
Einführung in Python
für Geisteswissenschaftler:innen
Melanie Althage (melanie.althage@hu-berlin.de)
4.-8. März 2024,
Campus Essen, Universität Duisburg-Essen
Tag 5

Lernziele des heutigen Workshop-Tages

(Fortsetzung) Erweiterte Python-Kenntnisse:

- Verständnis der Grundkonzepte der Data-Mining-Bibliothek Pandas
- Fähigkeit, einfache Datenmanipulationen und Explorationen mit Pandas umzusetzen
- Fähigkeit, einfache Visualisierungen zur Kommunikation und Veranschaulichung von Daten zu erstellen

Ggf. Verständnis dafür, was reguläre Ausdrücke sind und wofür sie verwendet werden

- Überblick über die wichtigsten Metazeichen und Ihre Bedeutung
- Überblick über zentrale Funktionen zur Arbeit mit Reg. Ausdrücken

Wichtiger Hinweis!

Binder Usage Guidelines

Laden Sie regelmäßig die von Ihnen bearbeiteten Notebooks (und Dateien) aus der Binder-Umgebung herunter! Diese werden nicht dauerhaft gespeichert.

Wenn Ihre Binderinstanz längere Zeit inaktiv war (mehr als **10 Minuten**!), dann wird ihre Session terminiert, alle nicht gesicherten Daten und Notebooks sind dann verloren.

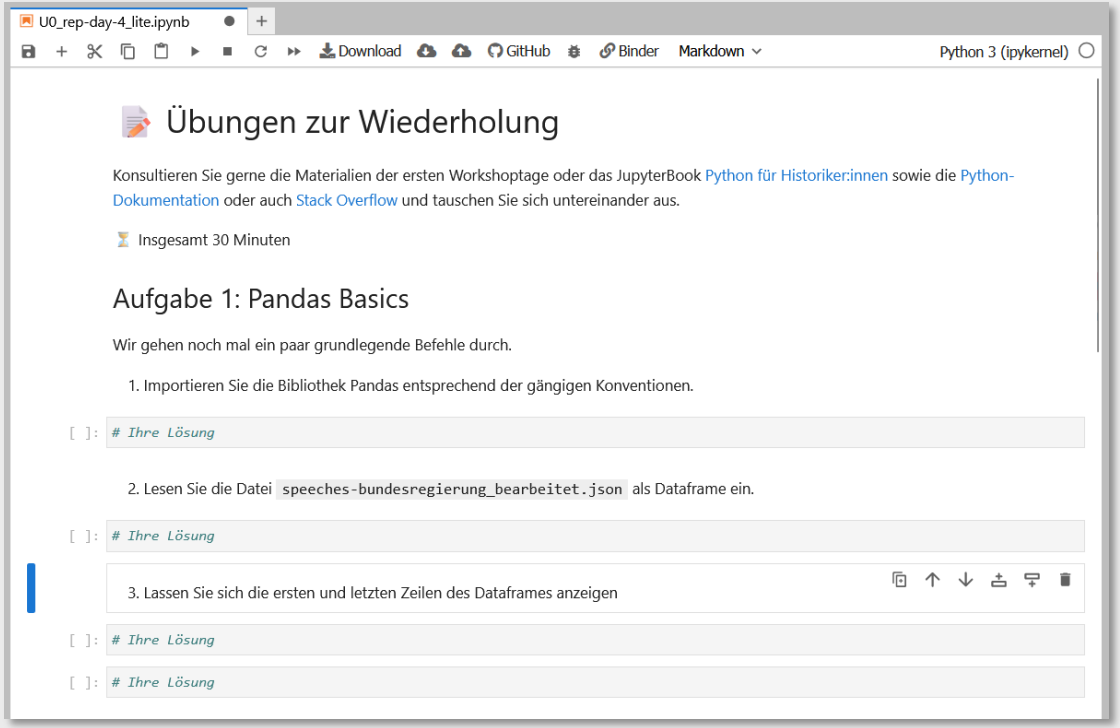
Heruntergeladene Notebooks können hochgeladen und weiter bearbeitet werden.

Zeitplan

9:00	Übungen zur Wiederholung der Inhalte von Tag 4
10:00	Datenvisualisierung I
10:45	PAUSE
11:00	Datenvisualisierung II
12:00	MITTAGSPAUSE
13:00	Ggf. Einstieg: Reguläre Ausdrücke
14:15	PAUSE
14:30	Ggf. Einstieg: Reguläre Ausdrücke
15:30	Abschlussrunde

1

Tag 4 revisited



Notebook: tag-5/U0_rep-day-4_lite.ipynb

Zeit: 30 Minuten

2

Reguläre Ausdrücke (aka. RegEx)

- Mit regulären Ausdrücken können komplexe Muster in Zeichenketten identifiziert und dadurch gezieltere Operationen als mit den nativen String-Funktionen ausgeführt werden.
  - Reguläre Ausdrücke sind Sequenzen von Zeichen, die ein Suchmuster definieren.
  - Anwendung bspw. Suchen und Ersetzen
- Reguläre Ausdrücke sind programmiersprachenunabhängig!
- Beispiel: So könnte ein regulärer Ausdruck für E-Mail-Adressen aussehen:

re.findall(r“\b[\w.-]+@[\w.-]+\.\w+\b“, text)

Hauptfunktionen

- re.search(): Sucht nach einem Muster in einem String
- re.match(): Überprüft, ob der Anfang eines Strings auf das Muster passt
- re.findall(): Findet alle Vorkommen eines Musters im String
- re.sub(): Ersetzt die Teile des Strings, die auf das Muster passen
- re.compile(): Kompiliert ein Regex-Muster in ein Regex-Objekt, das für die Suche verwendet werden kann

Häufig verwendete Metazeichen

- **.** (Punkt): Steht für jedes Zeichen außer einem neuen Zeilenzeichen
- **^**: Beginn eines Strings
- **$**: Ende eines Strings
- *****: Null oder mehr Vorkommen des vorangegangenen Zeichens
- **+**: Ein oder mehr Vorkommen des vorangegangenen Zeichens
- **?**: Null oder ein Vorkommen des vorangegangenen Zeichens
- **\d**: Ein Ziffernzeichen, äquivalent zu [0-9]
- **\w**: Ein Wortzeichen (Buchstabe, Ziffer oder Unterstrich)
- **\s**: Ein Leerzeichen (inklusive Tab und neue Zeile)

Weitere nützliche Funktionalitäten

Gruppierung und Alternativen:

- Runde Klammern **()**: Werden verwendet, um Gruppen zu erstellen
- Pipe **|**: Steht für "oder" und erlaubt die Alternation zwischen Mustern

Flags:

- **re.IGNORECASE (re.I)**: ignoriert Groß- und Kleinschreibung
- **re.MULTILINE (re.M)**: Lässt **^** und **$** den Anfang bzw. das Ende jeder Zeile (nach jedem neuen Zeilenzeichen) sowie den Anfang bzw. das Ende des gesamten Strings matchen
- **re.DOTALL (re.S)**: Lässt den Punkt (.) auch neue Zeilenzeichen matchen

3

Weiterführende Bibliotheken

Datenvisualisierung

- Neben Matplotlib und Seaborn:
  - Bokeh (interaktive Visualisierungen)
  - Plotly (insbesondere für webbasierte und interaktive Visualisierungen, auch Karten und 3D-Diagramme)
  - WordCloud (Wortwolken)

Arbeit mit XML-Daten:

  - ElementTree
  - lxml
  - BeautifulSoup (auch für HTML und damit zum Web Scraping geeignet)

Weiterführende Bibliotheken

Natural Language Processing und Text Mining

- **NLTK (Natural Language Toolkit)** – eine der ältesten Bibliotheken
  - Klassifizierung, Tokenisierung, Stemming, Tagging, Parsing, …
- **spaCy**
  - Moderne Sprachmodelle für Tasks wie in NLTK
  - Zusätzlich: NER, spezifisches Training von Sprachmodellen etc.
- **Gensim**
  - Klassifikation von Textkorpora u.a. mit Topic Modeling
- **Scikit-learn**
  - Nicht speziell für NLP, aber viele Algorithmen für Textvektorisierung, Clustering und Klassifikation
- **weitere Bibliotheken:** TextBlob (u.a. Sentimentanalyse), Tomotopy und BERTopic (Topic Modeling), flair (u.a. Named Entity Recognition), …

Weiterführende Bibliotheken

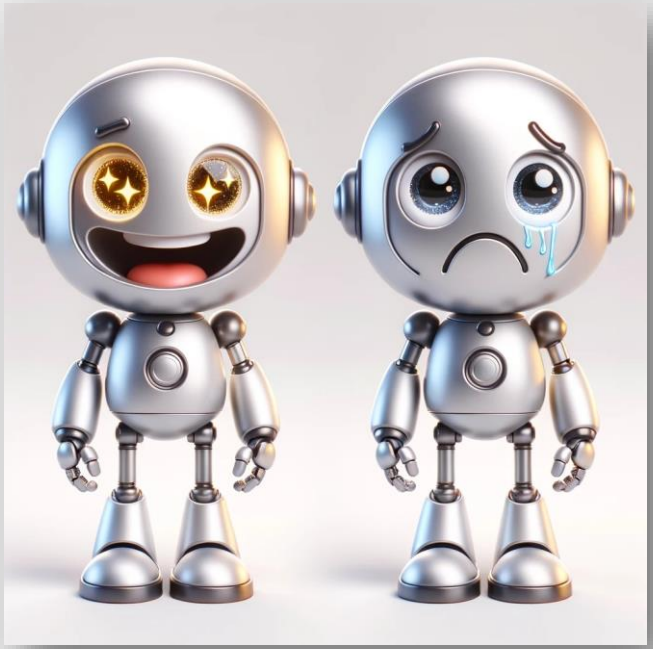
Geoinformationssysteme (GIS)

- Shapely – zur Arbeit mit planaren, geometrischen Objekten
- GeoPandas – Kombination von Pandas und Shapely, ersetzt PostGIS-Datenbanken in Python
- Pyproj – Python-Implementierung für Proj
- Cartopy – Python-Bibliothek zur Verarbeitung von Geodaten zur Erstellung von Karten und räumlichen Analysen

3

Abschlussrunde

- Was haben Sie vom Workshop mitgenommen?
- Worüber denken Sie jetzt anders als vor dem Workshop?
- Was hätte Ihr Lernerlebnis noch optimieren können?



Herzlichen Dank für den schönen Workshop!