**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**˗˗˗˗˗˗˗˗˗˗˗🕮˗˗˗˗˗˗˗˗˗˗˗**A blue circle with white text

Description automatically generated

**BÁO CÁO NHÓM**

**HỌC PHẦN: PHÂN TÍCH DỮ LIỆU**

**PHÂN TÍCH KHÁM PHÁ VỀ BỆNH ĐÁI THÁO ĐƯỜNG**

|  |  |
| --- | --- |
| **Mã học phần** | **841432** |
| **Học kỳ** | **1** |
| **GVHD** | **Đỗ Như Tài** |
| **MSSV** | **3122410490** |
| **Họ và tên** | **Nguyễn Hoàng Mai Vy** |

**TP. HỒ CHÍ MINH THÁNG 09 NĂM 2025**

**MỤC LỤC**

[**BẢNG PHÂN CÔNG CÔNG VIỆC** 2](#_Toc211019657)

[**I. THỐNG KÊ MÔ TẢ** 3](#_Toc211019658)

[***1. Thống kê mô tả `là gì? Khác gì với thống kê suy luận?*** 3](#_Toc211019659)

[***2. Các thước đo thống kê mô tả chính và khi nào dùng trung vị thay vì trung bình?*** 3](#_Toc211019660)

[***3. Cách xác định phân bố của dữ liệu & các loại phổ biến*** 3](#_Toc211019661)

[***4. Ý nghĩa của độ lệch chuẩn và phạm vi*** 4](#_Toc211019662)

[***5. Q1, Q2, Q3 trong boxplot*** 4](#_Toc211019663)

[***6. Xử lý giá trị thiếu (missing values)*** 4](#_Toc211019664)

[***7. Cách đọc và diễn giải histogram, boxplot*** 4](#_Toc211019665)

[***8. Xử lý outliers trong thống kê mô tả*** 5](#_Toc211019666)

[**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 5](#_Toc211019667)

[**PHẦN I. TIỀN XỬ LÝ DỮ LIỆU VÀ ỨNG DỤNG THỐNG KÊ** 7](#_Toc211019668)

[***Bài thực hành 1: Red Wine Quality*** 7](#_Toc211019669)

[***Bài thực hành 2: Pima Indian Diabetes*** 11](#_Toc211019670)

[**PHẦN II. XỬ LÝ VÀ TRỰC QUAN HÓA DỮ LIỆU** 16](#_Toc211019671)

[***2.1. Vai trò của trực quan hóa dữ liệu trong phân tích dữ liệu*** 16](#_Toc211019672)

[***2.2. Các loại biểu đồ phổ biến và khi nào nên sử dụng*** 16](#_Toc211019673)

[***2.3. Cách chọn loại biểu đồ phù hợp*** 16](#_Toc211019674)

[***2.4. So sánh các thư viện trực quan hóa trong Python*** 16](#_Toc211019675)

[***2.5. Nguyên tắc thiết kế biểu đồ hiệu quả*** 17](#_Toc211019676)

[***2.6. Bài toán dự đoán giá nhà Amsterdam (Amsterdam House Price Prediction)*** 17](#_Toc211019677)

[**PHẦN III. PHÂN TÍCH ĐƠN BIẾN VÀ HAI BIẾN** 57](#_Toc211019678)

[***3.1. Phân tích đơn biến (Univariate Analysis) là gì ? Nó khác gì với phân tích hai biến (bivariate analysis) trong khám phá dữ liệu?*** 57](#_Toc211019679)

[***3.2. Các thước đo thống kê trong phân tích đơn biến*** 57](#_Toc211019680)

[***3.3. Cách xác định mối quan hệ giữa hai biến*** 57](#_Toc211019681)

[***3.4 Sự khác biệt giữa Tương quan (Correlation) và Hiệp biến (Covariance)*** 58](#_Toc211019682)

[***3.5. Biểu đồ trực quan hóa khi nào dùng*** 58](#_Toc211019683)

[***3.6. Scatter plot & Heatmap*** 59](#_Toc211019684)

[***3.7. Trực quan hóa mối quan hệ giữa biến số và biến phân loại*** 60](#_Toc211019685)

[***3.8. Phân tích dữ liệu*** 61](#_Toc211019686)

[**Bài tập thực hành 1:** Tìm hiểu các tính năng và cách sử dụng sản phẩm SweetViz 73](#_Toc211019687)

[**Bài tập thực hành 2:** Tìm hiểu các tính năng và cách sử dụng sản phẩm AutoViz 76](#_Toc211019688)

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Công việc** |
| Nguyễn Hoàng Mai Vy | 3122410490 | * Phân công công cả nhóm * Đọc bộ dữ liệu * Code phần 2 * Làm slides |
| Dương Bình Minh | 3122410236 | * Đọc bộ dữ liệu * Code phần 3 * Viết báo cáo |
| Đỗ Thanh Duy | 3120410085 | * Đọc bộ dữ liệu * Code phần 1 với phụ phần 3 * Viết báo cáo |
| Nguyễn Minh Quang | 3123410290 | * Đọc bộ dữ liệu * Code phần 1 với phụ phần 3 * Làm slides |

# **I. THỐNG KÊ MÔ TẢ**

## ***1. Thống kê mô tả là gì? Khác gì với thống kê suy luận?***

**Thống kê mô tả (Descriptive statistics):** Là tập hợp các phương pháp dùng để tóm tắt, mô tả, trình bày dữ liệu thu thập được. Nó chỉ nói về mẫu/tập dữ liệu hiện có, không mở rộng ra ngoài. → Ví dụ: tính trung bình điểm thi của lớp, vẽ histogram, boxplot.

**Thống kê suy luận (Inferential statistics):** Dùng dữ liệu mẫu để suy ra, ước lượng, kiểm định giả thuyết cho cả quần thể. → Ví dụ: dự đoán điểm trung bình của toàn bộ sinh viên dựa vào một nhóm mẫu.

**Khác biệt:**

* Mô tả = chỉ “miêu tả hiện tại” dữ liệu có.
* Suy luận = đưa ra kết luận, dự đoán cho quần thể.

## ***2. Các thước đo thống kê mô tả chính và khi nào dùng trung vị thay vì trung bình?***

**Trung bình (mean):** giá trị bình quân. Dùng khi dữ liệu phân bố đối xứng, không có nhiều ngoại lai.

**Trung vị (median):** giá trị ở giữa sau khi sắp xếp. Dùng khi dữ liệu bị lệch (skewed) hoặc có ngoại lai (outlier) vì nó ít bị ảnh hưởng.

**Phương sai (variance):** đo mức độ phân tán quanh trung bình.

**Độ lệch chuẩn (standard deviation):** căn bậc hai của phương sai, giúp đánh giá dữ liệu “chênh lệch” nhiều hay ít so với trung bình.

## ***3. Cách xác định phân bố của dữ liệu & các loại phổ biến***

**Cách xác định:**

* Vẽ histogram hoặc density plot.
* Kiểm tra skewness, kurtosis.
* Dùng kiểm định (Shapiro-Wilk, Kolmogorov-Smirnov) để test phân bố chuẩn.
* Các loại phân bố phổ biến:
* Chuẩn (normal distribution): đối xứng, hình chuông.
* Lệch trái (left-skewed): đuôi dài bên trái.
* Lệch phải (right-skewed): đuôi dài bên phải.

## ***4. Ý nghĩa của độ lệch chuẩn và phạm vi***

Phạm vi (range): khoảng cách giữa giá trị lớn nhất và nhỏ nhất → cho biết “khoảng trải rộng” của dữ liệu.

Độ lệch chuẩn: cho biết các điểm dữ liệu thường “cách” trung bình bao nhiêu.

  Nhỏ → dữ liệu tập trung.  
  
  Lớn → dữ liệu phân tán nhiều.

## ***5. Q1, Q2, Q3 trong boxplot***

**Q1 (quartile 1)**: 25% dữ liệu nhỏ hơn giá trị này.

**Q2 (quartile 2, median):** 50% dữ liệu nhỏ hơn → chính là trung vị.

**Q3 (quartile 3):** 75% dữ liệu nhỏ hơn giá trị này.

Boxplot cho ta cái nhìn nhanh về phân bố, độ lệch, ngoại lai.

## ***6. Xử lý giá trị thiếu (missing values)***

Xóa: bỏ những hàng/cột có giá trị thiếu (nếu số lượng ít, không ảnh hưởng nhiều).

Thay thế (imputation):

Bằng trung bình/median (số liệu liên tục).

Bằng mode (dữ liệu phân loại).

Bằng giá trị dự đoán (sử dụng mô hình hồi quy, KNN imputation, …).

## ***7. Cách đọc và diễn giải histogram, boxplot***

*Histogram:*

* Dùng để xem phân bố dữ liệu (chuẩn, lệch trái, lệch phải).
* Nhìn “hình chuông” → dữ liệu gần chuẩn.
* Đỉnh cao bên trái hoặc phải → lệch.

*Boxplot:*

* Hộp thể hiện Q1–Q3, đường giữa hộp = median.
* Độ dài hộp = IQR (Q3 – Q1) → mức độ phân tán.
* Dấu chấm/cột ngoài → ngoại lai.

## ***8. Xử lý outliers trong thống kê mô tả***

* Kiểm tra lại xem có phải lỗi nhập liệu không → nếu có thì sửa hoặc bỏ.
* Nếu là giá trị hợp lệ nhưng quá cực đoan:
* Dùng trung vị thay cho trung bình.
* Dùng biến đổi dữ liệu (log, sqrt) để giảm ảnh hưởng.
* Giữ lại nếu chúng mang ý nghĩa (ví dụ: bệnh nhân cực kỳ cao tuổi).
* Cắt bỏ (winsorization) hoặc thay thế bằng ngưỡng (Q1 – 1.5×IQR, Q3 + 1.5×IQR).

# **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
* Nhiệm vụ 2: Khám phá và xử lý dữ liệu Marketing Campaign

**Khám phá dữ liệu COVID:**

Tải dữ liệu COVID từ OWID [https://ourworldindata.org/](https://www.google.com/url?q=https%3A%2F%2Fourworldindata.org%2F)

Nhận xét : Dữ liệu về số ca nhiễm COVID-19 mới trung bình 7 ngày trên mỗi triệu dân cho thấy sự phân bố không đồng đều và lệch phải mạnh. Giá trị trung bình là 99,51, trong khi trung vị chỉ 0,87, cho thấy phần lớn quốc gia và thời điểm ghi nhận rất ít hoặc không có ca nhiễm mới, nhưng một số trường hợp bùng phát mạnh đã kéo giá trị trung bình tăng cao. Độ lệch chuẩn lớn (505,53) phản ánh mức độ biến động mạnh giữa các quốc gia và thời kỳ. Giá trị nhỏ nhất bằng 0 và lớn nhất tới 37.463,75 chứng tỏ có sự chênh lệch rất lớn trong tình hình dịch bệnh. Nhìn chung, dữ liệu thể hiện rõ đặc trưng của đại dịch COVID-19: đa số khu vực kiểm soát ổn định, song vẫn tồn tại những đợt bùng phát nghiêm trọng tại một số quốc gia hoặc giai đoạn nhất định.

**Khám phá và xử lý dữ liệu Marketing Campaign:**

**Nhận xét :** Bộ dữ liệu gồm 511.939 quan sát (dòng) và 3 biến (cột), thể hiện số ca nhiễm COVID-19 mới được xác nhận theo từng quốc gia và từng ngày.  
Ba cột chính gồm:

1. Entity – tên quốc gia hoặc vùng lãnh thổ.
2. Day – ngày ghi nhận dữ liệu (định dạng YYYY-MM-DD).
3. Daily new confirmed cases of COVID-19 per million people (rolling 7-day average, right-aligned) – số ca nhiễm mới trung bình 7 ngày trên mỗi triệu dân.

Các giá trị ban đầu (ví dụ Afghanistan từ 2020-01-09 đến 2020-01-13) đều bằng 0, phản ánh giai đoạn đầu dịch chưa ghi nhận ca nhiễm.  
Dữ liệu có quy mô lớn, bao phủ nhiều quốc gia và thời điểm khác nhau, rất phù hợp cho việc phân tích xu hướng, so sánh mức độ bùng phát dịch, và đánh giá hiệu quả kiểm soát dịch bệnh theo thời gian.

# **PHẦN I. TIỀN XỬ LÝ DỮ LIỆU VÀ ỨNG DỤNG THỐNG KÊ**

## ***Bài thực hành 1: Red Wine Quality***

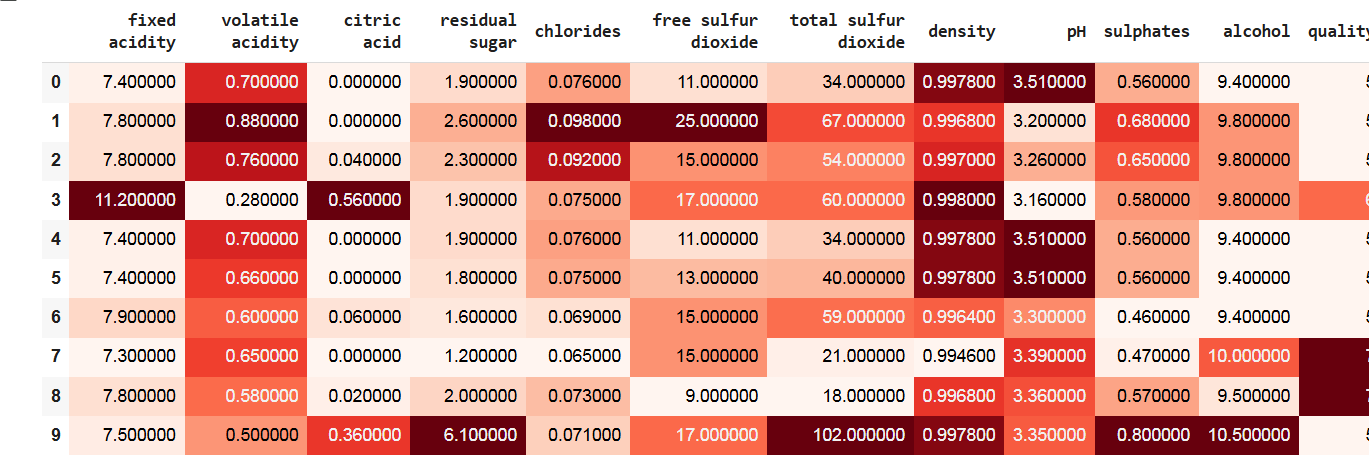
1. Khai báo thư viện
2. import pandas as pd
3. import numpy as np
4. import seaborn as sns
5. # import matplotlib as plt
6. import math
7. from sklearn.model\_selection import train\_test\_split
8. import matplotlib.pyplot as plt
9. Nạp dữ liệu từ Drive

from google.colab import drive

drive.mount('/content/drive/')

df\_rwine = pd.read\_csv('/content/drive/My Drive/winequality-red.csv', sep=',')

df\_rwine.head(10).style.background\_gradient(cmap="Reds")



1. Xem xét tính đầy đủ của dữ liệu

missing\_data = []

for col in df\_rwine.columns:

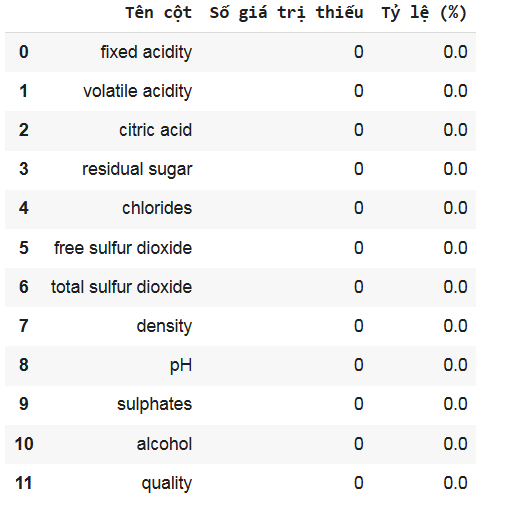
    missing\_value = df\_rwine[col].isna().sum()

    missing\_percent = missing\_value / len(df\_rwine) \* 100

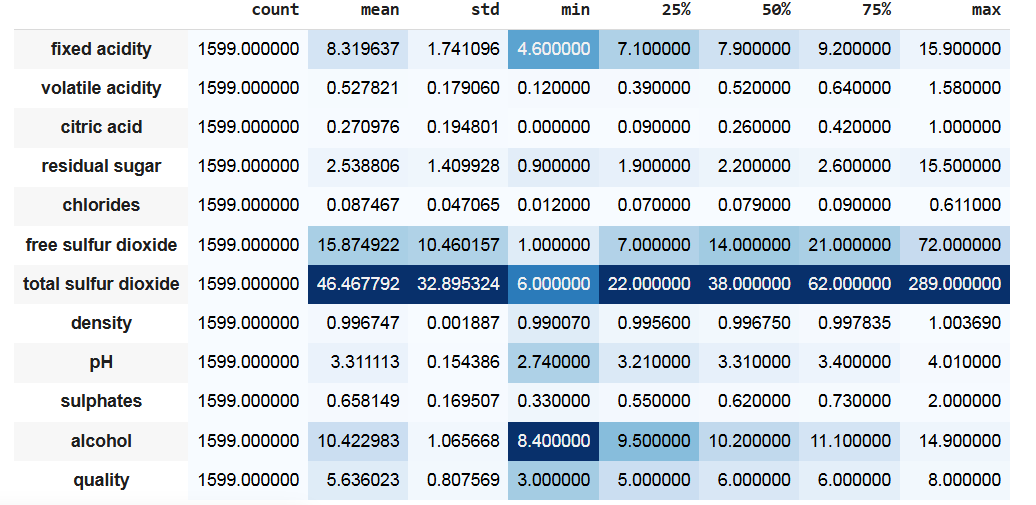
    missing\_data.append([col, missing\_value, missing\_percent])

missing\_df = pd.DataFrame(missing\_data, columns=['Tên cột', 'Số giá trị thiếu', 'Tỷ lệ (%)'])

display(missing\_df)



df\_rwine.describe().T.style.background\_gradient(cmap = "Blues")

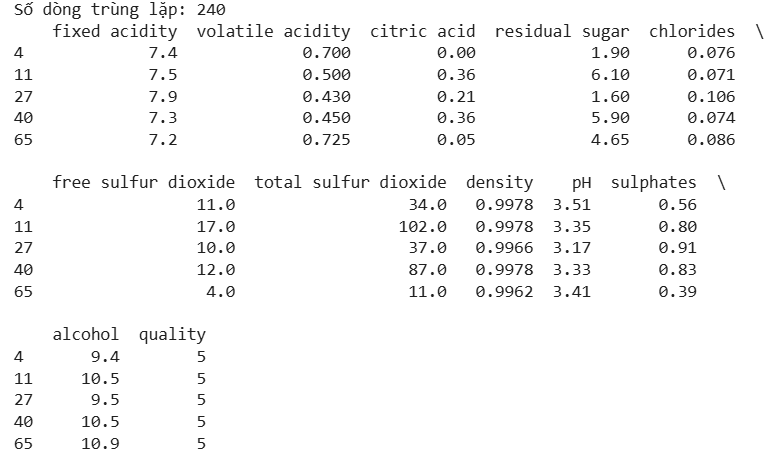


1. Loại bỏ các bản ghi trùng

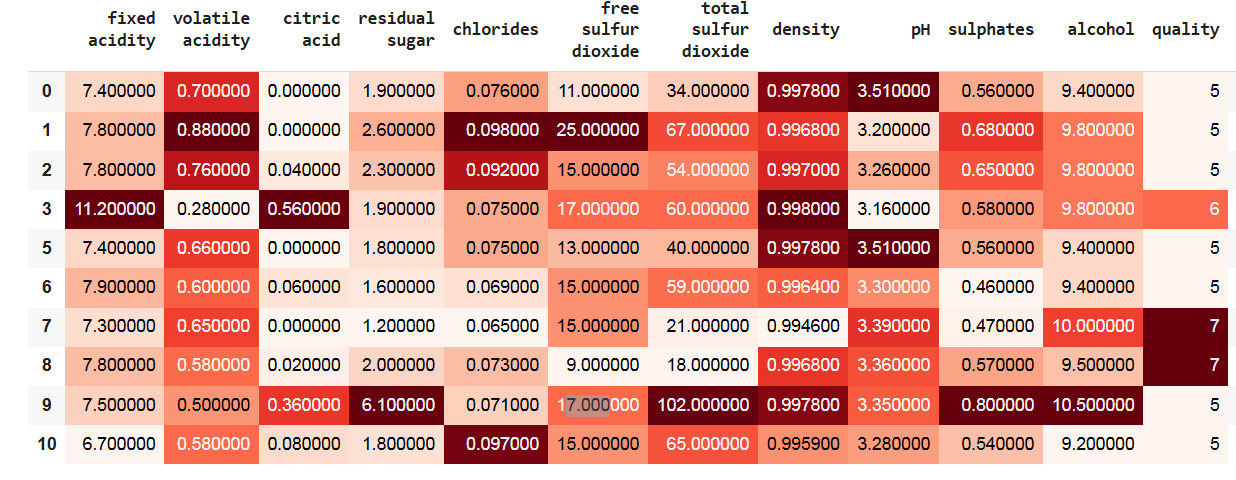
print("Số dòng trùng lặp:", df\_rwine.duplicated().sum())

duplicates = df\_rwine[df\_rwine.duplicated()]

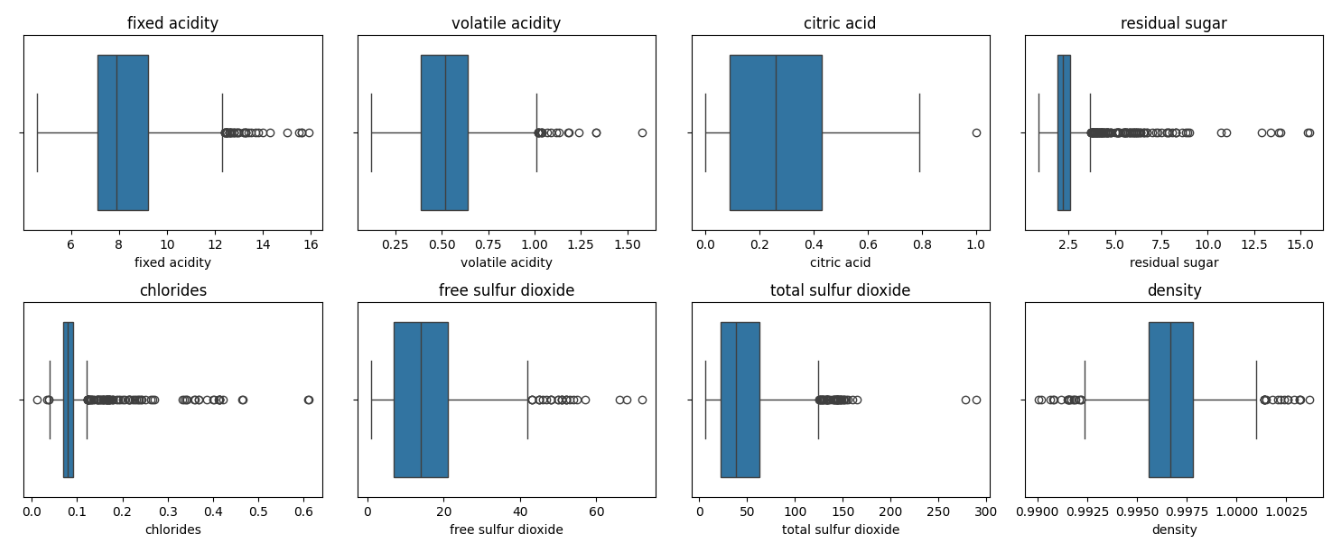
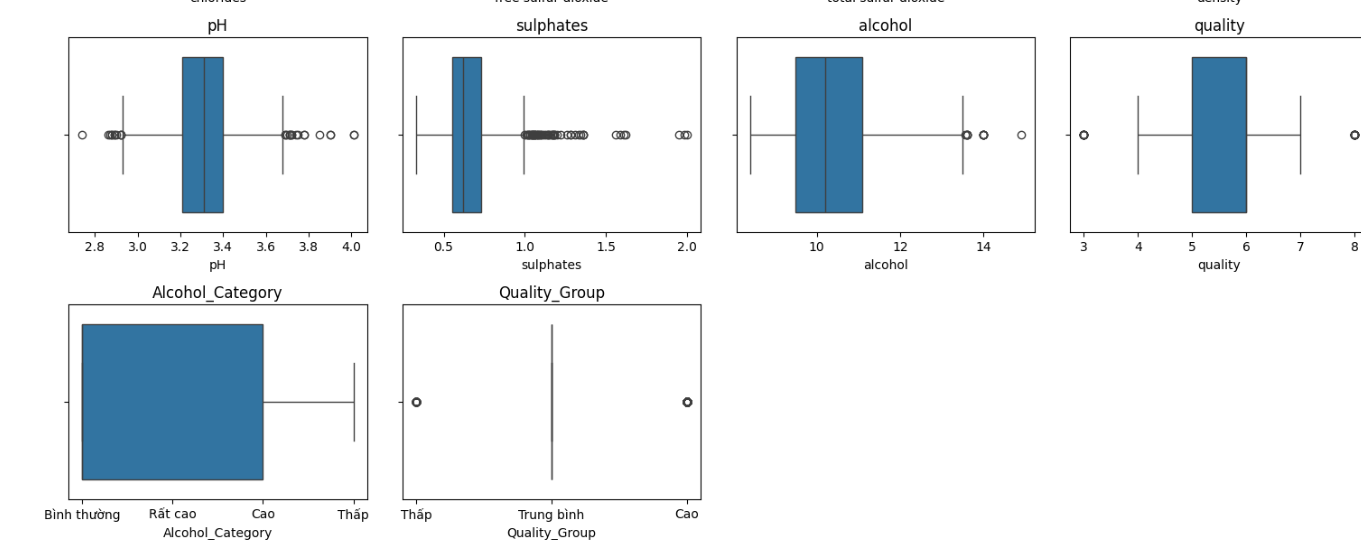
print(duplicates.head())



1. Chuẩn hóa dữ liệu:

1. Xem xét ngoại lai

1. Nhận xét:

Bài toán rượu vang đỏ là một **bài toán phân tích dữ liệu và dự đoán chất lượng sản phẩm dựa trên các chỉ tiêu hóa học**. Bộ dữ liệu chứa **1.599 mẫu rượu vang đỏ**, mỗi mẫu được mô tả bởi **11 đặc tính hóa lý** như độ axit, hàm lượng đường, nồng độ rượu, độ pH,… và một **biến mục tiêu** là điểm **chất lượng rượu (quality)** được chấm theo thang điểm 0–10 bởi chuyên gia.

Mục tiêu của bài toán là **tìm hiểu mối quan hệ giữa các đặc tính hóa học và chất lượng rượu**, từ đó có thể **xây dựng mô hình dự đoán chất lượng rượu vang** dựa trên các thông số đo được. Đây là một bài toán **học máy dạng hồi quy hoặc phân loại**, đồng thời cũng phù hợp để **phân tích thống kê mô tả và khám phá dữ liệu (EDA)**.

Nhìn chung, bài toán có ý nghĩa thực tiễn cao trong **ngành sản xuất và kiểm định rượu**, giúp **tự động hóa đánh giá chất lượng** và **tối ưu hóa quy trình sản xuất**.

## ***Bài thực hành 2: Pima Indian Diabetes***

1. Khai báo thư viện

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import math

from sklearn.model\_selection import train\_test\_split

import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

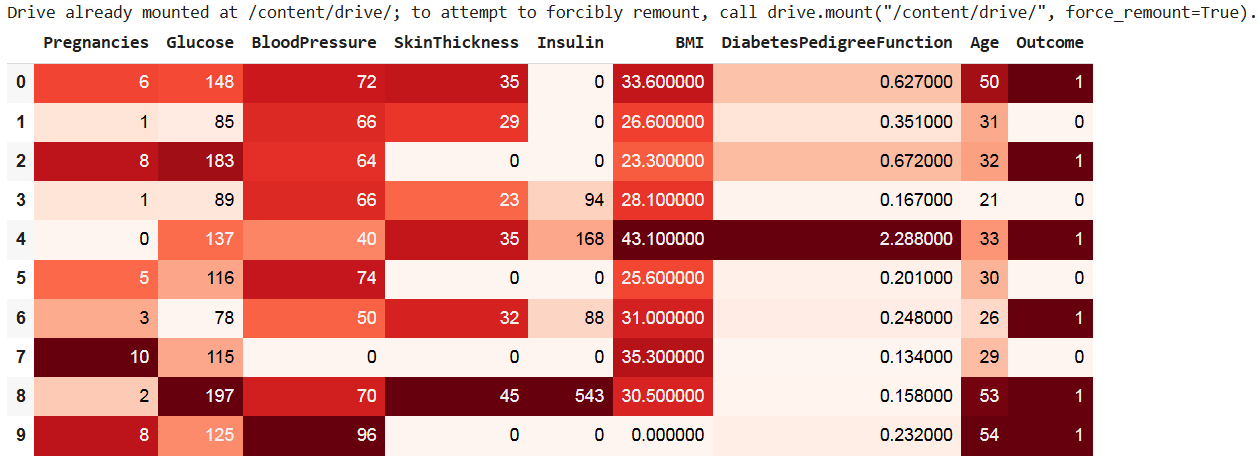
1. Nạp dữ liệu từ Drive

from google.colab import drive

drive.mount('/content/drive/')

df\_dia = pd.read\_csv('/content/drive/My Drive/diabetes.csv', sep=',')

df\_dia.head(10).style.background\_gradient(cmap="Reds")



1. Khám phá dữ liệu sơ bộ

Kiểm tra kích thước dữ liệu

print(f"Kích thước ban đầu của dữ liệu: {df\_dia.shape}")

Kiểm tra tính đầy đủ của dữ liệu

missing\_data = []

for col in df\_dia.columns:

    missing\_value = df\_dia[col].isna().sum()

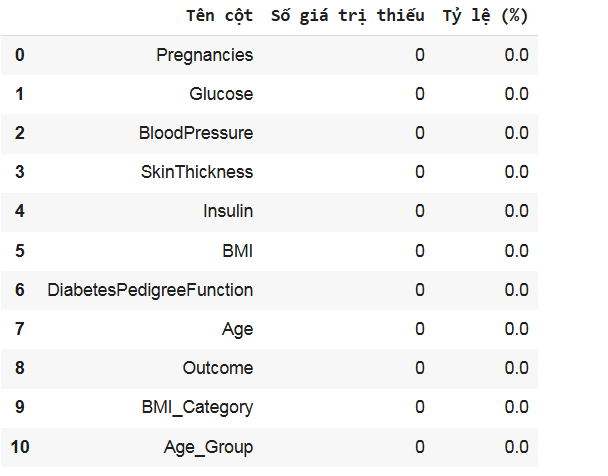
    missing\_percent = missing\_value / len(df\_dia) \* 100

    missing\_data.append([col, missing\_value, missing\_percent])

missing\_df = pd.DataFrame(missing\_data, columns=['Tên cột', 'Số giá trị thiếu', 'Tỷ lệ (%)'])

display(missing\_df)

df\_rwine.describe().T.style.background\_gradient(cmap = "Blues")



**--> Dữ liệu không bị thiếu xót**

1. Tìm kiếm trùng lặp và loại bỏ

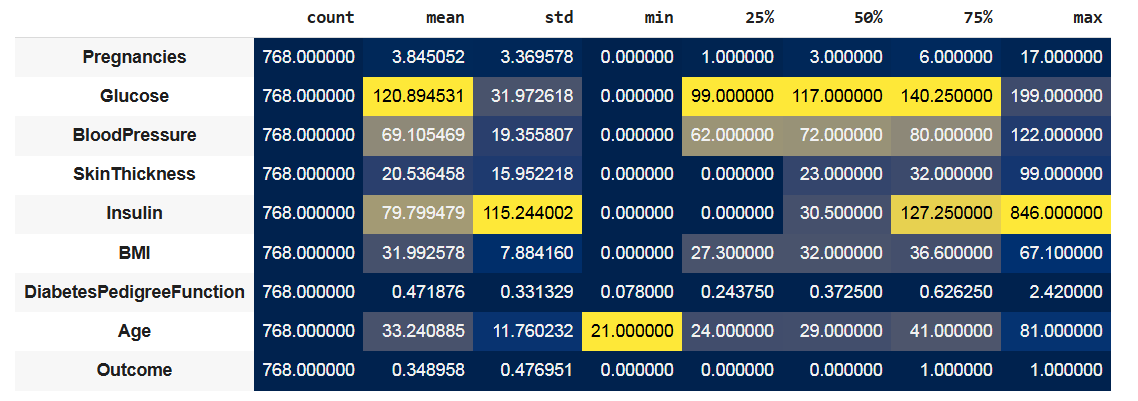
print("Số dòng trùng lặp:", df\_dia.duplicated().sum())



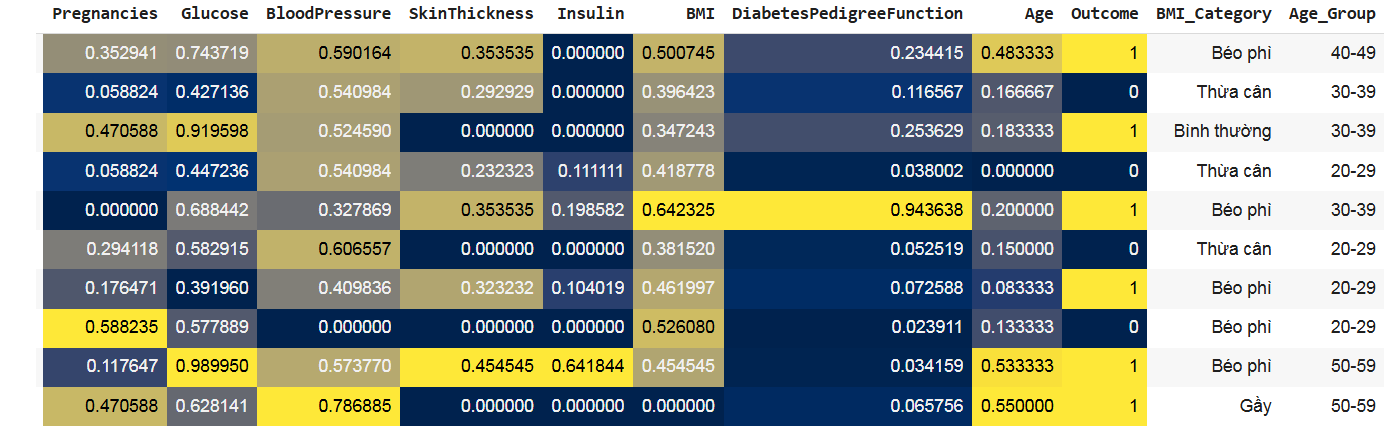
**=>Dữ liệu không bị trùng lặp bản ghi**

1. Mô tả dữ liệu:

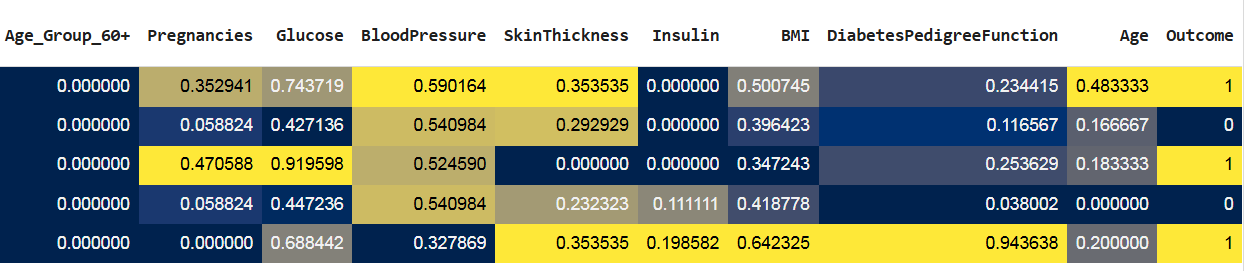
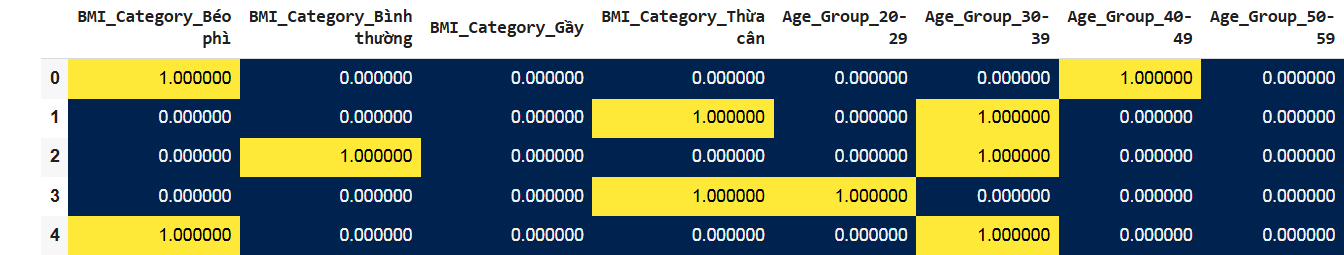
df\_dia.describe().T.style.background\_gradient(cmap="cividis")



1. Chuẩn hóa dữ liệu :



1. Mã hóa dữ liệu:



1. Chia train và test:

from sklearn.model\_selection import train\_test\_split

# Tách X (đặc trưng) và y (nhãn)

X = df\_dia\_scaled.drop('Outcome', axis=1)

y = df\_dia\_scaled['Outcome']

# Chia dữ liệu: 80% train, 20% test

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    X, y, test\_size=0.2, random\_state=42, stratify=y

)

print("Kích thước tập huấn luyện:", X\_train.shape)

print("Kích thước tập kiểm tra:", X\_test.shape)



1. Nhận Xét:

Bài toán tiểu đường là một **bài toán phân loại nhị phân (binary classification)**, trong đó mục tiêu là **dự đoán xem một người có mắc bệnh tiểu đường hay không (Outcome = 1 hoặc 0)** dựa trên các thông tin sinh lý và lâm sàng.

Bộ dữ liệu có quy mô vừa phải, các đặc trưng chủ yếu là **dạng số học**, phù hợp để áp dụng các phương pháp:

* **Phân tích thống kê mô tả** (trung bình, độ lệch chuẩn, tương quan),
* **Trực quan hóa dữ liệu** (boxplot, histogram, heatmap),
* **Mô hình học máy** như Logistic Regression, Decision Tree, Random Forest, v.v.

Dữ liệu mang tính **y học thực tiễn cao**, giúp **hỗ trợ chẩn đoán sớm** và **đánh giá nguy cơ mắc bệnh** ở bệnh nhân.

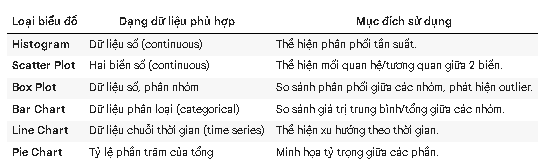
# **PHẦN II. XỬ LÝ VÀ TRỰC QUAN HÓA DỮ LIỆU**

### ***2.1. Vai trò của trực quan hóa dữ liệu trong phân tích dữ liệu***

Trực quan hóa dữ liệu (Data Visualization) giúp:

* Hiểu nhanh cấu trúc và xu hướng của dữ liệu. Phát hiện bất thường (outliers) hoặc sai lệch.
* Hỗ trợ ra quyết định dựa trên góc nhìn trực quan.
* Là công cụ quan trọng trong Exploratory Data Analysis (EDA) để nhận diện phân phối, tương quan và chất lượng dữ liệu.
* Tóm lại: trực quan hóa giúp ta “nhìn thấy” điều mà bảng số liệu không thể hiện rõ.

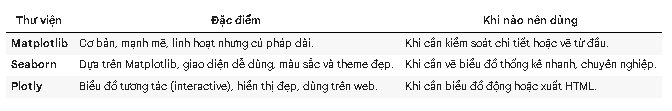
### ***2.2. Các loại biểu đồ phổ biến và khi nào nên sử dụng***



### ***2.3. Cách chọn loại biểu đồ phù hợp***

* Dữ liệu phân loại (Categorical): Bar chart, pie chart, count plot
* Dữ liệu số (Continuous): Histogram, boxplot, scatter plot
* Dữ liệu thời gian (Time series): Line chart, area chart
* Hai biến số trở lên: Scatter plot (cho 2 biến), pairplot hoặc heatmap (cho nhiều biến)

### ***2.4. So sánh các thư viện trực quan hóa trong Python***

****

### ***2.5. Nguyên tắc thiết kế biểu đồ hiệu quả***

1. Đơn giản & rõ ràng: Làm nổi bật thông tin chính.
2. Chọn màu hợp lý: Dễ nhìn, tránh gây rối mắt.
3. Chú thích rõ ràng: Tiêu đề, nhãn trục, đơn vị, chú thích.
4. Tỷ lệ đúng: Không bóp méo dữ liệu hoặc cắt trục sai lệch.
5. Nhất quán: Giữ cùng style và màu sắc giữa các biểu đồ trong báo cáo.

### ***2.6. Bài toán dự đoán giá nhà Amsterdam (Amsterdam House Price Prediction)***

#### 2.6.1 Định nghĩa vấn đề (Define problem)

* Mô tả: Bài toán dự đoán giá nhà tại Amsterdam là một bài toán học máy hồi quy (regression) nhằm ước lượng giá bán của các bất động sản dựa trên các đặc trưng như vị trí, diện tích, số phòng, và tình trạng nhà. Bộ dữ liệu bao gồm dữ liệu thực tế từ thị trường nhà ở Amsterdam (khoảng 2010-2020), được thu thập từ các nguồn công khai như Kadaster (cơ quan đăng ký bất động sản Hà Lan). Mục tiêu là xây dựng mô hình dự đoán chính xác giá nhà để hỗ trợ người mua/bán nhà, nhà đầu tư, hoặc cơ quan quy hoạch đô thị. Ví dụ: Với một căn nhà 100m² tại trung tâm, mô hình có thể dự đoán giá khoảng 500,000-700,000 EUR, giúp đánh giá rủi ro thị trường. Bài toán này phù hợp cho phân tích dữ liệu vì nó kết hợp xử lý dữ liệu thời gian thực tế, xử lý outliers (giá nhà cao bất thường), và đánh giá mô hình bằng metrics như RMSE.
* Dữ liệu đầu vào: Các đặc trưng đầu vào từ bộ dữ liệu (với ~7,000 hàng và 7 cột):
* Location/Area: Vị trí (ví dụ: postcode, neighborhood như Centrum, De Pijp).
* Size: Diện tích (m², ví dụ: 60-150m²).
* Bedrooms: Số phòng ngủ (1-5).
* Bathrooms: Số phòng tắm (1-3).
* Living Area: Diện tích sinh hoạt (m²).
* Plot Area: Diện tích đất (m², cho nhà có sân).
* Year Built: Năm xây dựng (ví dụ: 1900-2020).
* Các đặc trưng này có thể cần xử lý: encode categorical (location), xử lý missing values (~5-10% ở một số cột), và scale numerical features. Kết quả: Biến mục tiêu chính là Price (giá nhà, đơn vị: EUR), một giá trị số liên tục (ví dụ: 200,000 - 2,000,000 EUR). Đầu ra của mô hình là dự đoán giá cho một nhà mới dựa trên đầu vào. Ngoài ra, có thể mở rộng outputs như: phân loại giá (thấp/trung bình/cao) cho bài toán classification, hoặc confidence interval (ví dụ: giá dự đoán ±10%) để đánh giá độ tin cậy.

#### 2.6.2 Phân tích dữ liệu (Analyze data)

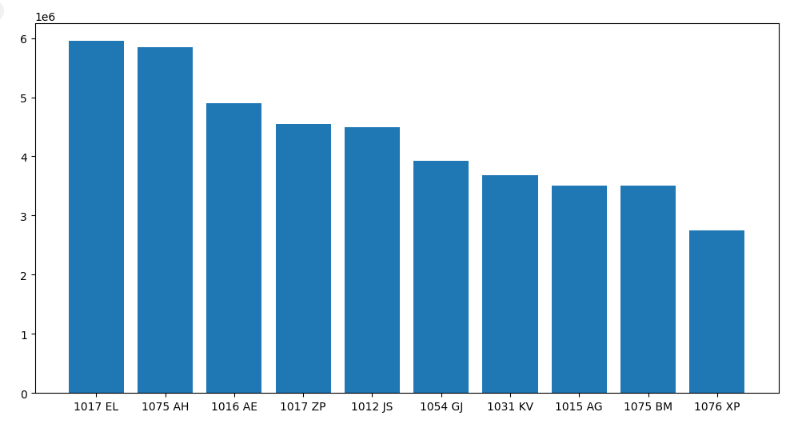


**Nhận xét:**

* Dữ liệu đã được sắp xếp giảm dần theo Price. Ví dụ: căn ở 1017 EL có giá 5.95 triệu Euro với diện tích 394 m².
* Không phải lúc nào giá cao cũng đồng nghĩa với giá/m² cao. Ví dụ:
  + Nhà ở 1012 JS có giá tổng thấp hơn (4.495 triệu Euro, diện tích chỉ 178 m²) nhưng PriceperSqm lên đến 25,252 Euro/m² → đây là căn có mật độ giá cao nhất trong top 5.
  + Ngược lại, căn 1016 AE có diện tích lớn (623 m²) giá gần 4.9 triệu nhưng PriceperSqm chỉ khoảng 7,865 Euro/m² → khá rẻ tính theo mét vuông.
* Điều này cho thấy khu vực (Zip) và diện tích ảnh hưởng mạnh đến giá/m². Các khu trung tâm thường có giá/m² cao dù diện tích nhỏ

#### 2.6.3 Trực quan hóa dữ liệu với thư viện Matplotlib

**Case 1**

****

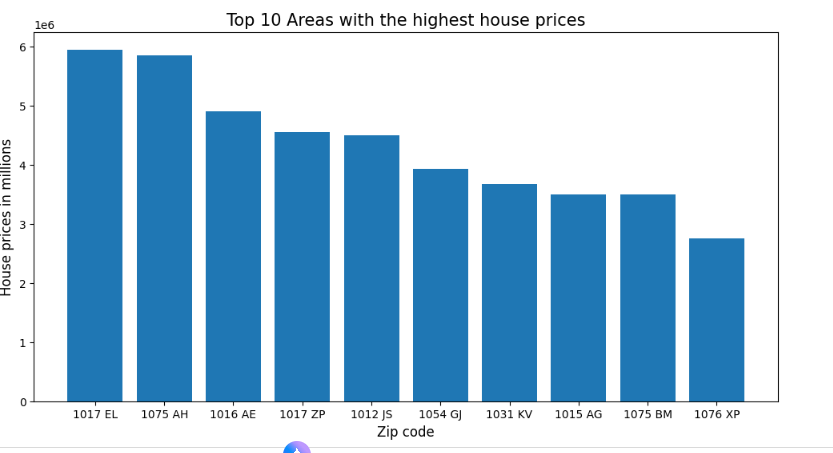
*Biểu đồ Matplotlib: Top 10 khu vực có giá cao nhất Amsterdam.*

**Nhận xét:**

* Giá nhà cao nhất
  + Zip code 1017 EL và 1075 AH có mức giá gần 6 triệu Euro – cao nhất trong danh sách.
  + Đây có thể là những khu vực trung tâm hoặc có giá trị đặc biệt (vị trí đắc địa, nhiều tiện ích).
* Chênh lệch giá giữa các khu vực
  + Sau 2 Zip code cao nhất, giá giảm dần:
    - 1016 AE ~ 4.9 triệu
    - 1017 ZP và 1012 JS ~ 4.5 triệu
    - Các Zip code tiếp theo (1054 GJ, 1031 KV, 1015 AG, 1075 BM) dao động từ 3.1 – 3.9 triệu.
  + Zip code thấp nhất trong nhóm top 10 là 1076 XP với ~2.8 triệu Euro.
* Khoảng cách giá
  + Sự chênh lệch khá rõ: từ gần 6 triệu (đắt nhất) đến khoảng 2.8 triệu (rẻ nhất trong top 10).
  + Điều này cho thấy ngay cả trong nhóm bất động sản đắt đỏ nhất, mức giá cũng phân hóa mạnh.
* Xu hướng
  + Giá giảm dần từ trái qua phải → biểu đồ minh họa tốt sự sắp xếp theo giá trị.
  + Có thể thấy một vài Zip code tập trung mức giá rất cao (1017, 1075), cho thấy khu vực này là điểm nóng thị trường bất động sản ở Amsterdam.

*Kết luận:* Biểu đồ cho thấy sự phân hóa rõ rệt giá bất động sản theo từng Zip code. Một số khu vực trung tâm (1017, 1075) có giá vượt trội, trong khi các khu vực khác dù vẫn nằm trong top 10 nhưng giá thấp hơn đáng kể.

**Case 2**

****

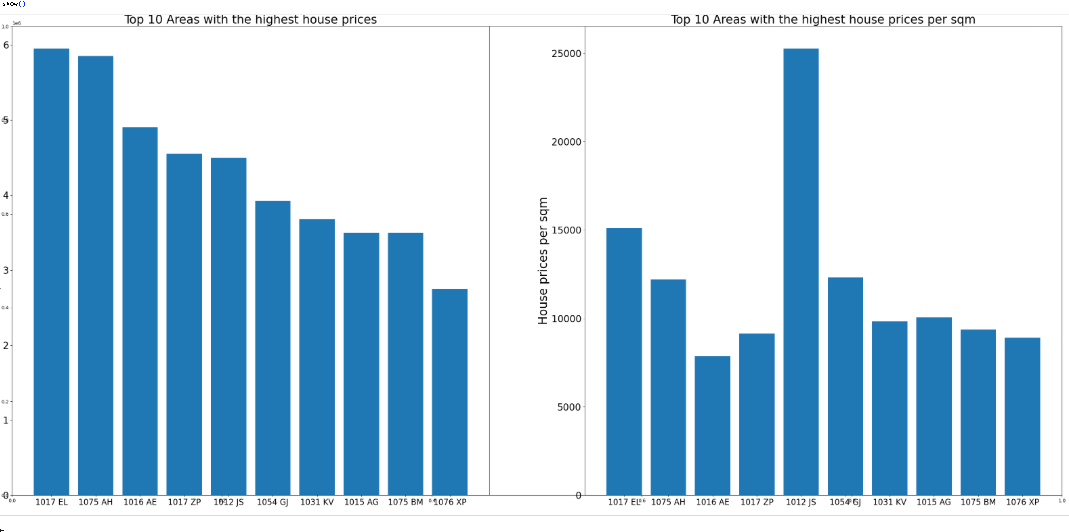
*Biểu đồ Matplotlib: Top 10 khu vực có giá cao nhất Amsterdam.*

**Nhận xét:**

* Hai khu vực dẫn đầu
  + 1017 EL và 1075 AH nổi bật nhất, giá nhà gần 6 triệu Euro → thuộc nhóm đắt đỏ bậc nhất ở Amsterdam.
  + Đây có thể là những khu trung tâm hoặc gần các khu thương mại/văn hóa quan trọng.
* Nhóm giá trung bình trong top
  + Các Zip code 1016 AE, 1017 ZP, 1012 JS có giá dao động từ 4.5 – 4.9 triệu Euro.
  + Đây là nhóm nằm ngay dưới 2 khu vực cao nhất, vẫn thuộc phân khúc cao cấp.
* Nhóm giá thấp hơn
  + 1054 GJ, 1031 KV, 1015 AG, 1075 BM có giá từ 3.5 – 4.0 triệu Euro, thấp hơn rõ so với nhóm đầu.
  + Tuy nhiên vẫn nằm trong top 10 toàn thành phố, cho thấy giá trị bất động sản khá cao.
* Khu vực thấp nhất trong top
  + 1076 XP đứng cuối với ~2.8 triệu Euro, thấp nhất trong nhóm.
  + Điều này thể hiện sự phân hóa giá rõ rệt: từ gần 6 triệu (cao nhất) đến 2.8 triệu (thấp nhất).

***Kết luận:*** Biểu đồ này cho thấy sự khác biệt mạnh mẽ về giá nhà giữa các khu vực Amsterdam. Một số Zip code trung tâm (1017 EL, 1075 AH) vượt trội rõ ràng, trong khi những khu khác vẫn đắt đỏ nhưng không đạt mức “siêu cao”. Điều này gợi ý rằng vị trí địa lý (Zip code) là yếu tố quyết định hàng đầu trong việc định giá bất động sản**.**

**Case 3**

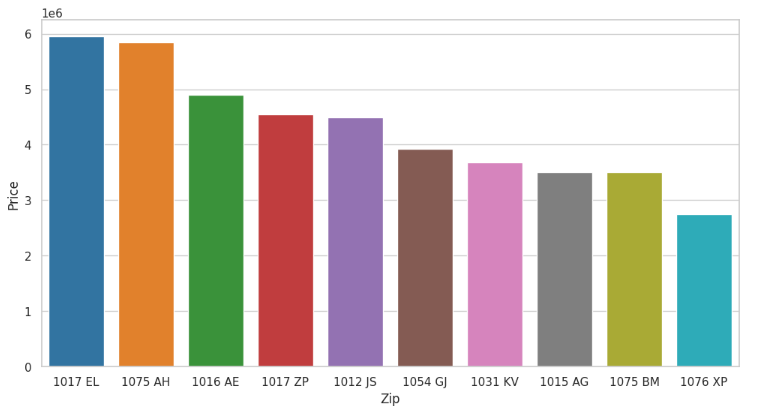
****

*Biểu đồ Matplotlib: So sánh 2 case 1 và case 2 về khu vực có giá cao nhất Amsterdam.*

**Nhận xét:**

* Biểu đồ bên trái – Top 10 giá nhà cao nhất (tổng giá)
  + Khu vực 1017 EL và 1075 AH dẫn đầu với gần 6 triệu Euro.
  + Các khu tiếp theo (1016 AE, 1017 ZP, 1012 JS) dao động từ ~4.5 – 4.9 triệu.
  + Nhóm cuối (1054 GJ, 1031 KV, 1015 AG, 1075 BM, 1076 XP) có giá từ ~2.8 – 3.9 triệu.
* *Điều này cho thấy sự phân hóa mạnh về tổng giá nhà, tập trung cao ở một vài khu vực trung tâm.*
* Biểu đồ bên phải – Top 10 giá nhà tính theo mét vuông (Price per sqm)
  + Bất ngờ nhất: 1012 JS vọt lên đứng đầu với hơn 25,000 €/m², dù tổng giá của nó không cao nhất.
  + Các khu như 1017 EL, 1075 AH có tổng giá cao, nhưng giá/m² thấp hơn nhiều (~12k–15k €/m²).
  + Một số khu khác như 1016 AE, 1017 ZP lại có giá/m² chỉ ~8k–9k €/m², thấp so với các khu khác.
* *Điều này phản ánh rằng những căn có diện tích nhỏ ở vị trí trung tâm thường có giá/m² cao vượt trội.*
* So sánh hai biểu đồ
  + Tổng giá cao ≠ Giá/m² cao.
  + Ví dụ:
    - 1017 EL: cao nhất về tổng giá nhưng không phải cao nhất về giá/m².
    - 1012 JS: tổng giá chỉ trung bình, nhưng giá/m² lại cao nhất.
* *Điều này cho thấy phải phân tích cả hai khía cạnh thì mới có cái nhìn chính xác về thị trường.*

#### 2.6.4. Trực quan hóa dữ liệu với thư viện Seaborn

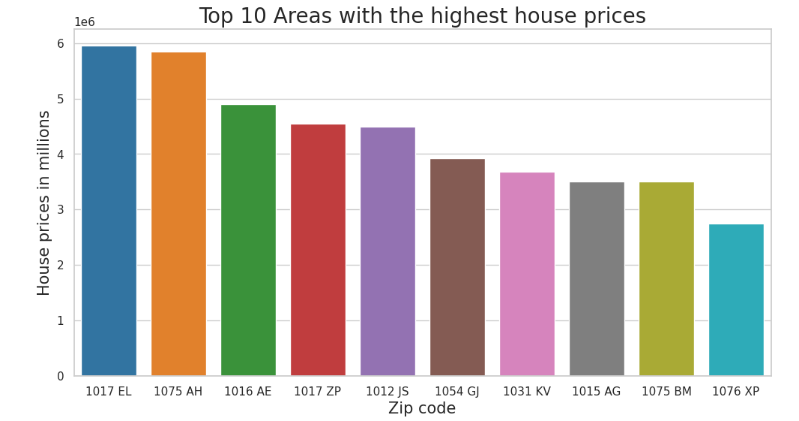


*Biểu đồ Seaborn: Top 10 khu vực có giá cao nhất Amsterdam.*

**Nhận xét:**

* Xu hướng chung:
  + Các khu vực 1017 EL và 1075 AH có mức giá nhà cao nhất (~5.8–5.9 triệu).
  + Các khu vực như 1076 XP có giá thấp hơn rõ rệt (~2.8 triệu).
  + Có sự chênh lệch khá lớn giữa Zip code cao nhất và thấp nhất trong nhóm top 10.
* So sánh giữa các khu vực:
  + Nhóm 4 khu vực đầu (1017 EL, 1075 AH, 1016 AE, 1017 ZP) có mức giá nổi trội so với phần còn lại.
  + Từ 1054 GJ đến 1076 XP giá giảm dần đều, không có đột biến mạnh.
* Trực quan:
  + Biểu đồ dùng Seaborn với palette="tab10", giúp các cột khác màu, dễ so sánh.
  + Có grid nền nhẹ từ sns.set(style="whitegrid"), giúp đọc giá trị trực quan hơn.
  + Tuy nhiên, nhãn giá hiện vẫn để mặc định (dùng số nguyên rất lớn, đơn vị triệu/millions chưa được format rõ ràng).

**Case 2 : Advanced 1**

*****Biểu đồ Seaborn: Top 10 khu vực có giá cao nhất Amsterdam.*

**Nhận xét:**

* Biểu đồ thể hiện rõ 10 khu vực có giá nhà cao nhất.
* Khu vực 1017 EL và 1075 AH có mức giá cao nhất, xấp xỉ 6 triệu euro.
* Các khu vực khác giảm dần theo sau, thấp nhất là 1076 XP khoảng 2.7 triệu euro.
* Ưu điểm: dễ nhìn, tập trung vào một chỉ số (giá nhà).
* Hạn chế: chỉ phản ánh tổng giá nhà, chưa cho thấy hiệu quả tính theo diện tích (sqm).

**Case 3 :View multiple perspectives at once**

****

*Biểu đồ Seaborn: So sánh case 1 và case 2 về khu vực có giá cao nhất Amsterdam.*

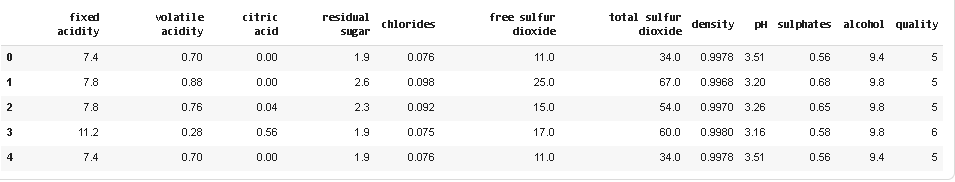
**Nhận xét:**

* Biểu đồ bên trái: giống hình 1, thể hiện giá nhà trung bình ở top 10 khu vực.
* Biểu đồ bên phải: cho thấy giá/m², tức là chi phí thực sự cho diện tích sử dụng.
* Điểm nổi bật:
  + Khu vực 1012 JS tuy không có giá nhà cao nhất (chỉ tầm 4.5 triệu) nhưng lại có giá/m² cao vượt trội (≈ 25,000/m²) → có thể là khu có diện tích nhỏ nhưng vị trí rất đắt đỏ.
  + Các khu vực như 1017 EL vẫn giữ giá/m² cao (≈ 15,000/m²), chứng tỏ vừa đắt tổng thể vừa đắt trên diện tích.
  + Một số khu (ví dụ 1016 AE, 1076 XP) có giá nhà cao nhưng giá/m² lại thấp hơn tương đối → có thể do diện tích rộng.

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

##### 2.6.5.1. Định nghĩa vấn đề (Define problem)

* Mô tả: Bài toán phân loại chất lượng rượu đỏ là một bài toán học máy supervised classification nhằm dự đoán chất lượng rượu (thang điểm 0-10) dựa trên các đặc trưng hóa học như nồng độ cồn, độ acid, và đường residual. Bộ dữ liệu bao gồm thông tin từ các mẫu rượu vang đỏ Bồ Đào Nha (Vinho Verde), thu thập từ 2009. Mục tiêu là xây dựng mô hình dự đoán chất lượng (thường được chia thành 3-5 lớp: thấp, trung bình, cao) để hỗ trợ nhà sản xuất tối ưu quy trình làm rượu hoặc người tiêu dùng lựa chọn sản phẩm. Ví dụ: Một chai rượu với cồn 12% và acid 0.5 g/L có thể được dự đoán chất lượng "trung bình" (5-6). Bài toán này phù hợp cho phân tích dữ liệu vì yêu cầu xử lý dữ liệu số, cân bằng lớp (imbalanced class), và đánh giá bằng metrics như accuracy, F1-score.
* Dữ liệu đầu vào: Các đặc trưng đầu vào từ bộ dữ liệu (1,599 hàng, 12 cột):
  + Fixed Acidity: Độ acid cố định (g/L, ví dụ: 4.6-15.9).
  + Volatile Acidity: Độ acid dễ bay hơi (g/L, ví dụ: 0.12-1.58).
  + Citric Acid: Nồng độ acid citric (g/L, ví dụ: 0-1.0).
  + Residual Sugar: Đường residual (g/L, ví dụ: 0.9-15.5).
  + Chlorides: Nồng độ muối (g/L, ví dụ: 0.012-0.611).
  + Free Sulfur Dioxide: Lưu huỳnh dioxide tự do (mg/L, ví dụ: 1-72).
  + Total Sulfur Dioxide: Tổng lưu huỳnh dioxide (mg/L, ví dụ: 6-289).
  + Density: Mật độ (g/cm³, ví dụ: 0.99007-1.00369).
  + pH: Độ pH (ví dụ: 2.74-4.01).
  + Sulphates: Nồng độ sulphates (g/L, ví dụ: 0.33-2.0).
  + Alcohol: Nồng độ cồn (% vol, ví dụ: 8.4-14.9).
  + Các đặc trưng này cần xử lý: scale (chuẩn hóa), kiểm tra correlation (cồn vs quality ~0.5), và xử lý outliers (ví dụ: density > 1.0).
* Kết quả:
  + Biến mục tiêu là Quality (chất lượng rượu), một giá trị nguyên từ 0-10 (thực tế tập trung 3-8). Đầu ra của mô hình là nhãn dự đoán (ví dụ: 5, 6, 7) hoặc xác suất thuộc từng lớp nếu dùng mô hình như Logistic Regression. Có thể nhóm chất lượng thành 3 lớp:
    - Thấp (3-4), Trung bình (5-6), Cao (7-8) để giảm vấn đề imbalanced.
    - Đầu ra phụ: Confusion matrix, ROC curve để đánh giá hiệu suất.



*Hình 2.6.1. Thống kê dữ liệu 5 dòng đầu tập dữ liệu về phân loại chất lượng rượu đỏ.*



*Hình 2.6.2. Thống kê dữ liệu 5 dòng đầu tập dữ liệu về phân loại chất lượng rượu đỏ.*

##### 2.6.5.2. Kiểm tra tính toàn vẹn của dữ liệu

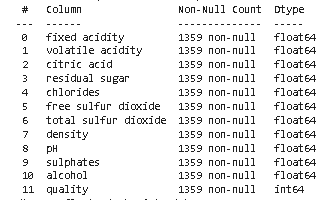
**Tính toàn vẹn dữ liệu:**

* Có giá trị Null: False
* Có giá trị NaN: False
* Số dòng trùng: 240

**Nhận xét:**

* Các giá trị Null và NaN là không có
* Dữ liệu bị trùng là: 244 dòng

**Xử lý trùng lặp:**

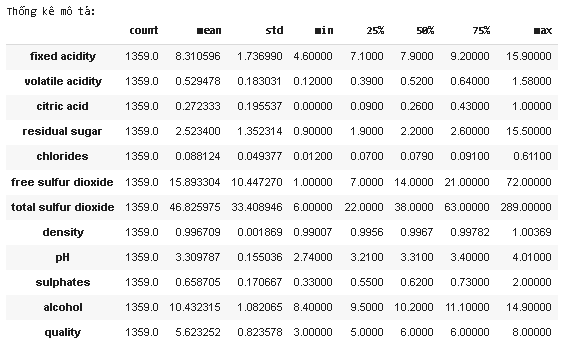
****

**Nhận xét:**

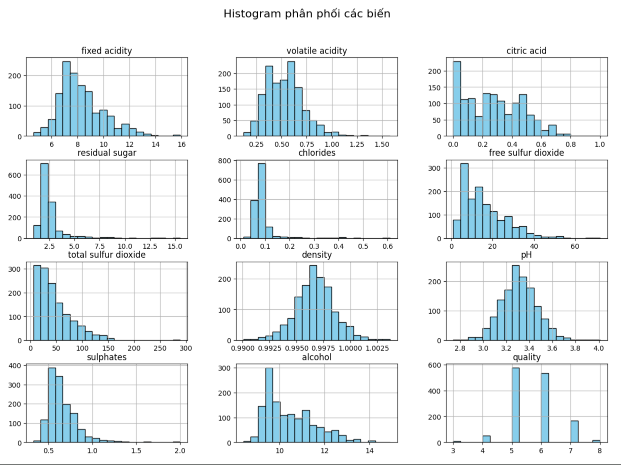
* Các xử lý các dòng bị trùng dữ liệu
* Sau khi làm sạch: dataset còn khoảng 1359 dòng, 12 cột

##### 2.6.5.2. Hiểu phân phối dữ liệu (Univariate Analysis)

* **Mục tiêu:** xem từng biến (cột) có phân phối thế nào.
  + Dùng .describe() để thống kê cơ bản (mean, std, min, max, quartiles).
  + Vẽ histogram, boxplot, kde để nhìn phân phối.

****

*Hình 2.5.2.1: Thống kê dữ liệu trên tập dữ liệu về phân loại chất lượng rượu đỏ.*

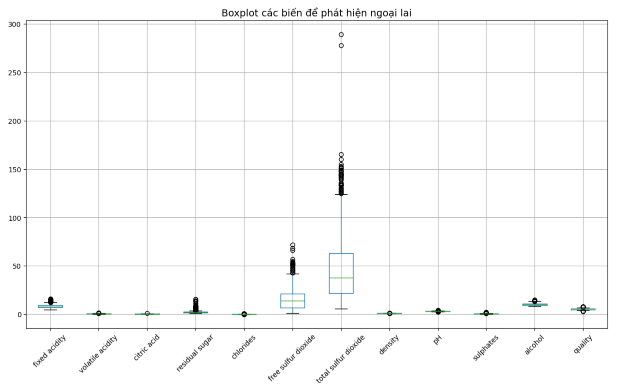
****

*Biểu đồ Histogram : Phân phối các biến về phân loại chất lượng rượu đỏ.*

**Nhận xét:**

* Các biến như *fixed acidity, volatile acidity, citric acid, alcohol…* phân phối không chuẩn, đa số lệch trái/phải.
* *residual sugar, free sulfur dioxide, total sulfur dioxide* có nhiều giá trị nhỏ, một số giá trị rất lớn → tiềm ẩn ngoại lai (outliers).
* Biến *quality* tập trung chủ yếu ở mức 5–6, ít mẫu ở mức thấp (3–4) và cao (7–8).

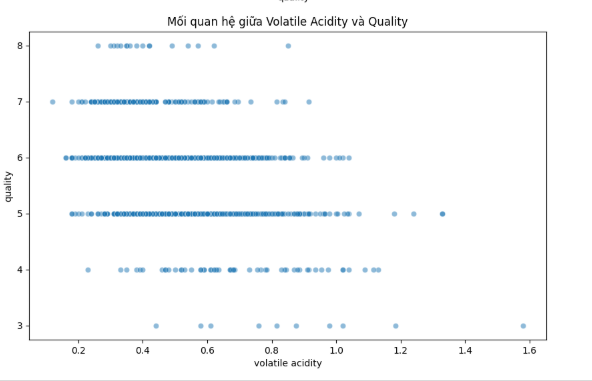
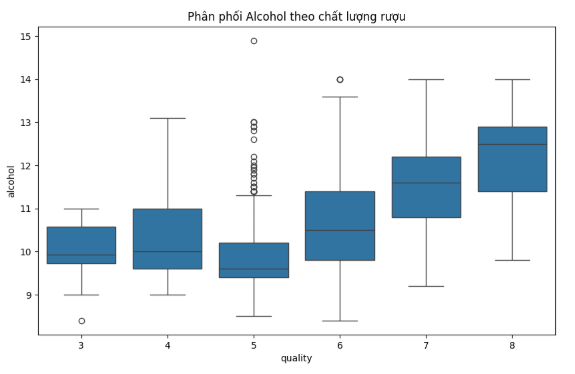
**Boxplot để thấy outlier từng biến**

****

**Nhận xét:**

* Nhiều biến có outliers rõ ràng (residual sugar, free sulfur dioxide, total sulfur dioxide, alcohol).
* *density, pH* ít ngoại lai hơn, giá trị tập trung hơn.

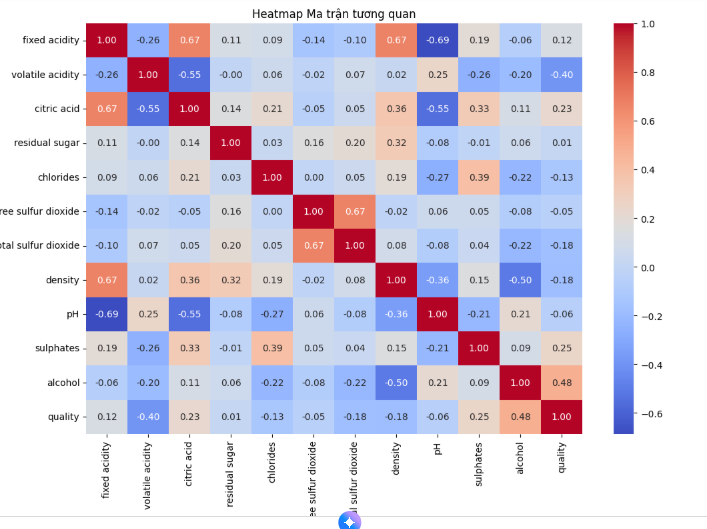
##### 2.6.5.3. Phân tích mối quan hệ giữa các biến (Bivariate Analysis)



**Nhận xét:** Boxplot Alcohol theo Quality

* Khi quality tăng, nồng độ alcohol cũng tăng.
* Ví dụ: rượu chất lượng cao (7–8) thường có alcohol cao hơn so với rượu chất lượng thấp (3–5).

##### 2.6.5.4. Ma trận tương quan (Correlation)



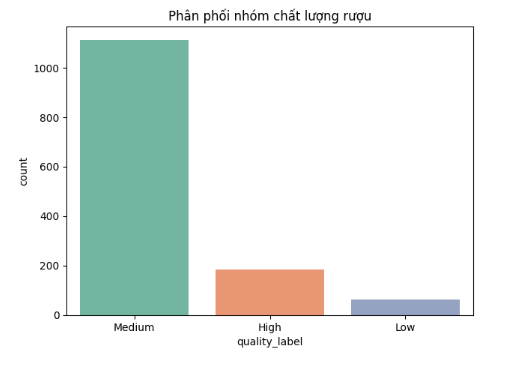
*Biểu đồ Heatmap: Ma trận tương quan về phân loại chất lượng rượu đỏ.*

**Nhận xét:**

* ***alcohol*** có tương quan dương khá mạnh với quality (0.48).
* ***volatile acidity*** có tương quan âm với quality (-0.40).
* Các biến khác (citric acid, sulphates) cũng có mức tương quan dương nhẹ với quality.
* Một số biến có tương quan cao với nhau (ví dụ fixed acidity – density – citric acid) → có thể gây đa cộng tuyến trong hồi quy.

##### 2.6.5.5. Feature Engineering (ví dụ tạo thêm biến)

* Nhóm chất lượng thành 3 mức: thấp, trung bình, cao
* Tần suất từng nhóm chất lượng:
* quality\_label
* Medium 1112
* High 184
* Low 63



***Nhận xét:***

* Bất cân bằng dữ liệu: phần lớn là Medium, rất ít Low và High.
* Nếu phân loại (Decision Tree, Random Forest, …) thì cần resampling (SMOTE, undersampling) để cân bằng dữ liệu.

**Kết luận & Hướng phân tích tiếp theo**:

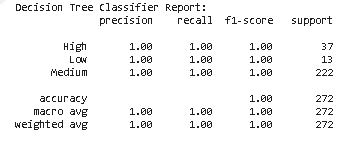
1. Dự báo quality (Regression)

* Biến mục tiêu: quality (dạng số).
* Quan trọng nhất: alcohol, volatile acidity, sulphates, citric acid.
* Cần xử lý: ngoại lai + chuẩn hóa dữ liệu + loại bớt biến có tương quan cao.

1. Phân loại quality (Decision Tree / Random Forest)

* Gom quality thành nhóm (Low–Medium–High).
* Cần xử lý mất cân bằng dữ liệu.
* Feature quan trọng: alcohol, volatile acidity, sulphates.

##### 2.6.5.6. Pipeline xử lý & phân tích dữ liệu rượu vang



**Nhận xét:**

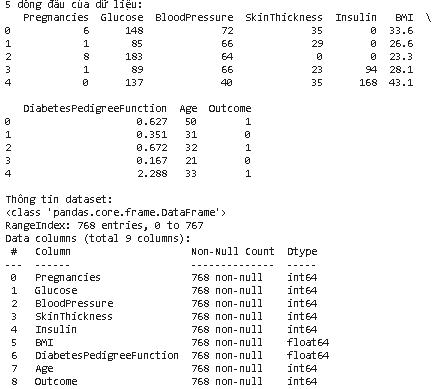
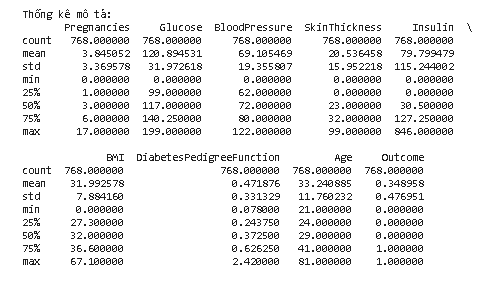
* Mô hình Decision Tree đạt độ chính xác tuyệt đối (**Accuracy = 1.00**).
* Tất cả các lớp đều có **Precision, Recall, F1-score = 1.00**.
* Điều này cho thấy mô hình **đang overfitting (quá khớp)**, có thể do huấn luyện và kiểm tra trên cùng tập dữ liệu.

#### 2.6.6. Thực hiện trực quan hóa dữ liệu trên tập dữ liệu về bệnh tiểu đường

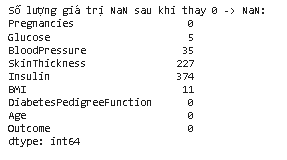
##### 2.6.6.1. Định nghĩa vấn đề (Define problem)

* Mô tả:
  + Bệnh tiểu đường (Diabetes) là một trong những bệnh mãn tính phổ biến trên toàn cầu, gây ảnh hưởng nghiêm trọng đến sức khỏe. Việc khai thác dữ liệu y tế nhằm trực quan hóa, phân tích và xây dựng mô hình dự đoán giúp hỗ trợ bác sĩ chẩn đoán sớm nguy cơ mắc bệnh tiểu đường.
* Dữ liệu đầu vào: Tập dữ liệu Pima Indians Diabetes Dataset từ Kaggle, gồm 768 mẫu và 9 thuộc tính:
  + Pregnancies: số lần mang thai
  + Glucose: nồng độ đường huyết
  + BloodPressure: huyết áp
  + SkinThickness: độ dày da
  + Insulin: nồng độ insulin
  + BMI: chỉ số khối cơ thể
  + DiabetesPedigreeFunction: chỉ số phả hệ về bệnh tiểu đường
  + Age: tuổi
  + Outcome: nhãn kết quả (0 = không tiểu đường, 1 = tiểu đường)
* Kết quả:
  + Tiền xử lý dữ liệu: thay thế giá trị 0 bằng NaN và xử lý bằng kỹ thuật Imputation.
  + Trực quan hóa dữ liệu: biểu đồ phân phối, boxplot, heatmap, pairplot… để hiểu rõ đặc trưng và mối quan hệ giữa các biến.
  + Xây dựng mô hình dự đoán bệnh tiểu đường: Logistic Regression, Decision Tree,…
  + Đánh giá mô hình bằng Classification Report, Confusion Matrix, ROC Curve, ROC AUC.

##### 2.6.6.2. Load dataset

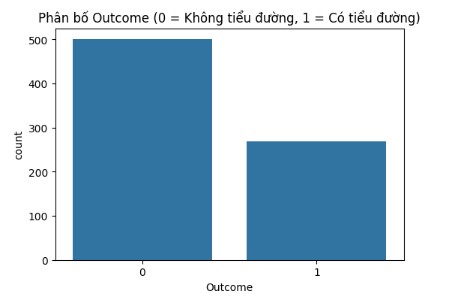


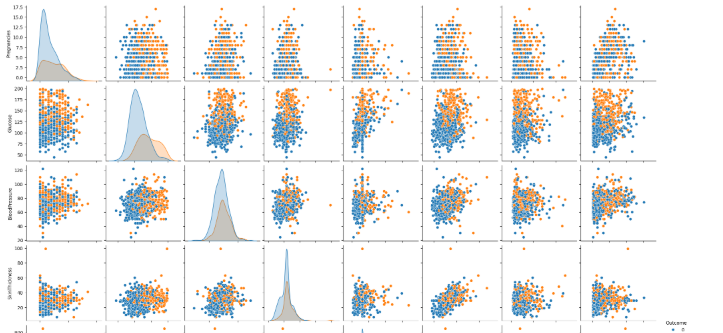
##### 2.6.6.3. Xử lý dữ liệu



##### 2.6.6.4. Trực quan hóa dữ liệu

**Outcome phân bố**





*Biều đồ Pairplot: Phân tích dữ liệu về phân loại chất lượng rượu đỏ.*

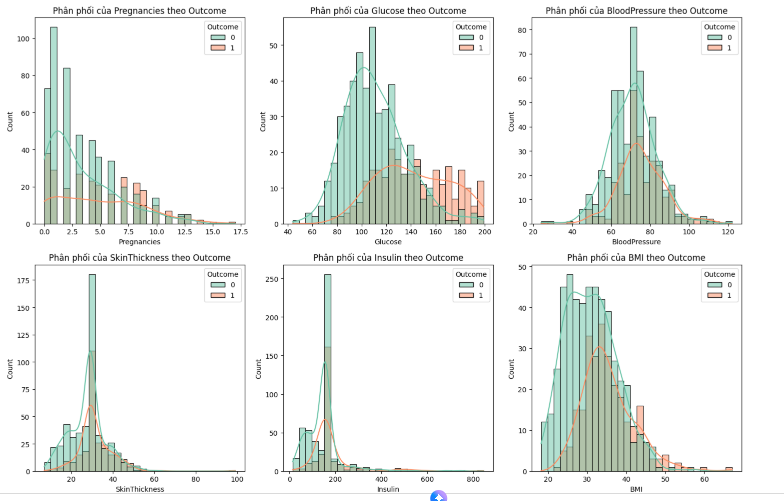
**Nhận xét:**

* **Glucose** cho thấy sự khác biệt rõ rệt giữa hai nhóm Outcome (0 và 1) — nhóm 1 có Glucose cao hơn.
* **BMI** và **Age** cũng cao hơn ở nhóm Outcome = 1.
* Các biến như **BloodPressure** và **SkinThickness** có sự chồng lấp giữa hai nhóm.

**Kết luận:**

* Các biến **Glucose**, **BMI**, và **Age** là **những đặc trưng quan trọng** giúp phân biệt giữa hai nhóm Outcome.

**Histogram phân phối theo Outcome**

****

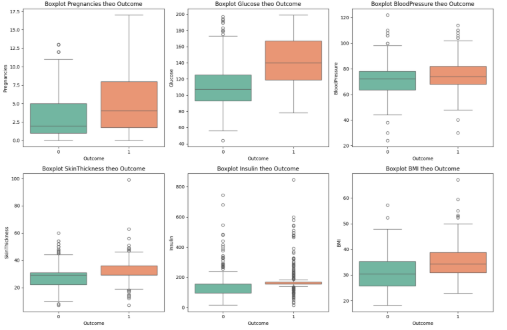
**Nhận xét:**

* Glucose: nhóm Outcome = 1 có giá trị cao hơn đáng kể.
* BMI: cao hơn ở nhóm Outcome = 1.
* Age: người mắc tiểu đường có xu hướng lớn tuổi hơn.
* Pregnancies: số lần mang thai nhiều hơn tương ứng với khả năng mắc tiểu đường cao hơn.
* Insulin, SkinThickness, BloodPressure: ít khác biệt rõ rệt giữa hai nhóm.

Kết luận:

* Các yếu tố Glucose, BMI, Age, Pregnancies có ảnh hưởng mạnh nhất đến khả năng mắc bệnh tiểu đường.

**Boxplot để so sánh phân phối**

****

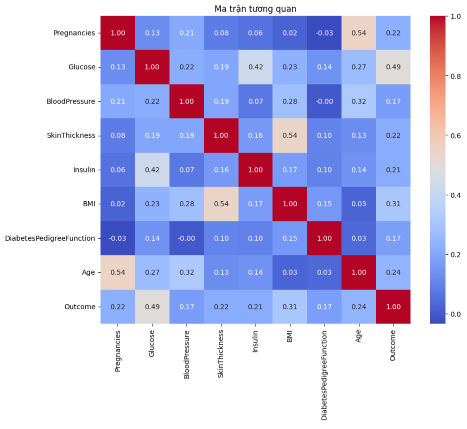
**Nhận xét:**

* Glucose, BMI, Age: trung vị cao hơn ở nhóm Outcome = 1.
* Pregnancies: nhóm Outcome = 1 có nhiều lần mang thai hơn.
* BloodPressure, SkinThickness, Insulin: sự khác biệt không rõ, có nhiều outlier.

Kết luận:

* Các biến có độ khác biệt rõ nhất giữa 2 nhóm Outcome là Glucose, BMI, Age, Pregnancies.

**Heatmap correlation**

****

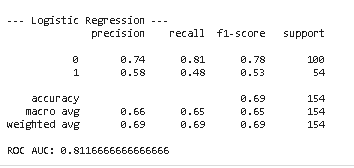
**Nhận xét:**

* Glucose ↔ Outcome: hệ số tương quan 0.49 → tương quan trung bình, đáng chú ý.
* Age ↔ Pregnancies: 0.54 → người lớn tuổi thường có nhiều lần mang thai hơn.
* BMI ↔ SkinThickness: 0.54 → người có lớp da dày thường có BMI cao.
* Các biến khác có tương quan yếu (< 0.3).

Kết luận:

* Biến Glucose là yếu tố có tương quan mạnh nhất với Outcome.
* Các biến Age, BMI, Pregnancies có ảnh hưởng hỗ trợ.

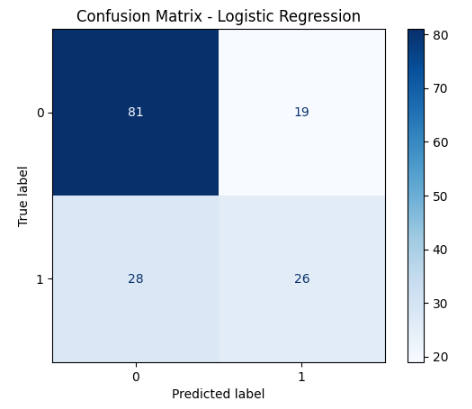
##### 2.6.6.5. Hồi quy logistic (Logistic Regression)



**Nhận xét:**

* **Độ chính xác (Accuracy)**: Mô hình đạt độ chính xác tổng thể là **69%**, một kết quả khá tốt.
* **Hiệu suất trên Lớp 1 ("Có bệnh")**: Hiệu suất của mô hình trên lớp này khá yếu.
  + **Recall = 0.48**: Đây là một con số đáng báo động. Mô hình chỉ phát hiện được **48%** các trường hợp "có bệnh", tức là đã **bỏ sót tới 28/54 ca (hơn 50%)**.
  + **Precision = 0.58**: Khi dự đoán "có bệnh", độ tin cậy chỉ ở mức trung bình.
  + **F1-Score = 0.53**: Chỉ số này phản ánh hiệu suất không tốt trên lớp dương tính.
* **Chỉ số ROC AUC**: Mặc dù các chỉ số khác không cao, **ROC AUC = 0.81** cho thấy mô hình có khả năng phân biệt tốt giữa hai lớp. Có thể hiệu suất sẽ được cải thiện nếu tinh chỉnh ngưỡng (threshold) phân loại

**Confusion Matrix**

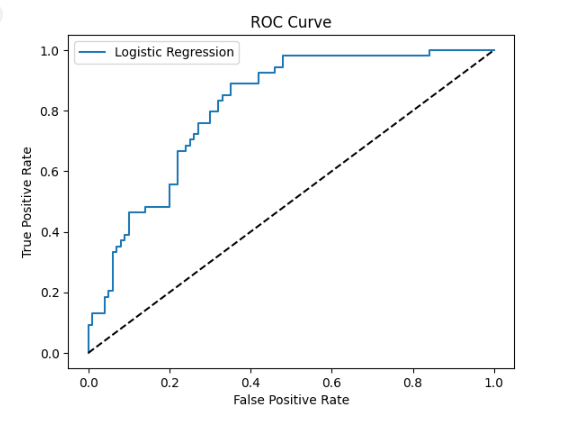
****

**Nhận xét:**

Ma trận nhầm lẫn (Confusion Matrix):

* 26 True Positives: Dự đoán đúng 26 ca có bệnh.
* 28 False Negatives: Bỏ sót đến 28 ca có bệnh. Tỷ lệ bỏ sót này quá cao đối với một bài toán y tế.

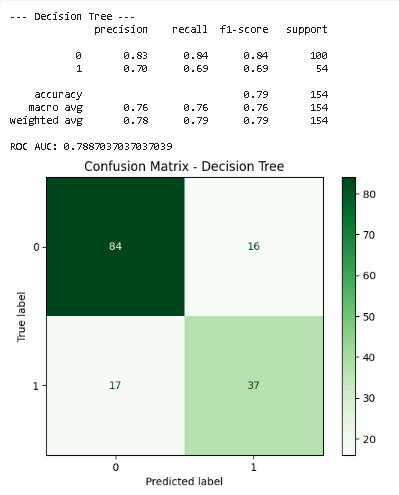
**ROC Curve**

****

**Nhận xét:**

Đường cong ROC: Mặc dù hiệu suất tổng thể không cao, đường cong ROC và chỉ số AUC = 0.81 cho thấy mô hình có khả năng phân tách tốt giữa hai lớp. Điều này có nghĩa là nếu điều chỉnh ngưỡng quyết định (threshold), có thể hiệu suất của mô hình sẽ được cải thiện, nhưng với ngưỡng mặc định (0.5), nó hoạt động không hiệu quả.

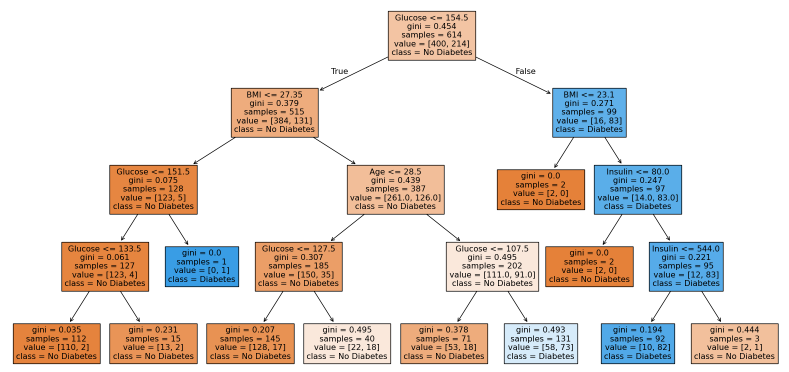
##### 2.6.6.6. Phân loại cây quyết định (Decision Tree Classifier)



**Nhận xét:**

* Hiệu suất tổng thể tốt: Với độ chính xác (Accuracy) là 79%, mô hình này có khả năng phân loại khá tốt trên tập dữ liệu.
* Dự đoán tốt lớp "Có bệnh" (Lớp 1):
  + Recall = 0.69: Mô hình phát hiện đúng 37 trong số 54 trường hợp thực sự có bệnh.
  + Precision = 0.70: Khi mô hình dự đoán một người có bệnh, nó đúng trong 70% trường hợp.
  + F1-Score = 0.69: Đây là một điểm số khá tốt, cho thấy sự cân bằng giữa Precision và Recall.
* Ma trận nhầm lẫn (Confusion Matrix):
  + 37 True Positives: Dự đoán đúng 37 ca có bệnh.
  + 17 False Negatives: Bỏ sót 17 ca có bệnh (đây là loại lỗi nguy hiểm trong chẩn đoán y tế).
  + 84 True Negatives: Dự đoán đúng 84 ca không có bệnh.
  + 16 False Positives: Dự đoán nhầm 16 ca không bệnh thành có bệnh.

**Vẽ cây quyết định**

****

**Nhận xét:**

* Được chia chủ yếu dựa vào Glucose, BMI, Age, và Insulin.
* Gini giảm dần thể hiện phân chia càng sâu thì dữ liệu càng thuần khiết hơn.
* Gốc cây chia tại Glucose <= 154.5, cho thấy Glucose là yếu tố quyết định mạnh nhất.
* Phần lớn mẫu không mắc bệnh tiểu đường (No Diabetes), điều này dễ thấy khi nhiều node lá có class là “No Diabetes”.

Kết luận:

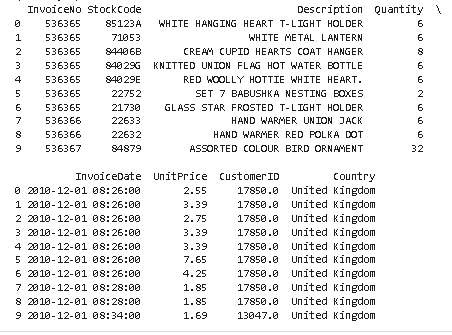
Biểu đồ cho thấy mô hình rất dễ diễn giải. Các đặc trưng quan trọng nhất để ra quyết định lần lượt là Glucose, BMI, Age, và Insulin. Ví dụ, quyết định đầu tiên của mô hình là dựa trên ngưỡng Glucose <= 154.5.

#### 2.6.7. Thực hiện EDA trên tập dữ liệu mua sắm tại siêu thị

##### 2.6.7.1 Định nghĩa vấn đề (Define problem)

* Mô tả: Trong lĩnh vực bán lẻ trực tuyến, việc phân tích dữ liệu mua sắm của khách hàng giúp doanh nghiệp hiểu rõ hơn về hành vi tiêu dùng, từ đó tối ưu hóa chiến lược kinh doanh. Thực hiện phân tích khám phá dữ liệu (EDA) cho bộ dữ liệu Online Retail nhằm phát hiện các xu hướng, mối quan hệ và đặc trưng quan trọng. Đồng thời, xây dựng mô hình dự đoán để hỗ trợ ra quyết định, ví dụ như dự đoán giá trị hóa đơn hoặc phân loại nhóm khách hàng.
* Dữ liệu đầu vào:
  + Bộ dữ liệu Online Retail Dataset từ Kaggle, chứa dữ liệu giao dịch của một cửa hàng trực tuyến Anh quốc trong năm 2010–2011, gồm các thuộc tính:
    - InvoiceNo: mã hóa đơn
    - StockCode: mã sản phẩm
    - Description: mô tả sản phẩm
    - Quantity: số lượng mua
    - InvoiceDate: ngày giao dịch
    - UnitPrice: giá sản phẩm (GBP)
    - CustomerID: mã khách hàng
    - Country: quốc gia khách hàng
* Kết quả:
  + Xử lý dữ liệu:
    - Làm sạch dữ liệu: loại bỏ giá trị thiếu (missing values), trùng lặp (duplicates), dữ liệu không hợp lệ (Quantity ≤ 0, UnitPrice ≤ 0).
    - Tạo thêm biến mới như: tổng doanh thu (TotalPrice = Quantity \* UnitPrice), RFM features (Recency, Frequency, Monetary).
  + Trực quan hóa dữ liệu:
    - Phân bố số lượng và doanh thu theo quốc gia, theo sản phẩm, theo khách hàng.
    - Biểu đồ xu hướng mua sắm theo thời gian (ngày/tuần/tháng).
    - Heatmap phân tích tương quan giữa các chỉ số RFM.
  + Mô hình hóa:
    - Hồi quy tuyến tính (Linear Regression): dự đoán tổng giá trị hóa đơn dựa trên các yếu tố Quantity, UnitPrice, Frequency…
    - Cây quyết định (Decision Tree Classifier): phân loại khách hàng thành nhóm có giá trị cao/thấp hoặc dự đoán khả năng mua hàng lại.
  + Đánh giá mô hình:
    - Hồi quy: MSE, R².
    - Phân loại: Classification Report, Confusion Matrix, ROC AUC.

##### 2.6.7.2 Load dataset



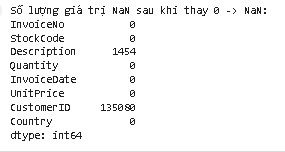
* Thống kê



**Nhận xét:**

* Quantity:
  + Có giá trị âm rất lớn (min = -80995), thể hiện có giao dịch trả hàng (refund) hoặc lỗi nhập liệu.
  + Trung bình ~9.55, trung vị = 3 ⇒ phần lớn đơn hàng có số lượng sản phẩm nhỏ.
  + Độ lệch chuẩn lớn (std = 218) ⇒ dữ liệu phân tán mạnh.
* UnitPrice:
  + Có giá trị âm (min = -11062.06), không hợp lý ⇒ cần làm sạch.
  + Trung bình ~4.61 GBP ⇒ đa phần sản phẩm giá rẻ.
  + Một số giá trị cực lớn (max = 38970) ⇒ khả năng có lỗi nhập hoặc sản phẩm cao cấp.
* InvoiceDate:
  + Khoảng thời gian dữ liệu: từ 01/12/2010 đến 09/12/2011 ⇒ dữ liệu trong 1 năm.
* CustomerID:
  + Có 406,829 giá trị hợp lệ / tổng 541,909 dòng ⇒ còn nhiều dòng bị thiếu ID khách hàng.

##### 2.6.7.3 Xử lý dữ liệu

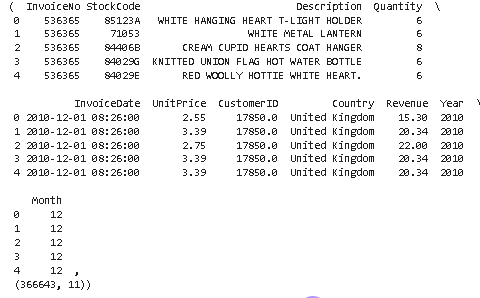


* Description: 1,454 giá trị thiếu.
* CustomerID: 135,080 giá trị thiếu ⇒ chiếm tỷ lệ lớn.
* Các cột khác: không còn NaN sau xử lý.

**Nhận xét:**

* Cần xử lý hoặc loại bỏ các dòng thiếu CustomerID nếu phân tích hành vi khách hàng.
* Dữ liệu mô tả sản phẩm có thể giữ lại nếu chỉ phân tích theo doanh thu.

**Drop missing CustomerID**

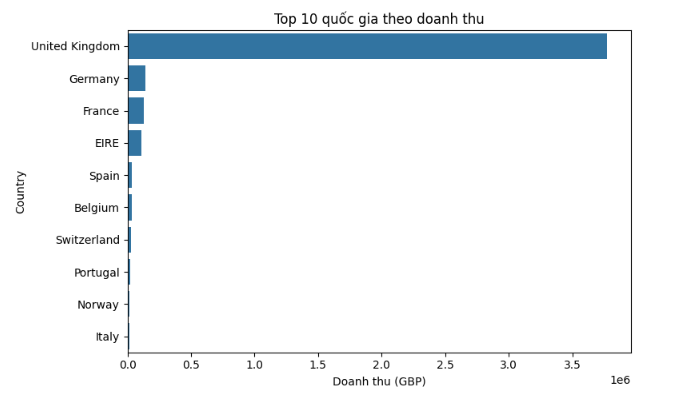
****

**Nhận xét:**

Các cột chính: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country, Revenue, Year, Month.

* Revenue = Quantity × UnitPrice (đúng theo logic doanh thu).
* Dữ liệu ví dụ đều ở tháng 12/2010, quốc gia "United Kingdom".
* Cấu trúc dữ liệu hợp lý, có thể dùng cho phân tích tiếp theo.

##### 2.6.7.4 Thống kê & trực quan hóa – quốc gia & khách hàng



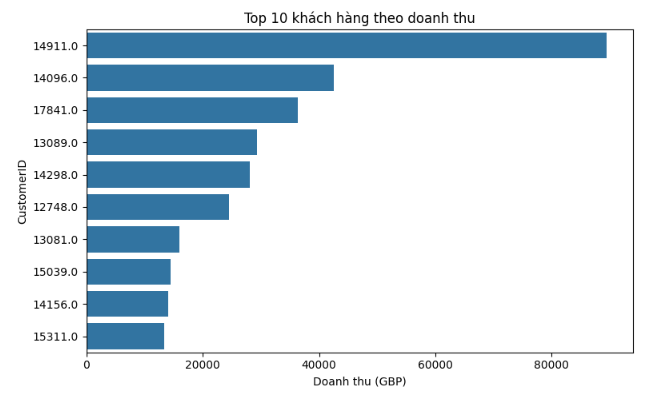
**Nhận xét:**

* United Kingdom chiếm áp đảo với doanh thu > 3.5 triệu GBP.
* Các quốc gia khác như Germany, France, EIRE, Spain... đóng góp rất nhỏ.
* Dữ liệu thiên lệch mạnh về thị trường Anh (UK là thị trường chính).

**Kết luận:**

* Khi trực quan hóa hoặc huấn luyện mô hình, nên xử lý mất cân bằng dữ liệu giữa UK và các quốc gia khác.

**Top khách hàng theo doanh thu**

****

**Nhận xét:**

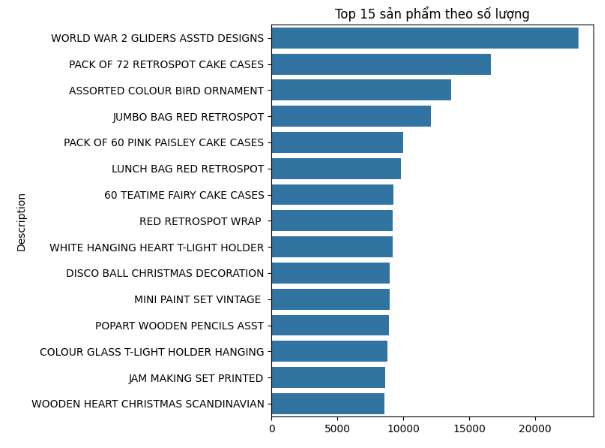
* Khách hàng có ID = 14911 có doanh thu cao nhất (~90,000 GBP).
* Các khách hàng còn lại dao động 15,000–60,000 GBP.
* Chênh lệch lớn ⇒ có khả năng khách hàng này là đại lý hoặc mua sỉ.

**Kết luận:**

* Có thể xác định nhóm khách hàng VIP để chăm sóc riêng.
* Phù hợp làm bước đầu cho phân tích RFM (Recency, Frequency, Monetary).

##### 2.6.7.5 Thống kê & trực quan hóa – quốc gia & khách hàng

**Top sản phẩm theo số lượng**

****

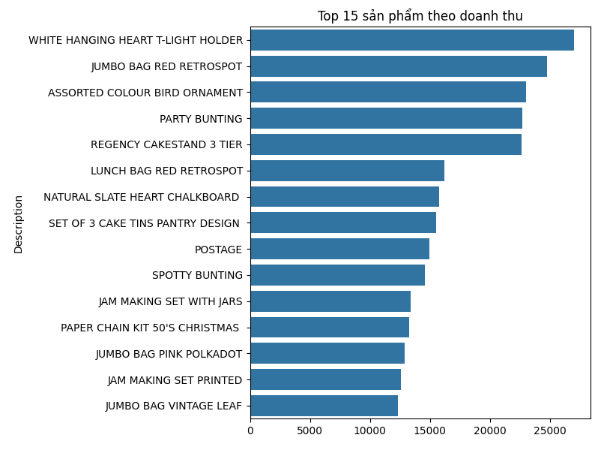
**Nhận xét:**

* Sản phẩm "WORLD WAR 2 GLIDERS ASSTD DESIGNS" bán được số lượng nhiều nhất, vượt trội so với các sản phẩm khác.
* Nhiều sản phẩm thuộc nhóm đồ trang trí, phụ kiện, túi quà, vật dụng nhỏ (ví dụ: “PACK OF 72 RETROSPOT CAKE CASES”, “ASSORTED COLOUR BIRD ORNAMENT”, “JUMBO BAG RED RETROSPOT”).
* Các mặt hàng này thường có giá trị nhỏ nhưng tiêu thụ mạnh.
* Phân phối sản phẩm thể hiện rõ xu hướng tiêu dùng các sản phẩm giá thấp, trang trí, quà tặng nhỏ.

Kết luận:

* Doanh nghiệp nên tập trung duy trì nguồn cung cho nhóm sản phẩm nhỏ, rẻ, và phổ biến vì có tần suất mua cao.
* Có thể nhóm chúng vào phân khúc "fast-selling low-value items".

**Top sản phẩm theo doanh thu**

****

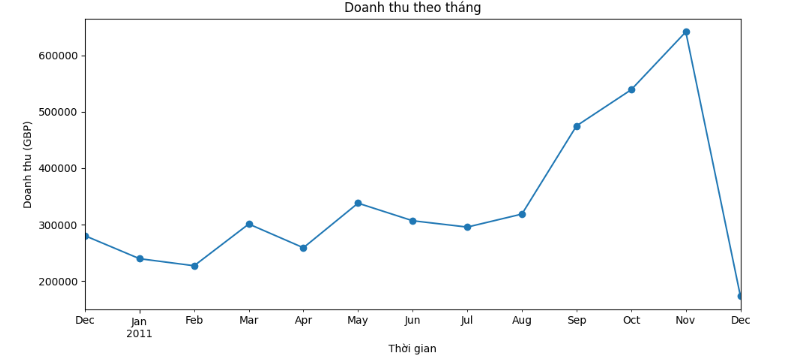
**Nhận xét:**

* Sản phẩm có doanh thu cao nhất là "WHITE HANGING HEART T-LIGHT HOLDER", kế đến là “JUMBO BAG RED RETROSPOT” và “ASSORTED COLOUR BIRD ORNAMENT”.
* Một số sản phẩm trùng với biểu đồ số lượng (ví dụ “JUMBO BAG RED RETROSPOT”) ⇒ cho thấy sản phẩm vừa bán nhiều vừa có giá trị cao.
* Các sản phẩm như “PARTY BUNTING” và “REGENCY CAKESTAND 3 TIER” có doanh thu cao dù không nằm trong top số lượng ⇒ có thể là sản phẩm giá trị đơn hàng cao.

Kết luận:

* Nhóm sản phẩm có doanh thu cao gồm cả hàng bán chạy và hàng giá cao.
* Doanh nghiệp nên chú trọng marketing, tồn kho và trưng bày cho nhóm sản phẩm này để tối đa hóa doanh thu.

**Doanh thu theo tháng**

****

**Nhận xét:**

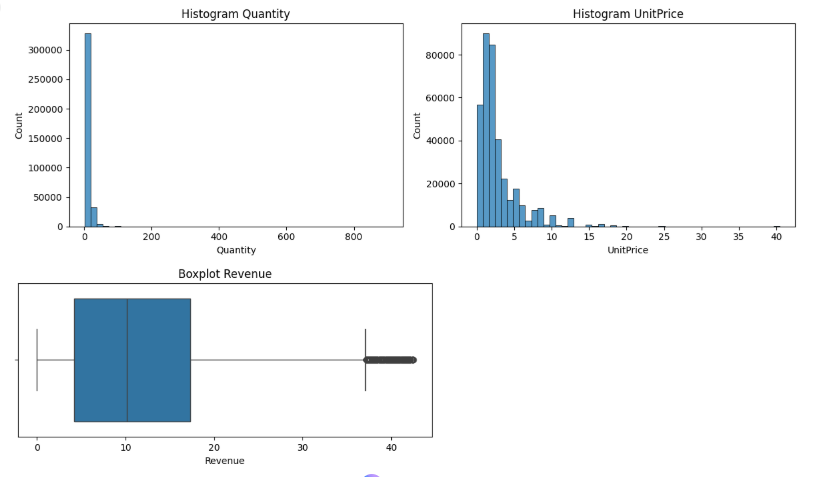
* Doanh thu dao động mạnh theo tháng, thấp nhất vào tháng 2 (~230,000 GBP) và tăng dần về cuối năm.
* Tăng mạnh từ tháng 9 đến tháng 11, đạt đỉnh vào tháng 11 (~650,000 GBP).
* Sau tháng 11, doanh thu giảm mạnh ở tháng 12 — có thể do dữ liệu tháng 12 chưa đầy đủ hoặc sau mùa lễ Giáng Sinh.

**Kết luận:**

* Có tính mùa vụ rõ ràng, doanh thu tăng vào quý 4 (thời điểm lễ hội, Giáng Sinh).
* Nên tập trung quảng bá, khuyến mãi, nhập hàng lớn từ tháng 9–11.

##### 2.6.7.6 Phân bố & kiểm tra outliers

* Histogram Quantity & UnitPrice
* Boxplot Revenue



*Biểu đồ Histogram và Boxplot Revenue*

**Nhận xét:** Biểu đồ Histogram

* **Quantity (số lượng sản phẩm)**:
  + Phân phối lệch phải mạnh (skewed right), tập trung chủ yếu ở giá trị nhỏ (1–20 sản phẩm mỗi hóa đơn).
  + Có một số giá trị ngoại lai (outliers) với số lượng cực lớn ⇒ có thể là đơn sỉ hoặc lỗi dữ liệu.
* **UnitPrice (đơn giá sản phẩm)**:
  + Cũng lệch phải mạnh, hầu hết sản phẩm có giá < 5 GBP.
  + Một số sản phẩm giá rất cao (>40 GBP) là ngoại lệ, có thể là hàng đặc biệt hoặc nhập lỗi.

**Kết luận:**

* Cần **xử lý ngoại lệ (outliers)** trong cả Quantity và UnitPrice trước khi xây dựng mô hình.
* Dữ liệu chủ yếu gồm các mặt hàng giá thấp, phù hợp thị trường tiêu dùng phổ thông.

**Nhận xét:** Biểu đồ Boxplot Revenue

* Biểu đồ cho thấy phân phối của biến **Revenue (doanh thu từng dòng giao dịch)** có **phân tán rộng và không đối xứng**.
* Phần lớn các giá trị Revenue nằm trong khoảng **0 – 20 GBP**, trong khi vẫn tồn tại một số **giá trị ngoại lai (outliers)** kéo dài đến hơn **40 GBP**.
* Trung vị (median) nằm gần 10 GBP ⇒ đa số giao dịch có doanh thu nhỏ.
* Hộp (box) khá rộng ⇒ mức độ biến thiên (dispersion) cao, thể hiện sự **đa dạng về quy mô đơn hàng**.
* Sự hiện diện của nhiều điểm ngoại lai bên phải cho thấy có những **giao dịch có doanh thu cao bất thường**, có thể là:
  + Đơn hàng sỉ (bán với số lượng lớn)
  + Hoặc lỗi nhập liệu (Quantity hoặc UnitPrice quá cao)

**Kết luận:**

* Dữ liệu Revenue **lệch phải rõ rệt (right-skewed)**.
* Cần xem xét xử lý ngoại lệ (outliers) trước khi thực hiện các phân tích thống kê hoặc mô hình dự báo.
* Có thể sử dụng log transformation (np.log1p(Revenue)) để làm phân phối cân bằng hơn khi xây dựng mô hình hồi quy hoặc clustering.

##### 2.6.7.7 Kết luận

* UK là thị trường chủ đạo
* Returns chiếm phần không nhỏ → cải thiện logistics
* Doanh thu tập trung vào một số khách hàng lớn
* Mùa cao điểm cuối năm → cần dự báo & kho hàng
* Mở rộng: phân tích RFM, dự báo doanh thu

## **PHẦN III. PHÂN TÍCH ĐƠN BIẾN VÀ HAI BIẾN**

### ***3.1. Phân tích đơn biến (Univariate Analysis) là gì ? Nó khác gì với phân tích hai biến (bivariate analysis) trong khám phá dữ liệu?***

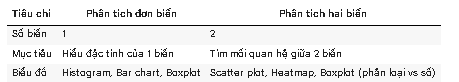
Phân tích đơn biến là việc phân tích và mô tả một biến duy nhất trong tập dữ liệu. Mục tiêu: hiểu phân phối (distribution), xu hướng trung tâm (central tendency) và mức độ phân tán (dispersion) của biến đó.  
→ Ví dụ:

* Với biến “tuổi” → ta xem độ tuổi trung bình, phân bố (nhiều người ở độ tuổi nào).
* Với biến “giới tính” → ta xem tỷ lệ nam/nữ.

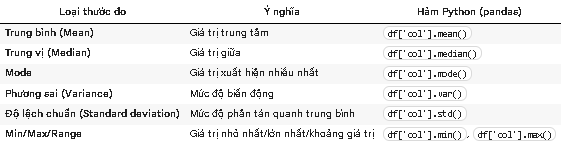
→Các biểu đồ thường dùng:

* Histogram (biểu đồ tần suất)
* Boxplot (hộp số)
* Bar chart (cho biến phân loại)

**Khác biệt:**



### ***3.2. Các thước đo thống kê trong phân tích đơn biến***



### ***3.3. Cách xác định mối quan hệ giữa hai biến***

***Tương quan (Correlation)***

→ Đo mức độ và chiều hướng của mối quan hệ tuyến tính giữa hai biến số.

* Giá trị trong khoảng [-1, 1]
* Gần 1: tương quan dương mạnh
* Gần -1: tương quan âm mạnh
* Gần 0: ít hoặc không tương quan

***Hàm***

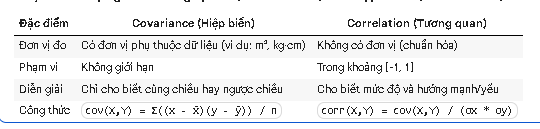
df[['height', 'weight']].corr()

***Nhân quả (Causation)***

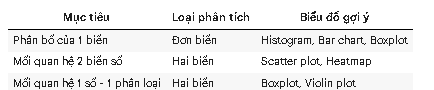
→ cho biết biến này gây ra biến kia.

Khó xác định bằng thống kê đơn thuần, cần thiết kế thí nghiệm hoặc mô hình hóa sâu hơn (ví dụ: hồi quy, kiểm định giả thuyết).

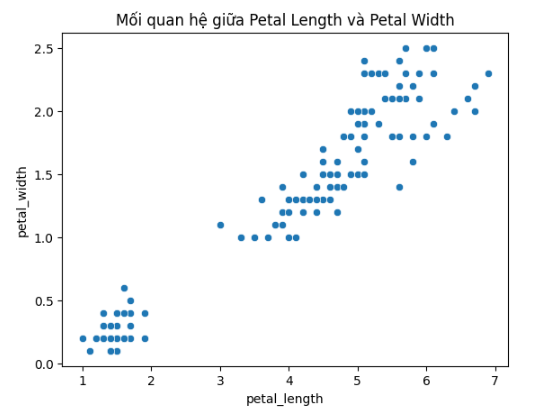
### ***3.4 Sự khác biệt giữa Tương quan (Correlation) và Hiệp biến (Covariance)***

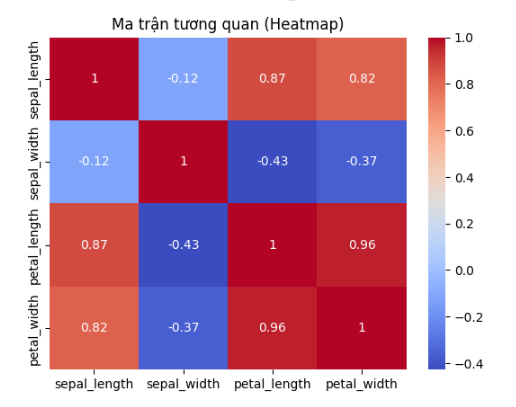


### ***3.5. Biểu đồ trực quan hóa khi nào dùng***



### ***3.6. Scatter plot & Heatmap***

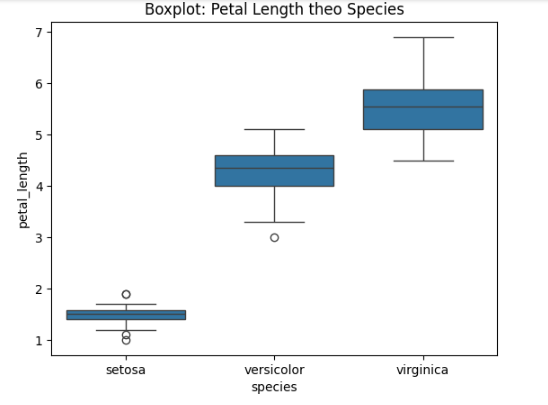




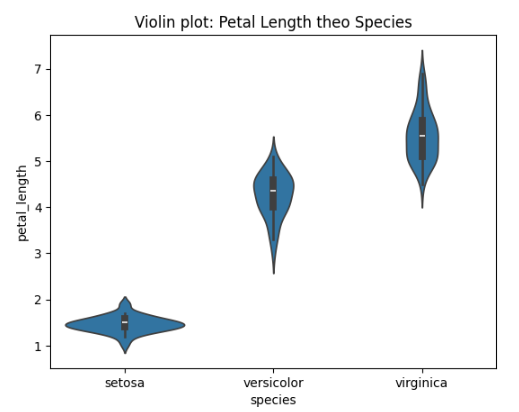
### ***3.7. Trực quan hóa mối quan hệ giữa biến số và biến phân loại***

Ví dụ: mối quan hệ giữa species (loài) và petal\_length trong bộ dữ liệu Iris.

**Boxplot**

****

**Violin plot**

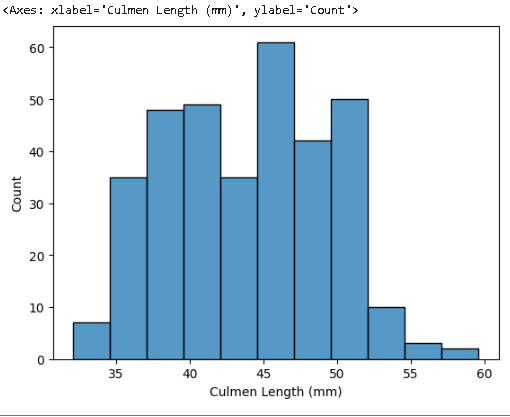
****

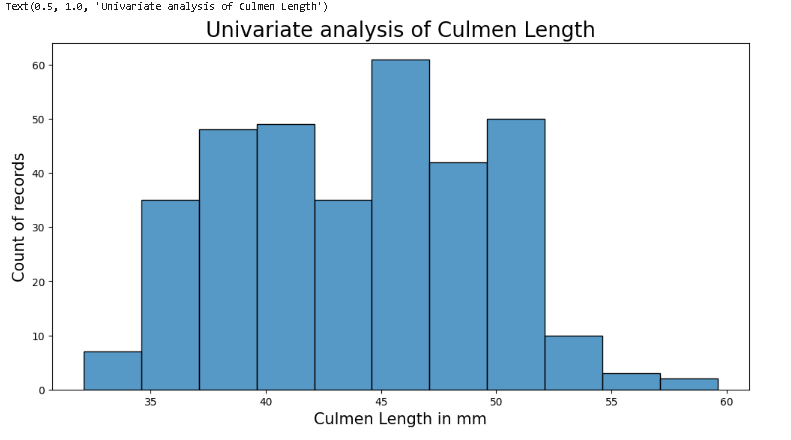
### ***3.8. Phân tích dữ liệu***

#### 3.8.1. Phân tích dữ liệu chim cánh cụt

##### 3.8.1.1. Phân tích đơn biến bằng Histogram

**case 1: basic**

****

**case 2: advanced**

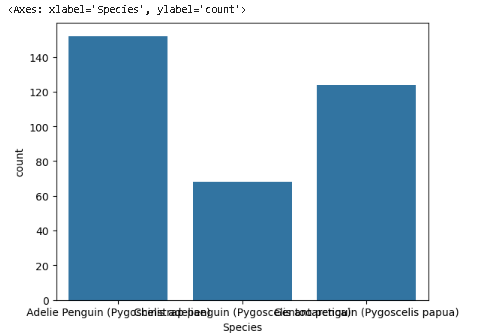
**Nhận Xét**

Biểu đồ Histogram cho thấy phân bố độ dài mỏ (Culmen Length) của các cá thể chim tập trung chủ yếu trong khoảng 35–50 mm. Cụ thể:

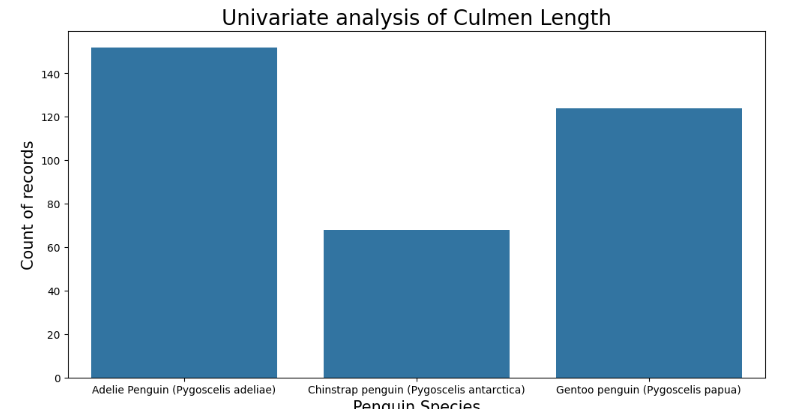
* Phần lớn các giá trị tập trung quanh trung bình khoảng 43–47 mm, thể hiện phân bố gần chuẩn (normal-like).
* Có một số ít giá trị ngoại biên (outliers) ở hai phía — đặc biệt là ở nhỏ hơn 35 mm và lớn hơn 55 mm, cho thấy chỉ có rất ít cá thể có mỏ ngắn hoặc dài bất thường.
* Phân bố này hơi lệch phải nhẹ, vì có một vài giá trị cao (55–60 mm) kéo dài phần đuôi bên phải.
* Nhìn chung, đa số cá thể chim có chiều dài mỏ ở mức trung bình, điều này cho thấy đặc điểm hình thái này khá ổn định trong quần thể.

##### 3.8.1.2. Phân tích đơn biến bằng bar chart

**case 1: basic**



**case 2: advanced**

****

**Nhận xét:**

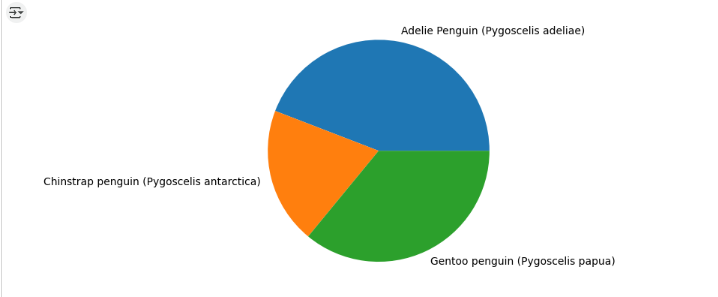
Loài Adelie chiếm ưu thế về số lượng mẫu, tiếp theo là Gentoo, còn Chinstrap ít nhất — cho thấy tập dữ liệu cần chú ý cân bằng khi dùng cho các phân tích sau. Biểu đồ cột thể hiện số lượng mẫu thu thập của ba loài chim cánh cụt trong tập dữ liệu.

Cụ thể:

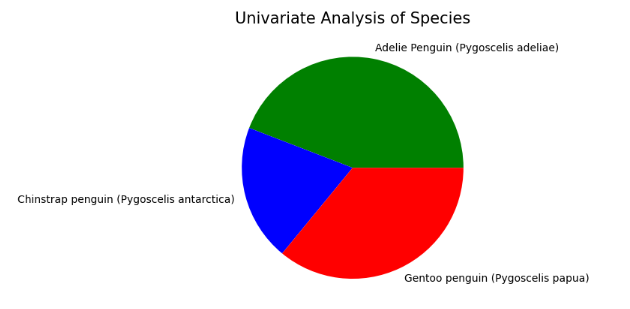
* Adelie Penguin (Pygoscelis adeliae) chiếm số lượng lớn nhất, với khoảng 150 cá thể, cho thấy đây là nhóm được thu thập dữ liệu nhiều nhất.
* Gentoo Penguin (Pygoscelis papua) đứng thứ hai với khoảng 120 cá thể, cũng chiếm tỷ lệ khá lớn trong quần thể.
* Chinstrap Penguin (Pygoscelis antarctica) có số lượng ít nhất, chỉ khoảng 70 cá thể, cho thấy đây là loài ít được ghi nhận hoặc phân bố hạn chế trong khu vực nghiên cứu.
* Tỷ lệ chênh lệch giữa các loài gợi ý rằng dữ liệu có thể mất cân bằng (class imbalance), điều này cần được lưu ý nếu dùng để huấn luyện mô hình phân loại.

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

**case 1: basic**



**case 2: advanced**

****

**Nhận xét :**

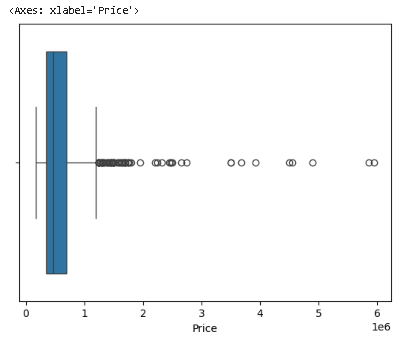
Biểu đồ tròn cho thấy tỷ lệ phân bố giữa ba loài chim cánh cụt trong tập dữ liệu không đồng đều:

* Adelie Penguin (Pygoscelis adeliae) chiếm tỷ lệ cao nhất, thể hiện rằng đây là loài phổ biến nhất hoặc được thu thập dữ liệu nhiều nhất.
* Gentoo Penguin (Pygoscelis papua) chiếm tỷ lệ trung bình, cho thấy sự hiện diện tương đối lớn nhưng không vượt trội.
* Chinstrap Penguin (Pygoscelis antarctica) chiếm tỷ lệ nhỏ nhất, thể hiện số lượng mẫu của loài này ít hơn đáng kể so với hai loài còn lại.
* Từ đó có thể thấy dữ liệu bị mất cân bằng (class imbalance), cần lưu ý nếu sử dụng để huấn luyện mô hình học máy nhằm tránh sai lệch dự đoán về loài phổ biến hơn.

#### 3.8.2. Phân tích dữ liệu đơn biến trên dữ liệu giá nhà

##### 3.8.2.1. Phân tích đơn biến dựa vào boxplot

**Case 1**

****

**Case 2**

****

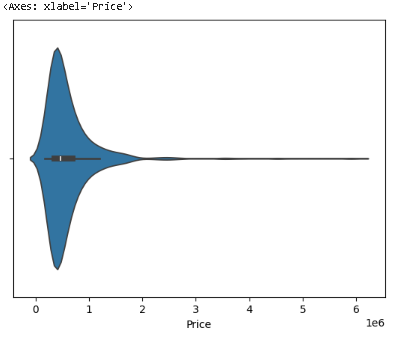
**Nhận xét :**

* Biểu đồ hộp cho thấy phần lớn giá nhà tập trung ở mức thấp hơn 1.000.000 triệu (tức khoảng dưới 1 tỷ đồng nếu tính theo đơn vị triệu). Tuy nhiên, có nhiều điểm ngoại lai (outliers) nằm rải rác ở các mức giá cao, thậm chí vượt quá 6.000.000 triệu.
* Điều này cho thấy phân bố giá nhà bị lệch phải (right-skewed) — phần lớn các căn nhà có giá vừa phải, trong khi một số ít nhà có giá trị rất cao kéo trung bình chung lên.

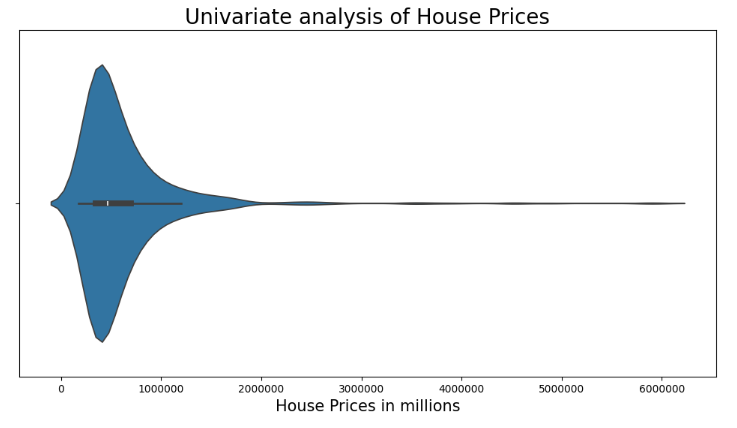
=> Cần cân nhắc loại bỏ hoặc xử lý ngoại lai trước khi thực hiện các mô hình dự đoán để tránh ảnh hưởng đến kết quả huấn luyện.

##### 3.8.2.2. Phân tích dữ liệu đơn biến dựa vào violin plot

**case 1: basic**

****

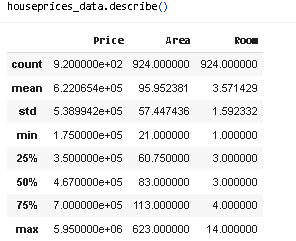
**case 2: advanced**

****

**Nhận xét :**

* Biểu đồ violin cho thấy phần lớn giá nhà tập trung ở mức thấp, chủ yếu nằm trong khoảng dưới 1.000.000 triệu.
* Phân phối dữ liệu lệch phải mạnh (right-skewed), thể hiện ở phần “đuôi” kéo dài về phía các giá trị cao hơn — điều này cho thấy tồn tại nhiều căn nhà có giá rất cao (outliers) so với mặt bằng chung.
* Phần thân đồ thị dày ở phía đầu cho thấy đa số căn nhà có giá ở mức trung bình thấp, chỉ một số ít thuộc nhóm cao cấp.

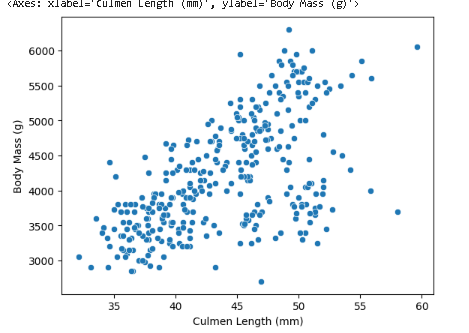
##### 3.8.2.3. Phân tích dữ liệu đơn biến dựa vào bản tóm tắt dữ liệu



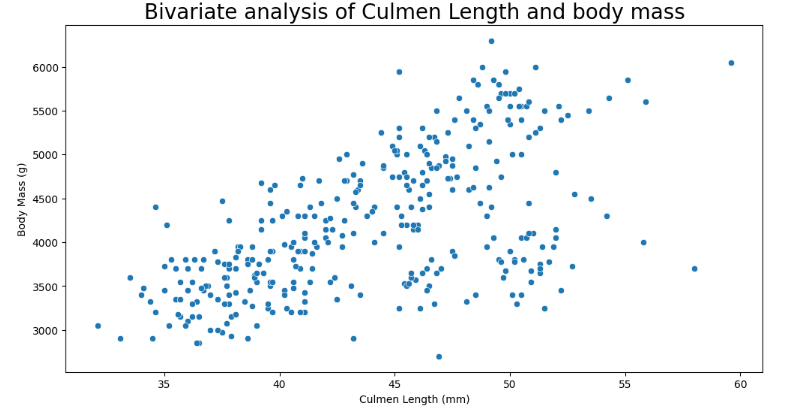
#### 3.8.3. Phân tích dữ liệu hai biến trên dữ liệu về chim cánh cụt

##### 3.8.3.1. Phân tích dữ liệu 2 biến dựa vào phương pháp scatterplot

**case 1: basic**



**case 2: advanced**

****

Nhận xét:

1. Xu hướng tổng thể (Overall trend):  
    Có mối tương quan thuận giữa độ dài mỏ và khối lượng cơ thể.  
    → Nghĩa là chim có mỏ dài hơn thì thường nặng hơn.
2. Độ chặt của mối quan hệ (Strength of correlation):  
   * Mặc dù có phân tán, nhưng xu hướng tăng khá rõ ràng.
   * Cho thấy mối liên hệ tuyến tính vừa đến mạnh (positive moderate-to-strong linear correlation).
3. Phân tán dữ liệu (Data spread):  
   * Có một vài điểm ngoại lệ (outliers) — ví dụ một số cá thể có mỏ dài nhưng khối lượng không cao, hoặc ngược lại.
   * Điều này có thể do khác biệt giữa giới tính, loài, hoặc độ tuổi.
4. Kết luận sơ bộ:  
    → Độ dài mỏ có thể là biến dự đoán hợp lý cho khối lượng cơ thể.  
    → Tuy nhiên, để khẳng định, nên tính hệ số tương quan Pearson và thử mô hình hồi quy tuyến tính để đánh giá chính xác hơn.

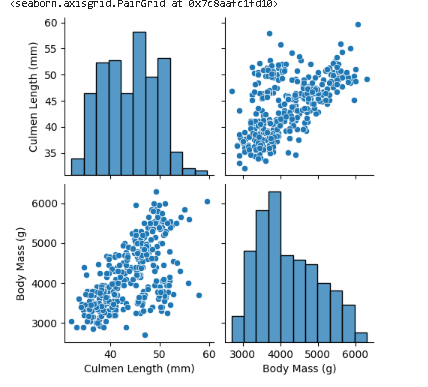
##### 3.8.3.2. Phân tích 2 biến dựa vào bảng crosstab/two-way



##### 3.8.3.3. Phân tích 2 biến sử dụng pivot\_table

****

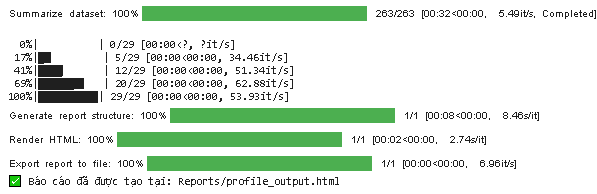
##### 3.8.3.4. Phân tích 2 biến sử dụng pairplot



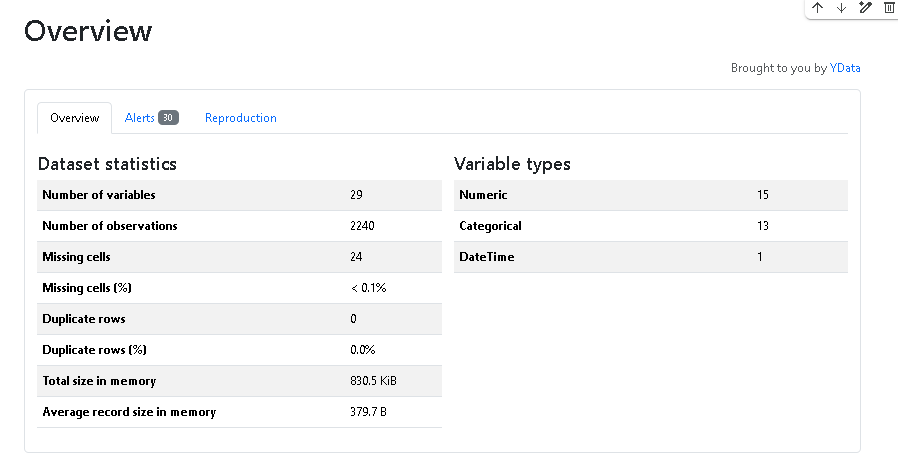
*Biểu đồ Pairplot: Phân tích hai biến của Chiều dài Culmen và khối lượng cơ thể.*

#### 3.8.4. Các công cụ hỗ trợ EDA tự động

##### 3.8.4.1. Tạo báo cáo EDA tự động

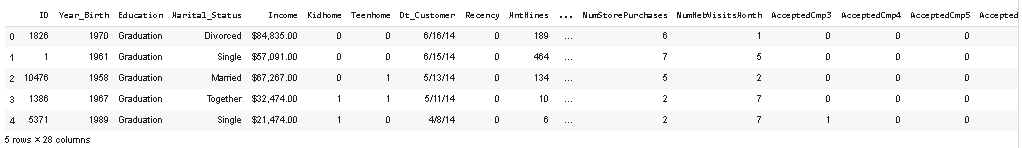


* Sử dụng lệnh profile.to\_notebook\_iframe() để mở báo cáo ngay trong collab



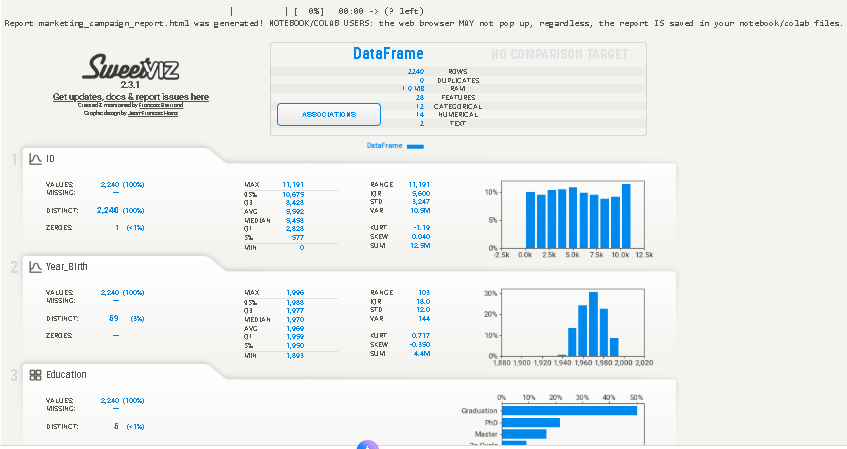
### **Bài tập thực hành 1:** Tìm hiểu các tính năng và cách sử dụng sản phẩm SweetViz

***1. Đọc tập dữ liệu Marketing Campaign***



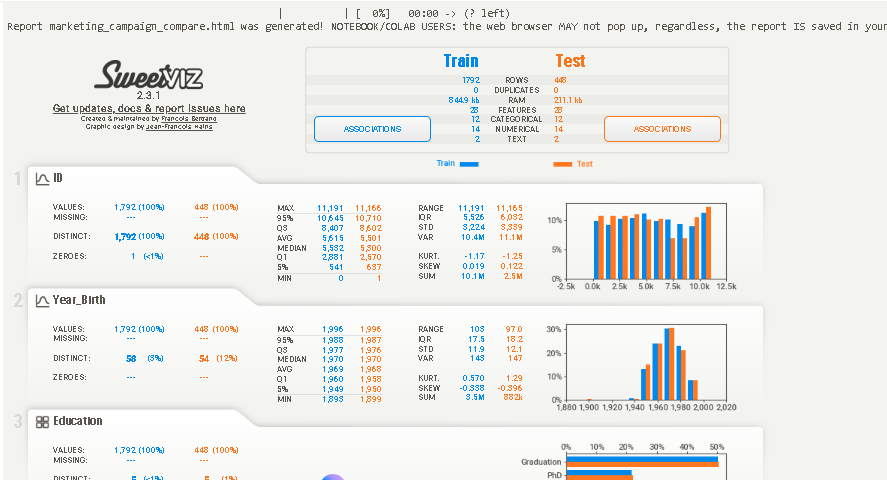
***2. Phân tích dữ liệu tự động với SweetViz***

* Tạo báo cáo tự động với analyze()

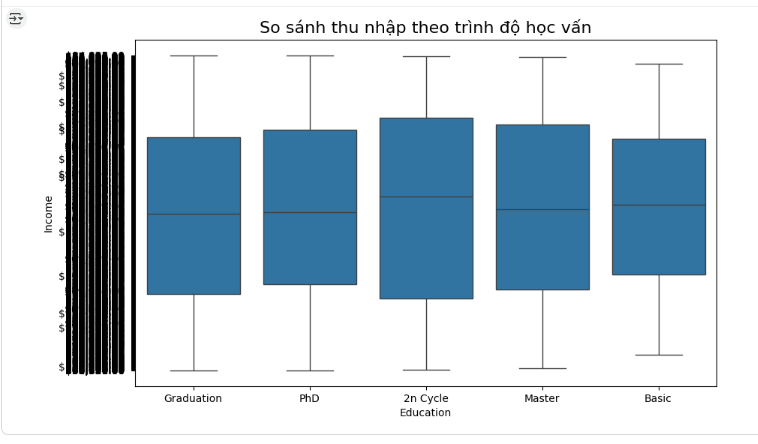


***3. So sánh hai tập dữ liệu***

* Sử dụng hàm compare() để so sánh 2 tập tin



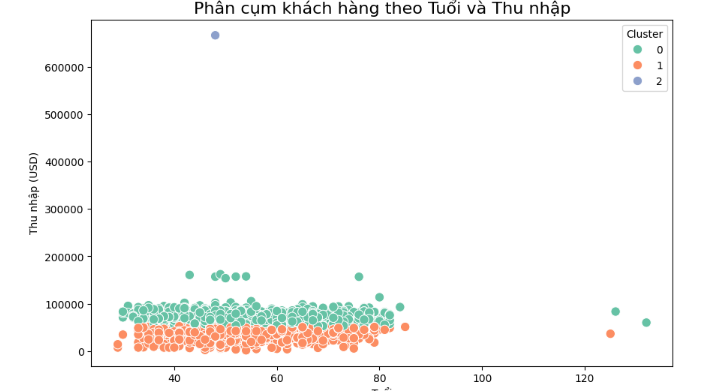
***4. Kết hợp SweetViz + Seaborn để phân tích sâu hơn***

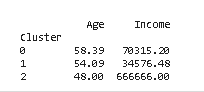
****

**Nhận xét chi tiết:**

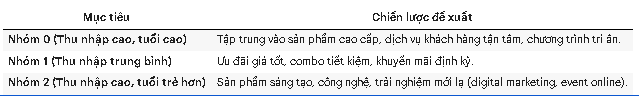
1. Tổng quan:  
   * Mỗi hộp thể hiện phân bố thu nhập của một nhóm trình độ học vấn.
   * Các đường giữa hộp là median (trung vị), còn hộp thể hiện khoảng tứ phân vị (IQR – Interquartile Range).
   * Đường râu (whiskers) thể hiện phạm vi dữ liệu không ngoại lệ.
2. So sánh giữa các nhóm:  
   * 2nd Cycle Education và Master có trung vị thu nhập cao hơn so với các nhóm khác.
   * Basic có trung vị thấp nhất, phản ánh rằng người học thấp thường có thu nhập thấp hơn.
   * PhD và Graduation nằm ở mức trung bình khá, nhưng không vượt trội hơn rõ ràng.
3. Phân bố dữ liệu:  
   * Mức độ phân tán thu nhập khá lớn ở tất cả các nhóm, cho thấy sự khác biệt thu nhập trong cùng một nhóm học vấn vẫn đáng kể.
   * Một vài nhóm (như *2nd Cycle Education*) có râu dài hơn, nghĩa là có nhiều cá nhân thu nhập rất cao hoặc rất thấp.
4. Kết luận sơ bộ:  
   * Có xu hướng: trình độ học vấn cao hơn → thu nhập cao hơn, nhưng không tuyệt đối.
   * Các yếu tố khác (như kinh nghiệm, lĩnh vực làm việc, vị trí địa lý...) cũng ảnh hưởng mạnh đến thu nhập.

***5. Phân cụm khách hàng theo độ tuổi và thu nhập (K-Means Clustering)***

****

****

* **Nhận xét:**

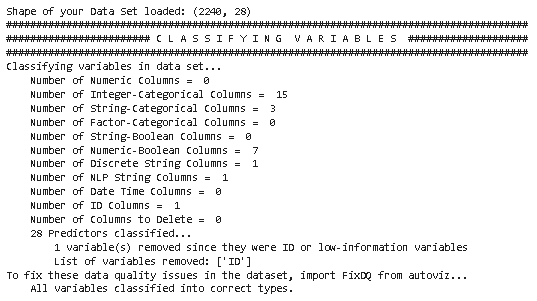
****

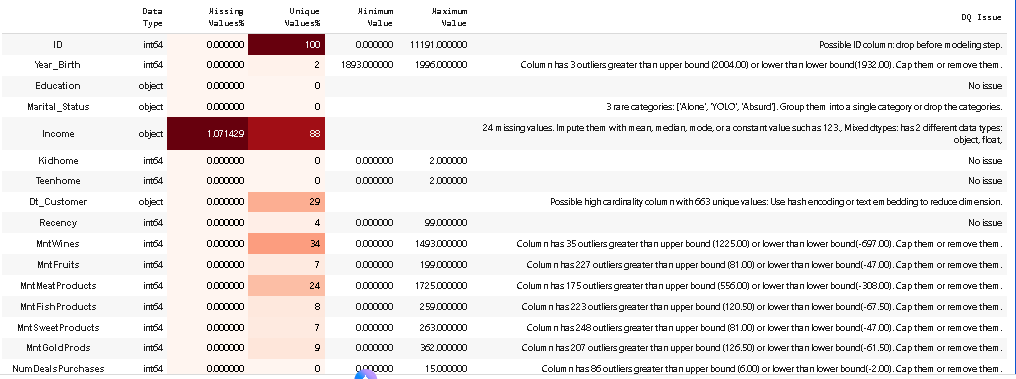
### **Bài tập thực hành 2:** Tìm hiểu các tính năng và cách sử dụng sản phẩm AutoViz

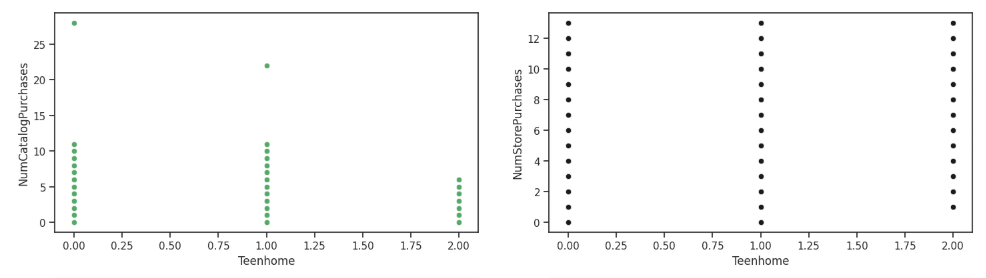
***1. Đọc tập dữ liệu***

****

***2. Khởi tạo AutoViz và chạy trên file CSV***

****

****

****