YMAP
Youth Meeting for AstroParticle

第4回 ROOT講習会
TGraph、TH2D

中村輝石（東北大）

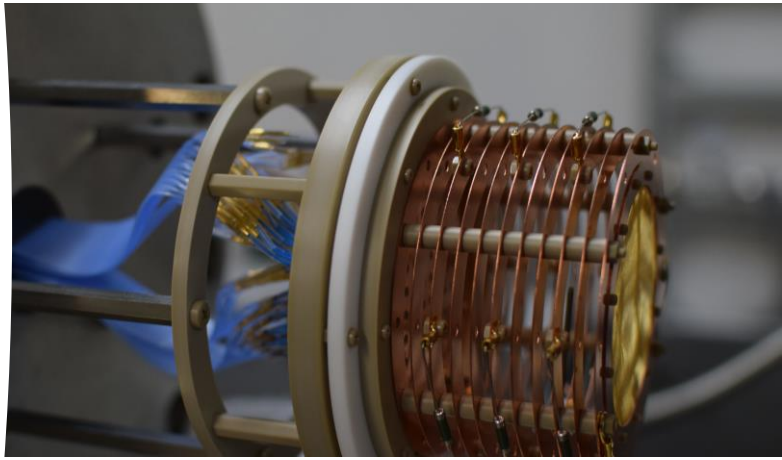
2021年度 第4回 ROOT講習会

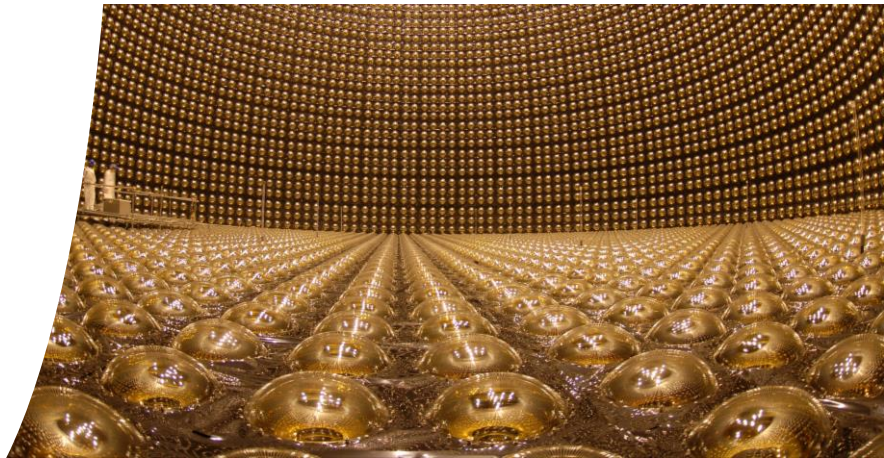
2021/6/3

自己紹介

- 名前：中村輝石（なかむらきせき）
- 所属：東北大学 素粒子実験（加速器）グループ
- 実験：AXEL、T2K、SK、HK、など
- 博士：NEWAGE実験で取得

神岡研究棟の廊下に貼ってある写真





はじめに

- わからないことがあれば，**講義を遮って良いのですぐ質問**してください。
  zoomの「手を挙げる」機能や，slackへの書き込みでもよいです。
- 簡単すぎる場合は途中退室してもよいです。
  わからない/ついてこれていない場合は質問してください。

- 本資料は，ROOTのClass Referenceを参考にしています。

  https://root.cern.ch/doc/master/classTGraph.html

  https://root.cern.ch/doc/master/classTH2.html

- 2019年度までの講師 奥村さんの資料も有用です。

  https://github.com/akira-okumura/RHEA

準備

- コード・データの最新版をダウンロードする

```
[keishi@mac ~] $ cd ROOT2021   ←前回までにgit cloneして作ったdirectory
[keishi@mac ~] $ git pull
```

- これでダメなら,

```
[keishi@mac ~] $ cd [ROOT2021を含むdirectory]
[keishi@mac ~] $ mv ROOT2021 ROOT2021_old
[keishi@mac ~] $ git clone https://github.com/ymap-team/ROOT2021.git
```

- ROOT2021のnkmr directoryに移動する

```
[keishi@mac ~] $ cd ROOT2021/nkmr
```

第4回講習会の到達目標

- TGraph
  - TGraphでグラフを自作/描画できるようになる
- TH2D
  - 2次元のヒストグラムを自作/描画できるようになる
- （実はTTreeのときに描いてるんだけどね）
- +α

TGraph

- x-y平面に点を打つクラス
- 線で繋げたりできる

ROOT
Reference Guide
Version master
Search

RHistBufferedFill.hxx
RHistConcurrentFill.hxx
RHistData.hxx
RHistDisplayItem.cxx
RHistDrawable.cxx
RHistImpl.hxx
RHistUtils.hxx
RHistView.hxx
TAxis
TBackCompFitter
TBinomialEfficiencyFitter
TConfidenceLevel
TEfficiency
TF1
TF12
TF1Convolution
TF1NormSum
TF2
TF3
TFitResult
TFormula
TFormula
TFormulaPrimitive
TGraph
TGraph2D
TGraph2DErrors
TGraphAsymmErrors
TGraphBentErrors
TGraphDelaunay
TGraphDelaunay2D
TGraphErrors
TGraphMultiErrors
TGraphSmooth
TGraphTime
TH1C
TH1D
TH1F
TH1I
TH1S

List of all members | Public Types | Public Member Functions | Static Public Member Functions | Protected Member Functions | Static Protected Member Functions | Protected Attributes | List of all members

TGraph Class Reference

Histogram Library

A **TGraph** is an object made of two arrays X and Y with npoints each.

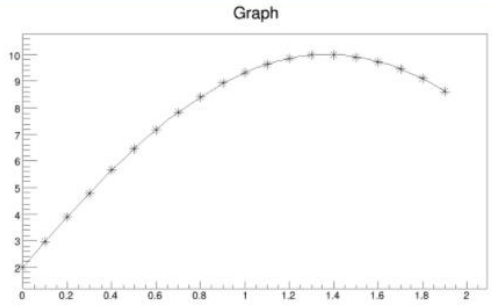
The **TGraph** painting is performed thanks to the **TGraphPainter** class. All details about the various painting options are given in this class.

Notes

- Unlike histogram or tree (or even **TGraph2D**), **TGraph** objects are not automatically attached to the current **TFile**, in order to keep the management and size of the **TGraph** as small as possible.
- The **TGraph** constructors do not have the **TGraph** title and name as parameters. A **TGraph** has the default title and name "Graph". To change the default title and name `SetTitle` and `SetName` should be called on the **TGraph** after its creation. **TGraph** was a light weight object to start with, like TPolyline or **TPolyMarker**. That's why it did not have any title and name parameters in the constructors.

The picture below gives an example:

```
{
   TCanvas *c1 = new TCanvas("c1","A Simple Graph Example",200,10,500,300);
   Double_t x[100], y[100];
   Int_t n = 20;
   for (Int_t i=0;i<n;i++) {
     x[i] = i*0.1;
     y[i] = 10*sin(x[i]+0.2);
   }
   TGraph* gr = new TGraph(n,x,y);
   gr->Draw("AC*");
}
```



TGraph：作り方その1

- マクロ graph.C の中身

```
{
  TGraph *g = new TGraph();
  g->SetPoint(0, 1.5, 0.511);
  g->SetPoint(1, 2.5, 106);
  g->SetPoint(2, 3.5, 1777);
  g->Draw();
}
```

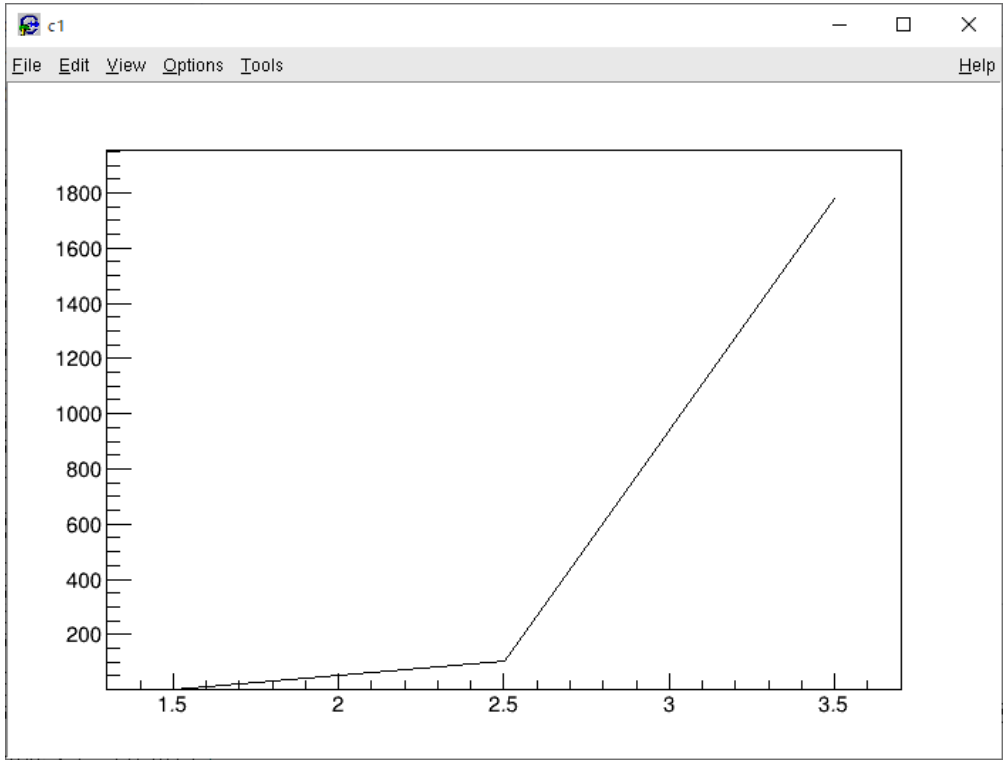
TGraphクラスのポインタを作成

点を登録

描画する

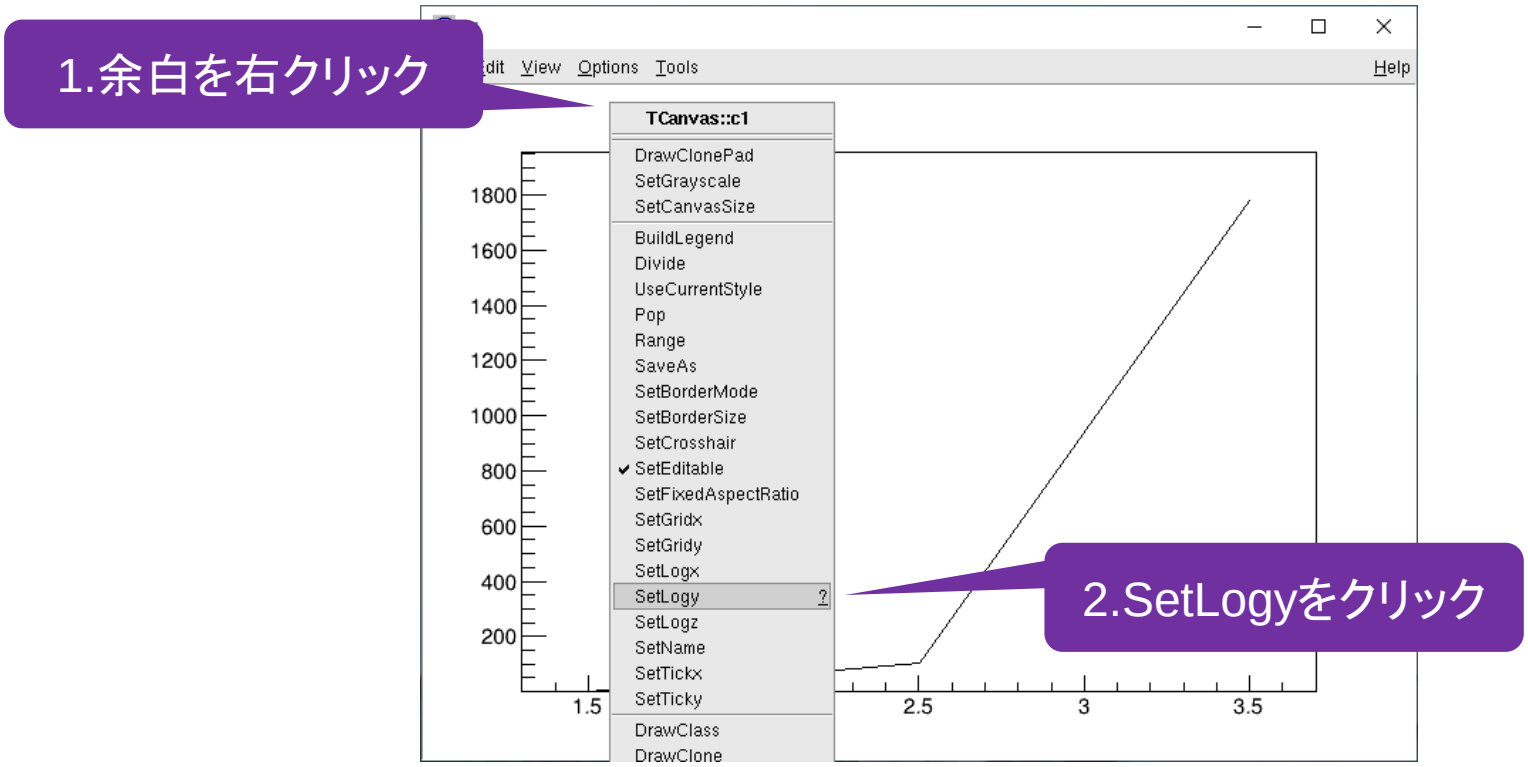
TGraph：作り方その1

- g->Draw()で描画



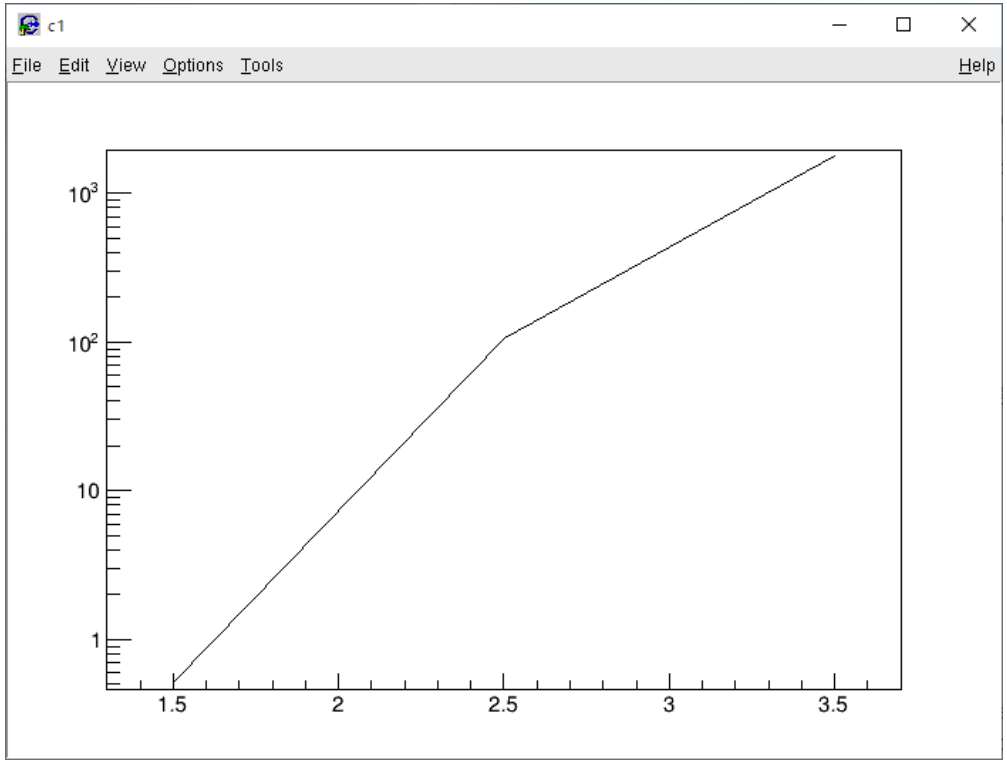
TGraph：作り方その1

- 縦軸をlogにする方法



TGraph：作り方その1

- 縦軸をlogにする方法

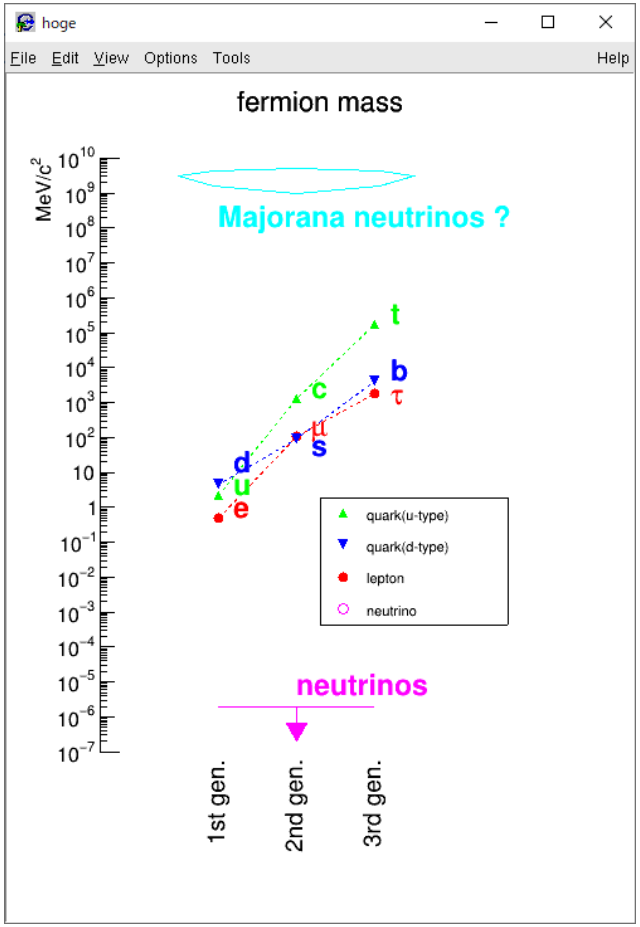


TGraph：装飾

- root graph_rich.C

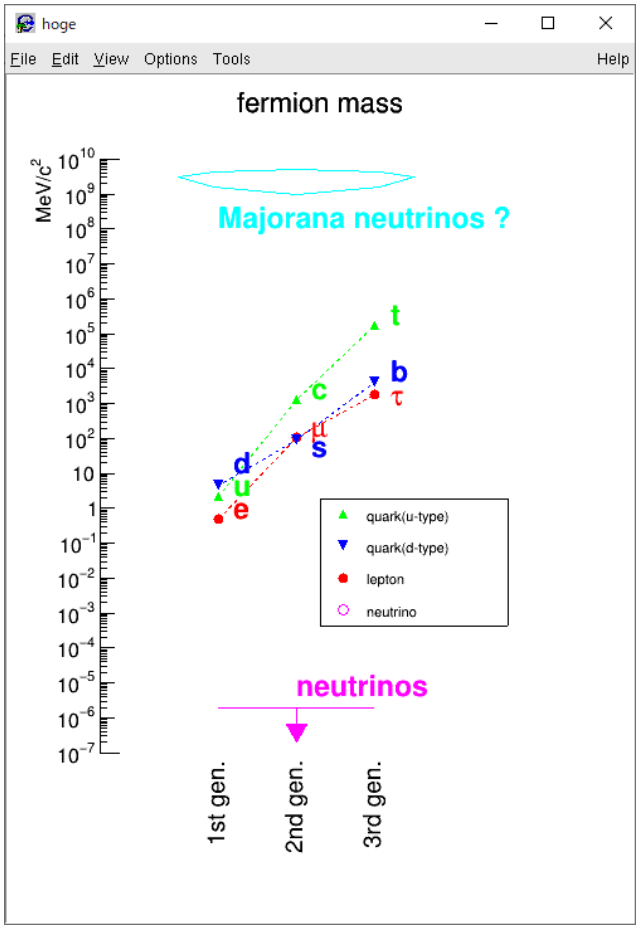
TGraph：装飾

- root graph_rich.C



TGraph：装飾

- root graph_rich.C

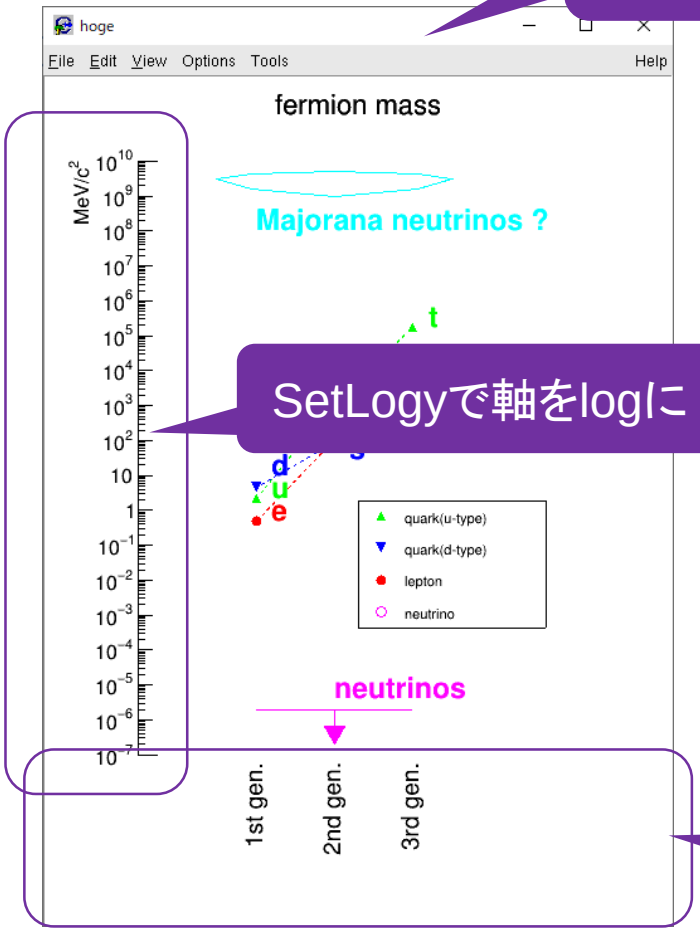



```
{
  gStyle->SetOptStat(0);
  TCanvas *c = new TCanvas("canvas","hoge",500,700);
  c->SetLogy(1);
  c->SetMargin(0.15,0.1,0.2,0.1);//lrbt

  TH1D *waku = new TH1D("waku","fermion mass;;MeV/c^{2}",6,0,6);
  waku->SetMinimum(1e-7);
  waku->SetMaximum(1e10);
  TString label[6]={"","1st gen.","2nd gen.","3rd gen.","",""};
  for(int i=0; i<6; i++) waku->GetXaxis()->SetBinLabel(1+i,label[i]);
  waku->GetXaxis()->SetLabelSize(0.06);
  waku->GetXaxis()->LabelsOption("v");
  waku->GetXaxis()->SetTickLength(0);
  waku->GetXaxis()->SetNdivisions(505);
  waku->Draw();

  TGraph *g[4];
  for(int i=0; i<4; i++) g[i] = new TGraph();
  g[0]->SetPoint(0, 1.5, 0.511);
  g[0]->SetPoint(1, 2.5, 106);
  g[0]->SetPoint(2, 3.5, 1777);
  g[1]->SetPoint(0, 1.5, 2.16);
  g[1]->SetPoint(1, 2.5, 1.27e3);
  g[1]->SetPoint(2, 3.5, 172.76e3);
  g[2]->SetPoint(0, 1.5, 4.67);
  g[2]->SetPoint(1, 2.5, 93);
  g[2]->SetPoint(2, 3.5, 4.18e3);

  int color[4] = {2,3,4,6}, style[4] = {8, 22, 23, 4};
  for(int i=0; i<4; i++){
    g[i]->SetMarkerColor(color[i]);
    g[i]->SetMarkerStyle(style[i]);
    g[i]->SetLineColor(color[i]);
    g[i]->SetLineStyle(2);
    if(i<3) g[i]->Draw("samePL");
  }

  TLegend *leg = new TLegend(0.5,0.35,0.8,0.5);
  leg->AddEntry(g[1],"quark(u-type)","P");
  leg->AddEntry(g[2],"quark(d-type)","P");
  leg->AddEntry(g[0],"lepton","P");
  leg->AddEntry(g[3],"neutrino","P");
  leg->SetFillColor(0);
  leg->Draw();

  TLatex *latex[9];
  latex[0] = new TLatex(1.5+0.2,    0.511,"e");
  latex[1] = new TLatex(2.5+0.2,      106,"#mu");
  latex[2] = new TLatex(3.5+0.2, 0.5*1777,"#tau");
  latex[3] = new TLatex(1.5+0.2,     2.16,"u");
  latex[4] = new TLatex(2.5+0.2,   1.27e3,"s");
  latex[5] = new TLatex(3.5+0.2, 172.76e3,"b");
  latex[6] = new TLatex(1.5+0.2,   2*4.67,"d");
  latex[7] = new TLatex(2.5+0.2,   0.3*93,"c");
  latex[8] = new TLatex(3.5+0.2,   4.18e3,"t");
  for(int i=0; i<9; i++){
    latex[i]->SetTextColor(color[(int)(i/3)]);
    latex[i]->Draw();
  }
 TBox *box[3];
  box[0] = new TBox(0.25,0.90*1e-7, 6.05,1.05*1e-7);
  box[1] = new TBox(5.95,0.90*1e-7, 6.05,1.05*1e10 );
  box[2] = new TBox(0.25,0.95*1e10, 6.05,1.05*1e10 );
  for(int i=0; i<3; i++){
    box[i]->SetFillColor(0);
    box[i]->SetLineColor(0);
    box[i]->Draw();
  }

  TLine *line = new TLine(1.5,2e-6,3.5,2e-6);
  line->SetLineColor(color[3]);
  line->Draw();

  TArrow *arr = new TArrow(2.5,2e-6,2.5,2e-7, 0.03,"|>");
  arr->SetLineColor(color[3]);
  arr->SetFillColor(color[3]);
  arr->Draw();

  TText *text = new TText(2.5,4e-6,"neutrinos");
  text->SetTextColor(color[3]);
  text->Draw();

  TEllipse *el = new TEllipse(2.5,3e9, 1.5,2e9);
  el->SetFillStyle(0);
  el->SetLineColor(7);
  el->Draw();

  TText *text2 = new TText(1.5,1e8,"Majorana neutrinos ?");
  text2->SetTextColor(7);
  text2->Draw();
}
```

TGraph：装飾

- TCanvas

TCanvasでグラフやヒストを描く画面を作る
タイトル名や画面サイズも指定できる



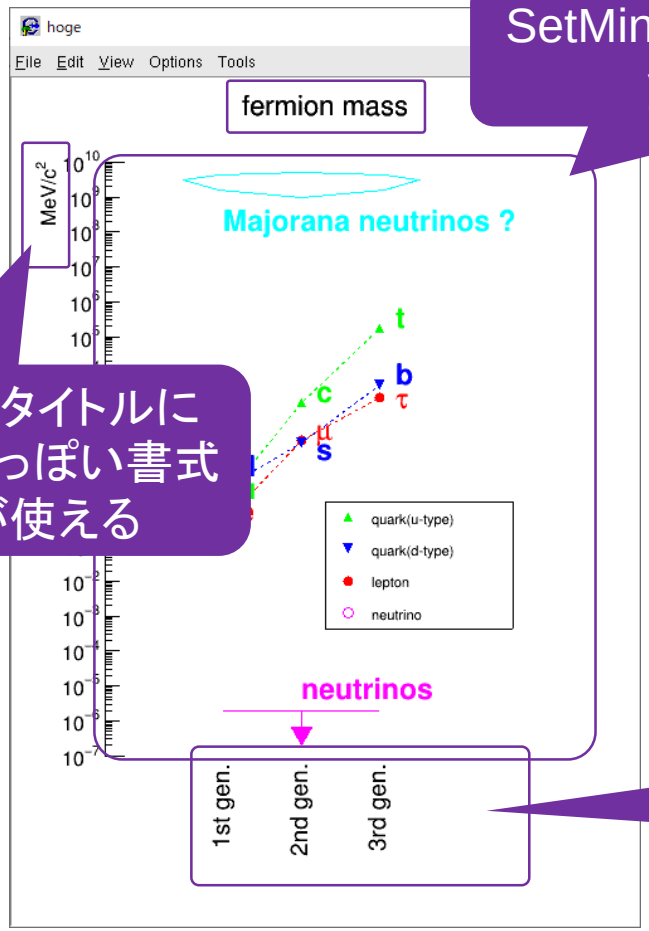

```
TCanvas *c = new TCanvas("canvas","hoge",500,700);
c->SetLogy(1);
c->SetMargin(0.15,0.1,0.2,0.1);//lrbt
```

SetLogyで軸をlogに

SetMarginで左右下上の
余白を設定

TGraph：装飾

- TH1Dで枠を作る



SetMinimumやSetMaximumで
y軸の範囲を指定

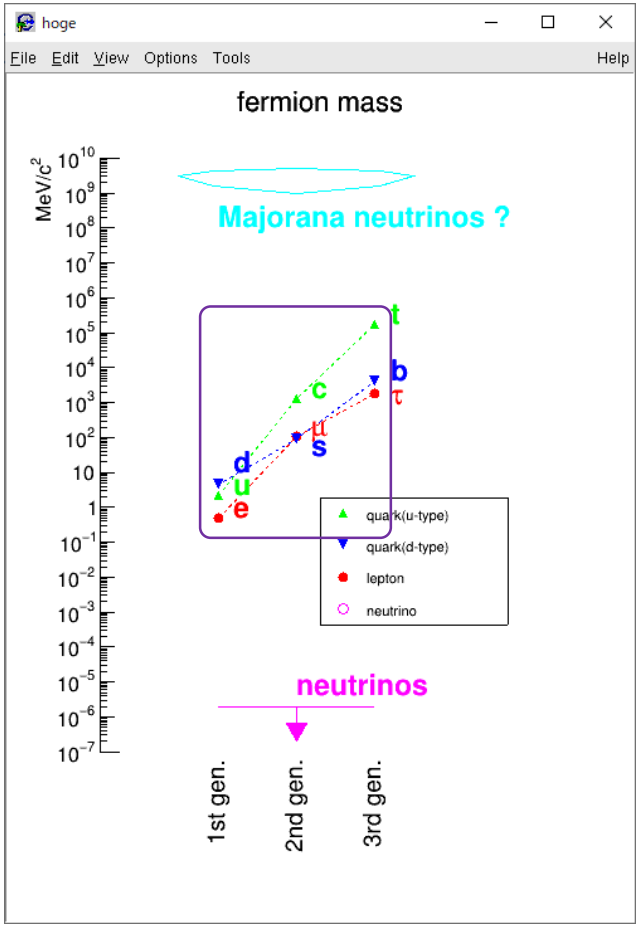
軸のタイトルに
はtexっぽい書式
が使える

```
TH1D *waku = new TH1D("waku","fermion mass;;MeV/c^{2}",6,0,6);
waku->SetMinimum(1e-7);
waku->SetMaximum(1e10);
TString label[6]={"","1st gen.","2nd gen.","3rd gen.","",""};
for(int i=0; i<6; i++) waku->GetXaxis()->SetBinLabel(1+i,label[i]);
waku->GetXaxis()->SetLabelSize(0.06);
waku->GetXaxis()->LabelsOption("v");
waku->GetXaxis()->SetTickLength(0);
waku->GetXaxis()->SetNdivisions(505);
waku->Draw();
```

SetBinLabelでビンに文
字列を割り当て

TGraph：装飾

- TGraphを複数作って装飾



4つ目のグラフは
凡例用のダミー

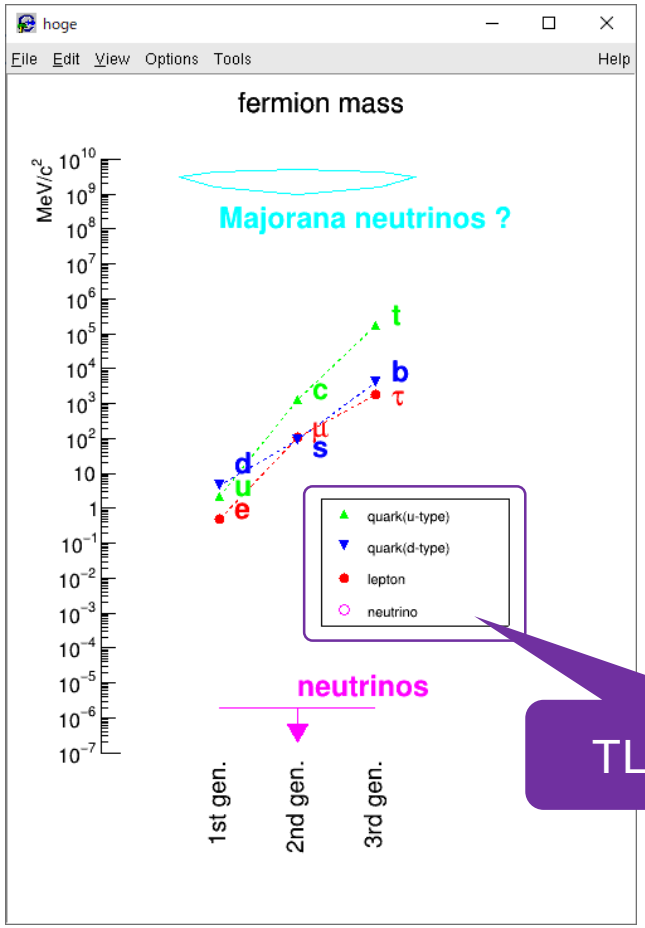
```
TGraph *g[4];
for(int i=0; i<4; i++) g[i] = new TGraph();
g[0]->SetPoint(0, 1.5, 0.511);
g[0]->SetPoint(1, 2.5, 106);
g[0]->SetPoint(2, 3.5, 1777);
g[1]->SetPoint(0, 1.5, 2.16);
g[1]->SetPoint(1, 2.5, 1.27e3);
g[1]->SetPoint(2, 3.5, 172.76e3);
g[2]->SetPoint(0, 1.5, 4.67);
g[2]->SetPoint(1, 2.5, 93);
g[2]->SetPoint(2, 3.5, 4.18e3);

int color[4] = {2,3,4,6}, style[4] = {8, 22, 23, 4};
for(int i=0; i<4; i++){
  g[i]->SetMarkerColor(color[i]);
  g[i]->SetMarkerStyle(style[i]);
  g[i]->SetLineColor(color[i]);
  g[i]->SetLineStyle(2);
  if(i<3) g[i]->Draw("samePL");
}
```

色や形を変える

"samePL" の描画オプションで、
重ね書き、点描画、線描画となる

TGraph：装飾

- TLegendで凡例を作成

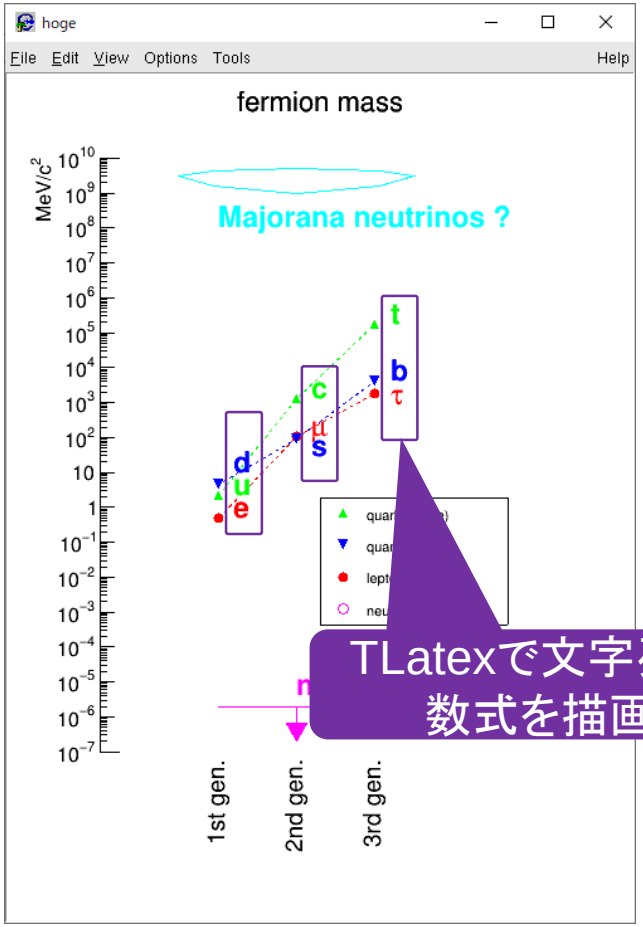



```
TLegend *leg = new TLegend(0.5,0.35,0.8,0.5);
leg->AddEntry(g[1],"quark(u-type)","P");
leg->AddEntry(g[2],"quark(d-type)","P");
leg->AddEntry(g[0],"lepton","P");
leg->AddEntry(g[3],"neutrino","P");
leg->SetFillColor(0);
leg->Draw();
```

TLegendで凡例

TGraph：装飾

- TLatexで文字列や数式を描画



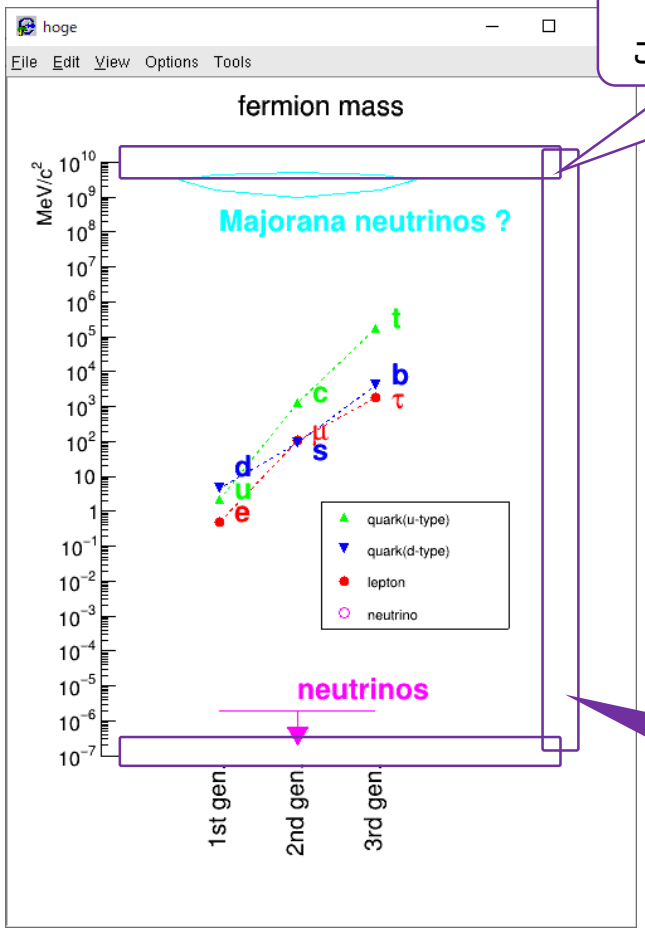
x座標　y座標　テキスト

```
TLatex *latex[9];
latex[0] = new TLatex(1.5+0.2,     0.511,"e");
latex[1] = new TLatex(2.5+0.2,       106,"#mu");
latex[2] = new TLatex(3.5+0.2, 0.5*1777,"#tau");
latex[3] = new TLatex(1.5+0.2,      2.16,"u");
latex[4] = new TLatex(2.5+0.2,    1.27e3,"c");
latex[5] = new TLatex(3.5+0.2, 172.76e3,"t");
latex[6] = new TLatex(1.5+0.2,    2*4.67,"d");
latex[7] = new TLatex(2.5+0.2,    0.3*93,"s");
latex[8] = new TLatex(3.5+0.2,    4.18e3,"b");
for(int i=0; i<9; i++){
  latex[i]->SetTextColor(color[(int)(i/3)]);
  latex[i]->Draw();
}
```

TLatexで文字列や数式を描画

TGraph：装飾

- TBoxで四角形を配置



ごり押しで枠を消す
こういうテクニックも役立つかも

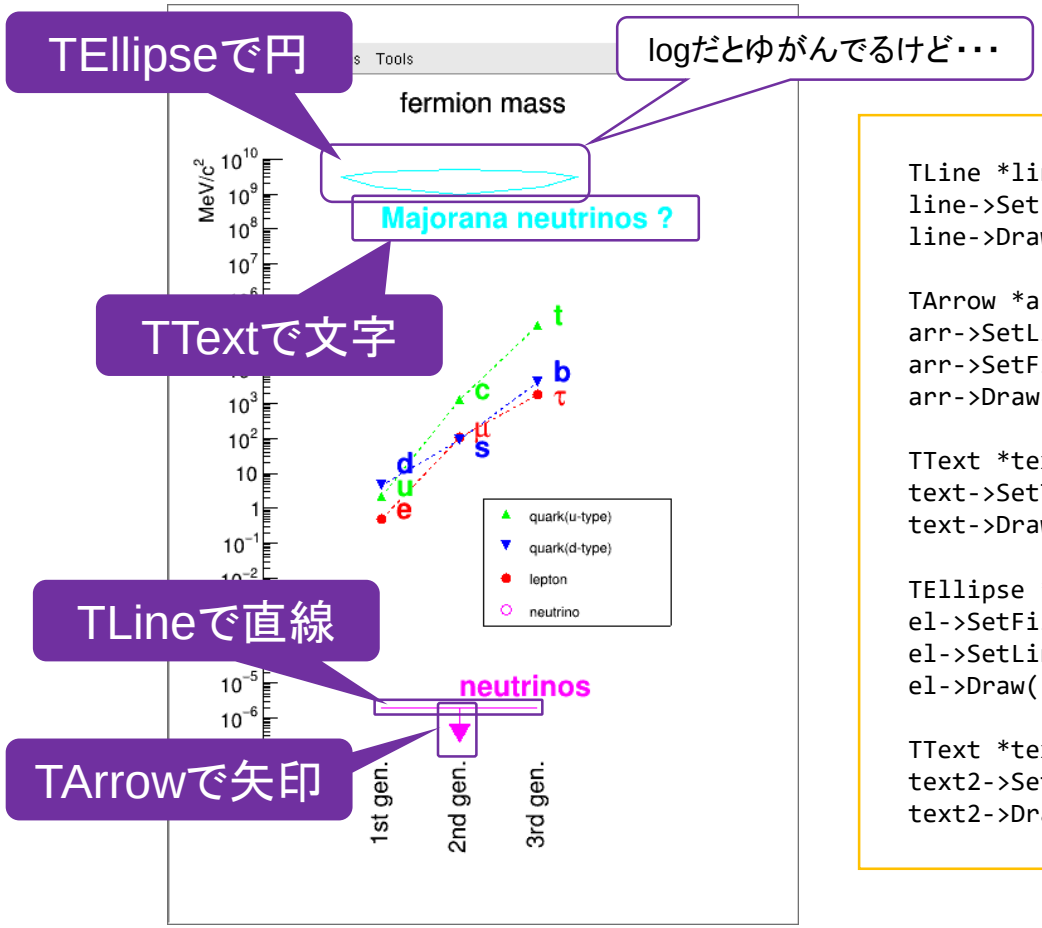
x1 y1 x2 y2

```
TBox *box[3];
box[0] = new TBox(0.25,0.90*1e-7, 6.05,1.05*1e-7);
box[1] = new TBox(5.95,0.90*1e-7, 6.05,1.05*1e10 );
box[2] = new TBox(0.25,0.95*1e10, 6.05,1.05*1e10 );
for(int i=0; i<3; i++){
  box[i]->SetFillColor(0);
  box[i]->SetLineColor(0);
  box[i]->Draw();
}
```

白い四角形を配置

TGraph：装飾

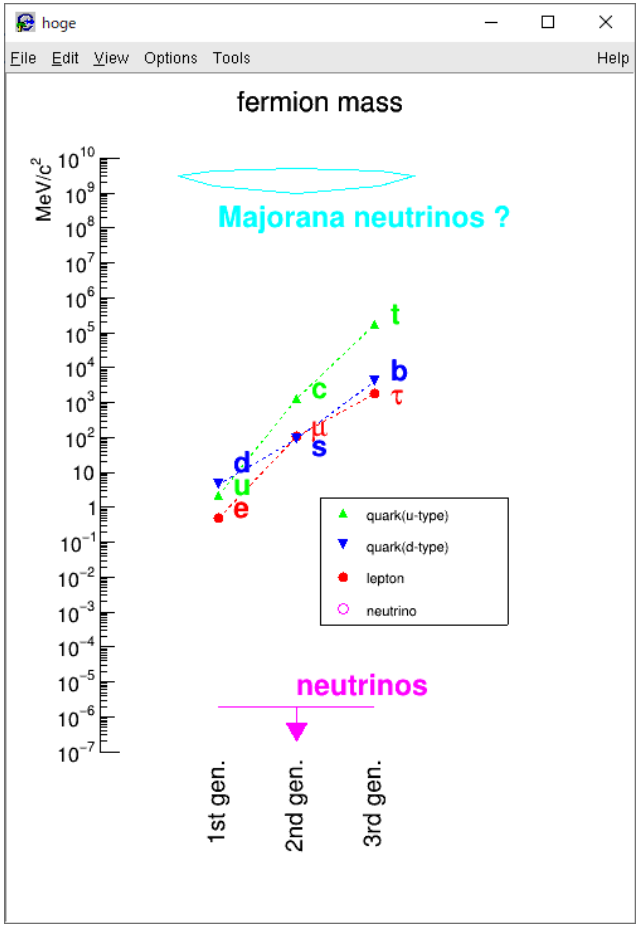
- いろんなものを描画



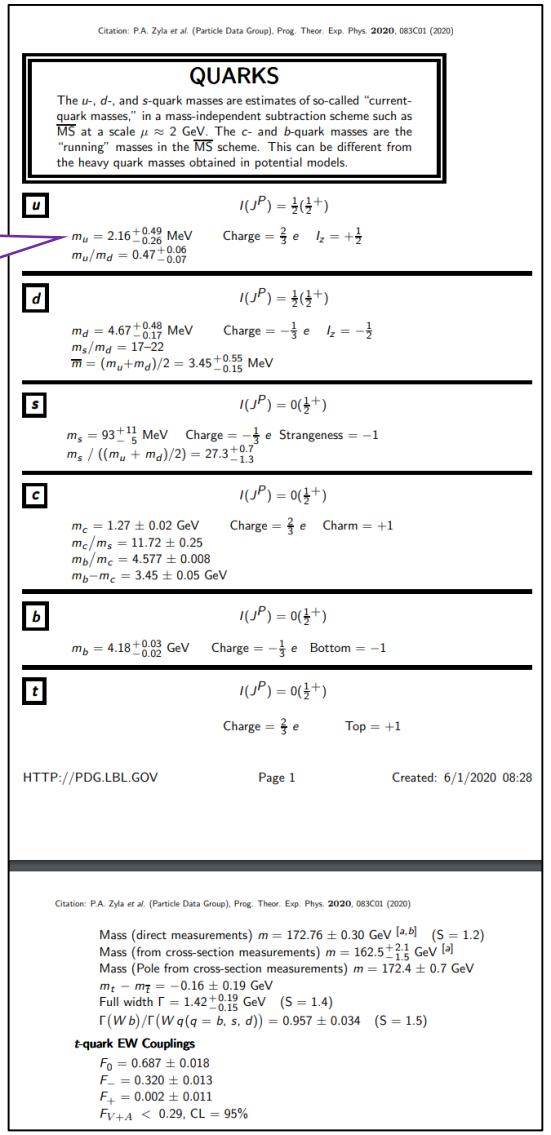

```
TLine *line = new TLine(1.5,2e-6,3.5,2e-6);
line->SetLineColor(color[3]);
line->Draw();

TArrow *arr = new TArrow(2.5,2e-6,2.5,2e-7, 0.03,"|>");
arr->SetLineColor(color[3]);
arr->SetFillColor(color[3]);
arr->Draw();

TText *text = new TText(2.5,4e-6,"neutrinos");
text->SetTextColor(color[3]);
text->Draw();

TEllipse *el = new TEllipse(2.5,3e9, 1.5,2e9);
el->SetFillStyle(0);
el->SetLineColor(7);
el->Draw();

TText *text2 = new TText(1.5,1e8,"Majorana neutrinos ?");
text2->SetTextColor(7);
text2->Draw();
```

TGraph：装飾

- マクロ graph_rich.C を実行してみる



よく見るとエラーバーがある
しかも上下非対称



TGraph：作り方その2

- 配列を定義してから一気に作る方法
- マクロ graph2.C の中身

```
{
  double gen[3]={1.5, 2.5, 3.5};
  double mass[3]={4.67, 93, 4.18e3};
  TGraph *g = new TGraph(3, gen, mass);
  g->Draw();
}
```

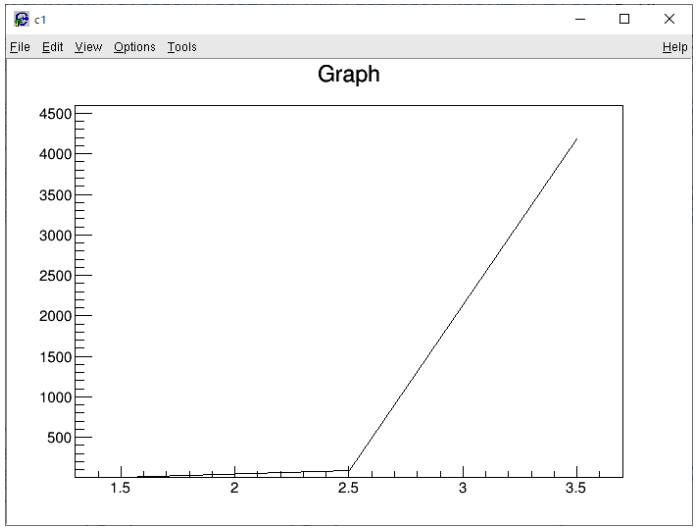
xの配列

yの配列

最初の引数(3)は点の数で定義

描画する

こっちの方がその1より
行数は短い



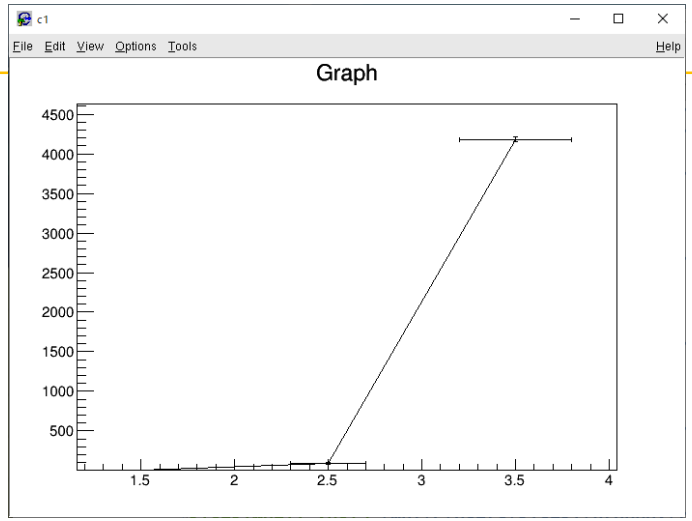
TGraphErrors

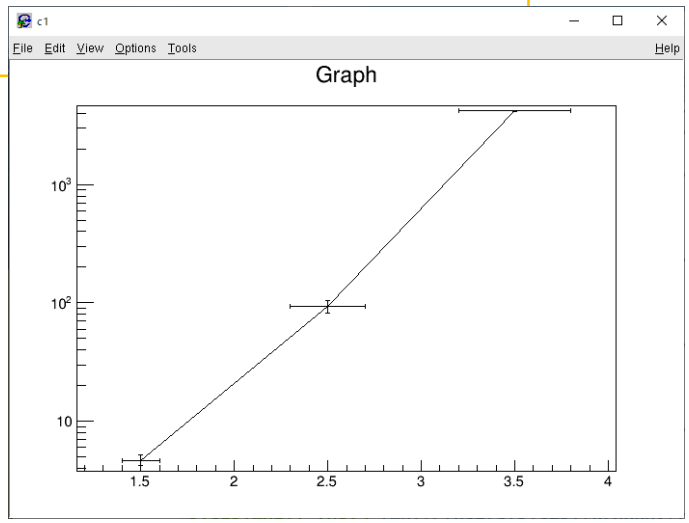
- 上下対称な誤差を付ける
- マクロ grapherrors.C

```
{
  double gen[3] = {1.5, 2.5, 3.5};
  double mass[3] = {4.67, 93, 4.18e3};
  double ex[3] = {0.1, 0.2, 0.3};
  double ey[3] = {0.48, 11, 0.03e3};
  TGraphErrors *g = new TGraphErrors(3, gen, mass, ex, ey);
  g->Draw();
}
```

xの誤差の配列

yの誤差の配列



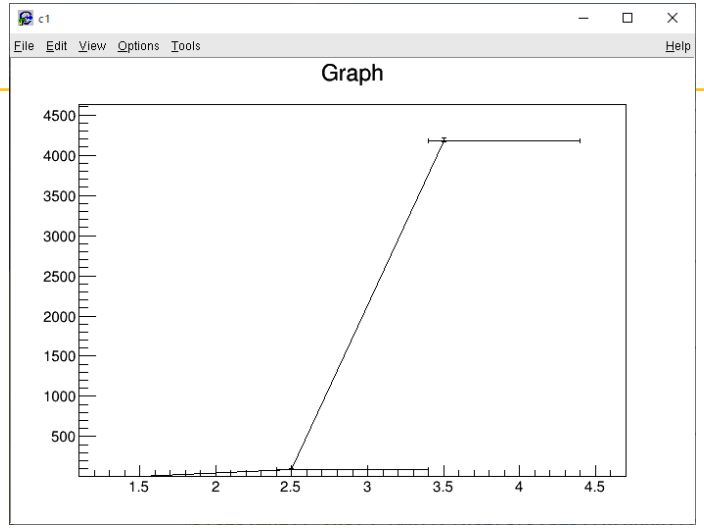


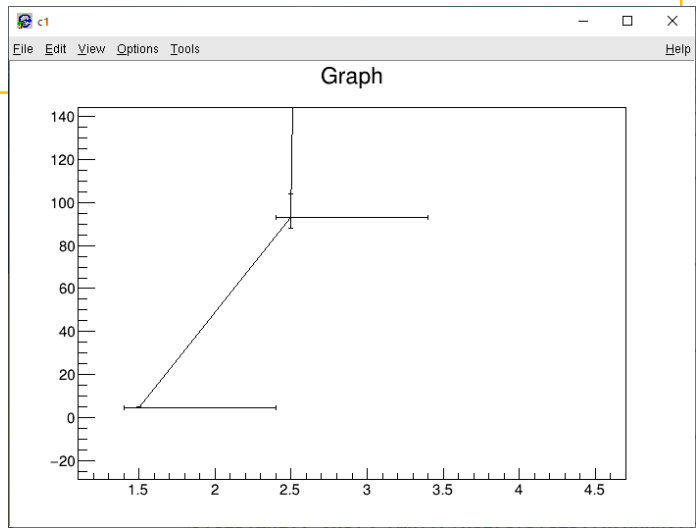
TGraphAssymErrors

- 上下非対称な誤差を付ける
- マクロ graphassymerrors.C

```
{
  double gen[3] = {1.5, 2.5, 3.5};
  double mass[3] = {4.67, 93, 4.18e3};
  double ex1[3] = {0.1, 0.1, 0.1};
  double ex2[3] = {0.9, 0.9, 0.9};
  double ey1[3] = {0.17,  5, 0.02e3};
  double ey2[3] = {0.48, 11, 0.03e3};
  TGraphAsymmErrors *g = new TGraphAsymmErrors(3, gen, mass, ex1, ex2, ey1, ey2);
  g->Draw();
}
```

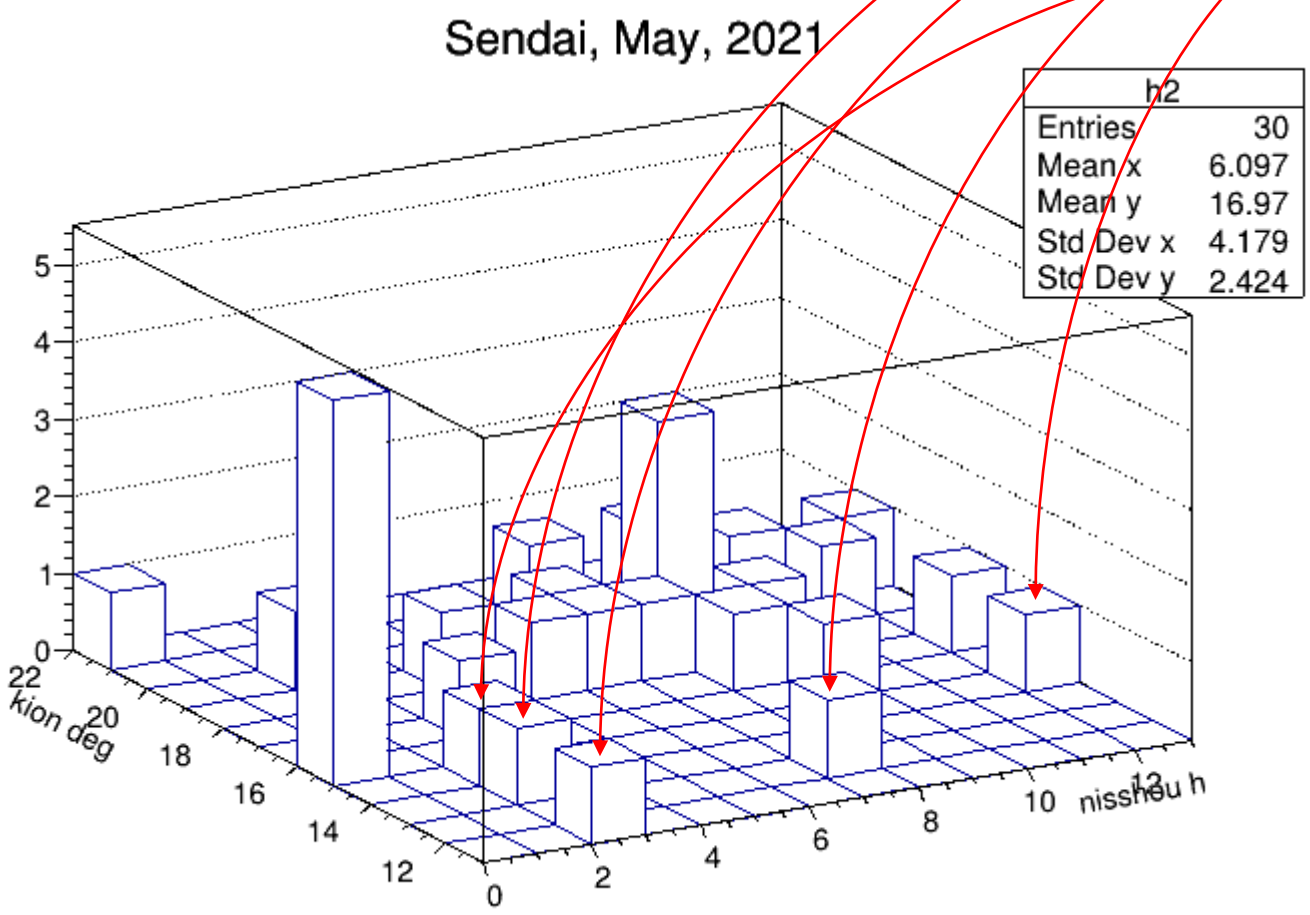
left right bottom top





TH2D とは

- 2個の変数の分布を描画する



気温(℃)	日照時間(h)
13.1	2.1
11.3	2.7
12.8	7.5
16.2	12.3
14.5	2.7
18.3	11.6
16.0	3.1
19.6	10.3
18.4	8.5
16.6	8.7
15.3	9.1
14.0	12.8
17.0	5.6
18.3	6.7
17.0	9.9
19.4	2.5
21.5	0.1
18.1	4.9
15.5	0.0
18.5	8.8
15.7	0.0
15.8	0.0
15.4	0.3
17.7	6.9
20.8	9.3
17.8	7.5
15.4	0.0
19.8	12.8
20.6	7.8
18.6	8.4

TH2D：つくりかた

- th2.C マクロ

```
{
  TH2D *h2 = new TH2D("h2","Sendai, May, 2021;nisshou h;kion deg", 13,0,13, 11,11,22);
  ifstream ifs("tenki.dat");
  TString dummy;
  ifs >> dummy >> dummy;
  double x, y;
  while(ifs >> y >> x) h2->Fill(x, y);
  h2->Draw("COLZ");
  //h2->Draw("LEGO");
}
```

x軸の bin,min,max

y軸の bin,min,max

値を2つ入れる

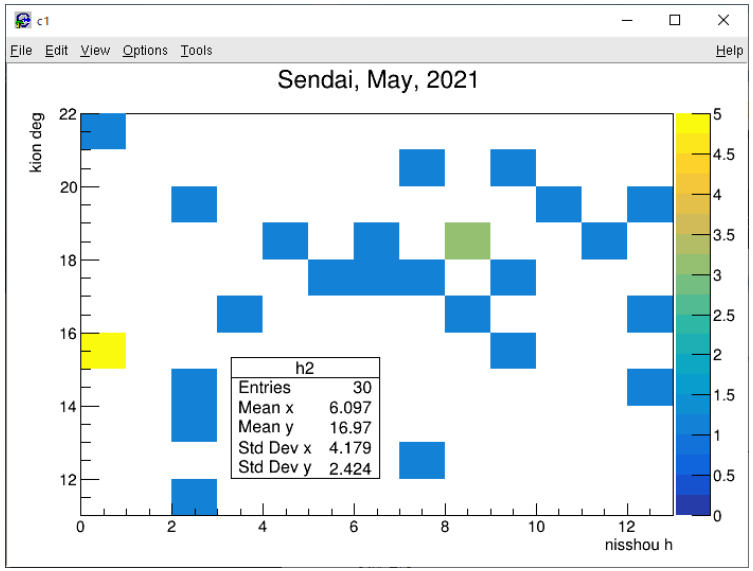
描画する

COLで色付き
Zでカラーバーを表示
LEGOはレゴ

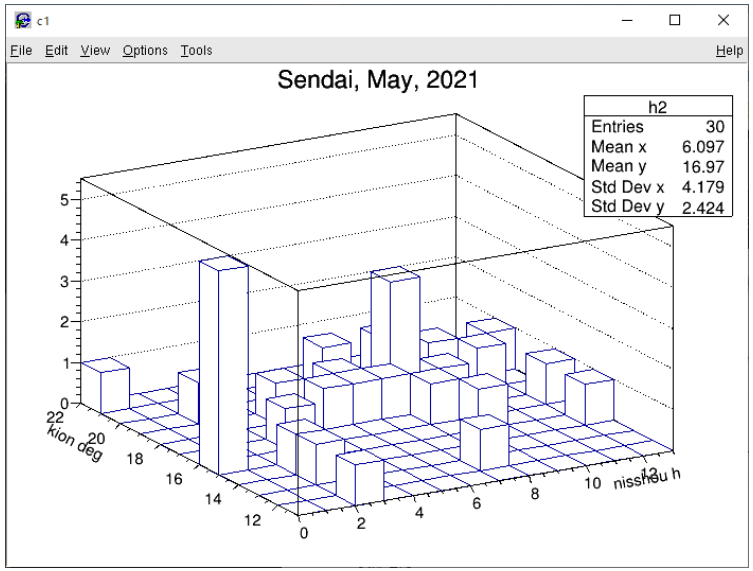
TH2D：つくりかた

- th2.C マクロ

COLZ



LEGO



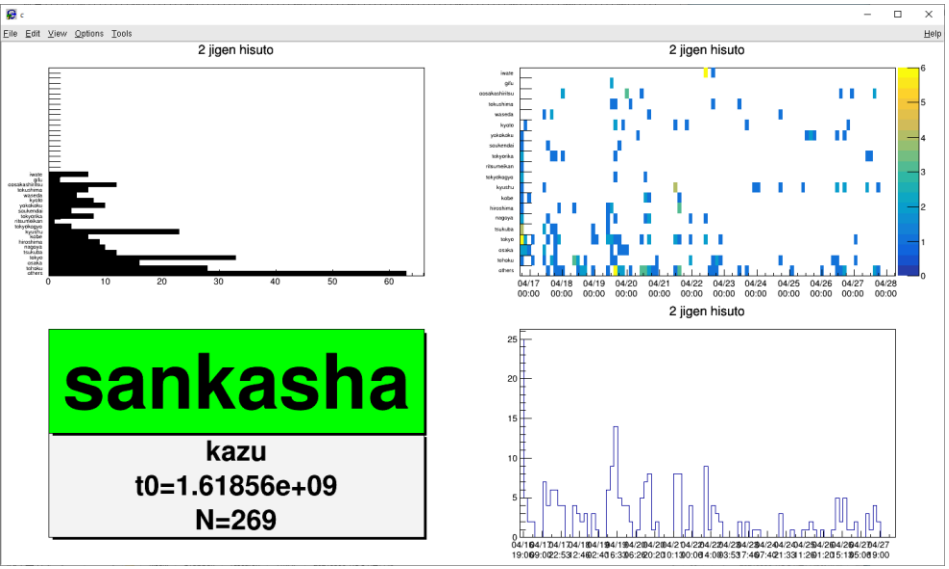
TH2D : 小技

- th2_rich.C マクロ

```
{
  gStyle->SetOptStat(0);

  TCanvas *c = new TCanvas("c","c",1400,800);
  c->Divide(2,2,0.001,0.001);

  double t0 = 1618559095;
  TH2D *h2 = new TH2D("hist","2 jigen hisuto", 100,t0,t0+1000000, 20,0,20);
  ifstream ifs("data.dat");
  double dummy, t, shozoku_id;
  while(ifs >> dummy >> t >> dummy >> shozoku_id){
    for(int i=0; i<8; i++) ifs >> dummy;
    h2->Fill(t+t0, shozoku_id);
  }
  TString label[20] = {"others","tohoku","osaka","tokyo","tsukuba","nagoya",
                       "hiroshima","kobe","kyushu","tokyokogyo","ritsumeikan",
                       "tokyorika","soukendai","yokokoku","kyoto","waseda",
                       "tokushima","oosakashiritsu","gifu","iwate"};
  for(int i=0; i<20; i++) h2->GetYaxis()->SetBinLabel(1+i, label[i]);
  h2->GetXaxis()->SetTimeDisplay(1);
  h2->GetXaxis()->SetTimeFormat("#splitline{%m/%d}{%H:%M}");
  h2->GetXaxis()->SetTimeOffset(32400,"gmt");
  h2->GetXaxis()->SetNdivisions(420);
  h2->GetXaxis()->SetLabelOffset(0.03);
  c->cd(2);
  h2->Draw("COLZ");

  c->cd(1);
  h2->ProjectionY()->Draw("hbar");

  c->cd(4);
  TH1D *h1 = (TH1D*)(h2->ProjectionX());
  h1->GetXaxis()->SetLabelOffset(0.03);
  h1->Draw();

  c->cd(3);
  TPaveText *pt = new TPaveText(0.1,0.1,0.9,0.5);
  pt->AddText("kazu");
  pt->AddText(Form("t0=%g",t0));
  pt->AddText(Form("N=%g",h2->GetEntries()));
  pt->Draw();
  TPaveText *pt2 = new TPaveText(0.1,0.5,0.9,0.9);
  pt2->AddText("sankasha");
  pt2->SetFillColor(3);
  pt2->Draw();
}
```

TH2D：小技

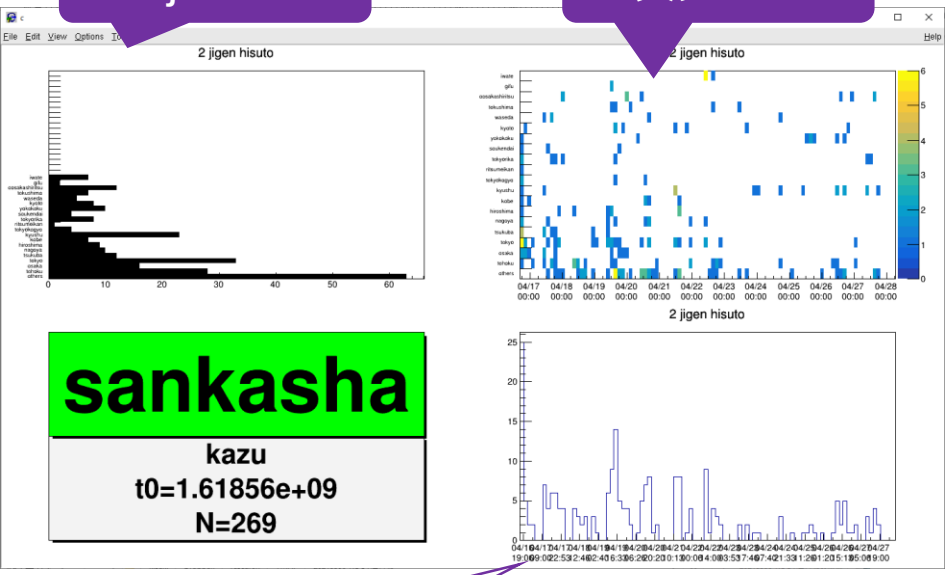
- th2_rich.C マクロ
- 実行してみる




```
{
  gStyle->SetOptStat(0);

  TCanvas *c = new TCanvas("c","c",1400,800);
  c->Divide(2,2,0.001,0.001);

  double t0 = 1618559095;
  TH2D *h2 = new TH2D("hist","2 jigen hisuto", 100,t0,t0+1000000, 20,0,20);
  ifstream ifs("data.dat");
  double dummy, t, shozoku_id;
  while(ifs >> dummy >> t >> dummy >> shozoku_id){
    for(int i=0; i<8; i++) ifs >> dummy;
    h2->Fill(t+t0, shozoku_id);
  }
  TString label[20] = {"others","tohoku","osaka","tokyo","tsukuba","nagoya",
                       "hiroshima","kobe","kyushu","tokyokogyo","ritsumeikan",
                       "tokyorika","soukendai","yokokoku","kyoto","waseda",
                       "tokushima","oosakashiritsu","gifu","iwate"};
  for(int i=0; i<20; i++) h2->GetYaxis()->SetBinLabel(1+i, label[i]);
  h2->GetXaxis()->SetTimeDisplay(1);
  h2->GetXaxis()->SetTimeFormat("#splitline{%m/%d}{%H:%M}");
  h2->GetXaxis()->SetTimeOffset(32400,"gmt");
  h2->GetXaxis()->SetNdivisions(420);
  h2->GetXaxis()->SetLabelOffset(0.03);
  c->cd(2);
  h2->Draw("COLZ");

  c->cd(1);
  h2->ProjectionY()->Draw("hbar");

  c->cd(4);
  TH1D *h1 = (TH1D*)(h2->ProjectionX());
  h1->GetXaxis()->SetLabelOffset(0.03);
  h1->Draw();

  c->cd(3);
  TPaveText *pt = new TPaveText(0.1,0.1,0.9,0.5);
  pt->AddText("kazu");
  pt->AddText(Form("t0=%g",t0));
  pt->AddText(Form("N=%g",h2->GetEntries()));
  pt->Draw();
  TPaveText *pt2 = new TPaveText(0.1,0.5,0.9,0.9);
  pt2->AddText("sankasha");
  pt2->SetFillColor(3);
  pt2->Draw();
}
```

TH2D：小技

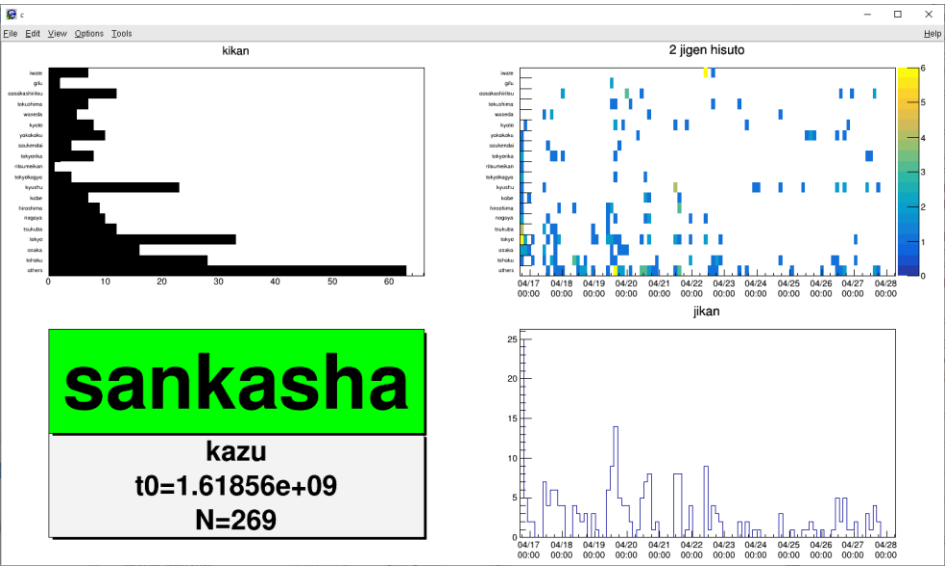
- th2_rich.C マクロ
- 実行してみる




```
{
  gStyle->SetOptStat(0);

  TCanvas *c = new TCanvas("c","c",1400,800);
  c->Divide(2,2,0.001,0.001);

  double t0 = 1618559095;
  TH2D *h2 = new TH2D("hist","2 jigen hisuto", 100,t0,t0+1000000, 20,0,20);
  ifstream ifs("data.dat");
  double dummy, t, shozoku_id;
  while(ifs >> dummy >> t >> dummy >> shozoku_id){
    for(int i=0; i<8; i++) ifs >> dummy;
    h2->Fill(t+t0, shozoku_id);
  }
  TString label[20] = {"others","tohoku","osaka","tokyo","tsukuba","nagoya",
                       "hiroshima","kobe","kyushu","tokyokogyo","ritsumeikan",
                       "tokyorika","soukendai","yokokoku","kyoto","waseda",
                       "tokushima","oosakashiritsu","gifu","iwate"};
  for(int i=0; i<20; i++) h2->GetYaxis()->SetBinLabel(1+i, label[i]);
  h2->GetXaxis()->SetTimeDisplay(1);
  h2->GetXaxis()->SetTimeFormat("#splitline{%m/%d}{%H:%M}");
  h2->GetXaxis()->SetTimeOffset(32400,"gmt");
  h2->GetXaxis()->SetNdivisions(420);
  h2->GetXaxis()->SetLabelOffset(0.03);
  c->cd(2);
  h2->Draw("COLZ");

  c->cd(1);
  h2->ProjectionY()->Draw("hbar");

  c->cd(4);
  TH1D *h1 = (TH1D*)(h2->ProjectionX());
  h1->GetXaxis()->SetLabelOffset(0.03);
  h1->Draw();

  c->cd(3);
  TPaveText *pt = new TPaveText(0.1,0.1,0.9,0.5);
  pt->AddText("kazu");
  pt->AddText(Form("t0=%g",t0));
  pt->AddText(Form("N=%g",h2->GetEntries()));
  pt->Draw();
  TPaveText *pt2 = new TPaveText(0.1,0.5,0.9,0.9);
  pt2->AddText("sankasha");
  pt2->SetFillColor(3);
  pt2->Draw();
}
```

TH2D：小技

- th2_rich2.C マクロ
- 実行してみる




```
{
  gStyle->SetOptStat(0);

  TCanvas *c = new TCanvas("c","c",1400,800);
  c->Divide(2,2,0.001,0.001);

  double t0 = 1618559095;
  TH2D *h2 = new TH2D("hist","2 jigen hisuto", 100,t0,t0+1000000, 20,0,20);
  TH1D *h1_kikan = new TH1D("kikan","kikan", 20,0,20);
  TH1D *h1_jikan = new TH1D("jikan","jikan", 100,t0,t0+1000000);
  ifstream ifs("data.dat");
  double dummy, t, shozoku_id;
  while(ifs >> dummy >> t >> dummy >> shozoku_id){
    for(int i=0; i<8; i++) ifs >> dummy;
    h2->Fill(t+t0, shozoku_id);
    h1_kikan->Fill(shozoku_id);
    h1_jikan->Fill(t+t0);
  }
  TString label[20] = {"others","tohoku","osaka","tokyo","tsukuba","nagoya",
                       "hiroshima","kobe","kyushu","tokyokogyo","ritsumeikan",
                       "tokyorika","soukendai","yokokoku","kyoto","waseda",
                       "tokushima","oosakashiritsu","gifu","iwate"};
  for(int i=0; i<20; i++) h2->GetYaxis()->SetBinLabel(1+i, label[i]);
  h2->GetXaxis()->SetTimeDisplay(1);
  h2->GetXaxis()->SetTimeFormat("#splitline{%m/%d}{%H:%M}");
  h2->GetXaxis()->SetTimeOffset(32400,"gmt");
  h2->GetXaxis()->SetNdivisions(420);
  h2->GetXaxis()->SetLabelOffset(0.03);
  for(int i=0; i<20; i++) h1_kikan->GetXaxis()->SetBinLabel(1+i, label[i]);
  h1_jikan->GetXaxis()->SetTimeDisplay(1);
  h1_jikan->GetXaxis()->SetTimeFormat("#splitline{%m/%d}{%H:%M}");
  h1_jikan->GetXaxis()->SetTimeOffset(32400,"gmt");
  h1_jikan->GetXaxis()->SetNdivisions(420);
  h1_jikan->GetXaxis()->SetLabelOffset(0.03);
  c->cd(2);
  h2->Draw("COLZ");

  c->cd(1);
  h1_kikan->Draw("hbar");

  c->cd(4);
  h1_jikan->Draw();

  c->cd(3);
  TPaveText *pt = new TPaveText(0.1,0.1,0.9,0.5);
  pt->AddText("kazu");
  pt->AddText(Form("t0=%g",t0));
  pt->AddText(Form("N=%g",h2->GetEntries()));
  pt->Draw();
  TPaveText *pt2 = new TPaveText(0.1,0.5,0.9,0.9);
  pt2->AddText("sankasha");
  pt2->SetFillColor(3);
  pt2->Draw();
}
```

TH2D [illegible]

- th2_rich2.C マクロ
- 実行してみる

キャンバスを4つのpadに分割する

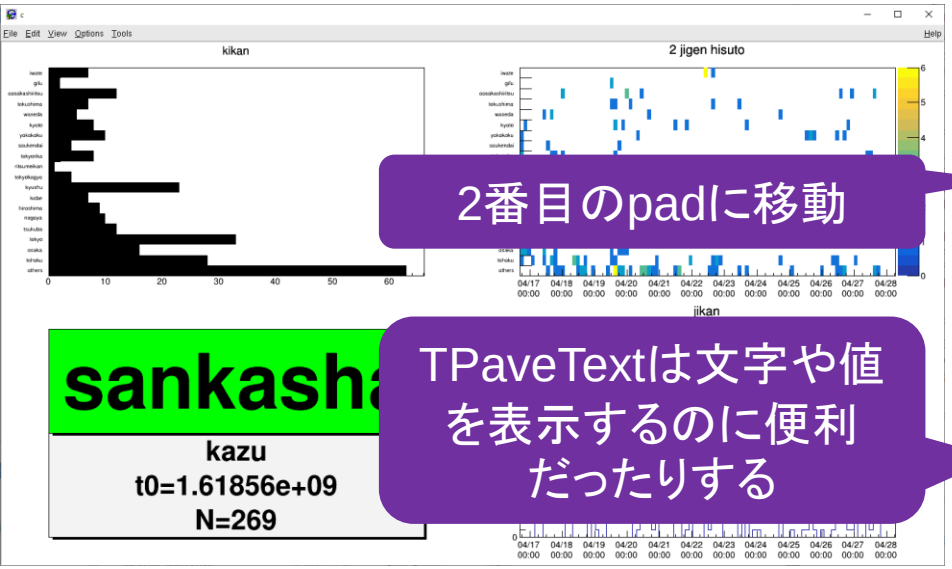
ヒストの定義
時間はユニックスタイム

データを読んで詰める

時間軸の表示設定

2番目のpadに移動

TPaveTextは文字や値を表示するのに便利だったりする




```
{
  gStyle->SetOptStat(0);

  TCanvas *c = new TCanvas("c","c",1400,800);
  c->Divide(2,2,0.001,0.001);

  double t0 = 1618559095;
  TH2D *h2 = new TH2D("hist","2 jigen hisuto", 100,t0,t0+1000000, 20,0,20);
  TH1D *h1_kikan = new TH1D("kikan","kikan", 20,0,20);
  TH1D *h1_jikan = new TH1D("jikan","jikan", 100,t0,t0+1000000);
  ifstream ifs("data.dat");
  double dummy, t, shozoku_id;
  while(ifs >> dummy >> t >> dummy >> shozoku_id){
    for(int i=0; i<8; i++) ifs >> dummy;
    h2->Fill(t+t0, shozoku_id);
    h1_kikan->Fill(shozoku_id);
    h1_jikan->Fill(t+t0);
  }
  TString label[20] = {"others","tohoku","osaka","tokyo","tsukuba","nagoya",
                       "hiroshima","kobe","kyushu","tokyokogyo","ritsumeikan",
                       "tokyorika","soukendai","yokokoku","kyoto","waseda",
                       "tokushima","oosakashiritsu","gifu","iwate"};
  for(int i=0; i<20; i++) h2->GetYaxis()->SetBinLabel(1+i, label[i]);
  h2->GetXaxis()->SetTimeDisplay(1);
  h2->GetXaxis()->SetTimeFormat("#splitline{%m/%d}{%H:%M}");
  h2->GetXaxis()->SetTimeOffset(32400,"gmt");
  h2->GetXaxis()->SetNdivisions(420);
  h2->GetXaxis()->SetLabelOffset(0.03);
  for(int i=0; i<20; i++) h1_kikan->GetXaxis()->SetBinLabel(1+i, label[i]);
  h1_jikan->GetXaxis()->SetTimeDisplay(1);
  h1_jikan->GetXaxis()->SetTimeFormat("#splitline{%m/%d}{%H:%M}");
  h1_jikan->GetXaxis()->SetTimeOffset(32400,"gmt");
  h1_jikan->GetXaxis()->SetNdivisions(420);
  h1_jikan->GetXaxis()->SetLabelOffset(0.03);
  c->cd(2);
  h2->Draw("COLZ");

  c->cd(1);
  h1_kikan->Draw("hbar");

  c->cd(4);
  h1_jikan->Draw();

  c->cd(3);
  TPaveText *pt = new TPaveText(0.1,0.1,0.9,0.5);
  pt->AddText("kazu");
  pt->AddText(Form("t0=%g",t0));
  pt->AddText(Form("N=%g",h2->GetEntries()));
  pt->Draw();
  TPaveText *pt2 = new TPaveText(0.1,0.5,0.9,0.9);
  pt2->AddText("sankasha");
  pt2->SetFillColor(3);
  pt2->Draw();
}
```

TH3D

- th3.C
- 三次元になっただけ
- （TH4は存在しない）

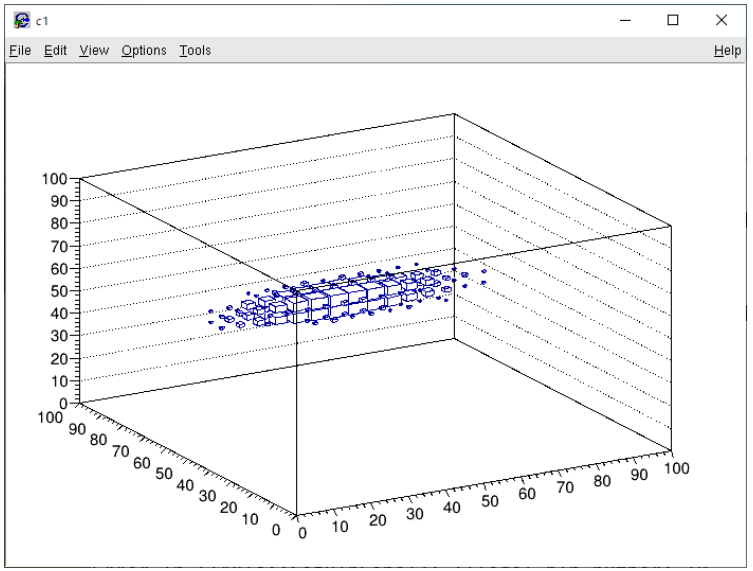
```
{
  TH3D *h = new TH3D("","", 20,0,100, 20,0,100, 20,0,100);
  for(int i=0; i<10000; i++) h->Fill(gRandom->Gaus(40,10), gRandom->Gaus(50,2), gRandom->Gaus(60,2));
  h->Draw("box");
}
```

TH3D

```
{
  TH3D *h = new TH3D("","", 20,0,100, 20,0,100, 20,0,100);
  for(int i=0; i<10000; i++) h->Fill(gRandom->Gaus(40,10), gRandom->Gaus(50,2), gRandom->Gaus(60,2));
  h->Draw("box");
}
```

- th3.C
- 三次元になっただけ
- （TH4は存在しない）

3次元空間の各bin（ボクセル）に、
値に比例した箱が描いてある感じ



TGraph2D

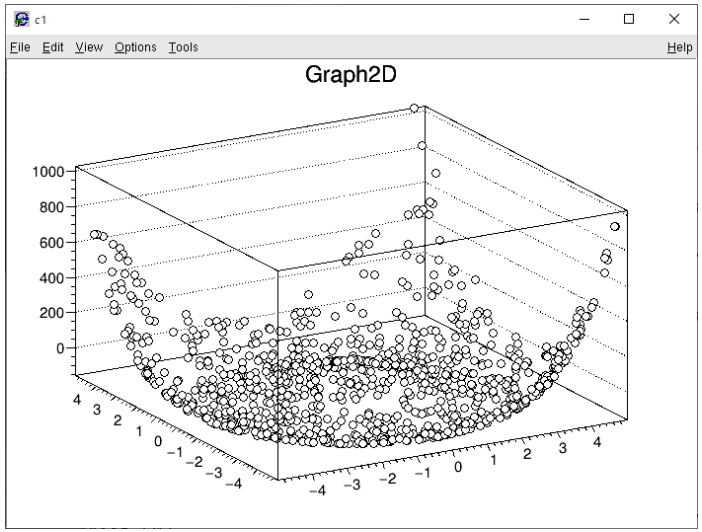
- graph2d.C
- xyz空間に点を打っていく

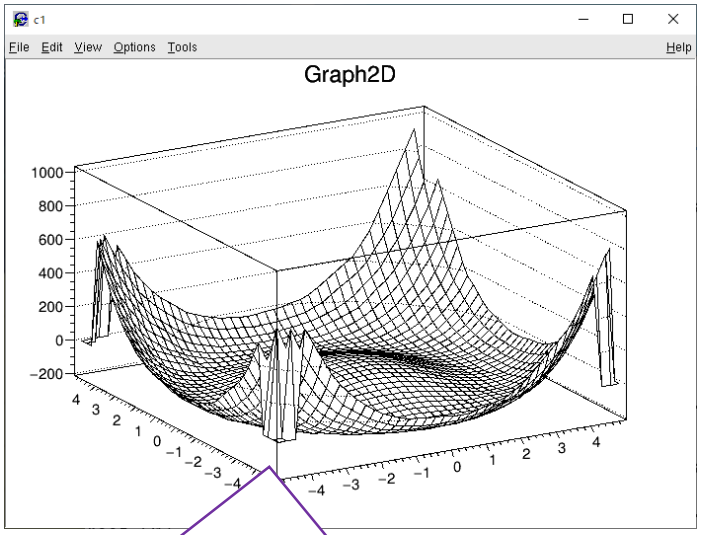
```
{
  TGraph2D *g = new TGraph2D();
  for(int i=0; i<1000; i++){
    double x = gRandom->Uniform(-5,5);
    double y = gRandom->Uniform(-5,5);
    double z = (x*x+y*y)*(x*x+y*y-25.0);
    g->SetPoint(i,x,y,z);
  }
  g->Draw();
  // g->Draw("surf");
}
```

TGraph2D

- graph2d.C
- xyz空間に点を打っていく

```
{
  TGraph2D *g = new TGraph2D();
  for(int i=0; i<1000; i++){
    double x = gRandom->Uniform(-5,5);
    double y = gRandom->Uniform(-5,5);
    double z = (x*x+y*y)*(x*x+y*y-25.0);
    g->SetPoint(i,x,y,z);
  }
  g->Draw();
  // g->Draw("surf");
}
```





x,yはランダムに振ってるので、
点がないとz=0あつかいで面をつなごうとする

おわりに

Day 2 宿題

- あなた自身で設定したテーマでデータを収集し，ROOTを使ってヒストグラムを描いて考察してください．
- テーマは物理に限りません．
- 形式はなんでもいい（レポート，スライド，その他）ので，slackの#homework2にアップロードしてください．
- 最低1枚はROOTで書いた図，ソースコード を載せてください．

- Day 3 はFitting，検定，グラフなので，集めたデータを実際に解析することもできるようになるとおもいます．

- お疲れさまでした！
- せっかくなので、水越宿題で作る図を、いい感じに装飾してみましょう

- 次回は6/10(木) 17:00-19:00
  - なんでも質問コーナー
  - 講師の人のマクロ紹介（時間あったら）

- YMAPへの参加もぜひ
  - http://www.icrr.u-tokyo.ac.jp/YMAP/join.html