

NeuroMat

Instituto de
Computação



Introdução à análise de séries temporais: 2. Técnicas convencionais

ARTHUR VALENCIO

Pós-doutorando IC/Unicamp

CEPID NeuroMat

Projeto TOPE Unicamp, Campinas, 9 de Outubro de 2019



>>Resumo do curso

Dia 1: Introdução à ferramentas de análise de dados

- Planilhas
- Linguagens de programação de propósito geral
- Linguagens de programação voltadas à dados

Dia 2: Técnicas convencionais de análise

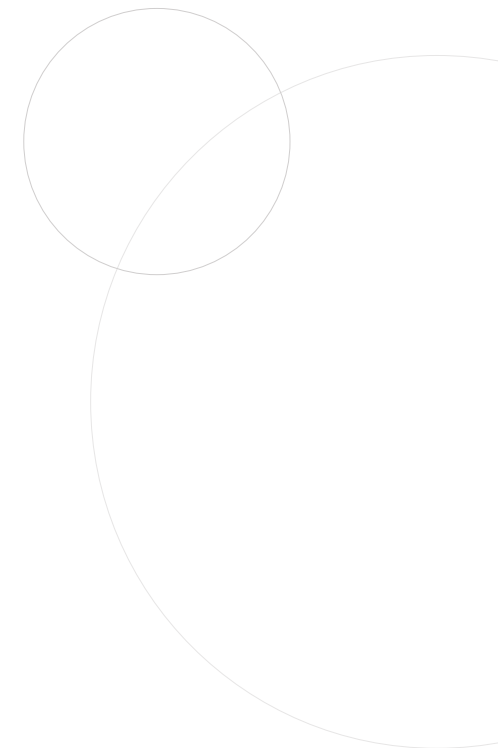
- análise espectral
- wavelet
- correlação de Pearson e Spearman
- regressão

Dia 3: Análise multivariada

- correlação parcial
- análise fatorial
- análise de componentes principais
- análise de componentes independentes

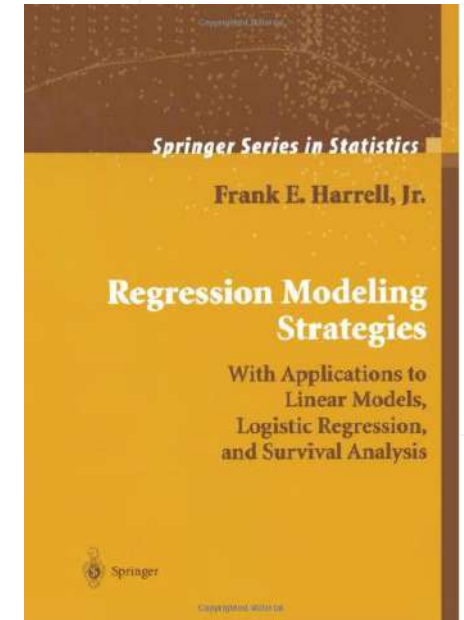
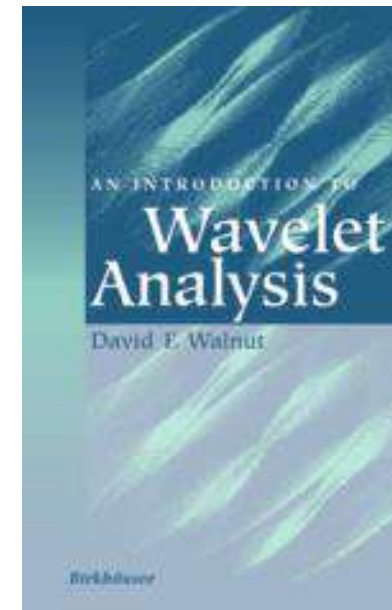
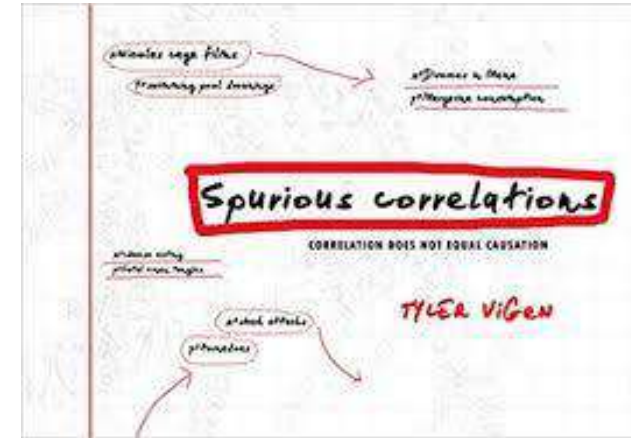
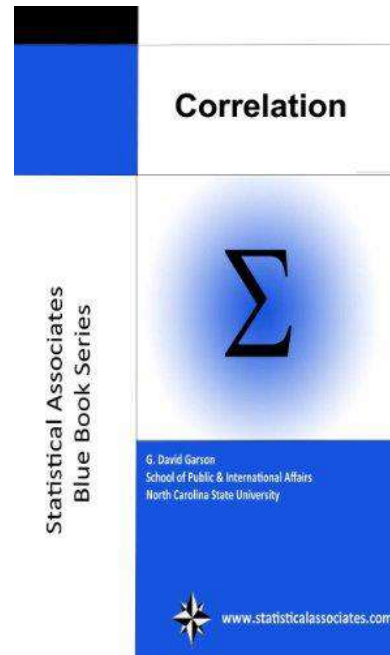
Dia 4: Séries não-lineares e teoria da informação

- reconstrução do espaço de fase
- expoentes de Lyapunov
- entropia de Shannon
- informação mútua, entropia de transferência, CaMI, direcionalidade



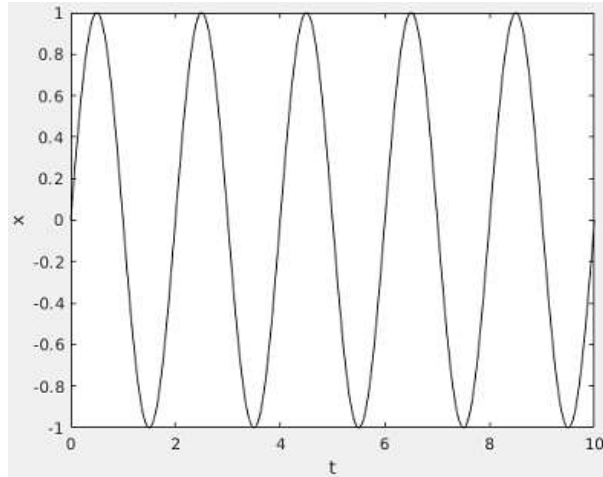


>>Quick refs



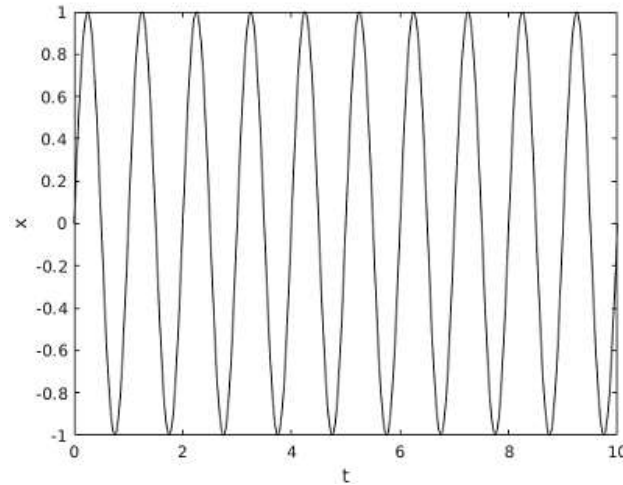


>>Análise espectral



$$x(t) = \text{sen}(\pi t)$$

1 ciclo a cada 2 segundos



$$x(t) = \text{sen}(2\pi t)$$

1 ciclo por segundo

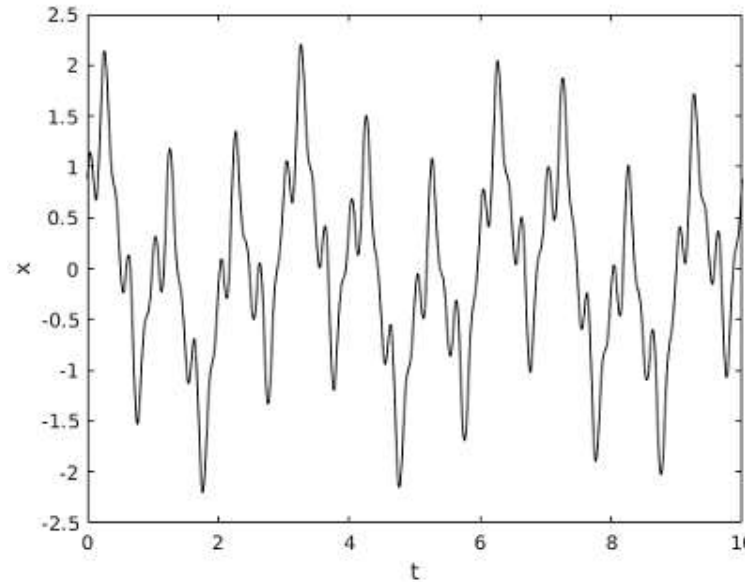
...

$$x(t) = \text{sen}(2\pi f t)$$

f ciclos por segundo



>>Análise espectral



Quais as frequências???

$$x(t) \rightarrow X(f)$$

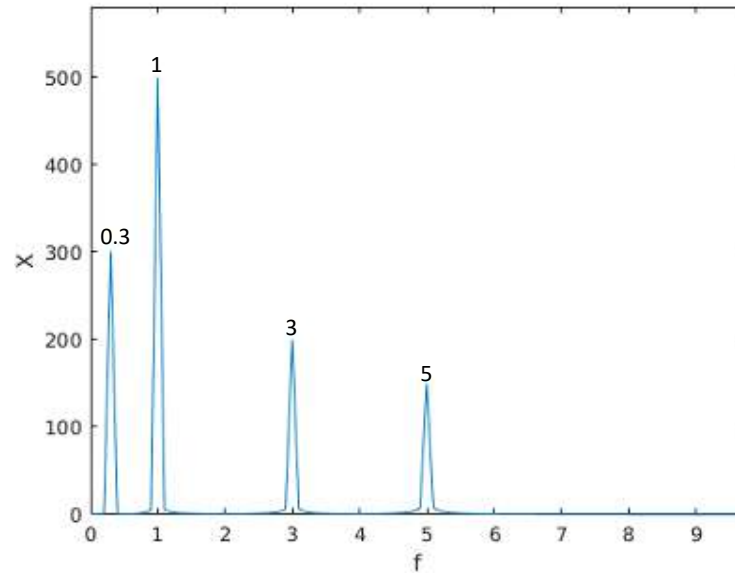
$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{2\pi i f t} dt$$

Vixe, é complexo...

A magnitude $|X(f)| = \sqrt{\text{Re}(X)^2 + \text{Im}(X)^2} = \sqrt{X X^*}$ apresenta as frequências



>>Análise espectral



Matlab:

```
X=fft(x);  
f=Fs*(0:length(x)/2)/length(x);  
plot(f,abs(X(1:length(f))));
```

Excel:

Carregue o Data Analysis ToolPak: Arquivo -> Opções -> Add-ins -> Ir -> Analysis ToolPak

Ir para aba Dados-> Análise de dados

Selecionar Análise de Fourier

Escolher o local de input e output dos dados

Atenção: só consegue analisar 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 ou 2048, ... pontos

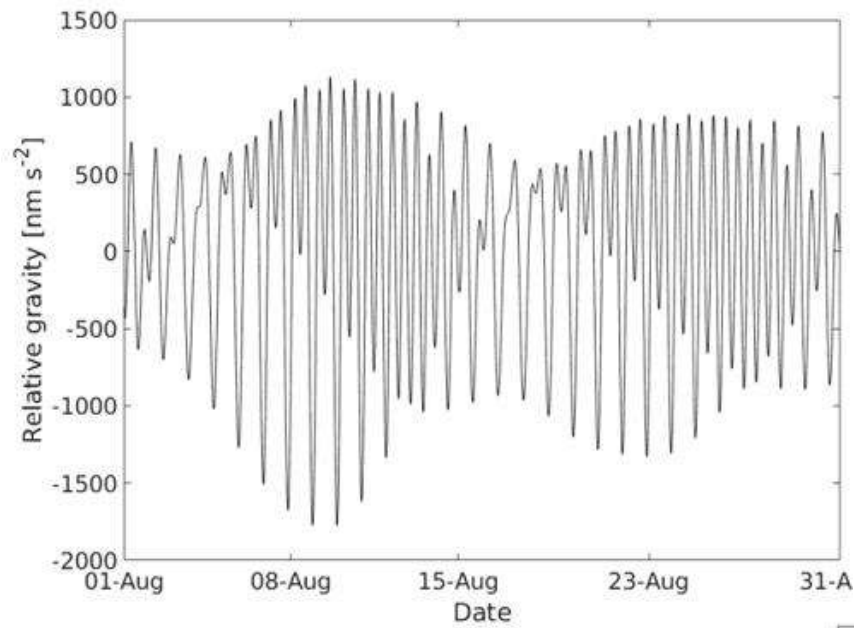
Calcular a magnitude (função IMABS)

Fazer coluna da frequência: $=Fs*(\text{linha}(x_i)-\text{linha}(\$x_\$0))/\text{contar}(\$x_\$0:\$x_\$fim)$

Inserir -> gráfico de linha ou dispersão etc.

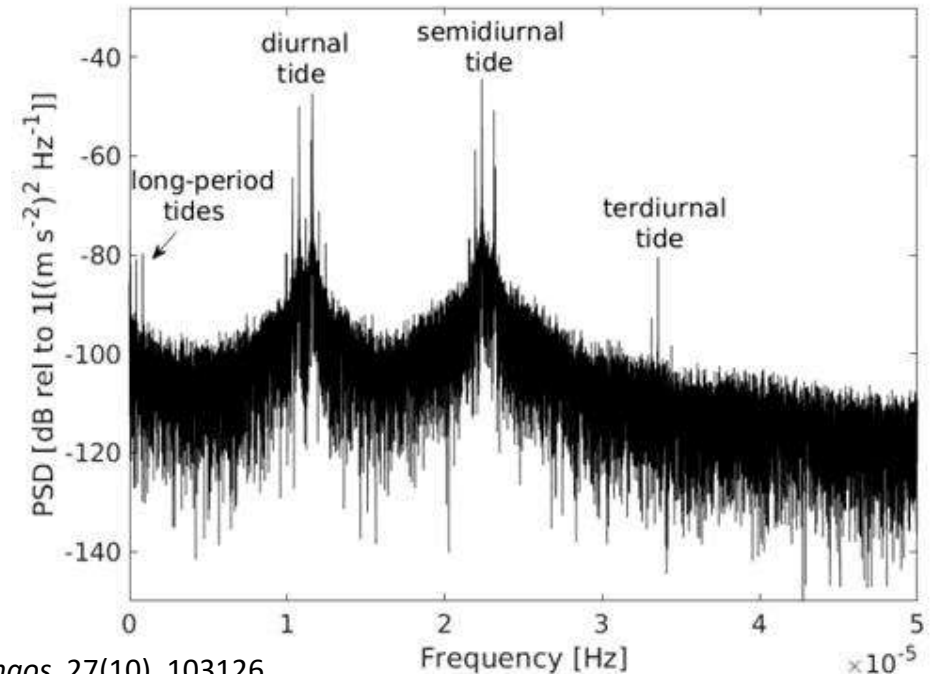


>>Análise spectral: aplicação



Variação da gravidade registrada em Sutherland, África do Sul com um sensor de alta precisão (gravímetro superconductor) em Agosto de 2010

Modos de maré identificados no sinal



Publicado em Valencio, Grebogi & Baptista (2017), *Chaos*, 27(10), 103126



>>Análise spectral: filtros

- Existe uma teoria complexa de filtros digitais que vai além do escopo deste curso.
- As formas de implementação variam de acordo com a linguagem de programação utilizada
- Procure por:
 - **FIR (Finite Impulse Response) filter**: mais confiáveis, porém mais custosos computacionalmente
 - **IIR (Infinite Impulse Response) filter**: menos confiáveis, porém mais rápidos

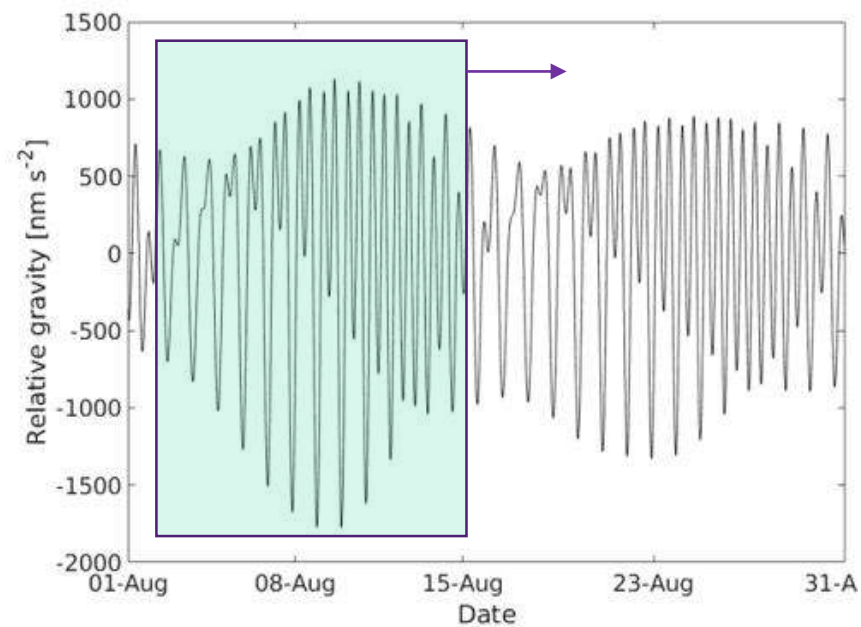
Tipos de filtragem:

- **Passa-baixa** (low frequency): corta as frequências acima de f_c → Remove variações lentas, ruídos de fundo
- **Passa-alta** (high-frequency): corta as frequências abaixo de f_c → Remove ruídos instrumentais e ambientais
- **Passa-banda** (bandpass): somente permite frequências entre f_{c1} e f_{c2} → Seleção de evento em medida muito sensível a outras variáveis
- **Rejeição/Corta-banda** (notch): remove frequências entre f_{c1} e f_{c2} → Remove efeitos específicos como ruído da rede elétrica (60Hz)

Publicado em Valencio, Grebogi & Baptista (2017), *Chaos*, 27(10), 103126



>>Short-time Fourier Transform



$$X(f, t)$$

Obs: pode-se escolher esta janela como retangular (mais fácil de implementar) ou de forma a dar maior peso ao centro e menor às bordas (Hamming, Gaussian, Kaiser, etc.)



>>Short-time Fourier Transform: aplicação

Problema:
Baixa resolução nas baixas frequências

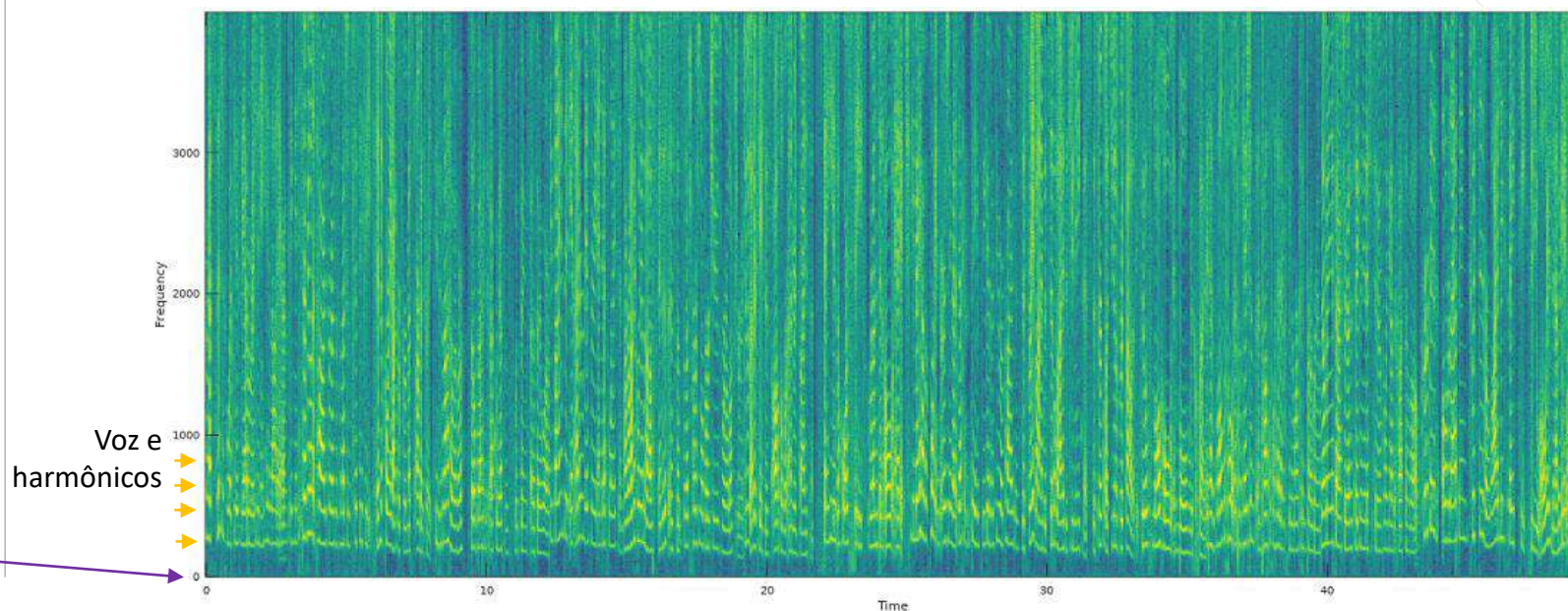
Matlab:

Basta usar a função `spectrogram`, mas requer `Signal Processing Toolbox` (adicional pago)

Octave:

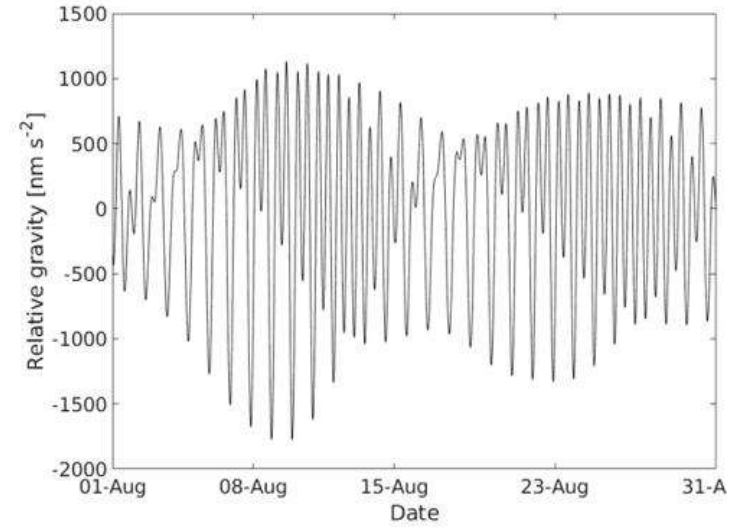
Usaremos a função `specgram`, do `package signal` (gratuito)

```
pkg load signal;  
[y,fs]=audioread(filename);  
window=2^nextpow2(ceil(100*fs/1000)); %janela da potência de 2 mais próxima de 100ms  
specgram(y>window,fs);
```



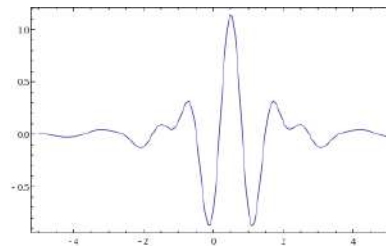


>>Wavelet (onduleta)

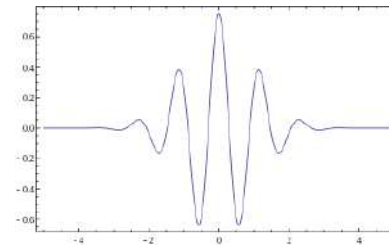


$$X = \int_{-\infty}^{+\infty} x(t) \psi \left(\frac{t - t_0}{a} \right) dt$$

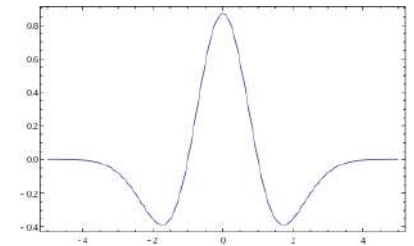
$X(f, t)$ → Composição de várias onduletas-mãe ψ



Meyer



Morlet



Sombrero





>>Wavelet (onduleta)

Matlab:

Basta usar a função `cwt`,
mas requer Wavelet Toolbox (adicional pago)

Python:

```
Bibliotecas { from scipy import signal
               import matplotlib.pyplot as plt
               import numpy as np
               import csv
Leitura dos dados { with open('Tohoku.csv') as csv_file:
                    csv_reader=csv.reader(csv_file,delimiter=',')
                    for row in csv_reader:
                        t.append(float(row[0]))
                        x.append(float(row[1]))
                    widths=np.arange(1,31)
Wavelet { cwtmatrix=signal.cwt(x,signal.ricker,widths)
          plt.imshow(cwtmatrix,cmap='binary',aspect='auto',vmin=abs(cwtmatrix).min)
          plt.show()
Terremoto (p/ comparação) { plt.plot(x)
                           plt.show()
```

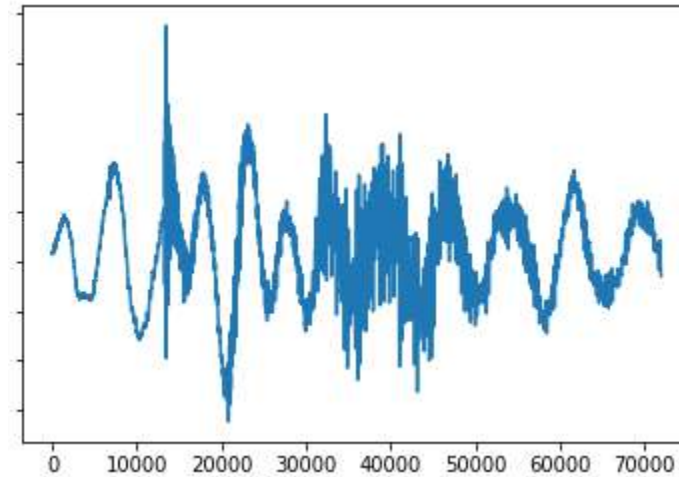
Onduletas “sombbrero”



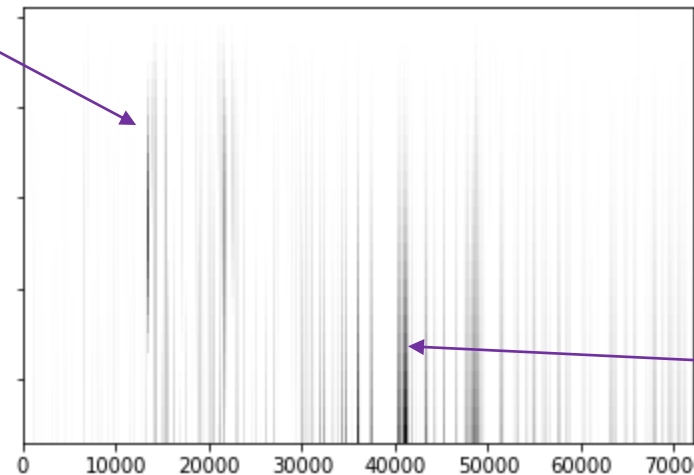
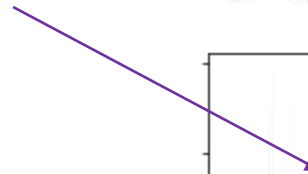
>>Wavelet (onduleta): aplicação

Terremoto de Tohoku-Oki (Fukushima), 2011

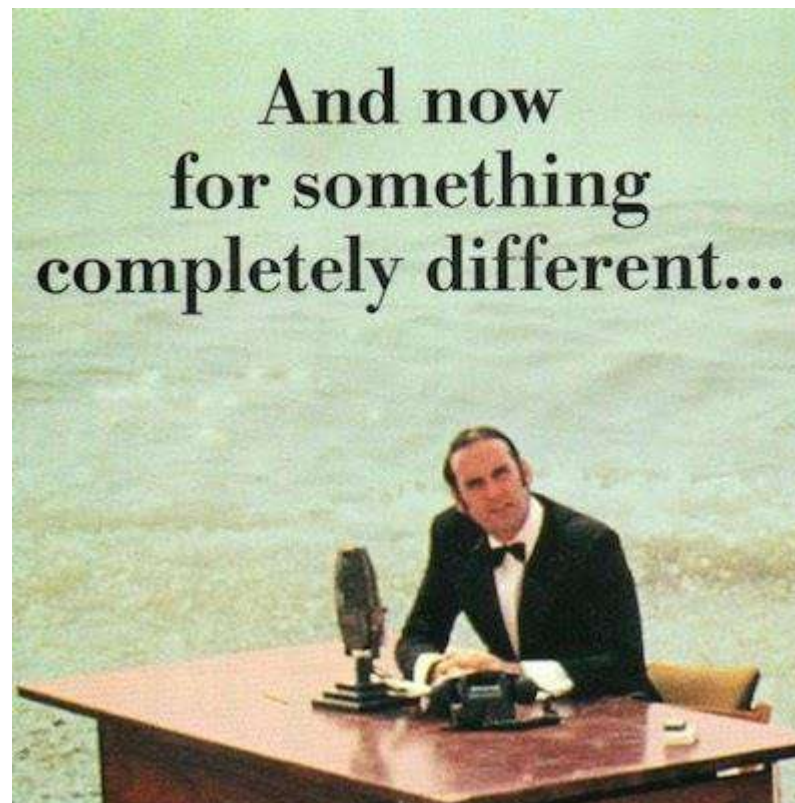
Dados da estação sismológica 'WMQ', em Ürümqi (Xinjiang), China



Começam
com a
chegada do
terremoto



Chegada das
ondas de
superfície,
com mais
energia





>>Correlação

Idéia: mensurar o quanto duas variáveis estão estatisticamente associadas ou

Em termos bem leigos: o quanto duas variáveis se “comportam parecidas”

Karl Pearson:

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$



$$r_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

+1 Correlação perfeita

0 Sem correlação: inconclusivo

-1 Anti-correlação perfeita

X e Y independentes $\rightarrow r=0$

mas

$r=0$ **não** significa que X,Y são independentes (ex: $Y = X^2$)



>>Correlação

>> Quão forte precisa ser a sua correlação?

Depende da área!

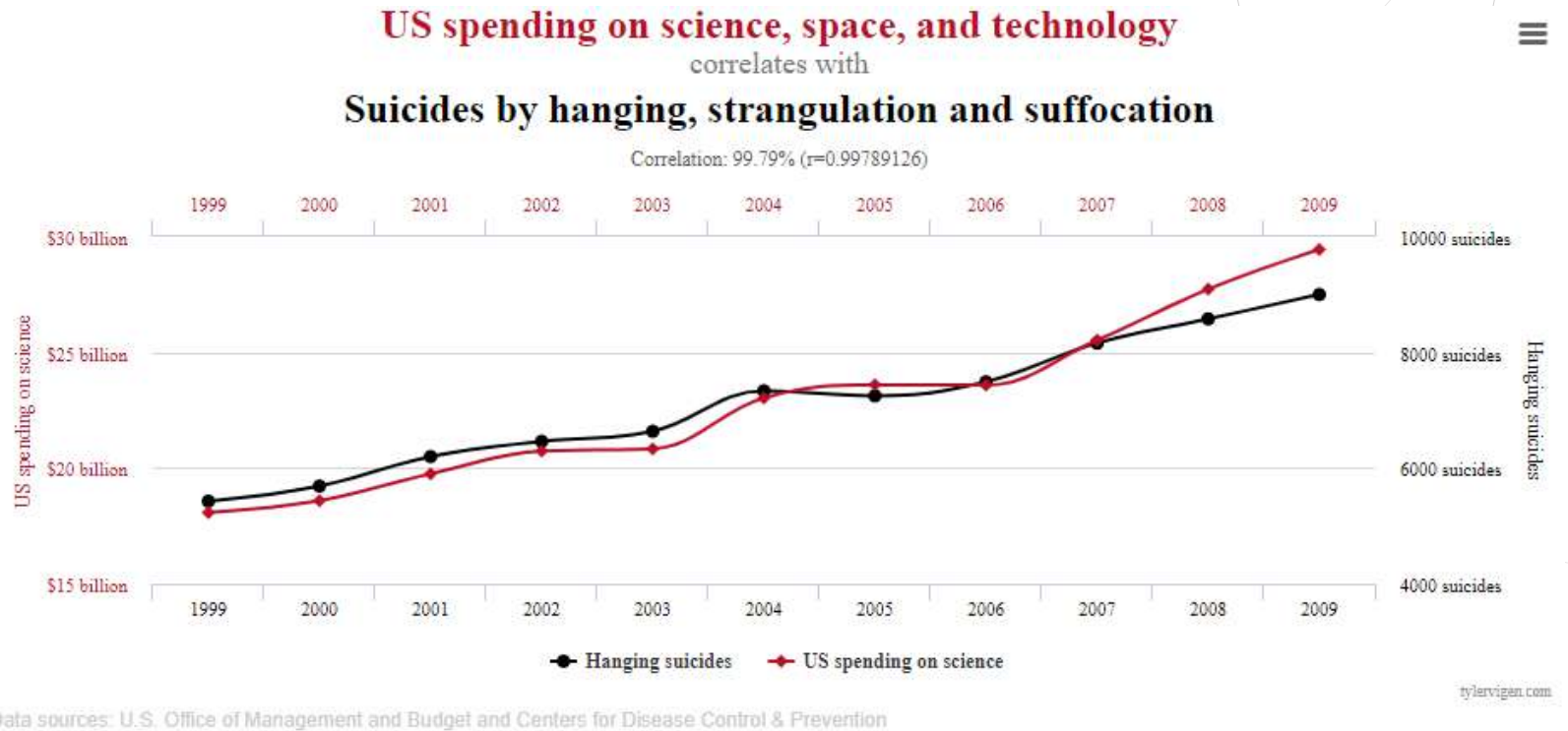
$ r $	Interpretação	Comum nas áreas
0,3	Associação não pode ser descartada	Ciências Humanas, Ciências da Saúde (Psicologia)
0,5	Provável associação (procurar por 3os fatores associados)	Ciências da Saúde (epidemiologia, nutrição)
0,7-0,75	Há correlação	Ciências da Saúde, Ciências Humanas e Engenharias
0,9	Forte correlação	Ciências da Saúde, Ciências Humanas, Engenharias e Ciências Exatas
0,95-0,99	Correlação muito forte	Parecerista #3



>>Correlação

≠

Causalidade



Correlação de 0.99789126!!

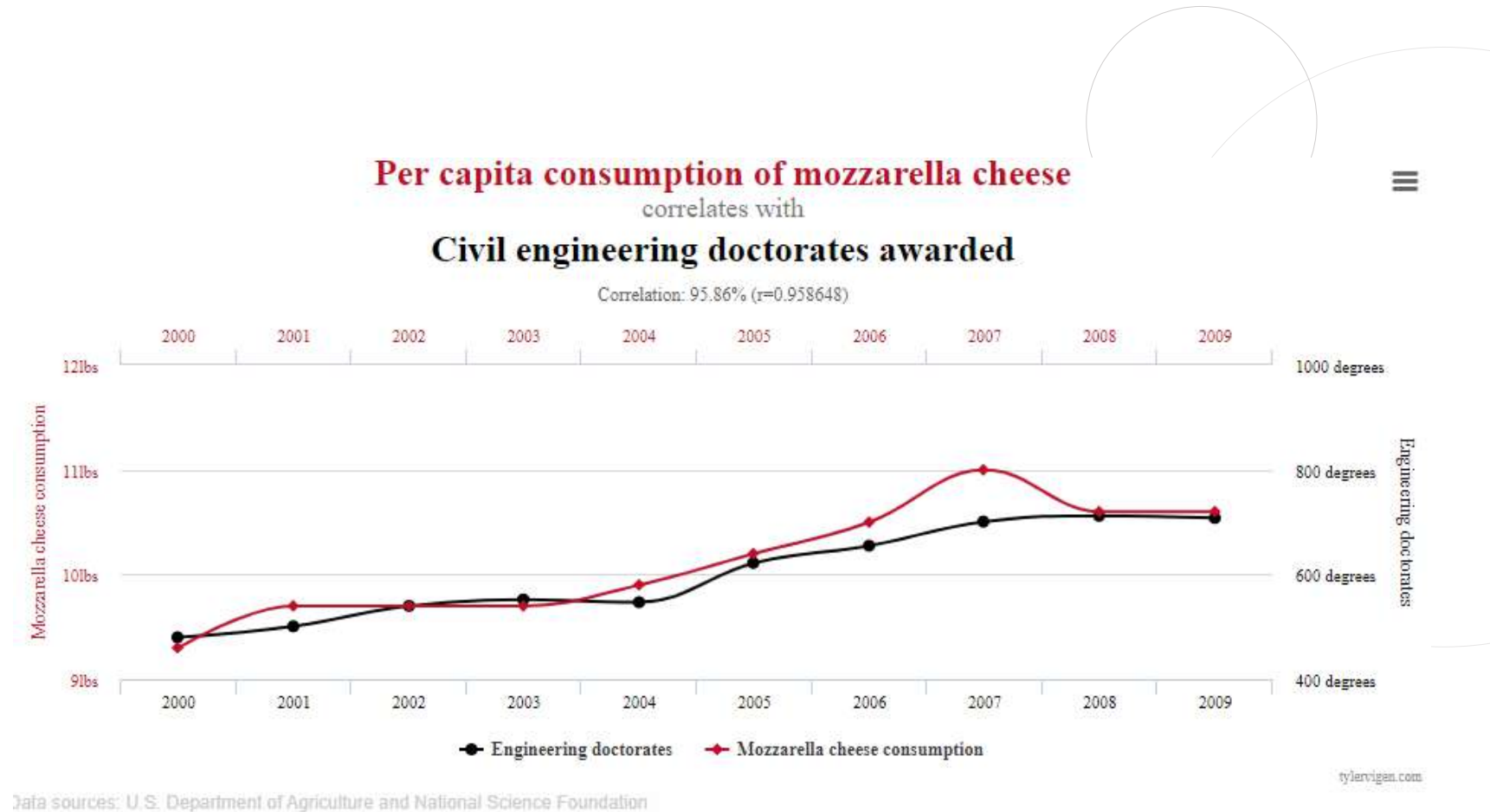
Vigen, T. (2015). Spurious correlations. New York: Hachette books.



>>Correlação

≠

Causalidade



Correlação de 0.958648!!

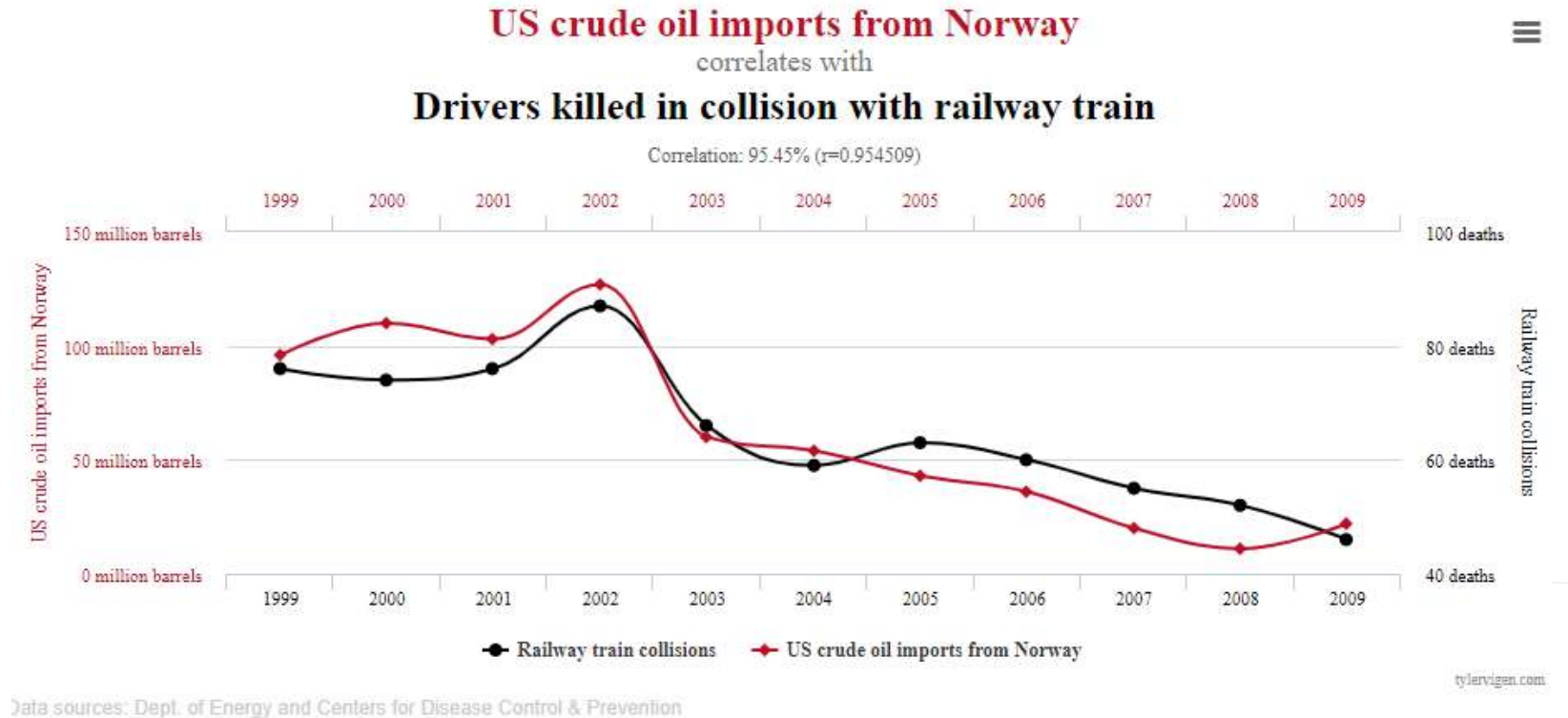
Vigen, T. (2015). Spurious correlations. New York: Hachette books.



>>Correlação

≠

Causalidade



Correlação de 0.954509!!

Vigen, T. (2015). Spurious correlations. New York: Hachette books.



>>Correlação: implementação

Matlab:

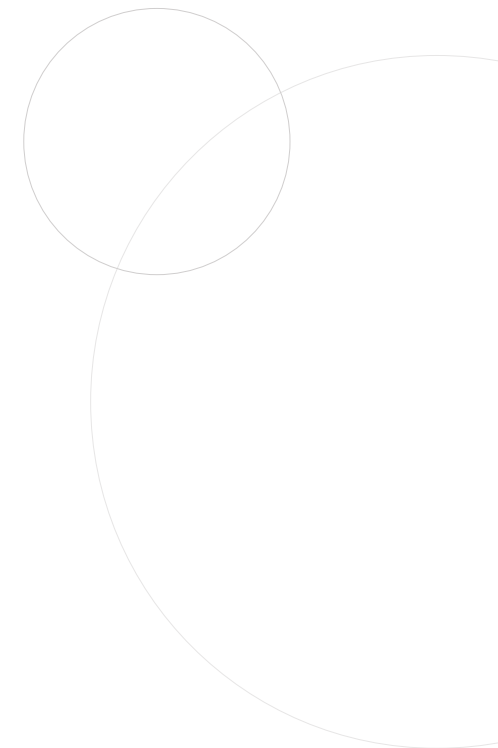
```
r=corrcoef(x,y)[1,2];
```

Python:

```
import numpy as np  
r=np.corrcoef(x,y)[0,1]
```

Excel:

```
=correl(x_0:x_fim,y_0:y_fim)
```





>>Correlação: implementação

Mas... correlação de Pearson tem dificuldades, por exemplo, quando x tem variações muito grandes (ou outliers) comparadas à y (ou vice-versa)

Proposta:

$$\text{Spearman: } \rho = \frac{\sum(r_{x_i} - r_{\bar{x}})(r_{y_i} - r_{\bar{y}})}{\sqrt{\sum(r_{x_i} - r_{\bar{x}})^2 \sum(r_{y_i} - r_{\bar{y}})^2}}$$

Excel/Matlab/Python: basta ver o posto (rank, função `ordem` do Excel) as variáveis antes de calcular o coeficiente

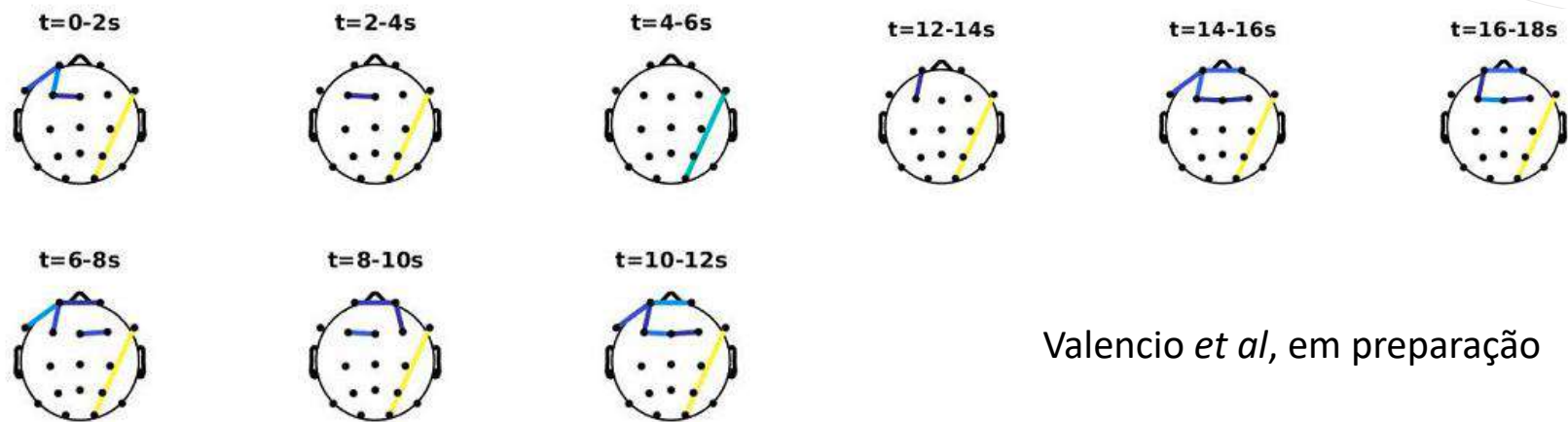
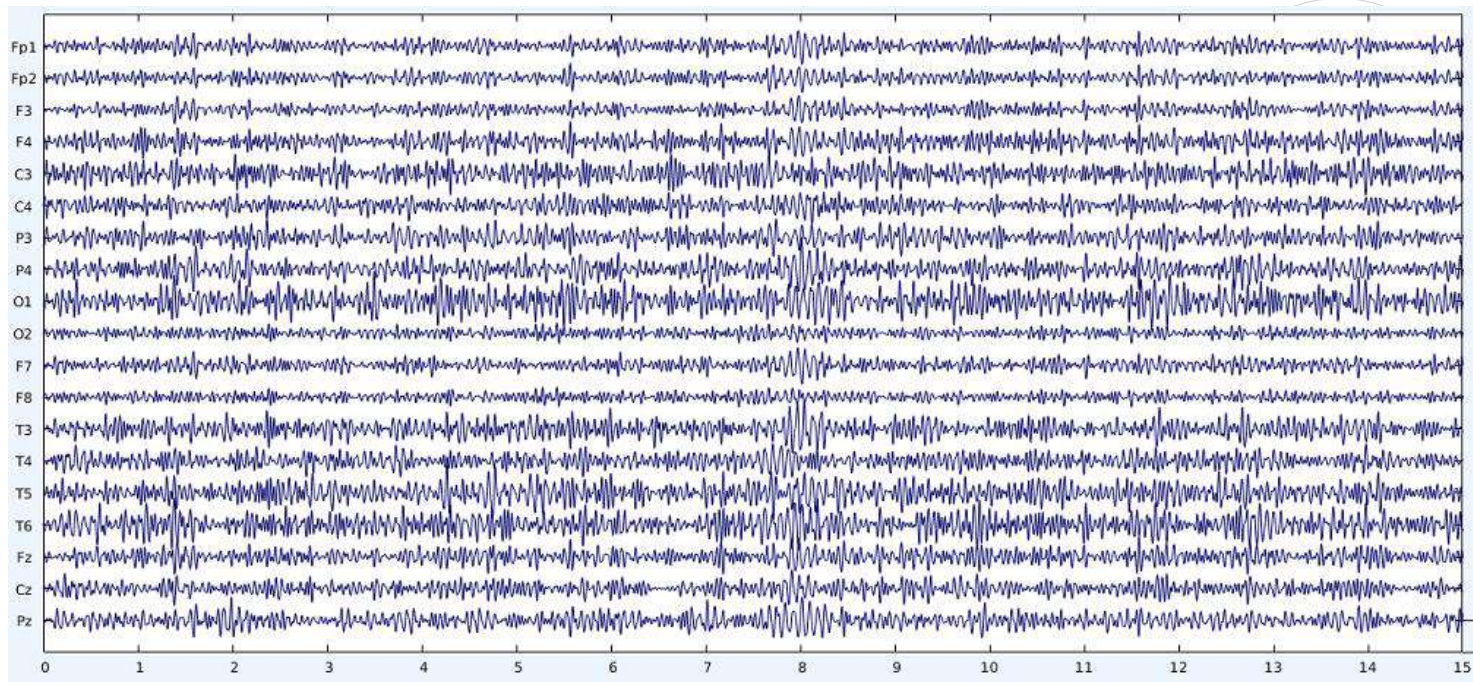
Alternativamente, pode-se usar a função `corr` com opção `'spearman'` do Statistics & Machine Learning Toolbox do Matlab

ou

A função `spearman` do statistics package do Octave



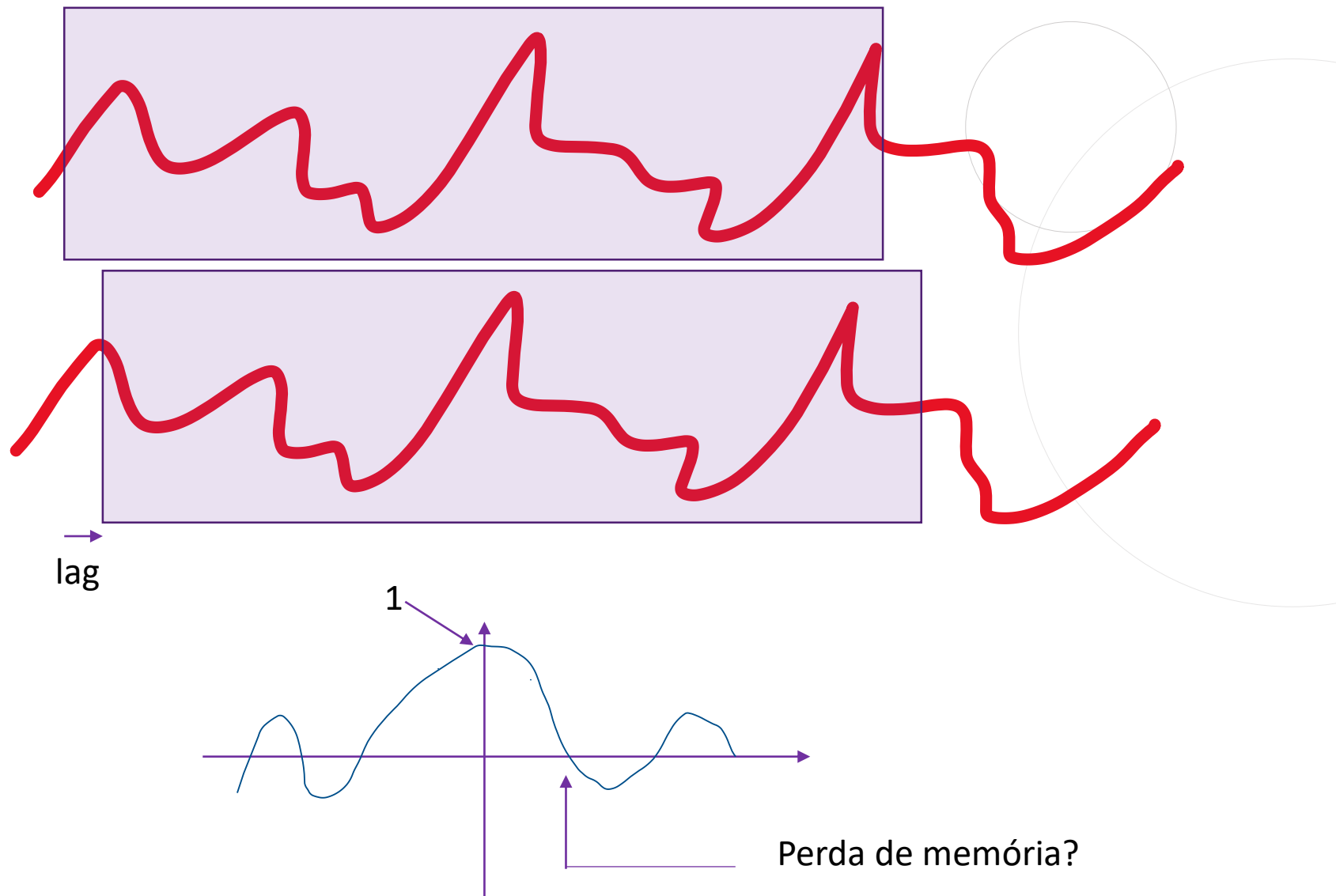
>>Correlação: aplicação



Valencio *et al*, em preparação



>> Autocorrelação e correlação cruzada





>>Regressão

Idéia: Propor um modelo, baseado em ajuste aos dados, de como as variáveis se relacionam

Espera-se:

1. que a partir do modelo possa-se prever novos resultados
2. que o modelo elucidie o fenômeno investigado

Ajustes convencionais:

- **Reta (regressão linear)**
- Polinomial
- Logística → Casos '0' e '1', p.ex.: doente vs saudável
- Exponencial
- Logarítmo

Lembre-se da navalha de Occam!





>>Regressão

Coeficiente linear (intercept)

$$y = ax + b + \varepsilon$$

Coeficiente angular (slope)
ou "taxa de variação"

Erro (pequeno, média zero, dist normal) do modelo.

Mínimos quadrados:

$$b = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$$a = \bar{y} - b\bar{x}$$

Coeficiente de determinação

("o quanto da variação de y é determinada pela variação de x")

$$R^2 = 1 - \frac{\sum(y_i - (ax_i + b))^2}{\sum(y_i - \bar{y})^2}$$



>>Regressão

Excel:

1. Pode-se fazer no próprio gráfico de dispersão: clique com o botão direito do mouse em um dos pontos e selecione “adicionar linha de tendência” nas opções, selecione “mostrar equação no gráfico” e “mostrar valor de R quadrado” no gráfico
2. Alternativamente pode-se utilizar as funções **inclinação** e **interceptação**
3. Pode-se também utilizar a formula de matriz **proj.lin**
4. Outra opção é usar o Data Analysis ToolPak

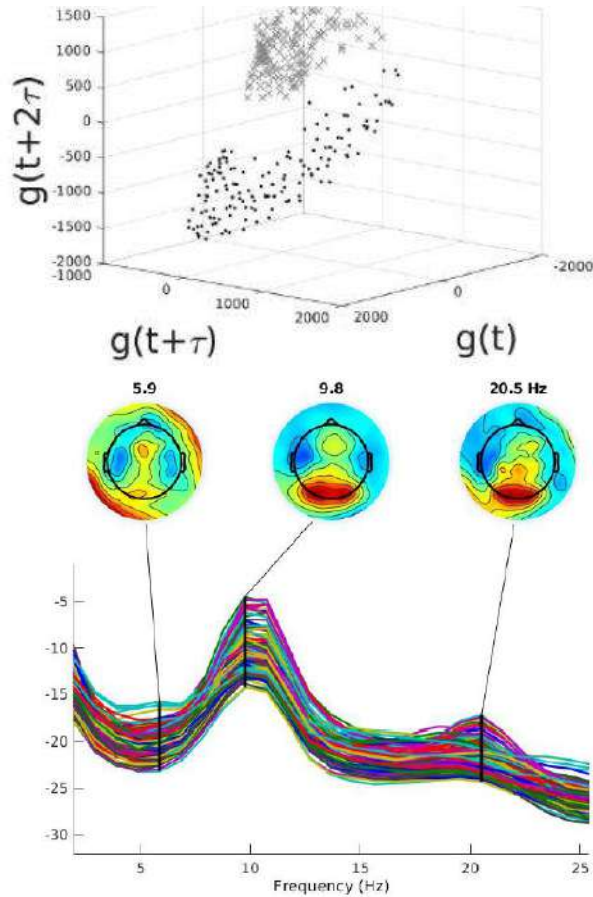
Matlab:

```
p=polyfit(x,y,1)
```

O valor de p terá ambos os coeficientes. Trocando 1 por n , faz-se regressão polinomio de ordem n : $y = a_1x + a_2x^2 + \dots + a_nx^n + b + \varepsilon$

Python:

```
import numpy as np
from sklearn.linear_model import LinearRegression
x=...
y=...
modelo=LinearRegrssion().fit(x,y)
R_square=modelo.score(x,y)
print('a:',modelo.coef_)
print('b:',modelo.intercept_)
```



Instituto de
Computação



NeuroMat

● ● ● ●

Obrigado!



arthur_valencio@physics.org



<http://www.arthurvalencio.com>
<http://neuromat.numec.prp.usp.br>

A.V. agradece à FAPESP por uma bolsa de pós-doutoramento (#2018/09900-8).

CEPID NeuroMat agradece apoio FAPESP (#2013/07699-0).

“As opiniões, hipóteses e conclusões ou recomendações expressas neste material são responsabilidade do(s) autor(es) e não necessariamente refletem a visão da Fapesp”