêu của Nghiên cứu Tình huống

Mục tiêu của nghiên cứu tình huống này là cung cấp một **phân tích toàn diện về hiệu suất bán hàng** cho siêu thị toàn cầu trong một khoảng thời gian.

Bảng điều khiển (*dashboard*) được thiết kế nhằm mục đích khám phá các hiểu biết sâu sắc về **hành vi khách hàng**, **xu hướng bán hàng**, và **phân bố địa lý của các đơn hàng**. Thông qua các trực quan hóa tương tác và **bộ lọc tùy chỉnh** (*customizable filters*), bảng điều khiển sẽ cho phép người dùng khám phá các chỉ số chính như:

- **Tổng giá trị bán hàng** (*total sales value*).
- **Số lượng bán hàng** (*number of sales*).
- Hiệu suất theo **danh mục** (*category-wise performance*).
- Phân tích theo **phân khúc** (*segment-wise analysis*).

Mục tiêu cuối cùng là **trao quyền cho người ra quyết định** với các hiểu biết sâu sắc có thể hành động được (*actionable insights*) nhằm thúc đẩy các quyết định kinh doanh chiến lược, tối ưu hóa hoạt động và nâng cao sự hài lòng của khách hàng.

6.4.1 Kể chuyện bằng Dữ liệu (Data Storytelling)

Quá trình kể chuyện bằng dữ liệu cho bảng điều khiển này bao gồm các bước sau:

a. Xác định Đối tượng và Mục đích:

- **Đối tượng:** Các cá nhân hoặc nhóm tham gia vào quá trình ra quyết định của doanh nghiệp bán lẻ, ví dụ như nhà phân tích kinh doanh, quản lý bán hàng, hoặc giám đốc điều hành. Họ được kỳ vọng có sự hiểu biết tốt về các chỉ số bán hàng và hoạt động kinh doanh bán lẻ.
- **Mục đích: Thông báo và giáo dục** (*inform and educate*), cung cấp cái nhìn toàn diện về hiệu suất bán hàng theo thời gian, trên các danh mục khác nhau và ở các vị trí địa lý khác nhau.

b. Xác định Thông điệp Chính và Độ sâu:

- **Các câu hỏi cần trả lời:** Làm thế nào các đơn hàng được phân bổ trên các tiểu bang khác nhau, tổng giá trị bán hàng là bao nhiêu, và giá trị bán hàng cùng số lượng bán hàng thay đổi như thế nào theo năm, phương thức vận chuyển và phân khúc.

- **Mối quan hệ trong dữ liệu:** Có thể có mối quan hệ giữa thời gian (tháng/năm) và tổng giá trị hoặc số lượng bán hàng, chỉ ra xu hướng theo mùa. Cũng có thể có mối quan hệ giữa danh mục/danh mục phụ của sản phẩm và tổng giá trị bán hàng, gợi ý rằng một số sản phẩm phổ biến hoặc có lợi nhuận cao hơn.

c. **Thiết kế Trực quan hóa và KPI:**

- Một **Chỉ số Hiệu suất Chính (KPI)** được sử dụng để trình bày tổng giá trị bán hàng.
- **Bộ lọc (Slicers)** được sử dụng để lọc dữ liệu theo năm, phương thức vận chuyển và phân khúc, cho phép người dùng khám phá dữ liệu sâu hơn.

d. **Chuẩn bị một Câu chuyện Hấp dẫn:**

- **Cốt truyện:** Bắt đầu với lời chào mừng đến "Superstore Sales Dashboard" [8], nơi khám phá dữ liệu bán lẻ để tìm kiếm insights và xu hướng. Bảng điều khiển này cung cấp tổng quan toàn diện về hiệu suất bán hàng của siêu thị toàn cầu trong nhiều năm.
- **Dòng chảy:** Câu chuyện nhấn mạnh việc khám phá động lực bán hàng, hành vi khách hàng, và xác định cơ hội phát triển và tối ưu hóa thông qua các biểu đồ và bản đồ tương tác.

6.4.2 Thiết kế Bảng điều khiển (Dashboard Design)

Bảng điều khiển cuối cùng được xây dựng bằng **Power BI** của Microsoft. Bảng điều khiển này minh họa các tính năng tương tác và hiển thị sau:

- **Bố cục:** Sơ đồ khung (*wireframe*) được sử dụng để mô tả bố cục của bảng điều khiển.
- **Tương tác:** Bảng điều khiển hiển thị đầy đủ các tùy chọn của bộ lọc trượt (*slicers*) được chọn, cũng như tác động của các bộ lọc đã áp dụng lên các biểu đồ được hiển thị.
- **Tính năng Chi tiết (Drill-down) và Chú giải Công cụ (Tooltip):** Bảng điều khiển thể hiện tính năng **drill-down** khi một danh mục phụ (ví dụ: bìa hồ sơ từ vật tư văn phòng) được chọn, và đồng thời làm nổi bật **chú giải công cụ** bật lên khi con trỏ được đặt trên thanh của biểu đồ thanh.

- Bảo mật và Kiểm tra:** Các bước về **bảo mật và kiểm soát truy cập** (*security and access control*) và **kiểm tra và lặp lại** (*test and iterate*) được đề cập đến trong nghiên cứu tình huống trước cũng được áp dụng.



Hình 6.6: Sơ đồ thiết kế bảng điều khiển.



Hình 6.7: Bảng điều khiển cho dữ liệu Superstore Sales.

6.5 Bài tập

Câu hỏi và chủ đề thảo luận

Bài 6.1. Làm thế nào việc kể chuyện bằng dữ liệu có thể bị thao túng để đánh lừa khán giả? cần cân nhắc khi kể một câu chuyện có dữ liệu?

Bài 6.2. Liệu KPI có thực sự là thước đo hiệu suất khách quan, hay chúng có thể bị ảnh hưởng bởi các yếu tố chủ quan? Làm thế nào chúng ta có thể đảm bảo tính công bằng và chính xác trong việc lựa chọn và diễn giải KPI?

Bài 6.3. Liệu bảng thông tin có dân chủ hóa dữ liệu, giúp những người không phải chuyên gia có thể truy cập được hay chúng có đơn giản hóa quá mức các tập dữ liệu phức tạp, có khả năng dẫn đến hiểu sai không?

Bài 6.4. Làm thế nào để chúng ta đạt được sự cân bằng giữa tính thẩm mỹ và tiện ích chức năng trong bảng điều khiển? Thiết kế bảng điều khiển? Một bảng điều khiển được thiết kế tốt có thể vừa đẹp vừa hữu ích, hay những mục tiêu này xung đột với nhau?

Bài 6.5. Thảo luận về mức độ trách nhiệm của các chuyên gia dữ liệu trong quá trình ra quyết định. Thảo luận về những tác động đạo đức của việc trình bày dữ liệu và thông tin chi tiết cho quá trình ra quyết định. những người sáng tạo và cách những chuyên gia này có thể đóng góp vào việc đưa ra quyết định có đạo đức và sáng suốt.

Đồ án cuối kỳ Đồ án cuối kỳ được thiết kế để áp dụng toàn diện kiến thức và kỹ năng bạn đã học được trong suốt khóa học về EDA. Trong đồ án này, bạn sẽ cơ hội lựa chọn tập dữ liệu theo ý muốn và tạo bảng thông tin tương tác, giàu thông tin để trực quan hóa và phân tích dữ liệu. Mục tiêu là áp dụng các nguyên tắc mô tả

Phân tích tích cực, trực quan hóa dữ liệu và thiết kế bảng điều khiển để truyền đạt hiệu quả những hiểu biết và mô hình ẩn giấu trong dữ liệu. Hơn nữa, dự án này sẽ thử thách bạn tư duy theo hướng kinh doanh và/hoặc có tác động, không chỉ nhằm mục đích tạo ra một bảng điều khiển sâu sắc mà còn phát triển ý tưởng cho một giải pháp dựa trên dữ liệu dựa trên tập dữ liệu và những hiểu biết bạn khám phá.

Yêu cầu của dự án:

1. Phát triển khái niệm kinh doanh hoặc tác động

- Bắt đầu bằng cách xác định một vấn đề hoặc cơ hội thực tế có thể được giải quyết hoặc tận dụng bằng cách sử dụng dữ liệu.
- Tạo ra một khái niệm kinh doanh hoặc tác động rõ ràng để xuất giá trị, đối tượng mục tiêu và tác động tiềm ẩn (xã hội, môi trường, tài chính, v.v.) của giải pháp dựa trên dữ liệu của bạn.

- Trình bày rõ ràng cách dự án của bạn có thể dẫn đến một cơ hội kinh doanh khả thi hoặc sự va chạm.

2. Lựa chọn tập dữ liệu

Chọn một tập dữ liệu mà bạn quan tâm và phù hợp với mục tiêu học tập hoặc nghề nghiệp của bạn. Tập dữ liệu này có thể liên quan đến bất kỳ lĩnh vực nào, chẳng hạn như tài chính, y tế, khoa học xã hội, trò chơi, môi trường, v.v.

3. Dọn dẹp và chuẩn bị dữ liệu

- Làm sạch và xử lý trước tập dữ liệu để xử lý các giá trị bị thiếu, giá trị ngoại lệ và bất kỳ dữ liệu nào vẫn đề về chất lượng.
- Ghi lại các bước dọn dẹp dữ liệu và tạo từ điển dữ liệu để giải thích các biến của tập dữ liệu và ý nghĩa của chúng

4. Phân tích mô tả

Tiến hành phân tích dữ liệu mô tả kỹ lưỡng để có được sự hiểu biết và tính Phân tích dữ liệu. Bao gồm phân phối tần suất, thước đo xu hướng trung tâm, thước đo độ biến thiên, thước đo hình thức và thước đo liên kết.

5. Trực quan hóa dữ liệu

- Áp dụng các nguyên tắc trực quan hóa dữ liệu để tạo ra các biểu diễn trực quan của dữ liệu.
- Thiết kế và triển khai nhiều loại hình trực quan hóa khác nhau để trình bày các khía cạnh khác nhau của tập dữ liệu, chẳng hạn như biểu đồ histogram, biểu đồ phân tán, biểu đồ thanh, biểu đồ đường, v.v.
- Chọn các kỹ thuật trực quan hóa phù hợp cho các loại dữ liệu và nghiên cứu khác nhau câu hỏi

6. Kể chuyện dữ liệu

- Xác định đối tượng và mục đích.
- Xác định thông điệp chính và chiều sâu.
- Thực hiện phân tích thăm dò (Bước 4 ở trên).
- Chuẩn bị một câu chuyện hấp dẫn.

7. Thiết kế và nguyên mẫu bảng điều khiển

- Chọn hình ảnh trực quan phù hợp
- Thiết kế Bố cục Bảng điều khiển
- Chọn một bảng màu
- Áp dụng tính tương tác
- Kiểm soát an ninh và truy cập
- Kiểm tra và lắp lại

8. Trình bày

Chuẩn bị một bài thuyết trình để giới thiệu dự án của bạn. Bài thuyết trình nên bao gồm phần trình diễn trực tiếp bảng điều khiển và hướng dẫn toàn bộ quy trình phân tích dữ liệu.

Tài liệu tham khảo

- [1] Tamara Munzner, Visualization Analysis and Design, AK Peters Visualization Series, CRC Press, 2014.
- [2] Edward R. Tufte, The Visual Display of Quantitative Information, 2nd Ed, Graphics Press, 2001.
- [3] Jonathan Schwabish, Better Data Visualizations: A Guide for Scholars, Researchers, and Wonks, Columbia University Press, 2021.
- [4] Cole Nussbaumer Knaflic, Storytelling with Data: A Data Visualization Guide for Business Professionals, Wiley, 2015.
- [5] Steve Wexler, Jeffrey Shaffer, Andy Cotgreave, The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios, Wiley, 2017.
- [6] Stephen Few, Show Me the Numbers: Designing Tables and Graphs to Enlighten, Analytics Press, 2012.
- [7] Colin Ware, Information Visualization: Perception for Design (Interactive Technologies), 4th Ed, Morgan Kaufmann, 2020.
- [8] Leland Wilkinson, D. Wills, D. Rope, A. Norton (Contributor), R. Dubbs, The Grammar of Graphics (Statistics and Computing), 2nd Ed, Springer, 2015 .
- [9] Stephen Few , Information Dashboard Design: The Effective Visual Communication of Data, O'Reilly Media, 2006.
- [10] Leandro Nunes de Castro, Exploratory data analysis: descriptive analysis, visualization and dashboard design, CRC Press, 2025.
- [11] Joshua N. Milligan, Blair Hutchinson, Mark Tossell, Roberto Andreoli , Learning Tableau 2022: Create effective data visualizations, build interactive visual analytics, and improve your data storytelling capabilities, 5th Edition, Packt Publishing, 2022.