ENVRI COMMUNITY
The Community of Environmental Research Infrastructures

e-Infrastructures for advanced research

Background material for the ENVRIplus Scientific Game Project

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 654182.

The project is currently coordinated by University of Helsinki. The coordination will switch to ICOS ERIC once it is founded

Authors

The ENVRIplus Scientific Gaming will motivate secondary school students by replicating the excitement of scientific research. The ENVRIplus Scientific Gaming is developed in the context of the ENVRIplus Project is a Horizon 2020 project bringing together Environmental and Earth System Research Infrastructures, projects and networks together with technical specialist partners to create a more coherent, interdisciplinary and interoperable cluster of Environmental Research Infrastructures across Europe.

Environmental Research Infrastructures provide key tools and instruments for the researchers to address specific challenges within their own scientific fields. However, to tackle the grand challenges facing human society (for example climate change, extreme events, loss of biodiversity, etc.), scientific collaboration across the traditional fields is necessary. More information about the ENVRIplus project you can find: http://envri.eu/

Table of Contents

Background material e-Infrastructures for advanced research

Are you interested in becoming a young researcher? Are you going to using advanced computing technology to resolve scientific challenges? Join us – we will get there together.
The Scientific Game focuses on the topic of the e-Infrastructure, a collaboration computing environment to address the needs of scientists.

Keywords: The evolution of computing machines, e-Infrastructures, EGI, the EGI Federated Cloud, Jupyter Notebook.

The History of Computing

Human beings have always been seekers of knowledge. The minute we discover something new, we want to share it with others and move onto the next achievement. Since the beginning of recorded history, and probably before, we have always strived to discover the mysteries of the planet, of Earth and of ourselves. How has learning evolved over the course of human history and what might the future hold for us ?

In this continuous search for replies, science and technology have played a key role to help human beings to answer these questions. This link, between science and technology, is more evidence starting from the end of the 19th century, when scientists began to use technological devices. These devices allowed them to investigate phenomena that could not be experienced by the human senses alone.

If we look back, since the origin of human beings, devices have been adopted to aid computations for thousands of years, mostly using one-to-one correspondence with fingers. Probably we will never know when or how humans first developed the ability to count. The process does not leave any physical evidence behind the archaeologists to find. What we do know is that the process is extremely ancient. Any of the so-called primitive peoples that have been studied have all had a highly developed sense of number and, to at least some degree, an ability to represent numbers in both words and symbols. Once humans had developed the ability to count, it must have become necessary to have a method of recording numbers.

The Abacus

The **Abacus**, also called a counting frame, is usually considered the calculating tool used primarily in parts of Asia for performing arithmetic processes.

What we now call the Roman abacus was used in Babylonia as early as 2400 BC.

Its original style of usage was by lines drawn in sand with pebbles. Today, abacuses are often constructed as a bamboo frame with beads sliding on wires, but originally they were beams or stones moved in grooves in sand or on tablets of wood, stone, or metal.

A reconstruction of a Roman hand abacus, made by the RGZ Museum in Mainz, 1977. The original is bronze and is held by the Bibliothèque nationale de France, in Paris.

The first analog computer

Analog computers are believed to have been first developed by the Greeks with the **Antikythera mechanism**, which was designed to calculate astronomical positions.

The Antikythera mechanism was discovered in the Antikythera wreck off the Greek island of Antikythera, between Kythera and Crete, in 1901. Subsequent investigation, particularly in 2006, dated it to about 150–100 BC; and hypothesised that it was on board a ship that sank in route from the Greek island of Rhodes to Rome.

Technological artefacts of similar complexity did not reappear until a thousand years later.

The Antikythera mechanism, designed to calculate astronomical positions (early 1st century BCE)

The development devices and first calculators

Though the different analogue computers were able to perform some of the useful arithmetic calculus, the story of devices that ultimately led to fully automatic computation really starts with the introduction of the development devices to perform the four standard arithmetic functions. By devising a system in which mechanical levers, gears, and wheels could replace the facilities of human intellect, the early pioneers in these devices showed the way towards the complete automation of the process of calculation.

The first calculator or adding machine to be produced in any quantity and actually used was the Pascaline, designed and built by the French mathematician-philosopher Blaise Pascal between 1642 and 1644. Pascal invented the machine for his father, a tax collector. The machine could add, subtract, multiply and divide two numbers being entered by manipulating its dials.

The Pascaline or Pascal's Calculator, by Blaise Pascal. It could add, subtract, multiply and divide two numbers (1642)

From the Pascal's Calculator to the first IBM personal computer (1981), calculating devices have come a long way. In the last 30 years or so, scientific computing has steadily evolved from mainframe-based centralized solutions to a really distributed environment.

Many of the great scientific discoveries of the past were made by single researchers working alone or in small groups (e.g. the discovery of oxygen by Joseph Priestley, Charles Darwin's theory of evolution, or Watson & Crick's seminal work on the structure of the DNA molecule).

Nowadays, the scientific landscape is changing and the age of the individual is quickly giving way to a new era of collaborations. Modern research breakthroughs addressing the grand challenges of the future call for large scale, cross-country and multi-disciplinary collaborations, involving dozens, some-times hundreds, of highly-qualified scientists working for a common goal (e.g. the genome projects, the High-Energy Physics experiments at CERN, or the effort to build a viable fusion reactor at ITER).

This has been also possible thanks to the concurrent availability of powerful "Commercial Off The Shelf" (COTS) computers and decrease of costs of Local Area Networks. In the first half of 90's the emergence of cluster computing for High Throughput Computing (HTC) applications was confirmed and farms" of computers with many-core processors, interconnected by very low latency networks, have become the norm also in the domain of High Performance Computing (HPC) at a point that in the last five years about 80% of the Top500 machines are based on a distributed architecture.

Furthermore, the steep decrease of costs of large/huge-bandwidth Wide Area Net-works has fostered in the recent years the spread and the uptake of the Grid Computing paradigm and the distributed computing ecosystem has become even more complex with the recent emergence of Cloud Computing.

Collaborative discovery will continue to evolve as a new research paradigm, propelled by new instruments capable of acquiring vast amounts of data and the sophisticated simulations needed to interpret it. 'Big Science' will be the basis of future breakthroughs, but relies heavily on computing power to store, process and analyse the data deluge. This relationship can catalyse scientific discovery, opening new possibilities, uncharted avenues of research and new knowledge transfer routes. But it could also hamper innovation, if the increase in computing capability lags behind the needs of the researchers.

E-Infrastructures

Recognising this trend, the European Commission has unveiled Europe 2020 – a strategy to harness the opportunities promised by the evolving scientific landscape and pave the way to a smart, sustainable and inclusive economy. One of the driving forces behind this strategy is the European Research Area (ERA) – a transnational effort combining research centres with on-going programmes and projects aimed at building multidisciplinary collaborations and enabling rapid knowledge transfer across borders.

The full implementation of the digital ERA depends heavily on the development of the e-Infrastructures that will enable the accomplishment of the 'fifth freedom' – free circulation of researchers, knowledge and technology across Europe. The European Commission is heavily investing through its Framework Programme in e-Infrastructures and these new platforms are by now considered as key enablers of the ERA.

What is an e-Infrastructure?

E-Infrastructure is a term that is mainly used in the research and development context. It designates the new generation of Information and Communication Technology (ICT)-based infrastructures. These new generation of infrastructures exploit several components and layers, such as networks, supercomputers and other computing resources, storage, remote resources and instrumentation e.g. sensors. Such elements are seamlessly inter-connected and can be access by users all around the world, regardless of their geographical location.

These e-Infrastructures allow scientists to share information securely, analyse data efficiently and collaborate with colleagues worldwide. They are an essential part of modern scientific research and a driver for economic growth. Together with other pan-European e-Infrastructures such as GÉANT (for networking) and PRACE (for supercomputing), EGI is a key enabler of the digital ERA by linking thousands of researchers across Europe to the resources they need.

The EGI e-Infrastructure

EGI is an e-Infrastructure for science, set up as a federation of resource providers to connect researchers from all disciplines with the reliable and innovative computing services they need to manipulate and interpret data, share information and collaborate with colleagues worldwide. The research supported by EGI covers areas such as the Large Hadron Collider (LHC) particle accelerator at CERN attempting to find the Higgs boson, environmental modelling to support studies on climate change, medical researchers finding innovative cures for diseases such as Alzheimer's or malaria or earthquake simulations to understand the impact of seismic waves propagations.

Many of these researches have achieved significant results. For example, the discovery of the Higgs particle by LHC experiments at CERN confirmed the theoretical discovery of a mechanism that contributes to our understanding of the origin of mass of subatomic particles, resulting in winning the Nobel price in 2013. As Rolf Heuer, the former Director-General of CERN, said, "Higgs discovery only possible because of the extraordinary achievements of Grid computing".

EGI can trace its origins back to the early 2000's – the pioneering days of scientific distributed computing, when the grid infrastructure emerged from the coalescence of local services to cater for compute- and data-oriented applied researchers who wanted to share services with their collaborators.

Over the years, thousands of scientists have integrated innovative technology from Europe and beyond into their daily activities. Today, EGI supports over 48,000 researchers from 15 different disciplines in their intensive data analysis research by running more than 1.7 Million jobs/day. This is an important result for EGI because it means that EGI is more and more user driven and the solutions provided by EGI are really used to support the day-by-day work of scientific communities and research infrastructures.

From a technical point of view, the EGI federation is coordinated by the EGI Foundation (also known as EGI.eu), a not-for-profit foundation established in February 2010 under the Dutch law and based in Amsterdam. The overall objective of this foundation is to manage the EGI federation on behalf of its participants: National Grid Initiatives (NGIs) and European Intergovernmental Research Organisations (EIROs). The EGI Foundation is governed by a Council of representatives from participant countries and institutions.

EGI members – national compute/data centers (so called NGIs) and Intergovernmental Research Organizations, such as CERN – operate the compute, storage, application and software services that comprise the 'EGI infrastructure'. These compute/storage providers allocate resources to scientific communities via 'Virtual Organisations' (VOs), according to institutional or regional/national priorities.

A VO is the online representation of a scientific user group whose members are usually work in similar or related research areas, or are part of the same scientific collaboration, for which reason they need the same applications, software and underlying hardware capabilities. Some of the biggest VOs of EGI represent experiments of the Large Hadron Collider (ALICE, ATLAS, CMS, LHCb), the VIRGO experiment, the Cerenkov Telescope Array Observatory or life science researchers from multiple countries and diverse background (biomed VO).

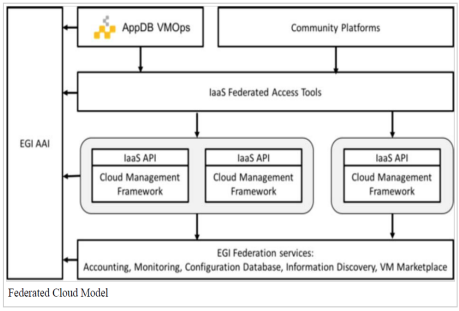
Over the last decade this e-infrastructure was the enabler of digital research conducted by researchers all over the world through the whole spectrum of science from High-Energy Physics, to Earth Sciences, Life Sciences, Chemistry, Astrophysics, and Humanities.

Values Proposition of e-Infrastructures

1.) Remote access to computing services, instrumentation and resources in general, creates new opportunities for researchers to bring existing applications to higher levels of usability and performance.
2.) Additionally, it enables researchers to deploy new strategies in approaching scientific problems with simulation tools and intensive applications.
3.) Stimulate the creation of new scientific communities, uniting researchers who are working on similar challenges and are willing to share resources and reach new levels of collaboration.
4.) Get access to scientific data and unique instruments located in top level laboratories around the world without the need to travel.

The EGI Federated Cloud

The EGI Federated Cloud is a network of academic private clouds built around open standards to target the requirements of the scientific communities. The original architecture was put into production in May 2014. The EGI community has refined the initial concept and evolved its architecture according to emerging user demands. As shown in the figure below, the architecture is based on the concept of an abstract Cloud Management Framework (CMF) that supports a set of cloud interfaces to communities. Each resource centre of the infrastructure operates an instance of this CMF according to its own technology preferences and integrates it with the federation by interacting with EGI core components. EGI does not mandate deploying any particular or specific Cloud Management Framework; it is the responsibility of the Resource Center to investigate, identify and deploy the solution that fits best their individual needs whilst ensuring that the offered services implement the required interfaces and domain languages of the federation.



Federated Cloud Model

Users can interact with cloud providers of the EGI Federated Cloud in several ways:

- Directly using the APIs of the resource centres to manage individual resources.
- Leveraging federated IaaS provisioning tools that allow managing and combining resources from different providers and enable the portability of application deployments between them. The EGI Federated Cloud task force evaluates and documents the best tools for this task at the EGI wiki [R1].
- Using the EGI AppDB VMOps dashboard [R2], a web-based GUI that simplifies the management of VMs on any provider of the EGI infrastructure.

How can the EGI Federated Cloud help scientists?

As we mentioned before, the EGI Federated Cloud is a federation of institutional private clouds, offering cloud services to scientists in Europe and worldwide.

If you are **a scientist, a member of a research project or a service provider**, the EGI Federated Cloud can provide you with a flexible computing and storage environment to run your applications and services.

This federation provides a flexible environment where end-users can use virtual resources that can scale dynamically upon request. Some examples:

The genetics of Salmonella infections

Konrad Förstner, a bioinformatician working at the University of Würzburg in Germany, and his colleagues used Cloud to run READemption, a pipeline for the computational analysis of RNA sequencing data, and analyse the RNAs produced by the *Salmonella*.
First, the team infected human HeLa cells with the type of *Salmonella* that causes salmonellosis. Then they analysed the combined RNA from the both organisms by using so called high-throughput sequencing.
Analysing dual RNA-Seq data is complex and time consuming. To streamline all the work, Förstner ran the READemption tool with Cloud to process most of the computational tasks.

Colour-enhanced scanning electron micrograph showing Salmonella (in red) invading cultured human cells Credit: Rocky Mountain Laboratories, NIAID, NIH (Wikimedia Commons)

Cloud helped Förstner and his team to handle computational demand peaks when new data sets arrived and that speed up the whole process significantly. Thanks to this novel approach, the research team found that a piece of *Salmonella* RNA called PinT is heavily involved in what happens right after the infection. In Förstner's words, PinT "fine-tunes the transition from the infection stage, which requires a large set of genes, to the survival stage, that needs a different set of genes."

When PinT activity was shut down in a follow up experiment, the team saw the changes in the bacteria and in the host cell as well. This shows that PinT also influences what happens in the host and is "of high relevance for understanding the infection process as well as the immune response," Förstner concludes.

Their results were published in *Nature (doi:10.1038/nature16547)* and show that a small piece of RNA has a say on both the infection process and the immune response.

> "The EGI Federated Cloud accelerated our analysis dramatically and helped us to focus on scientific solutions and results, instead of resource management and system administration."
>
> ***K. Förstner, University of Würzburg***

New viruses implicated in fatal snack disease

Boid Inclusion Body Disease is a threat to boas and pythons in homes and zoos around the world. The disease starts with lack of appetite and poor digestion, but the main tell-tale sign are the many protein inclusions that appear in all cell types. The disease is often fatal.

Besides the trouble it brings to conservation projects, the disease is also an economic burden – imposing strict quarantine rules for the transport of 200,000 snakes a year does not come cheap.

So far scientists know that the disease is somehow related to viruses from the arenavirus family, which are usually carried by mice and may cause haemorrhagic fever or meningitis if transmitted to humans.

Jussi Hepojoki, a virologist based at the University of Helsinki in Finland, and his team set out to sequence the genome of the viruses found in the samples of six snakes. The first step was to isolate the RNA of the viruses and sequence the samples using Next Generation Sequencing (NGS) techniques.

Then, Hepojoki used the Chipster platform to handle the millions of sequences generated by the NGS analysis. To assemble the genomes, Hepojoki ran the MIRA software using the computing resources provided by CSC – who represents Finland in the EGI Council.

Feeling poorly? Boid Inclusion Body Disease affects many captive snakes across the world. Image: Jakub Halun/wikicommons

Jupyter Notebook for Data Science

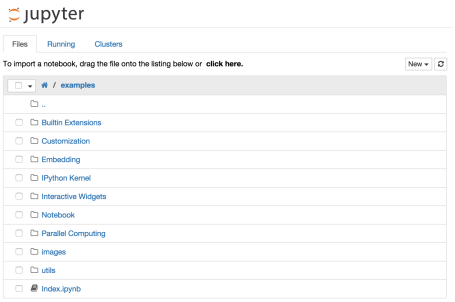
Nowadays different tools and platforms are available for supporting the day-by-day work of modern researchers and provide a transparent access to the distributed infrastructure. In the past few years, one of the most prominent solution is represented by Jupyter.

Jupyter [R3] is an open-source interactive platform for Data Science introduced for the first time in 2011.
Jupyter provides a shell console over the web, a text editor in the browser and a support for more than 40 different programming languages. For this reason, this is the favorite tool adopted for the Software and Data Carpentry workshops.

As a fun note, "Jupyter" is a loose acronym meaning Julia, Python, and R. These programming languages were the first target languages of the Jupyter application, but nowadays, the notebook technology also supports many other languages.

How to install Jupyter Notebook?

The official step-by-step guide along with some references is available here.
This is the Jupyter Notebook front-page when it is running:



There are three main tabs **Files**, **Running** and **Clusters**. You'll be mostly using the first two (when not in the actual notebook):

- **Files**: is the listing of your current working directory. When you first launch the notebook, the directory is the same where you launched the app.
- **Running**: is a list of all active notebooks, i.e. notebooks that have been running commands through one of the available kernels.
- **Clusters**: this is a listing of all clusters that are available for a back-end execution (will be empty, unless you have connected the Jupyter Notebook app to a cluster)

Overview of the Notebook

If you create a new notebook or open an existing one, you will be taken to the notebook user interface (UI). This UI allows you to run code and author notebook documents interactively.

The notebook UI has the following main areas:

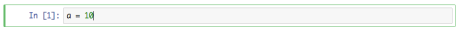
- Menu
- Toolbar
- Notebook area and cells

Modal editor

Starting with IPython 2.0, the Jupyter Notebook has a modal user interface. This means that the keyboard does different things depending on which mode the Notebook is in. There are two modes: edit mode and command mode.

Edit mode

Edit mode is indicated by a green cell border and a prompt showing in the editor area:



When a cell is in edit mode, you can type into the cell, like a normal text editor.

Command mode

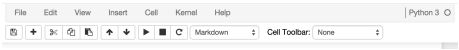
Command mode is indicated by a grey cell border with a blue left margin:

When you are in command mode, you are able to edit the notebook as a whole, but not type into individual cells. Most importantly, in command mode, the keyboard is mapped to a set of shortcuts that let you

perform notebook and cell actions efficiently. For example, if you are in command mode and you press `c`, you will copy the current cell - no modifier is needed.

Mouse navigation

All navigation and actions in the Notebook are available using the mouse through the menubar and toolbar, which are both above the main Notebook area:



The first idea of mouse based navigation is that **cells can be selected by clicking on them.** The currently selected cell gets a grey or green border depending on whether the notebook is in edit or command mode. If you click inside a cell's editor area, you will enter edit mode. If you click on the prompt or output area of a cell you will enter command mode.

The second idea of mouse based navigation is that **cell actions usually apply to the currently selected cell**. Thus if you want to run the code in a cell, you would select it and click the button in the toolbar or the Cell:Run" menu item. Similarly, to copy a cell you would select it and click the button in the toolbar or the "Edit:Copy" menu item. With this simple pattern, you should be able to do most everything you need with the mouse. Markdown and heading cells have one other state that can be modified with the mouse. These cells can either be rendered or un-rendered. When they are rendered, you will see a nice formatted representation of the cell's contents.

When they are un-rendered, you will see the raw text source of the cell. To render the selected cell with the mouse, click the button in the toolbar or the "Cell:Run" menu item.
To un-render the selected cell, double click on the cell.

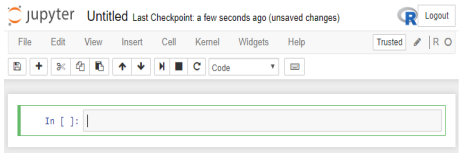
Keyboard navigation

The modal user interface of the Jupyter Notebook has been optimized for efficient keyboard usage. This is made possible by having two different sets of keyboard shortcuts: one set that is active in edit mode and another in command mode.

The most important keyboard shortcuts are `Enter`, which enters edit mode, and `Esc`, which enters command mode.

Creating a Notebook

Creating a Notebook is as straightforward as clicking on the New button on the top right, and selecting the kernel (i.e. the engine that will be interpreting our commands).



You'll notice the following points:

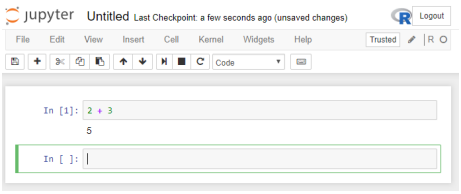
- There is an **`In[]`** section in the middle. This is called a cell and is essentially an interface where you can put your code, text, markdown, etc. Simply put, every cell that is an *input* is marked as **`In`** followed by an increasing number that corresponds to the relative order that the particular cell was executed.
- There is an indication of the current kernel being employed to execute each cell on the top right (in this instance the kernel is R).
- The notebook is still Untitled, but it has already been saved/created as a file (you can have a look at the working directory to verify this). Bear in mind that Jupyter automatically saves (i.e. auto saves) your notebooks, and also creates **checkpoints**, essentially snapshots of your Notebook that you can go back to (this is **NOT** versioning, as it doesn't capture all changes, just snapshots in time).

After writing some code/text in the currently active cell, the main keyboard command to remember is how to execute the code.

- **`Shift-Enter`**: Executes the code and creates a new cell underneath.
- **`Ctrl-Enter`**: Executes the code without creating a new cell.

Type the following code **2+3** into a cell and then hit **`Shift-Enter`**.

You will see the following screen (or similar):



First of all, you may have briefly seen an asterisk after `In`, i.e. `In [*]`.

The asterisk means that the kernel is currently trying to run the code, so you should be waiting for the output. After successful execution, the * will change to the next number of the Cell (1 in our instance), and the output of the command will be visible below (5 in our case). Finally, as we executed the code with `Shift-Enter`, a brand new cell has been created for us.

Mingling code and text

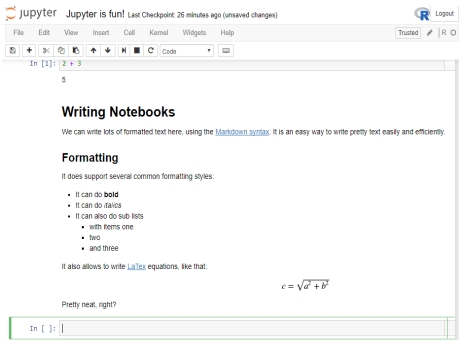
One of the most powerful things in Jupyter is the fact that you can write both text and code in the same notebook - much like a real Lab notebook where you have your text notes and your equations/figures/etc.

Let's try and put some text in our notebook. To do that, we need to tell Jupyter that the cell should be interpreted as text (Markdown-formatted) and not as code. Click on the empty cell (it should have a green outline), and then go to **`Cell -> Cell Type -> Markdown`**. You will notice that the `In [ ]` indicator just disappeared, as there will be no need to execute something (and therefore no output will be produced).

Let's copy the following text into the cell:

```
# Writing Notebooks

We can write lots of formatted text here,
using the [Markdown
syntax](https://en.wikipedia.org/wiki/Markdo
wn). It is an easy way to write pretty text
easily and efficiently.

## Formatting

It does support several common formatting
styles:

- It can do **bold**
- It can do _italics_
- It can also do sub lists
  * with items one
  * two
  * and three

It also allows to write
[LaTex](https://www.latex-project.org)
equations, like that:

$$c = \sqrt{a^2 + b^2}$$

Pretty neat, right?
```

If you press Shift-Enter after putting this text, it should look like that:



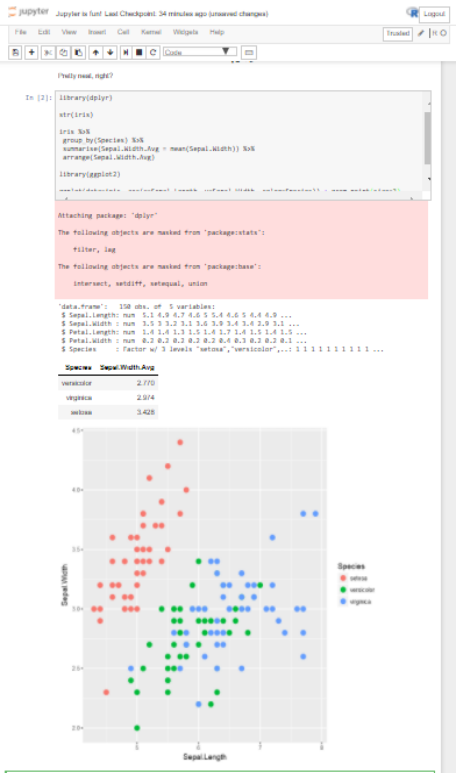
You'll also notice that, by default, Jupyter has changed the type of the new cell to `R`, so you won't have to change types constantly, but only when needed.

Finally, let's copy some more R code, to include some graphics as well.

```
library(dplyr)

str(iris)

iris %>%
group_by(Species) %>%
summarise(Sepal.Width.Avg=mean(Sepal.Width))
%>% arrange(Sepal.Width.Avg)

library(ggplot2)

ggplot(data=iris,
aes(x=Sepal.Length,
y=Sepal.Width,
color=Species))
+ geom_point(size=3)
```

`iris` is a small dataset that is included in the `dplyr` package.

If you have both `dplyr` and `ggplot2` libraries, installed, you should see something similar to this:



Stopping, Rename or duplicate a notebook

To shutdown, delete, duplicate, or rename a notebook check the checkbox next to it and an array of controls will appear at the top of the notebook list (as seen below). You can also use the same operations on directories and files when applicable.

Listing running Notebooks

To see all of your running notebooks along with their directories, click on the "Running" tab. This view provides a convenient way to track notebooks that you start as you navigate the file system in a long running notebook server.

The Kernel

A notebook kernel is a “computational engine” that executed the code contained in a Notebook document. When you open a Notebook document, the associated kernel is automatically launched. When the notebook is executed the kernel performs the computation and produces the results. dashboard serves as a home page for the notebook. A complete list of Jupyter kernels are available here.

The Future of Jupyter Notebook

According to the people that created and support Jupyter, it has now ~3 million users worldwide, and over 500k Notebooks on GitHub - so huge success!

The same team is now working on JupyterLab (currently in alpha), essentially the next generation of the Jupyter Notebook application. You can read more here and also have a look at their recent talk at SciPy2016.

How to get access

- Connect in your browser to [R4].
- Log in with your EGI SSO account.
- After logging in, the initial Jupyter screen will be displayed. Click the “Start my notebook” to create a personal notebook.
- Enjoy with your notebook!

Reading material

Some of the content is from the Data Camp module.
References for learning Python:

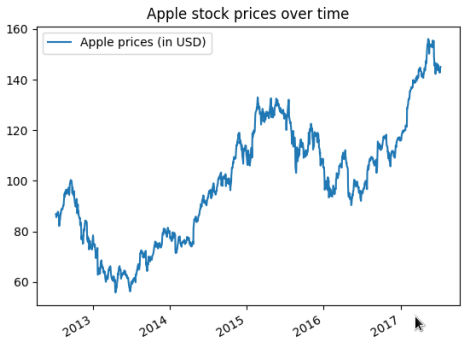
- http://rosalind.info/problems/locations/
- http://learnpythonthehardway.org/book
- http://www.learnpython.org/

Analyze datasets with Jupyter notebook

In this first exercise, we will show how the Jupyter Notebook can be used analyse external datasets. As example of datasets, we consider the stocks prices available through the NASDAQ portal [R5]. From this portal, search and download the Apple and Facebook stock prices in the last 5 years and save them in CVS files format.

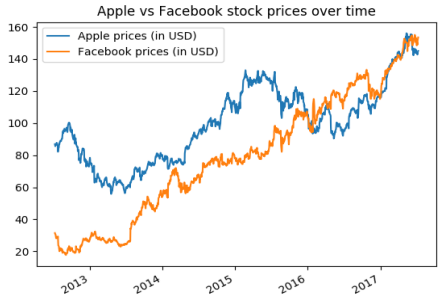
To analyze these datasets we use the following python script:

```
import matplotlib.pyplot as plt
import datetime as dt
import urllib2
import csv

# Apple stock prices
url = 'http://grid.ct.infn.it/cron_files/CODATA- RDA/AppleStock.csv'
response = urllib2.urlopen(url)

datasets = csv.reader(response, delimiter=',')

#Skipping the header from the CVS file
for row in datasets:
        if not ('/' not in row[0]):
                x.append(row[0])
                y.append(row[1])

x = [dt. datetime.strptime(d,'%Y/%m/%d').date()
for d in x]

# Creating the plot
plt.plot(x,y, label='Apple prices (in USD)')

plt.gcf().autofmt_xdate()

plt.title('Apple stock prices over time')
plt.legend()
plt.show()
plt.savefig('Apple_stocks.png')
```



To plot two different datasets in the same graph use the follow trick:

```
# Facebook stocks
url = 'http://grid.ct.infn.it/cron_files/ELIXIR_WS/FacebookStock.csv'
response = urllib2.urlopen(url)

datasets = csv.reader(response, delimiter=',')

# Reading data and saving it in x2, y2
[..]

# Plotting all together
plt.plot(x,y, label='Apple prices (in USD)')
plt.plot(x2,y2, label='Facebook prices (in USD)')

plt.gcf().autofmt_xdate()

plt.title('Apple vs Facebook stock prices over
time')
plt.legend()
plt.show()
plt.savefig('Stocks.png')
```



Source code can be downloaded from here [R6, R7].

larocca 24/7/17 10:16
Commenta [1]: This image has to be updated by Savina

Useful link

[R1] The Federated Cloud IaaS Orchestration available at: https://wiki.egi.eu/wiki/Federated_Cloud_IaaS_Orchestration
[R2] The EGI AppDB VMOps dashboard available at: https://dashboard.appdb.egi.eu/
[R3] The Jupyter Notebook is available at: jupyter.org
[R4] The EGI JupyterHub is available at: https://jupyterhub.fedcloud-tf.fedcloud.eu
[R5] NASDAQ historical stock prices available at: http://www.nasdaq.com/symbol/aapl/historical
[R6] AppleStock-py is available at: http://grid.ct.infn.it/cron_files/CODATA-RDA/AppleStock.py
[R7] FacebookvsAppleStock.py is available at: http://grid.ct.infn.it/cron_files/CODATA-RDA/FacebookvsAppleStock.py

References

- Alexander J. Westermann, Konrad U. Förstner, Fabian Amman, Lars Barquist, Yanjie Chao, Leon N. Schulte, Lydia Müller, Richard Reinhardt, Peter F. Stadler & Jörg Vogel – **Dual RNA-seq unveils noncoding RNA functions in host–pathogen interactions**, doi:10.1038/nature16547
- J. Hepojoki et al. – **Arenavirus Coinfections Are Common in Snakes with Boid Inclusion Body Disease**, 2015 Journal of Virology, doi:10.1128/JVI.01112-15
- M Aleksi Kallio, Jarno T Tuimala, Taavi Hupponen, Petri Klemelä, Massimiliano Gentile, Ilari Scheinin, Mikko Koski, Janne Käki and Eija I Korpelainen – **Chipster: user-friendly analysis software for microarray and other high-throughput data**, BMC Genomics 2011