Development of TRatioPlot in ROOT

SFT Group Meeting

Paul Gessinger

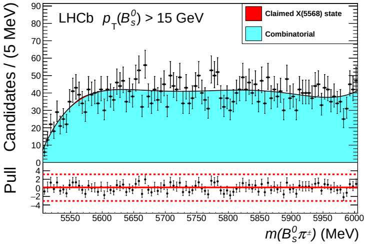
Johannes Gutenberg University Mainz

Supervisors:

Olivier Couet

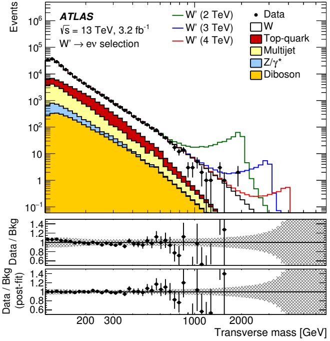
Lorenzo Moneta

19.09.2016



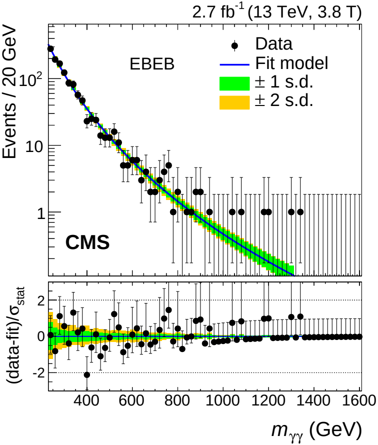
LHCb $p_T(B_s^0) > 15$ GeV
Claimed X(5568) state
Combinatorial
Candidates / (5 MeV)
Pull
90
80
70
60
50
40
30
20
10
0
4
2
0
−2
−4
5550
5600
5650
5700
5750
5800
5850
5900
5950
6000
$m(B_s^0\pi^\pm)$ (MeV)

[2]



ATLAS
$\sqrt{s}$ = 13 TeV, 3.2 fb^{-1}
W' → eν selection
W' (2 TeV)
W' (3 TeV)
W' (4 TeV)
Data
W
Top-quark
Multijet
Z/γ*
Diboson
Events
10^6
10^5
10^4
10^3
10^2
10
1
10^{-1}
Data / Bkg
1.4
1.2
1
0.8
0.6
Data / Bkg (post-fit)
1.4
1.2
1
0.8
0.6
200
300
1000
2000
Transverse mass [GeV]

[1]



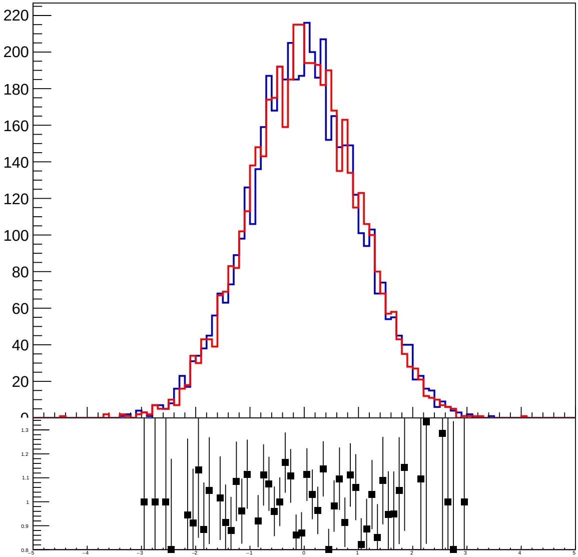
2.7 fb^{-1} (13 TeV, 3.8 T)
EBEB
Data
Fit model
± 1 s.d.
± 2 s.d.
CMS
Events / 20 GeV
10^2
10
1
(data-fit)/σ_{stat}
2
0
-2
400
600
800
1000
1200
1400
1600
$m_{\gamma\gamma}$ (GeV)

[3]

Existing facilities

- Pads can be subdivided into subpads
- All of the same size
- Manually create pads and align them
  - But: Pads have coordinate system depending on their drawn content
  - Font sizes are dependent on size of the pad within parent
  - Axis ranges are independent

Existing facilities

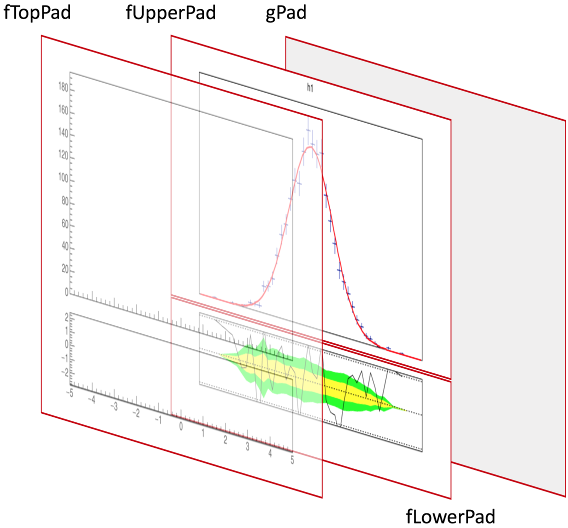


Design goals

- Make visualization consistent and remove redundancy
  - Link two `TPads` together
  - Consistent labels, ticks and tickmarks
- Improve interactively working with this type of plot
  - Displays should be synchronized automatically
  - Elements should be customizable through the Editor
- Implement most common calculations in the class, and implement usual graphical display
  - Ratio of two histograms
  - Difference between two histograms
  - Fit residual between a histogram and a fitted function (shows confidence interval bands)

Display structure

Display structure



fTopPad

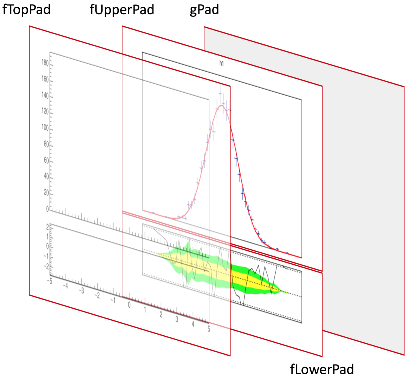
fUpperPad

gPad

fLowerPad

Display structure

- Two pads which contain the input histograms and the calculation output
- Additional transparent pad on top, which receives the `TGAxis`
- This pad needs to pass through interaction:
    - Modified `TPad` to not participate in Pick/ExecuteEvent calls when the bit `kCannotPick` is set



Drawing of graphical elements

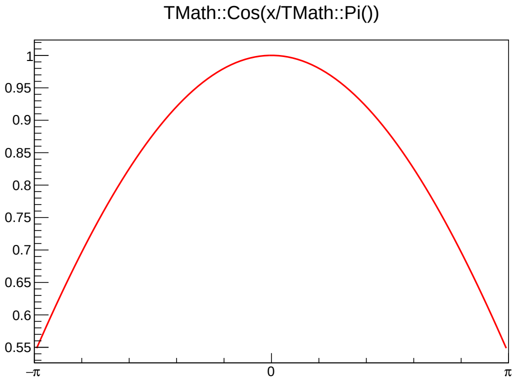
- Original axis drawing is disabled through draw options
- Axes are drawn on top pad, so all sizes are consistent
- Tick mark sizes depend on the length of the axis, normalization by ratio of pad sizes

- Lower plot can contain dashed lines that are drawn at specified y values
- Can alse be disabled with draw option
- Useful defaults are set for different modes

- New facility to change axis label attributes

New facility to change axis label attributes

```
{
   Double_t pi = TMath::Pi();
   TF1*     f = new TF1("f","TMath::Cos(x/TMath::Pi())", -pi, pi);
   TH1*     h = f->GetHistogram();
   TAxis* a = h->GetXaxis();
   a->SetNdivisions(-502);
   a->ChangeLabel(1,-1,-1,-1,-1,-1,"-#pi");
   a->ChangeLabel(-1,-1,-1,-1,-1,-1,"#pi");
   f->Draw();
}
```



User interactivity

Intercept and react to user interaction

- `TRatioPlot` needs to know when content in the pads changes
- Use Signal/Slot mechanism to be called whenever interaction occurs
    - Existing signal: `RangeAxisChanged`. Is emitted additionally in `TPad` on `SetLogx`, `SetLogy` and `SetLogz`
    - New signal: `Resized` when pad is resized
    - New signal: `UnZoomed` is called when the range of the axis is reset by the user
- `TRatioPlot` connects to those signals
- Synchronizes pad margins, axes ranges, logx, and ensures the upper and lower pad meet at one point
- Inherits as many properties as possible from `gPad`

Calculation modes

Available calculation and error modes

- Divide two histograms using `TGraphAsymmErrors::Divide` (default, options given to `TRatioPlot` are passed through)
- Divide two histograms using `TH1::Divide`, yields symmetric errors (option *divsym*, additional options are passed through)
- Subtract a histogram from another one (option *diff*)
- Subtract a histogram and divide by the uncertainty (option *diffsig*)
- Calculate residual between histogram and fitted function (invoked with corresponding constructor)

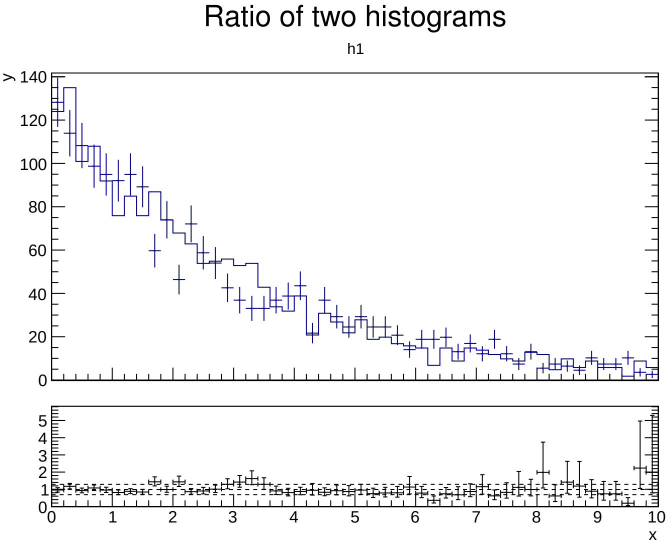
Available calculation and error modes

- Divide two histograms using `TGraphAsymmErrors::Divide` (default, options given to `TRatioPlot` are passed through)
- Divide two histograms using `TH1::Divide`, yields symmetric errors (option *divsym*, additional options are passed through)
- Subtract a histogram from another one (option *diff*)
- Subtract a histogram and divide by the uncertainty (option *diffsig*)
- Calculate residual between histogram and fitted function (invoked with corresponding constructor)

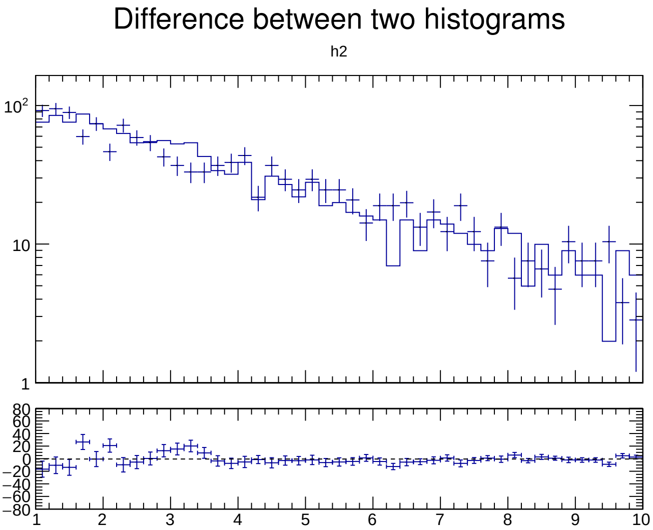
Option	Error mode
errasym	Uses calculated asymmetric errors from `TH1::GetBinErrorUp`/`TH1::GetBinErrorLow`. Note that you need to set `TH1::SetBinErrorOption` first
errfunc	Uses $\sqrt{f(x)}$ as the error

Examples

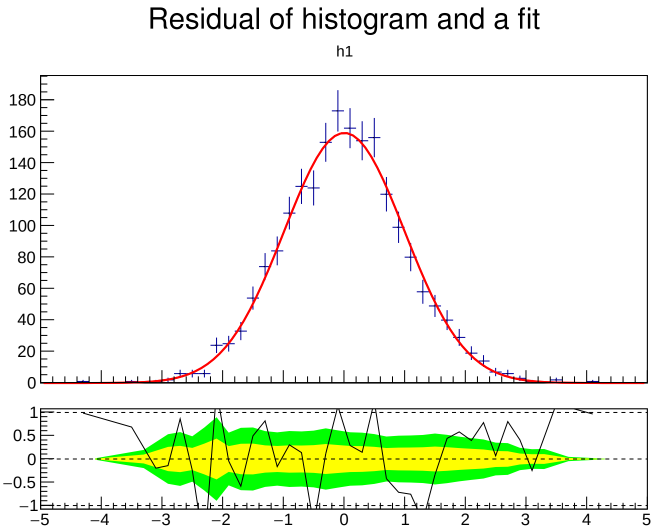
Examples


Ratio of two histograms
h1
y
140
120
100
80
60
40
20
0
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9
10
x

Examples


Difference between two histograms
h2
10^2
10
1
80
60
40
20
0
−20
−40
−60
−80
1
2
3
4
5
6
7
8
9
10

Examples


Residual of histogram and a fit
h1
180
160
140
120
100
80
60
40
20
0
1
0.5
0
−0.5
−1
−5
−4
−3
−2
−1
0
1
2
3
4
5

Programmatic usage

Programmatic steering of display

- In addition to interactivity, code duplication should be reduced
- Range and logx can be set on one pad and it is automatically synced
- Dashed lines can be customized by passing an array of y positions (`TRatioPlot::SetGridlines`)
- Access to pads and all internals is provided with getters, so customization is possible

Necessary code to produce good looking results is reduced **significantly**

Programmatic steering of display

Before: ≈ 50 lines, ≈ 20 lines of visual setup

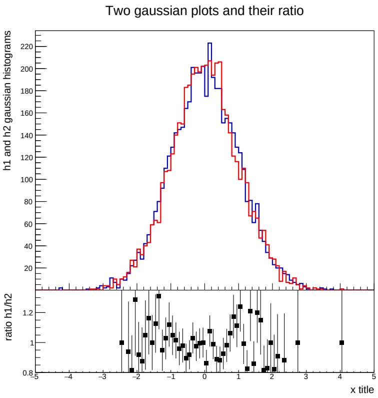
```
void ratioplotOld( ) {
   // ... create h1 and h2
   TCanvas *c = new TCanvas("c", "canvas", 800, 800);
   TPad *pad1 = new TPad("pad1", "pad1", 0, 0.3, 1, 1.0);
   // ...
   pad1->Draw();
   pad1->cd();
   h1->SetStats(0);
   h1->Draw();
   h2->Draw("same");

   h1->GetYaxis()->SetLabelSize(0.);
   TGaxis *axis = new TGaxis( -5, 20, -5, 220, 20,220,510,"");
   axis->SetLabelFont(43);
   axis->SetLabelSize(15);
   axis->Draw();

   c->cd();
   TPad *pad2 = new TPad("pad2", "pad2", 0, 0.05, 1, 0.3);
   // ...
   pad2->Draw();
   pad2->cd();

   TH1F *h3 = (TH1F*)h1->Clone("h3");
   // ...
   h3->Divide(h2);
   // ...
   h3->GetYaxis()->SetTitle("ratio␣h1/h2␣");
   h3->GetYaxis()->SetNdivisions(505);
   // ...
}
```

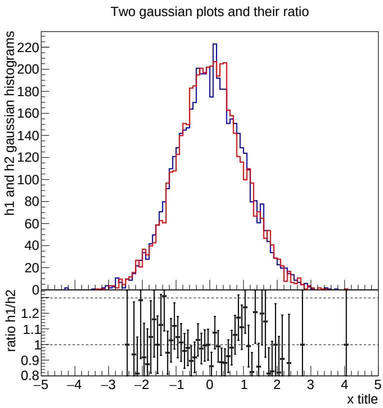
Programmatic steering of display

With TRatioPlot: ≈ 20 lines, 4 lines of visual setup

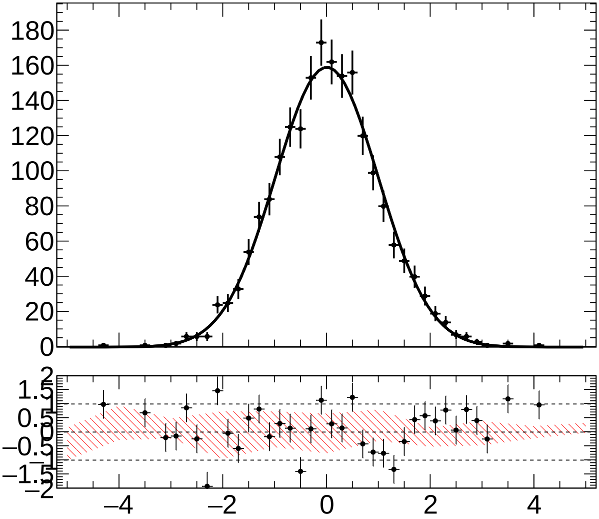
```
void ratioplot1() {
   gStyle->SetOptStat(0);
   TH1D *h1 = new TH1D("h1", "", 100, -5, 5);
   TH1D *h2 = new TH1D("h2", "h2", 100, -5, 5);
   h1->FillRandom("gaus");
   h2->FillRandom("gaus");

   h1->SetLineColor(kBlue+1);
   h1->SetLineWidth(2);

   h2->SetLineColor(kRed);
   h2->SetLineWidth(2);

   TCanvas *c1 = new TCanvas("c", "canvas", 800, 800);
   auto rp = new TRatioPlot(h1, h2, "pois␣midp");
   rp->SetH2DrawOpt("hist");
   rp->SetSeparationMargin(0);
   rp->Draw();

   rp->GetLowerRefGraph()->SetMinimum(0.8);
   rp->GetLowerRefGraph()->SetMaximum(1.35);
   rp->GetLowerRefGraph()->SetLineColor(kBlack);
   rp->GetLowerRefYaxis()->SetTitle("ratio␣h1/h2");
   rp->GetLowerRefYaxis()->SetTitleOffset(1.3);
   rp->GetUpperRefYaxis()->SetTitleOffset(1.3);
}
```

Programmatic steering of display





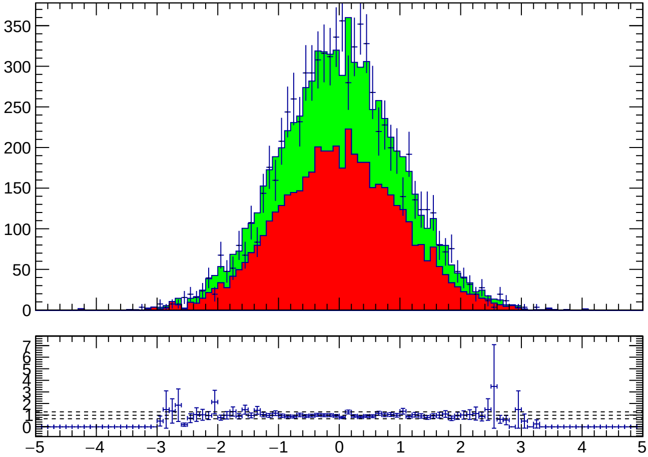
Customization: ATLAS style



Customization: ATLAS style

```
{
   gROOT->LoadMacro("AtlasStyle.C");
   SetAtlasStyle();
   auto c1 = new TCanvas("c1", "fit␣residual␣simple");
   auto h1 = new TH1D("h1", "h1", 50, -5, 5);
   // ... fill h1 and fit it
   c1->Clear();

   auto more = new TGraphAsymmErrors();
   // ... build custom error band and style it

   auto rp = new TRatioPlot(h1);
   rp->Draw();

   // build plot
   auto p = rp->GetLowerPad();
   p->cd();
   p->Clear();
   auto fitres = rp->GetCalculationOutputGraph();
   more->Draw("IA3");
   fitres->Draw("P+same");
}
```

Works with THStack out of the box



Works with THStack out of the box

```
{
   auto c1 = new TCanvas();

   auto h1 = new TH1D("h1", "", 100, -5, 5);
   auto h2 = new TH1D("h2", "", 100, -5, 5);
   auto h3 = new TH1D("h3", "", 100, -5, 5);

   // ... fill and style h1, h2 and h3

   auto stack = new THStack();
   stack->Add(h1);
   stack->Add(h2);

   auto rp = new TRatioPlot(stack, h3, "divsym");
   rp->Draw();
}
```

Summary

Status:

- Class performing drawing and synchronization of two pads has been created
- Performs most common calculations
- Merged in master and ready to go

Possible improvements:

- Refresh of canvas is still an issue, artifacts appear and occasionally stick until manual window resize
  - No problems observed in static case
- Only works with two pads
  - Generalization to have mutliple pads linked together would be useful (tested on prototype)
  - Calculation input would have to be flexible

[1] Morad Aaboud et al. “Search for new resonances in events with one lepton and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: (2016). arXiv: 1606.03977 [hep-ex].

[2] Roel Aaij et al. “Search for structure in the $B_s^0 \pi^{\pm}$ invariant mass spectrum”. In: (2016). arXiv: 1608.00435 [hep-ex].

[3] Vardan Khachatryan et al. “Search for resonant production of high-mass photon pairs in proton-proton collisions at sqrt(s) = 8 and 13 TeV”. In: *Phys. Rev. Lett.* 117.5 (2016), p. 051802. DOI: 10.1103/PhysRevLett.117.051802. arXiv: 1606.04093 [hep-ex].