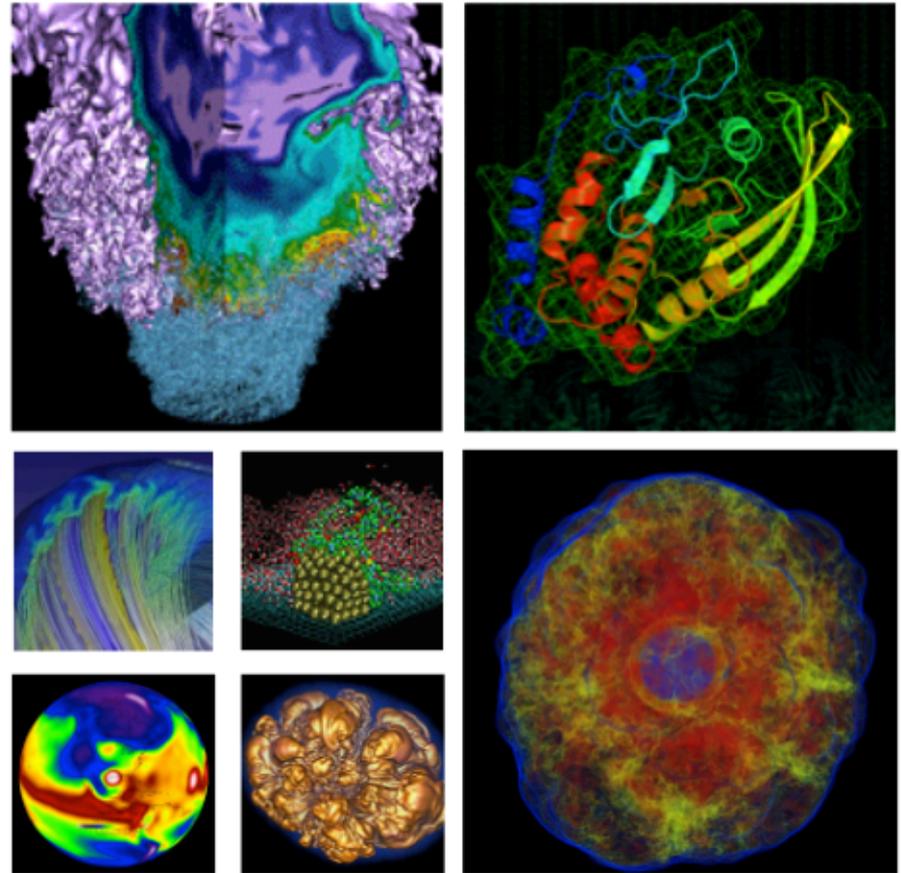


# Data Analytics at NERSC



Rollin Thomas

[rcthomas@lbl.gov](mailto:rcthomas@lbl.gov)

NERSC Data and Analytics Services

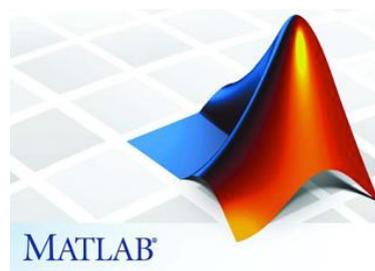
March 21, 2016

NERSC User Group Meeting

- **Data Analytics:** The key to unlocking insight from massive and complex data sets.
- NERSC supports a variety of general-purpose analytics tools and services.
- This talk will cover:
  - Data analytics tools available on the Cray machines.
  - Other analytics services enabled through the web.
  - How to get help with data analytics at NERSC.
  - What's coming?



## Data Analytics Tools





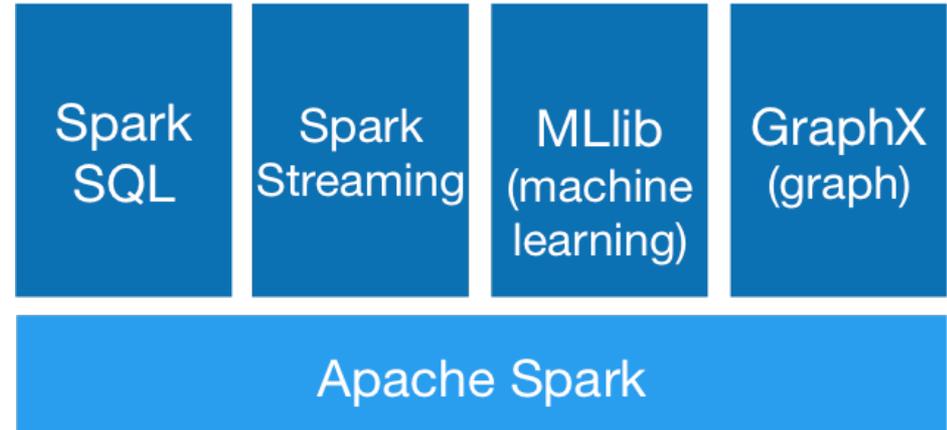
- **R**: Extensible language and environment for statistical computing and graphics.
- Linear, non-linear modeling, classical statistics, time series analysis, classification, clustering, visualization.
- To use R on Edison or Cori:
  - **module load R**
    - Interactive via login or compute nodes (salloc).
    - Or via batch script (sbatch).
    - Variety of approaches for achieving parallelism.
- Users may install packages in \$HOME or ask for system-wide installation via [consult@nersc.gov](mailto:consult@nersc.gov).

- **Python:** Interpreted, general-purpose, high-level programming language. Python 2.7.x and 3.4.x.
- Number of scientific computing packages: numpy, scipy, matplotlib, scikit-learn, mpi4py, ...
- To use Python on Edison or Cori, *always* module load:
  - **module load python** (NERSC-built)
  - Or, e.g., **module load python/2.7-anaconda**
  - Login, interactive (salloc), and batch (sbatch).
  - Parallelism: mpi4py, multiprocessing, Intel MKL.
- Users may install packages via pip, virtualenv, conda or ask for system-wide installation via [consult@nersc.gov](mailto:consult@nersc.gov).

- Demand for Python at NERSC is large and increasing.
- Want Python to become more of a first-class citizen.
- Parallelism issues and progress:
  - Improved launch times by mounting /usr/common read-only with client-side caching on compute.
  - Greater awareness of tools and strategies for scaling up Python applications.
- (Too?) many choices in distribution space:
  - But certain Cray-specific subtleties (parallelism) require NERSC to build certain packages.
  - Anaconda Python now includes Intel MKL support.
  - Users are encouraged to consider Anaconda.

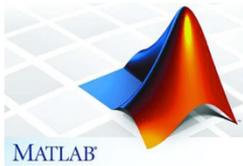
- **Spark:** Fast general purpose engine for large scale data processing, map-reduce, etc.

Computation Type	Spark Implementation
Machine Learning	MLib, Spark ML
Graph Computations	GraphX
Database Operations	Spark SQL
Streaming Analysis	Spark Streaming
Your Own Custom Analysis	Using Spark's Built In Functions

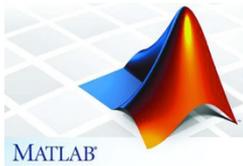


Computation types can be combined seamlessly all in the same piece of code.

- Spark has APIs in Java, Scala, Python, and R.
  - NERSC recommends using Spark on Cori:
    - Large memory and I/O bandwidth requirements.
    - **module load spark**
    - Interactive (salloc) and batch (sbatch) supported.
    - Do not load the module until batch job launches!
  - Spark support is experimental.
  - Contact [consult@nersc.gov](mailto:consult@nersc.gov) if you have questions.
- 
- 
- Also in the ML space: **neon**, a Python-based, scalable Deep Learning executable and library.



- Tools requiring a license checkout:
  - **Matlab:** `module load matlab`  
Compute, visualize, and program in a familiar environment that “looks like math.”
  - **Mathematica:** `module load mathematica`  
Symbolic mathematics, numerical calculations, visualization in a notebook interface.
  - **IDL:** `module load idl`  
Interactive data analysis and visualization environment.



- More tools (no license checkout):
  - **ROOT:** `module load root`  
Object-oriented framework for large-scale data analysis. Particle physics to data mining.
  - **Julia:** `module load julia` (Cori)  
Experimental high-level language for scientific computing with powerful type semantics.
- A NERSC best practice:
  - **Use NX** for interactive visualization tools and Mathematica notebooks.
  - <http://www.nersc.gov/users/connecting-to-nersc/using-nx/>





IP[y]: IPython  
Interactive Computing

## Web-Enabled Data Analytics Tools



# Jupyter/IPython and RStudio



## IP[y]: IPython Interactive Computing



Powerful interactive shell originally developed for Python. Also provides a web browser-based **notebook** supporting:

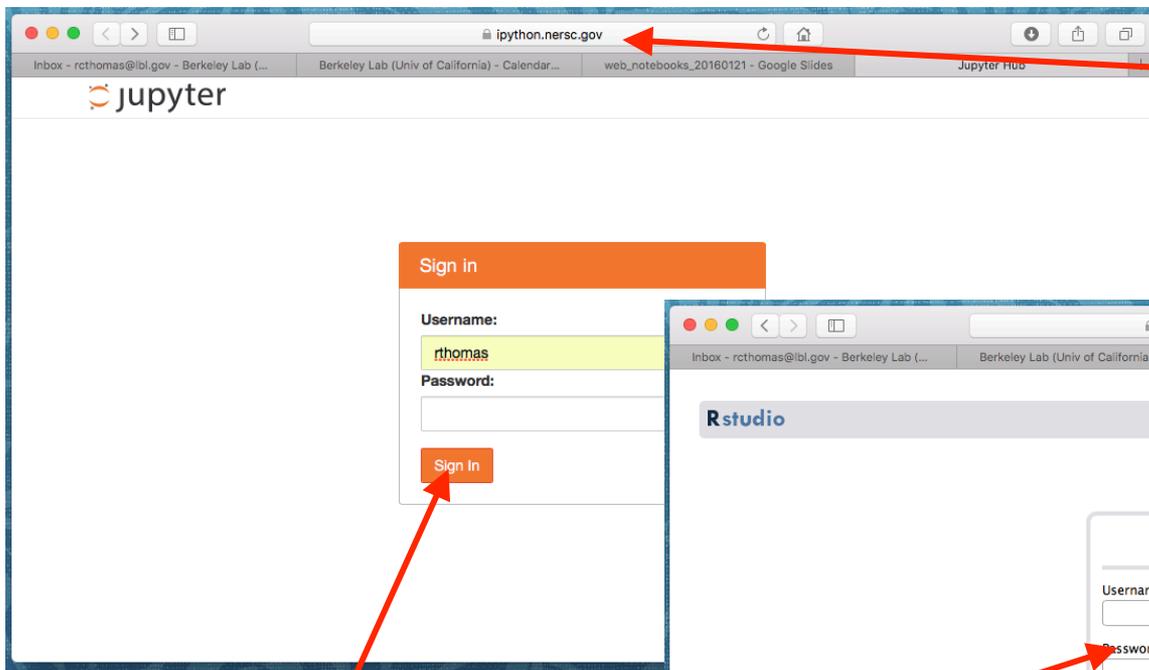
- Execution of code and annotation with text.
- In-line plotting and visualization.
- Interactive widgets.

Jupyter is the notebook part (language agnostic). IPython is the Python shell and a Jupyter “kernel.”



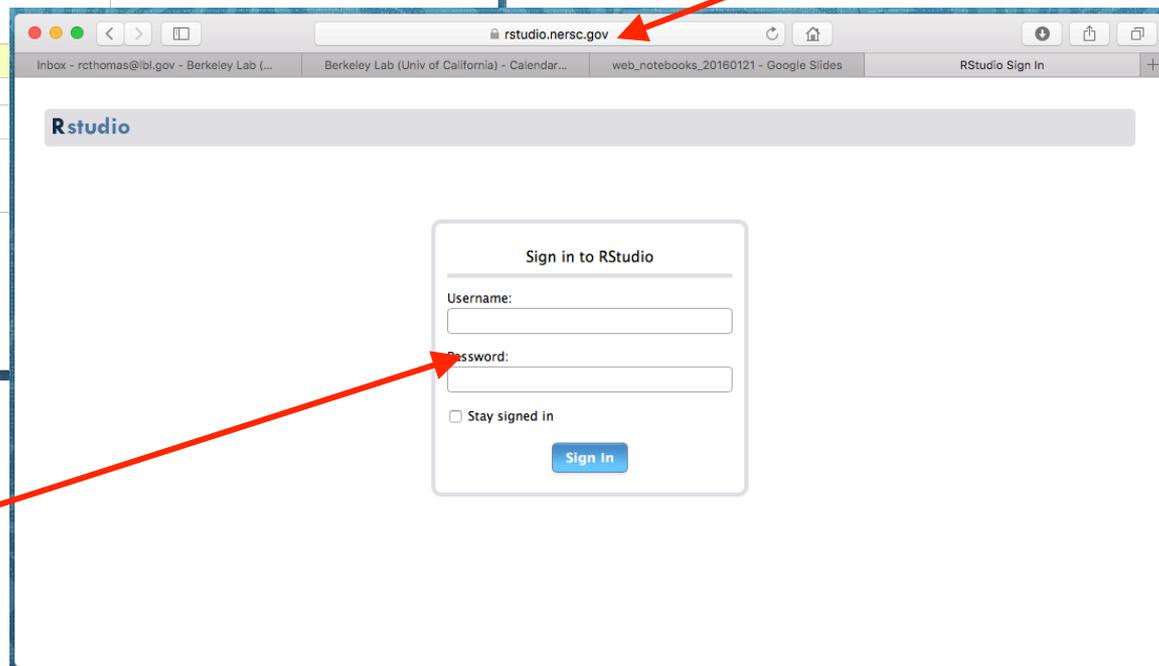
Integrated development environment (IDE) for R. (R is also available at NERSC at the command line.) RStudio provides a web browser-based IDE.

# How Do I Work This?



ipython.nersc.gov  
(or jupyter.nersc.gov)

rstudio.nersc.gov



Just your usual NERSC  
username & password.

# IPython



The screenshot shows the Jupyter web interface. At the top left is the 'jupyter' logo. To the right are 'Control Panel' and 'Logout' buttons. Below the logo are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' Below this is a file browser showing a list of folders: consult, intel, local, tmp, tmp123, tmp234, venv, work, Untitled.ipynb, Untitled1.ipynb, myquota.txt, out, try.py, and try.sh. On the right side, there are 'Upload' and 'New' buttons. A dropdown menu is open under 'New', showing options: Text File, Folder, Terminal, Notebooks, Python 2, and Python 3. A red arrow points from the text 'Click to launch notebook...' to the 'Python 2' option in the dropdown menu.

Click to launch notebook...

The screenshot shows a Jupyter notebook window titled 'Untitled2'. The top bar includes 'jupyter', 'Untitled2', 'Last Checkpoint: 3 minutes ago (unsaved changes)', and 'Control Panel' and 'Logout' buttons. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A toolbar contains icons for file operations and a 'Code' dropdown menu. The main area shows a code cell with the following code:

```
In [1]: import glob
glob.glob( "**")
```

The output cell shows the following output:

```
Out[1]: ['Untitled.ipynb',
'tmp',
'venv',
'intel',
'Untitled2.ipynb',
'Untitled1.ipynb',
'consult',
'myquota.txt',
'try.py',
'tmp234',
'out',
'local',
'work',
'try.sh',
'tmp123']
```

At the bottom, there is an input field for the next code cell:

```
In [ ]:
```

Notebook can see:

- \$HOME
- /project
- /global/project{a,b}
- /global/dna
- *Not* \$SCRATCH

# RStudio



The screenshot displays the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, and Help. The user is logged in as 'rthomas' and can sign out. The console window shows the R version 3.1.1 (2014-07-10) and copyright information. The environment pane is empty. The file explorer shows a directory listing with files like .Rprofile, consult, intel, local, myquota.txt, out, R, tmp, tmp123, tmp234, try.py, try.sh, and Untitled.ipynb files.

Console output:

```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-redhat-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]
> |
```

File Name	Size	Modified
.Rprofile	292 B	Dec 15, 2015, 3:35 PM
consult		
intel		
local		
myquota.txt	664 B	Dec 15, 2015, 9:51 AM
out	12.3 KB	Jan 14, 2016, 11:28 AM
R		
tmp		
tmp123		
tmp234		
try.py	65 B	Dec 15, 2015, 9:37 AM
try.sh	146 B	Jan 7, 2016, 2:02 PM
Untitled.ipynb	2.7 KB	Dec 16, 2015, 1:26 PM
Untitled1.ipynb	72 B	Jan 13, 2016, 1:55 PM
Untitled2.ipynb	1.3 KB	Jan 15, 2016, 2:51 PM
venv		

# Getting Help with Data Analytics at NERSC

# Ways to Get Help

- Ask us anything! Tell us anything! Suggest anything!
  - [consult@nerisc.gov](mailto:consult@nerisc.gov)
- Documentation:
  - <http://www.nerisc.gov/users/data-analytics/data-analytics/>
  - <http://www.nerisc.gov/users/data-analytics/data-analytics/python/>  
(Just reorganized and updated!)

FOR USERS

- » Live Status
- » My NERSC
- » Move to CRT
- » Getting Started
- » Accounts & Allocations
- » Computational Systems
- » Storage & File Systems
- » Data & Analytics
- » Data Management
- » Data Analytics
- MATLAB
- Mathematica
- Python
- Web Applications for Data
- IDL
- ROOT
- R
- Apache Spark
- Neon
- Omero
- Fiji
- ImageJ
- Julia

Home » For Users » Data & Analytics » Data Analytics

## DATA ANALYTICS

Analytics is key to gaining insights from massive, complex datasets. NERSC provides general purpose analytics (iPython, MATLAB, IDL, Mathematica, ROOT), statistics (R), machine learning (BDAS/Spark) and imaging (OMERO, Fiji) tools.

### MATLAB »

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. [Read More »](#)



### Mathematica »

Mathematica is a fully integrated environment for technical computing. Performs symbolic manipulation of equations, integrals, differential equations and almost any mathematical expression. Numeric results can be evaluated also. [Read More »](#)

### Python »

Python is an interpreted, general-purpose high-level programming language. Various versions of Python are installed on NERSC systems with a number of scientific computing libraries like numpy and scipy, and visualization libraries like matplotlib. [Read More »](#)

### Web Applications for Data Analytics »

## PYTHON

Python is an interpreted, general-purpose high-level programming language. Various versions of Python are installed on NERSC systems with a number of scientific computing libraries like numpy and scipy, and visualization libraries like matplotlib.

On Cori and Edison, Python is available either as a NERSC-built module or through the Anaconda distribution. Both approaches require at least one "module load" command. Using the system-provided Python (from /usr/bin) is *strongly discouraged* except for the simplest tasks, as it is generally a much older version of Python than provided by NERSC.

Python users may also be interested in the experimental [iPython/Jupyter](#) notebook web application service.

### NERSC Python Modules »

This page describes NERSC's installation of Python modules on the Cray systems and how users can take advantage of it. A partial list of installed Python packages is included. [Read More »](#)

### Anaconda Python »

The Anaconda distribution provides an alternative to NERSC's Python installation on the Cray systems. This page instructs users on how to use the Anaconda distribution at NERSC. [Read More »](#)

### Running Scripts »

Python scripts can be run on Cray compute nodes and login nodes (with considerations). This page describes how to run serial or parallel (multiprocessing or MPI) Python jobs on the Cray systems at NERSC. [Read More »](#)

### Scaling Up »

Creating parallel Python codes that robustly scale in modern high-performance computing environments can be challenging. Here we outline various strategies to scale parallel Python applications at NERSC. [Read More »](#)

### User Packages »

There's more than one way users can manage installation of Python packages on their own. Here are some tips for managing Python packages at NERSC. [Read More »](#)

### Best Practices »

Our goal is to provide options to Python users and help them pick the best solution for them. Here are some evolving best practices Python users should observe. [Read More »](#)

### Availability »

NERSC systems where Python modules are available, and what versions are available. [Read More »](#)

# Developments in Data Analytics at NERSC

# Recent and Near-Term Developments



- **Jupyter on Cori: (mid-to-late 2016).**
  - Jupyter notebooks using Cori compute nodes.
  - Access to Cori \$SCRATCH.
  - Use notebooks to launch large analytics workflows.
  - NERSC+: UC Berkeley, Cray, HPC community.
- **Consolidating Python package management:**
  - Update schedule tied with system upgrades.
  - Looking for user feedback and prioritization.
  - Ability to manage your own software stack?
  - Anaconda vs NERSC-built vs Intel distribution?

- **Julia, Spark on Cori:**

- Available experimentally. please kick the tires!

- **Data Analytics:** The key to unlocking insight from massive and complex data sets.
- NERSC supports a variety of general-purpose analytics tools and services.
- What this talk covered:
  - Data analytics tools available on the Cray machines.
  - Other analytics services enabled through the web.
  - How to get help with data analytics at NERSC.
  - What's new and coming up in data analytics tools.



# NERSC

National Energy Research Scientific  
Computing Center



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

