

と表示されます。これは、`TH1D::GetMean`<sup>\*9</sup>の引数の説明が表示されたものです。引数には整数 (integer) 値を取り、そのデフォルト (default) 値が1だという意味です (1はX軸を意味しています)。何も打たなくてもX軸を指定するデフォルト値が入っているのです、

```
root [3] hist->GetMean()
(const Double_t)8.33116798564013245e-03
```

と入力して改行しましょう。出力された値は、このヒストグラムの平均値です。ウインドウの右上に既に表示されている値と同一なはずです。同様に、`TH1D::GetRMS()`も試してみましょう。

同様に、

```
root [4] hist->Set
```

でタブ補完をすると、“Set”で始まる関数がたくさん表示されます。これらのメンバ関数はセッター (setter) と呼ばれ、ゲッターと対をなすものです。ゲッターとセッター以外にも多くのメンバ関数が存在しますが、ここでは説明しません。試しに

```
root [4] hist->SetLineColor(2)
root [5] hist->SetTitle("x")
root [6] hist->SetTitle("Number of Events")
root [7] hist->Draw("e")
```

としてみましょう。出力結果と見比べて、何が起きたか分かるはずですが、先ほどまでは`TH1D::Draw`の引数に何も与えていませんでしたが、今度は“e”という引数がついています。なぜこれで動作するかというと、これも、引数のデフォルト値が存在していたためです。

```
root [8] hist->Draw(
```

でタブを押して、意味を理解して下さい。`first_example.C`では、引数のデフォルト値を設定していません。そのため、引数を必ず両方とも指定しないと正しく動作しません。

## 2.3 やっておきたい初期設定

さて、図 2.4 を見ると、背景色が薄い灰色です。これでは、本書の紙面や通常の論文で使用するには問題があります。研究室のみで使うような図だとしても、プリンタのインクの無駄です。ヒストグラムのタイトル部分を見ると、“σ”と他の文字のフォントが一致していません。文章自体のフォントに対しても太すぎます。これでは格好悪いので、少しROOTの設定を変更してみましょう。

以下の内容を、`~/rootlogon.C`として保存して、再度ROOTを起動しましょう。

```
gROOT->SetStyle("Plain");
gStyle->SetTitleBorderSize(2);
gStyle->SetFrameFillColor(0);
gStyle->SetCanvasColor(0);
gStyle->SetPadBorderSize(0);

gStyle->SetPadTickX(1);
gStyle->SetPadTickY(1);
```

<sup>\*9</sup> この書き方は、クラス `TH1D` のメンバ関数 `GetMean` という意味です。

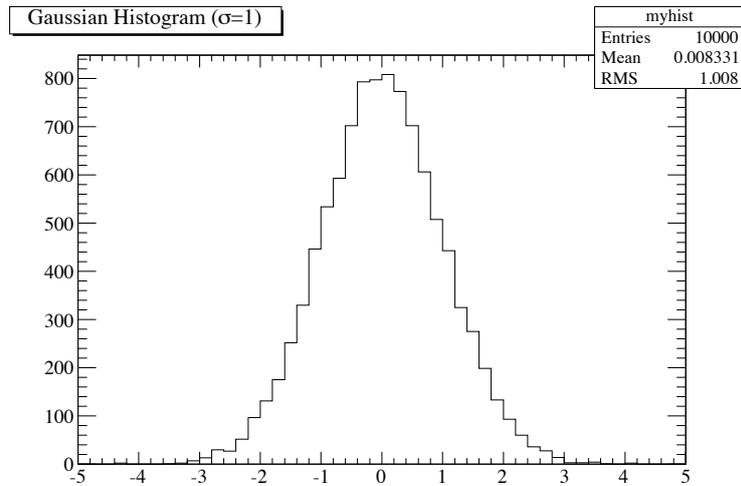


図 2.5 `~/rootlogon.C` を使って図 2.4 の見栄えを変更したもの

```

Int_t fontid = 132;
gStyle->SetStatFont(fontid);
gStyle->SetLabelFont(fontid, "XYZ");
gStyle->SetLabelFont(fontid, "");
gStyle->SetTitleFont(fontid, "XYZ");
gStyle->SetTitleFont(fontid, "");
gStyle->SetTitleOffset(1.2, "XYZ");
gStyle->SetTextFont(fontid);
gStyle->SetFuncWidth(2);
gStyle->SetLegendBorderSize(0);

gStyle->SetPalette(1);

```

再び `first.script.C` を走らせると、図 2.5 のような結果が得られます。まだまだ見栄えを変更可能ですが、ひとまずはよしとします。もし図 2.4 のほうが好みであれば、`~/rootlogon.C` の不要な行を消していきましょう。