1. Identify “weird” non-look looks.
   1. Consider STD
   2. Graph points for each valid / non-valid points
   3. Max Distance threshold

What I did:

Store the -1 from the original xValues (stored as withNegativeOne in inter() method)

Compared it with the xValues (in main under interpolateorno)

Pruned both into newxValues and newxValuesNegativeOnes

The corresponding values in newxValues should be the ones that are interpolated compared to the newxValuesNegativeOnes which should contain the -1 values. The reason for pruning (np.ma.compressed) is to take out all values where there are duplicate time values.

In calucation and calculation2, the corresponding -1 values are checked in “previous1” (within calculation method) (which is newxValuesNegativeOnes passed in as a parameter) when evaluating the values of the interpolated values which is in “xValues” (within calculation2 method) or (newxValues passed into parameter).

The number of missing -1’s are evaluated against total data points between each stage of “look” that is classified by slope and other constraints passed in from the user.

The pruning can be adjusted by user under the user input constraint in the beginning of the method :

Give a percent threshold for how much of data can be missing (0-100)%

Percentage = user input

X = The number of -1

Y = Total number of data points between each look

samplePercentage = x/y\*100

**samplePercentage < percentage**

**April 14 2017**

Save data for each file into CSV

Generate plot images for each participant

1. Found that “offset” is not same for each participant by looking at each of the graphs
   1. As a solution, I think it is better to have the offset be a simple 1 standard deviation from the norm

Generate standard deviation as offset for each file