BÁO CÁO KẾT QUẢ THỰC HÀNH LAB 1

I. CƠ SỞ LÝ THUYẾT

1. Thống kê mô tả & suy luận

Thống kê mô tả: tập trung mô tả một cách súc tích dữ liệu mẫu hiện có, giúp phát hiện xu hướng trung tâm, độ phân tán, hình dạng phân bố và các điểm bất thường để hình thành giả thuyết.

– Thước đo trung tâm: Mean (nhạy cảm outlier), Median (bền vững), Mode (giá trị xuất hiện nhiều nhất).

– Độ phân tán: Variance/Std (độ lệch quanh mean), Range (max–min, rất nhạy cảm outlier), IQR (Q3−Q1, bền vững).

– Hình dạng: Skewness (độ lệch trái/phải), Kurtosis (độ nhọn/độ dẹt).

– Biểu đồ: Histogram (tần suất), Boxplot (median/IQR/outlier), Violin (phân phối + median/IQR).

Thống kê suy luận: sử dụng dữ liệu mẫu để suy rộng cho quần thể với độ bất định có thể đo lường.

– Ước lượng: khoảng tin cậy (CI) cho trung bình/tỷ lệ/khác.

– Kiểm định giả thuyết: t-test, chi-square, ANOVA,… để trả lời liệu khác biệt/quan hệ có ý nghĩa thống kê hay không.

– Sai lầm loại I/II (α/β), p-value, power, cỡ mẫu.

# Ví dụ nhanh (pandas)  
 desc = df.describe(percentiles=[.05,.25,.5,.75,.95]).T  
 iqr = desc['75%'] - desc['25%']  
 skew = df.skew(numeric\_only=True)  
 kurt = df.kurt(numeric\_only=True)

2. Khi nào dùng median thay vì mean?

Dữ liệu lệch phải (ví dụ thu nhập, thời gian chờ): vài giá trị rất lớn kéo mean lên; median phản ánh trung tâm điển hình hơn.

Dữ liệu có outliers do lỗi đo hoặc phân phối heavy-tail: median không bị ảnh hưởng mạnh như mean.

Trong báo cáo kinh doanh, median thường làm đại diện 'khách hàng điển hình' tốt hơn mean.

Tuy nhiên, khi phân phối gần chuẩn và cần tính chất tuyến tính (ví dụ trong mô hình hồi quy), mean vẫn có giá trị.

center = {'mean': x.mean(), 'median': x.median()}  
 robust\_spread = x.quantile(0.75) - x.quantile(0.25) # IQR

3. Phân bố & nhận diện

Histogram: quan sát dạng chung, số bins ảnh hưởng độ mịn (quy tắc Freedman–Diaconis/Scott).

KDE (kernel density estimation): đường mượt biểu diễn mật độ; bandwidth quyết định độ mịn.

QQ-plot: so sánh phân vị mẫu với lý thuyết (chuẩn/log-normal…); điểm lệch đường chéo ⇒ không theo phân bố giả định.

Nhận diện nhanh:

– Chuẩn: đối xứng, đuôi ngắn; lệch phải: đuôi dài bên phải; lệch trái: đuôi dài bên trái; đa đỉnh: nhiều peak.

import seaborn as sns, matplotlib.pyplot as plt  
 fig, ax = plt.subplots(1,2, figsize=(10,4))  
 sns.histplot(x, kde=True, ax=ax[0])  
 import scipy.stats as st  
 st.probplot(x, dist='norm', plot=ax[1])  
 plt.tight\_layout()

4. IQR, outliers & xử lý thiếu

IQR = Q3 − Q1; ngưỡng outlier thường dùng: [Q1 − 1.5×IQR, Q3 + 1.5×IQR]. Có thể dùng 3×IQR cho lọc 'rất ngoại lai'.

Ngoài IQR: Z-score (|z|>3), robust z (MAD), mô hình (Isolation Forest), hoặc theo miền tri thức (ngưỡng y khoa).

Xử lý outlier: kiểm tra nguồn, winsorize, biến đổi (log/Box-Cox), hoặc loại bỏ có biện minh (chú ý bias).

Thiếu dữ liệu:

– Cơ chế thiếu: MCAR/MAR/MNAR; nên phân tích tỷ lệ và mô hình hoá khả năng thiếu.

– Chiến lược: dropna (đơn giản, mất dữ liệu), impute median/mean/mode (nhanh), KNN/Multiple Imputation (tốt hơn), ghi cờ missing.

q1, q3 = x.quantile(.25), x.quantile(.75)  
 iqr = q3 - q1  
 lower, upper = q1 - 1.5\*iqr, q3 + 1.5\*iqr  
 mask\_out = (x<lower)|(x>upper)  
 x\_imputed = x.fillna(x.median())

5. Trực quan hoá

Univariate: Histogram (tần suất), Boxplot (median/IQR/outlier), Violin (phân phối + tóm tắt hộp).

Bivariate: Scatter (quan hệ số–số, thêm hue/size/style), Heatmap (corr tuyến tính), Box/Violin theo nhóm, Crosstab/Pivot (danh mục↔số).

Thực hành tốt: tiêu đề rõ, nhãn trục + đơn vị, chú thích; hạn chế màu rối/3D; dùng tight\_layout() và lưu hình độ phân giải đủ.

sns.boxplot(data=df, x='Category', y='Value')  
 sns.scatterplot(data=df, x='X', y='Y', hue='Group')  
 sns.heatmap(df[num\_cols].corr(), cmap='viridis', annot=False)  
 plt.tight\_layout(); plt.savefig('plot.png', dpi=150)

1.6. Ghi chú đặc thù từng bộ dữ liệu

Wine Quality (UCI): CSV phân cách ';' ⇒ dùng sep=';'. Nên phân tích theo 'quality' (0–10), báo cáo đặc trưng nổi bật theo từng mức.

Pima Diabetes: coi 0 là thiếu cho Glucose/BloodPressure/SkinThickness/Insulin/BMI; impute median; so sánh theo Outcome (0/1).

Online Retail: loại hoá đơn huỷ ('C…'), lọc Quantity/UnitPrice > 0, tạo Revenue = Quantity×UnitPrice; resample theo ngày/tháng.

# Wine  
 wine = pd.read\_csv('winequality-red.csv', sep=';')  
 # Pima  
 for c in ['Glucose','BloodPressure','SkinThickness','Insulin','BMI']:  
 pima.loc[pima[c]==0, c] = None  
 pima.fillna(pima.median(numeric\_only=True), inplace=True)  
 # Retail  
 ret = ret[~ret['InvoiceNo'].astype(str).str.startswith('C')]  
 ret = ret[(ret['Quantity']>0)&(ret['UnitPrice']>0)]  
 ret['Revenue'] = ret['Quantity']\*ret['UnitPrice']

7. Công cụ EDA tự động

ydata\_profiling: báo cáo tổng quan (phân phối, tương quan, cảnh báo), hữu ích rà soát chất lượng dữ liệu; tránh tệp quá lớn.

from ydata\_profiling import ProfileReport  
 profile = ProfileReport(df, title='Data Profiling', minimal=True)  
 profile.to\_file('Reports/profile\_output.html')

SweetViz: nhanh cho overall/target; nếu có cột Response (0/1) hãy tạo báo cáo theo target để so sánh phân nhóm.

import sweetviz as sv  
 rep = sv.analyze(df)  
 rep.show\_html('Reports/sweetviz\_overall.html')  
 rep\_t = sv.analyze([df,'Marketing'], target\_feat='Response')  
 rep\_t.show\_html('Reports/sweetviz\_target\_Response.html')

AutoViz: sinh biểu đồ tự động; supervised khi có depVar (cột mục tiêu), nếu không có thì unsupervised để khám phá phân phối & tương quan.

from autoviz.AutoViz\_Class import AutoViz\_Class  
 AV = AutoViz\_Class()  
 dft = AV.AutoViz(filename='', sep=',', depVar='Response', dfte=df) # nếu có Response

II. PHÂN CÔNG THÀNH VIÊN

| Thành viên | Nhiệm vụ chính | Ghi chú/Kết quả |
| --- | --- | --- |
|  | - Tiền xử lý dữ liệu (làm sạch, chuẩn hoá)  - Thống kê mô tả & phân tích đơn biến (hist/box/violin)  - Viết phần A–B (mục tiêu, phương pháp) | - Tạo bảng describe(), percentiles, IQR / outliers  - Lưu các biểu đồ Histogram/Box/Violin |
|  | - Phân tích hai biến & insight (scatter/heatmap/crosstab/pairplot)  - EDA tự động (SweetViz/AutoViz/ydata\_profiling)  - Viết phần C–D (nhận xét, kết luận) | - Heatmap tương quan, crosstab/pivot  - Xuất báo cáo HTML (profiling/SweetViz), thư mục AutoViz |

III. PHÂN TÍCH MÃ ĐÃ THỰC HIỆN THEO TỪNG “BÀI TẬP THỰC HÀNH”

1. THỐNG KÊ MÔ TẢ

PHẦN A – Pima Indians Diabetes (diabetes.csv)

• Mục tiêu: thống kê mô tả, xử lý giá trị 0 như thiếu cho các biến sinh lý, so sánh theo Outcome (0/1) và phát hiện ngoại lai theo quy tắc IQR.

1) Tóm tắt mã nguồn

– Đọc dữ liệu và xác nhận cột kỳ vọng; ép kiểu số cho toàn bộ cột.

– Tạo hàm: iqr\_bounds(), descriptive\_table(), group\_descriptive(), outlier\_report().

– Phân tích RAW (chưa xử lý 0→NA) và CLEAN (sau khi coi 0 là NA và điền median).

– Bảng mô tả: count/mean/std/min/Q1/median/Q3/max + range, IQR, skew, kurtosis, mode; percentiles P5/P25/P50/P75/P95.

– Báo cáo ngoại lai theo IQR; thống kê theo nhóm Outcome (mean/median/std/Q1/Q3).

# Cấu trúc chính  
 df = pd.read\_csv(CSV\_PATH)  
 for c in ALL\_COLS: df[c] = pd.to\_numeric(df[c], errors="coerce")  
 # RAW  
 raw\_desc = descriptive\_table(df, NUM\_COLS)  
 raw\_outliers = outlier\_report(df, NUM\_COLS)  
 # CLEAN  
 clean = df.copy()  
 for c in ZERO\_AS\_NA\_COLS: clean.loc[clean[c]==0, c] = np.nan  
 for c in NUM\_COLS: clean[c] = clean[c].fillna(clean[c].median())  
 by\_outcome = group\_descriptive(clean, "Outcome", NUM\_COLS)

2) Kết quả chính (trích xuất từ log bạn cung cấp)

• Phân bố lớp (RAW): Outcome=0: 500; Outcome=1: 268 ⇒ dữ liệu mất cân bằng ở mức vừa (65% vs 35%).

• Thiếu dữ liệu (RAW): 0 ở tất cả cột (nhưng có nhiều giá trị 0 không thực tế).

• Thống kê RAW (chọn vài chỉ tiêu):

– Glucose mean=120.9, std=32.0; BloodPressure mean=69.1; BMI mean=32.0.

– Nhiều giá trị 0 ở SkinThickness, Insulin, BMI (thấy qua percentiles P25=0 cho SkinThickness/Insulin).

• Ngoại lai RAW (IQR): BloodPressure (45), Insulin (34), DPF (29)…

• Sau khi CLEAN (0→NA→impute median):

– Glucose mean tăng nhẹ 121.7; BloodPressure mean 72.4; SkinThickness mean ~29.1; Insulin mean ~140.7; BMI mean ~32.5.

– Ngoại lai CLEAN (IQR): Insulin tăng mạnh (346) do phân phối lệch/phân tách bởi median impute; SkinThickness (87).

• So sánh theo Outcome (CLEAN):

– Positive (1) có Glucose trung bình ~142 (cao hơn nhóm 0 ~111).

– BMI, Age, DPF cũng cao hơn ở Outcome=1; Pregnancies trung bình 4.87 so với 3.30 ở Outcome=0.

3) Nhận xét & ý nghĩa

• Việc coi 0 là NA cho các biến sinh lý là cần thiết: thống kê trung tâm và phân tán trở nên hợp lý hơn sau khi impute (ví dụ SkinThickness, Insulin).

• Glucose là tín hiệu mạnh phân tách Outcome (chênh lệch ~30 mg/dL). Age, BMI, DPF, Pregnancies cũng có khác biệt đáng kể giữa hai nhóm.

• Ngoại lai Insulin rất nhiều (do phân phối lệch phải và imputing bằng trung vị giữ phân phối hẹp quanh median, khiến các giá trị lớn bị gắn cờ). Với mô hình học máy, nên cân nhắc biến đổi log hoặc Winsorize cho Insulin.

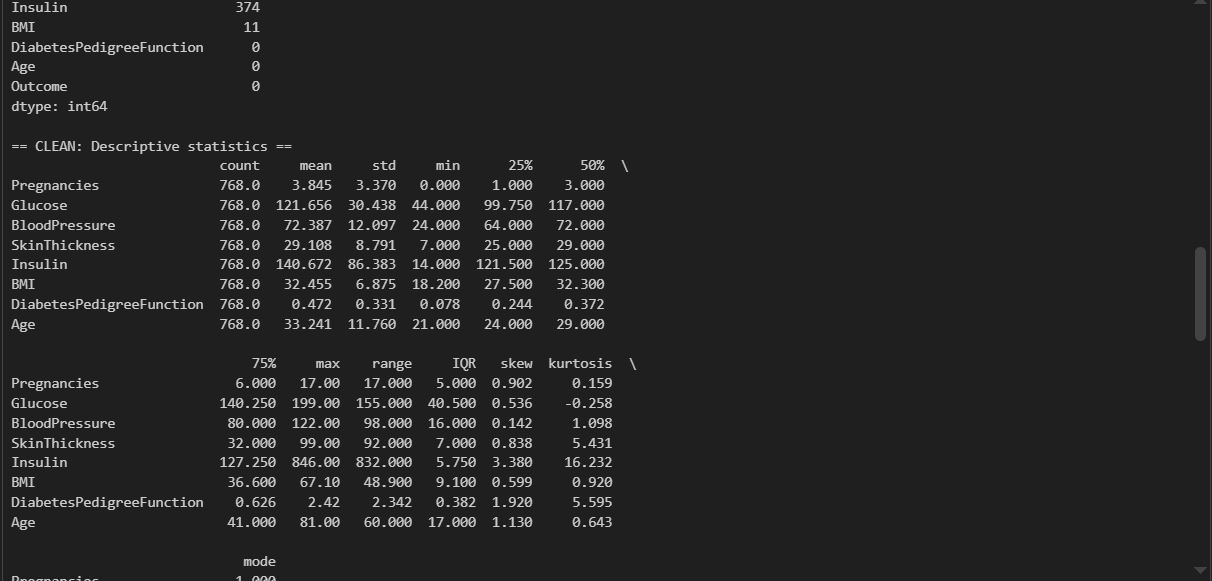
• Dữ liệu nhẹ mất cân bằng ⇒ cân nhắc dùng metrics cân bằng (AUC/PR-AUC, balanced accuracy) hoặc tái cân bằng (class\_weight).

4) Khuyến nghị ngắn

• Chuẩn hoá pipeline: thay 0→NA cho Glucose/BP/Skin/Insulin/BMI; impute median hoặc KNNImputer; thêm cờ ‘was\_zero’ nếu cần.

• Biến đổi log cho Insulin; chuẩn hoá/chuẩn vị trí-tỷ lệ cho các biến lệch.

• Tiền chọn đặc trưng: Glucose, Age, BMI, DPF, Pregnancies ưu tiên trước; thử mô hình tuyến tính/phi tuyến và đánh giá trên thang đo cân bằng.



PHẦN B – Wine Quality (UCI, red wine)

1) Tóm tắt mã nguồn

– Đọc dữ liệu; ép kiểu số; điền thiếu bằng median (nếu có).

– Bảng mô tả: count/mean/std/min/Q1/median/Q3/max + range, IQR, skew, kurtosis, mode; percentiles P5/P25/P50/P75/P95.

– Phát hiện ngoại lai theo IQR cho từng biến.

– Gom nhóm theo quality và tính mean/median/std/Q1/Q3 (làm phẳng MultiIndex cột).

df = pd.read\_csv(CSV\_PATH) # khuyến nghị sep=';' cho bộ UCI gốc  
 for c in NUM\_COLS+[TARGET\_COL]: df[c] = pd.to\_numeric(df[c], errors='coerce')  
 desc = df[NUM\_COLS].describe().T; desc['IQR'] = desc['75%']-desc['25%']  
 # Outliers theo IQR  
 # Group theo quality: df.groupby('quality')[NUM\_COLS].agg(['mean','median','std', q1, q3])

2) Kết quả chính (trích xuất từ log bạn cung cấp)

• Không có giá trị thiếu ở các cột số và ‘quality’.

• Thống kê chung (một số điểm nổi bật):

– residual sugar mean≈2.54, lệch phải mạnh (skew≈4.54, kurtosis≈28.6) ⇒ nhiều giá trị cao nhưng hiếm.

– chlorides skew≈5.68 (đuôi phải dài) ⇒ nhạy với một số mẫu mặn bất thường.

– alcohol mean≈10.42% (std≈1.07).

• Ngoại lai theo IQR (đếm): residual sugar (155), chlorides (112), sulphates (59), total sulfur dioxide (55)…

• Theo chất lượng (quality 3→8):

– volatile acidity giảm dần khi chất lượng tăng (ví dụ mean ~0.885 ở q=3 xuống ~0.404 ở q=7).

– sulphates và alcohol tăng cùng chất lượng (alcohol mean ~9.90 ở q=5, ~11.47 ở q=7, ~12.09 ở q=8).

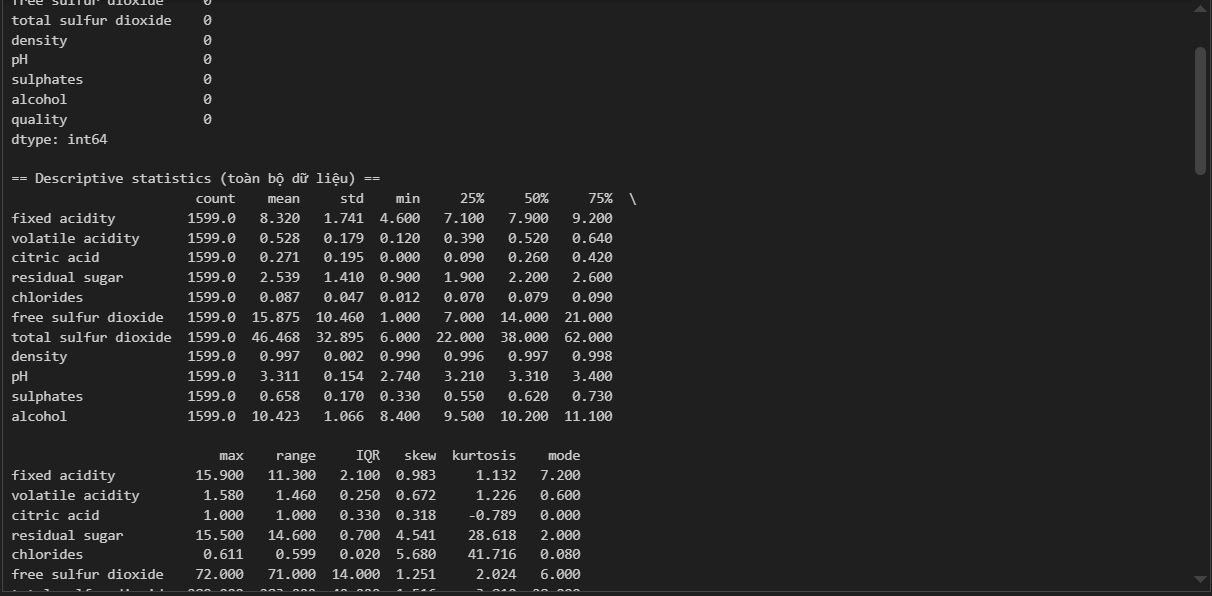
– Các thuộc tính này phù hợp hiểu biết cảm quan: rượu có VA thấp, sulfat và độ cồn cao hơn thường được chấm điểm tốt hơn.

3) Nhận xét & ý nghĩa

• Phân phối nhiều biến lệch phải rõ rệt; IQR phù hợp để phát hiện ngoại lai hơn z-score thuần tuý.

• Dấu hiệu chất lượng: giảm acid bay hơi (volatile acidity), tăng sulphates và alcohol là các tín hiệu mạnh.

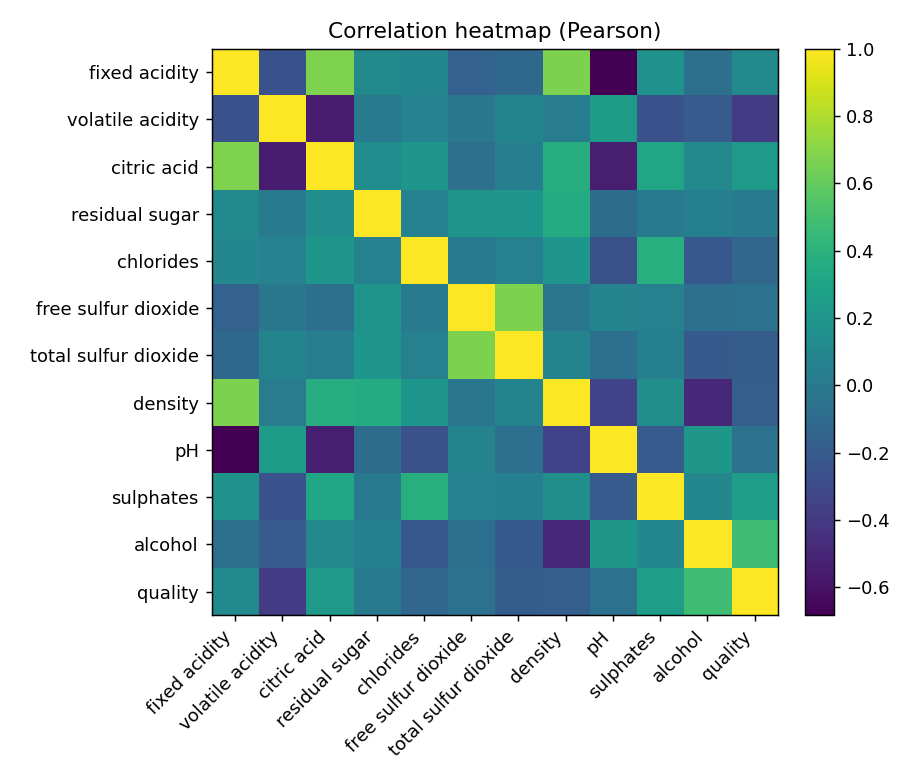
• Với mô hình phân loại chất lượng, nên chuẩn hoá, xử lý skew (log đối với residual sugar/chlorides/sulphates), và cân nhắc cắt ngưỡng quality (ví dụ good≥6).

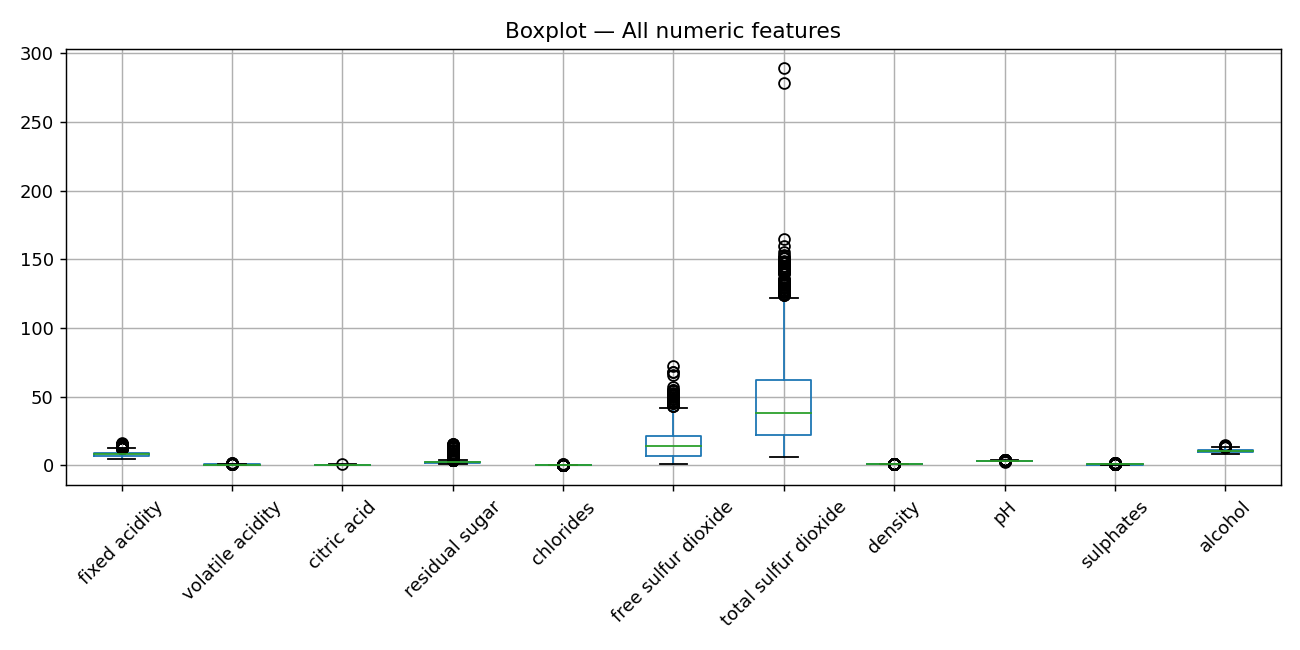


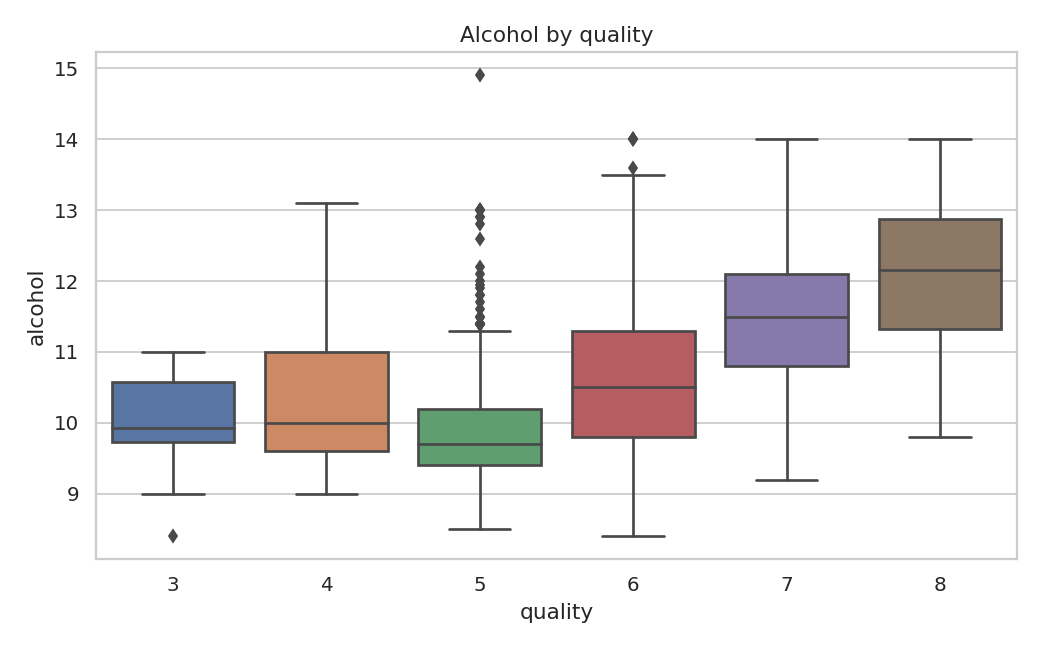
2. XỬ LÝ VÀ TRỰC QUAN HÓA DỮ LIỆU

PHẦN A

• Phạm vi mã: phần trực quan hoá (6.0 → 6.5) cho bộ dữ liệu Wine Quality (UCI – red). Mã giả định các biến đã tồn tại: df, NUM\_COLS, TARGET\_COL='quality', OUTPUT\_DIR.

• Mục tiêu: sinh các hình mô tả phân phối, hộp, tương quan và quan hệ theo chất lượng.





1) Luồng xử lý & đầu ra

– 6.0 Quality distribution: biểu đồ cột tần suất nhãn quality ⇒ file quality\_distribution.png.

– 6.1 Histogram từng biến số ⇒ hist\_<tên\_biến>.png (bins=40).

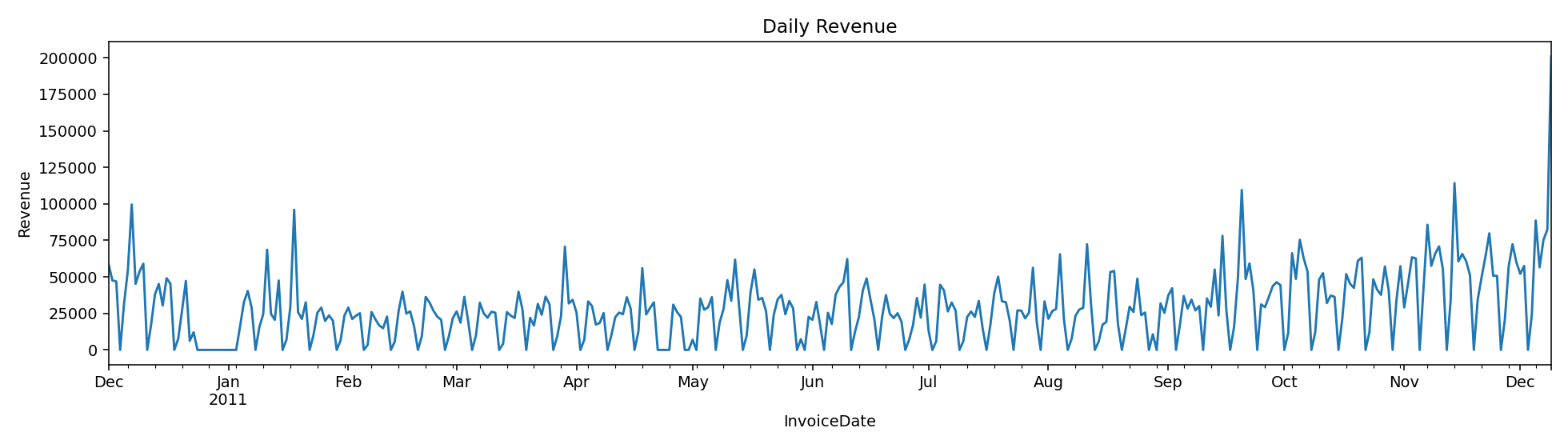
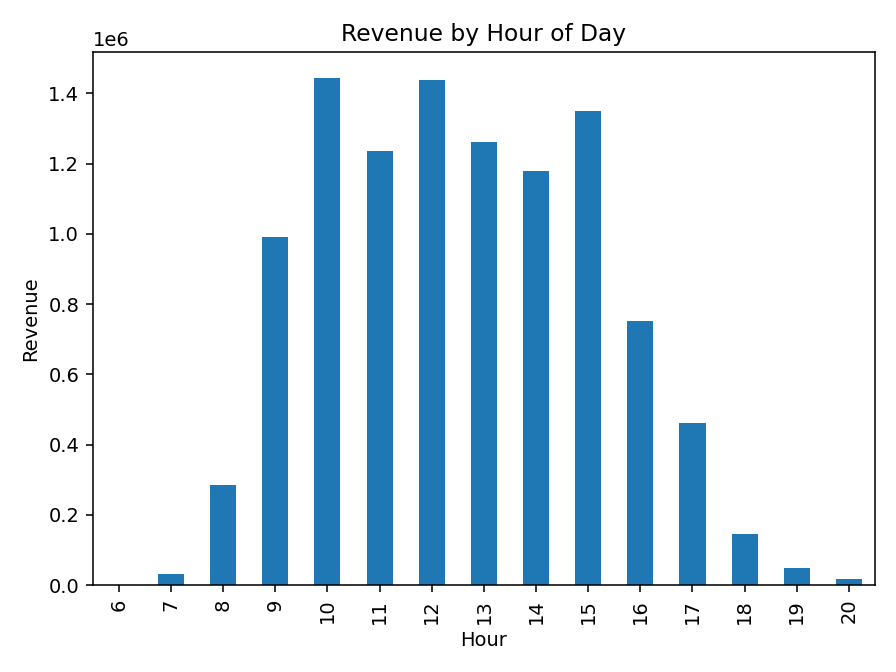
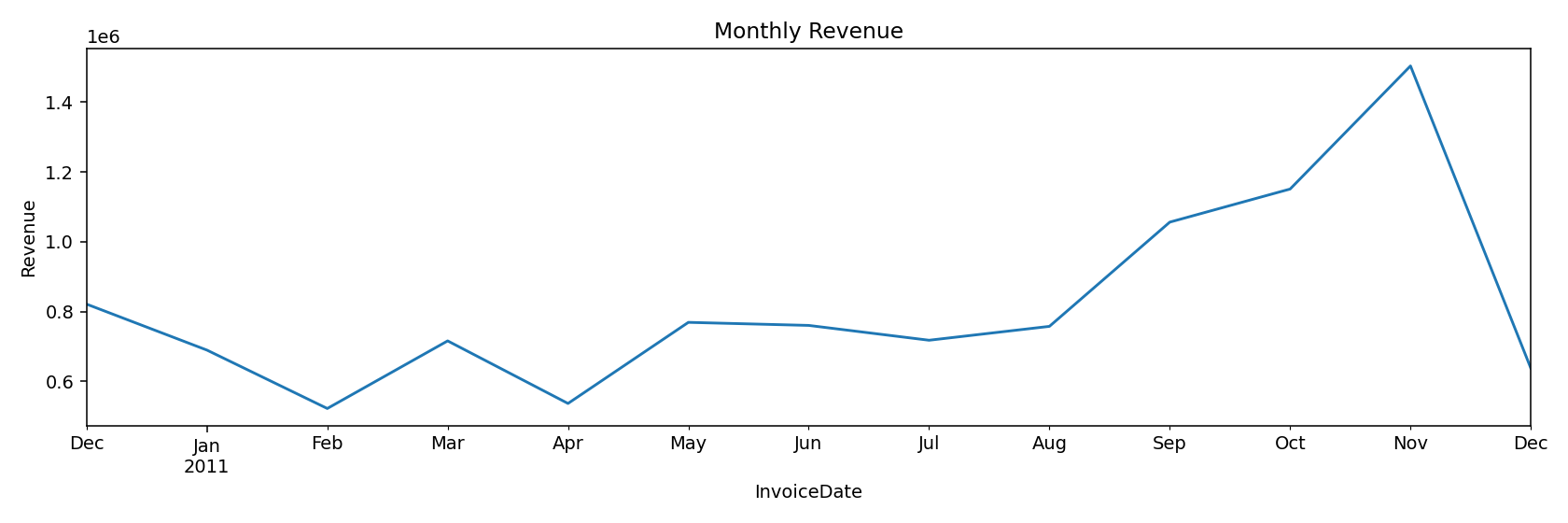
– 6.2 Boxplot tất cả biến số ⇒ boxplot\_all.png.

– 6.3 Heatmap tương quan Pearson (NUM\_COLS + quality) ⇒ corr\_heatmap.png.

– 6.4 Scatter alcohol vs. quality ⇒ scatter\_alcohol\_quality.png (nhìn nhanh xu hướng).

– 6.5 (tuỳ chọn, nếu cài seaborn): boxplot alcohol theo quality và violinplot volatile acidity theo quality ⇒ box\_alcohol\_by\_quality.png, violin\_va\_by\_quality.png.

PHẦN B

• Phạm vi mã: EDA cho bộ Online Retail sau khi đã tiền xử lý. Mã giả định đã có df (đã clean) với các cột chuẩn và cột Revenue; thư mục OUTPUT\_DIR và TABLE\_DIR tồn tại.

• Tiền điều kiện khuyến nghị: loại invoice huỷ (InvoiceNo bắt đầu 'C'), lọc Quantity>0 & UnitPrice>0, tạo Revenue=Quantity\*UnitPrice, chuẩn hoá InvoiceDate về datetime.

1) Luồng xử lý & đầu ra

– Top sản phẩm theo Revenue: groupby(Description).sum() ⇒ top10\_products\_by\_revenue.csv/.png.

– Top quốc gia theo Revenue: groupby(Country).sum() ⇒ top10\_countries\_by\_revenue.csv/.png.

– Chuỗi thời gian: đặt index là InvoiceDate; doanh thu theo ngày (resample('D')), theo tháng (resample('M')), theo giờ (groupby(Hour)). Xuất daily\_revenue.csv/.png; monthly\_revenue.csv/.png; revenue\_by\_hour.csv/.png.

– RFM cơ bản: Recency (ngày từ snapshot\_date), Frequency (số invoice duy nhất), Monetary (tổng Revenue). Xuất rfm.csv và histogram phân phối Recency/Frequency/Monetary ⇒ rfm\_\*\_hist.png.

– Top 20 khách hàng theo Monetary ⇒ top20\_customers\_monetary.png.

– Phân tích giỏ hàng (Co-occurrence): đếm cặp Description cùng xuất hiện trên 1 hoá đơn ⇒ top20\_pairs.csv và top20\_product\_pairs.png.

PHẦN C

• Phạm vi mã: trực quan hoá sau khi đã chuẩn hoá CLEAN (0→NaN→impute median) cho bộ Pima Indians Diabetes. Mã giả định các biến đã tồn tại: clean, NUM\_COLS, OUTPUT\_DIR, TABLE\_DIR.

• Mục tiêu: so sánh phân phối đặc trưng sau làm sạch; quan sát tương quan và cân bằng Outcome.

1) Luồng xử lý & đầu ra

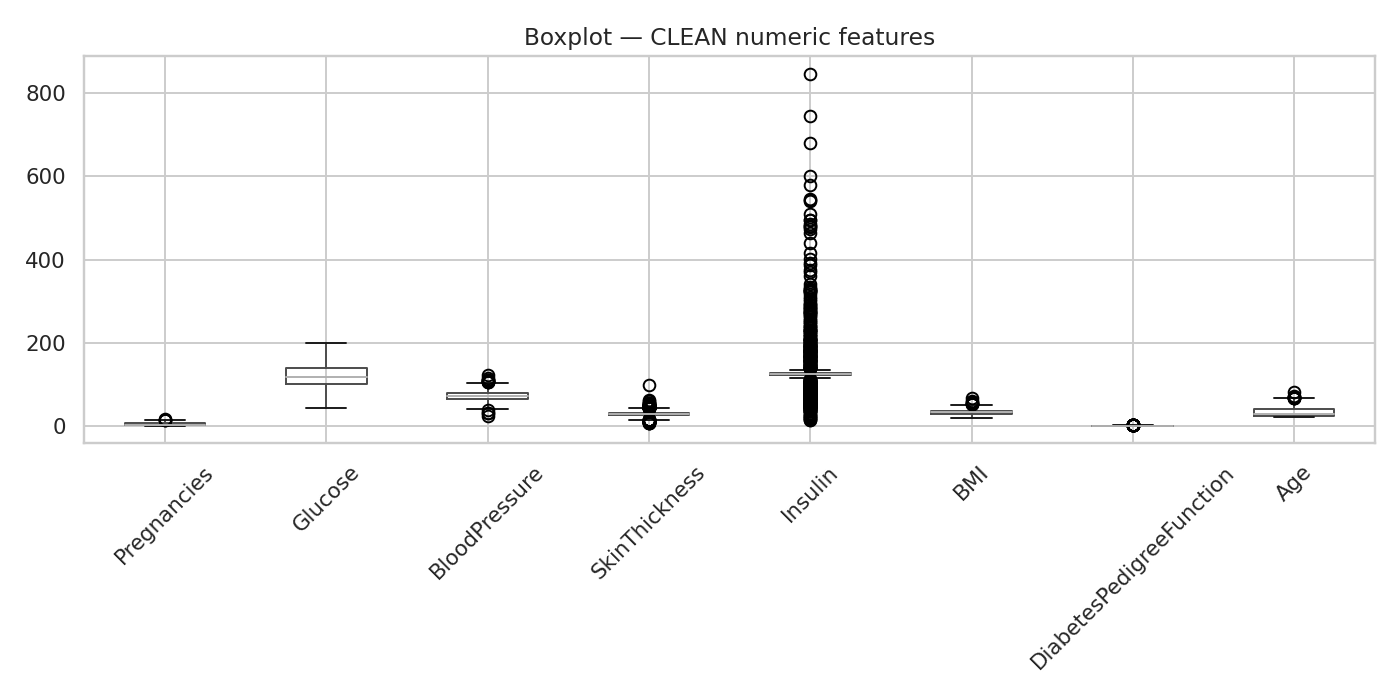
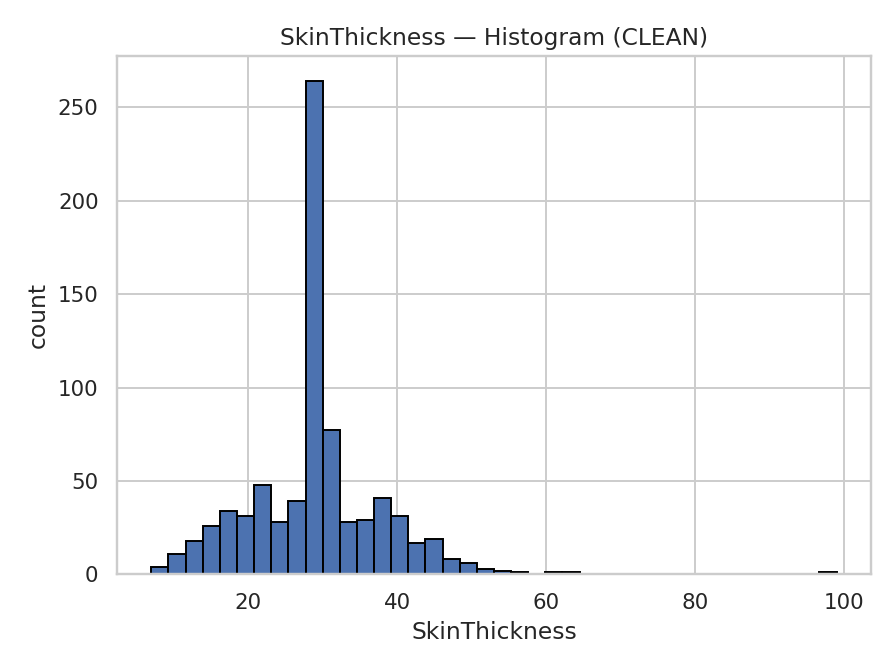
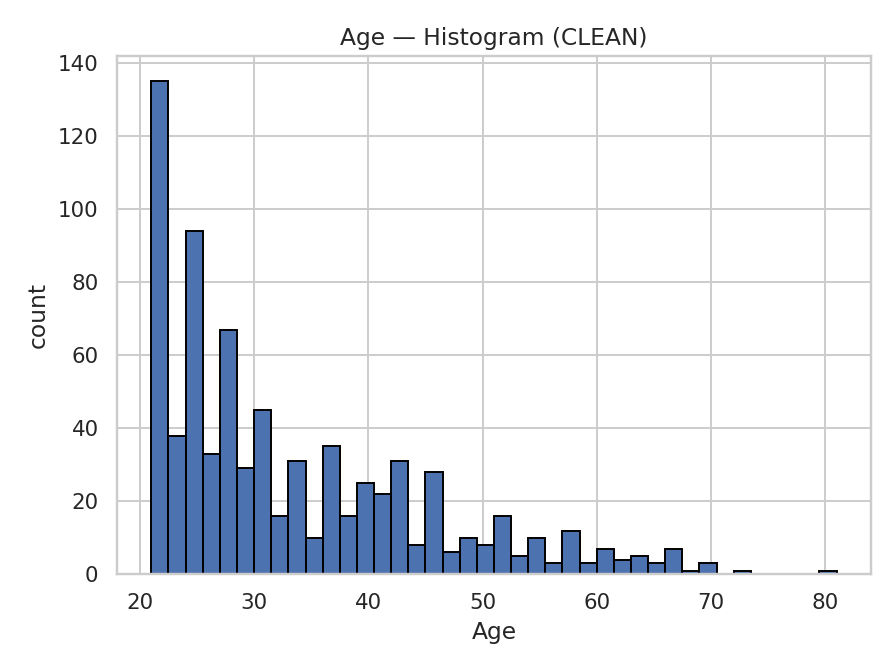
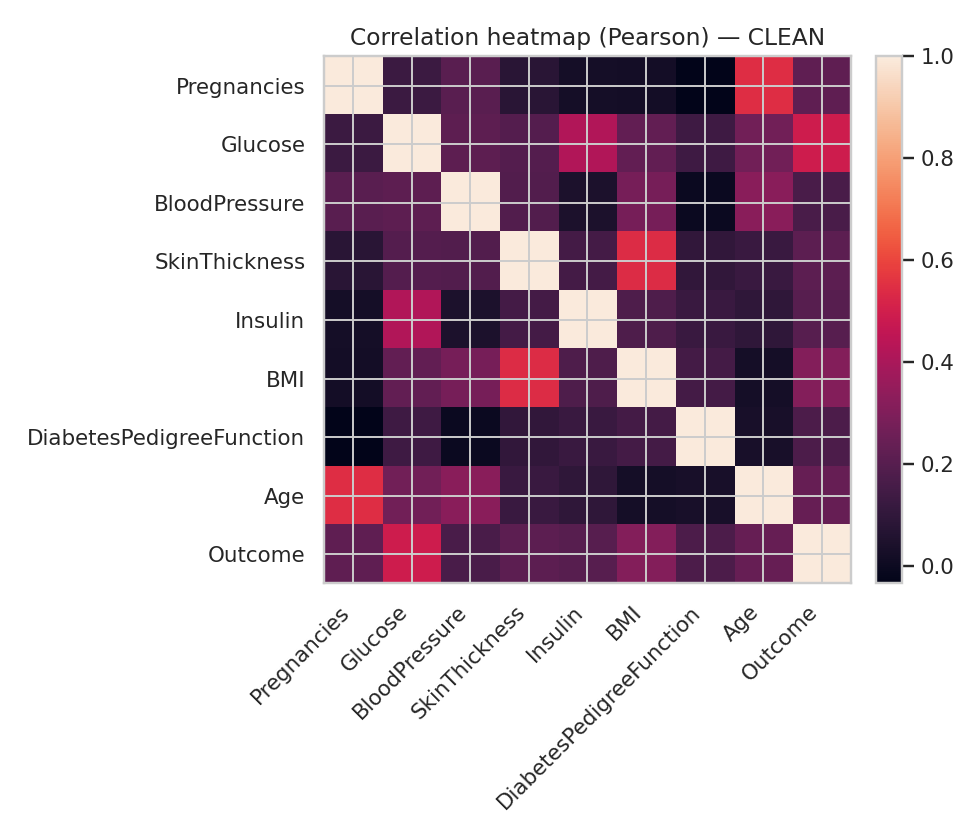
– 5.1 Histogram cho từng biến số (CLEAN) ⇒ hist\_<col>.png.

– 5.2 Boxplot all numeric (CLEAN) ⇒ boxplot\_clean.png.

– 5.3 Pearson correlation heatmap trên NUM\_COLS+Outcome ⇒ corr\_heatmap\_clean.png.

– 5.4 Outcome distribution (bar) ⇒ outcome\_balance.png.

– 5.5 (tuỳ chọn, nếu có seaborn): KDE theo Outcome cho từng biến ⇒ kde\_<col>\_by\_outcome.png.



3) Rủi ro/Pitfall & cách khắc phục

• Cần đảm bảo ‘clean’ đã được tạo từ pipeline chuẩn; nếu không sẽ NameError.

• Một số biến rất lệch (Insulin) ⇒ histogram dễ ‘dính vào trục trái’; cân nhắc log-scale.

• Impute trung vị làm thu hẹp phương sai; khi dùng mô hình, cân nhắc KNNImputer để giữ cấu trúc địa phương.

3.3. Phân tích đơn biến và hai biến

PHẦN 1 – Robust Loader + SweetViz

Luồng xử lý chính

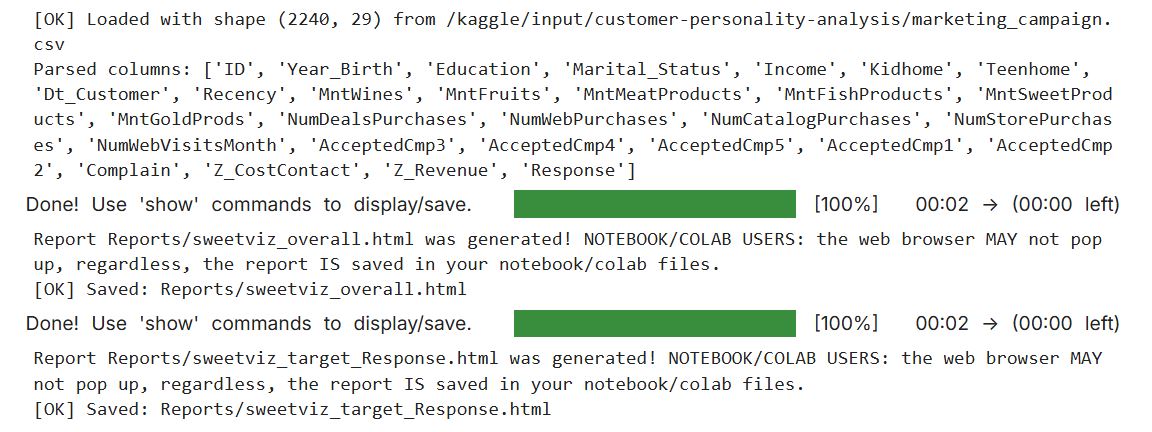
• Dùng danh sách đường dẫn dự phòng (CSV\_PATHS) để tìm file khả dụng.

• Đọc CSV bằng cơ chế tự suy ra delimiter (sep=None, engine='python'). Nếu chỉ ra 1 cột → thử lần lượt các dấu phân cách phổ biến [';', '\t', ',', '|'].

• Chuẩn hoá tên cột: strip() khoảng trắng; ép kiểu 'Response' sang int (coerce NA→0).

• Chuyển 'Dt\_Customer' sang datetime với dayfirst=True (xử lý định dạng dd/mm/yyyy).

• Sinh báo cáo SweetViz: (i) analyze overall; (ii) analyze target với target\_feat='Response'.



PHẦN 2 – AutoViz Pipeline (autoviz\_marketing\_campaign\_fixed\_v2.py)

Luồng xử lý chính

• Nạp CSV bằng cơ chế autodetect delimiter và fallback giống Phần 1; chuẩn hoá header và ép kiểu số cho các trường số.

• Chuẩn datetime ('Dt\_Customer') với dayfirst=True.

• Tạo đặc trưng: Age = 2025 - Year\_Birth; Children = Kidhome + Teenhome; HasChild (nhị phân).

• Tổng chi tiêu MntTotal = sum(Mnt\*); tổng số lần mua PurchasesTotal = sum(Num\* − Visits).

• Tính ShareWeb/Catalog/Store theo công thức an toàn: chỉ chia khi tổng kênh > 0 (tránh chia 0).

• Bỏ cột định danh ('ID') để tránh rò rỉ.

• Gọi AutoViz: nếu có 'Response' ⇒ supervised; nếu không ⇒ unsupervised. Lưu biểu đồ vào thư mục save\_plot\_dir.

• Xử lý kết quả trả về đa phiên bản (tuple hoặc object) để tránh lỗi 'too many values to unpack'.

