

# Lista 6 - Análise de Séries Temporais em Oceanografia

Lucas Salimene

Utilizando as bibliotecas NumPy, Matplotlib, SciPy, Pandas e Pywt, foi desenvolvido a função:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
import pandas as pd
import pywt
def wavelet(data,time,wavelet='morl',normal=True,filtro=True,
            alta=False):

    if normal==True:
        data_norm = (data - data.min()) / (data.max() - data.min())
        data = data_norm
    else:
        data=data
        data = pd.Series(data).interpolate().values
    if filtro==True:
        databaixa = signal.savgol_filter(data,73,2)
        datasem = data
        data = databaixa
    else:
        data=data
        if alta==True:
            data = datasem-databaixa
        bfs, bPs = signal.welch(data)
        cA, freq = pywt.cwt(data,np.arange(1, 2000), wavelet=wavelet)
        power = (abs(cA))**2
        period = 1./freq
        plt.figure(figsize=(12,10))
        plt.subplot(211)
        plt.title('Welch')
        plt.plot(bfs, bPs,'r')

    f, ax = plt.subplots(figsize=(15, 10))
    a=ax.contourf(time, np.log2(period), np.log2(power), 100,
        extend='both')

    ax.set_title(' Wavelet Power Spectrum ')
    ax.set_ylabel('Período')
    ax.set_xlabel('Frequência')

    Yticks = 2 ** np.arange(np.ceil(np.log2(period.min())) ,
        np.ceil(np.log2(period.max())) )
    ax.set_yticks(np.log2(Yticks))
    ax.set_yticklabels(Yticks)
    ax.invert_yaxis()
    ylim = ax.get_ylim()
    plt.colorbar(a)
    plt.show()
```

Essa função irá realizar a transformada de Wavelet e a PSD pelo método de Welch a partir das variáveis `data` e `time`, ou seja uma variável  $x$  e uma  $t$ . A função aceita os parâmetros `wavelet`, que define o tipo de transformada de Wavelet suportada pelo pacote Pywt se deseja realizar, sendo a padrão da do tipo morl. O parâmetro `normal` define se será realizada a normalização da série temporal. O parâmetro `filtro` define se será realizada a filtragem da série temporal e o parâmetro `alta` define se será realizada a filtragem para obter as altas frequências.

O função pode ser utilizada da seguinte forma para o dado *SST.mat*, que apresenta a temperatura da superfície do mar:

```
wavelet(ssti,time)
```

Que irá retornar as figuras

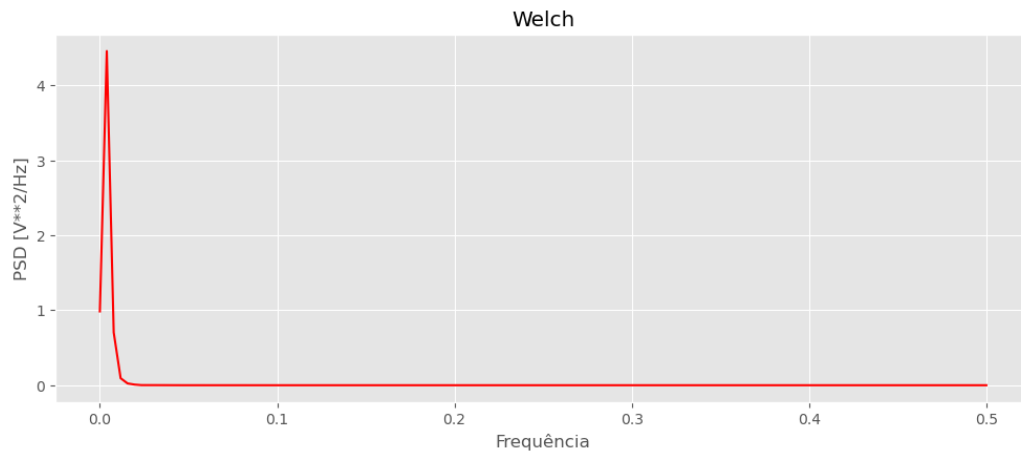


Figura 1: PSD por Welch

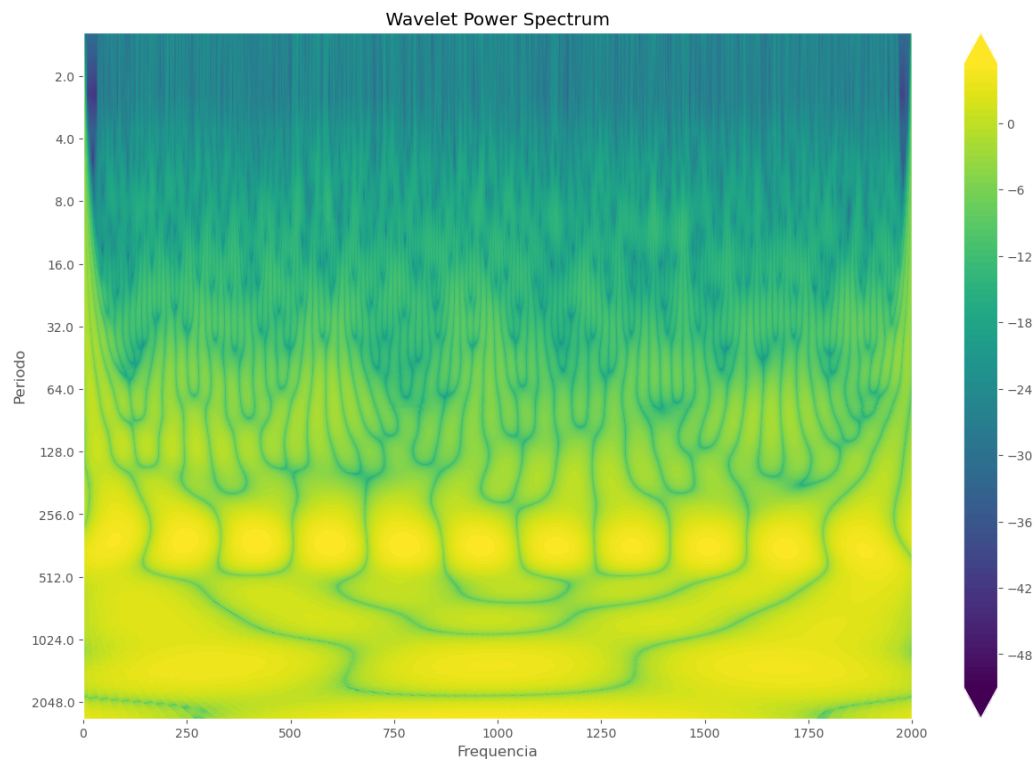


Figura 2: Wavelet por morll

Caso se deseje mudar algum parâmetro, basta adicionar ele na chamada da função, por exemplo, para a wavelet do tipo chapéu mexicano se usaria:

```
wavelet(ssti,time,wavelet='mexh')
```

Para uma wavelet do tipo de Gaus com a utilização de filtragem para apresentar as altas frequências:

```
wavelet(ssti,time,alta=True,wavelet='gaus1')
```

As saídas para diversos casos são encontradas no arquivo Jupyter da lista.