

The Big Book

Lukas Arnold	Simone Arnold	Florian Bagemihl
Matthias Baitsch	Marc Fehr	Maik Poetzsch
	Sebastian Seipel	

2024-12-03

Table of contents

Preface	3
I w-python-numpy-grundlagen	4
Preamble	5
Intro	6
Voraussetzungen	6
Verwendete Pakete und Datensätze	6
Pakete	6
Datensätze	6
Bearbeitungszeit	6
Lernziele	6
1 Einführung NumPy	8
1.1 Vorteile & Nachteile	8
1.2 Einbinden des Pakets	9
1.3 Referenzen	9

Preface

This is a Quarto book.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

Part I

w-python-numpy-grundlagen

Preamble



Bausteine Computergestützter Datenanalyse. “Numpy Grundlagen” von Lukas Arnold, Simone Arnold, Florian Bagemihl, Matthias Baitsch, Marc Fehr, Maik Poetzsch und Sebastian Seipel ist lizenziert unter [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/). Das Werk ist abrufbar unter <https://github.com/bausteine-der-datenanalyse/w-python-numpy-grundlagen>. Ausgenommen von der Lizenz sind alle Logos und anders gekennzeichneten Inhalte. 2024

Zitiervorschlag

Arnold, Lukas, Simone Arnold, Matthias Baitsch, Marc Fehr, Maik Poetzsch, und Sebastian Seipel. 2024. „Bausteine Computergestützter Datenanalyse. Werkzeugbaustein NumPy“. <https://github.com/bausteine-der-datenanalyse/w-python-numpy-grundlagen>.

BibTeX-Vorlage

```
@misc{BCD-Styleguide-2024,  
  title={Bausteine Computergestützter Datenanalyse. Werkzeugbaustein NumPy},  
  author={Arnold, Lukas and Arnold, Simone and Baitsch, Matthias and Fehr, Marc and Poetzsch, Maik},  
  year={2024},  
  url={https://github.com/bausteine-der-datenanalyse/w-python-numpy-grundlagen}}
```

Intro

Voraussetzungen

- Grundlagen Python
- Einbinden von zusätzlichen Paketen
- Plotten mit Matplotlib

Verwendete Pakete und Datensätze

Pakete

- [NumPy](#)
- [Matplotlib](#)

Datensätze

- [TC01.csv](#)
- [Bild: Mona Lisa](#)
- [Bild: Campus](#)

Bearbeitungszeit

Geschätzte Bearbeitungszeit: 2h

Lernziele

- Einleitung: was ist NumPy, Vor- und Nachteile
- Nutzen des NumPy-Moduls
- Erstellen von NumPy-Arrays
- Slicing

- Lesen und schreiben von Dateien
- Arbeiten mit Bildern

1 Einführung NumPy

NumPy ist eine leistungsstarke Bibliothek für Python, die für numerisches Rechnen und Datenanalyse verwendet wird. Daher auch der Name NumPy, ein Akronym für “Numerisches Python” (englisch: “Numeric Python” oder “Numerical Python”). NumPy selbst ist hauptsächlich in der Programmiersprache C geschrieben, weshalb NumPy generell sehr schnell ist.

NumPy bietet ein effizientes Arbeiten mit kleinen und großen Vektoren und Matrizen, die so ansonsten nur umständlich in nativem Python implementiert werden würden. Dabei bietet NumPy auch die Möglichkeit, einfach mit Vektoren und Matrizen zu rechnen, und das auch für sehr große Datenmengen.

Diese Einführung wird Ihnen dabei helfen, die Grundlagen von NumPy zu verstehen und zu nutzen.

1.1 Vorteile & Nachteile

Fast immer sind Operationen mit Numpy Datenstrukturen schneller. Im Gegensatz zu nativen Python Listen kann man dort aber nur einen Datentyp pro Liste speichern.

Warum ist numpy oftmals schneller?

NumPy implementiert eine effizientere Speicherung von Listen im Speicher. Nativ speichert Python Listeninhalte aufgeteilt, wo gerade Platz ist.

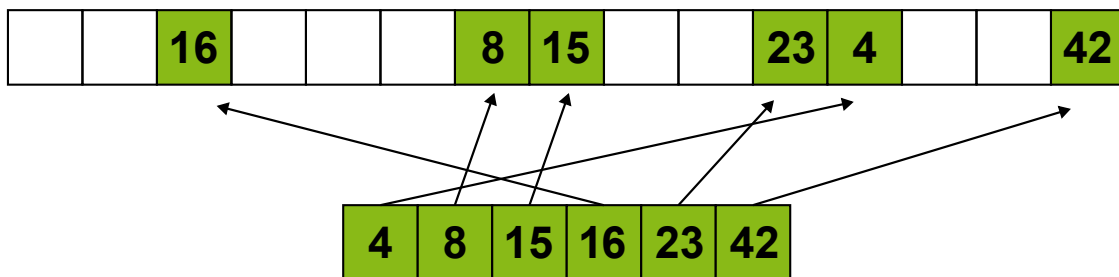


Figure 1.1: Speicherung von Daten in nativem Python

Dagegen werden NumPy Arrays und Matrizen zusammenhängend gespeichert, was einen effizienteren Datenaufruf ermöglicht.

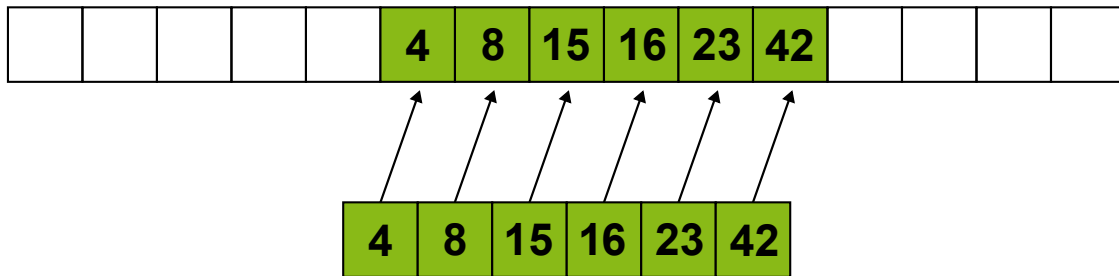


Figure 1.2: Speicherung von Daten bei Numpy

Dies bedeutet aber auch, dass es eine Erweiterung der Liste deutlich schneller ist als eine Erweiterung von Arrays oder Matrizen. Bei Listen kann jeder freie Platz genutzt werden, während Arrays und Matrizen an einen neuen Ort im Speicher kopiert werden müssen.

1.2 Einbinden des Pakets

NumPy wird über folgende Zeile eingebunden. Dabei hat sich global der Standard entwickelt, als Alias `np` zu verwenden.

```
import numpy as np
```

1.3 Referenzen

Sämtliche hier vorgestellten Funktionen lassen sich in der (englischen) NumPy-Dokumentation nachschlagen: [Dokumentation](#)