Introduction (not necessary to understand for understanding the technical problem)

- background modelization/spurious signal study, one searches to find the best functional form (=function) that describes the background in a Monte-Carlo.
- The traditional procedure consists to fit the backround-only MC sample with a model background+signal, that is : to search for a residual signal (so-called spurious signal), with fixed signal shape parameter (but yield floated) in the background, due to the non infinite statistics sample and to the limited number of parameters for the chosen functional form of the background

(the shape used for modelizing fake signal is chosen previously from a MC signal sample)

- In the computation of residual signal fitted in the background sample, one makes several hypotheses of the fixed position (mass) of the signal with respect to its Standard Model value

(example : if mass of signal SM is at 125 GeV one makes a fit of background + signal for masses hypotheses of signal in a loop of typically +/-4 GeV around 125 GeV

- The spurious signal is the maximum of the fake signal fitted in the background sample, for various hypotheses of mass of signal.
- the choice of background functional form will be the one that minimize the fitted spurious signal.

Introduction (not necessary to understand for understanding the technical problem)

- People involved in background modelling typically do the following procedure

1) They make the fit of the signal shape in a signal only MC, in order to obtain the shape parameters. Since no model would be able to cover the very large range of the invariant mass, one typically fit in the « main core » of the distribution. One does not fit in a very wide range that would cover very large tails, else no functional form would work. So ones fits with Range(…) option.

2) They save the pdf (RooAbsPdf) in a root file
3) The « load » the pdf in the program that computes the spurious signal.
4) In this last procedure, they loop over various hypothesis of the mass of the signal : for that, one shifts the « mean » parameter of the signal shape, to modelize the shifted signal in the loop and fix it, as well as all shape parameters

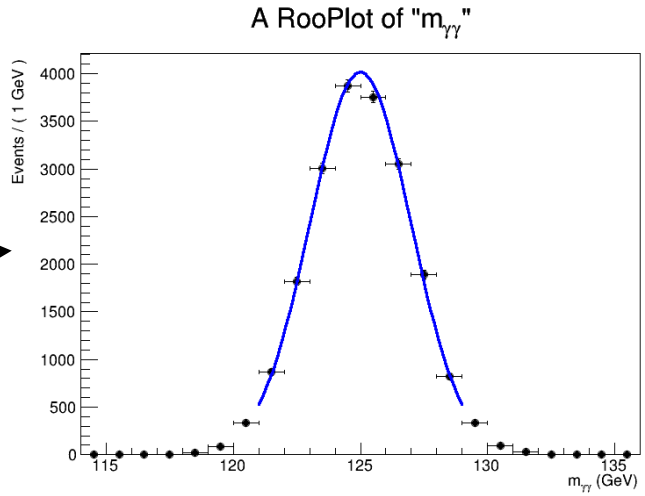
The point is that if people « load » the previous saved pdf in a root file, they will occur a big mistake.
If people « recreate » the pdf from the numerical parameters of the signal shape, they will not occur a problem.
The mistake is described in the following, due to a bug (?) or at least a vicious feature of roofit

Problem that would be encountered if exporting/importing signal pdf from root file

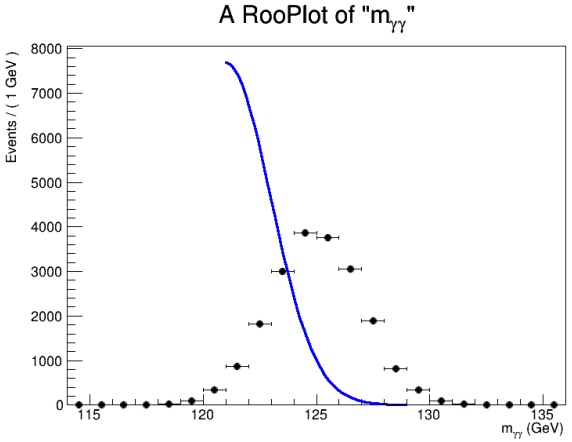
- Illustration of the problem. Let's fit the signal shape (in order to use it later for the background modelization)

- Let's fit in a restricted range

(of course, in reality, we would not fit in that so tiny restricted range, but the problem described would be exactly the same)



- Now what happens : if we shift [here to left] the mean parameter of the signal shape (in the loop over hypotheses of position of peak, for the spurious signal study), the pdf is distorted when we are outside the range of the historical chosen fit range (pdf « remembers » the range of the fit that was done previously, even if we save the pdf in a root file)



→people who shift the signal in a given position within their loop over mass hypothesis but who don't see one by one the plots will not realize that the shape is distortted outside the range of the previous fit

→Their results will be **completely wrong** because the fit « remembers its range »

→It is a buggy feature… Because of that, solution is to « recreate the pdf » from the numerical parameters

Problem that would be encountered if exporting/importing signal pdf from root file

Minimum 71 lines program to illustrate the point

```
#include <iostream>
#include <string.h>

#include <RooAbsPdf.h>
#include <RooArgSet.h>
#include <RooDataSet.h>
#include <RooGaussian.h>
#include <RooPlot.h>
#include <RooRealVar.h>
#include <TCanvas.h>

using namespace RooFit;
using namespace std;

void Draw_pdfs(RooDataSet *current_dataset);
RooAbsPdf *pdf_total;

RooDataSet *dataset;

RooArgSet argset_finalDV;

RooRealVar *roorealvar_m_yy;

int MinimumProblemShiftPdf()
{
  roorealvar_m_yy=new RooRealVar("m_yy","m_{#gamma#gamma}",22,114,136,"GeV");
  roorealvar_m_yy->setBins((roorealvar_m_yy->getMax()-roorealvar_m_yy->getMin())/1.); //m_yy: bin width=1 GeV
  roorealvar_m_yy->setRange("fitrange",121,129);

  argset_finalDV.add(*roorealvar_m_yy);

  RooRealVar *roorealvar_muGauss=new RooRealVar("muGauss","muGauss",125,100,140); //please note that mu gauss is defind in 100 ; 140, large enough to allow to shift it
  RooRealVar *roorealvar_sigmaGauss=new RooRealVar("sigmaGauss","sigmaGauss",2,0,10);
  pdf_total=new RooGaussian("pdf_total","pdf_total",*roorealvar_m_yy,*roorealvar_muGauss,*roorealvar_sigmaGauss);

  cout << "generate dataset for the minimum example" << endl;
  dataset=pdf_total->generate(RooArgSet(*roorealvar_m_yy),RooFit::NumEvents(20000));
  //=============================================================================
  cout << "m_yy : range for fit : " << roorealvar_m_yy->getRange("fitrange").first << ", " << roorealvar_m_yy->getRange("fitrange").second << endl; //this is a 'pair'

  pdf_total->fitTo(*dataset,RooFit::Range("fitrange"),RooFit::SumW2Error(kTRUE),RooFit::Save(kTRUE));

  //  Draw_pdfs(dataset); //pdf would be fine here

  //shift the mean to 121 : the pdf is "truncated" at the part where it was historically fitted
  ((RooRealVar *)pdf_total->getVariables()->find("muGauss"))->setVal(121);
  //that is the pdf has no more the possibility to exist outside the range of its historical range, apart if we recreate it from scratch : this behaviour is rather strange

  Draw_pdfs(dataset);
  return 0;
}
//=============================================================================================
void Draw_pdfs(RooDataSet *current_dataset)
{
  TCanvas *canvas=new TCanvas("canvas","canvas",800,600);

  RooPlot *rooplot=roorealvar_m_yy->frame();

  current_dataset->plotOn(rooplot,RooFit::DataError(RooAbsData::SumW2));

  pdf_total->plotOn(rooplot,Range("fitrange"),NormRange("fitrange"));

  rooplot->Draw();

  canvas->SaveAs("plot.png");

  delete rooplot;
  delete canvas;

  return;
}
```