PyROOT tutorial experience from SWC Workshop

By : Guillermo Fidalgo
Physics Department
University of Puerto Rico - Mayagüez

About me

Masters Student in Physics

- My research:
  - Analysis on Exotic Physics: Emerging Jets.
  - I do Machine Learning for DQM Studies for the CMS Tracker
- I want to know more about:
  - C++, Arduino

General feedback

- As a python user the C++ implementation of ROOT has not always been the most straightforward to learn.
- I view pyROOT as a bridge to start better understanding and learning how C++ ROOT implementation works.
- Fast, easier to learn because of python syntax, flexible.
- Motivated to search for more advanced features for my use cases
- Lots of documentation
- Well placed and "simple enough" exercises.

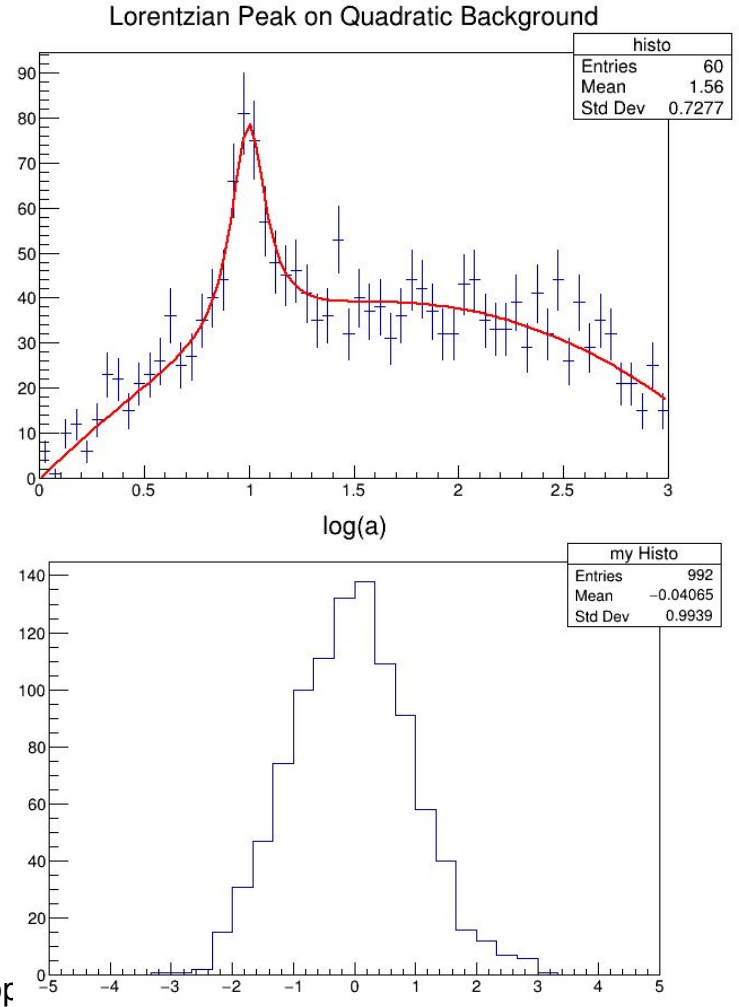
Personal Highlights

- The `%jsroot on` magic that converts static images into interactive plots
- Easy fit implementation via `h.Fit()` after creating a ROOT cpp function
- RDataFrame
- Motivated me to search for more complex functionality.

My thoughts on the exercises

- Excellent for new learners
  - Easy but challenging
  - Gratifying to solve
  - Actually useful
- For the RDataFrames
  - Not exactly unique to them but using strings in `Define()` and `Filter()` is like magic.
  - Not clear how it works unless there is previous knowledge on the data structure.

```
In [3]: 1 df1 = df.Define("good_pt", "sqrt(px*px + py*py)[E>100]")
```



More thoughts

Defining a cpp function and accessing it through ROOT.<cpp function> might be too convenient. Seems like one can unknowingly break important functionality.

```
%%cpp
double gaussian(double *x, double *par) {... }

fitFunc = ROOT.TF1("fitFunc", ROOT.gaussian, xmin=-5, xmax=5, npar=2)
```

Great!

```
%%cpp

float RDataFrame(float val){
      return val * val;
}

ROOT.RDataFrame(2)

4.0
```

Not so great

Conclusion

- To a new user PyROOT is a great for work or to transition to ROOT C++.
- Lots of great tutorials and documentation is provided.
- Operations with strings are powerful and easy `df1.Define("b", "square(a)")`
- Concerns
  - Accessing newly created functions from ROOT could break important functionality without knowing it