THIRD YEAR LABORATORY

Analysing Events with Z, W and Higgs Bosons with the ATLAS Experiment at the LHC

Dept. of Physics & Astronomy, University of Manchester. Version 2.91 September 2020.

1 Aims

This project aims to give you some appreciation of the physics processes that can occur in high energy proton-proton collisions at the LHC and to introduce you to event selection and measurement methods used in particle physics data analysis.

2 Objectives

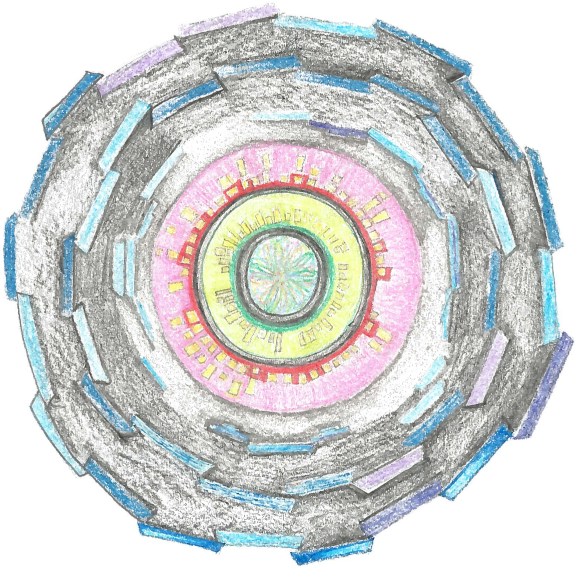
1. Design criteria to select the different sorts of particles and the different types of events produced at the LHC and eliminate unwanted "backgrounds". The focus is on events containing the decays of Z, W and Higgs bosons to electrons and muons.

2. Compare real data collected by the ATLAS experiment with computer-simulated "Monte Carlo" data.

3. Measure the cross sections for the production of Z, W and Higgs bosons at the LHC and estimate statistical and systematic uncertainties on the results.

4. Gain experience of using `ROOT` — a framework designed at CERN to analyse large data sets.

5. (Optional) Extend the event selections to other sources of events at the LHC, such as Higgs decaying to two photons, or events containing tau leptons, pairs of vector bosons, top quarks, or potential sources of new physics from "beyond the standard model".



The first version of this analysis experiment and the associated documentation was produced by summer student Nathan Simpson and Prof. Terry Wyatt. It was updated and extended to use the ATLAS 13 TeV data by summer student Lewis Powell and Terry Wyatt. It was updated for remote use, with Jupyter notebooks, by summer student Elliot Watton and Terry Wyatt.

	I	II	III
charge $= -1$	electron (e)	muon (μ)	tau (τ)
charge $= 0$	electron neutrino (ν_e)	muon neutrino (ν_μ)	tau neutrino (ν_τ)

Table 1: The three generations of leptons.

3 Introduction

This document is designed to be used electronically. References to websites, other documents, sections of this document, etc., are appropriately hyperlinked. Hyperlinks are indicated by text that is underlined and/or enclosed within coloured boxes. For example, here is a link to the ROOT users' guide. For ease of reference all the hyperlinks distributed throughout the document are also collected together in subsection 7.1.

4 Background reading

4.1 A femto-course in particle physics

Since many students (especially those doing this lab. in the 1st semester) have not had a formal introduction to particle physics at this point in the academic year, a brief introduction to particle physics is given here.

The most well-tested and robust theory of particle physics that we have today is called the **standard model**. In this model, all of matter is made up of fundamental constituent particles that come in two categories: quarks and leptons. We sort these particles into **three generations of matter** (I, II, III). Each successive generation contains two quarks and two leptons, making up twelve particles in total. If we then factor in the existence of **antiparticles** — particles with opposite charge to their counterparts, denoted with a bar (e.g. the up antiquark, $\bar{u}$) — we have twelve additional particles (six antiquarks and antileptons, three of which are charged and three of which are neutral antineutrinos).

4.1.1 Leptons

Leptons are spin-$\frac{1}{2}$ fundamental particles. The term spin refers to the particles' intrinsic angular momentum, which is given in terms of $\hbar$. They can have charges of -1, $+1$ (for antiparticles), and 0 in units of the electron charge $e \approx 1.6 \times 10^{-19}$ C. Each generation of leptons has a flavour — electron, muon, or tau — and consist of a charged lepton and a neutrino. These leptons are shown in Table 1. "Flavour" can simply be thought of as a way of categorizing particles into species.

4.1.2 Quarks

Quarks are also spin-$\frac{1}{2}$ fundamental particles, but differ from leptons in a number of aspects, including charge, mass, etc. There are two categories of quarks: **up-type** quarks with charge

	I	II	III
charge $= \frac{2}{3}$; up-type	up (u)	charm (c)	top (t)
charge $= -\frac{1}{3}$; down-type	down (d)	strange (s)	bottom (b)

Table 2: The three generations of quarks.

Particle	Force mediated	Mass (GeV)	Charge (e)
Z	Electroweak	91.1876 ± 0.0021	0
W^+/W^-	Electroweak	80.385 ± 0.015	$+1/-1$
Photon (γ)	Electroweak	0	0
Gluon (g)	Strong	0	0
Graviton (hypothetical)	Gravity	0	0

Table 3: The force-mediating bosons. The hypothetical graviton is included for completeness, although we have so far found no experimental evidence of its existence.

$\frac{2}{3}$ and **down type** quarks with charge $-\frac{1}{3}$. As far as we know, there are twelve quarks in total — three up-type, three down-type, and their corresponding antiparticles. A summary of the known quarks can be found in Table 2. Quarks can bind together under the influence of the strong interaction to form different types of composite particles called **hadrons**. These can be **baryons** (containing three quarks, or three antiquarks for antibaryons) or **mesons** (containing a quark and an antiquark).

4.1.3 Gauge Bosons

In addition to quarks and leptons, we have **guage bosons**, a category of particles that are responsible for mediating the forces of nature. These are listed in Table 3.

4.1.4 Higgs Boson

Finally, there is one further boson which does not directly mediate a force. This is the **Higgs boson**, H. It is an excitation of the Higgs field, the field which permeates space and gives rise to the masses of the W and Z bosons. It has a mass of 125.10 ± 0.14 GeV, and has zero electric charge.

4.2 Interactions & Feynman diagrams

The main interactions of interest for this experiment are the production and decay of Z, W and Higgs bosons. The most easily identified decay mode of the Z boson is to a pair of charged leptons. We write this as $Z \rightarrow l^+l^-$, where l denotes a particular lepton flavour ($l = e, \mu, \tau$). An analogous decay mode of the W^- boson is to a lepton and an antineutrino. We write this as $W^- \rightarrow l^-\bar{\nu}_l$. We can represent such processes with **Feynman diagrams**, as shown in Figure 1. Note that **electric charge is conserved at each vertex**. 4-momentum is also conserved at each vertex in a Feynman diagram. To learn more about Feynman diagrams see, e.g., this resource.

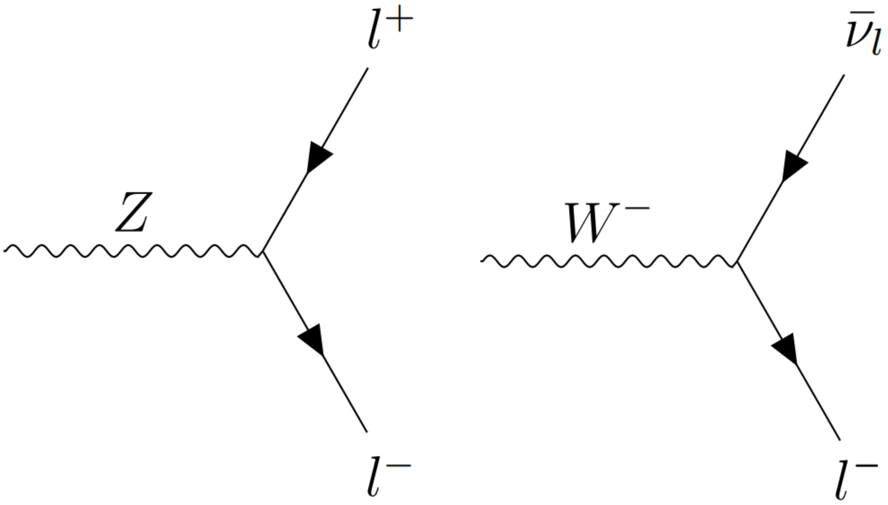


Figure 1: The Feynman diagrams for the decay processes $Z \to l^+l^-$. (left) and $W^- \to l^-\bar{\nu}_l$ (right).

The easiest Higgs decay mode to search for is the decay to a pair of Z bosons both of which decay to a pair of charged leptons. This is denoted by $H \to ZZ^{(*)} \to \ell\ell\ell\ell$. The asterisk means that one of the Z bosons is **virtual** since the Higgs boson's mass is not great enough for it to decay into two real Z bosons. The corresponding Feynman diagram is shown in Figure 2.

Although the beams of particles accelerated in the LHC are protons, the collisions we are most interested in here occur between individual quarks or gluons within the colliding protons. We use the collective term **parton** to describe a constituent of a hadron (be it a quark, an antiquark, or a gluon).

4.3 The experiment

Information about the ATLAS collaboration, the detector, the physics etc. can be found here. It is worth knowing your way around the detector and the experiment, as it will add some valuable context to the work you do.

A journal paper describing in quite some detail the measurements of Z and W production in the very early ATLAS data at the LHC can be found here. You might find this useful background reading outside the lab. Hopefully you can manoeuvre your way around some of the the technical jargon and get an idea of the kind of analysis that can be performed on the data.

4.4 About ATLAS OpenData

The ATLAS Open Data is a project designed by ATLAS to enable the "general public", including students, to explore data collected at the LHC the techniques used in modern experimental particle physics. ATLAS has produced several (largely unmodified) data sets from the actual 13 TeV ATLAS run for use in educational analysis.

For details on the data sets you are using, you can consult the ATLAS OpenData online documentation for reference. In addition, links to specific relevant pages will be given throughout this document

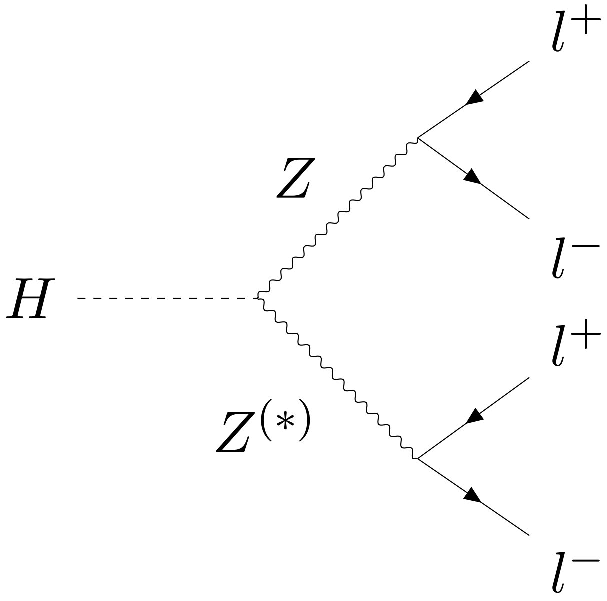


Figure 2: The Feynman diagram for the decay process $H \to ZZ^{(*)} \to \ell\ell\ell\ell$

where appropriate.

5 Getting started

To conduct this experiment remotely, you will need to use your personal computer or laptop to access one of the particle physics Linux lab. machines. You will start a Jupyter notebook server running on the lab. machine. You will then be able to connect to this server using a web browser running on your personal computer. Using the browser interface you can access all the files and scripts you will need to use to complete this experiment. No specialist software or data needs to be installed on your personal computer.

5.1 Logging into the lab. machine

In order to log into the lab. machine you have been assigned, you need to carry out the following steps:

1. Firstly, you must open up a terminal window on your personal computer. For Windows users this will be **Windows PowerShell**. For Unix/Linux users this will be a standard Unix/Linux shell.

2. To access the lab. machine, the following command must be typed:

   ssh atlaslabμ@atlaslabμ.blackett.manchester.ac.uk

   Here, you replace μ with the number of the lab. machine you are working on. You will be told this by your demonstrator.

3. You may see some text like:

```
 The authenticity of host 'atlaslab16.blackett.manchester.ac.uk
(195.194.107.11)' can't be established.
ECDSA key fingerprint is SHA256:2BOYSXfZx1Zqic6S8o2P8k1QZEmDQHNuBBZvY+RZZ2M.
ECDSA key fingerprint is MD5:f0:ea:1f:68:92:0e:5b:04:7b:dc:5b:be:0e:50:32:9b.
Are you sure you want to continue connecting (yes/no)?
```

If so, just answer **yes**. If you do not see any such message that is fine too.

4. Next, you will be prompted to enter the password for the specific machine. You will be told this by your demonstrator.

5. Once you have typed the correct password, you will be connected to the lab. machine and will be able to access its files and programs.

5.2 Setting up the code — *To do once at the start of the experiment:*

A skeleton version of the code you will use to analyse the ATLAS Open Data in this experiment is provided in two alternative forms: python and C++. In both cases, the analysis code is run and the output histograms are viewed using python scripts.

Before going further, you should choose as a pair whether you wish to write code in python or C++. Either you or your lab. partner needs to log into the lab. machine, as described above in section 5.1, and carry out the followowing steps:

- If you wish to work in python, type:

cp -r /opt/ATLAS-Project-py ATLAS-Project	This copies the starting material for python code to your local directory.

- Alternatively, if you wish to work in C++ , type:

cp -r /opt/ATLAS-Project-C ATLAS-Project	This copies the starting material for C++ code to your local directory.

- Now type in the following lines:

cd ATLAS-Project	Changes your "current directory" to `ATLAS-Project`.
mkdir outputPlots	Makes a sub-directory called *OutputPlots*, which is where you will save pictures of selected histograms for archiving and printing.
ls -l	Displays the files/folders in your current directory and their permissions.

In your directory *ATLAS-Project*, you should now be able to see the following files and subdirectories:

```
ATLAS-Project
└─Analysis.py (python) or
```

`Analysis.C and Analysis.h (C++)`	the analysis function you will be editing
`RunAnalysis.py`	python script to run the analysis function
`TotExpected.py`	script to get expected numbers of simulated events
`LabNotebook.ipynb`	Jupyter notebook used to run the code and view the results
`atlas-experiment-lab-script.pdf`	this lab. script
`backend`	a sub-directory with “service” code to make everything work smoothly
`input.txt`	a small text file you can use to specify a list of several input files
`out`	a sub-directory where RunAnalysis.py will save histogram files
`outputPlots`	a sub-directory where you will save pictures of histograms

Exercise 5.1. *Try to get yourself set up using the above instructions. Talk to your demonstrator at any point if something is not clear. Remember that you need to do this only once, at the beginning of your first lab. day!*

Once this has all been set up, you can safely log out of the lab. machine by entering the following command:

exit	This will log out this terminal from the lab. machine.

Tip 1. *The above commands are pretty much the only bits of Linux you will absolutely need in order to complete this lab. successfully.*

A few simple Linux tricks can help minimise the amount of repetitive typing you have to do:

*You can use the “**tab**” key on the keyboard to “auto-complete” the names of commands, files, and directories. For example, if you type ‘**ls atl**’ and then press “**tab**”, it should complete the filename for you and say ‘**ls atlas-experiment-lab-script.pdf**’.*

*You can use the “**up arrow**” key to recall commands that you have typed previously. N.B. This works both when you are in the Linux “shell” and also when you are typing on the command line within* ROOT*.*

*If you would like to find out more about a particular Linux command you can use the option “**--help**” after the command name. For example if you type “**mv --help**” it will tell you about the Linux command used to “move” files or directories from one location to another.*

Tip 2. *At the end of the day in the lab. we suggest you log off. Ensuring a fresh start by logging on at the beginning of each new lab. day is recommended.*

5.3 Starting up the Jupyter notebook server

These steps will need to be followed at the beginning of every lab. day by **either** you or your lab. partner. This is because you and your lab. partner will be using the same Jupyter notebook server.

1. Log into the lab. machine using the steps in section 5.1.

2. The next step is to run a Jupyter notebook server on the lab. machine. This is done through using the **Singularity** software installed on the lab. machine. To correctly configure Singularity the next two command lines must be entered:

   export SINGULARITY_TMPDIR=$HOME/.singularity/tmp
   singularity run -B /data/ATLAS:$HOME/ATLAS-Project/ATLAS $HOME/root_6.22.00-conda.sif bash

   The following text should appear straightaway:

   ```
   INFO: Convert SIF file to sandbox...
   ```

   It may then take around a minute for the prompt

   ```
   Singularity>
   ```

   to appear in the terminal window. Please be patient and don't hit any keys in the terminal window, while you are waiting ;-)

3. To then start up the Jupyter notebook server, type the following command:

   jupyter-notebook --no-browser --port=8888

   Here, we make sure that the Jupyter notebook browser does not open on the lab. machine (as you want to access it on your personal computer). Instead, we relate the Jupyter notebook server to a port with number by 8888. The port number can actually be any 4-digit number greater than 8000, but unless there is a good reason to change then for simplicity it is probably simplest to stick to 8888. If you do choose a different port number then "8888" will need to be changed in all the commands given below.

4. After typing this, you should notice that a large amount of text appears in your window. This represents the Jupyter notebook server starting up. It should look something like:

   ```
   Writing notebook server cookie secret to
   /home/atlaslab16/.local/share/jupyter/runtime/notebook_cookie_secret
   Loading IPython parallel extension
   Serving notebooks from local directory:  /home/atlaslab16
   The Jupyter Notebook is running at:
   http://localhost:8888/?token=a7ea520d2f4d6f62ea2026ac0aa1edd0a48e935b03539bc4
   or
   http://127.0.0.1:8888/?token=a7ea520d2f4d6f62ea2026ac0aa1edd0a48e935b03539bc4
   Use Control-C to stop this server and shut down all kernels (twice to skip
   confirmation).
   .
   To access the notebook, open this file in a browser:
   file:///home/atlaslab16/.local/share/jupyter/runtime/nbserver-15700-open.html
   Or copy and paste one of these URLs:
   http://localhost:8888/?token=a7ea520d2f4d6f62ea2026ac0aa1edd0a48e935b03539bc4
   or
   http://127.0.0.1:8888/?token=a7ea520d2f4d6f62ea2026ac0aa1edd0a48e935b03539bc4
   ```

   You need to keep this terminal window open on your personal computer during the entire period you will be working.

   N.B. If you notice an error message such as

   ```
   The port 8888 is already in use, trying another port.
   ```

   then replace "8888" with some other number greater than 8000.

5. You now need to open another terminal window on your personal computer. In this new terminal, you type:

   ssh -N -L localhost:1234:localhost:8888 atlaslabμ@atlaslabμ.blackett.manchester.ac.uk

   Here, 8888 will be the same port number chosen before. 1234 is the port through which the Jupyter notebook server will be accessed from a browser running on your personal computer. This can be any 4 digit number, but it is recommended to use something like 1234 for simplicity. Again, μ is the number of the lab. machine and matching account you are using.

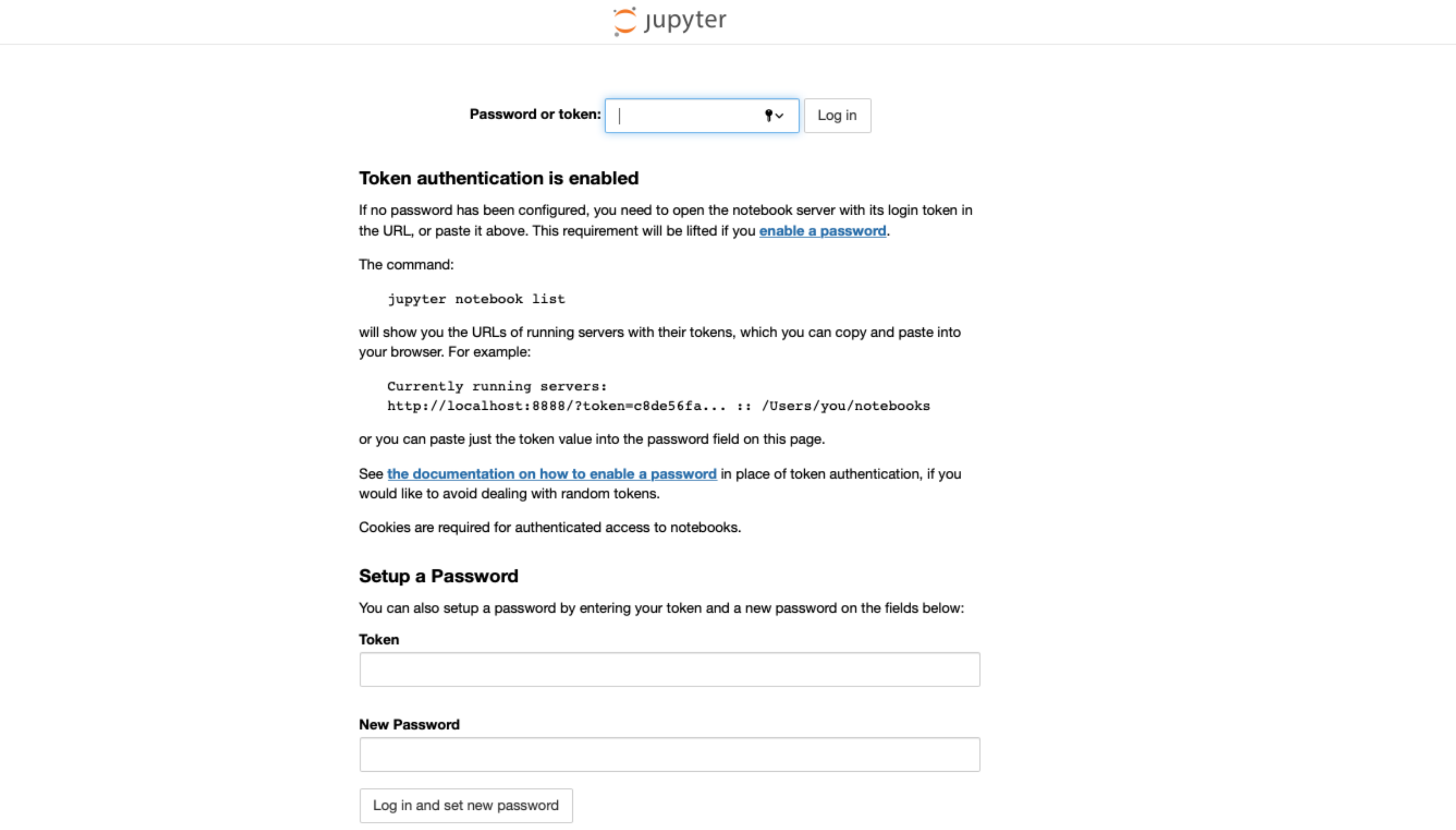
6. You will once again have to enter the password specific to your lab. machine. Once this has done there should be no further messages sent to that current terminal. You need to keep this also this second terminal window open on your personal computer during the entire period you will be working.

7. Now your personal computer can access the Jupyter notebook server that is being hosted by the lab. machine. To open up Jupyter notebook on your personal computer, enter the following into the URL address bar of any browser of your choice:

   localhost:1234

   Here, 1234 is the same port as specified above in the command **ssh -N -L . . . .**

8. If all these steps have been followed, you will be presented with a page that looks like the following.



Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

Token

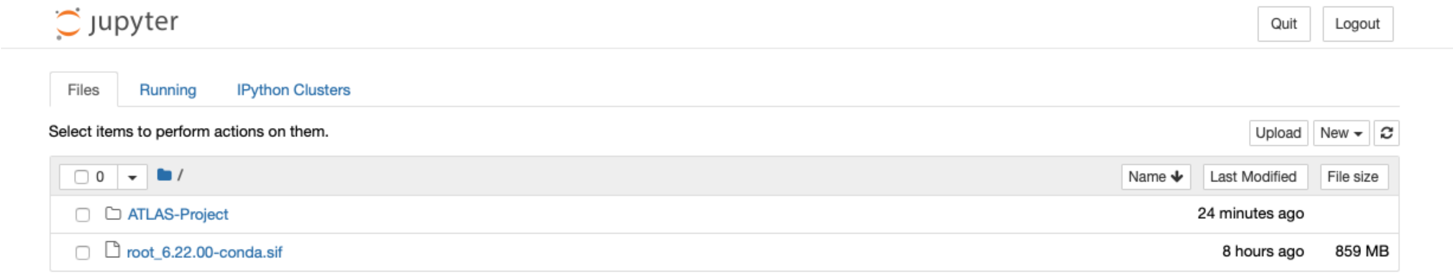
New Password

you must enter a **token** into the box labelled **Password or token** at the top of the page. This token can be found in the first terminal window you used. It is written in the URLs that are written to the terminal window after step 4 was completed. For example, if one of the URLs is

http://localhost:8888/?token=Z

Then the token is given by **Z** (normally this is a long string of random numbers and letters). Here, 8888 is again the port number chosen on the lab. machine in step 3 above.

9. After entering the token, you should be directed to the Jupyter notebook server's homepage where you can see the directories of the lab. machine.

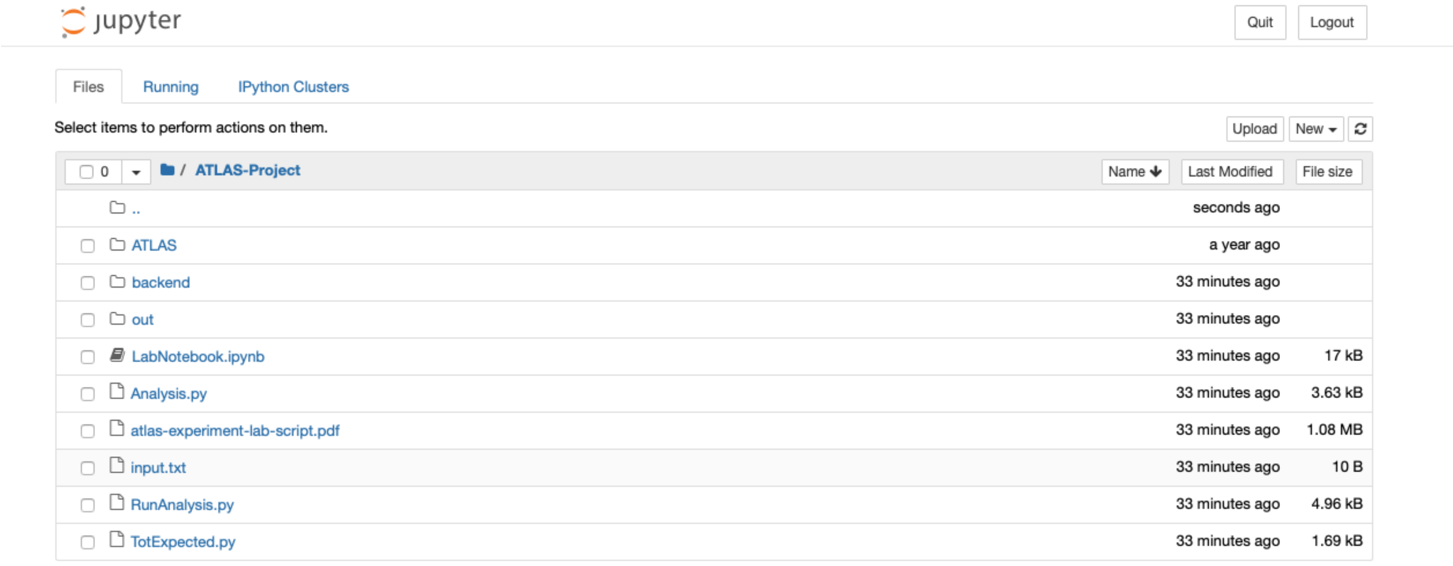


Select items to perform actions on them.

Name	Last Modified	File size
ATLAS-Project	24 minutes ago	
root_6.22.00-conda.sif	8 hours ago	859 MB

If this is the case then you have correctly set up your Jupyter notebook server browser window, and are ready to start some analysis. If you do not arrive at the homepage, then something has gone wrong; perhaps the token or one of the port numbers was copied incorrectly?

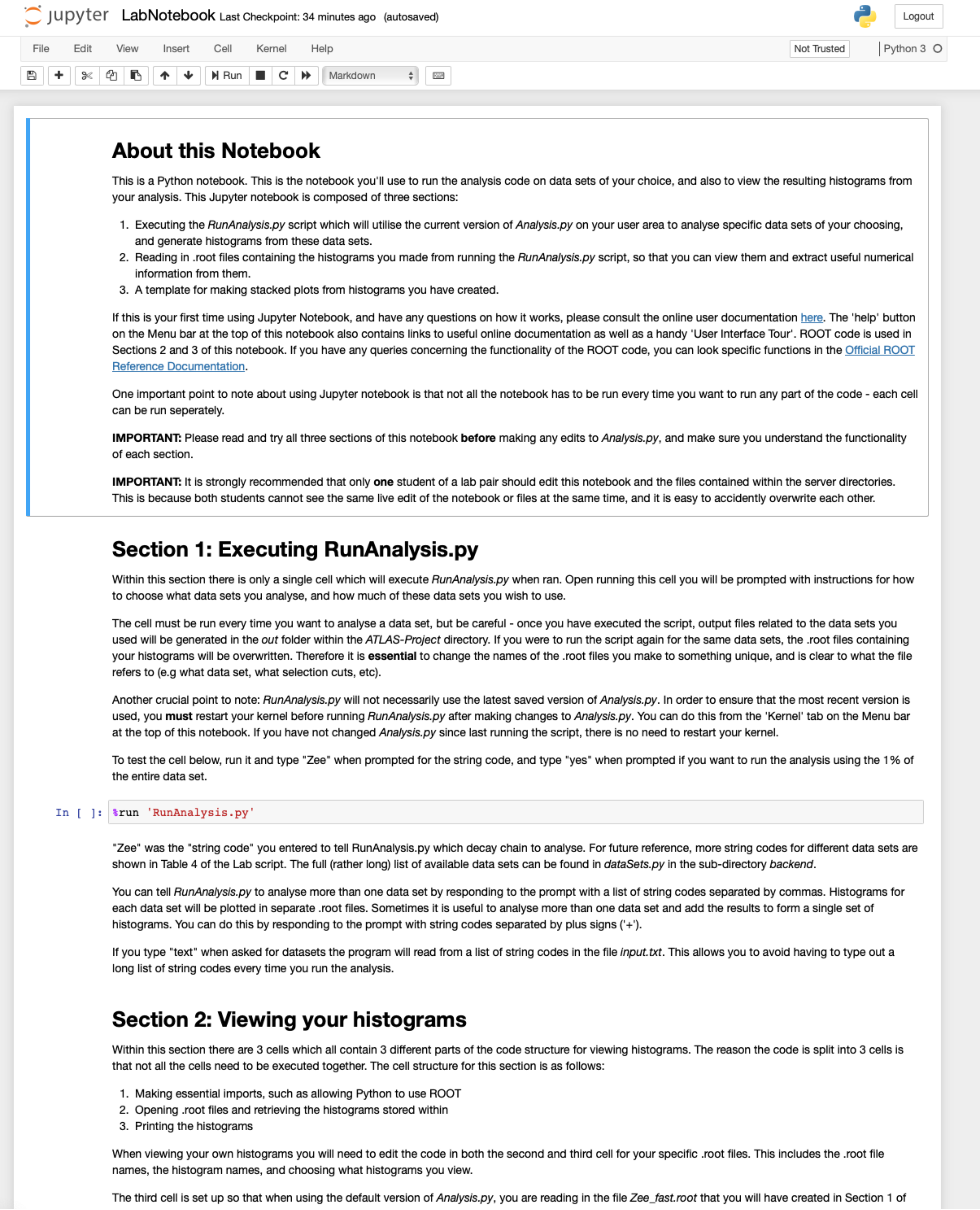
10. Click on the directory name `ATLAS-Project` in the browser window. This will display the contents of this directory



Select items to perform actions on them.

Name	Last Modified	File size
..	seconds ago	
ATLAS	a year ago	
backend	33 minutes ago	
out	33 minutes ago	
LabNotebook.ipynb	33 minutes ago	17 kB
Analysis.py	33 minutes ago	3.63 kB
atlas-experiment-lab-script.pdf	33 minutes ago	1.08 MB
input.txt	33 minutes ago	10 B
RunAnalysis.py	33 minutes ago	4.96 kB
TotExpected.py	33 minutes ago	1.69 kB

11. Click on the Jupyter notebook script `LabNotebook.ipynb` in the browser window. This will run the notebook and display


jupyter LabNotebook Last Checkpoint: 34 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Not Trusted
Python 3 ○
Run
Markdown

About this Notebook

This is a Python notebook. This is the notebook you'll use to run the analysis code on data sets of your choice, and also to view the resulting histograms from your analysis. This Jupyter notebook is composed of three sections:

1. Executing the *RunAnalysis.py* script which will utilise the current version of *Analysis.py* on your user area to analyse specific data sets of your choosing, and generate histograms from these data sets.
2. Reading in .root files containing the histograms you made from running the *RunAnalysis.py* script, so that you can view them and extract useful numerical information from them.
3. A template for making stacked plots from histograms you have created.

If this is your first time using Jupyter Notebook, and have any questions on how it works, please consult the online user documentation here. The 'help' button on the Menu bar at the top of this notebook also contains links to useful online documentation as well as a handy 'User Interface Tour'. ROOT code is used in Sections 2 and 3 of this notebook. If you have any queries concerning the functionality of the ROOT code, you can look specific functions in the Official ROOT Reference Documentation.

One important point to note about using Jupyter notebook is that not all the notebook has to be run every time you want to run any part of the code - each cell can be run seperately.

IMPORTANT: Please read and try all three sections of this notebook **before** making any edits to *Analysis.py*, and make sure you understand the functionality of each section.

IMPORTANT: It is strongly recommended that only **one** student of a lab pair should edit this notebook and the files contained within the server directories. This is because both students cannot see the same live edit of the notebook or files at the same time, and it is easy to accidently overwrite each other.

Section 1: Executing RunAnalysis.py

Within this section there is only a single cell which will execute *RunAnalysis.py* when ran. Open running this cell you will be prompted with instructions for how to choose what data sets you analyse, and how much of these data sets you wish to use.

The cell must be run every time you want to analyse a data set, but be careful - once you have executed the script, output files related to the data sets you used will be generated in the *out* folder within the *ATLAS-Project* directory. If you were to run the script again for the same data sets, the .root files containing your histograms will be overwritten. Therefore it is **essential** to change the names of the .root files you make to something unique, and is clear to what the file refers to (e.g what data set, what selection cuts, etc).

Another crucial point to note: *RunAnalysis.py* will not necessarily use the latest saved version of *Analysis.py*. In order to ensure that the most recent version is used, you **must** restart your kernel before running *RunAnalysis.py* after making changes to *Analysis.py*. You can do this from the 'Kernel' tab on the Menu bar at the top of this notebook. If you have not changed *Analysis.py* since last running the script, there is no need to restart your kernel.

To test the cell below, run it and type "Zee" when prompted for the string code, and type "yes" when prompted if you want to run the analysis using the 1% of the entire data set.

```
In [ ]: %run 'RunAnalysis.py'
```

"Zee" was the "string code" you entered to tell RunAnalysis.py which decay chain to analyse. For future reference, more string codes for different data sets are shown in Table 4 of the Lab script. The full (rather long) list of available data sets can be found in *dataSets.py* in the sub-directory *backend*.

You can tell *RunAnalysis.py* to analyse more than one data set by responding to the prompt with a list of string codes separated by commas. Histograms for each data set will be plotted in separate .root files. Sometimes it is useful to analyse more than one data set and add the results to form a single set of histograms. You can do this by responding to the prompt with string codes separated by plus signs ('+').

If you type "text" when asked for datasets the program will read from a list of string codes in the file *input.txt*. This allows you to avoid having to type out a long list of string codes every time you run the analysis.

Section 2: Viewing your histograms

Within this section there are 3 cells which all contain 3 different parts of the code structure for viewing histograms. The reason the code is split into 3 cells is that not all the cells need to be executed together. The cell structure for this section is as follows:

1. Making essential imports, such as allowing Python to use ROOT
2. Opening .root files and retrieving the histograms stored within
3. Printing the histograms

When viewing your own histograms you will need to edit the code in both the second and third cell for your specific .root files. This includes the .root file names, the histogram names, and choosing what histograms you view.

The third cell is set up so that when using the default version of *Analysis.py*, you are reading in the file *Zee_fast.root* that you will have created in Section 1 of

12. You can now follow the instructions displayed in the browser window to run the analysis

and examine the produced histograms.

If you are the member of your lab. pair that is not setting up the server, then you can access the same Jupyter notebook server by completing steps 5–9 using a single terminal window on your machine. In step 5, 8888 and μ should be the same as those used by your lab. partner. The 1234 can also be the same, although it does not have to be. The token used in step 8 will be the same token that your lab. partner used.

A reminder: all the terminal windows that you have used to set up and/or log on to the Jupyter notebook server should not be closed during the lab. day, unless you wish to log off.

5.4 Closing the Jupyter notebook server and logging out of the lab. machine

In order to shut down the Jupyter notebook server on the lab. machine and log off you must do the following:

- Close the terminal window where step 5 of section 5.3 was completed (or hit **ctrl+C** with the focus on that window).
- Within the terminal window where steps 1–4 of section 5.3 were completed you should hit **ctrl+C**; this means you should press the 'C' key on your keyboard while holding down the 'ctrl' (control) key.
- Now enter the following commands:

y	This will close down the Jupyter notebook server.
exit	This will make your terminal close Singularity.
exit	This will log this terminal out of the lab. machine.

If you are not the member of your lab. pair that started up the server (or you have other terminal windows longged into the lab. machine), you will need only to hit **ctrl+C** in the terminal window you are using to connect to the Jupyter notebook server on the lab. machine.

5.5 Using Analysis.py or Analysis.C

The only files you should need to edit are *Analysis.py* for python users, or *Analysis.C* and *Analysis.h* for C++ users. You can open and edit these files by clicking on them from within the *ATLAS-Project* directory using the Jupyter notebook server browser window. You will perform calculations and plot histograms in *Analysis.py* or *Analysis.C*. If you are using C++ you will also need to update *Analysis.h* whenever you wish to declare a new histogram.

> **Tip 3.** *It is highly recommended that at any one time only one person in your lab. pair opens up these files and edits them by clicking on them from within the Jupyter notebook server browser, whilst the other student watches their screen display through screen sharing via programs such as Zoom or Skype. This is so that there is only one version of the files that*

are currently being edited. If both you and your lab. partner have the files open then there is the possibility for one version to overwrite the other as Jupyter notebook has an autosave feature.

Tip 4. *It is important over the course of the lab. that both members of the pair get to do the editing and script execution from time to time; this more active role should not be monopolised by one member of the pair!*

It is worth mentioning at this point that **this is a physics experiment and not a computing exercise!** Your knowledge of python/C++ for this experiment need not extend past the use of `if` and `for` statements, which you should remember from last year. For those who need a refresher, please see the resources available on, e.g., docs.python.org, cplusplus.com.

You should now open *Analysis.py* or *Analysis.C* and *Analysis.h* from the Jupyter notebook server browser window on your machine. These files will be inside the ATLAS-Project directory.

> **Exercise 5.2.** *Try and get familiar with the layout of the skeleton code in Analysis.py or Analysis.C. Make sure to add your names and the date at the top in a comment to help you keep records! Make sure to read through the comments to help work out what everything does. N.B. Please do* ***not*** *modify any of the service code in the sub-directory backend!*

As you can see, the skelton code contains some simple analysis code and makes a few histograms. For example, the code accesses the integer variable `lep_n` — defined as the number of leptons identified in each event. The code uses the vector `lep_pt` — defined as the vector of the lepton momenta in the plane transverse to the beam direction[1] — e.g. for an event containing three leptons, `lep_pt` will contain the three leptonic momenta[2] at `lep_pt[0]`, `lep_pt[1]`, and `lep_pt[2]`.

The definition of all the available variables on the ATLAS OpenData data sets for this experiment can be found in this web-page. The variable names to use in your analysis code are given in the column labelled **ntuple branchname**. In the code the variable names are all declared in the file *CLoop.h* in the sub-directory *backend*. N.B. The values of energies and momenta are all given in units **MeV**!

5.6 Running the skeleton code and examining the produced histograms

Before you try adding to the code, you should try running the existing skeleton analysis code and check you can view the output histogram file(s).

Similarly to previous subsections it is highly recommended that one person in your lab. pair runs the code and examines the results, whilst the other student watches their screen display through screen sharing via programs such as Zoom or Skype.

In order to run the analysis code and examine the results, only one person should open up the *LabNotebook.ipynb* notebook. Instructions on how to use the notebook are detailed within the notebook when you run it. For example, to analyse the Open Data and produce histograms

[1] The plane transverse to the beam direction is the 'x-y plane' in cartesian coordinates. The z coordinate points along the beam direction.

[2] Remember that in C++ the first element in a vector is at the location `[0]` and not `[1]`.

Monte Carlo data sets

physics process	string code
$Z \to ll \quad m_{ll} > 60$ GeV, $\mathrm{n\_lep} > 1$	‘*Z*’
$Z \to ee \quad m_{ee} > 60$ GeV, $\mathrm{n\_lep} > 1$	‘*Zee*’
$Z \to \mu\mu \quad m_{\mu\mu} > 60$ GeV, $\mathrm{n\_lep} > 1$	‘*Zmumu*’
$Z \to \tau\tau \quad m_{\tau\tau} > 60$ GeV, $\mathrm{n\_lep} > 1$	‘*Ztautau*’
$Z \to ll \quad m_{ll} > 60$ GeV, $\mathrm{n\_lep} = 1$	‘*Z_1lep*’
$W^+ \to l^+\nu \quad \mathrm{n\_lep} = 1$	‘*Wplus*’
$W^- \to l^-\nu \quad \mathrm{n\_lep} = 1$	‘*Wminus*’
$W^+ \to l^+\nu \quad \mathrm{n\_lep} > 1$	‘*Wplus_2lep*’
$W^- \to l^-\nu \quad \mathrm{n\_lep} > 1$	‘*Wminus_2lep*’
$t\bar{t} \to leptons \quad \mathrm{n\_lep} > 1$	‘*ttbar_lep*’
$t\bar{t} \to leptons \quad \mathrm{n\_lep} = 1$	‘*ttbar_lep_1lep*’
$H \to ZZ \to llll \quad \mathrm{n\_lep} > 1$	‘*H*’
$ZZ^{(*)} \to llll \quad m_{ll} > 40$ GeV, $\mathrm{n\_lep} > 1$	‘*ZZllll*’
$H \to \gamma\gamma$	‘*Hyy*’

ATLAS real data sets

data set	string code
Two or more leptons per event	‘*2lep*’
Exactly one lepton per event	‘*1lep*’
Two photons	‘*yy*’

Table 4: The “string code” for each data set corresponds to the name you should type in *RunAnalysis.py* when you want to analyse that particular data set. Data sets are split by the number of leptons per event, $\mathrm{n\_lep}$. In most cases you will want to use the data with $\mathrm{n\_lep} > 1$. The full list of available processes can be found in the file *backend/dataSets.py*.

you should click on the line `%run ’RunAnalysis.py’` to highlight it and then click on the tab near the top of the browser page labelled `Run` and follow the instructions that will be displayed. The program will issue the message “*Run completed!*” when all the requested input Open Data files have been processed successfully.

If you have edited any of the scripts in *LabNotebook.ipynb* and wish to save the edited version as the new default you should select “`Save and Checkpoint`” from the “`File`” pull-down menu. If you wish to undo recent edits and return to the last version you explicitly saved then select “`Revert to Checkpoint`” from the “`File`” pull-down menu.

Table 4 shows the list of data sets that are available for analysis.

5.7 More information about the code in *Analysis.py*

The file *Analysis.py* contains a single function, *Analyse*. It is called once per data set analysed. In this function, you will select events, plot histograms, set their style and write them to an output file.

5.8 More information about the code in *Analysis.C* and *Analysis.h*

The file *Analysis.C* contains three functions:

- Function *Book* is where you create or 'book' new histograms. This function is called only once per input file of events. It is called BEFORE any events are read in.
- Function *FillHist* is where you select events and enter information into (or 'fill') your histograms. This function is called ONCE PER EVENT.
- Function *Style* is where you can modify your histograms in various ways and write them to an output file. This function is called only once per input file. It is called AFTER all the events have been processed by Function *FillHist*.

The file *Analysis.h* is is where you declare the C++ pointers to your histograms to let the program know about them.

5.9 Modifying the code

Exercise 5.3. *By modifying the examples in the code, try to plot histograms of lepton p_T in the $Z \to ee$ data set separately for positively charged and negatively charged leptons and separately for electrons and for muons. Remember, if you use C++ you will need to declare the pointer to each new histogram in Analysis.h.*

If you have edited the code and wish to save the edited version as the new default you should select "`Save`" from the "`File`" pull-down menu.

In order to keep a permanent record of a set of histograms for future reference it can be useful to rename the output file so that it does not get overwritten the next time you run through the same data set. This can be achieved by renaming the file in the *out* folder of the *ATLAS-Project* directory using the Jupyter notebook server browser window.

In a similar vein, you may want to consider keeping an "archive" version of your code at points at which you have completed one chunk of the analysis and are about to embark on major code modifications. Do this by making copies of *Analysis.py* or *Analysis.C* and *Analysis.h*, and renaming them suitable names.

For more information on booking, filling, and formatting histograms, please see the relevant sections in the online `ROOT` documentation.

6 Analysis

Now we can get started with some actual physics analysis!

6.1 General Introduction

Each of the separate "Monte Carlo" data sets given in Table 4 represents a computer simulation of a specific physics process. For example, in every event in the file corresponding to the string code '*Zmumu*' a Z boson has been produced and has decayed to an opposite-charge pair of muons ($Z \to \mu\mu$). In contrast, the files corresponding the real data from the ATLAS experiment contain all relevant physics processes mixed up together within the one file. One of the main challenges of this analysis project is to figure out how to select a sample of events corresponding to a specific physics process from the ATLAS data.

We shall use the individual Monte Carlo samples to allow us to find out about the characteristic properties of events that correspond to particular physics processes. This will enable us to design "selection cuts" that we can apply to select a sample of events in the ATLAS data that correspond to the desired particular physics processes (our "signal"). In choosing our selection cuts there is always a trade off between the conflicting aims of achieving as high as possible a selection efficiency for the signal and rejecting as much as possible of the "background" from other physics processes. There is no unique "right answer" to such problems. Whilst it is important to achieve a reasonable compromise in the choice of selection cuts, it is even more important to make estimates of the selection efficiency and level of residual background and to come up with credible statistical and systematic uncertainties on these estimates.

In each event in a particular Monte Carlo data set the specified physics process (e.g., the $Z \to \mu\mu$) will have been generated. Note, however, that the generated particles are not always actually observed in the ATLAS detector. Note also that the events may contain additional particles (e.g., jets of hadrons produced by radiation from the incoming partons before the annihilation that produced the Z boson).

6.2 Analysing Z bosons

Using relativistic kinematics, one can derive an expression for the invariant mass of a parent particle that decays into multiple daughter particles. For example, in the simple decay mode of a Z boson to two muons ($Z \to \mu\mu$), we can use 4-momentum conservation and a little bit of algebra to find out the invariant mass of the lepton pair system $m_{\ell\ell}$.

ATLAS uses a right-handed coordinate system, with its origin at the nominal interaction point in the centre of the detector and the z-axis along the beam pipe. In the transverse plane, the x-axis points from the interaction point to the centre of the LHC ring, the y-axis points upwards.

In the data, the kinematic variables you are given for each lepton are the transverse momentum (p_T), the azimuthal angle between the p_T and the x-axis (ϕ), and the pseudorapidity (η). Pseudorapidity is defined as

$$\eta = -\ln\left(\tan\left(\frac{\theta}{2}\right)\right), \tag{1}$$

where θ is the usual polar angle. For massless particles, this is equivalent to the rapidity (y), defined as

$$y = \frac{1}{2}\ln\left(\frac{|\mathbf{p}| + p_L}{|\mathbf{p}| - p_L}\right). \tag{2}$$

where p_L is the longitudinal momentum and **p** is the three-momentum of the particle.

Since you will most likely be far more comfortable working in terms of p_x, p_y and p_z, you will need to make the appropriate change of variables in order to plot a histogram with the data given.

Exercise 6.1. *Derive a conversion from the kinematic variable set (p_T, ϕ, η) to (p_x, p_y, p_z). You may find it useful to draw a diagram!*

Tip 5. *For working out an expression for p_z you may want to remind yourself of the half-angle formula for $\tan\theta$.*

Exercise 6.2. *Derive an expression for $m_{\ell\ell}$ for the decay $Z \to \mu^+\mu^-$.*

Tip 6. *You may want to remind yourself of the formula from your first year relativity lectures $m^2 = E^2 - p^2$. Remember that here E will be the total energy of the $\mu^+\mu^-$ system and p is the total momentum of the system (i.e., the* ***magnitude*** *of the* ***vector sum*** *of the momenta of the μ^+ and μ^-).*

Tip 7. *Alternatively, if you are comfortable with 4-vectors you may want to use the formula that $m^2 = \tilde{p}^2$, where $\tilde{p}$ is the sum of the 4-vectors of the μ^+ and μ^-.*

Tip 8. *You may neglect the mass of a muon in your final expression.* ***Question:*** *Why can we do this?*

The invariant mass of the system can also be expressed as directly using p_T, η and ϕ of the leptons;

$$m_{\ell\ell} = \sqrt{2p_{T1}p_{T2}\left(\cosh\left(\eta_1 - \eta_2\right) - \cos\left(\phi_1 - \phi_2\right)\right)}. \tag{3}$$

Exercise 6.3. *Try to prove Equation 3 using your result for $m_{\ell\ell}$ in Cartesian coordinates and your conversions from p_T, η and ϕ. This can be quite mathematically involved so* ***if you get stuck move on for now and return to this maybe outside lab. time****.*

Exercise 6.4. *Produce a histogram of $m_{\ell\ell}$ by running your code on the $Z \to \mu\mu$ MC data set. Look up the mass of the Z boson to verify your plot. Afterwards, run your code on the ATLAS '2lep' data set and produce a second plot. Discuss with your lab partner whether or not, and if so why, you think these plots are different.*

6.3 Making event selection cuts

To select the highest quality sample of signal events, particle physicists impose "selection criteria", or "cuts" on each event. For the decay $Z \to \mu^+\mu^-$, we are clearly only interested in events with **oppositely charged** leptons of the **same type** or **"flavour"** (that is electrons or muons). If you have not included these restrictions in your code already, you should do so — you may see a small change in your mass distribution, especially in the ATLAS data.

Something else we can use to restrict the events we plot is the p_T of the leptons. In general, we always want to use the particles with the highest p_T.

Exercise 6.5. *This exercise is useful only to those using the C++ version of the code; do not bother trying to do this if you are running the python version. You will find some events in which there are two or more leptons of the* ***same flavour*** *with the* ***same charge****. In each such case, select the lepton that has the highest p_T. Reproduce histograms of $m_{\ell\ell}$ by running your code on the $Z \to \mu\mu$ and the muon data sets. Compare your results with your previous analyses — again, you may see a small change in your mass distribution.*

Tip 9. *For C++ users only. For rather obscure reasons the C++ interpreter in* `ROOT` *does not always handle correctly C++* `vectors`*. Instead of creating your own private C++* `vectors` *and using the command* ``push.back'' *to store entries, we suggest you store the* ``index'' *into the* ``lep_xxx'' `vectors` *that are already supplied in the input data sets. Ask your demonstrator for help if it is not clear how to achieve this.*

Exercise 6.6. *Now, modify and run your analysis code to find the invariant mass of $Z \to ee$ on the appropriate MC data set and on the '2lep' data set. Discuss with your lab. partner other ways you think you could reduce the background.*

Leptons produced in the decay of W and Z bosons tend to be "isolated" from other particles produced in proton-proton collisions. In contrast, leptons from "background" processes (such as the decay of b quarks) tend to be accompanied by a "jet" of other particles. The ntuple variable `ptcone30` contains a sum over the p_T of all tracks contained within a cone of half-width 0.3 in ΔR around the lepton direction. ΔR is given by

$$\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2},$$

where $\Delta\phi$ and $\Delta\eta$ are, respectively, the differences in azimuthal angle and pseudorapidity between the lepton and any additional tracks. The ntuple variable `etcone20` contains a sum over the p_T of all calorimeter clusters contained within a cone of half-width 0.2 in ΔR around the lepton direction. The above sums exclude the tracks and clusters associated with the lepton candidate itself. The quantities `ptcone30` and `etcone20` are sometimes collectively referred to as the "isolation variables".

Exercise 6.7. *Plot graphs of the variables* `ptcone30` *and* `etcone20` *for the leptons that are members of the pairs of oppositely charged, same flavour leptons that you have selected above. Do this for each of the three data sets you have been using so far ('Zee', 'Zmumu', '2lep'). These are representative of the "signal" leptons from Z decay you are targeting. Can you think of any way of looking at the distributions of the variables* `ptcone30` *and* `etcone20` *for leptons that are more likely to be "background" than signal? Use the resulting histograms to help you decide whether or not, and if so where, to apply selection cuts using these variables.*

6.4 Event weights for Monte Carlo data sets

The two provided ATLAS data sets correspond to an "integrated luminosity", $\int L\,dt$, of 10.064 fb^{-1} or "inverse femtobarns". The integrated luminosity is a measure of the number of proton–proton collisions in ATLAS to which the data sets correspond. The provided Monte Carlo data sets correspond in general to higher integrated luminosities and thus have to be "scaled down" in order to be compared with the ATLAS data. In addition, various corrections need to be applied to the Monte Carlo to account for deficiencies in the simulation of the performance of the detector. This is achieved by means of "event weights" that must be used whenever a histogram is filled or a counter for the number of selected events is incremented. The relevant variable in the code is called *weight*; it is calculated for you event by event.

Tip 10. *In order to get useful additional information about the contents of your histograms in the “statistics box” that is usually displayed in the top right hand corner of the* `ROOT` *“Canvas” it is useful to execute the* `ROOT` *command* ***gStyle−>SetOptStat(1111111)****.*

Tip 11. *Note that for histograms of the Monte Carlo data sets it is important to distinguish between the number of* `Entries` *and the value* `Integral` *displayed in the statistics box. Technically speaking, the number of* `Entries` *corresponds to the total number of times the* `ROOT` *function* `Fill` *has been called for the histogram in question from within your job. In contrast, the value* `Integral` *corresponds to the sum of weights for all of the calls of* `Fill` *for which the plotted value lies between the lower and upper bounds you have specified for the histogram in question. When comparing histograms of ATLAS data and Monte Carlo, it is value of the* `Integral` *that should be compared, not the number of* `Entries`*.*

6.5 Cross Sections, backgrounds and efficiencies

The “cross section” σ for a given process, such as $Z \to \ell\ell$, can be evaluated as follows:

$$\sigma\,(pp \to Z \to \ell\ell) = \frac{N^{\mathrm{selected}} - N^{\mathrm{background}}}{\epsilon\,\int L\,dt}, \tag{4}$$

where N^{selected} is the total number of events in the ATLAS data that pass your final selection cuts, $N^{\mathrm{background}}$ is your estimate of the number of background events in the selected data sample. Therefore, $N^{\mathrm{selected}} - N^{\mathrm{background}}$ is your estimate of the number of “signal” events in the ATLAS data for the targeted physics process.

The quantity ϵ is the efficiency for selecting the “signal” events. “signal” here means whichever physics process you are measuring the cross section for: $Z \to \ell\ell$ in the first part of the lab. and $W \to \ell\nu$ in the second part. The value of ϵ can be estimated using the Monte Carlo event sample corresponding to the targetted signal process. The sum of weights for all signal Monte Carlo events that pass your selection cuts needs to be divided by that corresponding to all generated events for the relevant sample. This information is provided by running the python script *TotExpected.py*. For example, to get the sum of weights for all generated $Z \to \ell\ell$ signal Monte Carlo events:

python3.4 TotExpected.py	Launches the script to retrieve the sum of weights for all generated signal Monte Carlo events
Zee	Request this information for the $Z \to \ell\ell$ simulated data set

Tip 12. *In the first instance you are going to be using the $Z \to \ell\ell$ Monte Carlo samples ‘Zee’ and ‘Zmumu’ to evaluate the selection efficiency ϵ. Think carefully about your event selection cuts. Are they designed to exclude, as far as reasonably possible, any events that would not have been simulated in the ‘Zee’ and ‘Zmumu’ Monte Carlo samples?*

The fractional uncertainty on $\int L\,dt$ is 1.7%. The determination of the luminosity and its uncertainty in ATLAS at 13 TeV is described in the ATLAS note ATLAS-CONF-2019-021. It is usual to quote the uncertainty on the cross section arising from the luminosity separately from the uncertainties arising from other sources.

Other systematic uncertainties may arise, for example, from:

- backgrounds that you have not accounted for,
- disagreements between the simulation of the $Z \to \ell\ell$ signal in the Monte Carlo and the real ATLAS data.

A possible "catch-all" method to estimate the size of such effects is to re-calculate the cross section having changed your event selection cuts. With regard to $N^{\text{background}}$ you might want to look back at Table 4; can you think of any possible sources of background in your selected $Z \to \ell\ell$ samples and how you might investigate these? Can you think of any other sources of statistical and systematic uncertainty on your measurements of $\sigma\left(pp \to Z \to ee\right)$ and $\sigma\left(pp \to Z \to \mu\mu\right)$?

Present your measurements of the cross section for Z boson production $Z \to \ell\ell$ in the form:

$$\sigma\left(pp \to Z \to \ell\ell\right) = ??? \pm XXX(\text{stat.}) \pm YYY(\text{syst.}) \pm ZZZ(\text{lumi.}), \tag{5}$$

where XXX is your estimate of the statistical uncertainty, YYY is your estimate of the systematic uncertainty, and ZZZ is due to the luminosity uncertainty. Please make sure you keep a good record in your lab. book of your calculations of cross sections and the various uncertainty components! Please make sure you give the correct units for your cross section! What kind of "technical" and "physics" questions should you be asking yourself about your measured cross sections and uncertainties?

Tip 13. *The fast mode is very useful to speed things up when you are developing your analysis. However, for your final numbers and plots you should run through the full available data and MC files.*

6.6 Re-discovering the Higgs boson

Now that you know how to measure a cross section we can move on to particles which are trickier to find than the Z boson. In the simplest Higgs decay mode to search for ($H \to ZZ^{(*)} \to \ell\ell\ell\ell$) you will now need to look at events which produce four leptons. Since two Z bosons are formed as an intermediate state, the leptons should form two same-flavour, opposite-charge pairs.

To calculate the invariant mass of the four-lepton system, you can extend the expression you obtained in answer to Exercise 6.2. Alternatively, you can extend Equation 3 to sum over all unique combinations of two leptons per event.

$$m_{\ell\ell\ell\ell}{}^2 = 2\sum_{i=0}^{3}\sum_{j=i+1}^{3} p_{Ti}p_{Tj}\left(\cosh\left(\eta_i - \eta_j\right) - \cos\left(\phi_i - \phi_j\right)\right). \tag{6}$$

Much of the methodology of the H analysis will be the same as for the Z analysis. However, selection cuts will be more complex to apply now that there are four leptons in the final system.

Exercise 6.8. *Try to verify Equation 6.*

Exercise 6.9. *Produce a plot of the $m_{\ell\ell\ell\ell}$for the $H \to ZZ^{(*)} \to llll$ Monte Carlo. Repeat for the real data. You will find that these do not agree as nicely as the Z boson data. Why is this?*

6.7 Analysing W bosons

W bosons are somewhat more difficult to select and analyse than Z bosons, as they decay to produce a single charged lepton and a neutrino, $W \to \ell\nu$. Neutrinos do not interact in the detector and lead to an apparent imbalance in the vector sum of the p_T of all the observed particles in an event — this is called the "missing momentum" or "**missing energy**". The magnitude (E_T^{miss}) and azimuthal angle (ϕ^{miss}) of the missing energy are interpreted as the transverse components of the neutrino momentum[3]. Because very energetic particles may escape detection if they are produced very close to the beam direction, only the transverse components of the neutrino momenum can be measured in ATLAS. Because of this limitation we are unable to calculate the total invariant mass of the decay products of W bosons. Instead, we use a quantity called the "**transverse mass**", m_T, which uses only the transverse components of the charged lepton and neutrino momenta. It is defined as follows:

$$m_T = \sqrt{2E_T^{miss} p_T^l (1 - cos(\Delta\phi))}, \tag{7}$$

where $\Delta\phi = \phi^{miss} - \phi^l$, and ϕ^l and p_T^l are the ϕ and p_T of the lepton.

Tip 14. *When analysing W bosons you will need to include the data sets with 1 lepton per event as only one charged lepton is produced by the decay of a W boson.*

Exercise 6.10. *Produce a plot of m_T for one of the $W \to \mu\nu$ Monte Carlo data sets. Compare with the peak in $m_{\ell\ell}$ that you found for Z bosons. How do you interpret the differences in shape between the two distributions?*

Exercise 6.11. *Now produce a plot of m_T for the ATLAS '1lep' data set. Compare this with the results from the $W \to \mu\nu$ Monte Carlo. How do you interpret the differences in shape between the two distributions?*

The biggest challenges in analysing W decays are:

- to reduce the level of background from other processes in the selected event samples;
- to evaluate the residual numbers of background events in the selected event samples, together with an estimate of the uncertainty on that evaluation.

Both of these challenges can be addressed by the approach of plotting the distribution of each of the variables that most effectively discriminate between signal and background **without making a cut on the variable in question, but having applied all other selection cuts**. For each of these discriminating variables can you understand the distribution you see in the ATLAS data as the sum of contributions from the $W \to \ell\nu$ signal and those from the various backgrounds?

[3] On the ntuples these variables are called met_et and met_phi respectively.

6.8 Some thoughts on extensions to your analysis

This is a very open-ended project. Whilst the main aim is to measure the cross sections for Z, W and Higgs boson decays to leptons, if you have time the provided ATLAS data and Monte Carlo allow many other interesting investigations of the production of particles to be made. In addition, you get credit for spotting interesting/unexpected features in the data and following up on these; feel free to discuss your ideas and observations with your demonstrator! It may also be possible to select and study in the ATLAS data candidate event samples corresponding to some of the other physics processes for which Monte Carlo samples are listed in Table 4. For example, you might consider other sources of events at the LHC, such as Higgs decaying to two photons, or events containing tau leptons, pairs of vector bosons, top quarks, or potential sources of new physics from "beyond the standard model".

6.9 Your feedback

We very much welcome your feedback/constructive criticism on the experiment, software, and documentation. Please feel free to talk to your demonstrator and/or send email to Terry Wyatt at terry.wyatt@cern.ch.

7 Appendices

7.1 Collected hyperlinks

- The ROOT users' guide.
- To learn more about Feynman diagrams see, e.g., this resource.
- Information about the ATLAS collaboration, the detector, the physics etc. can be found here.
- A journal paper describing the measurements of Z and W production in the very early ATLAS data at the LHC can be found here.
- The ATLAS Open Data is a project designed by ATLAS to enable the "general public", including students, to explore data collected at the LHC the techniques used in modern experimental particle physics.
- For details on the data sets you are using, you can consult the ATLAS OpenData online documentation for reference.
- For a refresher on python, please see the resources available on, e.g., docs.python.org.
- For a refresher on C++, please see the resources available on, e.g., cplusplus.com.
- The definition of all the available variables on the ATLAS OpenData data sets for this experiment can be found in this web-page.
- For information on booking, filling, and formatting histograms, see the relevant sections

in the online `ROOT` documentation.

The determination of the luminosity and its uncertainty in ATLAS at 8 TeV is described in the ATLAS note ATLAS-CONF-2019-021.

7.2 What do I do if I get a pop-up on my Jupyter notebook server browser window message saying I have lost connection?

If this happens to you and you are connected to the internet, first copy all cells to a notepad file that you editted since the notebook was last saved. Then close all Jupyter windows in your browser. After this, please go to the terminal window you used to complete steps 4-9 of section 5.3, and a message should display saying ‘connection reset...’. Starting from the new command line that should appear, please complete steps 5-9 of section 5.3 again. Once you are back using Jupyter, return to the notebook, and copy and paste any material necessary from the notepad file you made to the appropriate place. If this happens to you and you are not connected to the internet anymore, first try to reconnect to the internet BEFORE completing the steps above.

7.3 Running the code from command line

The current default directories you copied across in section 5.2 when choosing which language to use are not set up to run the analysis script from the command line. However, there is a way around this.

Warning: Upon completing the following steps, you will no longer be able to run the *RunAnalysis.py* from Jupyter notebook until you carry out the steps listed at the end of this section.

To use the command line to run the script, you must complete the following:

- Open up a separate new terminal window and log in the lab. machine using the steps in section 5.1. Then you need to type the following commands:

cd ATLAS-Project	Changes your “current directory” to `ATLAS-Project`.
rm -f ATLAS	Removes the *ATLAS* directory already existing.
ln -s /data/ATLAS/ ATLAS	Creates a required logical link to the ATLAS data files.
python3.4 RunAnalysis.py	In the terminal window, you run a python script by typing **python3.4 filename**, where in our case, we are running the analysis macro *RunAnalysis.py*.

To revert these changes and return to using the notebook *LabNotebook.ipynb* to run *RunAnalysis.py*, you need to follow the next steps:

- Open up a separate new terminal window and log in the lab. machine using the steps in section 5.1. Then you need to type the following commands:

cd ATLAS-Project	Changes your “current directory” to `ATLAS-Project`.
rm -f ATLAS	Removes the logical link to the ATLAS data files.

- Next, close down and restart the Jupyter notebook server by following the appropriate steps in sections 5.4 and 5.3 respectively.

7.4 A few hints on archiving your work and keeping a logbook

It can be tempting to think that because you are making lots, and lots, and lots, of plots these can replace the need to keep a logbook. Please do not fall into this trap! You need to keep a systematic record of the investigations you have carried out, the calculations you have performed and, most important of all, a summary of conclusions you have reached. I can guarantee that part of the question and answer session following your talk will require you to look up in your logbook additional information that is not contained in your talk. Students who can quickly and efficiently find the required information and answer questions in a clear and concise fashion gain extra marks, and conversely . . . ;-)

Particularly in regard to the conclusions, I suggest sitting down at the end of each lab. day and carefully reviewing your logbook, writing down any details you forgot to record, and making sure you have written down appropriate conclusions. I strongly suggest that as part of this daily review you write down what you plan to do and what you hope to achieve during the **n**ext lab. day! Five minutes taken at the end of one lab. day can enormously improve your productivity during the next lab. day!

You should feel free to keep either a paper or an electronic logbook. The choice is yours, but . . .

- If you keep a paper logbook then make sure you print out and stick into your logbook copies of the absolutely vital plots that you have archived in the sub-directory *outputPlots*.

- If you keep an electronic logbook then make sure you do this on a laptop that you **must** bring to the interview. Make sure that you can add plots, details of calculations, etc. to your electronic logbook, and that under the time pressure of the interview you are able to find information as quickly and efficiently as if you’d kept a paper logbook. If you have any doubts on this score then the safer option may be to keep a paper logbook.

7.5 A few hints for preparing your talk

You should be aiming to deliver your talk in ten minutes. This is a quite a small amount of time: spend most of it telling us about **your** analysis, results and conclusions. Be specific: show us actual plots, numbers, uncertainties, and not just generalities. Describe the cuts you have employed and show plots to justify why you chose those cuts. Give quantitative information on your estimation of the uncertainties on your measurements.

Whilst giving a small amount of “background” and “introductory” information is fine, do not waste large amounts of time just telling us what we alredy know from this lab. script!

Please add **page numbers** to your talk! It helps us ask questions about specific items in your talk.

Please add **your names** to the tile page of your talk!

You might want to consider adding some “Backup” slides to your talk, that contain additional plots, details of your calculations, etc., that might be useful in answering our questions.

If you did anything that you think went “beyond the script” then remember to tell us about it!

Remember that although the “fast” mode in *RunAnalysis.py* is very useful to speed things up when you are developing your analysis, for your final numbers and plots you should run through the full available data and MC files!

Some really basic stuff that you should have learned from previous years’ labs:

Make sure every plot you show has an intelligible title. Make sure the axes are labelled and have appropriate units, etc.

Make sure you and your partner rehearse your talk together and with a timer running. This is essential to ensure you can deliver the material in a fluent fashion and that you are able to keep to the allotted time.

Please bring your lab. books to the interview. Make sure you are able to use your lab. books to answer detailed follow-up questions about your work (for example, your choice of selection cuts, details on calculations of cross sections, estimation of systematic uncertainties, etc).