# Root Histograms

# Last Time and Today

- Last time we started to talk about ROOT
- Today we will start to use it, beginning with Hisograms
  - What are histograms
  - What types of histogram are there.
  - Booking, Filling and Plotting Histograms.

# What is a histogram?

- A histogram is a way of collecting data and presenting it.
- Data would be in the form of a collection of data points.
  - Typically from events in a detector
    - energy
    - position
    - number of particles etc

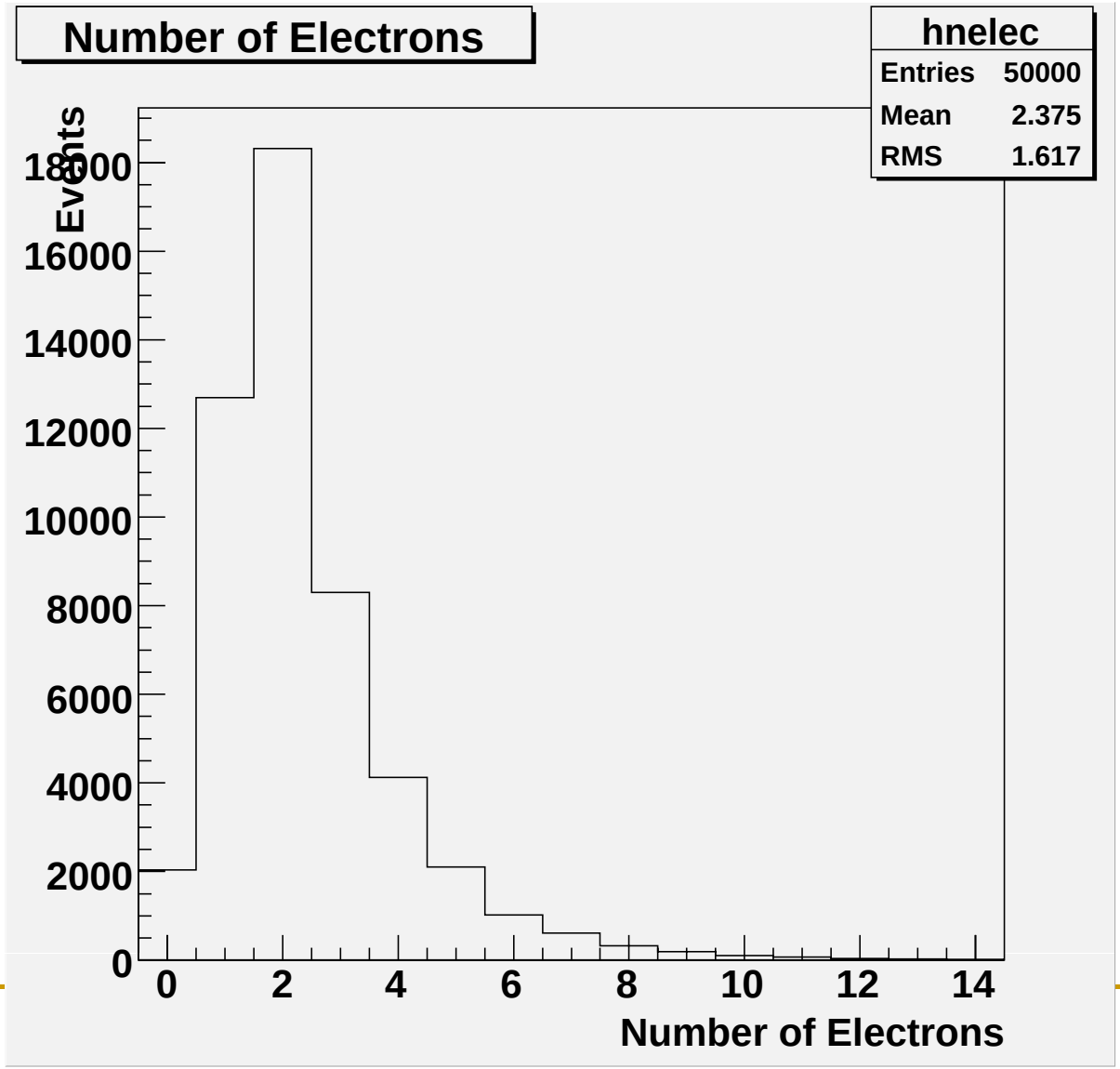
# What is a histogram?

- Given these data a histogram shows the number of events that fall into certain parameter ranges.
  - Each of these ranges are called bins
  - Typically these will be uniform
    - Not always
- Histograms can then be used for further analysis
  - To understand the underlying parent distribution.

# Our primary example

- $Z^0$ decays into electron in Atlas.
  - Primary electrons.
    - Direct decay of the $Z^0$
    - Invariant mass should be the $Z^0$ mass.
  - Secondary electrons.
    - Indirect electrons from decays of other particles
      - Heavy quark decay (b, c and s)
      - Tau decay
  - Mis-identified events
    - Events that aren't actually electrons.

# An example 1D histogram



# How to make this.

- First you need to declare your histogram object.
- The exact object depends on what type of variable you will be filling the histogram with.
  - Lets assume double precision numbers.
- Use a TH1F object.
- TH1F *hist = new TH1F(“hist”,”title”,nbins,xlow,xhigh);
  - title is the title of your histogram :  charaters
  - nbins is an integer with the number of bins
  - xlow is a double precision number with the low edge of the histogram
  - xhigh is also double precision and has the high edge of the histogram.

# Filling your histogram.

- You need to loop through your tree to get the number of electrons in each event.
- Use the method Fill for TH1F (and all other histogram types).

```
for (Int_t indx=0;indx<inTree->GetEntries();indx++){
    inTree->GetEntry(indx);
    hist->Fill(ElecNum);
}
```

# Drawing your histogram.

- You can now draw your histogram using the Draw method.
- hist->Draw();
- There are a number of arguments to Draw that can be used to improve the look of the histogram that we can consider using.
  - hist->Draw(“e1”);

# Some basic options

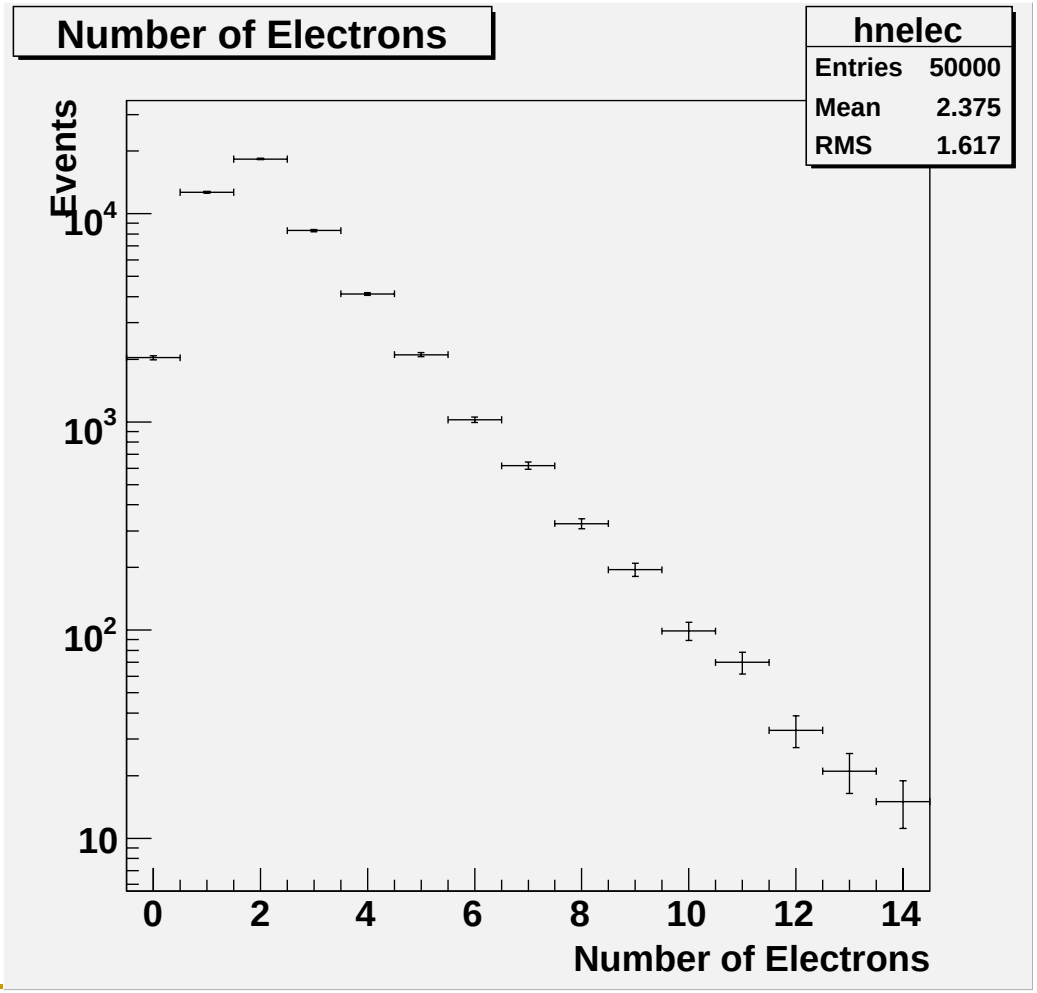
- E0-4 (I like E1)
  - Draw error bars on the points.
- SAME
  - Draw the histogram on the canvas without replacing what is already there.
  - Use to plot one histogram on top of another.
- B
  - Bar Chart
- P
  - Draw a marker at each point.
- L, LF2
  - Draw a line and fill an area.

# Setting your titles

- All histograms should have titles on the axis, and probably a title as well.
- To set the title on the axis you need to get the axis object from the histogram and set the title.
  - You can do this in one line
  - hist->GetXaxis()->SetTitle("x axis title");
  - hist->GetYaxis()->SetTitle("y axis title");
- The histogram title is set in the constructor
  - You can reset it with the SetTitle method
  - hist->SetTitle("histogram title");

# Using a log scale.

- Often you will want to use a log scale
  - Particularly on the y axis.
- This is a function of the pad or canvas.
  - GPad->SetLogy(1);



# Getting data out of a histogram

- A number of methods exist to extract data from a histogram:
  - hist->GetEntries();
  - hist->GetMean();
  - hist->GetRMS();
  - hist->GetIntegral();
- You can also look at the stats box
  - On by default.
  - Turn off with:
    - hist->SetStats(kFALSE);
  - Control contents with:
    - gStyle->SetOptStat(num);
    - num = 0001111 (say)
    - title, entries, mean, rms,
      - See the manual.

# Normalizing a histogram.

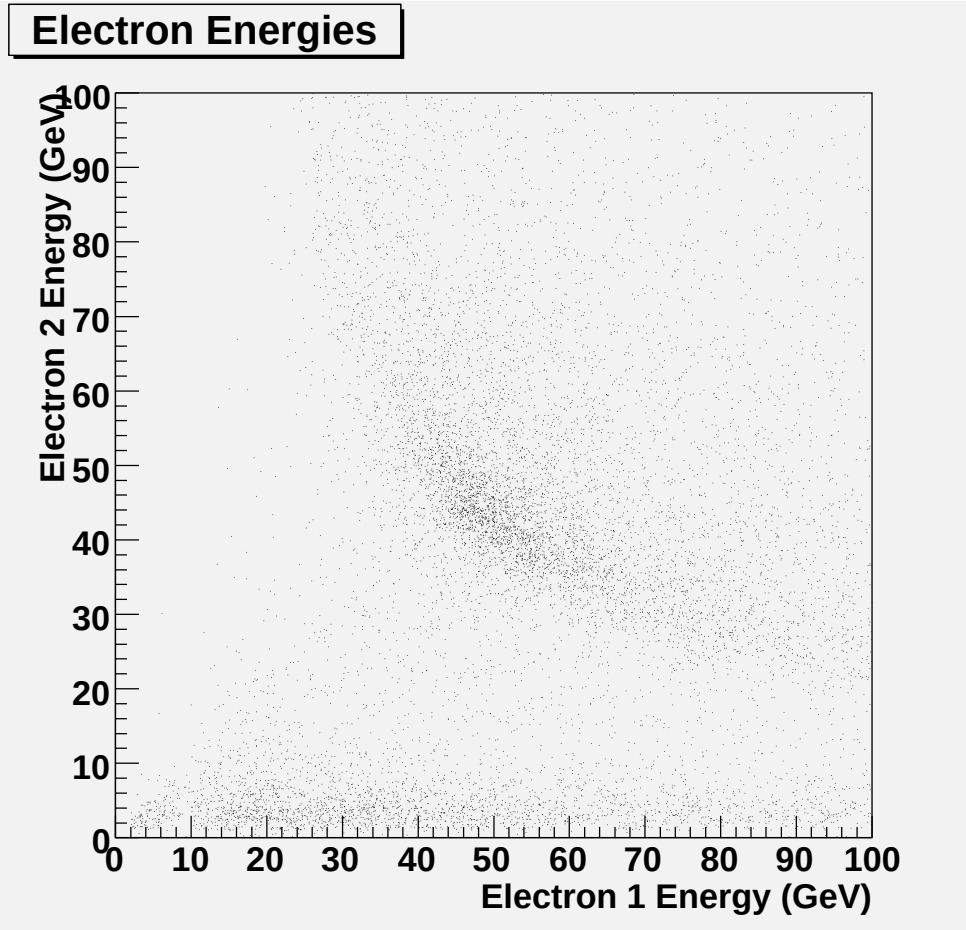
- Often you will want to normalize a histogram.
  - Useful when plotting one histogram on top of another.
  - Can be useful when turning a Monte Carlo data set into a probability distribution.

```
Double_t scale = norm/hist->Integral();

hist->Scale(scale);
```

# Two dimensional histograms.

- You can also make two dimensional histograms.
- Here I plot the energy of the two electrons against each other, making a cut such that only two electrons were seen.

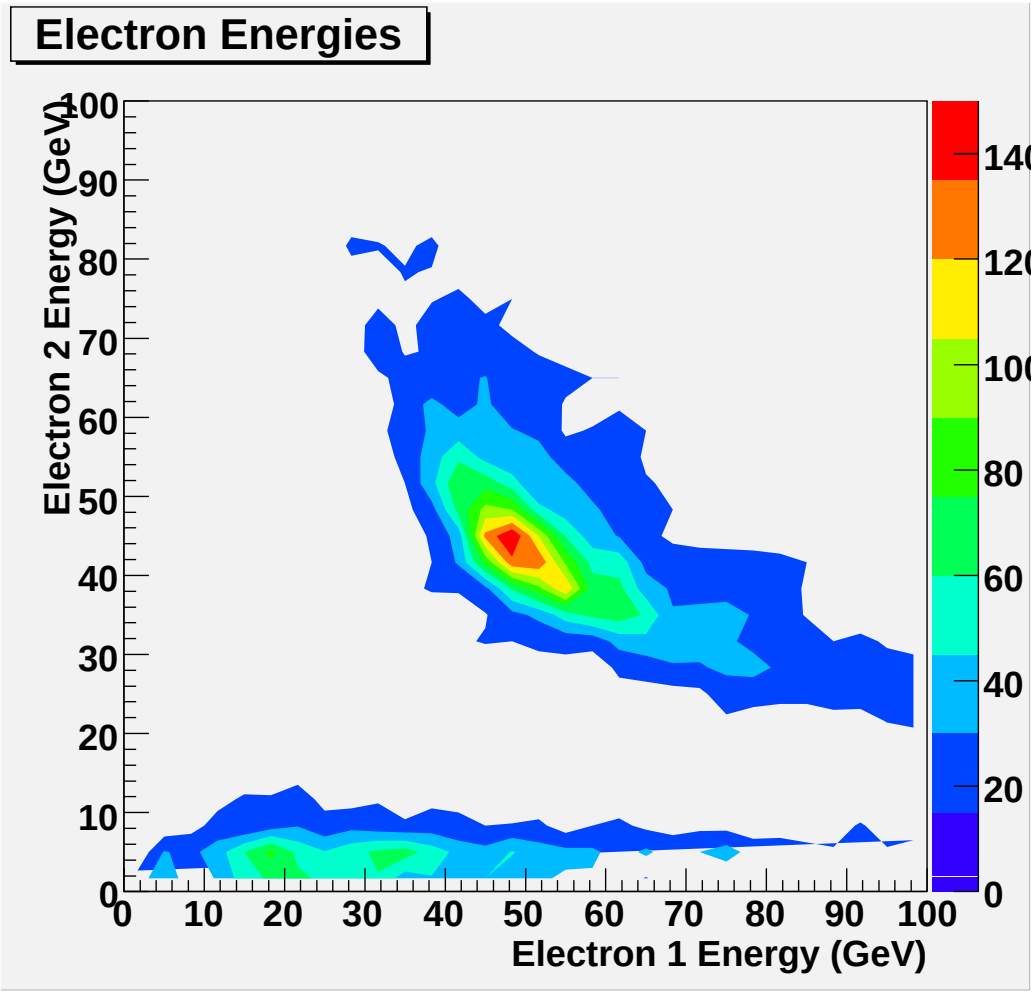


# How to make this.

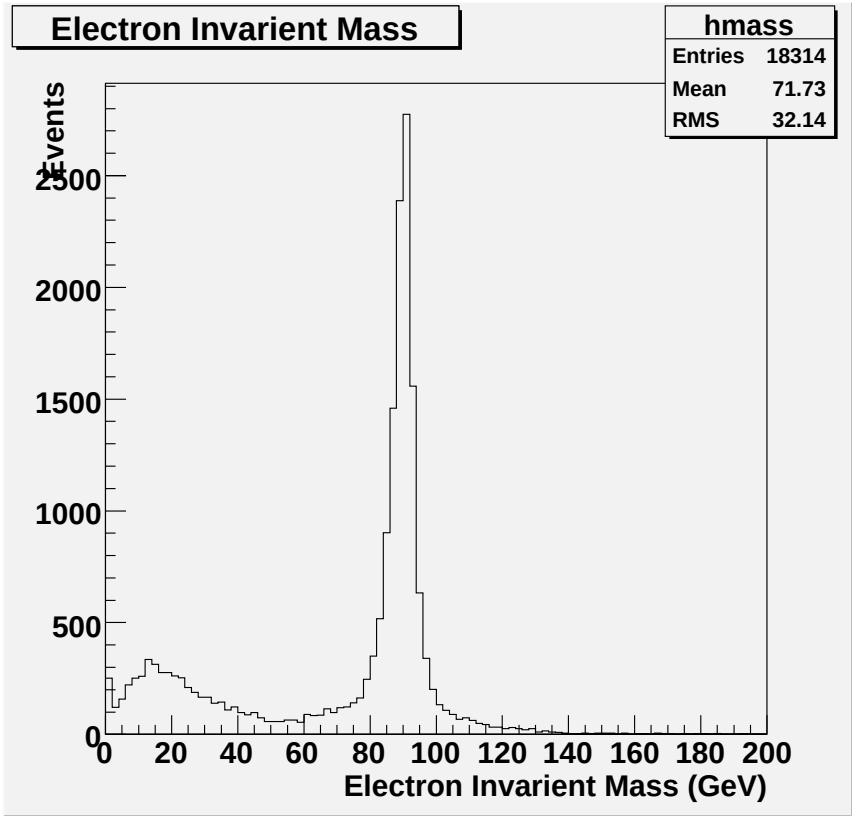
- We  need a 2D histogram object TH2F.
  - TH2F *h2d = new TH2F(“h2d”,”title”,nxbins,xlow, xhigh,nybins,ylow,yhigh)
- Fill works as you’d expect
  - h2d->Fill(x,y);
- Draw also, but you have more options.
  - h2d->Draw();
  - Contour, lego and surface plots are available.
- Titles, and data as before.

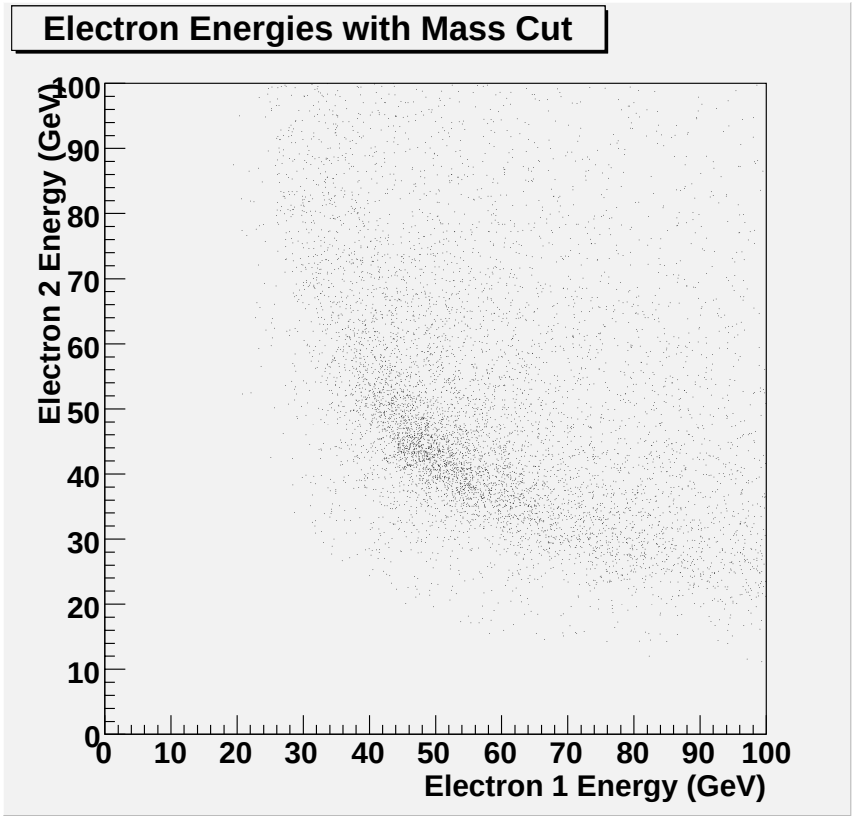
# Contour plots

- One common way to present 2D data is with a contour plot.
- Use the cont option in the Draw method
  - h2d->Draw("cont");
- Use contz to get the colour key.
- Try and avoid too many bins as statistical fluctuations can play havoc with contour plots.



# Cleaning up the data.

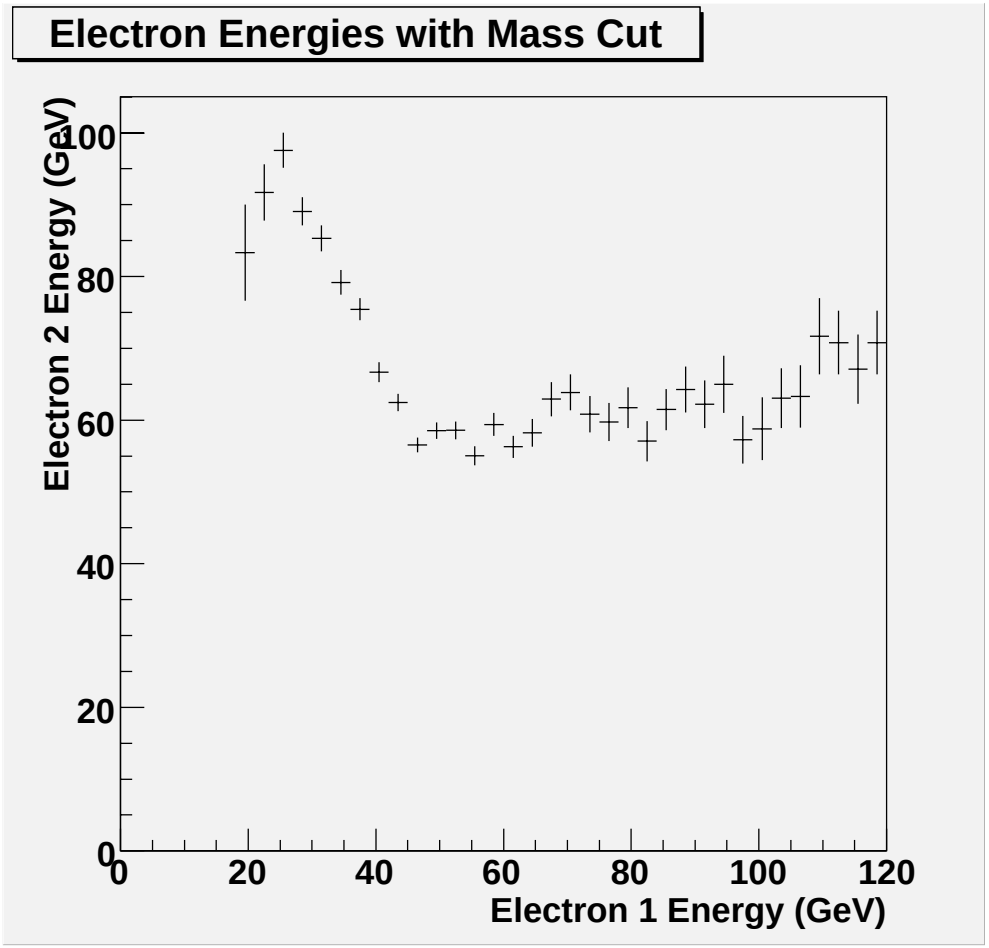



Cut on the invariant mass of the two electrons.
60GeV<M<120GeV

# Profile histograms.

- A profile histogram is another way of presenting the data.
- In each x bin the data is collected.
- The mean and error in the mean of the y variable is calculated.
  - This is the contents of the bin.



# Creating a profile histogram.

- Use a TProfile object
  - TProfile *hprof = new TProfile(“hprof”,”title”,nxbins, xlow,xhigh,ylow,yhigh)
  - Note that nybins is gone from TH2D
  - ylow and yhigh are now cuts on y
- Fill like a regular 2d histogram
  - hprof->Fill(x,y);
- Draw like a 1D histogram
  - Errors are on by default.

# Error bars in a profile histogram.

- There are options in the TProfile constructor to control the error bars.
  - TProfile *hprof = new TProfile(“hprof”,”title”,nxbins, xlow,xhigh,ylow,yhigh,option)
- The default sets the errors to the erorr in the mean
  - RMS/√n
- The spread option (option = “s”) give the spread directly.
  - RMS
- Which one you use will depend on what you are trying to show.

# Exercises

- Take a look at the macros used to generate the plots in this talk.
  - The base was produced with the MakeClass method of TTree which will be covered later, as will some of the other actions.
- Make your own histograms, to show the same distributions as I have for electrons.
  - Does the invariant mass vary from particle to particle?