

Photon_PY

February 3, 2021

1 Photon Classes

```
[1]: import ROOT
import os

# Create a ROOT dataframe for each dataset
path = "root://eospublic.cern.ch/eos/opendata/atlas/OutreachDatasets/
↳2020-01-22"
df = ROOT.RDataFrame("mini", (os.path.join(path, "GamGam/Data/data_A.GamGam.
↳root")))
```

Welcome to JupyROOT 6.20/06

1.1 first we implement a class for the photons

in a separate header file I have the following code:

```
//photon_struct.h
#include "ROOT/RVec.hxx"
#include <vector>

#ifndef PHOTON_H
#define PHOTON_H
struct photon {
    float pt;
    float eta;
    float phi;
    float m;

    photon(float init_pt=-999.0, float init_eta=0.0, float init_phi=0.0, float init_m=0.0)
        : pt(init_pt), eta(init_eta), phi(init_phi), m(init_m) {}
};

#endif

#ifdef __ROOTCLING__
#pragma link C++ class vector<photon,ROOT::Detail::VecOps::RAdoptAllocator<photon> >;
#pragma link C++ class ROOT::VecOps::RVec<photon>;
#endif
```

compiled by opening root and running `root [0] .L photon_struct.h+`

```
[2]: ROOT.gSystem.Load("photon_struct_h.so")
```

```
[2]: 0
```

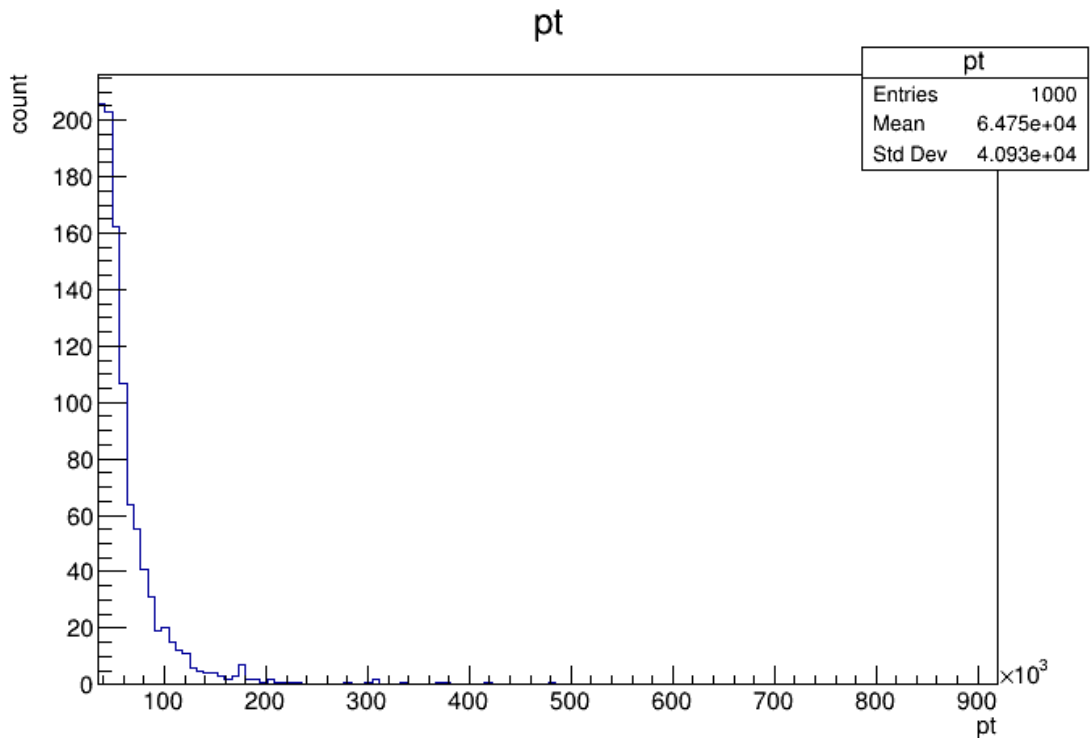
1.2 Use the photon class in my RDF

Here we use the VecOps Construct operation to populate the stucture

```
[3]: df = df.Define('photons', 'ROOT::VecOps::Construct<photon>(photon_pt, \u2192  
    \u2192photon_eta, photon_phi, photon_E)')
```

And check that it works...

```
[4]: photon_pt = df.Range(0,1000).Define("pt", "photons.at(0,photon()).pt").  
    \u2192Histo1D("pt")  
c = ROOT.TCanvas()  
photon_pt.Draw()  
c.Draw()
```



1.3 Errors.

1.3.1 How to snapshot this custom object?

```
[5]: branch_list = ROOT.vector('string')()
branch_list.push_back("photons")
snapshot_df = df.Range(0,1000).Snapshot('nominal', 'test_classes.root',
↳branch_list)
```

is the photon collection in the events?

```
[6]: f = ROOT.TFile.Open('test_classes.root')
for event in f.nominal:
    print(event.photons.pt)
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-6-ac1ddc596dea> in <module>
      1 f = ROOT.TFile.Open('test_classes.root')
      2 for event in f.nominal:
----> 3     print(event.photons.pt)

AttributeError: 'vector<photon,ROOT::Detail::VecOps::RAdoptAllocato' object has
↳no attribute 'pt'
```

nope

1.3.2 How to sort these collections?

```
[7]: df = df.Define('sorted_photons', 'ROOT::VecOps::Sort(photons, [] (photon p0,
↳photon p1){return p0.pt<p1.pt;})')
```

```
[8]: sorted_photon_pt = df.Range(0,1000).Define("pt", "sorted_photons.at(0,photon()).
↳pt").Histo1D("pt")
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-8-06d711e2bf21> in <module>
----> 1 sorted_photon_pt = df.Range(0,1000).Define("pt", "sorted_photons.
↳at(0,photon()).pt").Histo1D("pt")

TypeError: can not resolve method template call for 'Define'
```

```
input_line_223:1:68: error: variable has incomplete type
'__rdf1::sorted_photons9_type' (aka 'void')
namespace __rdf_3{ auto rdf_f = []() {__rdf1::sorted_photons9_type
sorted_photons;
```

1.3.3 How to fill these with standard values?

eg a massless photon would have a mass of zero (or data might have an event weight scale factor of 1, per object)

```
[9]: df = df.Define('massless_photons', 'ROOT::VecOps::Construct<photon>(photon_pt, photon_eta, photon_phi, 0)')
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-9-1ec45d37c9c9> in <module>  
----> 1 df = df.Define('massless_photons', 'ROOT::VecOps::  
      ↳Construct<photon>(photon_pt, photon_eta, photon_phi, 0)')
```

TypeError: can not resolve method template call for 'Define'

```
input_line_224:4:8: error: no matching function for call to 'Construct'  
return ROOT::VecOps::Construct<photon>(photon_pt, photon_eta, photon_phi, 0)  
      ^~~~~~  
/cvmfs/sft.cern.ch/lcg/releases/ROOT/v6.20.06-9e6ed/x86_64-centos7-gcc8-opt/include/ROOT/RVec.hxx:1644:9: note: candidate template ignored: could not match 'RVec<type-parameter-0-1>' against 'int'  
RVec<T> Construct(const RVec<Args_t> &... args)  
      ^
```