A primer on: Spectral power analyses

V. Müller Ewald for Hacky hour

Parker Lab 7.30.21

Getting to know you

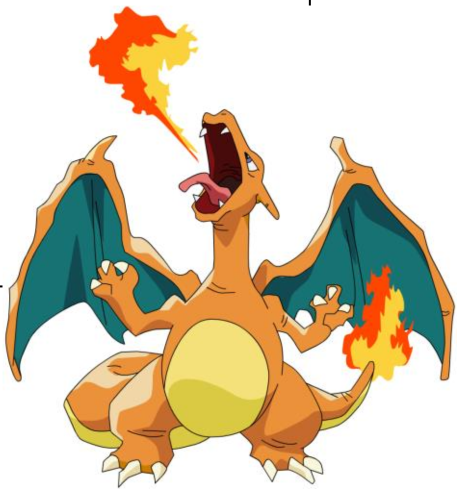
Already covered:
- Pre-processing ✓
- ERPs ✓
- Fast Fourier transform ✓

Please pull out 3 pieces of paper and a pen/pencil:
-> this will be relevant later for understanding the code
-> Not a lame ice-breaker

Getting to know you

Please send help

1st year graduate student Vicky



Topics we will cover:

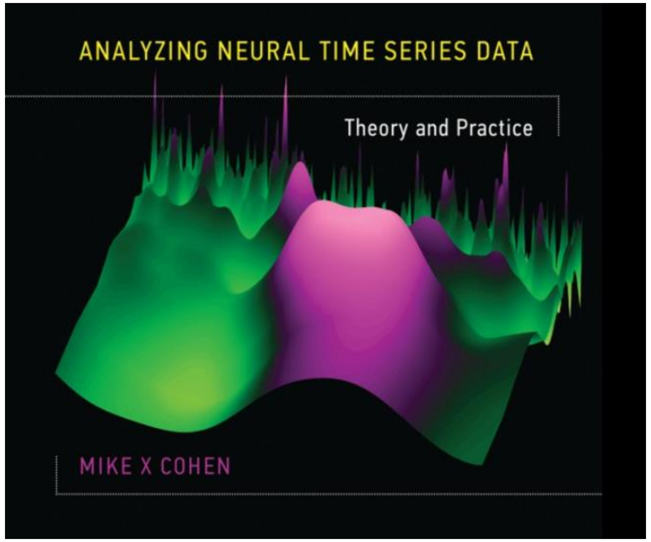
1. Where the heck do I start?
2. Overview of analysis pipeline
3. Time-varying signals as a collection of sinewaves
4. What is power?
5. Types of spectral analyses
6. Notes (baseline correction and stats)

Note: we will not have the time to cover what's under the hood. I will note important terms for you to look up in blue boxes.

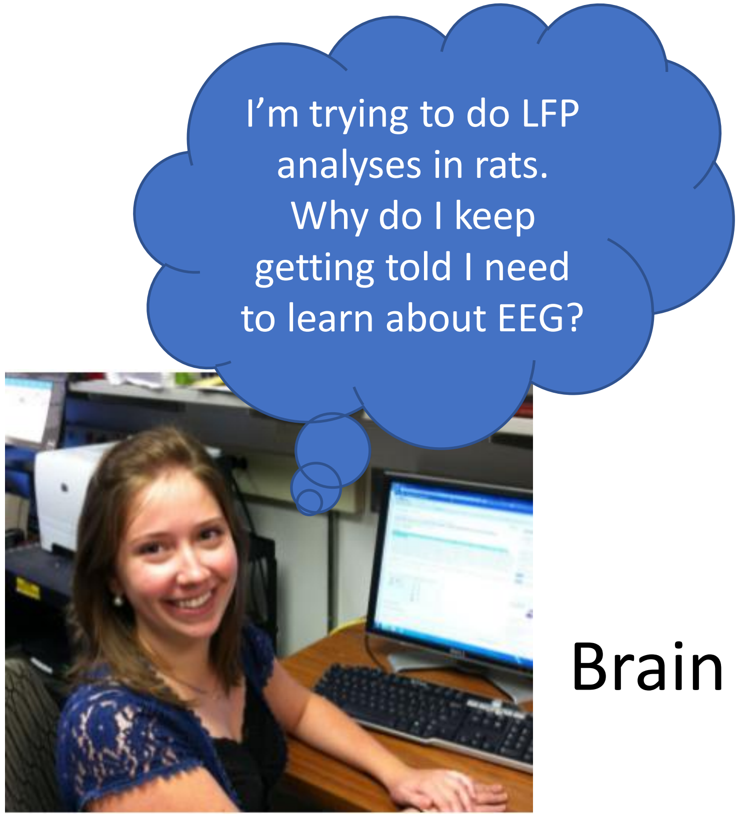
1. Where the heck do I start?

These analyses are hard!

1. Find a group:
   - Hacky hour,
   - PSY 7150:0000 Current topics in psychology (Dr. Wessel)
2. Find a mentor
3. Cohen's materials
   - Book
   - Youtube channel



A note on nomenclature



It’s all the same type of signal:

- EEG
- LFP
- Intra-cranial recordings

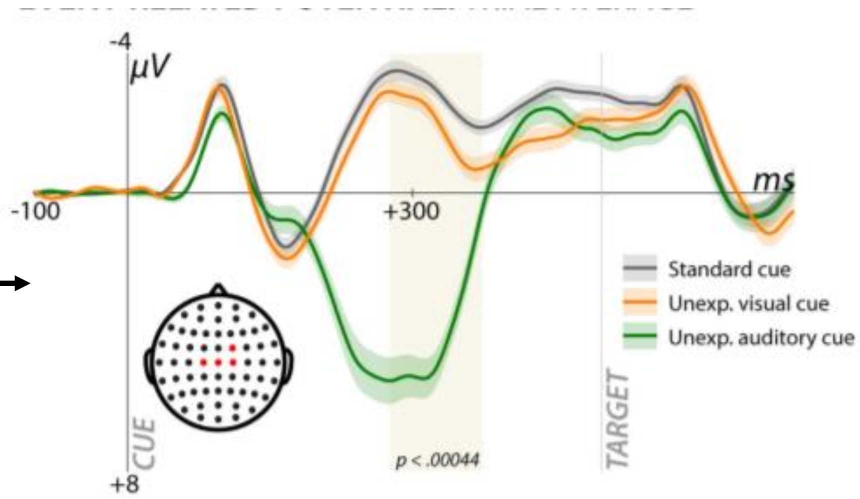
Brain signals that vary across time: it’s all time-series data

2. Analysis pipeline:

Raw data → Pre-processed data → ERP analyses / Spectral analyses / Microstate analyses

- Manual artifact rejection
- ICA
- Baseline correction
- Epoching

ERP analyses →



Huber 2019

2. Analysis pipeline:

Raw data → Pre-processed data

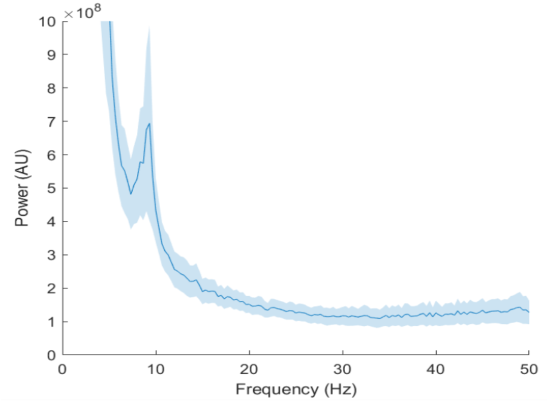
- Manual artifact rejection
- ICA
- Baseline correction
- Epoching

Pre-processed data → ERP analyses

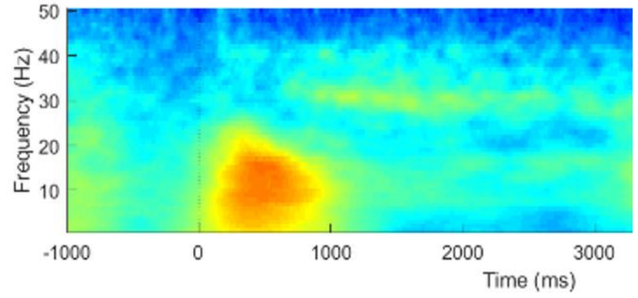
Pre-processed data → Spectral analyses

Pre-processed data → Microstate analyses

Spectral analyses → Non-time locked (spectral density)

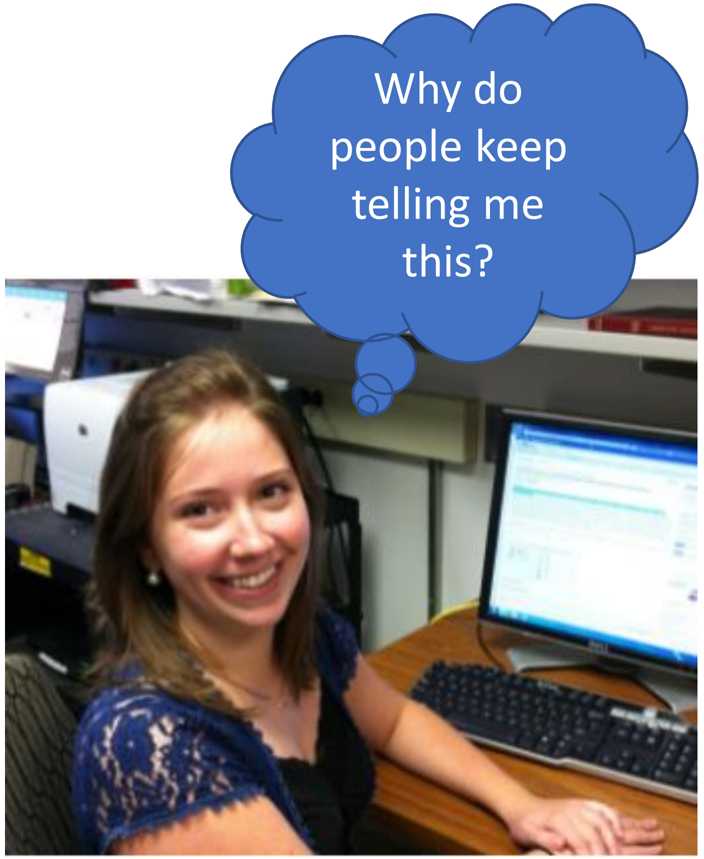


Spectral analyses → Time-locked (spectogram)



Muller Ewald Unpublished

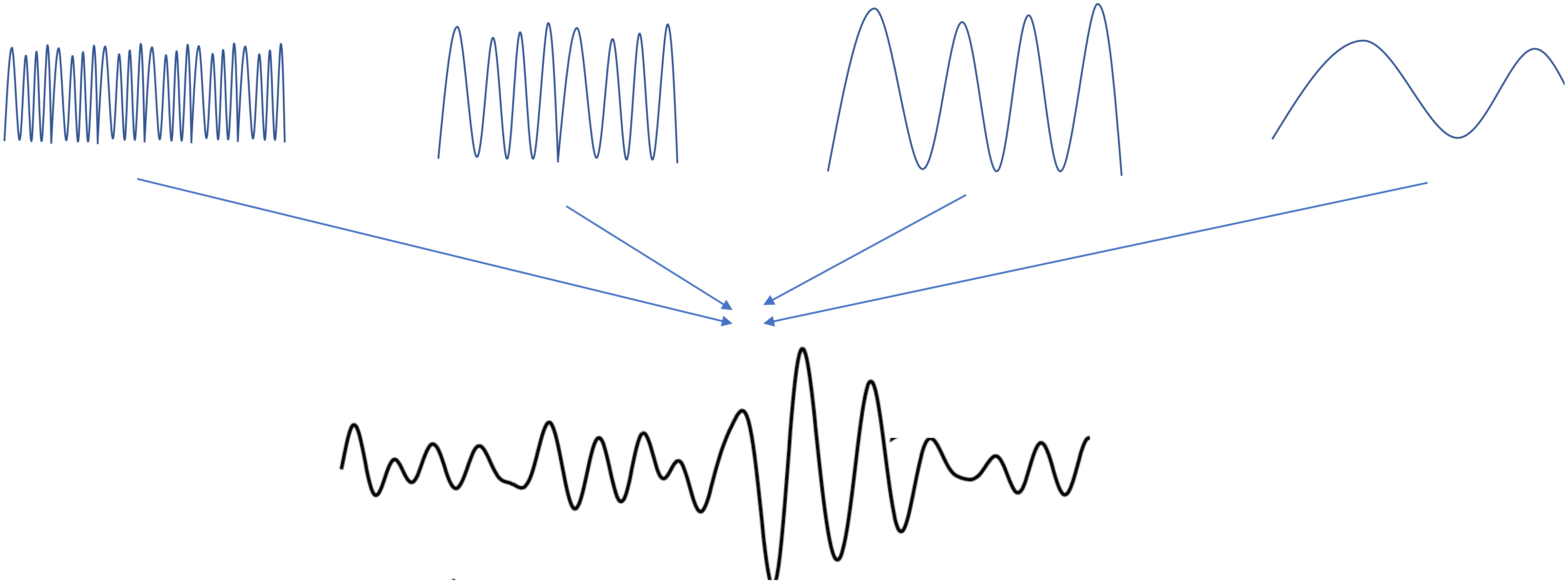
3. Time-varying signals as a collection of sinewaves



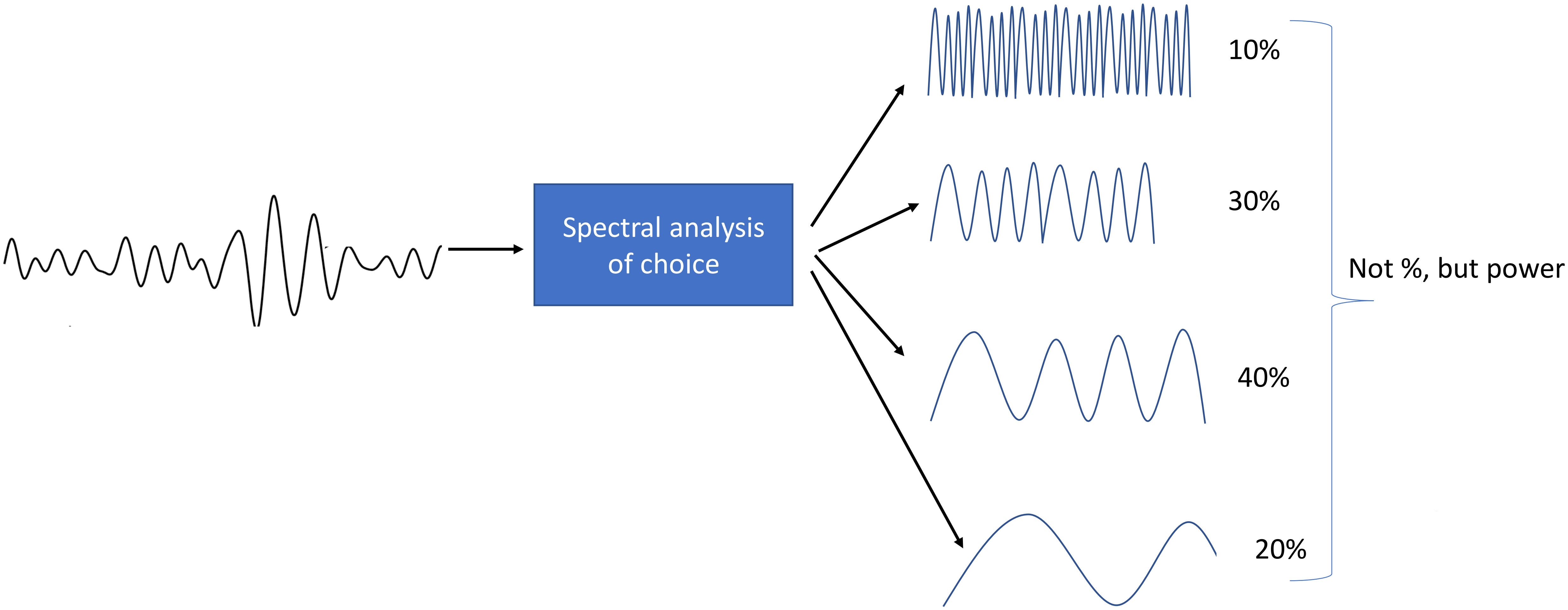
- Because it’s literally the most important thing.
- All spectral analyses are based on this principle.
- Spectral analysis: figuring out which sinewaves make up the signal and how much each sinewave contributes to the signal.

Any time-varying signal can be expressed as a collection of sinewaves

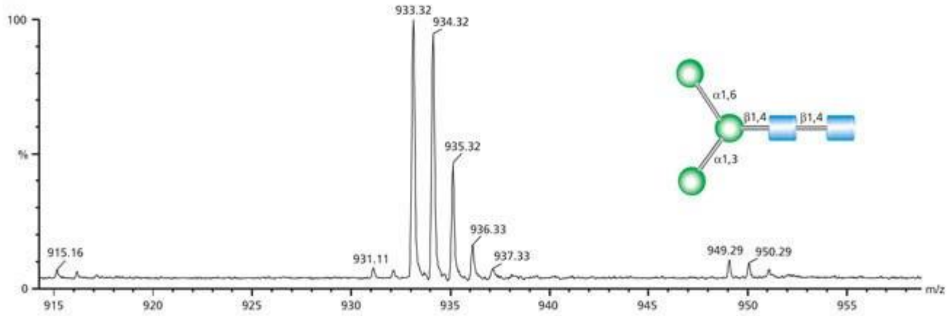
Pretend these are perfect sinewaves



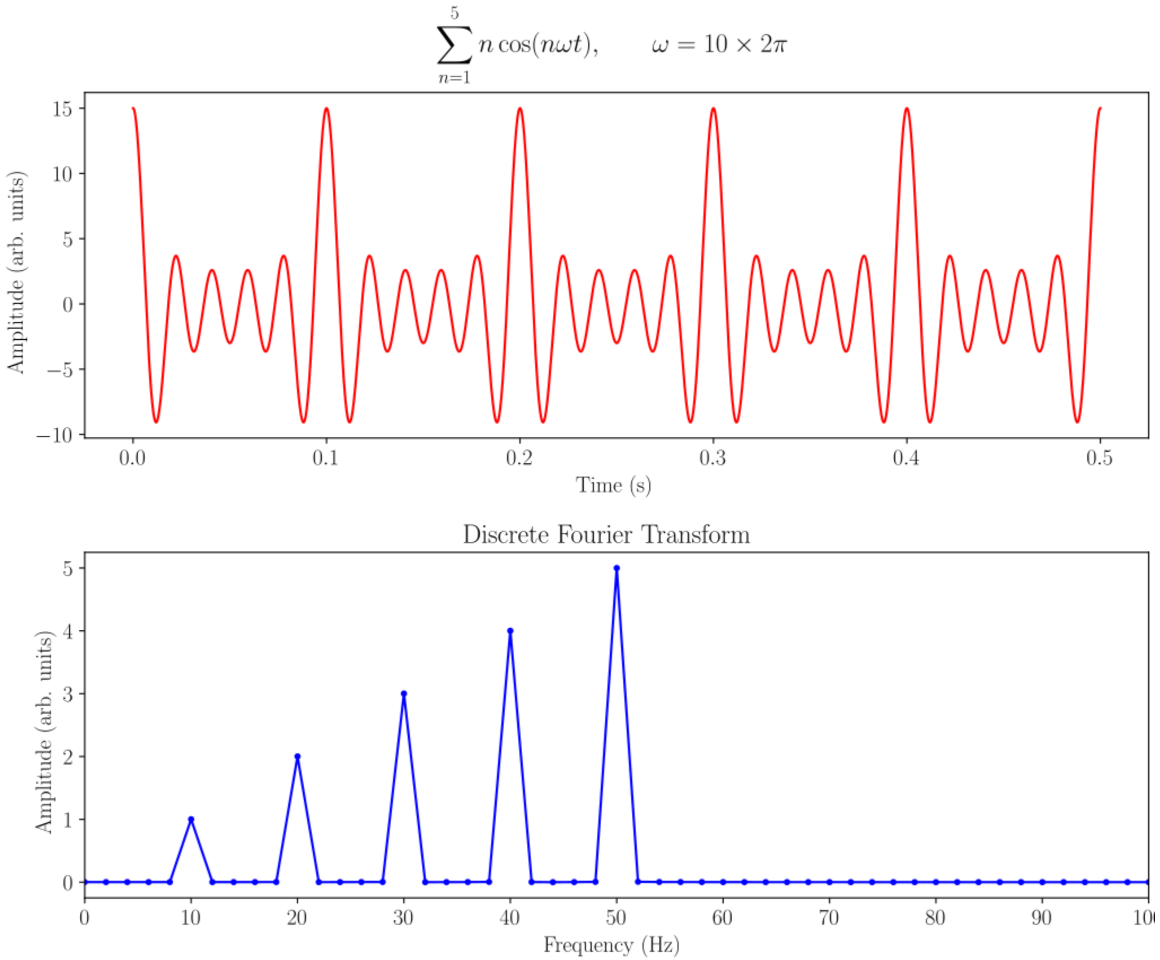
Any time-varying signal can be expressed as a collection of sinewaves



Any time-varying signal can be expressed as a collection of sinewaves



Think of spectral analyses as mass spec, but for time-series data.



FFT Wikipedia

Takeaway

Any time-varying signal can be expressed as a collection of sinewaves.

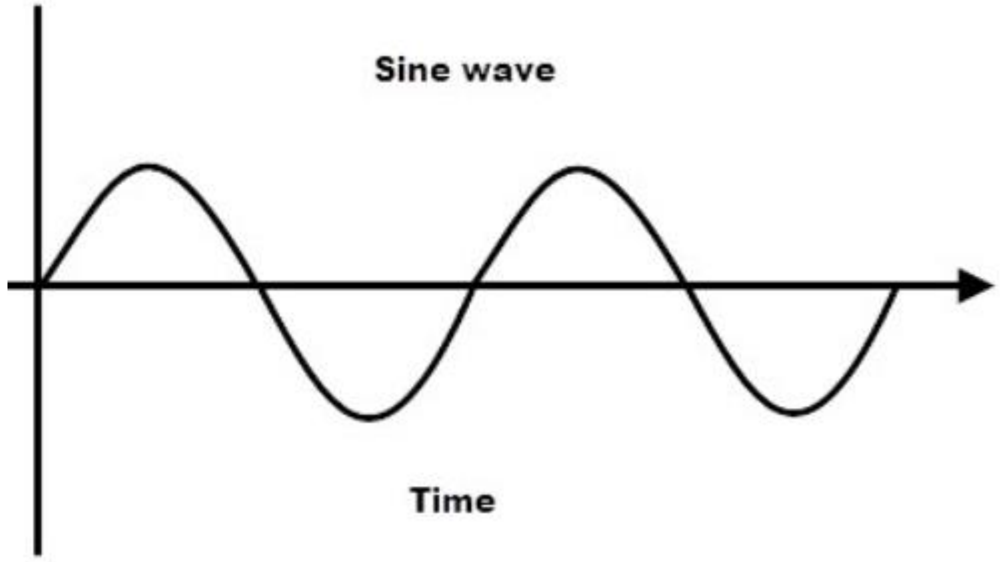
→

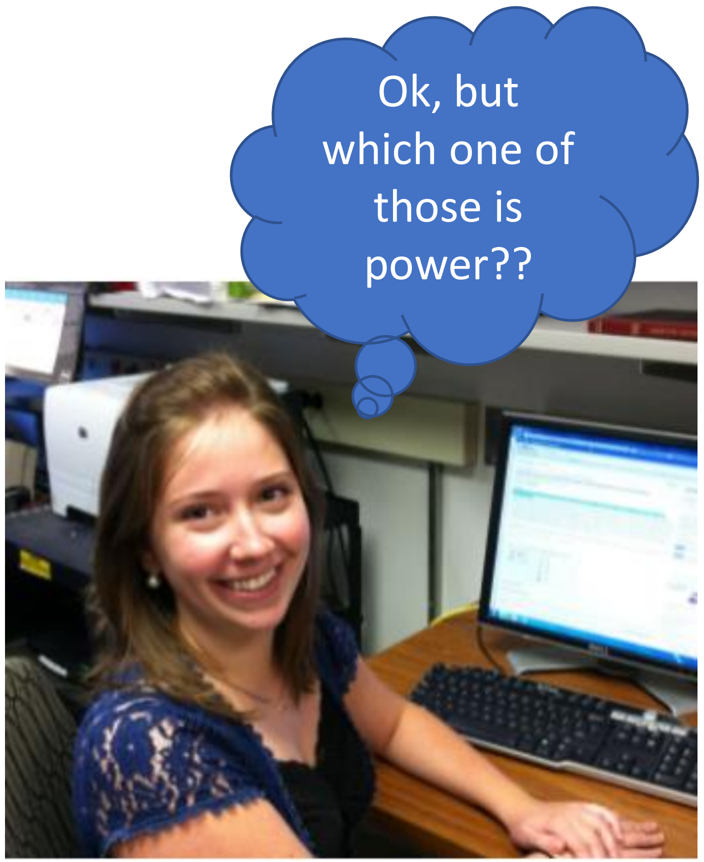
A spectral analysis breaks an EEG signal down into it's component sinewaves (and tells you how much of each is present).

→

This is relevant to neuroscientists because different oscillations (frequencies of sinewaves) are related to different neural processes.

4. What is power?





1. Amplitude – how high?
2. Frequency – how often/how fast?
3. Phase – when/at what degree?

What is power?

1. Amplitude and power are related – how strong?
2. Frequency – how often?
3. Phase – when?

Different explanations of what power is:

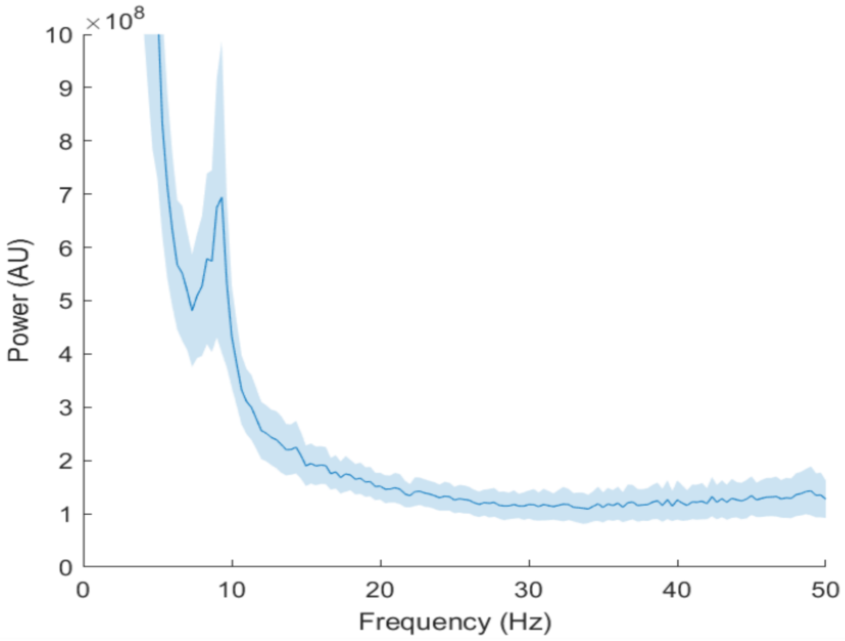
Level 1 – power is a property of a wave that tells you how prevalent that wave is in your signal.

Level 2 - power is the covariance between two signals.

Level 3 – power is the mapping of two vectors across time.

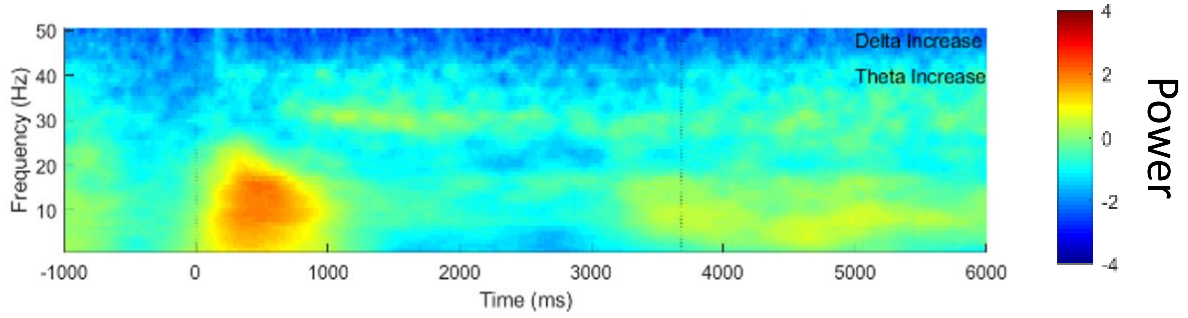
Level 1 – power is a property of a wave that tells you how prevalent that wave is in your signal

<u>Example 1: Spectral density plot</u>



Peak between 7 – 10 Hz represents high power in those frequencies.

<u>Example 2: Spectrogram</u>

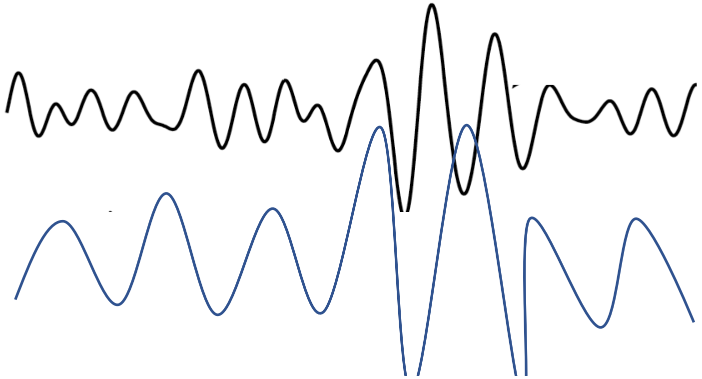


Orange shading from 0 to 1000 ms represents high power in frequencies 0 – 20 Hz.

Level 2 – power is the covariance between two signals

Signal 1 = EEG signal

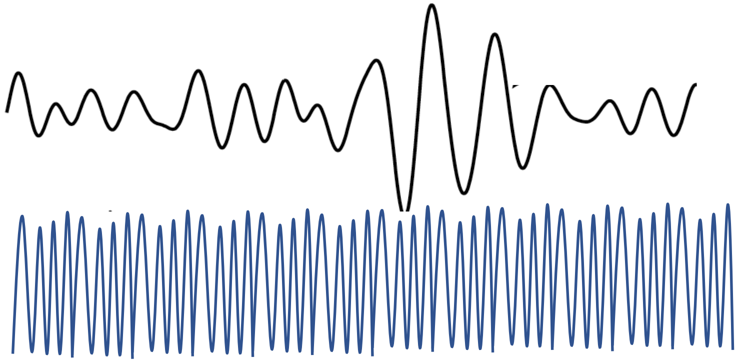
Signal 2 = a sinewave



High covariance = high power

Signal 1 = EEG signal

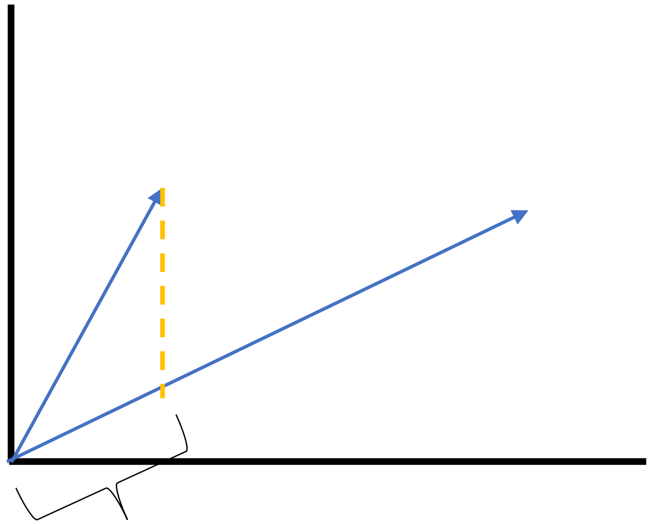
Signal 2 = a different sinewave



Low covariance = low power

How do I mathematically figure out the covariance between two vectors? This is where the dot product and convolution come in.

Level 3 – Mapping between two vectors across time



Dot product/projection

Vector 1 = your EEG data
Vector 2 = a wave/wavelet

$$a \cdot b = \Sigma_{a_i}\ b_i = a_1 b_1 + a_2 b_2 \ldots$$

Dot product equation across timepoints

Different explanations of what power is:

Level 1 – power is a property of a wave that tells you how prevalent that wave is in your signal.

Level 2 - power is the covariance between two signals.

Level 3 – power is the mapping of two vectors across time.

Level 4 – complex numbers!

5. Types of spectral analyses

A. Non time-locked

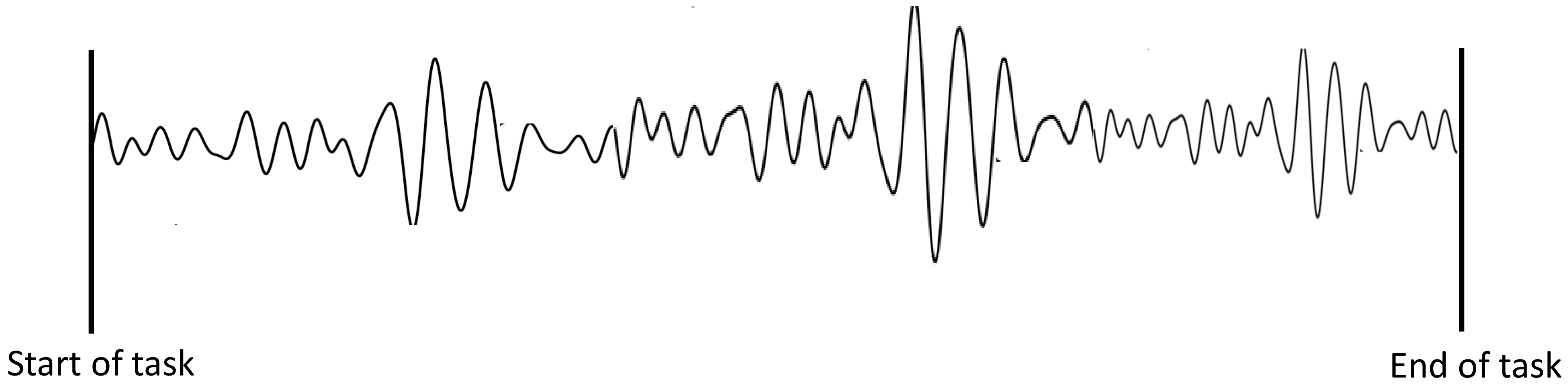
- Fast Fourier transform

B. Time locked

- Short term fast Fourier transform
- Hilbert transform
- Morlet wavelets

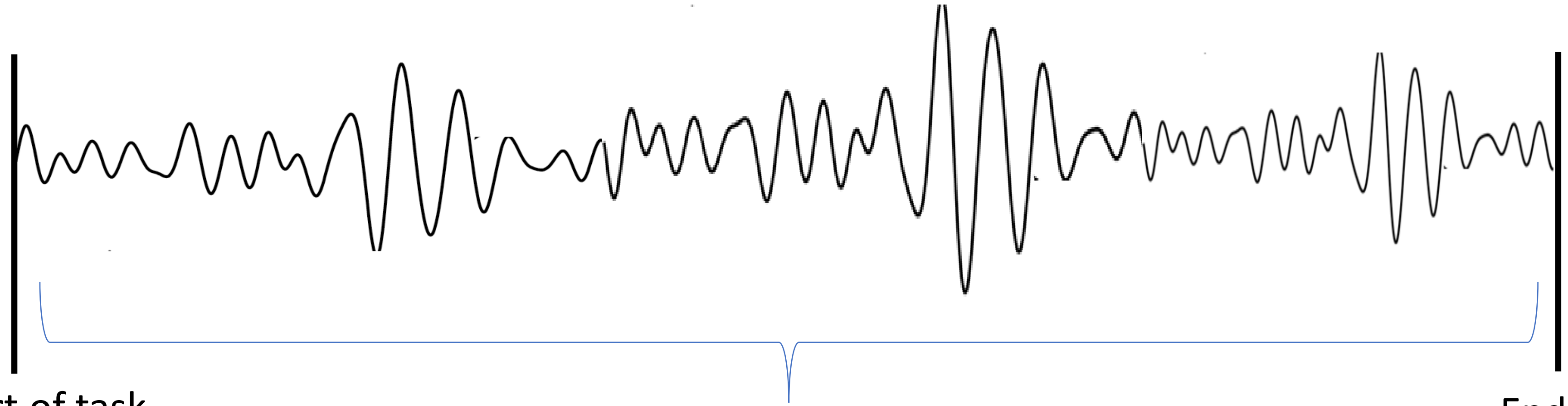
Non-time locked spectral analyses

- Fast Fourier transform



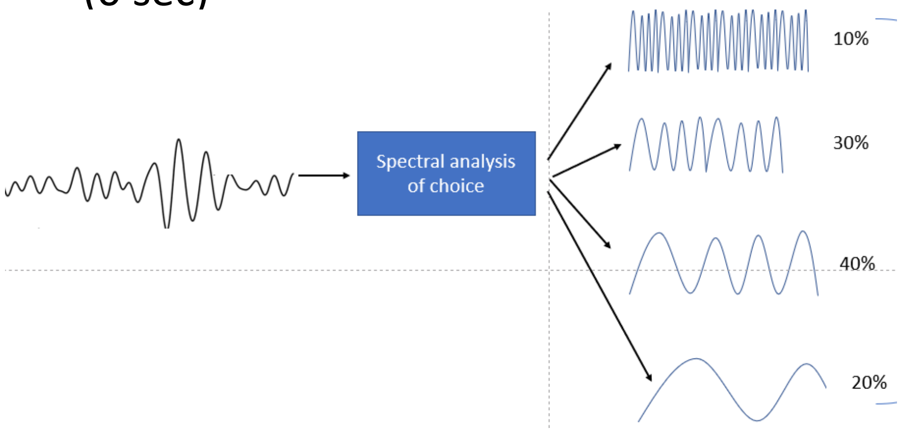
Non-time locked spectral analyses

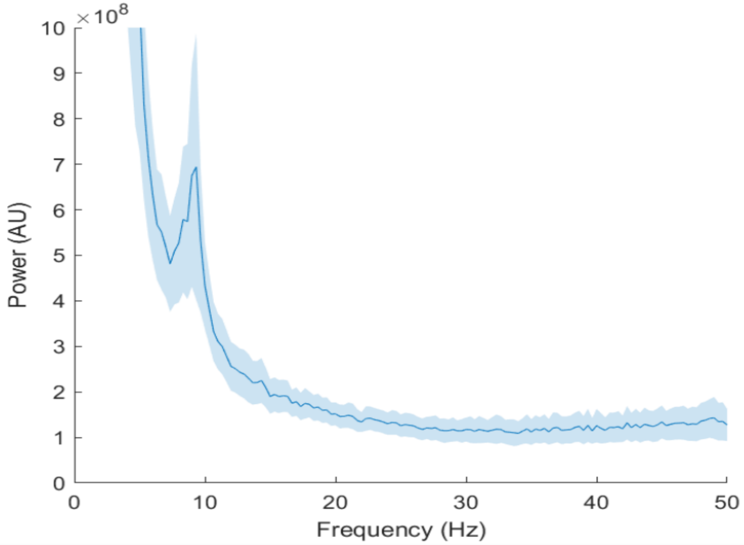
- Fast-fourier transform



Start of task
(0 sec)

End of task
(120 sec)



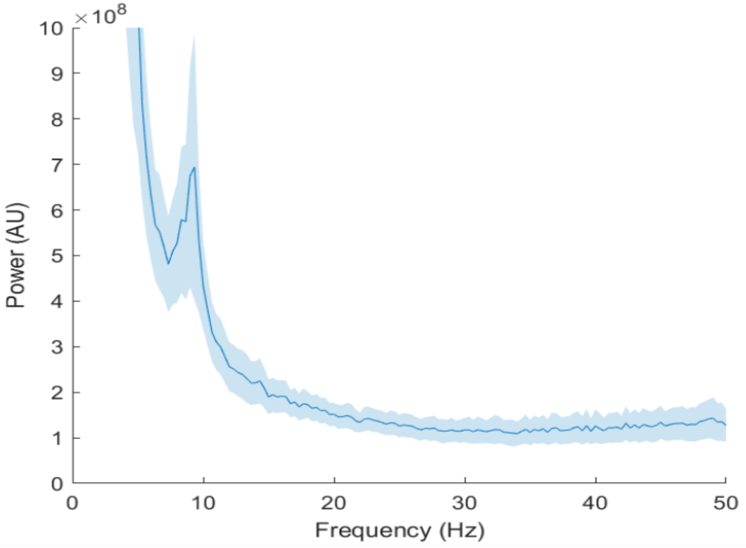


Fast Fourier transform: what the code looks like (in theory)

```
%% Load in the data for one subject
load(subjData);
%% Do the fft
for frequencyRUN = 0:maxFreq
        subjData % grab subject data
        sinewave %make a sinewave at this frequency that is the same size
                as the subject's data
        fftCoefficient = dot (subjData, sinewave) % get the dot product between
                                        the subject's data and your sinewave
        store fftCoefficient
end
```

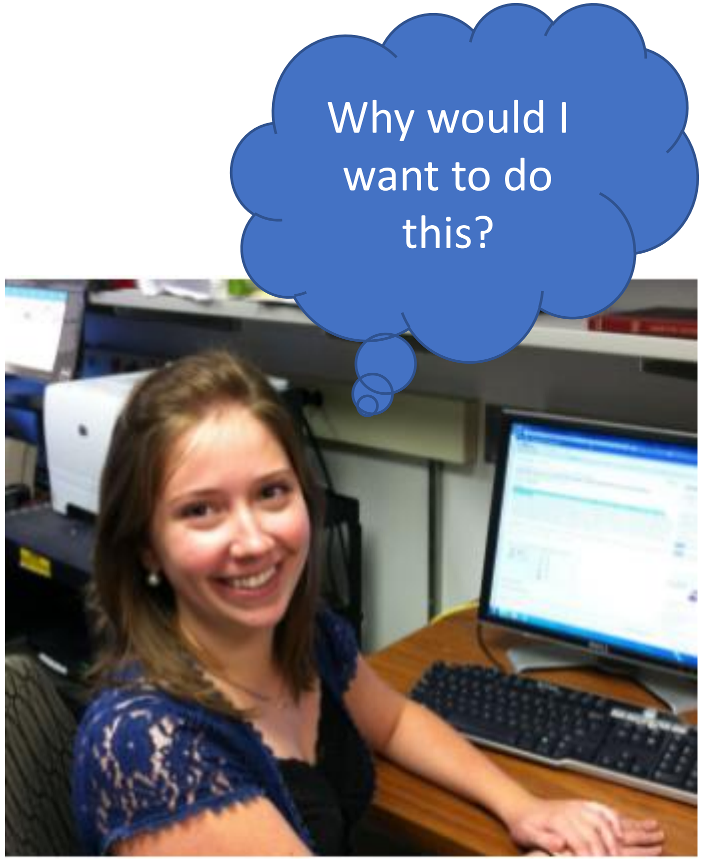
Fast Fourier transform: what the code looks like (in theory)

```
%% Load in the data for one subject
load(subjData);
%% Do the fft
for frequencyRUN = 0:maxFreq
        subjData % grab subject data
        sinewave %make a sinewave at this freque
                as the subject's data
        fftCoefficient = dot (subjData, sinewave) %                            en
                                the subject's d
        store fftCoefficient
end
```



Non-time locked spectral analyses

- Fast Fourier transform



Biggest downside: it's not time-locked

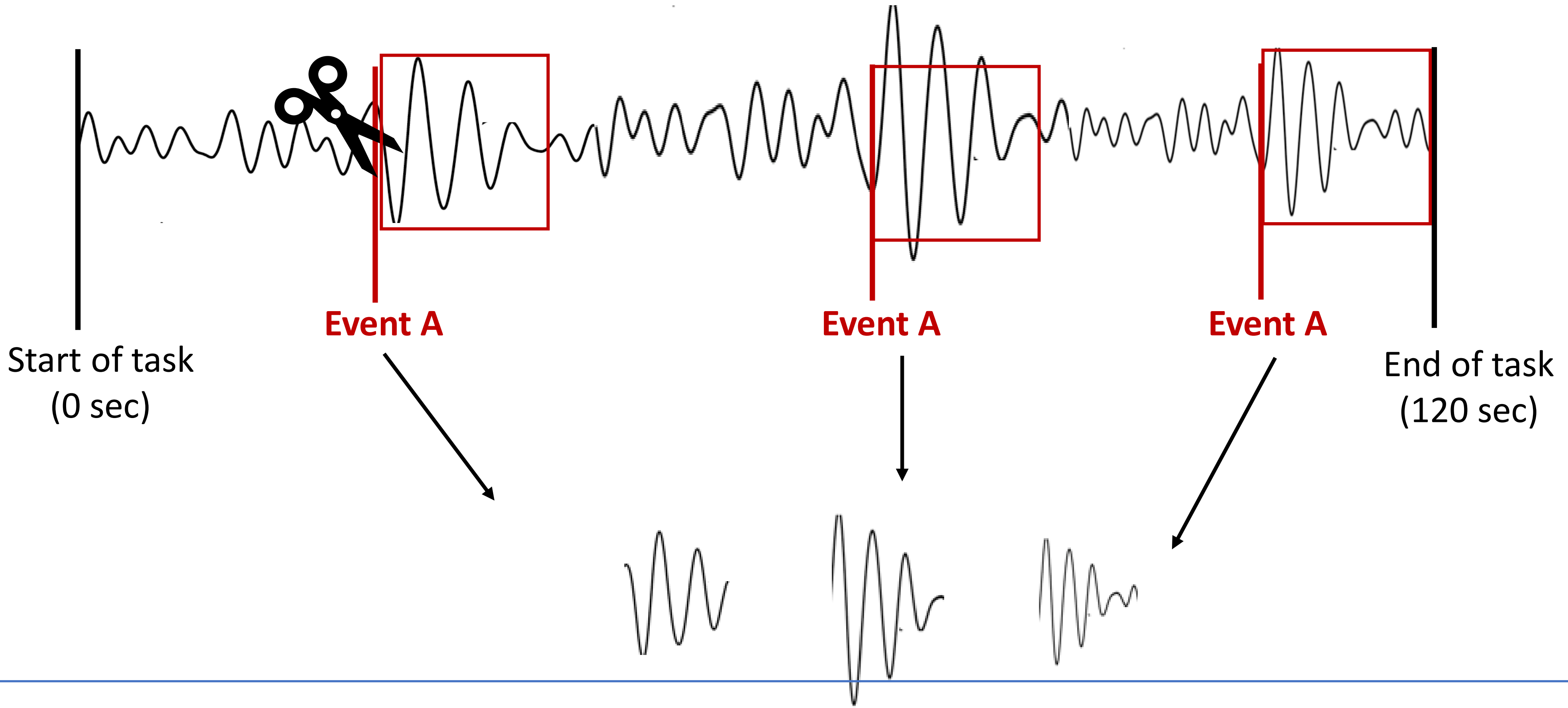
1. It's a great place to start
- Easier to code. Doesn't involve as many event time stamps

2. Great for looking at gross-level differences between conditions

3. Great if you don't have time-stamps/events
   - e.g. resting-state analysis

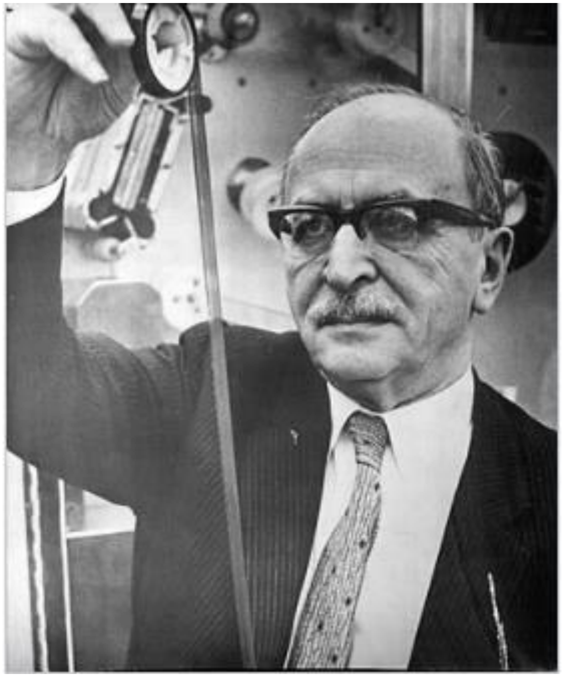
Note: you can get more precise with your data when using the fft

What if I want to use the FFT to compare general power levels in my data in the baseline vs the post-event period?



Time-locked spectral analyses

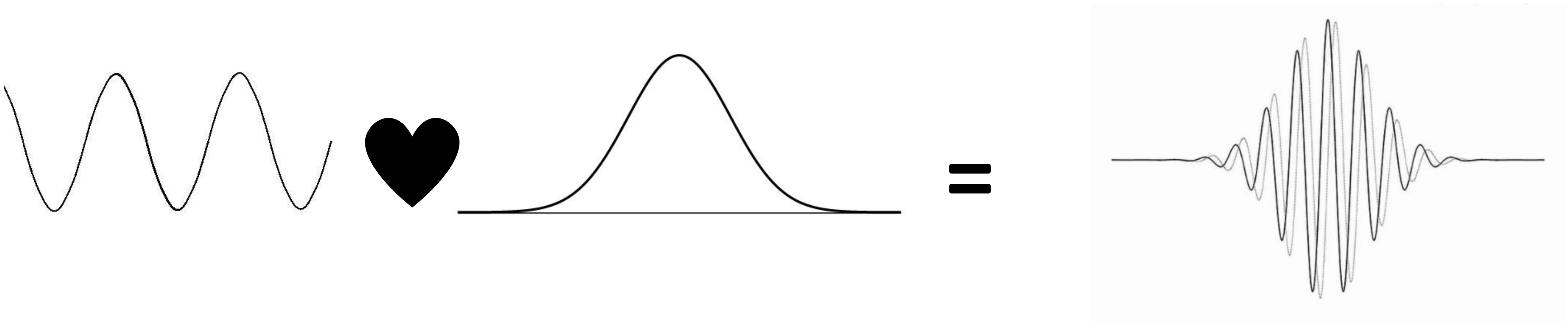
1. Short-term fast Fourier transform
2. Hilbert transform
3. Morlet Wavelets



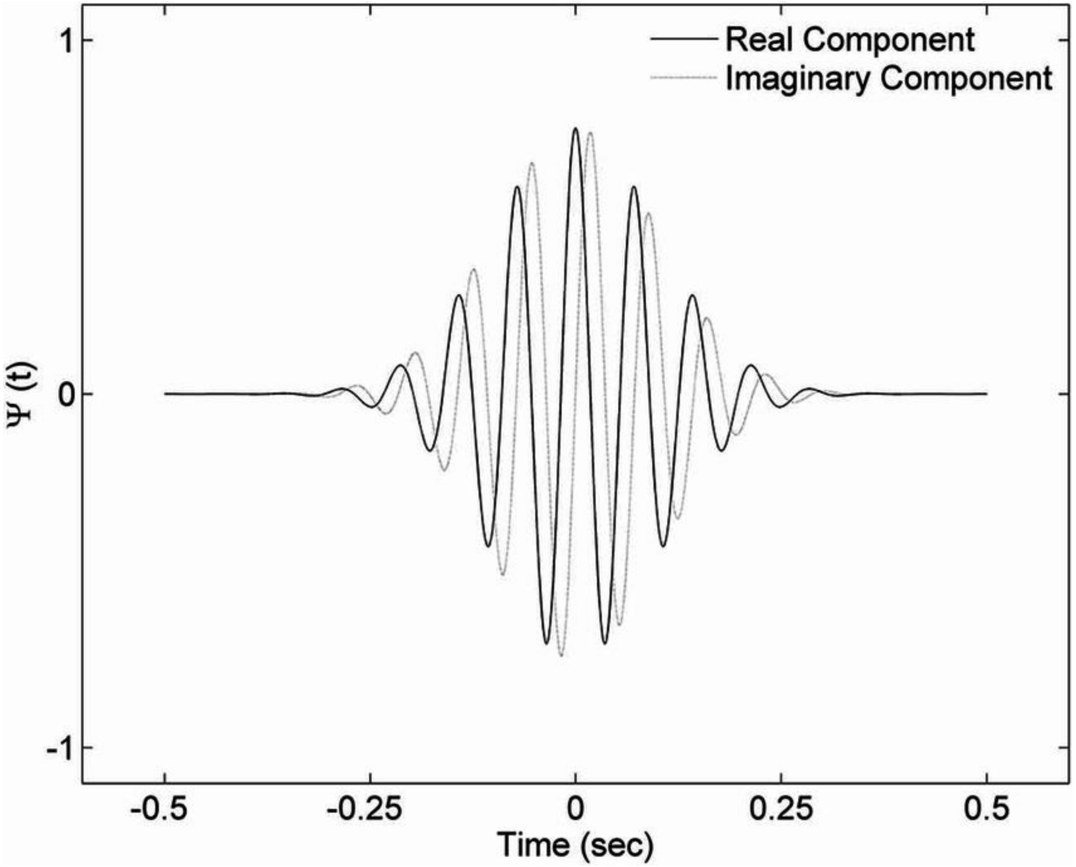
Gabor: professor of physics

Morlet wavelet method

- A Morlet wavelet is a sinewave mixed with a gaussian curve



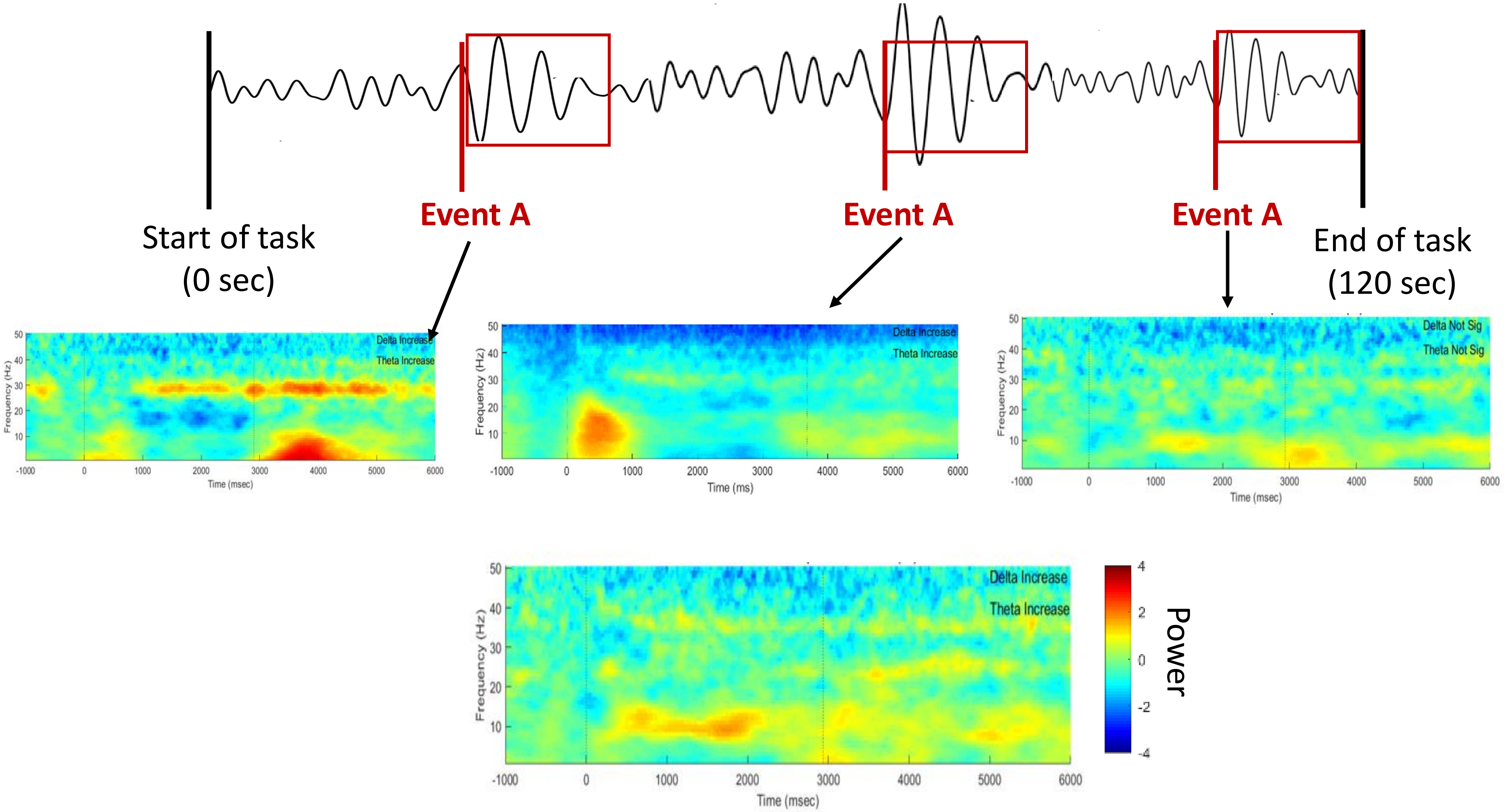
Morlet wavelet method



- Because your wavelet varies in time, you can now look at how your signal varies in time
- You can now time/event-lock

Morlet wavelets have a real and an imaginary component. This is important.

Morlet wavelet method



Morlet wavelet method: what the code looks like (in theory)

```
for subjectRUN = 1:maxSubjects % Load in the data for one subject
        data = load(subjData);







end
```

Morlet wavelet method: what the code looks like (in theory)

```
for subjectRUN = 1:maxSubjects % Load in the data for one subject
        data = load(subjData);
        %% Convolve
        for trialRUN = 1:maxTrials % Grab data from a single trial
                dataTrial = data(one trial); % grab data from one trial







        end
dataAverageTrials = nanmean(dataTrialX ,3); % Average across trials
end
```

Morlet wavelet method: what the code looks like (in theory)

```
for subjectRUN = 1:maxSubjects % Load in the data for one subject
        data = load(subjData);
        %% Convolve
        for trialRUN = 1:maxTrials % Grab data from a single trial
                dataTrial = data(one trial); % grab data from one trial
                for waveletRUN = 1: maxWavelets % Convolve your signal with wavelets at your
                                                  frequencies of interest


                end
                dataTrialX =  cat(3, dataTrialX, dataTrialY);
        end
dataAverageTrials = nanmean(dataTrialX ,3); % Average across trials
end
```

Morlet wavelet method: what the code looks like (in theory)

```
for subjectRUN = 1:maxSubjects % Load in the data for one subject
        data = load(subjData);
        %% Convolve
        for trialRUN = 1:maxTrials % Grab data from a single trial
                dataTrial = data(one trial); % grab data from one trial
                for waveletRUN = 1: maxWavelets % Convolve your signal with wavelets at your
                                                        frequencies of interest
                        dataTrial_singleWavelet = convolve(dataTrial,wavelet) ;
                        dataTrialY = vertcat(dataTrialY, dataTrial_singleWavelet );
                end
                dataTrialX =  cat(3, dataTrialX, dataTrialY);
        end
dataAverageTrials = nanmean(dataTrialX ,3); % Average across trials
end
```

Morlet wavelet method: what the code looks like (in theory)

```
for subjectRUN = 1:maxSubjects % Load in the data for one subject
        data = load(subjData);
        %% Convolve
        for trialRUN = 1:maxTrials % Grab data from a single trial
                dataTrial = data(one trial); % grab data from one trial
                for waveletRUN = 1: maxWavelets % Convolve your signal with wavelets at your
                                                  frequencies of interest
                        dataTrial_singleWavelet = convolve(dataTrial,wavelet)
                        dataTrialY = vertcat(dataTrialY, dataTrial_singleWavelet )
                end
                dataTrialX =  cat(3, dataTrialX, dataTrialY)
        end
dataAverageTrials = nanmean(dataTrialX ,3); % Average across trials
end
```

Note: this is not the actual computation we do because this would be too slow. Look up the convolution theorem.

Morlet wavelet method: what the code looks like (in theory)

1. Load in a subject
2. Convolve data from each trial
3. Average data across trials
4. Save averaged data in a matrix

Subject-level analyses:
Covered in previous slide

5. Average data across all subjects
6. Run stats
7. Make figure

5. Notes

a. Baseline correction
b. Stats

A. Baseline correction

1. No baseline correction
2. Baseline correction

Please consider baseline-correcting.

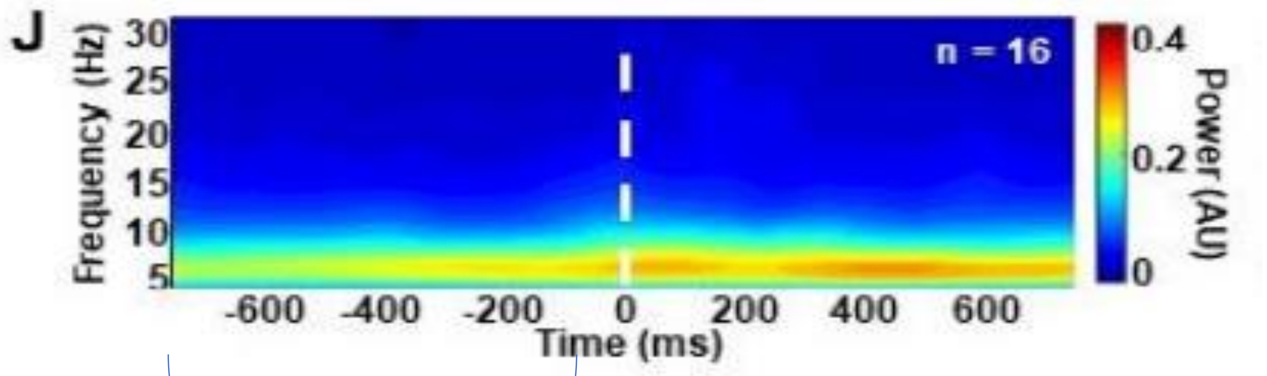
You can miss interesting aspects of the data (especially at higher frequencies) when you don’t baseline correct

A. Baseline correction

1. No baseline correction
2. Baseline correction

Please consider baseline-correcting.

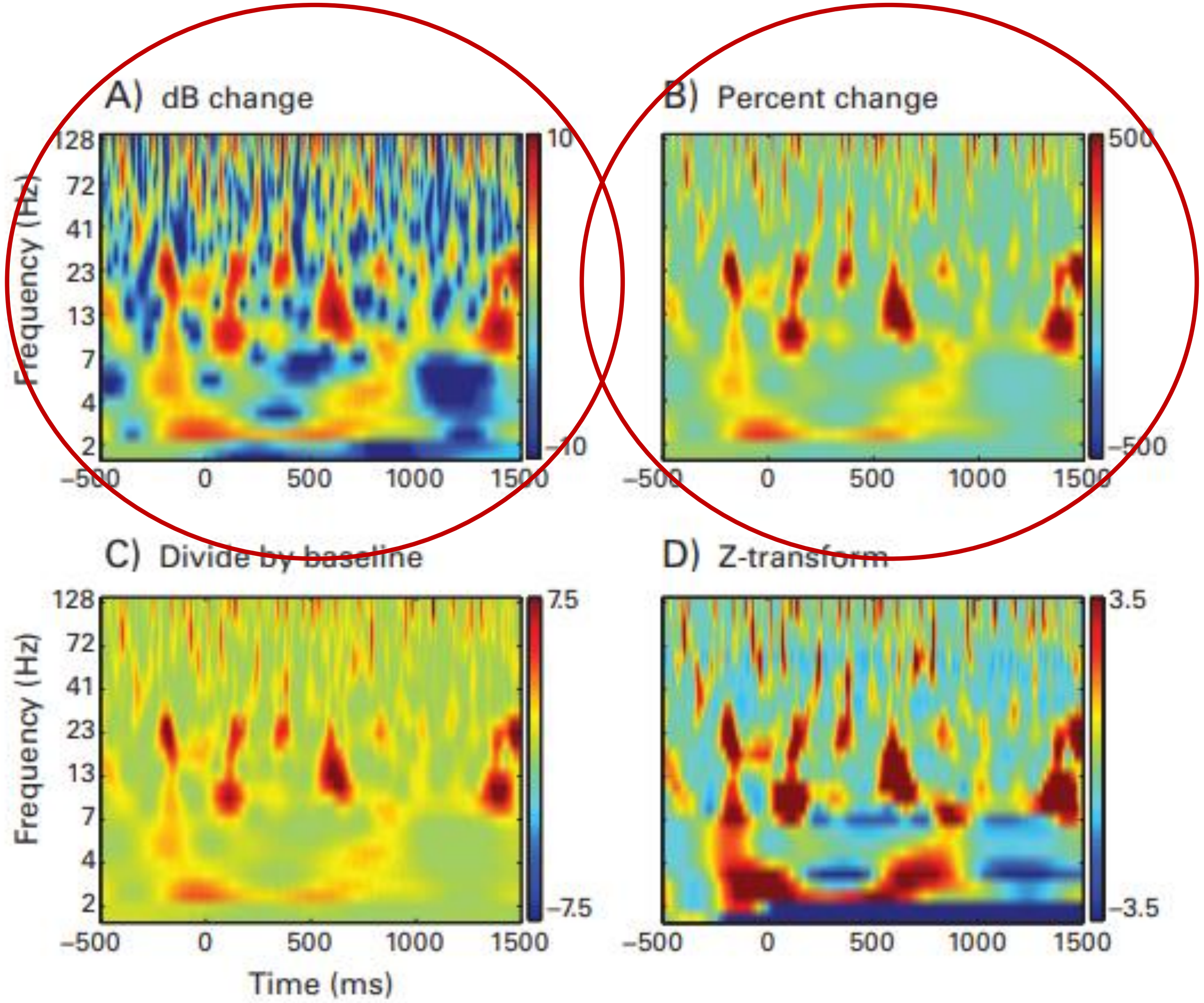
You can miss interesting aspects of the data (especially at higher frequencies) when you don't baseline correct.



Not baseline corrected
How do we know?

Baseline correction

2. Baseline correction
   - dB change
   - % change
   - Divide by baseline
   - Z-transform



Cohen 2014

B. How do stats work?

1. Average across ROI and compare

    - Only do if you have a very strong a-priori hypothesis

2. Sample-by-sample t-test

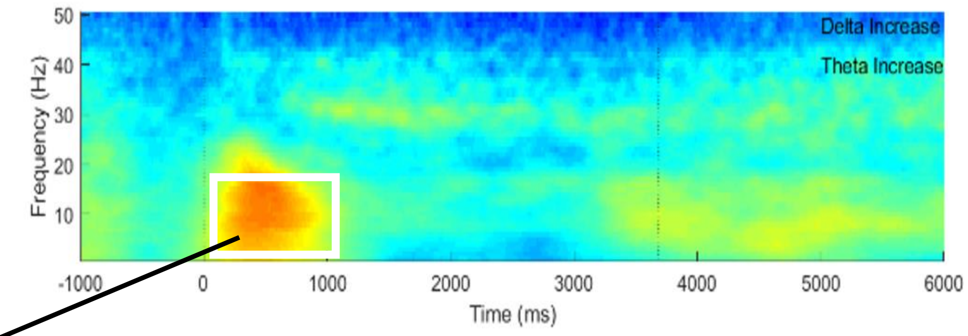
    - Preferred -> honors the resolution of the data

    - Don’t forget to correct for multiple comparisons (Bonferroni, Bonferroni-Holm, FDR, cluster-based)

How do stats work?

1. Average across ROI and compare

Subject 1



Condition 1

Subject	Averaged value
1	2.5

How do stats work?

1. Average across ROI and compare

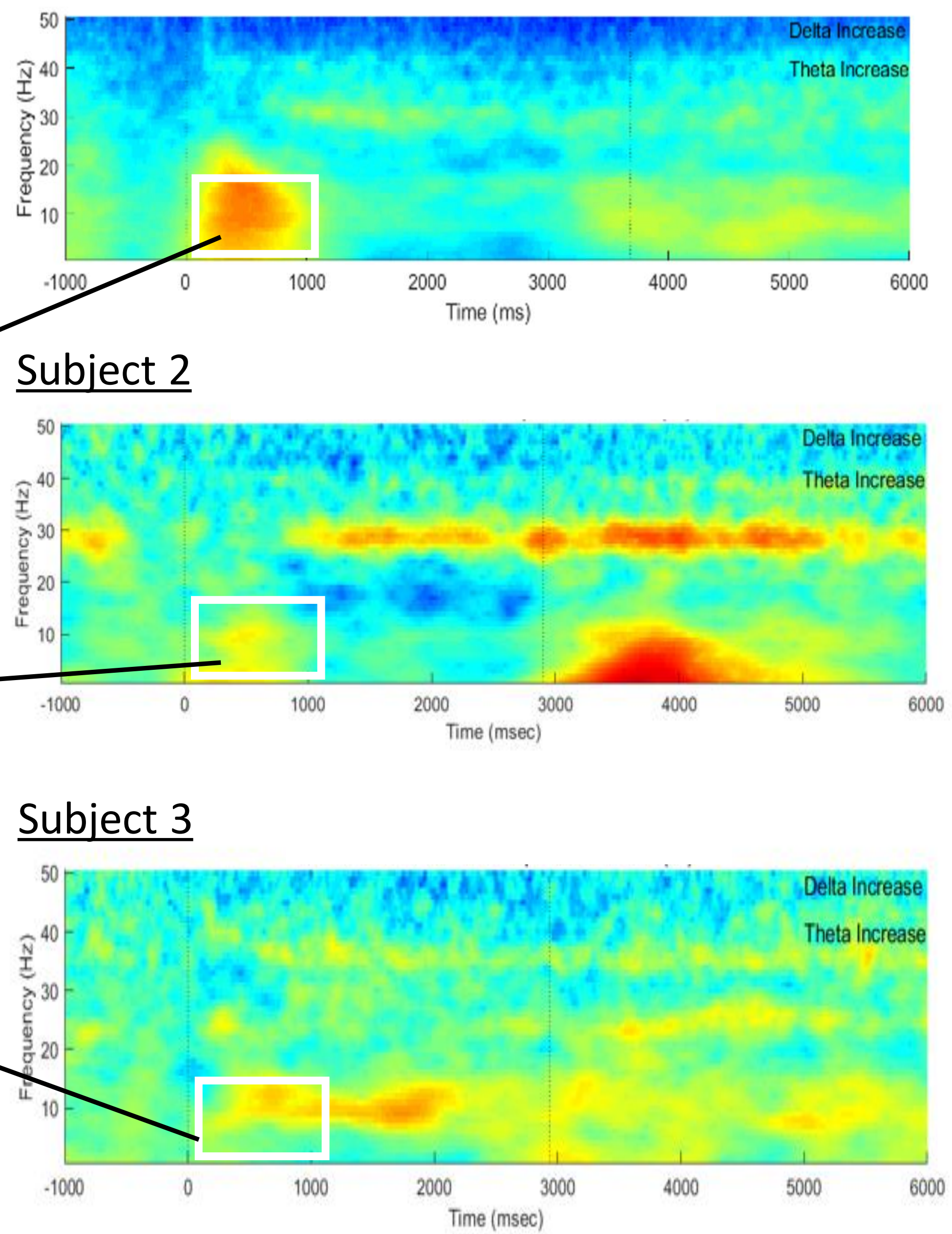
Condition 1

Subject	Averaged value
1	2.5
2	1
3	1.5

Subject 1

Subject 2

Subject 3



How do stats work?

1. Average across ROI and compare

Condition 1

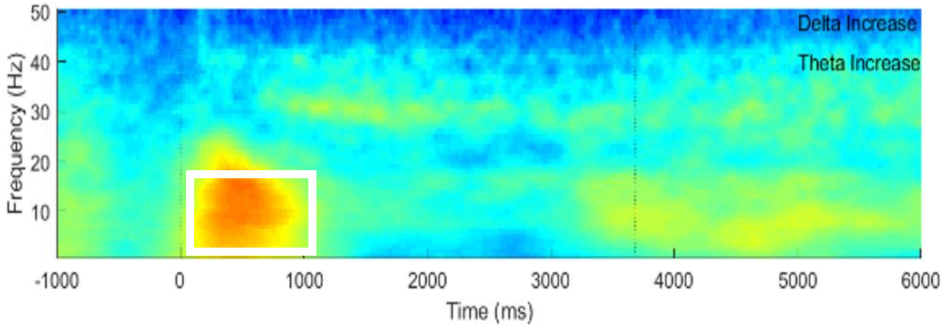
Subject	Averaged value
1	2.5
2	1
3	1.5

Condition 2

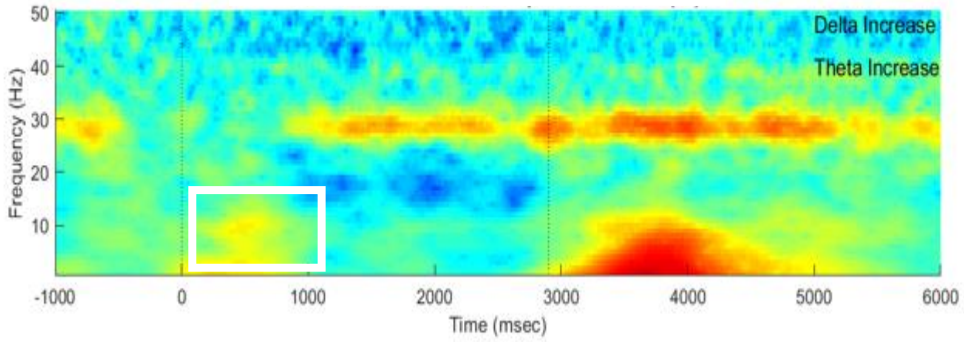
Subject	Averaged value
1	5
2	7
3	2.5

(Non-parametric) t-test

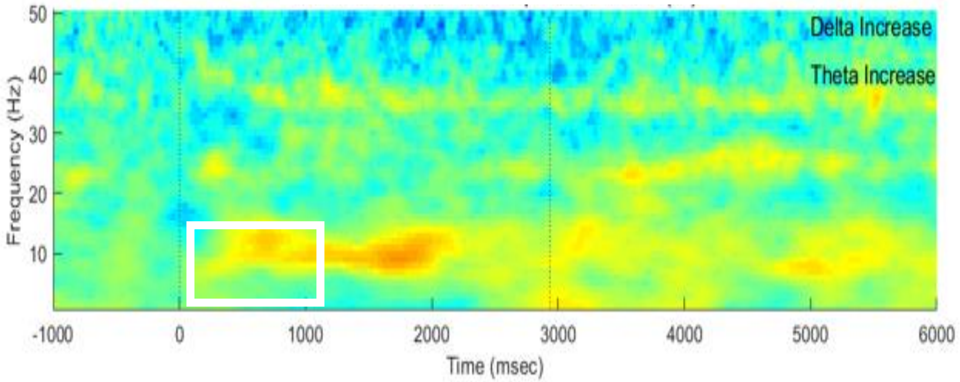
Subject 1



Subject 2



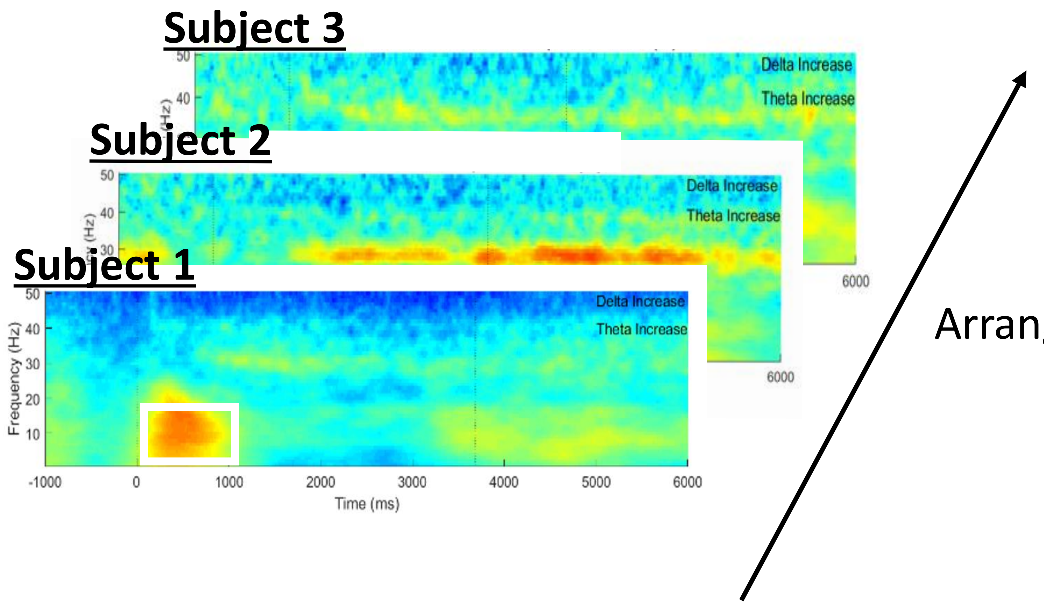
Subject 3



How do stats work?

2. Sample-by-sample t-test

Condition 1

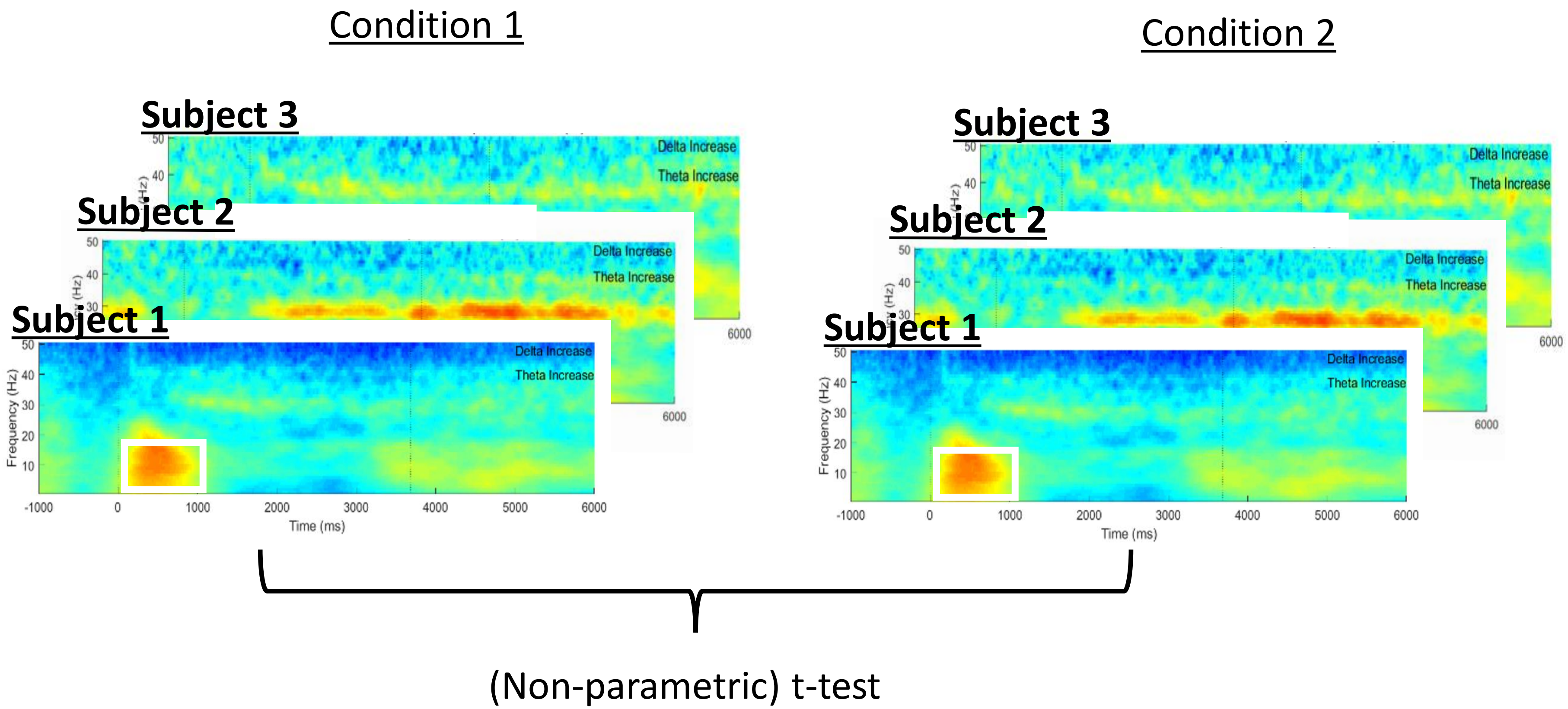


Arrange subjects in the 3rd dimension

How do stats work?

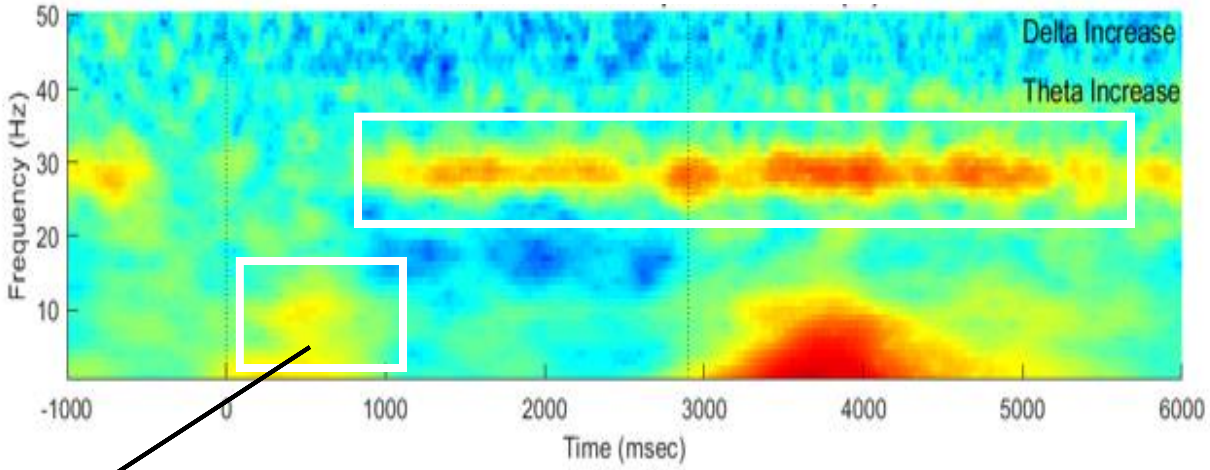
2. Sample-by-sample t-test

```
[~,p] = ttest(condition1, condition2, 'Dim', 3);
```



How do stats work?

- Beware of getting off-track based on what a spectrogram looks like.
- Stick to the stats.



ROI

1. Where the heck do I start?

These analyses are hard!

Anyone can borrow some code and run an analysis. Being able to understand what you are doing and troubleshoot problems is where the real work begins.

1. Find a group:
   - Hacky hour,
   - PSY 7150:0000 Current topics in psychology (Dr. Wessel)
2. Find a mentor
3. Cohen's materials
   - Book
   - Youtube channel

I'm still working on this part!