ALMA MATER STUDIORUM - Università di Bologna

87944 - STATISTICAL DATA ANALYSIS FOR NUCLEAR AND SUBNUCLEAR PHYSICS

Module 3 : Laboratory of Stat. Data Analysis for Nucl. and SubNucl. Physics teacher: G. Sirri

Hands-on n. 3: Systematics, Control regions. ROOSTATS

Submission Deadline: Recommended within two weeks

This assignment requires solving two exercises by writing a C++ macro for ROOT. Alternatively, solutions may use the RooFit libraries in a Python+ROOT script or a Jupyter notebook, though the provided hints are tailored for C++.

Students may choose between using the RooFit Factory or standard RooFit Classes to implement the solution. Teamwork is allowed; however, each student must submit their own work individually.

Assignment 3.1: Choose and Solve ONE Exercise on *Systematics, Control regions* from the following:

- [OPERA-nu oscillations]
  Discovery of tau neutrino appearance in the CNGS neutrino beam with the OPERA experiment
- [Exercise Composite.LHCb + Simultaneous Fit]
  First observation of the rare purely baryonic decay B^0→p p̄

Assignment 3.2: Choose and Solve ONE Exercise on *RooStats* from the Following:

- [ATLAS.H->4l]
  Measurement of the Higgs boson mass in the H→ZZ∗→4l and H→γγ channels with √s=13 TeV pp collisions using the ATLAS detector
- [CMS- Higgs Discovery 2012]
  Observation of a new boson at a mass of 125 GeV with theCMS experiment at the LHC
- [CMS.jpsi]
  J/ψ and ψ(2S) production in pp collisions at √s = 7 TeV

DOCUMENTATION:

- slides shown during the lecture.

- RooFit starting point: https://root.cern/manual/roofit/

  - RooFit Manual (PDF A4 format)
  - RooFit Quick Start Guide (PDF A4 format)
  - Here is a link to a 200 slide presentation on RooFit presented in the French School of Statistics 2008 (slides are in English)

- RooStats wiki page: twiki.cern.ch/twiki/bin/view/RooStats/WebHome

- ROOT Tutorials (https://root.cern/doc/master/group__Tutorials.html There are over hundred macros illustrating many aspects of RooFit, Roostats, TMVA functionalities

Exercise: OPERA-nu oscillations
Discovery of tau neutrino appearance in the CNGS neutrino beam with the OPERA experiment

Inspired by the analysis described in Phys. Rev. Lett. 115, 121802 (2015), arXiv:1507.01417v2 [hep-ex] 2 Nov 2015

https://arxiv.org/pdf/1507.01417.pdf

The OPERA experiment was designed to search for νμ→ντ oscillations in appearance mode, i.e. by detecting the τ-leptons produced in charged current ντ interactions. The experiment took data from 2008 to 2012 in the CERN Neutrinos to Gran Sasso beam.

The observation of 5 ντ candidate events allow assessing the discovery of νμ→ντ oscillations in appearance mode with a significance larger than 5 σ.

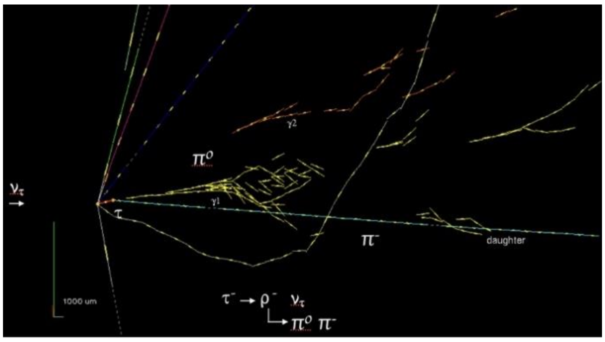


Figure 1 One of the tau neutrino interaction events detected with the ECC in the OPERA experiment.

A number counting analysis with 4 channels and background uncertainties

In this analysis, tau candidates are selected by looking into 4 channels corresponding to 4 different tau decay modes. The observed number of ντ candidates n_i for each individual τ decay channel i is considered as an independent Poisson process with expectation μ*s_i + β_i .where s_i (constant) and β_i (parameter) are expected signal and background events, respectively; the signal strength μ (parameter) is a continuous multiplicative parameter for the expected signal. The background-only hypothesis corresponds to μ = 0, and the nominal signal to μ = 1.

The likelihood includes Gaussian terms to account for the background uncertainties:

$$\mathcal{L} = \prod_{i=0}^{4} \mathrm{Pois}(\mathrm{n_i}|\mu * s_i + \beta_i)\mathrm{Gaus}(\mathrm{b_i}|\beta_i, \sigma_{b_i})$$

where σ_bi is the background uncertainty for channel i (from Table 3) and β_i are the background parameters Gaussian modelled.

PART 1 Counting Model in RooFit (one channel) with uncertain background

Just start with a further simplification: consider only one (overall) channel defined as the sum of events neglecting which decay mode they are associated with.

For this part you may take values from the row "Total" of Table 3. They are the observed events (observable), the nominal number of signal events (constant, neglect the error), the expected number of background events (parameter), and the uncertainty of the background events (constant).

- Create a counting model based on Poisson Statistics

  Let's suppose we observe **nobs** events when we expect **nexp**, where **nexp = μ s+b** (**μ** is the signal strength, **s** is the nominal number of signal events and **b** is the number of background events. The expected distribution for **nexp** is a Poisson **Poisson(nobs | μ s +b)**. **μ** is the parameter we want to estimate or set a limit (the parameter of interest), while **b** is the nuisance parameter. The real value of **b** is unknown. Its best estimate is "**b0**" with uncertainty **sigmab**. To express this uncertainty we add in the model a Gaussian constraint. We can interpret this term as having an additional measurement **b0** with an uncertainty **sigmab**: i.e. we have a likelihood for that measurement **Gaussian(b0 | b, sigmab)**.

  *Hints: i) look at the slide "**Counting Model in RooFit (one channel) With Signal Strength**" of the section "RooFit Model for a Number Counting Analysis", then ii) incorporate the gaussian contraint* *as shown in the slide “EXAMPLE: A simple counting experiment with an uncertain background” of the section “Incorporating systematics”*

You may also inspire your code by looking to this example:

https://twiki.cern.ch/twiki/pub/RooStats/RooStatsExercisesMarch2015/CountingModel.C

- We generate a hypothetical observed data set. Since we have a counting model, the data set will contain only one event and the observable nobs will have our desired value .

```
// make data set with the number of observed events; nobs is a RooRealVar
RooDataSet data("data","", nobs);
nobs setVal(5);
data.add(nobs);
```

- Import the model and the data in a RooFit RooWorkspace and save to file

```
// import data set in workspace and save it in a file
w.import(model)
w.import(data);
w.writeToFile("CountingModel.root", true);
```

- Create the **`ModelConfig`** object and import in the workspace. We need to add in the **`ModelConfig`** also **`b0`** as a “global observable”. The reason is that **`b0`** needs to be treated as an auxiliary observable in case of frequentist statistics and varied when tossing pseudo-experiments.

  As b0 is the global observable, b0 needs to have a range

  However, it is a constant variable (this is needed when we fit the model).

```
w.var("b0")->setConstant(true);
```

To estimate the significance, we need to perform an hypothesis test. We want to disprove the null model, i.e the background only model against the alternate model, the background plus the signal. In RooStats, we do this by defining two ModelConfig objects, one for the null model (the background only model in this case) and one for the alternate model (the signal plus background).

The null hypothesis (i.e. the observed number of events is just a statistical fluctuation of the background only) corresponds to a signal strength mu = 0.

Hints:
(method 1) see the slides and evaluate the significance using the ProfileLikelihoodCalculator.

(method 2) You may also refer to $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C to inspire own code

(method 3, recommended)

You can simply run $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C by passing your workspace as argument. Note that in this case it is not necessary to explicitly define a ModelConfig for the null hypothesis, it is automatically computed setting the parameter of interest (signal strength, mu) to zero. Let’s use calculator type 2 ‘Asymptotic calculator’ and test type 3 ‘Profile Likelihood one sided’. The calculator type 0 ‘Frequentist’ is infeasible at this level of significance…!

NOTE - ONLY FOR THIS EXERCISE - that this model is a number counting analysis implemented with an explicit Poisson function, so that from a technical point of view it is not “extended” for RooFit. You must inform the StandardHypoTestDemo(..) function about that by settuing the argument “bool useNC” as a true value)

PART 2 Counting Model in RooFit (4 independent channels) with uncertain backgrounds

Be more realistic and try to repeat the exercise creating a model as the product of 4 independent channels as done in the paper.

For each channel you may define a number observed events (observable), a nominal number of signal events (constant, neglect the error), a expected number of background events (parameter), and a uncertainty of the background events (constant).

Take values from Table 3.

Please, take care that the signal strength is just one for all the channels!

Hints: you may define 4 Poisson PDFs for the signals and 4 Gaussian constraing terms for the background.
The final model is the product (use RooProduct) of all PDFs.

The RooDataSet still contains only one event and is defined as a set of observables (RooArgSet)

```
RooDataSet data("obsData","", RooArgSet( obs1, obs2, obs3, obs4 ) );
obs1 setVal(…);
obs2.setVal(…);
…
data.add(  RooArgSet( obs1, obs2, obs3, obs4 ) );
```

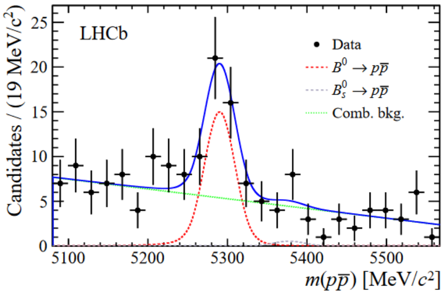
- Estimate significance and p-values and check the ones quoted in the paper.

(submit the code)

[Exercise Composite.LHCb + Simultaneous Fit]
First observation of the rare purely baryonic decay $B^0 \to p\,\bar{p}$

Inspired by Figure 1 of: “First observation of the rare purely baryonic decay B^0 → $p\ \bar{p}$” arXiv:1709.01156v2 [hep-ex] 6 Dec 2017; https://arxiv.org/abs/1709.01156

See also https://cerncourier.com/a/the-rarest-b0-decay-ever-observed/



In 2017 the LHCb collaboration has observed the rare baryonic decay $B^0 \to p\,\bar{p}$. The branching fraction was measured at the level of about 1.3 per 100 million decays, which ma kes this decay mode the rarest decay of a B0 meson ever observed. It is also the rarest observed hadronic decay of all beauty mesons.

The search for the rare decays $B^0 \to p\,\bar{p}$ and $B_s^0 \to p\,\bar{p}$ had previously been performed by LHCb with the full 3 fb^{-1} data sample collected during the first run of the LHC. An excess of $B^0 \to p\,\bar{p}$ candidates with respect to the background-only hypothesis is observed with a statistical significance of 5.3 standard deviations. The hint of a $B_s^0 \to p\,\bar{p}$ signal reported in 2013 is, however, not confirmed, and an upper limit for the corresponding branching fraction was set. The measured $B^0 \to p\,\bar{p}$ and $B_s^0 \to p\,\bar{p}$ branching fractions are compatible with the latest theoretical calculations.

The exercise aims to reproduce the Invariant Mass Distribution of Figure 1

Download **rarest_b0_decay.dat** dataset collected by a B-meson experiment

Load the **unbinned** dataset from the file rarest_b0_decay.dat

*Tip: RooDataSet data = *RooDataSet::read("rarest_b0_decay.dat", x, "v");*

Using RooFit, define a (non-extended) composite model for invariant mass.

The model components are:

- a background (suppose exponential with coefficient ranging from -1, -0.000001);
- a Gaussian peak around the B^0 mass
- a Gaussian peak around the B_s^0 mass

Fit the model to the data using a maximum likelihood fit. Plot data and model.

Superimpose each single component with different color

Tip: use the named functions RooFit::Components(…) and RooFit::LineColor(…).

Comment: The plot may not appear as expected. To address this, set reasonable ranges for the signal peaks and assume that B^0 and B_s^0 have the same widths (i.e. the same RooRealVar) to simplify the model.

PART 2: Make histogram of residual and pull distributions

Have a look to https://root.cern.ch/doc/master/rf109__chi2residpull_8C.html

Note: methods residHist(…) and pull(..) by default compute the residuals (pulls) of the latest-plotted histogram with respect to the latest-plotted curve.

- Construct a histogram with the residuals of the data w.r.t. the curve
- Construct a histogram with the pulls of the data w.r.t the curve
- Create a new frame to draw the residual distribution and add the distribution to the frame
- Create a new frame to draw the pull distribution and add the distribution to the frame

Visualize the correlation matrix

- Look at the correlation matrix of the fit.

To make a visual presentation of the correlation matrix, save the RooFitResult object returned by fitTo(..) (don’t forget to add RooFit::Save() as argument of fitTo(..))

- Then, add the following code:

```
TCanvas c;
gStyle->SetPalette(1) ;
fit_results->correlationHist()->Draw(“colz”) ;
```

PART 3: Control Region /simultaneous fit

Suppose you can measure the background in a CONTROL REGION where the observable ranges from 4000 to 5000 and no signal is present.

Open a new file and copy the model for the physics region and the read the data from the file as in PART1.

Don’t perform the fit, don’t create a plot.

Define a new observable for the control region which ranges from 4000 to 5000.

Create an exponential model (model_ctl) for the control region. NOTE: the coefficient is shared between the physics and the control region (…it means that you don’t need to define it twice).

Set the coefficient value to -1.0e-3 and generate 10000 events for the control region (`data_ctl`).

Now you **can follow the RoofiT tutorial 501** (https://root.cern/doc/master/rf501__simultaneouspdf_8C.html) to construct a simultaneous fit.

- Define category to distinguish physics and control samples events
- Construct combined dataset in (x,y, sample) where x is the observable in the physic region an y is the observable in the control region.

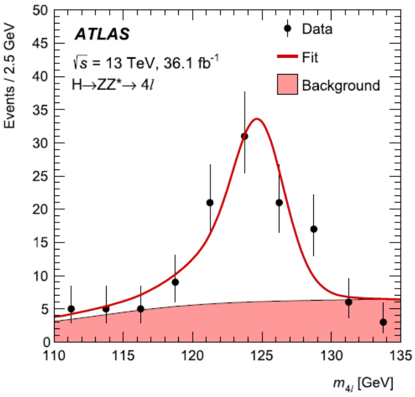
```
RooDataSet combData("combData","combined data", RooArgSet(x,y), Index(sample),
                          Import({{"physics", &data}, {"control", data_ctl}}) );
```

(warning: Import requires a pointer RooDataSet).*

- Construct a simultaneous pdf using category sample as index;
- Associate the model of the physics region with the physics state and model_ctl with the control state;
- Perform **simultaneous fit** of model to data and model_ctl to data_ctl

- Make a frame for the physics sample and Plot all data tagged as physics sample. Plot "physics" slice of simultaneous pdf.
- Make a frame for the control sample and Plot all data tagged as control sample Plot "control" slice of simultaneous pdf.

Exercise ATLAS.H->4l
Measurement of the Higgs boson mass in the H→ZZ∗→4l and H→γγ channels with √s=13 TeV pp collisions using the ATLAS detector



Inspired by FIGURE 1(a) of Physics Letters B 784(2018)345–366

https://www.sciencedirect.com/science/article/pii/S0370269318305884
arXiv:1806.00242 [hep-ex] 10 Oct 2018 https://arxiv.org/abs/1806.00242

This exercise aims to

i) reproduce the plot of Figure 1 (a)
ii) compute the excess significance in the region of 125 GeV,
iii) compute significance (p-value) as function of the signal mass
iv) compute the interval (or an upper limit) to the Higgs mass with Profile Likelihood and Feldman.Cousin using RooFit and RooStats.

PART 1. MODEL, FIT, PLOT

Please download the text file `higgs_4l.dat` with 123 events.

Events have been selected given the presence of 4 leptons with invariant mass in the 110-135 GeV range. Each line corresponds to one single event. The observable is the invariant mass of the 4-leptons system.

- define the observable as a RooRealVar

```
RooRealVar x{"x","invariant mass", 110, 135, "GeV"};
```

- set the number of bins according to the Figure 1 (a)
- import the dataset as RooFit unbinned dataset (`RooDataSet`)

  Hints: use ASCII import/export for datasets `RooDataSet::read(…)`

  as in `$ROOTSYS/tutorials/roofit/rf102_dataimport.C`

- define a model for the invariant mass distribution.

The p.d.f. has two components:

i) a Higgs signal model called `smodel`: H → 4l peak
ii) a background model called `bmodel`: modelled by a polynomial function of degree 2;

The Higgs signal model is a complex model derived from a Breit-Wigner distribution with width = 4.1 GeV.

For this exercise we use instead a Crystal Ball (CB) shape distribution implemented by the `RooCBshape` class. There are several constraints to be taken into account to define the parameters

- Higgs `mass` (the mean of CB) ranges is [110, 150],
- Higgs `width` (the width of CB) is 4.1/2.35 (constant),
- `alpha` is 0.6 (constant)
- the power-law's exponent `n` is 20 (constant)

The Background model is a polynomial PDF of degree 2 implemented by `RooPolynomial`. Given that the polynomial is normalized to 1 (automatically by RooFit), you must provide just 2 independent parameters.

```
RooRealVar a1("a1", "The a1 of background", -160, -100, -200);
RooRealVar a2("a2", "The a2 of background", 2.7, 2, 4);
```

More realistic constraints are described in the paper, but we can neglect it.

Define the expected yields for the different components of the spectrum (`nsignal`, `nbackground`).

With the Extended Likelihood Formalism build an Extended Composite Model called `model` by using `RooAddPdf` (or the RooFit factory as well).

- Fit the model to the data using a maximum likelihood fit.
- Plot data and model.
- Import dataset and model (and automatically all related elements) into a `RooWorkspace`.

Create a Model Config `"ModelConfig"` for the signal plus background model. Use the Higgs signal yield as the parameter of interest, the floating parameters are nuisance, and ensure constants are explicitily set as constant.

This model configuration must be used for:

- the Significance vs Mass plot
- hypothesis test (StandardHypoTestDemo)

Hint: how to set constantness for constant parameter:

```
w.var("width")->setConstant(true);
w.var("alpha")->setConstant(true);
w.var("n")->setConstant(true);
```

Hint: How to define a Model Config:

https://twiki.cern.ch/twiki/bin/view/RooStats/RooStatsExercisesMarch2015#Exercise_1_Create_a_ModelConfig

```
RooStats::ModelConfig mc("ModelConfig", &w);
mc.SetPdf(...
mc.SetParametersOfInterest(...
mc.SetObservables(...
// define set of nuisance parameters
w.defineSet("nuisParams","nbackground,a1,a2");
mc.SetNuisanceParameters(*w.set("nuisParams"));
mc.SetSnapshot(nsignal);
```

- IMPORTANT AND ONLY FOR THIS: set also the mass as a constant, required for the significance vs mass plot.

```
// mass.setConstant(true);  // this doesn't work, use the following..
w.var("mass")->setConstant(true);
```

- Import the ModelConfig `"ModelConfig"` into the workspace

Define another Model Config `"ModelConfig_mass"` where

- the parameter of interest is the HIGGS MASS
- the set of nuisance parameters is `nuisParams2` which includes the signal yield `nsignal`

```
w.defineSet("nuisParams2", "nbackground,a1,a2,nsignal,width");
```

This model configuration must be used for

- Higgs mass interval with profile likelihood (StandardProfileLikelihoodDemo)
- Higgs mass interval with Feldman Cousin (StandardFeldmanCousinsDemo)

- Import the ModelConfig `"ModelConfig_mass"` into the workspace
- Save the workspace to a file

PART 2. Higgs mass Profile Likelihood interval

- Compute a 68% C.L. interval on the **Higgs mass**, using RooStats Profile Likelihood Calculator.
- Plot the likelihood interval and adjust the range by hand (right click on axis and setrangeuser). x from 120 to 128, y from 0 to 20
- Try to change the confidence level.

Hints: Look at the code in the slides. You may refer to $ROOTSYS/tutorials/roostats/ StandardProfileLikelihoodDemo.C to guide your work or you can just run StandardProfileLikelihoodDemo.C by passing your workspace as argument (recommended).

PART 3. Higgs mass Feldman Cousin interval

- Compute a 90% C.L interval on the **Higgs mass**, using RooStats Feldman Cousin Calculator.
- Try to change the confidence level.

Hint: You may refer to $ROOTSYS/tutorials/roostats/StandardFeldmanCousinsDemo.C to guide your work or you can just run StandardFeldmanCousinsDemo.C with your workspace as argument (recommended).

PART 4. Compute the significance (Hypothesis Test)

To estimate significance, we perform a hypothesis test to reject the null (background-only) model in favor of the alternate (signal-plus-background) model. In RooStats, this involves defining two ModelConfig objects: one for the null and one for the alternate model.

The null hypothesis (i.e. the peak in the data is just a statistical fluctuation of the background only) corresponds to a Higgs signal yield `nsignal` = 0.

Hints:
(method 1) see the slides and evaluate the significance using the ProfileLikelihoodCalculator.

(method 2) refer to $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C to guide your work

*(method 3, recommended) Run the script $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C with your workspace as the argument. You don't need to manually define a ModelConfig for the null hypothesis—it's automatically set by assigning the parameter of interest (signal strength, μ) to zero. Use **Calculator Type 2** ("Asymptotic Calculator") and **Test Type 3** ("Profile Likelihood One-Sided"). Avoid **Calculator Type 0** ("Frequentist"), as it's impractical for this level of significance.*

PART 5. Compute significance (p-value) as function of the signal mass

<u>As an optional exercise in RooStats</u>, we'll explore how significance (or p-value) varies with the signal mass hypothesis in a Gaussian signal plus exponential background model

Using the AsymptoticCalculator, we'll test several mass points, plot the p-value for the null hypothesis (p0), and estimate the expected significance as a function of mass. The expected significance can be calculated with AsymptoticCalculator::GetExpectedPValues, using observed p-values for the null and alternate models. Refer to the Asymptotic formulae paper for details.

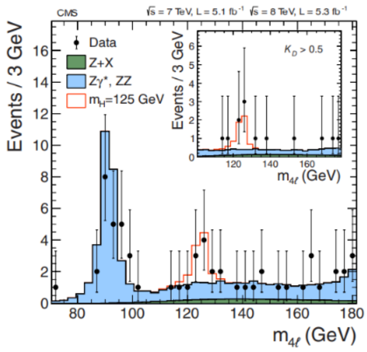
Hints: you can follow the solution described in Exercise 6b from

https://twiki.cern.ch/twiki/bin/view/RooStats/RooStatsExercisesMarch2015#Exercise_6b_Compute_significance

It is worth mentioning here that Higgs signal mass means the mean of the Crystal Ball distribution! Not the observable!

(submit the source code, the plots and all relevant information you got)

Exercise CMS- Higgs Discovery 2012, Observation of a new boson at a mass of 125 GeV with theCMS experiment at the LHC



Inspired by Figure 4 “Distribution of the four-lepton invariant mass” of “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC” arXiv:1207.7235v2 [hep-ex] 28 Jan 2013

https://arxiv.org/abs/1207.7235 , https://inspirehep.net/literature/1124338

Root files have been processed from the Higgs analysis example on the CMS 2011-2012 Open Data .

Data source was https://github.com/cms-opendata-analyses/HiggsExample20112012.

In 2012, the Compact Muon Solenoid (CMS) experiment at the LHC presented results from searches for the standard model Higgs boson in proton-proton collisions at sqrt(s) = 7 and 8 TeV. The search is performed in five decay modes: gamma gamma, ZZ, WW, tau tau, and b b-bar. An excess of events is observed above the expected background at a mass near 125 GeV, signalling the production of a new particle compatible with standard model Higgs boson.

This exercise aims to reproduce the Figure 4 representing the H → ZZ → 4 leptons decay mode

PART 1 – COMPOSITE MODEL

Define the four lepton invariant mass distribution as an observable. Set the range, binning and units as in the final plot-

Download:

_cms_higgs_data.txt	real data
_cms_higgs_TTbarto4l.txt	top contribution (scaled to the MC generator cross section).
_cms_higgs_DYto4l.txt	Drell Yan (Z/γ*) contribution (scaled to the Z peak)
_cms_higgs_ZZto4l.txt	ZZ contribution (scaled to describe the data in the independent mass range 180-600 GeV
~~_cms_higgs_HZZto4l.txt~~	~~Higgs to ZZ to 4 leptons contribution~~

All files contain binned data. The file format is [BIN-CENTER] [BIN WEIGHT].

The Background histograms are already scaled to the final distribution

The first (last) record is an underflow (overflow) bin and is outside the observable range. They should not be included.

Now, we want to build a composite model using the histograms stored in the files.

The final model is like:

```
model(x) = f_s * sign(x) + (1 – f_s) * bkg(x)
```

where

- f_s the fraction of signal
- sign (x) the H → ZZ → 4 leptons distribution
- bkg(x) the sum of background components to 4leptons from

The **background** is a sum of 3 components ttbar, DY and ZZ, and their distribution are stored in files.

Read all the **binned** dataset and fill a RooDataHist for each file:

Example code:

```
RooDataHist data{"data", "data", x};
ifstream file("filename.txt");
double val, weight;
while (!file.eof()) {
     file >> val >> weight;
     x.setVal(val);
     data.set(x weight);  // (*)
}
```

Hint:
Range and binning of the observable shall be defined in such a way to match the histogram of the input file

Use: RooRealVar::setBins(…)

(*) Don't import underflow (overflow) bin

Then, create a P.D.F. for each background component:

Hint: have a look to RooHistPdf in https://root.cern/doc/master/rf706__histpdf_8C_source.html

Build a composite model for the background only, such as

```
bkg(x) = f_ttbar * ttbar(x) + f_drly * drly(x) + (1- f_ttbar.- f_drly) * zz(x)
```

We want the fractions of the 3 background components to be fixed relative to one another.

Given that the input histograms are already scaled from the input files, f_ttbar and f_drly shall be set proportional to the relative weight of their RooDataHist integrals over the observable range.

Hint: use the method sum(kTRUE) of RooDataHist to get the integral over the observable range.

As approximation set a **signal** pdf using a Guassian distribution.

Let only the mean (mass of Higgs) be a floating parameter and fix the width to 3 GeV.

Fit the model to the data. Only f_s and the mass of the Higgs are floating parameters.

Save the fit results to an object RooFitResults.

Plot the data and the fitted model.

Plot also the signal component in red and the total bkg component in black.

Write the best fit parameters on the plot too. (hint: paramOn(…) method of RooAbsPdf).

PART 2: use a RooStats ProfileLikelihoodCalculator for a confidence interval.
Compute the 95% confidence interval for Higgs mass

The calculator considers systematics by eliminating nuisance parameters with the profile likelihood. This is equivalent to the method of MINOS (the algorithm of MINUIT!).

Specify components of model for statistical tools

Hint: create a model for statistical tools (ModelConfig), a model config must be defined inside a RooWorkspace

```
RooWorkspace w;
RooStats::ModelConfig mc("model_config", &w);
```

Set the model, the observable, the Higgs mass as Parameter of Interest and f_s as nuisance parameter.

Compute the 95% confidence interval for Higgs mass with a Profile Likelihood Calculator.

Plot the interval.

Hints: Look at the code in the slides. You may refer to $ROOTSYS/tutorials/roostats/StandardProfileLikelihoodDemo.C to guide your work or you can just run StandardProfileLikelihoodDemo.C by passing your workspace as argument (recommended).

PART 3. Higgs mass Feldman Cousin interval

- Compute a 90% C.L interval on the **Higgs mass**, using RooStats Feldman Cousin Calculator.
- Try to change the confidence level.

> *Hint: You may refer to $ROOTSYS/tutorials/roostats/StandardFeldmanCousinsDemo.C to guide your work or you can just run StandardFeldmanCousinsDemo.C with your workspace as argument (recommended).*

PART 4: use a RooStats ProfileLikelihoodCalculator for a Hypothesis test
Compute the p-value and its corresponding significance for Higgs discovery

The variable fraction of signal (f_s) is somehow the **signal strength** in this composite model.

To establish a discovery one tries to reject the background only hypothesis corresponding to f_s = 0.

~~Change the model config and set f_s as Parameter of Interest and the Higgs mass as nuisance parameter.~~

Compute Significance and P-VALUE.

The null hypothesis (i.e. the signal peak is a statistical fluctuation of the flat background) corresponds to f_s = 0, i.e. a null value for the parameter of interest.

> *Hints:*
> *(method 1) see the slides and evaluate the significance using the ProfileLikelihoodCalculator.*
>
> *(method 2) refer to $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C to guide your work*
>
> *(method 3, recommended) Run the script $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C with your workspace as the argument. You don't need to manually define a ModelConfig for the null hypothesis—it's automatically set by assigning the parameter of interest (signal strength, μ) to zero. Use* ***Calculator Type 2*** *("Asymptotic Calculator") and* ***Test Type 3*** *("Profile Likelihood One-Sided"). Avoid* ***Calculator Type 0*** *("Frequentist"), as it's impractical for this level of significance.*

PART 5. Compute significance (p-value) as function of the signal mass

As an optional exercise in RooStats, we'll explore how significance (or p-value) varies with the signal mass hypothesis in a Gaussian signal plus exponential background model

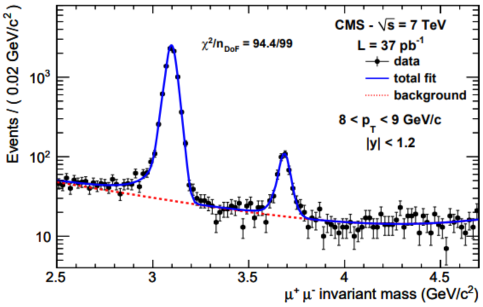
Using the AsymptoticCalculator, we'll test several mass points, plot the p-value for the null hypothesis (p0), and estimate the expected significance as a function of mass. The expected significance can be calculated with AsymptoticCalculator::GetExpectedPValues, using observed p-values for the null and alternate models. Refer to the Asymptotic formulae paper for details.

> *Hints: you can follow the solution described in Exercise 6b from*
>
> *https://twiki.cern.ch/twiki/bin/view/RooStats/RooStatsExercisesMarch2015#Exercise_6b_Compute_significance*
>
> *It is worth mentioning here that Higgs signal mass means the mean of the Gaussian distribution! Not the observable!*
>
> *(submit source code, plots, and results as text file)*

Exercise CMS.jpsi J/ψ and ψ(2S) production in pp collisions at √s = 7 TeV

Inspired by JHEP02(2012)011

https://link.springer.com/content/pdf/10.1007/JHEP02(2012)011.pdf
arXiv:1111.1557v1 [hep-ex] 7 Nov 2011 https://arxiv.org/abs/1111.1557



This exercise aims to reproduce the plot of Figure 1 (bottom) and then compute the excess significance in the region of 3.65 GeV, compute the interval (or an upper limit) to the ψ(2S) signal yield , calculate the significance using RooFit and RooStats.

PART 1. MODEL, FIT, PLOT

Download MuRun2010B.csv

This file, sourced from the CERN Open Data Portal, contains 100k dimuon events selected from the CMS Mu dataset (Run2010B) http://opendata.cern.ch/record/700 .

Events include exactly two muons with an invariant mass between 2–110 GeV. Each line represents an event, with the format detailed in the first line:

```
Run,Event,Type1,E1,px1,py1,pz1,pt1,eta1,phi1,Q1,Type2,E2,px2,py2,pz2,pt2,eta2,phi2,Q2,M
```

The observable is the invariant mass of the dimuon system, derived from other columns already present in the CSV file. Calculate the invariant mass for each event outside RooFit, then import the dataset as an unbinned RooDataSet in RooFit. Only include events where the rapidity (η) of both muons is less than 1.2

> *The following steps may be time-consuming depending on your laptop performance. In case of problems or if you simply prefer to be faster during the code prototyping, let's consider working with a subsample of data in order not to have a too high significance of your signal yield, which will then be difficult to estimate if we use pseudo-experiments*
>
> *Hints:*
> *You can find your own way or exploit RDataFrame, a modern C++ high-level interface for interacting with data in ROOT v6.18+. With RDataFrame a smallest dataset can be generated by calling RDataFrame::Range(…) .*

```
int create_dataset()
{
  auto tdf = ROOT::RDF::MakeCsvDataFrame("MuRun2010B.csv");
  tdf.Filter("eta1 < 1.2 && eta2 < 1.2")
      .Define("mass",
              "sqrt(pow(E1 + E2, 2) - (pow(px1 + px2, 2) + pow(py1 + py2, 2) + "
              "pow(pz1 + pz2, 2)))")
      .Range(0, 0, 3)
      .Snapshot("tree", "my_muons.root");
  return 1;
}
```

Now the filtered data with the additional column “mass” are available in a TTree stored in my_muons.root.

First, you may define the observable as a RooRealVar

```
RooRealVar mass{"mass", "#mu^{+}#mu^{-} invariant mass", 2., 6., "GeV"};
```

And then import the TTree (which is an unbinned dataset) to a RooDataSet.

```
auto f = TFile::Open("my_muons.root");
auto tree = static_cast<TTree*>(f->Get("tree"));
RooDataSet dataset("dataset", "dataset", RooArgSet(mass),
                   RooFit::Import(*tree));
dataset.Print();
```

Now, you must define a model for the invariant mass distribution. The p.d.f. has several components, as described in the section 4 "Inclusive yield determination" of the paper. They can be simplified as:

i) J/ψ → μμ peak, a Crystal Ball distribution
ii) ψ(2S) → μμ peak, a Crystal Ball distribution
iii) background, modelled by two exponentials.

Crystal Ball shape distribution is implemented by the RooCBshape class.

There are several constraints to be taken into account when you define the parameters:

- for J/ψ Crystal Ball: mean ranges from [2.7, 3.3], width from [0.00001,1.], alpha from [-10, 10] and the power-law's exponent n from [0.2, 10]
- for ψ(2S) Crystal Ball: mean ranges from [3.3, 3.9], while width, alpha and n are the same as J/ψ peak.
- No constraints exist for the exponential distributions.

More realistic constraints are described in the paper, but we can neglect them.

Define the expected yields for the different components of the spectrum (njpsi, npsi2s, njpsi, njpsi).

With the Extended Likelihood Formalism build an Extended Composite Model by using RooAddPdf or the factory.

Fit the model to the data using a maximum likelihood fit.

Plot data and model.

Import dataset and model (automatically all related elements) into a workspace.

Define the Model Config for the signal_plus_background using the ψ(2S) signal yield as parameter of interest, the remaining floating parameters are nuisance.

Hints: (w is the workspace)

```
 RooStats::ModelConfig mc("sig_plus_bkg", &w);
 mc.SetPdf(...
 mc.SetParametersOfInterest(...
 mc.SetObservables(...
 // define set of nuisance parameters
w.defineSet("nuisance_params",
      "NJpsi,Nbkg,a1,a2,a3,alphaJpsi,meanJpsi,meanpsi2S,nJpsi,sigmaJpsi");
 mc.SetNuisanceParameters(*w.set("nuisance_params"));
 mc.SetSnapshot(Npsi);
```

Import the ModelConfig into a workspace and save it to a file.

PART 2. ψ(2S) signal yield interval

Compute a 95% interval on the **ψ(2S) signal yield**, using RooStats Profile Likelihood Calculator.

Hints: Look at the code in the slides. You may refer to $ROOTSYS/tutorials/roostats/ StandardProfileLikelihoodDemo.C to guide your work or you can just run StandardProfileLikelihoodDemo.C by passing your workspace as argument (recommended).

PART 3. Compute the significance (Hypothesis Test)

To estimate significance, we perform a hypothesis test to reject the null (background-only) model in favor of the alternate (signal-plus-background) model. In RooStats, this involves defining two ModelConfig objects: one for the null and one for the alternate model.

The null hypothesis *(i.e. the* ψ(2S) *peak would be a statistical fluctuation of the background*) *corresponds to a* ψ(2S) signal yield npsi2s *= 0.*

> *Hints:*
> *(method 1) see the slides and evaluate the significance using the ProfileLikelihoodCalculator.*
>
> *(method 2) refer to $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C to guide your work*
>
> *(method 3, recommended) Run the script $ROOTSYS/tutorials/roostats/StandardHypoTestDemo.C with your workspace as the argument. You don't need to manually define a ModelConfig for the null hypothesis—it's automatically set by assigning the parameter of interest (signal strength, μ) to zero. Use **Calculator Type 2** ("Asymptotic Calculator") and **Test Type 3** ("Profile Likelihood One-Sided"). Avoid **Calculator Type 0** ("Frequentist"), as it's impractical for this level of significance.*

PART 4. Compute significance (p-value) as function of the signal mass

As an optional exercise in RooStats, we'll explore how significance (or p-value) varies with the signal mass hypothesis in a Gaussian signal plus exponential background model

Using the AsymptoticCalculator, we'll test several mass points, plot the p-value for the null hypothesis (p0), and estimate the expected significance as a function of mass. The expected significance can be calculated with AsymptoticCalculator::GetExpectedPValues, using observed p-values for the null and alternate models. Refer to the Asymptotic formulae paper for details..

> *Hints: to run this part in RooStats you have explicitly define the ModelConfig object for background only model:*

```
auto mcb = mc.Clone();
mcb->SetName("bkg_only");
Npsi.setVal(0);
mcb->SetSnapshot(Npsi);
w.import(*mcb);
```

> *import the background-only model config object to the workspace and save it to file.*
>
> *Hints: you can follow the solution described in Exercise 6b from*
>
> *https://twiki.cern.ch/twiki/bin/view/RooStats/RooStatsExercisesMarch2015#Exercise_6b_Compute_significance*
>
> *It is worth mentioning here that signal mass means the mean of ψ(2S) Crystal Ball distribution! Not the observable!*
>
> *(submit source code, plots, and results as text file)*