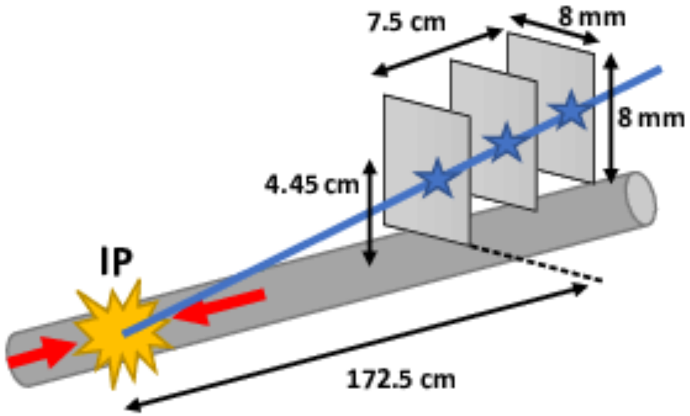
Detecting Anomalies in the PLT Detector

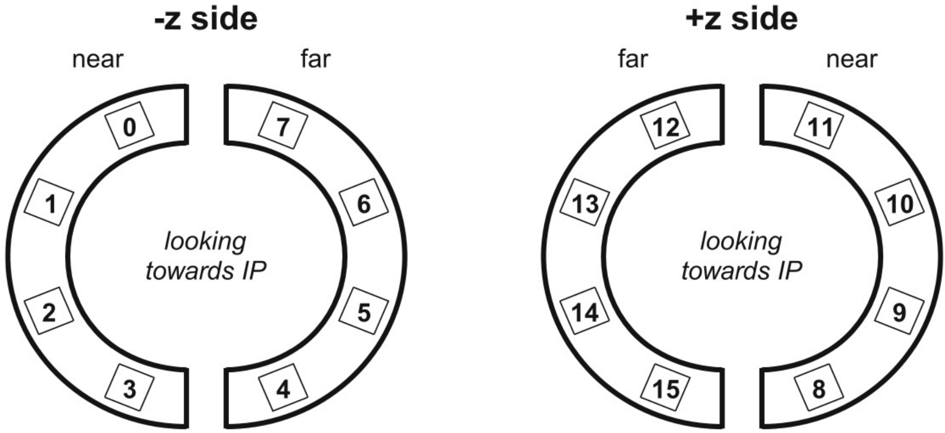
Katie Ream

June 24, 2024

Review of Project and Progress (talk 1)

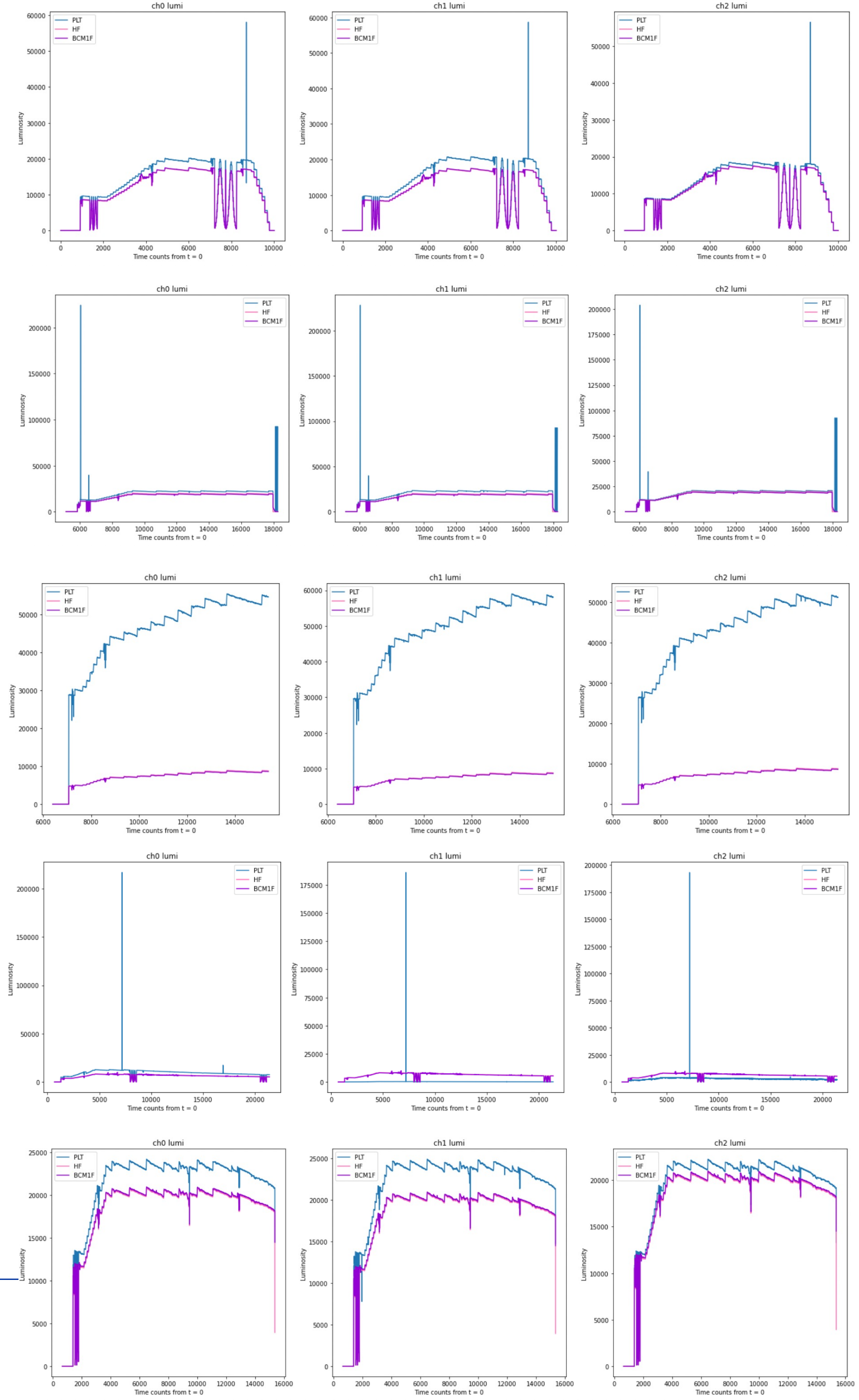


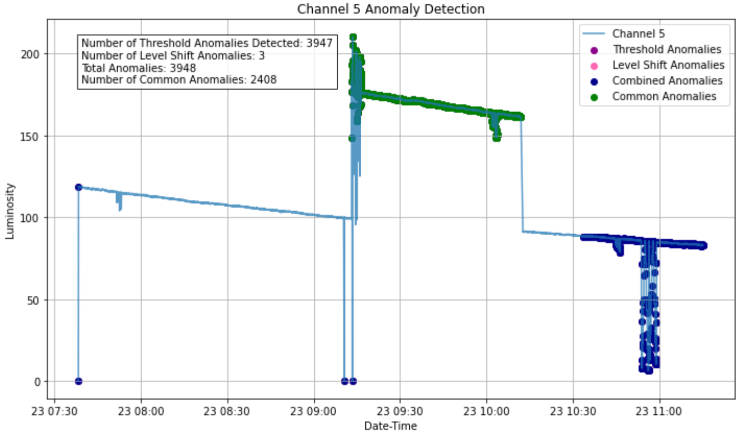
- The PLT telescope measures luminosity for CMS and consists of 16 channels
- These channels have anomalies due to one reason or another, thus we need to design an algorithm that can flag these anomalies so they can be removed
- For 2023 data, channel selection needs to be performed (anomaly detection on broader scale)
- Last talk I had a model that flagged “common anomalies” between two different graphs and tested out on a singular fill of data – needed to apply on a larger scale



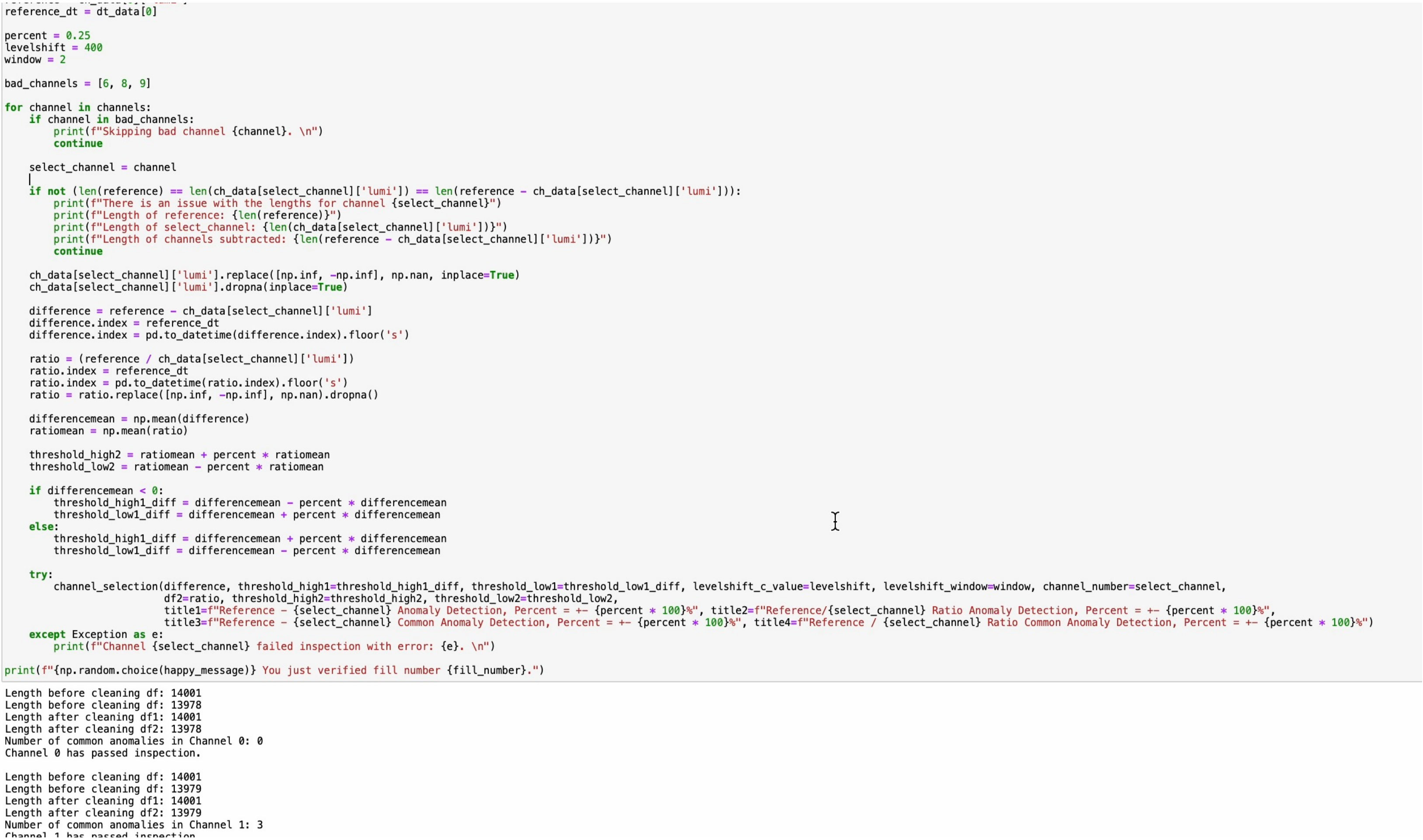
Progress Made Since Talk 1

- Model is complete and has been tested on a variety of fills (example is shown below)
  - For perspective, some examples of data variety shown on right
  - Plotted additionally with each PLT channel is BCM1F and HF data – 2 other detectors in CMS responsible for lumi measurements
- Example of channel selection tool is shown on next slide





Example Channel Verification (8675)

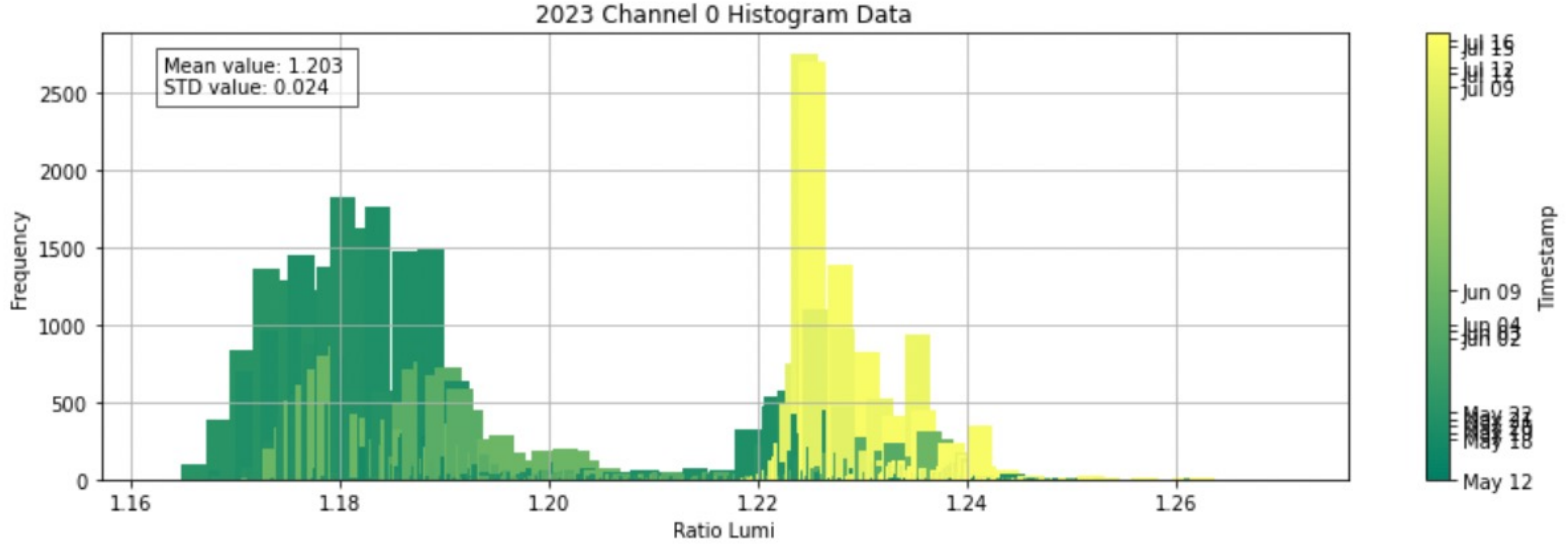


```python
reference_dt = dt_data[0]

percent = 0.25
levelshift = 400
window = 2

bad_channels = [6, 8, 9]

for channel in channels:
    if channel in bad_channels:
        print(f"Skipping bad channel {channel}. \n")
        continue

    select_channel = channel

    if not (len(reference) == len(ch_data[select_channel]['lumi']) == len(reference - ch_data[select_channel]['lumi'])):
        print(f"There is an issue with the lengths for channel {select_channel}")
        print(f"Length of reference: {len(reference)}")
        print(f"Length of select_channel: {len(ch_data[select_channel]['lumi'])}")
        print(f"Length of channels subtracted: {len(reference - ch_data[select_channel]['lumi'])}")
        continue

    ch_data[select_channel]['lumi'].replace([np.inf, -np.inf], np.nan, inplace=True)
    ch_data[select_channel]['lumi'].dropna(inplace=True)

    difference = reference - ch_data[select_channel]['lumi']
    difference.index = reference_dt
    difference.index = pd.to_datetime(difference.index).floor('s')

    ratio = (reference / ch_data[select_channel]['lumi'])
    ratio.index = reference_dt
    ratio.index = pd.to_datetime(ratio.index).floor('s')
    ratio = ratio.replace([np.inf, -np.inf], np.nan).dropna()

    differencemean = np.mean(difference)
    ratiomean = np.mean(ratio)

    threshold_high2 = ratiomean + percent * ratiomean
    threshold_low2 = ratiomean - percent * ratiomean

    if differencemean < 0:
        threshold_high1_diff = differencemean - percent * differencemean
        threshold_low1_diff = differencemean + percent * differencemean
    else:
        threshold_high1_diff = differencemean + percent * differencemean
        threshold_low1_diff = differencemean - percent * differencemean

    try:
        channel_selection(difference, threshold_high1=threshold_high1_diff, threshold_low1=threshold_low1_diff, levelshift_c_value=levelshift, levelshift_window=window, channel_number=select_channel,
                          df2=ratio, threshold_high2=threshold_high2, threshold_low2=threshold_low2,
                          title1=f"Reference - {select_channel} Anomaly Detection, Percent = +- {percent * 100}%", title2=f"Reference/{select_channel} Ratio Anomaly Detection, Percent = +- {percent * 100}%",
                          title3=f"Reference - {select_channel} Common Anomaly Detection, Percent = +- {percent * 100}%", title4=f"Reference / {select_channel} Ratio Common Anomaly Detection, Percent = +- {percent * 100}%")
    except Exception as e:
        print(f"Channel {select_channel} failed inspection with error: {e}. \n")

print(f"{np.random.choice(happy_message)} You just verified fill number {fill_number}.")
```

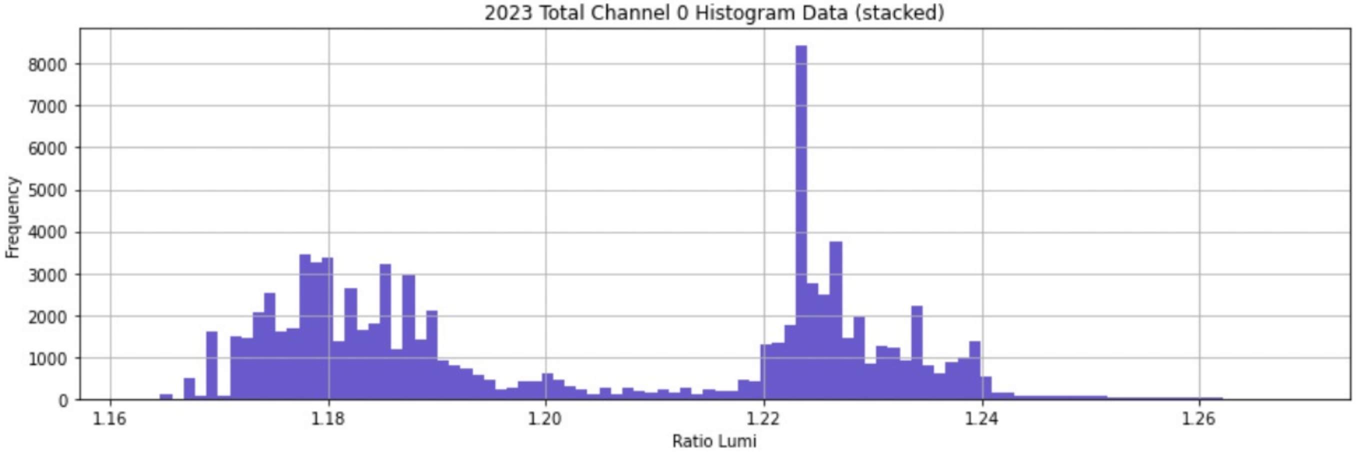
```
Length before cleaning df: 14001
Length before cleaning df: 13978
Length after cleaning df1: 14001
Length after cleaning df2: 13978
Number of common anomalies in Channel 0: 0
Channel 0 has passed inspection.

Length before cleaning df: 14001
Length before cleaning df: 13979
Length after cleaning df1: 14001
Length after cleaning df2: 13979
Number of common anomalies in Channel 1: 3
Channel 1 has passed inspection.
```

If a channel fails verification, I have another model that displays its respective plots and where the anomalies are flagged so the user can manually decide “keep” or “reject”

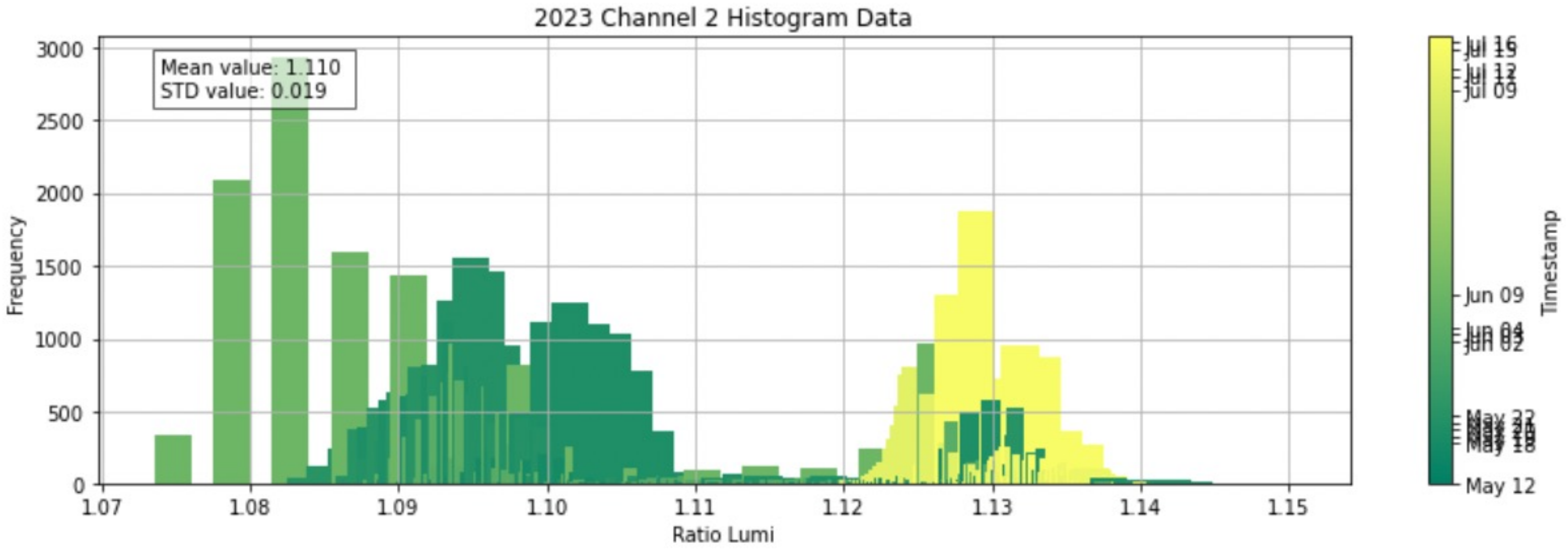
Creating Appended Histograms


2023 Channel 0 Histogram Data

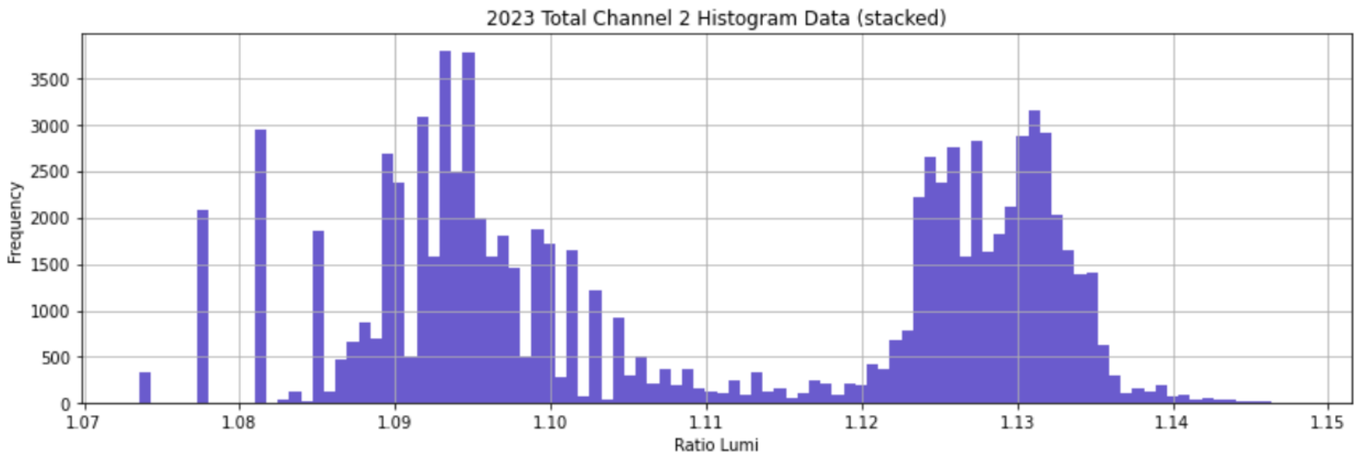
One goal we want to see is how the luminosity per channel shifts throughout the year and where the peaks are located


2023 Total Channel 0 Histogram Data (stacked)

Creating Appended Histograms



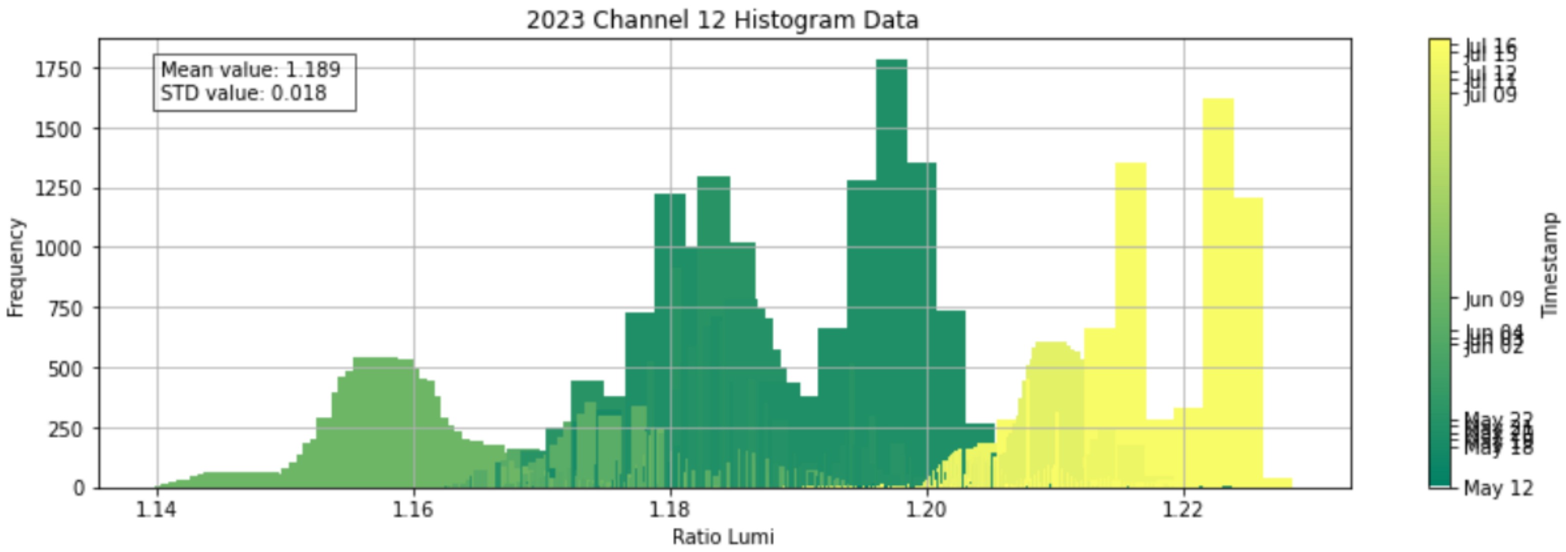
2023 Channel 2 Histogram Data

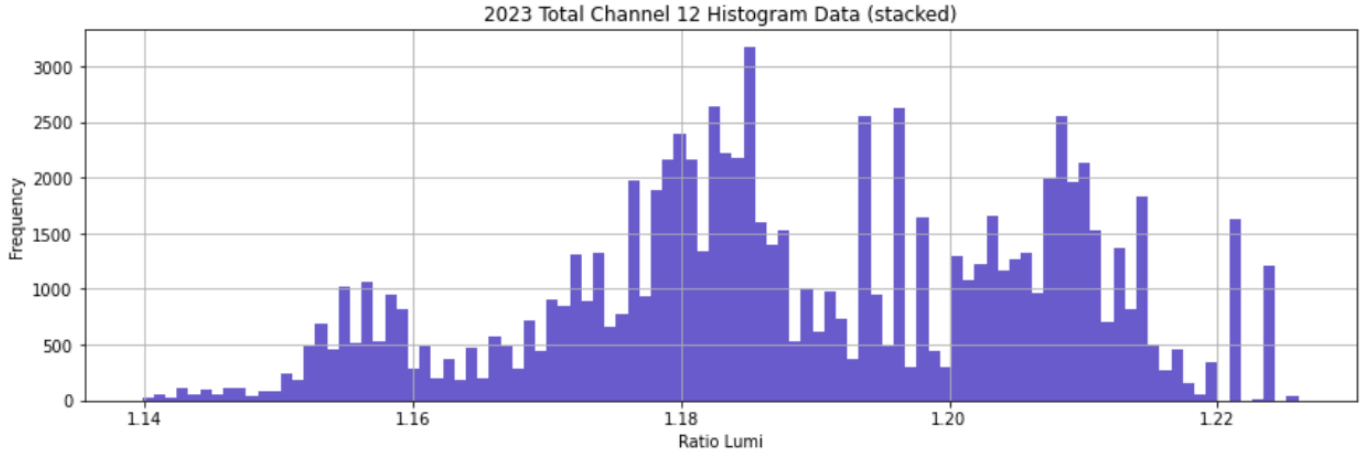


2023 Total Channel 2 Histogram Data (stacked)

One goal we want to see is how the luminosity per channel shifts throughout the year and where the peaks are located

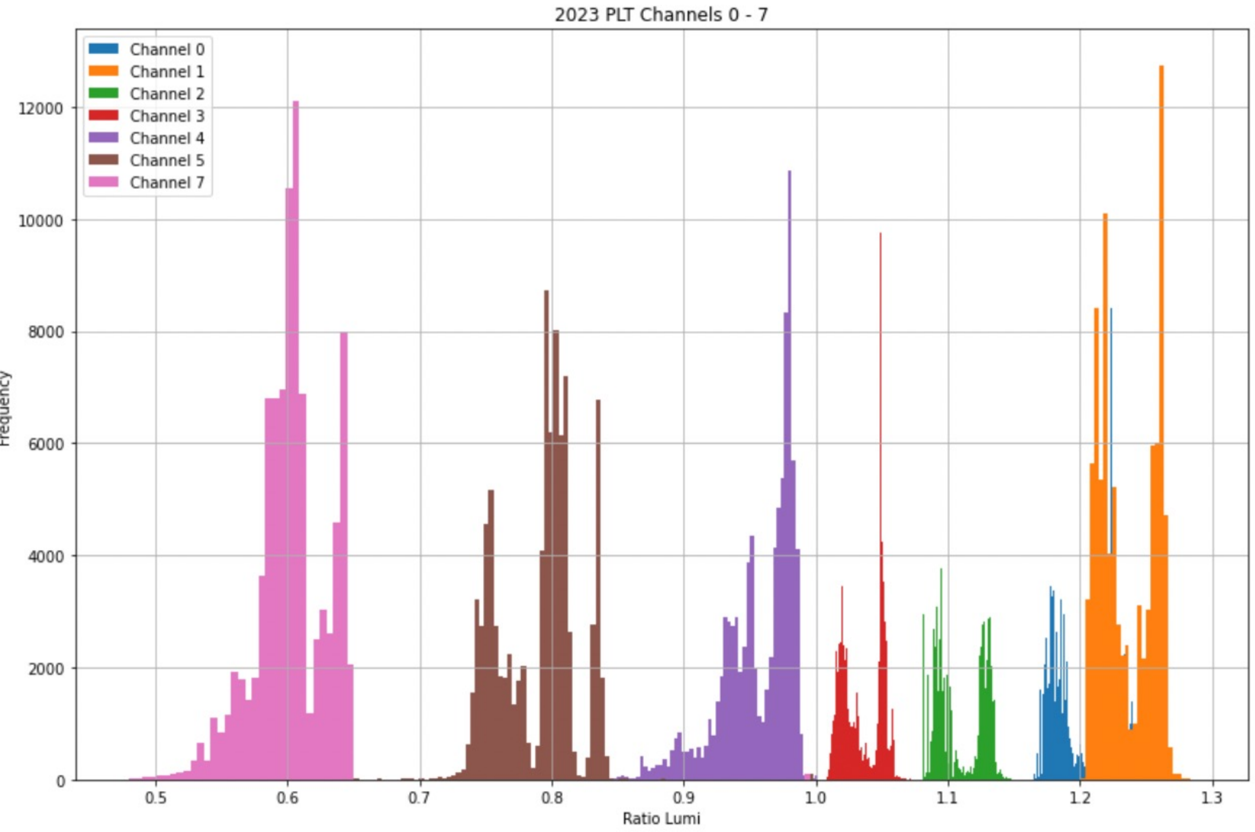
Creating Appended Histograms

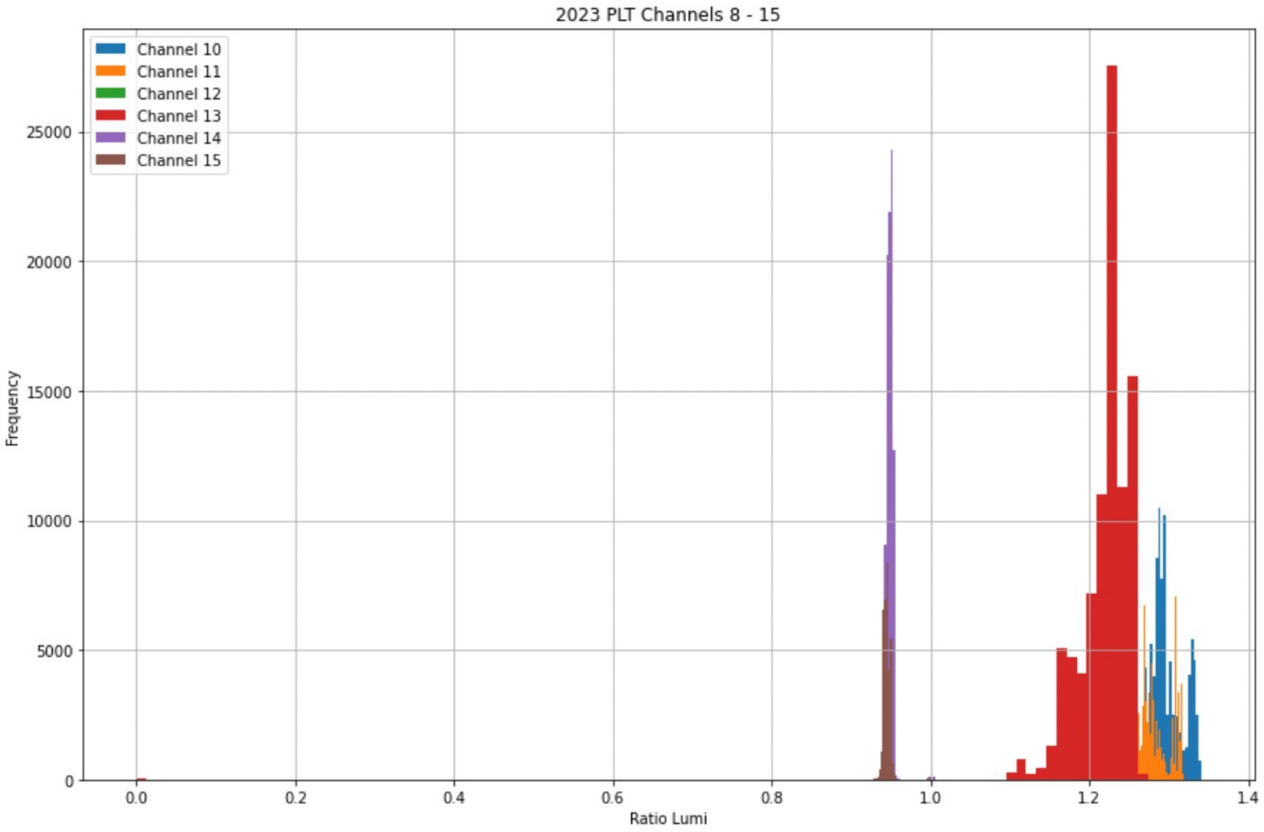




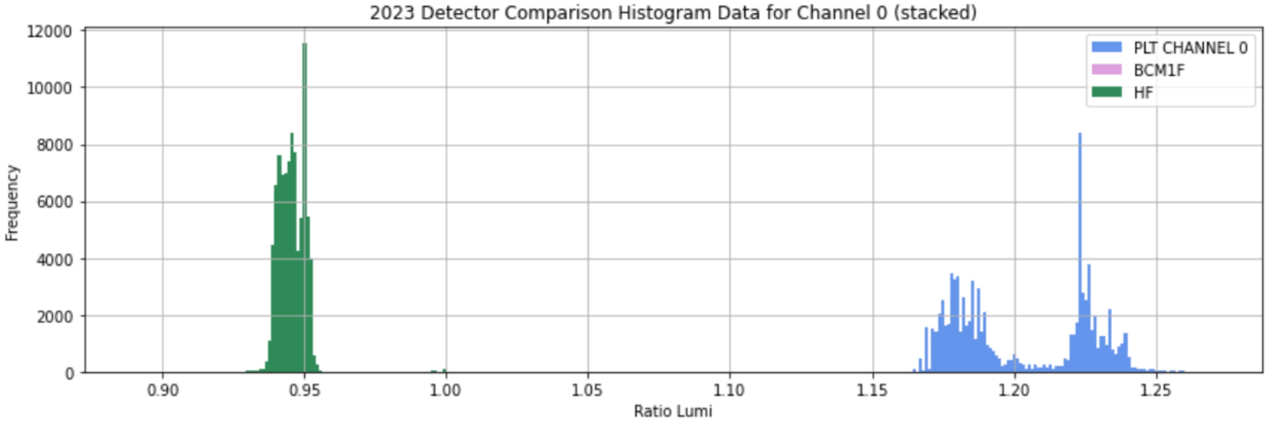
One goal we want to see is how the luminosity per channel shifts throughout the year and where the peaks are located

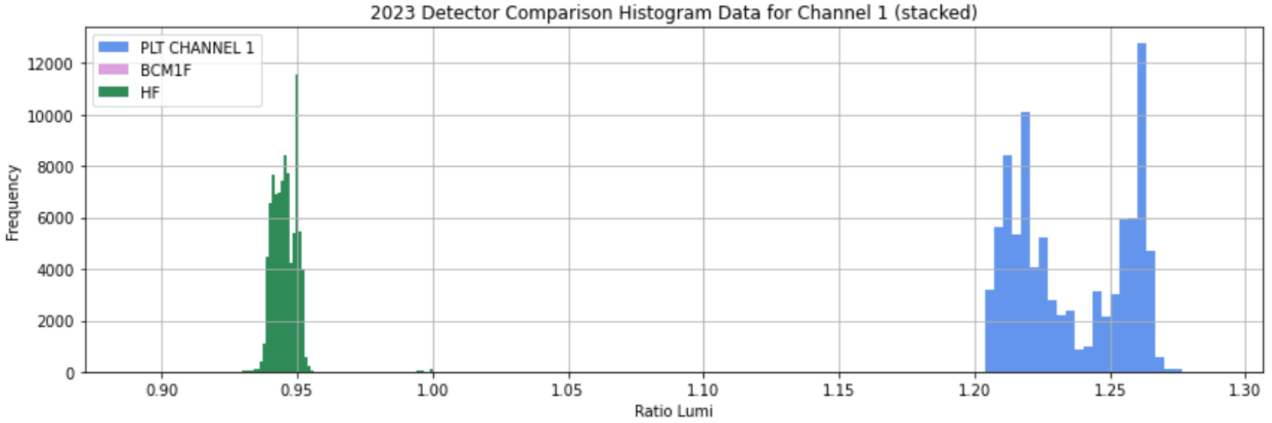
Comparing PLT Channel Distributions

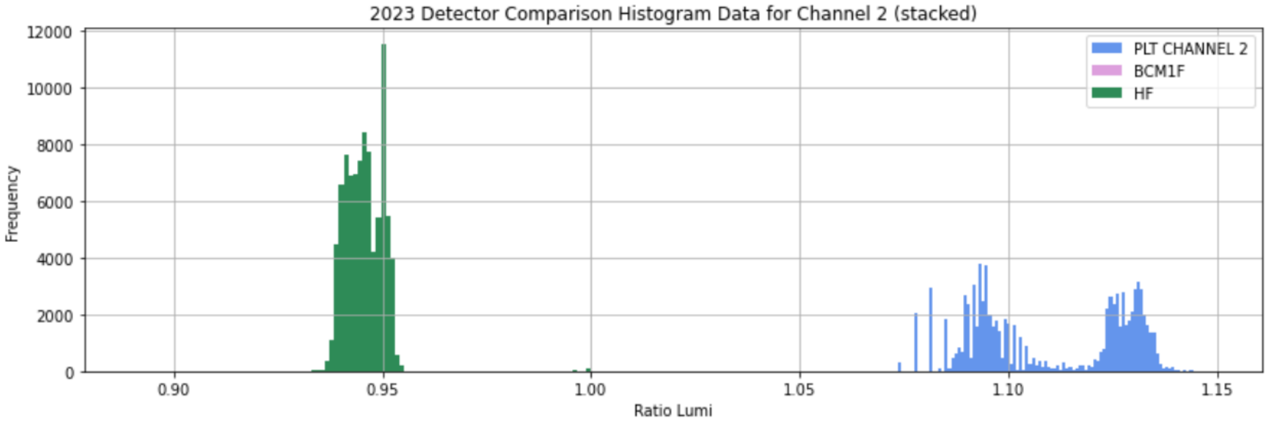


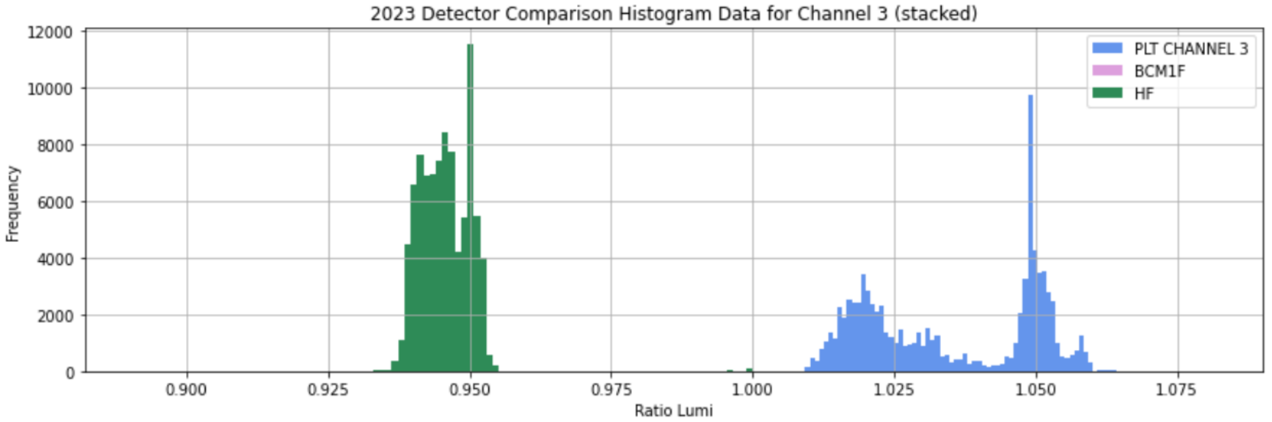


Comparing PLT to BCM1F and HF


2023 Detector Comparison Histogram Data for Channel 0 (stacked)


2023 Detector Comparison Histogram Data for Channel 1 (stacked)


2023 Detector Comparison Histogram Data for Channel 2 (stacked)


2023 Detector Comparison Histogram Data for Channel 3 (stacked)

Next Steps

- Continue performing 2023 PLT channel selection and finalize appended histogram distributions
- Develop an even faster mass applicable channel selection tool to compare results with based on appended histogram distributions
  - One example application is performing quick channel selection on the entire BCM1F detector which consists of 48 individual channels compared to PLTs 16 channels

#eurogirlsummer