Handout (pandas & plotly)

Zeitreihenanalyse der Luftqualität

Sören Sparmann

2025-09-14

Pandas

Import

Pandas-Bibliothek für Datenanalyse importieren.

```
import pandas as pd
```

Daten einlesen

Daten aus einer CSV-Datei in einen DataFrame einlesen.

```
# CSV-Datei einlesen
df = pd.read_csv(
    "data/example.csv",
    index_col=0,
    parse_dates=True
)
# Dataframe anzeigen
df
```

date	value	category
2024-01-01	10.5	a
2024-01-02	11.0	a
2024-01-03	10.7	b
2024-01-04	12.2	b
2024-01-05	13.1	b
2024-01-06	12.8	c

Spalten selektieren

Eine bestimmte Spalte aus dem DataFrame auswählen.

```
# Spalte value selektieren
values = df["value"]
values
```

```
date
2024-01-01    10.5
2024-01-02    11.0
2024-01-03    10.7
2024-01-04    12.2
2024-01-05    13.1
2024-01-06    12.8
Name: value, dtype: float64
```

Daten filtern mit Bedingungen

Daten anhand einer Bedingung auswählen (z.B. Werte größer als 11).

```
# Bedingung festlegen
cond = df["value"] > 11
# Bedingung anwenden
filtered = df[cond]
filtered
```

date	value	category
2024-01-04	12.2	b
2024-01-05	13.1	b
2024-01-06	12.8	c

Gruppieren & Aggregieren

Werte nach Kategorien gruppieren und aggregieren (z.B. Mittelwert berechnen).

```
# Daten nach Spalte category gruppieren
grouped = df.groupby("category")
# Mittelwert berechnen
grouped['value'].mean()
```

```
category
a    10.75
b    12.00
c    12.80
Name: value, dtype: float64
```

Resampling

Zeitreihen-Daten nach Zeitintervallen zusammenfassen und aggregieren.

```
# Monatlicher Mittelwert
df.resample("MS")['value'].mean()
# Einträge pro Jahr
df.resample("YS")['value'].count()
```

```
date
2024-01-01    6
Freq: YS-JAN, Name: value, dtype: int64
```

Plotly Express

Daten visualisieren

Import

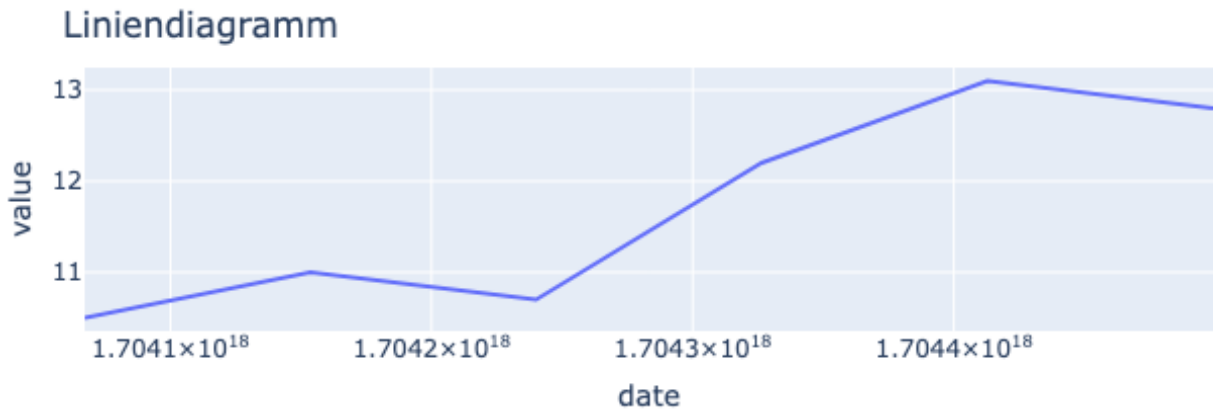
Plotly Express für interaktive Visualisierungen importieren.

```
import plotly.express as px
```

Liniendiagramm

Verlauf einer Variable über die Zeit darstellen.

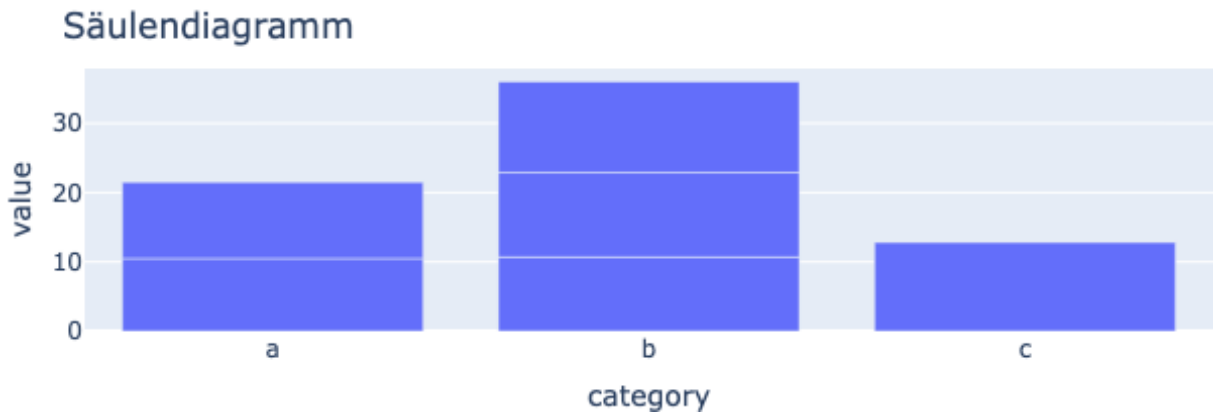
```
fig = px.line(df, y="value", title="Liniendiagramm", height=200)
fig.show()
```



Balken- und Säulendiagramm

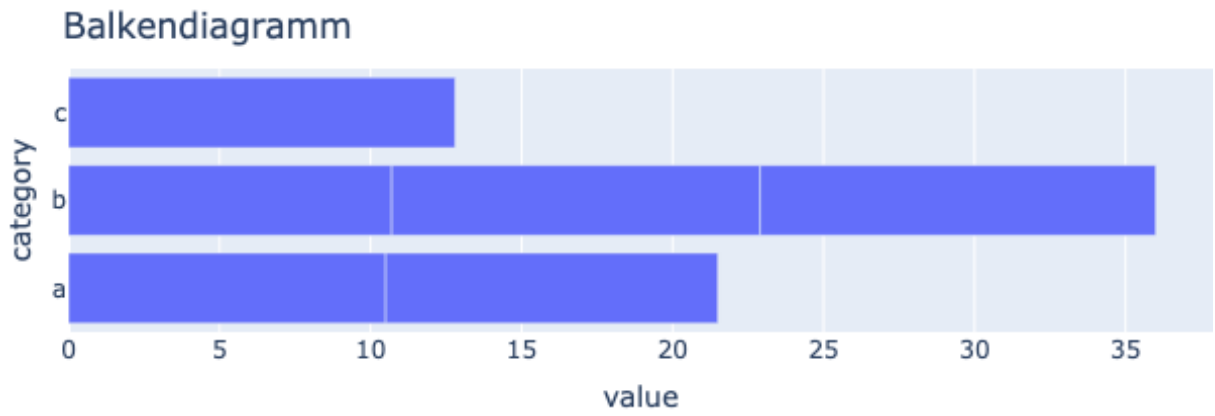
Werte nach Kategorien als Säulen darstellen.

```
fig = px.bar(df, x="category", y="value", title="Säulendiagramm", height=200)
fig.show()
```



Werte nach Kategorien als Balken darstellen.

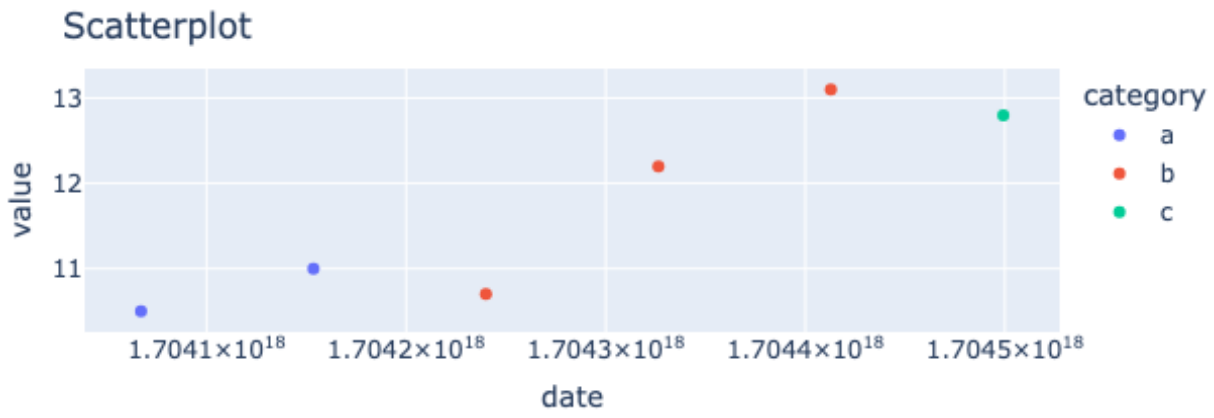
```
fig = px.bar(df, x="value", y="category", title="Balkendiagramm", orientation='h
fig.show()
```



Streudiagramm

Zusammenhang zwischen zwei Variablen darstellen.

```
fig = px.scatter(df, y="value", color="category", title="Scatterplot", height=20
fig.show()
```



Boxplot

Verteilung von Werten innerhalb von Gruppen analysieren.

```
fig = px.box(df, x="category", y="value", title="Boxplot", height=200)
fig.show()
```