GEOMATICS ENGINEERING DEPARTMENT

SECOND YEAR GEOMATICS

COMPUTER APPLICATIONS I

LECTURE NO: 3

DATA EXPLORATION TECHNIQUES

Dr. Eng. Reda FEKRY

Assistant Professor of Geomatics
reda.abdelkawy@feng.bu.edu.eg

Overview Of Today's Lecture

- Overview of Data Exploration in Python
- Code Examples Using Different Libraries
- Summary

DATA EXPLORATION TECHNIQUES

Data Cleaning and Preparation (using pandas)

```
import pandas as pd

# Load data from CSV file (replace 'data.csv' with your actual file path)
data = pd.read_csv("data.csv")

# Identify and handle missing values (e.g., filling with mean)
data["elevation"] = data["elevation"].fillna(data["elevation"].mean())

# Filter data based on specific criteria (e.g., selecting points within a specific area)
filtered_data = data[(data["X"] > 500000) & (data["X"] < 501000) & (data["Y"] > 4000000) & (data["Y"] < 4000200)]

# Explore descriptive statistics (optional)
print(data["elevation"].describe())  # Shows mean, median, etc. for elevation

# Prepare data for further analysis
clean_data = filtered_data[["X", "Y", "elevation"]]  # Select relevant columns
```

GEOMETRIC CALCULATIONS (USING SHAPELY)

- from shapely.geometry import Polygon
- # Define polygon points (replace with your coordinates)
- points = [(500000, 4000000), (500100, 4000000), (500100, 4000100), (500000, 4000100)]
- # Create a polygon object
- polygon = Polygon(points)
- # Calculate area of the polygon
- area = polygon.area # Area in square units (modify units based on data)
- # Additional calculations (optional)
- # - Calculate perimeter (polygon.length)
- # - Check if a point lies within the polygon (point.within(polygon))

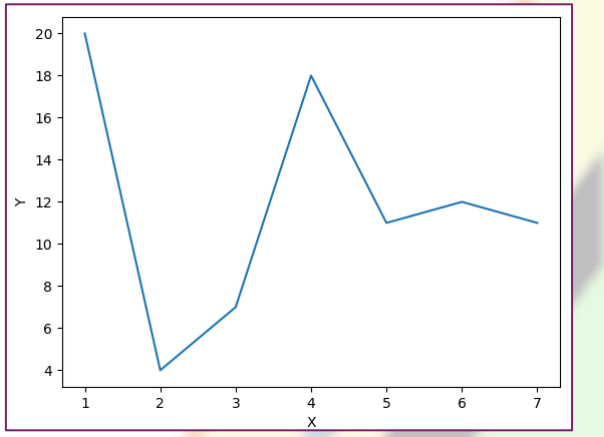
Coordinate Transformations (using pyproj)

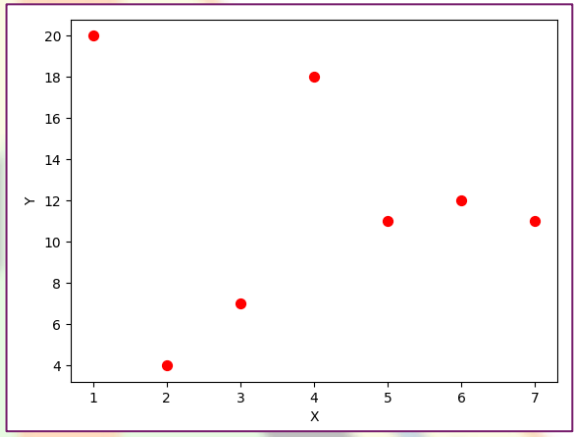
```
from pyproj import CRS, Transformer

# Define source and destination coordinate reference systems (CRS)
# Replace with your specific CRS codes (e.g., EPSG:4326 for WGS84)
source_crs = CRS.from_epsg(32617)  # Example: UTM Zone 17 North
destination_crs = CRS.from_epsg(4326)  # Example: WGS84 Geographic

# Define points to transform (replace with your coordinates)
points = [(500000, 4000000), (500100, 4000000)]
# Create a transformer object
transformer = Transformer.from_crs(source_crs, destination_crs)
# Transform the points
transformed_points = []
for point in points:
  x, y = point
  x_transformed, y_transformed = transformer.transform(x, y)
  transformed_points.append((x_transformed, y_transformed))
# Print the transformed points in the new CRS
print("Transformed points:", transformed_points)
```

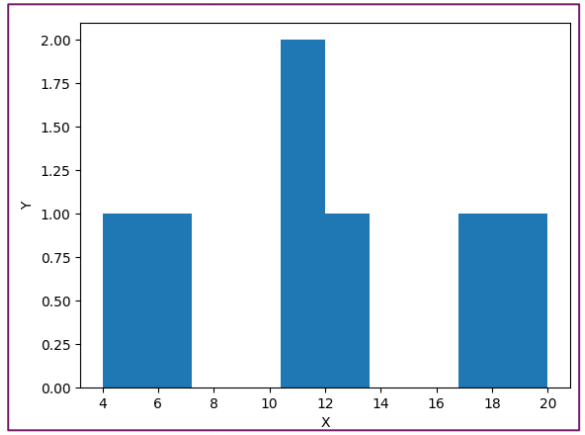
Data Visualization (using matplotlib)

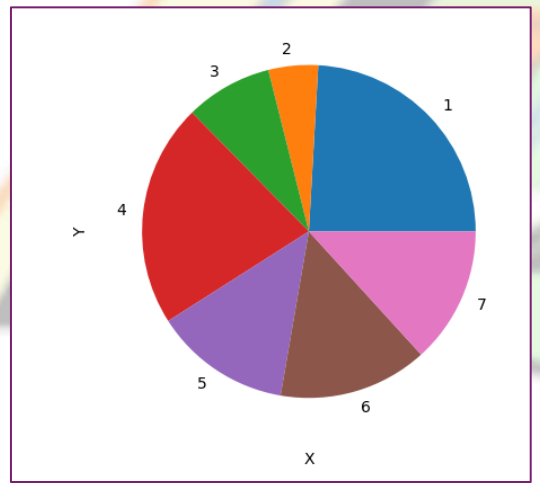
- Different types of plots are available: -

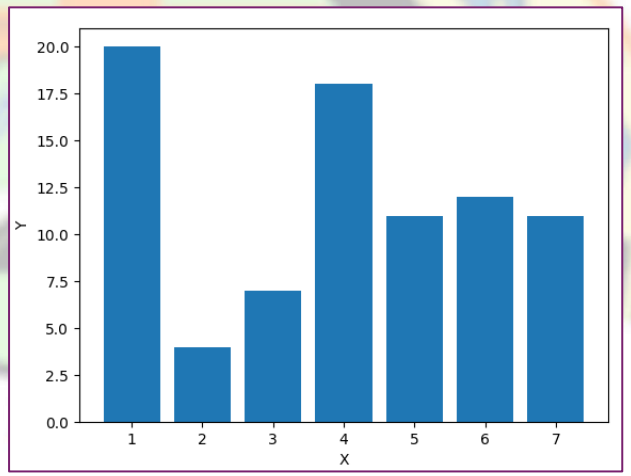
1. Line
2. Scatter
3. Bar
4. Histogram
5. Pie chart
6. Box plot
7. Area
8. Heatmap

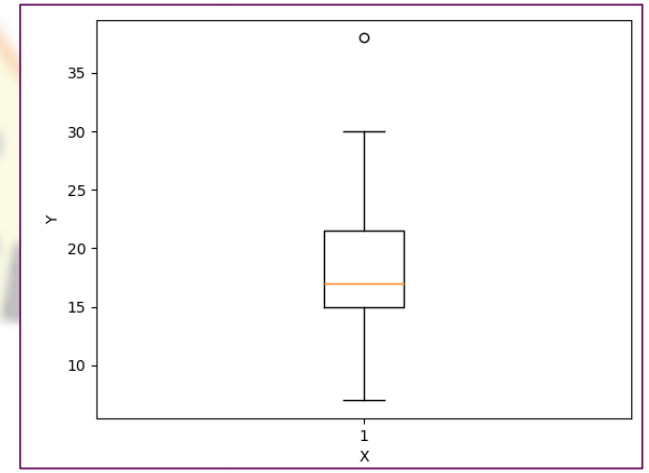












Data Visualization (using matplotlib)

- import matplotlib.pyplot as plt
- # Load elevation data from the clean data (assuming a column named 'elevation')
- elevations = clean_data["elevation"]
- # Create a histogram to visualize elevation distribution
- plt.hist(elevations)
- plt.xlabel("Elevation (meters)")
- plt.ylabel("Frequency")
- plt.title("Histogram of Elevation Data")
- plt.show()

END OF PRESENTATION

THANK YOU FOR ATTENTION!