ANALYTIXLABS

Introduction to Data Science

Introduction to Data Science

ANALYTIXLABS

What is Data Science?

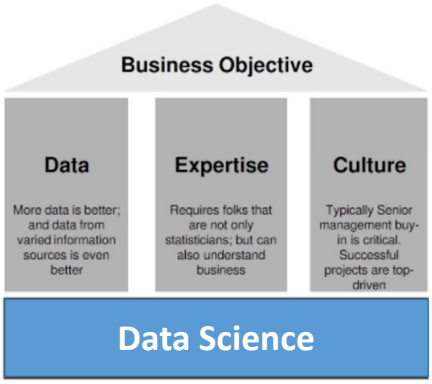
> *"To gain insights into data through computation, statistics, and visualization."*

Quora Threads for Expert Definitions

- What is Data Science?
- What does a Data Scientist do?

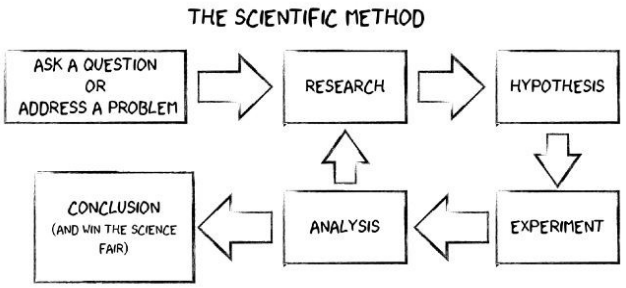
Data Science is Process

- ✓ **Ask an interesting question**
- ✓ **Get the data**
- ✓ **Explore the data**
- ✓ **Model the data**
- ✓ **Communicate and visualize your results**



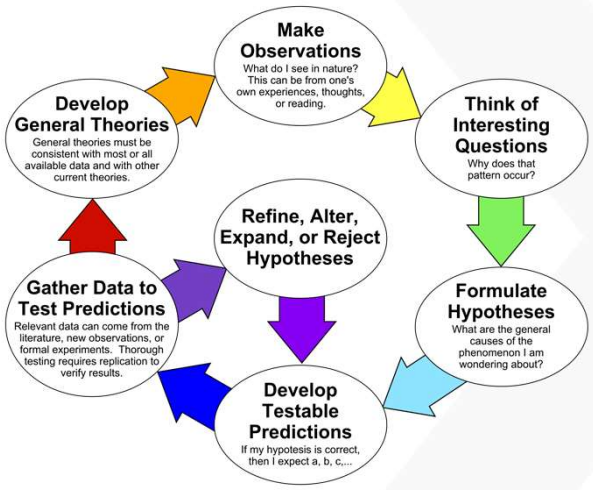
Data Science is Multidisciplinary

- The Scientific Method (wiki)
- Programming
- Databases
- Statistics
- Machine Learning
- Domain Knowledge



ANALYTIXLABS

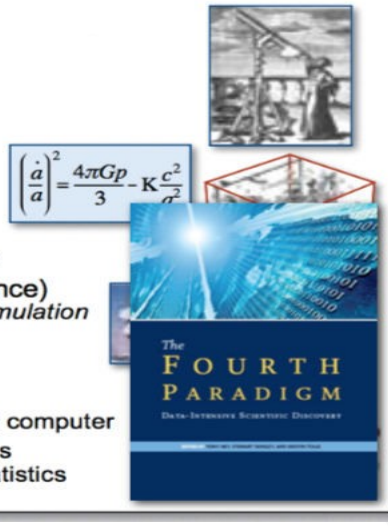
Data Science is Multidisciplinary



ANALYTIXLABS

Science Paradigm

- Thousand years ago: science was **empirical**
  describing natural phenomena
- Last few hundred years: **theoretical** branch
  using models, generalizations
- Last few decades: a **computational** branch
  simulating complex phenomena
- Today: **data exploration** (eScience)
  unify theory, experiment, and simulation
  - Data captured by instruments or generated by simulator
  - Processed by software
  - Information/knowledge stored in computer
  - Scientist analyzes database/files using data management and statistics



ANALYTIXLABS

Why Data Science?

• The ability to take **data** – to be able to **understand** it, to **process** it, to **extract value** from it, to **visualize** it, to **communicate** it's going to be a hugely important skill in the next decades, not only at the professional level but even at the educational level for elementary school kids, for high school kids, for college kids. Because now we really do have essentially free and **ubiquitous data**."

• – Hal Varian

ANALYTIXLABS

Who is Data Scientist?

"A data scientist… excels at **analyzing data**, particularly large amounts of data, to help a business gain a competitive edge."

"The analysis of data using the **scientific method**"

"A data scientist is an individual, organization or application that performs statistical analysis, data mining and retrieval processes on a large amount of data to **identify trends, figures and other relevant information**."

ANALYTIXLABS

WHO'S A DATA SCIENTIST

- "A data scientist is someone who knows more statistics than a computer scientist and more computer science than a statistician."

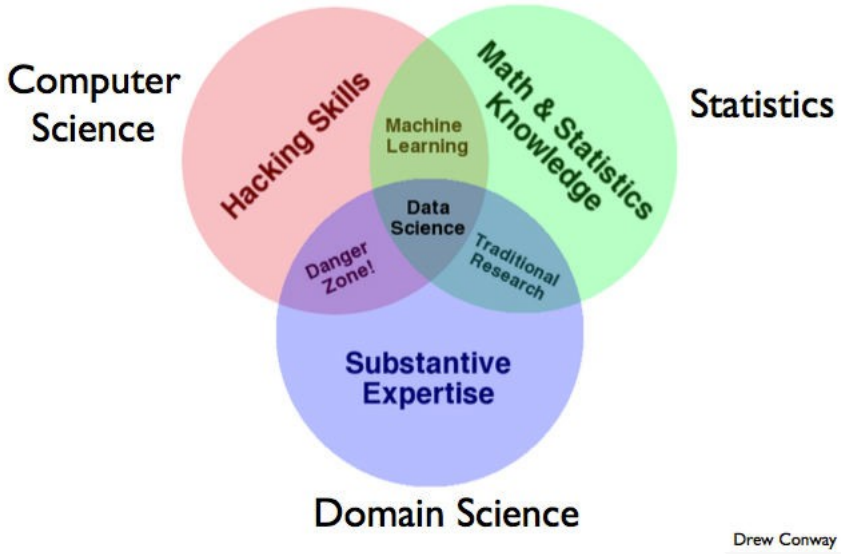
  - Josh Blumenstock

"Data Scientist = statistician + programmer + coach + storyteller + artist"

- Shlomo Aragmon

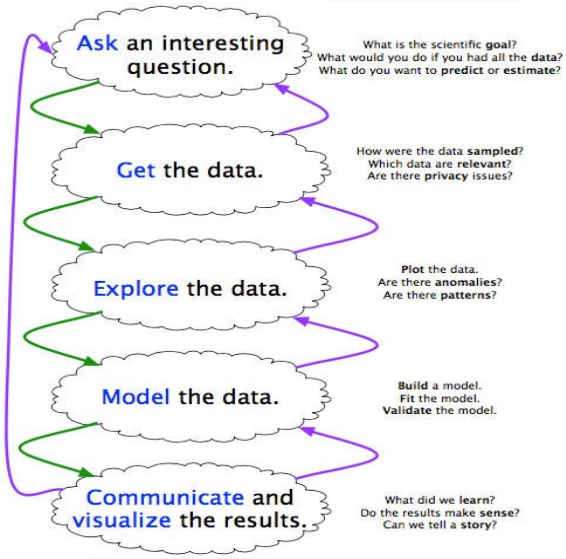
ANALYTIXLABS

WHO'S A DATA SCIENTIST



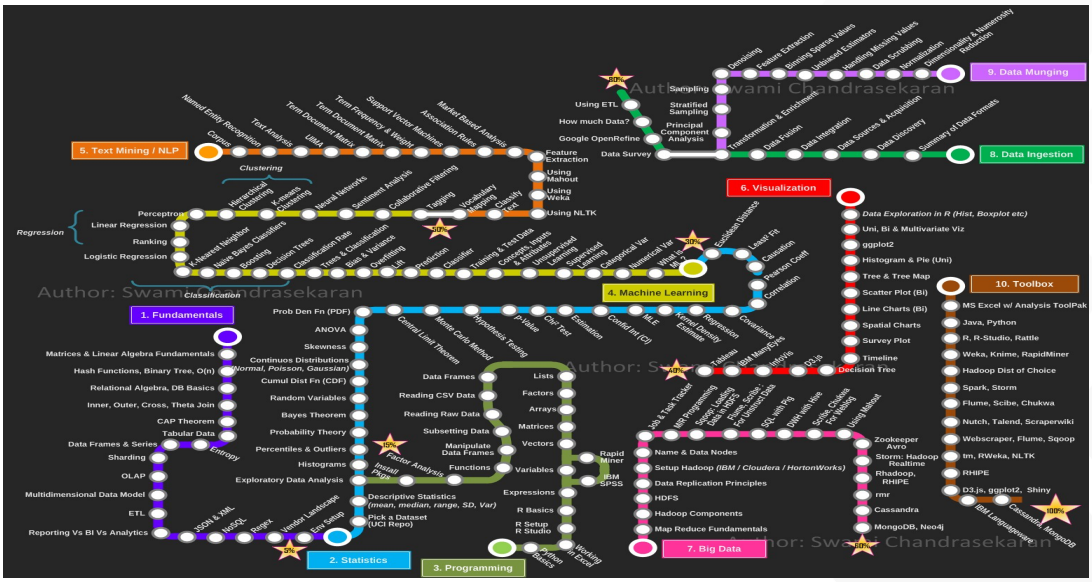
ANALYTIXLABS

Who is Data Scientist?



ANALYTIXLABS

Who's a Data Scientist?



ANALYTIXLABS

What does a Data Scientist Do?

BUILD DATA PRODUCTS

tools built with data to inform decision making

DESCRIPTIVE
PREDICTIVE
PRESCRIPTIVE

OSEMN Things!

Obtain data

Scrub data

Explore data

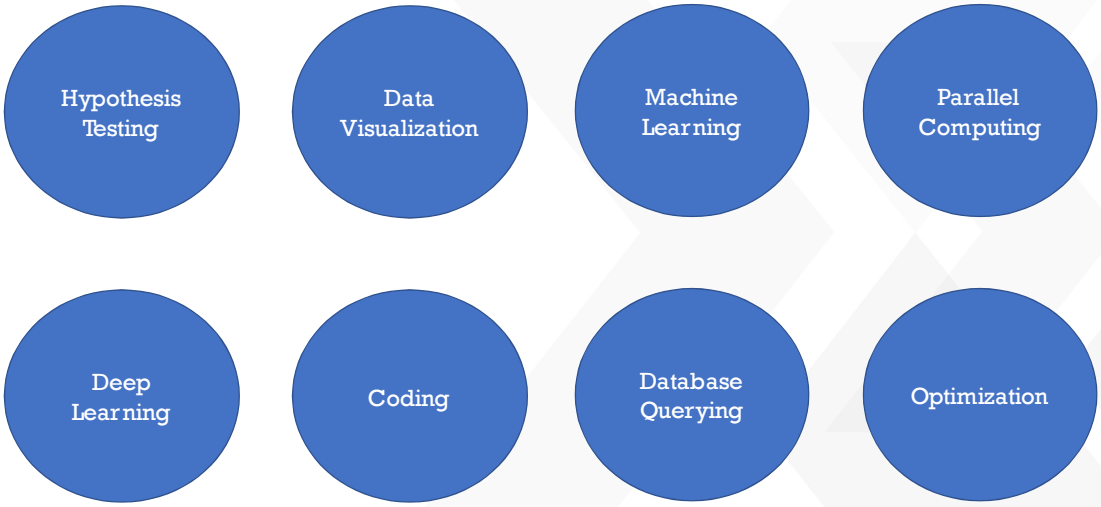
Build Models

iNterpret results

Hence the acronym
O-S-E-M-N
(pronounced, 'awesome')

ANALYTIXLABS

.. And This



ANALYTIXLABS

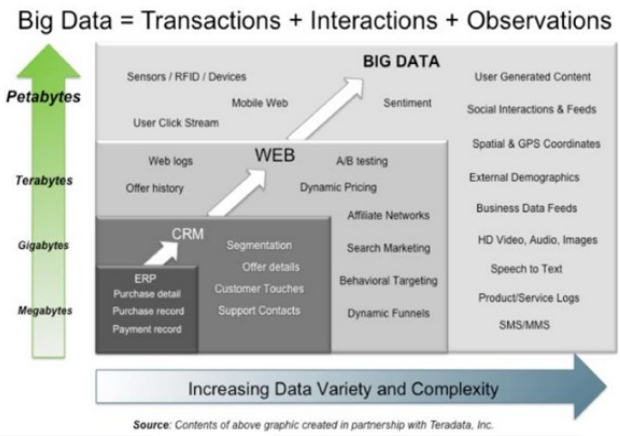
Key Concepts

- *use many data sources*
- *understand how the data were collected* (sampling is essential)
- *weight the data thoughtfully* (not all polls are equally good)
- *use statistical models* (not just hacking around in Excel)
- *understand correlations* (e.g., states that trend similarly)
- *think like a Bayesian, check like a frequentist* (reconciliation)
- *have good communication skills* (What does a 60% probability even mean?
- *visualize, validate, and understand the conclusions*

ANALYTIXLABS

Common Challenges

- *Big (massive) data* (millions of users, billions of events)
- *curse of dimensionality* (hundreds of variables)
- *missing data* (*not* missing at random)
- *need to avoid overfitting* (test data vs. training data)



ANALYTIXLABS

Common Tasks

- data munging/scraping/sampling/cleaning in order to get an informative, manageable data set;
- data storage and management in order to be able to access data quickly and reliably during subsequent analysis;
- exploratory data analysis to generate hypotheses and intuition about the data;
- prediction based on statistical tools such as regression, classification, clustering, forecasting and optimization; and
- communication of results through visualization, stories, and interpretable summaries.

ANALYTIXLABS

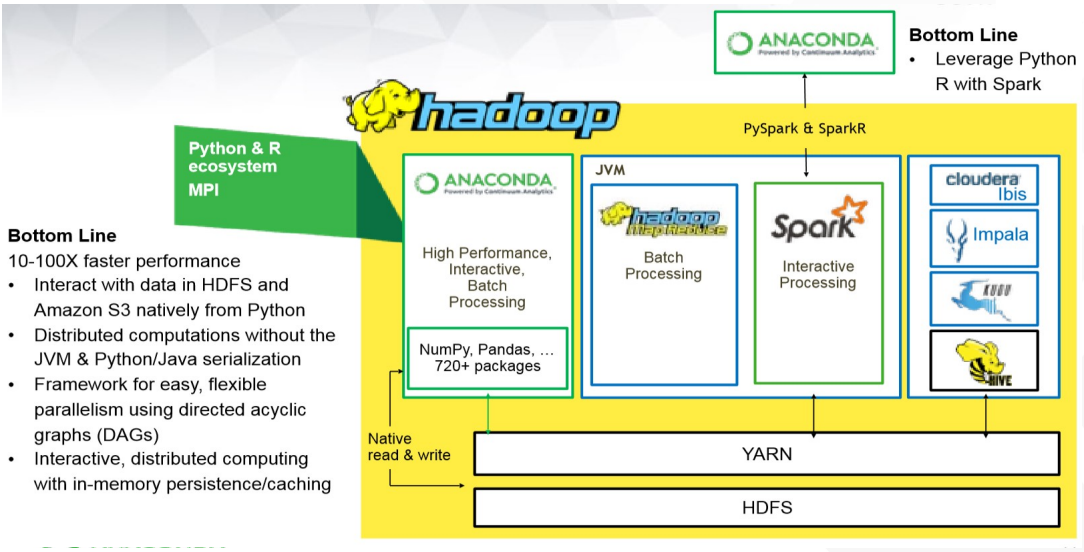
Tools for the course

ANALYTIXLABS

Tools for this course

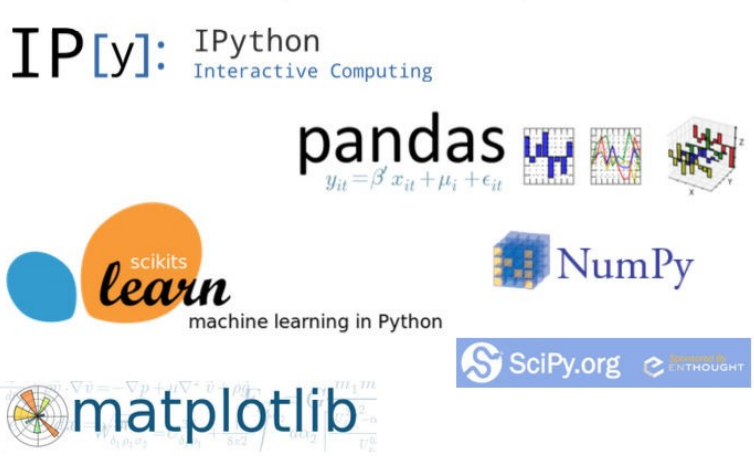
ANALYTIXLABS

Tools for the course



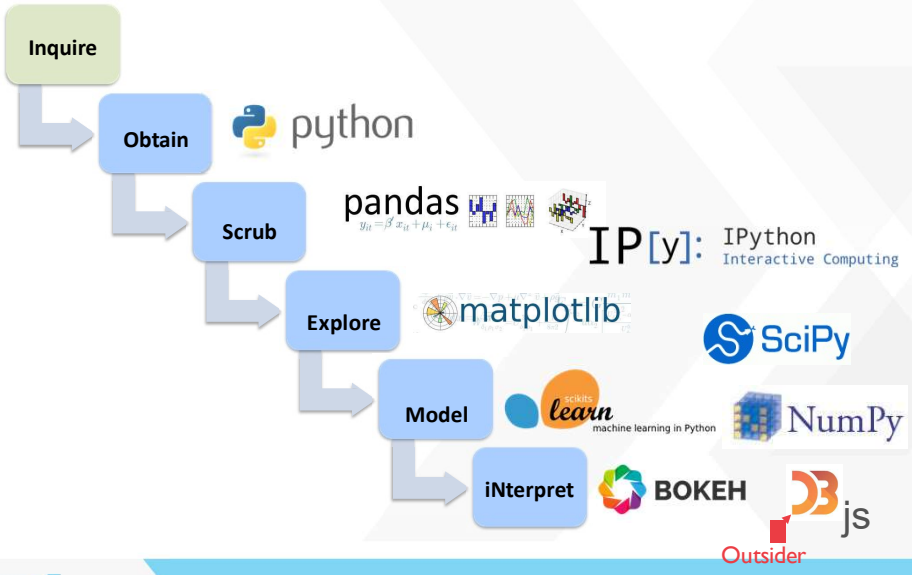
ANALYTIXLABS

Tools for the course - Python



ANALYTIXLABS

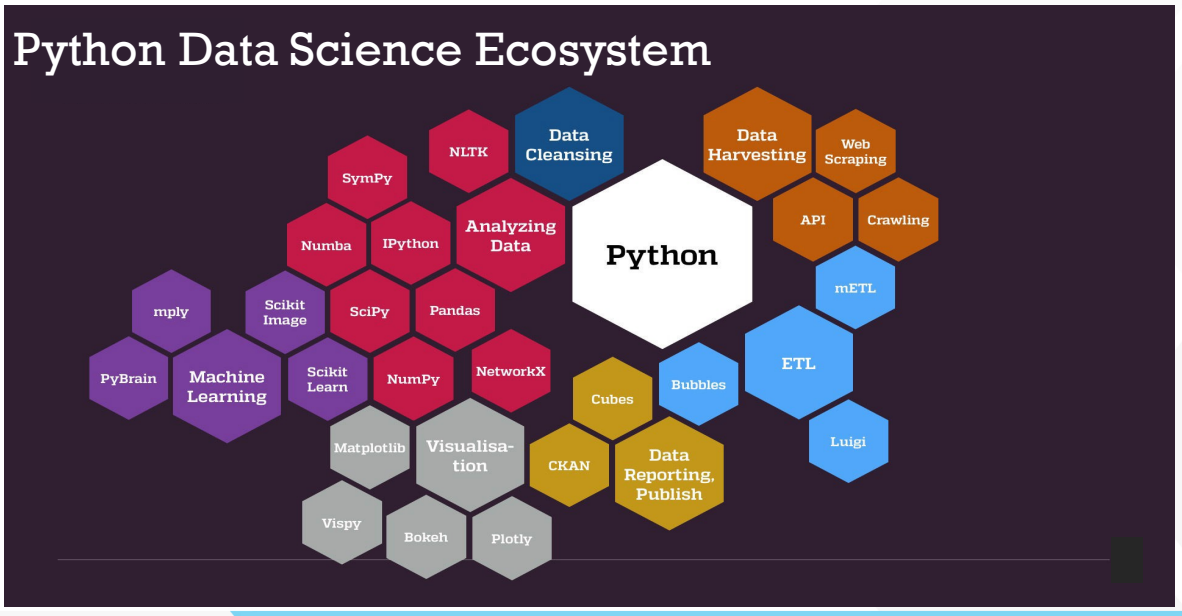
Python Is IOSEMN



ANALYTIXLABS

Python Data Science Ecosystem

ANALYTIXLABS



ANALYTIXLABS

Packages - Data Manipulation

- NumPy — Low level array operations
- pandas — Data tables and in-memory manipulation
- Dask — Parallel out-of-core array manipulation
- Blaze — High level interface for databases and different computational backends

Packages - Visualisation

- matplotlib — Widely used and powerful plotting package
- seaborn — Opinionated but beautiful data visualisations
- Bokeh — Interactive plotting with server option
- plotly — Graphics API with translation between languages (e.g. Python -> D3)

Packages - Modelling

- SciPy — FFTs, integration, other general algorithms
- StatsModels — Statistical distributions and tests
- scikit learn — Machine Learning pipelines
- PyMC3 — Bayesian Probabilistic Programming

IPython Notebooks

IP[y]: IPython Interactive Computing

jupyter

ANALYTIXLABS

Packages - Description

- **NumPy**

  NumPy is a low level library written in C (and FORTRAN) for high level mathematical functions. NumPy cleverly overcomes the problem of running slower algorithms on Python by using multidimensional arrays and functions that operate on arrays. Any algorithm can then be expressed as a function on arrays, allowing the algorithms to be run quickly.

  NumPy is part of the SciPy project, and is released as a separate library so people who only need the basic requirements can use it without installing the rest of SciPy.

  NumPy is compatible with Python versions 2.4 through to 2.7.2 and 3.1+

- **SciPy**

  SciPy is a library that uses NumPy for more mathematical functions. SciPy uses NumPy arrays as the basic data structure, and comes with modules for various commonly used tasks in scientific programming, including linear algebra, integration (calculus), ordinary differential equation solving and signal processing.

- **Numba**

  Numba is a NumPy aware Python compiler (just-in-time (JIT) specializing compiler) which compiles annotated Python (and NumPy) code to LLVM (Low Level Virtual Machine) through special decorators. Briefly, Numba uses a system that compiles Python code with LLVM to code which can be natively executed at runtime.

Packages - Description

- **scikit-learn**

  scikit-learn is a Python module for machine learning built on top of SciPy and distributed under the 3-Clause BSD license.

- **Pandas**

  Pandas is data manipulation library based on Numpy which provides many useful functions for accessing, indexing, merging and grouping data easily. The main data structure (DataFrame) is close to what could be found in the R statistical package; that is, heterogeneous data tables with name indexing, time series operations and auto-alignment of data.

- **Matplotlib**

  Matplotlib is a flexible plotting library for creating interactive 2D and 3D plots that can also be saved as manuscript-quality figures. The API in many ways reflects that of MATLAB, easing transition of MATLAB users to Python. Many examples, along with the source code to re-create them, are available in the matplotlib gallery.

Packages - Description

- **Rpy2**

  Rpy2 is a Python binding for the R statistical package allowing the execution of R functions from Python and passing data back and forth between the two environments. Rpy2 is the object oriented implementation of the Rpy bindings.

- **PsycoPy**

  PsychoPy is a library for cognitive scientists allowing the creation of cognitive psychology and neuroscience experiments. The library handles presentation of stimuli, scripting of experimental design and data collection.

Packages - Description

- **datetime (or) time**
  Date and time functions to manage date and time data
- **math**
  Core math functions and the constants like pi, e etc.
- **pickle**
  Serializes objects to file
- **os (or) os.path**
  Operating system interfaces.
- **re**
  A library of perl-like regular expression operations
- **string**
  Useful constants and classes related to strings.
- **sys**
  System parameters and functions

Who is using Python?

Financial Services
- Risk Mgmt., Quant modeling, Data exploration and processing, algorithmic trading, compliance reporting

Government
- Fraud detection, data crawling, web & cyber data analytics, statistical modeling

Healthcare & Life Sciences
- Genomics data processing, cancer research, natural language processing for health data science

High Tech
- Customer behavior, recommendations, ad bidding, retargeting, social media analytics

Retail & CPG
- Engineering simulation, supply chain modeling, scientific analysis

Oil & Gas
- Pipeline monitoring, noise logging, seismic data processing, geophysics

ANALYTIXLABS

Why should I become a Data Scientist?

DEMAND & SUPPLY

"We project a need for 1.5 million additional managers and analysts in the United States who can ask the right questions and consume the results of the analysis of Big Data effectively."

"A significant constraint on realizing value from Big Data will be a shortage of talent, particularly of people with deep expertise in statistics and machine learning, and the managers and analysts who know how to operate companies by using insights from Big Data."

Big data: The next frontier for innovation, competition, and productivity, McKinsey report

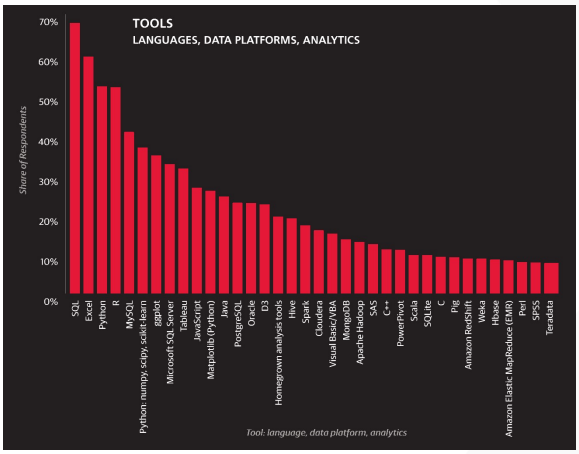
"By 2018 the United States will experience a shortage of 190,000 skilled data scientists, and 1.5 million managers and analysts capable of reaping actionable insights from the big data deluge."

Game changers: Five opportunities for US growth and renewal, McKinsey report

ANALYTIXLABS

25

OK. How so do I become a Data Scientist?



Read books on
- Statistics
- Machine Learning
- Programming
- Databases

Take University courses

Apply for internships to work on real-life projects

Spend hours debugging on StackOverflow

Participate in Data Hackathons/Data Driven competitions

What is Python?

- Programming language
- You write instructions to the computer
- Python “interpreter” runs those instructions

Why python?

- It's awesome and popular!
- Free and Open Source language.
- Readable syntax.
- Great for interactive work
- Easy to learn and has an active community.
- Large amount of libraries.
- High level, general purpose.
- Backed up with fast C & Fortran numerical libraries

Python - Applications

Python is a powerful multi-paradigm computer programming language. With Python, we can do many things. Below are some of the things that can be achieved using Python.

- ✓ **Systems Programming:** Python's built-in interfaces to operating-system services make it ideal for writing portable, maintainable system-administration tools and utilities (sometimes called shell tools). Python programs can search files and directory trees, etc.
- ✓ **GUIs:** Python's simplicity and rapid turnaround makes it a good match for graphical user interface programming on the desktop. Python comes with a standard object-oriented interface to the Tk GUI API called tkinter (Tkinter in 2.X) that allows Python programs to implement portable GUIs with a native look and feel.
- ✓ **Internet Scripting:** Python comes with standard Internet modules that allow Python programs to perform a wide variety of networking tasks in client and server modes.
- ✓ **Database Programming:** For traditional database demands, there are Python interfaces to all commonly used relational database systems like Sybase, Oracle, Informix, ODBC, MySQL, PostgreSQL, SQLite, and more.

Python - Applications

Rapid Prototyping: To Python programmers the components written in Python and C look the same. Because of this, it's possible to prototype systems in Python initially, and then move selected components to a compiled language such as C or C++ for delivery.

Numeric and Scientific Programming: Python is also heavily used in numeric programming, a domain that would not traditionally have been considered to be in the scope of scripting languages, but has grown to become one of Python's most compelling use cases.

Google makes extensive use of Python in its web search systems.

The other use cases are as follows:

- ✓ The popular YouTube video sharing service is largely written in Python.
- ✓ The Dropbox storage service codes, both its server and desktop client software, is primarily written in Python.
- ✓ The Raspberry Pi single-board computer promotes Python as its educational language.
- ✓ The widespread BitTorrent peer-to-peer file sharing system began its life as a Python program.
- ✓ Industrial Light & Magic, Pixar, and others use Python in their production of animated movies.
- ✓ Google's App Engine web development framework uses Python as an application language.

Is Python a Scripting Language?

- ✓ Python is a general-purpose programming language that is often applied in scripting roles. It is commonly defined as an object-oriented scripting language, a definition that blends support for OOP with an overall orientation toward scripting roles.
- ✓ A scripting language or script language is a programming language that supports scripts, programs written for a special run-time environment that can interpret (rather than compile) and automate the execution of tasks that could alternatively be executed one-by-one by a human operator. Python comes under in this category. So it is called a scripting language.
- ✓ Still, the term 'scripting' seems to have stuck to Python like glue. This may be because, people often use the word 'script' instead of 'program' to describe a Python code file.

How to get Python & Anaconda?

How to get Python?

There are two ways to get Python.

Base Python:

- ✓ You can download Python from the www.python.org/downloads.
- ✓ Once it down loaded, you can install the python
- ✓ Ensure that you have pip installed which is the package manager for Python and will enable you to easily install 3rd party packages that you'll need to perform data science tasks.

ANACONDA:

Free enterprise-ready cross platform Python distribution for large- scale data processing, predictive analytics, and scientific computing.

Download here http://docs.continuum.io/anaconda/install

Features

https://www.continuum.io/why-anaconda

We use ANACONDA for our sessions

Why ANACONDA?

Leading Open Data Science Platform powered by Python

Quickly Engage w/ Your Data

- 720+ Popular Packages
- Optimized & Compiled
- Free for Everyone
- Extensible via **conda** Package Manager
- Sandbox Packages & Libraries
- Cross-Platform - Windows, Linux, Mac
- Not just Python - over 230 R packages
- Foundation of our Enterprise Products

ANALYTIXLABS

Why ANACONDA?

Accelerating Adoption of Python for Enterprises

ANACONDA®

ENTERPRISE DATA INTEGRATION with optimized connectors & out-of-core processing	NumPy & Pandas	Numba	**PERFORMANCE** with compiled Python for lightning fast execution
COLLABORATIVE NOTEBOOKS with publication, authentication, & search	Jupyter/ IPython	Bokeh	**VISUAL APPS** for interactivity and streaming data
PARALLEL COMPUTING scaling up Python analytics on your cluster	Dask	Conda	**SECURE & ROBUST REPOSITORY** of data science libraries, scripts, & notebooks

ANALYTIXLABS

Python for big data

hadoop Streaming

PYDOOP

Pig UDFs in Jython

HADOOPY

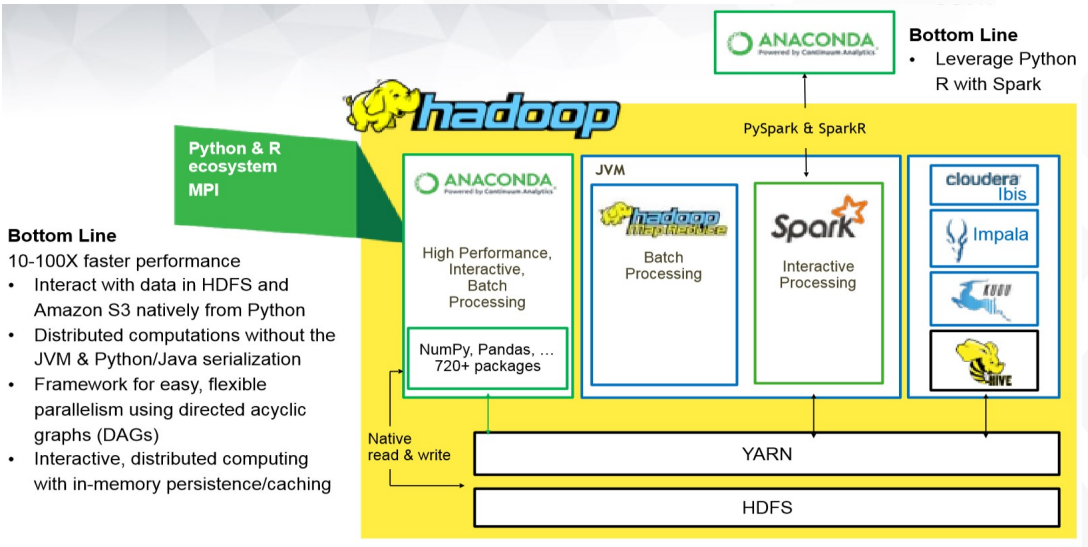
pySpark

Anaconda for Big Data

Bottom Line

10-100X faster performance

- Interact with data in HDFS and Amazon S3 natively from Python
- Distributed computations without the JVM & Python/Java serialization
- Framework for easy, flexible parallelism using directed acyclic graphs (DAGs)
- Interactive, distributed computing with in-memory persistence/caching

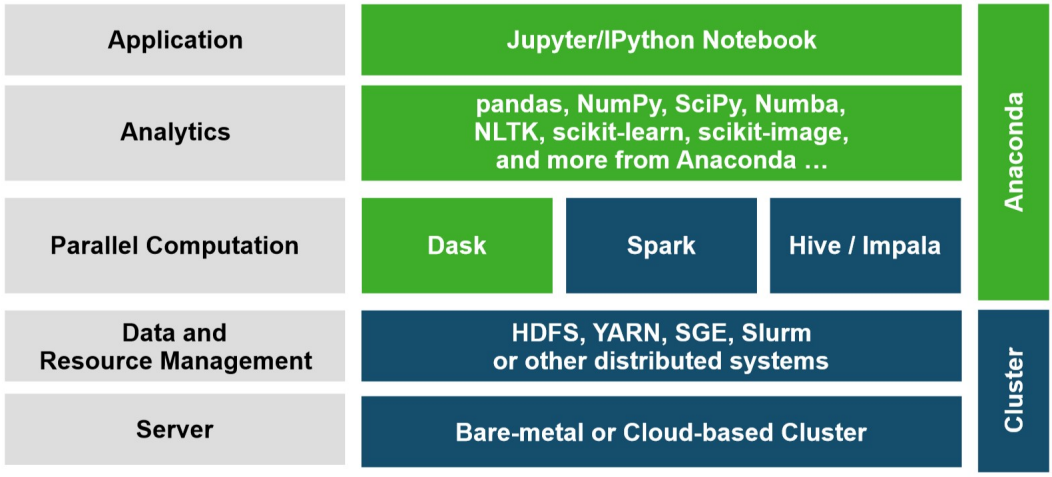


Remote Conda commands

Install packages	`conda install numpy scipy pandas numba`
Create environment	`conda create -n py34 python=3.4 numpy scipy pandas`
List packages	`conda list`
Conda information	`conda info`
Push environment	`conda push environment.yml`

ANALYTIXLABS

Anaconda for Analytics



ANALYTIXLABS

How to Run Python Code?

How to run Python Code?

3 ways to run the python interpreter from the terminal window:

- type 'python'
- type 'ipython'
- type 'python helloworld.py'

4th Ways is Using any IDE like Jupyter, Spider, Pycharm, Canopy, Rodeo etc.

We are using Jupyter notebook as part of our training,

Introduction to IPython Note Book

IPython Notebook

One of Python's most useful features is its interactive interpreter.
It allows for very fast testing of ideas without the overhead of creating test files as is typical in most programming languages. However, the interpreter supplied with the standard Python distribution is somewhat limited for extended interactive use.

Ipython:
A comprehensive environment for interactive and exploratory computing

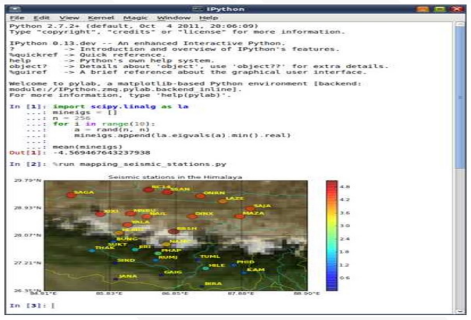
Three Main Components:
An enhanced interactive Python shell. A decoupled two-process c communication model , which allows for multiple clients to connect to a computation kernel, most notably the web-based notebook. An architecture for interactive parallel computing

Some of the many useful features of IPython includes:
- ✓ Command history, which can be browsed with the up and down arrows on the keyboard.
- ✓ Tab auto-completion.
- ✓ In-line editing of code.
- ✓ Object introspection, and automatic extract of documentation strings from python objects like classes and functions.
- ✓ Good interaction with operating system shell.
- ✓ Support for multiple parallel back-end processes, that can run on computing clusters or cloud services like Amazon EC2.

IPython Notebook

- IPython provides a rich architecture for interactive computing with:
  - ✓ A powerful interactive shell.
  - ✓ A kernel for Jupyter.
  - ✓ Easy to use, high performance tools for parallel computing.
  - ✓ Support for interactive data visualization and use of GUI toolkits.
  - ✓ Flexible, embeddable interpreters to load into your own projects.

- Beyond the Terminal ...
  - ✓ The REPL (read, eval, print loop) as a Network Protocol
  - ✓ Kernels
    - ✓ Execute Code
  - ✓ Clients
    - ✓ Read input
    - ✓ Present Output
  - ✓ Simple abstractions enable rich, sophisticated clients



ANALYTIXLABS

IPython Notebook

The Four Most Helpful Commands

- ✓ The four most helpful commands is shown to you in a banner, every time you start IPython:

Command	Description
?	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.

Tab Completion:

- ✓ Tab completion, especially for attributes, is a convenient way to explore the structure of any object you're dealing with. Simply type object_name.<TAB> to view the object's attributes. Besides Python objects and keywords, tab completion also works on file and directory names

ANALYTIXLABS

IPython Notebook

- The %run magic command allows you to run any python script and load all of its data directly into the interactive namespace. Since the file is re-read from disk each time, changes you make to it are reflected immediately (unlike imported modules, which have to be specifically reloaded). IPython also includes dreload, a recursive reload function.
- %run has special flags for timing the execution of your scripts (-t), or for running them under the control of either Python's pdb debugger (-d) or profiler (-p).
- The %edit command gives a reasonable approximation of multiline editing, by invoking your favorite editor on the spot. IPython will execute the code you type in there as if it were typed interactively.

Magic Functions ...

- The following examples show how to call the builtin %timeit magic, both in line and cell mode:

```
In [1]: %timeit range(1000)
100000 loops, best of 3: 7.76 us per loop

In [2]: %%timeit x = range(10000)
   ...: max(x)
   ...:
1000 loops, best of 3: 223 us per loop
```

The builtin magics include:

- Functions that work with code: %run, %edit, %save, %macro, %recall, etc.
- Functions which affect the shell: %colors, %xmode, %autoindent, %automagic, etc.
- Other functions such as %reset, %timeit, %%writefile, %load, or %paste.

ANALYTIXLABS

IPython Notebook

Exploring your Objects

- Typing object_name? will print all sorts of details about any object, including docstrings, function definition lines (for call arguments) and constructor details for classes.
- To get specific information on an object, you can use the magic commands %pdoc, %pdef, %psource and %pfile.

Magic Functions:

IPython has a set of predefined magic functions that you can call with a command line style syntax.
There are two kinds of magics, line-oriented and cell-oriented.

- Line magics are prefixed with the % character and work much like OS command-line calls: they get as an argument the rest of the line, where arguments are passed without parentheses or quotes.
- Cell magics are prefixed with a double %%, and they are functions that get as an argument not only the rest of the line, but also the lines below it in a separate argument.
- You can run the script.py. You can toggle this behaviour by running the %automagic magic.
- A more detailed explanation of the magic system can be obtained by calling %magic,
- To see all the available magic functions, call %lsmagic

ANALYTIXLABS

IPython Notebook

System Shell Commands:
To run any command at the system shell, simply prefix it with !. You can capture the output into a Python list. To pass the values of Python variables or expressions to system commands, prefix them with $.

System Aliases:
- ✓ It's convenient to have aliases to the system commands you use most often.
- ✓ This allows you to work seamlessly from inside IPython with the same commands you are used to in your system shell.
- ✓ IPython comes with some pre-defined aliases and a complete system for changing directories, both via a stack (%pushd, %popd and %dhist) and via direct %cd.
- ✓ The latter keeps a history of visited directories and allows you to go to any previously visited one.

System Shell Commands ...

```
!ping www.bbc.co.uk

files = !ls                    # capture

!grep -rF $pattern ipython/*  # passing vars
```


IPython Notebook

History
- ✓ IPython stores both the commands you enter, and the results it produces. You can easily go through previous commands with the up- and down-arrow keys, or access your history in more sophisticated ways.
- ✓ Input and output history are kept in variables called In and Out, keyed by the prompt numbers. The last three objects in output history are also kept in variables named _, __ and ___.
- ✓ You can use the %history magic function to examine past input and output. Input history from previous sessions is saved in a database, and IPython can be configured to save output history.
- ✓ Several other magic functions can use your input history, including %edit, %rerun, %recall, %macro, %save and %pastebin.

You can use a standard format to refer to lines:

```
%pastebin 3 18-20 ~1/1-5
```

This will take line 3 and lines 18 to 20 from the current session, and lines 1-5 from the previous session.

IPython Notebook

Debugging

- ✓ After an exception occurs, you can call %debug to jump into the Python debugger (pdb) and examine the problem. Alternatively, if you call %pdb, IPython will automatically start the debugger on any uncaught exception.
- ✓ You can print variables, see code, execute statements and even walk up and down the call stack to track down the true source of the problem. This can be an efficient way to develop and debug code, in many cases eliminating the need for print statements or external debugging tools.
- ✓ You can also step through a program from the beginning by calling %run -d theprogram.py.

.

Introduction to Jupyter

Jupyter Notebook

Jupyter Notebook

The Jupyter Notebook is a web application for **interactive** data science and scientific computing.

Using the Jupyter Notebook, you can author engaging documents that combine live-code with narrative text, equations, images, video, and visualizations. By encoding a complete and reproducible record of a computation, the documents can be shared with others on GitHub, Dropbox, and the Jupyter Notebook Viewer.

```
# Install
sudo apt-get install build-essential python-dev
pip install jupyter

# Start
jupyter notebook

# Previously
pip install "ipython[notebook]"
ipython notebook
```

ANALYTIXLABS

Jupyter

jupyter

- Open source, interactive data science and scientific computing across over 40 programming languages.
- The Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.
- Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.

Language of Choice

The Notebook has support for over 40 programming languages, including those popular in Data Science such as Python, R, Julia and Scala.

Share Notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.

Interactive Widgets

Code can produce rich output such as images, videos, LaTeX, and JavaScript. Interactive widgets can be used to manipulate and visualize data in realtime.

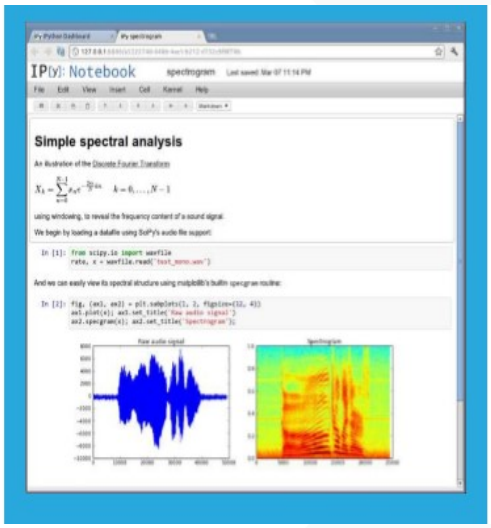
Big-Data Integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, dplyr, etc.

See: Jupyter.ORG Website

ANALYTIXLABS

Jupyter



Project Jupyter

The IPython Notebook (2011)

- Rich Web Client
- Text & Math
- Code
- Results
- Share, Reproduce.

See: **Fernando Perez**, IPython & Project Jupyter

Jupyter

Project Jupyter

IPython

- Interactive Python shell at the terminal
- Kernel for this protocol in Python
- Tools for Interactive Parallel computing

Jupyter

- Network protocol for interactive computing
- Clients for protocol
  - Console
  - Qt Console
  - Notebook
- Notebook file format & tools (nbconvert...)
- Nbviewer

jupyter

What's in a name?

- Inspired by the open languages of science:
  - Julia, Python & R
  - Not an acronym: all languages equal class citizens.
- Astronomy and Scientific Python: A long and fruitful collaboration
- Galileo's notebooks:
  - The original, open science, data-and-narrative papers
  - Authorea: "Science was Always meant to be Open"

Jupyter

Project Jupyter

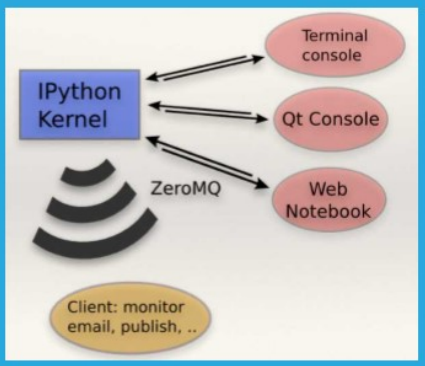
From IPython to Project Jupyter

- Not just about Python: Kernels in **any** language
- IPython : "Official"
- IJulia, IRKernel, IHaskell, IFSharp, Ruby, IScala, IErlang, .. Lots more! ~37 and counting
- Why is it called IPython, if it can do Julia, R, Haskell, Ruby, ... ?"

TL;DR

- Separation of the language-agnostic components
- **Jupyter**: protocol, format, multi-user server
- **IPython**: interactive Python console, Jupyter kernel
- Jupyter kernels = Languages which can be used from the notebook (37 and counting)

A Simple and Generic Architecture



See: **Fernando Perez**, IPython & Project Jupyter

ANALYTIXLABS

Jupyter

The Notebook

Notebook mode supports **literate computing** and **reproducible** sessions

- Allows to store chunks of python along side the results and additional comments (HTML, Latex, MarkDown)
- Can be exported in various file formats

Notebook are the de-facto standard for sharing python sessions.

The Notebook: "Literate Computing"

Computational Narratives

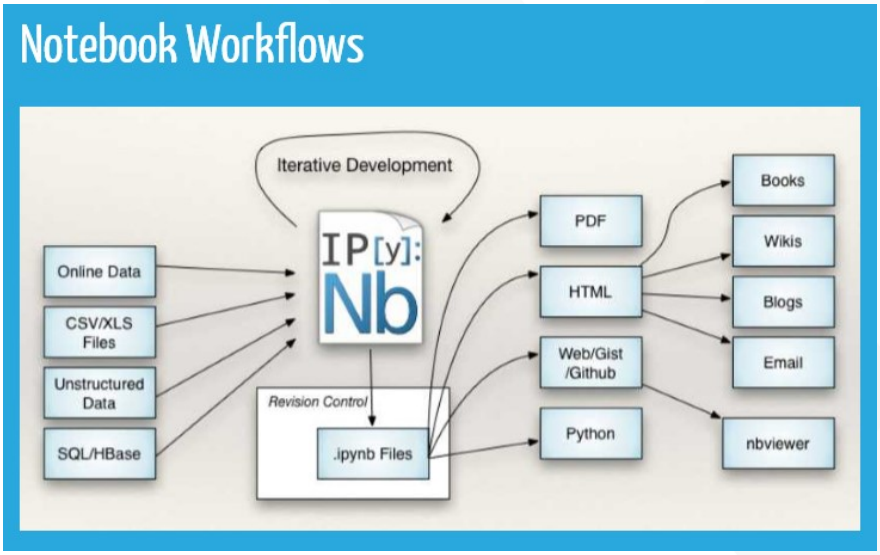
- Computers deal with code and data.
- Humans deal with narratives that communicate.

Literate Computing (not Literate Programming)

- Narratives anchored in a live computation, that communicate a story based on data and results.

ANALYTIXLABS

Jupyter



Jupyter

Seamless Notebook Sharing

nbviewer

- Zero-install reading of notebooks
- Just share a URL
- nbviewer.ipython.org

Jupyter – Getting Started

Review

IPython Notebook

IPython notebook is an HTML-based notebook environment for Python. It is based on the IPython shell, but provides a cell-based environment with great interactivity, where calculations can be organized and documented in a structured way.

Although using a web browser as graphical interface, IPython notebooks are usually run locally, from the same computer that run the browser.

IPython Notebook

- Web-based user interface to IPython, Interactive Python interpreter in the browser
- Literate computing, Open format combining executable code, text and multimedia
- Pretty graphs
- Version controlled science!

To start a new IPython notebook session, run the following command:

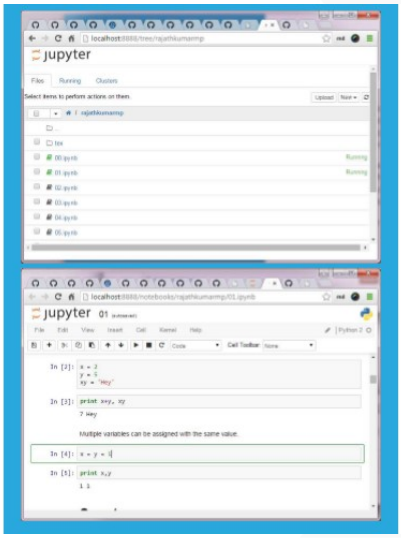
```
ipython notebook
# or
jupyter notebook
```

from a directory where you want the notebooks to be stored.

This will open a new browser window (or a new tab in an existing window) with an index page where existing notebooks are shown and from which new notebooks can be created.

ANALYTIXLABS

Jupyter



Up and Running

An IPython notebook lets you write and execute Python code in your web browser. IPython notebooks make it very easy to tinker with code and execute it in bits and pieces; for this reason IPython notebooks are widely used in scientific computing.

Once IPython is running, point your web browser at http://localhost:8888 to start using IPython notebooks. If everything worked correctly, you should see a screen showing all available IPython notebooks in the current directory.

If you click through to a notebook file, it will be executed and displayed on a new page.

See: **CS231n**: IPython Tutorial

ANALYTIXLABS

Jupyter

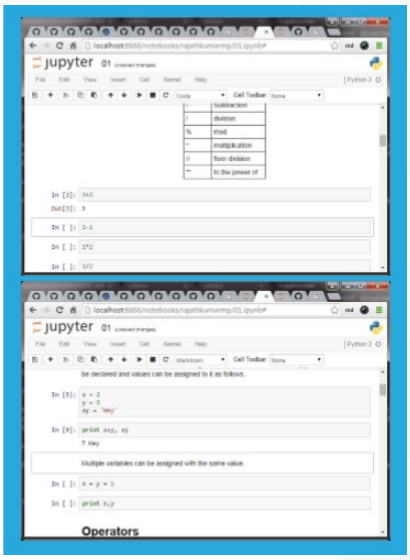
Up and Running

An IPython notebook is made up of a number of **cells**. Each cell can contain Python code. You can execute a cell by clicking on it and pressing `Shift-Enter`. When you do so, the code in the cell will run, and the output of the cell will be displayed beneath the cell. See example.

Global variables are shared between cells. See the notebook after executing the second cell.

By convention, IPython notebooks are expected to be run from top to bottom. Failing to execute some cells or executing cells out of order can result in errors.

See: **CS231n**: IPython Tutorial



ANALYTIXLABS

Exercise

1. Launch new Jupyter notebook (ipython) and save the code
2. Practice Assignment statements (create basic variables and perform mathematical operations)
3. Create markdown file with Some descriptions
4. Practice Short cuts (using Jupyter cheat sheet) like
    1. run the code,
    2. insert/delete the cell,
    3. add/remove the output & line numbers,
    4. Merge/split cells
    5. Toggle between different types of code formats, comments
    6. etc

ANALYTIXLABS

Introduction to Canopy

Canopy: Integrated Analysis Environment

Canopy is a comprehensive Python analysis environment that provides easy installation of the core scientific analytic and scientific Python packages, creating a robust platform you can explore, develop, and visualize on. In addition to its pre-built, tested Python distribution, Canopy has valuable tools for iterative data analysis, visualization and application development including:

- One-Click Python Package Deployment with a Graphical Package Manager
- Code Editor with IPython Notebook Support
- Interactive Graphical Python Code Debugger
- Integrated IPython Prompt
- Convenient Documentation Browser
- Python for Excel with PyXLL (*add-on*)
- Integration with the Intel MKL and Microsoft Python Tools for Visual Studio

Download Canopy from https://store.enthought.com/downloads/#default

Create IPython Notebook in Canopy

In Canopy, go to File -> New -> Jupyter (IPython) Notebook

Name the new notebook and click OK

The same Notebook will open on Web Browser

When you open a new IPython Notebook, an IPython interactive cell with the prompt In[ ]: to the left, appears. You can type code into this cell just as you would in the IPython shell of the *Canopy window*.

ANALYTIXLABS

Contact us

Visit us on: http://www.analytixlabs.in/

For course registration, please visit: http://www.analytixlabs.co.in/course-registration/

For more information, please contact us: http://www.analytixlabs.co.in/contact-us/
Or email: info@analytixlabs.co.in
Call us we would love to speak with you: (+91) 88021-73069

Join us on:
Twitter - http://twitter.com/#!/AnalytixLabs
Facebook - http://www.facebook.com/analytixlabs
LinkedIn - http://www.linkedin.com/in/analytixlabs
Blog - http://www.analytixlabs.co.in/category/blog/

ANALYTIXLABS