



Data Mining as a Service with ROOT

E. Tejedor, D. Piparo, P. Mató

on behalf of the ROOT team – EP-SFT

L. Mascetti, J. Moscicki, M. Lamanna – IT-ST

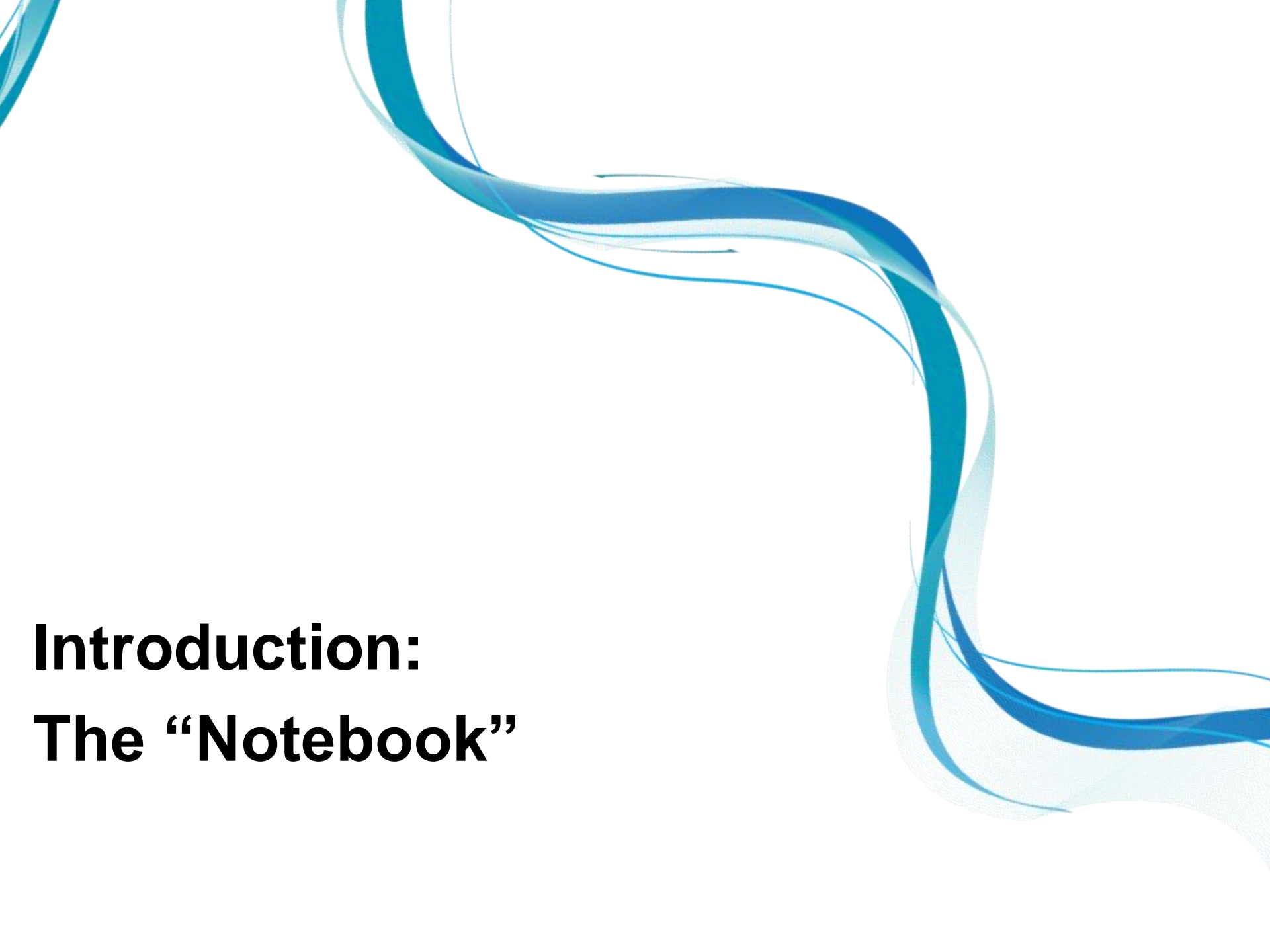
ACAT 2016. UTFSM, Valparaíso (Chile)

18/01/2016





- The DMaaS project
- Integration of ROOT with the Notebook technology
 - ROOTbooks for data analysis
- A new service: ROOT & Jupyter Notebooks within the CERN IT services' portfolio
 - Spotlight on storage: EOS, CERNBox

The background of the slide features several flowing, wavy lines in various shades of blue, ranging from light sky blue to a deeper teal. These lines originate from the top left and curve downwards and to the right, creating a sense of movement and fluidity. The lines are semi-transparent, allowing them to overlap and create darker tones where they intersect.

Introduction: The “Notebook”

In a browser

ROOT Notebook Functionalities

Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$

Text and Formulas

The image shows a web browser window displaying a Jupyter Notebook. The browser's address bar shows 'localhost:8888/tree'. The notebook's header includes a 'ROOT' logo, the title 'Notebook Functionalities', and buttons for 'Control Panel' and 'Logout'. Below the header is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A toolbar contains icons for file operations and cell execution, along with a 'Code' dropdown menu and a 'Cell Toolbar' set to 'None'. The main content area features a large heading 'Welcome to the Notebook Technology' and a paragraph explaining that it is a markdown cell. Below this is a code cell containing a Python function definition.

localhost:8888/tree

ROOT Notebook Functionalities Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$

```
In [1]: def thisFunction():
        return 42
```

Code

Home * +

localhost:8888/tree

ROOT Notebook Functionalities

Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$

```
In [1]: def thisFunction():
        return 42
```

This is a notebook in Python

Code

Home * +

localhost:8888/tree

ROOT Notebook Functionalities

Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$

```
In [1]: def thisFunction():
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

Code

```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSfee.jpg \  
        > SF.jpg
```



We can invoke commands in the shell...

Shell Commands


```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSfee.jpg \  
> SF.jpg
```

```
% Total      % Received % Xferd  Average Speed   Time  
Time        Time      Current           Dload  Upload   Total  
Spent       Left     Speed  
100 128k  100 128k    0      0 2731k      0  --:--:--  
--:--:--  --:--:-- 2787k
```



... and get their output

Shell Commands

```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootasdemo.web.cern.ch/rootasdemo/SaaSfee.jpg \  
> SF.jpg
```

```
% Total      % Received % Xferd  Average Speed   Time  
Time         Time    Current                      Dload  Upload   Total  
Spent       Left   Speed  
100 128k  100 128k    0     0 2731k      0  --:--:--  
--:--:--  --:--:-- 2787k
```

```
In [4]: from IPython.display import Image  
        Image(filename="./SF.jpg",width=225)
```

```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSfee.jpg \  
> SF.jpg
```

```
% Total      % Received % Xferd  Average Speed   Time  
Time        Time      Current                Dload  Upload   Total  
Spent       Left     Speed  
100 128k  100 128k    0      0 2731k      0  --:--:--  
--:--:--  --:--:-- 2787k
```

```
In [4]: from IPython.display import Image  
        Image(filename="./SF.jpg",width=225)
```

```
Out[4]:
```



Images

In [1]:

In a browser

In [2]: `thisFunction()`

Out[2]: 42

Text and Formulas

In [3]:

```
%%bash
curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSFee.jpg \
> SF.jpg
```

Code

Shell Commands

% Total Time Spent	% Received Left	Current Speed	Dload	Upload	Time Total
100	100	2787k	2731k	0	---

```
In [4]: from IPython.display import Image
Image(filename='./SF.jpg',width=225)
```

Out[4]:



Images

Data mining with ROOT “as a service”

Interface: Jupyter Notebooks

Goals:

- Use ROOT only with a web browser
 - Platform independent ROOT-based data analysis
 - Calculations, input and results “in the cloud”
- Allow easy sharing of scientific results: plots, data, code
 - Storage is crucial
- Simplify teaching of data processing and programming
- Potential integration with other analysis ecosystems: R, Python, ...





Integration of ROOT with Notebooks

ROOT

JupyROOT
(ROOT-Jupyter integration)



Terminal

Control Panel

Logout

Files Running Clusters

A Choice of Kernels

Select items to perform actions on them.

Upload New

Home icon

- PresentationNotebooks
- cernbox
- HowTo_ROOT-Notebooks.ipynb
- HowTo_ROOT-Notebooks_Long.ipynb
- My First Notebook.ipynb
- Untitled.ipynb

- Text File
- Folder
- Terminal
- Notebooks
- Python 2
- Python 3
- ROOT C++



File

Edit

View

Insert

Cell

Kernel

Help



Python 2 ○



Code

Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer
```

```
Welcome to JupyROOT 6.07/03
```

File

Edit

View

Insert

Cell

Kernel

Help



Python 2



Code

Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer
```

Welcome to JupyROOT 6.07/03

```
In [ ]: %%cpp
auto myHisto = TH
```

```
TH1
TH1C
TH1D
TH1F
TH1I
TH1K
TH1S
TH2
TH2C
TH2D
```

C++ Cells in Python
Notebooks

ROOT Tab
Completion

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer  
Welcome to JupyROOT 6.07/03
```

```
In [2]: %%cpp  
auto myHisto = TH1F("h", "MyData;X;Y", 64, -4, 4); // C++11
```

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer
```

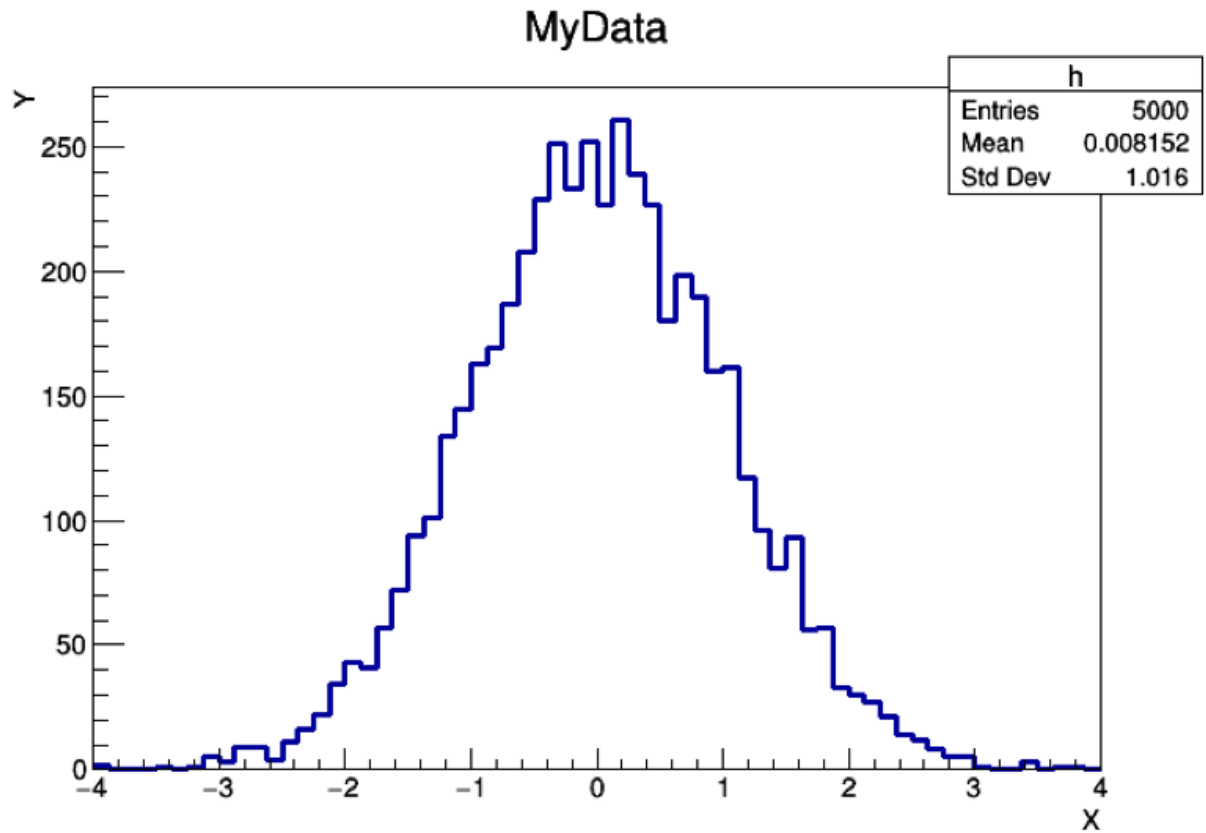
```
Welcome to JupyROOT 6.07/03
```

```
In [2]: %%cpp
auto myHisto = TH1F("h", "MyData;X;Y", 64, -4, 4); // C++11
```

```
In [3]: h = ROOT.myHisto # Find the variable back in Python!
h.FillRandom("gaus")
c = ROOT.TCanvas()
h.Draw()
c.Draw()
```

C++-Python
Interoperability

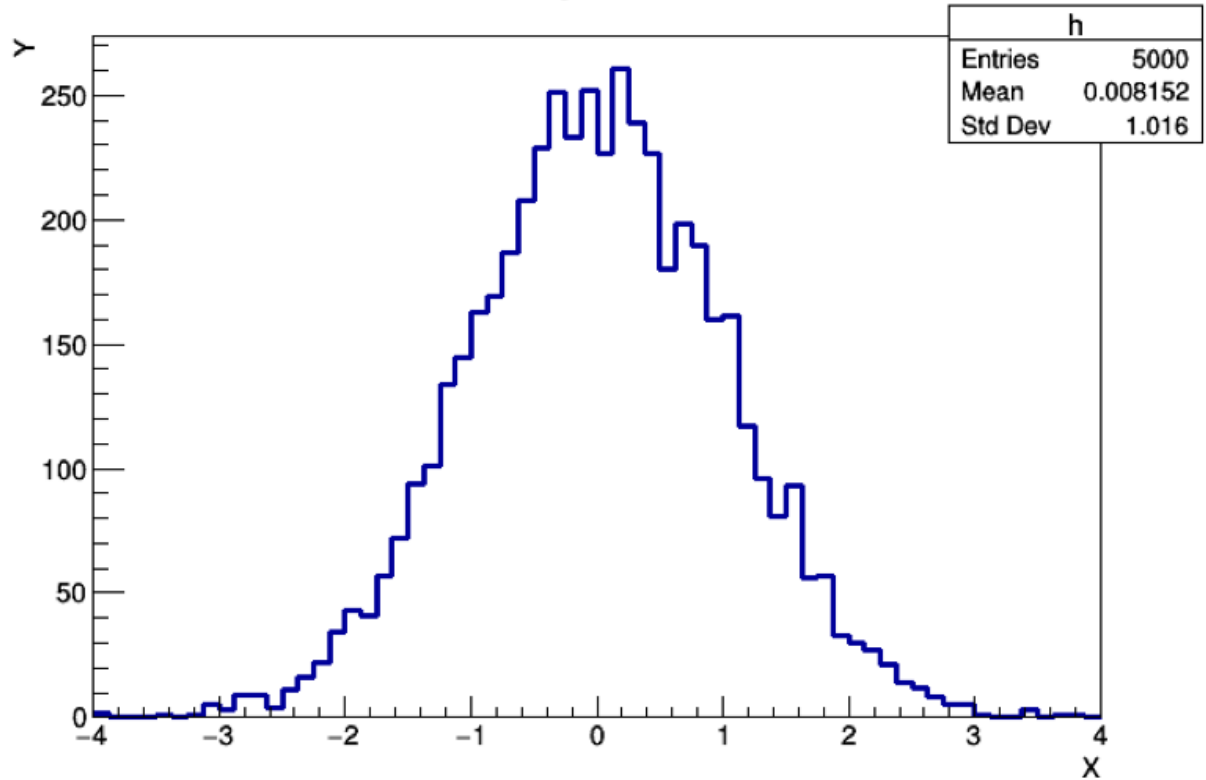
c.Draw()



**Seamless
display of
graphics**

c.Draw()

MyData



Syntax
Highlighting

```
In [4]: %%cpp -d
double myG(double* x, double* par){
    auto res = (x[0]-par[1])/par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
```

```
In [4]: %%cpp -d
double myG(double* x, double* par){
    auto res = (x[0]-par[1])/par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
```

```
In [5]: f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
fr = ROOT.h.Fit(f,"S") # Capture printouts
```

```
In [4]: %%cpp -d
double myG(double* x, double* par){
    auto res = (x[0]-par[1])/par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
```

```
In [5]: f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
fr = ROOT.h.Fit(f,"S") # Capture printouts
```

```
FCN=47.4997 FROM MIGRAD      STATUS=CONVERGED      69 CALLS      70 TO
TAL
```

```
EDM=2.04372e-09      STRATEGY= 1      ERROR MATRIX ACC
```

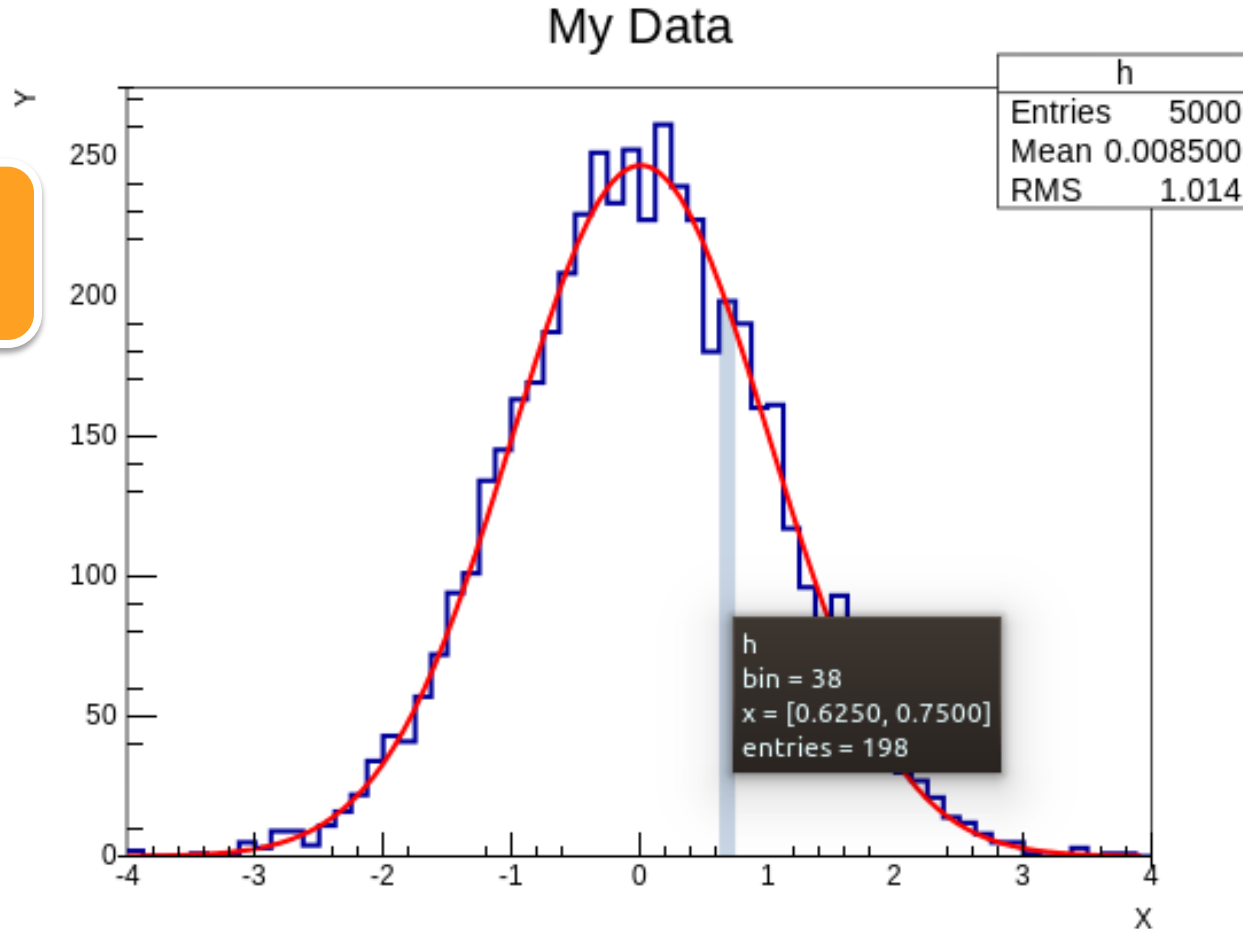
```
URATE
```

EXT	PARAMETER	STEP	FIRST
NO.	NAME	VALUE	ERROR
1	N	2.46469e+02	4.31493e+00
2	mu	1.04793e-02	1.43576e-02
3	sigma	1.00316e+00	1.03818e-02

```
1 N          2.46469e+02  4.31493e+00  1.19092e-02  -5.38026e-06
2 mu         1.04793e-02  1.43576e-02  4.87640e-05  4.15093e-03
3 sigma      1.00316e+00  1.03818e-02  2.86307e-05  -2.55310e-04
```






```
In [6]: ROOT.enableJSVis() # Not active by default yet!
c.Draw()
ROOT.disableJSVis()
```


JSROOT
Visualisation

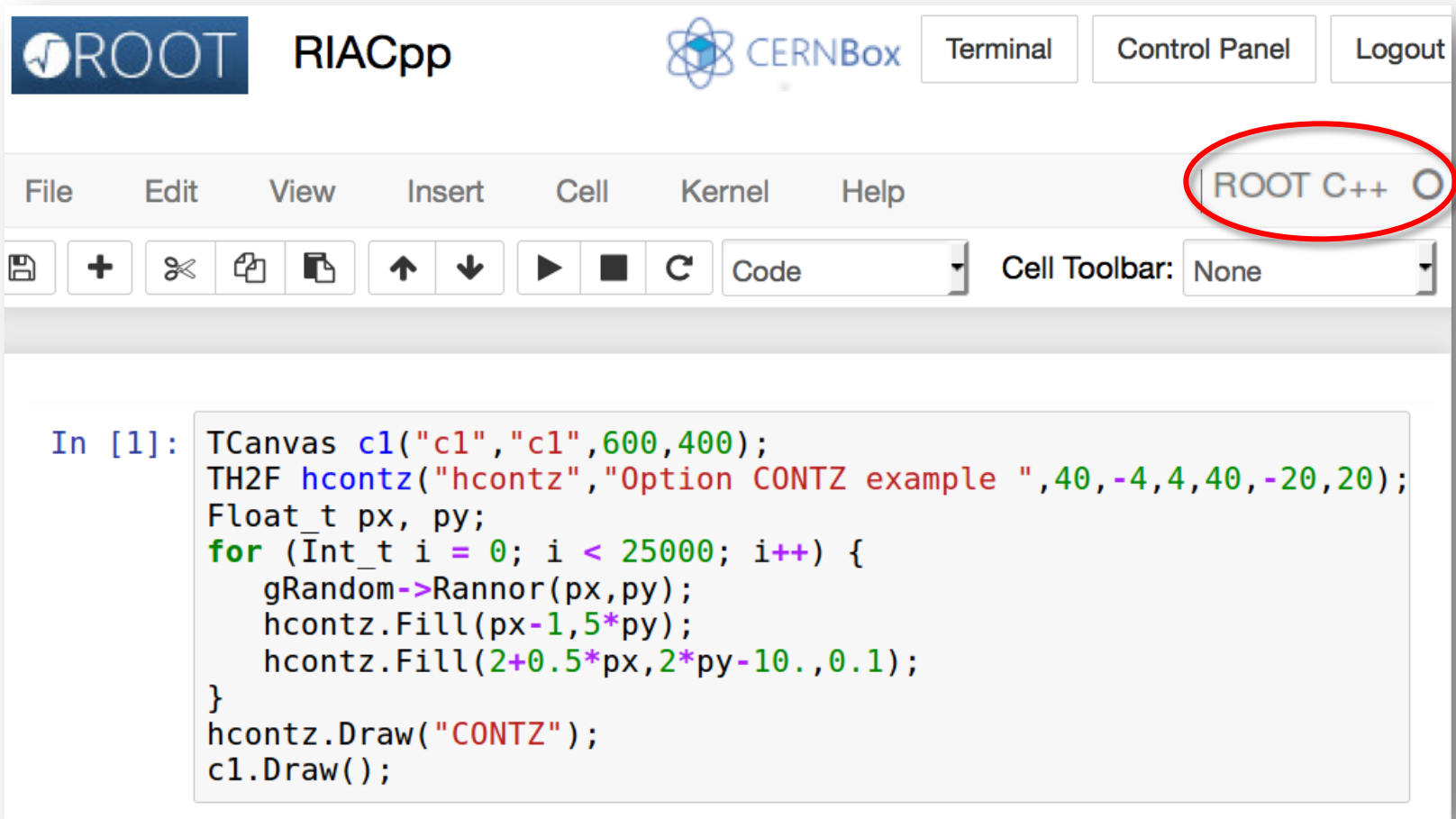


Select items to perform actions on them.

[Upload](#)[New ▾](#) ▾ 

-  [PresentationNotebooks](#)
-  [cernbox](#)
-  [HowTo_ROOT-Notebooks.ipynb](#)
-  [HowTo_ROOT-Notebooks_Long.ipynb](#)
-  [My First Notebook.ipynb](#)
-  [Untitled.ipynb](#)

- Text File
- Folder
- Terminal
-
- Notebooks
- Python 2
- Python 3
-  ROOT C++

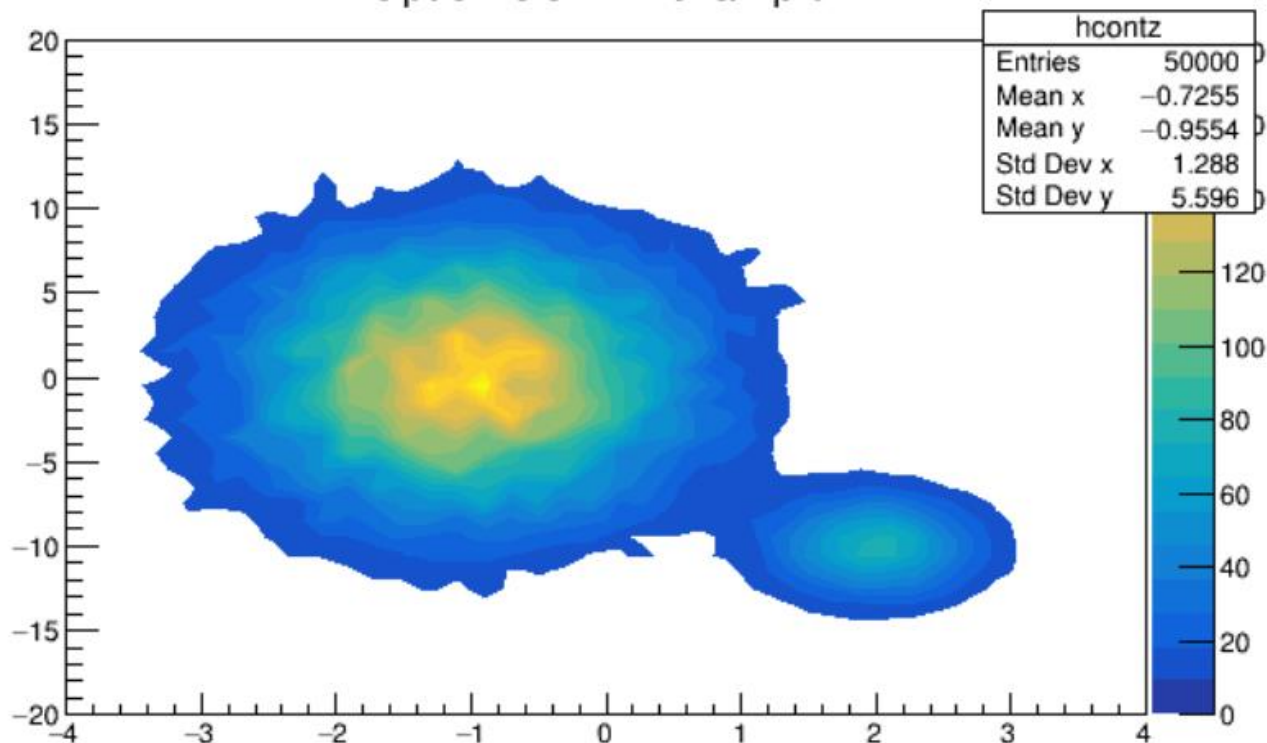


The screenshot shows the ROOT C++ interface. At the top, there is a ROOT logo, the text "RIACpp", and a CERNBox logo. To the right are buttons for "Terminal", "Control Panel", and "Logout". Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". On the far right of the menu bar, "ROOT C++" is circled in red. Below the menu bar is a toolbar with icons for file operations (save, new, copy, paste, undo, redo), execution (run, stop, refresh), and a dropdown menu for "Code". To the right of the toolbar is a "Cell Toolbar" dropdown menu set to "None". The main area contains a code cell with the following C++ code:

```
In [1]: TCanvas c1("c1","c1",600,400);
TH2F hcontz("hcontz","Option CONTZ example ",40,-4,4,40,-20,20);
Float_t px, py;
for (Int_t i = 0; i < 25000; i++) {
    gRandom->Rannor(px,py);
    hcontz.Fill(px-1,5*py);
    hcontz.Fill(2+0.5*px,2*py-10.,0.1);
}
hcontz.Draw("CONTZ");
c1.Draw();
```

```
hcontz.Draw("CONTZ");  
c1.Draw();
```


Option CONTZ example



Try It Out! – ROOT Binder




<http://mybinder.org/repo/cernphsft/rootbinder>




ROOT
Data Analysis Framework

ROOT is a framework for data processing, born at [CERN](#), at the heart of high-energy physics research. Every day, thousands of physicists use ROOT applications to analyze petabytes of data or to perform simulations.

Try a ROOTbook now: choose your favourite language!




Python

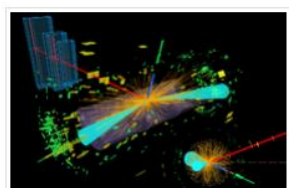


C++

More ROOTBooks!



3D Geometries Visualization



Modelling and Fitting: Higgs

Anonymous access

View, Create and Run
ROOTbooks!



Try It Out! - Local server

Follow some simple instructions in:

<https://root.cern.ch/how/how-create-rootbook>

and...

```
$ root --notebook
```

This command:

1. Starts a local notebook server
2. Connects to it via the browser

Provides a ROOT C++
kernel and the rest of
ROOTbook goodies



- **ROOTbooks How-Tos**

<https://root.cern.ch/howtos#Jupyter%20Notebooks>

- **ROOT bindings for Jupyter**

<https://github.com/root-mirror/root/tree/master/bindings/pyroot/JupyROOT>

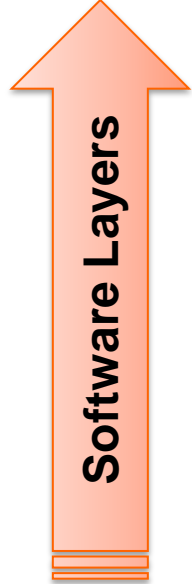
- **ROOT C++ Kernel**

<https://github.com/ipython/ipython/wiki/IPython-kernels-for-other-languages>

- **Examples** (15 already) from the *new ROOT Tutorials* can be found at:

<https://root.cern.ch/code-examples#notebooks>

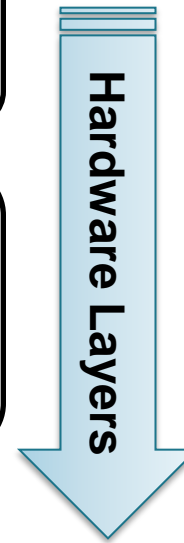
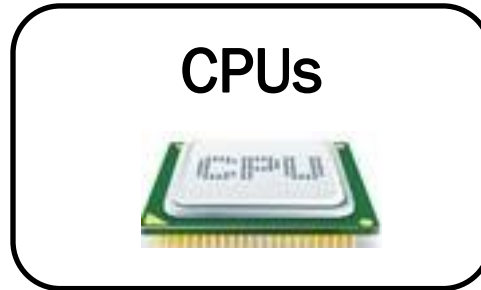
both in Python and C++ (and mixed!)



ROOT

JupyROOT
(ROOT-Jupyter integration)

JupyterHub +
CERN Add-ons



DMaaS and CERN services' Portfolio





Integration With CERN's Ecosystem

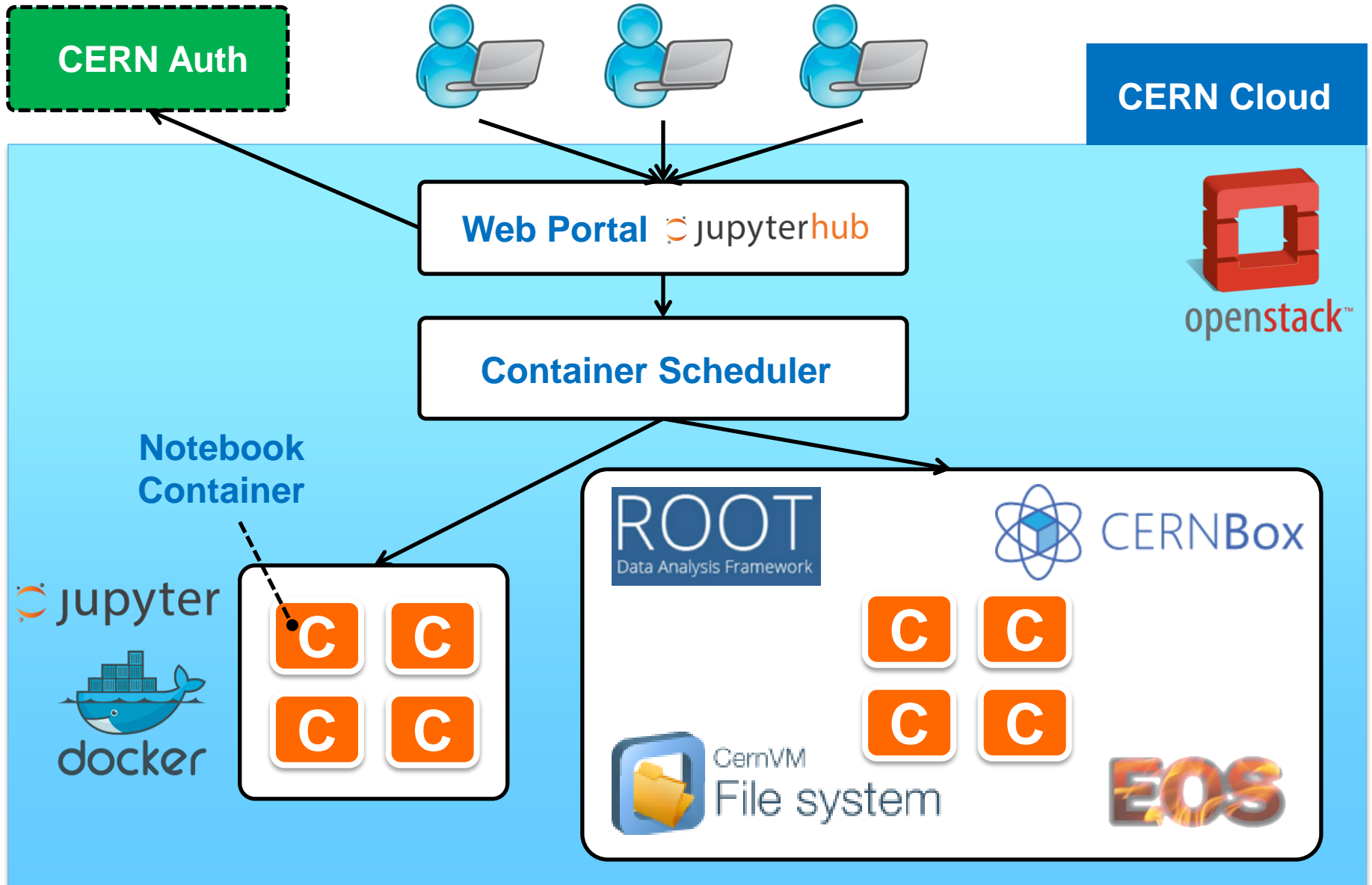
- Centrally provide ROOT as a Service
- Authentication with **CERN credentials**
- Infrastructure: **virtual machines** in OpenStack Cloud
- **Software distribution**: CVMFS
- **Storage access**: CERNBox, EOS
 - All data potentially available!
- Synergy with **document sharing**
 - Notebook visualiser already available in Indico



openstack™



Pilot Service (Distributed)





CERNBox: Store your data



Terminal

Control Panel

Logout

Select items to perform actions on them.

Upload

New ▾



/ cernbox



folder tutorials



file HowTo_ROOT-Notebooks.ipynb



file My First Notebook.ipynb



All files		New		
Favorites	<input type="checkbox"/>	Name ▲	Size	Modified
Shared with you		.ipynb_checkpoints	0 kB	2 minutes ago
Shared with others		My First Notebook.ipynbye0opi_m	0 kB	a few seconds ago
Shared by link		tutorials	0 kB	7 days ago
		HowTo_ROOT-Notebooks.ipynb	214 kB	7 days ago
		My First Notebook.ipynb	24 kB	2 minutes ago
		myOutputFile.root	6 kB	3 minutes ago
Deleted files		myPlot.pdf	14 kB	3 minutes ago



CERNBox: Share

	HowTo_ROOT-Notebooks.ipynb	214 kB	7 days ago
	My First Notebook.ipynb	25 kB	25 minutes ago
★	my... .root Download Versions Share	6 kB	a few seconds ...
	myPlot.pdf	14 kB	27 minutes ago



CVMFS: Software Environment

- Define a custom environment via a web form:

LCG release

82 ROOTaaS6

Platform

x86_64 SLC6 GCC4.9 Opt

Spawn



CernVM
File system

~150 packages

82

N releases



- CERN Summer Student Program, ROOT tutorial: Interactive notebooks offered
 - **50 participants, perfect scaling, a success!**

<https://indico.cern.ch/event/407519>
- Data Science @ LHC Workshop, Multivariate analysis tutorial

<http://indico.cern.ch/event/395374/>
- E-Planet exchange @ UERJ, Brazil
 - 30 participants, every day for a week, 3h a day

<https://indico.cern.ch/event/402660/>



- ROOT is now integrated with notebooks
 - Python and C++ interactive shells
 - Tab completion, C++/Python integration, syntax highlighting, graphics inlining
 - Available now!
 - Future: leverage ROOT-R integration, TMVA, ...
- Integration with the CERN services portfolio
 - Collaborating with IT department
 - Special thanks to IT-ST
 - Plan to deliver a Pilot Service in 2016