



Dados não estruturados - Processamento de sinais de Áudio

PDS - Aula 10

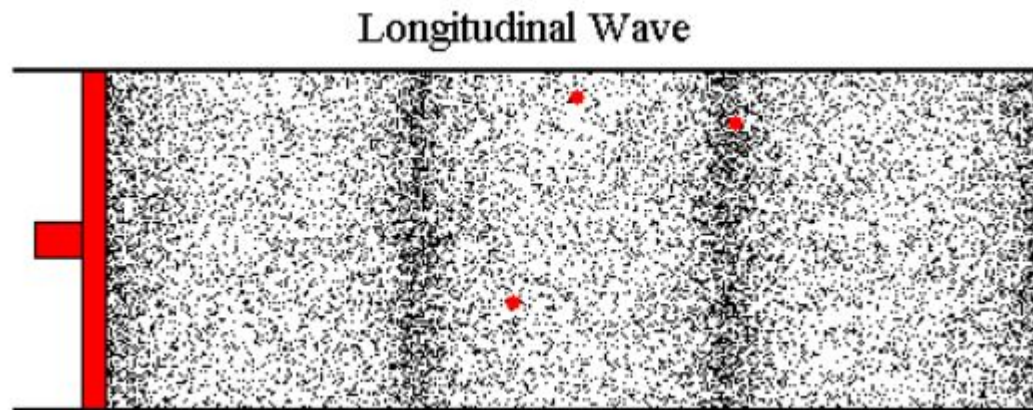


Objetivos

- Introdução ao processamento digital de sinais de áudio (PDS)
- PDS em Python

O que é um sinal de Áudio?

O som é um fenômeno ondulatório que consiste na perturbação (compressão e expansão) periódica de moléculas de ar sob a ação de alguma fonte sonora (voz, fone de ouvido, alto-falante, etc.)



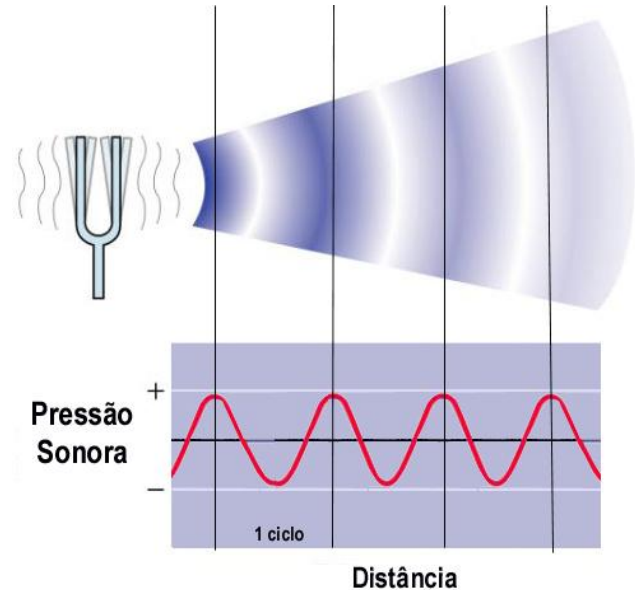
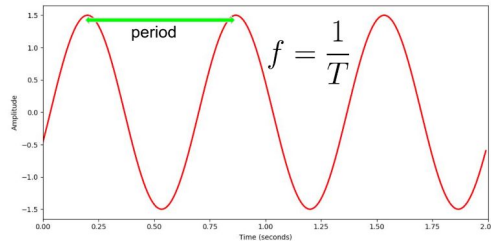
Conceitos: Frequência e Período

Período T

Frequência f

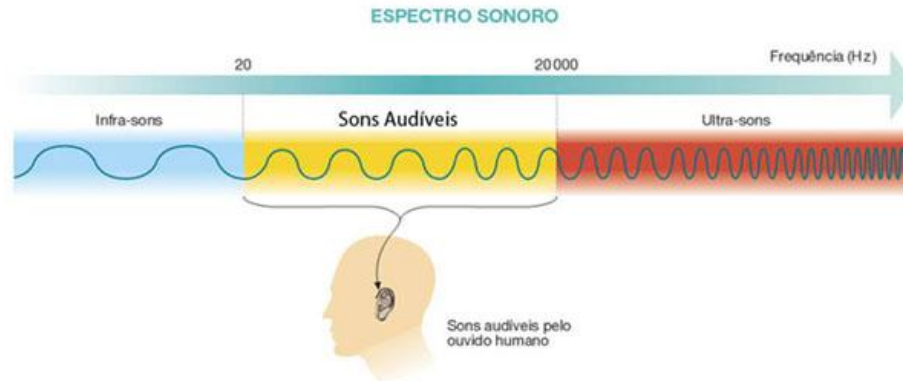
$f = 1/T$ (unidade **Hz** relacionado com T em **s**)

Ex: Se a frequência desse diapasão for de **440 Hz**, isto significa que ele vibra **440 vezes em cada segundo!**



Conceitos: Tonalidade

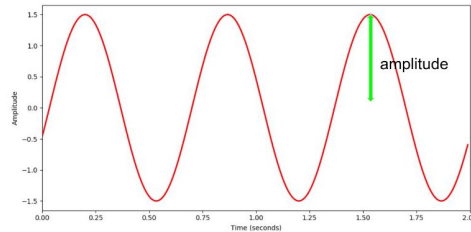
- Um outro poderia vibrar a 125 Hz e o nosso ouvido detectará essas diferenças como tonalidades diferentes: o de **maior frequência**, soaria mais **agudo**, e o de **menor frequência** soaria mais **grave**.
- No entanto, o ouvido humano só é capaz de ouvir sons que têm frequência entre 20 e 20.000 Hz.



Conceitos: Amplitude A

A amplitude de uma onda refere-se à intensidade da perturbação.

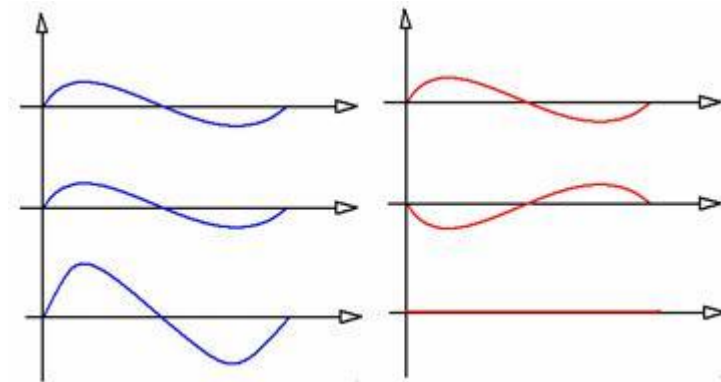
A unidade utilizada para a medida depende do tipo da onda. No caso de ondas sonoras e sinais de áudio, a unidade utilizada é o decibel (dB). A intensidade sonora é percebida como o "volume" de um som.



Conceitos: Fase (phi)

Duas ondas estão em fase quando suas cristas (topo da onda) das ondas estão alinhadas. Neste caso, ocorre **interferência construtiva**: há formação de uma onda cuja amplitude é a soma das amplitudes das duas ondas.

Se as ondas estão completamente fora de fase, há **interferência destrutiva**: subtração das amplitudes e destruição da onda.





Digitalização do Som: Amostragem

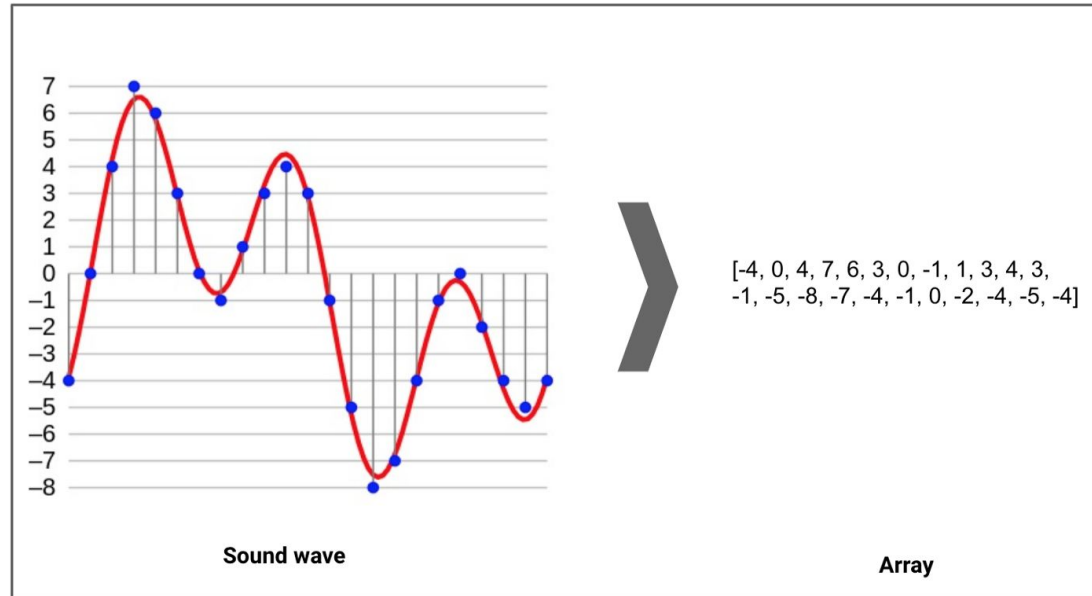
- **Amostragem (Sampling):** divisão da onda sonora em faixas de igual duração. Normalmente, os áudios são digitalizados a taxas de frequência entre 8kHz e 48kHz. Em um CD, o áudio é digitalizado à taxa de amostragem de 44.1kHz, ou seja, o som é dividido em intervalos de $1/44.100$ segundos, o que equivale a dizer que há 44.100 amostras de som em um segundo!



Digitalização do Som: Quantização

Quantização (quantizing) e codificação: consiste em tomar a amplitude do som nos pontos em que houve discretização, e armazenar esses valores de amplitude em uma lista numérica. Na quantização uniforme, o sinal é codificado em valores 8 bits (256 níveis) ou 16 bits (65.536 níveis).

Digitalização do som: Amostragem e Quantização



Digitalização do som



Amostragem: 44.1 kHz

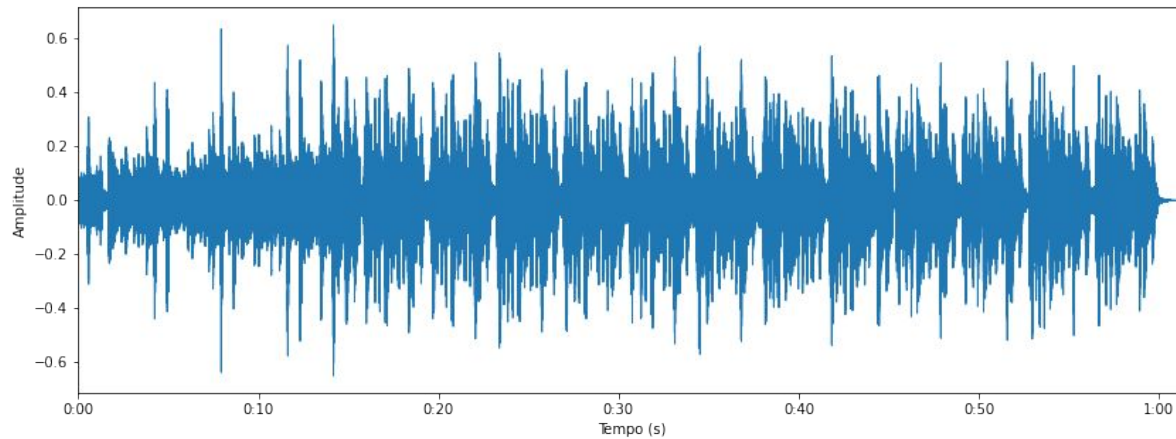
Quantização: 16 bits (65.536 níveis)



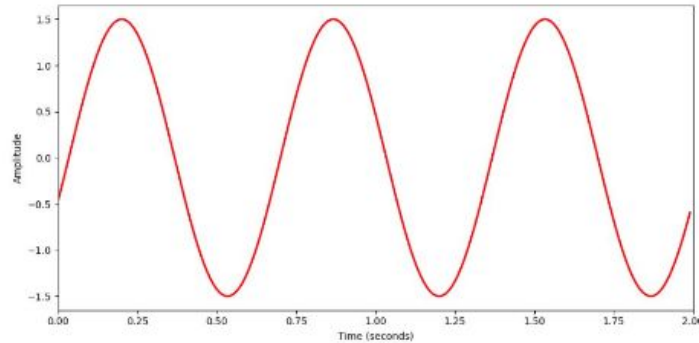
Aplicações

- 1) Reduções de ruído e melhora da qualidade sonora (filtragem digital, gps/gnss, etc.);
- 2) Transmissão, armazenamento e compressão de áudio (wav, mp3, etc.);
- 3) Síntese de voz;
- 4) Análise espectral (FFT);
- 5) Reconhecimento de voz.

Domínio Tempo

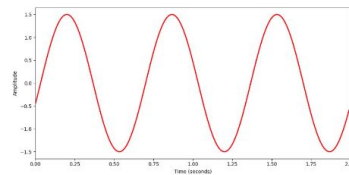


Onda Senoidal



$$y(t) = A \sin(2\pi ft + \varphi)$$

Mão na massa:



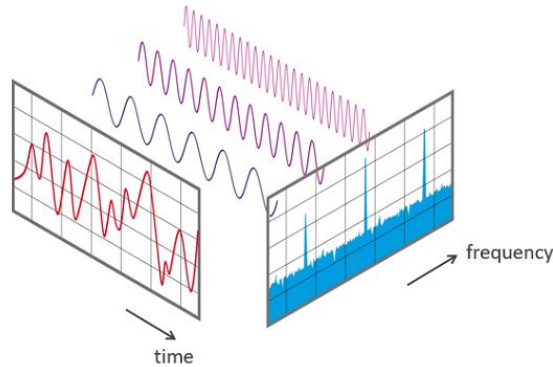
$$y(t) = A \sin(2\pi ft + \varphi)$$

- 1) Digitalizar uma onda senoidal no python

Frequência de amostragem de 44kHz, amplitude de 2 (sem unidade) para uma onda de com intervalo de 1 segundo e frequência natural de 200Hz dentro do range audível com fase igual a zero

- 2) Fazer o gráfico Amplitude por tempo (0 a 1s)
- 3) Fazer o gráfico Amplitude por tempo (0 a 0.002s)
- 4) Tocar o áudio com o `from IPython.display import Audio`

Domínio tempo x domínio frequência



Séries de Fourier

- Qualquer função periódica pode ser composta pela soma de senos e cossenos

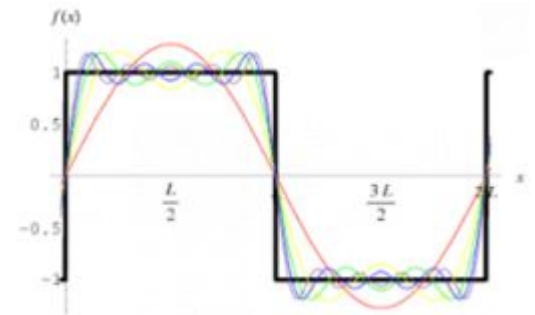


$$f(x) = a_0 / 2 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right) \right],$$

$$a_n = \frac{1}{L} \int_{-L}^{+L} f(x) \cos\left(\frac{n\pi x}{L}\right) dx$$

e

$$b_n = \frac{1}{L} \int_{-L}^{+L} f(x) \sin\left(\frac{n\pi x}{L}\right) dx$$





Transformada de Fourier





Transformada de Fourier

$$F(\omega) = R(\omega) + iI(\omega) \quad A = \pm \sqrt{R(\omega)^2 + I(\omega)^2}$$
$$\varphi = \tan^{-1} \frac{I(\omega)}{R(\omega)}$$



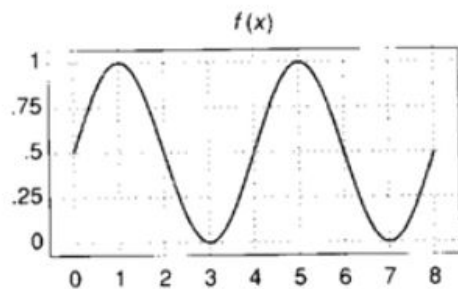
Transformada de Fourier

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i 2 \pi u x} dx$$

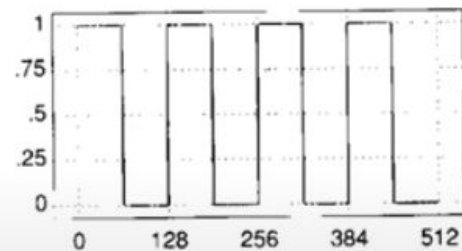
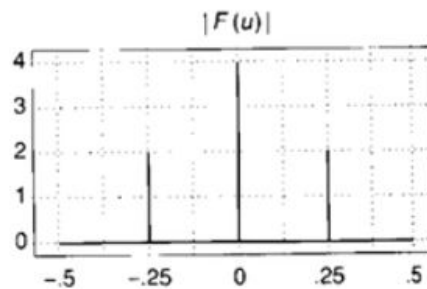
$$\text{Again: } e^{ik} = \cos k + i \sin k \quad i = \sqrt{-1}$$

Spatial Domain (x) \longrightarrow Frequency Domain (ω or u or *even* s)
(Frequency Spectrum $F(u)$ or $F(\omega)$)

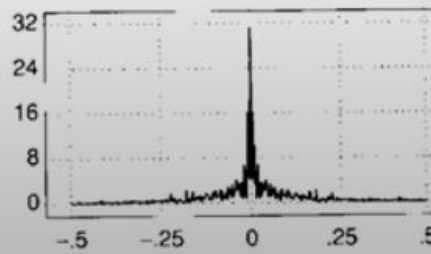
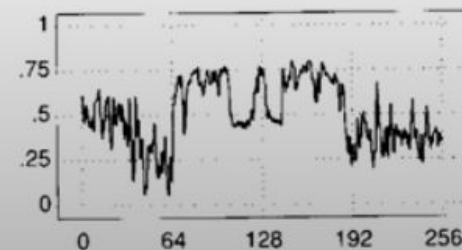
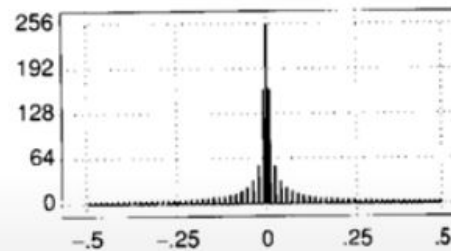
Exem



(a)



(b)





FFT - Fast Fourier Transform

- Método que calcula a transformada de fourier com complexidade $O(n \log n)$



Mão na massa

- 1) Com a função do exercício anterior, aplicar a fast fourier transform à onda senoidal *dica: usar `np.fft.fft`*
- 2) computar a **magnitude** e encontrar a frequência natural e amplitude *dica: usar `np.abs` e dividir pela frequência de amostragem para normalizar*



Mão na massa

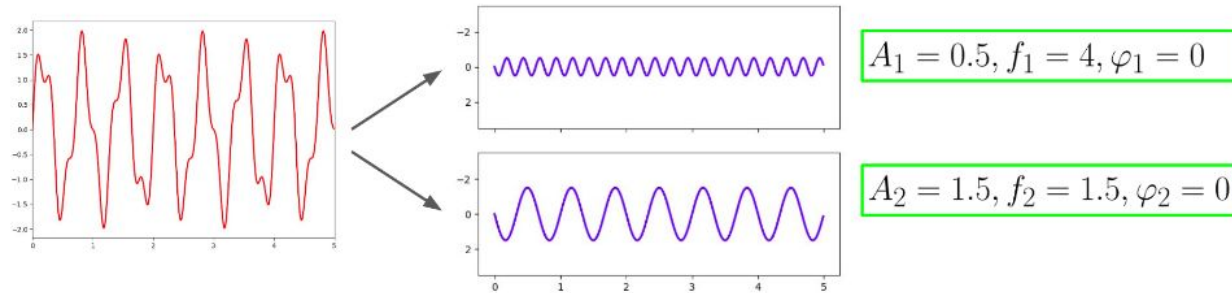
- 1) Montar uma segunda onda senoidal no python, com os seguintes parâmetros:

Frequência de amostragem de 44kHz, amplitude de 20 (sem unidade) para uma onda de com intervalo de 1 segundo e frequência natural de 440 Hz dentro do range audível

- 2) Fazer o gráfico Amplitude por tempo (0 a 1s)
- 3) Fazer o gráfico Amplitude por tempo (0 a 0.02s)
- 4) Tocar o áudio com o `from IPython.display import Audio`
- 5) Aplicar a transformada de fourier e encontrar as frequências naturais
- 6) Somar a primeira e segunda senoidal e graficar (domínio do tempo)
- 7) Aplicar a transformada de fourier na onda do exercício 6 e encontrar as frequências f naturais

Transformada de Fourier

- Decompor uma onda periódica numa soma de senos com diferentes frequências



$$s = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)$$



Librosa

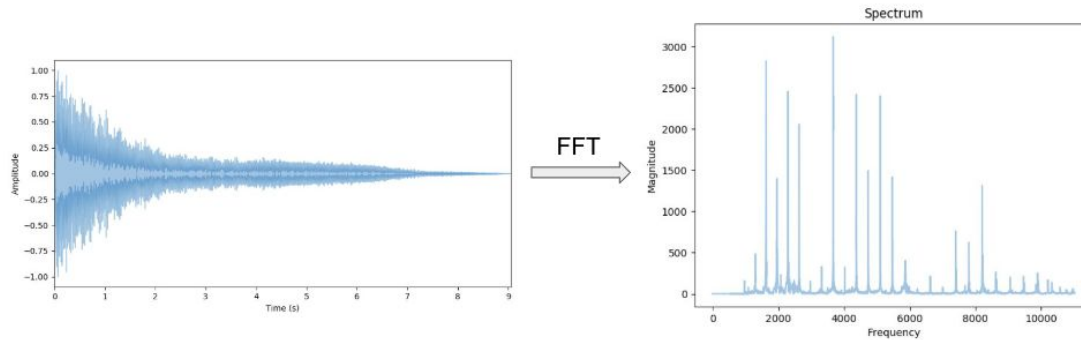




Librosa

- 1) Instalar librosa com **pip install librosa**
- 2) Importar o arquivo .wav com sample_rate (frequência de amostragem) de 22050
 - a) `librosa.util.example_audio_file()`
- 3) Tocar usando o `from IPython.display import Audio`
- 4) Plotar o domínio de tempo do áudio **`librosa.display.waveplot`**
- 5) Aplicar **`np.fft.fft`** e encontrar as frequências naturais e amplitudes
- 6) Salvar o sinal em um arquivo de áudio .wav, usar `librosa.output.write_wav`

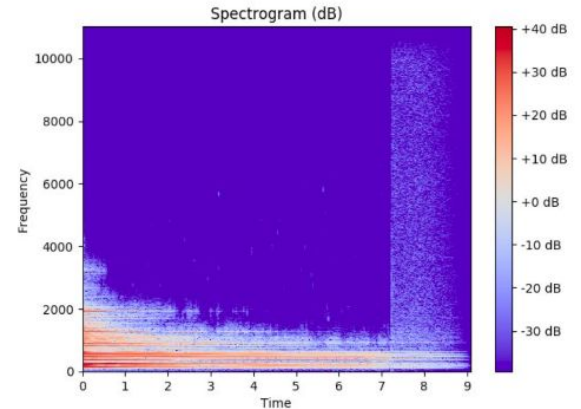
Transformada de Fourier



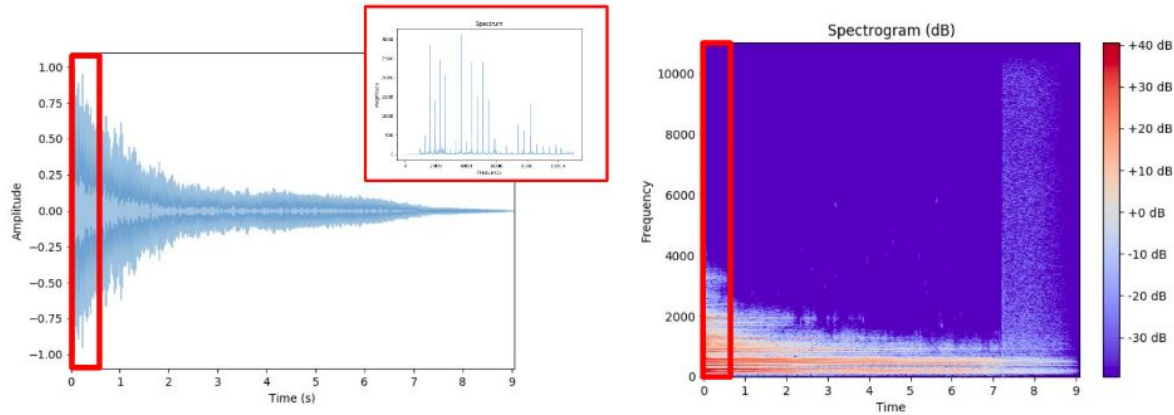
- Convertemos do domínio do tempo para o domínio de frequência
- Mas não temos mais informação de tempo

STFT (Short Time Fourier Transform)

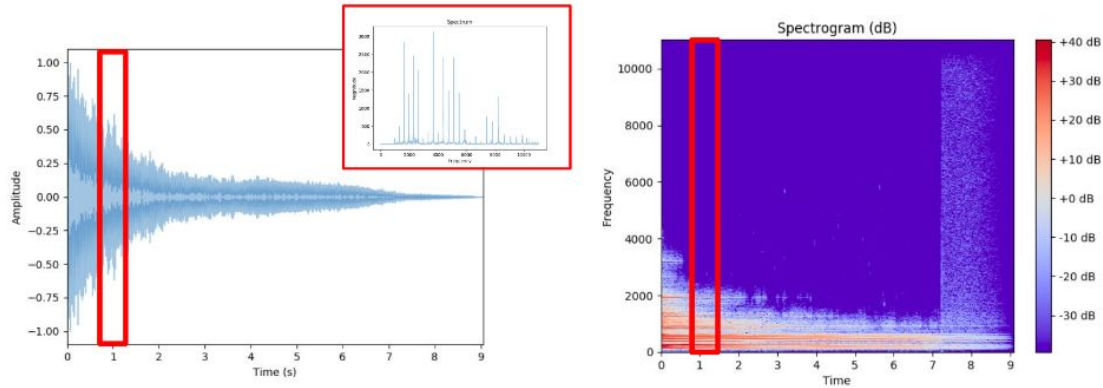
- Computa FFT para diferentes intervalos
- Preserva informação de tempo
- Espectrograma (relação de tempo + frequência + magnitude)



STFT (Short Time Fourier Transform)



STFT (Short Time Fourier Transform)





Próxima aula...