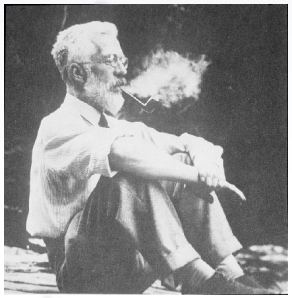
Applied Statistics

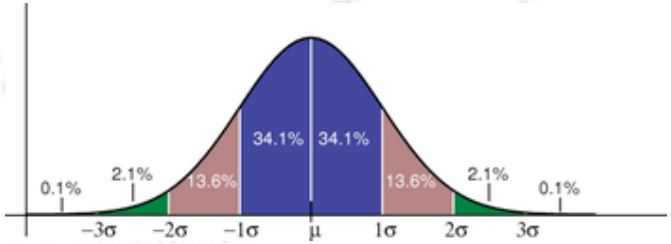
Background subtraction and sPlots







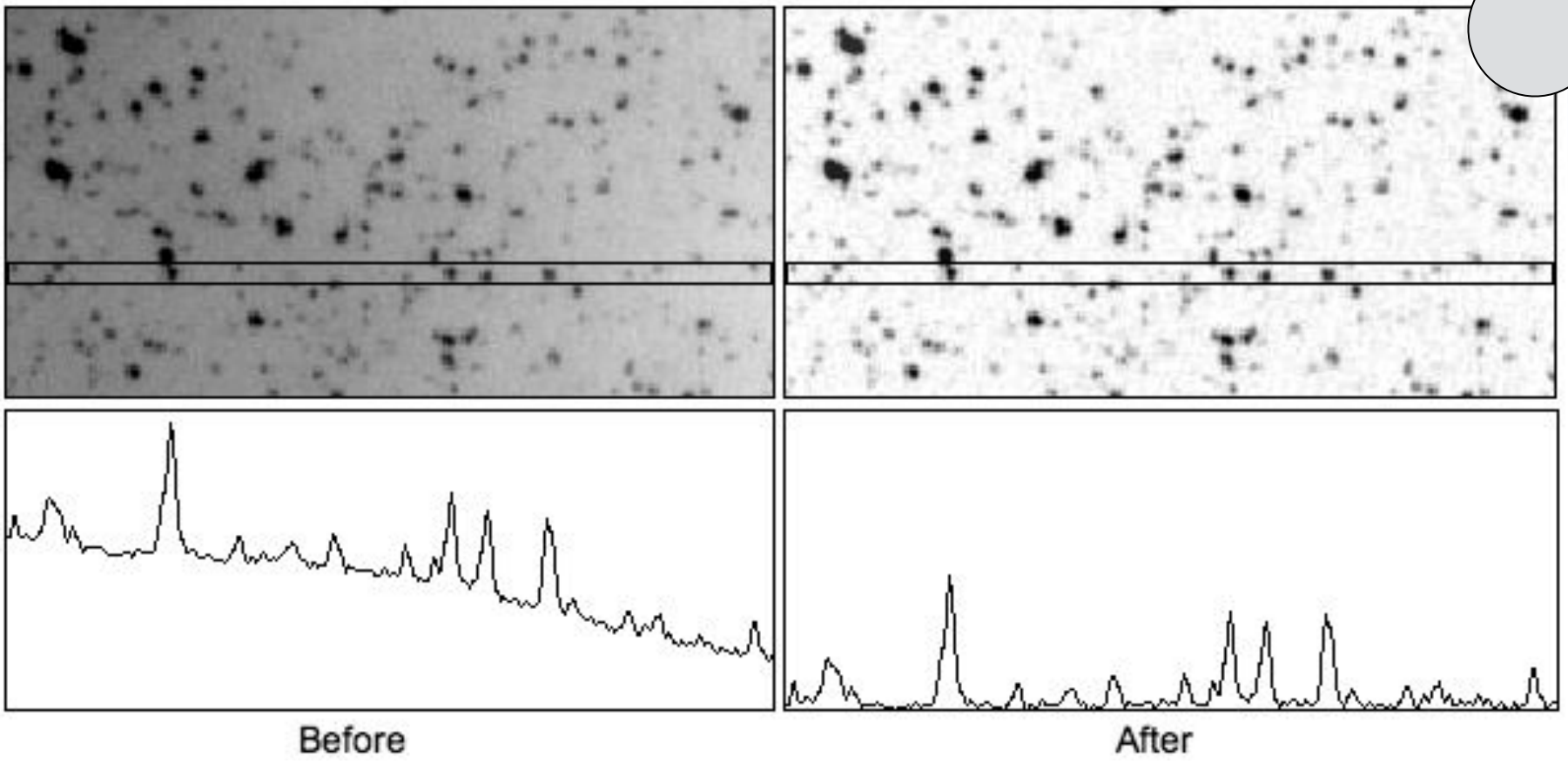

Troels C. Petersen (NBI)



"Statistics is merely a quantisation of common sense"

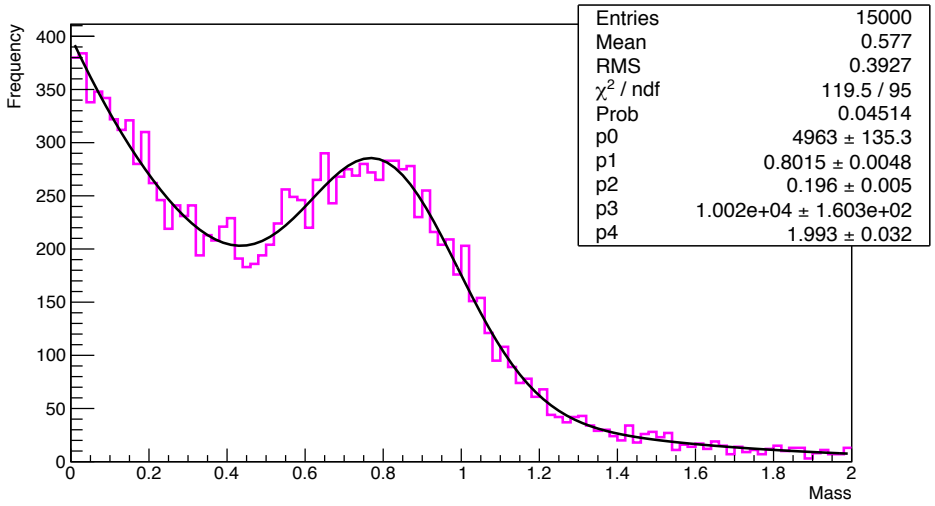
Background subtraction

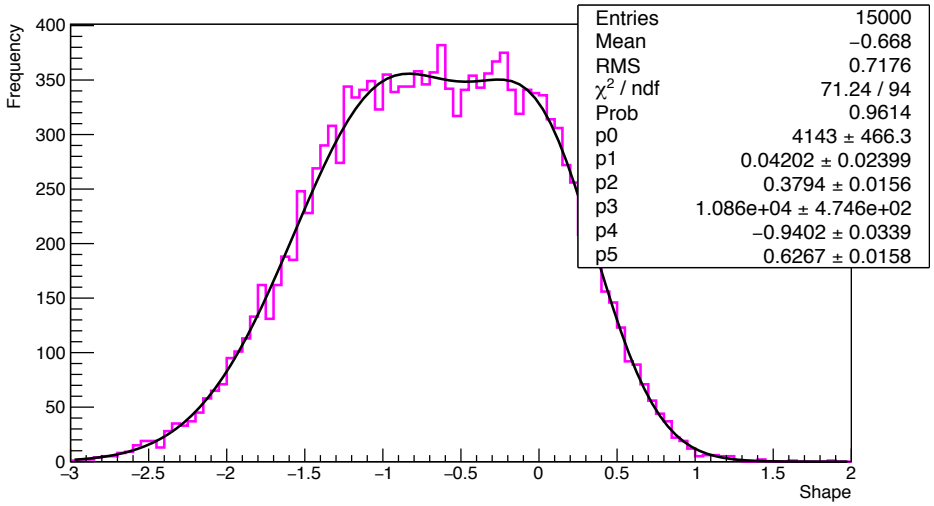
In many fields, data contains noise/background that one would like to get rid of. Instruments are build to do this, but in the quest for ever more sensitivity, we must deal with this problem. Typically, it is specific to each field of science, but in the end the requirement is the same… what does **"X"** look like in pure signal?



Stating the challenge

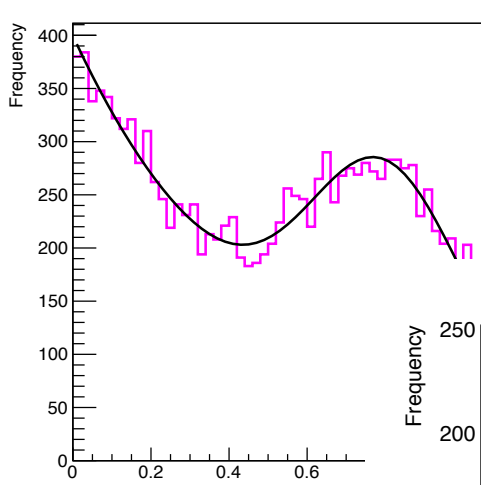
Given some variables that only partially distinguishes signal from background, how do you estimate the distribution of other uncorrelated variables?



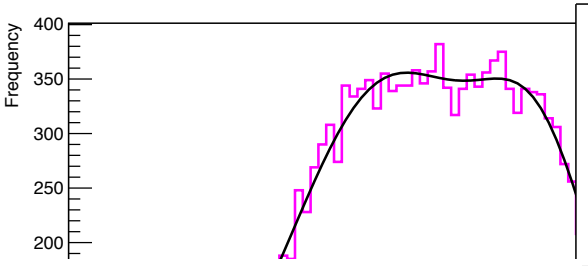


Stating the challenge

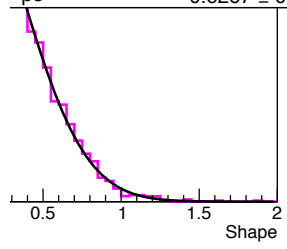
Given some variables that only partially distinguishes signal from background, how do you estimate the distribution of other uncorrelated variables?



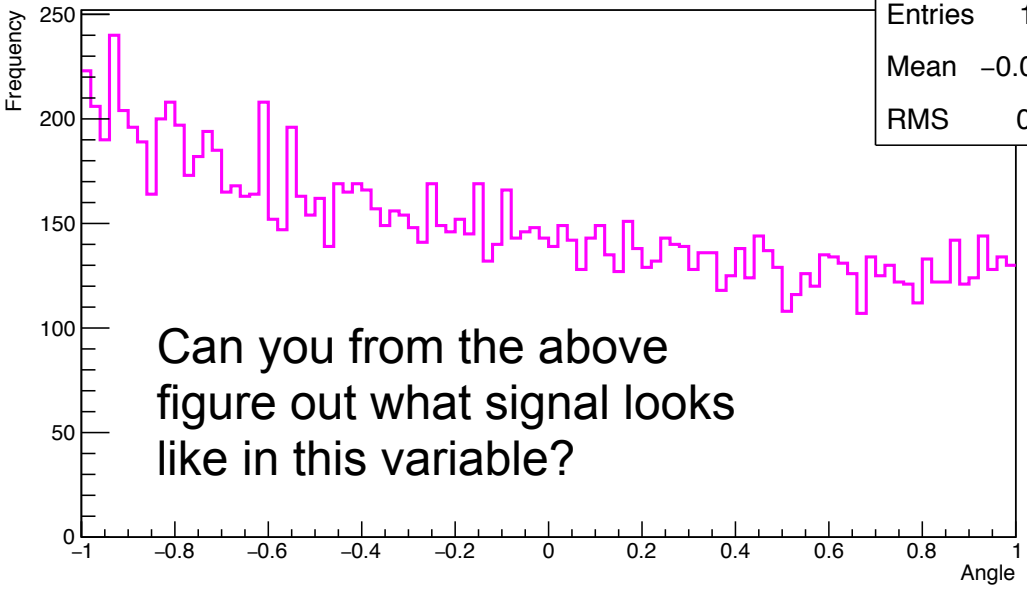
Entries	15000
Mean	0.577
RMS	0.3927
χ^2 / ndf	119.5 / 95
Prob	0.04514
p0	4963 ± 135.3
p1	0.8015 ± 0.0048
p2	0.196 ± 0.005
p3	1.002e+04 ± 1.603e+02
p4	1.993 ± 0.032



Entries	15000
Mean	−0.668
RMS	0.7176
χ^2 / ndf	71.24 / 94
Prob	0.9614
p0	4143 ± 466.3
p1	0.04202 ± 0.02399
p2	0.3794 ± 0.0156
p3	1.086e+04 ± 4.746e+02
p4	−0.9402 ± 0.0339
p5	0.6267 ± 0.0158



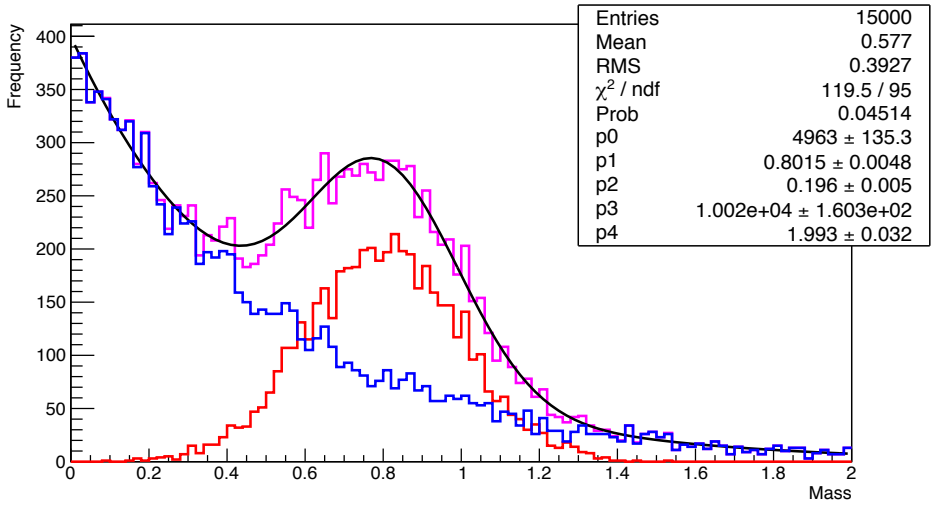
Entries	15000
Mean	−0.08619
RMS	0.5851

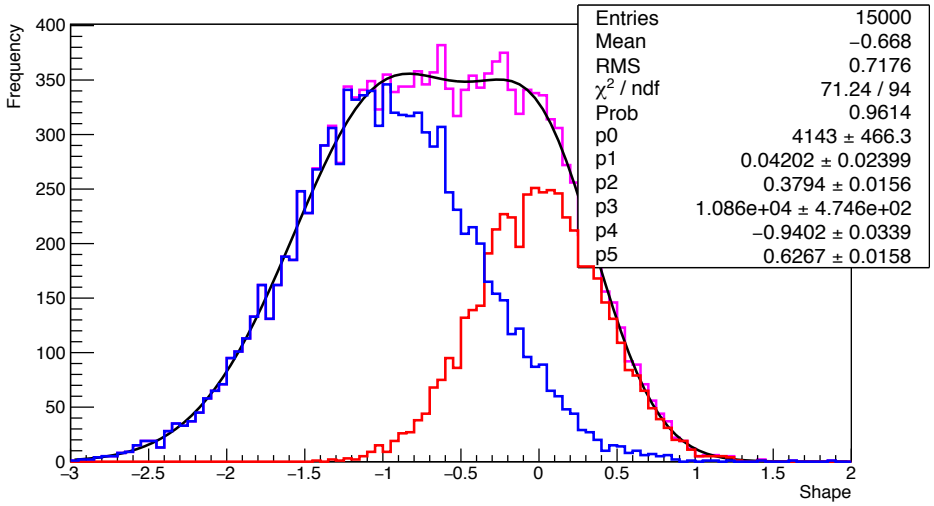


Can you from the above figure out what signal looks like in this variable?

Stating the challenge

Given some variables that only partially distinguishes signal from background, how do you estimate the distribution of other uncorrelated variables?





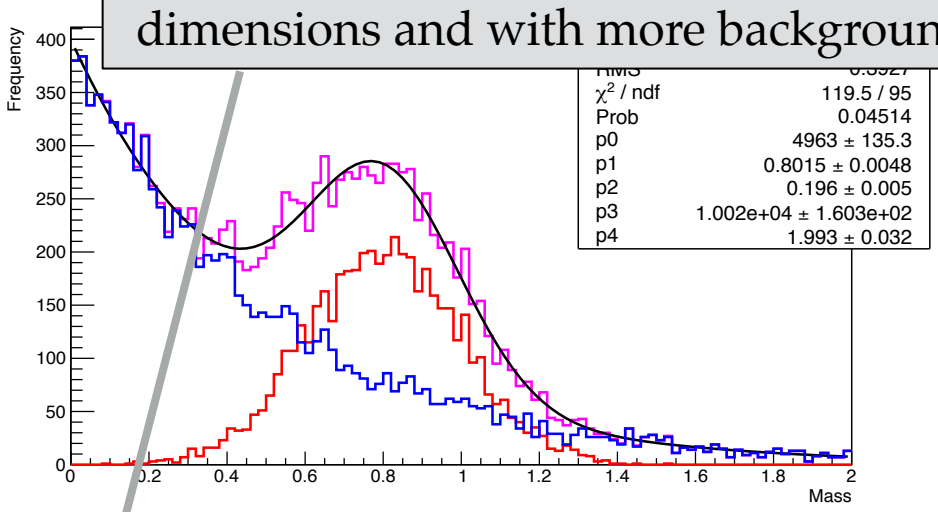
We may try two things:

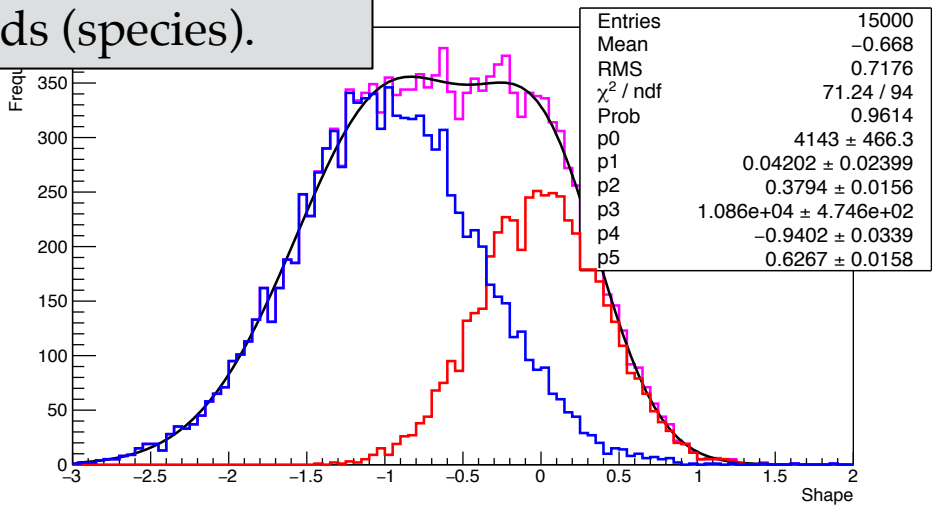
- Define a signal and a background region, plot the variable of interest for each of these, and subtract as much background as you estimate there is.
- Calculate a signal weight = PDFsig / (PDFsig + PDFbkg) and weigh each event by this weight.

Stating the challenge

G…al from background, h… variables?

It requires that you can find a region of pure background, and neither all signal nor all background is used in the estimate, hence it is suboptimal.
It is also technically challenging, especially in higher dimensions and with more backgrounds (species).





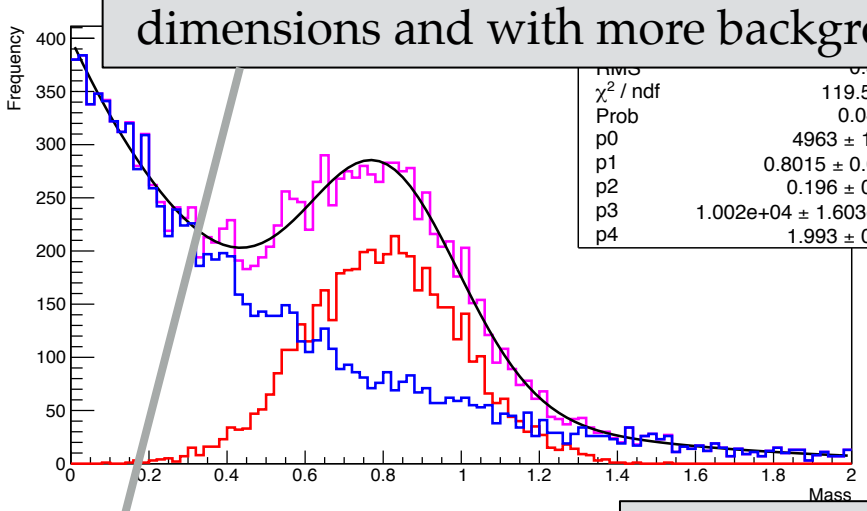
We may try two things:

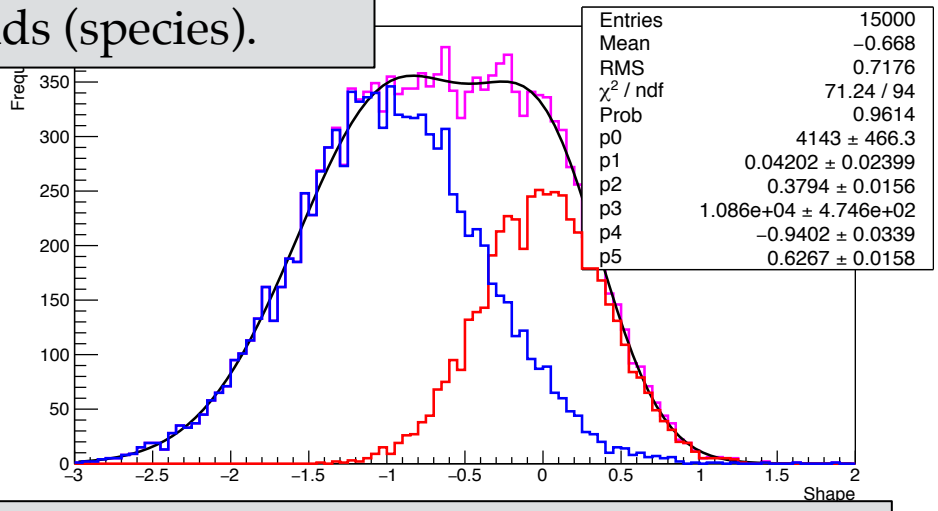
- Define a signal and a background region, plot the variable of interest for each of these, and subtract as much background as you estimate there is.
- Calculate a signal weight = PDFsig / (PDFsig + PDFbkg) and weigh each event by this weight.

Stating the challenge

G…al from background, h… variables?

It requires that you can find a region of pure background, and neither all signal nor all background is used in the estimate, hence it is suboptimal.
It is also technically challenging, especially in higher dimensions and with more backgrounds (species).





This yields a biased estimate (no entries below zero!), for which the uncertainty on each bin can not be determined. But the idea is on to something…

We may try two things:

- Define a signal and a ba… of these, and subtract as…
- Calculate a signal weight = PDFsig / (PDFsig + PDFbkg) and weigh each event by this weight.

The "solution"

The answer is of course "yes", there is a good "solution", which is called sPlots.

${}_s\mathcal{P}lot$:

a statistical tool to unfold data distributions

M. Pivk[a] and F.R. Le Diberder[b]

[a] *CERN,*
CH-1211 Geneva 23, Switzerland

[b] *Laboratoire de l'Accélérateur Linéaire,*
IN2P3-CNRS et Université de Paris-Sud, F-91898 Orsay, France

arXiv:physics/0402083v3 [physics.data-an] 2 Sep 2005

Defining the case

The log-Likelihood is expressed as:

$$\mathcal{L} = \sum_{e=1}^{N} \ln \left\{ \sum_{i=1}^{\mathrm{N_s}} N_i \mathrm{f}_i(y_e) \right\} - \sum_{i=1}^{\mathrm{N_s}} N_i \ , \tag{1}$$

where

- N is the total number of events in the data sample,
- $\mathrm{N_s}$ is the number of species of events populating the data sample,
- N_i is the number of events expected on the average for the i^{th} species,
- y is the set of discriminating variables,
- f_i is the Probability Density Function (PDF) of the discriminating variables for the i^{th} species,
- $\mathrm{f}_i(y_e)$ denotes the value taken by the PDFs f_i for event e, the later being associated with a set of values y_e for the set of discriminating variables,
- x is the set of control variables which, by definition, do not appear in the above expression of $\mathcal{L}$.

How to calculate an sWeight?

Given an event, the sWeight of it is calculated as follows:

$$ {}_s\mathcal{P}_{\mathrm{n}}(y_e) = \frac{\sum_{j=1}^{\mathrm{N_s}} \mathbf{V}_{\mathrm{n}j} \mathrm{f}_j(y_e)}{\sum_{k=1}^{\mathrm{N_s}} N_k \mathrm{f}_k(y_e)} $$

How to calculate an sWeight?

Given an event, the sWeight of it is calculated as follows:

$$ {}_s\mathcal{P}_{\mathrm{n}}(y_e) = \frac{\sum_{j=1}^{\mathrm{Ns}} \mathbf{V}_{\mathrm{n}j} \mathrm{f}_j(y_e)}{\sum_{k=1}^{\mathrm{Ns}} N_k \mathrm{f}_k(y_e)} $$

It may look complicated, but in fact it is not that far from the simple proposal:

$$ w = \frac{N_{sig} PDF_{sig}}{N_{sig} PDF_{sig} + N_{bkg} PDF_{bkg}} $$

We simply decided to put the covariance matrix in, which turns out to be the right choice.

How to calculate an sWeight?

Given an event, the sWeight of it is calculated as follows:

$$ {}_s\mathcal{P}_{\mathrm{n}}(y_e) = \frac{\sum_{j=1}^{\mathrm{N_s}} \mathbf{V}_{\mathrm{n}j} \mathrm{f}_j(y_e)}{\sum_{k=1}^{\mathrm{N_s}} N_k \mathrm{f}_k(y_e)} $$

It may look complicated, but in fact it is not that far from the simple proposal:

$$ w = \frac{V_{sig,sig} N_{sig} PDF_{sig} + V_{sig,bkg} N_{bkg} PDF_{bkg}}{N_{sig} PDF_{sig} + N_{bkg} PDF_{bkg}} $$

We simply decided to put the covariance matrix in, which turns out to be the right choice.

How to calculate an sWeight?

Given an event, the sWeight of it is calculated as follows:

$$ {}_s\mathcal{P}_{\mathrm{n}}(y_e) = \frac{\sum_{j=1}^{\mathrm{N_s}} \mathbf{V}_{\mathrm{n}j} \mathrm{f}_j(y_e)}{\sum_{k=1}^{\mathrm{N_s}} N_k \mathrm{f}_k(y_e)} $$

Eq. 14

This is the sWeight for signal (n) of a single event "e", which has discriminating observables y_e.

Covariance matrix between the signal and background (species) normalisations.

Number of events for each species.

Value of PDF for each species at the event in question.

Q: How do one get the covariance matrix V?

A: Either from fit or calculated as follows: $\mathbf{V}^{-1}_{\mathrm{n}j} = \frac{\partial^2(-\mathcal{L})}{\partial N_{\mathrm{n}} \partial N_j} = \sum_{e=1}^{N} \frac{\mathrm{f}_{\mathrm{n}}(y_e)\mathrm{f}_j(y_e)}{(\sum_{k=1}^{\mathrm{N_s}} N_k \mathrm{f}_k(y_e))^2}$

Eq. 10

The sPlot recipe

This Section is meant to show that using ${}_s\mathcal{P}lot$ is indeed easy. The different steps to implement the technique are the following:

1. One is dealing with a data sample in which several species of events are present.
2. A maximum Likelihood fit is performed to obtain the yields N_i of the various species. The fit relies on a discriminating variable y uncorrelated with a control variable x: the later is therefore totally absent from the fit.
3. The sWeights ${}_s\mathcal{P}$ are calculated using Eq. (14) where the covariance matrix is obtained by inverting the matrix given by Eq. (10).
4. Histograms of x are filled by weighting the events with the sWeights ${}_s\mathcal{P}$. The sum of the entries are equal to the yields N_i provided by the fit.
5. Error bars per bin are given by Eq. (22). The sum of the error bars squared are equal to the uncertainties squared $\mathbf{V}_{ii}$ provided by the fit.
6. The ${}_s\mathcal{P}lots$ reproduce the true distributions of the species in the control variable x, within the above defined statistical uncertainties.

The ${}_s\mathcal{P}lot$ method has been implemented in the ROOT framework under the class TSPlot [2].

I will in the following go through each of these steps, and so will the following exercise.

The sPlot recipe

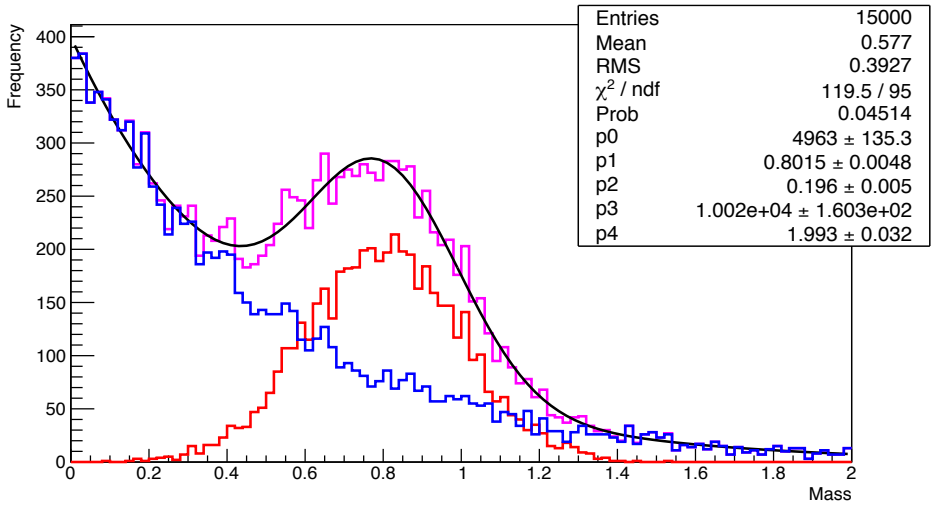
This Section is meant to show that using ${}_s\mathcal{P}lot$ is indeed easy. The different steps to implement the technique are the following:

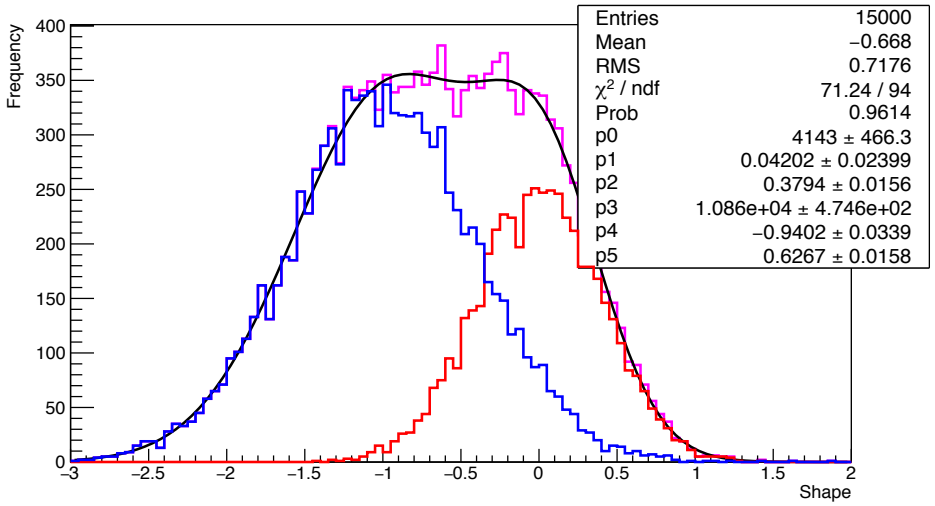
1. One is dealing with a data sample in which several species of events are present. ✓
2. A maximum Likelihood fit is performed to obtain the yields N_i of the various species. The fit relies on a discriminating variable y uncorrelated with a control variable x: the later is therefore totally absent from the fit.
3. The sWeights ${}_s\mathcal{P}$ are calculated using Eq. (14) where the covariance matrix is obtained by inverting the matrix given by Eq. (10).
4. Histograms of x are filled by weighting the events with the sWeights ${}_s\mathcal{P}$. The sum of the entries are equal to the yields N_i provided by the fit.
5. Error bars per bin are given by Eq. (22). The sum of the error bars squared are equal to the uncertainties squared $\mathbf{V}_{ii}$ provided by the fit.
6. The ${}_s\mathcal{P}lots$ reproduce the true distributions of the species in the control variable x, within the above defined statistical uncertainties.

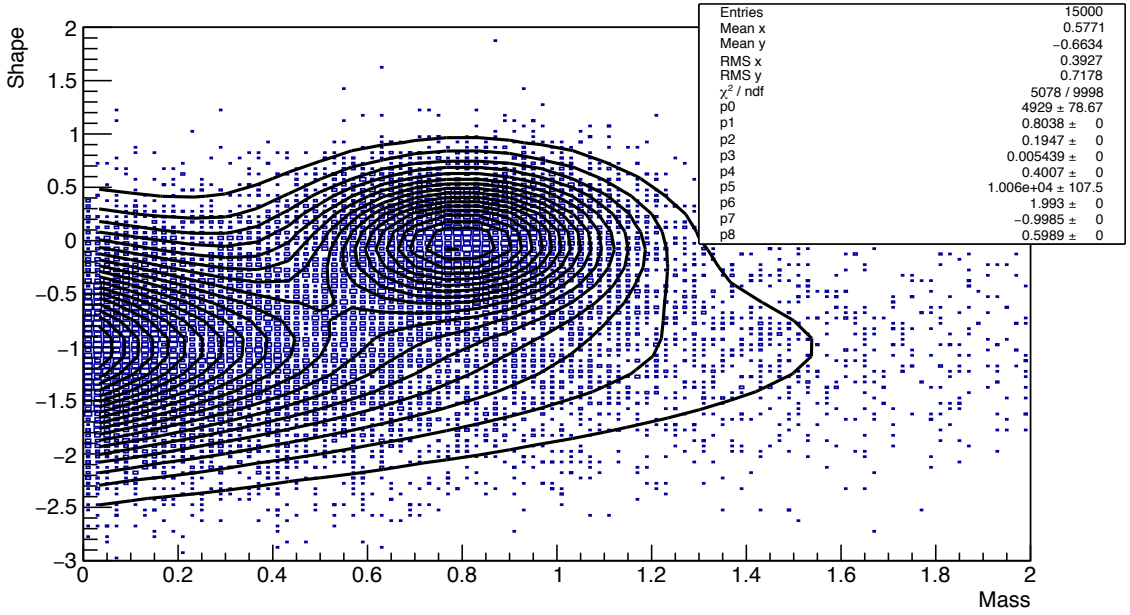
The ${}_s\mathcal{P}lot$ method has been implemented in the ROOT framework under the class TSPlot [2].

I will in the following go through each of these steps, and so will the following exercise.

OK - we get the fit going…







The sPlot recipe

This Section is meant to show that using ${}_s\mathcal{P}lot$ is indeed easy. The different steps to implement the technique are the following:

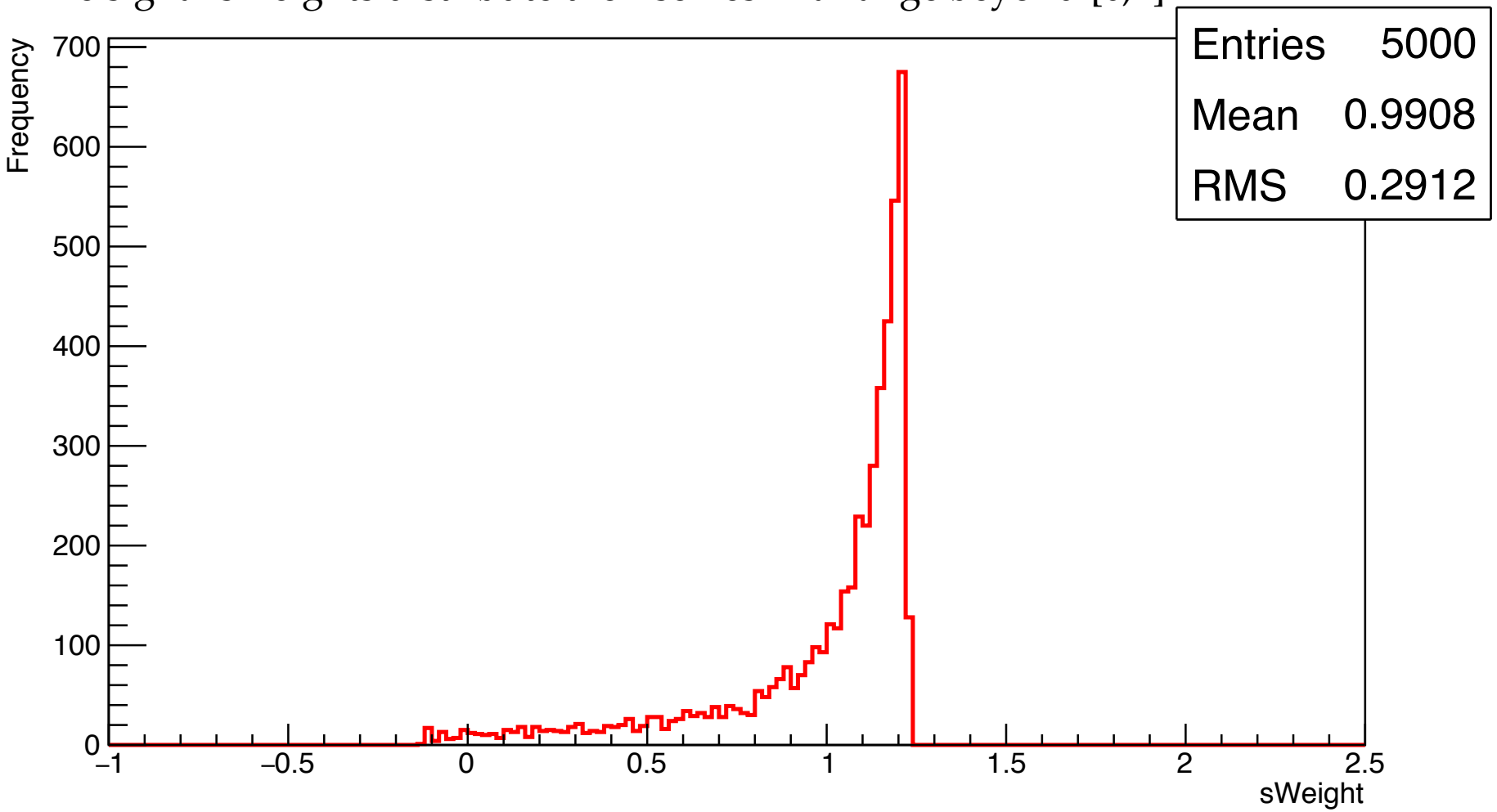
1. One is dealing with a data sample in which several species of events are present. ✓
2. A maximum Likelihood fit is performed to obtain the yields N_i of the various species. The fit relies on a discriminating variable y uncorrelated with a control variable x: the later is therefore totally absent from the fit. ✓
3. The sWeights ${}_s\mathcal{P}$ are calculated using Eq. (14) where the covariance matrix is obtained by inverting the matrix given by Eq. (10).
4. Histograms of x are filled by weighting the events with the sWeights ${}_s\mathcal{P}$. The sum of the entries are equal to the yields N_i provided by the fit.
5. Error bars per bin are given by Eq. (22). The sum of the error bars squared are equal to the uncertainties squared $\mathbf{V}_{ii}$ provided by the fit.
6. The ${}_s\mathcal{P}lots$ reproduce the true distributions of the species in the control variable x, within the above defined statistical uncertainties.

The ${}_s\mathcal{P}lot$ method has been implemented in the ROOT framework under the class TSPlot [2].

I will in the following go through each of these steps, and so will the following exercise.

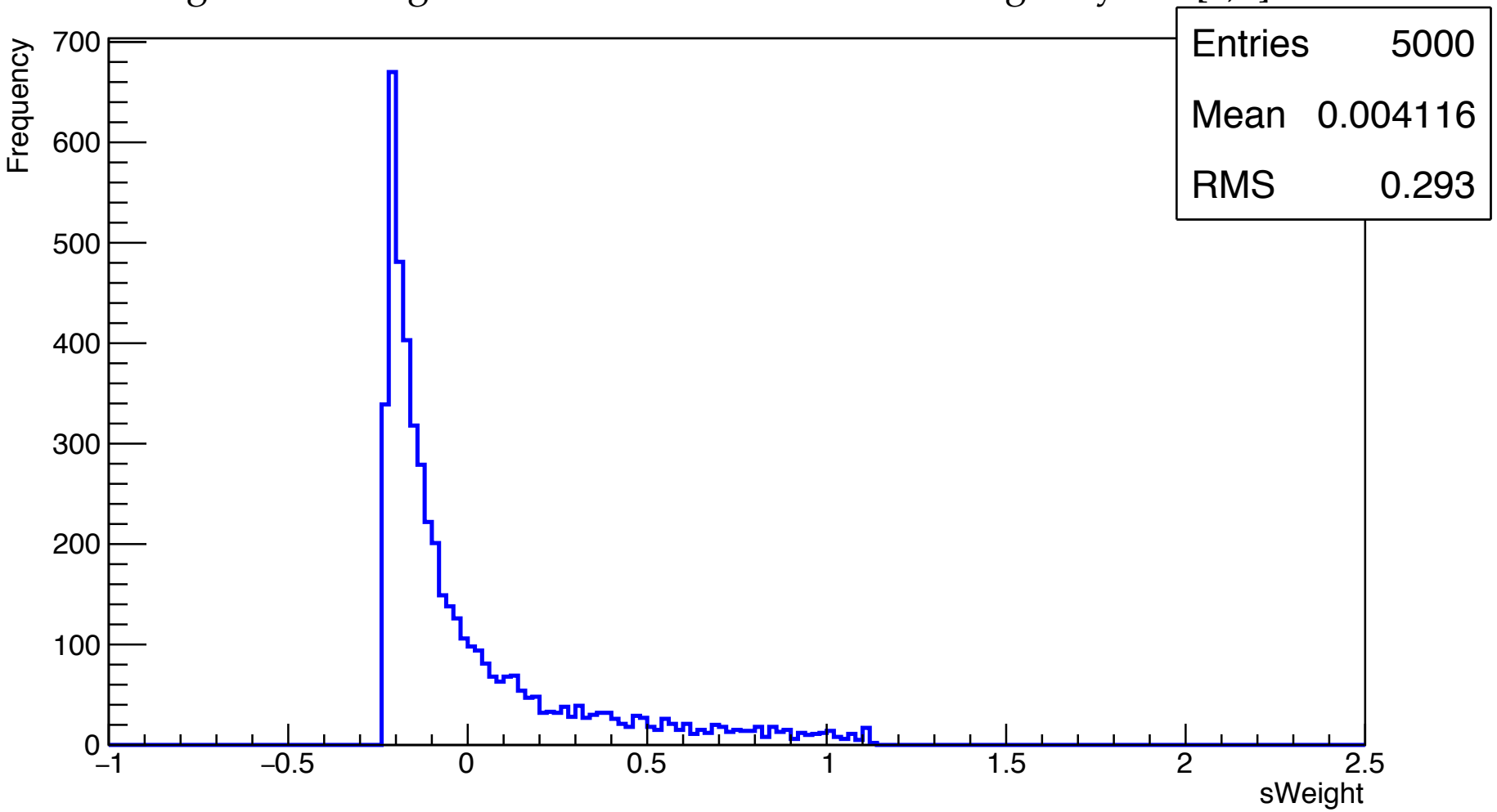
What do the sWeights look like?

The signal sWeights distribute themselves in a range beyond [0,1]



What do the sWeights look like?

The background sWeights distribute themselves in a range beyond [0,1]



The sPlot recipe

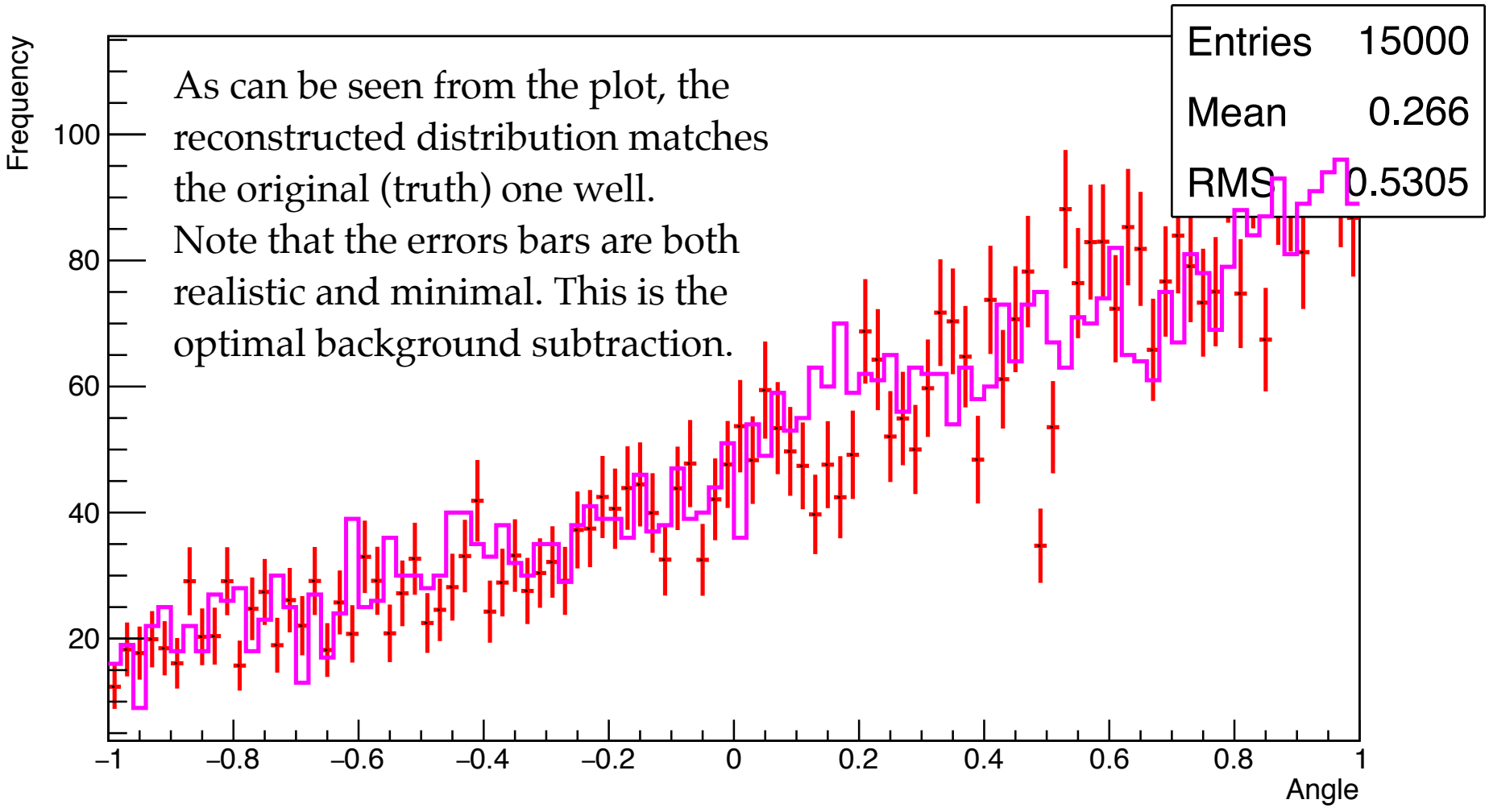
This Section is meant to show that using ${}_s\mathcal{P}lot$ is indeed easy. The different steps to implement the technique are the following:

1. One is dealing with a data sample in which several species of events are present. ✓
2. A maximum Likelihood fit is performed to obtain the yields N_i of the various species. The fit relies on a discriminating variable y uncorrelated with a control variable x: the later is therefore totally absent from the fit. ✓
3. The sWeights ${}_s\mathcal{P}$ are calculated using Eq. (14) where the covariance matrix is obtained by inverting the matrix given by Eq. (10). ✓
4. Histograms of x are filled by weighting the events with the sWeights ${}_s\mathcal{P}$. The sum of the entries are equal to the yields N_i provided by the fit.
5. Error bars per bin are given by Eq. (22). The sum of the error bars squared are equal to the uncertainties squared $\mathbf{V}_{ii}$ provided by the fit.
6. The ${}_s\mathcal{P}lots$ reproduce the true distributions of the species in the control variable x, within the above defined statistical uncertainties.

The ${}_s\mathcal{P}lot$ method has been implemented in the ROOT framework under the class TSPlot [2].

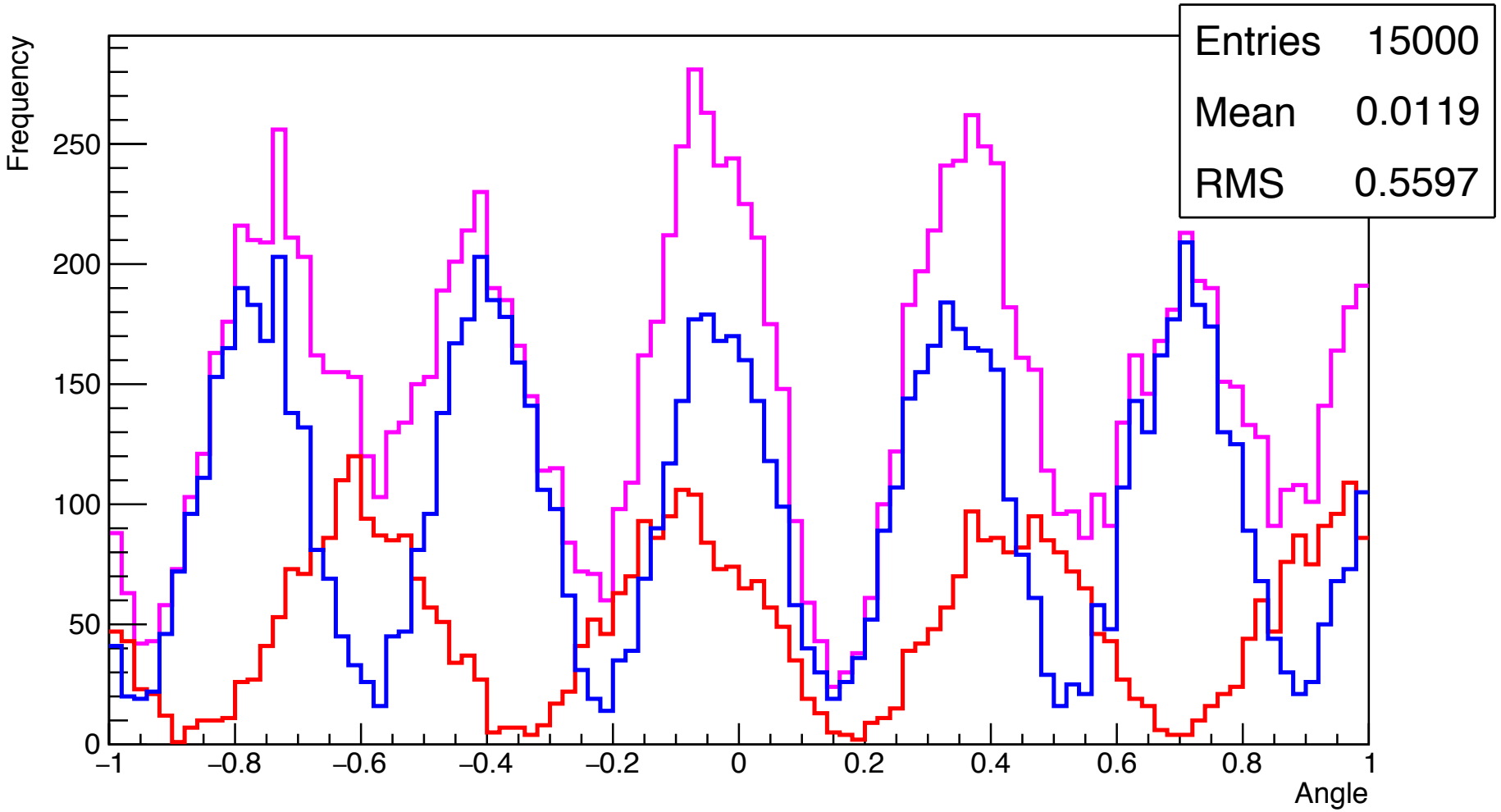
I will in the following go through each of these steps, and so will the following exercise.

Using the sWeights for an sPlot



Using the sWeights for an sPlot

This can be used for pretty much any case, where some “control” variables of known PDF are used to separate signal(s) from background(s).



The sPlot recipe

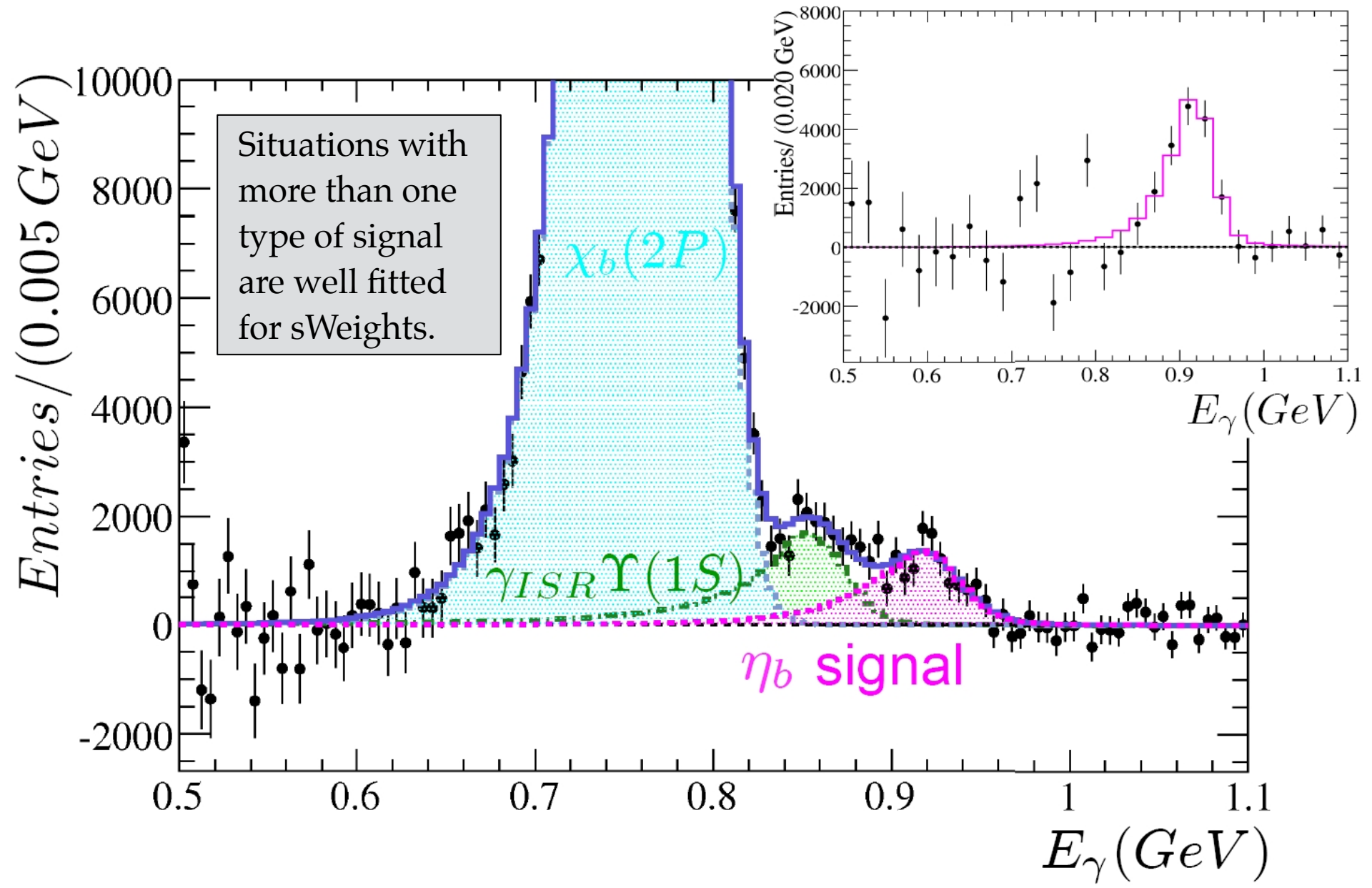
This Section is meant to show that using ${}_s\mathcal{P}lot$ is indeed easy. The different steps to implement the technique are the following:

1. One is dealing with a data sample in which several species of events are present. ✓
2. A maximum Likelihood fit is performed to obtain the yields N_i of the various species. The fit relies on a discriminating variable y uncorrelated with a control variable x: the later is therefore totally absent from the fit. ✓
3. The sWeights ${}_s\mathcal{P}$ are calculated using Eq. (14) where the covariance matrix is obtained by inverting the matrix given by Eq. (10). ✓
4. Histograms of x are filled by weighting the events with the sWeights ${}_s\mathcal{P}$. The sum of the entries are equal to the yields N_i provided by the fit. ✓
5. Error bars per bin are given by Eq. (22). The sum of the error bars squared are equal to the uncertainties squared $\mathbf{V}_{ii}$ provided by the fit. ✓
6. The ${}_s\mathcal{P}lots$ reproduce the true distributions of the species in the control variable x, within the above defined statistical uncertainties. ✓

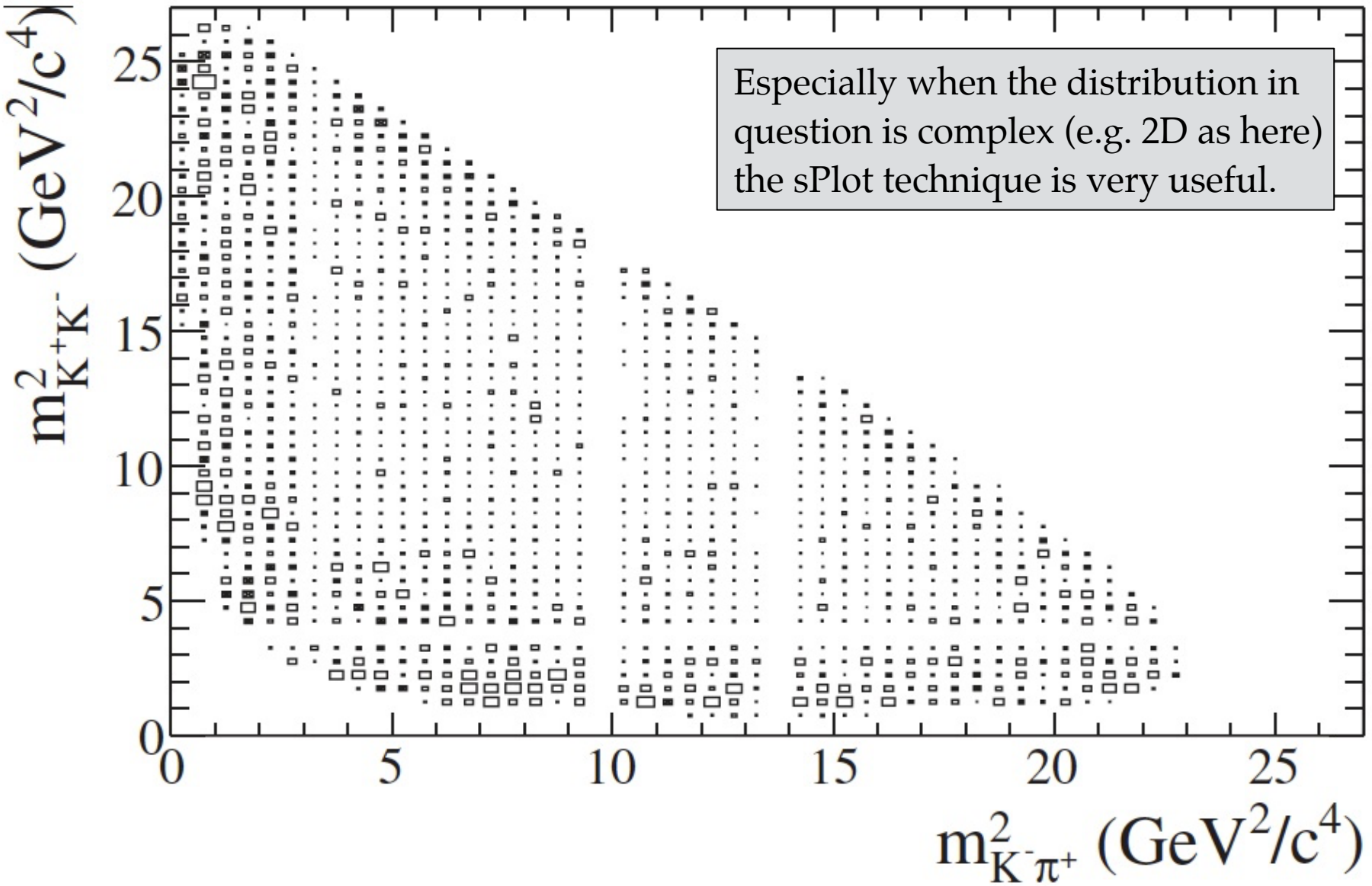
The ${}_s\mathcal{P}lot$ method has been implemented in the ROOT framework under the class TSPlot [2].

I will in the following go through each of these steps, and so will the following exercise.

Examples of use…



Examples of use…



Especially when the distribution in question is complex (e.g. 2D as here) the sPlot technique is very useful.

Comments & Conclusions

The sWeights and sPlots are suitable for problems in several dimensions with significant backgrounds yet signal that can be fitted.

Each event contributes exactly with unity weight, i.e. the sum of the sWeights for all contributions (signal(s)+background(s)) is one:

$$\sum_{l=1}^{\mathrm{N_s}} {}_s\mathcal{P}_l(y_e) = \sum_{l=1}^{\mathrm{N_s}} \frac{\sum_{j=1}^{\mathrm{N_s}} \mathbf{V}_{lj} \mathrm{f}_j(y_e)}{\sum_{k=1}^{\mathrm{N_s}} N_k \mathrm{f}_k(y_e)} = \frac{\sum_{j=1}^{\mathrm{N_s}} N_j \mathrm{f}_j(y_e)}{\sum_{k=1}^{\mathrm{N_s}} N_k \mathrm{f}_k(y_e)} = 1$$

Note that the variables of interest, x, must not be correlated with the discriminating variables, y.

The method has become widely used (in particle physics), which is evident from the number of citations the paper has (506 in total).