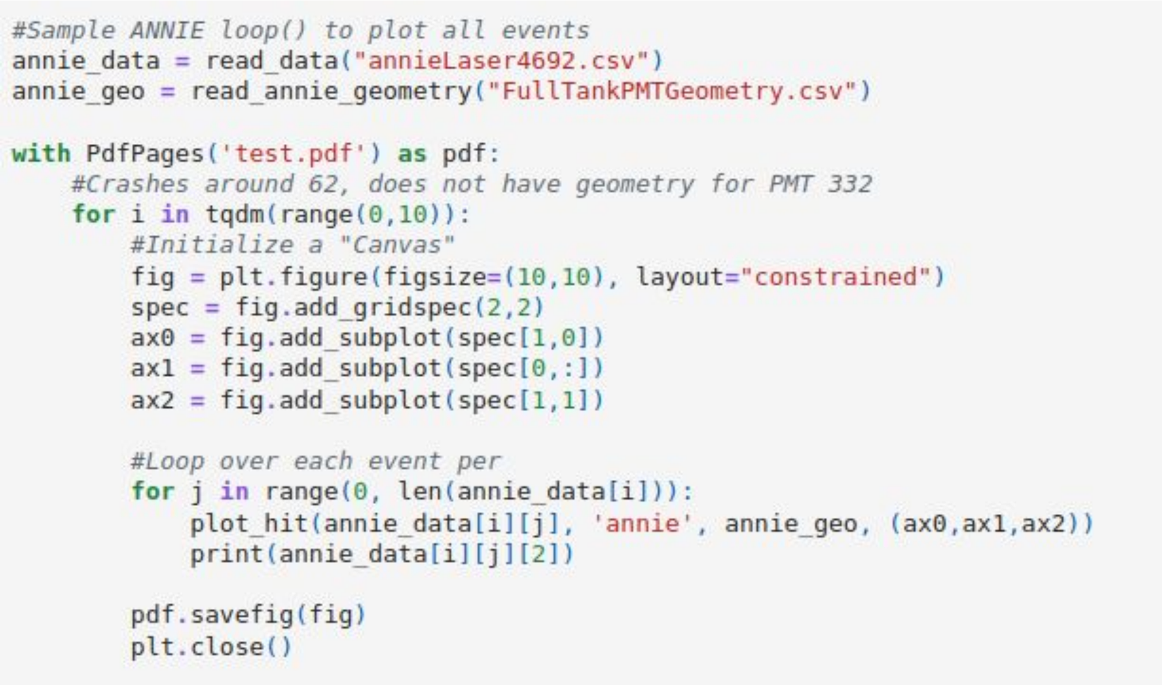
Event Display

Access Data Through extract.py or laserRunDataExtraction.ipynb

- ANNIE Data
    - Geometry is stored in “FullTankPMTGeometry.csv”
    - Data is stored in ntuple files and can be extracted with the jupyter notebook code
- EOS Data
    - Each EOS ntuple file has both geometry and data stored in them
    - The extract.py script will make two csv’s
        - One for geometry
        - One for data

Use pmtMapping.ipynb to Load and Present Data

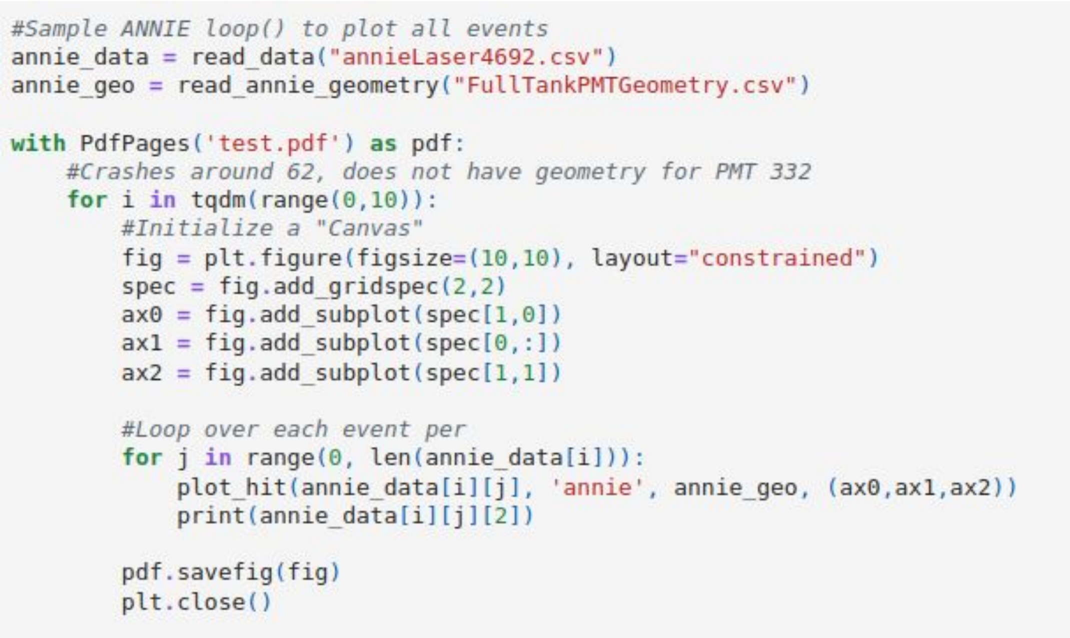
- The first half of the code contains functions to load data into the workspace
- The second half is up to the user to build displays as they see fit
- E.g.



```
#Sample ANNIE loop() to plot all events
annie_data = read_data("annieLaser4692.csv")
annie_geo = read_annie_geometry("FullTankPMTGeometry.csv")

with PdfPages('test.pdf') as pdf:
    #Crashes around 62, does not have geometry for PMT 332
    for i in tqdm(range(0,10)):
        #Initialize a "Canvas"
        fig = plt.figure(figsize=(10,10), layout="constrained")
        spec = fig.add_gridspec(2,2)
        ax0 = fig.add_subplot(spec[1,0])
        ax1 = fig.add_subplot(spec[0,:])
        ax2 = fig.add_subplot(spec[1,1])

        #Loop over each event per
        for j in range(0, len(annie_data[i])):
            plot_hit(annie_data[i][j], 'annie', annie_geo, (ax0,ax1,ax2))
            print(annie_data[i][j][2])

        pdf.savefig(fig)
        plt.close()
```

Think of the Loop as the MakeClass Loop in ROOT

- Load the Data and Geometry from the desired CSVs
- Use the PdfPages package to make a pdf
- The outer loop i, loops through events
  - Fig is analogous to TH2F
  - Spec and ax are like members of the Fig “class”
- Use the inner loop j, to go through and plot each hit for event i
- The figure is plotted and saved to the pdf



```
#Sample ANNIE loop() to plot all events
annie_data = read_data("annieLaser4692.csv")
annie_geo = read_annie_geometry("FullTankPMTGeometry.csv")

with PdfPages('test.pdf') as pdf:
    #Crashes around 62, does not have geometry for PMT 332
    for i in tqdm(range(0,10)):
        #Initialize a "Canvas"
        fig = plt.figure(figsize=(10,10), layout="constrained")
        spec = fig.add_gridspec(2,2)
        ax0 = fig.add_subplot(spec[1,0])
        ax1 = fig.add_subplot(spec[0,:])
        ax2 = fig.add_subplot(spec[1,1])

        #Loop over each event per
        for j in range(0, len(annie_data[i])):
            plot_hit(annie_data[i][j], 'annie', annie_geo, (ax0,ax1,ax2))
            print(annie_data[i][j][2])

        pdf.savefig(fig)
        plt.close()
```