



WELCOME TO FERMILAB STUDY GROUP!

Our very first session!



Skill Sharing

Share your favorite coding skills and tools with your friends and colleagues in friendly, no-pressure work-alongs.



Co-Working

Get together to work on your coding projects, help each other out and share your work.



Community Building

Meet new people in your field, organization or community - and find out what we can do when we work together.

WHAT'S A STUDY GROUP?

.....

- Mozilla Study Groups are a fun and informal way to meet up with friends and colleagues to share coding skills and get help with problems in your research by working together.

- Our goals:

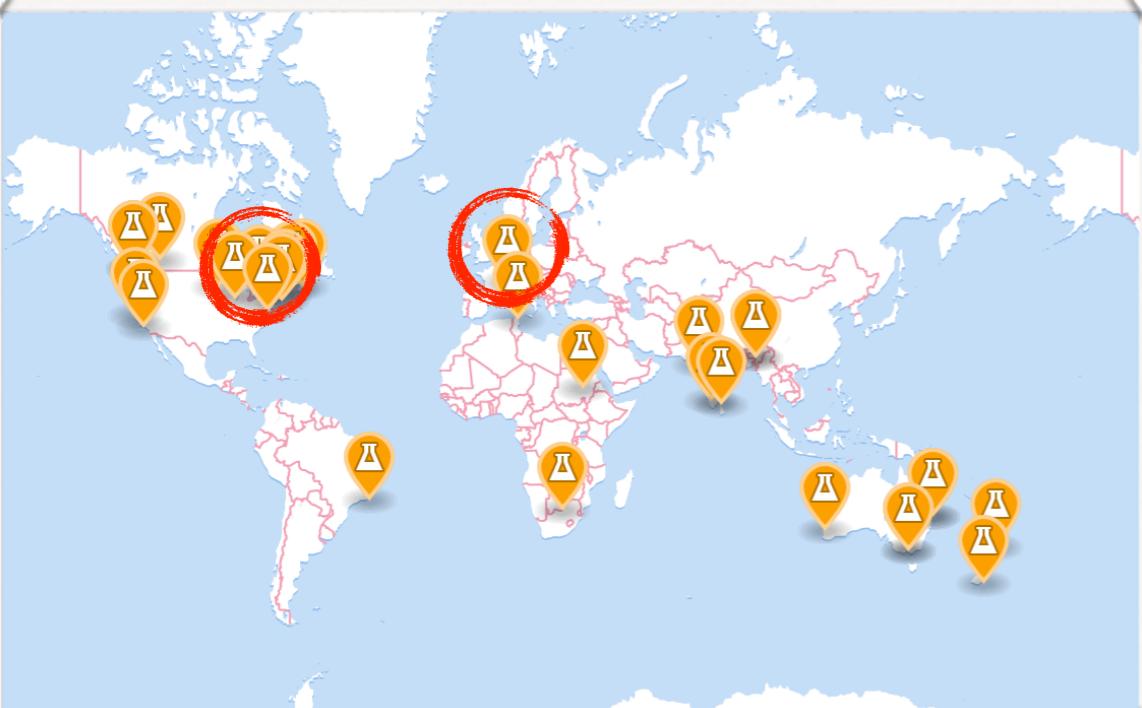
- Skill Sharing
- Idea Discovery
- Community Support



WHAT'S A STUDY GROUP?

.....

- We have more than 20 groups spread all over the world!
- CERN is a member, and now Fermilab as well!
- Everyone can join and be a collaborator.
- You discovered something really interesting that you would like to share with others? Let us know!
- You can suggest the topic for the next meeting, or even better, you can come for a demonstration and show us!





HOW TO JOIN

This repository | Search | Pull requests | Issues | Gist

FermilabStudyGroup / studyGroup forked from mozillascience/studyGroup

Code | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs

Gather together a group to skill-share, co-work, and create community <https://fermilabstudygroup.slack.com>

New Add topics

317 commits 8 branches

Branch: gh-pages | New pull request

This branch is 10 commits ahead of mozillascience:gh-pages.

sdporzio committed on GitHub Update members.yml

_data Update members.yml 5 minutes ago

_includes mobile layout fixed 8 months ago

_layouts Add a "Who We Are" page. 2 years ago

_plugins Revert "Fix expansion of tags in some data variables" a year ago

_posts Update 2017-03-28-firstEvent.markdown an hour ago

css/font-awesome first post 2 years ago

img experimenting with template 2 years ago

is first post 2 years ago

Notifications

Watch 0 Star 0 Fork 326

Not watching Be notified when participating or @mentioned.

Watching Be notified of all conversations.

Ignoring Never notify

Create new file Upload files Find file Clone or download

Pull request Compare

Latest commit b27fe07 5 minutes ago

- First thing to do:
Go on our repository on GitHub
(<https://github.com/FermilabStudyGroup/studyGroup>)
and click on “Watch”.
- This way you’ll be notified by email
about our next events!
- You can also join our Slack group, at:
<https://fermilabstudygroup.slack.com/signup>
- You’ll need to use a `@fnal.gov` address
to register. If you don’t want to use
it (or you don’t have one), you can
send me an email, asking for an
invitation, at:
salvatore.porzio@postgrad.manchester.ac.uk



slack

Product | Pricing | Support | Create a new team | Find your team | Sign in

Sign up for fermilabstudygroup.slack.com

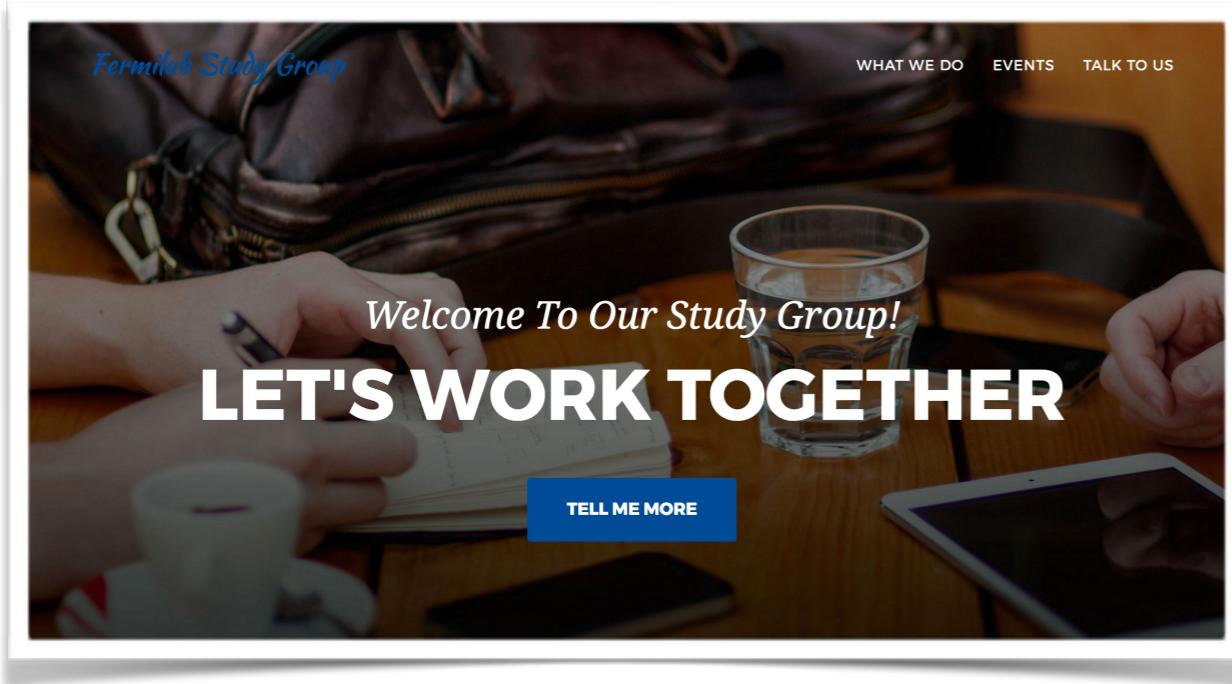
Enter your email address to get started.

@fnal.gov

Create Account

Don't have an @fnal.gov email address?
Contact your Team Administrator for an invitation.

WEBSITE AND INFORMATION



The image shows the "EVENTS" page of the Fermilab Study Group website. The header includes the "Fermilab Study Group" logo and navigation links for "WHAT WE DO", "EVENTS", and "TALK TO US". Below the header, the word "EVENTS" is prominently displayed in large white capital letters. A sub-header reads "Our upcoming and recent events". A welcome message states: "Welcome to Fermilab Study Group! Wilson Hall, Curia II [WH2SW], 28 March 2017, 12:00:00 PM" and "Our very first event. Bring lunch with you!". At the bottom are three blue buttons: "SUGGEST AN EVENT", "HOW TO LEAD A STUDY GROUP SESSION", and "SEE OUR PAST EVENTS".

- You can also visit our website here:

<https://fermilabstudygroup.github.io/studyGroup/>

- We will post periodically the date of the next meeting.

- You can suggest a date for an event as well!

- If you want to be a collaborator, just come forward! Anyone can contribute!



3 MINUTES BREAK TO SIGN-UP AND REGISTER

.....

Only if you want to, otherwise you can check Facebook in the meantime



<https://github.com/FermilabStudyGroup/studyGroup>



<https://fermilabstudygroup.slack.com/signup>

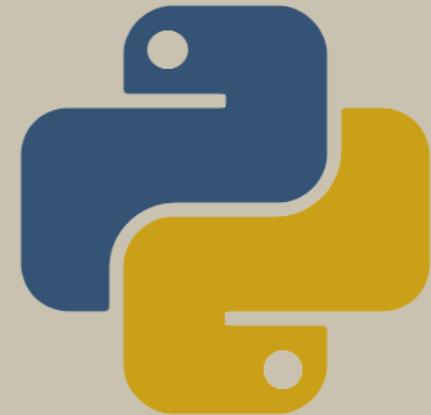
IMPORTANT

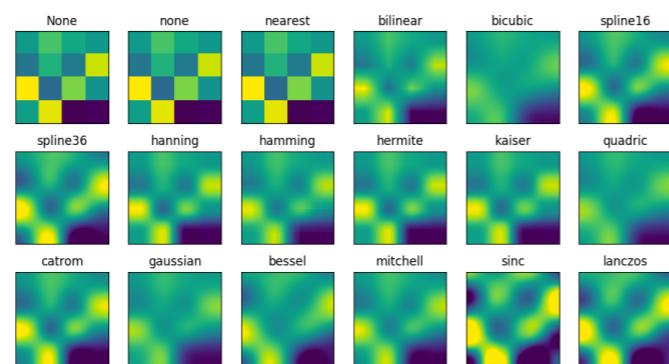
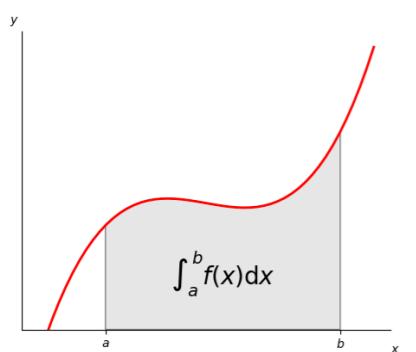
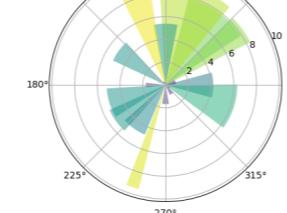
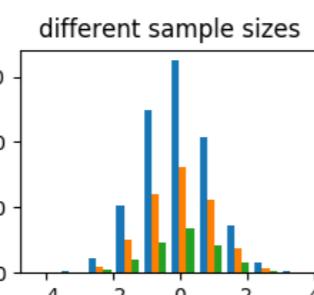
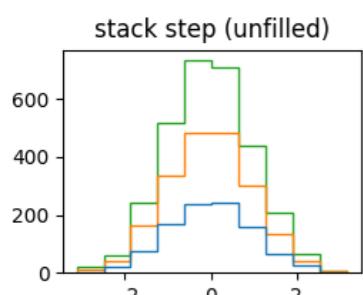
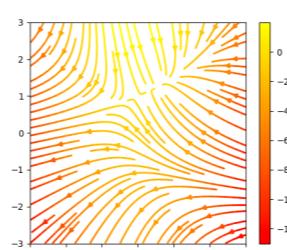
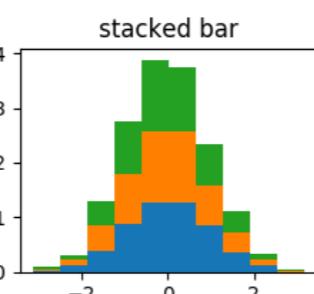
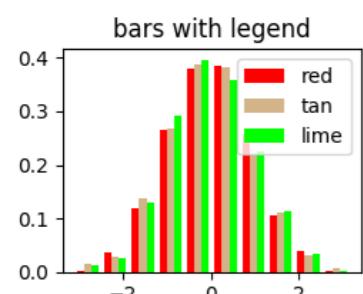
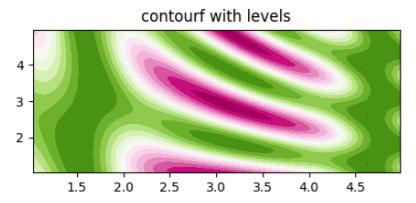
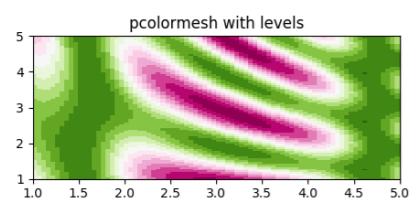
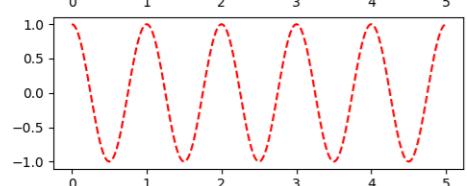
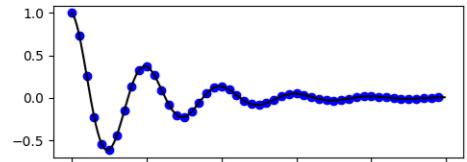
- I'm not an expert programmer, far from it.
- I started coding only 2 years ago and I'm still learning a lot.
- But on the way I'm finding a lot of cool things that make my life easier and programming more pleasant. I think that we could all benefit from sharing with each other our coding practices.
- Especially since many of you are probably physicists and, like me, you have received little training in coding. Especially compared to how much time of your day goes into coding.

- This goes for any of you. You don't have to be an expert to organize the next session. If you found something and think that other people could benefit from using it as well, come tell us about it!

PYTHON

A very short introduction*





WHY SHOULD I CARE ABOUT PYTHON

- Simple, short and extremely human-readable syntax.
 - Great strength of its core libraries (NumPy, SciPy, Pandas, Matplotlib).
 - High productivity for prototyping and building small and reusable systems.
 - True, everything that can be coded in Python can be done also using ROOT/C++.
 - But it would take me and (some of) you double the time to write that down.
 - Also, it just looks awesome!

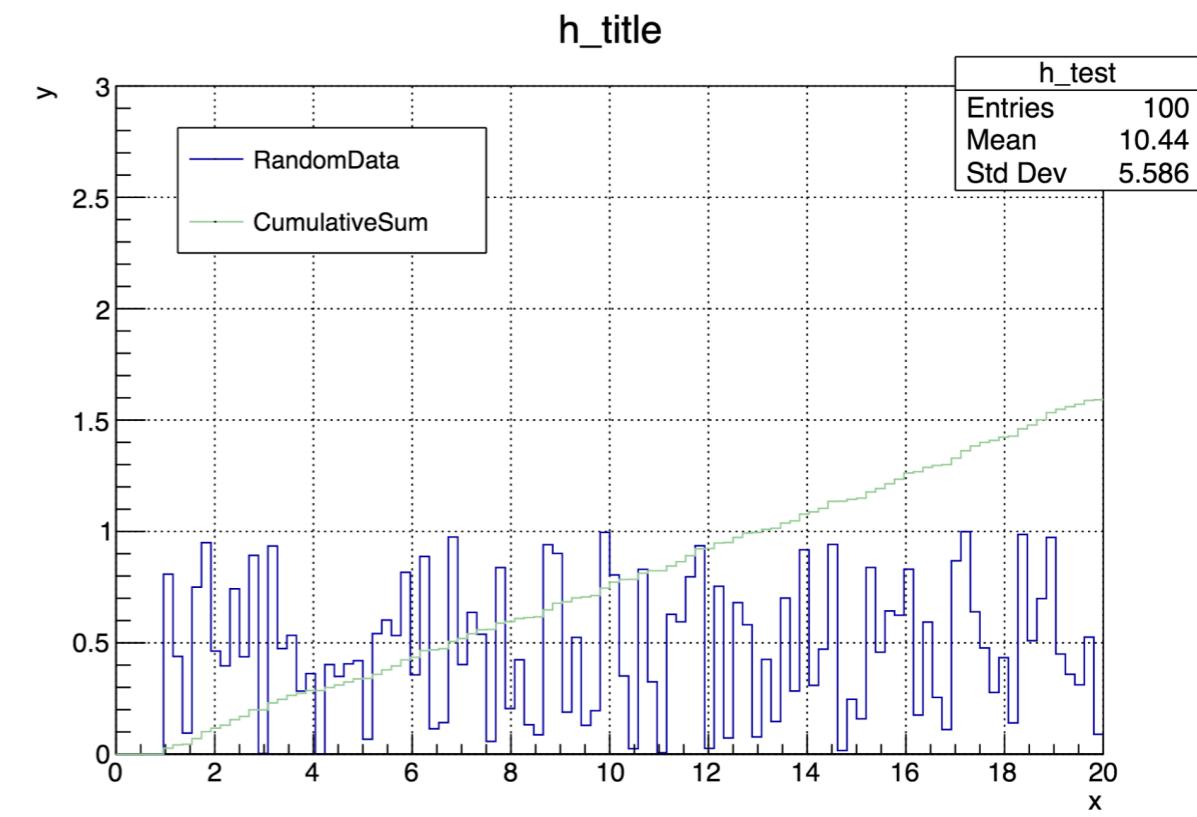
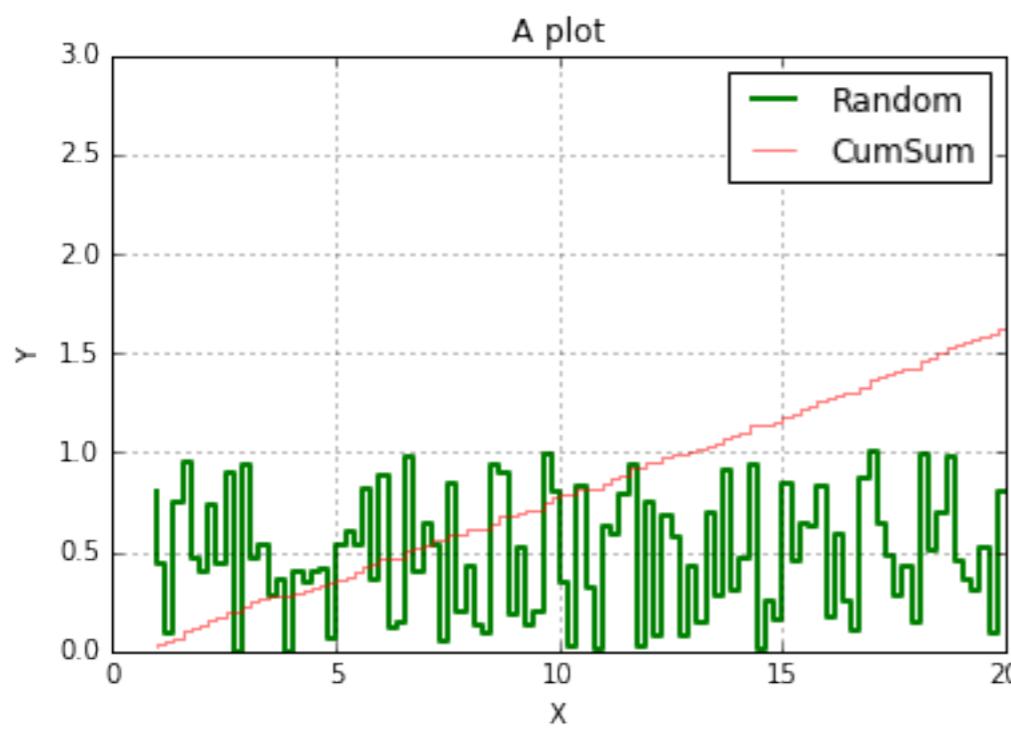
WHAT IF I DON'T KNOW PYTHON



- This lesson is inevitably going to be too fast for some people, and too slow or boring for others.
- We're aiming at showing you what (as far as I know) Python can do for you, and having it work on your laptop, not explaining what every single line of code does.
- If you want, we can have future lessons deal with that.
Or something more advanced.
Or something else entirely.
It's up to you!

PYTHON VS. ROOT

- Most of us use ROOT.
- Compare the two following plots:



PYTHON VS. ROOT

.....

- Compare the code now:

C++

PYTHON

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('data.csv')
plt.step(df['X'], df['Y'], color='green', lw=2, label='Random')
plt.step(df['X'], df['Y'].cumsum()/30., color='red', lw=0.5, label='CumSum')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('A plot')
plt.ylim(0,3)
plt.grid()
plt.legend()
plt.show()
```

- The C++ code, written by Colton and Adam has actually some nice points over the python script (like exception handling).

- If you're interested in it, you can find the code here:
<https://github.com/FermilabStudyGroup/snippets>

```
//c++ library includes
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <sstream>
#include <algorithm>

//ROOT includes - using Root v6.X
//ROOT includes
#include "TH1.h"
#include "TCanvas.h"
#include "TFile.h"
#include "TLegend.h"

//simple load doubles from text file
//function to read and return files
//note return type
std::vector< std::vector< double > > load_file(const char * file_name)
{
    std::vector< std::vector< double > > file_vector;
    std::ifstream myfile;
    myfile.open(file_name);
    //some exception handling for file io
    if(!myfile.is_open())
    {
        throw "File is not open!";
    }
    else
    {
        std::cout << "File opened: " << file_name << std::endl;
    }

    int events;
    int x;
    int y;
    int counter = 0;
    std::string line;
    std::vector<double> data_vector;

    //read the data out from the file
    //this readout assumes that values are ',' separated
    //also attempts to handle values which cannot be converted
    //from string to double
    while( !myfile.eof() )
    {
        std::string s;
        if (!getline( myfile, s )) break;
        std::istringstream ss( s );
        while ( ss )
        {
            std::string s;
            if (!getline( ss, s, ',' )) break;
            double data;
            try
            {
                data = std::stod(s);
            }
            catch(const std::exception& ex)
            {
                std::cerr << "No conversion to
                break;
            }
            //assumes that we have 3 values per line
            //file assumes 3 data products per line
            if(counter == 3)
            {
                file_vector.push_back(data_vector);
                data_vector.clear();
                counter = 0;
            }
            if(counter != 3)
            {
                data_vector.push_back(data);
                counter++;
            }
        }
        if(!data_vector.empty()) {file_vector.push_back(data_vector);
        data_vector.clear();}
    }
    myfile.close();
    return file_vector;
}

//histogram plotting function
void plot_histogram(const char * h_name,
                    const char * h_title,
                    const char * file_name,
                    const char * axis_title_x,
                    const char * axis_title_y,
                    std::vector< std::vector< double > > vector_name
)
{
    std::sort(vector_name.begin(), vector_name.end());

    const double max_bin = 20; //vector_name.back().at(1); //grab max x
    const double min_bin = 0; //vector_name.front().at(1); //grab min x
    TCanvas * c1 = new TCanvas();
    c1->SetGrid();
    TH1D * h1 = new TH1D(h_name, h_title, 104, min_bin, max_bin);
    TH1D * h2 = new TH1D(h_name, h_title, 104, min_bin, max_bin);
    for(int i = 0; i < vector_name.size(); i++)
    {
        h1->Fill(vector_name.at(i).at(1), vector_name.at(i).at(2));
    }

    // plot cumulative sum
    double cumulativeSum = 0;
    for(int iBin = 0; iBin < h1->GetNbinsX(); iBin++)
    {
        cumulativeSum = cumulativeSum + h1->GetBinContent(iBin+1);
        h2->SetBinContent(iBin+1, cumulativeSum/30.0);
    }

    // plot legend
    TLegend *leg = new TLegend(0.15, 0.7, 0.4, 0.85);
    leg->AddEntry(h1, "RandomData");
    leg->AddEntry(h2, "CumulativeSum");

    h1->SetMaximum(3.0);
    h1->GetXaxis()->SetTitle(axis_title_x);
    h1->GetYaxis()->SetTitle(axis_title_y);
    h2->SetLineColor(kGreen-8);

    h1->Draw("hist");
    h2->Draw("same");
    leg->Draw("same");
    c1->Print(file_name);
}

//main - call functions here
//try open file -> plot histogram
//try running functions here
int main()
{
    //file name
    const char * in_file = "plot_data.txt";
    std::vector< std::vector< double > > plot_data;
    //if problem opening and loading file, exit
    try
    {
        plot_data = load_file(in_file);
    }
    catch(const char * msg)
    {
        std::cerr << msg << std::endl;
        exit(1);
    }

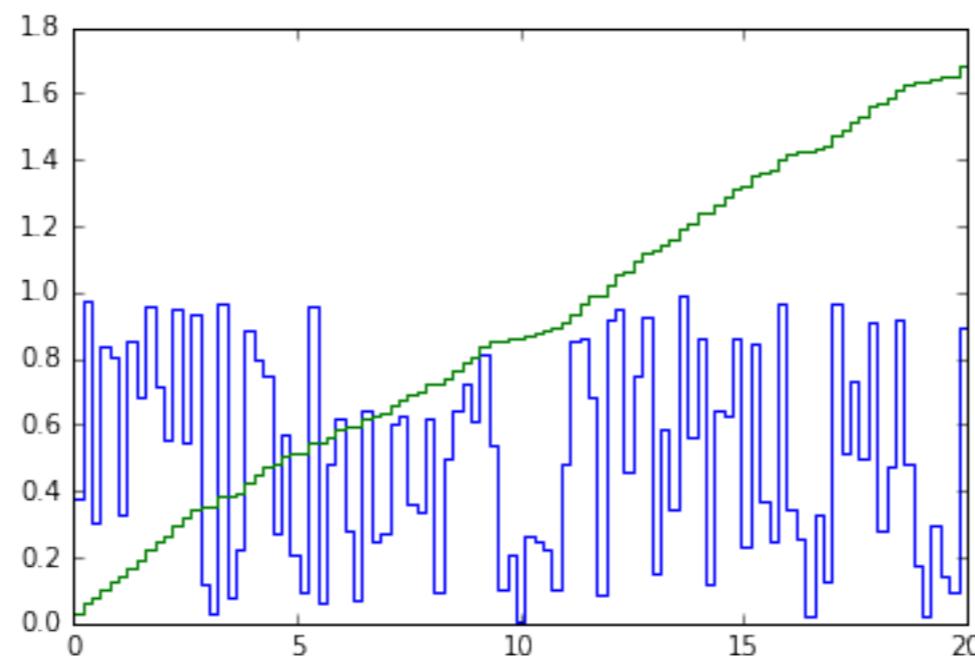
    //see function for variables
    plot_histogram("h_test", "h_title", "h_title.pdf", "x", "y",
    plot_data);
    return 0;
}
```

PYTHON VS. ROOT

- And if we are in a hurry and don't want to be fancy:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('data.csv')
plt.step(df['X'], df['Y'])
plt.step(df['X'], df['Y'].cumsum() / 30.)
```



PYTHON VS. ROOT

- Let's get to the lesson

```
git clone https://github.com/FermilabStudyGroup/lessons.git
```

- We are going to install a nice pythonic environment with all the modules we need to use via Anaconda.
- What's Anaconda? The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular Python, R and Scala packages for data science.
- There are other ways to install Jupyter Notebooks and PyROOT (e.g. via pip or homebrew). We can discuss these in other lessons. For now we will go with Anaconda because it's completely self-contained.
- We will need to follow the instructions in the readme.md in:
<https://github.com/FermilabStudyGroup/lessons/tree/master/Lesson1>



10 MINUTES BREAK TO MAKE IT WORK

.....
Maybe more

SOME NICE THINGS FROM PYTHON AND JUPYTER NOTEBOOK

- If the installation worked, now all you need to do is:

```
jupyter notebook <filename.ipynb>
```

- But how to use it? Let's look at:

```
Lessons/Lesson1/0_showcase.ipynb
```

- And later let's read a ROOT TTree with:

```
Lessons/Lesson1/1_readTTree.ipynb
```



TIME FOR EXERCISES

.....
Don't hesitate to ask for help!