

```

## PLOTTING 1-D SIMPLE DATA ##
Simple = SimpleLoader()

## LOADING/ADDING/SUBTRACTING 1-D/REDUCED DATA FROM A FILE ##
## Loads 1-D/Reduced scans data from HDF5 file
Simple.load(config, 'filename', 'y_stream', *args, **kwargs)
## *args = comma seperated list of scans to be plotted

## Loads and sums 1-D/Reduced scans data from HDF5 file
Simple.add(config, 'filename', 'y_stream', *args, **kwargs)
## *args = comma seperated list of scans to be plotted or added and then plotted

## Loads and subtrates 1-D/Reduced scans data from HDF5 file
Simple.subtract(config, 'filename', 'y_stream', *args, **kwargs)
## *args = s1, p1 -> The data from p1 is subtracted from s1
## *args = [s1, ..., sn], [p1, ..., pn] -> The sum of p1..pn is subtracted from the sum of s1...sn

## Loads and subtract scan from all previously loaded scans
Simple.background(config, 'filename', 'y_stream', *args, **kwargs)
## *args = s1 -> The scan to be subtracted from all previous load/add/subtract actions
## *args = [s1, ..., sn] -> The sum of scans s1..sn to be subtracted from all previous load/add/subtract

## REQUIRED VARIABLES ##
## config = RIXS -> RIXS Endstation
## config = RSXS -> RSXS Endstation
## filename = hdf5 filename -> extension not required
## y_stream -> y-axis values, any mne or list from documentation
## NOTE: Simple math allowed with xes_stream with constants and variables, i.e. +, -, /, *

## NOTES ON Y STREAMS ##
## The dimension of the y_stream needs to be 1
## The axis reduction for 2-D data types is y_stream[min:max] reducing dimension to 1
## The axis reduction for 3-D data types is y_stream[{min1:max1}, {min2:max2}] reducing dimension to 1

## **kwargs ##
## norm = True -> Scales the data such that its range is 0 to 1.
## twin_y = True -> Adds these plots to a secondary scale
## yoffset = [(S1,P1), ..., (SN,PN)] -> Adjusts y-axis scale to map SN to PN
## ycoffset = value -> Shifts y-axis scale by a constant value

## SET RANGE OF X OR Y VALUES ##
Simple.xlim(min, max)
Simple.ylim(min, max)

## PLOTTING SCAN DATA ##
Simple.plot(**kwargs)

## **kwargs ##
## title = 'New Title of plot' -> Replaces default title with user defined
## xlabel = 'x-axis label' -> Replaces default x-axis label with user defined
## ylabel = 'y-axis label' -> Replaces default y-axis label with user defined
## ylabel_right = 'right y-axis label' -> Replaces default right y-axis label with user defined
## plot_height = value -> The plot height in points, default is 600
## plot_width = value -> The plot width in points, default is 900
## norm = True -> Normalizes all the data between 0 and 1
## waterfall = offset -> Normalizes as above and shifts each by the offset

## EXPORTING PLOT DATA ##
Simple.export('filename', **kwargs)

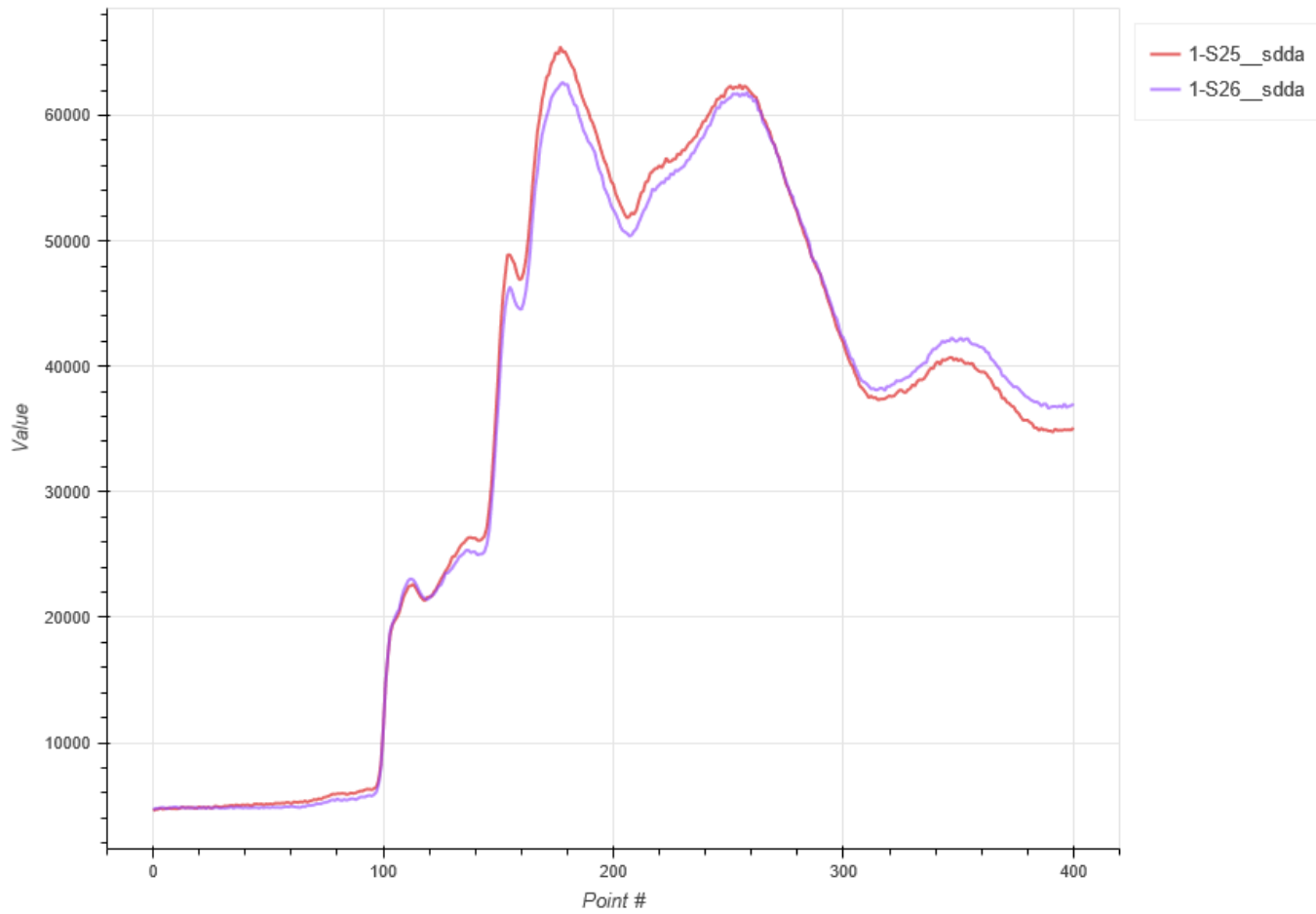
# REQUIRED VARIABLES ##
## filename = filename to be used for ASCII file, do not add extension
## NOTE: Data is exported as it displayed, only options in plotting methods are ignored.

## **kwargs ##
## split_files = True -> Saves each data stream with number appended to the filename

```

```
## Loading a series of scans
```

```
Simple = SimpleLoader()  
Simple.load(RIXS, 'HDF5_Notebook', 'sdda', 25, 26)  
Simple.plot()  
Simple.export('Simple_Data')
```



```
## Adding a series of scans
```

```
Simple = SimpleLoader()  
Simple.add(RIXS, 'HDF5_Notebook', 'sdda', 25, 26)  
Simple.plot()  
Simple.export('Simple_Add')
```

