



Practical Data Analysis and Visualisation

David Lassig

14.Juni 2019



Inhaltsübersicht

- 1** Idee
- 2 Was ist Jupyter?
- 3 Was ist Pandas?
- 4 UseCase 01: ipywidgets für eigene Analyse-GUI
- 5 UseCase 02: Bokeh für geographische Visualisierung
- 6 UseCase 03: Neo4J für graphische Datenbank



Ausgangssituation

- Durchführung BFD-Maßnahme DataAnalyst von Udacity
 - Datenanalyse, Datenvisualisierung und Einstieg DataScience mit Pandas
- Bedarf sofort erkannt



Konzept

- Nutzung Pandas in Jupyter um:
 - Daten säubern (Data Wrangling)
 - Daten sichten (Data Exploration)
 - Daten auswerten (Data Analysis)
 - Daten präsentieren (Explanatory Presentation)
 - Automatisierung von Abläufen
- zusätzlich Nutzung Bokeh um:
 - geographische Datensätze zu visualisieren
- zusätzlich Nutzung Neo4J:
 - um Verknüpfung von Datensätzen zu visualisieren



Zielsetzung für mich





Zielsetzung für Zuhörer

- Pandas und Jupyter sind unverzichtbar zur Datenanalyse (Open Source)
- Web Reconnaissance ist automatisierbar
- Software ist leicht zu ändern, **Daten nicht**
- PoC or GTFO
- Einfach mal machen



Inhaltsübersicht

- 1 Idee
- 2 Was ist Jupyter?**
- 3 Was ist Pandas?
- 4 UseCase 01: ipywidgets für eigene Analyse-GUI
- 5 UseCase 02: Bokeh für geographische Visualisierung
- 6 UseCase 03: Neo4J für graphische Datenbank



Eckdaten & Entwicklung

- **Project Jupyter** wurde 2014 als “Spinoff” von ipython gegründet
 - Jupyter unterstützt Execution Environments (**Kernels**) für mehrere Sprachen (Julia, R, Python, ...)
 - Philosophie: interaktive DataScience und wissenschaftliches Programmieren
- Ausprägungen:
 - Jupyter Notebook
 - Jupyter Lab
 - Jupyter Hub (Multi-User Server)





Jupyter

Jupyter101 Notebook

http://127.0.0.1:8888/myjupyter3r/notebooks/work/01_jupyter101.ipynb



Inhaltsübersicht

- 1 Idee
- 2 Was ist Jupyter?
- 3 Was ist Pandas?**
- 4 UseCase 01: ipywidgets für eigene Analyse-GUI
- 5 UseCase 02: Bokeh für geographische Visualisierung
- 6 UseCase 03: Neo4J für graphische Datenbank



Eckdaten & Entwicklung

- Python Library für Datenanalyse und Datenmanipulation
 - **DataFrame**-Objekt mit automatischer Indizierung ~ Liste von Tuple
 - erlaubt einfachen Import von CSV, Excel, JSON, SQL uvm.
- Entwickler WesMcKinney begann 2008 mit der Entwicklung bei einer Kapitalgesellschaft



Pandas101

Pandas101 Notebook

http://127.0.0.1:8888/myjupyter3r/notebooks/work/02_pandas101.ipynb



Inhaltsübersicht

- 1 Idee
- 2 Was ist Jupyter?
- 3 Was ist Pandas?
- 4 UseCase 01: ipywidgets für eigene Analyse-GUI**
- 5 UseCase 02: Bokeh für geographische Visualisierung
- 6 UseCase 03: Neo4J für graphische Datenbank



Was ist IPyWidgets?

- vorgefertigte Steuerungs- und Kontrollwidgets
- erlauben schnelle Erzeugung von eigenen Interaktions-GUIs
- wurde zu Beginn exklusiv für Jupyter Notebook entwickelt



IPyWidgets

IPyWidgets101 Notebook

[http://127.0.0.1:
8888/myjupyter3r/notebooks/work/03_ipywidgets101.ipynb](http://127.0.0.1:8888/myjupyter3r/notebooks/work/03_ipywidgets101.ipynb)

IPyWidgets102 Notebook

[http://127.0.0.1:
8888/myjupyter3r/notebooks/work/04_ipywidgets102.ipynb](http://127.0.0.1:8888/myjupyter3r/notebooks/work/04_ipywidgets102.ipynb)



Inhaltsübersicht

- 1 Idee
- 2 Was ist Jupyter?
- 3 Was ist Pandas?
- 4 UseCase 01: ipywidgets für eigene Analyse-GUI
- 5 UseCase 02: Bokeh für geographische Visualisierung**
- 6 UseCase 03: Neo4J für graphische Datenbank



Was ist Bokeh?

- interaktive Python Visualisierung-Library mit modernen Web Browsern als Ziel-Medium
- Interaktive Daten-Plots, Dashboards und geospatiale Visualisierungen
- gibt zahlreiche andere Libraries mit ähnlicher Zielsetzung, z.B. Plotly



Bokeh

Bokeh101 Notebook

http://127.0.0.1:8888/myjupyter3r/notebooks/work/05_bokeh101.ipynb

Bokeh102 Notebook

http://127.0.0.1:8888/myjupyter3r/notebooks/work/06_bokeh102.ipynb



Inhaltsübersicht

- 1 Idee
- 2 Was ist Jupyter?
- 3 Was ist Pandas?
- 4 UseCase 01: ipywidgets für eigene Analyse-GUI
- 5 UseCase 02: Bokeh für geographische Visualisierung
- 6 UseCase 03: Neo4J für graphische Datenbank**



Was ist Neo4J?

- javabasierte Open-Source-Graphdatenbank
- Daten sind in Graphen anstatt Tabellen gespeichert
 - Kanten, Knoten und Attribute
- einfacher Export und Standalone Graphen mit vis.js
- Python bietet mit Py2Neo ausgereifte API



Cypher

- Neo4J benutzt Cypher zur Datenbankabfrage
 - deklarative “Graph Query Language”
 - CRUD für Graphen (Create, Read, Update, Delete)
 - erlaubt Remoteaccess auf Neo4J Datenbank
- wurde als OpenCypher integriert von:
 - SAP HANA
 - Redis
 - Apache Spark





Py2Neo

Py2Neo101 Notebook

http://127.0.0.1:8888/myjupyter3r/notebooks/work/07_neo4j101.ipynb

Py2Neo102 Notebook

http://127.0.0.1:8888/myjupyter3r/notebooks/work/08_neo4j102.ipynb



ENDE

