

This is a test of highlighting code

Here is the code we want to show.

```
pdf->plotOn(plot);
//draw the two separate pdf's
pdf->plotOn(plot, RooFit::Components("bkg_pdf"), RooFit::LineStyle(kDashed) );
pdf->plotOn(plot, RooFit::Components("sig_pdf"), RooFit::LineColor(kRed), RooFit::LineStyle(kDash
plot->Draw(); // to show the RooPlot in the current ROOT Canvas

using namespace RooFit;

void CombinedModel() {

    RooWorkspace w("w");

    w.factory("Exponential:bkg1_pdf(x[0,10], a1[-0.5,-2,-0.2])");
    w.factory("Gaussian:sig_pdf(x, mass[2], sigma[0.3])");

    w.factory("prod:nsig1(mu[1,0,5],xsec1[30])");

    w.factory("SUM:model1(nsig1*sig_pdf, nbkg1[1000,0,10000]*bkg1_pdf)"); // for extended model

    w.factory("Exponential:bkg2_pdf(x, a2[-1,-2,-0.2])");

    w.factory("prod:nsig2(mu,xsec2[20])");
    w.factory("SUM:model2(nsig2*sig_pdf, nbkg2[100,0,10000]*bkg2_pdf)"); // for extended model

    // Create discrete observable to label channels
    w.factory("index[channel1,channel2]");

    // Create joint pdf (RooSimultaneous)
    w.factory("SIMUL:jointModel(index,channel1=model1,channel2=model2)");

    RooAbsPdf * pdf = w.pdf("jointModel");
    RooRealVar * x = w.var("x"); // the observable
    RooCategory * index = w.cat("index"); // the category

    // generate the data (since it is exetended we can generate the global model

    // use fixed random numbers for reproducibility
    RooRandom::randomGenerator()->SetSeed(111);
    // generate binned
    // plot the generate data in 50 bins (default is 100)
    x->setBins(50);

    // generate events of joint model
    // NB need to add also category as observable
    RooDataSet * data = pdf->generate( RooArgSet(*x,*index)); // will generate accordint to total
    data->SetName("data");
    w.import(*data);

    data->Print();

    RooPlot * plot1 = x->frame();
    RooPlot * plot2 = x->frame();
    data->plotOn(plot1,RooFit::Cut("index==index::channel1"));
    data->plotOn(plot2,RooFit::Cut("index==index::channel2"));
    // plot->Draw();

    RooFitResult * r = pdf->fitTo(*data, RooFit::Save(true), RooFit::Minimizer("Minuit2","Migrad")
    r->Print();

    pdf->plotOn(plot1,RooFit::ProjWData(*data),RooFit::Slice(*w.cat("index"),"channel1"));
```

## TestCodePage < RooStats < TWiki

```
pdf->plotOn(plot2,RooFit::ProjWData(*data),RooFit::Slice(*w.cat("index"),"channel2"));
//draw the two separate pdf's
pdf->paramOn(plot1);
c1 = new TCanvas();
c1->Divide(1,2);
c1->cd(1); plot1->Draw();
c1->cd(2); plot2->Draw();

RooStats::ModelConfig mc("ModelConfig",&w);
mc.SetPdf(*pdf);
mc.SetParametersOfInterest(*w.var("mu"));
mc.SetObservables(RooArgSet(*w.var("x"),*w.cat("index")));

// define set of nuisance parameters
w.defineSet("nuisParams","a1,nbkg1,a2,nbkg2");

mc.SetNuisanceParameters(*w.set("nuisParams"));

// set also the snapshot
mc.SetSnapshot(*w.var("mu"));

// import model in the workspace
w.import(mc);

// write the workspace in the file
TString fileName = "CombinedModel.root";
w.writeToFile(fileName,true);
cout << "model written to file " << fileName << endl;
}

using namespace RooFit;

void CombinedModel() {

RooWorkspace w("w");

w.factory("Exponential:bkg1_pdf(x[0,10], a1[-0.5,-2,-0.2])");
w.factory("Gaussian:sig_pdf(x, mass[2], sigma[0.3])");

w.factory("prod:nsig1(mu[1,0,5],xsec1[30])");

w.factory("SUM:model1(nsig1*sig_pdf, nbkg1[1000,0,10000]*bkg1_pdf)"); // for extended model

w.factory("Exponential:bkg2_pdf(x, a2[-1,-2,-0.2])");

w.factory("prod:nsig2(mu,xsec2[20])");
w.factory("SUM:model2(nsig2*sig_pdf, nbkg2[100,0,10000]*bkg2_pdf)"); // for extended model

// Create discrete observable to label channels
w.factory("index[channel1,channel2]");

// Create joint pdf (RooSimultaneous)
w.factory("SIMUL:jointModel(index,channel1=model1,channel2=model2)");

RooAbsPdf * pdf = w.pdf("jointModel");
RooRealVar * x = w.var("x"); // the observable
RooCategory * index = w.cat("index"); // the category

// generate the data (since it is extended we can generate the global model

// use fixed random numbers for reproducibility
RooRandom::randomGenerator()->SetSeed(111);
// generate binned
```

## TestCodePage < RooStats < TWiki

```
// plot the generate data in 50 bins (default is 100)
x->setBins(50);

// generate events of joint model
// NB need to add also category as observable
RooDataSet * data = pdf->generate( RooArgSet(*x,*index)); // will generate accordint to total
data->SetName("data");
w.import(*data);

data->Print();

RooPlot * plot1 = x->frame();
RooPlot * plot2 = x->frame();
data->plotOn(plot1,RooFit::Cut("index==index::channel1"));
data->plotOn(plot2,RooFit::Cut("index==index::channel2"));
// plot->Draw();

RooFitResult * r = pdf->fitTo(*data, RooFit::Save(true), RooFit::Minimizer("Minuit2","Migrad")
r->Print();

pdf->plotOn(plot1,RooFit::ProjWData(*data),RooFit::Slice(*w.cat("index"),"channel1"));
pdf->plotOn(plot2,RooFit::ProjWData(*data),RooFit::Slice(*w.cat("index"),"channel2"));
//draw the two separate pdf's
pdf->paramOn(plot1);
c1 = new TCanvas();
c1->Divide(1,2);
c1->cd(1); plot1->Draw();
c1->cd(2); plot2->Draw();

RooStats::ModelConfig mc("ModelConfig",&w);
mc.SetPdf(*pdf);
mc.SetParametersOfInterest(*w.var("mu"));
mc.SetObservables(RooArgSet(*w.var("x"),*w.cat("index")));

// define set of nuisance parameters
w.defineSet("nuisParams","a1,nbkg1,a2,nbkg2");

mc.SetNuisanceParameters(*w.set("nuisParams"));

// set also the snapshot
mc.SetSnapshot(*w.var("mu"));

// import model in the workspace
w.import(mc);

// write the workspace in the file
TString fileName = "CombinedModel.root";
w.writeToFile(fileName,true);
cout << "model written to file " << fileName << endl;
}
```

---

This topic: [RooStats > TestCodePage](#)

Topic revision: r7 - 2013-06-02 - LorenzoMoneta



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)