Tag 7
Inhaltsverzeichnis

- Referenzen
- Was ist ein Jupyter Notebook?
- Mein erstes Notebook Beispiel
- Daten mit dem Notebook lesen und anzeigen
- Beispiel GroupAllocator
- Übungen
- BYOQ

Referenzen
Viele davon auf Englisch

- Offizielle Dokumentation:
  https://jupyter-notebook.readthedocs.io/en/stable/

- Anaconda Download:
  https://www.anaconda.com/download
    - Installation auf Windows: hier
    - Installation auf Linux: hier

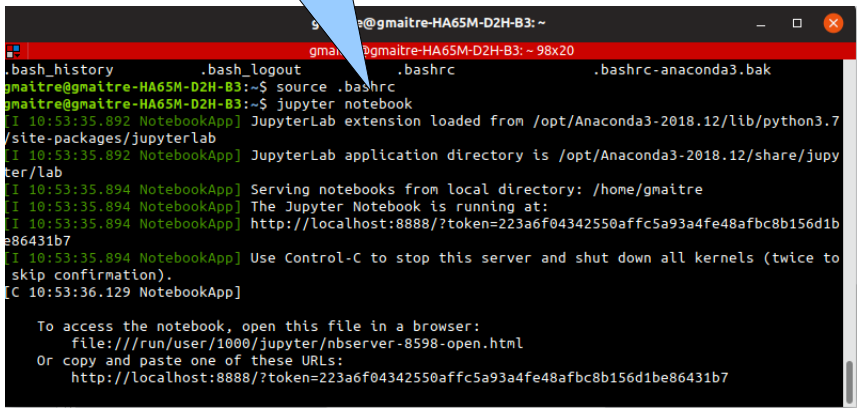
- Eine gute Einführung:
  https://www.dataquest.io/blog/jupyter-notebook-tutorial/

- Daten mit Plotly visualisieren:
  https://plotly.com/python/

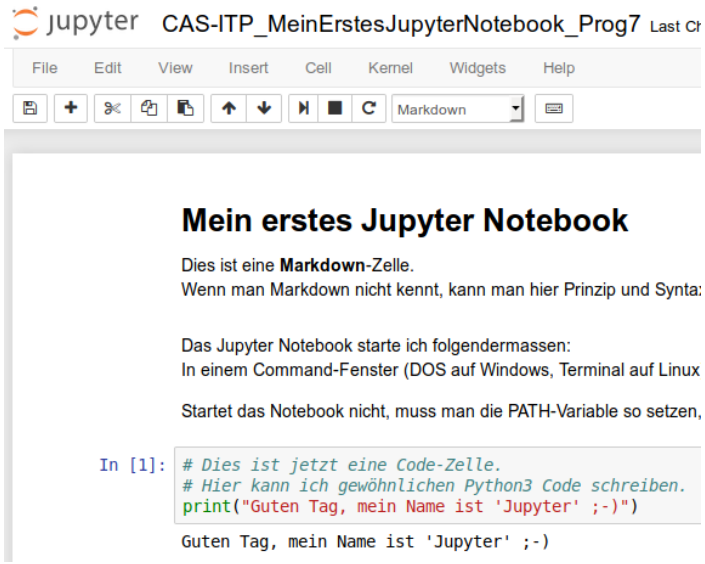
- Markdown
  https://de.wikipedia.org/wiki/Markdown

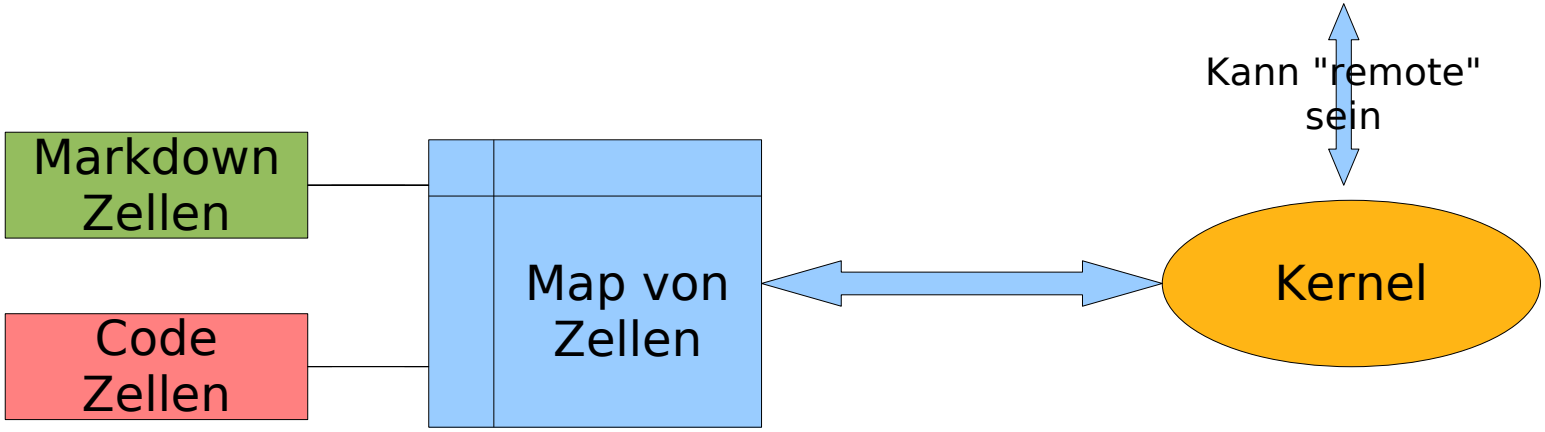
Was ist ein Jupyter Notebook?
Prinzip

Im DOS-Fenster: "jupyter notebook" eingeben



Interaktive Ausführung in einem Browser





Was ist ein Jupyter Notebook?
Struktur der Verzeichnisse

Wir brauchen ein Verzeichnis für die Notebooks und eins für die Daten.

1) Die Zip-Datei mit dem Name *CAS-ITP_DMOD_NotebooksAndData* steht auf Moodle zur Verfügung.

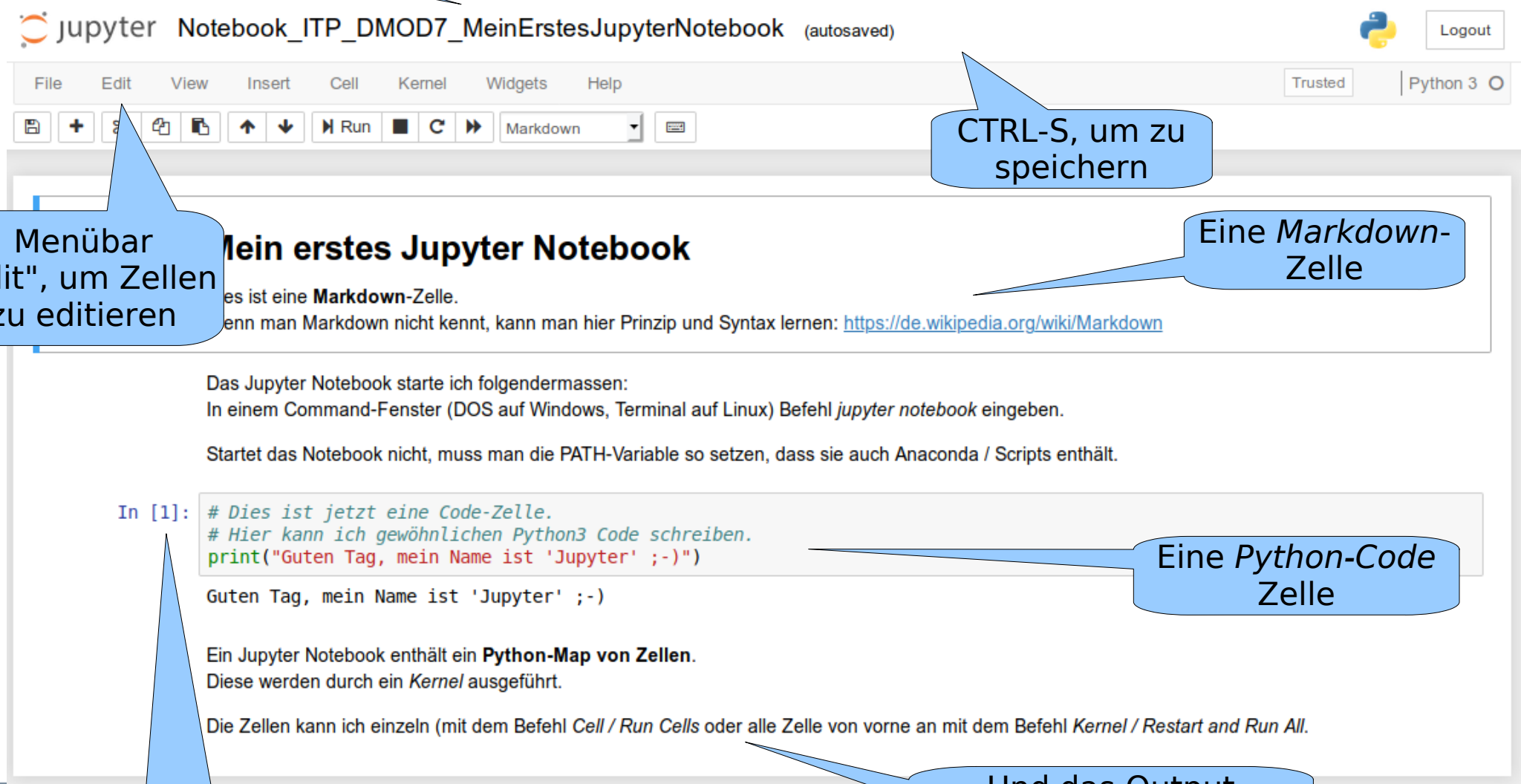
2) Diese Zip-Datei herunterladen, entpacken und die beiden Verzeichnisse ins Ihr eigenes CAS-ITP Verzeichnis kopieren.

3) Jetzt sollen Sie im Ihrem CAS-ITP Verzeichnis folgende Unterverzeichnisse haben:
    1) Notebooks
    2) Data

4) Geben Sie auf einem DOS-Fenster den Befehl "jupyter notebook" ein.

5) Ihr Interner-Browser sollte jetzt Jupyter Ihre Verzeichnisliste anzeigen.

Was ist ein Jupyter Notebook?
Mein erstes Beispiel



Was ist ein Jupyter Notebook?
Vergleich Skript vs Notebook-Entwicklung

Skript/Programm

- **Lineare** Entwicklung "Edit/(Compile)/Debug"-Zyklus
- Debugger/Tracing gehört dazu
- Ideal für "**produktive**" Entwicklung
- Nicht ideal für "**prospektive**" Entwicklung

Notebook

- **Interaktive** / **Inkrementelle** Entwicklung
- Kein Debugger => Arbeit mit den Zellen
- Nicht ideal für "**produktive**" Entwicklung
- Ideal für "**prospektive**" Entwicklung

Anwendungsbeispiel bei der SBB-Infrastruktur:
1) Data-Scientists/Bahn-Ingenieure entwickeln Notebooks-Apps
2) Software-Ingenieure *integrieren/weiterentwickeln* diese für TEST/PROD-Systeme

Was ist ein Jupyter Notebook?
Spickzettel

- Hier ist der Spickzettel
  - Die wichtigsten Befehle / Tastaturkürzel lauten:
    - <CTRL>-<ENTER> aktuelle Zelle ausführen
    - <ESC> "Edit" / "Command"-Modus wechseln
    - a / b Code-Zelle "above" / "below" addieren
    - d d aktuelle Zelle löschen (in Command-Modus)
    - z Gelöschte Zelle wieder beleben (in Command-Modus)
    - <CTRL>-s Checkpoint speichern
    - <TAB> "Code completion", je nach Kontext
    - ? Help anzeigen (Funktion oder Objekt)

- Wichtigste Pandas *DataFrame*-Befehle
    - df # Am Schluss einer Zelle: DataFrame anzeigen
    - df.shape # DataFrame-Dimensionen (Zeilen x Spalten)
    - df.head() # Erste 5 Zeilen anzeigen
    - df.info() # DateFrame-Datentypen anzeigen
    - df.columns # DataFrame-Spalten (ohne "()") anzeigen
    - df.index # DataFrame-Index (ohne "()") anzeigen
    - df.describe() # Einfache Statistik durchführen

Was ist ein Jupyter Notebook?
jupyter notebook vs. jupyter lab

jupyter notebook

- "Altes" Werkzeug für die Arbeit mit Notebooks
- Sehr gut bekannt und dokumentiert
- Ende der Entwicklung absehbar
- Rudimentär, weit weg einer Entwicklungsumgebung, wie PyCharm
- *Einfacheres Support von Plotly*
- Start mit "jupyter notebook"

jupyter lab

- Künftiges Werkzeug für die Arbeit mit Jupyter Notebooks
- 100% kompatibel mit den Jupyter Notebooks
- Weniger bekannt und dokumentiert
- In der Entwicklung, Überraschungen nicht ausgeschlossen ;-)
- Weniger weit weg einer Entwicklungsumgebung, wie PyCharm
- Integrierter CSV-*Reader*
- Start mit "jupyter lab"

Analyse Bio-LWBetriebe 1990-2019
Suchen nach Daten

- Ein erster Versuch: OpenData und dann Wiki Quickstart
  - Hier sieht man, welche Daten verfügbar sind.
  - Und spezifisch für die Schweiz. Gut zu wissen!

- Ein anderer Weg: Das Bundesamt für Statistik
  - Datenüberblick hier
  - Ich suche nach "landwirtschaft betriebe"
  - Unter Landwirtschafliche Betriebe und Beschäftigte nach Kanton finden wir etwas
  - Die Daten kann man hier runterladen, aber man muss sagen, welche…
  - Folgende Felder wähle ich:
    - Beschäftigte Total, alle Kantone (ohne Schweiz), "Bio-Betriebe" als Betriebssystem, für die Jahre 1990, 2000, 2010, 2019
    - Dateiformat: *Textdatei (Strichpunkt getrennt) mit Kopfzeile*
  - Datei auf anzeigen… Zeichenkodierung?
  - Die Datei mit Excel öffnen… Kodierung: "ISO-8859-1" wählen

Analyse Bio-LWBetriebe 1990-2019
Analyse im Notebook / Überblick

Folgende Operationen führen wir jetzt im Notebook aus:

1) Daten in ein Pandas *Data-Frame* laden und anzeigen

2) Ein paar Spalten löschen

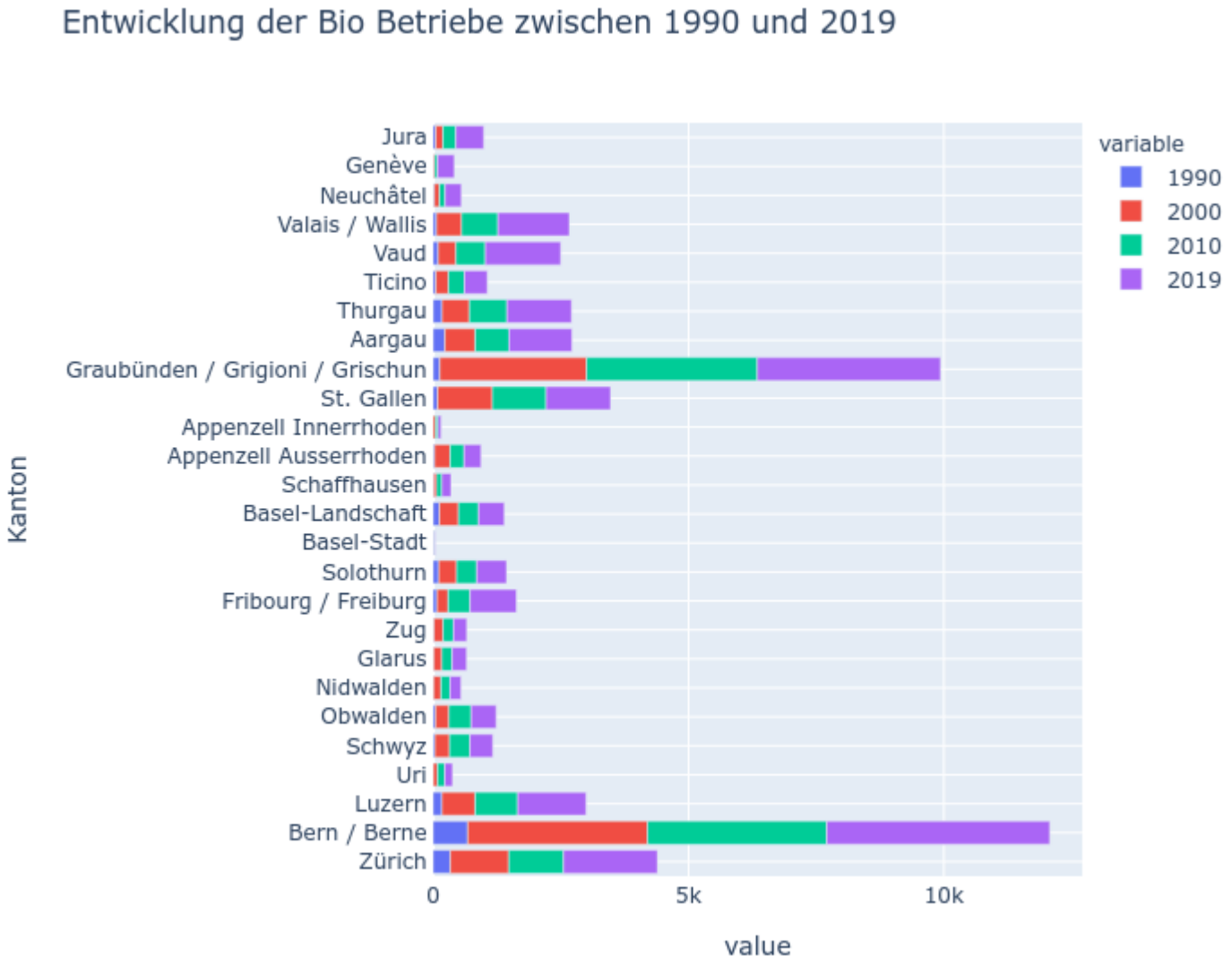
3) Einfache Statistik ausführen, mit *describe()*

4) Die Daten grafisch mit Plotly darstellen

5) Entwicklung zwischen 1990 und 2019 in % anzeigen

Jetzt ins Notebook umsteigen...

Analyse Bio-LWBetriebe 1990-2019
Daten mit Plotly anzeigen (1)



Berner Fachhochschule
Gilles Maitre

Analyse Bio-LWBetriebe 1990-2019
Daten mit Plotly anzeigen (2)

Plotly ist die *grosse Revolution*, um Daten zu visualisieren:

1) Arbeitet Hand in Hand mit Pandas *Data-Frames.*

2) Haufen von "Traces". Siehe hier.

3) Einfache Scatter-, Bar- und Line-Plots.

4) **Diese Diagramme sind interaktiv… und farbig**
   Tolle Beispiele hier. (jemand farbenblind?)

5) Die Diagramme kann man als PNG (statisch) oder HTML (mit Animation) speichern.

6) Leichte Installation für Jupyter Notebook. Schwierigere für Lab. Siehe Installationsanleitung auf der Kurs-Homepage.

Analyse Bio-LWBetriebe 1990-2019
Analyse im Notebook, Entwicklung in %

Jetzt noch die Entwicklung zwischen 1990 und 2019 in %

Dafür addieren wir eine neue Spalte *diff1990_2017*, die diese %-Zahlen enthält. Und sortieren das DataFrame nach dieser Spalte, aufsteigend.

Der Kanton Appenzell Innerrhoden hat eine unendliche Entwicklung hinter sich ;-)

```
biobetriebe_df['diff1990_2019'] = biobetriebe_df['2019']*100/biobetriebe_df['1990'] - 100
biobetriebe_df.sort_values(by = 'diff1990_2019')
```

	Kanton	2019	2010	2000	1990	diff1990_2019
11	Basel-Stadt	14	11	8	7	100.000000
12	Basel-Landschaft	507	391	372	125	305.600000
10	Solothurn	585	394	348	111	427.027027
18	Aargau	1232	664	591	232	431.034483
0	Zürich	1835	1073	1144	338	442.899408
1	Bern / Berne	4366	3504	3524	677	544.903988
19	Thurgau	1263	736	534	176	617.613636
2	Luzern	1339	832	649	174	669.540230
20	Ticino	447	309	249	55	712.727273
25	Jura	545	254	138	54	909.259259
5	Obwalden	490	433	265	47	942.553191

Analyse Bio-LWBetriebe 1990-2019
Fazit

1) Wir haben mit wenig Python-Code eine einfache Analyse der Daten durchgeführt.

2) Mit Excel oder LibreOffice Calc hätten wir etwas Ähnliches auch geschafft.

3) Sobald die Analyse komplexer wird, ist IMHO die Lösung mit Notebook wesentlich stärker.

4) Ihre Meinung?

Notebooks vergleichen/teilen
Wie geht das?

1) Jupyter Notebooks sind dafür nicht gut geeignet.

2) Pragmatische Lösung dazu:
    1) Das Notebook als Python-Skript (Menü "File") herunterladen.

    2) Mit einem Dateivergleichsprogramm (z.B. kdiff3) die Python-Skripts vergleichen.
3) Um Notebooks zu verteilen, geht es *am einfachsten*, die .ipynb-Dateien herum zu schicken.

4) Das werden Sie für die Projektarbeit brauchen.

Notebook Beispiel GroupAllocator

[3]:
```
studies = {'Vorname':['Silvy', 'StefanA', 'Dominic', 'Cora', 'Sabrina', 'Ursina', 'Astrid', 'Davide', 'André', 'ChristianC
                      'Cornelia', 'Fränzie', 'Patrick', 'ChristianL', 'Andreas', 'Alex', 'ThomasM', 'Jean-Louis', 'Miriam', '
data_df = pd.DataFrame(studies)
data_df.head()
```

[3]:

	Vorname
0	Silvy
1	StefanA
2	Dominic
3	Cora
4	Sabrina

Zufallzahlen + Gruppen erstellen und übernehmen

[4]:
```
# generate random integer values and groups
from random import seed
from random import randint
rnd_list = []
group_nr = []
counter = 0
for i in range(data_df.shape[0]):
    rnd_list.append(randint(0, 100))
    if (i % GROUP_SIZE == 0):
        counter += 1
    group_nr.append(counter)
print(rnd_list)
print(group_nr)
data_df['rnd'] = rnd_list
data_df.head()
```

```
[4, 41, 38, 11, 96, 21, 11, 72, 61, 32, 32, 84, 54, 93, 42, 67, 72, 62, 85, 47, 28, 63, 82, 61]
[1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 12, 12]
```

[4]:

	Vorname	rnd
0	Silvy	4
1	StefanA	41
2	Dominic	38
3	Cora	11
4	Sabrina	96

Magic... ;-)

[5]:
```
data_df.sort_values(by=['rnd'], inplace=True)
data_df['group_nr'] = group_nr
data_df.drop(['rnd'], axis=1, inplace=True)
data_df
```

[5]:

	Vorname	group_nr
0	Silvy	1
3	Cora	1
6	Astrid	2
5	Ursina	2

Übungen
Analyse der konventionellen Betriebe

1) Wenn Sie noch nicht installiert haben, ist es jetzt höchste Zeit… Siehe Seite 2.

2) Laden Sie das Notebook aus Moodle herunter und führen Sie selber eine Analyse der *konventionellen* Landwirtschaftsbetriebe am Ende des Notebooks aus.

3) Tipp: Machen Sie dafür eine *Kopie* meines originalen Notebooks.