Università di Genova
DICCA DIPARTIMENTO
DI INGEGNERIA CIVILE, CHIMICA
E AMBIENTALE

Python for Scientists

2nd Edition | April & May, 2022

Department of Civil, Chemical and Environmental Engineering (DICCA)
University of Genoa, Italia

Andrea Lira Loarca
andrea.lira.loarca@unige.it

Overview of the course

Team code
fi0pjq9

- Introduction to python, editors, scientific libraries
- Basics of python programming
- Numpy
- Matplotlib
- Scipy
- Pandas
- Overview of advanced libraries - Xarray, Cartopy, Seaborn, Scikit-Learn
- Exercises proposed by students

Why Python?

Scientific's needs

- Get data (simulations, experiments, databases)
- Manipulate and process data
- Visualize results, quickly to understand, but also with high quality figures, for reports or publications

> I think every kid in the world should learn to code. Whether your passion is in science or the arts, it's a way to express yourself.
>
> Tim Cook CEO OF APPLE

DATA SCIENTISTS

Why 'Data Scientist' Will Continue To Be 'the Sexiest Job Of the 21st Century'

As technology evolves, the exact skills and nature of jobs in data science will probably vary

y 2025, 463 exabytes of data will be created each day, according to some estimates. (For perspective, one exabyte of storage could hold 50,000 years of DVD-quality video.) It's now easier than ever to translate physical and digital actions into data, and businesses of all types have raced to amass as much data as possible in order to gain a competitive edge.

Data alone doesn't spur innovation — rather, it's data-driven storytelling that helps uncover hidden trends, powers personalization, and streamlines processes.

Published: 04 February 2015

Programming: Pick up Python

Jeffrey M. Perkel

Nature **518**, 125–126 (2015) | Cite this article

278 Accesses | **51** Citations | **689** Altmetric | Metrics

A powerful programming language with huge community support.

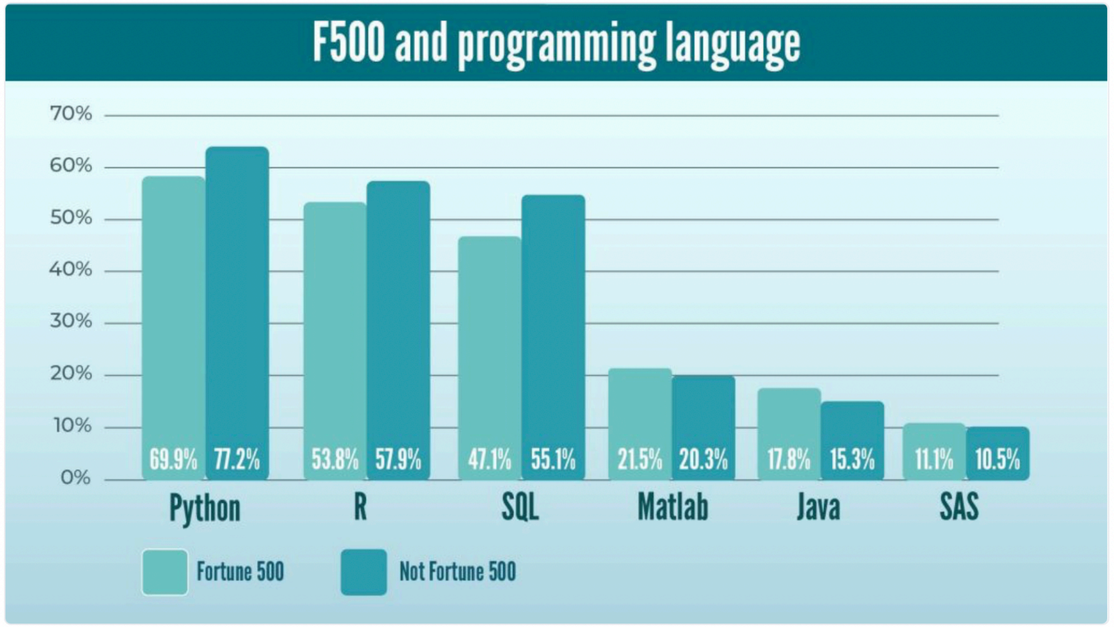
Credit: Illustration by the Project Twins

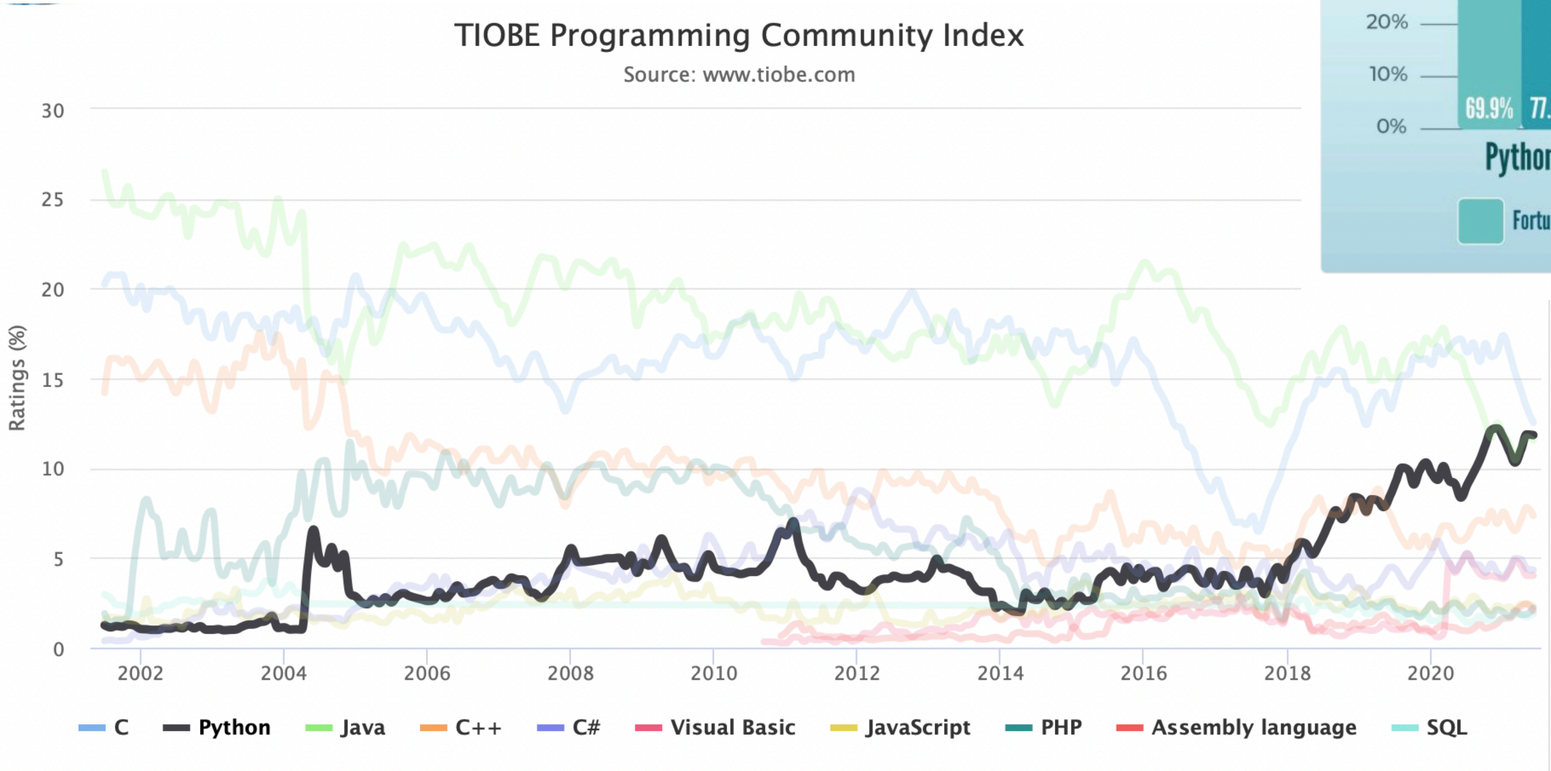
Why Python?

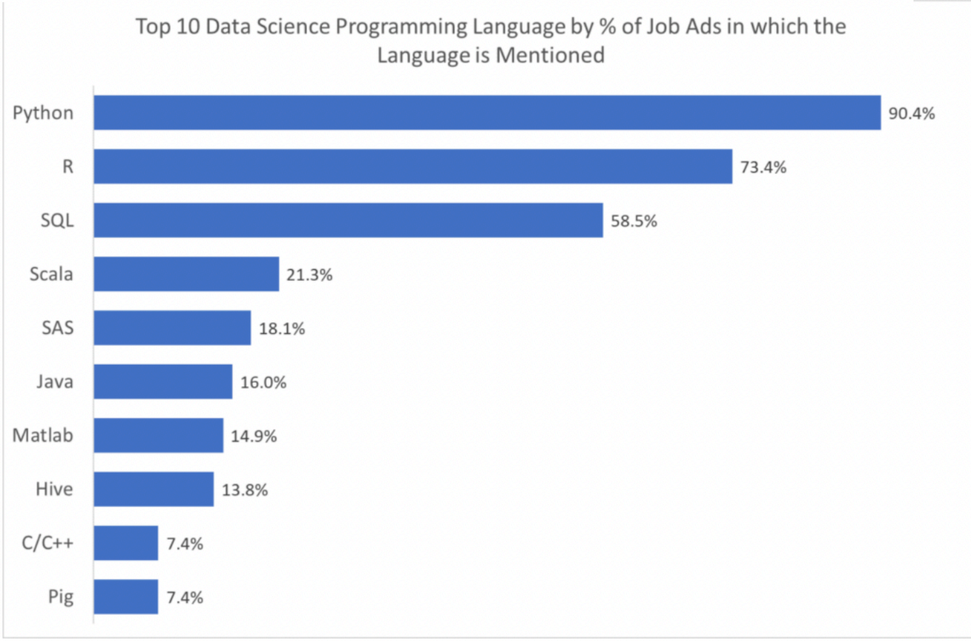
Python's strengths

- Extensive library: classic numerical methods, plotting or data processing tools. We don't want to re-program the plotting of a curve, a Fourier transform or a fitting algorithm. Don't reinvent the wheel!-
- Easy to learn: most scientists are not payed as programmers, neither have they been trained so. They need to be able to draw a curve, smooth a signal, do a Fourier transform in a few minutes.
- Easy communication: to keep code alive within a lab or a company it should be as readable as a book by collaborators, students, or maybe customers.
- Efficient code: Python numerical modules are computationally efficient.***
- Universal Python is a language used for many different problems: Learning Python avoids learning a new software for each new problem.
- Strong community: google new libraries when approaching a problem.

Python vs. Other Languages







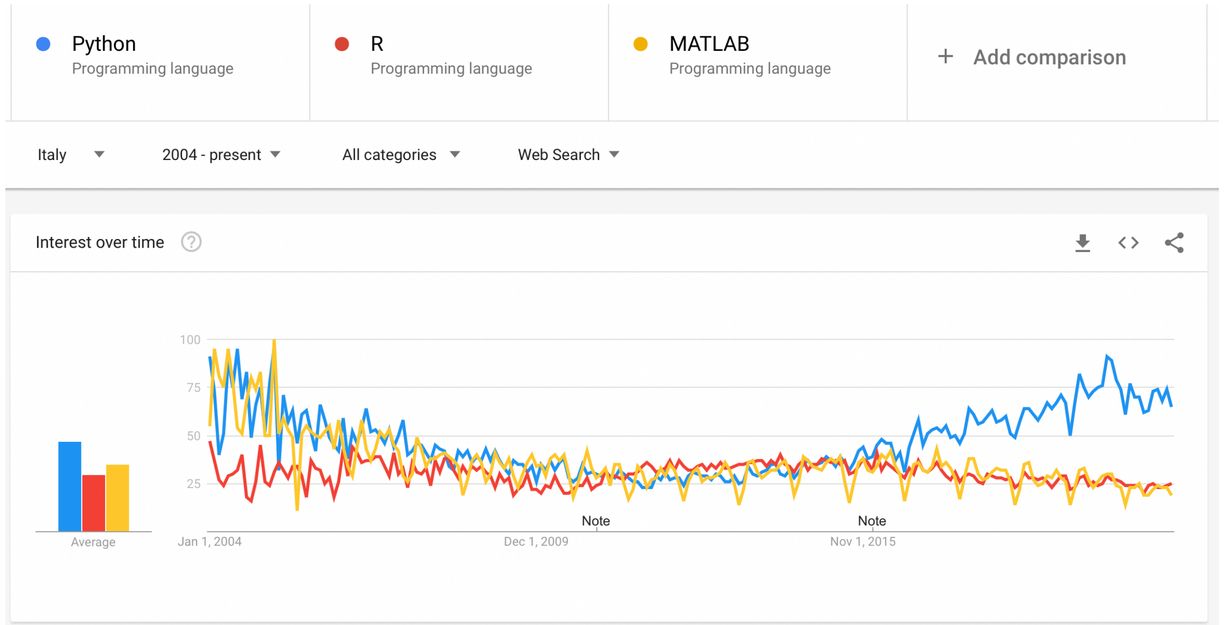
Python vs. Other Languages

Python

Pros:	• Very rich scientific computing libraries • Well thought out language, allowing to write very readable and well structured code: we “code what we think”. • Many libraries beyond scientific computing (web server, serial port access, etc.) • Free and open-source software, widely spread, with a vibrant community. • A variety of powerful environments to work in, such as IPython, Spyder, Jupyter notebooks, Pycharm, Visual Studio Code
Cons:	• Not all the algorithms that can be found in more specialized software or toolboxes.

Matlab scripting language

Pros:	• Very rich collection of libraries with numerous algorithms, for many different domains. Fast execution because these libraries are often written in a compiled language. • Pleasant development environment: comprehensive and help, integrated editor, etc. • Commercial support is available.
Cons:	• Base language is quite poor and can become restrictive for advanced users. • Not free.



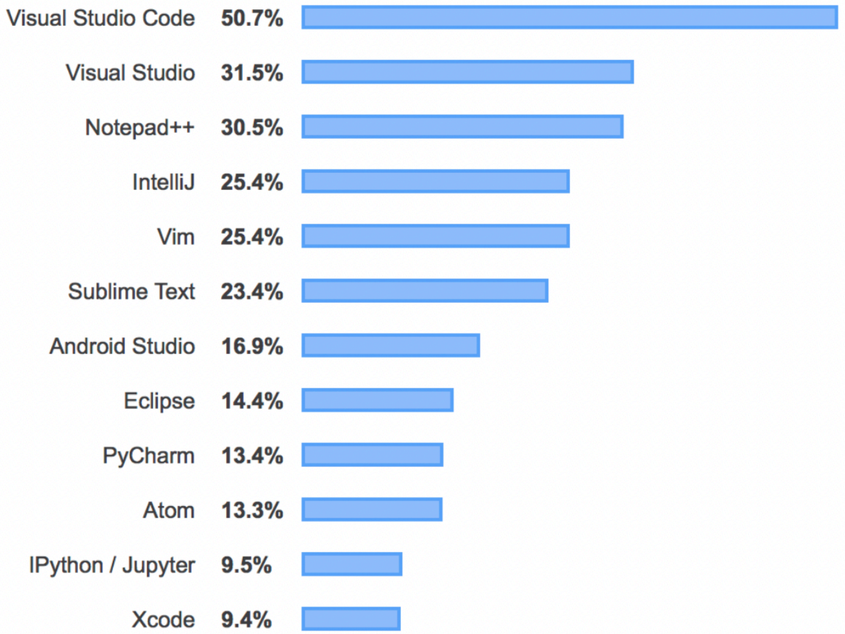
How to run Python?

1. Distribution - set of libraries, packages

2. Editor or application

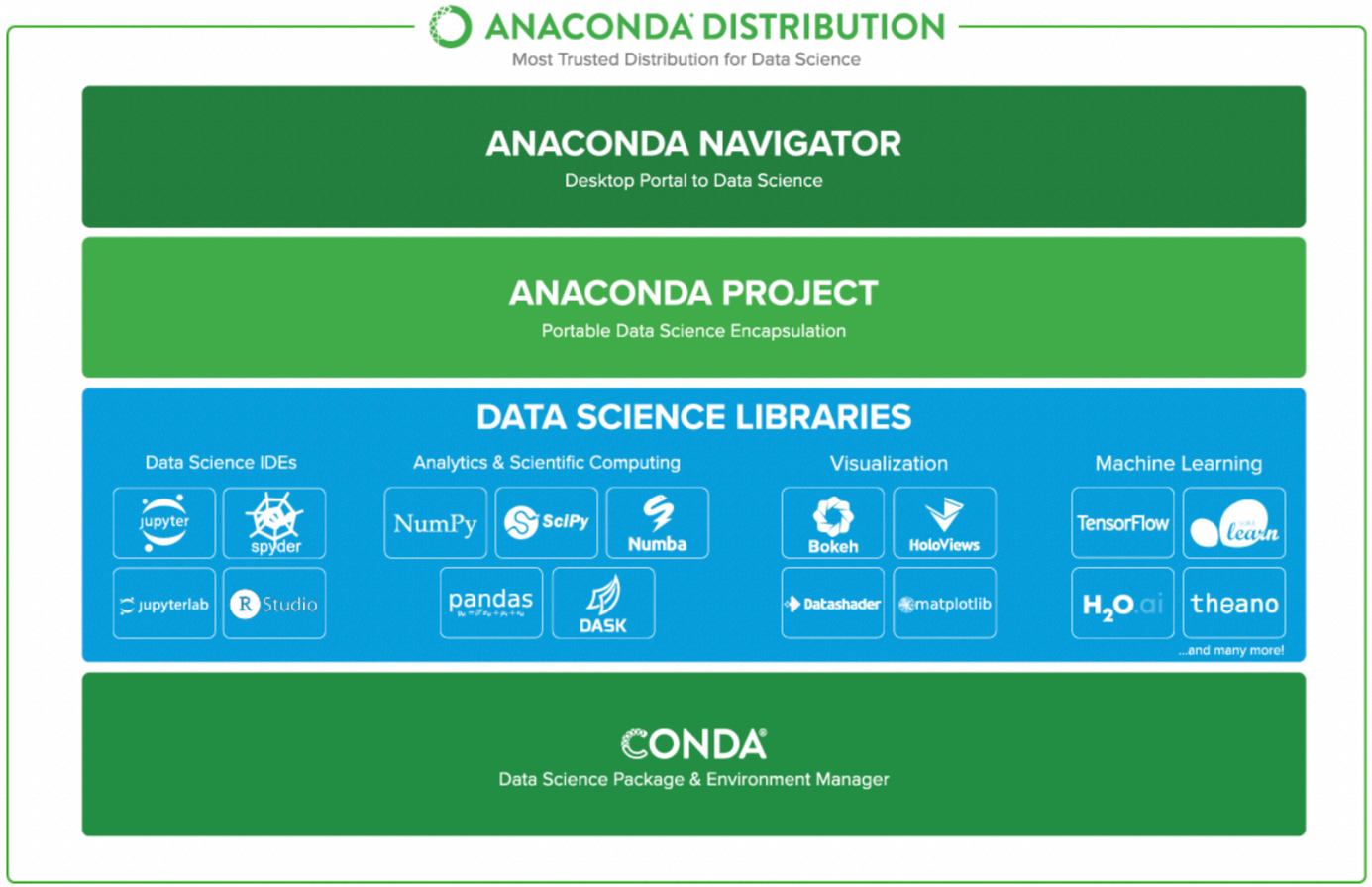
- Notepad++
- Vim
- Sublime Text
- VS Code
- Spyder
- Atom
- Pycharm
- Jupyter

…



Scientific-based Python distribution - Anaconda

https://www.anaconda.com/products/individual



Scientific-based Python distribution - Anaconda

The Fundamentals

Jupyter is an open-source project created to support interactive data science and scientific computing across programming languages. Jupyter offers a web-based environment for working with notebooks containing code, data, and text. Jupyter notebooks are the standard workspace for most Python data scientists.

A library for tabular data structures, data analysis, and data modeling tools, including built-in plotting using Matplotlib. pandas aims to be the fundamental high-level building block for doing practical, real world data analysis in Python

The SciPy library consists of a specific set of fundamental scientific and numerical tools for Python that data scientists use to build their own tools and programs. It provides many user-friendly and efficient numerical routines, such as routines for numerical integration, interpolation, optimization, linear algebra, and statistics.

A core package for scientific computing with Python. Numpy enables array formation and basic operations with arrays. Numpy is used for indexing and sorting but can also be used for linear algebra and other operations. Many other data-science libraries for Python are built on NumPy internally, including Pandas and SciPy.

Data Visualization

Matplotlib is the most well-established Python data visualization tool, focusing primarily on two-dimensional plots (line charts, bar charts, scatter plots, histograms, and many others). It works with many GUI interfaces and file formats, but has relatively limited interactive support in web browsers.

Bokeh is an interactive visualization library for modern web browsers. It provides elegant, concise construction of versatile graphics, and affords high-performance interactivity over large or streaming datasets. Bokeh can help anyone who would like to quickly and easily make interactive plots, dashboards, and data applications.

Bokeh

Plotly's Python graphing library makes interactive, publication-quality graphs. It is a popular and powerful browser-based visualization library that lets you create interactive, JavaScript-based plotsfrom Python.

HoloViz is an Anaconda project to simplify and improve Python-based visualization by adding high-performance server-side rendering (Datashader), simple plug-in replacement for static visualizations with interactive Bokeh-based plots (hvPlot), and declarative high-level interfaces for building large and complex systems (HoloViews and Param).

Scientific-based Python distribution - Anaconda

Scalable Computing

Numba

Numba is a high-performance Python compiler. It makes Python faster and optimizes the performance of Numpy arrays, reaching the speed of FORTRAN and C without a an additional compilation step.

DASK

Dask is a Python package used to scale NumPy workflows with parallel processing to enable multi-dimensional data analysis, enabling users to store and process data larger than their computer's RAM. Dask can scale out to clusters, or scale down to a single computer. Dask mimics the pandas and NumPy API, making it more intuitive for Python data scientists.

RAPIDS

The RAPIDS data science framework is a collection of libraries for running end-to-end data science pipelines completely on the GPU. The interaction is designed to have a familiar look and feel to working in Python, but utilizes optimized NVIDIA® CUDA® primitives and high-bandwidth GPU memory under the hood.

Apache Spark

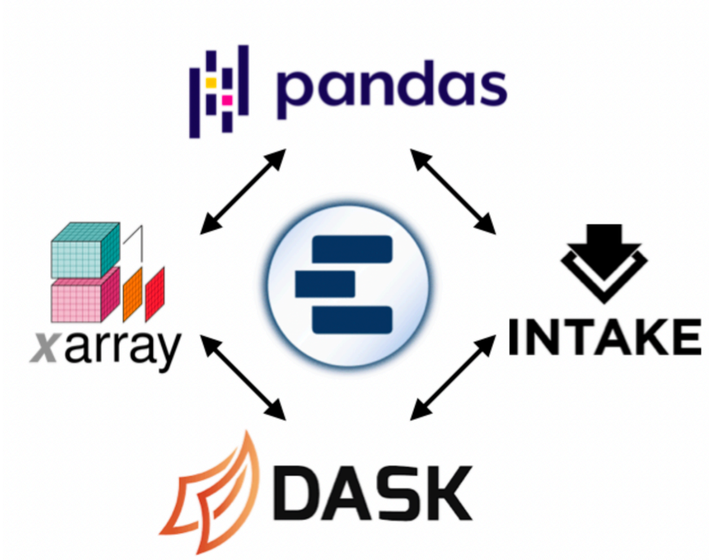
A fault-tolerant cluster computing framework and interface for programming clusters launched by UC Berkeley. Developed for Java/Hadoop ecosystem but with support for Python. PySpark is the Python API for Spark.

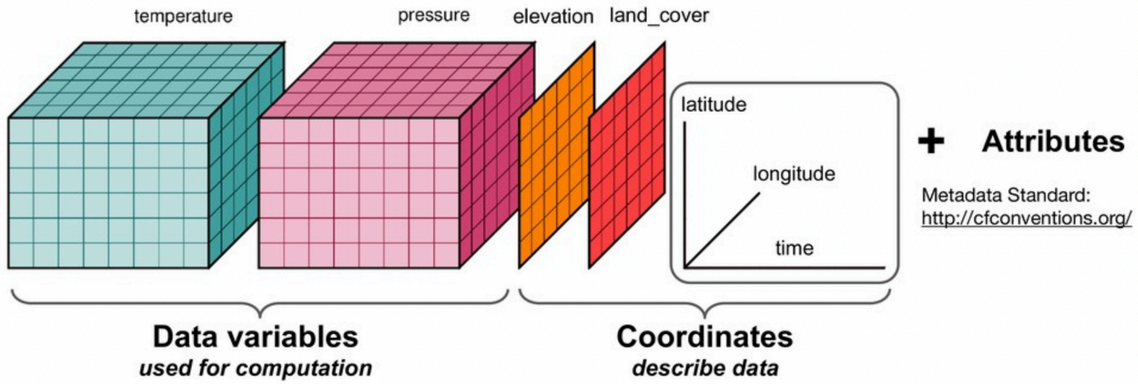
Powered by Dask

These software projects are well-integrated with Dask, or use Dask to power components of their infrastructure.

- **pandas** — Tabular data analysis
- **NumPy** — Array and numerical computing
- **scikit-learn** — Machine learning in Python
- **Scikit-image** — A collection of algorithms for image processing in Python
- **XGBoost** — Gradient boosted trees for machine learning. XGBoost can use Dask to bootstrap itself for distributed training
- **RAPIDS** — GPU Accelerated libraries for data science
- **XArray** — Brings the labeled data power of pandas to the physical sciences, by providing N-dimensional variants of the core pandas data structures
- **Iris** — A Python library for analysing and visualising Earth science data
- **Pangeo** — A community effort for big data geoscience in the cloud
- **Prefect** — A workflow management system, designed for modern infrastructure
- **Napari** — Multi-dimensional image viewer for Python
- **Snorkel** — Programmatically build training data for machine learning
- **Datashader** — Visualization packages for large data
- **Intake** — A lightweight package for finding, investigating, loading and disseminating data
- **TPOT** — A Python Automated Machine Learning tool that optimizes machine learning pipelines using genetic programming
- **MDAnalysis** — A Python toolkit to analyze molecular dynamics trajectories generated by a wide range of popular simulation packages

Scientific-based Python distribution - Anaconda





Machine Learning

Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

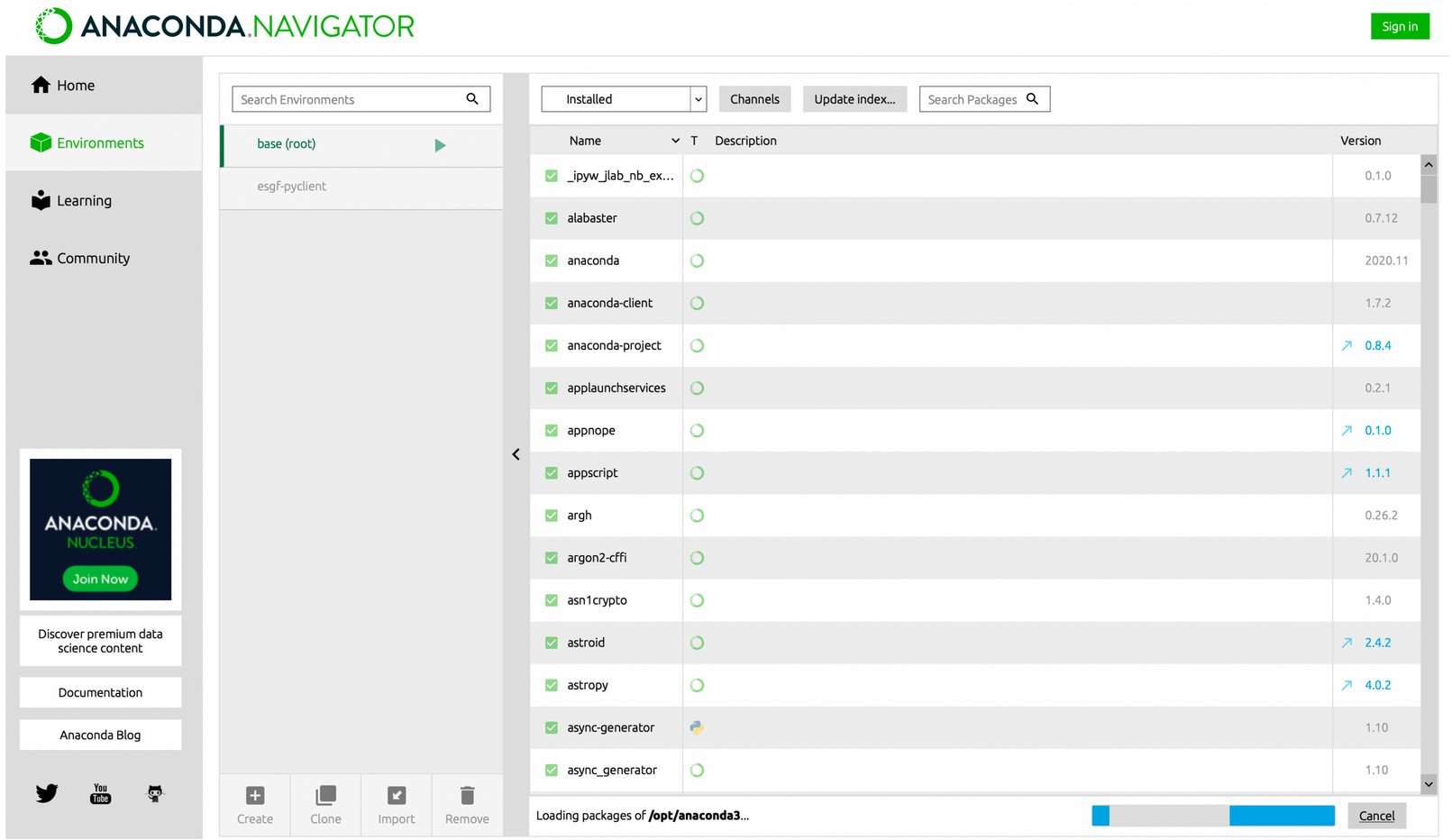
PYTORCH

An open-source deep learning framework using GPUs and CPUs that consists of fundamental tools and libraries for Python AI and machine learning development.

scikit-learn

A powerful and versatile machine learning library for machine learning basics like classification, regression, and clustering. It includes both supervised and unsupervised ML algorithms with important functions like cross-validation and feature extraction. Scikit-learn is the most frequently downloaded machine learning library.

Scientific-based Python distribution - Anaconda



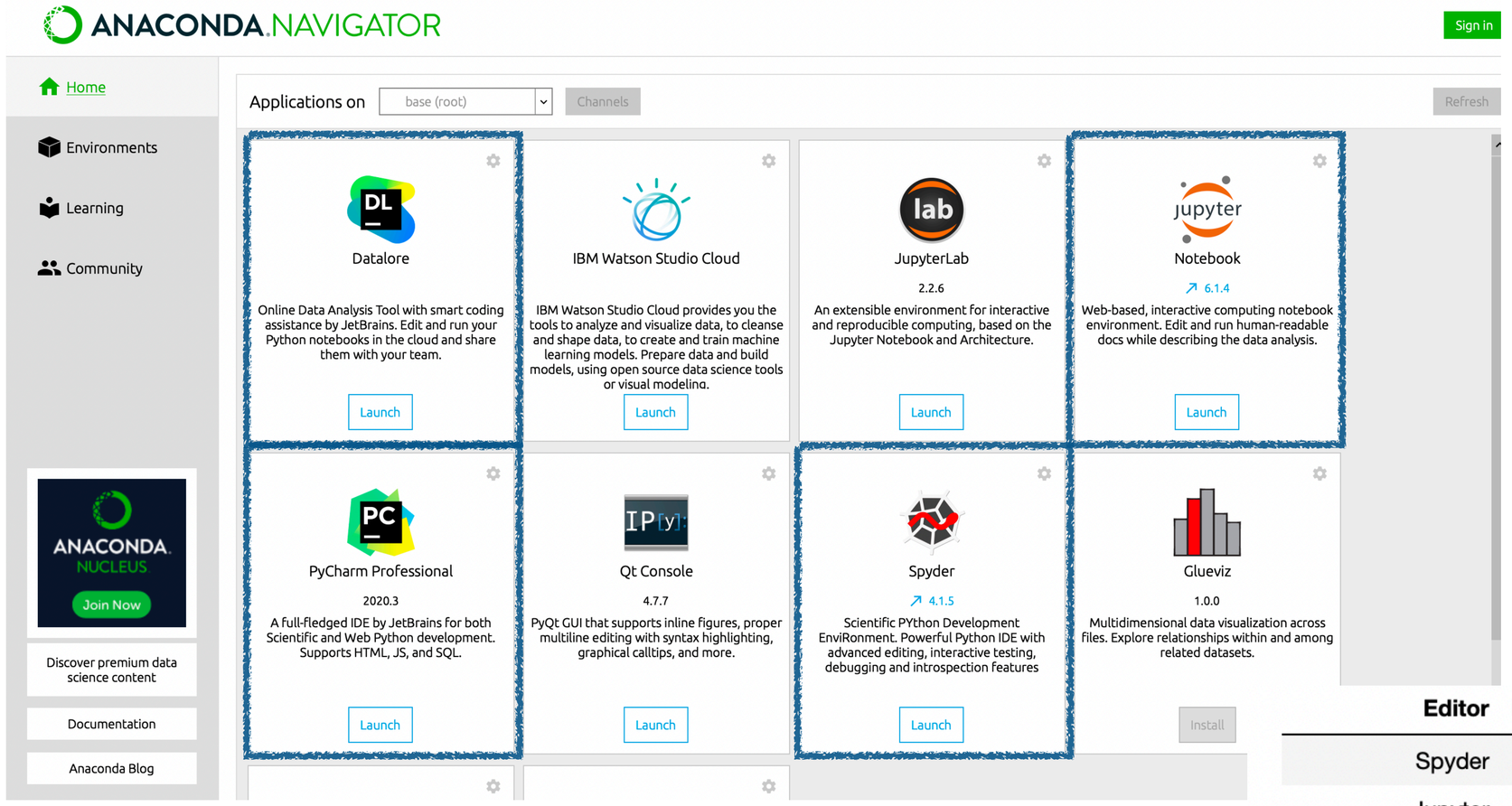
OR

from the terminal:

conda install …

conda install -c conda-forge …

Scientific-based Python distribution - Anaconda

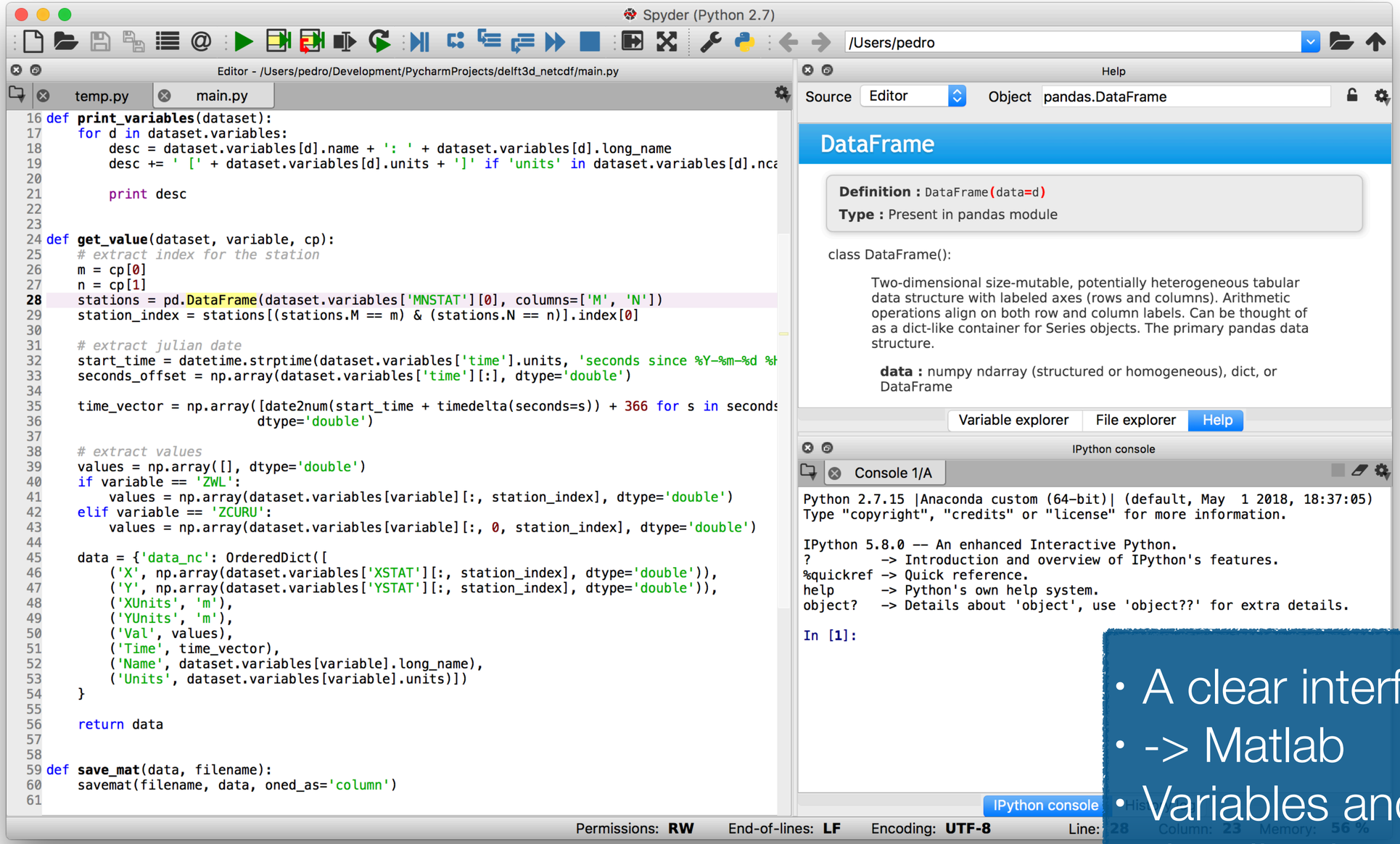


- Notepad++
- Vim
- Sublime Text
- **Spyder**
- Atom
- VS Code
- Pycharm

- Jupyter

...

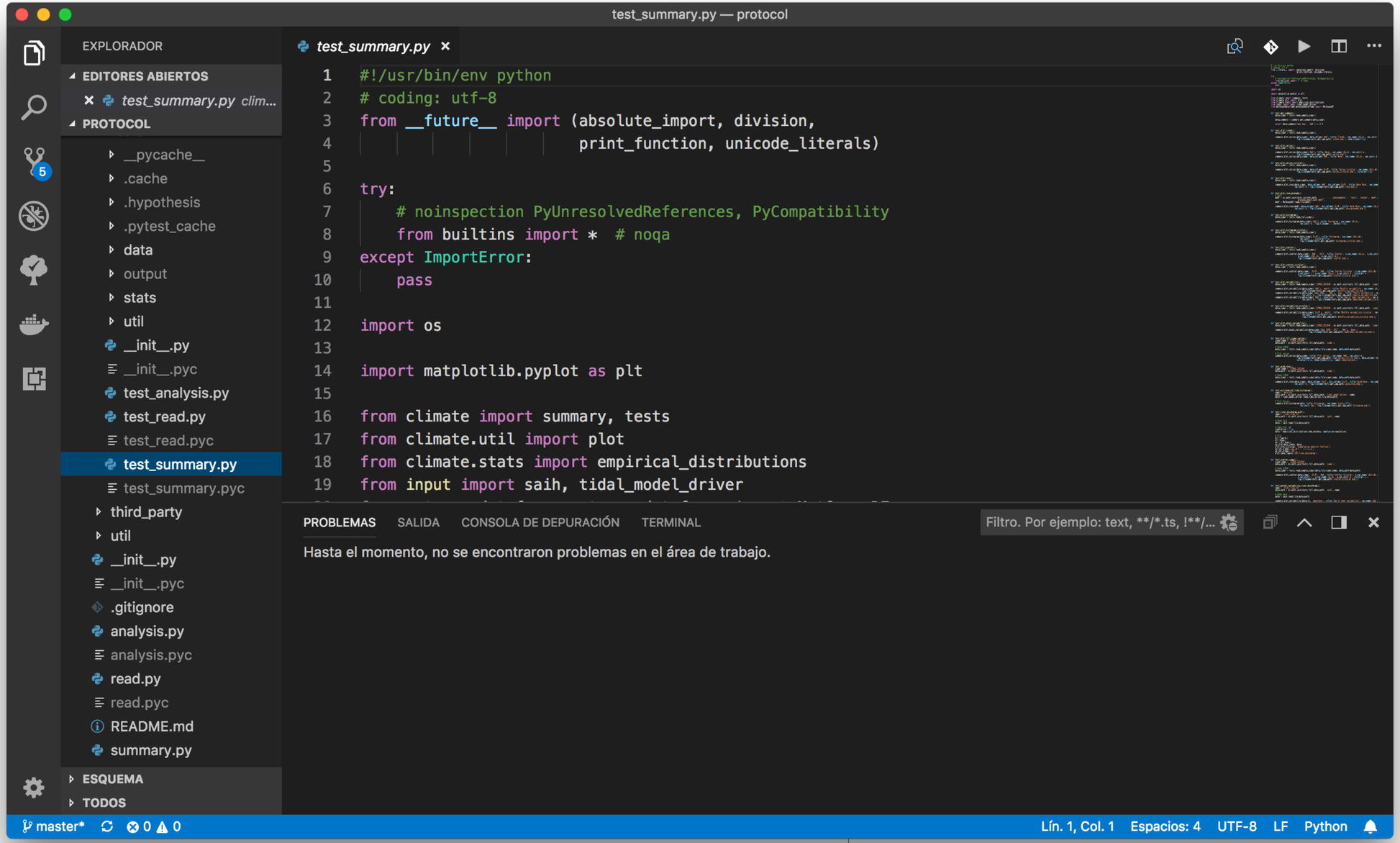
Editor	Learning curve	Users	Benefits
Spyder	pretty short	Matlab and R background	mature, many features
Jupyter	smooth	teachers	interactive
Visual Studio Code	moderate	scientifics / developers	code quality
PyCharm	steep	developers	professional code



- A clear interface
- -> Matlab
- Variables and figures visualization

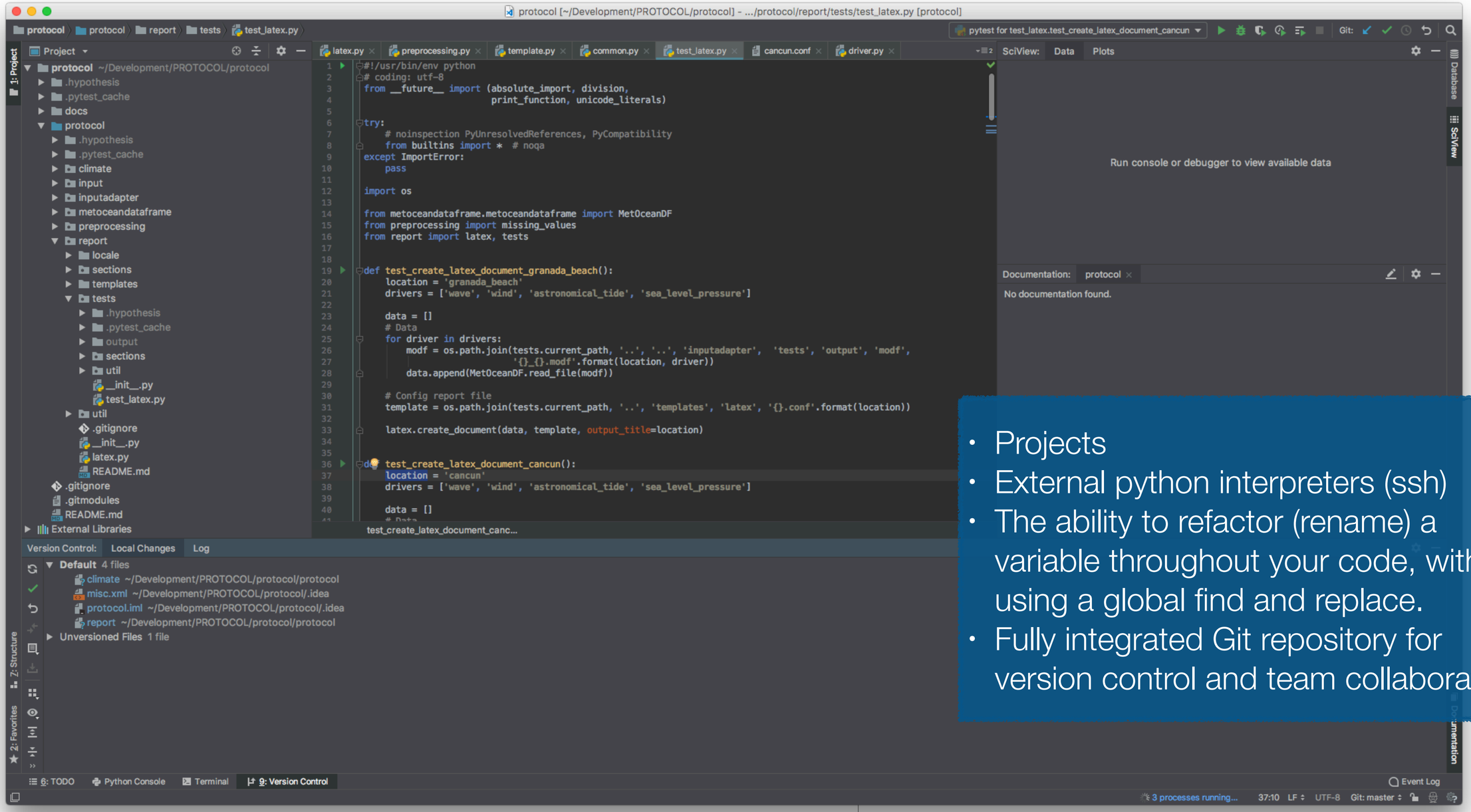
Integrated Development Environment - IDEs

Spyder



Integrated Development Environment - IDEs

VS Code



- Projects
- External python interpreters (ssh)
- The ability to refactor (rename) a variable throughout your code, without using a global find and replace.
- Fully integrated Git repository for version control and team collaboration.

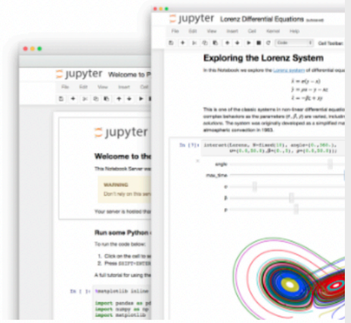
Integrated Development Environment - IDEs

Pycharm

jupyter INSTALL PROJECT DOCUMENTATION BLOG DONATE

Jupyter Notebook

The Jupyter Notebook is a web-based interactive computing platform that allows users to author data- and code-driven narratives that combine live code, equations, narrative text, visualizations, interactive dashboards and other media.



Run | Markdown | Edit Presentation | Show Presentation

In [57]:
Slide Type: -

```
from sympy import diff, sin, exp

diff(sin(x)*exp(x), x)
```

Out[57]: $e^x \sin(x) + e^x \cos(x)$

Slide Type: Fragment

Compute $\int (e^x \sin(x) + e^x \cos(x))\, dx$

In [58]:
Slide Type: -

```
from sympy import integrate, cos

integrate(exp(x) * sin(x) + exp(x) * cos(x), x)
```

Out[58]: $e^x \sin(x)$

Slide Type: Sub-Slide

Compute $\int_{-\infty}^{\infty} \sin(x^2)\, dx$

In [59]:
Slide Type: -

```
from sympy import oo

integrate(sin(x**2), (x, -oo, oo))
```

Out[59]: $\frac{\sqrt{2}\sqrt{\pi}}{2}$

Integrated Development Environment - IDEs | Jupyter

Cloud-based software

- Binder
- Kaggle Kernels
- Google Colaboratory (Colab)
- Microsoft Azure Notebooks -> Github Codespaces (beta)*
- CoCalc
- Datalore

https://www.dataschool.io/cloud-services-for-jupyter-notebook/

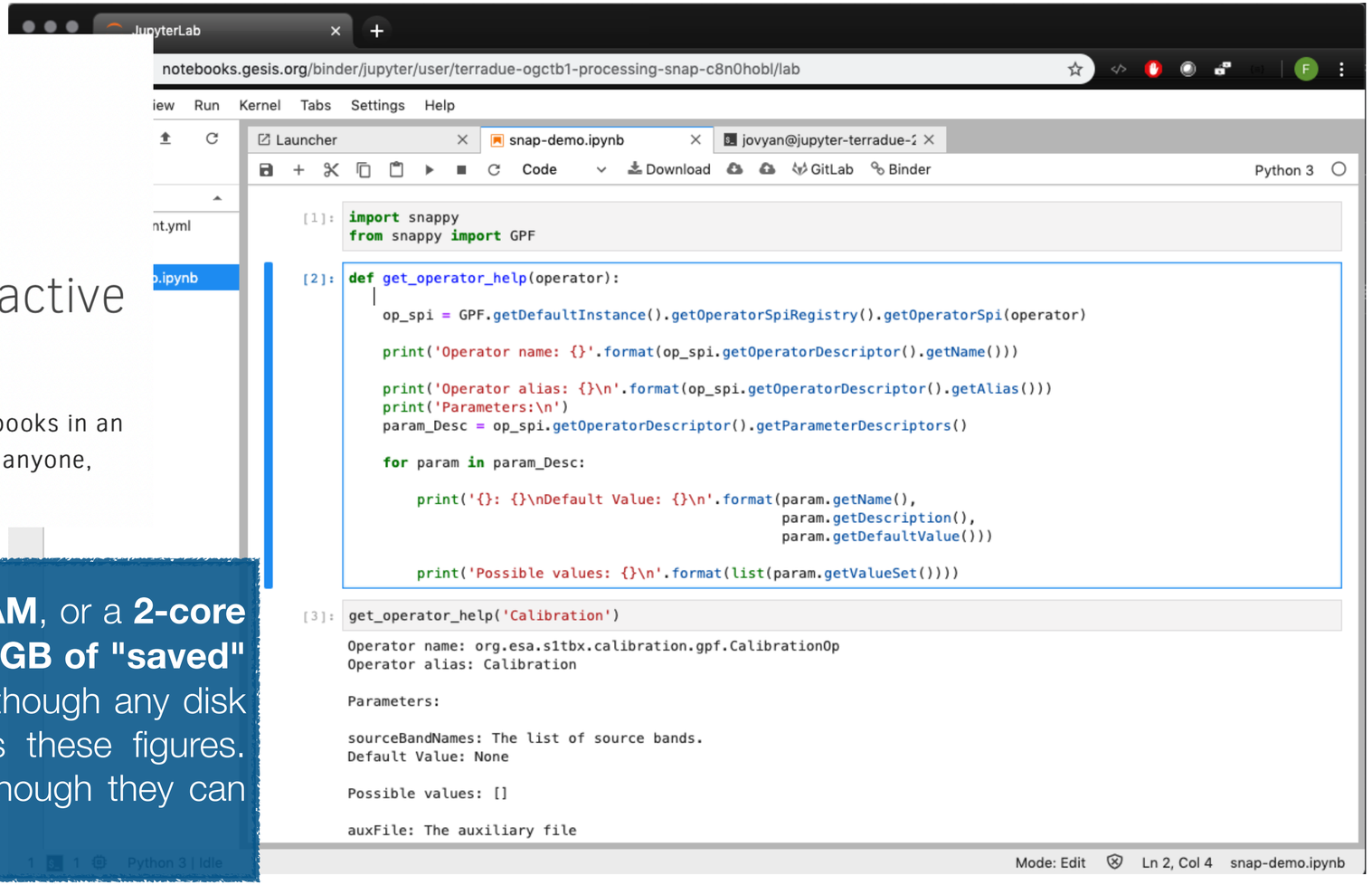
https://docs.google.com/spreadsheets/d/12thaaXg1Idr3iWST8QyASNDs08sjdPd6m9mbCGtHFn0/edit#gid=1505836451

Cloud-based software | Binder

binder

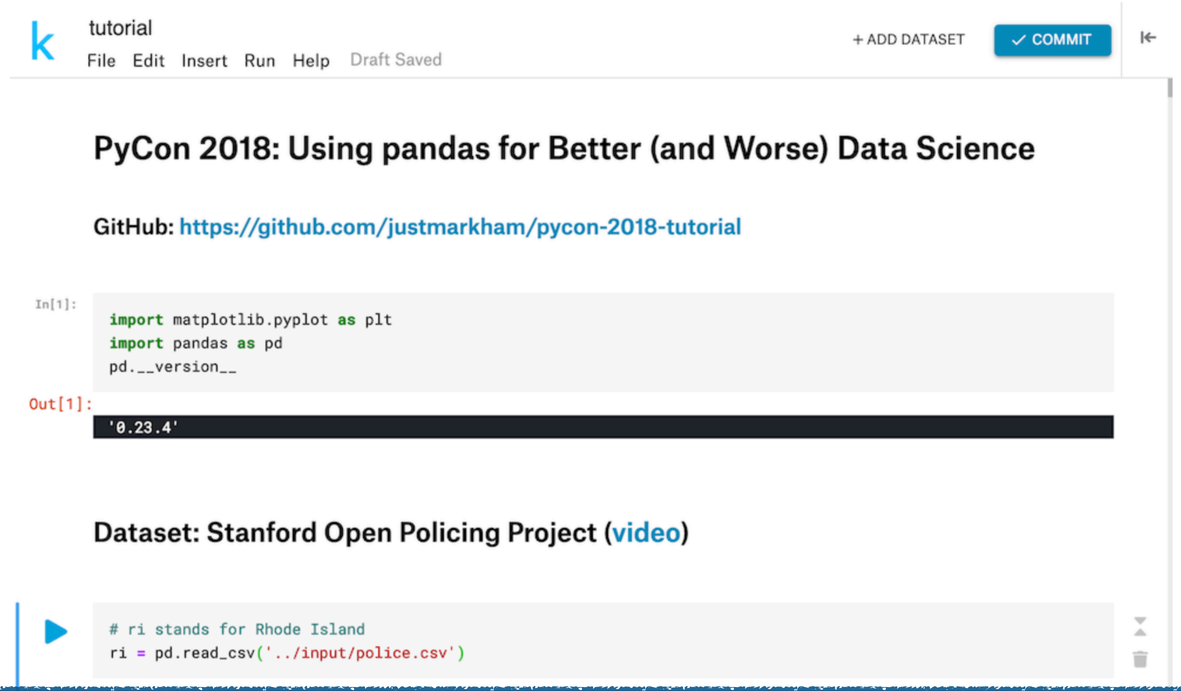
Turn a Git repo into a collection of interactive notebooks

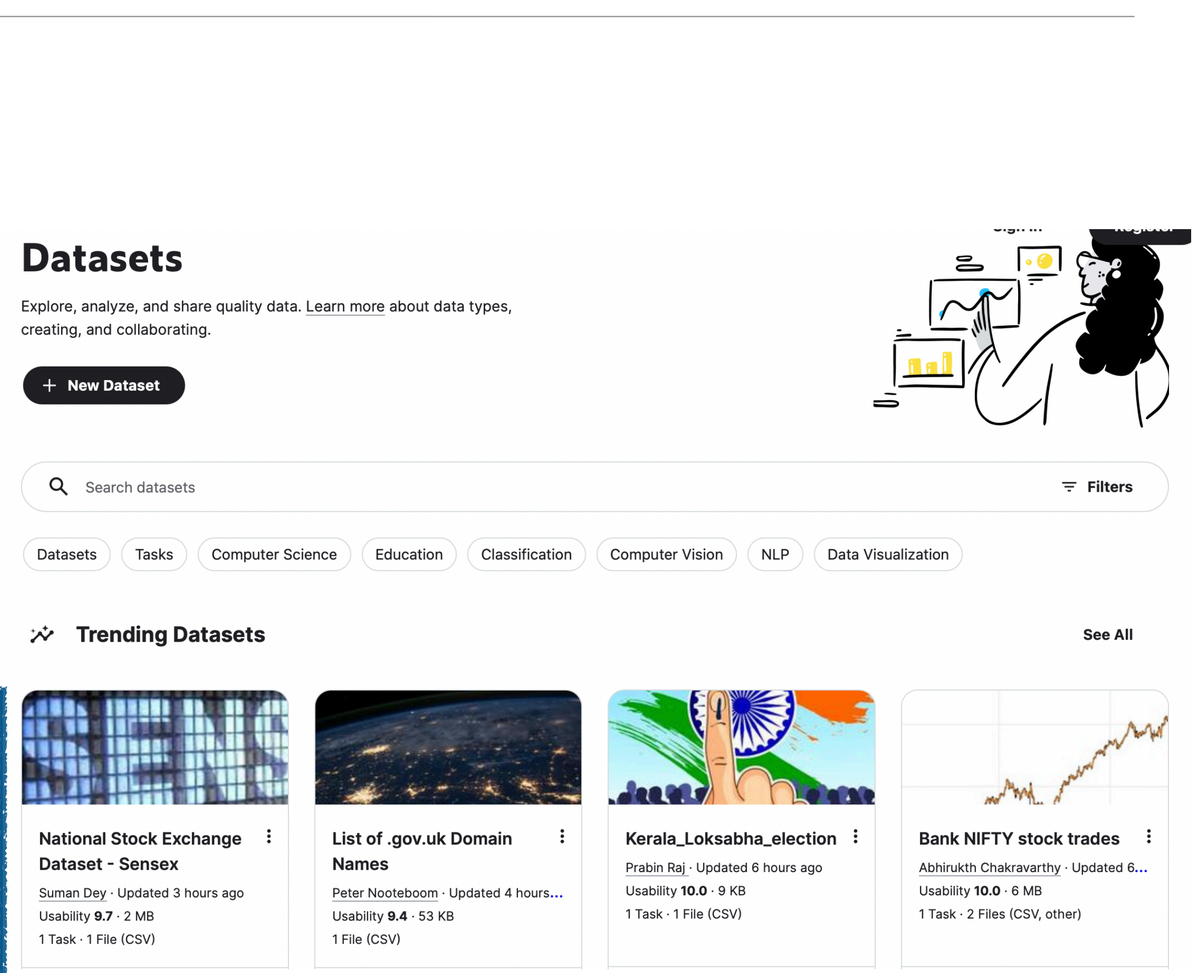
Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.



You can access either a **4-core CPU with 17 GB of RAM**, or a **2-core CPU with 14 GB of RAM plus a GPU**. You will have **5 GB of "saved" disk space and 17 GB of "temporary" disk space**, though any disk space used by your dataset does not count towards these figures. Sessions will shut down after 60 minutes of inactivity, though they can run for up to 9 hours.

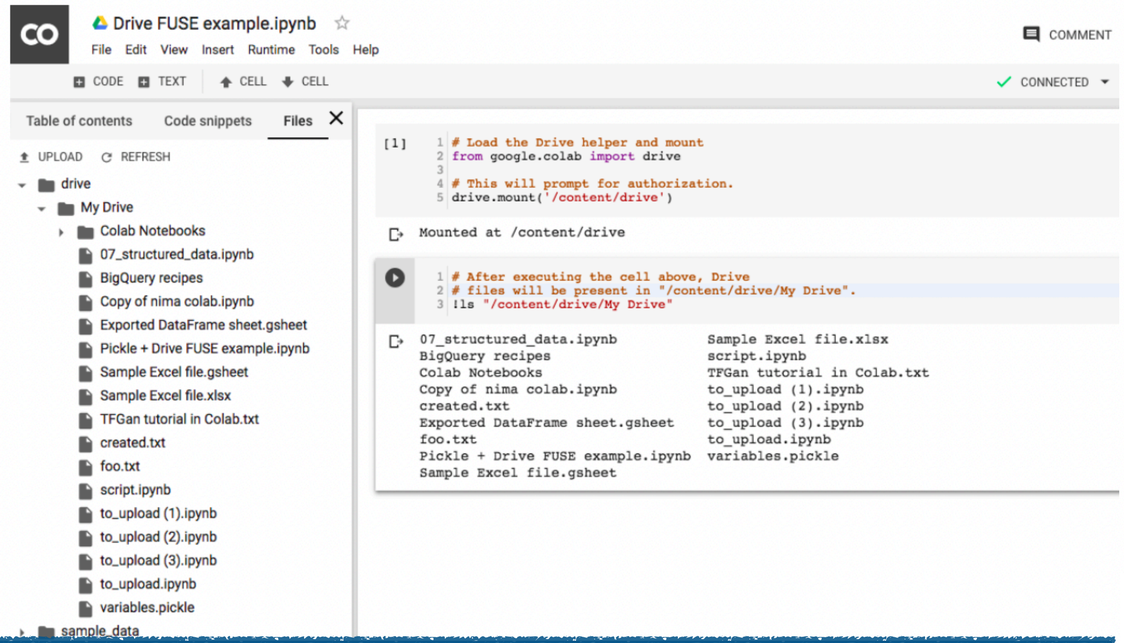
Cloud-based software | Kaggle





ou can access either a **4-core CPU with 17 GB of RAM**, or a **2-core CPU with 14 GB of RAM plus a GPU**. You will have **5 GB of "saved" disk space and 17 GB of "temporary" disk space,** though any disk space used by your dataset does not count towards these figures. Sessions will shut down after 60 minutes of inactivity, though they can run for up to 9 hours.

Cloud-based software | Google Colab



Colab does give you access to a GPU or a TPU. Up to 15 GB of disk space for storing your datasets. Sessions will shut down after 60 minutes of inactivity, though they can run for up to 12 hours.

Presently they provide ***2-core VMs with 12 Gb of RAM****, and either* ***50 Gb (non-GPU kernel) or 350 Gb (GPU kernel) of disk space****.*

Earth Engine

Earth Engine's public data archive includes more than forty years of historical imagery and scientific datasets, updated daily and available for online analysis.

ClimaCell - CBAM India Weather Forecasts
Climacell
High-resolution weather forecast model data serving India

Cloud-to-Ground Lightning Strikes
NOAA
Aggregated lightning strike data from 1987 to 2018

CMIP 6
NOAA
Climate Model Data from the World Climate Research Programme

GHCN Daily
NOAA
Global Historical Climatology Network Daily Weather Data

GHCN Monthly
NOAA
Global Historical Climatology Network Monthly Weather Data

Global Forecast System
NOAA
The National Weather Service's Primary Prediction Model

Global Historical Tsunami Database
NOAA
Tsunami and source event data as far back as 2000 BC

Global Hurricane Tracks (IBTrACS)
NOAA
Tropical Cyclone/Hurricane position and intensity

NOAA Rapid Refresh (RAP)
NOAA
Rapid Refresh operational weather prediction system data.

OnPoint Weather - Past Weather and Climatology Dat...
Weather Source
Past, Present and Forecast Global Weather Data

OnPoint Weather - Temperature History &...
Weather Source
Past, Present and Forecast Global Weather Data

Preliminary Storm Reports
NOAA
Preliminary storm reports from NWS's Storm Prediction Center

Project Sunroof
Google Project Sunroof
Find out how much sunlight hits your roof in a year

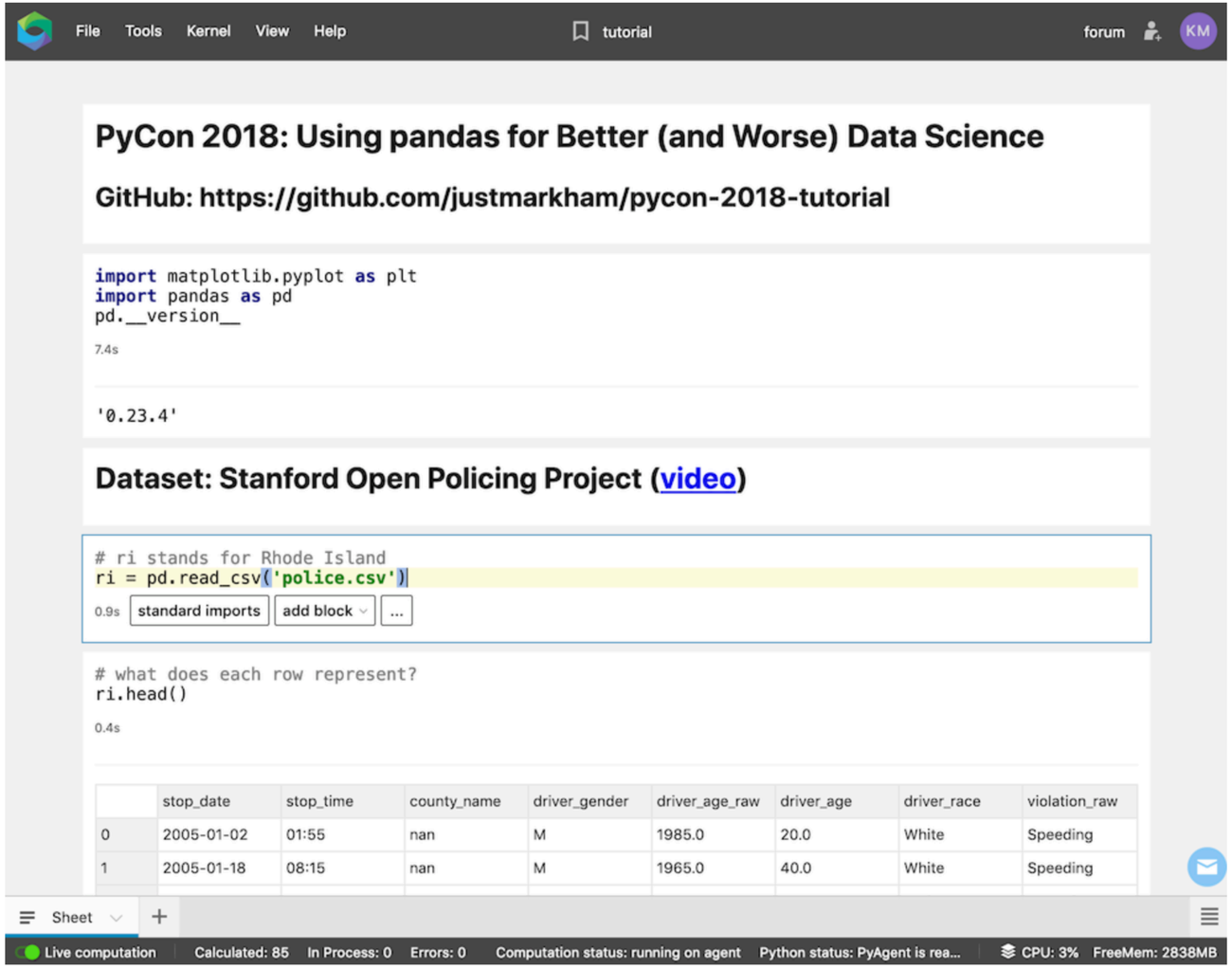
Real-time Air Quality
OpenAQ
Real-time air quality data from around the world

Sentinel-2
ESA
Comprehensive land, ocean, and atmosphere observation data

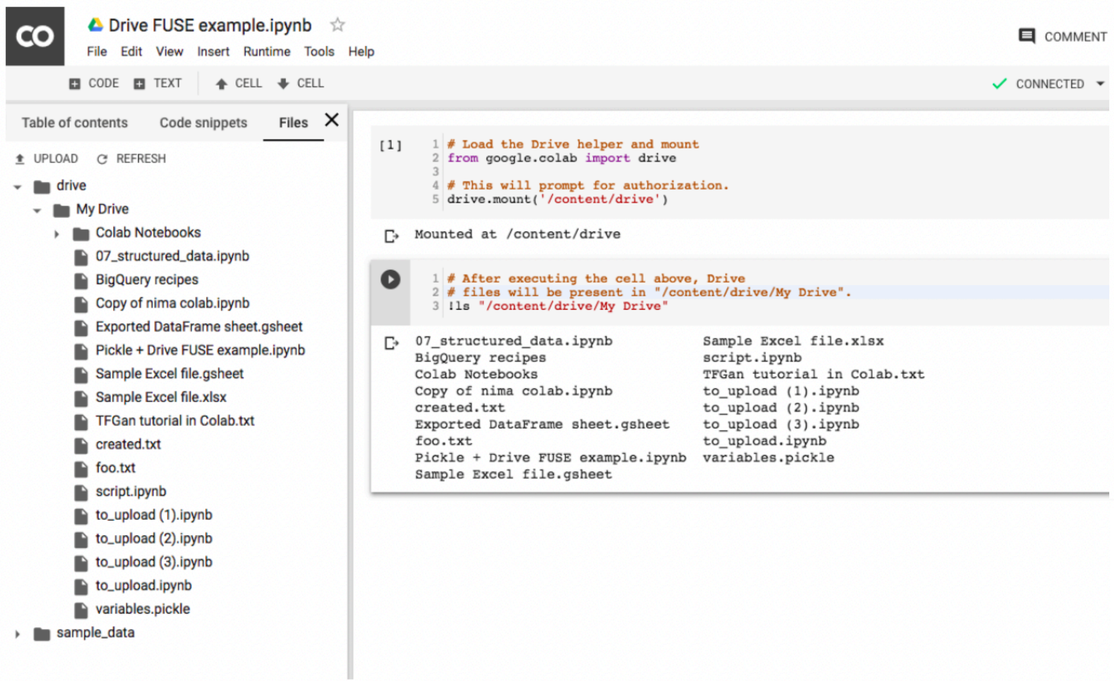
Severe Storm Event Details
NOAA
Detailed data about severe weather between 1950 and this year

Cloud-based software | Datalore

You will have access to a **2-core CPU with 4 GB of RAM,** and **10 GB of disk space**. Sessions will shut down after 60 minutes of inactivity, though there is no specific limit on the length of individual sessions. You can use the service for up to 120 hours per month.



Cloud-based software | Google Colab



Earth Engine

Earth Engine's public data archive includes more than forty years of historical imagery and scientific datasets, updated daily and available for online analysis.

climacell

ClimaCell - CBAM India Weather Forecasts
Climacell

High-resolution weather forecast model data serving India

Cloud-to-Ground Lightning Strikes
NOAA

Aggregated lightning strike data from 1987 to 2018

CMIP 6
NOAA

Climate Model Data from the World Climate Research Programme

GHCN Daily
NOAA

Global Historical Climatology Network Daily Weather Data

GHCN Monthly
NOAA

Global Historical Climatology Network Monthly Weather Data

Global Forecast System
NOAA

The National Weather Service's Primary Prediction Model

Global Historical Tsunami Database
NOAA

Tsunami and source event data as far back as 2000 BC

Global Hurricane Tracks (IBTrACS)
NOAA

Tropical Cyclone/Hurricane position and intensity

NOAA Rapid Refresh (RAP)
NOAA

Rapid Refresh operational weather prediction system data.

OnPoint Weather - Past Weather and Climatology Dat...
Weather Source

Past, Present and Forecast Global Weather Data

OnPoint Weather - Temperature History &...
Weather Source

Past, Present and Forecast Global Weather Data

Preliminary Storm Reports
NOAA

Preliminary storm reports from NWS's Storm Prediction Center

Project Sunroof
Google Project Sunroof

Find out how much sunlight hits your roof in a year

Real-time Air Quality
OpenAQ

Real-time air quality data from around the world

Sentinel-2
ESA

Comprehensive land, ocean, and atmosphere observation data

Severe Storm Event Details
NOAA

Detailed data about severe weather between 1950 and this year

Course docs

- https://github.com/andreall/python_course-dicca.git
- https://mybinder.org/v2/gh/andreall/python_course-dicca.git/HEAD

Additional resources

- https://projectpythia.org
- https://dabeaz-course.github.io/practical-python/Notes/Contents.html