⁉ Advanced EDA Exercise (3-Hour Challenge)

Dataset: UCI Wine Quality Dataset – Red Wine
Tools: `NumPy`, `Pandas`, `Matplotlib`, `Seaborn`

📝 Instructions

Download the file `winequality-red.csv` and complete the following tasks in a Jupyter Notebook.

🔧 Part 1: Advanced Data Wrangling

1. Load the data and display:
   - Top and bottom 3 rows.
   - DataFrame memory usage in MB (optimize by changing column dtypes if needed).
2. Detect and remove:
   - Exact duplicates.
   - Rows with more than one standard deviation below the mean in **more than 3 columns**.
3. Create new features:
   - `acid_sugar_ratio = fixed acidity / residual sugar`
   - `sulfate_acidity = sulphates / volatile acidity`
   - Cap values in `chlorides` and `volatile acidity` at 99th percentile using NumPy.

📊 Part 2: Multi-level Grouping and Aggregation (30 mins)

4. Bin the `quality` column into:
   - `Low (<=4)`, `Medium (5-6)`, and `High (>=7)` using `pd.cut`.
5. Using `groupby` and `agg`, compute:
   - Mean, std, and IQR for `alcohol`, `sulphates`, and `citric acid` across quality bins.
   - Mean difference of `alcohol` and `volatile acidity` between `High` and `Low` wines.
6. Create a multi-index pivot table:
   - Index: `quality_group`, Columns: quantile bins of `alcohol`, Values: `density` mean

📈 Part 3: Advanced Visual Exploration (45 mins)

7. Plot customized visuals:
   - Overlay KDE plots for `alcohol` by `quality_group`.
   - Plot violin plots for `residual sugar` grouped by `quality_group` and hue on binned `pH`.
8. Generate a heatmap showing:
   - Pearson and Spearman correlations side-by-side (subplot), sorted by correlation with `quality`.
9. Create a diverging bar chart showing:
   - Z-scores of mean `alcohol`, `citric acid`, and `density` by quality levels.

🔬 Part 4: Outlier Strategies and Detection (30 mins)

10. Implement 3 different outlier detection methods on `total sulfur dioxide`:
    - IQR method
    - Z-score method (NumPy)
    - MAD (Median Absolute Deviation)
11. Compare how many outliers are detected per method and create a Venn diagram of overlapping ones (use set logic).
12. Remove the most extreme outliers (intersection of all methods) and compare group-wise means **before and after**.

⚙️ Part 5: Functionalized Insights + Summary (30 mins)

13. Create a reusable function `feature_stats(df, col)` that returns:
    - Skewness, kurtosis, 95th percentile, and missing %.
14. Create a decorator `@timeit` to time the execution of heavy visualizations (matplotlib, seaborn).
15. Build a final dashboard with subplots using `matplotlib`:
    - Histograms of top 3 correlated features with `quality`.
    - A single `pairplot` of only `High` and `Low` quality wines with `citric acid`, `alcohol`, and `pH`.