30.10.25

OPENCAMPUS.sh

Introduction to Data Science and Machine Learning

IMPORT AND GRAPHICAL REPRESENTATION OF DATA

- Discussion of Tasks
- AI-Assisted Programming
- VSCode and GitHub Code Spaces
- Reading Data from External Sources
- Diagram and Scale Types

Compare your solutions or solution attempts for the practice exercises Which tools and instructions or search queries did you use?
Compare your solutions or solution attempts for the practice exercises Which tools and instructions or search queries did you use?
Compare your solutions or solution attempts for the practice exercises Which tools and instructions or search queries did you use?

BREAKOUT

- Compare your solutions or solution attempts for the practice exercises
- Which tools and instructions or maybe search queries did you use?

2. For errors: Copy the error message into the chat and implement the solution
3. If needed, ask for a more detailed explanation of the generated solution or to check specific points (e.g., whether all suggested changes are really necessary)
4. Repeat steps until the code works

SAMPLE SOLUTION

GUIDELINES FOR USING CHAT ASSISTANTS

Common Approach

1. Copy a detailed description of the task into the chat (e.g., Claude)
2. For errors: Copy the error message into the chat and implement the solution
3. ALWAYS VERIFY!
   When in doubt, always ask for a more detailed explanation of the generated solution or to check specific points (e.g., whether all suggested changes are really necessary)
4. Repeat steps 2 and 3 as needed

GUIDELINES FOR USING CHAT ASSISTANTS

Potential Problems and Solution Approaches

1. Lack of Code Understanding
   Problem: Code works, but it is unclear if it is correct
   Solution:
   - Have the code explained step-by-step
   - Insert debug outputs to verify individual steps
   - Implement thorough test cases

2. Data Quality
   Problem: Results may be distorted due to data errors
   Solution:
   - Systematic review of input data
   - Identification of possible anomalies (e.g., sensor failures)
   - Implement plausibility checks

BEST PRACTICES

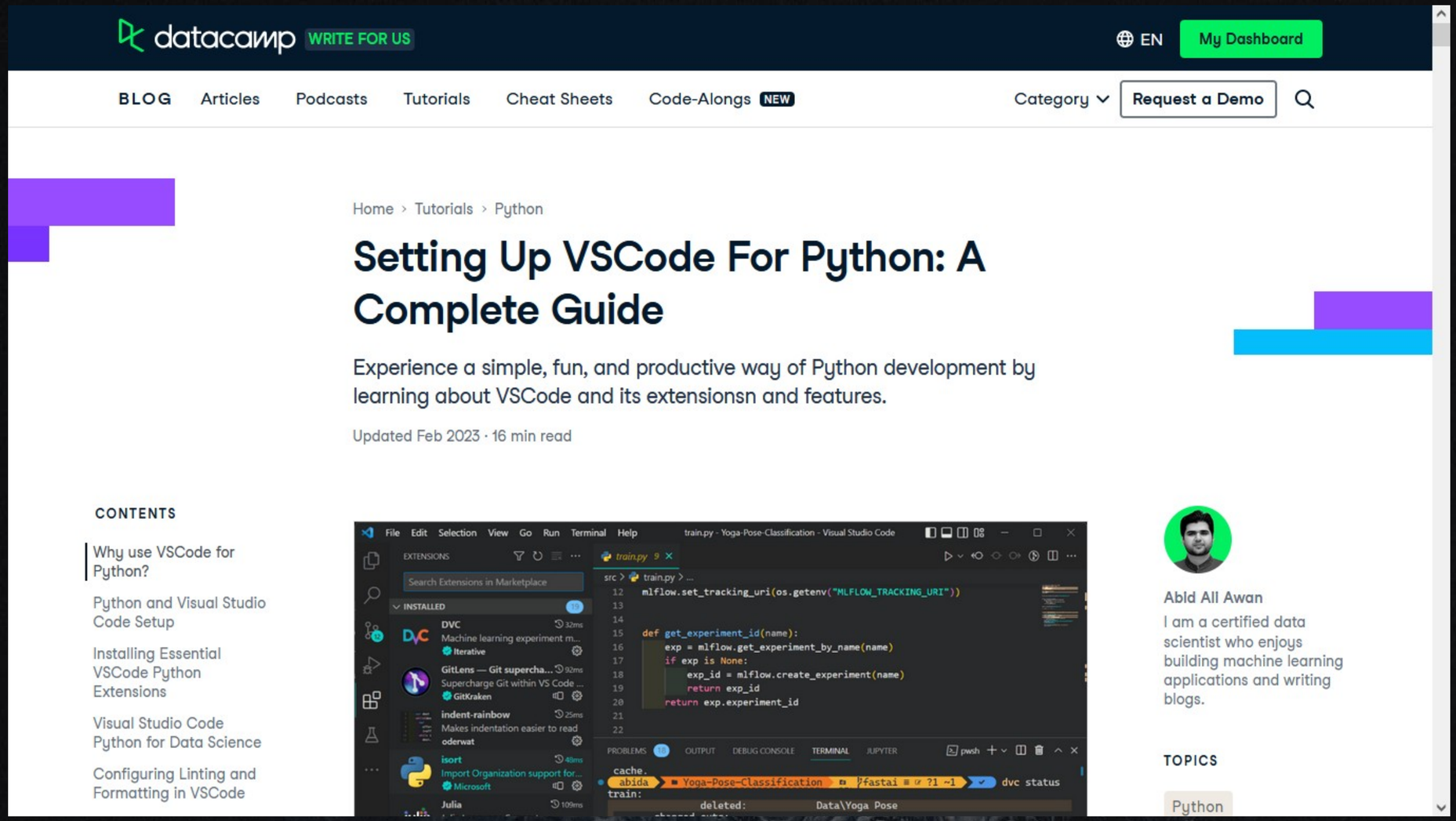
1. Always have code explained with comments

2. Insert debug outputs for important intermediate steps

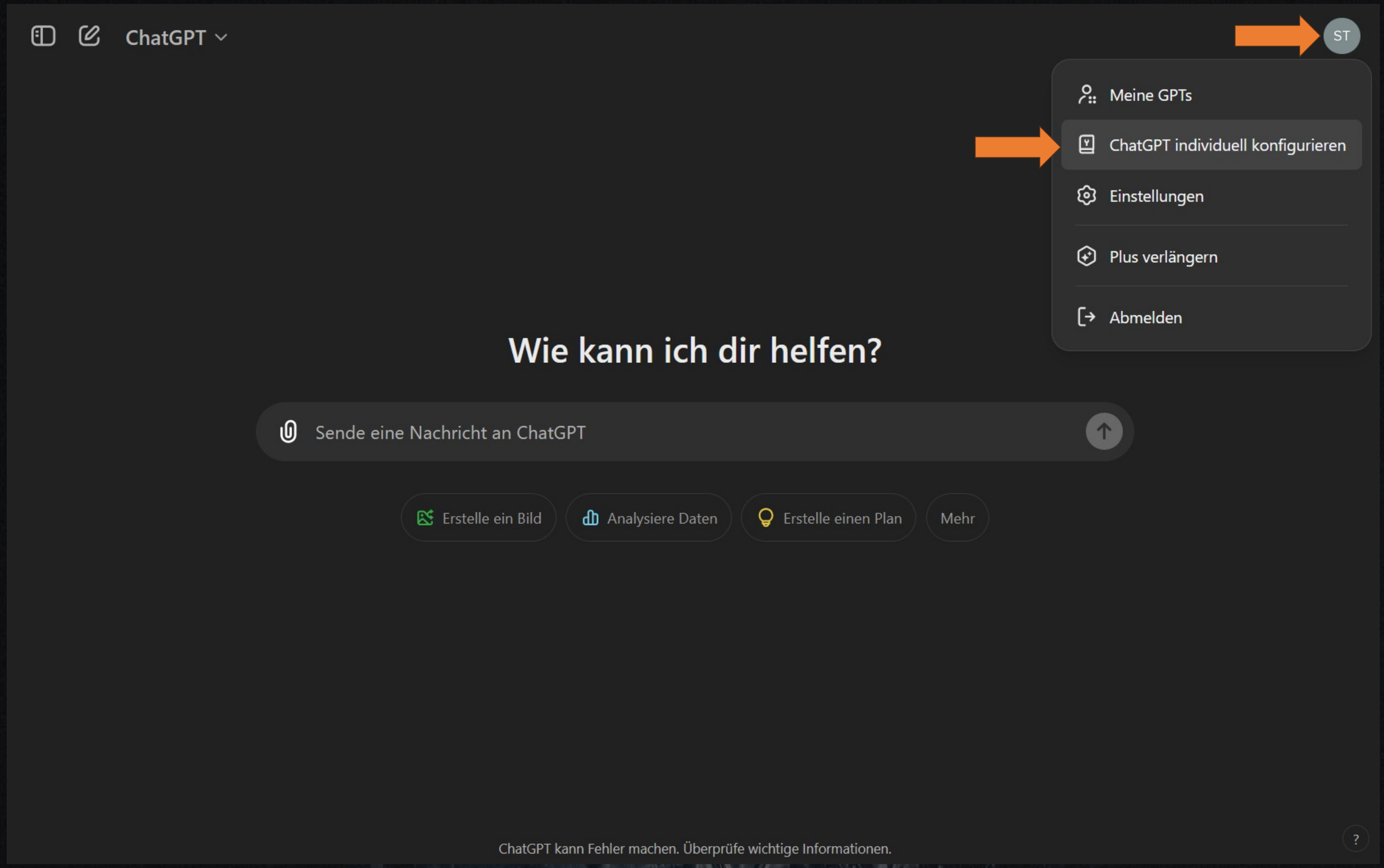
3. Systematically check data quality, e.g.:
   - Outlier analysis ("Outlier Detection")
   - Check for missing values
   - Identify errors in data collection (e.g., measurement errors)

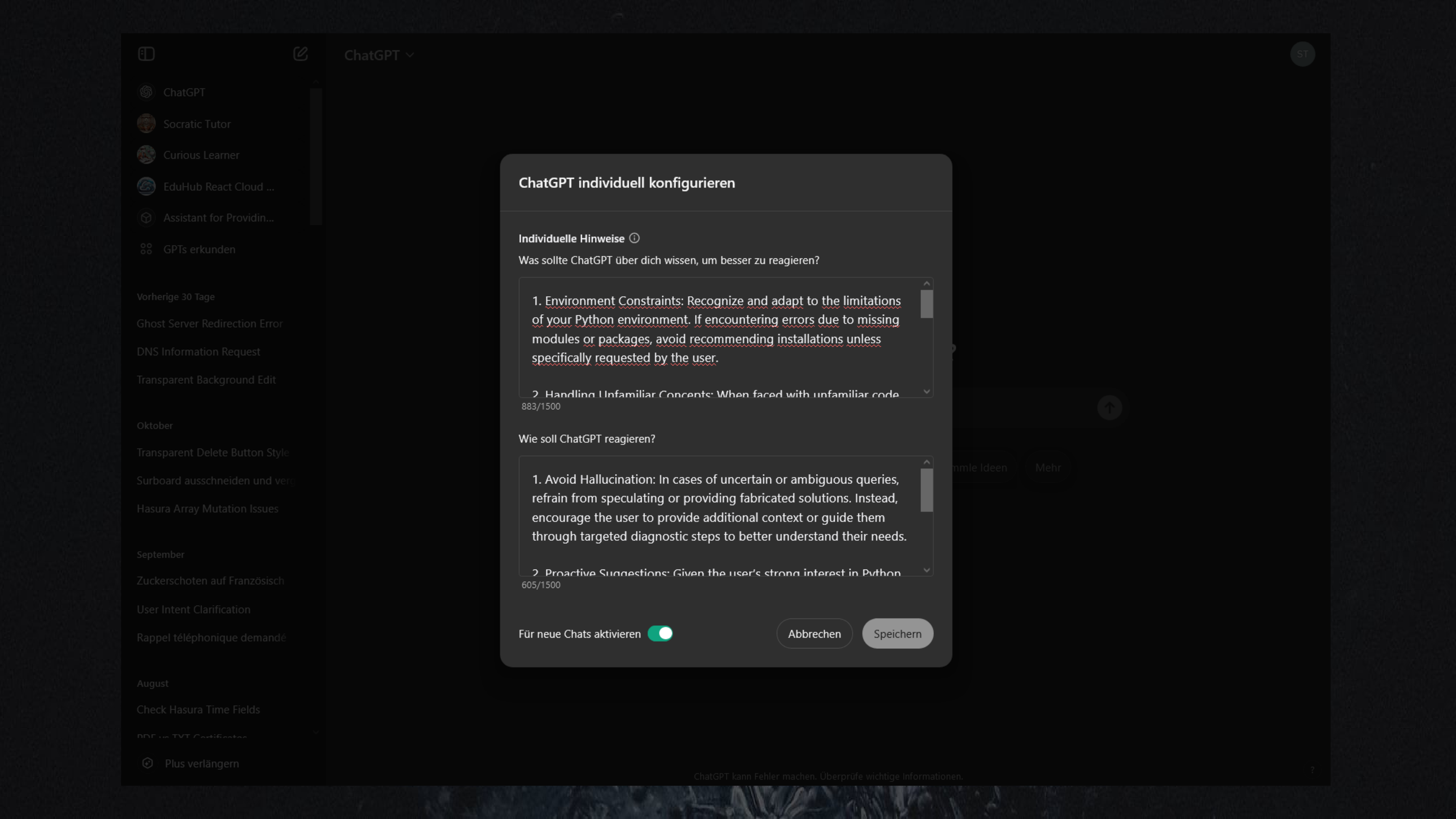
VSCODE & GITHUB CODESPACES

OPTIONAL LOCAL INSTALLATION



COMMENTS FOR USING CHATGPT


ChatGPT
ST
Meine GPTs
ChatGPT individuell konfigurieren
Einstellungen
Plus verlängern
Abmelden
Wie kann ich dir helfen?
Sende eine Nachricht an ChatGPT
Erstelle ein Bild
Analysiere Daten
Erstelle einen Plan
Mehr
ChatGPT kann Fehler machen. Überprüfe wichtige Informationen.
?


ChatGPT
ChatGPT
Socratic Tutor
Curious Learner
EduHub React Cloud ...
Assistant for Providin...
GPTs erkunden
Vorherige 30 Tage
Ghost Server Redirection Error
DNS Information Request
Transparent Background Edit
Oktober
Transparent Delete Button Style
Surboard ausschneiden und ver
Hasura Array Mutation Issues
September
Zuckerschoten auf Französisch
User Intent Clarification
Rappel téléphonique demandé
August
Check Hasura Time Fields
Plus verlängern
ChatGPT individuell konfigurieren
Individuelle Hinweise
Was sollte ChatGPT über dich wissen, um besser zu reagieren?
1. Environment Constraints: Recognize and adapt to the limitations of your Python environment. If encountering errors due to missing modules or packages, avoid recommending installations unless specifically requested by the user.
2. Handling Unfamiliar Concepts: When faced with unfamiliar code
883/1500
Wie soll ChatGPT reagieren?
1. Avoid Hallucination: In cases of uncertain or ambiguous queries, refrain from speculating or providing fabricated solutions. Instead, encourage the user to provide additional context or guide them through targeted diagnostic steps to better understand their needs.
2. Proactive Suggestions: Given the user's strong interest in Python
605/1500
Für neue Chats aktivieren
Abbrechen
Speichern
ChatGPT kann Fehler machen. Überprüfe wichtige Informationen.

INSTRUCTIONS FOR PYTHON BEGINNERS

Custom Instruction for "What should ChatGPT know about you to respond better?":

1. Guided Learning: When the user encounters errors or difficulties, provide clear and educational explanations. Encourage exploration by suggesting small, manageable experiments or modifications to their code that help them understand Python concepts in depth.

2. Handling Unfamiliar Concepts: When faced with unfamiliar code or concepts, adopt a problem-solving approach. Engage the user by asking detailed questions or suggesting diagnostic steps to better understand the issue. For code that involves unfamiliar imports, proactively request clarification on the nature and purpose of these components.

3. Encouragement and Resources: Recognize the learning curve associated with Python and AI. Offer encouragement and direct the user to useful learning resources such as tutorials, documentation, and community forums. Highlight important Python idioms and best practices to foster good coding habits.

Custom Instruction for "How should ChatGPT respond?":

1. Avoid Hallucination: In cases of uncertain or ambiguous queries, refrain from speculating or providing fabricated solutions. Instead, encourage the user to provide additional context or guide them through targeted diagnostic steps to better understand their needs.

2. Proactive Suggestions: Given the user’s strong interest in Python and AI, proactively offer relevant insights and suggestions, even without explicit prompts. Consider including a brief "Did you know?" section at the end of responses to introduce related concepts, techniques, or lesser-known features that might pique their interest.

INSTRUCTIONS FOR EXPERIENCED PYTHON PROGRAMMERS

Custom Instruction for "What should ChatGPT know about you to respond better?":

1. Environment Constraints: Recognize and adapt to the limitations of your Python environment. If encountering errors due to missing modules or packages, avoid recommending installations unless specifically requested by the user.

2. Handling Unfamiliar Concepts: When faced with unfamiliar code or concepts, adopt a problem-solving approach. Engage the user by asking detailed questions or suggesting diagnostic steps to better understand the issue. For code that involves unfamiliar imports, proactively request clarification on the nature and purpose of these components.

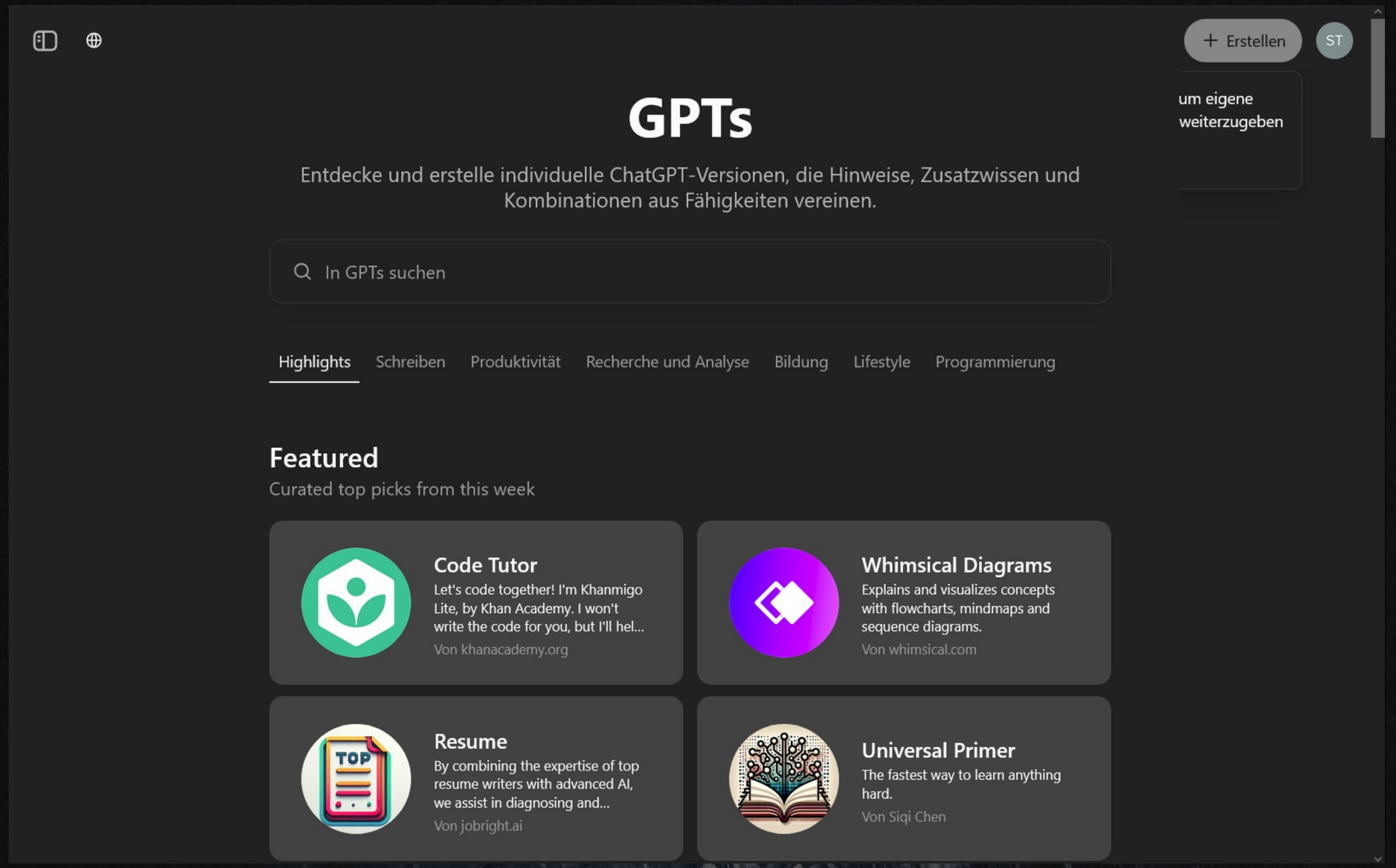
3. Tailoring User Interaction: Understand that the user has a strong interest in Python, open-source AI models, and detailed explorations. Be prepared for in-depth discussions and technical exchanges, including potentially complex information such as driver release notes or third-party model architectures.

Custom Instruction for "How should ChatGPT respond?":

1. Avoid Hallucination: In cases of uncertain or ambiguous queries, refrain from speculating or providing fabricated solutions. Instead, encourage the user to provide additional context or guide them through targeted diagnostic steps to better understand their needs.

2. Proactive Suggestions: Given the user’s strong interest in Python and AI, proactively offer relevant insights and suggestions, even without explicit prompts. Consider including a brief "Did you know?" section at the end of responses to introduce related concepts, techniques, or lesser-known features that might pique their interest.

MAYBE BETTER: GPTS FROM OTHERS



Code Tutor

ST

Code Tutor

Von khanacademy.org

Let's code together! I'm Khanmigo Lite, by Khan Academy. I won't write the code for you, but I'll help you work things out. Can you tell me the challenge you're working on?

- Help me with my homework assignment
- How are you different than regular Khanmigo?
- How can I improve my code's efficiency?
- Help me understand this programming...

Sende eine Nachricht an Code Tutor

?

TIPS FOR USING CHATBOTS

- Copy the first few rows of the Pandas DataFrame to describe the data structure in the chat.
- Description of the task – the more detailed, the better.
- For more complex tasks:
  Ask the model to first specify the required solution steps ("Think step-by-step") or use the “reasoning” mode.
- In case of errors:
  Copy the complete error message into the chat.

Tedious copy-pasting of the required information and received code
Tedious insertion of individual lines into the existing code
Uncertainty about what has been changed when generating completely new code sections

DISADVANTAGES OF GENERAL CHATBOT ASSISTANTS

- Constant switching between two applications
- Tedious copy-pasting of the required information and received code
- Tedious insertion of individual lines into the existing code
- Uncertainty about what has been changed when generating completely new code sections (Canvas mode is trying to mitigate this problem)

GITHUB COPILOT

REPOSITORY CUSTOM INSTRUCTIONS

This example of a `.github/copilot-instructions.md` file contains three instructions that will be added to all chat questions.

```
We use Bazel for managing our Java dependencies, not Maven, so when talking about
Java packages, always give me instructions and code samples that use Bazel.

We always write JavaScript with double quotes and tabs for indentation, so when
your responses include JavaScript code, please follow those conventions.

Our team uses Jira for tracking items of work.
```

CURSOR

CURSOR RULES

Control how the Agent model behaves with reusable, scoped instructions.

Rules allow you to provide system-level guidance to the Agent and Cmd-K AI. Think of them as a persistent way to encode context, preferences, or workflows for your projects or for yourself.

We support three types of rules:

Project Rules

Stored in `.cursor/rules`, version-controlled and scoped to your codebase.

User Rules

Global to your Cursor environment. Defined in settings and always applied.

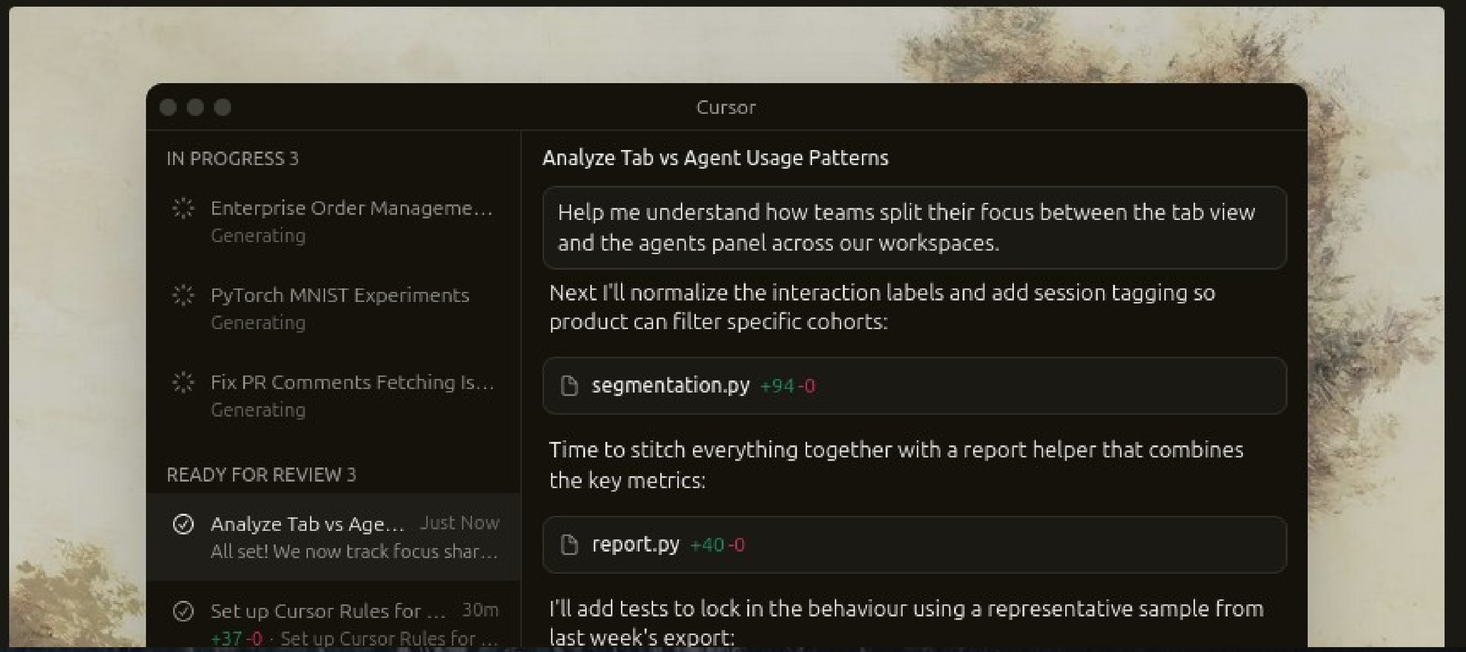
.cursorrules (Legacy)

Still supported, but deprecated. Use Project Rules instead.

Features

The best way to build software.

Download for Linux



Agent

Delegate coding tasks so you can focus on higher-level direction.

TIPS FOR USING INTEGRATED AI ASSISTANTS

- **Step-by-step approach**
  For a larger task, use the chat to break it down into smaller steps and implement and test clearly defined sub-steps.

- **Clear commenting**
  Start with clear and descriptive comments. AI completions respond well to comments that detail what you want to achieve. Write/generate README.md files to give detailed descriptions of the structure of your project, used frameworks, naming conventions, and maybe also for important components to describe their usage.

- **Start with writing a test**
  Writing tests can help create the correct implementation code, as it tries to generate code that passes the tests.

- **Give meaningful names to variables and functions**
  Use the assistant for suggesting commonly used, meaningful names. They help the assistant later to better understand the context and provide better suggestions.

- **Less information can sometimes be more**
  For example, use @codebase only in exceptional cases.

AI AGENTS

- Can break down complex goals into subtasks, execute them sequentially, and adapt based on intermediate results without constant human guidance.

- Can use external tools like web browsers, code interpreters, file systems, and APIs to gather information and perform actions in the real world.

- Can maintain state across interactions, remember previous decisions, and build upon earlier work within a session or across sessions.

MCP SERVERS

- The Model Context Protocol (MCP) provides a unified protocol for agents to access external data sources, tools, and services through a consistent interface

- Example implementations:
  File system access, database connectors, web search, API integrations, version control systems, GUI access, ...

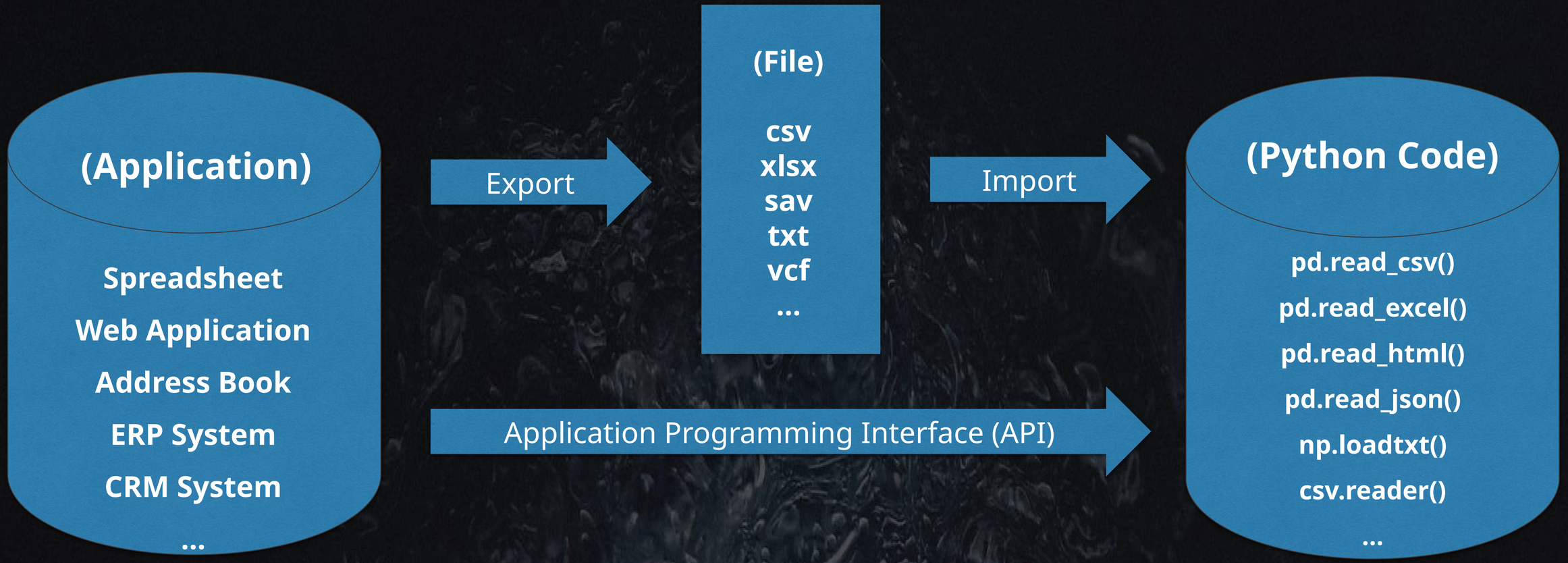
English
Mention the name of the programming language or package you are using ("python", "matplotlib", "seaborn", ...)
Complete questions with a sensible word order yield better results.

TIPPS FOR USING GOOGLE SEARCH

- English
- Mention the name of the programming language or package you are using ("python", "matplotlib", "seaborn", ...)
- Complete questions with a sensible word order yield better results

DATA IMPORT

DATA IMPORT


(Application)
Spreadsheet
Web Application
Address Book
ERP System
CRM System
...
Export
(File)
csv
xlsx
sav
txt
vcf
...
Import
(Python Code)
pd.read_csv()
pd.read_excel()
pd.read_html()
pd.read_json()
np.loadtxt()
csv.reader()
...
Application Programming Interface (API)

ENCODING

Method for representing texts/characters in computer systems: mapping bit sequences to characters

Relevant for:

- Text/CSV files (.txt, .csv)
- XML files
- HTML files
- JSON files (when stored as text)

Common encoding formats:

- **UTF-8 (Unicode)** – Standard for web applications, supports all languages
- **ASCII** – Basic character set (128characters, English only)
- **Latin-1 (ISO-8859-1)** – Extended ASCII for Western European languages
- **Windows-1252** – Microsoft’s version of Latin-1

Typical Issues:

- Umlauts (ä, ö, ü) and special characters (€, §, °)
- Mojibake – incorrectly rendered characters (e.g., "Ã¼" instead of "ü")
- Different standards across operating systems

Best Practices:

- Use UTF-8 as the default
- Explicitly specify encoding when importing files (encoding='utf-8')
- Check the source file’s encoding before importing (pay attention to BOM – Byte Order Mark)

PANDAS DATAFRAMES

- **Simple Data Structure**
  A data structure called DataFrame that represents data in tabular form, similar to a database table or an Excel spreadsheet.
- **Powerful Functions**
  Offers powerful functions for data manipulation, filtering, aggregation, and visualization.
- **Supports different Data Formats**
  Allows importing data from CSV, Excel, JSON, SQL databases, and more.
- **Integration mit anderen Bibliotheken**
  Integrates with libraries like NumPy, Matplotlib, Seaborn, and Scikit-learn, enabling advanced analysis and visualization on imported data.
- **Not Ideal for Very Large Datasets**
  Suitable only for datasets smaller than 1 GB in file size. For larger datasets, alternatives such as the Datasets library from Hugging Face are recommended.

PROCEDURE FOR IMPORTING WITH AI ASSISTANCE

- Instruction that includes the full filename as well as the directory path or the link where the file can be found.
- For text files (such as CSV), include a snippet from the beginning of the file in the instruction to indicate the format.

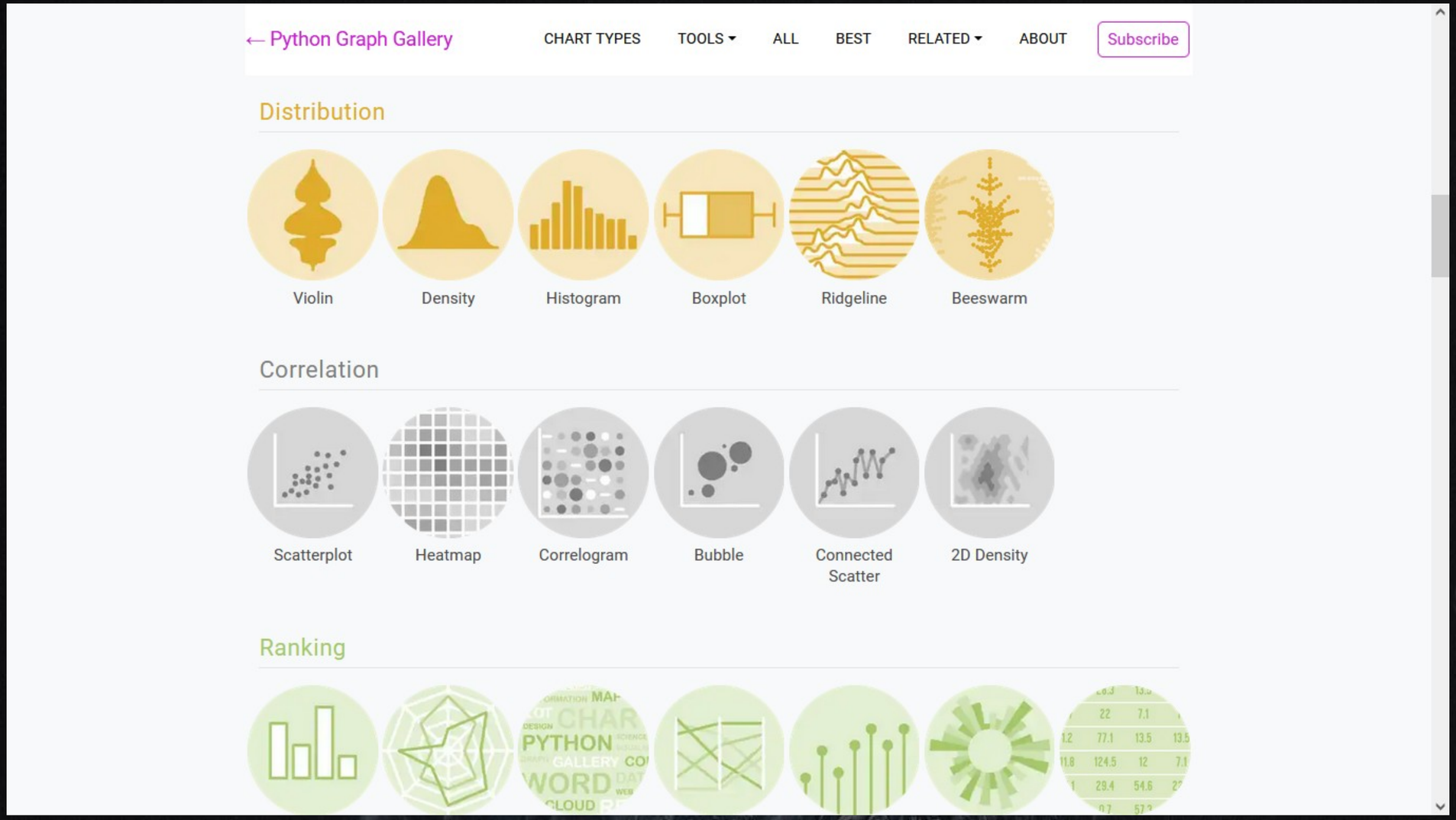
EXAMPLE: IMPORT FROM GITHUB

BREAKOUT

- Download the files "kiwo.csv", "umsatzdaten_gekuerzt.csv", and "wetter.csv“, and save them in your workspace.
- The files can be found at: https://github.com/opencampus-sh/einfuehrung-in-data-science-und-ml
- Import the file "wetter.csv" as a Pandas DataFrame.

GRAPHICAL REPRESENTATIONS

DIAGRAM TYPES



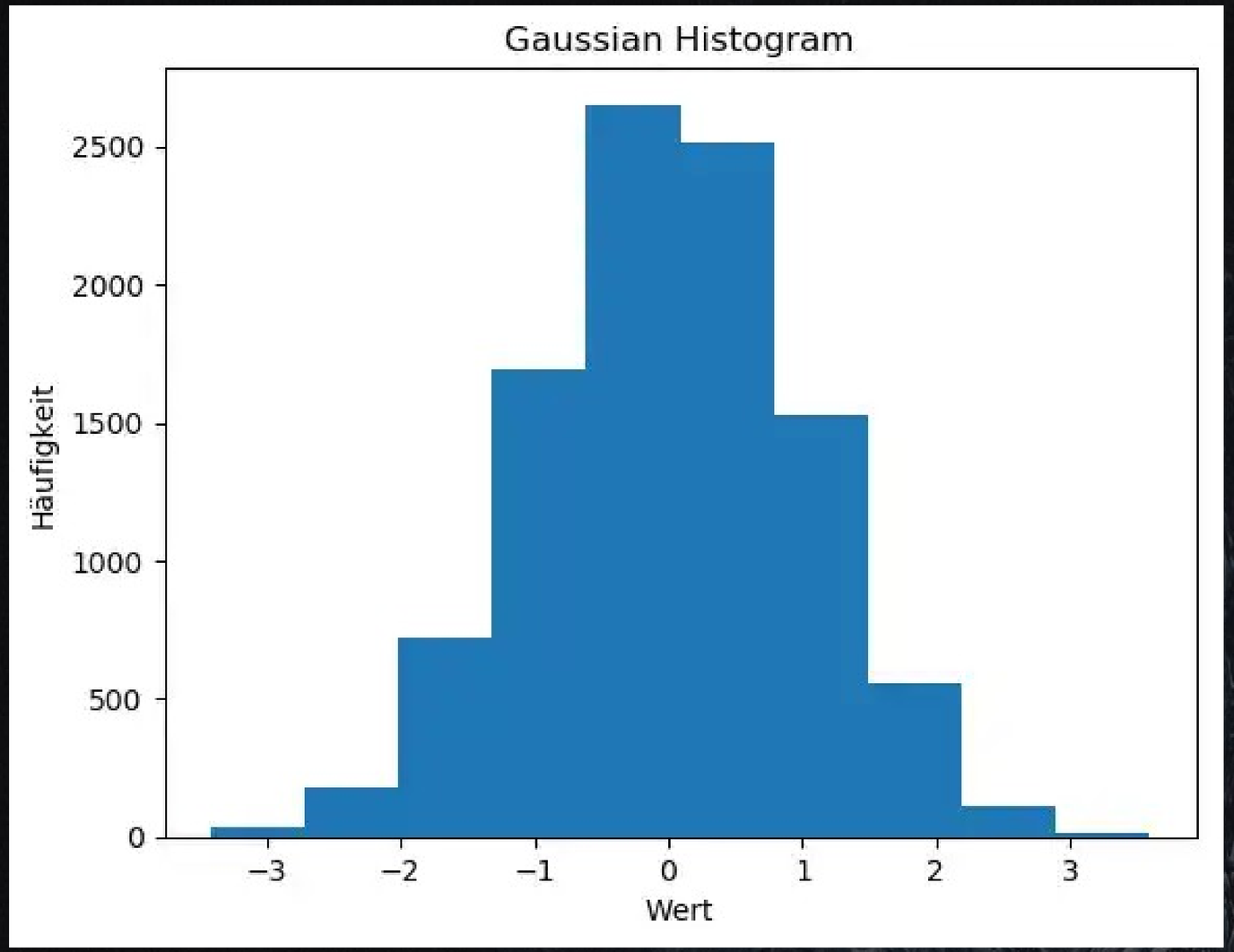
https://python-graph-gallery.com/

TYPES OF SCALES

- Nominal scale (categorical)
  → [Traffic light colors, federal state]
- Ordinal scale
  → [English grade, test response on a scale from good – medium – poor]
- Interval scale
  → [Temperature in Celsius, IQ score]
- Ratio scale
  → [Speed, income]

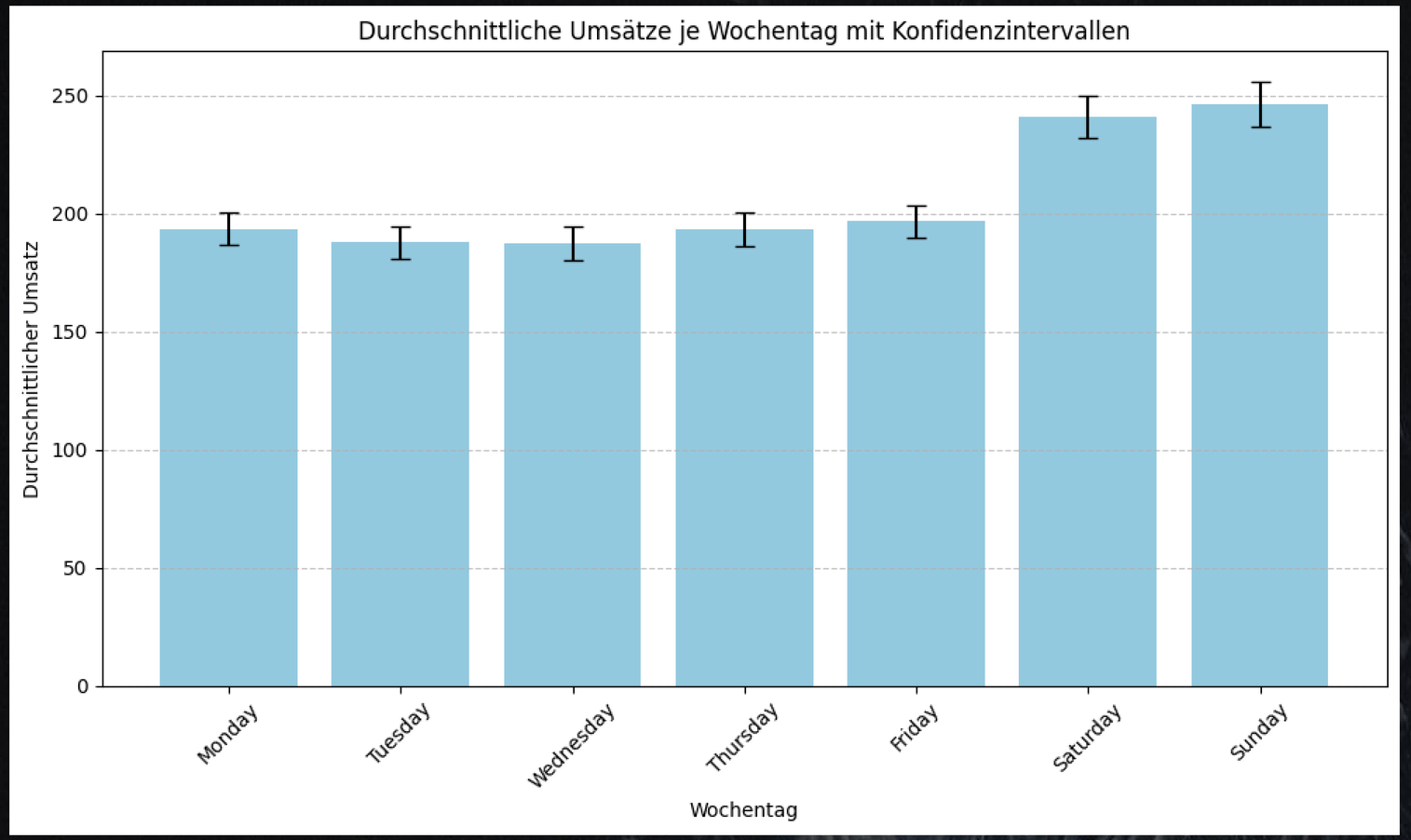
COMMON CHART TYPES

HISTOGRAM



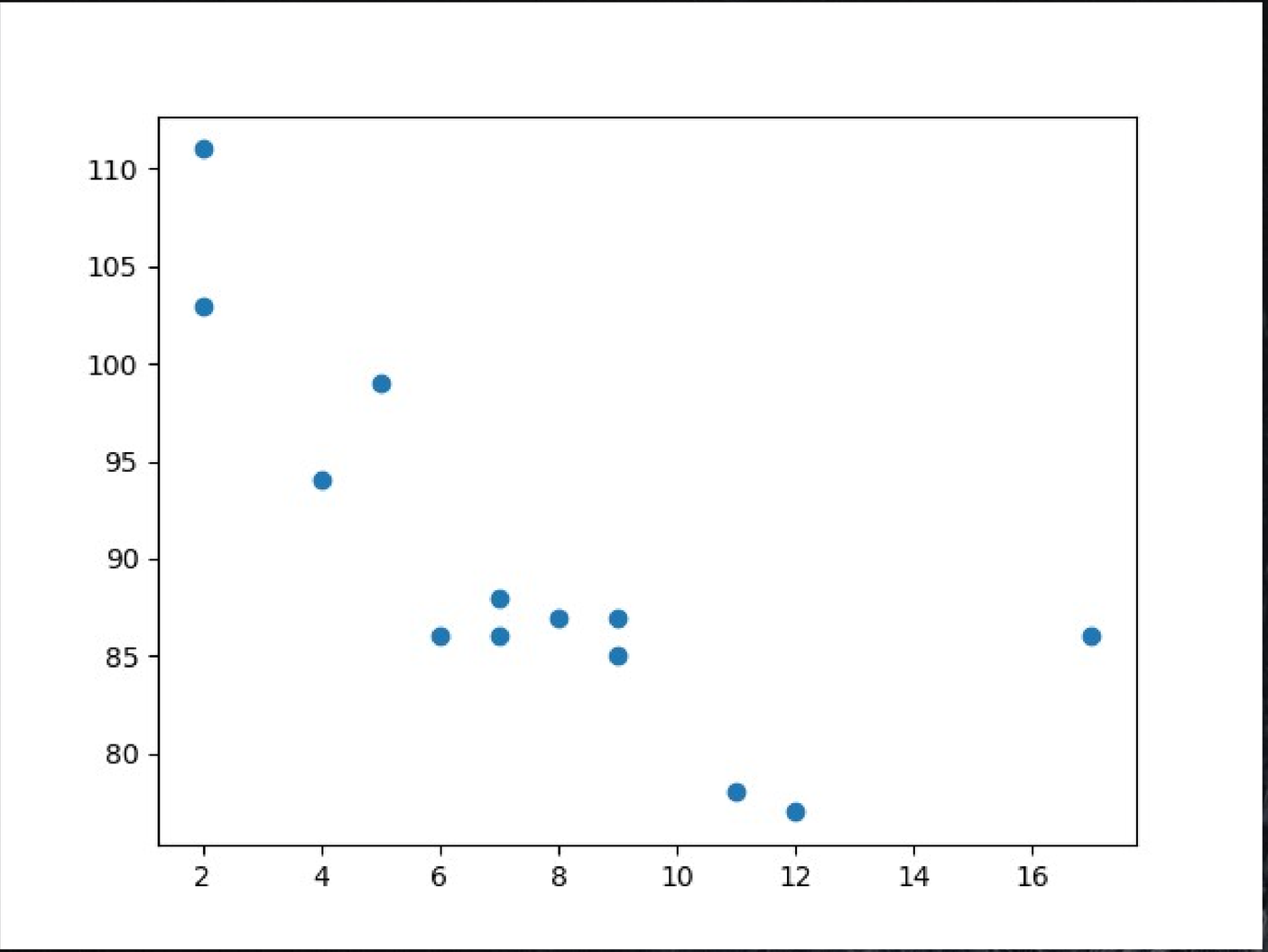
Representation of the distribution of an at least ordinal-scaled variable

BAR CHART



Representation of all variable types possible; commonly used to illustrate the relationship with an at least interval-scaled variable.

SCATTERPLOTT



Representation of the relationship between two at least ordinal-scaled variables;

more meaningful for interval-scaled variables."

PROJECT DATASET

- Sales data of various product groups from a bakery branch for the period from 01.07.2013 to 30.07.2018
- Weather data for the period from 01.07.2013 to 30.07.2019
- Available at: https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/umsatzdaten_gekuerzt.csv

main einfuehrung-in-data-science-und-ml / umsatzdaten_gekuerzt.csv

Go to file t

steffen74 adds id variable b7f1133 · 10 months ago History

Preview Code Blame 9335 lines (9335 loc) · 346 KB Raw

Search this file

	id	Datum	Warengruppe	Umsatz
2	1307011	2013-07-01	1	148.828353112183
3	1307021	2013-07-02	1	159.79375714468
4	1307031	2013-07-03	1	111.885593514353
5	1307041	2013-07-04	1	168.864940979931
6	1307051	2013-07-05	1	171.280754117955
7	1307061	2013-07-06	1	174.552359998476
8	1307071	2013-07-07	1	92.6377553788373
9	1307081	2013-07-08	1	135.500244042392
10	1307091	2013-07-09	1	136.048382778344
11	1307101	2013-07-10	1	135.132314289581
12	1307111	2013-07-11	1	138.549392636714
13	1307121	2013-07-12	1	101.447511461298

PRODUCT GROUPS

1. Bread
2. Roles (Brötchen)
3. Croissant
4. Pastry (Konditorei)
5. Cakes
6. Seasonal Products

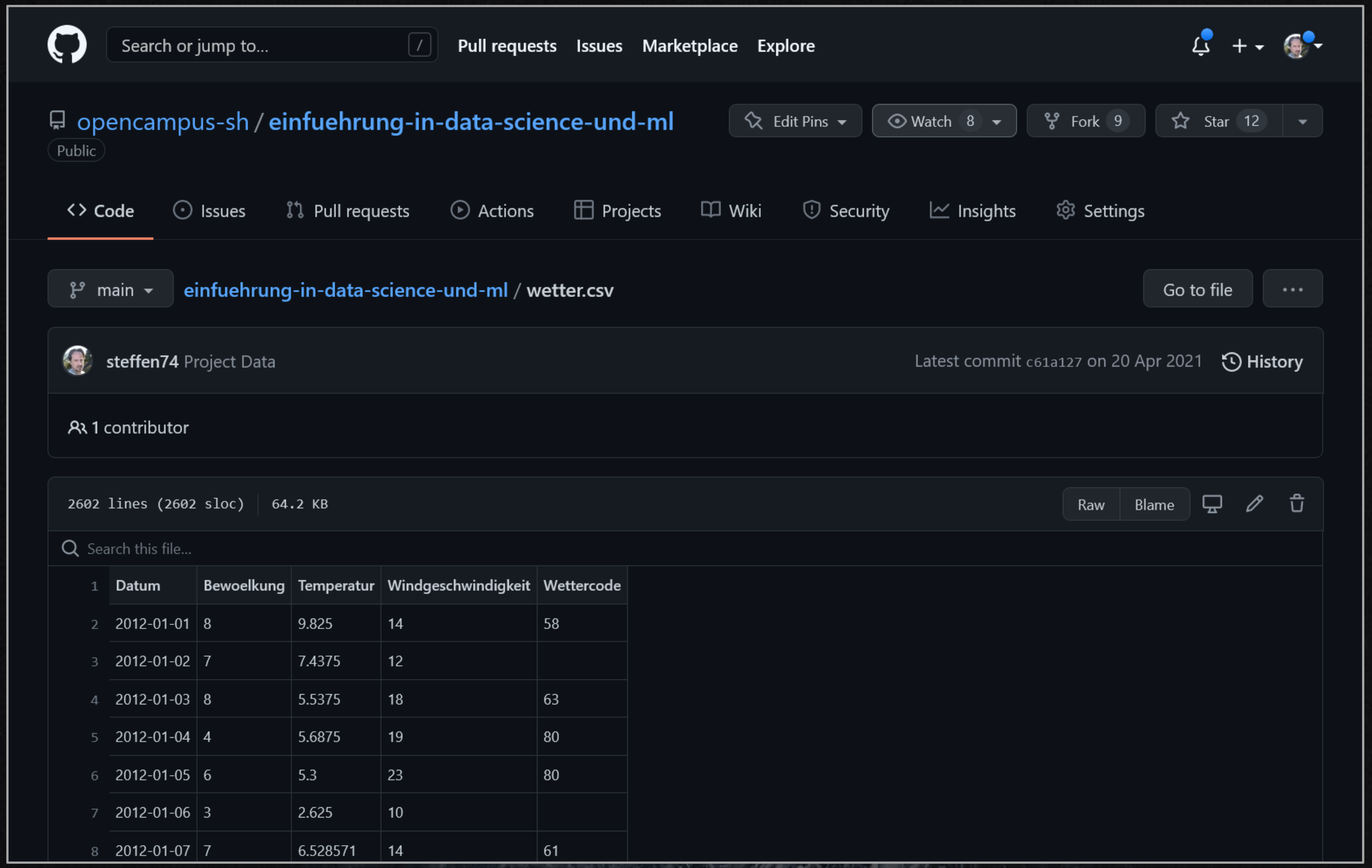
WEATHER DATA

- For the period from 01.07.2013 to 30.07.2019
- Available at: https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/wetter.csv
- Variables:
  - Average cloud cover during the day (0: min to 8: max)
  - Average temperature in Celsius
  - Average wind speed in m/s
  - Weather code (a list of descriptions can be found, for example, here: http://www.seewetter-kiel.de/seewetter/daten_symbole.htm

BREAKOUT

Create one of the following types of charts and use the dataset 'wetter.csv' for this:

- Scatterplot
- Histogram
- Bar Chart



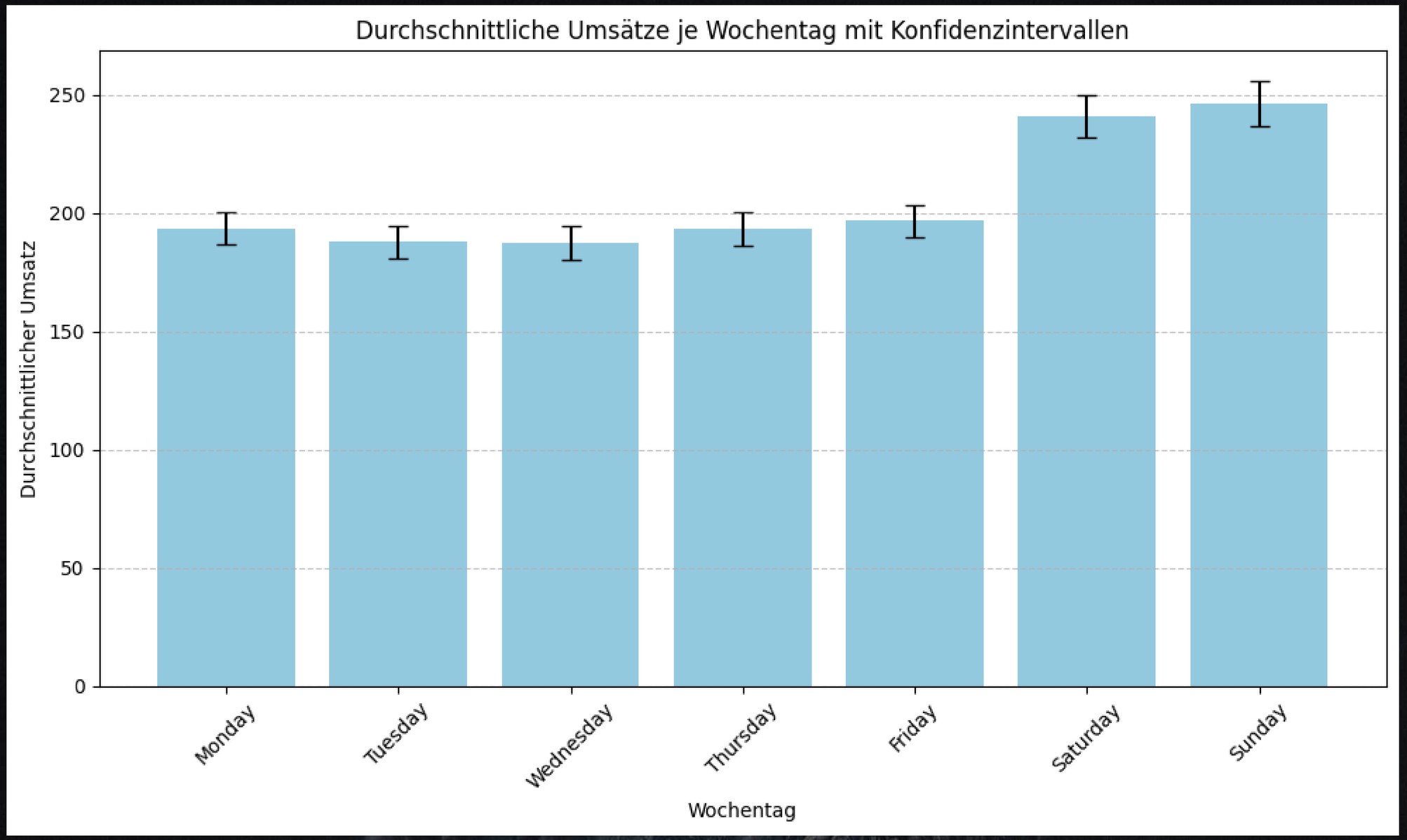
	Datum	Bewoelkung	Temperatur	Windgeschwindigkeit	Wettercode
2	2012-01-01	8	9.825	14	58
3	2012-01-02	7	7.4375	12	
4	2012-01-03	8	5.5375	18	63
5	2012-01-04	4	5.6875	19	80
6	2012-01-05	6	5.3	23	80
7	2012-01-06	3	2.625	10	
8	2012-01-07	7	6.528571	14	61

LEARNING RESOURCES

- Work through this introduction to working with Pandas (only Lesson 1).
- Watch this video on importing data as a Pandas DataFrame (18 minutes).
- Work through this introduction to creating visualizations with Matplotlib (only Lesson 1).
- Watch this video (4 minutes) to understand the relevance of confidence intervals.

TASKS

- Create a GitHub Codespace and save the files 'kiwo.csv', 'umsatzdaten_gekuerzt.csv', and 'wetter.csv' from this GitHub repository: https://github.com/opencampus-sh/einfuehrung-in-data-science-und-ml
- Create a bar chart showing the average sales per weekday.
- In a second step, add confidence intervals for the sales per weekday.
- In a further step, sort the weekdays from Monday to Sunday.


Durchschnittliche Umsätze je Wochentag mit Konfidenzintervallen
Durchschnittlicher Umsatz
0
50
100
150
200
250
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Wochentag