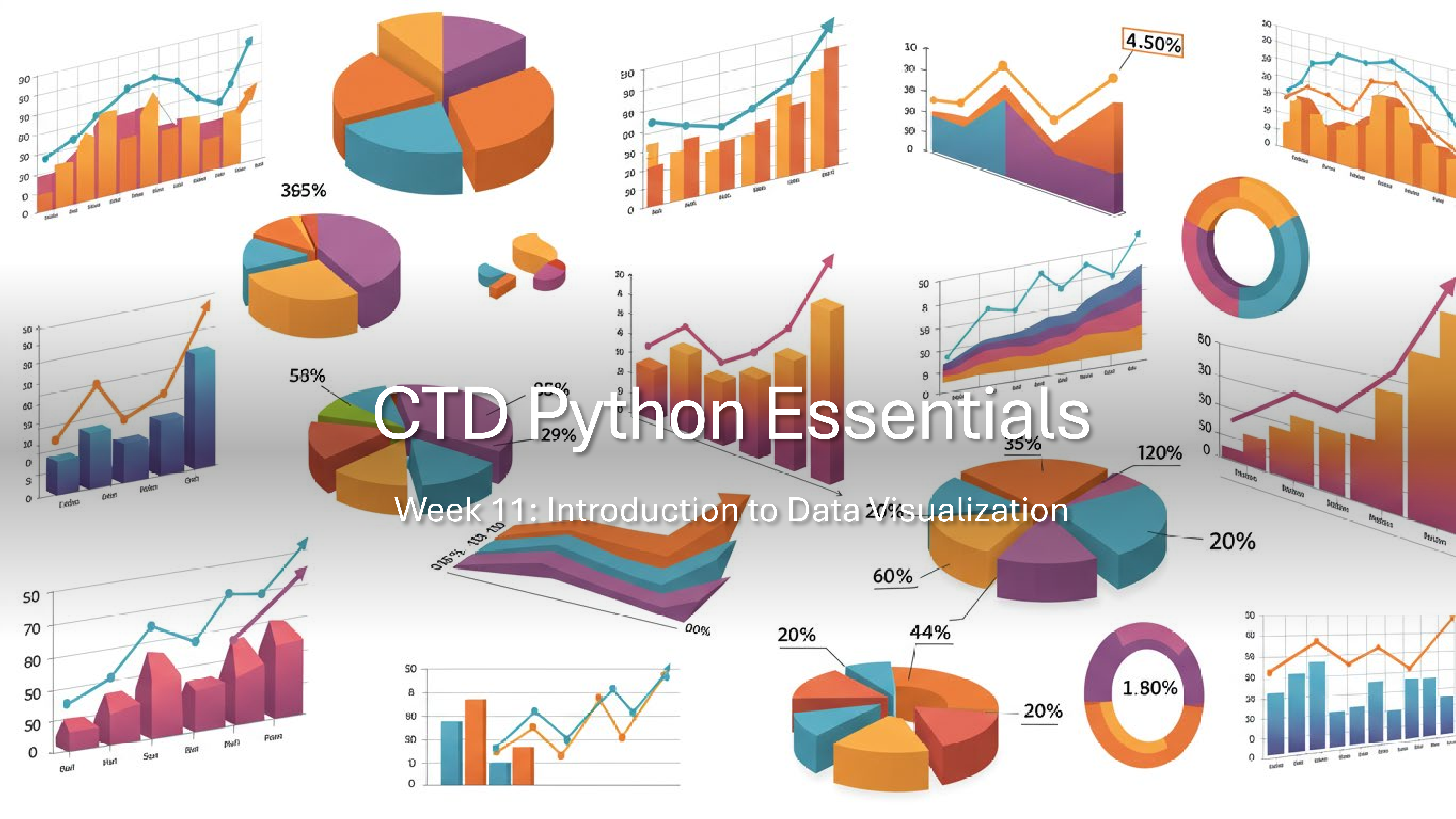


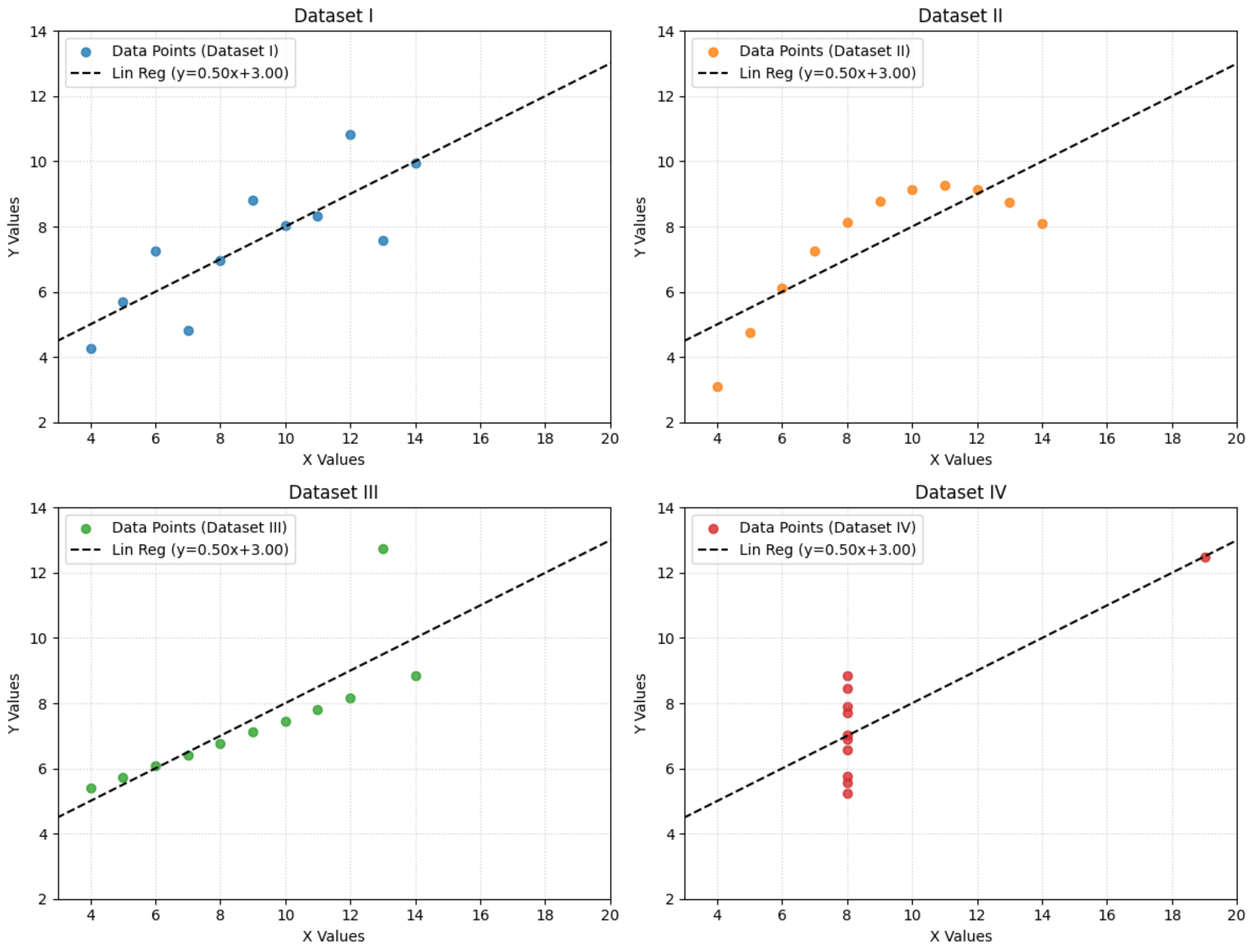
CTD Python Essentials

Week 11: Introduction to Data Visualization

A Picture is Worth 1000 Words

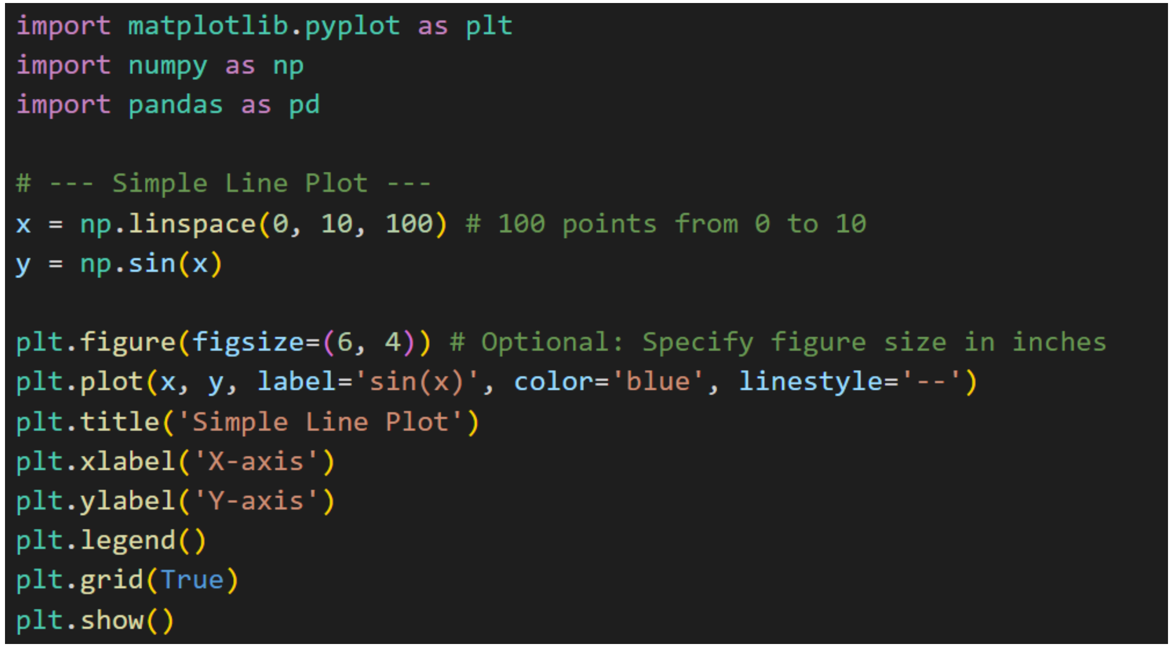
- Images engage the powerful parts of our brains which perform visual processing
    - Estimated at 50% of the cortex (surface of) the brain!
- It’s a way to gain intuition and insights which would be difficult to find using numerical metrics
- An image can instantly convey meaning

Anscombe's Quartet: Why Visualization Matters

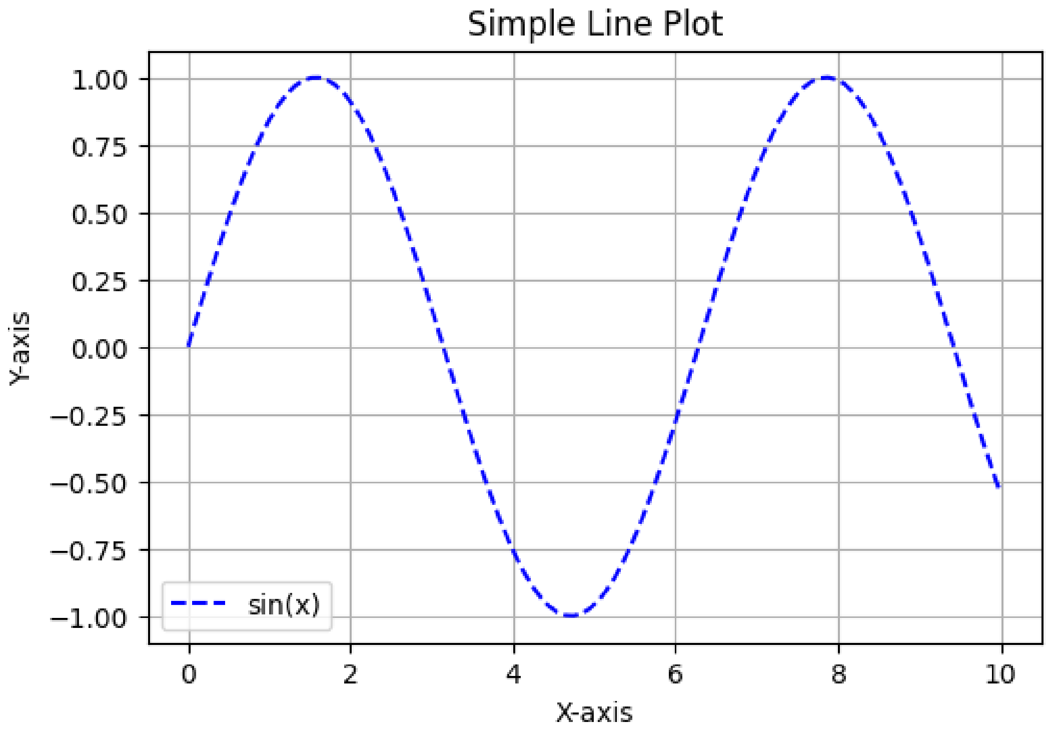


matplotlib

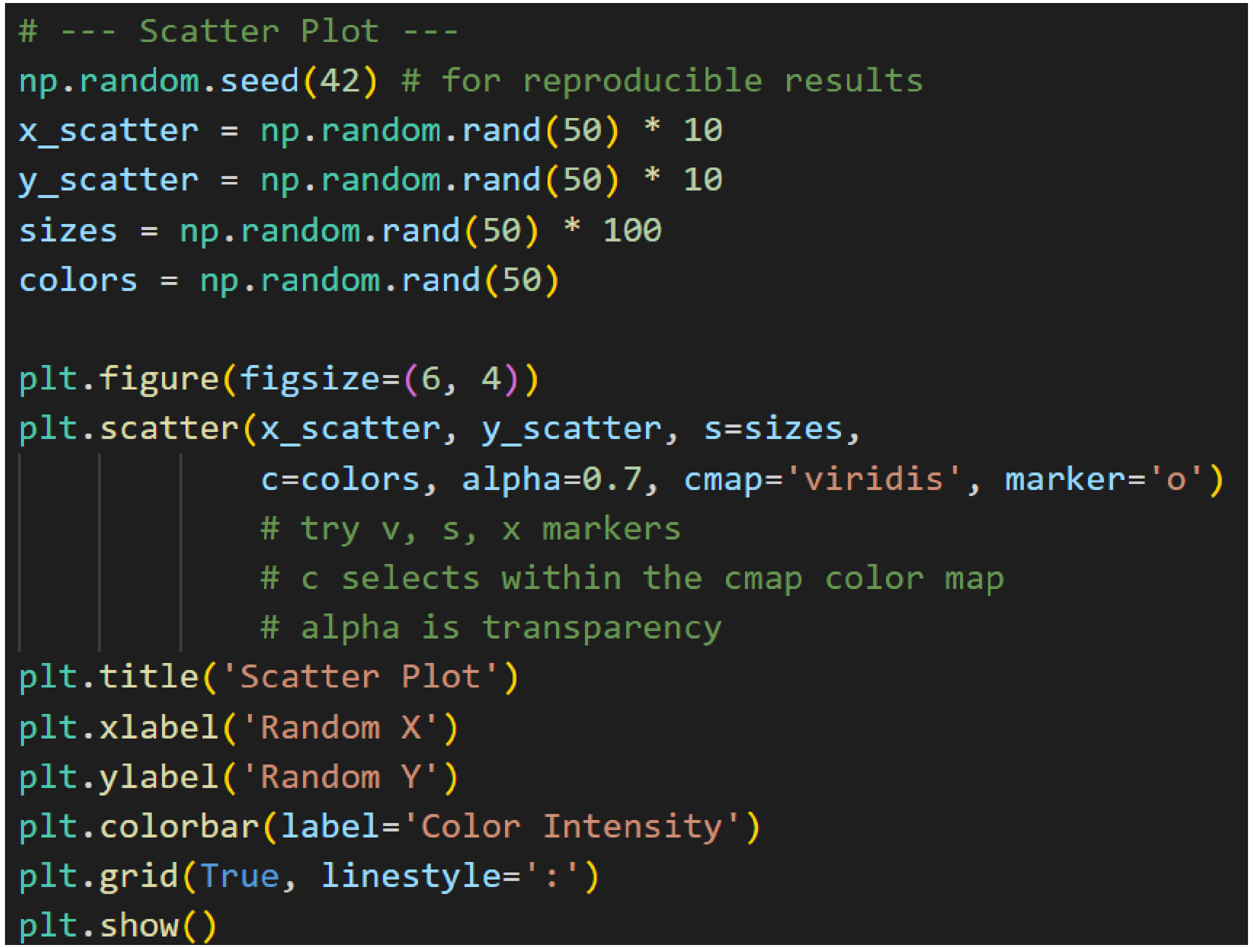
- Foundational data visualization library in python
- Comprehensive set of plot types
- Highly customizable
- Wide variety of output formats
- Seamless integration with a variety of frameworks such as numpy
- Underlying visualization engine used by many libraries such as pandas and seaborn
- Interfaces
    - State based – pyplot – similar to MATLAB
    - Object oriented - more powerful and flexible interface where you explicitly create and manipulate Figure and Axes objects



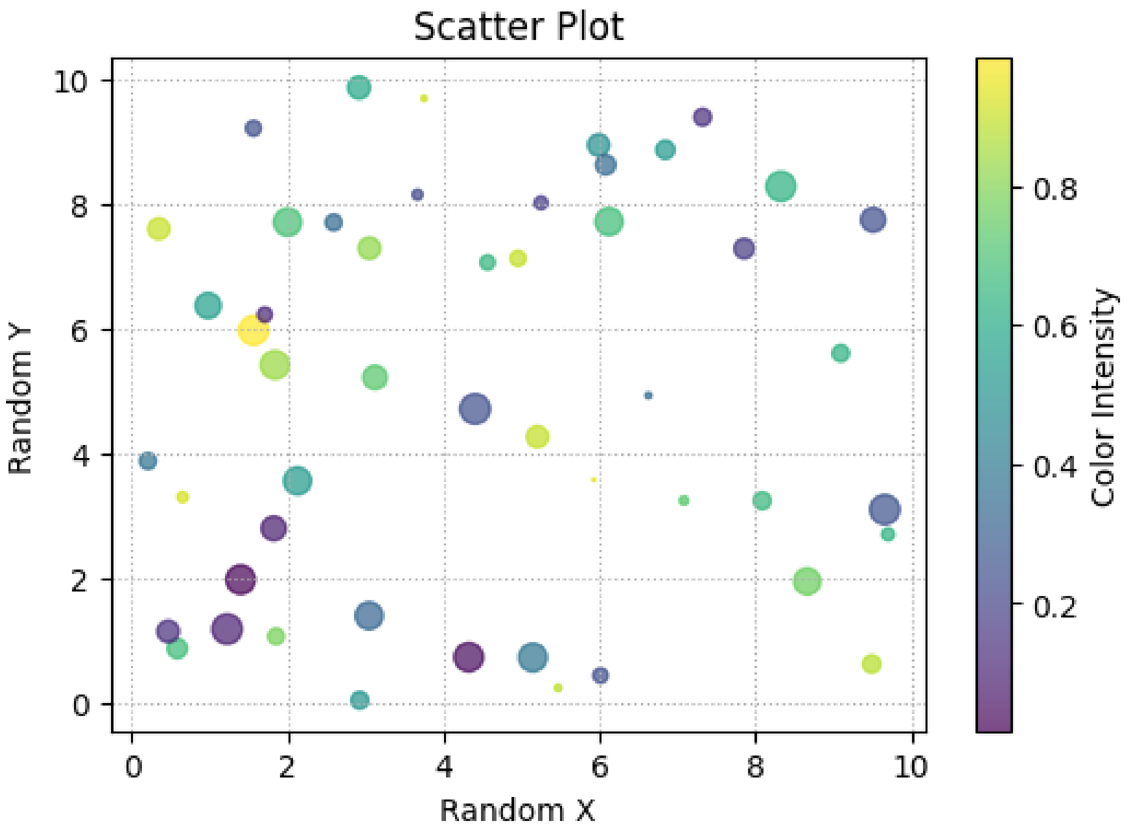
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# --- Simple Line Plot ---
x = np.linspace(0, 10, 100) # 100 points from 0 to 10
y = np.sin(x)

plt.figure(figsize=(6, 4)) # Optional: Specify figure size in inches
plt.plot(x, y, label='sin(x)', color='blue', linestyle='--')
plt.title('Simple Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.grid(True)
plt.show()
```



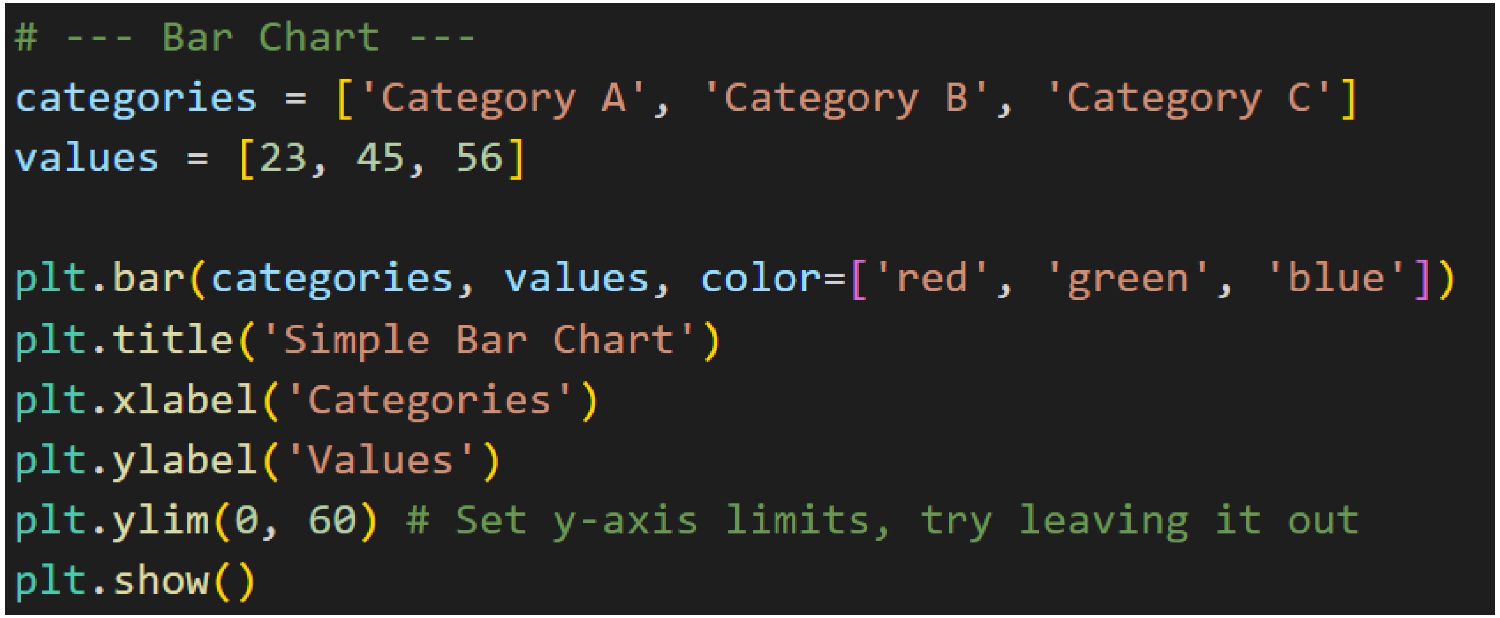
matplotlib examples



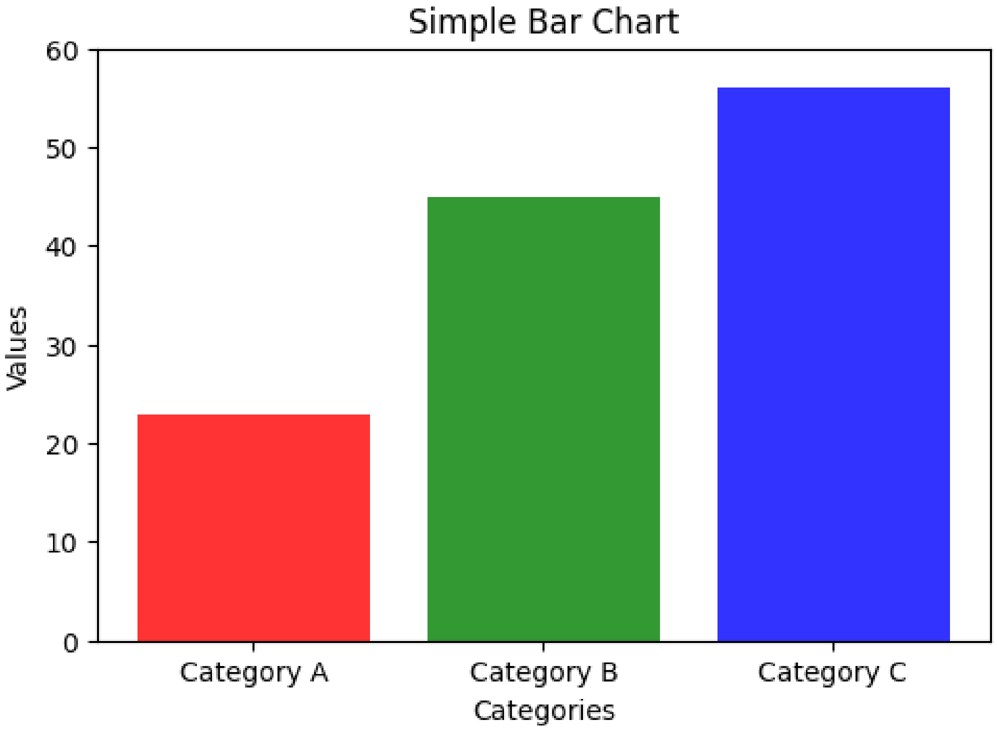
```
# --- Scatter Plot ---
np.random.seed(42) # for reproducible results
x_scatter = np.random.rand(50) * 10
y_scatter = np.random.rand(50) * 10
sizes = np.random.rand(50) * 100
colors = np.random.rand(50)

plt.figure(figsize=(6, 4))
plt.scatter(x_scatter, y_scatter, s=sizes,
            c=colors, alpha=0.7, cmap='viridis', marker='o')
            # try v, s, x markers
            # c selects within the cmap color map
            # alpha is transparency
plt.title('Scatter Plot')
plt.xlabel('Random X')
plt.ylabel('Random Y')
plt.colorbar(label='Color Intensity')
plt.grid(True, linestyle=':')
plt.show()
```



matplotlib examples

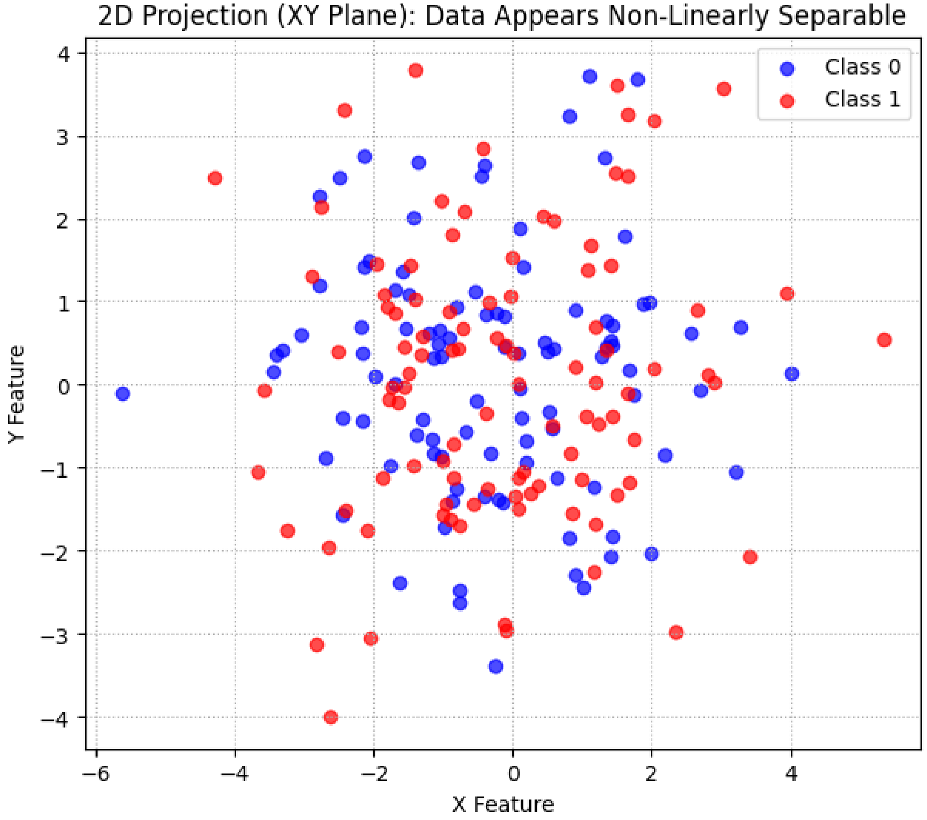


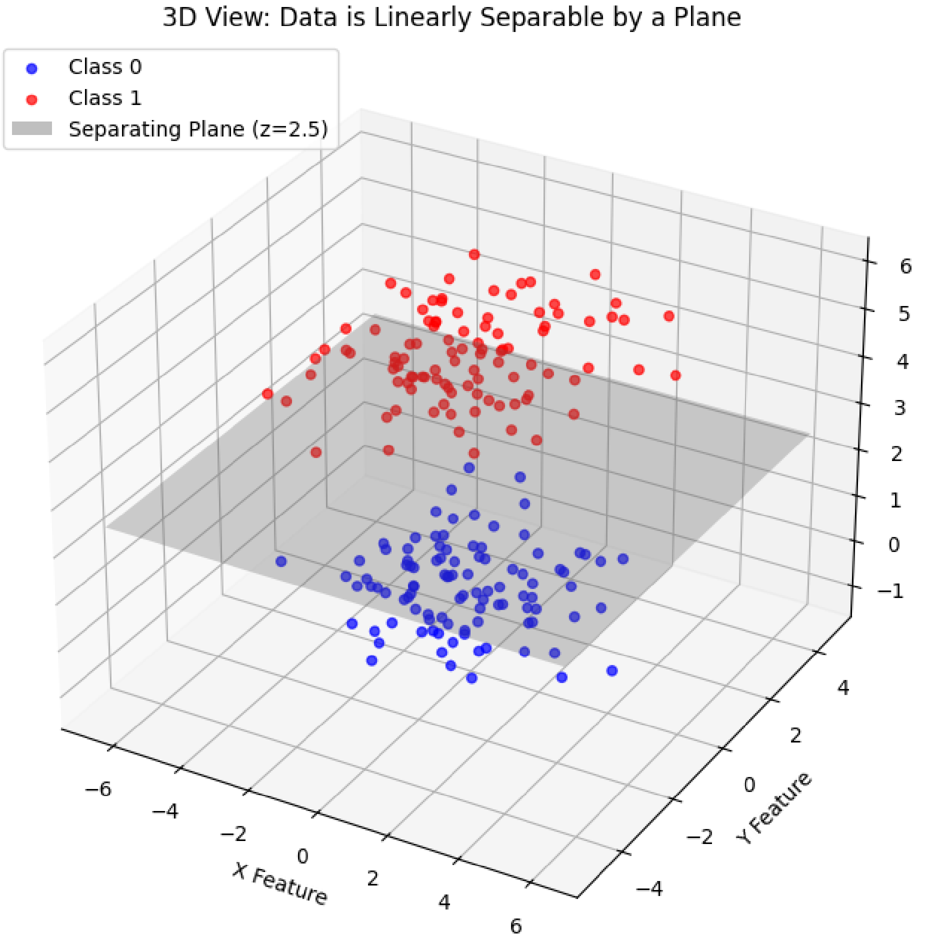
```python
# --- Bar Chart ---
categories = ['Category A', 'Category B', 'Category C']
values = [23, 45, 56]

plt.bar(categories, values, color=['red', 'green', 'blue'])
plt.title('Simple Bar Chart')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.ylim(0, 60) # Set y-axis limits, try leaving it out
plt.show()
```



matplotlib examples

- A case where a 3D plot shows linear separation of values
  - Code in this notebook
    - Also includes the Iris dataset

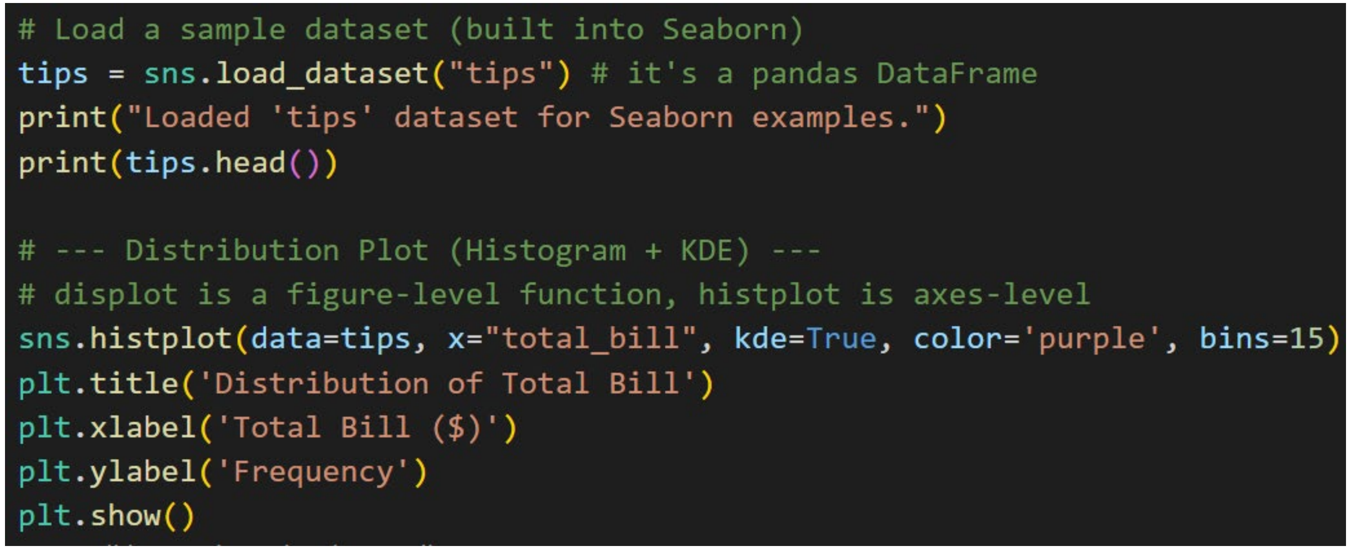




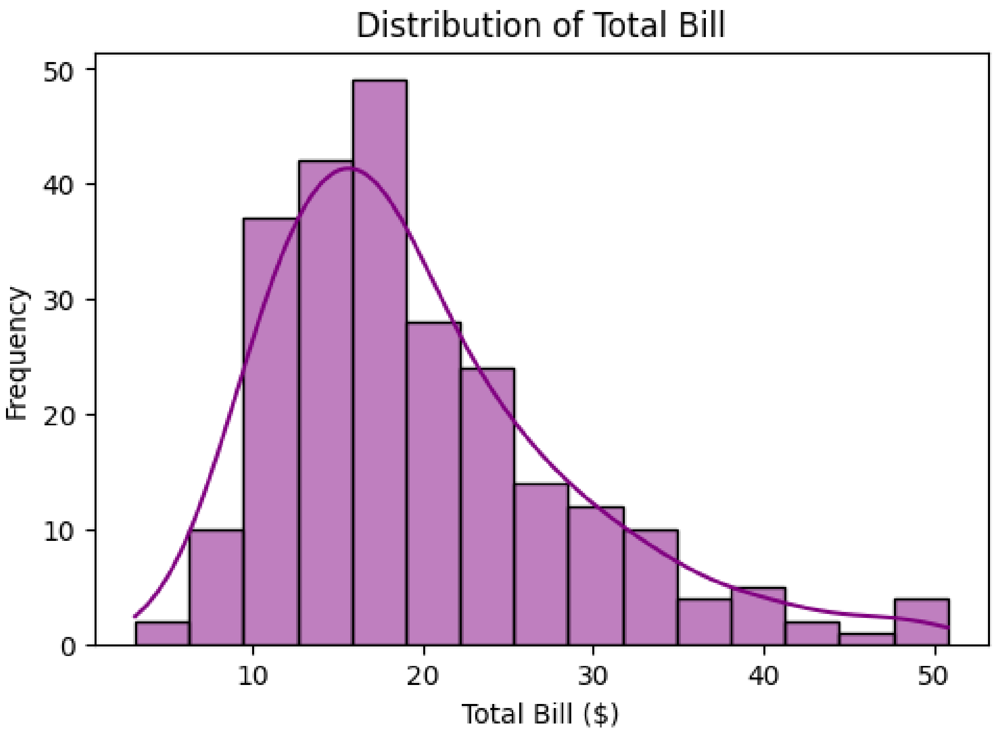
seaborn

- High level interface for drawing attractive statistical graphics
- Designed specifically for statistical data visualization
- Works directly with entire datasets, often pandas DataFrames
- Nice set of defaults
- Built on and complements matplotlib
- Specialized plots
  - comparing distributions (displot, histplot, kdeplot, ecdfplot)
  - visualizing relationships (relplot, scatterplot, lineplot)
  - plotting categorical data (catplot, boxplot, violinplot, stripplot, swarmplot)
  - creating multi-plot grids (FacetGrid, PairGrid, jointplot, pairplot)

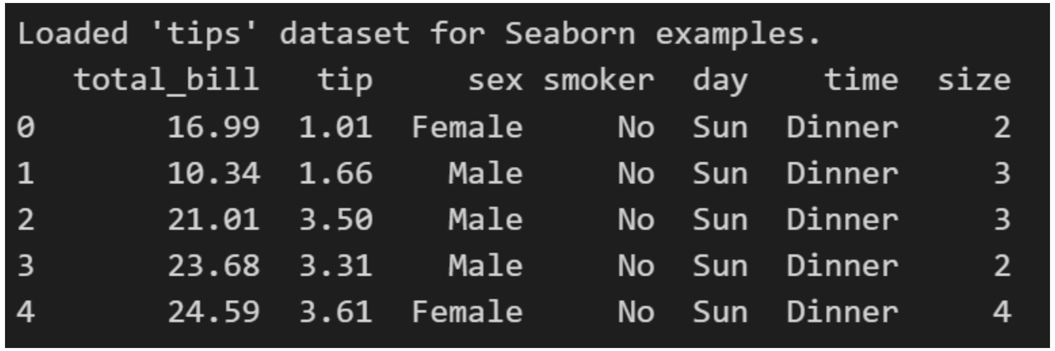
seaborn examples



```
# Load a sample dataset (built into Seaborn)
tips = sns.load_dataset("tips") # it's a pandas DataFrame
print("Loaded 'tips' dataset for Seaborn examples.")
print(tips.head())

# --- Distribution Plot (Histogram + KDE) ---
# displot is a figure-level function, histplot is axes-level
sns.histplot(data=tips, x="total_bill", kde=True, color='purple', bins=15)
plt.title('Distribution of Total Bill')
plt.xlabel('Total Bill ($)')
plt.ylabel('Frequency')
plt.show()
```

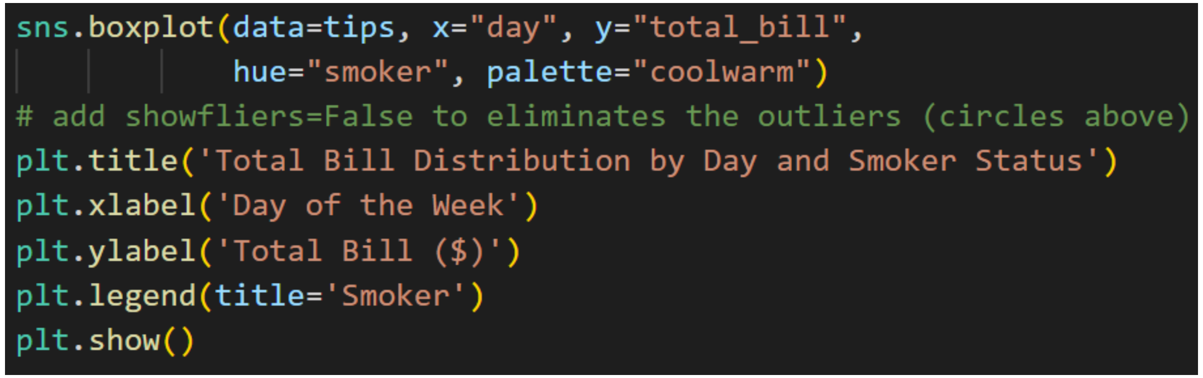


- KDE – Kernel Density Estimation – adds the curve
- Axes level - draws on single matplotlib axes
- Figure level - creates its own figure; can create multi-panel grids
- Frequency refers to the number of observations

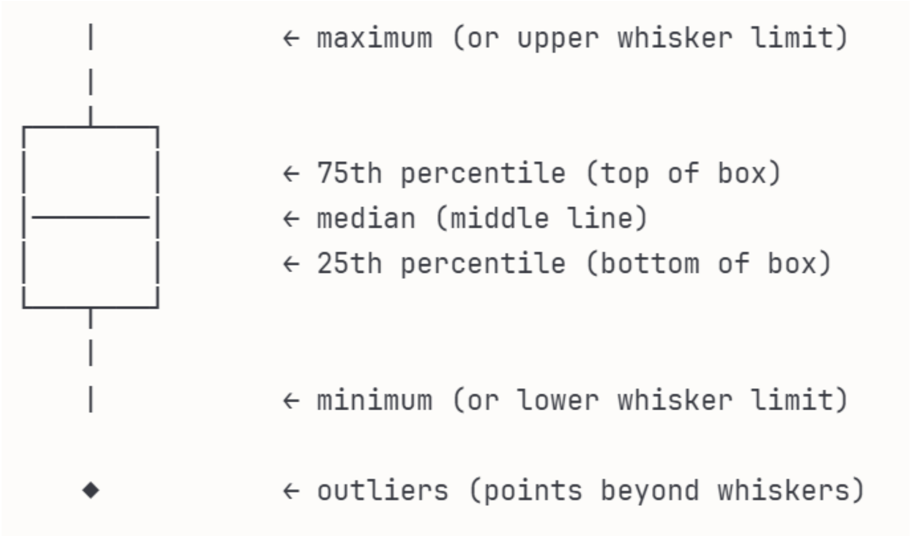


```
Loaded 'tips' dataset for Seaborn examples.
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

seaborn examples



```
sns.boxplot(data=tips, x="day", y="total_bill",
            hue="smoker", palette="coolwarm")
# add showfliers=False to eliminates the outliers (circles above)
plt.title('Total Bill Distribution by Day and Smoker Status')
plt.xlabel('Day of the Week')
plt.ylabel('Total Bill ($)')
plt.legend(title='Smoker')
plt.show()
```



← maximum (or upper whisker limit)

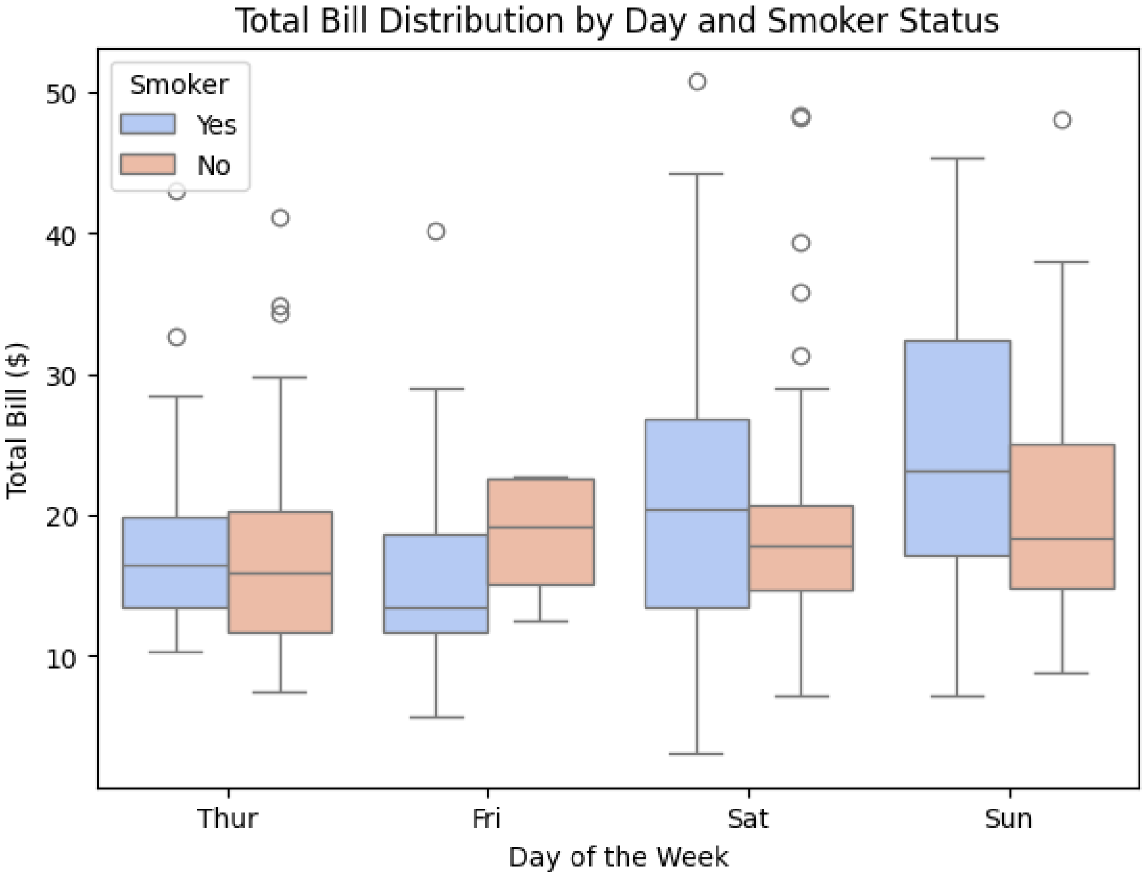
← 75th percentile (top of box)

← median (middle line)

← 25th percentile (bottom of box)

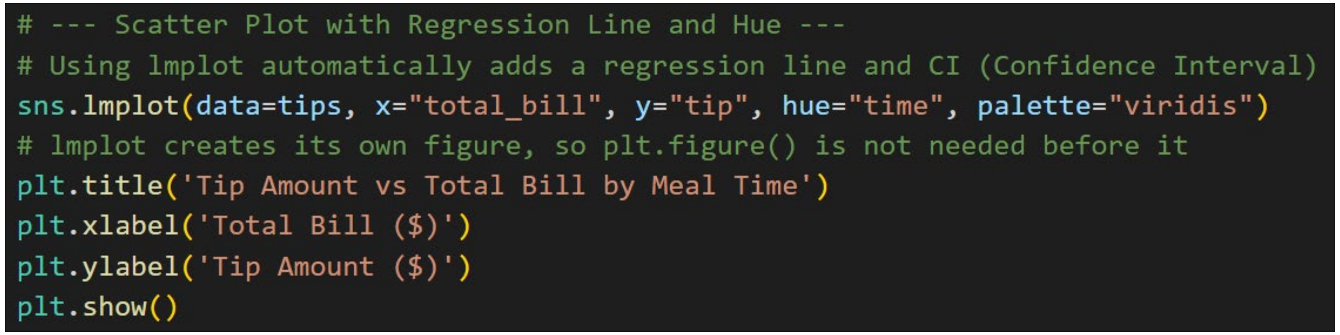
← minimum (or lower whisker limit)

← outliers (points beyond whiskers)

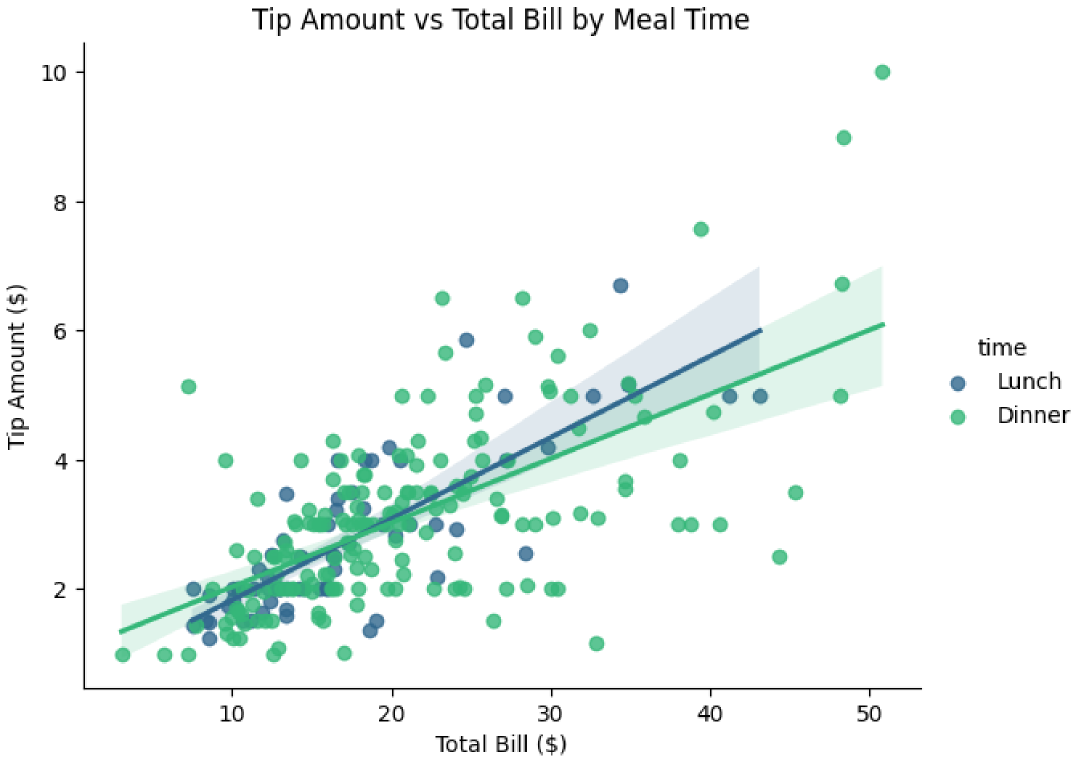


- Upper whisker: Q3 + 1.5 × IQR
- Lower whisker: Q1 − 1.5 × IQR
- Q1 = 25th percentile
- Q3 = 75th percentile
- IQR = Q3 − Q1 (interquartile range)

seaborn examples

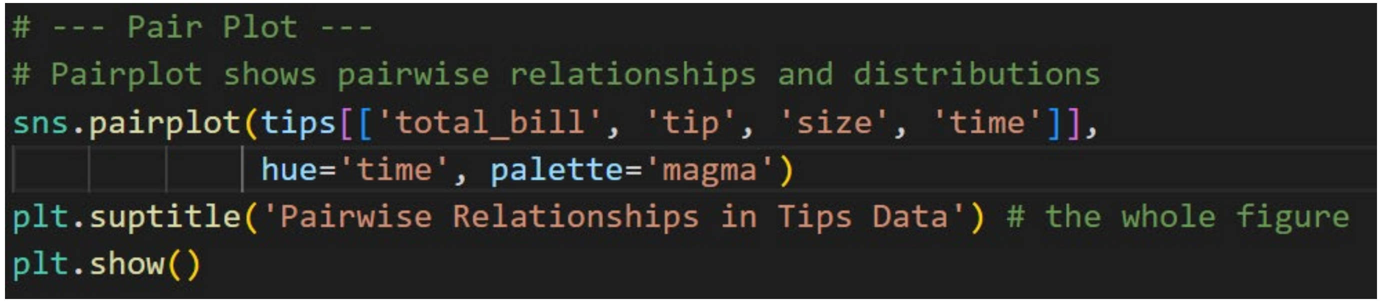


```
# --- Scatter Plot with Regression Line and Hue ---
# Using lmplot automatically adds a regression line and CI (Confidence Interval)
sns.lmplot(data=tips, x="total_bill", y="tip", hue="time", palette="viridis")
# lmplot creates its own figure, so plt.figure() is not needed before it
plt.title('Tip Amount vs Total Bill by Meal Time')
plt.xlabel('Total Bill ($)')
plt.ylabel('Tip Amount ($)')
plt.show()
```



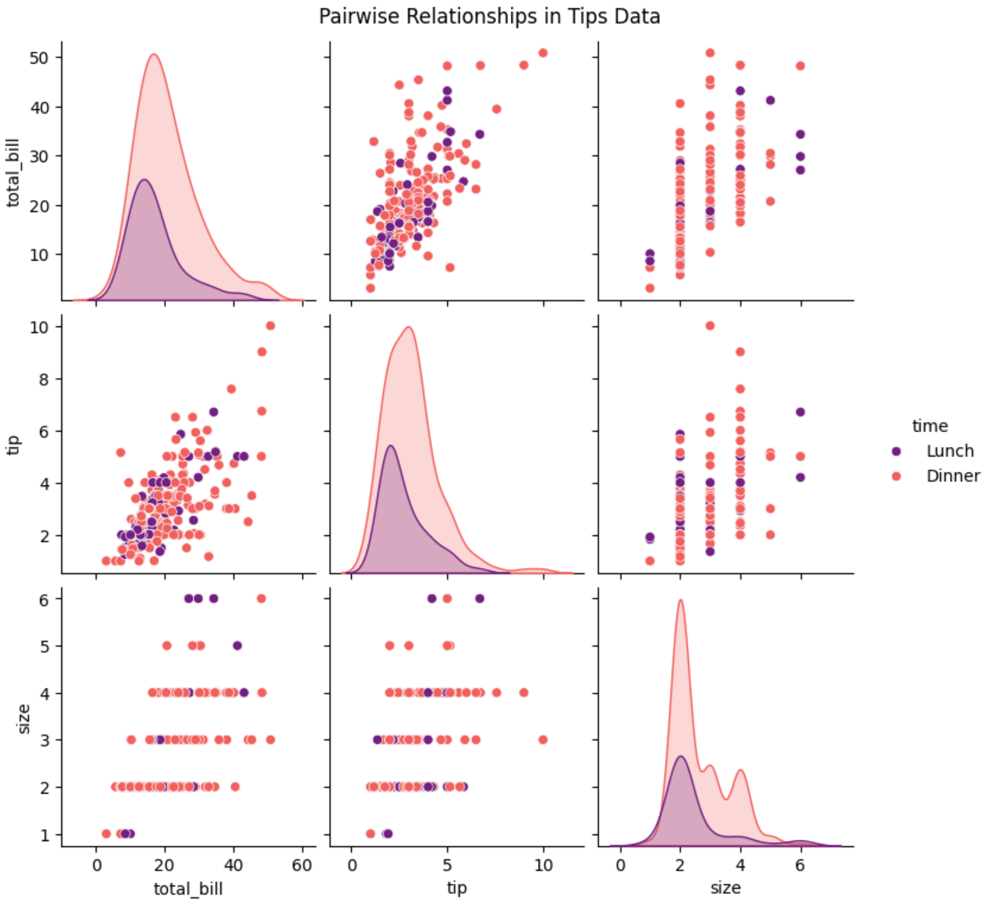
lmplot – Linear Model Plot
CI – Confidence interval, defaults to 95%
hue – Splits the data on meal time

seaborn examples



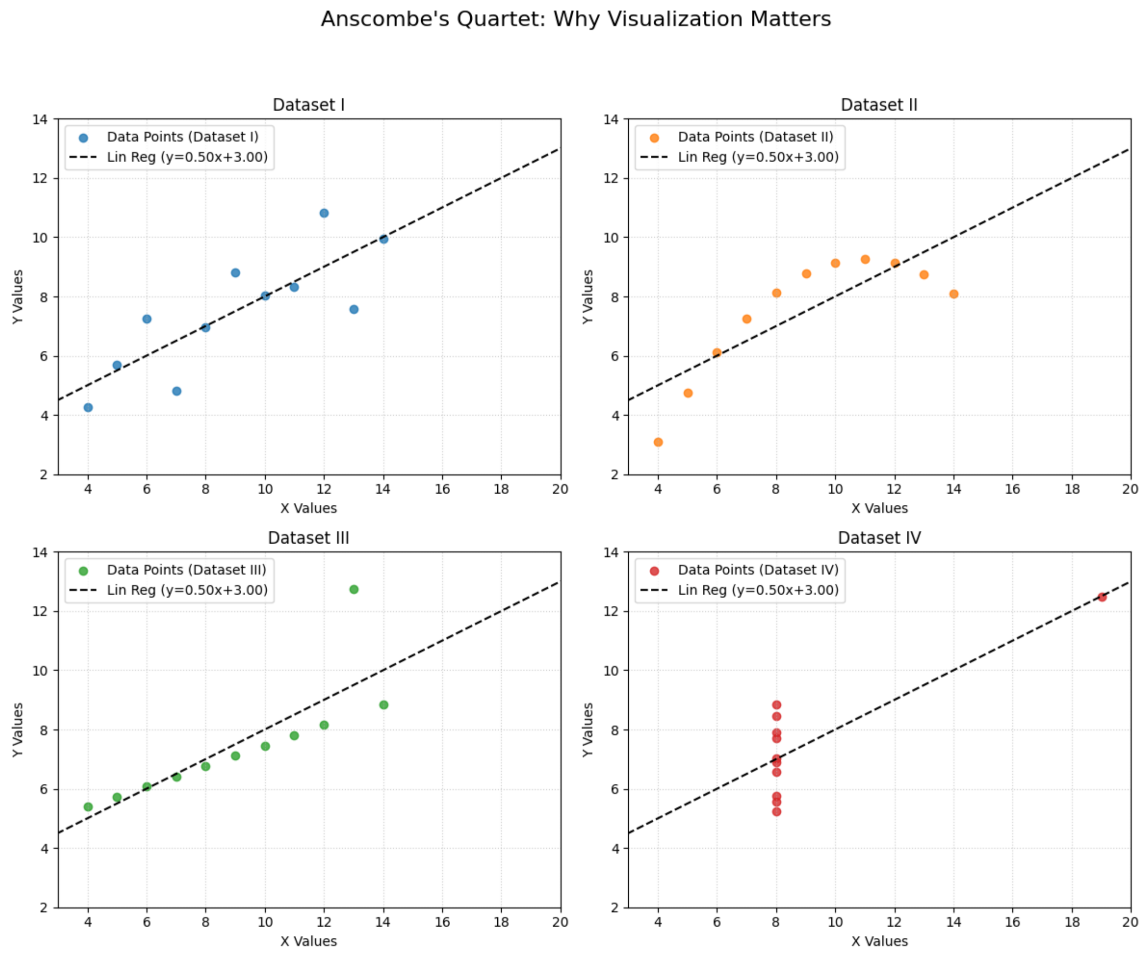
```
# --- Pair Plot ---
# Pairplot shows pairwise relationships and distributions
sns.pairplot(tips[['total_bill', 'tip', 'size', 'time']],
             hue='time', palette='magma')
plt.suptitle('Pairwise Relationships in Tips Data') # the whole figure
plt.show()
```

- The plots on the diagonal (same variable) show the shape of the probability density of that variable.
- The plots off the diagonal show pairwise scatter plots.
- The axes are swapped across the diagonal
- hue divides the data by mealtime
- size is the number of people dining



Case Study: Anscombe’s Quartet

- A classic dataset which is included in seaborn
- A Kaggle Notebook which illustrates the analysis and visualization



Q&A
and
Demo

Github gist with code examples:
CTD Python 100 Lesson 11 - Data Visualization