Data Analysis (Part 1)

NumPy, Scipy

L. Kos[1] G. Simič[1]

[1]Faculty of Mechanical Engineering
University of Ljubljana

EUDAT-PRACE summer school, September 2019

Overview

Demonstration and hands-on login.galileo.cineca.it

- Running a Jupyter notebook session
- Demonstrating numpy aspects
- Performing computations
- SciPy and Pandas
- Visualizing ECAS data

Examples for this session:

```
git clone https://gitlab.eudat.eu/eudat-prace-2019/intro-to-numpy-scipy.git
```

Note

`ecas-training-data.git` is not required for traning on Galileo cluster.

Running a Jupyter notebook session

For reading and writing numpy, scipy, pandas notebooks and netCDF4 files we need:

```
# Load anaconda module
module load anaconda/2019.07
# Initialize conda shell
conda init bash
# Create custom environment
conda create --prefix my_own_env
# Activate environment
conda activate ./my_own_env
# Install packages
conda install netCDF4 numpy scipy basemap \
matplotlib jupyter xlrd pandas
```

Create custom forward sbatch script

```
cd forward
# Copy the jupyter-conda.sbatch script
cd sbatches/galileo
cp jupyter-conda.sbatch jupyter-custom.sbatch
```

Before the last line in jupyter-custom.sbatch insert

conda activate ~/my_own_env

Starting and connecting to Jupyter session:

```
# Start a Jupyter session on galileo
cd forward
bash start.sh jupyter-custom
```

In the output a link is provided. Open the link on your local machine to connect to the Jupyter session.

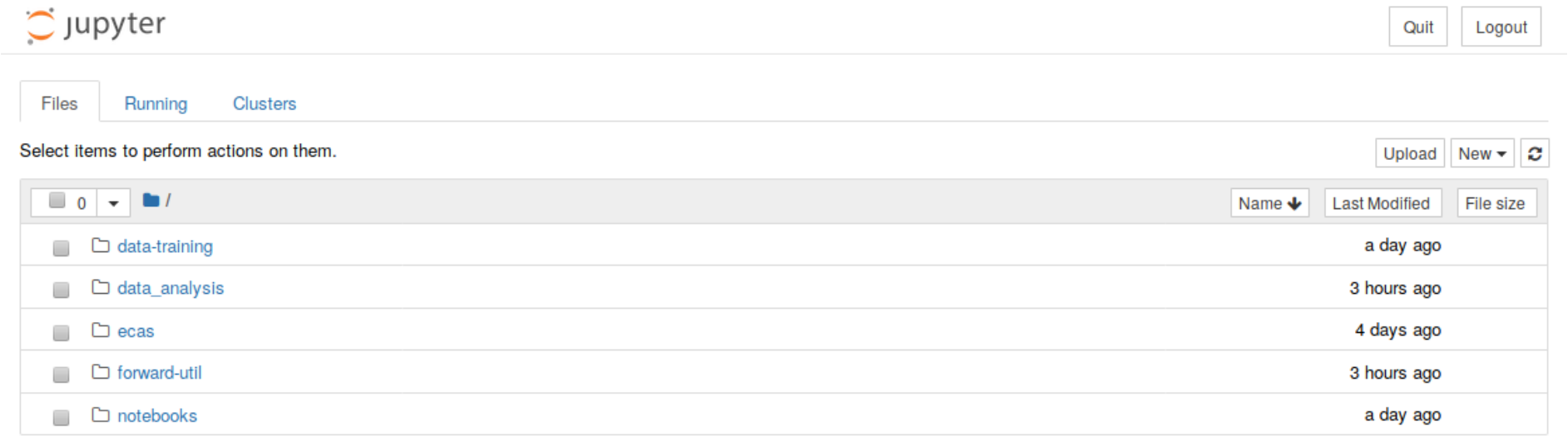
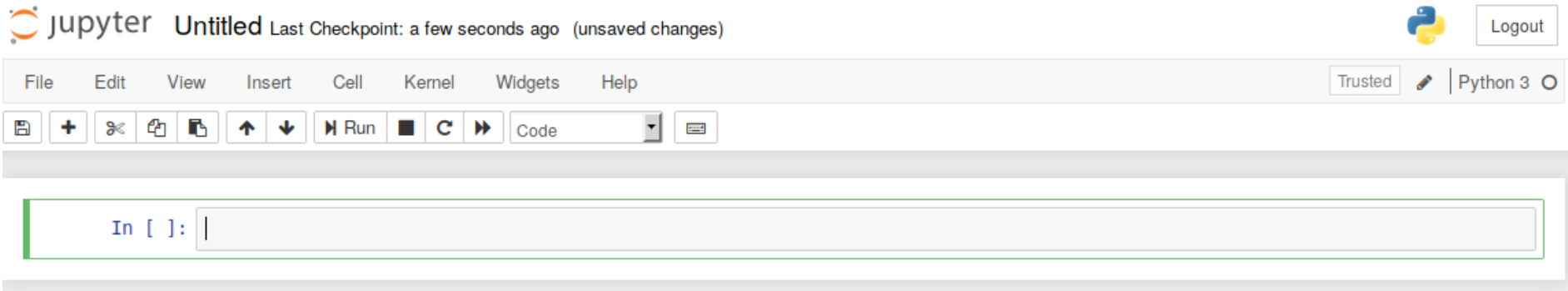


Figure: Jupyter session

Click on New and select Python 3



Numpy fast array interface

- Standard Python is not well suitable for numerical computations
  - lists are flexible but slow for using in numerical computations
- Numpy adds a new numpy **array** data type
  - Static, multidimensional
  - All elements of the array have the same type
  - Fixed number of elements in array, it's shape can be changed

Advantages of Numpy arrays over Python lists:

- Array operators: +, *, /, **
- Vectorization: Less for loops to write, better looking code
- Array operations and functions

Numpy acts as a base for writing numerical code in Python. Numpy arrays can be passed to C, C++ or Fortran, thus extending python with C/C++/Fortran.

Scipy, numerical library that provides user-friendly and efficient numerical routines for numerical integration, interpolation, optimization, linear algebra and statistics.

Written to use numpy arrays, thus functions take and operate on numpy arrays. For linear algebra it calls the BLAS/LAPACK functions, therefore it is recommended to have an optimised BLAS (ATLAS, MKL, OpenBLAS) installed.

Pandas, a high-performance, easy-to-use data structure and data analysis tools for Python.

Used for work with relational or labeled data easy and intuitive. It provide tools for inspection, statistics and visualization of data.

Hands-on session

Prepared Jupyter notebooks explain the basics needed for handling data in the following sessions.

numpy_examples Basics of NumPy: Operators, array manipulation,...

numpy_mandelbrot Calculating the Mandelbrot set using NumPy

scipy_find_minimum Search extrema with scipy

scipy_fitting_function Fitting a function through data using scipy

tasMaxMin ECAS data handling examples