# The ROOT Project at the end of Run 3 and towards HL-LHC

*Danilo Piparo[1] for the ROOT Project*

*24-10-2024*

*[1] CERN EP-SFT*

*24.10.24, CHEP `24, Kraków, Poland*

# This Talk

- The ROOT Project, today
- Achievements, current work and the future
- Conclusions

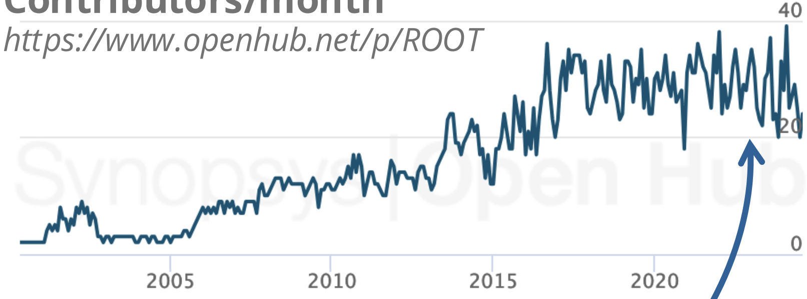
# The ROOT Project, Today

# ROOT's Pulse

- Supported by a lively developer and user community, widely used
- Long-term support model
- **ROOT's future is made of innovation, usability and reliability! (and lots of data)**

**Contributors/month**

*https://www.openhub.net/p/ROOT*



**Many non-core developers contribute to ROOT: your contributions are vital for the project and are very welcome!**

## Some user support metrics, 2024

- ~10k posts per year on the [ROOT Forum]
    - 6h to give a first answer
- Thousands of messages on Mattermost

# Who is ROOT?

**ROOT is its user community, contributors, and developers**

Core Team from https://root.cern , PhD students onwards

- **ROOT is an international collaboration**
- Steady contributions coming from the community, and institutional responsibilities.

From left to right, starting from the first row (the affiliation is CERN if not specified):

1. Devajith Valaparambil, Bertrand Bellenot, Danilo Piparo, Florine de Geus **(CERN & Uni Twente)**, Jakob Blomer
2. Jonas Hahnfeld **(CERN & Uni Frankfurt)**, Jonas Rembser, Marta Czurylo, Olivier Couet, Philippe Canal **(FNAL)**,
3. Vassil Vassilev **(Princeton)**, Lorenzo Moneta, Monica Dessole, Vincenzo Padulano, Serguei Linev **(GSI)**
4. Giacomo Parolini, Stephan Hageböck, Martin Føll **(Uni Oslo)**

Plus students, working with us a few months, up to ~1 year

*Not everybody in this slide is 100% dedicated to one project, but most are.*

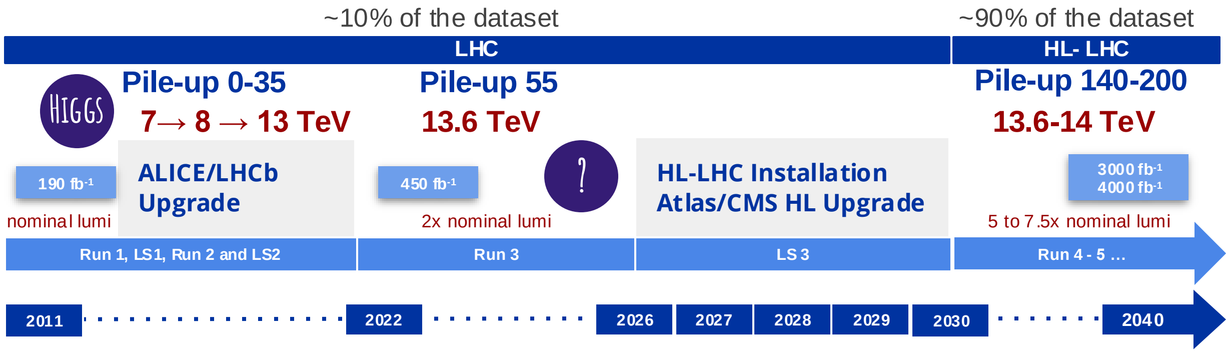
***Many of us are at CHEP: we are eager to talk to you!***

# ROOT's Strategic Goals

**Provide a unified software package for the storage, processing, visualisation and analysis of scientific data that is reliable, performant, supported and sustainable, that is easy to use and obtain, and that minimises computing resources and scientists' time needed to achieve results.**

**The success of experiments and all ROOT users at large is our priority**

# The Need of Strategic Thinking



~10% of the dataset

~90% of the dataset

LHC

HL- LHC

Pile-up 0-35

7→ 8 → 13 TeV

Pile-up 55

13.6 TeV

Pile-up 140-200

13.6-14 TeV

190 $\text{fb}^{-1}$

nominal lumi

ALICE/LHCb Upgrade

450 $\text{fb}^{-1}$

2x nominal lumi

HL-LHC Installation Atlas/CMS HL Upgrade

3000 $\text{fb}^{-1}$
4000 $\text{fb}^{-1}$

5 to 7.5x nominal lumi

Run 1, LS1, Run 2 and LS2 | Run 3 | LS3 | Run 4 - 5 …

2011 · · · 2022 · · · 2026 | 2027 | 2028 | 2029 | 2030 · · · 2040

*HL-LHC timescale: the EIC will also start producing data!*

| QoS | ALICE | ATLAS | CMS | LHCb | *Total* |
|---|---|---|---|---|---|
| Disk [PB] | 199 | 406 | 304 | 93 | 1002 |
| Tape [PB] | 283 | 666 | 673 | 250 | 1875 |

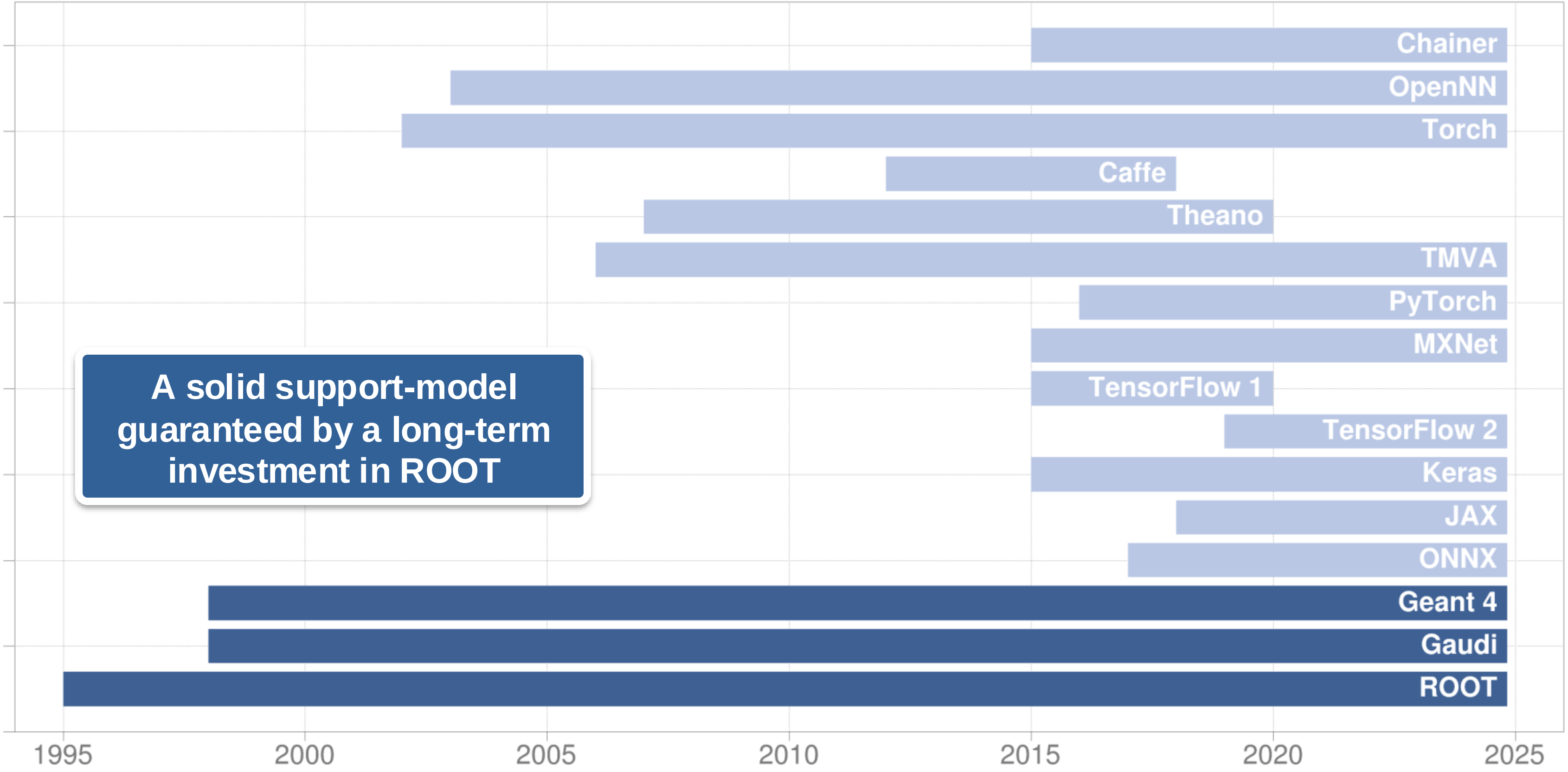
*`24 Pledges: source WLCG CRIC*

**The full exploitation of the physics potential of present and future accelerators also passes through ROOT**

LHC Runs 1,2,3 → **Today, >2 EB in ROOT format**

**HL-LHC: ~30 EB in ROOT format?**

# HEP Common Software Support Timeline

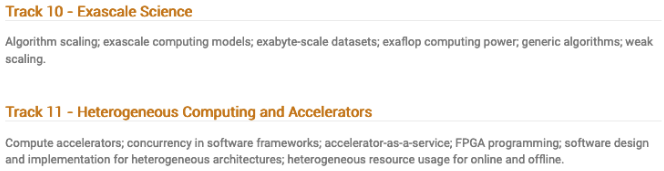


A solid support-model guaranteed by a long-term investment in ROOT

*Plot inspired by M. Mazurek*

# Hardware, Infrastructure, Policy are evolving

- **The CPU-only era, with faster cores every year is over for more than a decade**
- **HEP successfully embraced the multicore and GPUs paradigm shift**
- At the infrastructure scale, even more is changing, e.g. HPCs
  - **Substantial investments, HPCs are there to stay in our computing infrastructure**
  - Virtually all HPCs express most of their computing power through accelerators
- Programmable GPGPUS: **is the market converging towards AI-oriented computing?**
- HEP always cared about energy consumption, e.g. driven by economic factors
  - **Are we approaching an era where energy-aware computing is enforced by policy?**

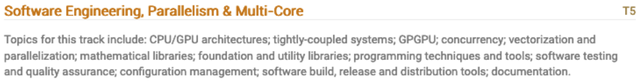


Track 10 - Exascale Science

Algorithm scaling; exascale computing models; exabyte-scale datasets; exaflop computing power; generic algorithms; weak scaling.

Track 11 - Heterogeneous Computing and Accelerators

Compute accelerators; concurrency in software frameworks; accelerator-as-a-service; FPGA programming; software design and implementation for heterogeneous architectures; heterogeneous resource usage for online and offline.



Software Engineering, Parallelism & Multi-Core T5

Topics for this track include: CPU/GPU architectures; tightly-coupled systems; GPGPU; concurrency; vectorization and parallelization; mathematical libraries; foundation and utility libraries; programming techniques and tools; software testing and quality assurance; configuration management; software build, release and distribution tools; documentation.

**Tough challenges ahead, that we can only face together, as a community**

# R&D in ROOT Becomes Production Quality

## The ROOT project is a veritable platform to make R&D blossom

- Cling (C++ interpreter) and Clang-REPL ('lightweight Cling')
- Automatic differentiation with Clad, now in RooFit
- RNTuple – 6y of R&D, now transitioning to production
- New thread friendly histograms

**ROOT: a project where result-oriented R&D activities can blossom and reach thousands of scientists.**

Also supported by supranational funding:

- HighLO: Project **High** Energy Physics Tools in **L**imit **O**rder Book Analysis (HighLO) applies particle physics methods and tools to financial market data.
- SYCLOPS (EU): Advancing AI/data mining for extremely large and diverse data for Europe and beyond, by democratizing its acceleration through open standards.
- CERN Experimental Physics Department R&D (EP R&D): innovative I/O formats (RNTuple), Python-C++ interoperability, Analysis at scale, Key4hep

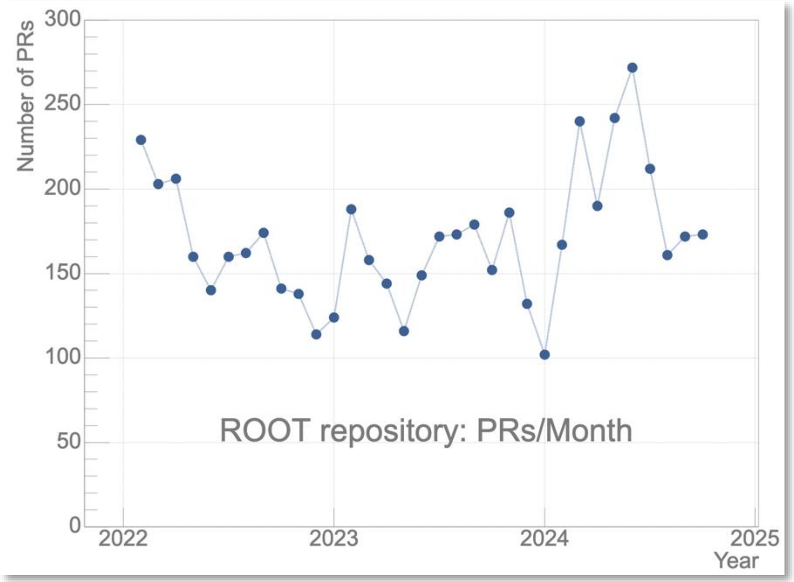
# An *Open* Approach to Boost Collaboration

**Open-source and Open-development**

- On GitHub, LGPL 2.1
- PR based model with public review process
- **Very visible authorship of contributions**

**Open-planning**: https://cern.ch/root-pow

- Yearly plan of work (PoW) formed internally
  - Discussed with users and then refined



- **You can influence the PoW, with your input, active engagement and contributions!**
  - E.g. integration and commissioning of RNTuple by ROOT and experiments Core SW teams
- Formal reporting process engaging experiments
- Hackathons, Workshops, Topical Meetings

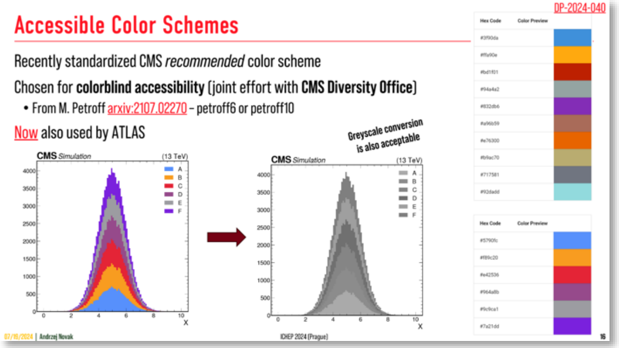
# Successful Collaborations with Users and Experiments

**Engagement and collaboration historically part of the ROOT Project.** Recent examples:

- PyHep(Dev) and WLCG&HSF events, conferences
- (Co-)Supervision with experiments, e.g. of IRIS-HEP fellows

  (e.g. [1], [2],[3]) and students
- Stakeholders engagement process with quarterly reports
- Frequent meetings with LHC experiments Core SW teams
- Regular updates during '(O&C) Weeks' of experiments

*Common Analysis Tools in CMS*, A. Nowak, ICHEP24



DP-2024-040

## Accessible Color Schemes

Recently standardized CMS *recommended* color scheme

Chosen for **colorblind accessibility** (joint effort with **CMS Diversity Office**)

- From M. Petroff arxiv:2107.02270 – petroff6 or petroff10

Now also used by ATLAS

Greyscale conversion is also acceptable

| Hex Code | Color Preview |
|---|---|
| #3f90da | 🟦 |
| #ffa90e | 🟧 |
| #bd1f01 | 🟥 |
| #94a4a2 | ⬜ |
| #832db6 | 🟪 |
| #a96b59 | 🟫 |
| #e76300 | 🟧 |
| #b9ac70 | 🟨 |
| #717581 | ⬛ |
| #92dadd | 🟦 |

| Hex Code | Color Preview |
|---|---|
| #5790fc | 🟦 |
| #f89c20 | 🟧 |
| #e42536 | 🟥 |
| #964a8b | 🟪 |
| #9c9ca1 | ⬜ |
| #7a21dd | 🟪 |

07/19/2024 | Andrzej Novak

ICHEP 2024 (Prague)

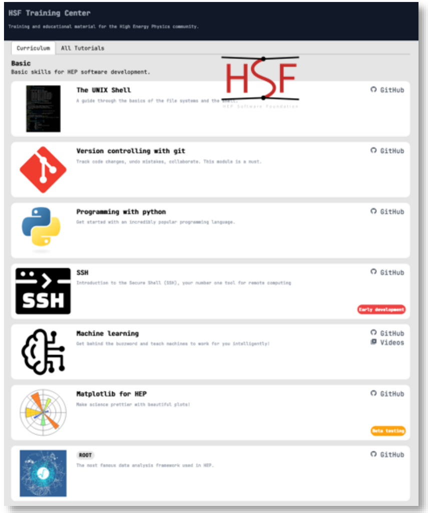
16

One success story out of many: new colour schemes implemented in ROOT in coordination with CMS CAT

**Your input and collaboration is essential!**

# Scaling up Training

- CERN Summer Student Course ~200 participants, ~5 per year
- With IRIS-HEP and HSF: Python for analysis course ~90 participants, ~3 per year
- **Based exclusively on ROOT's Python interface and notebooks**
- Besides the value of the trainings themselves:
  - Surveys: feedback received now incorporated in the material
  - Several ROOT devs involved: we trained to train!



HSF Training Center

Training and educational material for the High Energy Physics community.

Curriculum | All Tutorials

**Basic**

Basic skills for HEP software development.

**The UNIX Shell** — A guide through the basics of the file systems and the shell. — GitHub

**Version controlling with git** — Track code changes, undo mistakes, collaborate. This module is a must. — GitHub

**Programming with python** — Get started with an incredibly popular programming language. — GitHub

**SSH** — Introduction to the Secure Shell (SSH), your number one tool for remote computing — GitHub — Early development

**Machine learning** — Get behind the buzzword and teach machines to work for you intelligently! — GitHub, Videos

**Matplotlib for HEP** — Make science prettier with beautiful plots! — GitHub — Beta testing

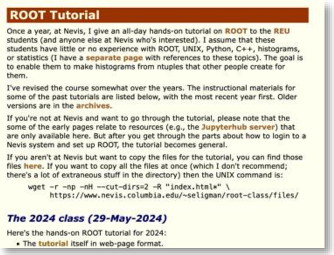
**ROOT** — The most famous data analysis framework used in HEP. — GitHub

HSF Training Centre

Open in SWAN

[Try it on SWAN (CERN's Jupyter service) now!]

**Interested in a ROOT Training? Talk to us!**



**ROOT Tutorial**

Once a year, at Nevis, I give an all-day hands-on tutorial on ROOT to the REU students (and anyone else at Nevis who's interested). I assume that these students have little or no experience with ROOT, UNIX, Python, C++, histograms, or statistics (I have a separate page with references to these topics). The goal is to enable them to make histograms from ntuples that other people create for them.

I've revised the course somewhat over the years. The instructional materials for some of the past tutorials are listed below, with the most recent year first. Older versions are in the archives.

If you're not at Nevis and want to go through the tutorial, please note that the some of the early pages relate to resources (e.g., the Jupyterhub server) that are only available here. But after you get through the parts about how to login to a Nevis system and set up ROOT, the tutorial becomes general.

If you aren't at Nevis but want to copy the files for the tutorial, you can find those files here. If you want to copy all the files at once (which I don't recommend; there's a lot of extraneous stuff in the directory) then the UNIX command is:

```
wget -r -np -nH --cut-dirs=2 -R "index.html*" \
     https://www.nevis.columbia.edu/~seligman/root-class/files/
```

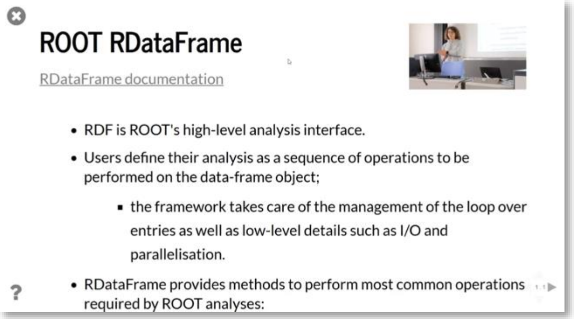
***The 2024 class (29-May-2024)***

Here's the hands-on ROOT tutorial for 2024:

- The tutorial itself in web-page format.

*Several ROOT trainings at institutes/labs (e.g. at Columbia)*

A new ROOT Video Course is available on CDS: check it out!

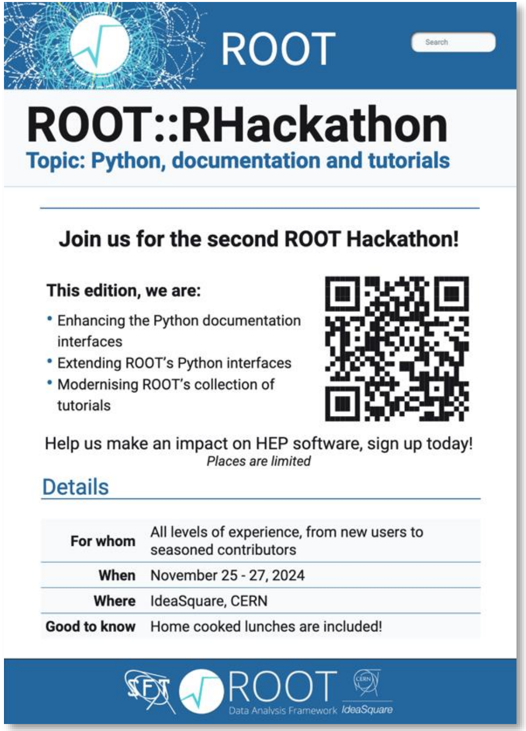


ROOT RDataFrame

RDataFrame documentation

- RDF is ROOT's high-level analysis interface.
- Users define their analysis as a sequence of operations to be performed on the data-frame object;
  - the framework takes care of the management of the loop over entries as well as low-level details such as I/O and parallelisation.
- RDataFrame provides methods to perform most common operations required by ROOT analyses:

# ROOT Hackathons: new in 2024


ROOT
Search
ROOT
Data Analysis Framework
IdeaSquare

## ROOT::RHackathon

**Topic: Python, documentation and tutorials**

**Join us for the second ROOT Hackathon!**

**This edition, we are:**

- Enhancing the Python documentation interfaces
- Extending ROOT's Python interfaces
- Modernising ROOT's collection of tutorials

Help us make an impact on HEP software, sign up today!

*Places are limited*

### Details

| | |
|---|---|
| **For whom** | All levels of experience, from new users to seasoned contributors |
| **When** | November 25 - 27, 2024 |
| **Where** | IdeaSquare, CERN |
| **Good to know** | Home cooked lunches are included! |

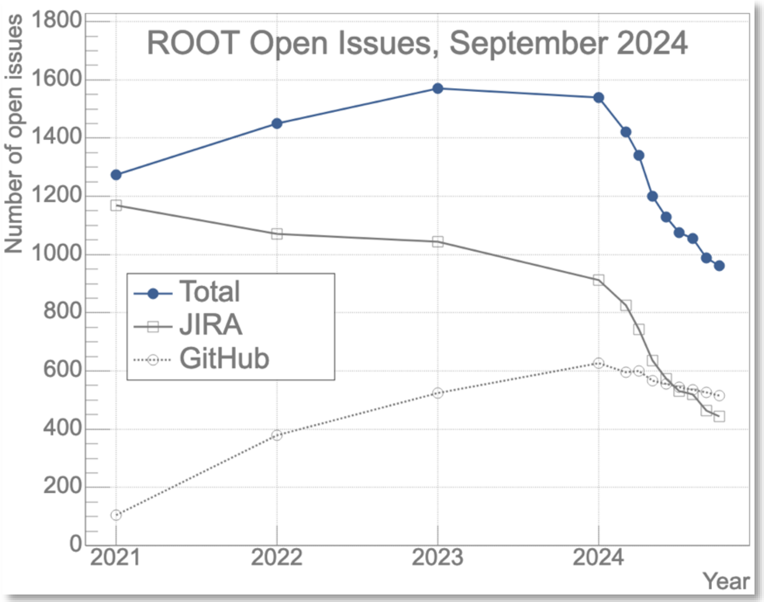
- **2nd ROOT Hackathon at Idea Square at CERN: Nov 25-27**
  - ~30 slots, for all levels of experience: from students to senior
  - An event born for ROOT core devs, open to everybody
- **A welcoming, positive and inclusive atmosphere**
- Topic: '**Python, Docs and Tutorials**'. Participants in 3 groups:
  - Python interface documentation enhancement
  - Python interface extensions ('Pythonisations')
  - ROOT tutorials modernisation campaign
- You'll be able to pick the group you prefer
- Lunches prepared by the ROOT core devs

**Register here if you can be there and are interested!**

More about the 1st ROOT Hackathon: the Fixathon, at Idea2 CERN on Feb 13-14

# Stability

- Strong focus on reducing number of open issues
- Backlog reduction is implicitly part of the PoW
- 1536 issues on Jan 1st, 961 on Sep 30th: -38%
- **Crucial effort to enable sustainable innovation**



**Stability is a priority, and a requirement for future evolution**

# Highlights of Current Activities and Future Directions

# An Evolutionary Approach Leading to ROOT 7

*La Liberté guidant le peuple, E. Delacroix*, from Wikipedia

***Is a revolution always better than an evolution?***

# An Evolutionary Approach Leading to ROOT 7

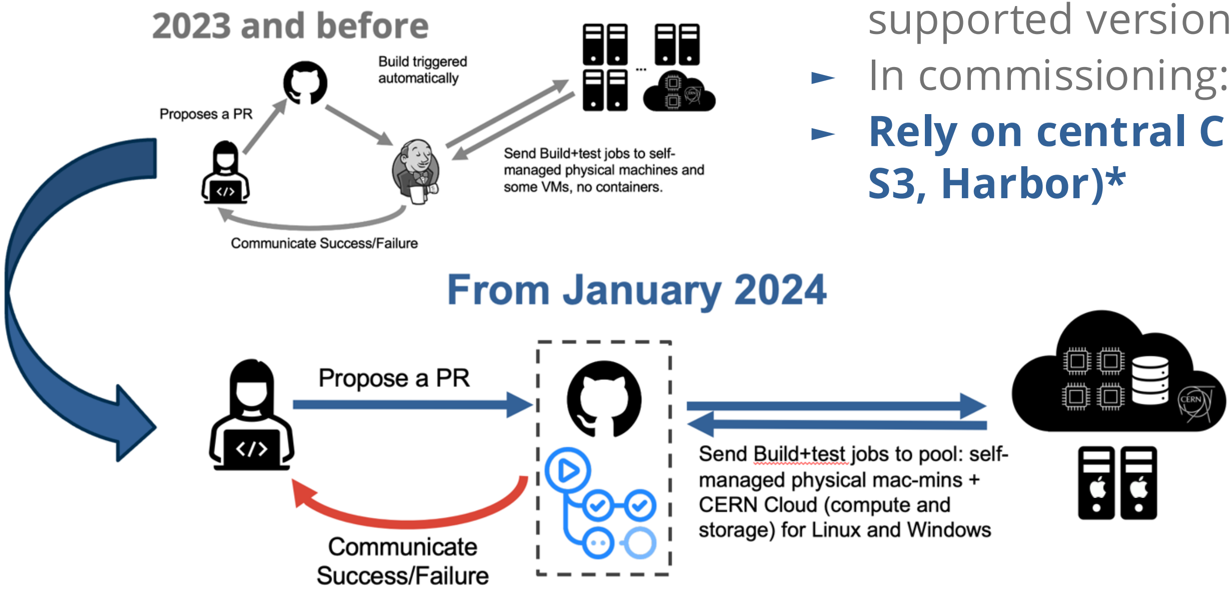
**ROOT has to evolve to meet the challenges of future scientific computation**

- Possible only by **dropping some of its older components or changing some behaviour**
  - E.g. is automatic memory management still needed?
- **Not a revolution, but a piecewise renovation**, leading to a completely new system
  - Not new: code using ROOT today has little to do with the one written in early Run 2
- New components are being introduced and adopted *today*
  - `RNTuple`, `RDataFrame`, new *Pythonic Interface*, completely new `RooFit`, …
- With the early adoption of new components, **the jump to ROOT 7 will not be large**
  - **Some changes will be backwards incompatible**, but a much smaller jump than ROOT5 →ROO6
  - Migration or transition paths will be documented and clearly communicated
- **ROOT 7.00.00 will be released during LS3**, well on time for Run 4 MC productions

# The Basis: A new CI System in Support of Stability

☞ ROOT's new Cost-efficient and Feature Rich GitHub-based CI

- Moved from Jenkins to GitHub Actions
- **Behave like many other open source products**
- **Reduces overall maintenance costs**
- 30-50 build+~3000 tests jobs per day
- Linux ~10 flavours (containers), macOS (all supported versions + beta), Windows
- In commissioning: Linux ARM and GPU nodes
- **Rely on central CERN services (e.g. Openstack, S3, Harbor)***


2023 and before
Build triggered automatically
Proposes a PR
Send Build+test jobs to self-managed physical machines and some VMs, no containers.
Communicate Success/Failure
From January 2024
Propose a PR
Send Build+test jobs to pool: self-managed physical mac-mins + CERN Cloud (compute and storage) for Linux and Windows
Communicate Success/Failure

** Thanks to the CERN IT-CD group for this scalable, powerful and reliable services!*

# ROOT continues to be Part of the Python Ecosystem

**The quality of the ROOT experience for Python users is a priority**

Example actions taken in 2024:

- Update to the latest version of Cppyy, ROOT's C++-Py 'interoperability engine'
- Provide a demo infrastructure to `pip install ROOT`
- Improve the usage of several classes from Python through "pythonisations"
- **Teach ROOT through its Python interface, especially for beginners' courses**

More actions are planned for the future, e.g.:

- Revisit Python tutorials and code examples
- Improve and extend the Python interface through pythonisations
- Evolve pip install ROOT to Beta mode during 2025 (e.g. automatic publication of wheels, multiple wheels, ...

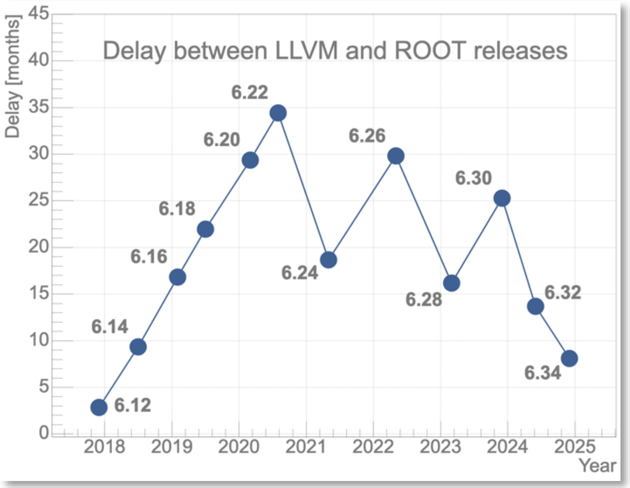
☞ pip install ROOT

**Clear direction: reinforce ROOT's presence in the Python ecosystem, prioritising the Python users experience**

# C++ and Python Interpreters

## Aim to a sustainable interpreters' infrastructure

- **Further simplify the LLVM-based Cling C++ interpreter and the Python compatibility layer**
  - E.g. through CppInterOp, that exposes APIs from Clang and LLVM in a backward compatible way
- **Stay even more up-to-date with LLVM versions**
  - E.g. ROOT 6.34 (next month) will be powered by LLVM 18 (March)
  - Reduce the complexity of the upgrades with smaller jumps
  - Exploit new features, e.g. performance, C++ standards
- Upstream Cling features to the LLVM repo with Clang-Repl
  - ***R**ead-**E**valuate-**P**rint-**L**oop*: 'a simpler Cling'
  - In the LLVM repository, with full test coverage



# RDataFrame: the Entry Point for ROOT Data Analysis

## RDataFrame: high level interface for data analysis

- **Processing of columnar datasets (and more), including TTree, and RNTuple**
- Available in Python and C++, simplicity and runtime performance are its two pillars
- **Widely adopted,** also as engine for specialised frameworks (e.g. bamboo, CROWN)

► Same code runs on 1 core, N cores (multithreading) or on a distributed system

- Optimisations, remote reads, Jitting: all handled internally

► RDataFrame runs everywhere, e.g. CERN Lxplus, bare-metal clusters, HPCs

- **Fits well the Analysis Facility approach**, see e.g. [1], [2], [3], [4] and Track 9!

► **Top performance/scaling achieved: shifting focus back on usability from Python**

Bamboo: A high-level HEP analysis library for RDataFrame

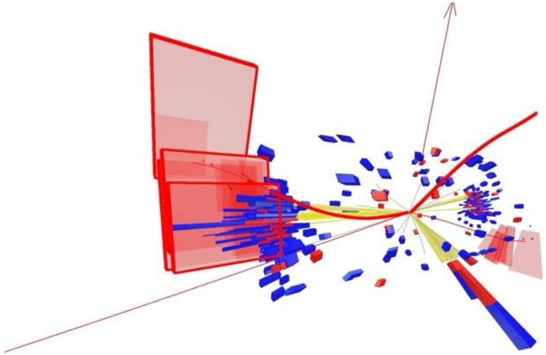
Distributed analysis in production with RDataFrame

**ROOT is ready to run at your favourite Analysis Facility, today**

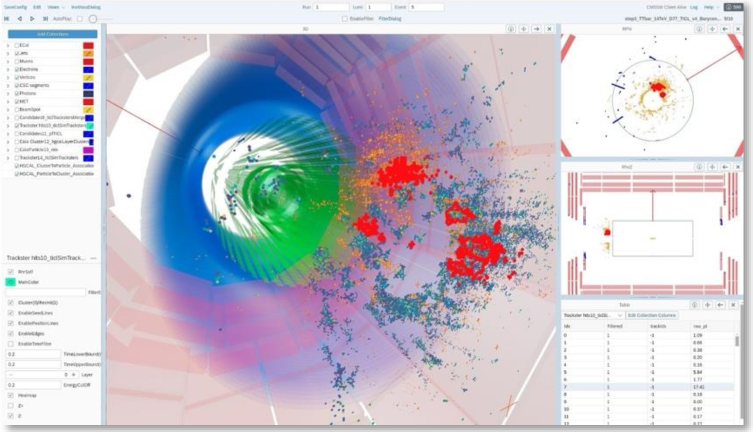
# Event Display: REve

## REve: ROOT-based event display

- Based on ROOT Web Graphics
  - Improved quality and stability during the past year
  - Default for Jupyter Notebooks in ROOT 6.34
- **Access to data formats: e.g. object filtering**
  - Exactly the same as it would be in physics analysis!
- Object selection, across all views, even into large collections (e.g. CMS HGCal rec-hits)



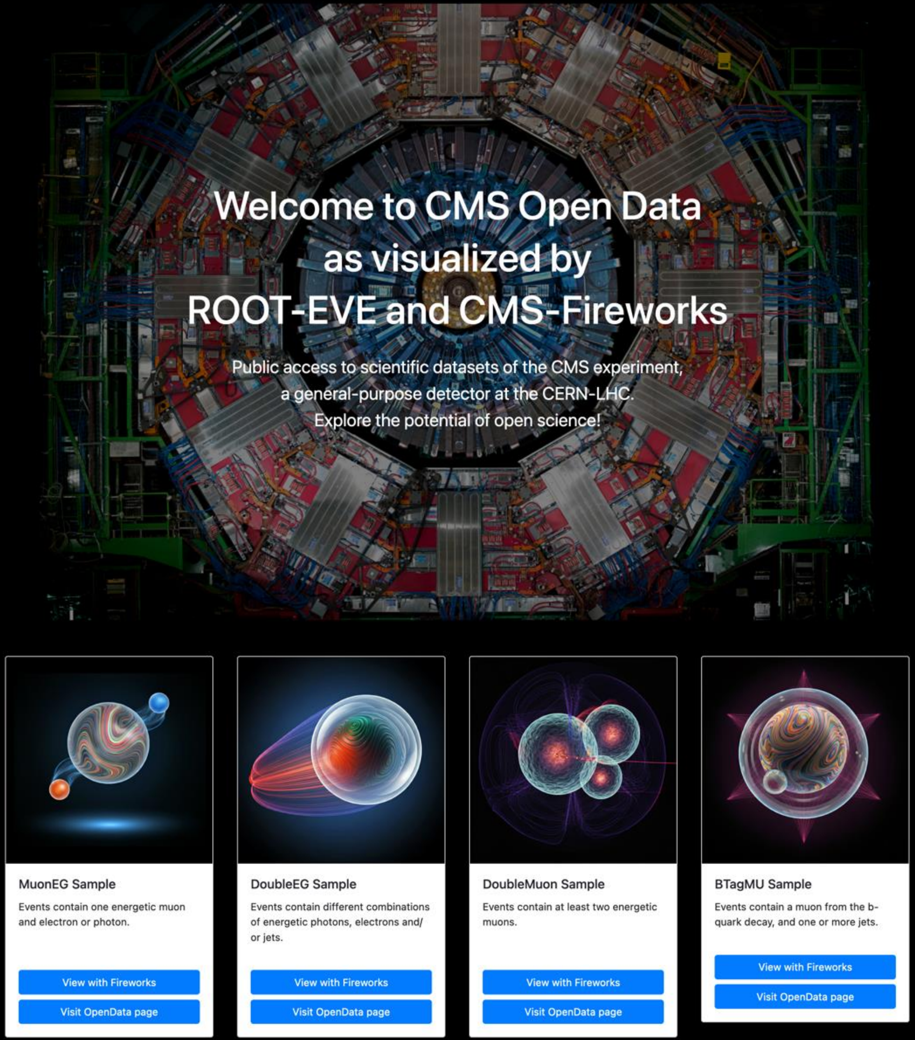
☞ Web-based Graphics in ROOT



**Try it now https://fireworks-open.cern.ch**

**Currently working closely with CMS OpenData, towards a potential adoption for Run 4**

**https://fireworks-open.cern.ch**



Welcome to CMS Open Data as visualized by ROOT-EVE and CMS-Fireworks

Public access to scientific datasets of the CMS experiment, a general-purpose detector at the CERN-LHC. Explore the potential of open science!

MuonEG Sample

Events contain one energetic muon and electron or photon.

DoubleEG Sample

Events contain different combinations of energetic photons, electrons and/ or jets.

DoubleMuon Sample

Events contain at least two energetic muons.

BTagMU Sample

Events contain a muon from the b-quark decay, and one or more jets.

# RNTuple: the Future HEP Data Format

- ► **Part of current ROOT releases, approaching production, on schedule and budget**
- ► Migration from TTree requires changes for experiment frameworks
  - **Drop-in replacement for RDataFrame based analyses** (no user code changes)

**Based on 25+ years of TTree experience**

- ► **Less storage, compute and network needs**
- ► Systematic use of **data checksums** and efficient support of **modern hardware.**

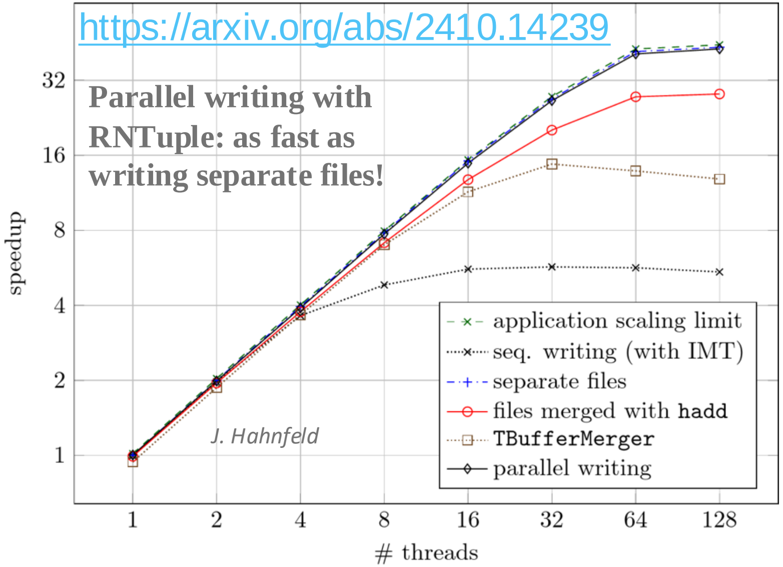
**Advanced status of integration in LHC experiments frameworks: on track for Run 4**

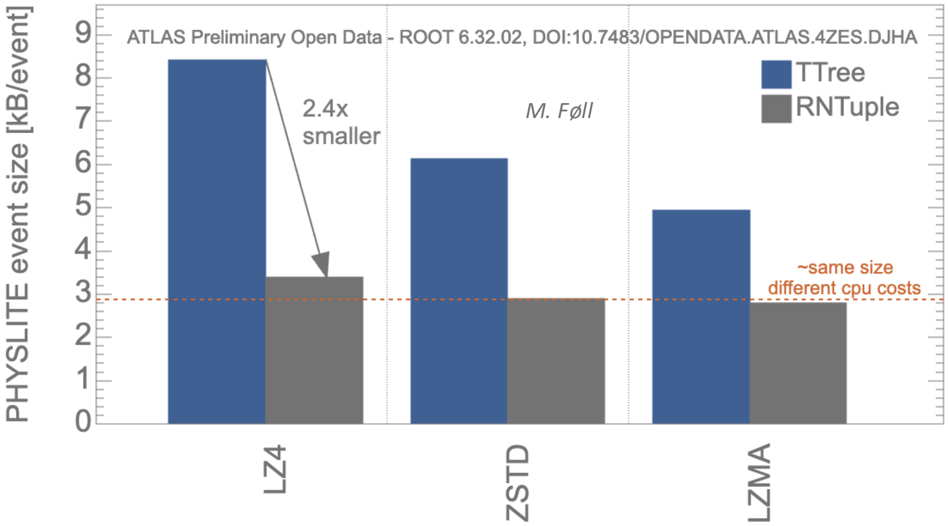
- ► **We are grateful to experiments for their hard work and our fruitful collaboration!**

Ongoing RNTuple API formal Review by HEP-CCE (Center for Computational Excellence)
Mid-term report delivered: excellent recommendations, being worked on already

U.S. DEPARTMENT OF ENERGY | Office of Science
Argonne NATIONAL LABORATORY
BROOKHAVEN NATIONAL LABORATORY
Fermilab
BERKELEY LAB Bringing Science Solutions to the World

# RNTuple: Some Results





**Faster, smaller files, more scalable and sustainable**

- Direct I/O for RNTuple Columnar Data
- On-the-fly concatenations and joins with ROOT RNTuple
- RNTuple in the ATLAS production framework Athena
- RNTuple & EOS: Comparative Analysis of Physics Data Formats
- Thread-safe N-tuple Writing in Gaudi with TTree and Migration to RNTuple

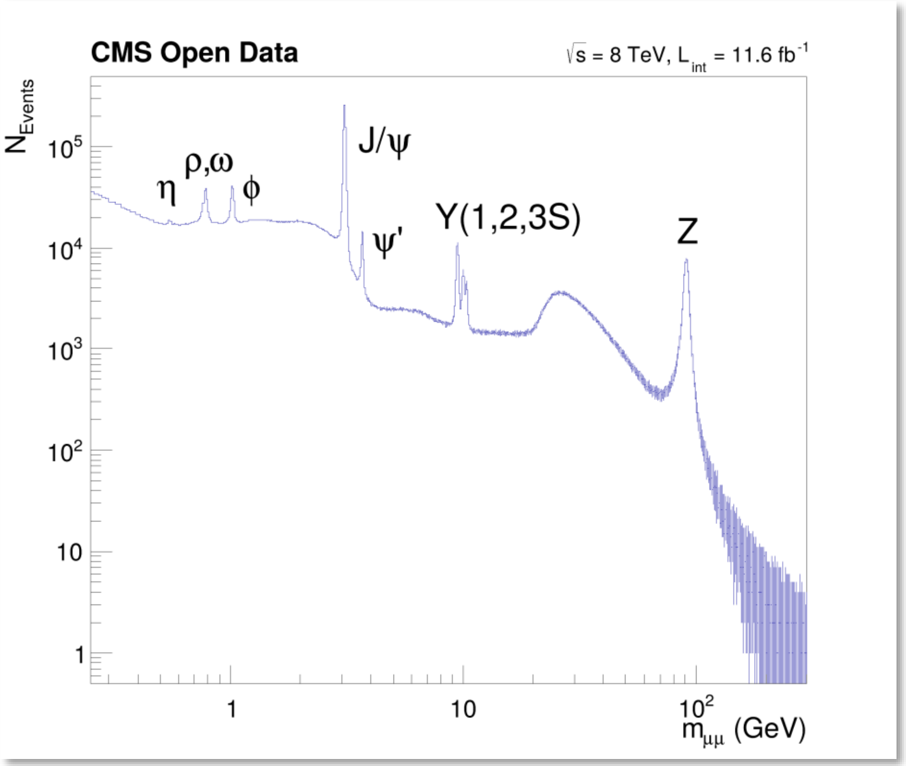
- RNTuple implementation in Julia
- ATLAS software tools to handle ROOT RNTuple
- RNTuple and EOS: The Next Generation of Event Data I/O
- RNTuple: A CMS Perspective

# TTree - RDataFrame Analysis

```python
import ROOT as rt

rt.EnableImplicitMT()

# Create dataframe from NanoAOD files
dataset = 'root://eospublic.cern.ch//eos/opendata/cms/derived-data/'\
          'AOD2NanoAODOutreachTool/Run2012BC_DoubleMuParked_Muons.root'
df = rt.RDataFrame ('Events', dataset)

# Select only events with exactly two muons and require opposite charge
df_2mu = df.Filter('nMuon == 2',
                   'Events with exactly two muons')
df_os = df_2mu.Filter('Muon_charge[0] != Muon_charge[1]',
                      'Muons with opposite charge')

# Compute invariant mass of the dimuon system
df_mass = df_os.Define('Dimuon_mass',
                       'InvariantMass(Muon_pt, Muon_eta, Muon_phi, Muon_mass)')

# Make histogram of dimuon mass spectrum. Note how we can set titles and axis labels in one go.
h = df_mass.Histo1D(('', 'Dimuon mass;m_{#mu#mu} (GeV);N_{Events}',
                     30000, 0.25, 300), 'Dimuon_mass')
# Request cut-flow report
report = df_mass.Report()

# Produce plot
rt.gStyle.SetOptStat(0); rt.gStyle.SetTextFont(42)
c.SetLogx(); c.SetLogy()
h.GetXaxis().SetTitleSize(0.04)
h.GetYaxis().SetTitleSize(0.04)
h.Draw()

label = rt.TLatex(); label.SetNDC(True)
label.DrawLatex(0.175, 0.740, '#eta'); label.DrawLatex(0.205, 0.775, '#rho,#omega')
label.DrawLatex(0.270, 0.740, '#phi'); label.DrawLatex(0.400, 0.800, 'J/#psi')
label.DrawLatex(0.415, 0.670, '#psi'); label.DrawLatex(0.485, 0.700, 'Y(1,2,3S)')
label.DrawLatex(0.755, 0.680, 'Z')
label.SetTextSize(0.04); label.DrawLatex(0.10, 0.92, '#bf{CMS Open Data}')
label.SetTextSize(0.03); label.DrawLatex(0.63, 0.92, '#sqrt{s} = 8 TeV, L_{int} = 11.6 fb^{-1}')
```

**TTree Version**

For more details see the di-muon analysis tutorial

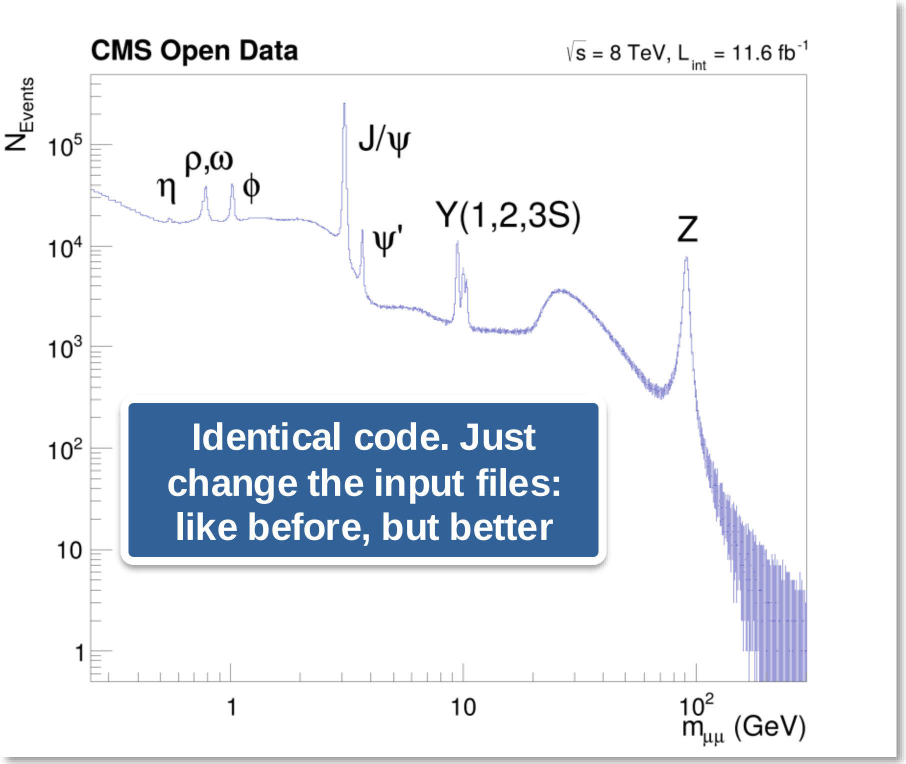


# RNTuple - RDataFrame Analysis

```python
import ROOT as rt

rt.EnableImplicitMT()

# Create dataframe from NanoAOD files
dataset = 'http://root.cern/files/tutorials/ntpl004_dimuon_v1rc2.root'

df = rt.RDataFrame ('Events', dataset)

# Select only events with exactly two muons and require opposite charge
df_2mu = df.Filter('nMuon == 2',
                   'Events with exactly two muons')
df_os = df_2mu.Filter('Muon_charge[0] != Muon_charge[1]',
                      'Muons with opposite charge')

# Compute invariant mass of the dimuon system
df_mass = df_os.Define('Dimuon_mass',
                       'InvariantMass(Muon_pt, Muon_eta, Muon_phi, Muon_mass)')

# Make histogram of dimuon mass spectrum. Note how we can set titles and axis labels in one go.
h = df_mass.Histo1D(('', 'Dimuon mass;m_{#mu#mu} (GeV);N_{Events}',
                     30000, 0.25, 300), 'Dimuon_mass')
# Request cut-flow report
report = df_mass.Report()

# Produce plot
rt.gStyle.SetOptStat(0); rt.gStyle.SetTextFont(42)
c.SetLogx(); c.SetLogy()
h.GetXaxis().SetTitleSize(0.04)
h.GetYaxis().SetTitleSize(0.04)
h.Draw()

label = rt.TLatex(); label.SetNDC(True)
label.DrawLatex(0.175, 0.740, '#eta'); label.DrawLatex(0.205, 0.775, '#rho,#omega')
label.DrawLatex(0.270, 0.740, '#phi'); label.DrawLatex(0.400, 0.800, 'J/#psi')
label.DrawLatex(0.415, 0.670, '#psi'); label.DrawLatex(0.485, 0.700, 'Y(1,2,3S)')
label.DrawLatex(0.755, 0.680, 'Z')
label.SetTextSize(0.04); label.DrawLatex(0.10, 0.92, '#bf{CMS Open Data}')
label.SetTextSize(0.03); label.DrawLatex(0.63, 0.92, '#sqrt{s} = 8 TeV, L_{int} = 11.6 fb^{-1}')
```

## RNTuple Version

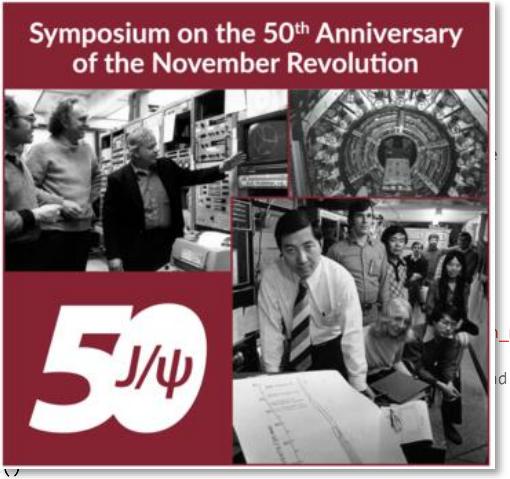
For more details see the di-muon analysis tutorial



**Identical code. Just change the input files: like before, but better**

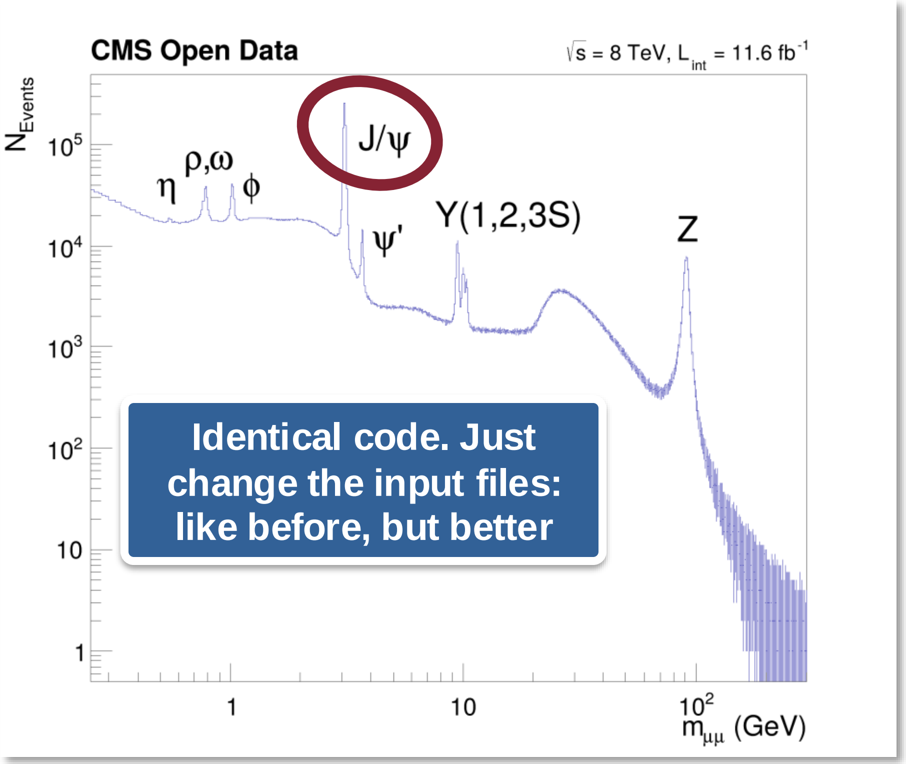
# RNTuple - RDataFrame Analysis

```
import ROOT as rt

rt.EnableImplicitMT()

# Create dataframe from
dataset = 'http://root.

df = rt.RDataFrame ('Ev

# Select only events wi                                                     e
df_2mu = df.Filter('nMu
                   'Eve
df_os = df_2mu.Filter('
                      '


# Compute invariant mas
df_mass = df_os.Define(
                                                                         _mass)')


# Make histogram of dim                                                  d axis labels in one go.
h = df_mass.Histo1D((''
                     30
# Request cut-flow repo
report = df_mass.Report()

# Produce plot
rt.gStyle.SetOptStat(0); rt.gStyle.SetTextFont(42)
c.SetLogx(); c.SetLogy()
h.GetXaxis().SetTitleSize(0.04)
h.GetYaxis().SetTitleSize(0.04)
h.Draw()

label = rt.TLatex(); label.SetNDC(True)
label.DrawLatex(0.175, 0.740, '#eta'); label.DrawLatex(0.205, 0.775, '#rho,#omega')
label.DrawLatex(0.270, 0.740, '#phi'); label.DrawLatex(0.400, 0.800, 'J/#psi')
label.DrawLatex(0.415, 0.670, '#psi'); label.DrawLatex(0.485, 0.700, 'Y(1,2,3S)')
label.DrawLatex(0.755, 0.680, 'Z')
label.SetTextSize(0.04); label.DrawLatex(0.10, 0.92, '#bf{CMS Open Data}')
label.SetTextSize(0.03); label.DrawLatex(0.63, 0.92, '#sqrt{s} = 8 TeV, L_{int} = 11.6 fb^{-1}')
```


Symposium on the 50th Anniversary of the November Revolution
50 J/ψ

## RNTuple Version

For more details see the di-muon analysis tutorial



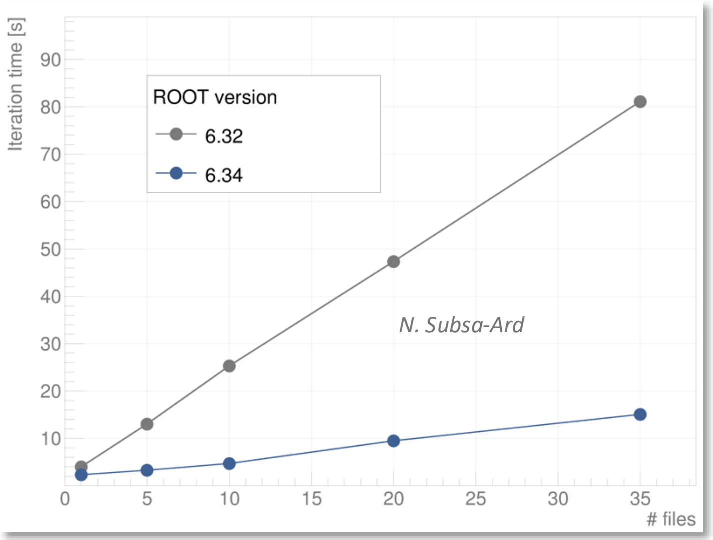
Identical code. Just change the input files: like before, but better

# Machine Learning and ROOT

## Interoperate with existing ML frameworks, do not compete

- **Provide efficient delivery of batched data for training**
  - Read datasets in ROOT format (TTree and RNTuple): no conversion, any data model, remote read capabilities, caching
  - Re-shape input with RDataFrame before delivery to the training
  - All with ergonomic interfaces, easily integration
- TMVA: maintained to support existing user community
- **ROOT will continue to provide SOFIE**
  - Generates C++ code for trained neural network models (ONNX)
  - Stable, simple, long-term supported inference in C++ w/o external dependencies
- **ML effort in ROOT being consolidated in synergy with CERN's ML4EP project**

☞ Zero-overhead training of machine learning models with ROOT data



☞ Benchmark Studies of ML Inference with TMVA SOFIE

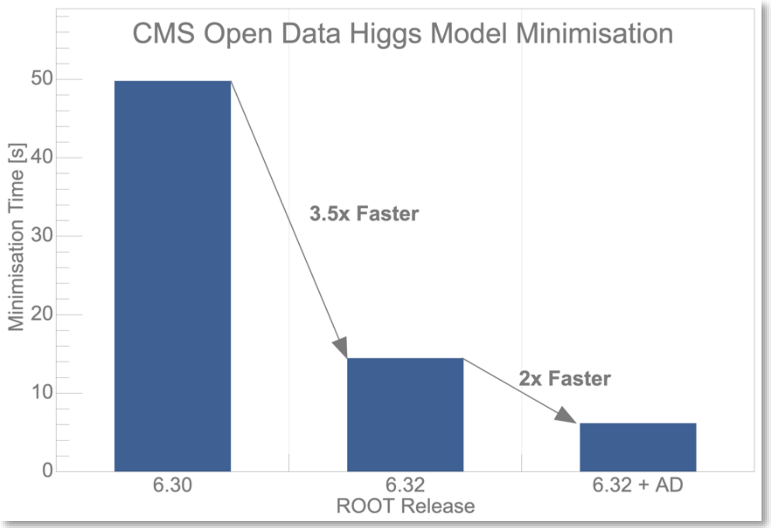
**ROOT enables efficient ML, well integrated with most used libraries**

# RooFit

- **Automatic Differentiation (AD) in RooFit with Clad**, in release 6.32.06
  - Improved numerical stability + reduced runtime
- Implementation **of the Analysis Grand Challenge statistical analysis in RooFit, following up on the existing RDataFrame implementation**
- RooFit runtime greatly reduced during the last few years, not only by AD:
  - Reorganisation of the calculations
  - Vectorisation and NVidia GPUs used in production
- **Shift focus on usability and Python interface**

**AD: R&D that became production grade, now available to the entire HEP Community**

☞ New RooFit Python interfaces for connections with ML



# Conclusions

# Conclusions

- The ROOT core team **is here to support you, listen to your needs and make your data processing and analysis a success!**
  - All sustained by a **long-term support model, and a rich result-oriented R&D program**
- **Open approach: open-source, open-development, open-planning**
  - For ROOT, collaborations and contributions are essential and highly valued!
- **ROOT is part of the Python ecosystem and will reinforce this position**
  - Prioritise the experience of Python users
- **ROOT 7.00.00 release: an evolutionary approach**
  - Prime importance given to the Python interface, stability and modularity
  - Once new interfaces (RDataFrame, RNTuple...) are adopted, the jump won't be large
  - Timeline: during LS3, on time for large MC productions

# ROOT Users Workshop: 17-21 Nov 2025

- **A welcoming, positive and inclusive atmosphere**
- **An opportunity to shape together the future of ROOT!**
- A venue for ROOT users, world-class experts of scientific computing and the ROOT core team to exchange ideas and learn from each other
- A rich program of presentations, tutorials, and most importantly, discussions
- Strong emphasis on the perspectives of early-career scientists and students

*The registration will be possible via the Indico page of the event, in the next few months*

**In Europe**
**17-21 November 2025**
**Save the dates!**