Exercise 3: Getting started with PyROOT

Instructions

- Learn differences between c++ and python based ROOT
- Untar `Ex3.tar`
- Read in root file: `toy_sigbkg.root`
- Get tree: `TreeS`
- Loop over entries and read in two variables `var1` and `var2`
- Make a 2D correlation plot of `var1` and `var2`
- Get the covariance of the the variables

Open a new python file for instance `Ex3.py`. Execute `python Ex3.py`.
As for every python program, you need to import the needed modules. Since the ROOT library is huge, it is advisable to specify the needed classes:

In [ ]:

```
import os,sys
from ROOT import TH1D,TH2D,TFile,TTree,TCanvas
```

Now read in the root file and tree.
Remember: No need to specify types in python (dynamic type) and no semicolons are need:

In [1]:

```
fFile = TFile("toy_sigbkg.root", "READ")
fTree = fFile.Get("TreeS")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-ba1d294fb1b1> in <module>()
----> 1 fFile = TFile("toy_sigbkg.root", "READ")
      2 fTree = fFile.Get("TreeS")

NameError: name 'TFile' is not defined
```

Similarly, we can define a canvas and a 2D histogram:

In [ ]:

```
fCanvas = TCanvas("c", "c", 600, 600)
fHist   = TH2D("var1var2", "", 20, -6, 6, 20, -6, 6)
```

Now, like in the C++ version, get the number of events in the tree (**`GetEntries()`**) and loop over the tree like this
`for i in range(0, nEntries):` in order to fill the histogram.
In contrast to C++ ROOT, you don't need to link the branches to variables, they are directly available via **`tree.var`**

In [ ]:

```
nEntries = fTree.GetEntries()

for i in range(0, nEntries):
    fTree.GetEntry(i)
    fHist.Fill(fTree.var1, fTree.var2)
fHist.Print()
```

You can actually do this even faster in PyROOT, just loop over the object `TTree`:

In [5]:

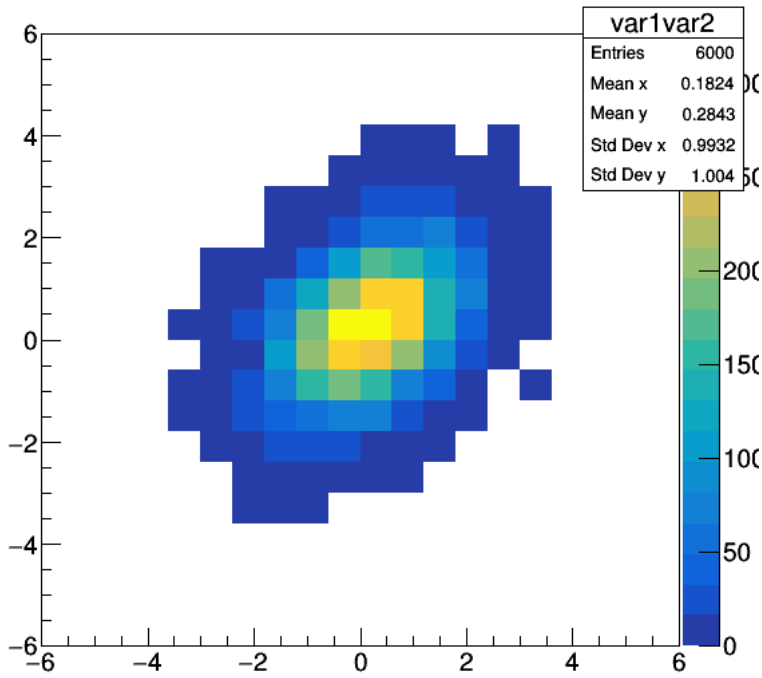
```
fHist.Reset();
for i in fTree:
    fHist.Fill(fTree.var1, fTree.var2)
fHist.Print()
```

```
TH1.Print Name  = var1var2, Entries= 6000, Total sum= 6000
```

Wow, that is really much easier!
Ok, almost done, we can now draw the histogram and print it to a file:

In [6]:

```
fHist.Draw("COLZ")
fCanvas.Draw()
fCanvas.Print("toy_sigbkg_corr.eps")
```




```
Info in <TCanvas::Print>: eps file toy_sigbkg_corr.eps has been created
```

And finally we can get the covariance of the two variables, which quantifies how strongly the are correlated:

In [7]:

```
print fHist.GetCovariance()
```

```
0.385349730897
```