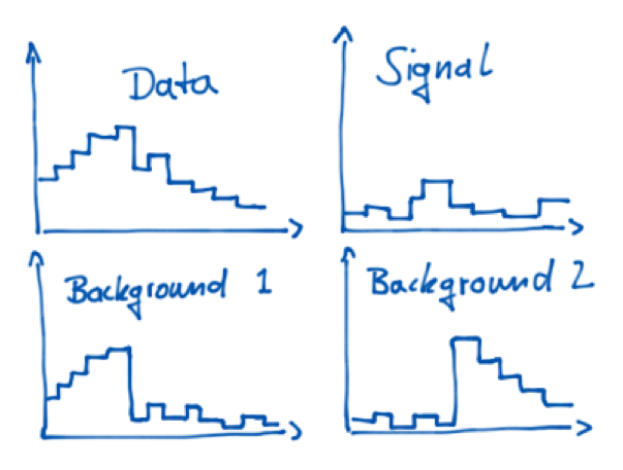
HistFactory

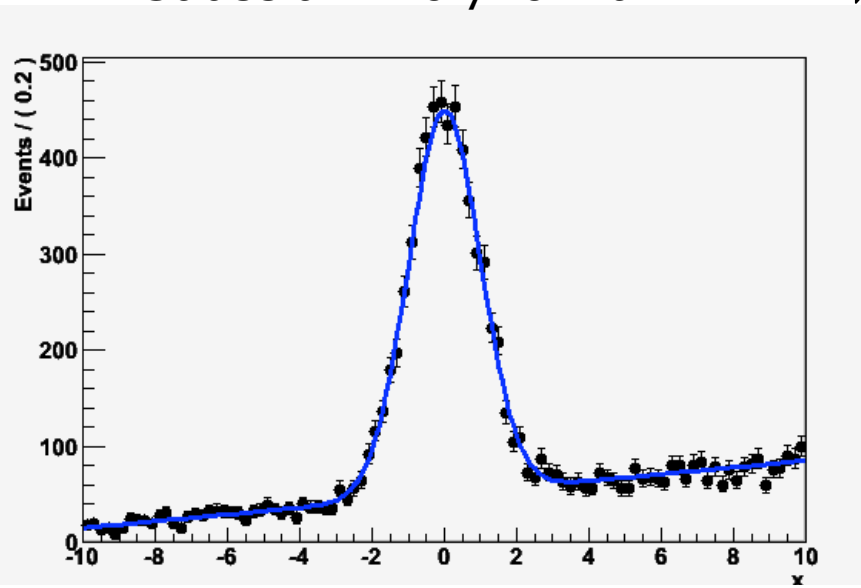


*see also HistFactory doc (*https://cdsweb.cern.ch/record/1456844/files/CERN-OPEN-2012-016.pdf*)*

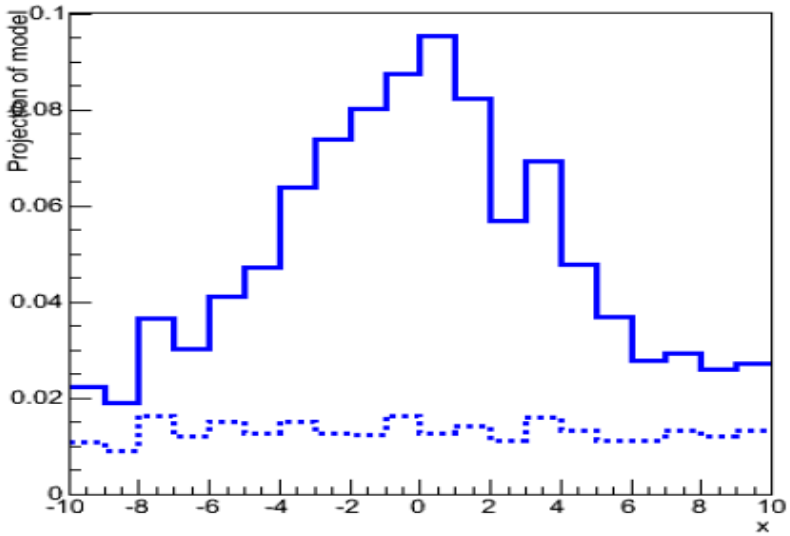
HistFactory – a new class of pdfs

- Focus of RooFit traditionally on analytical models
  - Assumes you can formulate signal/background in an analytical form
  - Often possible in e+e- experiments,
    shapes for hadron colliders cumbersome

Analytical form:
Gaussian+Polynomial



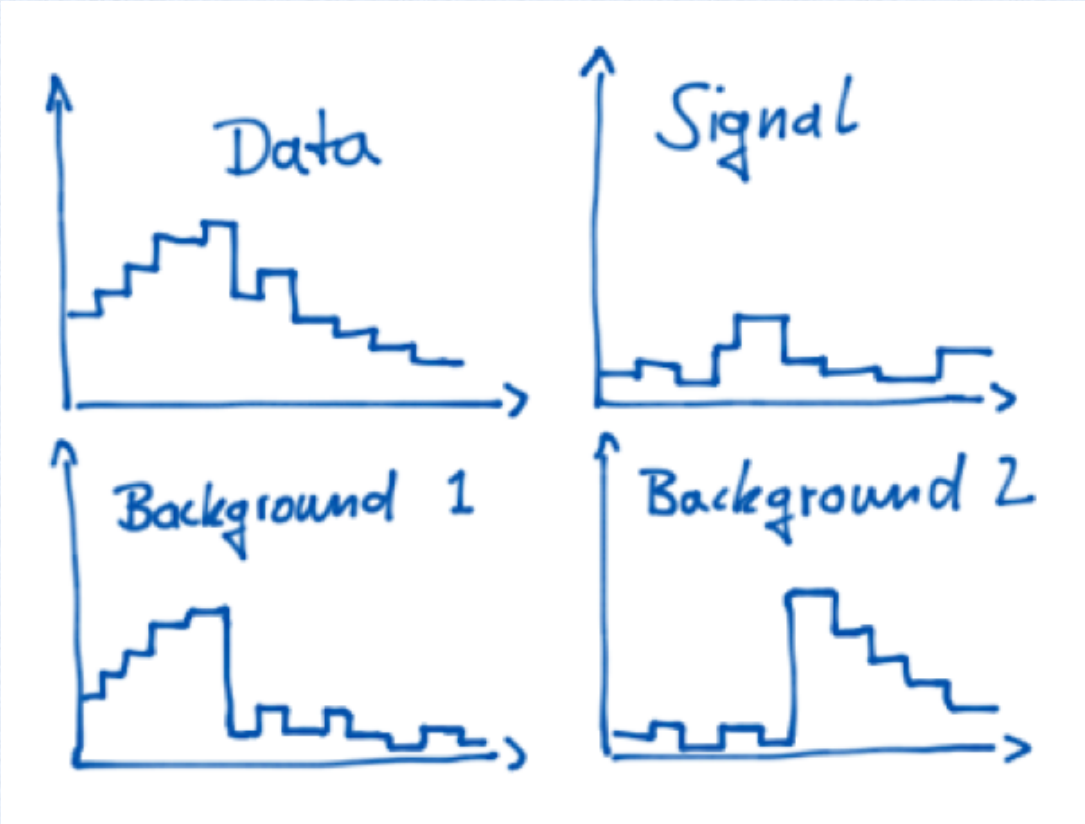
Template form:
Histogram (discrete)



K. Cranmer, G. Lewis, L. Moneta, A. Shibata, and W. Verkerke, *HistFactory: A tool for creating statistical models for use with RooFit and RooStats*, CERN-OPEN-2012-016 (2012).
http://cdsweb.cern.ch/record/1456844.

Model Building with HistFactory

- Tool to build models from input histograms



RooFit
Workspace

HistFactory concept

- Measurement
  - used to give global description of the model
  - can contain one or several channels
- Channel
  - disjoints selected regions of events
- Sample
  - set of process contributions to a channel

HistFactory

- Generalization of number counting models

$$\mathcal{P}(n_b|\mu) = \text{Pois}(n_{\text{tot}}|\mu S + B) \left[\prod_{b \in \text{bins}} \frac{\mu \nu_b^{\text{sig}} + \nu_b^{\text{bkg}}}{\mu S + B} \right]$$

where n_b is the data histogram

in general HistFactory produces model of this form

$$\mathcal{P}(n_c, x_e, a_p \,|\, \phi_p, \alpha_p, \gamma_b) = \prod_{c \in \text{channels}} \left[\text{Pois}(n_c|\nu_c) \prod_{e=1}^{n_c} f_c(x_e|\boldsymbol{\alpha}) \right] \cdot G(L_0|\lambda, \Delta_L) \cdot \prod_{p \in \mathbb{S}+\boldsymbol{\Gamma}} f_p(a_p|\alpha_p)$$

p.d.f — luminosity constraint — parameter constraint

$$f_c(x_e|\phi_p, \alpha_p, \gamma_b) = \frac{\nu_{cb_e}}{\nu_c}$$

expected events/bin

$$\nu_c = \sum_{b \in \text{bins of channel } c} \nu_{cb}$$

HistFactory Model

$$\mathcal{P}(n_{cb}, a_p \,|\, \phi_p, \alpha_p, \gamma_b) = \prod_{c\in\text{channels}} \prod_{b\in\text{bins}} \text{Pois}(n_{cb}|\nu_{cb}) \cdot G(L_0|\lambda, \Delta_L) \cdot \prod_{p\in\mathbb{S}+\mathbf{\Gamma}} P_p(a_p|\alpha_p)$$

expected number of events in a bin

luminosity constraint

parameter constraint

$$\nu_{cb}(\phi_p, \alpha_p, \gamma_b) = \lambda_{cs}\, \gamma_{cb}\, \phi_{cs}(\boldsymbol{\alpha})\, \eta_{cs}(\boldsymbol{\alpha})\, \sigma_{csb}(\boldsymbol{\alpha})$$

λ_{cs}	luminosity parameter for each sample of a channel
γ_{cb_e}	bin by bin scale factor (statistical + systematics)
$\phi_{cs} = \prod_{p\in\mathbb{N}_c} \phi_p$	product of unconstrained normalisation. Depend on P.O.I. (e.g. signal rate)
$\eta_{cs}(\boldsymbol{\alpha})$	normalisation uncertainty for each sample of a channel
σ_{csb_e}	nominal bin content and its uncertainty (from input histograms)

HistFactory Capabilities

- HistFactory can include:
  - multiple channels and samples
  - unconstrained normalisation for any sample
  - parametrize variation in normalization due to systematic effects
  - bin by bin statistical uncertainty (overall for all samples)
  - parametrize systematic variation of a single bin

	Constrained	Unconstrained
Normalization Variation	`OverallSys` (η_{cs})	`NormFactor` (ϕ_p)
Coherent Shape Variation	`HistoSys` σ_{csb}	–
Bin-by-bin variation	`ShapeSys` & `StatError` γ_{cb}	`ShapeFactor` γ_{csb}

HistFactory Capabilities (2)

- In addition the HistFactory can
  - can combine multiple channels
  - produce a RooFit workspace which can be used in RooStats
    - can be used to combine several measurements
- Configuration can be done in XML or directly in C++ or Python

How To Create a Model

- Simple counting model

$$\text{Poisson}(\, n_{obs} \mid \mu + b)\, \text{Gaussian}(\, b \mid b_0, \sigma_b)$$

```
// create first input histograms
int nobs = 3; double b = 1; double errb = 0.2;

// observed histogram
TH1D * hobs = new TH1D("hobs","hobs",1,0,1);
hobs->SetBinContent(1,nobs);

//signal histogram (assume expected one is 1)
TH1D * hs = new TH1D("hs","signal  histo",1,0,1);
hs->SetBinContent(1,1);


TH1D * hb = new TH1D("hb","bkg  histo",1,0,1);
hb->SetBinContent(1,b);
```

How To Create a Model (2)

- Create HistFactory Measurement class

```
HistFactory::Measurement meas("CountingModel","CountingModel");
meas.SetPOI("mu");

meas.SetLumi(1.0);
meas.SetLumiRelErr(0.1);  // not relevant
// this does not make lumi varying
meas.AddConstantParam("Lumi");
```

- Create Channels and Sample

```
HistFactory::Channel channel("SignalRegion");
channel.SetData(hobs);

HistFactory::Sample signal("signal");
signal.AddNormFactor("mu",1,0,30);
//signal.AddOverallSys("sig_unc",0.9, 1.1);
signal.SetHisto(hs);
channel.AddSample(signal);

HistFactory::Sample backg("background");
backg.SetHisto(h1_b);
backg.AddOverallSys("b_unc",1.-errb, 1+errb);  // b uncertainty
channel.AddSample(backg);

meas.AddChannel(channel);
```

How To Create a Model (3)

- Creating a RooWorkspace given the Measurement

```
RooWorkspace * w = HistFactory::MakeModelAndMeasurementFast(meas);
```

```
RooWorkspace(SignalRegion) SignalRegion workspace contents

variables
---------
(Lumi,alpha_b_unc,binWidth_obs_x_SignalRegion_0,binWidth_obs_x_SignalRegion_1,mu,nom_alpha_b_unc,nominalLumi,obs_x_S
nalRegion,weightVar)

p.d.f.s
-------
RooRealSumPdf::SignalRegion_model[ binWidth_obs_x_SignalRegion_0 * L_x_signal_SignalRegion_overallSyst_x_Exp +
binWidth_obs_x_SignalRegion_1 * L_x_background_SignalRegion_overallSyst_x_Exp ] = 2/2
RooGaussian::alpha_b_uncConstraint[ x=alpha_b_unc mean=nom_alpha_b_unc sigma=1 ] = 1
RooGaussian::lumiConstraint[ x=Lumi mean=nominalLumi sigma=0.001 ] = 1
RooProdPdf::model_SignalRegion[ lumiConstraint * alpha_b_uncConstraint * SignalRegion_model(obs_x_SignalRegion) ] =

functions
--------
RooProduct::L_x_background_SignalRegion_overallSyst_x_Exp[ Lumi * background_SignalRegion_overallSyst_x_Exp ] = 1
RooProduct::L_x_signal_SignalRegion_overallSyst_x_Exp[ Lumi * signal_SignalRegion_overallSyst_x_Exp ] = 1
RooStats::HistFactory::FlexibleInterpVar::background_SignalRegion_epsilon[ paramList=(alpha_b_unc) ] = 1
RooHistFunc::background_SignalRegion_nominal[ depList=(obs_x_SignalRegion) ] = 1
RooProduct::background_SignalRegion_overallSyst_x_Exp[ background_SignalRegion_nominal *
background_SignalRegion_epsilon ] = 1
RooHistFunc::signal_SignalRegion_nominal[ depList=(obs_x_SignalRegion) ] = 1
RooProduct::signal_SignalRegion_overallNorm_x_sigma_epsilon[ mu * signal_SignalRegion_epsilon ] = 1
RooProduct::signal_SignalRegion_overallSyst_x_Exp[ signal_SignalRegion_nominal *
signal_SignalRegion_overallNorm_x_sigma_epsilon ] = 1
```

HistFactory Output

- makes a combined workspace with data

```
RooWorkspace(combined) combined contents

variables
---------
(channelCat,nom_alpha_b_unc,obs_x_SignalRegion,weightVar)

datasets
--------
RooDataSet::asimovData(obs_x_SignalRegion,weightVar,channelCat)
RooDataSet::obsData(channelCat,obs_x_SignalRegion)

named sets
----------
ModelConfig_GlobalObservables:(nom_alpha_b_unc)
ModelConfig_Observables:(obs_x_SignalRegion,weightVar,channelCat)
globalObservables:(nom_alpha_b_unc)
observables:(obs_x_SignalRegion,weightVar,channelCat)
```

- create also a ModelConfig

```
=== Using the following for ModelConfig ===
Observables:             RooArgSet:: = (obs_x_SignalRegion,weightVar,channelCat)
Parameters of Interest:  RooArgSet:: = (mu)
Nuisance Parameters:     RooArgSet:: = (alpha_b_unc)
Global Observables:      RooArgSet:: = (nom_alpha_b_unc)
PDF:                     RooSimultaneous::simPdf[ indexCat=channelCat SignalRegion=model_SignalRegion ] = 2
```

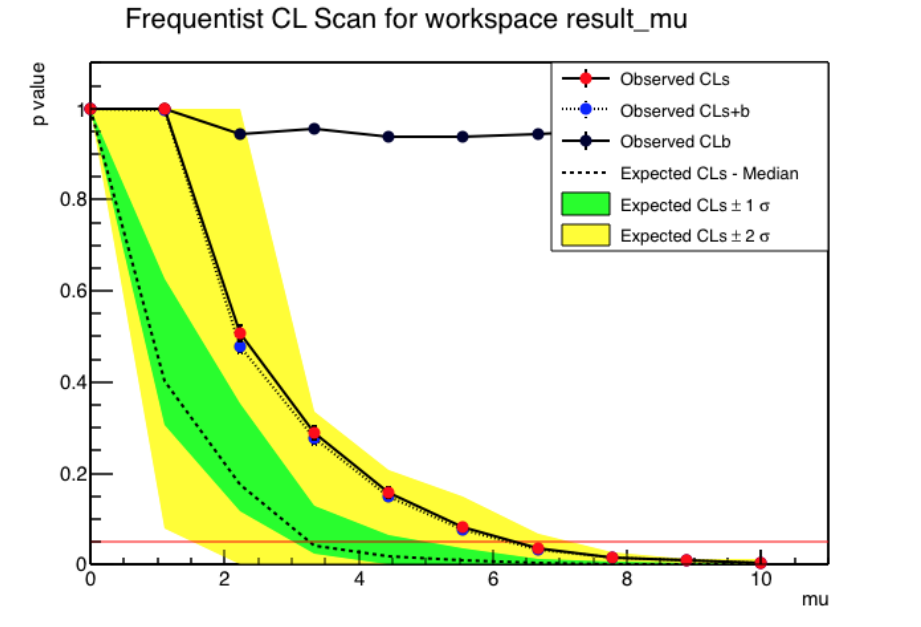
Using HistFactory Models

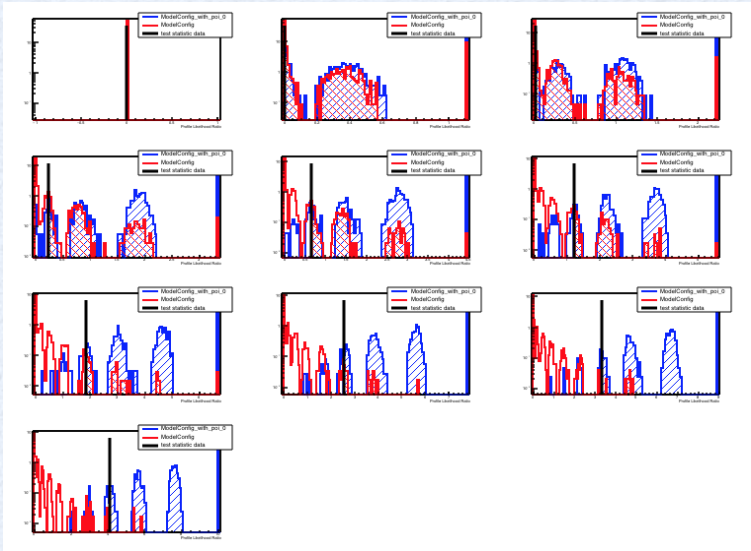
- Combined model saved in a ROOT file
- Model can be used directly in RooStats tools

```
root[] .L StandardHypoTestInvDemo.C
```

run for CLs (with frequentist calculator (type = 0) and one-side PL test statistics (type = 3) scan 10 points in [0,10]

```
root[] StandardHypoTestInvDemo("model.root","combined","ModelConfig","","obsData",0,3, true, 10, 0, 10)
```





Interpolation Options

HistFactory has different option for interpolating the systematic variations : $\eta(\alpha)$

- 0) Linear
- 1) Exponential
- 2) Quadratic interp. linear extrapolation
- 4) Polynomial interpolation Exponential extrapolation (default)

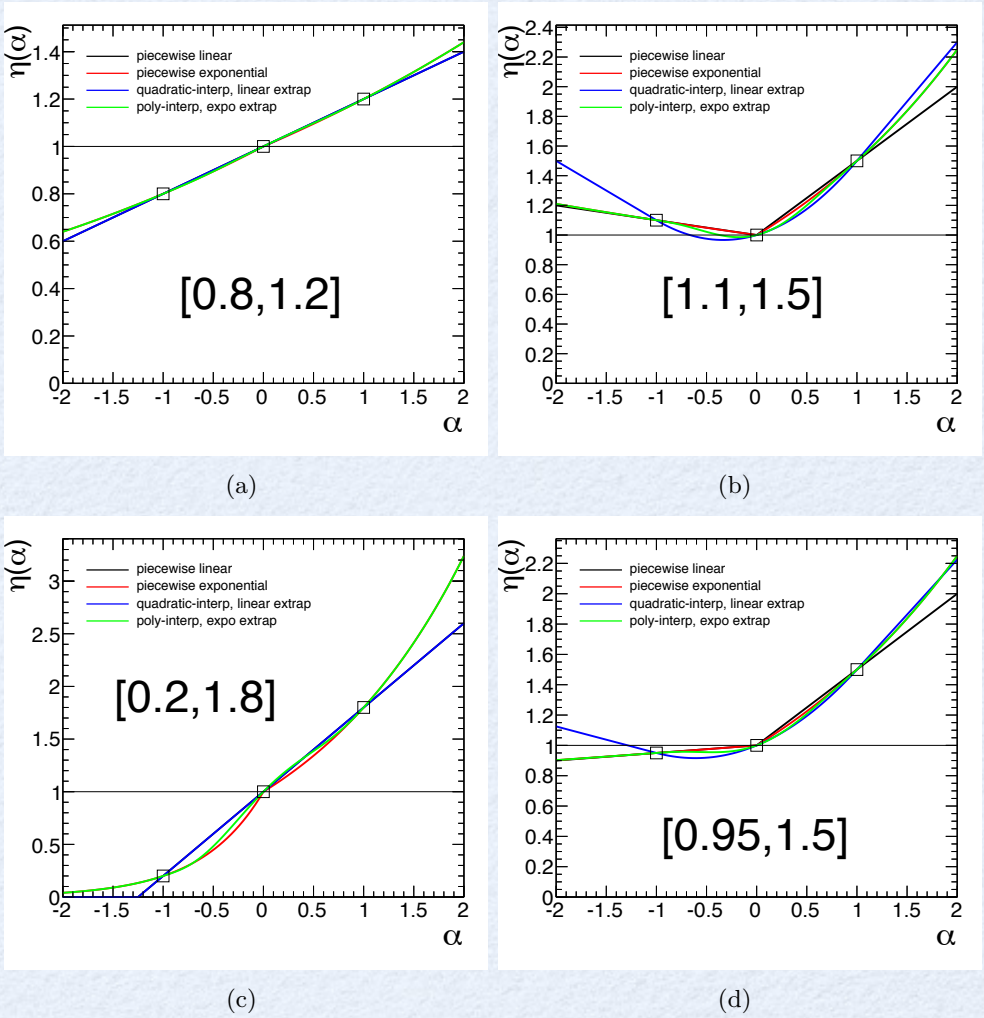


Figure 3: Comparison of the three interpolation options for different $\eta^{\pm}$. (a) $\eta^{-} = 0.8$, $\eta^{+} = 1.2$, (b) $\eta^{-} = 1.1$, $\eta^{+} = 1.5$, (c) $\eta^{-} = 0.2$, $\eta^{+} = 1.8$, and (d) $\eta^{-} = 0.95$, $\eta^{+} = 1.5$.

Summary

- Advanced statistical data/analysis with RooStats
  - LHC results (*e.g.* Higgs observation)
- Capable of using different tools and interpretations (Frequentist/Bayesian) on the same model
- Generic tools capable to deal with large variety of models
  - based on histograms or un-binned data
  - multi-dimensional observations
- Provide tools to facilitate complex model building
  - HistFactory for histogram based analysis

Documentation

- RooStats TWiki: https://twiki.cern.ch/twiki/bin/view/RooStats/WebHome
  - with links to presentations and Twiki pages with examples from schools
- HistFactory document: https://cdsweb.cern.ch/record/1456844/files/CERN-OPEN-2012-016.pdf
- RooStats tutorial macros: http://root.cern.ch/root/html534/tutorials/roostats/index.html
- RooStats user support:
  - Request support via ROOT talk forum: https://root.cern.ch/phpBB3/viewforum.php?f=15 (questions on statistical concepts accepted)
  - contact me directly (email: Lorenzo.Moneta at cern.ch)
- Contacts for statistical questions:
  - ATLAS statistics forum:
    - TWiki: https://twiki.cern.ch/twiki/bin/view/AtlasProtected/StatisticsTools
  - CMS statistics committee:
    - TWiki: https://twiki.cern.ch/twiki/bin/view/CMS/StatisticsCommittee

Time For Last Exercise !

Follow the Twiki page at
https://twiki.cern.ch/twiki/bin/view/RooStats/RooStatsExercisesMarch2015

Thank you !