

Introduction to the Combine Tool

30th Nov. 2016

Andrew Gilbert (KIT)

David Sperka (University of Florida)

Nick Wardle (CERN)

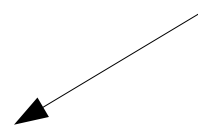
Overview of Combine

- Combine is a RooStats based command line tool which executes different statistical methods (compute limits/significance, make fits, etc.)
- Internally, there are two main components
 - text2workspace: a python module which converts a textual datacard into a RooStats model (“workspace”)
 - Datacards can also be used to load user created workspaces
 - C++ code which is responsible for the statistical methods
- Highly customizable: anything which is possible in RooStats is possible in the Combine package
- Supported by a HIG PAG Subgroup: Hcomb

[Much more information on the Combine Tool Twiki](#)

The commands in the blue boxes on each slide provide a hands on commands which can be run to follow along. You may also use the SWAN notebook `combine_intro/python_CombineIntro.ipynb`

```
cd HiggsAnalysis/CombinedLimit/combine_tutorials_2016/combine_intro/  
./get_ws.sh
```



Datacard: Simple Counting Experiment



```
# Simple counting experiment, with one signal and one background process
# Extremely simplified version of the 35/pb H->WW analysis for mH = 200 GeV,
# for 4th generation exclusion (EWK-10-009, arxiv:1102.5429v1)
imax 1  number of channels
jmax 1  number of backgrounds
kmax 2  number of nuisance parameters (sources of systematical uncertainties)
-----
# we have just one channel, in which we observe 0 events
bin      1
observation 0
-----
# now we list the expected events for signal and all backgrounds in that bin
# the second 'process' line must have a positive number for backgrounds, and 0 for signal
# then we list the independent sources of uncertainties, and give their effect (syst. error)
# on each process and bin
bin      1      1
process  ggh4G  Bckg
process  0      1
rate     4.76  1.47
-----
deltaS   lnN    1.20    -    20% uncertainty on signal
deltaB   lnN     -    1.50    50% uncertainty on background
```

- The “datacard” is a textual input which combine converts into a RooStats model (“workspace”)

```
cat simple-counting-experiment.txt
```

Datacard: Simple Counting Experiment



```
# Simple counting experiment, with one signal and one background process
# Extremely simplified version of the 35/pb H->WW analysis for mH = 200 GeV,
# for 4th generation exclusion (EWK-10-009, arxiv:1102.5429v1)
imax 1  number of channels
jmax 1  number of backgrounds
kmax 2  number of nuisance parameters (sources of systematical uncertainties)
-----
# we have just one channel, in which we observe 0 events
bin      1
observation 0
-----
# now we list the expected events for signal and all backgrounds in that bin
# the second 'process' line must have a positive number for backgrounds, and 0 for signal
# then we list the independent sources of uncertainties, and give their effect (syst. error)
# on each process and bin
bin      1      1
process   ggh4G  Bckg
process    0      1
rate      4.76  1.47
-----
deltaS  lnN    1.20    -    20% uncertainty on signal
deltaB  lnN      -    1.50    50% uncertainty on background
```

- First block specifies the number of channels, backgrounds, nuisances
- Can also use “*” and combine can determine on the fly, but specifying explicitly can help spot mistakes

```
cat simple-counting-experiment.txt
```

Datacard: Simple Counting Experiment



```
# Simple counting experiment, with one signal and one background process
# Extremely simplified version of the 35/pb H->WW analysis for mH = 200 GeV,
# for 4th generation exclusion (EWK-10-009, arxiv:1102.5429v1)
imax 1  number of channels
jmax 1  number of backgrounds
kmax 2  number of nuisance parameters (sources of systematical uncertainties)
-----
# we have just one channel, in which we observe 0 events
bin          1
observation 0
-----
# now we list the expected events for signal and all backgrounds in that bin
# the second 'process' line must have a positive number for backgrounds, and 0 for signal
# then we list the independent sources of uncertainties, and give their effect (syst. error)
# on each process and bin
bin          1      1
process      ggh4G  Bckg
process      0      1
rate         4.76  1.47
-----
deltaS  lnN    1.20    -    20% uncertainty on signal
deltaB  lnN     -    1.50    50% uncertainty on background
```

- ↓
- Second block labels the channel (here “1”, can be any string)
 - Specifies the number of of observed events

```
cat simple-counting-experiment.txt
```

Datacard: Simple Counting Experiment



```
# Simple counting experiment, with one signal and one background process
# Extremely simplified version of the 35/pb H->WW analysis for mH = 200 GeV,
# for 4th generation exclusion (EWK-10-009, arxiv:1102.5429v1)
imax 1  number of channels
jmax 1  number of backgrounds
kmax 2  number of nuisance parameters (sources of systematical uncertainties)
-----
# we have just one channel, in which we observe 0 events
bin      1
observation 0
-----
# now we list the expected events for signal and all backgrounds in that bin
# the second 'process' line must have a positive number for backgrounds, and 0 for signal
# then we list the independent sources of uncertainties, and give their effect (syst. error)
# on each process and bin
bin      1      1
process   ggh4G  Bckg
process    0      1
rate      4.76  1.47
-----
deltaS  lnN    1.20    -    20%  uncertainty on signal
deltaB  lnN     -    1.50    50%  uncertainty on background
```

- Third block specifies the number of expected events (“rate”) for each process
- Two process lines: one gives a label to each process, and the second a number which if ≤ 0 denotes signal and if > 0 denotes background
 - Signal processes will be given a free floating normalization parameter

```
cat simple-counting-experiment.txt
```


Datacard: Simple Counting Experiment



```
# Simple counting experiment, with one signal and one background process
# Extremely simplified version of the 35/pb H->WW analysis for mH = 200 GeV,
# for 4th generation exclusion (EWK-10-009, arxiv:1102.5429v1)
imax 1  number of channels
jmax 1  number of backgrounds
kmax 2  number of nuisance parameters (sources of systematical uncertainties)
-----
# we have just one channel, in which we observe 0 events
bin      1
observation 0
-----
# now we list the expected events for signal and all backgrounds in that bin
# the second 'process' line must have a positive number for backgrounds, and 0 for signal
# then we list the independent sources of uncertainties, and give their effect (syst. error)
# on each process and bin
bin      1      1
process   ggh4G  Bckg
process    0      1
rate      4.76  1.47
-----
deltaS  lnN      1.20      -      20% uncertainty on signal
deltaB  lnN      -      1.50      50% uncertainty on background
```

- Final block specifies the nuisance parameters affecting the processes
- A label and prior distribution (e.g. Log-normal, Gamma, Uniform) are given
- Nuisances with different names are uncorrelated (and a single nuisance is correlated across all processes which it affects)

```
cat simple-counting-experiment.txt
```

From Datacard to RooWorkspace

- The text2workspace.py script converts the textual datacard into a RooWorkspace, defining the Likelihood function used for the statistical methods
- The RooWorkspace contains variables, pdf's, functions, datasets, etc.
- One can inspect the workspace to see how the likelihood has been constructed

```

RooWorkspace(w) w contents
-----
variables
-----
(deltaB,deltaB_In,deltaS,deltaS_In,n_obs_binbin1,r)

p.d.f.s
-----
SimpleGaussianConstraint::deltaB_Pdf[ x=deltaB mean=deltaB_In sigma=1 ] = 1
SimpleGaussianConstraint::deltaS_Pdf[ x=deltaS mean=deltaS_In sigma=1 ] = 1
RooProdPdf::modelObs_b[ pdf_binbin1_bonly ] = 0.229925
RooProdPdf::modelObs_s[ pdf_binbin1 ] = 0.00196945
RooProdPdf::model_b[ modelObs_b * nuisancePdf ] = 0.229925
RooProdPdf::model_s[ modelObs_s * nuisancePdf ] = 0.00196945
RooProdPdf::nuisancePdf[ deltaS_Pdf * deltaB_Pdf ] = 1
RooPoisson::pdf_binbin1[ x=n_obs_binbin1 mean=n_exp_binbin1 ] = 0.00196945
RooPoisson::pdf_binbin1_bonly[ x=n_obs_binbin1 mean=n_exp_binbin1_bonly ] = 0.229925

functions
-----
RooAddition::n_exp_binbin1[ n_exp_binbin1_proc_ggh4G + n_exp_binbin1_proc_Bckg ] = 6.23
RooAddition::n_exp_binbin1_bonly[ n_exp_binbin1_proc_Bckg ] = 1.47
ProcessNormalization::n_exp_binbin1_proc_Bckg[ thetaList=(deltaB) asymmThetaList=( ) otherFactorList=( ) ] = 1.47
ProcessNormalization::n_exp_binbin1_proc_ggh4G[ thetaList=(deltaS) asymmThetaList=( ) otherFactorList=(r) ] = 4.76

datasets
-----
RooDataSet::data_obs(n_obs_binbin1)

named sets
-----
ModelConfig_GlobalObservables:(deltaS_In,deltaB_In)
ModelConfig_NuisParams:(deltaS,deltaB)
ModelConfig_Observables:(n_obs_binbin1)
ModelConfig_POI:(r)
ModelConfig_bonly_GlobalObservables:(deltaS_In,deltaB_In)
ModelConfig_bonly_NuisParams:(deltaS,deltaB)
ModelConfig_bonly_Observables:(n_obs_binbin1)
ModelConfig_bonly_POI:(r)
POI:(r)
globalObservables:(deltaS_In,deltaB_In)
nuisances:(deltaS,deltaB)
observables:(n_obs_binbin1)

generic objects
-----
RooStats::ModelConfig::ModelConfig
RooStats::ModelConfig::ModelConfig_bonly
RooArgSet::discreteParams
    
```

```

text2workspace.py simple-counting-experiment.txt
root -l simple-counting-experiment.root
root [1] RooWorkspace* w = (RooWorkspace*)_file0->Get("w")
root [2] w->Print()
    
```


Likelihood From the RooWorkspace

- Lets start from the pdf called `model_s` in the workspace, which is the full likelihood including the signal, and expand as much as we can:
 → For brevity, I won't always copy the full name of every object

```
L = model_s
```

```
L = modelObs_s * nuisancePdf
```

```
L = pdf_binbin1 * deltaS_Pdf * deltaB_Pdf
```

```
L = Poisson[x=n_obs mean=n_exp]
    * Gauss[x=deltaS mean=deltaS_In sigma=1] * Gauss[x=deltaB mean=deltaB_In sigma=1]
```

```
L = Poisson[x=n_obs mean=(n_exp_ggh4G + n_exp_Bckg)]
    * Gauss[x=deltaS mean=0 sigma=1] * Gauss[x=deltaB mean=0 sigma=1]
```

```
L = Poisson[x=n_obs mean=( n_nom_ggh4G*f(deltaB)*r + n_nom_Bckg*f(deltaS))]
    * Gauss[x=deltaS mean=0 sigma=1] * Gauss[x=deltaB mean=0 sigma=1]
```

- Which is the single bin example of a generic likelihood function:

$$L(r, \vec{\theta}) = \prod_i \frac{[r \cdot s_i(\vec{\theta}) + b_i(\vec{\theta})]^{n_i}}{n_i!} e^{-[r \cdot s_i(\vec{\theta}) + b_i(\vec{\theta})]} \prod_{\kappa} e^{-\frac{1}{2} \theta_{\kappa}^2}$$

- It has three parameters: `r`, `deltaS`, and `deltaB`. “`r`” is the POI and is unconstrained (except by the observed data), while `deltaS` and `deltaB` are nuisance parameters which have external constraints. These three parameters are jointly fitted to get the value of “`r`”.

Datacard: Realistic Counting Experiment



```
# Simple counting experiment, with one signal and a few background processes
# Simplified version of the 35/pb H->WW analysis for mH = 160 GeV
imax 1  number of channels
jmax 3  number of backgrounds
kmax 5  number of nuisance parameters (sources of systematical uncertainties)
-----
# we have just one channel, in which we observe 0 events
bin 1
observation 0
-----
# now we list the expected events for signal and all backgrounds in that bin
# the second 'process' line must have a positive number for backgrounds, and 0 for signal
# then we list the independent sources of uncertainties, and give their effect (syst. error)
# on each process and bin
bin          1      1      1      1
process      ggH   qqWW  ggWW  others
process      0     1     2     3
rate         1.47  0.63  0.06  0.22
-----
lumi    lnN    1.11    -    1.11    -    lumi affects both signal and gg->WW (mc-driven). lnN = lognormal
xs_ggH  lnN    1.16    -    -      -    gg->H cross section + signal efficiency + other minor ones.
WW_norm gmN 4   -    0.16    -    -    WW estimate of 0.64 comes from sidebands: 4 events in sideband times 0.16
xs_ggWW lnN    -     -    1.50    -    50% uncertainty on gg->WW cross section
bg_others lnN  -     -     -    1.30  30% uncertainty on the rest of the backgrounds
```

- Realistic datacards will have more processes and nuisance parameters, but we will end up with a similar likelihood function:

$$L(r, \vec{\theta}) = \prod_i \frac{[r \cdot s_i(\vec{\theta}) + b_i(\vec{\theta})]^{n_i}}{n_i!} e^{-[r \cdot s_i(\vec{\theta}) + b_i(\vec{\theta})]} \prod_{\kappa} e^{-\frac{1}{2} \theta_{\kappa}^2}$$

```
cat ../../CombineLimit/data/tutorials/realistic-counting-experiment.txt
```

Combination of Multiple Channels

```
imax 3 number of channels
jmax * number of backgrounds ('*' = automatic)
kmax * number of nuisance parameters (sources of systematical uncertainties)
-----
# three channels, each with it's number of observed events
bin          e tau mu tau e mu
observation  517  540  101
-----
# now we list the expected events for signal and all backgrounds in those three bins
# the second 'process' line must have a positive number for backgrounds, and 0 for signal
# for the signal, we normalize the yields to an hypothetical cross section of 1/pb
# so that we get an absolute limit in cross section in units of pb.
# then we list the independent sources of uncertainties, and give their effect (syst. error)
# on each process and bin
```

bin	e tau	e tau	e tau	mu tau	mu tau	mu tau	e mu	e mu	e mu
process	higgs	ZTT	QCD	higgs	ZTT	QCD	higgs	ZTT	other
process	0	1	2	0	1	2	0	1	2
rate	0.34	190	327	0.57	329	259	0.15	88	14
lumi	lnN	1.11	-	1.11	-	-	1.11	-	1.11
tauid	lnN	1.23	1.23	1.23	1.23	-	-	-	-
ZtoLL	lnN	-	1.04	-	1.04	-	-	1.04	-
effic	lnN	1.04	1.04	1.04	1.04	-	1.04	1.04	1.04
QCDel	lnN	-	-	-	-	-	-	-	-
QCDmu	lnN	-	-	-	-	1.10	-	-	-
other	lnN	-	-	-	-	-	-	-	1.1

A 11% lumi uncertainty, a
 The infamous 23% tau id u
 4% uncertainty on lumi*Z
 4% uncertainty on effici
 20% uncertainty on QCD in
 10% uncertainty on QCD in
 10% uncertainty on non-Z

- Additional channels can be added easily (different “bin” label, same process labels)
 - Defining conventions for process/nuisance labels will make your life easier
- Can be done by hand or using a tool: `combineCards.py ch1.txt ch2.txt ch3.txt > comb.txt`

```
cat ../../data/tutorials/realistic-multi-channel.txt
```

Shape Experiment: Binned

```
imax 1
jmax 1
kmax *
```

```
shapes * * input-shapes-TH1.root $PROCESS $PROCESS_$SYSTEMATIC
```

```
bin 1
observation 85
```

shapes process channel file histogram [histogram_with_systematics]

bin	1	1
process	signal	background
process	0	1
rate	10	100

```
lumi      lnN      1.10      1.0
bgnorm    lnN      1.00      1.3
alpha     shapeN2   -         1   uncertainty on background shape and normalization
sigma     shapeN2   0.5       -   uncertainty on signal resolution. Assume the histogram is a 2 sigma shift,
#                                                so divide the unit gaussian by 2 before doing the interpolation
```

- In a shape experiment, a line pointing to a .root file is added
- For a binned experiment this .root file contains the shapes (histograms, PDFs, etc.) for the nominal distribution, systematic variations, and observed data
- Normalization of histograms can be used to simultaneously vary shape and rate

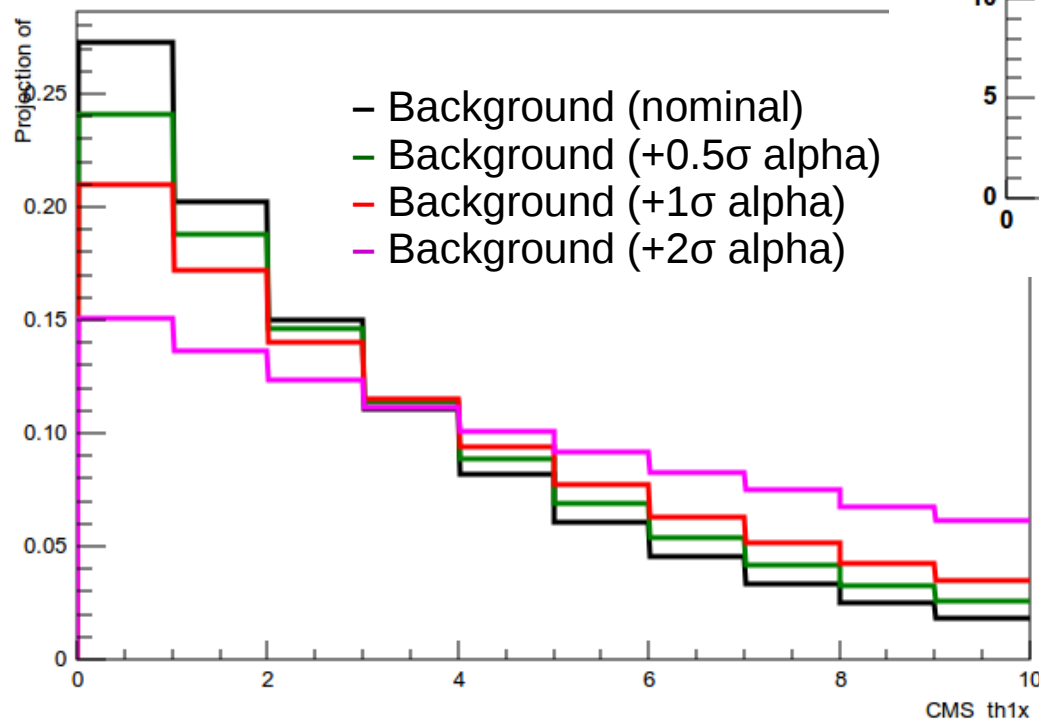
```
root [1] _file0->ls()
TFile** data/benchmarks/shapes/simple-shapes-TH1.root
TFile*  data/benchmarks/shapes/simple-shapes-TH1.root
KEY: TH1F signal;1 Histogram of signal__x
KEY: TH1F signal_sigmaUp;1 Histogram of signal__x
KEY: TH1F signal_sigmaDown;1 Histogram of signal__x
KEY: TH1F background;1 Histogram of background__x
KEY: TH1F background_alphaUp;1 Histogram of background__x
KEY: TH1F background_alphaDown;1 Histogram of background__x
KEY: TH1F data_obs;1 Histogram of data_obs__x
KEY: TH1F data_sig;1 Histogram of data_sig__x
```

```
cat simple-shapes-TH1.txt
root -l input-shapes-TH1.root
root [1] _file0->ls()
```

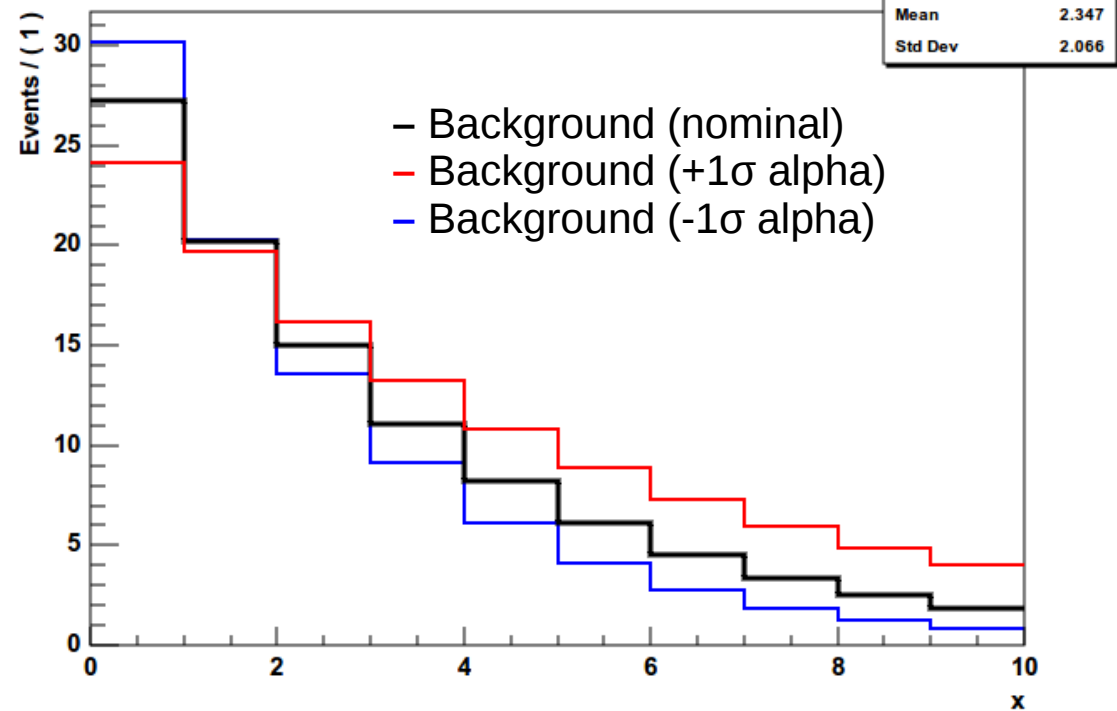
Shape Experiment: Morphing

- Given the histograms corresponding to ± 1 sigma for a particular nuisance parameter as input...

A RooPlot of "CMS_th1x"



Histogram of background__x



- ...Combine creates a morphing pdf which provides a shape for any value of the nuisance parameter

(SWAN only)

Shape Experiment: Parametric

Combination of `./couplings/hgg/hgg_8TeV_MVA.txt`

`imax 4` number of bins
`jmax 5` number of processes minus 1
`kmax *` number of nuisance parameters

```

shapes WH          cat0      hgg.inputsig_8TeV_MVA.root wsig_8TeV:hggpdfrel_wh_cat0
shapes ZH          cat0      hgg.inputsig_8TeV_MVA.root wsig_8TeV:hggpdfrel_zh_cat0
shapes bkg_mass    cat0      hgg.inputbkgdata_8TeV_MVA.root cms_hgg_workspace:pdf_data_pol_model_8TeV_cat0
shapes data_obs    cat0      hgg.inputbkgdata_8TeV_MVA.root cms_hgg_workspace:roohist_data_mass_cat0
shapes ggH         cat0      hgg.inputsig_8TeV_MVA.root wsig_8TeV:hggpdfrel_ggH_cat0
shapes qqH         cat0      hgg.inputsig_8TeV_MVA.root wsig_8TeV:hggpdfrel_vbf_cat0
shapes ttH         cat0      hgg.inputsig_8TeV_MVA.root wsig_8TeV:hggpdfrel_ttH_cat0
[... similarly for cat1, cat4, cat5 ... ]

```

bin	cat0	cat1	cat4	cat5
observation	-1.0	-1.0	-1.0	-1.0

bin	cat0	cat0	cat0	cat0	cat0	cat0	[... cat1, cat4, cat5 ...]
process	ZH	qqH	WH	ttH	ggH	bkg_mass	[... cat1, cat4, cat5 ...]
process	-4	-3	-2	-1	0	1	[... cat1, cat4, cat5 ...]
rate	6867.0000	19620.0000	12753.0000	19620.0000	19620.0000	1.0000	[... cat1, cat4, cat5 ...]

	lnN							[... cat1, cat4, cat5 ...]
CMS_eff_j	0.999125	0.964688	0.999125	0.998262	0.996483	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_JECmigration	-	-	-	-	-	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_UEPSmigration	-	-	-	-	-	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_eff_MET	-	-	-	-	-	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_eff_e	-	-	-	-	-	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_eff_m	-	-	-	-	-	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_eff_trig	1.01	1.01	1.01	1.01	1.01	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_n_id	1.034/0.958	1.039/0.949	1.034/0.958	1.053/0.915	1.035/0.958	-	-	[... cat1, cat4, cat5 ...]
CMS_hgg_n_pdf_1	-	0.998/0.996	-	-	1.002/0.998	-	-	[... cat1, cat4, cat5 ...]

[... more lnN nuisances ...]

```

CMS_hgg_nuissancedeltamcat4 param 0.0 0.001458
CMS_hgg_nuissancedeltafracright_8TeV param 1.0 0.002000
CMS_hgg_nuissancedeltamcat1 param 0.0 0.001470
CMS_hgg_nuissancedeltamcat0 param 0.0 0.001530
CMS_hgg_nuissancedeltasmearcat4 param 0.0 0.001122
CMS_hgg_nuissancedeltasmearcat1 param 0.0 0.001167
CMS_hgg_nuissancedeltasmearcat0 param 0.0 0.001230
CMS_hgg_globalscale param 0.0 0.004717

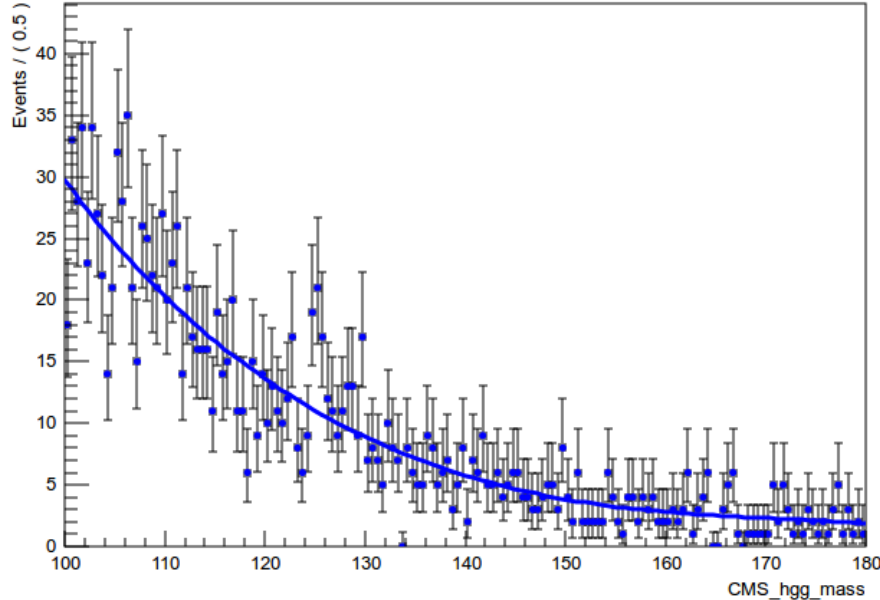
```

- In a parametric shape experiment, the “shapes” line points to a `.root` file which has a `RooWorkspace` which contains `RooAbsPdf`'s that describe the shape of each process

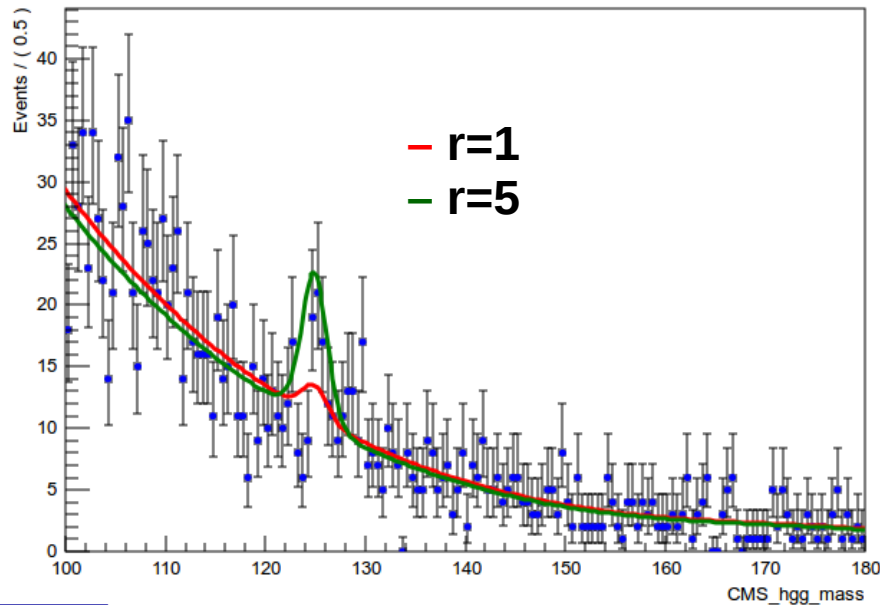
cat hgg_8TeV_MVA_cat0145.txt

Shape Experiment: Parametric

A RooPlot of "CMS_hgg_mass"

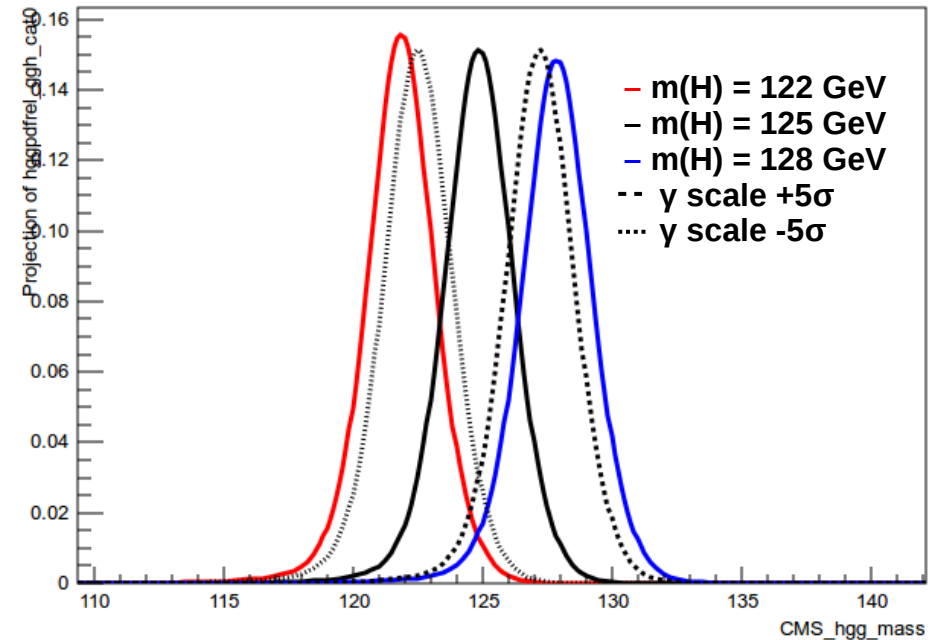


A RooPlot of "CMS_hgg_mass"



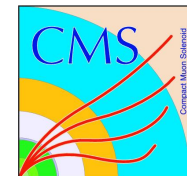
(SWAN only)

A RooPlot of "CMS_hgg_mass"

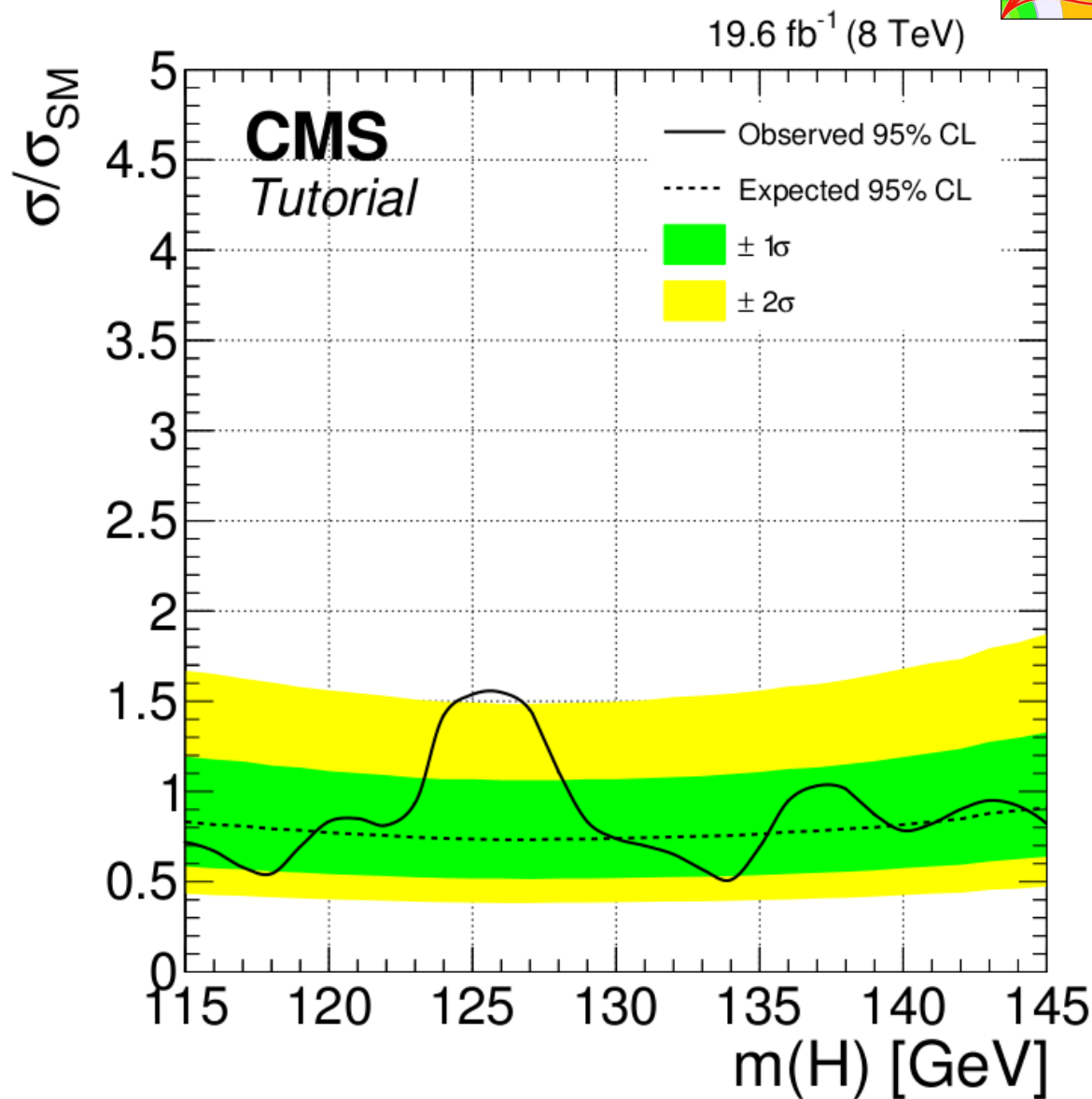


- Parametric experiments can be binned or unbinned (binned faster for larger datasets)
- How nuisance parameters affect the shape for a process can be defined in the workspace
- After converting to a workspace, one can draw the total signal+background PDF for any set of parameter values

Limit Setting



- Combine supports several methods for computation of limits on the model POIs
- Asymptotic CL_s limits: most common method used for setting limits in HIG PAG
- Uses the asymptotic approximation of the test-statistic distribution to quickly compute both observed and expected limits
- Other methods available:
 - Bayesian
 - Hybrid Bayesian-Frequentist
 - Fully Frequentist
 - 1 sided Feldman-Cousins Interval
- More CPU intensive

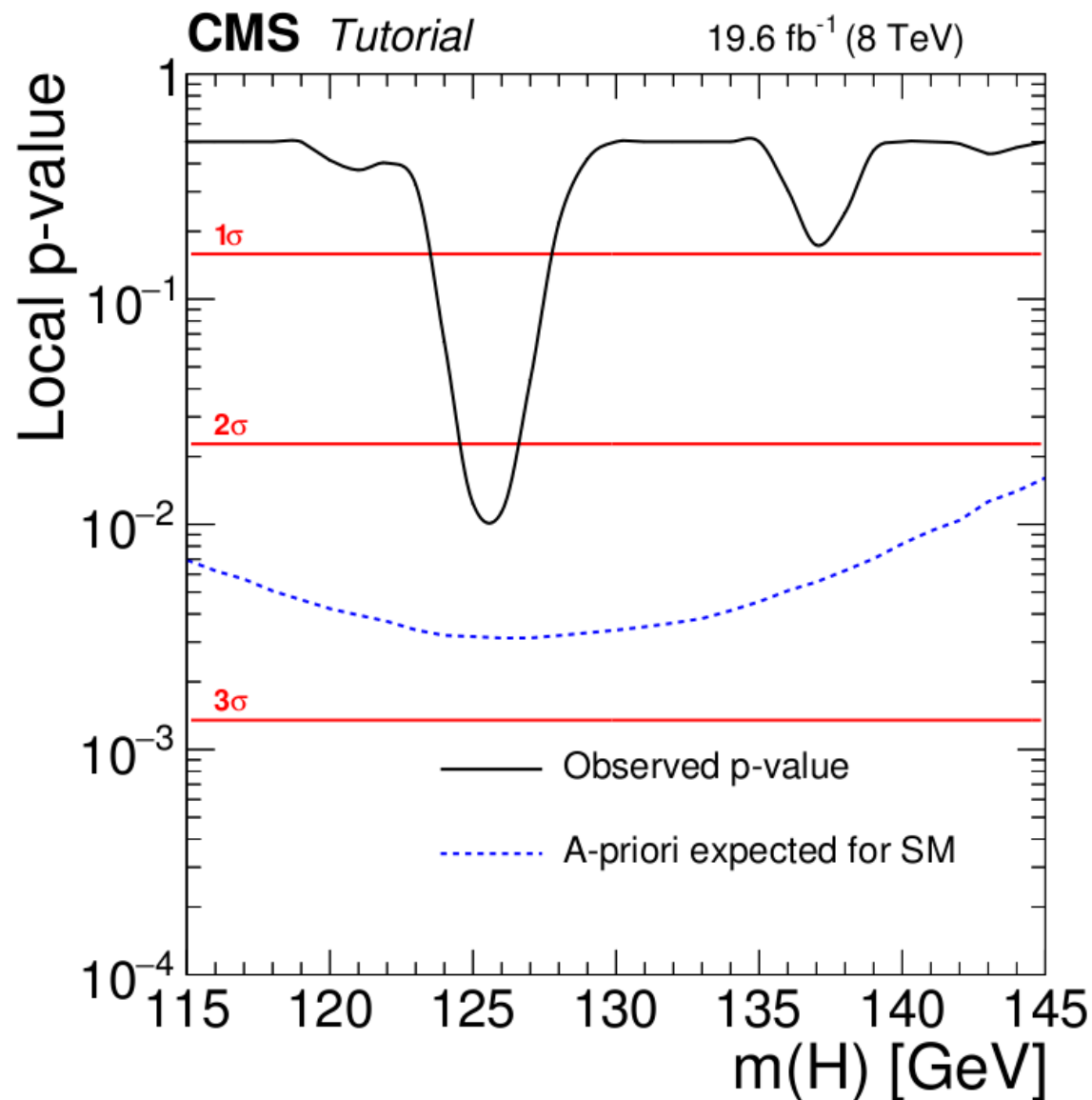


```
chmod u+x run_hgg_asymptotic.sh  
./run_hgg_asymptotic.sh  
python plotLimits.py
```

Extracting Significance



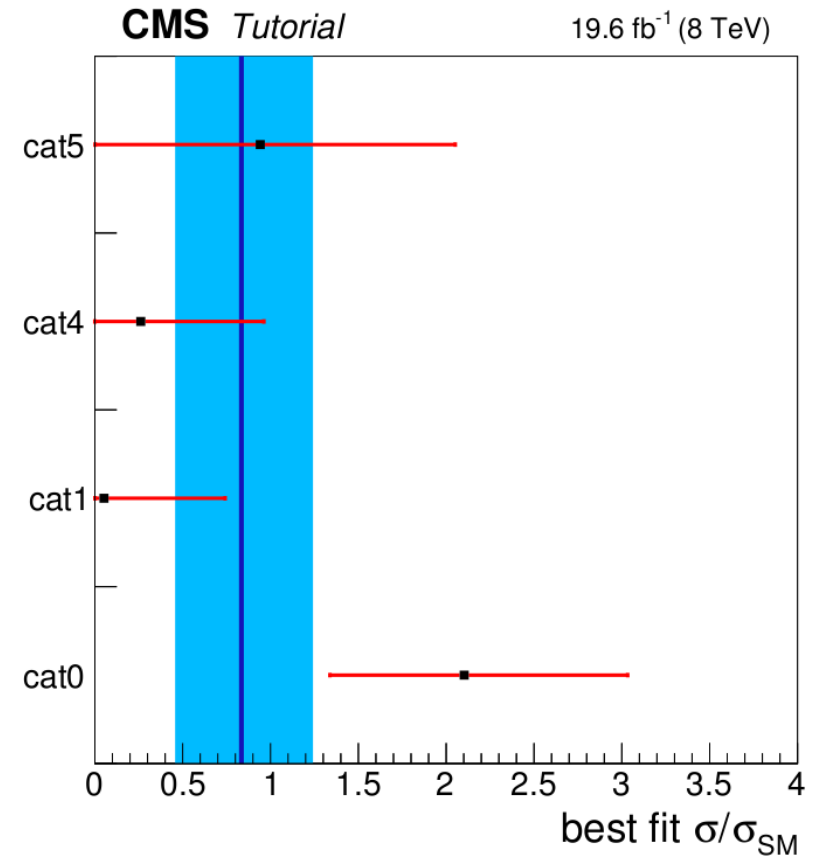
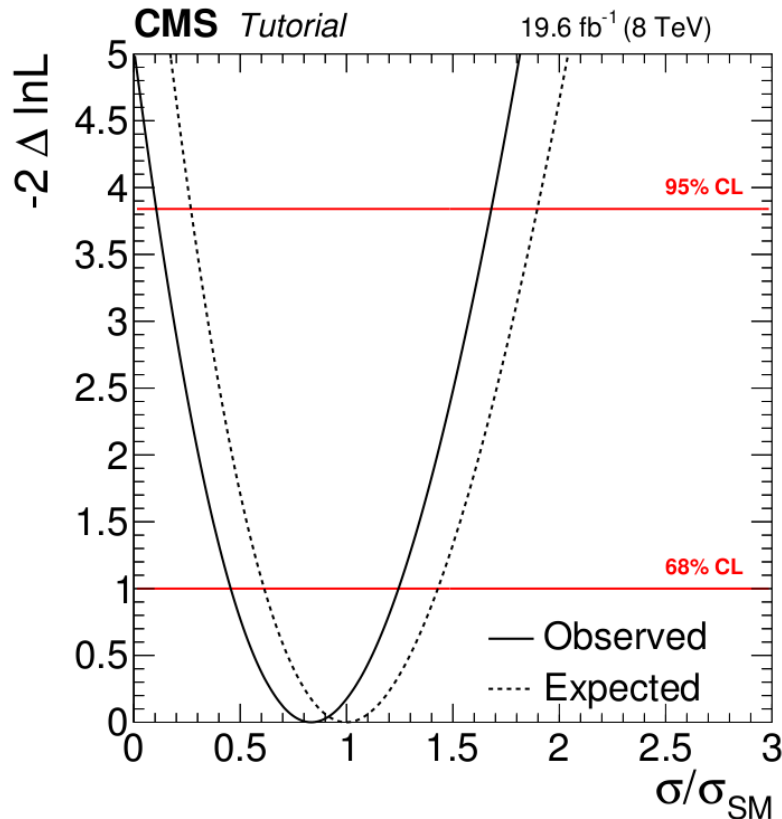
- Combine supports computation of the expected/observed significance of an excess
- Most common in HIG is to again use the profile likelihood approximation
 - For expected significance, can use the “Asimov” dataset or toys of signal+background
- Expected/Observed significance also supported for modified frequentist and fully frequentist methods



```
chmod u+x run_hgg_pvalue.sh  
./run_hgg_pvalue.sh  
python plotPvalue.py
```

Maximum Likelihood Fit

- Combine provides a Maximum Likelihood Fit method for extracting the (expected) best-fit and uncertainties for the POIs
- Can fit channels simultaneously with full correlation of nuisance parameters
- Can produce likelihood scans for the POI



```
combine -n Obs -M MultiDimFit -m 125 hgg_8TeV_MVA_cat0145.root --algo=grid --points 300 --setPhysicsModelParameterRanges r=0.0,3.0
combine -n Exp -M MultiDimFit -m 125 hgg_8TeV_MVA_cat0145.root -t -1 --expectSignal=1 --algo=grid --points 300 \
--setPhysicsModelParameterRanges r=0.0,3.0
python plotMuScan.py
combine -m 125 -M ChannelCompatibilityCheck hgg_8TeV_MVA_cat0145.root --saveFitResult
python cccPlot.py
```