

## **User Scenario 3: Experienced Data Scientist**

# Contents

- Overview of Matplotlib and Line Charts.....3**
  - Matplotlib Overview.....3
  - Line Chart Type.....3
- Advanced Customization for Charts.....4**
  - Customize the Line Chart.....4
- Verification Commands.....5**
  - Python Verification Command..... 5

# Overview of Matplotlib and Line Charts

---

## Matplotlib Overview

---

Matplotlib is a Python library widely used for creating data visualizations, such as line charts, bar charts, and scatter plots. This overview introduces Matplotlib and describes the requirements for installing and verifying it on your system.

### Prerequisites

Before installing Matplotlib, ensure that Python and pip, Python's package installer, are installed on your computer. These are necessary for managing and installing Python libraries like Matplotlib.

### Installation

Matplotlib can be installed using pip, which simplifies the process of adding libraries to your Python environment. Running the appropriate installation command in your terminal will complete the setup.

### Verification

To verify that Matplotlib has been installed correctly, you can check the library's version from your Python IDE. Displaying the version helps confirm the installation and the library's compatibility with your Python environment.

## Line Chart Type

---

A line chart is a type of data visualization that displays data points connected by straight line segments, useful for showing trends over time or the relationship between two variables. This concept provides an overview of creating a line chart using Python and the Matplotlib library.

### Importing Matplotlib

To create a line chart, first import the `matplotlib.pyplot` library into your Python environment. This module provides functions specifically for plotting data.

### Preparing Data for the Line Chart

Data points for a line chart are typically represented as two lists: one for the x-axis values and one for the y-axis values. Each list represents a series of values that will be connected on the chart, visually displaying the data's trend.

### Creating and Displaying the Line Chart

Using the `plot()` function in Matplotlib, a line chart can be generated by plotting the prepared x and y values. To display the plot, the `plt.show()` function is used, allowing the figure to render on your screen.

### Example Output

When the code is executed, the final plot will display as a visual representation of the data. The line chart provides a straightforward way to analyze patterns or changes across the dataset.

# Advanced Customization for Charts

---

## Customize the Line Chart

---

This task demonstrates how to customize a line chart by adding titles, axis labels, and other customizations using Python and Matplotlib.

1. Import Matplotlib into Python

To start, import the ``matplotlib.pyplot`` library to access plotting functionality.

2. Define Your Data

Define your data by creating the x and y values. These represent the points to be plotted on the graph.

3. Create the Line Plot

Use the ``plt.plot()`` function to create the line plot with your x and y values.

4. Add a Title to the Plot

Add a title to the plot using ``plt.title()`` to describe what the chart represents.

5. Add Axis Labels

Label the x and y axes for clarity using ``plt.xlabel()`` and ``plt.ylabel()``.

6. Add Grid Lines

Make the plot easier to read by adding grid lines with ``plt.grid()``.

7. Display the Plot

Use ``plt.show()`` to display the customized plot on your screen.

8. Final Customized Line Chart

After running all the steps, you should now have a customized line chart with a title, axis labels, grid lines, and a styled line.

# Verification Commands

---

## Python Verification Command

---

### Verification Commands for Python, pip, and Matplotlib

The following table lists common verification commands for checking the installation of Python, pip, and Matplotlib on your system. These commands ensure that required components are properly installed and available.

Component	Command	Description
Python	<pre>python3 --version</pre>	Checks if Python is installed and displays the installed version. If Python is not installed, an error message will appear.
pip	<pre>pip --version</pre>	Verifies that the pip package manager is installed and displays its version. If pip is not installed, an error message will appear.
Matplotlib	<pre>import matplotlibprint(matplotlib.__version__)</pre>	Imports Matplotlib and displays the installed version of the library. If Matplotlib is not installed, an import error will be raised.