# Practical Assignment 1 - Pandas and NumPy Fundamentals

**Registration No.**: `GF202343160`
**Name**: Ajay Sonkar
**GitHub Repository**: CSU1658-Statistical-Foundation-of-Data-Sciences
**Practical Direct Link**: https://github.com/skarFubatsu/CSU1658-Statistical-Foundation-of-Data-Sciences/tree/main/practicals/P001

## Solution Methodology

- `Synthetic Data Creation`: Created synthetic datasets with specified characteristics using NumPy and Pandas.
- `Problem 1`: Remove rows with NaN in required columns only, preserving valid data. Compute arithmetic mean and median of income on the cleaned subset (they ignore NaNs). For the age-weighted mean, treat age as reliability/importance weights and apply sum(age * income)/sum(age). Compare mean vs median to note skew/outlier influence; weighted mean highlights contributions of older (heavier weight) individuals. All computations are transparent and reproducible with simple pandas Series methods.
- `Problem 2`: Compute income mean and standard deviation with skipna to ignore missing values. Derive each z-score as (income − mean)/std, propagating NaN where income is NaN (no row drop). Flag outliers where absolute z-score exceeds 3 (|z| > 3). Summarize count of such extreme values to assess tail heaviness introduced by synthetic outliers. Retain original data structure while appending a z-score column for downstream analysis.
- `Problem 3`: Define explicit left-inclusive, right-exclusive age bins matching the specification using `pd.cut`. Copy prior z-score DataFrame to preserve calculations, assign each record to a bin, then `groupby` age_bin computing: observation count (non-NaN ages), mean income (ignoring NaNs), and median z-score (robust central tendency). Reset index and sort bins to maintain natural age order. This summarizes distributional shifts across life stages.
- `Problem 4`: Construct higher-dimensional random arrays to illustrate structural properties. Inspect shape/size/ndim, then reshape to 2D, transpose, and flatten to show memory/view transformations. Demonstrate negative indexing for tail access and intentionally trigger an index error via invalid slice.

Showcase broadcasting with scalar and shape-compatible addition, then compute dot product for matrix multiplication. Generate a square matrix to obtain determinant and (if non-singular) its inverse, linking numeric linear algebra concepts.

---

**Note**: The detailed code implementation and results can be found in the `notebook.ipynb` file within this directory.