

Eine zweite Messgröße, mit eigener Achse

Worum es geht

Temperatur und Feuchte in ein Diagramm zu zeichnen klingt harmlos. Es ist aber eine Behauptung: Wer beide an dieselbe senkrechte Achse hängt, sagt damit, dass 18 Grad und 18 Prozent dasselbe bedeuten. Tun sie nicht.

Die Achse ist die Bedeutung

Eine Zahl im Diagramm hat erst dann einen Sinn, wenn die Achse sagt, wovon sie handelt. Zwei Messgrößen brauchen deshalb zwei Achsen, links die eine, rechts die andere.

Ein Diagramm mit Griff

Bisher hast du `plt.plot()` benutzt, die kurze Schreibweise: matplotlib sucht sich selbst das Diagramm, auf das du zeigst. Für zwei Achsen brauchst du einen **Griff** an das Diagramm, eine Variable:

```
fig, ax1 = plt.subplots()
```

`fig` ist das Blatt Papier, `ax1` das Koordinatensystem darauf. Ab jetzt heißt es `ax1.plot()` statt `plt.plot()`. Dasselbe wie vorher, nur mit Namen.

So geht es

```
import matplotlib.pyplot as plt

# temperatur und feuchte sind zwei Listen von Zahlen,
# aus zwei Anfragen mit unterschiedlichem measurand.

fig, ax1 = plt.subplots()

ax1.plot(temperatur, color="tab:red")
ax1.set_ylabel("Temperatur in Grad Celsius", color="tab:red")

ax2 = ax1.twinx()
ax2.plot(feuchte, color="tab:blue")
ax2.set_ylabel("Feuchte in Prozent", color="tab:blue")

ax1.set_xlabel("Messung")
plt.title("Temperatur und Feuchte über Nacht")
plt.show()
```

`twinx()` macht ein zweites Koordinatensystem, das sich die waagerechte Achse mit dem ersten teilt und eine eigene senkrechte bekommt. Der Name kommt daher: Zwillinge in x.

Zwei Dinge, die du beachten musst

Ohne Farbe ist es unlesbar

Zwei Kurven in einem Bild, und nichts sagt, welche zu welcher Achse gehört. Deshalb steht die Farbe im Code zweimal: einmal an der Kurve, einmal an der Achsenbeschriftung. Erst das macht das Diagramm eindeutig.

Zwei Achsen lassen sich manipulieren

Wer die Bereiche geschickt wählt, kann zwei Kurven fast zur Deckung bringen und damit einen Zusammenhang suggerieren, den es nicht gibt. Das ist ein bekannter Trick in schlechten Grafiken.

Bei Temperatur und Feuchte gibt es den Zusammenhang wirklich, kalte Luft hält weniger Wasser. Nur zeigt das Bild ihn nicht, es passt bloß dazu.

Wenn du weitermachen willst

Eine dritte Messgröße geht auch, braucht aber Abstand, sonst liegen die Beschriftungen übereinander:

```
ax3 = ax1.twinx()  
ax3.spines["right"].set_position(("outward", 50))
```

Spätestens hier lohnt die Frage, ob drei Kurven in einem Bild noch helfen oder ob drei Bilder untereinander besser sind. Die Seite loramint.de/plots bietet genau diese Wahl an.