NA62 Statistics Forum
Introduction

Roberta Volpe

(Comenius University, Bratislava)

8 December 2020

Introduction

- Meetings every other week
- In general there will be:
  - an analysis to be discussed
  - discussions about the software, possibly related to the the "analysis of the day"

- Today meeting:
  - Yesterday evening Evgueni proposed to discuss (mainly to answer my questions on) K->μ + inv, I had no time to prepare a specific code example on that
  - In the following slides I'll give some generalities and clarification about the code already available on gitlab

Possible analyses

Interest, but not immediate commitment (because the analysis, or the analysts, have other priorities now)

- **ALP-> 2 photons (Babette et al.)**
- **K-> ενννν (Artur)**
- **K-> μν X (X->γγ) (Andrea e Francesca)**
- **K-> πX (X->μμ) X long and short lived (Slava and Lubos ?)**
- **K->μν X (X inv), K->μνννν, μ X (X inv) (?) (Evgueni, something today)**

Software

Yesterday Evgueni asked for something more user friendly

- **Upper limit chain:**

.root file **.root file**

1) Function which creates the model

2) Function which performs the upper limit

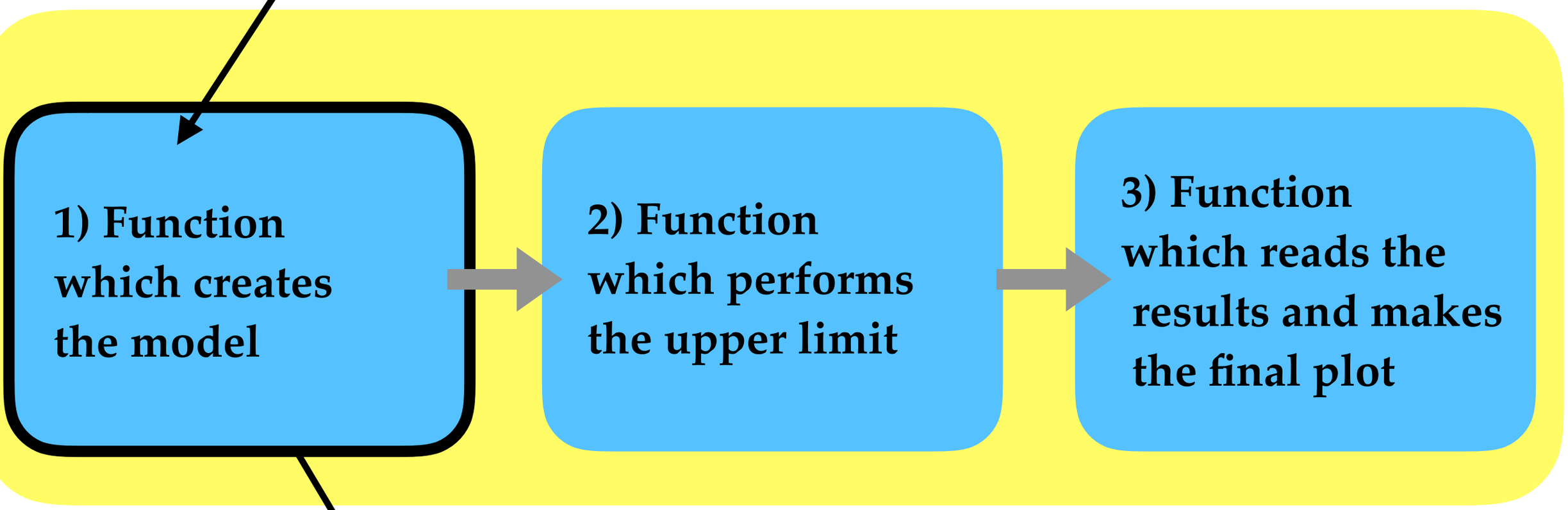
3) Function which reads the results and makes the final plot

Can be a single macro, or three different macros,
Both the options are in "statstutorials"
https://gitlab.cern.ch/volper/statstutorials/-/tree/master/

The full chain, for the model in the example, is very user friendly

Software

If you want to change the model, to adapt it to your analysis you need to modify the step 1)



This is heavily analysis-dependent, and these meetings are useful to discuss how to implement this step in several situations.

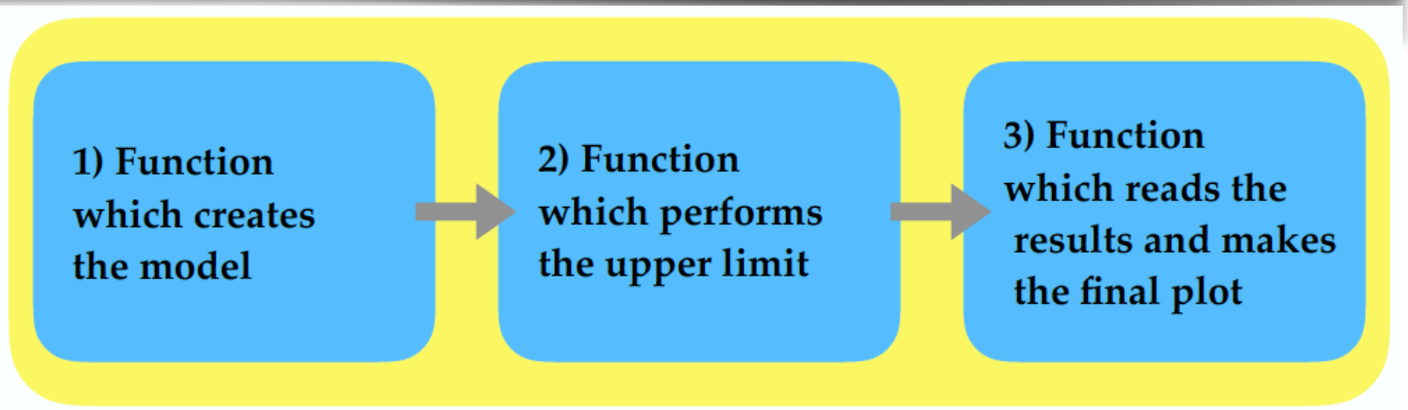
Software

For example, the step 1) cannot be the same function for LNV and HNL searches.

Step 1) is part of the analysis, and it should be implemented by the analyst.
I created the example for your convenience so that you have something to start with (even if it has most of the issues we face in our NA62 analyses, so it is almost ready to use).

If you create a model in a proper way, the **steps 2) and 3) should be the same for every analysis** (now I'm talking about upper limits, but I'll provide other code) **and you don't need to understand all the content of the functions**, as long as the inputs (created with step 1) are appropriate.
In this sense step 2 and 3 are very user friendly

Full chain in one macro



PlotULFrequentistPeakSearch.C

gitlab.cern.ch/volper/statstutorials/-/blob/master/PlotULFreque

GitLab Projects Groups Snippets Help

S Statstutorials

Project overview

Repository

Files

Commits

Branches

Tags

Contributors

Graph

Compare

Issues 0

Merge Requests 0

CI / CD

Operations

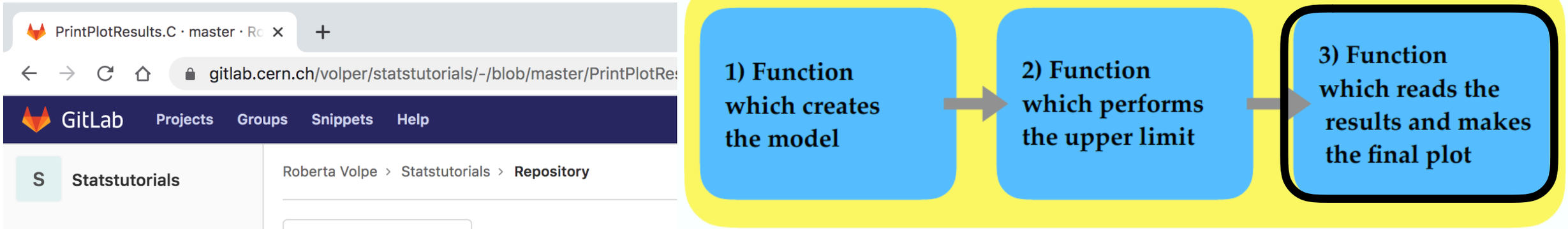
Packages & Registries

Collapse sidebar

PlotULFrequentistPeakSearch.C 32.1 KB

```
1  //=============================================================================================
2  // Example macro for a peak search (with a "small" sample)
3  //---------------------------------------------------------------------------------------------
4  // set an upper limit to a signal strenght as a function of the mass hypothesis
5  // using the frequentist calculator and Profile likelihood as test statistic
6  //=============================================================================================
7  // 1) Create the model and save it in a RooWorkspace                          Step 1
8  // 2) Use the Rooworkspace to set an upper limit on the signal strenght for each mass hypothesis   Step 2
9  // 3) Read the output file wich contains HypoTestInverterResult
10 // 4) print it in a text file
11 // 5) plot it as a function of the mass hypothesis (Brazilian plot style)   Step 3
12 //---------------------------------------------------------------------------
13 // Model created in wswrite():
14 // - Signal shape is a gaussian
15 // - Background shape is a 2nd order polynomial
16 // - Background yield modeled in the on/off approach
17 // ---------------------------------------------------------------------------
18 //  Upper limit computation in HypoTestInv, RunInverter, AnalyzeResult
19 //  adapted from RooStats tutorial StandardHypoTestInvDemo.C
20 //---------------------------------------------------------------------------
21 // [if you are on lxplus:
22 // source /cvmfs/sft.cern.ch/lcg/app/releases/ROOT/6.20.00/x86_64-centos7-gcc48-opt/bin/thisroot.sh]
23 // root -l
24 // root [0] PlotULFrequentistPeakSearch.C+
25 // root [1] ULFrequentistPeakSearch()
26 // root [2] PrintResults()
27 // root [3] PlotResults()
28 //=============================================================================================
29 // Author: Roberta Volpe
30 //=============================================================================================
31
32 #include "RooExtendPdf.h"
```

Step 3 added last week



PrintPlotResults.C · master · Rc

gitlab.cern.ch/volper/statstutorials/-/blob/master/PrintPlotRes

GitLab Projects Groups Snippets Help

S Statstutorials

Project overview

Repository

Files

Commits

Branches

Tags

Contributors

Graph

Compare

Issues 0

Merge Requests 0

CI / CD

Operations

Packages & Registries

Collapse sidebar

Roberta Volpe › Statstutorials › Repository

master statstutorials / **PrintPlotResults.C** Find file Blame History Permalink

adding a macro to read the upper limit results and plot them as a function of the mass hypothesis
Roberta Volpe authored 6 days ago 1092a513

PrintPlotResults.C 6.82 KB Edit Web IDE

```
1  //============================================================================
2  // Macro to read and plot the results from ULFrequentistPeakSearch.C
3  //============================================================================
4  // 1) Read output file wich contains HypoTestInverterResult,
5  // 2) print it in a text file and
6  // 3) plot it as a function of the mass hypothesis (Brazilian plot style)
7  //----------------------------------------------------------------------------
8  // (source /cvmfs/sft.cern.ch/lcg/app/releases/ROOT/6.20.00/x86_64-centos7-gcc48-opt/bin/thisroot.sh)
9  // root -l
10 // root [0] .L PrintPlotResults.C+
11 // root [1] PrintResults()
12 // root [2] PlotResults()
13 //----------------------------------------------------------------------------
14 // Author: Roberta Volpe roberta.volpe@cern.ch
15 //============================================================================
16 #include "TROOT.h"
17 #include "TStyle.h"
18 #include "TSystem.h"
19 #include "TCanvas.h"
20 #include "TPad.h"
21 #include "TLegend.h"
22 #include "TAxis.h"
```

Code example of step 1

Stats forum of 10 November:

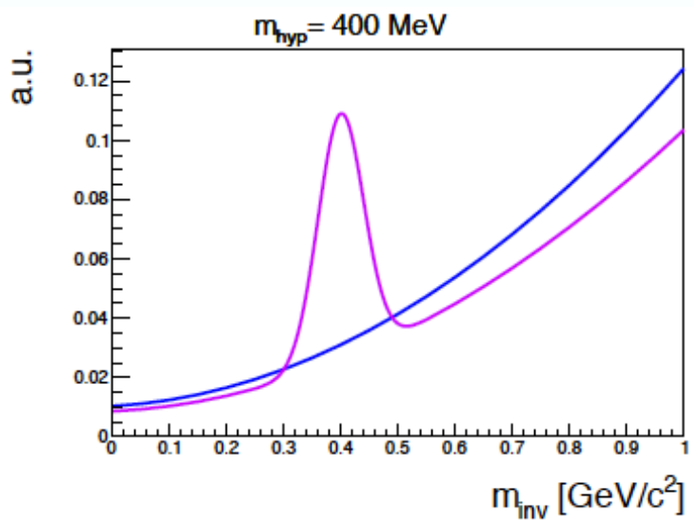
https://indico.cern.ch/event/973526/

https://indico.cern.ch/event/973526/contributions/4103437/attachments/2140218/3605937/StatSW_10Nov20.pdf

I explained in detail the code 1 month ago,
if you have questions you can ask also today, I'll take the old slides

The example will deal with setting an upper limit using the
- Profile Likelihood as test statistic,
- fully frequentist approach,
- shape analysis

We will see concepts which are useful also to other situations, but we will focus on this example to fix the ideas



Model for the distribution of background and signal in the observable (usually a discriminant variable), or shape:

- Gaussian signal
- Polynomial bkg

2 fake "observed" datasets considered:

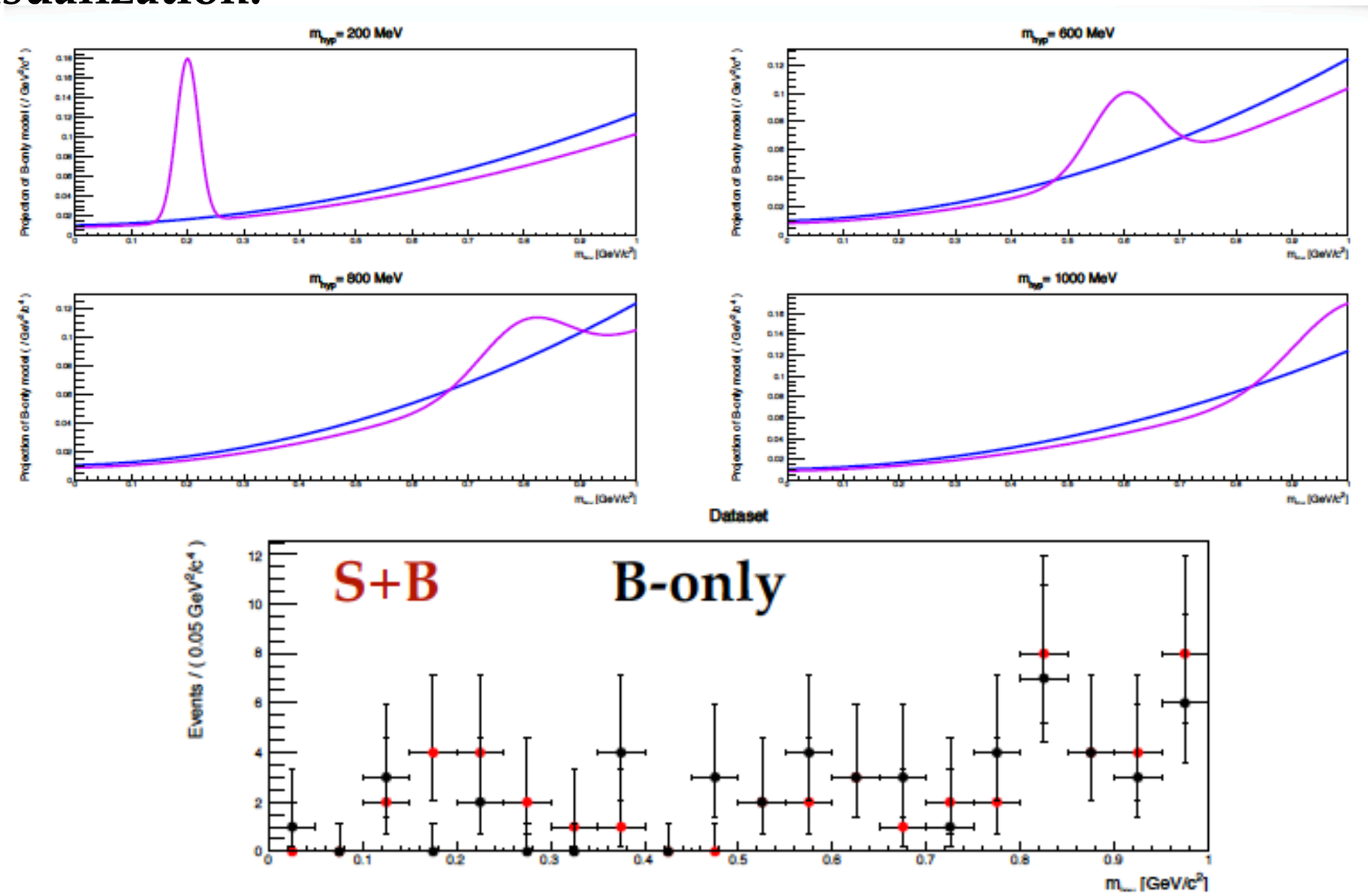
- **Generated from the B-only hypothesis**
- **Generated from the B+S (at 200 MeV) hypothesis**

10 Mass hypotheses considered,
but with a change in the loop you can have how many you want

Output of step 1)

I explained in detail the code 1 month ago,
if you have questions you can ask also today, I'll take the old slides

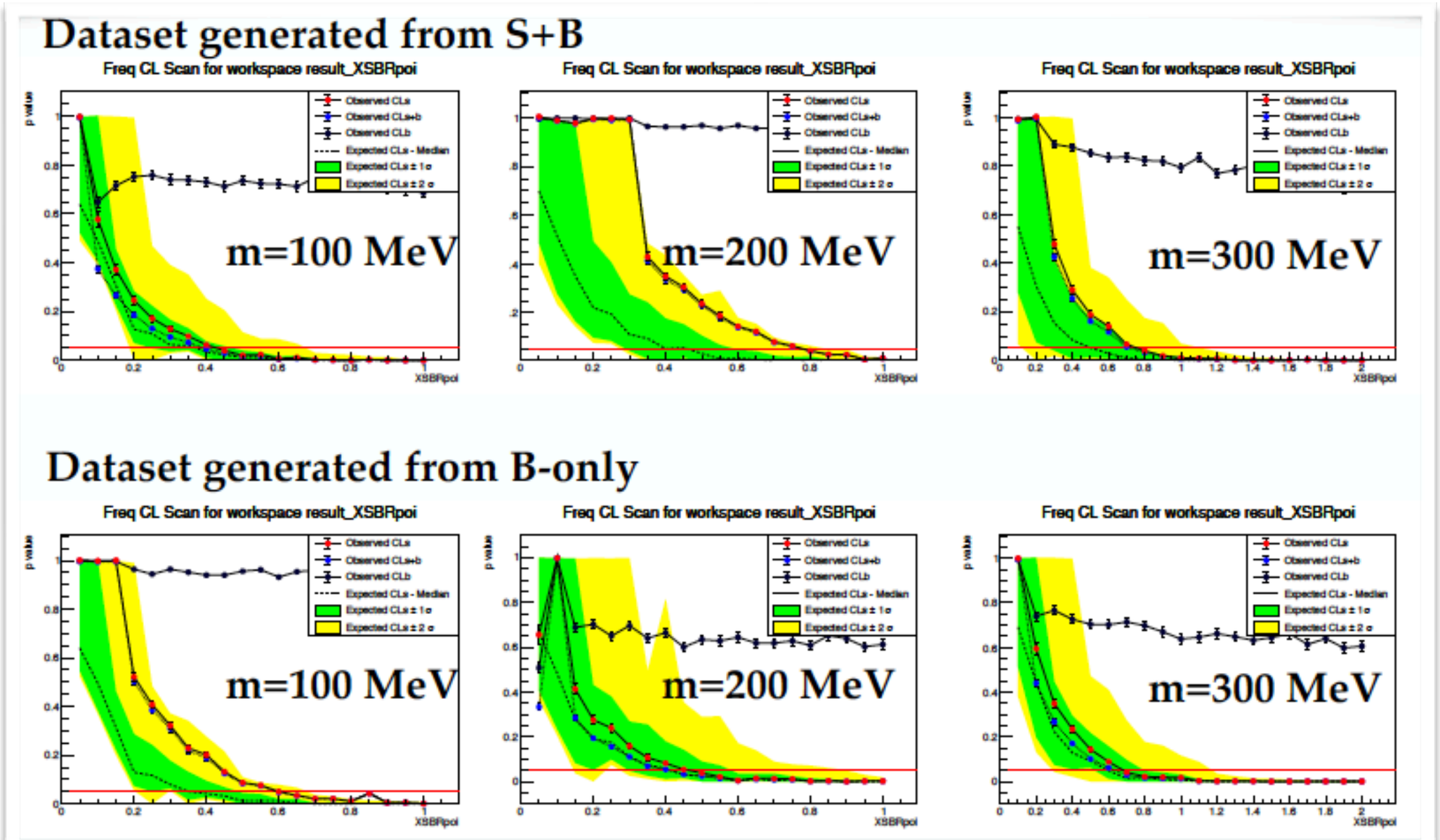
Pdf for visualization:



<u>And a .root file with all the information about the model, for each mass hypothesis</u>

Output of step 2)

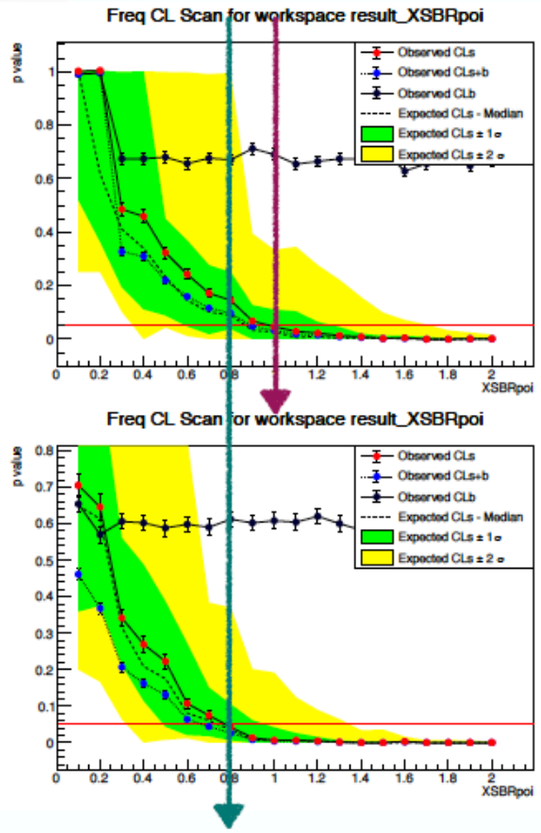
Examples:



Output of step 2)

Examples with and without systematics:

Systematics



example:
Mass hypothesis: 500 MeV
Dataset only Bkg

With Systematics

Without Systematics
UL on Signal strength
~ 20% better

Roberta Volpe STATS November 2020 25

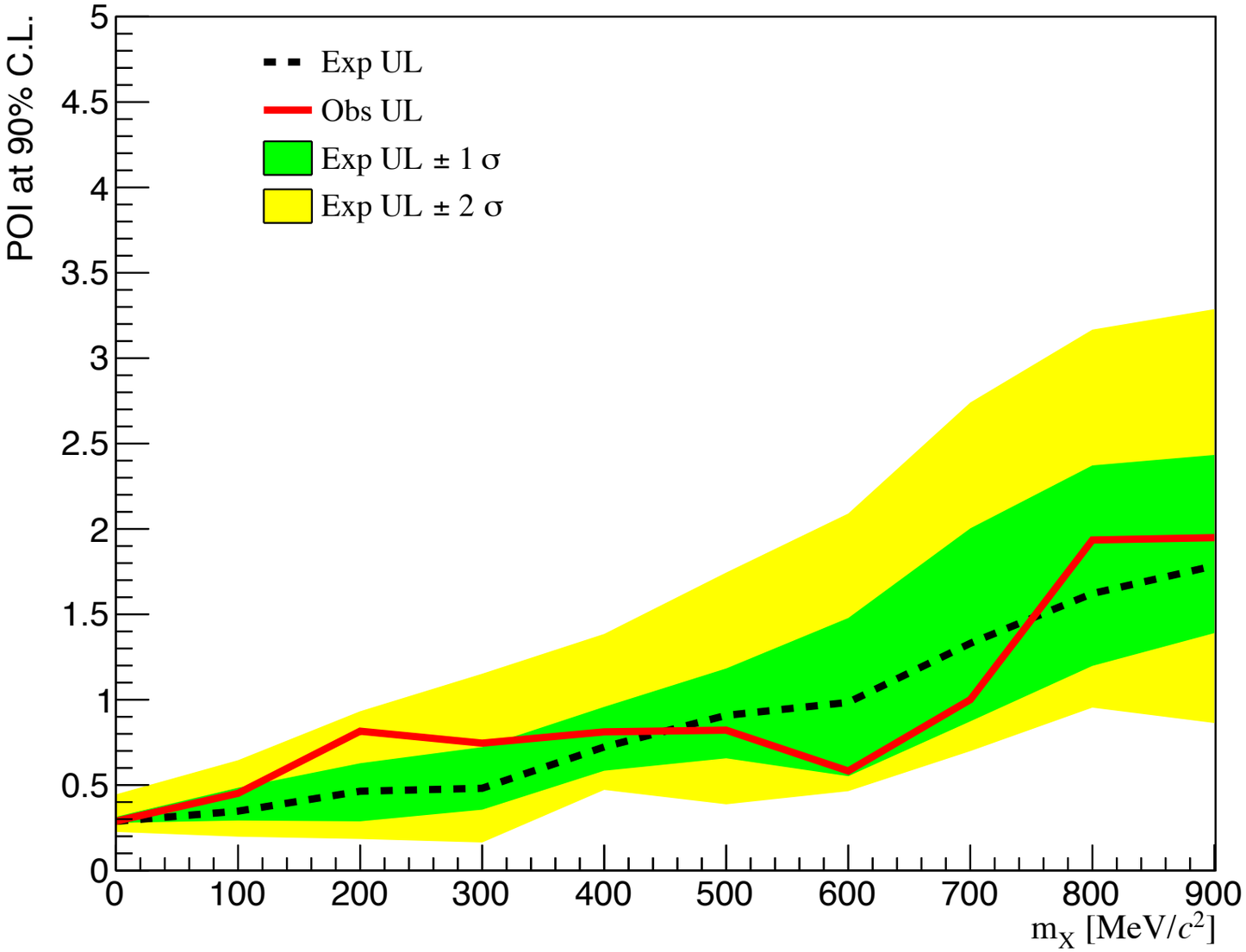
```
-bash-4.2$ ls
freqPLL_Fake_m200_m0.C            freqPLL_Fake_m200_m200.root     freqPLL_Fake_m200_m600.pdf
freqPLL_Fake_m200_m0_log.C        freqPLL_Fake_m200_m300.C        freqPLL_Fake_m200_m600.root
freqPLL_Fake_m200_m0_log.pdf      freqPLL_Fake_m200_m300_log.C    freqPLL_Fake_m200_m700.C
freqPLL_Fake_m200_m0.pdf          freqPLL_Fake_m200_m300_log.pdf  freqPLL_Fake_m200_m700_log.C
freqPLL_Fake_m200_m0.root         freqPLL_Fake_m200_m300.pdf      freqPLL_Fake_m200_m700_log.pdf
freqPLL_Fake_m200_m1000.C         freqPLL_Fake_m200_m300.root     freqPLL_Fake_m200_m700.pdf
freqPLL_Fake_m200_m1000_log.C     freqPLL_Fake_m200_m400.C        freqPLL_Fake_m200_m700.root
freqPLL_Fake_m200_m1000_log.pdf   freqPLL_Fake_m200_m400_log.C    freqPLL_Fake_m200_m800.C
freqPLL_Fake_m200_m1000.pdf       freqPLL_Fake_m200_m400_log.pdf  freqPLL_Fake_m200_m800_log.C
freqPLL_Fake_m200_m1000.root      freqPLL_Fake_m200_m400.pdf      freqPLL_Fake_m200_m800_log.pdf
freqPLL_Fake_m200_m100.C          freqPLL_Fake_m200_m400.root     freqPLL_Fake_m200_m800.pdf
freqPLL_Fake_m200_m100_log.C      freqPLL_Fake_m200_m500.C        freqPLL_Fake_m200_m800.root
freqPLL_Fake_m200_m100_log.pdf    freqPLL_Fake_m200_m500_log.C    freqPLL_Fake_m200_m900.C
freqPLL_Fake_m200_m100.pdf        freqPLL_Fake_m200_m500_log.pdf  freqPLL_Fake_m200_m900_log.C
freqPLL_Fake_m200_m100.root       freqPLL_Fake_m200_m500.pdf      freqPLL_Fake_m200_m900_log.pdf
freqPLL_Fake_m200_m200.C          freqPLL_Fake_m200_m500.root     freqPLL_Fake_m200_m900.pdf
freqPLL_Fake_m200_m200_log.C      freqPLL_Fake_m200_m600.C        freqPLL_Fake_m200_m900.root
freqPLL_Fake_m200_m200_log.pdf    freqPLL_Fake_m200_m600_log.C
freqPLL_Fake_m200_m200.pdf        freqPLL_Fake_m200_m600_log.pdf
```

For each mass hypothesis:

- **Pdf to visualize the result**
- **Root file with the results to be used instep 3**

Output of step 3)

Step 3 takes as input the output files .root (10 in this example) and produce the final "Brazilian" plot as a function of the mass hypothesis



Example with fake data generated with a signal at 200 MeV

Software

Some description in the last meetings:

https://indico.cern.ch/event/973526/

https://indico.cern.ch/event/973526/contributions/4103437/attachments/2140218/3605937/StatSW_10Nov20.pdf

- **Did anyone test the code?**
- **Did anyone try to modify the code?**
- **Do you have any question?**

K->μ + missing mass

Evgueni's analysis: K-> μN, μνX (X inv)

From NA62-20-09

- Roberta's shape analysis procedure is not able yet to account for correlated uncertainties among the mass bins. The dominant uncertainties in the $K_{\mu\nu X}$ case (due to the non-gaussian tail and the simulation of the LAV veto inefficiency) are highly correlated among the mass bins. The analysis procedure for this case is yet to be developed.

I don't understand this sentence,
if Evgueni agrees we could talk about this in one of the next meetings

Of course some work should be done to find the best model to describe this analysis, but I think it is not impossible.

This was my slide 2 weeks ago, Evgueni contacted me yesterday, I'll ask few questions and we can discuss now

Reminders

Mailing list: NA62-Stats@cern.ch

- **Meeting frequency: every other week (with possible changes)**
- **Meeting time: Tuesday 15:00**

Please subscribe to the mailing list to be up to date

Please send me suggestions about analyses to be discussed by Thursday before the meeting

I will update the code here:

https://gitlab.cern.ch/volper/statstutorials/-/tree/master/

And in NA62 GitLab

Coming soon!