



# Calculation of Exclusion Limits - ROOT & python -



23<sup>rd</sup> January 2018

**Adrian Perieanu**



# ROOT

```
[perieanus-MacBook-Pro% root -l  
[root [0] .q  
perieanus-MacBook-Pro% █
```

*start root*  
*quit root*

```
[perieanus-MacBook-Pro% root -l JourneyToWork.C  
root [0]  
Processing JourneyToWork.C...  
fga->Integral(-1s,1s) ██████████  
fga->Integral() ██████████  
fga->Integral() ██████████  
Info in <TCanvas::Print>: pdf file journeytowork.pdf has been created  
[root [1] .q  
perieanus-MacBook-Pro% █
```

*run a root macro*

```
void JourneyToWork(){  
    gStyle->SetOptStat(0);  
    TCanvas * c1 = new TCanvas("", "", 600, 600);  
    TH1F * h = new TH1F( "h_work", ";x;f(x)", 20, 7, 8.5);
```

*the easiest way is to give the same name  
also to your void function*

```
[perieanus-MacBook-Pro% root -l  
[root [0] TH1F *h  
[root [1] h->SetMarkerColor(
```

*if you want to find out the type of an input variable:*

*here you press*



```
void SetMarkerColor(Color_t tcolor = 1)  
[root [1] h->SetMarkerColor(█
```

*and details about the input parameters show up*

# python

```
[perieanus-MacBook-Pro% python
Python 2.7.10 (default, Feb  6 2017, 23:53:20)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.34)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> exit()
perieanus-MacBook-Pro% █
```

*start python*

*quit python*

```
[perieanus-MacBook-Pro% python binomial_distribution.py
perieanus-MacBook-Pro% █
```

*run a python macro*

```
import numpy as np
from scipy.stats import binom
from matplotlib import pyplot as plt
```

```
#-----
# Define the distribution parameters to be plotted
n_values = [20, 20, 40] # number of trials
b_values = [0.2, 0.4, 0.4] # probability of success
linestyles = ['-', '--', ':']
colours = ['black', 'blue', 'red']
x = np.arange(-1, 200)
```

*a python macro needs imports  
and to have a certain structure*

```
[perieanus-MacBook-Pro% python
Python 2.7.10 (default, Feb  6 2017, 23:53:20)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.34)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> help()
```

```
Welcome to Python 2.7! This is the online help utility.
```

```
If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://docs.python.org/2.7/tutorial/.
```

```
Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules. To quit this help utility and
return to the interpreter, just type "quit".
```

```
To get a list of available modules, keywords, or topics, type "modules",
"keywords", or "topics". Each module also comes with a one-line summary
of what it does; to list the modules whose summaries contain a given word
such as "spam", type "modules spam".
```

```
help> █
```

*type: help() to get help in python*

# python

```
[perieanus-MacBook-Pro% python
Python 2.7.10 (default, Feb  6 2017, 23:53:20)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.34)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> help()
```

*type: help() to get help in python*

```
Welcome to Python 2.7! This is the online help utility.
```

```
If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://docs.python.org/2.7/tutorial/.
```

```
Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules. To quit this help utility and
return to the interpreter, just type "quit".
```

```
To get a list of available modules, keywords, or topics, type "modules",
"keywords", or "topics". Each module also comes with a one-line summary
of what it does; to list the modules whose summaries contain a given word
such as "spam", type "modules spam".
```

```
help> █
```

```
Help on package scipy.stats in scipy:
```

## NAME

```
scipy.stats
```

## FILE

```
/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python/scipy/stats/__init__.py
```

## DESCRIPTION

```
=====
Statistical functions (:mod:`scipy.stats`)
=====
```

```
.. module:: scipy.stats
```

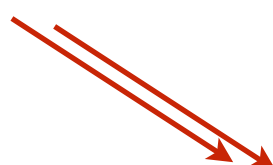
*type: script.stats or the name of a package  
and you get the descriptions about that package*



# few more hints

*macros available on USB bracelets need to be completed*

```
void sigma_interval_single_sided(){
    gStyle->SetOptStat(0);
    TH1F * h = new TH1F( "h_work",";x;f(x)", [redacted] );
    TF1* fga = new TF1( "f_work","gaus", [redacted] );
    fga->SetLineColor(3);
    fga->SetParameter( 0, [redacted] //normalisation
    fga->SetParameter( 1, [redacted] //mean
    fga->SetParameter( 2, [redacted] //width
```



```
void sigma_interval_single_sided(){
    gStyle->SetOptStat(0);
    TH1F * h = new TH1F( "h_work",";x;f(x)",100,-50, 50.);
    TF1* fga = new TF1( "f_work","gaus", -50, 50);
    fga->SetLineColor(3);
    fga->SetParameter( 0, 1.); //normalisation
    fga->SetParameter( 1, 0.); //mean
    fga->SetParameter( 2, 5.); //width
```

*if you figure out how to open the solution(s): you can go to the next level*

# few more tricks

*the solutions are password encrypted :)*

```
zip --encrypt -r sigma_interval_single_sided.zip sigma_interval_single_sided
```

*to open the zip files type:*

```
unzip sigma_interval_single_sided.zip
```

*then you will be asked for a password*

*at the end of each exercise you will get to know the password*