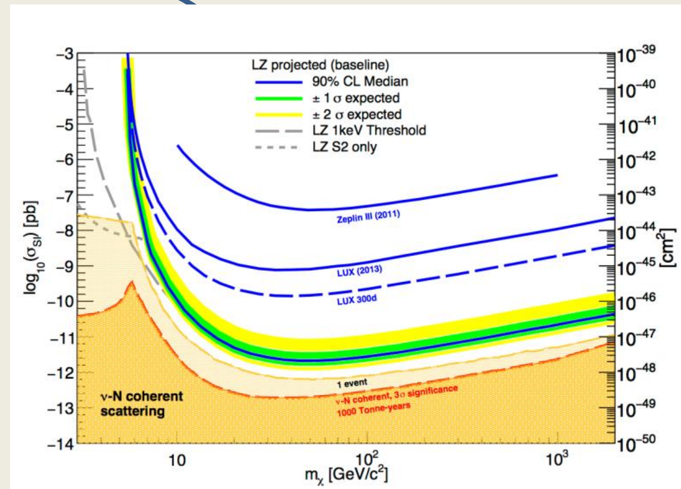


LUX PLR Overview

Shaun Alsum

LUX Goals

- Discover WIMPs
  - Necessary requirement: establish inconsistency of LUX results with no-WIMP scenario
- Set Limits (quantify inconsistency of data with the existance of WIMPs)

Profile Likelihood Ratio

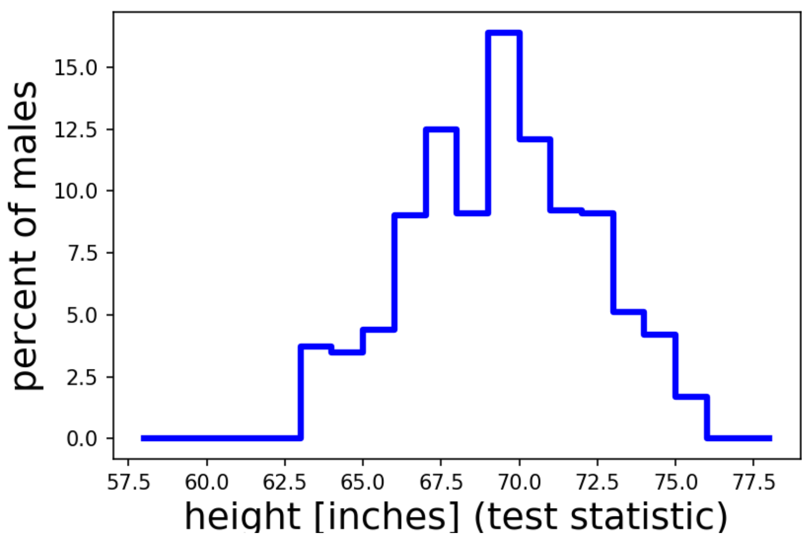
- The PLR is the statistical machinery that accomplishes the above goals
- The PLR is nothing more than a prescription for how to choose parameters for, and execute a series of *frequentist hypothesis tests*.

Hypothesis test

- Statistical procedure for establishing the consistency or inconsistency of a set of data with a stated hypothesis.
- The procedure is as follows:
  - Precisely state the *null hypothesis*
  - Choose a *test statistic*
  - Determine the PDF for the null hypothesis as a function of the test statistic
  - Establish an acceptance/rejection region4
  - Compare the test statistic of the measurement to the PDF to determine whether it lies in the acceptance region, or rejection region.

Example: verifying a person's sex by height

- Null Hypothesis: person X is a male.
- Test statistic: height in inches
- PDF



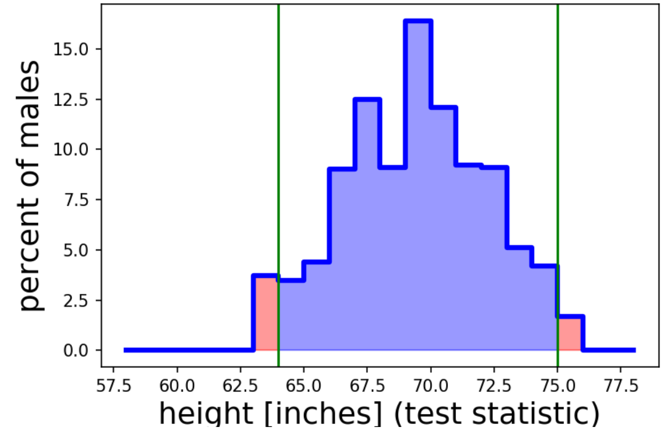
Data shown is for males between 20 and 29 years old from a US census survey conducted in 2007
https://www2.census.gov/library/publications/2010/compendia/statab/130ed/tables/11s0205.pdf

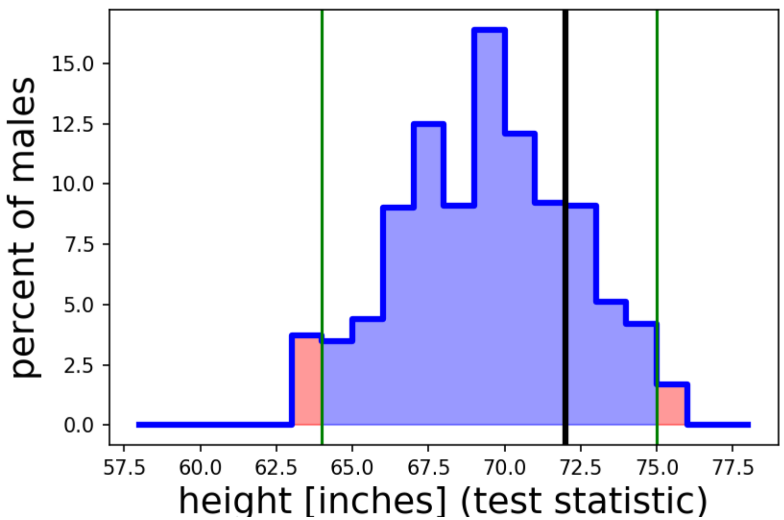
Example: verifying a person's sex by height (continued)

- Acceptance/rejection region
- Measure the height of person x

and compare

  – (say we measured 72")





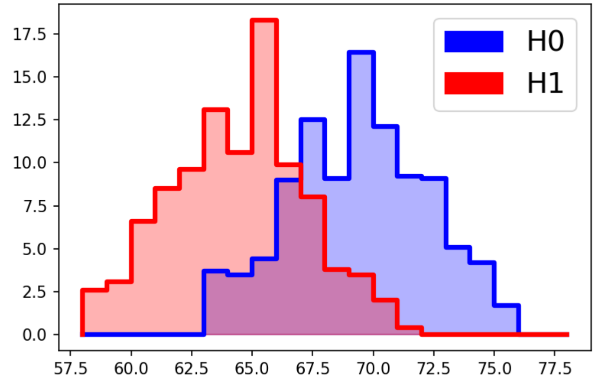
72" lies within our acceptance region, therefore person x is consistent with being male under our statistical test.

Hypothesis Tests Improved

- Sometimes we have a specific alternative to our null hypothesis we wish to test against. Our test can be improved by stating an *alternative hypothesis* in addition to our null hypothesis.
- There are now two possible errors in our test:
  - Type 1: the null hypothesis is true and we reject it
  - Type 2: the alternative hypothesis is true and we reject it
- The *Significance* of a test (typically denoted α) is the fraction of the time the null hypothesis would be rejected, even if it were true.
- The *Power* of a test is $1 - \beta$, where β is the fraction of the time that the null hypothesis is accepted, even if the alternative hypothesis is true.
- The best tests will make both α and β as small as possible.

Example: Sex by height revisited

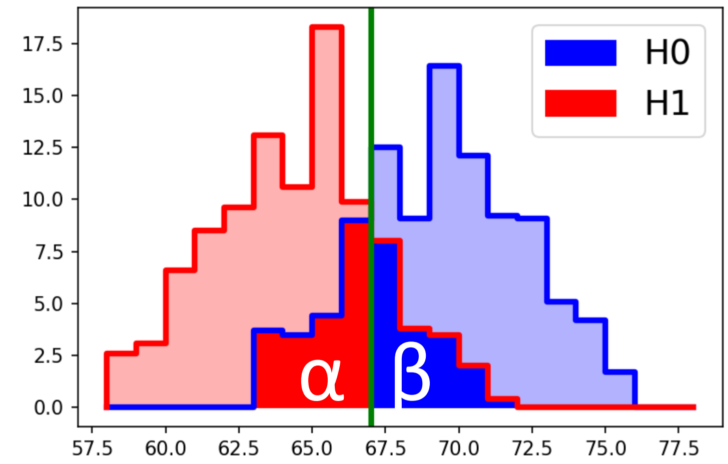
- Null Hypothesis (H0): person X is a male.
- Alternative Hypothesis (H1): person X is a female.
- Test statistic: height in inches
- PDF

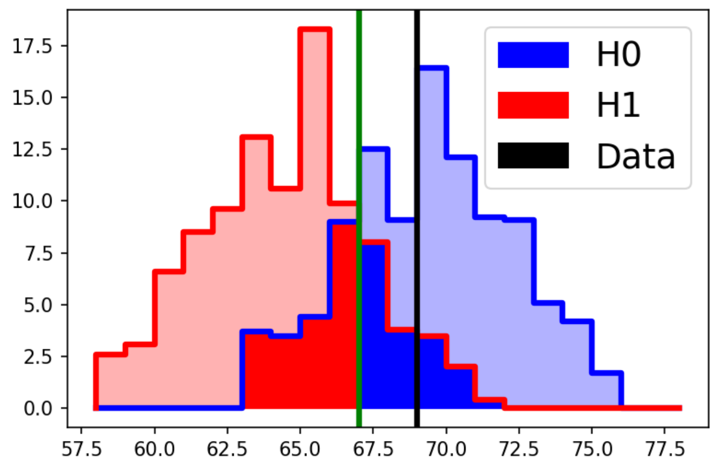


Same deal as before with the data

Example: Sex by height revisited (continued)

- Acceptance/Rejection Region
  - Probability of Type 1 error (given H0 true): α
  - Probability of Type 2 error (given H1 true): β
- Compare: Say we measure 69”





Conclusion: Person X is consistent with being male with significance α.

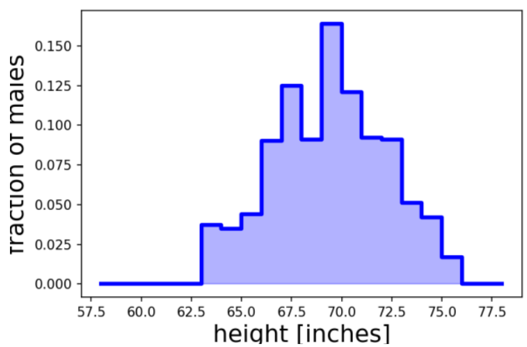
Caution: this does not mean person X has a 1-α probability of being male.

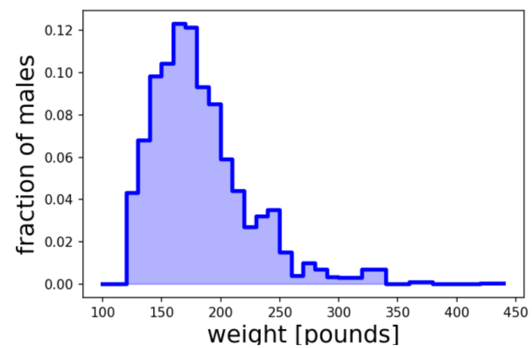
Hypothesis Tests: Improved More

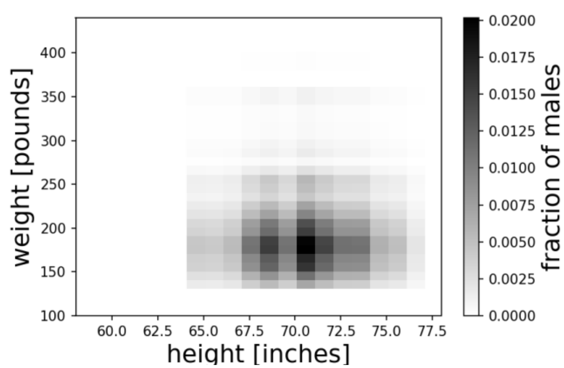
- Often we can measure more than one observable and would like to use both. In this case, we can choose our test statistic to be a function of multiple observables.
- An excellent way (though not the only way) to accomplish this is to use the likelihood of your model given your data as your test statistic.
  - Note: until this point, our PDF has served only inform our choice of acceptance region (and allow us to report our significance α). Now we are using it in our calculations; how else are we going to compare quantities of different units?

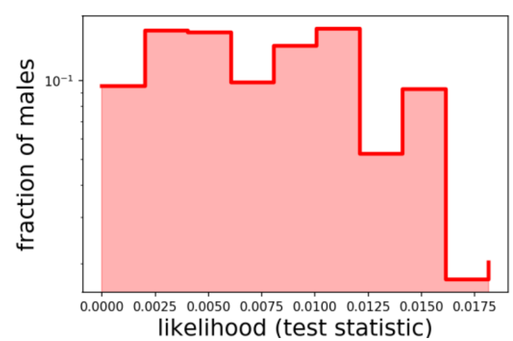
Example: Sex by Height and Weight

- Null Hypothesis (H0): person X is a male.
- Test Statistic: $\mathcal{L}(male;(h,w)) = P_{height}(x) * P_{weight}(w)$
  - here I assume height and weight are uncorrelated, clearly a bad assumption, but I didn't want to deal with it (and didn't have data in hand). Should have a 2D PDF instead of a product of 2 1-D PDFs
- PDF



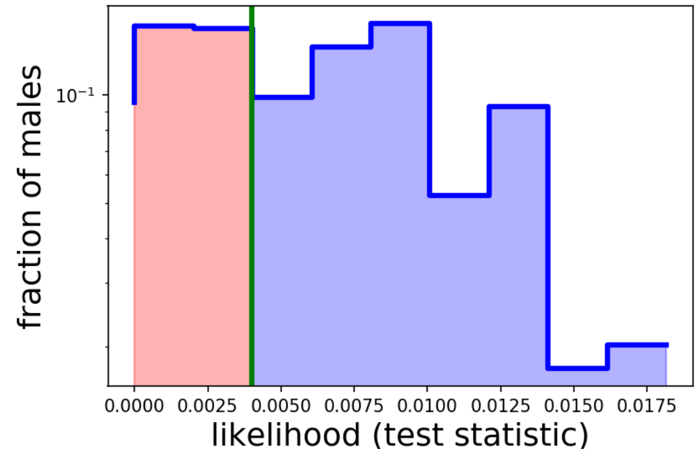


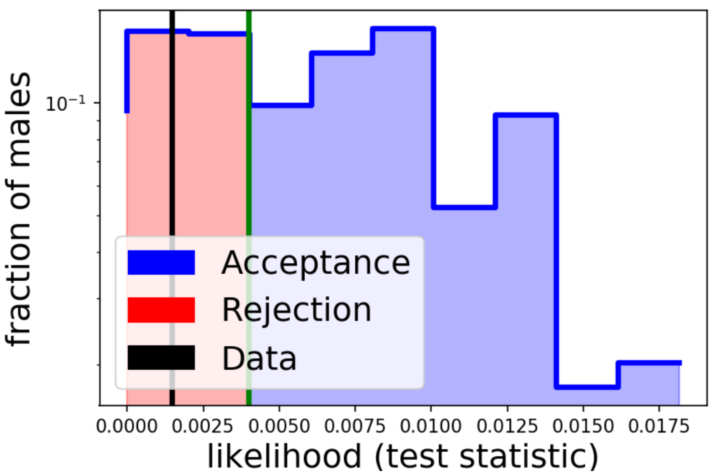




Example: Sex by Height and Weight (continued)

- Acceptance/Rejection region
  - accept region of higher likelihood
- Compare: say we measure 75" and 250 lbs
  - Calculate the likelihood of the male model based on this data: 0.00147





Conclusion: person X is inconsistent with being male with significance α

Hypothesis Tests: The Final Improvement (Neyman Pearson Lemma)

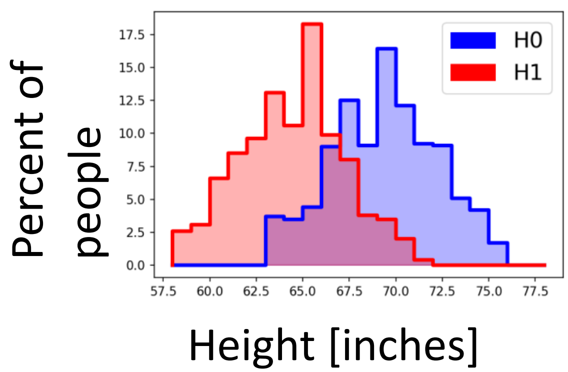
- When using an alternative hypothesis, we want to minimize both α and β.
- Neyman Pearson Lemma:

For a hypothesis test between two simple hypothesis (hypothesis whose PDFs can be precisely and completely specified), given a choice for either α or β, the choice of test statistic which minimizes the other is the ratio of the null hypothesis' likelihood to the alternative hypothesis' likelihood. And the choice of acceptance region is including regions of decreasing likelihood ratio until the remaining fractional area is α.

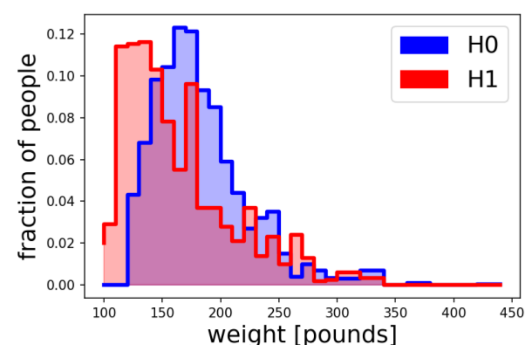
$\lambda = \frac{\mathcal{L}(H0;x)}{\mathcal{L}(H1;x)}$ sometimes the test statistic $q = -2\ln\lambda$ is used

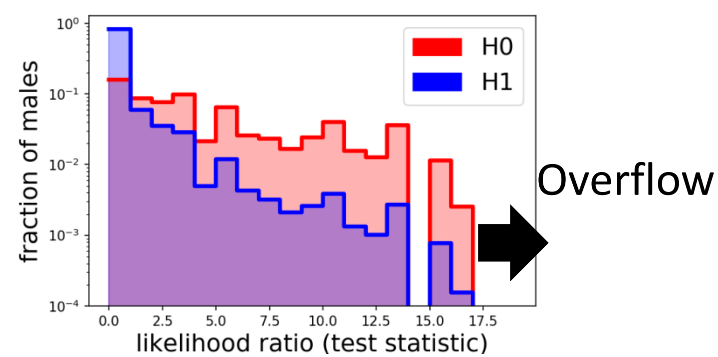
Example: Sex by Height and Weight Revisited

- Null Hypothesis (H0): person X is a male.
- Alternative Hypothesis (H1): person X is a female.
- Test Statistic: $\lambda = \frac{\mathcal{L}(H0;(h,w))}{\mathcal{L}(H1;(h,w))}$
- PDF:



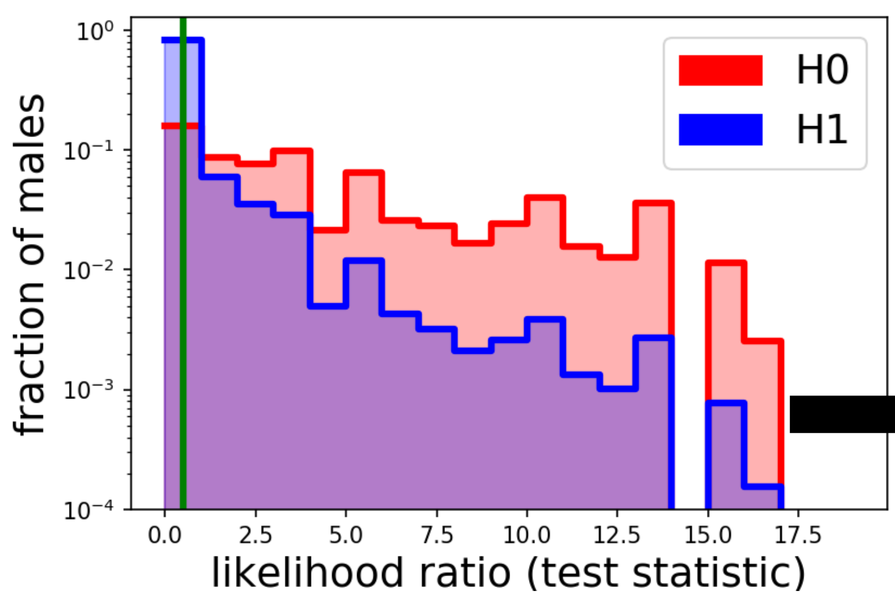
+





Example: Sex by Height and Weight Revisited (continued)

- Acceptance/Rejection region: binning too large to represent accurately, but essentially rejection region is a fraction of the left bin if we choose α = 0.05
- Compare: say that again we measure 75” and 250 lbs



Conclusion: Person X is consistent with being male

Data comparison

Hypothesis Tests: Extension to Non-simple Hypothesis

- What if we cannot specify our PDFs precisely? (as is actually most often the case in reality)
- the *most likely* model for the null and alternative hypotheses given the data in the likelihood ratio instead of a-priori determined PDFs.
  - Parameterize the PDFs in terms of nuisance parameters.
  - Choose the values of these parameters (independently for the null and alternative hypotheses) such that the likelihood for the null hypothesis in the numerator is maximized, and again so that the likelihood for the alternative hypothesis in the denominator is maximized.
  - If one has an idea of what these nuisance parameters might be, a term can be added to the likelihood function, *profiling* the parameter, typically decreasing the likelihood the further the parameter is from its expected value.

Example: Counting Experiment

- Suppose one wants to determine whether the activation of a beam causes a signal in a detector.
- The detector takes measurements for an amount of time T without the beam turned on and measures m events
- The detector then takes measurements for an amount of time T with the beam turned on and measures n events.
- Null hypothesis: the beam creates no signal
- Alternative hypothesis: the beam adds, on average, p events per time T
- Test statistic: likelihood ratio
  - Likelihood function no profiling:

  $Pois(n; b + s * p) * Pois(m; b)$ where s = 0 for the null hypothesis and s = 1 for the alternative hypothesis

  - Likelihood function with profiling:

  $Pois(n; b + s * p) * Pois(m; b)$*Gaus(b; b_{exp}, b_{std})

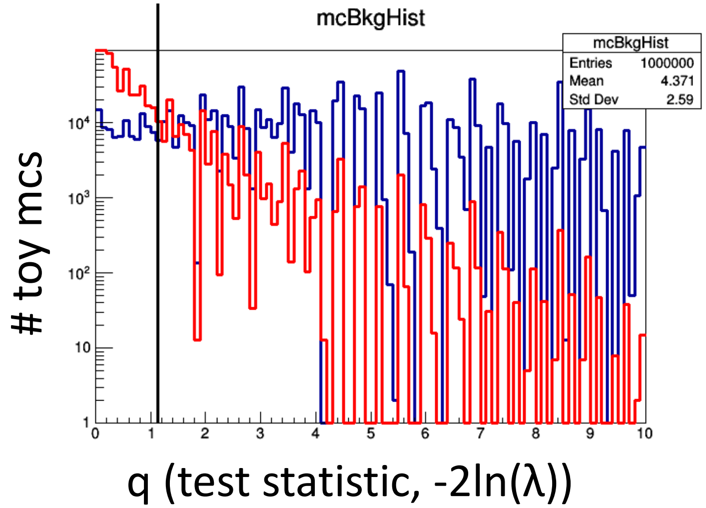
  Nuisance parameters: b

PLR: Generating the PDF

- We already stated that in our use case, we cannot determine the PDFs for the null or alternative hypotheses completely, so how do we proceed?
- Toy Monte Carlos: Generate data sets from our models using values of nuisance parameters drawn from our profiling.
- The collection of test statistics calculated from these toy mcs forms our PDF.

Example: Counting Experiment (Continued)

- PDF:



Red: signal + background
Blue: background only
Black: data

n = 15, m=20 (but bkg only run went for 2T)
p = 5 s = 1.1

Using the PLR to Discover a WIMP

- Null hypothesis (H_0): There are no WIMPs (parameter of interest = 0).
- Alternative hypothesis (H_1): There are WIMPs (parameter of interest ≠ 0).
- Test Statistic: $\lambda = \frac{\mathcal{L}(\mu=0;\,\hat{\hat{\theta}})}{\mathcal{L}(\mu\neq 0;\hat{\theta})}$, $q = -2\ln\lambda$
  - Numerator: fix POI (represented by μ), let all other nuisance parameters (represented by θ) float and maximize.
  - Denominator: POI float in addition to all nuisance parameters
    - The single vs double hat only serves to represent that the values settled on for the nuisance parameters need not be the same in the two likelihood expressions.
- Our significance for discovery is chosen to be 5σ, or $\alpha = 3\text{x}10^{-7}$.

Setting a Limit

- The limit we set is the bound of a 90% frequentist confidence interval.
  - This is a procedure for setting boundaries that will result in a band covering the true parameter 90% of the time.
  - Note, this does NOT mean that there is a 90% chance that our parameter of interest lies within the band.
    - From the original confidence band paper (by Neyman)

    "Can we say that in this particular case the probability of the true value [falling between these limits] is equal to α? The answer is obviously in the negative. The parameter is an unknown constant, and no probability statement concerning its value may be made..."

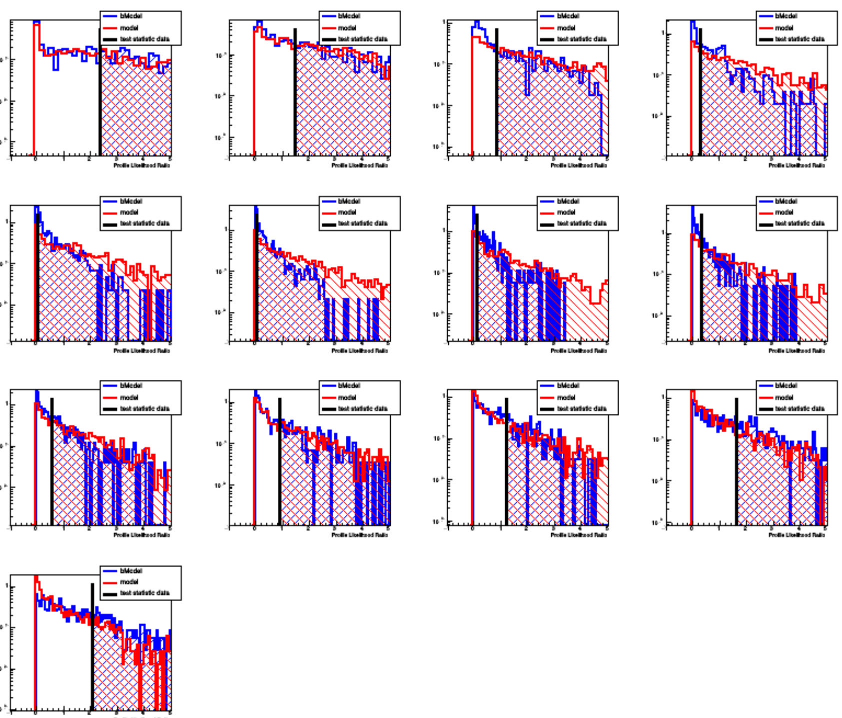
    Further expounding found on wikipedia…

    "[this idea] seems rooted in a (not uncommon) desire for Neyman-Pearson confidence intervals to provide something which they cannot legitimately provide; namely, a measure of the degree of probability, belief, or support that an unknown parameter value lies in a specific interval."

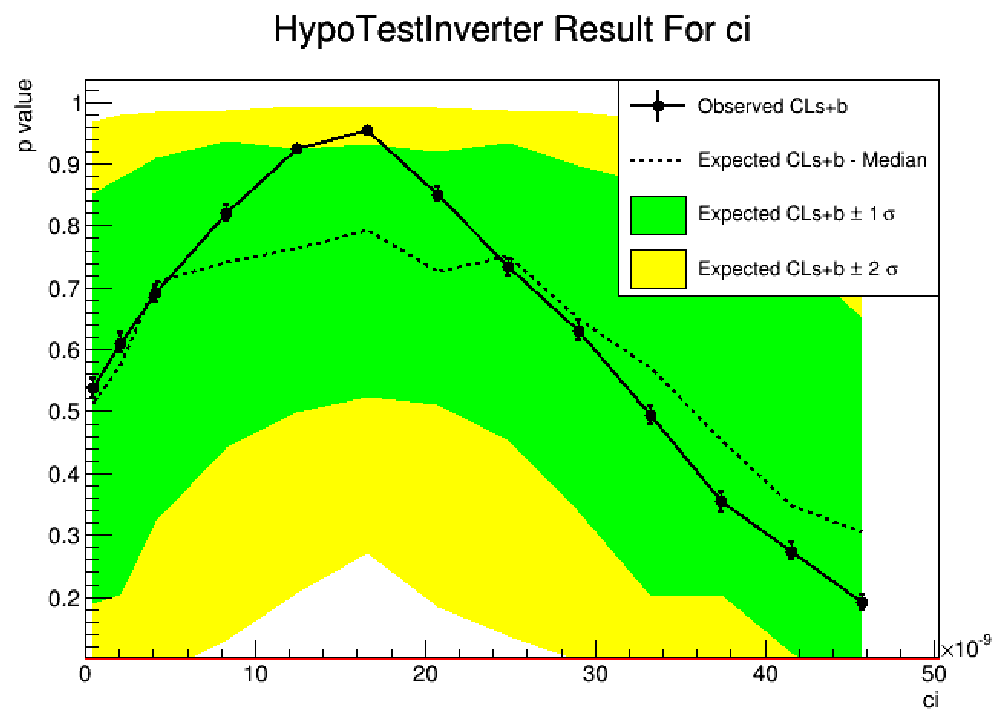
Calculating the Confidence Interval (Limit)

- The confidence interval for each mass is calculated independently.
- We do what is referred to as *hypothesis test inversion*:
  - Null hypothesis (H_0): WIMPs exist with the specific POI $\mu = \mu_{test}$
  - Alternative hypothesis (H_1): WIMPS exist (or don't) with some other POI $\mu \neq \mu_{test}$
- At each mass we run a series of hypothesis tests with different fixed values of $\mu_{test.}$
  - We carry out the same test procedure as before (Test Statistic: $\lambda = \frac{\mathcal{L}(\mu = \mu_{test};\, \hat{\hat{\theta}})}{\mathcal{L}(\mu \neq \mu_{test};\hat{\theta})}$, $q = -2\ln\lambda$) except that instead of simply comparing our data's test statistic to an acceptance region to determine whether to accept or reject it, we calculate its p-value (the probability that the null hypothesis yields something more outlandish).
- When we determine the value of μ_{test} that yields a p-value of 0.1, we record that value of μ_{test} and call it our limit. (this makes no sense to me)
  - Recall that a p-value of 0.1 means that 90% of the time with this hypothesis we would have gotten a result that better matches the null hypothesis (WIMPs with μ_{test}).

Limit Setting for One Mass Illustration

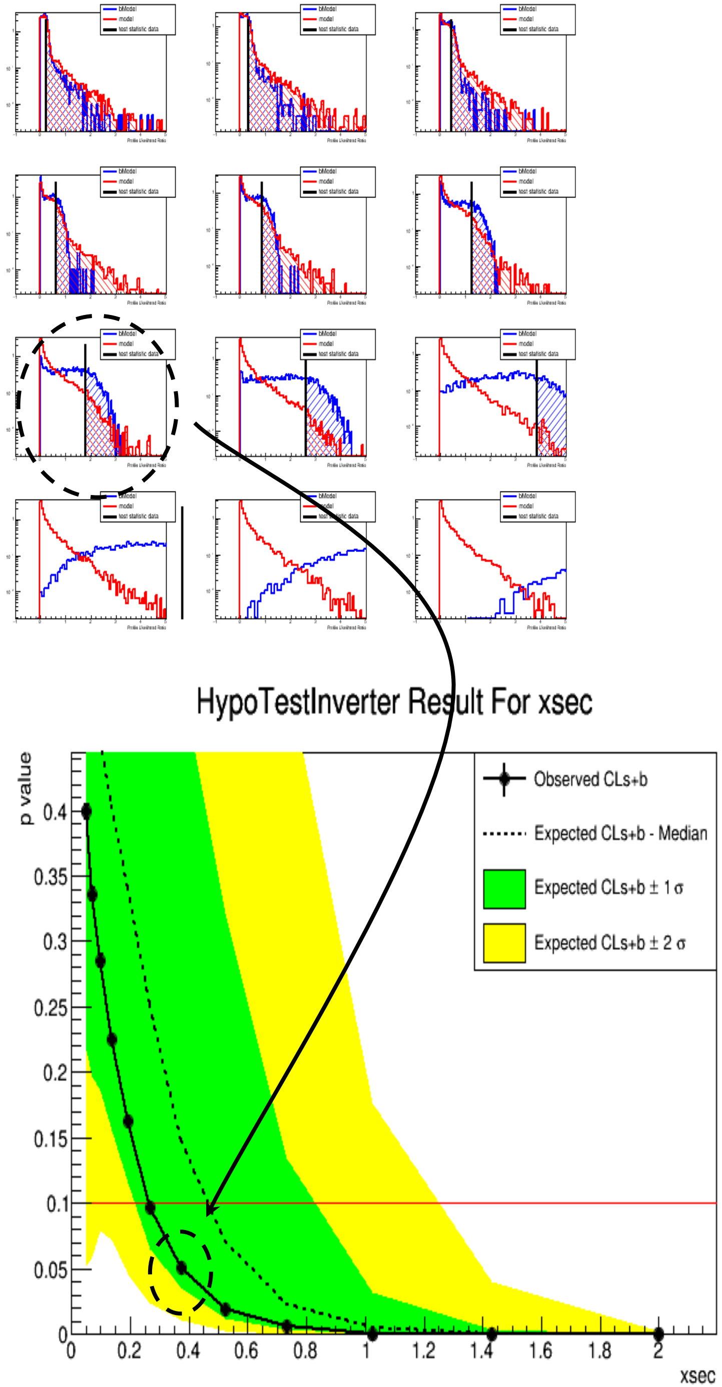


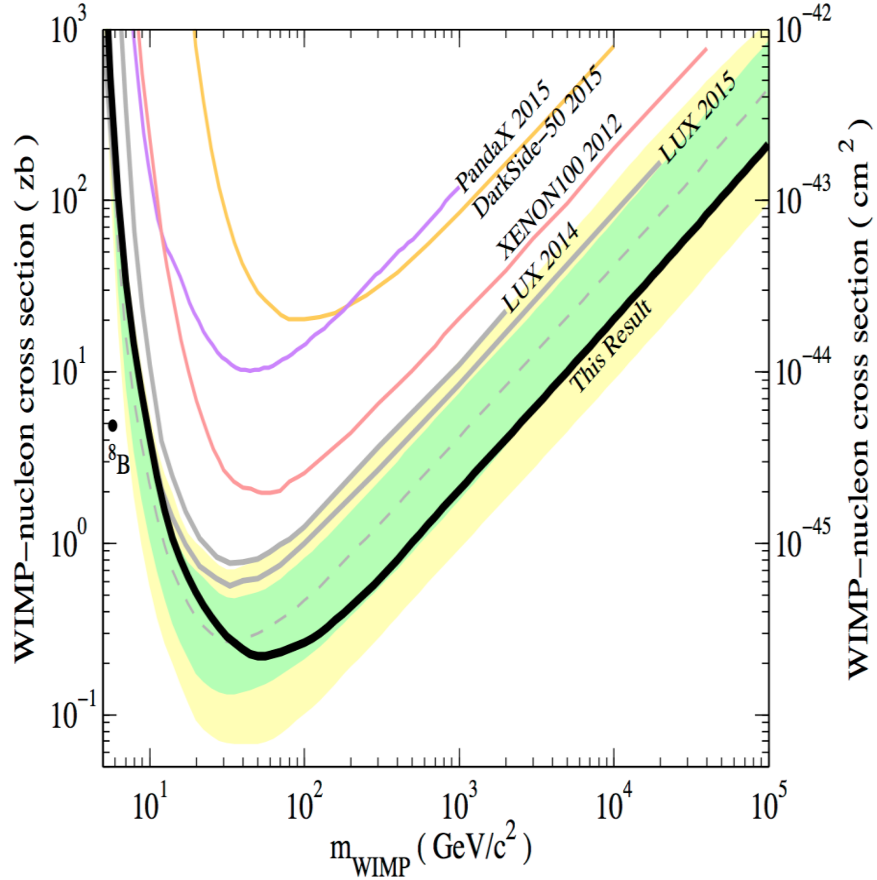
Hypothesis tests with varying c_1^0 (eft couplnig constant)



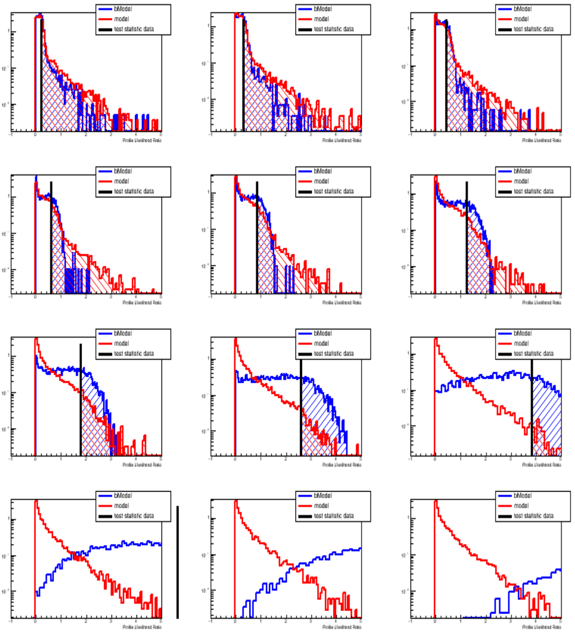
Looking for p-value of 0.1
(off the right side here, didn't go far enough)

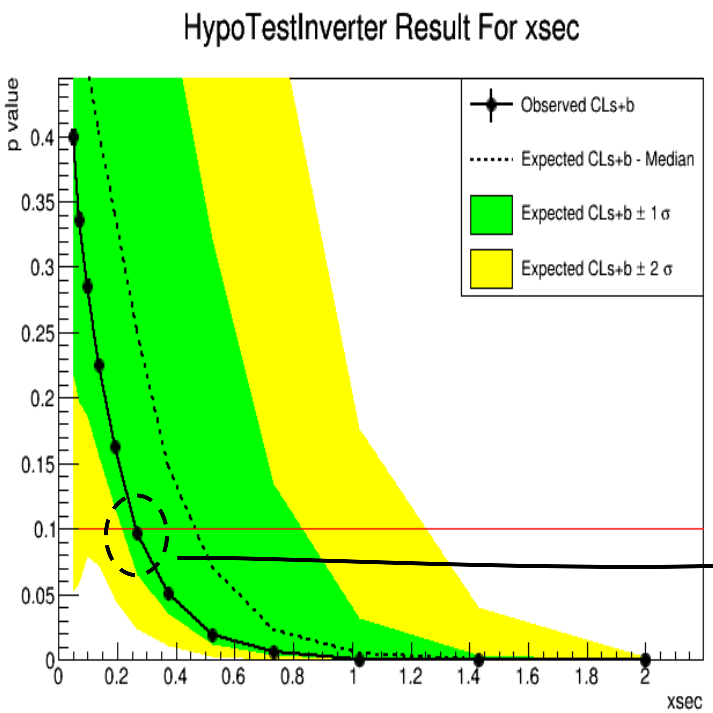
Limit Setting for One Mass Illustration

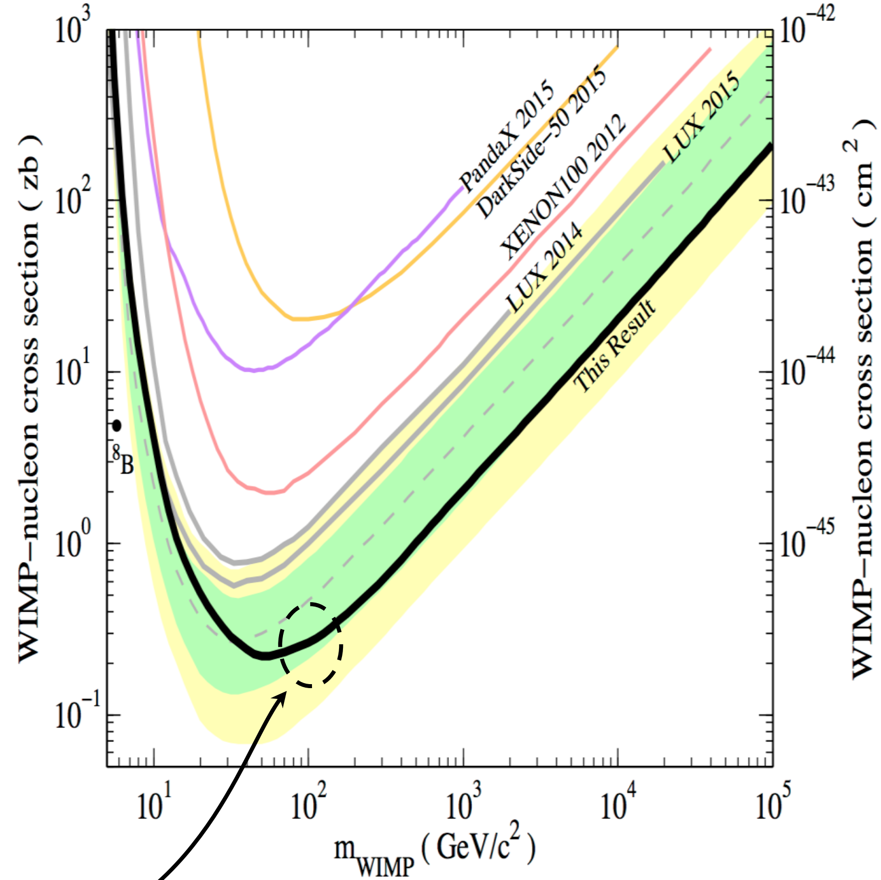




Limit Setting for One Mass Illustration







The LUX PLR: Misc Notes

- The run4 analysis is done in 16 different bins I have called "subDetectors."
  - 4 "Time Bins" and 4 "Z-Slices."
- The run4 analysis is done in "S2 space." That is, our spatial dimensions are S2r, S2phi, and drift (drift time). S2r and S2phi refer to the position where the detector measures that the electrons freed by an event emerge from the liquid.
- The PDF referred to in all of the introductory material is the PDF of the test statistic (q), there are many other PDFs that are about to be referred to, try not to get them confused as there are many layers of PDFs ☺.

The LUX PLR: Likelihood

- $\mathcal{L}(\mu;\ \theta) = \prod_{d \in subDetectors} \left[\underbrace{\mathrm{Pois}(n_{obs,d};\ n_{exp,d})}_{\text{Number of events}} \underbrace{\prod_{i=1}^{n_{obs,d}} \mathrm{PDF}_d(\vec{x_i};\mu)}_{\text{Shape information}} \right]$

  $\mathrm{X} \underbrace{\prod_{p \in nuisance} \mathrm{Gaus}(n_{exp,p}; \overline{n_{exp,p}}, \sigma_{exp,p})}_{\text{Profiling constraints}}$

  - $\mathrm{PDF_d}(\vec{x_i};\ \mu) = n_{exp,d,sig} f_{d,sig}(\vec{x_i}) + \sum_{p \in background} r_{d,p} n_{exp,p} f_{d,p}(\vec{x_i})$
  - $n_{exp,d} = r_{d,sig} t\mu + \sum_{p \in background} r_{d,p} n_{exp,p}$ where $r_{d,p}$ is the expected fraction of backgrounds (or signal) of type p in subDetector d, and t is a constant relating μ to the number of signal events.
  - fs are PDFs of the data-space (see below) specific to each subDetector and background type

  - Measured: n_{obs}, $\vec{x_i} = (S1, S2, r, drift, \phi, subDetector)$
  - Nuisance: $n_{exp,p}$

- Essentially: linear combination of signal PDF and background PDFs. Vary the overall scale factors. Penalize for varying away from our expected values.

The LUX PLR: Implementation

- Create a 5D PDF of the shape information
  - Weighted Sum of signal PDF and background PDFs
    - Current backgrounds: Boron 8 neutrinos, accidentals, Ar37, RnKr, comptonBottom, comptonRest, Kr83m, wall, (plan to add at least (alpha, n), and gamma-x).
- Create a PDF for the profile of each nuisance parameter
  - Just a Gaussian for the total expected number of each of the backgrounds mentioned above currently. Tested the effect of using G2 and a lindhart factor modifier as nuisance parameters but their effect was found to be negligible (at least at low energies).
- Take the direct product of the above 5D PDF and each of constraint Gaussians.
  - The result is a 5 + $n_{nuisance}$ dimensional PDF whose probability when evaluated on the data set is exactly the product of the red and green sections from the previous slide.
- Feed the resulting PDF and the data into HypoTestInverterDemo (root PLR function) along with options governing other parameters (number of toy mcs to run per (mass, POI), what POIs to use, etc)
- The code for this is in the EFTRun4.cxx file in the EFT limits branch of the LUX PLR code.

The LUX PLR: Signal Model

EFT Specific, but generalisable!

The Signal PDF

- Direct product of two independent PDFs for each subDetector
  - A 3D PDF uniform in (r, ϕ, z) transformed into (S1r, S2ϕ, drift)
  - A 2D PDF in (S1, S2) determined uniquely for each WIMP mass-operator-nucleon-subDetector combination
- Individual (S1, S2) PDFs saved root histograms 4 Z-Slices (drift bins) to a file, but 1 file per Wimp mass, operator, nucleon, time bin combination

- Implementation in EFTRun4/RooSignalPDF.cxx (.h)

Recoil Energy Spectrum

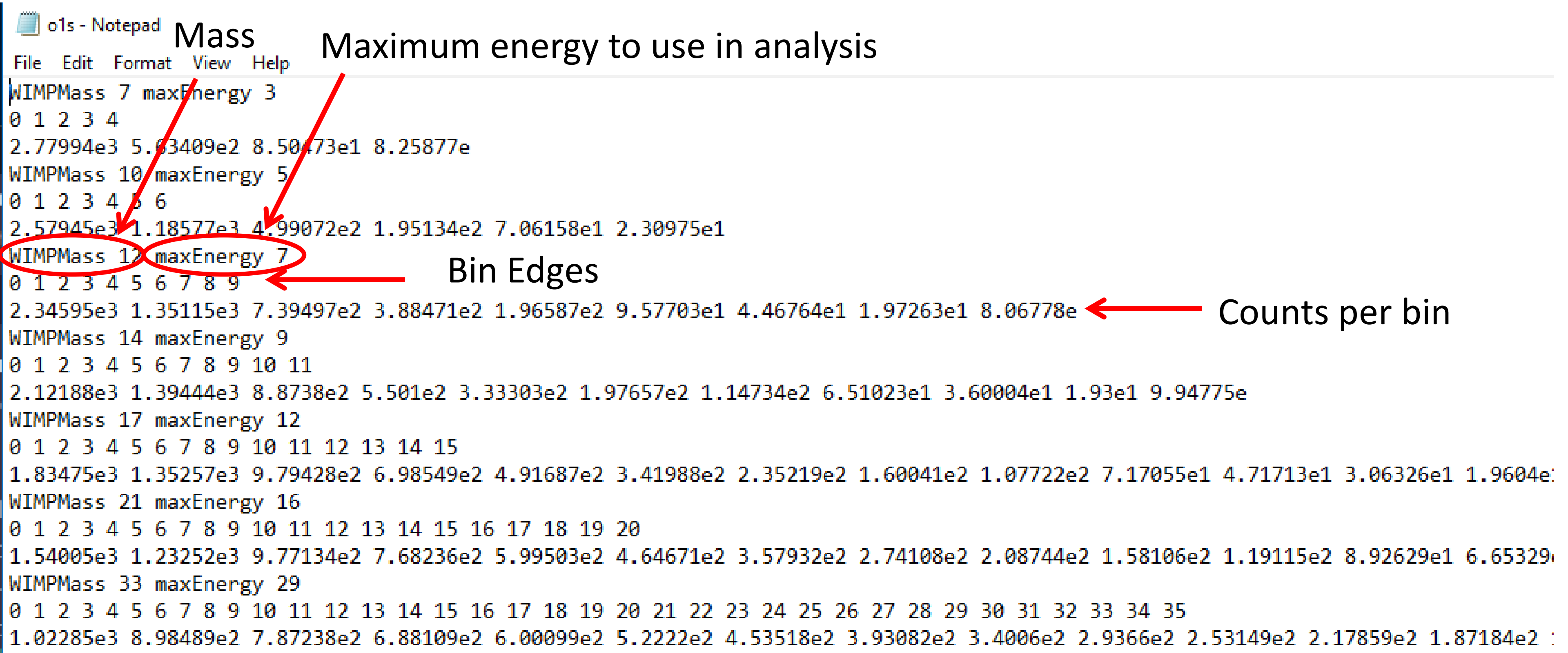
- Generated by “DMFormFactor” mathematica package
- Generate an analytic energy recoil spectrum for each isotope of Xenon (isotopes = {128, 129, 130, 131, 132, 134, 136}) as well as each of the following masses:
  - masses = {7,10,12,14,17,21,33,50,100,400,1000,4000}
- Weight the analytic spectra by their isotopic abundance and sum them together for each mass. This leaves one spectrum for each mass

Recoil Energy Spectrum

- These analytic spectra are insanely complicated and difficult (as well as unnecessary) to port to the limits code, so we bin it.
  - Spectrum is integrated (numerically) out to 350 keV (beyond our max energy range for this analysis) to obtain an energy cutoff.
  - Spectrum is then integrated in 1 keV bins from 0 until the total integrated reaches 99% of the total from 0 to 350 keV. This number is recoreded and the integration then proceeds to a factor of 1.2 higher in recoil energy. (This is because the standard deviation in the s1 and s2 distributions for a single energy is typically about 6 or 7%)
  - These bins and the values of the integrals are then stored and written to file.

Recoil Energy Spectrum

- Example text output (one for each coupling)

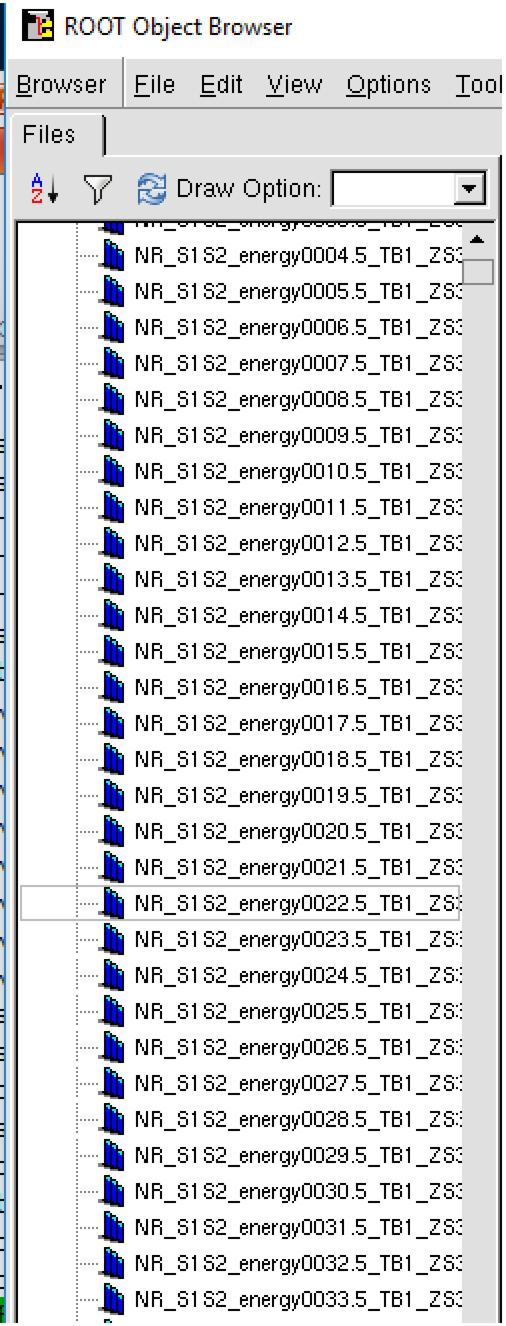


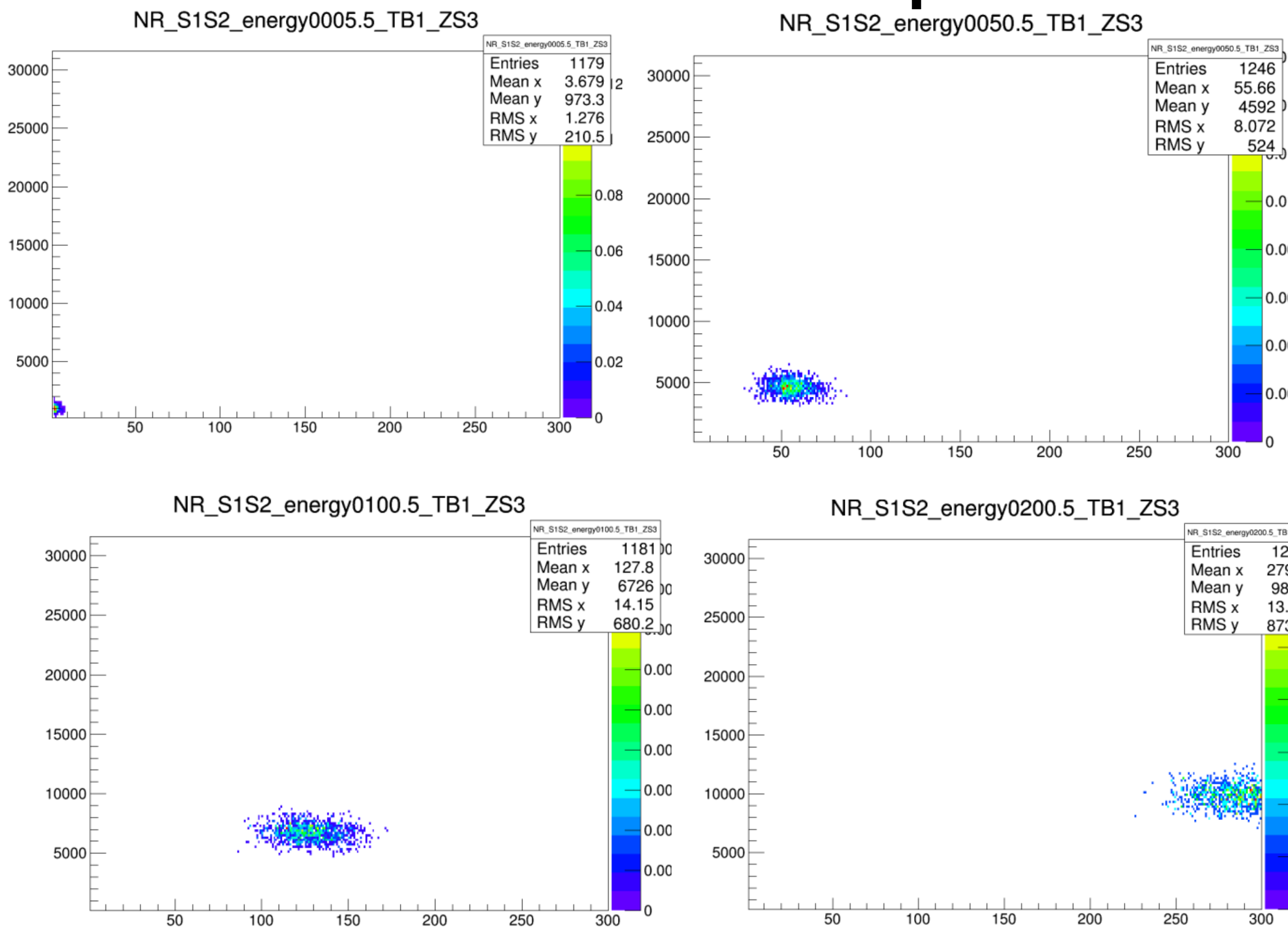

```
WIMPMass 7 maxEnergy 3
0 1 2 3 4
2.77994e3 5.63409e2 8.50473e1 8.25877e
WIMPMass 10 maxEnergy 5
0 1 2 3 4 5 6
2.57945e3 1.18577e3 4.99072e2 1.95134e2 7.06158e1 2.30975e1
WIMPMass 12 maxEnergy 7
0 1 2 3 4 5 6 7 8 9
2.34595e3 1.35115e3 7.39497e2 3.88471e2 1.96587e2 9.57703e1 4.46764e1 1.97263e1 8.06778e
WIMPMass 14 maxEnergy 9
0 1 2 3 4 5 6 7 8 9 10 11
2.12188e3 1.39444e3 8.8738e2 5.501e2 3.33303e2 1.97657e2 1.14734e2 6.51023e1 3.60004e1 1.93e1 9.94775e
WIMPMass 17 maxEnergy 12
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1.83475e3 1.35257e3 9.79428e2 6.98549e2 4.91687e2 3.41988e2 2.35219e2 1.60041e2 1.07722e2 7.17055e1 4.71713e1 3.06326e1 1.9604e
WIMPMass 21 maxEnergy 16
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1.54005e3 1.23252e3 9.77134e2 7.68236e2 5.99503e2 4.64671e2 3.57932e2 2.74108e2 2.08744e2 1.58106e2 1.19115e2 8.92629e1 6.65329
WIMPMass 33 maxEnergy 29
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
1.02285e3 8.98489e2 7.87238e2 6.88109e2 6.00099e2 5.2222e2 4.53518e2 3.93082e2 3.4006e2 2.9366e2 2.53149e2 2.17859e2 1.87184e2
```

Nuclear Recoil Templates

- Locations are generated randomly (uniformly) in real space.
- Locations are mapped to S2 space using JPM (electric field at the point simultaneously obtained).
- Fiducial cut applied, event is re-generated if the location does not pass.
- NEST is used to generate an s1c and s2c for the event.
- Events are sorted into z-slices based on drift time
- Events generated until each z-slice has at least 10,000 events
- This is done for each energy 0.5 keV to 350 keV in 1 keV increments for each time bin

Nuclear Recoil Templates

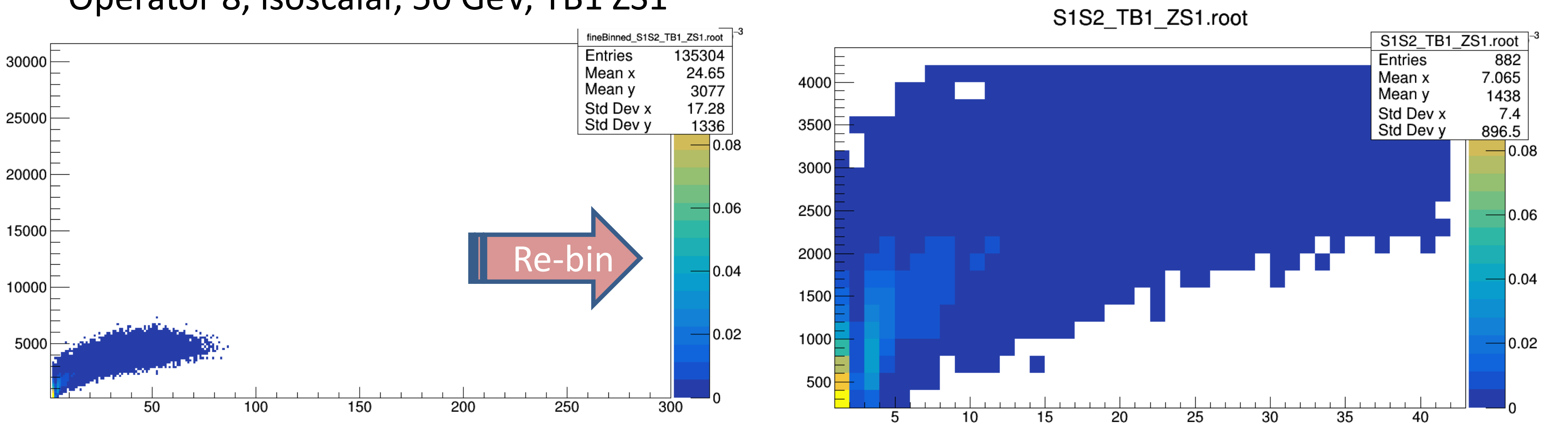




Nuclear Recoil Models

- Take the Templates and weight each energy and each time bin and z-slice and weight them by the appropriate number from the Recoil Energy Spectrum and add them together.
- Cut the S1 and S2 off at the mean values associated with the "maxEnergy" number, but add ones from higher energy to account for bleed-in.
- Re-binned into an appropriate number of bins for the PLR code to run with

Operator 8, isoscalar, 50 GeV, TB1 ZS1



The LUX PLR: Generic Background Model

- Most Backgrounds have a weak correlation between their spatial and energy (S1, S2) distributions.
- Like the signal model, the generic background model is a direct product of two independent PDFs for each subDetector
  - A 3D PDF in (S1r, S2ϕ, drift) determined uniquely for each background type (and subDetector)
  - A 2D PDF in (S1, S2) determined uniquely for each background type and subDetector
- The nuisance parameter dictating the number of expected events of each background type scales each model (in addition to the fraction of these expected in each subDetector which is fixed, and is not a nuisance parameter).
- Saved in the same manner as the signal PDFs except that there is also a file with the 3D spatial PDFs and each 2D PDF is normalized such that the full integral is the total number of expected events for that subDetector.

- This is implemented in EFTRun4/RooBkgPdf.cxx (.h) (notice here the df in Pdf is lower-case, opposite of the Signal model. Silly, I know, but I had two versions in development at once and this one worked and my brain works in funny ways and I never overwrote the original one)

The LUX PLR: Generic Background Model

- Others are in charge of delivering these:
  - Most: Wei or someone using 
  - Kr83m: Alice
  - (α, n): Me ☹
  - γ-x: Peter Rossiter (not modeled yet, don't know if we can make the uncorrelated model)

The LUX PLR: Wall Model

- Events (usually Rn daughters) on the walls have much of their S2 absorbed by the wall and so can't be included in the other background models.
- Unfortunately both S2r and drift correlate with S2 so our generic background model won't work for this model.
- Instead, we currently have two proposed formats for the model
  - 3D PDF(S1, S2, drift) direct product 1D PDF(S2ϕ) multiplied by the a function of S2r f(S2r; S1, S2, drift, S2ϕ) which is itself a functional of the other parameters as indicated
  - 4D PDF(S1, S2 , S2ϕ, drift) multiplied by f(S2r; S1, S2, drift, S2ϕ) as described above
    - The 4D PDF is slightly harder to implement and harder on computation time

- In charge of model: Claudio

- Not implemented yet (format not quite decided, waiting on whether correlation between S2ϕ and other variables is low enough to use option 1)

The LUX PLR: Bells and Whistles (New to EFT)

- The PLR can run a hypothesis test for a single specified value of the POI.
- Signal models easy to create (as long as features larger scale than 1 keV).
- Automatic S1 and S2 cutoff based on input signal recoil energy spectrum.
- Most PDFs have a set number of bins (configurable in Parameters.h), but S1 and S2 binning is determined dynamically based on the recoil spectrum.
- Automatically toggles between using Kr83m exclusion periods (smaller exposure time) and a Kr83m background model and using the longer exposure time and excluding the Kr83m background based on if the recoil spectrum approaches Kr83m energies.

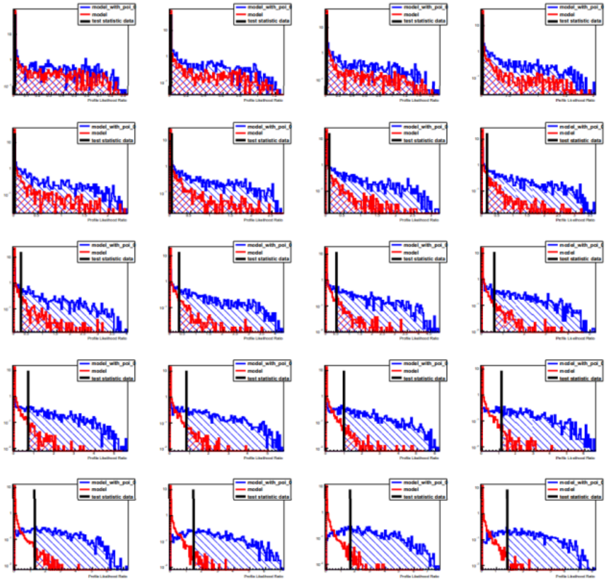
Code

- Available at https://github.com/luxdarkmatter/LUXLimits/blob/EFT/EFTRun4/ if you want to poke around and look at it.

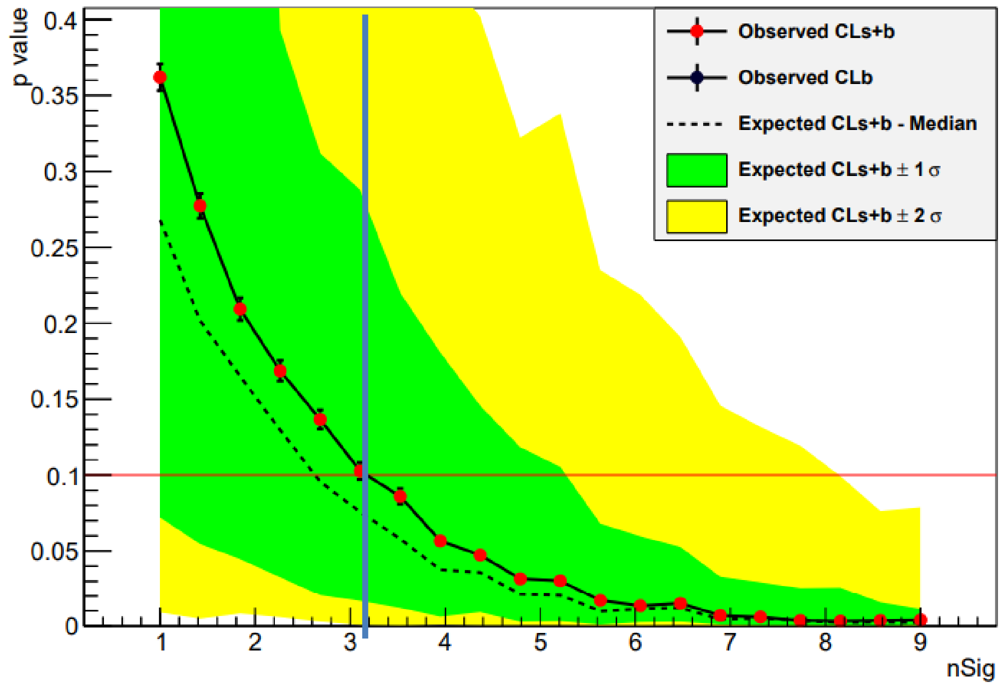
Thanks!

- With remaining time I could:
  1. Answer questions
  2. Walk through the code showing what is done where.
  3. Look at an example of a PLR for a simple counting experiment that I coded from scratch to get a better idea of what a PLR actually is.
  4. Cry because of all of your cruel criticism.

Limit Setting for One Mass Illustration



Different run than before



Intersects at ~3.2 signal events so our limit is the value of POI that yields an expected 3.2 signal events