

# **MATPLOTLIB BAR CHART GUIDE**

# Contents

- 1 Introduction to Matplotlib.....3**
  - 1.1 Matplotlib Overview.....3
- 2 Creating a Bar Chart.....5**
  - 2.1 Install Matplotlib.....5
  - 2.2 Create a Bar Chart.....6
- 3 Verification Commands.....9**
  - 3.1 Python Verification Command.....9

# 1 Introduction to Matplotlib

---

## 1.1 Matplotlib Overview

---

Matplotlib is a Python library for creating a variety of visualizations, including line charts, bar charts, scatter plots, and more. It is a fundamental tool for data visualization in Python.

Matplotlib is one of the most widely used Python libraries for creating static, interactive, and animated visualizations. Its versatility and ease of use make it a popular choice for developers, data scientists, and researchers working with data visualization tasks.

### Core Features of Matplotlib:

- Supports a wide range of chart types, including line charts, bar charts, scatter plots, histograms, and pie charts.
- Allows for highly customizable plots with options to adjust colors, fonts, line styles, grid lines, and annotations.
- Integrates seamlessly with other Python libraries, such as NumPy and Pandas, for streamlined data analysis and visualization workflows.
- Offers an object-oriented API for advanced users who need fine-grained control over their visualizations.

### Applications of Matplotlib:

Matplotlib is widely used in various fields, such as:

- **Data Science:** Visualizing large datasets to uncover trends and patterns.
- **Scientific Research:** Plotting experimental results and analyzing data distributions.
- **Finance:** Creating stock price charts and financial performance visualizations.

- **Education:** Teaching fundamental concepts of data visualization and Python programming.

### Why Choose Matplotlib?

Matplotlib's popularity stems from its balance of simplicity and flexibility. Beginners can quickly create visualizations with high-level commands like `plt.plot()`, while advanced users can build complex, publication-quality visualizations using the library's detailed customization features.

Whether you're building quick exploratory plots or designing detailed reports, Matplotlib provides the tools you need to effectively communicate insights from your data.

## 2 Creating a Bar Chart

---

### 2.1 Install Matplotlib

---

Install the matplotlib library in Python, which is necessary for creating data visualizations.

#### 1. Verify Dependencies

Before installing matplotlib, ensure that Python and pip are installed on your system. Run `python3 --version` to check Python and `pip --version` to check if pip is installed.

```
# Check Python version
python3 --version

# Check pip version
pip --version
```

#### 2. If either Python or pip is missing

If either Python or pip is missing, follow these steps to install them: Run `python3 -m ensurepip` to install pip and download Python from the official website if necessary.

```
# Install pip if missing
python3 -m ensurepip
```

#### 3. Install Matplotlib

Once Python and pip are installed, install Matplotlib with the following command: `pip install matplotlib`.

```
# Install Matplotlib
```

```
pip install matplotlib
```

### 4. Verify Installation

After installation, verify that Matplotlib is installed correctly by running the following Python code in your Python IDE: `import matplotlib.pyplot as plt` followed by `print(matplotlib.__version__)`.

```
# Verify Matplotlib installation
import matplotlib.pyplot as plt
print(matplotlib.__version__)
```

## 2.2 Create a Bar Chart

---

This task demonstrates how to create a bar chart using Python and Matplotlib.

### 1. Import matplotlib.pyplot into Python

This command imports the necessary library for creating plots in Python.

```
import matplotlib.pyplot as plt
```

### 2. Define Your Data

This step defines the categories and values for the bar chart.

```
categories = ['A', 'B', 'C', 'D', 'E']
values = [10, 20, 15, 25, 30]
```

### 3. Create the Bar Plot using the `plt.bar()` function

```
plt.bar(categories, values)
```

The bar chart is now created ...

4. Add a Title to the Plot using the `plt.title()` function

```
plt.title('Bar Chart Example')
```

A title is added to the bar chart.

5. Add Axis Labels using the `plt.xlabel()` and `plt.ylabel()` functions

```
plt.xlabel('Categories')  
plt.ylabel('Values')
```

Labels for the X and Y axes are added to the chart.

6. Customize the bar color to Blue, using `color` in `plt.bar()`

```
plt.bar(categories, values, color='blue')
```

The color of the bars in the chart is changed to blue.

7. Display the Plot using `plt.show()` again

```
plt.show()
```

The final chart with the blue bars is displayed again.

### 8. Final Bar Chart

```
# Final complete code
import matplotlib.pyplot as plt

categories = ['A', 'B', 'C', 'D', 'E']
values = [10, 20, 15, 25, 30]

plt.bar(categories, values, color='blue')
plt.title('Bar Chart Example')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.show()
```

The final version of the bar chart is displayed below.



# 3 Verification Commands

## 3.1 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     | <code>python3 --version</code>                                | Checks if Python is installed and displays the installed version. If Python is not installed, an error message will appear.           |
| pip        | <code>pip --version</code>                                    | Verifies that the pip package manager is installed and displays its version. If pip is not installed, an error message will appear.   |
| Matplotlib | <code>import matplotlib; print(matplotlib.__version__)</code> | Imports Matplotlib and displays the installed version of the library. If Matplotlib is not installed, an import error will be raised. |