



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Autorizada pelo Decreto Federal nº 77.496 de 27/04/76
Recredenciamento pelo Decreto nº 17.228 de 25/11/2016



PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS SEMANA NACIONAL DE CIENTÍFICA E TECNOLÓGICA - 2019

PROCESSAMENTO DE SÉRIES TEMPORAIS USANDO O MÉTODO DFA EM PYTHON

Natália Silva Rosa¹; Gilney Figueira Zebende²

1. Bolsista PIBIC/CNPq, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: srosanatalia@gmail.com
2. Orientador Gilney Figueira Zebende, Departamento de Física, Universidade Estadual de Feira de Santana, e-mail: gfzebende@hotmail.com

PALAVRAS-CHAVE: Python, DFA, séries temporais.

INTRODUÇÃO

As séries temporais são um dos tipos de dados mais comuns encontrados na natureza, que podem ser medidos, guardados e processados em tempo real ou off-line, usando um Computador. Exemplos destes dados temos a mudança no tempo dos preços dos produtos no mercado, do clima, do uso da energia, da mídia social, dos sinais biomédicos gerados pela atividade fisiológica da matéria viva, dentre outros muitos mais. A análise de séries temporais permite-nos analisar determinados dados ao longo de um período de tempo e compreender padrões nos dados ao longo do tempo, suas tendências, sazonalidade, etc. O grande volume de dados e as características específicas dos tipos de séries temporais nas diversas áreas do conhecimento, obriga aos pesquisadores o desenvolvimento de métodos computacionais e estatísticos que auxiliem nas suas análises e assim uma melhor tomada de decisões antes os problemas a estudar.

Entre os métodos estatísticos e a modelagem computacional como ferramenta para análises de dados de séries temporais, encontramos o método baseado nas Análises de Flutuações sem Tendências (DFA) proposto por Peng et al. (1994). Neste trabalho foram realizados estudos acerca do método DFA e sua implementação na linguagem Python.

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

A metodologia utilizada, é fundamentada basicamente pela revisão bibliográfica usando vários livros de análises estatística por Jens Feder (1988) e do aprimoramento da linguagem de código aberta Python. O *software* produto deste trabalho foi desenvolvido utilizando o Sistema Operacional *Microsoft Windows 10*, como editor de texto e compilador do programa a plataforma *Python 3.7 (32-bit)*. As interfaces gráficas de usuário, bem como os cálculos matemáticos foram feitas através de bibliotecas como *NumPy*, *Matplotlib*, *Os* e *Qt*.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

Os três primeiros meses de pesquisa constituíram-se em realizar estudos baseados no método *DFA*, além de estudos avançados em *Python*. Passado o primeiro trimestre, iniciou-se a implementação de rotinas em *Python* para a aplicação do método *DFA*.

A primeira etapa de estudos foi feita essencialmente utilizando artigos científicos que descreviam o método *DFA*.

Para a implementação e análise do método, foi imprescindível seguir os seguintes passos (Laczkowski, 2013):

i) A partir da série original, contendo N pontos, obter a série acumulada;

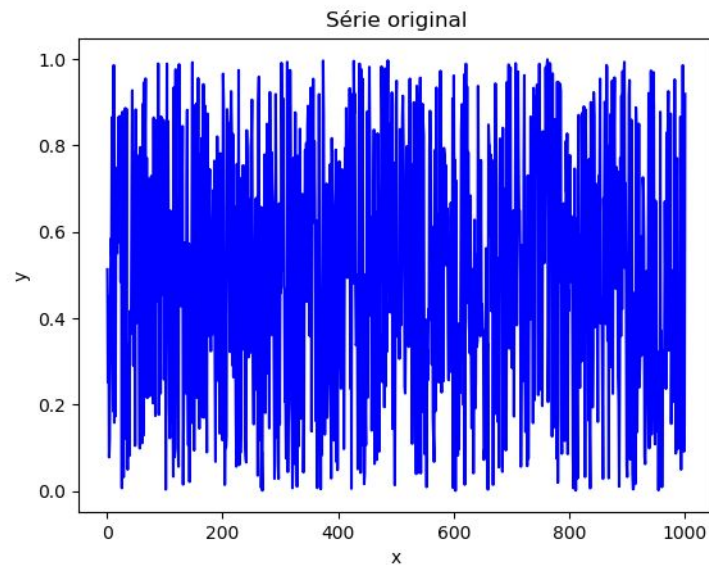


Figura 1: Gráfico contendo a série original, gerado a partir do arquivo de teste.

ii) Dividir a série acumulada em janelas ou subconjuntos de tamanho n ;

Foram realizados dois tipos de análises. A primeira com janelas não sobrepostas e a segunda utilizando a sobreposição de janelas. Para a série não sobreposta teríamos $\frac{N}{n}$ janelas e para a sobreposta $N - n$, sendo n o tamanho das janelas, possuindo como valor mínimo 4.

iii) Remover a tendência local em cada janela por meio de um ajuste polinomial;

iv) Calcular a função de flutuação $F(n)$ para as análises sobrepostas e não sobrepostas em termos do tamanho das janelas.

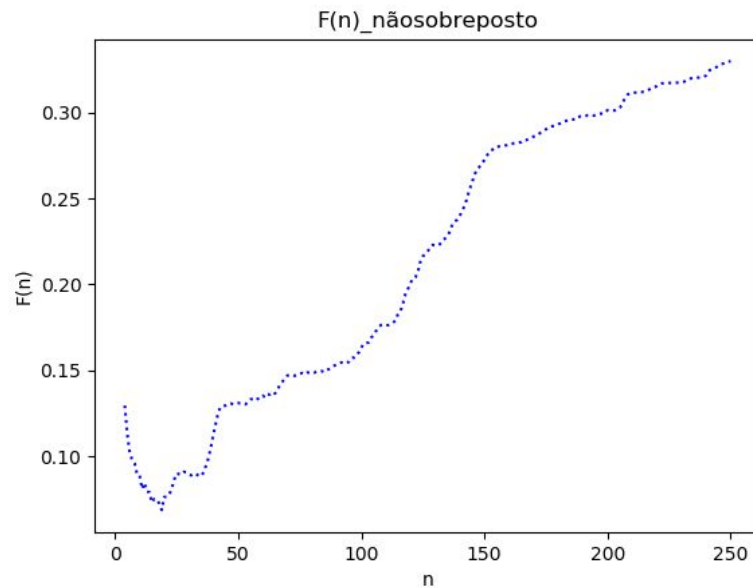


Figura 2: Gráfico gerado a partir das funções não sobrepostas calculadas a partir do arquivo de teste.

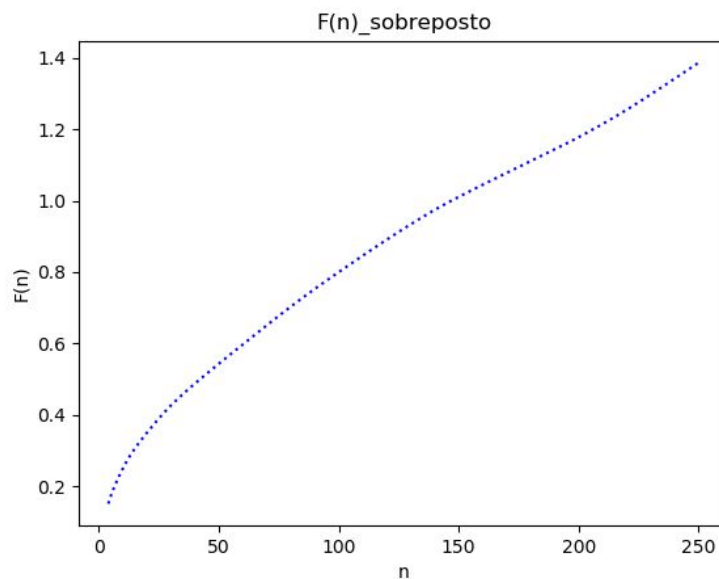


Figura 3: Gráfico gerado a partir das funções sobrepostas calculadas a partir do arquivo de teste.

O programa tem como entrada um arquivo de texto, escolhido pelo usuário, com extensão *.txt*, contendo a lista de elementos da série que será manipulada. Ao ler o arquivo, cada elemento é armazenado em uma posição de *array*, e para isso foi utilizada a biblioteca *NumPy* pois ela facilita operações sofisticadas sobre objetos *array*, como por exemplo o ajuste polinomial aplicado às janelas.

Também foi utilizada a biblioteca *Matplotlib*, para gerar os gráficos referentes às funções de flutuação, a biblioteca *Os*, para facilitar a manipulação do nome do diretório do arquivo de entrada e permitir que as saídas dos programas pudessem ser salvas no

mesmo caminho do *input*. Por fim foi utilizado o *PyQt5*, um empacotador da linguagem *Python* para a biblioteca *Qt*, que foi necessário para desenvolver a janela que permite ao usuário navegar por seus diretórios e selecionar o arquivo que irá realizar a aplicação do método.

As saídas do programa são dois arquivos de texto contendo as função de flutuação $F(n)$ em termos do tamanho das janelas, sendo um para as funções não sobrepostas e outro para as funções sobrepostas, além de duas imagens que correspondem ao gráfico das referidas funções. Os testes foram realizados utilizando um arquivo de texto contendo uma série de 1000 pontos, gerada pelo orientador e entregue ao bolsista.

CONSIDERAÇÕES FINAIS (ou Conclusão)

A fim de otimizar o *software* produto da pesquisa, seria interessante incluir funcionalidades que pudessem melhorar o desempenho do programa ao trabalhar com séries muito grandes, tornando a execução dos cálculos mais veloz.

Entretanto, os requisitos principais que foram solicitados, foram cumpridos durante a vigência da bolsa, restando apenas a escrita do artigo científico que será feita futuramente conforme o estudo a respeito do método *DFA* for ainda mais aprofundado. As rotinas escritas em *Python*, conseguiram alcançar os resultados esperados, realizando com eficácia os testes propostos, desta forma está garantida ao usuário total confiança nos dados gerados.

REFERÊNCIAS

PENG, C. et al. Mosaic organization of dna nucleotides. *Physical Review E, United States*, v. 49, n. 2, p. 1685-1989, 1994.

A sliding windows approach to analyse the evolution of bank shares in the European Union. PJ Ferreira, A Dionísio, EF Guedes, GF Zebende *Physica A: Statistical Mechanics and its Applications* 490, 1355-1367, 2018.

Oil price and exchange rate co-movements in Asian countries: Detrended cross-correlation approach. M Hussain, GF Zebende, U Bashir, D Donghong, *Physica A: Statistical Mechanics and its Applications* 465, 338-34613, 2017

Auto-correlation in the motor/imaginary human EEG signals: A vision about the FDFA fluctuations, GF Zebende, FM Oliveira Filho, JAL Cruz. *PloS one* 12 (9), e0183121, 2017.

Analysis of the variability in the sdB star KIC 10670103: DFA approach GF Zebende, BF Fernandez, MG Pereira, *Monthly Notices of the Royal Astronomical Society* 464 (3), 2638-2642 2, 2016.

Aplicação do método DFA na identificação de correlações e fractalidades em séries temporais sonoras. VII Encontro de Produções Científicas e Tecnológicas. IM Laczowski, MS Laczowski. 2013.