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 [1]:
```
import RunAnalysisThreads
import sys
sys.path.insert(0, "backend") # allow code to be imported from subdirectory
from AnalysisZee import Analyse

RunAnalysisThreads.main(Analyse)

#canvases = None   #uncomment to hide autoplotting
```

```
Dont forget you will need to restart the kernel for any edits you have made to Analysis.py to take effect, befor
e running RunAnalysis.py!

Welcome to JupyROOT 6.24/06
Please enter a comma-seperated list of decay chains.
Use '+' to add data sets together.
Write 'text' if you would prefer to read a list from 'input.txt':
text

Would you like to run in fast mode to only analyse 1% of data? (yes/no)
no
start thread to analysis 2lep...
start thread to analysis Zmumu...
start thread to analysis Ztautau...
start thread to analysis Zee...
A
start thread to analysis Wplus_2lep...
start thread to analysis Wminus_2lep...
Wplusenu_2lepstart thread to analysis ttbar_lep...

Wminusenu_2lepstart thread to analysis H...
start thread to analysis ZZllll...

ZH125_ZZ4lep
mc15_13TeV.361103.PwPy8EG_AZNLOCTEQ6L1_Wminusenu.2lep_raw.root  3.823 seconds
```

```
Wminusmunu_2lep
mc15_13TeV.361100.PwPy8EG_AZNLOCTEQ6L1_Wplusenu.2lep_raw.root  5.127 seconds
Wplusmunu_2lep
mc15_13TeV.341947.Py8EG_A14NNPDF23LO_ZH125_ZZ4l.2lep_raw.root  6.825 seconds
WH125_ZZ4lep
mc15_13TeV.361104.PwPy8EG_AZNLOCTEQ6L1_Wminusmunu.2lep_raw.root  3.199 seconds
Wminustaunu_2lep
mc15_13TeV.361105.PwPy8EG_AZNLOCTEQ6L1_Wminustaunu.2lep_raw.root  0.299 seconds
combined new filename: Wminus_2lep.root
mc15_13TeV.361101.PwPy8EG_AZNLOCTEQ6L1_Wplusmunu.2lep_raw.root  3.323 seconds
Wplustaunu_2lep
mc15_13TeV.361102.PwPy8EG_AZNLOCTEQ6L1_Wplustaunu.2lep_raw.root  0.325 seconds
combined new filename: Wplus_2lep.root
mc15_13TeV.341964.Py8EG_A14NNPDF23LO_WH125_ZZ4l.2lep_raw.root  5.897 seconds
VBFH125_ZZ4lep
mc15_13TeV.361108.PwPy8EG_AZNLOCTEQ6L1_Ztautau.2lep_raw.root  18.043 seconds
mc15_13TeV.361603.PwPy8EG_CT10nloME_AZNLOCTEQ6L1_ZZllll_mll4.2lep_raw.root  56.580 seconds
dataA_2lep.root  62.868 seconds
B
mc15_13TeV.344235.PwPy8EG_NNPDF30_AZNLOCTEQ6L1_VBFH125_ZZ4lep_notau.2lep_raw.root  50.811 seconds
ggH125_ZZ4lep
mc15_13TeV.345060.PwPy8EG_NNLOPS_nnlo_30_ggH125_ZZ4l.2lep_raw.root  37.122 seconds
combined new filename: H.root
mc15_13TeV.410000.PwPyEG_P2012_ttbar_hdamp172p5_nonallhad.2lep_raw.root  223.038 seconds
dataB_2lep.root  174.578 seconds
C
dataC_2lep.root  229.439 seconds
D
dataD_2lep.root  357.825 seconds
combined new filename: 2lep.root
mc15_13TeV.361107.PwPy8EG_AZNLOCTEQ6L1_Zmumu.2lep_raw.root  1257.209 seconds
mc15_13TeV.361106.PwPy8EG_AZNLOCTEQ6L1_Zee.2lep_raw.root  1846.308 seconds
All threads run completed!
```

"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 2 cells which all contain 2 different parts of the code structure for viewing histograms. The reason the code is split into 2 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. Retreiving and Printing the histograms from the .root file

When viewing your own histograms you will need to edit the code in the second cell for your specific .root files. This includes the .root file name, choosing which histograms you view and axis/title labels.

The second 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 this notebook. From this file you are viewing the first two histograms created: *h_lep_n* and *h_lep_pt*.

plotHistogram(...) can be found in the backend/ShowHistogram.py script. It takes the histogram (*hist*) as its only compulsory argument. The following are optional arguements: *title*, *x_label*, *y_label*, *color*, *log_y*. These change the labels that were initially set in Analysis.py.

A note about the second cell: Here a TCanvas is created for each histogram you would like to plot. Whenever you want to plot anything with ROOT you use a TCanvas. For more information on TCanvas, please click here. The second shell is also an example of showing two histograms from the .root file, it can easily be extended to view as many as needed.

In [9]:

```
#RUN THIS TO PLOT HISTOGRAMS WITHOUT RERUNNING THEM
import ROOT as r
from ShowHistogram import *
from RunAnalysis import *
from IPython.lib.display import Audio
import numpy as np


def plotFile(name, fMode):
    fStr = fastStr(fMode)
    # Open .root file - replace the first arguement with the name of the file
    hist_file = r.TFile.Open("out/" + name + fStr + ".root","READ")
    histograms, printString = histogramsFromFile(hist_file) #Retreives all histograms from hist_file
    print(printString) # Prints hist_id, titles and x-axis labels

    canvas = [0] * 50
    #Plot two histograms using their hist_id
    #canvas[0] = plotHistogram(histograms["h_lep_n"],x_label = "x-label") #Plots histogram["h_lep_n"]
    canvas[1] = plotHistogram(histograms["h_lep_pt"],title="Transverse Momentum pt")
```

```
    canvas[2] = plotHistogram(histograms["h_lep_pt0"],title="Transverse Momentum pt0")
    canvas[3] = plotHistogram(histograms["h_lep_pt1"],title="Transverse Momentum pt1")
    #canvas[4] = plotHistogram(histograms["h_lep_eta"],title="pseudo-rapidity lep_eta")
    canvas[5] = plotHistogram(histograms["h_lep_m"],title="Invariant Mass of the System", x_label = "mass(MeV)"
    #canvas[6] = plotHistogram(histograms["h_lep_ptcone"],title="ptcone of the events", x_label = "MeV", log_y
    #canvas[7] = plotHistogram(histograms["h_lep_etcone"],title="etcone of the events", x_label = "MeV")
    canvas[8] = plotHistogram(histograms["h_lep_pt_tot"],title="total pt of two leps", x_label = "MeV")
    canvas[10] = plotHistogram(histograms["h_lep_mll_cutpt"],title="Invariant Mass of the pt_cuts", x_label = "
    canvas[11] = plotHistogram(histograms["h_lep_mll_cutpttot"],title="Invariant Mass of the pttot_cuts", x_lab
    canvas[12] = plotHistogram(histograms["h_lep_mll_cutptcone"],title="Invariant Mass of the ptcone_cuts", x_l
    canvas[13] = plotHistogram(histograms["h_lep_mll_cutetcone"],title="Invariant Mass of the etcone_cuts", x_l

    tot_integral = histograms["h_lep_n"].Integral()
    mll_integral = histograms["h_lep_m"].Integral()

    hist_file.Close() # Close .root file

    print("name: " + name + fStr)
    print(f"  tot_integral: {tot_integral}")
    print(f"  mll_integral: {mll_integral}")
    print()

    return canvas,mll_integral


#canvas4 = plotFile("Zmumu_Ztautau_Wplus_2lep_Wminus_2lep_ttbar_lep_H_ZZllll", True)
fastflag = False


canvas3,tauint = plotFile("Ztautau", fastflag)
canvas2,Zmuint = plotFile("Zmumu", fastflag)
canvas1,Zeeint = plotFile("Zee", fastflag)
canvas4,Wplsint = plotFile("Wplus_2lep", fastflag)
canvas5,Wminint = plotFile("Wminus_2lep", fastflag)
canvas6,Ttbint = plotFile("ttbar_lep", fastflag)
canvas7,Hint = plotFile("H", fastflag)
canvas8,Z4lint = plotFile("ZZllll", fastflag)
canvas0,lepint = plotFile("2lep", fastflag)

Sigma_mu = (lepint - Zmuint - tauint - Wplsint - Wminint - Ttbint - Hint - Z4lint)/(10.064 *  Zeeint/ 19631161.
print()
print(Sigma_mu)
print()


framerate = 4410
play_time_seconds = 1

t = np.linspace(0, play_time_seconds, framerate*play_time_seconds)
audio_data = np.sin(2*np.pi*3000*t) + np.sin(2*np.pi*240*t)
Audio(audio_data, rate=framerate, autoplay=True)
```

```
histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: Ztautau
  tot_integral: 103123.90101122856
  mll_integral: 2342.3531485423446

histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: Zmumu
```

```
  tot_integral: 5225739.899299026
  mll_integral: 0.0

histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: Zee
  tot_integral: 4881660.1342282295
  mll_integral: 4117007.5978393555

histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: Wplus_2lep
  tot_integral: 182154.63067626953
  mll_integral: 302.87891800748184

histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: Wminus_2lep
  tot_integral: 148198.79583740234
  mll_integral: 315.5905334856361

histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: ttbar_lep
  tot_integral: 247589.19215613604
  mll_integral: 16663.490425109863
```

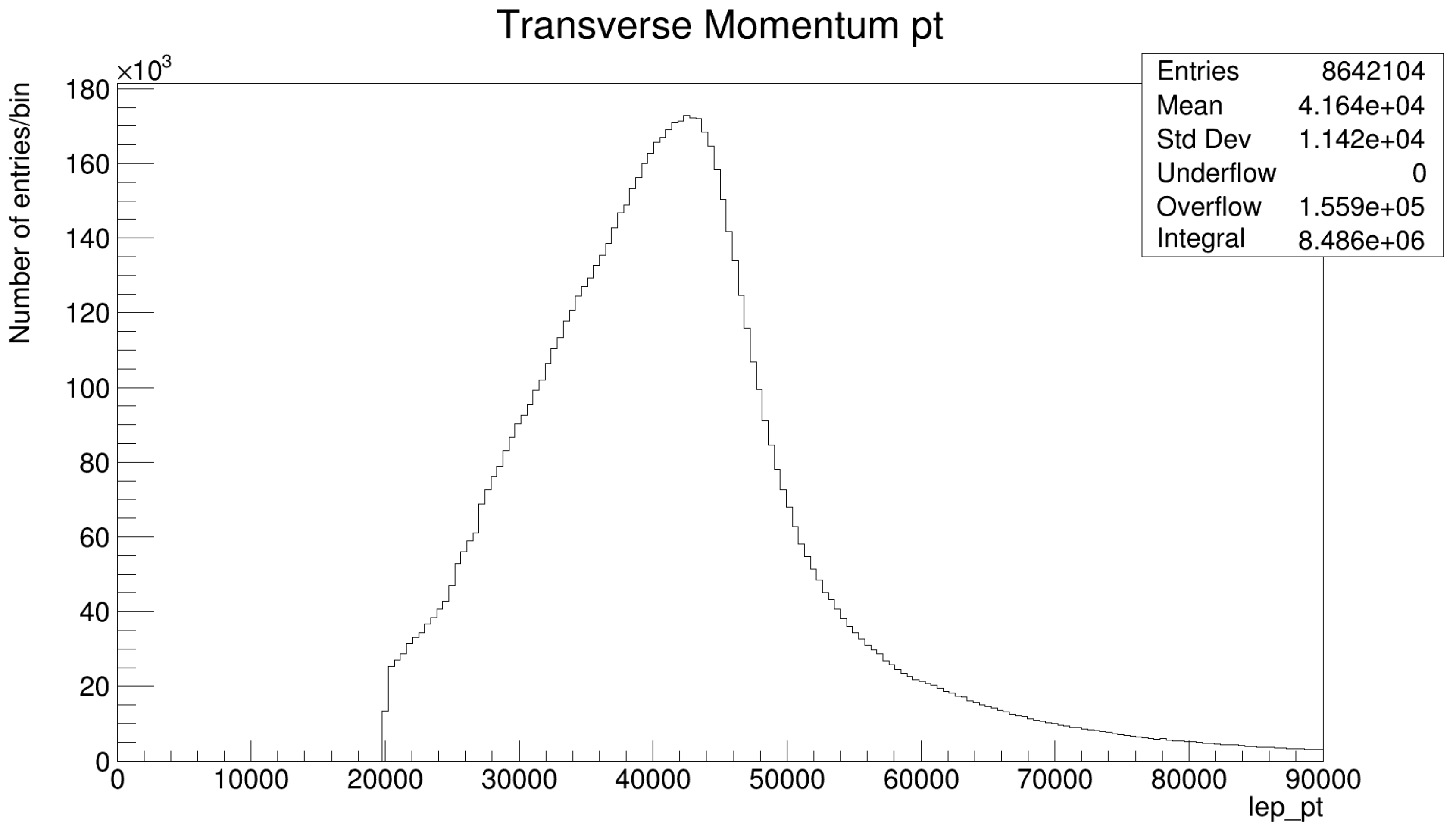
```
histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: H
  tot_integral: 32.46891002090659
  mll_integral: 1.0050395031948227

histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: ZZllll
  tot_integral: 2027.1842700652778
  mll_integral: 286.09752605110407

histograms["h_lep_n"], title: lep_n:1, x-axis: lep_n
histograms["h_lep_pt"], title: lep_pt:(lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11 &&
lep_type[1] == 11&&lep_eta[0] >=-2.4 && lep_eta[0] <=2.4 && lep_eta[1] >=-2.4 && lep_eta[1] <=2.4&&lep_pt[0] >=
20e3 && lep_pt[1] >= 20e3&&lep_ptcone30[0] < 4000 && lep_ptcone30[1] < 4000 && lep_etcone20[0] <= 5000 && lep_e
tcone20[1] <= 5000&&Mll >= 66e3), x-axis: lep_pt
histograms["h_lep_pt0"], title: lep_pt[0]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[0]
histograms["h_lep_pt1"], title: lep_pt[1]:lep_n == 2 && lep_charge[0]*lep_charge[1] == -1 && lep_type[0] == 11
&& lep_type[1] == 11, x-axis: lep_pt[1]
histograms["h_lep_m"], title: Invariant Mass of System, x-axis: Mll
histograms["h_lep_eta"], title: pseudo-rapidity lep_eta, x-axis: lep_eta
histograms["h_lep_ptcone"], title: ptcone30, x-axis: lep_ptcone30
histograms["h_lep_etcone"], title: etcone20, x-axis: lep_etcone20
histograms["h_lep_pt_tot"], title: lep_Pt_tot, x-axis: Pt_tot
histograms["h_lep_mll_cutpt"], title: lep_mll_cutpt, x-axis: Mll
histograms["h_lep_mll_cutpttot"], title: lep_mll_cutpttot, x-axis: Mll
histograms["h_lep_mll_cutptcone"], title: lep_mll_cutptcone, x-axis: Mll
histograms["h_lep_mll_cutetcone"], title: lep_mll_cutetcone, x-axis: Mll

name: 2lep
  tot_integral: 12205852.0
  mll_integral: 4270954.0


2014137.6775959835
```

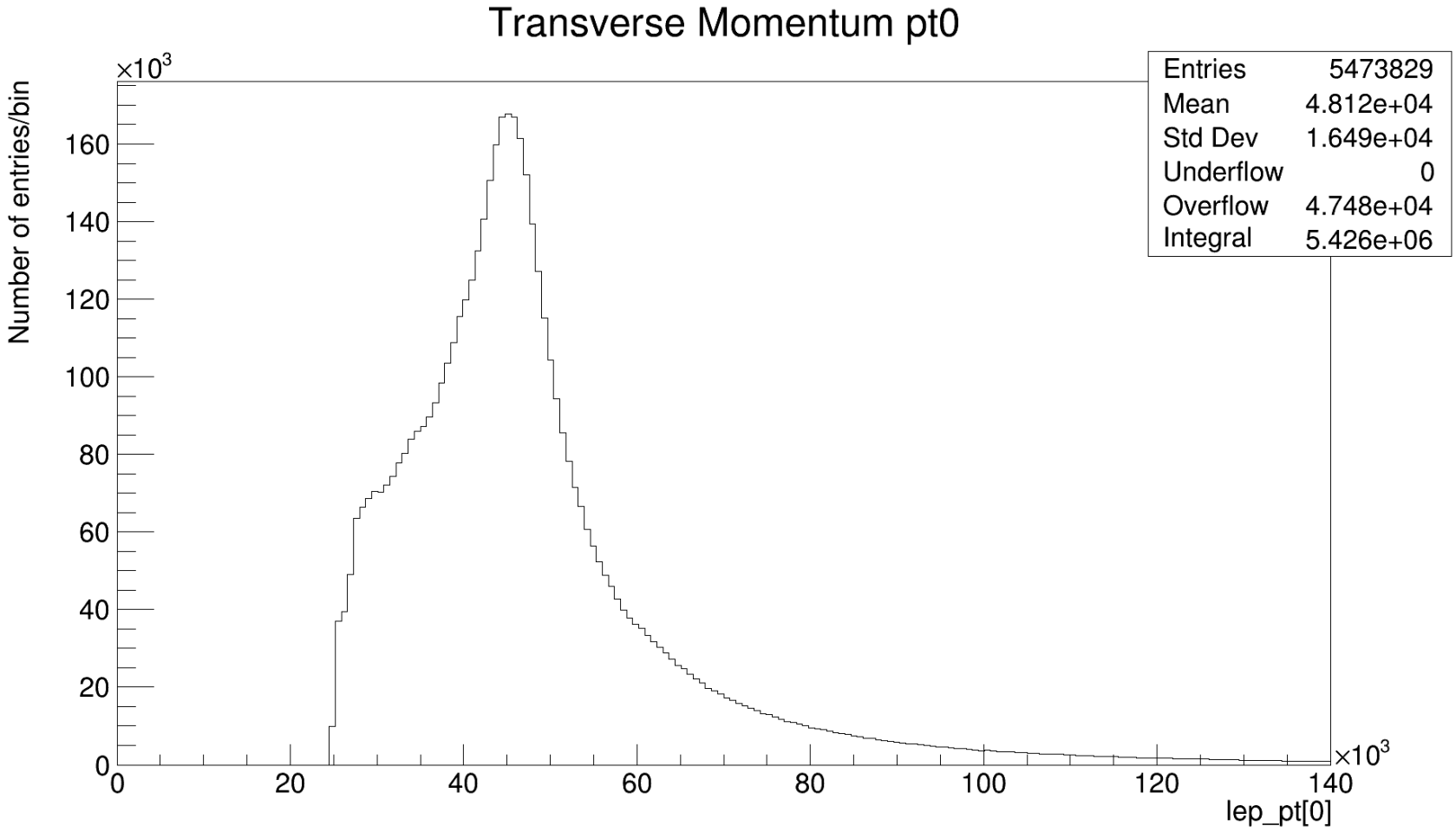
Out[9]:

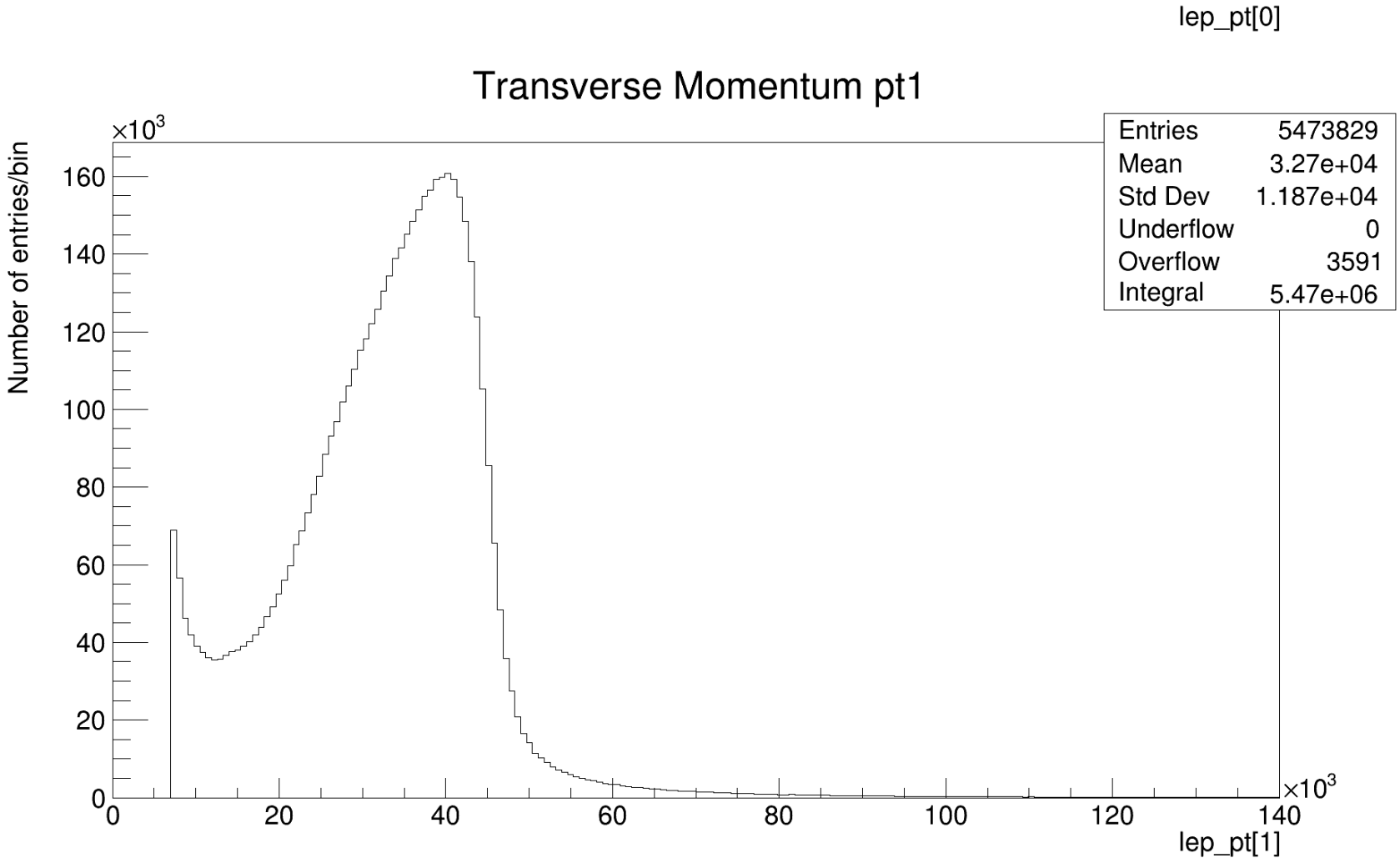
0:01 / 0:01

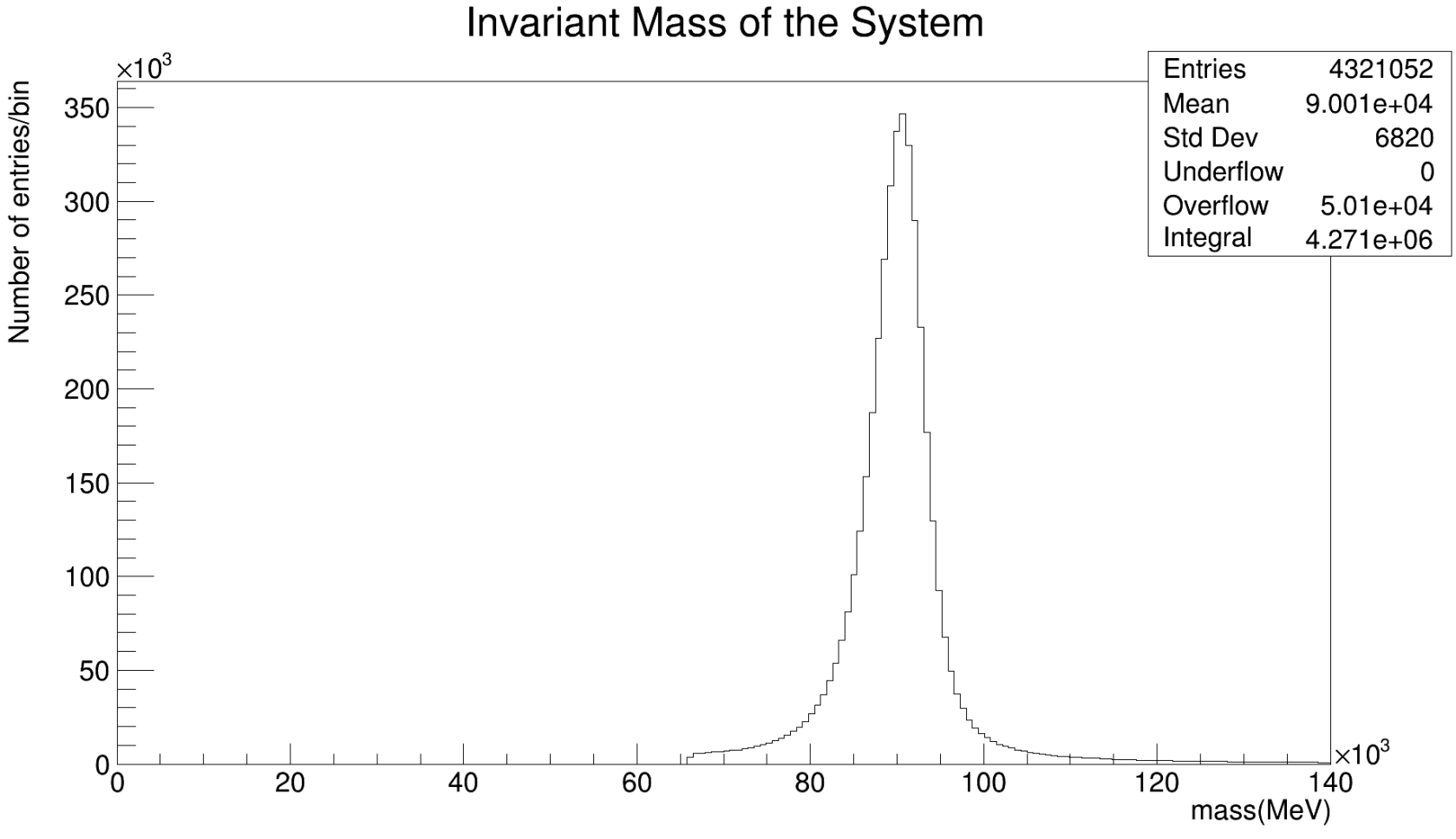
```
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
```

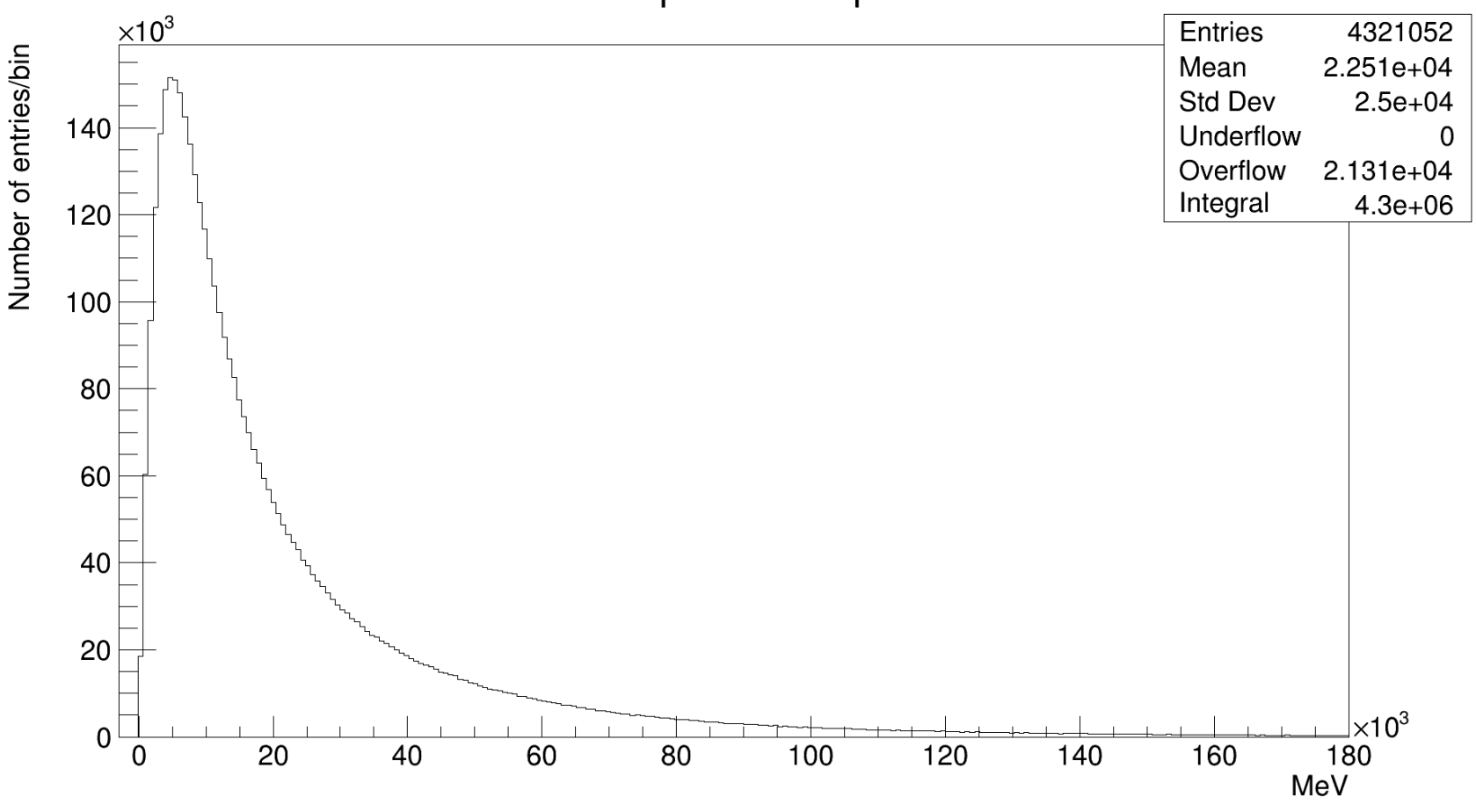
```
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt0
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt1
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_m
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_pt_tot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpt
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutpttot
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutptcone
Warning in <TCanvas::Constructor>: Deleting canvas with same name: canvas_h_lep_mll_cutetcone
```

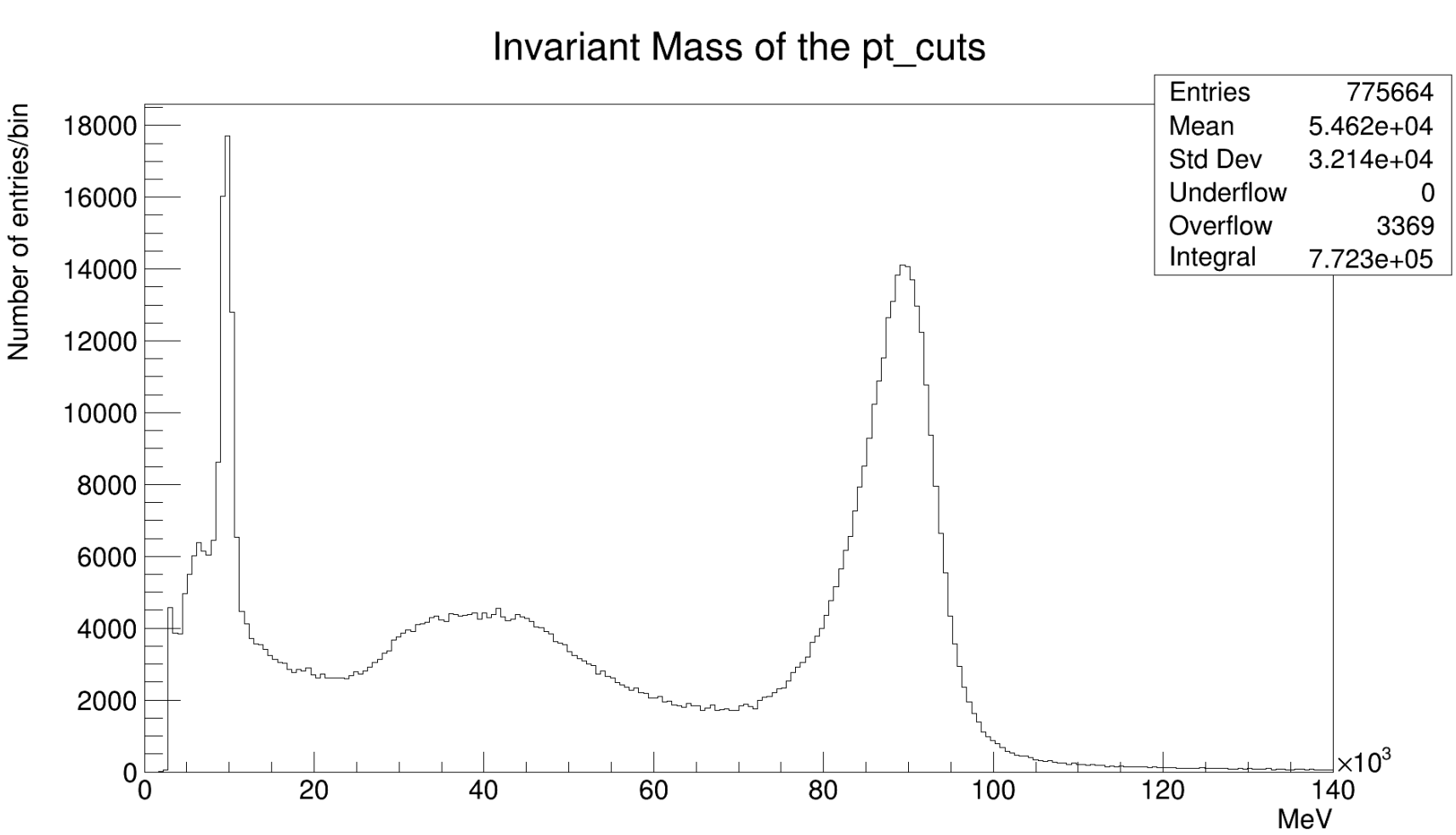



Transverse Momentum pt0
Number of entries/bin
×10^3
Entries 5473829
Mean 4.812e+04
Std Dev 1.649e+04
Underflow 0
Overflow 4.748e+04
Integral 5.426e+06
lep_pt[0]
×10^3

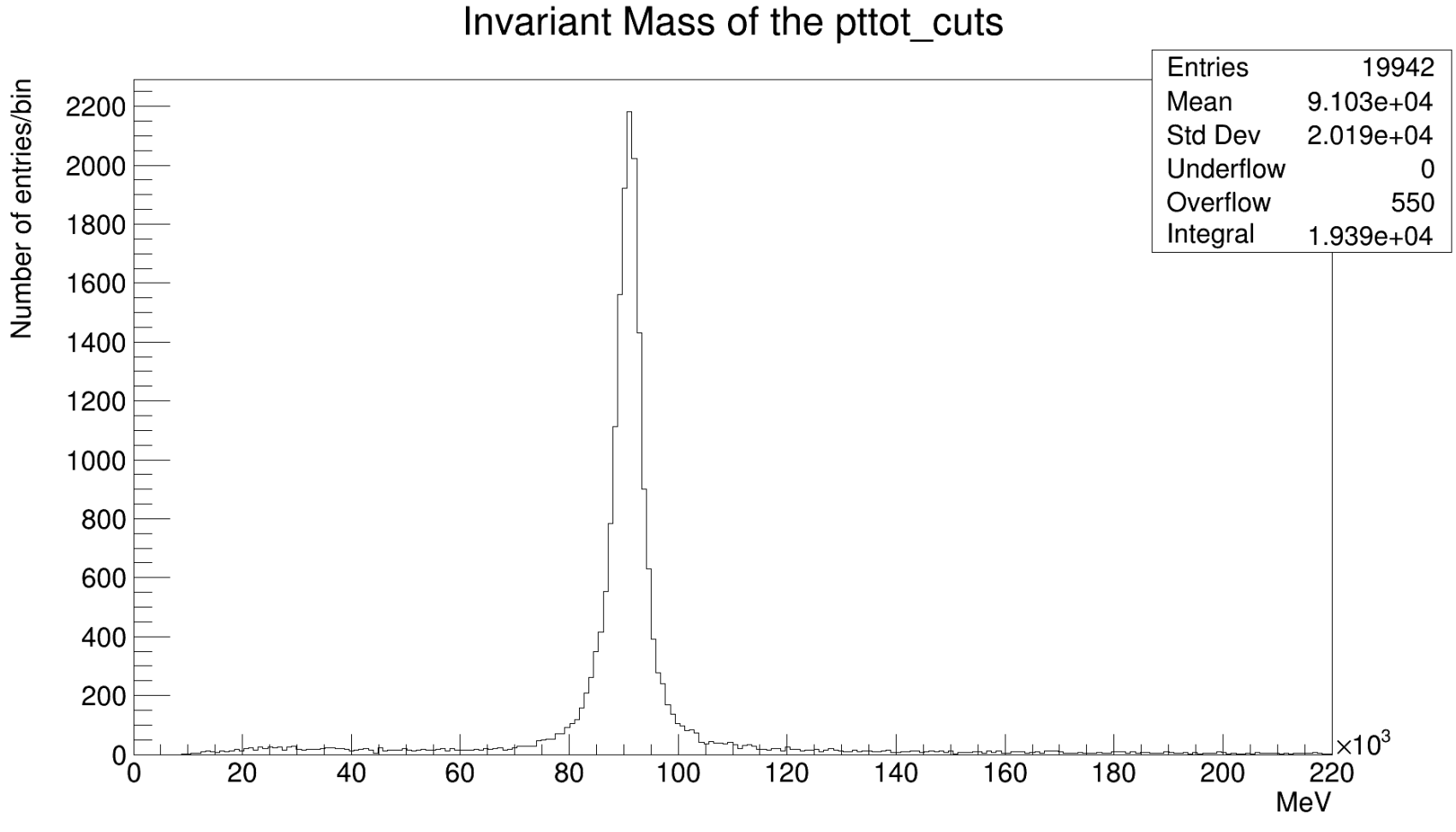

Transverse Momentum pt1
Number of entries/bin
×10^3
Entries 5473829
Mean 3.27e+04
Std Dev 1.187e+04
Underflow 0
Overflow 3591
Integral 5.47e+06
lep_pt[1]
×10^3

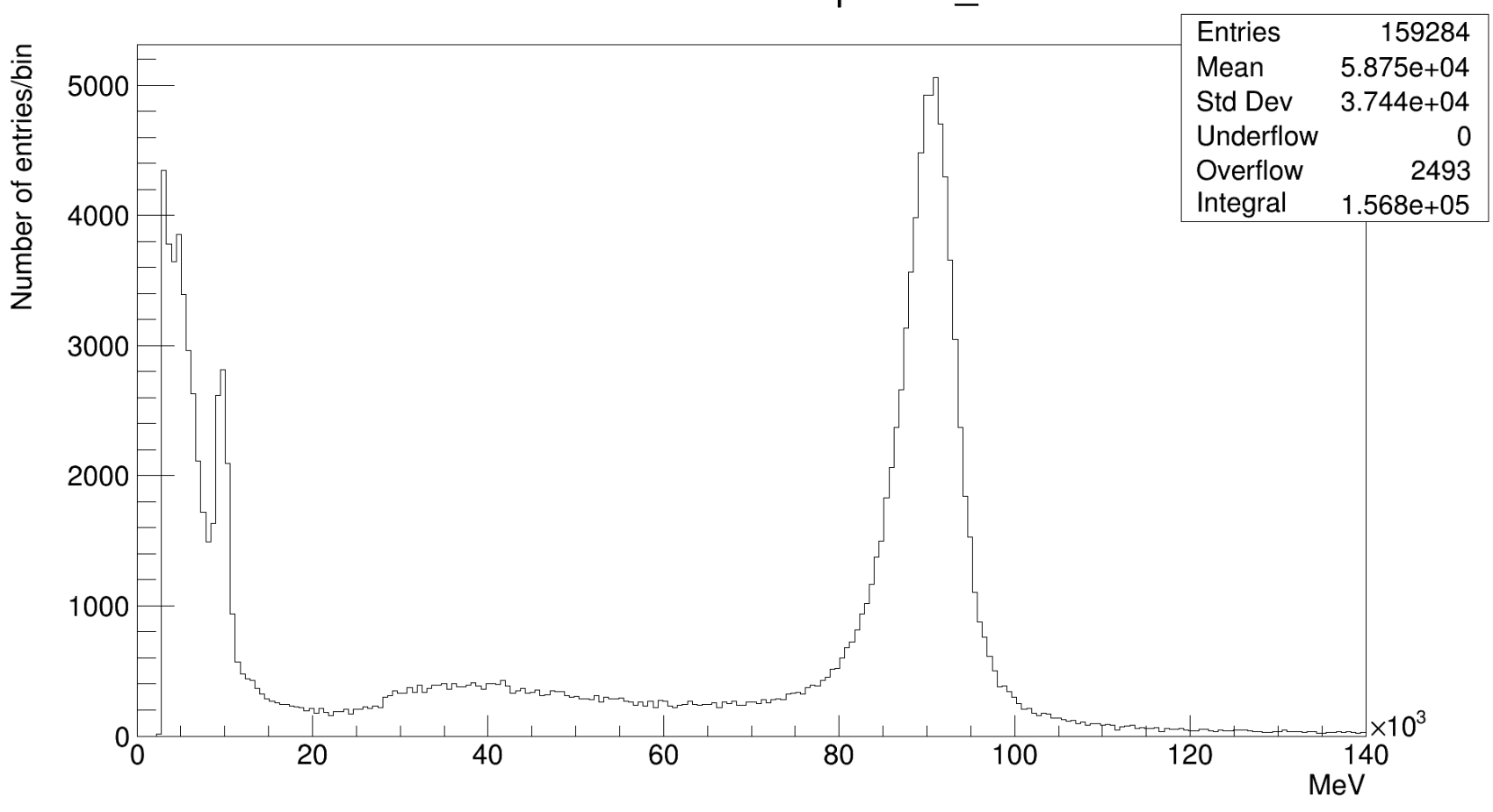

Invariant Mass of the System
Number of entries/bin
×10^3
Entries 4321052
Mean 9.001e+04
Std Dev 6820
Underflow 0
Overflow 5.01e+04
Integral 4.271e+06
mass(MeV)
×10^3

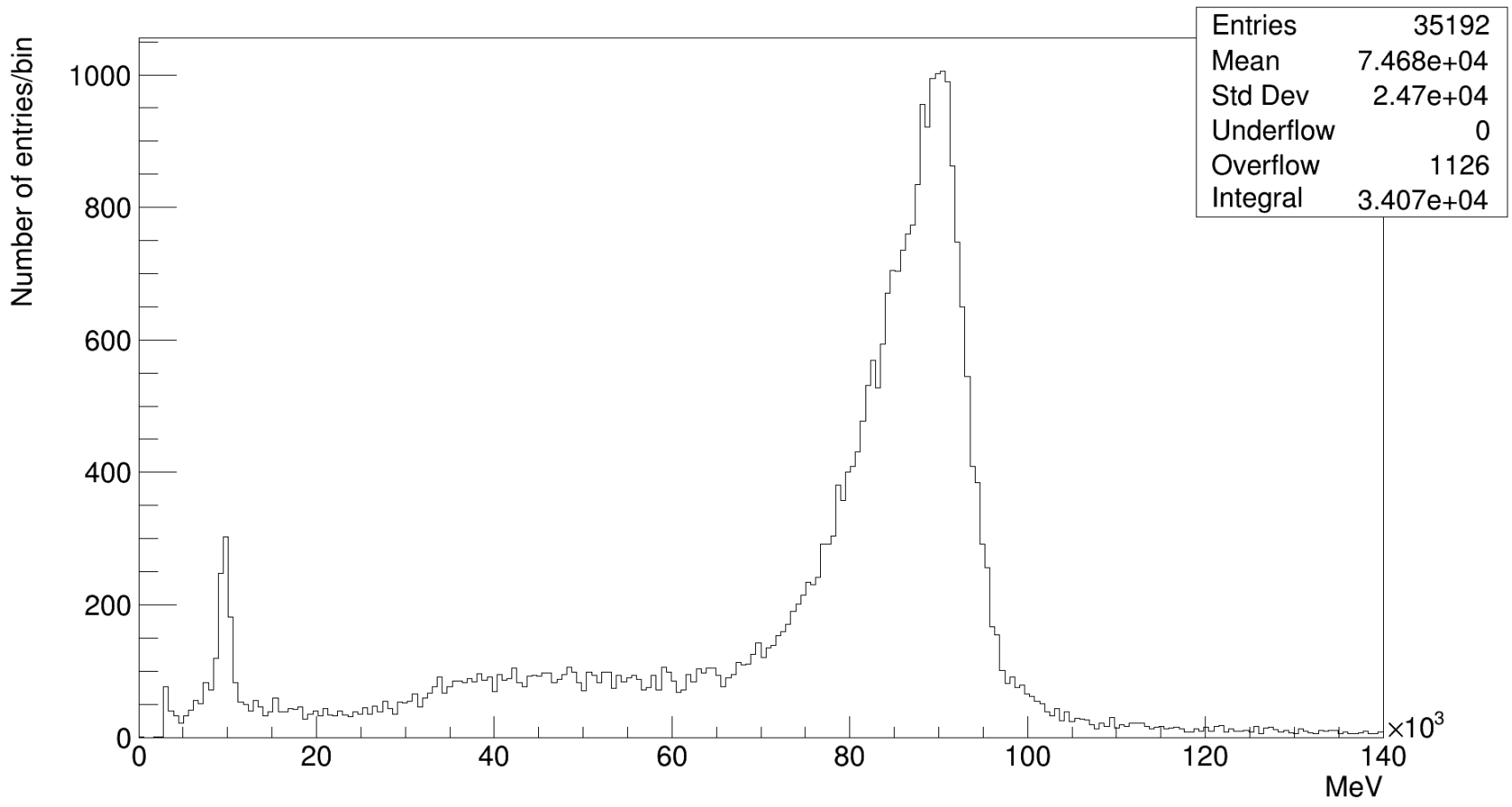




Invariant Mass of the pttot_cuts







In [3]:

```
# ESSENTIAL IMPORTS - This cell only needs to be run ONCE for every new kernel
import ROOT as r
from ROOT import *
import os
import sys
sys.path.insert(0, "backend") # allow code to be imported from subdirectory
from ShowHistogram import plotHistogram,histogramsFromFile
from ShowStackedPlot import plotStackedHist
```

It is important to note that any canvas' you make showing histograms can be saved as images on your computer by right-clicking the image and selecting "Save image as..."

Section 3: Making stacked plots

This section contains just a single cell which contains code do the following:

1. Open up multiple example .root files and retrieve histograms
2. Make a stacked histogram through the use of THStack, and add the read-in histograms to it
3. Add the histogram referring to ATLAS data so that it can be compared with the MC contributions

The function plotStackedHistogram(...) can be found in the 'backend' folder in ShowStackedPlot.py. For useful information on THStack, please click here.

This is a 'stacked' plot - the histograms of MC contributions are 'stacked' on top of each other. This means that the order in which you add each histogram to the stack will affect each contribution's position in the stacked plot.

Stacked plots can be used to identify the size of different contributions to data.

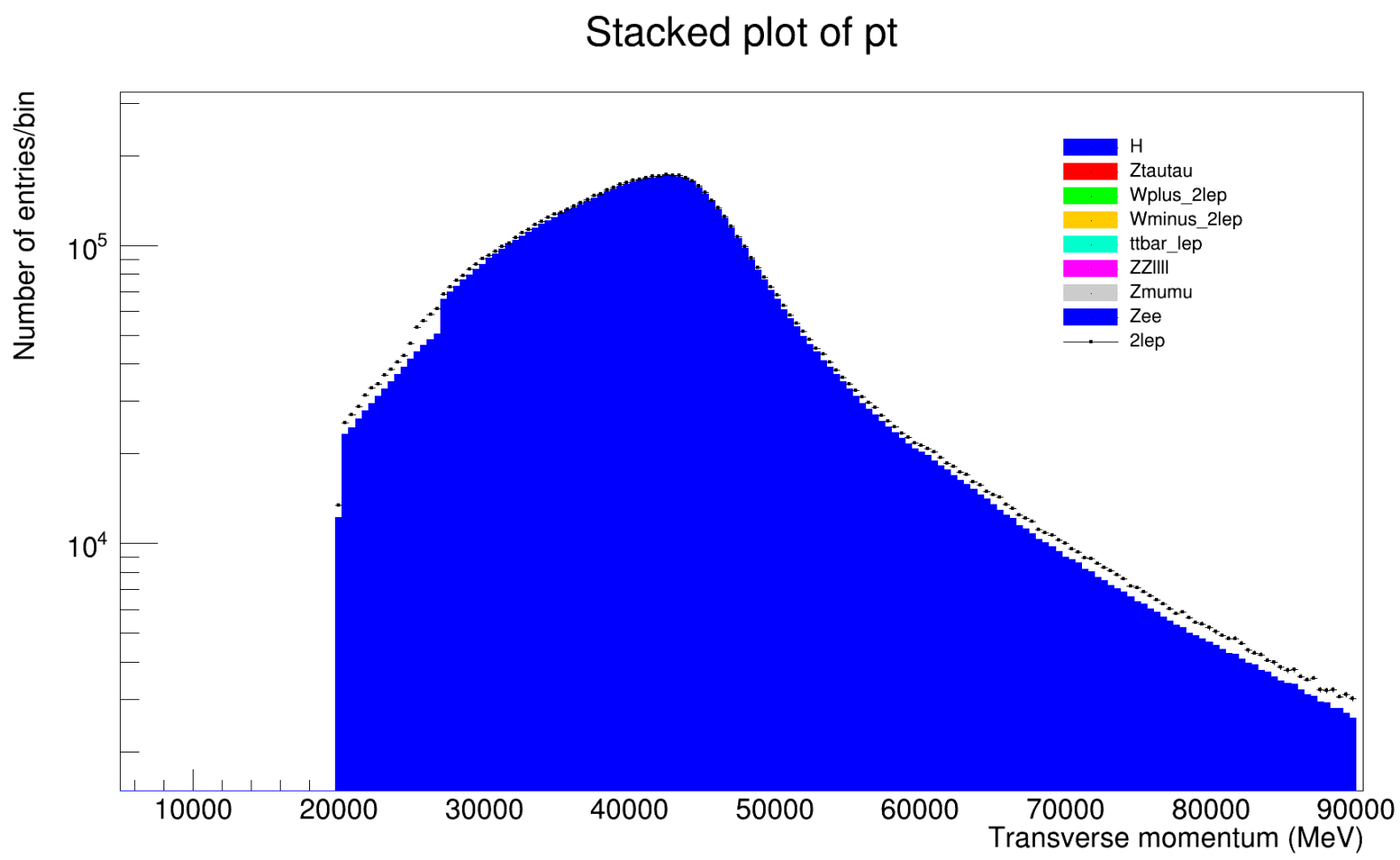
Beginning with an example stacked plot of mean lepton transverse momentum per event: Run *RunAnalysis.py* using the cell in Section 1 of this notebook, using the string codes "Zee", "Zmumu", "Ztautau" and "2lep", and use fast mode by typing "yes". Then run the cell below to generate the stacked plot. From this plot, you should be able to interpret that the combination of the processes of a Z boson decaying to different same-flavour lepton pairs contributes to most of the ATLAS data set.

These stacked plots can be applied to other histograms, such as those investigating invariant mass.

Make sure you run the 'Essential Imports' cell in Section 2 before running this code.

In [10]:

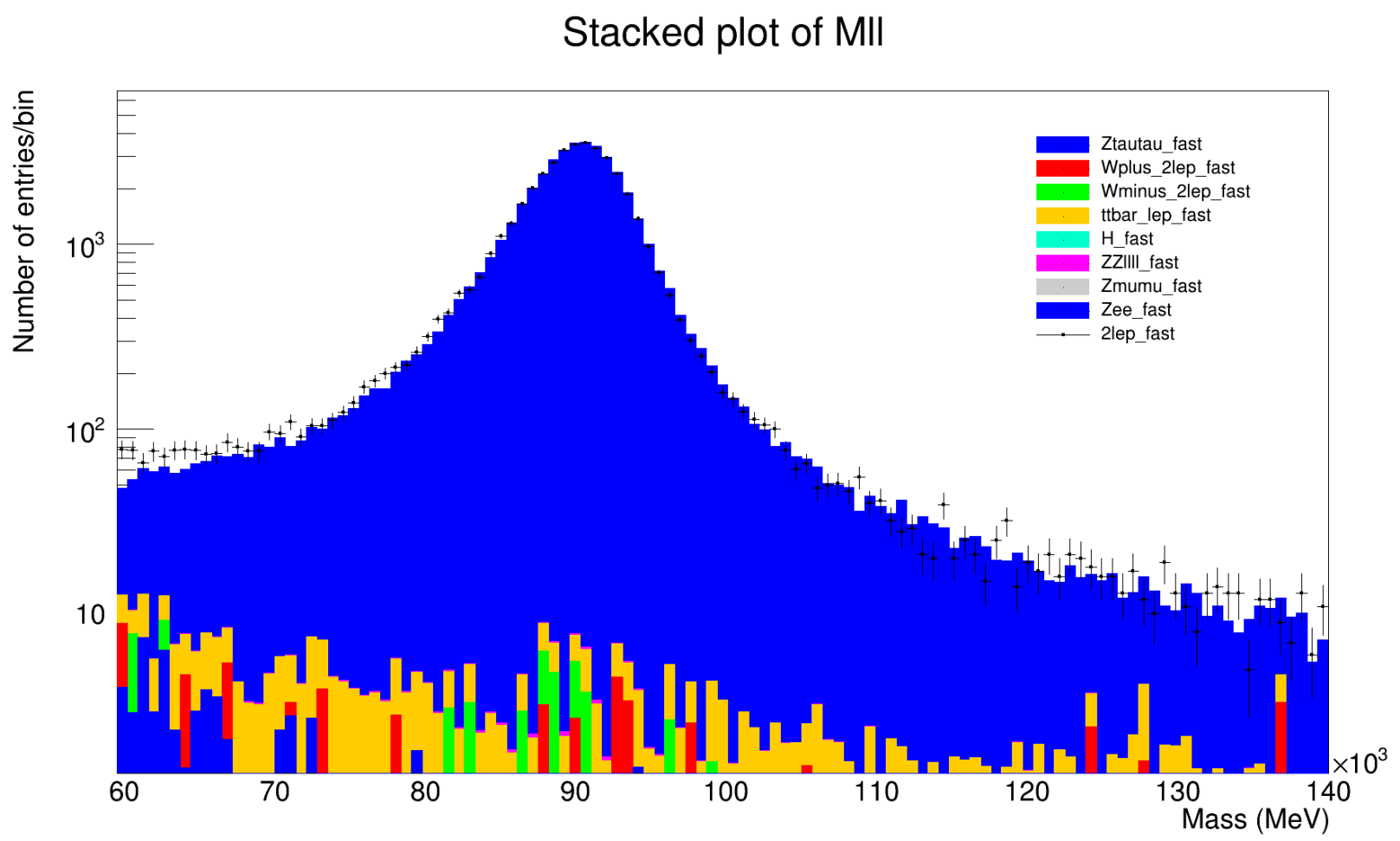
```
# Filenames for plotting
#dataFilename = "2lep_fast.root"
#StackedFilenames = ["Ztautau_fast.root",
#                    "Wplus_2lep_fast.root","Wminus_2lep_fast.root","ttbar_lep_fast.root",
#                    "H_fast.root", "ZZllll_fast.root","Zmumu_fast.root","Zee_fast.root"]

dataFilename = "2lep.root"
StackedFilenames = ["H.root", "Ztautau.root",
                    "Wplus_2lep.root","Wminus_2lep.root","ttbar_lep.root",
                    "ZZllll.root","Zmumu.root","Zee.root"]


#To use custom labels/colours, uncomment lines below and edit plotStackedHist(...) inputs
#nameArray = ["Data","Zee MC","Zmumu MC","Ztautau MC"]
#colourArray = [r.kBlack,r.kBlue,r.kRed]

#Plot Stacked Histogram
canvas,_ = plotStackedHist(dataName = dataFilename,
                           MCNames = StackedFilenames,
                           hist_id = "h_lep_pt",
                           x_min = 5e3, x_max = 150e3,
                           title="Stacked plot of pt",
                           x_label = "Transverse momentum (MeV)",
                           y_label = "Number of entries/bin",
                           log_y = True,
                           nameArray = None,
                           colourArray = None)

# Note: range of x_max and x_min cannot be larger than range used when
#       data and MC .root files were created
# Note: plotStackedHist(...) has compulsory arguements: dataName, MCNames and hist_id
#       The following arguments are optional: x_min,x_max,title,x_label,y_label,
#                                             nameArray,showLegend, colourArray
#                                             legend_text_size and log_y
```



In [5]:

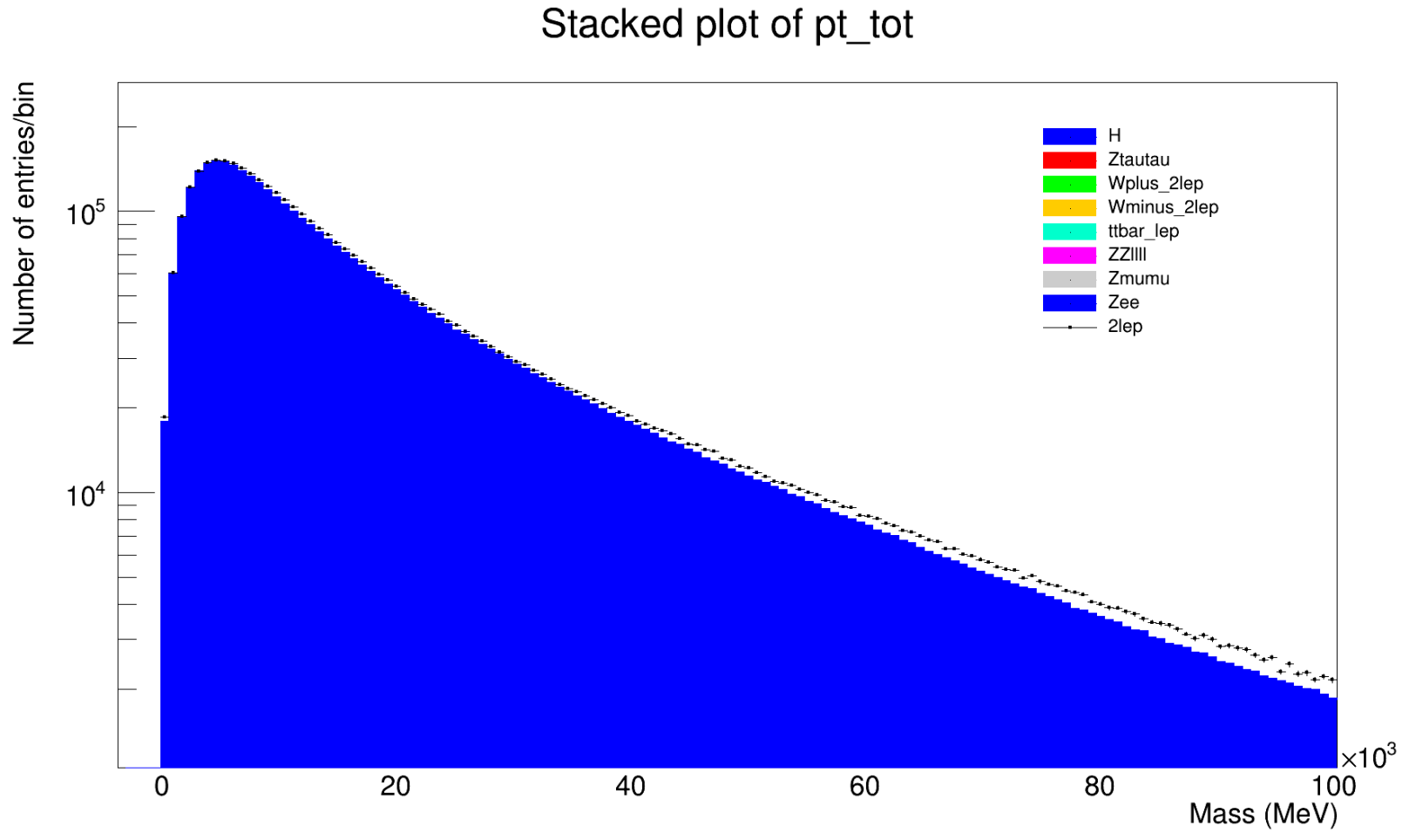
```
# Filenames for plotting
dataFilename = "2lep_fast.root"
StackedFilenames = ["Ztautau_fast.root",
                    "Wplus_2lep_fast.root","Wminus_2lep_fast.root","ttbar_lep_fast.root",
                    "H_fast.root", "ZZllll_fast.root","Zmumu_fast.root","Zee_fast.root"]


#To use custom labels/colours, uncomment lines below and edit plotStackedHist(...) inputs
#nameArray = ["Data","Zee MC","Zmumu MC","Ztautau MC"]
#colourArray = [r.kBlack,r.kBlue,r.kRed]

canvas,_ = plotStackedHist(dataName = dataFilename,
                           MCNames = StackedFilenames,
                           hist_id = "h_lep_m",
                           x_min = 60e3, x_max = 140e3,
                           title="Stacked plot of Mll",
                           x_label = "Mass (MeV)",
```

```
                    y_label = "Number of entries/bin",
                    log_y = True,
                    nameArray = None,
                    colourArray = None)
```



In [11]:

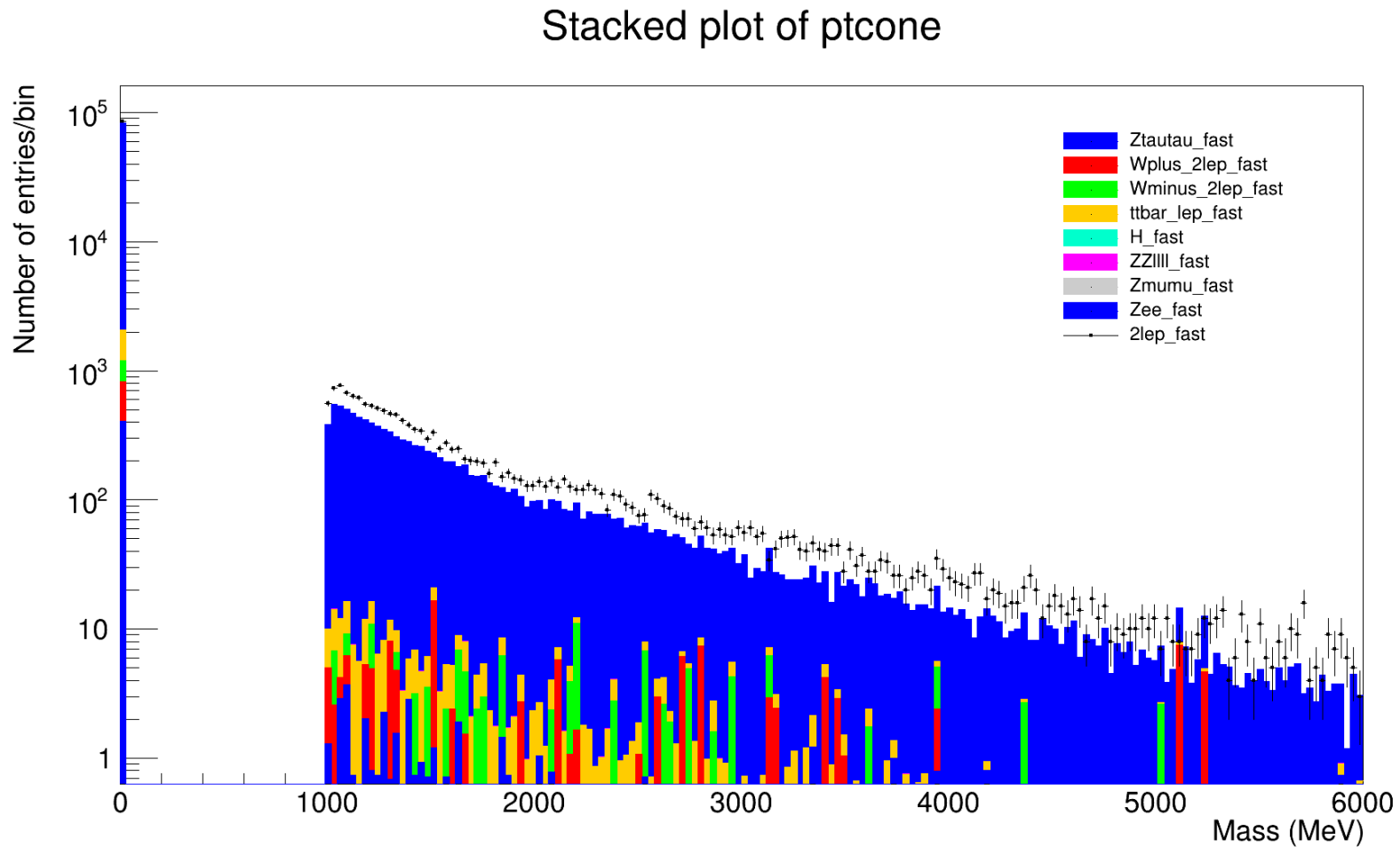
```
# Filenames for plotting
dataFilename = "2lep.root"
StackedFilenames = ["H.root", "Ztautau.root",
                    "Wplus_2lep.root","Wminus_2lep.root","ttbar_lep.root",
                    "ZZllll.root","Zmumu.root","Zee.root"]

canvas,_ = plotStackedHist(dataName = dataFilename,
                           MCNames = StackedFilenames,
                           hist_id = "h_lep_pt_tot",
                           x_min = -40e3, x_max = 100e3,
                           title="Stacked plot of pt_tot",
                           x_label = "Mass (MeV)",
                           y_label = "Number of entries/bin",
                           log_y = True,
                           nameArray = None,
                           colourArray = None)
```



In [7]:

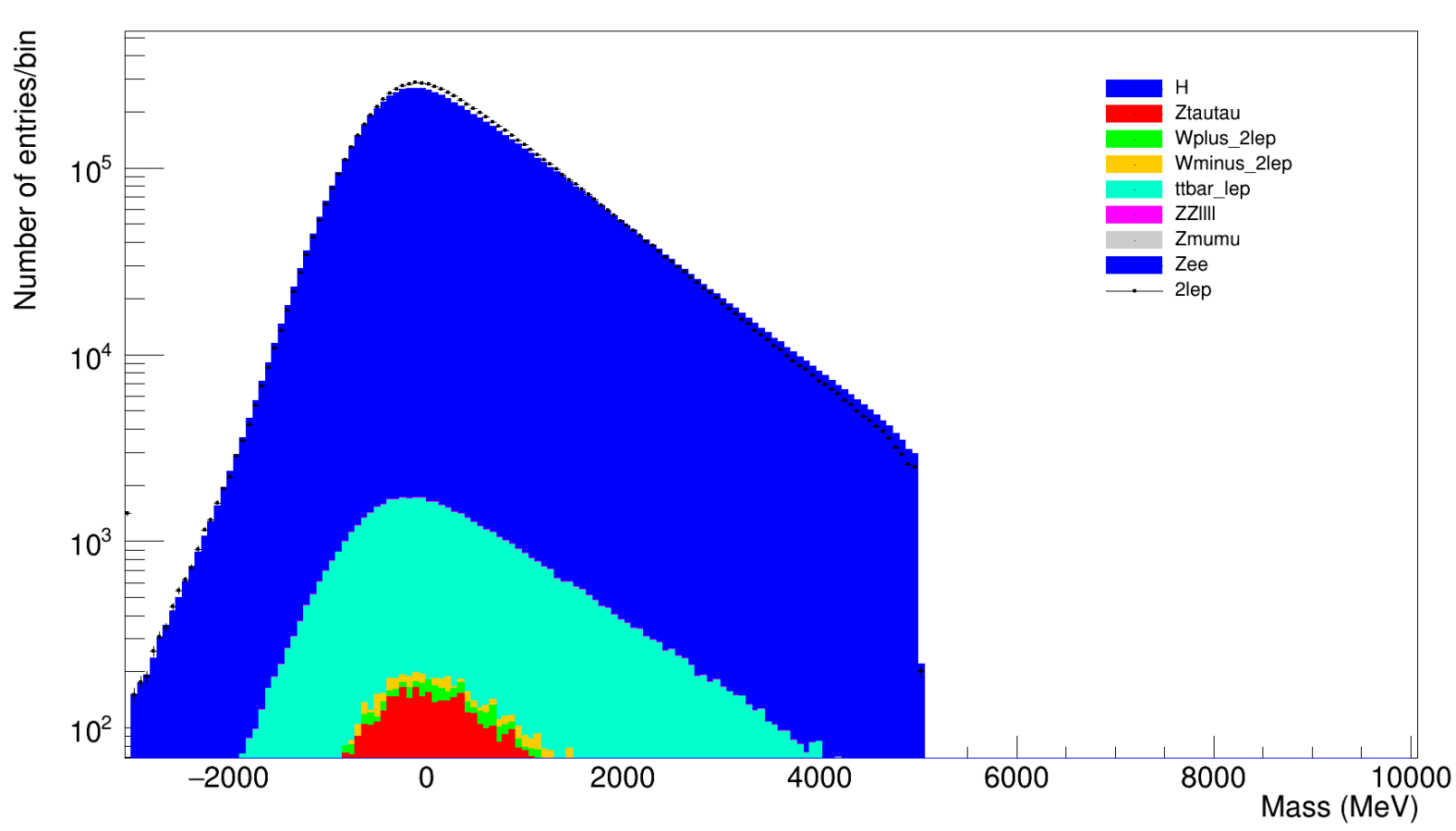
```
# Filenames for plotting
dataFilename = "2lep_fast.root"
StackedFilenames = ["Ztautau_fast.root",
                    "Wplus_2lep_fast.root","Wminus_2lep_fast.root","ttbar_lep_fast.root",
                    "H_fast.root", "ZZllll_fast.root","Zmumu_fast.root","Zee_fast.root"]

canvas,_ = plotStackedHist(dataName = dataFilename,
                           MCNames = StackedFilenames,
                           hist_id = "h_lep_ptcone",
```

```
x_min = 60e3, x_max = 140e3,
title="Stacked plot of ptcone",
x_label = "Mass (MeV)",
y_label = "Number of entries/bin",
log_y = True,
nameArray = None,
colourArray = None)
```



In [12]:

```
# Filenames for plotting
dataFilename = "2lep.root"
StackedFilenames = ["H.root", "Ztautau.root",
                    "Wplus_2lep.root","Wminus_2lep.root","ttbar_lep.root",
                    "ZZllll.root","Zmumu.root","Zee.root"]


canvas,_ = plotStackedHist(dataName = dataFilename,
                           MCNames = StackedFilenames,
                           hist_id = "h_lep_etcone",
                           x_min = -40e3, x_max = 100e3,
                           title="Stacked plot of etcone",
                           x_label = "Mass (MeV)",
                           y_label = "Number of entries/bin",
                           log_y = True,
                           nameArray = None,
                           colourArray = None)
```



Appendix

Frequenctly asked questions

What do I do if I get a pop-up message saying I have lost connection?

If this happens to you and you are connected to the internet, first copy all cells to a text editor 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 of the lab script, and a message should display saying "connection reset...". Starting from the new command line that should appear, please complete steps 5-9 again of Section 5.3 of the lab script. 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.

Why did my cell output change the second time I ran it, when it shouldn't have changed?

This is most likely due to at least one of the following:

1. Any files you have read in may not be the same as they were before. To avoid this you need to rename your .root files after every run of *RunAnalysis.py* to prevent them from being overwritten.
2. Any variables that are used in the cell may have been redefined outside of this cell. To avoid this, do not redefine variables in different cells unless necessary, e.g do not use 'canvas' for all your TCanvas variable names.

Why can't I read in the new histograms I made?

This is most likely because you did not restart the kernel before running *RunAnalysis.py*. This would mean the previous version of *Analysis.py* is being used by *RunAnalysis.py*, in which the histograms you are trying the access did not exist. Always make sure you save *Analysis.py* if you have made changes to it, and then restart your kernel before running the script.

Why are my histograms shown in a scrollable output box?

To expand the output box, move your mouse to the left side of the output box. a grey rectangle should appear. Clicking this rectangle will expand the box to put all histograms inline with the code. Clicking the rectangle again will recollapse the box. BEWARE: double clicking the rectangle will hide the output box.

In [ ]: