The code uses the first and second derivatives (jerk and snap) to identify the regions of interest. This script reads through a csv file. It filters data so it is faster to scan and it cleans the noise (moving average and kalman filters). It focuses on the Z axis to detect falls and calculates the first and second derivatives. It creates a window to scan this dataset. The window size is 'window\_size' data points and the window is advancing every 'step\_size' datapoints. It calculates the standard deviation over a specified window size with a specified step size. It plots all the graphs. It then identifies the ‘Regions of Interest’ in the data based on threshold criteria applied to the standard deviation values (AccZ\_sd) or threshols for significant jerk or snap. It filters out regions where the standard deviation is greater than or equal to the threshold value (threshold). Within each of these regions, it calculates the maximum acceleration on each of the three axes, namely, X, Y and Z. Based on these calculations, it provides: • The squared root (sqrt) of the sum of the squares of each max acceleration on each horizontal axis (Acc\_X, Acc\_Y) for the successful attempt • The squared root (sqrt) of the sum of the squares of each max acceleration on the X and Y axes only for the successful attempt (this has a better R^2 value in the regression) • The squared root (sqrt) of the sum of the squares of each max acceleration on each axis (Acc\_X, Acc\_Y, Acc\_Z) for each of the unsuccessful attempts. It has the option of using the min accel on the Z axis since the horse is falling (negative acceleration on Z axis) (perhaps this provides a more accurate reflection of what is actually happening). Then it calculates the recovery score based on whether there was only one single and successful attempt or there were more than one unsuccessful attempts to stand It then saves it onto a CSV file.