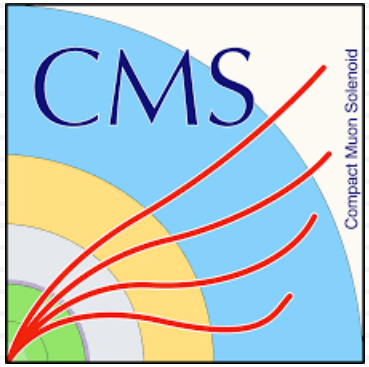
CMS数据分析统计拟合及工具



Jin Wang, Mingshui Chen

Estimation of signal

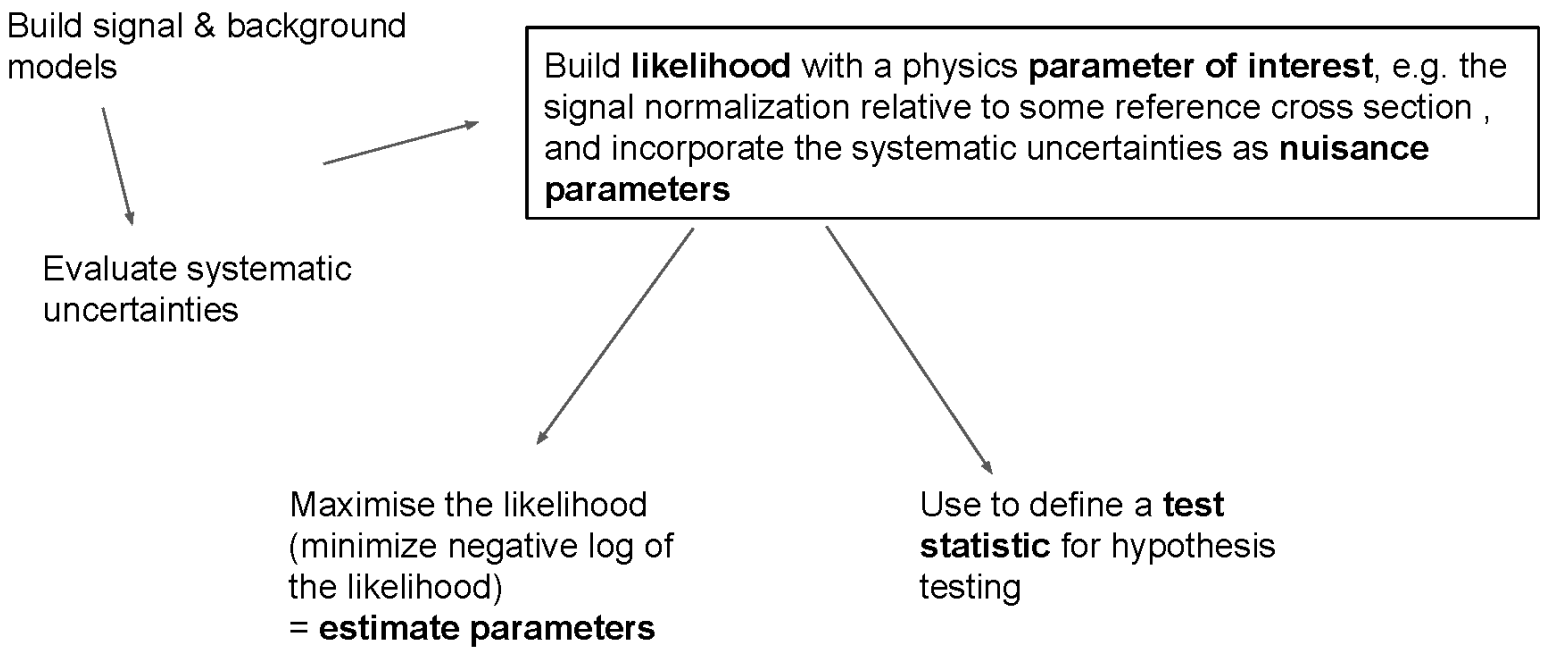
- Give observed data with $N_{data}^{obs.}$ events, and the expected backgrounds $N_b^{exp.}$, the measured number of signal would be
  - $\widehat{N_{sig.}} = N_{data}^{obs.} - N_b^{exp.}$

- Things are much more complex with real HEP data analysis

$$N_{sig.or\ b} = \sigma \times L \times \varepsilon \times A$$

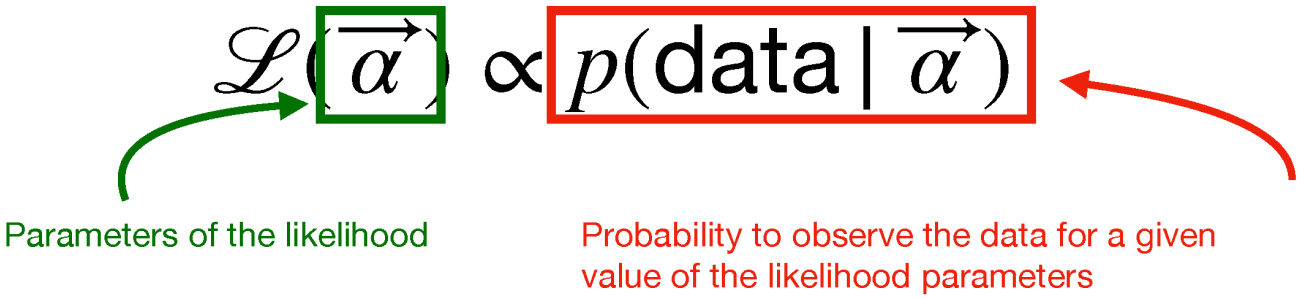
  - σ : cross section of physics process (signal or background)
  - L : integrated luminosity of our dataset
  - ε: efficiency of event selection
  - A : detector acceptance
  - Theoretical, experimental and statistical uncertainties
  - More results desired rather than just number of events
    - cross section, significance, upper limits etc..

Introduction of statistical analysis



Likelihoods

- **Likelihood** defined as



- Note:
  - The likelihood is not a probability (various normalisation terms are ignored)
- Likelihood parameters: $\vec{\alpha} = (\vec{\mu}, \vec{\theta})$

Parameters **o**f **I**nterest (POIs) = parameters we want to measure

Nuisance parameters (or NP)

Nuisance parameters

- The nuisance parameters $\vec{\theta}$ are usually constrained by external measurements (e.g. the luminosity measurement), so we introduce constraint terms

$$\pi(\vec{\theta}_0 \mid \vec{\theta})$$

Measured/nominal value

- The likelihood now is:

$$\mathscr{L}(\vec{\alpha}) \propto p(\text{data} \mid \vec{\alpha}) \cdot \pi(\vec{\theta}_0 \mid \vec{\theta})$$

example:

$$\mathscr{L}(\mu, \theta) = \frac{n_{\text{exp}}^{N} e^{-n\text{exp}}}{N!} e^{-\frac{1}{2}\theta^2}$$

where

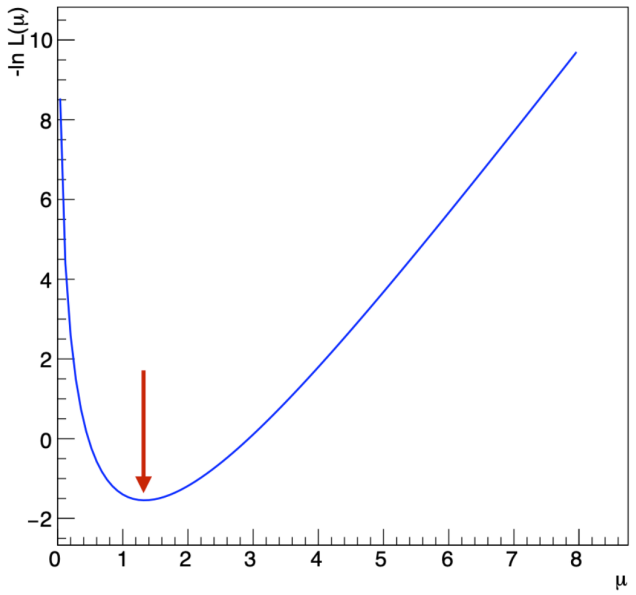
$$n_{\text{exp}} = \mu \sigma_{\text{sig}} \epsilon_{\text{sig}} A_{\text{sig}} L^{\text{int}} 1.025^{\theta} + \sigma_{\text{bkg}} \epsilon_{\text{bkg}} A_{\text{bkg}} L^{\text{int}} 1.025^{\theta}$$

This can of course be extended for multiple bins (product of poisson probabilities) and/or multiple nuisance parameters

Profile likelihood ratio

- We want to maximise the (profiled) likelihood - to avoid dealing with large or small values of the likelihood, we take the **N**egative **L**og of the **L**ikelihood (**NLL**) and **minimise** that instead
- We say that the minimum value of the curve is at $\hat{\mu}$
- Since the value of the likelihood curve at the minimum is not relevant, we can subtract the value at the minimum to obtain, for each value of μ the ΔNLL

$$-\Delta \ln \mathscr{L} = -\ln \mathscr{L}(\mu, \hat{\theta}(\mu)) - (-\ln \mathscr{L}(\hat{\mu}, \hat{\theta}))$$
$$= -\ln \frac{\mathscr{L}(\mu, \hat{\theta}(\mu))}{\mathscr{L}(\hat{\mu}, \hat{\theta})}$$



2x this quantity is known as the **profile likelihood ratio**. We use it as a test statistic for hypothesis testing (e.g. calculating a significance or setting an upper limit).

Hypothesis testing: Significance and Upper Limits

- Significance: exclude background only hypothesis
  - With current observed data, the possibility of signal s=0
    - Often relate significance with standard deviation of Gaussian “N sigma”

significance	1	2	3	4	5
probability (p-value)	16%	2.3%	0.14%	3×10^{-5}	3×10^{-7}

    - 3 sigma: evidence, 5 sigma: observation
    - Approximation: s/sqrt(b), or $\sqrt{2n_0\ln(1+s/b)-2s}$ with observed n_0 events
- Upper limits: exclude signal+background hypothesis with $s>s_x$
  - with current observed data, how much confidence we have that signal is with $s<=s_x$
    - Often use 95% Confidence Level (C.L.) upper limit

More in Mingshui’s talk

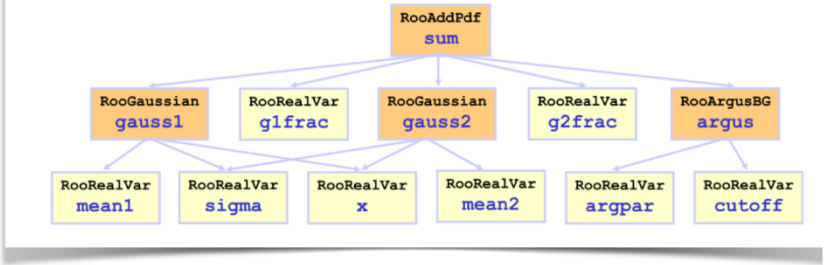
- With profiled likelihood ratio, we can determine the distribution of the test-statistics in the “Asymptotic limit”
  - https://arxiv.org/pdf/1007.1727.pdf

RooFit

- Framework built on top of ROOT for statistical analysis
- Objected-oriented approach
  - Specific PDFs deriving from abstract base classes, e.g. **`RooGaussian`** from **`RooAbsPdf`**
- Construct mathematical models by connecting objects together
- Provides interfaces for fitting and visualisation

Mathematical concept		RooFit class
variable	x	`RooRealVar`
function	$f(x)$	`RooAbsReal`
PDF	$f(x)$	`RooAbsPdf`
space point	$\vec{x}$	`RooArgSet`
integral	$\int_{x_{min}}^{x_{max}} f(x)dx$	`RooRealIntegral`
list of space points		`RooAbsData`

W. Verkerke



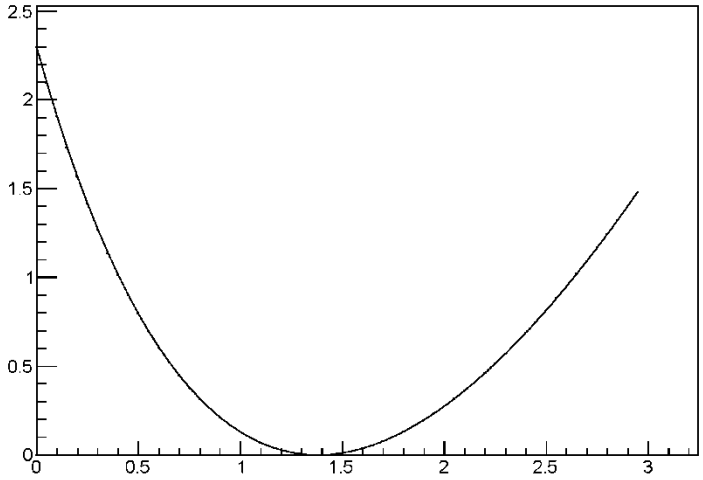
Roostat and CMS Combine Tool

- CMS Combine Tool: RooStats / RooFit - based software tools used for statistical analysis in CMS
  - It provides a command line interface to many different statistical techniques available inside RooFit/RooStats

- Download and setup
  - cp –r /data/pubfs/pku_visitor/wangjin/statistics/ .
  - cd statistics
  - source 00_downloadandsetup.sh
    - #git clone https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit.git HiggsAnalysis/CombinedLimit
    - cd HiggsAnalysis/CombinedLimit/
    - #git fetch origin
    - #git checkout v8.1.0
    - . env_standalone.sh
    - make

Counting experiment fit with workspace

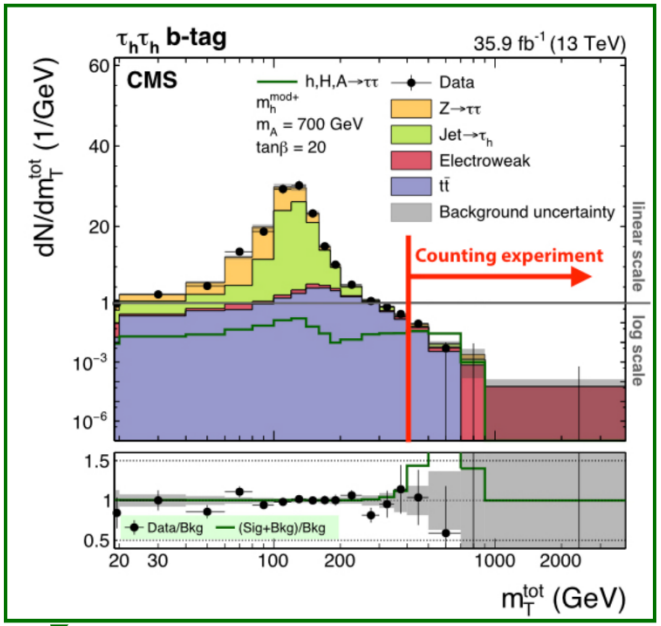
- Construction likelihood for simple counting measurement
  - w = RooWorkspace("w")
  - w.factory('expr::n("mu*s +b", mu[1.0,0,4], s[5],b[8.1])')
  - w.factory('Poisson::poisN(N[15],n)')
  - w.factory('expr::NLL("-log(@0)",poisN)')
    - nll = w.function("NLL")
    - minim = RooMinimizer(nll)
    - minim.setErrorLevel(0.5)
    - minim.minimize("Minuit2","migrad")
    - bestfitnll = nll.getVal()
- Perform fits for different values of mu
  - profiling nuisance parameters
  - get profiled likelihood ratio
    - deltanll = tmpfitnll-bestfitnll
- example:
  - 01_workspace.py, 02_simpleFit.py
  - Use python to run these examples



How to get uncertainty and significance from this?

Exercise with CMS combine tool

- Switching to Combine - you won't have to build the likelihood yourself!
- Using a simplified version of the MSSM H→ττ analysis for these exercises
- Events split into categories targeting the main di-tau final states, and two main production modes of BSM Higgs bosons: gluon fusion and b-associated production
- Example: backgrounds and expected signal in the fully hadronic final state for the category targeting b-associated production
- In this session we'll consider both a counting experiment in the high mass region & a shape analysis



Datacards

- Wrap all the workspace ingredients in the datacards

Datacard format - counting

Number of bins/channels

Number of processes

Number of nuisance parameters (*:determined automatically)

```
imax    1 number of bins
jmax    4 number of processes minus 1
kmax    * number of nuisance parameters
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
bin          signal_region
observation  10.0
--------------------------------------------------------------------------------
bin                     signal_region  signal_region  signal_region  signal_region  signal_region
process                 ttbar          diboson        Ztautau        jetFakes       bbHtautau
process                 1              2              3              4              0
rate                    4.43803        3.18309        3.7804         1.63396        0.711064
--------------------------------------------------------------------------------
CMS_eff_b          lnN  1.02           1.02           1.02           -              1.02
CMS_eff_t          lnN  1.12           1.12           1.12           -              1.12
CMS_eff_t_highpt   lnN  1.1            1.1            1.1            -              1.1
acceptance_Ztautau lnN  -              -              1.08           -              -
acceptance_bbH     lnN  -              -              -              -              1.05
acceptance_ttbar   lnN  1.005          -              -              -              -
lumi_13TeV         lnN  1.025          1.025          1.025          -              1.025
norm_jetFakes      lnN  -              -              -              1.2            -
xsec_Ztautau       lnN  -              -              1.04           -              -
xsec_diboson       lnN  -              1.05           -              -              -
xsec_ttbar         lnN  1.06           -              -              -              -
```

Unique channel label

Number of observed events in channel

Process label

Process ID (<=0 for signal)

Expected number of events

Systematic uncertainties

Name

Type

Effect on process

4

The FitDiagnostics method

- **FitDiagnostics** returns the best-fit value of the POI + uncertainty
  - Also gives additional information about the model
- Two fits are performed:
  - "background-only" fit: the POI (r) is fixed to zero
  - "signal+background" fit: the POI is floating
  - NB in case of multiple POIs, only the first one is frozen in the b-only fit
- Covariance matrix is saved for both fits
- FitDiagnostics can also produce pre- and post-fit distributions & their uncertainties

Commands:

```
combine -M FitDiagnostics inputs/datacard_counting_part2.txt -- forceRecreateNLL
```

Example: source 03_combineTool_simpleFit.sh

You can see the full set of supported options by doing "combine –h"

Datacard with shapes

- For more complex analysis, you could have many bins and many systematics
  - combine tool could read the histograms of multiple bins as inputs

Some additional information in the datacard for a shape-based analysis
- Link to the shapes in a ROOT file
- Addition of shape uncertainties

```
imax 1
jmax 1
kmax *
---------------
shapes * * simple-shapes-TH1_input.root $PROCESS $PROCESS_$SYSTEMATIC
shapes signal * simple-shapes-TH1_input.root $PROCESS$MASS $PROCESS$MASS_$SYSTEMATIC
---------------
bin bin1
observation 85
-------------------------------
bin             bin1       bin1
process         signal     background
process         0          1
rate            10         100
-------------------------------
lumi     lnN    1.10       1.0
bgnorm   lnN    1.00       1.3
alpha  shape    -          1
```

Links to the histograms saved in a root file

Shape uncertainty: 2 additional histograms per process supplied, with ±1σ shift

The example shape datacards


```
imax    1 number of bins
jmax    4 number of processes minus 1
kmax    * number of nuisance parameters
--------------------------------------------------------------------------------
shapes * signal_region datacard_part3.shapes.root signal_region/$PROCESS signal_region/$PROCESS_$SYSTEMATIC
shapes bbHtautau signal_region datacard_part3.shapes.root signal_region/bbHtautau$MASS signal_region/bbHtautau$MASS_$SYSTEMATIC
--------------------------------------------------------------------------------
bin          signal_region
observation  3416.0
--------------------------------------------------------------------------------
bin                                 signal_region   signal_region   signal_region   signal_region   signal_region
process                             ttbar           diboson         Ztautau         jetFakes        bbHtautau
process                             1               2               3               4               0
rate                                683.017         96.5185         742.649         2048.94         198.521
--------------------------------------------------------------------------------
CMS_eff_b                   lnN     1.02            1.02            1.02            -               1.02
CMS_eff_t                   lnN     1.12            1.12            1.12            -               1.12
CMS_eff_t_highpt            shape   1               1               1               -               1
CMS_scale_t_1prong0pi0_13TeV shape  1               1               1               -               1
CMS_scale_t_1prong1pi0_13TeV shape  1               1               1               -               1
CMS_scale_t_3prong0pi0_13TeV shape  1               1               1               -               1
acceptance_bbH              lnN     -               -               -               -               1.05
lumi_13TeV                  lnN     1.025           1.025           1.025           -               1.025
norm_jetFakes               lnN     -               -               -               1.2             -
top_pt_ttbar_shape          shape   1               -               -               -               -
xsec_diboson                lnN     -               1.05            -               -               -
xsec_Ztautau                lnN     -               -               1.04            -               -
acceptance_Ztautau          lnN     -               -               1.08            -               -
acceptance_ttbar            lnN     1.005           -               -               -               -
xsec_ttbar                  lnN     1.06            -               -               -               -
* autoMCStats 0
~
~
```

Process ID
B: >0
S: <=0

Create workspace from the datacards

text2workspace.py inputs/datacard_part3.txt -m 200 -o outputs/workspace_part3.root

combine -M FitDiagnostics --rMin -20 --rMax 20 outputs/workspace_part3.root -m 200

root -l workspace_part3.root

w->Print()

Content of workspace_part3.root

ProcessNormalization::n_exp_binsignal_region_proc_Ztautau[thetaList=(CMS_eff_b,CMS_eff_t,lumi_13TeV,xsec_Ztautau,acceptance_Ztautau) asymmThetaList=() otherFactorList=()] = 1

ProcessNormalization::n_exp_binsignal_region_proc_bbHtautau[thetaList=(CMS_eff_b,CMS_eff_t,acceptance_bbH,lumi_13TeV) asymmThetaList=() otherFactorList=(r)] = 1

ProcessNormalization::n_exp_binsignal_region_proc_diboson[thetaList=(CMS_eff_b,CMS_eff_t,lumi_13TeV,xsec_diboson) asymmThetaList=() otherFactorList=()] = 1

ProcessNormalization::n_exp_binsignal_region_proc_jetFakes[thetaList=(norm_jetFakes) asymmThetaList=() otherFactorList=()] = 1

POI:(r)

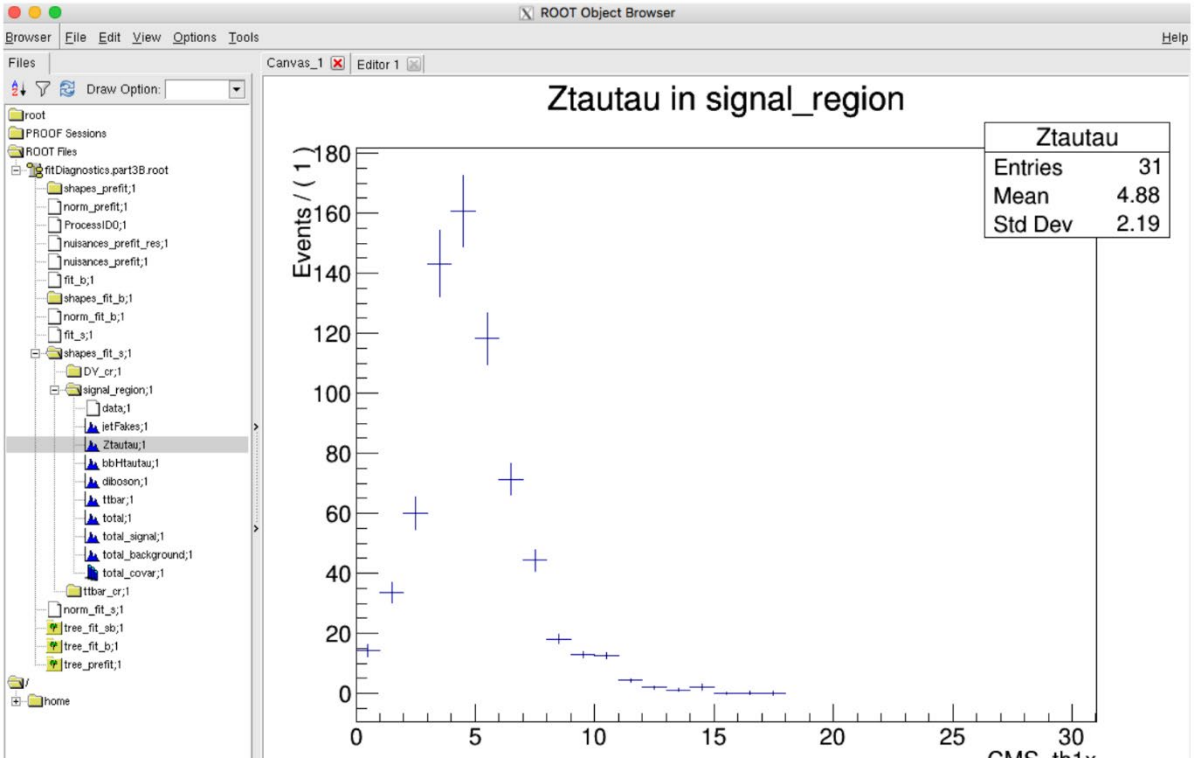
example:

source 04_combineTool_shape.sh

Prefit and postfit shapes with uncertainties

- FitDiagnostics mode can help us visualise the distributions we are fitting, and the uncertainties on those distributions, both before the fit is performed ("pre-fit") and after ("post-fit").
  - combine -M FitDiagnostics workspace_part3.root -m 200 --rMin -1 --rMax 2 --saveShapes --saveWithUncertainties

Combine will produce pre- and post-fit distributions (for fit_s and fit_b) in the fitdiagnostics.root output file:



Scale of signal with physics model

CombinedLimit/python/PhysicsModel.py

```
class PhysicsModelBase(object):
    __metaclass__ = ABCMeta
    def __init__(self):
        pass
    def setModelBuilder(self, modelBuilder):
        "Connect to the ModelBuilder to get workspac
        self.modelBuilder = modelBuilder
        self.DC = modelBuilder.DC
        self.options = modelBuilder.options
    def setPhysicsOptions(self,physOptions):
        "Receive a list of strings with the physics options from command line"
    @abstractmethod
    def doParametersOfInterest(self):
        """Create POI and other parameters, and define the POI set."""
    def preProcessNuisances(self,nuisances):
        "receive the usual list of (name,nofloat,pdf,args,errline) to be edited"
        pass # do nothing by default
    def getYieldScale(self,bin,process):
        "Return the name of a RooAbsReal to scale this yield by or the two special values 1 and 0 (don't scale, and set to zero)"
        return "r" if self.DC.isSignal[process] else 1;
```

```
class PhysicsModel(PhysicsModelBase):
    """Example class with signal strength as only POI"""
    def doParametersOfInterest(self):
        """Create POI and other parameters, and define the POI set."""
        self.modelBuilder.doVar("r[1,0,20]");
        self.modelBuilder.doSet("POI","r")
```

Build variables and POI

How yield gets scaled

Rescale events with parameterization

Example:

With interference

$$N = |\sqrt{\mu}*s+b|^2 = \mu*|s|^2 + |b|^2 + \sqrt{\mu}*|s|*|b|*\cos\Theta = \mu*S + B + \sqrt{\mu}*I$$

Way 2: put S, B, SBI separately in the datacards, and parametrize I by linear combinations

$$I = SBI - S - B$$

$$N = \mu*S + B + \sqrt{\mu}*I = \mu*S + B + \sqrt{\mu}*(SBI - S - B)$$

$$= (\mu-\sqrt{\mu})*S + (1-\sqrt{\mu})*B + \sqrt{\mu}*SBI$$

```
mumueq0jets   mumueq0jets   mumueq0jets
ggH_b         qqH_s         ggH_sbi
-5            -4            -3
13.0665       0.0170        12.2213
```

In the datacard

```
def getYieldScale(self,bin,process):
    if process == "ggH_sonl": return "ggH_s_func"
    elif process == "ggH_bonl": return "ggH_b_func"
    elif process == "ggH_sand": return "ggH_sbi_func"
```

In the physics model

```
self.modelBuilder.factory_( "expr::ggH_s_func(\"@0-sqrt(@0)\", CMS_zz2l2nu_mu)")
self.modelBuilder.factory_(  "expr::ggH_b_func(\"1-sqrt(@0)\", CMS_zz2l2nu_mu)")
self.modelBuilder.factory_(  "expr::ggH_sbi_func(\"sqrt(@0)\", CMS_zz2l2nu_mu)")
```

Modify workspace with new models

- To use a different physics model instead of the default one, use the option -P

text2workspace.py datacard -P HiggsAnalysis.CombinedLimit.PythonFile:modelName

Open the workspace, will see the contents below

```
functions
--------
RooFormulaVar::CMS_zz4l_mu[ actualVars=(r) formula="@0*0.0673*0.2*2/1000./0.012722" ] = 0.00211602
RooFormulaVar::ggH_b_func[ actualVars=(CMS_zz4l_mu) formula="1-sqrt(@0)" ] = 0.954
RooFormulaVar::ggH_s_func[ actualVars=(CMS_zz4l_mu) formula="@0-sqrt(@0)" ] = -0.0438842
RooFormulaVar::ggH_sbi_func[ actualVars=(CMS_zz4l_mu) formula="sqrt(@0)" ] = 0.0460002
ProcessNormalization::n_exp_bineeeq0jets_proc_ggH_bonl[ thetaList=() asymmThetaList=() otherFactorList=(ggH_b_func) ] = 10.0352
ProcessNormalization::n_exp_bineeeq0jets_proc_ggH_sand[ thetaList=() asymmThetaList=() otherFactorList=(ggH_sbi_func) ] = 1.86604
ProcessNormalization::n_exp_bineeeq0jets_proc_ggH_sonl[ thetaList=() asymmThetaList=() otherFactorList=(ggH_s_func) ] = -1.31079
```

Extract limits

- Combine has dedicated method for calculating upper limits
  - exclude POI at a given confidence level
  - most commonly use AsymptoticLimits
    - implements the CLs criterion and uses the profile likelihood ratio as the test statistic
    - the test statistic distributions are determined analytically in the asymptotic approximation

```
combine -M AsymptoticLimits datacard_counting_part2.txt
```

- Gives the result:

```
 -- AsymptoticLimits ( CLs ) --
Observed Limit: r < 10.8183
Expected  2.5%: r < 7.0537
Expected 16.0%: r < 9.8108
Expected 50.0%: r < 14.5625
Expected 84.0%: r < 22.3988
Expected 97.5%: r < 33.5971
```

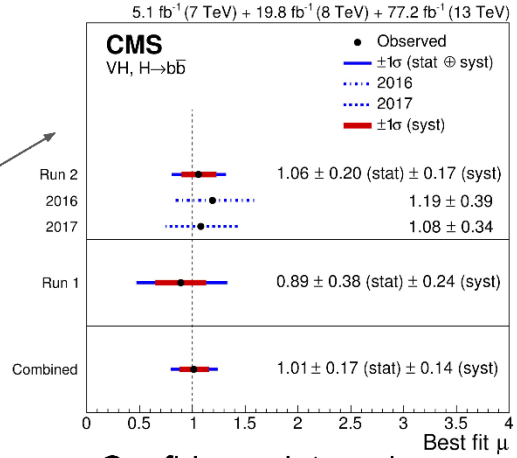
- But what did combine actually do? Can get a better idea by running with a higher verbosity level:

```
combine -M AsymptoticLimits datacard.txt -v 2
```

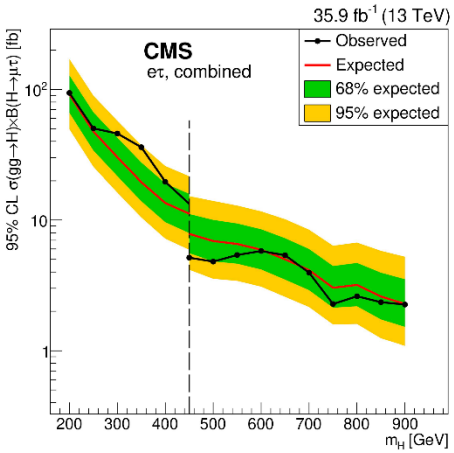
A lot more functions and outputs

You can find more tutorials here:
https://github.com/FNALLPC/statistics-das/blob/master/README.md

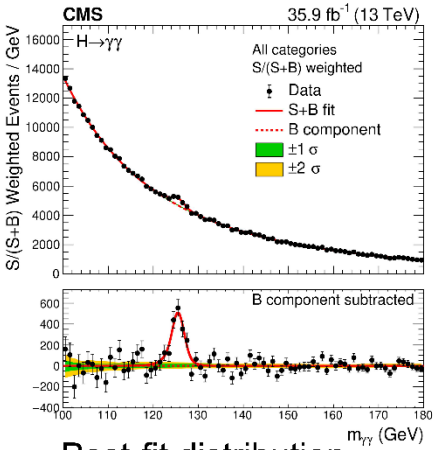
More outputs from CMS combine tool



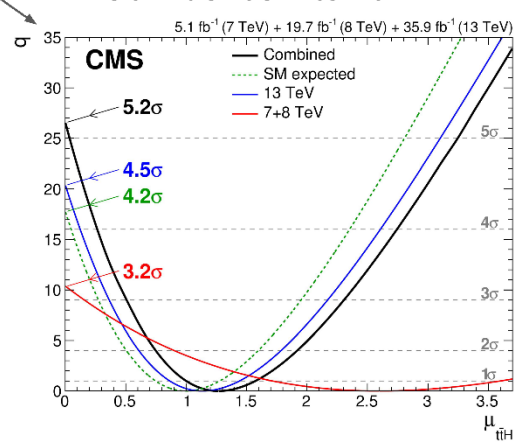
Confidence interval



Upper limit



Post-fit distribution



Significance / confidence interval

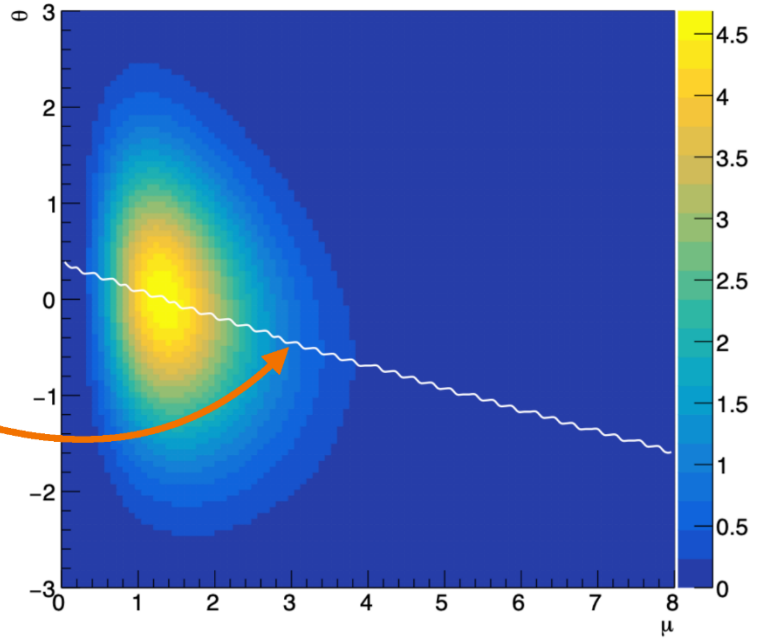
exercise scripts: /data/pubfs/pku_visitor/wangjin/statistics/

Backup

Profiling

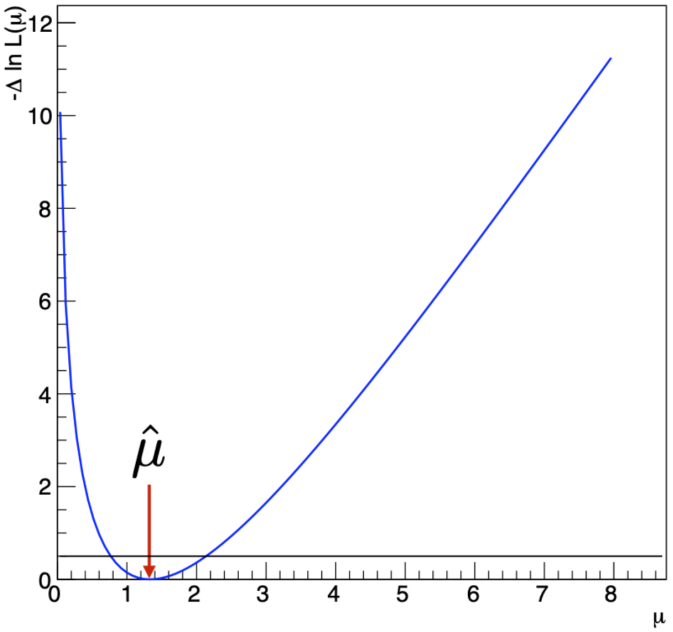
- By **profiling over the nuisance parameters** we mean to find the values of the nuisance parameters which maximise the likelihood for each value of the parameter of interest.
- Take the example 2D likelihood for a model with one POI and one NP
- The **profiled likelihood** is the value of the likelihood function along the line $\hat{\theta}(\mu)$

$$\mathscr{L}(\mu) = \mathscr{L}(\mu, \hat{\theta}(\mu)) \equiv \max_{\theta} \mathscr{L}(\mu, \theta)$$



Obtaining a confidence interval

- According to Wilks' theorem, in the limit of large sample sizes, 2 x the -ve log of the ratio of likelihoods is distributed as a χ^2 with N degrees of freedom, where N is the difference in number of free parameters between the numerator and denominator of the likelihood ratio (ie, 1 in our case)
  - Then we can just use the quantile function of the χ^2 distribution to see that for a 68% confidence interval, -2x $\Delta NLL < 1$ ➔ $-\Delta NLL < 0.5$



Significance and compatibility

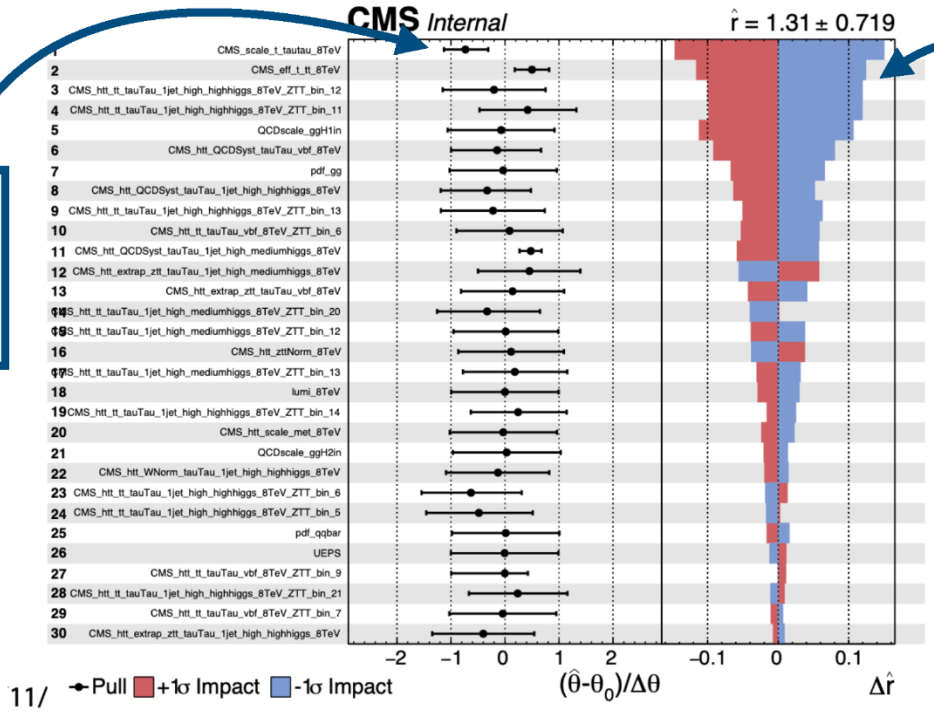
- P value for significance
  - cout<<endl<<"Test statistic null qmu is "<<qmu<<", minpoi is "<<muhat<<endl;
  - if(muhat<0) qmu=-qmu;
  - int sign=int(qmu==0 ? 0 : fabs(qmu)/qmu);
  - double Z=sign*sqrt(fabs(qmu));
  - return 1-ROOT::Math::gaussian_cdf(Z);

- P value for 1D compatibility
  - double pvalue_one = ROOT::Math::chisquared_cdf_c(dnll_one,ndof_one);

Impact plots

Impacts

- Define the **impact** of a nuisance parameter on the POI as the shift in the POI that is induced as the NP is fixed and brought to its +1σ or -1σ post-fit values



Also see the parameter constraint relative to the input uncertainty

Size of the bar ∝ Impact on "r"