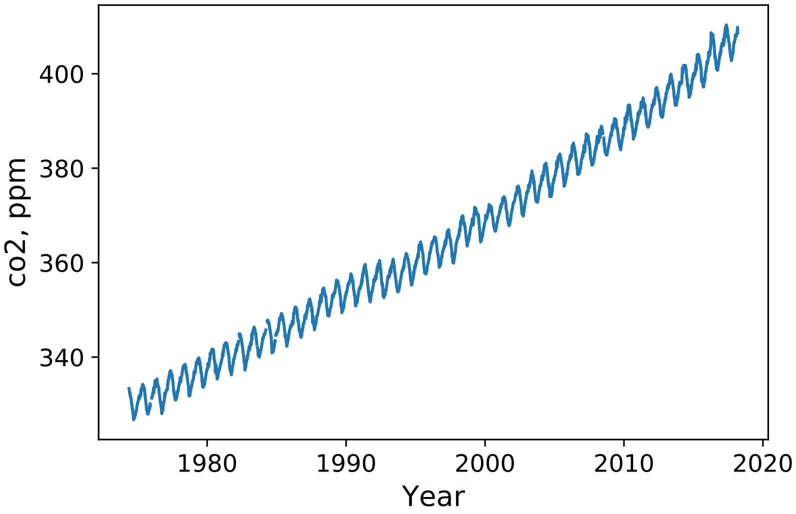
Lecture 20: Time-Frequency Decompositions

David Carlson

Overview

- How to remove known periodicities from the data
- What to do if we don't know the periodicities
- Building frequency features



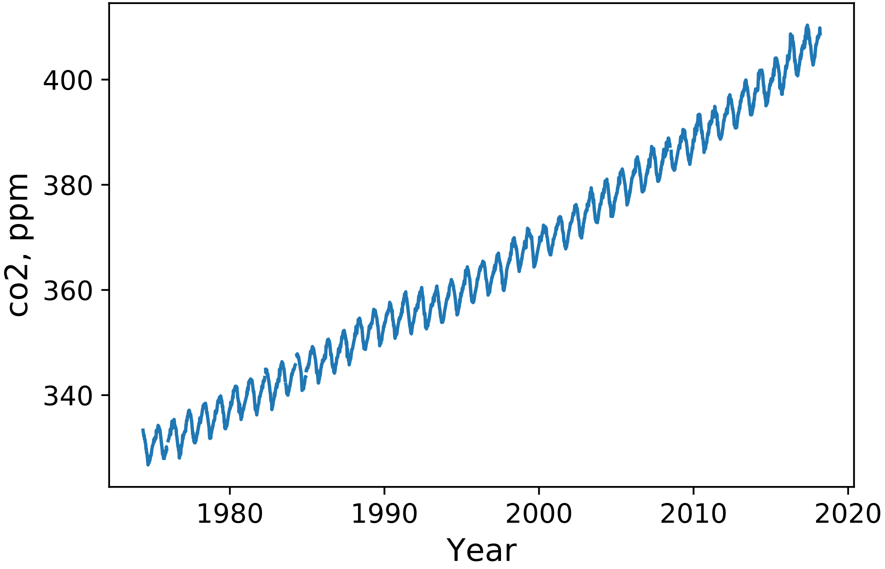
BEGINNING PERIODIC DETRENDING

Periodic Detrending

- Often there is “seasonality” in in real-world data
  - Yearly periods
  - Daily periods
  - Weekly periods
  - Moon cycle periods
- If we know these periods matter, we may want to remove it prior to analysis

CO2 Data

- There is a clear yearly pattern in this dataset
- Visually, we can see the overall trend, but may be clearer if we can remove the trend

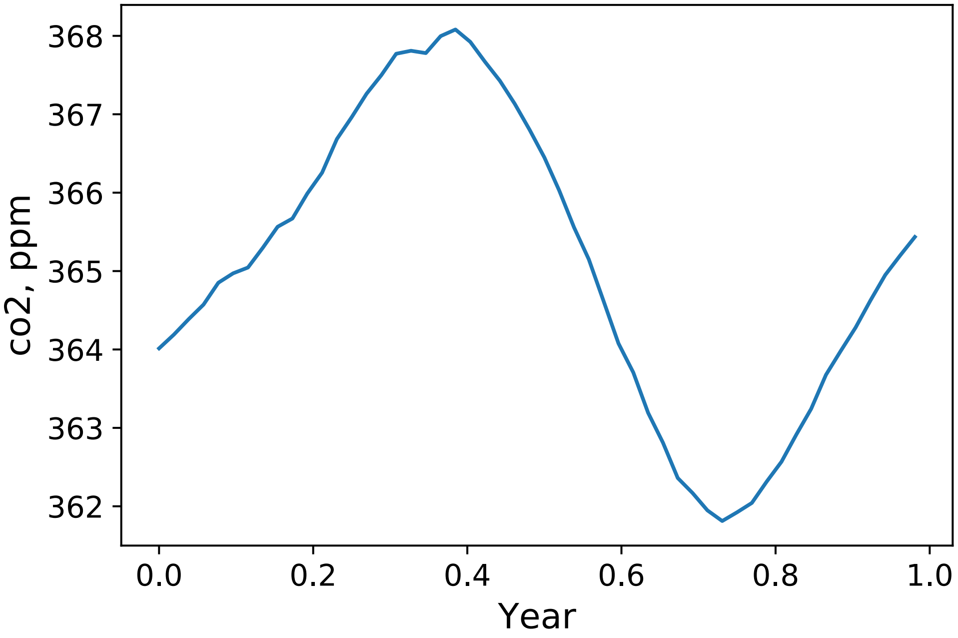


Yearly Average

If (a big if!) data is sampled at the same point in the year, we can average over those points to get the yearly average.

Here I have interpolated to common time points (1/52 of a year), which we will discuss later.

Note: this is an **initial** strategy. Flawed with the dynamics in the background.

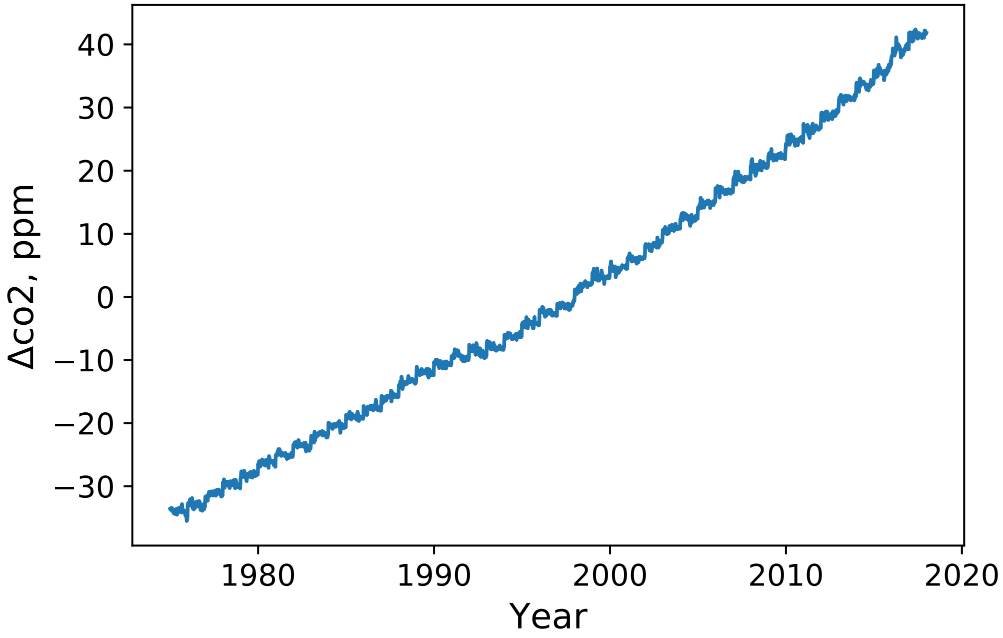


Removing Seasonality

We can remove this seasonal trend from the data by removing the yearly period.

Not flawless but is a reasonable initial strategy.

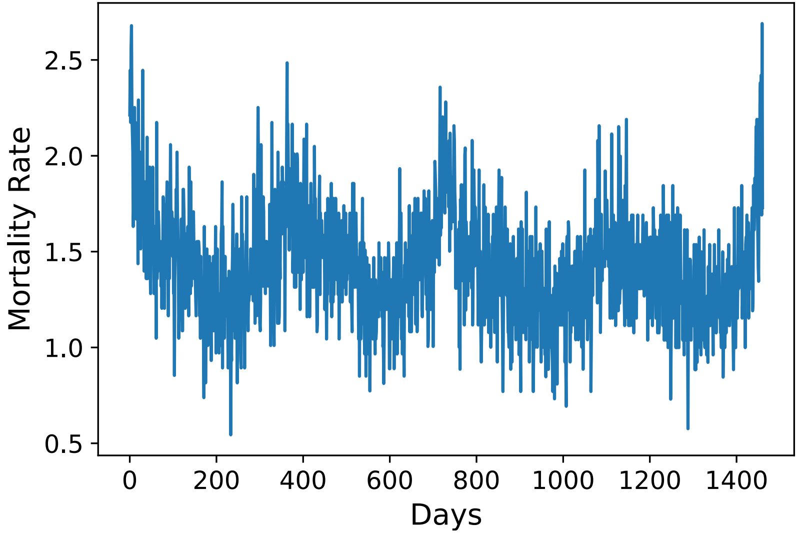
Can improve the strategy by considering frequency properties or including the seasonality in a statistical model.



FREQUENCY DECOMPOSITION

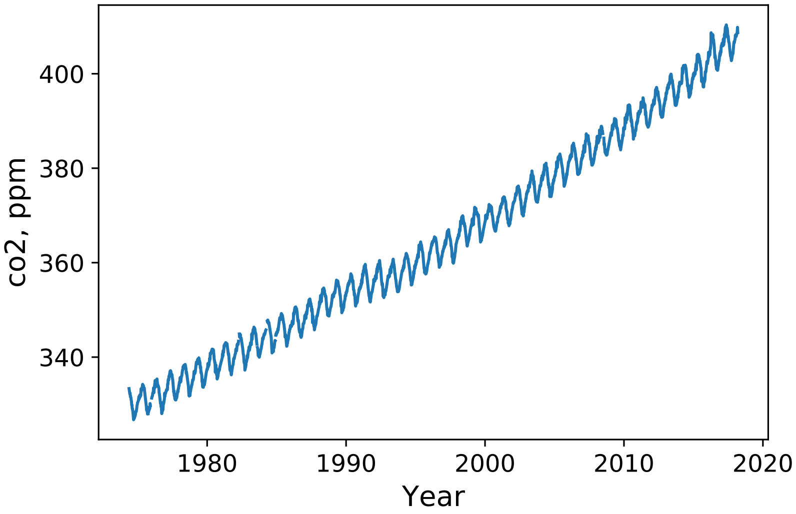
Why do we care about frequency decompositions?

- Often frequency representations are more natural for working with time series data.
- Consider the datasets from last lecture, there are natural periodicities that occur in the data.
  - Can learn the periodicities!
- Can be represented in terms of frequencies (or inverse periods).
- Can facilitate natural extensions and facilitate data preprocessing and analysis.

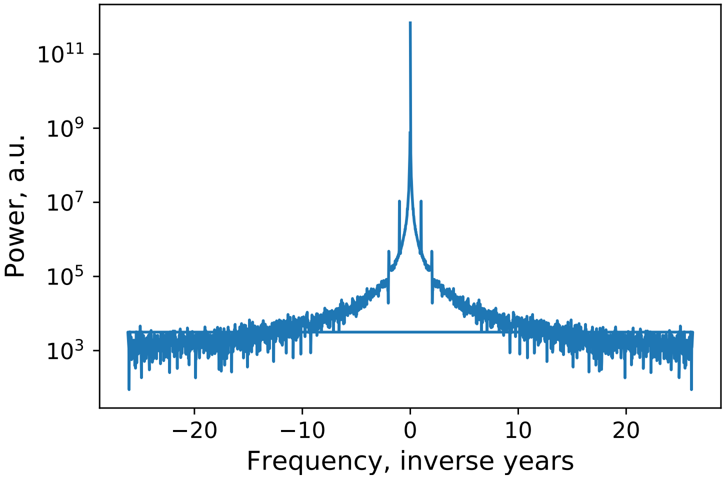


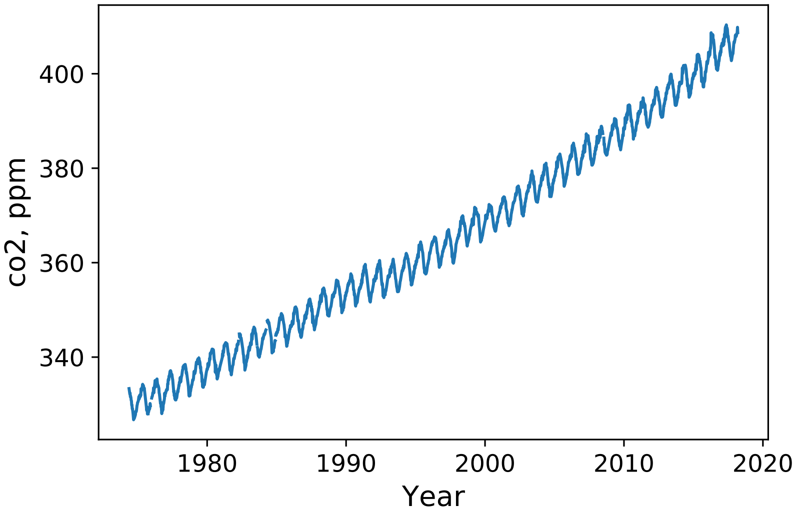
Power Spectral Density

- One thing that we may want to do is to look at the *power spectral density* (PSD) that says how much *power* is at each frequency.
- What do we mean by power?
- What is actually meant by frequency?
- What would happen if looked at the sequence in the time in its PSD?



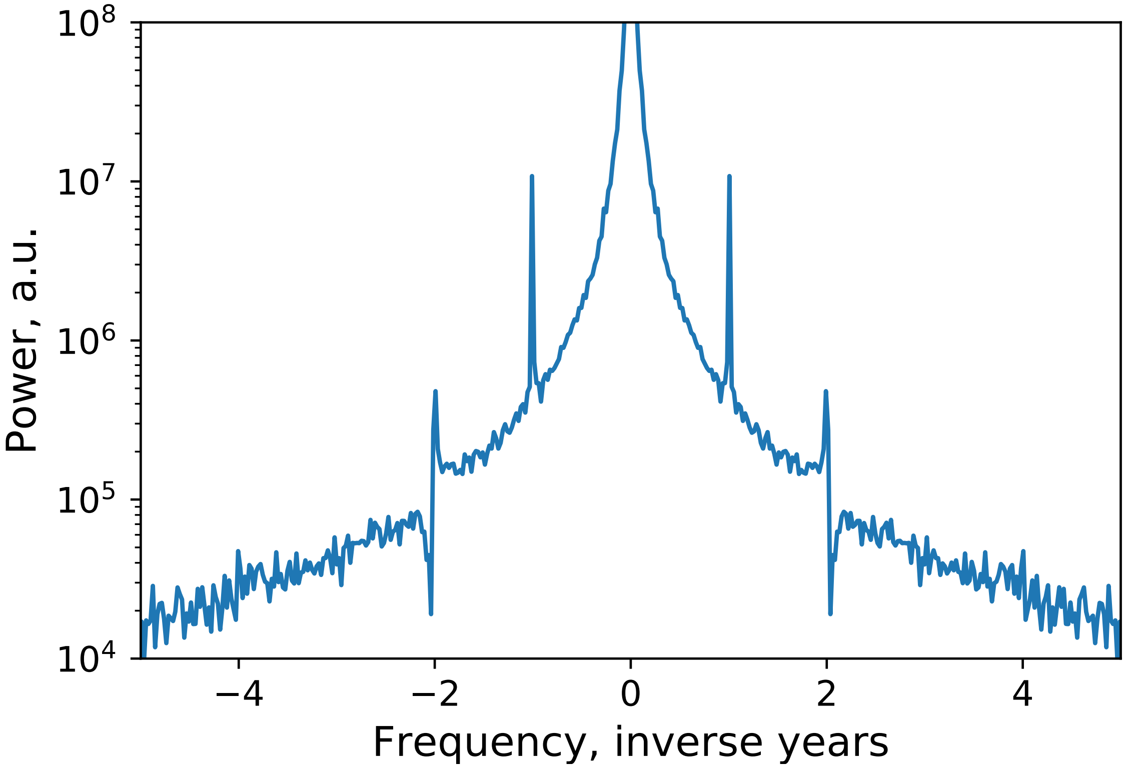
Visualization of Power Spectral Density





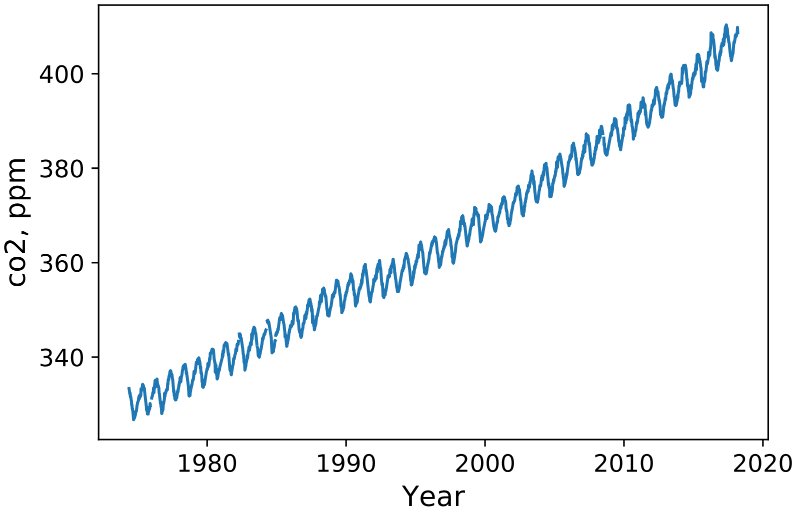
PSD Visualization

We can clearly see increase powe in trends at (-2, -1, 1, and 2) times per year.



This says that we had yearly patterns

- Looking at the original data, we could of easily pointed out yearly patterns.
- We often don't have that luxury, this is very *clean* data that I'm using to illustrate how these algorithms are working.
- Some follow-up questions:
  - How do we define each frequency?
  - How do we define the energy in each frequency?
- Later:
  - How do we deal with temporal variation in frequency power?



Fourier Transform

- The most common approach to transforming to the frequency domain is the *Fourier Transform*. For a continuous function $f(t)$, the transformation to the frequency space via the Fourier Transform is given as:
  - $h(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$.
- And the inverse transform is given as
  - $f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} h(\omega) e^{i\omega t} d\omega$.
- For most signals of interest, this is a one-to-one mapping and the inverse transform exists.
- So what is this actually doing?

Quick aside on Euler Notation

- Quick aside for anyone that hasn't seen it before, the notation:
  - $e^{ix} = \cos(x) + i\sin(x)$
- Where the *i* represents the imaginary number, so $i = \sqrt{-1}$.

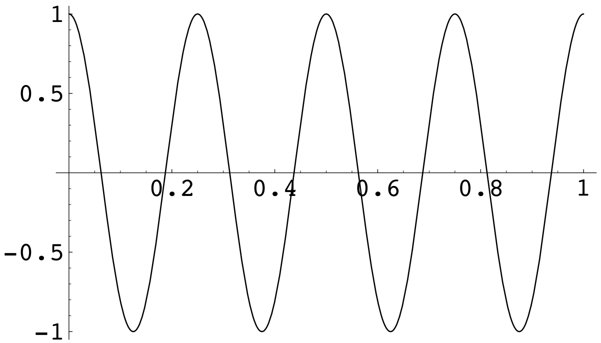
- So the Fourier transform maps signals into the *complex* space (i.e. both real and imaginary).

Fourier Transform Just Multiplies by Sinusoids

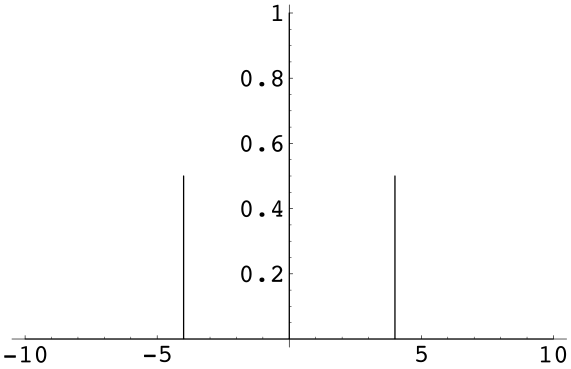
- Revisiting our Fourier Transform, which was represented by
  - $h(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$.
- Then we have that
  - $e^{-i\omega t} = \cos(-\omega t) + i\sin(-\omega t)$
- Thus, $h(\omega)$ is just looking for patterns that match sinusoids with angular frequency of ω.

Examples of Fourier Paris

$\cos(2\pi st)$

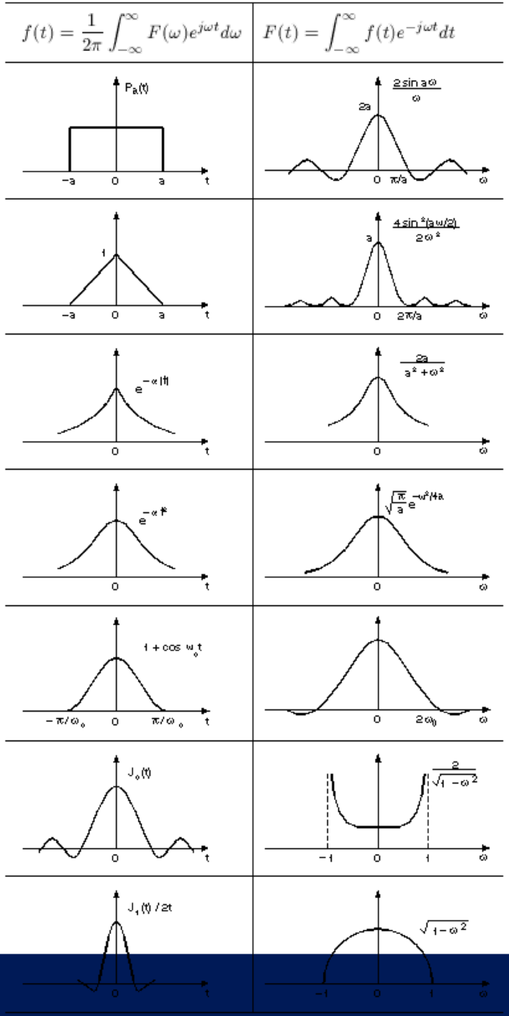


$\frac{1}{2}\delta(u-s)+\frac{1}{2}\delta(u+s)$



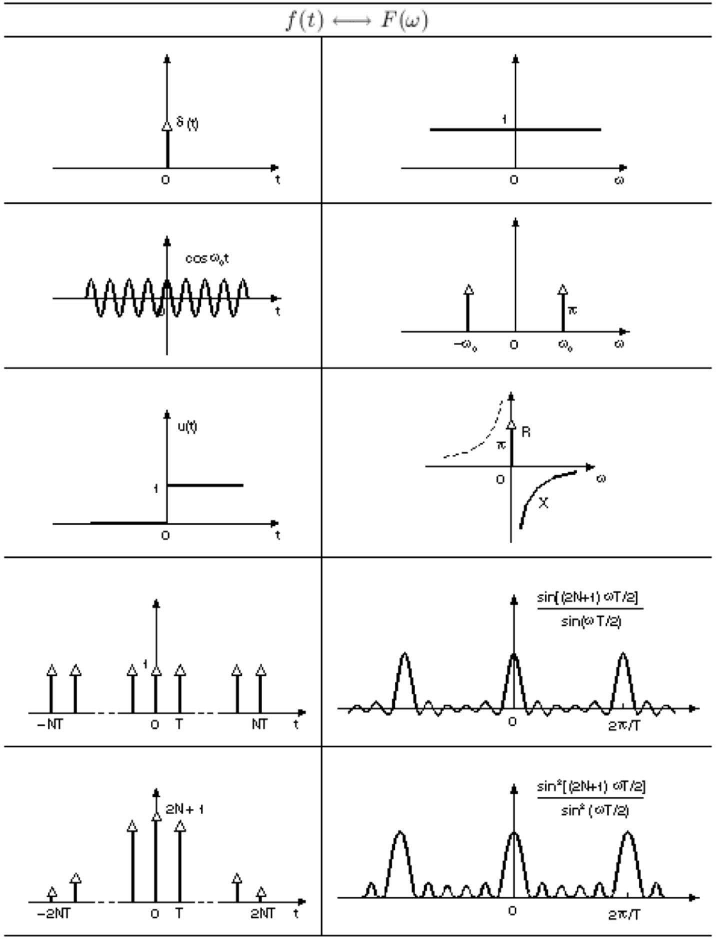
Lots of mathematical Equivalencies

Examples of Fourier transforms



Composition of Frequencies

Transforms of singularity functions



$\delta(x)\delta(y) \Leftrightarrow 1$ $\quad$ $\delta(x) \Leftrightarrow 2\pi\delta(u)$

$\varphi(x) \Leftrightarrow 2\pi\Phi(u)\delta(v)$ $\quad$ $\varphi(x)\delta(y) \Leftrightarrow \Phi(u)$

A Couple of Comments on the Fourier Transform

- Because the Fourier Transform looks for periodic signals, it is very helpful to discover underlying periodicities in the data.
- Typically, we'll be looking at a discrete version rather than the continuous formulation that I've introduced. Not much changes.

- Practically, the Discrete Fourier Transform (DFT) is almost almost implemented via a very famous algorithm called the Fast Fourier Transform (FFT), so the call in a programming language is typically under "fft."
- The runtime of the algorithm is $\mathcal{O}(n \log n)$.

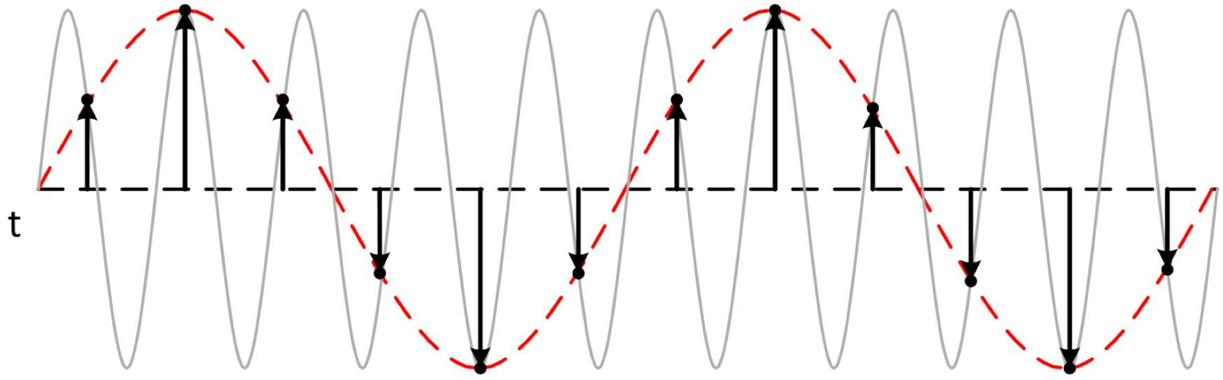
Nyquist Rate

- A very important property of regularly sampled time series (i.e. $t_{i+1} - t_i = \Delta$) is that they have a *maximally detectable frequency*.

- This is given by the Nyquist Rate, which is given as
  - $Nyquist\ Rate = \frac{1}{2} \times Sampling\ Rate = \frac{1}{2\Delta}$.
- If the true frequency is greater than the Nyquist Rate, then *aliasing* happens. The higher frequency will be represented as a lower frequency in the discrete time series.

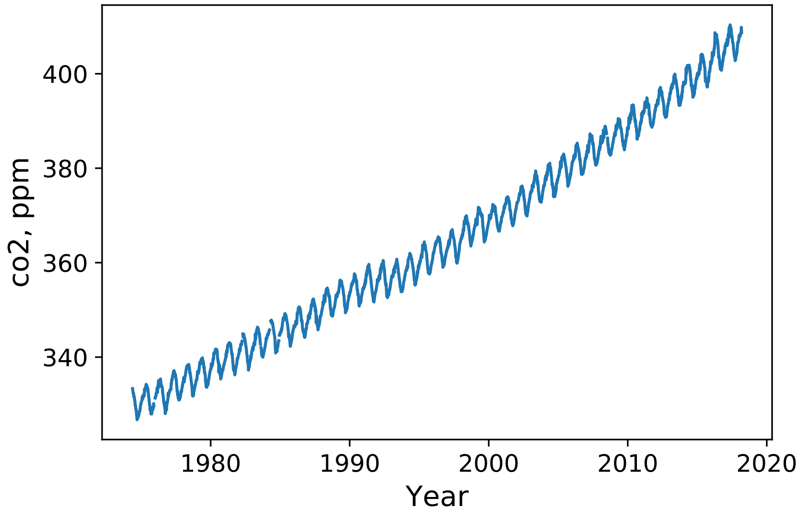
Example of Aliasing

Gray is true signal, and the black gives the observed points. The inferred signal is a much lower frequency sinusoid.

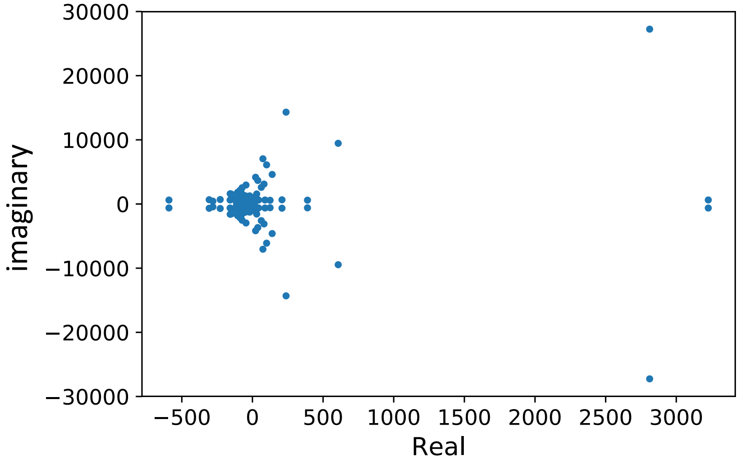


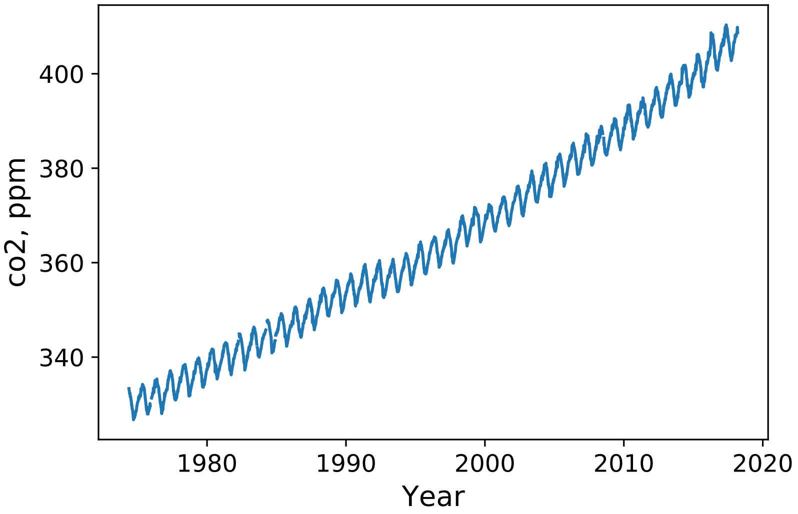
Returning to our CO2 Data

- We want to apply this technique to our CO2 time series to help determine what frequencies are present in the signal.
- Why don't we just take our Fourier Transform via the FFT and apply it and visualize it?



Visualizing the Fourier Transform

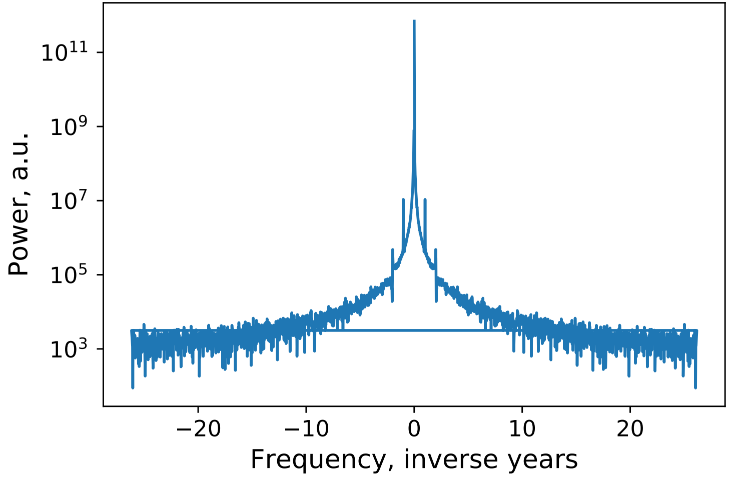




Power Spectral Density

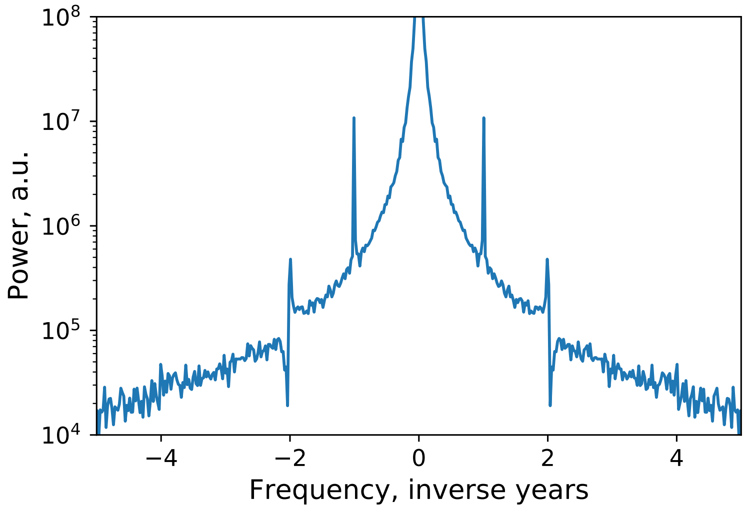
- Instead of visualizing the complex coefficients (which are often hard to interpret), when we visualize the transformation, we typically visualize the magnitude of the coefficients squared.

- The squaring comes from largely historical reasons. This was used heavily in the analysis of power systems and radar, where the power (the squared magnitude) was a more natural unit than the energy (the absolute magnitude).



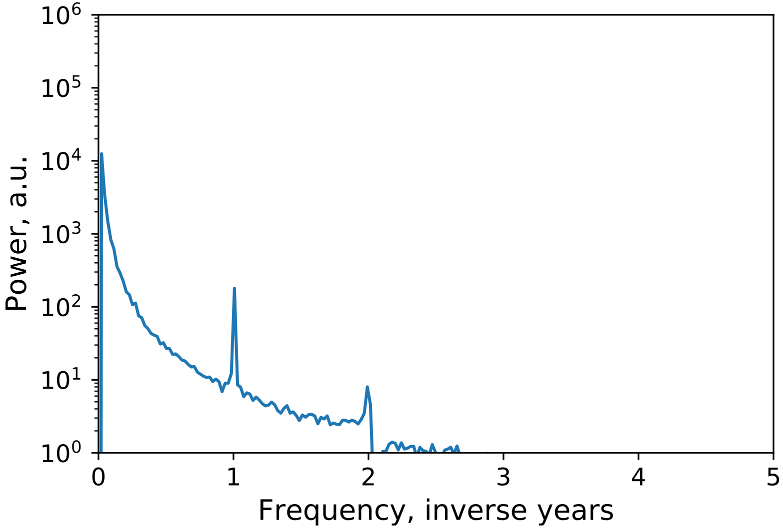
Some additional comments

- Depending on the field, this is often referred to as a periodogram rather than a PSD.
- Some fields also invert the units on the x-axis, so that the units are in periods rather than in frequencies.
- If our original signal is real, then the magnitudes are symmetric across the origin, so it is common to only show the positive frequencies.
- Still existing questions:
  - How can we remove the unwanted periodicities?
  - How can we deal with time-varying signals?



Improving the estimates:

- Directly using the FFT algorithm to estimate the PSD gives a noisy estimate. There are a number of methods to smooth out the spectrum, and can be asymptotically consistent.
- On the right, the estimate from Welch's method is shown.



FILTERING

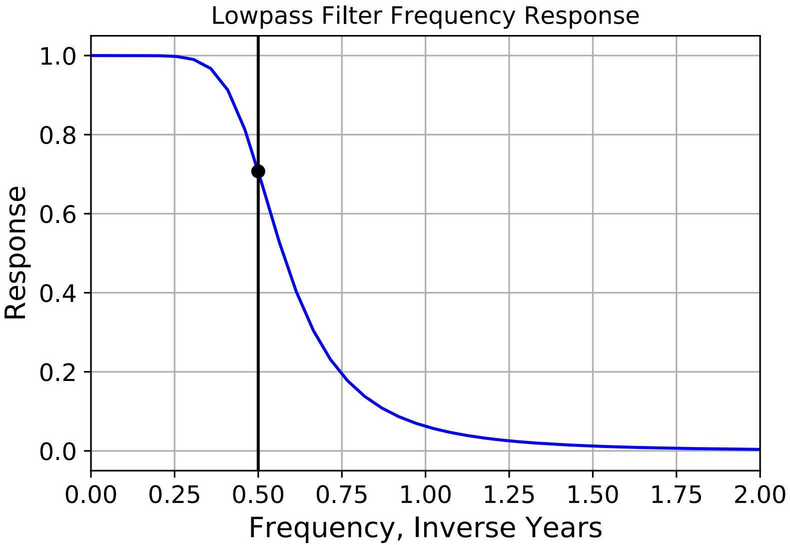
Filtering

- As a preprocessing step in time series, we often want to filter the data. There are 4 fundamental types of filters:
  - Low-pass Filter: Only frequencies below a chosen threshold will be kept in the signal
  - High-pass Filter: Only frequencies above a chosen threshold will be kept in the signal
  - Band-pass Filter: Only frequencies between two threshold will be kept in the signal
  - Notch Filter: A particular frequency (and a small set around it) will be removed from the signal
    - Very common to remove the 60Hz signal

What type of filter we we want for the CO_2 data?

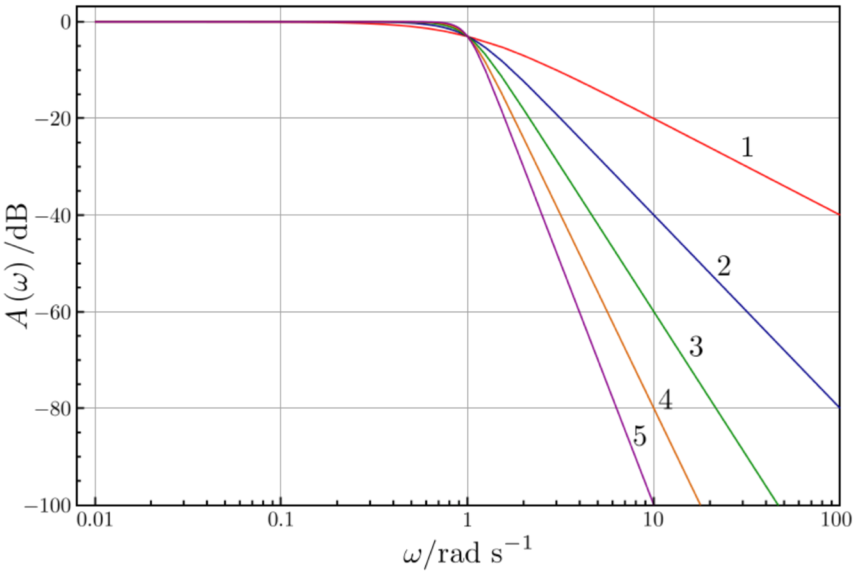
Low-Pass Filtering

- One approach that we could take on the CO2 data is to remove patterns with a higher frequency than our trend of interest.
- We're interested in the long term trends. Can we just filter out our annual periodicity?
- Consider the filter on the right that we could apply to the data. Is this a good filter?



Choosing the order of the filter

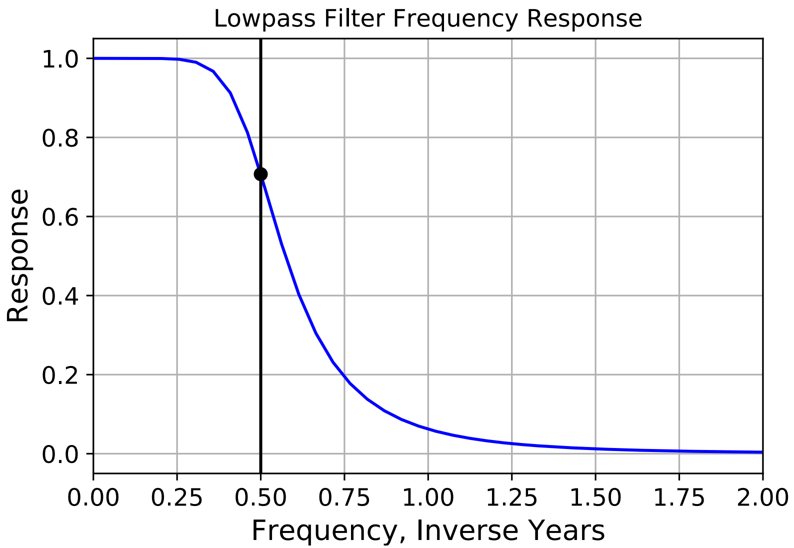
- In the filter, we get to choose a parameter, typically called an "order" that determines the complexity (or history) of the filter.
- This becomes a tradeoff:
  - The higher the order, the more precise the frequency cutoff is in the system (see right)
  - The higher the order, the more the data is manipulated; it becomes more sensitive to outliers, noise, and numerical precision. It also often *shifts* temporal events.
- Typically, want to choose a *low* an order as your application will allow.



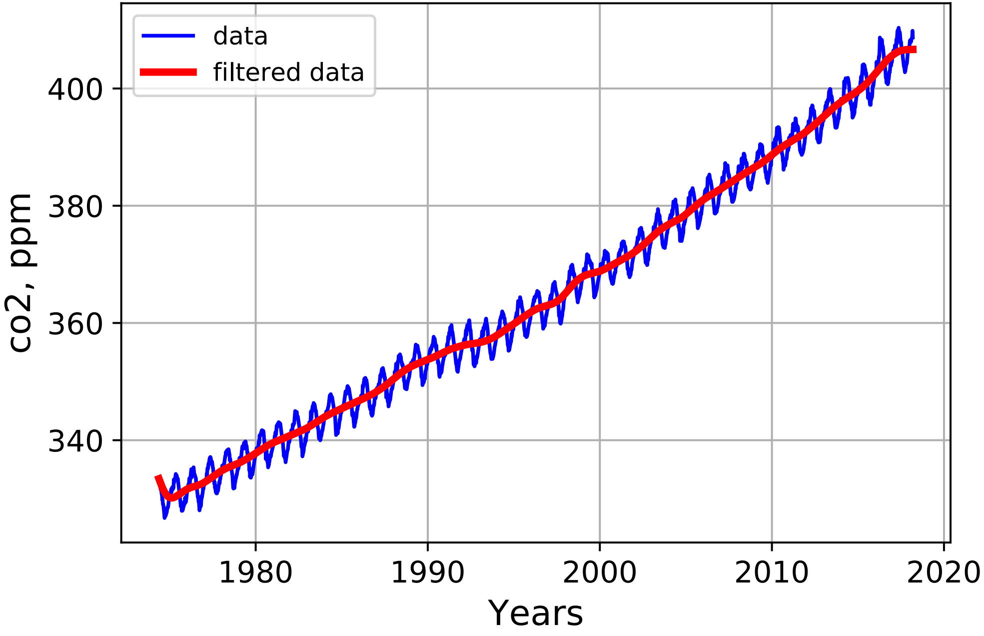
Applying the Filter

- The filter on the right is a 4th order Butterworth low-pass filter with a cutoff point at .5 yrs^{-1} .
- This is a very standard filter, and it works well for many applications.
- Lots of nuance in filter design! We're only touching the absolute basics here.

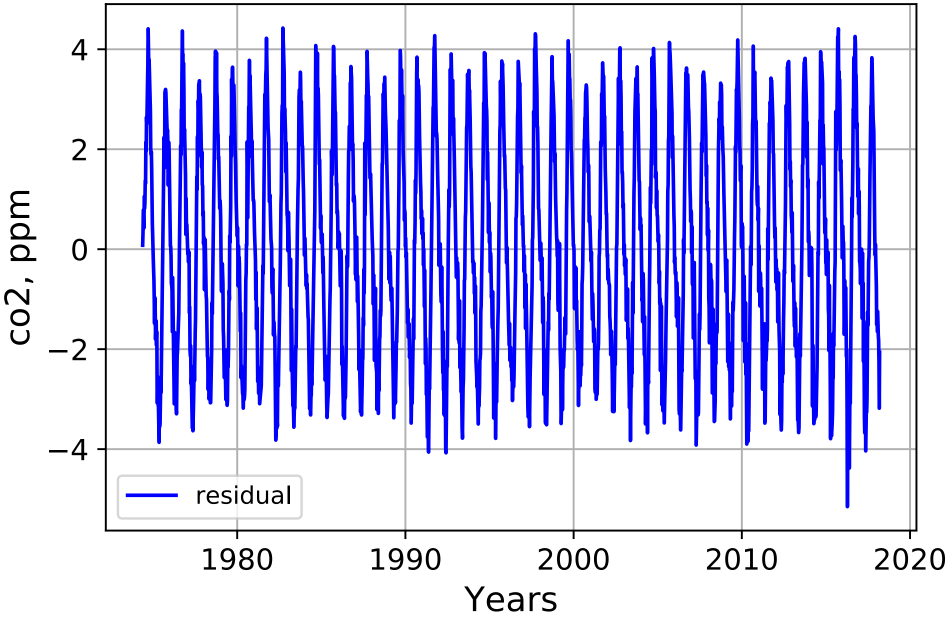
- We will apply it to the co2 data.



Results of Applying the Filter.



Residual After Filtering



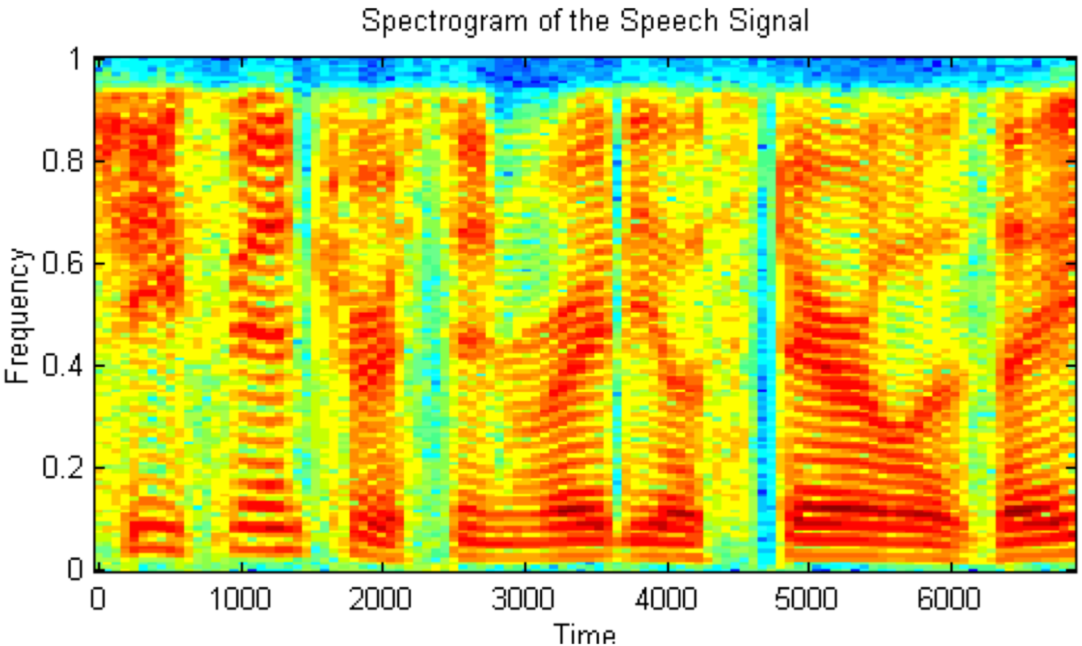
TIME FREQUENCY DECOMPOSITIONS

Time-Frequency Decompositions

- So far we've talked about the Discrete Fourier Transform, which is useful for diagnosing the frequencies present in the *entirety* of the signal or time series.
- However, we often want to understand what periodicities or frequencies are present in a *local* area in time.
- Some applications:
  - Detecting sparse events based on frequency signatures (e.g. neural firing)
  - Non-stationary data
  - Many natural tasks (e.g. speech, video, weather)

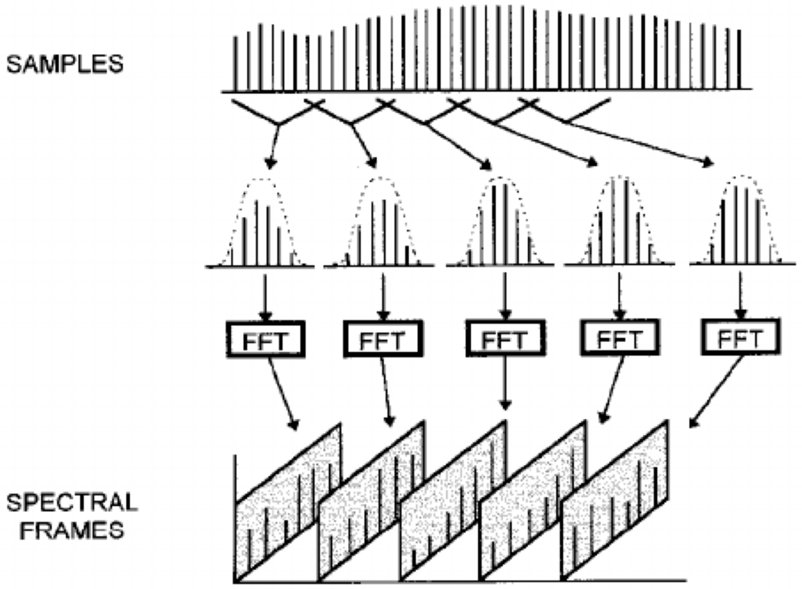
Time-Frequency Decomposition

One thing we often do is to construct a spectrogram, which shows the power as a function of frequency and time.



Short-Time Fourier Transform (STFT)

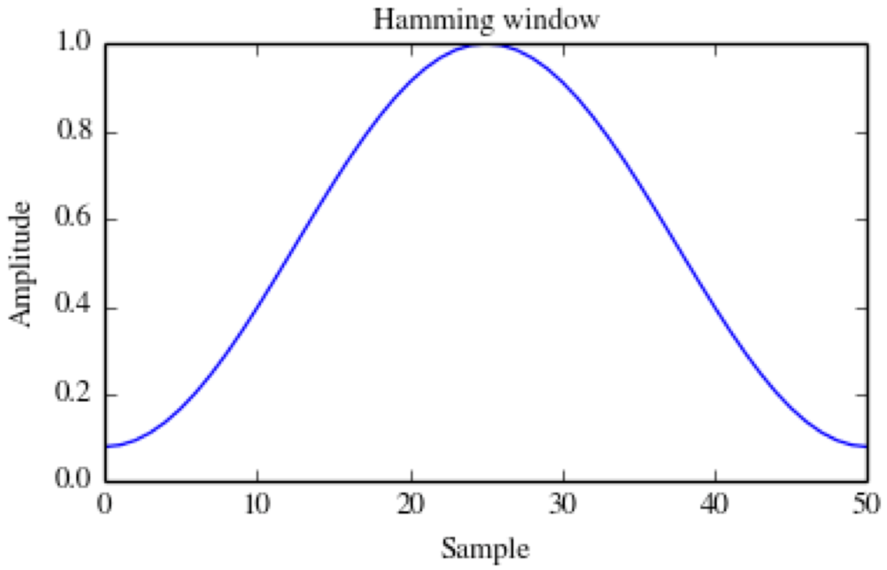
- One approach to try to build up local frequency transformations is to use *windows* of time.
- Each window consists of a time period (e.g. 1 second) that we want to know the local frequency decomposition at.
- Often the window is tapered at the edges so that there are smoothed edges rather than sharp edges (related to our discussion of order in a filter)



What does such a window look like?

The tapered edges (instead of the sharp edges if you just use a discrete period) help stabilize the estimate.

Many different types of windows, but a few standard ones have become default.

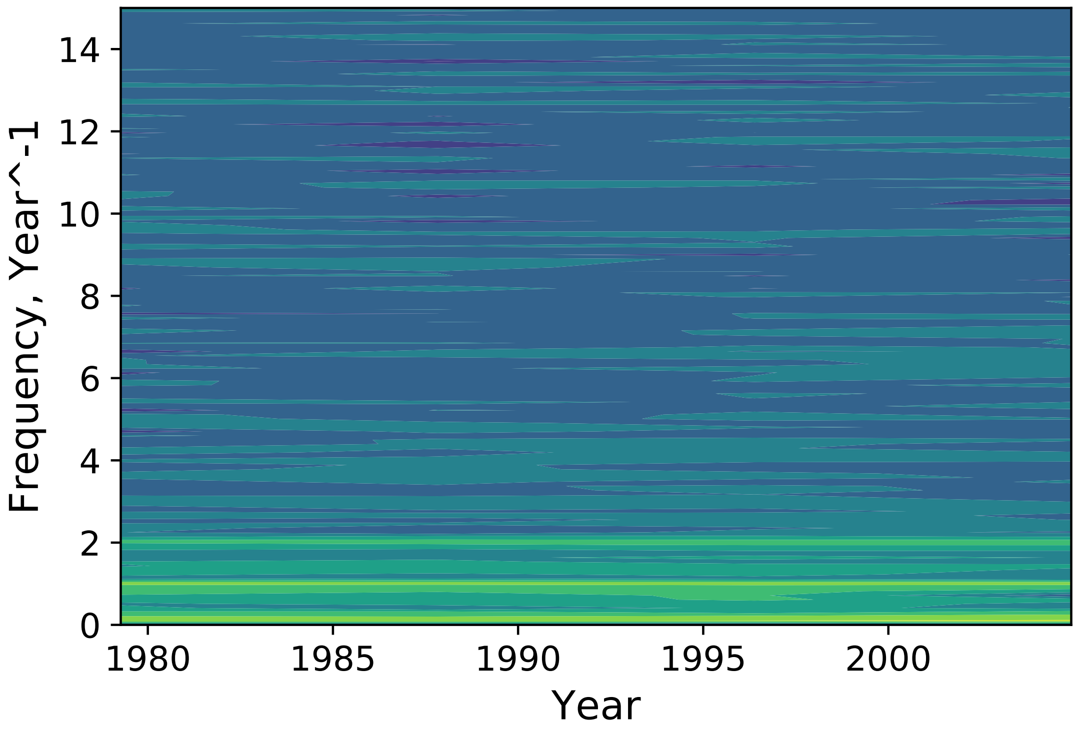


How do we choose the window size?

- There is a trade-off in window size.
- As the windows become smaller, we get greater precision in time (i.e. we know what point in time we're looking at), but we lose precision in frequency (i.e. become less sure that we know frequency we're trying to examine).
- Overlapping windows are common.
- Considering estimating the frequency of a 1Hz wave from a .01s window with noise. Will be very difficult to estimate the true frequency exactly!
- The precision in frequency and time is bounded by Heisenberg's uncertainty principle (or at least one with essentially the same mathematical proof).

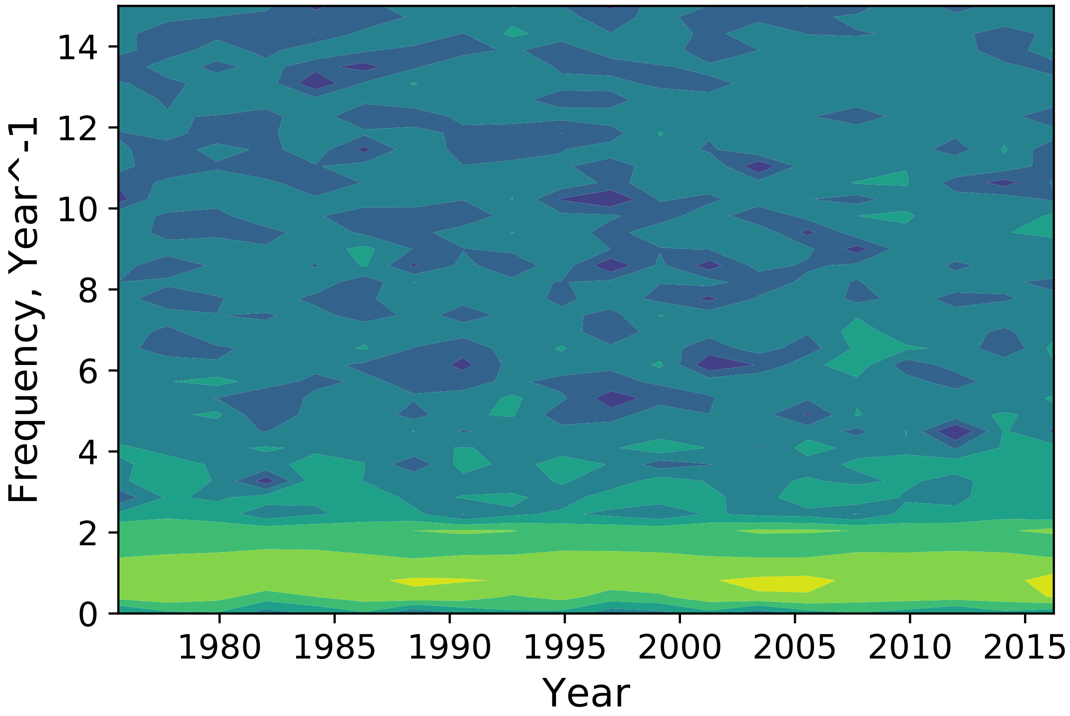
Spectrogram with a Long Window

Take the window, use FFT (+ some smoothing) to estimate the power spectral density through time.



Spectrogram with a Shorter Window

Lose the definition on the frequency space



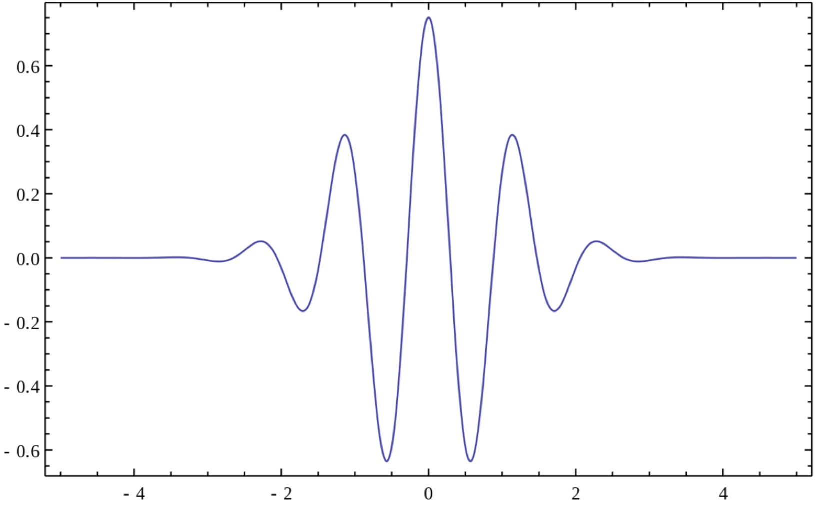
Frequency-Dependent Window Sizes?

- Consider the following uncertainties on our frequency estimates:
  - 2 Hz +/- 5 Hz
  - 500 Hz +/- 5 Hz
- This seems pretty terrible for the 2 Hz band and pretty good for the 500 Hz band.
- The constant window in the STFT implicitly gives us the same frequency vs time precision tradeoff on every frequency band.

- Would we be willing to trade off frequency precision at high frequencies for greater temporal precision and vice versa?
  - Depends on the application…

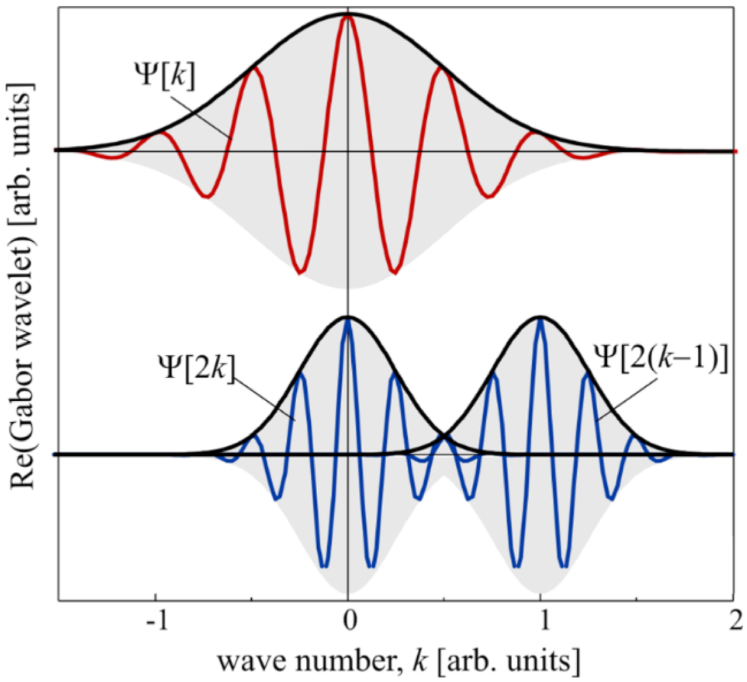
Wavelets

- *Wavelets* combine a temporal window with a particular frequency band.
- The example on the right is called the *Morlet Wavelet*, which is given as
- $\psi(t) \propto e^{-.5t^2}(e^{i\sigma t} - const)$
- There is only a single parameter in base definition of the wavelet, which is given by σ, and it controls the global tradeoff between frequency and time precision.
- This base function is scaled $\psi(\alpha t)$ when applied to data, but keeps the same shape.
- Widely used in things like electrocardiograms.



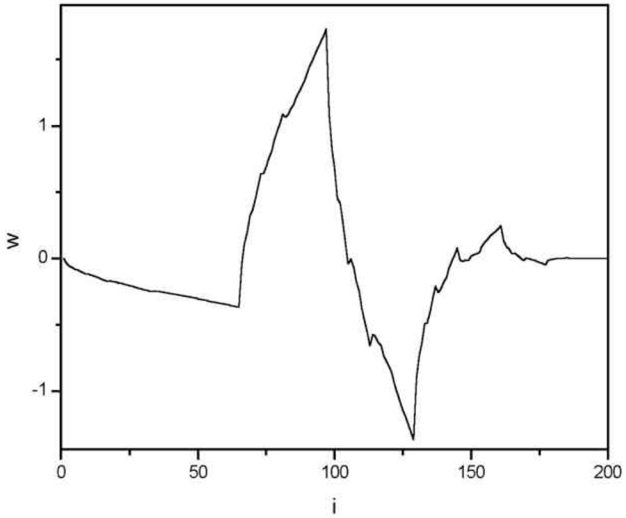
Scaling the “Mother Wavelet”

- When a wavelet is applied, it keeps it’s base shape.
- Shrinking the wavelet by two reduces the amount of time covered by 2 and increases the target frequency by a power of 2.
- Gained:
  - Time precision (x2)
- Lost:
  - Frequency Precision (x1/2)
- This can be immensely helpful depending on the application



Many Different Types of Wavelets

- Many different families of wavelets:
  - Morlet/Gabor
  - Meyer
  - Mexican Hat
  - Haar
  - Daubechies (DB-4 shown at right)
  - Coiflets
  - Biorthogonal
- Don't need to know all of these, but all have applications. (Daubechies/Biorthogonal powers JPEG-2000)

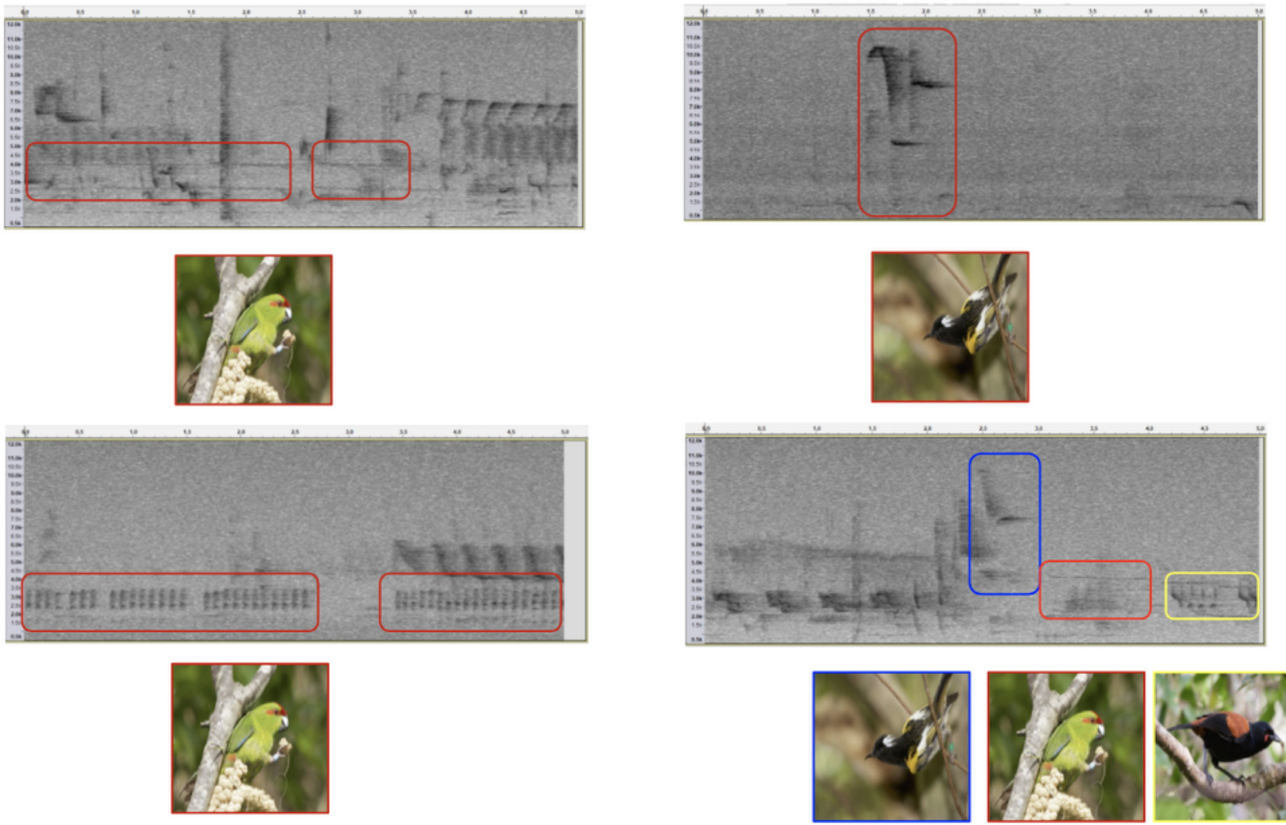


Caveats:

- If you're trying to build a real-time application, then using any of these transforms is going to introduce *lag*.
  - You can't get the frequency transformation until you've observed the complete window.
- Both of these transforms lead to pretty similar types of results; many situations where one works better than the other, but comes with a case by case basis.
- The features generated by a time-frequency decomposition can be input into a model predicting outcomes.
  - One issue: huge proliferation of features!
  - Often pairs with dimensionality reduction
  - If you're using power spectral densities, often nonnegative matrix factorization gives better predictions and is more interpretable (personal opinion, not proven)

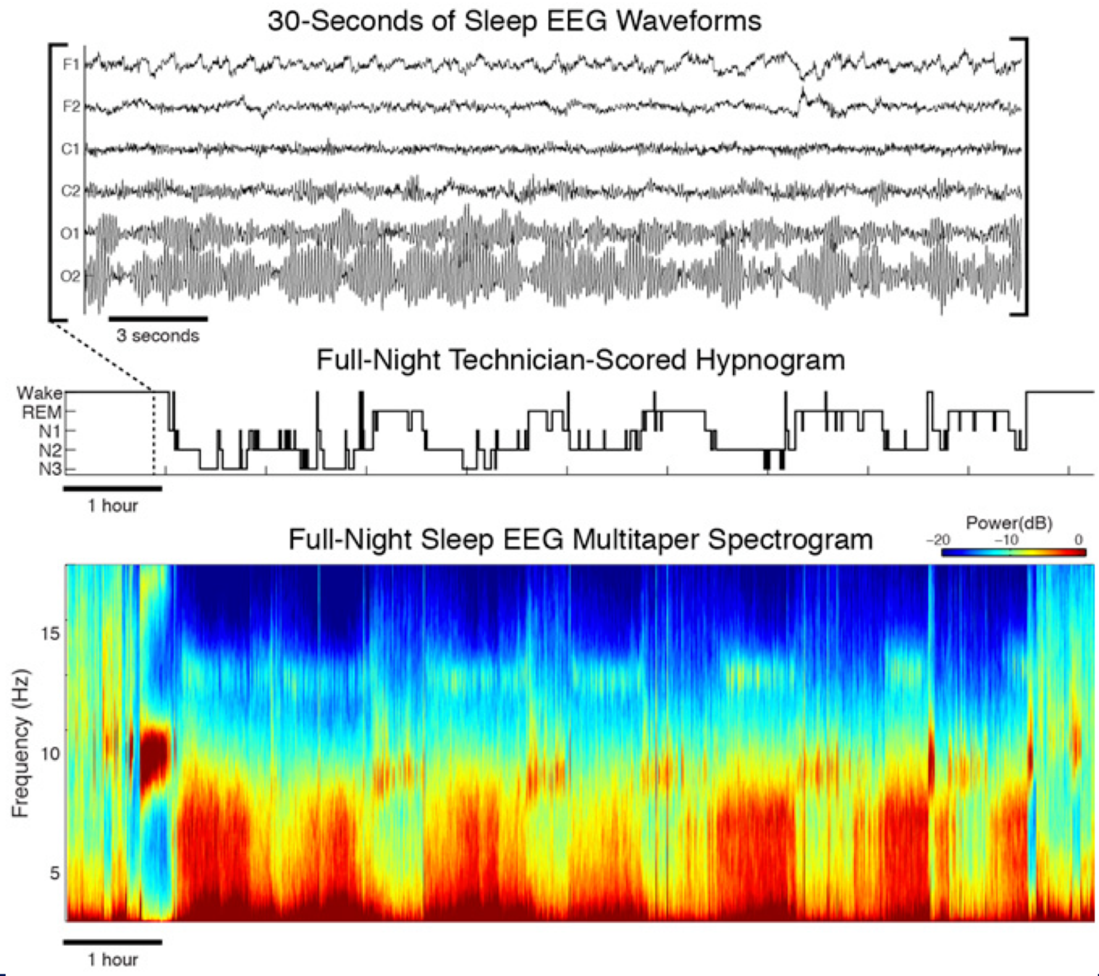
Lots of applications

A recent example in using spectrograms + classification techniques to learn birdsongs from different species



Widely Used to Generate Features

Here, these are measured electrical recordings



Conclusions

- Frequency analysis can help us clean up data and decompose it to help us understand.
- Time-frequency decompositions can help us understand the temporal locality of the signal.

- In the future:
  - Can be included in prediction models
  - Can be included in statistical time-series models

- Can also build intuition for time series models. Autoregressive models (which we'll briefly talk about) can be viewed in the frequency space.