School of Computing
UNIVERSITY OF GEORGIA

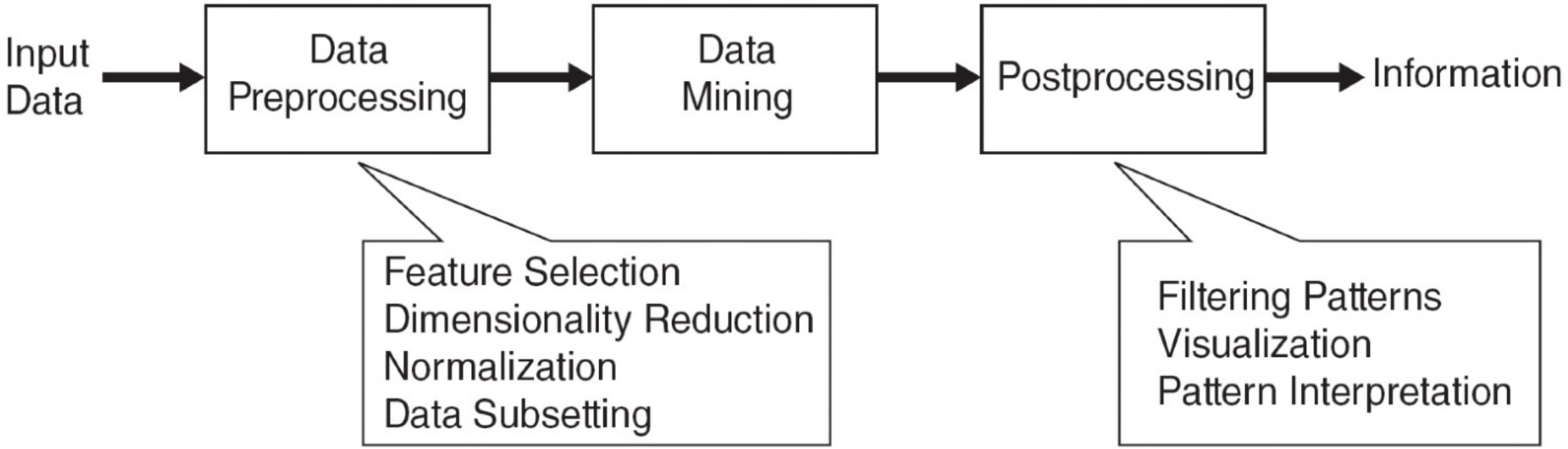
CSCI 4380/6380 DATA MINING

Fei Dou

Assistant Professor
School of Computing
University of Georgia

August 24, 2023

Recap: Data Mining Process



Recap: 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

Recap: 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

Recap: 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

Recap: 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**:

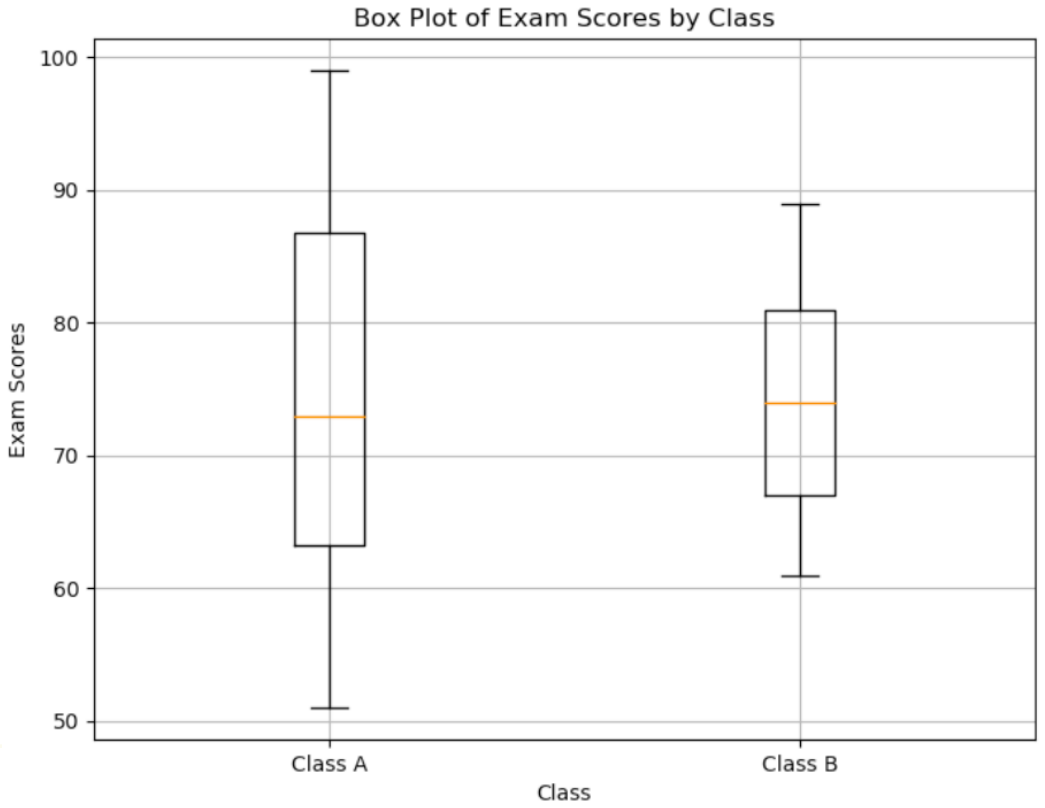
    $$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2,\ \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$
  - **Standard deviation**: s (or σ) square root of variance $s2$ or (σ2)

Recap: 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

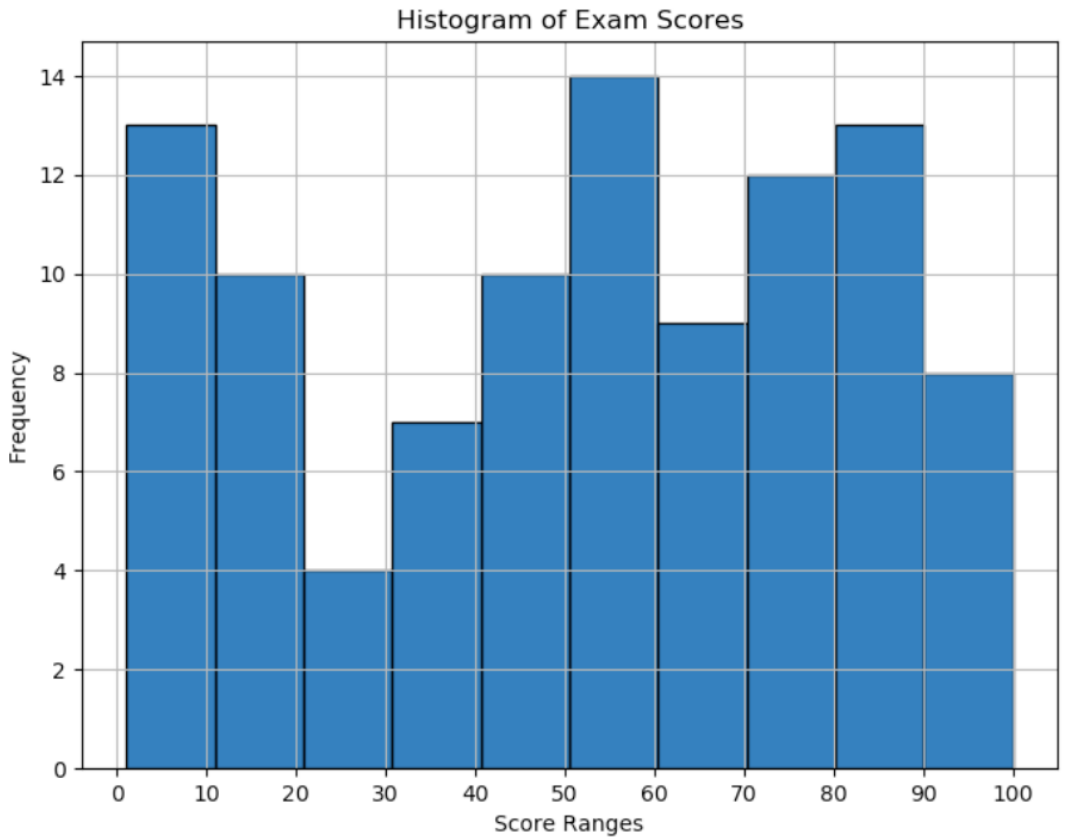
Boxplot

```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()
```



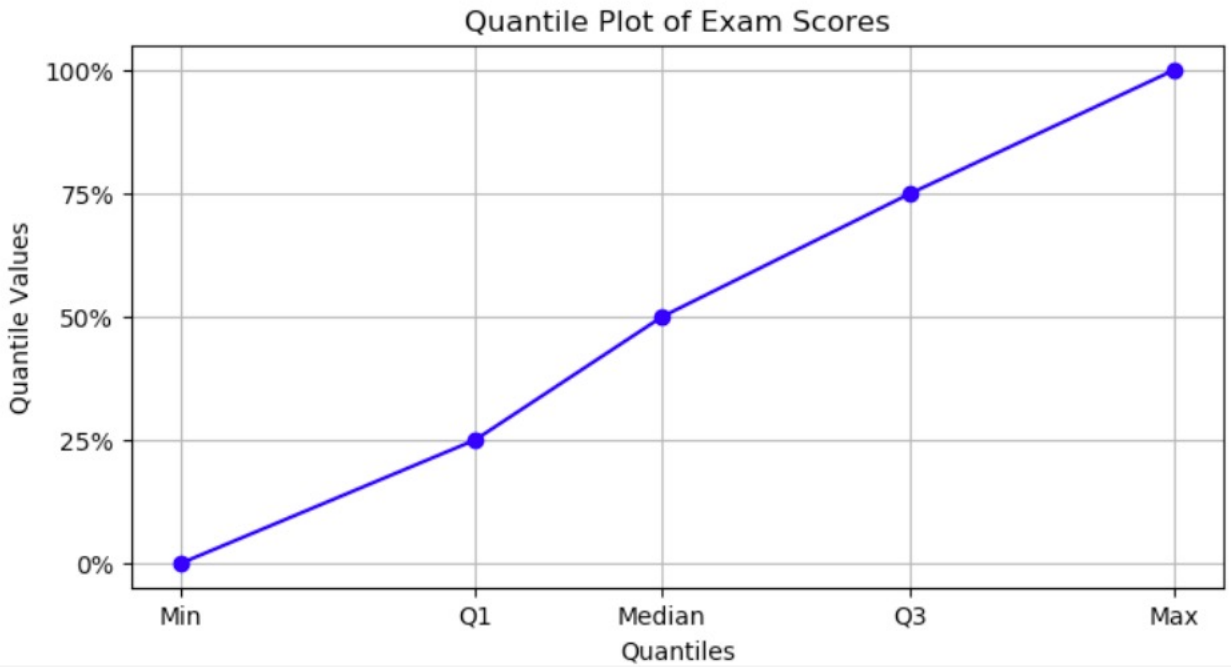
Histogram Analysis-Example 1

```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()
```



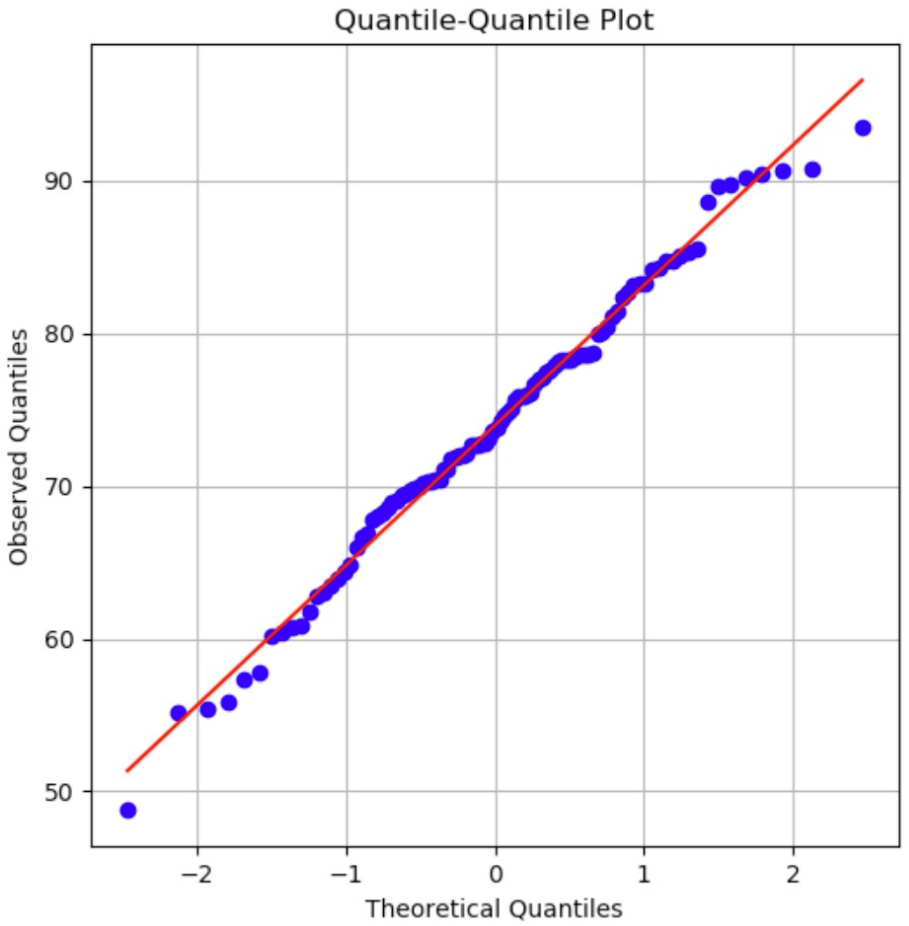
Quantile Plot

```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()
```

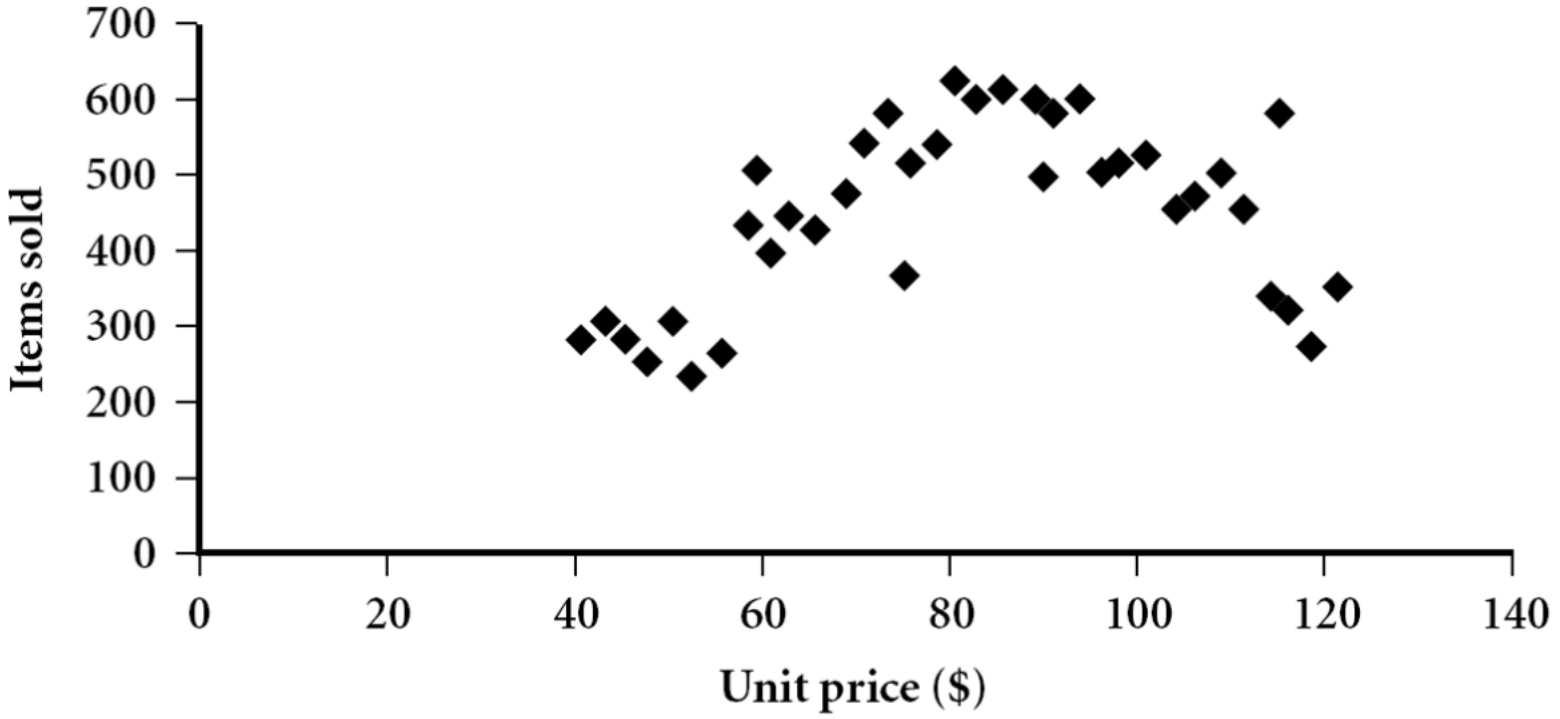


Quantile-Quantile Plot

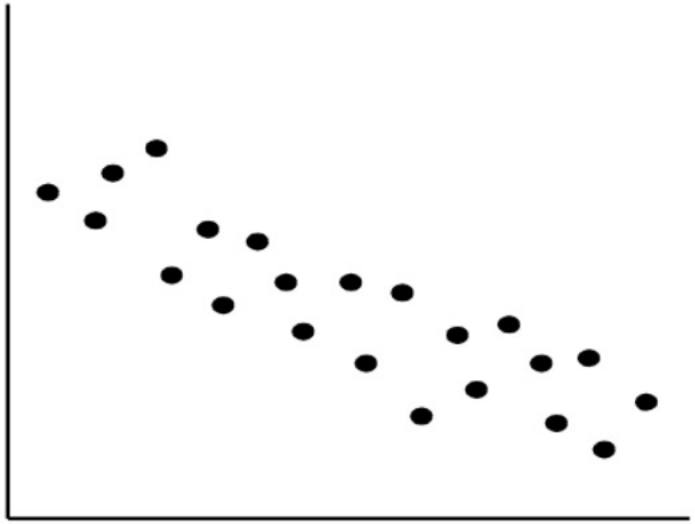
```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# 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()
```

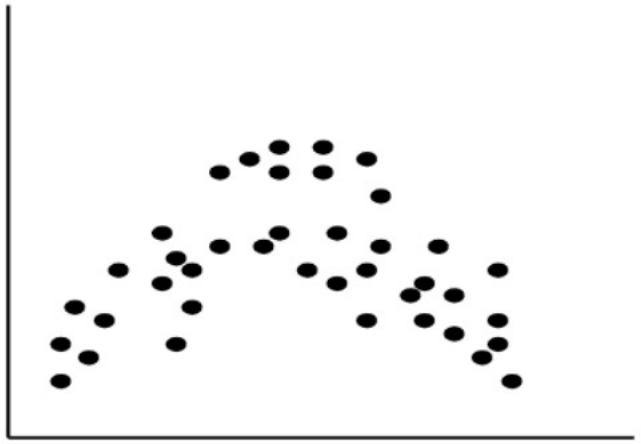


Scatter Plot



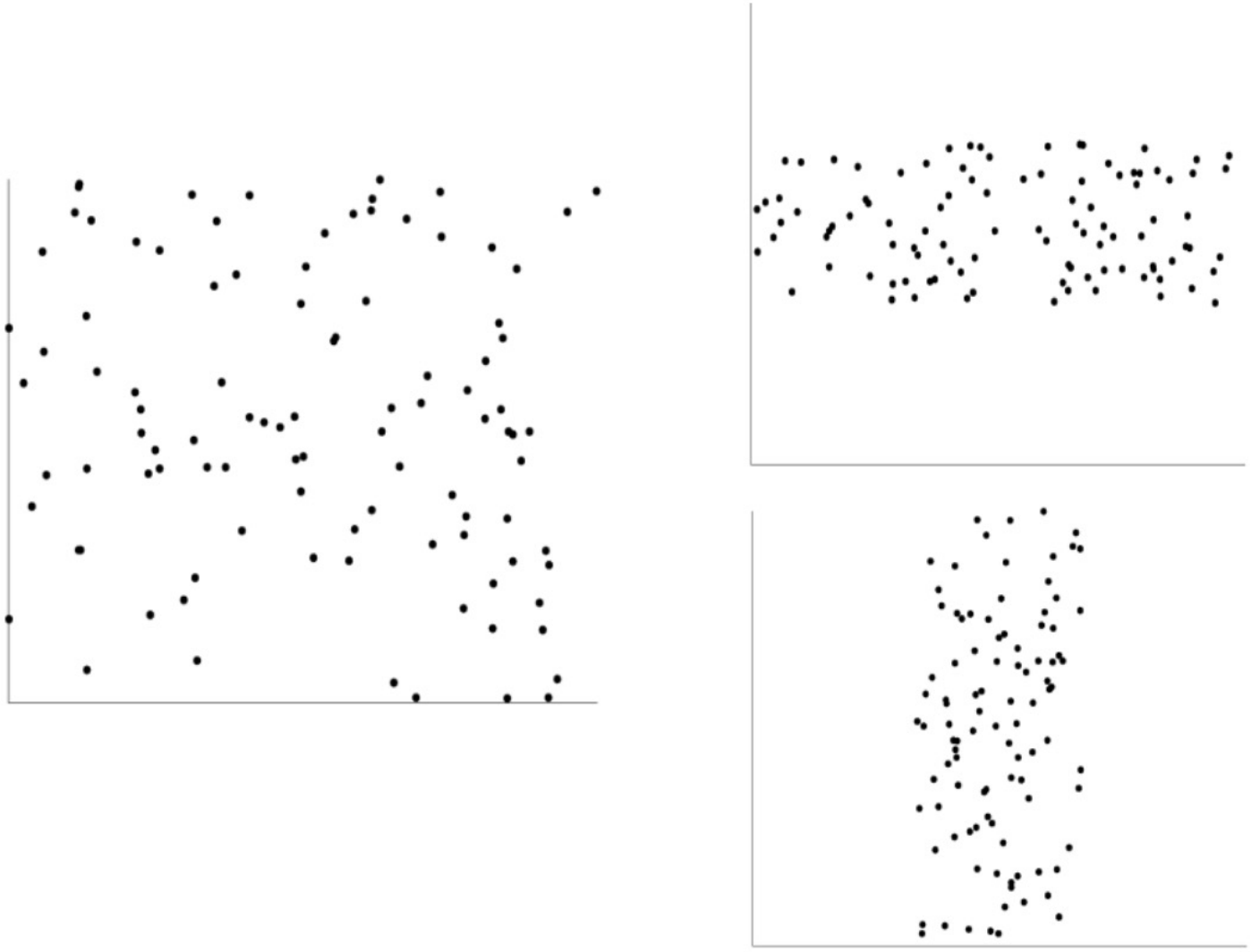
Positively and Negatively Correlated Data





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

Uncorrelated Data



Tasks

- Google Colab
- Python Basics
- NumPy
- Pandas
- Matplotlib
- *Pycharm/Visual Studio + Anaconda

Google Colab

- Google Colab is a free cloud service and now it supports free GPU!
- improve your Python programming language coding skills.
- develop deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV.
- The most important feature that distinguishes Colab from other free cloud services is; Colab provides GPU and is totally free.
- Details: https://research.google.com/colaboratory/faq.html

- **https://colab.research.google.com**

Python

- What is Python?
- Why use Python?
- What can Python do?

NumPy

- What is NumPy?
- Why use NumPy?
- Why is NumPy Faster Than Lists?
- Where is the NumPy Codebase?
  - https://github.com/numpy/numpy
- Totorials:
  - https://numpy.org/doc/stable/user/quickstart.html

Pandas

- What is Pandas?
- Why use Pandas?
- What Can Pandas Do?
- Where is the Pandas Codebase?
  - https://github.com/pandas-dev/pandas
- Tutorials:
  - https://pandas.pydata.org/docs/getting_started/tutorials.html

Matplotlib

- What is Matplotlib?

- Tutorials:
  - https://matplotlib.org/stable/tutorials/index.html

PyCharm/Visual Studio Code + Anaconda