

GEO418, Data Analysis Part

martin.bachmann@dlr.de

Stand: 07/2022

Vorbemerkung

- Meine Beispiele sind in **Python** (3.x) & komplett **OpenSource**
=> ich empfehle euch, dies auch zu installieren (s. nächste Folien)
- Wenn ihr euch sehr gut mit R oder IDL auskennt, könnt ihr dies aber auch verwenden
- Oder notfalls: einfach zuhören & zusehen ;) ich habe damit kein Problem!

Für die **Abschlussarbeit**:

- **Komplett freie Wahl des Auswerteprogramms!**
Python, R, IDL oder EXCEL (!) / OpenOffice... gehen allesamt.
Ich bereite euch die Daten so auf, dass dies problemlos geht!

Zusätzlich verwende ich für Demos (müsst ihr nicht nachvollziehen):

- IDL & ENVI

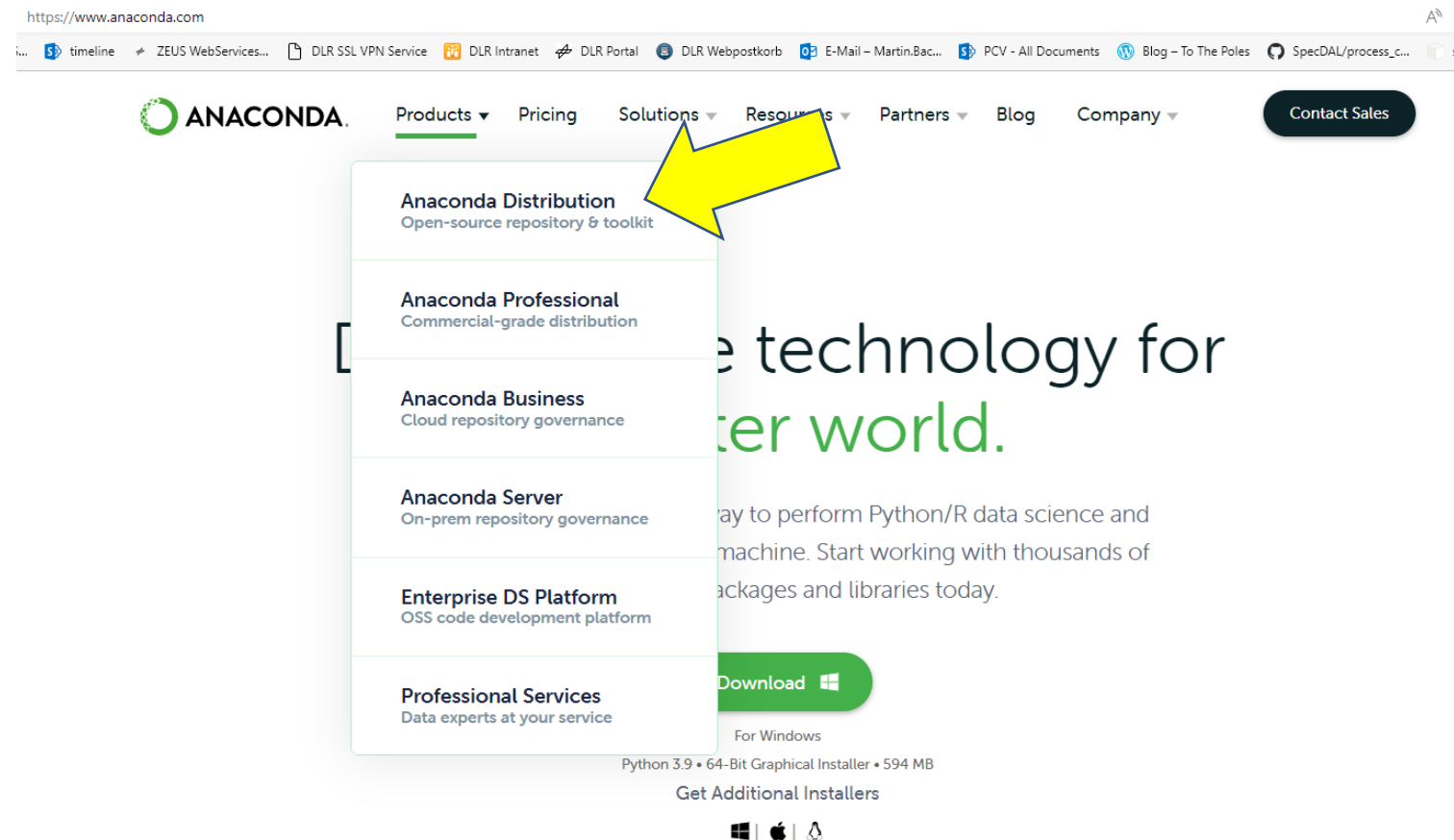
Hier könntet ihr z.B. QGIS als OpenSource Alternative verwenden

Installation – Python via ANACONDA

Empfehlung: Anaconda Distribution
<https://www.anaconda.com/>

„Anaconda Distr.“
ist OpenSource,
kostenlos und
vereinfacht vieles

Gibt's für Win, Mac und Linux

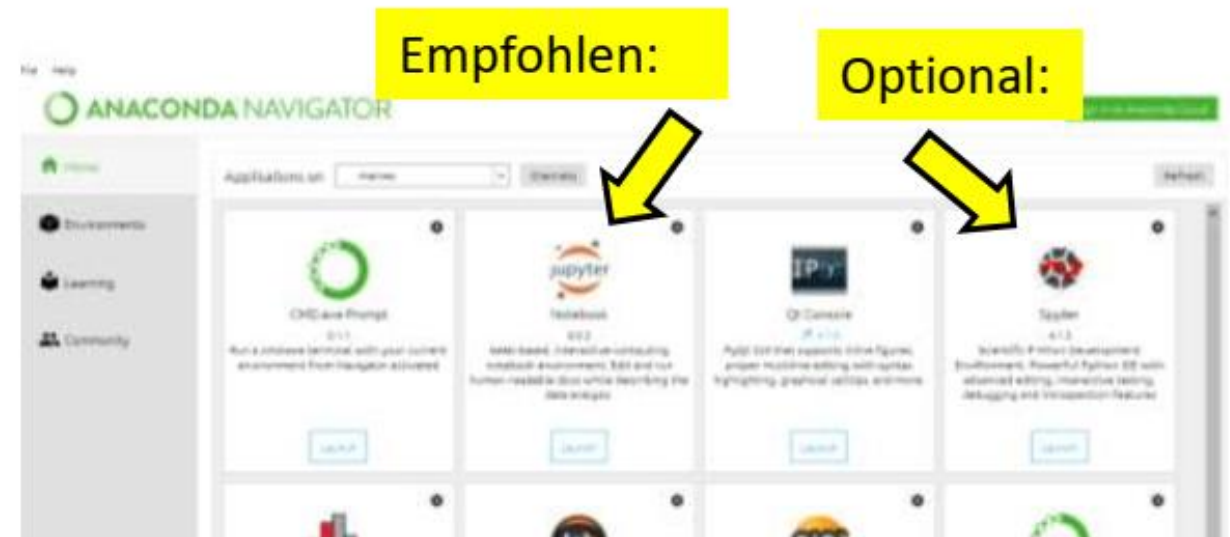


Installation Python (alternativ)

- Siehe:
<https://www.python.org/downloads/>

„Zusätze“ zu Python

- Weiterhin verwende ich (für euch optional, aber empfohlen):
 - Jupyter Notebook (=> Python läuft dann „im Webbrowser“)
 - Konsole / Editor eurer Wahl, ich verwende Spyder hierfür



Installation ohne ANACONDA:

- <https://docs.spyder-ide.org/current/installation.html>
- <https://jupyter.org/install.html>

... nach Installation von Python:

- Python packages installieren
 - Anacond / Conda: siehe nächste Folie
 - Allgemein an der command line via:

`pip install „name_des_package“`

Siehe: <https://packaging.python.org/tutorials/installing-packages/>

Benötigt werden:

- numpy => Matrizen etc.
- scipy => Filter etc.
- scikit-learn => Machine learning tools
- gdal, libgdal => Geodaten I/O
- matplotlib => für Diagramme & Visualisierung
- seaborn => verbesserte Datenvisualisierung
- tk => zur Erstellung von GUIs

Optional:

- csvkit => File I/O zum CSV Format
- pandas => verbessertes Datenhandling

Installation mit Conda / Anaconda:

The screenshot shows the Anaconda Navigator application window. The interface includes a sidebar on the left with 'Home', 'Environments', 'Learning', and 'Community' options. The 'Environments' section is active, showing a list of environments: 'base (root)', 'fcover', and 'meines'. The 'base (root)' environment is selected. The main panel displays a search for 'numpy' packages. The search results table lists various packages, with 'numpy' highlighted. The 'Apply' button is visible at the bottom right.

Hier „All“ auswählen...

... nach den Packages suchen...

U.U. dann ein Environment auswählen (base ist aber auch möglich)

... auswählen

... und „Apply“

Name	Description	Version
blaze	Numpy and pandas interface to big data	0.11.3
bottleneck	Fast numpy array functions specialized for use in orange	0.7.1
bottleneck	Fast numpy array functions written in cython	1.3.2
cupy	Cupy is an implementation of a numpy-compatible multi-dimensional array on cuda	6.0.0
ml_fft	Numpy-based implementation of fast Fourier transform using intel (r) math kernel library	1.1.0
ml_random	Intel (r) ml-powered package for sampling from common probability distributions into numpy arrays	1.1.1
msgpack-numpy	Numpy data serialization using msgpack	0.4.4.3
numba	Numpy aware dynamic python compiler using llvmlite	0.49.1
numexpr	Fast numerical expression evaluator for numpy	2.7.1
numpy	Array processing for numbers, strings, records, and objects	1.9.3
numpy-base		1.9.3
numpy-devel		1.9.3
numpydoc	Sphinx extension to support docstrings in numpy format	1.0.0
pytables	Brings together python, hdf5 and numpy to easily handle large amounts of data	3.6.1
snuggs	Snuggs are s-expressions for numpy	1.4.7

15 packages available matching "numpy" 1 package selected

Sign in to Anaconda Cloud

All

Channels

Update index...

numpy

Name	T	Description	Version
<input type="checkbox"/> blaze		Numpy and pandas interface to big data	0.11.3
<input type="checkbox"/> bottleneck		Fast numpy array functions specialized for use in orange	0.7.1
<input type="checkbox"/> bottleneck		Fast numpy array functions written in cython.	1.3.2
<input type="checkbox"/> cupy		Cupy is an implementation of a numpy-compatible multi-dimensional array on cuda.	6.0.0
<input checked="" type="checkbox"/> mkl_fft		Numpy-based implementation of fast fourier transform using intel (r) math kernel library.	1.1.0
<input checked="" type="checkbox"/> mkl_random		Intel (r) mkl-powered package for sampling from common probability distributions ...	1.1.1
<input type="checkbox"/> msgpack-numpy		Numpy data serialization using msgpack	0.4.4.3
<input checked="" type="checkbox"/> numba		Numpy aware dynamic python compiler using llvm	0.49.1
<input type="checkbox"/> numexpr		Fast numerical expression evaluator for numpy.	2.7.1
<input checked="" type="checkbox"/> numpy		Array processing for numbers, strings, records, and objects.	1.18.5
<input checked="" type="checkbox"/> numpy-base			1.18.5
<input type="checkbox"/> numpy-devel			

... und so wird's angezeigt, wenns korrekt installiert ist

... und auch „Spectral Python“:

... benötigt „händische“ Installation:

Innerhalb CONDA / Anaconda:

<https://anaconda.org/conda-forge/spectral>

– *To install this package with conda run one of the following:*

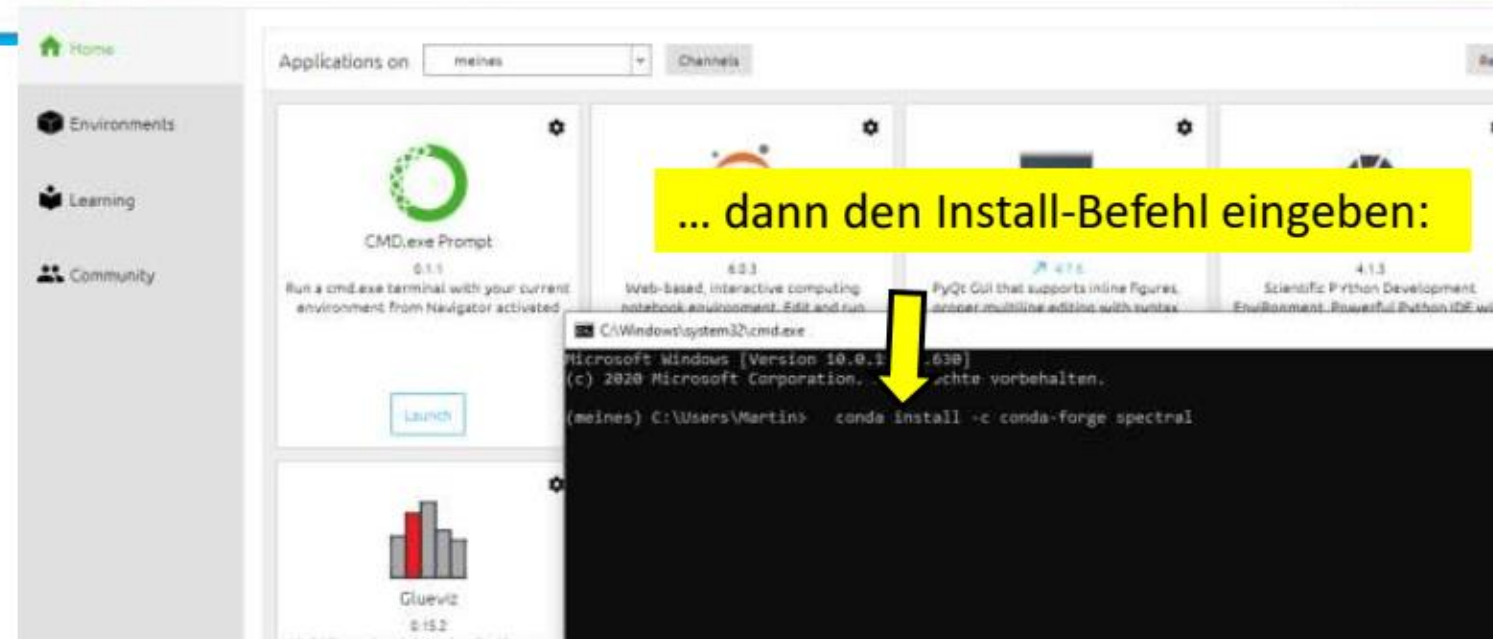
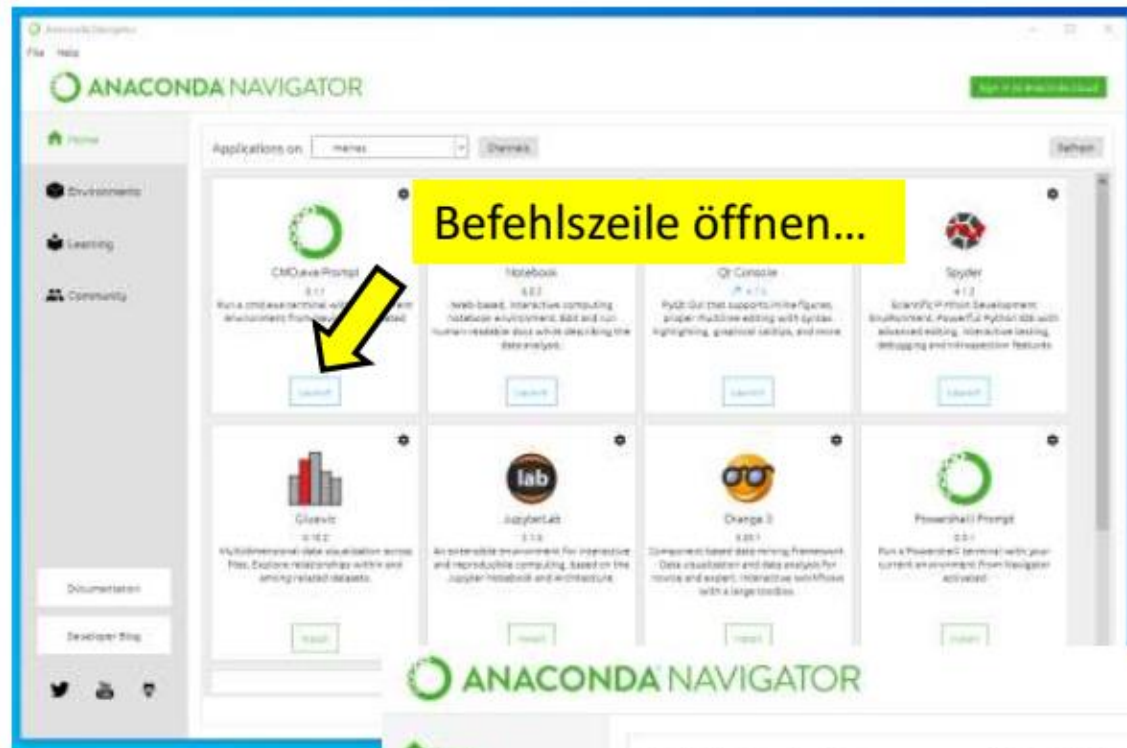
conda install -c conda-forge spectral

conda install -c conda-forge/label/gcc7 spectral

conda install -c conda-forge/label/cf201901 spectral

conda install -c conda-forge/label/cf202003 spectral

Siehe nächste Folie:



Alternativ ohne CONDA:

<http://www.spectralpython.net/installation.html>

– Via:

- pip install spectral
oder
- easy_install spectral