Eidgenössische Technische Hochschule Zürich

Datenanalyse in der Physik

Spring Semester 2026

Lecture Notes

Contents

1 Einführung in Python **1**
1.1 Ein Erstes Python Programm 1

2 Die Messung als Zufallsprozess **2**

References

[1] Vorlesungsskript

Datenanalyse in der Physik

Fynn Krebser–fkrebser@student.ethz.ch

Spring Semester 2026

Lec 1

1 Einführung in Python

Wir verwenden Python als Programmiersprache, unter anderem, da Python sehr populär ist. Ein Grund dafür ist, dass Python einfach zu lernen und zu lesen ist. Weiter ist Python interaktiver als zum Beispiel C++. Als Programmierumgebung in diesem Kurs verwenden wir Jupyter Notebooks.

1.1 Ein Erstes Python Programm

Betrachten wir das folgende Python Programm:

```
def compute_mean(measurements):
  """Compute mean of a list of measurements."""

  if len(measurements) == 0:
    return 0

  sum = 0
  for value in measurements:
    sum += value
  mean = sum / len(measurements)
  return mean

meaurements = [1, 0.1, 2.0]
compute_mean(meaurements)
```

Kommentare können auf Zwei Arten geschrieben werden. Einzeilige Kommentare beginnen mit einem `#` und mehrzeilige Kommentare werden von drei Anführungszeichen umschlossen.

In Python haben variablen keinen Typ. Dafür müssen Variablen initialisiert werden. Der Typ wird dann automatisch bestimmt.

```
a = 3
a/2
```

In diesem Beispiel erhalten wir 1.5.

Weiter wichtig in Python das Codeblöcke durch Einrückungen definiert werden.

Das if/else statement in Python sieht wie folgt aus:

```
if condition:
  # do something
elif condition2:
  # do something else
else:
  # do something else
```

Ähnlich funktionieren while schleifen:

```
x = 0
while x < 5:
  print(x)
  x = x+1
```

Eine for Schleife sieht wie folgt aus:

```
for x in range(0,5):
  print(x)
```

Hierbei erzeugt range(0,5) eine Liste mit den Zahlen 0,1,2,3,4.

In Python können Listen unterschiedliche Typen besitzen und sehen wie folgt aus:

```
l = [3, "Hund", "Katze"]
```

Die Indexierung ist analog zu C++ 0-basiert. Ausserdem hat das letzte Element Index -1.

Eine hilfreiche Funktion kann die `help` Funktion sein. Diese gibt die Dokumentation eines Objekts zurück. Zum Beispiel:

```
help(l)
```

gibt die Dokumentation der Liste zurück.

In Python ist ALLES eine Referenz. Wenn wir also b=a setzen und a bearbeiten, dann wird auch b bearbeitet.

Auch wichtig sind Tupel. Diese sind ähnlich wie Listen, aber unveränderlich. Sie werden mit runden Klammern definiert:

```
t = (1, 2, 3)
```

Die Klammern sind optional, aber es ist eine gute Praxis sie zu verwenden.

Tupel unpacking ist eine nützliche Funktion, die es ermöglicht, die Werte eines Tupels in einzelne Variablen zu entpacken. Zum Beispiel:

```
number, name = (1, "Hund")
```

Funktionen werden mit dem `def` Schlüsselwort definiert, wie im Einführungsbeispiel. Um mehrere Werte zurückzugeben, kann ein Tupel verwendet werden.

Sinnvolle Wissenschaftliche Bibliotheken beinhalten unter anderem: NumPy, SciPy, Pandas und Matplotlib.

Ein Array wird in NumPy wie folgt erstellt:

```
array_1d = np.array([1, 2, 3])
array_2d = np.array([[1, 2, 3], [4, 5, 6]])
array_range = np.arrange(0,4, step=0.5)
```

Mathematische Operationen können direkt auf Arrays angewendet werden:

```
np.square(array_1d)
np.sqrt(array_1d)
```

Um Matplotlib zu importieren, verwenden wir:

```
import matplotlib.pyplot as plt
```

Lec 2

2 Die Messung als Zufallsprozess

Ein Modell ist eine vereinfachte Vorstellung welches nie vollständig und nicht bewiesen werden kann. Mit einem Experiment können wir es lediglich falsifizieren.

Ein Experiment hingegen ist nie fehlerfrei und auch die Interpretation ist nicht eindeutig.

Betrachten wir als Modell periodische Funktionen. Hierbei ist $\phi(t)$ die Phase und es gilt

$$\tan(\phi) = \frac{\Im(z)}{\Re(z)}.$$

Mit einer linearen zeitabhängigen Phase erhalten wir eine Oszillation der Form

$$x(t) = x_0 \cdot \Re(e^{i\phi(t)}).$$

Der getriebene gedämpfte harmonische Oszillator erfüllt die Gleichung:

$$\ddot{x} + \Gamma\dot{x} + \omega_0^2 x = \frac{F_0 \cos(\omega t)}{m}.$$

Nach einer langen Zeit erreicht die Lösung die Form

$$x_0(\omega) = \left| \frac{F_0/m}{\omega_0^2 - \omega^2 + i\Gamma\omega} \right|.$$

Wenn wir einen Quarzkristall haben, dann sieht die Resonanzkurve wie folgt aus: INSERT FIGURE

Der unterschied zur erwarteten Kurve entsteht dadurch, dass die beiden Platten einen Kondensator bilden, welcher die Resonanzfrequenz verschiebt. somit ist

$$I_{\text{in}}(\omega) \propto |x(\omega) + K(\omega)| \neq x_0.$$

Somit haben wir einen systematischen Fehler begangen. Das Modell war nicht vollständig. Mit einem besseren Modell können wir diesen Fehler korrigieren.

Definition 2.1: Systematischer Fehler

Ein Fehler, welcher durch ein unvollständiges Modell entsteht, wird als systematischer Fehler bezeichnet. Er tritt bei jeder Messung auf und kann durch ein verbessertes Modell korrigiert werden.

Bei einer Messung können auch zufällige Fehler auftreten, was als RAUSCHEN bezeichnet wird. Wir simulieren uns einen Analog-Digital-Converter (ADC). Hierbei gibt es eine zeitliche Auflösung, eine maximal Messbare Spannung und eine Anzahl von Bits, welche die Auflösung der Messung bestimmt.

Wir definieren folgende Grössen:

- Zeitlicher Abstand zwischen Messungen: Δ_t
- Sampling Rate / Abtastrate: $f_{\text{sample}} = 1/\Delta_t$
- Bandbreite der Messung: $f_{\text{bandwidth}} = \frac{1}{2\Delta_t}$
- Totale Messzeit: t_{tot}
- Spektrale Auflösung: $\Delta f = 1/t_{\text{tot}}$
- Signalauflösung: U_{min}
- Signalbereich: U_{max}
- Rauschen der Messapparatur: U_{noise}

Die Meisten dieser Dinge führen zu statistischen Fehlern, welche durch wiederholte Messungen korrigiert werden können.

Index

rauschen, 2