

einführung-in-matplotlib

November 13, 2024

1 Einführung in Matplotlib

1.0.1 1. Was ist Matplotlib ?

Matplotlib ist eine weit verbreitete [Python-Bibliothek](#) zur Visualisierung von Daten. Es ermöglicht Benutzern, ansprechende und leicht interpretierbare Diagramme zu erstellen, die dabei helfen, komplexe Daten besser zu verstehen. [Matplotlib](#) ist besonders bei Wissenschaftlern, Ingenieuren und Datenanalysten beliebt, da es eine einfache Möglichkeit bietet, Daten zu visualisieren.

1.0.2 2. Vorteile von Matplotlib

Vorteil	Beschreibung
Einfach zu erlernen	Matplotlib hat eine intuitive Syntax, die besonders für Anfänger leicht verständlich ist.
Vielseitigkeit	Von einfachen Linienplots bis hin zu komplexen 3D-Diagrammen unterstützt Matplotlib eine Vielzahl von Diagrammtypen.
Anpassbarkeit	Benutzer können fast jedes Detail eines Diagramms anpassen, von Farben über Achsenbeschriftungen bis hin zu Linienbreiten.
Kompatibilität	Matplotlib funktioniert hervorragend mit anderen Python-Bibliotheken wie NumPy , Pandas und Seaborn , was die Datenvisualisierung für komplexe Analysen erleichtert.
Vielseitiges Ausgabeformat	Plots können in verschiedenen Formaten gespeichert werden, was nützlich für Publikationen oder Präsentationen ist.

1.0.3 3. Installation und erste Schritte mit Matplotlib

3.1 Installation in Jupyter Notebook:

in einem [Jupyter Notebook](#) kann Matplotlib mit folgendem Befehl installiert werden: `!pip install matplotlib`

3.2 Installation in Python:

Matplotlib kann in [Python](#) mit folgendem Befehl installiert werden: `pip3 install matplotlib`

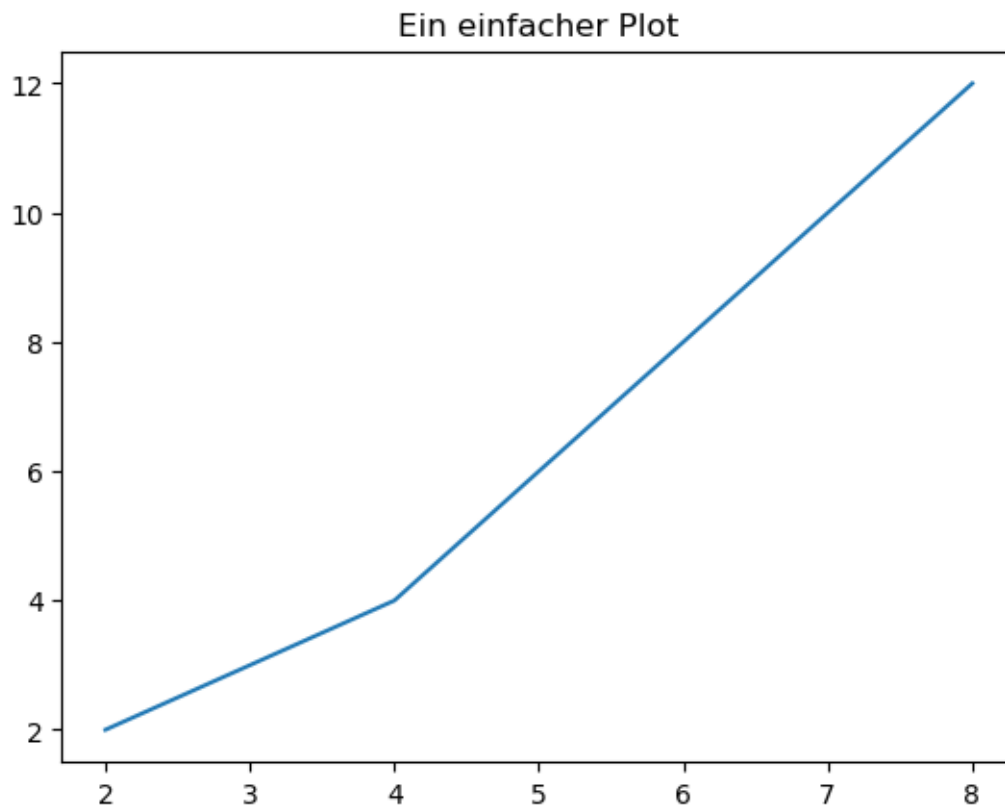
Bevor Sie den `pip` Befehl nutzen, stellen Sie sicher, dass Sie `pip` mit folgendem Befehl `sudo apt install python3-pip` installiert haben.

3.3 erster Schritt in Matplotlib:

Es ist ein einfaches Beispiel für Darstellung eines Plots in Matplotlib:

```
[44]: ### Importieren Sie die Bibliotheken
import matplotlib.pyplot as plt

# Einfache Linie zeichnen
plt.plot([2, 4, 6, 8], [2, 4, 8, 12])
plt.title("Ein einfacher Plot")
plt.show()
```



1.0.4 3.4 Erklärung des Codes

Code	Beschreibung
<code>import matplotlib.pyplot as plt</code>	Importiert die pyplot -Funktionen von Matplotlib, um einfache Plot-Befehle zu verwenden.
<code>plt.plot()</code>	Erstellt einen Linienplot basierend auf den angegebenen x- und y-Daten.
<code>plt.title()</code>	Fügt einen Titel zum Diagramm hinzu.

Übung

finden Sie heraus, was `plt.show()` im Code macht !

1.0.5 4.Funktionen von Matplotlib

Die typischen Funktionen von Matplotlib können wie folgt gelistet werden:

Funktion	Beschreibung
Vielfältige Diagrammtypen	Matplotlib unterstützt eine Vielzahl von Diagrammtypen, darunter Linien-, Balken-, Streu- und Kreisdiagramme sowie 3D-Visualisierungen.
Anpassung	Die meisten Aspekte eines Diagramms können detailliert angepasst werden (z.B. Titel, Achsen, Farben).
Mehrere Plots	Mit <code>subplots()</code> können mehrere Diagramme auf einer Abbildung erstellt werden, was es einfach macht, verschiedene Visualisierungen nebeneinander darzustellen.
Interaktive Funktionen	In Kombination mit Jupyter Notebooks bietet Matplotlib interaktive Diagramme, bei denen Nutzer zoomen und Datenpunkte hervorheben können.

1.0.6 6. Arten von Plots in Matplotlib

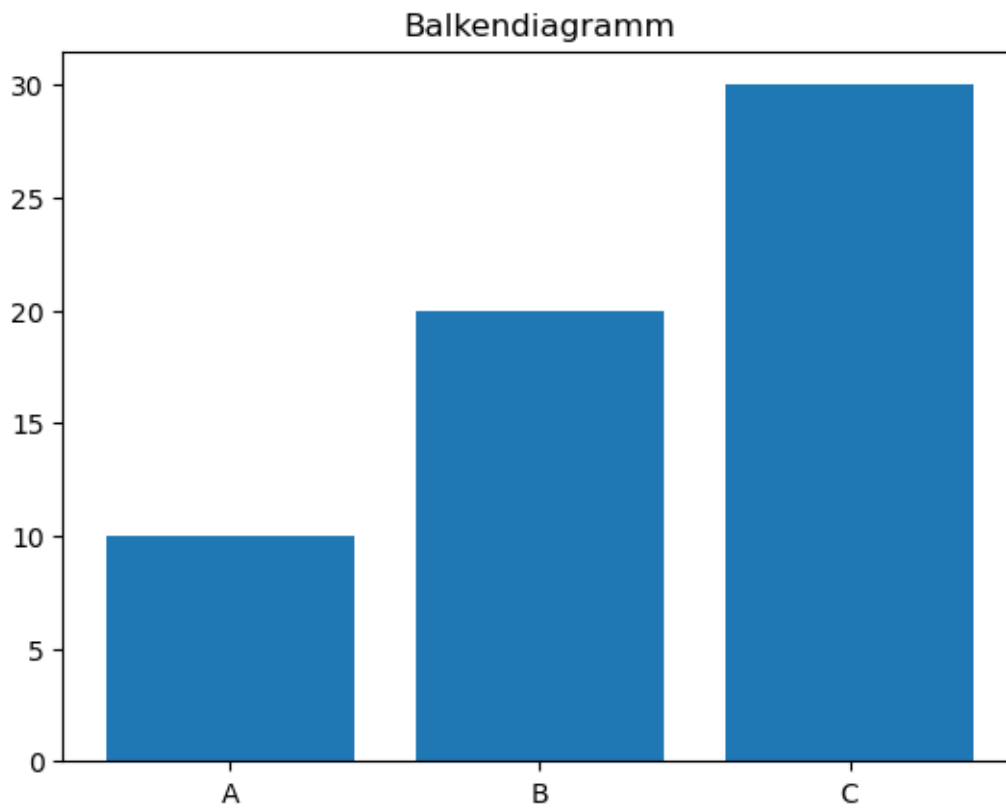
Plot-Typ	Beschreibung	Funktion
Linienplot	für zeitliche Trends und kontinuierliche Daten.	<code>plot()</code>
Streudiagramm	für die Visualisierung von Datenpunkten und deren Verteilung.	<code>scatter()</code>
Balkendiagramm	für den Vergleich von kategorischen Daten.	<code>bar()</code>
Histogramm	für die Darstellung von Häufigkeitsverteilungen.	<code>hist()</code>
Kuchendiagramm	für prozentuale Anteile an einem Ganzen.	<code>pie()</code>
Boxplot	für die Darstellung von Verteilungen und Ausreißern.	<code>boxplot()</code>
Flächendiagramm	für die Darstellung kumulativer Werte.	<code>fill_between()</code>
Dichteplot	für die Visualisierung der Wahrscheinlichkeitsdichte.	<code>hist()</code> mit der Option <code>density=True</code>
Heatmap	für die Visualisierung von Matrizen und Korrelationen.	<code>imshow()</code>
3D-Plot	für dreidimensionale Visualisierungen.	<code>plot_surface()</code> aus dem Modul <code>mpl_toolkits.mplot3d</code>

1.0.7 6.1 Beispiele zu Plots

hier werden einpaar Beispiele zu manchen der Plots gezeigt.

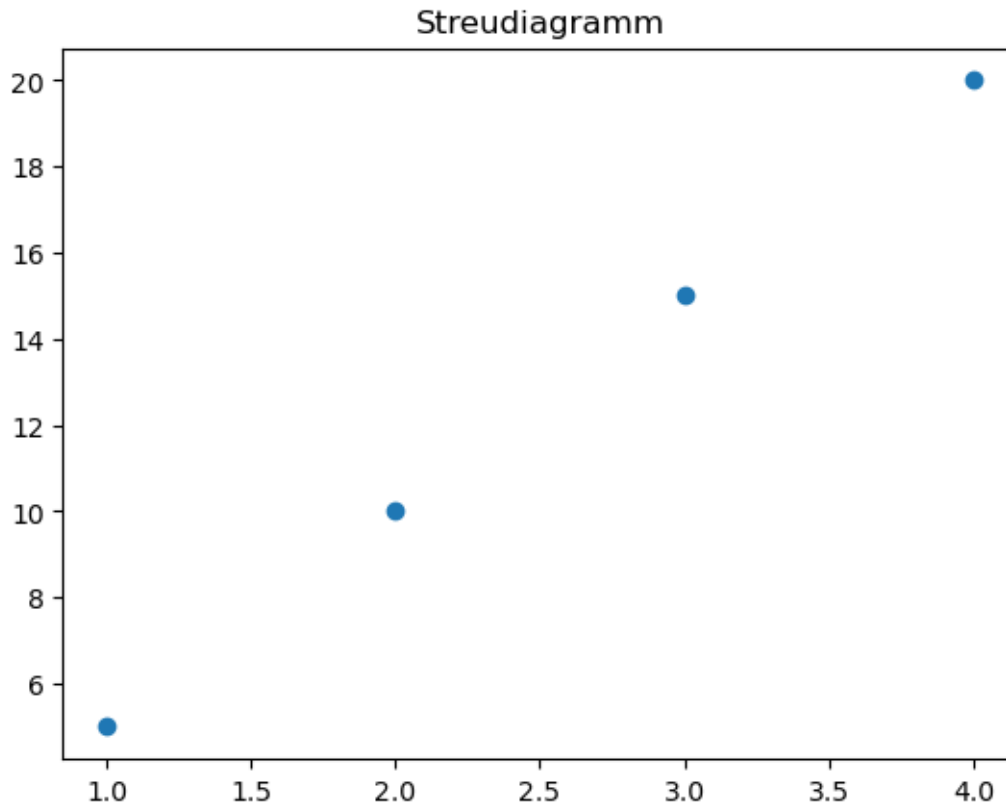
6.1.1 Beispiel zum Balkendiagramm

```
[82]: plt.bar(['A', 'B', 'C'], [10, 20, 30])  
plt.title("Balkendiagramm")  
plt.show()
```



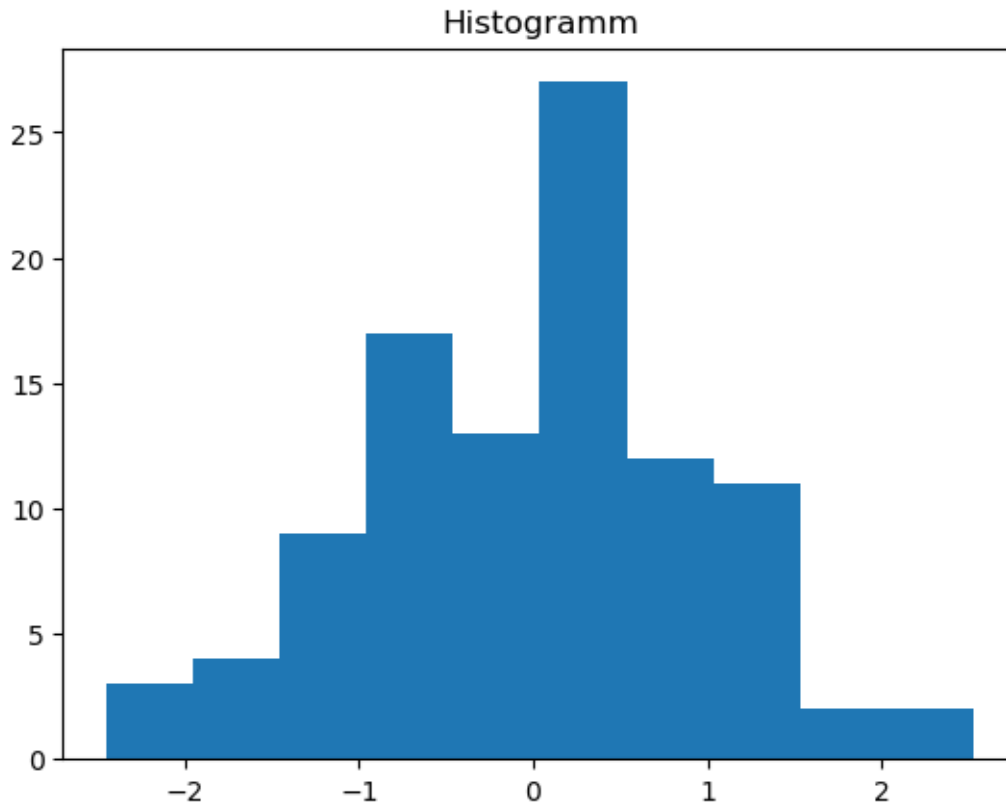
6.1.2 Beispiel zum Streudiagramm

```
[84]: plt.scatter([1, 2, 3, 4], [5, 10, 15, 20])  
plt.title("Streudiagramm")  
plt.show()
```



6.1.3 Beispiel zum Histodiagramm

```
[80]: import numpy as np
data = np.random.randn(100)
plt.hist(data, bins=10)
plt.title("Histogramm")
plt.show()
```



Aufgabe

versuchen Sie, Daten in 3 Diagrammarten Ihrer Wahl darzustellen !

1.0.8 7. Matplotlib im Vergleich zu anderen Bibliotheken

Seaborn: Baut auf Matplotlib auf, bietet jedoch eine einfachere API und speziellere Diagramme für statistische Daten. Nutzen Seaborn, wenn Sie besonders ästhetische und komplexe statistische Plots erstellen möchten.

Plotly: Interaktive Diagramme und Dashboards. Es ist ideal, wenn Sie Diagramme benötigen, die durch Benutzerinteraktionen dynamisch werden (z.B. in Webanwendungen).

1.0.9 8. häufige Probleme und deren Lösungen

Beim Erstellen von Diagrammen können Sie möglicherweise folgende Fehler sehen:

Problem	Lösung
Kein Plot wird angezeigt	Fügen Sie <code>%matplotlib inline</code> in Jupyter Notebooks hinzu, um Diagramme korrekt anzuzeigen.
Diagramme sind zu klein oder unleserlich	Verwenden Sie <code>figsize</code> in <code>plt.figure()</code> , um die Größe der Abbildung anzupassen.

```
[ ]: ### Beispiel, um Diagramme groß und leserlich zu machen  
plt.figure(figsize=(10, 6))
```

Übung

versuchen Sie, den Befehl auf erstem Diagramm des Notebooks zu implementieren !

1.0.10 9. Zusammenfassung

Matplotlib ist eine äußerst flexible Bibliothek für die Erstellung statischer, interaktiver und 3D-Diagramme in Python. Mit einer leicht verständlichen Syntax und vielen Anpassungsoptionen ist sie ideal für Datenanalyse, Berichte und wissenschaftliche Visualisierungen. Durch die Kombination mit anderen Bibliotheken wie NumPy oder Pandas wird die Visualisierung großer Datenmengen zum Kinderspiel.