ICTP-NCP School on LHC Physics I (smr 2632)

Prepared by: Taimoor Khurshid & Muhammad Ahmad

Course material: From CMSDAS

Introduction

In this set of exercises we will do four exercises. We will analyze the PATtuple. We will use different options of analyzing the PATtuple, namely,

EDAnalyzer,
FWLite executable,
FWLite Macro,
FWLite PyROOT.

We will basically re-make the Zpeak and few other histograms and store them in an output root file. In the exercise in the end we will try to fit with a

Gaussian function,
Breit-Wigner function.

Exercise 12 - Analyzing PATtuple with an EDAnalyzer

In this exercise we will analyze the PATuples created in Third Set of Exercise using an EDAnalyzer. In these PATtuples, if you recall, we saved only the Muons and Electrons. So do not look for Jets or Photons or other objects as they were simply not saved. We will use a python config file and an EDAnalyzer (a .cc file) to make a Z mass peak.

First we will add the PhysicsTools/PatExamples package as follows to OURWORKINGAREA/CMSSW_5_3_13/src. The PatExamples package has lot of examples for a user to try. However, we will add our own code and config file to it and then compile. To add this package, do this:

```
git cms-addpkg PhysicsTools/PatExamples
```

The python config file "*analyzePatBasics_cfg.py*" should be in YOURWORKINGAREA/CMSSW_5_3_13/src/PhysicsTools/PatExamples/test/ directory.

The EDAnalyzer named "*PatBasicAnalyzer.cc*" should be in your YOURWORKINGAREA/CMSSW_5_3_13/src/PhysicsTools/PatExamples/plugins/ directory.

First copy above mentioned two files from /scratch/t/taimoor/LHC_SCHOOL_4/course_material/

Then we will compile the code that you just saved. To do this, do:

```
scram b
```

When you do the above you should see among the output a line like the one below while compilation is going on, otherwise it is not compiled and you may end up assuming so

```
>> Compiling
/scratch/…/…/YOURWORKINGAREA/CMSSW_5_3_13/src/PhysicsTools/PatExamples/src/My
ZPeakAnalyzer.cc
```

After successful compilation, you must run the config file as follows:

```
cmsRun MyZPeak_cfg.py
```

Successful running of the above config file will produce an output file myZPeakCRAB.root.

Note: the output file myZPeakCRAB.root has several histograms, besides the ZPeak called mumuMass, like muonMult, muonEta, muonPhi, muonPt and similarly for electrons.

QUESTION 12 - What is the number of entries in the mumuMass plot ?

Before you move further, make sure you have the above ROOT file called myZPeakCRAB.root in your YOURWORKINGAREA/CMSSW_5_3_13/src. If you have not managed to do so, you can get this ROOT file from /scratch/t/taimoor/LHC_SCHOOL_4/course_material/myZPeakCRAB.root and move ahead.

Exercise 13 - Analyzing PATtuple with an FWLite executable

In this exercise we will make the same ROOT file myZPeakCRAB.root as in Exercise 12 but we call it myZPeakCRAB_fwlite.root so that you do not end of overwriting the file previously made in Exercise 12.

First make sure you have checked the following two packages:

```
git cms-addpkg PhysicsTools/FWLite
git cms-addpkg PhysicsTools/UtilAlgos
```

Then copy the following two files.

1. Replace the YOURWORKINGAREA/CMSSW_5_3_13/src/PhysicsTools/FWLite/bin/FWLiteWithPythonConfig.cc with this FWLiteWithPythonConfig.cc, from /scratch/t/taimoor/LHC_SCHOOL_4/course_material.

2. Copy parameters.py in YOURWORKINGAREA/CMSSW_5_3_13/src/ from the same location.

Then we will compile the code that you just saved. To do this, do:

```
rehash
scram b
```

When you do the above you should see among the output a line like the one below while compilation is going on, otherwise it is not compiled and you may end up assuming so

```
Compiling
/scratch/…/…/YOURWORKINGAREA/CMSSW_5_3_13/src/PhysicsTools/FWLite/bin/FWLiteW
ithPythonConfig.cc
```

After successful compilation, you must run the config file as follows:

```
FWLiteWithPythonConfig parameters.py
```

Note: How the parameters defined in parameters.py get input to the executable code FWLiteWithPythonConfig.cc.

A successful running of the FWLite executable FWLiteWithPythonConfig results in an output file called myZPeakCRAB_fwlite.root.

The output ROOT file myZPeakCRAB_fwlite.root is a bit different from myZPeakCRAB.root made in Exercise 12 since we did not make any of the electron histograms. The histograms do have the mumuMass, besides, muonEta, muonPhi and muonPt.

QUESTION 13 - What is the number in entries in the mumuMass obtained in Exercise 12 and 13 ?

Before you move further, make sure you have the above ROOT file called myZPeakCRAB_fwlite.root in your YOURWORKINGAREA/CMSSW_5_3_13/src. If you have not managed to do so, you can get this ROOT file from /scratch/t/taimoor/LHC_SCHOOL_4/course_material/myZPeakCRAB_fwlite.root.

Exercise 14 - Fitting the Z mass peak

The main intention of fitting the Z mass peak is to show how to fit a distribution. To do this exercise will need the root file that you made in Exercises 12 and 13 above. Let us take the root file myZPeakCRAB.root for this exercise. You can use myZPeakCRAB_fwlite.root also but just make sure to have the right name of the ROOT file. Both of these have the histogram mumuMass and that is what is important. The different distribution that we would fit to the Z mass peak are:

- Gaussian

$$G(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

- Relativistic Breit-Wigner

$$B(m; M, \Gamma) = N \cdot \frac{2}{\pi} \cdot \frac{\Gamma^2 M^2}{(m^2 - M^2)^2 + m^4(\Gamma^2/M^2)}$$

- Convolution of relativistic Breit-Wigner plus interference term with a Gaussian

$$P(m) = \int B(m'; M, \Gamma) \cdot G(m - m'; \mu, \sigma) dm'$$

Some general remarks about fitting a Z peak:

To fit a generator-level Z peak a Breit-Wigner fit makes sense. However, reconstructed-level Z peaks have many detector resolutions that smear the Z mass peak. If the detector resolution is relatively poor, then it is usually good enough to fit a gaussian (since the gaussian detector resolution will overwhelm the inherent Briet-Wigner shape of the peak). If the detector resolution is fairly good, then another option is to fit a Breit-Wigner (for the inherent shape) convoluted with a gaussian (to describe the detector effects). This is in the "no-background" case. If you have backgrounds in your sample (Drell-Yan, cosmics, etc...), and you want to do the fit over a large mass range, then another function needs to be included to take care of this - an exponential is commonly used.

Fitting a Gaussian

Before going further, create a rootlogon.C in your home area on lxs, for example /scratch/t/taimoor/. Copy it from /scratch/t/taimoor/LHC_SCHOOL_4/course_material/.

There are several options to fit a Gaussian

Using the inbuilt Gaussian in ROOT

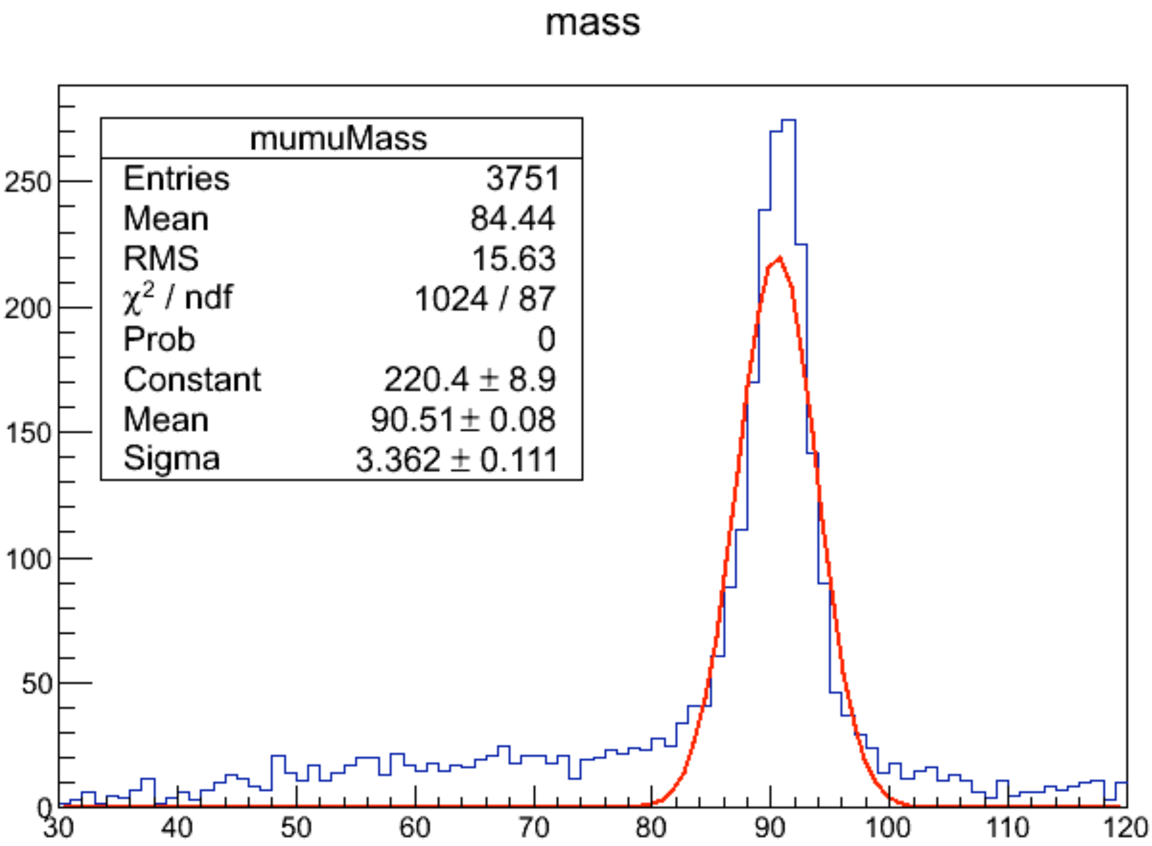
Login to ROOT as follows

```
root -l
```

and execute the following commands

```
TFile f("myZPeakCRAB.root");
f.cd("analyzeBasicPat");
gStyle->SetOptFit(111111);
mumuMass->Fit("gaus");
```

This will pop up the following histogram. Save this histogram as pdf or postscript or eps file using the menu of the histogram window. As you can see we should fit a sub-range as this fit is not a good fit. In the next part of this exercise, we will fit a sub-range of the mumuMass distribution but for this we will use a ROOT macro as using inbuilt ROOT functions have very minimal usage. For more complex or useful fitting functions, one has to use a macro.



Quit ROOT as follows

```
.q
```

The line gStyle->SetOptFit(111111); enables all the histogram statistics to be displayed.

QUESTION 14.1 - What are the values of the mean Z Mass and chisquare/ndf that you get?

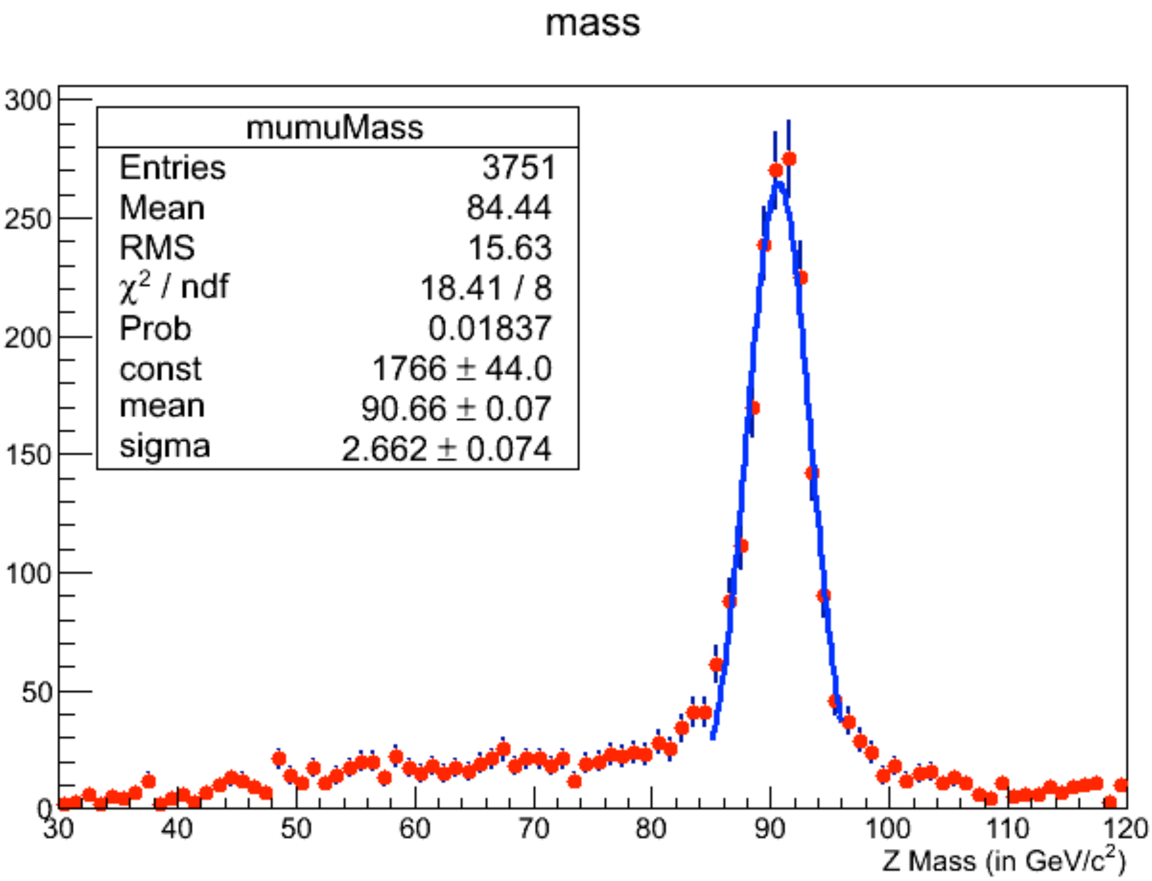
Using a macro of your own in ROOT

As you have seen above that we should fit a sub-range of the Z mass distribution as the fit in the full range is not all that great. In this exercise, we will fit a sub-range of the mumuMass distribution but for this we will use a ROOT macro as using inbuilt ROOT functions have very minimal usage. For more complex or useful fitting functions, one has to use a macro. The macro to run is FitZPeak.C. This macro calls another macro called BW.C. So please copy, paste and save them with the corresponding names in YOURWORKINGAREA/CMSSW_5_3_13/src. Note that now executing the macro itself, in addition to fitting the Z mass peak, opens the myZPeakCRAB.root file.

To run this macro, execute the following command from the area YOURWORKINGAREA/CMSSW_5_3_13/src.

```
root -l FitZPeak.C
```

This should pop up a histogram (shown below) and you will find yourself in a ROOT session.



mumuMass	
Entries	3751
Mean	84.44
RMS	15.63
χ^2 / ndf	18.41 / 8
Prob	0.01837
const	1766 ± 44.0
mean	90.66 ± 0.07
sigma	2.662 ± 0.074

You can save this plot from the menu on top of the histogram and quit ROOT session by doing the following:

```
.q
```

Here is some explanation of the macro. We have defined the Gaussian distribution that we want to fit in the macro BW.C as the following. Note that in the same macro we have also defined Breit-Wigner that you can try yourself to fit by using it. However, we fit Breit-Wigner in the later part of the exercise by using RooFit in the next part of this exercise.

```
Double_t mygauss(Double_t * x, Double_t * par)
  {
   Double_t arg = 0;
   if (par[2]<0) par[2]=-par[2];  // par[2]: sigma
   if (par[2] != 0) arg = (x[0] - par[1])/par[2];  // par[1]: mean
       //return par[0]*BIN_SIZE*TMath::Exp(-0.5*arg*arg)/
       //   (TMath::Sqrt(2*TMath::Pi())*par[2]);
     return par[0]*TMath::Exp(-0.5*arg*arg)/
           (TMath::Sqrt(2*TMath::Pi())*par[2]); // par[0] is constant
  }
```

par[0], par[1] and par[2] respectively are the constant, mean and the sigma parameters. Also x[0] refers to the x-axis variable. BW.C is called by FitZPeak.C in the code line gROOT->LoadMacro("BW.C"); . The initial values of the three fitted are defined as follows in the FitZPeak.C

```
func->SetParameter(0,1.0);
func->SetParName(0,"const");
func->SetParameter(2,5.0);
func->SetParName(2,"sigma");
func->SetParameter(1,95.0);
func->SetParName(1,"mean");
```

Also note that in the macro FitZPeak.C, we have commented the following lines and used the two lines below it. The reason being that we want to fit a sub-range. If you would want to fit the entire range of the histogram, get the minimum and maximum value of the range by instead using the lines that have been commented.

```
//float massMIN = mumuMass->GetBinLowEdge(1);
//float massMAX = mumuMass->GetBinLowEdge(division+1);
float massMIN = 85.0; float massMAX = 96.0;
```

You can also save the plot to a postscript file by un-commenting the following line in FitZPeak.C

```
 //c1->Print ("myZmass_Gausfitted.eps");
```

QUESTION 14.2 - What mean value of Z Mass in the fitted sub-range do you get?

Using a macro in RooFit

Before we start, have a look at the twiki RooFit (see link: https://twiki.cern.ch/twiki/bin/view/CMS/RooFit) to get a feeling for it. The macro to fit Z mass peak in RooFit is RooFitMacro.C. So please copy RooFitMacro.C in YOURWORKINGAREA/CMSSW_5_3_13/src. Then execute the following:

```
root -l RooFitMacro.C
```

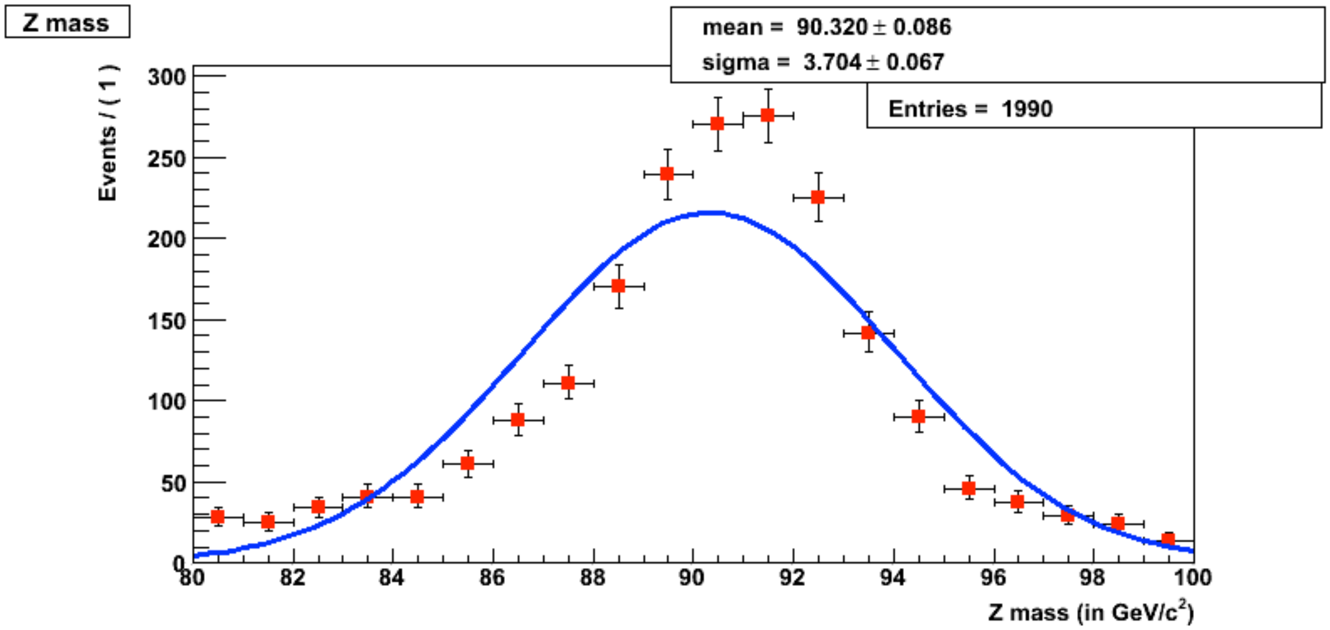
This should pop a histogram (shown below) and you will find yourself in a ROOT session. You can save this plot from the menu on top of the histogram and quit ROOT session by doing the following:

```
.q
```

Note that if you look at the macro, it has the following code lines:

```
 RooGaussian gauss("gauss","gauss",x,mean,sigma);
// RooBreitWigner gauss("gauss","gauss",x,mean,sigma);
// RooVoigtian gauss("gauss","gauss",x,mean,width,sigma);
```

To do a Gaussian, we simply commented out the Breit-Wigner and Voigtian (convolution of Breit-Wigner and Gaussian).



QUESTION 14.3 - What is the mean and sigma for each of the gaussian fit in RooFit?

Fitting a Breit-Wigner

Using a macro in ROOT

To make fit Breit-Wigner we first uncomment the Breit-Wigner and comment out the Gaussian part in FitZPeak.C as follows (using /* and */) :

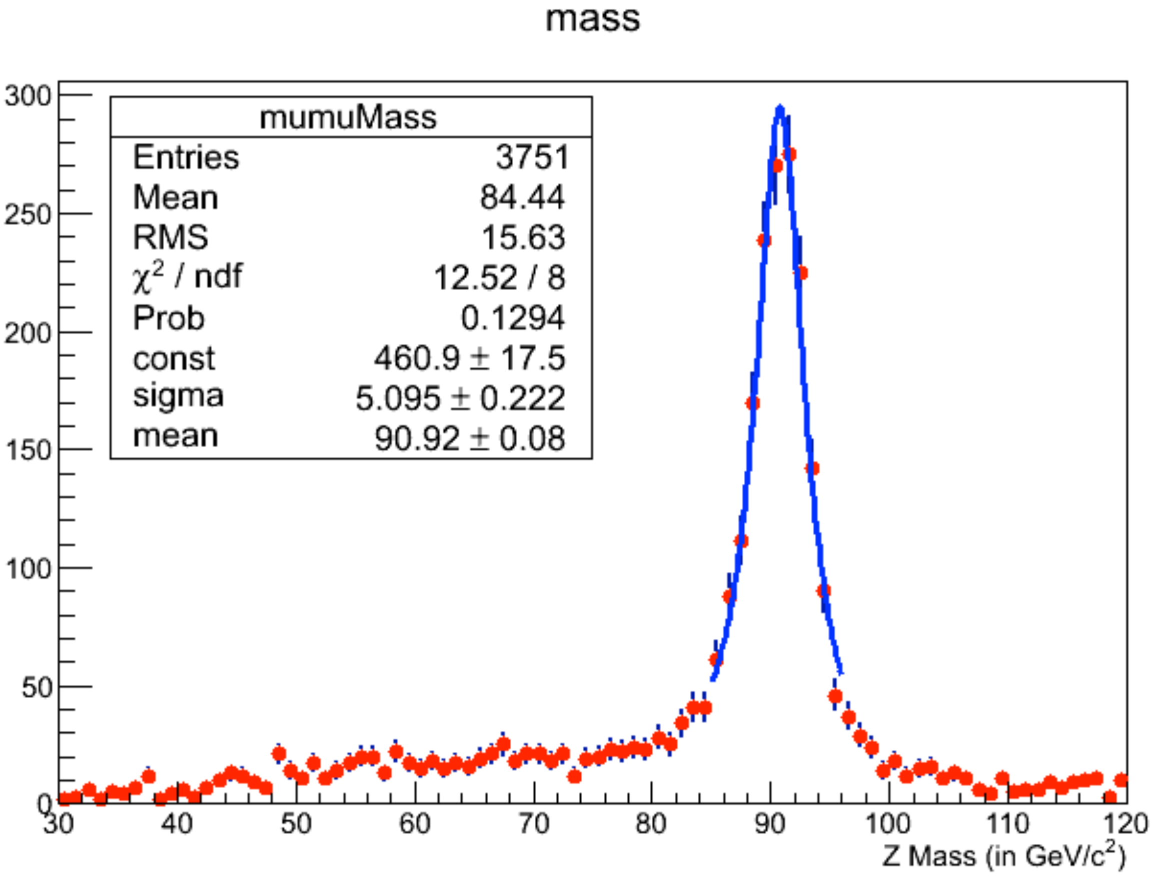
```
//////////////// //For Gaussian// ///////////////
/*
TF1 *func = new TF1("mygauss",mygauss,massMIN, massMAX,3);
  func->SetParameter(0,1.0);   func->SetParName(0,"const");
  func->SetParameter(2,5.0);   func->SetParName(2,"sigma");
  func->SetParameter(1,95.0);     func->SetParName(1,"mean");
  Z_mass->Fit("mygauss","QR"); TF1 *fit = Z_mass->GetFunction("mygauss");
*/
///////////////////// // For Breit-Wigner// ////////////////////
TF1 *func = new TF1("mybw",mybw,massMIN, massMAX,3);
 func->SetParameter(0,1.0);   func->SetParName(0,"const");
 func->SetParameter(2,5.0);     func->SetParName(1,"sigma");
 func->SetParameter(1,95.0);    func->SetParName(2,"mean");
```

```
Z_mass->Fit("mybw","QR"); TF1 *fit = Z_mass->GetFunction("mybw");
```

Then execute the following:

```
root -l FitZPeak.C
```

This should pop a histogram (shown below) and you will find yourself in ROOT seession.



You can save this plot from the menu on top of the histogram and quit ROOT session by doing the following:

```
.q
```

QUESTION 14.4 - What is the mean and sigma for Breit-Wigner fit using the macro?

Using a macro in RooFit

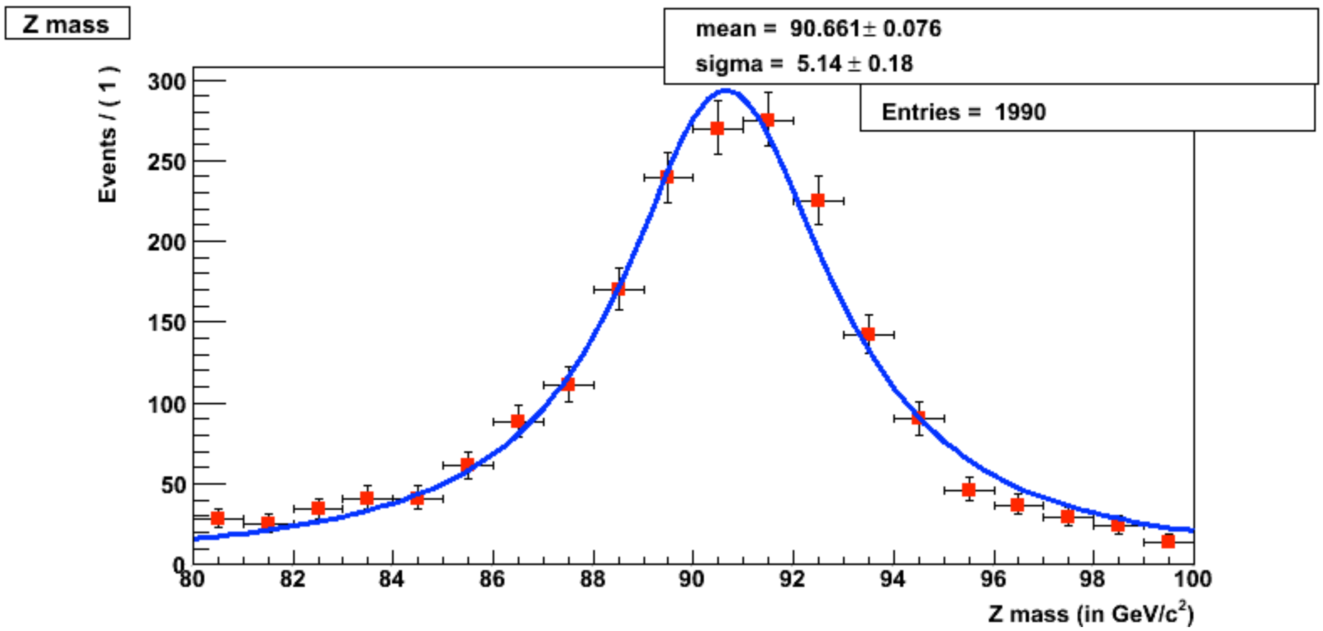
Before we proceed we need to uncomment and comment out few lines in RooFitMacro.C to have them look as follows:

```
 //RooGaussian gauss("gauss","gauss",x,mean,sigma);
  RooBreitWigner gauss("gauss","gauss",x,mean,sigma);
  // RooVoigtian gauss("gauss","gauss",x,mean,width,sigma);
```

Then execute

```
root -l RooFitMacro.C
```

This should pop a histogram (shown below) as follows and you will find yourself in ROOT session.



You can save this plot from the menu on top of the histogram and quit ROOT session by doing the following:

```
.q
```

QUESTION 14.5 - What is the mean and sigma for Breit-Wigner fit using RooFit tool ?

Fitting a Convolution of Gaussian and Breit-Wigner

Using a macro in RooFit

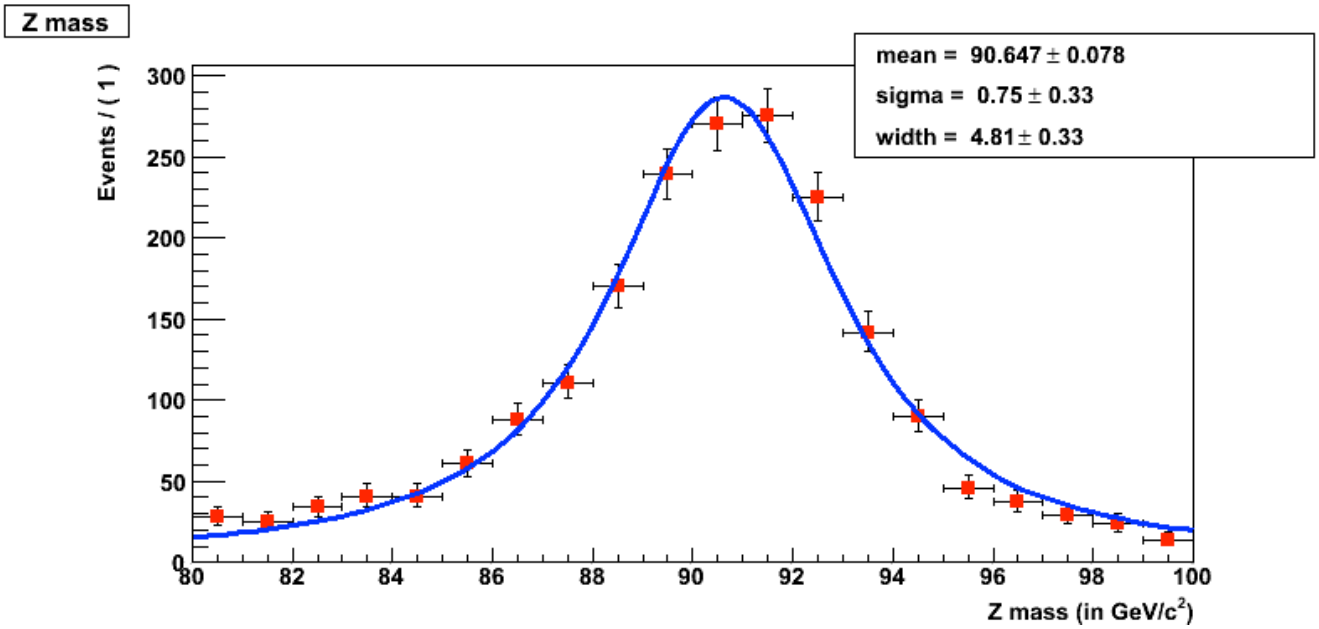
Before we proceed we need to uncomment and comment out few lines in RooFitMacro.C to have them look as follows:

```
 //RooGaussian gauss("gauss","gauss",x,mean,sigma);  // RooBreitWigner
gauss("gauss","gauss",x,mean,sigma);   RooVoigtian
gauss("gauss","gauss",x,mean,width,sigma);
```

Then execute

```
root -l RooFitMacro.C
```

This should pop a histogram (shown below) as follows and you will find yourself in ROOT seession.



You can save this plot from the menu on top of the histogram and quit ROOT session by doing the following:

```
.q
```

QUESTION 14.6 - What is the mean and sigma for the convoluted fit using - RooFit tool?