

**Zentrum für Bibliotheks- und  
Informationswissenschaftliche Weiterbildung**  
TH Köln  
Claudiusstraße 1  
50678 Köln

**Zertifikatskurs Data Librarian – Praxisprojekt (Modul 6)**

Konzeption einer Schulung zur  
Datenverarbeitung und -visualisierung mit Python (*pandas & matplotlib*)  
anhand der Daten des *World Happiness Score*  
für Einsteiger im Rahmen des Kurses “*Discover the Data*”  
an der Universitätsbibliothek Duisburg-Essen

von

**Tim Reschop**  
Universitätsbibliothek Duisburg-Essen  
Universitätsstraße 9-11  
45141 Essen  
Tim.Reschop@uni-due.de  
Tel.: 0201/183 4524

Dozent: **Prof. Dr. Konrad Förstner**  
Campus Südstadt  
Claudiusstraße 1  
50678 Köln

## Inhaltsverzeichnis

1. Einleitung	S. 3
2. Kontext und Ziel der Veranstaltung	S. 3
3. Didaktisches Konzept	S. 4
4. Inhaltlicher Aufbau	S. 6
5. Kompetenzen	S. 6
6. Motivation und Praxisnähe	S. 7
7. Nachhaltigkeit und Integration	S. 7
8. Quellen- und Literaturverzeichnis	S. 9
8.1. Quellen und Links	S. 9
8.2. Literatur	S. 9
9. Anhang	S. 10
9.1. Verlaufsplan	S. 10
9.2. Link zum Python-Basics Tutorium	S. 11
9.3. Handout über wichtige Befehle und Funktionen in <i>pandas</i> und <i>matplotlib</i>	S. 12

## 1. Einleitung

In einer zunehmend datengetriebenen Welt gehören Datenkompetenzen zu den zentralen Schlüsselqualifikationen im Studium, in der Wissenschaft und auf dem Arbeitsmarkt. Die Fähigkeit, Daten nicht nur zu interpretieren, sondern auch selbstständig zu verarbeiten und visuell aufzubereiten, gewinnt dabei stetig an Bedeutung.<sup>1</sup>

Programmiersprachen wie Python spielen in diesem Zusammenhang eine große Rolle: Sie ermöglichen die effiziente Analyse großer und komplexer Datensätze, fördern automatisiertes Arbeiten und eröffnen vielfältige Anwendungsfelder – von der empirischen Sozialforschung bis hin zur journalistischen Datenvisualisierung.

Gerade für Studierende, die bisher primär mit Tabellenkalkulationen wie Excel gearbeitet haben, stellt Python eine nachhaltige und erweiterbare Alternative dar, die nicht nur die technische, sondern auch die analytische Handlungskompetenz stärkt.<sup>2</sup> Dafür soll die hier konzipierte Veranstaltung Grundlagen schaffen.

## 2. Kontext und Ziel der Veranstaltung

Die geplante Präsenzveranstaltung wurde im Kontext des DataCampus der Universitätsbibliothek Duisburg-Essen konzipiert und richtet sich an Studierende, die im Rahmen des Basiskurses „Discover the Data“<sup>3</sup> erste Kenntnisse im Bereich Data Literacy und darüber hinaus 3 ECTS im Ergänzungsbereich für Methodenkompetenzen<sup>4</sup> erwerben. Die Veranstaltung ist zwar zunächst als Teilmodul des Basiskurses geplant, ist aber so konzipiert worden, dass sie auch selbstständig als

---

<sup>1</sup> Vuorikari, Riina et al.: DigiComp 2.2. The Digital Competence Framework for Citizens, Luxembourg 2022, S. 14 (im Folgenden zitiert als: Vuorikari: DigiComp 2.2.)

<sup>2</sup> Vuorikari: DigiComp 2.2, S. 33-34.

<sup>3</sup> Weitere Informationen zum DataCampus und dem Basiskurs Discover the Data unter: [https://www.uni-due.de/ub/datacampus/discover\\_the\\_data.php](https://www.uni-due.de/ub/datacampus/discover_the_data.php) (zuletzt abgerufen am 25.08.2025, 13:19)

<sup>4</sup> Zum Ergänzungsbereich: <https://www.uni-due.de/iwis/methodenkompetenzen.php> (zuletzt abgerufen am 25.08.2025, 13:20)

Workshop oder auch als Online-Schulung durchführbar ist. Der Fokus dieser 120-minütigen Schulung liegt auf der Datenverarbeitung und -visualisierung mit der Programmiersprache Python, insbesondere mit den Bibliotheken *pandas*<sup>5</sup> und *matplotlib*<sup>6</sup>.

Ziel ist es, eine niederschwellige, aber praxisnahe Einführung in zentrale Elemente der Datenanalyse und -visualisierung mit Python und den ausgewählten Bibliotheken zu bieten. Die Veranstaltung soll zeigen, dass mit wenig Code bereits aussagekräftige Auswertungen und Visualisierungen möglich sind. Die Studierenden sollen dadurch dazu befähigt werden, einfache Datensätze selbstständig zu laden, auszuwerten und visuell aufzubereiten sowie eigene Fragestellungen mit vorstrukturiertem Code zu bearbeiten.

Als Datengrundlage dient der *World Happiness Report*<sup>7</sup>, ein überschaubarer, aber vielseitig nutzbarer und darüber hinaus realer, gesellschaftlich relevanter Datensatz, der motivierende Fragestellungen und explorative Zugänge ermöglicht.

### 3. Didaktisches Konzept

Da die Veranstaltung insbesondere für Lernende ohne Vorkenntnisse konzipiert ist, werden die Teilnehmenden per Copy & Modify durch die einzelnen Inhalte geleitet. Der zugrundeliegende Gedanke: Der Code wird von der Lehrperson in einem *Jupyter Notebook*<sup>8</sup> vorgeschrieben, die Teilnehmenden kopieren den Code und modifizieren ihn selbstständig im Rahmen von Mini-Challenges und umfangreicheren Aufgaben.<sup>9</sup> Diese Herangehensweise soll kognitive Überforderung vermeiden, die Nachvollziehbarkeit erhöhen und frühzeitig zu sichtbaren Lernerfolgen führen, ohne dass ständig Fehlermeldungen ausgegeben werden. Damit bedient die Veranstaltung unterschiedliche Anforderungsbereiche:

---

<sup>5</sup> <https://pandas.pydata.org/> (zuletzt abgerufen am 25.08.2025, 13:29 Uhr)

<sup>6</sup> <https://matplotlib.org/> (zuletzt abgerufen am 25.08.2025, 13:29 Uhr)

<sup>7</sup> <https://www.worldhappiness.report/ed/2024/?lctg=6050f061f98ec7553cebfa1e> (zuletzt abgerufen am: 25.08.2025, 13:13 Uhr).

<sup>8</sup> <https://jupyter.org/> (zuletzt abgerufen am 25.08.2025, 07:38 Uhr).

<sup>9</sup> Vuorikari: DigiComp 2.2, S. 29.

- Reproduktion: Das Mitschreiben und schrittweise Nachvollziehen des Codes ist eine ausschließlich reproduktive Aufgabe, um die Code-Syntax in ihren Grundzügen erst einmal kennenzulernen und nachzuvollziehen.
- Modifikation: Das Modifizieren des Codes in Form von leichten Variationen (z. B. Filtern anderer Länder, Umbenennen von Variablen, Anpassen von Plots) festigt das Verständnis vom Zusammenhang zwischen Code und Ausgabe, also welche Veränderungen im Code was genau bewirken.
- Anwendung: Schließlich sollen die Teilnehmenden kleine explorative Aufgaben selbstständig lösen und im Idealfall eigene Fragestellungen an den Datensatz auf Grundlage der vorangegangenen Ergebnisse mit *pandas* und *matplotlib* bearbeiten.

Zentral ist dabei die Orientierung an der Zielgruppe: Die Teilnehmenden des Basiskurses sind zu einem überwiegenden Teil Studienanfänger:innen ohne Programmiererfahrung, die vor allem ein erstes Verständnis für die Potenziale und Anwendungen von Python und ausgewählter Bibliotheken und nicht tiefgehende Programmierkenntnisse entwickeln sollen.

Die größte didaktische Herausforderung liegt daher in der Reduktion: Welche Inhalte sind in 120 Minuten realistisch vermittelbar, ohne Überforderung zu erzeugen und dennoch eine nachhaltige Lernerfahrung zu schaffen?

Die Schulung fokussiert sich daher auf:

- das Datenmodell von *pandas* (DataFrame und Series),
- grundlegende Operationen (Zugriff, Filterung, Sortierung),
- einfache Visualisierungen mit *matplotlib* (plot, bar, line, scatter).

Komplexere Themen wie Listenverarbeitung, Schleifen, eigene Funktionen oder Datenbereinigung werden – wenn überhaupt – nur angerissen oder ganz ausgeklammert. So bleibt Raum für aktives Ausprobieren und Wiederholen innerhalb eines begrenzten Zeitfensters.

Um den zeitlichen Rahmen der Veranstaltung nicht zu sehr auszudehnen, wird den Teilnehmenden empfohlen, vorab ein Python-Basics-Tutorial zu absolvieren, das eigens zur Vorbereitung auf die Veranstaltung ebenfalls im Rahmen dieses Praxisprojekts erstellt wurde (s. Anhang) und bereits niedrigschwellig in die Grundlagen der Programmiersprache einführt, die dann in der Veranstaltung selbst noch einmal aufgegriffen werden.

## 4. Inhaltlicher Aufbau

Die Schulung umfasst folgende thematische Schwerpunkte:

- Einführung in Python und Jupyter Notebooks (Speichern von Variablen, Laden von Bibliotheken)
- Daten einlesen mit *pandas* (`pd.read_csv`)
- Erste Dateninspektion (`.head()`, `.columns()`)
- Filtern und Sortieren nach eigenen Bedingungen
- Berechnung einfacher Kennwerte (`mean()`, `median()`, `std()`)
- Visualisierung mit *matplotlib*: Balkendiagramm, Liniendiagramm, Scatterplot

Der Verlauf ist didaktisch reduziert und auf das Wesentliche fokussiert. Komplexe Strukturen wie Schleifen, das Erstellen eigener Funktionen oder OOP werden bewusst ausgeklammert. Stattdessen stehen Wiedererkennbarkeit, visuelle Rückmeldung und interaktives Ausprobieren im Vordergrund. Eine detaillierte Darstellung des Verlaufs findet sich im Anhang.

## 5. Kompetenzen

Die Schulung vermittelt Grundlagenkompetenzen in vier zentralen Bereichen der Data Literacy<sup>10</sup>:

1. Technische Kompetenz: Nutzung von Jupyter Notebook als interaktive Lern- und Programmierumgebung.

---

<sup>10</sup> vgl. dazu Vuorikari: DigiComp 2.2, S. 14, 89 u. 109-110.

2. Datenkompetenz: Fähigkeit, einfache Datensätze zu laden, zu inspizieren, logisch zu strukturieren und statistische Grundwerte zu berechnen.
3. Visualisierungskompetenz: Umsetzung grundlegender Diagramme zur Interpretation einfacher Zusammenhänge.
4. Reflexionskompetenz: Erste Überlegungen zu Dateninterpretation, Aussagekraft und möglicher Verzerrung.

Ziel ist nicht das vollständige Erlernen von Python, sondern ein methodischer Einstieg mit direkt anwendbarem Nutzen im Studienkontext.

## **6. Motivation und Praxisnähe**

Ein zentraler Motivationsfaktor ist die Verwendung realer, gesellschaftsrelevanter Daten: Der World Happiness Report regt zur kritischen Reflexion an und lässt sich mit persönlichen Fragestellungen verknüpfen („Wie glücklich ist mein Land?“, „Macht Geld wirklich glücklich?“).

Durch den sichtbaren Output – Diagramme, Vergleiche, Rankings – erhalten die Lernenden unmittelbares Feedback auf ihre Eingaben. Offene Aufgaben, z. B. die freie Auswahl von Ländern oder die Gestaltung eines eigenen Plots, stärken die Identifikation mit dem Lerngegenstand und erzeugen Varianz im Lernprozess.

## **7. Nachhaltigkeit & Integration**

Am Ende der Veranstaltung erhalten die Teilnehmenden eine kommentierte Jupyter Notebook-Datei, die alle bearbeiteten Schritte dokumentiert und als Vorlage bzw. „Nachschlagewerk“ für eigene Analysen dienen kann. Dieses Arbeitsdokument kann auch für begleitende Studienleistungen (z. B. Projektarbeiten, Hausarbeiten) genutzt werden. Außerdem soll den Studierenden ein Handout mit häufig verwendeten Funktionen bzw. Code, der im Rahmen der Veranstaltung benutzt wurde, ausgehändigt werden. Da *pandas* und *matplotlib* etablierte Bibliotheken sind, ist das Erlernte gut übertragbar auf spätere Anforderungen in Studium oder Beruf.

Die Schulung ist modular aufgebaut und lässt sich in künftige Kursformate des DataCampus integrieren, ist also auch außerhalb des Basiskurses „Discover the Data“ durchführbar – sei es als Online-Veranstaltung, Tutorium oder als eigenständiger Workshop.



## **8. Quellen- und Literaturverzeichnis**

### **8.1. Quellen und Links**

[https://www.uni-due.de/ub/datacampus/discover\\_the\\_data.php](https://www.uni-due.de/ub/datacampus/discover_the_data.php) (zuletzt abgerufen am 25.08.2025, 13:19).

<https://www.uni-due.de/iwis/methodenkompetenzen.php> (zuletzt abgerufen am 25.08.2025, 13:20).

<https://www.worldhappiness.report/ed/2024/?lctg=6050f061f98ec7553cebfale> (zuletzt abgerufen am: 25.08.2025, 13:13 Uhr).

<https://jupyter.org/> (zuletzt abgerufen am 25.08.2025, 07:38 Uhr).

### **8.2. Literatur**

Vuorikari, Riina et al.: DigiComp 2.2. The Digital Competence Framework for Citizens, Luxemburg 2022.

## 9. Anhang

### 9.1. Verlaufsplan

Zeit	Inhalt	Kompetenzen	Wichtiger Code
10 Minuten	Datenverarbeitung mit pandas (1)	<ul style="list-style-type: none"> <li>Importieren von Bibliotheken</li> <li>Einlesen von Datensätzen als DataFrame</li> <li>Handhabung von DataFrames</li> </ul>	<pre>import .. as .. pd.read  df.columns df.sort_values df[['column1', 'column2' ...]]</pre>
10 Minuten	Mini-Challenges	Festigen der Inhalte aus dem vorigen Abschnitt anhand vom selbstständigen bearbeiten von Aufgaben.	s.o.
10 Minuten	Datenverarbeitung mit pandas (2)	<ul style="list-style-type: none"> <li>Erstellung eines neuen, gefilterten DFs aus dem alten DF</li> <li>Umbenennen von Spalten</li> <li>Löschen von Spalten</li> </ul>	<pre>df_2 = df[df['Year'] == 2024]  df_2.rename(columns={...}, inplace=True) df_2.drop(columns=["..."], inplace=True)</pre>
5 Minuten	Mini-Challenges	Festigen der Inhalte aus dem vorigen Abschnitt anhand vom selbstständigen bearbeiten von Aufgaben.	s.o.
15 Minuten	Datenverarbeitung mit pandas (3)	<ul style="list-style-type: none"> <li>Berechnen statistischer Grundwerte (Mittelwert, Median, Stabw)</li> <li>Erstellen eines DF aus diesen errechneten Werten</li> <li>Berechnen des Mittelwertes mehrerer Jahre und anordnen in einem DF</li> </ul>	<pre>df_2['column1'].mean() df_2['column1'].median() df_2['column1'].std() pd.DataFrame({...})  df_2.groupby("Year")["Ladder score"].mean().reset_index()</pre>
10 Minuten	Mini-Challenges	Festigen der Inhalte aus dem vorigen Abschnitt anhand vom selbstständigen bearbeiten von Aufgaben.	s.o.
30 Minuten	Visualisierung mit matplotlib	<ul style="list-style-type: none"> <li>Erstellen verschiedener Diagramme (Säulendiagramm, Scatterplot, Liniendiagramm) aus ausgewählten Werten (subsets) anhand von Fragestellung</li> </ul>	Verschiedene plt.-Funktionen (plt.figure(figsize=(...)), plt.bar(), plt.title(), plt.xlabel(), plt.ylabel(), plt.show usw.)

30 Minuten	Praxisphase – Eigenständiges erstellen von Diagrammen mit matplotlib anhand von Fragestellungen an den WHR-Datensatz	Grundsätzlich werden hier noch einmal die Schritte aus der vorangegangenen Phase selbstständig durch die Teilnehmer*innen mit anderen Werten aus dem DF ausgeführt.	s.o.
------------	--	---	------

## 9.2. Link zum Python-Basics Tutorium

[https://timreschop.github.io/Modul\\_6-Data\\_Librarian\\_Praxisprojekt/](https://timreschop.github.io/Modul_6-Data_Librarian_Praxisprojekt/)

### 9.3. Handout über wichtige Befehle und Funktionen in *pandas* und *matplotlib*

Code	Erläuterung
<b>import ... as ...</b>	Importieren von Bibliotheken
<b>pd.read_csv</b> („/home/user/.../Datei.csv“) <b>pd.read_csv</b> (“https://url.datei”)	Einlesen einer csv-Datei oder anderer Dateien (excel, txt) in einen <i>pandas</i> -DataFrame aus einem Verzeichnis oder einer URL
<b>Handhabung DataFrame</b>	
<b>df.columns</b>	Gibt Namen der Spalten des DataFrames aus
<b>df.[[„Spaltenname1“, „Spaltenname2“]]</b>	Zeigt DF nur mit ausgewählten Spalten an
<b>df.sort_values</b> (by=“Spaltenname1“, „Spaltenname2“, ascending=[True, False]) [[„Spaltenname1“, „Spaltenname2“]]	Sortiert den DataFrame nach den angegebenen Spaltennamen (zuerst 1, dann 2) Spaltenname1 alphabetisch aufsteigend (True), Spaltenname2 absteigend (False) Gibt nur diese Spalten des DF aus
<b>df_2 = df[df[„Spaltenname3“] == 1]</b>	Filtert alle Zeilen aus dem DataFrame <i>df</i> , bei denen in der Spalte „Spaltenname3“ der Wert 1 steht, und speichert das Ergebnis in einen neuen DataFrame <i>df2</i> .
<b>df2.rename</b> (columns={„Spaltenname1“ : “Jahr“, inplace=True)	Der Code benennt die Spalte „Spaltenname1“ in „Jahr“ um und übernimmt die Änderung direkt ( <i>inplace=True</i> ) im DataFrame <i>df2</i>
<b>df.drop</b> (columns=[“Spaltenname2“], inplace=True)	Der Code entfernt die Spalte „Spaltenname2“ aus dem Dataframe und übernimmt die Änderung direkt ( <i>inplace=true</i> ) im <i>df2</i>
<b>Statistische Berechnungen</b>	
<b>mw = df2[“Spalte1“].mean()</b>	Berechnet den Mittelwert (.mean()) der Werte aus „Spalte1“ des <i>df2</i> und speichert ihn in der Variable „mw“ ab
<b>med = df2[„Spalte1“].median()</b>	Berechnet den Median (.median()) der Werte aus „Spalte1“ des <i>df2</i> und speichert ihn in der Variable „med“ ab
<b>stabw = df2[„Spalte1“].std()</b>	Berechnet die Standardabweichung (.std()) der Werte aus „Spalte1“ des <i>df2</i> und speichert ihn in der Variable „stabw“ ab

Anleitung zur Erstellung eines Diagramms	
<code>plt.figure(figsize=(10, 6))</code>	Erstellen einer Abbildung mit einer bestimmten Größe (Breite, Höhe in Zoll)
<code>plt.bar(df2[„Spalte1“], df2[„Spalte2“], color=“skyblue“)</code>	Die Abbildung soll ein Balkendiagramm (barplot) werden (x = Spalte 1, y = Spalte 2), Die Farbe wird auf hellblau (skyblue) gesetzt – mit <code>plt.scatter</code> oder <code>plt.plot</code> können auch andere Diagrammtypen erstellt werden.
<code>plt.title(“Balkendiagramm“)</code>	Der Titel des Diagramms wird gesetzt
<code>plt.xlabel(“X-Achse“)</code> <code>plt.ylabel(„Y-Achse“)</code>	Beschriftung der x- und y-Achse
<code>plt.ylim(2, 10) #optional</code>	Begrenzt die Werte auf der y-Achse von 2 - 10
<code>plt.xticks(rotation=45) #optional</code>	Dreht die Bezeichnungen auf der x-Achse um 45 Grad
<code>plt.grid(True) #optional</code>	Fügt ein Raster im Hintergrund des Diagramms ein
<code>plt.tight_layout() #optional</code>	Sorgt dafür, dass sich nichts überlappt (z.B. Achsenbeschriftungen)
<code>plt.show()</code>	Zeigt das fertige Diagramm an

## **Eigenständigkeitserklärung**

Hiermit bestätige ich, Tim Reschop, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Witten, 27.08.2025

(Ort, Datum)



---

(Unterschrift)