

# Çözümleme Örneđi

Gökhan Ünel / *UC Irvine*

**HPFBU okulu - Ocak 2009**

10 Ocak 2009

# İlk dersi hatırlayalım

---

## ❧ LHC başlamak üzere

❧ Yeni parçacıkların keşfi - heyecan verici

## ❧ Yeni çarpıştırıcılar plan aşamasında (LHeC, ILC, CLIC..)

❧ Bu aletlerin imkanlarını araştırmak lazım

## ❧ O halde bize ne gerekiyor? (Bir konu seçtikten sonra)

❧ Ölçümü önermek

❧ Çarpıştırıcıyı tanımak

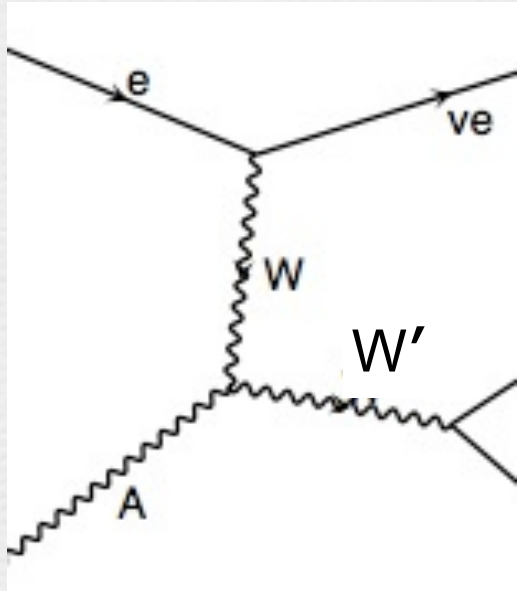
❧ Modeli bilgisayara girmek ve MC olay üretmek

❧ Öykünme, çözümleme, değerlendirme

# Konu

- Kimi SM ötesi modeller (örnek GUT)  $W$  benzeri yeni bir vektör bozon öneriyorlar :  $W'$
- Bu yeni bozonun
  - etkileşimleri SM ile aynı
  - kütlesi  $W$  dan daha ağır
- Bir  $\gamma$  e çarpıştırıcısı yapsak bu yeni parçacığı bulabilir miyiz?

• üretim örneği



bozunma ürünlerinden başlayıp keşif yapabilir miyiz?

Örnek jet kanalı:  
sinyal=  $2j$ +MET

# Nasıl yapılır?

## Uğraşmaya değer mi?

- Modelimizi bir üretici içine yerleştirelim
- 1 yıl içinde ne kadar olay bekliyoruz?
- Sinyal hangi izi bırakıyor?
- SM den gelecek olan arkaplan olayları neler?

Sinyalin en belirgin özelliği ne?

## Sinyal için MC olayları üretelim

- Algıç simülasyonu yapalım.
- Sinyali “kolayca” arkaplandan nasıl ayırt ederim?
- Sinyali yeniden oluşturabiliyor muyuz?

Sinyali nasıl tetikleyelim?

## SM de MC olayları üretelim, arkaplan için

- Algıç simülasyonu yapalım.
- Sinyal hala “görülüyor” mu?

Cevap “Evet” ise deneyi yapmaya başlayalım..

# Hedef ve Aletler

sinyal

$$\gamma e \rightarrow W' \nu \rightarrow 2j + \text{MET}$$

ardalan=arkaplan=gürültü

$$\gamma e \rightarrow 2j + \text{MET}$$

• Cain, GuineaPIG

• Işınlık hesabı

• Pythia, CompHEP, MadGraph

• Olay üretimi

• Pythia

• Parton yağmuru, hadronlaşma..

• PGS

• Algıç simülasyonu

• ROOT

• Veri incelemesi

# Yapılacak işler

## 1. Sinyal olaylarını - öğrendiğimiz aletlerle üretelim:

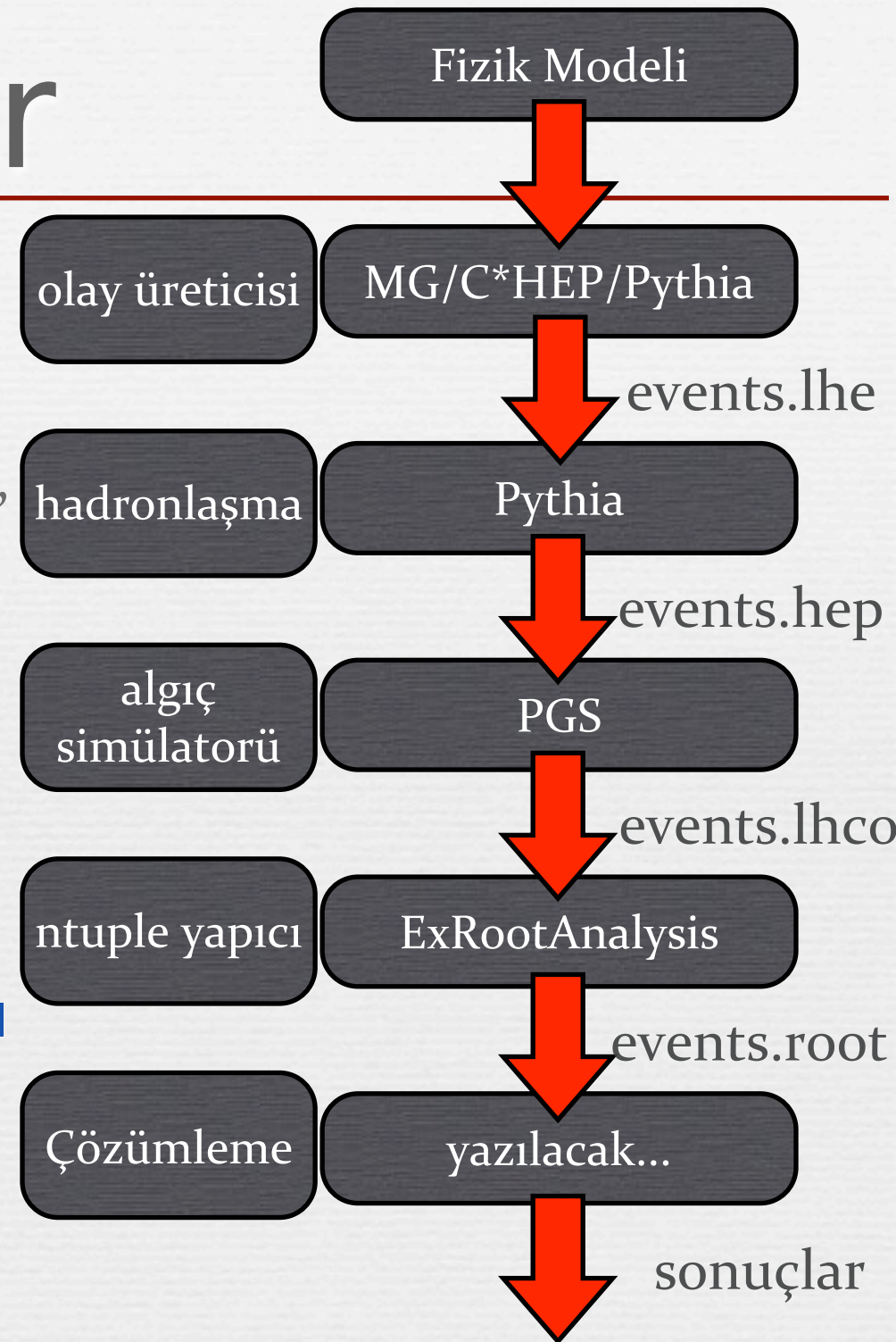
1.1.  $\gamma$   $e \rightarrow W' \nu \rightarrow 2j + \text{MET}$  CHEP'den,

1.2.  $\gamma$   $e$  ışınlığını CAIN'den bulalım.

## 2. MG ile ardalanı hesaplayalım.

## 3. Hem sinyali hem ardalanı algıç simülasyonundan geçirelim.

## 4. Bir çözümleme (analiz) programı yazıp sinyali gürültüden (ardalandan) ayırt edebiliyor muyuz bakalım.



# SM'e ek

• CompHEP başlangıcındaki listeden seçin:

```
CREATE NEW MODEL
```

• Yeni modelin adı Wp olsun, örnek olarak “SM, Unitary gauge” seçin.

• Yeni modeli seçip değiştirmeye başlayalım

```
FOURTH FAMILY ANOM, U  
_Wprime  
CREATE NEW MODEL
```

• Yeni bozonu ekleyelim (Particles)

• W ile kütlesi ve genişliği dışında aynı olmalı

```
Enter Decay Process  
Enter Scattering Process  
Edit Beams Table  
Edit Str. Functions Table  
Edit Model
```

z	boson	z	z	z	z	z	z	z	z
W	boson	W+	W-	2	MW	wW	1	W^+	W^-
Wp	boson	Wp	Wm	2	MWp	wWp	1	W'^+	W'^-

• Wp nin özelliklerini değişken olarak ekleyelim (Variables)

• m=770 varsayım, genişlik tahmin:

wH	0.0061744	width of Higgs
MWp	770	Wp nin kutlesi
wWP	10	Wp nin genisligi

• “Constraints” gerekmiyor,

• Wp jetini kolaylık için tanımlıyalım (Composite)

Clr	Rest	Del	Size
Abr	> elementa		
j1	u, U, d, D, G		
j2	u, U, d, D, s,		
j3	u, U, d, D, s,		
jb	b, B		
JW	W-, W+		
JWp	Wp, Wm		

# SM'e ek .. devam

## “Lagrangian” da W olan bütün satırları

• kopyala ve yapıştır

• kopyalarda bütün W özelliklerini Wp özelliklerine çevir:

P1	P2	P3	P4	>	Factor	< >	dLagrangian/
A	W+	W-			-EE		m3.p2*m1.m2-m1.
A	Wp	Wm			-EE		m3.p2*m1.m2-m1.

## Buraya kadar yaptıklarımızın doğruluğunu yoklayalım:

• bir hata varsa uyarı bilgisi verilir

```
Error in table ' Particles ' line 5 field 'width'  
unknown variable wWp  
Press any key
```

```
el of automatization.  
ne F2 key to get the informa  
e facilities and the F1 key  
elp.
```

• önceki sayfada küçük-büyük harf hatası var!!

```
Save corrections ?  
( Y / N ? )
```

WH	0.0061744	width of Higgs
MWp	770	Wp nin kutlesi
wWP	10	Wp nin genisligi

wWp olmalıydı!!!



$$W' \rightarrow x y$$

# W' genişliği:

- yeni model bir editör ile de girilebilir
- Model girildikten sonra W' genişliği hesap edelim
- $m=770$  GeV de  $\Gamma=25.5$  GeV
  - $8.5 \times 3 = 25.5\%$  leptonsal
  - 74.5% hadronsalsal bozunum
    - 23% top quark içeriyor
    - bizim sinyale uymuyor...

```
E lne lW- l l-EE*sqrt2/(4*SW) lG(m3)*(1
E lne lWm l l-EE*sqrt2/(4*SW) lG(m3)*(1
G lG lG l lGG l m2.p3*m1
G.C lG.c lG l lGG l m3.p2
H lH lH l l-3*EE*MH^2/(2*MW*SW) l1
H lW+ lW- l lEE*MW/SW l m2.m3
H lWp lWm l lEE*MWp/SW l m2.m3
```

```
Enter decayed particle: Wp
Enter Final State: Wp -> 2*x
```

```
EE= 0.31345
SW= 0.48076
s12= 0.2229
s23= 0.0412
s13= 0.0036
Mm= 0.10566
Mtau= 1.777
Mc= 1.65
Ms= 0.117
Mtop= 174.3
Mb= 4.85
MWp= 770
```

Process: Wp -> 2\*x (12 subprocesses)

Total width : 2.554667E+01 GeV			
Modes and fractions :			
t B -	23%	u D -	24%
nl L -	8.5%	ne E -	8.5%
c B -	0.043%	u S -	1.3%
u B -	0.00033%	S t -	0.039%
		c S -	24%
		nm H -	8.5%
		D c -	1.3%
		D t -	0.00076%

## Şimdi çarpışmalara bakalım

• içeri girenler:

```
1st Beam: ebeam
1st Beam Energy (GeV) : 500.000000
2nd Beam: gamL
2nd Beam Energy (GeV) : 500.000000
1st beam PDF number: 3
2nd beam PDF number: 2
```

• Dışarı çıkanlar:

```
State: ebeam, gamL -> ne, Pbm
ograms with
ums with
```

W' bozununca ~51% i 2jet sinyali veriyor

$$e \gamma \rightarrow W' \nu$$

# $W'$ oluşumu - yılda kaç tane?

## Bozunmamış sinyal

tesir kesitini hesaplayalım ve Kinematik özelliklere bakalım

20	1.6428E-03	4.53E-01	9792	
< >	1.6458E-03	9.64E-02	195840	0.6

1.64fb tesir kesiti bekleniyor

ILC Işınlık  $2.8 \times 10^{34} / \text{cm}^2 / \text{s}$

$e\gamma$ :  $\sim 2.8 \times 10^{33} / \text{cm}^2 / \text{s}$

1 yıl  $\sim 10^7$  saniye  $\Rightarrow e\gamma$ :  $\sim 28 \text{fb}^{-1}$

yıllık beklenen  $W'$  sayısı

$1.64 \times 28 \sim 46$  tane

50% istediğimiz şekilde bozunacak

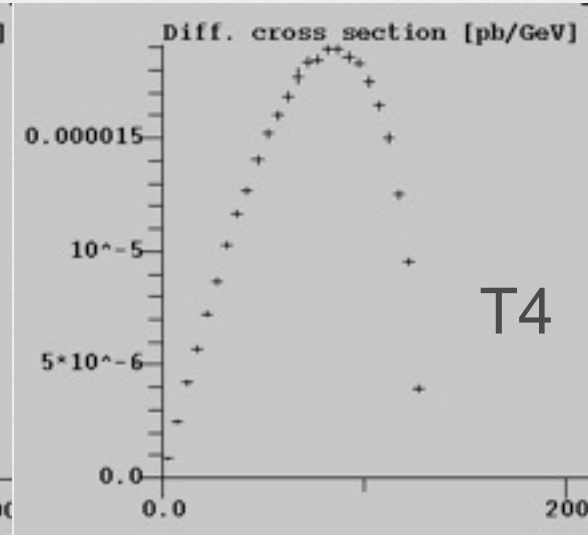
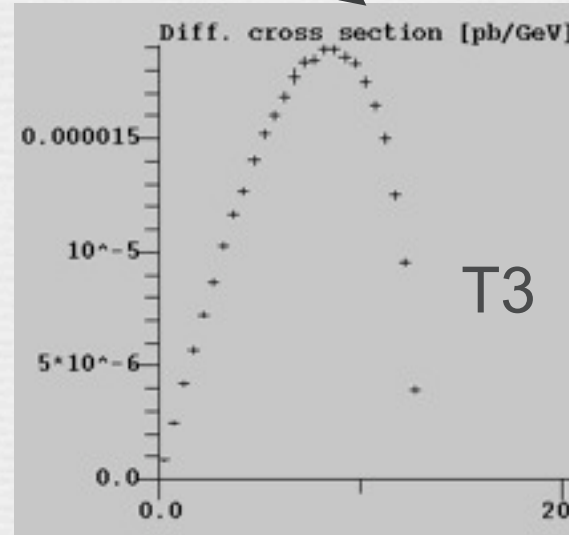
yılda 23 olay bekliyoruz

etkin tesir kesiti:  $1.64 \times 50\% \approx 0.8 \text{fb}$

(sub)Process: e, A -> Wm, ne

Distributions

Clr	Rest	Del	Size
Parameter	>	Min bound	< > Max
E12		500	1000
T3		0	500
T4		0	500



# W' bozunumu - sağlama

$$e \gamma \rightarrow W' \nu \rightarrow jj \nu$$

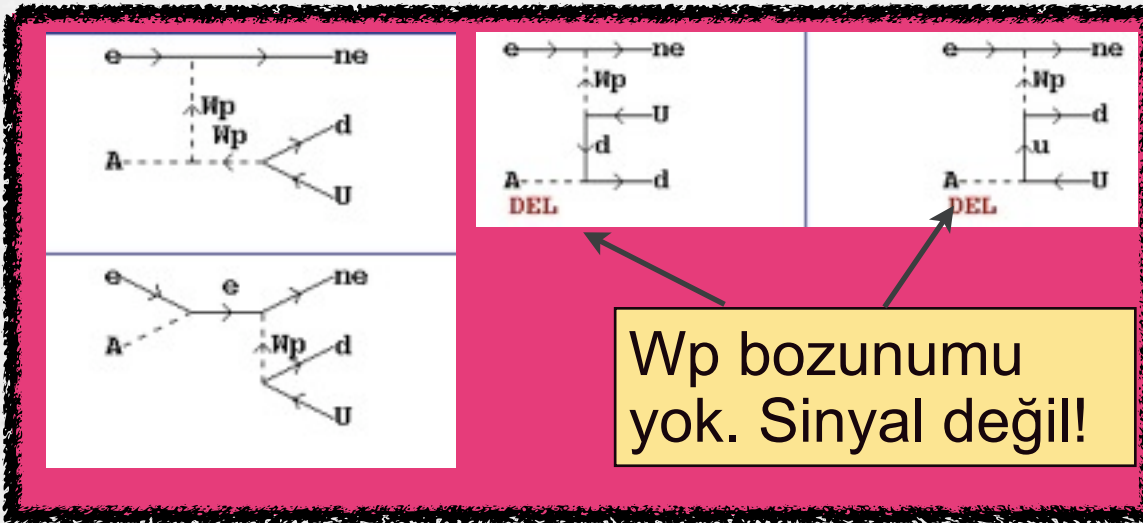
W' hadronsalsal bozup, sadece "sinyal kısmına bakmak" için biraz *akıllı tercih* biraz da *temizlik* gerekli:

```
Final State: ebeam, gamL -> ne, j3, j3
Delete diagrams with [redacted]
Delete diagrams with Wp, Wm
```

Process: ebeam, gamL -> ne, j3, j3

Feynman diagrams  
24 diagrams in 6 subprocesses are constructed.  
0 diagrams are deleted.

NN	Subprocess	Del	Rest
1	e, A -> ne, U, d	0	4
2	e, A -> ne, U, s	0	4
3	e, A -> ne, U, b	0	4
4	e, A -> ne, d, C	0	4
5	e, A -> ne, C, s	0	4
6	e, A -> ne, C, b	0	4



```
S.F.1: ISR(100 Beamstr.: 560, 0.40, 2.0E+10)
S.F.2: Laser photons
First particle momentum[GeV] = 500
Second particle momentum[GeV] = 500
```

< >	4.0409E-04	2.78E-01	153600	0.8
< >	2.1101E-05	2.51E-01	153600	0.7
< >	5.5292E-09	2.51E-01	153600	0.8
< >	2.1108E-05	2.48E-01	153600	0.5
< >	4.0133E-04	2.65E-01	153600	1
< >	7.2496E-07	2.56E-01	153600	1

✓ toplam tesir kesiti = 0.82fb, beklendiği gibi

# Ardalan - tesir kesiti ve olay üretimi

olay üreticisi

## • Olayları üretin (örnek: MadGraph)

- `cd Events ; ls`
- `unweighted_events.lhe` üretilmiş olmalı

hadronlaşma

## • Hadronlaşma yaptırın

- `../bin/run_pythia ; ls`
- `pythia_events.hep` üretilmiş olmalı

algıç simülatörü

## • Algıç öykünmesi

- `../bin/run_pgs ; ls`
- `pgs_events.lhco` üretilmiş olmalı

ntuple yapıcı

## • ROOT dosyasını hazırlayın

- `../../ExRootAnalysis/ExRootLHColympicsConverter pgs_events.lhco sm.root ; ls`
- `sm.root`

bu dosyayı inceleyin, çözümleyin.

MG ödevinde yapıldı!

# W' olay üretimi

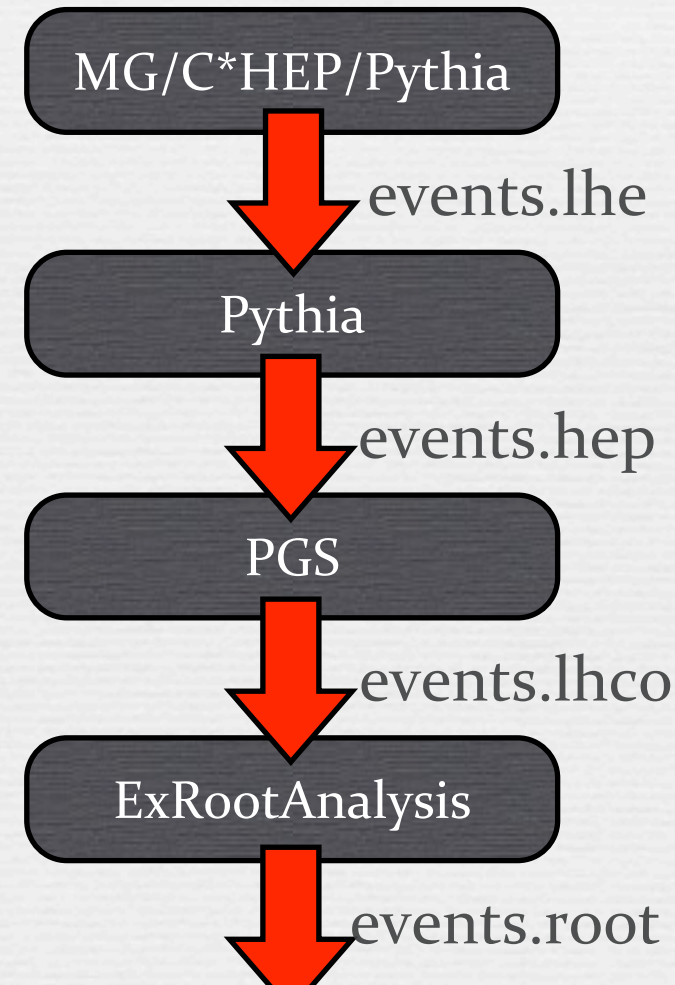
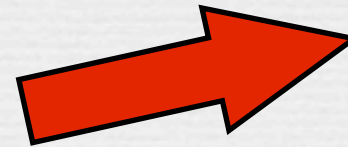
- Sinyal olaylarını bütün alt süreçlerden üretmeye gerek yok. Tesir kesiti  $10^{-7}$  pb ve daha az olanlardan olay üretmeyelim.
- geriye kalan 4 alt süreçten olay üretip harmanlayalım.

• cpyth ile

```
ngus$ ../mix_flows.exe events_9.txt events_10.txt events_12.txt e  
le events_9.txt exhausted): 10399
```

- Basit bir program yazıp üretim zincirine dahil edelim

• PEV biçiminde alıp, hadronlaşma sağlayıp, hep biçiminde yazalım.



# Çözümleme - her yiğidin yoğurt yiyişi farklıdır.

## Önce sinyale bakalım

- Laptop ile (ROOT altında) kullanabilelim ki gece uyku kaçınca çalışmaya devam edelim.

## ROOT dersinde öğrendik

- LHCO->MakeClass("HPFBU\_a")
- Basit 1-2 değere bakalım



Şimdilik sizler

Bizler

```
if (fChain == 0) return;
```

### Declarations

```
TH1F *jmult, *lmult;  
TH1F *jeteta, *jetphi;  
TH1F *jetPT, *MET;
```

### Definitions

```
jmult= new TH1F("jmult" , "Jet multiplicity",20, -0.5,19.5);  
lmult= new TH1F("lmult" , "Lepton multiplicity",20, -0.5,19.5);  
jeteta= new TH1F("jeteta" , "Jet #eta",50, -5.,5.);  
jetphi= new TH1F("jetphi" , "#phi of jets",50, -5.,5.);  
jetpt= new TH1F("jetpt" , "Jet P_{T}",40,0.,400.);  
ptmss= new TH1F("ptmss" , "Missing P_{T}",50,0.,250.);
```

```
nb = fChain->GetEntry(jentry);  
// if (Cut(tentry) < 0) continue  
  
// begin analysis  
jmult->Fill(Jet_);  
lmult->Fill(Muon_+Electron_);  
  
} // END of event loop  
jmult->Draw();
```



# Çözümleme 3

• Bir pencere yapalım:

```
// pencereler
TCanvas *c1 = new TCanvas("inv_mass_plots", "c1", 150, 10, 650, 500);
c1->SetFillColor(0); c1->SetBorderMode(0); c1->Divide(2,2,1e-5,1e-5);
c1->Draw();
```

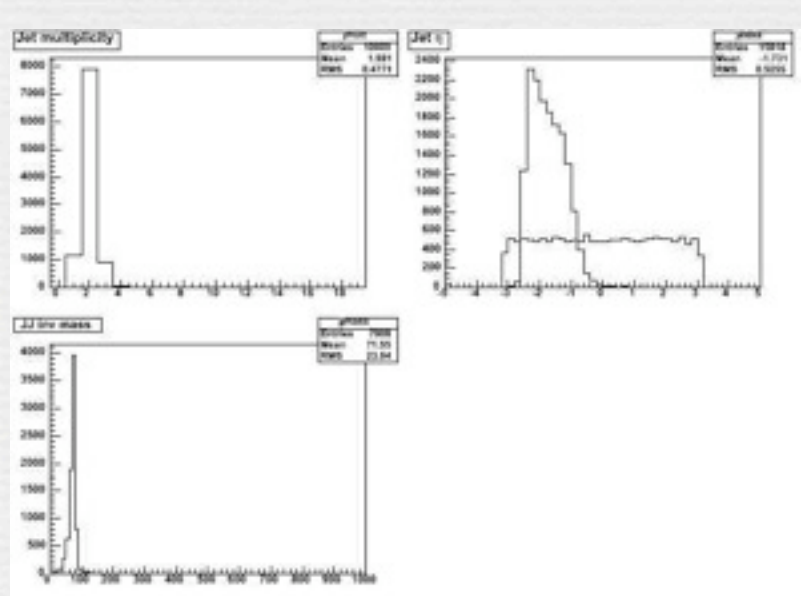
• 2 jet değişmez kütlelerini hazırlayalım:

```
TH1F *jetPT, *ptmss;
TH1F *jjmass;
```

```
ptmss= new TH1F("ptmss" , "Missing P_{T}", 50, 0., 250.);
jjmass= new TH1F("jjmass" , "JJ inv mass", 100, 0., 1000.);
```

• Hesaplayıp çizdirelim:

```
}
MJJ=jets[0]+jets[1];
jjmass->Fill(MJJ.M());
} // END of event loop
c1->cd(1);
jmult->Draw();
c1->cd(2);
jeteta->Draw();
jetphi->Draw("same");
c1->cd(3);
jjmass->Draw();
```





# Ardalan=gürültü

- Aradığımız ize benzer iz bırakan
- Ama aradığımız nesneyi ( $W'$ ) içermeyen herşey
  - bunlardan kurtulmak lazım
- BSM araştırmasında SM gürültüdür
- Samanı iğne'den ayıracak bir takım özellikler bulmak gerekli
  - “CUT” = “Tırpan”
- Hangi özellikler kullanılabilir?
  - $W'$  ağır olduğu için bundan çıkan jetler de yüksek  $p_T$  li olması beklenir
  - $M_{jj}$  kullanılamaz.



# Çözümleme 4

Jetlerin pT sini çizdirelim ve resmi kaydedelim.

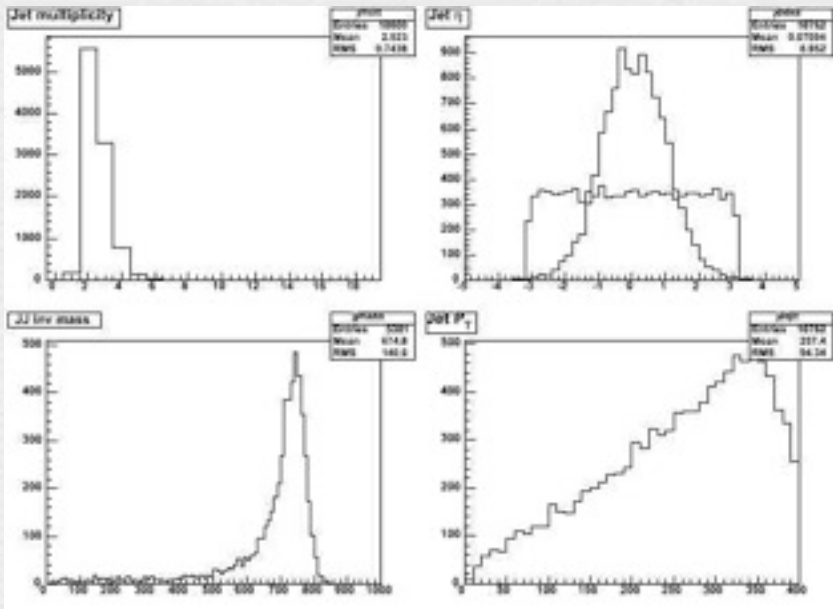
```
c1->cd(4);  
jetpt->Draw();  
c1->SaveAs("inceleme.jpg");
```

Ürettiğimiz histogramları saklayalım:

```
// save histograms  
TFile bb("sonuclar.root", "recreate");  
jetpt->Write();  
jjmass->Write();  
bb.Close();
```

Her seferinde yazmaktansa kendi kendine çalışsın.

root -q -x runme.C



```
void runme () {  
gROOT->LoadMacro("HPFBU_a.C");  
HPFBU_a si;  
si.Loop();  
return;  
}
```

# Tırpanlar



Hangi tırpan ile kaç olay kaybettik?

Her tırpandan sonra kalanları sayalım ve saklayalım:

```
// begin analysis
eff->Fill(1);
jmult->Fill(Jet_);
lmult->Fill(Muon_+Electron_);
if (Jet_ != 2) continue;
eff->Fill(2);

MET->Fill(MissingET_MET[0]);
if (MissingET_MET[0] < 20 ) continue;
eff->Fill(3);

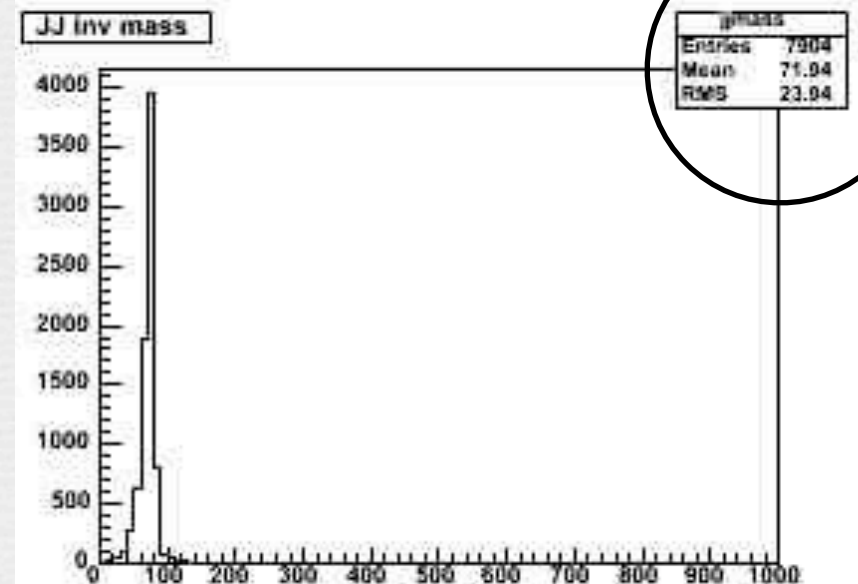
for (int i=0; i<2; i++)
```

```
TH1F *eff= new TH1F("eff", "Tırpanlar", 16, 0.5, 16.5);
eff->GetXaxis()->SetBinLabel(1, "all");
eff->GetXaxis()->SetBinLabel(2, "2 jets");
eff->GetXaxis()->SetBinLabel(3, "MET");
eff->GetXaxis()->SetBinLabel(4, "PTjets");
```

2 jet tırpanından kurtulan olayları sayar.

kurtulan olay sayısı

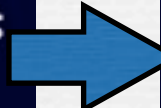
Daima önce çiziniz sonra kesin. Ne kaybettiğinizi bilin.



# si & sm

.h dosyasını deęiřtirelim

```
HPFBU_a(TTree *tree=0);  
virtual ~HPFBU_a();  
virtual Int_t    Cut(Long64_t entry);  
virtual Int_t    GetEntry(Long64_t entry);  
virtual Long64_t LoadTree(Long64_t entry);  
virtual void     Init(TTree *tree);  
virtual void     Loop();  
virtual Bool_t   Notify();  
virtual void     Show(Long64_t entry = -1);  
};  
  
#endif  
  
#ifdef HPFBU_a_cxx  
HPFBU_a::HPFBU_a(TTree *tree)
```



```
HPFBU_a(TTree *tree, char cname[128]);  
virtual ~HPFBU_a();  
virtual Int_t    Cut(Long64_t entry);  
virtual Int_t    GetEntry(Long64_t entry);  
virtual Long64_t LoadTree(Long64_t entry);  
virtual void     Init(TTree *tree);  
virtual void     Loop(char cname[128]="generic");  
virtual Bool_t   Notify();  
virtual void     Show(Long64_t entry = -1);  
};  
#endif  
  
#ifdef HPFBU_a_cxx  
HPFBU_a::HPFBU_a(TTree *tree, char cname[128])  
{  
    // If parameter tree is not specified (or zero), con  
    // used to generate this class and read the tree.  
    if (tree == 0) {  
        if (strcmp(cname, "sm")==0) {  
            TChain * chain = new TChain("LHCO", "");  
            chain->Add("SM_jjMET.root");  
            tree = chain;  
        } elseif (strcmp(cname, "si")==0 ) {  
            TChain * chain = new TChain("LHCO", "");  
            chain->Add("WSignal.root");  
            tree = chain;  
        }  
    }  
    Init(tree);  
}
```

.C dosyasını da..

```
void HPFBU_a::Loop(char cname[128])  
{  
    // save histograms  
    char aaa[128];  
    sprintf (aaa, "%s.root", cname);  
    TFile bb(aaa, "recreate");  
    jetpt->Write();  
}
```

# si & sm - karşılaştırma

runme.C de değişmeli

Çalıştıralım, sonuç:

```
22:32 si.root
22:32 sm.root
```

Şimdi sinyal ve ardalın olaylarını karşılaştırabiliriz..

Yeni C program yazalım.

si ve sm dosyalarını okusun

tesir kesitlerine göre yılda elde edilen olayları saysın

bu “deneyi” yapmış olsaydık kaç olay görürdük söylesin.

$$N = \mathcal{L} \times \sigma \times \epsilon$$

Işınlık

Tesir kesiti

Verim

```
void runme () {
gROOT->LoadMacro("HPFBU_a.C");

HPFBU_a sm(0, "sm");
sm.Loop("sm");
HPFBU_a si(0, "si");
si.Loop("si");

return;
}
```

```
void compare() {
Float_t si_cr=19; //fbarn
Float_t sm_cr=6.5; //fbarn
Float_t lumi=10; //tvinlik
Float_t sene=1; //zaman

TCanvas *ctst = new TCanvas("a1", "ctst", 960, 400);

TFile bg("sm.root", "");
TH1F *bgmjj = new TH1F(*jjmass);
TH1F *bgeff = new TH1F(*eff);
bgmjj->Scale( (lumi*sene*sm_cr) /bgeff->GetBinContent(1) );

TFile si("sm.root", "");
TH1F *simjj = new TH1F(*jjmass);
TH1F *sieff = new TH1F(*eff);
simjj->Scale( (lumi*sene*si_cr) /sieff->GetBinContent(1) );

TH1F *toplamjj = new TH1F(*bgmjj);
toplamjj->Add(simjj, bgmjj);
char aaa[128];
sprintf (aaa, "#events/%iGeV/%ifb^{ -1}", 10, lumi);
toplamjj->SetYTitle(aaa);
toplamjj->SetXTitle("M{jj} (GeV)");

ctst->cd();
bgmjj->SetLineColor(2);
simjj->SetLineColor(4);
toplamjj->SetLineColor(1);
toplamjj->Draw();
bgmjj->Draw("Same");
simjj->Draw("Same");
}
```

Tanımlar

```
void compare() {  
Float_t si_cr=19; //fbarn  
Float_t sm_cr=6.5; //fbarn  
Float_t lumi=10; //isinitk  
Float_t sene=1; //zaman
```

Pencere

```
TCanvas *ctst = new TCanvas("a1", "ctst", 960, 400);
```

senelik  
olay hesabı

```
TFile bg("sm.root", "");  
TH1F *bgmjj = new TH1F(*jjmass);  
TH1F *bgeff = new TH1F(*eff);  
bgmjj->Scale( (lumi*sene*sm_cr) /bgeff->GetBinContent(1) );
```

```
TFile si("sm.root", "");  
TH1F *simjj = new TH1F(*jjmass);  
TH1F *sieff = new TH1F(*eff);  
simjj->Scale( (lumi*sene*si_cr) /sieff->GetBinContent(1) );
```

toplam  
olay sayısı

```
TH1F *toplammjj = new TH1F(*bgmjj);  
toplammjj->Add(simjj,bgmjj);  
char aaa[128];  
sprintf (aaa, "#events/%iGeV/%ifb^{-1}", 10, lumi);  
toplammjj->SetYTitle(aaa);  
toplammjj->SetXTitle("M_{jj} (GeV)");
```

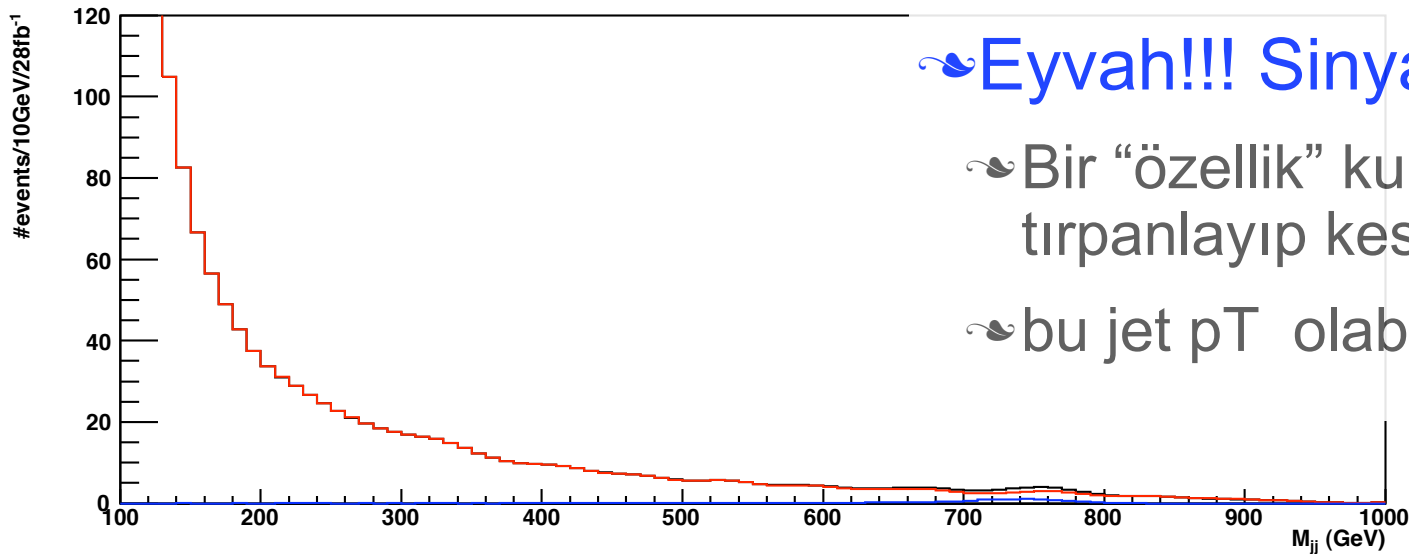
si + sm  
çizimi

```
ctst->cd();  
bgmjj->SetLineColor(2);  
simjj->SetLineColor(4);  
toplammjj->SetLineColor(1);  
toplammjj->Draw();  
bgmjj->Draw("same");  
simjj->Draw("same");  
}
```

verim =  
kurtulan olaylar  
-----  
baştaki olaylar

# Birleştirelim

W'



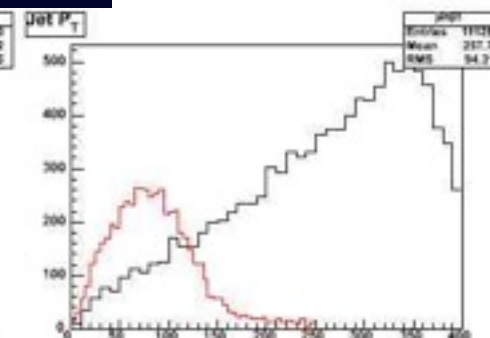
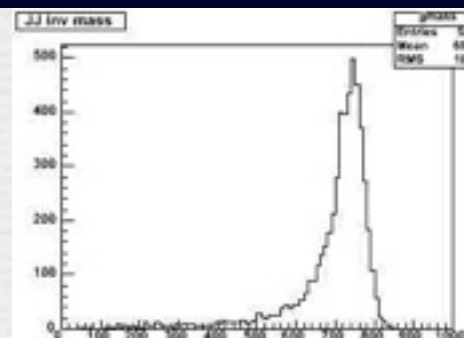
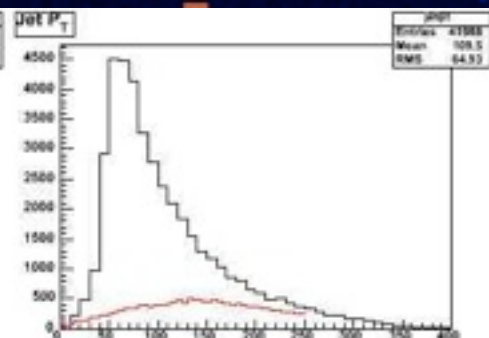
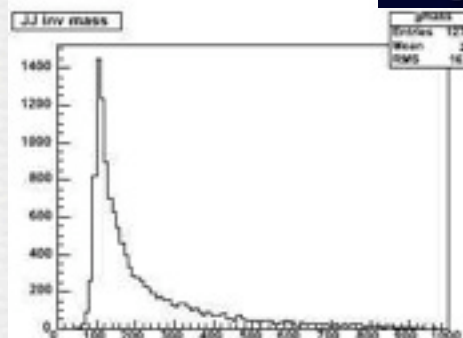
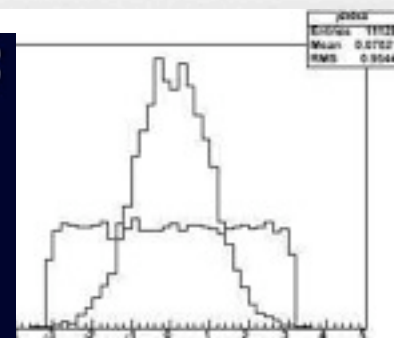
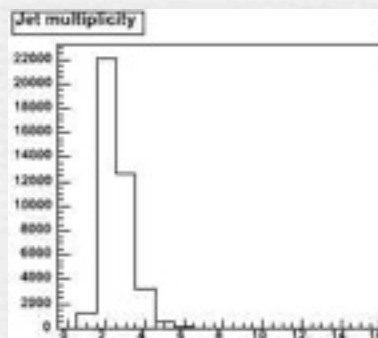
☞ Eyvah!!! Sinyal görünmüyor...

☞ Bir “özellik” kullanıp ardalara olaylarını tırpanlayıp kesmek lazım.

☞ bu jet  $p_T$  olabilir mi?

```
if (jets[0].Pt() < 250 && jets[1].Pt() < 250) continue;
eff->Fill(4);

MJJ=jets[0]+jets[1];
jjmass->Fill(MJJ.MO);
} // END of event loop
```

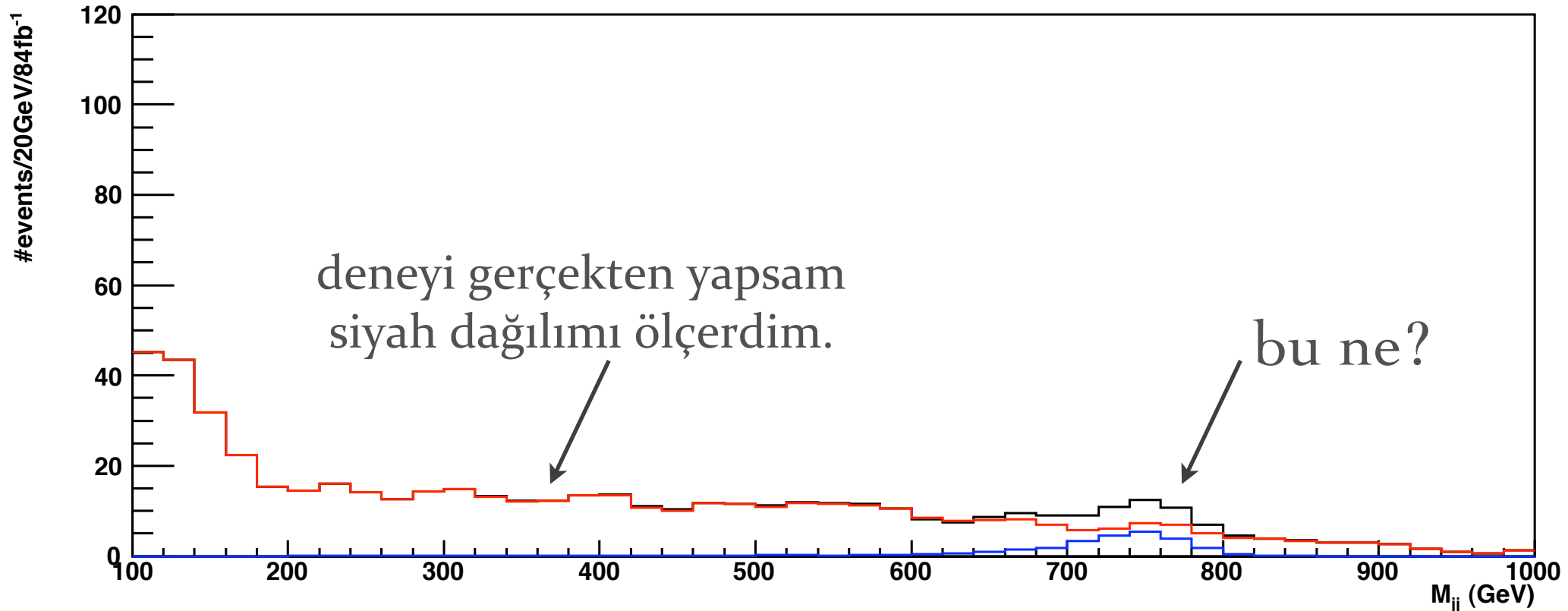


# Tırpanlayınca..

• Olay sayım azaldı,

• 3yıl bekleyim ve 20GeV'lik aralıklarla olayları sayayım.

W'





# Ölçüm gerçek mi?

• Anlamli mı yoksa istatistik hata mı?

- Ölçütler ( $\sigma$ )
- En az  $3\sigma \Rightarrow$  Gözlem
- En az  $5\sigma \Rightarrow$  Keşif

$$\left. \begin{array}{l} S / \sqrt{B} \\ S / \sqrt{S + B} \\ 2 \times (\sqrt{S + B} - \sqrt{B}) \\ \sqrt{2 \times \left[ (s + b) \ln \left( 1 + \frac{s}{b} \right) - s \right]} \end{array} \right\}$$

• Toplam olay eğrisine poly1+Gauss uydurup sinyal ve ardaan olay sayısını bulalım.

```
TF1 *total = new TF1("total", "pol1(0)+gaus(2)", 320, 920);
total->SetParameters(15, -0.5, 5, 750, 35);
total->SetFillColor(4);
total->SetLineWidth(2);
total->SetParLimits(2, 0, 10000);
total->SetParLimits(3, 700, 800);
total->SetParLimits(4, 10, 90);
```

```
toplamjj->Fit(total, "WR+");
```

```
Double_t par[5];
total->GetParameters(&par[0]);
```

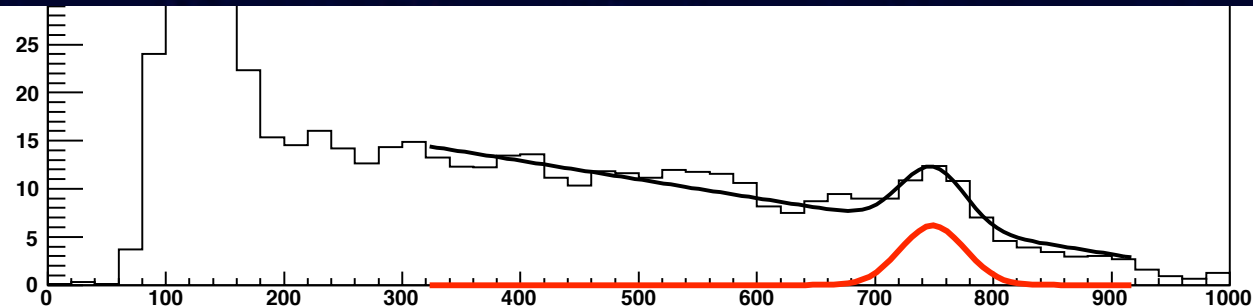
```
TF1 f2("f2", "gaus", 320, 920);
f2->SetParameters(&par[2]);
f2->SetLineColor(2);
f2->Draw("same");
```

```
ctst->SaveAs("fit.eps");
```

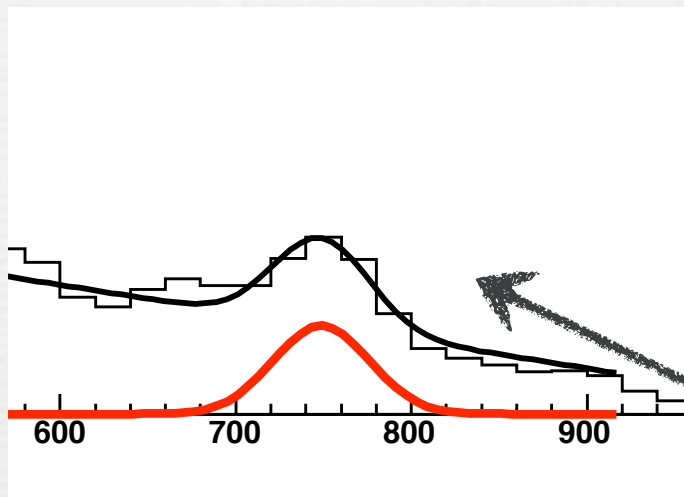
JJ inv mass

jjmass

```
Int_t ns=2;
Double_t nsig=(f2->Integral(par[3]-ns*par[4], par[3]+ns*par[4]))/(10*rb);
Double_t ntot=(total->Integral(par[3]-ns*par[4], par[3]+ns*par[4]))/(10*rb);
cout << "nsig=" << nsig << " ntot=" << ntot << " rbg=" << (ntot-nsig);
cout << " SIG=" << sqrt(2*(ntot*log(1+nsig/(ntot-nsig))-nsig)) << endl;
```



# Gerçek mi?



```
root [7] .x compare.C
Info in <TCanvas::Print>: eps file Kesif.eps has been
FCN=30.686 FROM MIGRAD STATUS=CONVERGED 613 C
EDM=2.2468e-11 STRATEGY= 1 B
EXT PARAMETER
NO. NAME VALUE ERROR STA
1 p0 2.06985e+01 7.16576e-01 -5.9564
2 p1 -1.94486e-02 1.18707e-03 9.3568
3 p2 6.17607e+00 8.51794e-01 -5.4922
4 p3 7.48826e+02 4.39496e+00 2.0032
5 p4 2.72674e+01 5.21748e+00 -4.4426
Info in <TCanvas::Print>: eps file fit.eps has been c
nsig=20.1461 ntot=53.6028 nbg=33.4567 SIG=3.19986
```

Bir sinyal gözledim!

- Keşif için daha çok veri kaydetmem lazım
- Bir kaç yıl daha deneyim çalışsın:

```
Float_t lumi=28;
Float_t sene=8;
```

VALUE	ERROR
5.51961e+01	7.08705e-01
-5.18629e-02	1.17498e-03
1.64695e+01	8.60333e-01
7.48826e+02	1.64761e+00
2.72675e+01	1.98693e+00

nsig=53.7231 ntot=142.941 nbg=89.2181 SIG=5.22535

**KESİF TAMAM !!!**

# Sonuç

- Artık yoğurt da sizin, kebab da sizin!
  - kendi usulünüzü bulana kadar yemeğe devam!!
  - LHC, SLHC, ILC, LHeC, CLIC.. sizleri bekliyor.
  - SuSY, GUT, ED, LH.. modelleri sizleri bekliyor.
- Çalışın ve başarılı olun.

