

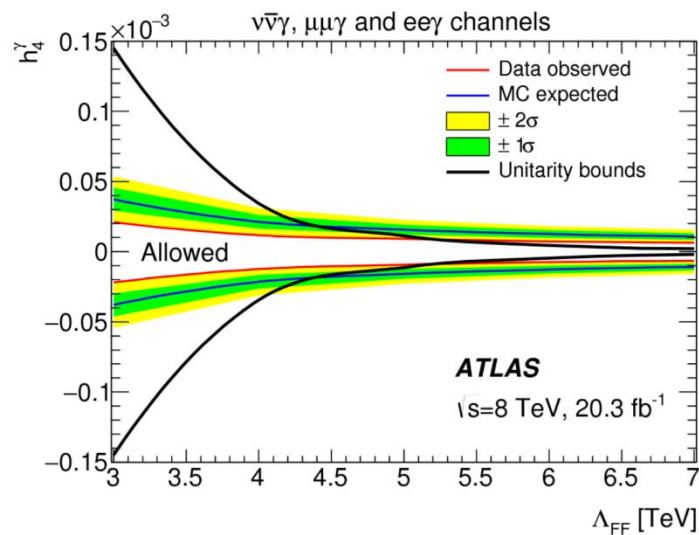
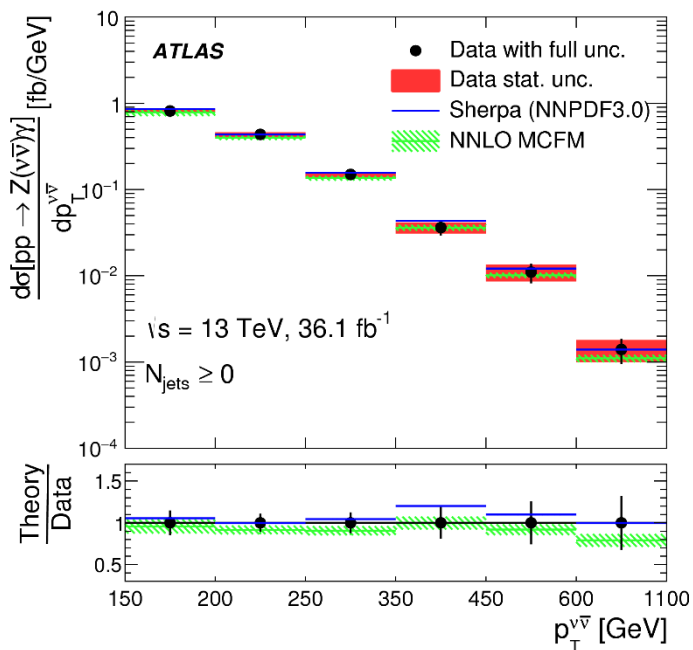
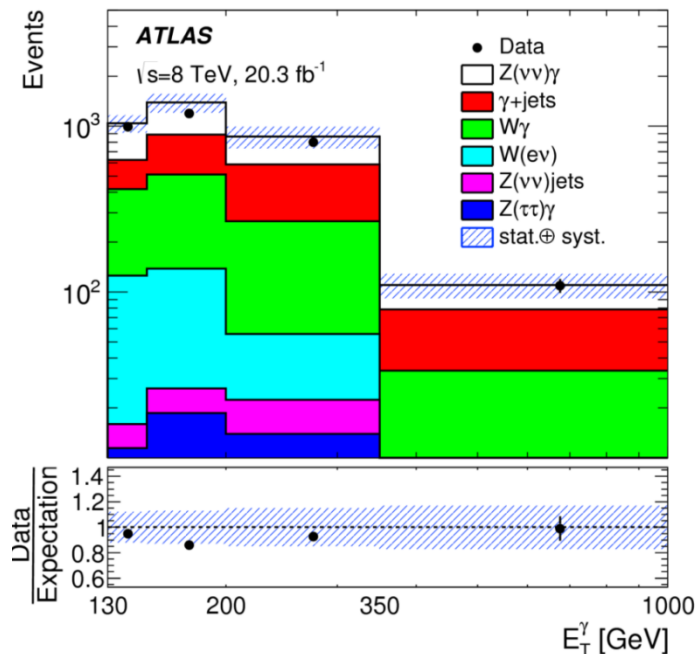
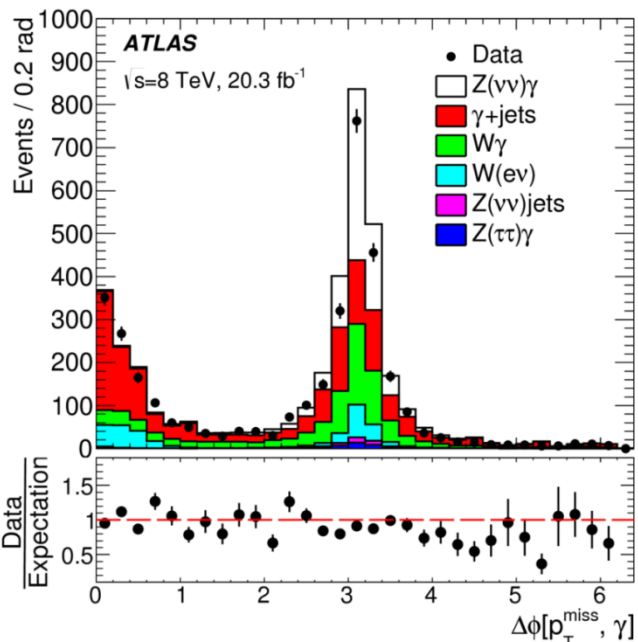


Лекции. Практические занятия

Солдатов Е.Ю.

2020 г.

ГИСТОГРАММЫ



ГИСТОГРАММЫ

- ROOT поддерживает гистограммы, в которых не более 3 измерений. Мы начнём с одномерных гистограмм.
- Гистограмма – это наглядное изображение распределения случайной величины при условии ограниченности статистики выборки
- Основные используемые классы:

TH1C, TH2C и TH3C резервируют по 1 байту на бин (максимальное содержимое 1 бина = 255)

TH1I, TH2I и TH3I резервируют одну переменную типа int на бин (максимальное содержимое бина = 2 147 483 647).

TH1F, TH2F и TH3F резервируют одну переменную типа float на бин (максимальная точность бина, значащих цифр = 7 digits).

TH1D, TH2D и TH3D резервируют одну переменную типа double на бин (максимальная точность бина, значащих цифр = 14 digits).

ГИСТОГРАММЫ

- Конструкторы:

```
THIF *hI = new THIF("HistName", "Histogram title", Nbins, xmin, xmax)
```

```
double xarray[Nbins+1]={xmin,...,xmax};
```

```
THIF *hI = new THIF("HistName", "Histogram title", Nbins, xarray)
```

Создается гистограмма (объект класса **THIF**) и указатель **hI** на этот объект.

- Параметры:

- **HistName** – имя гистограммы, без пробелов

- **Histogram title** – заголовок гистограммы

Заголовок гистограммы может сразу включать и заголовки осей, например:

“Histogram title; x [mm]; dN/dx”

- **Nbins** – число бинов в гистограмме (целая величина)

- **xmin, xmax** – диапазон изменения гистограммируемой величины

- **xarray** – массив границ бинов (для неравномерного биннирования)

ГИСТОГРАММЫ

Наиболее важные методы:

Заполнение и манипуляция данными внутри гистограммы

- o `h I->Fill(value, weight)` // `weight` – необязательный параметр.

`h I->SetBinContent(bin, value)`

`h I->SetBinError(bin, error)`

`h I->GetBinContent(bin)`

`h I->GetBinError(bin)`

Изменение предустановленных характеристик:

`h I->SetTitle("title")`

`h I->Rebin(n_m_bins, new_title, xarray)` // `xarray` - необязательный

Изменение внешнего вида

`h I->SetFillColor(color)`

`h I->SetLineWidth(n_of_pt)`

`h I->SetMarkerStyle(n_marker)`

Подписать оси гистограмм:

`h I->GetXaxis()->SetTitle("Xaxis title"); h I->GetYaxis()->SetTitle("Yaxis title");`

Получить интеграл гистограммы:

`h I->Integral()`

Нормировка гистограммы на величину `norm`:

`Double_t scale= norm/h I->Integral(); h I->Scale(norm);`

ГИСТОГРАММЫ

Вывод гистограммы

`h1->Draw("options")`

options – любые опции рисования, можно без пробелов (sameCHIST)

Некоторые варианты опций:

C — провести кривую через значения бинов;

P — нарисовать маркеры для каждого значения;

E — нарисовать погрешности значений (E0-E6 – разные варианты изображения погрешностей гистограммы).

Чтобы нарисовать две гистограммы на одном рисунке, следует задать опцию same для второй:

`h1->Draw()`

`h2->Draw("same")`

Подробнее: <https://root.cern/doc/v610/classTHistPainter.html#HP01>

ГИСТОГРАММЫ

- Создадим, заполним и выведем гистограмму:
 - `TH1F *h1 = new TH1F("h1", "New hist",25,-2.5,2.5);`
`h1->Fill(-2.3);`
`h1->Fill(1.5);`
`h1->Fill(0,4);`
`h1->Fill(-1.2,2);`
`h1->Fill(0.8,3);`
`h1->Draw();`

Окно стат.

информации:

h1 – имя объекта;

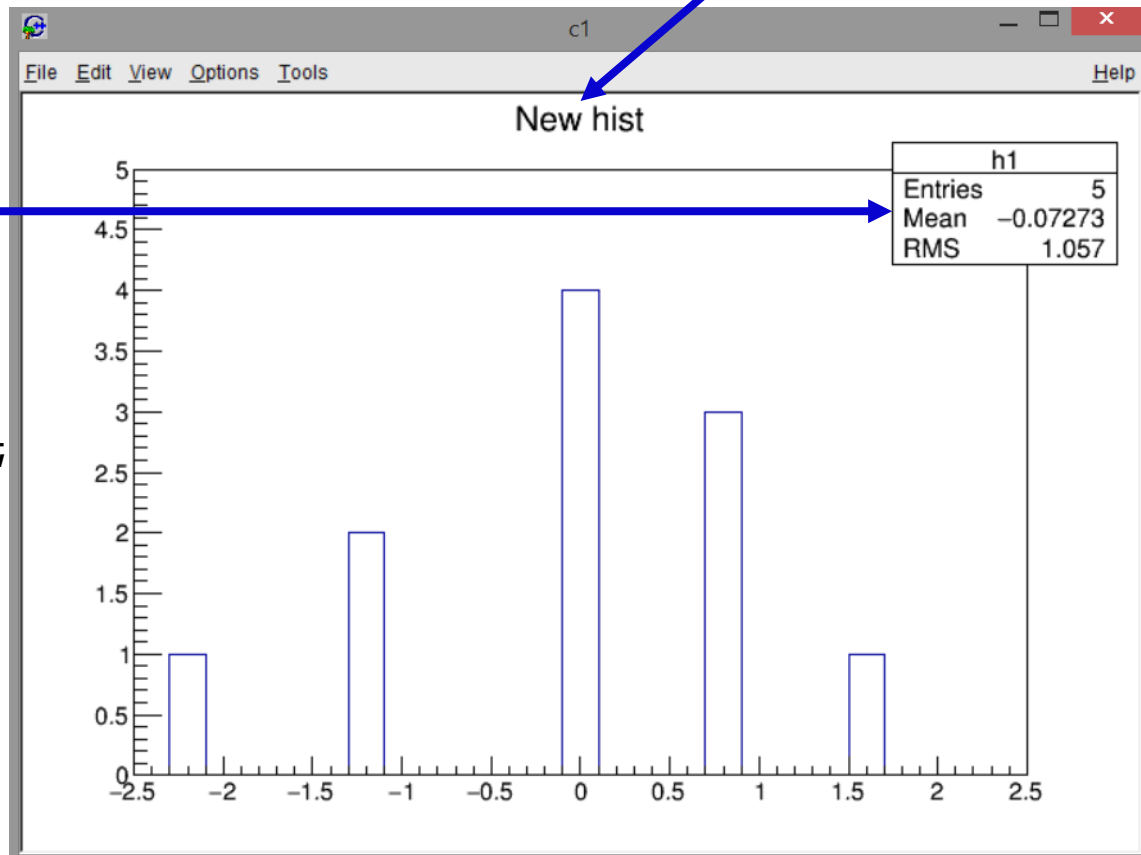
Entries - число вхождений;

Mean - среднее значение;

RMS - оценка стандартного отклонения.

С помощью параметра `SetOptStat` можно увеличить количество выводимой информации в этом окне

Заголовок гистограммы



ГИСТОГРАММЫ

- Методы `Add()`, `Divide()` и `Multiply()` позволяют складывать, делить и умножать гистограммы.

- Чтобы добавить к гистограмме h_1 гистограмму h_2

`h1->Add(h2)`

- Аналог, который не заменяет содержимое 1 из гистограмм:

`h3->Add(h1, h2, c1, c2) // this = c1*h1 + c2*h2`

h_1, h_2 – гистограммы, c_1 и c_2 – нормировочные коэффициенты.

- Можно добавить гистограмму с некоторым весом w

`h1->Add(h2,w)`

- Вычесть одну гистограмму из другой

`h1->Add(h2,-1)`

Складывать, делить и умножать можно только гистограммы с одинаковым числом бинов!

- Деление и умножение осуществляется аналогично:

`h1->Divide(h2)`

- Аналог, который не заменяет содержимое 1 из гистограмм:

`h3->Divide(h1, h2, c1, c2) // this = c1*h1/(c2*h2)`

`h1->Multiply(h2)`

`h3->Multiply(h1, h2, c1, c2) // this = (c1*h1)*(c2*h2)`

- Создать идентичную копию (клон) гистограммы:

`THIF *h1_clone = (THIF*)h1->Clone()`

ОБЛАСТЬ РИСОВАНИЯ «КАНВАС»

- Область, на которую выводятся графические объекты в ROOT, называется canvas (класс TCanvas):

```
TCanvas *c1 = new TCanvas ("c1", "canvas_new", 0.,0.,600,600);
```

Параметрами являются: название канваса, заголовок канваса и его размеры (координаты левой верхней и правой нижней точек). Вместо координат можно использовать просто параметр 1 и размер канваса будет автоматически задан.

- Объект, например, гистограмма, изображается на текущем активном canvas.
- Если canvas не существует, то он создается автоматически и имеет по умолчанию имя c1.
- Чтобы разделить canvas на несколько частей, можно воспользоваться методом `Divide(k,l)`, где k и l число разбиений по горизонтали и вертикали соответственно.
- Чтобы выбрать, на какой части canvas'a следует рисовать объект, следует применить метод `cd(n)`, где n — номер части. Разделы нумеруются слева направо, сверху вниз.

РАБОТА С ФАЙЛАМИ

- Чтобы сохранить гистограмму (или любые другие объекты) в файл, нужно прежде всего, создать файл (или открыть существующий)
- Работу с файлами обеспечивает класс **TFile**
Создание файла (объекта класса **TFile**):

```
TFile *f = new TFile("filehist.root", "new")
```

`filehist.root` - имя файла,

new или **create** - создать файл; **если файл с таким именем уже существует, он не будет открыт**

recreate - создать файл; **если файл с таким именем уже существует, он будет перезаписан**

update - открыть файл для записи; **если файла с таким именем не существует, он будет создан**

read - открыть файл для чтения (**по умолчанию**).

РАБОТА С ФАЙЛАМИ

- Файл после создания становится текущей ROOT-директорией
- Изначальная текущая директория - сессия ROOT
- Последний созданный файл есть текущая директория
- Глобальная переменная, указывающая на текущую директорию
`gDirectory`
- Показать текущую директорию
`gDirectory->pwd()` или `.pwd`
- Показать содержимое текущей директории
`gDirectory->ls()` или `.ls`

Сменить директорию `cd()`

Закрыть файл `Close()`

РАБОТА С ФАЙЛАМИ

- Пример

```
root [0] .pwd
Current directory: Rint:/
Current style:      Modern
root [1] TFile *file1=new TFile("file1.root","recreate")
(TFile *) 0xb828a30
root [2] .pwd
Current directory: file1.root:/
Current style:      Modern
root [3] TFile *file2=new TFile("file2.root","recreate")
(TFile *) 0xb7a7348
root [4] .pwd
Current directory: file2.root:/
Current style:      Modern
root [5] file1->cd()
(bool) true
root [6] .pwd
Current directory: file1.root:/
Current style:      Modern
root [7] file1->Close()
root [8] .pwd
Current directory: Rint:/
Current style:      Modern
```

ЗАПИСЬ В ФАЙЛ

- Чтобы осуществить запись гистограммы в файл, используется метод `Write()`
- Пример записи гистограммы:

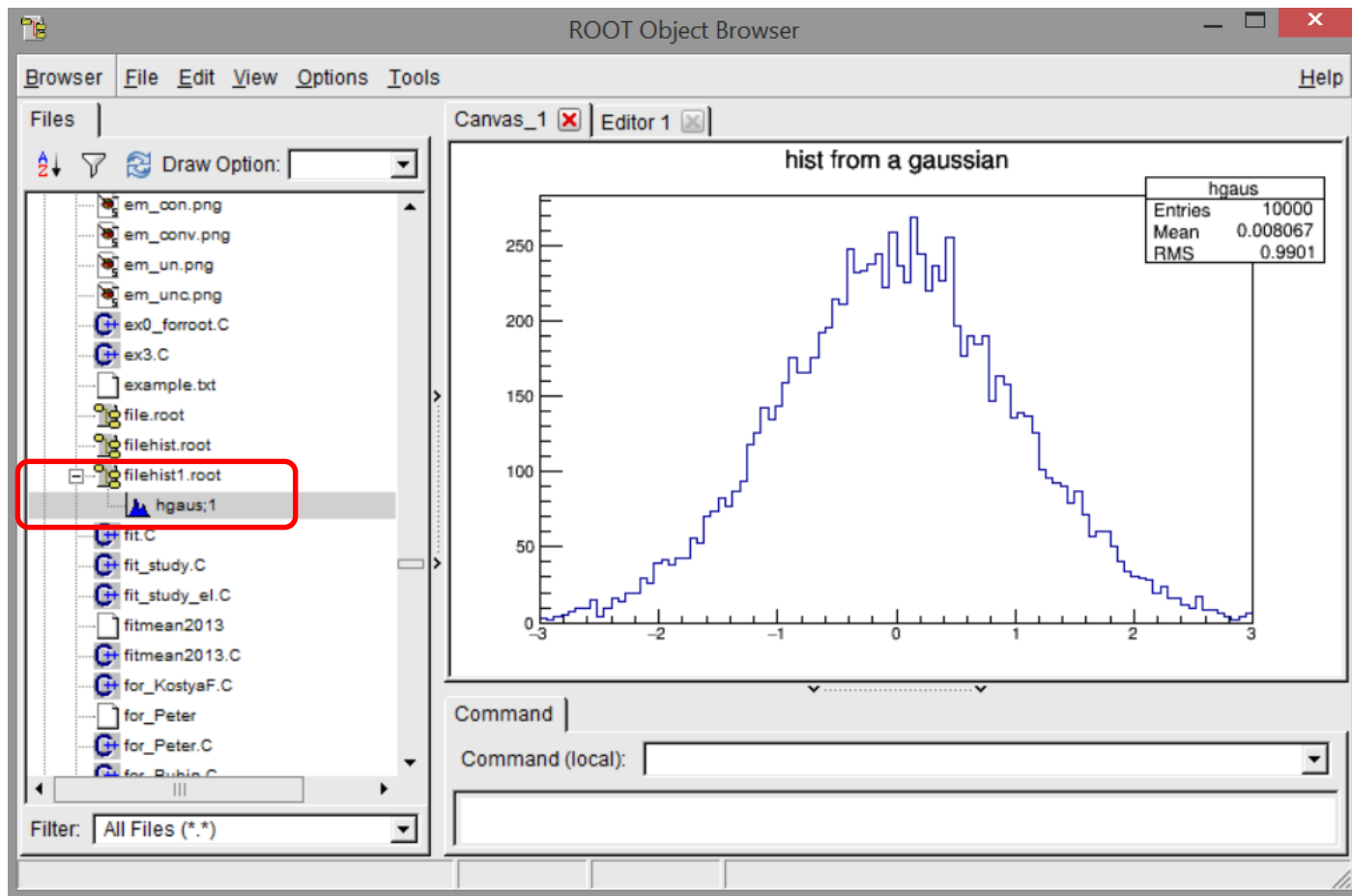
```
TFile *file = new TFile("filehist1.root", "new");  
TH1F *h1 = new TH1F("hgaus", "hist from a gaussian", 100, -3, 3);  
h1->FillRandom("gaus", 10000);  
h1->Write();
```

- Чтобы записать все объекты в текущей директории `file->Write()`

ПРОСМОТР ФАЙЛА В TBrowser

- Просмотреть содержимое .root файлов и объектов, записанных в них используется специальный браузер TBrowser.
- Чтобы запустить TBrowser, создайте объект этого типа в интерактивной сессии ROOT:

```
root [0] TBrowser b
```



The screenshot displays the ROOT Object Browser interface. On the left, a file tree shows a list of files, with 'filehist1.root' and its sub-object 'hgaus;1' highlighted by a red rectangle. The main canvas on the right shows a histogram titled 'hist from a gaussian'. The histogram has a y-axis from 0 to 250 and an x-axis from -3 to 3. A table in the top right corner of the canvas provides statistics for the 'hgaus' object:

hgaus	
Entries	10000
Mean	0.008067
RMS	0.9901

At the bottom of the window, there is a 'Command' field and a 'Command (local):' dropdown menu.

ЧТЕНИЕ ФАЙЛА

- Чтение гистограммы (объекта):
- Метод `Get("name")` возвращает указатель на объект с именем `name`
- Пример:

```
// открываем файл для чтения
TFile *file = new TFile("filehist1.root");
// берем указатель на hgaus (идентификатор объекта, который мы
видим в TBrowser
TH1F *h1 = (TH1F*)file->Get("hgaus");
// Рисуем гистограмму
h1->Draw();
```

В коде необходимо сделать **cast** к нужному типу.

