

# Einrichten von Python

## Kurzversion für Fortgeschrittene

Python 3 (3.7) zusammen mit einigen Paketen (-Versionen):

python	3.7.0
ipython	6.5.0
jupyter	1.0.0
jupyter_client	5.2.3
jupyter_core	4.4.0
notebook	5.6.0
matplotlib	2.2.3
numpy	1.15.1
pandas	0.23.4
patsy	0.5.0
scipy	1.1.0
statsmodels	0.9.0

[August 2018] Start des jupyter notebook

```
ImportError: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.20' not found
  (required by python3.7/site-packages/zmq/backend/cython/../../../../.
  ./libzmq.so.5)
```

lösbar durch downgrade der libgcc:

```
conda search libgcc
conda install libgcc=5.2.0
```

# Ausführlicher für Einsteiger

In der Vorlesung und in den Übungen kommt Python 3 zusammen mit einigen Paketen (-Versionen) zum Einsatz:

Damit die Übungen (und später natürlich auch richtige Berechnungen!) auf dem eigenen Computer laufen, sollte eine komplette Umgebung installiert werden.

- systemweit Python
  - apt-get install python3 / yum install python3 / ...
  - Download <https://www.python.org/downloads/windows/> (<https://www.python.org/downloads/windows/>) und installieren
  - Pakete
    - apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook python-panda (systemweit)
    - pip install xxx --user (innerhalb Python)
- Anaconda Distribution
  - <https://conda.io/docs/user-guide/install/download.html> (<https://conda.io/docs/user-guide/install/download.html>)
  - Paket-Manager "conda"

## Anaconda Distribution

Als Basis benutzt man am einfachsten die Anaconda-Python-Distribution (aktuell Python 3.7)

- kostenlos und frei
- Continuum Analytics, Austin TX,

Die Anaconda Distribution ist eine Software-Sammlung rund um Python, einschließlich der eigentlichen Python Programmiersprache, einem Manager *conda* und zahlreichen paketierten Bibliotheken (sog. *packages*), die aufeinander abgestimmt sind.

### Vorteile

- als User möglich, benötigt keinen root/Administrator-Zugang
- Versionskontrolle
- keine inkompatiblen Pakete
- virtuelle Umgebung:
  - spezielle Pakete und Versionen
  - kann geteilt werden
  - Pfade abgesichert vom System

### Nachteil

- jede Installation benötigt Platz (derzeit ca. 3GB)
- virtuelle Umgebung benötigt zusätzlichen Platz (nur diff zur Basis)

## miniconda

Man kann sich mit der kleineren Variante *miniconda* begnügen (Download hier <http://conda.pydata.org/miniconda.html> (<http://conda.pydata.org/miniconda.html>)), muß dann aber alle erforderlichen Pakete mit

```
conda install paketname
```

nachladen. Abhängigkeiten werden automatisch aufgelöst.

- Continuum verdient Geld mit Support und verwendet selbst Python für Analysen.

## Hinweise

### USB-Stick

- mit Dateisystem, auf dem Links angelegt werden können (zB. ext2)
- aktives Verzeichnis auf den Stick wechseln und einen relativen Pfad angeben
- Pfad hinzufügen, wenn keine Kollision mit systemeigenem python- bzw. Stick-Verzeichnis
  - sonst `export PATH=/path/of/Stick/anaconda3/bin:$PATH`

## [ana, mini] conda

### Überprüfen, ob conda funktioniert

**Linux / Mac OSX:** Öffne eine Konsole mit Prompt

**Windows:** Gib anaconda in das Suchfeld des Start-Menüs ein.

In der Kommandozeile `conda list` eingeben, das sollte die installierten Bibliotheken (packages) in der Arbeitsumgebung auflisten, beispielsweise:

Paketname	Version	
openssl	1.0.2p	h14c3975_0
pip	10.0.1	py37_0
python	3.7.0	hc3d631a_0
qt	5.9.6	h8703b6f_2
...		

Mehr zu conda hier: <https://conda.io/docs/user-guide/getting-started.html> (<https://conda.io/docs/user-guide/getting-started.html>). Wenn conda nicht gefunden wird, wurde die anaconda-Distribution nicht korrekt installiert.

## Pakete nachinstallieren oder aktualisieren

Manchmal benötigt man eine zusätzliche oder neuere Version einer Bibliothek. Mittels **conda** läßt sich diese leicht suchen und installieren:

```
conda list
conda search pandas
  Fetching package metadata: ....
  pandas      0.8.1      np16py26_0      pkgs/free
  pandas      0.8.1      np16py27_0      pkgs/free
  ....
  pandas      0.23.4     py36h04863e7_0  pkgs/main
  pandas      0.23.4     py36hf8a1672_0  conda-forge
  pandas      0.23.4     py37h04863e7_0  pkgs/main
  pandas      0.23.4     py37hf8a1672_0  conda-forge
```

Standard ist immer die neueste passende. Wenn man die Version 0.19.1 von pandas haben will :

```
conda install pandas=0.19.1
```

Abhängigkeiten zu anderen Bibliotheken/Paketen werden automatisch aufgelöst. Obacht: auch downgrade.

Alternativ kann auch das Python-eigene `pip install` benutzt werden.

```
pip install ggplot          # wenn nicht mittels conda verfügbar
```

## Python

### Python als (interaktive) Sprache

```
ipython
>>> print('hello world')
```

### Python als Interpreter/Compiler

```
python hello.py
```

# Python im Jupyter Notebook

- innerhalb des Programmtextes bewegen
- Programmzeilen und Daten ändern
  - zusammen mit den Ausgaben
    - Daten
    - Graphiken
- speichern
- weitergeben

Link: <http://jupyter-notebook.readthedocs.io/en/latest/> (<http://jupyter-notebook.readthedocs.io/en/latest/>)

## Aufruf

Das Notebook lebt im Browser (geliefert vom lokalen Server)

- Export als html, pdf, ...

**Linux / Mac OSX:** an der Kommandozeile

```
jupyter notebook
```

**Windows:** In der Startmenü-Suche

```
jupyter notebook
```

eingeben und anklicken.

## Formatierung

- Quelltext
- Kommentare als "*reStructured*" Text
- *LaTeX*-Formeln  $a^2+b^2=c^2$

Info zur *markdown* Syntax: <http://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/Working%20With%20Markdown%20Cells.html> (<http://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/Working%20With%20Markdown%20Cells.html>)

Ein Notebook wird im `.ipynb`-Format (*JSON*) gespeichert.

- (sprich: "Jason") JavaScript Object Notation

## Virtuelle Umgebung

Möchte man unterschiedliche Bibliotheken oder Versionen davon benutzen, bietet sich eine virtuelle Umgebung für jede Konfiguration an. Hier zB. Python 3.4, wenn miniconda im eigenen miniconda3 Verzeichnis installiert wurde:

```
~/miniconda3/bin/conda create -n myconda python=3.4 numpy=1.6    # Python in V
3.4, numpy in V1.6
source ~/miniconda3/bin/activate myconda
(myconda)[w@w ~/miniconda3]$
```

(Windows: activate myconda)

Das setzt Umgebungsvariablen wie zB. den Pfad korrekt. Nun reicht zum Starten des lokalen Notebook-Servers ein

```
jupyter notebook
```

und es öffnet sich der default-Browser mit der conda-Startseite: Inhalt des Startverzeichnisses mit beispielsweise Notebooks.

Die Umgebung verläßt man wieder mit

```
source deactivate
```

# Laborbuch

Man kann das [Notebook] als Laborbuch ansehen und verwenden.

- Ideen zu Papier<sup>H^H^H^H^H</sup> Bits bringen
- Schnelle Berechnungen ausführen
- Daten analysieren mitsamt graphischer Ausgabe, Statistik, ...

Link: <http://jupyter-notebook.readthedocs.io/en/latest/index.html> (<http://jupyter-notebook.readthedocs.io/en/latest/index.html>)

Der Umgang ist einfach und fällt nach einer gewissen Einarbeitungsphase leicht.

- <https://github.com/jupyter/notebook/blob/master/docs/source/examples/Notebook/Running%20Code.ipynb> (<https://github.com/jupyter/notebook/blob/master/docs/source/examples/Notebook/Running%20Code.ipynb>),
- [http://bebi103.caltech.edu/2015/tutorials/t0b\\_intro\\_to\\_jupyter\\_notebooks.html](http://bebi103.caltech.edu/2015/tutorials/t0b_intro_to_jupyter_notebooks.html) ([http://bebi103.caltech.edu/2015/tutorials/t0b\\_intro\\_to\\_jupyter\\_notebooks.html](http://bebi103.caltech.edu/2015/tutorials/t0b_intro_to_jupyter_notebooks.html)) oder

Und hier <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks> (<https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>) ein Ausblick, was man machen kann.

**Auch diese Vorlesung ist in Notebooks konzipiert.**

Gerne darf man Text und Code daraus in seinem eigenen Notebook verwenden.

## Veröffentlichen

- <http://nbviewer.jupyter.org/> (<http://nbviewer.jupyter.org/>)
- <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks#statistics-machine-learning-and-data-science> (<https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks#statistics-machine-learning-and-data-science>)

# Bibliotheken

Wir verwenden einige Bibliotheken. Alle sind *open source* und werden von Freiwilligen entwickelt und betreut.

Dokumentationen finden sich ausführlich im Web, die meisten Pakete haben eigene Webseiten.

Außerdem in den Paketen, sogar im Notebook zugänglich:

- Tab-Taste zur Auto-Vervollständigung / Auswahl
- Fragezeichen zum Hilfeaufruf

Hauptsächlich verwenden wir für die statistischen Analysen

Bibliothek	Link
scipy.stats	<a href="http://docs.scipy.org/doc/scipy/reference/stats.html">http://docs.scipy.org/doc/scipy/reference/stats.html</a> ( <a href="http://docs.scipy.org/doc/scipy/reference/stats.html">http://docs.scipy.org/doc/scipy/reference/stats.html</a> )
statsmodels	<a href="http://statsmodels.sourceforge.net/stable/index.html">http://statsmodels.sourceforge.net/stable/index.html</a> ( <a href="http://statsmodels.sourceforge.net/stable/index.html">http://statsmodels.sourceforge.net/stable/index.html</a> )
Pandas	<a href="http://pandas.pydata.org/pandas-docs/stable/index.html">http://pandas.pydata.org/pandas-docs/stable/index.html</a> ( <a href="http://pandas.pydata.org/pandas-docs/stable/index.html">http://pandas.pydata.org/pandas-docs/stable/index.html</a> )
	aufbauend auf
numpy	und
scipy	<a href="http://docs.scipy.org/doc/">http://docs.scipy.org/doc/</a> ( <a href="http://docs.scipy.org/doc/">http://docs.scipy.org/doc/</a> )
matplotlib	<a href="http://matplotlib.org/">http://matplotlib.org/</a> ( <a href="http://matplotlib.org/">http://matplotlib.org/</a> )

## und nun viel Spaß...

```
In [1]: print('Hallo Statistik-Studentinnen und -Studenten!')  
Hallo Statistik-Studentinnen und -Studenten!
```