

# Einführung in die Programmierung mit Python

Dr. Bernd Zey

Abteilung Forschungsdatenmanagement

Universitätsbibliothek, TU Dortmund

07.03.2025

- Worum geht es heute hier?
  - Kurz-Einstieg in Python
  - Anwendungsfall
    - Tabellarische Daten laden und auswerten
    - Ergebnisse grafisch darstellen
    - Ergebnisse als Datei speichern
  - Keine Python-/Programmierkenntnisse erforderlich
- Motivation
  - Warum Python?
  - Entwicklungsumgebungen (IDE)
- Ausblick, Take-Home
- Live-Coding
  - JupyterLab

# Warum Python?

- **Standard** in Data Science
- **Zahlreiche Module** für die statistische Auswertung und Visualisierung (*Open Source*)
  - **Pandas**, NumPy, Matplotlib, Seaborn, ...
- **Einsteigerfreundliche Programmiersprache**
  - Allgemeine Programmiersprache, Plattform-unabhängig
  - Viele (wissenschaftliche) Open Source-Projekte
  - Einfache Syntax
  - Objektorientierung, Prozedural, ...
  - Schnell umfangreiche Programme möglich
  - Einfaches Datei-Handling
  - Viele Module, u.a. für Machine Learning (TensorFlow, Keras, ...)
  - Online-Hilfe



- Entwicklungsumgebung – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>



# Entwicklungsumgebungen



Gemeinsam besser  
studieren, forschen und lehren

- Entwicklungsumgebungen
  - JupyterLab <https://jupyterlab.com/>

The screenshot shows the JupyterLab interface. On the left is a file browser with a search bar and a list of files. The file '2024-03-py...' is selected. The main area shows a notebook with the following content:

```
# Im nächsten Schritt wird das beispielhaft vorgeführt, wenn die Datei im Unterordner "data" liegt
# import os
```

### Dateien laden

- csv-Datei (comma separated value-Datei)
- Optional: falls Date im Unterordner: Pfad mit os zusammenbauen

```
[50]: # Dateiname der zu Ladenden Datei
#filename = os.path.join("data", "Titanic-Dataset.csv") # optional: wenn die Datei im Unte
filename = "Titanic-Dataset.csv"
```

```
[51]: # Datei Laden und in Variable "data" speichern
data = pd.read_csv(filename)
```

```
[52]: # Tabelle data anzeigen
# In JupyterLab direkt mit dem Namen möglich
data
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
[52]:	0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
	1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs)	female	38.0	1	0	PC 17599	71.2833	C85	C

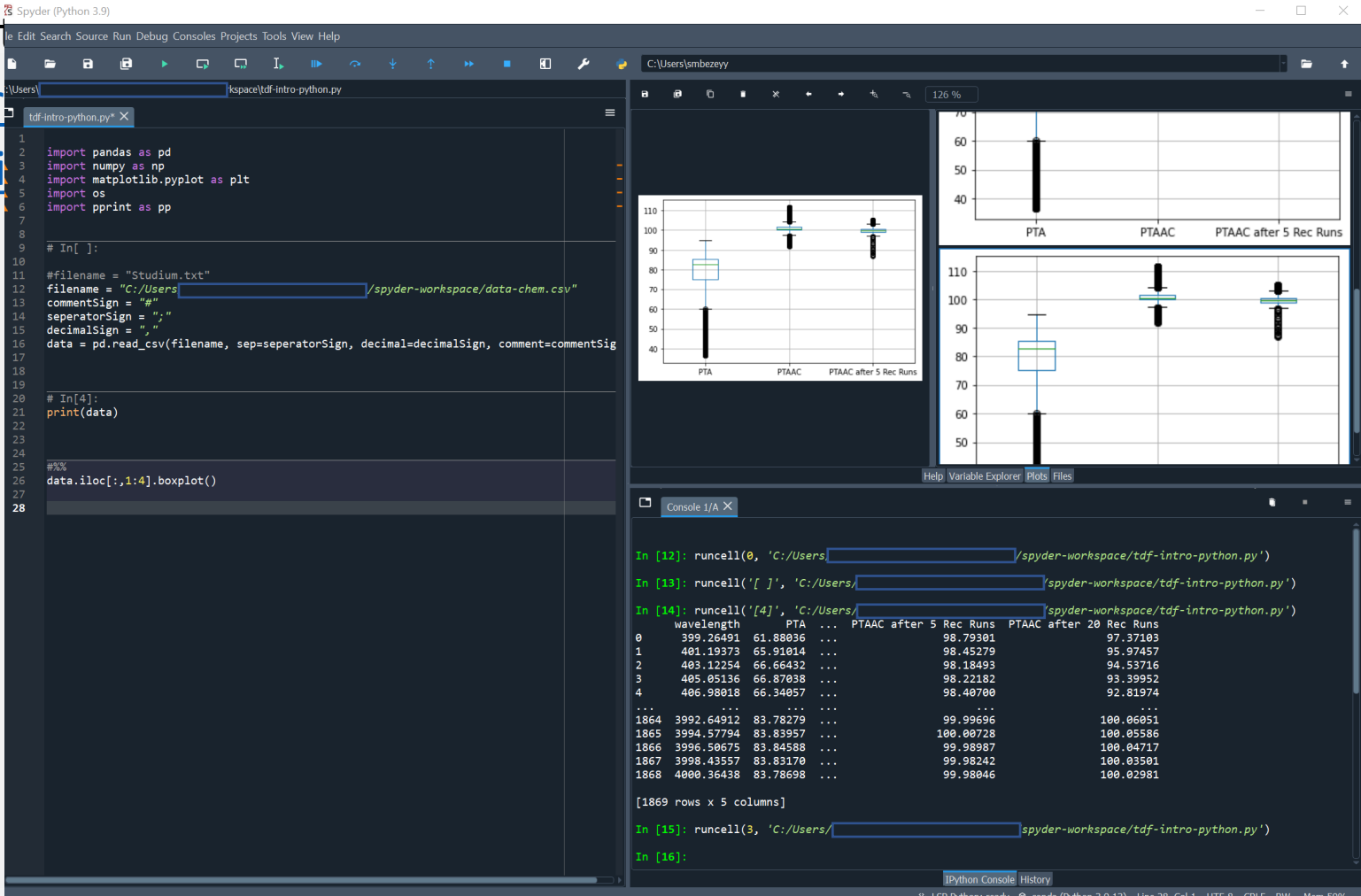
# Entwicklungsumgebungen

- Entwicklungsumgebung – IDE (Integrated Development Environment)
  - JupyterLab <https://jupyter.org/>
  - Spyder <https://www.spyder-ide.org/>



# Entwicklungsumgebungen

- Entwicklungsumgebung – IDE (Integrierte Entwicklungsumgebung)
  - JupyterLab <https://jupyter.org/>
  - Spyder <https://www.spyder-ide.org/>



```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import os
5 import pprint as pp
6
7
8
9 # In[ ]:
10
11 #filename = "Studium.txt"
12 filename = "C:/Users/[redacted]/spyder-workspace/data-chem.csv"
13 commentSign = "#"
14 seperatorSign = ";"
15 decimalSign = "."
16 data = pd.read_csv(filename, sep=seperatorSign, decimal=decimalSign, comment=commentSign)
17
18
19
20 # In[4]:
21 print(data)
22
23
24
25 #%%
26 data.iloc[:,1:4].boxplot()
27
28
```

	PTA	PTAAC	PTAAC after 5 Rec Runs
wavelength	399.26491	61.88036	98.79301
0	401.19373	65.91014	98.45279
1	403.12254	66.66432	98.18493
2	405.05136	66.87038	98.22182
3	406.98018	66.34057	98.40700
4	...	...	...
1864	3992.64912	83.78279	99.99696
1865	3994.57794	83.83957	100.00728
1866	3996.50675	83.84588	99.98987
1867	3998.43557	83.83170	99.98242
1868	4000.36438	83.78698	99.98046

# Entwicklungsumgebungen

- Entwicklungsumgebung – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code <https://code.visualstudio.com/>





- Entwicklungsumgebung – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>



- Spyder <https://www.spyder-ide.org/>



- PyCharm <https://www.jetbrains.com/pycharm/>



- Visual Studio Code <https://code.visualstudio.com/>



- Mein Tipp zum Einstieg: Anaconda



- Python-Distribution, installiert Python

- Liefert mehrere IDEs (u.a. alle oben genannten), Anaconda Navigator

- Einfache Environment- und Paket-Verwaltung

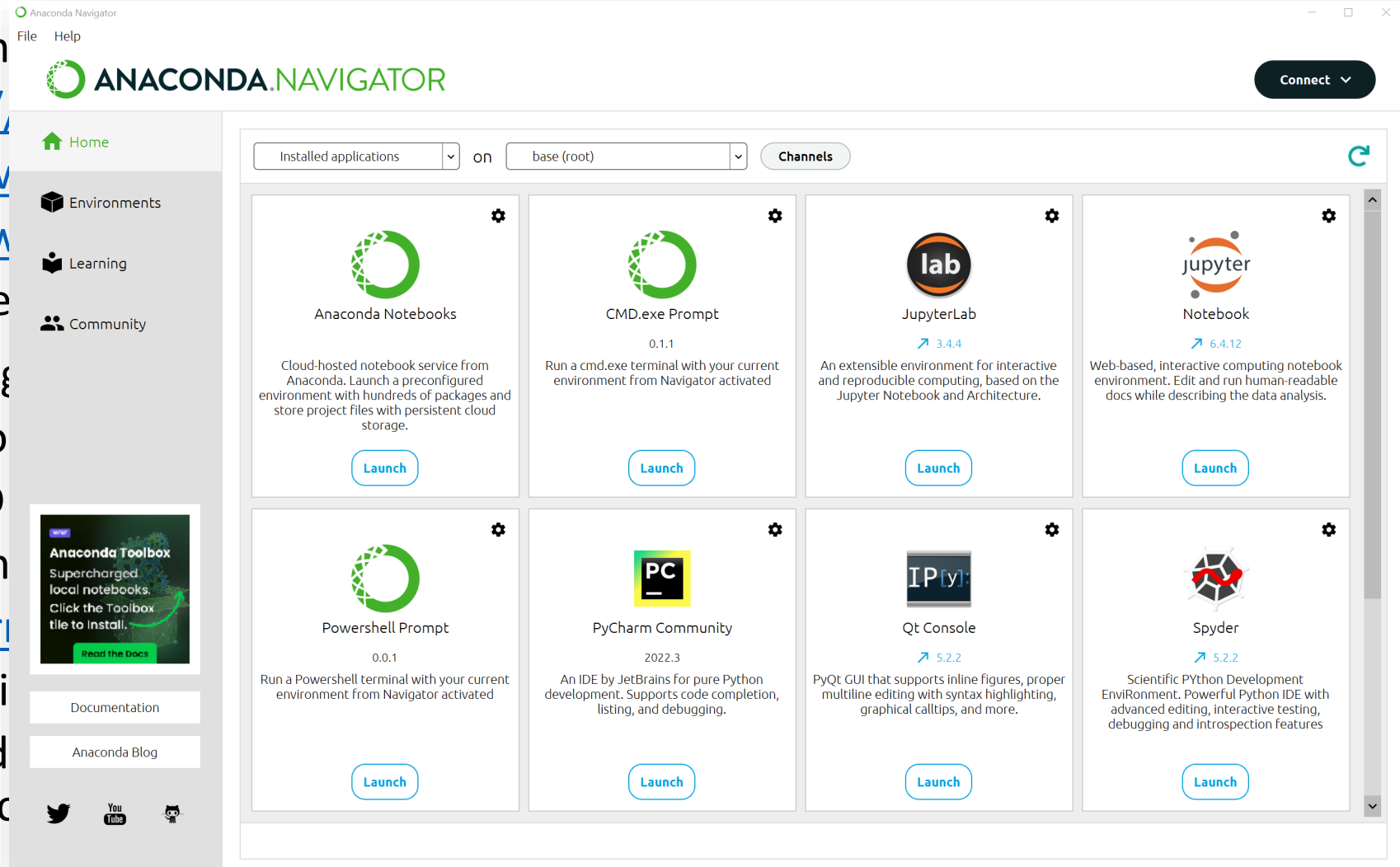
- Vorsicht, siehe [Terms of Use FAQ](#):

- “free for individuals and small organizations” (<200 employees, contractors)

- “free for all educational entities [...] at accredited universities worldwide” (course curriculum, including teaching, learning, and researching)

# Entwicklungsumgebungen

- Entwicklungsumgebungen
  - JupyterLab <https://jupyterlab.readthedocs.io/en/latest/>
  - Spyder <https://www.spyder-ide.org/>
  - PyCharm <https://www.jetbrains.com/pycharm/>
  - Visual Studio Code
- Mein Tipp zum Einstieg
  - Python-Distribution
  - Liefert mehrere IDE
  - Einfache Environment
  - Vorsicht, siehe [Termin](#)
    - “free for individual use”
    - “free for all educational curriculum, including for-profit institutions”



# Entwicklungsumgebungen

- Entwicklungsumgebung – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code <https://code.visualstudio.com/>



- Freie Alternative: z.B. Miniforge

- Installation-Beschreibung: siehe <https://fdm.tu-dortmund.de/informationen/workshop-einfuehrung-in-python/>



```
Miniforge Prompt - conda install pandas
(base) C:\Users\<redacted>>conda install pandas
Retrieving notices: done
Channels:
 - conda-forge
Platform: win-64
Collecting package metadata (repodata.json): \

Miniforge Prompt - jupyter lab
(base) C:\Users\<redacted>>jupyter lab
[I 2025-02-20 12:17:02.039 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2025-02-20 12:17:02.045 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2025-02-20 12:17:02.056 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-02-20 12:17:02.414 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-02-20 12:17:02.464 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-02-20 12:17:02.468 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-02-20 12:17:02.469 ServerApp] jupyter_server_terminals | extension was successfully loaded.
```

- Entwicklungsumgebung – IDE (Integrated Development Environment)

- JupyterLab <https://jupyter.org/>
- Spyder <https://www.spyder-ide.org/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Visual Studio Code <https://code.visualstudio.com/>



- Freie Alternative: z.B. Miniforge

- Online-Lösungen

- Temporär: <https://jupyter.org/try> → JupyterLab
  - `%pip install openpyxl`
  - Keine sensiblen Daten!
- Mit Google-Account:
  - Google Colab <https://colab.research.google.com/>

- JupyterLab, Live Coding
  - Abschnittsweise Ausführung
  - Markdown-Dokumentation und -Beschreibung
  - Direkte Ausgabe der Diagramme
- Code verfügbar auf Homepage
  - <https://fdm.tu-dortmund.de/informationen/workshop-einfuehrung-in-python/>
  - Lizenz: MIT
- Verwendete Daten
  - <https://www.kaggle.com/datasets/yasserh/titanic-dataset>
  - Datensatz der Titanic, Lizenz: CC0
    - Siehe FDM-Homepage

# Ziele

- *Take-Away Message:*  
Python *einfache* Programmiersprache  
→ **Automatisierung**
- Mächtige Module (heute: Pandas) für Data Science
- Online gute Hilfe (Diskussionen, Code-Schnipsel)
  - z.B. <https://stackoverflow.com>
  - Websuche → „python pandas *thema*“

