ROOT – Introduction

Rafał Staszewski

IFJ PAN PPSS 2020

Contents

Why statistical analysis in particle physics?

Makroscopic world (classical)

- Deterministic
- Known initial conditions (positions and momenta) $\rightarrow$ known future

Why statistical analysis in particle physics?

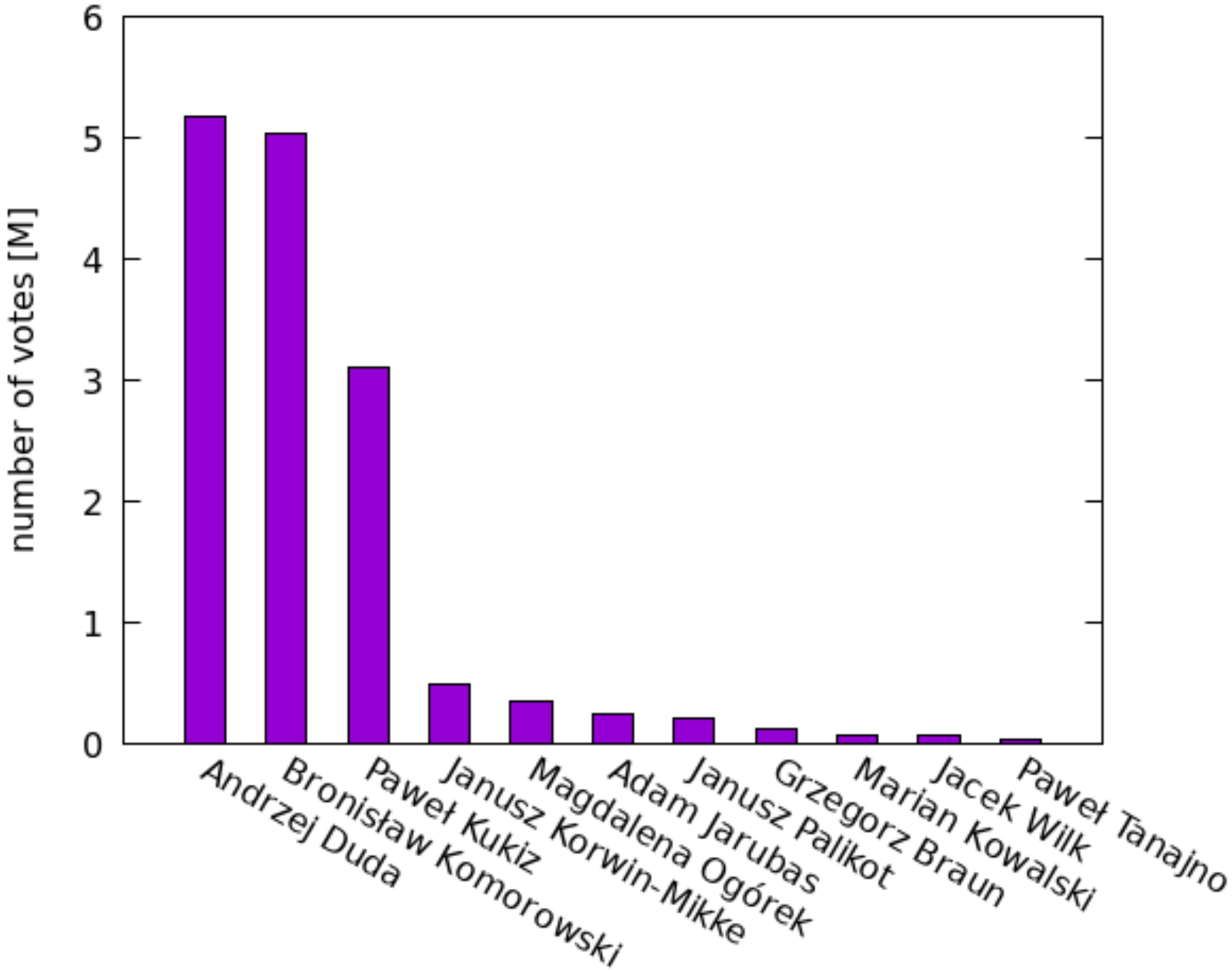
Makroscopic world (classical)

- Deterministic
- Known initial conditions (positions and momenta) $\rightarrow$ known future

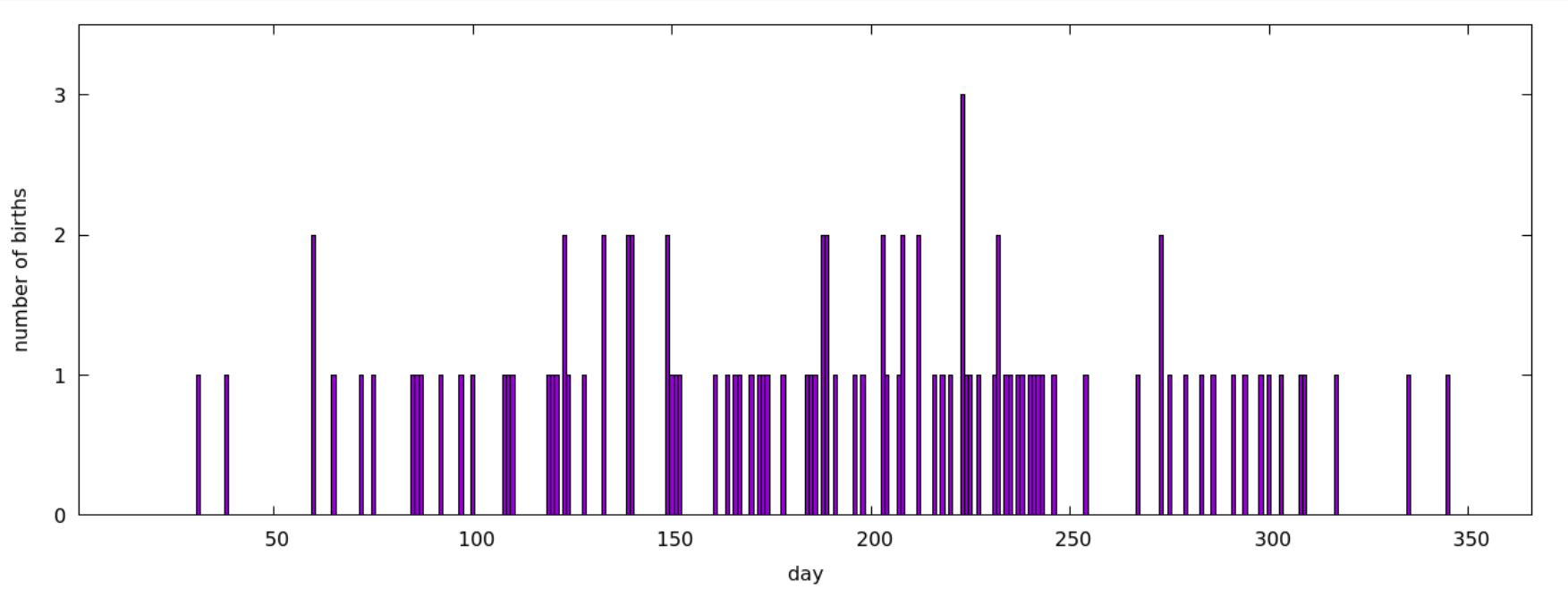
Mikroscopic world (quantum)

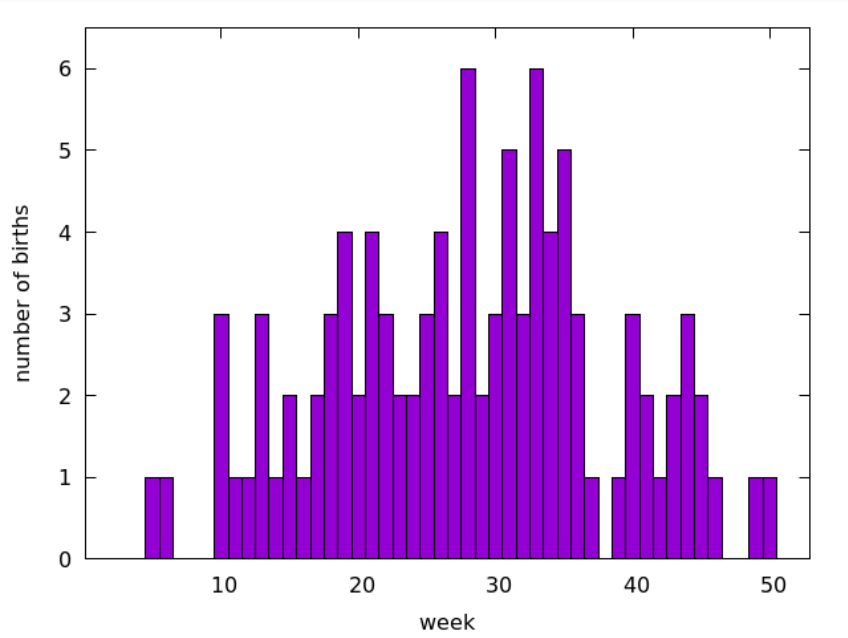
- Research tool – interactions (collisions) of particles
- Proton size: 1 fm = 10^{-15} m = 10^{-12} mm
- Electron size:: smaller than 1/1000 protonu (only limit!)
- Initial conditions not fully under control
- Quantum randomness
- Results of a single interaction not reproducible
- **We measure probabilities of different events**

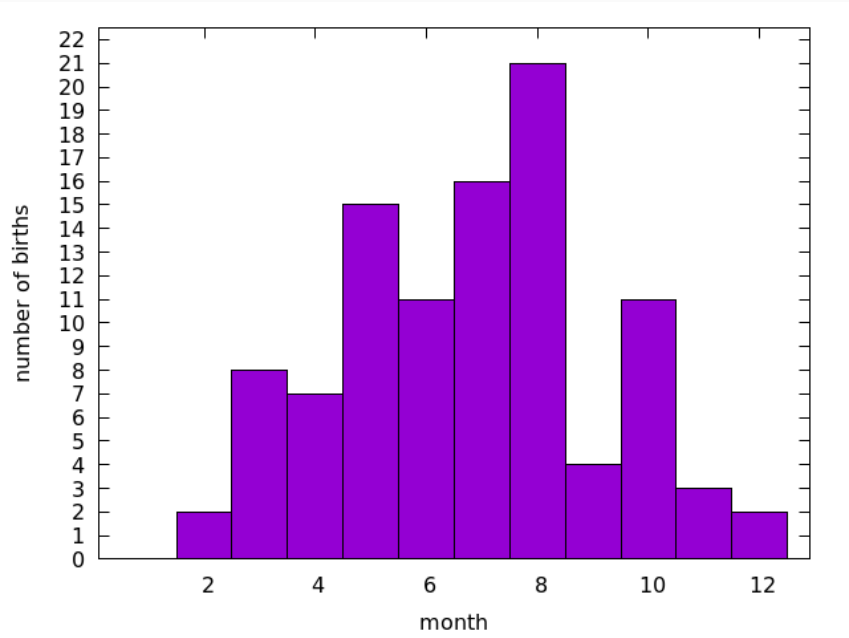
Statistical distribution



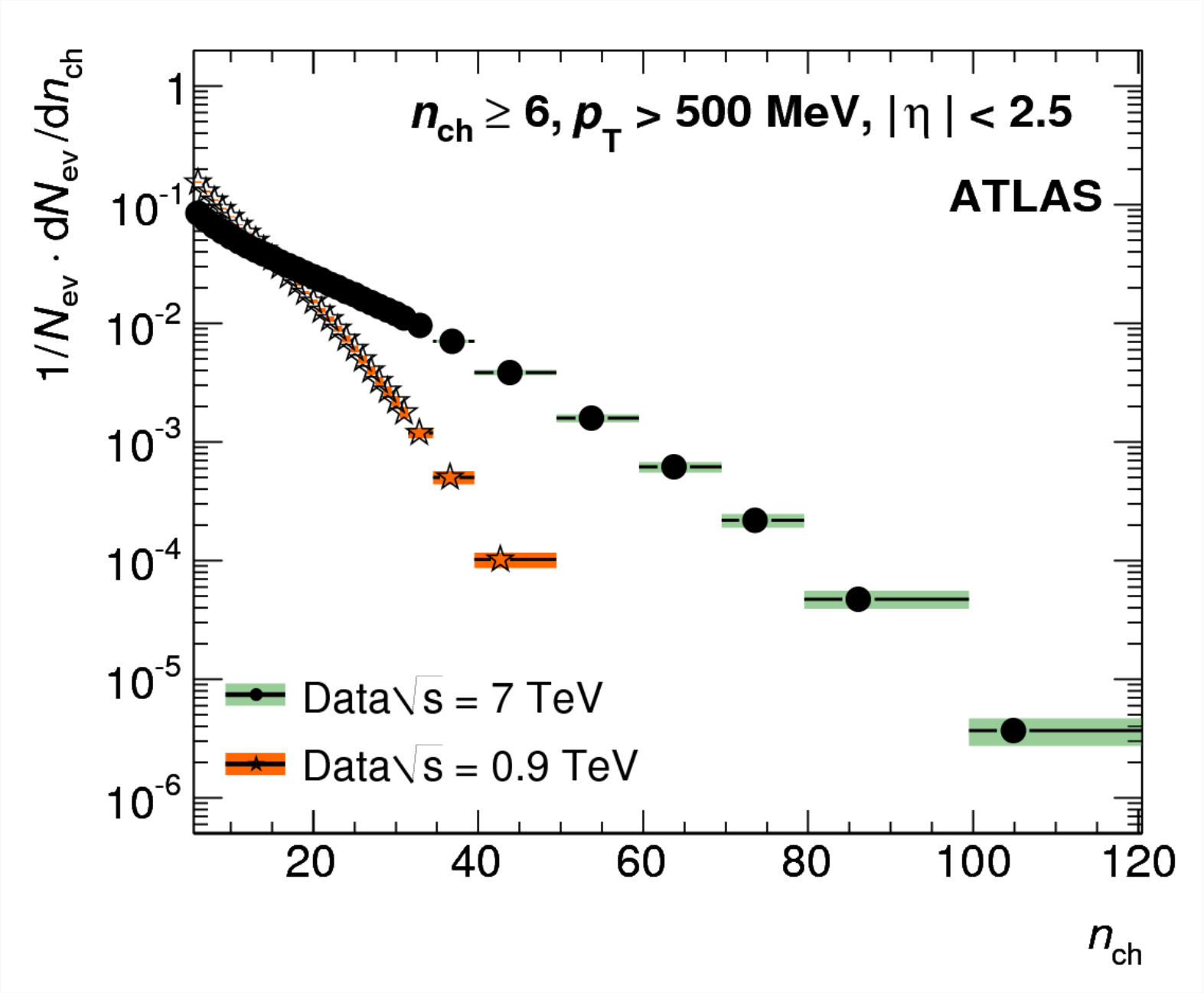
Histogram binning



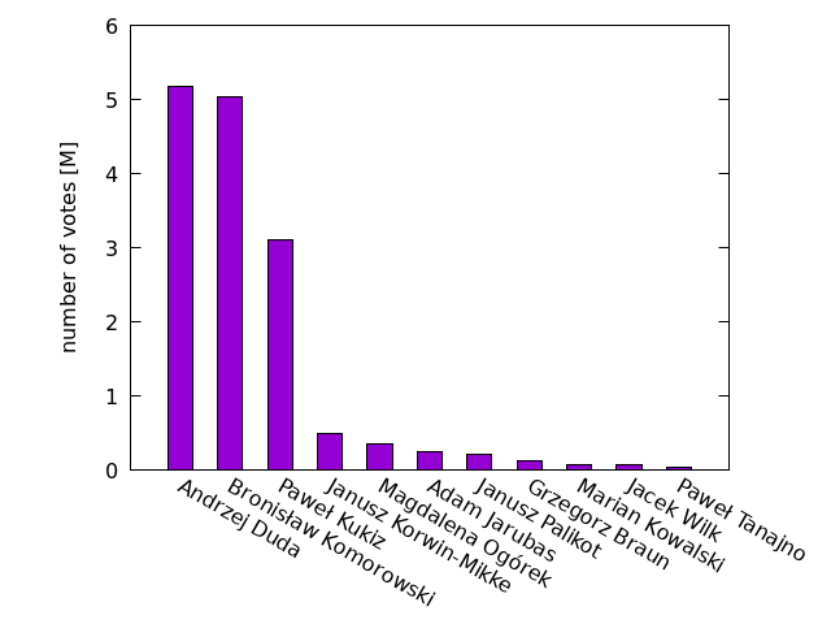




Example from particle physics



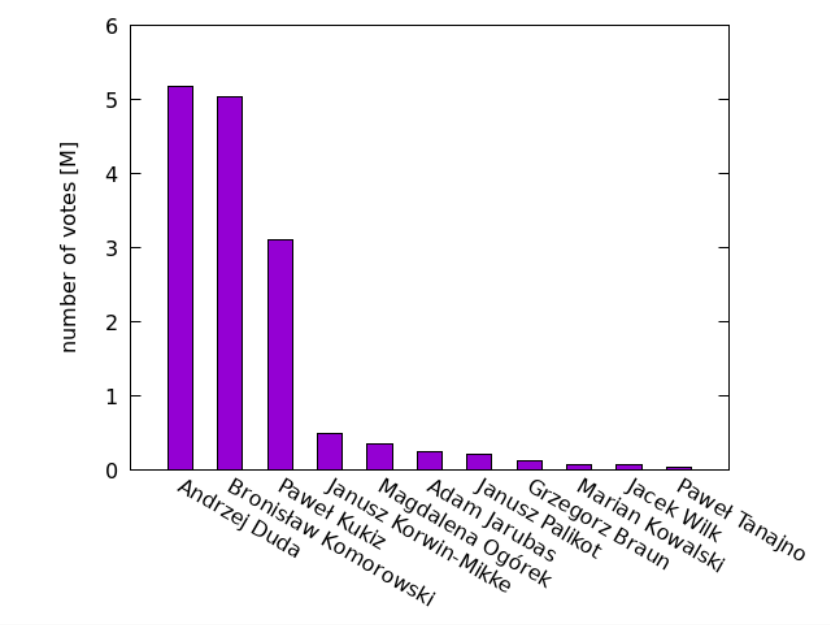
Logarithmic scale

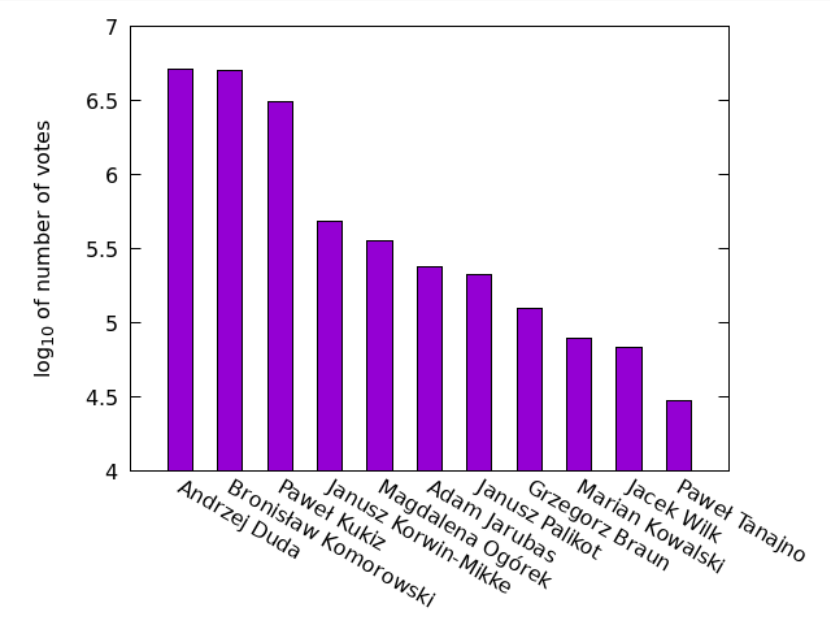


Problem:

- Linear scale appropriate for relatively flat distributions
- ”Tails” not well visible

Logarithmic scale



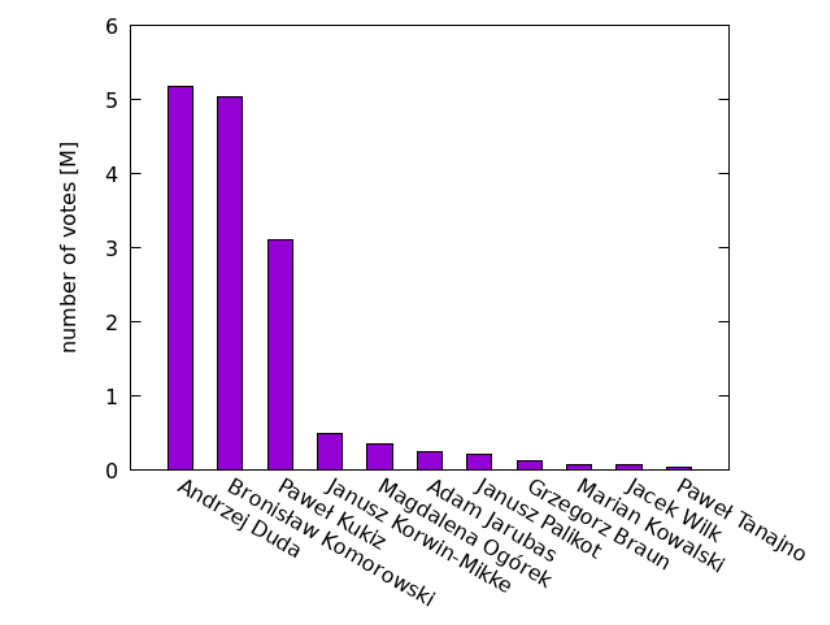


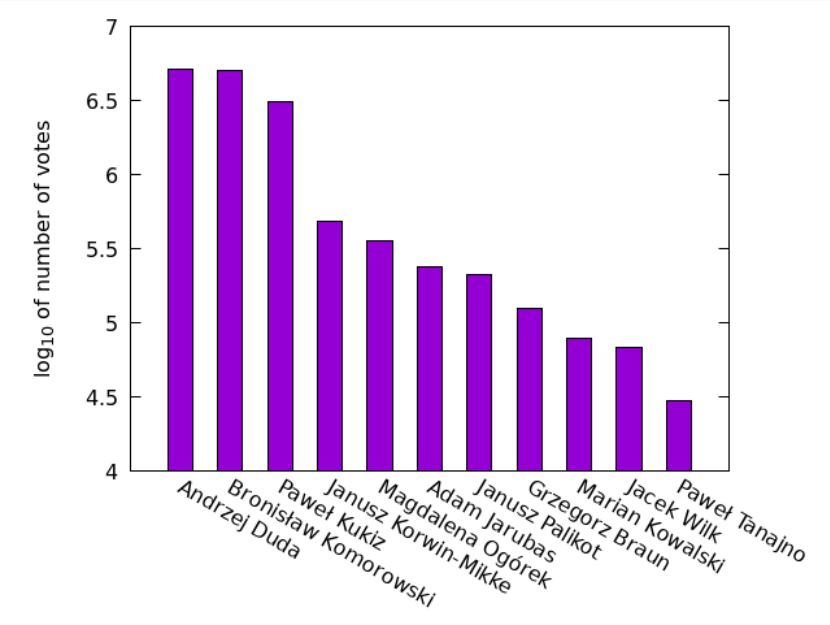
Problem:

- Linear scale appropriate for relatively flat distributions
- ”Tails” not well visible

Solution:

Logarithmic scale





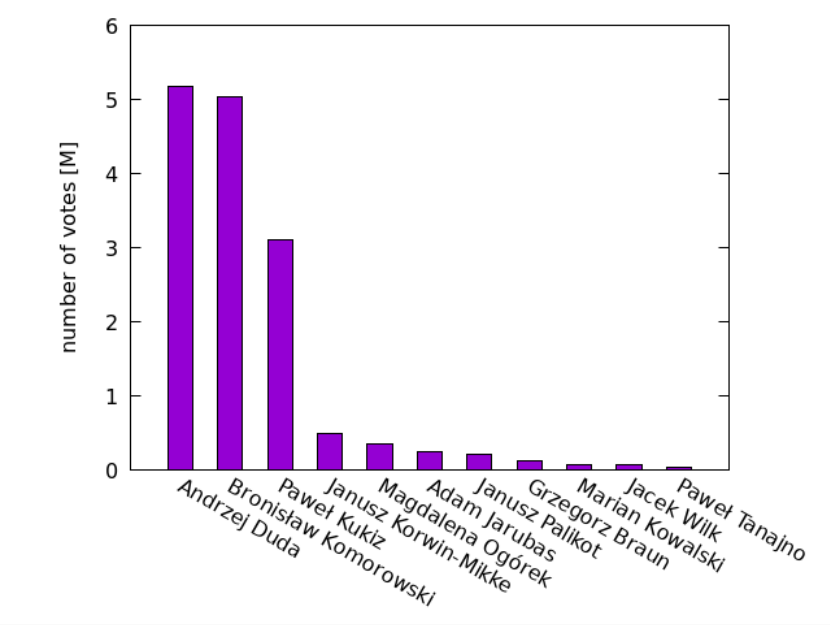
Problem:

- Linear scale appropriate for relatively flat distributions
- ”Tails” not well visible

Solution:

- Logarithm of number of votes

Logarithmic scale

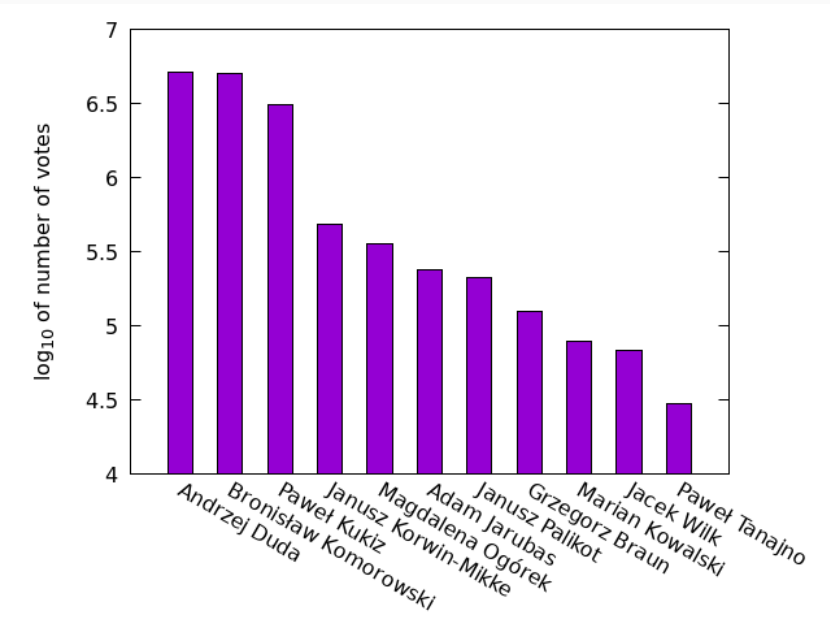


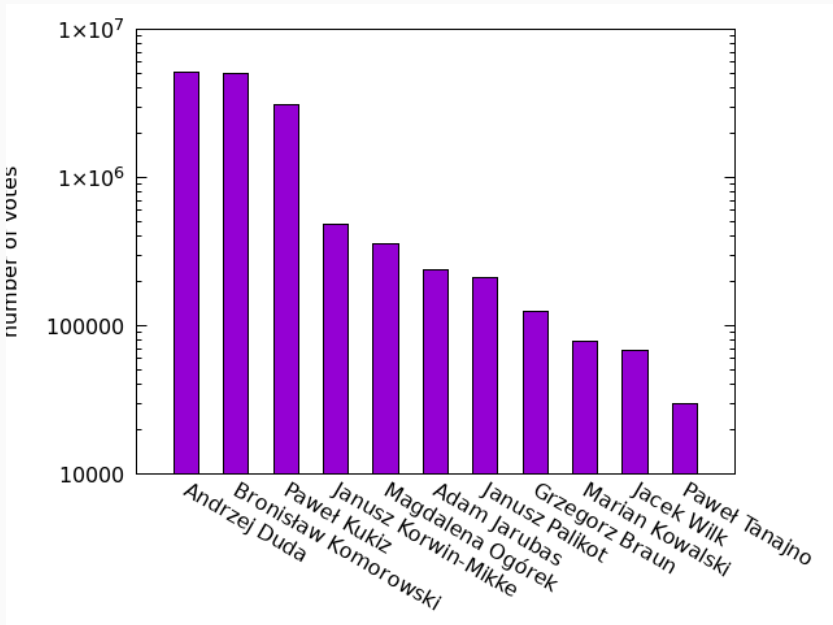
Problem:

- Linear scale appropriate for relatively flat distributions
- ”Tails” not well visible

Solution:

- Logarithm of number of votes
- Logarithmic scale



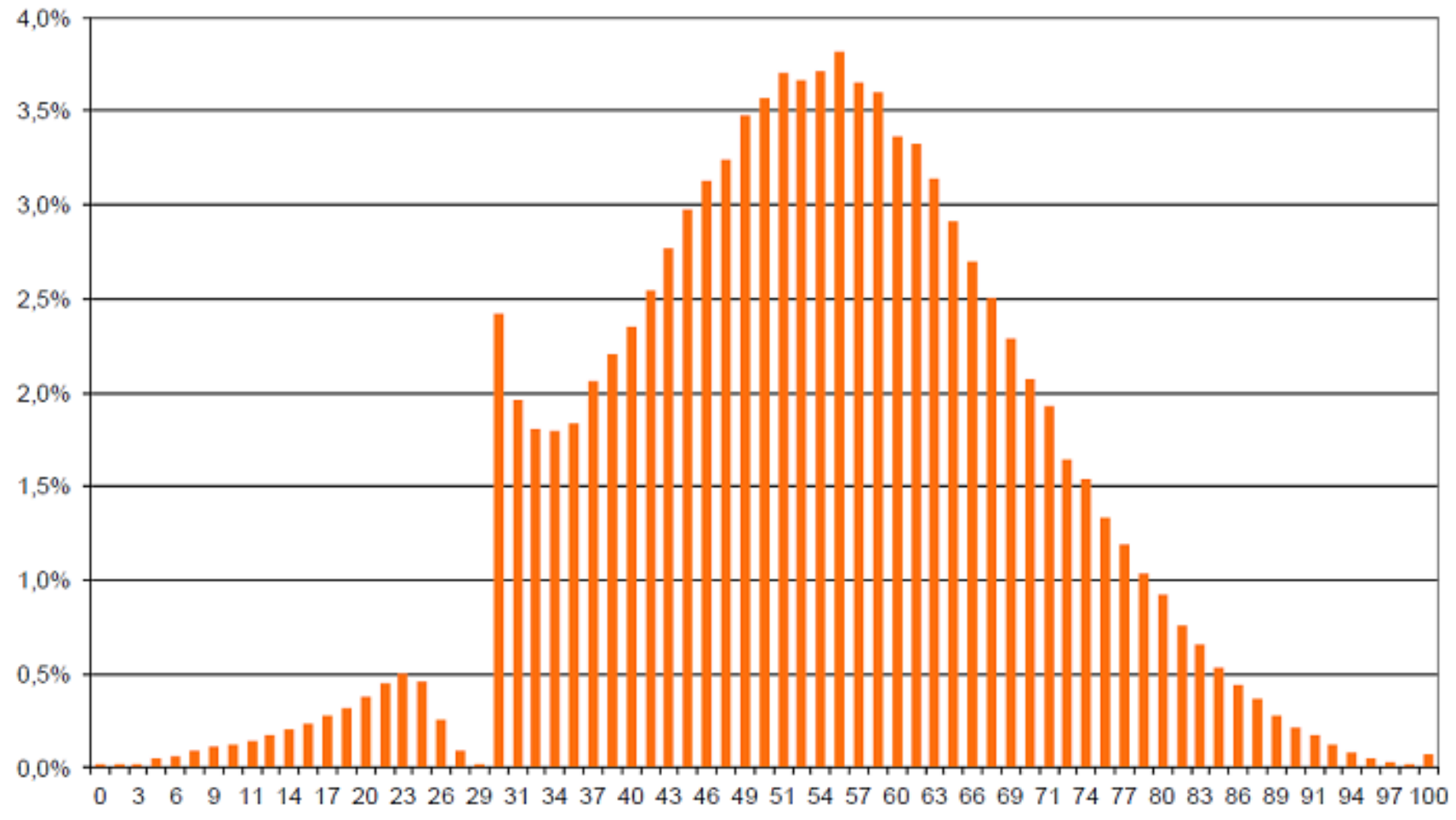


What can one learn from a statistical distribution?

Examination after high school in Poland. Marks: 0 – 100 points.
How do you think the distribution of results may look like?

What can one learn from a statistical distribution?

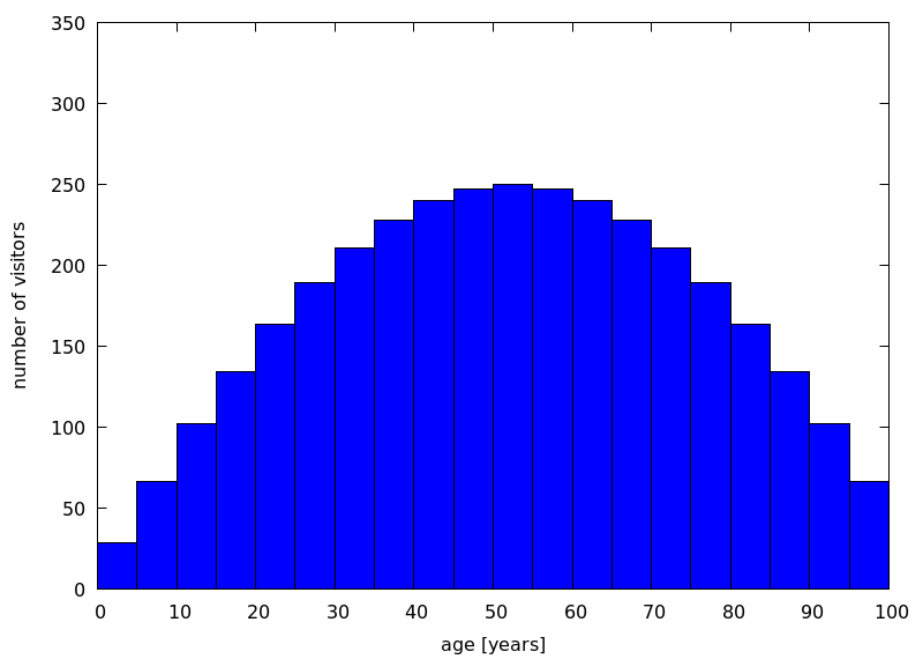
Examination after high school in Poland. Marks: 0 – 100 points.
How do you think the distribution of results may look like?



What can one learn from a statistical distribution?

Age of visitors in museum of particle physics

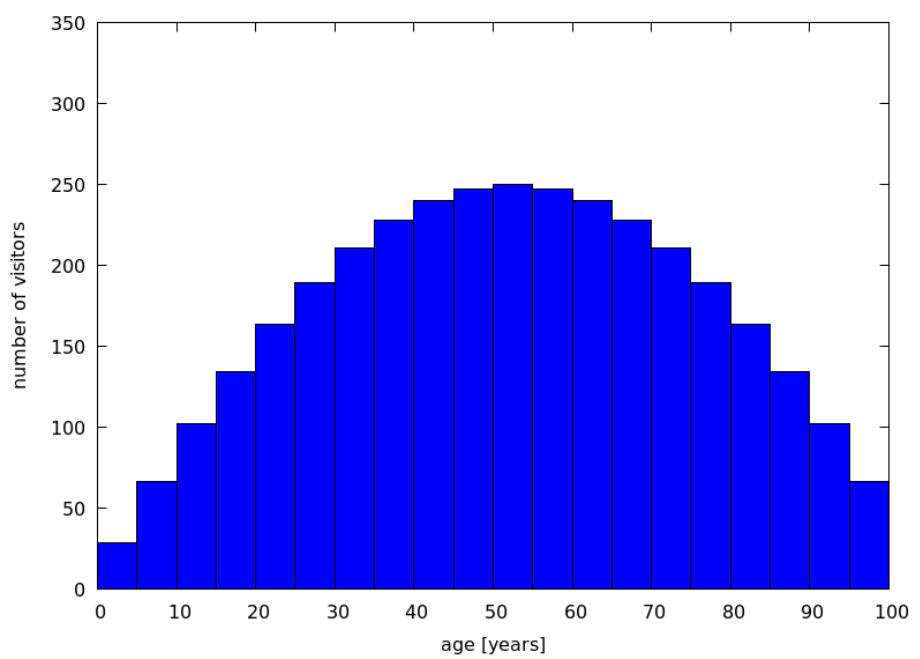
Typical distribution:



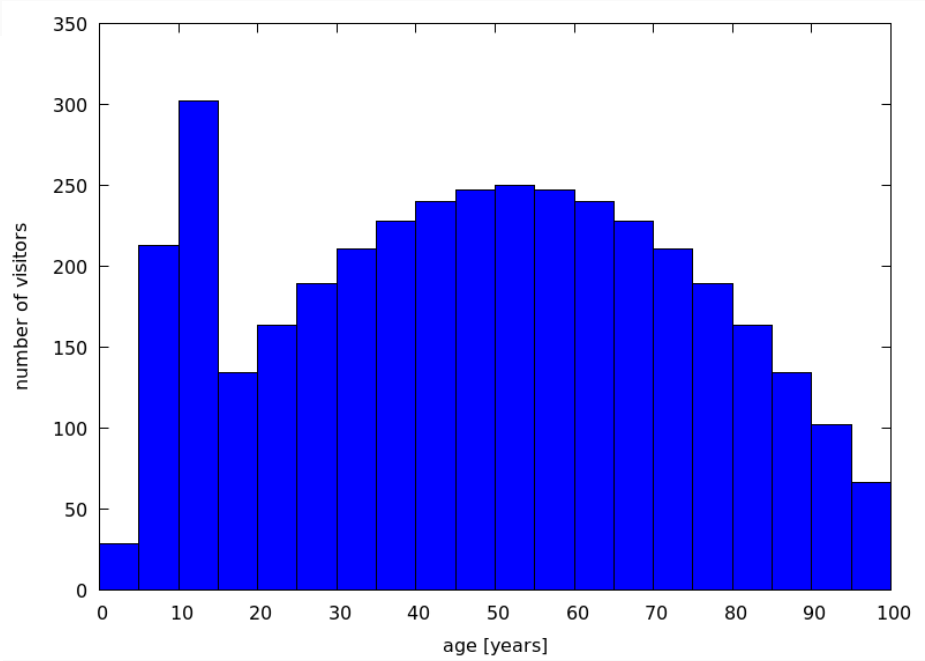
What can one learn from a statistical distribution?

Age of visitors in museum of particle physics

Typical distribution:



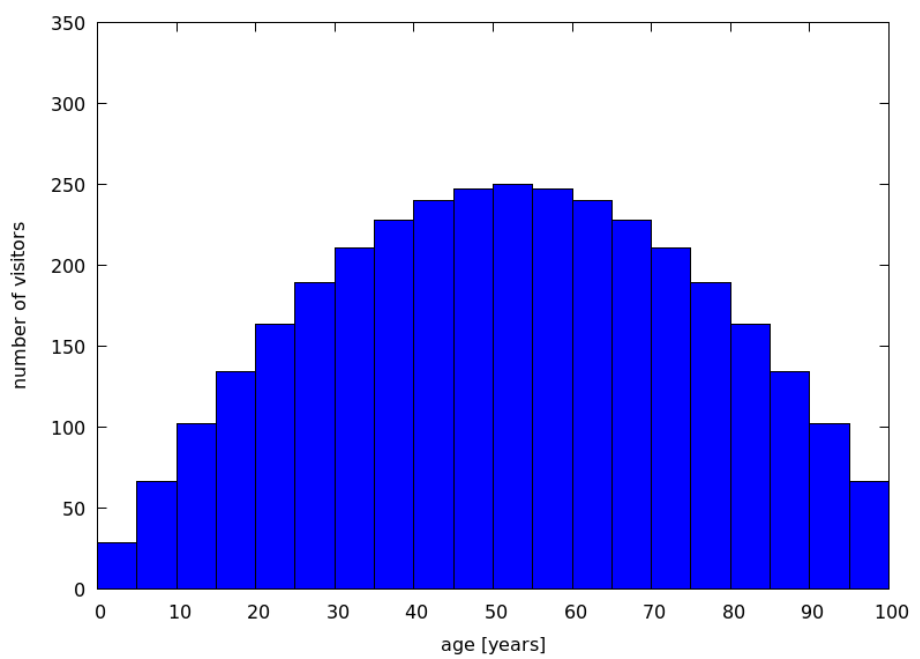
One day:



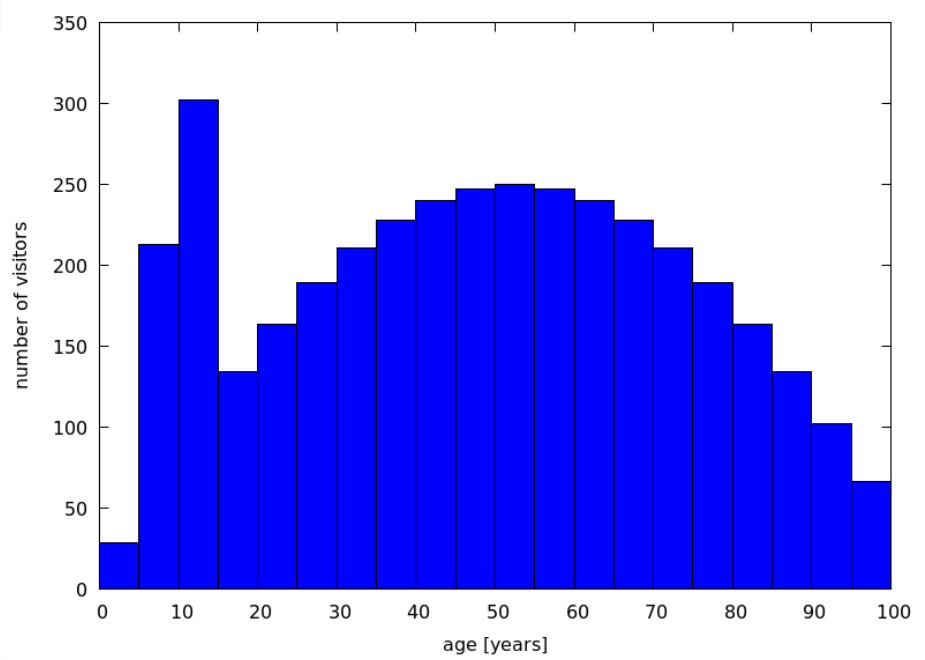
What can one learn from a statistical distribution?

Age of visitors in museum of particle physics

Typical distribution:

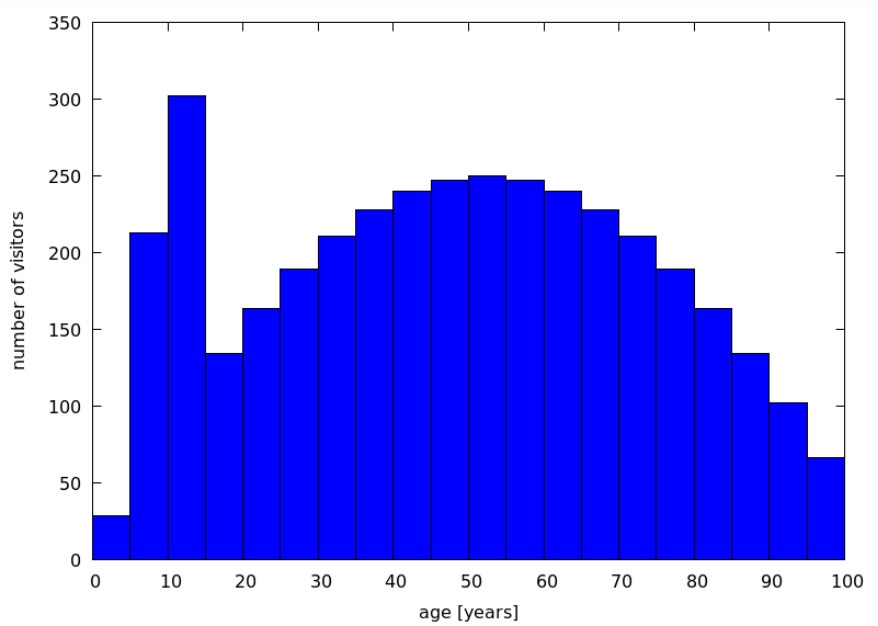


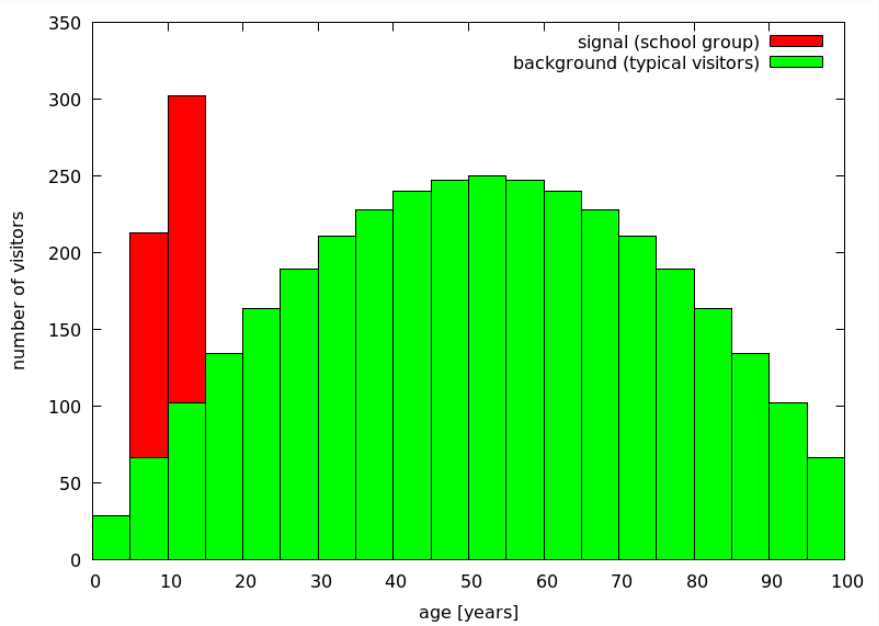
One day:



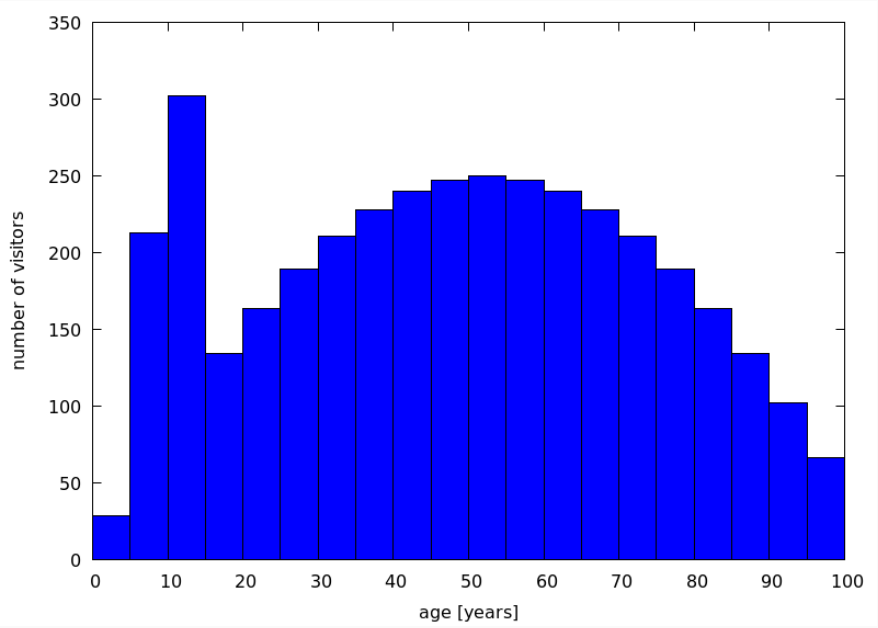
What conclusions can we draw?

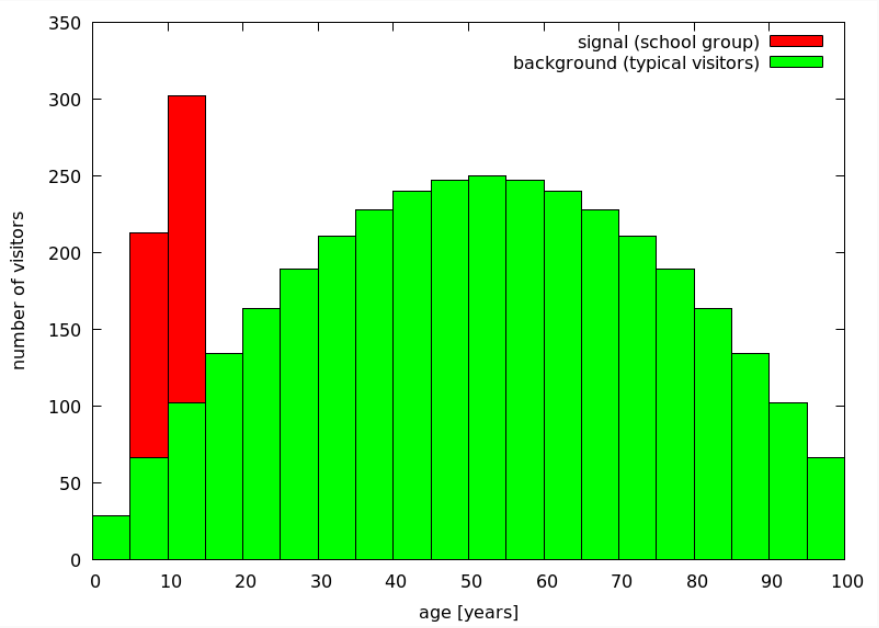
”Signal” and ”background”

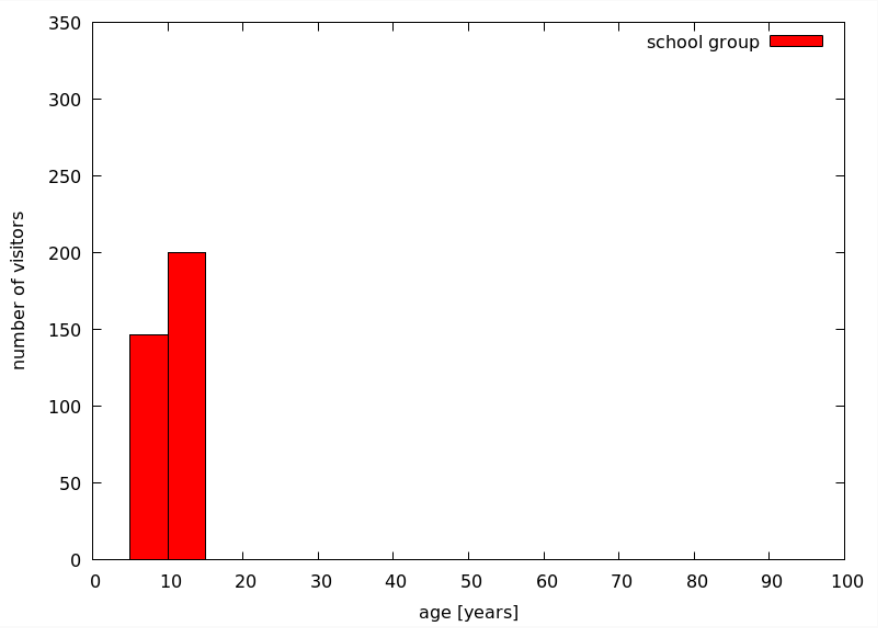


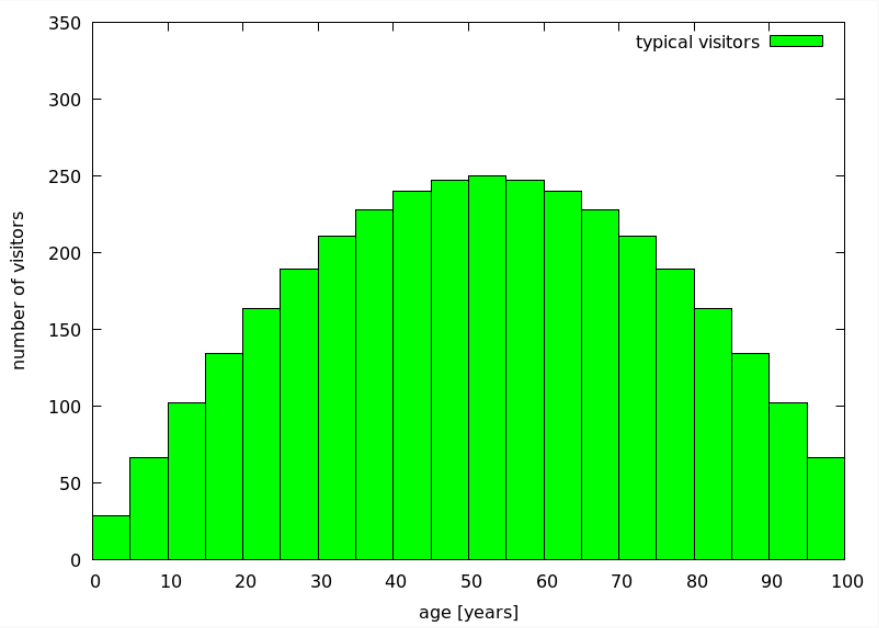


”Signal” and ”background”









Contents

Contents

Command	Description
`root -h`	show help for `root` command
`root [-l]`	open ROOT
`root [-l] file.root`	open ROOT and load a file
`.?`	show help in interactive ROOT
`.q[qq[qq[qq]]]`	close ROOT (more `q`'s – more violently)
`.ls`	list present ROOT directory
`tree->Show(2)`	show 3rd (!) tree entry
`tree->Scan()`	print entries as a table
`tree->Scan("n:x:y:z", "x>2")`	specify variables to show and condition
`new TBrowser`	open TBrowser
`ntuple->StartViewer()`	open an analysis GUI

Contents

C++ program

Program

```
#include <iostream>
#include "TCanvas.h"
#include "TF1.h"
using namespace std;
int main (){
  cout << "hello, world" << endl;
  TCanvas c("can1", "Canvas title", 800, 600);
  TF1 f("fun1", "sin(x)", -5, 5);
  f.Draw();
  c.SaveAs("sin.png");
  return 0;
}
```

Running:

```
g++ program.C -o program[.exe] `root-config --cflags
--glibs`
./program[.exe]
```

C++ macro

```
void macro(){
  cout << "hello, world" << endl;
  TCanvas c("can1", "Canvas title", 800, 600);
  TF1 f("fun1", "sin(x)", -5, 5);
  f.Draw();
  c.SaveAs("sin.png");
}
```

Function name = file name
(w/o extension)

Running:

```
root [-l] [-b] [-q] macro.C[+[+]]
```

or

```
root [-l] [-b]
.x macro.C[+[+]]
```

Python (PyROOT)

```
#!/usr/bin/env python

import ROOT
print "hello, world"
c = ROOT.TCanvas("can1", "Canvas title", 800, 600)
f = ROOT.TF1("fun1", "sin(x)", -5, 5)
f.Draw()
c.SaveAs("sin.png")
```

Running:

```
python script.py
```

or

```
root [-l] [-b]
.x macro.C[+[+]]
```

C++: object vs pointer

Objects:

```
void macro(){
  cout << "hello, world" << endl;
  TCanvas c("can1", "Canvas title", 800, 600);
  TF1 f("fun1", "sin(x)", -5, 5);
  f.Draw();
  c.SaveAs("sin.png");
}
```

Pointers:

```
void macronew(){
  cout << "hello, world" << endl;
  TCanvas * c = new TCanvas("can1", "Canvas title", 800, 600);
  TF1 * f = new TF1("fun1", "sin(x)", -5, 5);
  f->Draw();
  c->SaveAs("sin.png");
}
```

Contents

All you need and more can be found at

`root.cern.ch`