Esercitazione n.1 Corso di Laboratorio Analisi Dati Docente: Alexis Pompili

ESERCITAZIONE-1 del corso di LABORATORIO ANALISI DATI (A.Pompili)

In questa esercitazione si impara:

- 1) ad eseguire i principali comandi di ROOT,
- 2) ad eseguire una macro scritta in C (in ambiente ROOT)
- a fare un confronto dati reali dati simulati (Monte Carlo) con normalizzazione assoluta
- 4) a preparare un plot (output di ROOT) con caratteristiche e qualita' da pubblicazione
- 5) a rappresentare le varie componenti simulate con gli stacked plots .



Il plot che faremo e' del tipo di quello che appare nella pag.11 (fig.4) della pubblicazione *Performance of CMS muon reconstruction in pp collisions at sqrt{s}= 7TeV* sulla rivista **Journal of Instrumentation (JINST) 7, P10002 (2012).** [link dalla pagina <u>http://webcms.ba.infn.it/~pompili/data_analysis_lab/unix-links.html</u>]

Si consiglia preliminarmente di studiare le pagg.6-12 del suddetto articolo per capire il significato fisico dei concetti seguenti:

- Soft Muons, Tight Muons
- Prompt Muons, Muons from Beauty, Muons from Charm, Muons from Light Hadrons
- Fake Muons (hadrom punch-through), Duplicates (Ghost Muons)

Per iniziare ...

Collegatevi sulla macchina virtuale 90.147.102.28 del Tier2 di Bari
[ringrazio G.Donvito e V.Spinoso (Tier-2 managers) per la configurazione della macchina]:
1) via *Xming* se usate sistema operativo Windows (come per il desktop in laboratorio),
2) via *ssh* se usate un laptop con Linux/Unix: *ssh -X username@90.147.102.28*

Una volta entrati sulla macchina ci si trova nella directory: */home/username* Create il file *login_slc6.sh*: *>touch login_slc6.sh* Nel file scrivete la seguente riga (usate *vi* o *emacs*):

source /afs/cern.ch/sw/lcg/app/releases/ROOT/5.32.04/x86_64-slc5-gcc43-opt/root/bin/thisroot.sh

Rendetelo un file eseguibile: *chmod u+x login_slc6.sh* ed eseguitelo: *source login_slc6.sh* (lo farete ogni volta che aprite la sessione)

Create la sottodirectory /home/username/esercitazione-1: *mkdir esercitazione-1* Andate in tale sottodirectory e createne un'altra: *mkdir rootfiles*

In questa cartella vi copio i rootfiles seguenti (il primo contiene dati reali, l'altro simulati):

-rw-r-xr-- 1 pompili cms 272774 Nov 6 16:35 *Histos_Data_ZeroBias_1aprnew_goodZB_last_OK.root* -rw-r-xr-- 1 pompili cms 322854 Nov 6 16:35 *Histos_Mc_MinBias_1aprnew_goodZB_last.root* Qualche cenno generale su ROOT

What is ROOT?

- ROOT is an object oriented framework for data analysis
 - read data from some source
 - write data (persistent objects)
 - selected data with some criteria
 - produce results as plots, numbers, fits, ...
- Supports "interactive" (C/C++ like, Python) and "compiled" (C++) usage
- Integrates several tools like random number generations, fit methods (Minuit), Neural Network framework
- Developed and supported by High Energy Ph. community
 - homepage with documentation and tutorials: root.cern.ch

ROOT's user interface

	C++ in batch mode		
		root -b	-q myMacro.C > myMacro.log
	C++ interpreted code with	n CINT	T – the C++ interpreter
	in the command line:	root[0]	for (int i=0; i<10; i++) cout<<"hello "< <i<endl;< td=""></i<endl;<>
	loading a macro:	root[1] root[2]	L mySmallMacro.C; myFunction(1, 2, 3);
•	C++ compiled code via C	INT	root[] .L myScript.C+ Creating shared library /home//MyScript_C.so
	Python:		
	 Access to ROOT from Pyth 		>>> from ROOT import TLorentzVector >>> I = TLorentzVector
	 Access to Python from ROOT root [0] TPython::LoadMacro root [1] MyPyClass mpc; 		
			<pre>root [0] TPython::LoadMacro("MyPyClass.py") root [1] MyPyClass mpc;</pre>

PHYSTAT 05, Oxford

Histograms

- 1-2-3 dimensional histograms
 - Errors for each bin can be computed:
 - Default: as sqrt(bin content)
 - As sqrt(sum of squares of weights of the bin)
- 1-2 dimensional profile histograms
 - Mean value of Y and its standard deviation for each bin in X



In questa esercitazione lavoreremo sugli istogrammi. Come documentazione di riferimento trovate: *ftp://root.cern.ch/root/doc/3Histograms.pdf*

Analysis of TTrees

TTree::Draw method and TTreeViewer - an easy way to examine the tree:

- Producing histograms of user-defined expressions in up to 4 dimensions
- Expressions C++ formulas
- Selections expressions, user-defined macros or graphical cuts

Examples:

Tree.Draw("sqrt(x):y", "x>0 && y<1"); Tree.Draw("2*TMath::Log(x)", cut1 || cut2);



Fitting - interface

- Minimization packages: Minuit and Fumili
- Fitting can be done:
 - Directly in those packages with a user-defined function to minimize
 - Through the general interface of
 - TH1::Fit (binned data) Chisquare and Loglikelihood methods
 - TGraph::Fit (unbinned data)
 - TGraphErrors::Fit (data with errors)
 - TGraphAsymmErrors::Fit (taking into account asymmetry of errors)
 - TTree::Fit and TTree::UnbinnedFit
- RooFit package for object-oriented data modeling. Distributed with ROOT starting from version 5.02-00



15th September 2005

Graphs

- 1-d:
 - TGraph
 - TGraphErrors
 - TGraphAsymmErrors
 - TMultiGraph a collection of graphs
- 2-d:
 TGraph2D
 TGraph2DErrors





15th September 2005

PHYSTAT 05, Oxford

Inizio Esercitazione

Per lanciare il programma ROOT



Nota:per evitare l'apparire (temporaneo del logo di ROOT) fate: [username@alexis-corso rootfiles]\$ *root -l*

Per uscire dall'ambiente ROOT:root [0] .qPer eseguire un comando della shell unix: root [0] .!comando-shellPer vedere tutti i comandi:root [0] .?

Per aprire ed ispezionare una file *rootupla* con ROOT - 1

Per aprire la seguente rootupla con ROOT, Histos_Mc_MinBias_1aprnew_goodZB_last.root, fare:

[username@alexis-corso rootfiles]\$ root -1 Histos_Mc_MinBias_1aprnew_goodZB_last.root

E una volta entrati in ambiente ROOT apro il "TBrowser" (e' la ROOT GUI cioe' l'interfaccia grafica di ROOT) per vedere il file (che contiene, in questo caso, solo istogrammi): root [0] **TBrowser a**

Questo comando lancia il seguente pannello interattivo (con cui posso ispezionare il file):



Per aprire ed ispezionare una file *rootupla* con ROOT - 2

In alternativa: [username@alexis-corso rootfiles]\$ root -I

E una volta entrati... apro il file: root[0] Tfile f("Histos_Mc_MinBias_1aprnew_goodZB_last.root")

Posso ispezionare la lista degli istogrammi contenuti: root[1] f->ls()

TFile**	Histos_Mc_MinBias_1aprnew_goodZB_last.root			
TFile*	Histos_Mc_MinBias_1aprnew_goodZB_last.root			
KEY: TH1I	hMuons_nRecoMuons;1 number of reco muons per event			
KEY: TH1I	hMuons_nSOFT;1 number of soft muons per event			
KEY: TH1I	hMuons_nTIGHT;1 number of tight muons per event			
KEY: TH1I	hMuons_nRecoMuons_1gpv;1 number of reco muons per event – 1gpv			
KEY: TH1I	hMuons_nSOFT_1gpv;1 number of soft muons per event – 1gpv			
KEY: TH1I	hMuons_nTIGHT_1gpv;1 number of tight muons per event – 1gpv			
KEY: TH1I	hMuons_nRecoMuons_2gpv;1 number of muons per event - 2gpv			
KEY: TH1I	hMuons_nSOFT_2gpv;1 number of soft muons per event - 2gpv			
KEY: TH1I	hMuons_nTIGHT_2gpv;1 number of tight muonsper event - 2gpv			
KEY: TH1D	hMuons_p_SOFT_1gpv;1 SOFT p ONLY-1-goodPV			
KEY: TH1D	hMuons_qoverp_SOFT_1gpv;1 SOFT q/p ONLY=1=goodPV			
KEY: TH1D	hMuons_pt_SOFT_1gpv;1 SOFT pt ONLY-1-goodPV			
KEY: TH1D	hMuons_ptErr_SOFT_1gpv;1 SOFT ptErr ONLY-1-goodPV			
KEY: TH1D	hMuons_qtimespt_SOFT_1gpv;1 SOFT q*pt ONLY-1-goodPV			
KEY: TH1D	hMuons_barrelqtimespt_SOFT_1gpv;1 SOFT q*pt ONLY-1-goodPV BARREL			
KEY: TH1D	hMuons_endcapqtimespt_SOFT_1gpv;1 SOFT q*pt ONLY-1-goodPV ENDCAP			
KEY: TH1D	hMuons_eta_SOFT_1gpv;1 SOFT eta ONLY-1-goodPV			
KEY: TH1D	hMuons_etaErr_SOFT_1gpv;1 SOFT etaErr ONLY-1-goodPV			
KEY: TH1D	hMuons_phi_SOFT_1gpv;1 SOFT phi ONLY–1–goodPV			
KEY: TH1D	hMuons_barrelphi_SOFT_1gpv;1			
KEY: TH1D	hMuons_endcapphi_SOFT_1gpv;1 SOFT endcapphi ONLY–1–goodPV			
KEY: TH1D	hMuons_phiErr_SOFT_1gpv;1 SOFT phiErr ONLY-1-goodPV			
KEY: TH1D	hMuons_phibarrel_SOFT_1gpv;1 SOFT BARREL phi ONLY-1-goodPV			
KEY: TH1D	hMuons_plusphioverlap_SOFT_1gpv;1 SOFT_OVERLAP+ phi_ONLY-1-goodPV			
KEY: TH1D	hMuons_plusphiendcap1_SOFT_1gpv;1 SOFT_ENDCAP1+ phi_ONLY=1-goodPV			
KEY: TH1D	hMuons_plusphiendcap2_SOFT_1gpv;1 SOFT_ENDCAP2+_phi_ONLY-1-goodPV			
KEY: TH1D	hMuons_plusphiendcap3_SOFT_1gpv;1 SOFT_ENDCAP3+_phi_ONLY-1-goodPV			
KEY: TH1D	hMuons_minusphioverlap_SOFT_1gpv;1 SOFT_OVERLAPphi_ONLY-1-goodPV			
KEY: TH1D	hMuons_minusphiendcap1_SOFT_1gpv;1 SOFT_ENDCAP1phi_ONLY-1-goodPV			
KEY: TH1D	hMuons_minusphiendcap2_SOFT_1gpv;1 SOFT_ENDCAP2phi_ONLY-1-goodPV			
KEY: TH1D	hMuons_minusphiendcap3_SOFT_1gpv;1 SOFT_ENDCAP3phi_ONLY-1-goodPV			
KEY: TH1D	hMuons_chi2n_zoom_SOFT_1gpv;1 SOFT Normalized-chi2 ONLY-1-goodPV - zoom			
KEY: TH1D	hMuons_chi2n_SOFT_1gpv;1 SOFT Normalized-chi2 ONLY-1-goodPV			
KEY: TH1D	hMuons_ip3d_front_SOFT_1gpv;1 SOFT IP3D wrt best PV - ONLY-1-goodPV			

Infine posso aprire il browser interattivo:
root [2] TBrowser a
Oppure guardarmi da linea di comando il singolo istogramma:
root[2] hMuons_qoverp_SOFT_beauty_1gpv->Draw()





Una volta eseguita la *macro*, visualizzare il file con i plot:

[username@alexis-corso step0]\$ *display ./Plots/qtimespt_SOFT_7nov_Step0_Log.png*



Nota: gli overflow/underflow si apprezzano tipicamente con la scala logaritmica

La visualizzazione degli overflow/underflow negli istogrammi non e' un default in ROOT. Per ottenerla si procede con le seguenti linee di codice da inserire per ogni istogramma:

//-overflow
int nBins_hData = hData->GetNbinsX();
int nBins_ovflw_hData = hData->GetBinContent(nBins_hData + 1); // contenuto di overflow (nel bin n+1)
hData->AddBinContent(nBins_hData,nBins_ovflw_hData); // sommo al contenuto del bin n quello del bin n+1
//-underflow
int nBins_unflw_hData = hData->GetBinContent(0); // contenuto di underflow (nel bin 0)
hData->AddBinContent(1,nBins_unflw_hData); // sommo al contenuto del bin 1 quello del bin 0

Per fare apparire le entries dei dati come pallini e non come croci aggiungere: hData->SetMarkerStyle(20); per ridimensionarli aggiungere: hData->SetMarkerSize(0.55);

Per dare colori di riempimento agli istogrammi delle componenti simulate:



STEP-1 : normalizzazione assoluta – 1

11 // Con un tool di esperimento posso conoscere la luminosita' integrata corrispondente // ai run e lumisection dei dati reali usati per l'analisi e al trigger di selezione usato ("HLT_ZeroBias"): **T**DATA // L_dt = 469,996 microbarn^-1 $L_{\rm int}$ 11 // La stima della luminosita' integrata corrispondente al Monte Carlo usato ("Minimum Bias") // va invece calcolato a mano: // - # eventi di MB : N = 51602200 N^{GEN} // - sezione d'urto per MB con il generatore Pythia: Sigma = 71,26 millibarn = 71260 microbarn T MC evt // - luminosita' integrata : L_mc = N/Sigma = 51602200/71260 (microbarn^-1) = 724,140 microbarn^-1 MinBias σ_{Pythia} 11 // Ne deriva il fattore scala dati/mc : SF(dt/mc) = 469,996/724,140 = 0.64904 17 // Nota bene: se ci fossero state delle ragioni per cui il generatore di eventi QCD "Pythia" non descrive correttamente una componente si potrebbe opportunamente scalare quella 11 J_{SCALE} specifica componente per un altro fattore di scola. 11 11 Double_t ScaleLum1 = 0.64904: cout << "Absolute Normalization Scale Real/Simulated =" << ScaleLumi << endl;</pre> η. 11 // Tipicamente si usa scalare il Monte Carlo (non i Dati): 17 hFake->Scale(ScaleLumi); hGhost->Scale(ScaleLumi); hLightHadrons->Scale(ScaleLumi); hCharmFlavour->Scale(ScaleLumi); hBeautyFlavour->Scale(ScaleLumi);

STEP-1 : normalizzazione assoluta – 2

Create la sotto-directory **/home/username/esercitazione-1/step1/**, dove adesso vi copio la *macro* **main.C**, mentre voi create al suo interno la sottodir. **Plots**, ovvero dove la macro – in esecuzione - scrivera' i file contenenti i plot.

Stavolta la macro in linguaggio C/C⁺⁺ si esegue con :

root [0] .x main.C("7nov","png","SOFT","qtimespt","doub","1gpv","Log","Step1")

Una volta eseguita la *macro*, visualizzare il file con i plot:

[username@alexis-corso step1]\$ display ./Plots/qtimespt_SOFT_7nov_Step1_Log_scaleLumi.png

Si potra' apprezzare che le 5 distribuzioni simulate sono effettivamente scalate (notate il cambio di scala in ordinata rispetto al plot precedente). Si noti che che il numero di *entries* rimane lo stesso nel box della statistica!!!! (il che puo' ingenerare confusione ... ma basta guardare gli overflow/underflow che non sono piu' interi per essere certi dello scaling).

ESERCIZIO: provare a fare la normalizzazione relativa (confronto fra shape)

STEP-2 : sovrapposizione con *stacking* - 1

```
if (err == "Step2")
   11
    aStyle->SetOptStat(kFALSE); // le statistiche non servono piu' a questo punto !
   // ma l'opzione funzionera' solo forzando lo stile corrente ne; punto qiusto [vedi oltre (*)]
   11
   // if(scale == "Log") gStyle->SetOptLogy(); non funziona...
   // ma serve ripristingre la scala logaritmica (se scale == "Log") [vedi oltre (**)]
   11
   11
   // preparo lo stacking del MC
   11
   hFake->UseCurrentStyle(); // forzo lo stile corrente (*)
   TH1D *h1 = (TH1D*)hFake->Clone("h1");
   TH1D *h2 = (TH1D*)h1 ->Clone("h2");
   TH1D *h3 = (TH1D*)h2 ->Clone("h3");
   TH1D *h4 = (TH1D*)h3 \rightarrow Clone("h4");
   11
   // si notino i pesi unitari nella combinazione lineare (->somma aritmetica)
   h1->Add(hFake,hGhost,1.1.);
                                       // h1 ha 2 componenti sommate (fake+qhost)
   h2->Add(h1,hLightHadrons,1.,1.); // h2 ha 3 componenti sommate (fake+ghost+light)
   h3->Add(h2,hCharmFlavour,1.,1.); // h3 ha 4 componenti sommate (fake+ghost+light+charm)
   h4->Add(h3,hBeautyFlavour,1.,1.); // h4 ha tutte e 5 le componenti sommate
   11
   TH1D *h5 = (TH1D*)h4->Clone("h5"); // distribuzione MC totale // serve in sequito
   11
   // nello stacking l'ordine delle componenti visibili sara' (from bottom to top)
   11
   11
        hFake–>fake,
                          h1->ghost,
                                         h2->light,
                                                          h3->charm,
                                                                           h4->beauty
   11
   // adesso scelgo i colori gia' scelti in precedenza:
   // fake=grigio(14), ghost=verde(3), light=rosso(2), charm=giallo(5), beauty=ciano(7)
   11
   hFake->SetFillColor(14);
   h1->SetFillColor(3);
   h2->SetFillColor(2);
   h3->SetFillColor(5);
   h4->SetFillColor(7);
   11
   // alcune altre opzioni (colore del bordo, spessore del tratto del bordo)
   11
   hFake->SetLineColor(1); hFake->SetLineWidth(1.2);
   h1->SetLineColor(1); h1->SetLineWidth(1.2);
   h2->SetLineColor(1); h2->SetLineWidth(1.2);
   h3->SetLineColor(1); h3->SetLineWidth(1.2);
   h4->SetLineColor(1); h4->SetLineWidth(1.2);
   11
   // per eliminare il titolo dell'istogramma
   hFake->SetTitle("");
   hData->SetTitle("");
   h1->SetTitle(""); h2->SetTitle(""); h3->SetTitle(""); h4->SetTitle("");
   11
```

STEP-2 : sovrapposizione con *stacking* - 2

```
MyC->Update();
   MyC->Clear();
   //MyC->Divide(1,1); // pleonastico
    11
   MvC->cd():
   if(scale=="Log") MyC->SetLogy(); // (**)
    if(scale == "Log") hData->SetMinimum(0.5); // per avere sotto controllo le code
    11
   hData->Draw("EP");
    11
   h4->Draw("same");
   h3->Draw("same");
   h2->Draw("same");
   h1->Draw("same");
   hFake->Draw("same");
    11
   h5->Draw("Esame"); // per mettere gli errori giusti del MC complessivo (somma di componenti)
    17
    hData->UseCurrentStyle();
                                // forzo lo stile corrente (*)
   hData->SetMarkerStyle(20);
   hData->SetMarkerColor(1);
    17
   hData->Draw("Esame");
    gPad->RedrawAxis(); // serve perche' la colorazione puo' coprire, come in questo caso, parte dell'asse y a sinistra
    11
   MyC->SaveAs("./Plots/"+par+"_"+muonType+"_"+date+"_"+err+"_"+scale+"_stacked."+extens);
 }
11
```

STEP-2 : sovrapposizione con *stacking* - 3

Create la sotto-directory **/home/username/esercitazione-1/step2/**, dove adesso vi copio la *macro* **main.C**, mentre voi create al suo interno la sottodir. **Plots**, ovvero dove la macro – in esecuzione - scrivera' i file contenenti i plot.

Stavolta la macro in linguaggio C/C⁺⁺ si esegue con :

root [0] .x main.C("7nov","png","SOFT","qtimespt","doub","1gpv","Log","Step2")

Con...

[username@alexis-corso step2]\$ *display ./Plots/qtimespt_SOFT_7nov_Step2_Log_stacked.png* ... si ottiene:



STEP-3 : sistemazione grafica finale - 1

Create la sotto-directory **/home/username/esercitazione-1/step3/**, dove adesso vi copio la *macro main.C*, mentre voi create al suo interno la sottodir. **Plots**, ovvero dove la macro – in esecuzione - scrivera' i file contenenti i plot.

```
Stavolta la macro in linguaggio C/C<sup>++</sup> si esegue con :
```

root [0] .x main.C("7nov","png","SOFT","qtimespt","doub","1gpv","Log","Step3")

Il plot si ottiene con:

[username@... step3]\$ display ./Plots/qtimespt_SOFT_7nov_Step3_Log_stacked_final.png



STEP-3 : sistemazione grafica finale - 2



STEP-4 : rapporto dati/MC - 1

Work in progress

STEP-4 : rapporto dati/MC - 2

Work in progress

Esercizio per casa (e' stata la prova pratica del febbraio 2013)

A partire dai files usati nella prima lezione pratica, cioe' : *Histos_Mc_MinBias_1aprnew_goodZB_last.root* (file di Monte Carlo) *Histos_Data_ZeroBias_1aprnew_goodZB_last.root* (file di Dati Reali) modificare la macro di ROOT usata in tale lezione (si usi anche il file di stile *tdrstyle.C*) al fine di ottenere il seguente plot di confronto dati-MC per la variabile *signif2d* (che e' la significativita' del parametro d'impatto trasverso per i muoni *TIGHT* di *minimum bias*):



Inoltre ricavare il seguente plot del rapporto dati/MC per la stessa osservabile:

Provare infine a ricavare il seguente plot del rapporto riorganizzando i bin in modo piu' significativo (evitando cioe' di dipendere troppo dalle fluttuazioni) [suggerimento: usare il metodo *Rebin* di TH1D]:

