



2020

ANÁLISE ESTATÍSTICA E ESPECTRAL DE PROCESSOS ESTOCÁSTICOS

Uma Abordagem Computacional

reinaldo.rosa@inpe.br

MÓDULO B1: 22/24/27/29 - Abril

- 1. Eventos, Variáveis Aleatórias & PE
- 2. Espaço de Probabilidades
- 3. Fórmula de Bayes
- 4. Histogramas & PDF's
- 5. Momentos & Estacionaridade
- 6. Teoremas dos Limites Amostrais
- 7. Espaço de Cullen-Frey
- 8. Processos Estocásticos – p model
(eventos extremos)

MÓDULO B2: 04/06/11/13/18/20 - Maio

- 1. Função de Autocorrelação
- 2. Transformações para o Domínio Espectral
- 3. Transformada de Fourier
- 4. Transformada Rápida de Fourier
- 5. Densidade Espectral de Potências
- 6. Teorema de Wiener-Khinchin
- 7. Teorema de Parceval e Exponente de Holder
- 8. Espectro DFA
- 9. Modelos Multifractais
- 10. Espectro de Singularidades (MFdfa)

K-MÉDIAS P/ AGRUPAMENTO



CLIENTES DE UMA LOJA:
TOTAL DE 12 OBSERVAÇÕES

✓ Satisfação (S) (-1 a 1)

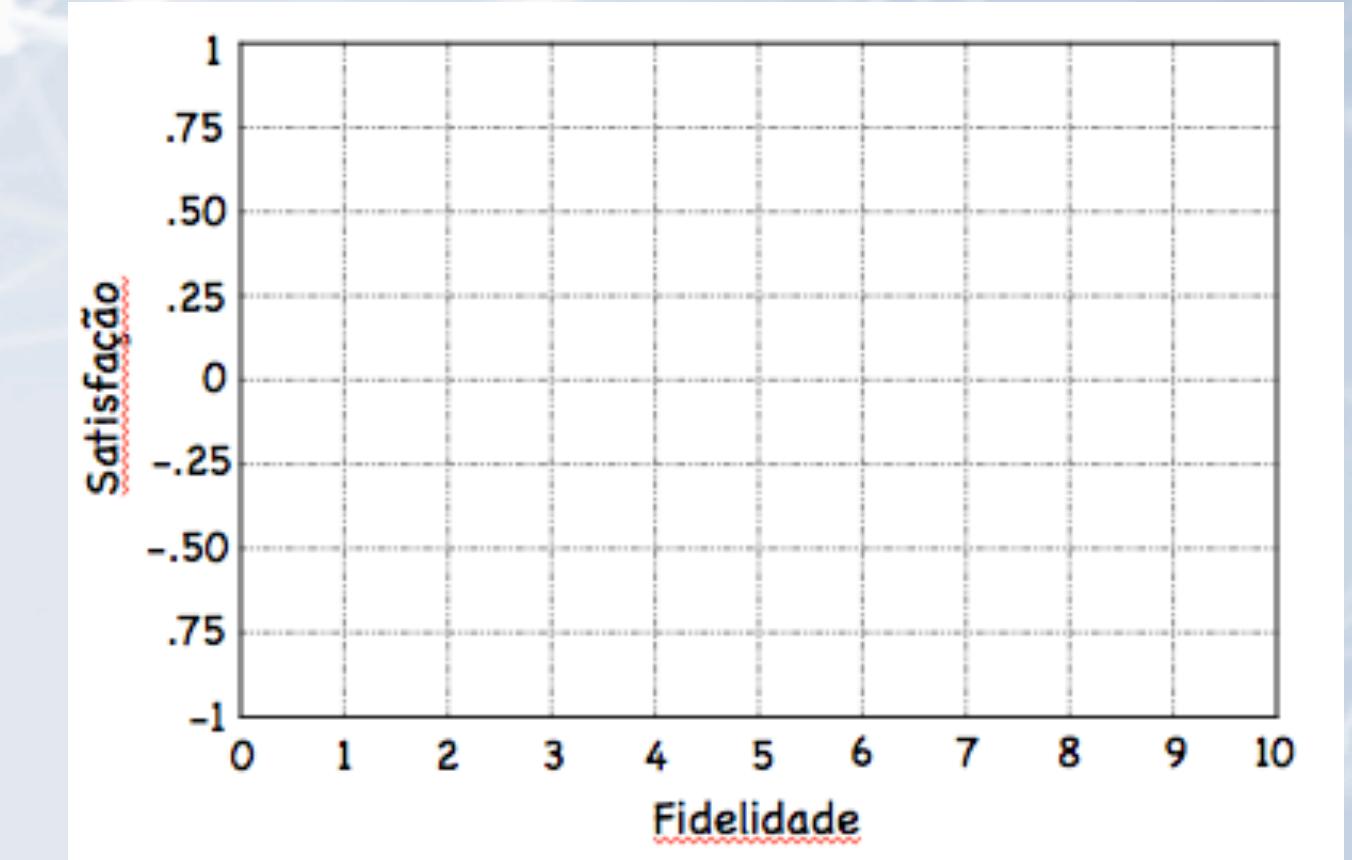
✓ Fidelidade (F) (1 a 10)



C	F	S
1	6	0.50
2	4	0.25
3	2	0.50
4	3	0.75
5	9	-0.25
6	9	0.50
7	8	0.75
8	7	-0.25
9	8	-0.50
10	2	-0.25
11	2	-0.75
12	1	-0.50

dados.csv

F, S
6, 0.50
4, 0.25
2, 0.50
3, 0.75
9, -0.25
9, 0.50
8, 0.75
7, -0.25
8, -0.50
2, -0.25
2, -0.75
1, -0.50

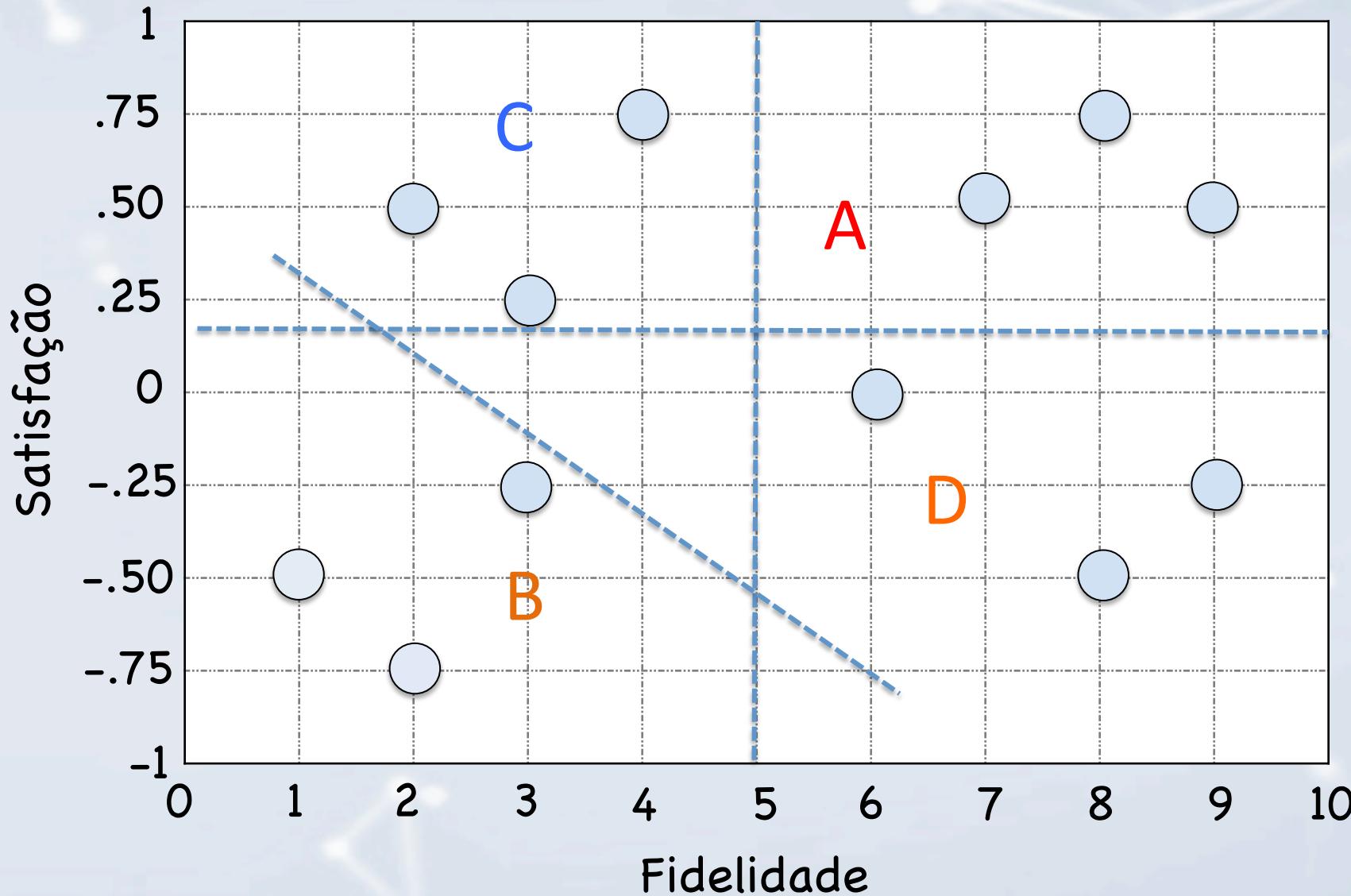


```

1 import csv
2 with open('dados.csv') as csv_file:
3     csv_reader = csv.reader(csv_file)
4     cabecalho = True
5     for row in csv_reader:
6         if cabecalho:
7             print(f'Nomes das colunas: {" ".join(row)}')
8             cabecalho = False
9         else:
10            print(f'{" ".join(row)})')

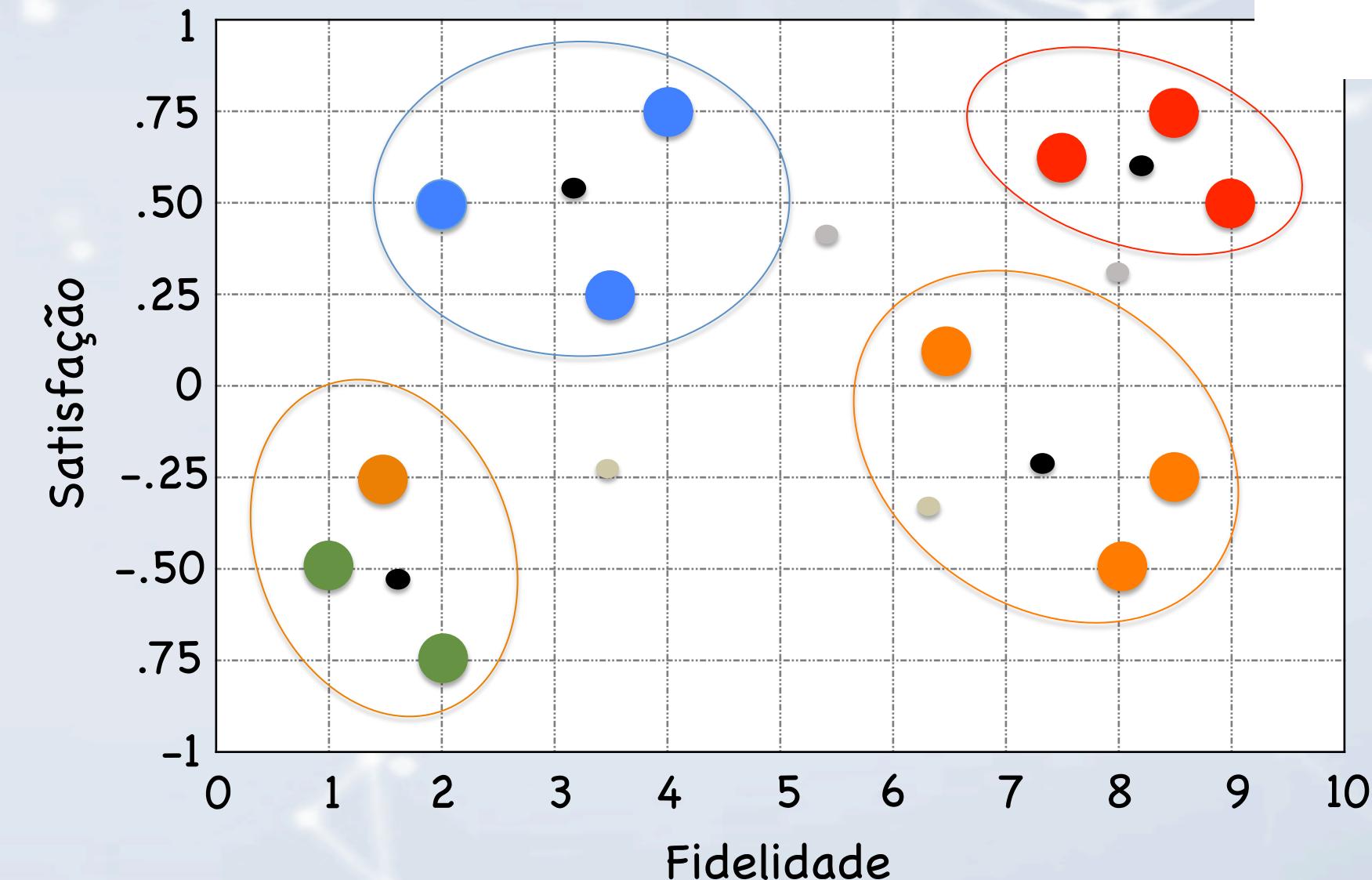
```

F, S
6, 0.00
4, 0.75
2, 0.50
3, 0.25
9, -0.25
8, 0.50
7, 0.50
9, 0.50
8, -0.50
3, -0.25
2, -0.75
1, -0.50



A: Fãs (ouro)
B: Aliens
C: Itinerantes
D: Apoiadores

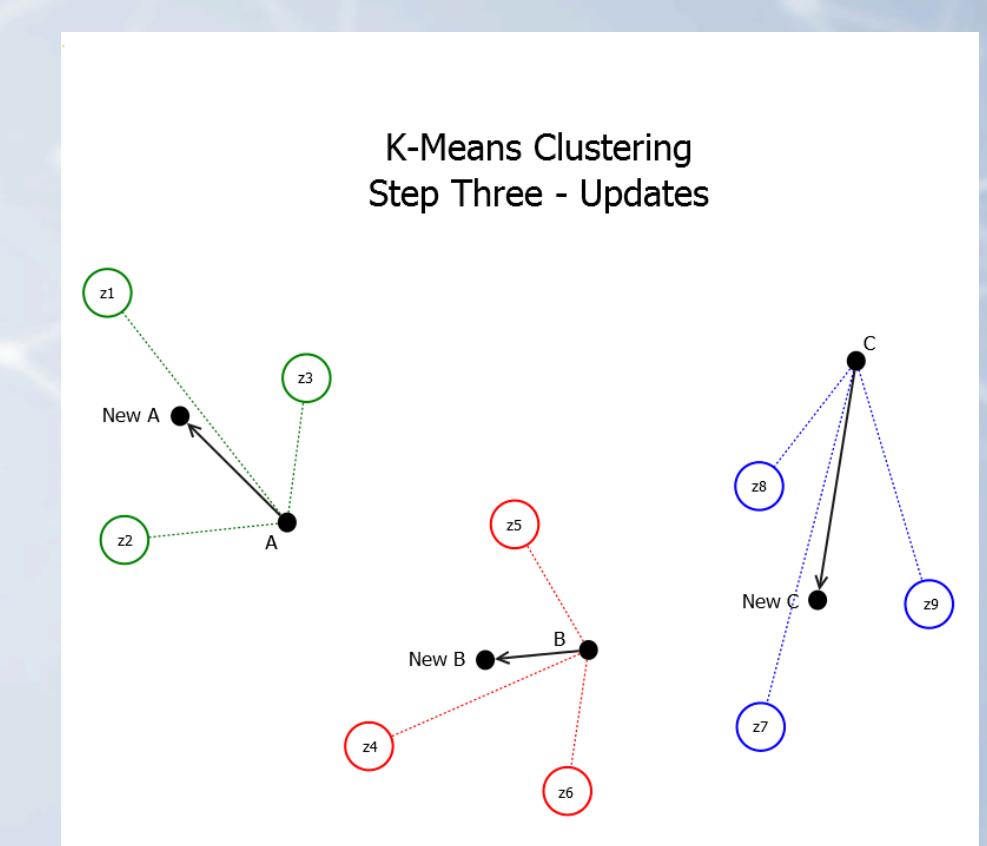
A: Alienados
 B: Fãs (ouro)
 C: Itinerantes
 D: Apoiadores
 $k=4$



K-Means

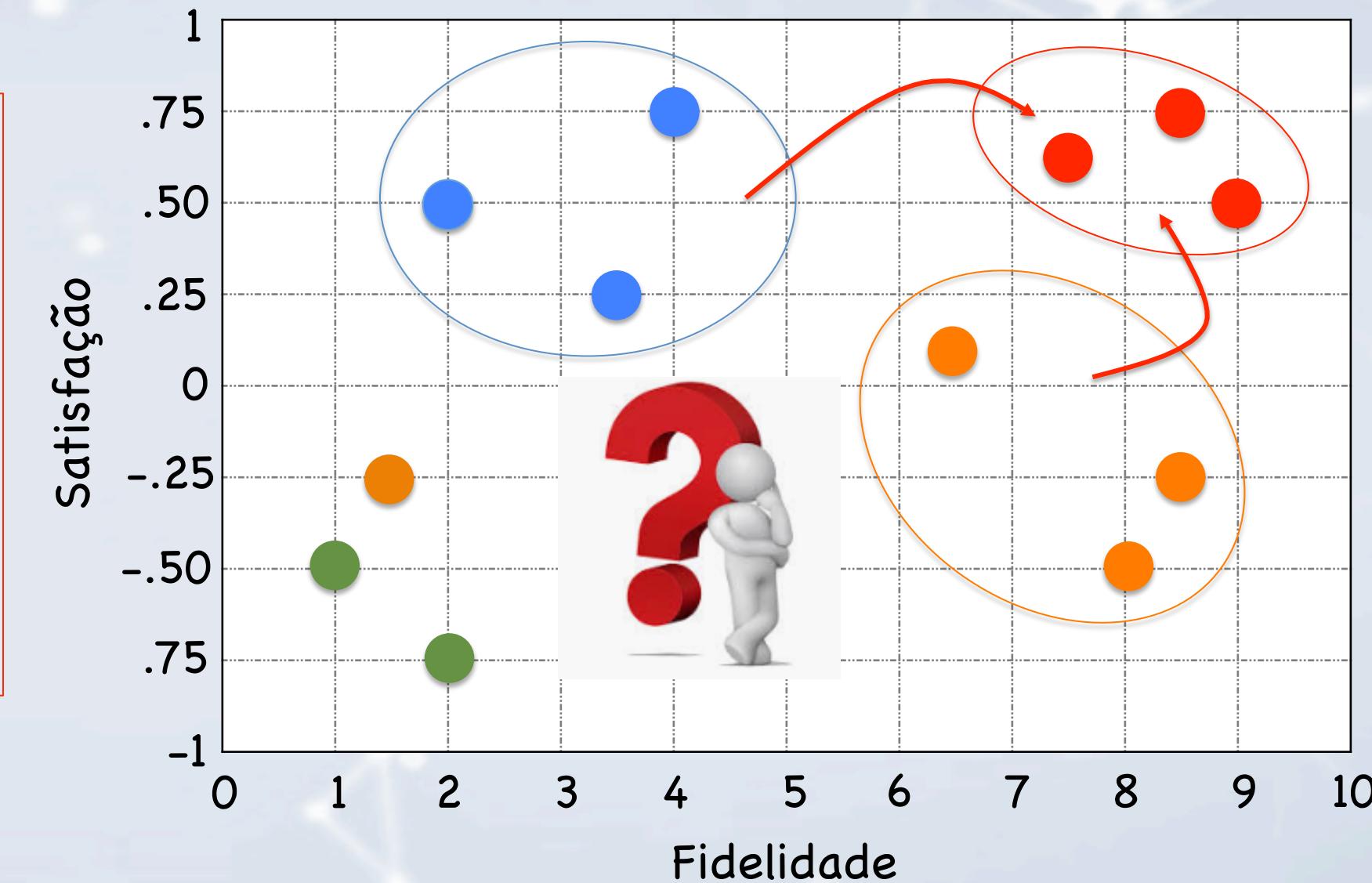
objective function $\leftarrow J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$

number of clusters k
 number of cases n
 case i
 centroid for cluster j
 Distance function



K-means é uma técnica (algoritmo) de *Mineração de Dados* (Data Mining) dentro do paradigma de Machine Learning NS

A: Fãs (ouro)
B: Aliens
C: Itinerantes
D: Apoiadores



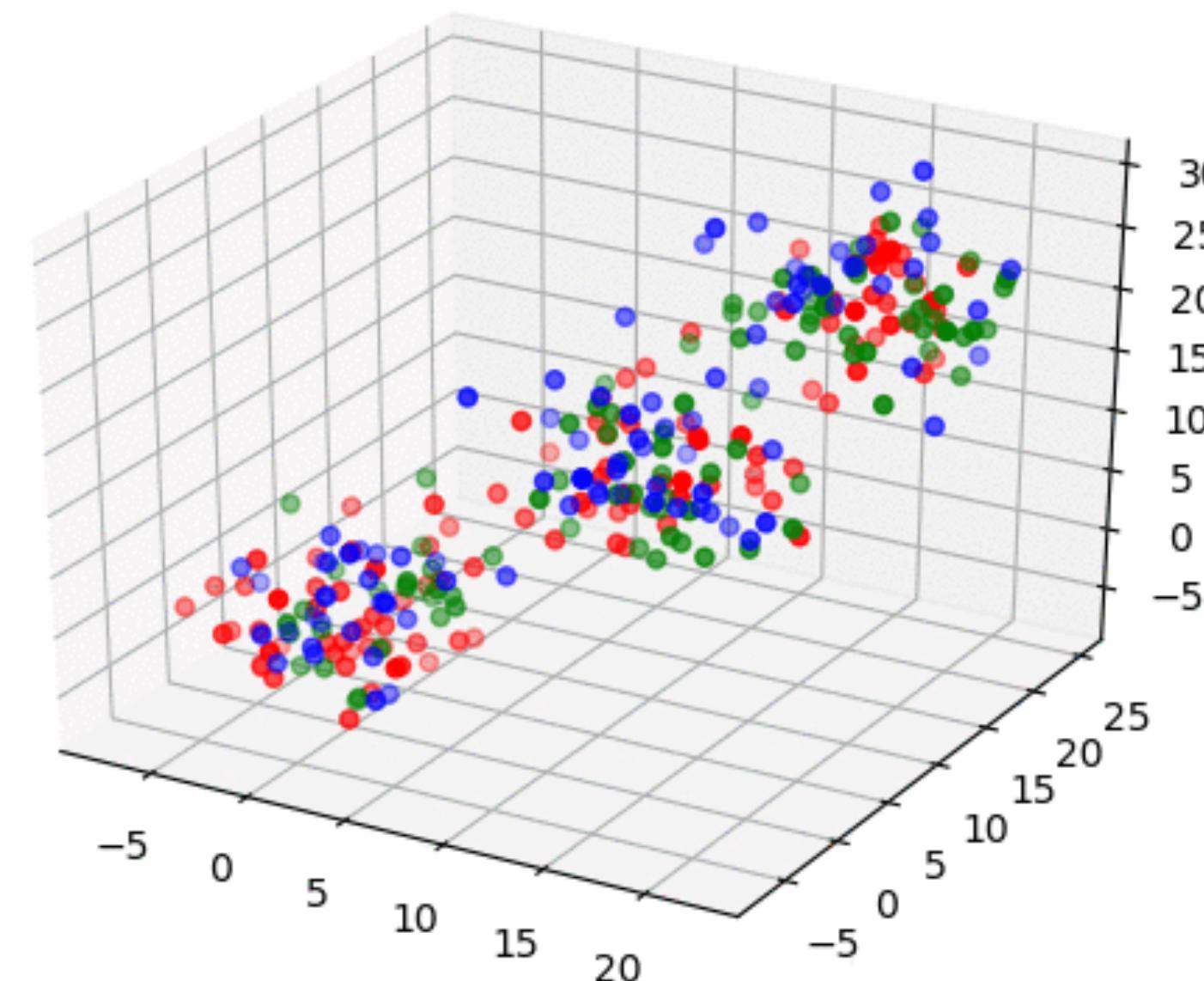
World Journal of Fish and Marine Sciences 5 (3): 266-270, 2013
 ISSN 2078-4589
 © IDOSI Publications, 2013
 DOI: 10.5829/idosi.wjfms.2013.05.03.66208

Study of the Activities of Digestive Enzymes, Amylase and Alkaline Phosphatase, in Kutum Larvae, *Rutilusfrisikutum* Fed Artemia Nauplii

¹F. Hassanatabar, ¹H. Ouraji, ¹A. Esmaeili and ²S.S. Babaei

¹Sari Agricultural Sciences and Natural Resources University (SANRU), Sari, Iran

²Fisheries Department, Faculty of Natural Resources and Marine Sciences,
 Tarbiat Modares University, Mazandaran, Iran



Digestive Enzymes

Enzyme Group	Enzyme	Substrate	Products
Amylases	Salivary amylase	Starch	Maltose
Lipases	Lipase	Fats	Fatty acid and glycerol
Proteases	Pepsin	Protein	Peptides

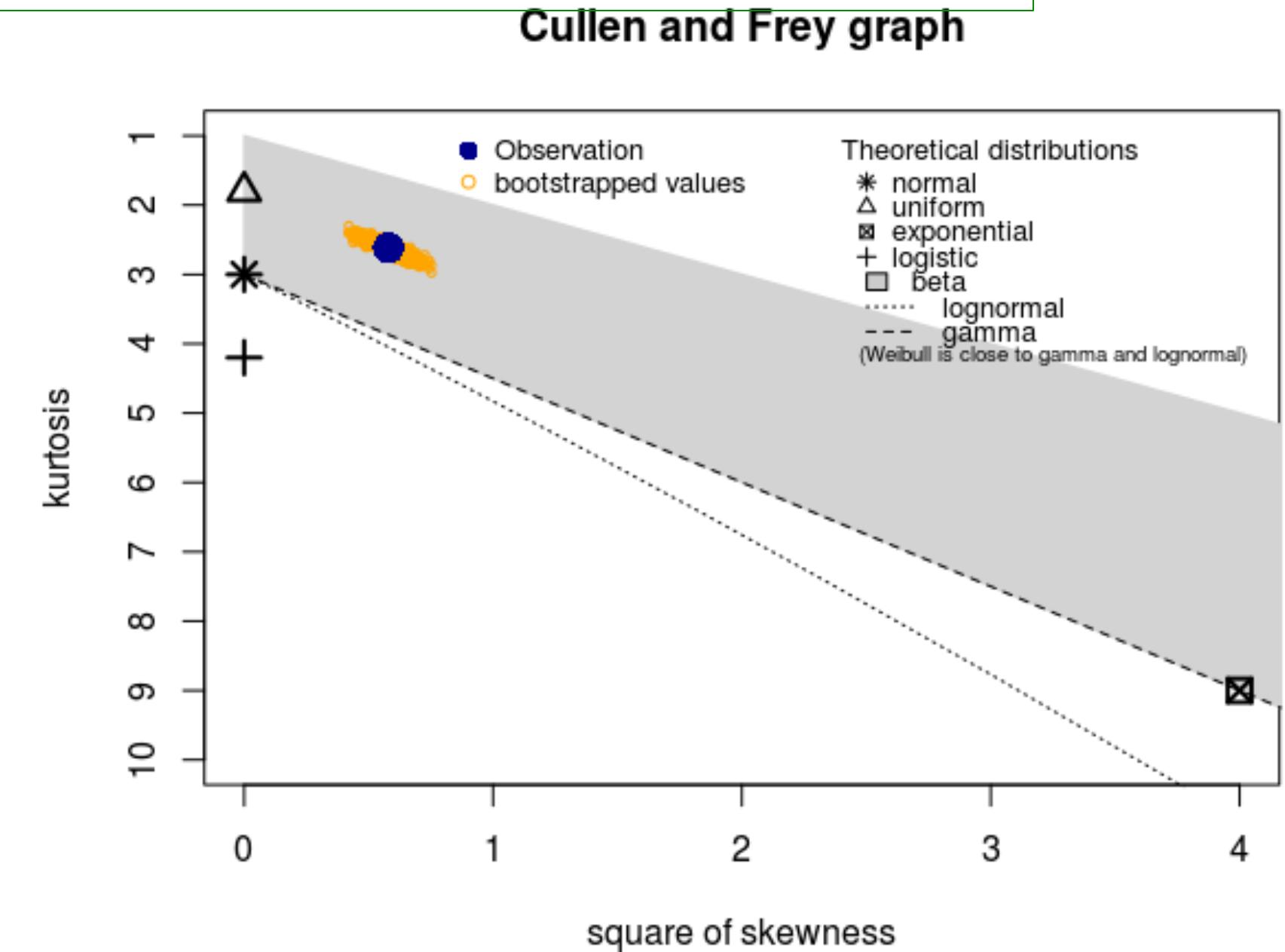
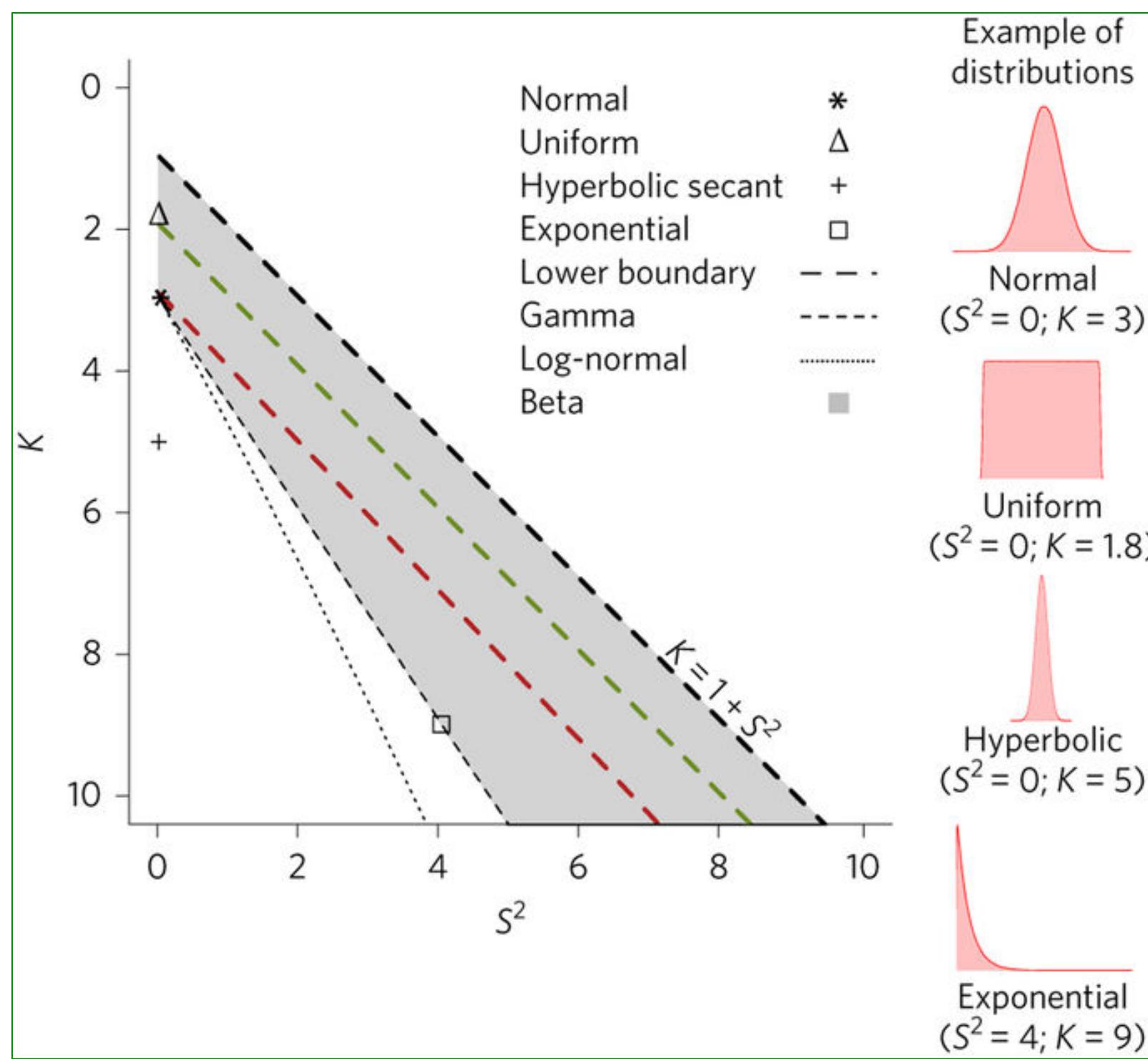
K-means é Aprendizagem de Máquina (ML)
 e não utiliza nenhum recurso de Neurocomputação.

Mas o fato de ser ML pode ser interpretada
 como uma ferramenta de IA?



7-ESPAÇO DE CULLEN-FREY (ECF): composto pelos valores de curtose e assimetria ao quadrado de várias funções de distribuições de probabilidades

Típicas cuja variação de parâmetros de cada uma determina sua localização neste espaço (gráfico). A variação fina dos parâmetros permite Determinar ainda sub-partições que ficam delimitadas dentro de um espaço restrito e podem ser usadas, em alguns casos, para caracterizar famílias de processos estocásticos. Na prática, é muito útil para encontrar qual o modelo estatístico de PDF a ser ajustado ao histograma das séries que definem uma família de variáveis aleatórias.





2020

ANÁLISE ESTATÍSTICA E ESPECTRAL DE PROCESSOS ESTOCÁSTICOS

Uma Abordagem Computacional

reinaldo.rosa@inpe.br

MÓDULO B1: 22/24/27/29 - Abril

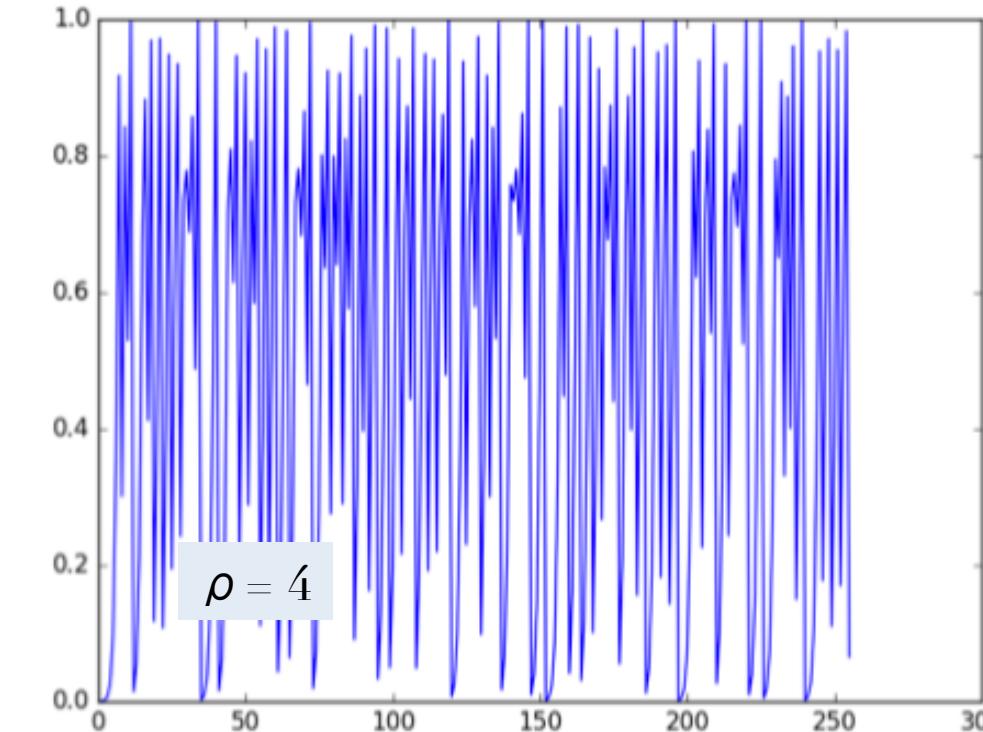
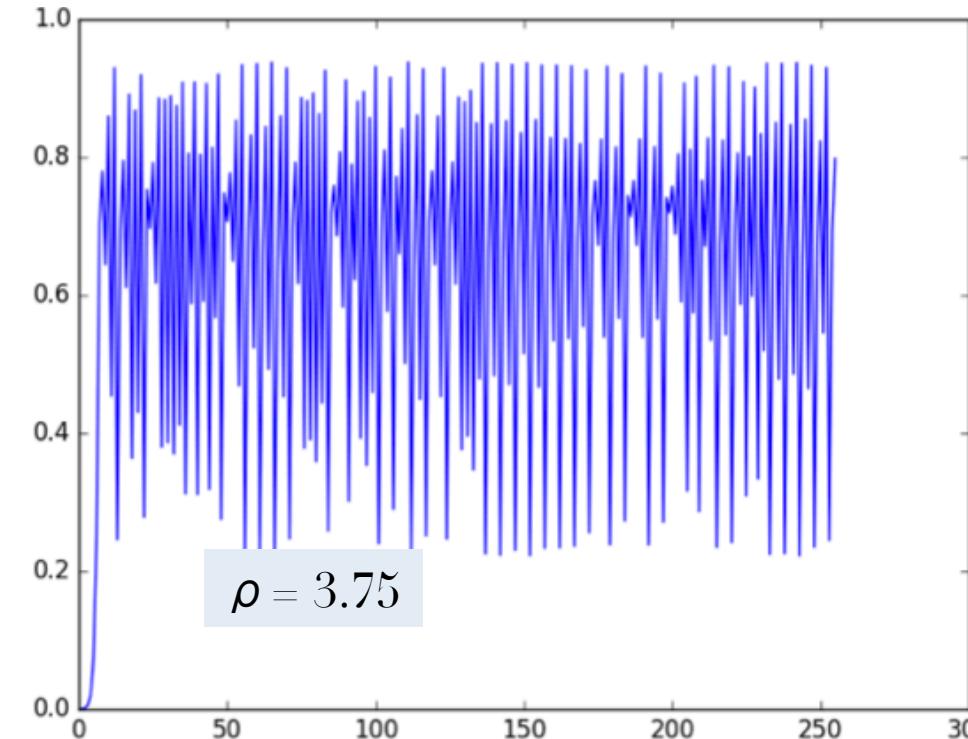
- 1. Eventos, Variáveis Aleatórias & PE
- 2. Espaço de Probabilidades
- 3. Fórmula de Bayes
- 4. Histogramas & PDF's
- 5. Momentos & Estacionaridade
- 6. Teoremas dos Limites Amostrais
- 7. Espaço de Cullen-Frey
- 8. Processos Estocásticos – p model
(eventos extremos)

MÓDULO B2: 04/06/11/13/18/20 - Maio

- 1. Função de Autocorrelação
- 2. Transformações para o Domínio Espectral
- 3. Transformada de Fourier
- 4. Transformada Rápida de Fourier
- 5. Densidade Espectral de Potências
- 6. Teorema de Wiener-Khinchin
- 7. Teorema de Parceval e Exponente de Holder
- 8. Espectro DFA
- 9. Modelos Multifractais
- 10. Espectro de Singularidades (MFdfa)

Exemplos de Mapeamentos Caóticos

Equação logística: $A_{n+1} = \rho A_n(1 - A_n)$. Gere duas séries, uma considerando $\rho = 3.75$ e outra com $\rho = 4$. Para ambos os casos considere $A_0 = 0.0001$

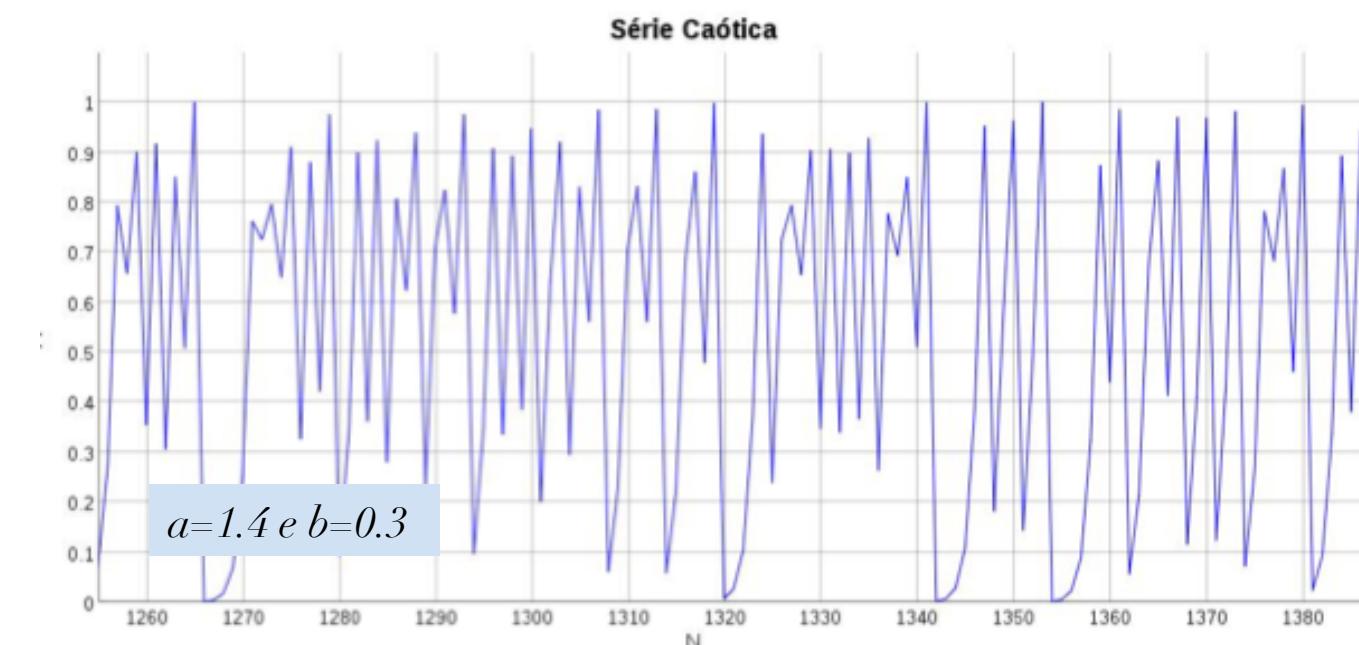


Conhecido também como:
Logistic Map
ou ainda como
Quadratic Equation

Henon Map

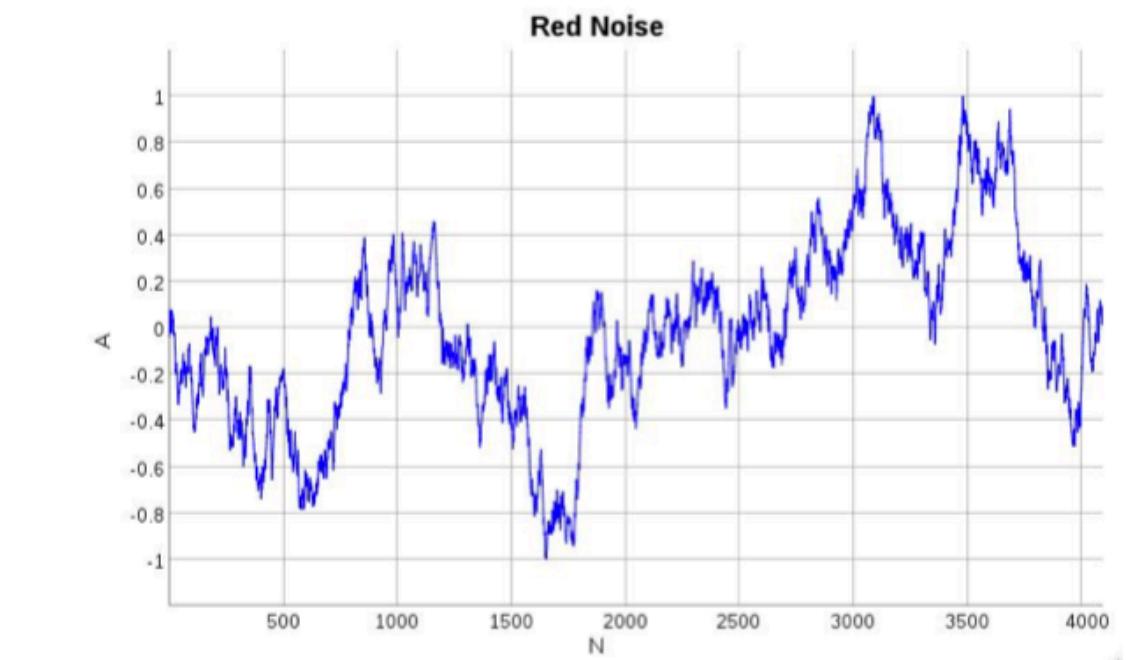
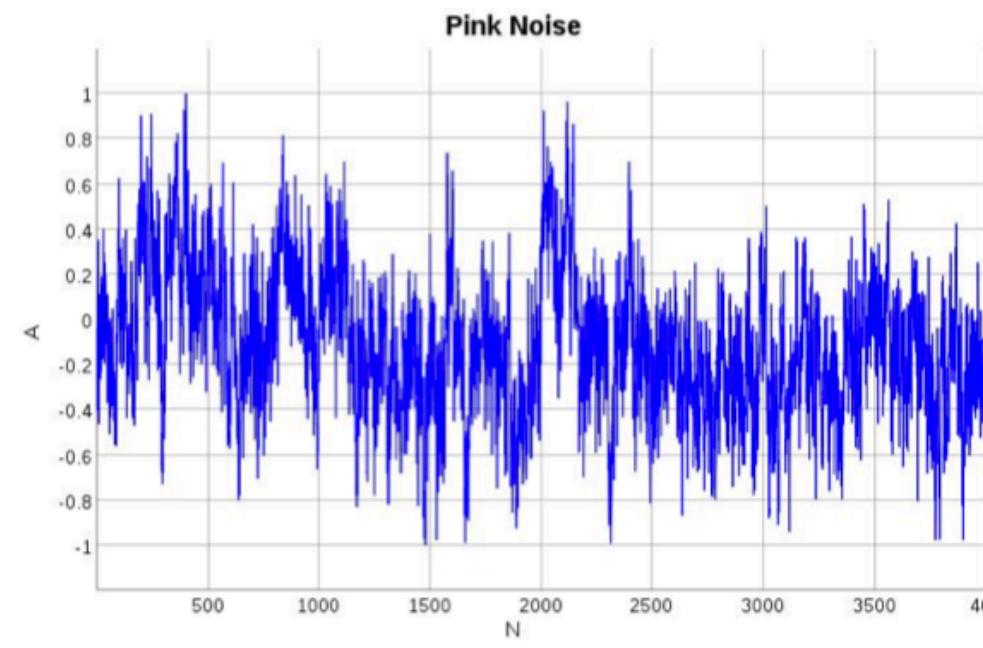
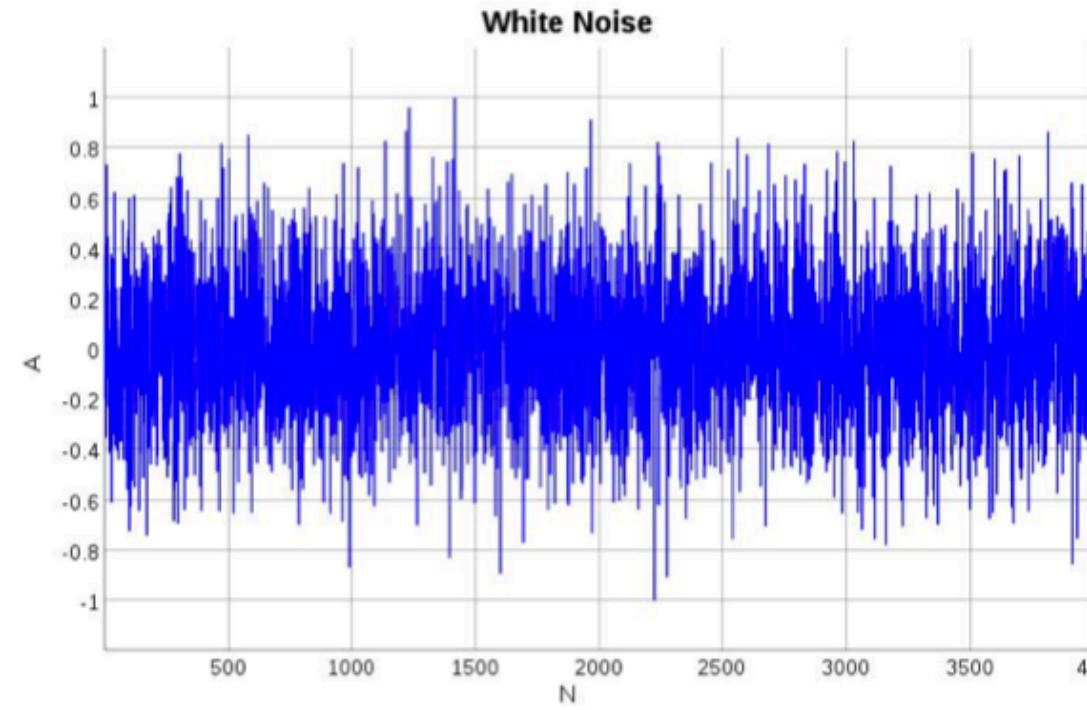
$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n. \end{cases}$$

$x(n)$

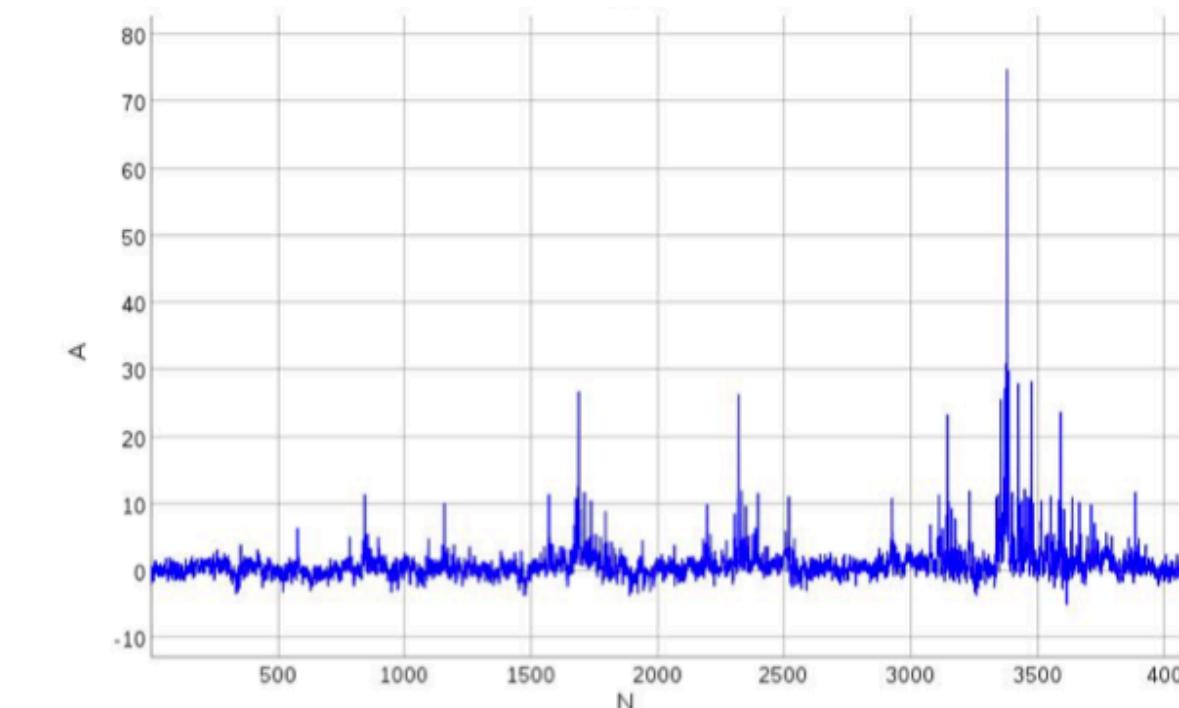
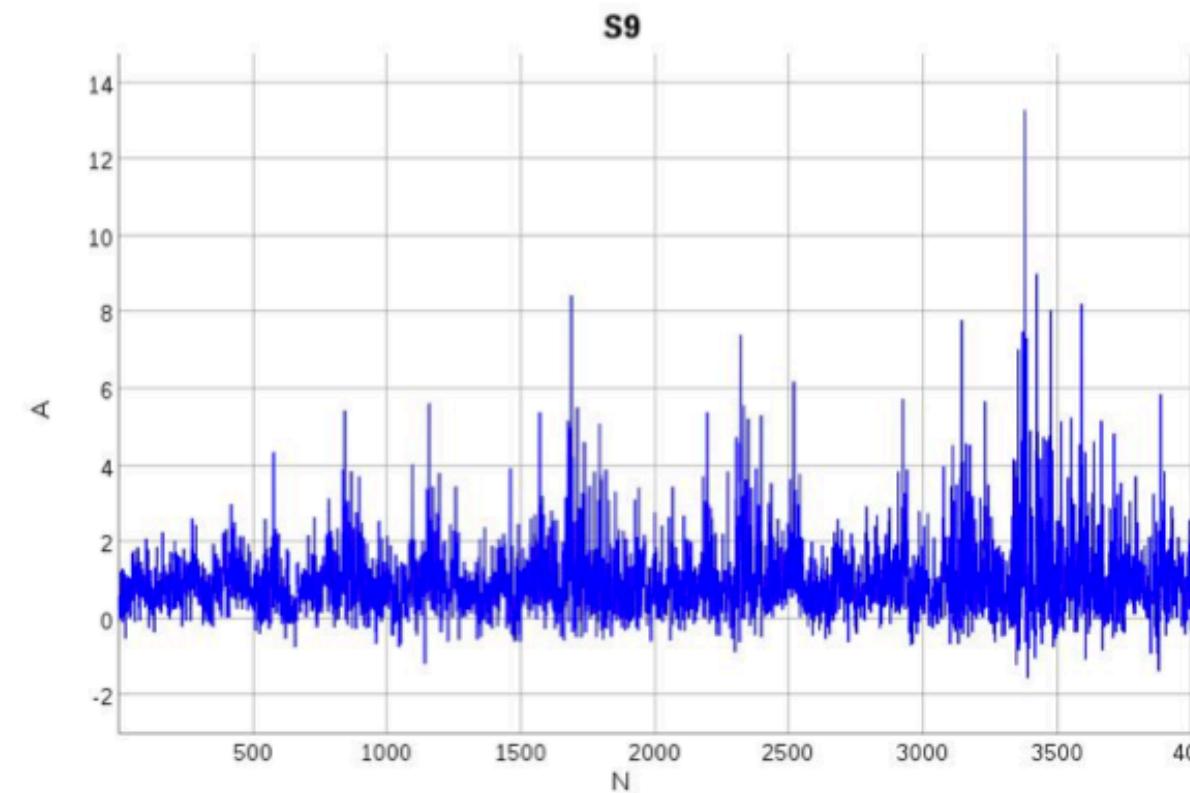


Exemplos de Ruidos $1/f^\beta$

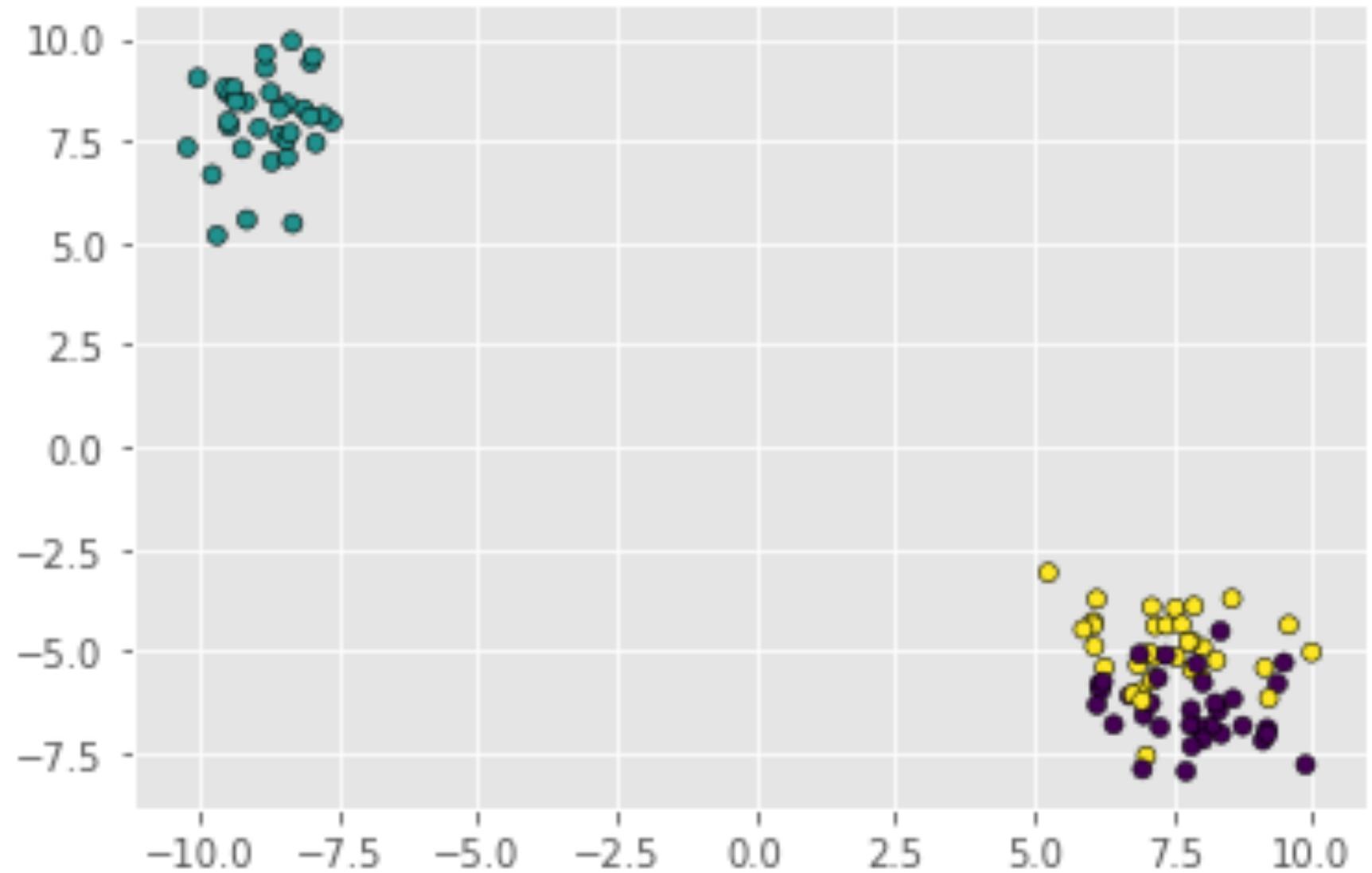
Input for Colorednoise.py: N e beta (0, 1, 2)



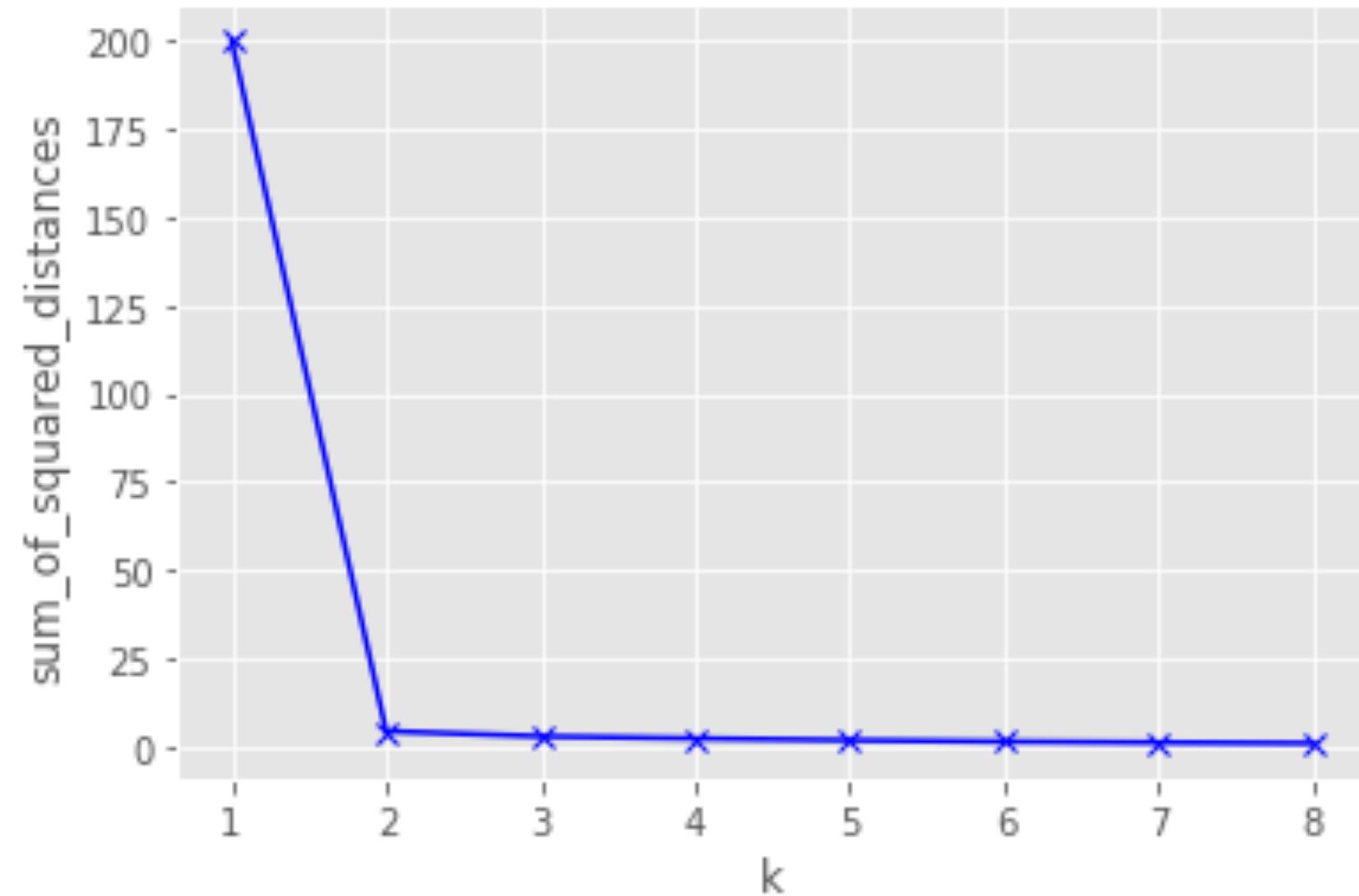
Input for pmodel.py: N, p (>0.5) e beta (0, 1, 2)



K blobs

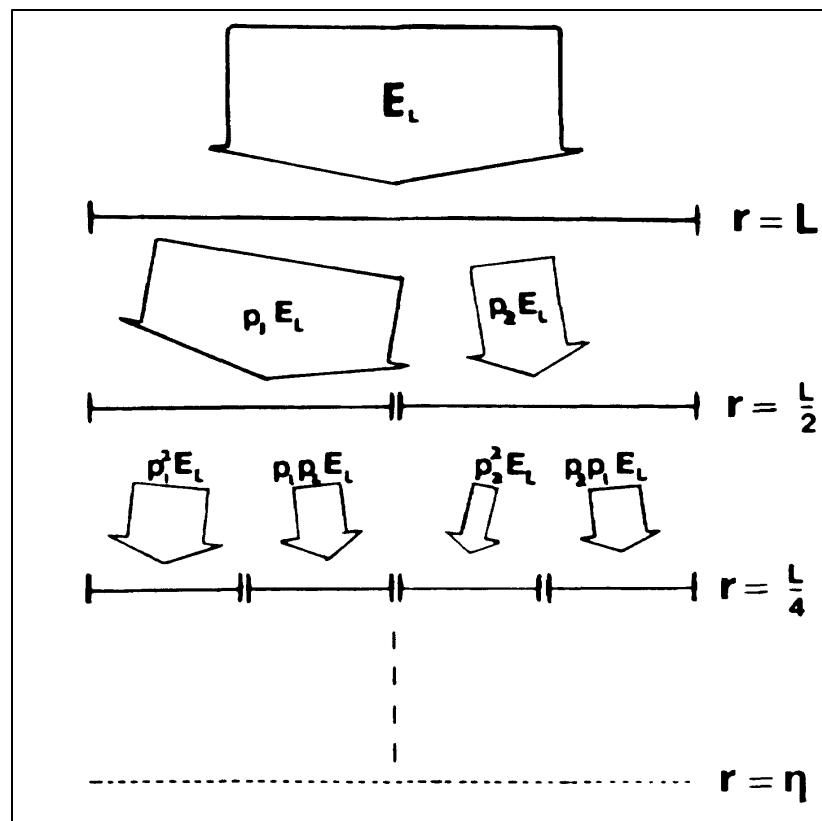


elbow method for optimal k



P model- Multifractal Inhomogeneous Multiplicative Cascade

- shows multifractal properties of one-dimensional sections of the dissipation field [Meneveau & Sreenivasan, 1987].
- Analytical formulation to determine the singularity spectrum [Halsey et al., 1986].



$$r_n = \frac{L}{2^n}.$$

$$\alpha = \frac{\ln p_1 + (n/m - 1) \ln p_2}{\ln l_1 + (n/m - 1) \ln l_2}$$

$$W_i^{\pm}) = 1 \pm (1 - 2p)$$

$$= \begin{cases} 2p & 0 \leq p \leq 1/2, \\ 2(1 - p) & 1/2 \leq p \leq 1 \end{cases}$$

$$f(\alpha) = \frac{(n/m - 1) \ln(n/m - 1) - (n/m) \ln(n/m)}{\ln l_1 + (n/m - 1) \ln l_2}$$

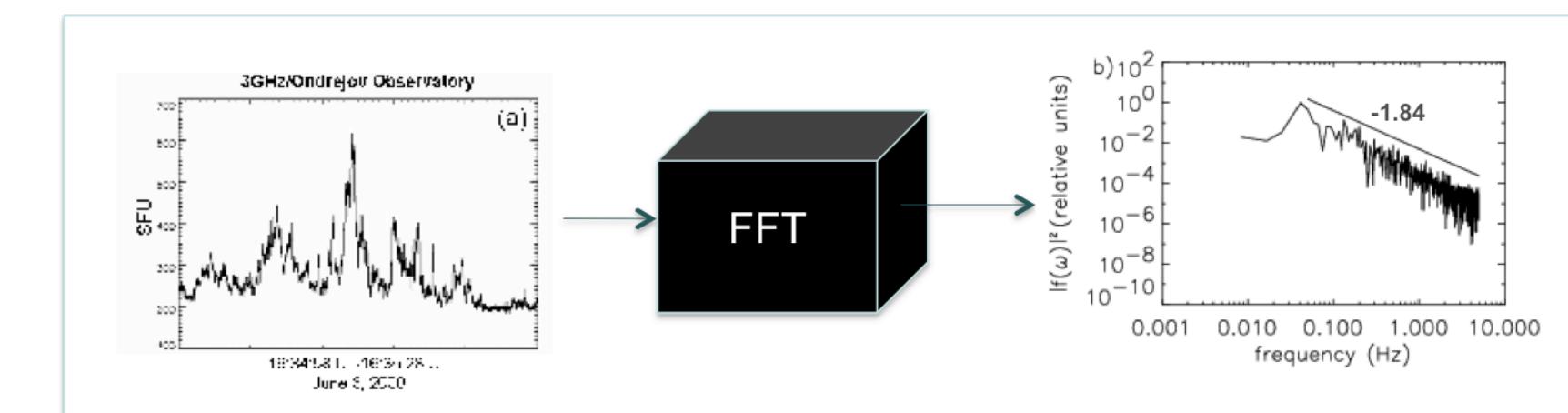
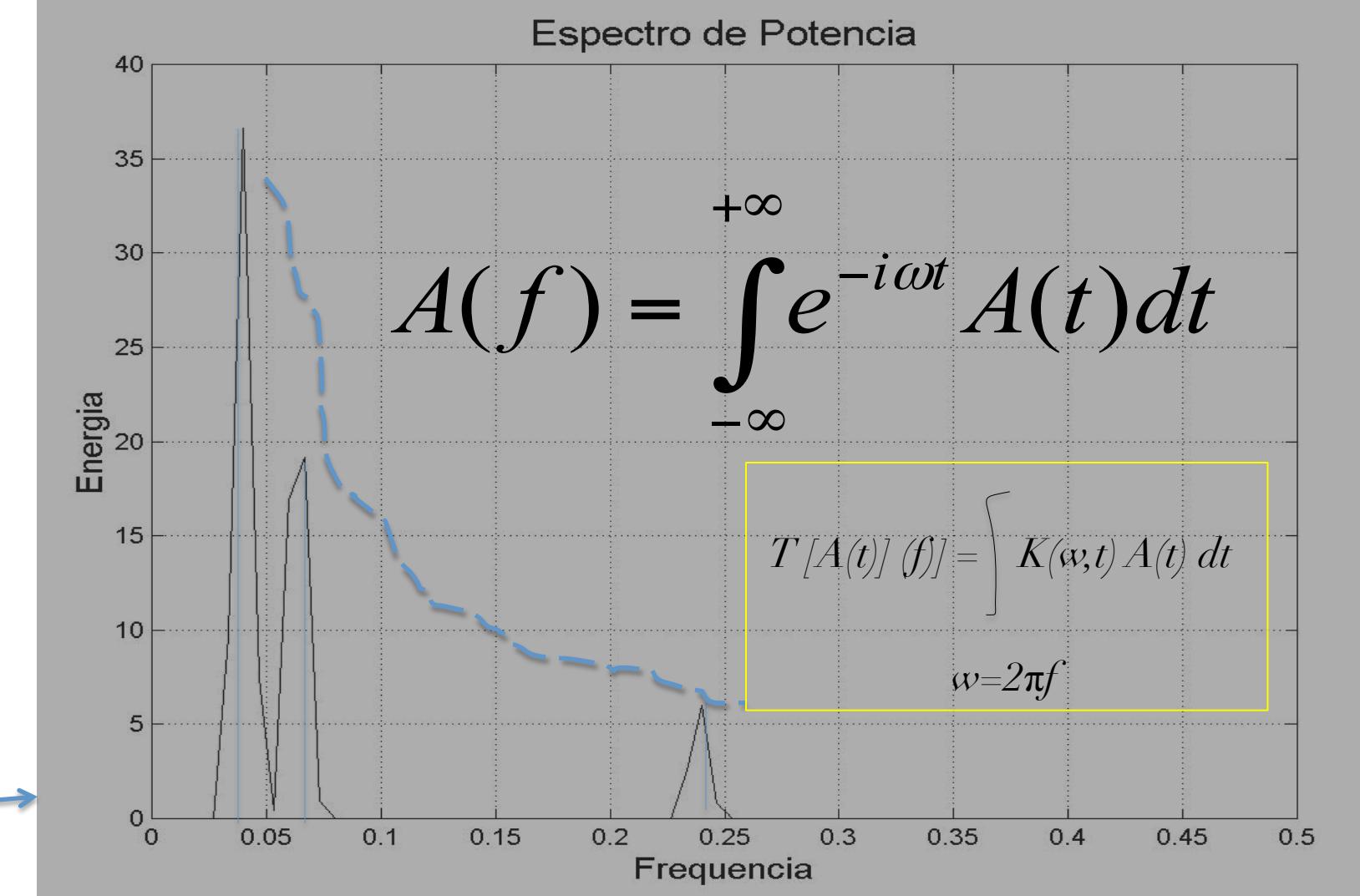
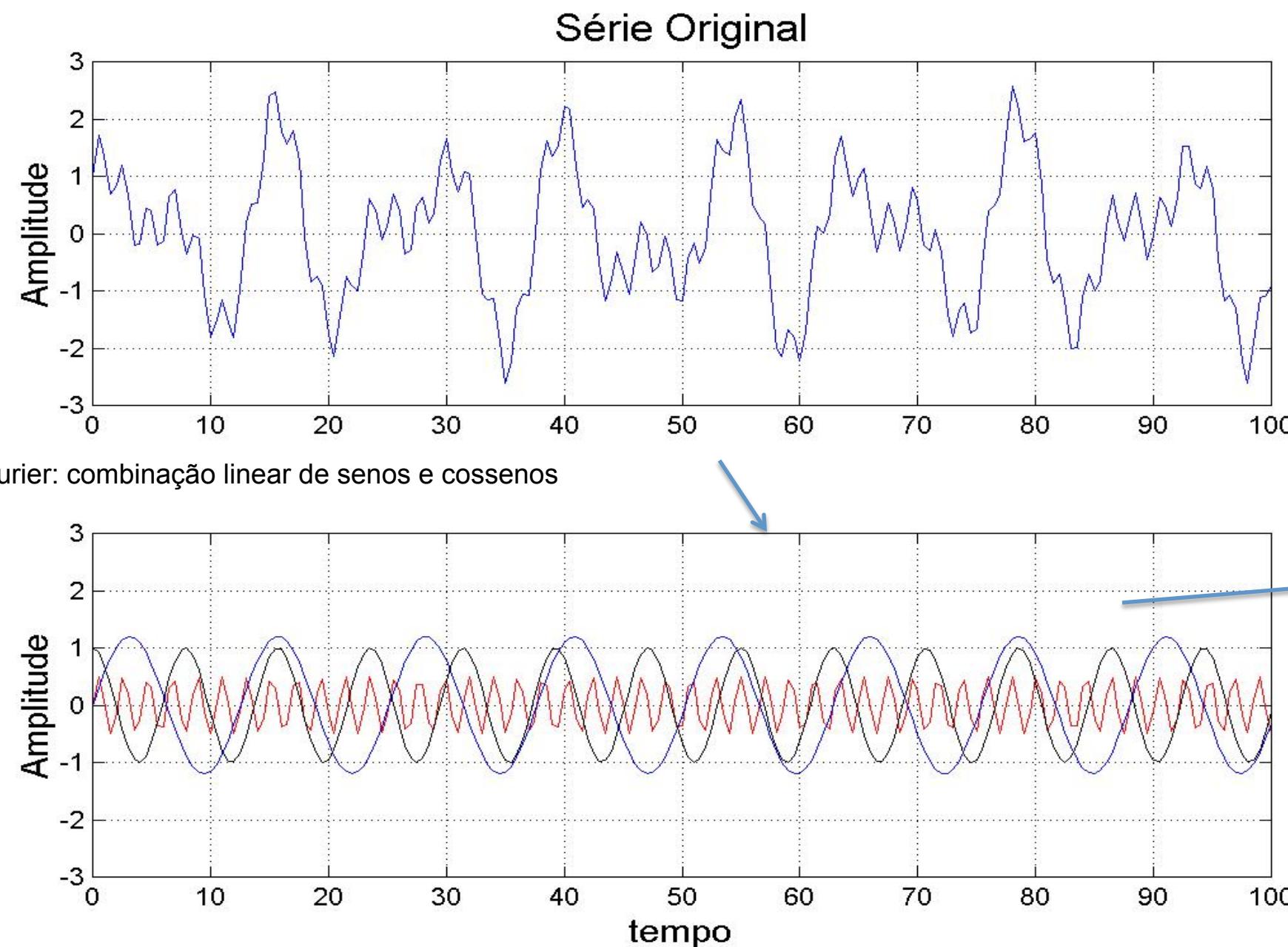
Probability parameters: p_1 and p_2 where $p_1 + p_2 = 1$
with dissipation parameter dp , $p_1 + p_2 + dp = 1$

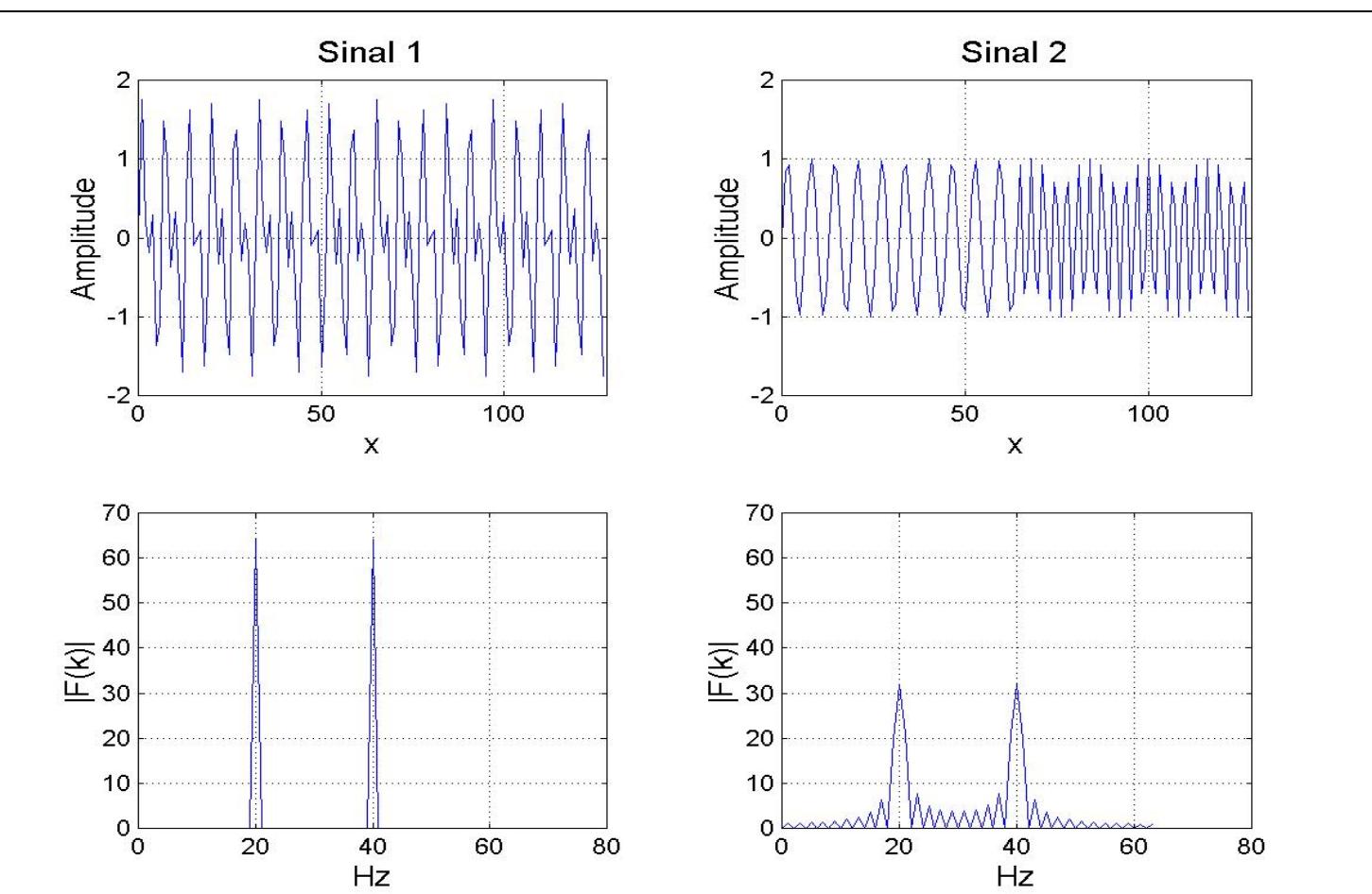
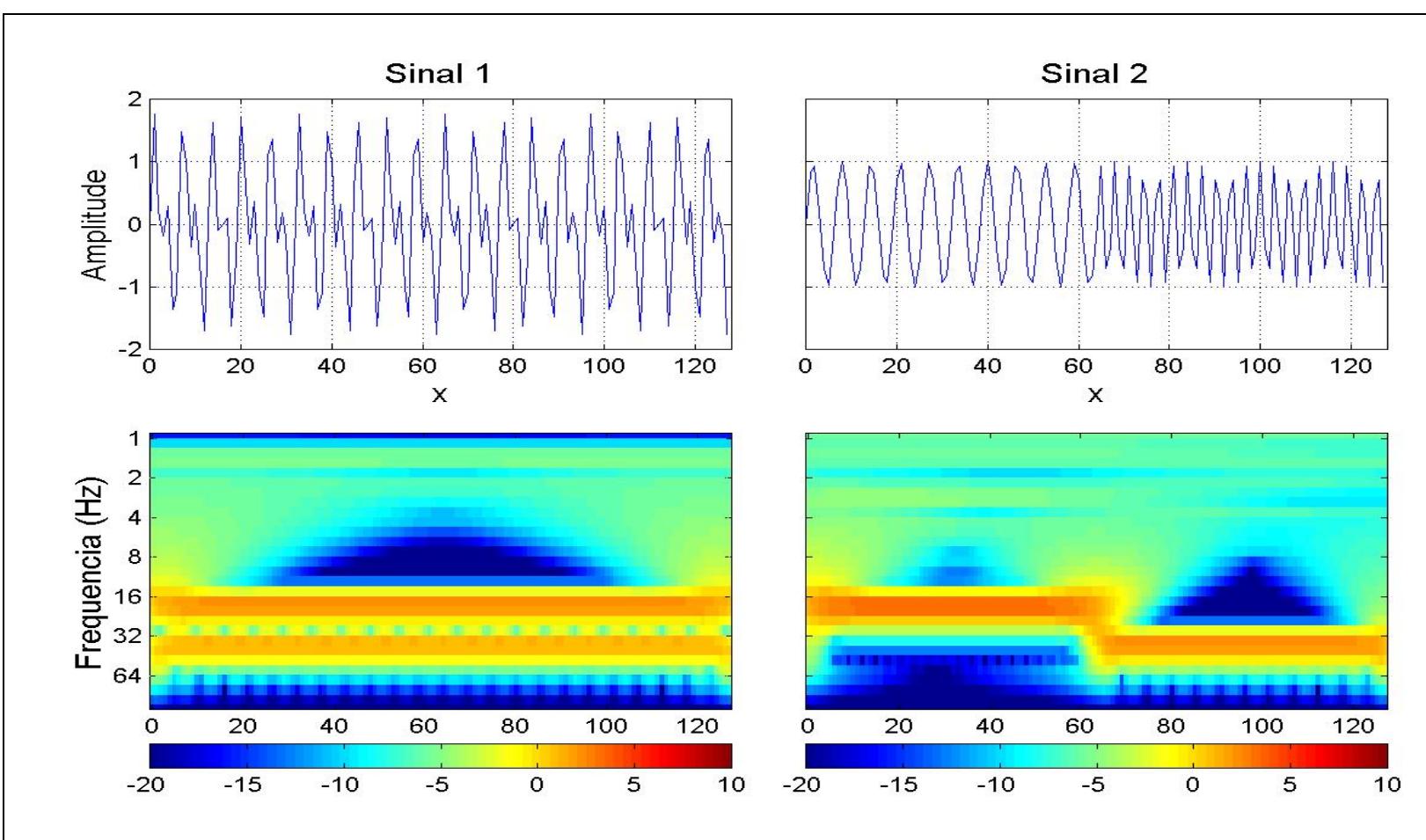
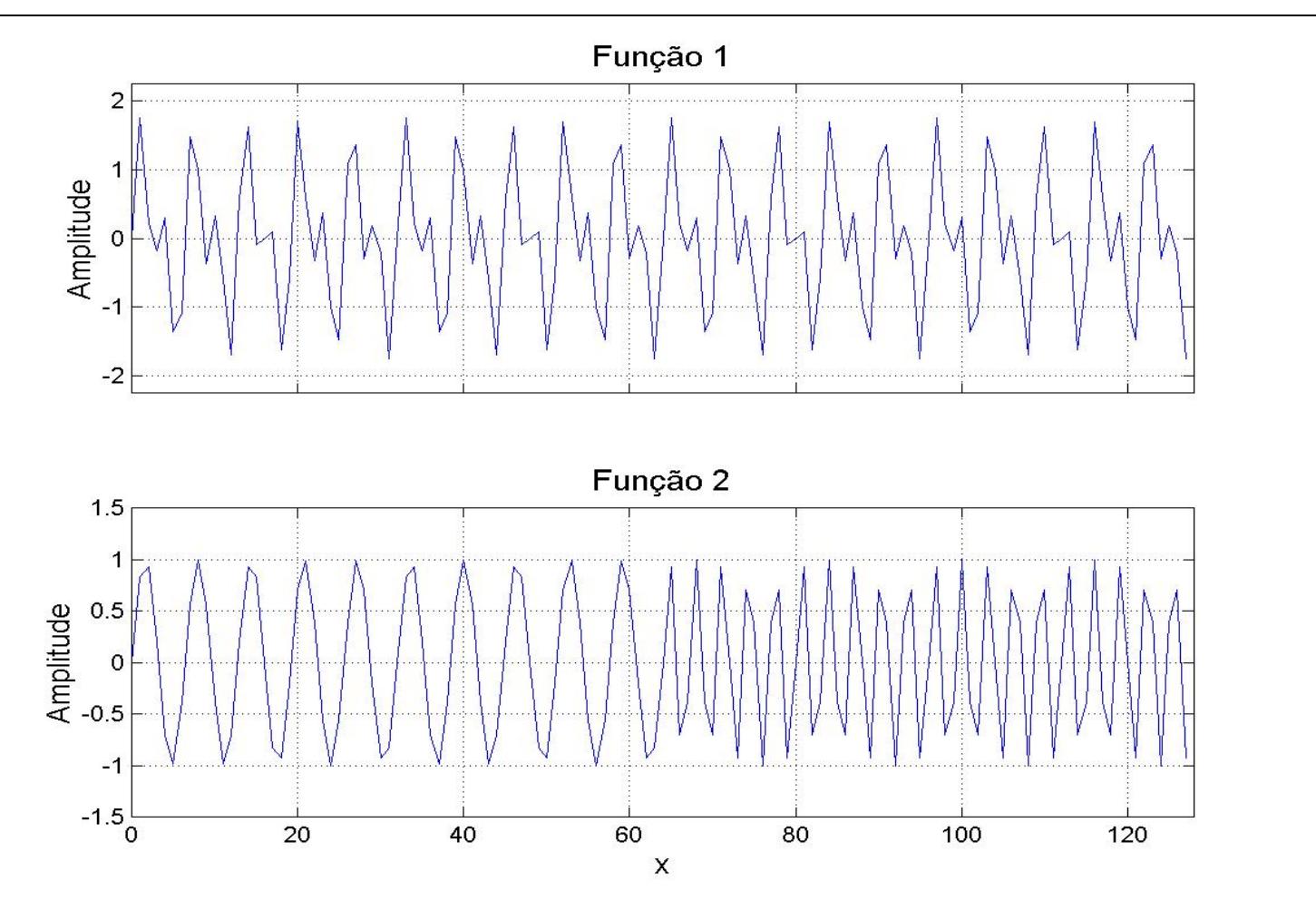
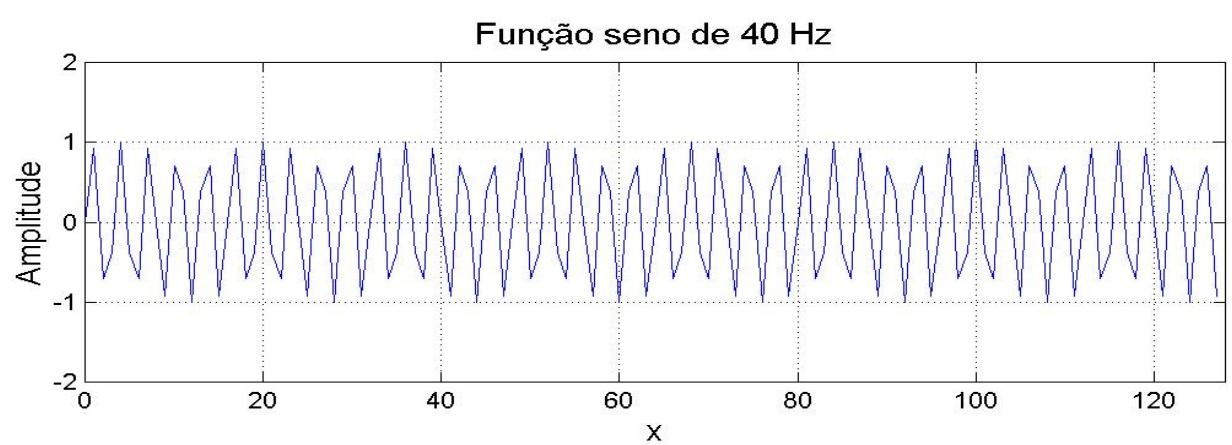
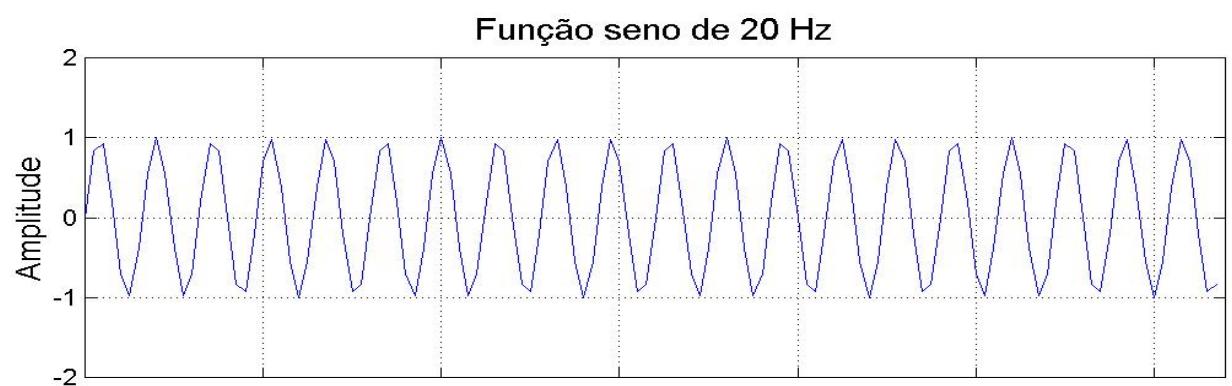
Length scales: l_1 and l_2 where $l_1 = l_2 = 0.5$

± random

SÉRIES E TRANSFORMADA DE FOURIER

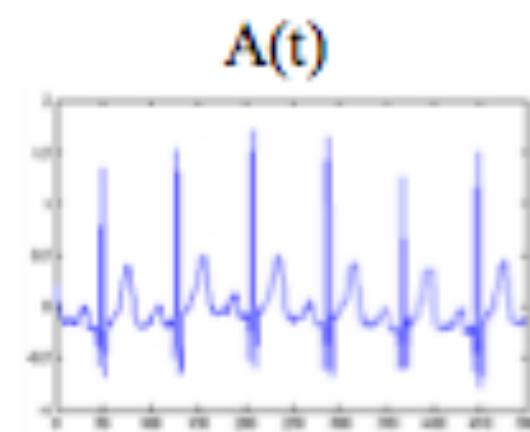
$$f(x) = \frac{a_0}{2} + \dots + a_n \cos(nx) + \dots + b_n \sin(nx)$$





Espectro de Potências

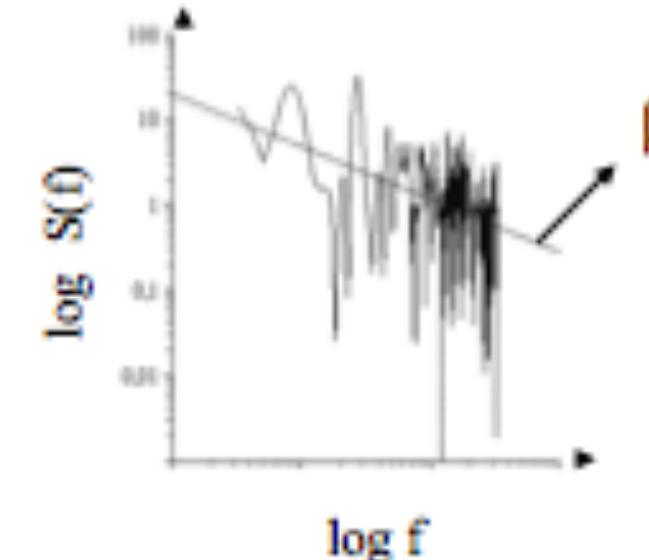
powspecscale.m



$$A(f) = \int_{-\infty}^{+\infty} e^{-i\omega t} A(t) dt$$



$$S(f) = |A(f)|^2$$



$O(N^2) \rightarrow O(N \log N)$

$$S(f) \sim f^{1/\beta}$$

$$E(k) \sim k^{-\beta},$$

$$S(f) = |\hat{x}_k|^2$$

$$\hat{x}_k = \frac{1}{\sqrt{N}} \sum_{n=1}^N x_n \exp\left[i \frac{2\pi n k}{N}\right], \quad k = 1, \dots, N.$$

$$E(k) \sim k^{-\beta}, \quad (1)$$

slope

In statistically isotropic situations, the “structure function” or “semivariogram” $\langle [\varphi(\mathbf{x} + \mathbf{r}) - \varphi(\mathbf{x})]^2 \rangle$ is a function of $r = |\mathbf{r}|$ alone; this function $D_\varphi(r)$ is related to the energy spectrum by a Wiener–Khinchin theorem (e.g., Monin and Yaglom 1975):

$$k=1/r \quad D_\varphi(r) = 2 \int_0^\infty [1 - \cos 2\pi kr] E_\varphi(k) dk. \quad (2)$$

The power law in Eq. (1) can be substituted into the above as long as $\beta \in (1, 3)$, yielding

$$D_\varphi = \langle [\varphi(\mathbf{x} + \mathbf{r}) - \varphi(\mathbf{x})]^2 \rangle \sim r^{2H_2} \quad (3)$$

with

$$H_2 = \frac{\beta - 1}{2} \in (0, 1). \quad (4)$$

$$\langle \ln \epsilon_n(x) \rangle \approx \frac{n}{2} \sum_{\pm} \ln[1 \pm (1 - 2p)]$$

Following the procedure described above for generating binomial multifractal time series, one can model the time series using p model parameters. To construct the time series, consider an initial length L with height ϵ_L . Divide it into two equal segments ($L/2$) with probabilities p_1 and p_2 (with $p_1 + p_2 = 1$) assigning randomly to the left or right part. Assign height as $2p_1\epsilon_L$ and $2p_2\epsilon_L$. Iterate this step for desired n^{th} generation to obtain time series. As probability p_1 is assigned randomly to the segments in each generation, different realization of the time series can be obtained in each run.

```
def pmodel(noValues=256, p=0.375, slope=[]):
    noOrders = int(np.ceil(np.log2(noValues)))
    noValuesGenerated = 2**noOrders

    fourierCoeff = fractal_spectrum_1d(noValues, slope/2)
    meanVal = np.mean(y)
    stdy = np.std(y)
    x = np.fft.ifft(y - meanVal)
    phase = np.angle(x)
    x = fourierCoeff * np.exp(1j*phase)
    x = np.fft.fft(x).real
    x *= stdy / np.std(x)
    x += meanVal
```

$$r_n = \frac{L}{2^n}.$$

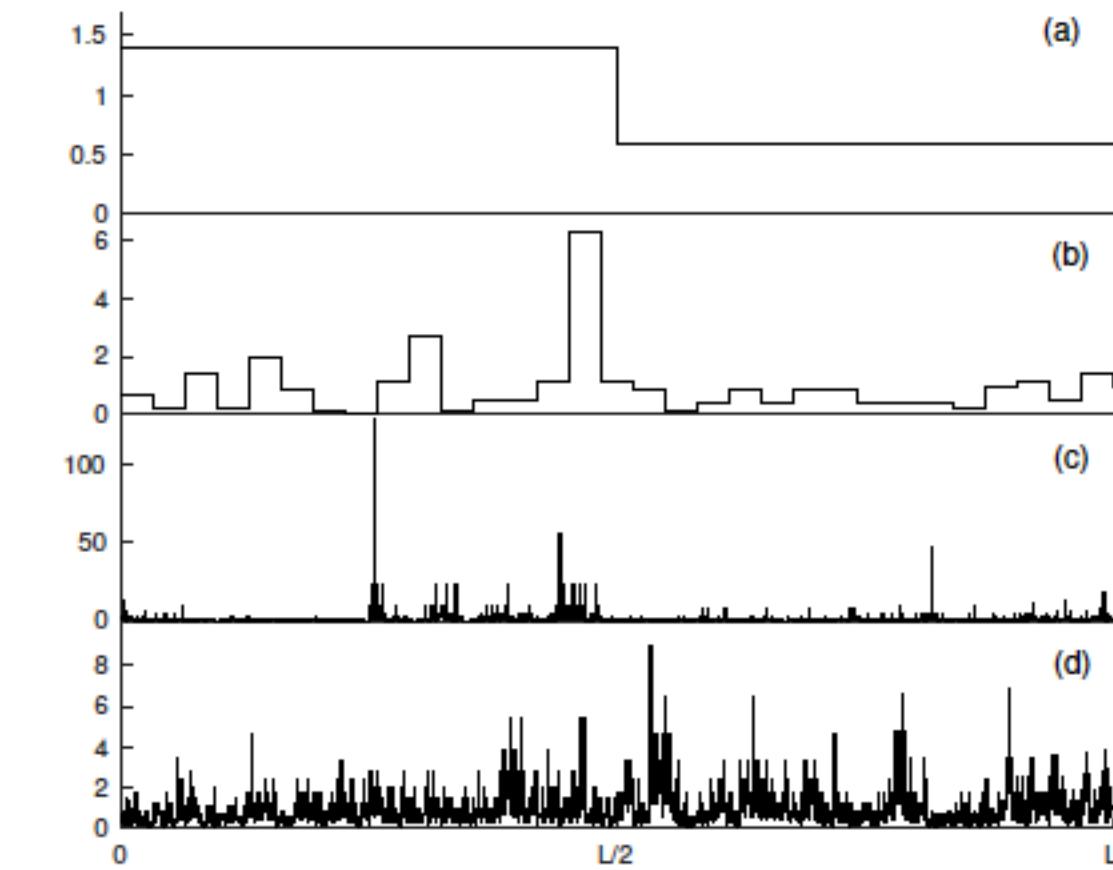


Figura 4.4 - p model time series: (a) first stage (b) fifth stage (c) twelfth stage (d) twelfth stage construction of time series with $p_1 = 0.418$.

Fonte: produced by author.

Conforme descrito em “Fast algorithm for a three-dimensional synthetic model of intermittent turbulence” (Malara et al., 2016) : No p-model uma distribuição espacial 1D de fluxo de energia em diferentes escalas, é construída através de um processo multiplicativo. Um redemoinho em uma escala L quebra em 2 redemoinhos na escala $L/2$, e o fluxo de energia E associado com o parente do redemoinho é desigualmente distribuído em dois redemoinhos filhos, com frações dadas por $2pE$ e $2(1-p)E$, respectivamente, com $0,5 \leq p \leq 1$.

Cria-se um array de um elemento contendo um número aleatório:

```
>>> sign = np.random.uniform(0, 1, length) - 0.5
```

```
>>> sign
```

```
array([ 0.3706906])
```

É extraída a informação se é positivo ou negativo, retornando 1 ou -1:

```
>>> sign = sign / abs(sign)
```

```
>>> sign
```

```
array([ 1.])
```

São calculadas as “frações $2pE$ e $2(1-p)E$ ”

```
>>> y2[::2] = y + sign * (1 - 2 * p) * y
>>> y2
array([ 1.04,  0. ])
>>> y2[1::2] = y - sign * (1 - 2 * p) * y
>>> y2
array([ 1.04,  0.96])
return y2
```

```
y = [1]
for n in range(1, noOrders + 1, 1):
    y = next_step_1d(y, p)
```

O parâmetro do p-model é “ p ” e com valores próximos de 0 ou 1 existem muitos picos. Com valores perto de 0,5 é mais calmo. Em 0,5 o resultado é um vetor constante.

“y” é a série temporal gerada pelo p-model.

Inicialmente y tem um elemento 1. Vamos supor $p=0.52$.

```
y = [1]
for n in range(1, noOrders + 1, 1):
    y = next_step_1d(y, p)
```

Neste caso “next_step_1d” é chamado com os parâmetros [1] e 0.52.

```
>>> y = [1]
>>> length = len(y)
>>> len (y)
1
```

A variável length recebe a quantidade atual de elementos na série (no caso 1).

É criado um array temporário com dois elementos vazios:

```
>>> y2 = np.zeros(length * 2)
>>> y2
array([ 0.,  0.])
```

P-model

Time Series Generator

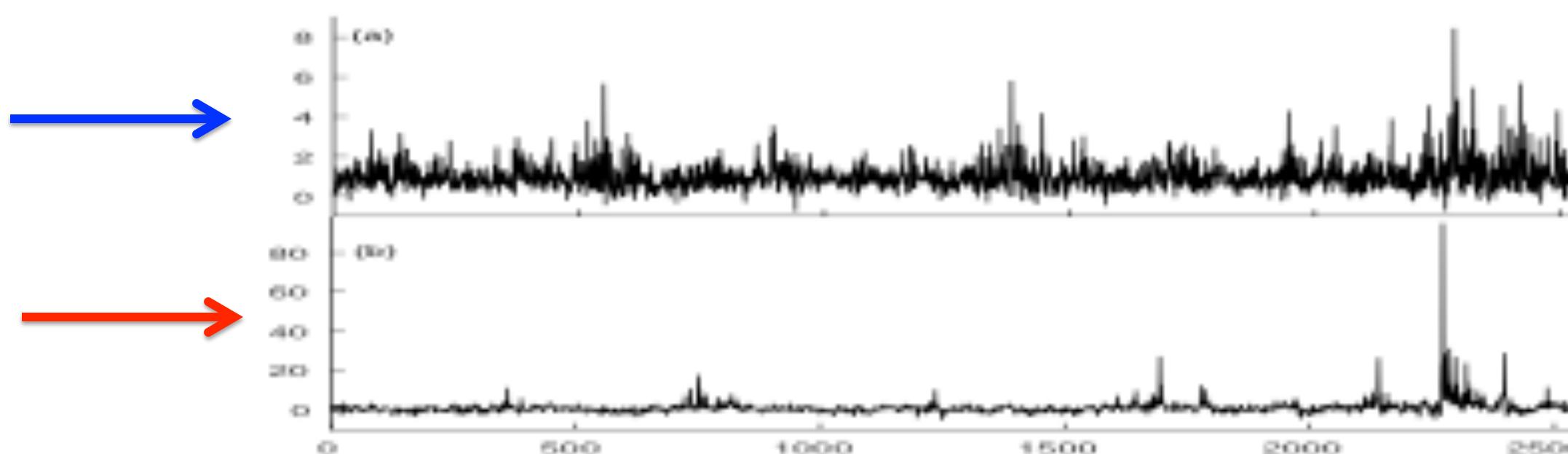
- » Sornette et al. (2004) studied book sale ranking series on Amazon website
- » book sale series similar as cascade process
- » characterized β endogenous = -0.4 & β exogenous = -0.7
- » Enhanced by determining the p ranges (Neelakshi, 2020):
endo: 0.32-0.42; exo: 0.18-0.28

Endogenous series:

- ▶ internal perturbation
- ▶ smoother continuous fluctuations around peak

Exogenous series:

- ▶ external perturbation
- ▶ rapid drop of fluctuations around suddenly risen peak



Mas o que é uma série multifractal gerada pelo p-model
em relação à teoria das variedades fractais?