

## Einführung in die Datenauswertung mit Python im physikalischen Praktikum für Lehramt- und Nebenfachstudierende

Maximilian Kühlkamp\*, Ralf Detemple\*, Dominik Dorsel\*, Heidrun Heinke\*

\*I. Physikalisches Institut IA, RWTH Aachen University  
maximilian.kuehlkamp@rwth-aachen.de

### Kurzfassung

In einer Zeit, in der die programmiersprachenbasierte Datenauswertung die Naturwissenschaften dominiert und informatische Grundkonzepte in der schulischen Bildung immer wichtiger werden, ist es auch für angehende Physik-Lehrkräfte wichtig, in diesen Bereichen ausreichende Kompetenzen zu erwerben. Auf Grundlage dessen wurde ein Modul zur Datenauswertung mit Python für Lehramtstudierende im physikalischen Praktikum entwickelt. In dem Praktikumsversuch sollen die Lehramtstudierenden mit Jupyter-Notebooks in die Grundlagen der Datenauswertung mit Python eingeführt und unter Verwendung eines Scaffolding-Ansatzes zum selbstständigen Umgang mit Python in zukünftigen Auswertungen befähigt werden. In diesem Beitrag wird das entwickelte Modul zur Einführung in die Datenauswertung mit Python vorgestellt und der geplante Einsatz im physikalischen Praktikum für Lehramtstudierende beschrieben. Zudem werden die bereits durch den Einsatz in einem freiwilligen Tutorium für Lehramtstudierende und im physikalischen Nebenfachpraktikum für Biologie- und Chemiestudierende gesammelten Erfahrungen diskutiert.

### 1. Motivation

Die digitale Messdatenanalyse ist eine wichtige Kompetenz angehender Naturwissenschaftler/innen. Vor allem in der Physik, aber auch in den anderen Naturwissenschaften wie z. B. der Chemie oder den Geowissenschaften, ist dabei eine programmiersprachenbasierte Datenanalyse längst etablierte Praxis. In dem Bachelorstudiengang Physik an der RWTH Aachen erwerben Studierende aus diesem Grund in einem Modul zur Einführung in das Programmieren Grundfertigkeiten, die sie in physikalischen Praktika in Anwendungsaufgaben nutzen können. Studierende der anderen Naturwissenschaften müssen in ihrem Studienverlauf ebenfalls ein physikalisches Praktikum absolvieren, um dort physikalische sowie experimentelle Kompetenzen zu erwerben. Die höchstrelevante und aktuelle programmiersprachenbasierte Datenanalyse ist dort bislang jedoch kein Inhalt.

Auch in der schulischen Bildung werden die digitalen Grundkompetenzen immer populärer (vgl. KMK, 2017). Zu diesen Kompetenzen zählen z. B. ein algorithmisches Verständnis sowie die Problemlösefähigkeit (vgl. Dörge, 2015, S. 347-349). Überraschenderweise sind solche informatischen Grundkompetenzen ebenfalls bislang kein zentraler Gegenstand der naturwissenschaftlichen Lehramtsausbildung. Sie könnten aber durch das Erlernen einer Programmiersprache zur Datenauswertung zumindest teilweise abgedeckt werden.

Diese Umstände sowie eine sinnvolle Kombination aus Messdatenaufnahme und -analyse führten zu der

Entwicklung eines Praktikumsversuchs zur Einführung in die Datenauswertung mit Python, der sowohl für Lehramtstudierende mit dem Unterrichtsfach Physik als auch für sogenannte Nebenfachstudierende verschiedener Studiengänge zum Einsatz kommen soll.

Die Wahl der Programmiersprache fiel auf Python, da Python die am meisten verwendete Programmiersprache weltweit ist (vgl. Statista Research Department, 2024) und vor allem in der Datenanalyse und dem Maschinernen Anwendung findet. Die simple Syntax und die Modularität mit hilfreichen externen Paketen für die Datenauswertung ermöglichen einen vergleichsweise einfachen Einstieg in die programmiersprachenbasierte Datenanalyse. In Kombination mit Jupyter-Notebooks lassen sich Auswertungsschritte direkt im selben Dokument mit Erklärungen und Hilfestellungen beschreiben.

### 2. Fachdidaktische Ansätze

Der Entwicklung des Moduls sowie einigen Umsetzungen der Inhalte liegen die folgenden fachdidaktischen Ansätze zugrunde:

Bei der Entwicklung wurde sich an dem Modell der didaktischen Rekonstruktion nach Kattmann orientiert (vgl. Kattmann et al., 1997) und ein Design-Based Research-Ansatz gewählt. Beides sind iterative Prozesse aus Design, Evaluation und Re-Design. Das Modell der didaktischen Rekonstruktion legt dabei den Fokus darauf, die Sichtweisen der Lernenden und

die Ergebnisse der fachlichen Klärung in der didaktischen Strukturierung wechselseitig zu verknüpfen. Das heißt, dass die zu vermittelnden fachwissenschaftlichen Inhalte nicht ohne eine Anpassung an die Vorkenntnisse, Weltanschauungen und verinnerlichten Werte der Lernenden in einem Unterrichtsgefüge integriert werden sollten (vgl. Kattmann et al., 1997). Wichtig sind somit vor der Entwicklung des Moduls eine Erhebung der Lernendenperspektive und anschließend die Abstimmung der Inhalte auf die Ergebnisse der Erhebung.

Durch die Verwendung eines Design-Based Research-Ansatzes wird sichergestellt, dass anhand von Evaluation und Re-Design nach jeder Durchführung die ursprüngliche Intervention (hier die Einführung in Python) optimiert wird und ein Beitrag zur Theorieentwicklung geleistet werden kann (vgl. Gess et al., 2014).

Der Aufbau des entwickelten Moduls folgt einem Scaffolding-Ansatz. Scaffolding beschreibt ein Gerüst an Hilfestellungen, das die Lernenden dazu befähigt, ein Problem zu lösen, welches sie ohne die Hilfestellungen nicht hätten lösen können. Um eine Selbstständigkeit der Lernenden zu erreichen, muss das Gerüst aus Hilfestellungen schrittweise abgebaut werden (vgl. Wood et al., 1976). Die Umsetzung des Scaffolding-Ansatzes im Modul wird in Abschnitt 5.1 näher ausgeführt.

Bei der Umsetzung der Inhalte wurden neben dem Scaffolding-Ansatz auch die didaktischen Prinzipien (vgl. Baumann, 1996) bzw. die Prinzipien didaktischen Handelns (vgl. Hubwieser, 2007) berücksichtigt. Diese grundlegenden Prinzipien unterstützen die Wahl der Methodik und Darstellung der Inhalte. Ebenfalls sind die Erkenntnisse des ARCS-Modells in die Entwicklung eingeflossen. Die vier Bedingungen *attention*, *relevance*, *confidence* und *satisfaction* müssen nach Keller erfüllt sein, damit eine Person wirklich motiviert ist (vgl. Keller, 1987). Der Einfluss des Modells in die Entwicklung lässt sich z. B. in der Einleitung der Praktikumsanleitung des entwickelten Moduls wiederfinden.

### 3. Erhebung zum Interesse und Vorwissen von Lehramtstudierenden

Gemäß der Berücksichtigung der Lernendenperspektive im Modell der didaktischen Rekonstruktion wurde vor der Entwicklung des Moduls eine Erhebung zum Interesse und Vorwissen von Physik-Lehramtstudierenden zur Programmierung durchgeführt. Es konnten dabei insgesamt 29 Studierende befragt werden. Da die Umfrage während ausgewählter Veranstaltungen stattfand, teilt sich die Stichprobe in 9 Studierende des 4. Semesters und 20 Studierende des 6. oder eines höheren Semesters. Trotz der relativ kleinen Stichprobe konnten durch die Durchführung in Pflichtveranstaltungen zwei Kohorten des aktuell

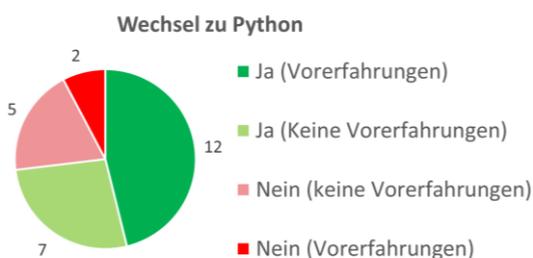
kleinen Physik-Lehramtstudiengangs für die Schulform Gymnasien und Gesamtschulen fast in ihrer Gänze abgebildet werden.

Die Erhebung orientierte sich an den folgenden drei Leitfragen:

1. Welches Interesse besteht an der Datenauswertung mit Python?
2. Welche Vorerfahrungen im Programmieren und im Umgang mit Python besitzen die Studierenden bereits?
3. Welche Erwartungen und Wünsche hätten die Studierenden an ein Modul zur Einführung in die Datenauswertung mit Python?

#### 3.1. Interesse

Die Erhebung ergab, dass 90 % der Physik-Lehramtstudierenden ihre Daten mit Excel (25<sup>1</sup>) oder Origin (1) auswerten. Von den 3 Studierenden, die ihre Messdaten bereits mit Python auswerten, studieren 2 als zweites Lehramtsfach Informatik. Die für die Entwicklung essenzielle und überraschende Erkenntnis war, dass 73 % (19) der Studierenden, die ihre Messdaten nicht bereits mit Python auswerten, angaben, dass sie nach einer expliziten Einführung zu einer Auswertung mit Python wechseln würden. Auffällig ist dabei auch die Aufteilung der Angaben der Studierenden bei bestehenden oder fehlenden Vorkenntnissen (s. Abbildung 1).



**Abb. 1:** Aufteilung der Angaben der Studierenden zu einem möglichen Wechsel zur Verwendung von Python nach einer Einführung in „Ja“ (grün) und „Nein“ (rot) bei bestehenden oder fehlenden Vorkenntnissen. N = 26

#### 3.2. Vorerfahrungen

Der erste mögliche Kontaktpunkt mit textueller Programmierung ist häufig der Informatikunterricht in der Schule. 76 % (22) der Befragten gaben an, bereits in der Schule Informatikunterricht besucht zu haben, davon jedoch 55 % (12) nur in der Sekundarstufe 1, in welcher die textuelle Programmierung eher selten behandelt wird und Python bis jetzt kein Inhalt des Kernlehrplans Informatik in NRW ist (vgl. Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen, 2023). 59 % (17) der Studierenden gaben an, bereits Erfahrungen mit Python gesammelt zu haben. Die häufigsten Angaben für die Herkunft der Vorerfahrungen waren Übungsblätter der Pflichtveranstaltungen Experimentalphysik 1 und 2 sowie der

<sup>1</sup> Anzahl der Personen

Einführung in die Theoretische Physik, die Inhalte einer Mathematik für Physiker/innen vermittelt. Die Frage zur Selbsteinschätzung im Umgang mit Python zur Datenauswertung zeigt mit einem Wert von ca. 2,3 auf einer sechsstufigen Likert-Skala von 1 (unsicher) bis 6 (sicher) jedoch, dass die vermittelten Kompetenzen in diesen Veranstaltungen nicht für ein selbstbewusstes Anwenden bei der Datenauswertung ausreichen. Werden die 3 Studierenden, die Python bereits zur Datenauswertung verwenden, aus dieser Stichprobe ausgenommen, liegt der numerische Wert für die Selbsteinschätzung sogar nur bei 1,7.

### 3.3. Erwartungen und Wünsche

Bei der Erfragung der Erwartungen und Wünsche an das Modul zur Einführung in Python zur Datenauswertung wurden verschiedene Aspekte genannt. Die häufigsten Nennungen waren dabei, dass das Modul

- eine grundlegende Einführung in die Datenauswertung bieten soll.
- die Grundkompetenzen von Python vermitteln soll.
- Vorteile der Datenauswertung mit Python aufzeigen soll.

### 3.4. Folgerungen für die Entwicklung des Moduls

Eine wichtige Erkenntnis für die Entwicklung des Moduls ist, dass die Zielgruppe eine große Heterogenität in Bezug auf die Vorerfahrungen im Umgang mit Python bei der Datenauswertung aufweist. Das Modul muss daher, wie auch in den Wünschen geäußert, eine grundlegende Einführung in Python als textuelle Programmiersprache bieten. Auch auffällig ist, dass die Akzeptanz eines Wechsels des Auswertungsmediums größer ist, wenn bereits erste Erfahrungen mit Python gemacht wurden (s. Abbildung 1). Eine Einführung in Python sollte deswegen die Vorteile der Nutzung von Python zur Datenauswertung im Vergleich zu Programmen wie Excel aufzeigen und verdeutlichen.

### 4. Ziele und Inhalte des Moduls

Aus den Ergebnissen der Erhebung sowie der fachlichen Klärung können die folgenden Ziele für das Modul definiert werden.

In erster Linie ist das Ziel des Moduls, den Studierenden das selbstständige Auswerten ihrer Messdaten mit Python in Praktikumsversuchen zu ermöglichen. Dabei lernen die Studierenden grundlegende Kompetenzen der Informatik kennen. Dazu gehören z. B. die Problemlösung mit den Teilkompetenzen der Recherche und des Debuggings sowie eine algorithmische Denkweise. Ein weiteres Ziel des Moduls ist es, für

die Studierenden die Vorteile der Verwendung von Python zur Datenauswertung sichtbar zu machen. Ein wichtiger Aspekt ist dabei die Verwendung von Paketen zur Automatisierung von Auswertungsschritten, vor allem bei der Betrachtung von Messunsicherheiten.

Inhalte des Moduls sind dementsprechend einerseits die Grundlagen von Python (Syntax, Datentypen/-strukturen, Funktionen, Pakete) und andererseits datenauswertungsspezifische Kompetenzen (Einlesen und Plotten von Messdaten, Fitten mathematischer Funktionen, Betrachtung und Berechnung von Unsicherheiten). Für die Datengrundlage wird das Experiment Federpendel eingesetzt. Die Studierenden sollen mit Hilfe von phyphox<sup>2</sup> den Beschleunigungsverlauf verschiedener Massen an einem Federpendel messen und anschließend in einem Jupyter-Notebook die Federkonstante bestimmen.

### 5. Beschreibung des Moduls

Das Modul besteht aus drei Produkten: einer Versuchsanleitung und zwei Jupyter-Notebooks. In der Versuchsanleitung wird das Modul motiviert, eine Anleitung zur Installation von Anaconda<sup>3</sup> und Visual Studio Code<sup>4</sup> gegeben sowie der Versuch Federpendel erklärt. Der Link zum GitHub-Repository des Moduls ist im Anhang des Beitrags zu finden.

#### 5.1. Scaffolding

Zur Umsetzung eines Scaffolding-Ansatzes wird das Modul in zwei Teile mit jeweils einem Jupyter-Notebook gegliedert (s. Abbildung 2).

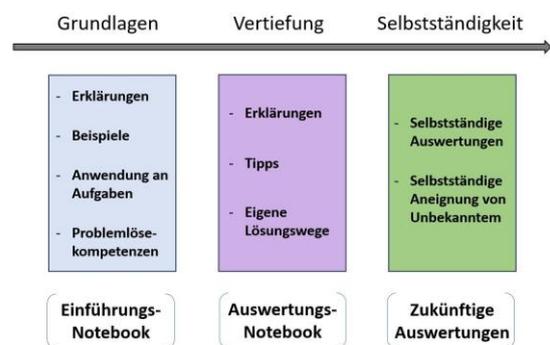


Abb. 2: Darstellung der Umsetzung des Scaffolding-Ansatzes im entwickelten Modul.

Im ersten Teil lernen die Studierenden mit Hilfe des Einführungsnotebooks die Grundlagen von Python und Jupyter-Notebooks kennen. Dort werden zusätzlich schon grundlegende Funktionen wie das Einlesen

<sup>2</sup> Smartphone-App zur Messdatenaufnahme mit den im Smartphone verbauten Sensoren. Website: <https://phyphox.org/de/home-de/> (Stand: 19.05.2024)

<sup>3</sup> Gängige Python-Distribution mit vielen für die Datenanalyse vorinstallierten Paketen. Website: <https://www.anaconda.com/> (Stand: 21.05.2024)

<sup>4</sup> Alternative zu Jupyter für die Bearbeitung von Notebooks. Visual Studio Code bietet gegenüber Jupyter viele nützliche Funktionen wie z. B. die Codevervollständigung von IntelliSense. Website: <https://code.visualstudio.com/> (Stand: 21.05.2024)

und Plotten von Messdaten thematisiert. Hilfestellungen, Beispiele und vertiefende Aufgaben sollen dabei den Kompetenzerwerb fördern. Im zweiten Teil, dem Datenauswertungsnotebook, werden die Messdaten des Federpendelversuchs ausgewertet. Hier werden die Hilfestellungen schrittweise abgebaut, um gemäß dem Scaffolding-Ansatz eine zunehmende Selbstständigkeit der Studierenden zu ermöglichen.

### 5.2. Einführungsnotebook

Das Einführungsnotebook ist als Vorbereitung auf das Datenauswertungsnotebook gedacht und ist so aufgebaut, dass ein neuer Inhalt (z. B. Datentypen) zuerst anhand von Erläuterungen und Beispielen eingeführt wird und anschließend mit Hilfe einer Aufgabe gefestigt werden kann.

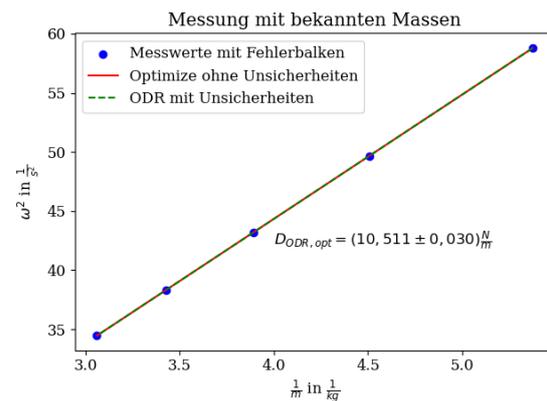
Das Notebook orientiert sich inhaltlich teilweise an dem Buch „Mit Jupyter durchs Physikpraktikum“ von Lew Classen (vgl. Classen, 2022). Das Buch bietet eine ausführliche Grundlage zur Einführung in die Datenauswertung mit Python, hat aber durch die fehlende Interaktivität eines Buches Nachteile gegenüber einer Umsetzung als Jupyter-Notebook. Ein Vorteil von Notebooks ist z. B., dass die Code-Beispiele von den Studierenden bearbeitet und dadurch Unklarheiten durch Abänderungen des Beispiels selbstständig überprüft werden können. Die Inhalte des Buches wurden zusätzlich mit wichtigen Kompetenzen wie dem Debugging und dem Recherchieren von Funktionsdokumentationen erweitert.

### 5.3. Datenauswertungsnotebook

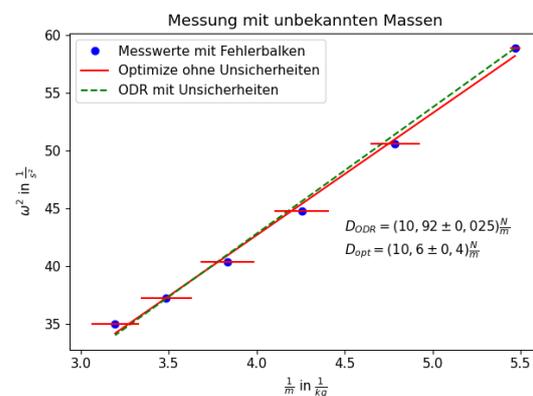
Das Datenauswertungsnotebook stellt eine Schritt-für-Schritt-Anleitung zur Auswertung des Versuchs Federpendel dar. Die einzelnen Auswertungsschritte und die dafür benötigten Funktionen bzw. Pakete werden in einem Text ausführlich beschrieben. Die Aufgabe der Studierenden ist es, mit den im Einführungsnotebook erlangten Kompetenzen und den Beschreibungen der Auswertungsschritte selbstständig den für die Auswertung benötigten Code zu schreiben.

Eine Besonderheit des auszuwertenden Versuchs ist die Verwendung von „unbekannten“ Massen bei der Messung der Beschleunigung am Federpendel. Durch die sehr genauen Messungen von phyphox und die Verwendung einer Fit-Methode in Python würden die Studierenden bei vorher gewogenen Massen sehr kleine Messunsicherheiten und eine nahezu perfekte lineare Funktion bei der geeignet linearisierten Auftragung der quadrierten Kreisfrequenz  $\omega^2$  als Funktion der reziproken Masse  $1/m$  erhalten (s. Abbildung 3). Da jedoch ein Inhalt des Notebooks der Unterschied zwischen den beiden in Python gängigen Fit-Methoden `optimize.curve_fit`<sup>5</sup> und `ODR`<sup>6</sup> ist, ist eine

kleine Abweichung der Messdaten von den theoretischen Erwartungen gewünscht. Aus diesem Grund wurden mit einem 3D-Drucker fünf gleichaussehende Massstücke mit unterschiedlichen Füllichten produziert. Die Studierenden sollen die Massstücke nicht auswiegen, sondern erhalten für die Berechnungen das durchschnittliche Gewicht aller Massstücke mit dem Bereich der Abweichungen als Unsicherheit. Durch diese synthetisch erzeugte große Unsicherheit kann der Unterschied zwischen den beiden Fit-Methoden besser verdeutlicht werden (s. Abbildung 4).



**Abb. 3:** Diagramm zur Bestimmung der Federkonstante bei bekannten Massen. Auf der x-Achse  $m^{-1}$  in  $kg^{-1}$  mit der Masse  $m$  und auf der y-Achse  $\omega^2$  in  $s^{-2}$  mit der Kreisfrequenz  $\omega$ . Beide linearen Fits der Methoden `optimize.curve_fit` (hier `Optimize`) (rot) und `ODR` (grün gestrichelt) liegen übereinander und führen zum gleichen Ergebnis für die Federkonstante.



**Abb. 4:** Gleiche Darstellung wie in Abbildung 3, jedoch nun mit den unbekanntem Massstücken sowie entsprechend sichtbaren Fehlerbalken und Abweichungen der Datenpunkte. Da `ODR` auch die Unsicherheiten in x-Richtung berücksichtigen kann, führen die beiden Fit-Methoden hier zu unterschiedlichen Federkonstanten.

<sup>5</sup> Methode des Pakets `scipy` zum Fitten von Messdaten. Ermöglicht nur die Übergabe von Unsicherheiten in y-Richtung.

<sup>6</sup> Ebenfalls Methode des Pakets `scipy` zum Fitten von Messdaten. Ermöglicht jedoch das Übergeben von Unsicherheiten in x- und y-Richtung.

Aufgrund von gesammelten Erfahrungen aus dem Einsatz im physikalischen Nebenfachpraktikum (s. Abschnitt 7) wurde das Datenauswertungsnotebook zur weiteren Differenzierung um Knöpfe mit Hilfestellungen erweitert. Die Knöpfe wurden an Stellen mit häufig auftretenden Schwierigkeiten platziert, damit die Studierenden nach eigenem Ermessen Hilfen einholen können, ohne zwingend eine/n Tutor/in konsultieren zu müssen. Das Skript für die Hilfestellungen ist zur besseren Übersichtlichkeit des Notebooks in einer zusätzlichen Datei ausgegliedert und kann am Anfang des Notebooks durch das Ausführen einer dafür vorgesehenen Code-Zelle eingebunden werden.

## 6. Einsatz in der Lehramtsausbildung

Der erste reguläre Einsatz des Moduls für Lehramtstudierende der Physik ist für das Kompetenzpraktikum (s. Abschnitt 6.1) im Sommersemester 2024 geplant. Erste Erfahrungen mit Physik-Lehramtstudierenden konnten jedoch schon bei einem freiwilligen Tutorium für Studierende des 5. Semesters gesammelt werden (s. Abschnitt 6.3).

### 6.1. Rahmenbedingungen des Kompetenzpraktikums

Das Modul wurde für den Einsatz im Kompetenzpraktikum, dem ersten von insgesamt drei Teilen im Grundpraktikum für Physik-Lehramtstudierende der RWTH Aachen, entwickelt. Das Kompetenzpraktikum ist nach der Prüfungsordnung von 2023 im 2. Semester verortet. Im Kompetenzpraktikum erlernen die Studierenden die Grundlagen des Experimentierens. Für das Absolvieren des Praktikums erhalten die Studierenden 3 ECTS Credits. Es ist somit einen Arbeitsaufwand von insgesamt ca. 90 Stunden vorgesehen. Bei einer Dauer von 13 Wochen ergibt sich ein wöchentlicher Umfang von ca. 7 Stunden mit einer Präsenzzeit am Praktikumstag von 2 Stunden und 15 Minuten. Die restliche Zeit kann für die Vorbereitung auf den Praktikumstag und die Nachbereitung bzw. Erstellung eines Versuchsprotokolls verwendet werden.

Das Kompetenzpraktikum startet für alle Studierende mit einer Einführung in die Grundlagen von Messdaten und ihren Unsicherheiten im Versuch MEDA (MEssDaten). Darauf folgt in der zweiten Woche ergänzend ein Tutorium zur Datenauswertung mit Excel. Nach dem zweiwöchigen Einführungsblock durchlaufen die Studierenden in Gruppen von bis zu 8 Personen zehn Wochen lang wöchentliche Versuche. In der Regel muss bei den regulären Versuchen für die erfolgreiche Teilnahme ein Versuchsbericht angefertigt und abgegeben werden. Das Praktikum wird am Ende der 13 Wochen mit einer 15-minütigen mündlichen Prüfung mit experimentellen Anteilen abgeschlossen.

### 6.2. Geplanter Einsatz

Das entwickelte Modul zur Einführung in die Datenauswertung mit Python ist als übergreifendes Tuto-

rium für alle Studierenden in der Mitte des Kompetenzpraktikums geplant. Auf diese Weise können die Studierenden erste Erfahrungen im Experimentieren sammeln und sich an die Gegebenheiten des Praktikums gewöhnen, ohne direkt zu Beginn von dem Lernen einer Programmiersprache überfordert und abgeschreckt zu werden. Gleichzeitig finden nach dem Tutorium noch genügend reguläre Versuche statt, in denen der Umgang mit Python zur Datenauswertung erprobt und gefestigt werden kann.

Die Studierenden sollen als Vorbereitung auf den Versuchstag gemäß der Praktikumsanleitung alles für die Datenauswertung Benötigte installieren und das Einführungsnotebook vollständig bearbeiten. Am Versuchstag werden zu Beginn Fragen und aufgetretene Probleme der Vorbereitung geklärt. Anschließend wird der Versuch Federpendel durchgeführt und die Messdaten werden aufgenommen. Diese werden dann in dem Datenauswertungsnotebook ausgewertet. Während der Präsenzzeit steht der Tutor bzw. die Tutorin für Fragen zur Verfügung. Das Datenauswertungsnotebook kann, wenn nötig, zuhause vervollständigt werden.

### 6.3. Erfahrungen aus freiwilligem Tutorium

Anfang April 2024 wurde das Modul für 9 freiwillige Studierende des 5. Semesters als Vorbereitung auf das Fortgeschrittenenpraktikum (letztes Praktikum des Physik-Lehramtstudiums im 6. Semester) in der Form eines Tutoriums angeboten. Das Tutorium hat nach Absprache mit den Teilnehmenden über den Verlauf eines Tages in Präsenz stattgefunden. Morgens wurde innerhalb von ca. 3 Stunden die Vorbereitung (Installation und Einführungsnotebook) abgeschlossen. Nach einer Mittagspause wurden die Messdaten am Federpendel aufgenommen (ca. 25 Minuten) und anschließend im Datenauswertungsnotebook ausgewertet. Nach ca. 3 weiteren Stunden war das Tutorium beendet.

Die Vorbereitung sowie die Messdatenaufnahme verliefen problemlos und es gab durchweg gutes Feedback. Erste Schwierigkeiten traten bei der Datenauswertung auf. Durch den Abbau der Hilfestellung zwischen Einführungs- und Datenauswertungsnotebook waren die Studierenden auch mit Hilfestellungen häufig mit Syntaxfehlern oder der falschen Verwendung der durch die Pakete gelieferten Funktionen (z. B. fehlende zu übergebende Parameter) beschäftigt. Das Auftreten und Lösen solcher Probleme führte teilweise zu Frustration, teilweise auch zu Euphorie. Es kostete aber in vielen Fällen so viel Zeit, dass die meisten Teilnehmenden die Datenauswertung in Präsenz aus zeitlichen Gründen nicht abschließen konnten. Trotz der genannten Schwierigkeiten war eine durchgehend intensive Beschäftigung mit der Auswertung erkennbar und das Feedback war insgesamt positiv. Die Erfahrungen überschneiden sich an einigen Stellen mit denen aus der Durchführung im Nebenfachpraktikum, weswegen in Abschnitt 8 ein zusammenfassendes Fazit gezogen wird.

## 7. Einsatz in Physikpraktika für Nebenfachstudierende

Nach der Entwicklung des Moduls wurde als weiteres Anwendungsfeld der Einsatz im physikalischen Praktikum für Chemie- und Biologie-Studierende identifiziert und in einer ersten Durchführung pilotiert. Die Überlegung war zunächst, die Einführung in Python für eine Freiwilligen-Gruppe mit 8 bis 10 Personen je Studiengang anzubieten. Die Anmeldungen zeigten jedoch, dass das Interesse an einer solchen Einführung weit größer war als vorher angenommen. Insgesamt 80 Studierende (30 von insgesamt 133 Biologie-Studierenden und 50 von insgesamt 106 Chemie-Studierenden) waren gewillt, das Modul durchzuführen. Entsprechend wurde das Modul im Wintersemester 2023/24 in 3 Biologie- und 5 Chemie-Gruppen mit insgesamt 80 Teilnehmenden eingesetzt und evaluiert.

### 7.1. Rahmenbedingungen des physikalischen Nebenfachpraktikums

Das physikalische Praktikum für Chemie- und Biologie-Studierenden ist wie das Kompetenzpraktikum für Lehramtstudierende mit 3 ECTS Credits versehen. Die Präsenzzeit ist jedoch mit 3 Stunden pro Woche länger als das Kompetenzpraktikum im Lehramtstudium. Dadurch sinkt die Zeit für Vor- und Nachbereitung auf 4 Stunden, was durch einen weniger umfangreichen Versuchsbericht ausgeglichen wird. Das Praktikum startet in Kleingruppen von 8 Personen mit einem zweiwöchigen Einführungsblock aus dem Versuch MEDA (MEssDATen) und dem Tutorium TDA (Tutorium Datenauswertung mit Excel). Darauf folgen 9 reguläre Versuche mit zusätzlich 2 Tutorien, in welchen die Studierenden auf die nachfolgenden Versuche vorbereitet werden. Für die Gruppen, die das Modul zur Einführung in Python besuchen, wird das Tutorium TDA mit dem Versuch PYT ersetzt.

### 7.2. Ablauf des Versuchs PYT

Die Studierenden bereiten sich zu Hause mithilfe der Praktikumsanleitung und dem Einführungsnotebook auf den Praktikumstag vor. Sie installieren vor dem Praktikumstag alle benötigten Programme und bearbeiten vollständig das Einführungsnotebook. Der Betreuer bzw. die Betreuerin steht per Mail für Fragen zur Verfügung. Um mehr Zeit für die Datenauswertung zu schaffen und fehlerbehaftete Messungen zu vermeiden, nehmen die Studierenden die Messdaten nicht selbst auf, sondern erhalten im Vorfeld eine geeignete Messreihe zum Versuch Federpendel. Die Messdaten sollen dann am Praktikumstag im Datenauswertungsnotebook ausgewertet werden. Für die erfolgreiche Teilnahme am Versuch ist neben der Anwesenheit eine sichtbare, hinreichend intensive Beschäftigung mit den Versuchsinhalten ausreichend. Durch die in Abschnitt 3 angesprochene große Heterogenität der Vorkenntnisse in Bezug auf schulische Vorerfahrungen muss berücksichtigt werden, dass trotz differenzierter Hilfestellungen nicht erwartet

werden kann, dass alle Studierenden gleichermaßen das Lernziel erreichen.

### 7.3. Erfahrungen aus dem Praktikum

Wie auch bei der Durchführung als Tutorium für Lehramtstudierende (s. Abschnitt 6.3) fällt das Feedback zum Vorbereitungsmaterial durchweg positiv aus. Zu Beginn des Versuchstages fühlen sich die Studierenden gut auf den Praktikumstag vorbereitet und es gibt wenig Unklarheiten zu besprechen. Die Vorbereitungszeit liegt im Durchschnitt zwischen 2 und 4 Stunden.

Am Versuchstag hatten die Studierenden sichtlich Schwierigkeiten, die bei der Vorbereitung erlangten Kompetenzen in der Datenauswertung umzusetzen. Ein häufiger Grund dafür war, dass Erklärungen und Anweisungen nicht ausreichend genau durchgelesen oder sogar direkt übersprungen wurden. Aus diesem Grund wiederholten sich die Hilfestellungen der Tutorinnen bzw. Tutoren oft bei verschiedenen Personen. Diese Erkenntnis führte zur Implementierung der in Abschnitt 5.3 angesprochenen Knöpfe mit Hilfestellungen, die die Tutorinnen und Tutoren entlasten sollen.

Als Feedback äußerten die Studierenden, dass der Umfang des Datenauswertungsnotebooks möglicherweise zu groß für einen einzelnen Versuchstermin sei, weswegen eine Aufteilung auf zwei Termine sinnvoll wäre.

Am Ende des Versuchstages wurde in einem Fragebogen unter anderem erhoben, ob die Studierenden versuchen werden, die zukünftigen Versuche mit Python auszuwerten und ob sie den Versuch PYT ihren Kommilitoninnen und Kommilitonen weiterempfehlen würden. Bei einer Stichprobe von  $N = 55$  gaben ca. 73 % (40) der Studierenden nach der Einführung in Python an, dass sie versuchen werden, die zukünftigen Versuche mit Python auszuwerten. Trotz der in der Pilotierung beobachteten Schwierigkeiten, die vor allem den zeitlichen Umfang und die verfügbaren Hilfen zur eigenständigen Problemlösung betrafen, würden sogar ca. 82 % (45) der befragten Studierenden das Modul ihren Kommilitoninnen und Kommilitonen weiterempfehlen.

## 8. Zusammenfassung und Ausblick

Aufgrund der zunehmenden Bedeutung informatischer Kompetenzen und der programmiersprachenbasierten Datenanalyse wurde ein Modul zur Datenauswertung mit Python für physikalische Praktika entworfen und in diesem Beitrag vorgestellt. In dem Modul lernen die Studierenden in zwei Jupyter-Notebooks zuerst die Grundlagen von Python und anschließend die Datenauswertung mit Python anhand eines Versuchs zum Federpendel kennen. Durch einen Scaffolding-Ansatz sollen sie zur selbstständigen Datenauswertung mit Python befähigt werden.

Das Modul besteht aus einer Vorbereitung und der eigentlichen Datenauswertung. In der Vorbereitung

sollen (in der Regel zuhause) anhand der Praktikumsanleitung alle nötigen Programme installiert und das Einführungsnotebook mit den Grundlagen von Python bearbeitet werden. Am Praktikumstag werden die Messdaten am Versuch zum Federpendel aufgenommen (bzw. im Nebenfachpraktikum bereitgestellt) und anschließend im Datenauswertungsnotebook ausgewertet.

Eine Erhebung unter Lehramtstudierenden des 4. und höheren Semesters ergab, dass die Mehrheit der Studierenden nach einer Einführung in Python durch ein solches Modul von der Verwendung von Excel für die Datenauswertung zu Python wechseln würden. Zusätzlich konnte durch die Erhebung gemäß dem Modell der didaktischen Rekonstruktion die Lernendenperspektive in Erfahrung gebracht und bei der Entwicklung des Moduls berücksichtigt werden.

Ursprünglich war der Einsatz des Moduls zunächst nur für das Kompetenzpraktikum im 2. Semester des Lehramtstudiums Physik angedacht. Ein Test der Akzeptanz des Moduls im Rahmen des Nebenfachpraktikums für Studierende der Biologie und Chemie erbrachte ein überraschend großes Interesse dieser Studierenden an einer Einführung in Python zur Datenauswertung und führte zum Einsatz im physikalischen Nebenfachpraktikum. Dort wurde das Modul im Wintersemester 2023/24 von insgesamt 80 Biologie- und Chemiestudierenden durchgeführt. Zusätzlich konnte es im April 2024 in einem freiwilligen Tutorium für Physik-Lehramtstudierende des 5. Semesters angeboten werden.

Die Erfahrungen aus diesen ersten Einsätzen zeigen, dass die Vorbereitung mit der Installation und dem Einführungsnotebook gut funktioniert. Bei der Datenauswertung arbeiten die Studierenden fokussiert, haben jedoch häufig Schwierigkeiten, das Gelernte aus der Vorbereitung auf den neuen Kontext anzuwenden und tun sich insbesondere mit neuen Inhalten schwer. Die nicht vertraute Syntax einer Programmiersprache stellt dabei sicherlich eine große, wenn nicht die größte Problemquelle dar. Erste Anpassungen durch differenzierte Hilfestellungen in der Form von Knöpfen an häufig problematischen Stellen des Notebooks können die Schwierigkeiten nur teilweise lösen, weswegen über zusätzliche Hilfestellungen nachgedacht werden sollte. Aufgrund der bisher auftretenden Schwierigkeiten übersteigt zudem aktuell der zeitliche Umfang der Datenauswertung noch die Dauer eines Praktikumstages. Eine mögliche Lösung bei vorhandenen Ressourcen wäre die Aufteilung auf zwei Praktikumstage.

Insgesamt kann jedoch festgestellt werden, dass trotz der Schwierigkeiten bei der Datenauswertung das Modul bei den Studierenden gut ankommt. Tatsächlich würden 82 % der befragten Nebenfachstudierenden das Modul, nachdem sie es durchlaufen haben, ihren Kommilitoninnen und Kommilitonen weiterempfehlen. Das Modul wird deswegen auch in Zukunft im physikalischen Nebenfachpraktikum sowie

im Kompetenzpraktikum für Lehramtstudierende an der RWTH Aachen angeboten und kontinuierlich weiterentwickelt.

## 9. Literatur

- Baumann, R. (1996). Didaktik der Informatik (2., vollst. neu bearb. Aufl). Klett.
- Classen, L. (2022). Mit Jupyter durchs Physikpraktikum: Auswerten mit Python leicht gemacht. Springer Spektrum.  
[https://doi.org/10.1007/978-3-658-37723-6\\_1](https://doi.org/10.1007/978-3-658-37723-6_1)
- Dörge, C. (2015). Informatische Schlüsselkompetenzen: Konzepte der Informationstechnologie im Sinne einer informatischen Allgemeinbildung. Universitätsverlag Potsdam.
- Gess, C., Rueß, J. & Deicke, W. (2014). Design-based Research als Ansatz zur Verbesserung der Lehre an Hochschulen - Einführung und Praxisbeispiel. Qualität in der Wissenschaft, 8(1), 10-16.
- Hubwieser, P. (2007). Didaktik der Informatik: Grundlagen, Konzepte, Beispiele (3., überarbeitete und erweiterte Auflage). Springer.
- Kattmann, U., Duit, R., Gropengießer, H. & Komorek, M. (1997). Modell der Didaktische Rekonstruktion - Ein Rahmen für naturwissenschafts-didaktische Forschung und Entwicklung. In: Zeitschrift für Didaktik der Naturwissenschaften, 3, 3-18.
- Keller, J. M. (1987). Development and use of the ARCS model of instructional design. Journal of Instructional Development, 10(3), 2-10.  
<https://doi.org/10.1007/BF02905780>
- Kultusministerkonferenz (KMK). (2017). Bildung in der digitalen Welt: Strategie der Kultusministerkonferenz. Beschluss der Kultusministerkonferenz vom 08.12.2016 in der Fassung vom 07.12.2017.
- Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen (Hrsg.) (2023). Kernlehrplan für die Sekundarstufe I Gymnasium in Nordrhein-Westfalen Wahlpflichtfach Informatik. Abgerufen am 21.05.2024, von [https://www.schulentwicklung.nrw.de/lehrpläne/up-load/klp\\_SI/GY19/wpif23/g9\\_wpif\\_klp\\_2023\\_06\\_01.pdf](https://www.schulentwicklung.nrw.de/lehrpläne/up-load/klp_SI/GY19/wpif23/g9_wpif_klp_2023_06_01.pdf)
- Wood, D., Bruner, J. S., & Ross, G. (1976). The Role of Tutoring in Problem Solving. Journal of Child Psychology and Psychiatry, 17(2), 89-100. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>
- Statista Research Department. (2024). Die beliebtesten Programmiersprachen weltweit laut PYPL-Index im Mai 2024. Statista. Abgerufen am 21.05.2024, von <https://de.statista.com/statistik/daten/studie/678732/umfrage/beliebteste-programmiersprachen-weltweit-laut-pypl-index/>

**Anhang**

GitHub-Repository des Moduls:

<https://github.com/KuehliM/Modul-Python> (Stand:  
20.05.2024)