ERP CHALLENGES

In [ ]:
```
import numpy as np                    # Import functions for calculations.
import matplotlib.pyplot as plt       # Import functions for plotting
from scipy.io import loadmat          # Import function to read data.

data = loadmat('EEG-2.mat')
```

In [ ]:
```
data.keys() #EEGa (electrode 1), EEGb (electrode 2), t (time)
```

Out[ ]:
```
dict_keys(['__header__', '__version__', '__globals__', 't', 'EEG'])
```

In [ ]:
```
t = data['t'][0]
EEG2 = data['EEG']
```

In [ ]:
```
print('Shape of EEG from EEG2.mat is', np.shape(EEG2))
```

```
Shape of EEG from EEG2.mat is (1000, 500)
```

In [ ]:
```
ntrials2 = np.shape(EEG2)[0]
print('The number of trials in EEG2.mat is ', ntrials2)

nsamples2 = np.shape(EEG2)[1]
print("The number of samples in EEG2.mat is", nsamples2)
```

```
The number of trials in EEG2.mat is  1000
The number of samples in EEG2.mat is 500
```

1A - 1C EEG-2.mat

A.) What is the time between samples (dt) in seconds?

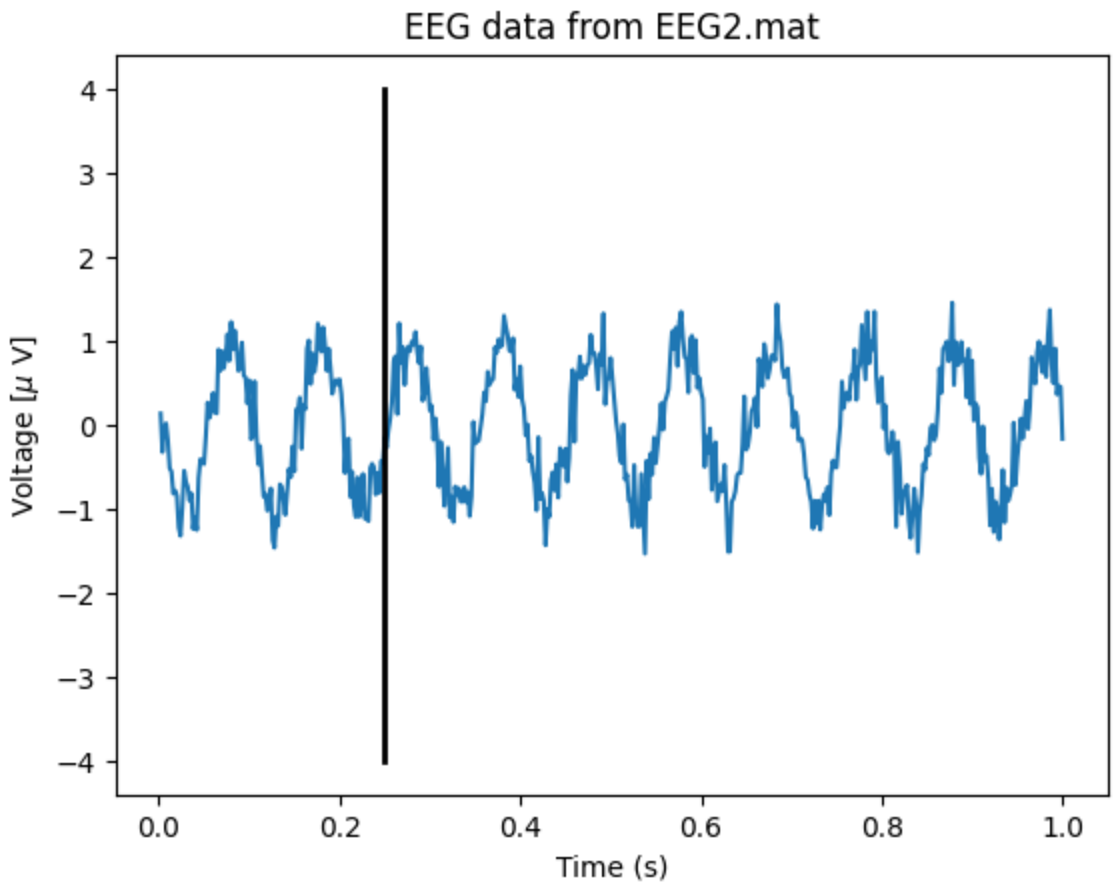
In [ ]:
```
dt = t[1] - t[0]
print('The time between samples in seconds is', dt)
```

```
The time between samples in seconds is 0.002
```

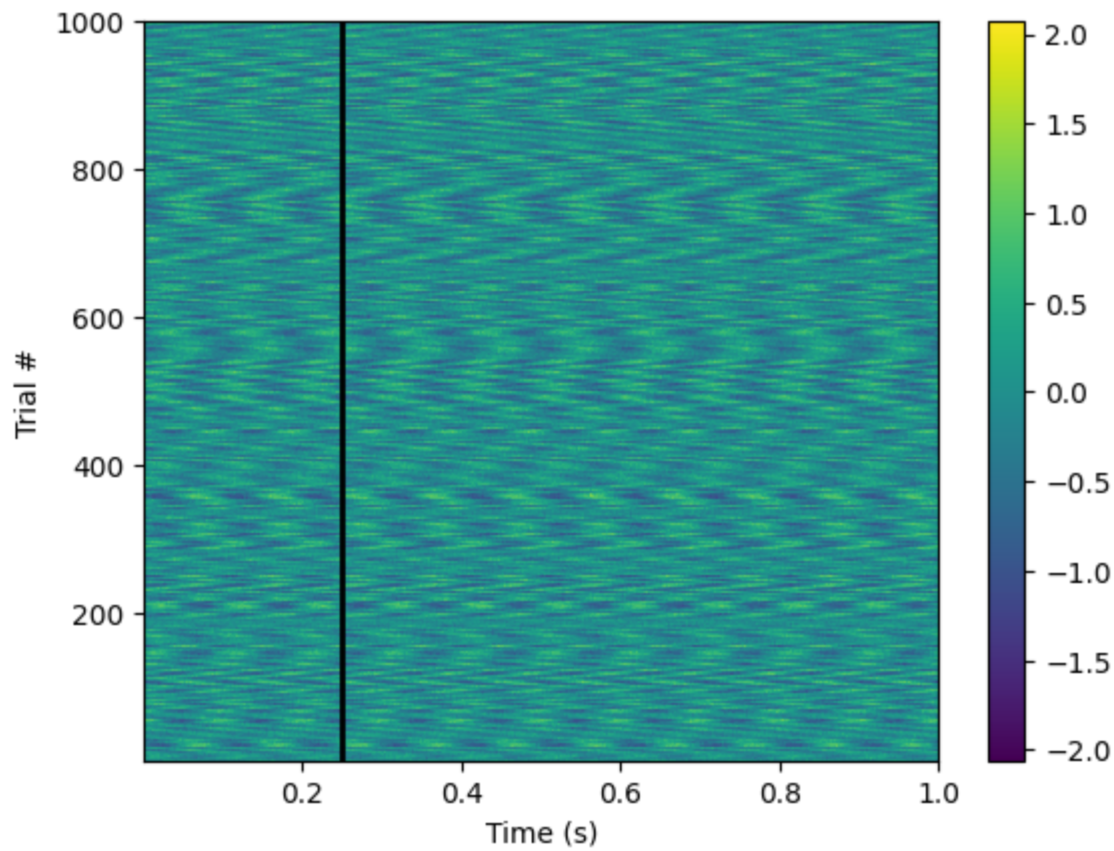
B.) Examine these data. Explain what you observe in pictures and words. From your visual inspection, do you expect to find an ERP in these data?

In [ ]:
```
plt.plot(t, EEG2[0,:])                                    # Plot something useful.
plt.xlabel('Time (s)')                                  # Label the x-axis ...
plt.ylabel('Voltage [$\mu$ V]')                                  # Label the y-axis ...
plt.title('EEG data from EEG2.mat')                                  # Add a title ...
plt.plot([0.25, 0.25], [-4,4], 'k', lw=2)    # Add a vertical line to indicate the s
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x2b2a12efa50>]
```



In [ ]:
```
plt.imshow(EEG2,                                          # Make an *image* of something useful
           extent=[t[0], t[-1], 1, ntrials2],# ... with nice x-limits.
           aspect='auto',                                # ... set aspect ratio
           origin='lower')                               # ... put origin in lower left corner
plt.xlabel('Time (s)')                                          # Label the axes
plt.ylabel('Trial #')
plt.colorbar()                                          # Show voltage to color mapping
plt.vlines(0.25, 1, 1000, 'k', lw=2)                   # Indicate stimulus onset with line
plt.show()
```



Do you expect to find an ERP?

Observation of the EEG plot: Inspecting this plot, it's pretty hard to tell if there is an ERP or not. This data hasn't been processed or analyzed any way just yet (to our knowledge), and making a simple line plot doesn't tell us too much. It might be more useful to visualize it using a spectrogram.

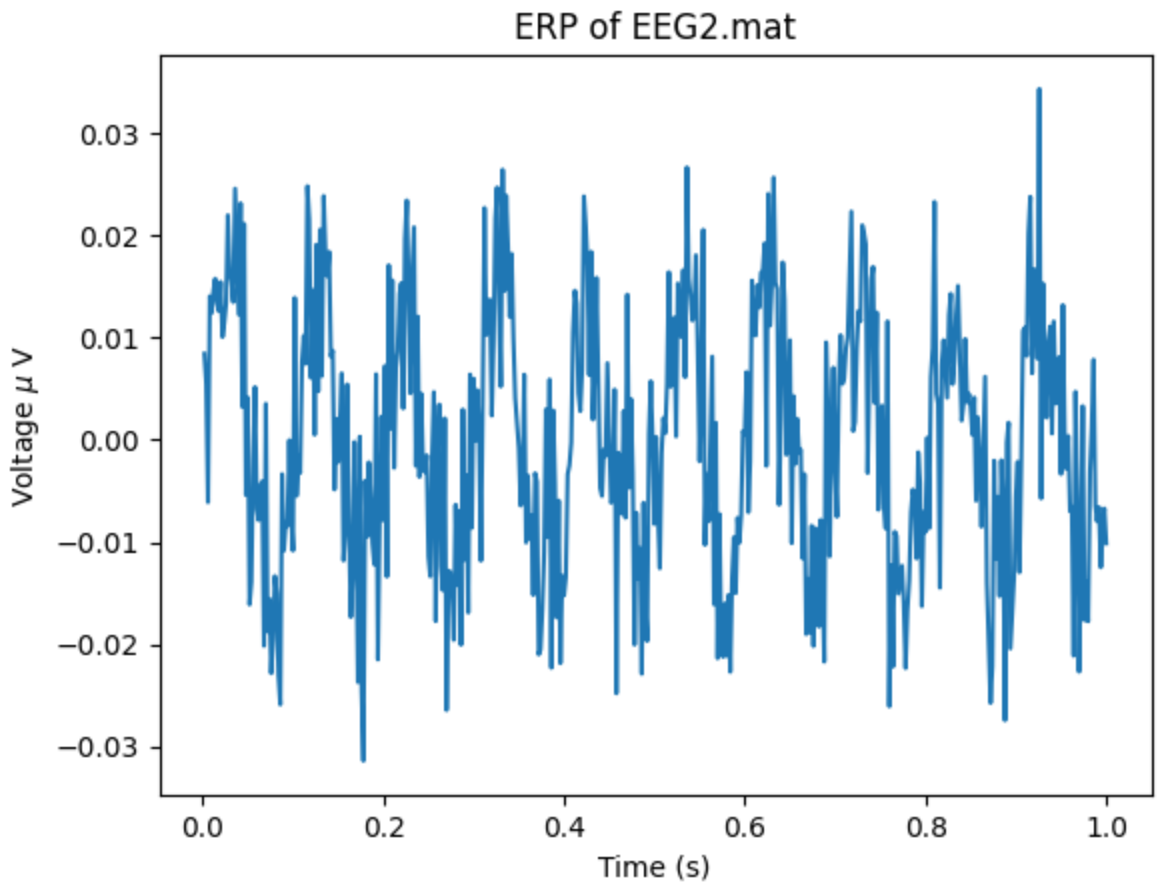
Observation of the spectrogram: Inspecting this data, it looks like there are some changes in charge across time and throughout the trials. These changes are distinct, so there might be an ERP in the data. Unfortunately the changes are spread across a very wide range of times, so perhaps there isn't an ERP?

C.) Compute the ERP for these data, and plot the results. Do you observe an ERP (i.e., times at which the 95% confidence intervals do not include zero)? Include 95% confidence intervals in your ERP plot, and label the axes.

In [ ]:
```
ERP2 = np.mean(EEG2,0)
```

In [ ]:
```
#c.
plt.plot(t, ERP2)                                  # Plot the ERP for each condition.
plt.xlabel('Time (s)')                                  # ... label the axes
plt.ylabel('Voltage $\mu$ V')
plt.title('ERP of EEG2.mat')                                  # ... provide a title
```

Out[ ]:
```
Text(0.5, 1.0, 'ERP of EEG2.mat')
```

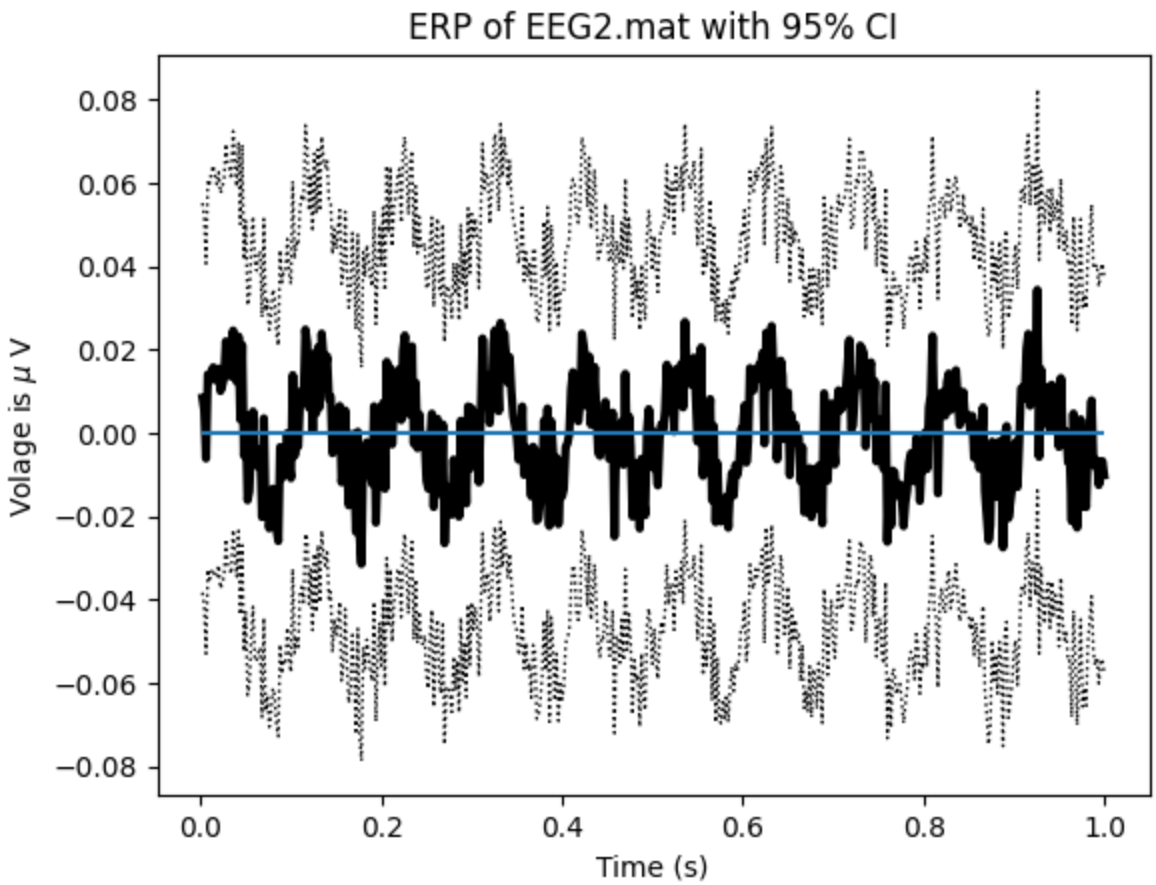


In [ ]:
```
sd2 = np.std(EEG2, 0) #compute the std across trials
print(sd2[0])
mn2 = EEG2.mean(0) #this is the mean across trials
```

```
0.738977040740755
```

In [ ]:
```
sdmn2 = sd2/np.sqrt(ntrials2) #compute the std of the mean
```

In [ ]:
```
plt.plot(t, mn2, 'k', lw=3) #plots the values from the t array on the x-axis and th
plt.plot(t, mn2 + 2 * sdmn2, 'k:', lw=1) #upper CI
plt.plot(t, mn2 - 2 * sdmn2, 'k:', lw=1) #lower CI
plt.xlabel('Time (s)')
plt.ylabel('Volage is $\mu$ V')
plt.title('ERP of EEG2.mat with 95% CI')
plt.hlines(0, t[0], t[-1]); #line at y =0 that represents 0 voltage
```



Explain in a few sentences the results of your analysis, as you would to a colla

The 95% CIs are very wide. This probably indicates a lot of variability in the data - there might be a lot of differences among the collected samples or noise within the measurements. This also indicates some uncertainity in the mean response, because large ranges for an upper and lower 95% CI indicates a wide range of values where the true mean may be located. This wide range probably signifies a lack of statistical signifance if this data represents a different experimental condition than the other data sets. This data probably needs more data collected to reduce variability, or there need to be some changes to the measurement techinique or experimental design to obtain a more precise estimate for the mean ERP.

REUSE code and questions from above for EEG-3 and EEG-4

2A - 2C EEG-3.mat

In [ ]:
```
data3 = loadmat('EEG-3.mat')
```

In [ ]:
```
t = data3['t'][0]
EEG3 = data3['EEG']
```

In [ ]:
```python
print('The shape of EEG from EEG3.mat is', np.shape(EEG3))

ntrials3 = np.shape(EEG3)[0]
print('The number of trials in EEG3.mat is ', ntrials3)

nsamples3 = np.shape(EEG3)[1]
print("The number of samples in EEG3.mat is", nsamples3)
```

```
The shape of EEG from EEG3.mat is (1000, 500)
The number of trials in EEG3.mat is  1000
The number of samples in EEG3.mat is 500
```

A.) What is the time between samples (dt) in seconds?

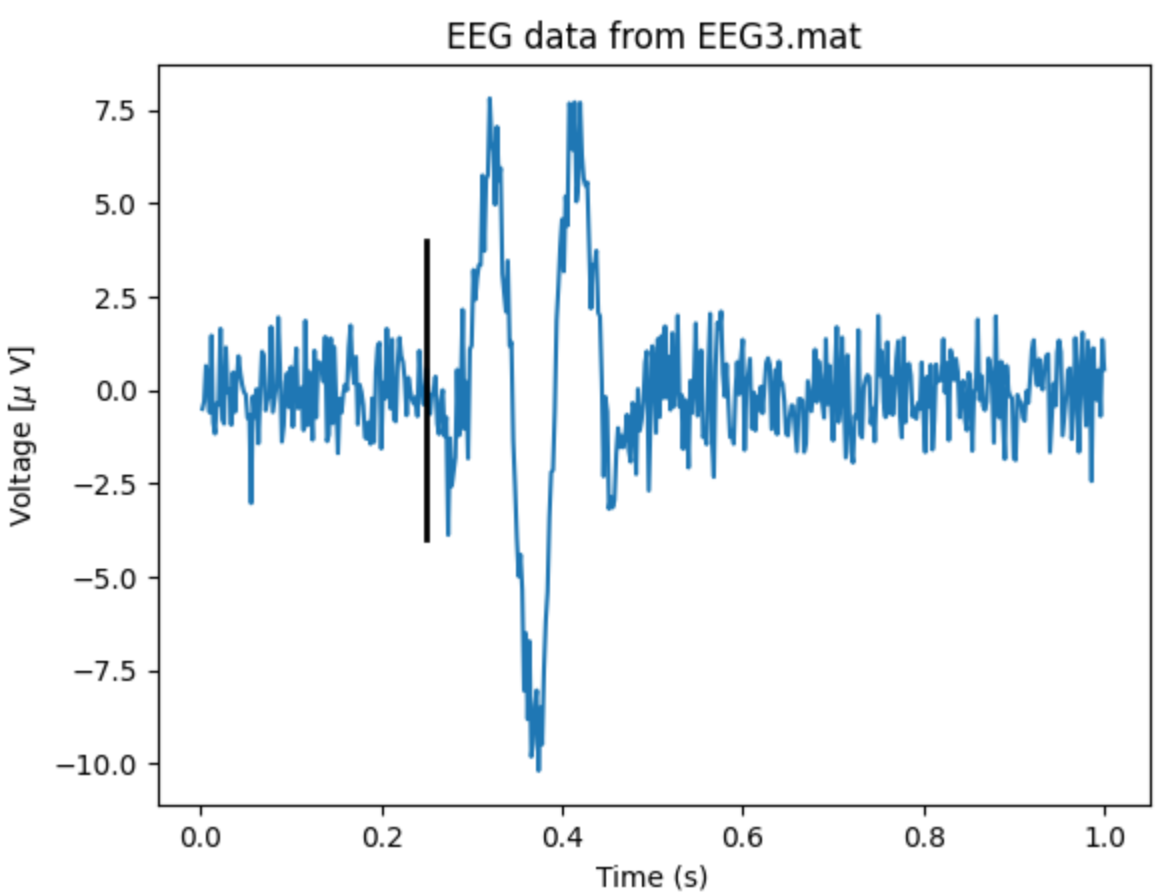
In [ ]:
```python
dt3 = t[1] - t[0]
print('The time between samples in seconds is', dt3)
```

```
The time between samples in seconds is 0.002
```

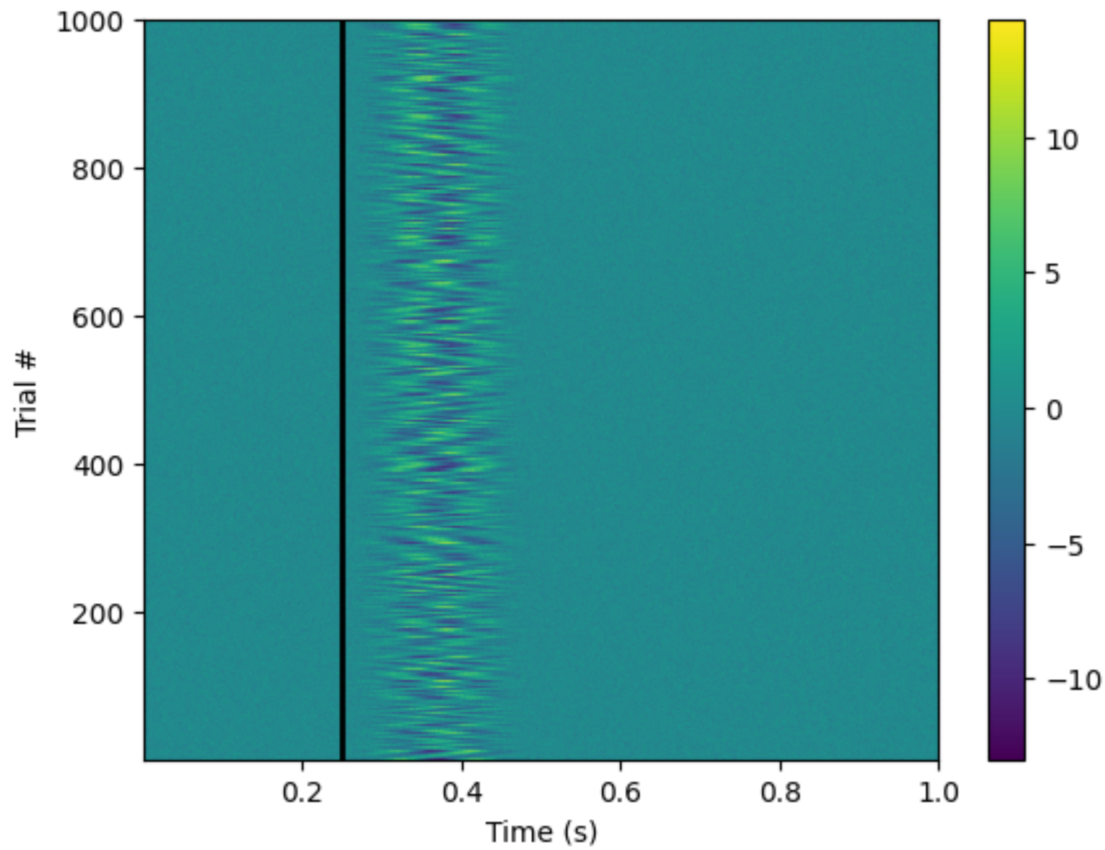
Examine these data. Explain what you observe in pictures and words. From y

In [ ]:
```python
plt.plot(t, EEG3[0,:])                                      # Plot something useful.
plt.xlabel('Time (s)')                                    # Label the x-axis ...
plt.ylabel('Voltage [$\mu$ V]')                                    # Label the y-axis ...
plt.title('EEG data from EEG3.mat')                                    # Add a title ...
plt.plot([0.25, 0.25], [-4,4], 'k', lw=2)   # Add a vertical line to indicate the s
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x2b2a04176d0>]
```



In [ ]:
```
plt.imshow(EEG3,                                  # Make an *image* of something useful
           extent=[t[0], t[-1], 1, ntrials3],# ... with nice x-limits.
           aspect='auto',                        # ... set aspect ratio
           origin='lower')                       # ... put origin in lower left corner
plt.xlabel('Time (s)')                                  # Label the axes
plt.ylabel('Trial #')
plt.colorbar()                                   # Show voltage to color mapping
plt.vlines(0.25, 1, 1000, 'k', lw=2)             # Indicate stimulus onset with line
plt.show()
```



Do you expect to find an ERP?

Observation of the EEG plot: Inspecting the plot, it's pretty hard to tell if there is an ERP or not. There is a sharp uptick from 0.25-0.30 seconds, then a sharp decline before 0.4 seconds, then an uptick back up to 0.45s. There is probably an ERP.

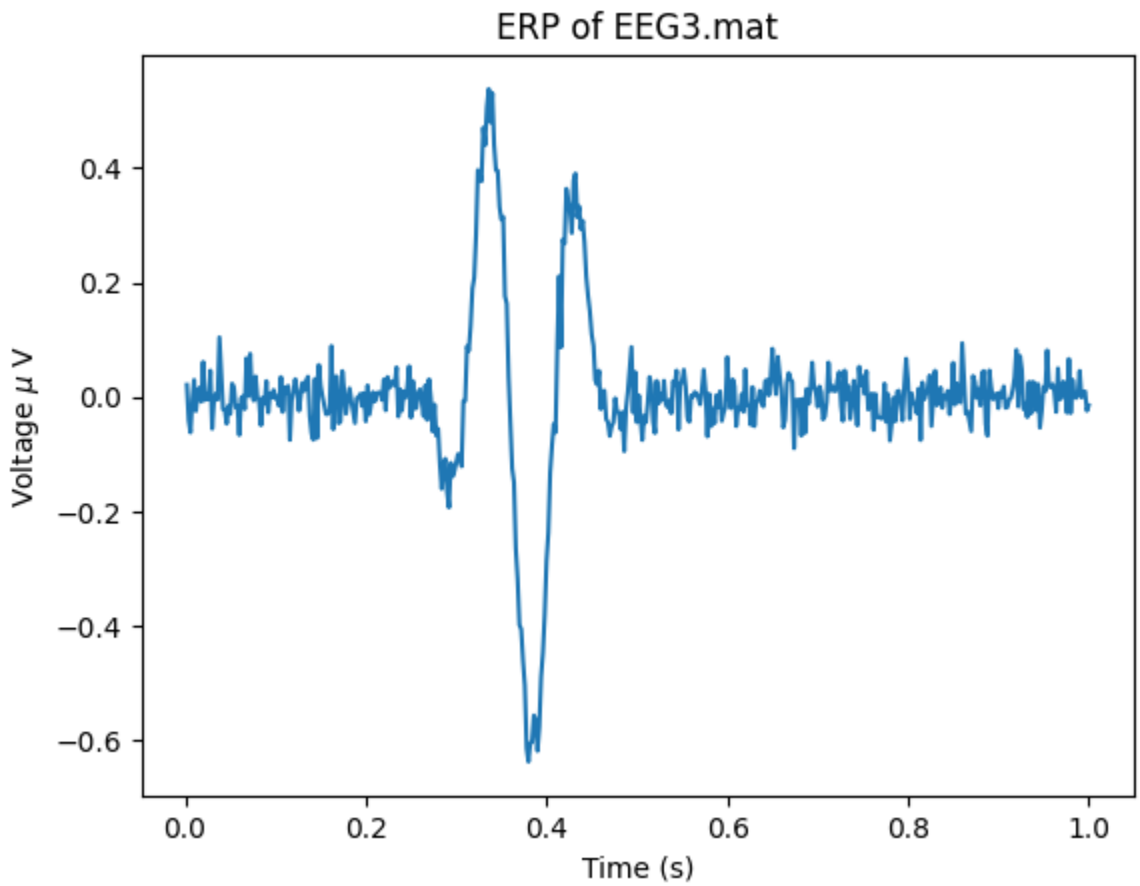
Observation of the spectrogram: Inspecting this data, it looks there are concentrated changes in activity levels from around 0.3 to 0.4 seconds. These changes are distinct, so there is likel yan ERP in the data. More analysis will be needed, but it looks promising so far.

C.) Compute the ERP for these data, and plot the results. Do you observe an ERP (i.e., times at which the 95% confidence intervals do not include zero)? Include 95% confidence intervals in your ERP plot, and label the axes.

In [ ]:
```
ERP3 = np.mean(EEG3,0)
```

In [ ]:
```
plt.plot(t, ERP3)                                          # Plot the ERP for each condition.
plt.xlabel('Time (s)')                                           # ... label the axes
plt.ylabel('Voltage $\mu$ V')
plt.title('ERP of EEG3.mat')
```

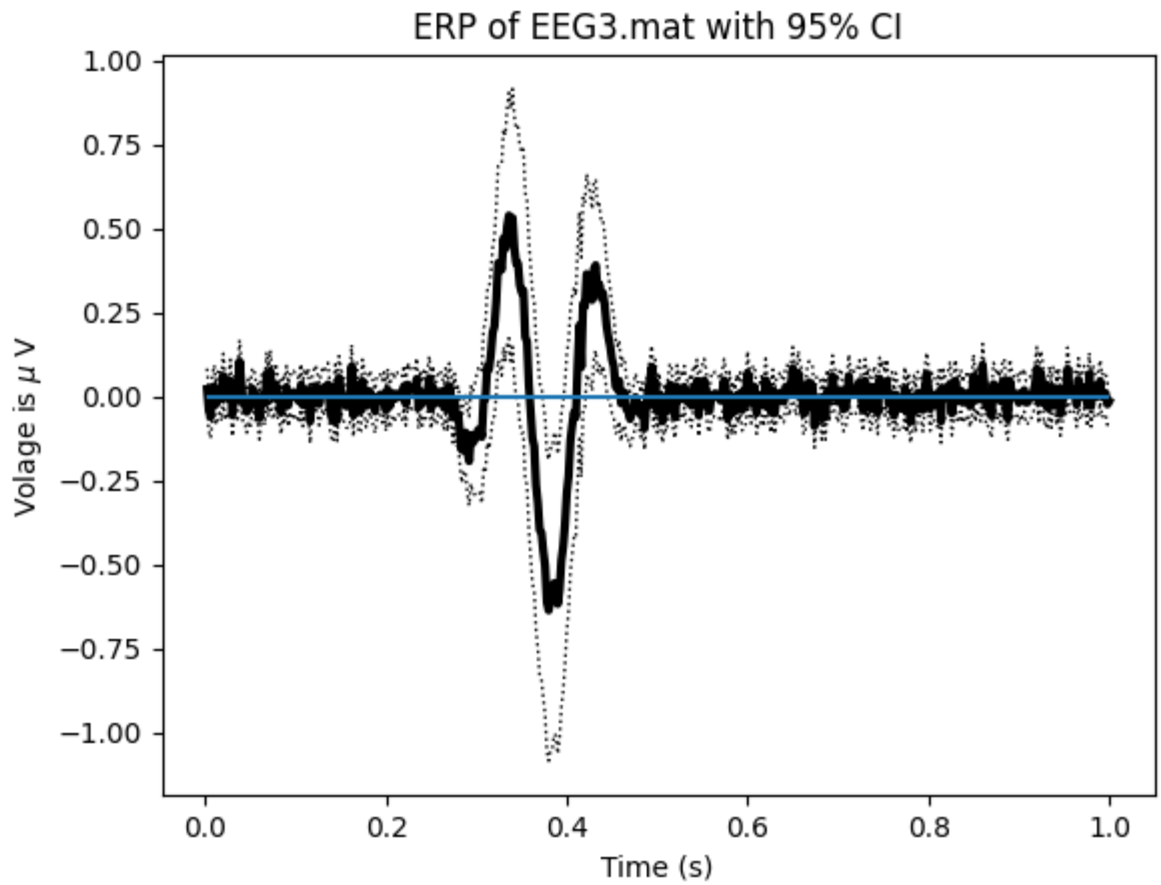
Out[ ]: Text(0.5, 1.0, 'ERP of EEG3.mat')



In [ ]:
```
sd3 = np.std(EEG3, 0) #compute the std across trials
print(sd3[0])
mn3 = EEG3.mean(0) #this is the mean across trials

sdmn3 = sd3/np.sqrt(ntrials3)
```

0.9675149453999141

In [ ]:
```
plt.plot(t, mn3, 'k', lw=3) #plots the values from the t array on the x-axis and th
plt.plot(t, mn3 + 2 * sdmn3, 'k:', lw=1) #upper CI
plt.plot(t, mn3 - 2 * sdmn3, 'k:', lw=1) #lower CI
plt.xlabel('Time (s)')
plt.ylabel('Volage is $\mu$ V')
plt.title('ERP of EEG3.mat with 95% CI')
plt.hlines(0, t[0], t[-1]); #line at y =0 that represents 0 voltage
```



Explain in a few sentences the results of your analysis, as you would to a colla

The 95% CIs are very close to the plotted ERP data. This indicates that there is reduced varaibility, good realiability within the data, and a very narrow range in which the true mean may be located. All of these traits indicates strong evidence for an ERP and that there is relatively low uncertainity in the estimation of the ERP components. Statistical significance in activity difference is likely if this data represents a different epperimental condition than the other data sets.

3A - 3C EEG-4.mat

In [ ]:
```
data4 = loadmat('EEG-4.mat')
```

In [ ]:
```
t = data4['t'][0]
EEG4 = data4['EEG']
print('Shape of EEG from EEG4.mat is', np.shape(EEG4))
```

```
Shape of EEG from EEG4.mat is (1000, 500)
```

In [ ]:
```
ntrials4 = np.shape(EEG4)[0]
print('The number of trials in EEG4.mat is ', ntrials4)

nsamples4 = np.shape(EEG4)[1]
print("The number of samples in EEG4.mat is", nsamples4)
```

```
The number of trials in EEG4.mat is  1000
The number of samples in EEG4.mat is 500
```

A.) What is the time between samples (dt) in seconds?

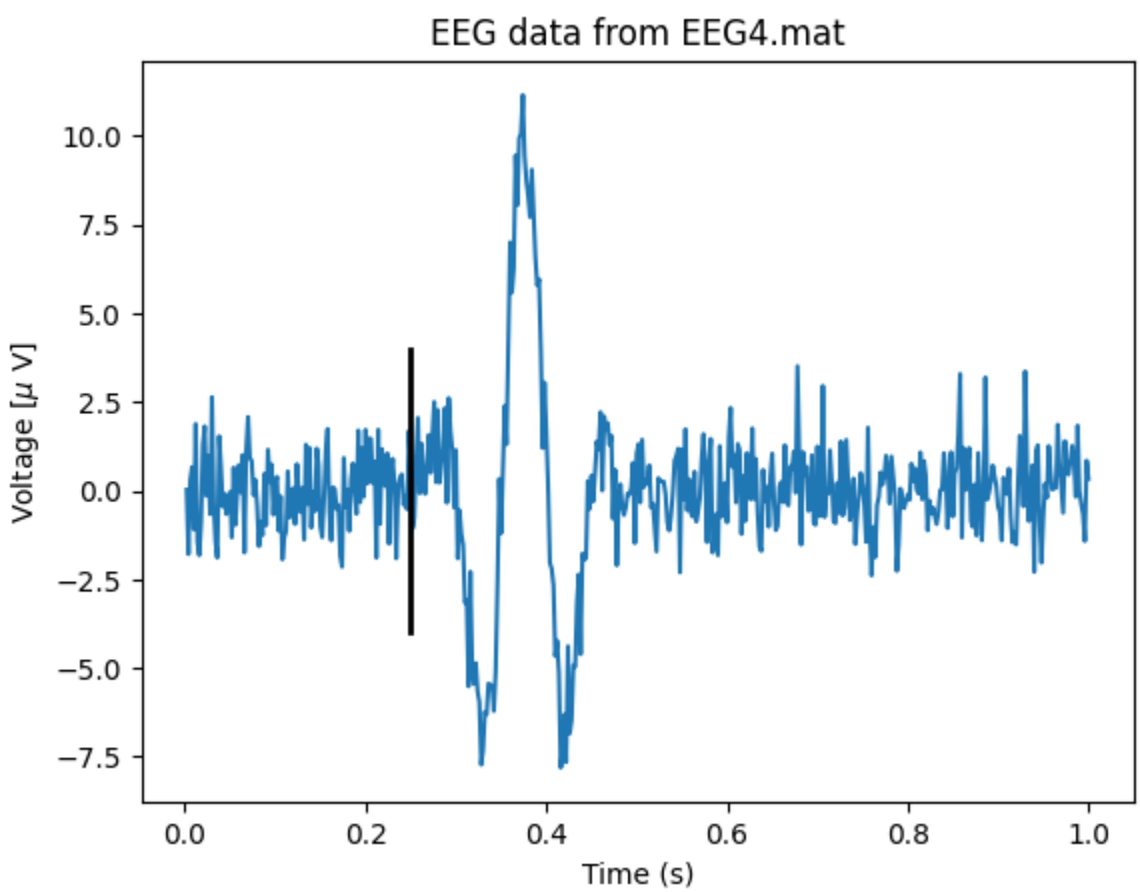
In [ ]:
```
dt4 = t[1] - t[0]
print('The time between samples in seconds is', dt4)
```

```
The time between samples in seconds is 0.002
```

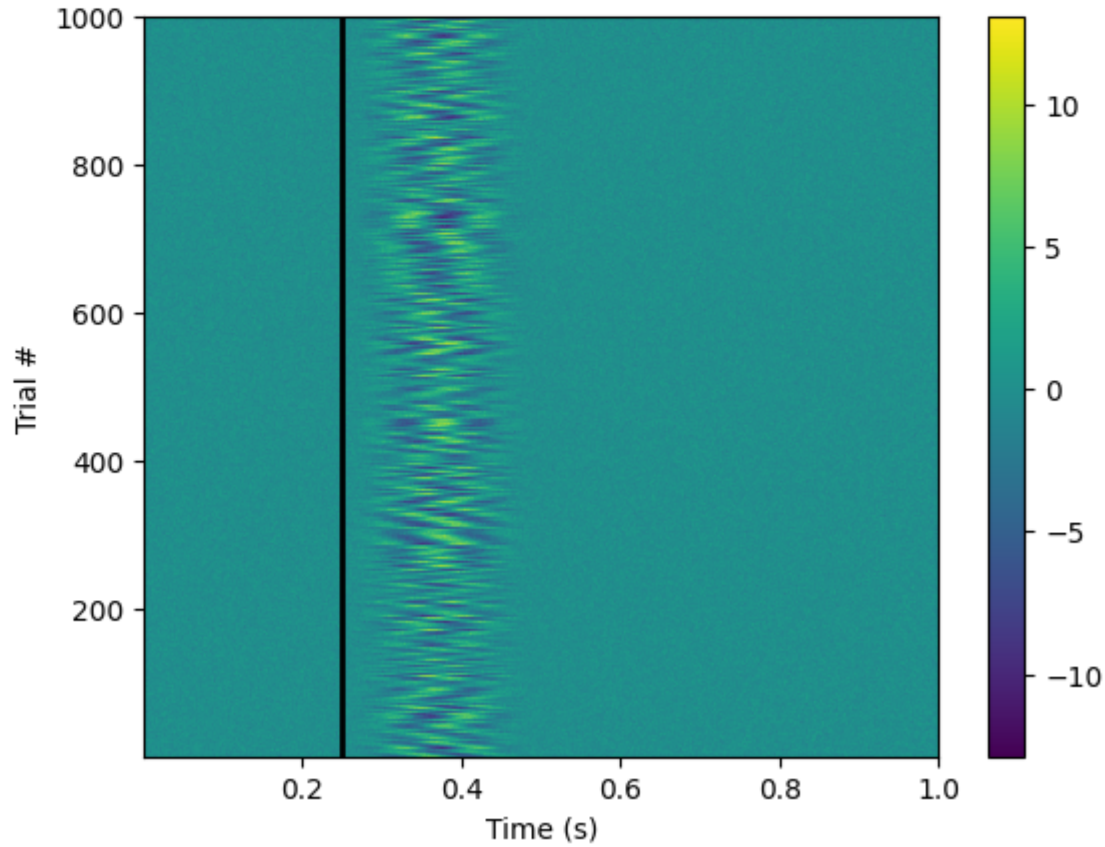
B. Examine these data. Explain what you observe in pictures and words. From your visual inspection, do you expect to find an ERP in these data?

In [ ]:
```
plt.plot(t, EEG4[0,:])                                          # Plot something useful.
plt.xlabel('Time (s)')                                         # Label the x-axis ...
plt.ylabel('Voltage [$\mu$ V]')                                    # Label the y-axis ...
plt.title('EEG data from EEG4.mat')                                    # Add a title ...
plt.plot([0.25, 0.25], [-4,4], 'k', lw=2)
```

Out[ ]: `[<matplotlib.lines.Line2D at 0x2b2a1379e90>]`



In [ ]:
```
plt.imshow(EEG4,                                    # Make an *image* of something useful
           extent=[t[0], t[-1], 1, ntrials4],# ... with nice x-limits.
           aspect='auto',                          # ... set aspect ratio
           origin='lower')                         # ... put origin in lower left corner
plt.xlabel('Time (s)')                                    # Label the axes
plt.ylabel('Trial #')
plt.colorbar()                                     # Show voltage to color mapping
plt.vlines(0.25, 1, 1000, 'k', lw=2)               # Indicate stimulus onset with line
plt.show()
```



Do you expect to find an ERP?

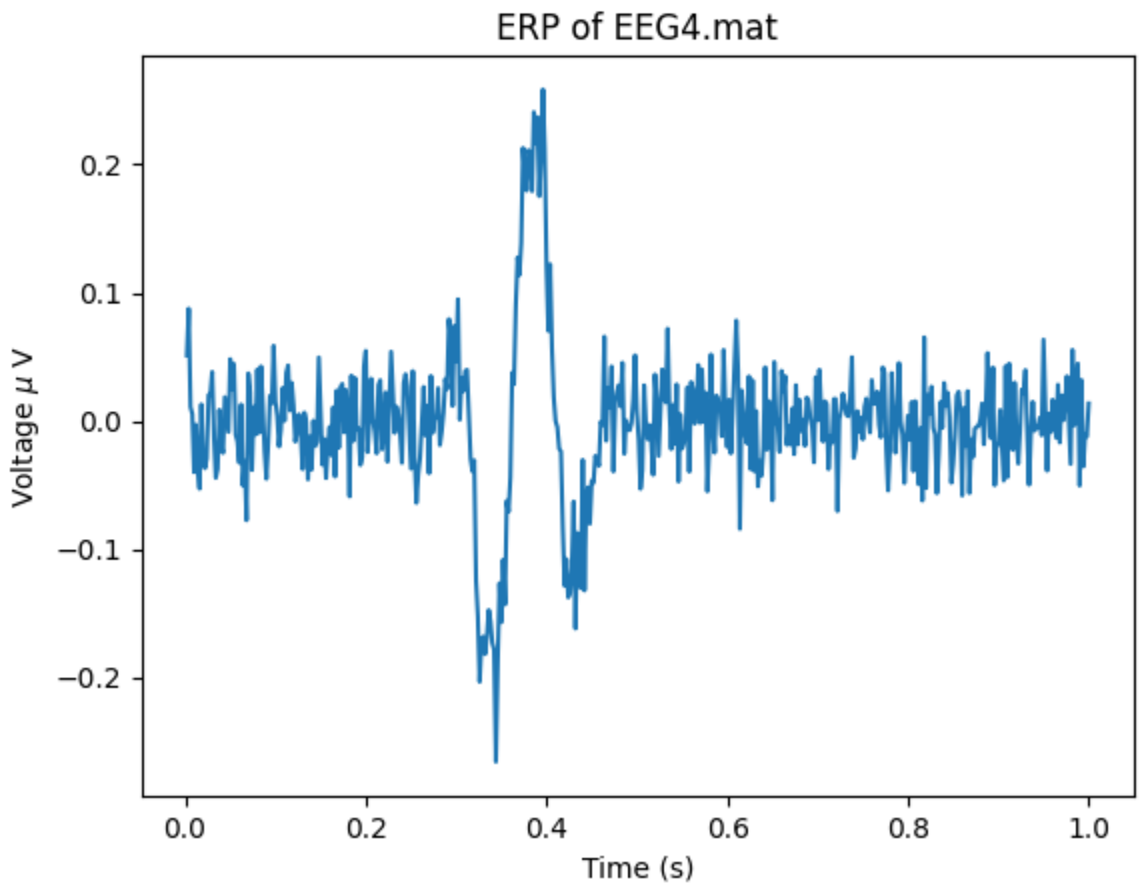
From my observation of the EEG plot and the spectrogram, I expect to find an ERP. In the spectrogram, it is very clear that significant changes in neural activity are occurring during a specific time point.

C. Compute the ERP for these data, and plot the results. Do you observe an ERP (i.e., times at which the 95% confidence intervals do not include zero)? Include 95% confidence intervals in your ERP plot, and label the axes.

In [ ]:
```
ERP4 = np.mean(EEG4,0)
```

In [ ]:
```
#c.
plt.plot(t, ERP4)                                    # Plot the ERP for each condition.
plt.xlabel('Time (s)')                                      # ... label the axes
plt.ylabel('Voltage $\mu$ V')
plt.title('ERP of EEG4.mat')
```

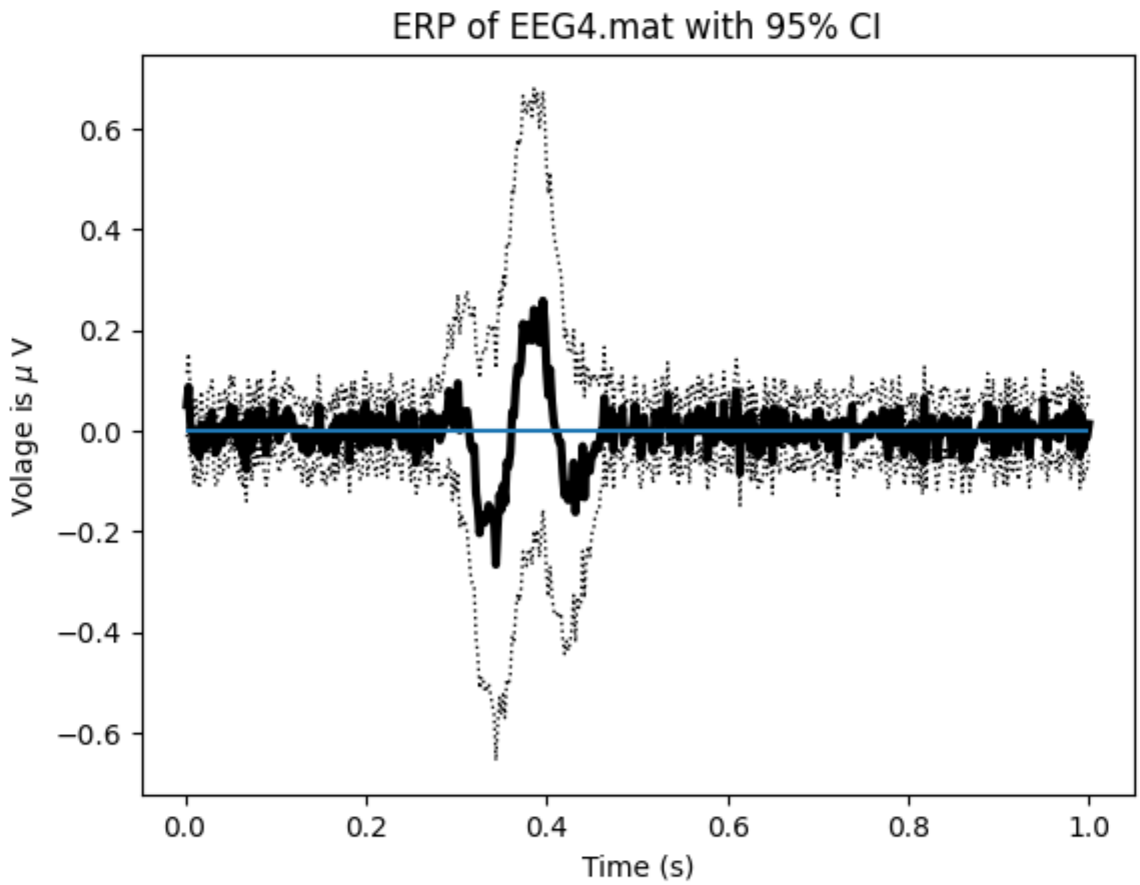
Out[ ]:
```
Text(0.5, 1.0, 'ERP of EEG4.mat')
```



In [ ]:
```
sd4 = np.std(EEG4, 0) #compute the std across trials
print(sd4[0])
mn4 = EEG4.mean(0) #this is the mean across trials
sdmn4 = sd4/np.sqrt(ntrials4)
```

```
1.0045701995581138
```

In [ ]:
```
plt.plot(t, mn4, 'k', lw=3) #plots the values from the t array on the x-axis and th
plt.plot(t, mn4 + 2 * sdmn4, 'k:', lw=1) #upper CI
plt.plot(t, mn4 - 2 * sdmn4, 'k:', lw=1) #lower CI
plt.xlabel('Time (s)')
plt.ylabel('Volage is $\mu$ V')
plt.title('ERP of EEG4.mat with 95% CI')
plt.hlines(0, t[0], t[-1]);
```



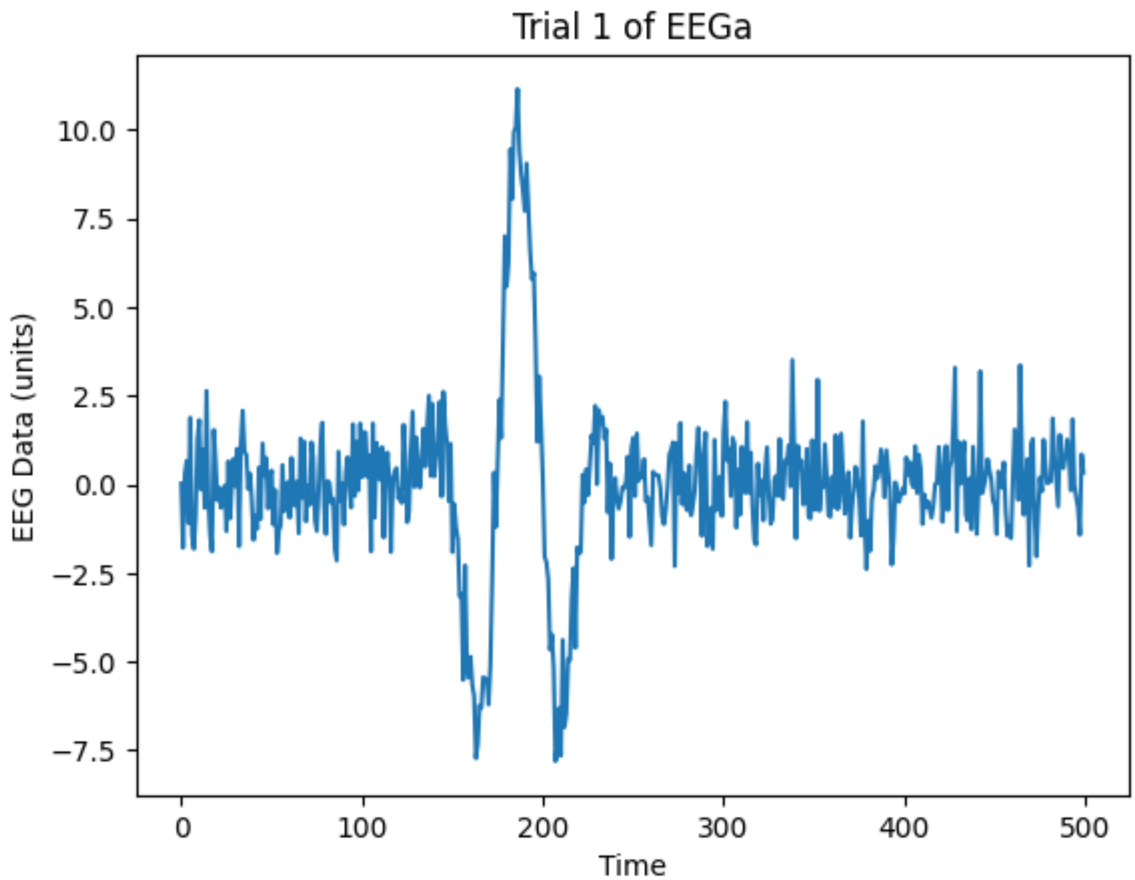
Explain in a few sentences the results of your analysis, as you would to a colla

The 95% CIs are very close around the stable neural activity, but the upper and lower CIs are very wide around the potential ERP. This probably reflects some uncertainity about the exact characterisitcs of the ERP component - perhaps data regarding it's latency or amplitude is high in variability (same stimuli evoking widely different changes in neural activity across participants/animals). This might also reflect that the ERP is not statistically significant or reliably present.

4.) In the previous question, you considered the dataset EEG-4.mat and analyzed these data for the presence of an ERP. To do so, you (presumably) averaged the EEG data across trials. The results may have surprised you . . . Modify your analysis of these data (in some way) to better illustrate the appearance (or lack thereof) of an evoked response. Explain “what's happening” in these data as you would to a colleague or experimental collaborator.

In [ ]:
```
plt.figure()
plt.plot(EEG4[0,:])
plt.ylabel('EEG Data (units)')
plt.xlabel('Time')
plt.title('Trial 1 of EEGa')
```

Out[ ]:
```
Text(0.5, 1.0, 'Trial 1 of EEGa')
```



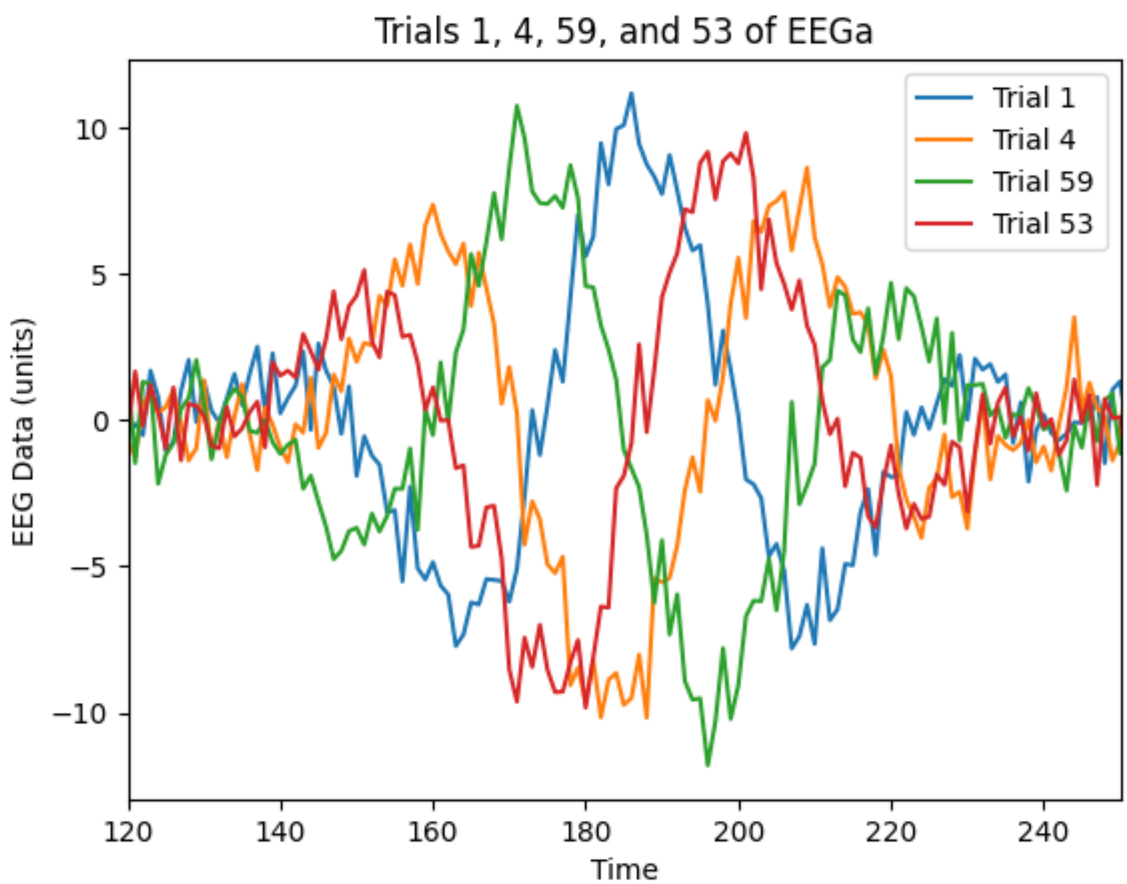
Zooming in to look at one singular trial (trial 1) isn't the most helpful, but it does give some ideas. We could try plotting multiple trials on top of one another, and that may be more insightful. We could also zoom in to the time we think the ERP is occuring at (~ 100 - 200 seconds) to get a better picture of how the dynamics are differing across trials during the potential ERP.

In [ ]:

```
plt.figure()
plt.plot(EEG4[0,:], label='Trial 1')
plt.plot(EEG4[5,:], label='Trial 4')
plt.plot(EEG4[60,:], label='Trial 59')
plt.plot(EEG4[54,:], label='Trial 53')
plt.ylabel('EEG Data (units)')
plt.xlabel('Time')
plt.xlim(120, 250)
plt.legend()
plt.title('Trials 1, 4, 59, and 53 of EEGa')
```

Out[ ]:

```
Text(0.5, 1.0, 'Trials 1, 4, 59, and 53 of EEGa')
```



Having plotted multiple trials ontop of one another and zooming in around the suspected ERP timespan, we can now see more clearly why the CIs are so large for the potential ERP. Some of the trials peak in activity before other trials, while some trials never peak as high as other trials. There seems to be some small, but still noticeable differences in timing and activity (units).

It may be possible that the EEG-4 dataset contains multiple ERPs that are being generated by different stimuli/conditions, variability in cognitive processes (like attention), or by individual differences within the experimental population (adolescent vs adult, adhd vs no adhd, etc.).

5.) Compare the datasets EEG-3.mat and EEG-4.mat studied in the previous problems. Use a bootstrap procedure to test the hypothesis that the evoked response is significantly different in the two datasets.

In [ ]:
```
mn3 = np.mean(EEG3,0)           # Determine ERP for condition A
mn4 = np.mean(EEG4,0)           # Determine ERP for condition B
mnD = mn3 - mn4                 # Compute the differenced ERP
stat = np.max(np.abs(mnD))      # Compute the statistic
print('stat = {:.4f}'.format(stat))
```

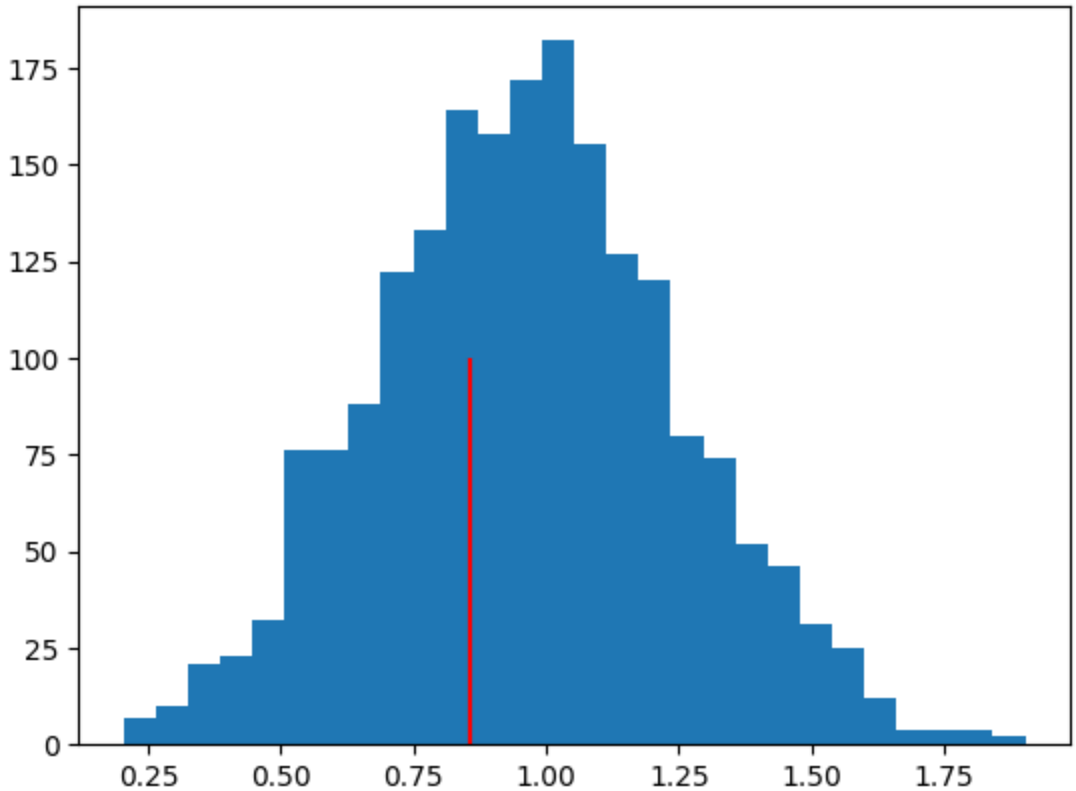
```
stat = 0.8550
```

The statistic (stat) represents the maximum absolute difference in the event-related potential (ERP) between EEG3 and EEG4.

In [ ]:
```
EEG = np.vstack((EEG3, EEG4))  # Step 1. Merge EEG data from all trials
np.random.seed(123)            # Fix seed for reproducibility

N_resample = 2000;
stat0 = np.zeros(N_resample)
for k in np.arange(0,N_resample):

    i      = np.random.randint(0, 1000, size = ntrials3)    # Create resampled indi
    EEG0   = EEG3[i,:]    # Create a resampled EEG for "condition A".
    ERP0_3 = EEG0.mean(0)    # Create a resampled ERP for "condition A".

    i      = np.random.randint(0, 1000, size=ntrials4)    # Create resampled indice
    EEG0   = EEG4[i,:]    # Create a resampled EEG for "condition B".
    ERP0_4 = EEG0.mean(0)     # Create a resampled ERP for "condition B".

    MnDD = ERP0_3 - ERP0_4
    stat0[k] = np.max(np.abs(MnDD))
```

stat0 is the distribution of the maximum absolute ERP differences between the resampled EEG3 (condition a) and EEG4 (condition b).

In [ ]:
```
# Plot the observed stat vs the resampled stat.
plt.figure()
plt.hist(stat0, bins='auto')
plt.vlines(stat, 0, 100, color = "red")
np.size(np.where(stat0>stat)) / stat0.size
plt.show()

p_value = np.size(np.where(stat0>stat)) / stat0.size
print(f'p-value = {p_value:.4f}')
```




```
p-value = 0.6425
```

The stat0 distribution is somewhat symmetric. When we look at the values that are greater than stat, we see that the resulting p-value is 0.6425. This indicates that there is no strong evidence that there is a significant difference between the ERP of the two datasets. In this code, we used the p-value to represent the probability of observing a result as extreme as our stat value in our resampled datasets. Our p-value of 0.6425 indicates that the observed difference between the two datasets is not statistical significant and does not support the belief that the ERPs of the two datasets are significantly different.