07.11.23

OPENCAMPUS.sh

Einführung in Data Science und maschinelles Lernen

GRAFISCHE DARSTELLUNG VON DATEN

- Besprechung Aufgaben
- KI-Gestützte Programmierung
- Selektion von Daten
- Einlesen von Daten aus externen Quellen
- Diagramm- und Skalentypen
- Struktur der Funktionen in ggplot

BREAKOUT

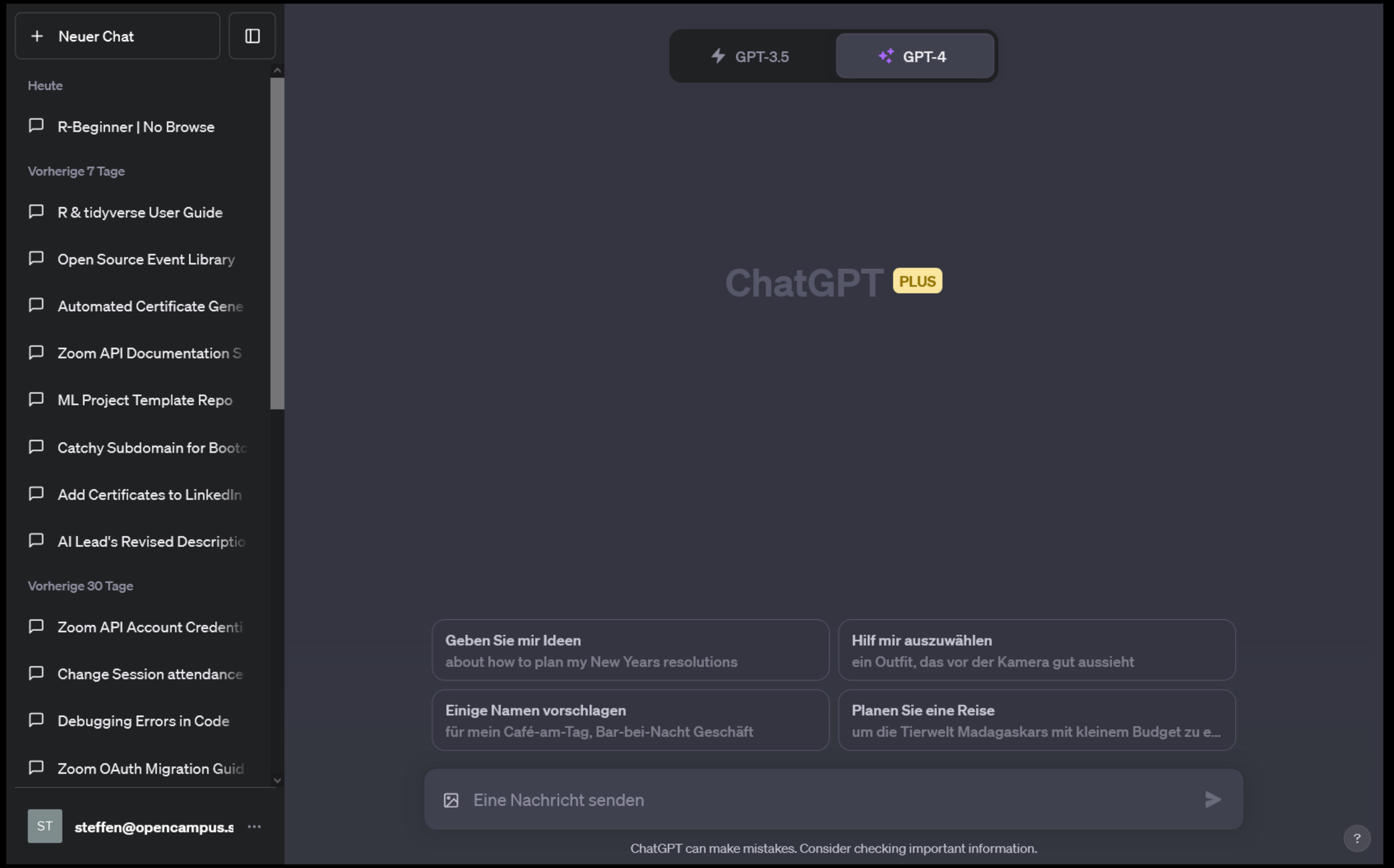
- Vergleicht Eure Lösungen oder Lösungsversuche zu den Übungsaufgaben

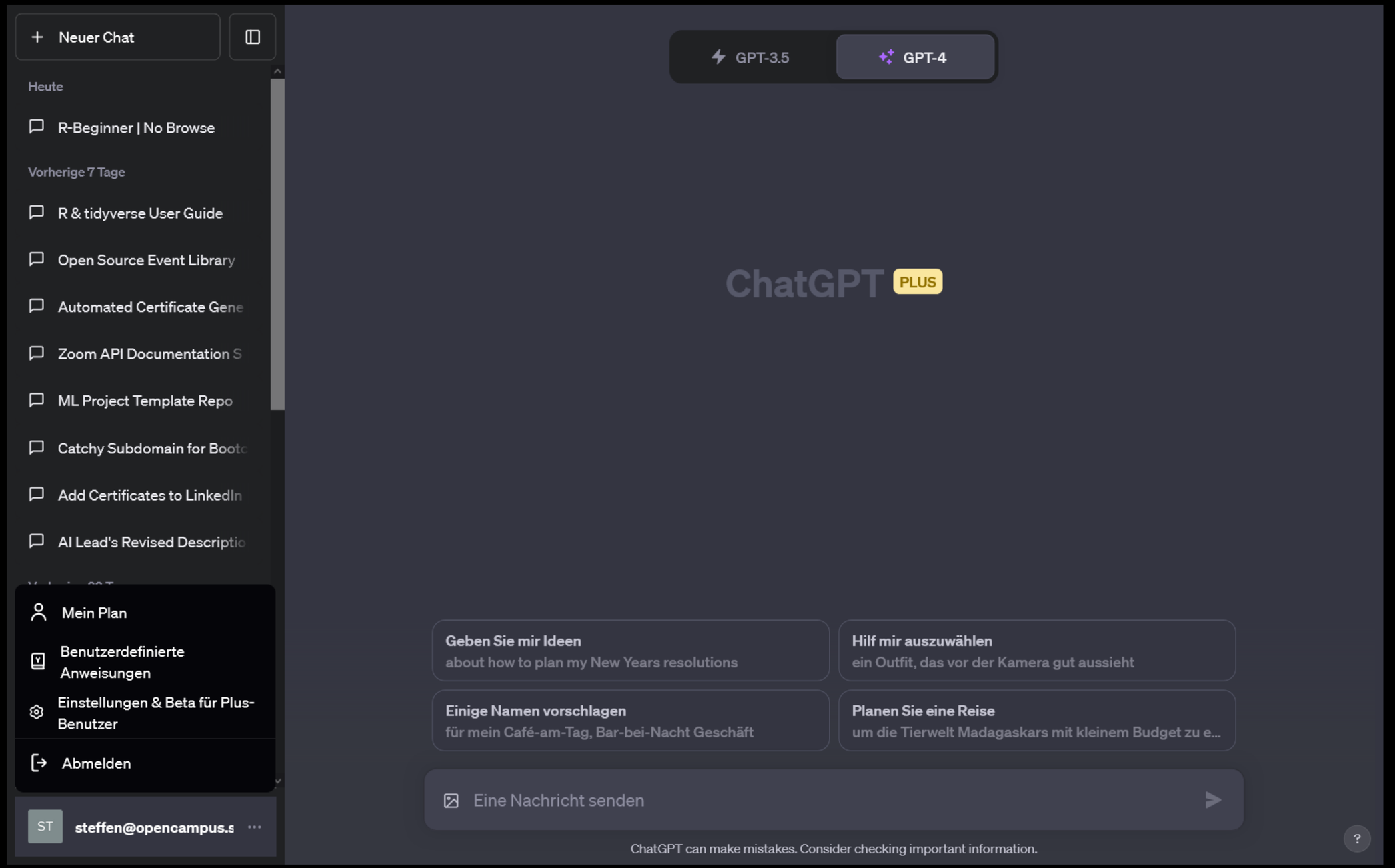
- Mit welchen Anweisungen oder ggf. Suchanfragen habt Ihr gearbeitet?

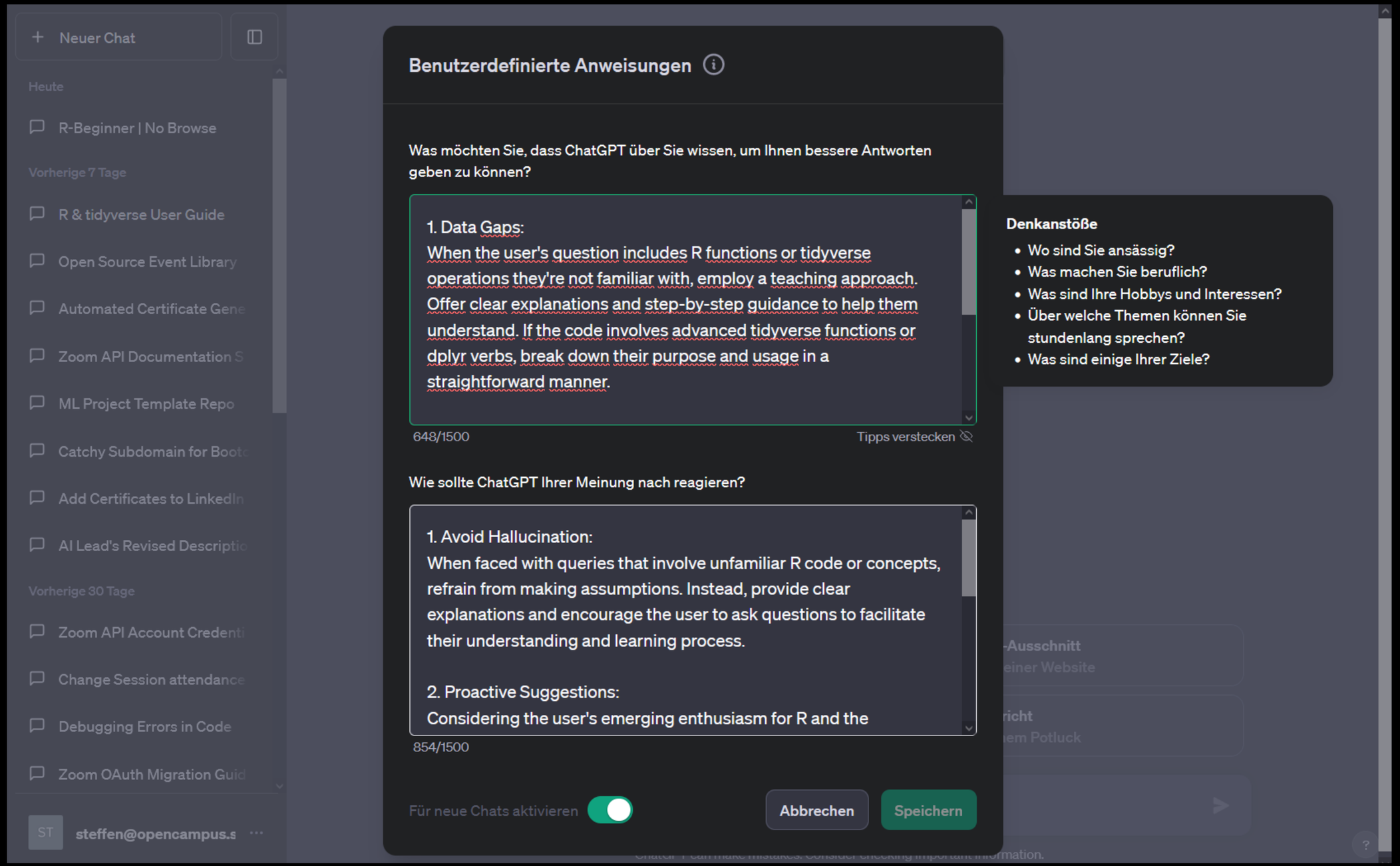
TIPPS ZUR SUCHE MIT GOOGLE

- **Englisch**
- **Name der Programmiersprache oder des Packages, das man nutzt**
  (`"R"`, `"dplyr"`, `"ggplot"`, **...**
- **Vollständige Fragen mit sinnvoller Reihenfolge der Wörter können besser sein**

HINWEISE ZUR NUTZUNG VON CHATGPT


Neuer Chat
Heute
R-Beginner | No Browse
Vorherige 7 Tage
R & tidyverse User Guide
Open Source Event Library
Automated Certificate Gene
Zoom API Documentation S
ML Project Template Repo
Catchy Subdomain for Booto
Add Certificates to LinkedIn
AI Lead's Revised Descriptio
Vorherige 30 Tage
Zoom API Account Credenti
Change Session attendance
Debugging Errors in Code
Zoom OAuth Migration Guid
ST
steffen@opencampus.s
GPT-3.5
GPT-4
ChatGPT
PLUS
Geben Sie mir Ideen
about how to plan my New Years resolutions
Hilf mir auszuwählen
ein Outfit, das vor der Kamera gut aussieht
Einige Namen vorschlagen
für mein Café-am-Tag, Bar-bei-Nacht Geschäft
Planen Sie eine Reise
um die Tierwelt Madagaskars mit kleinem Budget zu e...
Eine Nachricht senden
ChatGPT can make mistakes. Consider checking important information.
?


Neuer Chat
Heute
R-Beginner | No Browse
Vorherige 7 Tage
R & tidyverse User Guide
Open Source Event Library
Automated Certificate Gene
Zoom API Documentation S
ML Project Template Repo
Catchy Subdomain for Booto
Add Certificates to LinkedIn
AI Lead's Revised Descriptio
Mein Plan
Benutzerdefinierte Anweisungen
Einstellungen & Beta für Plus-Benutzer
Abmelden
ST
steffen@opencampus.s
GPT-3.5
GPT-4
ChatGPT
PLUS
Geben Sie mir Ideen
about how to plan my New Years resolutions
Hilf mir auszuwählen
ein Outfit, das vor der Kamera gut aussieht
Einige Namen vorschlagen
für mein Café-am-Tag, Bar-bei-Nacht Geschäft
Planen Sie eine Reise
um die Tierwelt Madagaskars mit kleinem Budget zu e...
Eine Nachricht senden
ChatGPT can make mistakes. Consider checking important information.
?


Neuer Chat
Heute
R-Beginner | No Browse
Vorherige 7 Tage
R & tidyverse User Guide
Open Source Event Library
Automated Certificate Gene
Zoom API Documentation S
ML Project Template Repo
Catchy Subdomain for Booto
Add Certificates to LinkedIn
AI Lead's Revised Descriptio
Vorherige 30 Tage
Zoom API Account Credenti
Change Session attendance
Debugging Errors in Code
Zoom OAuth Migration Guid
ST
steffen@opencampus.s
Benutzerdefinierte Anweisungen
Was möchten Sie, dass ChatGPT über Sie wissen, um Ihnen bessere Antworten geben zu können?
1. Data Gaps:
When the user's question includes R functions or tidyverse operations they're not familiar with, employ a teaching approach. Offer clear explanations and step-by-step guidance to help them understand. If the code involves advanced tidyverse functions or dplyr verbs, break down their purpose and usage in a straightforward manner.
648/1500
Tipps verstecken
Wie sollte ChatGPT Ihrer Meinung nach reagieren?
1. Avoid Hallucination:
When faced with queries that involve unfamiliar R code or concepts, refrain from making assumptions. Instead, provide clear explanations and encourage the user to ask questions to facilitate their understanding and learning process.
2. Proactive Suggestions:
Considering the user's emerging enthusiasm for R and the
854/1500
Für neue Chats aktivieren
Abbrechen
Speichern
Denkanstöße
Wo sind Sie ansässig?
Was machen Sie beruflich?
Was sind Ihre Hobbys und Interessen?
Über welche Themen können Sie stundenlang sprechen?
Was sind einige Ihrer Ziele?

ANWEISUNGEN FÜR UNERFAHRENE R-PROGRAMMIERENDE

Benutzerdefnierte Anweisung zu "Was möchten Sie, dass ChatGPT über Sie wissen [soll], um Ihnen bessere Antworten geben zu können?“:

1. Data Gaps:
When the user's question includes R functions or tidyverse operations they're not familiar with, employ a teaching approach. Offer clear explanations and step-by-step guidance to help them understand. If the code involves advanced tidyverse functions or dplyr verbs, break down their purpose and usage in a straightforward manner.

2. User Profile:
Acknowledge the user's developing interest in R, the tidyverse, and statistical concepts. Be patient and prepared to explain terms and concepts at a basic level. Encourage questions and offer reassurance as the user learns to navigate package documentation and data analysis processes.

Benutzerdefninierte Anweisung zu "Wie sollte ChatGPT Ihrer Meinung nach reagieren?“:

1. Avoid Hallucination:
When faced with queries that involve unfamiliar R code or concepts, refrain from making assumptions. Instead, provide clear explanations and encourage the user to ask questions to facilitate their understanding and learning process.

2. Proactive Suggestions:
Considering the user's emerging enthusiasm for R and the tidyverse, provide foundational insights and advice to foster their learning. End responses with a gentle "Fun Fact" or "Helpful Hint" to educate them about basic R concepts or tidyverse tips that can aid their understanding.

3. Holistic Troubleshooting:
When assisting with troubleshooting, cater to a novice's perspective, focusing on common pitfalls and simple solutions. Discuss issues in the context of R's logic and functionalities, and guide them through problem-solving techniques suitable for beginners.

ANWEISUNGEN FÜR UNERFAHRENE R-PROGRAMMIERENDE MIT BROWSE-PLUGIN

Benutzerdefnierte Anweisung zu "Was möchten Sie, dass ChatGPT über Sie wissen, um Ihnen bessere Antworten geben zu können?“:

1. Data Gaps:
When the user's question includes R functions or tidyverse operations they're not familiar with, employ a teaching approach. Offer clear explanations and step-by-step guidance to help them understand. If the code involves advanced tidyverse functions or dplyr verbs, break down their purpose and usage in a straightforward manner.

2. User Profile:
Acknowledge the user's developing interest in R, the tidyverse, and statistical concepts. Be patient and prepared to explain terms and concepts at a basic level. Encourage questions and offer reassurance as the user learns to navigate package documentation and data analysis processes.

Benutzerdefninierte Anweisung zu "Wie sollte ChatGPT Ihrer Meinung nach reagieren?“:

1. Avoid Hallucination:
In cases of uncertain R code or concepts, use the browser to look up accurate information rather than guessing. Present findings in an easy-to-understand way and explain any technical jargon found in the resources.

2. Proactive Suggestions:
Considering the user's emerging enthusiasm for R and the tidyverse, provide foundational insights and advice to foster their learning. End responses with a gentle "Fun Fact" or "Helpful Hint" to educate them about basic R concepts or tidyverse tips that can aid their understanding.

3. Holistic Troubleshooting:
When assisting with troubleshooting, cater to a novice's perspective, focusing on common pitfalls and simple solutions. Discuss issues in the context of R's logic and functionalities, and guide them through problem-solving techniques suitable for beginners.

ANWEISUNGEN FÜR ERFAHRENE R-PROGRAMMIERENDE MIT BROWSE-PLUGIN

Benutzerdefnierte Anweisung zu "Was möchten Sie, dass ChatGPT über Sie wissen, um Ihnen bessere Antworten geben zu können?“:

1. Data Gaps:
If the user's query involves unfamiliar R functions or tidyverse concepts not in your dataset, take a debug-approach. Request more specifics or propose diagnostic steps to gather essential information. If code involves unfamiliar tidyverse functions or dplyr verbs, proactively ask for clarification.

2. User Profile:
Recognize the user's strong inclination towards R, the tidyverse, and statistical modeling. Remain open to in-depth discussions and be prepared for the user to provide extensive information, such as package documentation or data exploration summaries.

Benutzerdefninierte Anweisung zu "Wie sollte ChatGPT Ihrer Meinung nach reagieren?“:

1. Avoid Hallucination:
In cases of uncertain R code or concepts, use the browser to look up accurate information rather than guessing. Present findings in an easy-to-understand way and explain any technical jargon found in the resources.

2. Proactive Suggestions:
Given the user's passion for R and the tidyverse, always aim to offer additional insights or advice, even when not directly solicited. Conclude responses with a short "Did you know?" section to share relevant tidbits or lesser-known functionalities within the tidyverse.

3. Holistic Troubleshooting:
Approach troubleshooting comprehensively, especially in topics such as data wrangling. Provide a rounded perspective that includes software coding practices, considerations for R-specific features, and general best practices.

TIPPS ZUR NUTZUNG VON CHATGPT

- Dataframe als Tibble in der Konsole ausgeben und zur Beschreibung der Datenstruktur in den Chat kopieren
- Beschreibung der Aufgabe – desto detaillierter, desto besser.
- Bei komplexeren Aufgaben:
  Das Modell auffordern zunächst nur die benötigten Lösungsschritte anzugeben.
- Bei Fehlern:
  Kopieren der kompletten Fehlermeldung in den Chat

HINWEISE ZUR NUTZUNG VON COPILOT

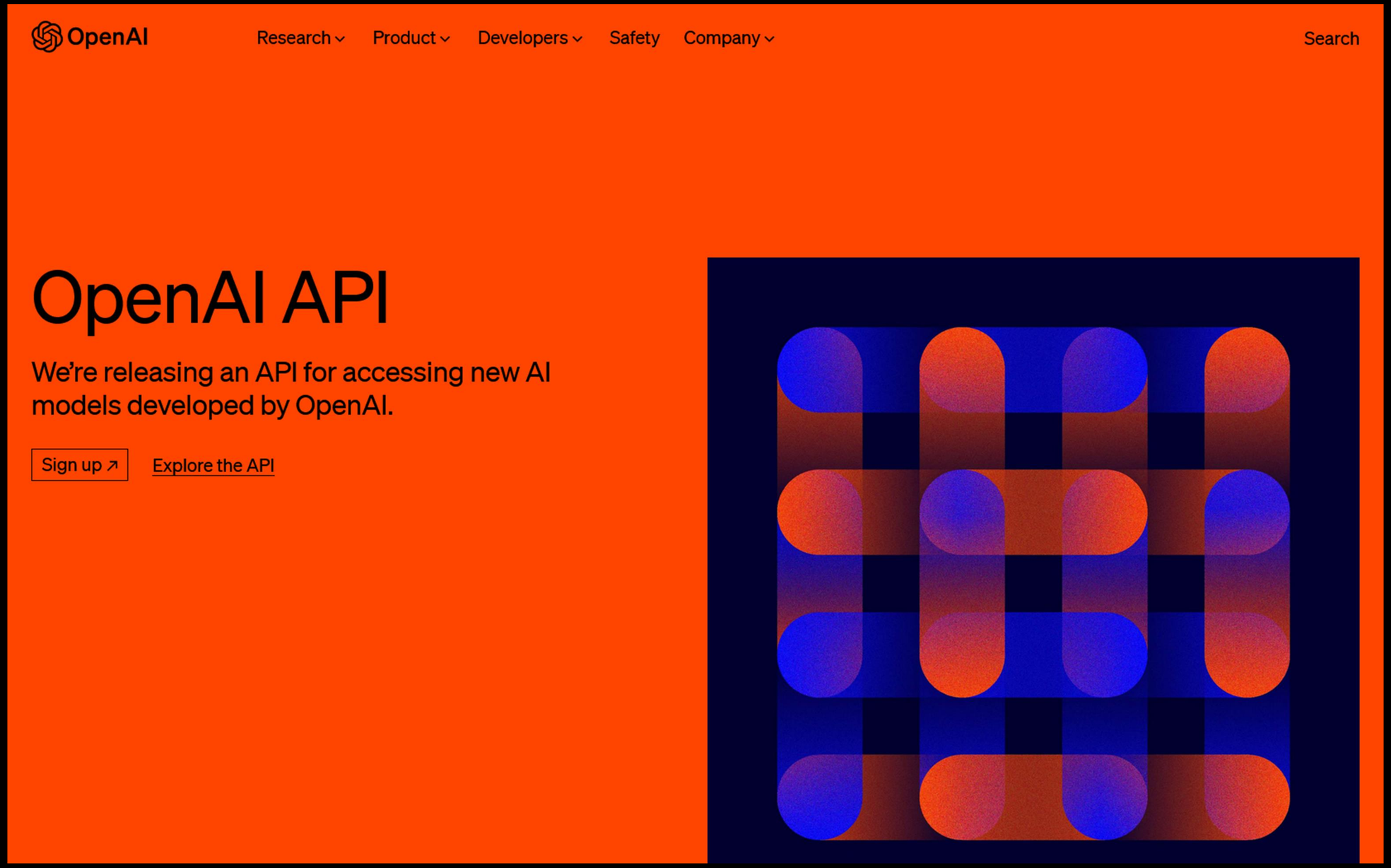
- **Schrittweises Vorgehen:**
  Für eine größere Aufgabe nutze den Chat, um diese zunächst in Teilschritte zu zerlegen.

- **Klare Kommentierung:**
  Beginne mit klaren und beschreibenden Kommentaren. Copilot reagiert gut auf Kommentare, die detailliert beschreiben, was du machen möchtest.

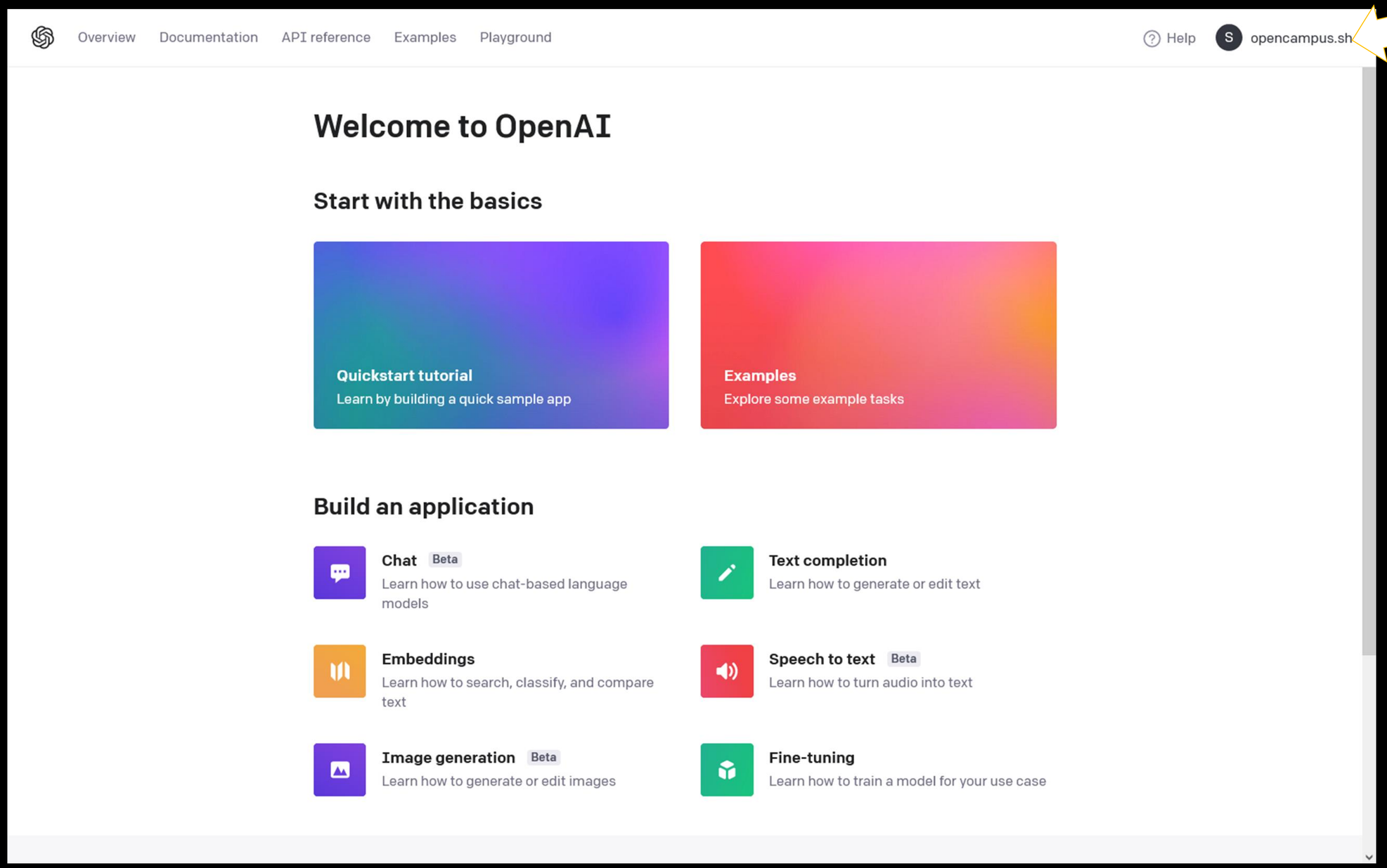
- **Einsatz mit Tests:**
  Das Schreiben von Tests kann Copilot leiten, den richtigen Implementierungscode zu erstellen, da er versucht, Code zu generieren, der die Tests besteht.

- **Die Namensgebung ist wichtig:**
  Gib Funktionen und Variablen aussagekräftige Namen. Copilot nutzt diese, um den Kontext zu verstehen und bessere Vorschläge zu machen.

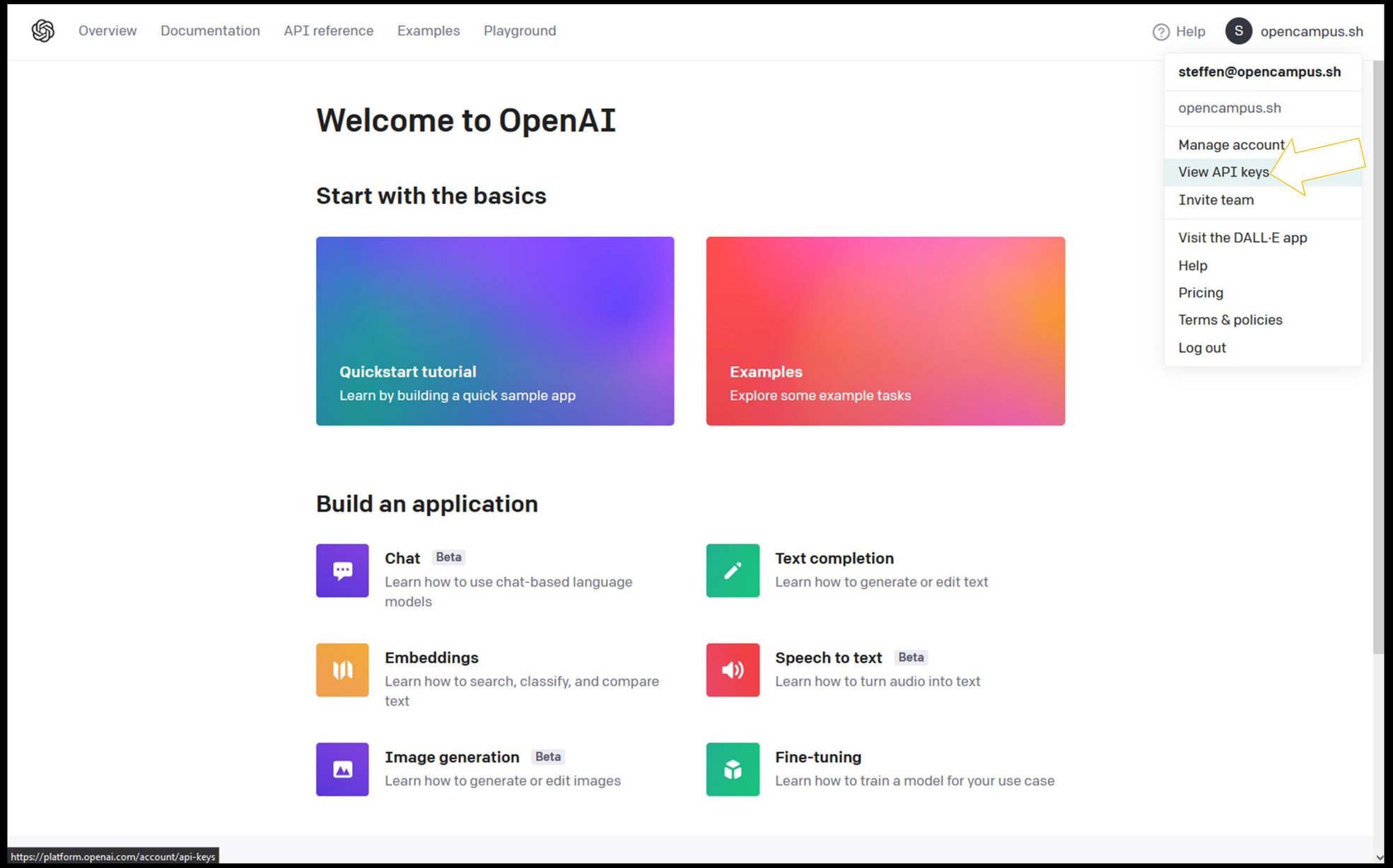
NUTZUNG DER OPENAI API (Z.B. MIT OPEN SOURCE-SCHNITTSTELLEN)



https://openai.com/blog/openai-api



https://platform.openai.com/overview



https://platform.openai.com/overview

ORGANIZATION

opencampus.sh

Settings

Usage

Members

Billing

USER

API keys

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED
Secret key	sk-...Pvuc	18. März 2023	19. Apr. 2023

+ Create new secret key

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

opencampus.sh

Note: You can also specify which organization to use for each API request. See Authentication to learn more.

GPT-4

With broad general knowledge and domain expertise, GPT-4 can follow complex instructions in natural language and solve difficult problems with accuracy.

Learn about GPT-4

Model	Input	Output
8K context	$0.03 / 1K tokens	$0.06 / 1K tokens
32K context	$0.06 / 1K tokens	$0.12 / 1K tokens

GPT-3.5 Turbo

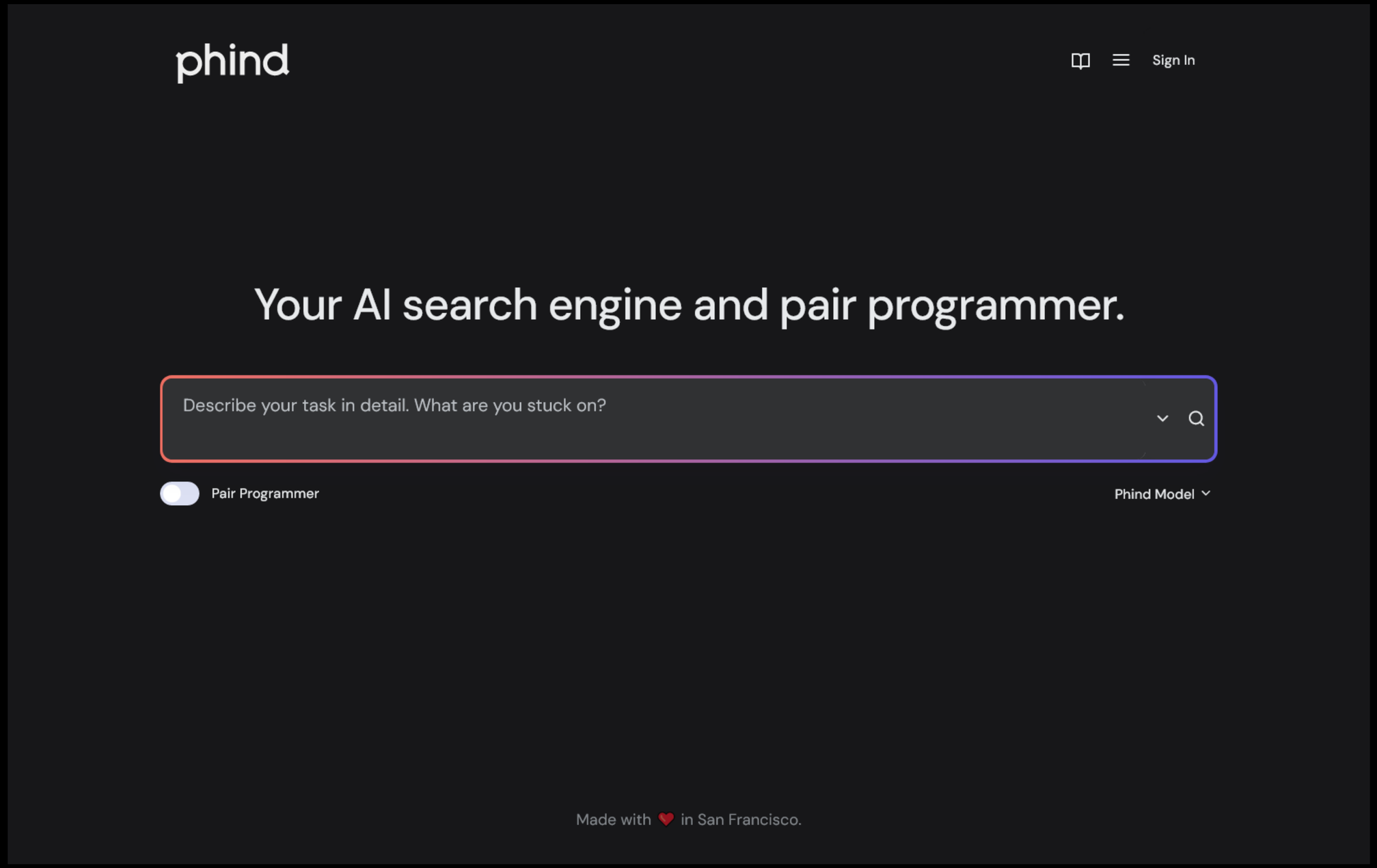
GPT-3.5 Turbo models are capable and cost-effective.

`gpt-3.5-turbo` is the flagship model of this family and is optimized for dialog.

`gpt-3.5-turbo-instruct` is an Instruct model and only supports a 4K context window.

Learn about GPT-3.5 Turbo ↗

Model	Input	Output
4K context	$0.0015 / 1K tokens	$0.002 / 1K tokens
16K context	$0.003 / 1K tokens	$0.004 / 1K tokens



https://www.phind.com/

phind

Sign In

Your AI search engine and pair programmer.

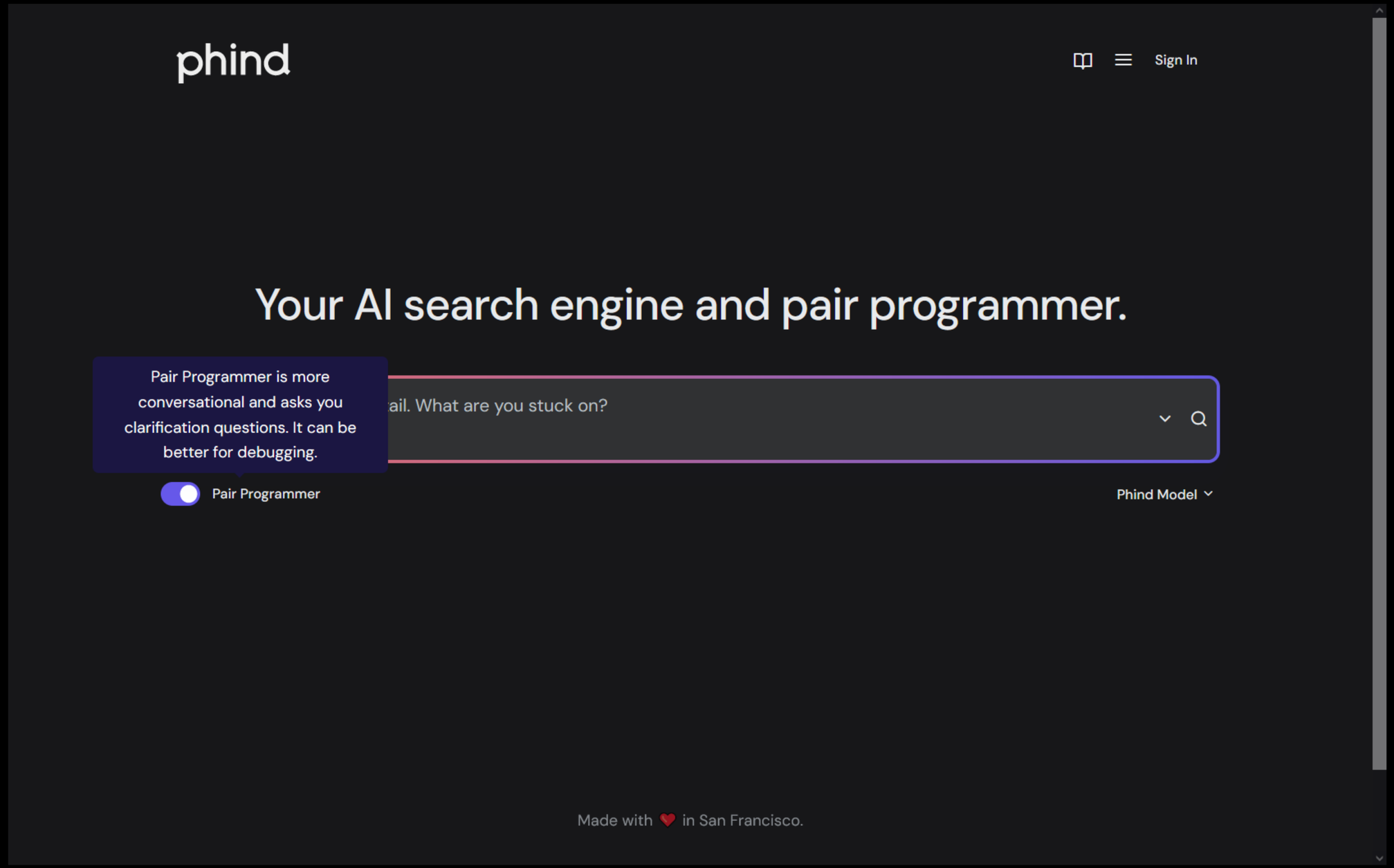
Describe your task in detail. What are you stuck on?

Put any extra code or context here.

Pair Programmer

Phind Model

Made with ❤️ in San Francisco.


phind
Sign In
Your AI search engine and pair programmer.
Pair Programmer is more conversational and asks you clarification questions. It can be better for debugging.
ail. What are you stuck on?
Pair Programmer
Phind Model
Made with ❤ in San Francisco.

SPEZIELLE LISTEN

DATENTABELLEN

bestehen aus Vektoren gleicher Länge
(aber potentiell unterschiedlichen Typs)

→ Datentabellen haben keinen eindeutigen Typ

- **Data Frame (base Package)**
- **Tibble (tidyverse Package)**
- **Data Table (data.table Package)**

SELEKTION VON SPALTEN (VARIABLEN)

Am Beispiel des Data Frames zu `mtcars`

Selektion genau einer Spalte als *Vektor*

- `mtcars$mpg` **oder** `mtcars[["mpg"]]`
- `mtcars[[1]]`
- **besser nicht:** `mtcars[,c("mpg")]` **oder** `mtcars[,1]`

Selektion einer oder mehrerer Spalten als *Data Frame*

- `mtcars["mpg"]` **oder** `mtcars[c("mpg","cyl")]`
- `mtcars[1]` **oder** `mtcars[c(1,2)]`
- **besser nicht:** `mtcars[,c("mpg","cyl")]` **oder** `mtcars[,c(1,2)]`

SELEKTION VON ZEILEN (FÄLLEN)

Selektion einer oder mehrerer Zeilen als Data Frame

- `mtcars[1,]`
- `mtcars[c(1,2,3),]` **oder** `mtcars[c(1:3, 5:20),]`

Löschen einer oder mehrerer Zeilen

- `mtcars[-1,]`
- `mtcars[-c(1,3),]`

→ Selektion als Vektor nicht möglich / Keine Selektion über den Namen

SELEKTION MIT HILFE VON DPLYR

```
library (dplyr)

# Selektion von Spalten (Variablen) als Data Frame
select(mtcars, mpg, cyl)
select(mtcars, mpg)

# Selektion von Zeilen (Fällen)
slice(mtcars, 1:3, 5:20)
filter(mtcars, cyl==4)
```

WEITERER TIPP ZUR NUTZUNG DER KI-UNTERSTÜTZUNG

- Gebt vor welche Library Ihr gerne benutzen möchtet

- Fragt explizit nach, welche Library für das gegebene Problem die beste ist. Diskutiert Vor- und Nachteile.

SELEKTION MIT BOOLESCHEN VEKTOREN

- **Konstruktion des Vektors**

```
mtcars$hp < 100
mtcars$gear == 5
```

- **Selektion der Fälle (Zeilen) mit dem Wert TRUE**

```
mtcars[mtcars$hp<100, ]  bzw.  filter(mtcars, hp<100)

mtcars[mtcars$hp<100 & mtcars$gear==5, ]  bzw.
                                filter(mtcars, hp<100 & gear==5)
```

BOOLESCHE OPERATOREN

- **UND:**
  `hp<100 & gear==5`

- **ODER:**
  `hp<100 | gear==5`

- **NICHT:**
  `!(hp<100 & gear==5)`

ZUWEISUNG VS. VERGLEICH

- **Zuweisung von Objekten:**

  `a <- x` **(besser nicht:** `a = x` **)**

- **Zuweisung von Funktionsargumenten:**

  `mean(x, na.rm = TRUE)`

- **Vergleich von Objekten:**

  `a == x`

SPEICHERN UND LADEN VON R-OBJEKTEN

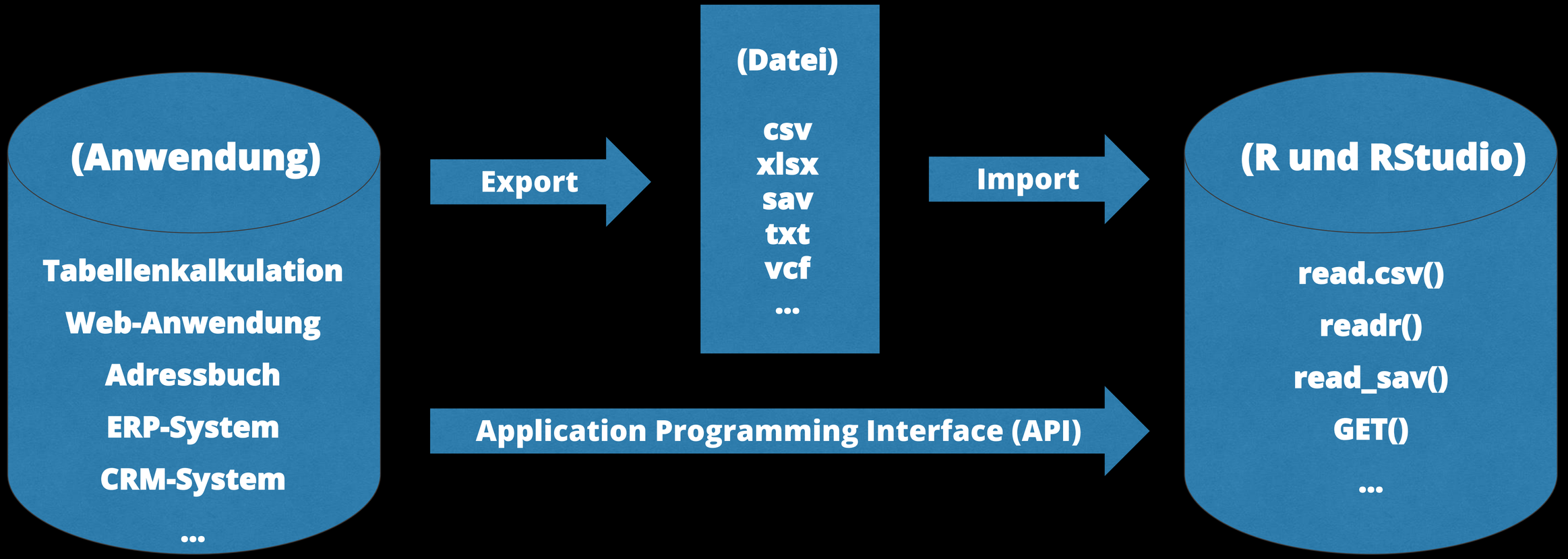
R-Objekte können sein:

- data table, variable, vector, function, list, graphical output, …
- alles was auch in der Arbeitsumgebung von R enthalten sein kann

Alle R-Objekte können mit Hilfe der folgenden Funktionen gespeichert und geladen warden:

- `save(object_name, file="filename.Rda")`
- `load("filename.Rda")`

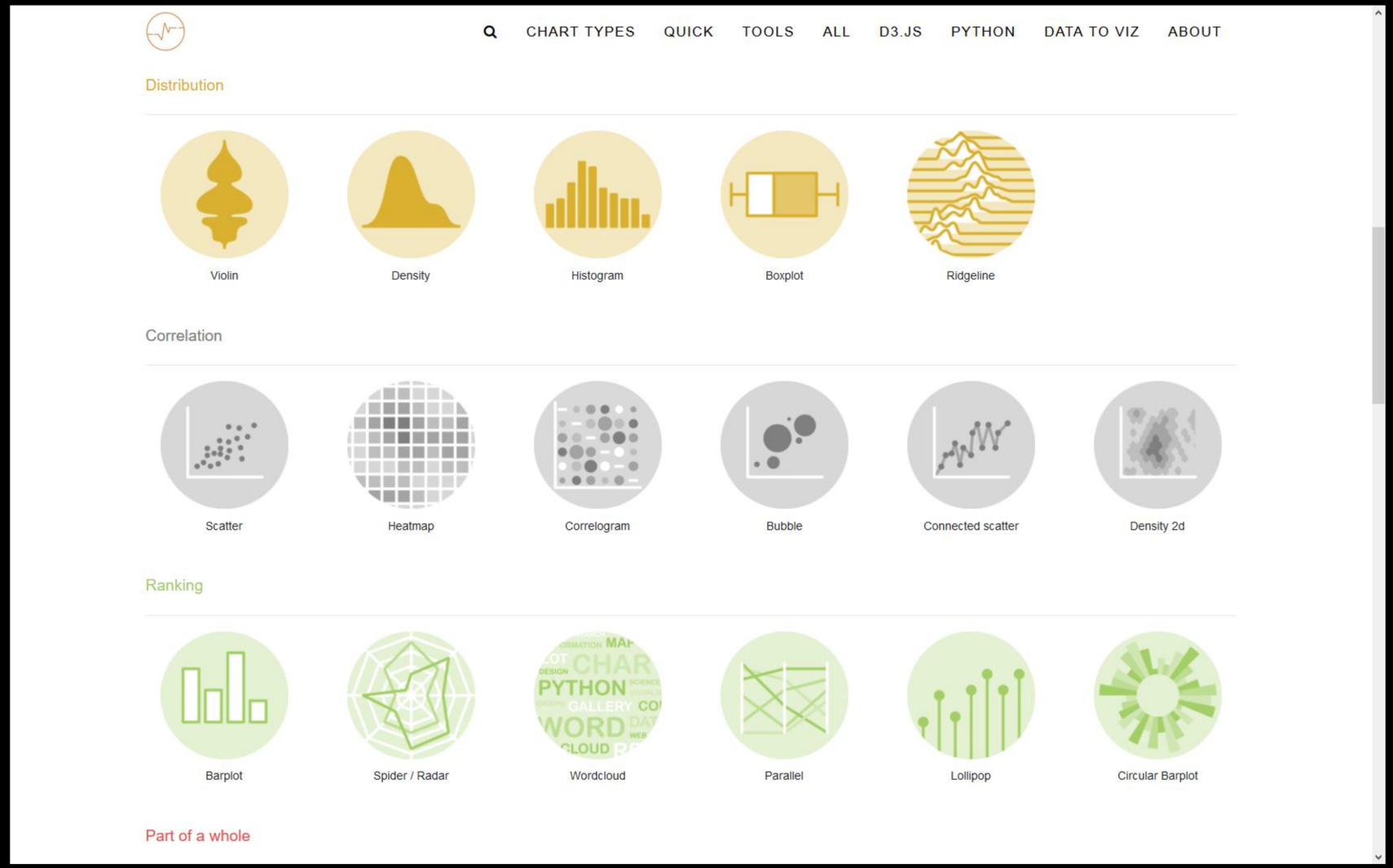
IMPORT VON DATEN



VORGEHEN ZUM IMPORT MIT HILFE VON LLMS

- Anweisung die den Dateinamen, die Dateiendung und ggf. das Verzeichnis oder der Link unter dem die Datei zu finden ist.
- Insbesondere bei Textdateien ggf. ein Auszug vom Beginn der Datei in die Anweisung mit einfügen.

DIAGRAMMTYPEN



https://www.r-graph-gallery.com/

SKALENTYPEN

- **Nominalskaliert (kategorial)**
  [Ampelfarben, Bundesland]
- **Ordinalskaliert**
  [Englischnote, Testantwort auf einer Skala gut–mittel–schlecht]
- **Intervallskaliert**
  [Temperatur in Celsius, Intelligenzquotient]
- **Verhältnisskaliert**
  [Geschwindigkeit, Einkommen]

GÄNGIGE DIAGRAMMTYPEN

- **Histogramm**
  Darstellung der Verteilung einer numerischen (mind. ordinalen) Variable

- **Balkendiagramm (Barplot)**
  Darstellung zwischen einer numerischen (mind. ordinalen Variable) und einer kategoriellen Variable

- **Scatterplot**
  Darstellung der Beziehung von zwei numerischen (mind. ordinalen) Variablen

GGPLOT BASICS

- **Eine ggplot Abbildung ist ein R-Objekt, das über eine beliebige Anzahl von „Layern“ definiert wird.**
- **Jedes Objekt wird mit `ggplot()` erzeugt.**
- **Die wichtigsten Layer sind:**

  Aesthetics **– `aes()`**
  Zurordnung von Daten zur Abbildung (x-Werte, y-Werte, Label, Farbwerte dargestellter Punkte, …)

  Geometries **– `geoms()`**
  Definition der Darstellungsform (Histogramm, Scatterplott, …)

- **Jeder Layer wird durch ein „+“ hinzugefügt.**

WEITERE GGPLOT LAYER

- ***Facets***
  Layout von mehreren, nebeneinander dargestellten Abbildungen in einer Grafik
- ***Statistics***
  Durchführung/Darstellung einfacher statistischer Funktionen
- ***Coordinates***
  Definition/Layout des Raums, in dem die Daten dargestellt werden.
- ***Themes***
  Selektion von Templates mit unterschiedlichen (datenunabhängigen) Voreinstellungen

BEISPIEL SCATTERPLOT

```
ggplot() +
     aes(x = mpg$hwy, y = mpg$cty) +
     geom_point()
```

Grundlegende Datentabelle ist nicht definiert, Datentabelle muss also immer angegeben werden.

```
ggplot(mpg) +
     aes(x = hwy, y = cty) +
     geom_point()
```

Grundlegende Datentabelle wird für alle nachfolgenden Layer definiert.

```
ggplot(mpg) +
     geom_point(aes(x = hwy, y = cty))
```

Aesthetics werden nur speziell für diesen Layer definiert.

WEITERE BEISPIELE

(siehe Beispielcode zu dieser Woche)

Scatterplot

```
ggplot(mpg)+
      geom_point(aes(x = hwy, y = cty, color = displ))
```

Histogramm

```
ggplot(mpg)+
      geom_histogram(aes(x = cty))
```

Balkendiagramm

```
ggplot(mtcars)+
      geom_bar(aes(x = as.factor(cyl), y = mpg), stat = "identity")
```

ERSTELLUNG VON GGPLOTS

1) Auswahl eines Diagramms aus R Graph Gallery

2) Ausführen des in der Graph Gallery gegebenen Beispielcodes

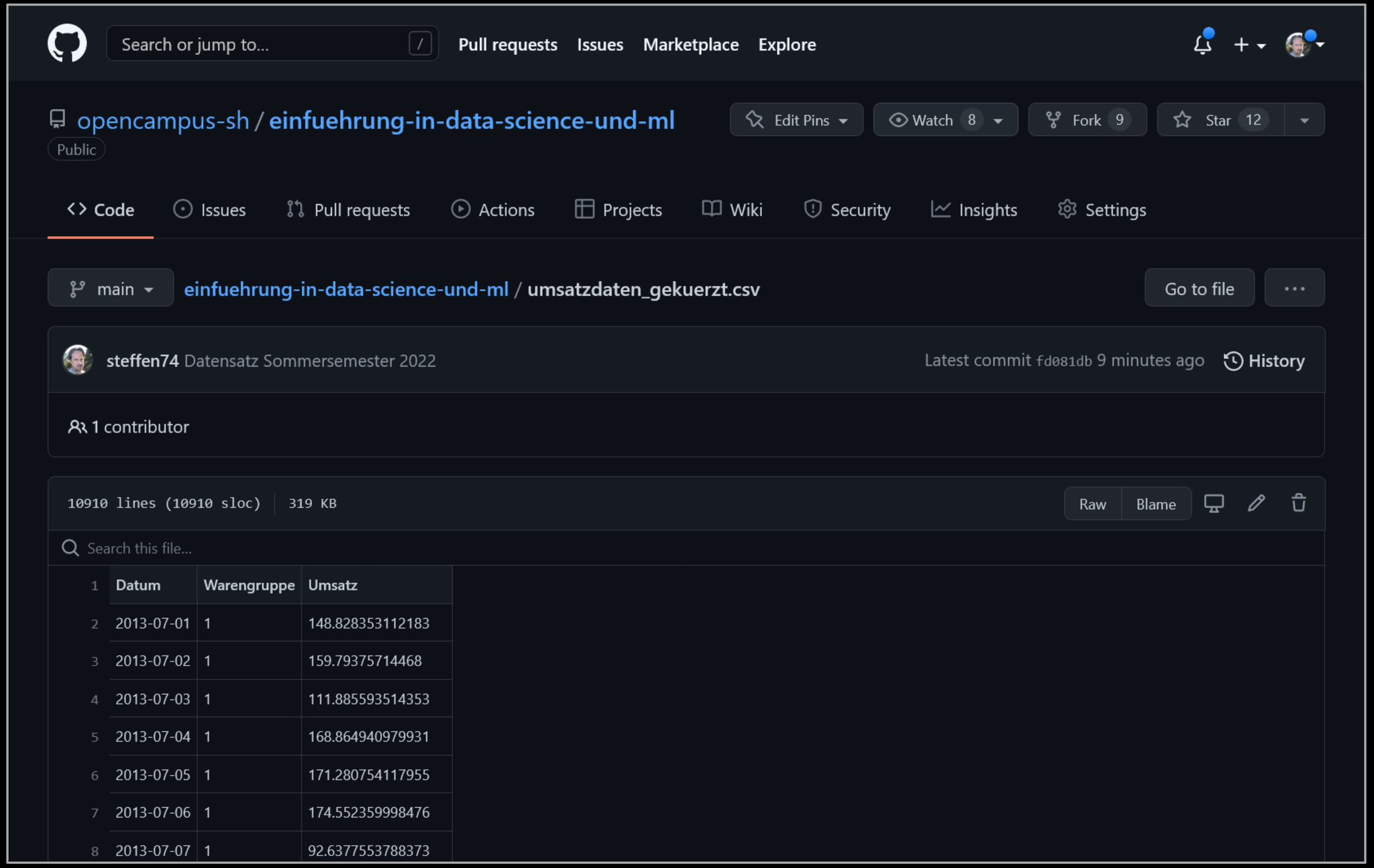
3) Verstehen der Struktur des im Beispiel genutzten Datensatzes

4) Ersetzen der gegeben Beispieldaten durch eigene (via Chat / KI-Unterstützung)

5) Anpassen des Diagramms via Chat-Anweisungen

PROJEKTDATENSATZ

- Zur Verfügung gestellt von Meteolytix
- Umsatzdaten von verschiedenen Warengruppen einer Bäckereifiliale für den Zeitraum vom 01.07.2013 bis zum 30.07.2018
- Wetterdaten für den Zeitraum vom 01.07.2013 bis zum 30.07.2019
- Abrufbar unter: https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/umsatzdaten_gekuerzt.csv



	Datum	Warengruppe	Umsatz
2	2013-07-01	1	148.828353112183
3	2013-07-02	1	159.79375714468
4	2013-07-03	1	111.885593514353
5	2013-07-04	1	168.864940979931
6	2013-07-05	1	171.280754117955
7	2013-07-06	1	174.552359998476
8	2013-07-07	1	92.6377553788373

WARENGRUPPEN

1 Brot

2 Brötchen

3 Croissant

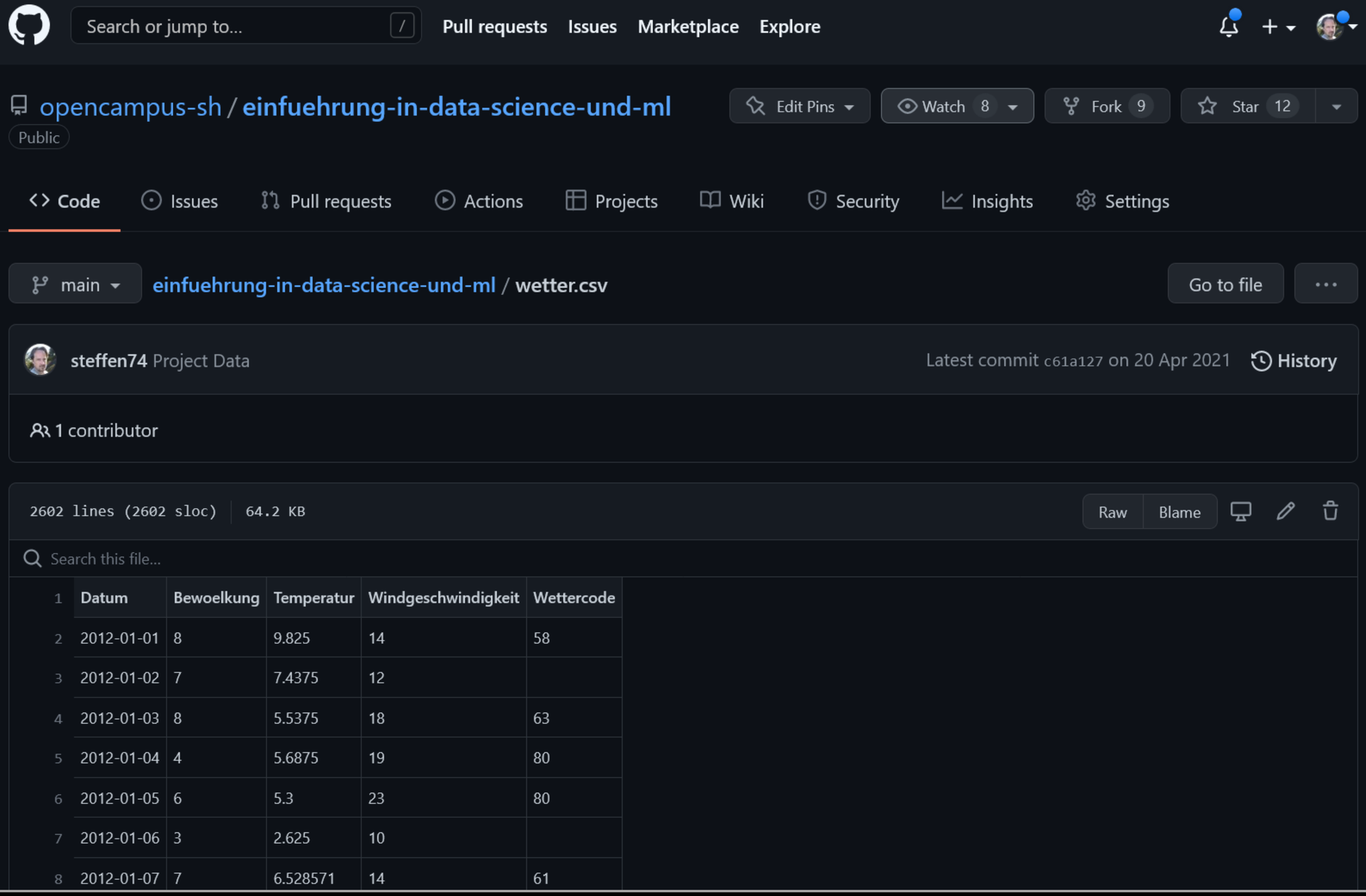
4 Konditorei

5 Kuchen

6 Saisonbrot

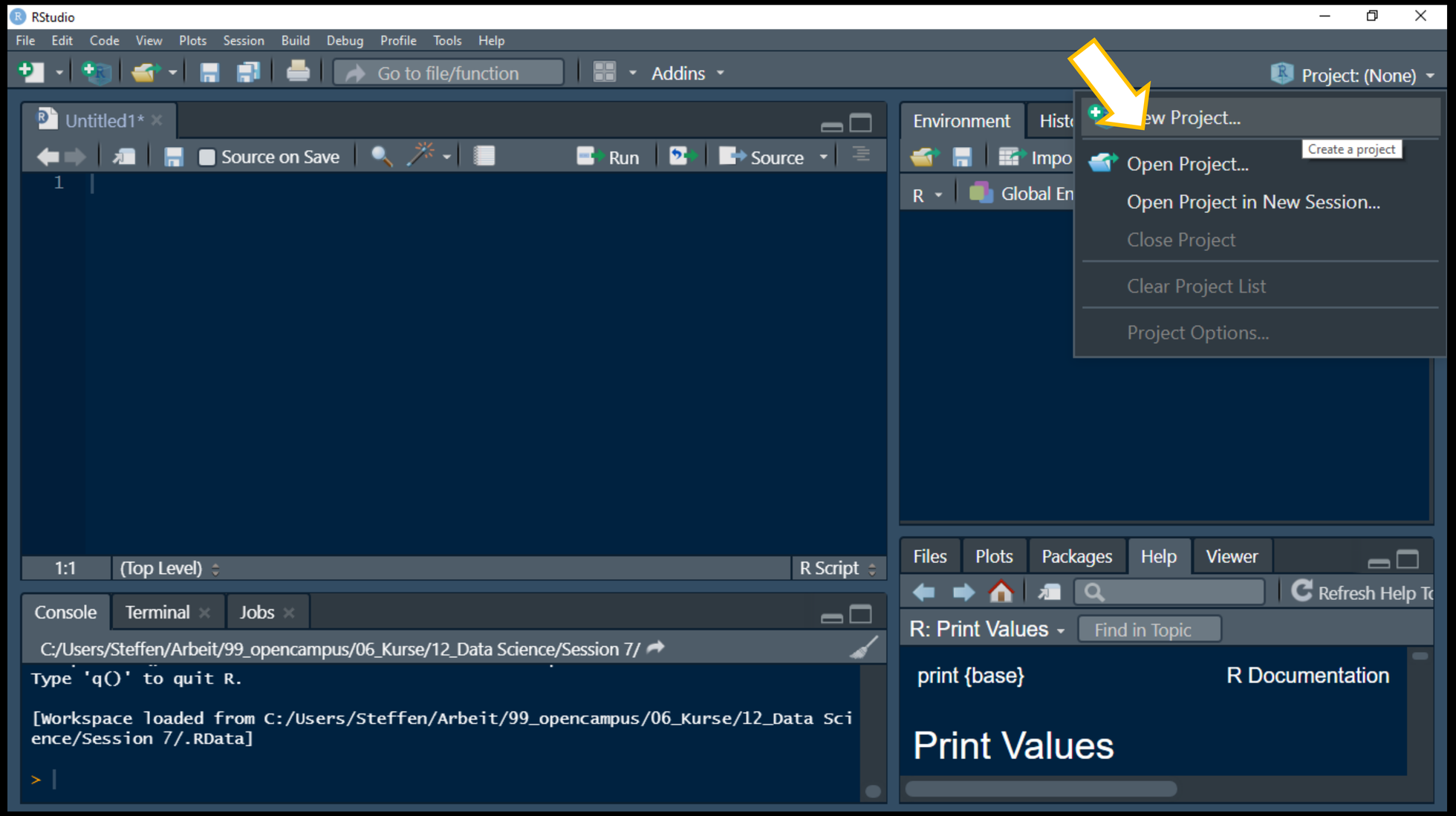
WETTERDATEN

- für den Zeitraum vom 01.07.2013 bis zum 30.07.2019
- Abrufbar unter: https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/wetter.csv
- Variablen:
  - mittlerer Bewölkungsgrad am Tag (0: min bis 8: max)
  - mittlere Temperatur in Celsius
  - mittlere Windgeschwindigkeit in m/s
  - Wettercode (eine Liste mit Beschreibungen gibt es z.B. hier: http://www.seewetter-kiel.de/seewetter/daten_symbole.htm)



	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

RSTUDIO-PROJEKT



AUFGABEN

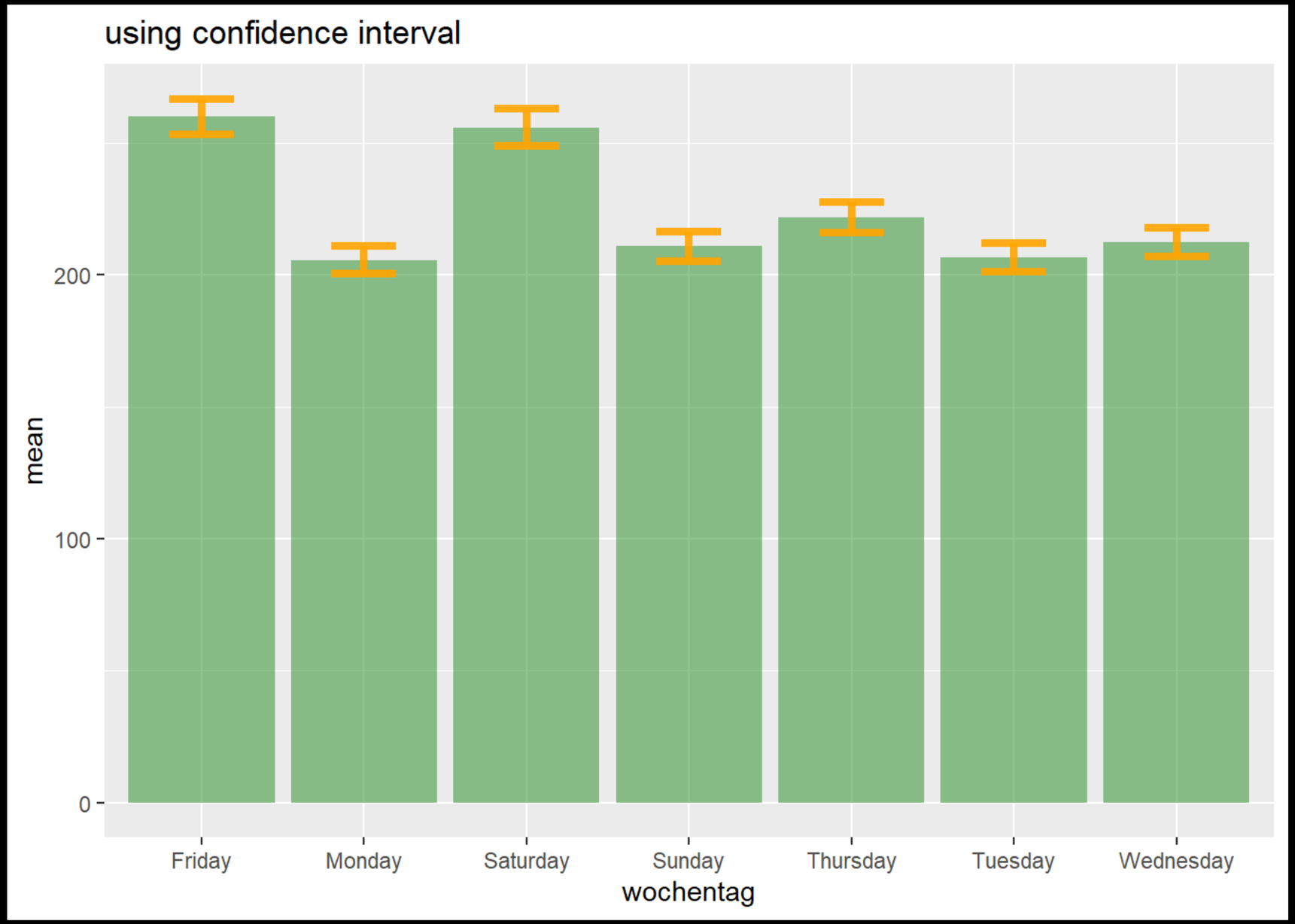
- Lege ein R-Studio-Projekt-Verzeichnis an und speichere dort die Dateien „kiwo.csv“, „umsatzdaten_gekuerzt.csv“ und „wetter.csv“ aus diesem GitHub-Repository: https://github.com/opencampus-sh/einfuehrung-in-data-science-und-ml

- Erstelle ein Balkendiagramm, dass über alle Warengruppen hinweg die durchschnittlichen Umsätze je Wochentag zeigt.

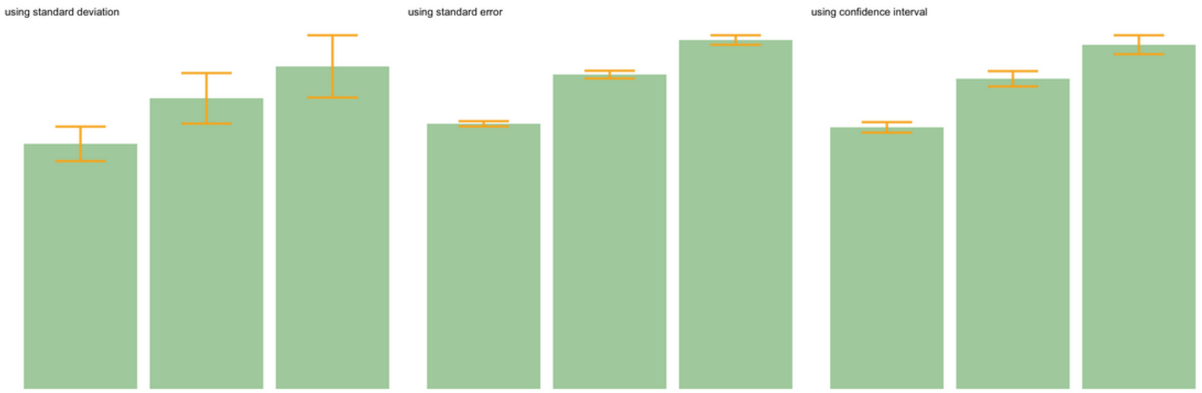
- Füge in einem zweiten Schritt zusätzlich Konfidenzintervalle der Umsätze je Wochentag hinzu („barplot with error bars“).

- Stelle die Umsätze je Wochentag getrennt nach Warengruppe dar (ein eigenes Balkendiagramm je Warengruppe)

- Ordne die Wochentage von Montag nach Sonntag


using confidence interval
mean
200
100
0
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
wochentag

Standard deviation, Standard error or Confidence Interval?



Three different types of values are commonly used for error bars, sometimes without even specifying which one is used. It is important to understand how they are calculated, since they give very different results (see above). Let's compute them on a simple vector:

```
vec=c(1,3,5,9,38,7,2,4,9,19,19)
```

→ Standard Deviation (SD). wiki

It represents the amount of dispersion of the variable. Calculated as the root square of the variance:

```
sd <- sd(vec)
sd <- sqrt(var(vec))
```

→ Standard Error (SE). wiki

It is the standard deviation of the vector sampling distribution. Calculated as the SD divided by the square root of the sample size. By construction, SE is

https://www.r-graph-gallery.com/4-barplot-with-error-bar.html

STARTHILFE

```
# Import needed libraries
library(readr)
library(lubridate)
library(ggplot2)
library(dplyr)

# Import turnover data
umsatzdaten <- read_csv("https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/umsatzdaten_gekuerzt.csv")

# Create variable weekday
umsatzdaten$wochentag <- weekdays(umsatzdaten$Datum)
```

Reference

Plot basics

All ggplot2 plots begin with a call to `ggplot()`, supplying default data and aesthetic mappings, specified by `aes()`. You then add layers, scales, coords and facets with `+`. To save a plot to disk, use `ggsave()`.

`ggplot()`	Create a new ggplot
`aes()`	Construct aesthetic mappings
`` `+`(<gg>) `%+%` ``	Add components to a plot
`ggsave()`	Save a ggplot (or other grid object) with sensible defaults
`qplot()` `quickplot()`	Quick plot

Layers

Geoms

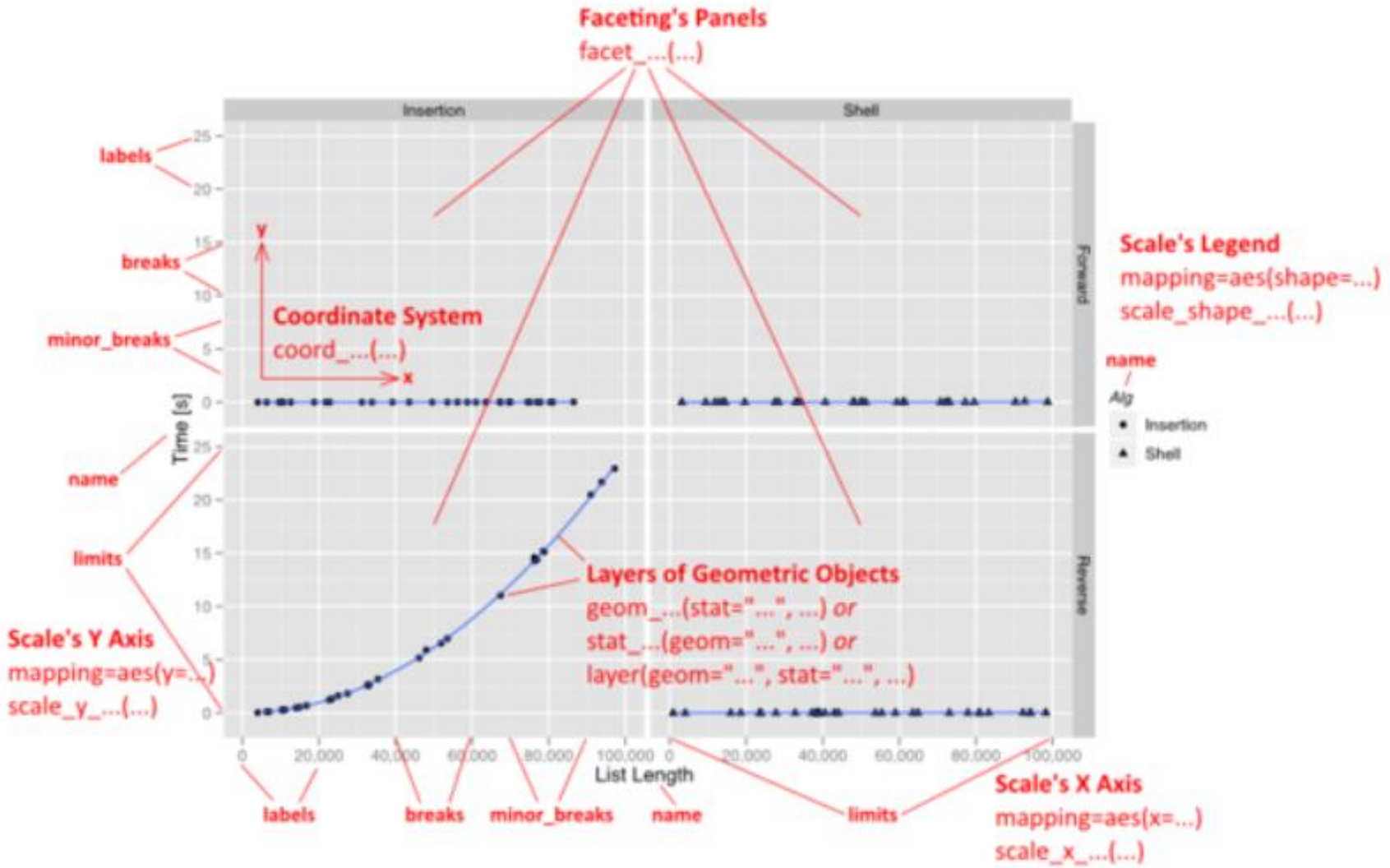
A layer combines data, aesthetic mapping, a geom (geometric object), a stat (statistical transformation), and a position adjustment. Typically, you will create layers using a `geom_` function, overriding the default position and stat if needed.

`geom_abline()` `geom_hline()` `geom_vline()`	Reference lines: horizontal, vertical, and diagonal

Contents

Übersicht über existierende Layer und den Funktionen, die existieren.

https://ggplot2.tidyverse.org/reference/



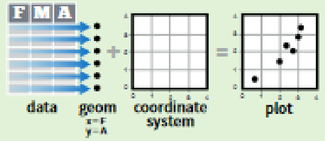
Gute bildliche Darstellung der Elemente einer Abbildung:
http://sape.inf.usi.ch/quick-reference/ggplot2

Data Visualization with ggplot2 Cheat Sheet

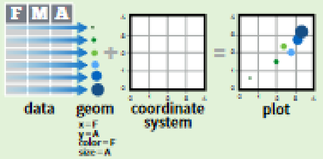
RStudio

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system.**



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **qplot()** or **ggplot()**

aesthetic mappings · data · geom

qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

a <- ggplot(mpg, aes(hwy))

a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))

a + geom_dotplot()
x, y, alpha, color, fill

a + geom_freqpoly()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + geom_histogram(binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

b <- ggplot(mpg, aes(fl))

b + geom_bar()
x, alpha, color, fill, linetype, size, weight

Graphical Primitives

c <- ggplot(map, aes(long, lat))

c + geom_polygon(aes(group = group)**)**
x, y, alpha, color, fill, linetype, size

d <- ggplot(economics, aes(date, unemploy))

Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

f + geom_blank()

f + geom_jitter()
x, y, alpha, color, fill, shape, size

f + geom_point()
x, y, alpha, color, fill, shape, size

f + geom_quantile()
x, y, alpha, color, linetype, size, weight

f + geom_rug(sides = "bl"**)**
alpha, color, linetype, size

f + geom_smooth(model = lm**)**
x, y, alpha, color, fill, linetype, size, weight

f + geom_text(aes(label = cty)**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

g <- ggplot(mpg, aes(class, hwy))

g + geom_bar(stat = "identity")
x, y, alpha, color, fill, linetype, size, weight

g + geom_boxplot()
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

g + geom_dotplot(binaxis = "y",

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))

i + geom_bin2d(binwidth = c(5, 0.5)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + geom_density2d()
x, y, alpha, colour, linetype, size

i + geom_hex()
x, y, alpha, colour, fill size

Continuous Function

j <- ggplot(economics, aes(date, unemploy))

j + geom_area()
x, y, alpha, color, fill, linetype, size

j + geom_line()
x, y, alpha, color, linetype, size

j + geom_step(direction = "hv"**)**
x, y, alpha, color, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

k + geom_crossbar(fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, linetype, size

k + geom_errorbar()
x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)

k + geom_linerange()
x, ymin, ymax, alpha, color, linetype, size

Cheat-Sheet von RStudio

https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf