Digital Applications & Data Management

WS25/26

Dr. Jens Kohl

Roadmap Vorlesung

1. Einführung und Übersicht
2. Grundlagen Data Science
3. Vorgehen Data Science Use Case
4. Case Study Data Science
5. Grundlagen unüberwachtes Lernen
6. Grundlagen überwachtes Lernen (tabellarische Daten)
7. Case Study überwachtes Lernen (tabellarische Daten)
8. Grundlagen überwachtes Lernen (Bilddaten)
9. Case Study überwachtes Lernen und Transfer Learning (Bilddaten)
10. Grundlagen Generative AI
11. Generative AI mit Texten und Prompt Engineering
12. Agentic AI
13. Ausblick: Machine Learning in der Cloud und Reinforcement Learning

Vorlesung 3:
Vorgehen Data Science Use Case

Was machen wir heute?

Motivation

Ask an interesting question

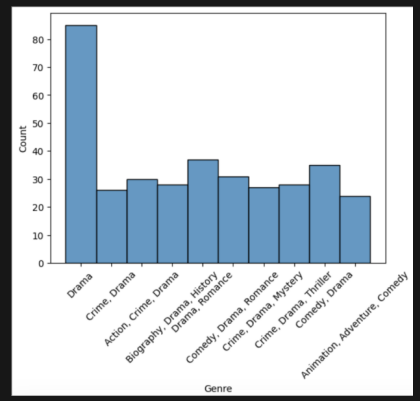
Get the Data

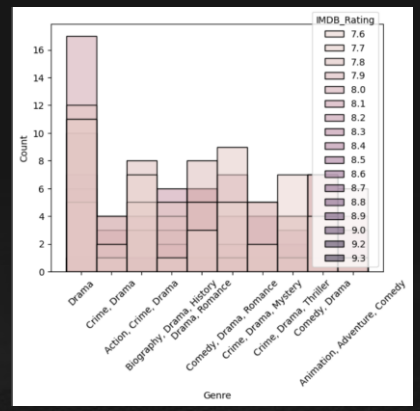
Explore the Data

Model the Data

Communicate/Visualize the Results

Generische Vorgehensweise Data Science





Standardisierte Vorgehensweise, um aus Datensätzen Erkenntnisse zu gewinnen

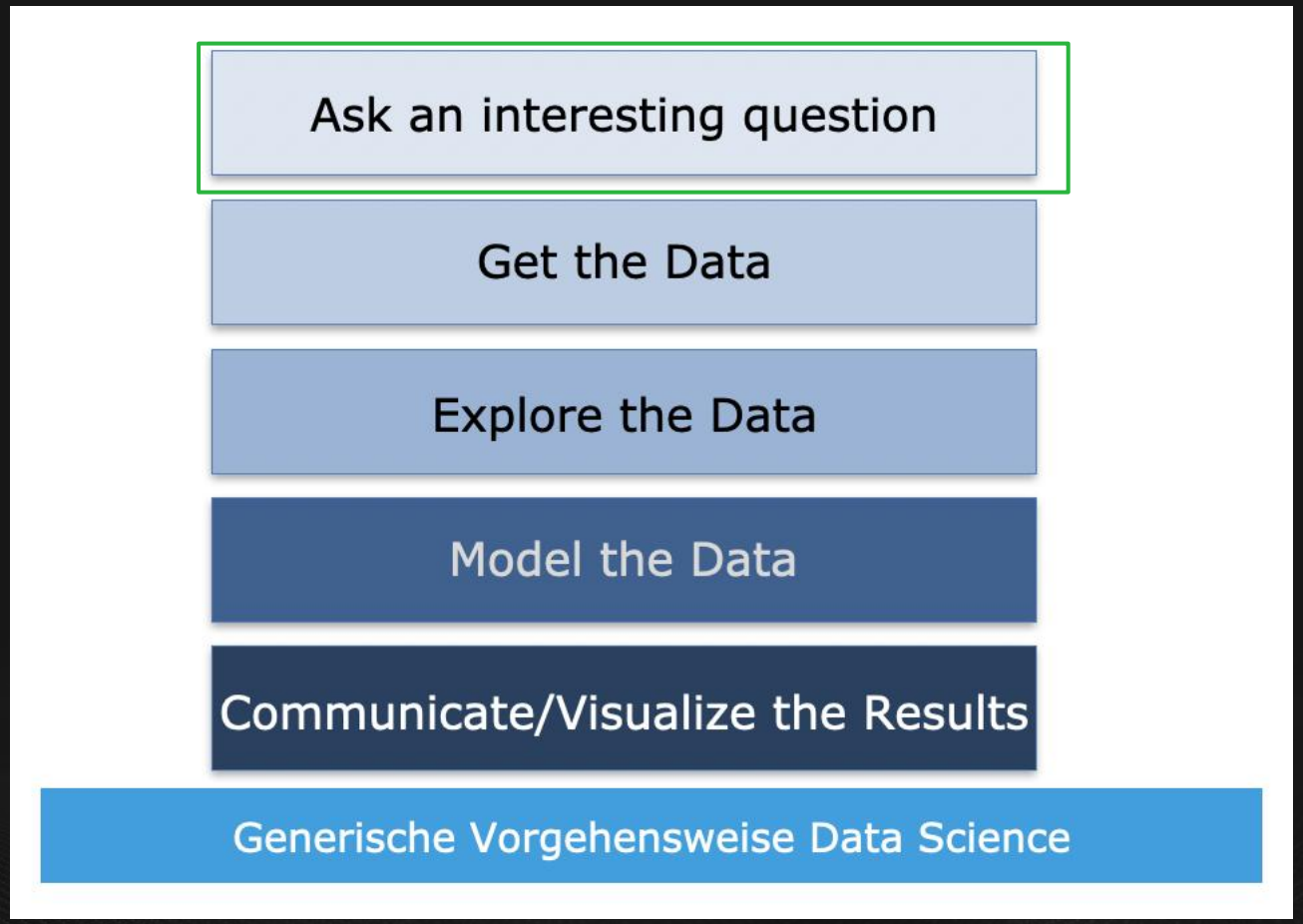
Vorgehen Data Science Use Case

Übersicht

- Datensatz: Auszug der Top 1000 Filme aus der Internet Movie Database.
- einfacher Datensatz: in der Realität haben Sie meist viel mehr Daten und Features und vor allem MEHR Data- und Feature Engineering Aufgaben zu lösen, um gute Ergebnisse zu erhalten.
- ACHTUNG: Sie sehen auf den folgenden Folien die einzelnen Befehle.
  In CoLab können Sie per GenAI diese Anweisungen auch automatisiert generieren!

Vorgehen Data Science Use Case

Vorgehen Data Science Use Case



Was sind die für den Geschäftszweck wichtigsten Informationen, die Sie aus dem Datensatz lernen können?

Quelle: Protopapas, Rader, Tanner, CS109 Data Science, 2020, Link, abgerufen am 15.09.2025

Vorgehen Data Science Use Case

Vorgehen Data Science Use Case

Für unseren Datensatz bieten sich beispielsweise folgende Fragen an:

- Was sind die am besten bewerteten Filme?
- Aus welchem Filmgenre kommen die meisten Filme?
- Gibt es einen Zusammenhang zwischen bestimmten Genres und Ratings?
- Was sind die besten Thriller?

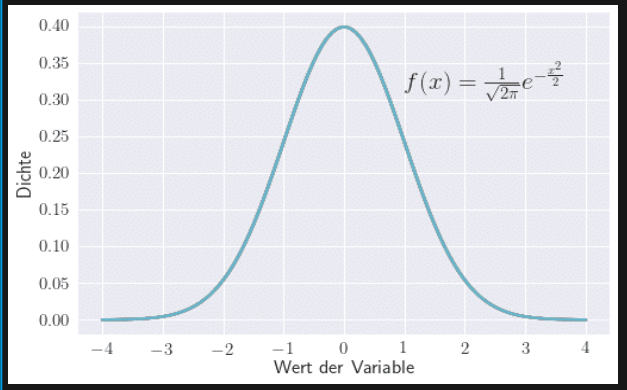
Vorgehen Data Science Use Case

Übersicht Statistik

- Deskriptive Statistik: Daten durch Graphiken oder Tabellen visuell beschreiben.
- Explorative Statistik: Zusammenhänge/ Muster zwischen Daten finden und bewerten, Entdecken von Hypothesen

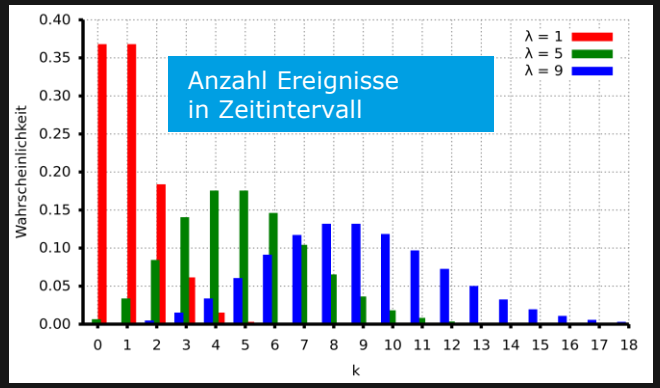
Vorgehen Data Science Use Case

Übersicht wichtiger statistischer Verteilungen



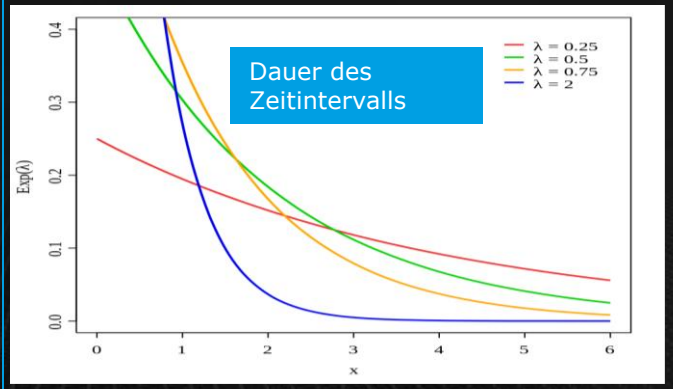
Normalverteilung:
Modellierung vieler natürlicher und statistischer Prozesse.

Beispiele:
- Größe Bevölkerung
- Prüfungsergebnisse
- Prozessqualität in einer Fabrik.



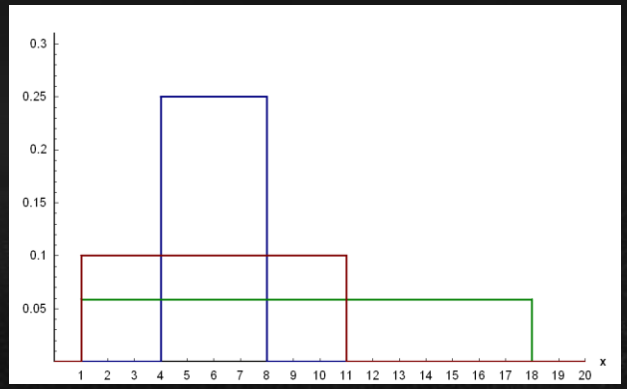
Poisson-Verteilung:
Modellierung Ereignisse, die bei konstanter mittlerer Rate unabhängig voneinander in einem festen Zeitintervall oder räumlichen Gebiet eintritt.

Beispiele:
- Hotline-Anrufe je Stunde
- Website-Ausfälle je Stunde



Expontentialverteilung:
Modellierung von Zeitintervallen.

Beispiele:
- Zeit bis Ausfall eines Geräts
- Wartezeit in Hotline



Gleichverteilung:
jeder Wert ist gleich wahrscheinlich (konstanter y-Wert).

Beispiele:
- Wurf einer idealen Münze oder Würfel

Deskriptive Statistik

Wichtige Parameter

Lageparameter:

- Mean: Mittelwert.
- Median: teilt Verteilung in 2 genau gleich große Hälften. Stabiler gegenüber Extremwerten als Mean.
- Modus: häufigster Wert der Verteilung.
- Min: kleinster Wert der Verteilung
- Max: größter Wert der Verteilung
- P-Quantil: Schwellenwert, der größer als p in % Elemente der Verteilung ist.

Streuungsparameter:

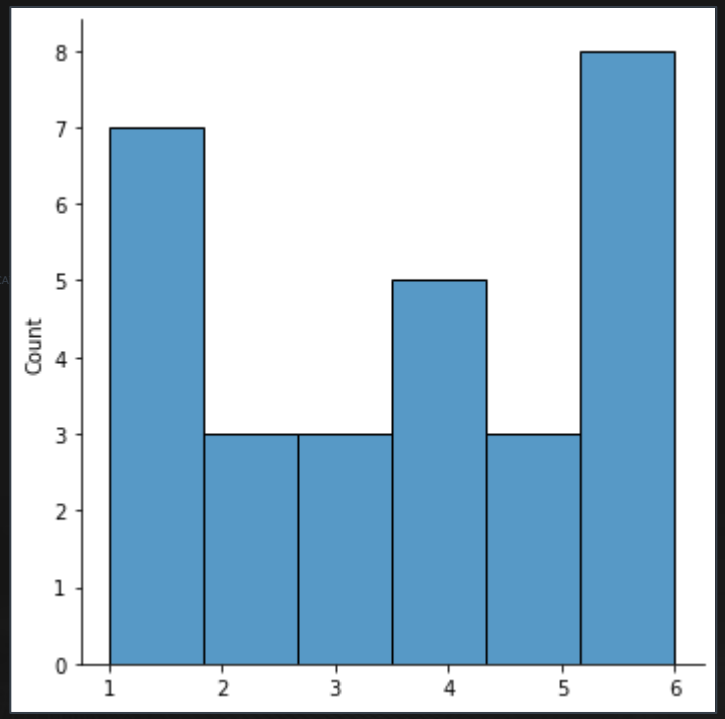
- Spannweite: Abstand Min und Max-Wert
- Varianz: (quadratische) Abweichung Werte vom Mittelwert. Basis für Standardabweichung.
- Standardabweichung: durchschnittliche Abweichung/Streuung Werte um Mittelwert.
- Schiefe: beschreibt Assymetrie Verteilung. Bei Rechtsschief sind häufiger Werte kleiner als Mittelwert, bei linksschief größer.
- Wölbung: Verteilungen mit geringer Wölbung streuen gleichmäßig; hohe Wölbung bedeutet extremere, seltenere Ergebnisse.

PARAMETER ERMÖGLICHEN EINE KOMPRIMIERTE ERFASSUNG EINER VERTEILUNG

Deskriptive Statistik

Detaillierung Lageparameter

Mean = Durchschnitt
Median = Wert, der Menge in genau 2 gleiche Hälften teilt

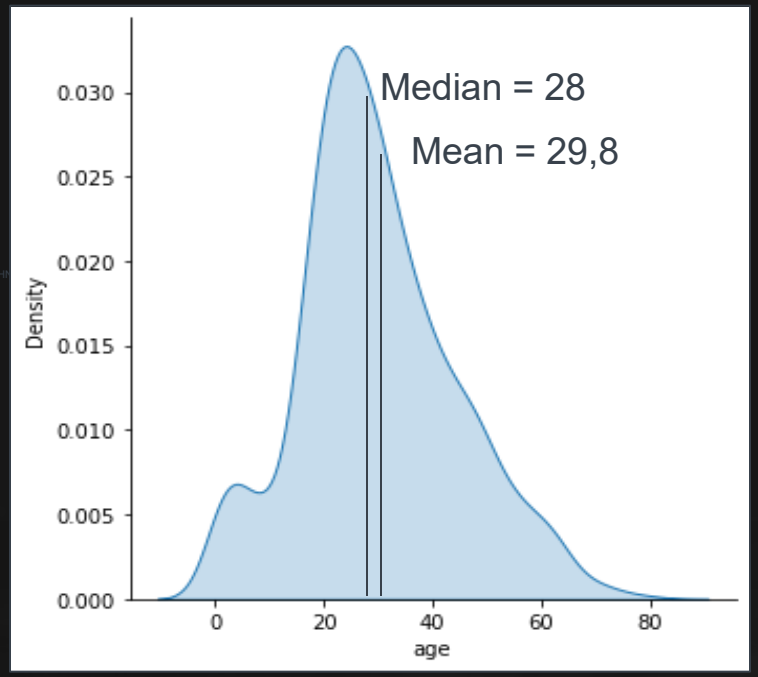


ERGEBNISSE WÜRFELN

Mean = 3.62
Median: 4
Modus: 6 ist häufigstes Ergebnis
Min: 1 ist niedrigster Ergebniswert
Max: 6 ist höchster Ergebniswert
P-Quantil:
25% = 2 (7 von 30 Ergebnissen kleiner als 2)
50% = Median
75% = 6 (21 von 30 Ergebnissen kleiner als 6)

Deskriptive Statistik

Detaillierung Streuungsparameter



Mean: 29.881138
Std: 14.413493
Min: 0.170000
25%: 21
50%: 28
75%: 39
Max: 80

ALTERSVERTEILUNG TITANIC-PASSAGIERE

Spannweite: 80 Jahre – 0,29 Jahre = 79,71 Jahre

Varianz: 207.55

Standardabweichung:14,41 -> weite Streuung Alter

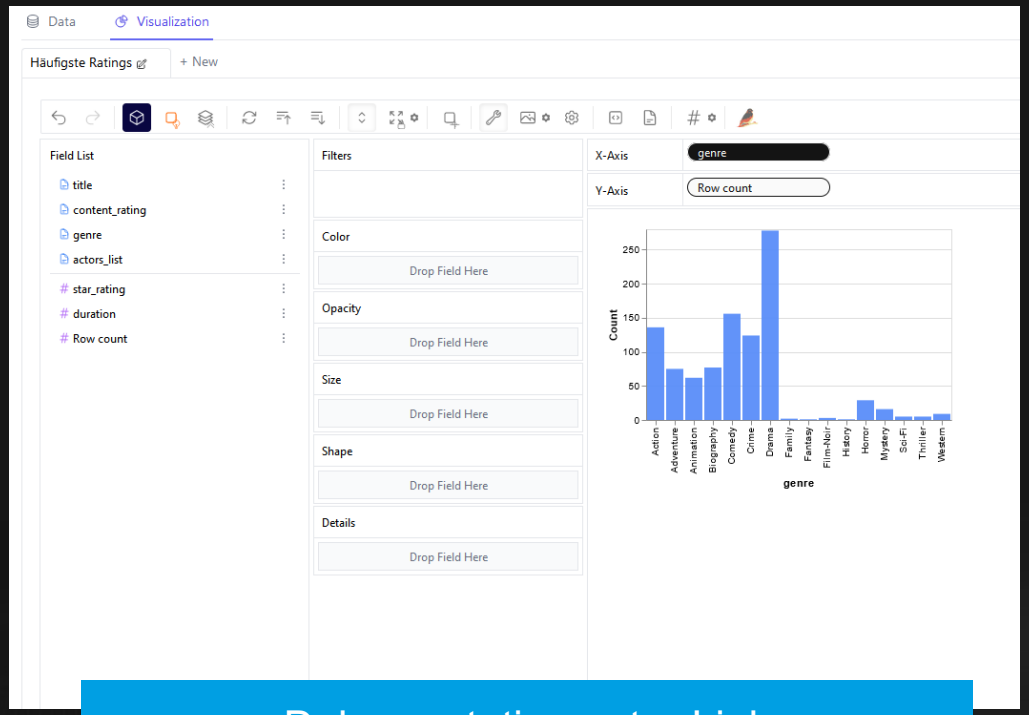
Schiefe: rechtsschief, da Median kleiner als Mean. Mehr als 50% der Passagiere jünger als Durchschnittsalter.

Wölbung: geringe Wölbung, gleichmäßige Streuung.

Vorgehen Data Science Use Case

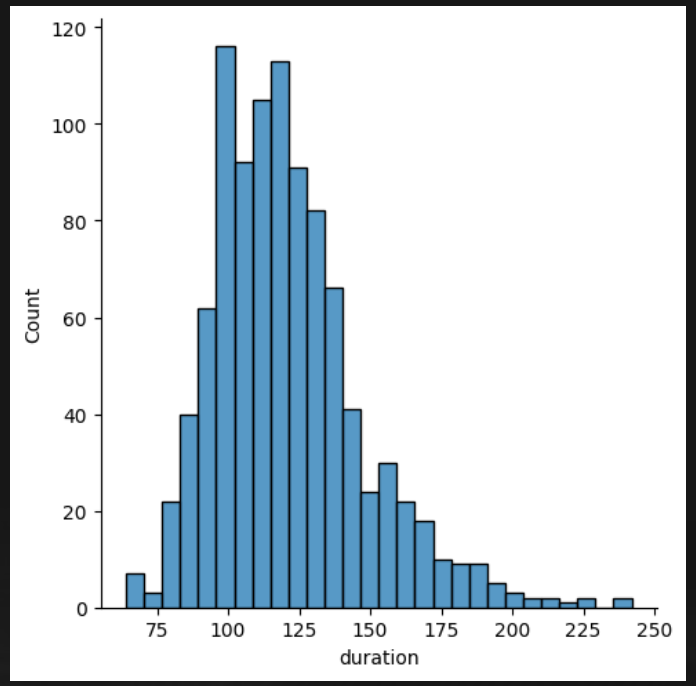
Explorative Statistik

PygWalker



Dokumentation unter Link

Matplotlib

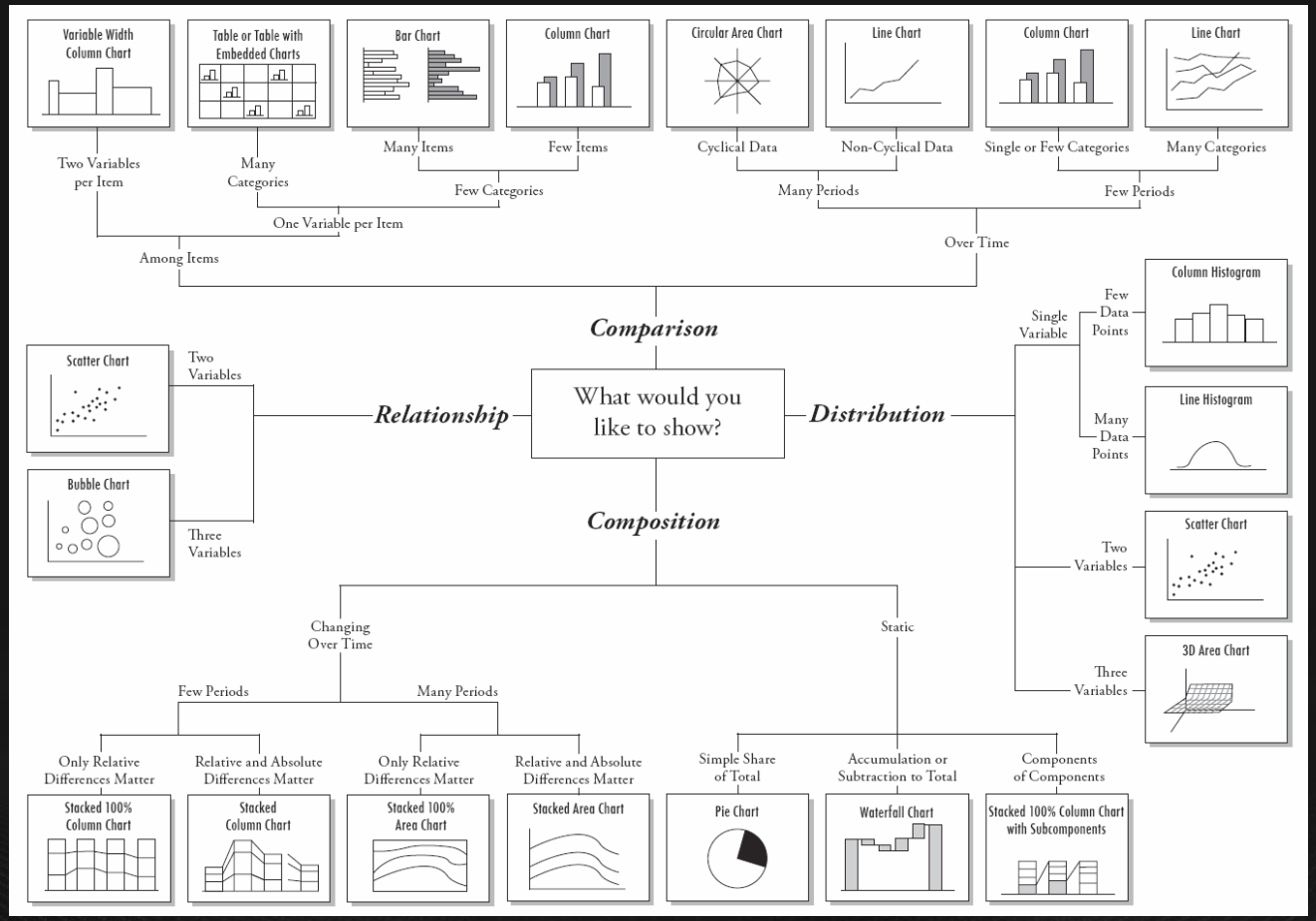


Dokumentation unter Link

- PyGWalker: neues Werkzeug und ermöglicht schnelle Auswertungen.
- MatPlotLib: mehr Möglichkeiten, erfordert aber ein wenig Einarbeitung.

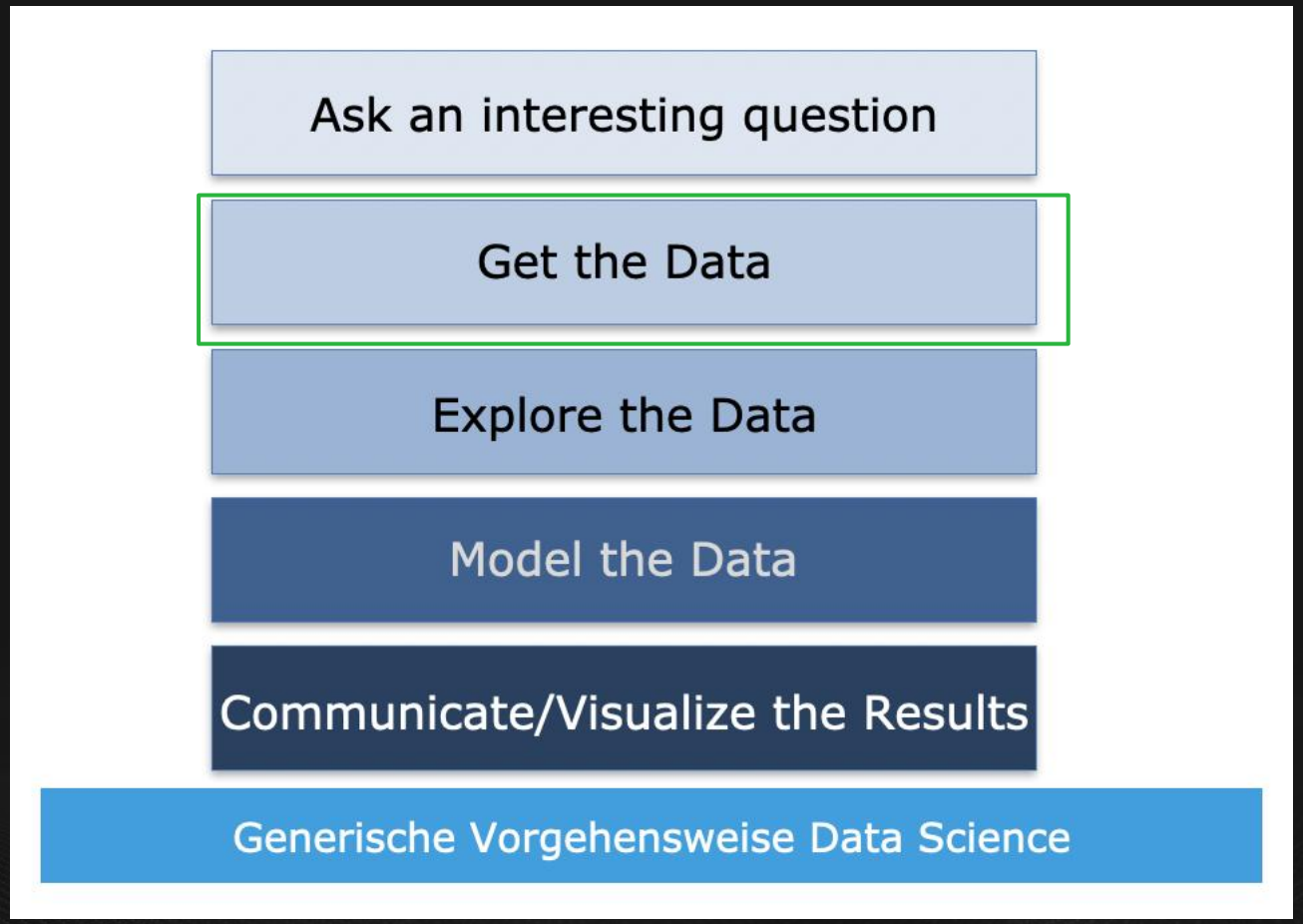
Vorgehen Data Science Use Case

Cheat sheet plots



Vorgehen Data Science Use Case

Übersicht



Wie wurden die Daten generiert? Sind alle Daten relevant? Datenschutz?

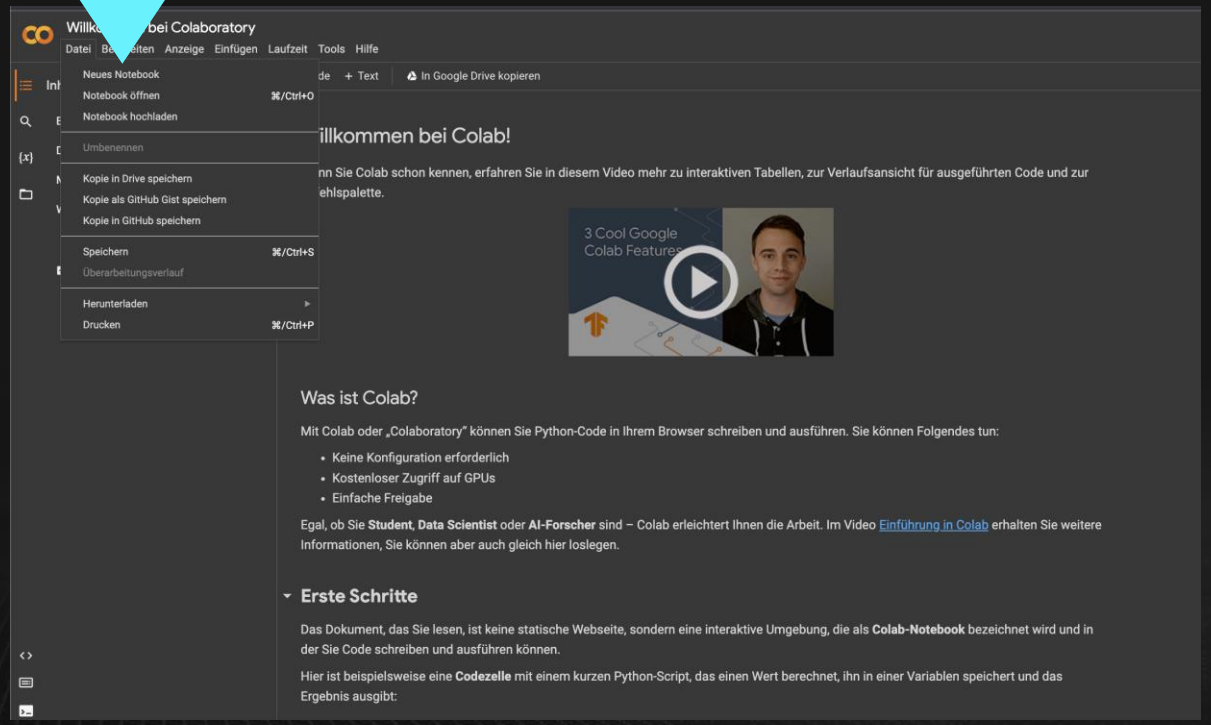
Quelle: Protopapas, Rader, Tanner, CS109 Data Science, 2020, Link, abgerufen am 15.09.2025

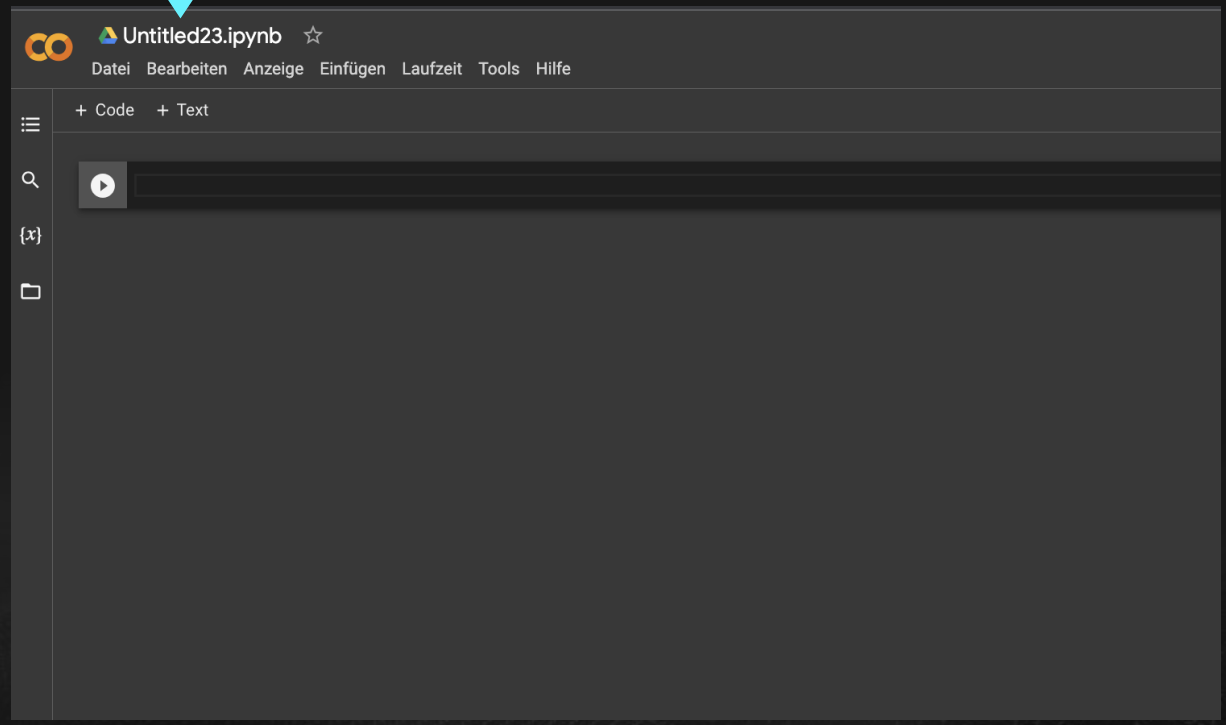
Vorgehen Data Science Use Case

Erstellen Notebook

Neues Notebook öffnen

Notebook benennen, z.B. IMDB





LINK: HTTPS://COLAB.RESEARCH.GOOGLE.COM/

Vorgehen Data Science Use Case

Laden der wichtigsten Bibliotheken

```
[ ]  import pandas as pd   # Importieren Standard-Library für das Bearbeiten und Laden von Daten ("Data Engineering").
     import matplotlib.pyplot as plt # Standard-Library für das Plotten von Graphen.
     import seaborn as sns   # verschönert Matplotlib-Graphiken
     import numpy as np # Standard-Library für Rechnen
```

```
[ ]  # mit diesem Befehl installieren Sie pygwalker in dem Notebook.
     # der parameter -q sorgt dafür, daß keine weiteren Bildschirmausgaben erfolgen
     !pip install pygwalker -q
     import pygwalker as pyg
```

Parameter -q sorgt dafür, daß keine weiteren Bildschirmausgaben erfolgen

Befehle: import as → Laden einer Library unter einem für verständlichen Namen
install → Installieren einer Library (hier: pygwalker)

Vorgehen Data Science Use Case

Laden des Datensatzs (per Website)

```
# wir speichern den Pfad für die Datei in der Variable url_webpage
url_webpage = 'https://raw.githubusercontent.com/thechaudharysab/imdb-data-pandas-visualization/master/data/imdb_1000.csv'
IMDB_df = pd.read_csv(url_webpage, sep=',') # einzelnen Einträge in CSV sind durch , getrennt. Oft werden diese aber auch per ; getrennt
```

Befehle in Pandas werden mit () angezeigt, in der die weiteren/keine weiteren Spezifikationen definiert werden.

Parameter sep gibt an, wie im Datensatz die einzelnen Einträge getrennt sind. Meist per , oder ;

Vorgehen:

1. Variable mit dem Pfad der Website erstellen
2. Datei in einen (von uns vergebenen) dataframe einlesen

Befehle: read() → Einlesen einer Datei → pd. (in Pandas), _csv (csv-Datei)

Vorgehen Data Science Use Case

Laden des Datensatzs (per Festplatte)

```
# Hochladen Datensatz über eine Eingabemaske von lokalem Rechner etc.
from google.colab import files
files.upload()
IMDB_df = pd.read_csv('imdb_1000.csv', sep=',')
```

Dateien auswählen | Keine ausgewählt | Cancel upload

Parameter sep gibt an, wie im Datensatz die einzelnen Einträge getrennt sind. Meist per , oder ;

von Festplatte (Schreibtisch bei Mac)

Vorgehen:

1. Datei von lokalem Speicher hochladen
2. Datei in einen (von uns vergebenen) Dataframe einlesen; Dataframe ist eine Art sehr große Tabelle

Befehle: from import → Konfigurieren der Bibliothek für das Dateiladen
files.upload() → Hochladen der konkreten Datei
read() -> einlesen einer datei → pd. (in pandas), _csv (csv-datei)

Vorgehen Data Science Use Case

Laden des Datensatzes

```
[ ]  # Geben Sie den Namen des Dataframes an und führen Sie die Zelle aus. Damit sehen Sie, ob das Einlesen funktioniert hat
     IMDB_df
```

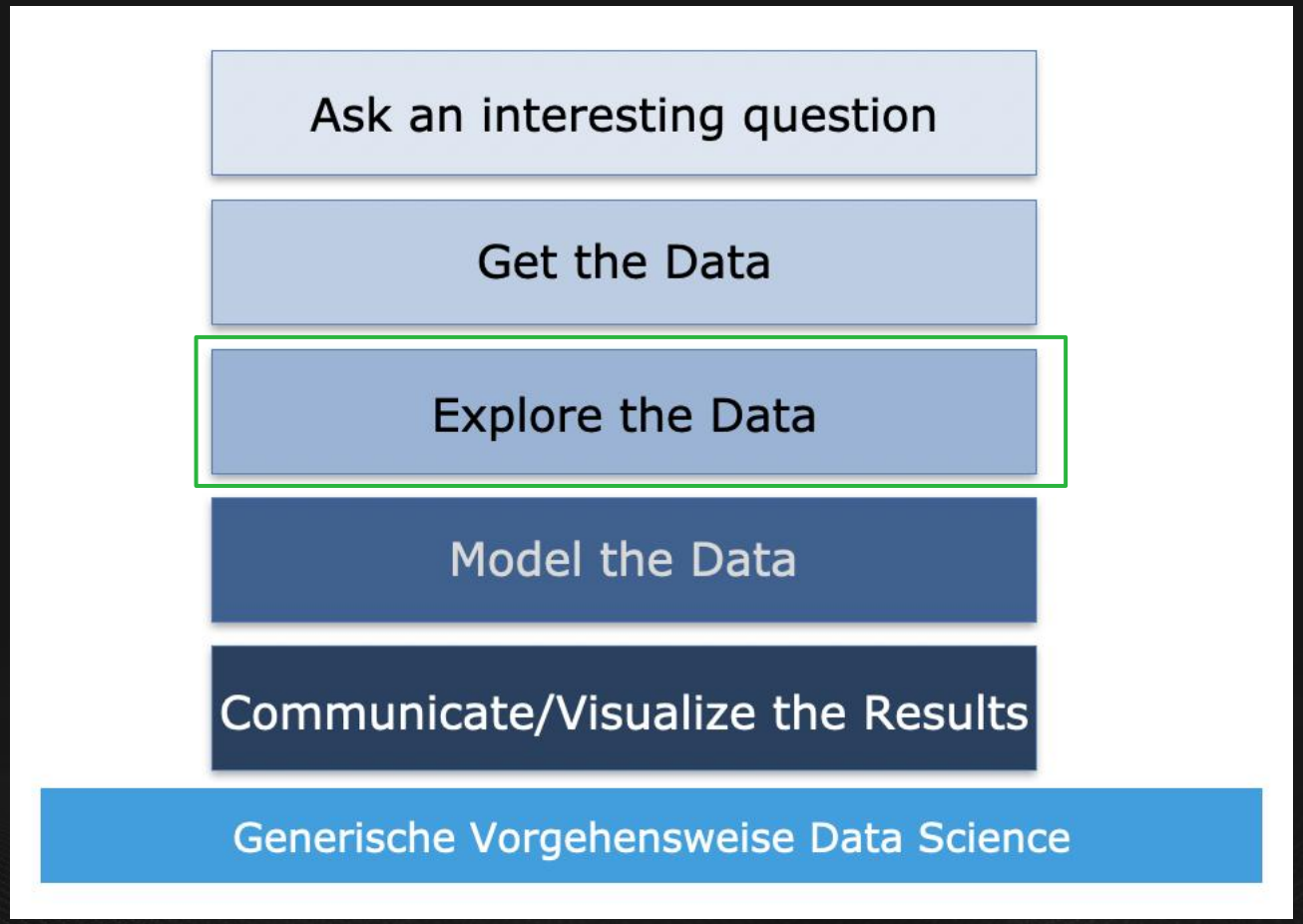
	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	9.0	The Dark Knight	PG-13	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	8.9	Pulp Fiction	R	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...	...	...	...	...
974	7.4	Tootsie	PG	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	7.4	Back to the Future Part III	PG	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	7.4	Master and Commander: The Far Side of the World	PG-13	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	7.4	Poltergeist	PG	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	7.4	Wall Street	R	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 6 columns

Befehle: IMDB_df → zeigt Datensatz mit seinen Features an. So sehen Sie, ob die Datei richtig eingelesen wurde.

Vorgehen Data Science Use Case

Vorgehen Data Science Use Case



Erstes visuelles Verständnis? Gibt es Anomalien? Unplausible Werte? Sieht man erste Muster?

Quelle: Protopapas, Rader, Tanner, CS109 Data Science, 2020, Link, abgerufen am 15.09.2025

Vorgehen Data Science Use Case

Deskriptive Statistik

```
# Verwenden Sie die Methode describe mit Parameter include='all auf Ihr Dataframe
IMDB_df.describe(include='all')
```

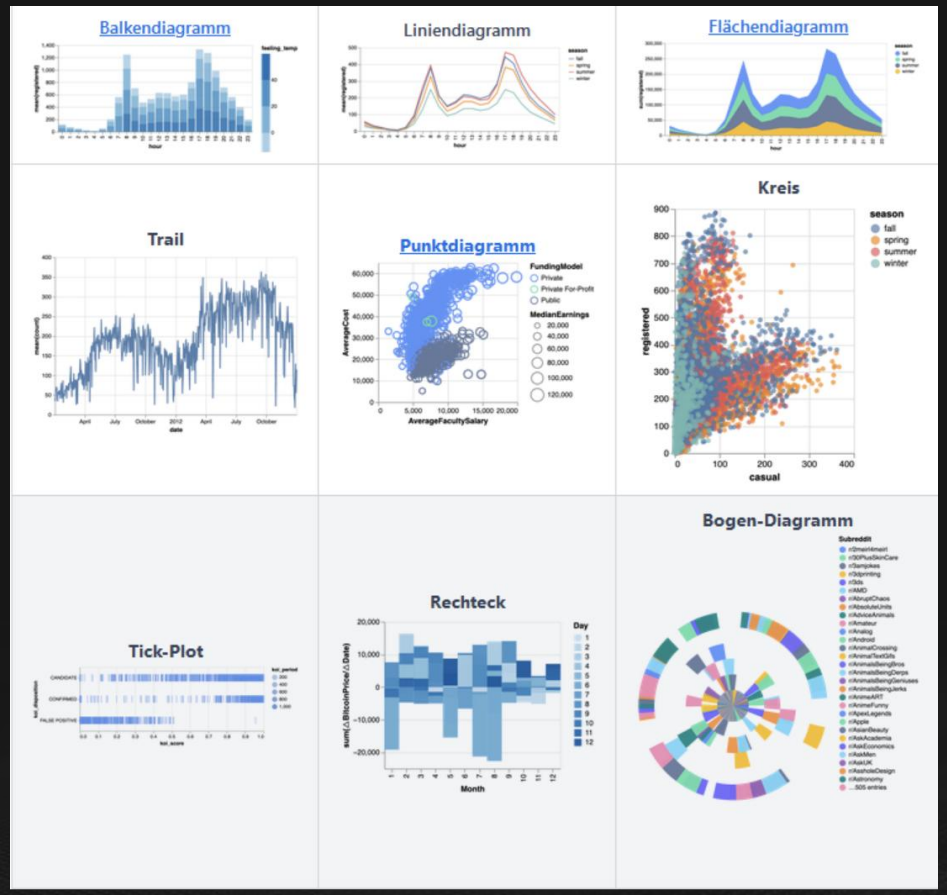
	Poster_Link	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating
count	1000	1000	1000	899	1000	1000	1000.000000
unique	1000	999	100	16	140	202	NaN
top	https://m.media-amazon.com/images/M/MV5BMDFkYT...	Drishyam	2014	U	100 min	Drama	NaN
freq	1	2	32	234	23	85	NaN
mean	NaN	NaN	NaN	NaN	NaN	NaN	7.949300
std	NaN	NaN	NaN	NaN	NaN	NaN	0.275491
min	NaN	NaN	NaN	NaN	NaN	NaN	7.600000
25%	NaN	NaN	NaN	NaN	NaN	NaN	7.700000
50%	NaN	NaN	NaN	NaN	NaN	NaN	7.900000
75%	NaN	NaN	NaN	NaN	NaN	NaN	8.100000
max	NaN	NaN	NaN	NaN	NaN	NaN	9.300000

- Count: Anzahl Werte je Spalte, z.B. fehlen bei Certificate 101 Werte
- Unique: Anzahl eindeutige Werte Spalte, z.B. 140 verschiedene Runtimes.
- Top: häufigsten Werte in Spalte, z.B. 2014 bei Release.
- Frequency: wie häufig top Eintrag vorkommt, hier 32 mal bei Release
- Mean: Mittelwert, z.B. 7,9 beim Rating
- Std: Standardabweichung, d.h. durchschnittliche Breite der Streuung einzelner Werte um den Mittelwert herum.
- Min/Max: geringster/ höchster Wert der Spalte
- 50%: Wert, der genau in der Mitte der Daten liegt, d.h. er teilt die Daten der Spalte in 2 genau gleich große Untermengen (Median)
- 25%: Wert, der größer ist als 25% der Daten.
- 75%: Wert, der größer ist als 75% der Daten.

befehle: .describe() → statistischer Überblick über Dataframe (Zeilen und Spalten)
→ include="all" (alles zeigen)

Vorgehen Data Science Use Case

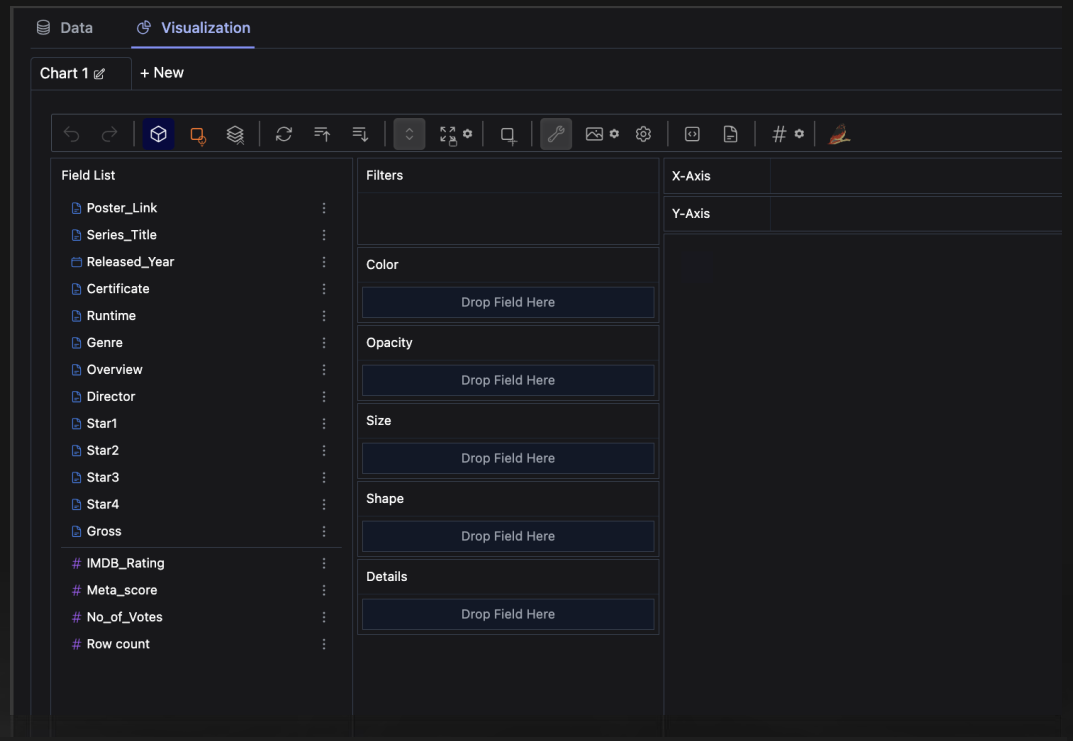
Datenexploration mit pygwalker



Vorgehen Data Science Use Case

Datenexploration mit pygwalker

```
[ ]  walker = pyg.walk(
         IMDB_df,
         spec="./chart_meta_1.json",
         use_kernel_calc=True,
     )
```

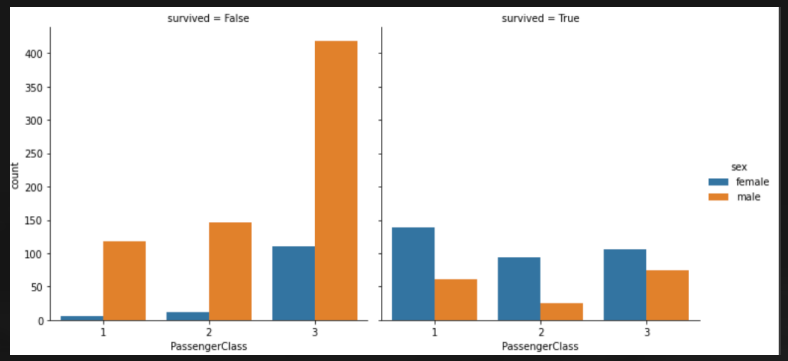



```
# this json file will save your chart state, you need to click save button in ui mannual when you finish a chart, 'autosave' will be supported in the future.
# set `use_kernel_calc=True`, pygwalker will use duckdb as computing engine, it support you explore bigger dataset(<=100GB).
```

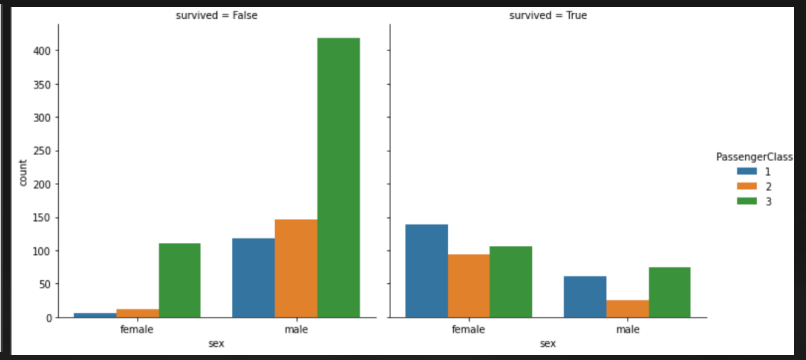
Vorgehen Data Science Use Case

Datenexploration mit Matplotlib

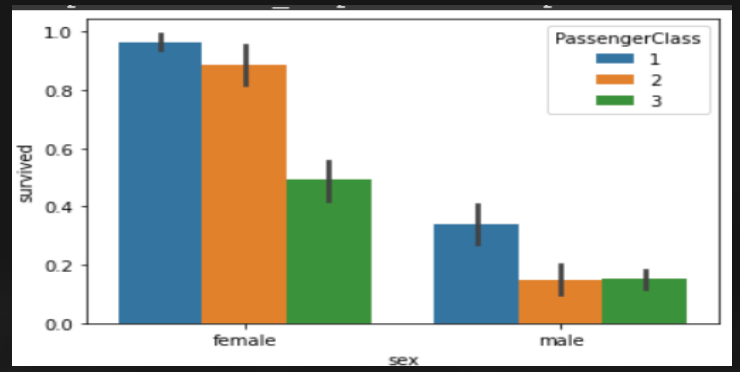
```
g = sns.catplot(x = "PassengerClass",
                hue = "sex",
                col = "survived",
                data = titanic_df,
                kind = "count")
```




```
g = sns.catplot(x = "sex",
                hue = "PassengerClass",
                col = "survived",
                data = titantic_plot_df,
                kind = "count")
```




```
sns.barplot(x = "sex",
            hue = 'PassengerClass',
            y = "survived",
            data = titantic_plot_df
            )
```



BEFEHLE: sns.catplot bzw. sns.barplot (AUS SEABORN):
Anzeigen der Jeweiligen Plots mit entsprechenden Einstellungen
(z.B. x = „Was steht auf X-Achse?“, hue = „Was sollen Balken anzeigen?“, col = „Welche Werte sollen verglichen werden?“)

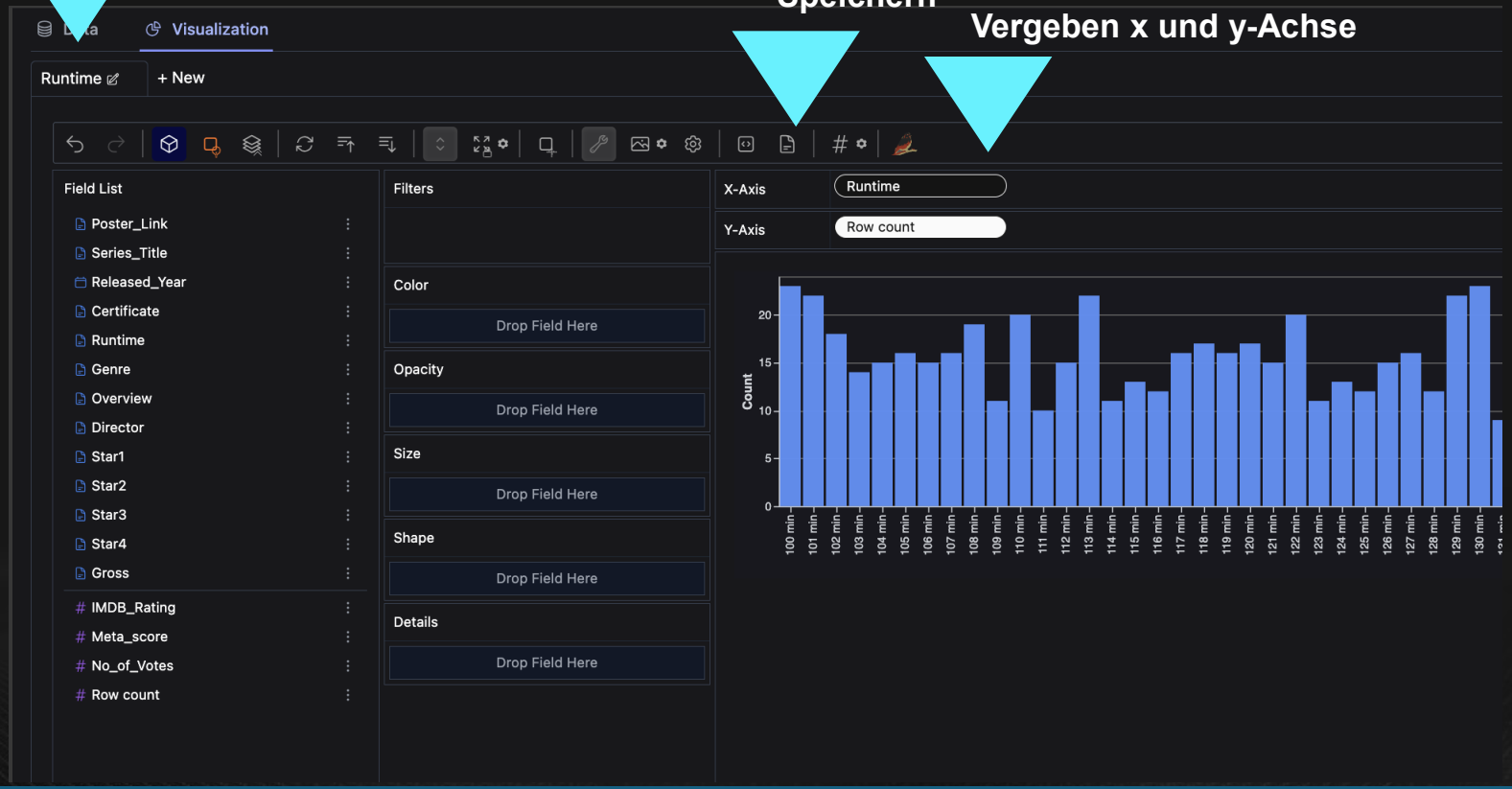
Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Pygwalker

Definition Name der Auswertung

Speichern

Vergeben x und y-Achse

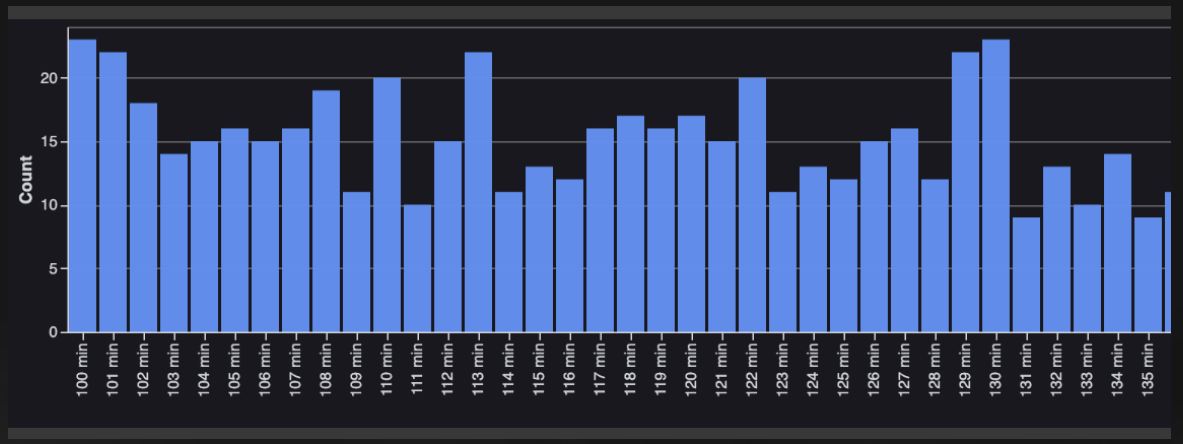


Fragestellung: was ist die häufigste Länge eines Films?

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Pygwalker

```
[24] # per PyGWalker
     walker.display_chart("Runtime")
```

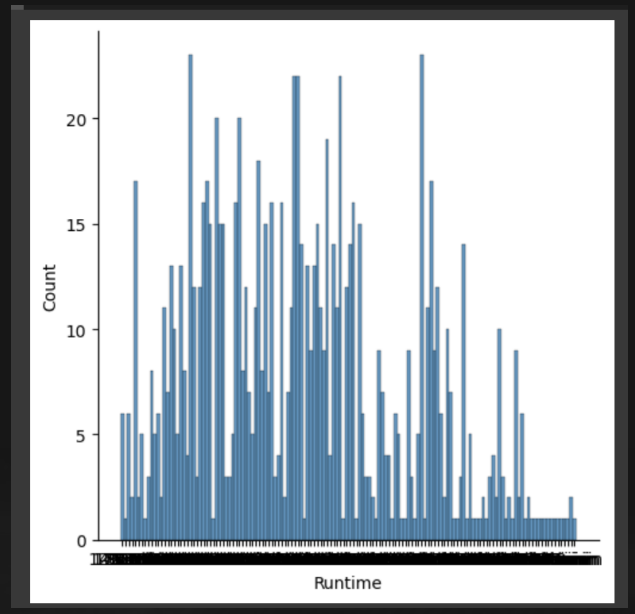


Befehle: walker.display_chart() → Chart anzeigen

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Matplotlib

```
ax = sns.displot(data=IMDB_df, x="Runtime")
```



Befehle: sns.displot (aus seaborn):
Anzeigen der jeweiligen Plots mit den entsprechenden Einstellungen
(z.b. x = „was steht auf der x-achse)

Fragestellung: was ist die häufigste Länge eines Films?

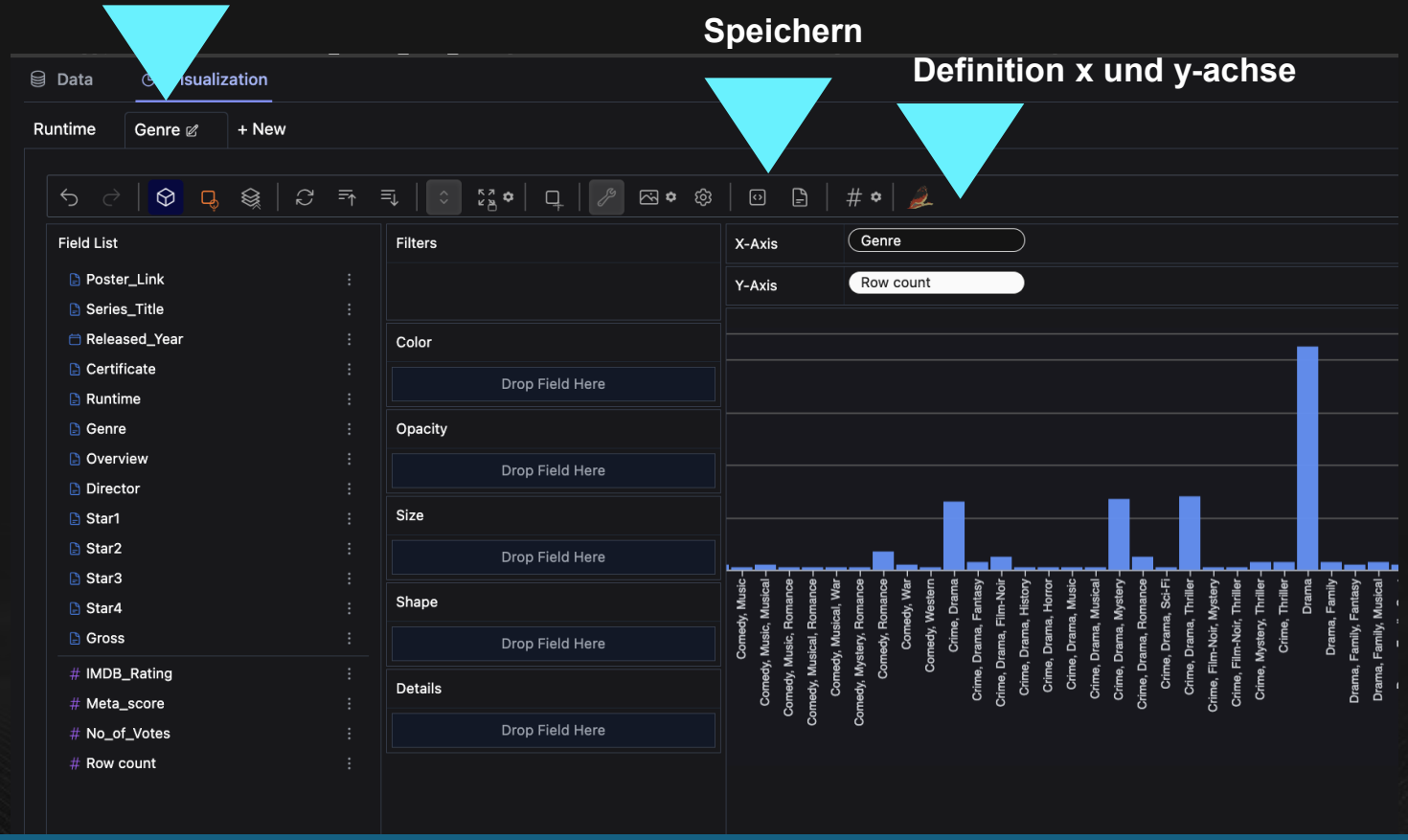
Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Pygwalker

Definition Name der Auswertung

Speichern

Definition x und y-achse

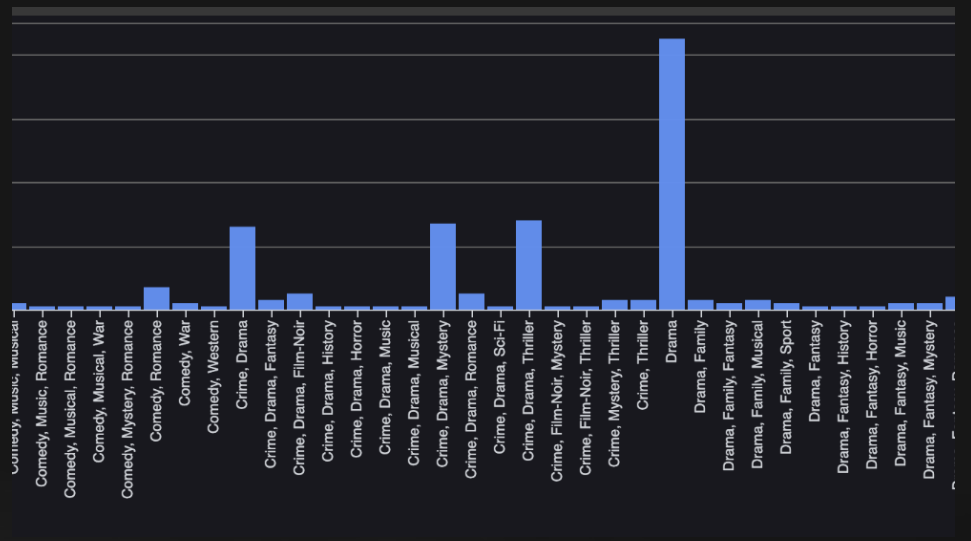


Fragestellung: Aus welchem Genre kommen die meisten Filme (PYGWALKER)?

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Pygwalker

```
walker.display_chart("Genre")
```

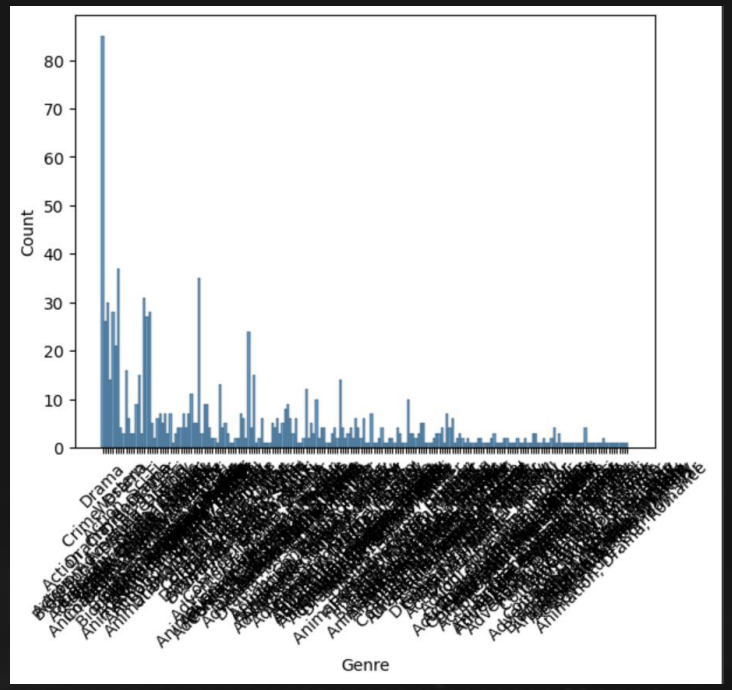


Fragestellung: Aus welchem Genre kommen die meisten Filme (PYGWALKER)?

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Matplotlib

```
ax = sns.histplot(x="Genre", data= IMDB_df);
plt.xticks(rotation=45);
```



Befehle: sns.Histplot (aus Seaborn):
Anzeigen der jeweiligen Plots mit Einstellungen
plt.Xticks(rotation=45): Dreht X-Achse um 45° Grad

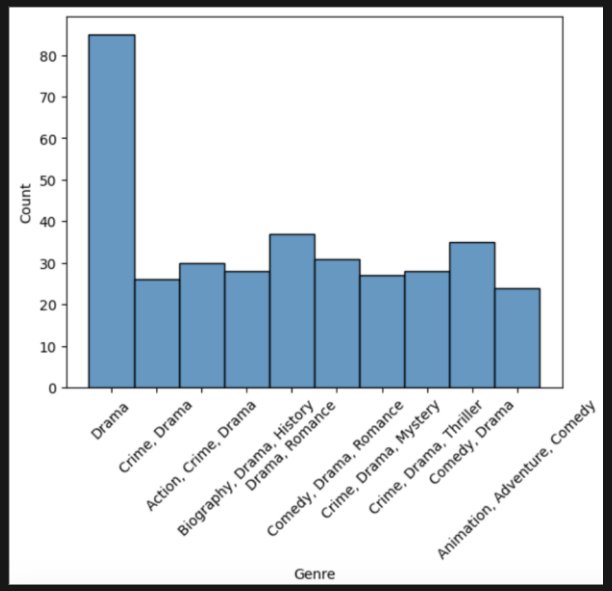
Fragestellung: Aus welchem Genre kommen die meisten Filme (Matplotlib)?

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Matplotlib

```
# Determine the top 10 genres by count
top_10_genres = IMDB_df['Genre'].value_counts().nlargest(10).index

# Filter the dataframe to only include movies from these genres
filtered_df = IMDB_df[IMDB_df['Genre'].isin(top_10_genres)]
```

```
ax = sns.histplot(x="Genre", data= filtered_df);
plt.xticks(rotation=45);
```



Vorgehen Filterung:
- Filteranweisung erstellen
- Neuen (gefilterten) Dataframe erstellen

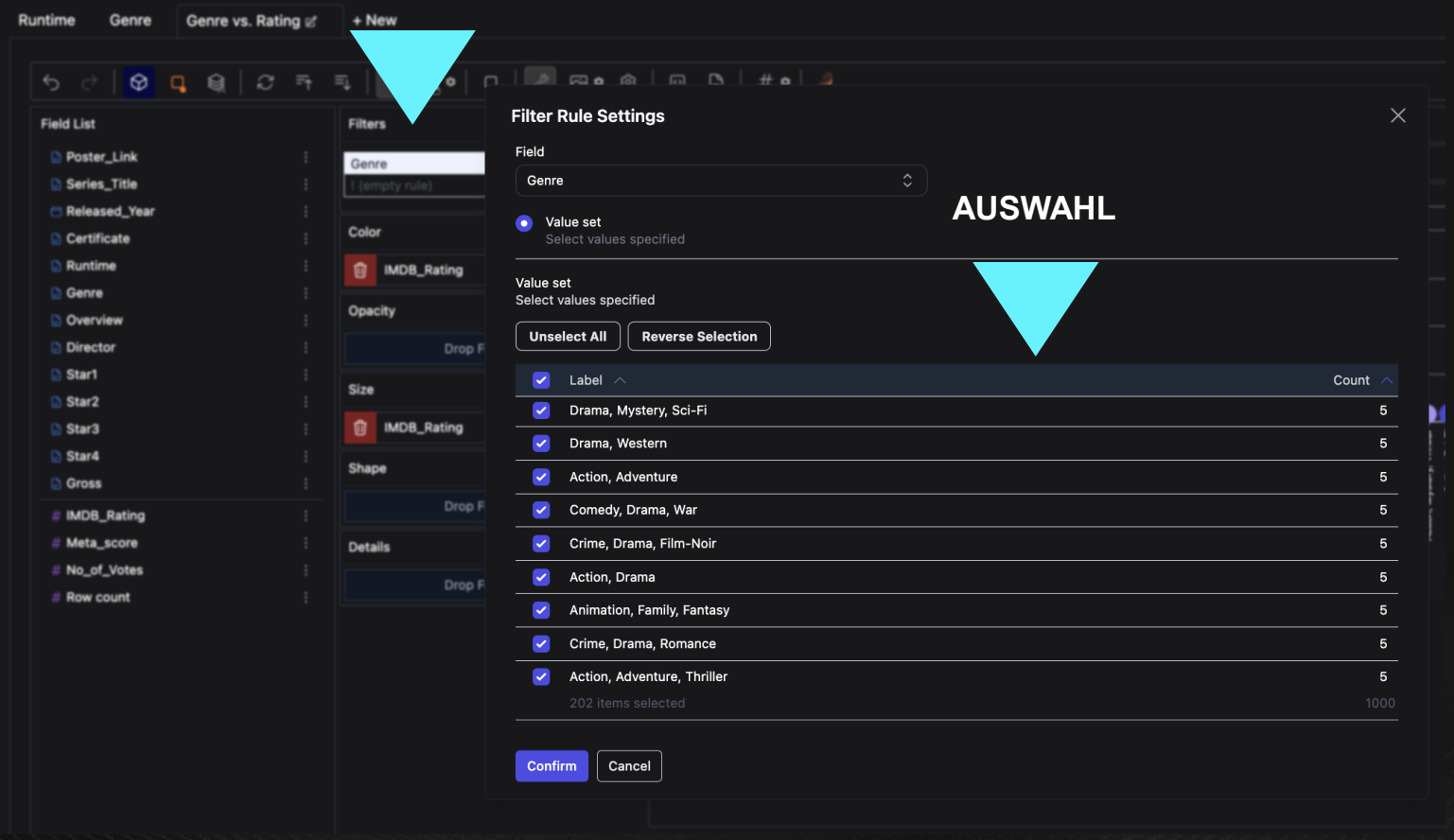
Befehle: value_counts() → Werte zählen
.largest(10) → die 10 grössten Werte
.isin() → Hier können Anweisungen für die Filterung gegeben werden

Fragestellung: Aus welchem Genre kommen die meisten Filme (Matplotlib – andere Darstellung)?

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Pygwalker

SETZEN FILTER

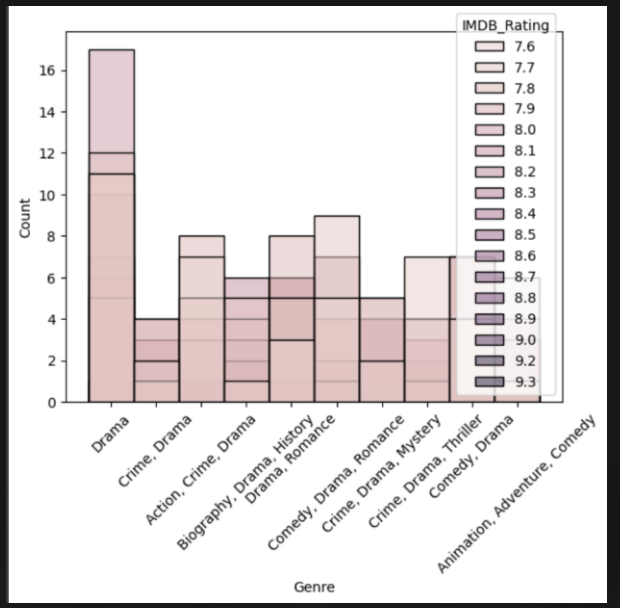


Hypothese: es gibt einen Zusammenhang zwischen Genre und Rating (pygwalker)

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Pygwalker

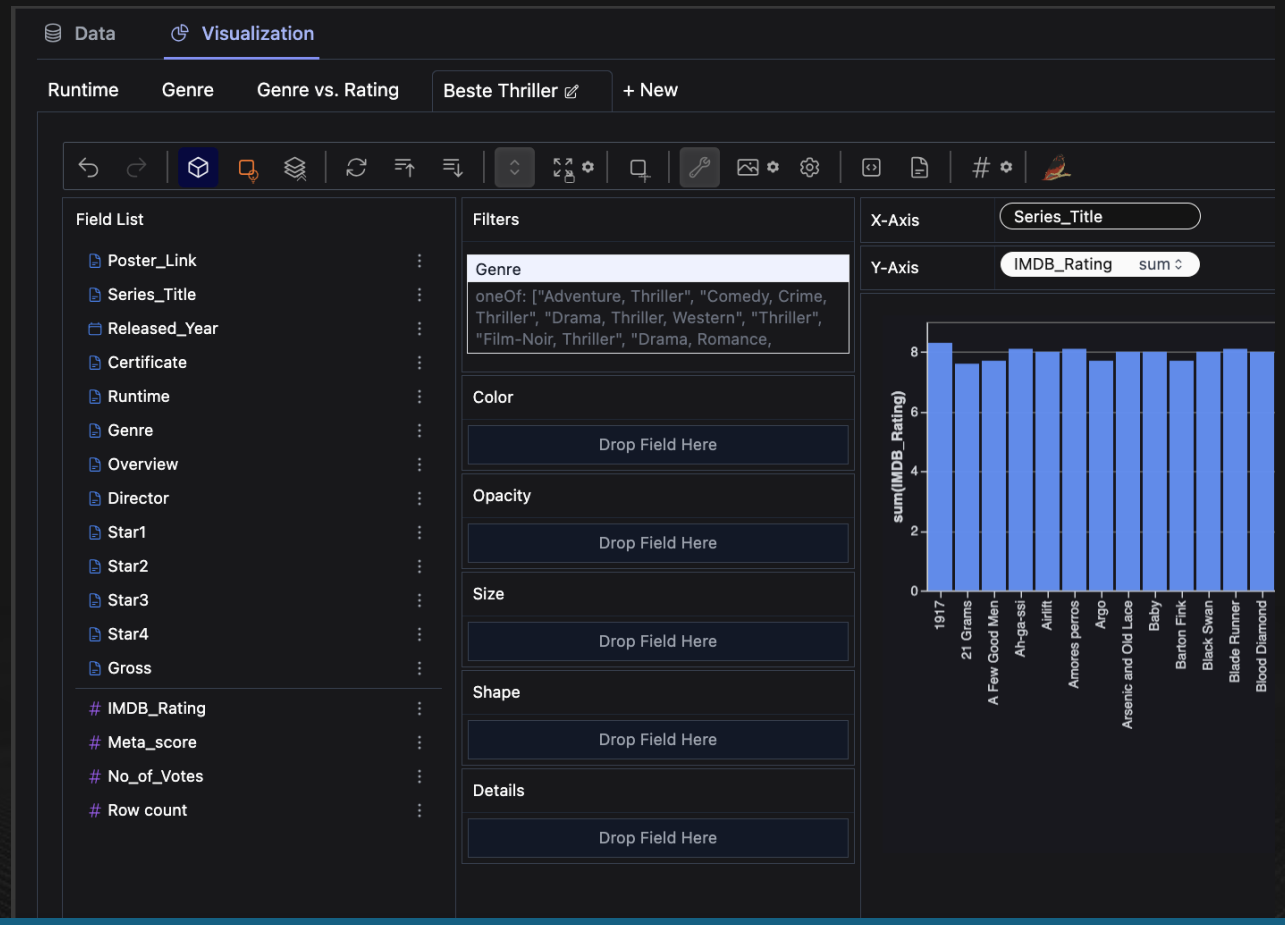
```
# Determine the top 10 genres by count
top_10_genres = IMDB_df['Genre'].value_counts().nlargest(10).index

# Filter the dataframe to only include movies from these genres
filtered_df = IMDB_df[IMDB_df['Genre'].isin(top_10_genres)]

# Plot the histogram
ax = sns.histplot(x="Genre", hue="IMDB_Rating", stat="count", data=filtered_df)
plt.xticks(rotation=45)
plt.show()
```



Hypothese: es gibt einen Zusammenhang zwischen Genre und Rating (pygwalker)

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Pygwalker

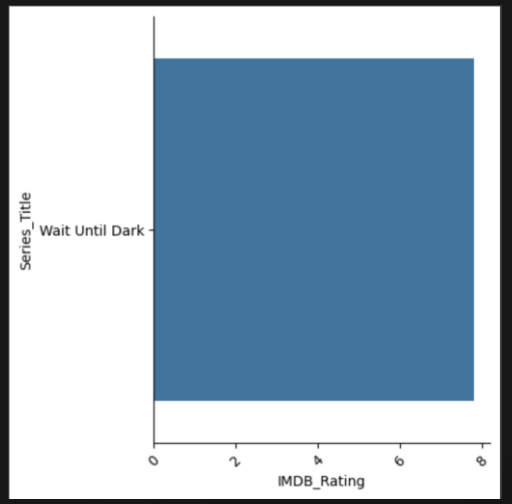


Fragestellung: was sind die besten Thriller (PYGWALKER)?

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Matplotlib

```
ax = sns.catplot(y="Series_Title", x="IMDB_Rating", data=IMDB_df[IMDB_df['Genre']=='Thriller'], kind="bar");
plt.xticks(rotation=45);
```



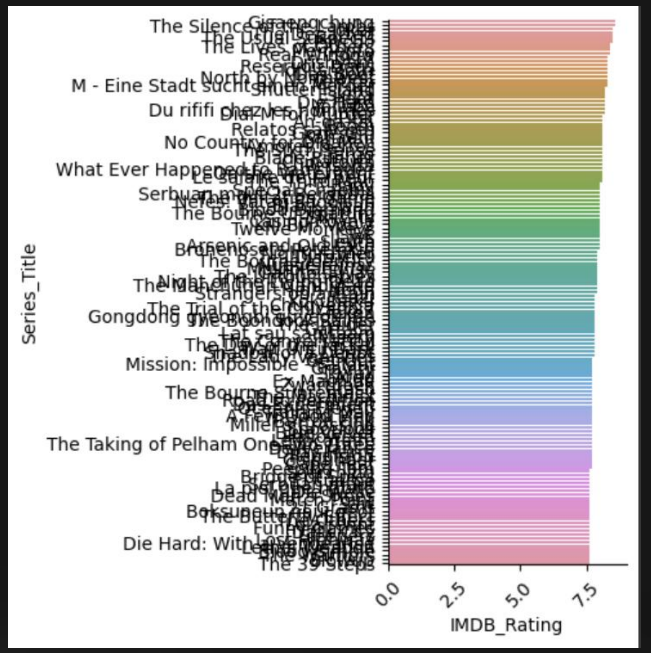
Befehle: sns.catplot (x und y für Bestimmung der x- und y-Achse),
data=IMDB_df[IMDB_df['genre']=='Thriller']: filtert alle Zeilen aus Dataset mit Genre = Thriller

Fragestellung: was sind die besten Thriller (Matplotlib)?

Vorgehen Data Science Use Case

Beispielhafte Auswertung mit Matplotlib

```python
# Filter the dataframe to include movies with genres containing the word "Thriller"
filtered_df = IMDB_df[IMDB_df['Genre'].str.contains('Thriller', case=False)]

# Plot the catplot
ax = sns.catplot(y="Series_Title", x="IMDB_Rating", data=filtered_df, kind="bar")
plt.xticks(rotation=45)
plt.show()
```



Vorgehen:
- Filteranweisung erstellen
- Neuen (gefilterter) Dataframe plotten

Befehle: .str.contains() → Filter auswählen
case=true → case_sensitiv, d.h. Filter funktioniert nur auf „Thriller“ nicht „thriller

Fragestellung: was sind die besten Thriller (Matplotlib)?

Vorgehen Data Science Use Case

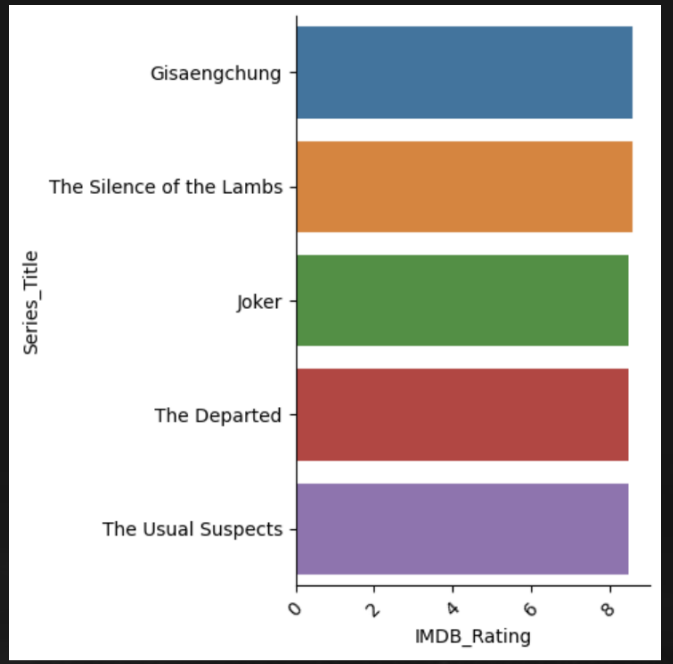
Übung

Zeigen Sie nur die 5 Filme mit dem besten Rating

Vorgehen Data Science Use Case

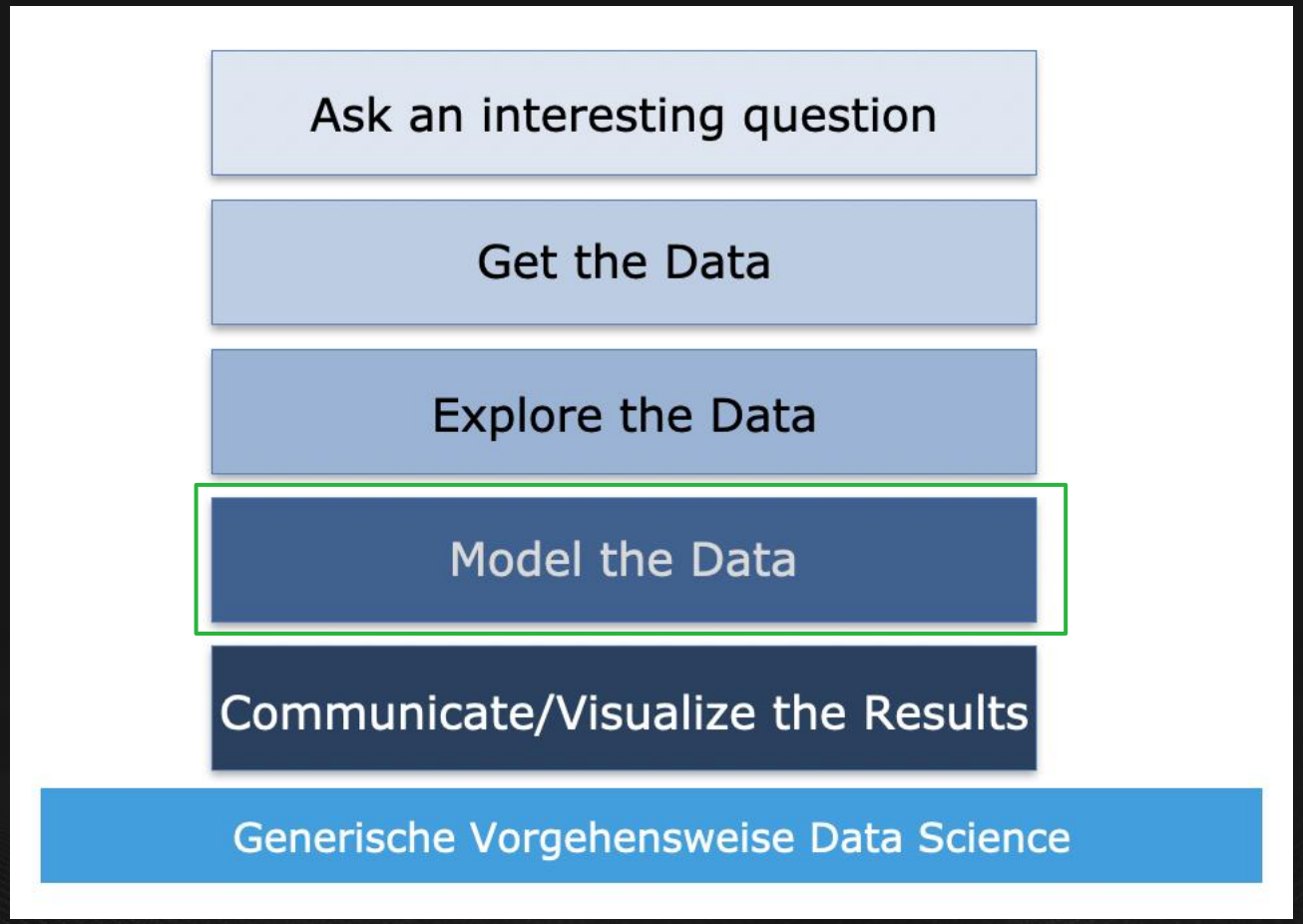
Übung

```python
# Filter and sort the dataframe
filtered_df = IMDB_df[IMDB_df['Genre'].str.contains('Thriller', case=False)]
top_5_movies = filtered_df.nlargest(5, 'IMDB_Rating')

# Plot the catplot for top 5 movies by IMDB Rating
ax = sns.catplot(y="Series_Title", x="IMDB_Rating", data=top_5_movies, kind="bar")
plt.xticks(rotation=45)
plt.show()
```



Vorgehen Data Science Use Case

Übersicht

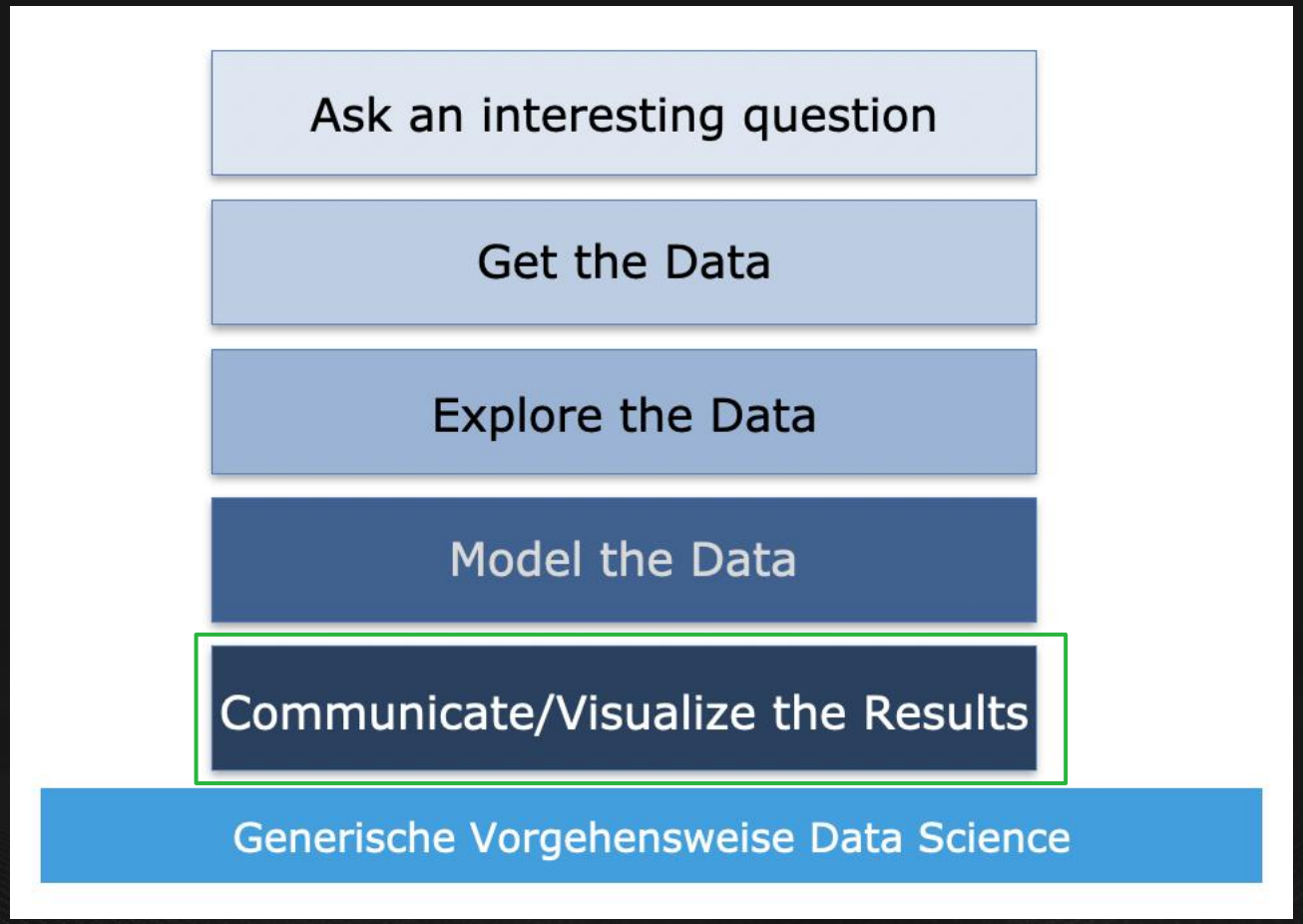


Modellerstellung, Modelltraining, Modellvalidierung

Quelle: Protopapas, Rader, Tanner, CS109 Data Science, 2020, Link, abgerufen am 15.09.2025

Vorgehen Data Science Use Case

Übersicht



Was lernen wir aus den Daten? Ergebnisse sinnvoll? Wie lassen sich die Ergebnisse kommuzieren?

Quelle: Protopapas, Rader, Tanner, CS109 Data Science, 2020, Link, abgerufen am 15.09.2025

Literatur

Statistik

- James, Witten, Hastie and Tibshirani: An Introduction to Statistical Learning (freies Ebuch unter: Link)
- Spiegelhalter: The Art of Statistics: Learning from data
- Witte: Statistics (10th Edition)
- Silver: The Signal and the noise
- Taleb: Black Swan
- Huff: How to Lie with Stastistics
- Wheelan: Naked statistics