II Congreso Colombiano de
Investigación Operativa
ASOCIO 2017

Una introducción a Analytics con Anaconda Python

JUAN DAVID VELÁSQUEZ HENAO, MSc, PhD
Profesor Titular
Departamento de Ciencias de la Computación y la Decisión
Facultad de Minas
Universidad Nacional de Colombia, Sede Medellín

jdvelasq@unal.edu.co
@jdvelasquezh
https://github.com/jdvelasq
https://goo.gl/prkjAq
https://goo.gl/vXH8jy

¿Qué es y qué NO es Analytics?

Big Data

Área relacionada con la infraestructura computacional requerida para el almacenamiento de grandes volúmenes de datos y la ejecución de algoritmos de forma paralela y distribuida.

Data Science

Área relacionada con los procesos y sistemas para la extracción de conocimiento de datos almacenados electrónicamente (¿para la toma de decisiones? ¿para probar hipótesis?)

Analytics

Proceso científico de transformación de datos en conocimiento para mejorar el proceso de toma de decisiones [Informs].

- Problema organizacional
- Transformación en un problema de analytics
- Datos
- Selección de la metodología
- Desarrollo del modelo
- Puesta en marcha de la solución
- Gestión del ciclo de vida del modelo

Data Science and Data Scientists: What's in a Name?

Business Intelligence Practitioner

Combinación de negocios + tecnología con el fin de proveer información a las unidades de negocios para toma de decisiones

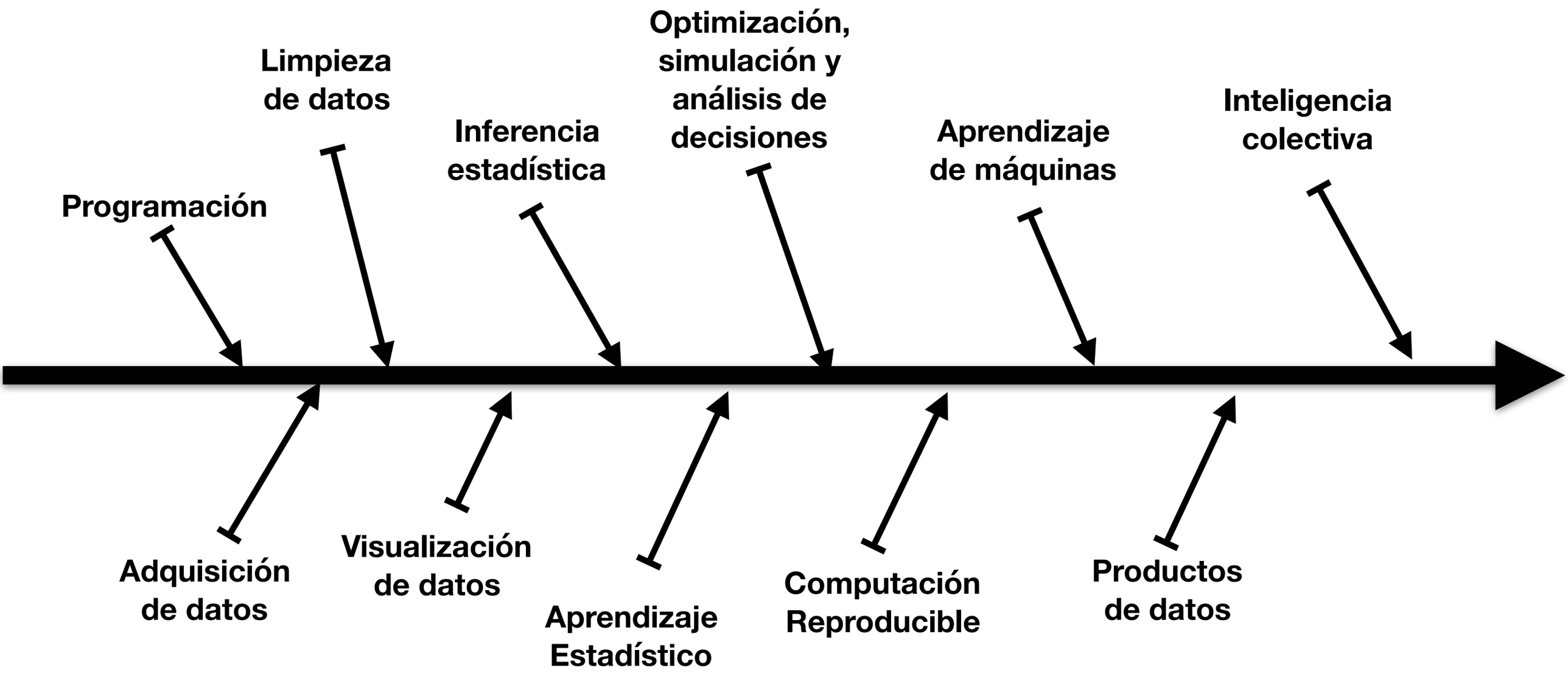
Data Scientist

Habilidades en la programación de computadores para manejo de datos y modelado predictivo (estadística, aprendizaje de máquinas, minería de datos, etc.).

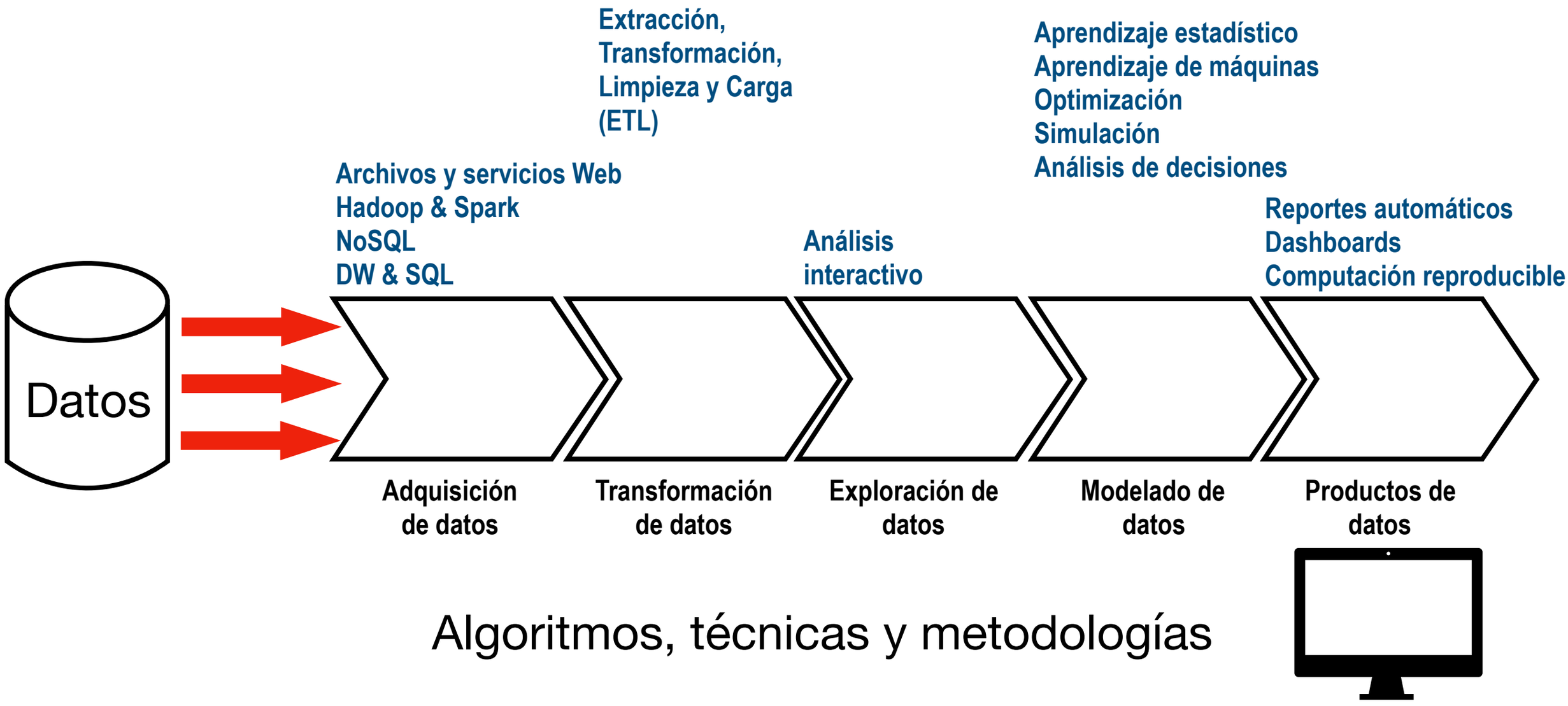
Analytics Practitioner

Data Science + Inteligencia de Negocios + Optimización + Simulación + Toma de Decisiones

Analytics se refiriere al proceso integral, no a sus partes



Proceso de modelado en Analytics



Infraestructura computacional

- Un procesador
  - Computación en máquinas locales
- Muchos procesadores
  - Computación en la nube

El lenguaje Python

Python es un lenguaje de alto nivel desarrollado originalmente por Guido van Rossum en 1989 como un proyecto de programación por hobby; Python sería sucesor del lenguaje ABC, que fue diseñado para prototipado y enseñanza.

Su nombre fue inspirada en la serie de televisión *Monty Python*'s *Flying Circus* transmitida originalmente entre 1969 y 1973.

Python es un lenguaje interpretado y multi-paradigma que soporta completamente la programación orientada a objetos y la programación estructurada. Hay soporte parcial para metaprogramación (los programas son tratados como objetos) y programación orientada a aspectos.

El lenguaje Python

El lenguaje da soporte parcial a la programación funcional:
implementa las funciones map(), reduce() y filter (); comprehensions
para listas, diccionarios y conjuntos; y generadores.

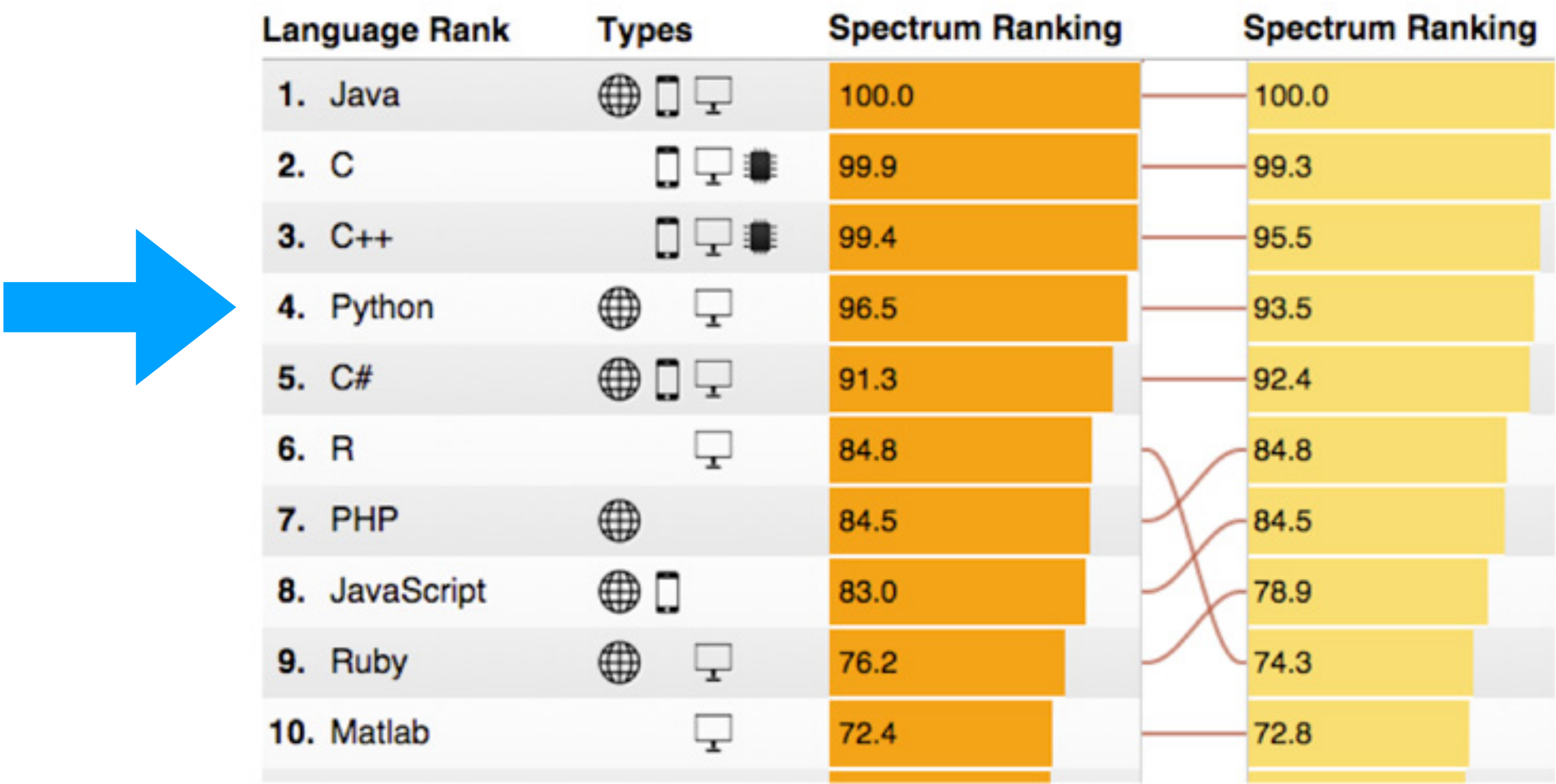
```
S = [2 * x for x in range(101) if x ** 2 > 3]
```

Su diseño se basa en la claridad y la simplicidad (pythonic):

“Es mejor una sola forma obvia de hacer las cosas”.

Es un lenguaje sencillo, con sintaxis inspirada en los lenguajes
ALGOL, C, C++, Haskell, Java, ec. al donde la funcionalidad se
obtiene a partir de un core sencillo y una gran librería estándar.
Esto permite que sea ideal para el desarrollo de prototipos.

The **2015** Top Ten Programming Languages (IEEE Spectrum)



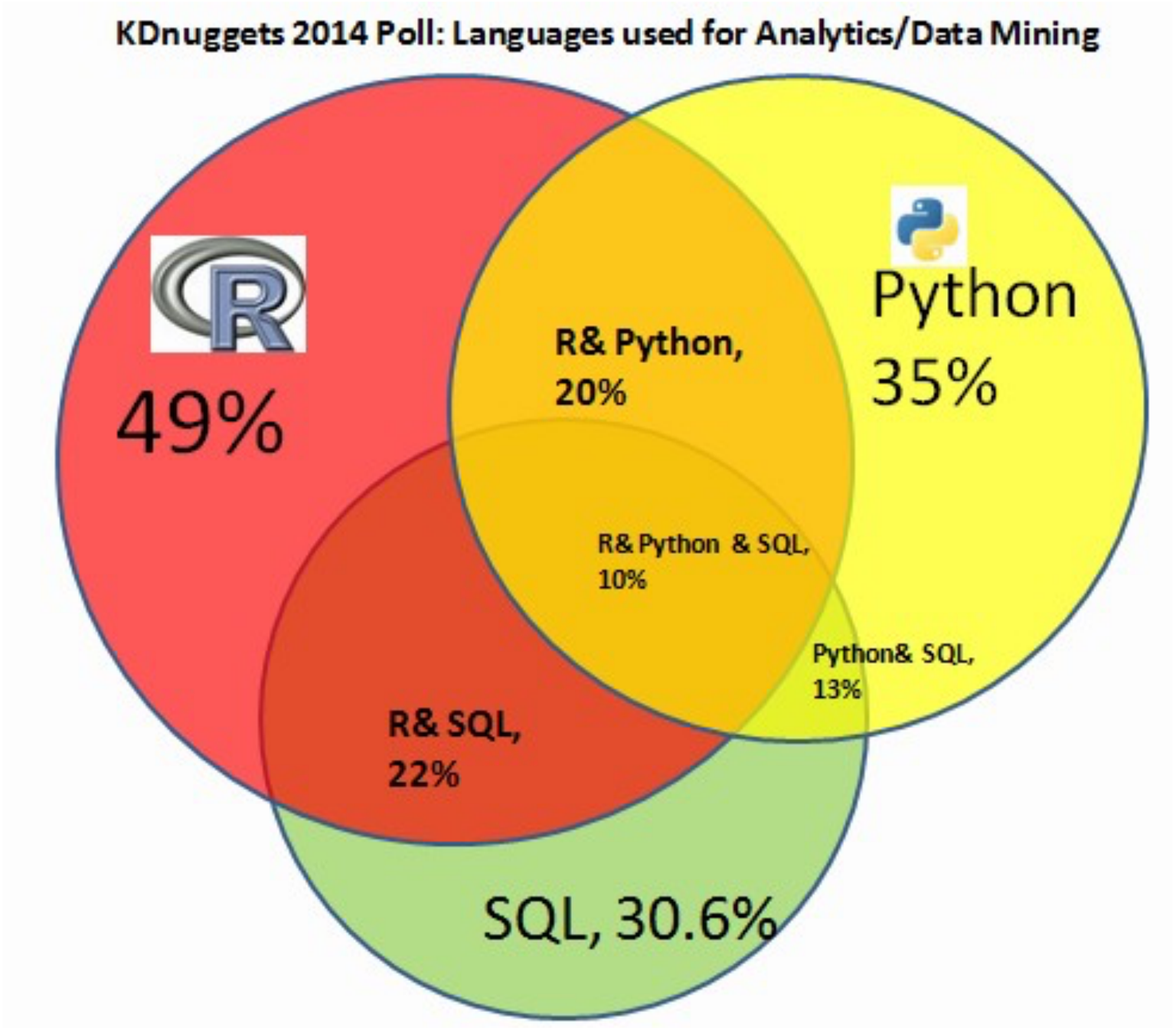
The **2016** Top Ten Programming Languages (IEEE Spectrum)

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

The **2017** Top Ten Programming Languages (IEEE Spectrum)

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.5
4. C++		97.1
5. C#		87.7
6. R		87.7
7. JavaScript		85.6
8. PHP		81.2
9. Go		75.1
10. Swift		73.7

Lenguajes usados en Analytics/Data Science

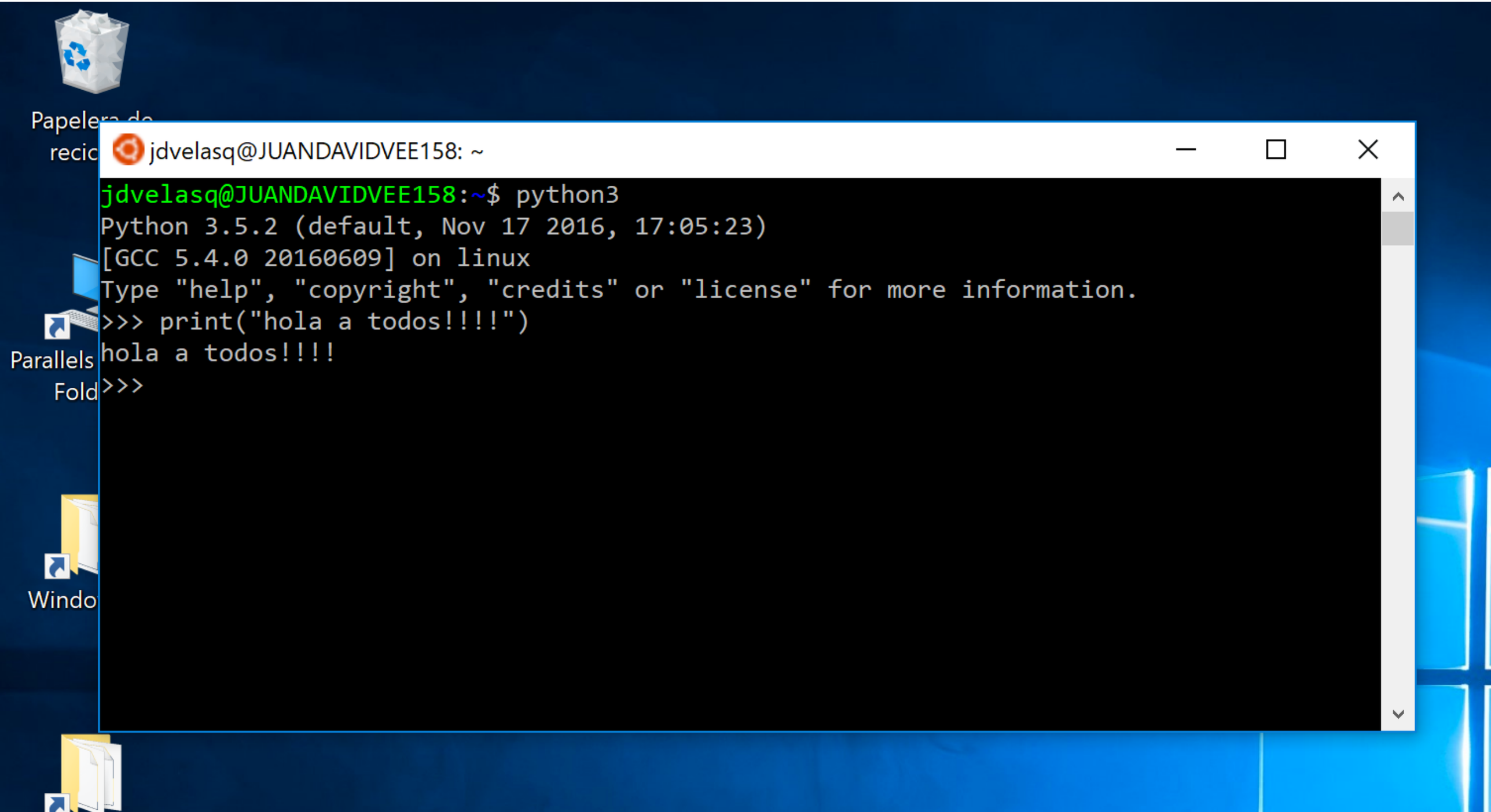


http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html

Distribuciones de Python

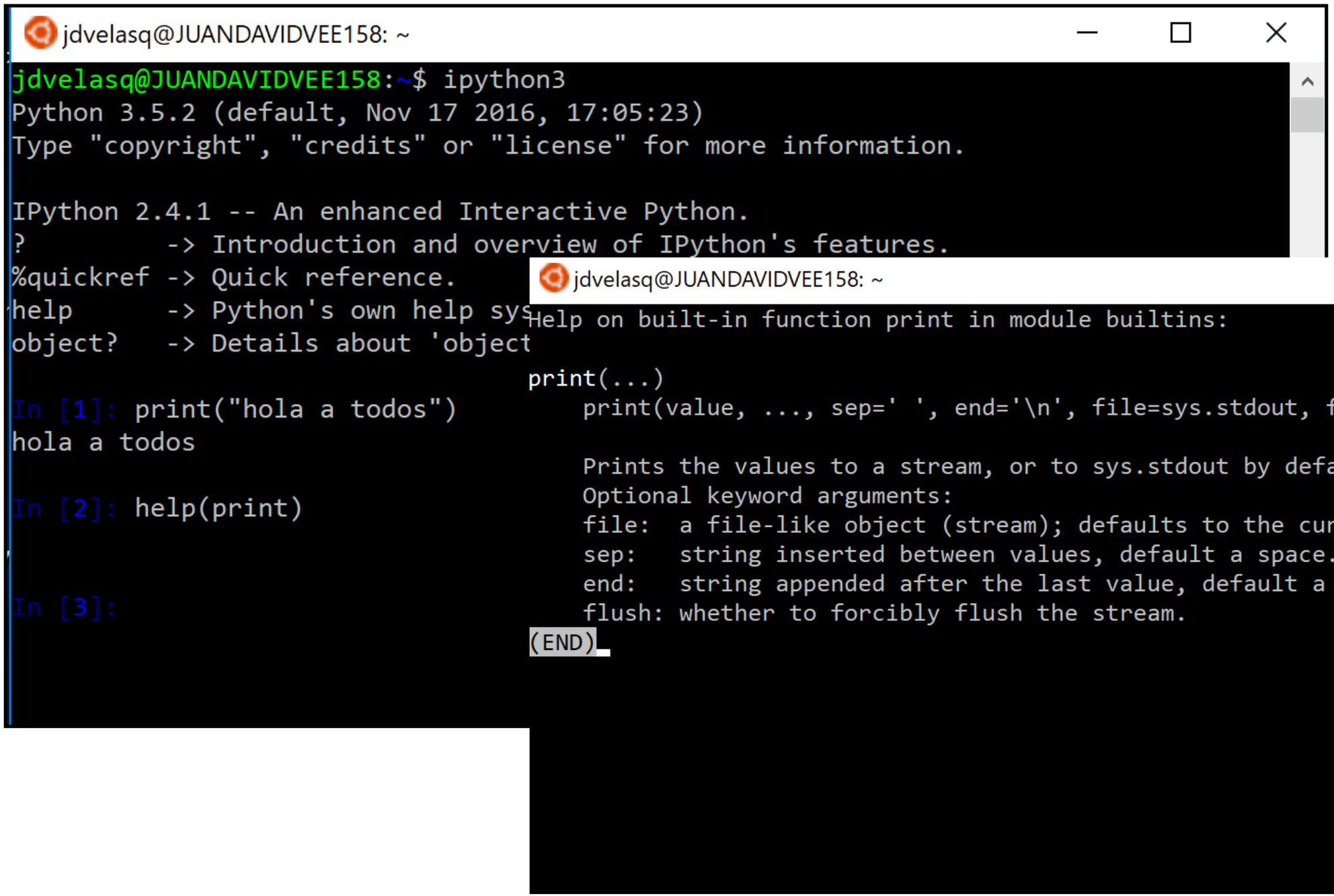
- Preinstalado en los sistemas operativos Linux, mac OS y Bash on Ubuntu on Windows
- Python Software Foundations — www.python.org
- Anaconda — www.continuum.io
  - Blaze
  - Bokeh
  - conda
  - Dask
  - llvmlite
  - PhosphorJS
  - Numba
- Enthought Canopy — www.enthought.com/products/canopy/
  - Python for Excel (PyXLL)

Python en Windows 10



IPython en Windows 10

```
$ sudo apt install ipython3
```



Alternativas para ejecutar un programa en Python

En el prompt de comandos

`$ nano hola.py`

jdvelasq@JUANDAVIDVEE158: ~

```
 GNU nano 2.5.3                                File: hola.py

print("hola mundo cruel")
```

jdvelasq@JUANDAVIDVEE158: ~

```
jdvelasq@JUANDAVIDVEE158:~$ nano hola.py
jdvelasq@JUANDAVIDVEE158:~$ python3 hola.py
hola mundo cruel
jdvelasq@JUANDAVIDVEE158:~$
```

Alternativas para ejecutar un programa en Python

En el prompt de Python de forma interactiva

```
jdvelasq@JUANDAVIDVEE158: ~
jdvelasq@JUANDAVIDVEE158:~$ python3
Python 3.5.2 (default, Nov 17 2016, 17:05:23)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hola mundo cruel")
hola mundo cruel
>>>
```

Alternativas para ejecutar un programa en Python

En el prompt de IPython de forma interactiva

```
jdvelasq@JUANDAVIDVEE158: ~
jdvelasq@JUANDAVIDVEE158:~$ ipython3
Python 3.5.2 (default, Nov 17 2016, 17:05:23)
Type "copyright", "credits" or "license" for more information.

IPython 2.4.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: print("hola mundo cruel")
hola mundo cruel

In [2]: quit()
jdvelasq@JUANDAVIDVEE158:~$
```

Alternativas para ejecutar un programa en Python

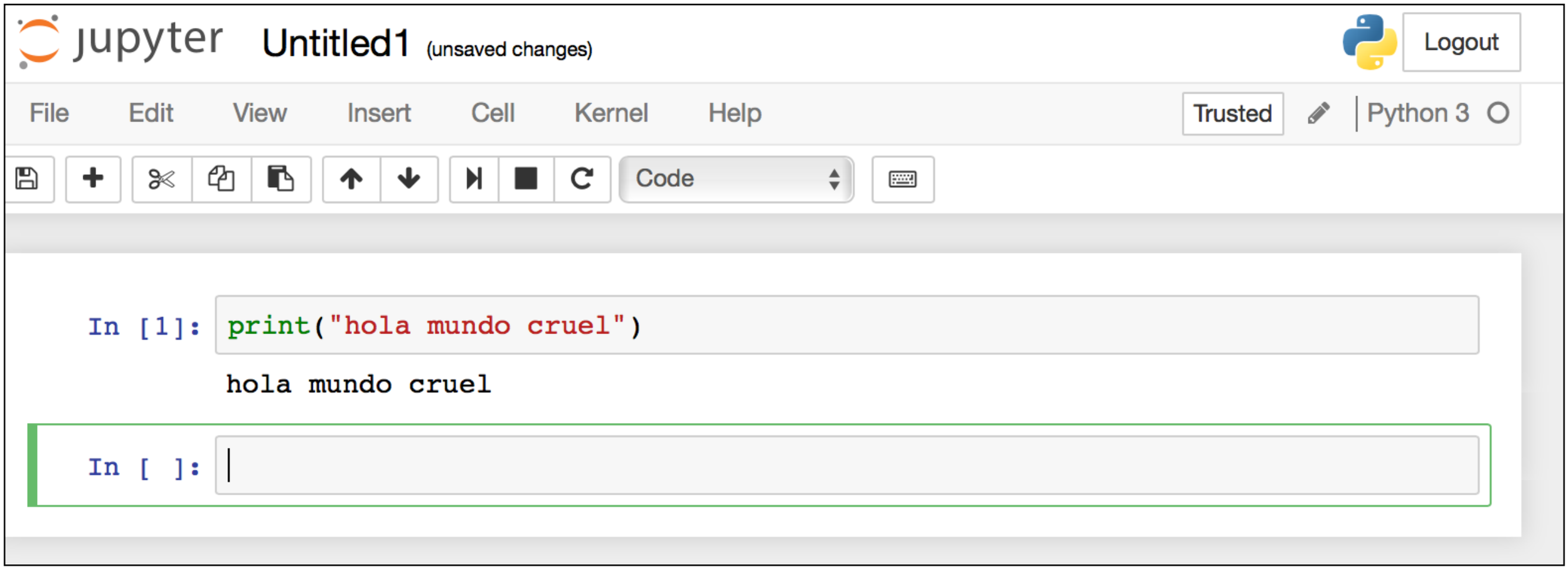
Desde un entorno de desarrollo

- **PyCharm — www.jetbrains.com/pycharm/**
- **Wingware — http://wingware.com**
- **Spyder — https://pythonhosted.org/spyder/**
- **IDLE (nativo, viene con la instalación del lenguaje).**
- **IDLEX (extensiones que amplían la capacidad de IDLE).**

Alternativas para ejecutar un programa en Python

Desde un entorno interactivo

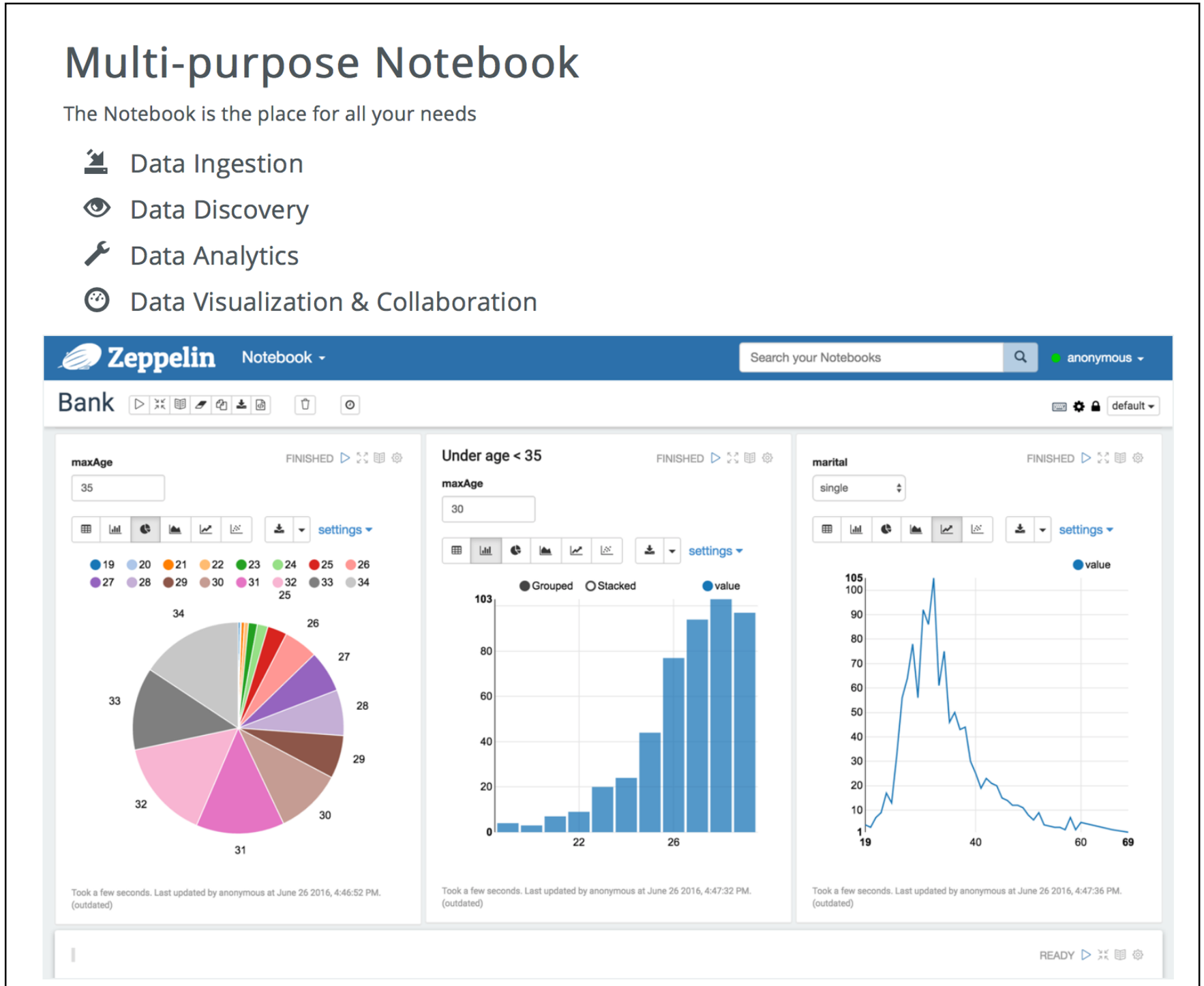
Jupyter Notebook



Alternativas para ejecutar un programa en Python

Desde un entorno interactivo

Apache Zepellin



Alternativas para ejecutar un programa en Python

Desde un entorno interactivo

Beaker / BeakerX

BeakerX: Beaker extensions for Jupyter

build passing | chat on gitter | JitPack 0.1.1 | npm package 0.0.6 | pypi

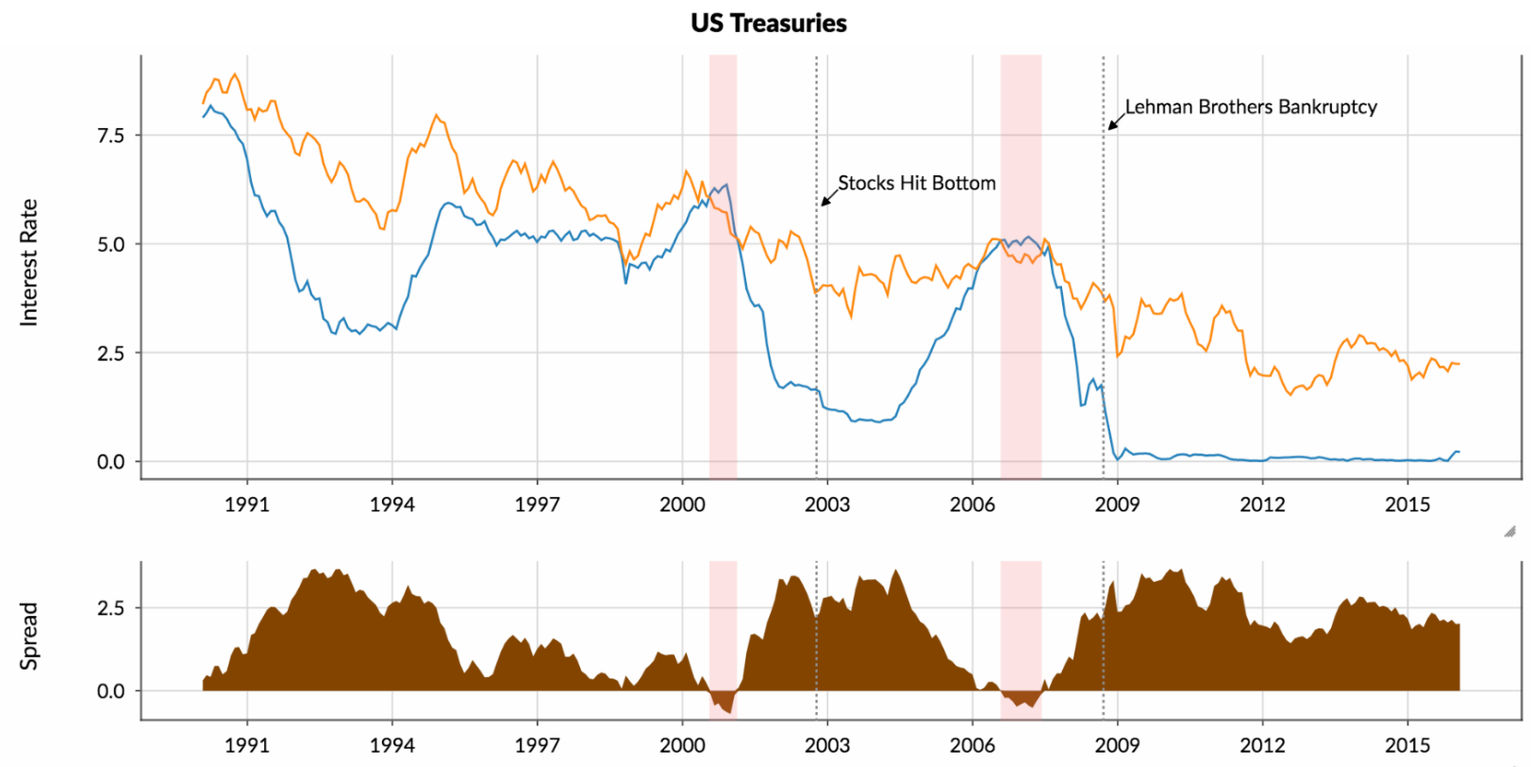
BeakerX is a collection of JVM kernels with widgets, plotting, t
Notebook and Jupyter Lab. BeakerX is in alpha, with major fea
changes without notice.

The documentation consists of tutorial notebooks on GitHub.

BeakerX is the successor to the Beaker Notebook (source cod
are hiring.

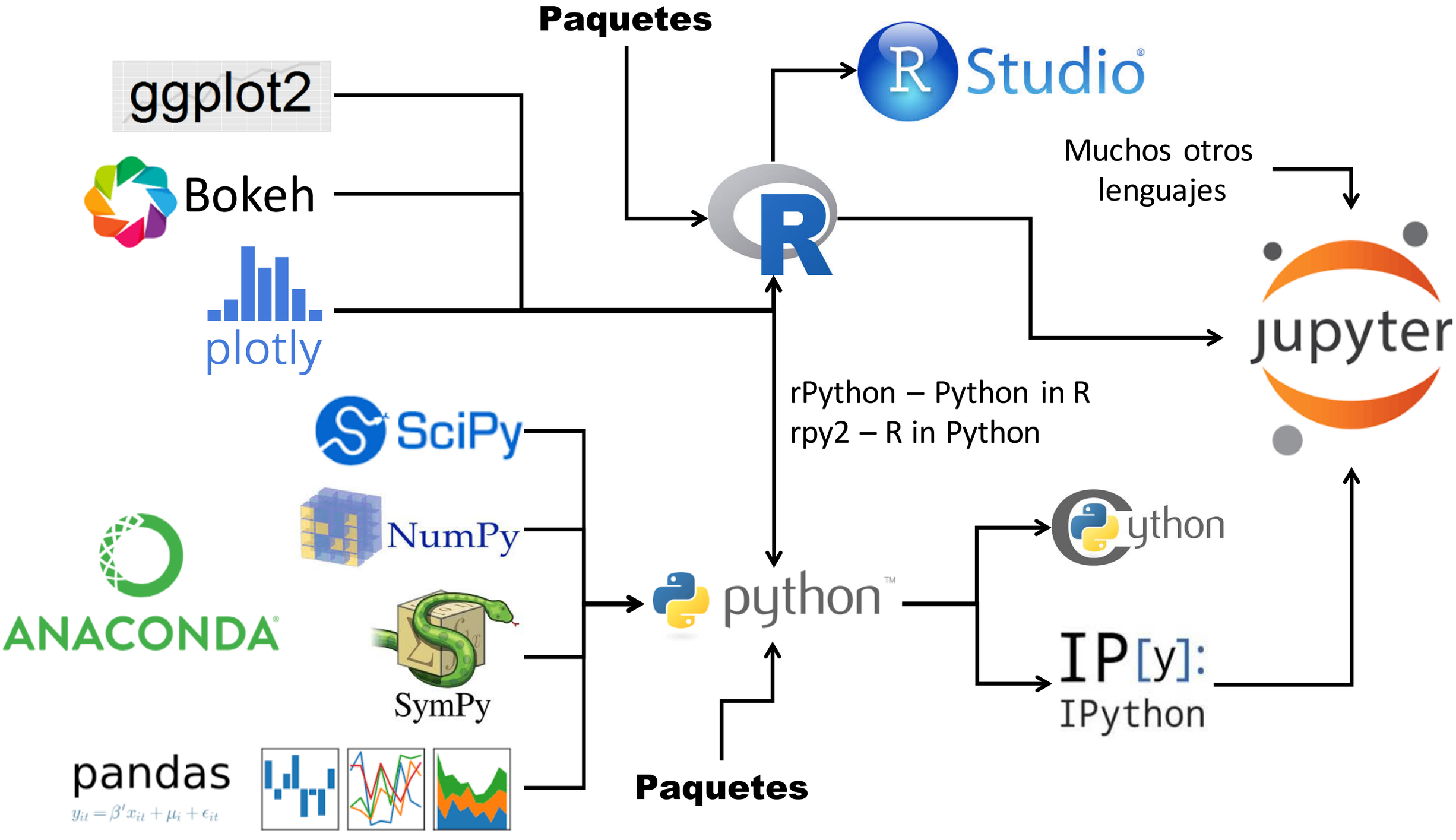
Groovy with Interactive Plotting and Tables:

```
// Then use a CombinedPlot to get stacked plots with linked X axis.
def c = new CombinedPlot(title: "US Treasuries", initWidth: 1000)

// add both plots to the combined plot, and including their relative heights.
c.add(p1, 3)
c.add(p2, 1)
```



https://github.com/twosigma/beakerx

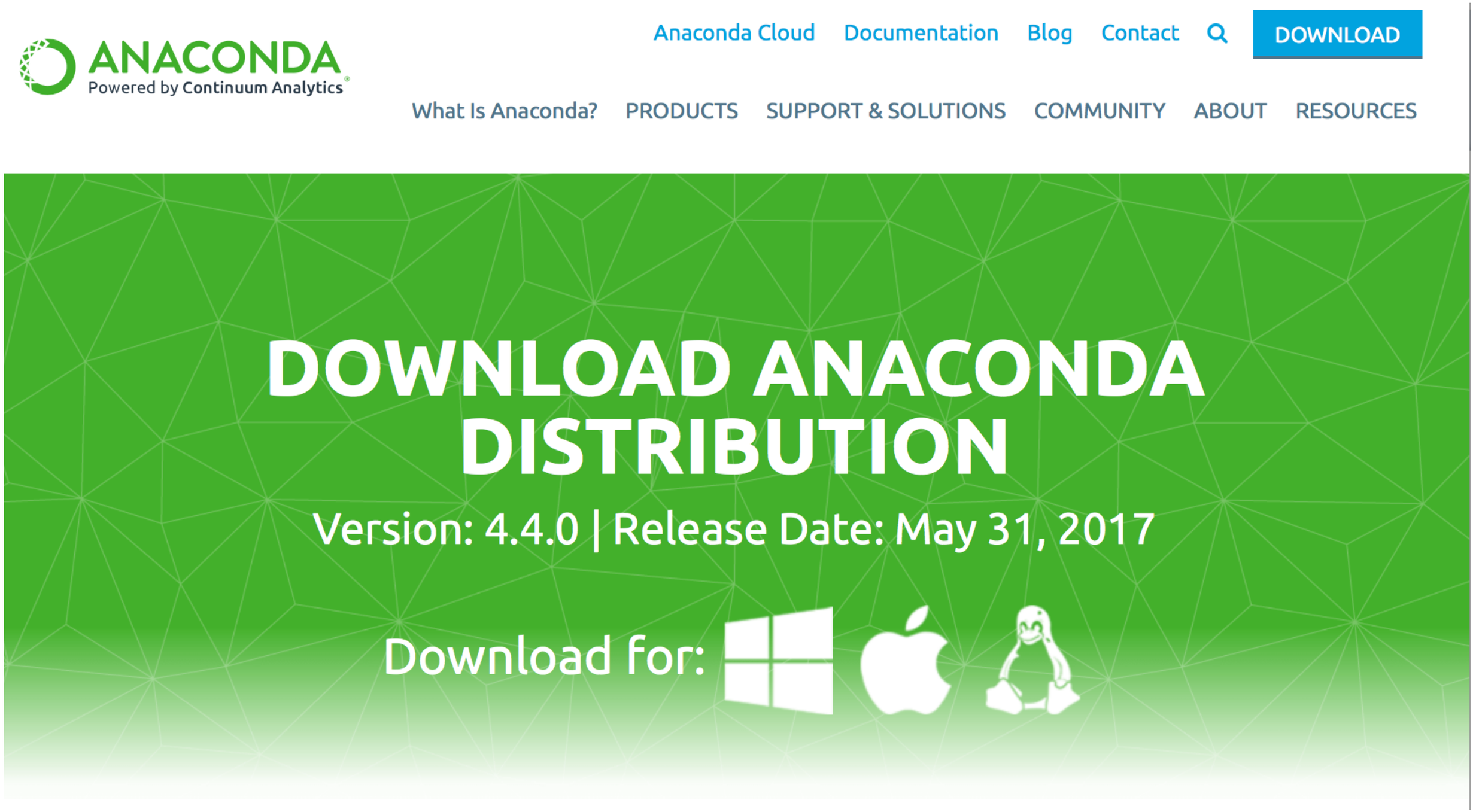
Ecosistema de computación científica



Anaconda Python

Instalación de Anaconda Python

https://www.continuum.io/downloads



ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

- Home
- Environments
- Projects (beta)
- Learning
- Community

Applications on root | Channels | Refresh

notebook
5.0.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

qtconsole
4.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

spyder
3.1.4
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

glueviz
0.10.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

orange3
3.4.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

rstudio
1.0.153
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install

Documentation

Developer Blog

Feedback

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

- Home
- Environments
- Projects (beta)
- Learning
- Community

Documentation

Developer Blog

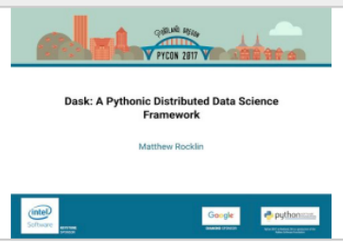
Feedback

Documentation (11) | Training (2) | Video (20) | Webinar (20)

Search

Anaconda Skills Accelerator Program

Explore



Dask: A Pythonic Distributed Data Science Framework PyCon 2017

View

Parallel Data Analysis PyCon 2017

View

Future of Data Science in the City of Boston | Andrew Therriault | AnacondaCON 2017

View

The Next Generation of Data Products | Hilary Mason | AnacondaCON 2017

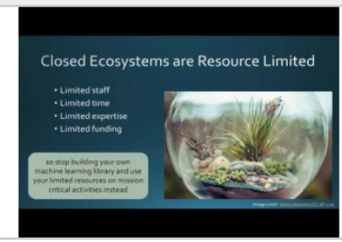
View

Open Data Science Education | Jonathan Cornelissen | AnacondaCON 2017

View

Developing & Deploying Credit Ri Models with Anaconda | Hussain Sultan | AnacondaCON 2017

View



Leveraging Open Technologies in Closed Ecosystem | Dharhas Pothina | AnacondaCON 2017

View

Keynote: The Anaconda Roadmap Michele Chambers | AnacondaCO 2017

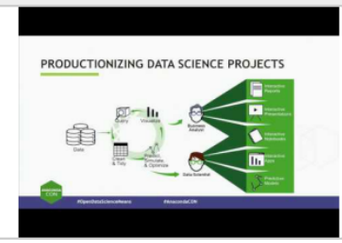
View

Intel Partner Showcase | AnacondaCON 2017

View

AnacondaCREW Panel | AnacondaCON 2017

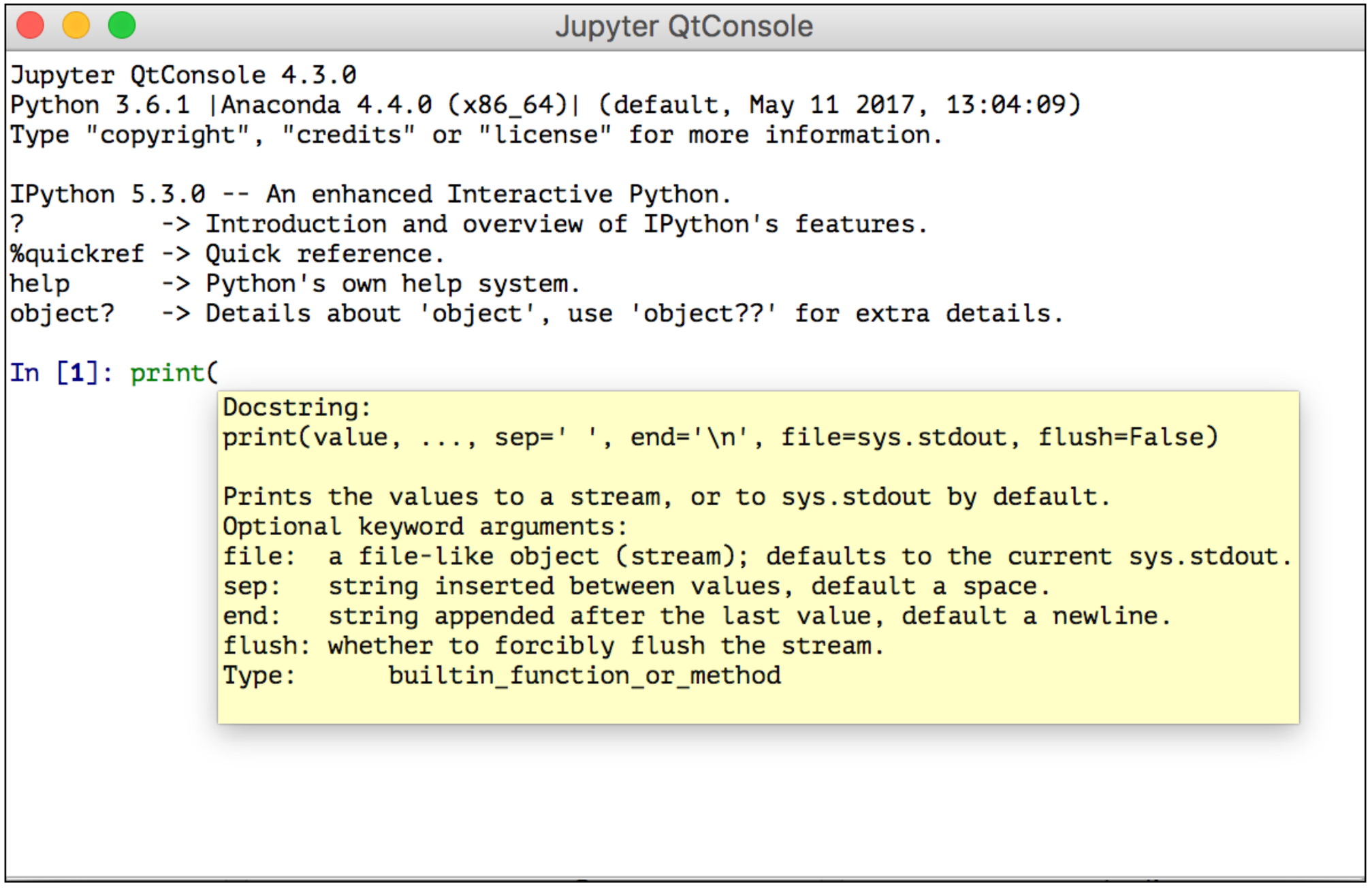
View



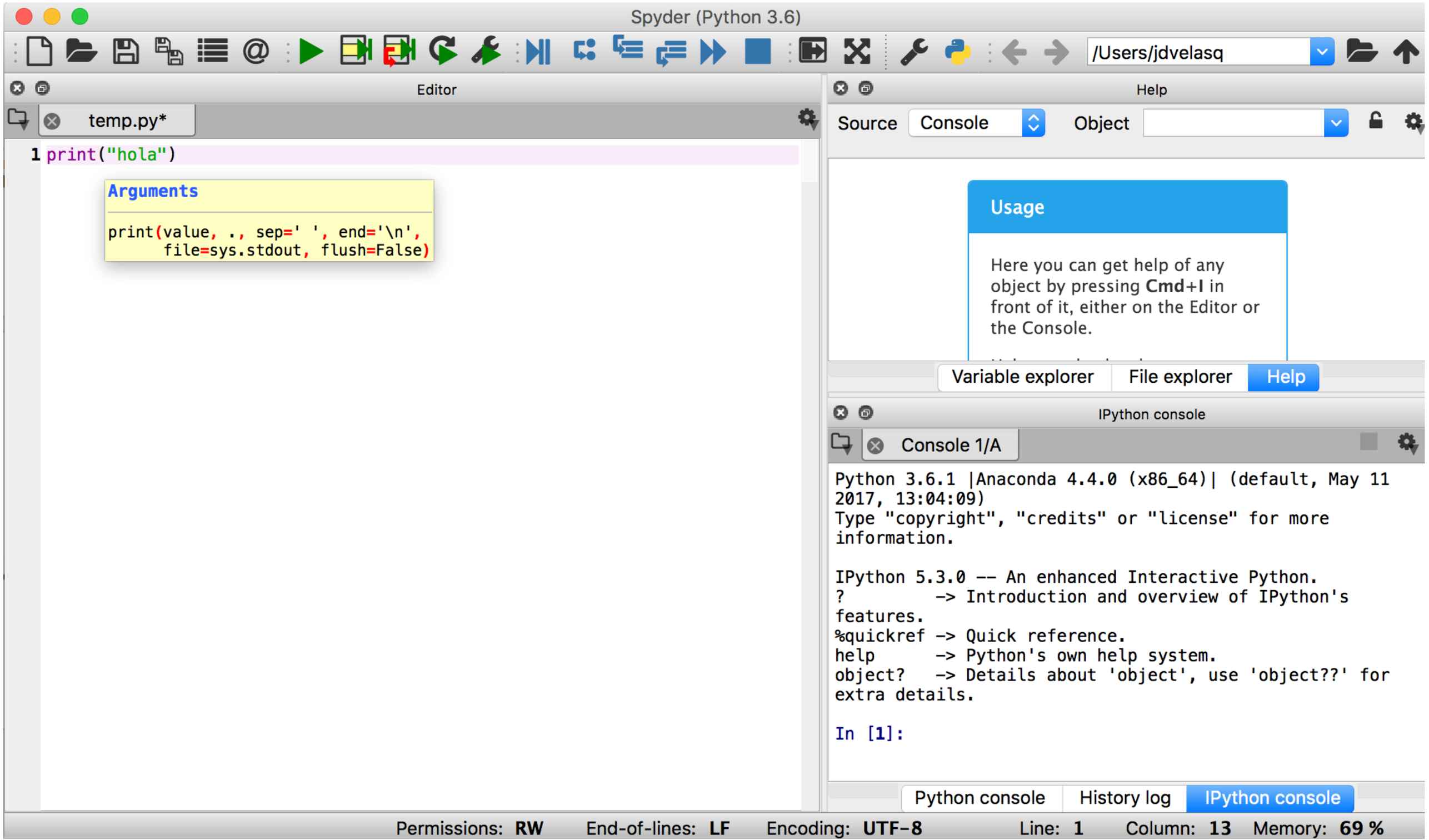
Scalable & Deployable Data Scien with Anaconda | Kris Overholt | AnacondaCON 2017

View

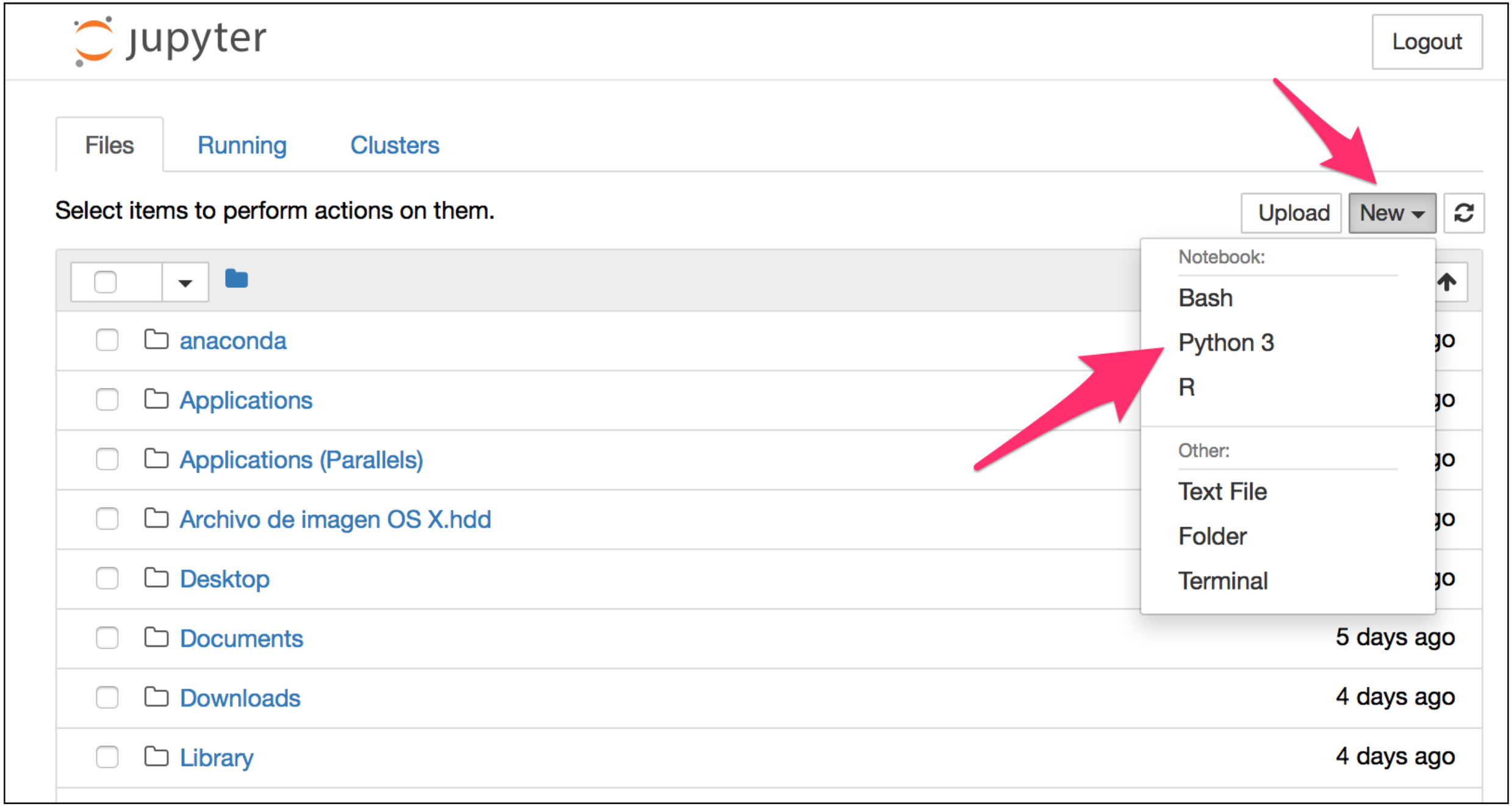
Jupyter QtConsole



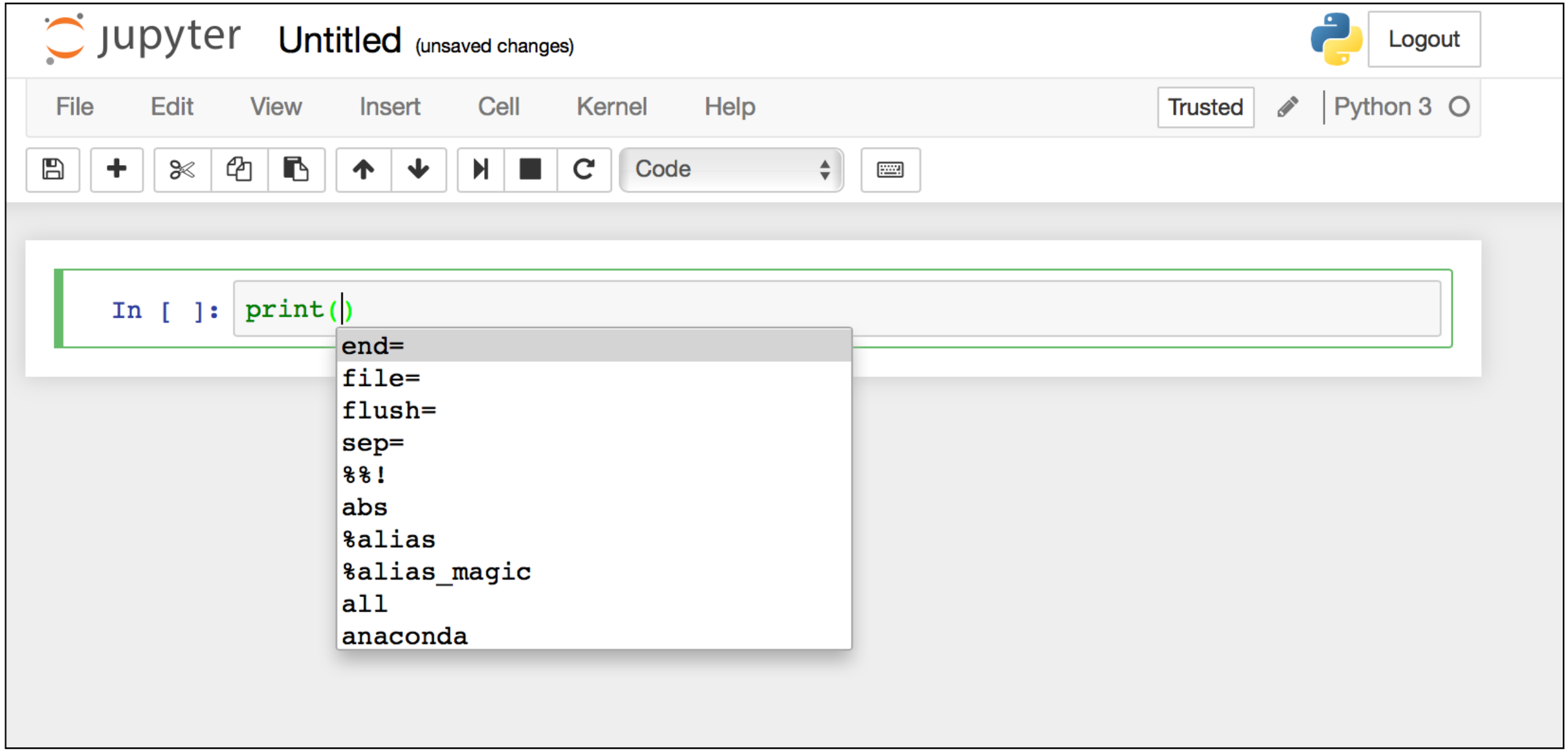
Spyder


Spyder (Python 3.6)
/Users/jdvelasq
Editor
temp.py*
1 print("hola")
Arguments
print(value, ., sep=' ', end='\n',
file=sys.stdout, flush=False)
Help
Source
Console
Object
Usage
Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.
Variable explorer
File explorer
Help
IPython console
Console 1/A
Python 3.6.1 |Anaconda 4.4.0 (x86_64)| (default, May 11 2017, 13:04:09)
Type "copyright", "credits" or "license" for more information.
IPython 5.3.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
In [1]:
Python console
History log
IPython console
Permissions: RW
End-of-lines: LF
Encoding: UTF-8
Line: 1
Column: 13
Memory: 69 %

Jupyter Notebook



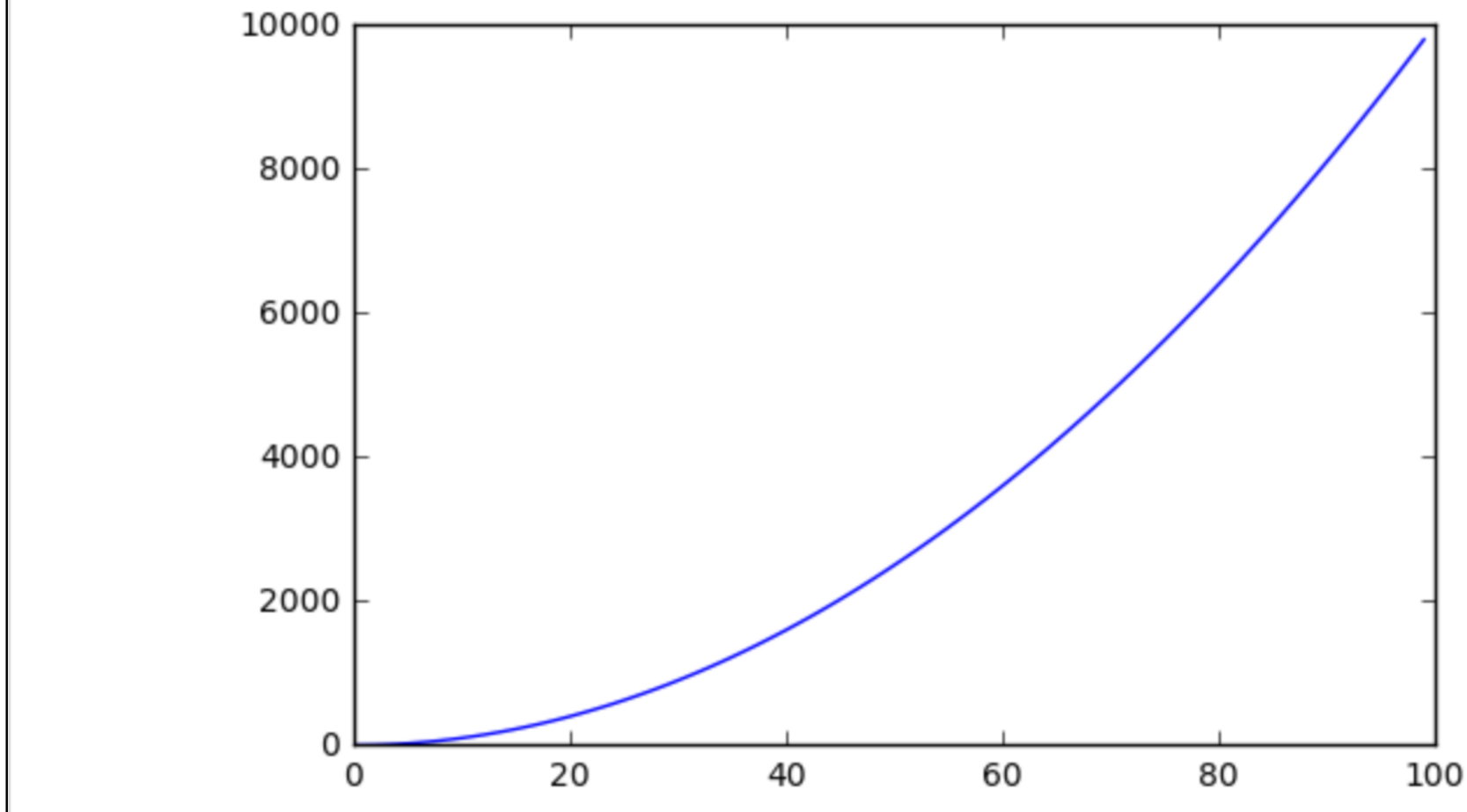
Jupyter Notebook



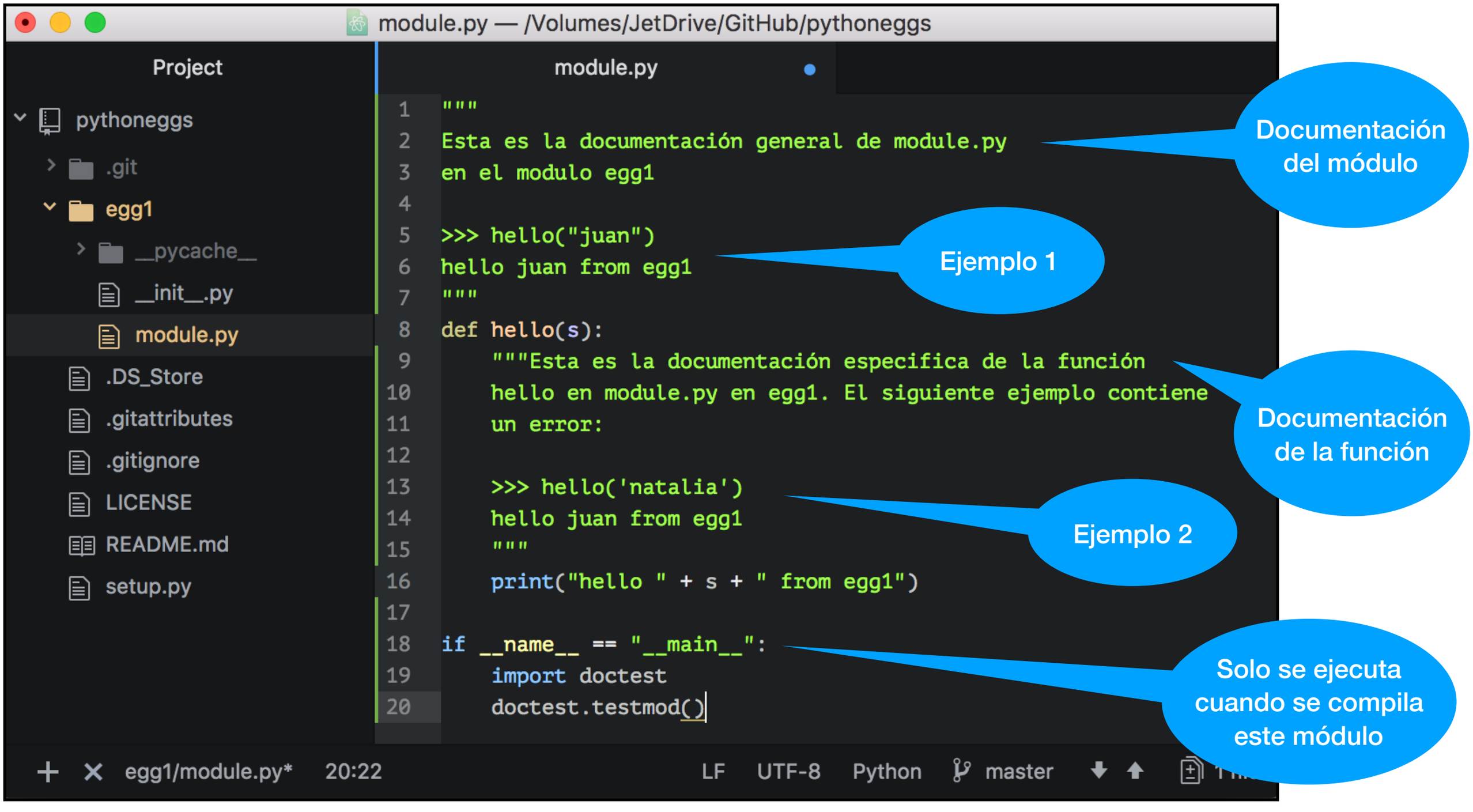
Jupyter Notebook

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
%matplotlib inline
```

In [2]:
```
x = range(100)
y = [value ** 2 for value in x]
plt.plot(x, y)
plt.show()
```



Programación



Programación

IPython: GitHub/pythoneggs

```
-bash-3.2$ pwd
[redacted]/GitHub/pythoneggs/egg1
-bash-3.2$ python module.py
**********************************************************************
File "module.py", line 13, in __main__.hello
Failed example:
    hello('natalia')
Expected:
    hello juan from egg1
Got:
    hello natalia from egg1
**********************************************************************
1 items had failures:
   1 of   1 in __main__.hello
***Test Failed*** 1 failures.
-bash-3.2$
```

Reporte automático de errores

Programación

This repository Search Pull requests Issues Marketplace Gist

jdvelasq / IPython-for-data-science

Unwatch 1 Star 0 Fork 19

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Notas de clase - Extracción, transformación, visualización y carga de datos usando IPython

Edit

Add topics

3 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request

Create new file Upload files Find file Clone or download

jdvelasq readme

- Anaconda white papers
- files
- images
- precios-energia
- .gitignore
- IPy-01-uso-interactivo.ipynb
- IPy-02-programacion.ipynb

Cálculos numéricos

Contenido

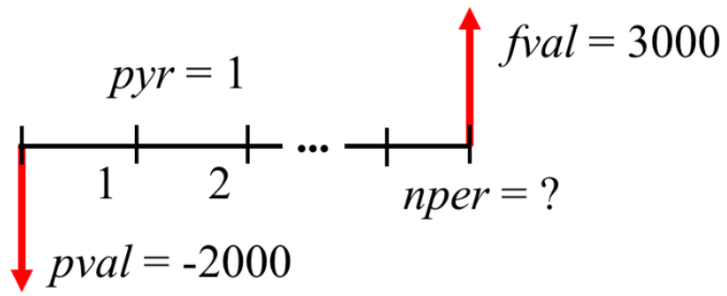
IPython puede ser usado de forma interactiva como una calculadora. Esto permite que el análisis de datos pueda ser realizado de forma interactiva, de forma similar a como pueden usarse otras herramientas como el lenguaje R o Matlab. A continuación se ejemplifican los cálculos aritméticos básicos.

```
In [1]: 2 + 2 + 1
Out[1]: 5

In [1]: 50 - 5 * 6 + 8
Out[1]: 28

In [3]: (50 - 5 * 6) / 4
Out[3]: 5.0
```

https://github.com/jdvelasq/IPython-for-data-science

Programación

Ejemplo.-- [3, pág. 88] Se depositan $ 2000 en una cuenta de ahorros que paga un interés anual del 7.2% (capitalizado anualmente). Si no se hacen otros depósitos en la cuenta, ¿cuánto tiempo se requiere para que la cuenta tenga $ 3000? R/ 5.83




```
In [9]: cf.pvfv(nrate =     7.2,    #
                pval  = -2000,      #
                fval  = +3000)      #
```

```
Out[9]: 5.8318433820838607
```

```
In [10]: # Ya que nper es un valor ent
         # para tener un balance de al
         # El balance al final de los
         cf.pvfv(nrate =     7.2, # ta
                 pval  = -2000,   # va
                 nper  =     6)   # nu
```

```
Out[10]: 3035.2796326007801
```

cashflows

Financial investment modeling and advanced engineering economics using Python.

The library can be installed as usual using `pip`:

```
pip install cashflows
```

Click here to access the last documentation of the package.

https://github.com/jdvelasq/cashflows

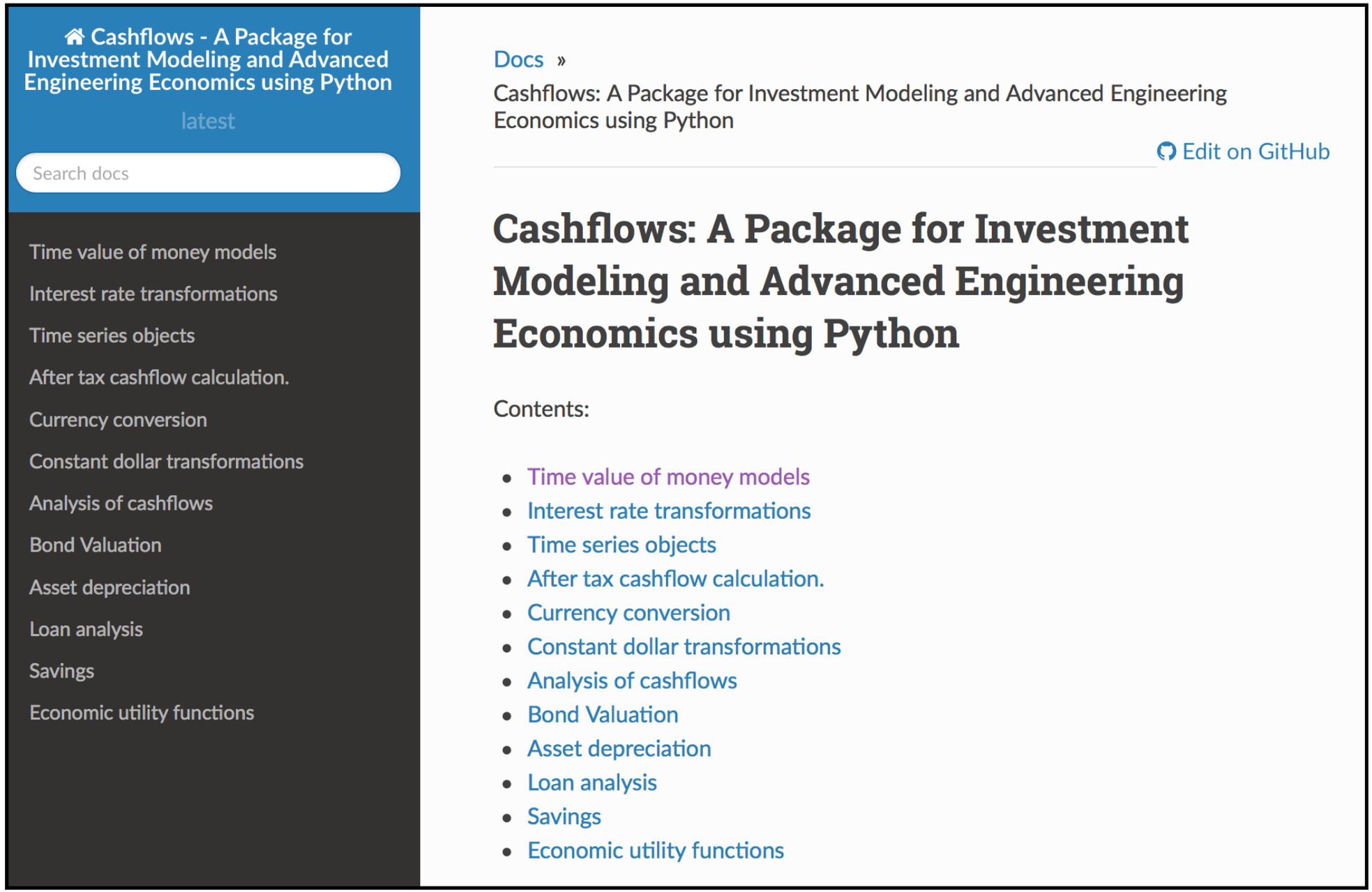
Programación

Programación en la máquina local

Testing automático usando unittests y doctests

Control de versiones en GitHub

Documentación automática con Sphinx



Programación

Computación reproducible con conda

Conda user cheat sheet

CONTINUUM ANALYTICS

Take a conda test drive at bit.ly/tryconda

For full documentation of any command, type the command followed by **`--help`**.

```
conda create --help
```

TIP: *Many options after two dashes (--) have shortcuts.*

```
conda create --help  or  conda create -h
```

Managing conda and anaconda

`conda info`	Verify conda is installed, check version #
`conda update conda`	Update conda package and environment manager to current version
`conda update anaconda`	Update the anaconda meta package (the library of packages ready to install with **`conda`** command)

Managing environments

`conda info --envs` *or* **`conda info -e`**	Get a list of all my environments, active environment shown with *
`conda create --name snowflakes biopython` *or* **`conda create -n snowflakes biopython`**	Create an environment and install program(s) ***TIP:*** *To avoid dependency conflicts, install all programs in the environment (snowflakes) at the same time.* ***TIP:*** *Environments install by default into the envs directory in your conda directory. You can specify a different path; see* **`conda create --help`** *for details.*
`source activate snowflakes` *(Linux, OS X)* **`activate snowflakes`** *(Windows)*	Activate the new environment to use it ***TIP:*** *Activate prepends the path to the snowflakes environment.*
`conda create -n bunnies python=3.4 astroid`	Create a new environment, specify Python version

Adquisición de datos

Archivos y servicios Web
- JSON, CSV, XLS
- Formato Científico & GIS formats
- Open source
  - Pandas
  - HDF5 / NetCDF4
  - Blaze

Big Data & Spark
- Blaze ecosystem
- pyspark

NoSQL
- Blaze
- Odo
- pymongo

DW & SQL
- Blaze
- SQLAlchemy
- pyODBC
- mxODBC

Archivos delimitados por caracteres con Pandas

Contenido

> pandas.DataFrame.to_csv
> pandas.DataFrame.from_csv
> pandas.to_csv
> pandas.read_csv

In [61]:
```
## escribe el archivo
df.to_csv('files/data.csv',   # el nombre del archivo
          index = False)      # imprime los nombres de las filas?

## verifica el archivo creado
print(open('files/data.csv', 'r').read())
```

```
index,name,value
1,A,3.03
2,B,5.14
3,C,0.4
4,D,1.13
5,E,8.25
```

In [62]:
```
pandas.read_csv('files/data.csv')
```

Out[62]:

	index	name	value
0	1	A	3.03
1	2	B	5.14
2	3	C	0.40
3	4	D	1.13
4	5	E	8.25

Adquisición de datos

Blaze

HOME OVERVIEW PROJECTS TALKS BLOG

Sponsored by: CONTINUUM ANALYTICS

The Blaze Ecosystem

The Blaze ecosystem is a set of libraries that help users store, describe, query and process data. It is composed of the following core projects:

- Blaze: An interface to query data on different storage systems
- Dask: Parallel computing through task scheduling and blocked algorithms
- Datashape: A data description language
- DyND: A C++ library for dynamic, multidimensional arrays
- Libndtypes: A C/C++ library for a low-level version of Datashape
- Ndtypes-python: Python bindings for libndtypes
- Odo: Data migration between different storage systems

Adquisición de datos

Blaze

Combining separate, gzipped csv files.

```
>>> from blaze import odo
>>> from pandas import DataFrame
>>> odo(example('accounts_*.csv.gz'), DataFrame)
   id      name  amount
0   1     Alice     100
1   2       Bob     200
2   3   Charlie     300
3   4       Dan     400
4   5     Edith     500
```

http://blaze.readthedocs.io/en/latest/queries.html#examples

Adquisición de datos

Blaze

Split-Apply-Combine

```
>>> from blaze import data, by
>>> t = data('sqlite:///%s::iris' % example('iris.db'))
>>> t.peek()
    sepal_length  sepal_width  petal_length  petal_width      species
0            5.1          3.5           1.4          0.2  Iris-setosa
1            4.9          3.0           1.4          0.2  Iris-setosa
2            4.7          3.2           1.3          0.2  Iris-setosa
3            4.6          3.1           1.5          0.2  Iris-setosa
4            5.0          3.6           1.4          0.2  Iris-setosa
5            5.4          3.9           1.7          0.4  Iris-setosa
6            4.6          3.4           1.4          0.3  Iris-setosa
7            5.0          3.4           1.5          0.2  Iris-setosa
8            4.4          2.9           1.4          0.2  Iris-setosa
9            4.9          3.1           1.5          0.1  Iris-setosa
...
>>> by(t.species, max=t.petal_length.max(), min=t.petal_length.min())
           species  max  min
0      Iris-setosa  1.9  1.0
1  Iris-versicolor  5.1  3.0
2   Iris-virginica  6.9  4.5
```

http://blaze.readthedocs.io/en/latest/queries.html#examples

Adquisición de datos

Odo

Overview

Odo migrates between many formats. These include in-memory structures like `list`, `pd.DataFrame` and `np.ndarray` and also data outside of Python like CSV/JSON/HDF5 files, SQL databases, data on remote machines, and the Hadoop File System.

The `odo` function

`odo` takes two arguments, a source and a target for a data transfer.

```
>>> from odo import odo
>>> odo(source, target)  # load source into target
```

It efficiently migrates data from the source to the target.

http://odo.readthedocs.io/en/latest/overview.html

Adquisición de datos

SQLite

```python
import sqlite3
conn = sqlite3.connect('example.db')

c = conn.cursor()
c.execute('''
          CREATE TABLE person
          (id INTEGER PRIMARY KEY ASC, name varchar(250) NOT NULL)
          ''')
c.execute('''
          CREATE TABLE address
          (id INTEGER PRIMARY KEY ASC, street_name varchar(250), street_number varchar(250),
           post_code varchar(250) NOT NULL, person_id INTEGER NOT NULL,
           FOREIGN KEY(person_id) REFERENCES person(id))
          ''')

c.execute('''
          INSERT INTO person VALUES(1, 'pythoncentral')
          ''')
c.execute('''
          INSERT INTO address VALUES(1, 'python road', '1', '00000', 1)
          ''')

conn.commit()
conn.close()
```

```python
import sqlite3
conn = sqlite3.connect('example.db')

c = conn.cursor()
c.execute('SELECT * FROM person')
print c.fetchall()
c.execute('SELECT * FROM address')
print c.fetchall()
conn.close()
```

http://pythoncentral.io/introductory-tutorial-python-sqlalchemy/

Adquisición de datos

SQLAlchemy

```python
import os
import sys
from sqlalchemy import Column, ForeignKey, Integer, String
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import relationship
from sqlalchemy import create_engine

Base = declarative_base()

class Person(Base):
    __tablename__ = 'person'
    # Here we define columns for the table person
    # Notice that each column is also a normal Python instance attribute.
    id = Column(Integer, primary_key=True)
    name = Column(String(250), nullable=False)

class Address(Base):
    __tablename__ = 'address'
    # Here we define columns for the table address.
    # Notice that each column is also a normal Python instance attribute.
    id = Column(Integer, primary_key=True)
    street_name = Column(String(250))
    street_number = Column(String(250))
    post_code = Column(String(250), nullable=False)
    person_id = Column(Integer, ForeignKey('person.id'))
    person = relationship(Person)

# Create an engine that stores data in the local directory's
# sqlalchemy_example.db file.
engine = create_engine('sqlite:///sqlalchemy_example.db')

# Create all tables in the engine. This is equivalent to "Create Table"
# statements in raw SQL.
Base.metadata.create_all(engine)
```

http://pythoncentral.io/introductory-tutorial-python-sqlalchemy/

Hardware

Numba — https://numba.pydata.org

ipyparallel — https://github.com/ipython/ipyparallel

mpi4py — http://pythonhosted.org/mpi4py/

Theano — http://deeplearning.net/software/theano/

pyCUDA —https://mathema.tician.de/software/pycuda/

```python
from numba import jit
from numpy import arange

# jit decorator tells Numba to compile this function.
# The argument types will be inferred by Numba when function is called.
@jit
def sum2d(arr):
    M, N = arr.shape
    result = 0.0
    for i in range(M):
        for j in range(N):
            result += arr[i,j]
    return result

a = arange(9).reshape(3,3)
print(sum2d(a))
```

https://numba.pydata.org

Analytics

Preparación de datos
- Pandas
- Blaze
- GeoPandas
- R plyr, R dplyr, R tidyr, Rreshape2, …

Estadística
- SciPy
- PyMC
- StatsModels

Machine Learning & Deep Learning
- Scikit-learn
- NLTK
- NetworkX
- Theano
- pycaffe
- Pylearn2
- R caret, R glmnet, R randomForest

Simulación y optimización
- SimPy
- PyJMI
- PyFMI
- PyMC
- Pyomo
- CVXOPT
- CVXPY
- tao4py
- pyopt
- Pylpopt
- PyGMO

Analytics

StatsModels

```
In [1]: import numpy as np

In [2]: import statsmodels.api as sm

In [3]: import statsmodels.formula.api as smf

# Load data
In [4]: dat = sm.datasets.get_rdataset("Guerry", "HistData").data

# Fit regression model (using the natural log of one of the regressors)
In [5]: results = smf.ols('Lottery ~ Literacy + np.log(Pop1831)', data=dat).fit()

# Inspect the results
In [6]: print(results.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:                Lottery   R-squared:                       0.348
Model:                            OLS   Adj. R-squared:                  0.333
Method:                 Least Squares   F-statistic:                     22.20
Date:                Tue, 28 Feb 2017   Prob (F-statistic):           1.90e-08
Time:                        21:38:05   Log-Likelihood:                -379.82
No. Observations:                  86   AIC:                             765.6
Df Residuals:                      83   BIC:                             773.0
Df Model:                           2
Covariance Type:            nonrobust
```

http://www.statsmodels.org/stable/index.html

Analytics

Pyomo

PYOMO

HOME / ABOUT / DOWNLOAD / DOCUMENTATION / BLOG

Documentation

Online Documentation

Pyomo Online Documentation (html, pdf, epub)
PySP Online Documentation (pdf)
Pyomo Wikipedia Page (html)

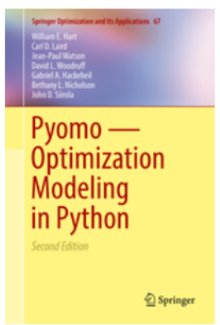
Examples

Pyomo Gallery (browse)
Online examples from the Pyomo software repository: (browse) (zipfile)

Citation

If you use Pyomo for your work, please cite the Pyomo book (bibtex) and the Pyomo paper (bibtex).
If you use PySP for your work, please cite the PySP paper (bibtex).

The Pyomo Book



Hart, William E., Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John D. Siirola. *Pyomo – Optimization Modeling in Python*. Second Edition. Vol. 67. Springer, 2017.

The Second Edition of the book describes capabilities in the Pyomo 5.x series. The First Edition (2012) describes the capabilities from the Coopr 3.1 release. Some changes beginning in the Pyomo 4.0 release are not backwards compatible with the First Edition.

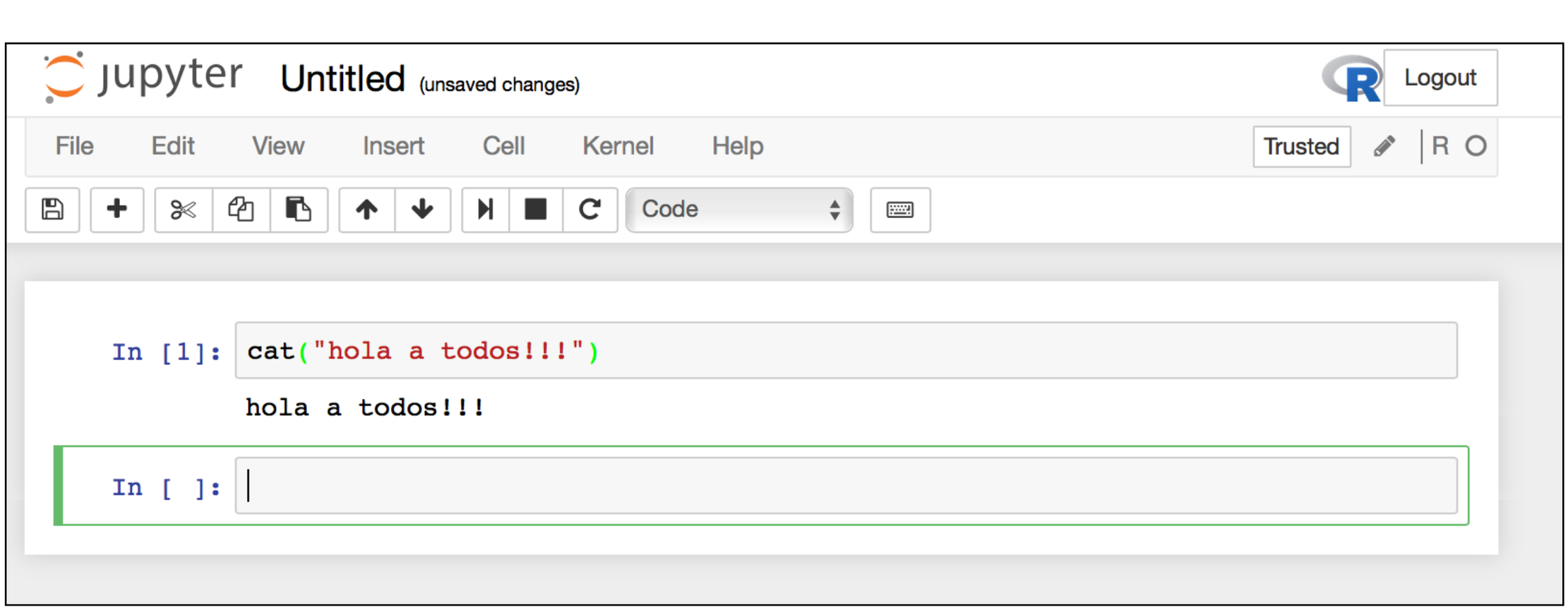
http://www.pyomo.org/documentation

Storyboards

Notebooks
- Jupyter / IPython
- R Jupyter
- R Markdown

Data IDEs
- R Studio
- Spyder

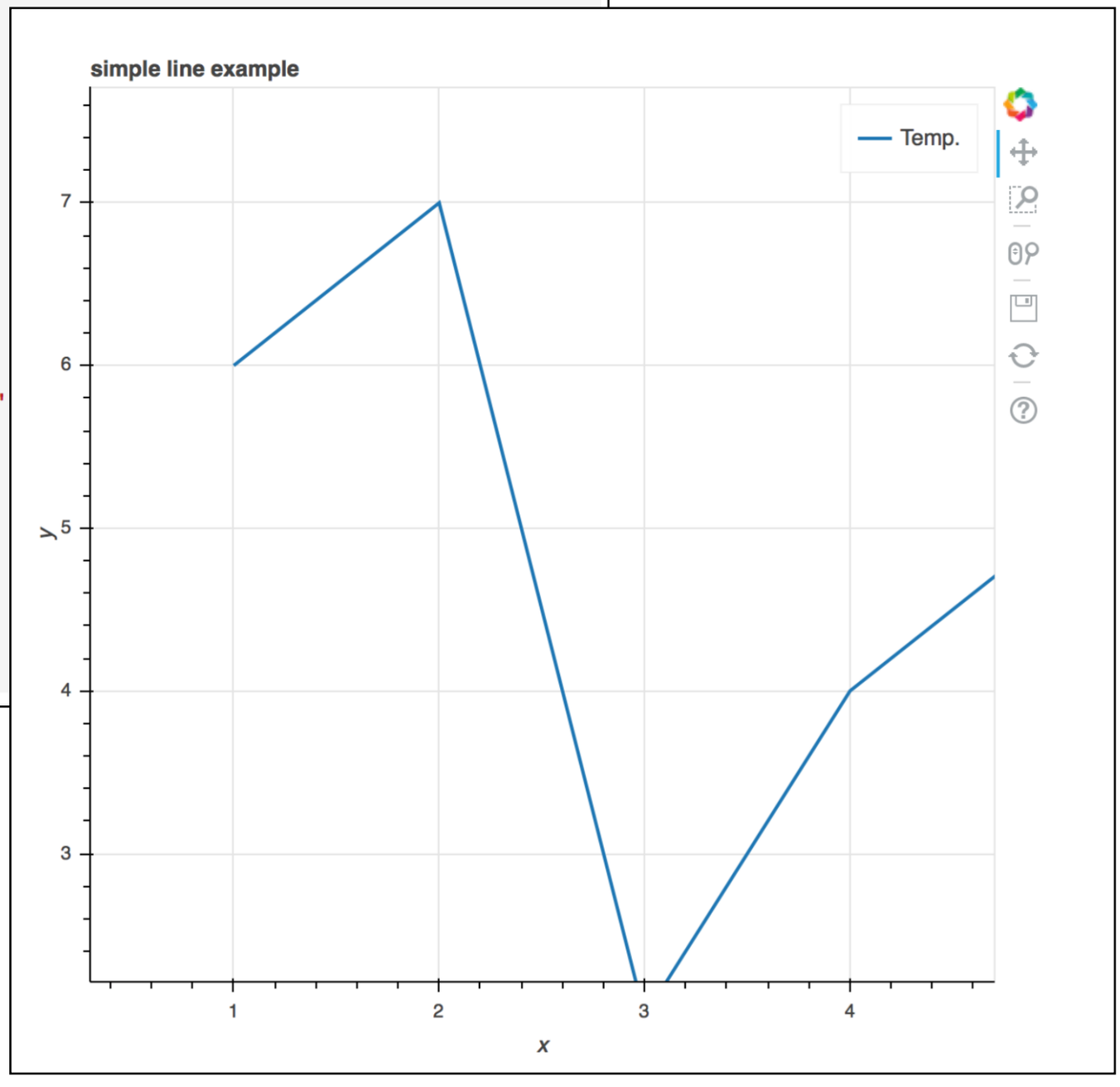
Exploración interactiva
- Bokeh
- Jupyter
- R Shiny



Storyboards

Bokeh

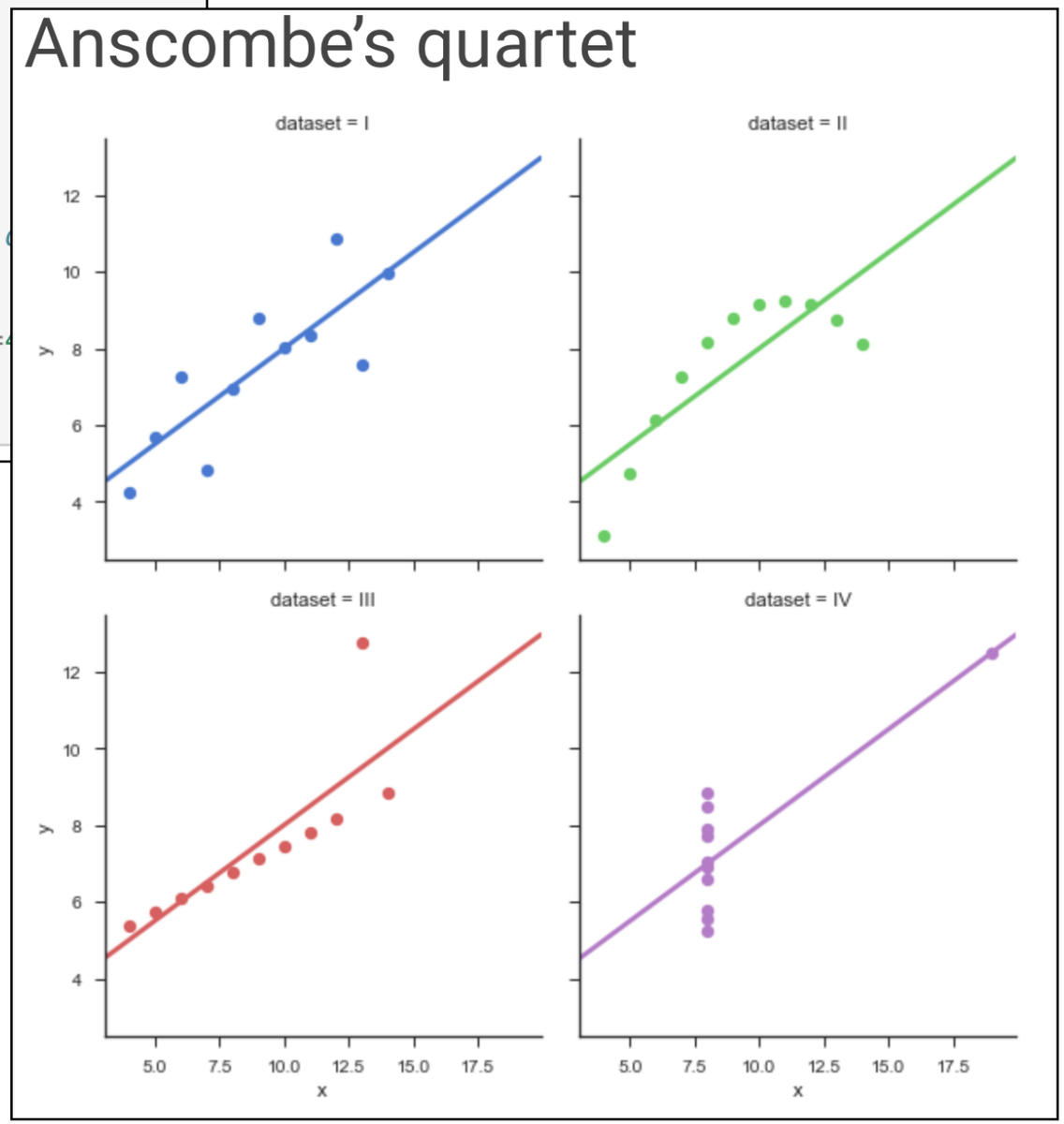
```
from bokeh.plotting import figure, output_file, show

# prepare some data
x = [1, 2, 3, 4, 5]
y = [6, 7, 2, 4, 5]

# output to static HTML file
output_file("lines.html")

# create a new plot with a title and axis labels
p = figure(title="simple line example", x_axis_label='x'

# add a line renderer with legend and line thickness
p.line(x, y, legend="Temp.", line_width=2)

# show the results
show(p)
```



Visualización

- Bokeh
- matplolib
- seaborn
- R ggplot2
- PyQTgraph
- Veusz
- NetworkX

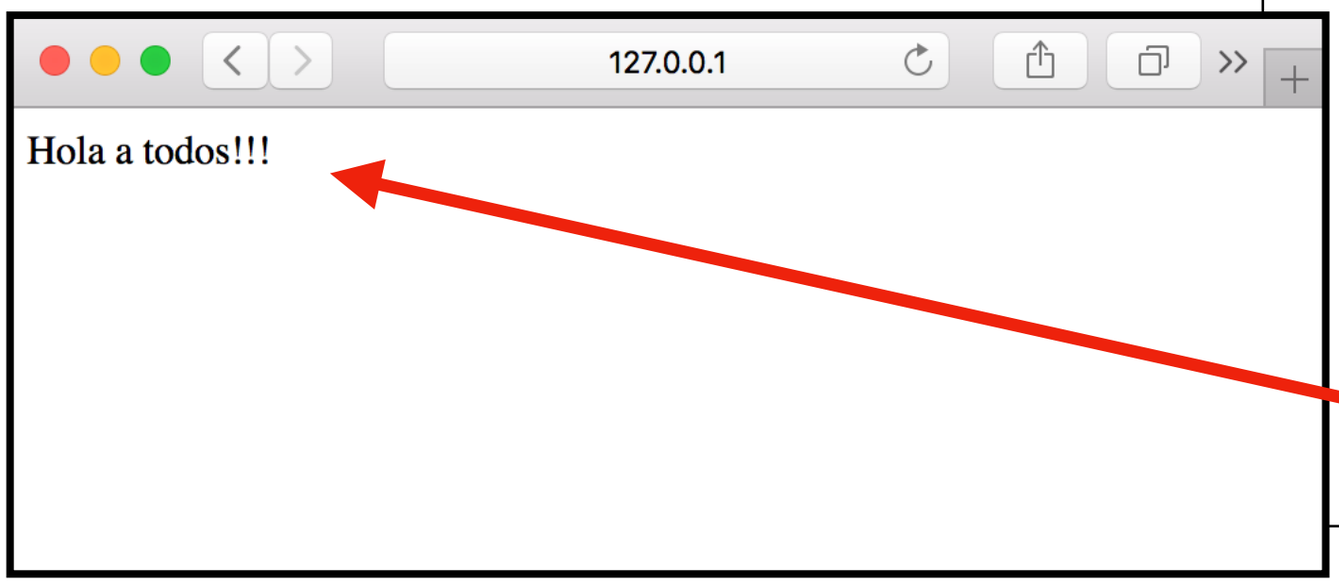
```
import seaborn as sns
sns.set(style="ticks")

# Load the example dataset for Anscombe's quartet
df = sns.load_dataset("anscombe")

# Show the results of a linear regression within each
sns.lmplot(x="x", y="y", col="dataset", hue="dataset",
           col_wrap=2, ci=None, palette="muted", size=
           scatter_kws={"s": 50, "alpha": 1})
```



https://seaborn.pydata.org/examples/anscombes_quartet.html

Interfases de usuario

- **Flask**
- **Pyforms**
- **Pyramid**
- **PyQT**
- **Tkinter**
- **WxPython**



http://flask.pocoo.org

Interfases de usuario

Pyforms

Create the Python file that will store your applications.

Example: **SimpleExample.py**

Import the library.

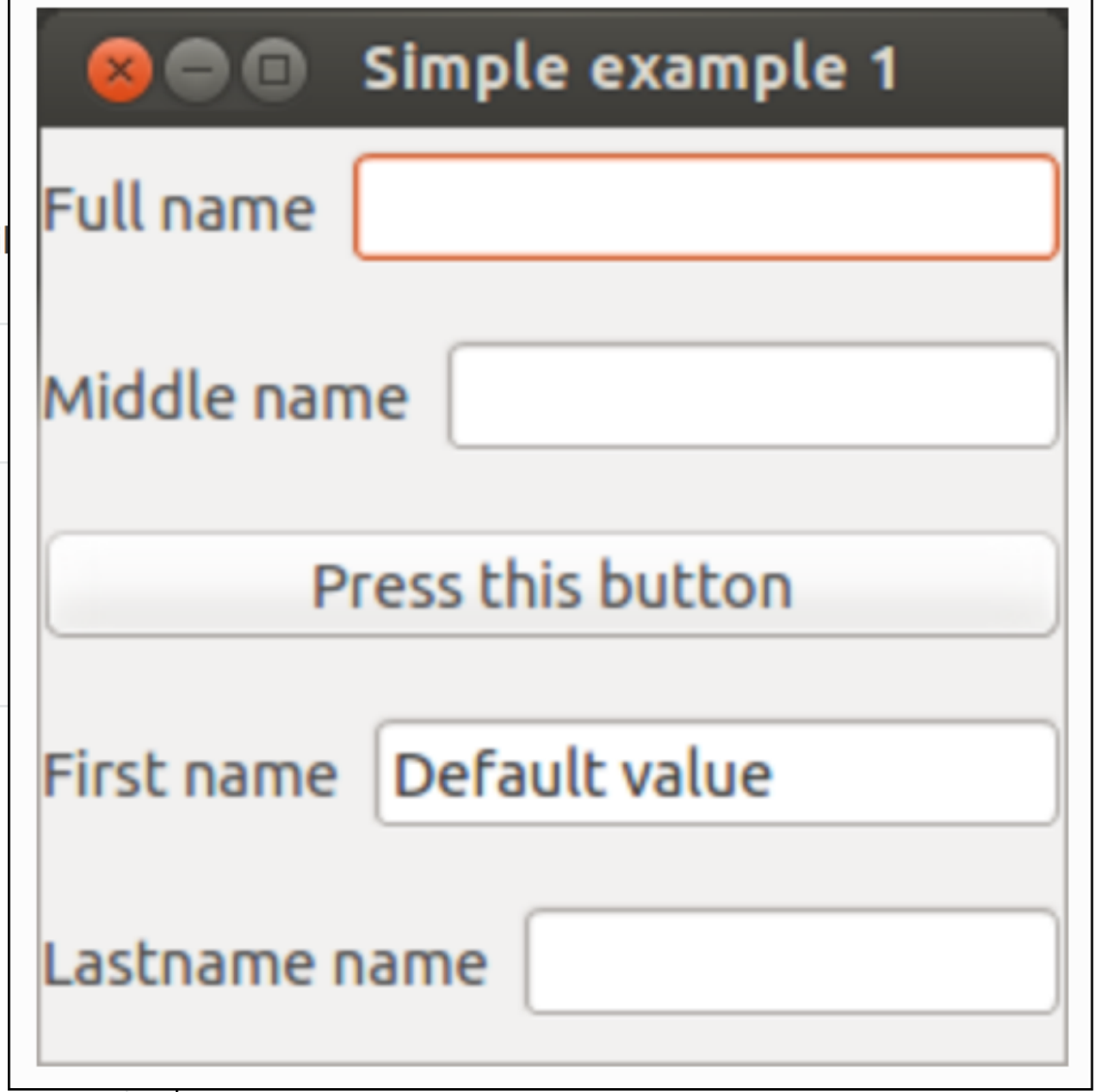
Import the pyforms library, the BaseWidget and the Controls classes that you wi

```
import pyforms
from   pyforms          import BaseWidget
from   pyforms.Controls import ControlText
from   pyforms.Controls import ControlButton
```

Create your application class.

This class should inherit from the class BaseWidget.

```
class SimpleExample1(BaseWidget):

    def __init__(self):
        super(SimpleExample1,self).__init__('Simple example 1')

        #Definition of the forms fields
        self._firstname     = ControlText('First name', 'Default value')
        self._middlename    = ControlText('Middle name')
        self._lastname      = ControlText('Lastname name')
        self._fullname      = ControlText('Full name')
        self._button        = ControlButton('Press this button')


#Execute the application
if __name__ == "__main__":   pyforms.start_app( SimpleExample1 )
```



http://pyforms.readthedocs.io/en/v2.0/getting-started/the-basic/

II Congreso Colombiano de
Investigación Operativa
ASOCIO 2017

Una introducción a Analytics con Anaconda Python

JUAN DAVID VELÁSQUEZ HENAO, MSc, PhD
Profesor Titular
Departamento de Ciencias de la Computación y la Decisión
Facultad de Minas
Universidad Nacional de Colombia, Sede Medellín

jdvelasq@unal.edu.co
@jdvelasquezh
https://github.com/jdvelasq
https://goo.gl/prkjAq
https://goo.gl/vXH8jy