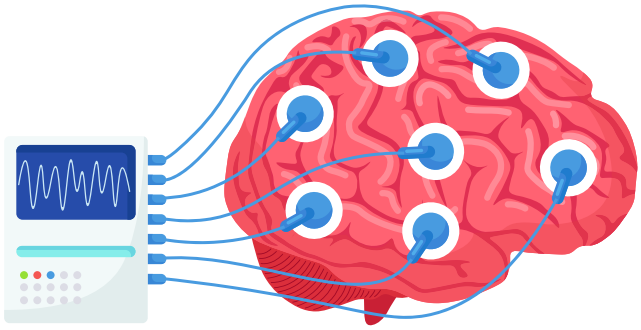
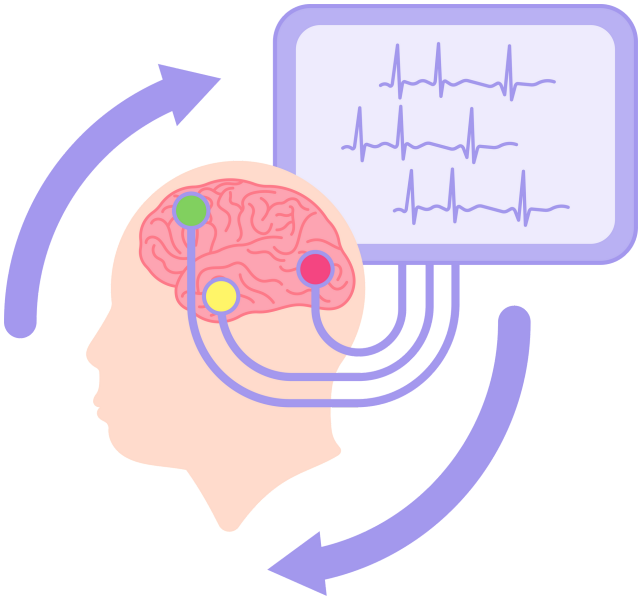




BCI COURSE

HW2: EEG Analysis

Part 1.
Multiple Choice

Problem 1

Ans：C

SNR = sqrt(n) * (Signal / Noise)
We are given that the desired SNR is 2:1, which means that SNR = 2. We are also given that the amplitude of the ERP effect (Signal) is 5 μV, and the standard deviation of the noise is 10 μV. Plugging these values into the formula, we get:
2 = sqrt(n) * (5/10) = sqrt(n) * 0.5
sqrt(n) = 4
n = 16

Problem 2

Ans：A, D, E

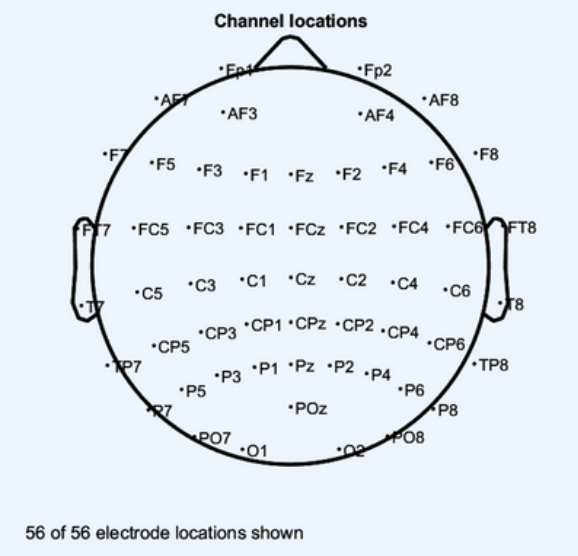
PCA and ICA are linear methods that can identify the underlying sources of variation in the data without requiring prior knowledge of the class labels. K-means clustering is a clustering method that can group similar patterns in the data without using class labels.

On the other hand, LDA (Linear Discriminant Analysis) and CSP (Common Spatial Pattern) are supervised techniques that require prior knowledge of class labels or conditions to identify the features that are most discriminative between them.

Part 2. Programming Problem

Problem 1

1 Plot 2D channel location map

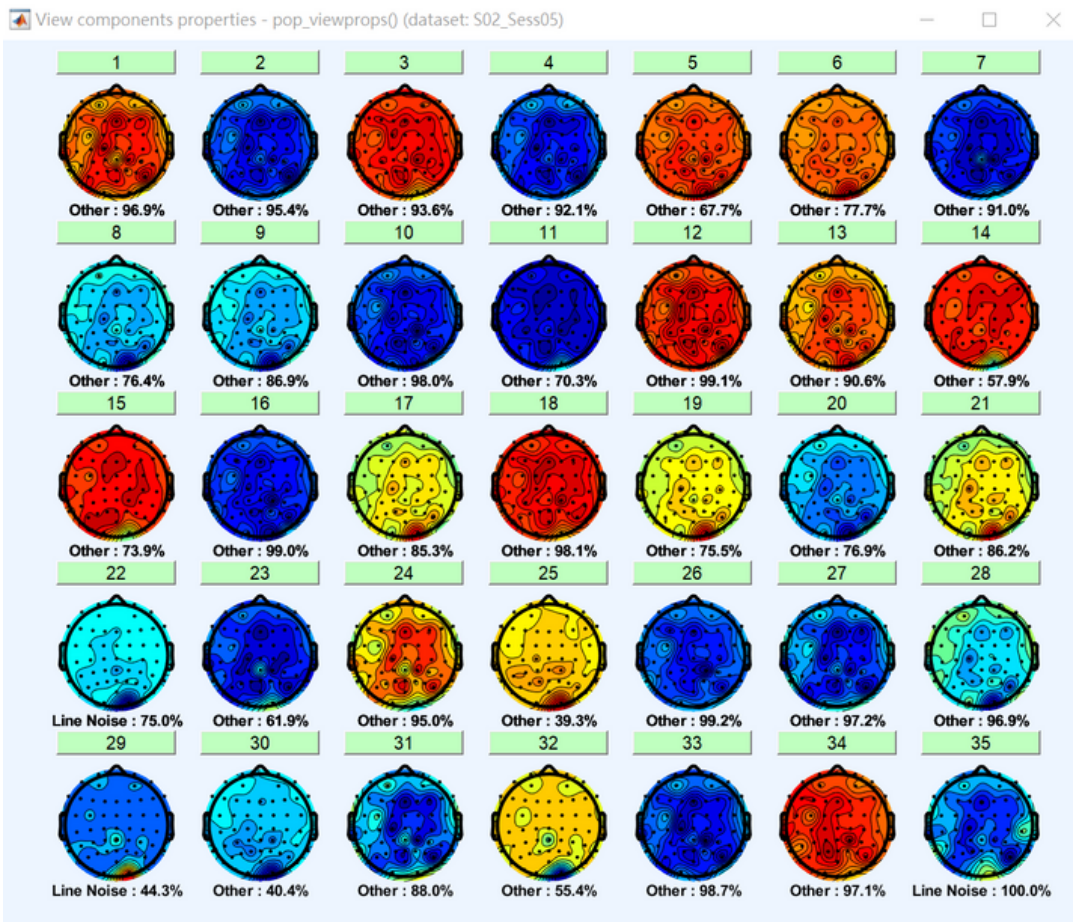


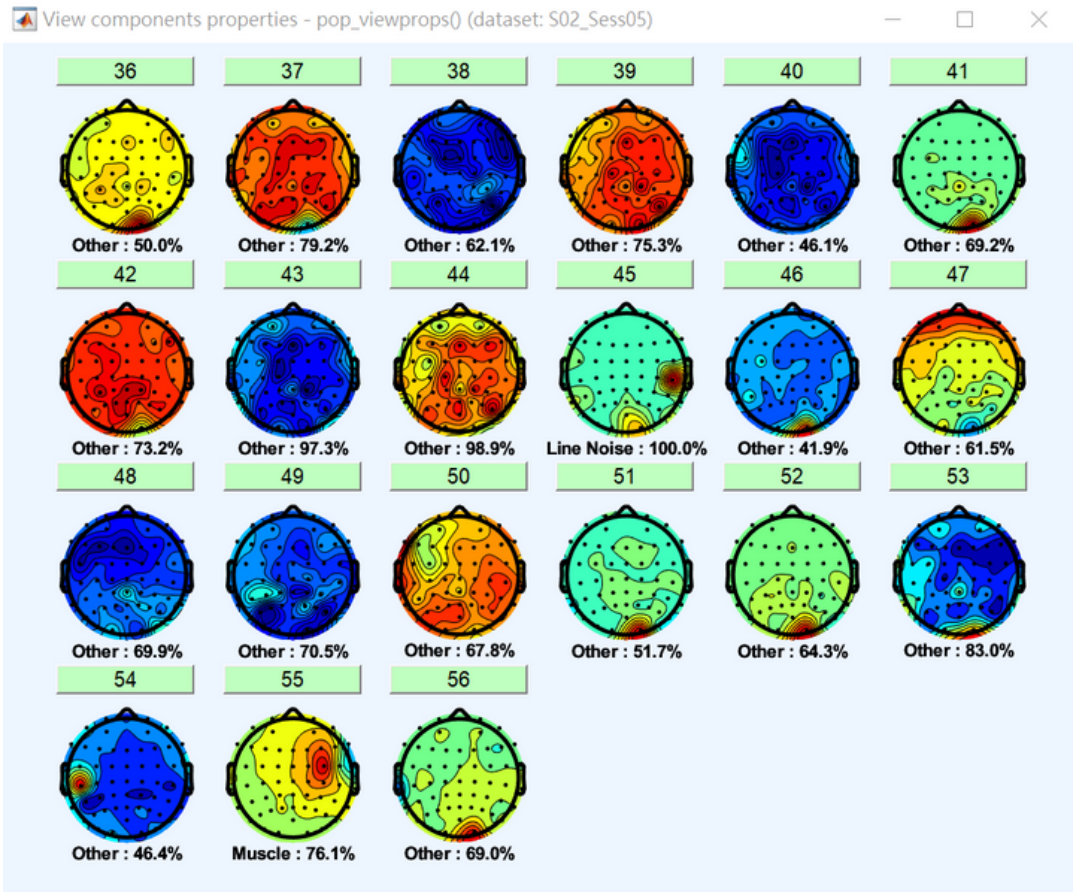
2 Run ICA and record computational time of ICA by code

```
Attempting to convert data matrix to double precision for more accurate ICA results.

Input data size [56,196001] = 56 channels, 196001 frames/nFinding 56 ICA components using extended ICA.
Kurtosis will be calculated initially every 1 blocks using 6000 data points.
Decomposing 62 frames per ICA weight ((3136)^2 = 196001 weights, Initial learning rate will be 0.001, block size 61.
Learning rate will be multiplied by 0.98 whenever angledelta >= 60 deg.
More than 32 channels: default stopping weight change 1E-7
Training will end when wchange < 1e-07 or after 512 steps.
Online bias adjustment will be used.
Removing mean of each channel ...
Final training data range: -1506.9 to 1140.2
Computing the sphering matrix...
Starting weights are the identity matrix ...
Sphering the data ...
Beginning ICA training ... first training step may be slow ...
step 1 - lrate 0.001000, wchange 46.42501498, angledelta  0.0 deg
step 2 - lrate 0.001000, wchange 4.14622881, angledelta  0.0 deg
step 3 - lrate 0.001000, wchange 1.23929244, angledelta 91.0 deg
step 4 - lrate 0.000980, wchange 1.06091816, angledelta 94.1 deg
step 5 - lrate 0.000960, wchange 1.27261990, angledelta 101.9 deg
step 6 - lrate 0.000941, wchange 0.71288510, angledelta 104.5 deg
step 7 - lrate 0.000922, wchange 0.66903507, angledelta 102.0 deg
step 8 - lrate 0.000904, wchange 0.91043450, angledelta 109.9 deg
step 380 - lrate 0.000000, wchange 0.00000031, angledelta 97.9 deg
step 381 - lrate 0.000000, wchange 0.00000033, angledelta 97.8 deg
step 382 - lrate 0.000000, wchange 0.00000030, angledelta 98.2 deg
step 383 - lrate 0.000000, wchange 0.00000031, angledelta 98.8 deg
step 384 - lrate 0.000000, wchange 0.00000026, angledelta 99.3 deg
step 385 - lrate 0.000000, wchange 0.00000024, angledelta 94.3 deg
step 386 - lrate 0.000000, wchange 0.00000022, angledelta 101.1 deg
step 387 - lrate 0.000000, wchange 0.00000021, angledelta 89.7 deg
step 388 - lrate 0.000000, wchange 0.00000018, angledelta 96.1 deg
step 389 - lrate 0.000000, wchange 0.00000016, angledelta 97.4 deg
step 390 - lrate 0.000000, wchange 0.00000017, angledelta 96.5 deg
step 391 - lrate 0.000000, wchange 0.00000015, angledelta 92.2 deg
step 392 - lrate 0.000000, wchange 0.00000013, angledelta 98.3 deg
step 393 - lrate 0.000000, wchange 0.00000010, angledelta 95.6 deg
step 394 - lrate 0.000000, wchange 0.00000011, angledelta 93.7 deg
step 395 - lrate 0.000000, wchange 0.00000012, angledelta 95.7 deg
step 396 - lrate 0.000000, wchange 0.00000011, angledelta 94.5 deg
step 397 - lrate 0.000000, wchange 0.00000010, angledelta 93.7 deg
step 398 - lrate 0.000000, wchange 0.00000009, angledelta 98.5 deg
Sorting components in descending order of mean projected variance ...
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Done.
```

Problem 1

3 Plot component maps in 2D





Problem 1

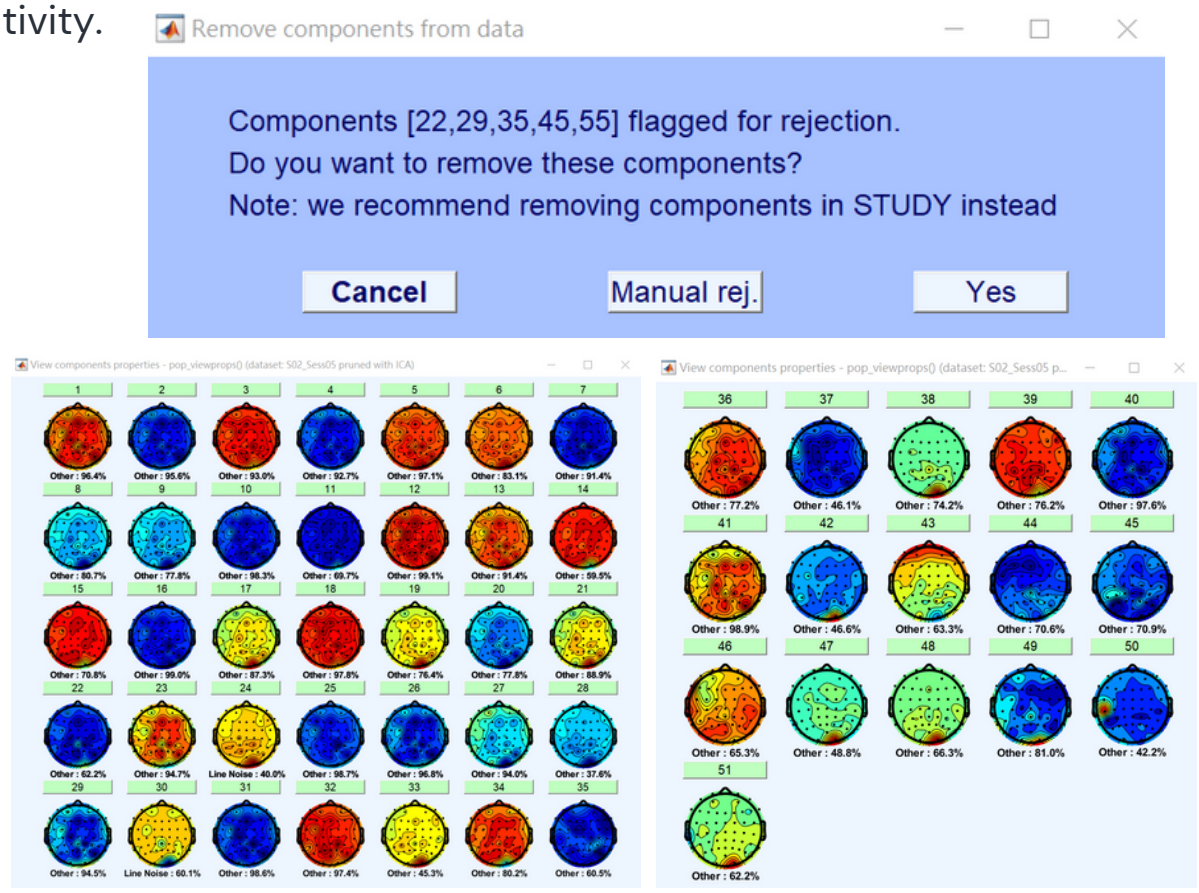
4 Indicate noise component(s) if they exist and explain the reason why you identify this component as noise or artifacts.

The noise components I reject： Muscle, line noise

I delete noise components of muscle and line noise because they are common sources of interference that can obscure the true brain electrical activity signals. Muscle activity interference mainly comes from muscle movement, such as head, eye, and facial muscle movement. This activity generates high-frequency, high-amplitude signals known as muscle noise or eye movement noise.

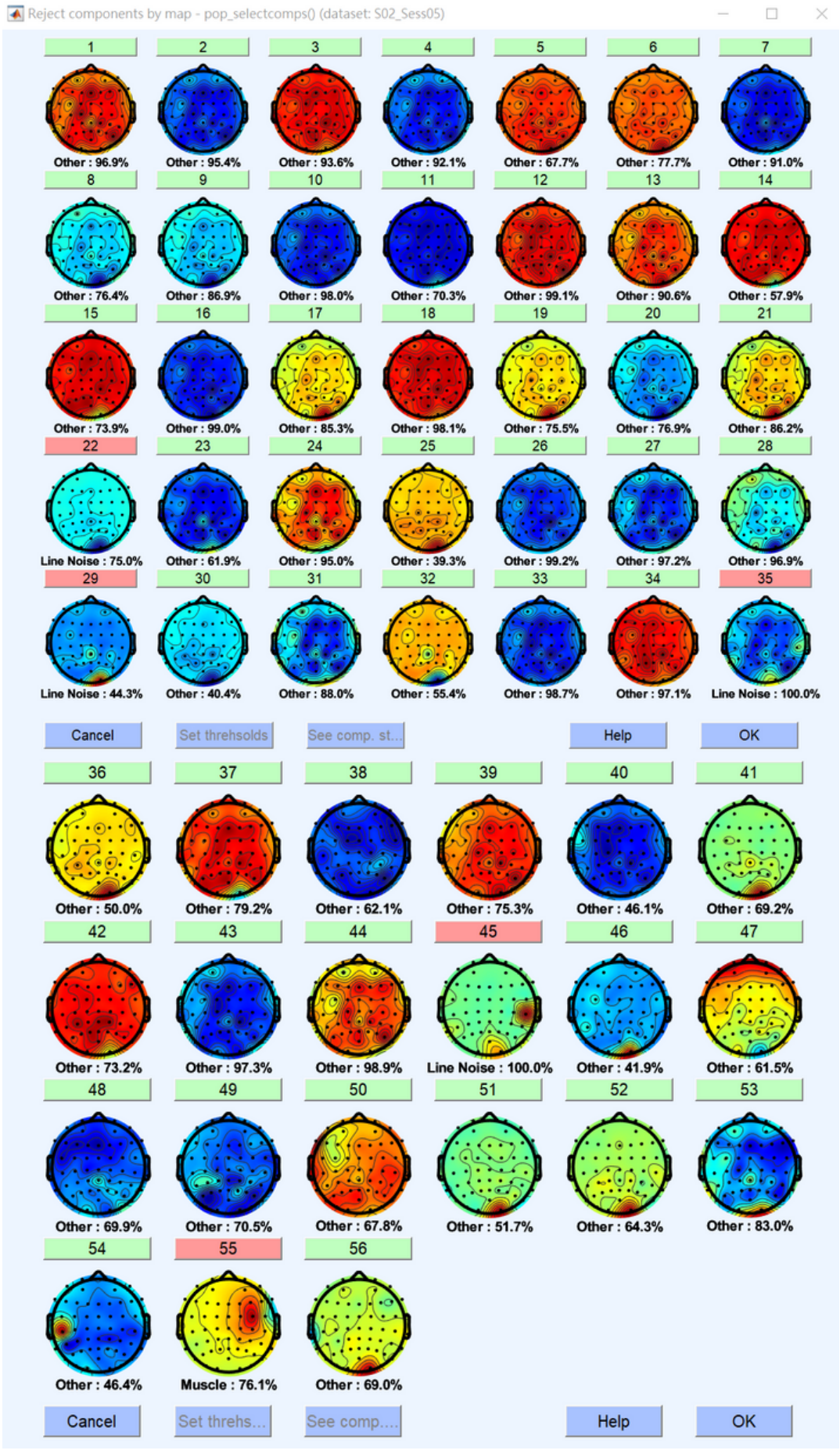
On the other hand, line noise interference is generated by electromagnetic fields from AC power networks and can produce distinct periodic waveforms in EEG signals. This type of interference usually occurs at frequencies of 50 Hz or 60 Hz, depending on the location and equipment used. So I remove it.

Removing muscle and line noise can helps improve the signal-to-noise ratio and more accurately detect and analyze brain electrical activity.



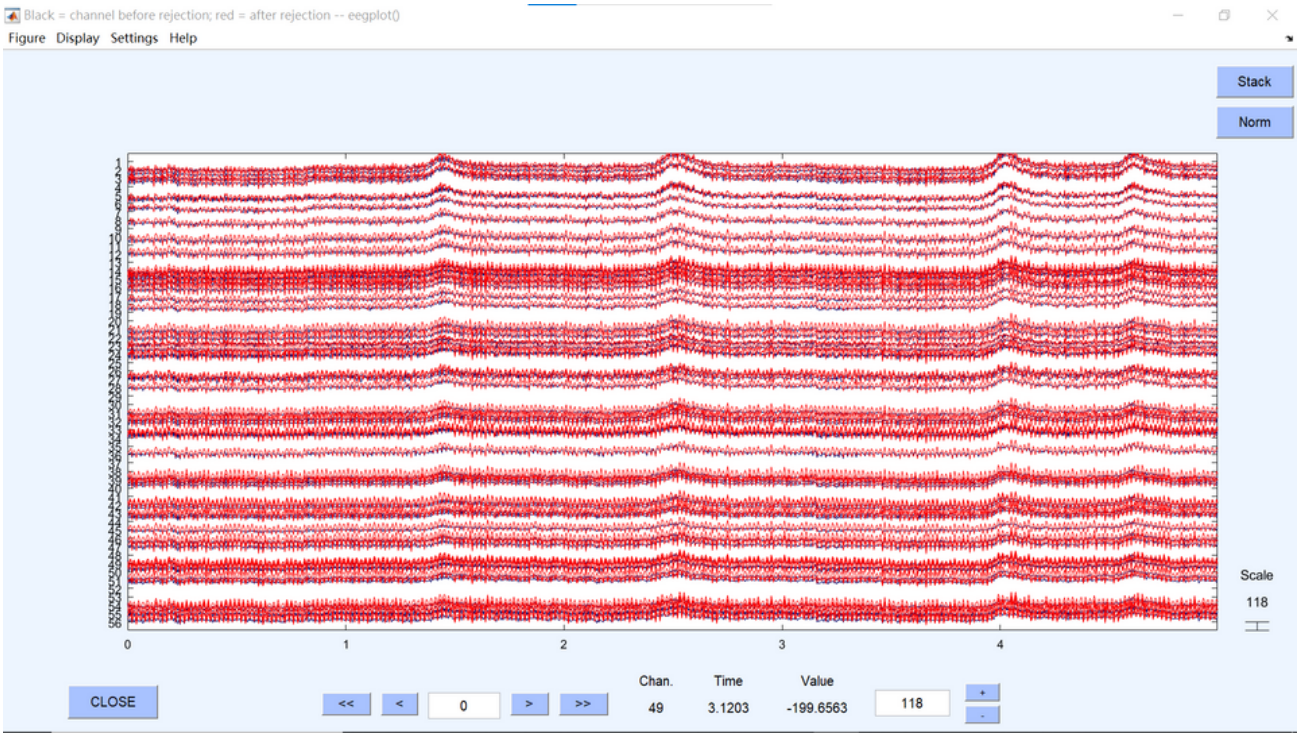
Problem 1

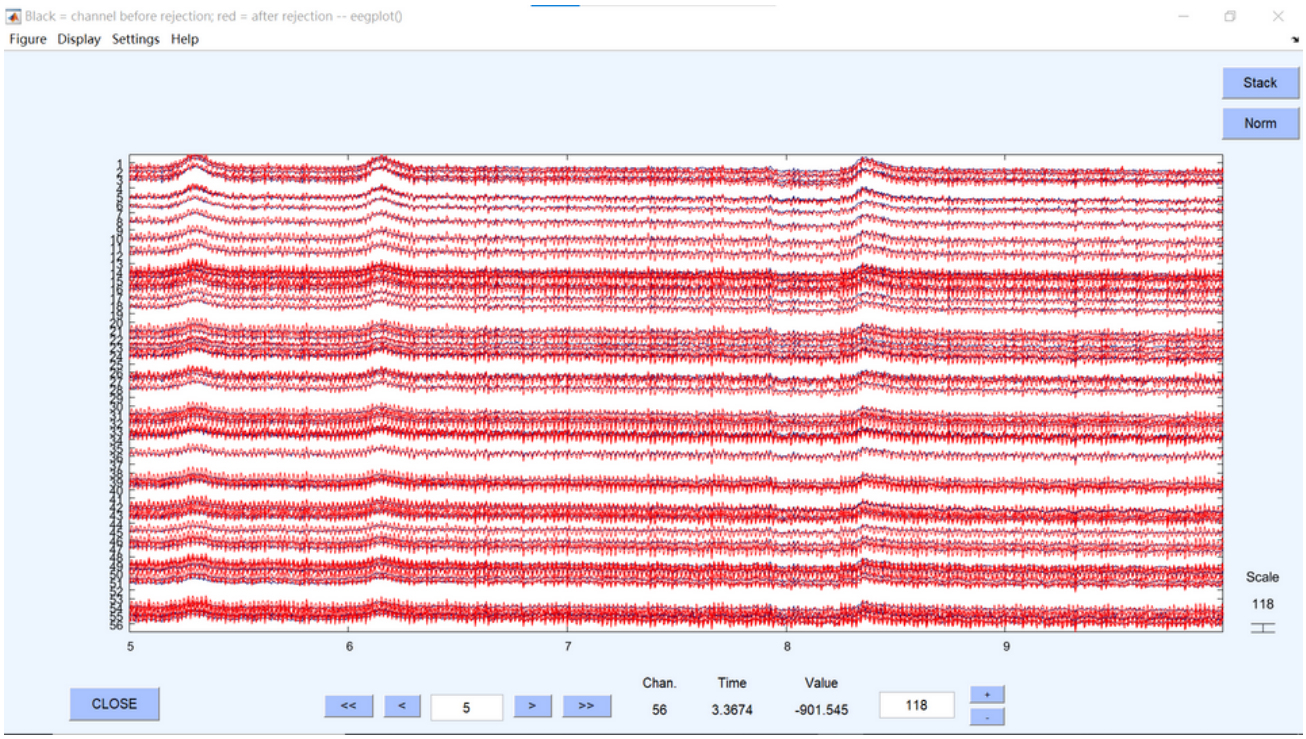
4 **Indicate noise component(s) if they exist and explain the reason why you identify this component as noise or artifacts.**



Problem 1

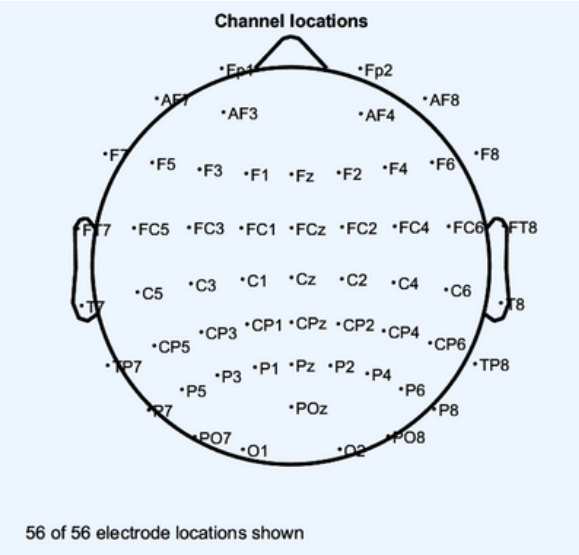
5 Plot first 10-second channel data before and after deleting noise/artifact component(s)



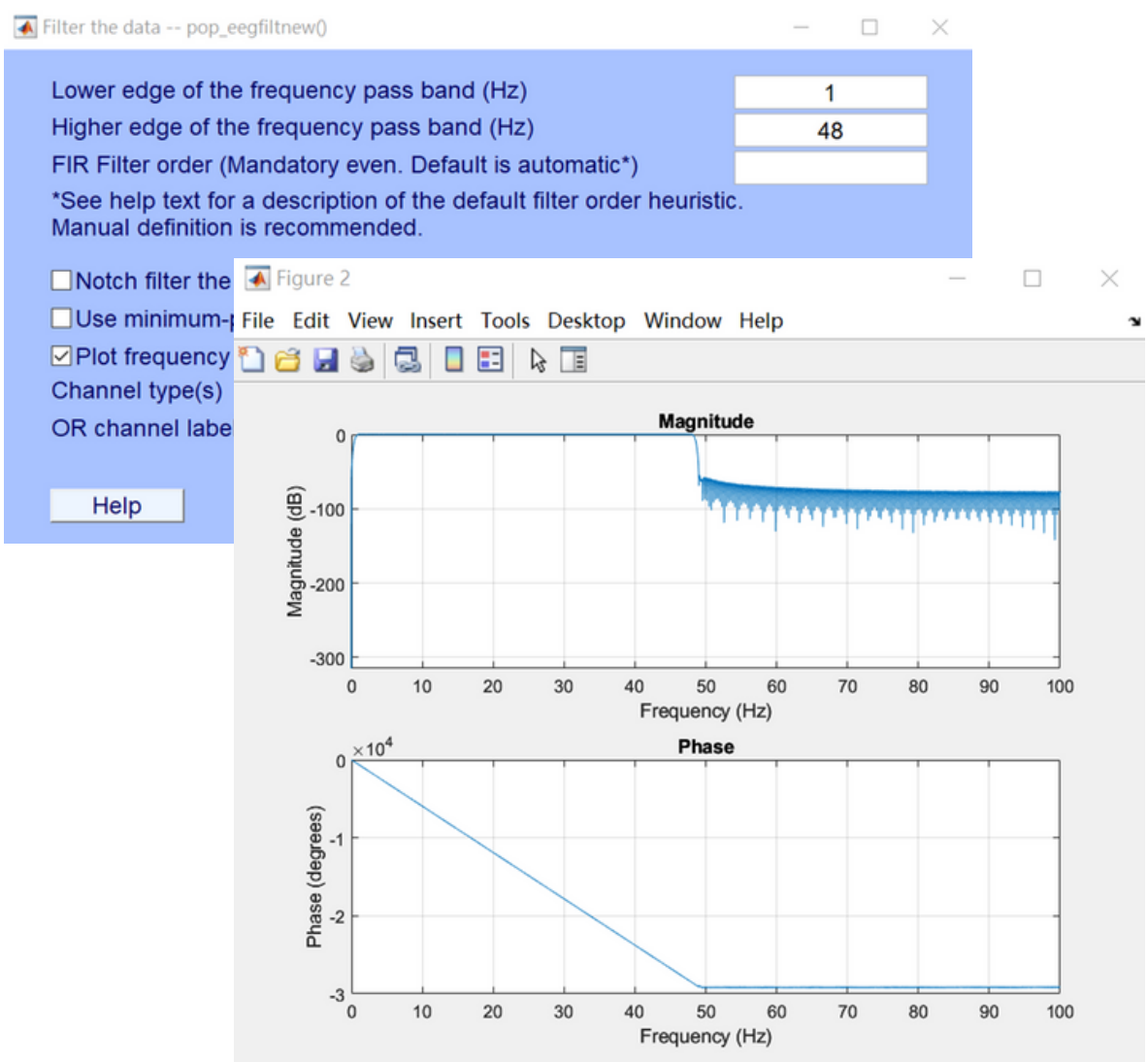


Problem 2

1 Plot component maps in 2D



2 Bandpass filtering [1, 48]Hz



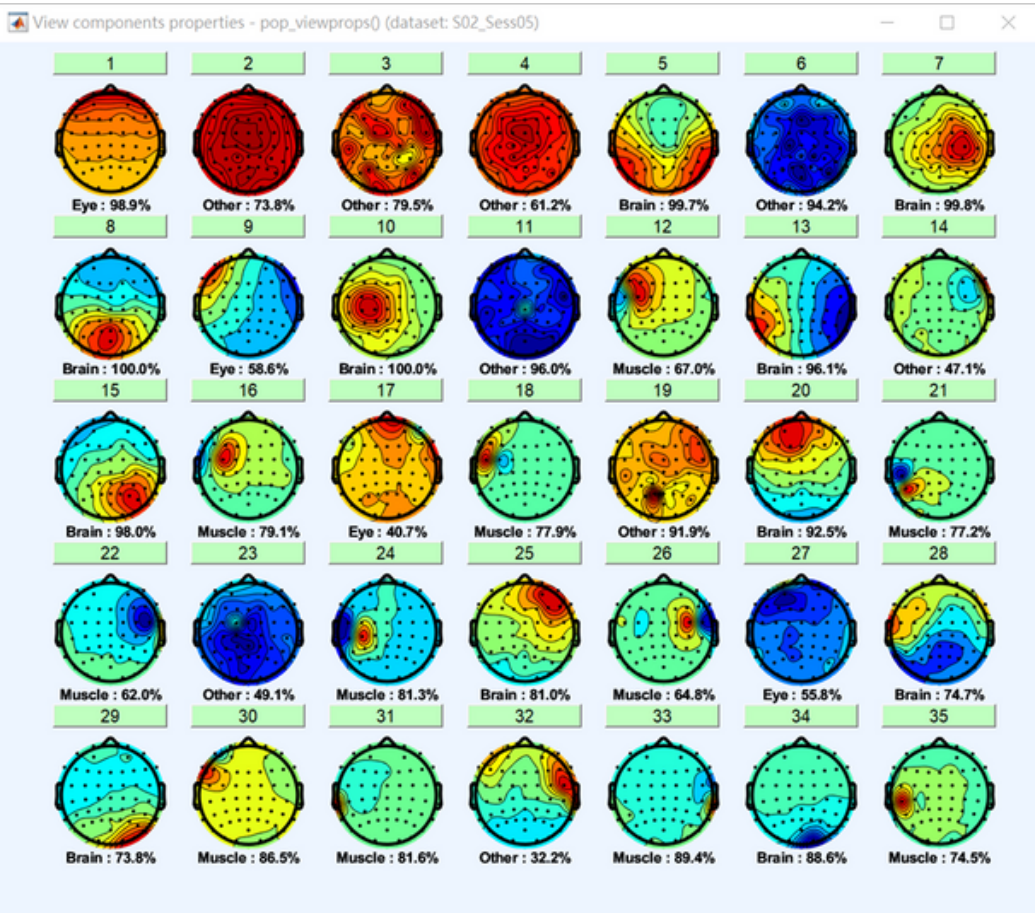
Problem 2

3 Run ICA and record computational time of ICA by code.

```
Attempting to convert data matrix to double precision for more accurate ICA results.

Input data size [56,196001] = 56 channels, 196001 frames/nFinding 56 ICA components using extended ICA.
Kurtosis will be calculated initially every 1 blocks using 6000 data points.
Decomposing 62 frames per ICA weight ((3136)^2 = 196001 weights, Initial learning rate will be 0.001, block size 61.
Learning rate will be multiplied by 0.98 whenever angledelta >= 60 deg.
More than 32 channels: default stopping weight change 1E-7
Training will end when wchange < 1e-07 or after 512 steps.
Online bias adjustment will be used.
Removing mean of each channel ...
Final training data range: -131.631 to 212.363
Computing the sphering matrix...
Starting weights are the identity matrix ...
Sphering the data ...
Beginning ICA training ... first training step may be slow ...
Lowering learning rate to 0.0009 and starting again.
Lowering learning rate to 0.00081 and starting again.
Lowering learning rate to 0.000729 and starting again.
Lowering learning rate to 0.0006561 and starting again.
Lowering learning rate to 0.00059049 and starting again.
Lowering learning rate to 0.000531441 and starting again.
Lowering learning rate to 0.000478297 and starting again.
step 1 - lrate 0.000478, wchange 31.81167821, angledelta  0.0 deg
Lowering learning rate to 0.000430467 and starting again.


Lowering learning rate to 0.000531441 and starting again.
Lowering learning rate to 0.000478297 and starting again.
step 1 - lrate 0.000478, wchange 31.81167821, angledelta  0.0 deg
Lowering learning rate to 0.000430467 and starting again.
Lowering learning rate to 0.00038742 and starting again.
step 1 - lrate 0.000387, wchange 29.96084484, angledelta  0.0 deg
Lowering learning rate to 0.000348678 and starting again.
step 1 - lrate 0.000349, wchange 29.05484481, angledelta  0.0 deg
Lowering learning rate to 0.000313811 and starting again.
step 1 - lrate 0.000314, wchange 29.26265108, angledelta  0.0 deg
step 2 - lrate 0.000314, wchange 1.91722705, angledelta  0.0 deg
step 3 - lrate 0.000314, wchange 1.22708517, angledelta 79.5 deg
step 4 - lrate 0.000308, wchange 1.31368342, angledelta 87.1 deg
step 5 - lrate 0.000301, wchange 1.57915915, angledelta 128.2 deg
step 6 - lrate 0.000295, wchange 0.90480765, angledelta 123.9 deg
step 7 - lrate 0.000289, wchange 0.81566718, angledelta 118.5 deg
step 8 - lrate 0.000284, wchange 0.86149043, angledelta 131.1 deg
step 9 - lrate 0.000278, wchange 1.08662501, angledelta 82.9 deg
step 10 - lrate 0.000272, wchange 1.28493646, angledelta 130.6 deg
step 11 - lrate 0.000267, wchange 0.59666345, angledelta 121.0 deg
step 12 - lrate 0.000262, wchange 0.94931858, angledelta 111.8 deg
step 13 - lrate 0.000256, wchange 0.99288639, angledelta 97.1 deg
step 14 - lrate 0.000251, wchange 1.87031991, angledelta 103.7 deg
step 15 - lrate 0.000246, wchange 1.26736792, angledelta 115.1 deg

step 419 - lrate 0.000001, wchange 0.00000606, angledelta 10.6 deg
step 420 - lrate 0.000001, wchange 0.00000418, angledelta 13.6 deg
step 421 - lrate 0.000001, wchange 0.00000277, angledelta 13.5 deg
step 422 - lrate 0.000001, wchange 0.00000205, angledelta 22.1 deg
step 423 - lrate 0.000001, wchange 0.00000139, angledelta 20.9 deg
step 424 - lrate 0.000001, wchange 0.00000099, angledelta 25.1 deg
step 425 - lrate 0.000001, wchange 0.00000086, angledelta 36.8 deg
step 426 - lrate 0.000001, wchange 0.00000050, angledelta 31.6 deg
step 427 - lrate 0.000001, wchange 0.00000046, angledelta 41.0 deg
step 428 - lrate 0.000001, wchange 0.00000036, angledelta 45.8 deg
step 429 - lrate 0.000001, wchange 0.00000027, angledelta 47.3 deg
step 430 - lrate 0.000001, wchange 0.00000033, angledelta 59.5 deg
step 431 - lrate 0.000001, wchange 0.00000030, angledelta 64.0 deg
step 432 - lrate 0.000001, wchange 0.00000023, angledelta 94.8 deg
step 433 - lrate 0.000001, wchange 0.00000022, angledelta 85.8 deg
step 434 - lrate 0.000000, wchange 0.00000016, angledelta 93.7 deg
step 435 - lrate 0.000000, wchange 0.00000030, angledelta 92.2 deg
step 436 - lrate 0.000000, wchange 0.00000016, angledelta 105.5 deg
step 437 - lrate 0.000000, wchange 0.00000011, angledelta 103.3 deg
step 438 - lrate 0.000000, wchange 0.00000008, angledelta 85.7 deg
Sorting components in descending order of mean projected variance ...
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Done.
```

Problem 2

4 Plot component maps in 2D



Problem 2

5 **Indicate noise component(s) if they exist and explain the reason why you identify this component as noise or artifacts**

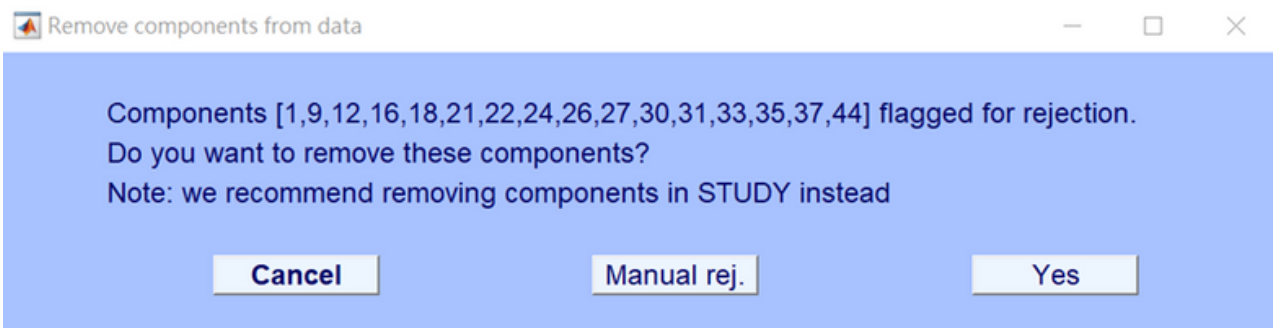
The noise components I reject： Muscle, Eyenoise

I delete noise components of eye and line noise because they are common sources of interference that can significantly distort the EEG signals and obscure the underlying brain activity.

Muscle noise in EEG signals is caused by the electrical activity of muscles, including facial muscles, jaw muscles, and scalp muscles. Muscle noise can be highly variable and typically has a higher amplitude and frequency range than the EEG signals of interest, making it challenging to distinguish from brain activity. If not removed, muscle noise can mask or obscure EEG signals, leading to inaccurate interpretation of the data.

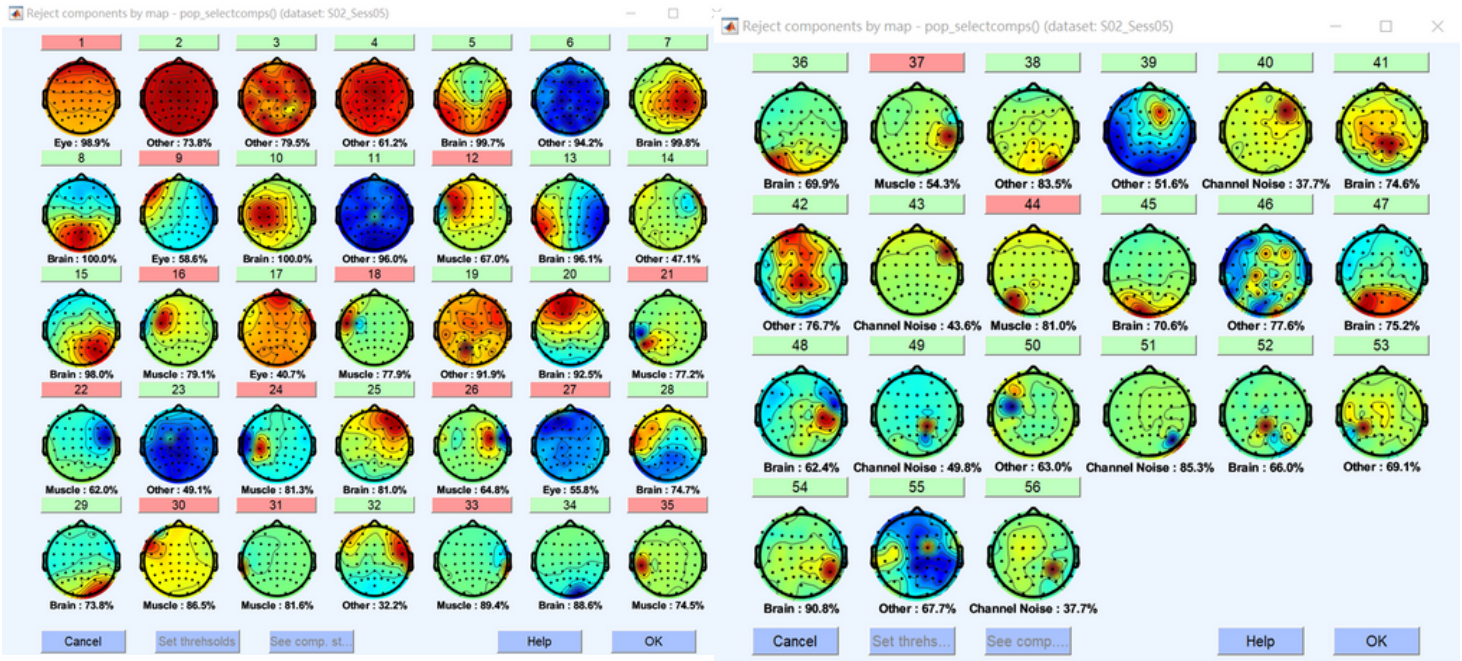
Eye noise is caused by the electrical activity generated by eye movements, such as blinks, saccades, and fixation shifts. Eye noise can be particularly problematic for EEG studies that involve visual tasks or require participants to maintain fixation. If not removed, eye noise can interfere with the measurement of brain activity related to the task or cognitive process of interest.

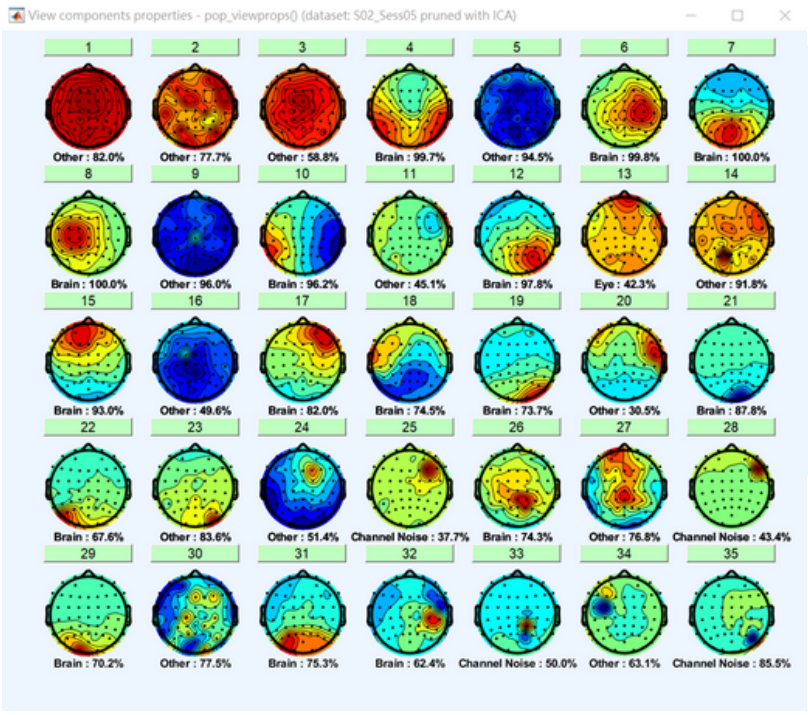
Overall, removing muscle and eye noise is crucial in obtaining high-quality EEG signals that accurately reflect the underlying brain activity and minimize the risk of misinterpretation or false conclusions.

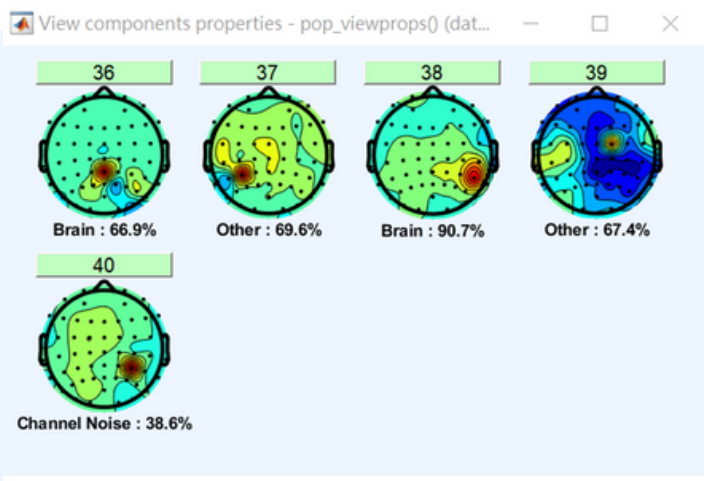


Problem 2

5 Indicate noise component(s) if they exist and explain the reason why you identify this component as noise or artifacts

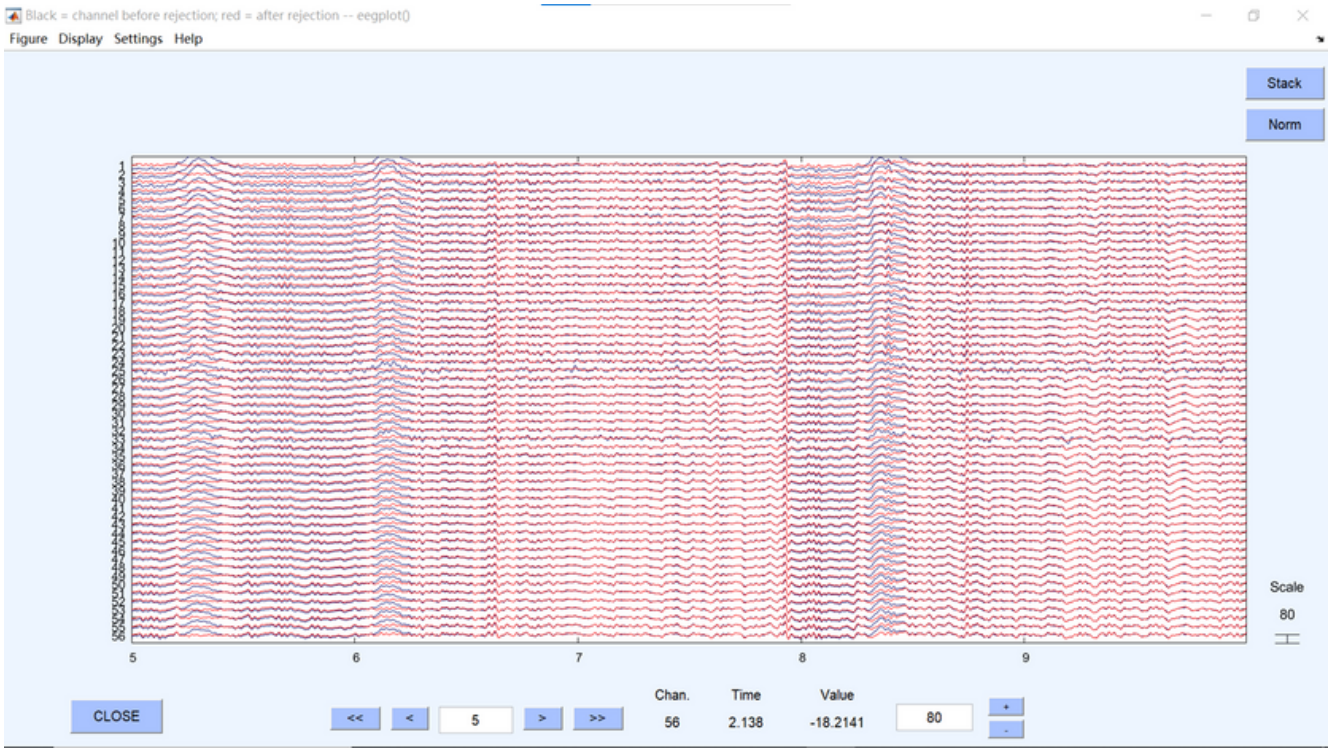






Problem 2

6 Plot first 10-second channel data before and after deleting noise/artifact component(s)



Problem 2

7 Discuss the effect of bandpassing(highpassing) the signal before running ICA

Bandpass filtering or highpass filtering the signal before running ICA can have several effects on the resulting independent components.

First, highpass filtering can remove low-frequency drift and other slow fluctuations that are not of interest in EEG analysis, thus reducing the dimensionality of the data and making it easier to identify independent components. This can help improve the computational efficiency and accuracy of the ICA algorithm, since it is less likely to get stuck in local optima or identify spurious components that may be related to low-frequency artifacts.

Second, highpass filtering can help separate brain activity from other sources of noise or artifacts that may be present in the data. For example, muscle activity, eye movements, or other sources of noise tend to have slower temporal dynamics than brain activity, and thus may be effectively removed by highpass filtering. This can help improve the signal-to-noise ratio and enhance the quality of the resulting independent components, making them more interpretable and relevant for subsequent analysis.

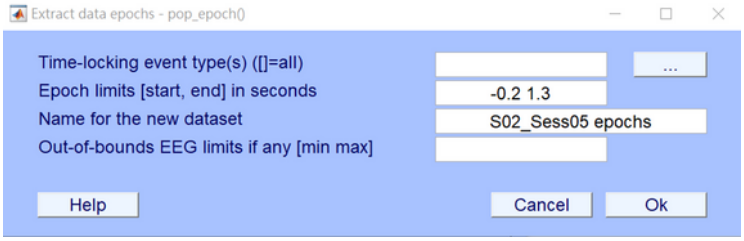
However, it is important to note that highpass filtering can also remove some of the relevant EEG activity that occurs at low frequencies, such as slow-wave oscillations or other types of cortical activity that are relevant to the research question. Therefore, it is important to carefully consider the frequency range of interest and the potential trade-off between removing noise and preserving relevant signal when deciding whether to highpass filter the data before running ICA.

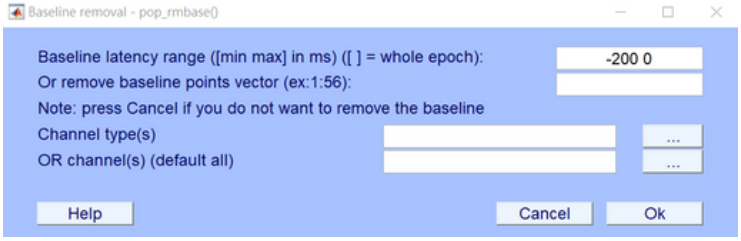
Problem 3

1~3 processing data

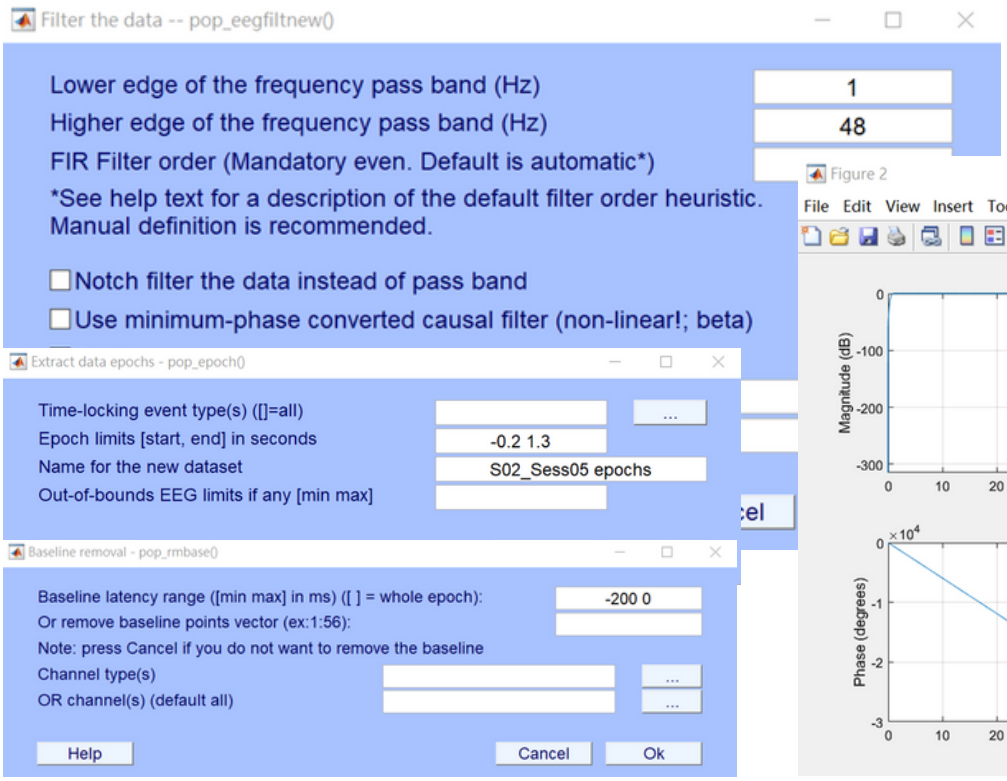
For problem 3, I followed a multi-step process to obtain the required output. Firstly, I processed the necessary tasks by utilizing the EEG GUI interface, completing steps 1 to 3. Once this was completed, I exported the processed data into a new file. Finally, I generated the desired picture by writing and running MATLAB code.

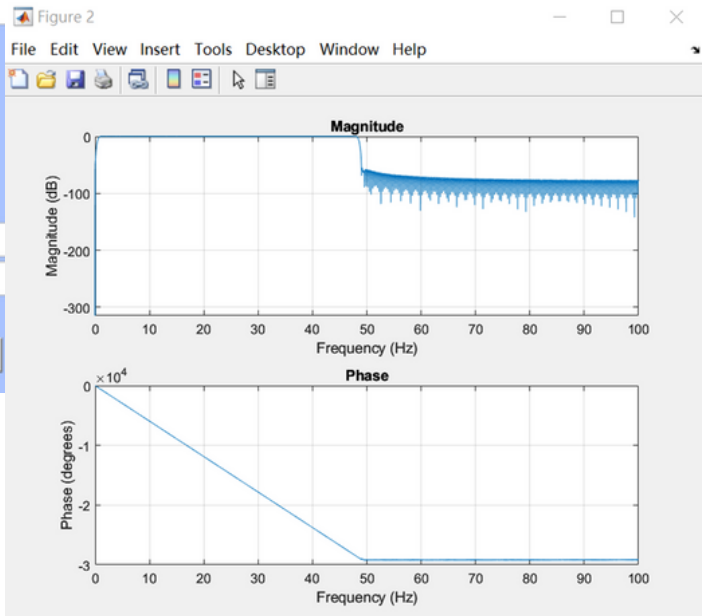
A. Without any operation
-> Epoch the continuous EEG with a time interval [-0.2 1.3] sec
-> Remove the epoch baseline mean





B. Bandpass the signal (1~48 Hz)
-> Epoch the continuous EEG with a time interval [-0.2 1.3] sec
-> Remove the epoch baseline mean





Problem 3

1~3 processing data

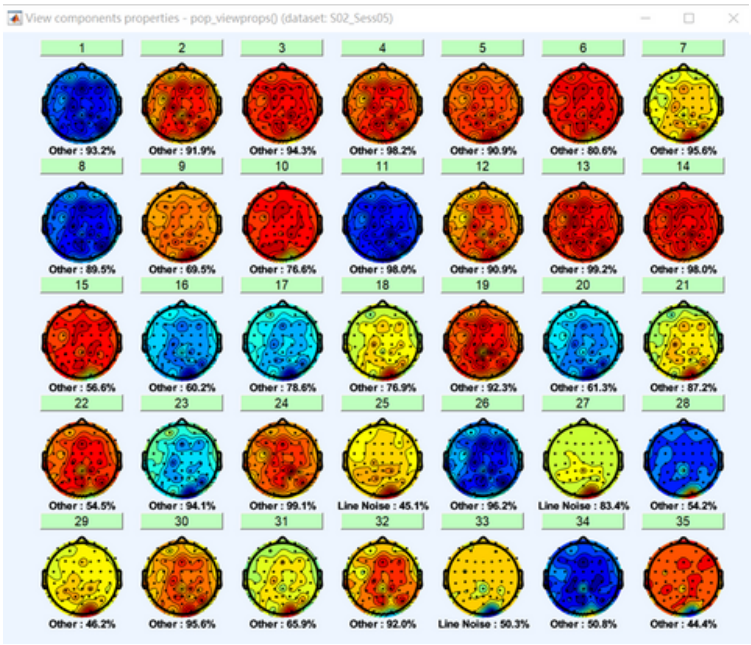
C. Run ICA and remove bad components
-> Epoch the continuous EEG with a time interval [-0.2 1.3] sec
-> Remove the epoch baseline mean

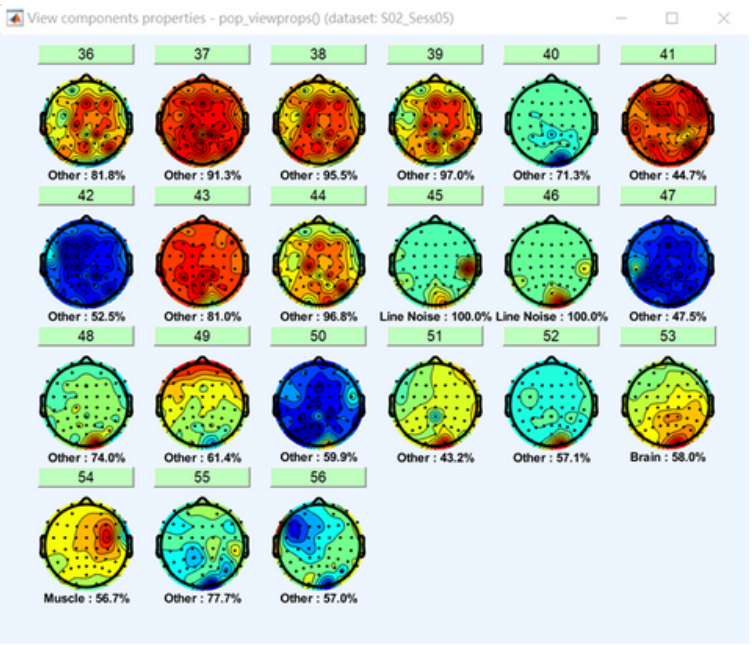
```
Attempting to convert data matrix to double precision for more accurate ICA results.

Input data size [56,196001] = 56 channels, 196001 frames/nFinding 56 ICA components using extended ICA.
Kurtosis will be calculated initially every 1 blocks using 6000 data points.
Decomposing 62 frames per ICA weight ((3136)^2 = 196001 weights, Initial learning rate will be 0.001, block size 61.
Learning rate will be multiplied by 0.98 whenever angledelta >= 60 deg.
More than 32 channels: default stopping weight change 1E-7
Training will end when wchange < 1e-07 or after 512 steps.
Online bias adjustment will be used.
Removing mean of each channel ...
Final training data range: -131.631 to 212.363
Computing the sphering matrix...
Starting weights are the identity matrix ...
Sphering the data ...
Beginning ICA training ... first training step may be slow ...
Lowering learning rate to 0.0009 and starting again.
Lowering learning rate to 0.00081 and starting again.
Lowering learning rate to 0.000729 and starting again.
Lowering learning rate to 0.0006561 and starting again.
Lowering learning rate to 0.00059049 and starting again.
Lowering learning rate to 0.000531441 and starting again.
Lowering learning rate to 0.000478297 and starting again.
Lowering learning rate to 0.000430467 and starting again.
step 1 - lrate 0.000430, wchange 31.08554074, angledelta  0.0 deg

 Lowering learning rate to 0.00038742 and starting again.
 step 1 - lrate 0.000387, wchange 29.05900791, angledelta  0.0 deg
 step 2 - lrate 0.000387, wchange 2.91292442, angledelta  0.0 deg
 step 3 - lrate 0.000387, wchange 2.54267316, angledelta 89.2 deg
 step 4 - lrate 0.000380, wchange 1.87215957, angledelta 108.1 deg
 step 5 - lrate 0.000372, wchange 1.82085710, angledelta 120.0 deg
 step 6 - lrate 0.000365, wchange 2.20076391, angledelta 112.9 deg
 step 7 - lrate 0.000357, wchange 1.68120961, angledelta 111.2 deg
 step 8 - lrate 0.000350, wchange 2.76306956, angledelta 107.9 deg
 step 9 - lrate 0.000343, wchange 2.49754337, angledelta 112.4 deg
 Lowering learning rate to 0.000302697 and starting again.
 step 1 - lrate 0.000303, wchange 26.95744749, angledelta  0.0 deg
 step 2 - lrate 0.000303, wchange 1.15689059, angledelta  0.0 deg
 step 3 - lrate 0.000303, wchange 1.31246127, angledelta 86.7 deg
 step 4 - lrate 0.000297, wchange 1.44294533, angledelta 109.4 deg
 step 5 - lrate 0.000291, wchange 1.38808418, angledelta 118.5 deg
 step 6 - lrate 0.000285, wchange 0.59230801, angledelta 110.0 deg
 step 7 - lrate 0.000279, wchange 0.89571786, angledelta 90.1 deg
 step 8 - lrate 0.000274, wchange 1.28833100, angledelta 129.9 deg
 step 9 - lrate 0.000268, wchange 0.87913939, angledelta 116.0 deg
 step 10 - lrate 0.000263, wchange 1.17824153, angledelta 113.2 deg
 step 11 - lrate 0.000258, wchange 1.21464107, angledelta 105.6 deg

step 494 - lrate 0.000000, wchange 0.00000624, angledelta  4.5 deg
step 495 - lrate 0.000000, wchange 0.00003347, angledelta 150.4 deg
step 496 - lrate 0.000000, wchange 0.00034814, angledelta  6.8 deg
step 497 - lrate 0.000000, wchange 0.00034823, angledelta  7.1 deg
step 498 - lrate 0.000000, wchange 0.00001044, angledelta 159.9 deg
step 499 - lrate 0.000000, wchange 0.00004679, angledelta  7.3 deg
step 500 - lrate 0.000000, wchange 0.00003512, angledelta  7.3 deg
step 501 - lrate 0.000000, wchange 0.00002675, angledelta  7.5 deg
step 502 - lrate 0.000000, wchange 0.00033043, angledelta 167.3 deg
step 503 - lrate 0.000000, wchange 0.00031743, angledelta  0.6 deg
step 504 - lrate 0.000000, wchange 0.00002351, angledelta  3.5 deg
step 505 - lrate 0.000000, wchange 0.00009793, angledelta 177.0 deg
step 506 - lrate 0.000000, wchange 0.00006915, angledelta  1.3 deg
step 507 - lrate 0.000000, wchange 0.00005166, angledelta  1.8 deg
step 508 - lrate 0.000000, wchange 0.00003904, angledelta  1.3 deg
step 509 - lrate 0.000000, wchange 0.00002983, angledelta  1.7 deg
step 510 - lrate 0.000000, wchange 0.00002298, angledelta  2.0 deg
step 511 - lrate 0.000000, wchange 0.00001781, angledelta  2.2 deg
step 512 - lrate 0.000000, wchange 0.00001387, angledelta  2.5 deg
Sorting components in descending order of mean projected variance ...
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Done.
```

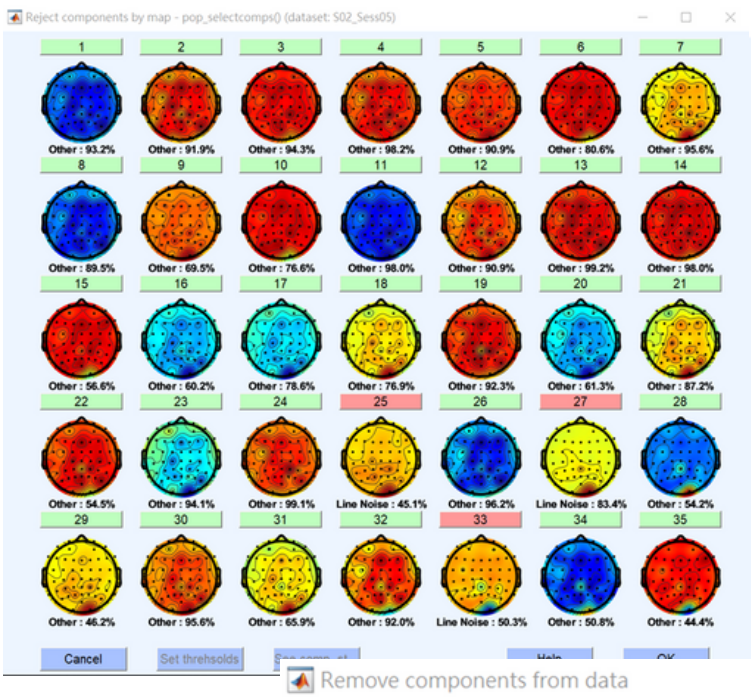
Problem 3

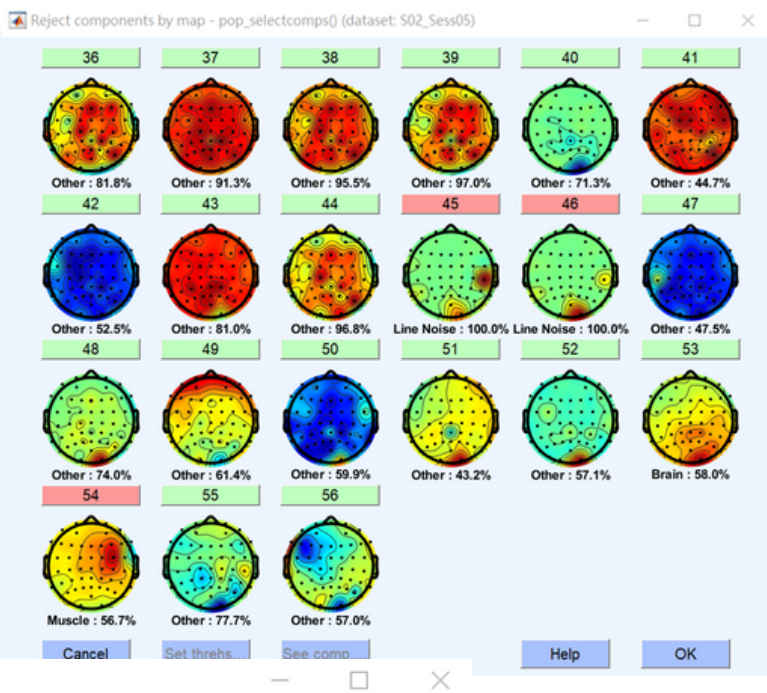
1~3 processing data

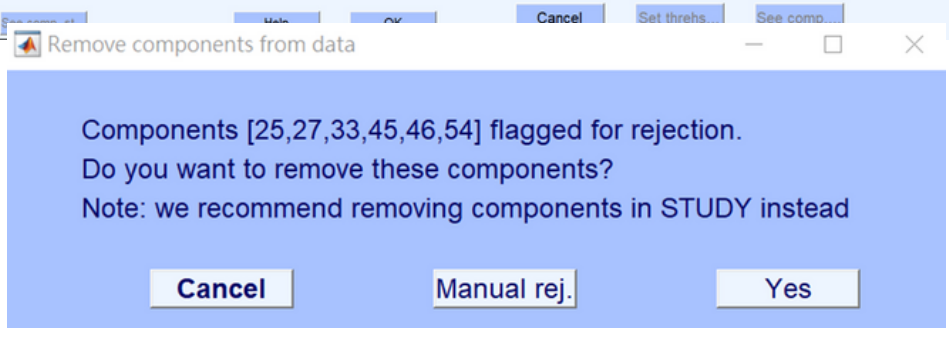
C. Run ICA and remove bad components
-> Epoch the continuous EEG with a time interval [-0.2 1.3] sec
-> Remove the epoch baseline mean







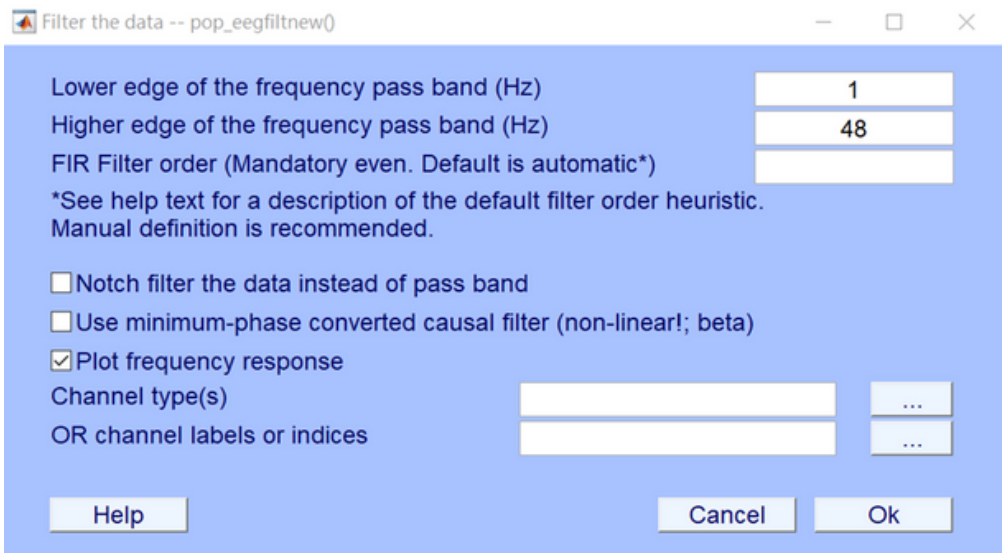


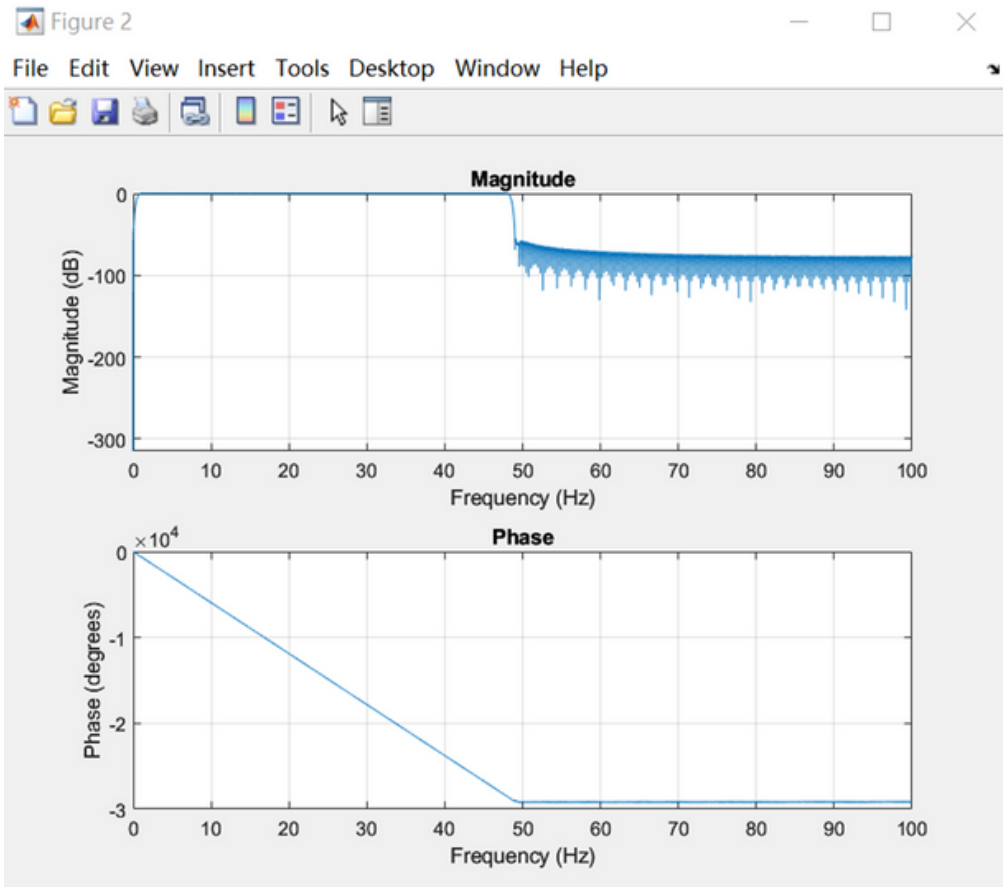


Problem 3

1~3 processing data

D. Bandpass the signal (1~48 Hz) first and run ICA to remove bad components
-> Epoch the continuous EEG with a time interval [-0.2 1.3] sec
-> Remove the epoch baseline mean





Problem 3

1~3 processing data

D. Bandpass the signal (1~48 Hz) first and run ICA to remove bad components

-> Epoch the continuous EEG with a time interval [-0.2 1.3] sec

-> Remove the epoch baseline mean

```
Attempting to convert data matrix to double precision for more accurate ICA results.

Input data size [56,196001] = 56 channels, 196001 frames/nFinding 56 ICA components using extended ICA.
Kurtosis will be calculated initially every 1 blocks using 6000 data points.
Decomposing 62 frames per ICA weight ((3136)^2 = 196001 weights, Initial learning rate will be 0.001, block size 61.
Learning rate will be multiplied by 0.98 whenever angledelta >= 60 deg.
More than 32 channels: default stopping weight change 1E-7
Training will end when wchange < 1e-07 or after 512 steps.
Online bias adjustment will be used.
Removing mean of each channel ...
Final training data range: -131.631 to 212.363
Computing the sphering matrix...
Starting weights are the identity matrix ...
Sphering the data ...
Beginning ICA training ... first training step may be slow ...
Lowering learning rate to 0.0009 and starting again.
Lowering learning rate to 0.00081 and starting again.
Lowering learning rate to 0.000729 and starting again.
Lowering learning rate to 0.0006561 and starting again.
Lowering learning rate to 0.00059049 and starting again.
Lowering learning rate to 0.000531441 and starting again.
Lowering learning rate to 0.000478297 and starting again.
Lowering learning rate to 0.000430467 and starting again.
step 1 - lrate 0.000430, wchange 31.08554074, angledelta  0.0 deg
Lowering learning rate to 0.00038742 and starting again.
step 1 - lrate 0.000387, wchange 29.05900791, angledelta  0.0 deg
step 2 - lrate 0.000387, wchange 2.91292442, angledelta  0.0 deg
step 3 - lrate 0.000387, wchange 2.54267316, angledelta 89.2 deg
step 4 - lrate 0.000380, wchange 1.87215957, angledelta 108.1 deg
step 5 - lrate 0.000372, wchange 1.82085710, angledelta 120.0 deg
step 6 - lrate 0.000365, wchange 2.20076391, angledelta 112.9 deg
step 7 - lrate 0.000357, wchange 1.68120961, angledelta 111.2 deg
step 8 - lrate 0.000350, wchange 2.76306956, angledelta 107.9 deg
step 9 - lrate 0.000343, wchange 2.49754337, angledelta 112.4 deg
Lowering learning rate to 0.000302697 and starting again.
step 1 - lrate 0.000303, wchange 26.95744749, angledelta  0.0 deg
step 2 - lrate 0.000303, wchange 1.15689059, angledelta  0.0 deg
step 3 - lrate 0.000303, wchange 1.31246127, angledelta 86.7 deg
step 4 - lrate 0.000297, wchange 1.44294533, angledelta 109.4 deg
step 5 - lrate 0.000291, wchange 1.38808418, angledelta 118.5 deg
step 6 - lrate 0.000285, wchange 0.59230801, angledelta 110.0 deg
step 7 - lrate 0.000279, wchange 0.89571786, angledelta 90.1 deg
step 8 - lrate 0.000274, wchange 1.28833100, angledelta 129.9 deg
step 9 - lrate 0.000268, wchange 0.87913939, angledelta 116.0 deg
step 10 - lrate 0.000263, wchange 1.17824153, angledelta 113.2 deg
step 11 - lrate 0.000258, wchange 1.21464107, angledelta 105.6 deg
step 494 - lrate 0.000000, wchange 0.00000624, angledelta  4.5 deg
step 495 - lrate 0.000000, wchange 0.00003347, angledelta 150.4 deg
step 496 - lrate 0.000000, wchange 0.00034814, angledelta  6.8 deg
step 497 - lrate 0.000000, wchange 0.00034823, angledelta  7.1 deg
step 498 - lrate 0.000000, wchange 0.00001044, angledelta 159.9 deg
step 499 - lrate 0.000000, wchange 0.00004679, angledelta  7.3 deg
step 500 - lrate 0.000000, wchange 0.00003512, angledelta  7.3 deg
step 501 - lrate 0.000000, wchange 0.00002675, angledelta  7.5 deg
step 502 - lrate 0.000000, wchange 0.00033043, angledelta 167.3 deg
step 503 - lrate 0.000000, wchange 0.00031743, angledelta  0.6 deg
step 504 - lrate 0.000000, wchange 0.00002351, angledelta  3.5 deg
step 505 - lrate 0.000000, wchange 0.00009793, angledelta 177.0 deg
step 506 - lrate 0.000000, wchange 0.00006915, angledelta  1.3 deg
step 507 - lrate 0.000000, wchange 0.00005166, angledelta  1.8 deg
step 508 - lrate 0.000000, wchange 0.00003904, angledelta  1.3 deg
step 509 - lrate 0.000000, wchange 0.00002983, angledelta  1.7 deg
step 510 - lrate 0.000000, wchange 0.00002298, angledelta  2.0 deg
step 511 - lrate 0.000000, wchange 0.00001781, angledelta  2.2 deg
step 512 - lrate 0.000000, wchange 0.00001387, angledelta  2.5 deg
Sorting components in descending order of mean projected variance ...
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Scaling components to RMS microvolt
Done.
```

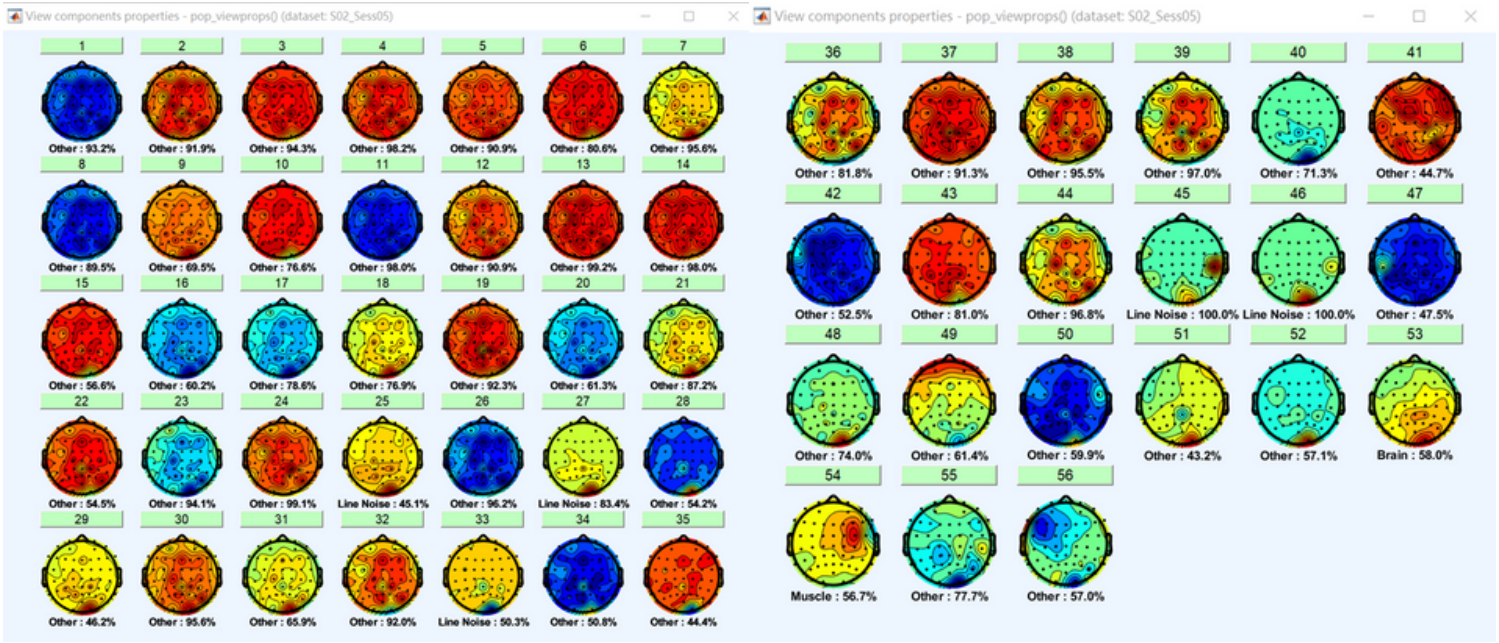
Problem 3

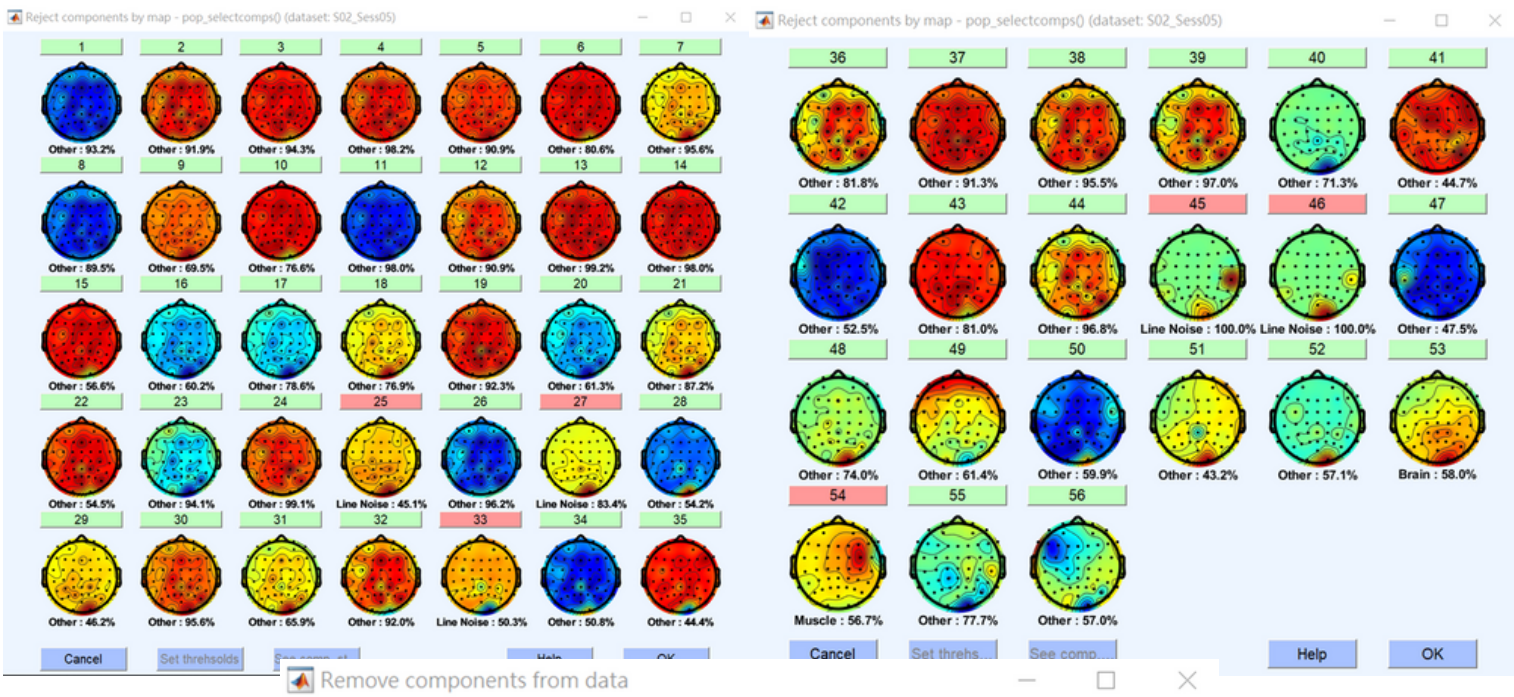
1~3 processing data

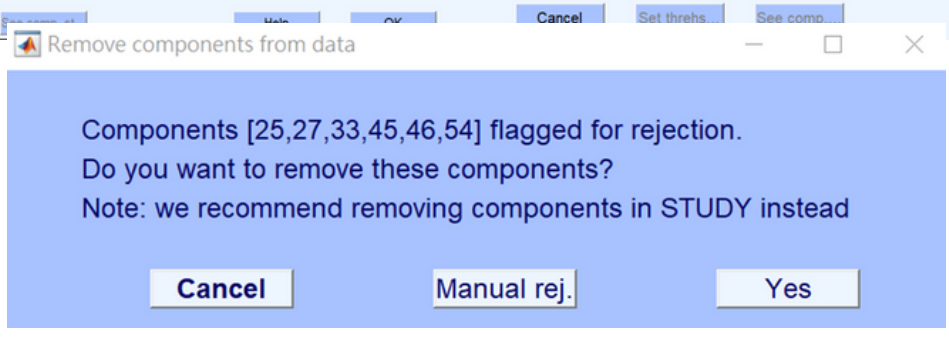
D. Bandpass the signal (1~48 Hz) first and run ICA to remove bad components

-> Epoch the continuous EEG with a time interval [-0.2 1.3] sec

-> Remove the epoch baseline mean



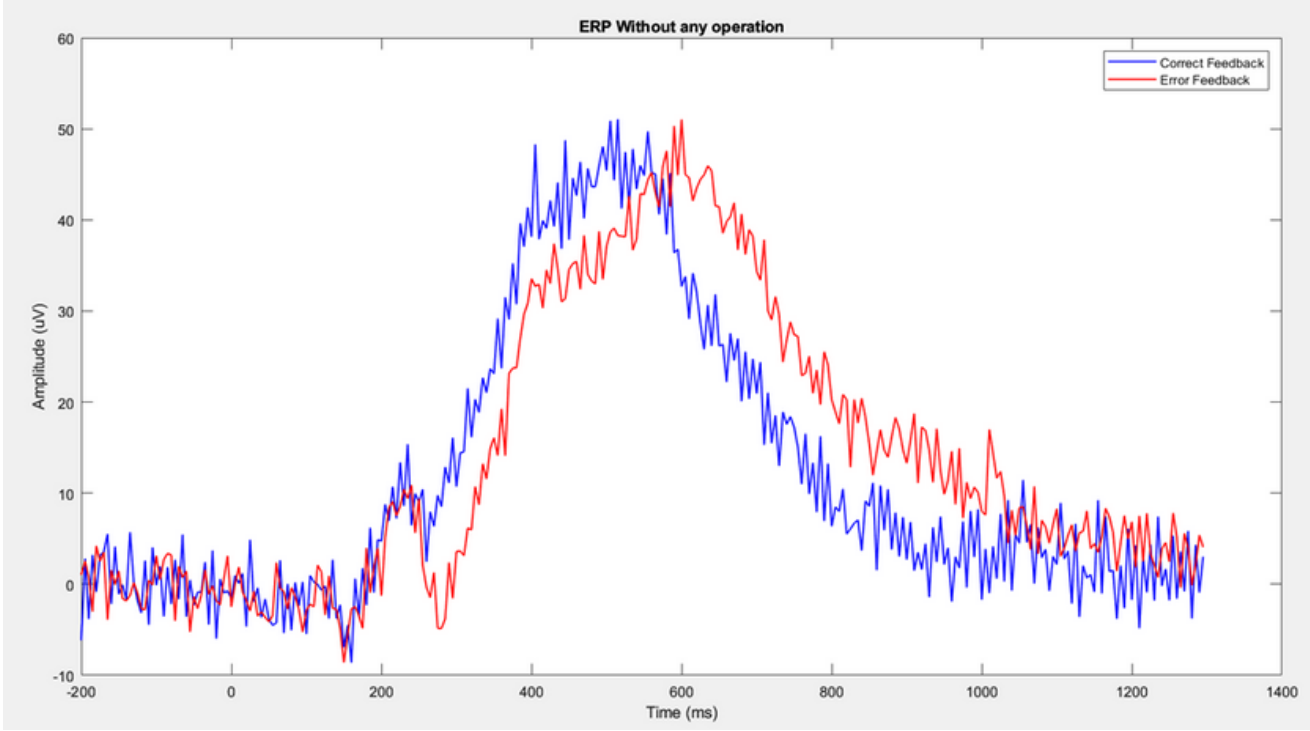




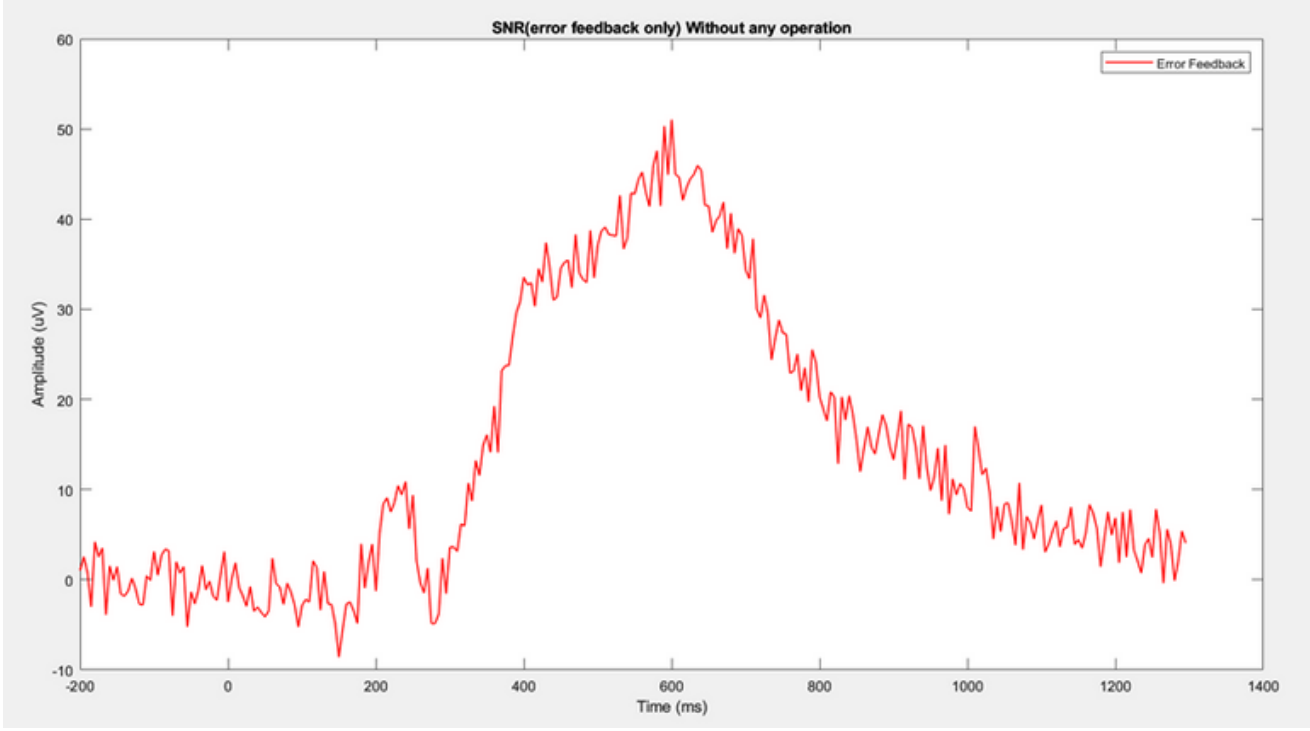
Problem 3

4~5 Plot ERP and SNR

A. Without any operation
-> ERP plot for 2 types of feedback



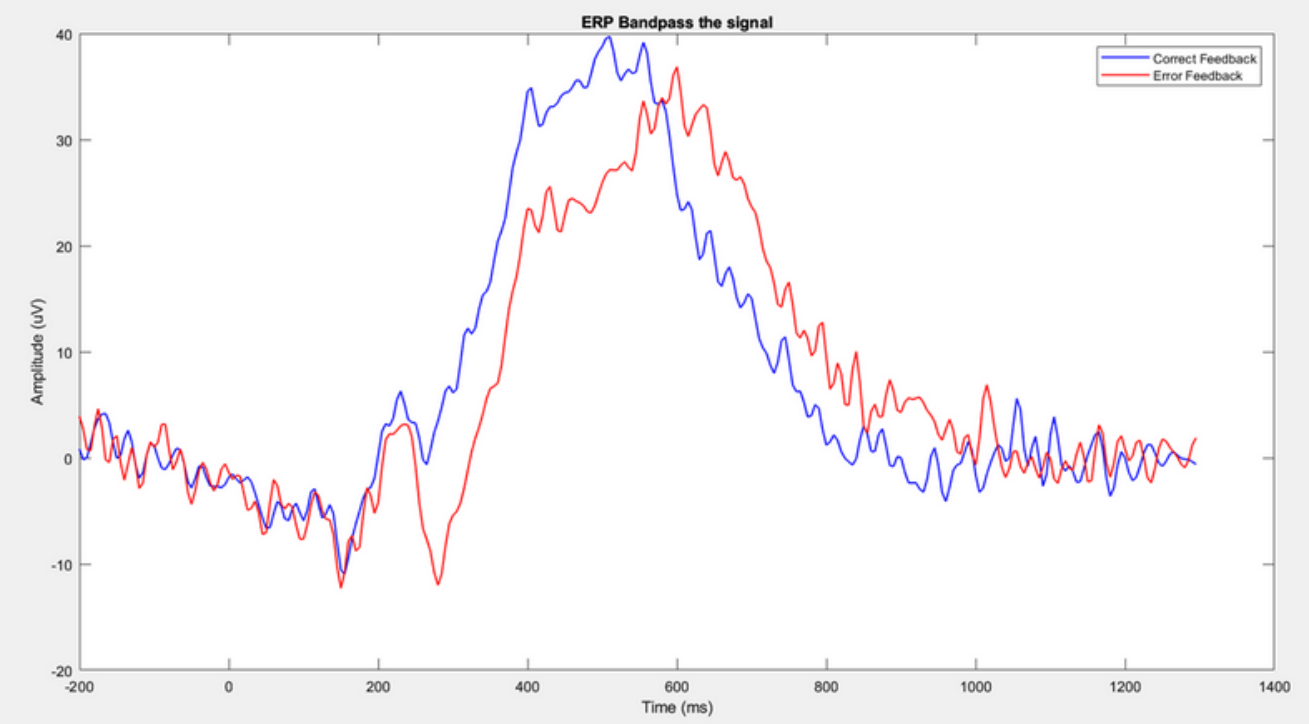
-> SNR plot (error feedback only)



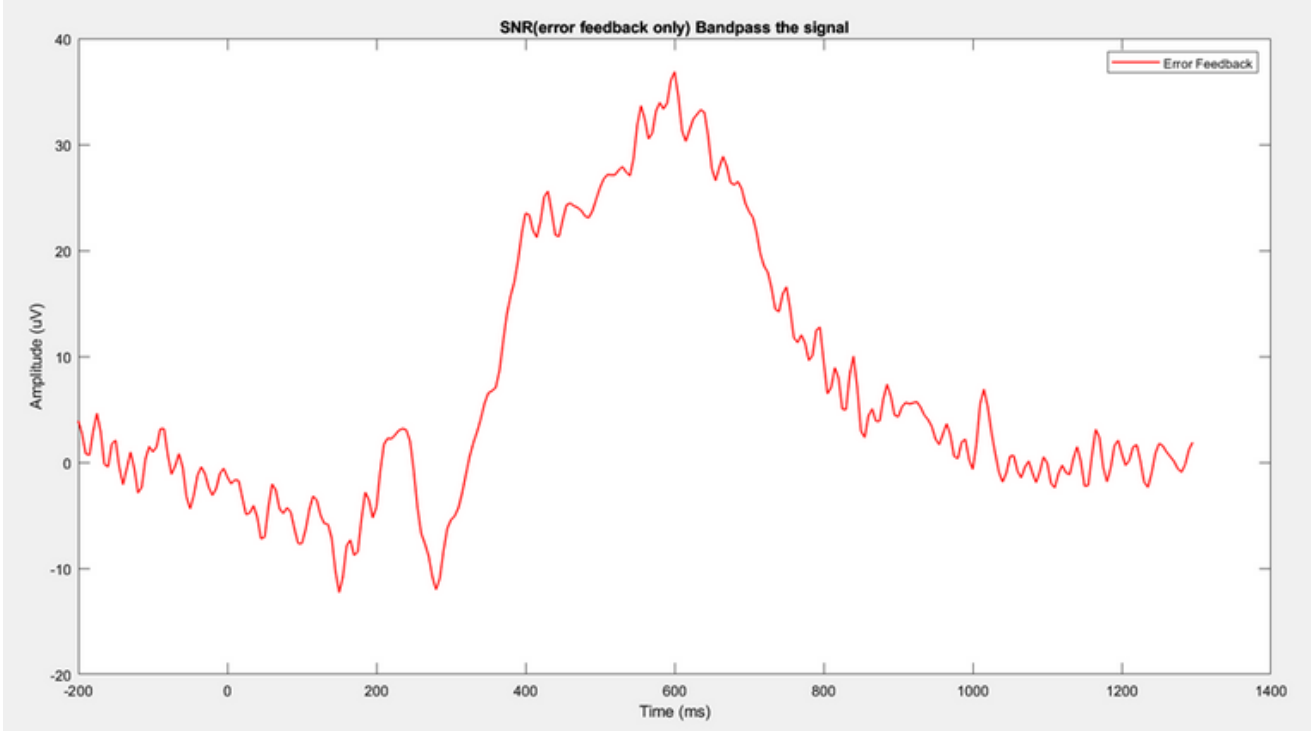
Problem 3

4~5 Plot ERP and SNR

B. Bandpass the signal (1~48 Hz)
-> ERP plot for 2 types of feedback



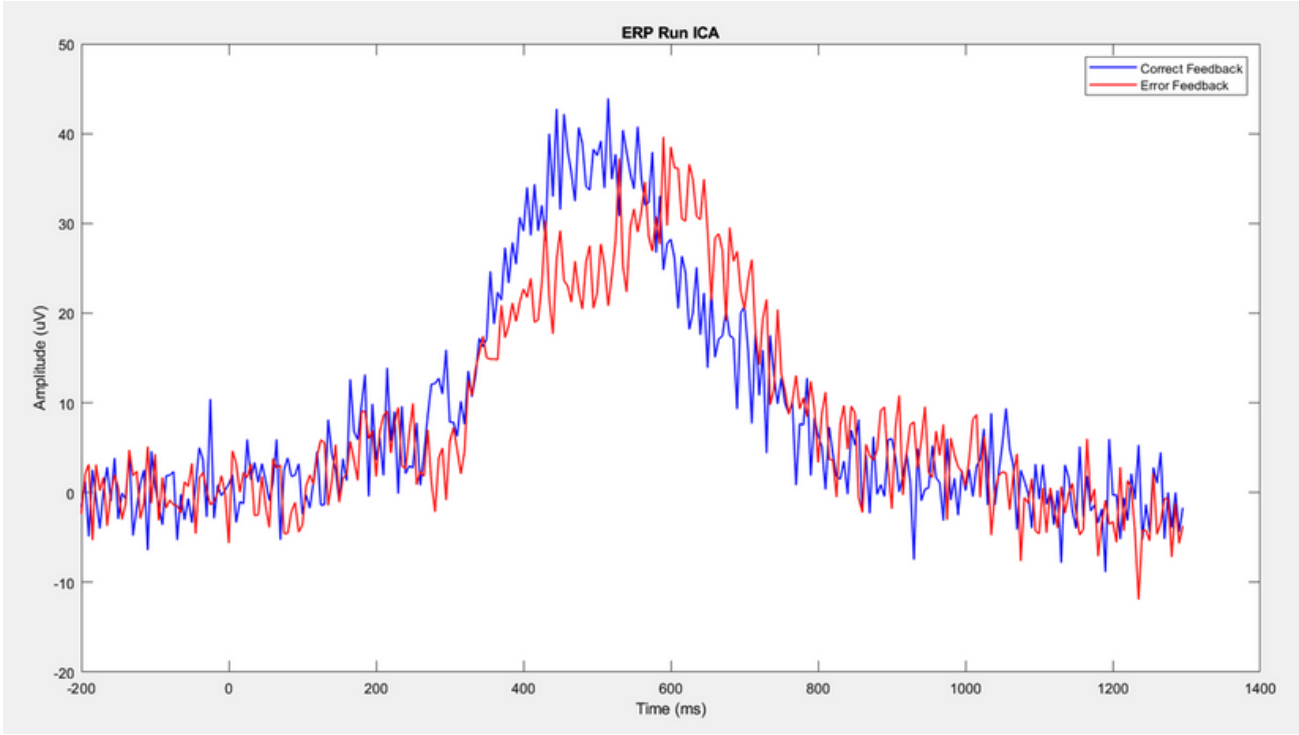
-> SNR plot (error feedback only)



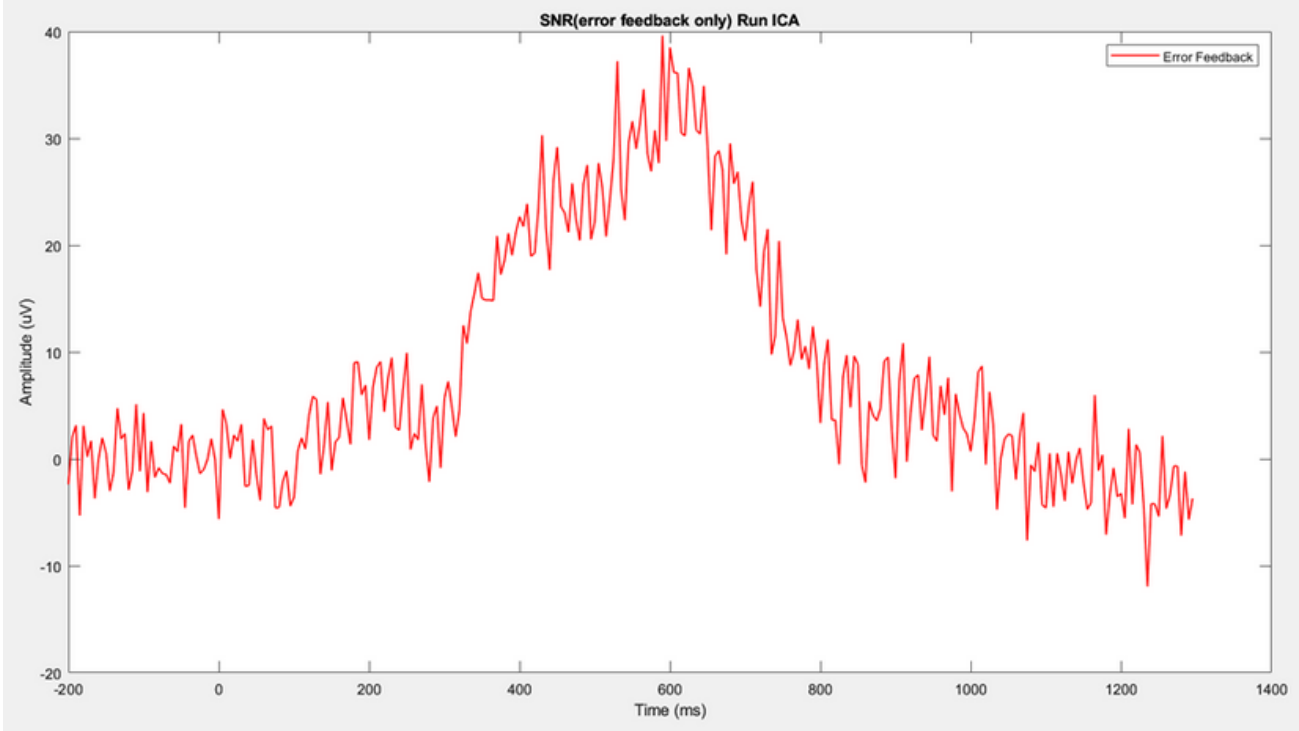
Problem 3

4~5 Plot ERP and SNR

C. Run ICA and remove bad components
-> ERP plot for 2 types of feedback



-> SNR plot (error feedback only)

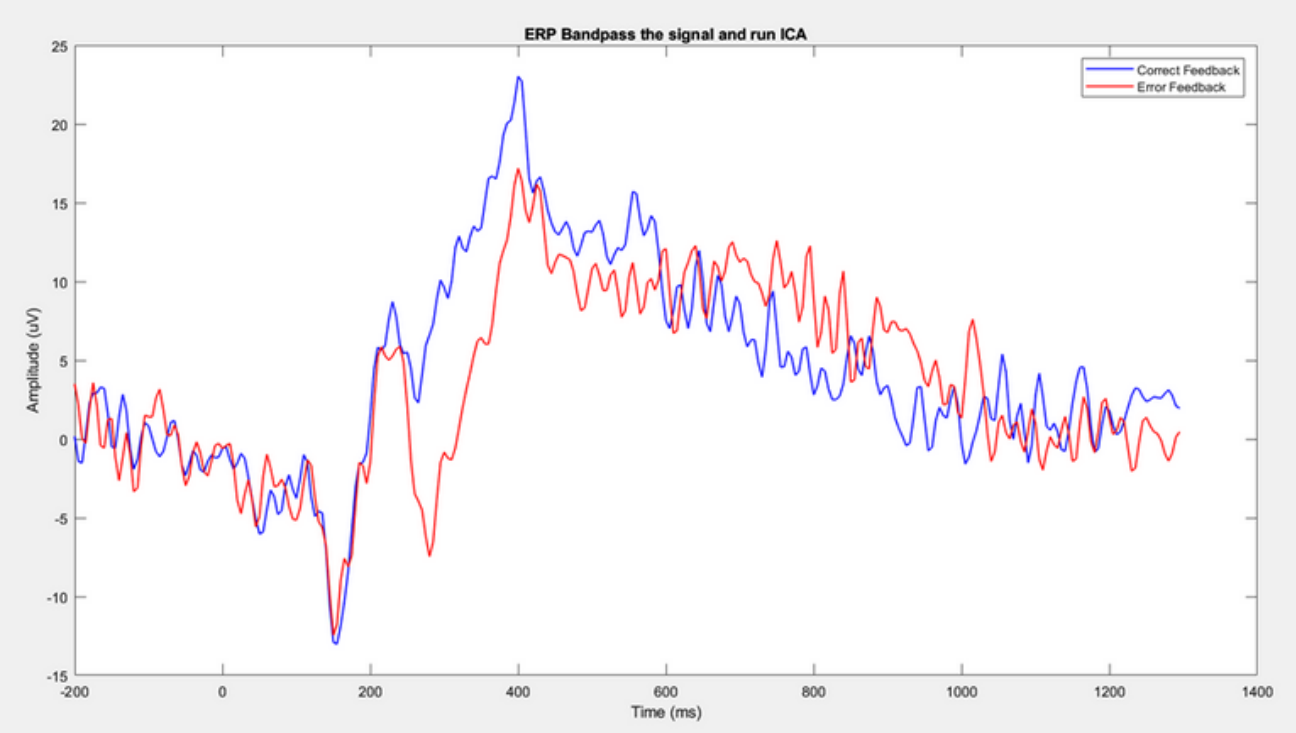


Problem 3

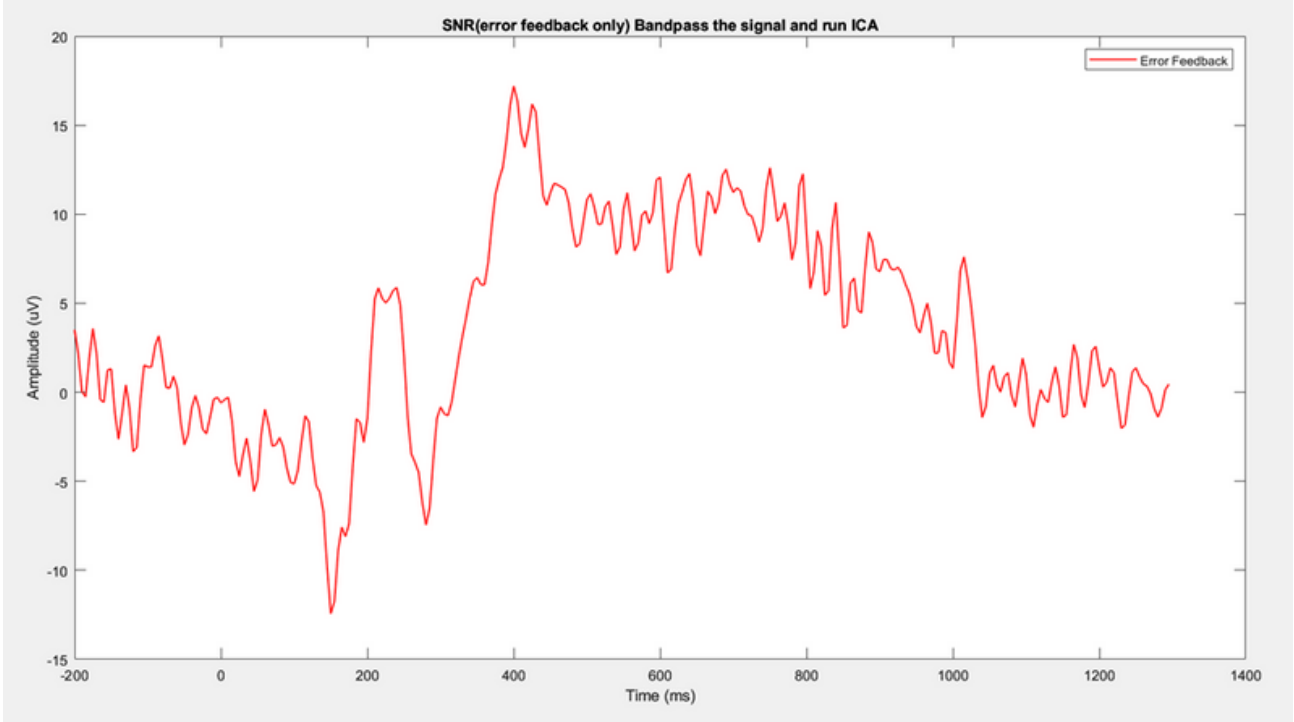
4~5 Plot ERP and SNR

D. Bandpass the signal (1~48 Hz) first and run ICA to remove bad components

-> ERP plot for 2 types of feedback

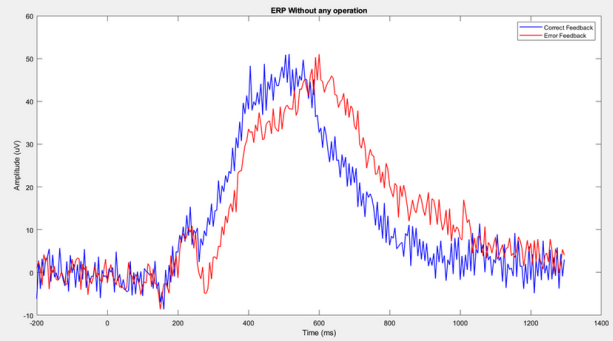
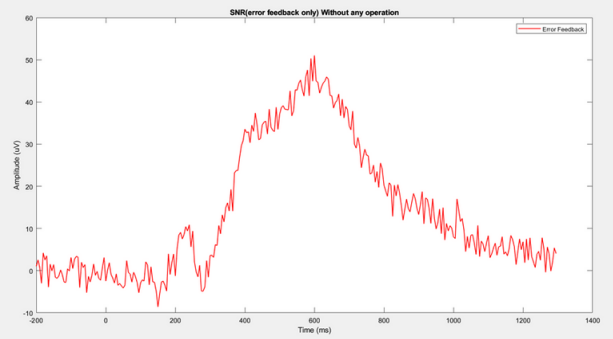
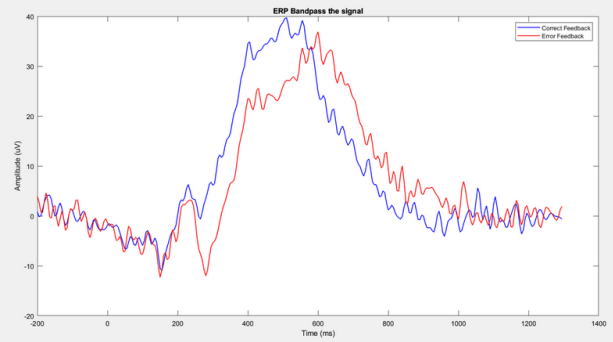
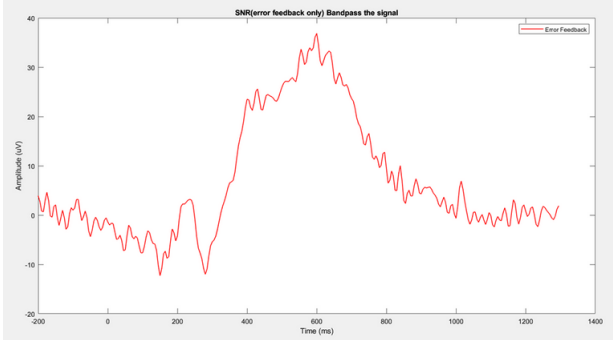
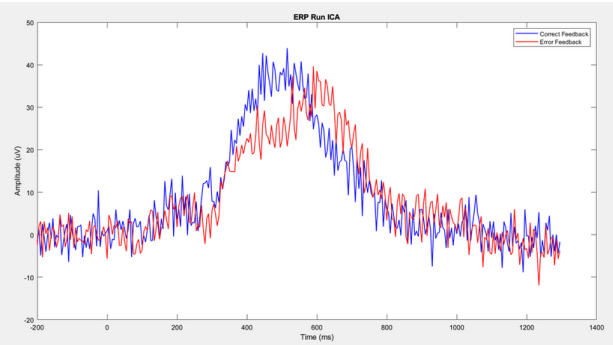
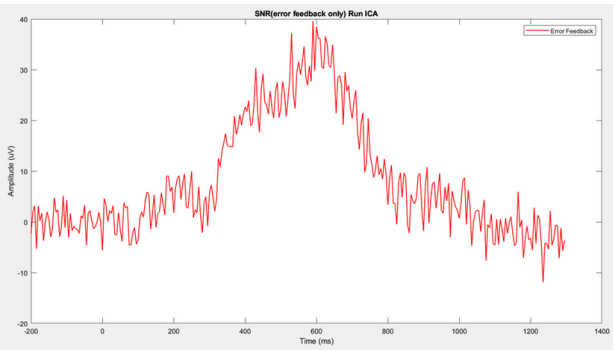
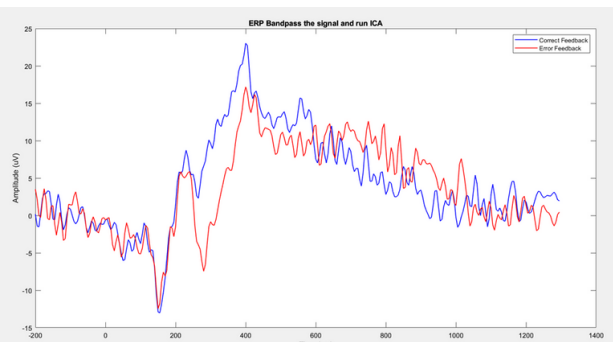


-> SNR plot (error feedback only)



Problem 3

4~5 Plot ERP and SNR

Preprocessing Methods	ERP plot for 2 types of feedback	SNR(error feedback only)
Without any operation		
Bandpass only		
IC removal only		
Bandpass +IC removal		

Part 3.
Bonus Problem

Bonus Problem : Solving ICA problem by kurtosis (z is white signal)

$$argmax_w \|kurt(w^T z)\| \; with \; w^T w = 1$$

1. Initialize the transformation matrix w with random values.
2. Repeat until convergence:
    a. Compute the transformed signal Y = wz.
    b. Compute the kurtosis of each component of Y.
    c. Update the transformation matrix w using the formula:
    w = w + α(Σ[g(Y)Y^T]/N - w)
    where α is the learning rate, N is the number of samples, g(Y) is a function that maximizes kurtosis , and Σ[g(Y)Y^T]/N is the expected value of the product of the nonlinearity g(Y) and the input signal Y.
    d. Normalize the rows of w so that w^T*w=I.