Visual Data Analysis: Anscombe's Quartet

JZ – 11ZZ 894589
Port Credit Secondary School
GitHub: @JZ894589

October 7, 2025

Overview: This project explores Anscombe's Quartet four datasets with nearly identical summary statistics but dramatically different visual distributions. Using Python tools such as NumPy, pandas, matplotlib, and seaborn, we compute descriptive statistics and generate a series of visualizations to demonstrate the importance of exploratory data analysis (EDA).

Contents

1 Executive Summary

This report presents an exploratory data analysis (EDA) of **Anscombe's Quartet**, a classic example demonstrating that datasets with identical summary statistics can have vastly different patterns when visualized.

Using Python libraries including `numpy`, `pandas`, `matplotlib`, and `seaborn`, we calculated the mean, variance, correlation, and linear regression coefficients for four datasets. We then generated scatter plots with regression lines, residual plots, and box/violin plots.

This study shows that summary statistics alone are insufficient and that visualizations are essential for uncovering hidden data structures and outliers.

2 Introduction

Anscombe's Quartet, introduced by statistician Francis Anscombe in 1973, consists of four datasets that share nearly identical descriptive statistics but display distinct relationships when plotted.

The purpose of this analysis is to highlight why visualization should always accompany statistical summaries in data analysis. Exploratory Data Analysis (EDA) helps detect anomalies, identify patterns, and prevent misinterpretation of numerical results.

3 Data

Each dataset consists of paired (x, y) values. Datasets share identical x values, while Dataset has a different x pattern.

```
x123 = [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5]
y1   = [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68]
y2   = [9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74]
y3   = [7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73]
x4   = [8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8]
y4   = [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89]
```

The data were defined directly in Python using NumPy arrays for computational efficiency.

4 Methods

4.1 Statistical Methods

For each dataset (I–IV), we computed:

- Mean, variance, and standard deviation of x and y
- Covariance and correlation between x and y
- Linear regression coefficients (a, b) and R^2

4.2 Visualization Techniques

Visualizations were created using `matplotlib` and `seaborn`, including:

- Scatter plots with regression lines
- Residual plots
- Overlaid comparison plots
- Box and violin plots for x and y

5 Summary Statistics

5.1 Equations

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{(Mean)}$$

$$s_x^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \qquad \text{(Variance)}$$

$$s_x = \sqrt{s_x^2} \qquad \text{(Standard Deviation)}$$

$$cov(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1} \qquad \text{(Covariance)}$$

$$r = \frac{cov(x,y)}{s_x s_y} \qquad \text{(Correlation Coefficient)}$$

$$\hat{y} = a + bx \qquad \text{(Regression Line)}$$

$$b = \frac{cov(x,y)}{s_x^2}, \quad a = \bar{y} - b\bar{x} \qquad \text{(Slope and Intercept)}$$

$$R^2 = r^2 \qquad \text{(Coefficient of Determination)}$$

5.2 Example Results

Statistic	Value
Mean of x	9.0000
Mean of y	7.5009
Variance of x	11.0000
Variance of y	4.1273
Std. Dev. of x	3.3166
Std. Dev. of y	2.0316
Covariance (x, y)	5.5010
Correlation (r)	0.8164
Regression slope (b)	0.5001
Regression intercept (a)	3.0000
R-squared	0.6665

6 Visualizations

6.1 Scatter Plots with Regression Lines

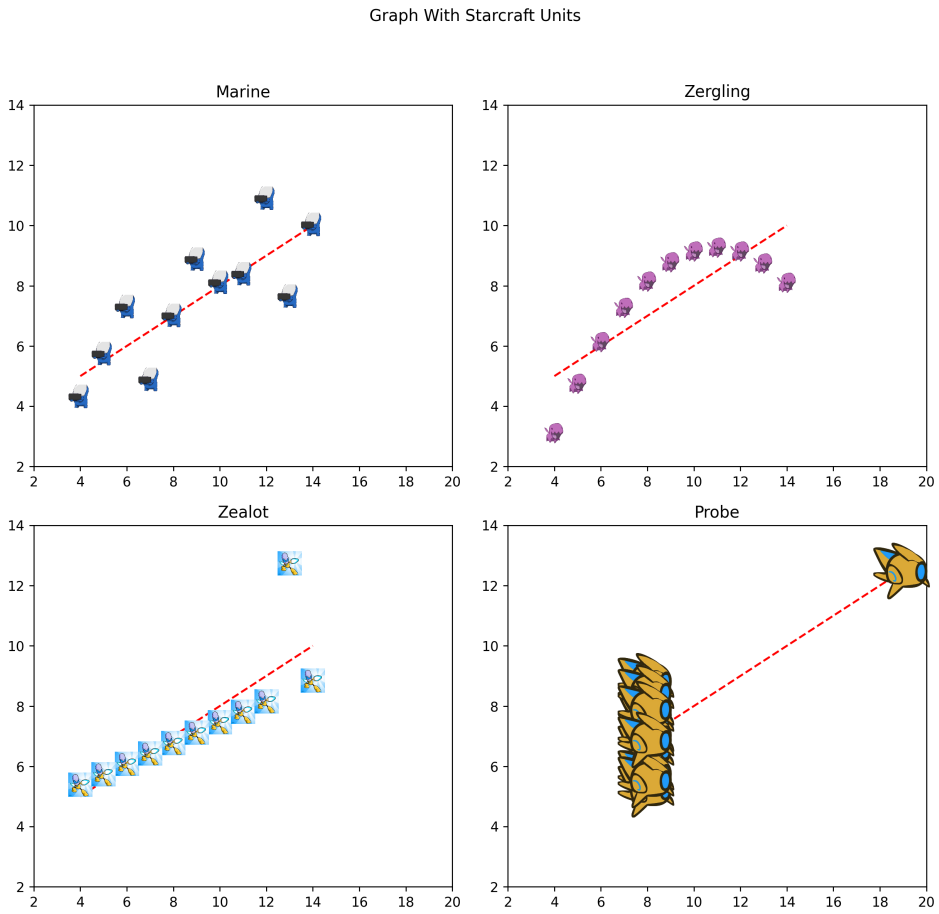

Figure 1: Scatter plots of the four datasets with regression line $\hat{y} = 3 + 0.5x$.

6.2 Residual Plots

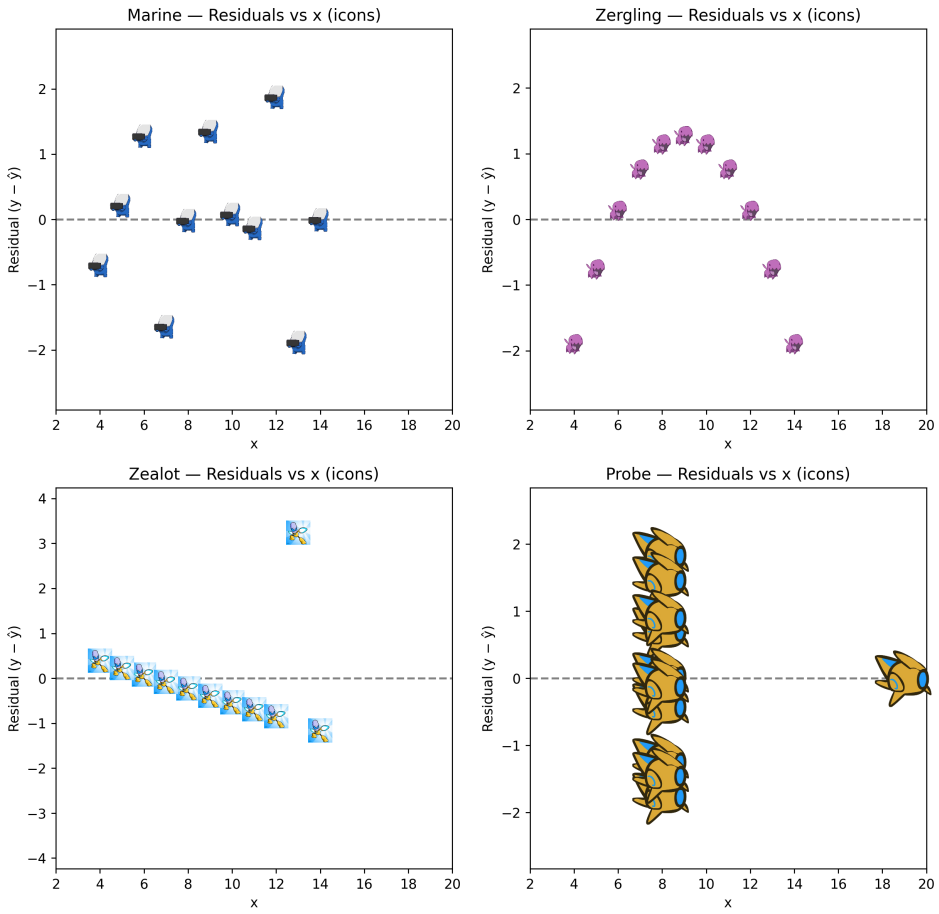

Figure 2: Residual plots showing deviation from the regression line for each dataset.

6.3 Box and Violin Plots

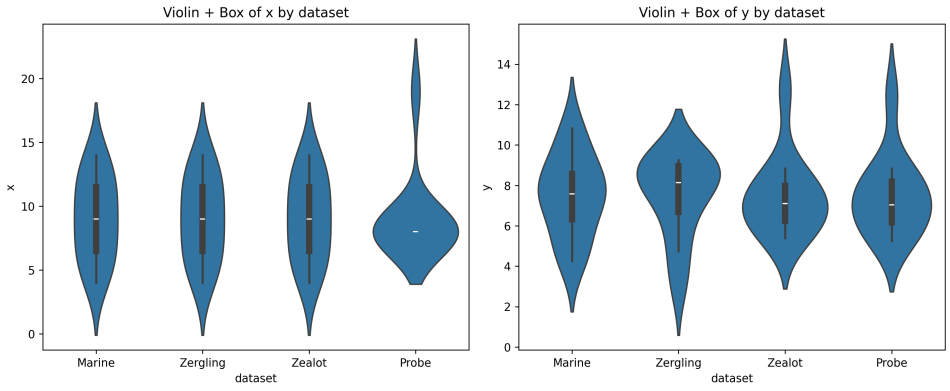

Figure 3: Boxplot (left) and violin plot (right) comparing x and y distributions.

6.4 Overlaid Comparison

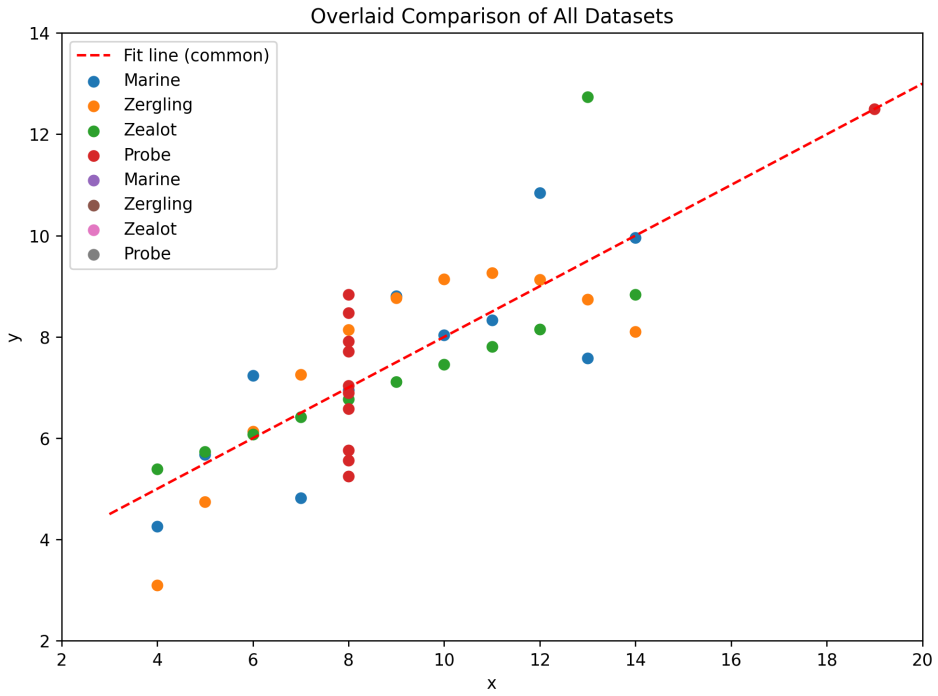

Figure 4: All four datasets plotted together with a common regression line.

7 Interpretation

Summary statistics provide convenient numerical summaries of data, but they can be misleading when interpreted in isolation. Datasets with identical means or variances can exhibit vastly different patterns, spreads, and outliers—characteristics that raw statistics fail to reveal.

In this analysis, the Marine and Zergling datasets have comparable average values, yet their spreads and visual distributions differ significantly, as shown in the boxplots and violin plots. These differences emphasize how visualization exposes nuances that summary tables overlook.

The scatter plots with regression lines and the residual plots further highlight the limitations of purely statistical summaries. Residual plots, in particular, reveal how well (or poorly) the linear model fits each dataset. For instance, the Zealot dataset displays larger deviations from the regression line, suggesting that a simple linear model does not capture its structure effectively.

Box and violin plots also visualize the internal variation of each dataset. While some distributions appear compact and consistent, others contain extreme outliers. A clear example is the Probe dataset, which contains a distinct outlier point that heavily influences both its regression line and residuals.

Overall, these visualizations uncover information invisible to raw statistics. By combining numerical summaries with visual plots, we achieve a more complete and accurate understanding of each dataset—reinforcing one of the key principles of Exploratory Data Analysis (EDA): *numbers summarize, but visuals reveal.*

8 Reproducibility

8.1 Environment and Dependencies

To ensure this analysis can be reproduced, both `requirements.txt` and `environment.yml` files are included.

requirements.txt

```
# Core libraries
numpy==1.26.4
pandas==2.2.3
matplotlib==3.9.2
seaborn==0.13.2
pillow==10.4.0

# Optional (if used for extra visuals or regression)
plotly==5.24.1
statsmodels==0.14.2
```

To install using pip:

```
pip install -r requirements.txt
```

environment.yml

```
name: anscombe-eda
channels:
  - defaults
  - conda-forge
dependencies:
  - python=3.11
  - numpy=1.26.4
  - pandas=2.2.3
  - matplotlib=3.9.2
  - seaborn=0.13.2
  - pillow=10.4.0
  - plotly=5.24.1
  - statsmodels=0.14.2
```

To create and activate the environment:

```
conda env create -f environment.yml
conda activate anscombe-eda
```

9 Conclusion and Future Work

This analysis demonstrates that numerical summaries alone cannot fully describe datasets. Visualizations reveal structure, patterns, and outliers invisible in tables.

Future improvements: Automate EDA reporting via Python notebooks.

Appendix: Full Code

All code is available in the GitHub repository: https://jz12456.github.io/Portfolio/

Example snippet:

```
def fit_line(x):
    return 3 + 0.5 * x

def plot_scatter_with_icons():
    fig, axes = plt.subplots(2, 2, figsize=(10, 10))
    axes = axes.flatten()
    for ax, (x, y, label) in zip(axes, datasets):
        x_line = np.array([min(x), max(x)])
        ax.plot(x_line, fit_line(x_line), 'r--', label='fit line')
        ax.set_title(label)
        ax.set_xlim(2, 20)
        ax.set_ylim(2, 14)
    plt.suptitle("Scatter Plots with Regression Lines")
    plt.show()
```