ROOT Exercise 3

The aim of this exercise is to perform a typical analysis in particle physics: having a model of signal and background (in the form of generated ”Monte Carlo” events – MC samples), figure out a selection criteria that can be applied on data events to suppress the background contribution.

Data samples

The analysis will contain processing three different samples:

- data events (integrated luminosity: 100 nb^{-1}),
- signal events generated according to a model (cross section: 5 nb),
- background events generated according to a model (cross section: 500 nb).

In this analysis, the signal is a production of a new electrically neutral particle decaying into a muon pair: $R \to \mu^+\mu^-$. Background contains other processes that can be observed. Data sample contains a mixture of events of both types with an unknown ratio.

Event structure

Each sample contains a set of events. Each **event** consists of some number of **particles**, each particle has the following properties:

- x, y, z – production position
- e, px, py, pz – four-momentum components
- charge – charge
- id – in data: dummy value, in MC samples: particle type (`pdg.lbl.gov/2007/reviews/montecarlorpp.pdf`)
- mu_like – compatibility of the measured particle with a muon

The variables are stored as `std::vector` objects, i.e. vector of `x` values, vector of `y` values, etc. Element number `i` of each vector corresponds to particle number `i`.

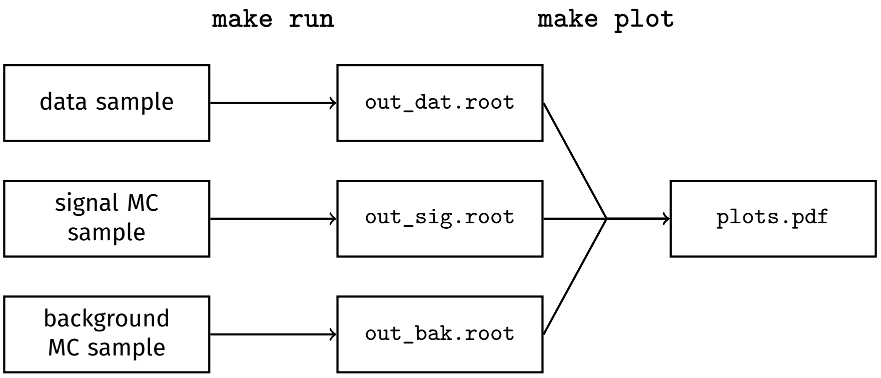
Analysis framework

All the samples will be processed by the same analysis code. The results will be combined by a plotting macro. You will use a simple analysis framework dedicated to this exercise. It consists of the following files:

ntuple.h – definition of class for reading events (created with `MakeClass`)
ntuple.C – analysis program; here all histograms should be declared and filled
plot.C – macro for plotting histograms created by the analysis program
Makefile – definitions of compilation rules, *etc.*

The analysis framework is uses the *GNU Make* tool. The following commands can be used:

make – compiles ntuple.h and ntuple.C into ntuple.exe
make download – downloads the samples
make run – runs `ntuple.exe` (it is automatically compiled, if needed) on the data, signal and background samples and produces root files with histograms: `out_dat.root`, `out_sig.root` and `out_bkg.root`, respectively
make plot – runs the `plot.C` macro and produces `plots.pdf`



Tasks

1. Download the analysis framework:
   `wget ppss.ifj.edu.pl/materials_2019/analysis_framework.tar.gz`
   and unpack it:
   `tar xvzf analysis_framework.tar.gz`
2. Download the samples (`make download`).
3. Skim through the samples in the interactive mode (Show, Scan, TBrowser, StartViewer).
4. Run the analysis framework (for example **make run && make plot**) to obtain the plots that are already implemented as an example.
5. In MC samples, compare distributions of `mu_like` for muons and other particles (normalize the histograms to unity). What selection can be applied on `mu_like` so that 95–99% of background is rejected, while more than 50% of signal is kept?
6. In each event, find all opposite-charge muon pairs. Create distributions of invariant mass of the pairs (use `TLorentzVector` for calculations). Is the resonance peak visible in the data sample?
7. Based on MC model, find an event selection procedure that would keep signal events and reject the background. Try for example the angle and distance between muons, transverse momenta of two muons (and correlations between them).
8. Once your the selection gets better, you may need bigger background samples. You can get them by `make download_big` and then running `make run_big` instead of `make run`.
9. Find the mass of the particle by a Gaussian fit to the data sample after selection.
10. Produce a set of nice plots that explain all analysis steps.