```
## PLOTING 3-D GENERAL DATA ##
Stack = StackLoader()

## LOADING/ADDING/SUBSTRACTING 3-D DATA FROM A FILE ##
## Loads 3-D scans data from HDF5 file
Stack.load(config,'filename', 'scan_stream', 'detector', arg, **kwargs)
## args = scan number to be loaded

## Loads and sums 3-D scans data from HDF5 file
Stack.add(config,'filename', 'scan_stream', 'detector', *args, **kwargs)
## *args = comma seperated list of scans to be plotted or added and then plotted

## Loads and subtracts 3-D scans data from HDF5 file
Stack.subtract(config,'filename', 'scan_stream', 'detector', *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


## REQUIRED VARIABLES ##
## config = RIXS        -> RIXS Endstation
## config = RSXS        -> RSXS Endstation
## filename = hdf5 file -> Extension .h5 not needed
## scan_stream          -> index values, any mne or list from documentation
## detector             -> alias to image stream, needs to be stack of images
## NOTE: Simple math allowed with xes_stream with contstants and variables, i.e. +, -, /, *

## *kwargs options ##
## norm = True                      -> Scales the data such that its range is 0 to 1.
## xoffset = [(S1,P1),...,(SN,PN)] -> Adjusts x-axis scale to map SN to PN
## xcoffset = value                 -> Shifts x-axis scale by a constant value
## yoffset = [(S1,P1),...,(SN,PN)] -> Adjusts y-axis scale to map SN to PN
## ycoffset = value                 -> Shifts y-axis scale by a constant value
## grid_x = [start,stop,delta]      -> Change x-axis grid to be uniform
## grid_y = [start,stop,delta]      -> Change y-axis grid to be uniform

# PLOTTING SCAN DATA ##
Stack.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
## 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

## EXPORTING PLOT DATA ##
Stack.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

## EXPORTING TO MOVIE ##
Stack.movie('filename', **kwargs)

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

## **kwargs ##
## interval = value  -> Duration of each frame ms
## aspect = fraction -> Ratio of vertical over horizontal
## xlim = (min, max) -> Sets the x-range of movie exported
## ylim = (min, max) -> Sets the y-range of movie exported
```

```
## Example of plotting stack
Stack = StackLoader()
Stack.load(RSXS, 'LNSCO110b', 'epoch','mcpIMG',586)
Stack.plot(norm = True)
Stack.movie('Movie_Scale', aspect = 0.25, xlim = (122,133), interval = 100)
Stack.export('Movie_ASCII')
```