

## Introduction to Data Analysis for the EEG

Version 0.3

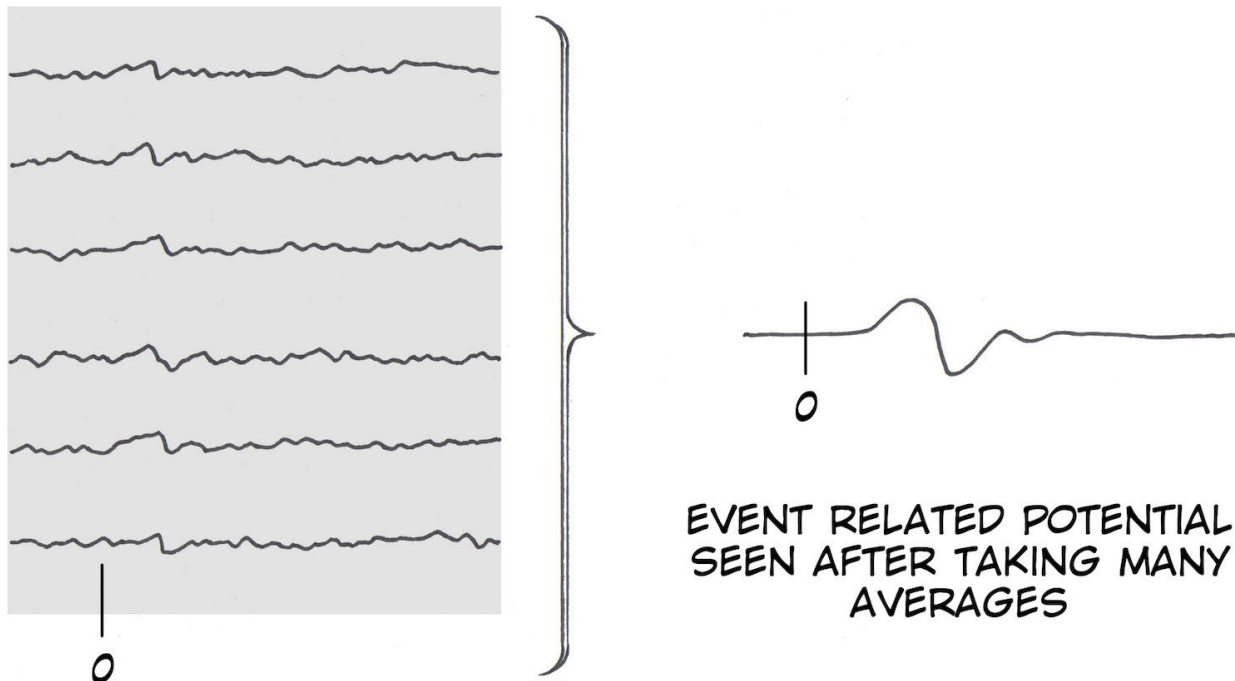
Analyzing electroencephalograms (EEGs) is trickier than analyzing the action potentials from human muscles (using our Muscle SpikerBox) or cockroach leg nerves (using our Neuron SpikerBox). When recording EEG, we are observing the collective electrical activity of many neurons in your brain at the same time, filtered through your skull and scalp. Thus, we cannot simply do a “spikes per second” count like we do with neurons or muscles, or even a “beats per minute” and “interbeat interval” count like we do with heart activity. Analyzing EEGs is mathematically complex, as it is a signal with multiple frequencies changing over time, often with a lot of embedded noise.



The two most common analysis techniques are 1) averaging the signal over many trials to see an “evoked potential,” and 2) building a spectrogram to see frequency changes over time.

In an evoked potential, we average the EEG signal over many repeated stimulus presentations and get an average “evoked response.” This is what we do in our P300 experiment.

<https://backyardbrains.com/experiments/p300>



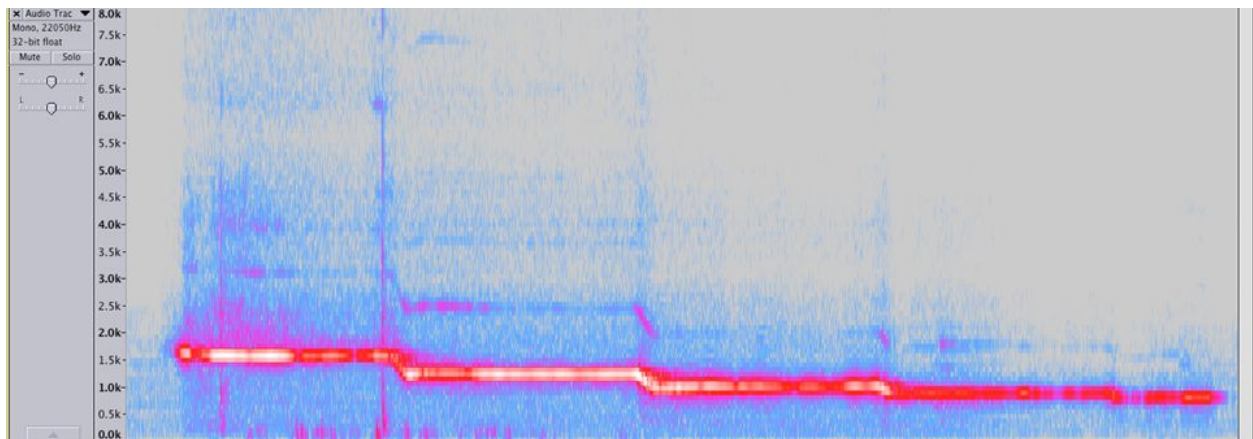
In contrast, building a spectrogram decomposes a complex signal into its many frequency components often using a “[Fourier Transform](#),” which states that any complex signal can be broken down into distinct sine and cosine waves.



**JEAN-BAPTISTE JOSEPH FOURIER**  
GRENOBLE, FRANCE, 1768 - 1830

$$f(t) = \int_0^{\infty} [a(\lambda) \cos 2\pi \lambda t + b(\lambda) \sin 2\pi \lambda t] d\lambda$$

This Fourier Transform is what data analysis programs use to generate spectrograms, which show the frequency content of a signal changing over time. For example, below is an example of a spectrogram when one of us whistled, and we decreased the pitch (made it lower-sounding) as we whistled.

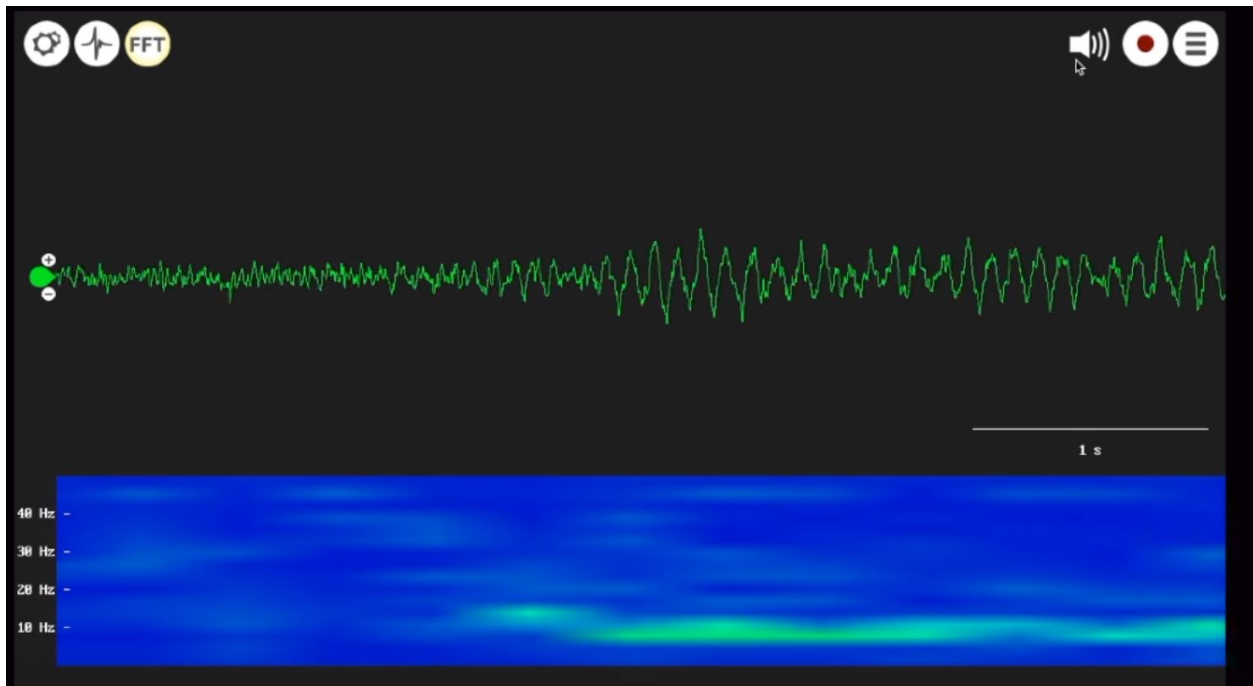


Notice that the whistle frequency started at 1500 Hz and dropped to 500 hz.

We can similarly do this analysis in an EEG recording on our Spike Recorder Software. When you are doing an EEG recording, you can push the “FFT” button (fast fourier transform).

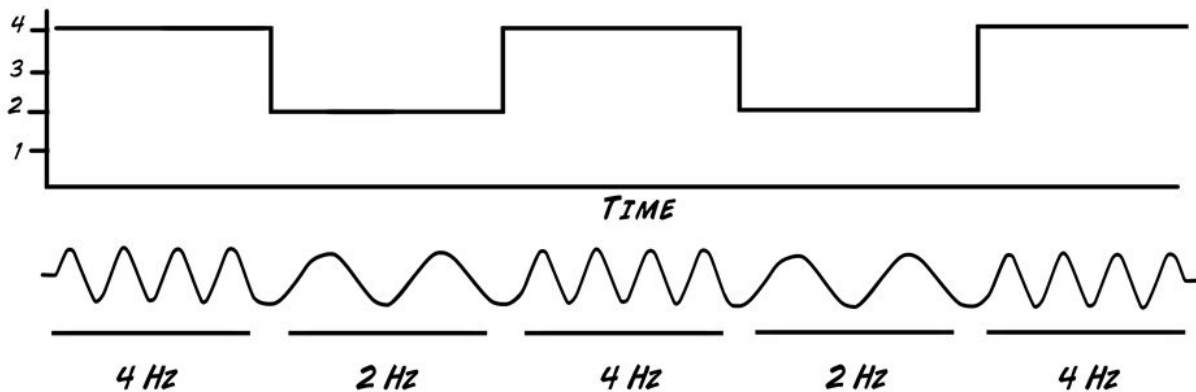


In Spike Recorder, a spectrogram will then appear below the moving EEG trace.



This was a recording over the visual cortex, and when we closed our eyes, the prominent 8 Hz alpha waves appeared, which you can see in the raw trace as well as in the spectrogram. We can then measure the “power” of the alpha waves over time by analyzing the spectrogram.

For example, in a simple example, if you imagine a signal alternating between 2 and 4 Hz, you can eventually generate the figure below.



At the moment our Spike Recorder software does not provide any quantitative analysis of the EEG after you record it (stayed tuned), thus we will now try to generate a figure using the popular analysis software Matlab. This software can be intimidating to use, but just be patient. We will do an example below.

## Extracting Frequency Bands for EEG Analysis - Matlab test

Use the Test Recording of our Visual Cortex Recording at Backyard Brains

[https://backyardbrains.com/experiments/files/EEG\\_Alpha\\_SampleData.zip](https://backyardbrains.com/experiments/files/EEG_Alpha_SampleData.zip)

(this file is also referenced in the EEG experiment page)

<https://backyardbrains.com/experiments/eeg>

You can download and install a free 30 day trial version of Matlab.

[https://www.mathworks.com/programs/trials/trial\\_request.html](https://www.mathworks.com/programs/trials/trial_request.html)

When installing, select the “Signal Processing & Communications” Additional Package

Also download the “curve fitting toolbox” trial after previously installing Matlab

<https://www.mathworks.com/products/curvefitting.html>

Once all is installed, you can begin plotting your data

### To plot the spectrogram:

1) In the file menu, select import data, and select the .wav recording of the sample data, two variables will appear in your workspace - data, and fs

2) We need to downsample the data from 10,000 Hz to 500 Hz to be less demanding on the computer, so we use this line

```
data_new = downsample(data,20);
```

An alternative, that will reduce “aliasing errors”, is the decimate command

<https://www.mathworks.com/help/signal/ug/downsampling-aliasing.html>

```
data_new = decimate(data,20);
```

With both methods we now having a file with a sampling rate of 500 Hz

3) We will now generate the spectrogram with the following command:

```
spectrogram(data_new,256,250,256,500,'yaxis');
```

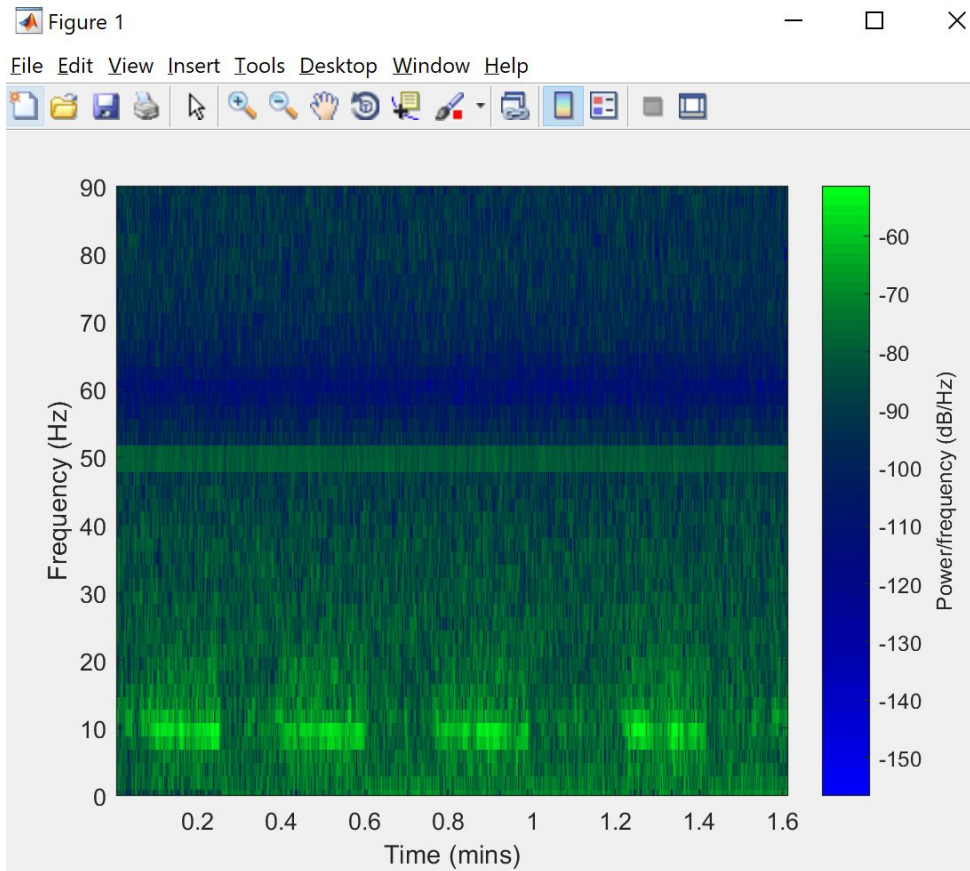
data\_new is the data, 256 is the number of samples for the frequency analysis, 250 is the number of samples for the overlap in the moving window, 256 is related to the smoothness of the fourier transform (dft), 500 is the sample rate, and 'yaxis' is a formatting command

4) We will limit the y-axis to focus on the frequencies of interest, and will change the colors a bit.

```
ylim([0 90]);
```

```
colormap(winter);
```

```
brighten(-0.7);
```



The lighter green spots at ~8-10 Hz are when our eyes were closed and there was increased alpha wave power.

### To Select and Plot Alpha Power as well.

1) We will make a new figure

*figure;*

2) We will now generate stored data of the spectrogram

```
[d,f,t]=spectrogram(data_new,256,250,256,500,'yaxis');
```

d is the data, f is the frequency, y t is the time in seconds

3) Fourier transforms generate complex numbers, so we will convert them to real numbers

```
d_real=real(d);
```

3) Looking at f, we see elements 5-7 represent the frequencies of 7.8 Hz, 9.8 Hz, and 11.7 Hz, so we will extract these out of the matrix d\_real and average them

```
AlphaRange = mean(d_real((5:7),:),1);
```

Note the 5:7 line is extracting the alpha waves centered at 8-12 hz, for other frequencies you need to change this, e-mail us if you need help.

4) We need to smooth the data, and we will use a moving average. We will also normalize the data with an absolute value (abs) command.

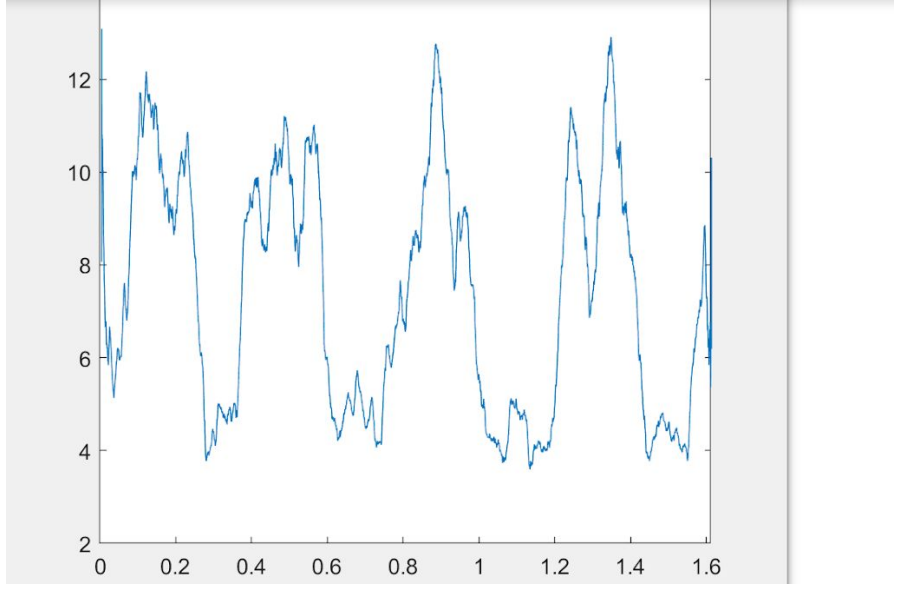
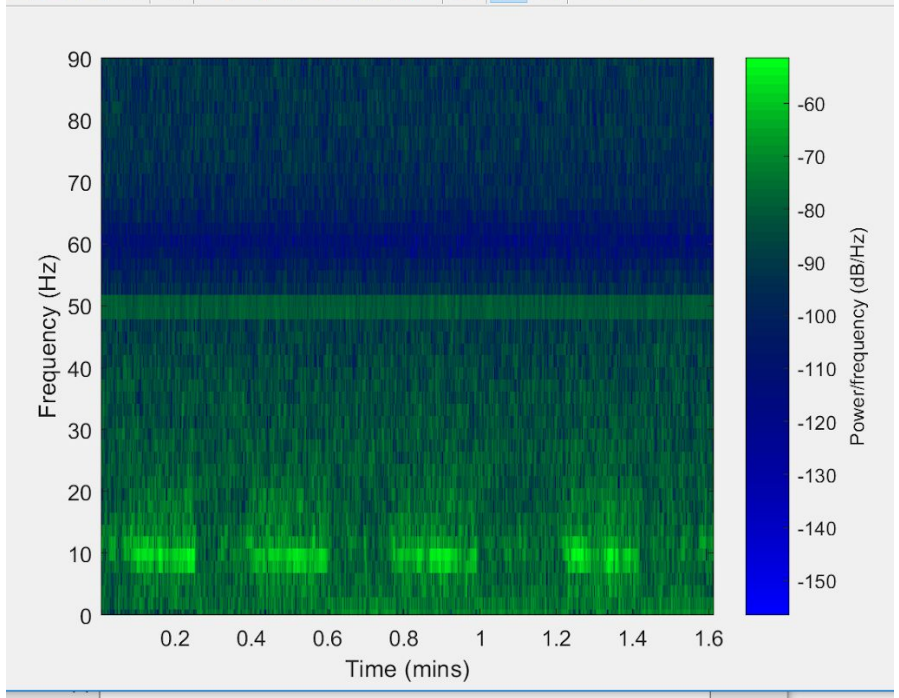
```
Smoothed_AlphaRange=smooth((abs(AlphaRange)),250,'moving');
```

5) We are now ready to plot the data. We will divide t by 60 to convert to minutes, and scale the x-axis a bit.

```
plot(t/60,Smoothed_AlphaRange);  
xlim([0 max(t/60)]);
```

You can see the new figure below the spectrogram on the following page. We now have a moving quantitative measure of alpha wave power you can use for your data analysis.

Want to analyze more frequencies? Go below the figure on the next page for more details. Want to cut and paste into excel? we cover that too!





To analyze select frequencies of the EEG, we will look at the matrix “f” that we created when we extracted the data from the spectrogram

1	0.0	31	58.6	90	173.8	121	234.4
2	2.0	32	60.5	91	175.8	122	236.3
3	3.9	33	62.5	92	177.7	123	238.3
4	5.9	34	64.5	93	179.7	124	240.2
5	7.8	35	66.4	94	181.6	125	242.2
6	9.8	36	68.4	95	183.6	126	244.1
7	11.7	37	70.3	96	185.5	127	246.1
8	13.7	38	72.3	97	187.5	128	248.0
9	15.6	39	74.2	98	189.5	129	250.0
10	17.6	40	76.2	99	191.4		
11	19.5	41	78.1	100	193.4		
12	21.5	42	80.1	101	195.3		
13	23.4	43	82.0	102	197.3		
14	25.4	44	84.0	103	199.2		
15	27.3	45	85.9	104	201.2		
16	29.3	46	87.9	105	203.1		
17	31.3	47	89.8	106	205.1		
18	33.2	48	91.8	107	207.0		
19	35.2	49	93.8	108	209.0		
20	37.1	50	95.7	109	210.9		
21	39.1	51	97.7	110	212.9		
22	41.0	52	99.6	111	214.8		
23	43.0	53	101.6	112	216.8		
24	44.9	54	103.5	113	218.8		
25	46.9	55	105.5	114	220.7		
26	48.8	56	107.4	115	222.7		
27	50.8	57	109.4	116	224.6		
28	52.7	58	111.3	117	226.6		
29	54.7	59	113.3	118	228.5		
30	56.6	60	115.2	119	230.5		
				120	232.4		

As you can see, “f” is a 129 digit matrix starting from 0 Hz (position 1) to 250 Hz (position 129). When we previously analyzed the alpha waves, we were interesting in the frequencies around 8-10 Hz, so we choose position 5 (7.8 Hz), position 6 (9.8 Hz), and position 7 (11.7 Hz). Hence the command we used above

```
AlphaRange = mean(d_real((5:7),:),1);
```

Notice the 5:7 part? That is where we are selecting the alpha band of the EEG.



If we wanted to analyze the beta frequencies from 12 - 28 Hz, we would want to select position 8 (13.7 Hz), position 9 (15.6 Hz), position 10 (17.6 Hz), position 11 (19.5 Hz), position 12 (21.5 Hz), and position 13 (23.4 Hz). And our new code to plot the beta waves power would now be:

```
BetaRange = mean(d_real((8:13),:),1);
Smoothed_BetaRange=smooth((abs(BetaRange)),250,'moving');
plot(t/60,Smoothed_BetaRange);
xlim([0 max(t/60)]);
```

If you look closely, all we changed were the variable names (to make it less confusing for us), and the numbers “8:13”

You can also click on Smooth\_BetaRange in your variable space, and copy it into excel. To get time in minutes, you can use the command.

```
t_minutes = t'/60;
```

Note the excel spreadsheet can get large, a 1.6 minute recording has 8048 rows, but now you can create means around specific times when your subject was doing something.

	A	B	C
1	Beta Power	Time	
2	0.006459798	0.004266667	
3	0.004320389	0.004466667	
4	0.003177486	0.004666667	
5	0.004372741	0.004866667	
6	0.004490681	0.005066667	
7	0.004200117	0.005266667	
8	0.004306215	0.005466667	
9	0.004275011	0.005666667	
10	0.004083283	0.005866667	
11	0.003975788	0.006066667	
12	0.004249287	0.006266667	
13	0.004082755	0.006466667	
14	0.00417715	0.006666667	
15	0.004267843	0.006866667	
16	0.004240019	0.007066667	
17	0.004282	0.007266667	
18	0.004244724	0.007466667	
19	0.004279199	0.007666667	

**Stay tuned! We will fine tune all this as we improve our data tools.**