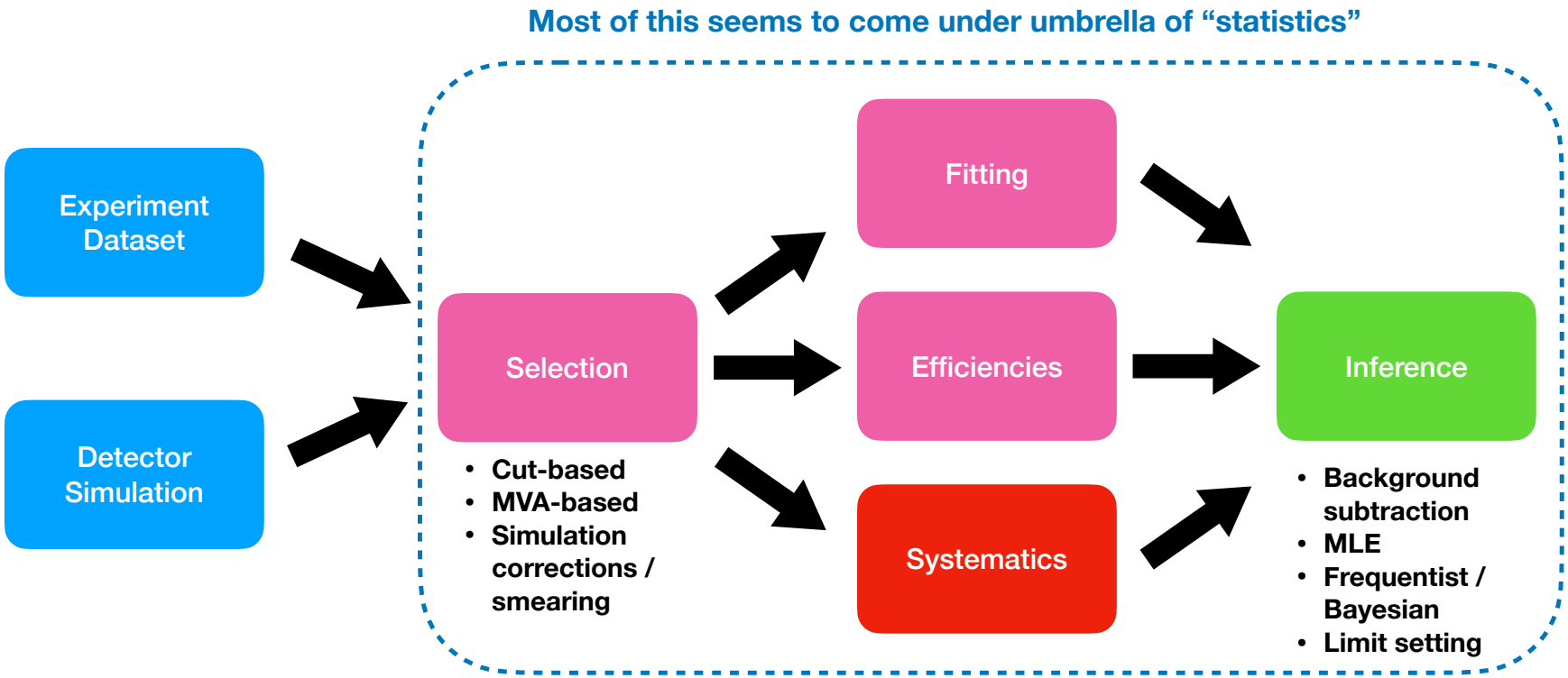
Statistical Methods at LHCb

Matthew Kenzie
on behalf of the LHCb Collaboration

QCHS 2021, "Stavanger" (my garage)

2nd August 2021

What do we mean by statistics?



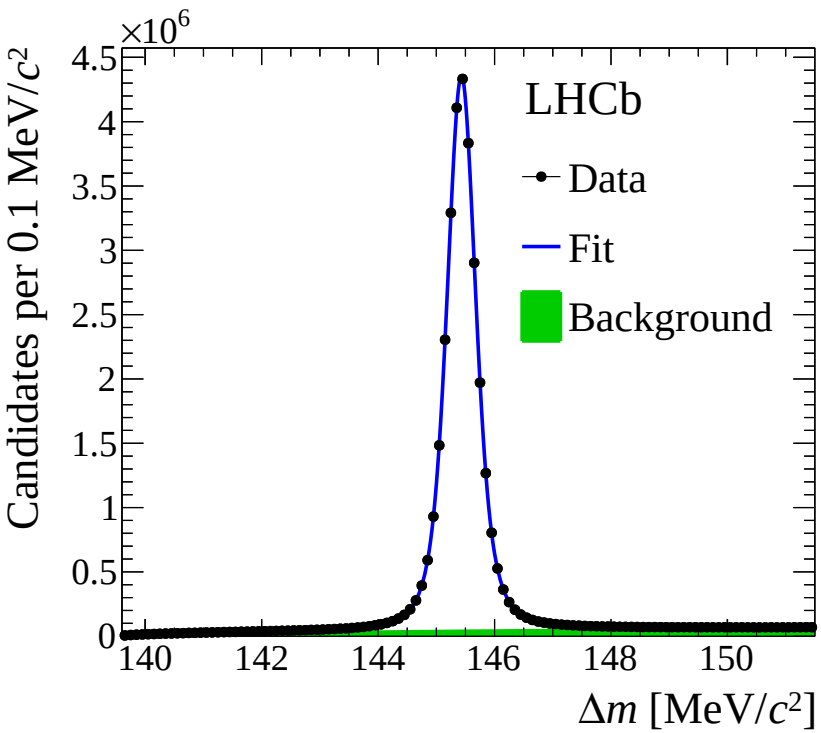
What sort of physics are we doing at LHCb?

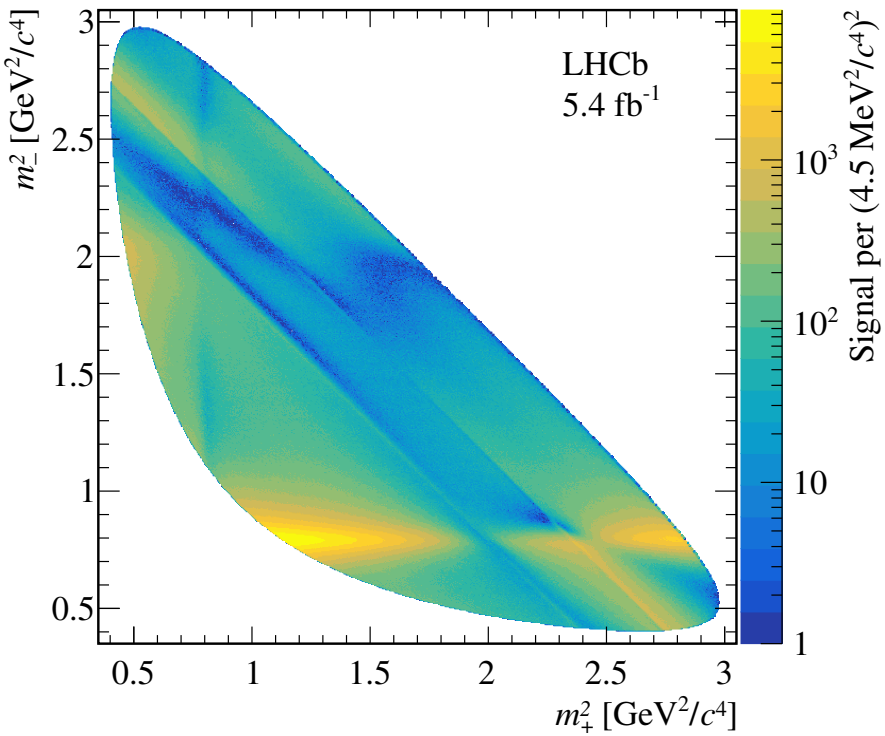
- Predominantly comes under the umbrella of "heavy-flavour physics"
- But from a statistics point of view it covers a broad spectrum
  - Very rare processes
    - Limit setting
  - Precision measurements
    - Large interference effects (*e.g.* CKM angles γ and β)
    - Small interference effects (*e.g.* charm and ϕ_s)
  - Amplitude studies
    - Often difficulty with models (and model uncertainties)
  - Bump hunting / spectroscopy
    - Nearly always as intermediate resonance
  - Averaging / combinations
- But some very common themes
  - Hadron experiment so always need background modelling or background substraction (which is not always easy to simulate)
  - A quite well known sector so often have high statistics control modes / regions
- Lots of overlap with averaging groups: PDG and HFLAV

A few example analyses

Measurement of D^0–$\overline{D}^0$ mixing

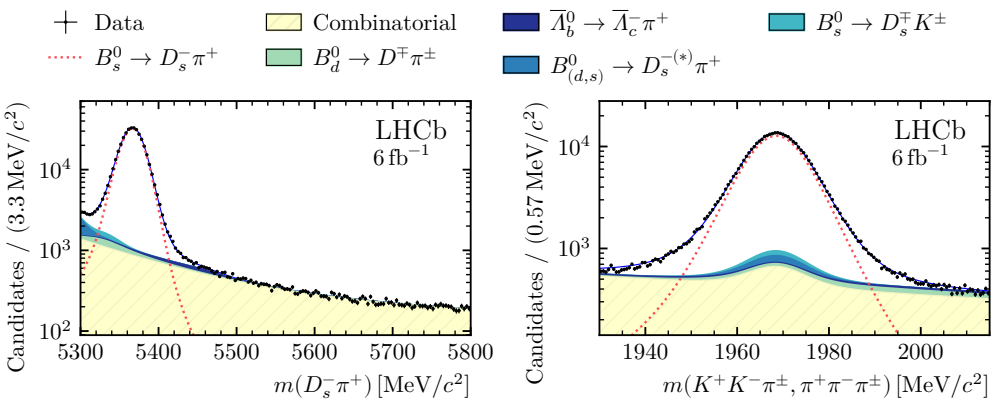
- Submitted to PRL - [arXiv:2106.03744]
- Use flavour tagged D^0 and $\overline{D}^0$ decays to $K^0_S\pi^+\pi^-$ to measure charm mixing and *CP* violation in charm mixing
- Huge samples - 30.6M signal candidates (only the prompt $D^{*+} \to D^0\pi^+$ candidates and only 2016–2018 data)
- Very small mixing effect (even smaller *CP* violation effect)

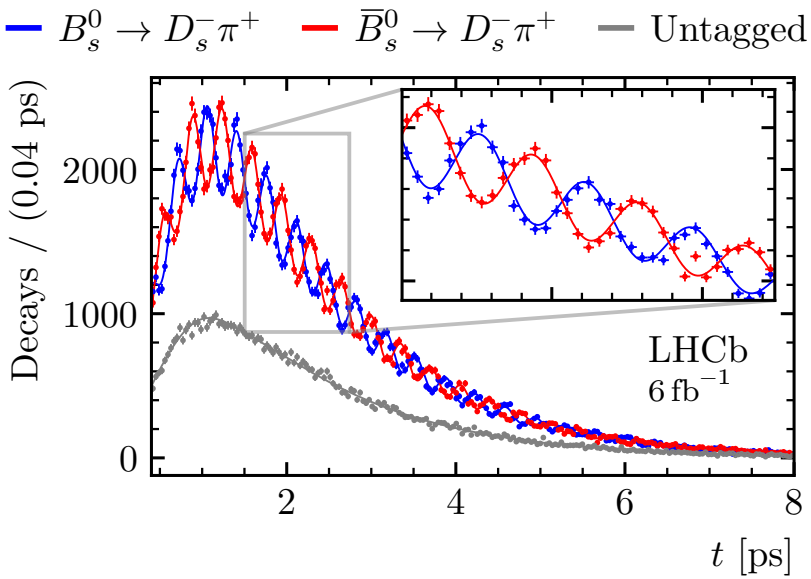




Measurement of B_s^0–$\overline{B}_s^0$ mixing

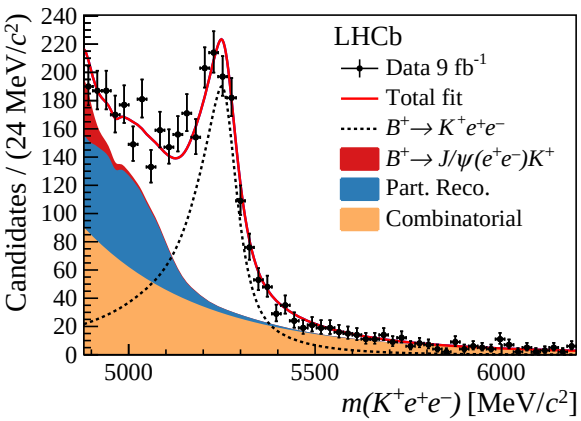
- Submitted to Nature Physics - [arXiv:2104.04421]
- Use relatively large sample of 380K $B_s^0 \to D_s^- \pi^+$ candidates (2011–2018 data)
- No flavour tagging decay (much weaker tagging power): $\epsilon = 80\%$, $\omega = 36\%$
- Large mixing effect (negligible *CP* violation effect)

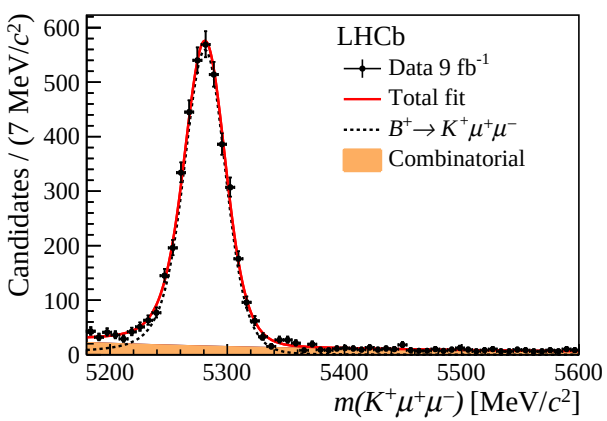


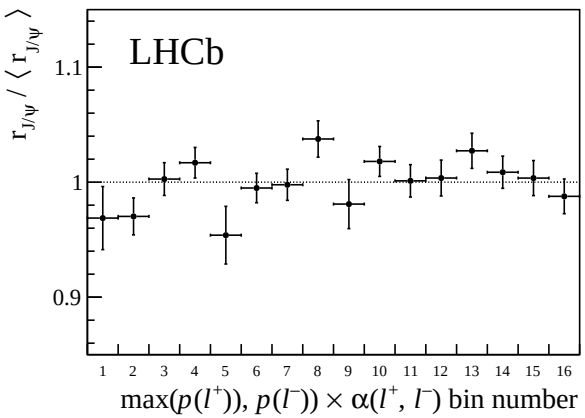


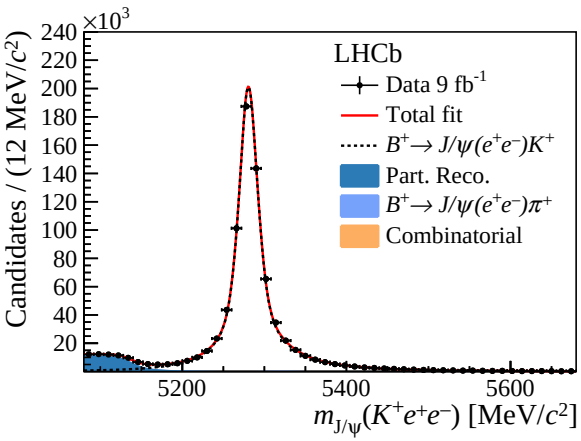
Measurement of RK

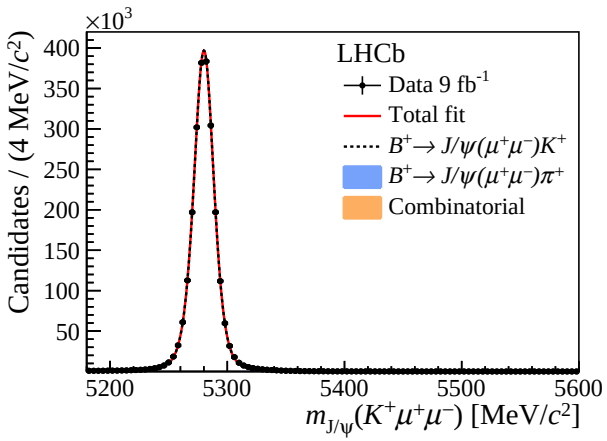
- Submitted to Nature Physics - [arXiv:2103.11769]
- Large sample control modes (750K / 2.3M) but small sample rare modes (1640 / 3850)
- Understanding efficiencies is crucial





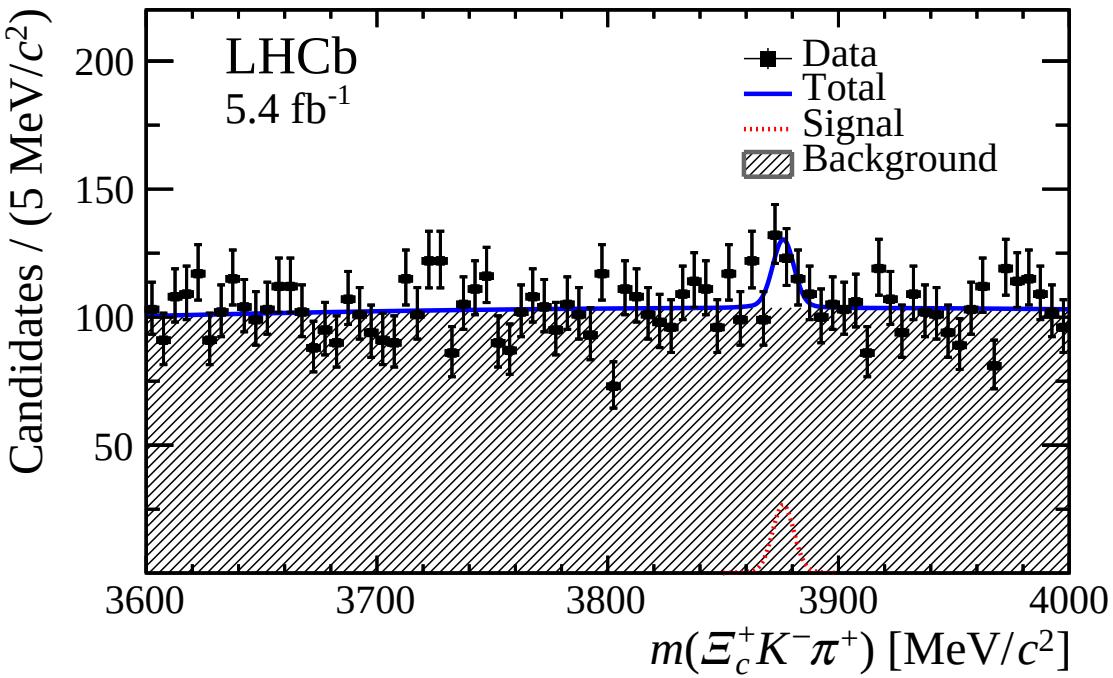






Search for the doubly charmed Ω^+cc baryon

- Submitted to Sci. China Phys. Mech. Astr. - [arXiv:2105.06841]
- Resonance search using $\Omega_{cc}^+ \to \Xi_c^+ K^- \pi^+$



Background subtraction

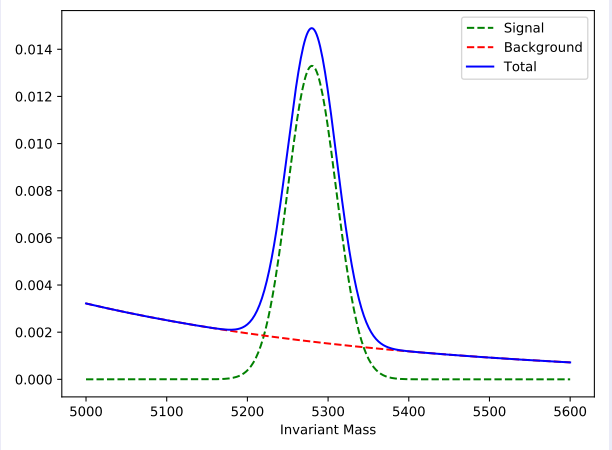
Background subtraction and sWeights

- A key component of almost all LHCb analyses
- In many cases we rely on the use of sWeights
  - Particle ID calibration performed on sWeighted control channels.
  - Flavour tagging calibration performed on sWeighted control channels.
  - Many *CP* and ampltiude fits rely on sWeighted data samples.
  - Quite often our Selection MVAs are trained on sWeighted control channels.
- The *sPlot* method is a statistical tool for unfolding the signal distribution in some control variable if you can distinguish it from the background using an independent discriminating variable
- The *sPlot* technique is widley used in HEP, based on the M. Pivk and F. Le Diberder paper [1]
- Actually gets mentioned way before this by R. Barlow but not cited [2] and before that in a slightly different context by P. Condon and P. Cowell [3]
- Wide discussion within LHCb (and further afield) on this topic (*e.g.* PHYSTAT-2020 workshop) mainly due to development by M. Schmelling and contributions from MK, H. Dembinski and C. Langenbruch
- New ideas are being written up for publication, in particular a generalisation of sWeights dubbed "COWs" (Custom Orthogonal Weight functions)

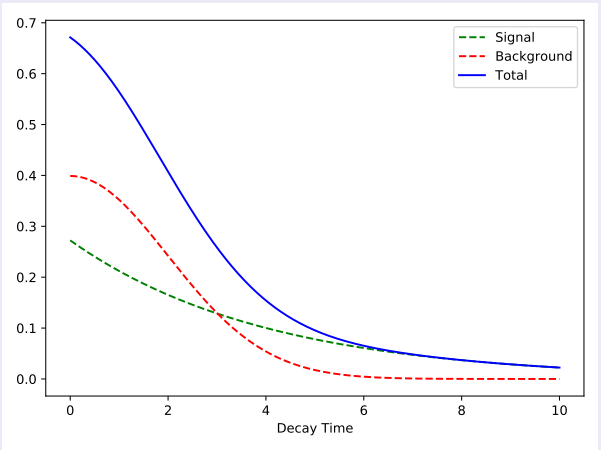
Setting up the problem

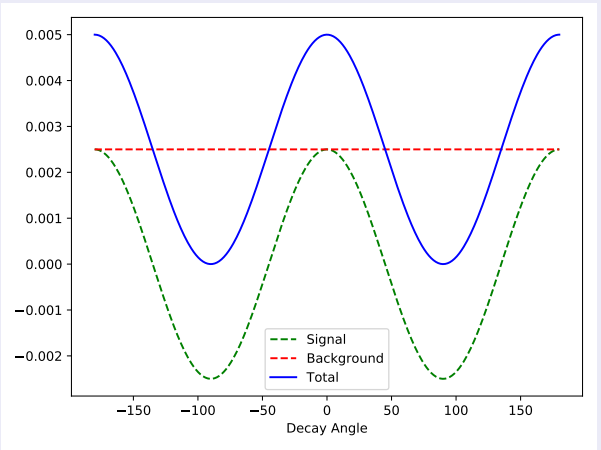
- In particle physics we often want to extract some properties of an observed signal
- But we typically have a non-neglible background contribution, usually distinguished using invariant mass
- The properties we want to extract are in some other dimension
  - Lifetime: Decay time distribution
  - Spin: Angular distributions
  - Amplitudes: Dalitz distributions

"Discriminant variable"



"Control variable(s)"





Setting up the problem

So what choices do we have?

- Fit the full nD distribution
  - Requires a suitable model description for each component in each dimension
- Sideband subtraction or "slicing"
  - Not statistically optimal (must be binned)
  - Requires the discriminant and control variables to be independent
- sWeighting
  - Essentially "per-event" slicing
  - Requires the discriminant and control variables to be independent
- For a total p.d.f. of the form

$$f(m,t) = \underbrace{z g_s(m) h_s(t)}_{\text{Signal}} + \underbrace{(1-z) g_b(m) h_b(t)}_{\text{Background}} \tag{1}$$

- Then can project signal and background out with functions (proof in backup),

$$w_s(m) = \underbrace{\frac{\alpha_s g_s(m) + \alpha_b g_b(m)}{g(m)}}_{\text{Signal}} \quad \text{and} \quad \underbrace{w_b(m) = \frac{\beta_s g_s(m) + \beta_b g_b(m)}{g(m)}}_{\text{Background}}, \tag{2}$$

by solving

$$\begin{pmatrix} W_{ss} & W_{sb} \\ W_{sb} & W_{bb} \end{pmatrix} \cdot \begin{pmatrix} \alpha_s & \beta_s \\ \alpha_b & \beta_b \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{where} \quad W_{xy} = \int \frac{g_x(m) g_y(m)}{g(m)} dm. \tag{3}$$

Application to a finite sample

- The original sPlot formalism comes with a few caveats
  - All shape parameters are known
  - All yields are freely floating and not expressed as fractions
  - Can only get weights for the unbinned sample you have fitted

- The advantanges with what Schmelling has found is that you simply need to provide a description of the p.d.f.s and you get back a weight function for each component
  - Shapes and yields can be determined however you want (even with constraints)
  - Can perform fit to a different sample to the one you use to extract weights (*e.g.* wider fit range)
  - Can perform the fit to a binned sample (if there are many events) and still extract a weight per-event

- There are still some caveats which apply to both
  - The description for each component must factorise in the disciminant and control variables
  - Factorisation means independence (which is more than just not-linearly-correlated)
  - However this can be circumvented with the use of COWs (although unfortunately I don't have time to go into this now but keep an eye on arXiv for more details)

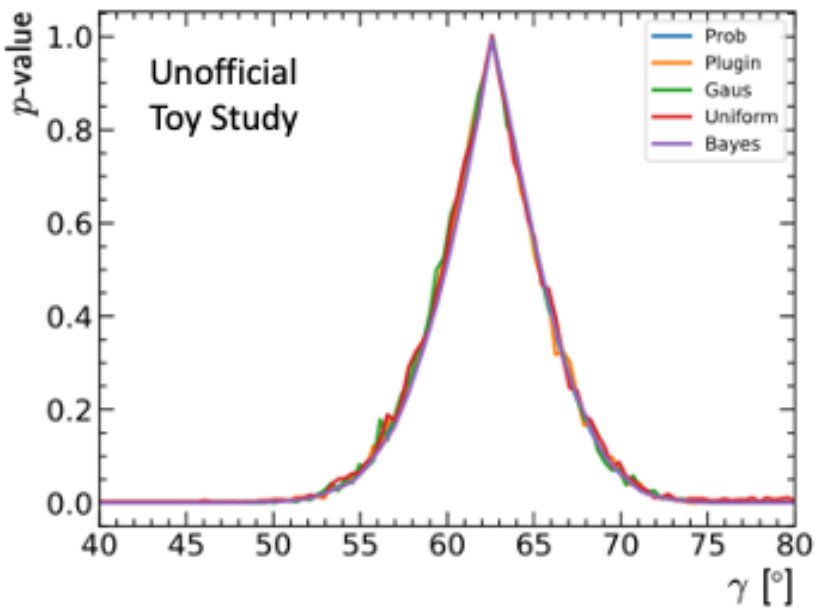
Fitting and inference

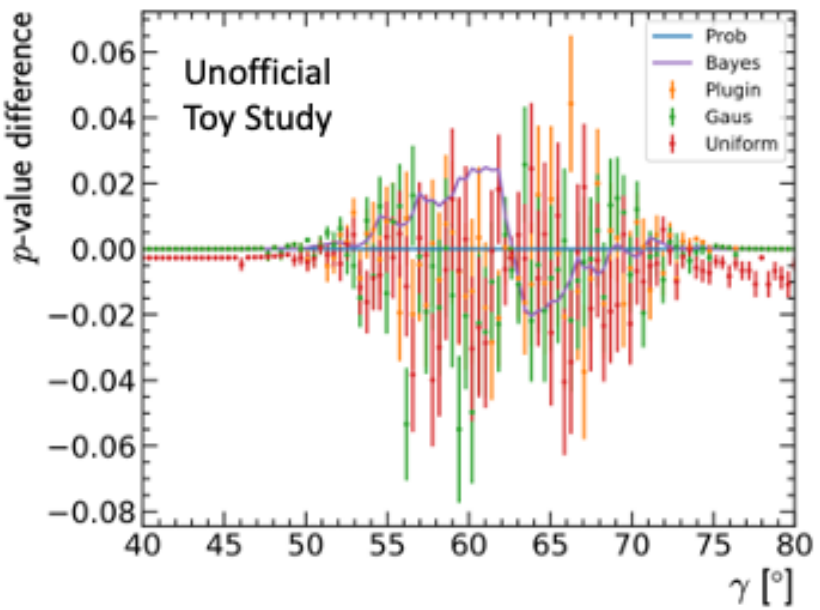
Fitting and inference software packages

- Most analyses in LHCb still make use of `RooFit` (and a bit of `RooStats`) a lot, especially for simple mass fits
- However it seems to be increasing that other tools are in use, especially for complex fits
  - `HistFactory` and `pyhf` - template fits (used a lot in semi-leptonics and EW)
  - `hepstats` - `Scikit-HEP` package with Bayesian block aglo. and LR based tests for discovrey, limits and intervals
  - `zfit` - a scaleable pythonic fit implementation a bit like `RooFit`
  - `Laura++` and many others - amplitude fits
  - `GammaCombo` - combinations and averaging
  - And many other custom implementations
- *Everything* uses `Minuit` as the backend
  - For stand-alone python implementation then `iminuit` - link
- Other commonly used useful tools
  - `uncertainties` - python package for error propagation and covariance tracking
  - `boost-histogram` - very fast multi-dimensional histogramming (python and C++)
  - `numba-stats` - parallelised pdf computation
  - `PyTorch` - machine learning
  - `XGBoost` - machine learning

Methods of inference

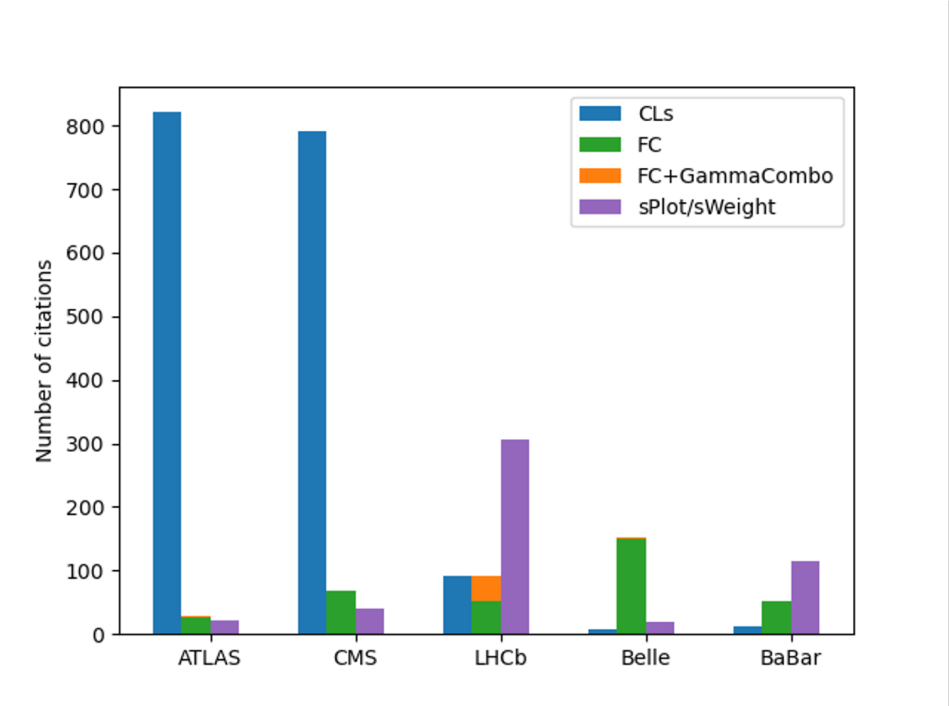
- MLE and Profile Likelihood
  - undoubtedly the most common and implemented by nearly every analysis
- Frequentist - Feldman-Cousins
  - often used if the coverage of the above is poor or near a physical boundary
  - What is done with the nuisance parameters?
    - Plugin / $\hat{\mu}$ - fix to profiled values - most common in LHCb
    - Gaussian sampling - sample from profiled value and uncertainty - rare in LHCb
    - Berger-Boos - uniform sample in region $(1-\beta)$ and the correct $p \to p+\beta$ - rare in LHCb
    - Cousins-Highland - take the median or expected value at profiled point (requirement of prior as need to know the nuisance parameter distribution - not used in LHCb





Methods of inference

- Bayesian
  - Occasionally used in LHCb with either simple MC or Markov Chain
- CLs
  - Very commonly used for setting limits
- Bootstrapping
  - Commonly used for robustness checks and systematics (rarer for interval estimation)



- Something we are trying to improve is use of **more than one** inference method to compare these different techniques
- See for example the latest LHCb combination of CKM angle γ and charm mixing parameters LHCb-CONF-2021-001

Systematic uncertainties

Systematic uncertainties

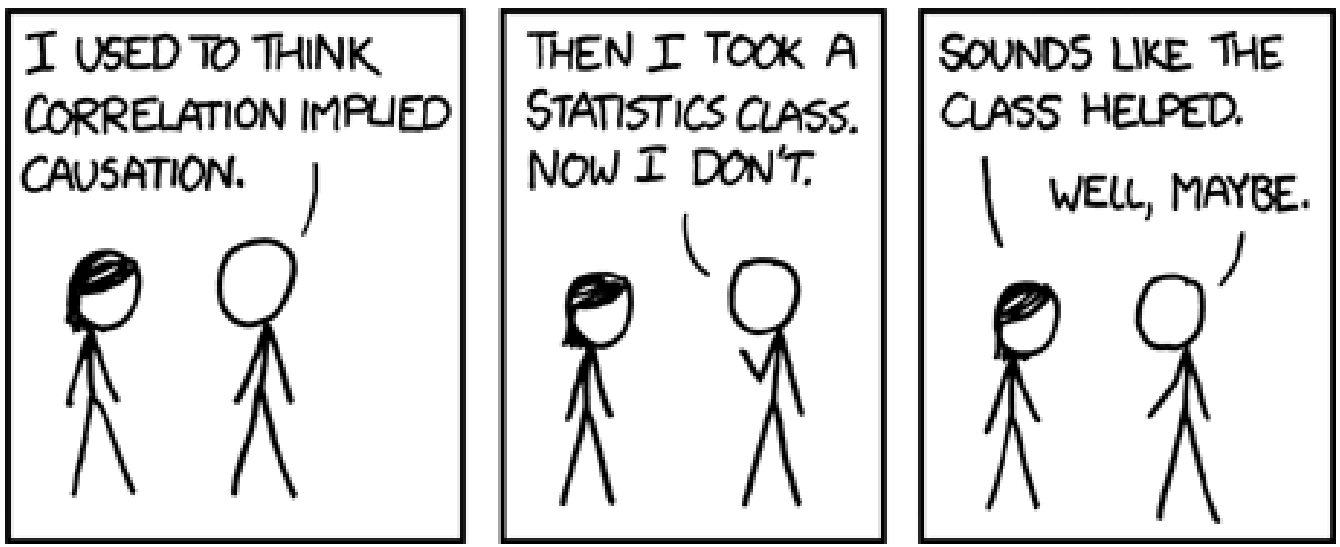
We use a few core principles based on R. Barlow [4]

- There is a distinction and difference between a "systematic check" and a "systematic error"
- Systematic errors can be frequentist or Bayesian
  - Thus we **always**, where possible, quote statistical (which have coverage) and systematic (which don't necessarily) intervals separately
- When we quote a systematic uncertainty we mean that it contains 68.3% of the distribution
- We advise against overestimating systematic uncertainties and simply calling this "conservative"
  - Take the RMS of different models or profile using the envelope method [5]
  - Do not take the range of results but profile or integrate
- Wherever possible make use of the ensemble - compute systematics with MC

Lots of interest in the concept of "uncertainties on uncertainties" *à la* G. Cowan [6] but not yet implemented anywhere in LHCb

Conclusions

- Gave a very brief overview of some aspects of statistical analysis at LHCb
- In HEP "statistics" seems to cover almost all aspects of data analysis - from ML to inference
  - This is probably right because we are really in the business of appropriately propagating errors and ensuring they contain X% of the distribution
- Gave a brief overview of some new insights into sWeights
- Discussed principal methods of inference used at LHCb
- Discussed our "mantra" for computation of systematic uncertainties



References I

[1] Muriel Pivk and Francois R. Le Diberder.
SPlot: A Statistical tool to unfold data distributions.
Nucl. Instrum. Meth. A, 555:356–369, 2005.

[2] Roger J. Barlow.
Extended maximum likelihood.
Nucl. Instrum. Meth. A, 297:496–506, 1990.

[3] Paul E. Condon and Paul L. Cowell.
Channel likelihood: An extension of maximum likelihood for multibody final states.
Phys. Rev. D, 9:2558–2562, May 1974.

[4] Roger Barlow.
Systematic errors: Facts and fictions.
In *Conference on Advanced Statistical Techniques in Particle Physics*, 7 2002.

[5] P. D. Dauncey, M. Kenzie, N. Wardle, and G. J. Davies.
Handling uncertainties in background shapes: the discrete profiling method.
JINST, 10(04):P04015, 2015.

References II

[6] Glen Cowan.
Statistical models with uncertain error parameters.
The European Physical Journal C, 79(2), Feb 2019.

BACKUP

A fresh look at sWeights as orthogonal functions

- Require signal and background components both factorise in the discriminant and control variables
- In other words our total p.d.f. has the form

$$f(m,t) = \underbrace{\boxed{z g_s(m) h_s(t)}}_{\text{Signal}} + \underbrace{\boxed{(1-z) g_b(m) h_b(t)}}_{\text{Background}} \tag{4}$$

- We then want to find a weight function, $w_s(m)$, which when multiplied by $f(m,t)$ projects out $h_s(t)$

$$z h_s(t) = \int w_s(m) f(m,t) dm \tag{5}$$

$$= \int w_s(m) \left[z g_s(m) h_s(t) + (1-z) g_b(m) h_b(t) \right] dm \tag{6}$$

$$= z h_s(t) \int w_s(m) g_s(m) dm + (1-z) h_b(t) \int w_s(m) g_b(m) dm \tag{7}$$

- Therefore we require

$$\underbrace{\boxed{\int w_s(m) g_s(m) dm = 1}}_{w_s(m) \text{ is normal to } g_s(m)} \quad \text{and} \quad \underbrace{\boxed{\int w_s(m) g_b(m) dm = 0}}_{w_s(m) \text{ is orthogonal to } g_b(m)} \tag{8}$$

Choosing the orthonomal functions

- There are infinitely many choices for $w_s(m)$ but choose the one which minimises the variance over the discriminating p.d.f. $g(m)$
- This is a constrained optimisation problem which can be solved with Lagrange multipliers (calculation is in the back up). The solution is

$$w_s(m) = \frac{\alpha_s g_s(m) + \alpha_b g_b(m)}{g(m)}, \tag{9}$$

where the constants α_s and α_b are obtained by solving

$$\begin{pmatrix} W_{ss} & W_{sb} \\ W_{sb} & W_{bb} \end{pmatrix} \cdot \begin{pmatrix} \alpha_s \\ \alpha_b \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{10}$$

where

$$W_{xy} = \int \frac{g_x(m) g_y(m)}{g(m)} dm. \tag{11}$$

- You can then follow this through for any component and generalise to

$$\underbrace{\begin{pmatrix} W_{ss} & W_{sb} \\ W_{sb} & W_{bb} \end{pmatrix}}_{W_{kl}} \cdot \underbrace{\begin{pmatrix} \alpha_s & \beta_s \\ \alpha_b & \beta_b \end{pmatrix}}_{A_{kl}=W_{kl}^{-1}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{12}$$

Application to a finite sample

- The above derivation assumed knowledge of the true p.d.f. to compute the W-matrix
- In practise these are unknown and would be replaced by a sample estimate (typically obtained from a fit)
- The plugin estimate for W is then simply

$$\hat{W}_{xy} = \int \frac{\hat{g}_x(m)\hat{g}_y(m)}{\hat{g}(m)} dm \tag{13}$$

sWeights "integration" method

- This can also be replaced with a sum over observations (for a large sample) because

$$\int \phi(m) dm = \int g(m) \frac{\phi(m)}{g(m)} dm = \underbrace{\left\langle \frac{\phi(m)}{g(m)} \right\rangle}_{\text{expectation value}} \rightarrow \underbrace{\frac{1}{N} \sum_i \frac{\phi(m_i)}{g(m_i)}}_{\text{arithmetic mean}} \tag{14}$$

- So an alternative computation is

$$\hat{W}_{xy} = \frac{1}{N} \sum_i \frac{\hat{g}_x(m)\hat{g}_y(m)}{\hat{g}(m)^2} \tag{15}$$

sWeights "summation" method

- This also has the nice property that the sum of weights is the number of events *i.e.* $\sum_i \hat{w}_s(m_i) = N\hat{z} = \hat{N}_s$

Application to a finite sample

- There is then an interesting connection between the result in Eq. 15 and an extended maximum likelihood fit
- Turns out that the W-matrix is closely related to the covariance matrix of an EML fit with only yields floating

$$\begin{pmatrix} \hat{\alpha}_s & \hat{\beta}_s \\ \hat{\alpha}_b & \hat{\beta}_b \end{pmatrix} = \frac{1}{N^2} \begin{pmatrix} C_{ss} & C_{sb} \\ C_{sb} & C_{bb} \end{pmatrix} \tag{16}$$

sWeights "covariance" method

- Most of the above (at least the finite sample case) was already shown in the sPlot paper [1] although it takes a slightly different approach
- They find the link with the correlation matrix in Eq. (16) and name that as the "sWeight"
  - The implementation in `TSPlot` uses the Minuit / HESSE covariance matrix directly (numerical inaccuracies)
  - The implementation in `RooStats::SPlot` directly computes Eq. (15)