Introduction/Notes:

For simplicity and time restraint, I used the data of user 09 for this project. I have also confirmed with a TA that I can use only user 09's data instead of all users; as well as R instead of Python/MATLAB.

For the accompanying R script to work, the right packages has to be available (or installed) and the path to files has to be edited accordingly.

Phase 1:

As required I took the Ground Data and multiplied it by 100/30 to get the start and end of the row numbers of "eating" part of the datasets. Any row that was not "eating" was then assigned to "non-eating". This way I initially got four arrays of: 2896 rows for eating(fork), 3299 rows for eating(spoon), 25323 for non-eating (fork) and 34576 for non-eating(spoon). I merged the eating datasets to form and an array containing 6195 observations. I took the first 2896 rows and 3299 rows of the respective non-eating arrays and made a final non-eating dataset.

Finally I had two datasets of 6195 observations each, which are attached as "eating09.csv" and "non_eating09.csv".

Please see the attached R script "Project1.R" under "PHASE 1" for details on the code and comments as appropriate.

Phase 2:

There is an assumption that data is not randomly scattered but consist of subspaces that effectively represent the data [1]. Feature extraction techniques can be used to transform or project a space of many dimension into an alternative representation of a fewer dimensions [2-3]. This involves creating new reduced set of features that summarizes most of the information in the original set of features. Feature extraction has the advantage of reducing overfitting, improving accuracy, speeding up training, improving data visualization and enhancing the explanation of models [2].

For this phase of the project, the five feature extraction methods I used are independent component analysis (ICA), mean, median, standard deviation (std) and max.

a) When using PCA, we aim to separate a signal into a subspace that is signal and a subspace that is essentially noise. This is done by assuming "that only the eigenvectors associated with the p largest eigenvalues represent the signal, and the remaining (M − p) eigenvalues are associated with the noise subspace" [3]. To maximize the independence between the eigenvectors, orthogonal subspaces is required. But since, the differences between signals and noise may not always be clear, orthogonal subspaces may not always be adequate to differentiate between the constituent sources in a measured signal. ICA instead thinks of the data as a mixture of signals (independent components of your data) and aims to separate (blind source separation - BSS) out the signals.

Mean and median are the two most common kinds of "averages" in statistics. Mean is what you get when you add up numbers and divide it by the sample size. Mean is what comes to mind the most once you hear the word "average".

Median is the middle value in an ordered list. They both try to summarize a dataset by giving a single representative number for the dataset.

Another very common feature reduction method is std. Unlike mean and median which measure the central tendency of the data, std measures the spread (dispersion) of the data.

A less commonly used but also important feature reduction method is max. Max is the highest value in the dataset. Finding the max helps us understand the coverage/span of our data. Depending on the study, max can assist in having a feel of the data and also enhance data visualization.

b) An intuitive classical example of problem that ICA can be applied to is the Cocktail Party Problem. In order for us to ‘pick out’ a voice from an ensemble of voices in a crowded room, we can perform ICA (a type of BSS) to recover the original sources from the observed mixture. In our case it is to ‘pick out” eating signals and non-eating signals from a mixture that includes both and a mixing matrix.

For the same problem, the simpler statistics like mean, median, std and max can also reduce the data to representative values, which describes the central tendency, coverage or variability in the data. With the other four the intuition is to be able to see a representation that is distinct between eating and non-eating datasets.

Specifically the mean should show an average representation of the datasets that is different. The median should show what the middle “weight” of the two datasets look like and ideally should be distinguishable. The std should show how much more or less spread the eating signals. The max should show us if the eating signals are more or less extreme than the non-eating signals. These features should also show differences in pattern over time between the two datasets.

c) The code for extracting the features are in the “Project1.R” file under “PHASE 2”. Comments are also provided to provide useful details to the code.

d) Code for generating the plots are also in “PHASE 2” and the generated plots are attached as pdf files. The pdf files all start with the prefix “Phase2”. There are 13 such pdf files.

Eight of them are plots of each of the raw EMG data. The “Phase2raw1.pdf” to “Phase2raw8.pdf” files represent the EMG1 to EMG8 data respectively. The intent of these plots is to observe the difference between eating and not eating in the raw data and then compare the plots of the reduced features.

The remaining five pdf files’ names are self-explanatory as to which feature is in the plot.

e) Overall my initial intuition was that the eating and noneating signals will have distinctive patterns across all the features. I thought there will better distinction between the plots of the raw data and the features. However the relevance of the features are not completely irrelevant, in that each of these five feature plots provide a representation plot “representative” of the eight raw plots. Also the non-eating signals across all the plots were more dispersed compared to the eating signals; and this was clearly visible carried into the feature plots.

By eyeballing the individual raw EMG plots, one can see that both eating and non-eating signals concentrated between -30 to 30 for EMG1, 10 to 10 for EMG2 and EMG4 and -5 to 5 for the other EMGs. In addition the non-eating signals were more spread out than the eating signals. The mean plot as expected concentrated the signals in a representative “average” that ranged roughly between -7 and 7 for eating and -9 to 9 for non-eating.

The standard deviation plot concentrated the data between 0 and roughly 10, with the deviation of non-eating being slightly higher. This also met my expectation of this specific plot showing a representation spread of the data. Together with the mean, the std could very well represent the 8 raw data.

The median also represented the “middle point” of the eight raw variables by concentrating the signals between roughly -4 and 4. This range will easily represent all the 8 datasets, so I also was satisfied my intuition in terms of representation of the data.

The maxima showed that while the eating signal is bound to less than 115, the non-eating could reach up to 125. The plot concentrated the plot around 20. What is interesting is that the four features could be seen to be a good representation of the data individually and collaboratively. The features also seems to give the same information in all the raw data that; which is that the non-eating signals are higher in dimension compared to the eating signals.

The ICA feature was extracted from the mean of the raw EMGs. I was expecting it to reproduce the mean data. While it extracted a similar signal that concentrated roughly between -3 and 3 for eating signal; it was a far call from mean.

Phase 3:

See "PHASE 3" of the Project1.R file for code.

Subtask1: As required in the rubrics for this subtask, I have provided two arrays – "pcaInput_eating.csv" and "pcaInput_non_eating.csv" files. The code is under "Subtask1" of PHASE 3.

Subtask2: The code for PCA is under "Subtask2.1" of PHASE 3. The code for getting eigenvectors is under "Subtask2.2" of PHASE 3 and generated eigenvector array is attached as "Phase3eigenvectors.csv". The code for generating spider plot of eigenvectors is under "Subtask2.3" of PHASE 3 and generated graph is attached as "Phase3_SpiderPlot.pdf".

Subtask3: From the spider plot it can be deduced that eigenvectors with top five eigenvalues corresponds to Component 1. Component 1 (eigenvectors) determine the direction of the new feature space, and the eigenvalues explain the variance of the data along the new feature axe. The eigenvectors turned out the way they did, because five new features were generated by using various combinations of the old features, which is defined by the eigenvectors. By looking at the (see code under "Subtask3:1" of PHASE 3) summary statistics, one can see that the first set of combinations explained 46% of the data variance, while the other components explained 27%, 20%, 6% and <1% of the data variance respectively. This is the reason for Component 1 having the eigenvectors with the top eigenvalues. The five modified features are generated under “Subtask3:2” of PHASE 3 and attached as “Phase3newFeatures.csv” file

Subtask4: From above it makes sense to utilize the first three components in order to be able to account for above 85% of the variability in the data. The code for generating the new feature matrix based on Components 1-3 is under “Subtask4” and the generated matrix is attached as “Phase3newFeatureMatrix.csv” file.

Subtask5: To compare the new feature matrix modified by PCA with the unmodified ones from Phase2, plots of the modified features were generated. The plots are Phase3mean.pdf, Phase3median.pdf, Phase3std.pdf, Phase3max.pdf, Phase3ica.pdf. Code for the plots are under “Subtask5” of PHASE 3. My observation is that the plots of the features modified by PCA are similar to the plots of the original features - shape. However, the magnitude of the modified ones were quite smaller. In the context of this assignment, I do not think PCA data reduction was useful. Because even after PCA extraction, there was still no clear distinction between the eating signals and the non-eating signals.

References:

1. Video lectures
2. Pier Paolo Ippolito. Feature Extraction Techniques.
   https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be
3. G.D. Clifford (2005-2008) Chapter 15 - BLIND SOURCE SEPARATION: Principal & Independent Component Analysis
   http://www.mit.edu/~gari/teaching/6.555/LECTURE_NOTES/ch15_bss.pdf