

ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN

THƯ KÝ



BÁO CÁO

Học phần: Phân tích dữ liệu

BÀI TẬP:

Phân tích và khám phá dữ liệu

Giảng viên hướng dẫn: ThS. Đỗ Nhu Tài

Thành viên nhóm:

Nguyễn Vũ Hào – MSSV: 3122410098

Trần Duy Khương – MSSV: 3122410192

Võ Thị Thương – MSSV: 3122410408

Tp.Hồ Chí Minh, 2025

Họ và Tên	Mã số sinh viên	Phân công công việc
Nguyễn Vũ Hào	3122410098	<ul style="list-style-type: none"> - Làm mục 1.1 - Viết doc Chương 1 - Làm slide Phần 1
Trần Duy Khương	3122410192	<ul style="list-style-type: none"> - Làm mục 1.3 - Viết doc Chương 3 - Làm slide Phần 3
Võ Thị Thương	3122410408	<ul style="list-style-type: none"> - Làm mục 1.2 - Viết doc Chương 2 - Làm slide Phần 2

MỤC LỤC

DANH MỤC HÌNH ẢNH	4
DANH MỤC CÁC CHỮ VIẾT TẮT	7
LỜI CẢM ƠN	8
LỜI MỞ ĐẦU	9
I. Lý thuyết thống kê mô tả	10
1. Định nghĩa	10
2. Thước đo thống kê mô tả	12
3. Phân bố của một tập dữ liệu	13
4. Ý nghĩa của việc phân bố	14
5. Khác biệt giữa các thước đo.....	14
6. Xử lý giá trị thiếu	15
7. Cách diễn giải một biểu đồ.....	16
8. Tập dữ liệu có giá trị ngoại lai	16
CHƯƠNG 2: TRỰC QUAN HÓA DỮ LIỆU	31
I. Lý thuyết trực quan hóa dữ liệu	31
1. Vai trò của trực quan hóa dữ liệu.....	31
2. Các loại biểu đồ phổ biến	32
3. Cách lựa chọn biểu đồ phù hợp.....	34
4. Thư viện trực quan hóa trong Python	35
5. Nguyên tắc thiết kế biểu đồ	36
6. Cách tạo biểu đồ.....	37
7. Cách xuất biểu đồ	38
II. Làm quen với trực quan hóa dữ liệu	39
1. Bài toán 1	39
2. Bài toán 2	44
3. Bài toán 3	51
4. Bài toán 4	54
CHƯƠNG 3: PHÂN TÍCH ĐƠN BIỀN VÀ HAI BIỀN	66

I.	Lý thuyết Phân tích Đơn biến và Hai biến	66
1.	Phân tích đơn biến	66
2.	Phân tích hai biến	66
3.	Thước đo thống kê phân tích đơn biến	66
4.	Xác định mối quan hệ trong phân tích hai biến	68
5.	Sự khác biệt giữa tương quan (correlation) và hiệp biến (covariance)	72
6.	Khi nào nên sử dụng biểu đồ trực quan hóa trong phân tích đơn biến so với phân tích hai biến?	73
7.	Mẫu code ví dụ về tạo biểu đồ scatter plot	73
8.	Cách trực quan hóa mối quan hệ giữa một biến số và một biến phân loại bằng biểu đồ boxplot	75
II.	Làm quen với các hàm và thư viện hỗ trợ phân tích dữ liệu đơn biến	77
1.	Bài toán 1	78
2.	Bài toán 2:	83
3.	Bài toán 3:	86
III.	Giới thiệu các tính năng và cách sử dụng thư viện SweetViz	87
1.	Giới thiệu thư viện SweetViz	87
2.	Những tính năng chính của SweetViz	87
3.	Cài đặt và sử dụng	88
	TÀI LIỆU THAM KHẢO	100

DANH MỤC HÌNH ẢNH

Hình 1. 1 Kiểm tra thông tin về dữ liệu COVID	18
Hình 1. 2 Tính toán các giá trị trong dữ liệu của new_case	18
Hình 1. 3 Tính toán các giá trị trong dữ liệu của total_case.....	18
Hình 1. 4 Thông tin về dữ liệu marketing_data.....	20
Hình 1. 5 Tiền xử lý dữ liệu.....	21
Hình 1. 6 Thông tin về dữ liệu phân loại chất lượng rượu	23
Hình 1. 7 Tính toán các phép đo cơ bản của tập dữ liệu	23
Hình 1. 8 Thống kê mô tả cho toàn bộ dữ liệu	24
Hình 1. 9 Dữ liệu thu được sau quá trình lọc	24
Hình 1. 10 Kiểm tra thông tin trùng lặp của dữ liệu.....	25
Hình 1. 11 Xử lý giá trị trùng lặp của dữ liệu.....	25
Hình 1. 12 Kiểm tra giá trị thiếu của dữ liệu	26
Hình 1. 13 Kiểm tra thông tin dữ liệu đầu vào của bệnh tiểu đường	27
Hình 1. 14 Các thống kê cơ bản của dữ liệu.....	28
Hình 1. 15 Kiểm tra dữ liệu bị thiếu trong tập dữ liệu	28
Hình 1. 16 Kiểm tra dữ liệu có chứa giá trị 0	29
Hình 1. 17 Xử lý tập dữ liệu có chứa giá trị thiếu	29
Hình 1. 18 Dữ liệu sau khi áp dụng MinMaxScaler	30
Hình 2. 1 Ví dụ về biểu đồ histogram.....	32
Hình 2. 2 Ví dụ về biểu đồ Scatter plot	33
Hình 2. 3 Ví dụ về biểu đồ Box plot.....	34
Hình 2. 4 Biểu đồ Bar chart	34
Hình 2. 5 Biểu đồ phân bố dữ liệu.....	37
Hình 2. 6 Biểu đồ Bar Chart	38
Hình 2. 7 Dữ liệu được sắp xếp theo giá cả	40
Hình 2. 8 Biểu đồ thể hiện top 10 khu vực hàng đầu	41
Hình 2. 9 Biểu đồ top 10 khu vực có giá nhà cao nhất.....	41
Hình 2. 10 Biểu đồ biểu thị top 10 khu vực theo giá nhà(bên trái) và giá trên mỗi mét vuông(bên phải).....	42
Hình 2. 11 Biểu đồ trực quan hóa Price theo Zip	43
Hình 2. 12 Biểu đồ top 10 giá nhà cao nhất	43
Hình 2. 13 Biểu đồ sử dụng seaborn để so sánh giá nhà và giá trên mỗi mét vuông.....	44
Hình 2. 14 Phân bố tất cả các biến trong dataset rượu vang	47
Hình 2. 15 Biểu đồ phân bố boxplot theo chất lượng rượu	47
Hình 2. 16 Biểu đồ phân bố boxplot theo chất lượng rượu(1)	48
Hình 2. 17 Biểu đồ phân bố boxplot theo chất lượng rượu(2)	48
Hình 2. 18 Biểu đồ phân bố boxplot theo chất lượng rượu(3)	49
Hình 2. 19 Biểu đồ ma trận tương quan giữa các biến trong tập dữ liệu	50
Hình 2. 20 Biểu đồ phân bố các biến trong dataset.....	52
Hình 2. 21 Biểu đồ boxplot biểu thị các giá trị trong dataset.....	52

Hình 2. 22 Ma trận tương quan giữa các biến	53
Hình 2. 23 Biểu đồ thể hiện sự khác biệt phân bố giữa 2 nhóm	54
Hình 2. 24 Thông tin dữ liệu ban đầu của việc mua sắm tại siêu thị	55
Hình 2. 25 Thống kê mô tả dữ liệu	55
Hình 2. 26 Biểu đồ quan sát sự phân bố Quantity.....	56
Hình 2. 27 Biểu đồ quan sát sự phân bố UnitPrice	56
Hình 2. 28 Biểu đồ thể hiện số lượng giao dịch cao nhất.....	57
Hình 2. 29 Biểu đồ số lượng sản phẩm bán ra theo năm.....	58
Hình 2. 30 Kiểm tra dữ liệu thiếu.....	58
Hình 2. 31 Mapping từ tockCode sang Description	59
Hình 2. 32 Bộ dữ liệu đơn bán dựa trên Quantity và InvoiceNo	59
Hình 2. 33 Bộ dữ liệu đơn bán hợp lệ dựa trên Quantity và InvoiceNo	59
Hình 2. 34 Thống kê mô tả dữ liệu của cột UnitPrice	60
Hình 2. 35 Thống kê mô tả dữ liệu của cột Quantity	60
Hình 2. 36 Biểu đồ thể hiện sự phân bố của biến Quantity	61
Hình 2. 37 Biểu đồ thể hiện sự phân bố của biến UnitPrice	61
Hình 2. 38 Top 10 quốc gia theo tổng doanh thu	62
Hình 2. 39 Top 10 sản phẩm bán chạy nhất.....	62
Hình 2. 40 Biểu đồ doanh thu theo tháng.....	63
Hình 2. 41 Biểu đồ phân phối số lượng giao dịch trong ngày.....	63
Hình 2. 42 Biểu đồ ma trận tương quan giữa các biến định lượng	64
Hình 2. 43 Top 10 quốc gia có tỷ lệ hoàn hàng cao nhất	65
Hình 2. 44 Top 10 sản phẩm bị hoàn hàng nhiều nhất	65
Hình 3. 1 Ví dụ về biểu đồ Scatter Plot.....	69
Hình 3. 2 Ví dụ biểu đồ BoxPlots.....	70
Hình 3. 3 Ví dụ về Violin Plot	70
Hình 3. 4 Bar Chart with error bars	71
Hình 3. 5 Code để tạo biểu đồ	74
Hình 3. 6 Kết quả khi chạy code tạo biểu đồ.....	74
Hình 3. 7 Code khai báo tập dữ liệu	75
Hình 3. 8 Code tạo biểu đồ	75
Hình 3. 9 Biểu đồ hộp.....	76
Hình 3. 10 Code để tạo biểu đồ Violin	77
Hình 3. 11 Biểu đồ Violin.....	77
Hình 3. 12 Import thư viện và tạo biểu đồ	78
Hình 3. 13 Biểu đồ Histogram của cột culmen_length_mm	78
Hình 3. 14 Code tạo biểu đồ	79
Hình 3. 15 Biểu đồ Bar chart.....	79
Hình 3. 16 Code tạo biểu đồ	80
Hình 3. 17 Biểu đồ Pie-Chart	80
Hình 3. 18 Import thư viện và tạo biểu đồ	81
Hình 3. 19 Biểu đồ Box	81
Hình 3. 20 Code tạo biểu đồ Violin	81

Hình 3. 21 Biểu đồ Violin.....	82
Hình 3. 22 Bảng thống kê mô tả.....	82
Hình 3. 23 Chuẩn bị dữ liệu và tạo scatterplot	84
Hình 3. 24 Kết quả biểu đồ scatterplot	84
Hình 3. 25 Thống kê crosstab/two-way.....	85
Hình 3. 26 Code tạo bảng Pivot table.....	85
Hình 3. 27 Code tạo biểu đồ Pairplot	85
Hình 3. 28 Code tạo báo cáo html	86
Hình 3. 29 Giao diện báo cáo html.....	87
Hình 3. 30 Code để mở giao diện D-Table.....	87
Hình 3. 31 Giao diện cài đặt thư viện.....	88
Hình 3. 32 Giao diện tổng quan của báo cáo.....	91
Hình 3. 33 Vị trí nhấn để xem Associations	91
Hình 3. 34 Giao diện của Associations.....	92
Hình 3. 35 Phần giao diện chính của biểu đồ dựa trên biến.....	93
Hình 3. 36 Phần giao diện chính của biểu đồ dựa trên biến	93
Hình 3. 37 Phần giao diện biểu đồ Histogram và các loại thống kê khác	95
Hình 3. 38 Phần giao diện biểu đồ Histogram và các loại thống kê khác của biến phân loại	97
Hình 3. 39 Phần giao diện biểu đồ Histogram và các loại thống kê khác khi tạo báo cáo kèm biến mục tiêu.....	98
Hình 3. 40 Phần giao diện biểu đồ Histogram và các loại thống kê khác khi tạo báo cáo với hợp biến	99

DANH MỤC CÁC CHỮ VIẾT TẮT

STT	Chữ viết tắt	Giải thích đầy đủ
1	ANOVA	Phân tích phương sai (<i>Analysis of Variance</i>)
2	BMI	Chỉ số khối cơ thể (<i>Body Mass Index</i>)
3	EDA	Phân tích và khám phá dữ liệu (<i>Exploratory Data Analysis</i>)
4	IQR	Khoảng tứ phân vị (<i>Interquartile Range - Q3 – Q1</i>)
5	KURT	Độ nhọn (Kurtosis)
6	Matplotlib	Thư viện trực quan hóa dữ liệu cơ bản trong Python
7	MinMaxScaler	Công cụ chuẩn hóa (<i>đưa toàn bộ giá trị về khoảng</i>)
8	Plotly	Thư viện trực quan hóa dữ liệu tương tác (Python)
9	Q-Q plot	Biểu đồ phân vị-phân vị (<i>Quantile-Quantile plot</i>)
10	Seaborn	Thư viện trực quan hóa thống kê (xây dựng dựa trên Matplotlib)
11	SKEW	Độ lệch trái/phải (Skewness)
12	SO ₂	Dioxide Lưu huỳnh (<i>Sulfur Dioxide</i>)
13	STD	Độ lệch chuẩn (<i>Standard Deviation</i>)
14	SweetViz	Thư viện hỗ trợ EDA tự động (Python)
15	VAR	Phương sai (Variance)

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến ThS. Đỗ Như Tài, Giảng viên hướng dẫn học phần Phân tích dữ liệu tại Trường Đại học Sài Gòn. Thầy đã tận tình hướng dẫn, truyền đạt những kiến thức quý báu và hỗ trợ chúng em trong suốt quá trình học tập và thực hiện báo cáo này.

Những kiến thức và kinh nghiệm mà thầy chia sẻ, đặc biệt là về Thống kê Mô tả, Trục quan hóa Dữ liệu, và các phương pháp Phân tích Đơn biến/Hai biến, không chỉ giúp nhóm sinh viên chúng em hiểu rõ hơn về các phương pháp phân tích dữ liệu mà còn rèn luyện khả năng xử lý và trình bày dữ liệu một cách khoa học.

Chúng em cũng xin chân thành cảm ơn Quý Thầy/Cô trong Khoa và Trường Đại học Sài Gòn đã tạo điều kiện thuận lợi trong suốt quá trình học tập. Đồng thời, chúng em xin gửi lời cảm ơn đến các bạn cùng lớp đã chia sẻ, trao đổi, và hỗ trợ trong quá trình thực hiện đề tài.

Mặc dù nhóm đã nỗ lực cố gắng hoàn thiện báo cáo một cách tốt nhất, song do kiến thức và kinh nghiệm còn hạn chế, bài báo cáo chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được những góp ý quý báu từ Thầy để nhóm có thể hoàn thiện hơn trong tương lai.

Chúng em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

Báo cáo này được thực hiện nhằm hoàn thành Bài tập lớn của học phần *Phân tích dữ liệu* tại Trường Đại học Sài Gòn, với trọng tâm là quá trình *Phân tích và khám phá dữ liệu* (Exploratory Data Analysis – EDA). Mục tiêu của báo cáo không chỉ dừng lại ở việc áp dụng các kỹ thuật thống kê để mô tả dữ liệu mà còn hướng đến việc giúp người nghiên cứu hiểu sâu hơn về bản chất của dữ liệu — cụ thể là dữ liệu “trông như thế nào”, có cấu trúc ra sao, xu hướng và mức độ phân tán như thế nào, cũng như các mối quan hệ tiềm ẩn giữa các biến. Quá trình này đóng vai trò quan trọng trong việc hình thành nền tảng cho các bước phân tích nâng cao như mô hình hóa hoặc dự đoán. Việc kết hợp giữa các phương pháp thống kê mô tả và trực quan hóa dữ liệu giúp báo cáo không chỉ có tính khoa học mà còn trực quan, dễ hiểu, tạo điều kiện để phát hiện các quy luật hoặc bất thường tiềm ẩn trong tập dữ liệu.

Nội dung của báo cáo được triển khai theo hướng đi từ lý thuyết cơ bản đến thực hành ứng dụng trên nhiều bộ dữ liệu khác nhau, bao gồm dữ liệu về COVID-19, chiến dịch marketing, chất lượng rượu đỏ và bệnh tiểu đường. Cụ thể, phần đầu tiên trình bày các kiến thức nền tảng về *thống kê mô tả*, bao gồm các thước đo xu hướng trung tâm (trung bình, trung vị, mode), độ phân tán (phương sai, độ lệch chuẩn, phạm vi) và cách xác định dạng phân bố dữ liệu, cùng các phương pháp xử lý giá trị thiếu và giá trị ngoại lai. Tiếp theo là phần *trực quan hóa dữ liệu*, trong đó nhấn mạnh vai trò của biểu đồ và đồ thị như Histogram, Boxplot, Scatter Plot và Bar Chart trong việc trình bày dữ liệu phức tạp dưới dạng trực quan dễ hiểu. Phần này cũng giới thiệu việc sử dụng các thư viện Python như *Matplotlib* và *Seaborn* để hỗ trợ quá trình minh họa và phân tích. Cuối cùng, phần *phân tích đơn biến và hai biến* tập trung vào việc khai thác mối quan hệ giữa các biến, xác định xu hướng, mô hình và mức độ tương quan hoặc hiệp biến giữa chúng.

Thông qua việc thực hiện các bài toán và nhiệm vụ trong báo cáo, nhóm mong muốn áp dụng hiệu quả các kiến thức lý thuyết đã học vào thực tế, rèn luyện tư duy logic, khả năng xử lý và trình bày dữ liệu một cách khoa học.

CHƯƠNG 1: THỐNG KÊ MÔ TẢ

I. Lý thuyết thống kê mô tả

1. Định nghĩa

Thống kê mô tả là một nhánh của thống kê, bao gồm các phương pháp và kỹ thuật được sử dụng để tóm tắt, sắp xếp, mô tả và trình bày các đặc điểm chính của một tập dữ liệu đã được thu thập (có thể là một mẫu hoặc toàn bộ tổng thể)[1, 2, 3, 4].

Mục tiêu cốt lõi là giúp người nghiên cứu hiểu rõ dữ liệu "trong như thế nào", bao gồm cấu trúc, xu hướng và sự phân tán của nó, mà không đưa ra bất kỳ kết luận hay suy luận nào về một tổng thể rộng lớn hơn.

Thống kê suy luận là tập hợp các phương pháp sử dụng lý thuyết xác suất để phân tích dữ liệu mẫu và đưa ra các dự đoán, kết luận hoặc tổng quát hóa có ý nghĩa về các đặc điểm (tham số) của toàn bộ tổng thể mà mẫu đó được rút ra.

Nó giải quyết câu hỏi: "Dữ liệu mẫu này có ý nghĩa gì đối với toàn bộ nhóm lớn hơn?"

- *Bảng so sánh:*

- Khác biệt cốt lõi

Loại thống kê	Thống kê Mô Tả	Thống kê suy luận
Mục tiêu chính	Tóm tắt và Trình bày dữ liệu.	Tổng quát hóa và Dự đoán dữ liệu.
Phạm vi	Dữ liệu đã có sẵn (Mẫu → Mẫu).	Dữ liệu mẫu được dùng cho Tổng thể (Mẫu → Tổng thể).

- Các tiêu chí khác:

Đặc điểm	Thống Kê Mô Tả	Thống Kê Suy Luận
Mục tiêu chính	Mô tả, tóm tắt, sắp xếp các đặc điểm của dữ liệu đã thu thập.	Dự đoán, tổng quát hóa, và rút ra kết luận về Tổng thể dựa trên dữ liệu từ mẫu.
Phạm vi	Chỉ giới hạn trong tập dữ liệu hiện có (Mẫu hoặc Tổng thể).	Mở rộng ra ngoài mẫu, hướng tới toàn bộ Tổng thể lớn hơn.
Các công cụ/Chỉ số	1. Xu hướng tập trung: Mean, Median, Mode. 2.	1. Ước lượng: Khoảng tin cậy (Confidence

	Độ phân tán: Standard Deviation, Variance, Range. 3. Trình bày: Bảng tần số, Biểu đồ (Histogram, Pie chart).	Intervals). 2. Kiểm định: T-test, ANOVA, Chi-square, P-value. 3. Phân tích quan hệ: Hồi quy (Regression).
Tính Chất Kết quả	Phản ánh sự thật của dữ liệu. Kết quả là các con số chắc chắn (ví dụ: điểm trung bình là 7.5).	Mang tính xác suất và rủi ro (biên độ lỗi). Kết quả là các dự đoán đi kèm với mức độ tin cậy.
Câu hỏi Trả lời	Dữ liệu này là gì? Đặc điểm của nó như thế nào?	Dữ liệu này có ý nghĩa gì đối với Tổng thể? Liệu có sự khác biệt có ý nghĩa nào không?
Đầu ra	Các Thống kê (Statistics) và đồ thị.	Các Tham số Parameters) của Tổng thể, Giá trị P, Khoảng tin cậy.
Ví dụ	"45% sinh viên trong mẫu khảo sát này là nữ."	"Chúng tôi ước tính rằng tỷ lệ sinh viên nữ trong toàn trường là $45\% \pm 5\%$, với độ tin cậy 95%."

- Giải thích điểm đặc biệt

- Sự Khác Biệt về Hành động: Mô tả vs. Dự đoán. Thống kê Mô tả: Thực hiện hành động "Ké lại" hay "Tóm tắt". Nó giống như việc bạn đọc một cuốn nhật ký và ghi lại tóm tắt các sự kiện đã xảy ra. Ví dụ: Bạn đếm được 500 người trong phòng và tính ra tuổi trung bình của 500 người đó là 35. Đây là một sự thật không cần suy đoán. Thống kê Suy luận: Thực hiện hành động "Suy đoán" hay "Mở rộng". Nó giống như việc bạn đọc một vài trang nhật ký (mẫu) và cố gắng dự đoán toàn bộ cuộc đời của người viết (tổng thể). Ví dụ: Bạn khảo sát tuổi của 100 người (mẫu) trong phòng và dùng con số đó để ước tính tuổi trung bình của tất cả 5.000 người (tổng thể) trong tòa nhà. Kết luận này luôn đi kèm với rủi ro và xác suất (ví dụ: Khoảng tin cậy 95%).
- Sự Khác Biệt về Kết quả: Chắc chắn với Xác suất. Thống kê mô tả đưa ra các kết quả là sự thật về dữ liệu được quan sát. Thống kê suy luận đưa ra các kết luận có thể đúng về Tổng thể và luôn gắn liền với xác suất (p-value) và biên độ lỗi (margin of error). Nói tóm lại, nếu

bạn chỉ muốn biết điều gì đang xảy ra trong tập dữ liệu bạn có, bạn dùng Mô tả. Nếu bạn muốn biết điều gì xảy ra ngoài tập dữ liệu bạn có, bạn dùng Suy luận. [5]

- Vai trò và Ứng dụng của Thống kê Mô tả

Thống kê mô tả (Descriptive Statistics) đóng vai trò quan trọng trong phân tích dữ liệu, giúp nhà nghiên cứu tóm tắt và làm rõ các đặc điểm cơ bản của một tập dữ liệu trước khi tiến hành các phân tích sâu hơn. Các thước đo chính như trung bình, trung vị, phương sai và độ lệch chuẩn được sử dụng để phản ánh xu hướng trung tâm và mức độ phân tán của dữ liệu, qua đó mang lại cái nhìn khái quát về cấu trúc thông tin thu thập được ([1]; [2]).

2. Thước đo thống kê mô tả

Trung bình (Mean) là thước đo phổ biến nhất, thể hiện giá trị trung tâm thông qua trung bình cộng. Nó phù hợp khi dữ liệu phân phối đối xứng, không có giá trị ngoại lai. Tuy nhiên, trong thực tế, dữ liệu thường tồn tại các giá trị cực đoan làm méo mó kết quả. Trong những trường hợp này, trung vị (Median) là lựa chọn tốt hơn vì phản ánh giá trị ở giữa của tập dữ liệu và ít bị ảnh hưởng bởi ngoại lai, đặc biệt hữu ích với dữ liệu lệch hoặc dữ liệu dạng thứ bậc như thang đo mức độ hài lòng [6].

Bên cạnh đó, phương sai (Variance) và độ lệch chuẩn (Standard Deviation) là các chỉ số quan trọng để đo lường độ biến thiên. Phương sai cho biết mức độ dữ liệu phân tán quanh trung bình, nhưng đơn vị tính bị bình phương nên khó trực quan. Ngược lại, độ lệch chuẩn là căn bậc hai của phương sai, giữ nguyên đơn vị ban đầu của dữ liệu nên thường được dùng để diễn giải mức độ dao động trong thực tế [7].

Tóm lại, thống kê mô tả không chỉ giúp đơn giản hóa dữ liệu phức tạp mà còn hỗ trợ nhà nghiên cứu xác định xu hướng và mức độ phân tán, từ đó tạo nền tảng vững chắc cho các phân tích thống kê suy luận tiếp theo.

Mặc dù Trung bình là thước đo phổ biến, Trung vị được coi là thước đo đáng tin cậy hơn để đại diện cho "trung tâm" của dữ liệu trong các trường hợp sau:

- Khi dữ liệu bị lệch (Skewed): Đặc biệt là trong các phân phối lệch phải (ví dụ: thu nhập, giá nhà) [1]. Trung bình sẽ bị kéo về phía đuôi dài (giá trị cao), khiến nó cao hơn Trung vị và không còn phản ánh đúng mức giá/thu nhập "điển hình" của đa số.

- Khi có Giá trị Ngoại lai (Outliers): Một hoặc vài giá trị cực đoan sẽ làm Trung bình bị sai lệch lớn, trong khi Trung vị hầu như không bị ảnh hưởng và giữ được tính đại diện [2].
- Khi dữ liệu là Thứ bậc (Ordinal Data): Dữ liệu chỉ mang tính chất thứ tự (ví dụ: thang đo mức độ hài lòng 1-5). Trung vị phản ánh tốt nhất giá trị trung tâm vì tính Trung bình cho loại dữ liệu này có thể kém ý nghĩa [6].

Tóm lại, trung bình phù hợp cho dữ liệu đối xứng; Trung vị là lựa chọn mặc định tốt nhất khi phân phối dữ liệu không đối xứng, có ngoại lai, hoặc mang tính thứ bậc.

3. Phân bố của một tập dữ liệu

Để xác định phân bố của một tập dữ liệu, có thể dùng hai cách chính: trực quan và định lượng.

Phương pháp trực quan thường bao gồm việc vẽ các loại đồ thị khác nhau. Histogram (biểu đồ tần suất) là công cụ phổ biến nhất, cho phép ta quan sát trực tiếp hình dạng phân bố dữ liệu: đối xứng, lệch trái, hay lệch phải (Moore, McCabe & Craig, 2017) [7]. Boxplot (biểu đồ hộp) lại hữu ích trong việc tóm tắt dữ liệu, xác định trung vị, phạm vi tứ phân vị và đặc biệt là phát hiện các giá trị ngoại lai (Seltman, 2018) [4]. Ngoài ra, Q-Q plot (Quantile-Quantile plot) được dùng để kiểm tra tính phù hợp với một phân bố lý thuyết, chẳng hạn như phân bố chuẩn. Nếu các điểm dữ liệu nằm gần đường chéo 45° , ta có cơ sở để cho rằng dữ liệu phân bố chuẩn (UCLA IDRE, 2023) [6].

Phương pháp định lượng dựa vào các số liệu thống kê mô tả. Việc so sánh trung bình và trung vị giúp nhận diện dạng phân bố: nếu trung bình gần bằng trung vị, dữ liệu có xu hướng đối xứng; nếu trung bình lớn hơn trung vị, dữ liệu thường lệch phải; còn nếu nhỏ hơn, dữ liệu có thể lệch trái (Laerd Statistics, 2018) [8].Thêm vào đó, hệ số độ lệch (skewness) cung cấp một thước đo chính xác hơn: giá trị dương biểu thị lệch phải, âm biểu thị lệch trái, còn gần bằng 0 gọi ý tính đối xứng (Seltman, 2018) [9]. Bên cạnh đó, độ nhọn (kurtosis) đo mức độ “nhọn” hay độ dày của đuôi phân bố; phân bố chuẩn có kurtosis xấp xỉ 3 (Moore, McCabe & Craig, 2017) [7].

Một số dạng phân bố cơ bản thường gặp trong phân tích dữ liệu gồm:

- Phân bố đối xứng (Symmetrical Distribution). Đây là dạng phân bố mà đồ thị có dạng đối xứng qua trung tâm. Trường hợp điển hình nhất là phân bố chuẩn (Normal/Gaussian Distribution), có hình chuông, đối xứng hoàn hảo và chỉ có một đỉnh. Trong phân bố này, trung bình

= trung vị = một (Moore, McCabe & Craig, 2017) [1]; nhiều hiện tượng tự nhiên như chiều cao, huyết áp hay điểm số thi thường tuân theo dạng này (UCLA IDRE, 2023) [6].

- Phân bố lệch (Skewed Distributions). Nếu dữ liệu không đối xứng, ta gặp hai trường hợp chính. Phân bố lệch phải (positively skewed) có đuôi kéo dài về phía các giá trị lớn, khiến trung bình lớn hơn trung vị; ví dụ thường gặp là thu nhập, giá nhà hoặc thời gian chờ đợi (Laerd Statistics, 2018) [8]. Ngược lại, phân bố lệch trái (negatively skewed) có đuôi kéo dài về phía giá trị nhỏ, với trung bình nhỏ hơn trung vị; ví dụ là điểm thi đại học (nhiều học sinh đạt điểm cao) hoặc tuổi thọ (Seltman, 2018) [9].
- Phân bố đa đỉnh (Multimodal Distribution). Khi dữ liệu có nhiều hơn một đỉnh, ta gọi là phân bố đa đỉnh. Dạng này thường phản ánh sự pha trộn của nhiều nhóm khác nhau trong cùng một tập dữ liệu. Ví dụ: chiều cao của một mẫu gồm cả nam và nữ thường tạo ra hai đỉnh riêng biệt (Moore, McCabe & Craig, 2017) [7].

4. Ý nghĩa của việc phân bố

Độ lệch chuẩn (Standard Deviation) và Phạm vi (Range) là hai thước đo quan trọng trong thống kê mô tả để đánh giá sự phân tán của dữ liệu.

Trước hết, phạm vi (range) là khoảng cách giữa giá trị lớn nhất và nhỏ nhất trong tập dữ liệu. Đây là thước đo đơn giản nhất về mức độ trải rộng, cho ta biết dữ liệu “trải” rộng bao nhiêu (Moore, McCabe & Craig, 2017) [7]. Tuy nhiên, phạm vi chỉ dựa trên hai điểm cực trị nên dễ bị ảnh hưởng bởi các giá trị ngoại lai, vì vậy nó thường được dùng để có một cái nhìn sơ bộ, nhanh chóng về độ phân tán (Seltman, 2018) [9].

Trong khi đó, độ lệch chuẩn (standard deviation) phản ánh mức độ mà các điểm dữ liệu dao động quanh giá trị trung bình. Nếu độ lệch chuẩn nhỏ, dữ liệu tập trung gần trung bình; nếu lớn, dữ liệu phân tán rộng hơn (UCLA IDRE, 2023) [6]. Khác với phạm vi, độ lệch chuẩn tận dụng toàn bộ dữ liệu trong mẫu, nên là một thước đo ổn định và đáng tin cậy hơn để so sánh sự biến thiên giữa các tập dữ liệu (Laerd Statistics, 2018) [8].

Nói cách khác, phạm vi cho thấy sự chênh lệch tối đa, còn độ lệch chuẩn cho thấy “mức độ dao động trung bình” của dữ liệu. Vì vậy, cả hai thước đo thường được dùng bổ sung cho nhau: phạm vi để tóm tắt nhanh sự trải rộng, và độ lệch chuẩn để phân tích sâu hơn về mức độ phân tán tổng thể.

5. Khác biệt giữa các thước đo

Trong biểu đồ hộp (boxplot), ba thước đo Q1, Q2, Q3 chính là các tứ phân vị (quartiles), có vai trò chia dữ liệu thành bốn phần bằng nhau và cho thấy sự phân tán của dữ liệu.

- Q1 (Quartile 1 hay Lower Quartile): Là giá trị phân vị thứ nhất, tức điểm nằm tại vị trí 25% dữ liệu nhỏ nhất. Q1 thể hiện ranh giới giữa nhóm dữ liệu thấp và phần còn lại. Nó thường được dùng để xác định “đáy” của hộp trong boxplot (Moore, McCabe & Craig, 2017) [7].
- Q2 (Quartile 2 hay Median): Chính là trung vị (median) của tập dữ liệu, chia dữ liệu thành hai nửa bằng nhau. Trong boxplot, Q2 được vẽ dưới dạng đường ngang bên trong hộp, thể hiện giá trị trung tâm của dữ liệu (Laerd Statistics, 2018) [8].
- Q3 (Quartile 3 hay Upper Quartile): Là giá trị phân vị thứ ba, tức điểm nằm tại vị trí 75% dữ liệu nhỏ nhất (hoặc 25% dữ liệu lớn nhất). Q3 biểu thị ranh giới của nhóm dữ liệu cao. Nó tạo thành “đỉnh” của hộp trong boxplot (UCLA IDRE, 2023) [6].

Sự khác biệt chính giữa Q1, Q2 và Q3 nằm ở vị trí trong phân phối dữ liệu: Q1 phản ánh ngưỡng thấp, Q2 phản ánh điểm giữa, còn Q3 phản ánh ngưỡng cao. Khoảng cách giữa Q3 và Q1 gọi là khoảng tứ phân vị (Interquartile Range – IQR = Q3 – Q1), được dùng để đo mức độ phân tán chính của dữ liệu và phát hiện giá trị ngoại lai (Seltman, 2018) [9].

6. Xử lý giá trị thiếu

Xử lý giá trị thiếu (missing values) là bước quan trọng trong tiền xử lý dữ liệu, giúp đảm bảo tính chính xác của các chỉ số thống kê mô tả như trung bình, trung vị, phương sai, hay độ lệch chuẩn. Có một số cách tiếp cận phổ biến:

- Loại bỏ giá trị thiếu (Deletion). Nếu số lượng giá trị thiếu nhỏ và phân bố ngẫu nhiên, ta có thể loại bỏ các quan sát chứa missing values. Phương pháp này đơn giản nhưng có nguy cơ làm giảm kích thước mẫu và tính đại diện của dữ liệu (Moore, McCabe & Craig, 2017) [7].
- Thay thế bằng thống kê tóm tắt (Simple Imputation). Một cách phổ biến là thay thế giá trị thiếu bằng trung bình (mean), trung vị (median) hoặc mode (mode) của biến. Trung bình thường dùng với dữ liệu đối称, còn trung vị được ưu tiên khi dữ liệu có ngoại lai hoặc bị lệch (Laerd Statistics, 2018) [8].
- Sử dụng phương pháp nội suy hoặc ước lượng nâng cao. Trong các tập dữ liệu phức tạp, có thể dùng nội suy (interpolation), hồi quy (regression

imputation), hoặc kỹ thuật Multiple Imputation để dự đoán giá trị bị thiếu dựa trên mối quan hệ với các biến khác (Seltman, 2018) [9].

- Gán giá trị đặc biệt hoặc giữ nguyên. Trong một số trường hợp (ví dụ: khảo sát xã hội học), giá trị thiếu có thể mang ý nghĩa riêng. Khi đó, nhà nghiên cứu có thể gán nhãn “không trả lời” hoặc giữ nguyên để phân tích riêng nhóm này (UCLA IDRE, 2023) [6].

Nói chung, lựa chọn phương pháp xử lý phụ thuộc vào tỷ lệ dữ liệu thiếu, cơ chế thiếu (ngẫu nhiên hay có hệ thống), và mục tiêu phân tích. Nếu dữ liệu thiếu ít và ngẫu nhiên, loại bỏ có thể chấp nhận được; nhưng với dữ liệu quan trọng hoặc thiếu nhiều, cần dùng đến kỹ thuật bù đắp hợp lý để tránh sai lệch trong các chỉ số thống kê mô tả.

7. Cách diễn giải một biểu đồ

Khi phân tích dữ liệu, histogram là công cụ trực quan giúp nhận diện hình dạng phân bố. Nếu histogram có dạng chuông đối xứng, dữ liệu có xu hướng gần với phân bố chuẩn. Nếu biểu đồ có đuôi kéo dài về bên phải, phân bố bị lệch phải – thường gặp ở các biến như thu nhập hoặc giá nhà. Ngược lại, nếu đuôi kéo dài về bên trái, phân bố bị lệch trái – ví dụ như điểm thi khi phần lớn học sinh đạt điểm cao. Histogram cũng có thể cho thấy dữ liệu có một đỉnh hay nhiều đỉnh; trường hợp có nhiều đỉnh thường gọi ý rằng dữ liệu gồm nhiều nhóm khác biệt [7].

Trong khi đó, boxplot tập trung mô tả dữ liệu thông qua năm số thống kê: giá trị nhỏ nhất, Q1, Q2 (trung vị), Q3 và giá trị lớn nhất. Hộp trong boxplot biểu diễn khoảng từ phân vị (IQR), phản ánh sự phân tán của 50% dữ liệu trung tâm. Nếu hộp dài, dữ liệu biến động mạnh; nếu ngắn, dữ liệu ổn định hơn. Vị trí trung vị trong hộp cũng cung cấp thông tin về xu hướng phân bố: trung vị ở giữa thể hiện phân bố gần đối xứng, lệch xuống dưới gợi ý phân bố lệch phải, và lệch lên trên phản ánh phân bố lệch trái. Các “râu” cho biết phạm vi dữ liệu chính, còn những điểm ngoài râu được coi là ngoại lai cần được chú ý [6].

Như vậy, histogram chủ yếu cho thấy hình dạng phân bố, trong khi boxplot nhấn mạnh độ phân tán, xu hướng trung tâm và ngoại lai. Kết hợp cả hai sẽ giúp ta có cái nhìn toàn diện và chính xác hơn về tập dữ liệu.

8. Tập dữ liệu có giá trị ngoại lai

Khi gặp một tập dữ liệu có giá trị ngoại lai (outliers), bước đầu tiên là cần xác định chúng bằng các công cụ trực quan như boxplot hoặc bằng các quy tắc thống kê, chẳng hạn như quy tắc $1.5 \times IQR$ (giá trị nhỏ hơn $Q1 - 1.5 \times IQR$ hoặc lớn hơn $Q3 + 1.5 \times IQR$ được xem là ngoại lai) hoặc kiểm tra điểm z-score ($|z| >$

3). Sau khi phát hiện, việc xử lý phụ thuộc vào mục tiêu phân tích và bản chất của dữ liệu.

Trong một số trường hợp, ngoại lai có thể là lỗi nhập liệu hoặc lỗi đo lường, khi đó cần kiểm tra và loại bỏ hoặc sửa chữa giá trị đó để dữ liệu phản ánh chính xác hơn thực tế [7]. Nếu ngoại lai là các quan sát hợp lệ nhưng hiếm gặp (ví dụ: một người có thu nhập cực cao trong khảo sát thu nhập), chúng không nên bị loại bỏ ngay. Thay vào đó, có thể sử dụng các thước đo ít nhạy cảm với ngoại lai như trung vị thay cho trung bình, hoặc dùng IQR thay cho độ lệch chuẩn để mô tả sự phân tán [6]. Một số kỹ thuật khác bao gồm biến đổi dữ liệu (log, căn bậc hai) để giảm tác động của ngoại lai, hoặc phân tích song song hai trường hợp: giữ nguyên và loại bỏ ngoại lai để so sánh kết quả.

Tóm lại, xử lý ngoại lai đòi hỏi sự cân nhắc cẩn thận. Việc loại bỏ, giữ lại hay biến đổi cần dựa trên mục tiêu nghiên cứu, bản chất của dữ liệu, và tác động mà ngoại lai gây ra cho các chỉ số thống kê mô tả.

II. Làm quen với thống kê mô tả

1. Bài toán 1

Thực hiện các nhiệm vụ trong bài toán 1 để làm quen với các thao tác cần làm để khám phá dữ liệu.

Nhiệm vụ 1: Khám phá dữ liệu COVID. Tính mean, median, mode, variance, standard deviation, range, percentile, quartile, interquartile range (IQR) sử dụng thư viện numpy và stats trên tập dữ liệu COVID.

Ý nghĩa từng giá trị:

- mean: Giá trị trung bình cộng của dữ liệu.
- median: Giá trị nằm giữa khi dữ liệu được sắp xếp.
- mode: Giá trị xuất hiện nhiều nhất trong dữ liệu.
- variance: Đo độ phân tán của dữ liệu quanh giá trị trung bình.
- standard deviation: Độ lệch chuẩn, căn bậc hai của phương sai.
- range: Khoảng cách giữa giá trị lớn nhất và nhỏ nhất.
- percentile: Giá trị tại một phần trăm cụ thể của dữ liệu đã sắp xếp.
- quartile: Các giá trị chia dữ liệu thành bốn phần bằng nhau.
- interquartile range (IQR): Khoảng giữa tứ phân vị thứ 1 (Q1) và thứ 3 (Q3).

Kiểm tra thông tin về dữ liệu:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 533674 entries, 0 to 533673
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   continent    493610 non-null   object 
 1   location     533674 non-null   object 
 2   date         533674 non-null   object 
 3   total_cases  519912 non-null   float64
 4   new_cases    516574 non-null   float64
dtypes: float64(2), object(3)
memory usage: 20.4+ MB

(533674, 5)

```

Hình 1. 1 Kiểm tra thông tin về dữ liệu COVID

Sử dụng các hàm có trong python để có thể tính các giá trị cụ thể đối với new_case:

```

Mean: 10520.897379659062
Median: nan
Mode: 0.0
Variance: 13743778056.568876
Standard Deviation: 117233.86053768286
Max: 8401906.0
Min: 0.0
60th Percentile: nan
75th Percentile (Quartile): nan
IQR: nan

```

Hình 1. 2 Tính toán các giá trị trong dữ liệu của new_case

Tương tự với các cột dữ liệu tiếp theo(total_case):

```

Mean: 14171784.75685693
Median: nan
Mode: 0.0
Variance: 5439115895065073.0
Standard Deviation: 73750361.99955274
Max: 778602438.0
Min: 0.0
60th Percentile: nan
75th Percentile (Quartile): nan
IQR: nan

```

Hình 1. 3 Tính toán các giá trị trong dữ liệu của total_case

Nhiệm vụ 2: Khám phá và xử lý dữ liệu Marketing Campaign. Ý nghĩa của từng biến dữ liệu đầu vào:

- ID: Mã định danh duy nhất của khách hàng.
- Year_Birth: Năm sinh của khách hàng.
- Education: Trình độ học vấn của khách hàng (ví dụ: Graduation, PhD, Master, Basic, 2n Cycle).
- Marital_Status: Tình trạng hôn nhân của khách hàng (ví dụ: Married, Single, Together, Divorced).
- Income: Thu nhập hàng năm của hộ gia đình khách hàng.
- Kidhome: Số lượng trẻ nhỏ trong gia đình khách hàng.
- Teenhome: Số lượng thanh thiếu niên (teen) trong gia đình khách hàng.
- Dt_Customer: Ngày khách hàng đăng ký thành khách hàng của công ty.
- Recency: Số ngày kể từ lần mua hàng cuối cùng của khách hàng.
- NumStorePurchases: Số lần mua hàng được thực hiện trực tiếp tại cửa hàng.
- NumWebVisitsMonth: Số lần khách hàng truy cập trang web của công ty trong tháng gần nhất

Tập dữ liệu marketing_data bao gồm 2.240 bản ghi và 11 thuộc tính, chứa thông tin liên quan đến đặc điểm nhân khẩu học và hành vi mua hàng của khách hàng trong chiến dịch marketing.

Sử dụng hàm info() của thư viện Pandas, nhóm thu được thông tin tổng quát về cấu trúc dữ liệu, kiểu dữ liệu của từng cột, cùng số lượng giá trị không rỗng. Kết quả cho thấy một số cột như Year_Birth, Education, Marital_Status, Income, và Dt_Customer có một vài giá trị bị thiếu.

```

Thông tin về dữ liệu marketing_data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   ID                2240 non-null    int64  
 1   Year_Birth        2239 non-null    float64 
 2   Education         2239 non-null    object  
 3   Marital_Status    2239 non-null    object  
 4   Income             2215 non-null    float64 
 5   Kidhome           2239 non-null    float64 
 6   Teenhome          2239 non-null    float64 
 7   Dt_Customer       2239 non-null    object  
 8   Recency            2239 non-null    float64 
 9   NumStorePurchases 2239 non-null    float64 
 10  NumWebVisitsMonth 2239 non-null    float64 
dtypes: float64(7), int64(1), object(3)
memory usage: 192.6+ KB

```

Hình 1. 4 Thông tin về dữ liệu marketing_data

Thống kê mô tả được thực hiện bằng `describe()` giúp nhận diện đặc điểm của các biến định lượng. Cụ thể:

- Thu nhập khách hàng (Income) có giá trị trung bình khoảng 52,244 với độ lệch chuẩn 25,178, cho thấy sự chênh lệch khá lớn giữa các nhóm khách hàng.
- Năm sinh (Year_Birth) trung bình khoảng 1969, trải rộng từ 1893 đến 1996.
- Các chỉ số hành vi mua hàng như Recency (khoảng cách ngày mua gần nhất) và NumStorePurchases có phân bố khá đồng đều, thể hiện sự đa dạng về hành vi tiêu dùng.

Dữ liệu sẽ được xử lý để loại bỏ các yếu tố gây nhiễu như loại bỏ dòng trùng lặp nhằm đảm bảo tính toàn vẹn dữ liệu, xóa các dòng hoặc cột không cần thiết. Việc làm sạch dữ liệu giúp bộ dữ liệu marketing_data trở nên gọn gàng, chính xác hơn, tạo tiền đề cho việc phân tích thống kê và trực quan hóa ở các bước tiếp theo.

	ID	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	NumStorePurchases	NumWebVisitsMonth
0	5524	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2174	Graduation	Single	46344.0	1.0	1.0	08/03/2014	38.0	2.0	5.0
2	4141	Graduation	Together	71613.0	0.0	0.0	21/08/2013	26.0	10.0	4.0
3	6182	Graduation	Together	26646.0	1.0	0.0	10/02/2014	26.0	4.0	6.0
4	5324	PhD	Married	58293.0	1.0	0.0	19/01/2014	94.0	6.0	5.0
...
2235	10870	Graduation	Married	61223.0	0.0	1.0	13/06/2013	46.0	4.0	5.0
2236	4001	PhD	Together	64014.0	2.0	1.0	10/06/2014	56.0	5.0	7.0
2237	7270	Graduation	Divorced	56981.0	0.0	0.0	25/01/2014	91.0	13.0	6.0
2238	8235	Master	Together	69245.0	0.0	1.0	24/01/2014	8.0	10.0	3.0
2239	9405	PhD	Married	52869.0	1.0	1.0	15/10/2012	40.0	4.0	7.0

Hình 1. 5 Tiền xử lý dữ liệu

Trong giai đoạn tiền xử lý, dữ liệu được chuẩn hóa và chuyển đổi để đảm bảo tính nhất quán và dễ hiểu:

- Cột Teenhome được thay thế giá trị số bằng các nhãn định danh "has teen" và "has no teen", giúp tăng khả năng diễn giải.
- Cột Income được xử lý các giá trị bị thiếu bằng cách thay thế bằng 0, đảm bảo không có giá trị rỗng trong quá trình phân tích.
- Sau đó, kiểu dữ liệu của Income được chuyển từ float sang int nhằm đồng bộ hóa định dạng dữ liệu.

Những bước này giúp làm sạch và chuẩn hóa dữ liệu, hỗ trợ cho các bước thống kê và phân tích sâu hơn.

Dữ liệu ban đầu được kiểm tra các giá trị bị thiếu bằng phương thức isnull().sum(), giúp xác định số lượng phần tử NaN trong từng cột. Sau khi xác định, nhóm quyết định loại bỏ các bản ghi có giá trị bị thiếu bằng phương thức dropna(how='any').

Kết quả thu được là một bộ dữ liệu hoàn chỉnh, không còn giá trị thiếu, đảm bảo tính toàn vẹn và đồng nhất cho các bước phân tích tiếp theo. Việc so sánh kích thước trước và sau xử lý cho thấy số lượng bản ghi bị loại bỏ là không đáng kể, do đó không ảnh hưởng lớn đến chất lượng mẫu dữ liệu.

2. Bài toán 2

Thực hiện thống kê mô tả trên tập dữ liệu về phân loại chất lượng rượu đỏ. Tập dữ liệu này liên quan đến các loại rượu Vinho Verde của Bồ Đào Nha, chỉ bao gồm rượu đỏ. Dữ liệu được thu thập và công bố bởi Cortez et al. (2009).

Tập dữ liệu này chỉ chứa các biến lý hóa (physicochemical) và cảm quan (sensory). Do các vấn đề về bảo mật và hậu cần, không có dữ liệu về các yếu tố thương mại hay nông nghiệp như loại nho, thương hiệu rượu, hoặc giá bán.

Các Biến Đầu vào (Input Variables - Dựa trên kiểm tra lý hóa) - Đây là các đặc trưng được sử dụng để dự đoán chất lượng rượu:

- fixed acidity: Độ axit cố định (không dễ bay hơi), chủ yếu là axit tartaric.
- volatile acidity: Độ axit dễ bay hơi, chủ yếu là axit acetic (lượng cao thường cho thấy rượu bị hỏng).
- citric acid: Axit citric, có thể làm tăng "độ tươi" và hương vị cho rượu.
- residual sugar: Lượng đường còn lại sau khi quá trình lên men dừng lại.
- chlorides: Lượng muối trong rượu.
- free sulfur dioxide: Dạng SO₂ không liên kết, được thêm vào để ngăn chặn sự phát triển của vi khuẩn và quá trình oxy hóa.
- total sulfur dioxide: Tổng lượng SO₂ ở dạng tự do và liên kết
- density: Mật độ, có liên quan đến nồng độ cồn và lượng đường.
- pH: Mức độ axit, có ảnh hưởng đến màu sắc và tiềm năng oxy hóa của rượu.
- sulphates: Chất phụ gia rượu (thường ở dạng kali_sulphate) có thể góp phần vào mức SO₂ tổng thể.
- Alcohol: Nồng độ cồn trong rượu.
- quality: Chất lượng rượu, được đánh giá dựa trên dữ liệu cảm quan (thường là điểm trung bình từ các nhà nếm rượu chuyên nghiệp), với thang điểm từ 0 đến 10.

Thông tin cơ bản về tập dữ liệu:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   fixed acidity      1599 non-null    float64
 1   volatile acidity   1599 non-null    float64
 2   citric acid        1599 non-null    float64
 3   residual sugar     1599 non-null    float64
 4   chlorides          1599 non-null    float64
 5   free sulfur dioxide 1599 non-null    float64
 6   total sulfur dioxide 1599 non-null    float64
 7   density            1599 non-null    float64
 8   pH                 1599 non-null    float64
 9   sulphates          1599 non-null    float64
 10  alcohol            1599 non-null    float64
 11  quality             1599 non-null    int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB

```

Hình 1. 6 Thông tin về dữ liệu phân loại chất lượng rượu

Số dòng và số cột của dataset là: 1599 dòng và 12 cột. Nhưng cần kiểm tra thêm để chắc chắn rằng không có dữ liệu nào bị thiếu.

Để phân tích đặc trưng thống kê của biến *quality*, nhóm đã sử dụng các phép đo mô tả cơ bản bao gồm: Mode (giá trị xuất hiện nhiều nhất), Variance (phương sai), Percentile 50 (trung vị), Quartile (Q3 – tứ phân vị trên), IQR (khoảng tứ phân vị). Các giá trị này được tính bằng các hàm trong thư viện *scipy.stats* và *numpy*, giúp mô tả sự phân bố của chất lượng rượu trong tập dữ liệu.

```

# Get the mode of the data
data_mode = stats.mode(df["quality"])
# Obtain the variance of the data
data_variance = np.var(df["quality"])
# Obtain the 50th percentile of the data
data_percentile = np.percentile(df["quality"], 50)
# Obtain the quartiles of the data
data_quartile = np.quantile(df["quality"], 0.75)
# Get the IQR of the data
data_IQR = stats.iqr(df["quality"])
print(data_mode, data_variance, data_percentile, data_quartile, data_IQR)

```

ModeResult(mode=5, count=681) 0.6517605398308277 6.0 6.0 1.0

Hình 1. 7 Tính toán các phép đo cơ bản của tập dữ liệu

Thông kê mô tả cho toàn bộ dữ liệu ở tập này với các giá trị như: mean, srd, min, phân vị thứ 25, phân vị thứ 50, phân vị thứ 75, max

	count	mean	std	min	25%	50%	75%	max
fixed acidity	1599.000000	8.319637	1.741096	4.600000	7.100000	7.900000	9.200000	15.900000
volatile acidity	1599.000000	0.527821	0.179060	0.120000	0.390000	0.520000	0.640000	1.580000
citric acid	1599.000000	0.270976	0.194801	0.000000	0.090000	0.260000	0.420000	1.000000
residual sugar	1599.000000	2.538806	1.409928	0.900000	1.900000	2.200000	2.600000	15.500000
chlorides	1599.000000	0.087467	0.047065	0.012000	0.070000	0.079000	0.090000	0.611000
free sulfur dioxide	1599.000000	15.874922	10.460157	1.000000	7.000000	14.000000	21.000000	72.000000
total sulfur dioxide	1599.000000	46.467792	32.895324	6.000000	22.000000	38.000000	62.000000	289.000000
density	1599.000000	0.996747	0.001887	0.990070	0.995600	0.996750	0.997835	1.003690
pH	1599.000000	3.311113	0.154386	2.740000	3.210000	3.310000	3.400000	4.010000
sulphates	1599.000000	0.658149	0.169507	0.330000	0.550000	0.620000	0.730000	2.000000
alcohol	1599.000000	10.422983	1.065668	8.400000	9.500000	10.200000	11.100000	14.900000
quality	1599.000000	5.636023	0.807569	3.000000	5.000000	6.000000	6.000000	8.000000

Hình 1. 8 Thông kê mô tả cho toàn bộ dữ liệu

Thông qua dữ liệu ở hình trên ta thu được những giá trị trung bình, giá trị cao nhất như sau:

	Biến	Giá trị trung bình	Giá trị cao nhất
0	Fixed acidity (axit cố định)	8.31	15.90
1	Volatile acidity (axit bay hơi)	0.52	1.58
2	Citric acid (axit citric)	0.27	1.00
3	Residual sugar (đường dư)	2.53	15.50
4	Chlorides (clorua)	0.08	0.61
5	Free sulfur dioxide (SO_2 tự do)	15.87	72.00
6	Total sulfur dioxide (SO_2 tổng)	46.46	289.00
7	Density (khối lượng riêng)	0.99	1.00
8	pH	3.31	4.01
9	Sulphates (sunfat)	0.65	2.00
10	Alcohol (ancol)	10.42	14.90
11	Quality (chất lượng)	5.63	8.00

Hình 1. 9 Dữ liệu thu được sau quá trình lọc

Nhóm tiến hành kiểm tra các bản ghi trùng lặp trong tập dữ liệu bằng hàm `duplicated()` của thư viện Pandas.

```

fixed acidity volatile acidity citric acid residual sugar chlorides \
4           7.4          0.700     0.00      1.90    0.076
11          7.5          0.500     0.36      6.10    0.071
27          7.9          0.430     0.21      1.60    0.106
40          7.3          0.450     0.36      5.90    0.074
65          7.2          0.725     0.05      4.65    0.086
...
1563         7.2          0.695     0.13      2.00    0.076
1564         7.2          0.695     0.13      2.00    0.076
1567         7.2          0.695     0.13      2.00    0.076
1581         6.2          0.560     0.09      1.70    0.053
1596         6.3          0.510     0.13      2.30    0.076

free sulfur dioxide total sulfur dioxide density pH sulphates \
4             11.0          34.0   0.99780  3.51    0.56
11            17.0          102.0  0.99780 3.35    0.80
27            10.0          37.0   0.99660 3.17    0.91
40            12.0          87.0   0.99780 3.33    0.83
65            4.0           11.0   0.99620 3.41    0.39
...
1563          12.0          20.0   0.99546 3.29    0.54
1564          12.0          20.0   0.99546 3.29    0.54
1567          12.0          20.0   0.99546 3.29    0.54
1581          24.0          32.0   0.99402 3.54    0.60
1596          29.0          40.0   0.99574 3.42    0.75

alcohol quality
4            9.4          5
11           10.5          5
27           9.5          5
40           10.5          5
65           10.9          5
...
1563          10.1          5
1564          10.1          5
1567          10.1          5
1581          11.3          5
1596          11.0          6

```

Hình 1. 10 Kiểm tra thông tin trùng lặp của dữ liệu

Kết quả cho thấy tập dữ liệu có chứa một số dòng bị trùng hoàn toàn về giá trị ở tất cả các cột, điển hình như các bản ghi có chỉ số 1563, 1564 và 1567 có cùng thông tin về độ axit, nồng độ rượu, và chất lượng.

Việc tồn tại các bản ghi trùng lặp có thể gây sai lệch cho kết quả thống kê và mô hình phân tích, do đó nhóm đã tiến hành loại bỏ các dòng này bằng hàm `drop_duplicates()` để đảm bảo dữ liệu duy nhất và chính xác hơn cho các bước xử lý tiếp theo.

```

# Xử lý giá trị trùng lặp
df_duplicate = df.drop_duplicates()
df_duplicate_dropped = df_duplicate[df_duplicate.duplicated()]
print("Rows:", df_duplicate.shape, "\n", df_duplicate_dropped)

Rows: (1359, 12)
Empty DataFrame
Columns: [fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, quality]
Index: []

```

Dataset sau khi xử lý 240 hàng giá trị bị trùng còn lại 1359 hàng.

Hình 1. 11 Xử lý giá trị trùng lặp của dữ liệu

Có thể thấy rằng số hàng đã có sự thay đổi, chứng tỏ dữ liệu trùng lắp đã bị xóa đi. Để đảm bảo tính đầy đủ và chính xác của dữ liệu, nhóm tiến hành kiểm tra các giá trị bị thiếu trong tập dữ liệu bằng phương thức `isna().sum()` của thư viện Pandas.

```
# Hiển thị giá trị bị thiếu cho từng cột
print(df.isna().sum())

fixed acidity      0
volatile acidity   0
citric acid        0
residual sugar     0
chlorides          0
free sulfur dioxide 0
total sulfur dioxide 0
density            0
pH                 0
sulphates          0
alcohol            0
quality            0
dtype: int64
```

Hình 1. 12 Kiểm tra giá trị thiếu của dữ liệu

Kết quả cho thấy toàn bộ các cột, bao gồm: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, và quality đều có giá trị bị thiếu bằng 0.

Điều này chứng tỏ tập dữ liệu hoàn chỉnh và thân thiện, không có cột nào bị khuyết giá trị. Do đó, không cần thực hiện các bước xử lý thiếu dữ liệu (như điền giá trị trung bình, loại bỏ hàng, hoặc nội suy) mà có thể chuyển sang giai đoạn trực quan hóa dữ liệu ngay.

3. Bài toán 3

Thực hiện thống kê mô tả trên tập dữ liệu về bệnh tiểu đường. Bệnh tiểu đường là một trong những bệnh mạn tính phổ biến và nguy hiểm, đặc biệt trong nhóm người có yếu tố nguy cơ như béo phì, tuổi cao, hoặc có tiền sử gia đình. Bộ dữ liệu Pima Indians Diabetes được thu thập từ phụ nữ Pima (Arizona, Mỹ) ≥ 21 tuổi, nhằm mục tiêu phân tích các yếu tố sức khỏe liên quan đến khả năng mắc bệnh tiểu đường.

Thông tin cơ bản về dữ liệu bao gồm 768 mẫu với 8 thuộc tính đặc trưng và 1 biến mục tiêu(0 là không tiểu đường, 1 là tiểu đường). Với các thuộc tính bao gồm:

- Pregnancies: Số lần mang thai (kể cả những lần sảy thai hoặc thai chết lưu). Số nguyên (≥ 0)
- Glucose: Nồng độ glucose trong huyết tương sau khi xét nghiệm miệng dung nạp glucose trong 2 giờ (mg/dL). Giá trị 0 là bất thường và thường được coi là giá trị thiếu.
- BloodPressure: Huyết áp tâm trương (huyết áp dưới) tính bằng mmHg. Giá trị 0 là bất thường và thường được coi là giá trị thiếu.
- SkinThickness: Độ dày nếp gấp da cơ tam đầu(mm). Được sử dụng để ước tính tỷ lệ mỡ cơ thể. Giá trị 0 là bất thường.
- Insulin: Nồng độ insulin trong huyết thanh sau 2 giờ. Giá trị 0 là bất thường và thường được coi là giá trị thiếu.
- BMI: Chỉ số khối cơ thể (Body Mass Index)(kg/m²). Được tính bằng cân nặng chia cho bình phương chiều cao. Giá trị 0 là bất thường.
- DiabetesPedigreeFunction: Hàm phả hệ bệnh tiểu đường. Số thực (≥ 0) Đo lường nguy cơ di truyền mắc bệnh tiểu đường dựa trên tiền sử gia đình.
- Age: Tuổi của bệnh nhân. Số nguyên (năm)
- Outcome: Biến mục tiêu (biến kết quả). {0,1} 1: Mắc bệnh tiểu đường. 0: Không mắc bệnh tiểu đường.

Tiến hành kiểm tra thông tin dữ liệu đầu vào thông qua hàm info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   Pregnancies      768 non-null    int64  
 1   Glucose          768 non-null    int64  
 2   BloodPressure    768 non-null    int64  
 3   SkinThickness    768 non-null    int64  
 4   Insulin          768 non-null    int64  
 5   BMI              768 non-null    float64 
 6   DiabetesPedigreeFunction 768 non-null    float64 
 7   Age              768 non-null    int64  
 8   Outcome          768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Hình 1.13 Kiểm tra thông tin dữ liệu đầu vào của bệnh tiểu đường

Sau đó sử dụng phương thức describe() để thu được các thống kê cơ bản như giá trị trung bình, độ lệch chuẩn, giá trị nhỏ nhất, lớn nhất và các phần tử của từng thuộc tính trong tập dữ liệu.

Kết quả giúp đánh giá sơ bộ đặc điểm phân bố và mức độ biến động của các biến, hỗ trợ cho quá trình phân tích và trực quan hóa dữ liệu ở các bước tiếp theo.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Hình 1. 14 Các thống kê cơ bản của dữ liệu

Nhóm sử dụng lệnh pima_data.isnull().sum() để kiểm tra số lượng giá trị bị thiếu trong từng cột của tập dữ liệu. Kết quả cho thấy không có cột nào chứa giá trị thiếu (tất cả đều bằng 0). Do đó, dữ liệu hiện tại đầy đủ và không cần thực hiện bước xử lý giá trị thiếu, có thể tiến hành phân tích ngay.

```

Pregnancies          0
Glucose              0
BloodPressure        0
SkinThickness        0
Insulin              0
BMI                  0
DiabetesPedigreeFunction 0
Age                  0
Outcome              0
dtype: int64

```

Hình 1. 15 Kiểm tra dữ liệu bị thiếu trong tập dữ liệu

Trong quá trình làm sạch dữ liệu, nhóm tiến hành kiểm tra số lượng giá trị bằng 0 trong từng cột của bộ dữ liệu nhằm phát hiện các giá trị không hợp lý. Kết quả được thể hiện cho thấy các cột Insulin, SkinThickness và BloodPressure có số lượng giá trị 0 chiếm tỷ lệ cao, trong khi các cột DiabetesPedigreeFunction và Age không có giá trị 0 nào. Điều này cho thấy cần tiến hành xử lý hoặc thay thế các giá trị 0 ở những cột không hợp lý để đảm bảo tính chính xác của mô hình phân tích sau này.

```

Pregnancies          111
Glucose              5
BloodPressure        35
SkinThickness        227
Insulin              374
BMI                  11
DiabetesPedigreeFunction 0
Age                  0
Outcome              500
dtype: int64

```

Hình 1. 16 Kiểm tra dữ liệu có chứa giá trị 0

Nhóm tiến hành kiểm tra dữ liệu trùng lặp trong tập dữ liệu bằng lệnh pima_data.duplicated().sum(). Kết quả cho thấy không có bản ghi nào bị trùng lặp, do đó có thể giữ nguyên toàn bộ dữ liệu cho quá trình phân tích tiếp theo.

Một số cột trong tập dữ liệu như Glucose, BloodPressure, SkinThickness, Insulin và BMI chứa các giá trị bằng 0 - điều này không hợp lý về mặt y học. Do đó, nhóm đã thay thế các giá trị 0 bằng giá trị khuyết (NaN) để thuận tiện cho việc xử lý sau. Tiếp theo, phương pháp điền trung bình (Mean Imputation) được áp dụng thông qua SimpleImputer(strategy='mean') nhằm bổ sung giá trị hợp lý thay thế cho các giá trị thiếu.

Sau khi xử lý, tập dữ liệu không còn giá trị thiếu, sẵn sàng cho các bước phân tích và huấn luyện mô hình tiếp theo.

Số lượng giá trị thiếu sau khi xử lý:

```

Pregnancies          0
Glucose              0
BloodPressure        0
SkinThickness        0
Insulin              0
BMI                  0
DiabetesPedigreeFunction 0
Age                  0
Outcome              0
dtype: int64

```

Hình 1. 17 Xử lý tập dữ liệu có chứa giá trị thiếu

Để đảm bảo các đặc trưng (features) có cùng thang đo, nhóm tiến hành chuẩn hóa dữ liệu bằng công cụ MinMaxScaler trong thư viện Scikit-learn. Phương pháp này đưa toàn bộ giá trị của từng thuộc tính về trong khoảng [0, 1], giúp mô hình học

máy hội tụ nhanh hơn và tránh việc các biến có giá trị lớn chi phối kết quả huấn luyện. Với công thức chuẩn hóa được sử dụng như:

===== , , , , , , , , , , , ,

Trong đó:

- (X) là giá trị gốc,
 - (X_{\min}) là giá trị nhỏ nhất của cột,
 - (X_{\max}) là giá trị lớn nhất của cột,
 - (X') là giá trị đã được scale (nằm trong $[0,1]$).

Kết quả sau khi áp dụng MinMaxScaler cho toàn bộ dữ liệu (trừ cột mục tiêu Outcome) cho thấy các giá trị đầu vào đã được chuyển đổi về cùng một thang đo, sẵn sàng cho các bước phân tích và huấn luyện mô hình.

5 dòng đầu của dữ liệu đã chuẩn hóa:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	0.352941	0.670968	0.489796	0.304348	0.170130	0.314928	
1	0.058824	0.264516	0.428571	0.239130	0.170130	0.171779	
2	0.470588	0.896774	0.408163	0.240798	0.170130	0.104294	
3	0.058824	0.290323	0.428571	0.173913	0.096154	0.202454	
4	0.290000	0.600000	0.162265	0.284248	0.105206	0.500203	

	DiabetesPedigreeFunction	Age	Outcome
0	0.234415	0.483333	1
1	0.116567	0.166667	0
2	0.253629	0.183333	1
3	0.038002	0.000000	0
4	0.943638	0.200000	1

Hình 1.18 Dữ liệu sau khi áp dụng MinMaxScaler

Sau khi hoàn tất các bước xử lý và chuẩn hóa, tiến hành kiểm tra lại thông tin tổng quát của bộ dữ liệu bằng info(). Kết quả cho thấy không còn giá trị thiếu hoặc sai kiểu dữ liệu, các cột đã được chuẩn hóa thống nhất.

Đồng thời, sử dụng `value_counts()` để quan sát phân bố của biến mục tiêu (Outcome), giúp đánh giá sự cân bằng giữa hai lớp (0 và 1). Điều này rất quan trọng trong giai đoạn xây dựng mô hình học máy, nhằm tránh tình trạng mất cân bằng dữ liệu ảnh hưởng đến độ chính xác dự đoán.

CHƯƠNG 2: TRỰC QUAN HÓA DỮ LIỆU

I. Lý thuyết trực quan hóa dữ liệu

1. Vai trò của trực quan hóa dữ liệu

Trực quan hóa dữ liệu là quá trình sử dụng các yếu tố hình ảnh như đồ thị, biểu đồ hoặc bản đồ để trình bày dữ liệu. Quá trình này chuyển đổi dữ liệu phức tạp, có dung lượng lớn hoặc dữ liệu số thành hình ảnh trình bày trực quan có thể xử lý dễ dàng hơn. Các công cụ trực quan hóa dữ liệu cải thiện và tự động hóa quá trình giao tiếp bằng hình ảnh nhằm đảm bảo độ chính xác và chi tiết. Bạn có thể sử dụng những hình ảnh trình bày trực quan để trích xuất những thông tin chuyên sâu hữu ích từ dữ liệu thô[1].

Nếu ví dữ liệu là “nguyên liệu thô”, thì trực quan hóa chính là cách chế biến giúp nguyên liệu đó dễ “tiêu hóa” hơn cho người sử dụng – đặc biệt là những người không có chuyên môn về phân tích số liệu. Việc trực quan hóa giúp người xem không cần đọc hết dữ liệu vẫn “nhìn thấy” thông tin một cách nhanh chóng, trực tiếp và dễ hiểu[2].

Ví dụ đơn giản: Thay vì xem một bảng Excel dài 10.000 dòng để tìm hiểu doanh thu theo từng tháng, doanh nghiệp có thể dùng biểu đồ cột hoặc biểu đồ đường để thấy ngay tháng nào có doanh số tăng đột biến, đâu là giai đoạn doanh số đi xuống, hoặc xu hướng doanh thu trong cả năm[2].

Việc trình bày dữ liệu bằng hình ảnh không chỉ giúp dễ theo dõi, mà quan trọng hơn là hỗ trợ ra quyết định nhanh và chính xác hơn. Dưới góc độ doanh nghiệp, data visualization có các vai trò sau[2]:

- **Biến dữ liệu phức tạp thành hình ảnh dễ hiểu:** Khi một báo cáo tài chính có quá nhiều dòng và số liệu, người quản lý thường khó xác định đâu là điều quan trọng. Biểu đồ sẽ giúp tập trung sự chú ý vào các chỉ số chính – như lợi nhuận ròng, chi phí vượt kế hoạch, hoặc doanh số theo từng vùng.
- **Hỗ trợ ra quyết định nhanh hơn:** Thay vì mất nhiều thời gian tổng hợp, đọc và phân tích, người điều hành có thể nhìn dashboard (bảng điều khiển dữ liệu trực quan) để nắm được tình hình hoạt động chỉ trong vài phút, từ đó đưa ra hành động kịp thời, chẳng hạn như điều chỉnh ngân sách, thay đổi kế hoạch bán hàng hoặc phân bổ nhân sự.
- **Kết nối dữ liệu với chiến lược kinh doanh:** Trực quan hóa giúp dữ liệu không còn là công việc riêng của phòng phân tích, mà trở thành công cụ hỗ trợ ra quyết định chung cho toàn bộ doanh nghiệp. Khi tất cả các phòng ban đều nhìn thấy cùng một bức tranh dữ liệu, chiến lược và hành động sẽ đồng nhất, rõ ràng và hiệu quả hơn.

Các doanh nghiệp hiện đại thường xử lý lượng lớn dữ liệu từ nhiều nguồn dữ liệu khác nhau, chẳng hạn như:

- Trang web nội bộ và bên ngoài
- Thiết bị thông minh
- Hệ thống thu thập dữ liệu nội bộ
- Mạng xã hội

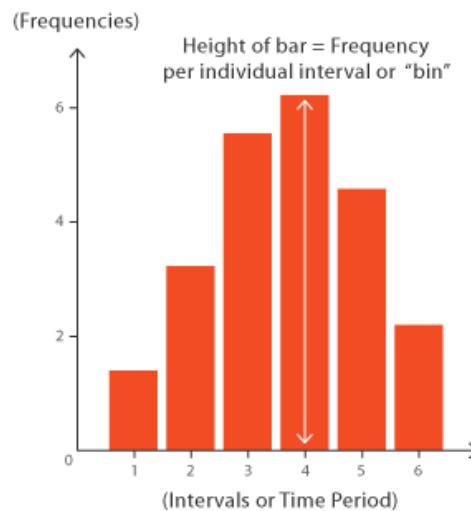
Tuy nhiên, dữ liệu thô có thể khó hiểu và khó sử dụng. Do đó, các nhà khoa học dữ liệu chuẩn bị và trình bày dữ liệu theo ngữ cảnh phù hợp. Họ định hình dữ liệu ở dạng trực quan để những người phụ trách đưa ra quyết định có thể xác định mối quan hệ giữa dữ liệu và phát hiện ra các mẫu hoặc xu hướng ẩn. Trực quan hóa dữ liệu tạo ra các thông điệp giúp nâng cao nghiệp vụ thông minh và hỗ trợ đưa ra quyết định cũng như lập kế hoạch chiến lược dựa trên dữ liệu[1].

2. Các loại biểu đồ phổ biến

Một số loại biểu đồ phổ biến như[3]:

Histogram: Là một dạng biểu đồ thể hiện tần suất dạng cột, hiển thị sự phân bố của một biến. Trục x thể hiện phạm vi và trục y biểu thị tần số.

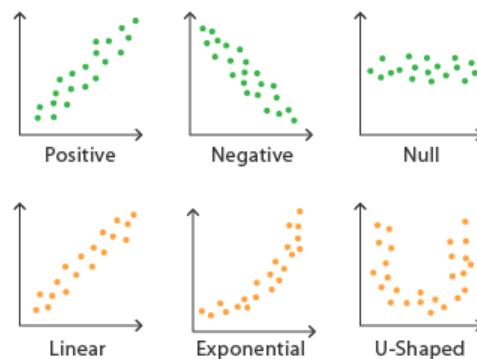
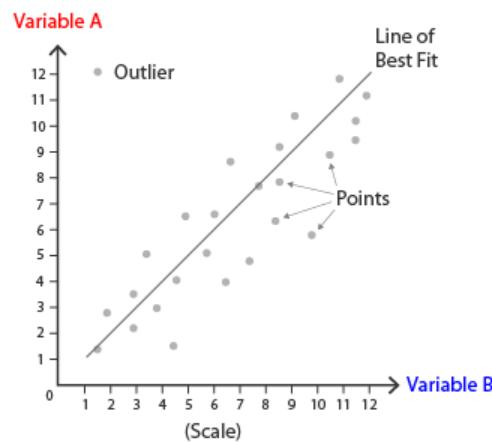
Histograms giúp đưa ra các ước tính về vị trí các giá trị tập trung, các điểm cực trị là gì, có khoảng trống hoặc giá trị bất thường nào không. Bên cạnh đó, chúng cũng hữu ích để đưa ra cái nhìn sơ bộ về phân phối xác suất.



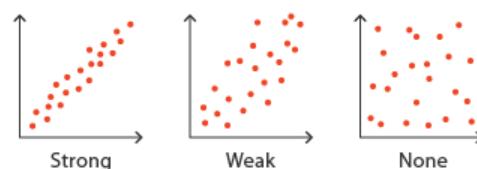
Hình 2.1 Ví dụ về biểu đồ histogram

Scatter plot chart: Là biểu đồ thường được sử dụng khi cần quan sát mối quan hệ giữa hai biến. Đây được xem là biểu đồ khá hữu ích khi nó nhanh chóng xác định mối tương quan tiềm năng giữa các điểm dữ liệu. Biểu

đồ phân tán thường được sử dụng để thể hiện các mối quan hệ nguyên nhân - kết quả. Mỗi tương quan này được biểu diễn dưới dạng các dấu chấm tròn đại diện cho 2 biến, với một biến phụ thuộc chạy cố định trên trục tung và một biến độc lập chạy cố định dựa vào trục hoành. Có nhiều loại tương quan khác nhau như: dương (các giá trị tăng cùng nhau), âm (giá trị này tăng khi giá trị kia giảm), null (không tương quan), tuyến tính, hàm mũ,...



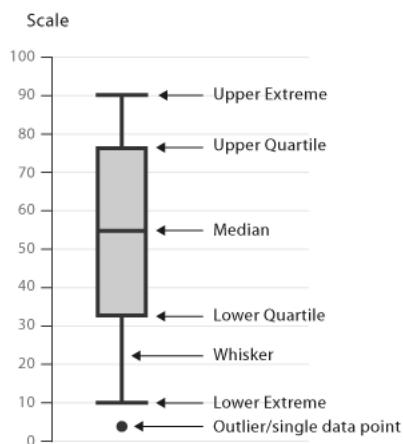
Correlation Strength:



Hình 2. 2 Ví dụ về biểu đồ Scatter plot

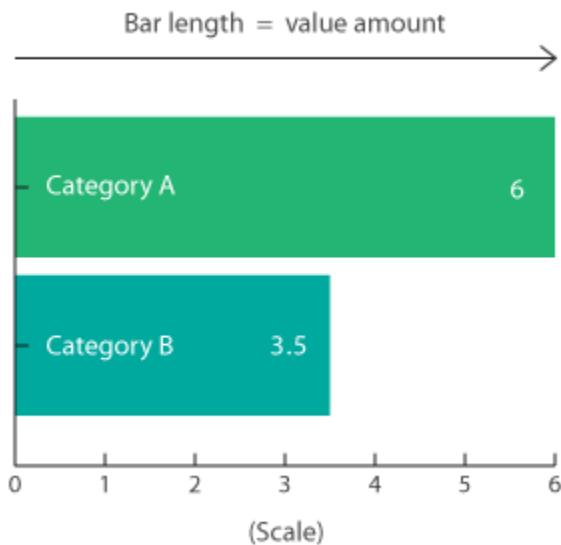
Box plot: Là biểu đồ diễn tả 5 vị trí phân bố của dữ liệu: giá trị nhỏ nhất (min), tứ phân vị thứ nhất (Q1), trung vị (median), tứ phân vị thứ 3 (Q3) và giá trị lớn nhất (max). Mặc dù Box plot còn khá mới so với Histogram và

Density plot, nhưng chúng có lợi thế là chiếm ít không gian hơn. Điều này rất hữu ích khi so sánh phân phối giữa nhiều nhóm hoặc datasets.



Hình 2. 3 Ví dụ về biểu đồ Box plot

Bar chart: Được biết đến là Bar Graph hay Column Graph, đây là một trong những biểu đồ dễ đọc nhất, giúp so sánh các dữ liệu phân loại. Một trục chứa các danh mục, trục còn lại chứa các giá trị.



Hình 2. 4 Biểu đồ Bar chart

3. Cách lựa chọn biểu đồ phù hợp

Để lựa chọn loại biểu đồ phù hợp, người phân tích cần dựa vào đặc điểm của dữ liệu cũng như mục tiêu thể hiện thông tin. Mỗi loại dữ liệu có cách trực quan

hóa riêng, giúp người xem dễ dàng nhận biết xu hướng, mối quan hệ hay sự khác biệt giữa các nhóm giá trị. Việc chọn đúng biểu đồ không chỉ làm dữ liệu trở nên dễ hiểu mà còn giúp kết quả phân tích trở nên thuyết phục và sinh động hơn.

Đối với dữ liệu phân loại, tức là những dữ liệu được chia thành các nhóm hoặc danh mục như giới tính, khu vực, loại sản phẩm hay ngành nghề, các biểu đồ cột và biểu đồ tròn thường được sử dụng phổ biến. Biểu đồ cột (bar chart) giúp so sánh giá trị giữa các nhóm khác nhau, cho phép người xem dễ dàng nhận ra nhóm nào có giá trị cao hơn hoặc thấp hơn. Trong khi đó, biểu đồ tròn (pie chart) thể hiện tỷ lệ phần trăm của từng nhóm trong tổng thể, giúp người xem hình dung được phần đóng góp của mỗi nhóm một cách trực quan.

Với dữ liệu số, tức là những dữ liệu biểu diễn bằng con số và có thể đo lường, các loại biểu đồ như histogram, boxplot và scatter plot được sử dụng rộng rãi. Biểu đồ histogram cho thấy phân bố của dữ liệu, giúp xác định xem dữ liệu có tập trung, lệch hay trải đều hay không. Biểu đồ hộp (boxplot) giúp nhận biết giá trị trung vị, phạm vi dữ liệu và các điểm ngoại lệ, rất hữu ích khi cần so sánh nhiều nhóm dữ liệu khác nhau. Còn biểu đồ phân tán (scatter plot) thể hiện mối quan hệ giữa hai biến số, giúp phát hiện xu hướng tăng, giảm hoặc những điểm bất thường trong dữ liệu.

Đối với dữ liệu chuỗi thời gian, tức là dữ liệu được thu thập theo trình tự thời gian như doanh thu theo tháng, nhiệt độ theo ngày hay số lượng người dùng theo năm, biểu đồ đường (line chart) là lựa chọn phù hợp nhất. Biểu đồ này thể hiện rõ sự thay đổi của giá trị theo thời gian, giúp người xem nhận biết xu hướng tăng giảm hoặc dao động. Ngoài ra, khi kết hợp nhiều đường trên cùng một biểu đồ, ta còn có thể so sánh sự biến động của nhiều yếu tố khác nhau trong cùng khoảng thời gian.

Tóm lại, việc lựa chọn biểu đồ phù hợp với đặc điểm dữ liệu là một bước quan trọng trong quá trình phân tích và trực quan hóa dữ liệu. Nó giúp truyền đạt thông tin một cách hiệu quả, làm nổi bật được bản chất của dữ liệu và hỗ trợ người xem trong việc đưa ra nhận định hoặc quyết định chính xác hơn.

4. Thư viện trực quan hóa trong Python

Trong Python, có nhiều thư viện hỗ trợ trực quan hóa dữ liệu, trong đó nổi bật nhất là Matplotlib, Seaborn và Plotly. Mỗi thư viện đều có những ưu điểm, hạn chế và mục đích sử dụng riêng, phù hợp với từng nhu cầu cụ thể của người dùng. Việc hiểu rõ sự khác biệt giữa chúng giúp ta lựa chọn công cụ hiệu quả nhất cho từng loại phân tích và biểu diễn dữ liệu.

Matplotlib là thư viện cơ bản và lâu đời nhất trong hệ sinh thái Python, được xem là nền tảng cho hầu hết các công cụ trực quan hóa khác. Điểm mạnh của Matplotlib là khả năng tùy chỉnh linh hoạt, cho phép người dùng điều chỉnh hầu như mọi chi tiết của biểu đồ — từ màu sắc, trục tọa độ, nhãn, đèn chói. Tuy nhiên, nhược điểm của nó là cần nhiều dòng lệnh để tạo ra biểu đồ hoàn chỉnh, nên đôi khi không phù hợp cho người mới bắt đầu hoặc những ai cần trực quan hóa nhanh chóng.

Seaborn được xây dựng dựa trên Matplotlib nhưng cung cấp giao diện thân thiện và đơn giản hơn. Thư viện này đặc biệt mạnh trong việc trực quan hóa dữ liệu thống kê, giúp người dùng dễ dàng tạo các biểu đồ như boxplot, heatmap, hay pairplot chỉ với vài dòng lệnh. Ngoài ra, Seaborn còn tự động xử lý các yếu tố như bảng màu, tỷ lệ, và phong cách, giúp biểu đồ trở nên đẹp mắt và chuyên nghiệp hơn mà không cần quá nhiều tùy chỉnh thủ công. Đây là lựa chọn lý tưởng khi cần biểu diễn mối quan hệ giữa các biến hoặc khám phá dữ liệu trong giai đoạn phân tích ban đầu.

Plotly, khác với hai thư viện trên, nổi bật với khả năng tạo biểu đồ tương tác. Các biểu đồ được xây dựng bằng Plotly cho phép người xem phóng to, thu nhỏ, di chuột để xem chi tiết, hoặc lọc dữ liệu trực tiếp trên biểu đồ. Chính vì vậy, Plotly rất phù hợp khi cần phát triển dashboard, báo cáo trực tuyến hoặc ứng dụng web có yếu tố tương tác cao. Thư viện này cũng hỗ trợ nhiều loại biểu đồ hiện đại và có thể kết hợp tốt với các framework như Dash để xây dựng hệ thống trực quan hóa dữ liệu hoàn chỉnh.

Tóm lại, Matplotlib phù hợp cho người dùng cần tùy chỉnh chi tiết và linh hoạt, Seaborn thích hợp cho trực quan hóa thống kê nhanh và đẹp mắt, còn Plotly là lựa chọn tối ưu khi cần biểu đồ tương tác cho các ứng dụng web hoặc dashboard động.

5. Nguyên tắc thiết kế biểu đồ

Để tạo ra một biểu đồ trực quan dễ hiểu và hiệu quả, cần tuân thủ một số nguyên tắc cơ bản. Trước hết, biểu đồ nên đơn giản và rõ ràng, tránh sử dụng quá nhiều chi tiết gây rối mắt. Mọi yếu tố hiển thị phải hướng đến việc làm nổi bật thông tin chính.

Tiếp theo, sử dụng màu sắc và tỷ lệ hợp lý giúp người xem dễ phân biệt các nhóm dữ liệu mà không bị rối. Màu sắc nên có độ tương phản vừa phải và thể hiện đúng ý nghĩa, chẳng hạn như dùng gam nóng cho giá trị cao và gam lạnh cho giá trị thấp.

Ngoài ra, cần nhấn mạnh thông tin quan trọng bằng cách dùng màu nổi bật hoặc làm nổi phần dữ liệu cần chú ý. Điều này giúp người xem nhanh chóng nhận ra điểm chính của biểu đồ.

Cuối cùng, biểu đồ phải có nhãn trực, tiêu đề và chủ thích rõ ràng để người xem hiểu nội dung mà không cần suy đoán. Một biểu đồ tốt là biểu đồ truyền tải thông tin chính xác, trực quan và dễ ghi nhớ.

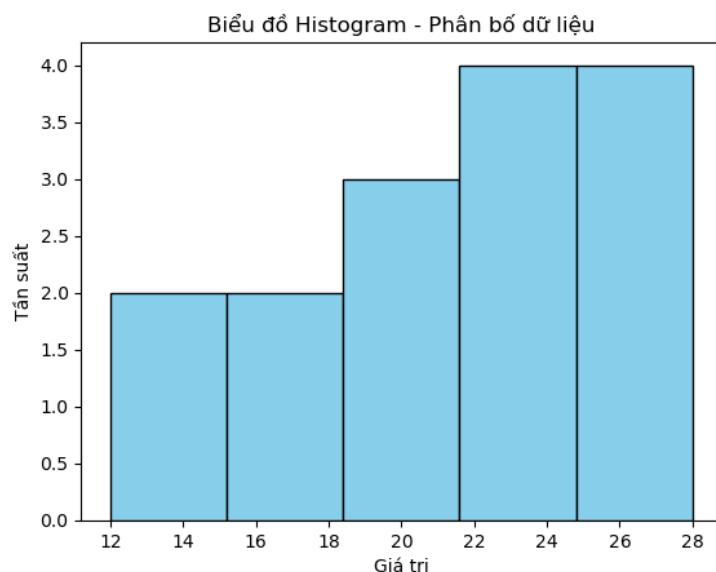
6. Cách tạo biểu đồ

Để tạo một biểu đồ đơn giản như **histogram** hoặc **bar chart** bằng thư viện **Matplotlib** trong Python, bạn chỉ cần vài dòng lệnh cơ bản. Dưới đây là ví dụ minh họa cho từng loại biểu đồ.

- Biểu đồ histogram:

```
import matplotlib.pyplot as plt
# Dữ liệu mẫu
data = [12, 15, 17, 18, 19, 20, 21, 22, 22, 23, 24, 25, 26, 27, 28]

# Vẽ biểu đồ histogram
plt.hist(data, bins=5, color='skyblue', edgecolor='black')
plt.title('Biểu đồ Histogram - Phân bố dữ liệu')
plt.xlabel('Giá trị')
plt.ylabel('Tần suất')
plt.show()
```



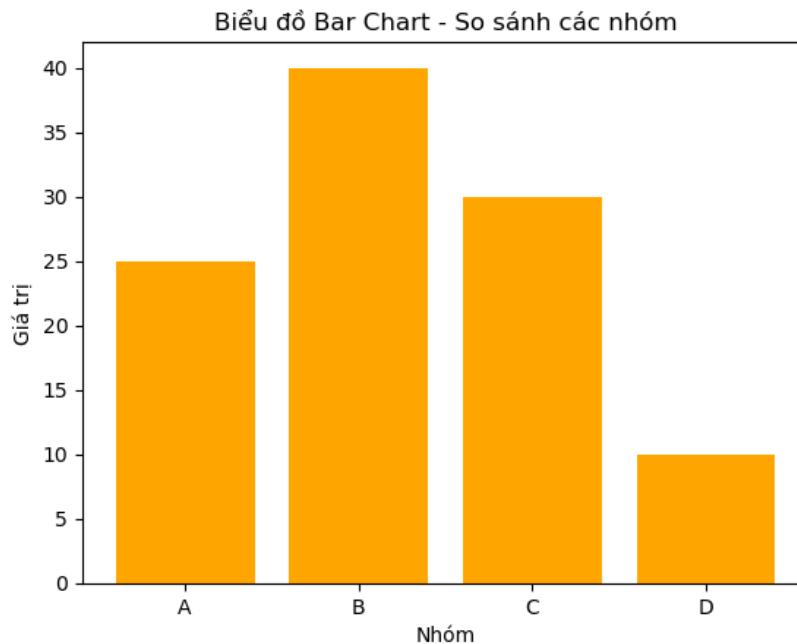
Hình 2. 5 Biểu đồ phân bố dữ liệu

- Biểu đồ bar chart:

```

import matplotlib.pyplot as plt
# Dữ liệu mẫu
categories = ['A', 'B', 'C', 'D']
values = [25, 40, 30, 10]
# Vẽ biểu đồ cột
plt.bar(categories, values, color='orange')
plt.title('Biểu đồ Bar Chart - So sánh các nhóm')
plt.xlabel('Nhóm')
plt.ylabel('Giá trị')
plt.show()

```



Hình 2. 6 Biểu đồ Bar Chart

7. Cách xuất biểu đồ

Sau khi tạo biểu đồ trong Python, bạn có thể dễ dàng xuất chúng ra các định dạng như PNG, PDF, hoặc HTML để sử dụng trong báo cáo, bài thuyết trình hoặc chia sẻ trực tuyến. Tùy theo thư viện được sử dụng, cách lưu sẽ khác nhau.

Matplotlib hỗ trợ xuất biểu đồ ra nhiều định dạng hình ảnh khác nhau như PNG, JPG, PDF hoặc SVG. Sau khi vẽ biểu đồ, chỉ cần dùng lệnh `plt.savefig()` với tên file và phần mở rộng tương ứng.

```

import matplotlib.pyplot as plt
plt.bar(['A', 'B', 'C'], [10, 20, 15], color='skyblue')
plt.title('Ví dụ biểu đồ cột')

# Lưu biểu đồ ra file PNG
plt.savefig("bieu_do.png")
# Hoặc lưu ra định dạng PDF
plt.savefig("bieu_do.pdf")

```

Plotly cho phép xuất biểu đồ tương tác dưới dạng file HTML, rất hữu ích khi cần trình bày trực tuyến hoặc chèn vào website.

```

import plotly.express as px

fig = px.bar(x=['A', 'B', 'C'], y=[10, 20, 15], title='Biểu đồ cột với Plotly')

# Xuất biểu đồ ra file HTML có thể tương tác
fig.write_html("bieu_do.html")

```

II. Làm quen với trực quan hóa dữ liệu

1. Bài toán 1

Thực hiện các nhiệm vụ trong bài toán để làm quen với các công cụ trực quan hóa dữ liệu. Dữ liệu thực hiện là dữ liệu về giá nhà.

Nhiệm vụ 1: Chuẩn bị dữ liệu cho trực quan hóa. Để chuẩn bị dữ liệu cho việc trực quan hóa, các bước tiền xử lý cơ bản đã được thực hiện:

- **Nạp dữ liệu:** Tập dữ liệu HousingPricesData.csv được nạp bằng thư viện Pandas. Bộ dữ liệu ghi lại 924 bất động sản tại Amsterdam (tháng 8/2021) với các thông tin về giá bán, diện tích, số phòng và vị trí địa lý, phục vụ cho việc phân tích thị trường và dự đoán giá nhà.

- **Cấu trúc dữ liệu:** Gồm 924 dòng, 8 cột:

- Unnamed: 0: Số thứ tự chỉ mục
- Address: Địa chỉ đầy đủ của bất động sản tại Amsterdam
- Zip: Mã bưu chính (postal code)
- Price: Giá bán bất động sản (EUR)
- Area: Diện tích (m^2)
- Room: Số lượng phòng
- Lon: Kinh độ
- Lat: Vĩ độ

- **Chọn biến:** Tập dữ liệu được giới hạn bao gồm các biến: Zip, Price, Area, và Room.

- **Kỹ thuật đặc trưng (Feature Engineering):** Một biến mới được tạo ra là Giá trên mỗi mét vuông (PriceperSqm) bằng cách chia biến Price

cho biến Area. Biến này giúp so sánh giá trị tài sản dựa trên mật độ (giá/diện tích) thay vì chỉ dựa trên tổng giá trị tuyệt đối

```
import pandas as pd
houseprices_data = pd.read_csv("data/HousingPricesData.csv")
houseprices_data = houseprices_data[['Zip', 'Price', 'Area', 'Room']]
# Create a PriceperSqm variable based on the Price and Area variables:
houseprices_data['PriceperSqm'] = houseprices_data['Price']/houseprices_data['Area']
```

Nhiệm vụ 2: Trực quan hóa dữ liệu với Matplotlib. Matplotlib là thư viện nền tảng được sử dụng để tạo các biểu đồ tĩnh. Trong nhiệm vụ này, biểu đồ thanh (bar chart) đã được chọn để trực quan hóa 10 khu vực (Zip code) có giá nhà cao nhất. Để thực hiện, dữ liệu đã được sắp xếp giảm dần theo cột Price.

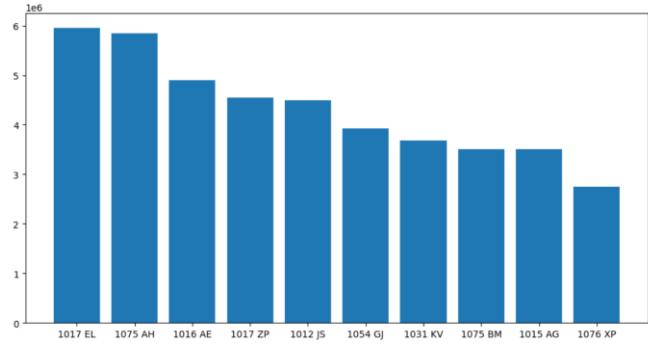
```
houseprices_sorted = houseprices_data.sort_values('Price', ascending = False)
houseprices_sorted.head()
```

	Zip	Price	Area	Room	PriceperSqm
195	1017 EL	5950000.0	394	10	15101.522843
837	1075 AH	5850000.0	480	14	12187.500000
305	1016 AE	4900000.0	623	13	7865.168539
103	1017 ZP	4550000.0	497	13	9154.929577
179	1012 JS	4495000.0	178	5	25252.808989

Hình 2. 7 Dữ liệu được sắp xếp theo giá cả

- Trường hợp cơ bản: Thực hiện tạo biểu đồ thanh cơ bản hiển thị Price theo Zip của 10 khu vực hàng đầu

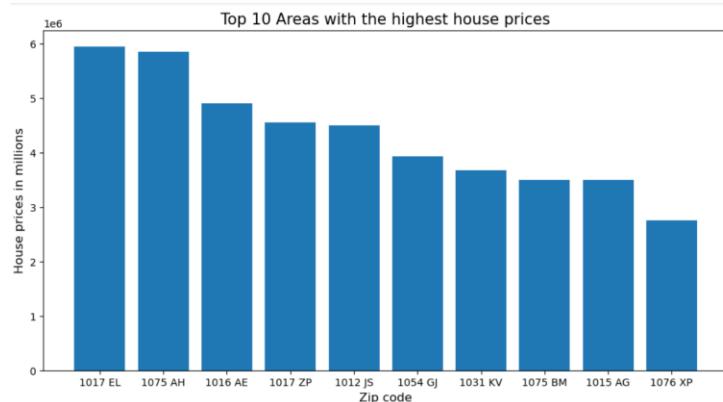
```
plt.figure(figsize= (12,6))
x = houseprices_sorted['Zip'][0:10]
y = houseprices_sorted['Price'][0:10]
plt.bar(x,y)
plt.show()
```



Hình 2. 8 Biểu đồ thẻ hiện top 10 khu vực hàng đầu

- Trường hợp nâng cao: Để tăng tính dễ đọc và chuyên nghiệp của biểu đồ, các tùy chỉnh nâng cao đã được áp dụng, bao gồm: Điều chỉnh kích thước hình ảnh (figsize= (12,6)), Thiết lập tiêu đề chính xác cho biểu đồ (plt.title), Thêm nhãn cho trục hoành (plt.xlabel) và trục tung (plt.ylabel), đồng thời tùy chỉnh cỡ chữ cho các nhãn và giá trị trực.

```
plt.figure(figsize= (12,6))
plt.bar(x,y)
plt.title('Top 10 Areas with the highest house prices', fontsize=15)
plt.xlabel('Zip code', fontsize = 12)
plt.xticks(fontsize=10)
plt.ylabel('House prices in millions', fontsize=12)
plt.yticks(fontsize=10)
plt.show()
```



Hình 2. 9 Biểu đồ top 10 khu vực có giá nhà cao nhất

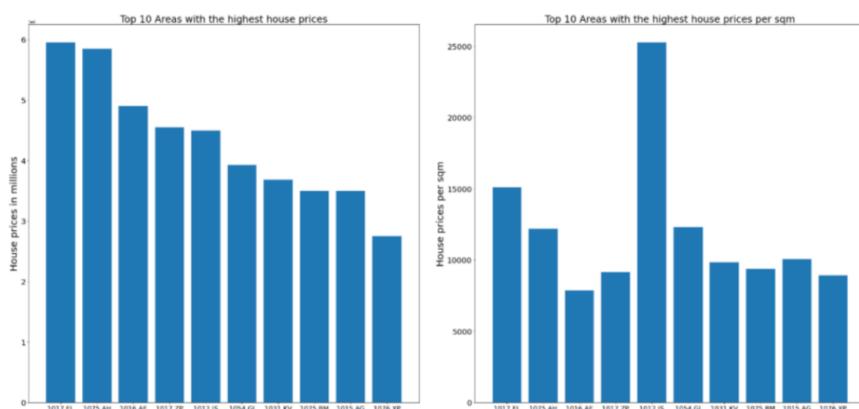
- Trình bày đa góc độ: Để có cái nhìn toàn diện hơn về 10 khu vực đắt nhất, hai biểu đồ con được tạo ra trên cùng một hình (figure) bằng cách sử dụng plt.subplot(1,2,x). Hai biểu đồ này so sánh:

- Biểu đồ bên trái: 10 khu vực theo **Giá nhà (Price)**.

- Biểu đồ bên phải: 10 khu vực theo **Giá trên mỗi mét vuông (PriceperSqm)**.

Việc trình bày đa góc độ giúp người phân tích dễ dàng nhận ra liệu các khu vực có giá cao nhất có thực sự là những khu vực có giá trị mật độ cao nhất hay không

```
fig, ax = plt.subplots(figsize=(40,18))
x = houseprices_sorted['Zip'][0:10]
y = houseprices_sorted['Price'][0:10]
y1 = houseprices_sorted['PriceperSqm'][0:10]
plt.subplot(1,2,1)
plt.bar(x,y)
plt.xticks(fontsize=17)
plt.ylabel('House prices in millions', fontsize=25)
plt.yticks(fontsize=20)
plt.title('Top 10 Areas with the highest house prices',
          fontsize=25)
plt.subplot(1,2,2)
plt.bar(x,y1)
plt.xticks(fontsize=17)
plt.ylabel('House prices per sqm', fontsize=25)
plt.yticks(fontsize=20)
plt.title('Top 10 Areas with the highest house prices per sqm',
          fontsize=25)
plt.show()
```



Hình 2. 10 Biểu đồ biểu thị top 10 khu vực theo giá nhà(bên trái) và giá trên mỗi mét vuông(bên phải)

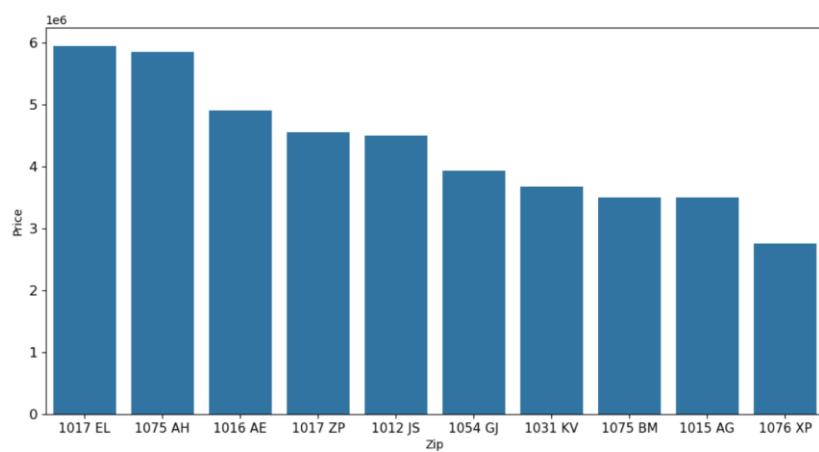
Nhiệm vụ 3: Trực quan hóa dữ liệu với thư viện Seaborn. Seaborn là thư viện được xây dựng dựa trên Matplotlib, chuyên dùng để tạo các biểu đồ thống kê với tính thẩm mỹ cao hơn.

- Trường hợp cơ bản: Thực hiện tạo biểu đồ thanh cơ bản bằng hàm sns.barplot để trực quan hóa Price theo Zip

```

import matplotlib.pyplot as plt
import seaborn as sns
# case 1: basic
plt.figure(figsize= (12,6))
data = houseprices_sorted[0:10]
sns.barplot(data= data, x= 'Zip',y = 'Price')

```



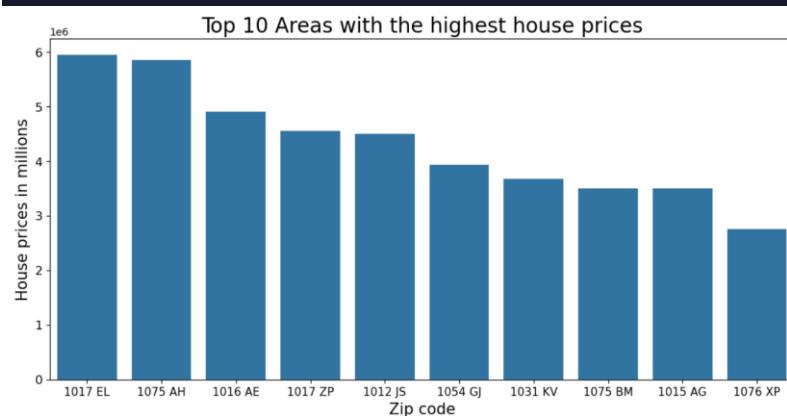
Hình 2. 11 Biểu đồ trực quan hóa Price theo Zip

- Trường hợp nâng cao: Các tùy chỉnh nâng cao được áp dụng thông qua đối tượng *ax* để thiết lập nhãn trực và tiêu đề, đảm bảo biểu đồ rõ ràng và cung cấp đầy đủ thông tin.

```

# case 2: advanced 1
plt.figure(figsize= (12,6))
data = houseprices_sorted[0:10]
ax = sns.barplot(data= data, x= 'Zip',y = 'Price')
ax.set_xlabel('Zip code',fontsize = 15)
ax.set_ylabel('House prices in millions', fontsize = 15)
ax.set_title('Top 10 Areas with the highest house prices', fontsize= 20)

```

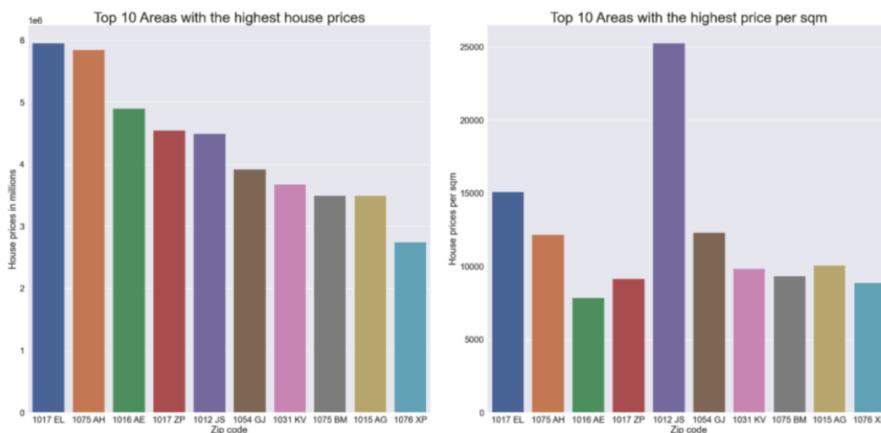


Hình 2. 12 Biểu đồ top 10 giá nhà cao nhất

- Trình bày đa góc độ: Tương tự như Matplotlib, sử dụng *plt.subplots(1, 2)* để tạo ra hai biểu đồ con, kết hợp với hàm

`sns.barplot` của Seaborn để so sánh Giá nhà và Giá trên mỗi mét vuông. Thao tác này còn bao gồm việc thiết lập `sns.set(font_scale = 3)` để điều chỉnh kích thước phông chữ toàn cục cho các biểu đồ con, giúp tăng cường khả năng hiển thị.

```
# case 3: view multiple perspectives at once
fig, ax = plt.subplots(1, 2, figsize=(40,18))
data = houseprices_sorted[0:10]
sns.set(font_scale = 3)
ax1 = sns.barplot(data= data, x= 'Zip',y = 'Price', ax = ax[0])
ax1.set_xlabel('Zip code')
ax1.set_ylabel('House prices in millions')
ax1.set_title('Top 10 Areas with the highest house prices')
ax2 = sns.barplot(data= data, x= 'Zip',y = 'PriceperSqm',
ax=ax[1])
ax2.set_xlabel('Zip code')
ax2.set_ylabel('House prices per sqm')
ax2.set_title('Top 10 Areas with the highest price per sqm')
```



Hình 2. 13 Biểu đồ sử dụng seaborn để so sánh giá nhà và giá trên mỗi mét vuông

2. Bài toán 2

Thực hiện trực quan hóa dữ liệu trên tập dữ liệu về phân loại chất lượng rượu đó.

Nhiệm vụ 1: Chuẩn bị dữ liệu cho trực quan hóa. Để chuẩn bị dữ liệu cho việc trực quan hóa, các bước tiền xử lý cơ bản đã được thực hiện:

- **Nạp dữ liệu:** Tập dữ liệu này chỉ chứa các biến lý hóa (physicochemical) và cảm quan (sensory). Do các vấn đề về bảo mật và hậu

cần, không có dữ liệu về các yếu tố thương mại hay nông nghiệp như loại nho, thương hiệu rượu, hoặc giá bán.

- **Cấu trúc dữ liệu:** Gồm 1599 dòng, 12 cột:

- fixed acidity: Độ axit cố định (không dễ bay hơi), chủ yếu là axit tartaric.
- volatile acidity: Độ axit dễ bay hơi, chủ yếu là axit acetic (lượng cao thường cho thấy rượu bị hỏng).
- citric acid: Axit citric, có thể làm tăng "độ tươi" và hương vị cho rượu.
- residual sugar: Lượng đường còn lại sau khi quá trình lên men dừng lại.
- chlorides: Lượng muối trong rượu.
- free sulfur dioxide: Dạng SO₂ không liên kết, được thêm vào để ngăn chặn sự phát triển của vi khuẩn và quá trình oxy hóa.
- total sulfur dioxide: Tổng lượng SO₂ ở dạng tự do và liên kết
- density: Mật độ, có liên quan đến nồng độ cồn và lượng đường.
- pH: Mức độ axit, có ảnh hưởng đến màu sắc và tiềm năng oxy hóa của rượu.
- sulphates: Chất phụ gia rượu (thường ở dạng kali_sulphate) có thể góp phần vào mức SO₂ tổng thể.
- Alcohol: Nồng độ cồn trong rượu.
- quality: Chất lượng rượu, được đánh giá dựa trên dữ liệu cảm quan (thường là điểm trung bình từ các nhà nếm rượu chuyên nghiệp), với thang điểm từ 0 đến 10.

- **Chọn biến:** Bộ dữ liệu rượu vang gồm 12 biến đầu vào mô tả đặc tính hóa học và 1 biến đầu ra biểu thị chất lượng rượu. Tuy nhiên, trong quá trình trực quan hóa, chỉ một số biến tiêu biểu được lựa chọn để biểu diễn bằng biểu đồ Histogram, nhằm tránh sự trùng lặp và giúp việc so sánh trở nên rõ ràng hơn. Cụ thể, các biểu đồ histogram thể hiện phân bố của các biến đại diện như fixed_acidity, volatile_acidity, citric_acid, residual_sugar, chlorides và alcohol. Những biến này phản ánh các nhóm thành phần quan trọng (axit, đường, muối và nồng độ cồn), có ảnh hưởng đáng kể đến cảm quan và chất lượng rượu. Các biến còn lại được thể hiện trong những phần trực quan khác như biểu đồ nhiệt (heatmap) và biểu đồ quan hệ (pairplot) để phân tích mối tương quan giữa chúng và biến quality

Nhiệm vụ 2: Trực quan hóa dữ liệu trong Matplotlib.

Trong quá trình phân tích dữ liệu thăm dò (Exploratory Data Analysis – EDA), nhóm đã trực quan hóa phân bố của tất cả các biến trong bộ dữ liệu Wine Quality thông qua các biểu đồ histogram. Mục tiêu là nhằm quan sát đặc điểm tổng thể, xu hướng và mức độ phân tán của từng biến, từ đó phát hiện các đặc điểm nổi bật và bất thường trong dữ liệu.

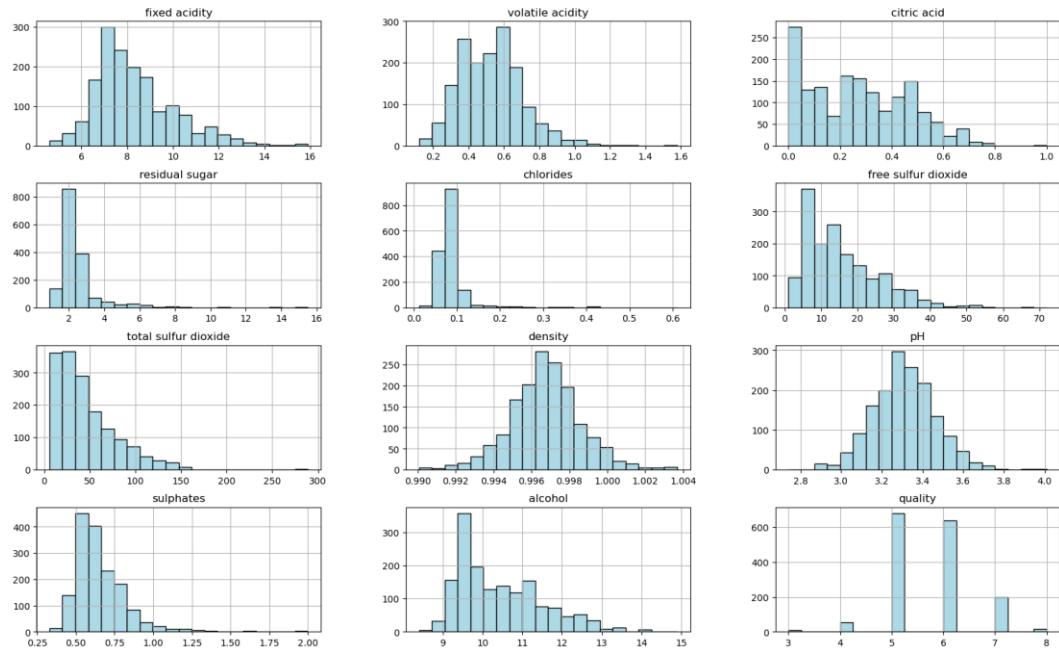
Kết quả trực quan cho thấy hầu hết các biến như fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, density, pH, sulphates và alcohol đều có phân bố lệch (skewed distribution) ở các mức độ khác nhau, không hoàn toàn tuân theo quy luật chuẩn.

Đặc biệt, biến alcohol có xu hướng phân bố lệch phải, cho thấy phần lớn rượu vang trong bộ dữ liệu có nồng độ còn trung bình, chỉ một số ít mẫu có độ còn cao hơn. Ngược lại, biến volatile acidity và chlorides có phân bố lệch trái, thể hiện phần lớn mẫu rượu có giá trị thấp ở các thông số này.

Bên cạnh đó, khi quan sát phân bố theo biến mục tiêu quality, ta nhận thấy chất lượng rượu chủ yếu nằm trong khoảng 5–6, tức là phần lớn các mẫu rượu có chất lượng trung bình. Điều này cho thấy dữ liệu có sự mất cân bằng nhẹ giữa các mức chất lượng, cần lưu ý khi xây dựng mô hình dự đoán sau này.

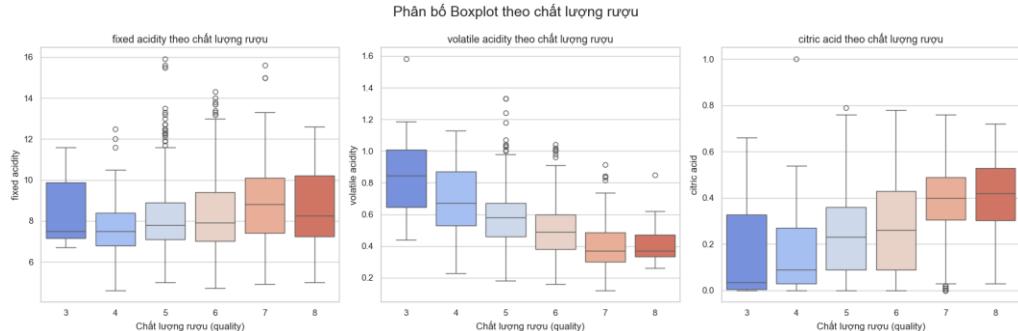
Nhìn chung, việc trực quan hóa giúp làm rõ đặc điểm của từng biến, hỗ trợ việc lựa chọn kỹ thuật tiền xử lý phù hợp như chuẩn hóa (normalization) hoặc xử lý lệch phân bố (log-transform), đồng thời định hướng cho các bước phân tích tương quan và mô hình hóa kế tiếp.

Phân bố tất cả các biến trong dataset rượu vang



Hình 2. 14 Phân bố tất cả các biến trong dataset rượu vang

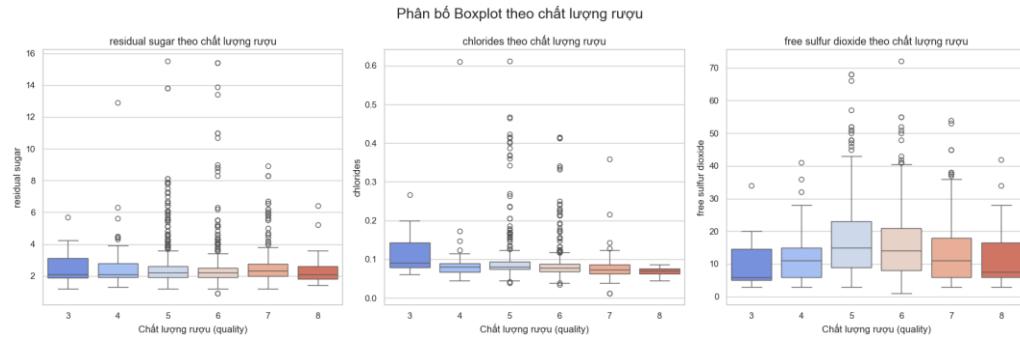
Trong phần này, nhóm sử dụng biểu đồ Boxplot để quan sát sự thay đổi của các đặc trưng hóa học theo các mức chất lượng rượu (quality). Mục tiêu là xác định những biến có mối quan hệ rõ rệt với chất lượng, từ đó rút ra đặc điểm nổi bật của rượu vang có chất lượng cao.



Hình 2. 15 Biểu đồ phân bố boxplot theo chất lượng rượu

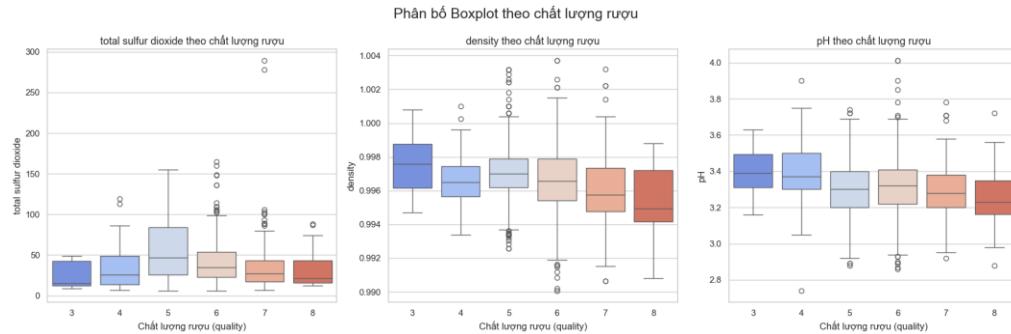
Biểu đồ cho thấy biến fixed acidity (độ axit cố định) có xu hướng tăng nhẹ ở các mẫu rượu có chất lượng cao hơn. Điều này cho thấy rượu ngon thường có độ axit tự nhiên cân bằng, góp phần tạo cảm giác tươi mát khi nếm. Ngược lại, volatile acidity (độ axit bay hơi) giảm rõ rệt khi chất lượng tăng. Rượu có độ axit bay hơi cao thường để lại vị chua gắt hoặc mùi giấm, do đó đây là đặc tính tiêu cực ảnh

hướng đến chất lượng. Citric acid (axit citric) tăng dần theo chất lượng. Rượu có nồng độ axit citric cao hơn thường mang vị chua dịu và tươi hơn, góp phần vào hương vị tổng thể cân đối và dễ chịu của rượu chất lượng tốt.



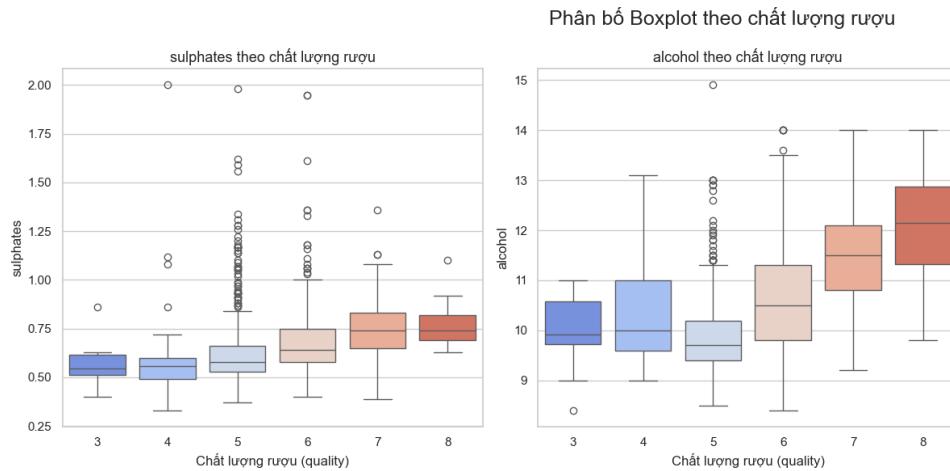
Hình 2. 16 Biểu đồ phân bố boxplot theo chất lượng rượu(1)

Residual sugar (đường dư) không thể hiện xu hướng rõ rệt theo chất lượng. Có thể thấy lượng đường dư trong rượu vang đỏ thường được kiểm soát ở mức ổn định và ít ảnh hưởng đến điểm đánh giá. Chlorides (hàm lượng muối) giảm dần khi chất lượng rượu tăng. Rượu có nhiều muối hơn thường bị đánh giá thấp do gây cảm giác gắt hoặc đắng. Đây là yếu tố phản ánh nguyên liệu hoặc quy trình sản xuất có thể ảnh hưởng đến vị tổng thể. Free sulfur dioxide (SO_2 tự do) có sự phân bố khá rộng ở mọi mức chất lượng, cho thấy biến này không phải yếu tố chính quyết định chất lượng rượu mà chủ yếu dùng để bảo quản.



Hình 2. 17 Biểu đồ phân bố boxplot theo chất lượng rượu(2)

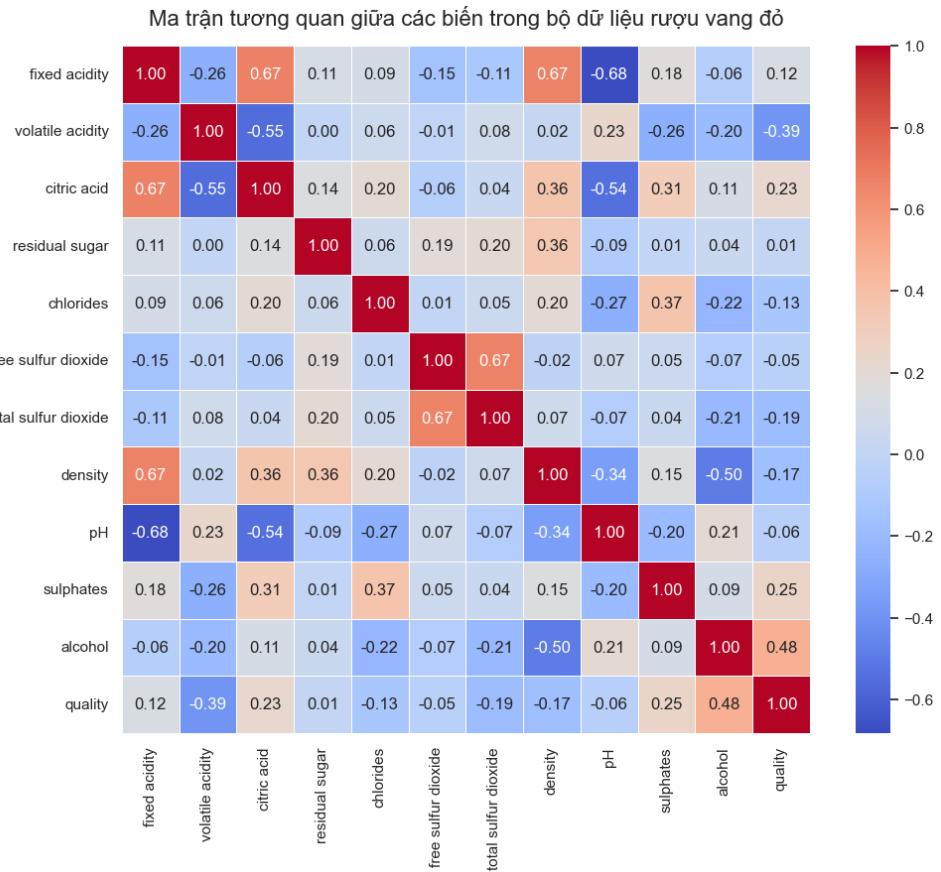
Total sulfur dioxide (tổng SO_2) thường thấp hơn ở các loại rượu chất lượng cao. Việc kiểm soát tốt lượng SO_2 giúp rượu giữ được hương tự nhiên và không gây vị cay hoặc hắc. Density (mật độ) giảm nhẹ khi chất lượng tăng. Rượu có chất lượng cao thường chứa ít đường dư và có cấu trúc cân bằng hơn, dẫn đến mật độ thấp hơn so với rượu kém chất lượng. pH biến thiên nhẹ, nhưng xu hướng chung là tăng nhẹ theo chất lượng. Rượu có pH cao hơn thường bớt chua và dễ uống hơn, tạo cảm giác dịu khi thưởng thức.



Hình 2. 18 Biểu đồ phân bố boxplot theo chất lượng rượu(3)

Sulphates (muối sunfat) có xu hướng tăng theo chất lượng. Chúng giúp ổn định màu sắc, cấu trúc và hương vị của rượu, nên nồng độ sulphates cao vừa phải là đặc điểm tích cực. Alcohol (nồng độ cồn) tăng rõ rệt theo chất lượng, là yếu tố quan trọng nhất trong việc đánh giá rượu. Rượu có độ cồn cao thường mang lại cảm giác đậm đà, mùi thơm mạnh và hậu vị kéo dài. Tổng thể, các biến sulphates và alcohol cho thấy ảnh hưởng tích cực đến chất lượng rượu. Rượu có nồng độ cồn cao, sulphates cân đối, và độ chua hài hòa thường được đánh giá là rượu ngon.

Biểu đồ ma trận tương quan dưới đây minh họa mối quan hệ giữa các biến hóa học trong bộ dữ liệu *Wine Quality - Red Wine*. Các ô màu thể hiện mức độ tương quan giữa các biến: màu đỏ biểu thị tương quan thuận mạnh, trong khi màu xanh thể hiện tương quan nghịch.



Hình 2. 19 Biểu đồ ma trận tương quan giữa các biến trong tập dữ liệu

Có thể thấy Alcohol (nồng độ cồn) có tương quan thuận mạn nhất với chất lượng rượu (quality) (~0.48). Điều này cho thấy rượu có nồng độ cồn cao thường được đánh giá là rượu ngon, có hương vị đậm đà hơn. Volatile acidity (độ axit bay hơi) có tương quan nghịch mạnh với chất lượng (~ 0.39). Nghĩa là rượu có độ axit bay hơi cao thường bị đánh giá kém hơn, do tạo vị chua gắt và mùi khó chịu. Citric acid có tương quan thuận nhẹ đến trung bình với chất lượng (~0.23). Rượu có nhiều citric acid thường tươi hơn và có hương vị dễ chịu. Density (mật độ) có tương quan nghịch đáng kể với chất lượng (~ -0.17). Rượu ngon thường có mật độ thấp hơn (do ít đường dư, cấu trúc cân bằng hơn). Sulphates cũng có tương quan thuận vừa phải với chất lượng (~0.25). Đây là chất giúp ổn định và tăng hương vị, nên rượu có sulphates hợp lý thường có chất lượng cao. Các biến như residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, pH hầu hết chỉ có tương quan yếu hoặc gần như không đáng kể với chất lượng rượu.

Mối quan hệ giữa các biến hóa học Fixed acidity, citric acid, và density có mối tương quan thuận với nhau, cho thấy những rượu có axit cao thường cũng có mật độ lớn hơn. Free sulfur dioxide và total sulfur dioxide có tương quan thuận rất

mạnh (~0.67), điều này hợp lý vì SO₂ tự do là một phần của SO₂ tổng. pH tương quan nghịch nhẹ với fixed acidity (~ -0.68), phản ánh đúng bản chất hóa học: độ pH càng thấp thì axit càng cao.

3. Bài toán 3

Thực hiện trực quan hóa dữ liệu trên tập dữ liệu về bệnh tiểu đường. Bệnh tiểu đường là một trong những bệnh mạn tính phổ biến và nguy hiểm, đặc biệt trong nhóm người có yếu tố nguy cơ như béo phì, tuổi cao, hoặc có tiền sử gia đình. Bộ dữ liệu Pima Indians Diabetes được thu thập từ phụ nữ Pima (Arizona, Mỹ) ≥ 21 tuổi, nhằm mục tiêu phân tích các yếu tố sức khỏe liên quan đến khả năng mắc bệnh tiểu đường.

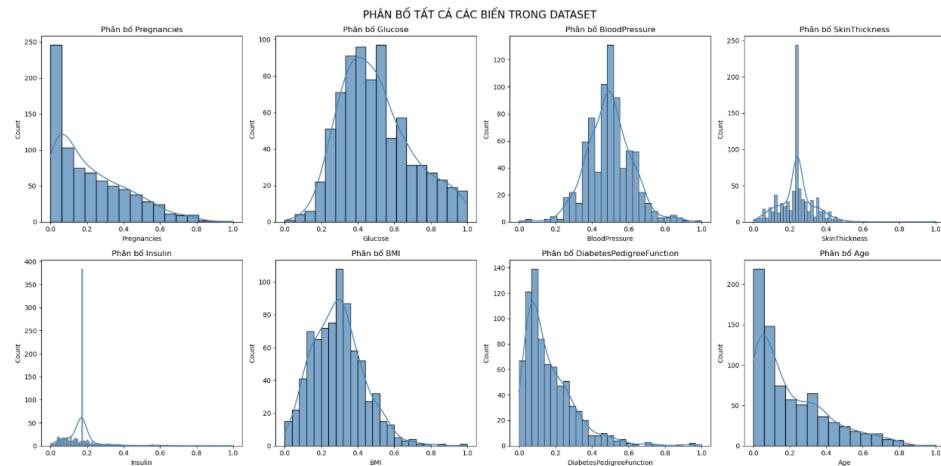
Thông tin cơ bản về dữ liệu bao gồm 768 mẫu, 8 thuộc tính đặc trưng và 1 biến mục tiêu. Với từng cột có các ý nghĩa như sau:

- Pregnancies: Số lần mang thai (kể cả những lần sảy thai hoặc thai chết lưu). Số nguyên (≥ 0)
- Glucose: Nồng độ glucose trong huyết tương sau khi xét nghiệm miệng dung nạp glucose trong 2 giờ (mg/dL). Giá trị 0 là bất thường và thường được coi là giá trị thiếu.
- BloodPressure: Huyết áp tâm trương (huyết áp dưới) tính bằng mmHg. Giá trị 0 là bất thường và thường được coi là giá trị thiếu.
- SkinThickness: Độ dày nếp gấp da cơ tam đầu(mm). Được sử dụng để ước tính tỷ lệ mỡ cơ thể. Giá trị 0 là bất thường.
- Insulin: Nồng độ insulin trong huyết thanh sau 2 giờ. Giá trị 0 là bất thường và thường được coi là giá trị thiếu.
- BMI: Chỉ số khối cơ thể (Body Mass Index)(kg/m²). Được tính bằng cân nặng chia cho bình phương chiều cao. Giá trị 0 là bất thường.
- DiabetesPedigreeFunction: Hàm phả hệ bệnh tiểu đường. Số thực (≥ 0) Đo lường nguy cơ di truyền mắc bệnh tiểu đường dựa trên tiền sử gia đình.
- Age: Tuổi của bệnh nhân. Số nguyên (năm)
- Outcome: Biến mục tiêu (biến kết quả). {0,1} 1: Mắc bệnh tiểu đường. 0: Không mắc bệnh tiểu đường.

Dữ liệu sau khi đã được tiền xử lý ở chương trước, chúng ta có thể sử dụng nó để tiếp tục trực quan hóa dữ liệu.

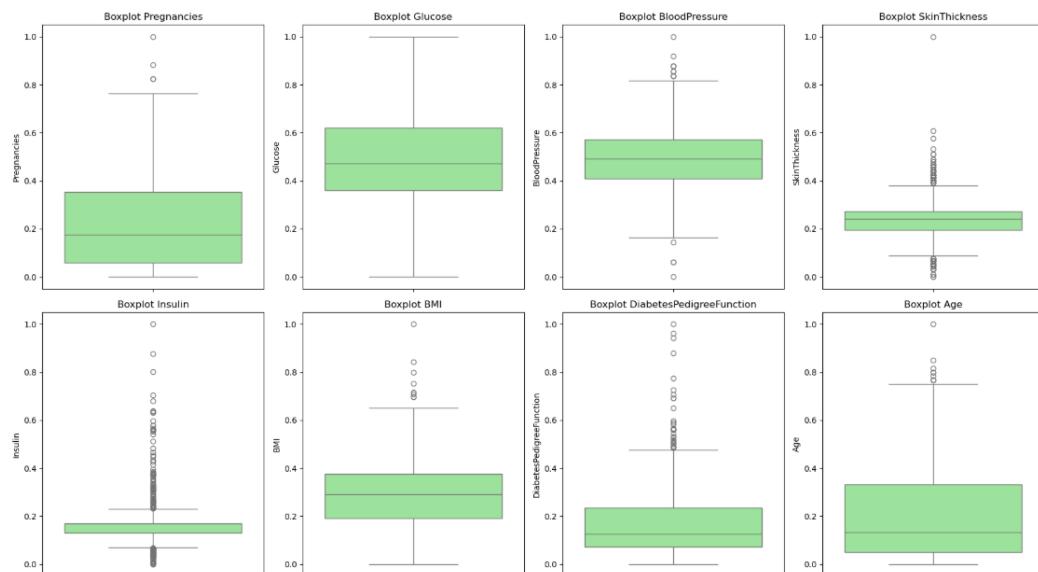
Biểu đồ thể hiện phân bố của 8 biến trong bộ dữ liệu, giúp quan sát đặc điểm và xu hướng của từng chỉ số sức khỏe. Glucose và BMI có phân bố gần chuẩn (tập

trung quanh giá trị trung bình). Insulin, Pregnancies, và Age lêch phai mạnh, thể hiện nhiều giá trị thấp và một số giá trị cao bất thường. Các biến như BloodPressure và SkinThickness tương đối tập trung quanh trung bình, cho thấy ít biến động.. Việc hiểu rõ đặc điểm phân bố giúp hỗ trợ quá trình chuẩn hóa và lựa chọn mô hình phù hợp cho bước phân tích tiếp theo.



Hình 2. 20 Biểu đồ phân bố các biến trong dataset

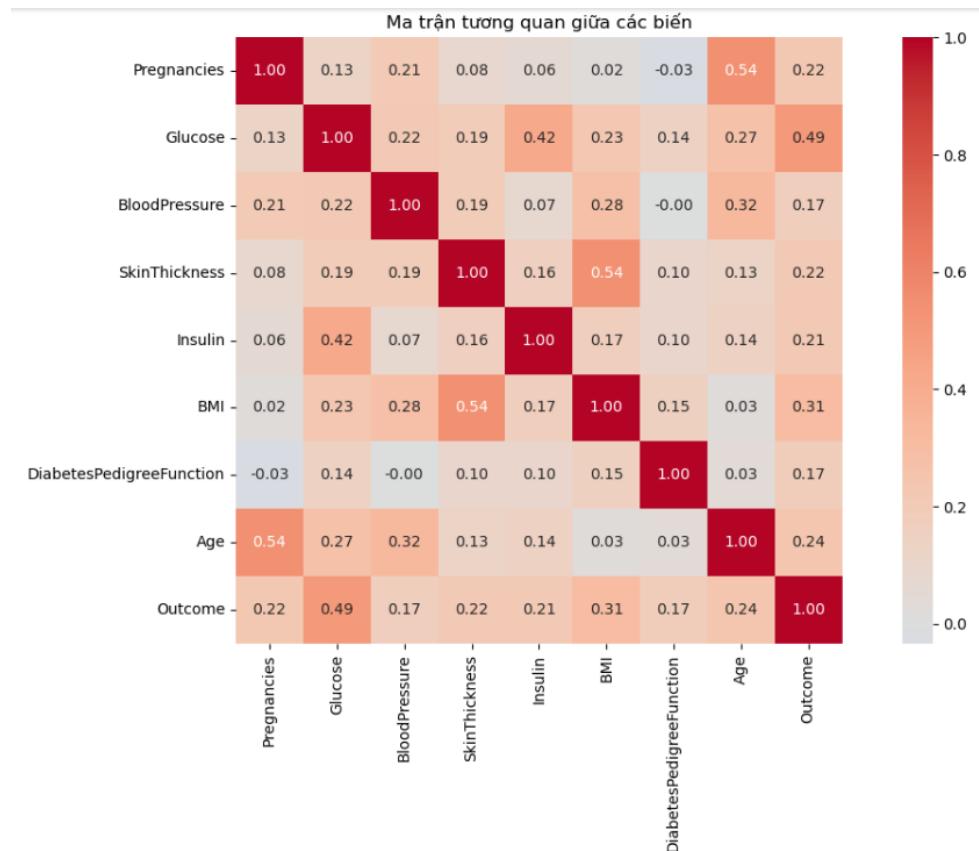
Biểu đồ boxplot cho thấy hầu hết các biến đều có một số giá trị ngoại lệ (outliers), đặc biệt rõ ở các biến Insulin, SkinThickness và DiabetesPedigreeFunction. Trong khi đó, các biến như Glucose và BloodPressure có phân bố tương đối ổn định, ít ngoại lệ hơn. Biểu đồ giúp phát hiện sớm các điểm bất thường để xử lý trước khi huấn luyện mô hình học máy.



Hình 2. 21 Biểu đồ boxplot biểu thị các giá trị trong dataset

Biểu đồ heatmap thể hiện ma trận tương quan giữa các biến định lượng trong bộ dữ liệu bệnh tiểu đường. Qua ma trận, có thể nhận thấy rằng biến Glucose (mức đường huyết) có mối tương quan mạnh nhất với biến mục tiêu Outcome (giá trị 0.49), cho thấy những người có nồng độ đường trong máu cao hơn thường có nguy cơ mắc tiểu đường cao hơn. Bên cạnh đó, các biến như BMI (chỉ số khối cơ thể) và Age (tuổi) cũng thể hiện mối tương quan dương với Outcome, phản ánh thực tế rằng người có chỉ số BMI lớn hoặc tuổi cao dễ bị ảnh hưởng bởi bệnh tiểu đường.

Ngoài ra, giữa các biến độc lập cũng tồn tại những mối tương quan đáng chú ý, chẳng hạn SkinThickness và BMI (0.54) hay Pregnancies và Age (0.54), cho thấy mối liên hệ hợp lý về mặt sinh lý - người lớn tuổi thường có số lần mang thai nhiều hơn, và chỉ số BMI thường tăng cùng độ dày da. Nhìn chung, ma trận tương quan giúp xác định được những đặc trưng có ảnh hưởng lớn nhất đến khả năng mắc bệnh, từ đó hỗ trợ bước chọn đặc trưng (feature selection) và xây dựng mô hình dự đoán hiệu quả hơn.



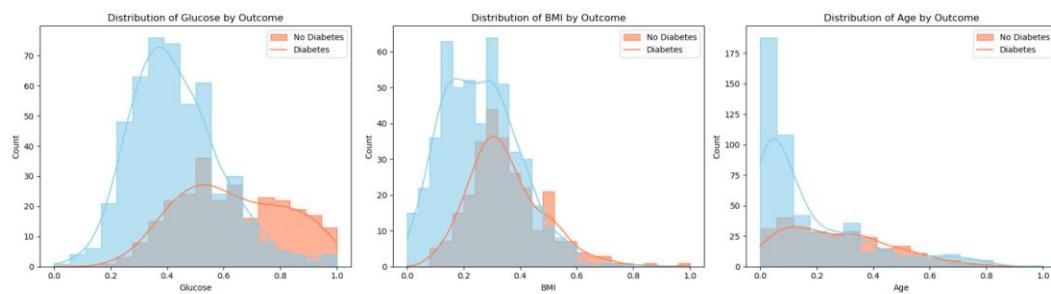
Hình 2. 22 Ma trận tương quan giữa các biến

Biểu đồ trên minh họa phân bố của ba biến Glucose, BMI và Age theo hai nhóm đối tượng: có tiểu đường và không tiểu đường. Quan sát cho thấy, những

người mắc tiểu đường thường có mức Glucose cao hơn rõ rệt, thể hiện sự khác biệt đáng kể về nồng độ đường huyết giữa hai nhóm.

Đối với biến BMI, nhóm người bị tiểu đường cũng có xu hướng có chỉ số khối cơ thể cao hơn, cho thấy mối liên hệ giữa thừa cân và nguy cơ mắc bệnh. Trong khi đó, phân bố của biến Age cho thấy nhóm người bị tiểu đường thường lớn tuổi hơn, phản ánh xu hướng tuổi tác là một yếu tố quan trọng.

Nhìn chung, cả ba đặc trưng này đều thể hiện sự khác biệt phân bố giữa hai nhóm Outcome, góp phần khẳng định vai trò của Glucose, BMI và Age là những biến quan trọng trong việc dự đoán khả năng mắc tiểu đường.



Hình 2.23 Biểu đồ thể hiện sự khác biệt phân bố giữa 2 nhóm

4. Bài toán 4

Thực hiện EDA trên tập dữ liệu mua sắm tại siêu thị. Đây là dữ liệu giao dịch bán lẻ trực tuyến, ghi lại các đơn hàng từ một cửa hàng bán đồ gia dụng, quà tặng, trang trí... trong khoảng thời gian 2010–2011.

Các biến đầu vào bao gồm:

- InvoiceNo: Mã hóa đơn (bắt đầu bằng "C" là đơn hủy)
- StockCode: Mã sản phẩm
- Description: Tên mô tả sản phẩm
- Quantity: Số lượng sản phẩm trong giao dịch (có thể âm nếu là hủy hàng)
- InvoiceDate: Thời gian giao dịch
- UnitPrice: Giá tiền trên một đơn vị sản phẩm
- CustomerID: Mã khách hàng (có giá trị null)
- Country: Quốc gia của khách hàng

Sử dụng hai lệnh `data.shape` và `data.info()` để xem quy mô và đặc điểm của bộ dữ liệu. `data.shape` cho biết số dòng (bản ghi) và số cột (thuộc tính) trong tập dữ liệu. `data.info()` cung cấp thông tin chi tiết hơn như tên cột, kiểu dữ liệu, số lượng

giá trị không bị thiếu, giúp đánh giá mức độ đầy đủ và tính phù hợp của dữ liệu trước khi xử lý.

```
data.shape
```

```
(541909, 8)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   InvoiceNo   541909 non-null   object 
 1   StockCode    541909 non-null   object 
 2   Description  540455 non-null   object 
 3   Quantity     541909 non-null   int64  
 4   InvoiceDate  541909 non-null   object 
 5   UnitPrice    541909 non-null   float64
 6   CustomerID   406829 non-null   float64
 7   Country      541909 non-null   object 
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

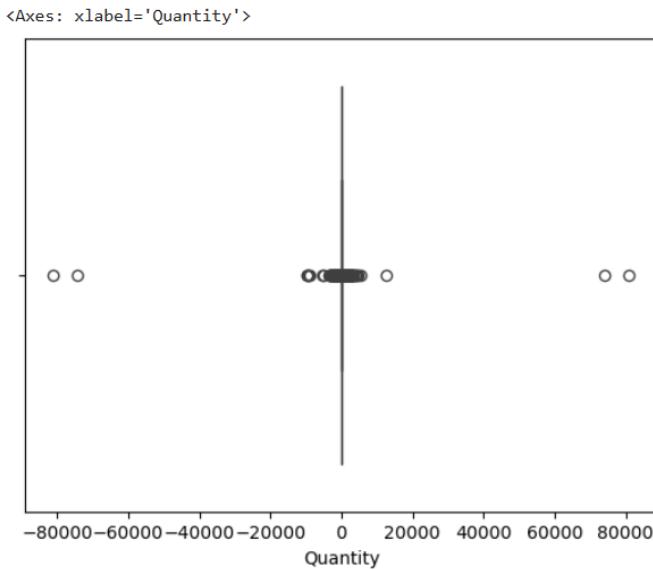
Hình 2. 24 Thông tin dữ liệu ban đầu của việc mua sắm tại siêu thị

Sử dụng lệnh `data.describe(include="all")` để hiển thị thống kê mô tả cho toàn bộ các cột trong tập dữ liệu, bao gồm cả biến định lượng (số) và biến định tính (chuỗi). Với biến số: giá trị trung bình, độ lệch chuẩn, giá trị nhỏ nhất, lớn nhất, tứ phân vị (25%, 50%, 75%). Với biến phân loại: số lượng giá trị hợp lệ, số giá trị duy nhất, giá trị xuất hiện nhiều nhất (mode) và tần suất của nó. Nhờ đó, ta có cái nhìn tổng quan về phân bố, phạm vi và đặc điểm dữ liệu trước khi tiến hành phân tích chuyên sâu.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
count	541909	541909		540455	541909.000000	541909	541909.000000	406829.000000
unique	25900	4070		4223	NaN	23260	NaN	NaN
top	573585	85123A	WHITE HANGING HEART T-LIGHT HOLDER	NaN	2011-10-31 14:41:00	NaN	NaN	United Kingdom
freq	1114	2313		2369	NaN	1114	NaN	NaN
mean	NaN	NaN		9.552250	NaN	4.611114	15287.690570	NaN
std	NaN	NaN		218.081158	NaN	96.759853	1713.600303	NaN
min	NaN	NaN		-80995.000000	NaN	-11062.060000	12346.000000	NaN
25%	NaN	NaN		1.000000	NaN	1.250000	13953.000000	NaN
50%	NaN	NaN		3.000000	NaN	2.080000	15152.000000	NaN
75%	NaN	NaN		10.000000	NaN	4.130000	16791.000000	NaN
max	NaN	NaN		80995.000000	NaN	38970.000000	18287.000000	NaN

Hình 2. 25 Thống kê mô tả dữ liệu

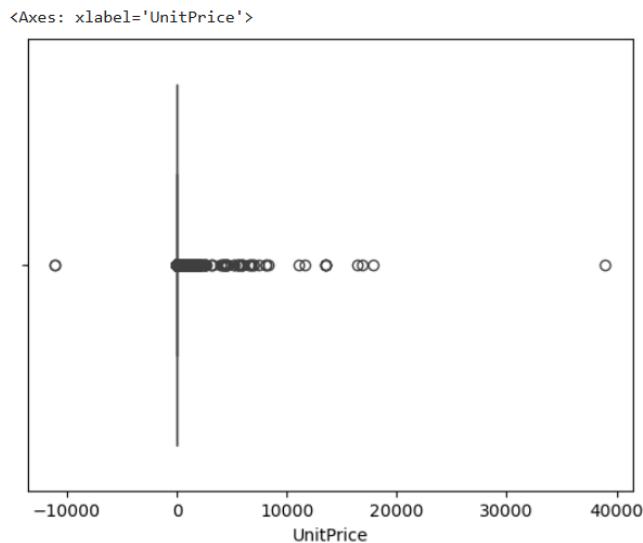
Sử dụng biểu đồ Boxplot để quan sát sự phân bố và phát hiện các giá trị ngoại lai của cột Quantity:



Hình 2. 26 Biểu đồ quan sát sự phân bố Quantity

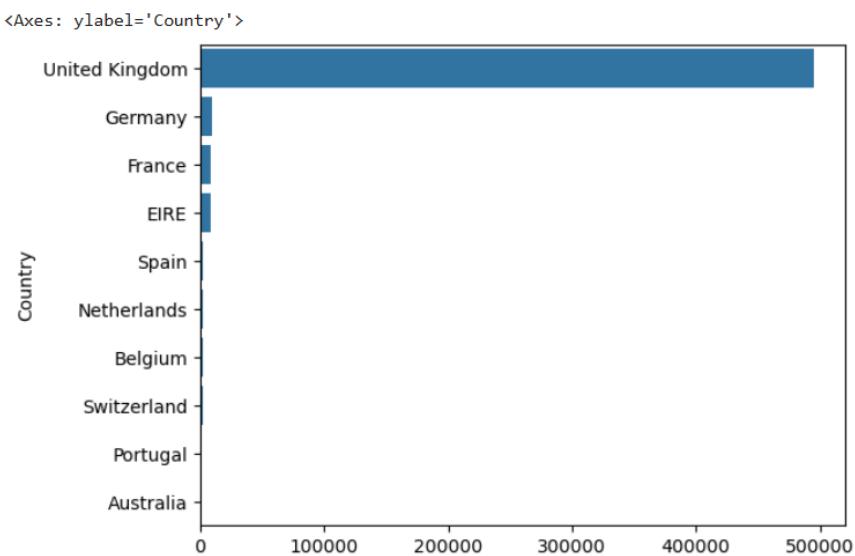
Biểu đồ Boxplot cho thấy Quantity có nhiều giá trị âm và ngoại lệ lớn, gây lệch phân phối. Cần loại bỏ các dòng có $Quantity \leq 0$ và xử lý outlier để đảm bảo dữ liệu phản ánh chính xác số lượng bán thực tế.

Đối với thuộc tính UnitPrice, biến này có phân phối lệch phải rõ rệt, xuất hiện một số giá trị âm và các giá trị cực cao bất thường. Do đó, cần loại bỏ hoặc điều chỉnh các giá trị âm và ngoại lai (outlier) trước khi tiếp tục tính toán thống kê hoặc phân tích doanh thu.



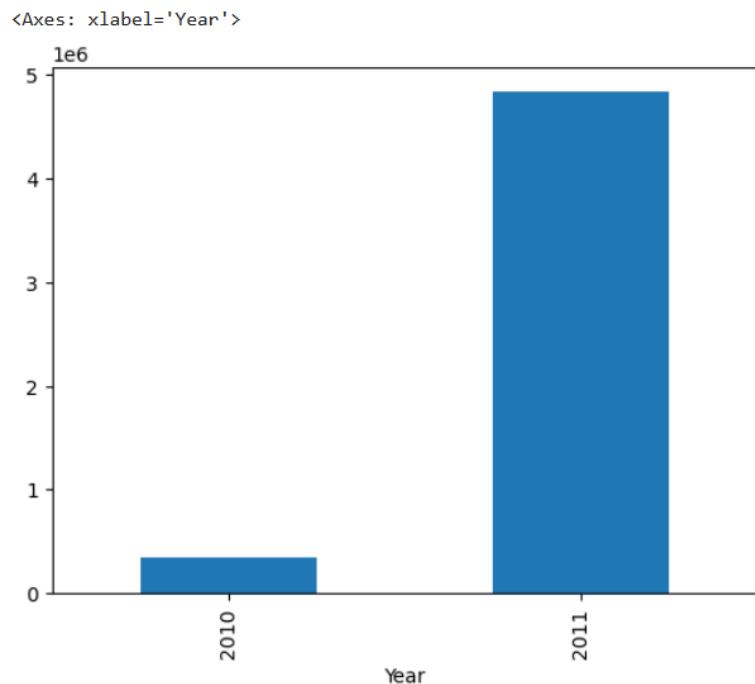
Hình 2. 27 Biểu đồ quan sát sự phân bố UnitPrice

Biểu đồ dưới đây thể hiện 10 quốc gia có số lượng giao dịch cao nhất. Có thể thấy United Kingdom chiếm ưu thế tuyệt đối, trong khi các quốc gia còn lại (như Germany, France, hay EIRE) chỉ chiếm tỷ lệ nhỏ trong tổng doanh số, cho thấy dữ liệu tập trung chủ yếu ở thị trường Anh.



Hình 2. 28 Biểu đồ thể hiện số lượng giao dịch cao nhất

Khi xem xét số lượng sản phẩm bán ra theo năm, ta nhận thấy sự chênh lệch lớn giữa hai năm 2010 và 2011. Dữ liệu năm 2010 chỉ bao gồm các giao dịch trong những tháng cuối năm, do đó không phản ánh đầy đủ hoạt động kinh doanh của giai đoạn này. Vì vậy, các phân tích doanh thu và hành vi khách hàng nên tập trung vào năm 2011 để đảm bảo độ chính xác.



Hình 2. 29 Biểu đồ số lượng sản phẩm bán ra theo năm

Trong quá trình phân tích dữ liệu bán hàng, việc tiền xử lý là rất quan trọng nhằm đảm bảo tính chính xác của các kết quả thống kê và mô hình dự đoán. Người ta sử dụng hàm `drop_duplicates()` để loại bỏ dữ liệu trùng lặp. Trong dữ liệu này có đến 5268 dòng bị trùng lặp.

Ta lại tiếp tục kiểm tra các dữ liệu thiếu thông qua hàm `isnull()`, có thể được biểu diễn trong hình dưới đây, có thể thấy rằng dữ liệu bị trống khá nhiều ở cột `Description` và `CustomerID`. `CustomerID` có giá trị `Nan` có thể là khách vãng lai, `Description` sẽ tìm tương ứng theo `StockCode` của nó.

```
print(df1.isnull().sum())
```

InvoiceNo	0
StockCode	0
Description	1454
Quantity	0
InvoiceDate	0
UnitPrice	0
CustomerID	135037
Country	0
Year	0

dtype: int64

Hình 2. 30 Kiểm tra dữ liệu thiếu

Đối với cột Description, có 1.454 giá trị bị thiếu. Để đảm bảo tính toàn vẹn của dữ liệu, tiến hành tạo một bảng ánh xạ (mapping) từ StockCode sang mô tả sản phẩm (Description) phổ biến nhất tương ứng. Cụ thể, sử dụng nhóm (groupby) theo StockCode và lấy mô tả xuất hiện nhiều nhất trong mỗi nhóm làm đại diện. Sau đó, các dòng có Description = NaN sẽ được điền lại dựa trên mapping này. Sau đó thêm cột Year và TotalAmount để hỗ trợ trực quan hóa.

	InvoiceNo	StockCode	Description	Quantity	\	
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6		
1	536365	71053	WHITE METAL LANTERN	6		
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8		
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6		
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6		
	InvoiceDate	UnitPrice	CustomerID	Country	Year	\
0	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	2010	
1	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	
2	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	2010	
3	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	
4	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	
	TotalAmount					
0	15.30					
1	20.34					
2	22.00					
3	20.34					
4	20.34					

Hình 2. 31 Mapping từ StockCode sang Description

Chia bộ dữ liệu ra làm 2 nhóm : Đơn bán (sales_df) và Đơn hoàn (returns_df) dựa trên Quantity và InvoiceNo. Ta sẽ thu được hình dưới đây:

Số đơn hoàn hàng: 10490										
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Year	TotalAmount
141	C536379	D	Discount	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom	2010	-27.50
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1	2010-12-01 09:49:00	4.65	15311.0	United Kingdom	2010	-4.65
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12	2010-12-01 10:24:00	1.65	17548.0	United Kingdom	2010	-19.80
236	C536391	21984	PACK OF 12 PINK PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom	2010	-6.96
237	C536391	21983	PACK OF 12 BLUE PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom	2010	-6.96

Hình 2. 32 Bộ dữ liệu đơn bán dựa trên Quantity và InvoiceNo

Số đơn hợp lệ: 526039										
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Year	TotalAmount
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	2010	15.30
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	2010	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010	20.34

Hình 2. 33 Bộ dữ liệu đơn bán hợp lệ dựa trên Quantity và InvoiceNo

Trong quá trình tiền xử lý dữ liệu, cột UnitPrice được kiểm tra và phát hiện có các giá trị âm và ngoại lệ quá lớn. Để đảm bảo tính chính xác của

phân tích, các giá trị UnitPrice ≤ 0 được loại bỏ, đồng thời chỉ giữ lại các giá trị nhỏ hơn phân vị 99% nhằm loại bỏ các outlier. Sau khi xử lý, dữ liệu giá trở nên hợp lý hơn, tránh sai lệch khi tính toán doanh thu hoặc thống kê.

```

q_hi_price = sales_df["UnitPrice"].quantile(0.99)
# Giữ lại các đơn hàng có UnitPrice <= ngưỡng này
sales_df = sales_df[(sales_df["UnitPrice"] > 0) & (sales_df["UnitPrice"] < q_hi_price)].copy()

print(q_hi_price)
print(df1["UnitPrice"].describe())

```

16.98	
count	536529.000000
mean	4.633623
std	97.243243
min	-11062.060000
25%	1.250000
50%	2.080000
75%	4.130000
max	38970.000000
Name:	UnitPrice, dtype: float64

Hình 2. 34 Thống kê mô tả dữ liệu của cột UnitPrice

Tương tự như với giá bán, cột Quantity (số lượng bán) cũng được kiểm tra và xử lý ngoại lệ. Các giá trị nhỏ hơn phân vị 10% và lớn hơn phân vị 99% được xem là bất thường nên bị loại bỏ. Việc này giúp loại trừ các giao dịch có số lượng quá nhỏ hoặc quá lớn, đảm bảo dữ liệu phân phối hợp lý và phản ánh đúng xu hướng mua hàng thực tế.

```

q_hi_qty   = sales_df["Quantity"].quantile(0.99)
q_low_qty  = sales_df["Quantity"].quantile(0.1)

q_high_qty = sales_df["Quantity"].quantile(0.99)  # ngưỡng trên 99%
sales_df = sales_df[(sales_df["Quantity"] >= q_low_qty) & (sales_df["Quantity"] <= q_high_qty)]

print(df1["Quantity"].describe())

```

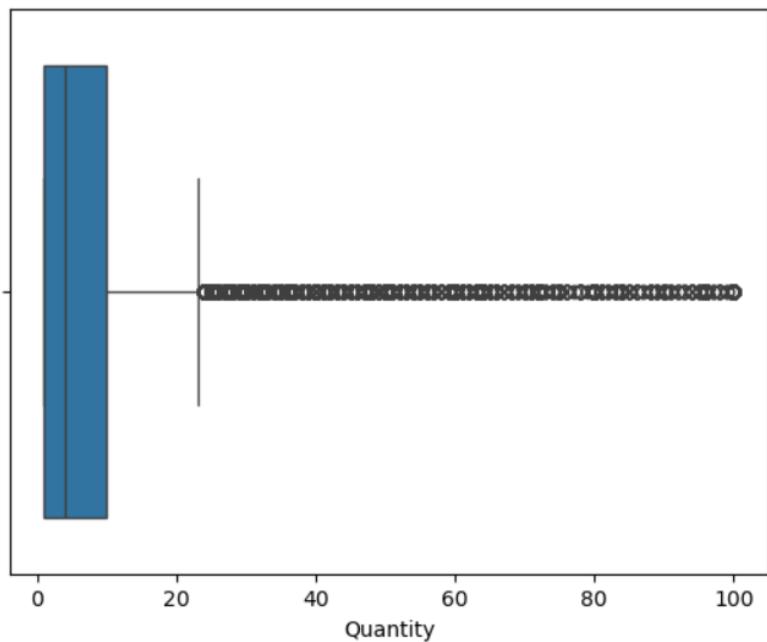
count	536529.000000
mean	9.623748
std	219.152755
min	-80995.000000
25%	1.000000
50%	3.000000
75%	10.000000
max	80995.000000
Name:	Quantity, dtype: float64

Hình 2. 35 Thống kê mô tả dữ liệu của cột Quantity

Biểu đồ hộp (Boxplot) trên thể hiện phân bố của hai biến Quantity và UnitPrice sau khi đã loại bỏ ngoại lệ. Có thể thấy rằng Quantity vẫn còn một số giá trị cao bất thường, tuy nhiên phần lớn dữ liệu tập trung ở khoảng nhỏ hơn 20 — phản ánh lượng mua phổ biến là ít sản phẩm. UnitPrice chủ yếu tập trung trong khoảng từ 1 đến 5, chỉ có một vài điểm giá cao hơn — thể hiện sản phẩm có giá trị lớn hoặc đặc biệt. Việc trực quan hóa này giúp đánh

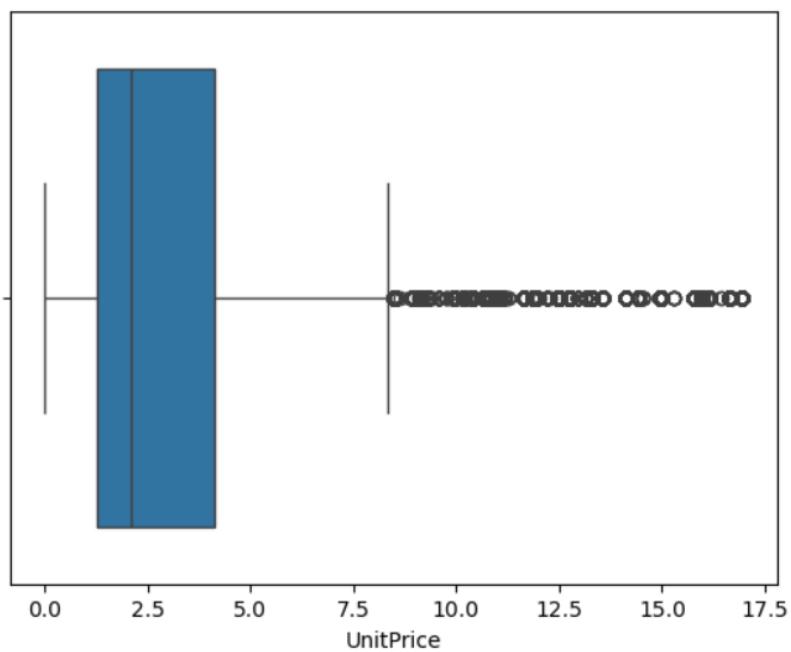
giá hiệu quả quá trình làm sạch dữ liệu và hiểu rõ hơn về xu hướng phân phối của các biến định lượng trong tập dữ liệu.

<Axes: xlabel='Quantity'>



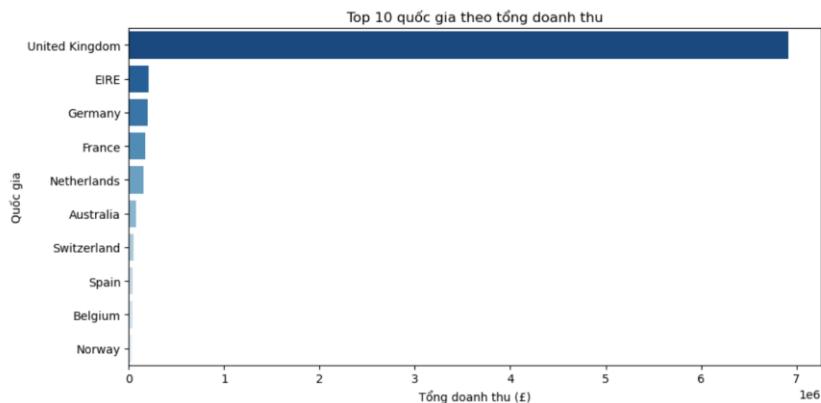
Hình 2. 36 Biểu đồ thể hiện sự phân bố của biến Quantity

<Axes: xlabel='UnitPrice'>



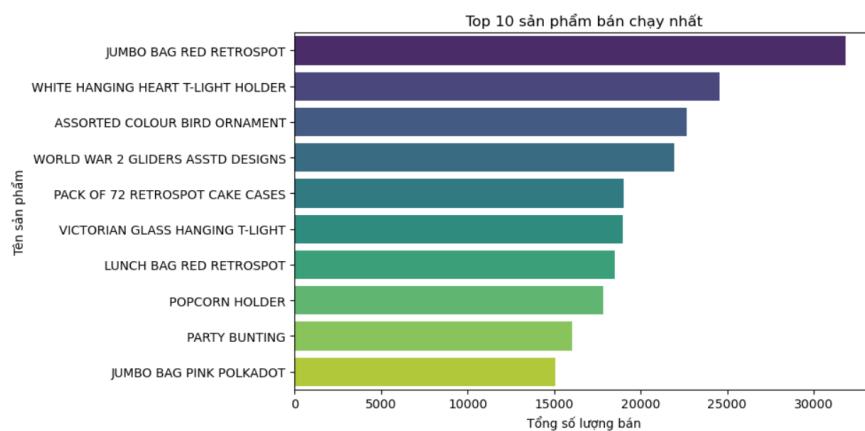
Hình 2. 37 Biểu đồ thể hiện sự phân bố của biến UnitPrice

Biểu đồ dưới đây hiện Top 10 quốc gia có tổng doanh thu cao nhất. Kết quả cho thấy United Kingdom chiếm ưu thế tuyệt đối về doanh thu, vượt xa các quốc gia còn lại — phản ánh đây là thị trường chính của doanh nghiệp. Các quốc gia khác như Netherlands, EIRE, Germany, France chỉ đóng góp một phần nhỏ, thể hiện hoạt động bán hàng quốc tế còn hạn chế. Nhìn chung, doanh thu tập trung mạnh vào thị trường nội địa, gợi ý cơ hội mở rộng kinh doanh ra các thị trường nước ngoài trong tương lai.



Hình 2. 38 Top 10 quốc gia theo tổng doanh thu

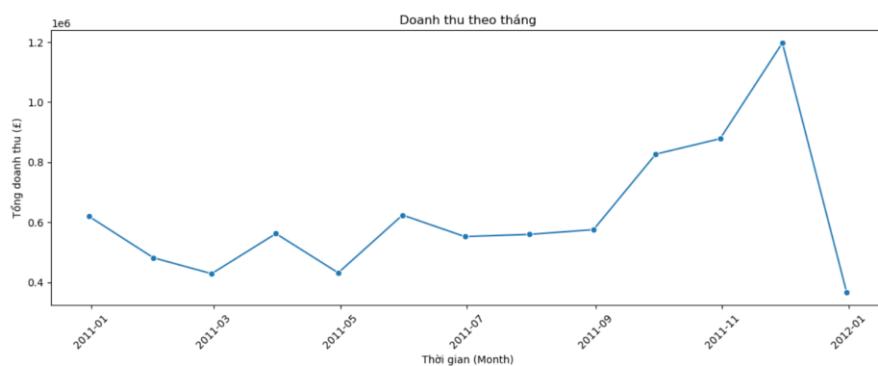
Top 10 sản phẩm bán chạy nhất dựa trên tổng số lượng bán ra cho thấy một số mặt hàng như Jumbo Bag Red Retrospot, White Hanging Heart T-Light Holder, Assorted Colour Bird Ornament chiếm số lượng bán vượt trội so với các sản phẩm khác. Điều này cho thấy những sản phẩm này có nhu cầu cao và tính phổ biến lớn trong danh mục hàng hóa, có thể được xem là nhóm sản phẩm chủ lực. Từ kết quả này, doanh nghiệp có thể ưu tiên nhập kho, quảng bá hoặc mở rộng danh mục liên quan để tối ưu doanh thu.



Hình 2. 39 Top 10 sản phẩm bán chạy nhất

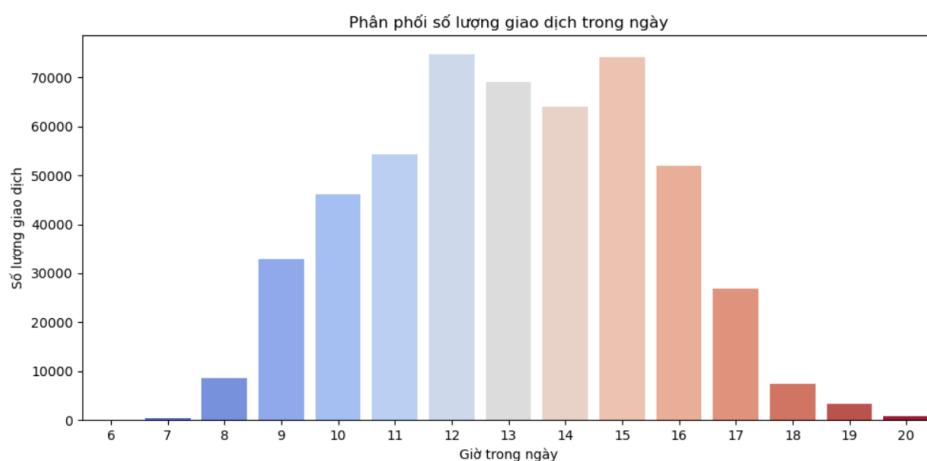
Biểu đồ doanh thu theo tháng cho thấy sự biến động doanh thu của doanh nghiệp trong từng giai đoạn. Nhìn chung, doanh thu có xu hướng tăng dần ở một số tháng nhất định, phản ánh các chu kỳ mua sắm hoặc mùa cao điểm tiêu dùng. Tuy nhiên, vẫn xuất hiện những giai đoạn sụt giảm nhẹ, cho thấy sự ảnh hưởng của yếu tố thời vụ hoặc nhu cầu thị trường.

Từ kết quả này, doanh nghiệp có thể lập kế hoạch nhập hàng và chiến lược marketing theo mùa, nhằm tối ưu hóa doanh thu và nguồn lực kinh doanh.



Hình 2. 40 Biểu đồ doanh thu theo tháng

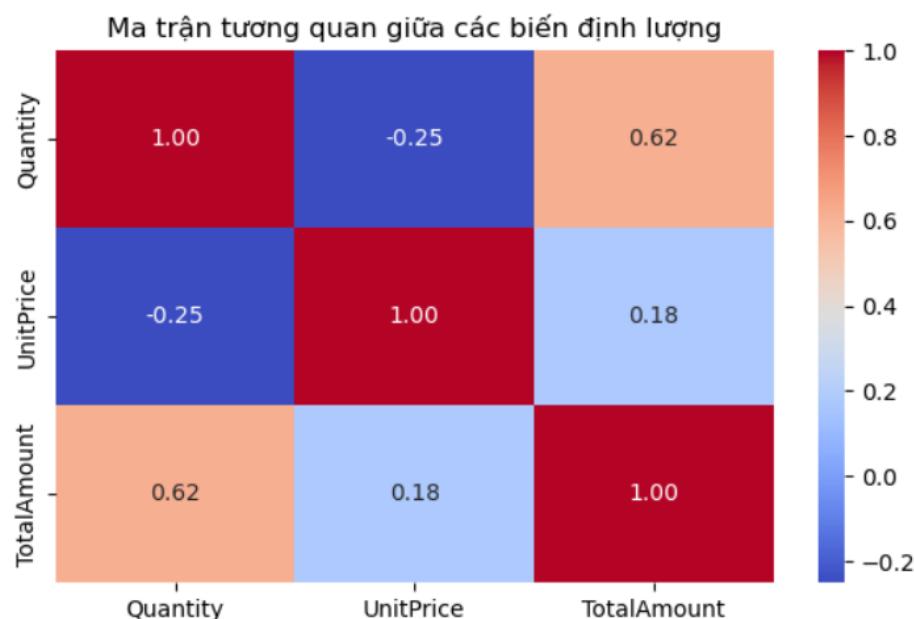
Biểu đồ phân phối số lượng giao dịch theo giờ trong ngày cho thấy phần lớn các giao dịch tập trung vào ban ngày, đặc biệt là từ khoảng 10h đến 16h – đây có thể là khung giờ làm việc chính của khách hàng hoặc thời điểm mua sắm trực tuyến cao điểm. Ngoài khung giờ này, lượng giao dịch giảm mạnh, cho thấy hoạt động kinh doanh ít sôi động hơn vào buổi tối và rạng sáng. Kết quả này giúp doanh nghiệp xác định thời điểm cao điểm để tối ưu nhân lực, hệ thống xử lý đơn hàng và các chiến dịch quảng cáo trong ngày.



Hình 2. 41 Biểu đồ phân phối số lượng giao dịch trong ngày

Phân tích ma trận tương quan giữa các biến định lượng cho thấy rằng tổng doanh thu (TotalAmount) có mối tương quan mạnh với số lượng bán ra (Quantity), thể hiện rằng số lượng sản phẩm tiêu thụ là yếu tố chính quyết định doanh thu, trong khi đơn giá (UnitPrice) chỉ có ảnh hưởng nhỏ hơn. Ngoài ra, có mối tương quan âm nhẹ giữa Quantity và UnitPrice, cho thấy các sản phẩm giá thấp thường được mua với số lượng lớn hơn.

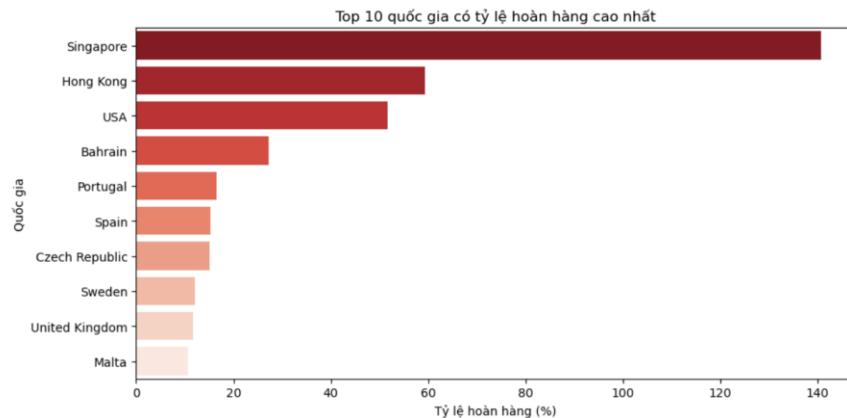
Nhìn chung, kết quả này giúp doanh nghiệp tập trung vào việc tăng khối lượng bán và đẩy mạnh các sản phẩm có giá hợp lý để tối ưu hóa doanh thu.



Hình 2. 42 Biểu đồ ma trận tương quan giữa các biến định lượng

Top 10 quốc gia có tỷ lệ hoàn hàng cao nhất cho thấy sự khác biệt rõ rệt giữa các thị trường. Một số quốc gia có tỷ lệ hoàn hàng vượt trội, phản ánh khả năng tồn tại các vấn đề về chất lượng sản phẩm, vận chuyển, hoặc sự khác biệt trong kỳ vọng của khách hàng.

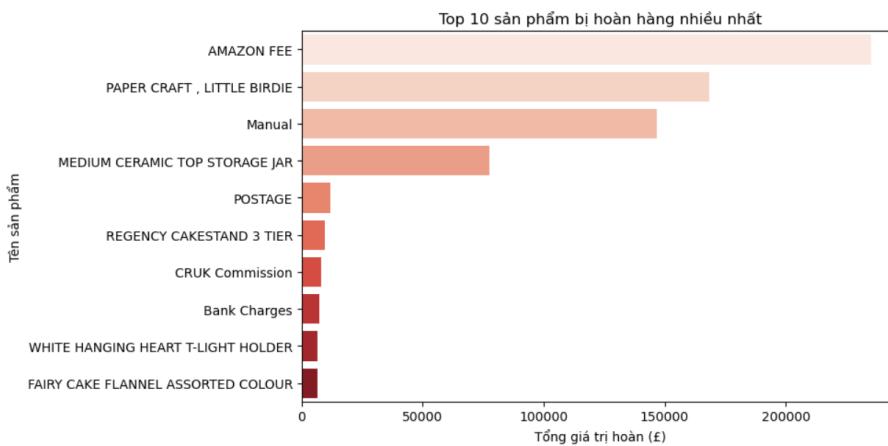
Tỷ lệ hoàn hàng cao là chỉ báo quan trọng về mức độ hài lòng của khách hàng, do đó doanh nghiệp cần xem xét lại quy trình đóng gói, kiểm soát chất lượng, và dịch vụ hậu mãi tại các quốc gia này nhằm giảm thiểu tổn thất doanh thu và nâng cao trải nghiệm khách hàng.



Hình 2. 43 Top 10 quốc gia có tỷ lệ hoàn hàng cao nhất

Biểu đồ dưới đây cho thấy một số mặt hàng chiếm tỷ trọng hoàn hàng đáng kể so với phần còn lại. Điều này có thể phản ánh vấn đề về chất lượng, đóng gói, hoặc kỳ vọng khách hàng không được đáp ứng.

Những sản phẩm này cần được kiểm tra kỹ hơn về quy trình sản xuất và vận chuyển, đồng thời doanh nghiệp có thể xem xét điều chỉnh chính sách đổi trả hoặc cải thiện mô tả sản phẩm để giảm thiểu rủi ro hoàn hàng và nâng cao sự hài lòng của khách hàng.



Hình 2. 44 Top 10 sản phẩm bị hoàn hàng nhiều nhất

CHƯƠNG 3: PHÂN TÍCH ĐƠN BIẾN VÀ HAI BIẾN

I. Lý thuyết Phân tích Đơn biến và Hai biến

1. Phân tích đơn biến

Định nghĩa: là kỹ thuật phân tích dữ tập trung vào việc nghiên cứu và diễn giải một biến duy nhất để mô tả đặc điểm, tìm kiếm mô hình và đo lường xu hướng trung tâm, độ phân tán của dữ liệu mà không xem xét mối quan hệ với các biến khác.

Mục đích: Hiểu rõ đặt điểm cơ bản của biến đó. Phát hiện các xu hướng hoặc mẫu hình trong dữ liệu của biến đó

Các kỹ thuật phổ biến:

+ **Thống kê tóm tắt:** Sử dụng các chỉ số như giá trị trung bình (mean), trung vị (median), giá trị lớn nhất (max), giá trị nhỏ nhất (min), độ lệch chuẩn (standard deviation) để tóm tắt dữ liệu.

+ **Biểu đồ:** Sử dụng biểu đồ cột, biểu đồ thanh, hoặc biểu đồ tần suất để trực quan hóa dữ liệu, giúp dễ dàng nhận thấy xu hướng.

2. Phân tích hai biến

Định nghĩa: Phân tích hai biến là một phương pháp thống kê nhằm kiểm tra và xác định mối quan hệ giữa hai biến riêng biệt, cho phép nhà nghiên cứu hiểu được liệu chúng có liên quan với nhau hay không, mức độ mạnh yếu của mối liên hệ, và liệu một biến có thể được sử dụng để dự đoán biến kia hay không.

Mục đích: Kiểm tra giả thuyết: Giúp xác định các giả thuyết đơn giản về mối liên hệ giữa hai yếu tố. Xác định mối quan hệ: Tìm ra liệu hai biến có tương quan (liên quan) với nhau hay không.

Đánh giá cường độ và hướng mối quan hệ: Nếu có mối quan hệ, phân tích này cho biết mối quan hệ đó mạnh hay yếu, và nó là mối quan hệ thuận hay nghịch chiều.

Vậy sự khác nhau chủ yếu là Phân tích đơn biệt mục đích để kiểm tra chính biến đó thay đổi như thế nào, Phân tích 2 biến nhằm mục đích kiểm tra mối quan hệ coi chúng có liên quan, ảnh hưởng với nhau không.

3. Thủ tục đo thống kê phân tích đơn biến

Thống kê mô tả trung tâm:

+ Trung bình (Mean): Là tổng tất cả các giá trị chia cho số lượng các giá trị. Đây là thước đo phổ biến nhất.

- Công dụng: Cho ta một con số đại diện cho toàn bộ tập dữ liệu.
- Nhược điểm: Rất nhạy cảm với các giá trị ngoại lai (outliers). Một giá trị quá lớn hoặc quá nhỏ có thể kéo giá trị trung bình lệch đi đáng kể.

+ Trung vị (Median): Là giá trị nằm chính giữa của tập dữ liệu sau khi đã được sắp xếp theo thứ tự từ nhỏ đến lớn.

- Công dụng: Là thước đo tốt hơn giá trị trung bình khi dữ liệu có outliers hoặc bị lệch (skewed).
- Cách tìm: Nếu số lượng giá trị là lẻ, trung vị là số ở giữa. Nếu là chẵn, trung vị là trung bình của hai số ở giữa.

+ Mode: Giá trị xuất hiện thường xuyên nhất trong tập dữ liệu.

- Công dụng: Hữu ích nhất cho dữ liệu phân loại (categorical data). Một tập dữ liệu có thể không có mode, có một mode (unimodal) hoặc nhiều mode (bimodal, multimodal).
- Các thước đo độ phân tán: Nhóm này cho chúng ta biết dữ liệu trải rộng hay co cụm quanh giá trị trung tâm như thế nào.

+ Khoảng giá trị (Range):

- Định nghĩa: Là hiệu số giữa giá trị lớn nhất (max) và giá trị nhỏ nhất (min).
- Công dụng: Cung cấp cái nhìn nhanh nhất về sự biến thiên của dữ liệu.
- Nhược điểm: Giống như giá trị trung bình, nó rất nhạy cảm với outliers vì chỉ phụ thuộc vào hai giá trị đầu và cuối.

+ Phương sai (Variance):

- Là trung bình của bình phương khoảng cách từ mỗi điểm dữ liệu đến giá trị trung bình.
- Công dụng: Đo lường mức độ phân tán của dữ liệu. Phương sai lớn có nghĩa là dữ liệu trải rộng, phương sai nhỏ có nghĩa là dữ liệu co cụm gần giá trị trung bình.
- Nhược điểm: Đơn vị của nó là bình phương đơn vị của dữ liệu gốc (ví dụ: nếu dữ liệu là mét, phương sai là mét vuông), gây khó khăn trong việc diễn giải.

+ Độ lệch chuẩn (Standard Deviation):

- Là căn bậc hai của phương sai.

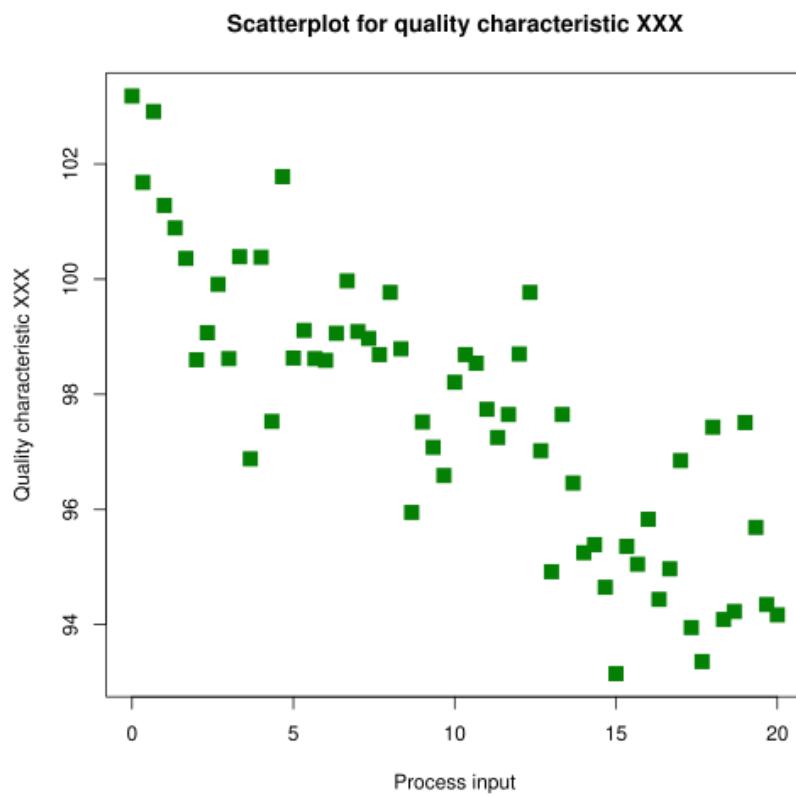
- Công dụng: Đây là thước đo độ phân tán phổ biến và quan trọng nhất. Nó cho biết trung bình mỗi điểm dữ liệu cách giá trị trung bình bao xa.
 - Ưu điểm: Có cùng đơn vị với dữ liệu gốc, giúp việc diễn giải trở nên trực quan và dễ hiểu hơn nhiều so với phương sai.
 - Ví dụ: Một độ lệch chuẩn nhỏ cho thấy các điểm dữ liệu có xu hướng rất gần với giá trị trung bình, trong khi độ lệch chuẩn lớn cho thấy các điểm dữ liệu trải rộng trên một phạm vi giá trị lớn hơn.
- + Khoảng tứ phân vị (Interquartile Range - IQR): Là khoảng cách giữa tứ phân vị thứ ba ($Q_3 - 75\%$) và tứ phân vị thứ nhất ($Q_1 - 25\%$) là "khoảng giá trị" của 50% dữ liệu ở giữa.
- Công dụng: Để đo độ phân tán khi dữ liệu có outliers hoặc bị lệch, vì nó không bị ảnh hưởng bởi 25% giá trị nhỏ nhất và 25% giá trị lớn nhất.

4. Xác định mối quan hệ trong phân tích hai biến

Phân tích hai biến là quá trình nghiên cứu mối quan hệ giữa hai biến (hay còn gọi là hai tập dữ liệu) để xem chúng có liên quan, ảnh hưởng, hoặc đồng biến với nhau hay không.

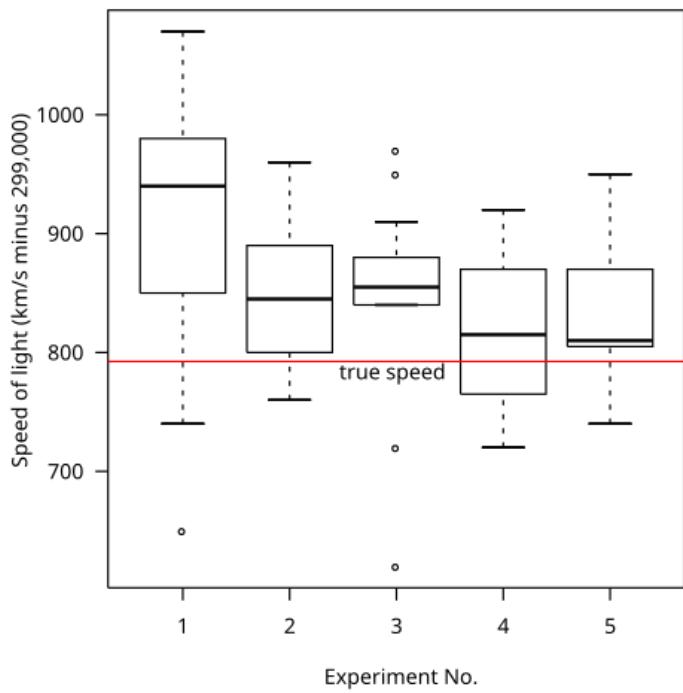
Xác định Mối quan hệ (Tương quan): tìm hiểu xem hai biến có "cùng nhau di chuyển" hay không. Tức là khi một biến thay đổi, biến kia có xu hướng thay đổi theo một quy luật nào đó không. Phương pháp sẽ phụ thuộc vào loại dữ liệu của hai biến mà ta xét.

- + Khi cả hai biến đều là biến số (Numerical)
 - Công cụ trực quan hóa: Biểu đồ phân tán (Scatter Plot) giúp ta có thể **nhìn thấy** hình dạng của mối quan-hệ.
 - **Tương quan dương:** Các điểm dữ liệu có xu hướng đi lên từ trái sang phải (khi biến X tăng, biến Y cũng có xu hướng tăng).
 - **Tương quan âm:** Các điểm dữ liệu có xu hướng đi xuống từ trái sang phải (khi biến X tăng, biến Y có xu hướng giảm).
 - **Không tương quan:** Các điểm dữ liệu phân tán ngẫu nhiên như một đám mây, không có quy luật rõ ràng.
 - **Quan hệ phi tuyến:** Các điểm có thể tạo thành một đường cong (hình chữ U, chữ J, v.v.), cho thấy có mối quan-hệ nhưng không phải là đường thẳng.



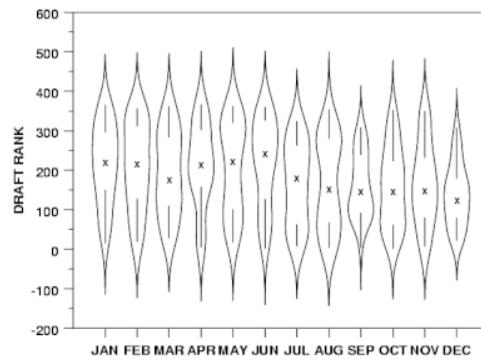
Hình 3. 1 Ví dụ về biểu đồ Scatter Plot

- Thước đo thống kê: Hệ số tương quan (Correlation Coefficient): Hệ số này định lượng độ mạnh và hướng của mối quan-hệ **tuyến tính**.
Hệ số tương quan Pearson (r): Phổ biến nhất. Giá trị của nó dao động từ -1 đến +1.
 - $r = +1$: Tương quan dương tuyến tính hoàn hảo.
 - $r = -1$: Tương quan âm tuyến tính hoàn hảo.
 - $r = 0$: Không có tương quan tuyến tính.
 Quy ước về độ mạnh:
 - $|r| > 0.7$: Tương quan mạnh.
 - $0.5 < |r| < 0.7$: Tương quan vừa phải.
 - $0.3 < |r| < 0.5$: Tương quan yếu.
 - $|r| < 0.3$: Tương quan rất yếu hoặc không đáng kể.
 + Khi một biến là Phân loại (Categorical) và một biến là Số (Numerical): Mục tiêu là so sánh giá trị của biến số qua các nhóm khác nhau của biến phân loại. Công cụ trực quan hóa:
 - **Biểu đồ hộp theo nhóm (Grouped Box Plots):** Rất hiệu quả để so sánh sự phân phối (trung vị, IQR, outliers) của biến số giữa các nhóm.



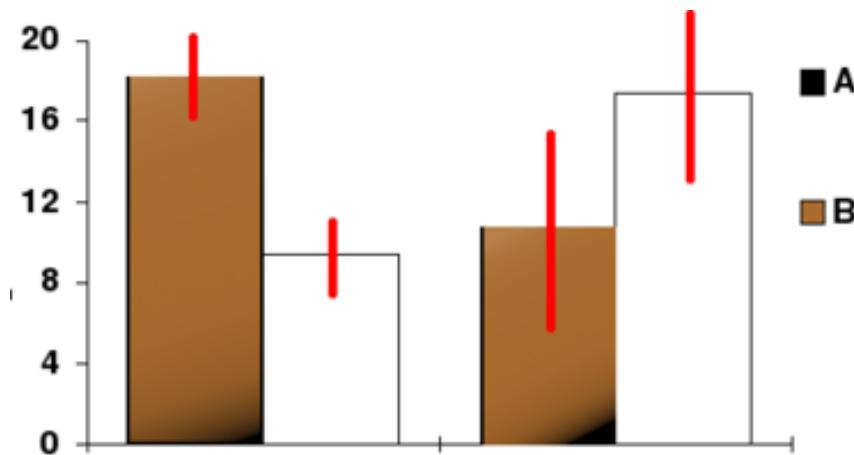
Hình 3. 2 Ví dụ biểu đồ BoxPlots

- **Biểu đồ Violin (Violin Plots):** Tương tự biểu đồ hộp nhưng thể hiện cả mật độ phân phối của dữ liệu.



Hình 3. 3 Ví dụ về Violin Plot

- **Biểu đồ cột với thanh lỗi (Bar chart with error bars):** So sánh giá trị trung bình và độ không chắc chắn (ví dụ: độ lệch chuẩn hoặc sai số chuẩn) của các nhóm.



Hình 3. 4 Bar Chart with error bars

Kiểm định thống kê:

- **T-test (2 nhóm):** Dùng để kiểm tra xem có sự khác biệt đáng kể về mặt thống kê giữa giá trị trung bình của hai nhóm hay không.
- **ANOVA (Analysis of Variance) (>2 nhóm):** Dùng khi có từ ba nhóm trở lên, để kiểm tra xem có ít nhất một nhóm có giá trị trung bình khác biệt so với các nhóm còn lại hay không.

+ Khi cả hai biến đều là biến Phân loại (Categorical): Mục tiêu là xem xét liệu có mối liên-kết (association) giữa các danh mục của hai biến này hay không.

- Phân biệt Tương quan và Nhân quả: "Tương quan không bao hàm nhân quả"

Ví dụ: Vào mùa hè, doanh số bán kem và số vụ đuối nước đều tăng lên. Chúng có tương quan dương rất mạnh. Nhưng điều này không có nghĩa là ăn kem gây ra đuối nước.

- **Vậy Tại sao tương quan không phải là nhân quả?**

+ **Biến ẩn/Biến nhiễu (Confounding Variable):** Thường có một biến thứ ba (C) tác động lên cả A và B.

Trong ví dụ trên, biến ẩn là **nhiệt độ thời tiết**. Thời tiết nóng (C) khiến người ta đi bơi nhiều hơn (dẫn đến tăng nguy cơ đuối nước - A) và cũng khiến người ta ăn kem nhiều hơn (B).

+ **Quan hệ ngược (Reverse Causality):** Có thể B gây ra A thay vì A gây ra B.

- + **Quan hệ ngẫu nhiên (Spurious Correlation):** Hai biến hoàn toàn không liên-quan nhưng lại cho thấy một mối tương quan mạnh chỉ do sự trùng hợp ngẫu nhiên trong dữ liệu.

5. Sự khác biệt giữa tương quan (correlation) và hiệp phương sai (covariance)

Tương quan (Correlation) chính là phiên bản được chuẩn hóa của Hiệp phương sai (Covariance).

a) Định nghĩa và Mục đích

- Hiệp phương sai (Covariance):

+ **Mục đích:** Đo lường **chiều hướng** của mối quan hệ tuyến tính giữa hai biến. Nó cho chúng ta biết liệu hai biến có xu hướng *cùng nhau thay đổi* hay không.

+ **Cách hoạt động:** Nó tính toán mức độ mà hai biến cùng nhau di chuyển ra khỏi giá trị trung bình của chúng.

- **Hiệp phương sai dương (> 0):** Khi biến X tăng, biến Y cũng có xu hướng tăng.
- **Hiệp phương sai âm (< 0):** Khi biến X tăng, biến Y có xu hướng giảm.
- **Hiệp phương sai bằng 0:** Không có mối quan hệ tuyến tính nào giữa hai biến.

- Tương quan (Correlation):

+ **Mục đích:** Đo lường cả **chiều hướng và độ mạnh** của mối quan hệ tuyến tính giữa hai biến.

+ **Cách hoạt động:** Về bản chất, tương quan là hiệp phương sai được "chuẩn hóa" bằng cách chia cho tích độ lệch chuẩn của hai biến.

$$\text{Correlation}(X, Y) = \text{Covariance}(X, Y) / \text{Standard Deviation}(X, Y)$$

b) Thang đo và Đơn vị

- Hiệp phương sai:

+ **Khoảng giá trị:** Không giới hạn, có thể chạy từ $-\infty$ đến $+\infty$.

+ **Đơn vị:** Bị phụ thuộc vào đơn vị của hai biến. Ví dụ: tính hiệp phương sai giữa chiều cao (mét) và cân nặng (kg), đơn vị của hiệp phương sai sẽ là "mét-kg".

+ **Hạn chế:** Vì giá trị của nó không được chuẩn hóa, nên không thể so sánh trực tiếp hiệp phương sai giữa các cặp biến khác nhau. Một hiệp phương sai là 2000 không nhất thiết chỉ ra mối quan hệ mạnh hơn một hiệp phương sai là 0.5 nếu chúng được tính từ các bộ dữ liệu với đơn vị và thang đo khác nhau.

- Tương quan:

+ **Khoảng giá trị:** Luôn luôn nằm trong khoảng từ **-1** đến **+1**.

+ **Đơn vị:** Không có đơn vị (dimensionless).

+ **Ưu điểm:** Chính vì được chuẩn hóa và không có đơn vị, hệ số tương quan cho phép chúng ta:

- **Đánh giá độ mạnh** của mối quan hệ một cách khách quan.
- **So sánh độ mạnh** của các mối quan hệ giữa các cặp biến khác nhau (ví dụ: so sánh mối tương quan giữa "chiều cao-cân nặng" với "thu nhập-chi tiêu").

c) Kết luận

- Hiệp phương sai:

+ Chỉ có thể diễn giải **dấu** của nó (dương, âm, hoặc bằng 0) để biết chiều hướng của mối quan hệ.

+ Và **Không thể** dựa vào độ lớn của hiệp phương sai để kết luận về độ mạnh của mối quan hệ.

- Tương quan có thể diễn giải cả **dấu và độ lớn** của nó:

+ **Dấu:** Cho biết chiều hướng (dương là cùng chiều, âm là ngược chiều).

+ **Độ lớn (giá trị tuyệt đối):** Cho biết độ mạnh của mối quan hệ tuyến tính.

- Gần 1 (hoặc -1): Mối quan hệ tuyến tính rất mạnh.
- Gần 0: Mối quan hệ tuyến tính rất yếu hoặc không tồn tại.

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

- **Phân tích đơn biến:** Sử dụng khi muốn có một cái nhìn sơ bộ về nó như sự biến động, hình dạng, trung tâm, phân tán.
- Sử dụng khi:

+ Muốn hiểu phân phối của một biến

+ Tìm kiếm outliers

+ Kiểm tra tính chuẩn của dữ liệu

+ So sánh các nhóm khác nhau của cùng một biến

- **Phân tích hai biến:** Sử dụng trực quan hóa trong phân tích hai biến khi bạn muốn hiểu mối quan hệ, tương quan, và cách một biến thay đổi khi biến kia thay đổi.
- Sử dụng khi:

+ Muốn khám phá mối quan hệ giữa hai biến

+ So sánh các nhóm khác nhau

+ Tìm kiếm patterns hoặc clusters

+ Chuẩn bị cho modeling

7. Mẫu code ví dụ về tạo biểu đồ scatter plot

Biểu đồ này là công cụ chính để phân tích mối quan hệ giữa **hai biến định lượng** (Quantitative Variables), giúp thấy rõ **tương quan** (correlation).

```

[1] # Khai báo thư viện
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

[2]

[3] # Chuẩn bị dữ liệu mẫu
data = {
    'Kich_thuoc_m2': [
        60, 70, 85, 90, 95, 100, 110, 120, 130,
        140, 150, 160, 170, 180, 190, 200, 220, 250, 300
    ],
    'Gia_ban_nghin_USD': [
        230, 250, 280, 310, 320, 330, 350, 360, 400, 420,
        440, 480, 500, 530, 560, 590, 620, 700, 820, 1000
    ]
}
df = pd.DataFrame(data)

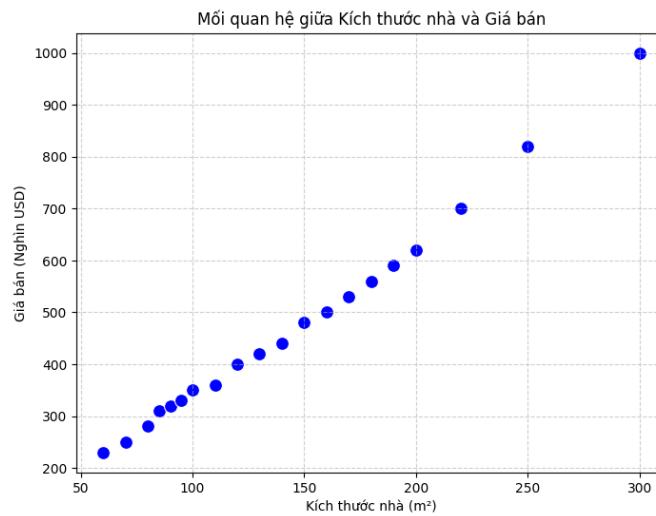
[4]

[5] # Tạo Scatter Plot
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Kich_thuoc_m2', y='Gia_ban_nghin_USD', data=df, s=100, color='blue')
plt.title('Mối quan hệ giữa Kích thước nhà và Giá bán')
plt.xlabel('Kích thước nhà (m2)')
plt.ylabel('Giá bán (Nghìn USD)')
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()

```

Hình 3. 5 Code để tạo biểu đồ

- Kết quả:



Hình 3. 6 Kết quả khi chạy code tạo biểu đồ

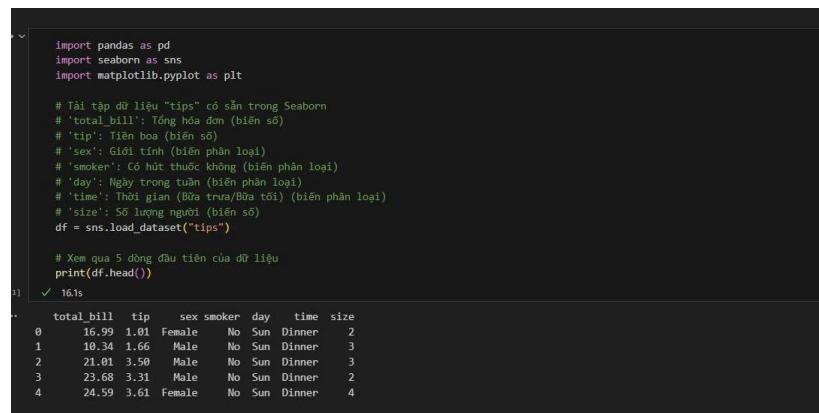
- + Biểu đồ sẽ hiển thị các điểm gần như nằm trên một đường thẳng có độ dốc dương.
- + Điều này cho thấy có một mối tương quan dương mạnh giữa hai biến: Kích thước nhà càng lớn, giá bán có xu hướng càng cao (đồng biến).

8. Cách trực quan hóa mối quan hệ giữa một biến số và một biến phân loại bằng biểu đồ boxplot

- Biến định lượng: là biến giá trị, thể hiện bằng những con số
 - Biến phân loại: ví dụ: giới tính, khu vực, loại sản phẩm
 - Ở đây chúng ta sẽ sử dụng biểu đồ hộp để trực quan hóa mối quan hệ này:
- + Giả sử chúng ta có một tập dữ liệu về các bữa ăn tại một nhà hàng (tips dataset, có sẵn trong Seaborn) và chúng ta muốn tìm hiểu mối quan hệ giữa ngày trong tuần (biến phân loại) và tổng hóa đơn (biến số).

Bước 1: Chuẩn bị môi trường và dữ liệu

Cài đặt thư viện cần thiết `pip install pandas seaborn matplotlib`



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

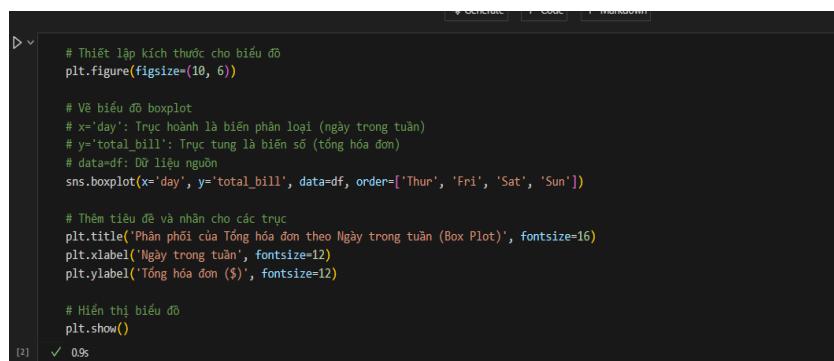
# Tải tập dữ liệu "tips" có sẵn trong Seaborn
# 'total_bill': Tổng hóa đơn (biến số)
# 'tip': Tiền tip (biến số)
# 'sex': Giới tính (biến phân loại)
# 'smoker': Có hút thuốc không (biến phân loại)
# 'day': Ngày trong tuần (biến phân loại)
# 'time': Thời gian (Bữa trưa/Bữa tối) (biến phân loại)
# 'size': Số lượng người (biến số)
df = sns.load_dataset("tips")

# Xem qua 5 dòng đầu tiên của dữ liệu
print(df.head())
[1]    ✓ 16.15
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	18.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Hình 3. 7 Code khai báo tập dữ liệu

Bước 2: Trực quan hóa bằng Biểu đồ Hộp (Box Plot)



```
# Thiết lập kích thước cho biểu đồ
plt.figure(figsize=(10, 6))

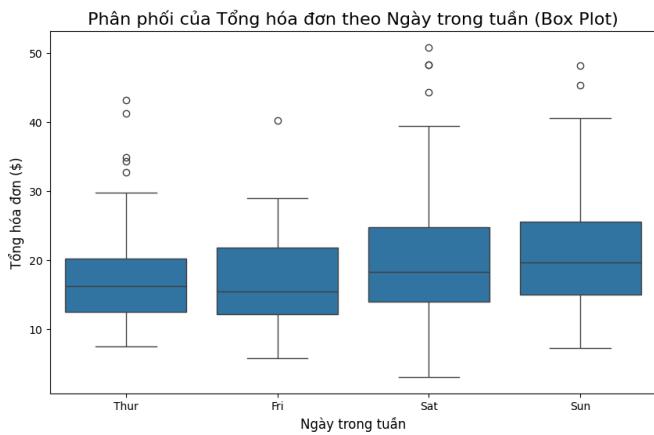
# Vẽ biểu đồ boxplot
# x='day': Trục hoành là biến phân loại (ngày trong tuần)
# y='total_bill': Trục tung là biến số (tổng hóa đơn)
# data=df: Dữ liệu nguồn
sns.boxplot(x='day', y='total_bill', data=df, order=['Thur', 'Fri', 'Sat', 'Sun'])

# Thêm tiêu đề và nhãn cho các trục
plt.title('Phân phối của Tổng hóa đơn theo Ngày trong tuần (Box Plot)', fontsize=16)
plt.xlabel('Ngày trong tuần', fontsize=12)
plt.ylabel('Tổng hóa đơn ($)', fontsize=12)

# Hiển thị biểu đồ
plt.show()
[2]    ✓ 0.9s
```

Hình 3. 8 Code tạo biểu đồ

- Kết quả:



Hình 3. 9 Biểu đồ hộp

- Diễn giải kết quả từ Box Plot:
 - + **So sánh Trung vị (Median):** Đường kẻ ngang bên trong mỗi hộp là giá trị trung vị. Nhìn vào biểu đồ, chúng ta có thể thấy trung vị của tổng hóa đơn vào cuối tuần (Thứ 7 - Sat, Chủ Nhật - Sun) cao hơn so với các ngày trong tuần (Thứ 5 - Thur, Thứ 6 - Fri).
 - + **So sánh Độ phân tán (Spread):** Chiều dài của hộp (khoảng từ phân vị - IQR) cho biết mức độ phân tán của 50% dữ liệu ở giữa. Hộp của ngày Thứ 7 và Chủ Nhật dài hơn, cho thấy tổng hóa đơn vào những ngày này có sự biến động lớn hơn.
 - + **Phát hiện Outliers:** Các chấm tròn nằm ngoài "râu" của biểu đồ là các giá trị ngoại lai. Chúng ta thấy rằng tất cả các ngày đều có một vài hóa đơn cao bất thường.

Bước 3: Trực quan hóa bằng Biểu đồ Violin (Violin Plot) với cùng tập dataset ở trên: Biểu đồ violin kết hợp ưu điểm của biểu đồ hộp và biểu đồ mật độ (density plot). Ngoài việc hiển thị các thông số như trung vị, IQR, nó còn cho thấy hình dạng phân phối đầy đủ của dữ liệu.

```

# Thiết lập kích thước cho biểu đồ
plt.figure(figsize=(10, 6))

# Vẽ biểu đồ violin plot
# Các tham số tương tự như boxplot
sns.violinplot(x='day', y='total_bill', data=df, order=['Thur', 'Fri', 'Sat', 'Sun'])

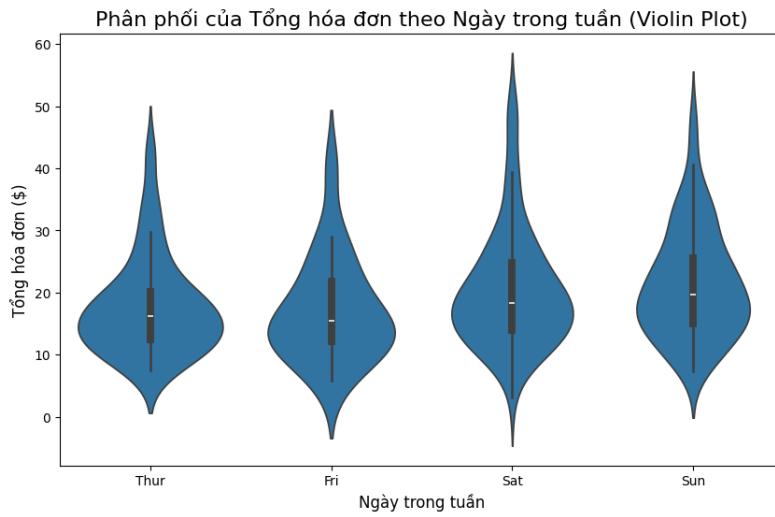
# Thêm tiêu đề và nhãn cho các trục
plt.title('Phân phối của Tổng hóa đơn theo Ngày trong tuần (Violin Plot)', fontsize=16)
plt.xlabel('Ngày trong tuần', fontsize=12)
plt.ylabel('Tổng hóa đơn ($)', fontsize=12)

# Hiển thị biểu đồ
plt.show()

```

Hình 3. 10 Code để tạo biểu đồ Violin

+ Kết quả:



Hình 3. 11 Biểu đồ Violin

- Diễn giải kết quả từ Violin Plot:

- + **Hình dạng Phân phối:** "Thân" của violin cho thấy mật độ dữ liệu. Phần thân càng rộng, mật độ dữ liệu tại giá trị đó càng cao. Ví dụ, vào Thứ 5 (Thur), có một sự tập trung lớn các hóa đơn quanh khoảng 12-18\$.
- + **So sánh với Box Plot:** Biểu đồ violin cung cấp nhiều thông tin hơn về hình dạng. Chúng ta có thể thấy rằng vào Thứ 7 (Sat), dữ liệu có vẻ tập trung ở nhiều mức giá khác nhau, trong khi vào Thứ 5 thì tập trung rõ rệt hơn ở một khoảng hẹp.
- + **Các thành phần bên trong:** Dấu chấm trắng là trung vị. Thanh màu đen dày là khoảng từ phân vị (IQR). Đường kẻ đen mỏng là khoảng tin cậy 95%.

II. Làm quen với các hàm và thư viện hỗ trợ phân tích dữ liệu đơn biến.

1. Bài toán 1

Bài toán này được thực hiện trên 2 tập dữ liệu là tập dữ liệu về chim cánh cụt và tập dữ liệu giá nhà.

Nhiệm vụ 1: phân tích dữ liệu đơn biến trên dữ liệu về chim cánh cụt

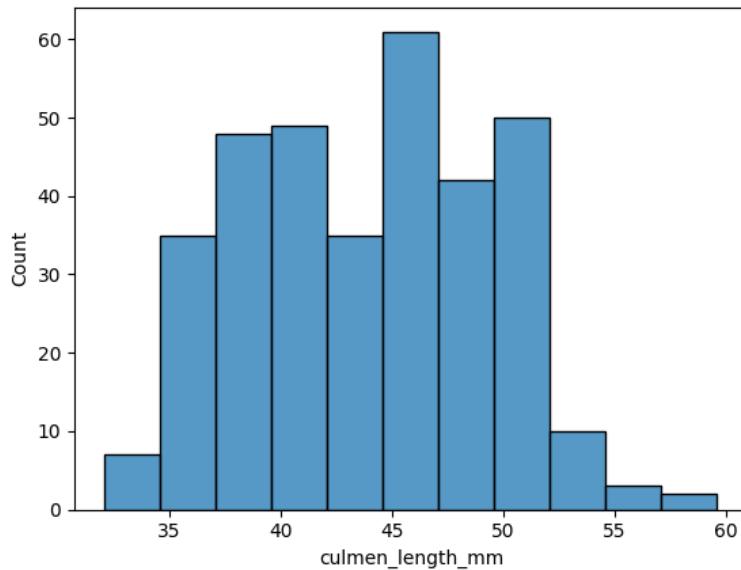
- Phân tích đơn biến bằng Histogram

```
#1. Import thư viện và nạp dữ liệu
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
penguins_data = pd.read_csv("data/penguins_size.csv")
penguins_data = penguins_data[['species','culmen_length_mm']]

#2. Phân tích đơn biến bằng Histogram
# case 1: basic
sns.histplot( data = penguins_data, x= penguins_data["culmen_length_mm"])
# case 2: advanced
plt.figure(figsize= (12,6))
ax = sns.histplot( data = penguins_data, x=
penguins_data["culmen_length_mm"])

ax.set_xlabel('Culmen Length in mm',fontsize = 15)
ax.set_ylabel('Count of records', fontsize = 15)
ax.set_title('Univariate analysis of Culmen Length',fontsize= 20)
```

Hình 3. 12 Import thư viện và tạo biểu đồ



Hình 3. 13 Biểu đồ Histogram của cột culmen_length_mm

- Phân bố của culmen_length_mm có dạng xấp xỉ chuẩn (bell-shaped).
- Giá trị phổ biến nhất (mode) rơi vào khoảng 40–50 mm — đây là chiều dài mỏ trung bình của đa số chim.

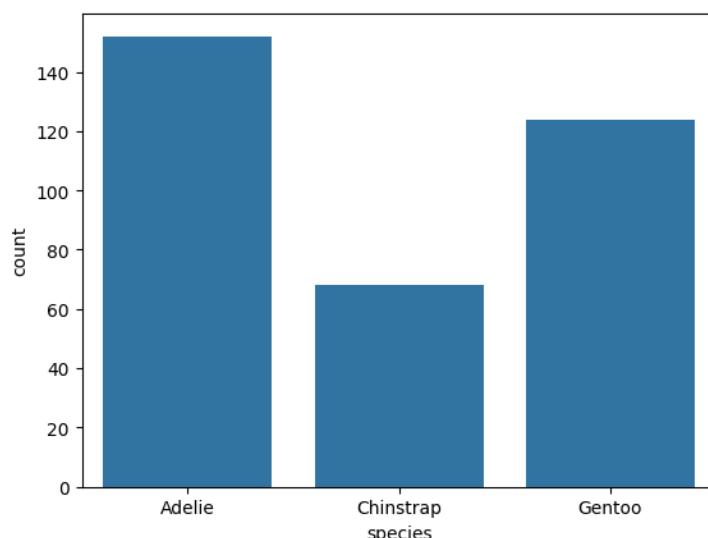
- Phạm vi dữ liệu dao động từ khoảng 30 mm đến 60 mm, cho thấy sự đa dạng giữa các loài (Adelie, Chinstrap, Gentoo).
- Có thể xuất hiện nhiều đỉnh (multi-modal distribution). Ví dụ:
 - + Một đỉnh quanh 38 mm (loài Adelie).
 - + Một đỉnh khác quanh 48–55 mm (loài Gentoo hoặc Chinstrap).

➔ Điều này cho thấy biến culmen_length_mm có thể dùng để phân biệt loài chim

- Phân tích đơn biến bằng Bar chart

```
#3. Phân tích đơn biến bằng bar chart
# case 1: basic
sns.countplot(data = penguins_data, x= penguins_data['species'])
# case 2: advanced
plt.figure(figsize= (12,6))
ax = sns.countplot(data = penguins_data, x= penguins_data['species'])
ax.set_xlabel('Penguin Species', fontsize = 15)
ax.set_ylabel('Count of records', fontsize = 15)
ax.set_title('Univariate analysis of Penguin Species', fontsize= 20)
ax.set_title('Univariate analysis of Culmen Length', fontsize= 20)
```

Hình 3. 14 Code tạo biểu đồ



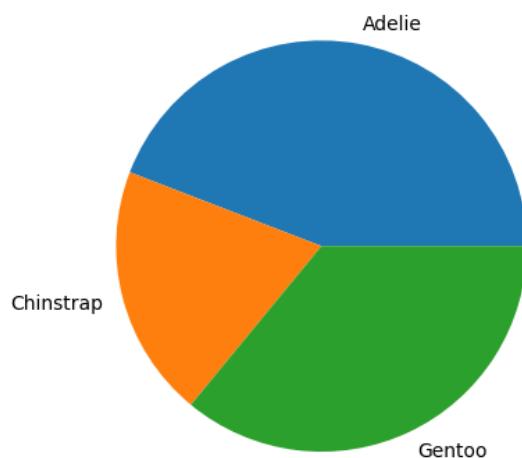
Hình 3. 15 Biểu đồ Bar chart

- + Loài chim Adelie: Khoảng 150 mẫu, Là loài phổ biến nhất trong dataset.
- + Loài chim Chinstrap: Khoảng 68 mẫu, Ít hơn Adelie, nhưng đủ đại diện.
- + Loài chim Gentoo: Khoảng 124 mẫu, Đứng thứ hai về số lượng.

- Phân tích đơn biến bằng biểu đồ tròn (Pie-chart)

```
#4. Phân tích đơn biến bằng biểu đồ tròn (Pie-chart)
penguins_group = penguins_data.groupby('species').count()
penguins_group = penguins_group.reset_index()
# case 1: basic
plt.pie(penguins_group["culmen_length_mm"], labels =
penguins_group['species'])
plt.show()
# case 2: advanced
cols = ['g', 'b', 'r']
plt.pie(penguins_group["culmen_length_mm"], labels =
penguins_group['species'], colors = cols)
plt.title('Univariate Analysis of Species', fontsize=15)
plt.show()
```

Hình 3. 16 Code tạo biểu đồ



Hình 3. 17 Biểu đồ Pie-Chart

- + Loài chim Adelie: Khoảng 150 mẫu, ~ 44%
- + Loài chim Chinstrap: Khoảng 68 mẫu, ~ 20%
- + Loài chim Gentoo: Khoảng 124 mẫu, ~ 36%

Nhiệm vụ 2: Phân tích dữ liệu đơn biến trên dữ liệu giá nhà

- Import thư viện, nạp dữ liệu giá nhà và phân tích đơn biến dựa vào boxplot

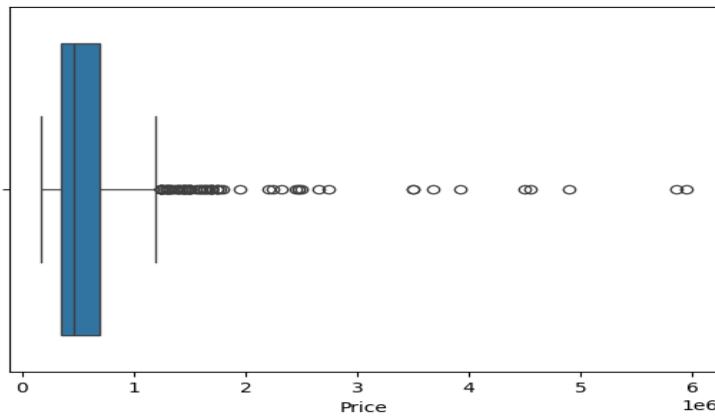
```

#1. Import thư viện, nạp dữ liệu giá nhà và phân tích đơn biến dựa vào boxplot
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
houseprices_data = pd.read_csv("data/HousingPrices-Amsterdam-August-2021.csv")
houseprices_data = houseprices_data[['Zip', 'Price', 'Area', 'Room']]

# case 1: Create a boxplot using the boxplot method
sns.boxplot(data = houseprices_data, x= houseprices_data["Price"])
# case 2: advanced
plt.figure(figsize= (12,6))
ax = sns.boxplot(data = houseprices_data, x= houseprices_data["Price"])
ax.set_xlabel('House Prices in millions', fontsize = 15)
ax.set_title('Univariate analysis of House Prices', fontsize= 20)
plt.ticklabel_format(style='plain', axis='x')

```

Hình 3. 18 Import thư viên và tạo biểu đồ



Hình 3. 19 Biểu đồ Box

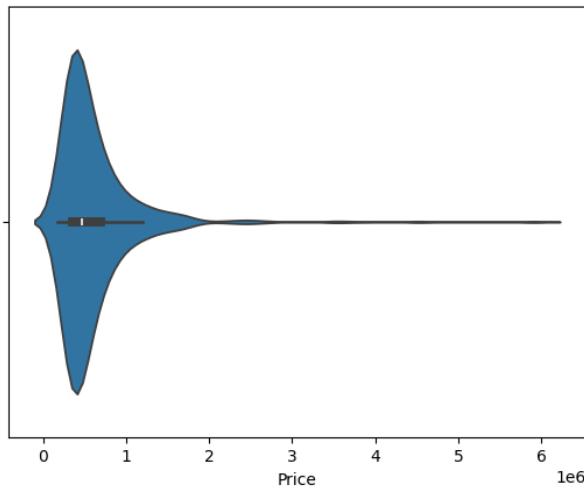
- Phân tích dữ liệu đơn biến dựa vào violin plot

```

#2. Phân tích dữ liệu đơn biến dựa vào violin plot
# case 1: basic
sns.violinplot(data = houseprices_data, x= houseprices_data["Price"])
# case 2: advanced
plt.figure(figsize= (12,6))
ax = sns.violinplot(data = houseprices_data, x=houseprices_data["Price"])
ax.set_xlabel('House Prices in millions', fontsize = 15)
ax.set_title('Univariate analysis of House Prices', fontsize= 20)
plt.ticklabel_format(style='plain', axis='x')

```

Hình 3. 20 Code tạo biểu đồ Violin



Hình 3. 21 Biểu đồ Violin

- Phân tích dữ liệu đơn biến dựa vào bản tóm tắt dữ liệu

	Price	Area	Room
count	9.200000e+02	924.000000	924.000000
mean	6.220654e+05	95.952381	3.571429
std	5.389942e+05	57.447436	1.592332
min	1.750000e+05	21.000000	1.000000
25%	3.500000e+05	60.750000	3.000000
50%	4.670000e+05	83.000000	3.000000
75%	7.000000e+05	113.000000	4.000000
max	5.950000e+06	623.000000	14.000000

Hình 3. 22 Bảng thống kê mô tả

- Nhận xét:
 - a. Biến Price (Giá nhà)
- + Biến Price đại diện cho giá bán của từng căn nhà trong tập dữ liệu.
- + Giá nhà trung bình là 622,065, trong khi giá trung vị chỉ đạt 467,000, cho thấy phân phối dữ liệu bị lệch phải (right-skewed) — tức là tồn tại một số căn nhà có giá rất cao làm kéo trung bình lên.
- + Khoảng giá dao động lớn (từ 175,000 đến 5,950,000) chứng tỏ dữ liệu bao gồm nhiều phân khúc thị trường từ nhà bình dân đến cao cấp.

+ Giá trị lớn nhất (5,950,000) có thể được xem là outlier, đại diện cho nhà biệt thự hoặc bất động sản đặc biệt.

➔ Kết luận: Dữ liệu giá nhà có sự phân tán cao, lệch phai mạnh, nên cần xem xét chuẩn hóa hoặc log-transform trước khi xây dựng mô hình dự đoán.

b. Biến Area (Diện tích nhà)

+ Biến Area biểu thị diện tích của ngôi nhà (đơn vị m^2).

Giá trị trung bình đạt $95.95 m^2$, với độ lệch chuẩn 57.45 , thể hiện mức biến động lớn giữa các loại hình nhà ở.

+ Phần lớn nhà có diện tích từ $60-120 m^2$, phù hợp với nhóm nhà phố hoặc căn hộ trung bình.

+ Tuy nhiên, diện tích tối đa lên tới $623 m^2$, cho thấy có sự xuất hiện của các căn biệt thự cao cấp — đây cũng là các outlier tiềm năng cần lưu ý trong quá trình phân tích.

➔ Kết luận: Biến Area có phân phối lệch phai nhẹ, tập trung chủ yếu quanh $80-100 m^2$, nhưng tồn tại một số mẫu diện tích lớn cần được xử lý.

c. Biến Room (Số lượng phòng)

+ Biến Room phản ánh số phòng trong mỗi căn nhà.

+ Trung bình, mỗi căn có 3.57 phòng, độ lệch chuẩn 1.59 cho thấy mức phân tán tương đối thấp — đa số nhà tập trung quanh $3-4$ phòng.

+ Giá trị nhỏ nhất là 1 phòng (studio hoặc căn hộ nhỏ), trong khi giá trị lớn nhất là 14 phòng, có thể là biệt thự hoặc nhà nhiều tầng.

➔ Kết luận: Phân phối của biến Room gần chuẩn (near-normal), không có sự lệch rõ rệt, phù hợp để sử dụng làm biến dự đoán trong mô hình hồi quy.

d. Nhận xét tổng thể

+ Price và Area có tương quan dương tiềm năng — nhà diện tích lớn thường có giá cao.

+ Room là biến rời rạc, nhưng phản ánh khá tốt quy mô nhà ở.

+ Tồn tại một số outliers ở biến Price và Area cần xử lý trong các bước tiền xử lý dữ liệu.

+ Có thể áp dụng các kỹ thuật như log-transformation, standardization hoặc winsorization để cải thiện tính ổn định cho mô hình học máy.

2. Bài toán 2:

Thực hiện các nhiệm vụ trong bài toán 2 để làm quen với việc phân tích hai biến với các hàm trong thư viện scikit-learn.

Nhiệm vụ 1: phân tích dữ liệu hai biến trên dữ liệu về chim cánh cụt.

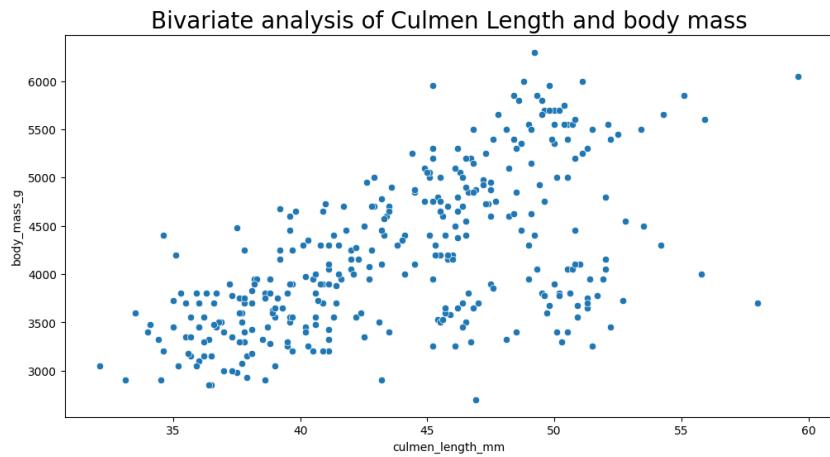
```

#1. Import thư viện và chuẩn bị dữ liệu phân tích
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
penguins_data = pd.read_csv("data/penguins_size.csv")
penguins_data = penguins_data[['species', 'culmen_length_mm', 'body_mass_g']]

#2. Phân tích dữ liệu 2 biến dựa vào phương pháp scatterplot
# case 1: basic
sns.scatterplot(data = penguins_data, x= penguins_data["culmen_length_mm"], y= penguins_data["body_mass_g"])
# case 2: advanced
plt.figure(figsize=(12,6))
ax = sns.scatterplot(data = penguins_data,
x= penguins_data["culmen_length_mm"], y= penguins_data["body_mass_g"])
ax.set_title('Bivariate analysis of Culmen Length and body mass', fontsize=20)

```

Hình 3. 23 Chuẩn bị dữ liệu và tạo scatterplot



Hình 3. 24 Kết quả biểu đồ scatterplot

Phân tích kết quả quan sát được:

- + Xu hướng các điểm tăng dần từ trái sang phải (càng dài mỏ → càng nặng cân), ta có thể rút ra:
- + Mối quan hệ dương (positive correlation) giữa chiều dài mỏ và khối lượng cơ thể.

➔ Chim có mỏ dài thường có khối lượng lớn hơn.

- + Nếu các điểm nằm rải rác nhưng vẫn có xu hướng tăng nhẹ theo đường chéo, đó là mối quan hệ tuyến tính yếu (weak positive correlation).
- + Nếu dữ liệu tạo thành các cụm rõ ràng, mỗi cụm có thể tương ứng với một loài chim khác nhau (ví dụ: Adelie, Chinstrap, Gentoo).
- + Một vài điểm nằm xa khỏi vùng tập trung → outliers, có thể do lỗi đo hoặc cá thể khác biệt.

- Phân tích 2 biến dựa vào bảng crosstab/two-way

```

import pandas as pd
penguins_data = pd.read_csv("data/penguins_size.csv")
penguins_data = penguins_data[['species', 'culmen_length_mm', 'body_mass_g', 'sex']]

#3. Phân tích 2 biến dựa vào bảng crosstab/two-way
pd.crosstab(index= penguins_data['species'], columns= penguins_data['sex'])


```

	sex	FEMALE	MALE
species			
Adelie	0	73	73
Chinstrap	0	34	34
Gentoo	1	58	61

Hình 3. 25 Thông kê crosstab/two-way

- Thông kê giới tính của từng loài

```

import pandas as pd
import numpy as np
penguins_data = pd.read_csv("data/penguins_size.csv")
penguins_data = penguins_data[['culmen_length_mm', 'species']]

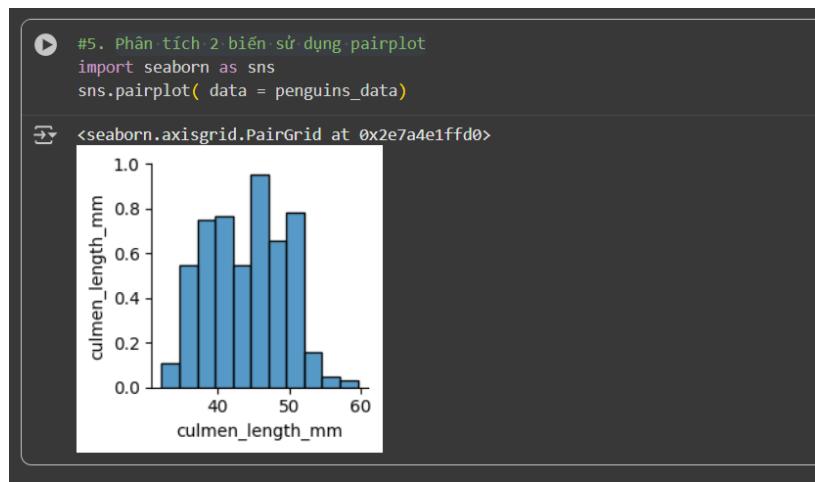
#4. Phân tích 2 biến sử dụng pivot_table
pd.pivot_table(penguins_data, values='culmen_length_mm', index='species', aggfunc=np.mean)


```

species	culmen_length_mm
Adelie	38.791391
Chinstrap	48.833824
Gentoo	47.504878

Hình 3. 26 Code tạo bảng Pivot table

- Phân tích 2 biến sử dụng pairplot



Hình 3. 27 Code tạo biểu đồ Pairplot

Nhận xét

+ Biến culmen_length_mm có phân phối trung chủ yếu trong khoảng 40–50 mm.

- + Có sự xuất hiện của một số cá thể đặc biệt với mỏ dài hơn 55 mm.
- + Dữ liệu này phù hợp để sử dụng cho phân tích thống kê hoặc huấn luyện mô hình phân loại loài chim dựa trên đặc điểm hình thái.

3. Bài toán 3:

Thực hiện các nhiệm vụ trong bài toán 3 để làm quen với việc sử dụng các công cụ hỗ trợ EDA tự động.

Nhiệm vụ 1: Sử dụng pandas profiling trên dữ liệu Customer Personality Analysis.

- Giả sử ta có tệp tin dữ liệu marketing_campaign, file này chứa thông tin mô phỏng của một chiến dịch marketing. Các bản ghi thể hiện thông tin khách hàng và hành vi tham gia chiến dịch.
- Các trường dữ liệu có trong file:

- + CustomerID – Mã định danh duy nhất của khách hàng.
- + Age – Tuổi của khách hàng.
- + Gender – Giới tính (Male / Female).
- + Income – Mức thu nhập hàng năm (USD).
- + Campaign_Joined – Tên chiến dịch marketing mà khách hàng tham gia (Campaign_A, Campaign_B, v.v.).
- + Purchase_Amount – Số tiền mà khách hàng đã chi tiêu trong chiến dịch.
- + Channel – Kênh tiếp thị mà khách hàng tiếp cận (Email, Social Media, TV, Web, v.v.).

```

# 1. Cài đặt pandas_profiling sau này đổi tên thành ydata_profiling
# pip install pandas-profiling
# pip install ydata-profiling

# 2. Sử dụng công cụ
import pandas as pd
from ydata_profiling import ProfileReport
marketing_data = pd.read_csv("data/marketing_campaign.csv")
# Create an Automated EDA report using the ProfileReport class in the ydata_profiling library.
# Use the to_file method to output the report to an HTML file
profile = ProfileReport(marketing_data)
profile.to_file("reports/profile_output.html")

# Sau khi thực thi, mở file HTML trong thư mục "reports" để xem báo cáo chi tiết.

```

Hình 3. 28 Code tạo báo cáo html

YData Profiling Report

Overview

Brought to you by YData

Dataset statistics		Variable types
Number of variables	1	Text
Number of observations	2240	
Missing cells	0	
Missing cells (%)	0.0%	
Duplicate rows	0	
Duplicate rows (%)	0.0%	
Total size in memory	17.6 kB	
Average record size in memory	8.1 B	

Hình 3. 29 Giao diện báo cáo html

```
#1. Cài đặt dtale
# pip install dtale

▶ #2. Sử dụng công cụ
import pandas as pd
import dtale
marketing_data = pd.read_csv("data/marketing_campaign.csv")
dtale.show(marketing_data).open_browser()

# Sau khi thực thi, trình duyệt web sẽ mở ra giao diện của D-Tale để bạn khám phá dữ liệu.
```

Hình 3. 30 Code để mở giao diện D-Table

III. Giới thiệu các tính năng và cách sử dụng thư viện SweetViz

1. Giới thiệu thư viện SweetViz

- **Sweetviz** là thư viện Python mã nguồn mở dùng để **Exploratory Data Analysis (EDA) tự động**. Nó tạo ra báo cáo HTML trực quan, tương tác với các phân tích thống kê chi tiết.
- Mục tiêu của nó là hỗ trợ để có một cái nhìn tổng quan dữ liệu mình có: phân bố, mối liên hệ giữa biến, giá trị thiểu, so sánh giữa các tập dữ liệu (ví dụ train vs test), và phân tích biến mục tiêu nếu có.

2. Những tính năng chính của SweetViz

- **Phân tích biến mục tiêu (Target analysis):** Điều này cho thấy giá trị mục tiêu liên quan như thế nào đến các tính năng khác.
- **So sánh hai tập dữ liệu:** Bạn có thể so sánh hai DataFrame (ví dụ: train và test) để xem phân bố biến, giá trị thiểu, mối quan hệ giữa các biến trong mỗi tập.
- **Liên kết giữa các loại biến hỗn hợp:** Với dữ liệu mà có biến số (numerical), biến phân loại (categorical) hoặc hỗn hợp, SweetViz dùng các phép đo tương

thích (ví dụ Pearson, hệ số không chắc chắn, correlation ratio) để đánh giá mối liên hệ.

- **Phát hiện kiểu dữ liệu tự động / ghi đè thủ công:** SweetViz tự nhận biết biến số, phân loại, văn bản, nhưng bạn có thể ghi đè nếu muốn.
- **Báo cáo thống kê & tổng hợp:** Cho mỗi biến: số lượng, giá trị thiếu, giá trị duy nhất, giá trị hay gặp nhất, min, max, trung vị, trung bình, độ lệch chuẩn, độ lệch (skewness), độ nhọn (kurtosis), hệ số biến động, v.v.

3. Cài đặt và sử dụng

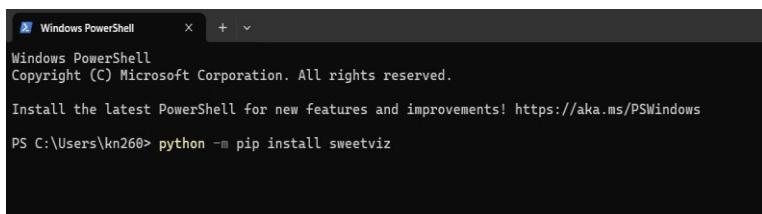
Bước 1: Cài đặt thư viện

```

*pip install sweetviz*

```

Ở trong Terminal hoặc PowerShell

A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the command "python -m pip install sweetviz" being typed at the prompt. The background of the window is dark, and the text is white.

Hình 3. 31 Giao diện cài đặt thư viện

Lưu ý: Phiên bản sweetviz lần cuối được cập nhật là khoảng tháng 11/2023, dẫn đến việc các phiên bản python và numpy (do sweetviz yêu cầu numpy) mới hơn xảy ra vấn đề không tương thích. Khuyến nghị khi sử dụng sweetviz nên cài python khoảng ver 3.9 và numpy ver 1.23.4 để đảm bảo hoạt động đúng.

Bước 2: Nhập thư viện & chuẩn bị dữ liệu

- Tải dữ liệu mẫu tại: [kaggle](#)
- + **Bối cảnh dữ liệu:** Phân tích Tính cách Khách hàng là một phân tích chi tiết về khách hàng lý tưởng của một công ty. Nó giúp doanh nghiệp hiểu rõ hơn về khách hàng của mình và giúp họ dễ dàng điều chỉnh sản phẩm theo nhu cầu, hành vi và mối quan tâm cụ thể của từng nhóm khách hàng khác nhau.

+ Phân tích tính cách khách hàng giúp doanh nghiệp điều chỉnh sản phẩm dựa trên khách hàng mục tiêu từ các phân khúc khách hàng khác nhau. Ví dụ, thay vì chi tiền để tiếp thị sản phẩm mới cho tất cả khách hàng trong cơ sở dữ liệu của công ty, doanh nghiệp có thể phân tích phân khúc khách hàng nào có khả năng mua sản phẩm cao nhất và sau đó chỉ tiếp thị sản phẩm cho phân khúc cụ thể đó.

- Một số thông tin về tập dữ liệu:

+ Thông tin về người (People):

- ID: Mã định danh khách hàng
- Year_Birth: năm sinh khách hàng
- Education: trình độ học vấn của khách hàng
- Marital_Status: Tình trạng hôn nhân (8 loại bao gồm các giá trị bất thường như "YOLO", "Absurd", "Alone")
- Income: Thu nhập hàng năm (có giá trị missing)
- Kidhome: Số trẻ em nhỏ trong gia đình
- Teenhome: Số thanh thiếu niên trong gia đình
- Dt_customer: Ngày khách hàng đăng ký với công ty
- Recency: Số ngày kể từ lần mua hàng cuối cùng của khách hàng
- Complain: 1 nếu khách hàng đã khiếu nại trong 2 năm qua, 0 nếu không

+ Hành vi mua hàng (products):

- MntWines: Số tiền chi cho rượu vang trong 2 năm qua
- MntFruits: Số tiền chi cho trái cây trong 2 năm qua
- MntMeatProducts: Số tiền chi cho thịt trong 2 năm qua
- MntFishProducts: Số tiền chi cho hải sản trong 2 năm qua
- MntSweetProducts: Số tiền chi cho đồ ngọt trong 2 năm qua
- MntGoldProds: Số tiền chi cho vàng trong 2 năm qua

+ Khuyến mãi (promotion):

- NumDealsPurchases: Số lượng mua hàng được giảm giá
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise. 1 nếu khách hàng chấp nhận ưu đãi trong chiến dịch đầu tiên, 0 nếu không
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise. Lần thứ 2

- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise. Lần thứ 3
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise. Lần thứ 4
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise. Lần thứ 5
- Response: 1 if customer accepted the offer in the last campaign, 0 otherwise. Lần cuối.

+ Địa điểm (Place):

- NumWebPurchases: Số lượng mua hàng được thực hiện thông qua trang web của công ty
- NumCatalogPurchases: Số lần mua hàng qua Catalog
- NumStorePurchases: Số lần mua hàng trực tiếp tại cửa hàng
- NumWebVisitsMonth: Số lần truy cập web mỗi tháng

Bước 3: Đọc dữ liệu bằng Pandas

```
```
import pandas as pd
df = pd.read_csv('marketing_campaign.csv', delimiter='\t')
````
```

Bước 4: Tạo báo cáo

```
```
report = sv.analyze(df)
report.show_html("marketing_campaign.html")
````
```

+ # Sweetviz sẽ:

+ # Sweetviz tự động:

- Đếm số lượng: rows, columns
- Xác định kiểu dữ liệu: numeric, categorical, text, datetime
- Phát hiện duplicate rows
- Tính tỷ lệ missing values cho từng cột

Phân tích từng biến (Per-Feature Analysis):

Với biến số (Numerical) là các thống kê mô tả, Biểu đồ phân phối

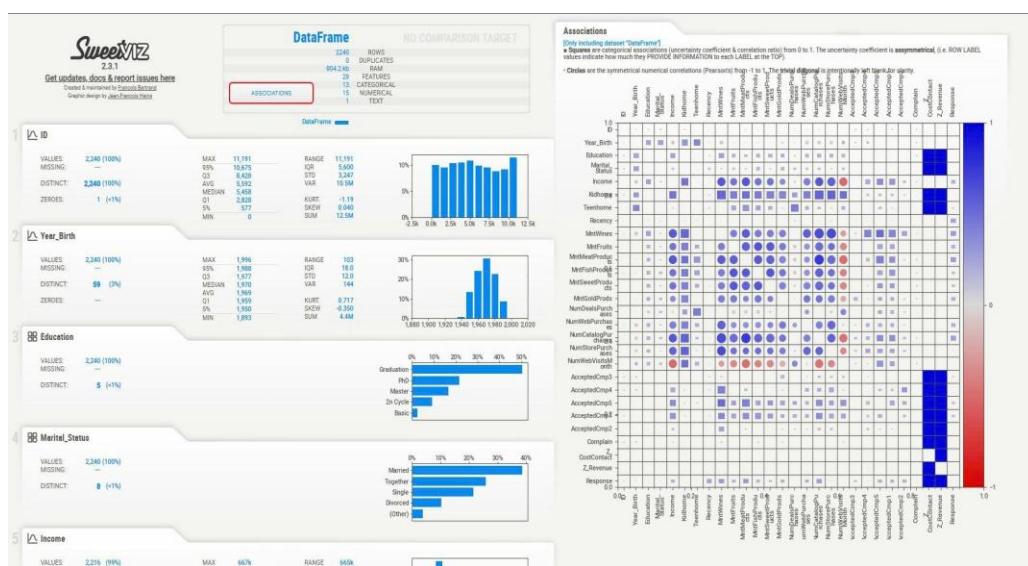
Với biến phân loại (Categorical) là các count của biến phân loại

Tạo cấu trúc báo cáo (Report Generation)

Render HTML

Sau khi tạo xong chỉ cần mở nó chạy trên trình duyệt.

- Kết quả

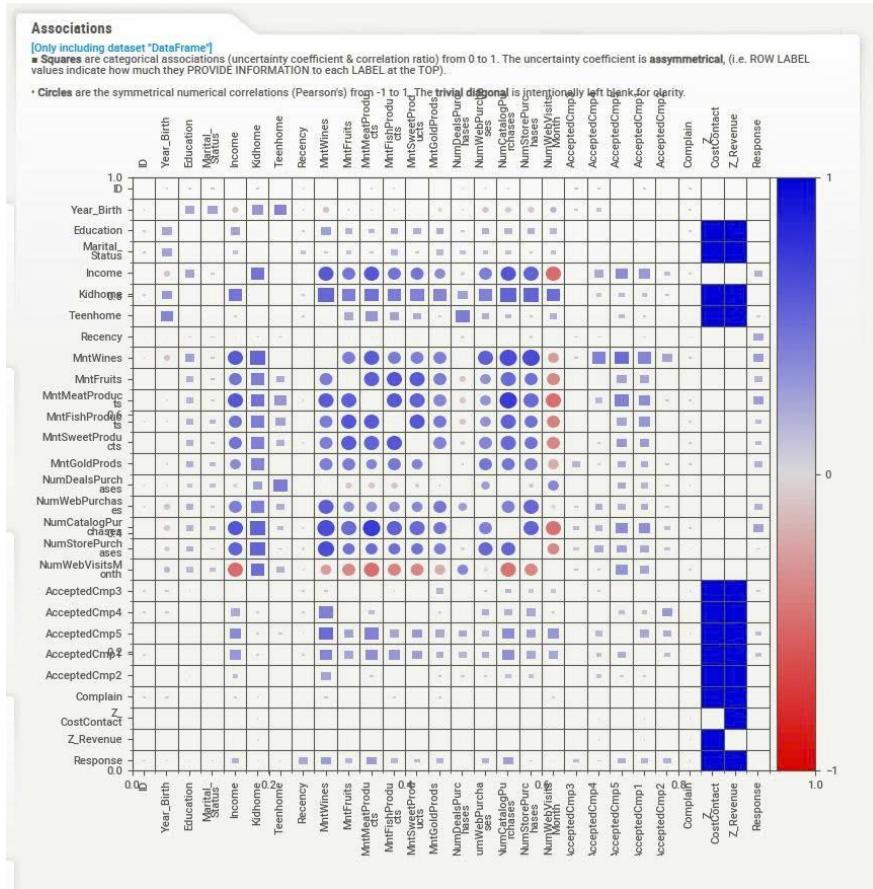


Hình 3. 32 Giao diện tổng quan của báo cáo

- Biểu đồ Associations được hiển thị khi bạn bấm vào:



Hình 3. 33 Vị trí nhấp để xem Associations



Hình 3.34 Giao diện của Associations

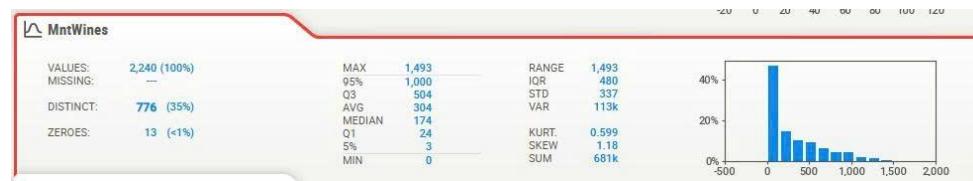
- Mức ý nghĩa:
 - + **Hình tròn (circles)**: là *correlation* giữa hai biến số (Pearson).
 - Kích thước vòng / kích thước điểm → độ lớn tuyệt đối của hệ số tương quan ($|r|$).
 - Màu sắc cho biết dấu: trong báo cáo của bạn, **xanh đậm** $\approx +1$, **đỏ** ≈ -1 , màu nhạt \approx gần 0.
 - Vòng to + xanh dương: tương quan dương mạnh; vòng to + đỏ: tương quan âm mạnh.
 - + **Hình vuông (squares)**: là chỉ số hiệp hội cho **biến phân loại / hỗn hợp** (SweetViz dùng uncertainty coefficient / correlation ratio).

- Vuông càng đậm/càng to → mối liên hệ mạnh (phạm vi 0..1). **Lưu ý: đây là chỉ số bát đối xứng** — nghĩa là hàng → cột chưa chắc bằng cột → hàng; SweetViz hiển thị theo quy ước: **ROW LABEL** cho biết bao nhiêu thông tin được cung cấp cho LABEL ở **TOP**.
- **Đường chéo (diagonal)** thường để trống (trivial) — không có thông tin hữu ích ở đó.
- **Thanh màu (colorbar)** bên phải: cho biết quy ước màu (xanh = +1, đỏ = -1).
 - Như đã đề cập trước đó Sweetviz sẽ Phân tích từng biến và hiển thị. Để xem được biểu đồ đơn biến chỉ cần ấn vào 1 trong danh sách các biến bên trái:



Hình 3. 35 Phản giao diện chính của biểu đồ dựa trên biến

- Kết quả với biến giá trị:
 - + Các thông số trên thẻ biến:



Hình 3. 36 Phản giao diện chính của biểu đồ dựa trên biến

- + **VALUES: 2,240 (100%)**: Số lượng bản ghi không rỗng.
- + **MISSING**: Số hoặc kí hiệu nếu có giá trị thiếu.
- + **DISTINCT: 776 (35%)** Số giá trị khác nhau (unique) và tỉ lệ so với tổng. Ví dụ MntWines có 776 giá trị khác nhau, chiếm 35% tổng bản ghi.
- + **ZEROS: 13 (<1%)**: Số bản ghi có giá trị = 0 (và tỉ lệ%). Quan trọng để biết có nhiều khách hàng không mua mặt hàng đó hay không.

+ **MIN / 5% / Q1 / MEDIAN / AVG / Q3 / 95% / MAX**

+ Các chỉ số phân vị và thống kê cơ bản:

- **MIN**: giá trị nhỏ nhất.
- **5%**: phân vị 5% (giá trị dưới 5% thấp nhất).
- **Q1**: 25% quantile. Đây là tứ phân vị thứ 1 (25%) nghĩa là **25% khách hàng chi tiêu ≤ 24** cho rượu.
- **MEDIAN**: 50% quantile (giá trị ở giữa).
- **AVG**: trung bình (mean).
- **Q3**: Đây là tứ phân vị thứ 3 (75%). Có nghĩa là **75% khách hàng chi tiêu ≤ 504**, và chỉ 25% khách hàng chi tiêu nhiều hơn 504.
- **95%**: phân vị 95% (giúp thấy mức giá trị rất cao).
- **MAX**: giá trị lớn nhất.

+ **RANGE** = MAX – MIN(khoảng giá trị).

+ **IQR** = Q3 – Q1 (inter-quartile range — độ rộng khoảng giữa 25% và 75%).

+ **STD** (standard deviation) Độ lệch chuẩn — đo độ phân tán.

+ **VAR** (variance) Phương sai (STD^2).

+ **KURT. (kurtosis)** Độ nhọn/đuôi của phân phối.

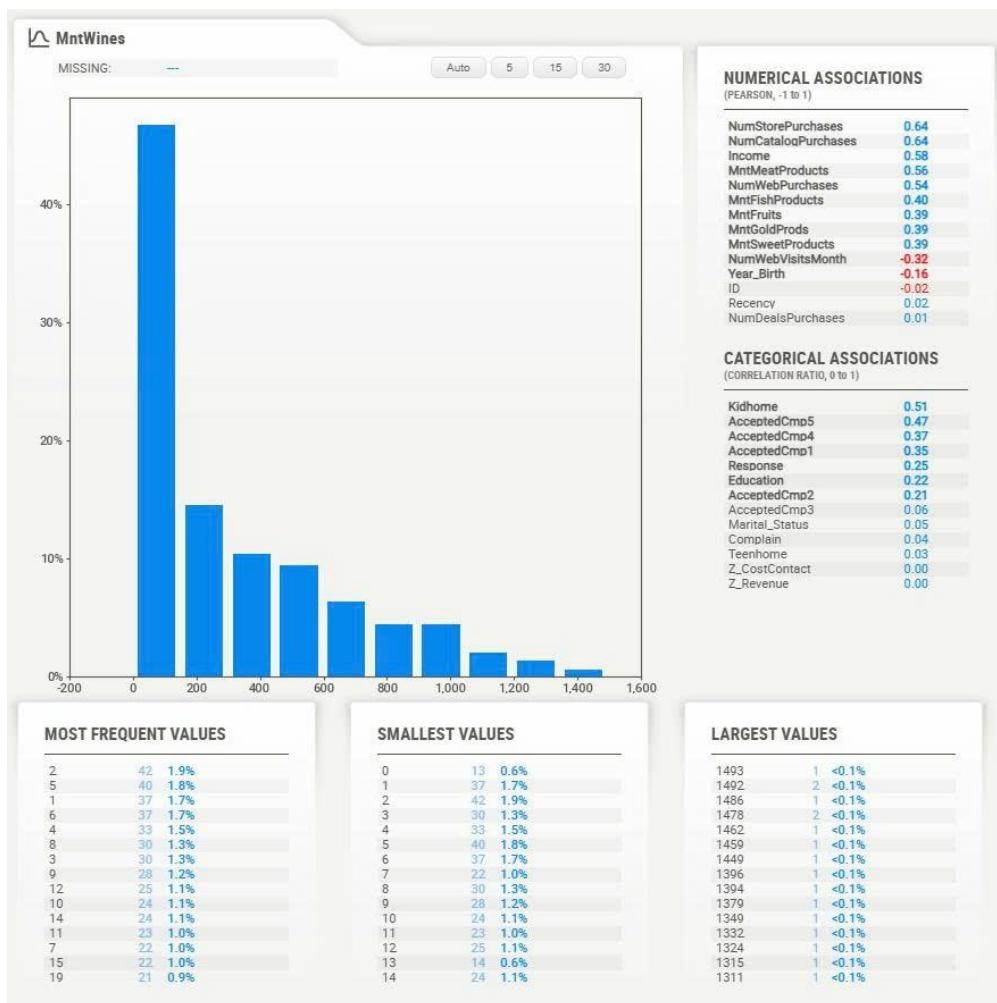
- Kurtosis > 0: đuôi dày (outliers nhiều hơn so với chuẩn).
- Kurtosis < 0: phân phối bằng phẳng hơn.

+ **SKEW (skewness)** Độ lệch trái/phải:

- > 0: *right-skewed* (đa số nhỏ, vài giá trị lớn — đuôi sang phải).
- < 0: *left-skewed*.

+ **SUM** Tổng giá trị (ví dụ tổng tiền chi cho MntWines).

Biểu đồ Histogram:



Hình 3. 37 Phần giao diện biểu đồ Histogram và các loại thống kê khác

- + **Histogram** thể hiện phân phối giá trị.
- + Trục hoành = giá trị biến (tiền chi tiêu), trục tung = tần suất (số lượng hoặc phần trăm).
- + Nếu histogram có **cột cao bên trái** và đuôi kéo dài sang phải → phân phối lệch phải (right-skew).
- + Ở đây: phần lớn khách hàng chi tiêu thấp (cột cao ở vùng gần 0–200 và giảm dần), có một “đuôi dài” tới các giá trị lớn (một vài khách bỏ ra rất nhiều tiền).

Các biểu đồ “most frequent / smallest / largest values”:

- + **Most frequent values:** danh sách các giá trị xuất hiện nhiều nhất (và tỉ lệ). Thường áp dụng cho biến rời rạc.
 - **Smallest values:** các giá trị nhỏ xuất hiện (và tần suất).
 - **Largest values:** các giá trị lớn nhất (và tần suất — thường rất ít).

+ Numerical associations:

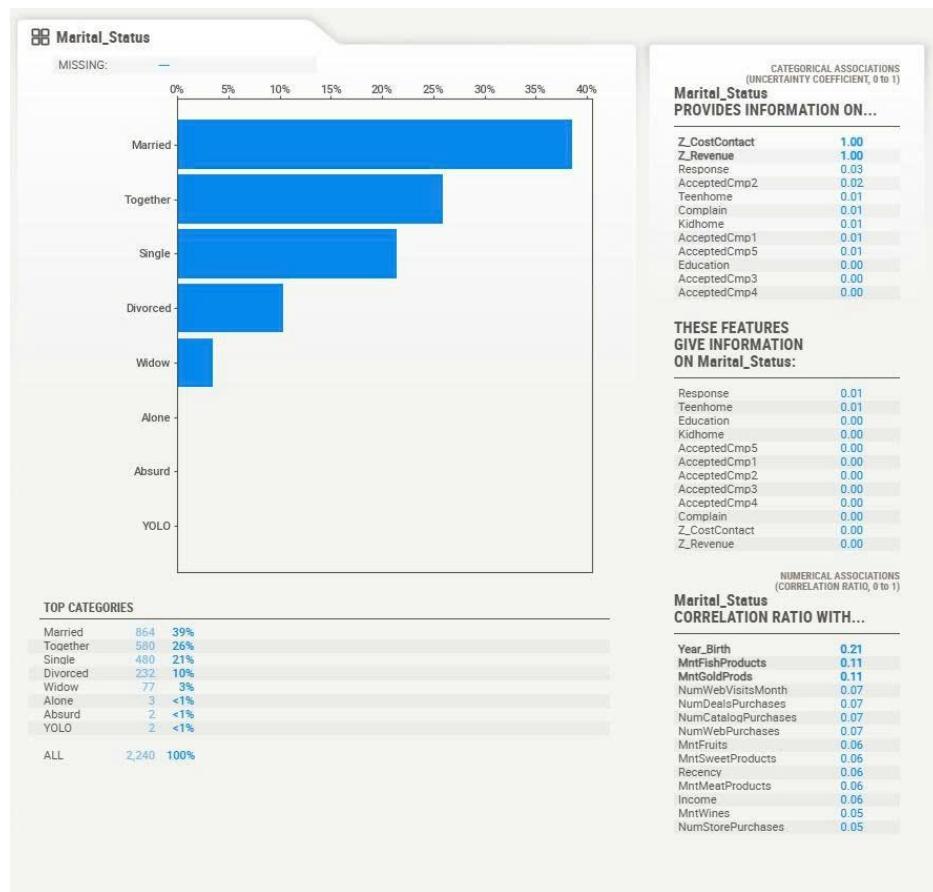
- Hiển thị các **Pearson correlations** (từ -1 đến +1) giữa biến đang xem (ví dụ MntWines) và các biến số khác.
- Giá trị gần +1: tương quan dương mạnh (ký hiệu màu xanh).
- Giá trị gần -1: tương quan âm mạnh (màu đỏ).
- Trị số giúp biết biến nào “liên quan” với MntWines — ví dụ MntWines có thể tương quan tốt với NumStorePurchases, NumCatalogPurchases, Income...

+ Categorical associations:

- Cho biết sự liên hệ giữa biến đang xem và **các biến phân loại** (dùng correlation ratio / uncertainty coefficient; thang 0..1).
- Ví dụ Kidhome 0.51 nghĩa là có mức liên hệ đáng kể giữa có con nhỏ và giá trị MntWines (nếu đó là cách SweetViz hiển nghĩa).
- Dùng để xem các biến category giải thích được bao nhiêu biến khác nhau.
 - Với biến phân loại

+ Các thông số trên thẻ biến được rút gọn chỉ còn: Values, Missing, Distinct

+ Biểu đồ:



Hình 3. 38 Phàn giao diện biểu đồ Histogram và các loại thông kê khác của biến phân loại

+ Top categories: Hiển thị tần suất xuất hiện trong data của biến phân loại

➔ Phàn lớn khách hàng là Married hoặc Together (tổng cộng ~65%), Single chiếm 21%.
Những giá trị hiếm như YOLO, Absurd, Alone có thể được coi là outliers / lỗi nhập liệu.

Categorical associations:

+ Marital_Status PROVIDES INFORMATION:

Xem biến Marital_Status cung cấp thông tin cho biến nào khác.

Ở đây, các giá trị cao nhất:

- Z_CostContact = 1.00 và Z_Revenue = 1.00 (nhưng hai biến này trong dataset mặc định là hằng số = không có ý nghĩa).
- Response = 0.03 → tình trạng hôn nhân có chút ảnh hưởng đến việc khách hàng phản hồi chiến dịch, nhưng rất yếu.
- Các biến khác <0.02 → gần như không có tác động.

+ THESE FEATURES GIVE INFORMATION ON Marital_Status:

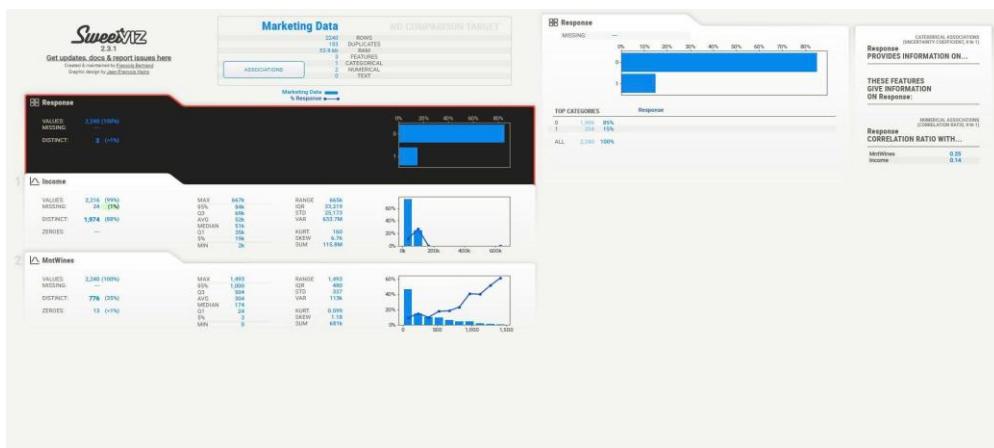
Ngược lại: biến nào giải thích cho Marital_Status.

- Tất cả hệ số rất thấp (max chỉ 0.01). → Nghĩa là tình trạng hôn nhân gần như độc lập so với các biến khác trong dataset.
 - + Numerical associations:
 - Hiển thị mức độ liên hệ (0..1) giữa biến phân loại và các biến số.
 - Year_Birth = 0.21 → tuổi (năm sinh) có liên quan vừa phải đến tình trạng hôn nhân (hợp lý: người trẻ thường Single / Together, lớn tuổi thường Married / Widow).
 - Tạo báo cáo với biến mục tiêu: Sweetviz có thể tạo báo cáo trên biến mục tiêu với các cột chỉ định
- ```

```
data = df[["Response", "Income", "MntWines"]].copy()
# Tạo báo cáo SweetViz, với 'Response' là biến mục tiêu
report = sv.analyze([data, "Marketing Data"], target_feat="Response")

#show báo cáo
report.show_html("sweetviz_response_report.html")
```
```

- Kết quả:



Hình 3. 39 Phản giao diện biểu đồ Histogram và các loại thống kê khác khi tạo báo cáo kèm biến mục tiêu

- Ngoài ra Sweetviz cũng có thể trực quan hợp biến:
- ```

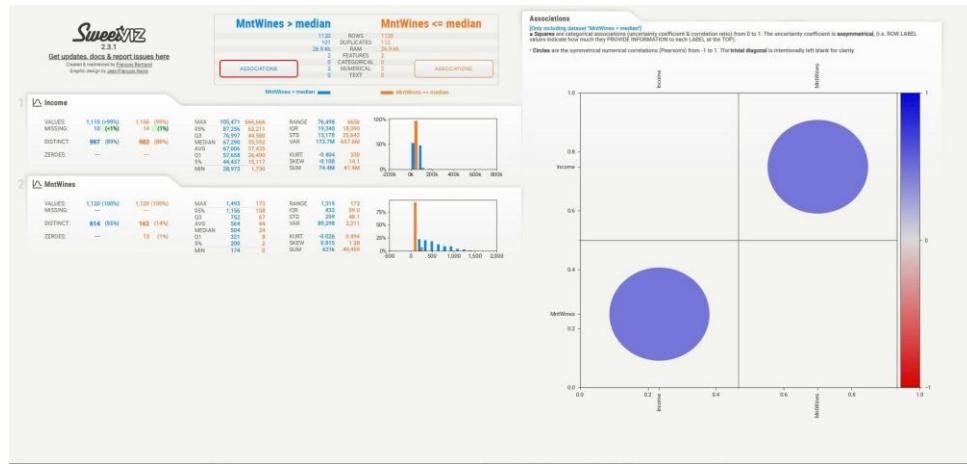
```
Chọn 2 biến cần phân tích
data1 = df[["Income", "MntWines"]].copy()
median_wine = data1["MntWines"].median()
```

```

report = sv.compare_intra(
 data1,
 data1["MntWines"] > median_wine,
 ["MntWines > median", "MntWines <= median"]
)
report.show_html("sweetviz_income_wine.html")
```

```

- Kết quả



Hình 3. 40 Phản giao diện biểu đồ Histogram và các loại thống kê khác khi tạo báo cáo với hợp biên

TÀI LIỆU THAM KHẢO

- [1] Bradley University Online, “What’s the difference between descriptive and inferential statistics?”, *Bradley University Online*, [Online]. Available: <https://onlinedegrees.bradley.edu/blog/descriptive-vs-inferential-statistics/>.
- [2] Laerd Statistics, “Descriptive and inferential statistics,” *Laerd Statistics*, 2018. [Online]. Available: <https://statistics.laerd.com/>.
- [3] D. S. Moore, G. P. McCabe, and B. A. Craig, *Introduction to the Practice of Statistics*, 9th ed. New York, NY: W.H. Freeman and Company, 2017.
- [4] UCLA Institute for Digital Research and Education (IDRE), “What is a normal distribution?”, *UCLA IDRE*, 2023. [Online]. Available: <https://stats.oarc.ucla.edu/>.
- [5] H. J. Seltman, *Experimental Design and Analysis*, Carnegie Mellon University, 2018. [Online]. Available: <http://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf>.
- [6] ChatGPT, “Trao đổi và phân tích về thống kê mô tả và suy luận,” *OpenAI ChatGPT*, 2025.
- [7] Gemini AI, “Phân biệt thống kê mô tả và thống kê suy luận,” *Google DeepMind*, 2025.
- [8] Amazon Web Services (AWS), “What is data visualization?”, *AWS*, [Online]. Available: <https://aws.amazon.com/vi/what-is/data-visualization/>.
- [9] Lạc Việt, “Trực quan hóa dữ liệu,” *Lạc Việt*, [Online]. Available: <https://lacviet.vn/truc-quan-hoa-du-lieu/>.
- [10] 200Lab, “Danh mục các loại biểu đồ trong data visualization,” *200Lab Blog*, [Online]. Available: <https://200lab.io/blog/danh-muc-cac-loai-bieu-do-trong-data-visualization/>.