

Replotting the published ALICE figure

March 16, 2021

In this third ROOT exercise we cover the following new frequently used classes in ROOT:

1. TLatex (<http://root.cern.ch/root/html/TLatex.html>)

The ROOT Exercise #3 consists of two parts. After you have accomplished both of them, you will demonstrate that you have acquired enough skills to make the figures of the same quality which appears in the official publications at LHC experiments!

First part

Another frequently encountered case in practice is when somebody sends you the results of his measurements stored in arrays, like for instance me now in the attachment with the file `Example_3.C`, instead in a file with three or more columns.

Question 1: *Please have a look at the attached file for such an example. Add lines in the attached macro which will implement TGraphErrors object which will be automatically initialized with the contents of those arrays (Hint: Now you may prefer another constructor of TGraphErrors, please have a look at the online documentation of that class). I have added some comments, which shall clarify the meaning of each array \Rightarrow this is a very generic example, so please try to memorize it. Please set the following:*

- a) Markers of TGraphErrors object to be blue full circles, the color of error lines blue as well;
- b) Title on y-axis v_2 , where 2 is in the subscript. Hint: ROOT recognizes a lot of LaTeX commands. In general, you produce in ROOT some LaTeX symbol if you use '#' prefix (Example: `\frac{a}{b}` in the ROOT world turns into `#frac{a}{b}`, etc., see further examples in <https://root.cern.ch/root/html/TLatex.html>);
- c) Title on x-axis to be 'centrality';
- d) Please send me the resulting figure in pdf format (Remark: If you didn't bother to introduce separate TCanvas object in your macro, than just use directly a pointer to your TGraphErrors object and call a method Draw() by passing 'pa' as an arguments, 'p' forces drawing of markers, 'a' forces drawing of both axes, and save the figure 'by clicking').

Second part

Question 1: *Please fetch from the HEPData website the data points for Fig. 3 in ALICE publication <http://arxiv.org/pdf/1011.3914v2.pdf>, and reproduce as close as possible this figure in your own plotting macro (i.e. try to resemble the marker styles, colors, axes labels, legend, etc.). Two remarks, though, to make your life simpler:*

- a) Forget about STAR measurements in this figure which are indicated with colorful bands and which are there only for comparison sake;
- b) Some of the ALICE measurements are shifted along x-axis for visibility sake (literally in order to prevent some markers to overlap, this is a common practice in the field!). In order to achieve this 'shifting along x-axis', you may find the following function beneficial:

```

void ShiftAlongXaxis(TGraphErrors *ge, Double_t shift)
{
    // Shift original TGraphErrors along x-axis by amount determined by 'shift'.

    if(!ge)
    {
        printf("\n WARNING: ge is NULL in ShiftAlongXaxis() !!!! \n\n");
        return;
    }

    Int_t nPoints = ge->GetN();
    Double_t x = 0.;
    Double_t y = 0.;
    for(Int_t p=0;p<nPoints;p++)
    {
        ge->GetPoint(p,x,y);
        x+=shift;
        ge->SetPoint(p,x,y);
    } // end of for(Int_t p=0;p<nPoints;p++)

} // end of void ShiftAlongXaxis(TGraphErrors *ge, Double_t shift)

```

If you copy and paste the above routine in your plotting macro, you use it as simply as this:

```
ShiftAlongXaxis(pointer-to-TGraphErrors-object,value-of-displacement);
```

Please send me the figure you have obtained via email, as well as code snippets, so I can provide feedback also on your coding!