Course Summary

Block1

Session 1: Getting started

Goals

1) navigate in the shell application: cmd or terminal
2) have python and anaconda installed, all required packages installed
3) now how to open and close, Python console, IPython console, jupyter notebook

Content

Shell: learn shell commands like `ls, cd, pwd, mkdir, cp, mv, rm, rm -r, touch, echo, cat` (and the associated commans on windows cmd) Python: download and installing anaconda, open python console, close it, install packages using anaconda. IPython: open console, getting help, advantage compared to python console. Jupyter notebook: open it, creat a notebook, write code in cells, write markdown in cells, execute and create cells, delete cells, restart the kernel, close the notebook, stop the notebook server.

Resources

- https://www.codecademy.com/learn/learn-the-command-line
- https://www.python.org/about/
- https://ipython.readthedocs.io/en/stable/
- https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook
- https://jupyter-notebook.readthedocs.io/en/stable/
- https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

Exercise

- https://www.practicepython.org/exercise/2014/01/29/01-character-input.html

---

Session 2: Basics of python

Goals

1) Know different ways to write and execute python code: console, script, jupyter notebook
2) Be familiar with python variables and data types, know how to access the documentation, "everything is an object!"
3) Be familiar with booleans, dictionaries, if-else statements, loops, indexing, slicing, mutability, generators, iterators

Content

- python data types: `boolean, int float, string, list, dict, tuple`
- operators: `+, -, /, *, **, %`
- methods associated with data types, e.g., essential string methods, essential list methods.
- if-elif-else statements, conditional variable assignment, `in` operator
- list and tuple indexing, dict indexing, mutability vs. immutability,
- `while` loops, `for` loops, concept of iterator and generator, using `enumerate`, and `zip`, combining them.
- iterating through dictionaries
- list comprehension, conditional list comprehension

Resources

- https://github.com/cne-tum/msne-datascience-2018/blob/master/notebooks/block1/block1_session2_bas
- https://github.com/cne-tum/msne-datascience-2018/blob/master/notebooks/block1/block1_session2_pro
- https://www.codecademy.com/learn/learn-python
- https://s3.amazonaws.com/assets.datacamp.com/blog_assets/PythonForDataScience.pdf

Exercises

- https://www.practicepython.org/exercise/2014/02/26/04-divisors.html
- https://www.practicepython.org/exercise/2014/03/19/07-list-comprehensions.html
- https://www.practicepython.org/exercise/2017/01/24/33-birthday-dictionaries.html
- https://www.datacamp.com/courses/intro-to-python-for-data-science

---

Session 3: Git and GitHub

Goals

1) Understanding the concept of version control and its importance
2) Differentiation between Git and GitHub
3) Familiarize with the Git (version) control terminology
4) Creating repositories and practicing common workflow

Content

- Git is a version control system that:
  - Keeps track of changes to files (**who** made **what** changes, **when** and **why**).
  - Notice conflicts between changes made by different people.
  - Synchronize files between different computers.
- GitHub is a web-based hosting service for version control using Git, with some extra features (e.g, access control, etc.)
- Common practices (workflow) in git
  - clone (i.e, donwload) a (remote) repository from GitHub
  - (Make branches and) make changes on your local repository (copy of the remote repo)
  - commit (i.e., save) the changes
  - push the changes to the remote repo (update the remote repo wrt the local repo)
  - pull changes from the remote repo (update the local repo wrt the remote repo)
- Create a branch for a change, choose descriptive names for the branch, and keep the number of branches low. Once new changes are stable, merge them with the master branch and remove the other branch.

Resources

- A nice interactive course offered by DataCamp.com (link)
- A collection of resources to learn git (link)

Exercises

1. you can complete the DataCamp course mentioned above.
2. (if you haven;t done it already,) please crete a repository on github and clone it.
3. withing this repo, on your local machine, open a juputer notebook and solve the exercises for Numpy and Matplotlib (can be found in the following sections).

Session 4: numpy

Goals

1) What is a Numpy array, and why to use them? (motivation)
2) Importing and Generating Data
3) Getting insight about the Data (type, dimension, size, etc.)
4) Manipulating the array (arithmetic operations, transpose, etc.)
5) Slicing and Masking
6) Combining arrays
7) Saving data

Content

- Import data:
  - `np.load()`
  - `np.loadtxt()`
  - `np.genfromtxt()`
- Creating Numpy arrays
  - `np.zeros()`
  - `np.ones()`
  - `np.random.random()`
  - `np.empty()`
  - `np.full()`
  - `np.full_like()`
  - `np.eye()`
  - `np.identity()`
- Data Inspection (assuming we have numpy a array object called `data`)
  - `data.dtype`
  - `data.ndim`
  - `data.shape`
  - `data.size`
  - `data.strides`
  - `data.min()`
  - `data.max()`
  - `data.mean()`
  - `data.std()`
  - `data.cumsum()`
- Data Transformation (assuming we have a numpy array object called `data`)
  - `data.T`
  - `data.reshape()`
  - `data.resize()`

  - `np.expand_dims()`
  - `np.ravel()`
  - `np.add()`, `np.subtract()`, `np.multiply()`, `np.divide()`, `np.remainder()`
  - `np..exp()`, `np.log()`
  - Masking using list (or array) of True and False values
- Combining multiple arrays ans splitting an array
  - `np.concatenate()`
  - `np.append()`
  - `np.hstack()`
  - `np.vstack()`
  - `np.hsplit()`
  - `np.vsplit()`
- Saving numpy arrays
  - save(): saves data in .npy format
  - savez(): Save several arrays into an uncompressed .npz archive
  - savez_compressed():
  - savetxt():

Resources

- https://www.datacamp.com/community/tutorials/python-numpy-tutorial
- Cheat sheet

Exercises

Please refer to the notebook (called "Numpy_exercises.ipynb") provided in the course repository.

Session 5: matplotlib

Goals

- Create different types of figures
- Customize the figure
- Put several figures together
- Save the figure

Content

- Create a simple line plot using `plt.plot()`
- Create a simple scatter plot using `plt.scatter()`
- Modify the data representation (line color, width, point size, markers, and style)
- Modify the axes (xlim, ylim, ticks and ticklabels, etc.)
- Save a (high quality) figure using `plt.savefig()`

Resources

- https://www.datacamp.com/courses/introduction-to-data-visualization-with-python (first two blocks)
- https://matplotlib.org/gallery.html
- https://github.com/matplotlib/AnatomyOfMatplotlib
- https://github.com/jbmouret/matplotlib_for_papers
- https://jakevdp.github.io/blog/2013/07/10/XKCD-plots-in-matplotlib/
- https://jakevdp.github.io/blog/2012/10/07/xkcd-style-plots-in-matplotlib/
- Cheat sheet

Exercises

Re-create the figure below.

```
<img src ="notebooks/block1/img/ExercisePlot.png" height="800" width="800"/>
```