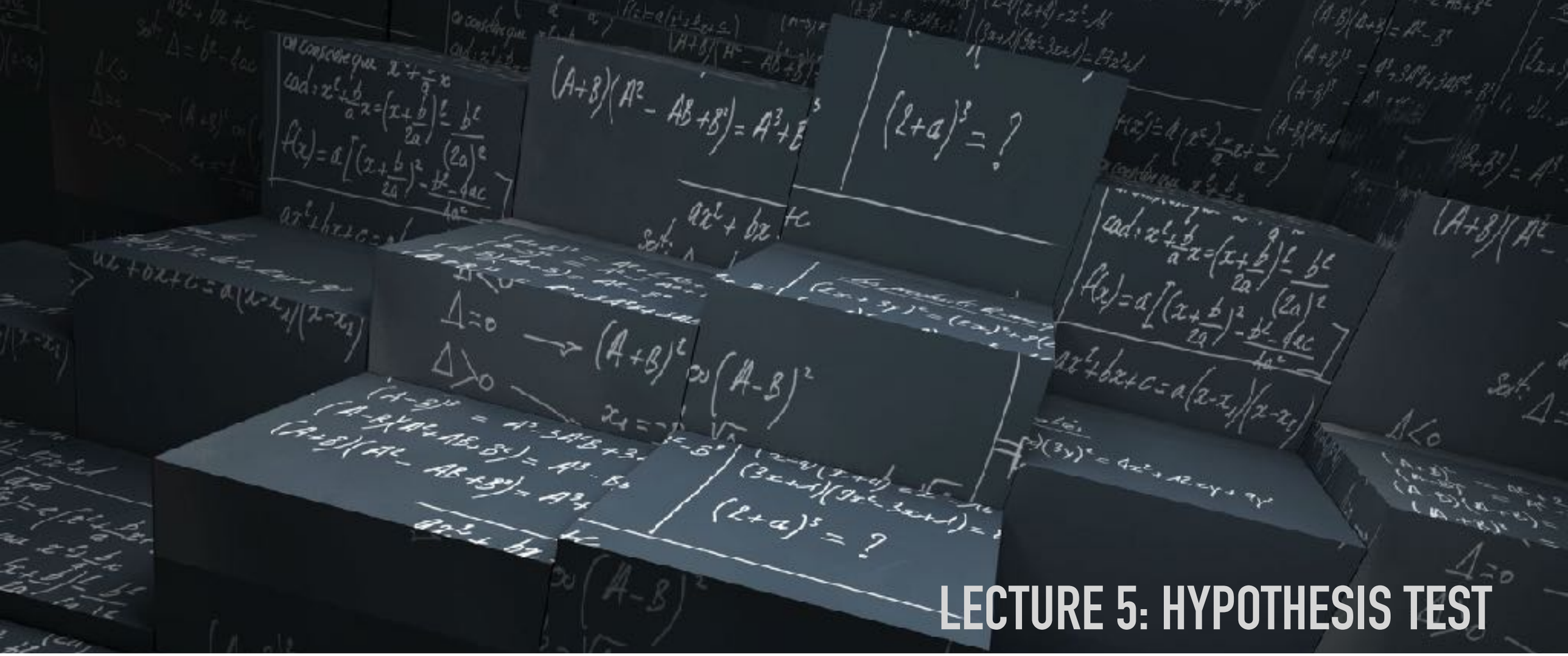


LECTURE 5: HYPOTHESIS TEST

STATISTICAL ANALYSIS IN EXPERIMENTAL PARTICLE PHYSICS

Kai-Feng Chen
National Taiwan University

HYPOTHESIS TEST: OVERVIEW

- The key idea: **compare two hypotheses to see which one better describes the observed data**; or what is the best way to separate the events into two classes, which originating from each of the two hypotheses. One shall distinguish the two cases, for a given parameter of interests:
  - If nothing is known a priori, the method of parameter estimation introduced in the earlier lecture should be adopted;
  - If a (theoretical) prediction exists, it is more appropriate to formulae the problem as a test.
- The two hypotheses in discussion are classically labeled as:
  - **H_0 : the null hypothesis** (*e.g. a sample contains only background*)
  - **H_1 : the alternative hypothesis** (*e.g. a sample contains both background and signal*)

HYPOTHESES: SIMPLE VERSUS COMPOSITE

Simple Hypothesis

- When the hypotheses $\mathbf{H_0}$ and $\mathbf{H_1}$ are completely set/specified (ie. with no free parameters), they are the **simple hypotheses**.
- Hypothesis testing for simple hypotheses has been well understood, the method works for both large and small samples.

Composite Hypothesis

- If the hypothesis still contains one or more free parameters, then it becomes a composite hypothesis. Generally there is only an asymptotic theory for the testing.
  - You can image that the composite hypotheses are more common in a real problems. One can obtain the exact answers for small samples using Monte Carlo methods.

MORE TERMINOLOGY...

Test Statistic

- A variable computed from our sample, and provides the discriminating power between the two hypotheses $\mathbf{H_0}$ and $\mathbf{H_1}$, as a "summary" of the information available in the sample.

level of significance (α)

- The probability to reject $\mathbf{H_1}$ if $\mathbf{H_0}$ is assumed to be true.
- **Error of the first kind.**
- **α = 1 – misidentification probability**

misidentification probability (β)

- The probability to reject $\mathbf{H_0}$ if $\mathbf{H_1}$ is assumed to be true.
- **Error of the second kind.**
- **1 – β = power of the test = selection efficiency**

MORE TERMINOLOGY . . . (CONT.)

- If W is the space of all possible data, one can find a **Critical Region** $w \in W$ (in which we reject H_0) which gives the measure:
  - $P(\text{data } X \in w | H_0) = \alpha$, chosen to be small;
  - $P(\text{data } X \in (W-w) | H_1) = \beta$, is made as small as possible.

		H_0 TRUE	H_1 TRUE
$X \notin w$ (not in critical region)	Accept H_0	Good: Acceptance of H_0 Prob = $1-\alpha$	Contamination Error of the 2nd kind Prob = β
$X \in w$ (in critical region)	Reject H_0	Loss/inefficient: Error of the 1st kind Prob = α	Good: Rejection of H_0 Prob = $1-\beta$

USEFULNESS OF A TEST

- The "usefulness of a test" is the ability to discriminate against the alternative hypothesis $\mathbf{H_1}$. The measure of this factor is the **power of the test**, defined as the probability $1-\beta$, where the X falling into the critical region if $\mathbf{H_1}$ is true:

$$P(X \in w|H_1) = 1 - \beta$$

- While β is the probability that X will fall in the acceptance region for $\mathbf{H_0}$ if $\mathbf{H_1}$ is true:

$$P(X \in W - w|H_1) = \beta$$

- Determination of a multidimensional critical region may be difficult in practice, instead, a single test statistic $t(X)$ variable can be introduced. In this case the critical region is redefined along t instead of X.

EVENTS SEPARATION IN 2 CLASSES

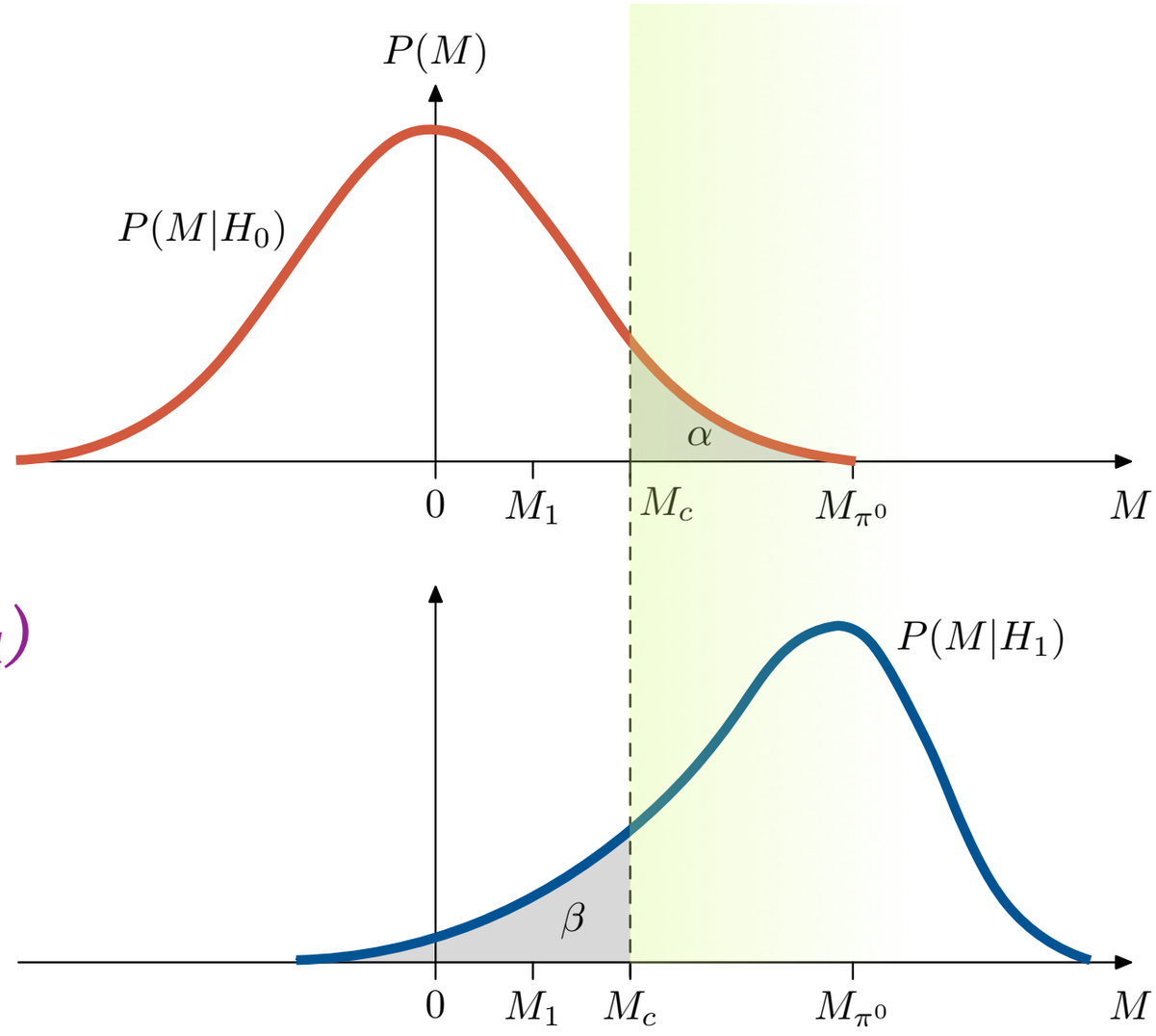
- Distinguish elastic proton scattering events

  pp→pp *(hypothesis under test, H_0)*

  from inelastic scattering events

  pp→ppπ⁰ *(alternative hypothesis, H_1)*

  in an experimental setup which measures the proton momentum, but the π⁰ cannot be detected. If one choose the **missing mass** distribution as the test statistic:



Missing mass M under the hypotheses H_0 and H_1, with the critical region $M > M_c$.

THE NEYMAN–PEARSON TEST

- For a hypothesis test for $\mathbf{H_0}$ against $\mathbf{H_1}$ with a given significance α, the most powerful test is reached with the **best critical region w** with the smallest value of β.
- Suppose a random variable $X = (X_1, X_2, \ldots, X_N)$ has PDF $f(X|\mathrm{H}_0)$ and $f(X|\mathrm{H}_1)$. Based on the definitions of α and the test power $1-\beta$, one can express

$$\alpha = \int_w f(X|H_0)dX \qquad 1-\beta = \int_w f(X|H_1)dX$$

$$1-\beta = \int_w \frac{f(X|H_1)}{f(X|H_0)} f(X|H_0)dX = E_w\left[\left.\frac{f(X|H_1)}{f(X|H_0)}\right|_{H_0}\right]$$

expected value with H_0 in space w

this will be maximal if and only if w (with given fraction α) contains the largest values of $f(X|H_1)/f(X|H_0)$

THE NEYMAN–PEARSON TEST (CONT.)

- Thus the **best critical region *w*** should be consistent with the points satisfying the following condition:

$$R(X, H_0, H_1) \equiv \frac{f(X|H_1)}{f(X|H_0)} \geq C_\alpha$$

- The procedure gives the following clear criteria:
  - if $R(X, \mathbf{H_0}, \mathbf{H_1}) > C_\alpha$, choose $\mathbf{H_1}$
  - if $R(X, \mathbf{H_0}, \mathbf{H_1}) \leq C_\alpha$, choose $\mathbf{H_0}$
- This is exactly the **Neyman–Pearson test**. The test statistic R is just the **ratio of the likelihoods for the two hypotheses**, and this ratio must be "calculable" at all points of the observable space.
- Thus the two hypotheses $\mathbf{H_0}$ and $\mathbf{H_1}$ must be completely specified as **simple hypotheses**, and then this method will give the **best test**.

LIKELIHOOD RATIO TEST

- The likelihood ratio test is an extension of the Neyman-Pearson test to the case of composite hypotheses, although its properties are only known asymptotically.
- For the observations $\boldsymbol{X}$ with PDF $f(\boldsymbol{X}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ contains the full set of parameters $\boldsymbol{\theta}=(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2,\ldots)$. The likelihood function can be expressed as

$$L(X|\theta) = \prod_{i=1}^{N} f(X_i|\theta)$$

- Now the null as well as the alternative hypotheses can be defined by the total space $\boldsymbol{\Theta}$ as well as its subspace $\boldsymbol{\Omega}$ for any test of parametric hypotheses:

$$\mathbf{H_0} : \boldsymbol{\theta} \in \boldsymbol{\Omega}$$
$$\mathbf{H_1} : \boldsymbol{\theta} \in \boldsymbol{\Theta} - \boldsymbol{\Omega}$$

LIKELIHOOD RATIO TEST (CONT.)

- Now one can define the **maximum likelihood ratio** a test statistic for $\mathbf{H_0}$:

$$\lambda = \frac{\max_{\theta \in \Omega} L(X|\theta)}{\max_{\theta \in \Theta} L(X|\theta)}$$

← maximizing in subspace Ω

← maximizing in all space

- If $\mathbf{H_0}$ and $\mathbf{H_1}$ are simple hypotheses, λ would reduce to the Neyman-Pearson test statistic as given earlier.
- For composite hypotheses, λ is always a function of the sufficient statistic for the problem with a large set of observations (*asymptotic behavior*).
- Furthermore, if $\mathbf{H_0}$ imposes k constraints on the total parameters, then the value of $-2\ln\lambda$ is just distributed as a **χ^2 distribution of k degree of freedom** under $\mathbf{H_0}$.
  - The confidence level α can be extracted from a table of χ^2!
  - But this is only true asymptotically, the only way to know how good the approximation is to do a Monte Carlo study.

LIKELIHOOD RATIO TEST EXAMPLE

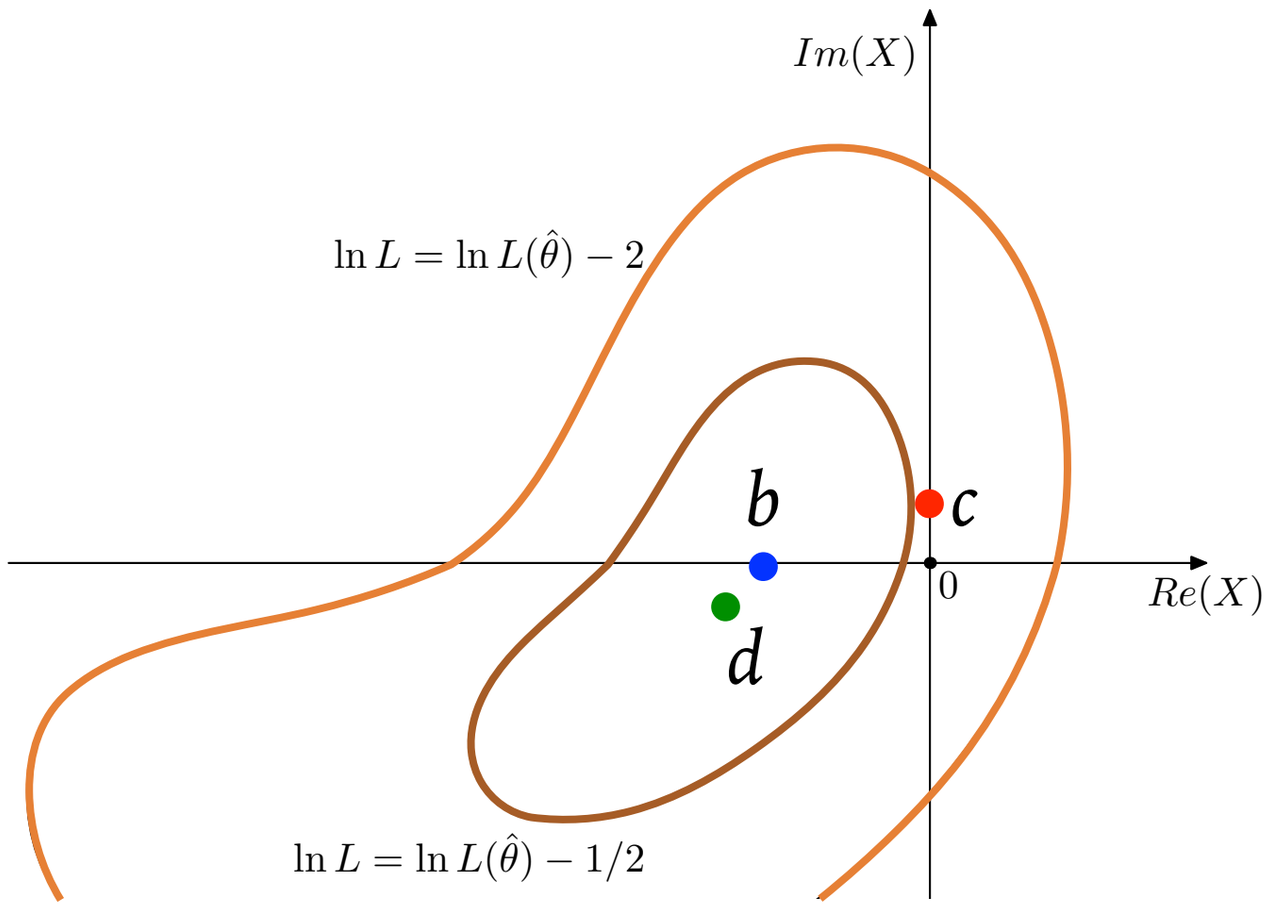
- Suppose you are performing a measurement of an amplitude X, where it is allowed to be any complex number. There are three existing different theories which predict the following for X:
  - Theory A predicts $X=0$;
  - Theory B predicts X is real: $\mathrm{Im}(X)=0$;
  - Theory C predicts X is purely imaginary and non-zero: $\mathrm{Re}(X)=0$, but $\mathrm{Im}(X)\neq 0$
- If the value of X is interesting only as it could distinguish between the hypotheses A, B, C or the general case. In this hypothesis testing:
  - **Hypothesis A is a simple hypothesis;**
  - **Hypothesis B is composite, however it includes hypothesis A as a special case;**
  - **Hypothesis C is also composite, and separates from A and B.**
  - **The alternative to all these hypotheses is that $\mathrm{Re}(X)$ and $\mathrm{Im}(X)$ are both non-zero.**

LIKELIHOOD RATIO TEST EXAMPLE (II)

- **As a test of theory A:**
  the maximum likelihood ratio for hypothesis A versus the general case is

$$\lambda_a = \frac{L(X=0)}{L(X=d)}$$

- If the hypothesis A is true, the $-2\ln\lambda_a$ is distributed asymptotically as a χ^2 **distribution of 2 degrees of freedom**, since both **Re**(X) and **Im**(X) are constrained to be zero in hypothesis A.



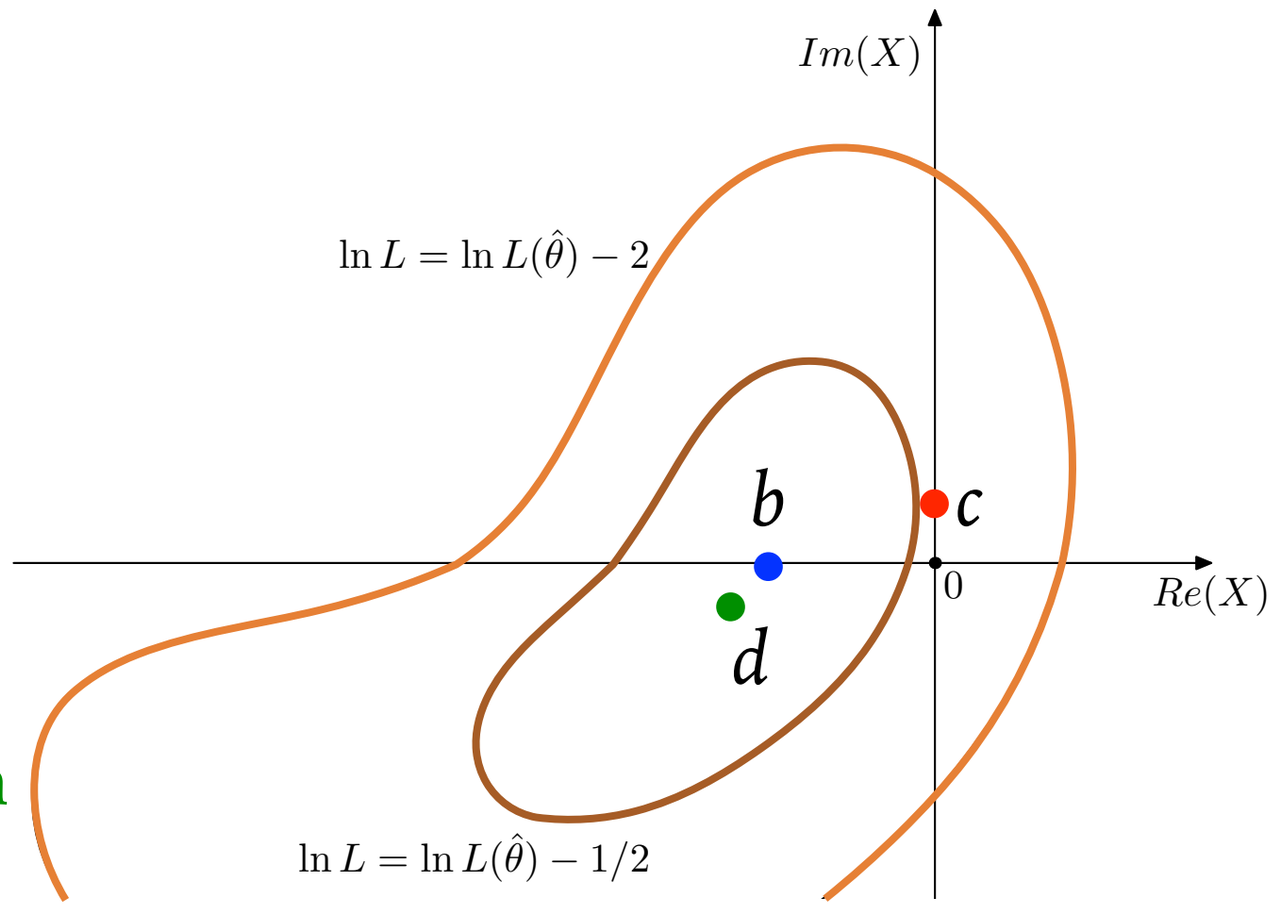
$X = d$ is the point where L is maximal;
$X = b$ is the maximum of L when $Im(X) = 0$;
$X = c$ is the maximum of L when $Re(X) = 0$.

LIKELIHOOD RATIO TEST EXAMPLE (III)

- **As a test of theory B:**
  the maximum likelihood ratio for hypothesis B versus the general case is

$$\lambda_b = \frac{L(X = b)}{L(X = d)}$$

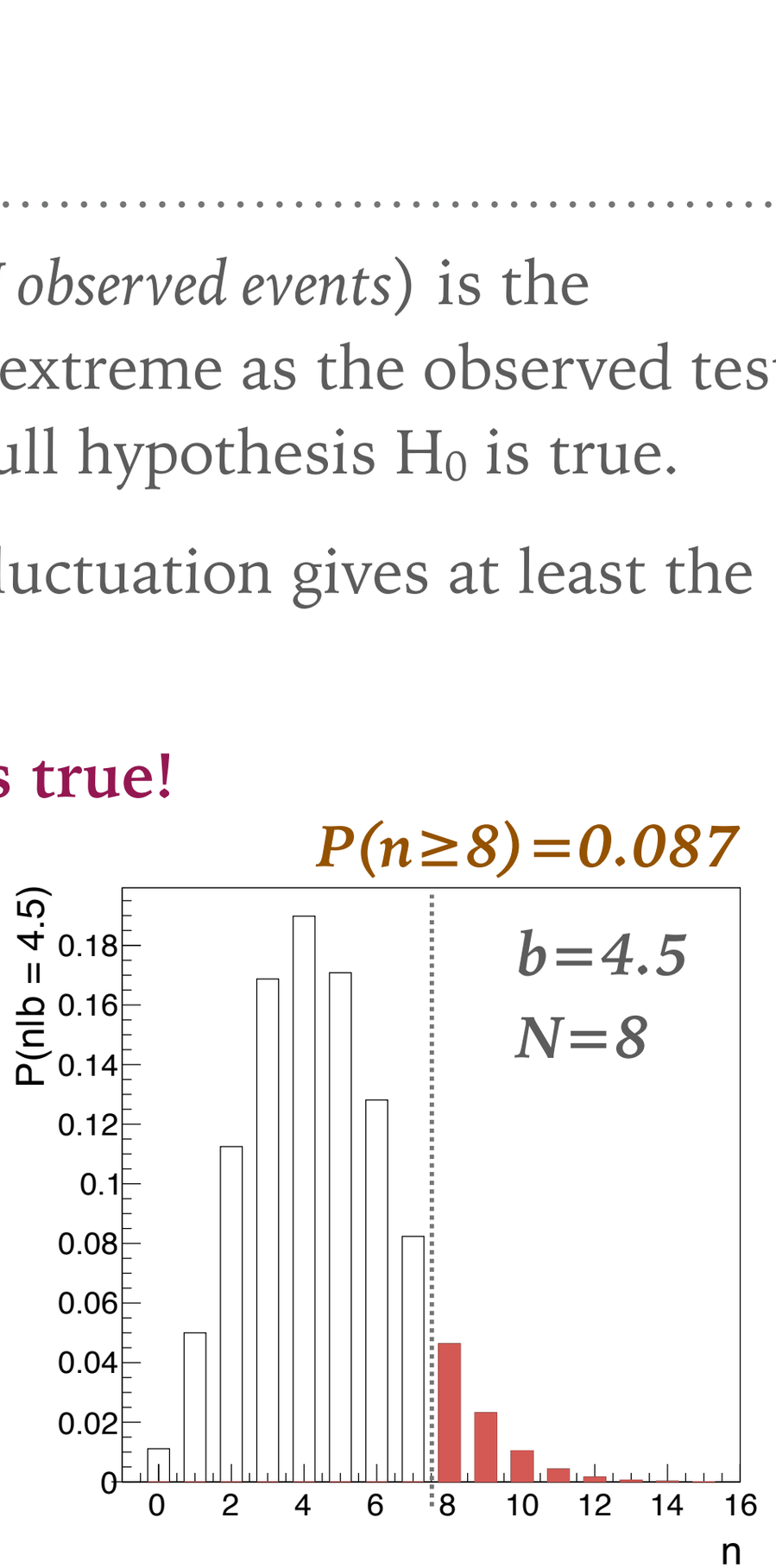
- If the hypothesis B is true, the $-2\ln\lambda_b$ is distributed asymptotically as a **χ^2 distribution of 1 degree of freedom**, since only **Im(*X*)** are constrained to be zero in hypothesis B.
- **As a test of theory C:**
  just replace ***b*** by ***c*** as above!



***X = d** is the point where **L** is maximal;*
***X = b** is the maximum of **L** when **Im(X)=0**;*
***X = c** is the maximum of **L** when **Re(X)=0**.*

BENCHMARK OF DISCOVERY: P-VALUE

- From frequentist definition, p-value (*of **N** observed events*) is the probability to observe a result at least as extreme as the observed test statistic (e.g. at least $\geq N$ events) if the null hypothesis H_0 is true.
  - Probability that a background (over-)fluctuation gives at least the observed number of events.
  - **NOT the probability for which H_0 is true!**
- If H_0 is true, the distribution of the p-value itself is uniform if the distribution is continuous, or approximately uniform in case of discrete distributions.



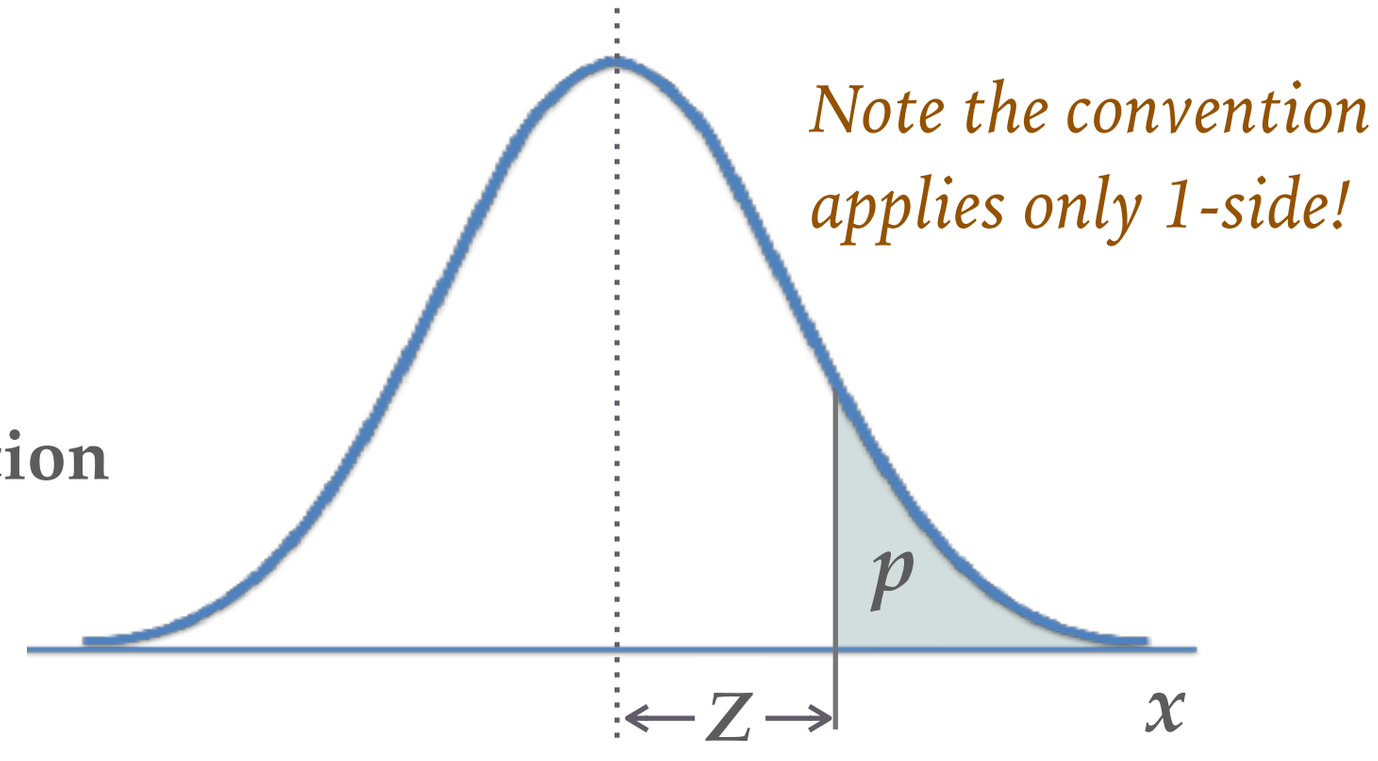
P-VALUE AND SIGNIFICANCE

- The p-value measures the observed incompatibility with the background-only hypothesis.
- However it is often converted into # of standard deviations corresponding to a Gaussian distribution ⇒ **"nσ" significance**.

$$p = \int_Z^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{Z}{\sqrt{2}}\right)\right]$$

$$Z = \Phi^{-1}(1-p)$$

Φ(*x*): cumulative distribution of a Gaussian



THE FIVE SIGMA RULE

- In your search for a signal, for example, a bump on top of a smooth background, once you observed an effect of
  - **$>3\sigma$, or p-value<0.00135**, a **"hint"** of the proposed signal can be claimed.
  - **$>4\sigma$, or p-value$<3.17\times10^{-5}$**, an **"evidence"** can be claimed.
  - **$>5\sigma$, or p-value$<2.87\times10^{-7}$**, it's time for **"discovery/observation"**!
- This is a more-or-less standard criterion commonly adopted in modern HEP experiments.
  - Quote from Rosenfeld's early review of PDG: "*…we are now generating at least 100,000 potential bumps per year, and should expect several 4-sigma and hundreds of 3-sigma fluctuations. What are the implications? To the theoretician or phenomenologist the moral is simple; wait for nearly 5-sigma effects.*"

APPROXIMATE FORMULAE

- Consider a Poisson counting experiment, consists of an observed number n of events, modeled as a Poisson distribution with a mean of $\eta s+b$, where s and b are the expected numbers of signal and background.
- In the large mean limit, the Poisson variable n can be approximated as a Gaussian with mean $\eta s+b$ and variance $\sigma^2=\eta s+b$.
- The p-value (*and the corresponding significance*) from $\eta=0$ hypothesis is the probability to find n greater than or equal to the value observed,

$$p = \Phi\left(\frac{n-b}{\sqrt{b}}\right) \Rightarrow Z = \Phi^{-1}(1-p) = \frac{n-b}{\sqrt{b}}$$

- The median of n assuming $\eta=1$ is $s+b$, and therefore the median discovery significance is

$$\mathrm{med}[Z|\eta=1] = \frac{s}{\sqrt{b}}$$

The widely used "figure of merit"!

APPROXIMATE FORMULAE (II)

- A better approximation for the Poisson counting experiment, can be obtained by testing $\eta=0$ using the **likelihood ratio test**:

$$\lambda = \frac{L(\eta = 0)}{L(\eta = \hat{\eta})}, \quad \text{where } L(\eta) = \frac{(\eta s + b)}{n!} \exp[-(\eta s + b)]$$

- The best value $\hat{\eta}$ is given by $(n-b)/s$ after maximizing the likelihood function. Here both s and b are assumed to be known and no nuisance parameters. The relevant signal models correspond to positive η, one may test the $\eta=0$ hypothesis using the test statistic $Q=-2\ln\lambda$. In the large-sample limit, the discovery significance can be expressed as

$$Z = \sqrt{Q} = \sqrt{2\left(n \ln \frac{n}{b} + b - n\right)}$$

- Again the median of n assuming $\eta=1$ is $s+b$, and therefore the median discovery significance is

$$\text{med}[Z|\eta = 1] = \sqrt{2\left[(s+b)\ln\left(1+\frac{s}{b}\right) - s\right]}$$

Another widely used "figure of merit"!

APPROXIMATE FORMULAE (III)

- In some analyses, the goal may not be to establish discovery of a signal process but rather to measure the signal rate as accurately as possible.
- If we consider again the Poisson counting experiment described by the likelihood function given in the previous slide, the best value $\hat{\eta} = (\mathbf{n}-\mathbf{b})/\mathbf{s}$ has a variance (*assuming* $\eta = 1$):

$$V(\hat{\eta}) = V\left(\frac{n-b}{s}\right) = \frac{1}{s^2}V(n) = \frac{s+b}{s^2}$$

- Take the inverted standard deviation of $\hat{\eta}$:

$$\frac{1}{\sigma(\hat{\eta})} = \frac{s}{\sqrt{s+b}}$$

  One may therefore us this as a figure of merit to be maximized in order to obtain the best measurement accuracy of a rate parameter.
- Now you know why there are 3 commonly used definitions of figure of merit!

DISCOVER OR EXCLUDE A SIGNAL HYPOTHESIS

- Assuming a given value of signal strength $\eta > 1$, a corresponding p-value can be computed.
- In this case the p-value also measures the probability of a signal (under)fluctuation for $n \leq n_{obs}$.
- The exclusion of a signal hypothesis usually has relaxed requirements comparing to discovery of signal:
  - **$p < 0.05$ (95% confidence level) $\Leftrightarrow Z = 1.64$**
  - **$p < 0.10$ (90% confidence level) $\Leftrightarrow Z = 1.28$**

Discover a new signal need a more stringent significance than excluding it!

RECALL: PROBLEMS IN UPPER LIMIT ESTIMATION

- Remember we have commented some of the issues of using Feldman-Cousins unified approach before the end of last lecture:
  - In some cases, a statistical (*under*-)fluctuation of the background may lead to the exclusion of zero signal.
  - In some other cases, when adding the channels with low signal sensitivity, may produce upper limits that are worse than without adding them.
- So let's introduce the modified frequentist method: **the CLs method** which solves the problems mentioned above in the following slides!

THE MODIFIED FREQUENTIST METHOD: CL_S (LEP VER.)

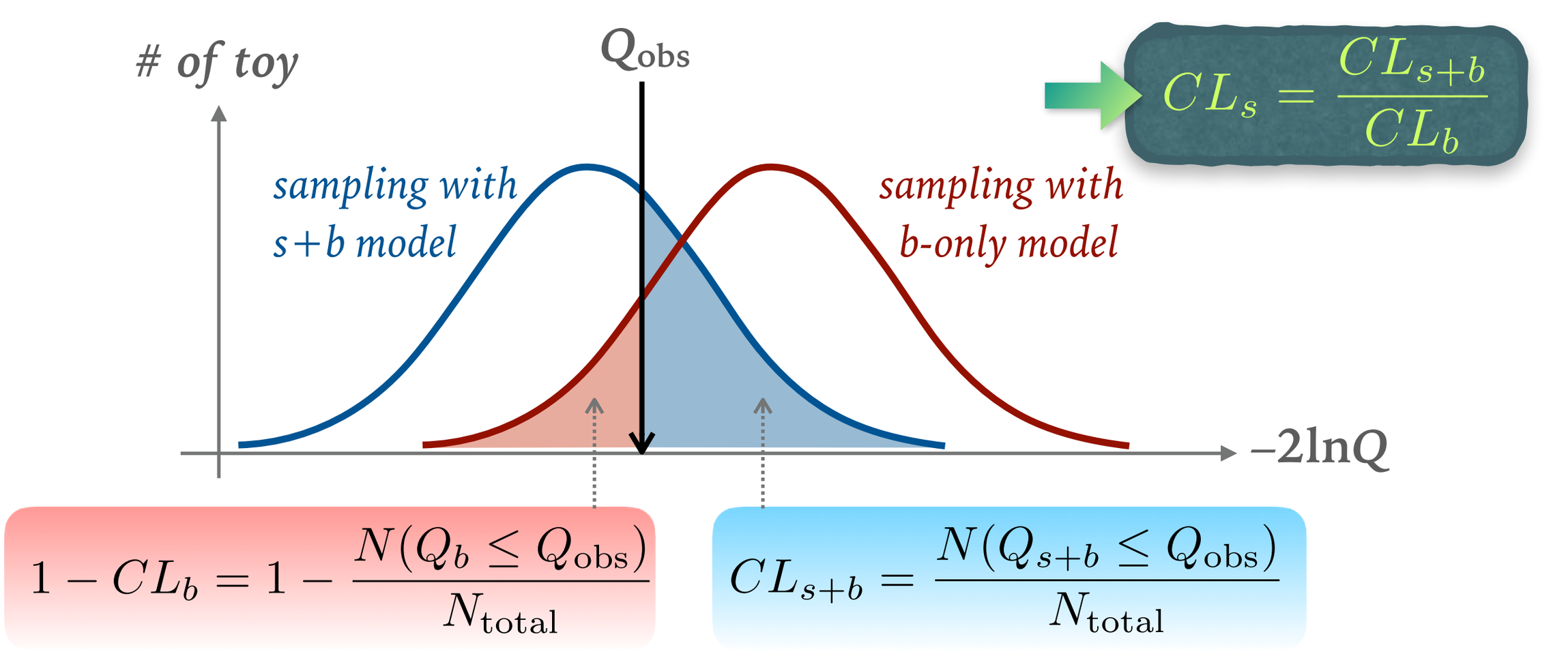
- Method was first developed for Higgs limit estimation at LEP-II, since there were multiple analysis channels from multiple experiments.
- Using the **likelihood ratio** as the test statistic $Q = L(s+b)/L(b)$.
- The confidence levels estimator CL_S is different from the Feldman-Cousins approach:

$$CL_s = \frac{CL_{s+b}}{CL_b} = \frac{P(Q_{s+b} \leq Q_{\rm obs})}{P(Q_b \leq Q_{\rm obs})} = \frac{N(Q_{s+b} \leq Q_{\rm obs})}{N(Q_b \leq Q_{\rm obs})}$$

- It's more conservative in general –– gives over-coverage comparing to the classical limit based on CL_{s+b}
- Limit for counting with Poissonian data is identical to Bayesian method.
- No problem when adding channels with low discrimination power.

CL_S METHOD IN PRACTICE

- In practice, the actual estimators CL_{s+b}, CL_b, and the corresponding CL_s can be computed using toy Monte Carlo.
- Sampling test statistic distributions for both **null hypothesis (b-only)** and **alternative hypothesis (s+b)** configurations.



$$CL_s = \frac{CL_{s+b}}{CL_b}$$

$$1 - CL_b = 1 - \frac{N(Q_b \leq Q_{\text{obs}})}{N_{\text{total}}}$$

$$CL_{s+b} = \frac{N(Q_{s+b} \leq Q_{\text{obs}})}{N_{\text{total}}}$$

EXAMPLE: POISSON COUNTING WITH KNOWN BACKGROUND

- Suppose there is a **Poisson process** with a known **background of 4.5 events**, and the experiment observed **8 events**.

```
double CL_scan(double target_ns = 5., int ntoys = 1000, bool display = false)
{
    const int nobs = 8; const double nb = 4.5;

    double L0 = TMath::Poisson(nobs,nb);
    double L1 = TMath::Poisson(nobs,nb+target_ns);
    double Q_obs = -2.*(log(L1)-log(L0)); // observed test statistic

    double CL[2] = {0.,0.};
    vector<double> stat[2];
    for (int hypo=0; hypo<=1; hypo++) {
        for (int idx=0; idx<ntoys; idx++) {
            int n = 0;
            if (hypo==0) n = rnd.Poisson(nb); // toy sampling b-only
            if (hypo==1) n = rnd.Poisson(nb+target_ns); // toy sampling s+b

            double L0 = TMath::Poisson(n,nb);
            double L1 = TMath::Poisson(n,nb+target_ns);
            double Q = -2.*(log(L1)-log(L0)); // sampled test statistic
            stat[hypo].push_back(Q);

            if (Q>=Q_obs) CL[hypo] += 1.;
        }
        CL[hypo] /= (double)ntoys;
    }
    double CLs = CL[1]/CL[0];
```

calculate the value of test statistic for $n_{obs} = 8$

calculate the test statistic distribution from toy

CL_{s+b} and CL_b

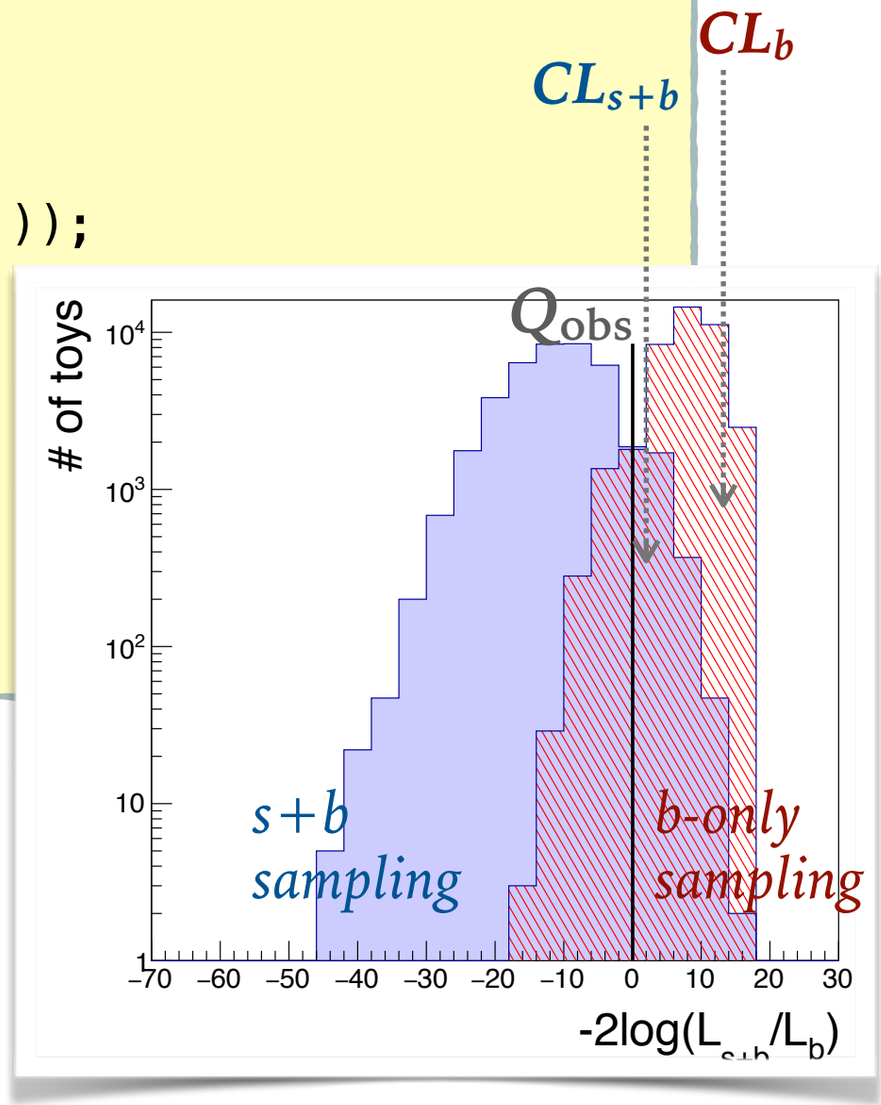
partial example_01.cc

EXAMPLE: POISSON COUNTING WITH KNOWN BACKGROUND (CONT.)

partial example_01.cc

```
if (display) {
      TH1D* h_stat0 = new TH1D("h_stat0","",25,-70.,30.);
      TH1D* h_stat1 = new TH1D("h_stat1","",25,-70.,30.);
      for (int idx=0; idx<ntoys; idx++) {
          h_stat0->Fill(stat[0][idx]);
          h_stat1->Fill(stat[1][idx]);
      }

      TCanvas *c1 = new TCanvas("c1","",600,600);
      c1->SetLogy();
      h_stat1->Draw();
      h_stat0->Draw("same");
      TLine l1;
      l1.DrawLine(Q_obs,0.,Q_obs,h_stat1->GetMaximum());
  }
  printf("Target ns=%.1f, CLs+b=%.3f, CLb=%.3f, CLs=%.3f\n",
      target_ns,CL[1],CL[0],CLs);

  return CLs;
}

void example_01()
{  CL_scan(7.5,40000,true); }
```

convert the stored test statistic distributions to histograms

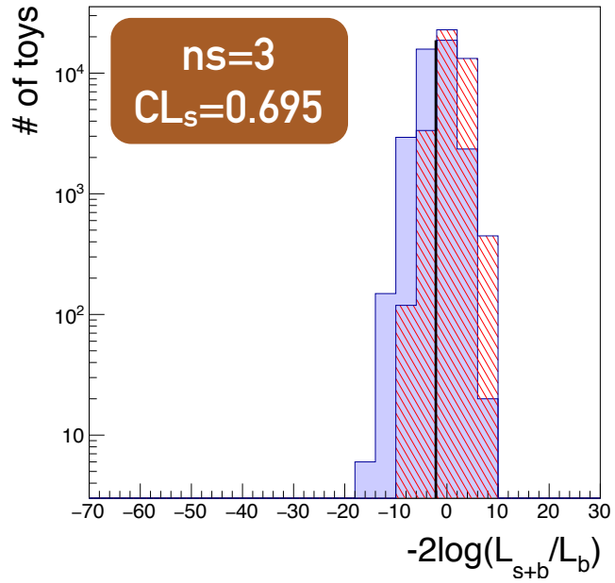


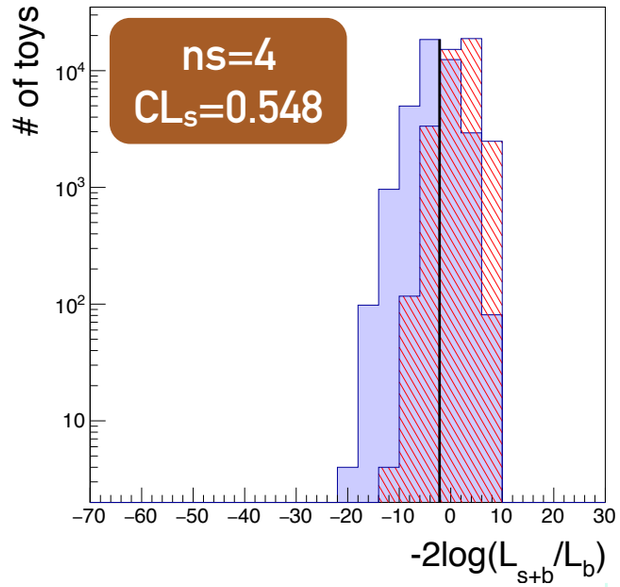

```
Target ns = 7.5,
CLs+b = 0.156, CLb = 0.958, CLs = 0.163
```

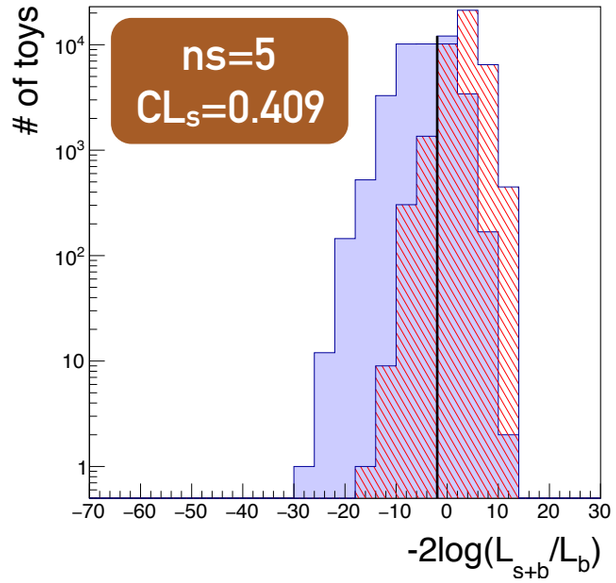
= 7.5 event is an upper limit @ 83.7% C.L.

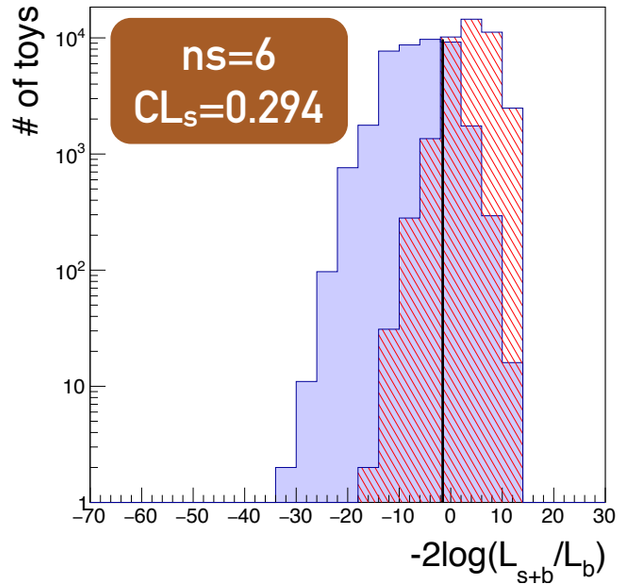
BUT I WANT A 90% AND/OR 95% C.L. LIMIT..

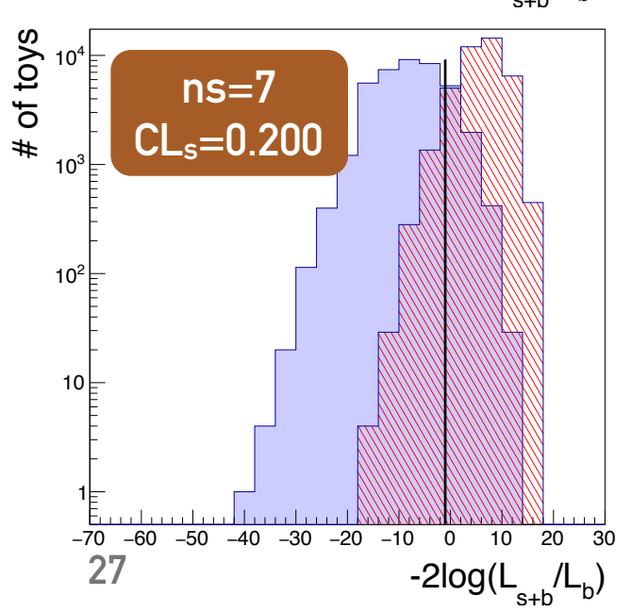
- For a given signal strength, by applying CLs rule the upper limit for the corresponding confidence level (=1–CL_s) can be computed. In order to get the upper limit at desired C.L., one has to scan over the signal strength, e.g.

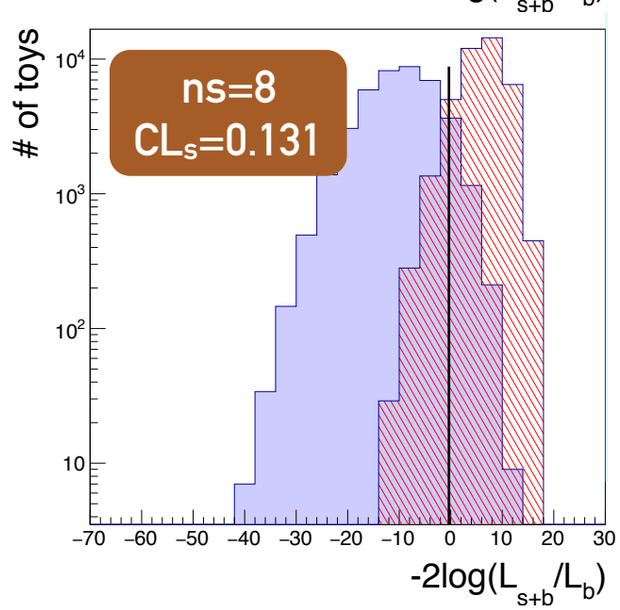


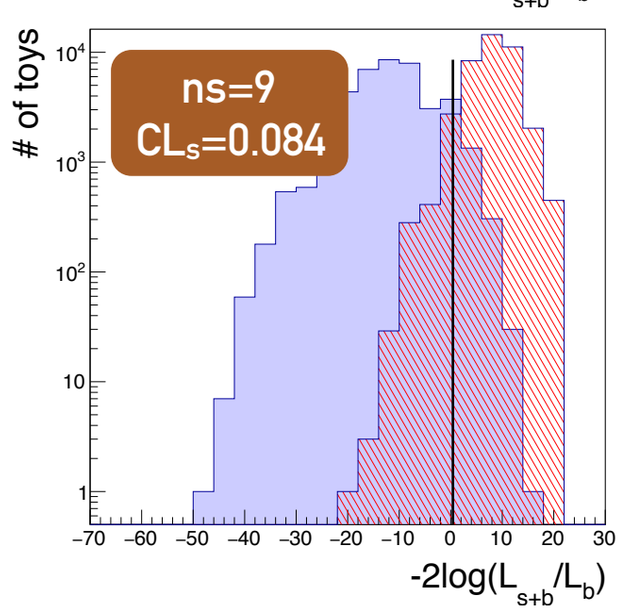


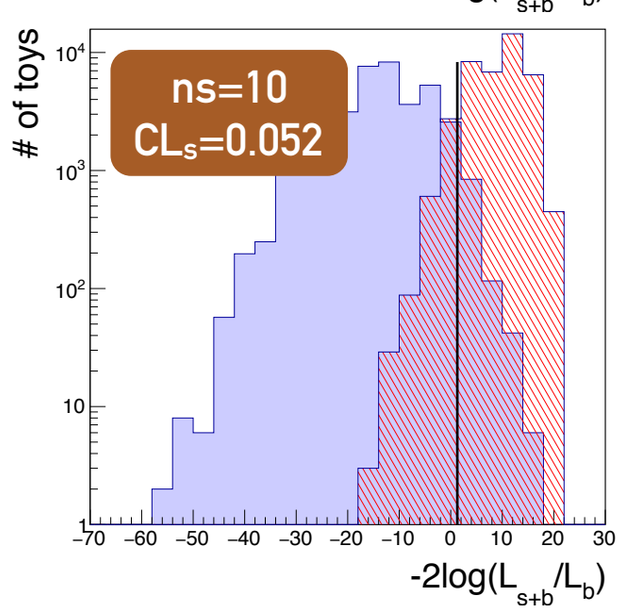










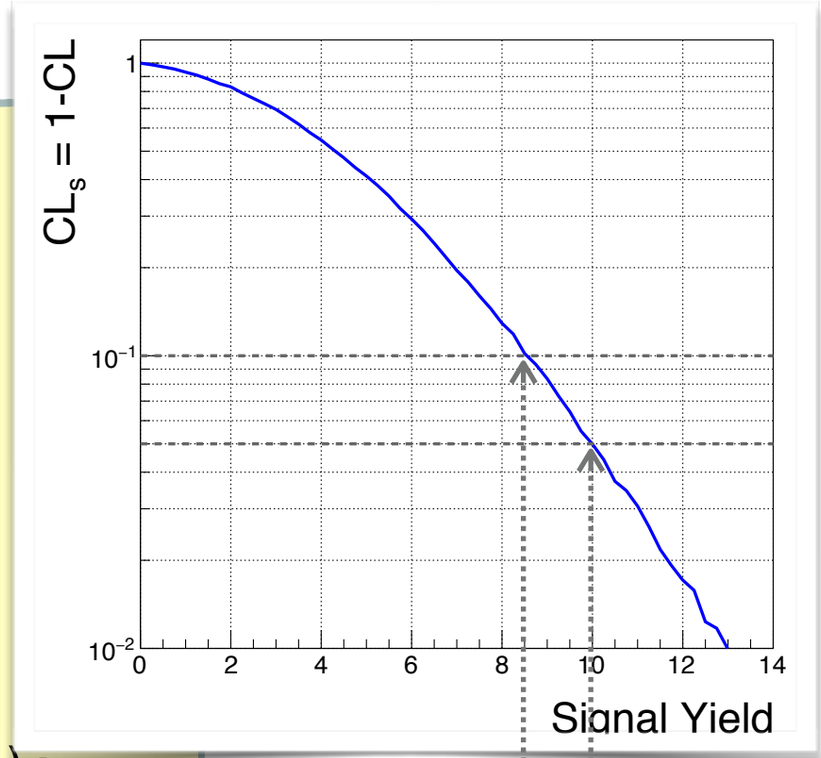


EXAMPLE: SCANNING OVER SIGNAL STRENGTH

- Let's perform such an easy scan and find the proper relation between CL_s and signal strength:

partial **example_01a.cc**

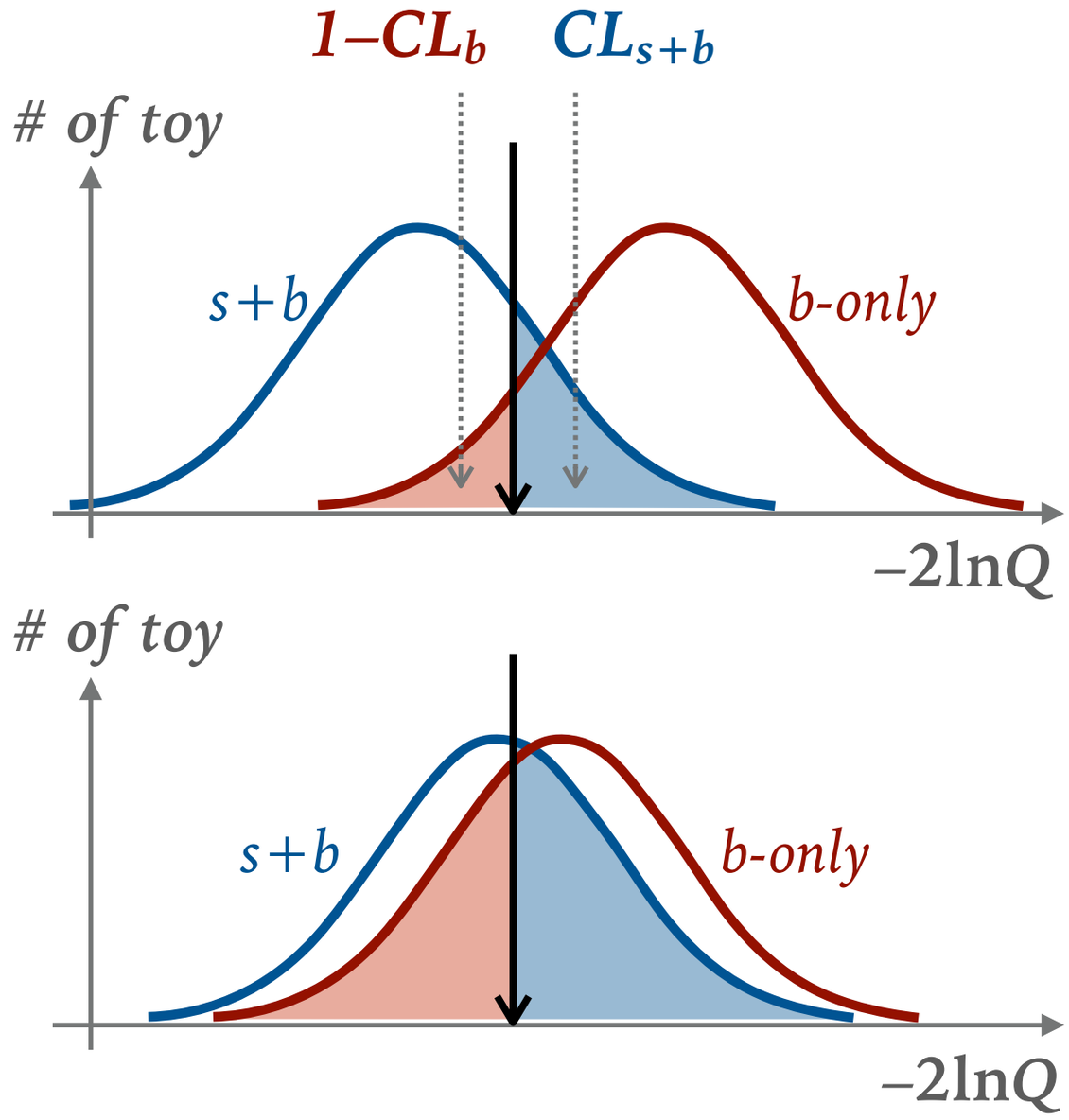
```
void example_01a()
{
    vector<double> vec_x, vec_obs;
    double target_ns = 0.;
    while (target_ns<=14.) {
        double CLs = CL_scan(target_ns,40000);

        vec_x.push_back(target_ns);
        vec_obs.push_back(CLs);
        target_ns += 0.25;
    }

    TGraph *gr_obs = new
        TGraph(vec_x.size(),vec_x.data(),vec_obs.data());

    TCanvas *c1 = new TCanvas("c1","",600,600);
    c1->SetGrid();
    c1->SetLogy();

    TH2D *frame = new TH2D("frame","",10,0.,14.,10,0.01,1.2);
    frame->Draw("");

    gr_obs->Draw("same");

    TLine lin;
    lin.DrawLine(0.,0.05,14.,0.05);
    lin.DrawLine(0.,0.10,14.,0.10);
}
```



~8.5 event is the upper limit @ 90% C.L.
~10 event is the upper limit @ 95% C.L.

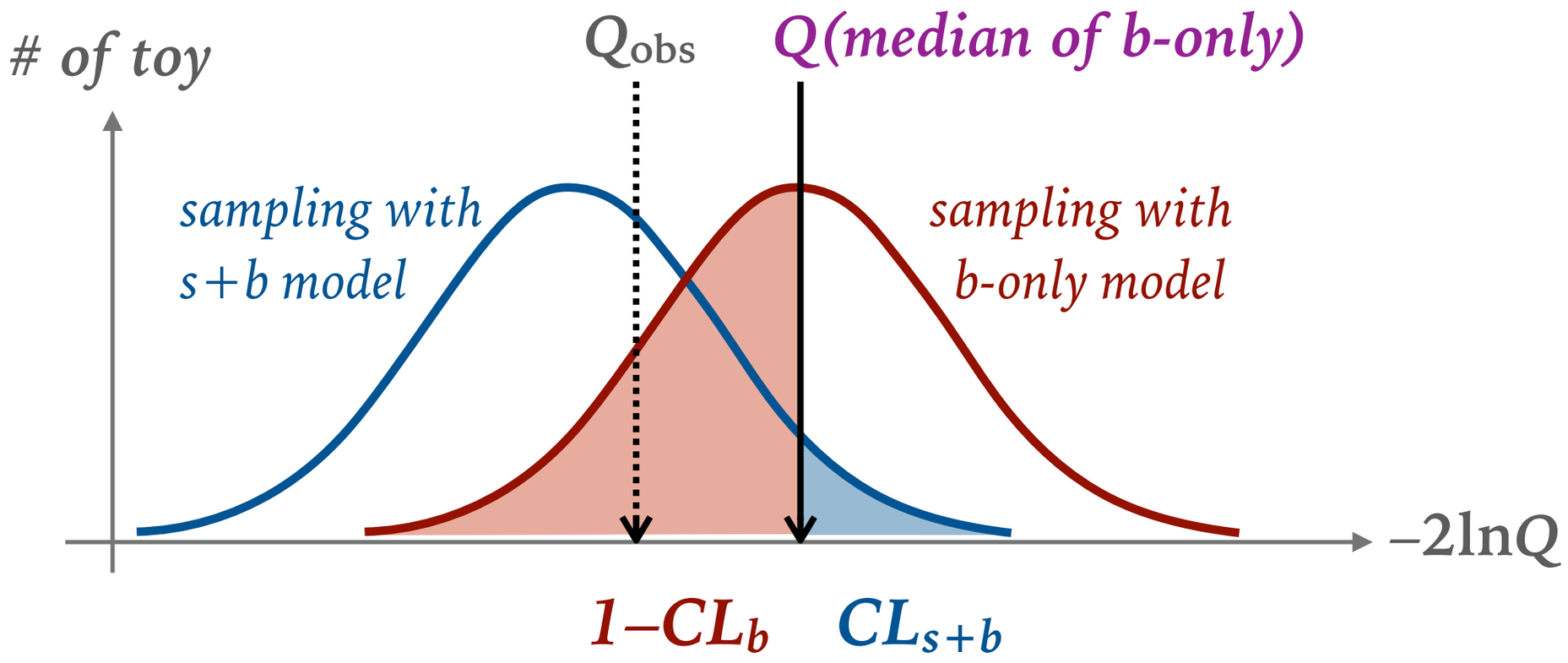
THE FEATURES OF CL_S METHOD

- CL_{s+b}: the probability to obtain a result which is less compatible with the signal than the observed result in the **s+b hypothesis**;
- CL_b: probability to obtain a result less compatible with the signal than the observed one in the **b-only hypothesis**;
- If the two distributions are very well separated, $CL_b \sim 1$ and $CL_s \sim CL_{s+b}$
- If the two distributions are very close each other, CL_b is reduced and preventing CL_s to drop too quick.



HOW ABOUT THE "EXPECTED LIMIT"?

- In the search of unknown signals, it is rather common to provide the "expected limit" for null hypothesis as well as its uncertainty bands.
- The trick is to replace the **observed test statistic** with the **median** of the sampled test statistic from **null hypothesis**.



EXAMPLE: NOW WE HAVE THE EXPECTED LIMIT

partial example_01b.cc

```
void CL_scan(double *CLs, double target_ns = 5.,
             int ntoys = 1000, bool display = false)
{
    const int nobs = 8;
    const double nb = 4.5;
    double Q_thres[4] = {0.,0.,0.,0.};

    double L0 = TMath::Poisson(nobs,nb);
    double L1 = TMath::Poisson(nobs,nb+target_ns);
    Q_thres[0] = -2.*(log(L1)-log(L0)); // observed test statistic
    vector<double> stat[2];
. . . . . .
    sort(stat[0].begin(),stat[0].end());
    Q_thres[1] = stat[0][(int)(0.159*ntoys)]; // null expected 16% (-1s)
    Q_thres[2] = stat[0][(int)(0.500*ntoys)]; // null expected 50% (median)
    Q_thres[3] = stat[0][(int)(0.841*ntoys)]; // null expected 84% (+1s)

    for (int type=0; type<4; type++) {
        double CL[2] = {0.,0.};
        for (int hypo=0; hypo<=1; hypo++) {
            for (int idx=0; idx<ntoys; idx++)
                if (stat[hypo][idx]>=Q_thres[type]) CL[hypo] += 1.;
            CL[hypo] /= (double)ntoys;
        }
        CLs[type] = CL[1]/CL[0];
    }
```

Now we need to keep multiple thresholds on the test statistic

Use the resulting sampling of b-only toy to calculate the thresholds

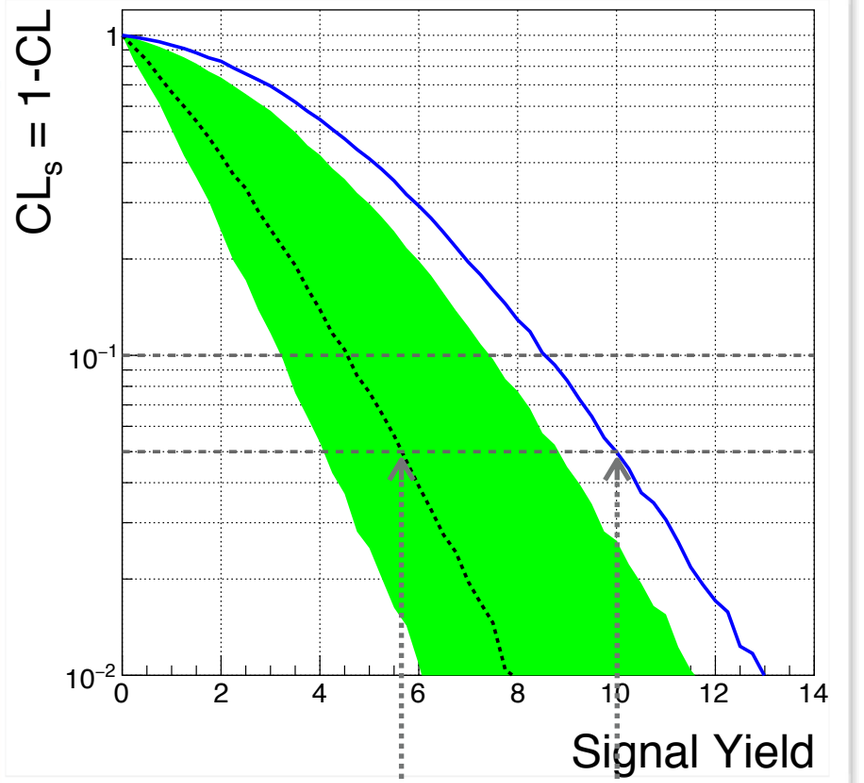
Get the CL_s values for 4 thresholds

EXAMPLE: NOW WE HAVE THE EXPECTED LIMIT (CONT.)

partial example_01b.cc

```
void example_01b()
{
    vector<double> vec_x, vec_obs, vec_exp[3];
    double target_ns = 0.;
    while (target_ns<=14.) {
        double CLs[4];
        CL_scan(CLs,target_ns,40000);

        vec_x.push_back(target_ns);
        vec_obs.push_back(CLs[0]);
        vec_exp[0].push_back(CLs[1]);
        vec_exp[1].push_back(CLs[2]);
        vec_exp[2].push_back(CLs[3]);
        target_ns += 0.25;
    }

    TGraph *gr_obs = new
        TGraph(vec_x.size(),vec_x.data(),vec_obs.data());
    TGraph *gr_exp = new
        TGraph(vec_x.size(),vec_x.data(),vec_exp[1].data());
    TGraph *gr_experr = new TGraph(vec_x.size()*2);
    for (int i=0; i<vec_x.size(); i++) {
        gr_experr->SetPoint(i,vec_x[i],vec_exp[2][i]);
        gr_experr->SetPoint(vec_x.size()*2-1-i,vec_x[i],vec_exp[0][i]);
    }
. . . . . .
    gr_experr->Draw("fsame");
    gr_exp->Draw("same");
    gr_obs->Draw(“same");
```

4 values in one call



Observed limit @ 95% C.L. ~10 events
Expected limit @ 95% C.L. ~5.8 events

NUISANCE PARAMETERS TREATMENT

- Whenever we are doing the experiments, nuisance parameters (*those uninterested parameters, systematics, etc*) are something cannot be avoid.
- Two main approaches exist:
  - **Add the nuisance parameters to your likelihood model**
    - The model might become complicated.
    - Easier to incorporate in a fit but not in the upper limits
  - **Bayesian way ⇒ "Integrate it out"**

$$P(\theta|X) = \int P(\theta,\lambda|X)d\lambda = \frac{\int L(X|\theta,\lambda)\pi(\theta,\lambda)d\lambda}{\int L(X|\theta',\lambda)\pi(\theta',\lambda)d\theta' d\lambda}$$

Obtain P(θ|X) as a "marginal PDF" by integrate out λ.

θ: Parameter of interests
λ: nuisance parameters
π(θ,λ): prior PDF

THE FREQUENTIST WAY

- One can introduce a **complementary dataset** to constrain the nuisance parameters (*e.g. calibration data point, background estimates from control sample, etc.*).
- Extend the likelihood function to formulate the statistical problem in terms of both the main data sample (X) and control sample (Y):

$$L(X, Y|\theta, \lambda) = L(X|\theta, \lambda)L(Y|\lambda)$$

- The "control sample" can be just a randomized constrained PDF mean if it was introduced as a constraint in the original data fit.
- This method makes the likelihood function more complex, will take more time to minimize at Minuit (usually CPU intensive!).
- **Alternative solution ⇒ the "hybrid" method.**

THE COUSINS–HIGHLAND HYBRID APPROACH

- Unfortunately there is no fully solid general approach to incorporate nuisance parameters in a frequentist approach.
- The hybrid approach was proposed by **Cousins & Highland** **[Ref. NIM A320 (1992) 331-335]**
- Like the Bayesian method ⇒ **integrate ("*marginalize*") the likelihood function over the nuisance parameters**.

$$L^{\text{hybrid}}(X|\theta,\widetilde{\lambda}) = \int L(X|\theta,\lambda) f(\widetilde{\lambda};\lambda) d\lambda$$

- Take the Bayesian way of integration over PDF, then use the likelihood in a frequentist way:
  - **Bayesian-frequentist "hybrid" approach**.
  - Not guaranteed to provide exact frequentist coverage.
  - Gives very similar results to Bayesian limit with a uniform prior.

RooStats::HybridCalculator available

PROFILE LIKELIHOOD APPROACH

RooStats::
ProfileLikelihoodCalculator
available

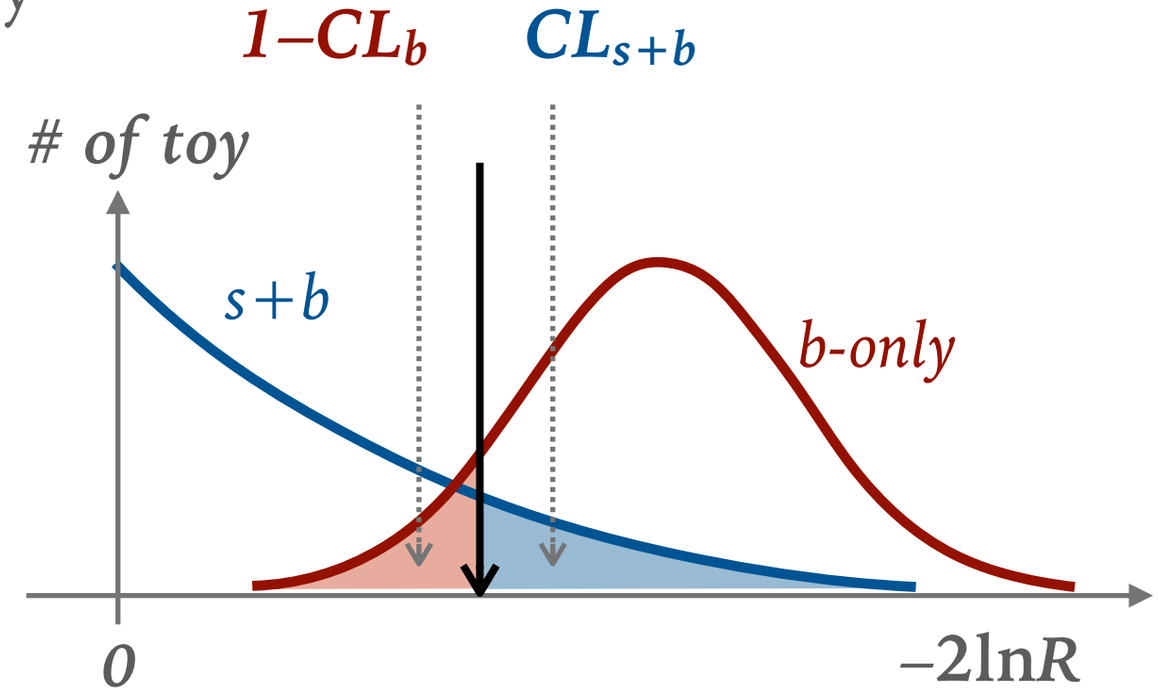
- As we already discussed in the previous lecture, the **profile likelihood** is the way to reduce the parameters and only focus on the main one.
- Here a different test statistic is introduced (instead of classical **L(s+b)/L(b)**):

$$R = \frac{L(\theta, \hat{\hat{\lambda}})}{L(\hat{\theta}, \hat{\lambda})}$$

← fix θ, fit λ

← fit θ and λ

- The likelihood function is "*broadened/smeared*" by nuisance parameters.
- Distribution of **$-2\ln R$** tends to a **χ^2 distribution with 1 degree of freedom** if there is only one parameter of interest.



Resulting different test statistic distributions
(basically the same as the profile likelihood treatment in F&C study)

THE "LHC" VERSION OF PROFILE LIKELIHOOD CL_S

- The standard agreement between CMS and ALTAS (for the Higgs combination, of course).
- It use **profile likelihood** as test statistics in CL_s, but with an upper bound of target signal strength in the minimization.

$$R = \frac{L(\theta, \lambda)}{L(\hat{\hat{\theta}}, \hat{\hat{\lambda}})} \quad \begin{matrix} \leftarrow \text{fix } \theta, \text{ fit } \lambda \\ \leftarrow \text{fit } \theta \text{ and } \lambda \end{matrix} \quad \Rightarrow \quad 0 \leq \hat{\theta} \leq \theta \quad \leftarrow \text{fitted } \theta \text{ cannot be larger then the target strength } \theta$$

- This constraint ensures that upward fluctuations of the data are not considered as evidence against the signal hypothesis (pure 1-side limit).
- The agreed nuisance treatment is the **Frequentist way** instead of the hybrid method.

FROM LEP, TEVATRON, TO LHC

- Let's summarize the methods commonly used in different experiments.
- Now you might be aware of these different statistical interpretations when you are reading the papers from them!

	Test statistic	Profiled?	Nuisance treatment
LEP	$-2\ln\dfrac{L(\theta,\widetilde{\lambda})}{L(0,\widetilde{\lambda})}$	**No**	**Hybrid method**
Tevatron	$-2\ln\dfrac{L(\theta,\hat{\lambda}_\theta)}{L(0,\hat{\lambda}_0)}$	**Yes**	**Hybrid method**
LHC	$-2\ln\dfrac{L(\theta,\hat{\lambda}_\theta)}{L(\hat{\theta},\hat{\lambda})}$	**Yes w/ upper cap $(0\leq\hat{\theta}\leq\theta)$**	**Frequentist**

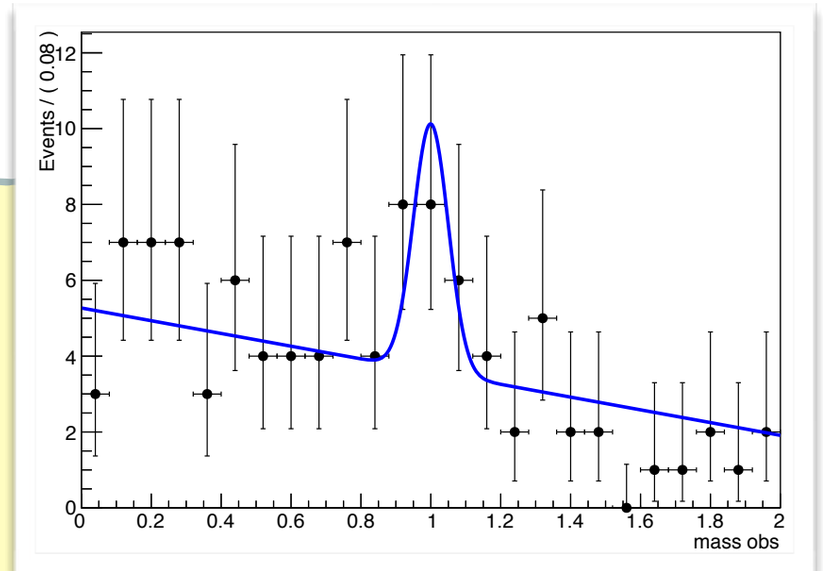
$\widetilde{\lambda}$: nominal nuisance $\qquad$ $\hat{\theta},\hat{\lambda}$: minimized POI and nuisance

$\hat{\lambda}_\theta$: minimized nuisance at θ

$\hat{\lambda}_0$: minimized nuisance at $\theta = 0$

STEP-BY-STEP LHC-STYLE CL_S LIMIT CALCULATION

- Before ending of this lecture, let's perform a step-by-step calculation of the "LHC-style" CL_s limit.
- Adopt a slightly different model from the previous F&C study example: (note: `example_splusb.root` *is available on the lecture web as well!*):



example_02.cc

```
  TFile *fin = new TFile("example_splusb.root");
  TNtupleD* nt = (TNtupleD *)fin->Get("nt");

  RooRealVar mass("mass","mass obs",0.,2.);

. . . . . .

  RooRealVar ns("ns","ns",10,0.,1000.);
  RooRealVar nb("nb","nb",90,0.,1000.);
  RooAddPdf model("model","mass PDF",RooArgList(gaus,linear),RooArgList(ns,nb));

  RooRealVar slope_mu("slope_mu","slope mean",-0.3);
  RooRealVar slope_sigma("slope_sigma","slope sigma",0.03);
  RooGaussian cons("cons","constrained PDF",slope,slope_mu,slope_sigma);

  RooDataSet data("data","data",nt,RooArgSet(mass));

  model.fitTo(data,ExternalConstraints(cons));
```

In order to demo the nuisance treatment, add a Gaussian constraint to the "slope"

TOY MC WITH FULL FREQUENTIST APPROACH

- In the toy Monte Carlo generation we need to take into account the constrained parameter (*here it is the slope of background shape*).
- As discussed in the earlier slide, the likelihood function has to be extended to include both the main sample and complementary sample (*here it is exactly the constrained term!*):

$$L(M|n_s, n_b, S) = L(M|n_s, n_b, S)L(S; S_\mu, S_\sigma)$$ ←constrained PDF

- Note we already have the joint likelihood function by adding the ExternalConstraints option in RooFit. So all we need to do is to have the proper complementary sample generated together with the main sample.
- In practice this is just **randomizing the constrained mean** (S_μ) at each toy fit, since it is the **complementary sample**!

TOY MC WITH FULL FREQUENTIST APPROACH (II)

partial example_03.cc

```
void freq_toy(double target_ns = 10., int ntoys = 100)
{
  RooWorkspace *ws = new RooWorkspace("wspace");
  buildModel(ws);
```

Import the model into a RooWorkspace from previous example

```
  ws->var("ns")->setVal(target_ns);
  ws->var("ns")->setConstant(true);
  ws->pdf("model")->fitTo(*ws->data("data"),ExternalConstraints(*ws->pdf("cons")));

  TH1D *pull_ns = new TH1D("pull_ns","",60.,-6.,6.);
  TH1D *pull_slope = new TH1D("pull_slope","",60.,-6.,6.);
```

Fit to data w/ fixed signal to target ns

```
  for (int idx=0; idx<ntoys; idx++) {
    RooWorkspace* ws_toy = new RooWorkspace(*ws);
```

Nominal toy generation

```
    // main data
    RooDataSet *toy=ws_toy->pdf("model")->generate(*ws_toy->var("mass"),Extended(true));

    // complementary data
    ws_toy->var("slope_mu")->setVal(rnd.Gaus(ws_toy->var("slope")->getVal(),
        ws_toy->var("slope_sigma")->getVal()));
```

randomize slope mean

```
    ws_toy->var("ns")->setConstant(false);
    ws_toy->pdf("model")->fitTo(*toy,ExternalConstraints(*ws_toy->pdf("cons")));

    pull_ns->Fill((ws_toy->var("ns")->getVal()-ws->var("ns")->getVal())/
                  ws_toy->var("ns")->getError());
    pull_slope->Fill((ws_toy->var("slope")->getVal()-ws->var("slope")->getVal())/
                     ws_toy->var("slope")->getError());
   }
. . . . . .
```

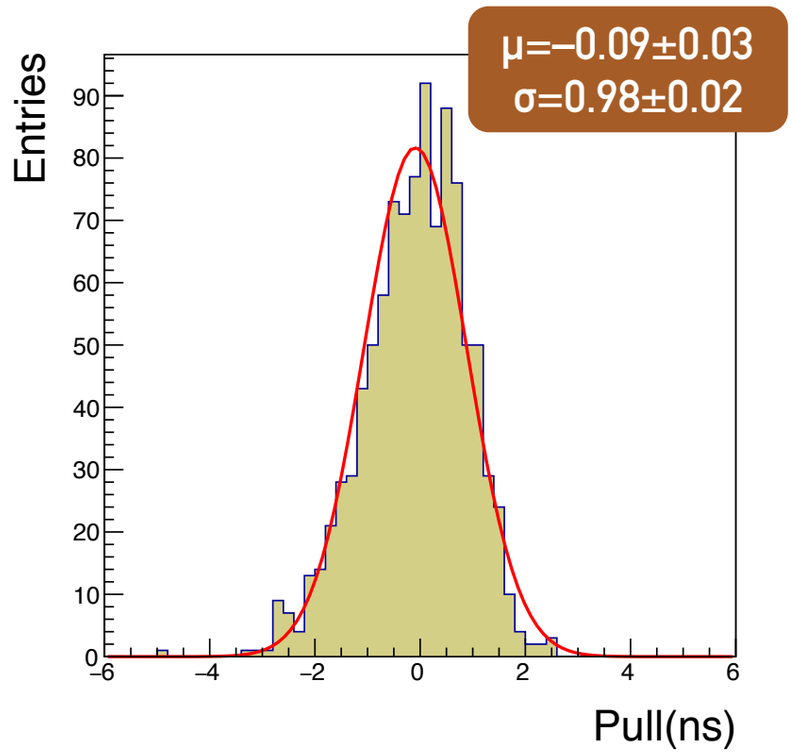
Display the pull distributions

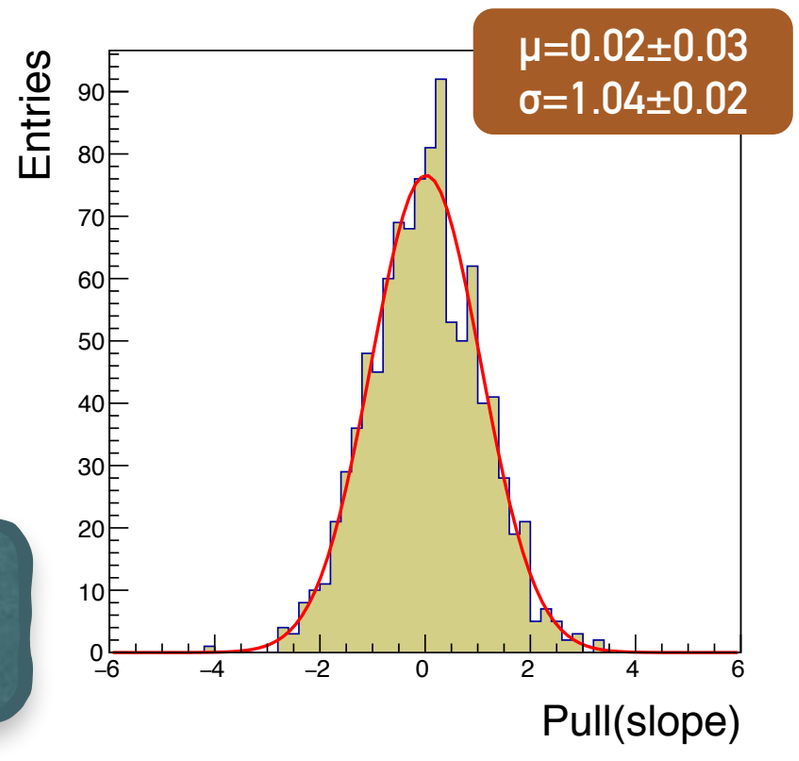
TOY MC WITH FULL FREQUENTIST APPROACH (III)

- The rest is just plotting and fit.
- You can try to switch off the operation of randomizing slope mean, the pull distribution will definitely broken!

partial example_03.cc

```
. . . . . .
  TCanvas *c1 = new TCanvas("c1","",400,800);
  c1->Divide(1,2);
  for (auto& hist : {pull_ns, pull_slope}) {
      hist->SetStats(false);
      hist->GetYaxis()->SetTitle("Entries");
  }
  c1->cd(1);
  pull_ns->GetXaxis()->SetTitle("Pull(ns)");
  pull_ns->Fit("gaus","L");
  c1->cd(2);
  pull_slope->GetXaxis()->SetTitle("Pull(slope)");
  pull_slope->Fit("gaus","L");

  delete wspace;
}

void example_03()
{  freq_toy(10.,1000); }
```





This will give you the ~ok pull distributions!

MERGE THEM INTO THE CL_S CALCULATION

- Now this is the real work: calculation of CL_s according to the test statistic distributions from toy.
- Recall: "LHC-style" takes the profiled likelihood ratio as the test statistic, and there is an upper bound in the minimization:

partial example_04.cc

```
void CL_scan(double *CLs, double *CLs_err, double target_ns = 10., . . .)
{
  RooWorkspace *wspace = new RooWorkspace("wspace");
  buildModel(wspace);
  double Q_thres[4] = {0.,0.,0.,0.};

  wspace->var("ns")->setMax(target_ns);
  wspace->var("ns")->setVal(target_ns);
  wspace->var("ns")->setConstant(false);
  RooFitResult *res0 = wspace->pdf("model")->fitTo(
    *wspace->data("data"),ExternalConstraints(*wspace->pdf("cons")),Save(true));

  wspace->var("ns")->setVal(target_ns);
  wspace->var("ns")->setConstant(true);
  RooFitResult *res1 = wspace->pdf("model")->fitTo(
    *wspace->data("data"),ExternalConstraints(*wspace->pdf("cons")),Save(true));

  Q_thres[0] = max(0.,res1->minNll()-res0->minNll())*2.; // observed test stat
. . . . . .
```

Fit #1: everything float, but ns<=target_ns

Fit #2: ns=target_ns, minimizing everything else.

$$-2\ln\frac{L(\theta,\hat{\hat{\lambda}}_\theta)}{L(\hat{\theta},\hat{\lambda})}$$

MERGE THEM INTO THE CL_S CALCULATION (II)

partial example_04.cc

```
vector<double> stat[2];
for (int hypo=0; hypo<=1; hypo++) {
  for (int idx=0; idx<ntoys; idx++) {
    RooWorkspace* ws_toy = new RooWorkspace(*wspace);
    if (hypo==0) ws_toy->var("ns")->setVal(0.);
    if (hypo==1) ws_toy->var("ns")->setVal(target_ns);

    // main data
    RooDataSet *toy = ws_toy->pdf("model")->generate(*ws_toy->var("mass"));

    // complementary data
    ws_toy->var("slope_mu")->setVal(rnd.Gaus(
      ws_toy->var("slope")->getVal(),ws_toy->var("slope_sigma")->getVal()));

    ws_toy->var("ns")->setVal(target_ns);
    ws_toy->var("ns")->setConstant(false);
    RooFitResult *res0 = ws_toy->pdf("model")->fitTo(
      *toy,ExternalConstraints(*ws_toy->pdf("cons")),Save(true));

    ws_toy->var("ns")->setVal(target_ns);
    ws_toy->var("ns")->setConstant(true);
    RooFitResult *res1 = ws_toy->pdf("model")->fitTo(
       *toy,ExternalConstraints(*ws_toy->pdf("cons")),Save(true));

    double Q = max(0.,res1->minNll()-res0->minNll())*2.; // sampled test stat
    stat[hypo].push_back(Q);
  }
}
```

Either b-only or s+b hypothesis

full frequentist approach

Fit #1

Fit #2

MERGE THEM INTO THE CL_S CALCULATION (III)

partial example_04.cc

```
sort(stat[0].begin(),stat[0].end());
Q_thres[1] = stat[0][(int)(0.159*ntoys)]; // null expected 16% (-1s)
Q_thres[2] = stat[0][(int)(0.500*ntoys)]; // null expected 50% (median)
Q_thres[3] = stat[0][(int)(0.841*ntoys)]; // null expected 84% (+1s)
```

Same method as the previous example

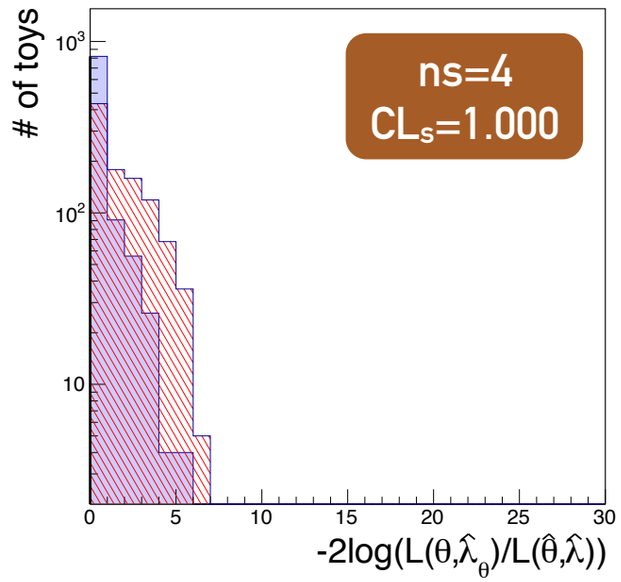
```
for (int type=0; type<4; type++) {
  double CL[2] = {0.,0.}, CL_err[2] = {0.,0.};
    for (int hypo=0; hypo<=1; hypo++) {
      for (int idx=0; idx<ntoys; idx++)
        if (stat[hypo][idx]>=Q_thres[type]) CL[hypo] += 1.;
      CL[hypo] /= (double)ntoys;
      CL_err[hypo] = sqrt(CL[hypo]*(1.-CL[hypo])/ntoys);
    }
  CLs[type] = CL[1]/CL[0];
  if (CLs[type]>0.) CLs_err[type] =
      sqrt(pow(CL_err[0]/CL[0],2)+pow(CL_err[1]/CL[1],2))*CLs[type];
}
```

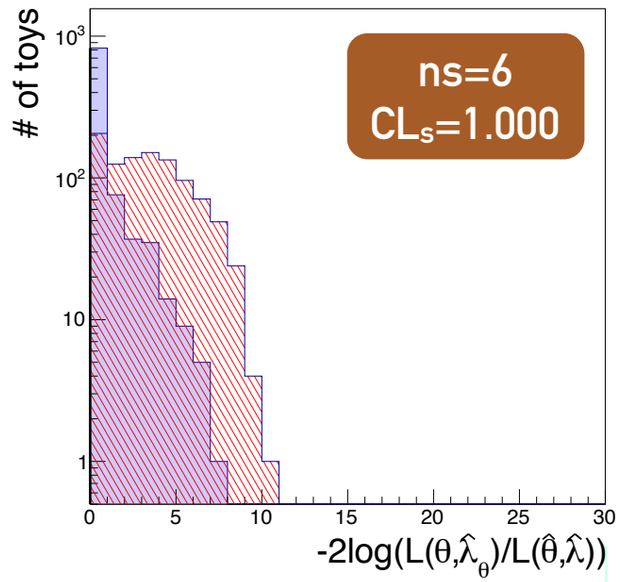
Also the same as before, except now we also calculate the errors.

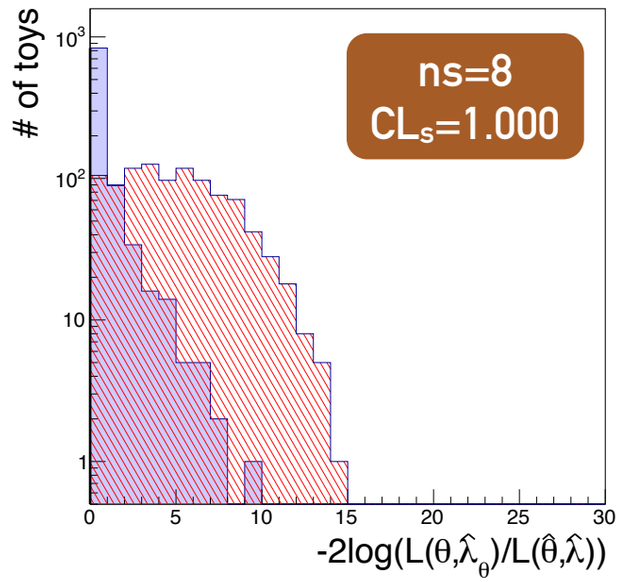
```
  if (display) {
. . . . . .
  }

  printf("Target ns = %.1f, observed:%.3f +- %.3f\n",target_ns,CLs[0],CLs_err[0]);
  printf("16%% null expected: %.3f +- %.3f\n",CLs[1],CLs_err[1]);
  printf("50%% null expected: %.3f +- %.3f\n",CLs[2],CLs_err[2]);
  printf("84%% null expected: %.3f +- %.3f\n",CLs[3],CLs_err[3]);
```

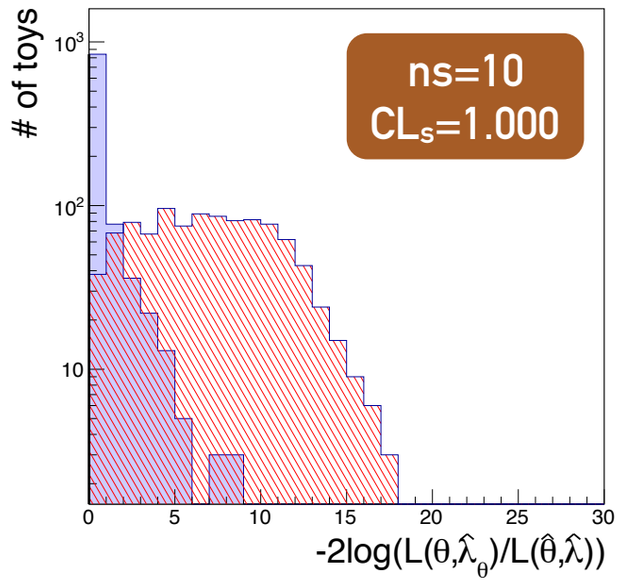
MERGE THEM INTO THE CL_S CALCULATION (IV)

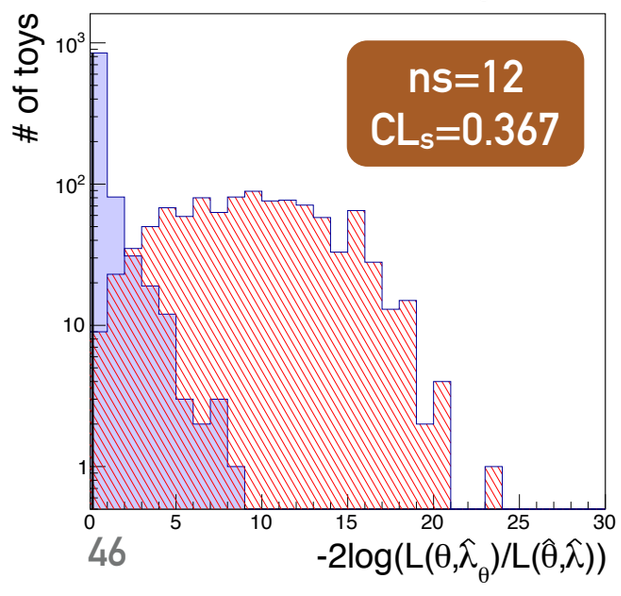
- Almost the same as the previous example (*except for a more complicated test statistic!*)
- In order to get the upper limit at the desired confidence level, one has to scan over the signal strength as well, e.g.

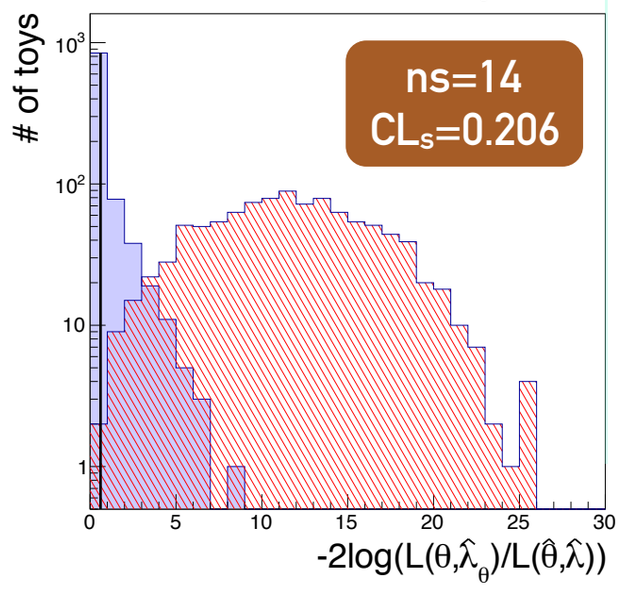


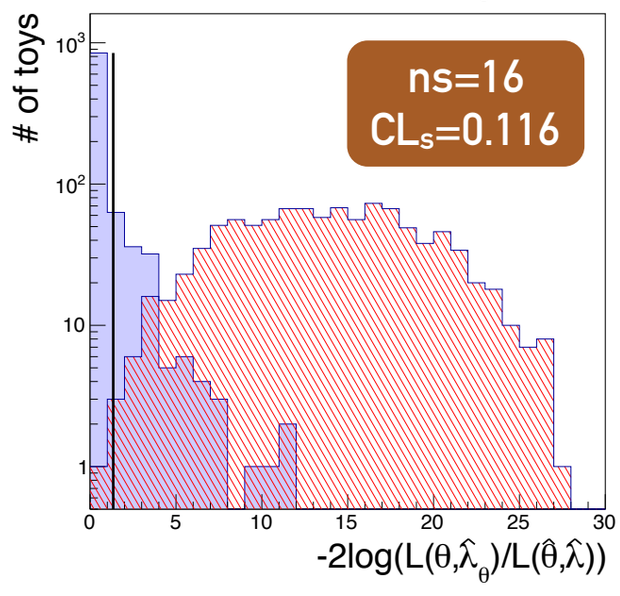


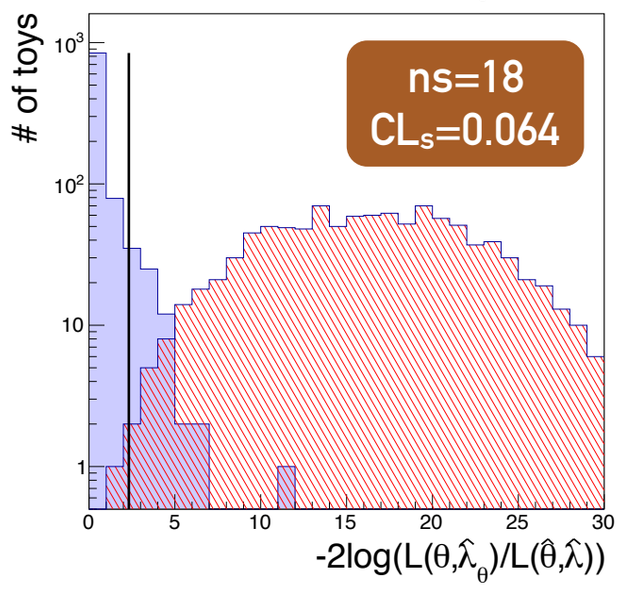










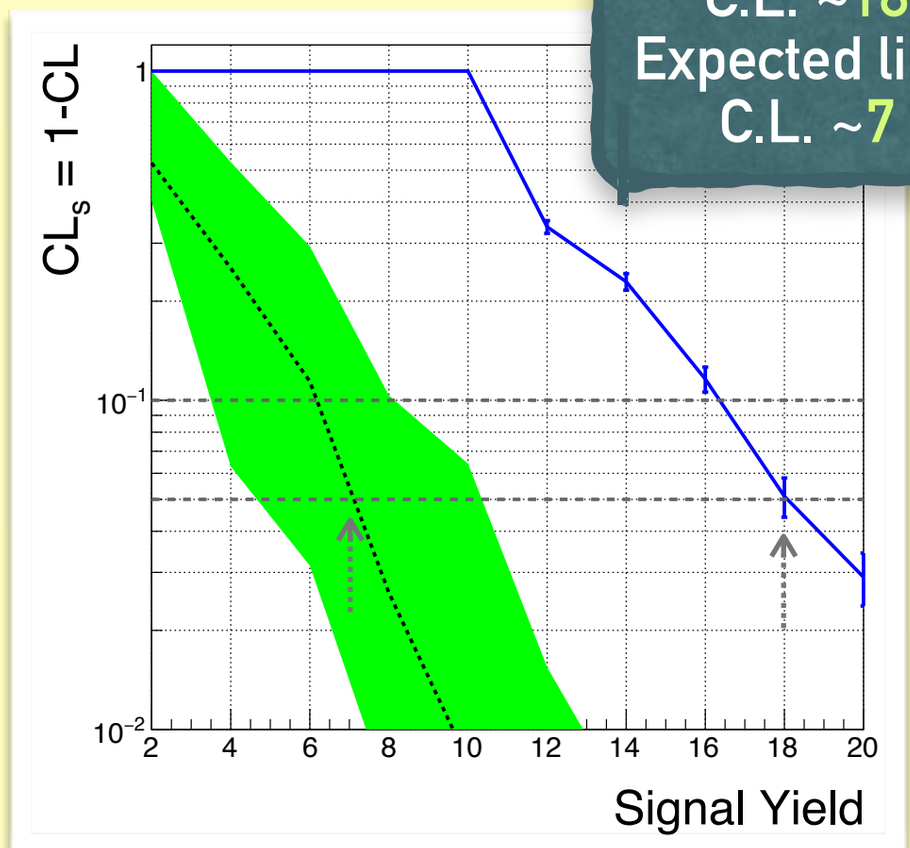


SCANNING OVER SIGNAL STRENGTH

- As we already did in the Poissonian example, a scanning over the signal strength can be carried out in a similar way:

partial example_04a.cc

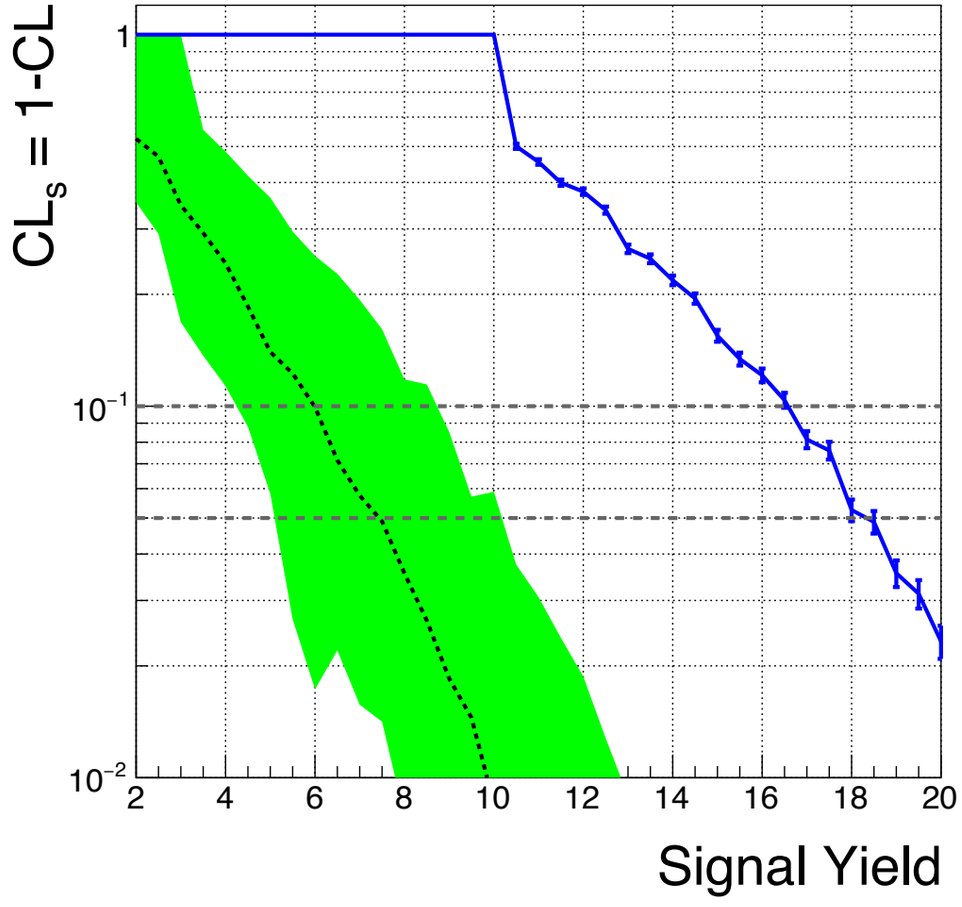
```
void example_04a()
{
  vector<double> vec_x, vec_obs,
                 vec_obserr, vec_exp[3];
  double target_ns = 2.;
  while (target_ns<=20.) {

    double CLs[4], CLs_err[4];
    CL_scan(CLs,CLs_err,target_ns,1000);

    vec_x.push_back(target_ns);
    vec_obs.push_back(CLs[0]);
    vec_obserr.push_back(CLs_err[0]);
    vec_exp[0].push_back(CLs[1]);
    vec_exp[1].push_back(CLs[2]);
    vec_exp[2].push_back(CLs[3]);

    target_ns += 2.;
  }

  TGraphErrors *gr_obs = new TGraphErrors(
    vec_x.size(),vec_x.data(),vec_obs.data(),0,vec_obserr.data());
  TGraph *gr_exp = new TGraph(vec_x.size(),vec_x.data(),vec_exp[1].data());
  TGraph *gr_experr = new TGraph(vec_x.size()*2);
. . . . . .
```

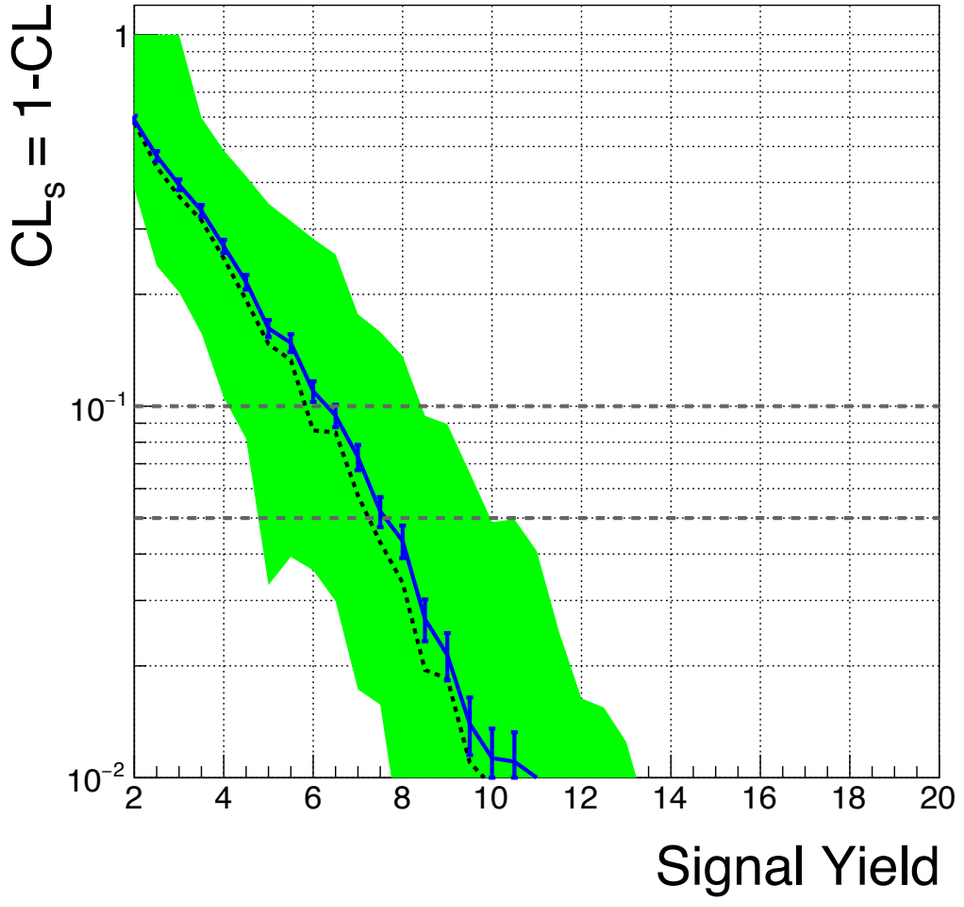


Observed limit @ 95% C.L. ~18 events
Expected limit @ 95% C.L. ~7 events

SCANNING OVER SIGNAL STRENGTH (II)

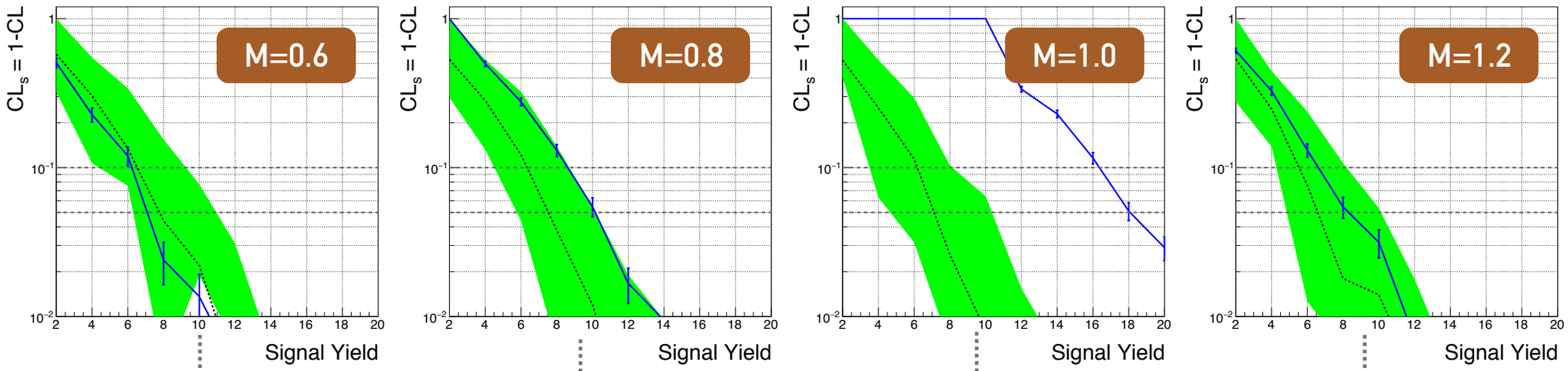
- You may try to run it with more toys as well as finer step.
- Or, there is another data file contains no signal excess (`example_bonly.root`). If you replace it as the observed data, you can see now the “observed limit” is pretty much agree with the “expected limit”!

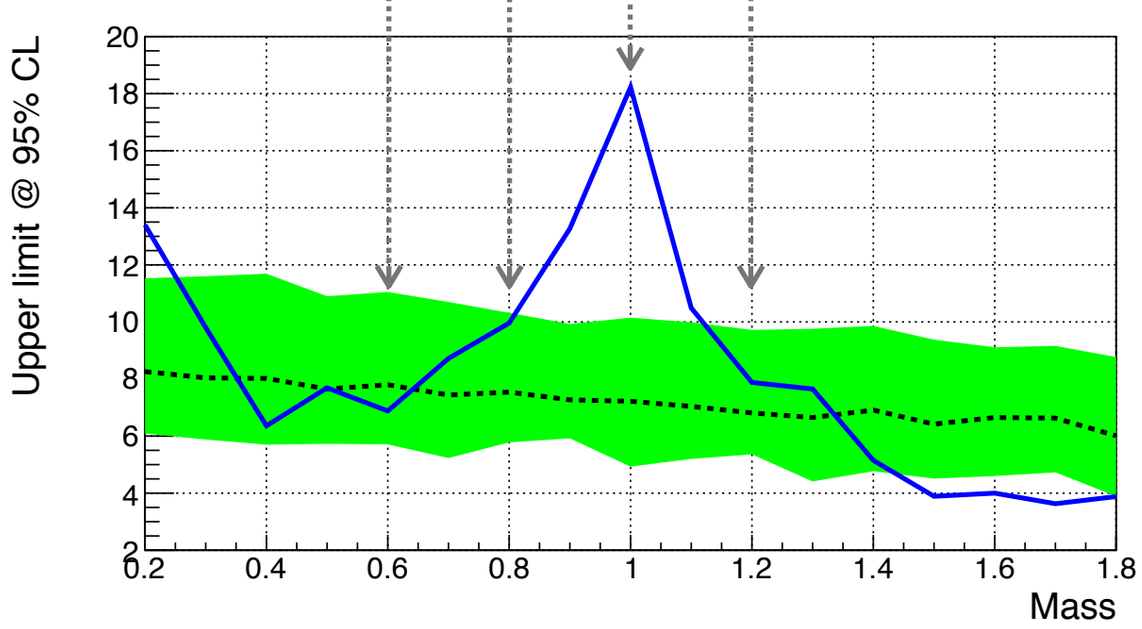




SCANNING OVER SIGNAL STRENGTH (III)

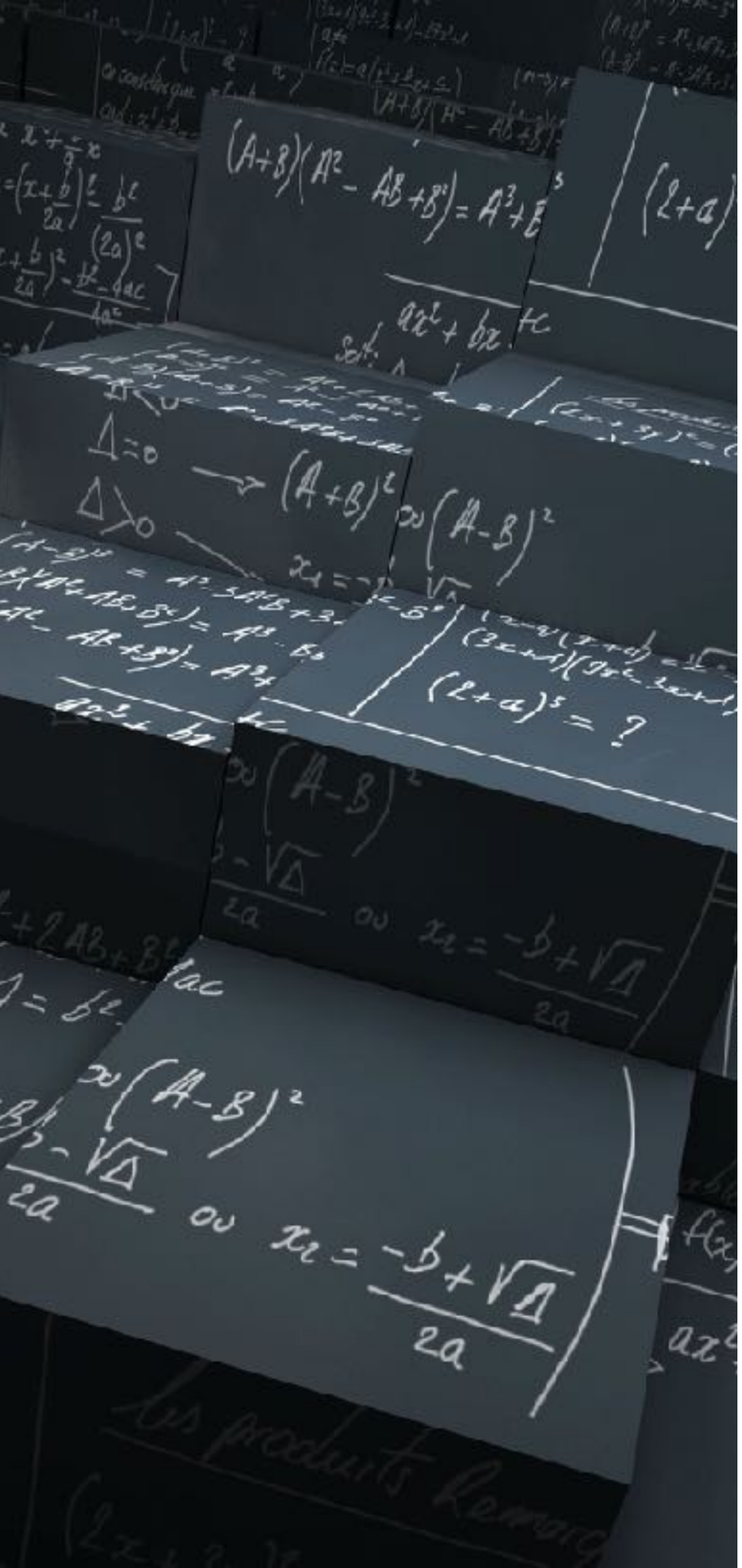
- Note the previous calculate was always assumed the signal position (*mass*) is fixed at the given value. In principle one has to repeat the study for different signal assumption, e.g.





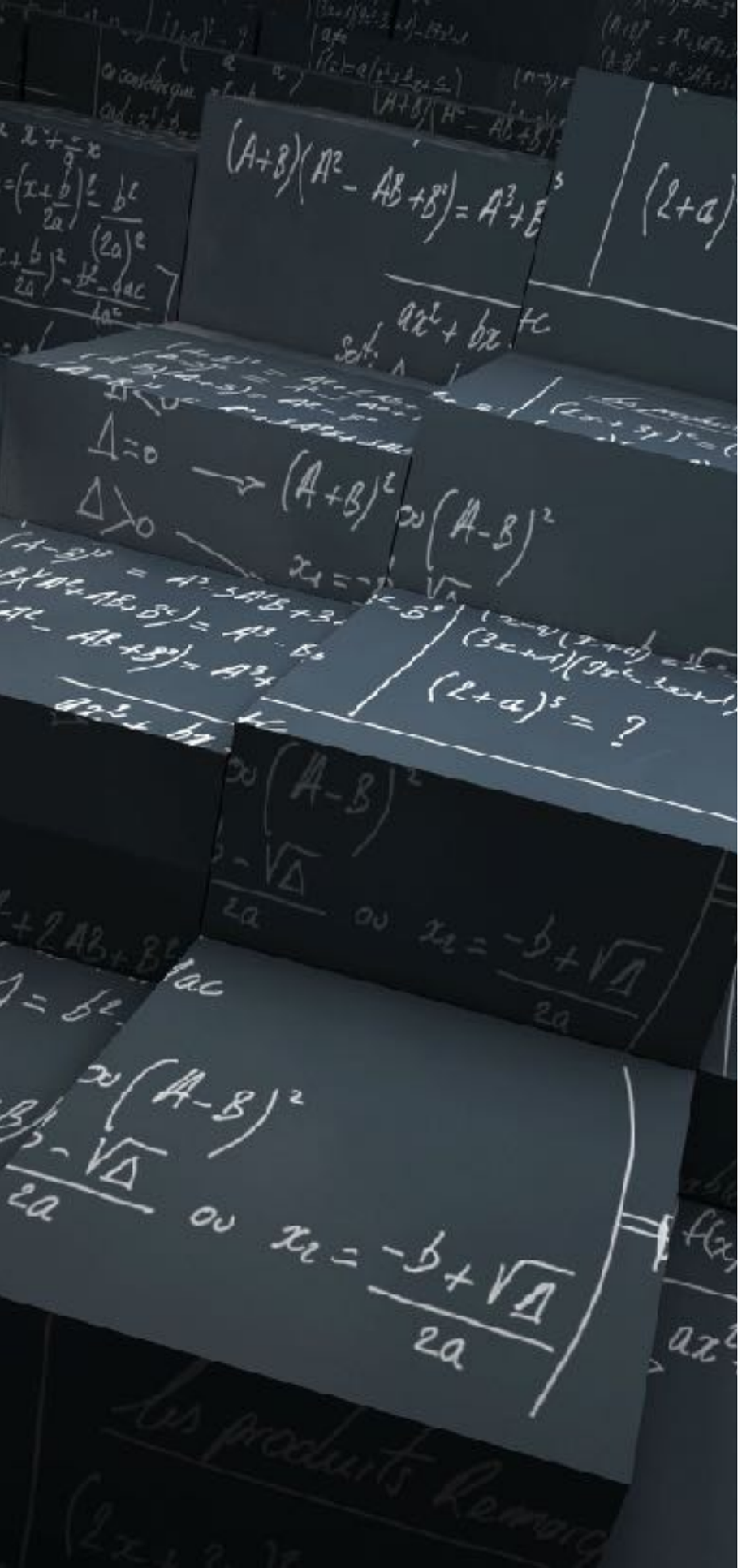
Scanning over different signal position assumption

→ A familiar limit plot?



SUMMARY

- In this lecture we touched the basis of hypotheses testing as well as the most-adopted (*and most confusing…*) upper limit calculation method CL_s in the LHC analyses.
- This should allow you to go a little bit deeper, when producing an upper limit as your analysis result.
- Now you should be able to do the same calculation as the official package!



COMMENT: SKIPPED TOPICS

- We end our lecture here with the following topics skipped. If you are interested (*or need for your own work*), you are welcome to discuss with me offline.
  - Statistics in multivariate analysis;
  - Goodness-of-Fit test;
  - Kolmogorov test;
  - Asymptotic approximations in limit calculation;
  - Asimov data sets;
  - Look-Elsewhere Effect;
  - …and anything you can think of!