

Lista de exercícios Matemática Computacional
Parte B – Prof. Dr. Reinaldo Rosa - 2020

Denis M. A. Eiras

Exercício 4 – Descrição

4- Espaço de Cullen-Frey e Distribuições de Probabilidades

4.1. Considere duas das séries das famílias N8 dos exercícios anteriores. Classifique as mesmas no espaço de Cullen-Frey.

4.2.1 Escreva (ou utilize) um algoritmo em Python que permita ajustar uma PDF (Gaussiana ou GEV) aos respectivos histogramas selecionadas em 2.1.

4.2.2 Escreva (ou adapte um programa) que permita ler arquivo de uma série genérica com diferentes tamanhos.

Exercício 4.1 – Detalhes da implementação

O exercício utiliza a função `gerador_de_sinais_pmodel`, do arquivo `exercicio3.py`, para gerar as duas séries escolhidas: séries p-Model.

O método para plotar as séries no espaço de Cullen-Frey utilizado foi convertido da linguagem R para a linguagem Python, e incorporado à pasta deste exercício.

Exercício 4.2.1 – Detalhes da implementação

Foi utilizado o gerador de PDF “`genextreme`” do pacote `scipy.stats` para se fazer a curva de ajuste sob o histograma.

Exercício 4.2.2 – Detalhes da implementação

A função `ler_serie_generica_de_arquivo_ou_url` foi criada para ler arquivos ou urls contendo arquivos, que são analisados quanto ao seu tipo antes da leitura.

Foram utilizadas as bibliotecas `pandas`, para ler arquivos csv e as bibliotecas `python-magic` e `mimetypes` para descobrir o tipo de arquivo com extensão genérica.

Exercício 4.1 – Análise e resultado.

Foram utilizados os sinais p-Model endógenos e exógenos para gerar a figura 1.

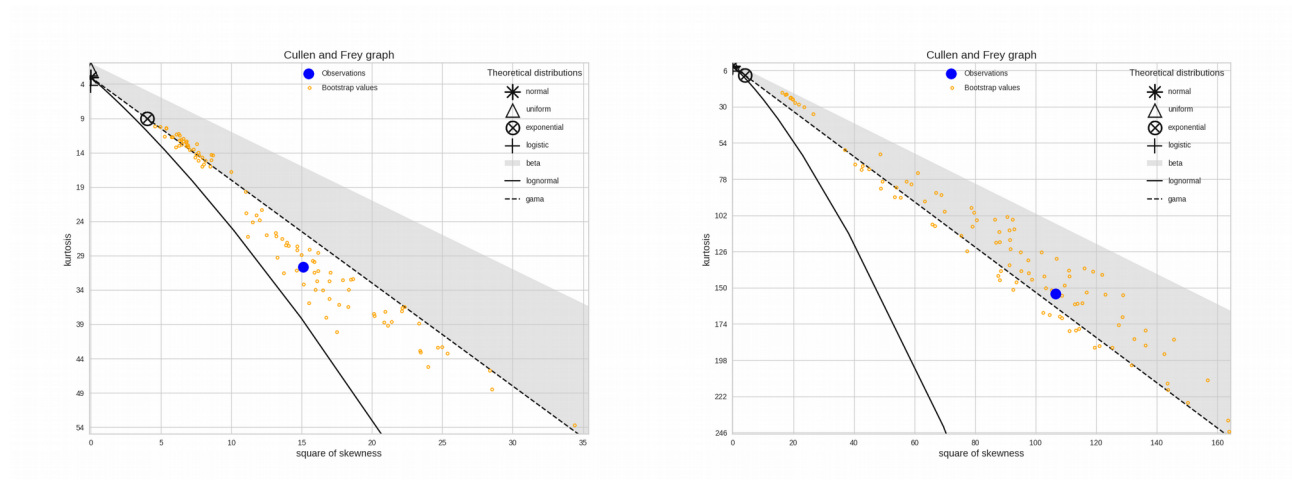


Figura 1. Espaços de Cullen-Frei. a) Sinal endógeno. b) Sinal exógeno

Como podemos observar, a figura 1.a classifica o sinal endógeno entre os espaços lognormal e gama, e a figura 1.b, classifica o sinal exógeno dentro do espaço beta, em sua maioria, mas também com pontos no espaço gama.

Exercício 4.2.1 – Resultados

Foram utilizados seguintes parâmetros da PDF para ajuste da curva do sinal endógeno, gerado no exercício4_1.py. A curva da figura 2 foi gerada com a utilização desses parâmetros sob o histograma.

$c = 0.3$; $\text{loc} = -1.0$; $\text{scale} = 1.0$

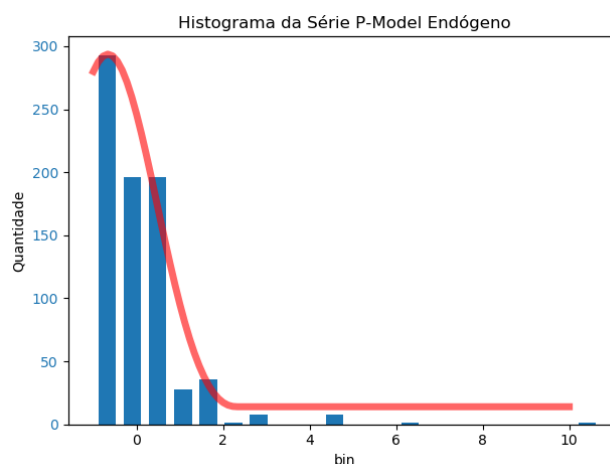


Figura 2. Ajuste da PDF sobre o sinal endógeno

Onde a função PDF é calculada da seguinte forma:

$$f(x, c) = \begin{cases} \exp(-\exp(-x)) \exp(-x) & \text{for } c = 0 \\ \exp(-(1 - cx)^{1/c})(1 - cx)^{1/c-1} & \text{for } x \leq 1/c, c > 0 \end{cases}$$

A densidade de probabilidade acima é definida na forma "padronizada". Para mudar e / ou escalar a distribuição, os parâmetros loc e scale devem ser alterados.

Foram utilizados os parâmetros abaixo da PDF para ajuste da curva do sinal exógeno, gerado no exercício4_1.py. A curva da figura 3 foi gerada com a utilização desses parâmetros sob o histograma.

$c = 0.2$; $loc = 0$; $scale = 1.0$

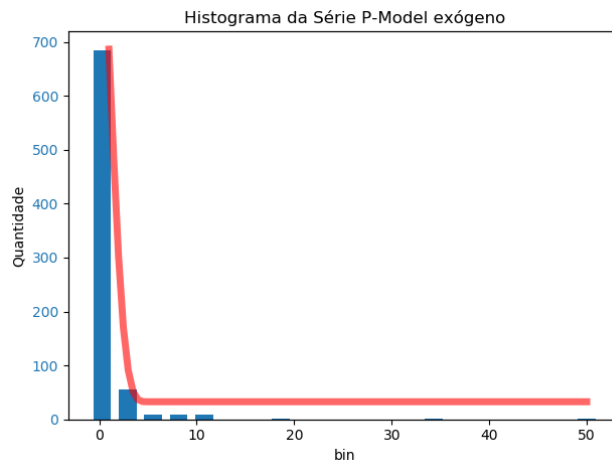


Figura 3. Ajuste da PDF sobre o sinal exógeno

Exercício 4.2_2 – Resultados

O código implementado foi capaz de ler os arquivos texto, até mesmo com extensão genérica, por exemplo .dat. Também foi possível ler um arquivo através de uma url.

A função implementada retorna um array, para uma série simples contendo uma coluna, ou arrays de arrays, para séries contendo mais de uma coluna. Também foi implementado o parâmetro `is_obter_csv_como_dataframe`, que quando ligado, retorna um objeto do tipo `DataFrame` da biblioteca Pandas, após ler um arquivo .csv. Assinatura da função:

```
def ler_serie_generica_de_arquivo_ou_url(nome_arquivo_ou_url, is_url=False,
is_obter_csv_como_dataframe=False):
```

Exemplos de arquivos lidos:

```
# serie texto
serie = ler_serie_generica_de_arquivo_ou_url('./resultados4_2_2/surftemp504.txt')
print(serie)
# serie texto, com extensão diferente
serie = ler_serie_generica_de_arquivo_ou_url('./resultados_4_2_2/sol3ghz.dat')
print(serie)
# serie csv
serie = ler_serie_generica_de_arquivo_ou_url('./resultados_4_2_2/teste_noise.csv')
print(serie)
# serie csv para DataFrame
serie = ler_serie_generica_de_arquivo_ou_url('./resultados_4_2_2/teste_noise.csv', is_obter_csv_como_dataframe=True)
print(serie)
# binário não lê
serie = ler_serie_generica_de_arquivo_ou_url('/bin/echo')
print(serie)
# texto com cabeçalho
serie = ler_serie_generica_de_arquivo_ou_url('./resultados_4_2_2/arq_texto_2_colunas_sem_cabecalho')
print(serie)
# texto sem cabeçalho
serie = ler_serie_generica_de_arquivo_ou_url('./resultados_4_2_2/arq_texto_2_colunas_com_cabecalho')
print(serie)
# url
url_owid_covid_data = 'https://covid.ourworldindata.org/data/owid-covid-data.csv'
serie = ler_serie_generica_de_arquivo_ou_url(url_owid_covid_data, is_url=True)
print(serie)
```