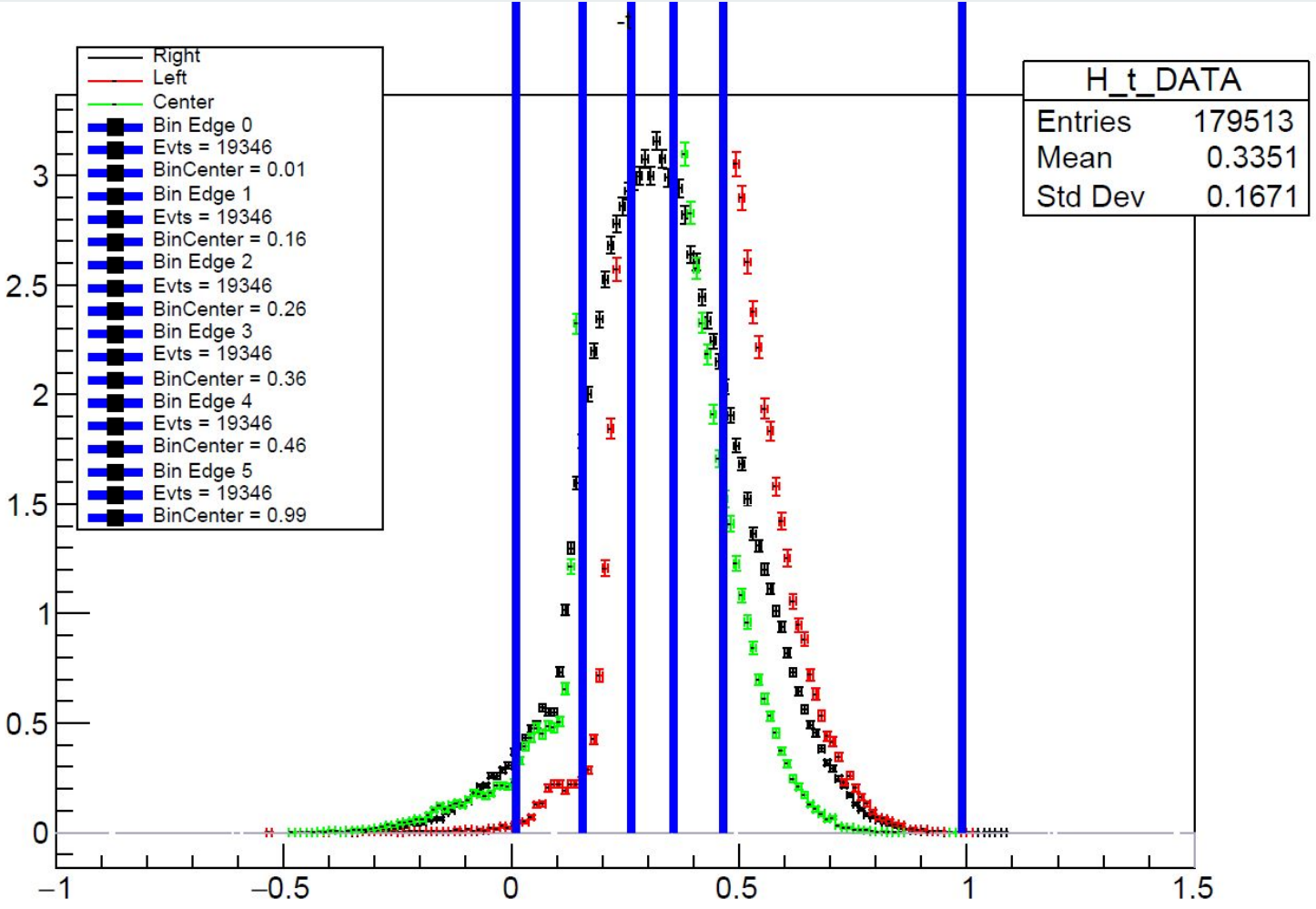
Kaon LT Status Update

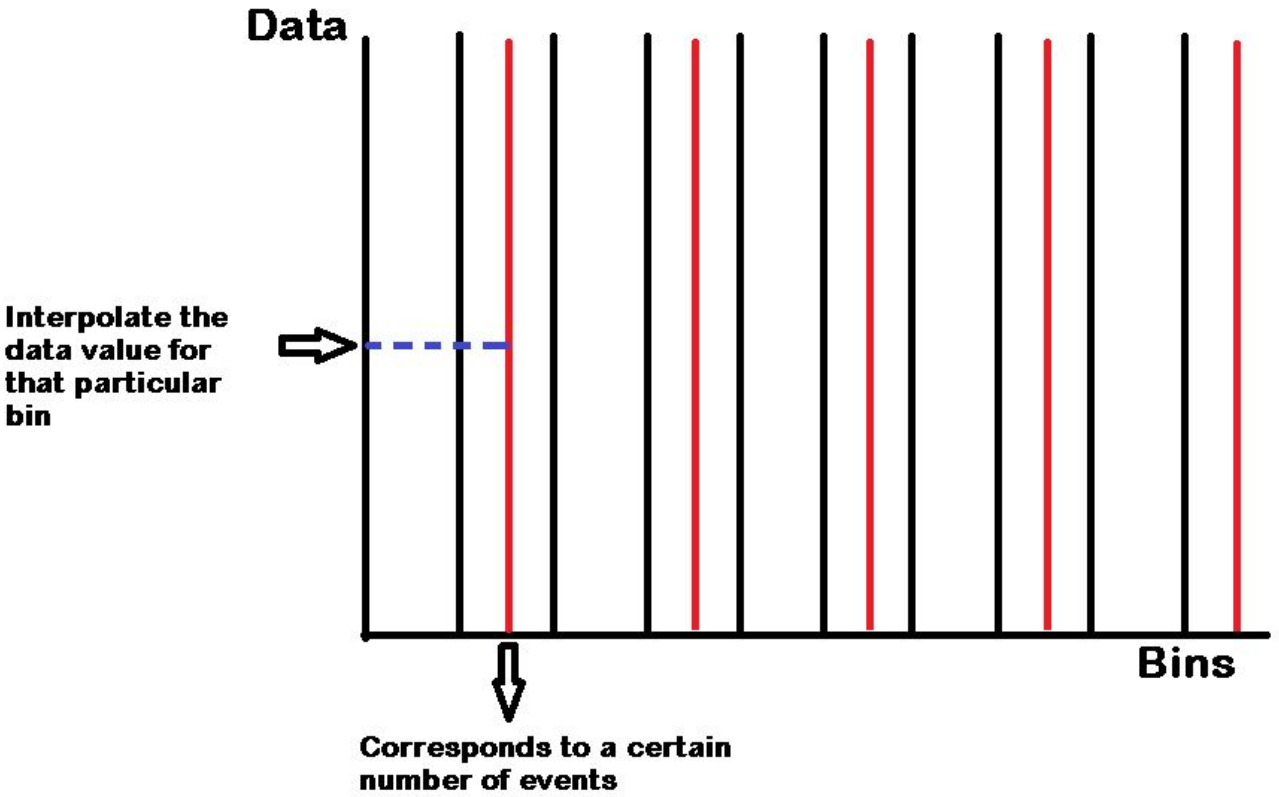
February 2nd, 2023

Richard Trotta

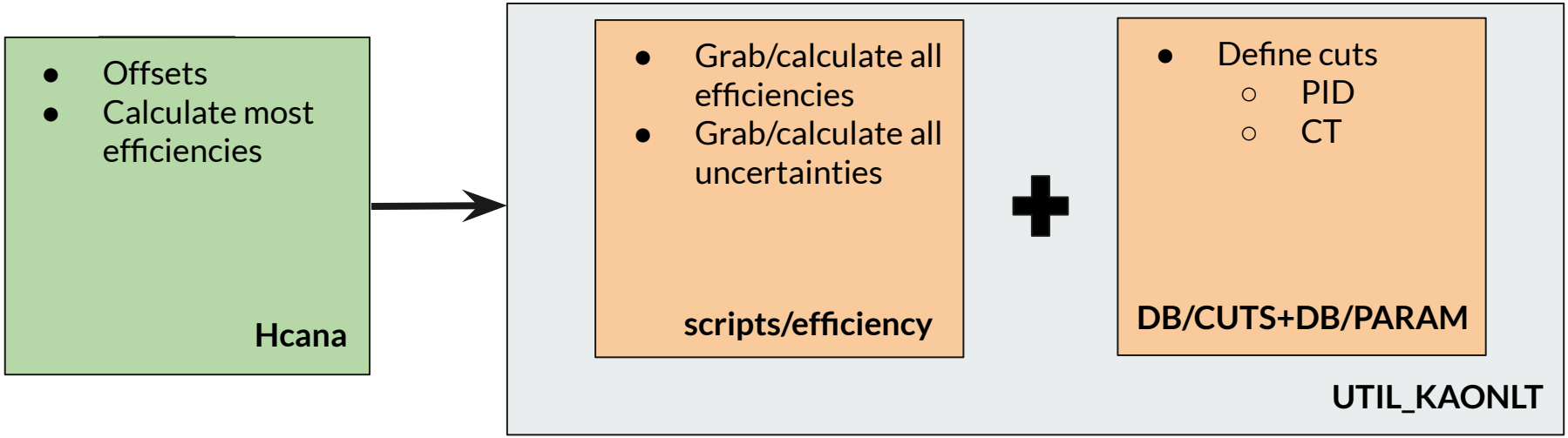

Right
Left
Center
Bin Edge 0
Evts = 19346
BinCenter = 0.01
Bin Edge 1
Evts = 19346
BinCenter = 0.16
Bin Edge 2
Evts = 19346
BinCenter = 0.26
Bin Edge 3
Evts = 19346
BinCenter = 0.36
Bin Edge 4
Evts = 19346
BinCenter = 0.46
Bin Edge 5
Evts = 19346
BinCenter = 0.99
H_t_DATA
Entries 179513
Mean 0.3351
Std Dev 0.1671
3
2.5
2
1.5
1
0.5
0
−1
−0.5
0
0.5
1
1.5

```
# Histogram takes the array data set and the bins as input
# The bins are determined by a linear interpolation (see function above)
# This returns the binned data with equal number of events per bin
# Returns...
# n -> The values of the histogram bins
# bins -> The edges of the bins
# patches -> Container of individual artists used to create the histogram or list of
# such containers if there are multiple input datasets.
n, bins, patches = plt.hist(H_t_BinTest, histedges_equalN(H_t_BinTest, NumtBins))
```

```python
def histedges_equalN(x, nbin):
    # Grab number of events in array
    npt = len(x)
    # One-dimensional linear interpolation for monotonically increasing sample points.
    # Returns the one-dimensional piecewise linear interpolant to a function with given
    # discrete data points (xp, fp), evaluated at x.
    #
    # np.interp(x, xp, fp)
    # x -> np.linspace(0, npt, nbin + 1) : The x-coordinates at which to evaluate the interpolated values
    # In this case, this is an array of evenly spaced t-bins
    # xp -> np.arange(npt) : The x-coordinates of the data points
    # In this case, this returns evenly spaced values within a given interval
    # yp -> np.sort(x) : he y-coordinates of the data points
    # In this case, this returns a sorted copy of the array
    return np.interp(np.linspace(0, npt, nbin + 1),np.arange(npt),np.sort(x))
```


Data
Interpolate the data value for that particular bin
Bins
Corresponds to a certain number of events

https://github.com/JeffersonLab/hallc_replay_lt/tree/LTSep_Analysis_2022
https://github.com/JeffersonLab/UTIL_KAONLT/tree/spring_2022



https://github.com/trottar/lt_analysis/tree/winter_2023

- Efficiencies
- Uncertainties
- PID cuts
- CT cuts

UTIL_KAONLT

- Apply PID cuts
- Apply CT cuts
- **Merges runs for each phi setting

Prod/Analysed_Prod.py

- Calculate effective charge per run

findEffectiveCharge.py

- Apply diamond cuts
- t-binning
- Generate plots for each phi setting

Prod/find_tBinRange.py

- Create inputs for LT_analysis framework

Prod/createPhysicsList.py

LTsep+LT_analysis/scripts/

** done in bash script

https://github.com/trottar/lt_analysis/tree/winter_2023