



Лекции. Практические занятия

Солдатов Е.Ю.

2025 г.

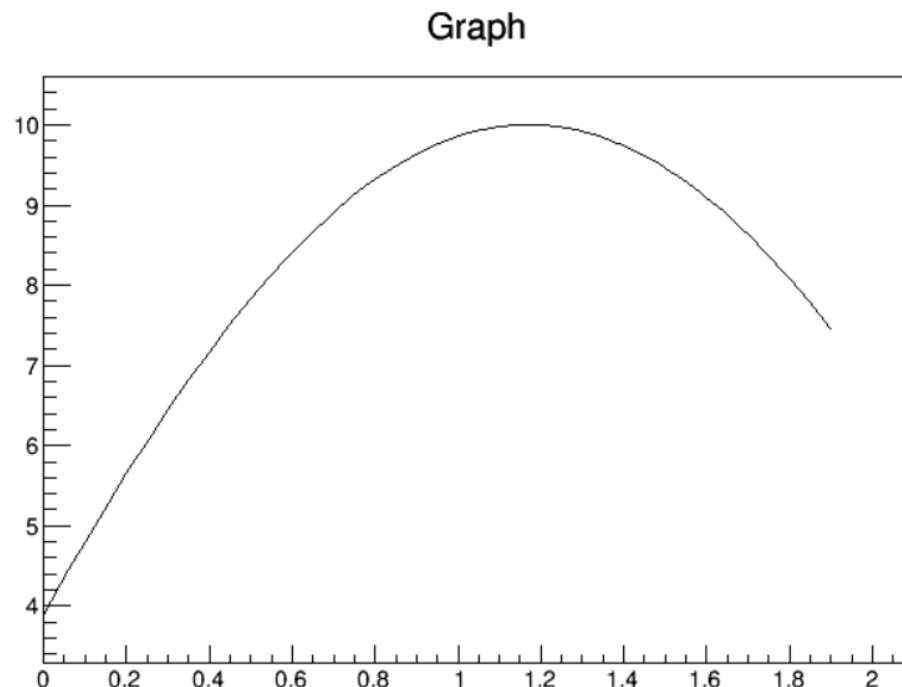
ГРАФИКИ

- Работа с графиками обеспечивается классом TGraph
- Для создания графика нужно определить два массива, содержащих n значений абсцисс и ординат точек.

Пример.

График функции $y(x) = 10 \cdot \sin(x + 0.4)$

```
Int_t n = 20;  
Double_t x[20], y[20];  
for (Int_t i=0; i<n; i++)  
{  
    x[i] = i*0.1;  
    y[i] = 10*sin(x[i]+0.4);  
}  
TGraph *gr1 = new TGraph(n, x, y);  
// Рисование графика  
gr1->Draw("ACP");
```



ГРАФИКИ

- Различные опции рисования для графиков
 - A** – нарисовать координатные оси
 - L** – провести через точки ломаную
 - C** – провести плавную кривую через точки графика
 - *** - нарисовать значок звездочки в каждой точке
 - P** – нарисовать маркер текущего стиля в каждой точке
- Изменить стиль маркера можно командой `grl->SetMarkerStyle(N)`
- Изменить цвет маркера `grl->SetMarkerColor(M)`
- Изменить цвет соединяющей линии `grl->SetLineColor(M)`
- Изменить ширину соединяющей линии `grl->SetLineWidth(K)`
- Изменить стиль соединяющей линии `grl->SetLineStyle(L)`
- Свойства линии и маркера можно редактировать в режиме GUI, кликнув правой кнопкой мыши по кривой и выбрав `SetLineAttributes`.

ГРАФИКИ

- Таблица цветов ROOT:

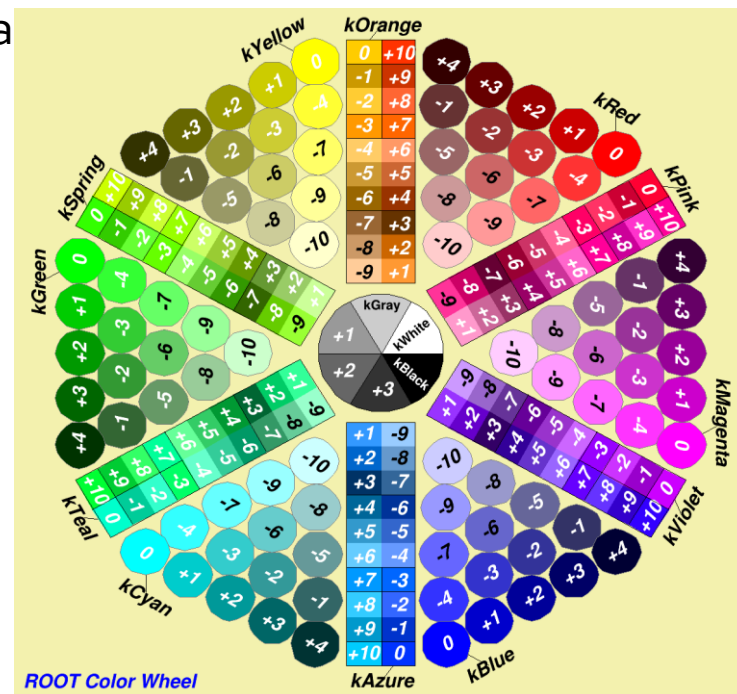
<https://root.cern.ch/doc/master/classTColor.html>

40	41	42	43	44	45	46	47	48	49
30	31	32	33	34	35	36	37	38	39
20	21	22	23	24	25	26	27	28	29
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

- Цвета можно задавать из т.н. «колеса цветов» с помощью констант

```
myObject->SetFillColor(kRed);
myObject->SetFillColor(kYellow-10);
myObject->SetLineColor(kMagenta+2);
```

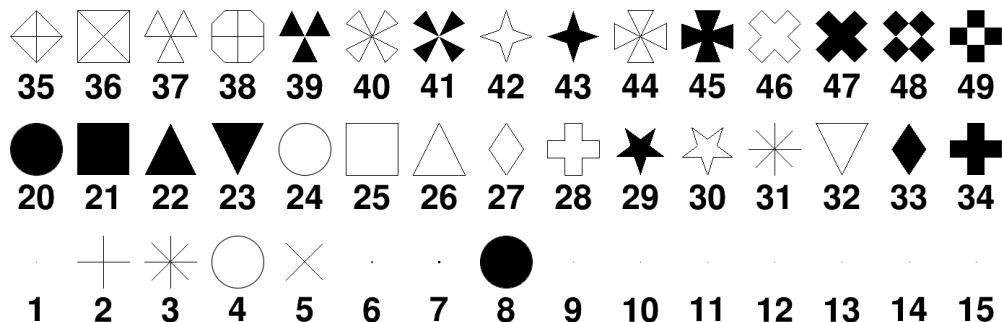
- Цвета можно задавать кодами RGB
- ```
TColor *color;
hist->SetLineColor(color->GetColor("#C0C0C0"));
```



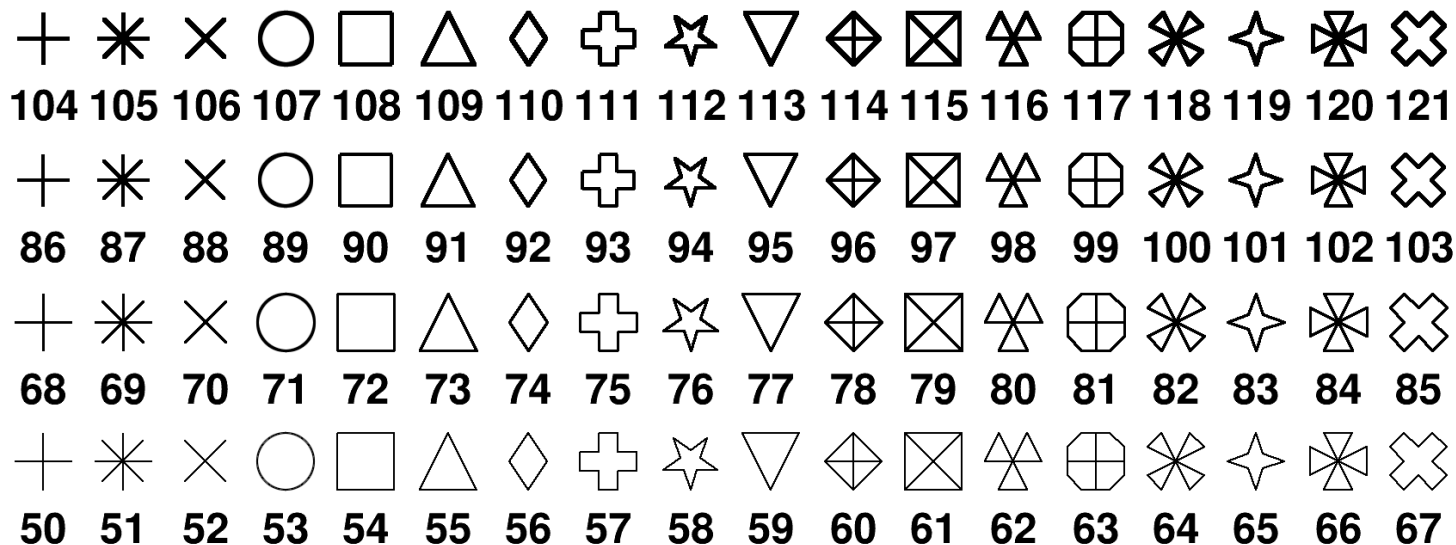
# ГРАФИКИ

- Таблица стилей маркеров

<https://root.cern.ch/doc/master/classTAttMarker.html>



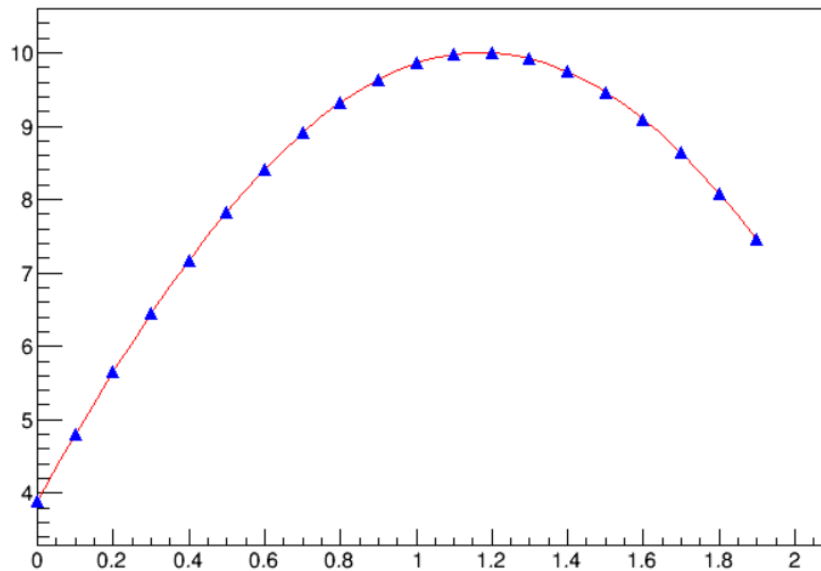
- Ещё стили маркеров:



# ГРАФИКИ

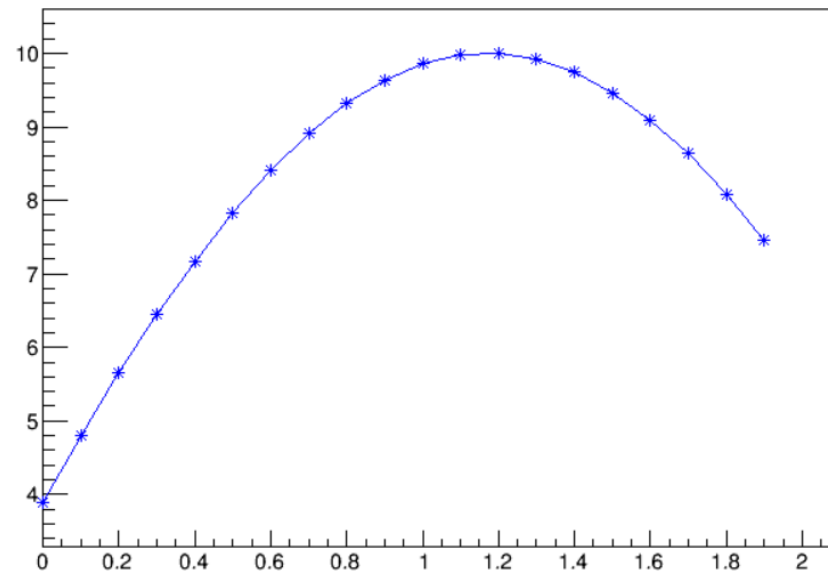
```
gr l->SetMarkerStyle(22)
gr l->SetMarkerColor(4)
gr l->SetLineColor(2)
gr l->Draw("ACP")
```

Graph



```
gr l->SetLineColor(4)
gr l->SetMarkerColor(4)
gr l->Draw("AL*")
```

Graph

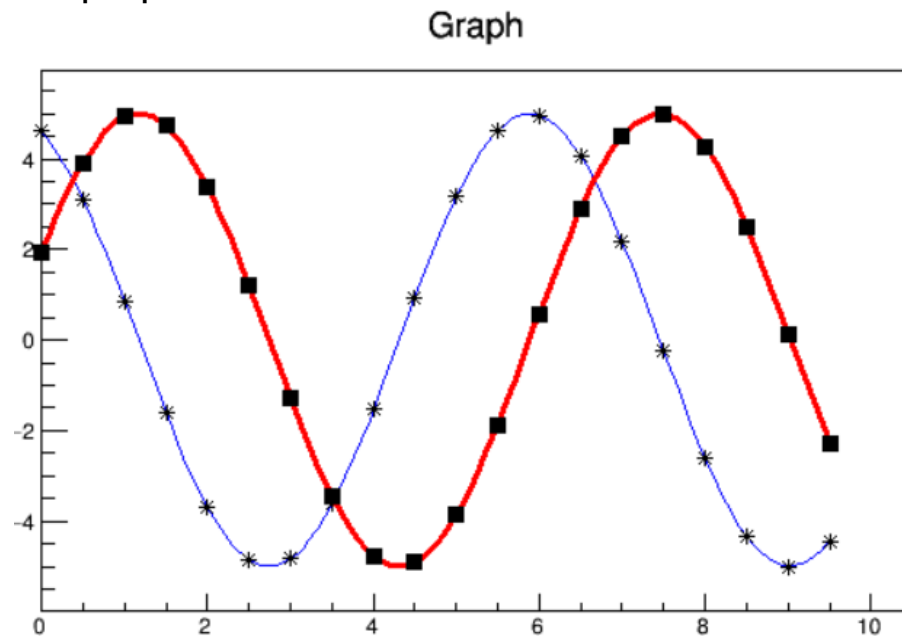


# ГРАФИКИ

- Чтобы нарисовать два графика на одном и том же canvas, следует опустить параметр "A" у второго графика.

Пример скрипта:

```
{
 Int_t n = 20;
 Double_t x[n], y[n], x1[n], y1[n];
 for (Int_t i=0; i<n; i++) {
 x[i] = i*0.5;
 y[i] = 5*cos(x[i]+0.4);
 x1[i] = i*0.5;
 y1[i] = 5*sin(x[i]+0.4);
 }
 TGraph *gr1 = new TGraph(n,x,y);
 TGraph *gr2 = new TGraph(n,x1,y1);
 TCanvas *c1 = new TCanvas("c1","Two Graphs",200,10,600,400);
 gr1->SetLineColor(4);
 gr1->Draw("AC*");
 gr2->SetLineWidth(3);
 gr2->SetMarkerStyle(21);
 gr2->SetLineColor(2);
 gr2->Draw("CP");
}
```



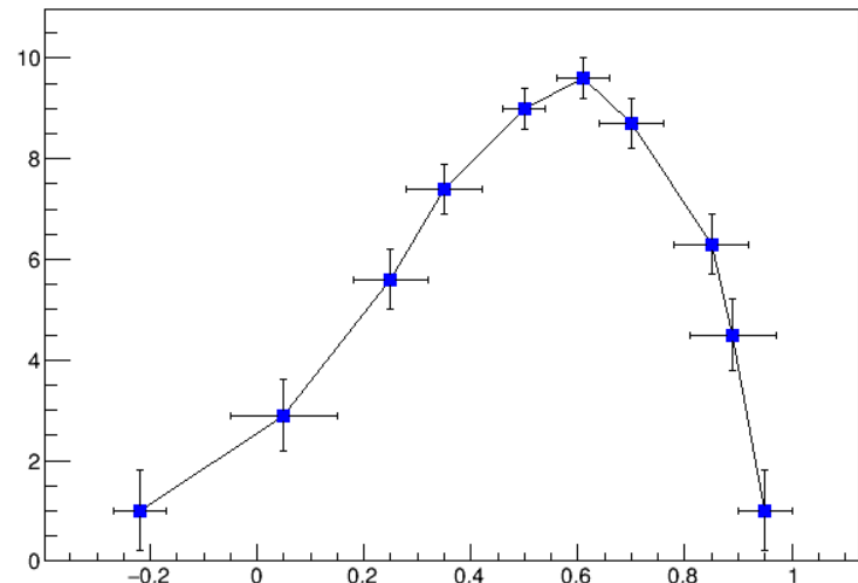
# ГРАФИКИ С ПОГРЕШНОСТЯМИ

- Для графиков с погрешностями используется класс TGraphErrors

Пример скрипта:

```
{
Int_t n = 10;
Float_t x[n] = {-0.22, 0.05, 0.25, 0.35, 0.5, 0.61, 0.7, 0.85, 0.89, 0.95};
Float_t y[n] = {1, 2.9, 5.6, 7.4, 9, 9.6, 8.7, 6.3, 4.5, 1};
Float_t ex[n] = {0.05, 0.1, 0.07, 0.07, 0.04, 0.05, 0.06, 0.07, 0.08, 0.05};
Float_t ey[n] = {0.8, 0.7, 0.6, 0.5, 0.4, 0.4, 0.5, 0.6, 0.7, 0.8};
TGraphErrors *gr= new TGraphErrors(n,x,y,ex,ey);
gr->SetTitle("TGraphErrorsExample");
gr->SetMarkerColor(4);
gr->SetMarkerStyle(21);
gr->Draw("ALP");
}
```

TGraphErrorsExample



# ГРАФИКИ С АСИММЕТРИЧНЫМИ ПОГРЕШНОСТЯМИ

- Для графиков с погрешностями используется класс TGraphAsymmErrors

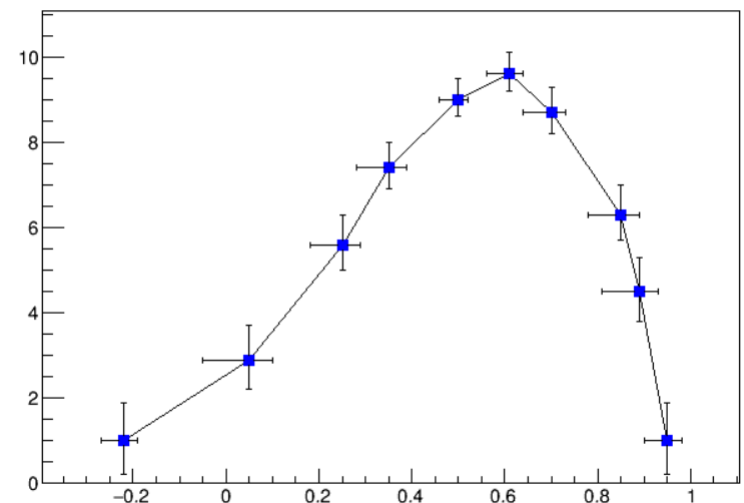
Пример скрипта:

```
{
 Int_t n = 10;
 Float_t x[n] = {-0.22, 0.05, 0.25, 0.35, 0.5, 0.61, 0.7, 0.85, 0.89, 0.95};
 Float_t y[n] = {1, 2.9, 5.6, 7.4, 9, 9.6, 8.7, 6.3, 4.5, 1};
 Float_t exup[n] = {0.05, 0.1, 0.07, 0.07, 0.04, 0.05, 0.06, 0.07, 0.08, 0.05};
 Float_t eyup[n] = {0.8, 0.7, 0.6, 0.5, 0.4, 0.4, 0.5, 0.6, 0.7, 0.8};
 Float_t exdown[n] = {0.03, 0.05, 0.04, 0.04, 0.02, 0.03, 0.03, 0.04, 0.04, 0.03};
 Float_t eydown[n] = {0.9, 0.8, 0.7, 0.6, 0.5, 0.5, 0.6, 0.7, 0.8, 0.9};
 TGraphAsymmErrors *gr = new
 TGraphAsymmErrors(n, x, y, exup, exdown, eyup, eydown);
 gr->SetTitle("TGraphAsymmErrorsEx");
 gr->SetMarkerColor(4);
 gr->SetMarkerStyle(21);
 gr->Draw("ALP");
}
```

Ref. manual:

<https://root.cern.ch/doc/master/classTGraphAsymmErrors.html>

TGraphAsymmErrorsExample



# ГРАФИКИ ИЗ ФАЙЛА

Если у вас есть текстовый массив x-ов и y-ов, то можно воспользоваться следующим:

`TGraphErrors("filename", format, option)`

`filename` – имя файла, откуда берём данные

`format`:

`"%lg %lg"` – считать только первые 2 столбца в x и y

`"%lg %lg %lg"` – считать только первые 3 столбца в x, y, y\_error

`"%lg %lg %lg %lg"` – считать только первые 4 столбца в x, y, x\_error, y\_error

Разделителем может быть ‘ ‘ или `\t` или ;

Можно настраивать разделитель, но это замедлит процесс.

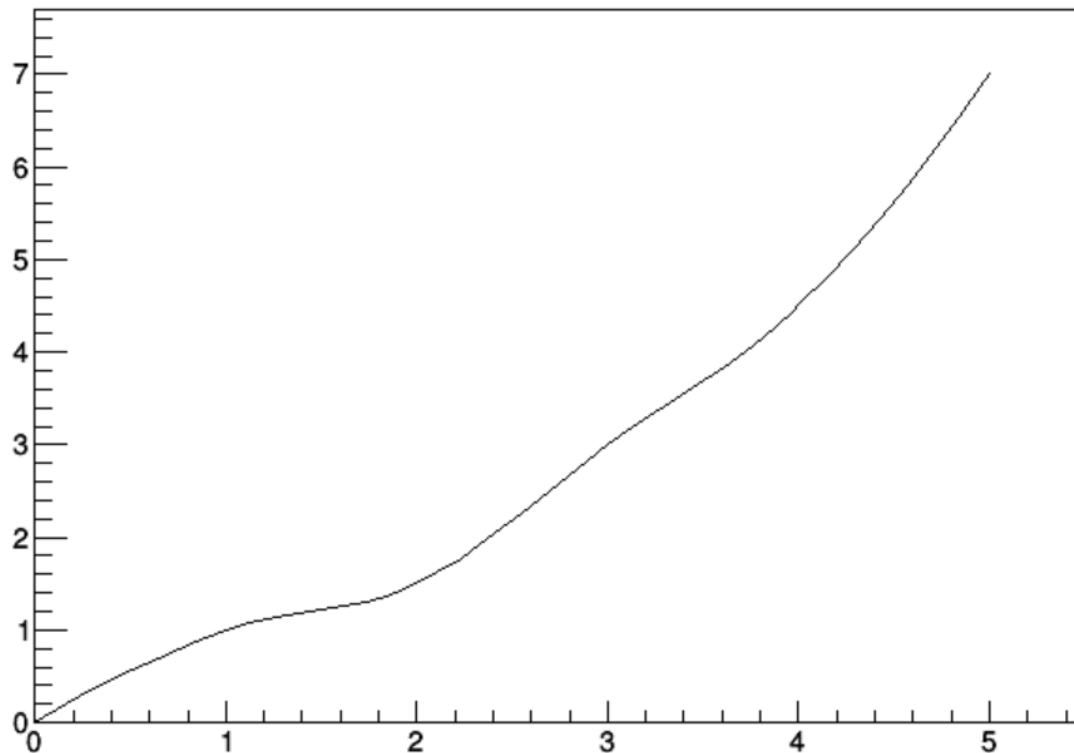
Таким образом можно повторно использовать один макрос для множества различных наборов данных.

# ГРАФИКИ С ДРУГИМ ЗАПОЛНЕНИЕМ

```
{
TGraph* gr = new TGraph(5);
gr->AddPoint(1,1);
gr->AddPoint(2,1.5);
gr->AddPoint(3,3);
gr->AddPoint(4,4.5);
gr->AddPoint(5,7);
gr->Draw("AC");
}
```

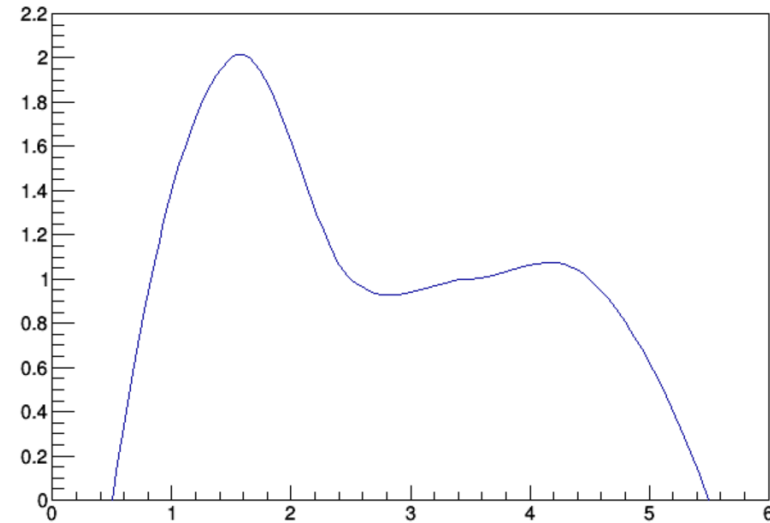
Существует только для класса  
TGraph

Graph

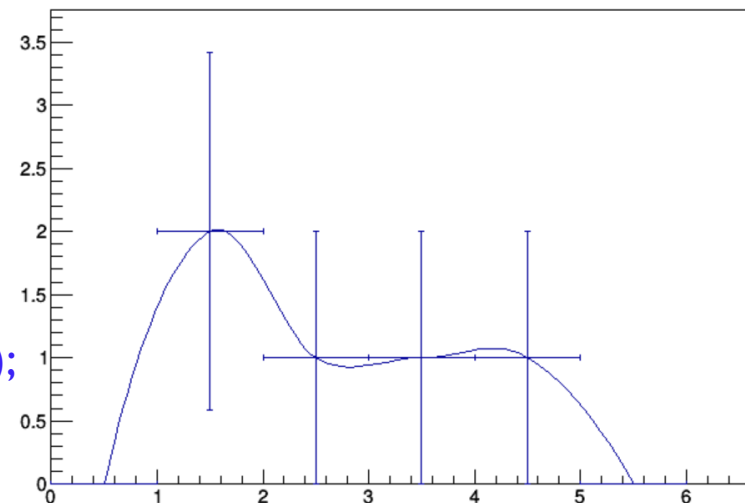


# ГРАФИКИ ИЗ ГИСТОГРАММ

```
{
 TH1F* hist = new TH1F("hist","hist",6,0,6);
 hist->Fill(1.5);
 hist->Fill(1.9);
 hist->Fill(2.1);
 hist->Fill(3.9);
 hist->Fill(4.4);
 TGraph* gr = new TGraph(hist);
 gr->Draw("AC");
}
```



```
{
 TH1F* hist = new TH1F("hist","hist",6,0,6);
 hist->Fill(1.5);
 hist->Fill(1.9);
 hist->Fill(2.1);
 hist->Fill(3.9);
 hist->Fill(4.4);
 TGraphErrors* gr = new TGraphErrors(hist);
 gr->Draw("AC");
}
```



## ГРАФИКИ: ЗАДАЧА

Дана гистограмма:

Биннинг (границы бинов): 0, 1, 2, 3, 5, 10

Значения: 1560, 708, 338, 148, 102

Веса при заполнении не применены. Изображение ошибок включено, центры бинов соединены. Изобразите.

Сделайте то же самое с помощью графиков с ошибками.

Картинки должны быть идентичными.