School of Computing
UNIVERSITY OF GEORGIA

CSCI 4380/6380 DATA MINING

Fei Dou

Assistant Professor
School of Computing
University of Georgia

August 23, 2023

Grading

4380 Section (Undergrads)
Homework (35%)
Test 1 (10%)
Test 2 (10%)
Final Exam (20%)
Term Project (25%)

6380 Section (Grads)
Homework (30%)
Test 1 (10%)
Test 2 (10%)
Presentation (10%)
Final Exam (20%)
Term Project (20%)

4380 Section (Undergrads)
Homework (35%)
Test 1 (10%)
Test 2 (10%)
Test 3 (10%) (tentative)
Term Project (35%) (or 45%)

6380 Section (Grads)
Homework (30%)
Test 1 (10%)
Test 2 (10%)
Test 3 (10%) (tentative)
Presentation (10%)
Term Project (30%) (or 40%)

Term Project Description

- Each team has 2 ~ 3 students. (or 1 if send email to me and explain)

- Consists of a presentation, a final report, and the executable source codes.
  - Presentations are scheduled before the last week, reports and codes are due at the end of the last week (Midnight Dec. 10);

- Any topics related to the course. Topic selection suggestions:
  - Standard tasks: https://www.kaggle.com/. (don't directly use the code or solution from Kaggle)
  - Real-world applications: Problems or applications that you are interested in.
  - Top conferences: KDD, NeurIPS, ICML, ICLR, AAAI, IJCAI, CVPR, ICCV, ECCV

- Detailed requirements to be introduced later.

Final Presentation

- November 26 - 28.
- All team members should present their slides.
- Each presentation is a talk (20 minutes) + QA (5 minutes) = 25 minutes.
- Slides should be emailed to the instructor before presentation.
- Content to be covered:
  - Background.
  - The formal problem definition.
  - The details of your solution.
  - Experimental results on the datasets, compared with baseline methods (at least three), under the evaluation metrics.
  - Conclusion.

Final Report

- Latex template: To be provided later
- Length: ≥ 6 pages + unlimited references Format:
  - Introduction
  - Literature Review (short)
  - Problem Definition
  - The Proposed Method
  - Experiments and Analysis
  - Conclusion
  - References
- Failing to following the template → 20% penalty.

Data Understanding

Data Quality

- Poor data quality negatively affects many data processing efforts

- Data mining example: a classification model for detecting people who are loan risks is built using poor data
  - Some credit-worthy candidates are denied loans
  - More loans are given to individuals that default

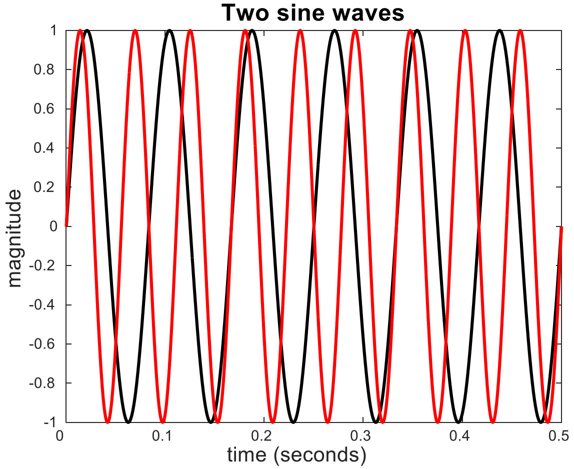
Data Quality

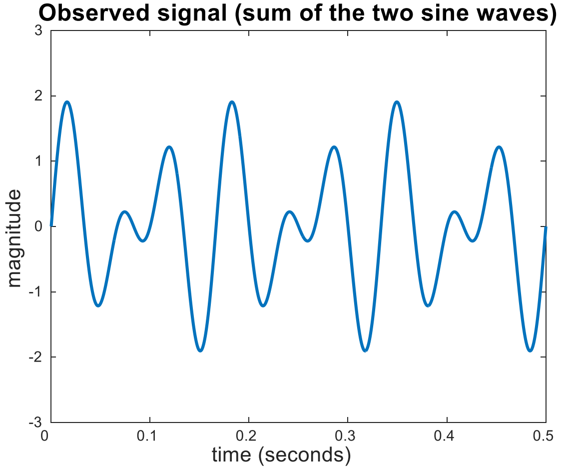
- What kinds of data quality problems?
- How can we detect problems with the data?
- What can we do about these problems?

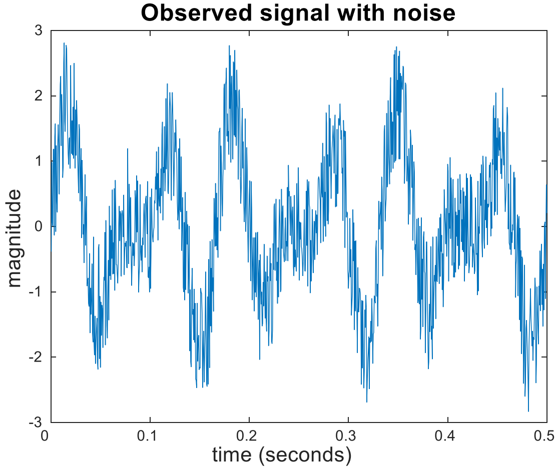
- Examples of data quality problems:
  - Noise and outliers
  - Wrong data
  - Fake data
  - Missing values
  - Duplicate data

Noise

- For objects, noise is an extraneous object
- For attributes, noise refers to modification of original values
  - Examples: distortion of a person’s voice when talking on a poor phone and “snow” on television screen
  - The figures below show two sine waves of the same magnitude and different frequencies, the waves combined, and the two sine waves with random noise
    - The magnitude and shape of the original signal is distorted

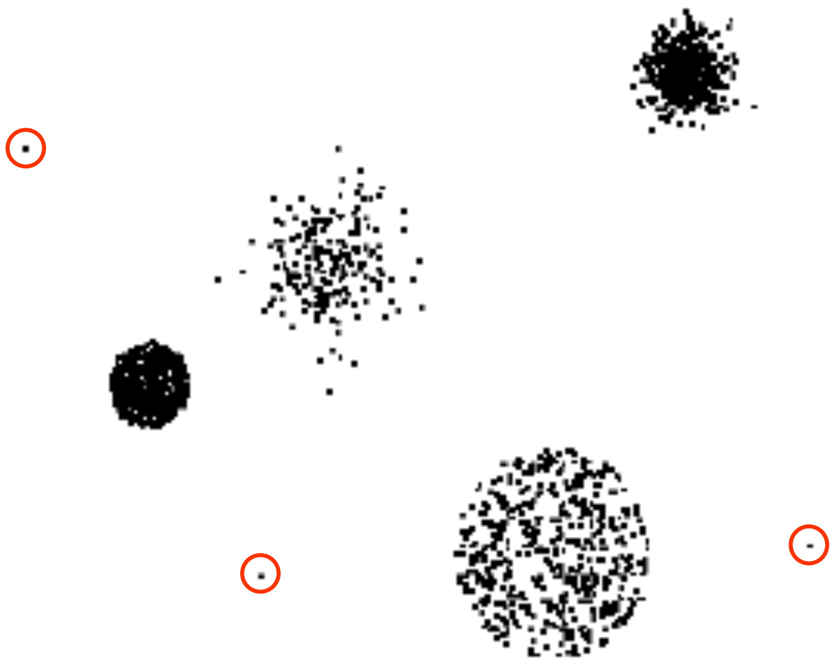






Outliers

- ***Outliers*** are data objects with characteristics that are considerably different than most of the other data objects in the data set
  - **Case 1:** Outliers are noise that interferes with data analysis
  - **Case 2:** Outliers are the goal of our analysis
    - Credit card fraud
    - Intrusion detection



- Causes?

Missing Values

- Reasons for missing values
  - Information is not collected
    (e.g., people decline to give their age and weight)
  - Attributes may not be applicable to all cases
    (e.g., annual income is not applicable to children)
  - Transmission error/preprocessing

	A	B	C
1	Original Data set		
2	Name	Age	Gender
3	Robin	28	Male
4	Heather	29	Female
5	Jamie	22	
6	Carl	32	Male
7		35	Male
8	Sarah	26	Female

Duplicate Data

- Data set may include data objects that are duplicates, or almost duplicates of one another
  - Major issue when merging data from heterogeneous sources

- Examples:
  - Same person with multiple email addresses

- Data cleaning
  - Process of dealing with duplicate data issues

- When should duplicate data not be removed?

Basic Statistical Description

- **Motivation**. To better understand the data: central tendency, variation, and spread
- **Data dispersion characteristics**: median, max, min, quantiles, outliers, variances, etc.
- **Numerical dimensions** correspond to sorted intervals.
  - Data dispersion: analyzed with multiple granularities of precision
  - Boxplot or quantile analysis on sorted intervals
- **Dispersion analysis on computed measures**
  - Folding measures into numerical dimensions
  - Boxplot or quantile analysis on the transformed cube

Measuring the Central Tendency

- **Mean:** sample vs. population

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ vs. } \mu = \frac{\sum x}{N}$$

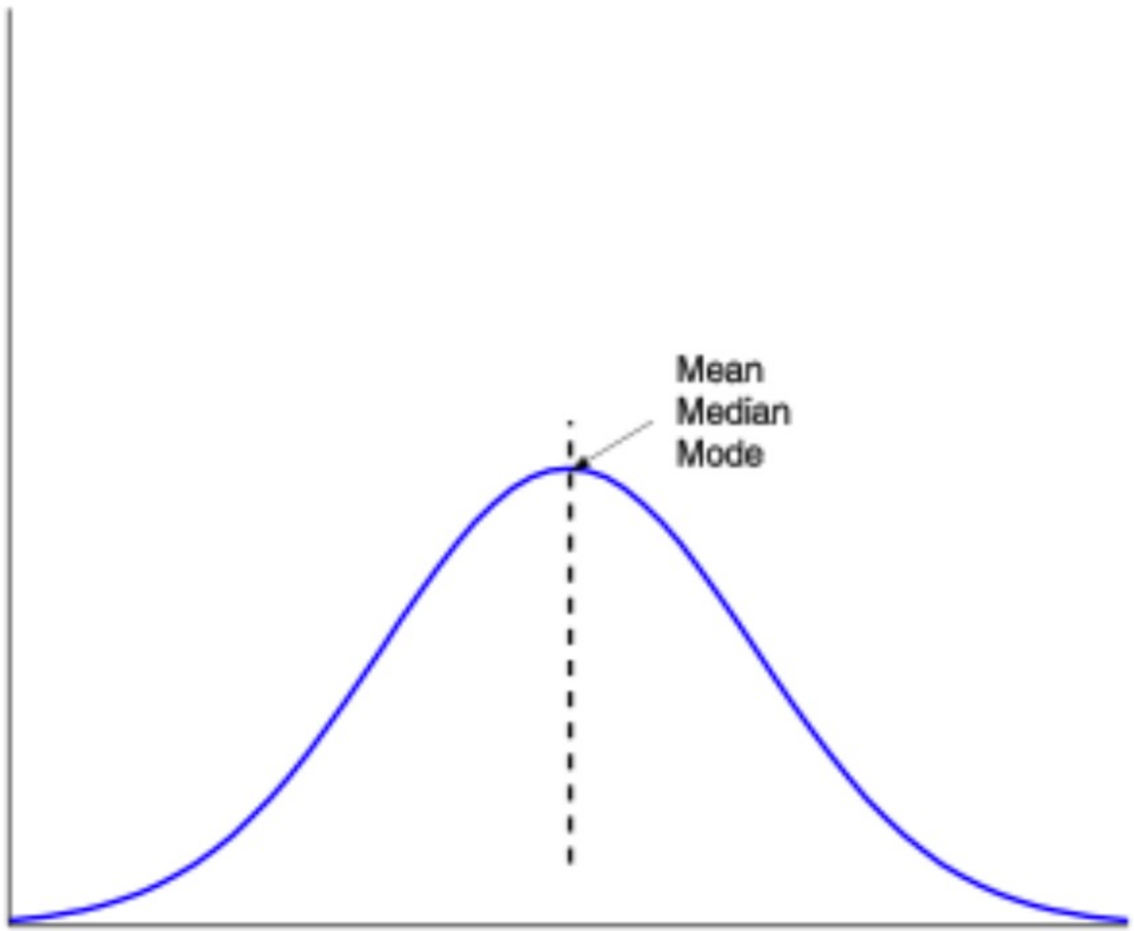
- **Median:** Middle value if odd number of values, or average of the middle two values otherwise

- **Mode:** Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula: mean-mode=3×(mean-median)

age	*frequency*
1–5	200
6–15	450
16–20	300
21–50	1500
51–80	700
81–110	44

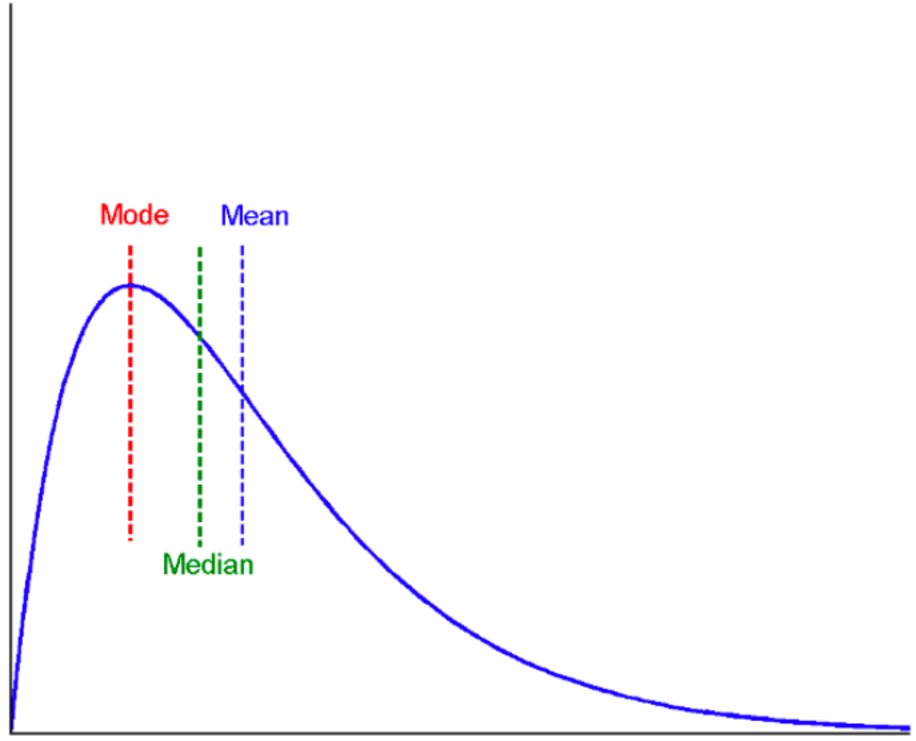
Example

- Median, mean and mode of symmetric data



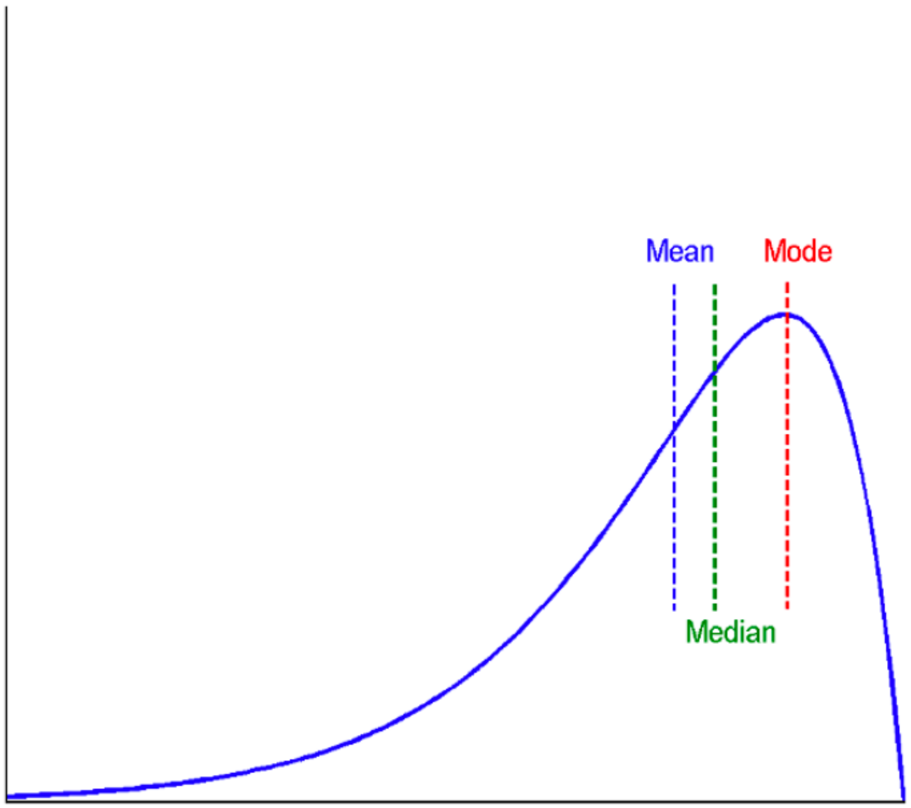
Example

- Median, mean and mode of positively skewed data



Example

- Median, mean and mode of negatively skewed data



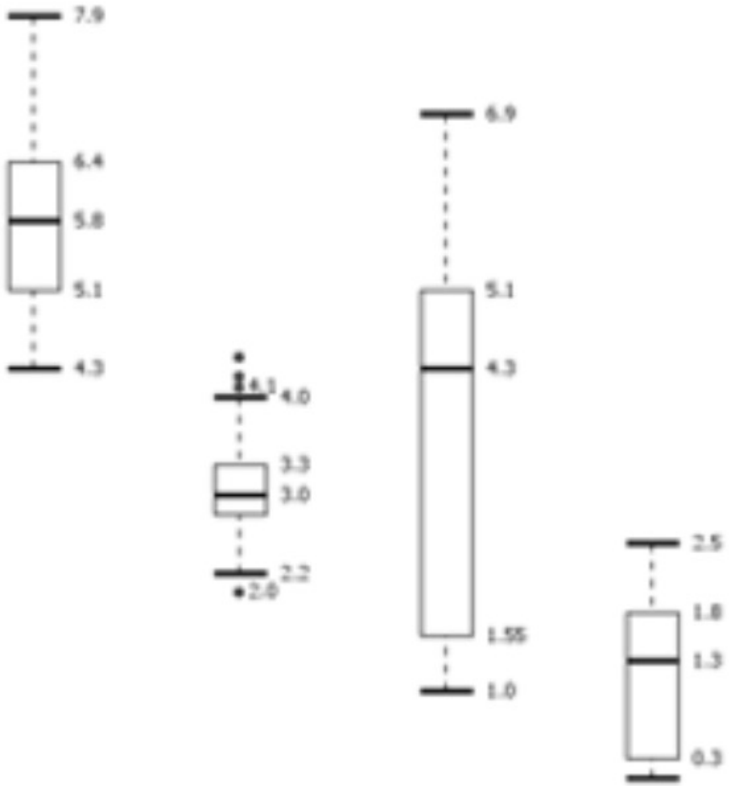
Measuring the Dispersion of Data

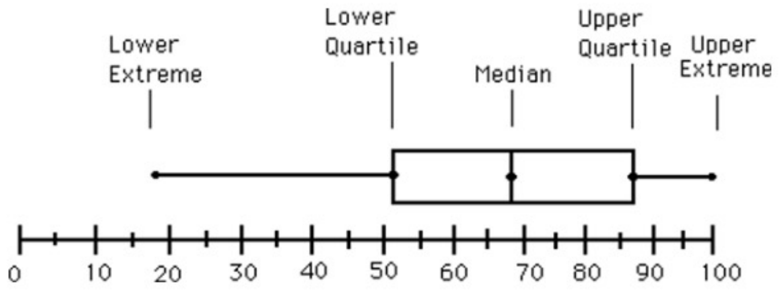
- Quartiles, outliers, and boxplots
  - **Quartiles**: $Q1$ (25th percentile), $Q3$ (75th percentile)
  - **Inter-quartile range**: IQR = $Q3 - Q1$
  - **Five number summary**: min, $Q1$, median, $Q3$, max
  - **Boxplot**: ends of the box are the quartiles; median is marked; add whiskers, and plot outliers individually
  - **Outlier**: usually, a value higher/lower than 1.5 x IQR
- Variance and standard deviation
  - **Variance**:

    $$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2, \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$
  - **Standard deviation**: s (or σ) square root of variance $s2$ or (σ2)

Boxplot Analysis

- **Five-number summary** of a distribution
  - Minimum, Q1, Median, Q3, Maximum
- **Boxplot**



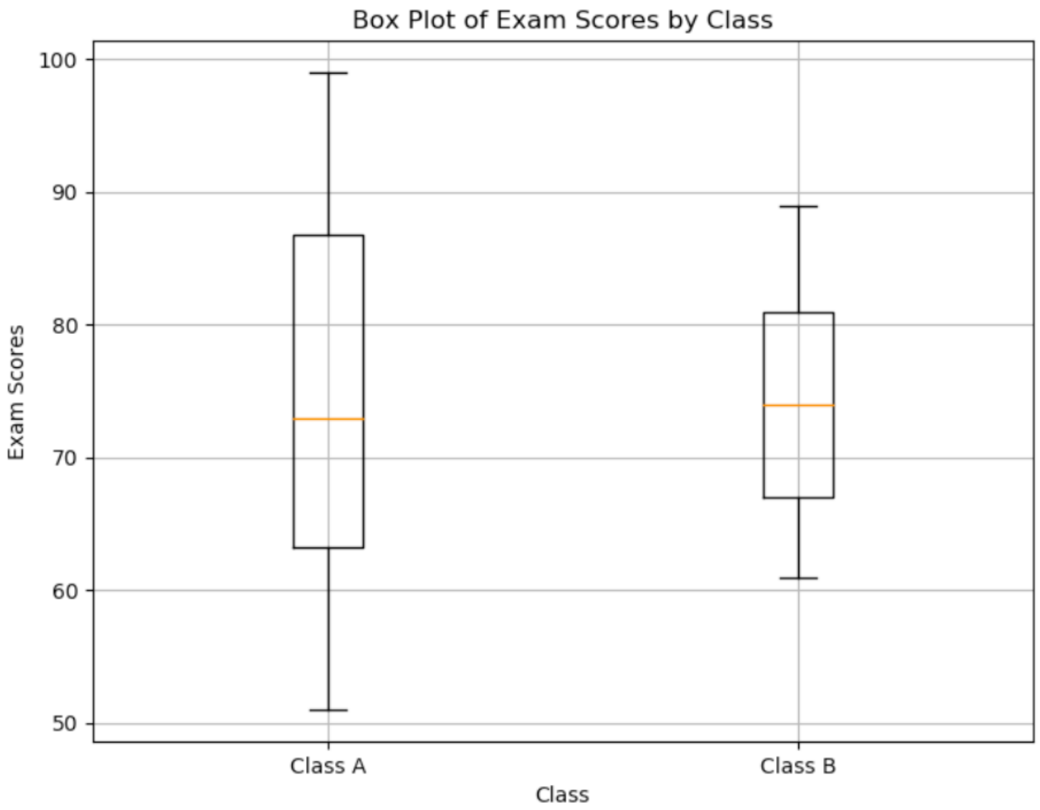


Displays of Basic Statistical Descriptions

- **Boxplot**: graphic display of five-number summary
- **Histogram**: x-axis are values, y-axis repres. frequencies
- **Quantile plot**: each value *xi* is paired with fi indicating that approximately 100 *fi* % of data are *i*
- **Quantile-quantile (q-q) plot**: graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- **Scatter plot**: each pair of values is a pair of coordinates and plotted as points in the plane

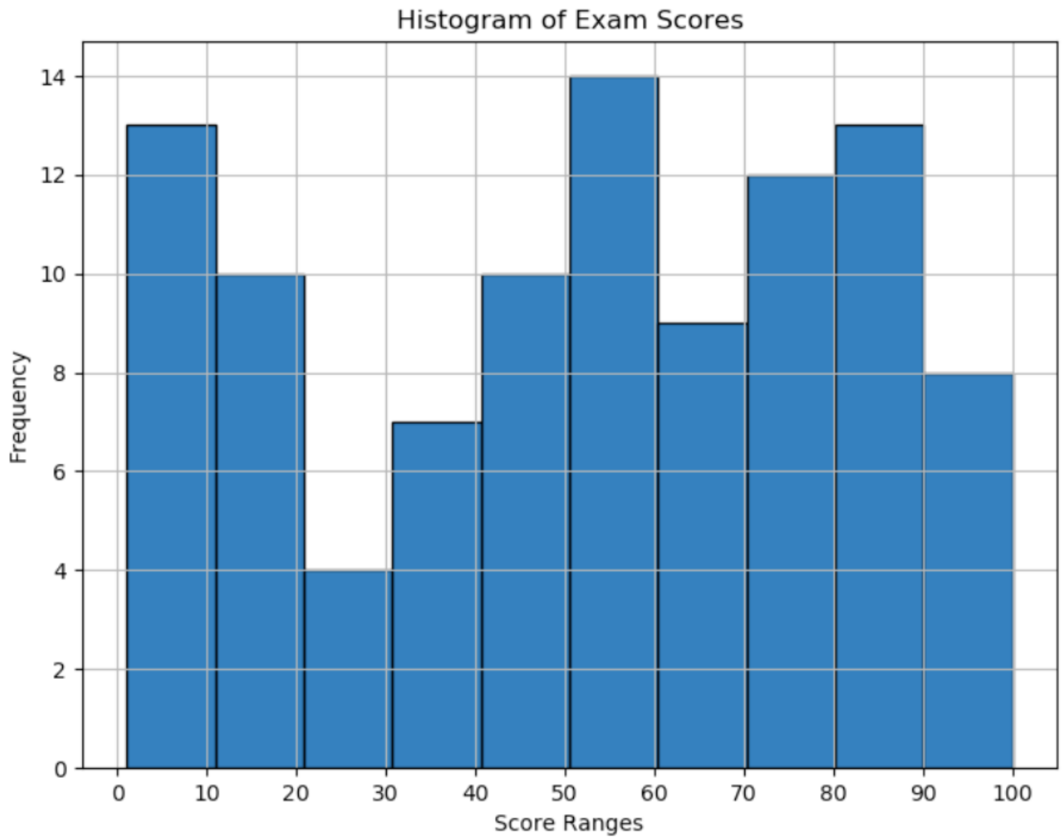
Boxplot

```python
import matplotlib.pyplot as plt
import numpy as np

# Generate exam scores for two classes
np.random.seed(42)
class_A_scores = np.random.randint(50, 100, 50)  # Generate 50 scores between 50 and 100
class_B_scores = np.random.randint(60, 90, 50)   # Generate 50 scores between 60 and 90

# Create a box plot
plt.figure(figsize=(8, 6))
plt.boxplot([class_A_scores, class_B_scores], labels=['Class A', 'Class B'])
plt.title('Box Plot of Exam Scores by Class')
plt.xlabel('Class')
plt.ylabel('Exam Scores')
plt.grid(True)
plt.show()
```



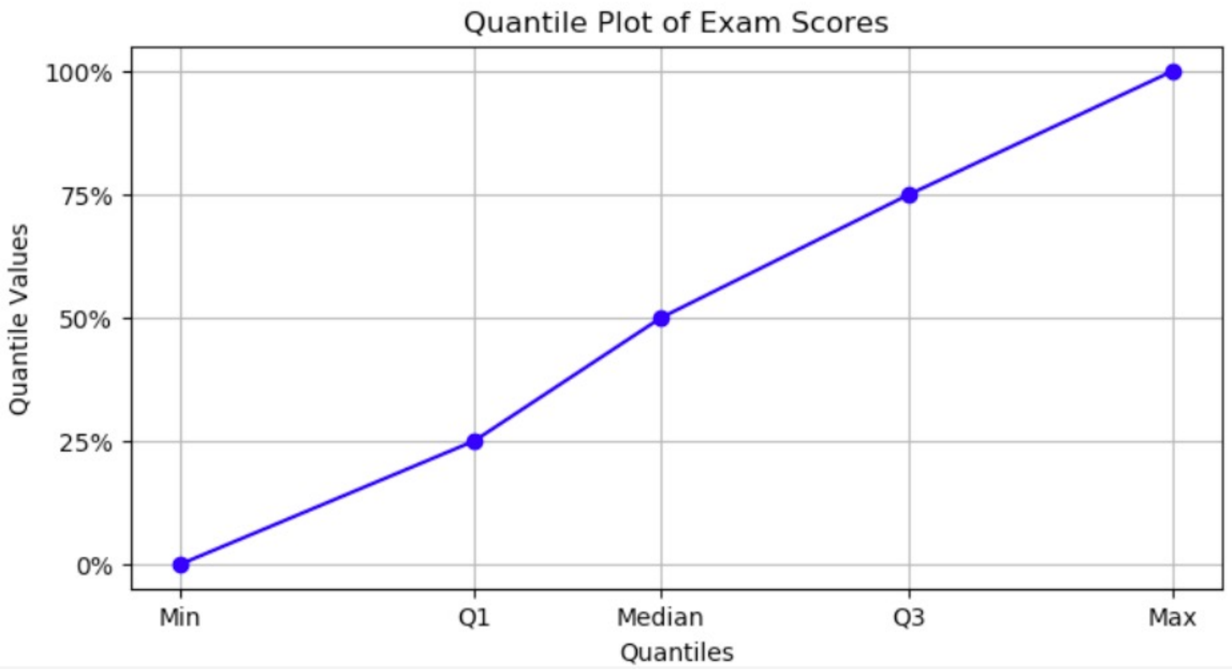
Histogram Analysis-Example 1

```python
import matplotlib.pyplot as plt
import numpy as np

# Generate a dataset of exam scores
np.random.seed(42)
exam_scores = np.random.randint(0, 101, 100)  # Generate 100 scores between 0 and 100

# Create a histogram plot
plt.figure(figsize=(8, 6))
plt.hist(exam_scores, bins=10, edgecolor='black')  # Divide scores into 10 bins
plt.title('Histogram of Exam Scores')
plt.xlabel('Score Ranges')
plt.ylabel('Frequency')
plt.xticks(range(0, 101, 10))  # Set x-axis tick labels
plt.grid(True)
plt.show()
```



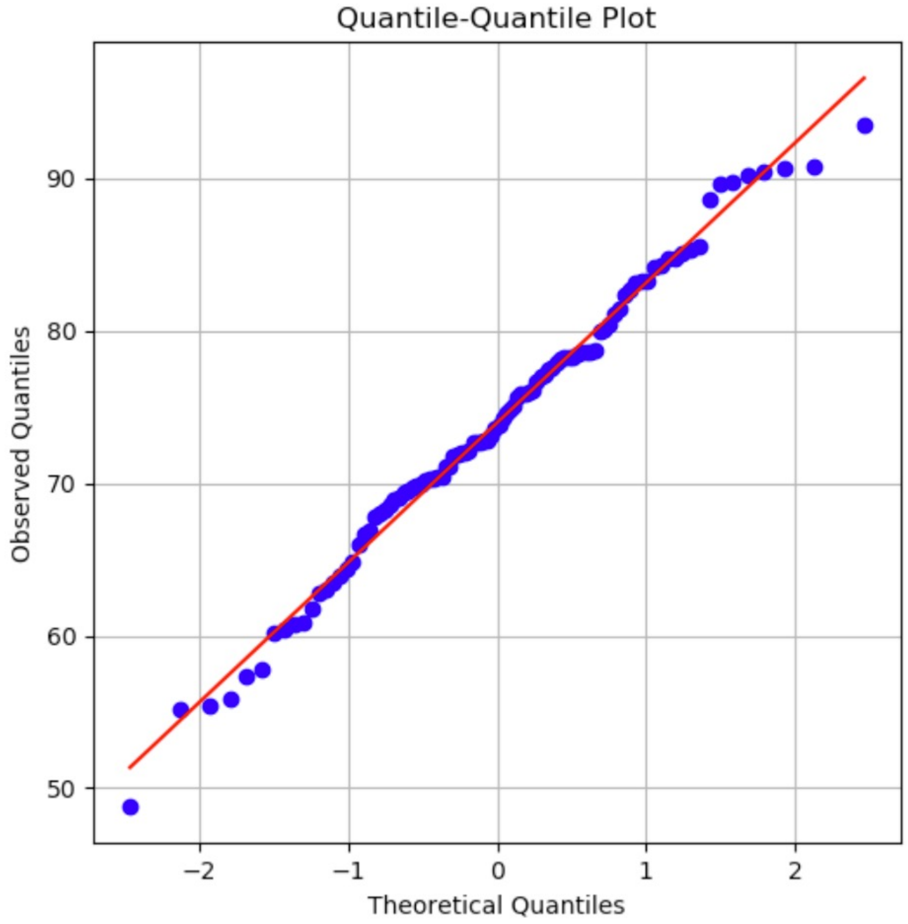
Quantile Plot

```python
import matplotlib.pyplot as plt
import numpy as np

# Exam scores dataset
scores = [65, 72, 75, 78, 82, 85, 88, 90, 92, 95,
          98, 100, 105, 110, 112, 115, 118, 120, 125, 130]

# Sorting the data
sorted_scores = sorted(scores)

# Dividing into quantiles
quantiles = np.percentile(sorted_scores, [0, 25, 50, 75, 100])

# Plotting the quantile plot
plt.figure(figsize=(8, 4))
plt.plot(quantiles, [0, 1, 2, 3, 4], marker='o', linestyle='-', color='blue')
plt.title('Quantile Plot of Exam Scores')
plt.xlabel('Quantiles')
plt.ylabel('Quantile Values')
plt.xticks(quantiles, ['Min', 'Q1', 'Median', 'Q3', 'Max'])
plt.yticks([0, 1, 2, 3, 4], ['0%', '25%', '50%', '75%', '100%'])
plt.grid(True)
plt.show()
```

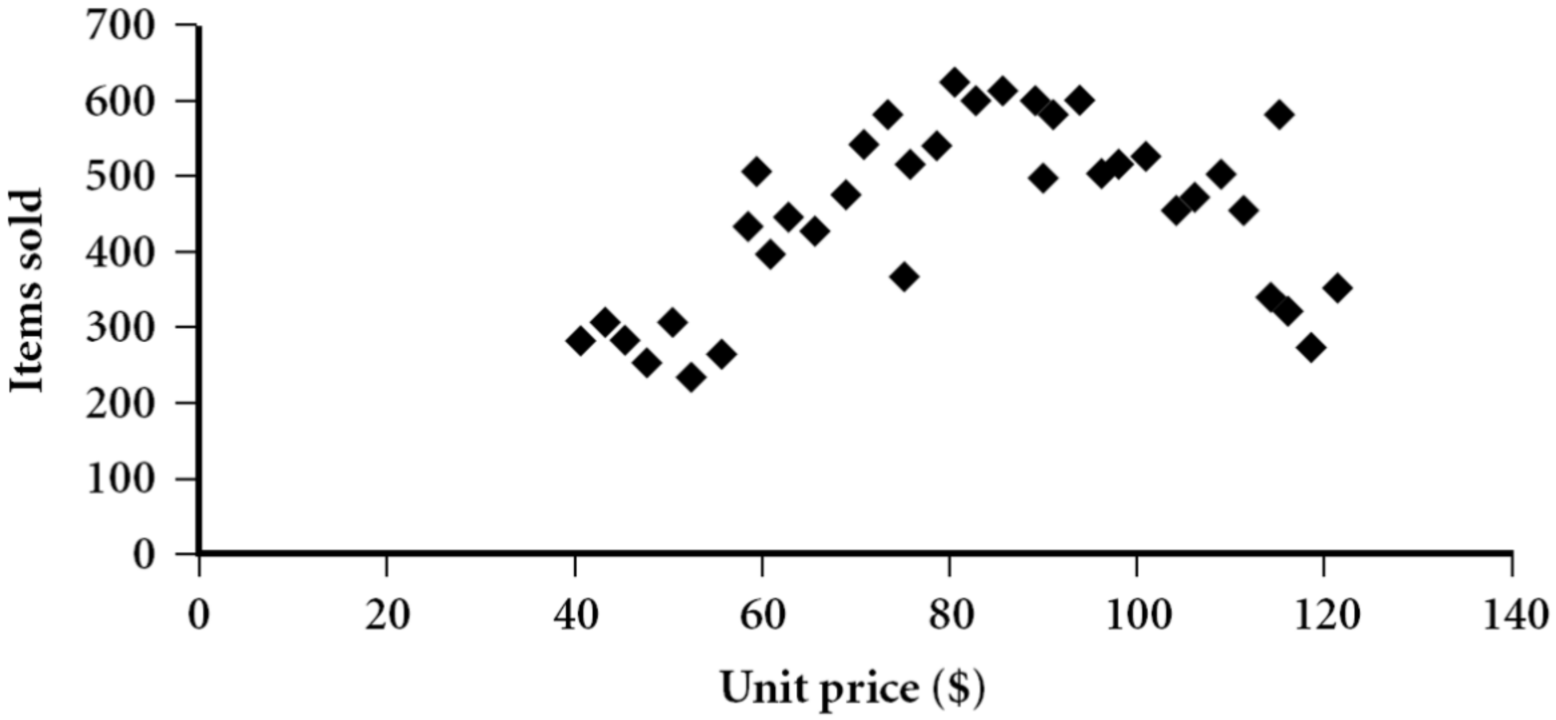


Quantile-Quantile Plot

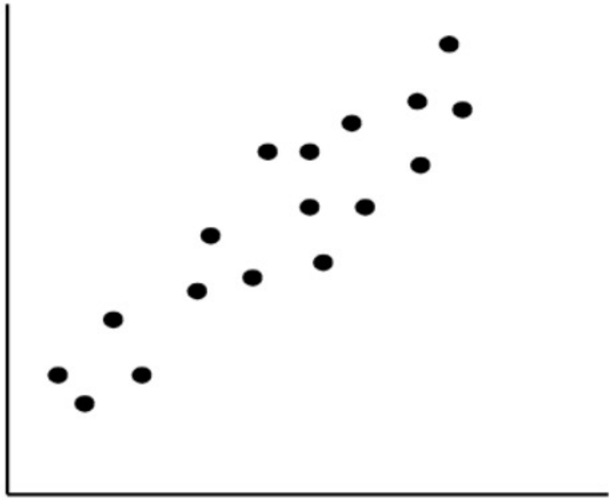
```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# Generate a dataset of observed exam scores
np.random.seed(42)
observed_scores = np.random.normal(75, 10, 100)  # Mean = 75, Standard Deviation = 10

# Create a Q-Q plot
plt.figure(figsize=(6, 6))
stats.probplot(observed_scores, dist='norm', plot=plt)
plt.title('Quantile-Quantile Plot')
plt.xlabel('Theoretical Quantiles')
plt.ylabel('Observed Quantiles')
plt.grid(True)
plt.show()
```

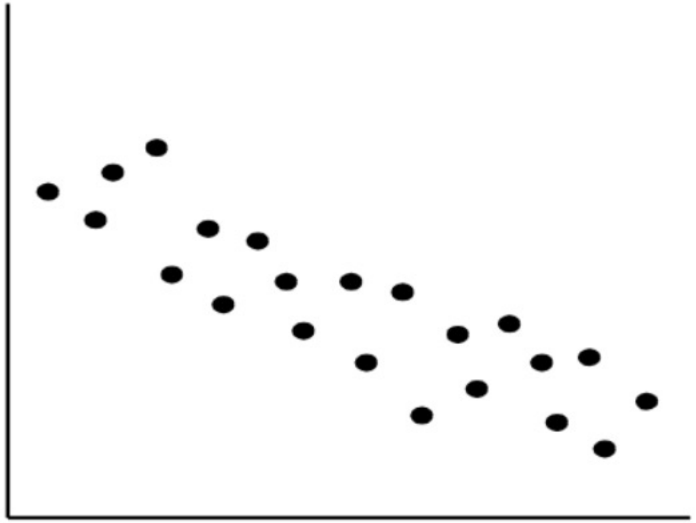


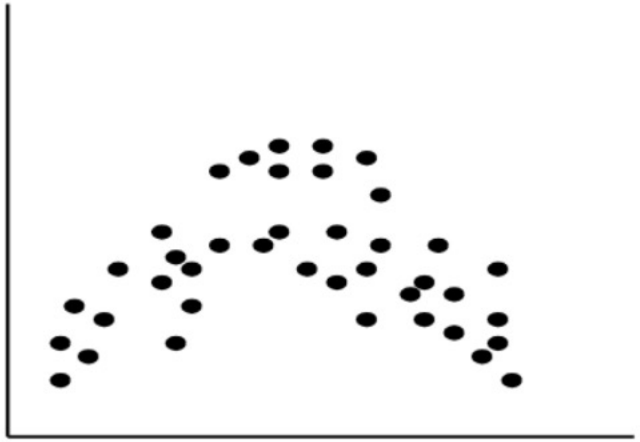
Scatter Plot



Positively and Negatively Correlated Data







- The left half fragment is positively correlated
- The right half is negative correlated

Uncorrelated Data