

# ROOT Fitting

March 2021 J. Smallcombe

# The Importance of Initial Guesses

## ROOT fit minimiser (default Minuit)

Very powerful

**Multi-purpose** (knows nothing)

$$\chi^2 \equiv \sum_i \frac{(\text{model}_i - \text{data}_i)^2}{(\text{uncertainty}_i)^2}$$

Minimizes chi-squared of fit function vs data

No user input : Parameter range  $\pm\infty$

Few parameter – can manage

Many parameters + no user guidance:

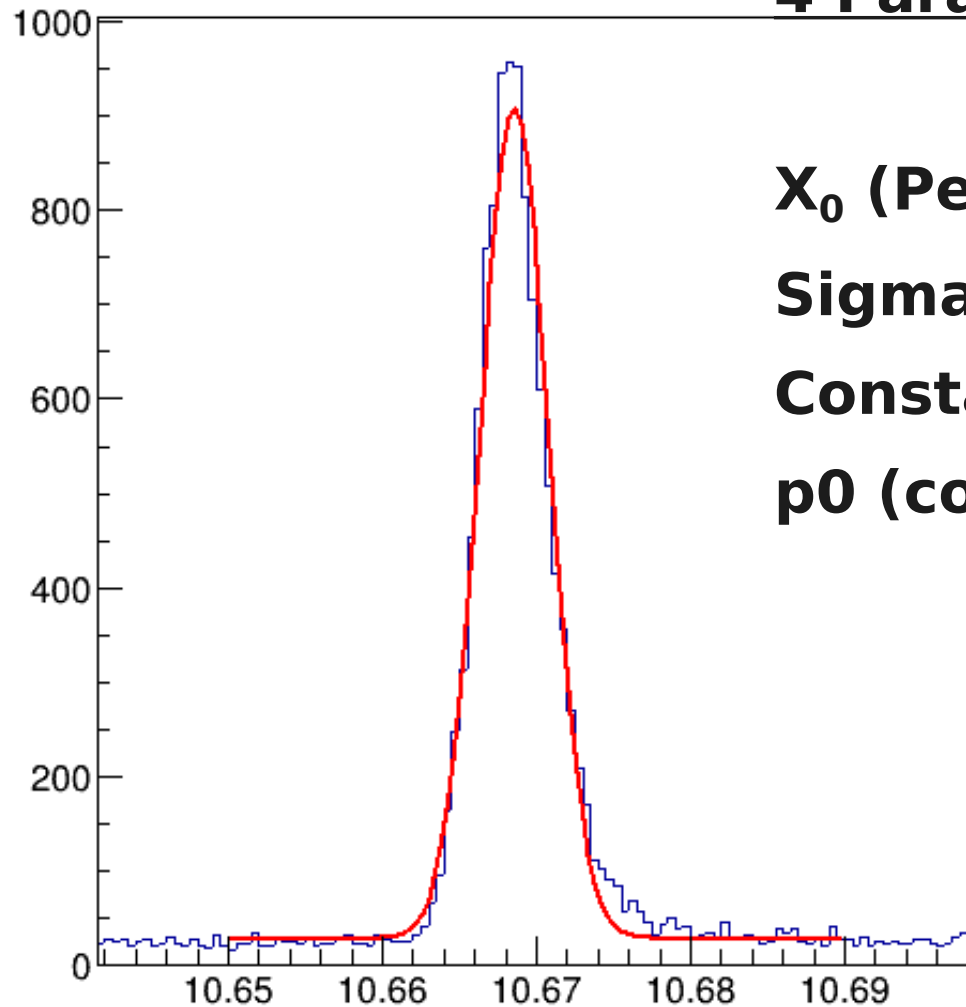
Easily caught in **false minima** of N-dimensional chi-squared surface.

**Initial Parameter Guesses & Ranges Important**

# The Importance of Initial Guesses

**Gaussian + Background**

**4 Parameter Fit**



$X_0$  (Peak Centroid)

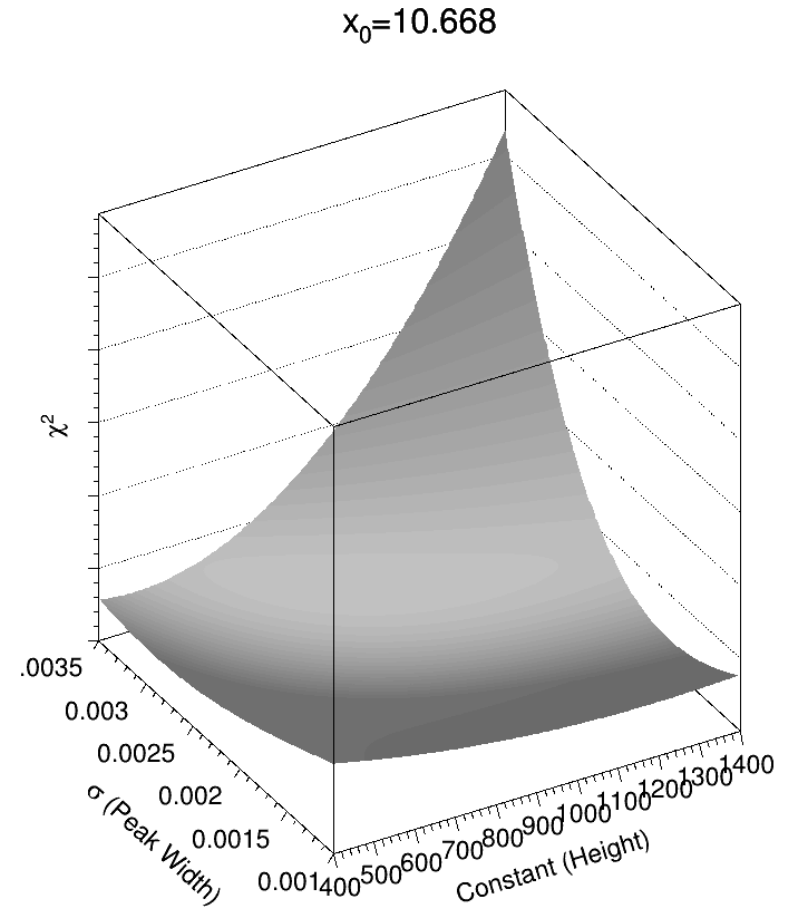
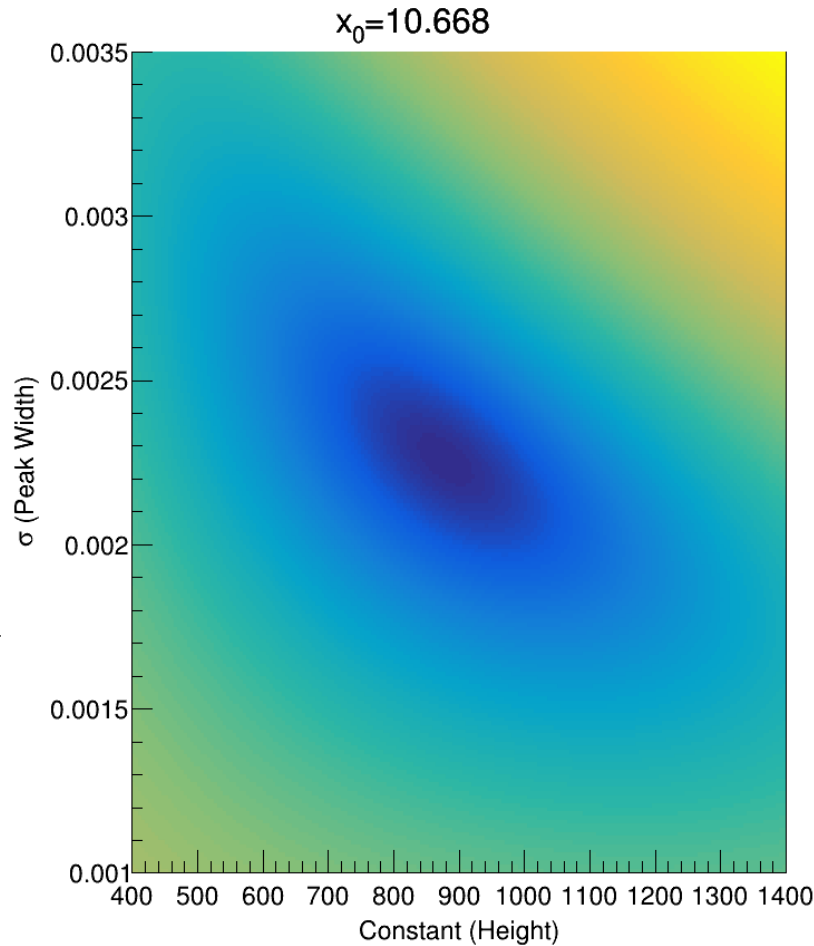
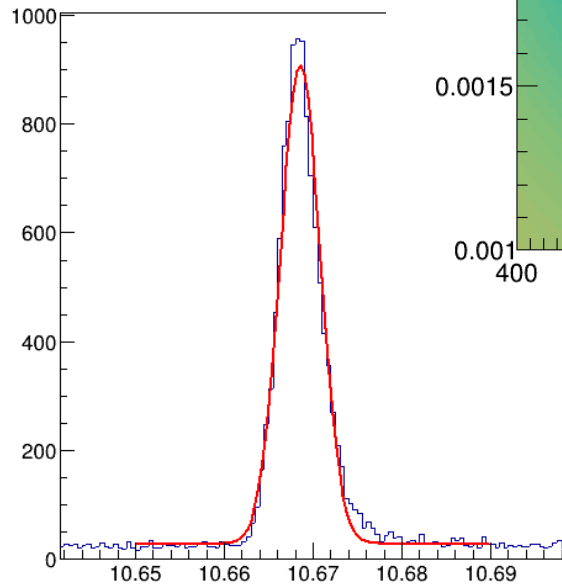
Sigma (Peak Width)

Constant (Peak height)

$p_0$  (constant background/baseline)

# The Importance of Initial Guesses

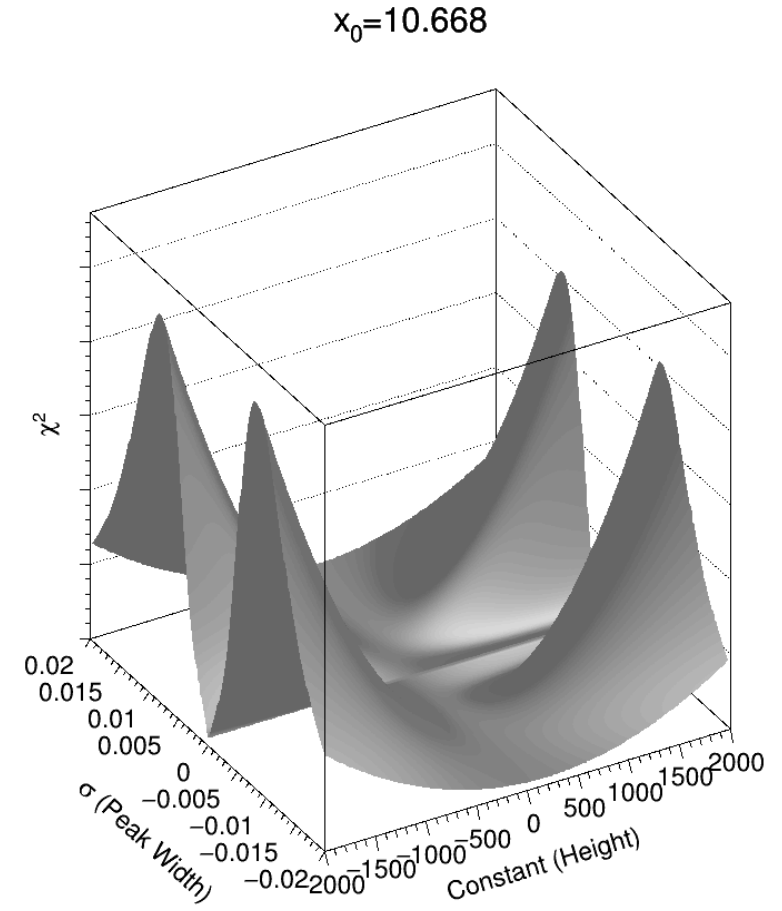
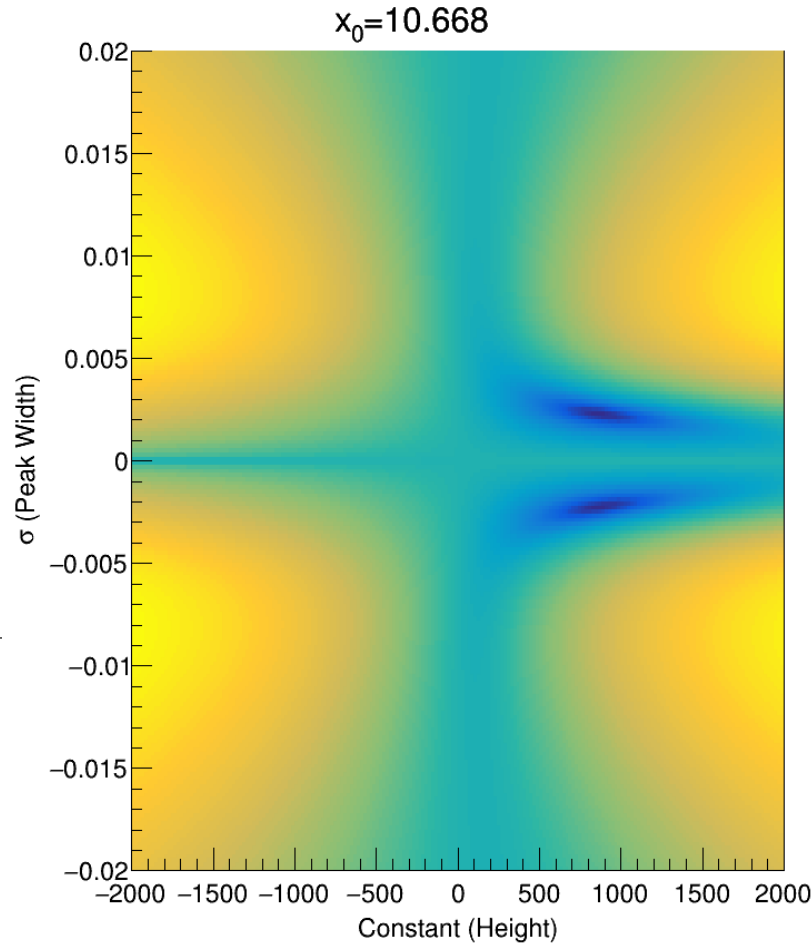
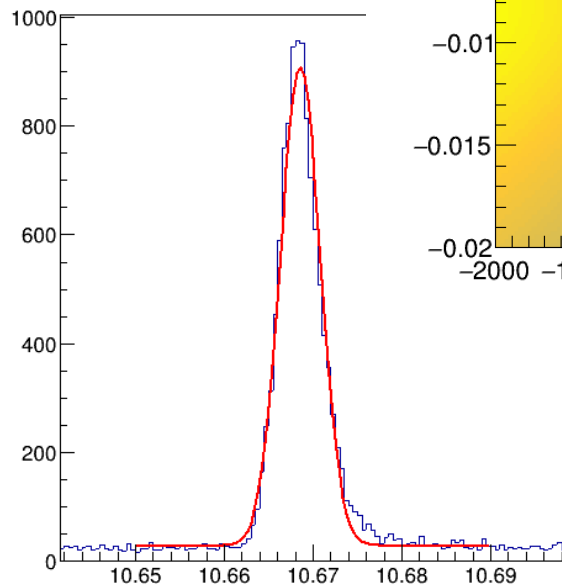
True  
Minimum  
Parameter  
range  
limited



$$\chi^2 \equiv \sum_i \frac{(\text{model}_i - \text{data}_i)^2}{(\text{uncertainty}_i)^2}$$

# The Importance of Initial Guesses

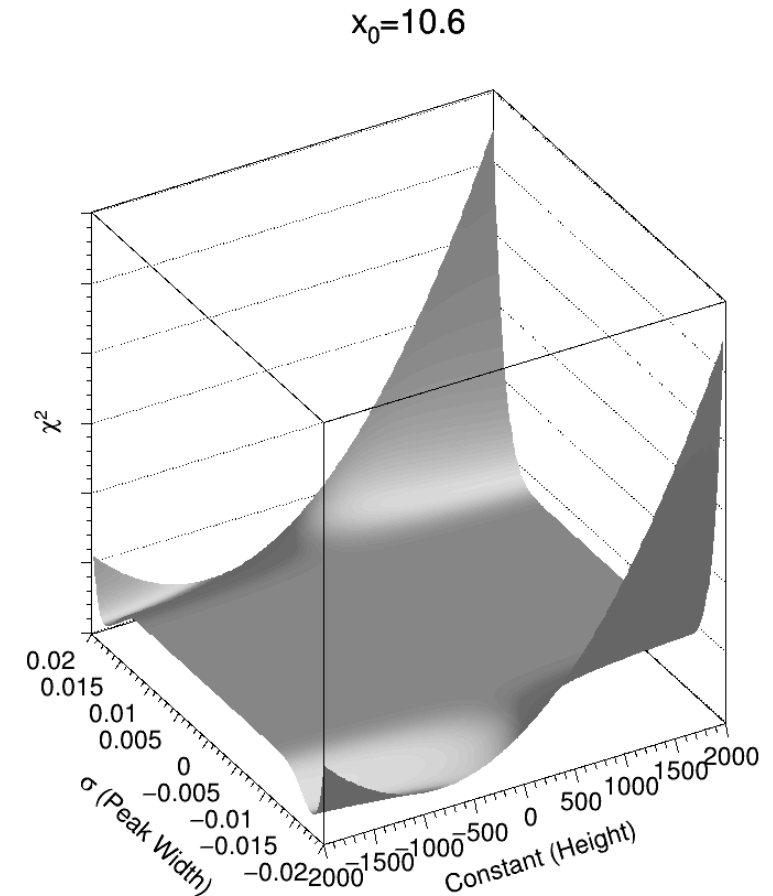
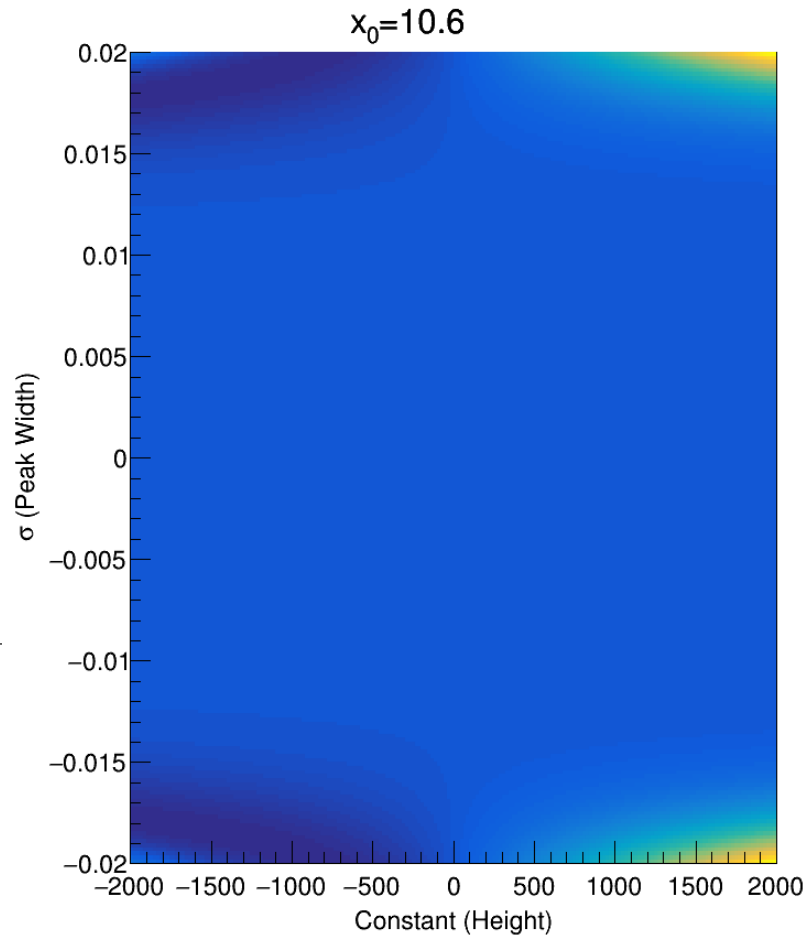
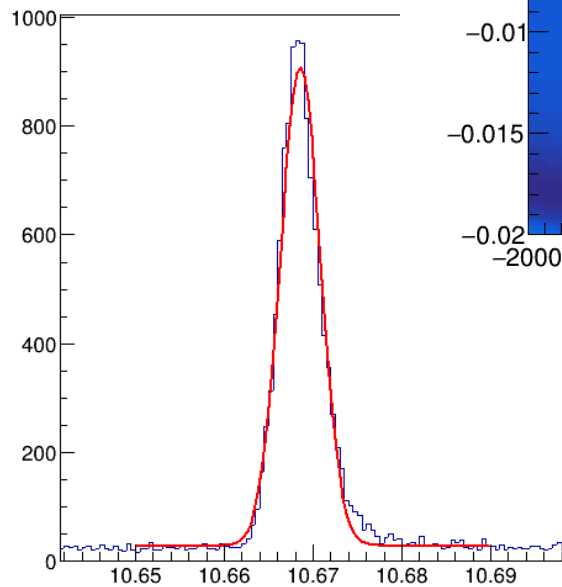
True  
Minimum  
  
Greater  
range



$$\chi^2 \equiv \sum_i \frac{(\text{model}_i - \text{data}_i)^2}{(\text{uncertainty}_i)^2}$$

# The Importance of Initial Guesses

$X_0$   
changed  
False  
Minima



$$\chi^2 \equiv \sum_i \frac{(\text{model}_i - \text{data}_i)^2}{(\text{uncertainty}_i)^2}$$

# Fit Function Class TF1

## Typical constructor:

***TF1("name", "formula", RL , RU )***

Name - ROOT special name

Formula - Input of the function of x as a string-type to interpreted.

(C++ code or predefined functions)

RL - Range Lower (optional)

RU - Range Upper (optional)

Many alternate forms (inc. direct reference to C function or C++ class defined in code)

# Fit Function Class TF1

**Typical constructor:**

***TF1("name", "formula", RL , RU )***

Examples:

```
TF1 func1("exfn1", "[0]+x*[1]")
```

```
TF1 func2("exfn2", "sin(x/TMath::Pi())", 1120, 1240)
```

```
TF1* funcPtr = new TF1("exfn2", "gaus+pol0")
```



# Do Perform a Fit

Call the Fit function of the object containing the data to be fit i.e. TH1 or TGraph

***TH1::Fit(TF1\* Pointer, “Options” )***

Many possible options maybe combined in the string “options”, most important:

“R” - use the range defined in the TF1

hist1.(funcPtr, “R”)

hist1.(&func1)

# Do Perform a Fit

Call the Fit function of the object containing the data to be fit i.e. TH1 or TGraph

***TH1::Fit(TF1\* Pointer, “Options” )***

Many possible options maybe combined in the string “options” most important.

& provide a pointer to a local object

hist1.(&func1)

# TF1 Useful Class Functions

**TF1::SetParameters(value, value, value ... )**

Set starting values for all parameters

**TF1::SetParameter(i, value)**

Set values for parameters i

**TF1::SetParLimits(i, lower, range)**

Set limits on allowed values of parameters i

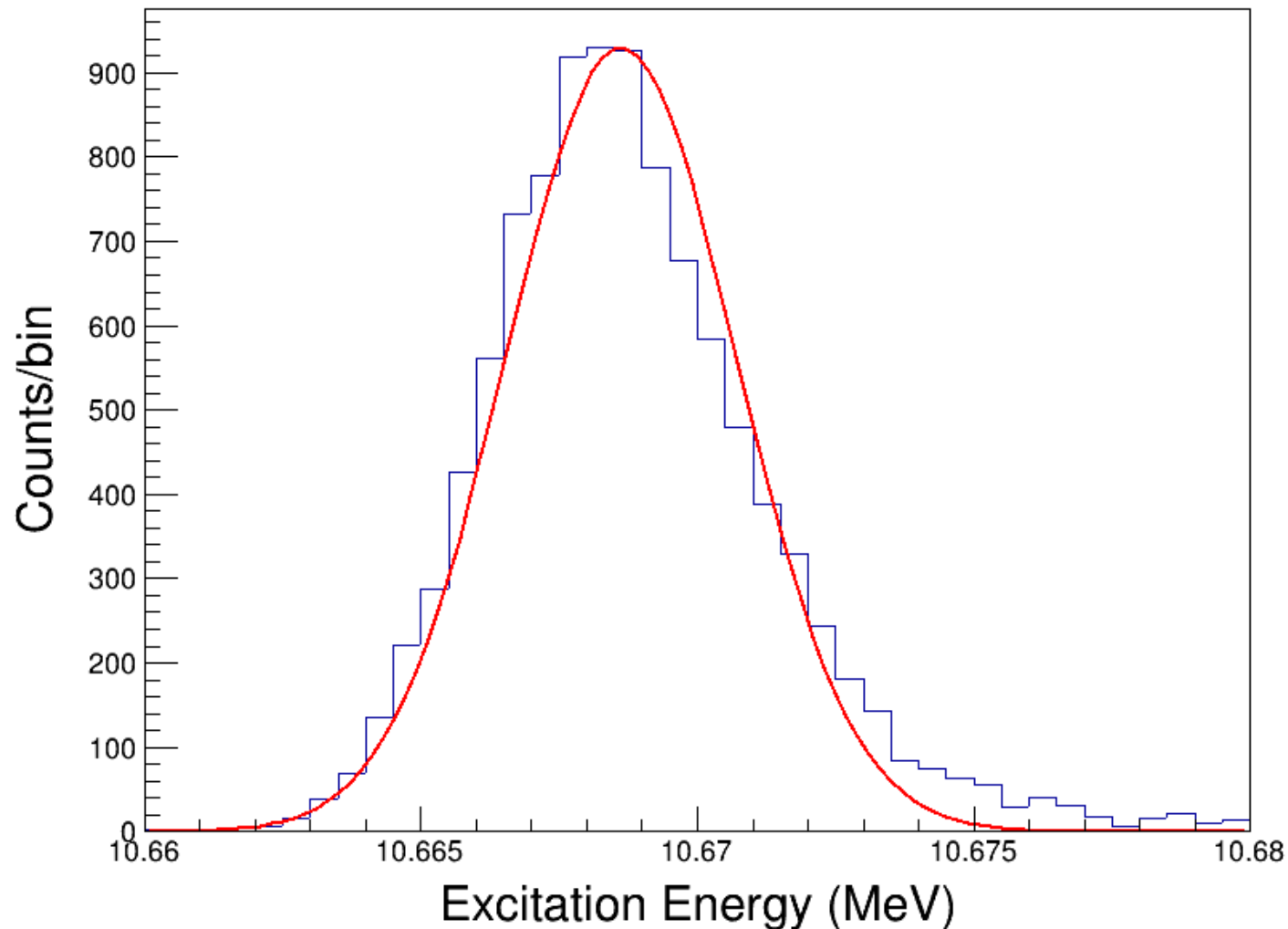
**TF1::GetParameter(i)**

**TF1::GetParError(i)**

Return determined parameter/error following a fit

# TF1 and Bin Width

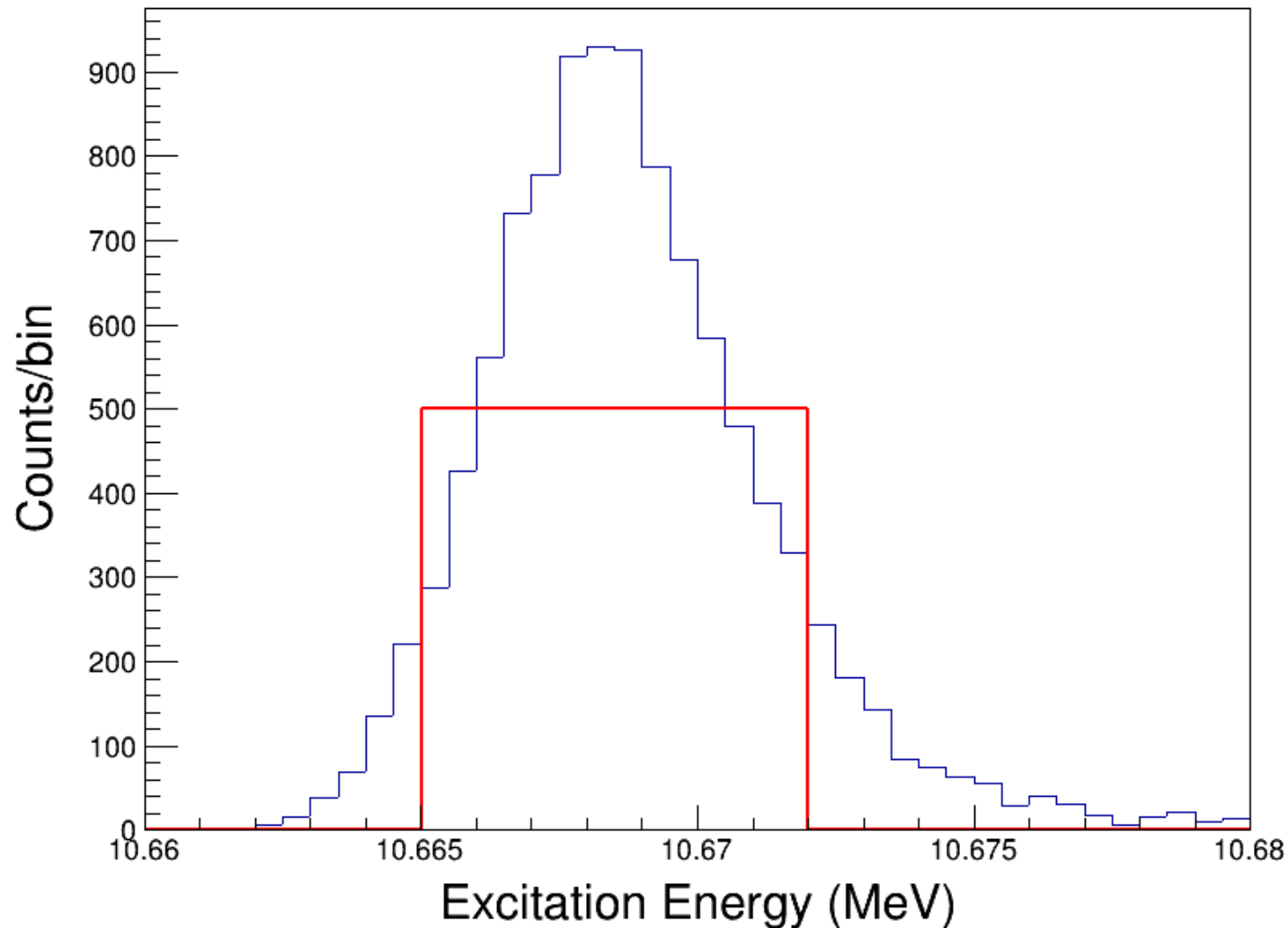
**TF1 in user coordinated, not bins! XY mismatch**  
**Histogram height - depends on bin width.**



# TF1 and Bin Width

**TF1 is in user coordinated not bins!**

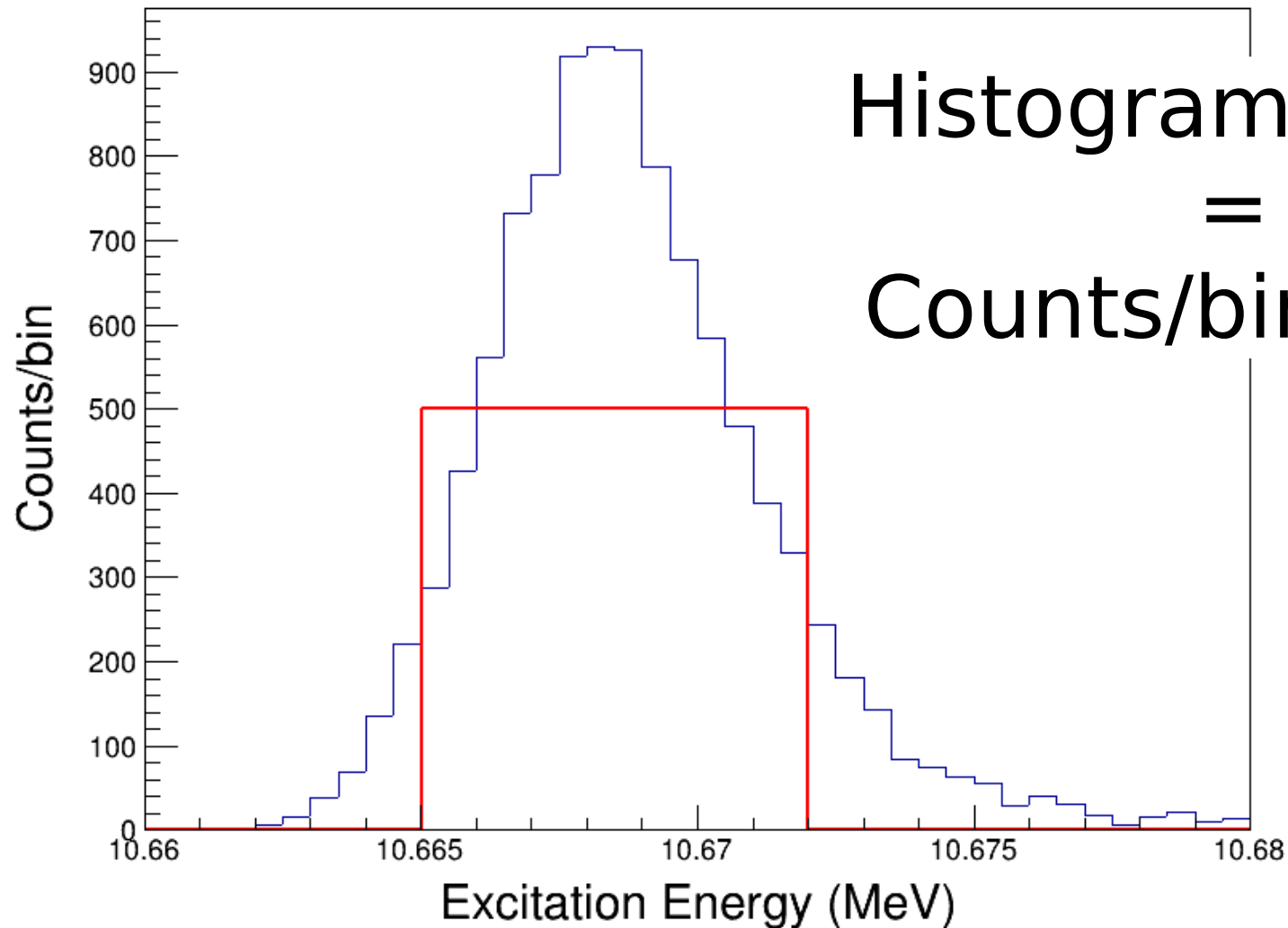
**Histogram height - depends on bin width.**



# TF1 and Bin Width

**TF1 is in user coordinated not bins!**

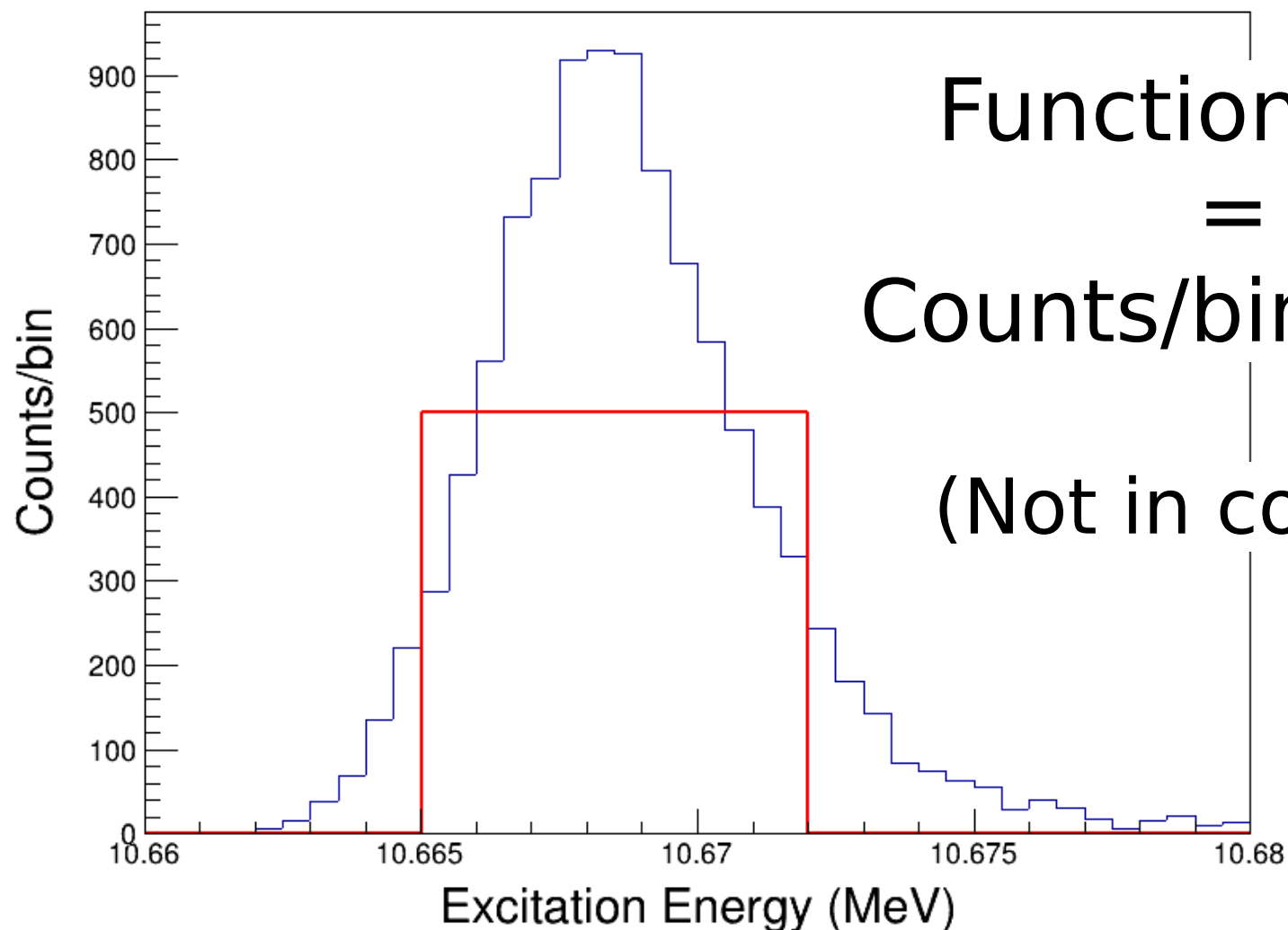
**Histogram height - depends on bin width.**



# TF1 and Bin Width

**TF1 is in user coordinated not bins!**

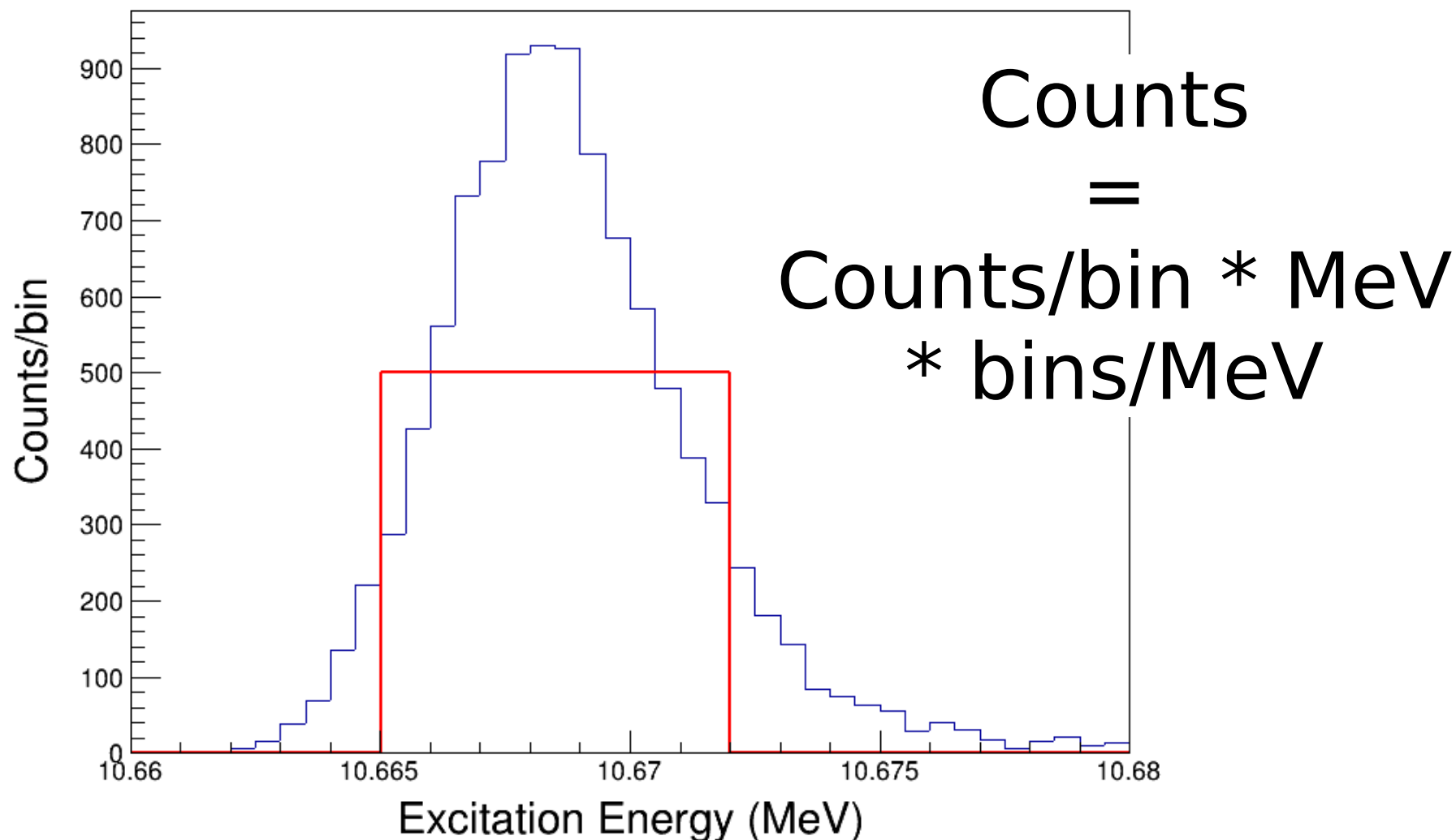
**Histogram height - depends on bin width.**



# TF1 and Bin Width

**TF1 is in user coordinated not bins!**

**Histogram height - depends on bin width.**





# TF1 and Bin Width

**TF1 is in user coordinated not bins!**

**Histogram height - depends on bin width.**

