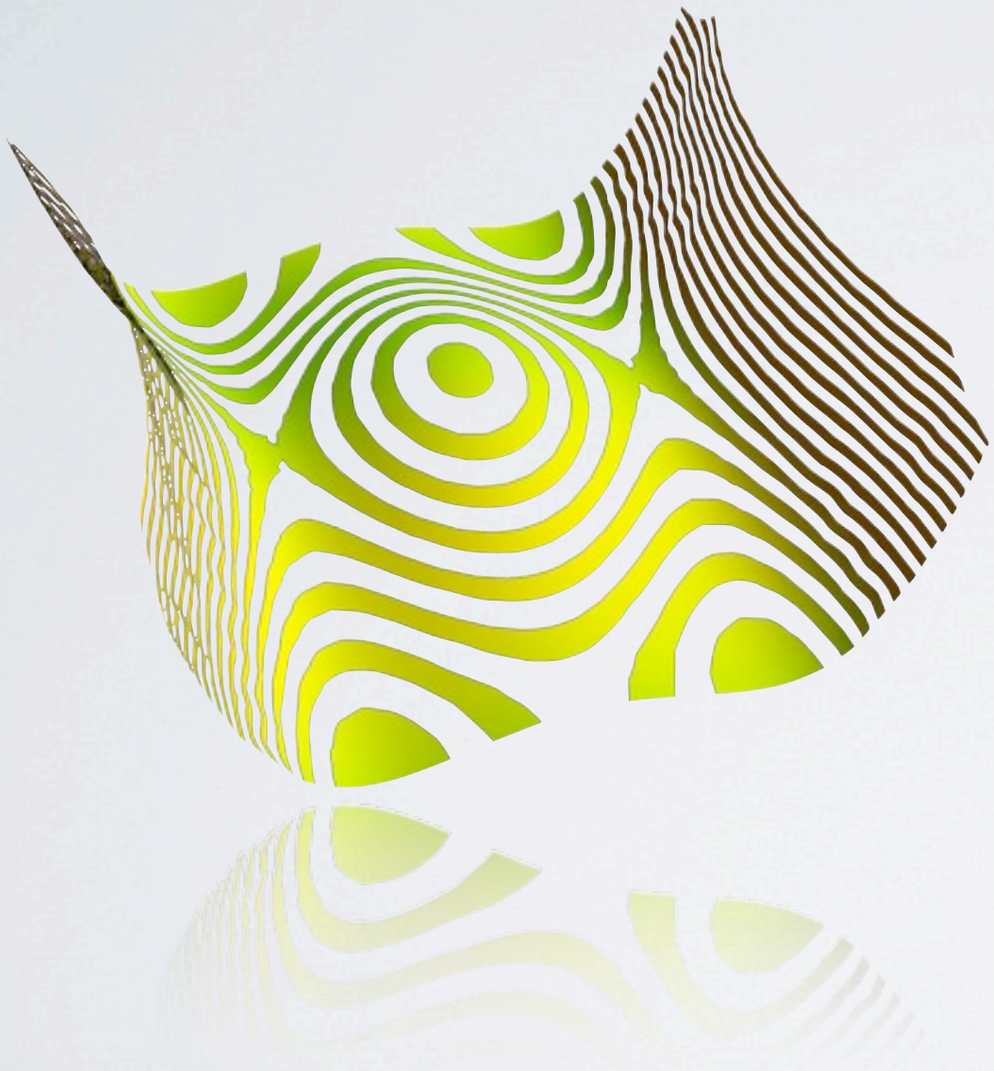


NEW YORK UNIVERSITY

RooFit/RooStats Tutorial

Statistics School

Hamburg
April 2-5, 2012

Sven Kreiss (NYU)
Lorenzo Moneta (CERN)

From Wouter:

Constructing joint pdfs

- Operator class SIMUL to construct joint models at the pdf level

```
// Pdfs for channels 'A' and 'B'
w.factory("Gaussian::pdfA(x[-10,10],mean[-10,10],sigma[3])") ;
w.factory("Uniform::pdfB(x)") ;

// Create discrete observable to label channels
w.factory("index[A,B]") ;

// Create joint pdf
w.factory("SIMUL::joint(index,A=pdfA,B=pdfB)") ;
```

- Can also construct joint datasets

```
RooDataSet *dataA, *dataB ;
RooDataSet dataAB("dataAB","dataAB",Index(w::index),
                  Import("A",*dataA),Import("B",*dataB)) ;
```

Using joint models

- When constructing joint models and likelihoods:

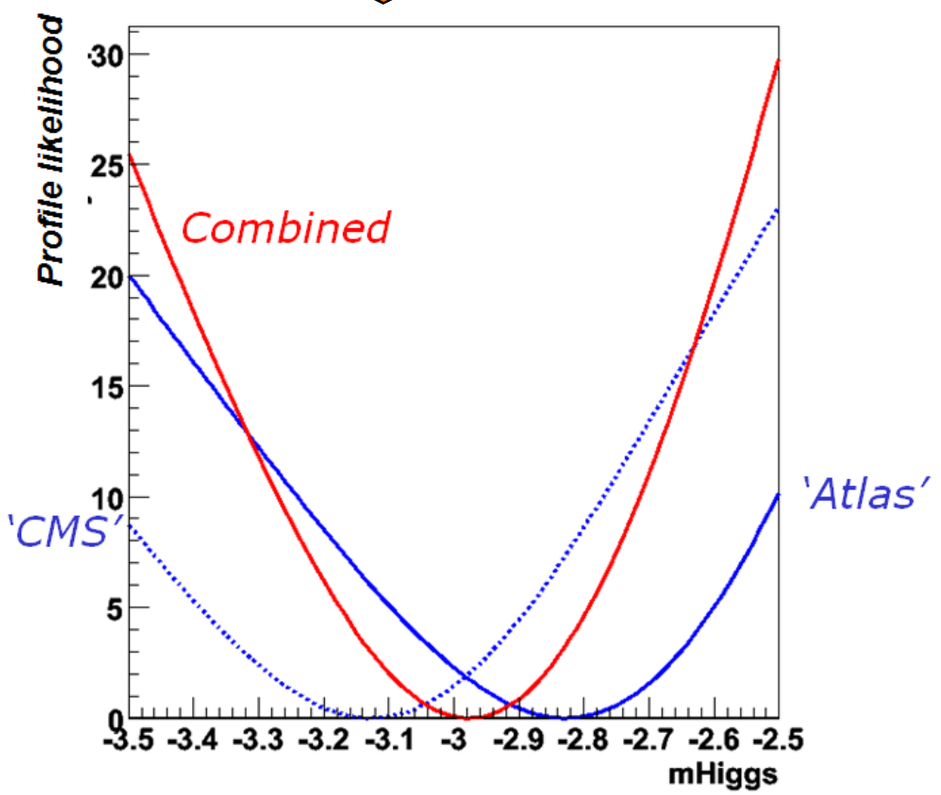
  parameters with the same name = same parameter

- If intentional, you are done at this point.

```
RooAbsReal* pllJoint = nllJoint->createProfile(paramOfInterest) ;
```

  - Takes all parameter correlations fully into account
  - To add additional correlations, simply multiply joint pdf with appropriate **`RooMultiVarGaussian`** pdf in parameters of choice

```
w.factory("MultiVarGaussian::corr
                ({a,b},{0,0},COV)");
w.factory("PROD::jointc(joint,corr)") ;
```



STANDARD EXAMPLE

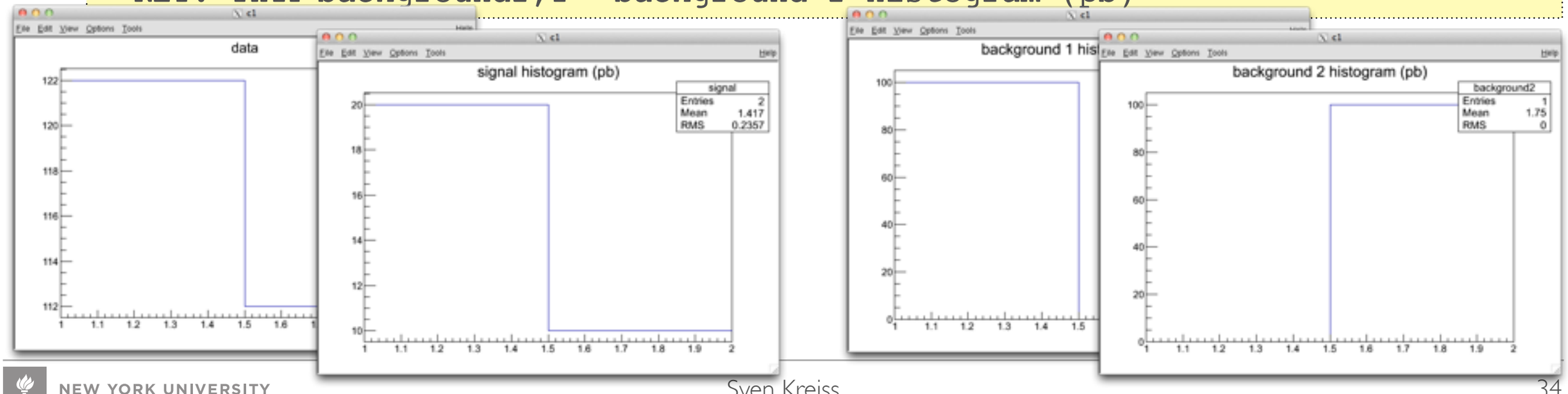
Creating the Example

- go to an empty directory

```
[lxplus] ATLASDanalysis > prepareHistFactory

[lxplus] ATLASDanalysis > ls
config data result
```

- What is in data?

```
[lxplus] ATLASDanalysis > root -l data/example.root
root [0]
Attaching file example.root as _file0...
root [1] .ls
TFile**     example.root
 KEY: TH1F data;1  data
 KEY: TH1F signal;1    signal histogram (pb)
 KEY: TH1F background1;1   background 1 histogram (pb)
 KEY: TH1F background2;1   background 2 histogram (pb)
```



HistFactory Models

Data: think of it as data points in a histogram

Model: looks the same (it is also a histogram), but one should think about it as a shape (a PDF) that is extended with the number of events in the histogram.

- ➡ x_i events in bin i really means: probability of an event in this bin is $x_i / \Sigma_j x_j$ and the PDF is extended with $\Sigma_j x_j$ (for bins with equal width).
- ➡ If there is only one bin, this reduces to “number counting form”.

From the HistFactory User Guide:

shapes (integral is one)

$$\mathcal{P}(\{x_1 \dots x_n\}|\mu) = \mathrm{Pois}(n|\mu S + B) \left[\prod_{e=1}^{n} \frac{\mu S f_{\mathrm{S}}(x_e) + B f_{\mathrm{B}}(x_e)}{\mu S + B} \right]$$

total number of signal (S) and background (B) events including “signal strength modifier” μ

Example Channel

- config/example_channel.xml

zween — 112×33

ssh | bash | bash ...

```
<!--
    Single channel configuration exampple.
    The top level configuration XML is example.xml

    NormalizedByTheory should be "True" (not "TRUE" or "true") for all non-data-driven backgrounds.

    If you comment or remove the <Data> tag then it will use the expected data.

    Histogram inputs should be in pb and in top-level xml the lumi should be in 1/pb
    (The important thing is that they match... fb and 1/fb is also ok)

    Note: Config.dtd needs to be accessible.  It can be found in ROOT release area.
    The file system path is relative to location of this XML file, not the executable.
-->

<!DOCTYPE Channel  SYSTEM 'HistFactorySchema.dtd'>

  <Channel Name="channel1" InputFile="./data/example.root" HistoName="" >
    <Data HistoName="data" HistoPath="" />
    <Sample Name="signal" HistoPath="" HistoName="signal">
      <OverallSys Name="syst1" High="1.05" Low="0.95"/>
      <NormFactor Name="SigXsecOverSM" Val="1" Low="0." High="3." Const="True" />
    </Sample>
    <Sample Name="background1" HistoPath="" NormalizeByTheory="True" HistoName="background1">
      <OverallSys Name="syst2" Low="0.95" High="1.05"/>
    </Sample>
    <Sample Name="background2" HistoPath="" NormalizeByTheory="True" HistoName="background2">
      <OverallSys Name="syst3" Low="0.95" High="1.05"/>
      <!-- <HistoSys Name="syst4" HistoPathHigh="" HistoPathLow="histForSyst4"/>-->
    </Sample>
  </Channel>
~
"example_channel.xml" 31L, 1424C                                    12,5          All
```

file we just looked at

names of the histograms from previous page

Example Model

- config/ example.xml

zween — 117×46

ssh | bash | bash | ...

```
<!--
//-------------------------------------------------------------------------------
// Name        : example.xml
//-------------------------------------------------------------------------------
-->

<!--
    Top-level configuration,  details for the example channel are in example_channel.xml.
    This is the input file to the executable.

    Note: Config.dtd needs to be accessible.  It can be found in ROOT release area.
    The file system path is relative to location of this XML file, not the executable.
-->

<!DOCTYPE Combination  SYSTEM 'HistFactorySchema.dtd'>

<Combinati OutputFilePrefix="./results/example" ode="comb" >
  <Input>./config/example_channel.xml</Input>

  <Measurement Name="GaussExample" Lumi="1." LumiRelErr="0.1" BinLow="0" BinHigh="2" Mode="comb" >
    <POI>SigXsecOverSM</POI>
    <ParamSetting Const="True">Lumi alpha_syst1</ParamSetting>
    <!-- don't need <ConstraintTerm> default is Gaussian-->
  </Measurement>

  <Measurement Name="GammaExample" Lumi="1." LumiRelErr="0.1" BinLow="0" BinHigh="2" Mode="comb" >
    <POI>SigXsecOverSM</POI>
    <ParamSetting Const="True">Lumi alpha_syst1</ParamSetting>
    <ConstraintTerm Type="Gamma" RelativeUncertainty=".3">syst2</ConstraintTerm>
  </Measurement>

  <Measurement Name="LogNormExample" Lumi="1." LumiRelErr="0.1" BinLow="0" BinHigh="2" Mode="comb" >
    <POI>SigXsecOverSM</POI>
    <ParamSetting Const="True">Lumi alpha_syst1</ParamSetting>
    <ConstraintTerm Type="LogNormal" RelativeUncertainty=".3">syst2</ConstraintTerm>
  </Measurement>

  <Measurement Name="ConstExample" Lumi="1." LumiRelErr="0.1" BinLow="0" BinHigh="2" Mode="comb" ExportOnly="True">
    <POI>SigXsecOverSM</POI>
    <ParamSetting Const="True">Lumi alpha_syst1</ParamSetting>
  </Measurement>


</Combination>
                                                                              1,1           All
```

use that channel

do not run ProfileLikelihoodCalculator for this Measurement

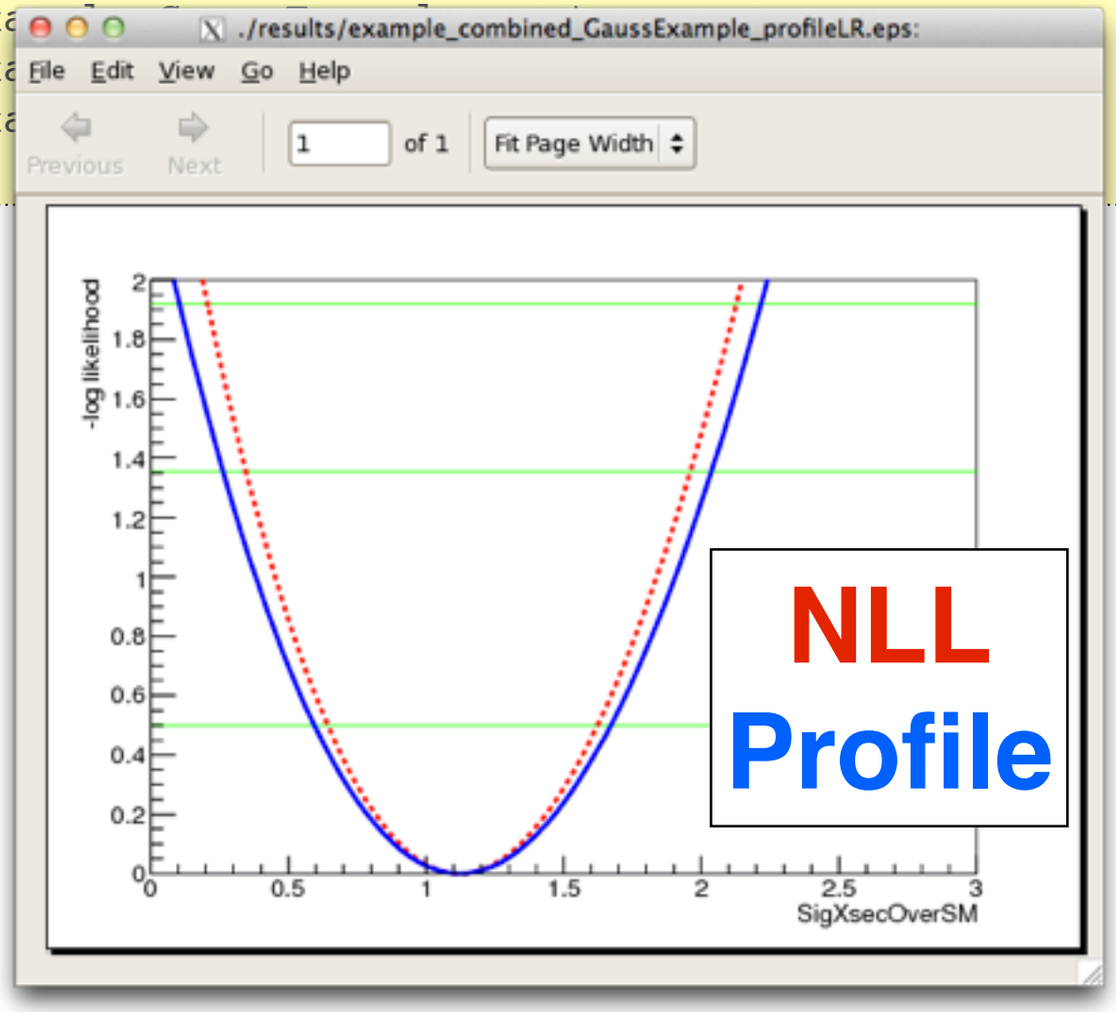
Running Example

- from the main directory:

```
[lxplus] ATLASDanalysis > hist2workspace config/example.xml
... producing a lot of output ...

[lxplus] ATLASDanalysis > ls results/
example_channel1_ConstExample_model.root        example_combined_GaussExample_model.root
example_channel1_GammaExample_model.root        example_combined_GaussExample_profileLR.eps
example_channel1_GammaExample_profileLR.eps     example_combined_LogNormExample_model.root
example_channel1_GaussExample_model.root        example_combined_LogNormExample_profileLR.eps
example_channel1_GaussExample_profileLR.eps     example_ConstExample.root
example_channel1_LogNormExample_model.root      example_GammaExample.root
example_channel1_LogNormExample_profileLR.eps   exa
example_combined_ConstExample_model.root        exa
example_combined_GammaExample_model.root        exa
example_combined_GammaExample_profileLR.eps
```

- this created many .root files and also .eps files

  ➡ eps files are the outputs of the ProfileLikelihoodCalculator that was run automatically (use ExportOnly="True" to switch that off)



Look at result

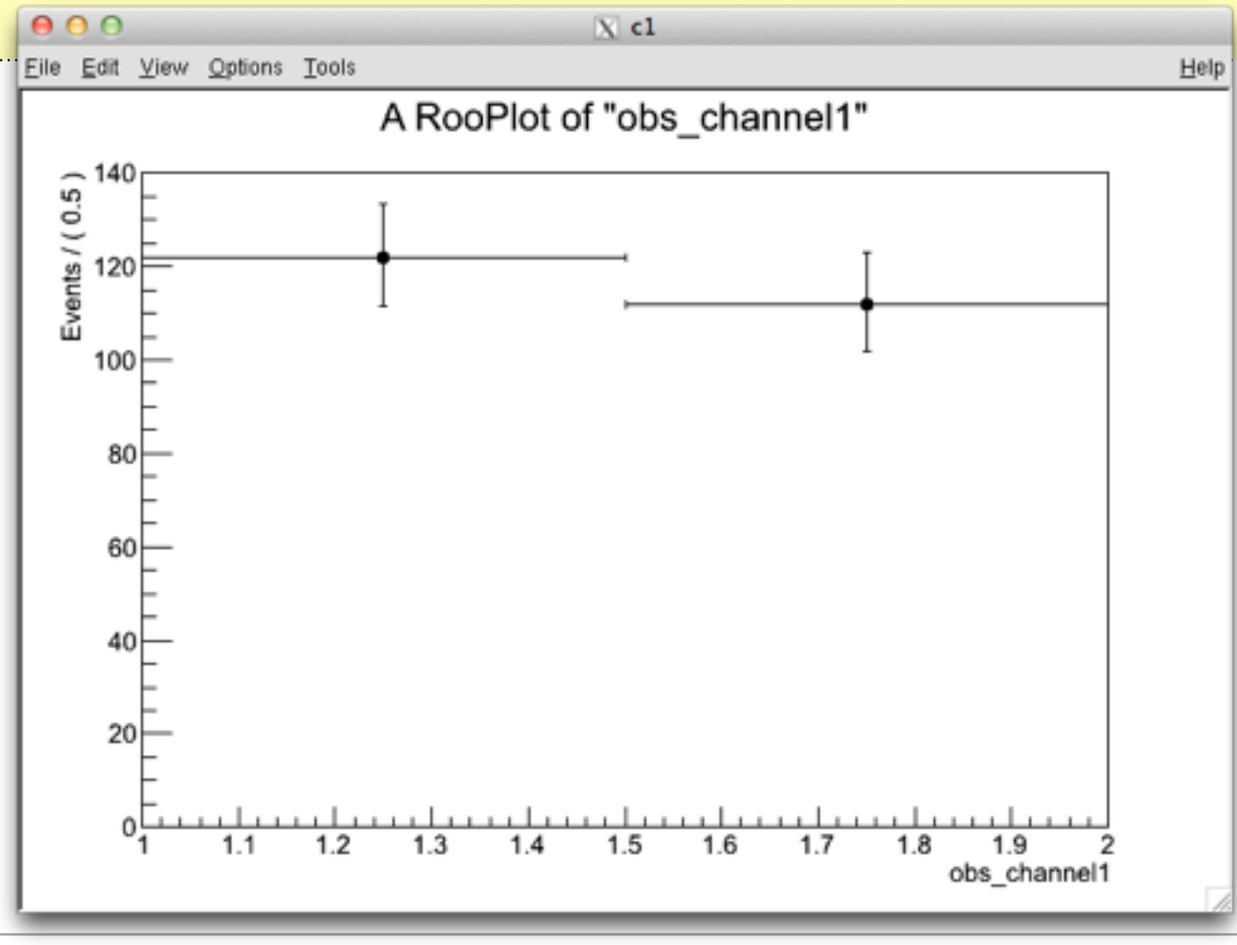
- Find out workspace name, model name and data name:

```
[lxplus] ATLASDanalysis > root -l results/
example_combined_GaussExample_model.root

root [1] .ls
TFile**     results/example_combined_GaussExample_model.root
 TFile*     results/example_combined_GaussExample_model.root
  KEY: RooWorkspace combined;1  combined
  KEY: TProcessID   ProcessID0;10c6e344e-2565-11e0-9717-ecd28a89beef


root [2] combined->Print()
RooWorkspace(combined) combined contents
... print out of variables, p.d.f.s, functions, named sets, and ...

datasets
--------
RooDataSet::asimovData(channelCat,obs_channel1)
RooDataSet::obsData(channelCat,obs_channel1)

generic objects
---------------
RooStats::ModelConfig::ModelConfig

root [3]
```

Reading obsData

- in the standard form, the model is built using RooHistFuncs, which is more efficient than the number counting form
  - ➡ look at data like this:

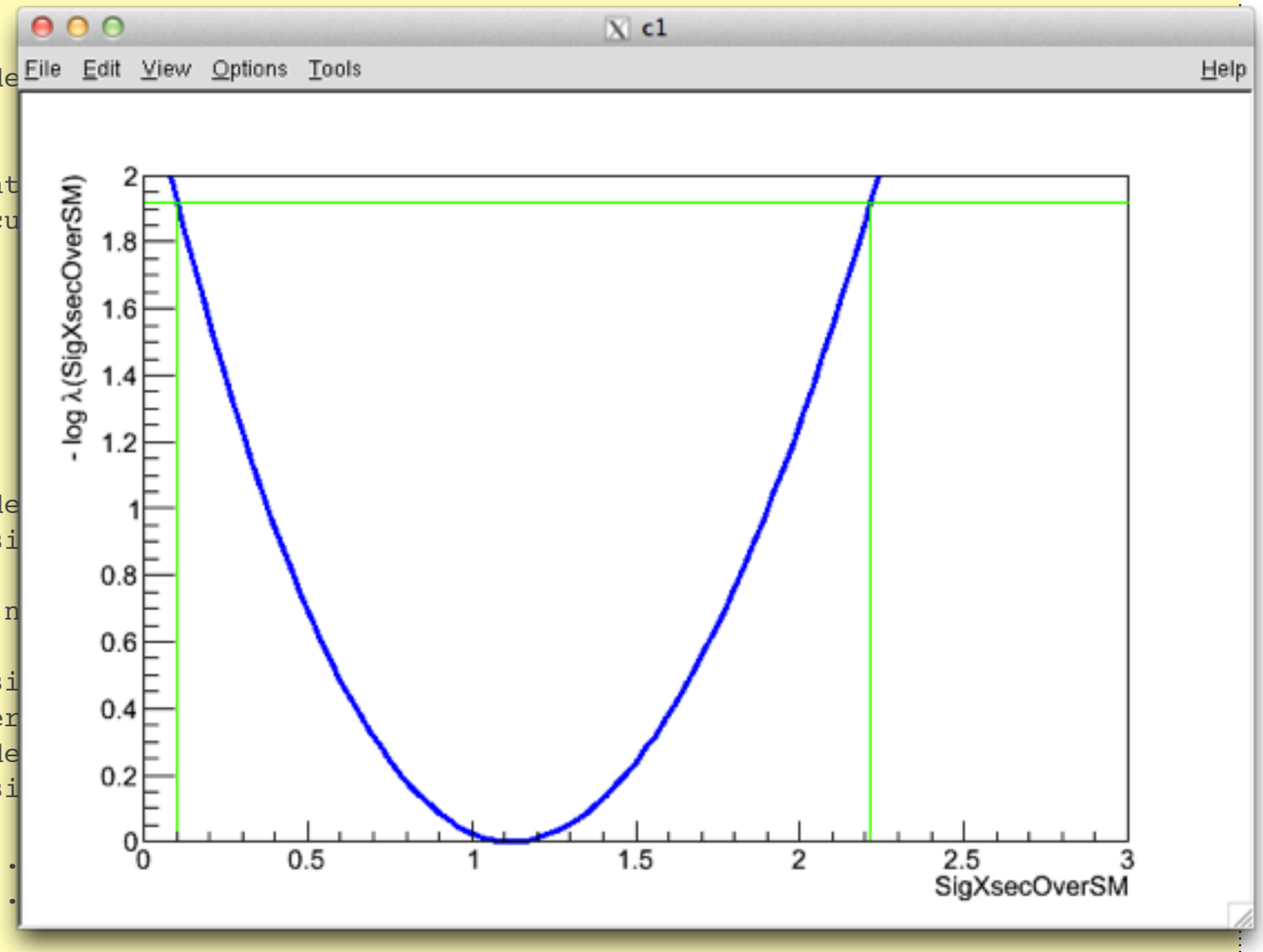
```
root [2] f = combined->var("obs_channel1")->frame()
(const class RooPlot*)0x7fc173162800
root [3] combined->data("obsData")->plotOn(f)
(const class RooPlot*)0x7fc173162800
root [4] f->Draw()
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [5]
```



Using a HistFactory model

```
root [5] .x /afs/cern.ch/sw/lcg/app/releases/ROOT/5.30.01/x86_64-slc5-gcc43-opt/root/tutorials/
roostats/StandardProfileLikelihoodDemo.C("results/example_combined_GaussExample_model.root",
"combined", "ModelConfig", "obsData")

[#1] INFO:Minization --  Including the following contraint terms in minimization:
(alpha_syst2Constraint,alpha_syst3Constraint)
ProfileLikelihoodCalculator::DoGlobalFit - using Minuit / Migrad with strategy 1
[#1] INFO:Minization --  Including the following contraint terms in minimization:
(alpha_syst2Constraint,alpha_syst3Constraint)
[#1]                                          rLevel(nll_simPdf_simData_with_constr) Summation contains a RooNLLVar, using
its e
[#1]                                          eConst:
[#1] INFO:Fitting -- RooAbsTestStatistic::initSimMode
[#1] INFO:Minization -- RooMinimizer::optimizeConst:

  RooFitResult: minimized FCN value: 8.44132, estimat
                covariance matrix quality: Full, accu

    Floating Parameter    FinalValue +/-  Error
  --------------------  --------------------------
         SigXsecOverSM    1.1212e+00 +/-  5.26e-01
           alpha_syst2   -1.3646e-02 +/-  9.75e-01
           alpha_syst3    2.7826e-02 +/-  9.19e-01

[#1] INFO:Fitting -- RooAbsTestStatistic::initSimMode
[#1] INFO:Minization -- RooProfileLL::evaluate(nll_si
MINUIT
[#1] INFO:Fitting -- RooAddition::defaultErrorLevel(n
its error level
[#1] INFO:Minization -- RooProfileLL::evaluate(nll_si
likelihood for current configurations w.r.t all obser
[#1] INFO:Fitting -- RooAbsTestStatistic::initSimMode
[#1] INFO:Minization -- RooProfileLL::evaluate(nll_si
(SigXsecOverSM=1.12102)
.....................................................
.....................................................
default TCanvas with name c1

95% interval on SigXsecOverSM is : [0.102174, 2.21605]
root [6]
```

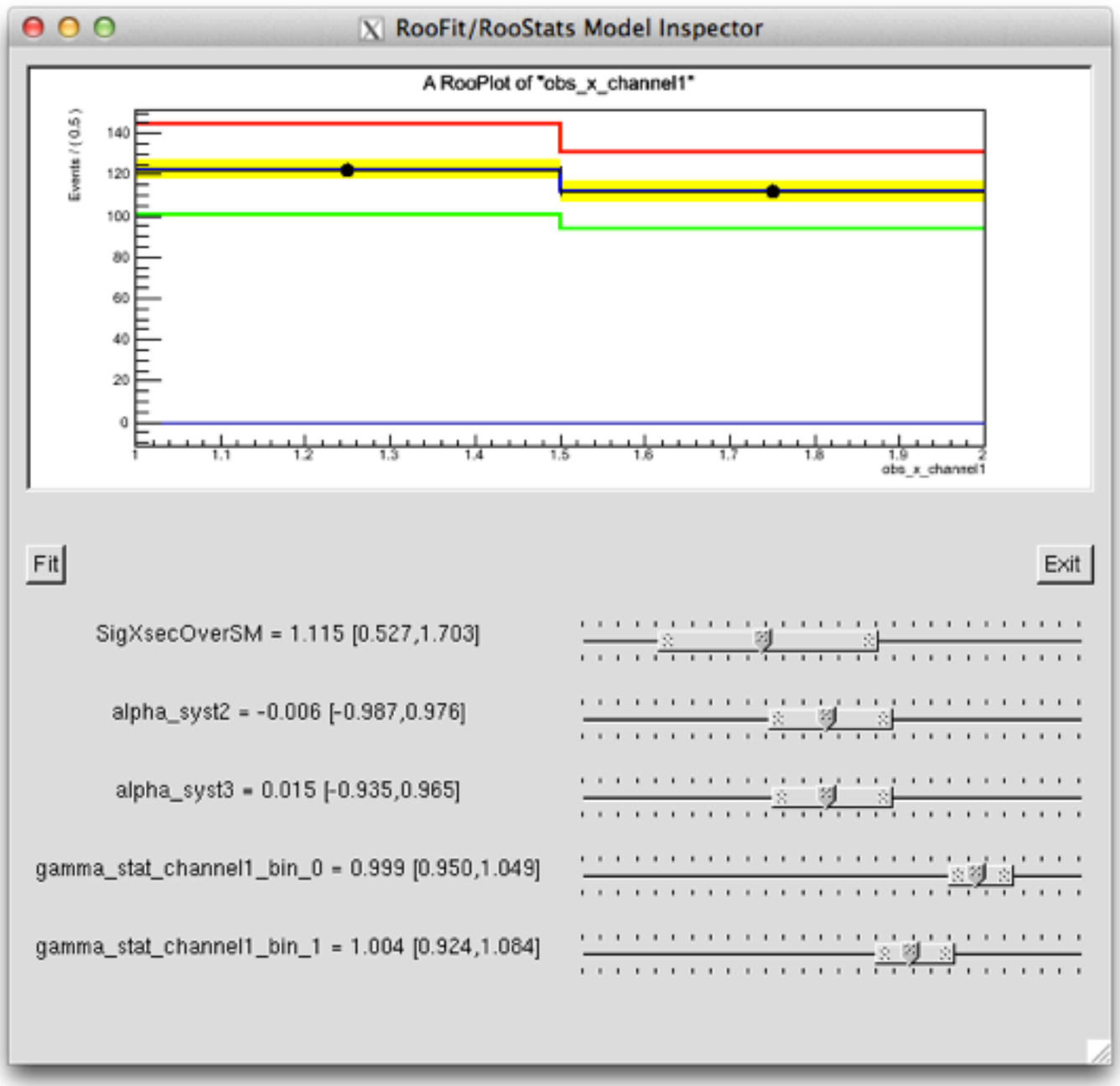
from previous pages



Exercise: ModelInspector.C

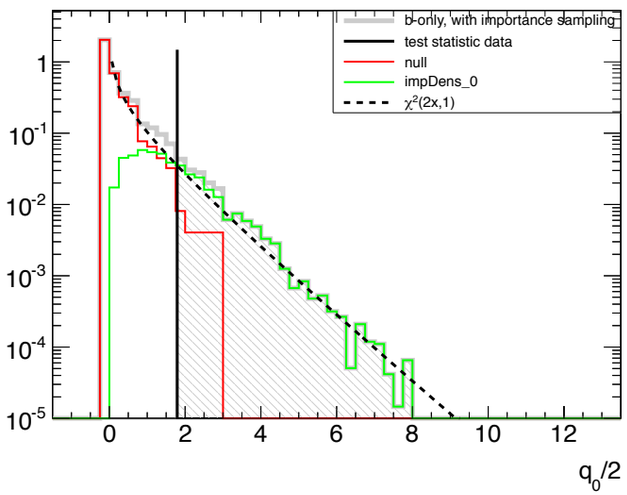
```
root -l '$ROOTSYS/tutorials/roostats/ModelInspector.C+("results/
example_combined_GaussExample_model.root","combined","ModelConfig"
,"obsData")'
```

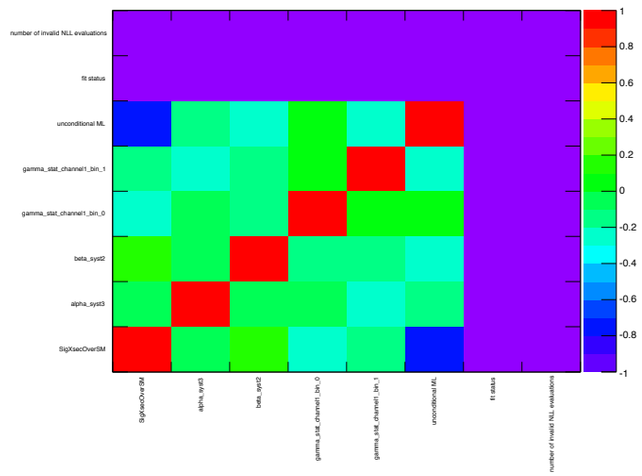
and click on ‘Fit’.

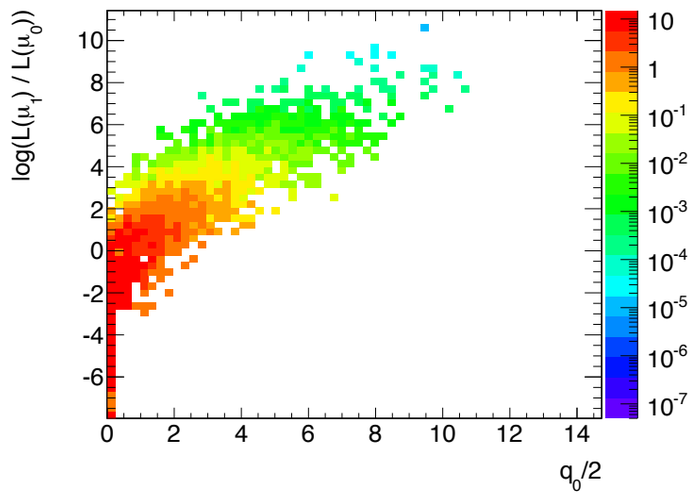


New in 5.33 / 5.34

ToyMCSampler: Detailed output (nuisance parameter distributions, correlations, status bits, ...) and runs with multiple test statistics. Importance Sampling.







HistFactory will be configurable directly from C++ and Python without XML:

- branches/dev/roostats/tutorials/histfactory/example.C
- branches/dev/roostats/tutorials/histfactory/example.py

MultiNest (another Bayesian method) is in preparation for 5.34.

RooStats was accepted as mentoring organization for *Google Summer of Code*. Some projects might be of interest to you.

Summary

- RooStats supports many methods for discovery and limits (asymptotics, toys, Bayesian)
  - ➡ is powerful (used for real-world complicated analyses and combinations, see ttbar combination on top right)
  - ➡ proven in individual search channels (e.g. various Higgs and SUSY channels), top physics and Higgs combinations
- separates model, method and storage: each is improving all the time
- **use it, report bugs ... and if you need more, contribute a new tool**
- RooStats has many more tools:
  - ➡ Importance Sampling for all toy based tools
  - ➡ Jeffrey's Prior, Keys PDF, Bernstein Correction, SPlot, ...