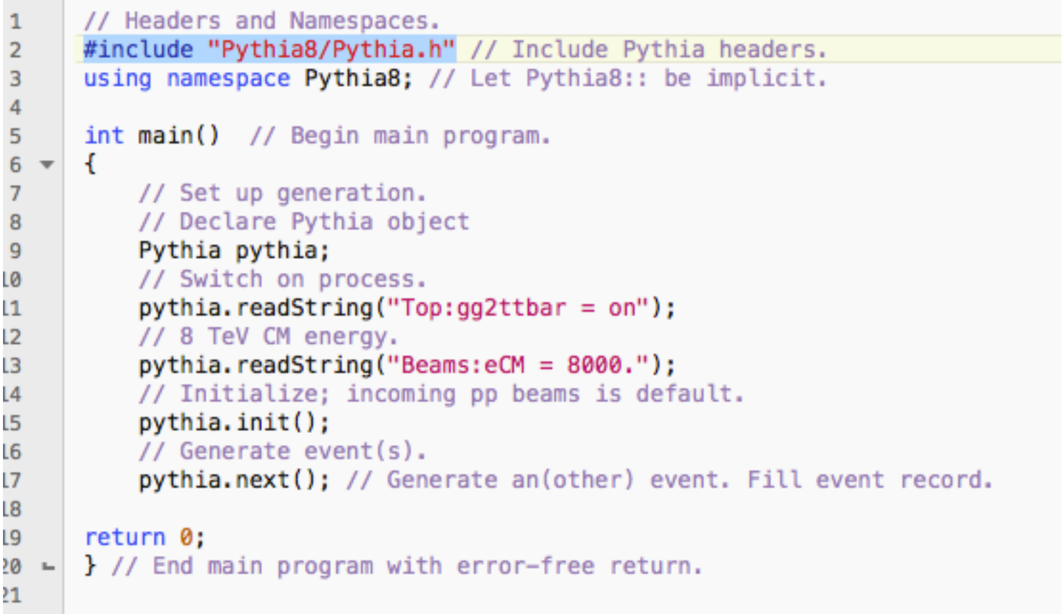
Beauty Project Log Book

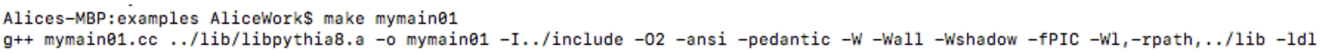
30/01/17 Week 1

Previously I have installed Python, Xcode and a text wrangler IED.
Used the http://home.thep.lu.se/~torbjorn/pdfdoc/worksheet8200.pdf to install Pythia
Completed section 3 “Hello world tutorial” generate a single gg → tt event at the LHC, using Pythia standalone.



```
// Headers and Namespaces.
#include "Pythia8/Pythia.h" // Include Pythia headers.
using namespace Pythia8; // Let Pythia8:: be implicit.

int main()  // Begin main program.
{
    // Set up generation.
    // Declare Pythia object
    Pythia pythia;
    // Switch on process.
    pythia.readString("Top:gg2ttbar = on");
    // 8 TeV CM energy.
    pythia.readString("Beams:eCM = 8000.");
    // Initialize; incoming pp beams is default.
    pythia.init();
    // Generate event(s).
    pythia.next(); // Generate an(other) event. Fill event record.

return 0;
} // End main program with error-free return.

```

Some issues with the include file when using the “make mymain01” command I encounter the error “fatal error: 'Pythia8/Pythia.h' file not found” I tried altering the code to include the header file Pythia8 or Pythia.h but these also had the same error type. I reran the original code but this time it worked.



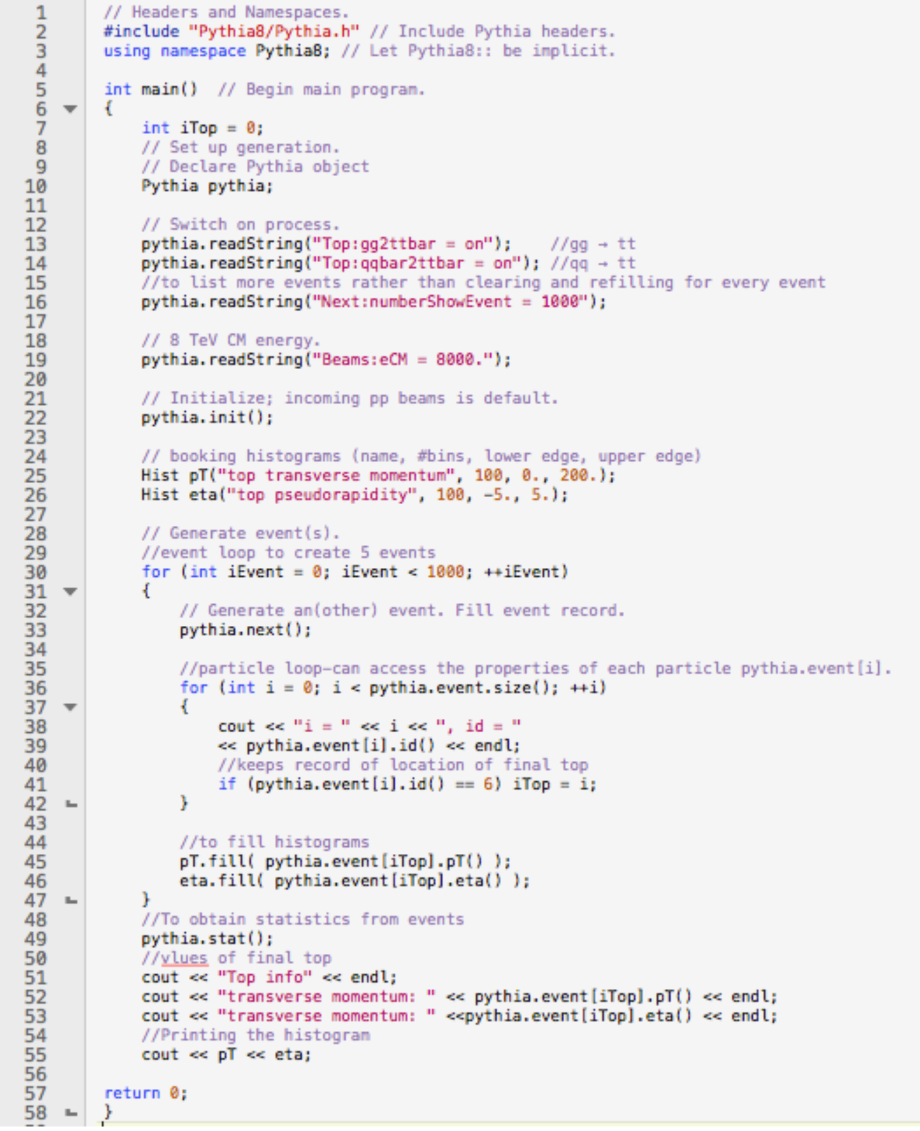
```
Alices-MBP:examples AliceWork$ make mymain01
g++ mymain01.cc ../lib/libpythia8.a -o mymain01 -I../include -O2 -ansi -pedantic -W -Wall -Wshadow -fPIC -Wl,-rpath,../lib -ldl
```

6/02/17 Week 2

I continued with the Pythia work sheet adjusting mymain01.cc to make a more complex simulation.

- Added a qq → tt process
- Used a for loop to generate more than one event known as the Event loop
- obtained statistics on the number of events generated of the different kinds, and the estimated cross sections.
- Used a for loop To access all the particles in the event record known as the Particle loop
- Obtained the location of the last top quark and learned how to find its properties
- Learned how to book, fill and print histograms

The code used to do this is explained more explicitly in the worksheet but my commented code is shown below along with the output.

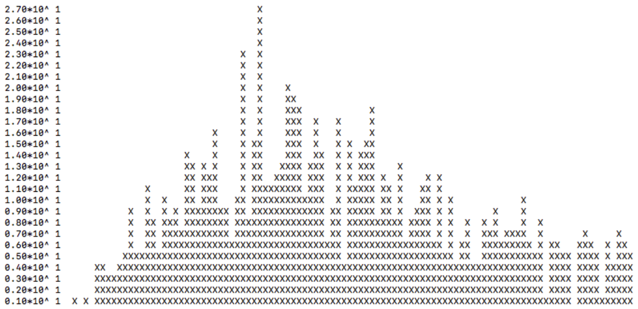


```
// Headers and Namespaces.
#include "Pythia8/Pythia.h" // Include Pythia headers.
using namespace Pythia8; // Let Pythia8:: be implicit.

int main()  // Begin main program.
{
    int iTop = 0;
    // Set up generation.
    // Declare Pythia object
    Pythia pythia;

    // Switch on process.
    pythia.readString("Top:gg2ttbar = on");     //gg → tt
    pythia.readString("Top:qqbar2ttbar = on"); //qq → tt
    //to list more events rather than clearing and refilling for every event
    pythia.readString("Next:numberShowEvent = 1000");

    // 8 TeV CM energy.
    pythia.readString("Beams:eCM = 8000.");

    // Initialize; incoming pp beams is default.
    pythia.init();

    // booking histograms (name, #bins, lower edge, upper edge)
    Hist pT("top transverse momentum", 100, 0., 200.);
    Hist eta("top pseudorapidity", 100, -5., 5.);

    // Generate event(s).
    //event loop to create 5 events
    for (int iEvent = 0; iEvent < 1000; ++iEvent)
    {
        // Generate an(other) event. Fill event record.
        pythia.next();

        //particle loop-can access the properties of each particle pythia.event[i].
        for (int i = 0; i < pythia.event.size(); ++i)
        {
            cout << "i = " << i << ", id = "
            << pythia.event[i].id() << endl;
            //keeps record of location of final top
            if (pythia.event[i].id() == 6) iTop = i;
        }

        //to fill histograms
        pT.fill( pythia.event[iTop].pT() );
        eta.fill( pythia.event[iTop].eta() );
    }
    //To obtain statistics from events
    pythia.stat();
    //vlues of final top
    cout << "Top info" << endl;
    cout << "transverse momentum: " << pythia.event[iTop].pT() << endl;
    cout << "transverse momentum: " <<pythia.event[iTop].eta() << endl;
    //Printing the histogram
    cout << pT << eta;

return 0;
}
```

This returned reams of data, but I will only show the histograms below.

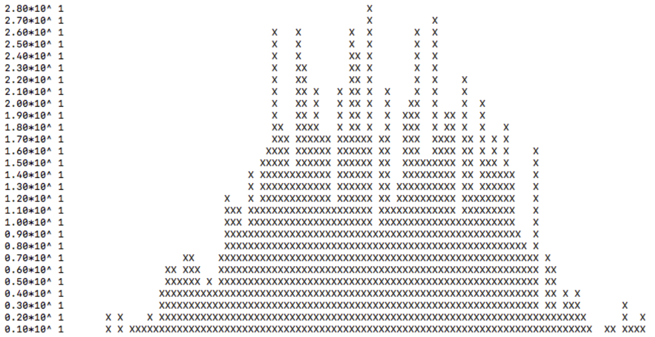
```
Top info
transverse momentum: 1.227e+02
transverse momentum: -3.005e+00
```

```
  2017-02-07 12:22       top transverse momentum
```



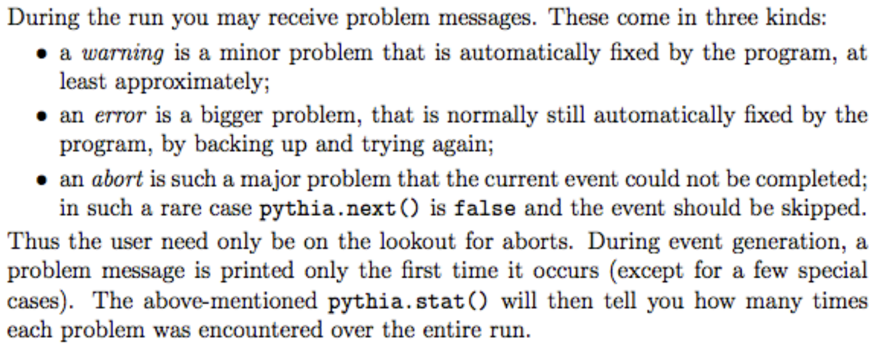
```
       Contents
         *10^ 1  0000000000000100100011011100012012111121111110010111110110100011010100000000000001000000000000000000
         *10^ 0  1010443345955185089743932699603857112309803746974514588218368912625046844879677706581665506765164765
         *10^-1  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
         *10^-2  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

       Low edge
         *10^ 2  0000000000000000000000000000000000000000000000000011111111111111111111111111111111111111111111111111
         *10^ 1  0000011111222223333344444555556666677777888889999900000111112222233333444445555566666777778888899999
         *10^ 0  0246802468024680246802468024680246802468024680246802468024680246802468024680246802468024680246802468

   Entries  =        1000    Mean =  9.5242e+01    Underflow =  0.0000e+00    Low edge  =  0.0000e+00
   All chan =  8.6600e+02    Rms  =  4.7260e+01    Overflow  =  1.3400e+02    High edge =  2.0000e+02
```

```
  2017-02-07 12:22       top pseudorapidity
```



```
       Contents
         *10^ 1  000000000000000000000000011101111211122121112122120121112211211112112111110001000000000000000000
         *10^ 0  00000202011121466577645472119415668746381771176478971139065577990272057484987627624342100110311 2
         *10^-1  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
         *10^-2  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

       Low edge  --------------------------------------------------
         *10^ 0  5444444444433333333332222222222111111111100000000000000000000111111111122222222223333333333444444
         *10^-1  0987654321098765432109876543210987654321012345678901234567890123456789012345678901234567890123456
         *10^-2  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

   Entries  =        1000    Mean =  1.3327e-02    Underflow =  1.0000e+00    Low edge  = -5.0000e+00
   All chan =  9.9800e+02    Rms  =  1.7378e+00    Overflow  =  1.0000e+00    High edge =  5.0000e+00
```

This section also discussed errors that may occur when using Pythia.



During the run you may receive problem messages. These come in three kinds:

- a *warning* is a minor problem that is automatically fixed by the program, at least approximately;
- an *error* is a bigger problem, that is normally still automatically fixed by the program, by backing up and trying again;
- an *abort* is such a major problem that the current event could not be completed; in such a rare case `pythia.next()` is `false` and the event should be skipped.

Thus the user need only be on the lookout for aborts. During event generation, a problem message is printed only the first time it occurs (except for a few special cases). The above-mentioned `pythia.stat()` will then tell you how many times each problem was encountered over the entire run.

Meeting with lily 10/2/17

Lily gave me a paper to read from one of Fabs previous students, we discussed the work I have already done, lily was happy with first Pythia histograms, also liked the log book, from now on I will upload it on github to.

We discussed the plan for the next week or so, the main aims are:

1. Edit main programme to generate HiggsSM:gg2H
   http://home.thep.lu.se/~torbjorn/pythia81html/Welcome.html
2. Try to find the process needed to make Z->bb in Pythia8. Refer to
   https://www.hep.ucl.ac.uk/theses/luke_lambourne_thesis.pdf
   http://www.nikhef.nl/pub/services/biblio/theses_pdf/thesis_N_Naumann.pdf
3. Try to print out only bs from Higgs.

We have decided to make my project different to the others by comparing higgs bb decays to Z bb decays to do this I need to write a program that generates higgs particles, and find the Pythia function that makes Z particles and also include this in the code. Then we want to alter the code to print only the bbs decays.

(Look at Andy's code use if statements for if there is a b particle and if its mother particle is a higgs etc.) https://github.com/SussexMPhys2016/go/issues/27

Next Friday we will also aim to meet and create our first graph.

13/2/17 Week 3

The main process for the creation of Z bosons seems to be pp -> Z -> bb, although I am yet to find the Pythia command that creates this however another MC generator 'MCFM NLO/LO' is mentioned a lot as well as 'Herwig' I also found the following useful paper and Pythia page

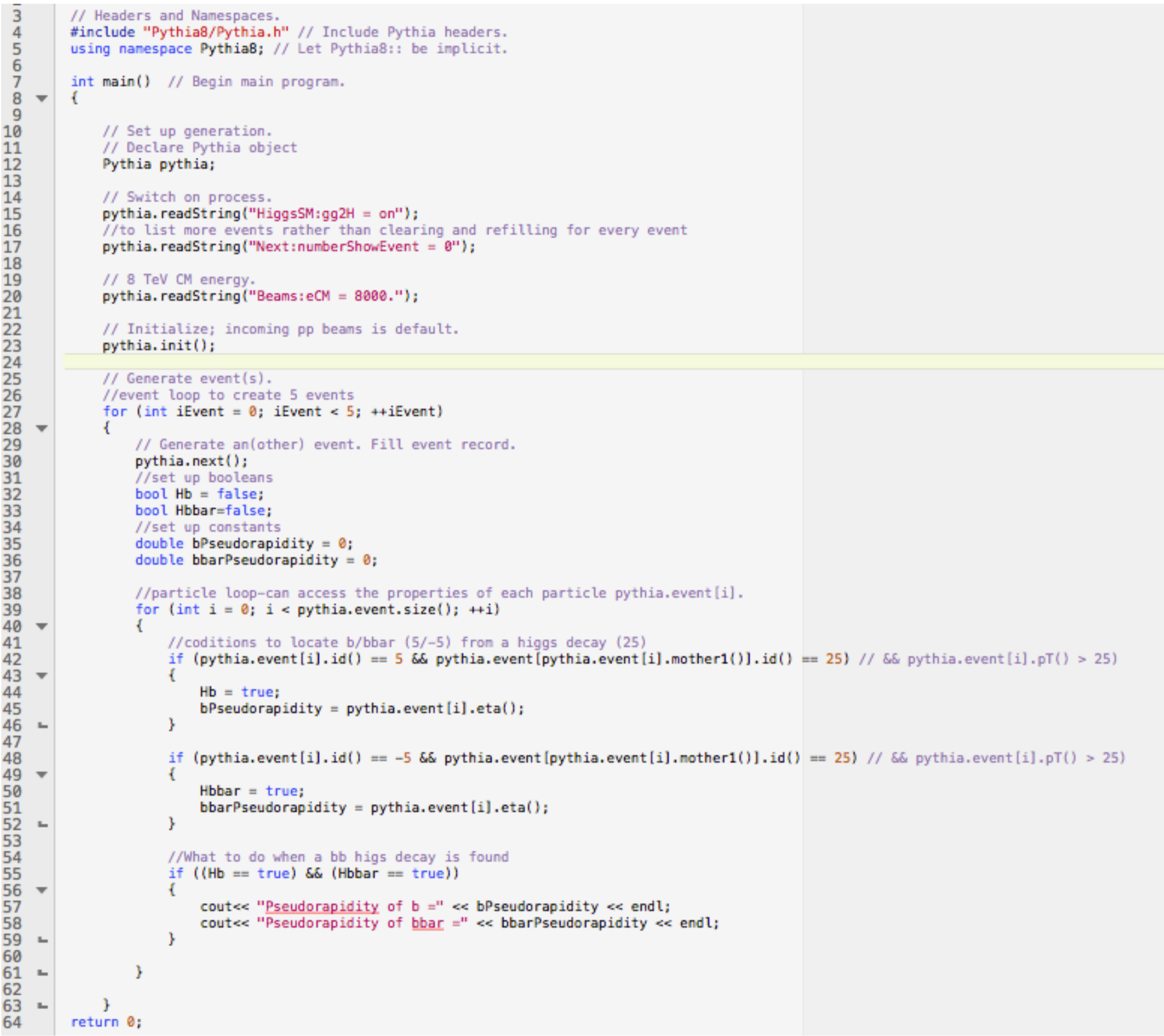
http://www.nikhef.nl/pub/services/biblio/theses_pdf/thesis_N_Naumann.pdf
https://hep.pa.msu.edu/wiki/bin/view/ZPrime/E6Generation
http://home.thep.lu.se/~torbjorn/pythia81html/Welcome.html

I have started writing my own code to generate the creation and the decay of the Higgs particle I am doing this using gg -> H using ‘pythia.readString("HiggsSM:gg2H = on");’ in my event loop I then created two Boolean variables Hb and Hbbar, I then set if statements in my particle loop, setting Hb and Hbbar to true if b/bbar is detected (event id 5/-5) from a Higgs decay (event id 25).

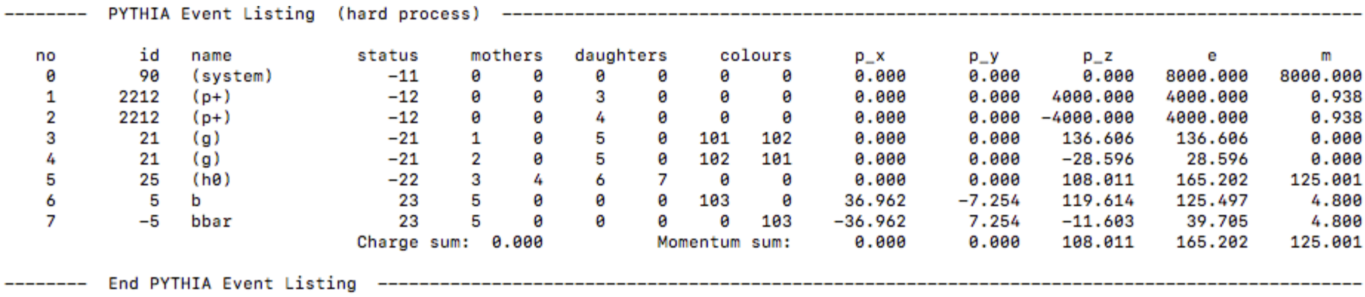
In Andy’s code he also has the condition that pythia.event[i].pT() > 25 but I am unsure as to why, I will ask Lily.

There is also an if condition that if both Hb and Hbbar are true, i.e. a b and a bbar pair are formed from the decay of a Higgs then it should print something out, at the moment I have it printing out the pseudo rapidity but this is just a filler until I know what information I will need to print. The code and output are shown below.

Although the code compiles fine I am unsure of whether it is doing the desired task, I’m still unsure of what I am looking at/for again I will ask Lily.

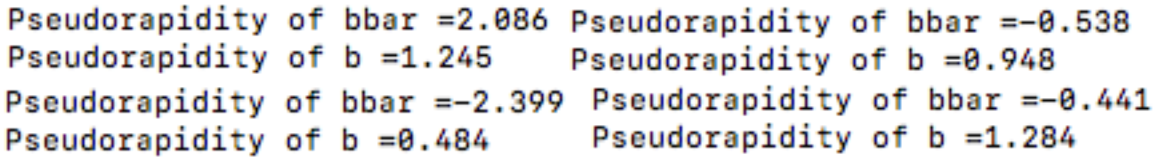


```
// Headers and Namespaces.
#include "Pythia8/Pythia.h" // Include Pythia headers.
using namespace Pythia8; // Let Pythia8:: be implicit.

int main()  // Begin main program.
{

    // Set up generation.
    // Declare Pythia object
    Pythia pythia;

    // Switch on process.
    pythia.readString("HiggsSM:gg2H = on");
    //to list more events rather than clearing and refilling for every event
    pythia.readString("Next:numberShowEvent = 0");

    // 8 TeV CM energy.
    pythia.readString("Beams:eCM = 8000.");

    // Initialize; incoming pp beams is default.
    pythia.init();

    // Generate event(s).
    //event loop to create 5 events
    for (int iEvent = 0; iEvent < 5; ++iEvent)
    {
        // Generate an(other) event. Fill event record.
        pythia.next();
        //set up booleans
        bool Hb = false;
        bool Hbbar=false;
        //set up constants
        double bPseudorapidity = 0;
        double bbarPseudorapidity = 0;

        //particle loop-can access the properties of each particle pythia.event[i].
        for (int i = 0; i < pythia.event.size(); ++i)
        {
            //coditions to locate b/bbar (5/-5) from a higgs decay (25)
            if (pythia.event[i].id() == 5 && pythia.event[pythia.event[i].mother1()].id() == 25) // && pythia.event[i].pT() > 25)
            {
                Hb = true;
                bPseudorapidity = pythia.event[i].eta();
            }

            if (pythia.event[i].id() == -5 && pythia.event[pythia.event[i].mother1()].id() == 25) // && pythia.event[i].pT() > 25)
            {
                Hbbar = true;
                bbarPseudorapidity = pythia.event[i].eta();
            }

            //What to do when a bb higs decay is found
            if ((Hb == true) && (Hbbar == true))
            {
                cout<< "Pseudorapidity of b =" << bPseudorapidity << endl;
                cout<< "Pseudorapidity of bbar =" << bbarPseudorapidity << endl;
            }

        }

    }
return 0;
```



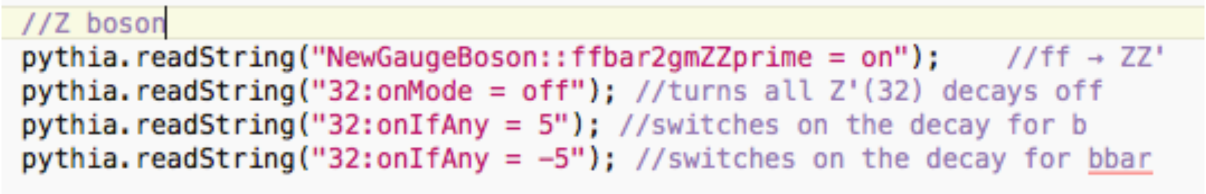
```
 --------  PYTHIA Event Listing  (hard process)  -----------------------------------------------------------------------------------

    no         id   name            status     mothers   daughters     colours      p_x        p_y        p_z         e          m
     0         90   (system)           -11     0     0     0     0     0     0      0.000      0.000      0.000   8000.000   8000.000
     1       2212   (p+)               -12     0     0     3     0     0     0      0.000      0.000   4000.000   4000.000      0.938
     2       2212   (p+)               -12     0     0     4     0     0     0      0.000      0.000  -4000.000   4000.000      0.938
     3         21   (g)                -21     1     0     5     0   101   102      0.000      0.000    136.606    136.606      0.000
     4         21   (g)                -21     2     0     5     0   102   101      0.000      0.000    -28.596     28.596      0.000
     5         25   (h0)               -22     3     4     6     7     0     0      0.000      0.000    108.011    165.202    125.001
     6          5   b                   23     5     0     0     0   103     0     36.962     -7.254    119.614    125.497      4.800
     7         -5   bbar                23     5     0     0     0     0   103    -36.962      7.254    -11.603     39.705      4.800
                                   Charge sum:  0.000           Momentum sum:      0.000      0.000    108.011    165.202    125.001

 --------  End PYTHIA Event Listing  -----------------------------------------------------------------------------------------------
```

The main results I got for this are shown below and were each repeated over 100 times.



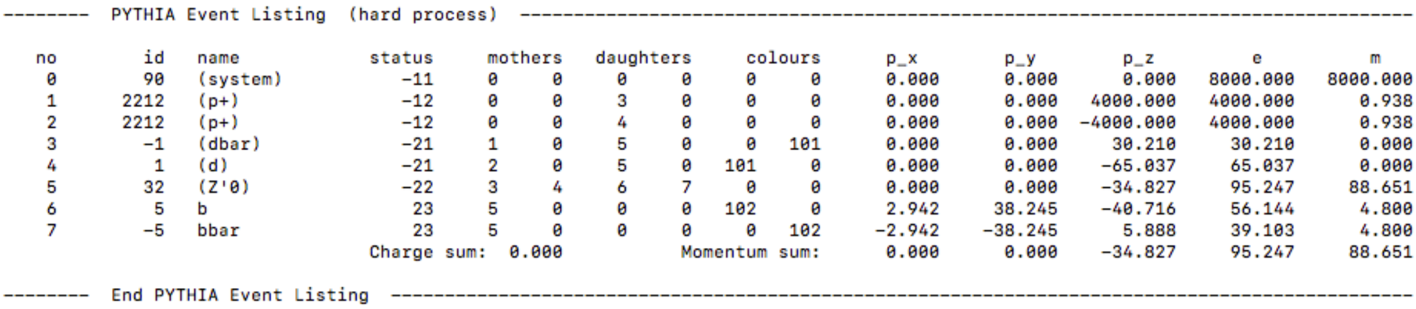
```
Pseudorapidity of bbar =2.086  Pseudorapidity of bbar =-0.538
Pseudorapidity of b =1.245     Pseudorapidity of b =0.948
Pseudorapidity of bbar =-2.399  Pseudorapidity of bbar =-0.441
Pseudorapidity of b =0.484      Pseudorapidity of b =1.284
```

I then attempted to turn on the creation of Z bosons, I did this using instructions from the website https://hep.pa.msu.edu/wiki/bin/view/ZPrime/E6Generation and wrote the following code.



```
//Z boson
pythia.readString("NewGaugeBoson::ffbar2gmZZprime = on");    //ff → ZZ'
pythia.readString("32:onMode = off"); //turns all Z'(32) decays off
pythia.readString("32:onIfAny = 5"); //switches on the decay for b
pythia.readString("32:onIfAny = -5"); //switches on the decay for bbar
```

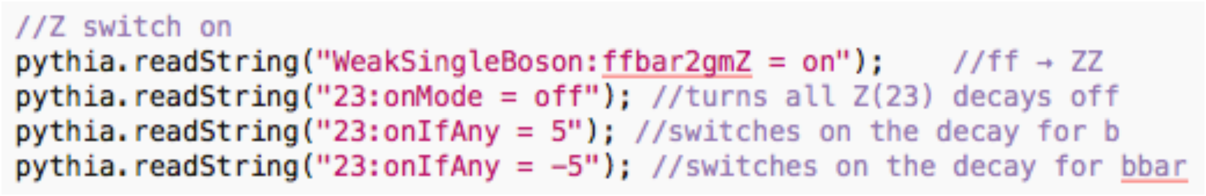
This creates Z' and produces the following output but no longer seems to produce Higgs particles.



```
 --------  PYTHIA Event Listing  (hard process)  -----------------------------------------------------------------------------------

    no         id   name            status     mothers   daughters     colours      p_x        p_y        p_z         e          m
     0         90   (system)           -11     0     0     0     0     0     0      0.000      0.000      0.000   8000.000   8000.000
     1       2212   (p+)               -12     0     0     3     0     0     0      0.000      0.000   4000.000   4000.000      0.938
     2       2212   (p+)               -12     0     0     4     0     0     0      0.000      0.000  -4000.000   4000.000      0.938
     3         -1   (dbar)             -21     1     0     5     0     0   101      0.000      0.000     30.210     30.210      0.000
     4          1   (d)                -21     2     0     5     0   101     0      0.000      0.000    -65.037     65.037      0.000
     5         32   (Z'0)              -22     3     4     6     7     0     0      0.000      0.000    -34.827     95.247     88.651
     6          5   b                   23     5     0     0     0   102     0      2.942     38.245    -40.716     56.144      4.800
     7         -5   bbar                23     5     0     0     0     0   102     -2.942    -38.245      5.888     39.103      4.800
                                   Charge sum:  0.000           Momentum sum:      0.000      0.000    -34.827     95.247     88.651

 --------  End PYTHIA Event Listing  -----------------------------------------------------------------------------------------------
```

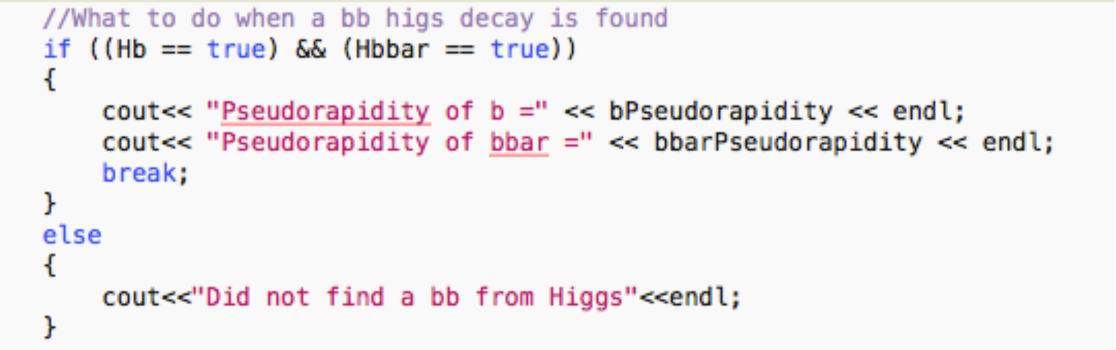
Meeting with lily 17/2/17

Lily looked at my code and we worked through a few of the issues I had, firstly I realised that I had created the wrong particle, Z' is a theoretical particle, where we want to study the Z decay so we altered the string turning on Z and the ones turning the Z decays off (as Z' is 32 and Z is 23) The new code can be seen bellow.



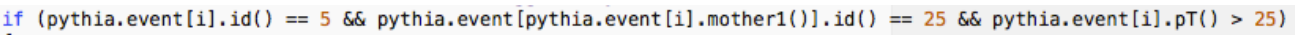
```
//Z switch on
pythia.readString("WeakSingleBoson:ffbar2gmZ = on");    //ff → ZZ
pythia.readString("23:onMode = off"); //turns all Z(23) decays off
pythia.readString("23:onIfAny = 5"); //switches on the decay for b
pythia.readString("23:onIfAny = -5"); //switches on the decay for bbar
```

Lily then mentioned that I should do the H and Z creation and decay in different files otherwise I will just override one with the other. We also fixed the issue I had with the pseudo rapidity being printed out hundreds of times, as once a particle was found that set the Hb and Hbbar to true they stayed like that and were true for all of the next values so I had to insert a break statement, I also inserted an else statement to print when a bb decay was not found.



```
//What to do when a bb higs decay is found
if ((Hb == true) && (Hbbar == true))
{
    cout<< "Pseudorapidity of b =" << bPseudorapidity << endl;
    cout<< "Pseudorapidity of bbar =" << bbarPseudorapidity << endl;
    break;
}
else
{
    cout<<"Did not find a bb from Higgs"<<endl;
}
```

Lily also recommended I include an extra condition in my if statements about the pT of the event at low levels these is a lot of background noise so we also want to exclude particles with a pT of below 25.



```
if (pythia.event[i].id() == 5 && pythia.event[pythia.event[i].mother1()].id() == 25 && pythia.event[i].pT() > 25)
```

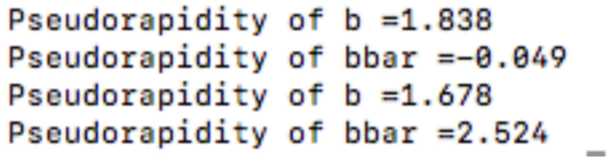
We also then discussed my next steps in the project:

- Comparing Higgs with WeakSingleBoson:ffbar2gmZ
- Two programmes: one for H and one for Z
- What fraction of your H->bb and Z->bb events pass the pT cuts?
- What fraction of your H and Z events decay to bb?
- Print the following Variables for the bs from the Higgs/Z:
    1. eta()
    2. phi()
    3. pT()
    4. rap()
    5. px()
    6. py()
    7. pz()
    8. e()

- Print the mass for the Higgs and the Z themselves using m()
- Complete section 7 of the hello world worksheet http://home.thep.lu.se/~torbjorn/pythia81html/Welcome.html

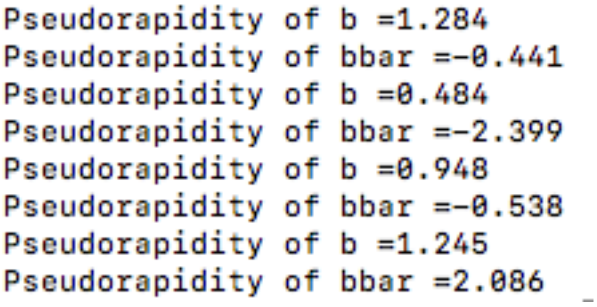
Week 4 20/2/16

Created two separate files for Higgs and Z decays, these only configure correctly when named my main otherwise they have an issue with header files, so the Higgs is in the file mymain02.cc and the Z decay is in file mymain03.cc. However, they produce less results than expected, for the Z decay I ran the simulation 5 times however it only ever produces two bb decays but all other decay types are turned off so this makes no sense. Also, running the program several times leads to the same results each time which are shown below.



```
Pseudorapidity of b =1.838
Pseudorapidity of bbar =-0.049
Pseudorapidity of b =1.678
Pseudorapidity of bbar =2.524
```

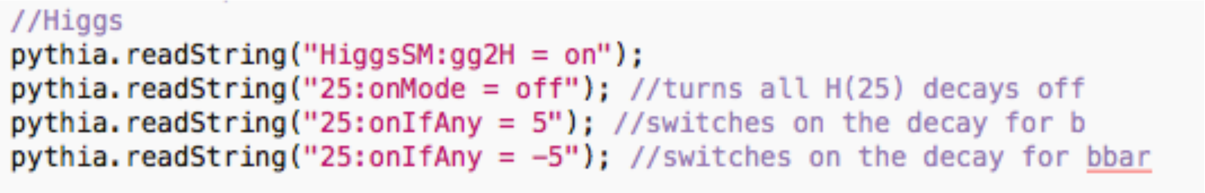
The same is true for the Higgs case only 4 decays are shown for 5 runs, although the Higgs isn't forced to decay to bb by the code we would only expect it to decay to this 56% of the time due to the BR whereas this gives a 80% BR. Each time I run the code I also get the same output. The information printed out every time a bb decay is found for a Higgs is shown below.



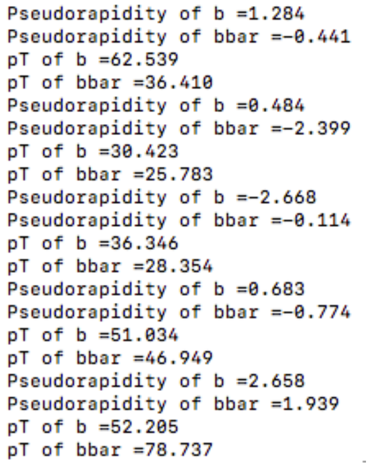
```
Pseudorapidity of b =1.284
Pseudorapidity of bbar =-0.441
Pseudorapidity of b =0.484
Pseudorapidity of bbar =-2.399
Pseudorapidity of b =0.948
Pseudorapidity of bbar =-0.538
Pseudorapidity of b =1.245
Pseudorapidity of bbar =2.086
```

If I alter the code for the Z if statement and get rid of the pT cuts then I get 5 bb decays for 5 runs as expected. However, this does not work for the Higgs decay suggesting that one of the Higgs just doesn't decay into bb. I am still unsure why I get the same values each time. I tried using pT of the particles instead of Pseudo rapidity but I still generate the same 4 values each time. If I run the program with 6 runs I then get 5 values.

I then included the code bellow to force the Higgs to decay via bb and found 5 results as expected, these are also shown below.

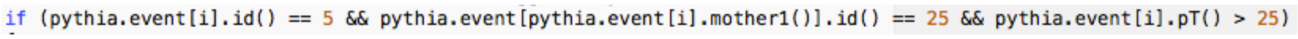


```
//Higgs
pythia.readString("HiggsSM:gg2H = on");
pythia.readString("25:onMode = off"); //turns all H(25) decays off
pythia.readString("25:onIfAny = 5"); //switches on the decay for b
pythia.readString("25:onIfAny = -5"); //switches on the decay for bbar
```

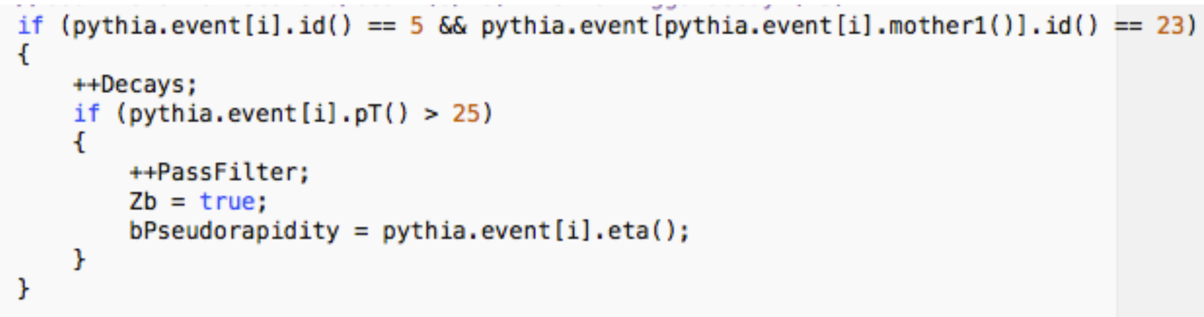


```
Pseudorapidity of b =1.284
Pseudorapidity of bbar =-0.441
pT of b =62.539
pT of bbar =36.410
Pseudorapidity of b =0.484
Pseudorapidity of bbar =-2.399
pT of b =30.423
pT of bbar =25.783
Pseudorapidity of b =-2.668
Pseudorapidity of bbar =-0.114
pT of b =36.346
pT of bbar =28.354
Pseudorapidity of b =0.683
Pseudorapidity of bbar =-0.774
pT of b =51.034
pT of bbar =46.949
Pseudorapidity of b =2.658
Pseudorapidity of bbar =1.939
pT of b =52.205
pT of bbar =78.737
```

I started the task of recording the fraction of bb decays that make it through the filter, this was done by splitting the pT pass into a separate if statement you can see the before and after below.

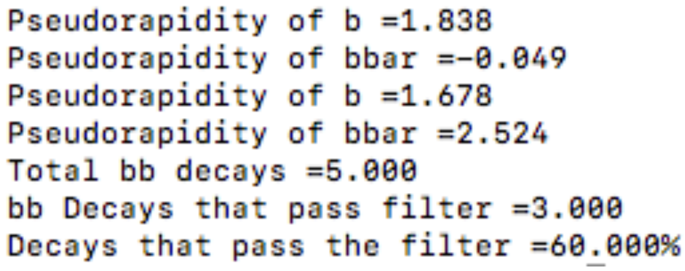


```
if (pythia.event[i].id() == 5 && pythia.event[pythia.event[i].mother1()].id() == 25 && pythia.event[i].pT() > 25)
```



```
if (pythia.event[i].id() == 5 && pythia.event[pythia.event[i].mother1()].id() == 23)
{
    ++Decays;
    if (pythia.event[i].pT() > 25)
    {
        ++PassFilter;
        Zb = true;
        bPseudorapidity = pythia.event[i].eta();
    }
}
```

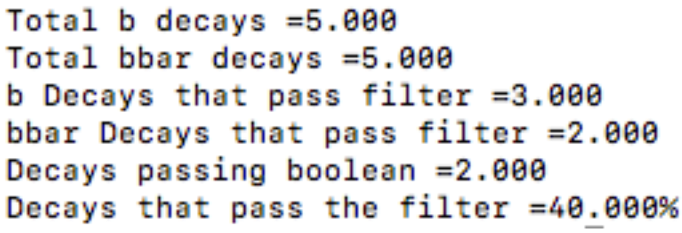
The doubles ‘Decays’ and ‘PassFilter’ are used to keep a tally of the number of bb decays and the decays that get through the pT cut. This produces the following output of Z.



```
Pseudorapidity of b =1.838
Pseudorapidity of bbar =-0.049
Pseudorapidity of b =1.678
Pseudorapidity of bbar =2.524
Total bb decays =5.000
bb Decays that pass filter =3.000
Decays that pass the filter =60.000%
```

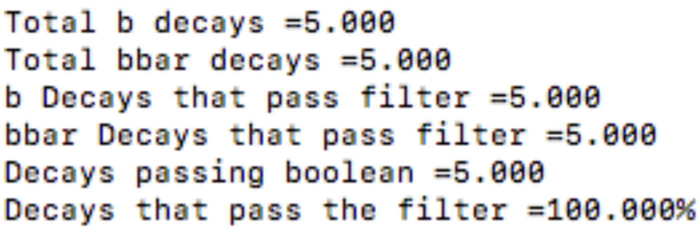
This is not what I expected for two reasons, firstly we previously always got two decays that passed the pT filter and now we apparently have 3 and secondly we only have information for 2 out of these 3 decays. I have only used the ++PassFilter command on the b particles as I only wanted to count each decay once and I assumed as I have told the generator the Z particles can only decay via bb that the generation of a b particle that passed the pT would mean a bbar particle that also passed the pT cut ~~as the cut is for the pT of the mother particle.~~ However, it seems that I am creating a b particle that somehow passes the cut and adds to the PassFilter tally but then the corresponding bbar particle doesn’t pass the cut so it does not qualify the Boolean if statement and print information about it. ~~But this makes no sense as both particles share that same mother, so both should either make the cut or not.~~ The cut does not depend on the mother

If I keep a separate tally of everything I produce the following



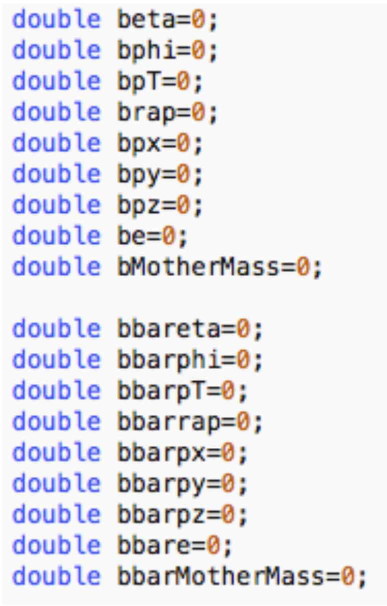
```
Total b decays =5.000
Total bbar decays =5.000
b Decays that pass filter =3.000
bbar Decays that pass filter =2.000
Decays passing boolean =2.000
Decays that pass the filter =40.000%
```

For the Higgs decays, I get as expected all of my decays pass the filters and decay as expected to bb pairs. This is shown in the code output bellow

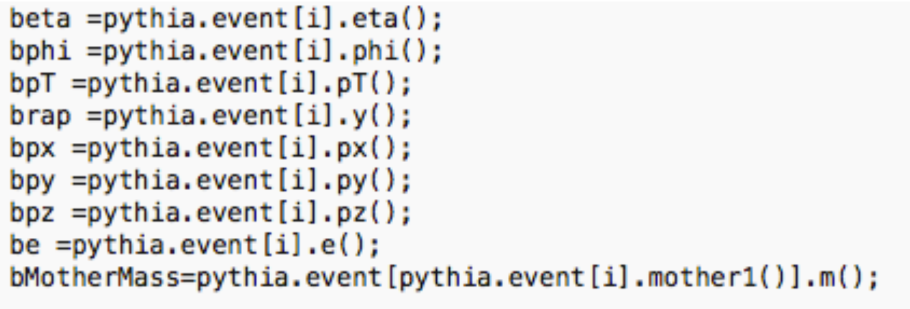


```
Total b decays =5.000
Total bbar decays =5.000
b Decays that pass filter =5.000
bbar Decays that pass filter =5.000
Decays passing boolean =5.000
Decays that pass the filter =100.000%
```

I will ask lily about this. I have also initiated all of the variables we wanted to know and made them print using the following code.



```
double beta=0;
double bphi=0;
double bpT=0;
double brap=0;
double bpx=0;
double bpy=0;
double bpz=0;
double be=0;
double bMotherMass=0;

double bbareta=0;
double bbarphi=0;
double bbarpT=0;
double bbarrap=0;
double bbarpx=0;
double bbarpy=0;
double bbarpz=0;
double bbare=0;
double bbarMotherMass=0;
```

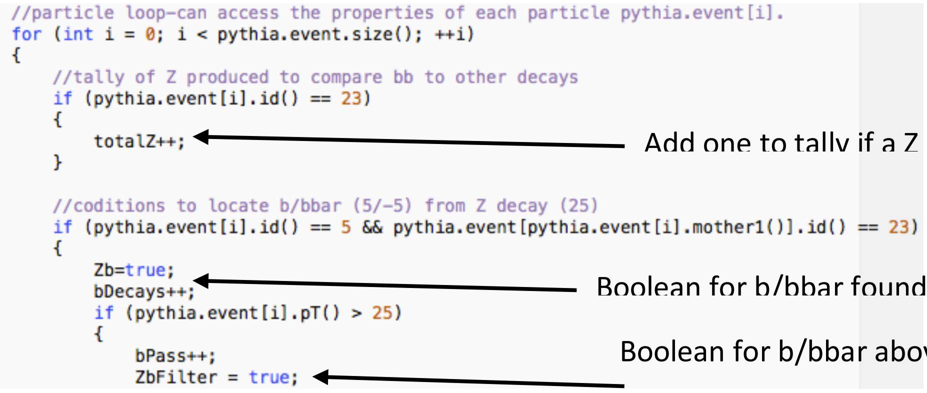


```
beta =pythia.event[i].eta();
bphi =pythia.event[i].phi();
bpT =pythia.event[i].pT();
brap =pythia.event[i].y();
bpx =pythia.event[i].px();
bpy =pythia.event[i].py();
bpz =pythia.event[i].pz();
be =pythia.event[i].e();
bMotherMass=pythia.event[pythia.event[i].mother1()].m();
```

I also attempted to write the code to calculate how many of my decays happen via bb, I have done this by inserting a if statement in the particle loop that if a particle is a Z/H to add one to the Z/H tally and then at the end I compare the total numbers to the numbers that pass the cut. At the moment, I am forcing the Z and Higgs to decay via bb so these ratios should be close to 100% minus the few that didn't make the cut, however they aren't I think this is because Pythia stores every interaction of the H/Z as a new particle when it is not, I will have to look into how to only count the creation of the particle.

I wanted to compare the number of H/Z created to the total number that decay to bb, not just the bb decays that pass the filter. To do this I created a second set of Boolean variables that were true if a b or bbar were found with a mother particle of Z/H depending on what we were looking for.

The code where the Booleans are set true, where they are used in a condition and the output are seen below.



```
//particle loop-can access the properties of each particle pythia.event[i].
for (int i = 0; i < pythia.event.size(); ++i)
{
    //tally of Z produced to compare bb to other decays
    if (pythia.event[i].id() == 23)
    {
        totalZ++;
    }

    //coditions to locate b/bbar (5/-5) from Z decay (25)
    if (pythia.event[i].id() == 5 && pythia.event[pythia.event[i].mother1()].id() == 23)
    {
        Zb=true;
        bDecays++;
        if (pythia.event[i].pT() > 25)
        {
            bPass++;
            ZbFilter = true;
```

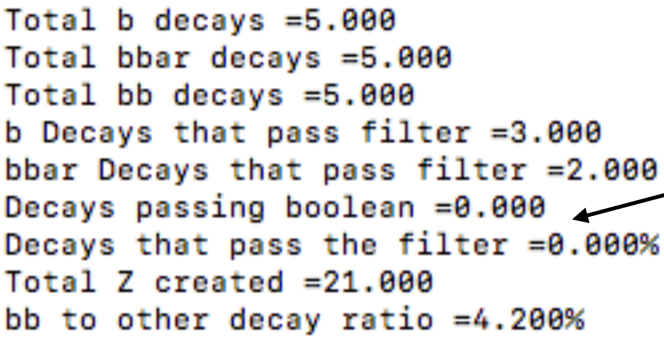
Add one to tally if a Z is found

Boolean for b/bbar found

Boolean for b/bbar above filter threshold

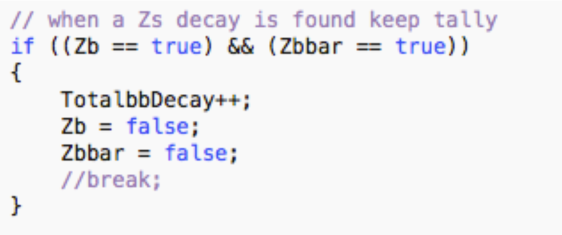
6/3/2017 Week 6 (ill week 5)

This week I had to rewrite some of the z code, I have previously tried to use two Booleans to form different clauses of when a Z bb decay was present and when one was present that passes the filter. Previously we had used the break function but this breaks completely out of the loop so If I found a decay that satisfied the first Boolean it would exit the loop before I could see if it satisfied the second condition. However, if I removed the break command once the Boolean variables were set as true they measured every event as a decay so I decided to reset the Boolean values to false at the end of my Boolean condition section so that they would continue to search all of the particles and not break out of the loops but would also not continue to report everything as a bb decay.

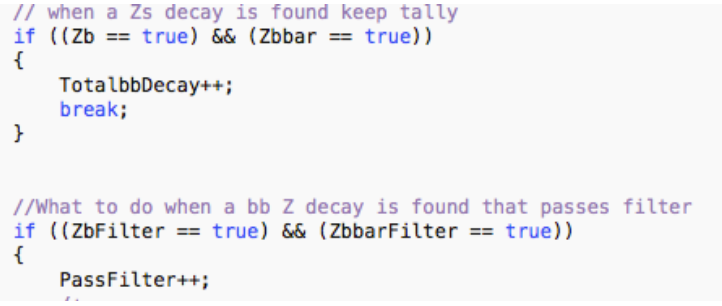


```
Total b decays =5.000
Total bbar decays =5.000
Total bb decays =5.000
b Decays that pass filter =3.000
bbar Decays that pass filter =2.000
Decays passing boolean =0.000
Decays that pass the filter =0.000%
Total Z created =21.000
bb to other decay ratio =4.200%
```

These should not be zero, something is wrong with the code, Possibly the Booleans creating this, before the second Boolean was added this worked fine.



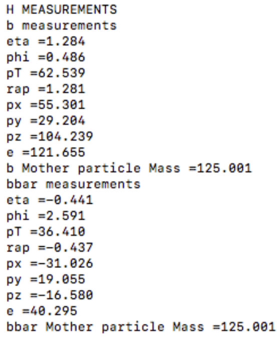
```
// when a Zs decay is found keep tally
if ((Zb == true) && (Zbbar == true))
{
    TotalbbDecay++;
    Zb = false;
    Zbbar = false;
    //break;
}
```



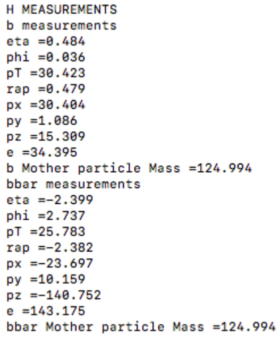
```
// when a Zs decay is found keep tally
if ((Zb == true) && (Zbbar == true))
{
    TotalbbDecay++;
    break;
}


//What to do when a bb Z decay is found that passes filter
if ((ZbFilter == true) && (ZbbarFilter == true))
{
    PassFilter++;
```

Now I have both of the Booleans working I can start to work out branching ratios once I have sorted out how to stop Pythia from saving every interaction of H/Z as a new particle.

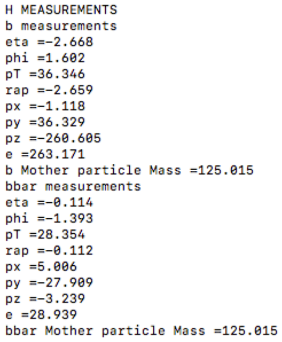
I wanted to start comparing and contrasting the data I have for the Z and H bb decays as I have already got my code to print all of the information for it. So, the following are readouts for both Z and H decays. The most obvious difference so far is in pT.



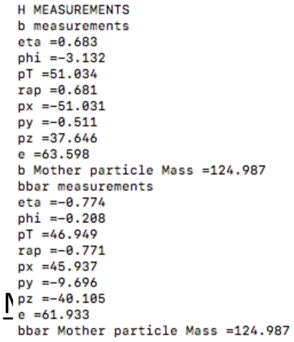
```
H MEASUREMENTS
b measurements
eta =1.284
phi =0.486
pT =62.539
rap =1.281
px =55.301
py =29.204
pz =104.239
e =121.655
b Mother particle Mass =125.001
bbar measurements
eta =-0.441
phi =2.591
pT =36.410
rap =-0.437
px =-31.026
py =19.055
pz =-16.580
e =40.295
bbar Mother particle Mass =125.001
```



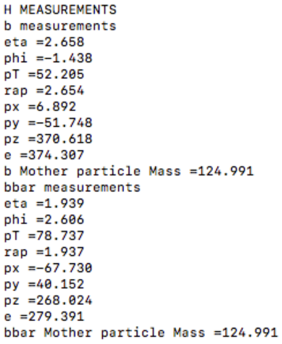
```
H MEASUREMENTS
b measurements
eta =0.484
phi =0.036
pT =30.423
rap =0.479
px =30.404
py =1.086
pz =15.309
e =34.395
b Mother particle Mass =124.994
bbar measurements
eta =-2.399
phi =2.737
pT =25.783
rap =-2.382
px =-23.697
py =10.159
pz =-140.752
e =143.175
bbar Mother particle Mass =124.994
```



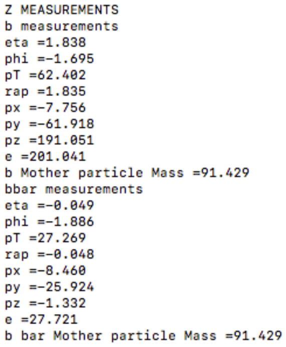
```
H MEASUREMENTS
b measurements
eta =-2.668
phi =1.602
pT =36.346
rap =-2.659
px =-1.118
py =36.329
pz =-260.605
e =263.171
b Mother particle Mass =125.015
bbar measurements
eta =-0.114
phi =-1.393
pT =28.354
rap =-0.112
px =5.006
py =-27.909
pz =-3.239
e =28.939
bbar Mother particle Mass =125.015
```



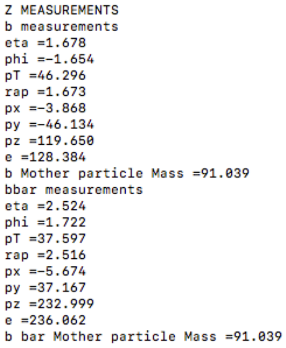
```
H MEASUREMENTS
b measurements
eta =0.683
phi =-3.132
pT =51.034
rap =0.681
px =-51.031
py =-0.511
pz =37.646
e =63.598
b Mother particle Mass =124.987
bbar measurements
eta =-0.774
phi =-0.208
pT =46.949
rap =-0.771
px =45.937
py =-9.696
pz =-40.105
e =61.933
bbar Mother particle Mass =124.987
```



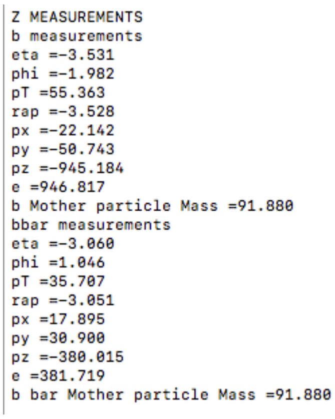
```
H MEASUREMENTS
b measurements
eta =2.658
phi =-1.438
pT =52.205
rap =2.654
px =6.892
py =-51.748
pz =370.618
e =374.307
b Mother particle Mass =124.991
bbar measurements
eta =1.939
phi =2.606
pT =78.737
rap =1.937
px =-67.730
py =40.152
pz =268.024
e =279.391
bbar Mother particle Mass =124.991
```



```
Z MEASUREMENTS
b measurements
eta =1.838
phi =-1.695
pT =62.402
rap =1.835
px =-7.756
py =-61.918
pz =191.051
e =201.041
b Mother particle Mass =91.429
bbar measurements
eta =-0.049
phi =-1.886
pT =27.269
rap =-0.048
px =-8.460
py =-25.924
pz =-1.332
e =27.721
b bar Mother particle Mass =91.429
```



```
Z MEASUREMENTS
b measurements
eta =1.678
phi =-1.654
pT =46.296
rap =1.673
px =-3.868
py =-46.134
pz =119.650
e =128.384
b Mother particle Mass =91.039
bbar measurements
eta =2.524
phi =1.722
pT =37.597
rap =2.516
px =-5.674
py =37.167
pz =232.999
e =236.062
b bar Mother particle Mass =91.039
```



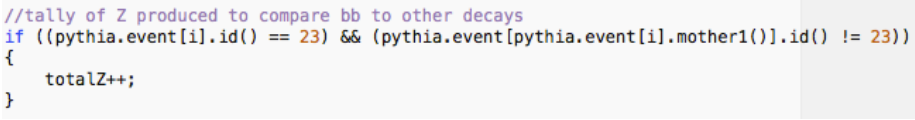
```
Z MEASUREMENTS
b measurements
eta =-3.531
phi =-1.982
pT =55.363
rap =-3.528
px =-22.142
py =-50.743
pz =-945.184
e =946.817
b Mother particle Mass =91.880
bbar measurements
eta =-3.060
phi =1.046
pT =35.707
rap =-3.051
px =17.895
py =30.900
pz =-380.015
e =381.719
b bar Mother particle Mass =91.880
```

[illegible] 7

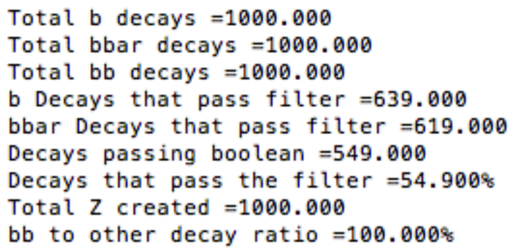
- How to stop it recording every interaction as a new H/Z particle – include not statement to get rid of z daughters
- Why much fewer bbar decays pass the filter than b – statistically unimportant as event number increases, sqrt n uncertainties, simple statistic test.
- How to get the histograms to work? Try again and see Lily Monday 4pm

- Look at Boolean at 1000.
- Write an outline plan for Dissertation

I have now got rid of the copy particles, I did this by excluding particles with the mother particle the same as themselves, the works and gives the output below.



```
//tally of Z produced to compare bb to other decays
if ((pythia.event[i].id() == 23) && (pythia.event[pythia.event[i].mother1()].id() != 23))
{
    totalZ++;
}
```

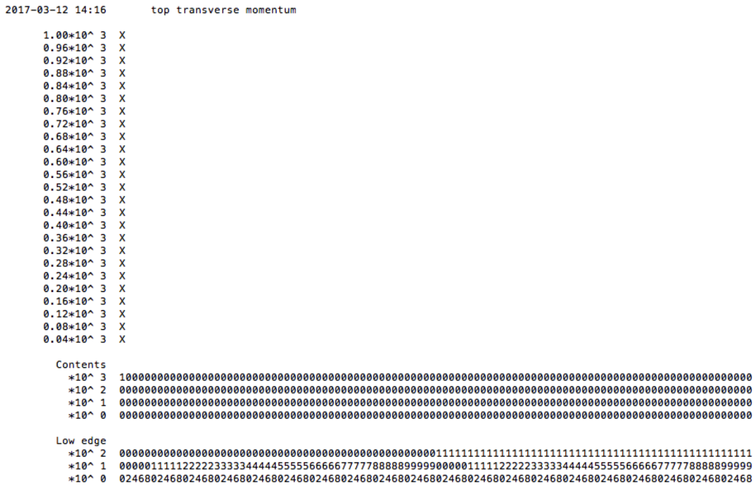


```
Total b decays =1000.000
Total bbar decays =1000.000
Total bb decays =1000.000
b Decays that pass filter =639.000
bbar Decays that pass filter =619.000
Decays passing boolean =549.000
Decays that pass the filter =54.900%
Total Z created =1000.000
bb to other decay ratio =100.000%
```

This is the sort of output I was expecting a 1000 trials creates a 1000 Z/H particles and 100% of them decay as bb pair as I am currently forcing them too.

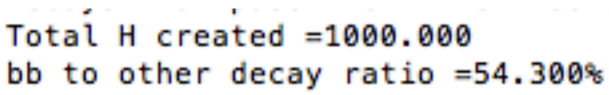
Although I can't quite explain the A-symmetry in the bb decays of the Z particle it becomes statistically insignificant when using the square root test at large run values, also I previously feared my Boolean wasn't working as it did not match the minimum value of the b/bbar cut pass but I've realised some of the b/bbars that didn't make the cut may be from the same pair but some may not so this makes sense.

I still can't seem to get the graphs to work so I will address this with Lily on our next meeting.

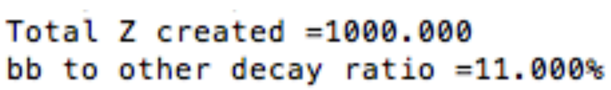


```
2017-03-12 14:16       top transverse momentum

      1.00*10^ 3  X
      0.96*10^ 3  X
      0.92*10^ 3  X
      0.88*10^ 3  X
      0.84*10^ 3  X
      0.80*10^ 3  X
      0.76*10^ 3  X
      0.72*10^ 3  X
      0.68*10^ 3  X
      0.64*10^ 3  X
      0.60*10^ 3  X
      0.56*10^ 3  X
      0.52*10^ 3  X
      0.48*10^ 3  X
      0.44*10^ 3  X
      0.40*10^ 3  X
      0.36*10^ 3  X
      0.32*10^ 3  X
      0.28*10^ 3  X
      0.24*10^ 3  X
      0.20*10^ 3  X
      0.16*10^ 3  X
      0.12*10^ 3  X
      0.08*10^ 3  X
      0.04*10^ 3  X

       Contents
         *10^ 3  1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
         *10^ 2  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
         *10^ 1  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
         *10^ 0  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

       Low edge
         *10^ 2  0000000000000000000000000000000000000000000000000011111111111111111111111111111111111111111111111111
         *10^ 1  0000011111222223333344444555556666677777888889999900000111112222233333444445555566666777778888899999
         *10^ 0  0246802468024680246802468024680246802468024680246802468024680246802468024680246802468024680246802468
```

I have also tried to see if the branching ratios are correct, by removing the code that forces Z & H to decay via bb pairs, I find that H decays through bb pairs 54.3% of the time in reality this should be 57% of the time, I find that Z decays via bb 11% of the time, Im not sure of Z's branching ratio to find if this is true.
Useful link I have found gives branching ratios (page 3) of Z boson
http://pdg.lbl.gov/2012/listings/rpp2012-list-z-boson.pdf



```
Total H created =1000.000
bb to other decay ratio =54.300%
```



```
Total Z created =1000.000
bb to other decay ratio =11.000%
```

13/3/17Week 7

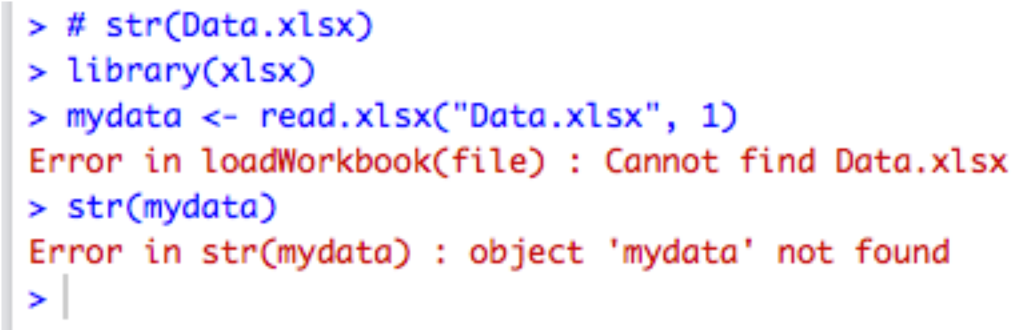
14/3/17 Meeting with lily

I asked lily about getting H decays or 53.3% and Z decayd of 11% and she said they should be around 57% and 15% but using the square root test we can show that this is statistically fine. Running errors and warnings are to be expected and nothing to worry about. We were still unable to get the graphs to work in Pythia, so we have decided to try and do it in R.

The first step was to import homebrew and some Java packages so that we could outpu Pythia data as xlsx files and get R to read these files this was done using the following advice from Lily:

```
#download xlsx library
#https://cran.r-project.org/bin/macosx/mavericks/contrib/3.3/xlsx_0.5.7.tgz
#unpack by double clicking on the  xlsx_0.5.7.tgz
#install (from anywhere)
R CMD INSTALL ~/Downloads/xlsx

# need dependencies!
https://cran.r-project.org/web/packages/rJava/index.html
https://cran.r-project.org/web/packages/xlsxjars/index.html
#blah unpack
R CMD INSTALL ~/Downloads/rJava
R CMD INSTALL ~/Downloads/xlsxjars

#need JDK!
# get homebrew https://brew.sh/
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
brew install Caskroom/cask/java

library(xlsx)
mydata <- read.xlsx("test1.xlsx", 1)
str(mydata)
```

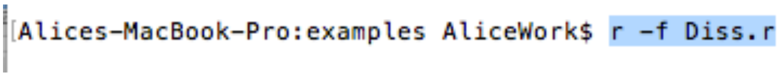
I installed all the packages and homebrew and it all worked fine, but I tried to import things to R and it’s not working.
When I try and run the 'mydata <- read.xlsx("Data.xlsx", 1)' command, I get the error message 'Error in loadWorkbook(file) : Cannot find Data.xlsx' even though the file is definitely called Data.xlsx, I even tried saving the R file to the same folder as the Data file but I’m still getting the same message.



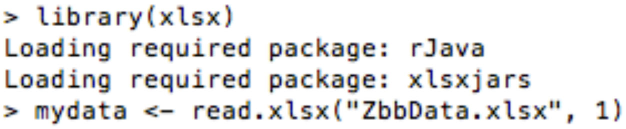
```
> # str(Data.xlsx)
> library(xlsx)
> mydata <- read.xlsx("Data.xlsx", 1)
Error in loadWorkbook(file) : Cannot find Data.xlsx
> str(mydata)
Error in str(mydata) : object 'mydata' not found
> 
```

20/3/17 week 8

Break through, Alice Jordan is a star and came up with the idea of running R through the bash terminal with the command ‘r -f Diss.r’ where Diss.r is my R file This finally worked

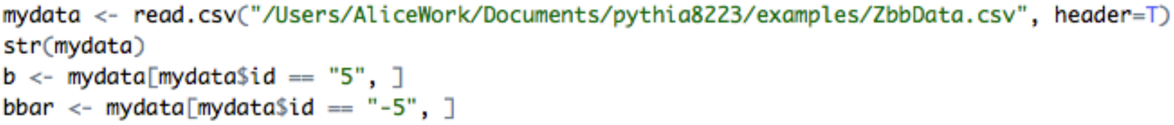


```
[Alices-MacBook-Pro:examples AliceWork$ r -f Diss.r
```

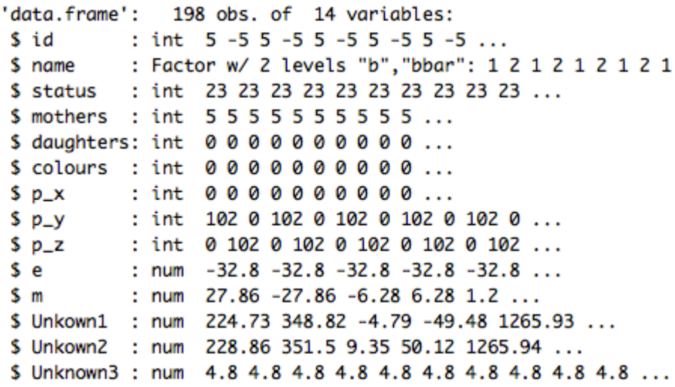


```
> library(xlsx)
Loading required package: rJava
Loading required package: xlsxjars
> mydata <- read.xlsx("ZbbData.xlsx", 1)
```

I’m still struggling to get R to work with xlsx files so have attempted instead to convert the xlsx files to cvs files, I have had some success with this and no longer need to run this through the bash terminal. I have also found a way to create sub sets of data this is shown in the R code below.



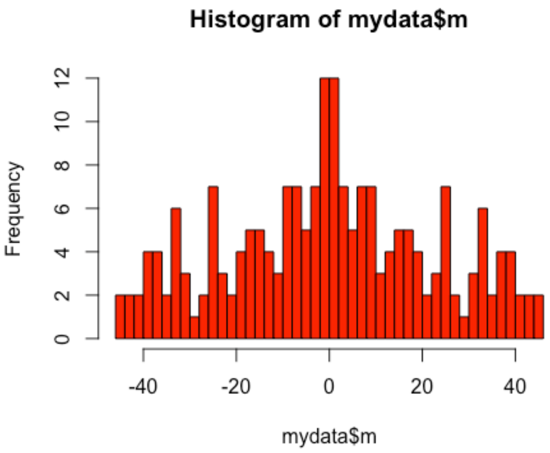
```
mydata <- read.csv("/Users/AliceWork/Documents/pythia8223/examples/ZbbData.csv", header=T)
str(mydata)
b <- mydata[mydata$id == "5", ]
bbar <- mydata[mydata$id == "-5", ]
```

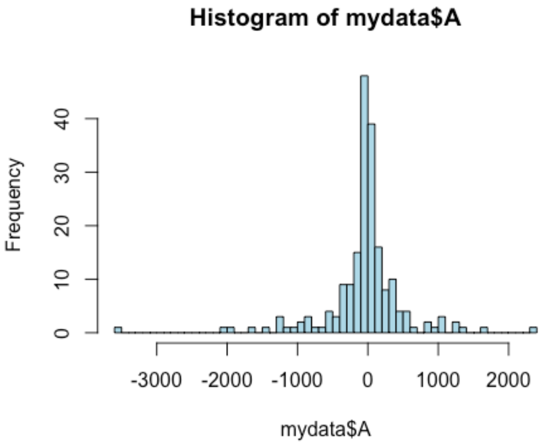


```
'data.frame':   198 obs. of  14 variables:
 $ id       : int  5 -5 5 -5 5 -5 5 -5 5 -5 ...
 $ name     : Factor w/ 2 levels "b","bbar": 1 2 1 2 1 2 1 2 1 2 ...
 $ status   : int  23 23 23 23 23 23 23 23 23 23 ...
 $ mothers  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ daughters: int  0 0 0 0 0 0 0 0 0 0 ...
 $ colours  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ p_x      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ p_y      : int  102 0 102 0 102 0 102 0 102 0 ...
 $ p_z      : int  0 102 0 102 0 102 0 102 0 102 ...
 $ e        : num  -32.8 -32.8 -32.8 -32.8 -32.8 ...
 $ m        : num  27.86 -27.86 -6.28 6.28 1.2 ...
 $ Unkown1  : num  224.73 348.82 -4.79 -49.48 1265.93 ...
 $ Unkown2  : num  228.86 351.5 9.35 50.12 1265.94 ...
 $ Unknown3 : num  4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 ...
```

I have found a usefull website for information on CVS files in R
http://www.plantsci.cam.ac.uk/teaching/matlab/ia-mb/lent-2/practical-5.pdf

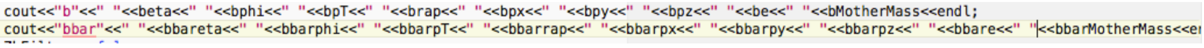
I have so far created histograms from the Z->bb data, although some of the data columns don’t have labels so for now I’ve called them A, B and C. I will ask lily about this.



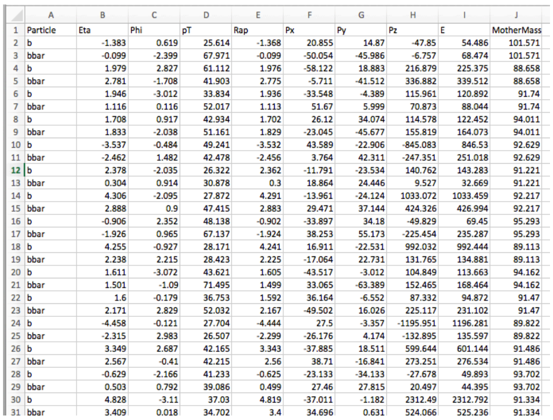


27/3/17 week 9

Now that I've discovered how to create histograms in R I need to change my Pythia output to be in a format of a table with all of the variables going across the top and all of the values for each printed underneath to do this I am changing my code so that it prints all of the information on a particle in one line with spaces in between this way It will come out as a table when I convert it to a CSV file using excel I convert to a CSV in excel by opening the txt file from Pythia (./mymain03 > filename.txt) choosing the delimited option, and selecting tab and space and then deleting the unnecessary cells. The code use to produce the txt output in the format desired is shown below. Spaces are used to separate the values so that excel knows to put them in different cells.

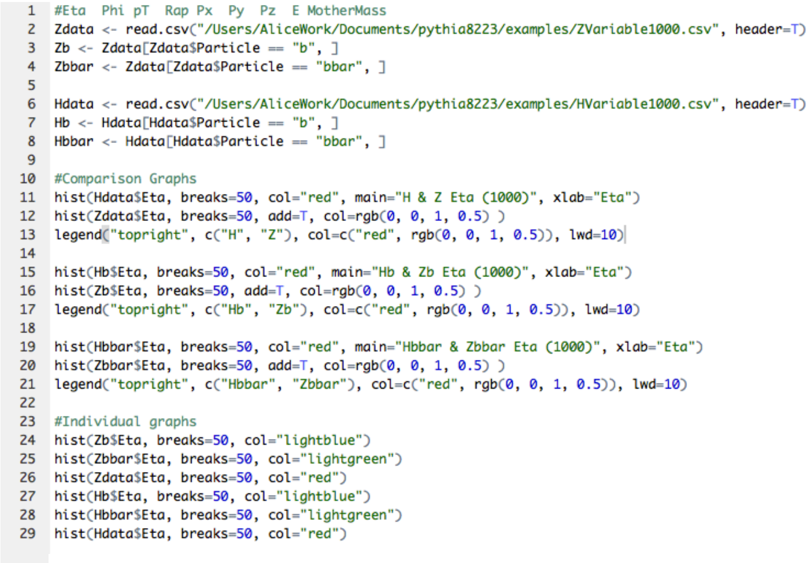


```
cout<<"b"<<" "<<beta<<" "<<bphi<<" "<<bpT<<" "<<brap<<" "<<bpx<<" "<<bpy<<" "<<bpz<<" "<<be<<" "<<bMotherMass<<endl;
cout<<"bbar"<<" "<<bbareta<<" "<<bbarphi<<" "<<bbarpT<<" "<<bbarrap<<" "<<bbarpx<<" "<<bbarpy<<" "<<bbarpz<<" "<<bbare<<" "<<bbarMotherMass<<e
```

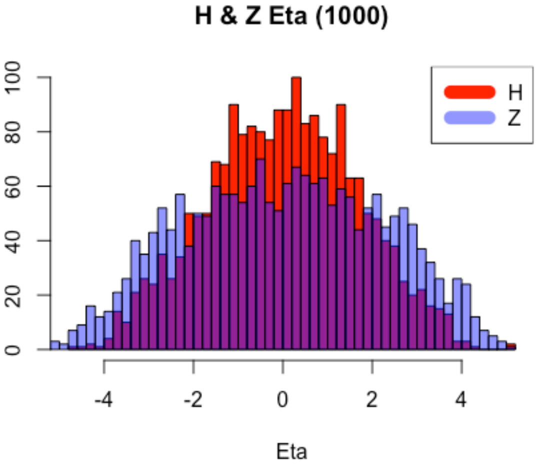


	A	B	C	D	E	F	G	H	I	J
1	Particle	Eta	Phi	pT	Rap	Px	Py	Pz	E	MotherMass
2	b	-1.383	0.619	25.614	-1.368	20.855	14.87	-47.85	54.486	101.571
3	bbar	-0.099	-2.399	67.971	-0.099	-50.054	-45.986	-6.757	68.474	101.571
4	b	1.979	2.827	61.112	1.976	-58.122	18.883	216.879	225.375	88.658
5	bbar	2.781	-1.708	41.903	2.775	-5.711	-41.512	336.882	339.512	88.658
6	b	1.946	-3.012	33.834	1.936	-33.548	-4.389	115.961	120.892	91.74
7	bbar	1.116	0.116	52.017	1.113	51.67	5.999	70.873	88.044	91.74
8	b	1.708	0.917	42.934	1.702	26.12	34.074	114.578	122.452	94.011
9	bbar	1.833	-2.038	51.161	1.829	-23.045	-45.677	155.819	164.073	94.011
10	b	-3.537	-0.484	49.241	-3.532	43.589	-22.906	-845.083	846.53	92.629
11	bbar	-2.462	1.482	42.478	-2.456	3.764	42.311	-247.351	251.018	92.629
12	b	2.378	-2.035	26.322	2.362	-11.791	-23.534	140.762	143.283	91.221
13	bbar	0.304	0.914	30.878	0.3	18.864	24.446	9.527	32.669	91.221
14	b	4.306	-2.095	27.872	4.291	-13.961	-24.124	1033.072	1033.459	92.217
15	bbar	2.888	0.9	47.415	2.883	29.471	37.144	424.326	426.994	92.217
16	b	-0.906	2.352	48.138	-0.902	-33.897	34.18	-49.829	69.45	95.293
17	bbar	-1.926	0.965	67.137	-1.924	38.253	55.173	-225.454	235.287	95.293
18	b	4.255	-0.927	28.171	4.241	16.911	-22.531	992.032	992.444	89.113
19	bbar	2.238	2.215	28.423	2.225	-17.064	22.731	131.765	134.881	89.113
20	b	1.611	-3.072	43.621	1.605	-43.517	-3.012	104.849	113.663	94.162
21	bbar	1.501	-1.09	71.495	1.499	33.065	-63.389	152.465	168.464	94.162
22	b	1.6	-0.179	36.753	1.592	36.164	-6.552	87.332	94.872	91.47
23	bbar	2.171	2.829	52.032	2.167	-49.502	16.026	225.117	231.102	91.47
24	b	-4.458	-0.121	27.704	-4.444	27.5	-3.357	-1195.951	1196.281	89.822
25	bbar	-2.315	2.983	26.507	-2.299	-26.176	4.174	-132.895	135.597	89.822
26	b	3.349	2.687	42.165	3.343	-37.885	18.511	599.644	601.144	91.486
27	bbar	2.567	-0.41	42.215	2.56	38.71	-16.841	273.251	276.534	91.486
28	b	-0.629	-2.166	41.233	-0.625	-23.133	-34.133	-27.678	49.893	93.702
29	bbar	0.503	0.792	39.086	0.499	27.46	27.815	20.497	44.395	93.702
30	b	4.828	-3.11	37.03	4.819	-37.011	-1.182	2312.49	2312.792	91.334
31	bbar	3.409	0.018	34.702	3.4	34.696	0.631	524.066	525.236	91.334

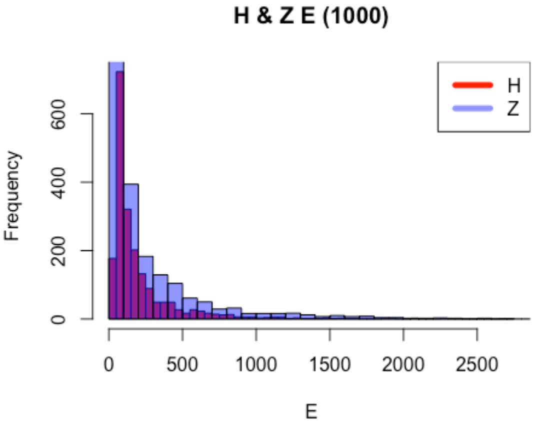
I ran enough simulations to get 1000 decays for both H and Z, I then used the following code to plot the frequency histograms of Eta for Hb, Hbbar, H b & bbar, Zb, Zbbar and Z b & bbar. I also made a histogram with both the b and bbar Z and H Eta for comparison, this can be seen below.



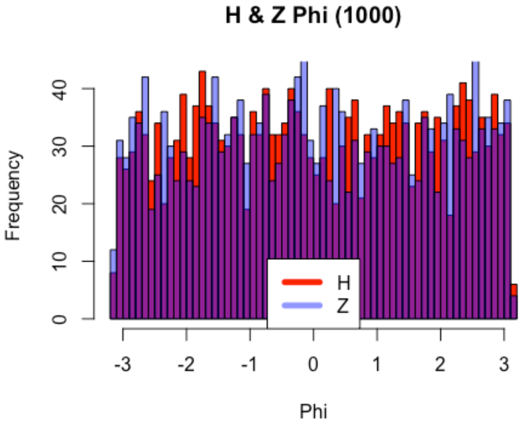
```
#Eta  Phi pT  Rap Px  Py  Pz  E MotherMass
Zdata <- read.csv("/Users/AliceWork/Documents/pythia8223/examples/ZVariable1000.csv", header=T)
Zb <- Zdata[Zdata$Particle == "b", ]
Zbbar <- Zdata[Zdata$Particle == "bbar", ]

Hdata <- read.csv("/Users/AliceWork/Documents/pythia8223/examples/HVariable1000.csv", header=T)
Hb <- Hdata[Hdata$Particle == "b", ]
Hbbar <- Hdata[Hdata$Particle == "bbar", ]

#Comparison Graphs
hist(Hdata$Eta, breaks=50, col="red", main="H & Z Eta (1000)", xlab="Eta")
hist(Zdata$Eta, breaks=50, add=T, col=rgb(0, 0, 1, 0.5) )
legend("topright", c("H", "Z"), col=c("red", rgb(0, 0, 1, 0.5)), lwd=10)

hist(Hb$Eta, breaks=50, col="red", main="Hb & Zb Eta (1000)", xlab="Eta")
hist(Zb$Eta, breaks=50, add=T, col=rgb(0, 0, 1, 0.5) )
legend("topright", c("Hb", "Zb"), col=c("red", rgb(0, 0, 1, 0.5)), lwd=10)

hist(Hbbar$Eta, breaks=50, col="red", main="Hbbar & Zbbar Eta (1000)", xlab="Eta")
hist(Zbbar$Eta, breaks=50, add=T, col=rgb(0, 0, 1, 0.5) )
legend("topright", c("Hbbar", "Zbbar"), col=c("red", rgb(0, 0, 1, 0.5)), lwd=10)

#Individual graphs
hist(Zb$Eta, breaks=50, col="lightblue")
hist(Zbbar$Eta, breaks=50, col="lightgreen")
hist(Zdata$Eta, breaks=50, col="red")
hist(Hb$Eta, breaks=50, col="lightblue")
hist(Hbbar$Eta, breaks=50, col="lightgreen")
hist(Hdata$Eta, breaks=50, col="red")
```



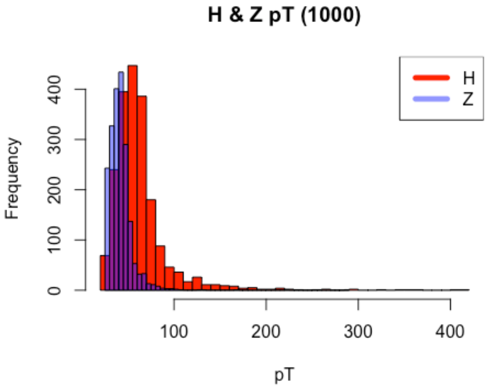
H & Z Eta (1000)



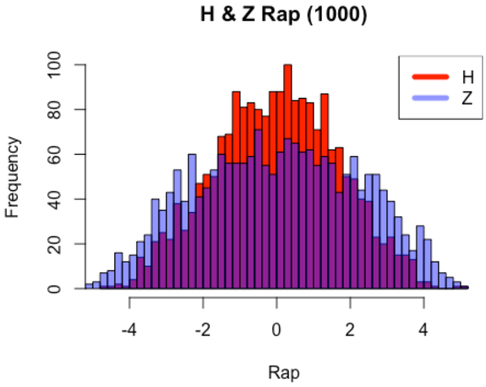
H & Z E (1000)



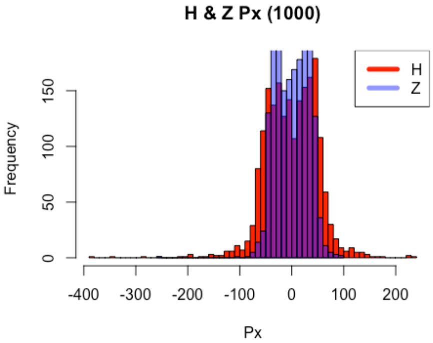
H & Z Phi (1000)



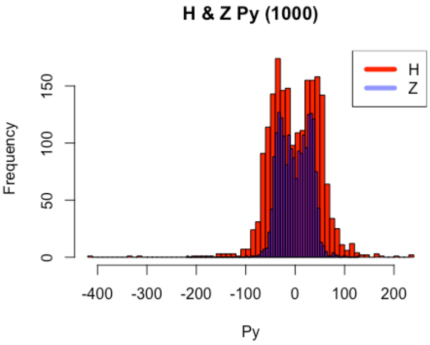
H & Z pT (1000)



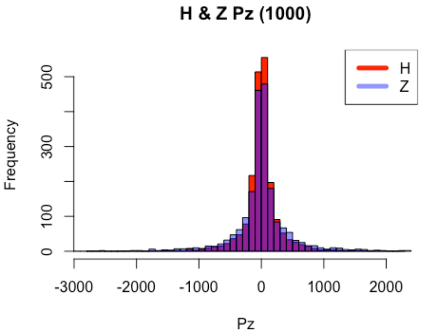
H & Z Rap (1000)



H & Z Px (1000)



H & Z Py (1000)



H & Z Pz (1000)

The only graphs that seem to separate the H and Z data are pT, Px and Py but these may be correlated.

10:30 friday meeting for the next two weeks. Start report, scikit-learn.

Meeting with Lily:

- Is it okay to use CSV rather than XLSX?
- What are the units for the variables?
- Check the Mother Mass Graphs
- Are pT, Px & Py correlated?
- Next steps