

# Histogram

1. V ROOTu vytvořte histogram z naměřených hodnot uložených v souboru `data.dat`.  
Nalezněte optimální šířku binu.

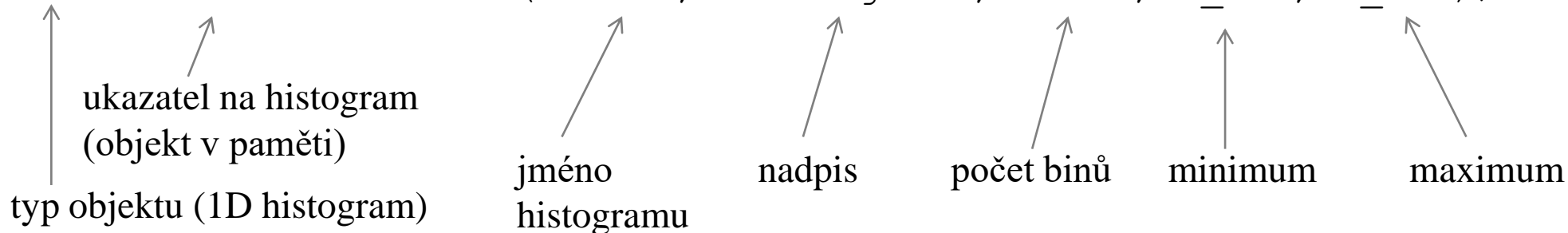
# Histogram

1. V ROOTu vytvořte histogram z naměřených hodnot uložených v souboru `data.dat`  
Nalezněte optimální šířku binu.

Práce s histogramy v ROOTu:

vytvoření histogramu (constructor)

```
TH1D *hist = new TH1D("hist", "histogram", nbins, x_min, x_max);
```



metody objektu THD:

přidání prvku  $x_i$  do histogramu

```
hist -> Fill(x[i]);
```

nakreslení histogramu

```
hist -> Draw();
```

# Histogram

histogram.c

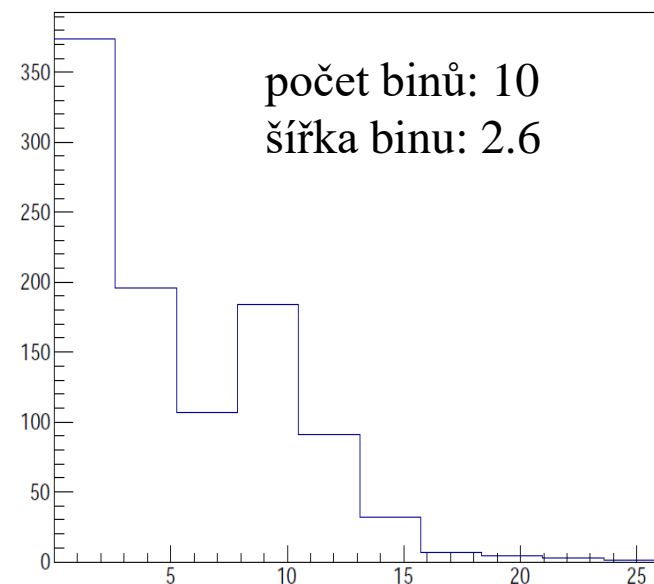
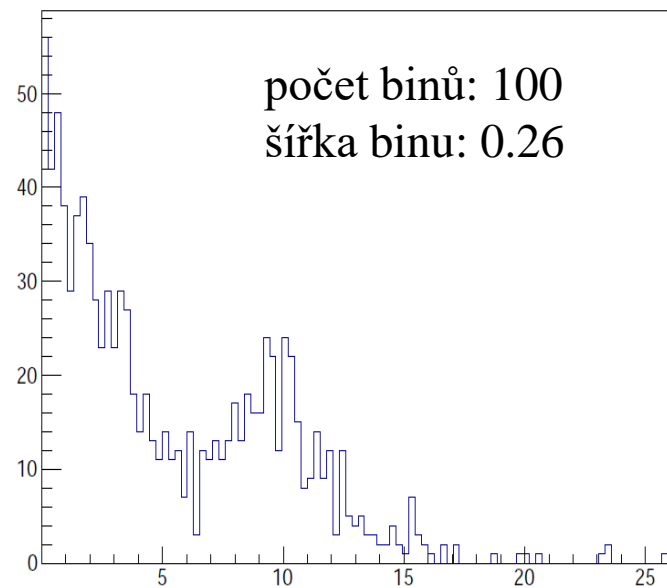
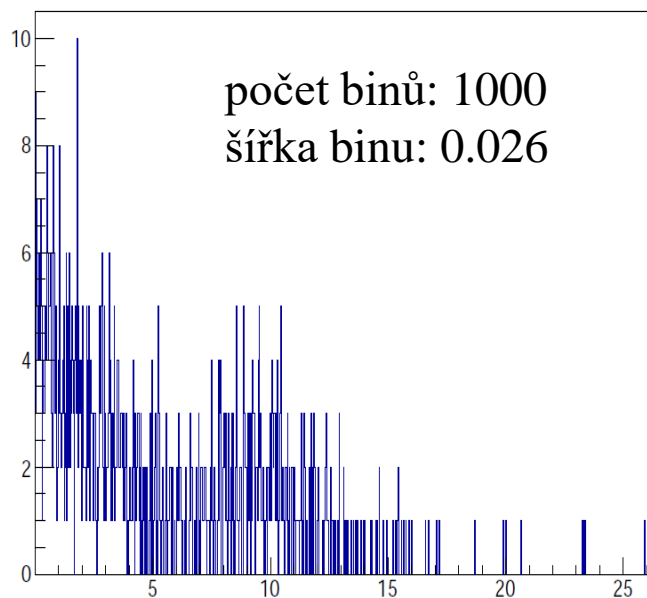
```
//histogram
#define ndata 1000 //pocet dat
#define nbins 100 //pocet binu
int run()
{
    FILE *fin,*fout; //soubor odkud se prectou a kam se zapisi data
    double x[ndata];
    double x_min,x_max; //minimalni a maximalni hodnota - meze histogramu
    fin=fopen("data.dat","r"); //vstupni soubor s daty
    fout=fopen("histogram.txt","w"); //vstupni soubor s daty
    for (int i = 0; i < ndata; ++i)
    {
        fscanf(fin,"%lf",&x[i]); //nacte data
        if(i==0) x_min=x_max=x[i]; //aby nebyly nedefinovane
        if(x[i]<x_min) x_min=x[i];
        if(x[i]>x_max) x_max=x[i];
    }
    fclose(fin);
    printf("min value: %lf\n",x_min);
    printf("max value: %lf\n",x_max);
    printf("bin width: %lf\n", (x_max-x_min)/nbins);
    TH1D *hist = new TH1D("hist", "histogram", nbins, x_min, x_max);
    for (int i = 0; i < ndata; ++i) hist->Fill(x[i]); //naplneni histogramu
    TCanvas *c1 = new TCanvas("c1", "histogram",5,5,800,600); //definice okna
    hist->Draw(); //nakresli histogram
    for (int i = 0; i < nbins; ++i) //zapis histogramu do souboru fout
    {
        counts=hist->GetBinContent(i); //zjisti hodnoty v i-tem binu
        fprintf(fout,"%ld %ld\n",i,counts);
    }
    fclose(fout);
    return(0);
}
```

# Histogram

1. V ROOTu vytvořte histogram z naměřených hodnot uložených v souboru `data.dat`  
Nalezněte optimální šířku binu.

počet dat: 1000  
minimum: 0.004  
maximum: 26.198

`histogram.c`



# Histogram + Monte Carlo simulace

2. Hodnoty v souboru `data.dat` odpovídají situaci kdy v experimentu mohou být detekovány dva typy událostí:

- v  $\frac{3}{4}$  případů exponenciální rozpad s časovou konstantou 4
- v  $\frac{1}{4}$  případů výběr z normálního rozdělení s očekávanou hodnotou 10 a standardní odchylkou 2.

V ROOTu proveďte simulaci tohoto experimentu a nasimulovaná uložte do histogramu

Generátor náhodných čísel v ROOTu:

vytvoření generátoru náhodných čísel (constructor objektu)    Perioda  $2^{19937}-1$

```
TRandom3 *gRandom = new TRandom3();
```

↑  
ukazatel na histogram  
(objekt v paměti)

↑  
typ objektu (generátor náhodných čísel)

generování náhodného čísla  $U(0,1)$

```
x[i] = gRandom->Rndm();
```

Další rozdělení:

```
gRandom->Exp(tau)
```

```
gRandom->Gaus(mean, sigma)
```

```
gRandom->Poisson(mean)
```

```
gRandom->Binomial(n, p)
```

# Histogram + Monte Carlo simulace

```
//generator nahodnych cisel z rozdeleni exp + norm
#define nmax 1000 //kolik hodnot
#define tau 4 //parametr exponencialniho rozdeleni (doba zivota)
#define mu 10 //ocekavana hodnota Normalniho rozdeleni
#define sigma 2 //standardni odchylka normalniho rozdeleni
#define P_exp 0.75 //pravdepodobnost ze to bude exponencialni rozdeleni
#define nbins 100 //pocet binu
int simulate()
{
    FILE *fout; //soubor kam se zapisi data
    double x[nmax];
    double branch;
    double x_min, x_max; //minimalni a maximalni hodnota - meze histogramu
    TRandom3 *gRandom = new TRandom3(); //vytvori generator nahodnych cisel
    fout=fopen("data.dat","w"); //otevreni souboru
    for (int i = 0; i < nmax; ++i)
    {
        branch=gRandom->Rndm();
        if(branch<P_exp) x[i]=gRandom->Exp(tau); //generovani nahodneho cisla E(tau)
        else x[i]=gRandom->Gaus(mu,sigma); //generovani nahodneho cisla N(mu,sigma)
        if(i==0) x_min=x_max=x[i]; //aby nebyly nedefinovane
        if(x[i]<x_min) x_min=x[i];
        if(x[i]>x_max) x_max=x[i];
        fprintf(fout,"%lf\n",x[i]); //zapis dat do souboru
    }
    printf("min value: %lf\n",x_min);
    printf("max value: %lf\n",x_max);
    TH1D *hist = new TH1D("data", "exp+gauss", nbins, x_min, x_max);
    for (int i = 0; i < nmax; ++i) hist->Fill(x[i]); //naplneni histogramu
    TCanvas *c1 = new TCanvas("c1", "histogram",5,5,800,600); //definice okna
    hist->Draw(); //nakresli histogram
    fclose(fout);
    return(0);
}
```