Corina Nantais
Nu/DM meeting
21 June 2018

- number of events for a given file, numu 000
- follow all weights

- should scale to pot, look at mode and pnu

Follow number of events

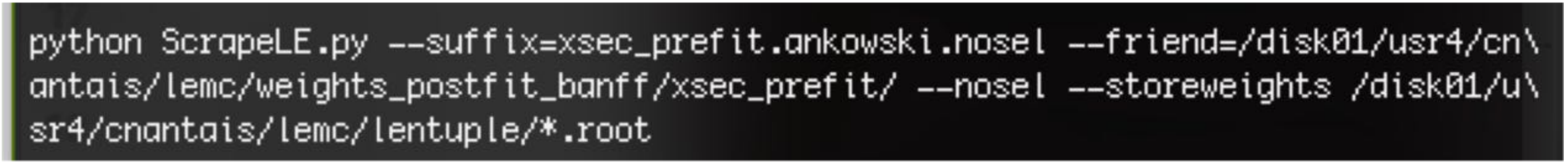
- neutfile/ .dat (3000 events * 100 files * 3 flavours) = 9e5 events
- neutfile/select/ .dat means <3000 events each (* 100 files * 3 flavours)
- ProcessNCEL_mc.sh takes in neutfile/select .dat and does detsim, lowfit, lentuple, and weights_postfit_banff
- ScrapeLE.py (and Scrapper.py and Sel.py) take lentuple and weights_postfit_banff and makes ncgamma.xsec_prefit.ankowski.nosel.root

for a given file, numu 000:

• neutfile/ .dat	3000	decoder for .dat and .zbs
• neutfile/select/ .dat	1842	
• destim/ .zbs	1842	
• lowfit/ .zbs	736	lowfit cuts events I guess?
• lentuple/ .root	736	
• weights_postfit_banff/xsec_prefit/ .root	736	
• ncgamma.xsec_prefit.ankowski.nosel.root	150,343	combination of all 300 files? (300* ~700 = 210,000)

- ncgamma.xsec_prefit.ankowski.nosel.root made by runscrape.csh,
- so ScrapeLE.py, which imports Sel.py and ScrapeLE.py

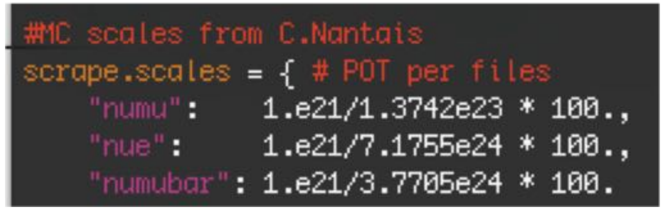
runscrape.csh



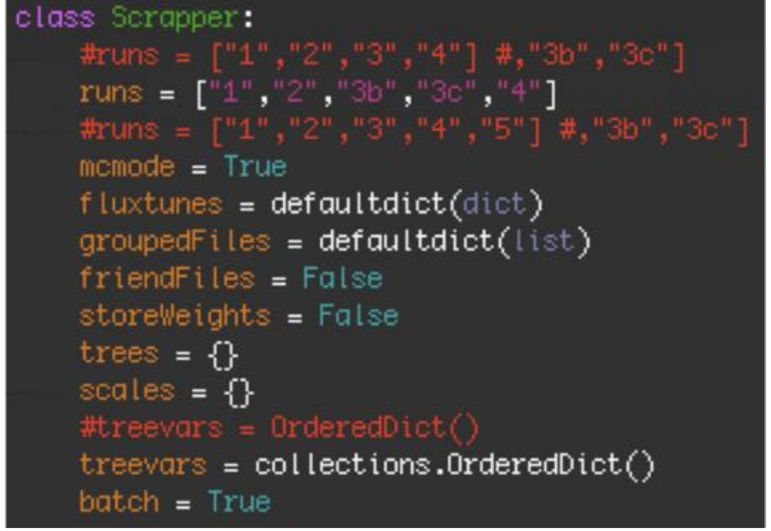
```
python ScrapeLE.py --suffix=xsec_prefit.ankowski.nosel --friend=/disk01/usr4/cn\
antais/lemc/weights_postfit_banff/xsec_prefit/ --nosel --storeweights /disk01/u\
sr4/cnantais/lemc/lentuple/*.root
```

- ScrapeLE.py has MC scales

ScrapeLE.py



```
#MC scales from C.Nantais
scrape.scales = { # POT per files
    "numu":    1.e21/1.3742e23 * 100.,
    "nue":     1.e21/7.1755e24 * 100.,
    "numubar": 1.e21/3.7705e24 * 100.
```



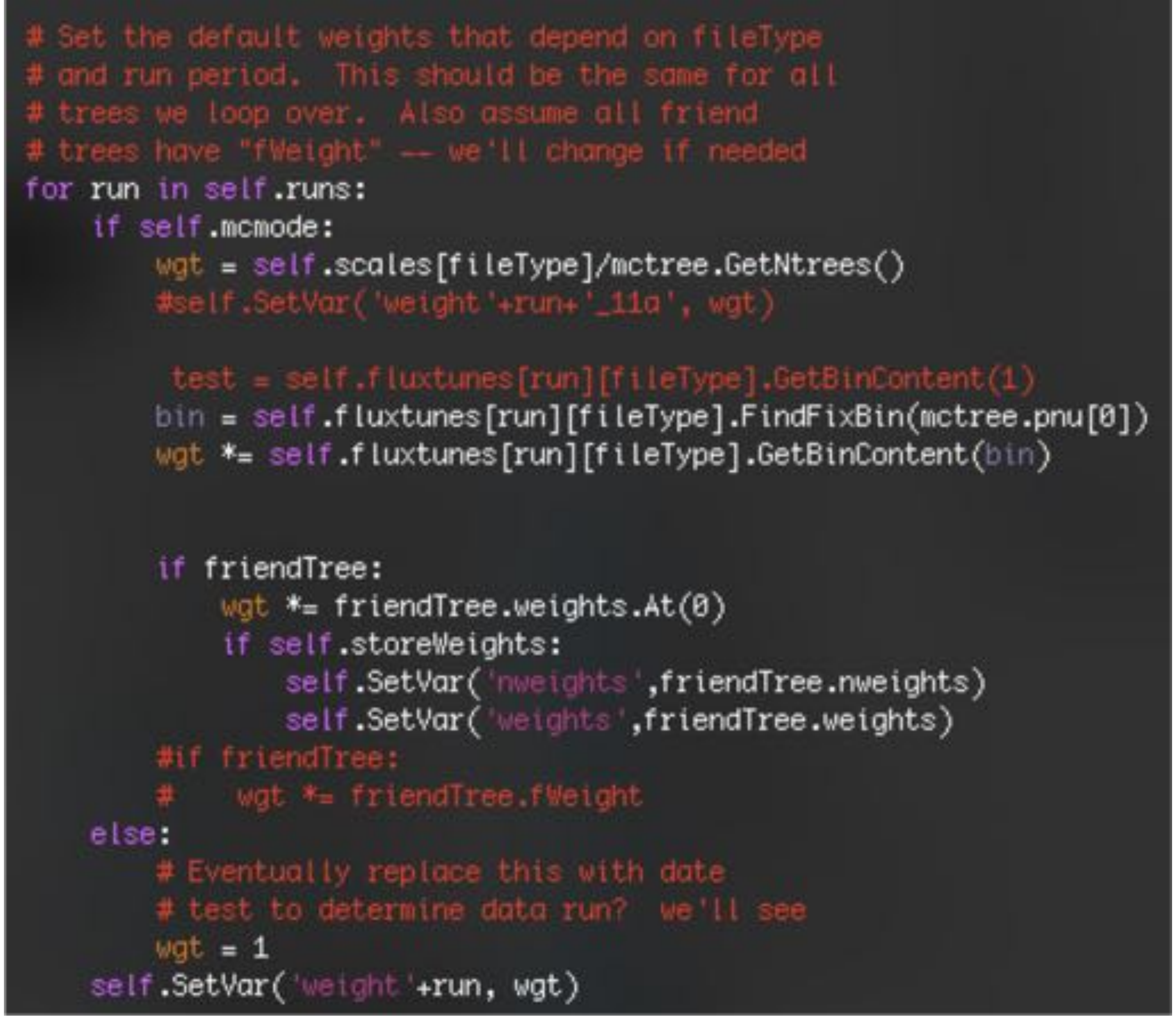
```
class Scrapper:
    #runs = ["1","2","3","4"] #,"3b","3c"]
    runs = ["1","2","3b","3c","4"]
    #runs = ["1","2","3","4","5"] #,"3b","3c"]
    mcmode = True
    fluxtunes = defaultdict(dict)
    groupedFiles = defaultdict(list)
    friendFiles = False
    storeWeights = False
    trees = {}
    scales = {}
    #treevars = OrderedDict()
    treevars = collections.OrderedDict()
    batch = True
```

Scrapper.py

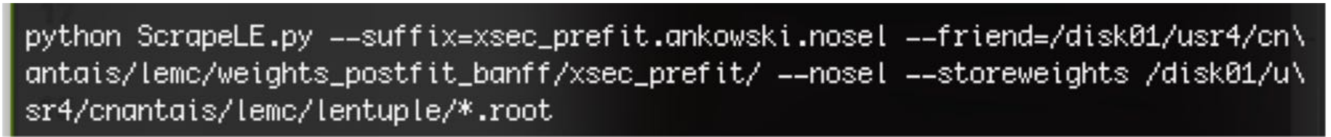
weight4 is branch in ncgamma.xsec_prefit.ankowski.nosel.root

weight4

- scales
- flux
- xsec (from xsec_prefit)



```
# Set the default weights that depend on fileType
# and run period.  This should be the same for all
# trees we loop over.  Also assume all friend
# trees have "fWeight" -- we'll change if needed
for run in self.runs:
    if self.mcmode:
        wgt = self.scales[fileType]/mctree.GetNtrees()
        #self.SetVar('weight'+run+'_11a', wgt)

         test = self.fluxtunes[run][fileType].GetBinContent(1)
        bin = self.fluxtunes[run][fileType].FindFixBin(mctree.pnu[0])
        wgt *= self.fluxtunes[run][fileType].GetBinContent(bin)


        if friendTree:
            wgt *= friendTree.weights.At(0)
            if self.storeWeights:
                self.SetVar('nweights',friendTree.nweights)
                self.SetVar('weights',friendTree.weights)
        #if friendTree:
        #    wgt *= friendTree.fWeight
    else:
        # Eventually replace this with date
        # test to determine data run?  we'll see
        wgt = 1
    self.SetVar('weight'+run, wgt)
```

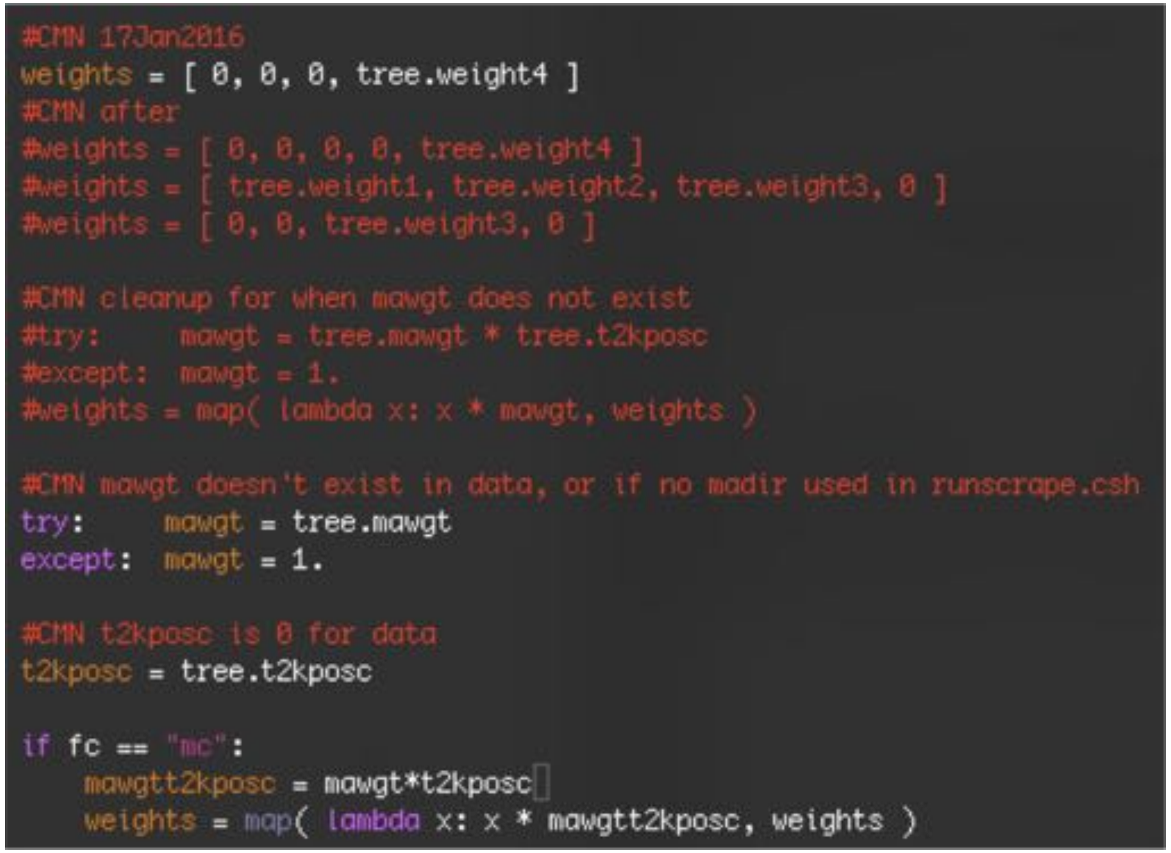


```
python ScrapeLE.py --suffix=xsec_prefit.ankowski.nosel --friend=/disk01/usr4/cn\
antais/lemc/weights_postfit_banff/xsec_prefit/ --nosel --storeweights /disk01/u\
sr4/cnantais/lemc/lentuple/*.root
```

- weight4 is applied in SelectionPlots.py

weights

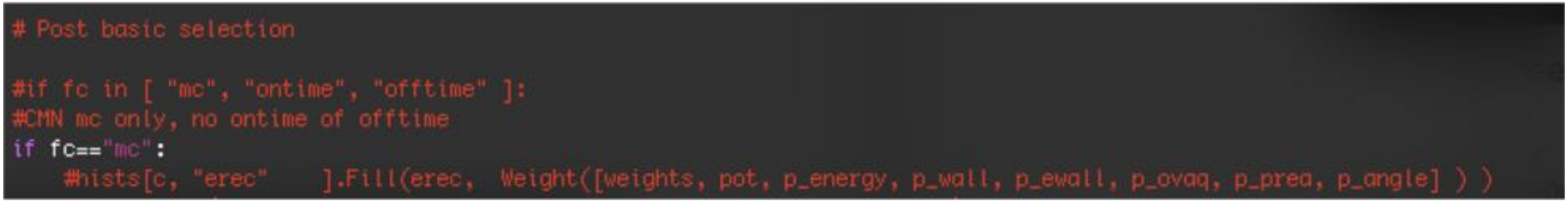
- weight4 (scales, flux, xsec)
- (mawgt)
- t2kposc



```
#CMN 17Jan2016
weights = [ 0, 0, 0, tree.weight4 ]
#CMN after
#weights = [ 0, 0, 0, 0, tree.weight4 ]
#weights = [ tree.weight1, tree.weight2, tree.weight3, 0 ]
#weights = [ 0, 0, tree.weight3, 0 ]

#CMN cleanup for when mawgt does not exist
#try:     mawgt = tree.mawgt * tree.t2kposc
#except:  mawgt = 1.
#weights = map( lambda x: x * mawgt, weights )

#CMN mawgt doesn't exist in data, or if no madir used in runscrape.csh
try:     mawgt = tree.mawgt
except:  mawgt = 1.

#CMN t2kposc is 0 for data
t2kposc = tree.t2kposc

if fc == "mc":
    mawgtt2kposc = mawgt*t2kposc
    weights = map( lambda x: x * mawgtt2kposc, weights )
```

Final Weight in SelectionPlots.py

SelectionPlots.py

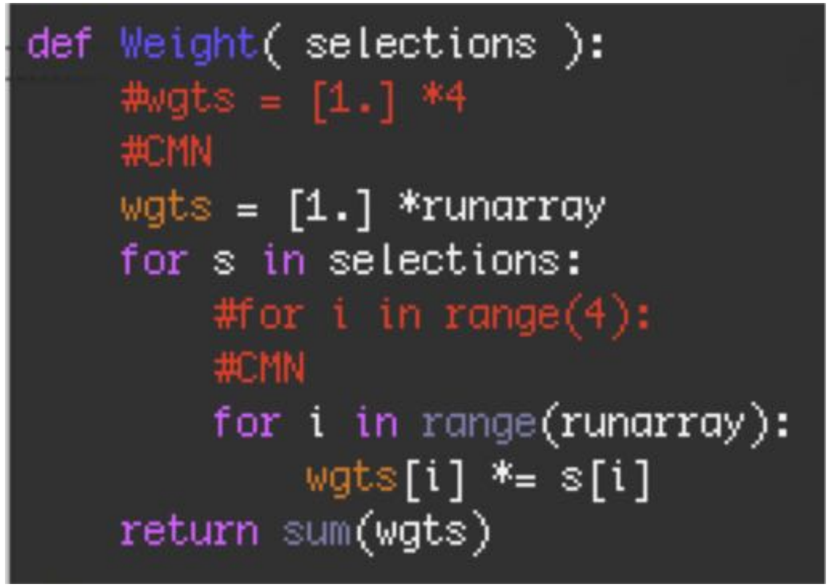


```
# Post basic selection

#if fc in [ "mc", "ontime", "offtime" ]:
#CMN mc only, no ontime of offtime
if fc=="mc":
    #hists[c, "erec"    ].Fill(erec,  Weight([weights, pot, p_energy, p_wall, p_ewall, p_ovaq, p_prea, p_angle] ) )
```

Weight

- weights (weight4 (scales, flux, xsec), mawgt, t2kposc)
- pot
- cuts for energy, wall, ewall, ovaq, prea, and angle

combination of Weight for each run, but we’re only doing Run 4



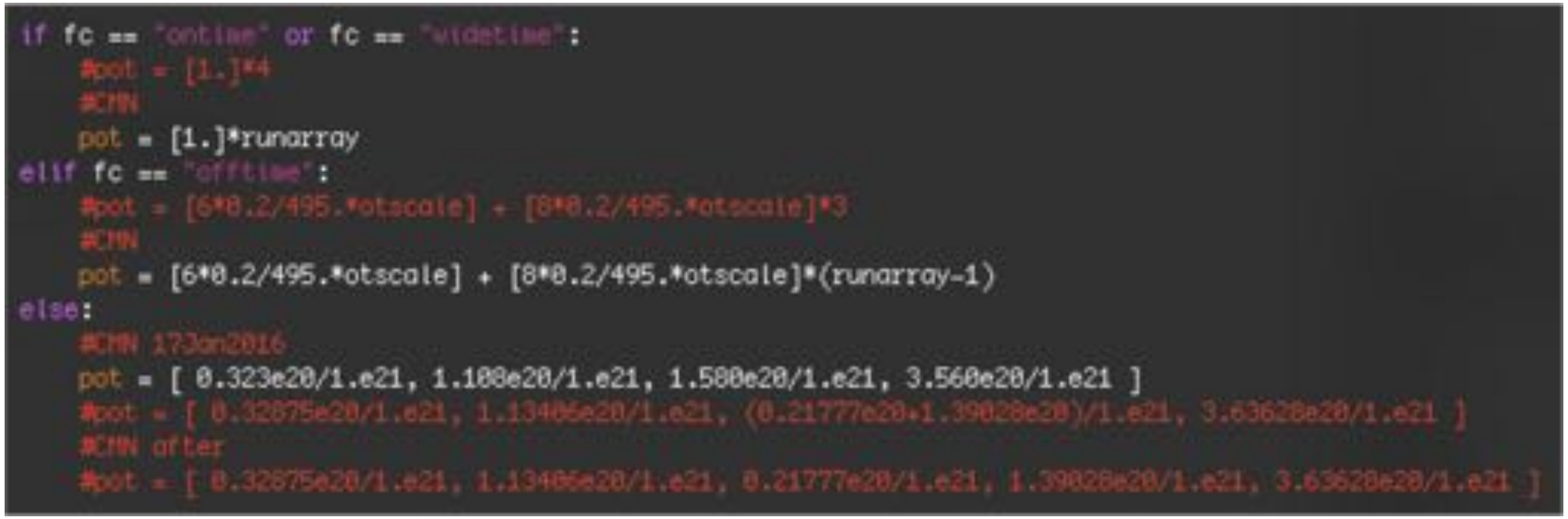
```
def Weight( selections ):
    #wgts = [1.] *4
    #CMN
    wgts = [1.] *runarray
    for s in selections:
        #for i in range(4):
        #CMN
        for i in range(runarray):
            wgts[i] *= s[i]
    return sum(wgts)
```

SelectionPlots.py

Scale to POT

- previously, had turned off all weights
- want to scale to POT
- compare mode and pnu, for 17Jan2016 and current

SelectionPlots.py



```
if fc == "ontime" or fc == "widetime":
    #pot = [1.]*4
    #CMN
    pot = [1.]*runarray
elif fc == "offtime":
    #pot = [6*0.2/495.*otscale] + [8*0.2/495.*otscale]*3
    #CMN
    pot = [6*0.2/495.*otscale] + [8*0.2/495.*otscale]*(runarray-1)
else:
    #CMN 17Jan2016
    pot = [ 0.323e20/1.e21, 1.108e20/1.e21, 1.580e20/1.e21, 3.560e20/1.e21 ]
    #pot = [ 0.32875e20/1.e21, 1.13406e20/1.e21, (0.21777e20+1.39828e20)/1.e21, 3.63628e20/1.e21 ]
    #CMN after
    #pot = [ 0.32875e20/1.e21, 1.13406e20/1.e21, 0.21777e20/1.e21, 1.39828e20/1.e21, 3.63628e20/1.e21 ]
```

I updated POT