TIMESERIES 3

10.26.2020

RECAP + DEMO

* oscillations/periodic signals
  * often sinusoidal!

  * even when not sinusoidal, can be decomposed into a sum of sinusoids

    * this is the **fourier transform**

RECAP + DEMO

* to see which frequencies are present in a timeseries, simple fourier transforms are not the best tool

* instead, we use **power spectral density (psd)** estimators

  * this is like a *regularized* fourier transform

RECAP + DEMO

* if we compute psd for small snippets of time and then stack them together into an array

* this is the **spectrogram**

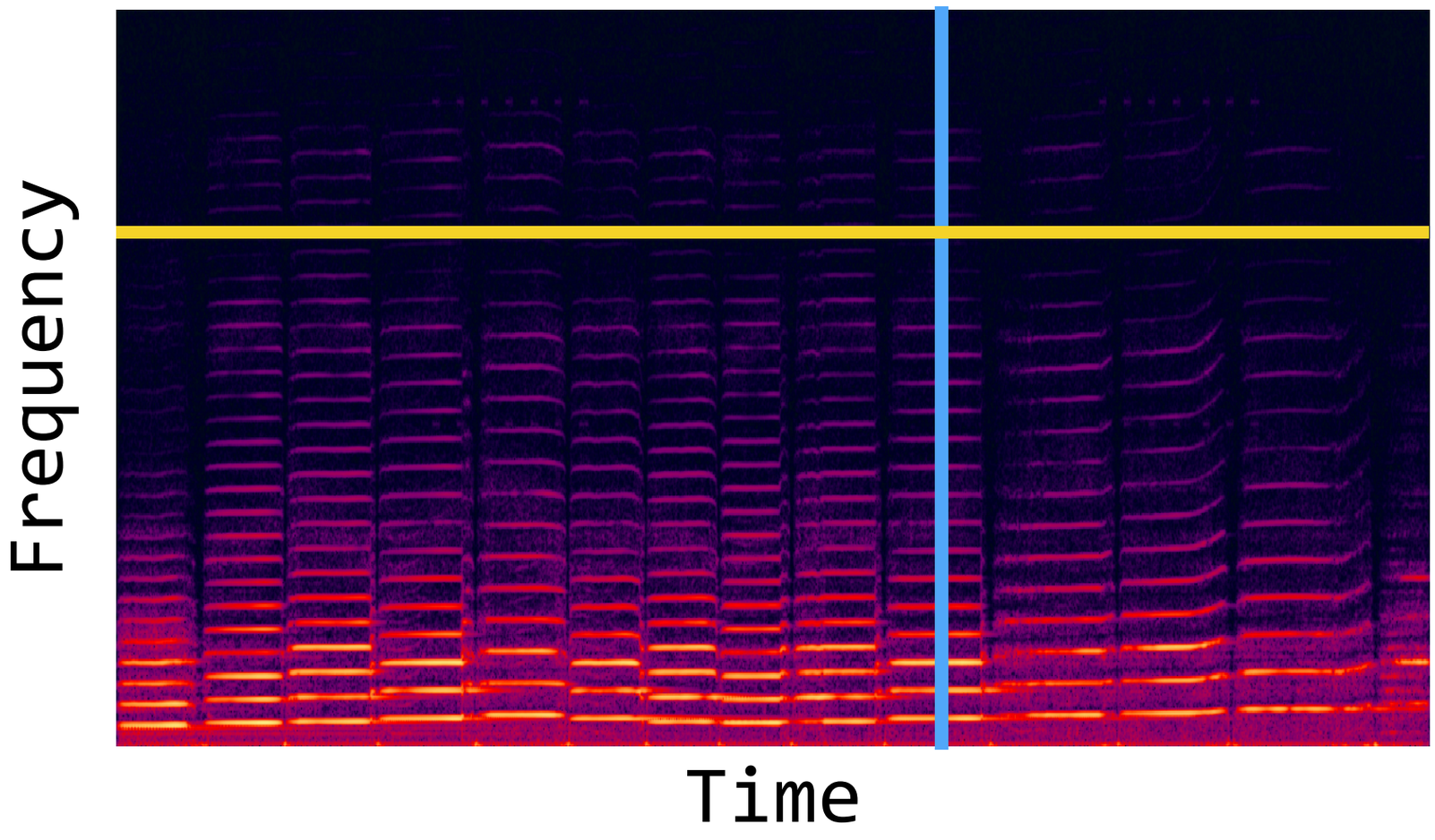
* it shows which frequencies are present in a timeseries at each point in time

* *you should know how to read a spectrogram*

THE SPECTROGRAM

* each **column** is the fourier transform of a short snippet

* what about each **row**? what does one row mean?



FILTERING

* **filtering** is a process that removes some frequencies from a timeseries and lets others remain (or even amplifies them)

* this is accomplished by convolving your timeseries with a **filter**, a small array that is designed to have a specific effect

FILTERING

* **low-pass filter:** removes high frequencies, allows low frequencies through

* **high-pass filter:** removes low frequencies, allows high frequencies through

* **band-pass filter:** removes all frequencies except for a specific band (the “pass band”)

FILTERING

* **low-pass filter:** removes high frequencies, allows low frequencies through

* **high-pass filter:** removes low frequencies, allows high frequencies through

* **band-pass filter:** removes all frequencies except for a specific band (the “pass band”)

FILTERING

* back to the spectrogram:
* one row of a spectrogram is a lot like a **band-pass filtered** version of a timeseries

FILTERING

* suppose we have some EEG data from a human subject and we want to filter it so that only alpha-band oscillations remain

  * (this is a band-pass filter)

* how do you make a filter that has the properties you want?

FILTERING

* **scipy.signal** is a module in scipy that contains lots of useful functions for filter design

* **scipy.signal.firwin** creates “finite impulse response” filters with desired properties

END