

# ROOT講習会 第4回

## TTree & TH2

高橋光成 (名大ISEE)

1st/June/2022

# 自己紹介

- ・名前：高橋光成
- ・所属：名古屋大学宇宙地球環境研究所宇宙線研究室
- ・プロジェクト：CTA, Fermi-LAT, MAGIC
- ・研究：光検出器（PMT・SiPM）、ガンマ線バースト、活動銀河核



# 第4回でやること

- 「TTree」 を使えるようになる
  - データからTTreeを作る
  - TTreeからヒストグラム、グラフを描く
- 2次元ヒストグラム「TH2」 を使えるようになる
  - データからTH2を作る
  - TH2を見やすく表示できるようになる

TTree

# TTreeとは？

- ・ 「イベント」データの処理・保存に特化したクラス
  - ・ 膨大な量のデータを扱うのに適している
  - ・ 様々な型が保存でき使い所が多い
- ・ 表（テーブル）に似ている
  - ・ 列→ブランチ、行→エントリと呼ばれる
  - ・ 便利で使いやすい機能が色々ある

# とりあえず作ってみよう

- 例としてガンマ線望遠鏡Fermi-LATのデータを用いる
  - root\_lecture/macros/tkhs/lat-data/lat\_photons.dat
- 中身を見てみると...



```
(base) mitsunari@White-CTA-iMac lat-data % head allsky_photons.dat
# ENERGY      L      B      ZENITH_ANGLE  EARTH_AZIMUTH_ANGLE  TIME
 4931.1074  186.40201  66.69105   110.99175    30.568527  4.5610560374749446E8
 2988.297   57.856243  52.6086    111.2355    331.76807   4.5610560458237594E8
 516.18414   82.95196  67.50891   111.29916    352.37335   4.5610560459165955E8
 442.3429   205.57336  63.7599    115.185455   38.6572    4.561056051064911E8
 1036.0813   42.96566  39.94485   113.50167    314.379    4.561056051262641E8
 778.4724   53.01713  48.3351    111.54488   326.1625    4.5610560537444526E8
 367.00674   184.2203  69.56075   112.504745   27.951687   4.561056057890055E8
 404.6052   45.00639  39.51818   111.96931   314.87454   4.561056061210675E8
 77387.74   168.77509  2.4389622  50.717117    58.118896  4.56105606131219E8
```

エネルギー

到来方向の座標

到来時刻

1行が1個の光子に対応

# TTreeを作り、テキストファイルからデータを読み込む

- root\_lecture/macros/tkhs/examples/plant\_tree.C

```
File Edit Options Buffers Tools C Help
// plant_tree.C

void plant_tree(){
    // Open file
    TFile *file = new TFile("allsky_photons.root", "RECREATE");
    file->cd();

    // Create TTree
    TTree *tree = new TTree("photontree", "Photon tree");      名前      タイトル
                                                                ブランチの名前と型を指定、
                                                                ブランチは「:」で区切る
    // Read text file
    tree->ReadFile("../lat-data/allsky_photons.dat", "ENERGY/D:L:B:ZENITH_ANGLE:EARTH_AZIMUTH_ANGLE:TIME", ':');   読み込むファイル
                                                                型は「/」の後に指定。前のブランチと同じ
                                                                型なら省略できる

    // Save TTree
    tree->Write();
    file->Close();
}
```

読み込むファイルの区切り文字を指定

読み込むファイル  
ブランチは「:」で区切る

型は「/」の後に指定。前のブランチと同じ  
型なら省略できる

- root -l plant\_tree.Cで走らせてみよう

# 作ったTTreeを見てみよう

```
(base) mitsunari@White-CTA-iMac examples % root allsky_photons.root
-----| Welcome to ROOT 6.26/00          https://root.cern |
| (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers |
| Built for macosx64 on Mar 03 2022, 06:51:13 |
| From tags/v6-26-00@v6-26-00 |
| With Apple clang version 13.1.6 (clang-1316.0.21.2) |
| Try '.help', '.demo', '.license', '.credits', '.quit'/.q' |
-----
root [0]
Attaching file allsky_photons.root as _file0...
(TFile *) 0x7f7ee7fab0c80
root [1] .ls
TFile**      allsky_photons.root
TFile*       allsky_photons.root
  KEY: TTree photontree;1 Photon tree ScanでTTreeの中身を見られる
root [2] photontree->ScanO
*****
* Row * ENERGY.EN * L.L * B.B * ZENITH_AN * EARTH_AZI * TIME.TIME *
*****
* 0 * 4931.1074 * 186.40201 * 66.69105 * 110.99175 * 30.568527 * 456105604 *
* 1 * 2988.297 * 57.856243 * 52.6086 * 111.2355 * 331.76807 * 456105605 *
* 2 * 516.18414 * 82.95196 * 67.50891 * 111.29916 * 352.37335 * 456105605 *
* 3 * 442.3429 * 205.57336 * 63.7599 * 115.18545 * 38.6572 * 456105605 *
* 4 * 1036.0813 * 42.96566 * 39.94485 * 113.50167 * 314.379 * 456105605 *
* 5 * 778.4724 * 53.01713 * 48.3351 * 111.54488 * 326.1625 * 456105605 *
* 6 * 367.00674 * 184.2203 * 69.56075 * 112.50474 * 27.951687 * 456105606 *
* 7 * 404.6052 * 45.00639 * 39.51818 * 111.96931 * 314.87454 * 456105606 *
* 8 * 77387.74 * 168.77509 * 2.4389622 * 50.717117 * 58.118896 * 456105606 *
* 9 * 594.9343 * 126.68344 * 74.78837 * 111.78197 * 9.015549 * 456105606 *
* 10 * 902.35706 * 46.04191 * 39.77542 * 111.43423 * 315.57397 * 456105607 *
* 11 * 330.524 * 53.571083 * 50.973965 * 112.65732 * 328.8631 * 456105607 *
* 12 * 2155.7253 * 63.219284 * 57.27128 * 111.39319 * 337.88257 * 456105607 *
* 13 * 2252.0422 * 67.19626 * 66.6195 * 115.30289 * 347.35477 * 456105607 *
* 14 * 1174.9325 * 63.162838 * 58.240513 * 111.98207 * 338.72086 * 456105607 *
* 15 * 920.9188 * 37.06417 * 77.51638 * 127.42241 * 355.65735 * 456105608 *
* 16 * 1097.4371 * 195.36998 * 72.10814 * 116.78786 * 28.872795 * 456105609 *
* 17 * 643.58124 * 112.97637 * 74.5411 * 112.28110 * 5.351816 * 456105610 *
* 18 * 352.52347 * 189.8112 * 67.80789 * 112.47748 * 31.072926 * 456105610 *
* 19 * 848.8125 * 43.245518 * 44.524258 * 115.70504 * 319.19974 * 456105610 *
* 20 * 494.76962 * 163.18736 * 74.97039 * 113.34265 * 19.249445 * 456105611 *
* 21 * 414.68167 * 57.930553 * 51.78563 * 110.91196 * 331.4011 * 456105611 *
* 22 * 338.51016 * 56.576687 * 49.816 * 110.54596 * 329.1406 * 456105611 *
* 23 * 321.63272 * 196.61794 * 63.111916 * 111.31907 * 36.928875 * 456105612 *
* 24 * 469.16013 * 50.57432 * 48.78799 * 113.37095 * 326.03464 * 456105612 *
Type <CR> to continue or q to quit ==> q
*****
(long long) 25
root [3] |
```

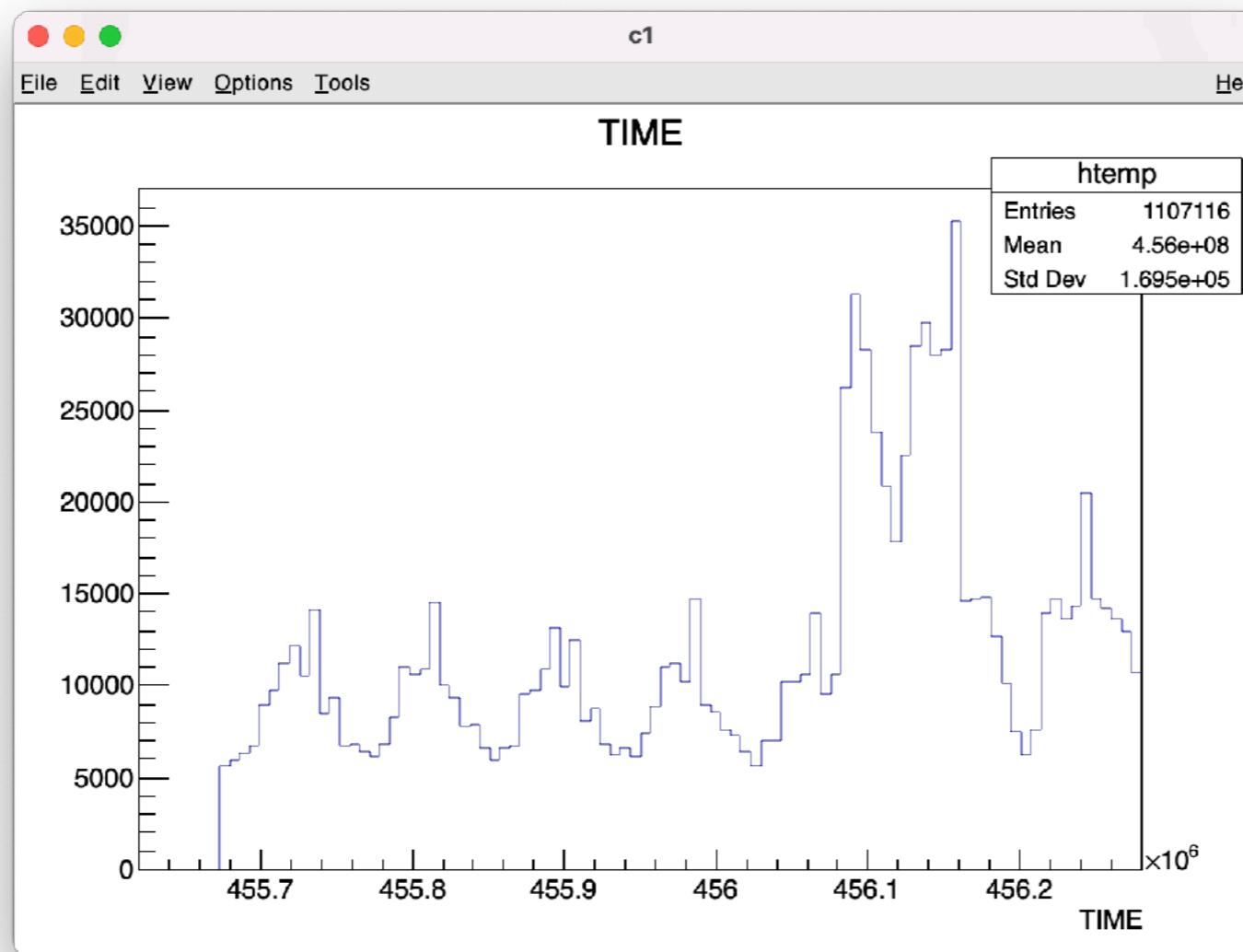
ブランチの名前・詳細を確認するにはPrint

```
root [2] photontree->Print()
*****
*Tree   :photontree: Photon tree
*Entries : 1107116 : Total =      53273919 bytes File Size = 37829012 *
*          : Tree compression factor = 1.41
*****
*Br    0 :ENERGY   : ENERGY/D
*Entries : 1107116 : Total Size= 8878629 bytes File Size = 6151890 *
*Baskets : 216 : Basket Size= 3200000 bytes Compression= 1.44 *
*.....
*Br    1 :L       : L/D
*Entries : 1107116 : Total Size= 8877529 bytes File Size = 6246390 *
*Baskets : 216 : Basket Size= 3199488 bytes Compression= 1.42 *
*.....
*Br    2 :B       : B/D
*Entries : 1107116 : Total Size= 8877529 bytes File Size = 7480402 *
*Baskets : 216 : Basket Size= 3199488 bytes Compression= 1.19 *
*.....
*Br    3 :ZENITH_ANGLE : ZENITH_ANGLE/D
*Entries : 1107116 : Total Size= 8880054 bytes File Size = 6331020 *
*Baskets : 217 : Basket Size= 3200512 bytes Compression= 1.40 *
*.....
*Br    4 :EARTH_AZIMUTH_ANGLE : EARTH_AZIMUTH_ANGLE/D
*Entries : 1107116 : Total Size= 8881601 bytes File Size = 6256451 *
*Baskets : 217 : Basket Size= 3201536 bytes Compression= 1.42 *
*.....
*Br    5 :TIME     : TIME/D
*Entries : 1107116 : Total Size= 8878189 bytes File Size = 5350740 *
*Baskets : 216 : Basket Size= 3200000 bytes Compression= 1.66 *
*.....
root [3] |
```

# TTreeの内容を描画してみる

TIME(到来時刻)のヒストグラムを描く

```
root [3] photontree->Draw("TIME")
Info in <TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [4] ■
```

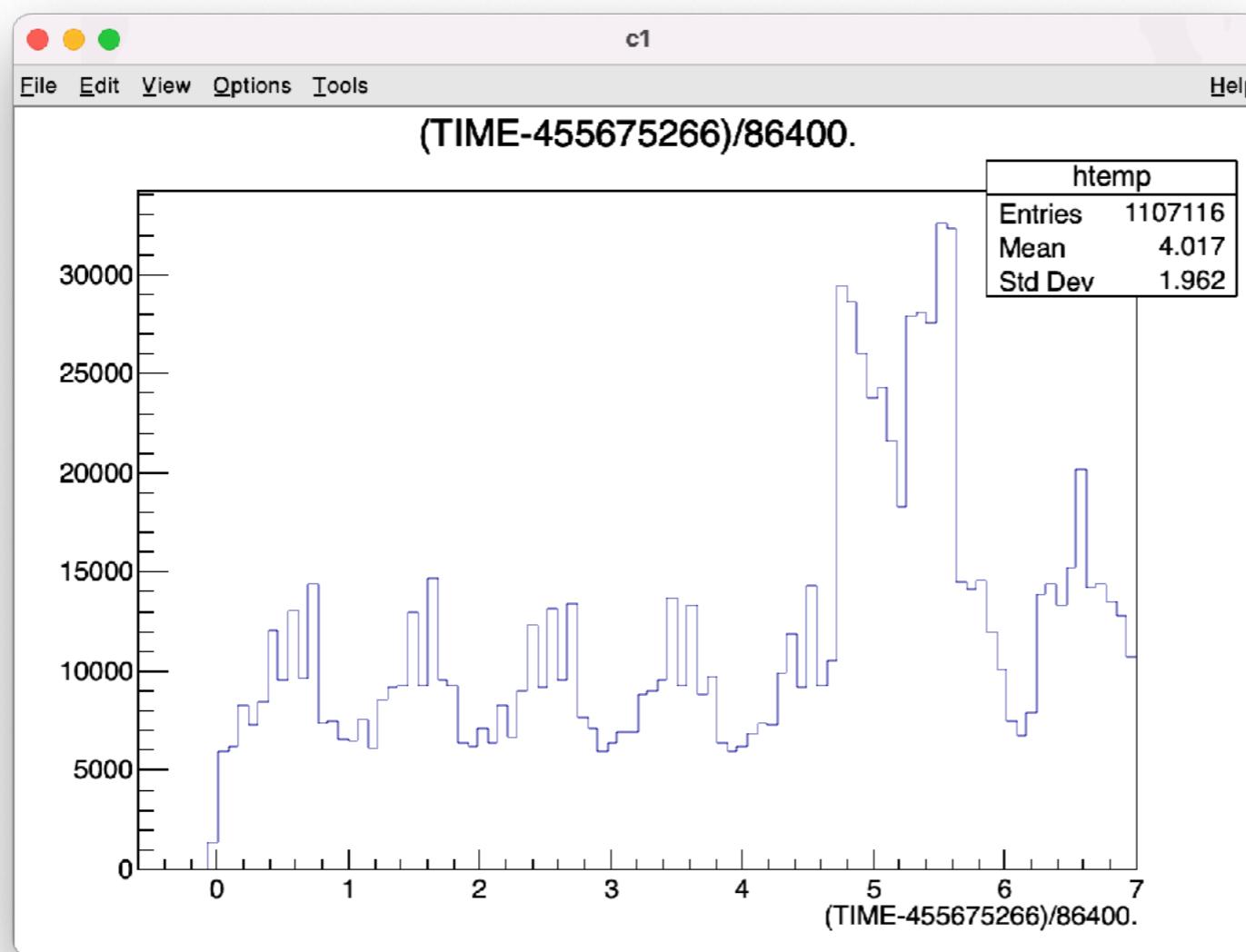


数字が大き過ぎてわかりにくい...

# 計算を加えた結果を描画できる

最初のイベントのTIMEを引いてゼロで始まるようにし、86400秒/日で割って単位を日にする

```
root [4] photontree->Draw("(TIME-455675266)/86400.")  
root [5] █
```

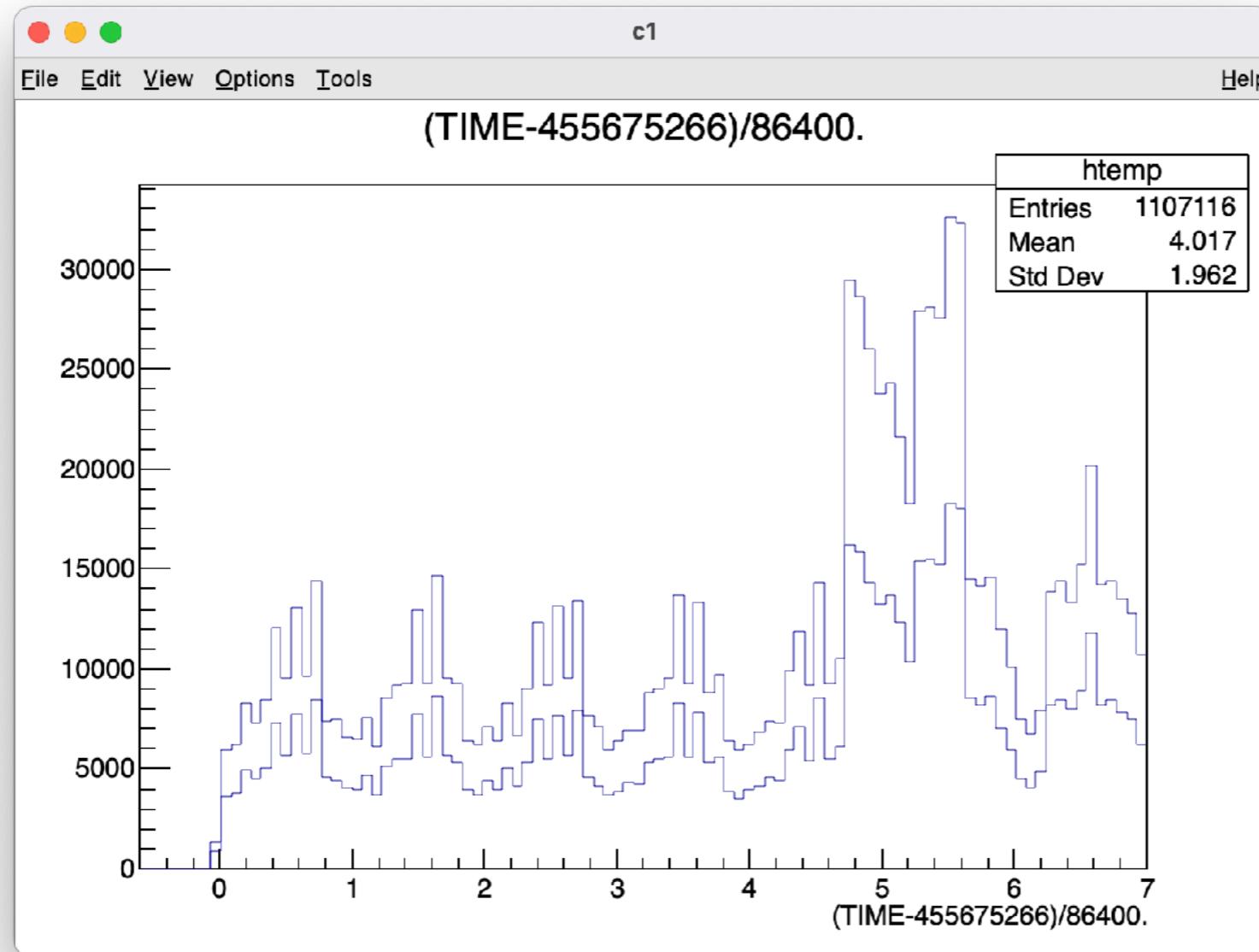


- 1週間のデータであること
- 1日周期の成分があること
- 5～6日目に何かあったことが分かる

# Selectionをかけた場合の描画

第2引数で 500 MeV以上のエントリのみ選んでいる\* 第3引数"same"で重ね描き

```
root [2] photontree->Draw("(TIME-455675266)/86400.", "ENERGY>500", "SAME")
(long long) 647818
root [3] ■
```



\*実際には、第2引数は重みを与えており（式の評価結果が偽であるエントリの重みを0にしている）

# きちんと処理をしてTTreeを作る

- 単にファイルを読み込むのでは不十分な場合

```
// fill_tree.C

void fill_tree(){
    // Open output file
    TFile *file = new TFile("allsky_photons_1GeV.root", "RECREATE");
    file->cd();

    // Create TTree
    TTree *tree = new TTree("photontree", "Photon tree");

    Float_t energy, l, b, zenith, azimuth;
    Double_t time;
    tree->Branch("ENERGY", &energy, "ENERGY/F");
    tree->Branch("L", &l, "L/F");
    tree->Branch("B", &b, "B/F");
    tree->Branch("ZENITH_ANGLE", &zenith, "ZENITH_ANGLE/F");
    tree->Branch("EARTH_AZIMUTH_ANGLE", &azimuth, "EARTH_AZIMUTH_ANGLE/F");
    tree->Branch("TIME", &time, "TIME/D");

    // Open input file
    ifstream ifs("../lat-data/allsky_photons_300MeV.dat");
    // Check if input file is open
    if(!ifs.is_open()){
        cout << "Input file is not opened!" << endl;
        return 0;
    }

    while(ifs >> energy >> l >> b >> zenith >> azimuth >> time){
        if(energy>=1000){ // ENERGY>=1 GeV
            l = l>180 ? l-360.:l; // Modify Galactic longitude to center Origin (Galactic Center)
            /* Same as
            if(l>180.)
                l = l-360.;
            */
            tree->Fill();
        }
    }

    // Save TTree
    tree->Write();
    file->Close();
}
```

## fill\_tree.C

# TBranchの設定

- TTreeにTBranchを作り、値を格納する変数を用意する

```
// fill_tree.C

void fill_tree(){
    // Open output file
    TFile *file = new TFile("allsky_photons_1GeV.root", "RECREATE");
    file->cd();

    // Create TTree
    TTree *tree = new TTree("photontree", "Photon tree");

    Float_t energy, l, b, zenith, azimuth;    値を入れる変数（のアドレス）を定義する
    Double_t time;                           TTreeにTBranchを作る
    tree->Branch("ENERGY", &energy, "ENERGY/F");          名前、アドレス、型を指定する
    tree->Branch("L", &l, "L/F");
    tree->Branch("B", &b, "B/F");
    tree->Branch("ZENITH_ANGLE", &zenith, "ZENITH_ANGLE/F");
    tree->Branch("EARTH_AZIMUTH_ANGLE", &azimuth, "EARTH_AZIMUTH_ANGLE/F");
    tree->Branch("TIME", &time, "TIME/D");
```

# 値を詰める

- 変数に値を入れ、各ブランチに詰める

```
// Open input file
ifstream ifs("../lat-data/allsky_photons_300MeV.dat");    データのファイルを開く
// Check if input file is open
if(!ifs.is_open()){
    cout << "Input file is not opened!" << endl;           開けているか確認
    return 0;
}

while(ifs >> energy >> l >> b >> zenith >> azimuth >> time){      データファイルを1行（1イベント）ずつ読む
    if(energy>=1000){ // ENERGY>=1 GeV                      読み込んだ値でカット
        l = l>180 ? l-360.:l; // Modify Galactic longitude to center Origin (Galactic Center)
        /* Same as
           if(l>180.)
           l = l-360.;
        */
        tree->Fill();  各ブランチに各変数の値を詰める
    }
}

// Save TTree
tree->Write();
file->Close();
}
```

# 新しく作ったTTreeを見る

```
(base) mitsunari@White-CTA-iMac examples % root allsky_photons_1GeV.root
-----
| Welcome to ROOT 6.26/00          https://root.cern |
| (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers |
| Built for macosx64 on Mar 03 2022, 06:51:13 |
| From tags/v6-26-00@v6-26-00      |
| With Apple clang version 13.1.6 (clang-1316.0.21.2) |
| Try '.help', '.demo', '.license', '.credits', '.quit'/.q' |
-----

root [0]
Attaching file allsky_photons_1GeV.root as _file0...
(TFile *) 0x7fa49ff13c20
root [1] .ls
TFile**      allsky_photons_1GeV.root
TFile*       allsky_photons_1GeV.root
  KEY: TTree   photontree;1   Photon tree
root [2] photontree->Scan()
*****
*   Row * ENERGY.EN    L.L *      B.B * ZENITH_AN * EARTH_AZI * TIME.TIME *
*****
*   0   * 1734.6042 * 56.416725 * -1.585244 * 40.337535 * 9.9737558 * 455675267 *
*   1   * 2250.6831 * 128.35607 * 13.668188 * 113.05253 * 15.051140 * 455675268 *
*   2   * 1007.6257 * 111.42964 * -2.184901 * 93.226303 * 27.278684 * 455675268 *
*   3   * 1477.3946 * 130.26603 * 63.306930 * 112.53724 * 320.87927 * 455675270 *
*   4   * 1578.6613 * 4.1571307 * 0.4715414 * 19.673778 * 250.33096 * 455675270 *
*   5   * 1496.0278 * 32.0723 * 1.9438155 * 21.390857 * 337.21276 * 455675270 *
*   6   * 14584.369 * 131.36285 * 68.659835 * 111.61387 * 315.19024 * 455675273 *
*   7   * 1452.1862 * 24.438737 * 5.2494506 * 20.784582 * 311.78186 * 455675283 *
*   8   * 1456.3000 * 123.85214 * 36.965881 * 111.09139 * 349.61956 * 455675284 *
*   9   * 1482.8450 * 128.05723 * 54.517719 * 113.36602 * 330.55477 * 455675286 *
*  10   * 1616.6696 * 125.56723 * 15.727902 * 110.14595 * 12.371626 * 455675287 *
*  11   * 3737.7441 * 127.11082 * 61.620899 * 111.87662 * 323.02877 * 455675288 *
*  12   * 1289.2533 * 23.13661 * -10.65462 * 6.2149453 * 341.06066 * 455675290 *
*  13   * 1221.6317 * 125.61043 * 56.086135 * 111.88708 * 329.06173 * 455675292 *
*  14   * 3770.4719 * 131.09960 * 0.5792256 * 111.85050 * 29.494863 * 455675292 *
*  15   * 6341.7407 * 126.16232 * 61.842865 * 111.48688 * 322.87637 * 455675293 *
*  16   * 1199.8693 * 135.33554 * 71.642005 * 112.64481 * 311.69622 * 455675296 *
*  17   * 1356.7735 * -19.41369 * 5.1698908 * 43.163589 * 229.35191 * 455675300 *
*  18   * 13307.443 * 130.53462 * 1.9224127 * 111.32248 * 27.910661 * 455675300 *
*  19   * 1954.1691 * 127.96235 * 4.1090612 * 109.24486 * 24.917682 * 455675304 *
*  20   * 3121.4338 * 124.80567 * 35.733749 * 111.57348 * 351.07711 * 455675310 *
*  21   * 1254.5511 * 128.08013 * 24.870594 * 113.24803 * 3.0874838 * 455675311 *
*  22   * 2188.9477 * 123.76422 * 46.09235 * 111.28762 * 339.92865 * 455675312 *
*  23   * 1736.6676 * 129.25770 * 14.807723 * 112.55344 * 14.076561 * 455675315 *
*  24   * 1082.9132 * 131.92800 * 61.728233 * 114.70411 * 322.58300 * 455675316 *

Type <CR> to continue or q to quit ==> c
```

ENERGYが1000 (1 GeV) 以上になっている

Lが-180から180の間になっている

**TH2**

# リメンバー 第2回の内容

- ヒストグラムのクラスはTH1D.
- 新しく自力で書いてみる

```
// first_hist.C
1 void first_hist(){
2
3     int total_bin = 10;
4     int min_bin = 0;
5     int max_bin = 10;
6     TH1D* hist = new TH1D("hist", "hist",
7                           total_bin, min_bin, max_bin);
8
9     hist->Fill(2);
10    hist->Fill(1);
11    hist->Fill(2);
12    hist->Fill(3);
13    hist->Fill(4);
14
15    hist->Draw();
16 }
```

ヒストグラムのクラス、TH1Dのインスタンス(実体)をコンストラクタ(初期化用メソッド)を使って作成する。

形式は TH1D("名前", "タイトル",  
Bin数, 最小値, 最大値);  
ポインタをつかってnewで定義する。

ヒストグラムに対してデータを詰める  
Fill()というメソッドを使う  
ポインタで定義した時は.(ドット)ではなく  
->(アロー)でメソッドを呼ぶ

最後にDraw()メソッドを呼べば  
図が出てくるはず

# 2次元ヒストグラム

```
// hist2D.C

void hist2D(){
    // X-axis
    const Int_t total_xbin = 20;
    const Int_t min_xbin = 0;
    const Int_t max_xbin = 10;
    // Y-axis
    const Int_t total_ybin = 40;
    const Int_t min_ybin = -10;
    const Int_t max_ybin = 10;

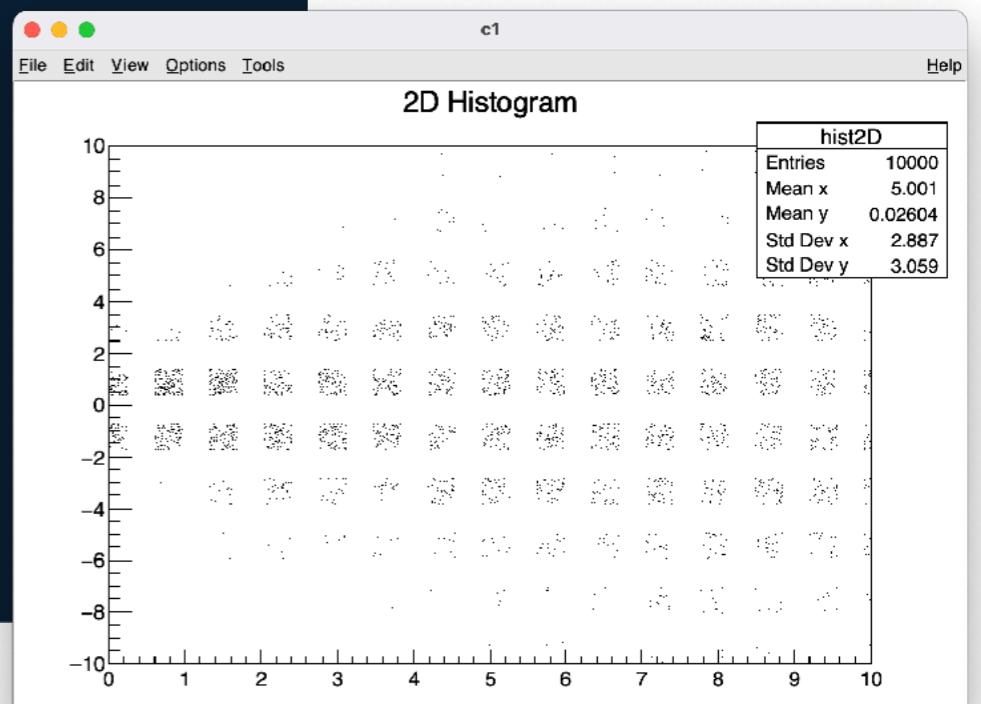
    TH2D * hist2D = new TH2D("hist2D", "2D Histogram",
        total_xbin, min_xbin, max_xbin,
        total_ybin, min_ybin, max_ybin); 名前とタイトルを付けるのは同じ
                                                ビン数・最小値・最大値はX軸とY
                                                軸それぞれに与える

    // Diffusion function
    TF1* func_diffuse = new TF1("func_diffuse",
        "TMath::Exp(-x*x/(4.*[0]*[1]))/(2.*TMath::Sqrt(TMath::Pi())*[0]*[1])",
        min_ybin, max_ybin);
    func_diffuse->SetParameter(0,1); // Diffusion coefficient

    Double_t val;
    Double_t t=0.001;
    while(t<max_xbin){
        func_diffuse->SetParameter(1,t);
        val = func_diffuse->GetRandom(min_ybin, max_ybin);
        //cout << t << ":" << val << endl;
        hist2D->Fill(t, val); ← 値を詰めるには Fill(xの値, yの値)
        t += 0.001;
    }

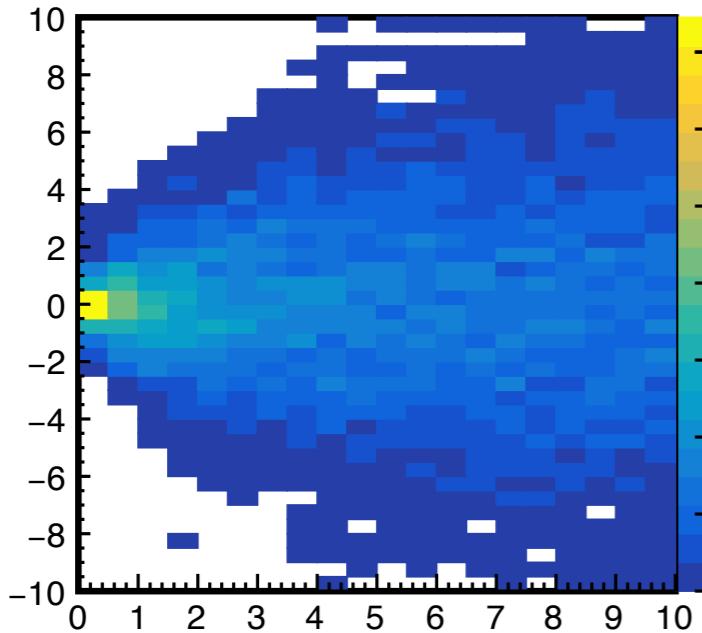
    hist2D->Draw(); 点が打たれているが薄くてよく分からない →
}
```

- hist2D.C

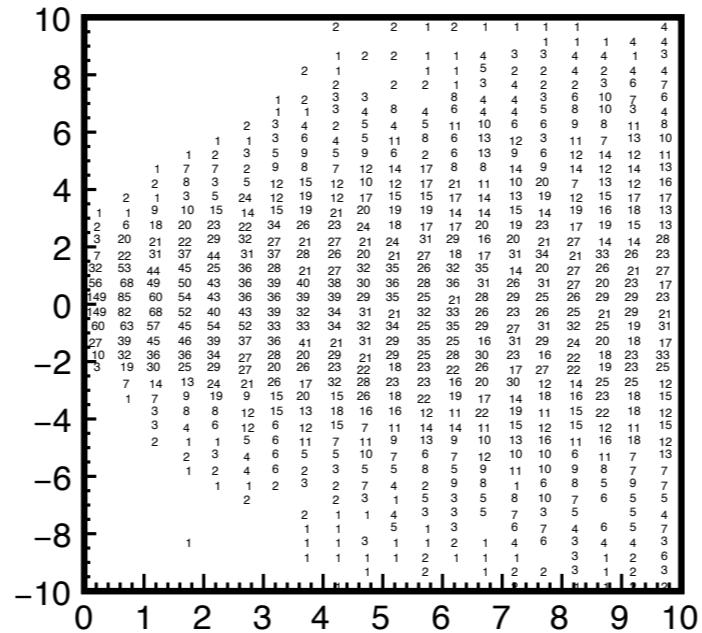


# 分かりやすく表示する

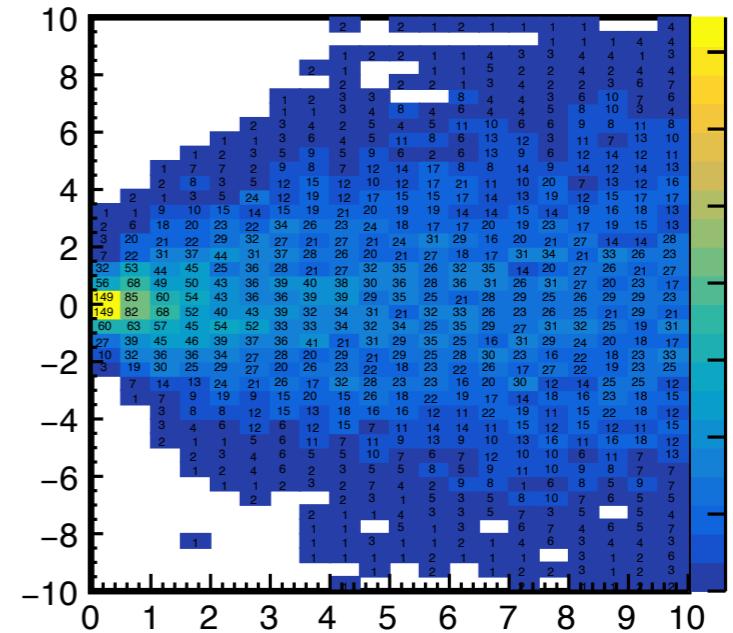
Draw("COLZ")



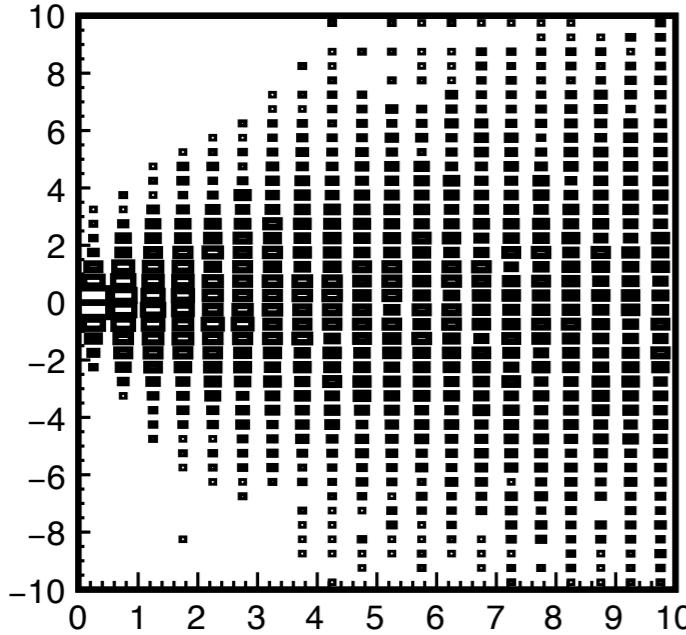
Draw("TEXT")



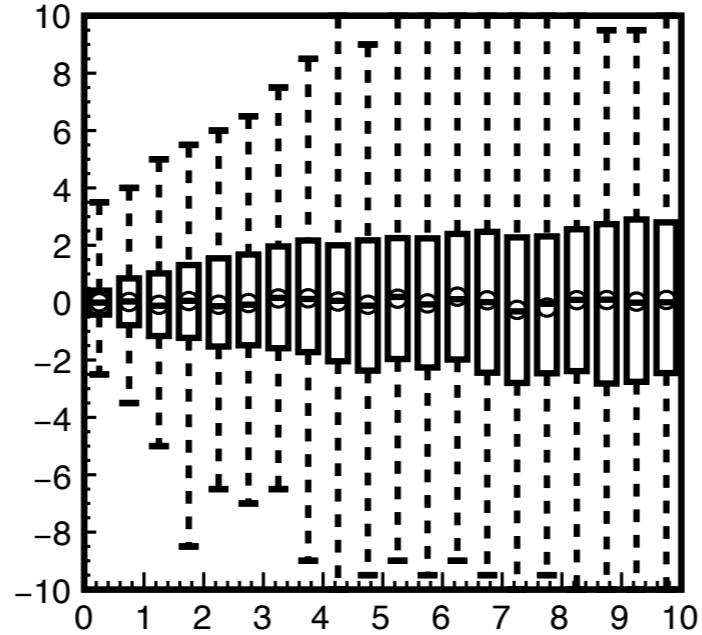
Draw("COLZ TEXT")



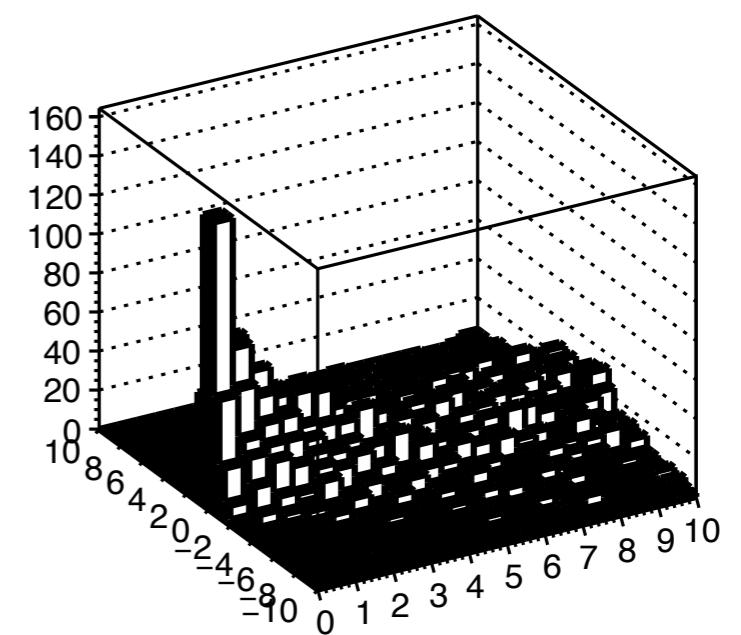
Draw("BOX")



Draw("CANDLE")



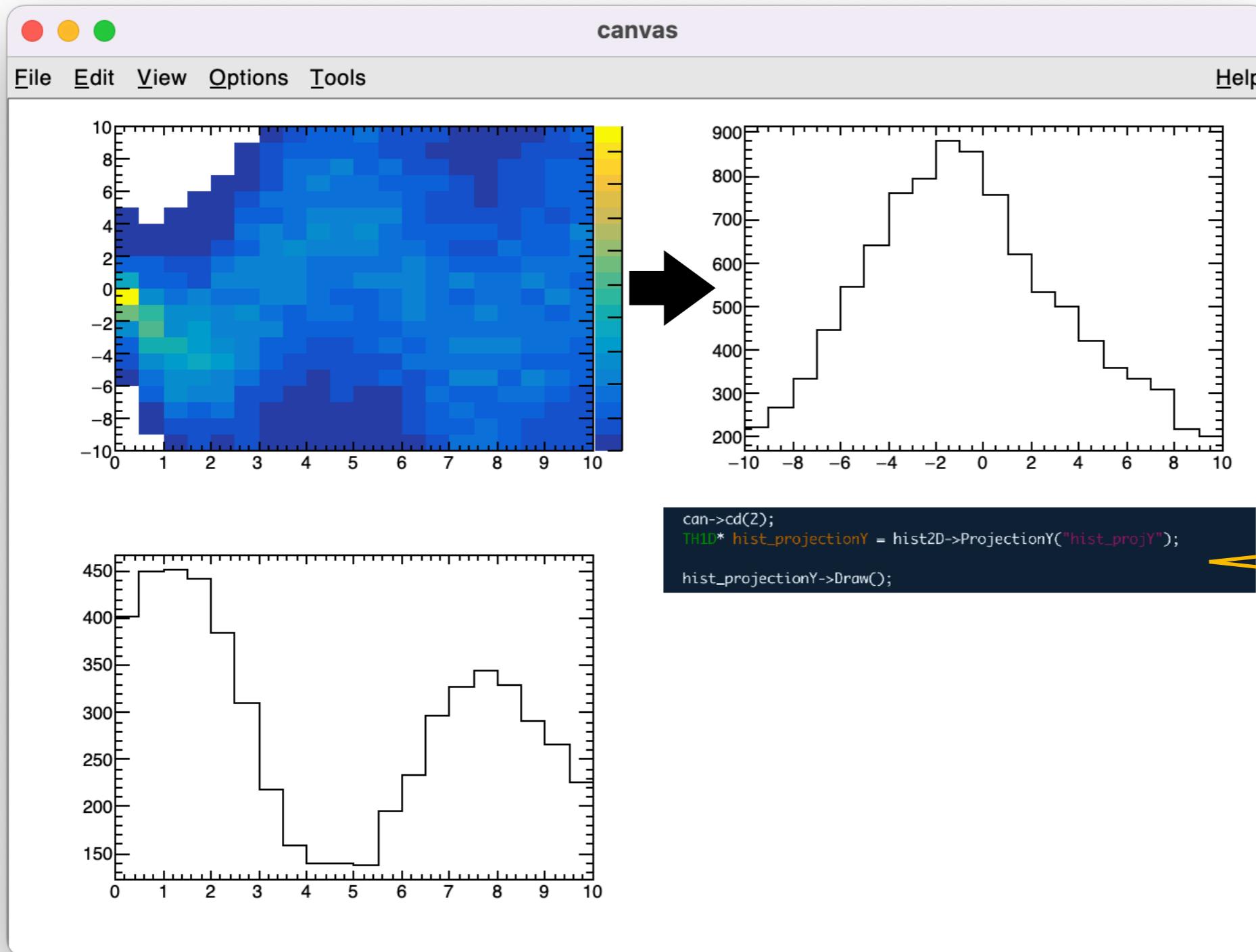
Draw("LEGO")



その他は[このページを参照](#)

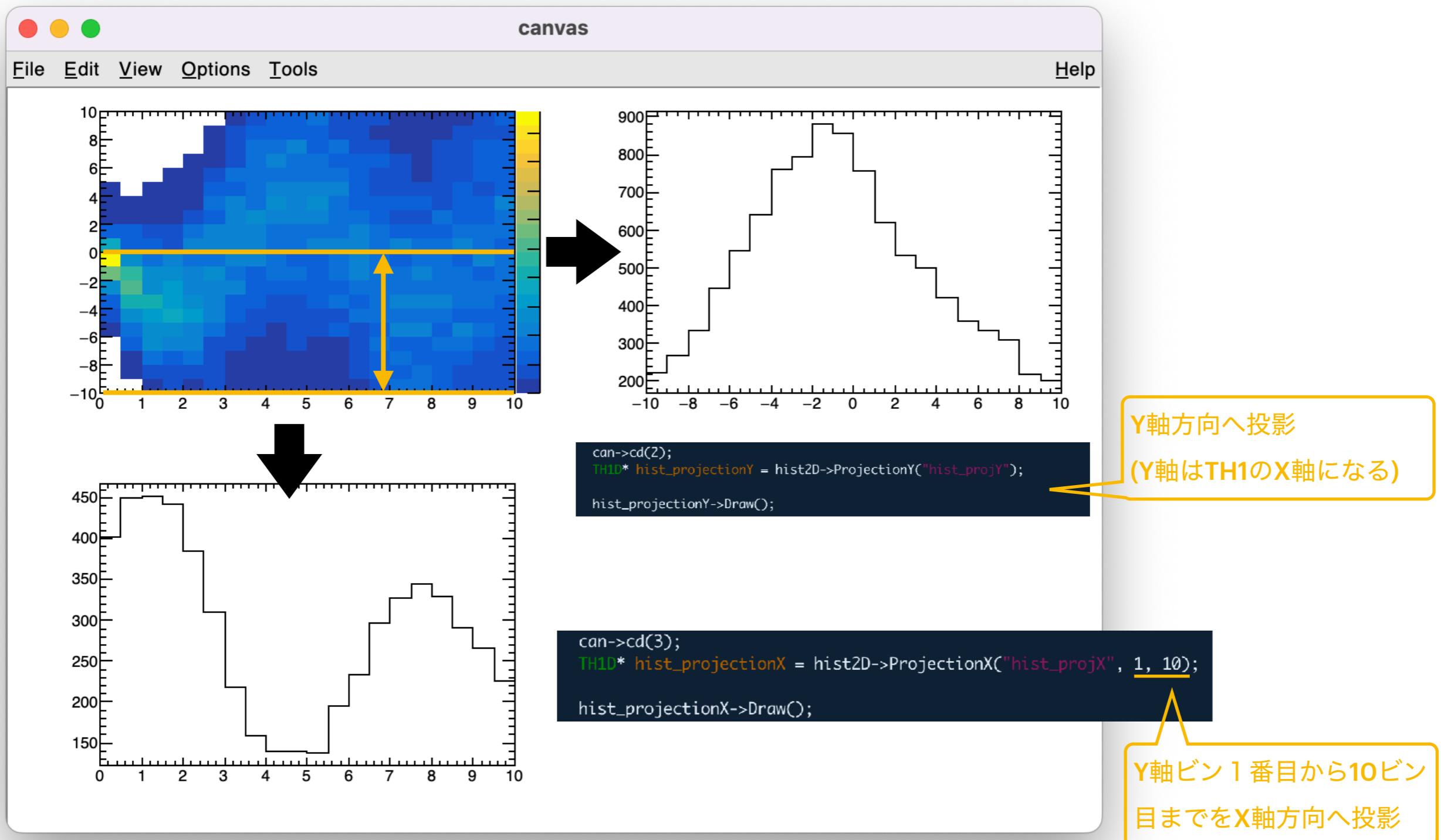
# TH1へ投影する

- project\_hist2D.C



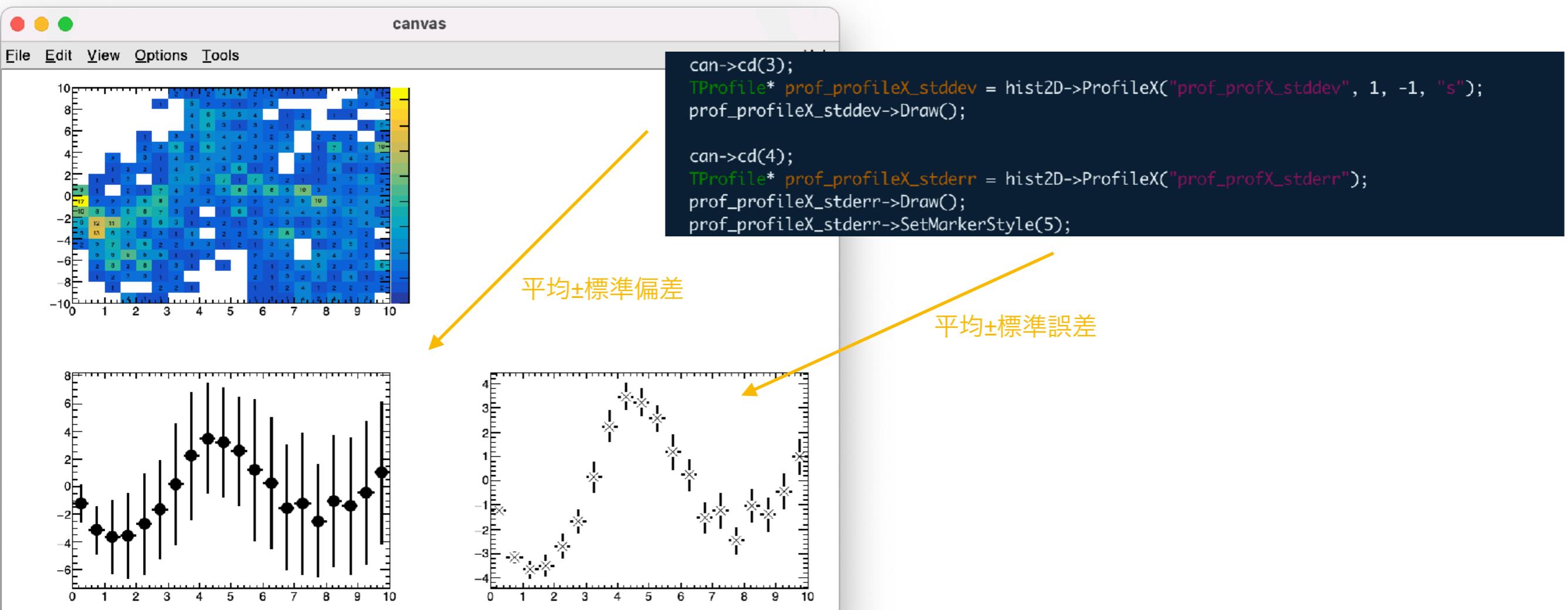
# TH1へ投影する

- project\_hist2D.C



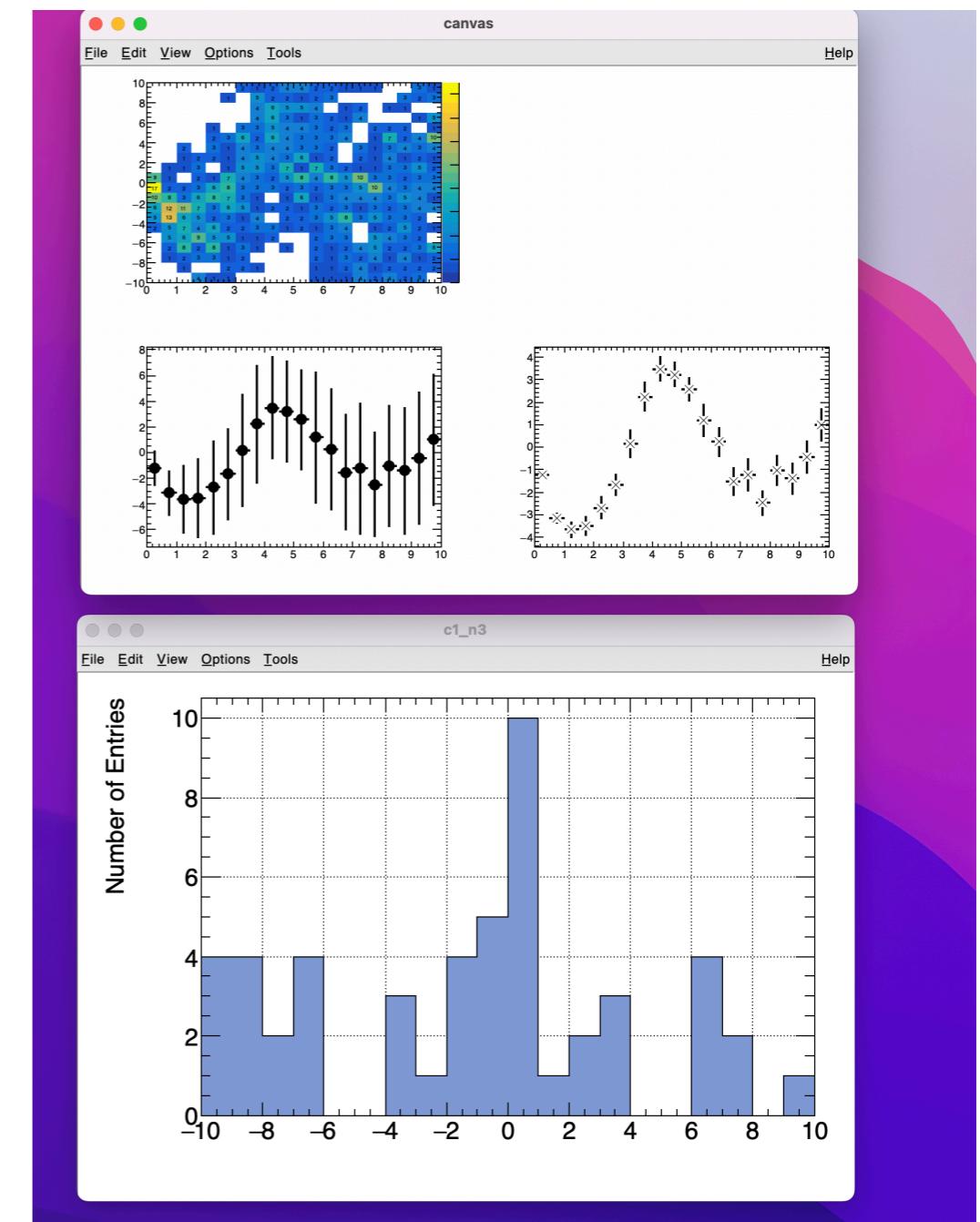
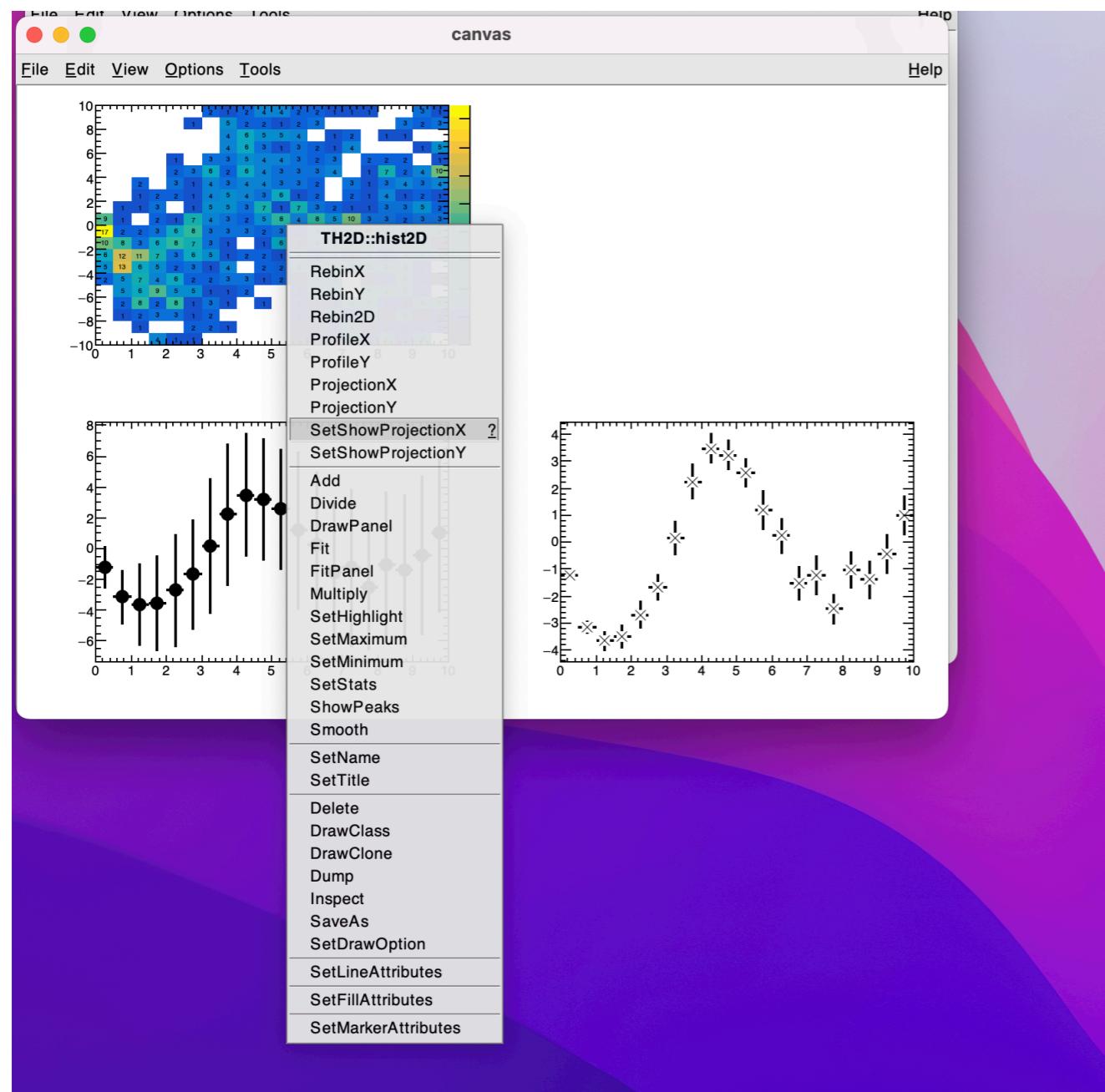
# TProfile

- 2次元ヒストグラムは全体的な傾向・推移を掴むには向いていない場合もある  
→TH2からTProfileを作る
- 平均値と誤差（デフォルトでは標準誤差）を表示
- profile\_hist2D.C



# SetShowProjection

- スライスするビンを変えながらインタラクティブに投影させることができる

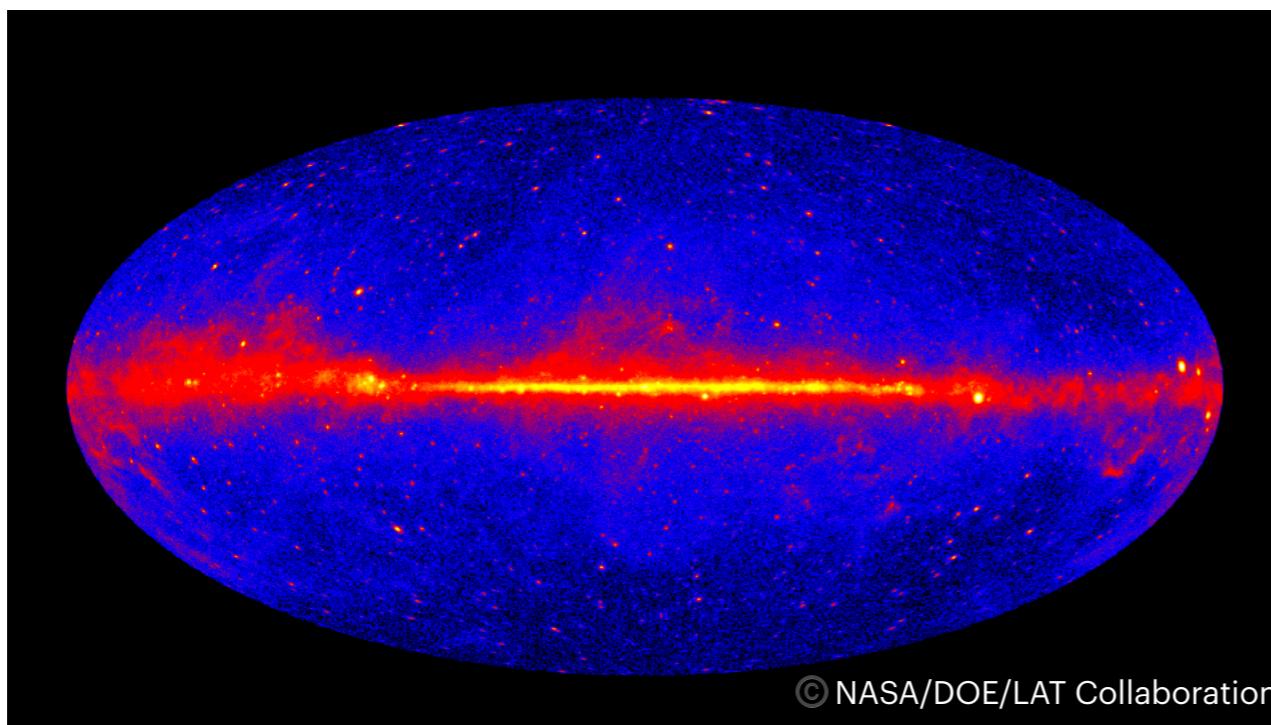


TTreeからTH2を描く

# 目標

- TTreeでDraw("Y軸の変数名:X軸の変数名")とすることで2次元ヒストグラムを描ける
- ガンマ線の到来方向マップを描いてみよう

こんな感じに天の川銀河が見えるはず↓



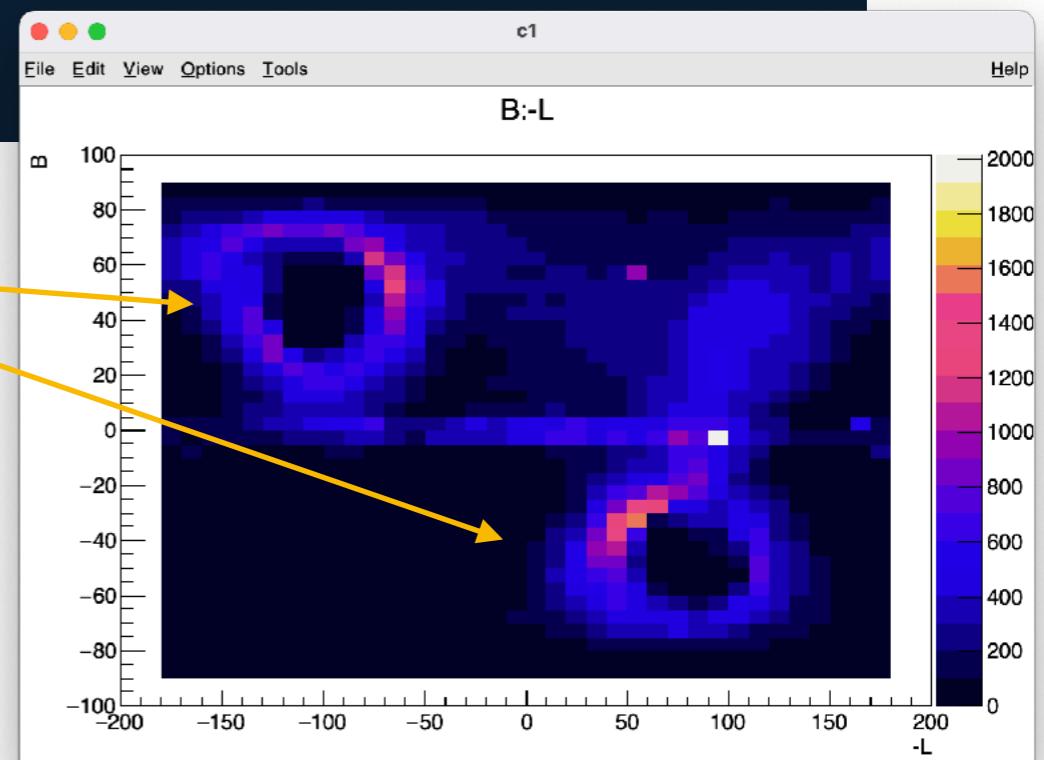
# 2変数をDraw

```
(base) mitsunari@White-CTA-iMac examples % root allsky_photons_1GeV.root
```

```
| Welcome to ROOT 6.26/00           https://root.cern |
| (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers |
| Built for macosx64 on Mar 03 2022, 06:51:13 |
| From tags/v6-26-00@v6-26-00 |
| With Apple clang version 13.1.6 (clang-1316.0.21.2) |
| Try '.help', '.demo', '.license', '.credits', '.quit'/.q' |
```

```
root [0]
Attaching file allsky_photons_1GeV.root as _file0...
(TFile *) 0x7fb85c29fd90
root [1] gStyle->SetPalette(62)
root [2] photontree->Draw("B:-L", "", "COLZ") ただし銀緯Lにはマイナスをつけた（天文学では東を左に描く都合）
Info in <TCanvas::MakeDefCanvas>: created default TCanvas with name c1
(long long) 264822
root [3]
```

何か変な構造がある...



# 謎のガンマ線イベントは何？

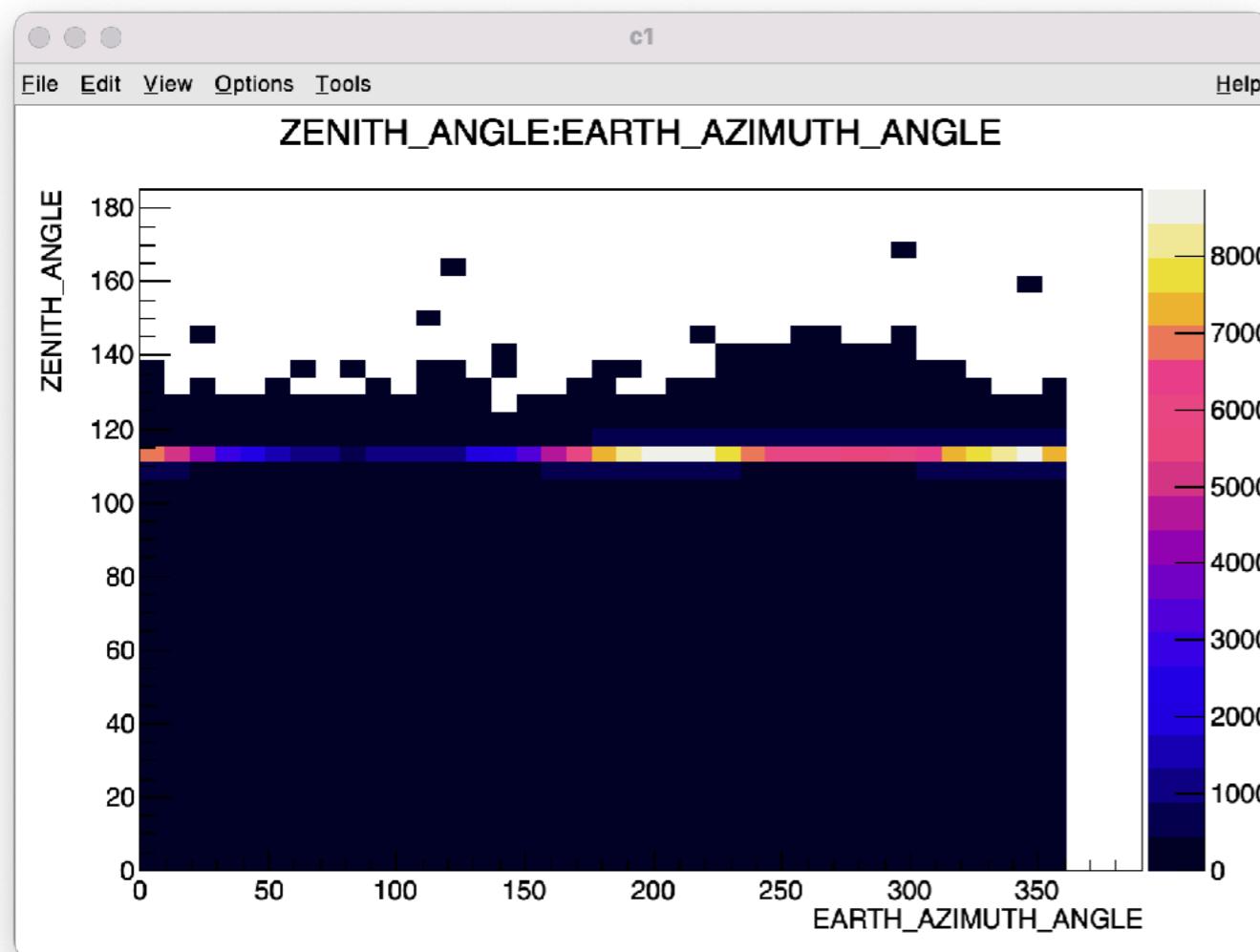
- 改めてScanの出力を見てみると、ZENITH\_ANGLEが $113^\circ$ くらいのエントリがやたら多いことに気づく

```
root [3] photontree->Scan()
*****
*   Row   * ENERGY.EN *      L.L *      B.B * ZENITH_AN * EARTH_AZI * TIME.TIME *
*****
*   0 * 1734.6042 * 56.416725 * -1.585244 * 40.337535 * 9.9737558 * 455675267 *
*   1 * 2250.6831 * 128.35607 * 13.668188 * 113.05253 * 15.051140 * 455675268 *
*   2 * 1007.6257 * 111.42964 * -2.184901 * 93.226303 * 27.278684 * 455675268 *
*   3 * 1477.3946 * 130.26603 * 63.306930 * 112.53724 * 320.87927 * 455675270 *
*   4 * 1578.6613 * 4.1571307 * 0.4715414 * 19.673778 * 250.33096 * 455675270 *
*   5 * 1496.0278 * 32.0723 * 1.9438155 * 21.390857 * 337.21276 * 455675270 *
*   6 * 14584.369 * 131.36285 * 68.659835 * 111.61387 * 315.19024 * 455675273 *
*   7 * 1452.1862 * 24.438737 * 5.2494506 * 20.784582 * 311.78186 * 455675283 *
*   8 * 1456.3000 * 123.85214 * 36.965881 * 111.09139 * 349.61956 * 455675284 *
*   9 * 1482.8450 * 128.05723 * 54.517719 * 113.36602 * 330.55477 * 455675286 *
*  10 * 1616.6696 * 125.56723 * 15.727902 * 110.14595 * 12.371626 * 455675287 *
*  11 * 3737.7441 * 127.11082 * 61.620899 * 111.87662 * 323.02877 * 455675288 *
*  12 * 1289.2533 * 23.13661 * -10.65462 * 6.2149453 * 341.06066 * 455675290 *
*  13 * 1221.6317 * 125.61043 * 56.086135 * 111.88708 * 329.06173 * 455675292 *
*  14 * 3770.4719 * 131.09960 * 0.5792256 * 111.85050 * 29.494863 * 455675292 *
*  15 * 6341.7407 * 126.16232 * 61.842865 * 111.48688 * 322.87637 * 455675293 *
*  16 * 1199.8693 * 135.33554 * 71.642005 * 112.64481 * 311.69622 * 455675296 *
*  17 * 1356.7735 * -19.41369 * 5.1698908 * 43.163589 * 229.35191 * 455675300 *
*  18 * 13307.443 * 130.53462 * 1.9224127 * 111.32248 * 27.910661 * 455675300 *
*  19 * 1954.1691 * 127.96235 * 4.1090612 * 109.24486 * 24.917682 * 455675304 *
*  20 * 3121.4338 * 124.80567 * 35.733749 * 111.57348 * 351.07711 * 455675310 *
*  21 * 1254.5511 * 128.08013 * 24.870594 * 113.24803 * 3.0874838 * 455675311 *
*  22 * 2188.9477 * 123.76422 * 46.09235 * 111.28762 * 339.92865 * 455675312 *
*  23 * 1736.6676 * 129.25770 * 14.807723 * 112.55344 * 14.076561 * 455675315 *
*  24 * 1082.9132 * 131.92800 * 61.728233 * 114.70411 * 322.58300 * 455675316 *
Type <CR> to continue or q to quit ==> q
```

- (ZENITH\_ANGLE, EARTH\_AZIMUTH\_ANGLE) は地球に対する座標
- こっちの座標で2次元ヒストグラムを描いてみる

# 別の座標変数でDraw

```
root [5] photontree->Draw("ZENITH_ANGLE:EARTH_AZIMUTH_ANGLE", "", "COLZ")
(long long) 264822
root [6] █
```



- あるZENITH\_ANGLE(天頂角)に集中している→宇宙ではなく地球由来のガンマ線
- ビンが粗いので細かくしてみる

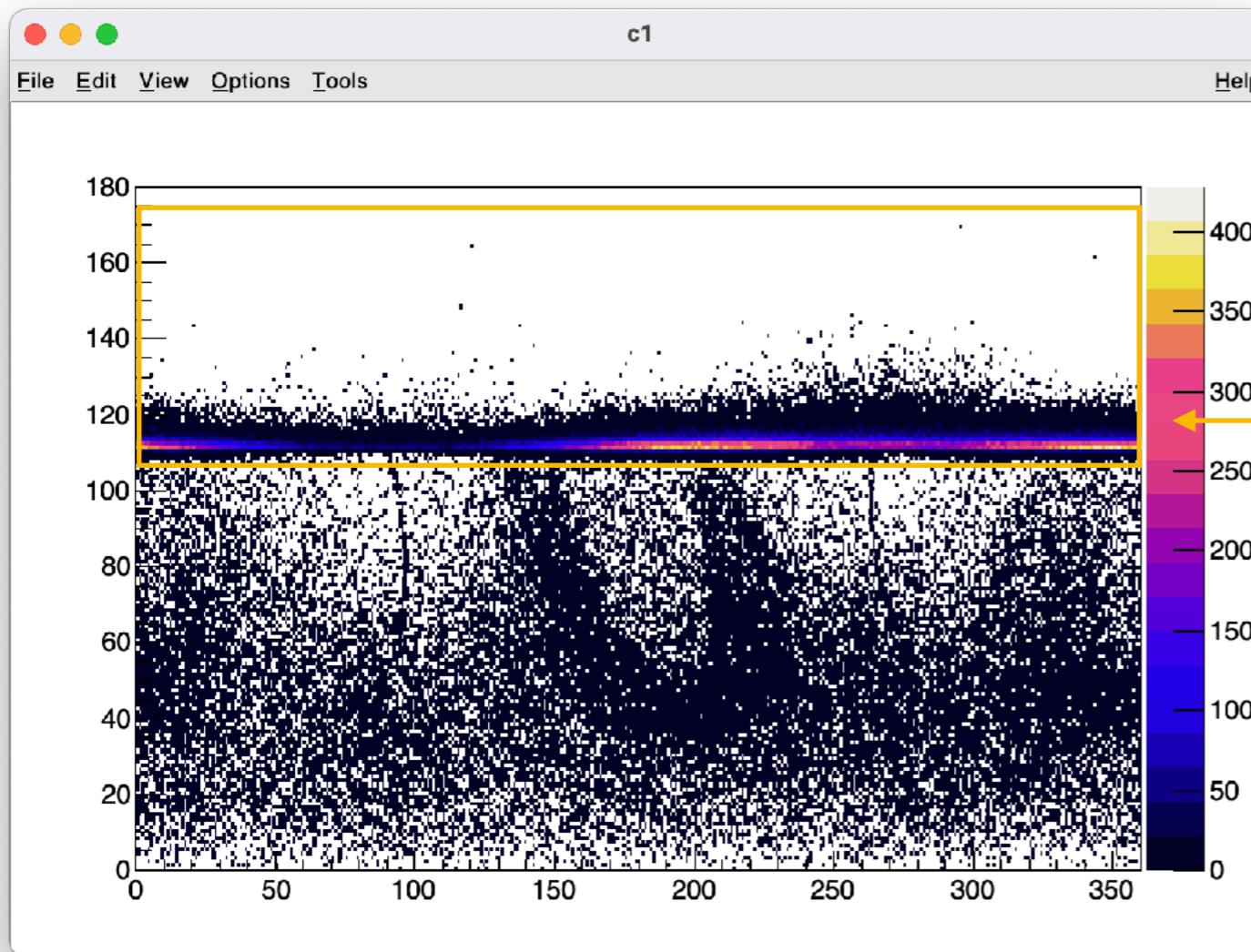
# Drawするヒストグラムの詳細を指定

ヒストグラムに名前をつける

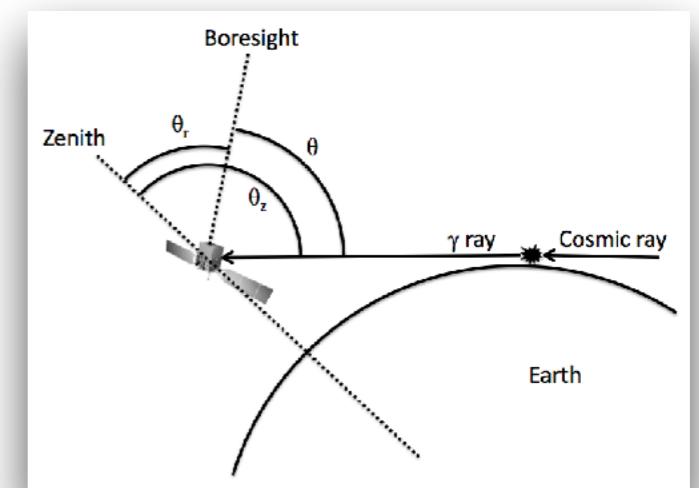
```
root [6] photontree->Draw("ZENITH_ANGLE:EARTH_AZIMUTH_ANGLE>>hist_earth(360, 0, 360, 180, 0, 180)", "", "COLZ")
(long long) 264822
```

(X軸のBIN数, X軸最小値, X軸最大値, Y軸のBIN数, Y軸最小値, Y軸最大値)

こっちはX, Yの順序なので注意



これは“Earth limb”と呼ばれる  
ガンマ線バックグラウンド



# 見つけたバックグラウンドを除外

- plot\_nice\_allskymap.C

```
// plot_nice_allskymap.C

void plot_nice_allskymap(){
    gStyle->SetOptStat(0);
    gStyle->SetPalette(62);

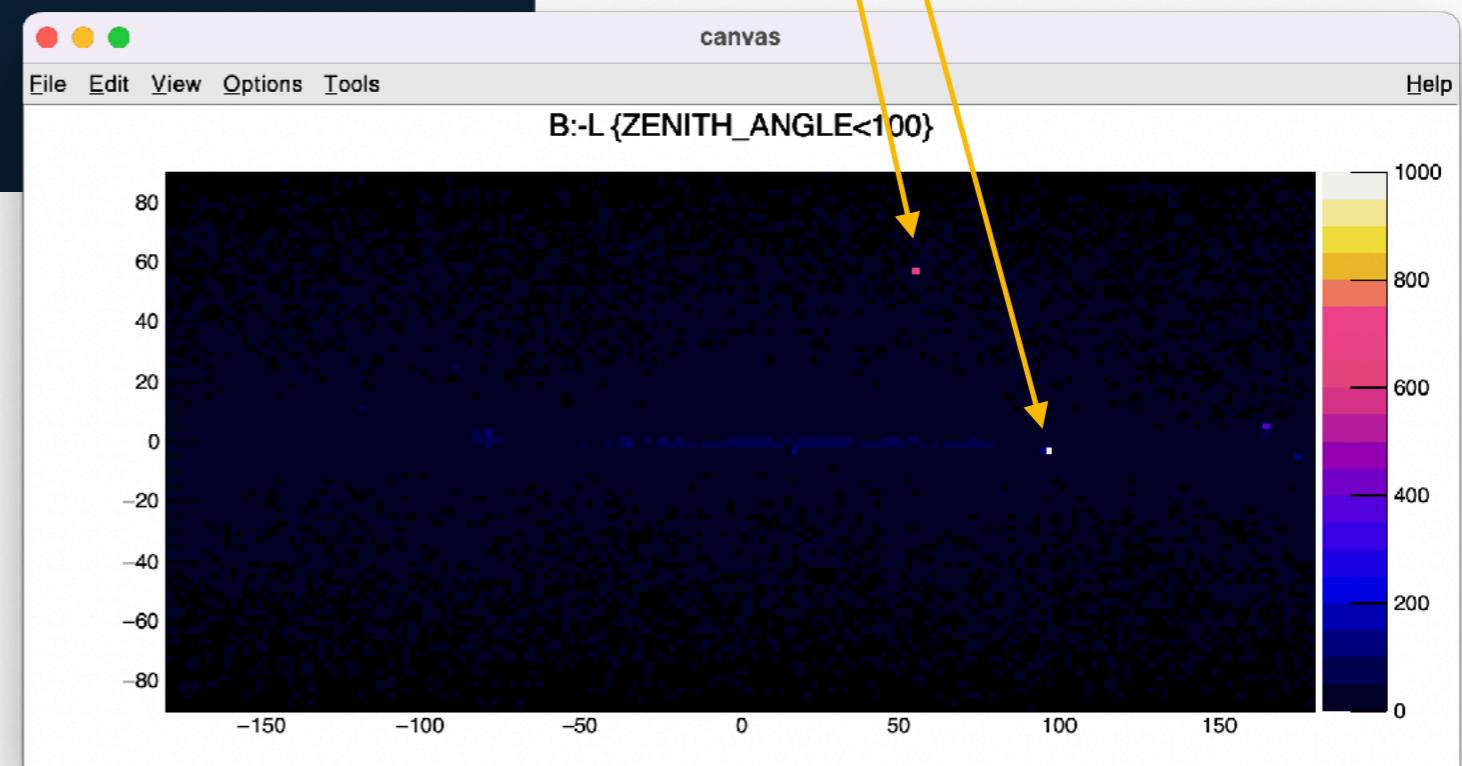
    // Open TTree file
    TFile *file = new TFile("allsky_photons_1GeV.root", "READ");
    // Get object of tree from file
    TTree *photontree = (TTree*)file->Get("photontree");

    TCanvas *can = new TCanvas("canvas", "canvas", 800, 400);

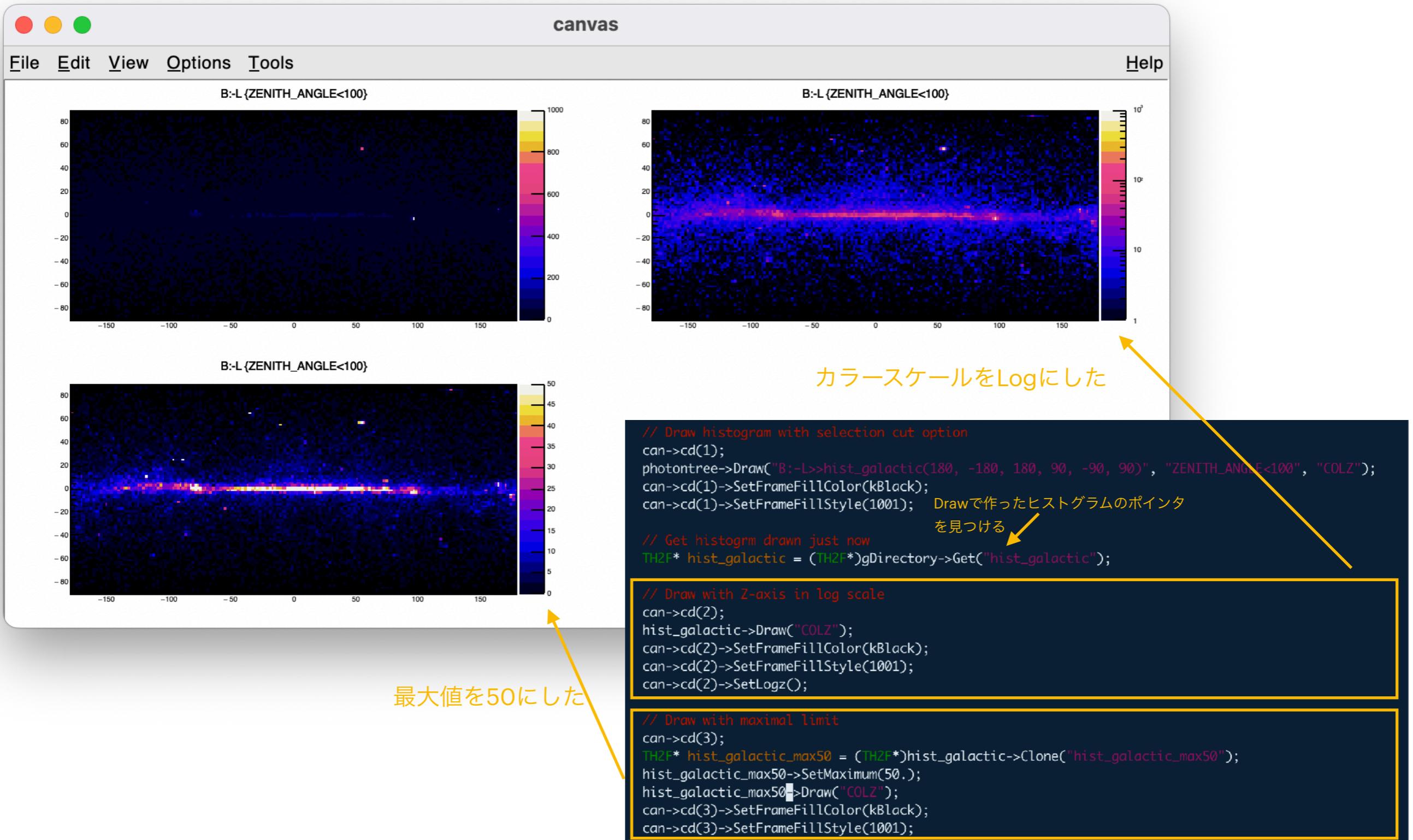
    // Draw histogram with selection cut option
    photontree->Draw("B:-L>>hist_galactic(180, -180, 180, 90, -90, 90)", "ZENITH_ANGLE<100", "COLZ");
    can->SetFrameFillColor(kBlack); ← 背景を黒にする
    can->SetFrameFillStyle(1001);
}
```

Drawの第2引数でEarth  
limbをカットしよう

いくつかのBINだけエントリが多いため  
他が見にくくなっている



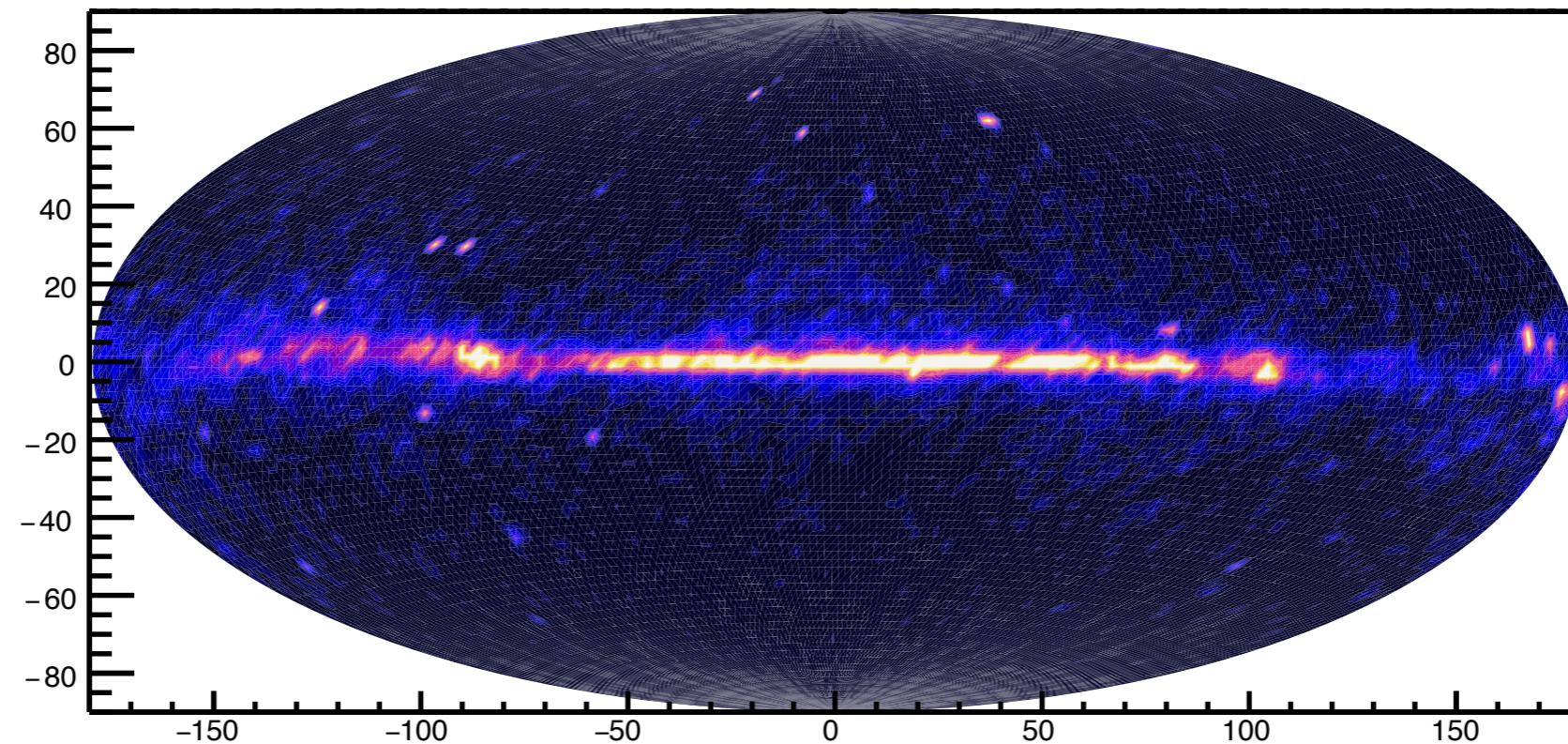
# 見た目を調整する



# おまけとしてmapping(図法)を変えてみる

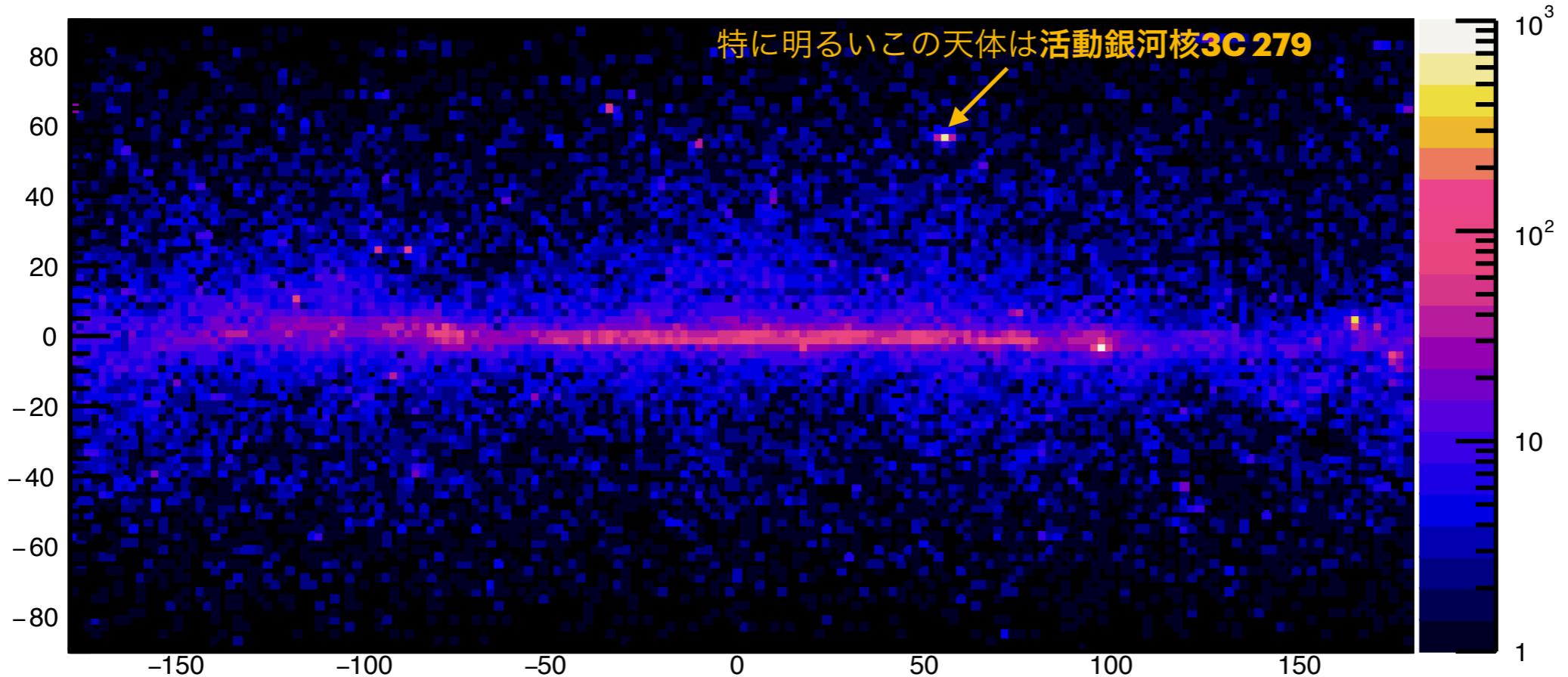
```
// Draw histogram by AITOFF projection  
can->cd(4);  
hist_galactic_max50->Draw("AITOFF"); Aitoff図法
```

B:-L {ZENITH\_ANGLE<100}



- いい感じになった
- 天の川銀河の円盤のほか、いくつかのパルサー星雲や活動銀河核が見える

# 作ったTTreeを読み出して使う



- 作ったTTreeの値を使って、3C 279の回りのイベントだけ選んでみよう
- 座標は  $(l, b) = (305^\circ.104320, 57^\circ.062406)$

# 各エントリの値を読み出す

```
// friend_tree.C

void friend_tree(){
    // Target object: 3C 279
    const Double_t L_obj = 305.104320 * TMath::DegToRad(); // Galactic latitude
    const Double_t B_obj = 57.062406 * TMath::DegToRad(); // Galactic longitude

    // Make 3D-vector toward 3C 279
    const Double_t X_obj = TMath::Cos(L_obj) * TMath::Cos(B_obj);
    const Double_t Y_obj = TMath::Sin(L_obj) * TMath::Cos(B_obj);
    const Double_t Z_obj = TMath::Sin(B_obj);
    cout << "3C 279" << endl;
    cout << "(X, Y, Z)=" << X_obj << "," << Y_obj << "," << Z_obj << endl;

    // Open input file
    TFile *file = new TFile("allsky_photons_300MeV.root", "UPDATE");
    TTree *tree = (TTree*)file->Get("photontree"); ← 光子イベントの入っているツリーを得る

    Double_t l_deg, b_deg;
    tree->SetBranchAddress("L", &l_deg); ← 読み出す値を入れる変数を設定
    tree->SetBranchAddress("B", &b_deg);

    // Open output file
    TFile *file_out = new TFile("allsky_photons_300MeV_3C279.root", "RECREATE");
    TTree * friendtree = new TTree("friendtree", "Friend tree");
    Double_t angle_deg;
    friendtree->Branch("ANGLE_3C279", &angle_deg, "ANGLE_3C279/D");

    Double_t l_ph, b_ph;
    Double_t x_ph, y_ph, z_ph;
    Double_t dotprod, angle_rad;

    for(Int_t i=0; i<tree->GetEntries(); i++){
        tree->GetEntry(i); ← i番目のエントリの各ブランチの値を読み出す
        l_ph = l_deg * TMath::DegToRad(); ← 読み出した値を使って計算ができる
        b_ph = b_deg * TMath::DegToRad();

        // Make 3D-vector toward gamma-ray arrival direction
        x_ph = TMath::Cos(l_ph) * TMath::Cos(b_ph);
        y_ph = TMath::Sin(l_ph) * TMath::Cos(b_ph);
        z_ph = TMath::Sin(b_ph);

        // Calculate dot-product
        dotprod = X_obj*x_ph + Y_obj*y_ph + Z_obj*z_ph;
```

- friend\_tree.C

# Friend Tree

```

TFile *file_out = new TFile("allsky_photons_300MeV_3C279.root", "RECREATE");
TTree * friendtree = new TTree("friendtree", "Friend tree"); ← 新しいTTreeを作成
Double_t angle_deg;
friendtree->Branch("ANGLE_3C279", &angle_deg, "ANGLE_3C279/D");

    計算する値を入れる変数を設定し、ブランチを作る

Double_t l_ph, b_ph;
Double_t x_ph, y_ph, z_ph;
Double_t dotprod, angle_rad;

for(Int_t i=0; i<tree->GetEntries(); i++){
    tree->GetEntry(i);

    l_ph = l_deg * TMath::DegToRad();
    b_ph = b_deg * TMath::DegToRad();

    // Make 3D-vector toward gamma-ray arrival direction
    x_ph = TMath::Cos(l_ph) * TMath::Cos(b_ph);
    y_ph = TMath::Sin(l_ph) * TMath::Cos(b_ph);
    z_ph = TMath::Sin(b_ph);

    // Calculate dot-product
    dotprod = X_obj*x_ph + Y_obj*y_ph + Z_obj*z_ph;

    // Convert dot-product to angle
    if(dotprod>=1)
        angle_rad = 0;
    else if(dotprod<=-1)
        angle_rad = -TMath::Pi();
    else
        angle_rad = TMath::ACos(dotprod);

    angle_deg = angle_rad * TMath::RadToDeg(); 計算した値を変数に入れる
    friendtree->Fill(); 新しいTTreeに詰める

} // Save TTree
file_out->cd();
friendtree->Write();

// Add new tree to main tree as friend
tree->AddFriend(friendtree->GetName(), file_out); Friendとして元のTTreeに設定
                                                AddFriend(ツリーの名前, ツリーのファイル)

file->cd();
tree->Write();
file->Close();
file_out->Close();

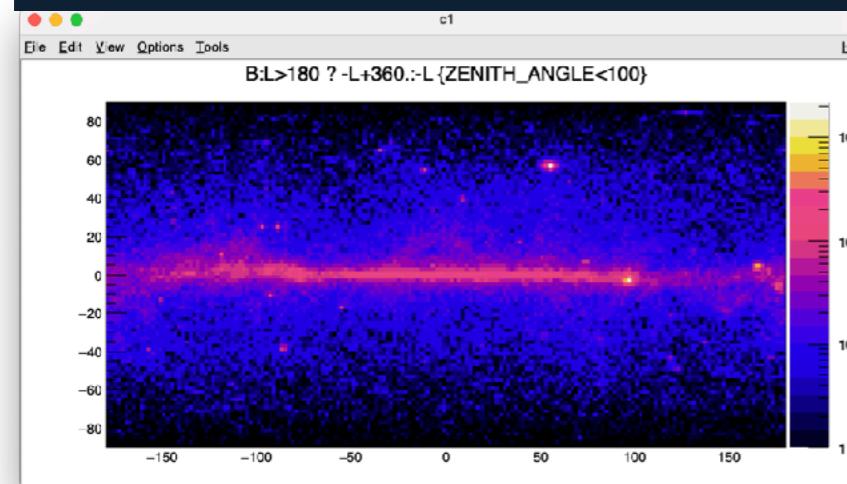
```

- TTreeの各エントリに関する計算をした場合、結果は元のTTreeに追加するよりも新たにTTreeを作って格納するのが推奨されている
- 元のTTreeとはAddFriend関数を用いて結び付けられる

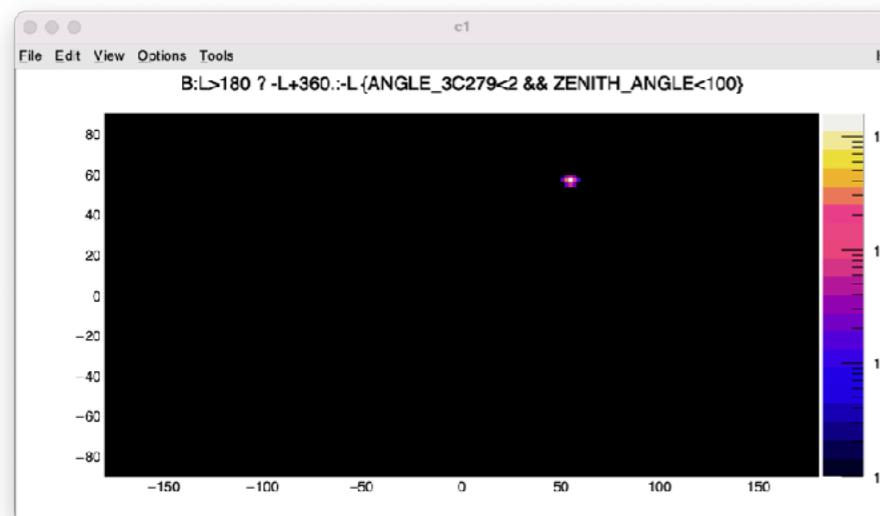
# Friend treeの内容にアクセス

- あるツリーにAddFriendしたツリーのブランチには、そのツリーのブランチと同様にアクセスできる

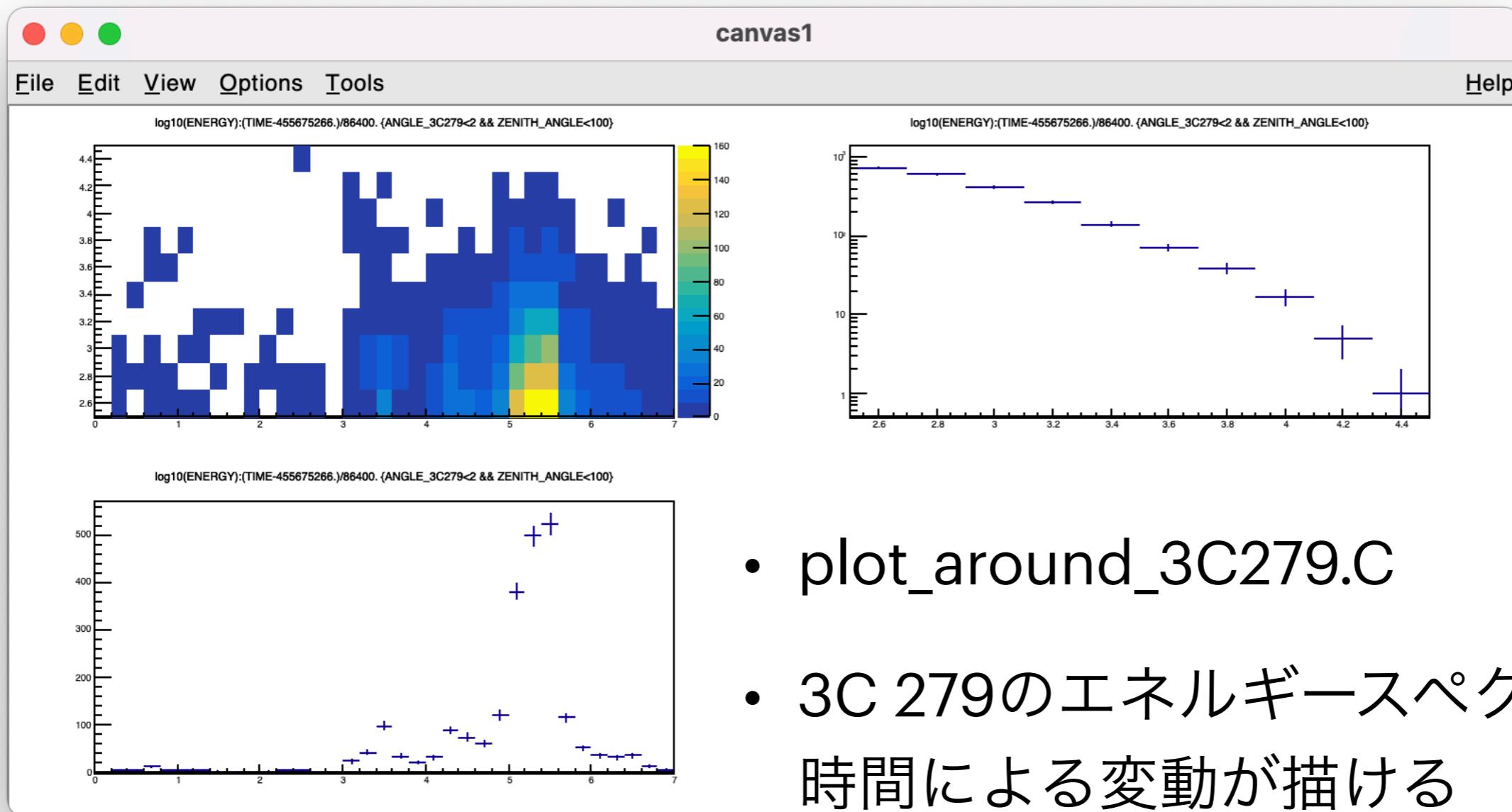
```
root [9] photontree->Draw("B:L>180 ? -L+360.: -L>>(180, -180, 180, 90, -90, 90)", "ZENITH_ANGLE<100", "COLZ")
(long long) 165223
root [10]
```



```
root [8] photontree->Draw("B:L>180 ? -L+360.: -L>>(180, -180, 180, 90, -90, 90)", "ANGLE_3C279<2 && ZENITH_ANGLE
<100", "COLZ")
(long long) 2356
root [9]
```



# 天体由来（の可能性が高い）イベントを解析



# おわり

- やったこと
  - TTreeを作成した
    - テキストファイルから読み込む場合
    - 値を処理して1エントリずつ詰める場合
  - TTreeを1次元／2次元ヒストグラムとして描写した
  - ヒストグラムが分かりやすくなるように見た目を変えた
  - TTreeを描く際にエントリを選別した
  - TTreeに詰められた値を1エントリずつ読み出した
  - 別に作成したTTreeをfriendとして一緒にアクセスできるようにした
- 質問のススメ
  - ROOTのヒストグラム、ツリーの機能は極めて豊富で、使い方の情報が見つけにくい場合もあるのでどんどん質問すると良いです。
  - YMAPへの参加、ROOT講習会の運営・講師としての参加も歓迎します!!  
<http://www.icrr.u-tokyo.ac.jp/YMAP/join.html>