

---

# Datenmodellierung mit *RooFit*

---

Julian Boelhauve  
11. März 2022  
Programmierkurs

# Was ist *RooFit*?

- Bibliothek zur Datenmodellierung mit *ROOT*
- Verwendung der *Maximum-Likelihood*-Methode
- Hilfe
  - *RooFit*-Handbuch (*user's manual*)
  - *RooFit*-Codebeispiele (*tutorials*)
  - *ROOT*-Forum

# Laden eines Datensatzes

```
import ROOT as R
```

*ROOT*-Modul in *Python*

```
file = R.TFile.Open("programming_course.root", "READ")
```

*Tree* aus *ROOT*-Datei

```
tree = file.Get("Gaussian")
```

```
x = R.RooRealVar("x", "x", 1.0, 7.0)
```

Variable mit (*Branch*-)Namen,  
Titel und Intervall

```
data = R.RooDataSet("data", "data", tree, R.RooArgSet(x))
```

Datensatz aus *Tree*

# Fitten einer Wahrscheinlichkeitsdichtefunktion

```
mu = R.RooRealVar("mu", "mu", 4.0, 3.0, 5.0)
sigma = R.RooRealVar("sigma", "sigma", 1.0, 0.1, 2.0)
gauss = R.RooGaussian("gauss", "gauss", x, mu, sigma)
gauss.fitTo(data)
```

Variablen mit Startwerten und  
Intervallen

Gauß-Verteilung

Fitten

Argumente von `fitTo()`: Anzahl der Prozessorkerne, Speichern des Fitresultates, ...

# Fitresultat

```
START MIGRAD MINIMIZATION. STRATEGY 1. CONVERGENCE WHEN EDM .LT. 1.00e-03
FCN=764.33 FROM MIGRAD STATUS=INITIATE 25 CALLS 26 TOTAL
EDM= unknown STRATEGY= 1 NO ERROR MATRIX
EXT PARAMETER CURRENT GUESS STEP FIRST
NO. NAME VALUE ERROR SIZE DERIVATIVE
1 mu 4.00000e+00 2.00000e-01 0.00000e+00 -4.65747e+01
2 sigma 5.36400e-01 1.90000e-01 -5.18532e-01 9.45600e+01
ERR DEF= 0.5
MIGRAD MINIMIZATION HAS CONVERGED.
MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=762.945 FROM MIGRAD STATUS=CONVERGED 48 CALLS 49 TOTAL
EDM=1.455e-07 STRATEGY= 1 ERROR MATRIX ACCURATE
EXT PARAMETER STEP FIRST
NO. NAME VALUE ERROR SIZE DERIVATIVE
1 mu 4.01341e+00 1.64091e-02 3.12571e-04 2.27344e-02
2 sigma 5.18925e-01 1.16031e-02 2.80155e-04 -5.39700e-03
```

# Fitresultat

- Etwaige Warnungen nicht ignorieren
- Bei Konvergenzproblemen
  - Startwerte der Fitparameter geeigneter wählen
  - Grenzen der Fitparameter anpassen
- Resultat nur mit Parameterunsicherheiten und korrekt bestimmter Kovarianzmatrix vertrauenswürdig

# Plotten

```
canv = R.TCanvas("canv", "canv")
```

 Bereich zum Plotten

```
frame = x.frame()
```

```
data.plot0n(frame)
```

```
gauss.plot0n(frame)
```

```
frame.Draw()
```

Darzustellende Variable und Komponenten

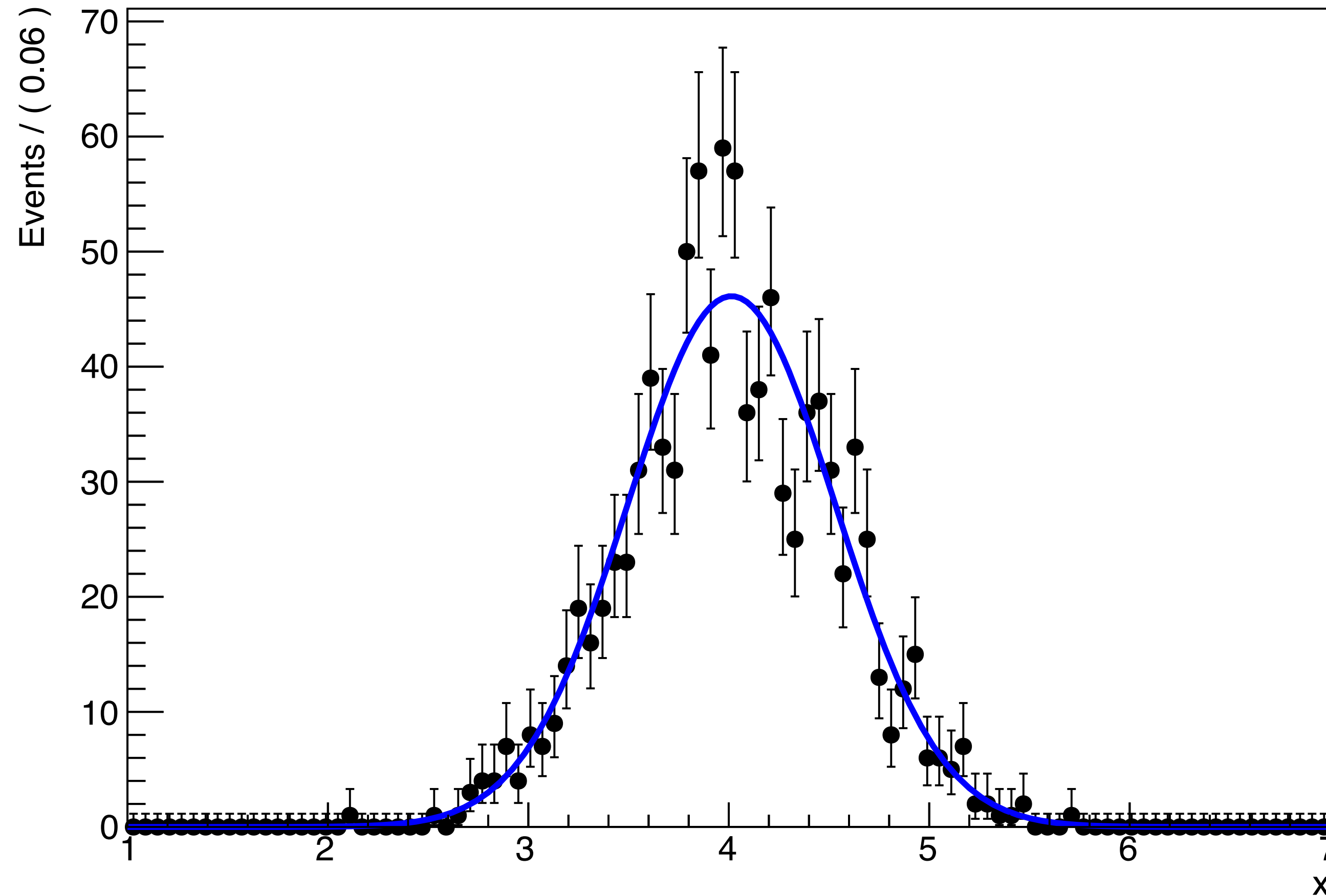
```
canv.SaveAs("1.pdf")
```

Speichern

Unterdrücken des *Canvas*-Fensters: `R.gROOT.SetBatch(True)`

# Plotresultat

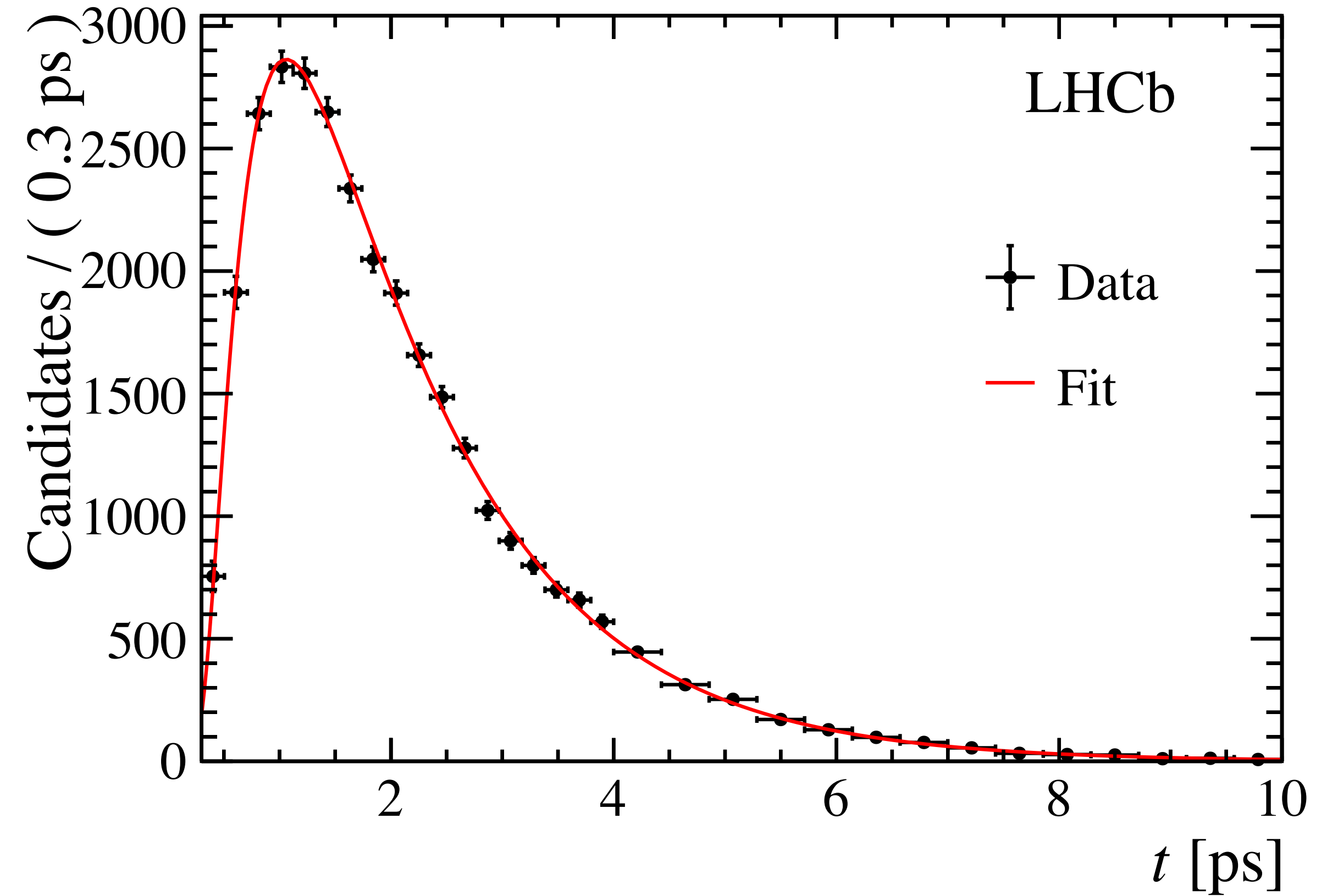
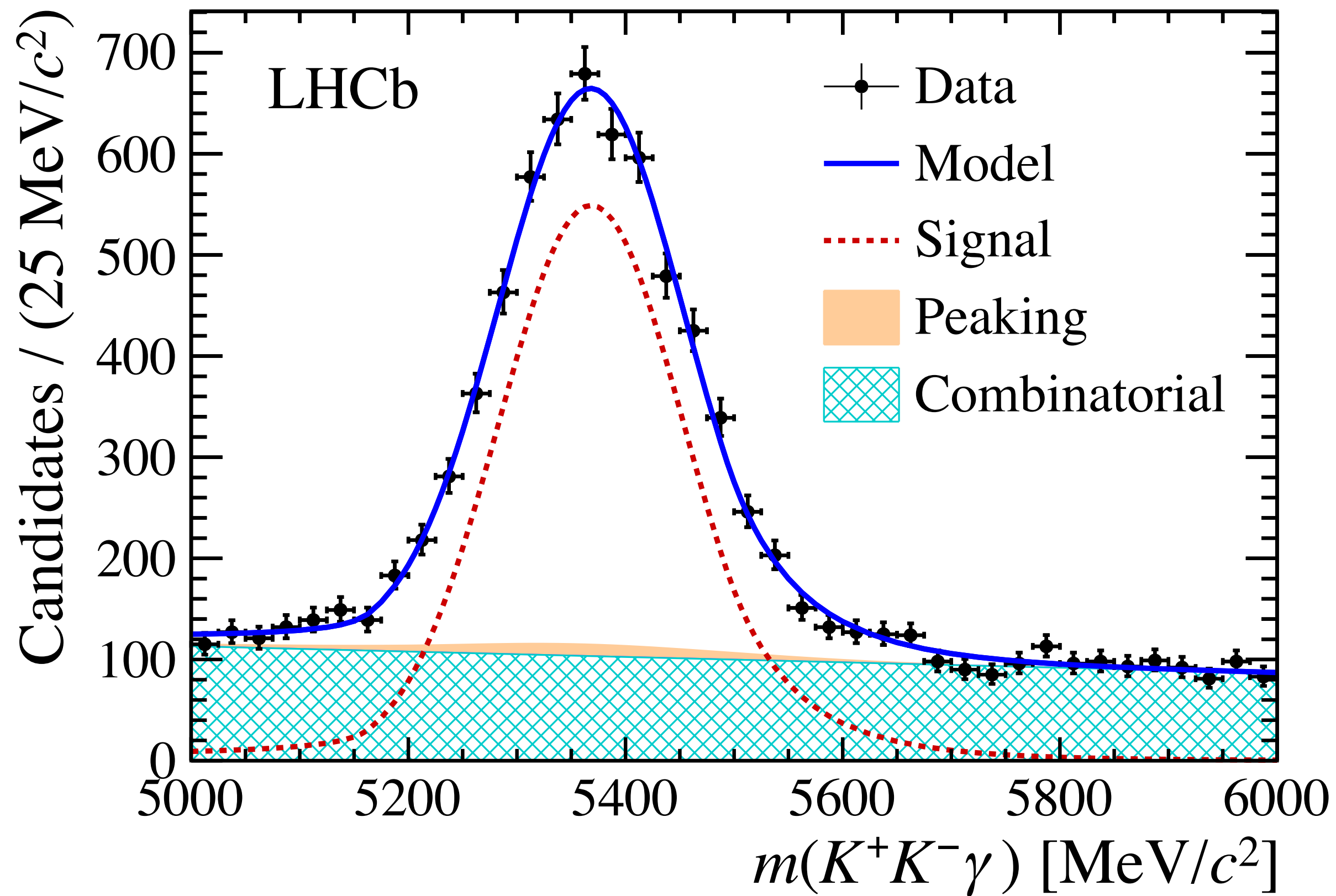
A RooPlot of "x"



- Fit erfolgreich
- Titel wenig aussagekräftig
- Möglicherweise fehlende Einheit in den Achsenbeschriftungen
- Keine Legende



# Plotbeispiele



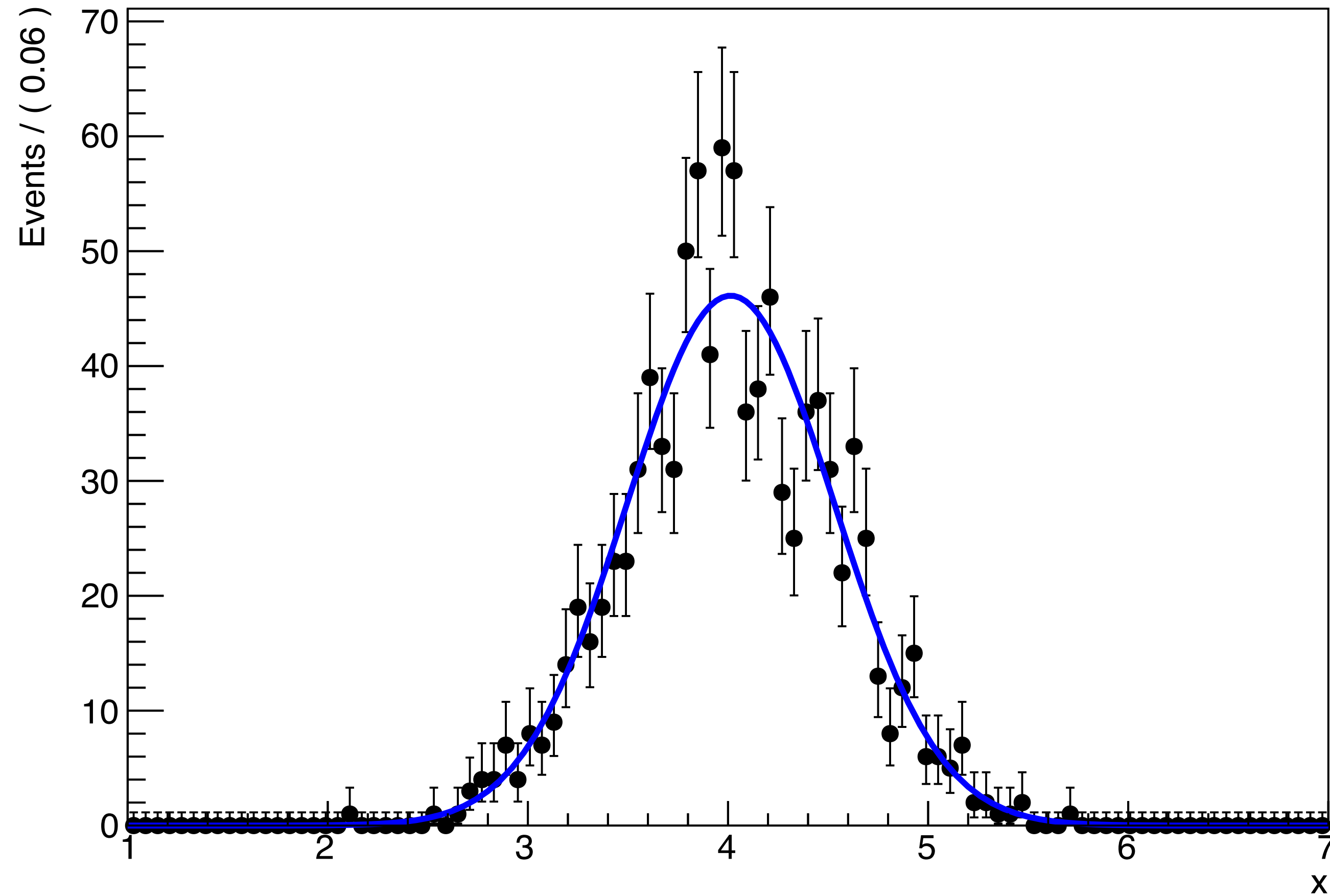
[Phys. Rev. Lett. 123 (2019) 081802]

# Aufgabe 1

- Öffne die Datei „programming\_course.root“ und lies den *Tree* „Gaussian“ ein.
- Erstelle aus dem *Tree* einen Datensatz mit der Variablen  $x$ .
- Definiere eine Gauß-Verteilung und fitte diese an die Daten.
- Erzeuge einen Plot mit den Daten und dem Fit.

# Plot zu Aufgabe 1

A RooPlot of "x"

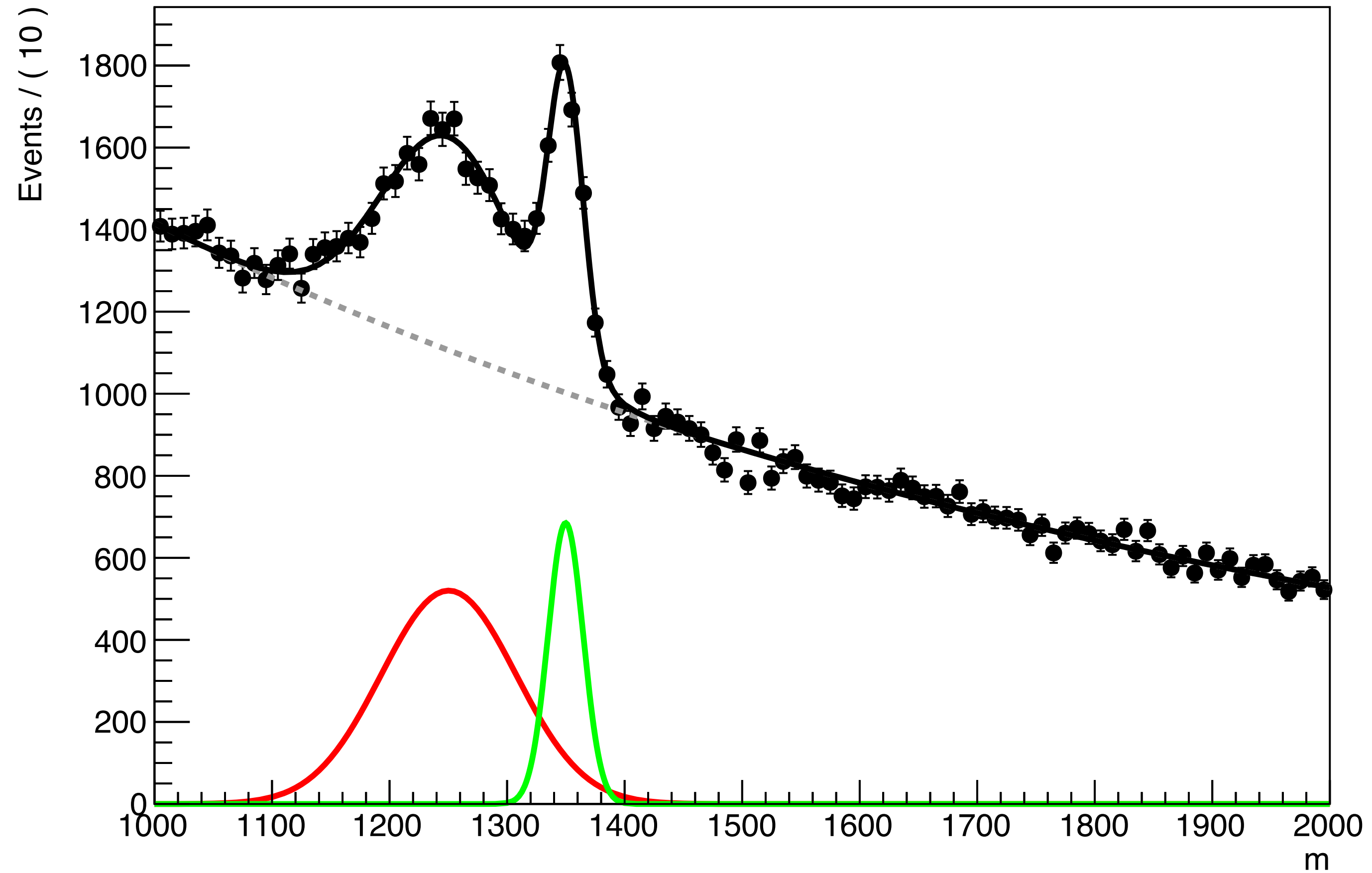


# Aufgabe 2

- Informiere dich über die Funktionsweise von RooAddPdf.
- Lies den *Tree* „RooAddPdf“ ein.
- Definiere ein Modell aus zwei Gauß-Verteilungen und einer Exponentialverteilung, welche den Untergrund beschreiben soll.
- Führe sowohl einen *Maximum-Likelihood-Fit* als auch einen *Extended-Maximum-Likelihood-Fit*, welcher direkt die Anzahl der Kandidaten in den einzelnen Fitkomponenten liefert, durch.
- Erstelle einen Plot, der die Fitkomponenten zeigt.

# Plot zu Aufgabe 2

A RooPlot of "m"

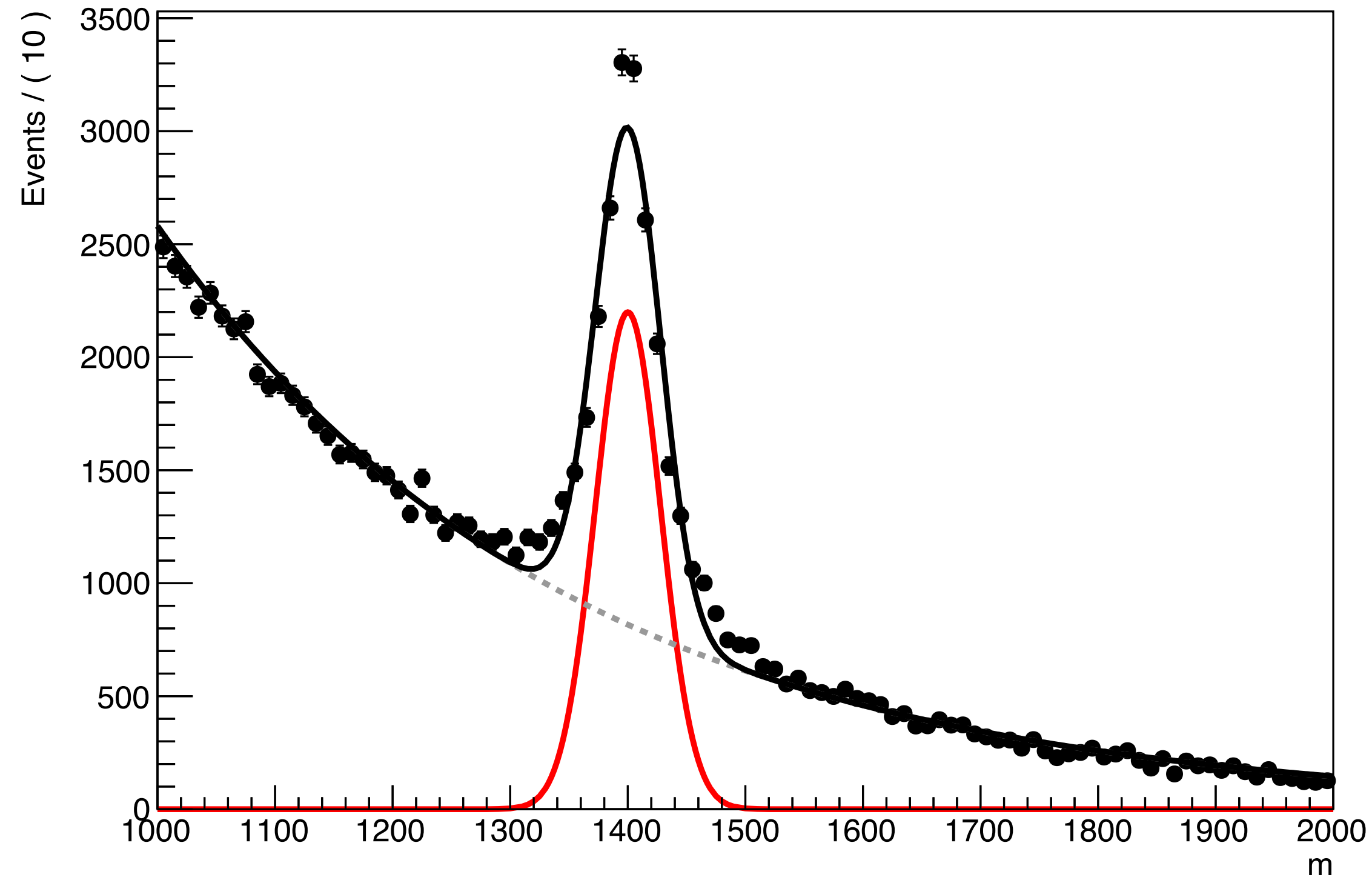


# Aufgabe 3

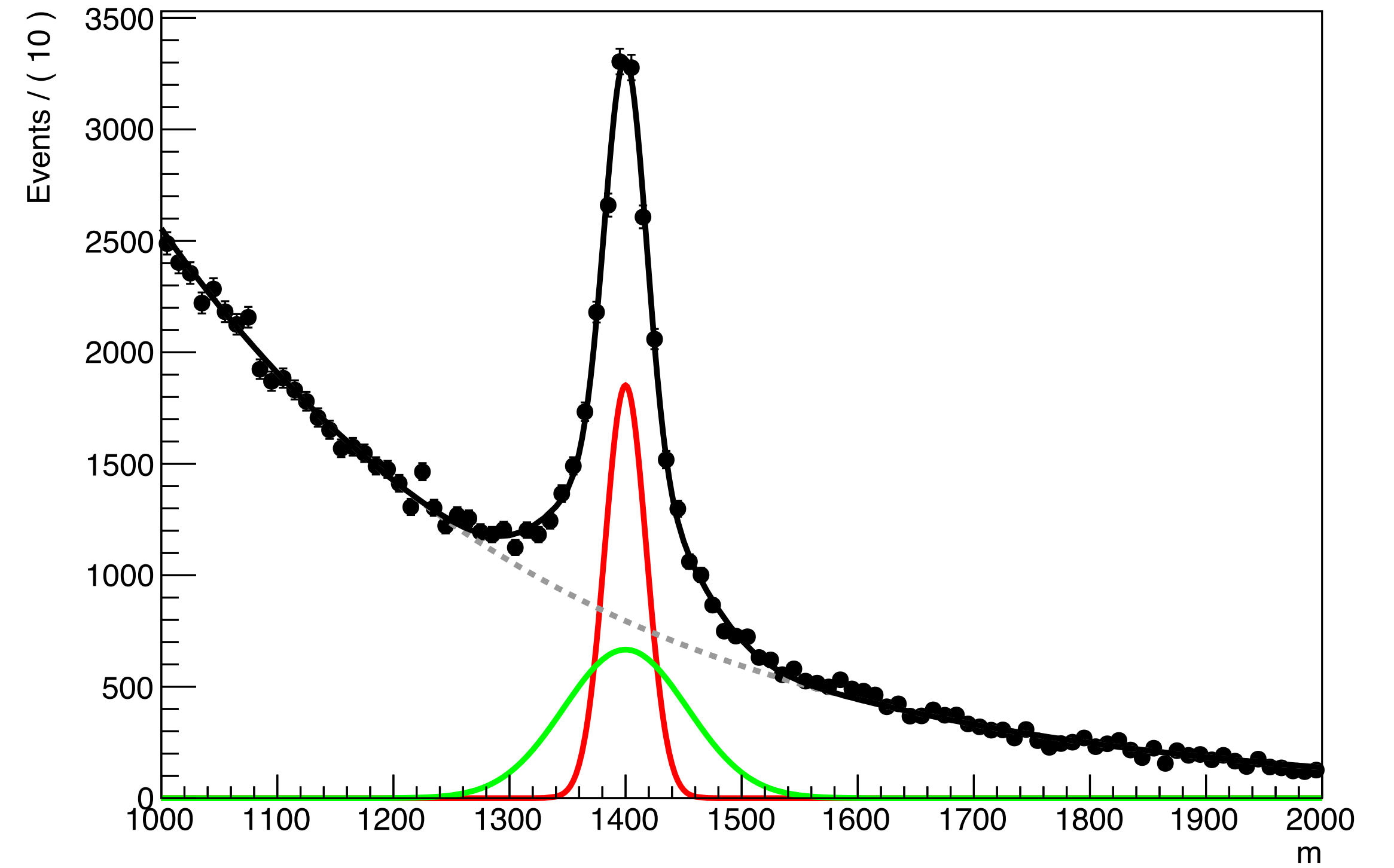
- Lies den *Tree* „Double\_Gaussian“ ein.
- Fitte die Daten mit einer Gauß-Verteilung und einer Exponentialverteilung.
- Erstelle einen Plot.
- Verwende dann zwei Gauß-Verteilungen, die den gleichen Mittelwert besitzen, zur Beschreibung des Signals.
- Fitte und plotte erneut.

# Plots zu Aufgabe 3

A RooPlot of "m"



A RooPlot of "m"



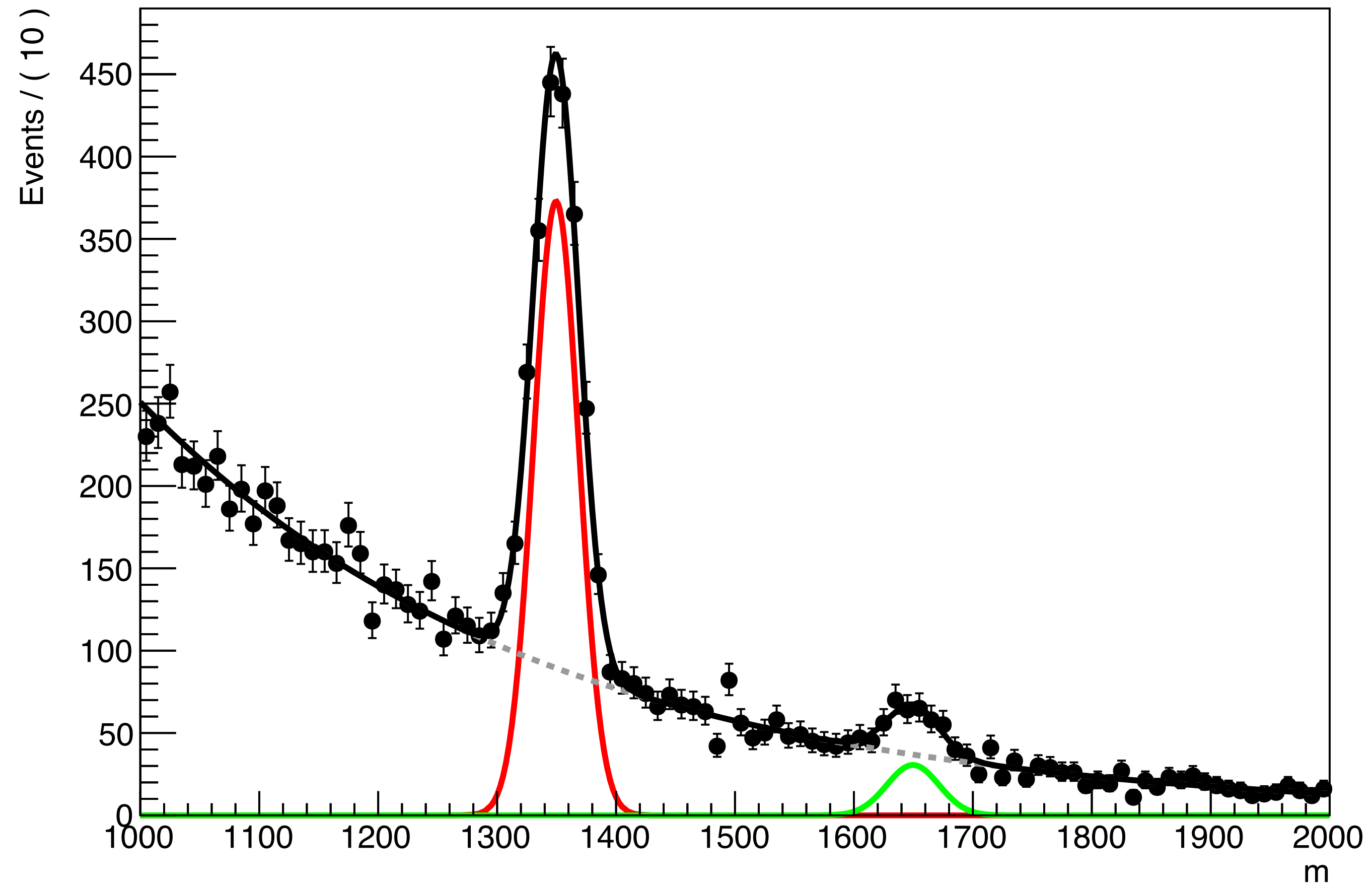
# Aufgabe 4

- Informiere dich über die Funktionsweise von `RooFormulaVar`.
- Lies den *Tree* „RooFormulaVar“ ein.
- Definiere ein Modell, das die beiden Resonanzen und den Untergrund beschreibt.
- Stelle den Mittelwert der rechten Resonanz, welche nur eine geringe Statistik besitzt, als Summe aus dem Mittelwert der linken Resonanz und einer Konstanten mit dem Wert  $300 \text{ MeV}/c^2$  dar.
- Fitte und plotte.



# Plot zu Aufgabe 4

A RooPlot of "m"



# Aufgabe 5

- Lies den *Tree* „Sidebands“ ein.
- Fitte die beiden Seitenbänder mit einer Exponentialverteilung. Schränke dazu den Fitbereich entsprechend ein.
- Erstelle einen Plot, der den Fit im gesamten Massenbereich zeigt.

# Plot zu Aufgabe 5

A RooPlot of "m"

