**UỶ BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**
**TRƯỜNG ĐẠI HỌC SÀI GÒN**
**KHOA TOÁN - ỨNG DỤNG**

# BÁO CÁO:
# PHÂN TÍCH KHÁM PHÁ DỮ LIỆU
# HỌC PHẦN: KHAI PHÁ DỮ LIỆU (858016)

Giảng viên hướng dẫn: Đỗ Như Tài

Sinh viên thực hiện: Trần Quốc Hoàng (3123580015)

Trần Chí Vỹ (3123580065)

Phan Trần Hữu Tấn (3123580042)

Lớp: DDU1231

*Thành phố Hồ Chí Minh, ngày 12 tháng 10 năm 2025*

# BẢNG PHÂN CÔNG

| Tên - MSSV | Phân công | Tham gia (%) |
|---|---|---|
| Trần Quốc Hoàng - 3123580015 (Nhóm trường) | 1.3.1, 1.1.4, 1.2.3, làm tiểu luận | 100% |
| Trần Chí Vỹ - 3123580065 | 1.1.1, 1.3.3, 1.3.4 | 100% |
| Phan Trần Hữu Tấn - 3123580042 | 1.2.1, 1.1.3, 1.2.2 | 100% |

# LỜI CẢM ƠN

Lời nói đầu tiên cho chúng em xin trân trọng gửi lời cảm ơn chân thành và sự kính trọng tới Giảng viên Đỗ Như Tài đã tận tình chỉ bảo chúng em trong suốt quá trình học tập trong lớp và hướng dẫn bọn em cách thực hiện bài tiểu luận này.

Dù đã rất cố gắng trong quá trình nghiên cứu và thực hành, nhưng do kiến thức và kinh nghiệm còn hạn chế, chắc chắn bài làm vẫn không tránh khỏi những sai sót. Em mong Thầy có thể thông cảm và xem xét bỏ qua cho chúng em.

# MỤC LỤC

# A. MỤC TIÊU

Bài thực hành này nhằm giúp người học nắm vững các kỹ thuật cơ bản trong khám phá dữ liệu để hiểu rõ đặc điểm và cấu trúc của tập dữ liệu. Cụ thể, sinh viên sẽ thực hiện các bước phân tích thống kê mô tả để xác định các đặc trưng chính như giá trị trung bình, trung vị, độ lệch chuẩn và phân bố của dữ liệu.

Đồng thời, bài thực hành hướng dẫn sử dụng các công cụ trực quan hóa như biểu đồ histogram, boxplot, và scatter plot để phát hiện các mẫu, xu hướng, hoặc bất thường trong dữ liệu. Sinh viên sẽ được làm quen với các thư viện Python như Pandas, Matplotlib, và Seaborn để xử lý và trực quan hóa dữ liệu hiệu quả.

Ngoài ra, bài thực hành giúp nhận diện các vấn đề như giá trị thiếu, giá trị ngoại lai, hoặc sự không nhất quán trong dữ liệu, từ đó đề xuất các phương pháp tiền xử lý phù hợp. Kết quả cuối cùng là sinh viên có thể đưa ra các nhận định ban đầu về dữ liệu, đặt nền tảng cho các bước phân tích sâu hơn hoặc xây dựng mô hình khai thác dữ liệu trong các ứng dụng thực tiễn như phân tích khách hàng hoặc dự đoán xu hướng.

# B. KẾT CẤU THỰC HÀNH

Thực hành bao gồm 3 phần:
- Thống kê mô tả.
- Xử lý và trực quan hóa dữ liệu.
- Phân tích đơn biến và hai biến.

# C. NỘI DUNG THỰC HÀNH

## 1.1. THỐNG KÊ MÔ TẢ

### 1.1.1. Ôn tập lý thuyết

**+ Thống kê mô tả là gì? Nó khác gì với thống kê suy luận (inferential statistics)?**

- Thống kê mô tả (Descriptive statistics): Tóm tắt, tổ chức và trình bày dữ liệu để mô tả đặc trưng cơ bản của tập dữ liệu, không suy diễn ra ngoài phạm vi dữ liệu.

+ Ví dụ: Tính trung bình chiều cao của học sinh trong lớp, vẽ biểu đồ phân bố để xem xu hướng.

- Thống kê suy luận (Inferential statistics): Dùng dữ liệu mẫu để suy ra đặc điểm của tổng thể, dựa trên xác suất, kiểm định giả thuyết và ước lượng tham số.

+ Ví dụ: Dùng dữ liệu 100 học sinh để ước lượng chiều cao trung bình của toàn trường.

- Khác biệt chính:

- Thống kê mô tả: Chỉ mô tả dữ liệu hiện có, không suy luận.
- Thống kê suy luận: Suy ra đặc tính tổng thể từ mẫu, có yếu tố xác suất và không chắc chắn.

**+ Các thước đo thống kê mô tả chính (ví dụ: trung bình, trung vị, phương sai, độ lệch chuẩn) được sử dụng để làm gì? Trong trường hợp nào thì nên dùng trung vị thay vì trung bình?**

- Các thước đo thống kê mô tả chính:

- Trung bình (Mean): Giá trị trung bình cộng, đại diện cho xu hướng trung tâm của dữ liệu. Ví dụ: Trung bình điểm số phản ánh hiệu suất chung của lớp.
- Trung vị (Median): Giá trị ở giữa khi sắp xếp dữ liệu; ít bị ảnh hưởng bởi giá trị ngoại lai (outlier). Ví dụ: Trung vị thu nhập phản ánh mức điển hình của đa số người dân.

- ➢ Phương sai (Variance): Trung bình bình phương độ lệch giữa các giá trị và trung bình; đo mức độ phân tán của dữ liệu. Ví dụ: Phương sai cao cho thấy dữ liệu biến động lớn.
- ➢ Độ lệch chuẩn (Standard deviation): Căn bậc hai của phương sai, có cùng đơn vị với dữ liệu, giúp dễ diễn giải hơn. Ví dụ: Độ lệch chuẩn chiều cao cho biết đa số nằm quanh giá trị trung bình.

- Khi nào dùng trung vị thay trung bình:

→ Khi dữ liệu có outliers hoặc phân bố lệch (skewed), vì trung vị ổn định hơn và phản ánh trung tâm thực tế tốt hơn. Ví dụ: Trung vị thu nhập hợp lý hơn trung bình nếu có vài người thu nhập quá cao.

**+ Làm thế nào để xác định phân bố của một tập dữ liệu? Các loại phân bố phổ biến là gì (ví dụ: phân bố chuẩn, lệch trái, lệch phải)?**

- Cách xác định phân bố dữ liệu:
  - ➢ Vẽ histogram hoặc density plot để quan sát hình dạng phân bố.
  - ➢ Tính trung bình, trung vị, mode, skewness (độ lệch) và kurtosis (độ nhọn).
    - **skewness = 0**: đối xứng.
    - **skewness > 0**: lệch phải.
    - **skewness < 0**: lệch trái.
  - ➢ Dùng Python (pandas, seaborn) hoặc R để vẽ và tính tự động.
- Các loại phân bố phổ biến:
  - ➢ Phân bố chuẩn (Normal): Hình chuông đối xứng, trung bình = trung vị = mode. Ví dụ: IQ, chiều cao.
  - ➢ Lệch trái (Negative skew): Đuôi dài bên trái, trung bình < trung vị. Ví dụ: điểm thi khi đề dễ, đa số điểm cao.
  - ➢ Lệch phải (Positive skew): Đuôi dài bên phải, trung bình > trung vị. Ví dụ: thu nhập, giá nhà.
- Khác:
  - ➢ Phân bố đều (Uniform): mọi giá trị có xác suất bằng nhau.
  - ➢ Nhị thức (Binomial): mô tả số lần thành công trong n lần thử.

- Poisson: mô tả số lần xảy ra sự kiện hiếm trong khoảng thời gian.

- Ý nghĩa: Phân bố giúp hiểu đặc điểm dữ liệu và chọn phương pháp phân tích hoặc mô hình thống kê phù hợp.

**+ Độ lệch chuẩn và phạm vi (range) có ý nghĩa gì trong việc đánh giá sự phân tán của dữ liệu?**

- Độ lệch chuẩn (Standard Deviation): Đo mức độ dữ liệu phân tán quanh giá trị trung bình.
  - Độ lệch chuẩn càng lớn → dữ liệu càng biến động mạnh.
  - Với phân bố chuẩn: khoảng 68% dữ liệu nằm trong ±1 độ lệch chuẩn, 95% trong ±2.
  - Ví dụ: Nếu điểm trung bình là 70 và độ lệch chuẩn là 10 → hầu hết điểm nằm trong khoảng 60–80.
- Phạm vi (Range): Là hiệu giữa giá trị lớn nhất và nhỏ nhất → cho biết độ rộng của dữ liệu.
  - Ví dụ: Chiều cao 150–190 cm → range = 40 cm.
  - Tuy nhiên, phạm vi bị ảnh hưởng mạnh bởi giá trị ngoại lai (outliers), nên không phản ánh đầy đủ mức phân tán chung.
- So sánh:

  + Độ lệch chuẩn phản ánh phân tán toàn bộ dữ liệu, còn range chỉ cho biết khoảng biến thiên cực đại. Vì vậy, độ lệch chuẩn được ưa dùng hơn trong phân tích thống kê nghiêm túc.

**+ Sự khác biệt giữa các thước đo như Q1, Q2, Q3 trong biểu đồ hộp (boxplot) là gì?**

| Thước đo | Phần trăm dữ liệu dưới nó | Vai trò trong boxplot |
|---|---|---|
| Q1 | 25% | Đáy hộp |
| Q2 | 50% | Đường giữa hộp (trung vị) |
| Q3 | 75% | Đỉnh hộp |

- Ý nghĩa trong boxplot:
  - Hộp (box) được tạo bởi Q1 và Q3 → chứa 50% dữ liệu trung tâm.
  - IQR (Interquartile Range) = Q3 − Q1 → đo độ phân tán của phần lớn dữ liệu.
  - Điểm ngoại lai (outlier) là giá trị nằm ngoài khoảng **[Q1 − 1.5 × IQR, Q3 + 1.5 × IQR]**.

**+ Làm thế nào để xử lý giá trị thiếu (missing values) trước khi tính toán các chỉ số thống kê mô tả?**

1) Xác định giá trị thiếu:

→ Dùng df.isnull().sum() trong pandas để đếm số lượng và tỉ lệ missing trên mỗi cột.

2) Phân tích nguyên nhân:
   - MCAR (Missing Completely at Random): Thiếu ngẫu nhiên → có thể xóa an toàn.
   - MAR (Missing at Random): Thiếu có điều kiện → nên thay thế bằng giá trị ước lượng.
   - MNAR (Missing Not at Random): Thiếu không ngẫu nhiên → cần xem xét kỹ, không nên xử lý cơ học.
3) Cách xử lý:
   - Xóa hàng/cột: Dùng df.dropna() nếu tỷ lệ thiếu nhỏ (< 5%) và dữ liệu đủ lớn.
   - Thay thế (imputation):
     - Dữ liệu số: Dùng mean hoặc median (nếu phân bố lệch).
     - Dữ liệu phân loại: Dùng mode.
     - Dữ liệu phức tạp: Dùng mô hình (KNNImputer, Regression).
   - Giữ nguyên & đánh dấu: Nếu missing có ý nghĩa (ví dụ: "chưa trả lời"), thêm cột chỉ báo như is_missing = df['age'].isnull().astype(int).

4) Kiểm tra lại:

→ Sau khi xử lý, tính lại trung bình, trung vị, độ lệch chuẩn để đảm bảo không làm méo phân bố dữ liệu.

**+ Bạn có thể giải thích cách đọc và diễn giải một biểu đồ histogram hoặc boxplot từ dữ liệu thực tế không?**

- Histogram:
  - Dùng cho dữ liệu liên tục, xem phân bố và xu hướng.
  - Cột cao: giá trị xuất hiện nhiều.
  - Đối xứng: phân bố chuẩn.
  - Lệch phải: nhiều giá trị nhỏ, ít giá trị lớn.
  - Lệch trái: nhiều giá trị lớn, ít giá trị nhỏ.

  + Ví dụ: Histogram điểm thi lệch phải → đa số điểm cao, ít điểm thấp.
- Boxplot:
  - Cho thấy trung vị (Q2), phân tán (IQR) và ngoại lệ (outlier).
  - Đường giữa hộp: trung vị.
  - Hộp dài: dữ liệu biến động lớn.
  - Chấm ngoài hộp: outlier.

  + Ví dụ: Boxplot lương lệch phải → có vài người lương rất cao.

**+ Khi gặp một tập dữ liệu có giá trị ngoại lai (outliers), bạn sẽ xử lý chúng như thế nào trước khi thực hiện thống kê mô tả?**

1) Xác định outlier:
   - Dùng boxplot → giá trị ngoài phạm vi **[Q1 - 1.5 × IQR, Q3 + 1.5 × IQR].**
   - Dùng z-score → $|z| > 3$ là nghi ngờ ngoại lai.

   + Ví dụ: Chiều cao 250 cm trong tập 160–180 cm là outlier rõ ràng.
2) Phân tích nguyên nhân:
   - Lỗi nhập dữ liệu? → nên loại bỏ.
   - Giá trị thật nhưng hiếm? → có thể giữ lại (ví dụ: vận động viên, doanh số dịp lễ).

3) Cách xử lý:

- Giữ nguyên: nếu có ý nghĩa thực tế.
- Xóa: nếu là lỗi hoặc ảnh hưởng mạnh mà không đại diện.
- Thay thế: giới hạn bằng biên trên/dưới (winsorizing).
- Biến đổi: log-transform để giảm ảnh hưởng.

4) Kiểm tra lại:

→ So sánh thống kê trước và sau xử lý để đảm bảo không làm méo phân bố.

→ Ghi chú rõ cách xử lý trong báo cáo để đảm bảo minh bạch.

- Nguyên tắc: Không xóa outlier chỉ vì "xấu mắt" — hãy hiểu vì sao nó xuất hiện trước khi quyết định.

### 1.1.2. Bài làm mẫu

**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** lấy tại https://ourworldindata.org/coronavirus

1. 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.

```
# Giả sử tải được file csv

import numpy as np
import pandas as pd
from scipy import stats
# Load the .csv into a dataframe using read_csv
covid_data = pd.read_csv("covid-data.csv")
covid_data = covid_data[['iso_code','continent',
'location','date','total_cases','new_cases']]
# Take a quick look at the data
covid_data.head(5)
covid_data.dtypes
covid_data.shape
```

| | iso_code | continent | location | date | total_cases | new_cases |
|---|---|---|---|---|---|---|
| 0 | IND | Europe | New York | 2020-01-01 | 182051 | 2702 |
| 1 | CHN | Asia | Delhi | 2020-01-02 | 97661 | 4006 |
| 2 | BRA | Europe | New York | 2020-01-03 | 924483 | 3868 |
| 3 | CHN | Europe | Moscow | 2020-01-04 | 943693 | 4200 |
| 4 | CHN | America | Beijing | 2020-01-05 | 697960 | 201 |

```
iso_code        object
continent       object
location        object
date            object
total_cases      int64
new_cases        int64
dtype: object
```

```
# Get the mean of the data
data_mean = np.mean(covid_data["new_cases"])
# Get the median of the data
data_median = np.median(covid_data["new_cases"])
# Get the mode of the data
data_mode = stats.mode(covid_data["new_cases"])
# Obtain the variance of the data
data_variance = np.var(covid_data["new_cases"])
# Obtain the standard deviation of the data
data_sd = np.std(covid_data["new_cases"])
# Compute the maximum and minimum values of the data
data_max = np.max(covid_data["new_cases"])
data_min = np.min(covid_data["new_cases"])
# Obtain the 60th percentile of the data
data_percentile = np.percentile(covid_data["new_cases"],60)
# Obtain the quartiles of the data
data_quartile = np.quantile(covid_data["new_cases"],0.75)
# Get the IQR of the data
data_IQR = stats.iqr(covid_data["new_cases"])
```

| | Statistic | Value |
|---|---|---|
| 0 | Mean | 2.589629e+03 |
| 1 | Median | 2.620000e+03 |
| 2 | Mode | 8.820000e+02 |
| 3 | Variance | 1.961764e+06 |
| 4 | Standard Deviation | 1.400630e+03 |
| 5 | Range | 4.991000e+03 |
| 6 | 60th Percentile | 3.167400e+03 |
| 7 | Q1 (25%) | 1.453250e+03 |
| 8 | Q2 (50%) | 2.620000e+03 |
| 9 | Q3 (75%) | 3.753000e+03 |
| 10 | IQR | 2.299750e+03 |

**Nhiệm vụ 2: Khám phá và xử lý dữ liệu Marketing Campaign** lấy tại https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis

1. Import thư viện và nạp dữ liệu vào notebook

```
import pandas as pd
# Đọc và sao chép dữ liệu
marketing_data = pd.read_csv("data/marketing_campaign.csv", sep ="\t")
df = marketing_data.copy()
df = df[['ID','Year_Birth', 'Education', 'Marital_Status','Income','Kidhome',
'Teenhome', 'Dt_Customer', 'Recency', 'NumStorePurchases',
'NumWebVisitsMonth']]
# Xem 5 dòng đầu
df.head()
```

| | ID | Year_Birth | Education | Marital_Status | Income | Kidhome | Teenhome | Dt_Customer | Recency | NumStorePurchases | NumWebVisitsMonth |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 5524 | 1957 | Graduation | Single | 58138.0 | 0 | 0 | 04-09-2012 | 58 | 4 | 7 |
| 1 | 2174 | 1954 | Graduation | Single | 46344.0 | 1 | 1 | 08-03-2014 | 38 | 2 | 5 |
| 2 | 4141 | 1965 | Graduation | Together | 71613.0 | 0 | 0 | 21-08-2013 | 26 | 10 | 4 |
| 3 | 6182 | 1984 | Graduation | Together | 26646.0 | 1 | 0 | 10-02-2014 | 26 | 4 | 6 |
| 4 | 5324 | 1981 | PhD | Married | 58293.0 | 1 | 0 | 19-01-2014 | 94 | 6 | 5 |

2. Loại bỏ dữ liệu trùng lặp

```
# Remove duplicates across the columns in our dataset:
df_duplicate = df.drop_duplicates()
# Delete a specified row at index value 1:
df.drop(labels=[1], axis=0)
# Delete a single column
df.drop(labels=['Year_Birth'], axis=1)
```

| | ID | Education | Marital_Status | Income | Kidhome | Teenhome | Dt_Customer | Recency | NumStorePurchases | NumWebVisitsMonth |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 5524 | Graduation | Single | 58138.0 | 0 | 0 | 04-09-2012 | 58 | 4 | 7 |
| **1** | 2174 | Graduation | Single | 46344.0 | 1 | 1 | 08-03-2014 | 38 | 2 | 5 |
| **2** | 4141 | Graduation | Together | 71613.0 | 0 | 0 | 21-08-2013 | 26 | 10 | 4 |
| **3** | 6182 | Graduation | Together | 26646.0 | 1 | 0 | 10-02-2014 | 26 | 4 | 6 |
| **4** | 5324 | PhD | Married | 58293.0 | 1 | 0 | 19-01-2014 | 94 | 6 | 5 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2235** | 10870 | Graduation | Married | 61223.0 | 0 | 1 | 13-06-2013 | 46 | 4 | 5 |
| **2236** | 4001 | PhD | Together | 64014.0 | 2 | 1 | 10-06-2014 | 56 | 5 | 7 |
| **2237** | 7270 | Graduation | Divorced | 56981.0 | 0 | 0 | 25-01-2014 | 91 | 13 | 6 |
| **2238** | 8235 | Master | Together | 69245.0 | 0 | 1 | 24-01-2014 | 8 | 10 | 3 |
| **2239** | 9405 | PhD | Married | 52869.0 | 1 | 1 | 15-10-2012 | 40 | 4 | 7 |

2240 rows × 10 columns

3. Thay thế dữ liệu và thay đổi định dạng của dữ liệu

```
# Replace the values in Teenhome with has teen and has no teen
df['Teenhome_replaced'] = df['Teenhome'].replace([0,1,2],['has no teen','has teen','has teen'])
# Fill NAs in the Income column
df['Income'] = df['Income'].fillna(0)
# Change the data type of the Income column from float to int
df['Income_changed'] = df['Income'].astype(int)

# Xem 5 dòng cuối
df.tail()
```

| | ID | Year_Birth | Education | Marital_Status | Income | Kidhome | Teenhome | Dt_Customer | Recency | NumStorePurchases | NumWebVisitsMonth | Teenhome_replaced | Income_changed |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2235** | 10870 | 1967 | Graduation | Married | 61223.0 | 0 | 1 | 13-06-2013 | 46 | 4 | 5 | has teen | 61223 |
| **2236** | 4001 | 1946 | PhD | Together | 64014.0 | 2 | 1 | 10-06-2014 | 56 | 5 | 7 | has teen | 64014 |
| **2237** | 7270 | 1981 | Graduation | Divorced | 56981.0 | 0 | 0 | 25-01-2014 | 91 | 13 | 6 | has no teen | 56981 |
| **2238** | 8235 | 1956 | Master | Together | 69245.0 | 0 | 1 | 24-01-2014 | 8 | 10 | 3 | has teen | 69245 |
| **2239** | 9405 | 1954 | PhD | Married | 52869.0 | 1 | 1 | 15-10-2012 | 40 | 4 | 7 | has teen | 52869 |

4. Xử lý dữ liệu thiếu

```
# Check for missing values using the isnull and sum methods
df.isnull().sum()
```

```
ID                   0
Year_Birth           0
Education            0
Marital_Status       0
Income               0
Kidhome              0
Teenhome             0
Dt_Customer          0
Recency              0
NumStorePurchases    0
NumWebVisitsMonth    0
Teenhome_replaced    0
Income_changed       0
dtype: int64
```

```
# Drop missing values using the dropna method
marketing_data_withoutna = df.dropna(how = 'any')
marketing_data_withoutna.shape
```

(2240, 13)

### 1.1.3. Bài tập thực hành 1

Thực hiện thống kê mô tả trên tập dữ liệu về phân loại chất lượng rượu đỏ. Dữ liệu lấy tại https://www.kaggle.com/code/eisgandar/red-wine-quality-eda-classification

```
# Import thư viện
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
# 1. Đọc dữ liệu từ file CSV
wine_data = pd.read_csv("winequality-red.csv")

# 2. Xem các thông tin cơ bản
print(wine_data.head(5))
print(wine_data.dtypes)
print(wine_data.shape)
```

```
===== THÔNG TIN DỮ LIỆU =====
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
Kiểu dữ liệu:
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                   int64
dtype: object

Kích thước dữ liệu: (1599, 12)
```

Tập dữ liệu này chứa 1599 mẫu rượu vang đỏ, mỗi mẫu được mô tả bởi 11 đặc trưng hóa học và 1 biến đầu ra (chất lượng).

**CÁC THUỘC TÍNH TRONG TẬP DỮ LIỆU:**

1. **FIXED ACIDITY** – ĐỘ AXIT CỐ ĐỊNH.
2. **VOLATILE ACIDITY** – ĐỘ AXIT BAY HƠI.
3. **CITRIC ACID** – HÀM LƯỢNG AXIT CITRIC.
4. **RESIDUAL SUGAR** – LƯỢNG ĐƯỜNG CÒN LẠI.
5. **CHLORIDES** – HÀM LƯỢNG MUỐI.
6. **FREE SULFUR DIOXIDE** – LƯỢNG KHÍ $SO_2$ TỰ DO.
7. **TOTAL SULFUR DIOXIDE** – LƯỢNG KHÍ $SO_2$ TỔNG.
8. **DENSITY** – MẬT ĐỘ CỦA RƯỢU.
9. **PH** – ĐỘ AXIT (GIÁ TRỊ PH).
10. **SULPHATES** – HỢP CHẤT LƯU HUỲNH GÂY ẢNH HƯỞNG HƯƠNG VỊ.
11. **ALCOHOL** – HÀM LƯỢNG CỒN.
12. **QUALITY** – ĐIỂM CHẤT LƯỢNG (TỪ 0 ĐẾN 10).

```
# 3. Thực hiện thống kê mô tả cho cột "alcohol"
data_mean = np.mean(wine_data["alcohol"])
data_median = np.median(wine_data["alcohol"])
data_mode = stats.mode(wine_data["alcohol"], keepdims=True)
data_variance = np.var(wine_data["alcohol"])
data_sd = np.std(wine_data["alcohol"])
data_max = np.max(wine_data["alcohol"])
data_min = np.min(wine_data["alcohol"])
data_percentile = np.percentile(wine_data["alcohol"], 60)
data_quartile = np.quantile(wine_data["alcohol"], 0.75)
data_IQR = stats.iqr(wine_data["alcohol"], nan_policy='omit')

# 4. In kết quả
print("\n===== KẾT QUẢ THỐNG KÊ TRÊN CỘT 'ALCOHOL' =====")
print("Mean (Trung bình):", data_mean)
print("Median (Trung vị):", data_median)
print("Mode (Giá trị xuất hiện nhiều nhất):", data_mode.mode[0])
print("Variance (Phương sai):", data_variance)
print("Standard Deviation (Độ lệch chuẩn):", data_sd)
print("Max:", data_max)print("Min:", data_min)
print("60th Percentile:", data_percentile)
print("3rd Quartile (Q3):", data_quartile)
print("IQR:", data_IQR)

# 5. Thống kê mô tả tổng quan toàn dataset
print("\n===== THỐNG KÊ MÔ TẢ TỔNG QUAN =====")
print(wine_data.describe())
```

```
===== KẾT QUẢ THỐNG KÊ TRÊN CỘT 'ALCOHOL' =====
Mean (Trung bình): 10.422983114446529
Median (Trung vị): 10.2
Mode (Giá trị xuất hiện nhiều nhất): 9.5
Variance (Phương sai): 1.1349371714888994
Standard Deviation (Độ lệch chuẩn): 1.0653343003437463
Max: 14.9
Min: 8.4
60th Percentile: 10.5
3rd Quartile (Q3): 11.1
IQR: 1.5999999999999996
```

```
===== THỐNG KÊ MÔ TẢ TỔNG QUAN =====
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1599.000000       1599.000000  1599.000000     1599.000000
mean        8.319637          0.527821     0.270976        2.538806
std         1.741096          0.179060     0.194801        1.409928
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.390000     0.090000        1.900000
50%         7.900000          0.520000     0.260000        2.200000
75%         9.200000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  1599.000000          1599.000000           1599.000000  1599.000000
mean      0.087467            15.874922             46.467792     0.996747
std       0.047065            10.460157             32.895324     0.001887
min       0.012000             1.000000              6.000000     0.990070
25%       0.070000             7.000000             22.000000     0.995600
50%       0.079000            14.000000             38.000000     0.996750
75%       0.090000            21.000000             62.000000     0.997835
max       0.611000            72.000000            289.000000     1.003690

                pH    sulphates      alcohol      quality
count  1599.000000  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983     5.636023
std       0.154386     0.169507     1.065668     0.807569
min       2.740000     0.330000     8.400000     3.000000
25%       3.210000     0.550000     9.500000     5.000000
50%       3.310000     0.620000    10.200000     6.000000
75%       3.400000     0.730000    11.100000     6.000000
max       4.010000     2.000000    14.900000     8.000000
```

### 1.1.4. Bài tập thực hành 2

Thực hiện thống kê mô tả trên tập dữ liệu về bệnh tiểu đường. Dữ liệu lấy tại https://www.kaggle.com/code/vincentlugat/pima-indians-diabetes-eda-prediction-0-906

```
import pandas as pd
import numpy as np
from scipy import stats
# Đọc file dữ liệu
data = pd.read_csv("pima_indians_diabetes.csv")

# Xem vài dòng đầu để kiểm tra
display(data.head(5))

# Thống kê mô tả cơ bản (pandas có sẵn)
basic_stats = data.describe().T
display(basic_stats)
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| **1** | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| **2** | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| **3** | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| **4** | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Pregnancies** | 768.0 | 3.845052 | 3.369578 | 0.000 | 1.00000 | 3.0000 | 6.00000 | 17.00 |
| **Glucose** | 768.0 | 120.894531 | 31.972618 | 0.000 | 99.00000 | 117.0000 | 140.25000 | 199.00 |
| **BloodPressure** | 768.0 | 69.105469 | 19.355807 | 0.000 | 62.00000 | 72.0000 | 80.00000 | 122.00 |
| **SkinThickness** | 768.0 | 20.536458 | 15.952218 | 0.000 | 0.00000 | 23.0000 | 32.00000 | 99.00 |
| **Insulin** | 768.0 | 79.799479 | 115.244002 | 0.000 | 0.00000 | 30.5000 | 127.25000 | 846.00 |
| **BMI** | 768.0 | 31.992578 | 7.884160 | 0.000 | 27.30000 | 32.0000 | 36.60000 | 67.10 |
| **DiabetesPedigreeFunction** | 768.0 | 0.471876 | 0.331329 | 0.078 | 0.24375 | 0.3725 | 0.62625 | 2.42 |
| **Age** | 768.0 | 33.240885 | 11.760232 | 21.000 | 24.00000 | 29.0000 | 41.00000 | 81.00 |
| **Outcome** | 768.0 | 0.348958 | 0.476951 | 0.000 | 0.00000 | 0.0000 | 1.00000 | 1.00 |

Tập dữ liệu này chứa thông tin về phụ nữ gốc da đỏ Pima, với 768 mẫu quan sát. Mỗi mẫu mô tả các chỉ số sức khỏe cơ bản liên quan đến khả năng mắc bệnh tiểu đường, được biểu diễn qua 8 đặc trưng đầu vào và 1 biến đầu ra (kết quả chẩn đoán).

## CÁC THUỘC TÍNH TRONG TẬP DỮ LIỆU:

**Pregnancies**: Số lần mang thai.

**Glucose**: Mức glucose huyết tương sau khi ăn (mg/dL).

**BloodPressure**: Huyết áp tâm trương (mm Hg).

**SkinThickness**: Độ dày nếp gấp da (mm).

**Insulin**: Nồng độ insulin huyết tương (µU/mL).

**BMI**: Chỉ số khối cơ thể (Body Mass Index).

**DiabetesPedigreeFunction**: Chỉ số di truyền liên quan đến bệnh tiểu đường.

**Age**: Tuổi của đối tượng (năm).

**Outcome**: Kết quả chẩn đoán (1: có tiểu đường, 0: không).

```
# Tính thêm các chỉ số thống kê nâng cao
advanced_stats = pd.DataFrame({
    "Mean": data.mean(),
    "Median": data.median(),
    "Mode": data.mode().iloc[0],
    "Variance": data.var(),
    "Standard Deviation": data.std(),
    "Range": data.max() - data.min(),
    "25th Percentile (Q1)": data.quantile(0.25),
    "50th Percentile (Q2)": data.quantile(0.5),
    "75th Percentile (Q3)": data.quantile(0.75),
    "IQR": data.quantile(0.75) - data.quantile(0.25)
})
display(advanced_stats.round(2))
```

| | Mean | Median | Mode | Variance | Standard Deviation | Range | 25th Percentile (Q1) | 50th Percentile (Q2) | 75th Percentile (Q3) | IQR |
|---|---|---|---|---|---|---|---|---|---|---|
| **Pregnancies** | 3.85 | 3.00 | 1.00 | 11.35 | 3.37 | 17.00 | 1.00 | 3.00 | 6.00 | 5.00 |
| **Glucose** | 120.89 | 117.00 | 99.00 | 1022.25 | 31.97 | 199.00 | 99.00 | 117.00 | 140.25 | 41.25 |
| **BloodPressure** | 69.11 | 72.00 | 70.00 | 374.65 | 19.36 | 122.00 | 62.00 | 72.00 | 80.00 | 18.00 |
| **SkinThickness** | 20.54 | 23.00 | 0.00 | 254.47 | 15.95 | 99.00 | 0.00 | 23.00 | 32.00 | 32.00 |
| **Insulin** | 79.80 | 30.50 | 0.00 | 13281.18 | 115.24 | 846.00 | 0.00 | 30.50 | 127.25 | 127.25 |
| **BMI** | 31.99 | 32.00 | 32.00 | 62.16 | 7.88 | 67.10 | 27.30 | 32.00 | 36.60 | 9.30 |
| **DiabetesPedigreeFunction** | 0.47 | 0.37 | 0.25 | 0.11 | 0.33 | 2.34 | 0.24 | 0.37 | 0.63 | 0.38 |
| **Age** | 33.24 | 29.00 | 22.00 | 138.30 | 11.76 | 60.00 | 24.00 | 29.00 | 41.00 | 17.00 |
| **Outcome** | 0.35 | 0.00 | 0.00 | 0.23 | 0.48 | 1.00 | 0.00 | 0.00 | 1.00 | 1.00 |

# 1.2. XỬ LÝ VÀ TRỰC QUAN HÓA DỮ LIỆU

## 1.2.1. Ôn tập lý thuyết

**+ Trực quan hóa dữ liệu có vai trò gì trong phân tích dữ liệu? Tại sao nó quan trọng trong khám phá dữ liệu (EDA)?**

- Trực quan hóa dữ liệu giúp biểu diễn dữ liệu bằng biểu đồ, đồ thị, hình ảnh để dễ hiểu và dễ phân tích hơn.
- Vai trò:
  - Giúp nhận biết nhanh xu hướng, mẫu hình, mối quan hệ và điểm bất thường trong dữ liệu.
  - Hỗ trợ so sánh giữa các nhóm hoặc biến khác nhau.

- Giúp truyền đạt kết quả phân tích rõ ràng và thuyết phục hơn.

- Tầm quan trọng trong EDA:
  - Là bước quan trọng để hiểu dữ liệu trước khi xây dựng mô hình.
  - Giúp phát hiện sai sót, dữ liệu thiếu, ngoại lệ.
  - Giúp định hướng chọn phương pháp phân tích phù hợp.

**+ Các loại biểu đồ phổ biến (như histogram, scatter plot, boxplot, bar chart) được sử dụng trong các trường hợp nào?**

- Histogram (biểu đồ tần suất): Dùng để xem phân bố của một biến liên tục (ví dụ: chiều cao, cân nặng, điểm số).
- Scatter plot (biểu đồ phân tán): Dùng để quan sát mối quan hệ giữa hai biến liên tục (ví dụ: doanh thu và chi phí).
- Boxplot (biểu đồ hộp): Dùng để so sánh phân bố và phát hiện giá trị ngoại lệ giữa các nhóm dữ liệu.
- Bar chart (biểu đồ cột): Dùng để so sánh giá trị giữa các nhóm hoặc danh mục (ví dụ: doanh số theo khu vực).

**+ Làm thế nào để chọn loại biểu đồ phù hợp với đặc điểm của dữ liệu (ví dụ: dữ liệu phân loại, dữ liệu số, dữ liệu thời gian)?**

- Dữ liệu phân loại → Bar chart, Pie chart – so sánh nhóm.
- Dữ liệu số → Histogram, Boxplot – phân bố, ngoại lệ.
- Dữ liệu thời gian → Line chart, Area chart – xu hướng theo thời gian.
- Hai biến liên tục → Scatter plot – mối tương quan.

- Nguyên tắc: Chọn biểu đồ phù hợp với loại dữ liệu và mục tiêu phân tích.

**+ Sự khác biệt giữa các thư viện trực quan hóa trong Python như Matplotlib, Seaborn và Plotly là gì?**

- Matplotlib: Cơ bản, linh hoạt, tùy chỉnh chi tiết, biểu đồ tĩnh.
- Seaborn: Dễ dùng, giao diện đẹp, tích hợp Pandas, biểu đồ thống kê.

➢ Plotly: Biểu đồ tương tác, hiện đại, dùng cho dashboard hoặc web.

- Matplotlib → nền tảng cơ bản; Seaborn → EDA nhanh, đẹp; Plotly → trình bày tương tác.

**+ Những nguyên tắc thiết kế nào cần tuân thủ để tạo ra một biểu đồ trực quan hóa dễ hiểu và hiệu quả?**

1) Xác định rõ thông điệp chính: Biểu đồ phải truyền tải một ý cụ thể, không chỉ đơn thuần là "cho có hình".
2) Chọn loại biểu đồ phù hợp: Dựa vào loại dữ liệu (phân loại, số, thời gian) và mối quan hệ cần thể hiện.
3) Giữ thiết kế đơn giản nhưng đầy đủ: Loại bỏ chi tiết thừa, song vẫn đảm bảo người xem hiểu trọn vẹn nội dung.
4) Dùng màu sắc có chủ đích: Nhất quán giữa các biểu đồ, tránh lạm dụng màu quá sặc sỡ và lưu ý người bị mù màu.
5) Gắn nhãn, tiêu đề rõ ràng: Mỗi trục, đơn vị và tiêu đề phải dễ đọc, dễ hiểu.
6) Trung thực về tỷ lệ trục: Không cắt hoặc bóp méo trục làm sai lệch cảm nhận dữ liệu.
7) Làm nổi bật điểm chính: Dùng màu, kích thước hoặc chú thích để nhấn mạnh nội dung quan trọng.
8) Kiểm tra khả năng đọc hiểu: Hiển thị thử trên nhiều thiết bị, cỡ màn hình, và nhờ người khác xem có dễ hiểu không.

**+ Làm thế nào để tạo một biểu đồ đơn giản như histogram hoặc bar chart bằng Matplotlib? Bạn có thể chia sẻ đoạn code mẫu không?**

```{Python}
import matplotlib.pyplot as plt
import numpy as np
# Histogram
scores = np.random.normal(70, 10, 100)
plt.hist(scores, bins=10, color='skyblue', edgecolor='black')
plt.title('Phân bố điểm thi')
plt.xlabel('Điểm')
```

```
plt.ylabel('Tần suất')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
# Bar chart
categories = ['A', 'B', 'C', 'D']
values = [23, 45, 12, 30]
plt.bar(categories, values, color='lightgreen', edgecolor='black')
plt.title('Doanh số theo sản phẩm')
plt.xlabel('Sản phẩm')
plt.ylabel('Doanh số')
plt.ylim(0, max(values)+10)
plt.tight_layout()
plt.show()
```

**+ Làm thế nào để xuất biểu đồ từ Python ra các định dạng như PNG, PDF hoặc HTML để sử dụng trong báo cáo?**

- Matplotlib:
  - Dùng plt.savefig('tenfile.png', dpi=300) để lưu biểu đồ (nên gọi trước plt.show()).
  - Hỗ trợ nhiều định dạng: PNG, PDF, SVG, JPG...
  - Plotly:
  - Dùng fig.write_html('tenfile.html') để lưu biểu đồ tương tác.
  - Dùng fig.write_image('tenfile.png') để lưu ảnh tĩnh (cần cài kaleido).
- Định dạng gợi ý:
  - PNG/JPG → báo cáo, trình chiếu.
  - PDF → in ấn, lưu trữ chất lượng cao.
  - HTML → dashboard, web tương tác.
  - SVG → dùng cho web, in vector sắc nét.

## 1.2.2. Bài làm mẫu

**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 lấy

từ https://www.kaggle.com/datasets/thomasnibb/amsterdam-house-price-prediction

**Nhiệm vụ 1**:

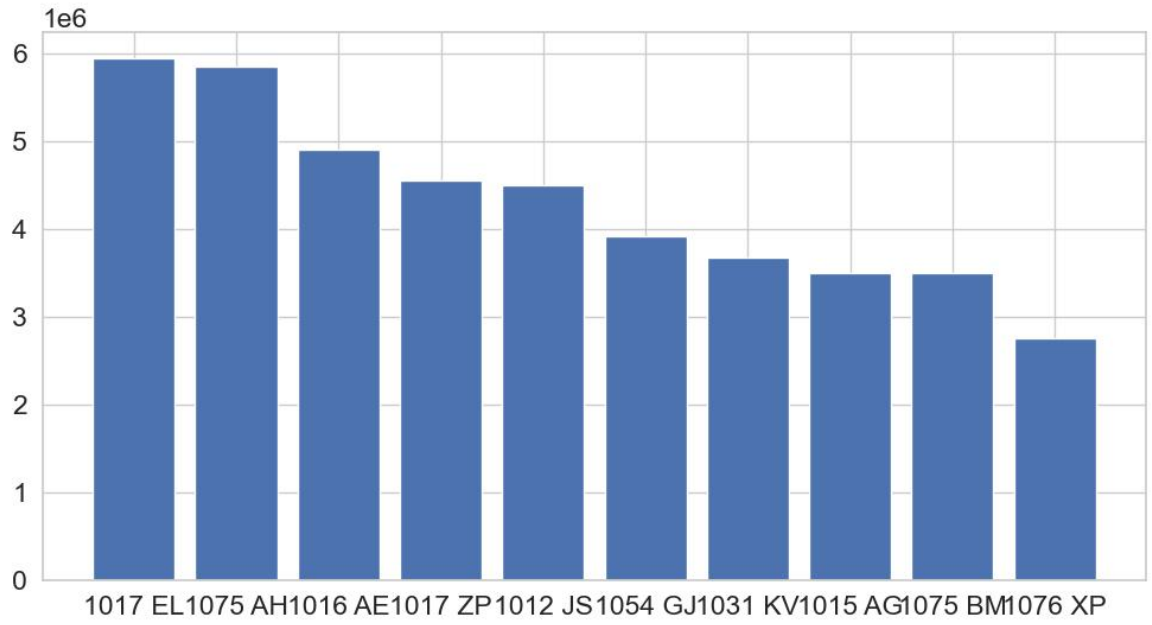
1. Chuẩn bị dữ liệu cho trực quan hóa dữ liệu

```
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
houseprices_data = pd.read_csv("amsterdam_house_price.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']
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 |

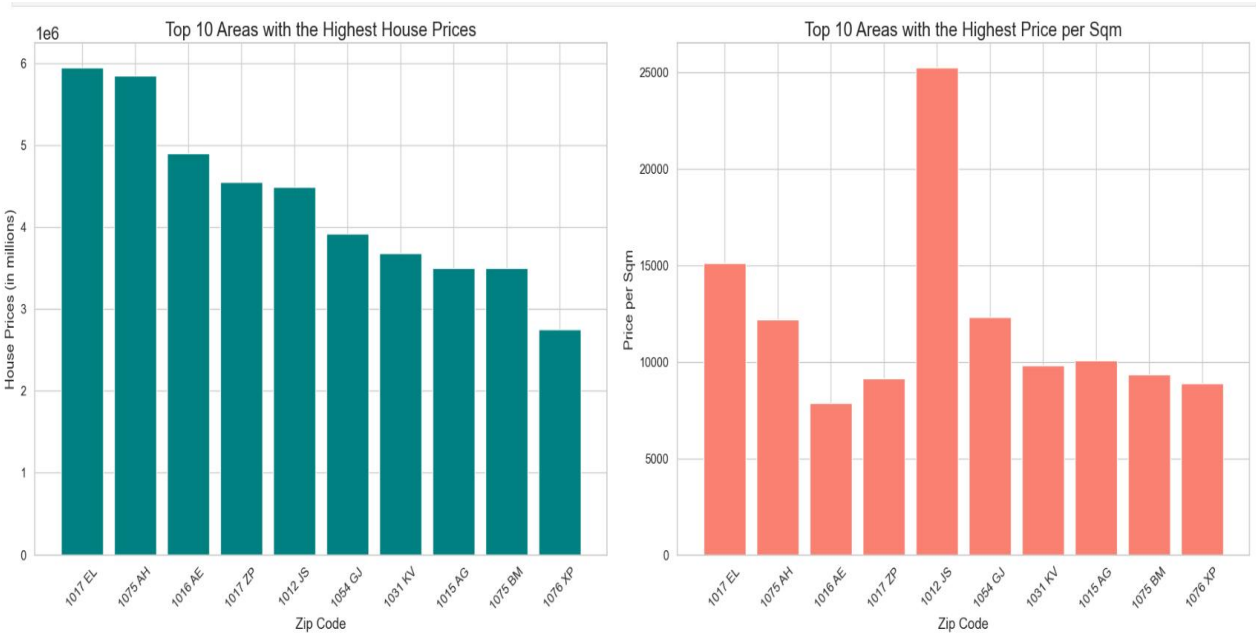
2. Trực quan hóa dữ liệu với thư viện **Matplotlib**

```
# case 1: basic
plt.figure(figsize= (12,6))
x = houseprices_sorted['Zip'][0:10]
y = houseprices_sorted['Price'][0:10]
plt.bar(x,y)
plt.show()
```



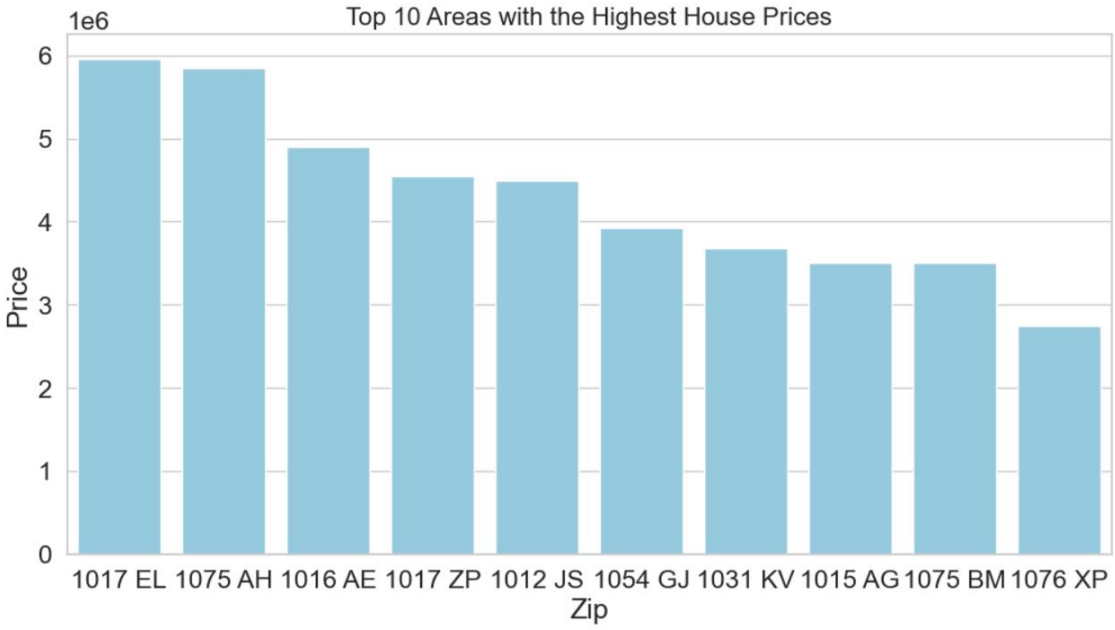

```
# case 2: advanced 1
plt.figure(figsize=(12,6))
plt.bar(x, y, color='skyblue')
plt.title('Top 10 Areas with the Highest House Prices', fontsize=15)
plt.xlabel('Zip Code', fontsize=12)
plt.xticks(fontsize=10, rotation=45)
plt.ylabel('House Prices (in millions)', fontsize=12)
plt.yticks(fontsize=10)
plt.tight_layout()
plt.show()
```

```
# case 3: advanced 2 ---
fig, axes = plt.subplots(1, 2, figsize=(20, 8))

# Biểu đồ 1: Giá nhà cao nhất
axes[0].bar(x, y, color='teal')
axes[0].set_title('Top 10 Areas with the Highest House Prices', fontsize=18)
axes[0].set_xlabel('Zip Code', fontsize=14)
axes[0].set_ylabel('House Prices (in millions)', fontsize=14)
axes[0].tick_params(axis='x', labelrotation=45, labelsize=12)
axes[0].tick_params(axis='y', labelsize=12)

# Biểu đồ 2: Giá trên mỗi mét vuông
y1 = houseprices_sorted['PriceperSqm'].head(10)
axes[1].bar(x, y1, color='salmon')
axes[1].set_title('Top 10 Areas with the Highest Price per Sqm', fontsize=18)
axes[1].set_xlabel('Zip Code', fontsize=14)
axes[1].set_ylabel('Price per Sqm', fontsize=14)
axes[1].tick_params(axis='x', labelrotation=45, labelsize=12)
axes[1].tick_params(axis='y', labelsize=12)

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

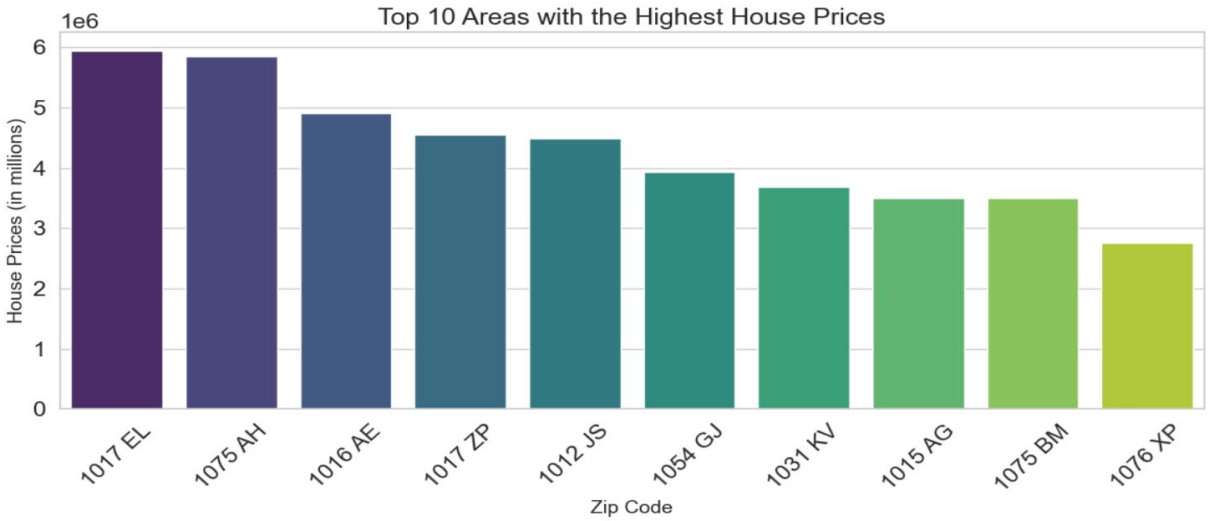



3. Trực quan hóa dữ liệu với thư viện **Seaborn**

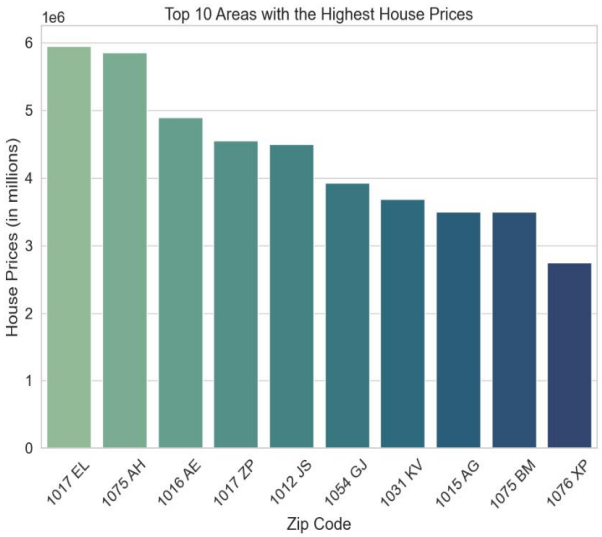
```
import seaborn as sns
# case 1: basic
plt.figure(figsize=(12,6))
houseprices_data=houseprices_sorted.head(10)
sns.barplot(data=houseprices_data, x='Zip', y='Price', color='skyblue')
plt.title("Top 10 Areas with the Highest House Prices", fontsize=16)
plt.show()
```

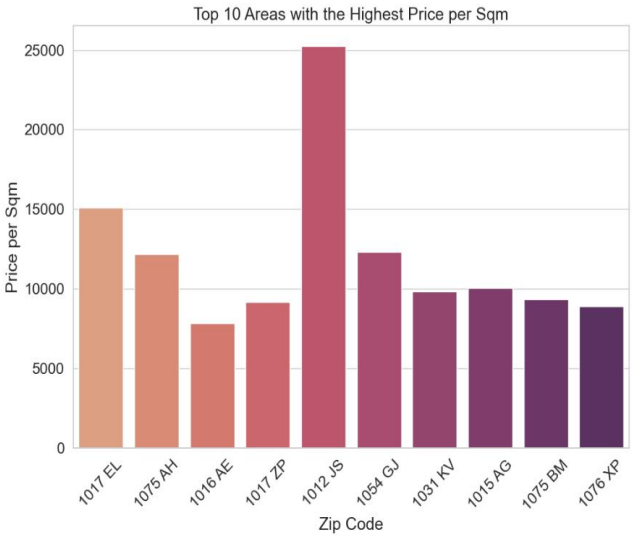



```
# case 2: advanced 1
plt.figure(figsize=(12,6))
houseprices_data = houseprices_sorted.head(10)
ax = sns.barplot(data=houseprices_data, x='Zip', y='Price', palette='viridis')
ax.set_xlabel('Zip Code', fontsize=14)
ax.set_ylabel('House Prices (in millions)', fontsize=14)
ax.set_title('Top 10 Areas with the Highest House Prices', fontsize=18)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
# case 3
fig, axes = plt.subplots(1, 2, figsize=(20,8))
houseprices_data = houseprices_sorted.head(10)
sns.set(font_scale=1.5, style="whitegrid")
# Biểu đồ 1
sns.barplot(data = houseprices_data, x='Zip', y='Price', ax=axes[0],
palette='crest')
axes[0].set_xlabel('Zip Code')
axes[0].set_ylabel('House Prices (in millions)')
axes[0].set_title('Top 10 Areas with the Highest House Prices')
axes[0].tick_params(axis='x', rotation=45)
# Biểu đồ 2
sns.barplot(data = houseprices_data, x='Zip', y='PriceperSqm', ax=axes[1],
palette='flare')
axes[1].set_xlabel('Zip Code')
axes[1].set_ylabel('Price per Sqm')
axes[1].set_title('Top 10 Areas with the Highest Price per Sqm')
axes[1].tick_params(axis='x', rotation=45)
plt.tight_layout()
plt.show()
```




## 1.2.3. Bài tập thực hành 1

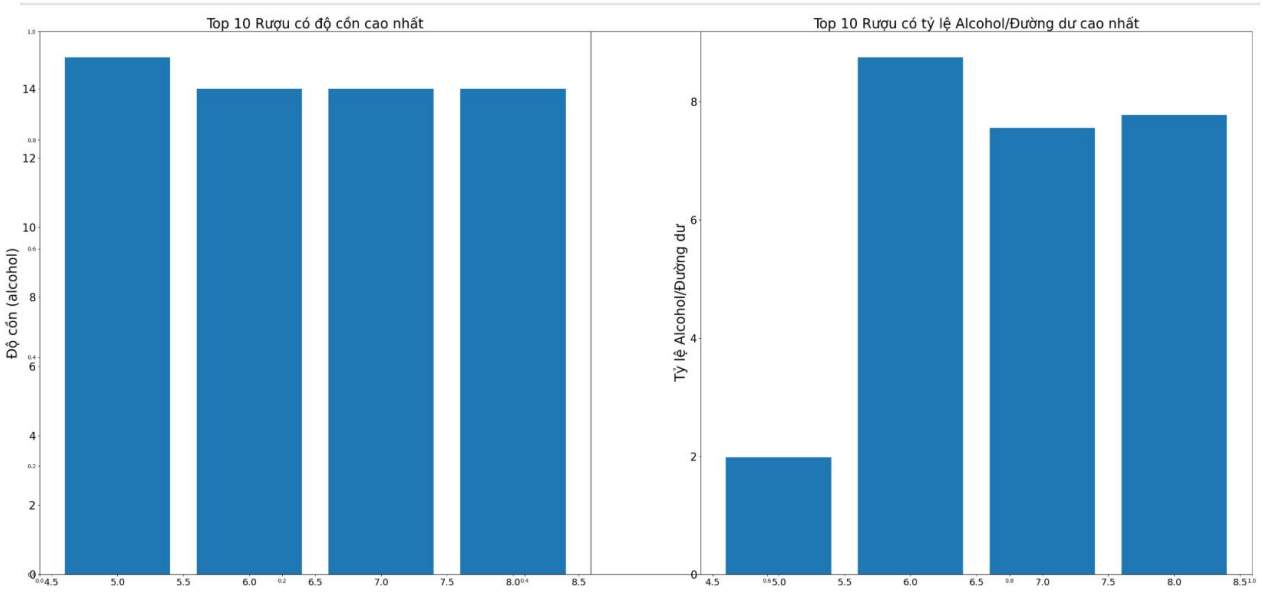
+ Thực hiện trực quan hóa dữ liệu trên tập dữ liệu về phân loại chất lượng rượu đỏ. Dữ liệu lấy tại https://www.kaggle.com/code/eisgandar/red-wine-quality-eda-classification

```
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
# Đọc dữ liệu từ file CSV
wine_data = pd.read_csv("winequality-red.csv")
# Xem thông tin cơ bản
print("===== THÔNG TIN DỮ LIỆU =====")
print(wine_data.head(5))
print("\nKiểu dữ liệu:")
print(wine_data.dtypes)
print("\nKích thước dữ liệu:", wine_data.shape)
```

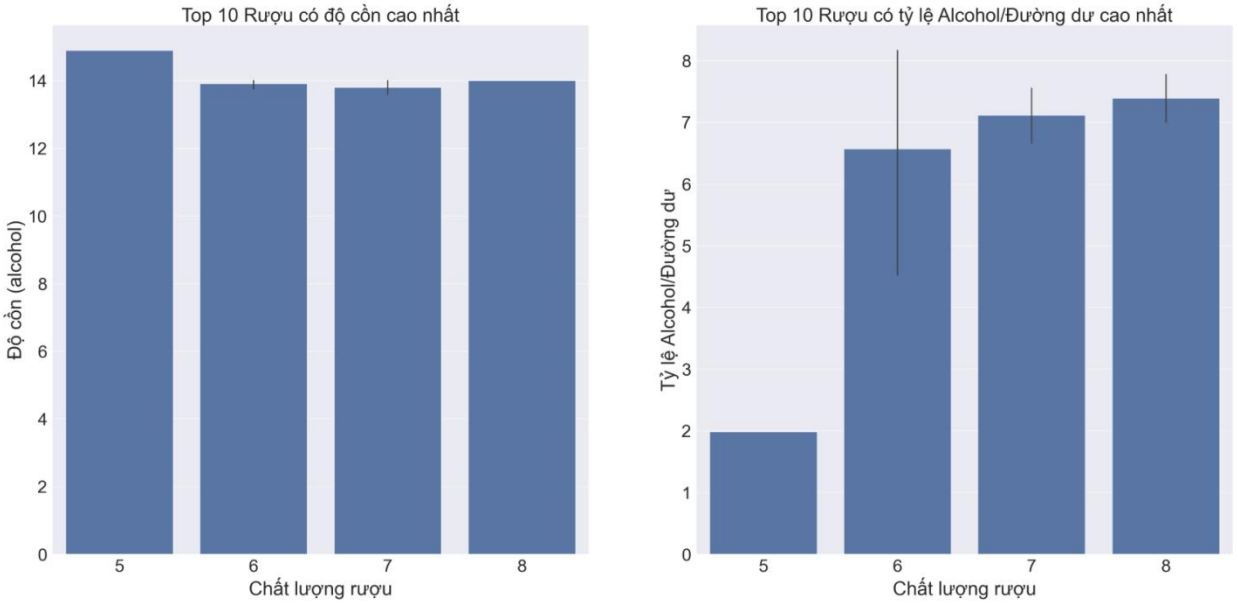
```
===== THÔNG TIN DỮ LIỆU =====
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
```

```
Kiểu dữ liệu:
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                   int64
dtype: object

Kích thước dữ liệu: (1599, 12)
```

```
# Chuẩn bị dữ liệu cho biểu đồ
wine_data = wine_data[['quality', 'alcohol', 'citric acid', 'residual sugar']]
wine_data['AlcoholRatio'] = wine_data['alcohol'] / wine_data['residual sugar']
# Sắp xếp theo độ cồn giảm dần
wine_sorted = wine_data.sort_values('alcohol', ascending=False)

# Vẽ hình bằng Matplotlib
fig, ax = plt.subplots(figsize=(40,18))
x = wine_sorted['quality'][0:10]
y = wine_sorted['alcohol'][0:10]
y1 = wine_sorted['AlcoholRatio'][0:10]

plt.subplot(1,2,1)
plt.bar(x, y)
plt.xticks(fontsize=17)
plt.ylabel('Độ cồn (alcohol)', fontsize=25)
plt.yticks(fontsize=20)
plt.title('Top 10 Rượu có độ cồn cao nhất', fontsize=25)

plt.subplot(1,2,2)
plt.bar(x, y1)
plt.xticks(fontsize=17)
plt.ylabel('Tỷ lệ Alcohol/Đường dư', fontsize=25)
plt.yticks(fontsize=20)
plt.title('Top 10 Rượu có tỷ lệ Alcohol/Đường dư cao nhất', fontsize=25)
plt.show()
```



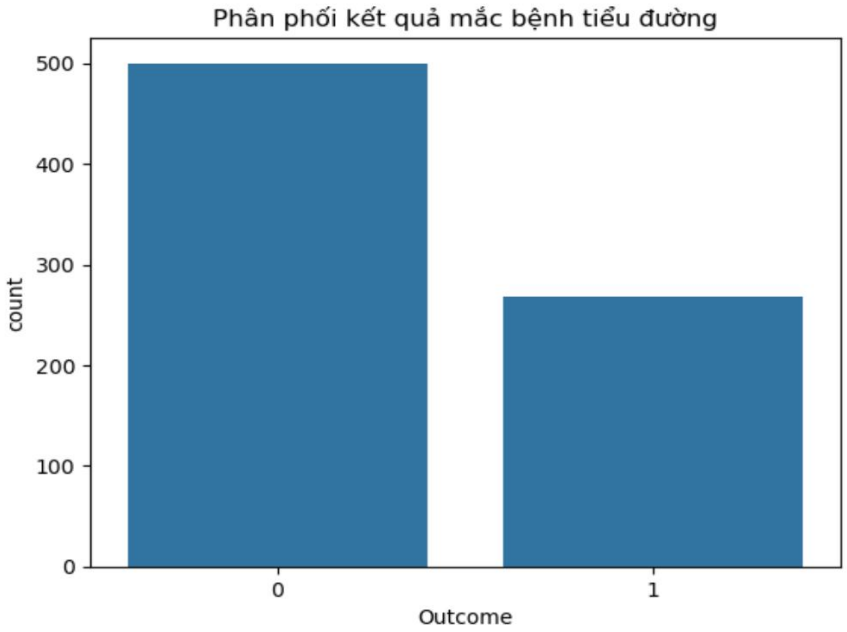

```
# Vẽ hình bằng Seaborn
fig, ax = plt.subplots(1, 2, figsize=(40,18))
data = wine_sorted[0:10]
sns.set(font_scale=3)
ax1 = sns.barplot(data=data, x='quality', y='alcohol', ax=ax[0])
ax1.set_xlabel('Chất lượng rượu')
ax1.set_ylabel('Độ cồn (alcohol)')
ax1.set_title('Top 10 Rượu có độ cồn cao nhất')
ax2 = sns.barplot(data=data, x='quality', y='AlcoholRatio', ax=ax[1])
ax2.set_xlabel('Chất lượng rượu')
ax2.set_ylabel('Tỷ lệ Alcohol/Đường dư')
ax2.set_title('Top 10 Rượu có tỷ lệ Alcohol/Đường dư cao nhất')
plt.show()
```



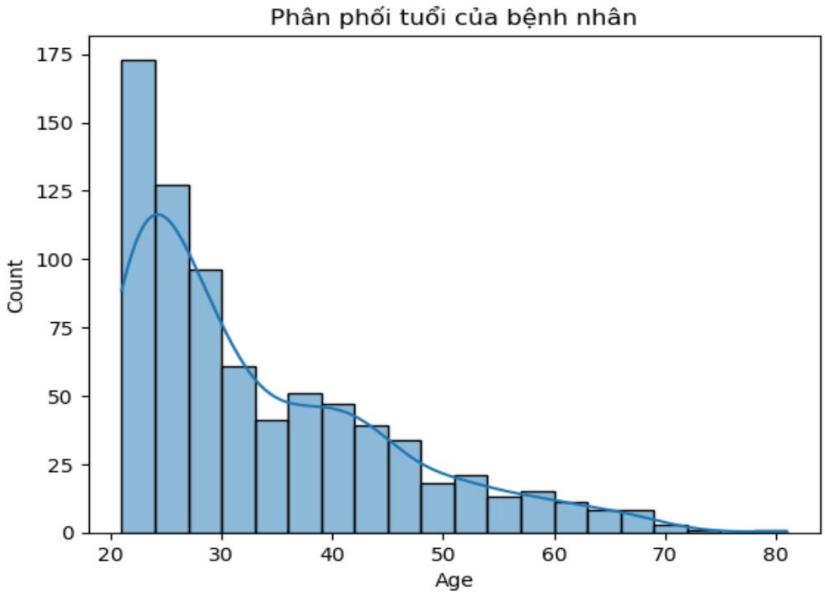
### 1.2.4. Bài tập thực hành 2

+ Thực hiện trực quan hóa dữ liệu trên tập dữ liệu về bệnh tiểu đường.
Dữ liệu lấy tại https://www.kaggle.com/code/vincentlugat/pima-indians-diabetes-eda-prediction-0-906

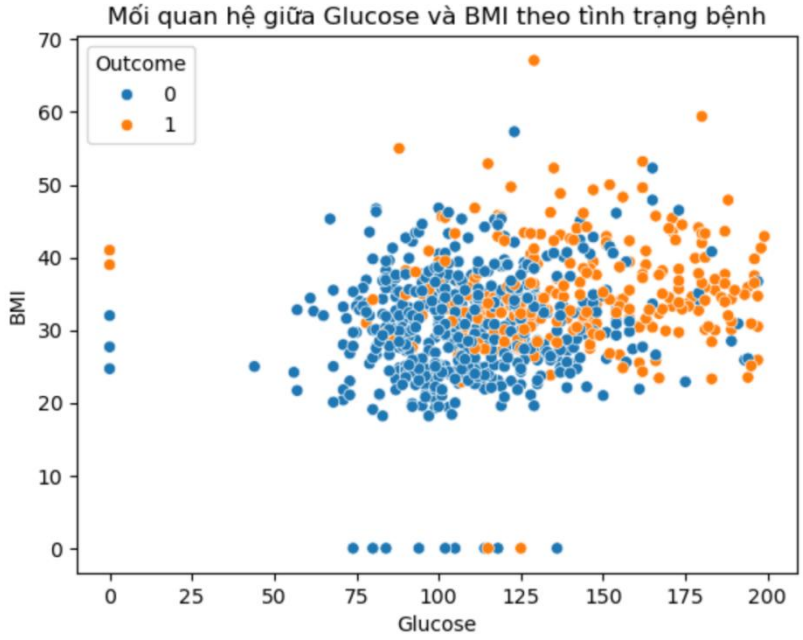
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Đọc dữ liệu
df = pd.read_csv("pima_indians_diabetes.csv")
# Xem vài dòng đầu
print(df.head(5))
print(df.shape)

# Phân phối biến theo kết quả mắc bệnh (Outcome)
sns.countplot(x="Outcome", data=df)
plt.title("Phân phối kết quả mắc bệnh tiểu đường")
plt.show()
```



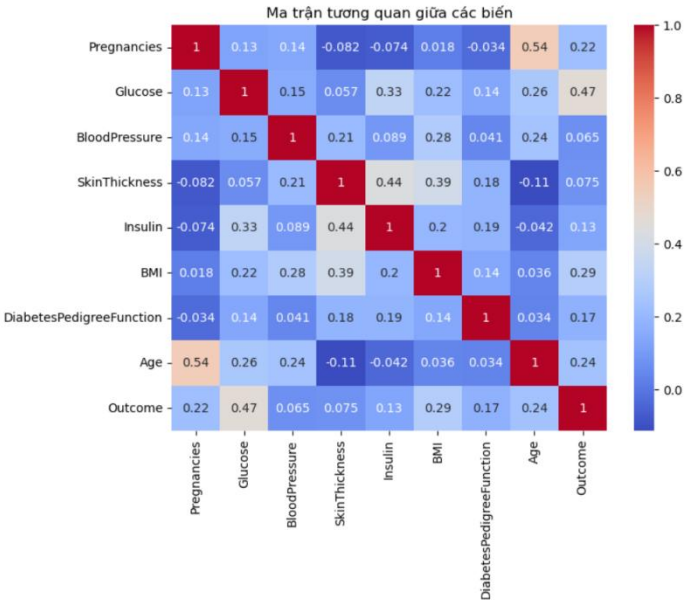

```
# Biểu đồ phân phối tuổi (Age)
sns.histplot(df["Age"], bins=20, kde=True)
plt.title("Phân phối tuổi của bệnh nhân")
plt.show()
```



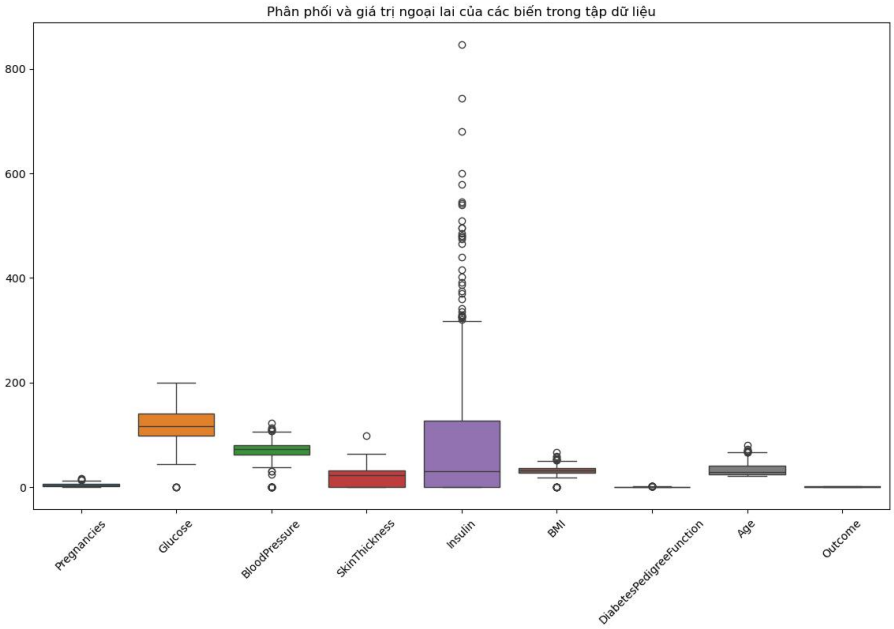

```
# Mối quan hệ giữa Glucose và BMI
sns.scatterplot(x="Glucose", y="BMI", hue="Outcome", data=df)
plt.title("Mối quan hệ giữa Glucose và BMI theo tình trạng bệnh")
plt.show()
```




```
# Ma trận tương quan
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Ma trận tương quan giữa các biến")
plt.show()
```




```
# Boxplot tổng hợp các biến số
plt.figure(figsize=(14, 8))
sns.boxplot(data=df)
plt.title("Phân phối và giá trị ngoại lai của các biến trong tập dữ liệu")
plt.xticks(rotation=45)
plt.show()
```




+ Thực hiện EDA trên tập dữ liệu mua sắm tại siêu thị. Tập dữ liệu lấy từ https://www.kaggle.com/code/rajatkumar30/eda-online-retail

**Xem ở thư mục Code, file có tên là online_retail_eda.html.**

# 1.3. PHÂN TÍCH ĐƠN BIẾN VÀ HAI BIẾN

## 1.3.1. Ôn tập lý thuyết

**+ 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 (Univariate analysis):
  - Xem xét một biến duy nhất để hiểu phân bố, xu hướng trung tâm, độ phân tán.
  - Dùng các thước đo như mean, median, mode, variance, std hoặc biểu đồ histogram, boxplot.

  + Ví dụ: Phân tích phân bố điểm thi của học sinh.
- Phân tích hai biến (Bivariate analysis):
  - Xem xét mối quan hệ giữa hai biến (độc lập và phụ thuộc).
  - Dùng biểu đồ scatter plot, heatmap, crosstab, correlation.

  + Ví dụ: Mối liên hệ giữa giờ học và điểm thi.
- Khác biệt:
  - Đơn biến: mô tả đặc tính của một biến riêng lẻ.
  - Hai biến: tìm hiểu mối quan hệ hoặc tương quan giữa hai biến.

**+ Các thước đo thống kê nào thường được sử dụng trong phân tích đơn biến (ví dụ: trung bình, trung vị, mode, độ lệch chuẩn)?**

- Trung bình (Mean): Giá trị đại diện cho xu hướng trung tâm của dữ liệu.
- Trung vị (Median): Giá trị giữa khi sắp xếp dữ liệu, ít bị ảnh hưởng bởi outliers.
- Mode (Giá trị mode): Giá trị xuất hiện nhiều nhất, dùng cho cả dữ liệu số và phân loại.
- Phương sai (Variance): Đo mức độ phân tán của dữ liệu quanh trung bình.
- Độ lệch chuẩn (Standard deviation): Căn bậc hai của phương sai, dễ diễn giải hơn.
- Min – Max – Range: Cho biết khoảng giá trị dữ liệu.

- IQR (Khoảng tứ phân vị): Đo độ phân tán của 50% dữ liệu trung tâm.

**+ Trong phân tích hai biến, làm thế nào để xác định mối quan hệ giữa hai biến (ví dụ: tương quan, nhân quả)?**

- Trong phân tích hai biến, ta cần xác định xem hai biến có mối quan hệ với nhau hay không, và nếu có thì mối quan hệ đó là như thế nào.
  - Nếu cả hai biến đều là biến số (numeric), ta thường dùng hệ số tương quan (correlation) để đo mức độ liên hệ tuyến tính giữa chúng. Ví dụ, khi nhiệt độ tăng thì doanh số bán kem cũng tăng – điều này thể hiện mối tương quan dương. Ngoài ra, ta có thể vẽ biểu đồ scatter plot để quan sát trực quan xu hướng tăng hay giảm giữa hai biến.
  - Nếu một biến là biến số và biến còn lại là biến phân loại (categorical), ta có thể dùng biểu đồ boxplot hoặc violin plot để so sánh sự khác biệt về phân bố giá trị giữa các nhóm.
  - Còn nếu cả hai biến đều là biến phân loại, ta thường sử dụng bảng chéo (crosstab) hoặc kiểm định Chi-square để xem chúng có phụ thuộc lẫn nhau không.
- Cần phân biệt rõ giữa tương quan (correlation) và nhân quả (causation):
  - Tương quan chỉ cho biết hai biến thay đổi cùng nhau, nhưng không nói biến nào gây ra biến nào.
  - Nhân quả là khi một biến thật sự ảnh hưởng đến biến còn lại. Ví dụ, nhiệt độ tăng làm doanh số kem tăng là mối quan hệ nhân quả; còn mối liên hệ giữa doanh số kem và số người đuối nước chỉ là tương quan, không phải nhân quả.
- Trong thực tế, ta thường xác định tương quan trước để hiểu mối liên hệ, rồi mới phân tích sâu về nhân quả bằng các mô hình hoặc thí nghiệm kiểm chứng.

**+ Sự khác biệt giữa tương quan (correlation) và hiệp biến (covariance) trong phân tích hai biến là gì?**

- Hiệp biến (Covariance) và tương quan (Correlation) đều đo mối quan hệ giữa hai biến, nhưng khác nhau ở cách thể hiện và khả năng so sánh.

➢ Hiệp biến cho biết hai biến thay đổi cùng chiều hay ngược chiều. Nếu giá trị dương → cùng chiều, âm → ngược chiều. Tuy nhiên, hiệp biến phụ thuộc vào đơn vị đo, nên khó so sánh giữa các cặp biến khác nhau.

➢ Tương quan là phiên bản chuẩn hóa của hiệp biến, có giá trị nằm trong khoảng từ −1 đến +1.

- +1: mối quan hệ tuyến tính hoàn hảo cùng chiều.
- −1: mối quan hệ tuyến tính hoàn hảo ngược chiều.
- 0: không có quan hệ tuyến tính.

- Nói ngắn gọn:

➢ Covariance cho biết xu hướng biến động cùng hay ngược chiều,

➢ Còn Correlation cho biết mức độ mạnh yếu của mối quan hệ tuyến tính (và dễ so sánh hơn).

**+ 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 (Univariate): Dùng để xem phân bố và đặc trưng của một biến duy nhất. Thường sử dụng khi muốn hiểu dữ liệu tổng quan hoặc phát hiện giá trị ngoại lai.

+ Ví dụ:

- Histogram: cho biết dạng phân bố (chuẩn, lệch trái, lệch phải).
- Boxplot: thể hiện trung vị, tứ phân vị và outlier.
- Bar chart: dùng cho dữ liệu phân loại để thấy tần suất hoặc tỷ lệ từng nhóm.

- Phân tích hai biến (Bivariate): Dùng để tìm hiểu mối quan hệ giữa hai biến, xem chúng có liên quan hay không và mức độ ra sao.

+ Ví dụ:

- Scatter plot: cho hai biến định lượng, thể hiện mức độ tương quan.
- Grouped bar chart: cho một biến định tính và một biến định lượng.
- Heatmap: thể hiện hệ số tương quan giữa nhiều biến cùng lúc.

**+ Đoạn code mẫu để tạo biểu đồ scatter plot hoặc heatmap để phân tích mối quan hệ giữa hai biến?**

- Scatter Plot (Biểu đồ phân tán): Dùng để xem mối quan hệ giữa hai biến liên tục (số học).

```{Python}
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Tạo dữ liệu mẫu
data = {
   'Chiều cao (cm)': [150, 160, 165, 170, 175, 180, 185],
   'Cân nặng (kg)': [45, 50, 55, 60, 70, 75, 80]}
df = pd.DataFrame(data)
# Vẽ scatter plot
sns.scatterplot(data=df, x='Chiều cao (cm)', y='Cân nặng (kg)')
plt.title('Mối quan hệ giữa Chiều cao và Cân nặng')
plt.show()
```

+ Cách đọc:

- Nếu các điểm tạo thành đường chéo đi lên → tương quan dương (chiều cao tăng → cân nặng tăng).
- Nếu đi xuống → tương quan âm.
- Nếu rải lung tung → không có mối quan hệ rõ ràng.

- Heatmap (Bản đồ nhiệt hệ số tương quan): Dùng để xem mức độ tương quan giữa nhiều biến cùng lúc.

```{Python}
# Tính ma trận tương quan
```

```
corr = df.corr(numeric_only=True)
# Vẽ heatmap
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Heatmap thể hiện hệ số tương quan giữa các biến')
plt.show()
```

+ Cách đọc:

- Hệ số tương quan (r) nằm trong [-1, 1].
- r > 0: tương quan dương (cùng chiều).
- r < 0: tương quan âm (ngược chiều).
- r ≈ 0: gần như không có quan hệ.

**+ Làm thế nào để trực quan hóa mối quan hệ giữa một biến số và một biến phân loại bằng biểu đồ boxplot hoặc violin plot trong Python?**

- Biểu đồ Boxplot: Dùng để xem trung vị, khoảng tứ phân vị (IQR) và phát hiện outliers.

```{Python}
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
# Dữ liệu mẫu
data = {
    'Giới tính': ['Nam', 'Nữ', 'Nam', 'Nữ', 'Nam', 'Nữ', 'Nam', 'Nữ'],
    'Điểm thi': [80, 85, 78, 90, 75, 95, 88, 92]}
df = pd.DataFrame(data)
# Vẽ boxplot
sns.boxplot(x='Giới tính', y='Điểm thi', data=df, palette='pastel')
plt.title('Phân bố điểm thi theo giới tính')
plt.show()
```

+ Cách đọc:

- Đường giữa hộp = trung vị (median).
- Hộp = Q1 → Q3 (50% dữ liệu trung tâm).
- Điểm ngoài hộp = outliers.

- So sánh 2 hộp → nhóm nào có trung vị cao hơn → nhóm đó có điểm cao hơn trung bình.

- Biểu đồ Violin Plot: Hiển thị phân bố xác suất (density) và trung vị, nhìn "mềm" và chi tiết hơn boxplot.

```{Python}
sns.violinplot(x='Giới tính', y='Điểm thi', data=df, inner='quartile', palette='Set2')
plt.title('Phân bố điểm thi theo giới tính (Violin Plot)')
plt.show()
```

+ Cách đọc:

- Phần "béo" = nơi dữ liệu tập trung nhiều.
- Đường trắng giữa = trung vị.
- So sánh hình dạng → cho biết phân bố có lệch hay không.

| Biểu đồ | Dùng cho | Ưu điểm |
|---|---|---|
| Boxplot | So sánh trung vị, outliers | Dễ đọc, trực quan |
| Violin plot | Xem chi tiết phân số | Hiển thị dạng xác suất, đẹp hơn |

### 1.3.2. Bài làm mẫu

**Bài toán 1**: Thực hiện các nhiệm vụ trong bài toán 1 để làm quen với các hàm và thư viện hỗ trợ phân tích dữ liệu đơn biến. 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** lấy tại https://www.kaggle.com/datasets/parulpandey/palmer-archipelago-antarctica-penguin-data ,**vào thư mục Code, file có tên là penguin.html.**

1. Import thư viện và nạp dữ liệu
2. Phân tích đơn biến bằng Histogram
3. Phân tích đơn biến bằng bar chart
4. Phân tích đơn biến bằng biểu đồ tròn (Pie-chart)

**Nhiệm vụ 2**: Phân tích dữ liệu đơn biến trên dữ liệu giá nhà lấy từ https://www.kaggle.com/datasets/thomasnibb/amsterdam-house-price-prediction

1. Import thư viện, nạp dữ liệu giá nhà và phân tích đơn biến dựa vào boxplot
2. Phân tích dữ liệu đơn biến dựa vào violin plot
3. Phân tích dữ liệu đơn biến dựa vào bản tóm tắt dữ liệu

--- **Vào thư mục Code, file có tên là amsterdam_house_analysis.html.**

**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. Dữ liệu lấy tại https://www.kaggle.com/datasets/parulpandey/palmer-archipelago-antarctica-penguin-data

1. Import thư viện và chuẩn bị dữ liệu phân tích
2. Phân tích dữ liệu 2 biến dựa vào phương pháp scatterplot
3. Phân tích 2 biến dựa vào bảng crosstab/two-way
4. Phân tích 2 biến sử dụng pivot_table
5. Phân tích 2 biến sử dụng pairplot

--- **Vào thư mục Code, file có tên là penguin_analysis.html.**

**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. Dữ liệu lấy tại https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis

1. Cài đặt pandas_profiling sau này đổi tên thành ydata_profiling (xem thông tin chi tiết tại https://pypi.org/project/pandas-profiling/3.1.0)
Vào Cmd, kiểm tra phiên bản Python có trên 3.7 không (python --version) nếu thỏa thì gõ tiếp: pip install ydata-profiling, nếu lỗi thì xem trên mạng.
2. Sử dụng công cụ
3. Tiến hành EDA trên trang tập tin **profile_output.html**

**Nhiệm vụ 2: Sử dụng dtale trên dữ liệu Marketing Campaign.** Dữ liệu lấy từ https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis

1. Cài đặt dtale (xem thông tin chi tiết tại https://dtale.readthedocs.io/en/latest/)
2. Sử dụng công cụ

### 1.3.3. Bài tập thực hành 1

Tìm hiểu các tính năng và cách sử dụng sản phẩm SweetViz (https://pypi.org/project/sweetviz) áp dụng trên tập dữ liệu Marketing Campaign

### 1.3.4. Bài tập thực hành 2

Tìm hiểu các tính năng và cách sử dụng sản phẩm AutoViz (https://pypi.org/project/autoviz) áp dụng trên tập dữ liệu Marketing Campaign

--- **Vào thư mục Thực hành, file có tên là Autoviz&Sweetviz.html và Tìm hiểu các chức năng của Sweetviz và Autoviz.docx**

# D. TÓM TẮT THỰC HÀNH

**Khám phá dữ liệu (Exploratory Data Analysis - EDA)** là một bước quan trọng trong phân tích dữ liệu và khai thác dữ liệu, nhưng quá trình này không tránh khỏi những khó khăn. Một trong những thách thức lớn nhất là chất lượng dữ liệu không đảm bảo, bao gồm giá trị thiếu, giá trị ngoại lai hoặc dữ liệu không nhất quán, đòi hỏi kỹ năng tiền xử lý phức tạp và tốn thời gian. Bên cạnh đó, việc xử lý khối lượng dữ liệu lớn có thể gây khó khăn trong việc xác định các mẫu hoặc xu hướng có ý nghĩa, đặc biệt khi sử dụng các công cụ không được tối ưu hóa cho dữ liệu lớn. Chương đã trình bày một số kỹ thuật cơ bản khi sử dụng Python và các công cụ phát triển bằng Python giúp thực hiện việc khám phá dữ liệu được hiệu quả hơn.