Pandas, Matplotlib, and Seaborn are fundamental Python libraries for data analysis and visualization.

1. Pandas:

Pandas is a library for data manipulation and analysis, offering powerful data structures like:
Series: A one-dimensional labeled array capable of holding any data type (integers, strings, floats, Python objects, etc.). Think of it as a single column in a spreadsheet.

```
import pandas as pd
s = pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])
print(s)
```

DataFrame: A two-dimensional labeled data structure with columns of potentially different types. It's like a spreadsheet or a SQL table.
Python

```
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'London', 'Paris']}
df = pd.DataFrame(data)
print(df)
```

Pandas provides extensive functionality for data cleaning, transformation, merging, filtering, and aggregation.

2. Matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It's the foundation upon which many other plotting libraries are built.
Pyplot: A Matplotlib module that provides a MATLAB-like interface for plotting.

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

x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Sine Wave')
```

```
plt.show()
```

Matplotlib allows for fine-grained control over every aspect of a plot, including labels, titles, colors, line styles, and more.

3. Seaborn:
Seaborn is a high-level data visualization library built on top of Matplotlib. It provides a more convenient and aesthetically pleasing way to create a wide range of statistical graphics, especially when working with Pandas DataFrames.
Simplified Syntax: Seaborn often requires less code than Matplotlib for common statistical plots.
Attractive Defaults: It offers appealing default styles and color palettes.
Statistical Plot Types: It excels at creating plots like heatmaps, violin plots, pair plots, and regression plots.
Python

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

data = {'Age': [25, 30, 35, 40, 45],
        'Salary': [50000, 60000, 70000, 80000, 90000]}
df = pd.DataFrame(data)

sns.scatterplot(x='Age', y='Salary', data=df)
plt.title('Age vs Salary')
plt.show()
```

Seaborn leverages Matplotlib for rendering but provides a higher-level API for statistical visualization, making it a popular choice for exploring and presenting data.