

What is ROOT?
Why do we use it?

Answer:

ROOT does what physicists do:

It makes plots.

Before you're insulted, consider my audience

- My ROOT tutorial has been part of Nevis Lab's REU program since 2001.
 - REU = Research Experience for Undergraduates
 - Funded by NSF (US National Science Foundation)
 - Provides a 10-week research program for undergraduates
 - Nevis emphasizes admission for students that come from smaller institutions.
 - Having this experience on their transcripts helps the undergrads when they apply to grad school.
- Most of the students in my tutorial are REU students
 - They all have laptops
 - They might know Python
 - Mostly they don't know programming at all!

My audience (cont'd)

- More over-generalizations
 - They don't have familiarity with a command line
 - “Shell? That's something you find on the sea shore, right?”
 - They tend to save all their files on the Desktop
 - They don't know what a “directory” is.
 - They have no experience with statistics.
 - In particular, they don't know what a histogram is.
 - They probably have worked with an “office suite”.
 - Microsoft Word/Excel/Powerpoint
 - Apple Pages/Numbers/Keynote
 - Google Docs/Sheets/Slides
 - They may also have worked with Mathematica or Matlab.

My official mandate

- This what John Parsons asked me to teach back in 2001. This directive has been unchanged since then:
 - Assume the students are given an ntuple
 - Make histograms of variables in that ntuple
 - Fit a histogram to a function
 - Apply a cut to a variable

What I think the students need

Over the next two days, you will learn how to:

- look up ROOT command references
 - plot a function
 - histogram a variable
 - fit a histogram
 - get a variable from an n-tuple
 - apply cuts
 - do a quick study using TreeViewer (optional)
 - create C++ or python code for an n-tuple
 - use the Jupyter notebook server for quick coding
- but not necessarily in this order!

Realities

- There are several NSF-funded research groups at Nevis
 - In 2022, I'll be teaching students from ATLAS, Neutrino (MicroBooNE, SBND, DUNE), VERITAS/CTA, XENON, Astrochemistry, RARAF.
 - Each individual group's use of ROOT is different.
 - At least one group doesn't use ROOT at all, but send their students to the tutorial to spare the researchers a couple of days before they need to supervise anyone.
 - Some of the groups want their students to learn Python, others want them to learn C++ (because that's the language of their analysis frameworks).
- At most, I'll have the students for three 3-4 hour sessions in their first couple of days at Nevis. After that, they'll be tossed into the wild jungle of research.

Motivating students – my approach

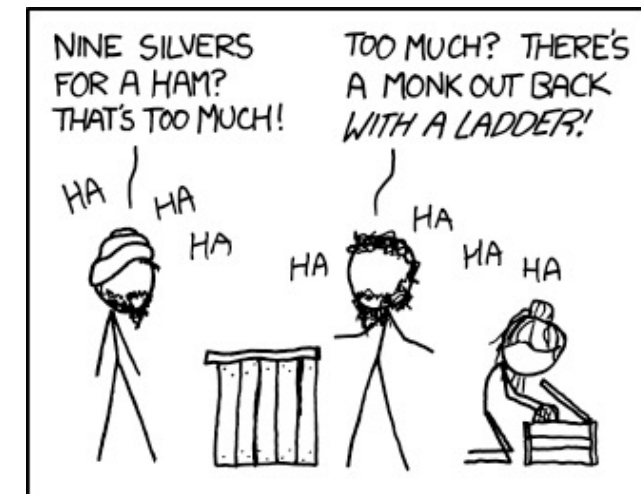
- Justify ROOT
 - e.g., Why not use Mathematica? Excel? Matplotlib?
- Teach them what “histograms” and “ntuples” are, from first principles.
- The same with a “fit”; they may not even seen a linear regression before.
- Describe just enough Linux for the students to complete the tutorial (e.g., `ls`, `cd`, `less`, `man`)
- How to use ROOT documentation.
- Humor
 - Maybe too much, but I think the result is that the students are annoyed at *me* instead of being annoyed at ROOT.

What is ROOT?
Why do we use it?

Answer:

ROOT does what physicists do:

It makes plots.



THERE'S NO REASON TO THINK THAT PEOPLE THROUGHOUT HISTORY DIDN'T HAVE JUST AS MANY INSIDE JOKES AND CATCHPHRASES AS ANY MODERN GROUP OF HIGH-SCHOOLERS.

<https://xkcd.com/794/> by Randall Munroe

Teaching students – my approach

- It's more “workshop” than “tutorial”
 - I have a 15-minute talk to introduce concepts and a simple demo.
 - The rest is having each students work through a written tutorial on their own.
 - We all sit in a room. If the students encounter a problem with the material, they call me over to work one-on-one.
- For the most part, I do *not* give pre-written code examples.
 - By typing in the commands themselves, they inevitably make mistakes... and learn from them.
 - My observation is that the students don't learn much from filled-in notebooks.
 - Pre-existing notebooks are invaluable as references, and I've personally learned quite a bit from the example notebooks on the ROOT site.
 - For a student who doesn't know programming, they're just confusing.

Nevis ROOT resources

- I use [environment modules](#) to distribute ROOT versions on the CentOS 7 systems of the Nevis particle-physics Linux cluster.
 - As of May-2022: ROOT 6.26.00, Python 3.11.0, GCC 11.2.0
 - I compile these packages from source, to offer the latest-and-greatest.
 - Most Nevis researchers use their experiments' analysis frameworks, each of which has their own versions of ROOT, Python, etc. It's typically only the summer students who use `module load root`.
- I have a JupyterHub server with ROOT C++ and Python available to the students, and to the particle-physics research community at Nevis.
 - I installed other Jupyter kernels on the JupyterHub server; e.g., Julia, Octave, R, SageMath. I don't believe anyone at Nevis uses them, but they're there!
 - As of May-2022: ROOT 6.24.06, Python 3.6.8 (the CentOS 7 versions on EPEL).

Nevis ROOT resources (cont'd)

- I specifically instruct my students *not* to attempt to install ROOT on their laptops.
 - At the start, I tell them to use ssh to access a Nevis server and use interactive ROOT. After some initial exercises, I show them our JupyterHub server. They choose how to proceed with the tutorial from there.
 - Reasons:
 - There's not enough time for me to micro-manage the installation of ROOT on heterogenous laptop environments.
 - Even fiddling with package managers like conda would take too long; recall that many students have never used their command-line environment before.
 - If the students want to take the risk, or someone is taking my tutorial who's not at Nevis, I have instructions late in the tutorial on how to install ROOT: conda, docker, singularity...
 - The docker image I point them to is [wgseligman/jupyter-pyroot](#).

Teaching RDataFrame?

- It's an optional part of my tutorial, but it's so far down in the sequence of pages that I suspect none of the students have read it.
- Reasons:
 - RDataFrame doesn't may not make sense if you don't understand the idea of looping over events.
 - If I'm teaching the students basic programming, I have to teach them about for-loops and if statements anyway.
 - To use RDataFrame for anything but the most simple tasks, you need to be moderately skilled at programming (e.g., functors/lambda in C++, decorations/pythonizations in Python). This is beyond what I can teach them (see previous slides).
 - The concept of "lazy evaluation" is another idea that's hard to grasp for new students.
 - As far as I know, none of the research groups at Nevis use RDataFrame in their current analyses. This may change, but until then it may be a waste of the students' time.
- **If I had a week, instead of just 1-1.5 days...**

Manual and Tutorials on the ROOT web site

- Umm... umm... How do I put this?
- Clearly many hours of time and effort were spent on the site. But...
- For the reasons I've listed on the previous slides, they're simply not a useful tool for my students.
 - The ROOT manual and tutorials assume that the reader already knows C++ or Python
 - They assume that the reader has some formal understanding of the difference between a programming statement and an interpreter command (e.g., `.ls`).
 - The code in the tutorials have almost no comments.
 - **SHAME! SHAME! SHAME!**
- I feel that most useful introductory reference is still the paper version of the rapidly-aging [ROOT User's Guide](#). It's easy for a student to skim past the material that's not relevant to them.

As long as I'm whining...

(This is not so much a Train-the-Trainers issue as it is a Developers issue. But it has a training impact.)

- I'm sure ROOT works just fine when you run on it your laptop.
- But when run on a remote server with NFS-mounted filesystems, ROOT versions have grown steadily more sluggish.
 - This creates a lousy first impression for new students in my workshop.
 - I have to remember to set the options in `/${ROOTSYS}/etc/system.rootrc`
 - `ExpandDirectories` to `no`
 - `Browser.Name` to `TRootBrowser` or `TRootBrowserLite`
 - If you leave the current default of `ROOT::Experimental::RWebBrowserImp`, and start `TBrowser`, there'll be very long delay as it starts a web browser *on the remote system* and pumps the graphics to your laptop.
 - Yesterday, Serguei Linev taught us how to use `RBrowser` remotely via ssh port forwarding. Again, this is too much for our students.
 - Even with `ExpandDirectories` set to “no”, `TBrowser` now takes a long time to start up; about 30 seconds on my home Macintosh!
 - Of course, maybe this is just me.

What's next for the Nevis ROOT tutorial

- I'm teaching the incoming group on June 1-2, 2022.
 - Due to COVID, I'll have to teach via Zoom again this year.
- I've switched from using a PDF file to using web pages.
 - Sphinx document management
 - Pages written in Markdown variants MyST and reStructuredText
 - ReadTheDocs theme
- I've learned things from the 2022 ROOT Users Workshop. They'll find their way into my 2023 tutorial.
 - Who knows? I could figure out a way to teach about conda and RDataFrames to the REU students without confusing them.
 - Maybe Lorenzo Moneta's and Enrico Guiraud's talks will change my approach. I hope so!
- The 2022 version of my tutorial will be posted here within the next couple of weeks:

<https://www.nevis.columbia.edu/~seligman/root-class/>