Oscilações Econômicas e a Transformada de Fourier

MAP2220 - Fundamentos de Análise Numérica
Prof. Pedro da Silva Peixoto
Prof. Antoine Laurain

Entrega: 13/11/2018

Enviar para: ppeixoto@usp.br ;
laurain@ime.usp.br ; paula.neves.araujo@hotmail.com
Assunto: MAP2220-EP2-diurno/noturno

Instruções

O exercício deve ser feito em Python 3.x (o mesmo usado nos cursos de MAC, veja mais detalhes em https://panda.ime.usp.br/panda/python).

O exercício pode ser feito em duplas, sempre com alguém da mesma turma.

Apenas um aluno deve entregar o exercício, destacando no relatório e código o nome de ambos os alunos.

A entrega deve conter o relatório (em .pdf), contendo a análise do problema estudado, e o código usado para as simulações computacionais (arquivos .py). A entrega pode ser feita em um arquivo compactado único.

O seu código deve estar bem comentado e estruturado. A entrada e saída devem ser feitas de forma a ajudar o usuário a executar o programa e deve facilitar a análise dos resultados. Inclua qualquer arquivo adicional necessário para o seu programa no arquivo compactado a ser entregue.

O relatório deve conter, pelo menos: (i) Uma breve introdução ao assunto estudado. (ii) Uma descrição do método e como foi implementado. (iii) Resultados dos testes descritos abaixo (tanto dos testes iniciais quanto dos da aplicação).

Você deve implementar o exercício programa em Python3.x

Pode usar:

- Matplotlib - para fazer gráficos
- NumPy - apenas para trabalhar com aritmética de vetores, leitura/escrita de arquivo csv. As funções RFFT e IRFFT também devem ser usadas para comparar com as suas próprias implementações.
- Bibliotecas auxiliares: sys, datetime, os, math.

Não pode usar:

- SciPy
- Funções do NumPy diversas das explicitadas acima.

1 Transformadas de Fourier

Consideremos uma função $F(x)$ 2π-periódica tabelada em $2N$ pontos uniformemente espaçados em $[0, 2\pi]$. Denotemos os pontos por $x_j = j\pi/N, j = 0, 1, .., 2N-1$. Os valores de $F(x_j)$ podem ser reais ou complexos. Na descrição que segue iremos considerar o caso mais geral e caso $F(x)$ seja real, consideramos que sua parte imaginária é nula.

A transformada discreta de Fourier de $F(x)$ associa aos valores $(F(x_0), F(x_1), ..., F(x_{2N-1}))$ os coeficientes de Fourier de F: $(a_{1-N}, a_{2-N}, ..., a_N)$ tais que:

$$F(x_j) = \sum_{k=1-N}^{N} a_k e^{ikx_j} \quad , j = 0, .., 2N-1 \tag{1}$$

No caso em que F é uma função real, os coeficientes a_k e a_{-k} são complexos conjugados ($a_k = \alpha_k + i\beta_k$ e $a_{-k} = \overline{a_k} = \alpha_k - i\beta_k$), de tal forma que

$$a_{-k}e^{-ikx_j} + a_k e^{ikx_j} = 2(\alpha_k \cos(kx_j) - \beta_k \sin(kx_j)) \tag{2}$$

representa o harmônico de ordem k de $F(x)$, com amplitude $A_k = 2\sqrt{\alpha_k^2 + \beta_k^2} = 2|a_k|$. Os coeficientes a_0 e a_N têm parte imaginária nula quando F é real.

O cálculo da transformada de Fourier é feito através da expressão:

$$a_k = \frac{1}{2N} \sum_{j=0}^{2N-1} F(x_j) e^{-ikx_j}, \quad k = 1-N, .., N \tag{3}$$

enquanto que a transformada inversa, através da qual se recupera os valores de $F(x_j)$ a partir dos coeficientes a_k, é dada por (1).

Note que se definirmos o produto interno complexo $< f, g >= \frac{1}{2N} \sum_{j=0}^{2N-1} f(x_j)\overline{g(x_j)}$, temos que as funções e^{ikx}, $k = 1-N, N$ formam um conjunto ortonormal em relação a ele e que $a_k =< F(x), e^{ikx} >$.

Observação: Note que

$$e^{-ikx_j} = e^{-ikj\pi/N} = e^{i2\pi j} e^{-ikj\pi/N} = e^{i(2N-k)j\pi/N} = e^{i(2N-k)x_j} \tag{4}$$

e portanto podemos alternativamente escrever as transformadas (1) e (3) como

$$F(x_j) = \sum_{k=0}^{2N-1} c_k e^{ikx_j} \quad , j = 0, .., 2N-1 \tag{5}$$

e

$$c_k = \frac{1}{2N} \sum_{j=0}^{2N-1} F(x_j) e^{-ikx_j}, \quad k = 0, ..., 2N-1 \tag{6}$$

com $c_k = a_k$, $k = 0, 1, .., N$ e $c_{2N-k} = a_{-k}, k = 1, ..., N-1$.

2 Transformada Rápida de Fourier (FFT)

Note que o cálculo das transformadas (direta) de Fourier (5) e sua inversa (6) requerem cada um a avaliação de $2N$ somas, cada uma com $2N$ elementos. Assim, o cálculo destas transformadas requer $O(N^2)$ operações, se feito diretamente usando estas expressões. Em 1965, Cooley e Tukey (Cooley, James W.; Tukey, John W. (1965), An algorithm for the machine calculation of complex Fourier series, Math. Comput. 19, 297-301) propuseram um algoritmo, mostrando como calcular transformadas de Fourier com apenas $O(N \log N)$ operações, trazendo grande progresso a várias aplicações computacionais. Embora as idéias envolvidas no algoritmo remontem a Gauss (veja por exemplo Heideman, M. T., D. H. Johnson, and C. S. Burrusi (1984), Gauss and the history of the fast Fourier transform, IEEE ASSP Magazine 1, (4), 14-21), foi apenas após o trabalho de Cooley e Tukey que o método realmente se efetivou.

A efetividade do algoritmo depende de $2N$ ser fatorável em vários fatores primos. Vamos aqui nos restringir ao caso em N é uma potência de 2, embora haja diversas implementações muito eficientes para transformadas de comprimento do tipo $2^l 3^m 5^n 7^p$. Vamos descrever o método para a transformada inversa (5) no caso em que $N = 2^l$. Temos que:

$$\begin{aligned} F(x_j) &= \sum_{k=0}^{2N-1} c_k e^{ikx_j} \\ &= \sum_{k=0}^{N-1} c_{2k} e^{i2k\pi j/N} + \sum_{k=0}^{N-1} c_{2k+1} e^{i(2k+1)\pi j/N} \\ &= \sum_{k=0}^{2(N/2)-1} c_{2k} e^{ik\pi j/(N/2)} + e^{i\pi j/N} \sum_{k=0}^{2(N/2)-1} c_{2k+1} e^{ik\pi j/(N/2)} \\ &= \text{Even}_j + e^{i\pi j/N} \text{Odd}_j \ , \ j = 0, .., N-1 \end{aligned} \tag{7}$$

e

$$\begin{aligned} F(x_{j+N}) &= \sum_{k=0}^{2N-1} c_k e^{ikx_(j+N)} \\ &= \sum_{k=0}^{N-1} c_{2k} e^{i2k\pi(j+N)/N} + \sum_{k=0}^{N-1} c_{2k+1} e^{i(2k+1)\pi(j+N)/N} \\ &= \sum_{k=0}^{2(N/2)-1} c_{2k} e^{ik\pi j/(N/2)} + e^{i\pi(j+N)/N} \sum_{k=0}^{2(N/2)-1} c_{2k+1} e^{ik\pi j/(N/2)} \\ &= \text{Even}_j - e^{i\pi j/N} \text{Odd}_j \quad , \ j = 0, .., N-1. \end{aligned} \tag{8}$$

Vemos então que as duas somas (cada uma com 2N termos) para a avaliação de $F(x_j)$ e $F(x_{j+N})$ requerem apenas a avaliação de duas somas ($Even_j$ e Odd_j), cada uma com N termos (e mais duas adições e uma multiplicação). Note ainda que estas duas somas são do mesmo tipo que a inicial e se N é par, podem novamente ser reduzidas a somas com metade dos termos. A aplicação recursiva desta ideia, leva à transformada rápida, com $O(N \log N)$ operações (onde $\log N$ é o número de etapas em que se reduz o tamanho das somas pela metade).

De mesma maneira pode-se calcular a transformada direta (6) (deixando a divisão por $2N$ para o final):

$$\begin{aligned} c_k &= \sum_{j=0}^{2N-1} F(x_j) e^{-ik\pi j/N}, \quad k = 0, ..., 2N-1 \\ &= \sum_{j=0}^{N-1} F(x_{2j}) e^{-ik\pi j/(N/2)} + e^{-ik\pi/N} \sum_{j=0}^{N-1} F(x_{2j+1}) e^{-ik\pi j/(N/2)} \\ &= \text{Even}_k + e^{-ik\pi/N} \text{Odd}_k \ , \ k = 0, .., N-1 \end{aligned} \tag{9}$$

e

$$c_{k+N} = \text{Even}_k - e^{-ik\pi/N} \text{Odd}_k \ , \ k = 0, .., N-1 \tag{10}$$

Detalhamos um pouco mais como implementar um procedimento recursivo para o cálculo das transformadas na próxima seção.

FFT recursiva

Um pseudo código para fft recursiva (com $N = 2^n$) é dado por

fftrec(c,f,n,dir)

complexos f(0:2n-1),c(0:2n-1),even(0:n-1),odd(0:n-1),fe(0:n-1),fo(0:n-1)

```
logico dir

Se n=1 então

    c(0) = f(0)+f(1)
    c(1) = f(0)-f(1)

senão

    para j=0,n-1
        fe(j)=f(2j)
        fo(j)=f(2j+1)
    fim do para
    fftrec(even,fe,n/2,dir)
    fftrec(odd ,fo,n/2,dir)
    para j=0,n-1
        se (dir) então
            eij = exp(- i * j * pi / n)
            senão
            eij = exp(i * j * pi / n)
        fim do se
        c(j) = even(j)+eij * odd(j)
        c(j+n) = even(j)-eij * odd(j)
    fim do para

fim do se
```

Observações: As variáveis no código acima são complexas. O mesmo programa serve tanto para a transformada direta como para a transformada inversa de Fourier, sendo cada caso controlado pela variável lógica dir. Se dir é verdadeira, a transformada direta é executada e caso contrário, executa-se a transformada inversa. No caso da transformada direta, ao final da execução ainda deve-se dividir os coeficientes c_k pelo fator de normalização $2N$.

Análise de Fourier e filtragem

Dados os valores de uma função F em pontos uniformemente espaçados usamos as transformadas de Fourier para fazer sua decomposição em harmônicos, correpondentes às diversas frequências. Em nossa descrição sobre a transformada de Fourier, supusemos um período 2π. Sendo o período igual a T, temos apenas que fazer uma mudança de variáveis. Assim a k-ésima frequência associada a um número de onda k é dada por $\omega_k = k/T$. Note que a amplitude do harmônico de ordem k é dada por $A_k = 2|a_k| = 2|c_k| = 2|c_{2N-k}|, k = 1, .., N-1$, quando temos $2N$ pontos amostrados. A N-ésima frequência $\omega_N = N/T$ é a máxima frequência que pode ser resolvida com esse número de pontos, e é chamada de frequência de Nyquist (com amplitude do harmônico igual a $|a_N| = |c_N|$). Calculando a transformada de Fourier de F, podemos então verificar, por exemplo, quais são suas frequências dominantes. Podemos também aplicar uma filtragem seletiva das frequências.

Um filtro consiste de um corte (ou redução) de algumas frequências da função em questão (um sinal ou série temporal, por exemplo). O filtro é do tipo "passa baixa" quando ele deixa passar frequências baixas (ou seja, as mantém sem alterações) e elimina as frequências altas. A aplicação de um tal filtro zera os coeficientes de frequências mais altas, a partir de uma dada frequência de corte K (ou seja, zeramos os coeficientes dos harmônicos de ordem $k > K$, que são $c_k,\ k = K+1, .., 2N-K-1$). A seguir, através da transformada inversa de Fourier, reconstitui-se a função (filtrada) com as frequências altas eliminadas.

Analogamente, temos filtros "passa alta", em que frequências abaixo de um valor de corte K são zeradas, mantendo as frequências altas inalteradas. O coeficiente c_0 referente ao termo constante não é zerado, caso se deseje a preservação da amplitude média do sinal.

Temos ainda filtros “passa banda”, que mantêm apenas as frequências dentro de uma certa faixa $K_1 < k < K_2$, zerando as restantes.

Note que a frequência ω_k (associada ao coeficiente c_k) tem k ciclos completos de oscilação no período completo T. Portanto o período P_k de uma oscilação relativa à frequência ω_k será dado por $P_k = 1/\omega_k = T/k$.

Tarefas iniciais e verificação

Você deverá implementar as transformadas diretas e inversas de Fourier, tanto em sua forma mais direta (e menos eficiente), como descrito nas equações (5) e (6), como através da FFT recursiva. Através dos testes e da aplicação das transformadas posteriormente, você irá comprovar a significativa diferença de eficiência computacional entre as duas versões. Você deverá confrontar ainda a eficência de sua implementação com a versão das transformadas de Fourier pertencentes à biblioteca numpy (rotinas rfft e irfft). O ”r”nos nomes das rotinas se refere ao fato de que as funções a serem transformadas são reais. Devido à anti-simetria dos coeficientes neste caso, pode-se economizar memória e tempo computacional. Mais ainda, estas implementações não fazem uso de funções recursivas. Apesar de por vezes facilitar a implementação de algoritmos, o uso de funções recursivas costuma ser pouco eficiente em comparação com códigos que acabam por resolver a recursão. Para entender bem isto você pode calcular $n!$ através de uma função recursiva (se $n = 1$, $fat(n) = 1$, senão $fat(n) = n * fat(n-1)$) ou através de um loop ($fatn = 1$, para i de 2 a n faça $fatn = fatn * i$) e comparar. Você deve comparar os tempos computacionais das 3 versões das transformadas que irá testar.

Veja a documentação das rotinas do numpy! Você irá notar que os resultados da transformação direta não são normalizados ao final (falta a divisão por $2N$). A normalização é utilizada na transformada inversa. Note este fato na hora de comparar resultados.

Execute os seguintes testes para validar suas implementações das transformadas.

a) considere a função com os seguintes valores em $0, \pi/2, \pi$ e $3\pi/2$: (5,-1,3,1). O resultado da transformada direta deve ser: $c = (2, 1/2 + i/2, 2, 1/2 - i/2)$. Aplicando a transformada inversa deve-se recuperar os valores da função original.

b) considere a função com os seguintes valores $f = (6, 2, 5, 2, 11, 2, 8, 8)$ e compare seus códigos com as rotinas do numpy.

c) Seja $F(x) = 10sin(x) + 7cos(30x) + 11sin(352x) - 8cos(711x)$. Construa o vetor com os valores $F(x_j)$, com $x_j = j\pi/512, j = 0, 1023$ e calcule a transformada direta. Interprete os valores obtidos. Aplique a transformada inversa para a recuperação de F.

Tratamento de dados incompletos

Nas aplicações você deverá transformar funções que foram medidas em instantes (ou posições) uniformemente espaçados. No entanto, você pode ter que lidar com duas situações:

a) É possível que os dados sejam incompletos, com lacunas no meio. Por exemplo, se são dados diários, pode ser que faltem informações referentes a alguns dias. Neste caso, a tabela deverá ser completada através de interpolação cúbica (use por exemplo o método de Lagrange). Tome dois valores anteriores e dois posteriores mais próximos ao ponto desejado e interpole para a posição desejada. Caso exista apenas um ponto de um dos lados, você pode trocar para interpolação linear neste caso.

b) Caso o número de valores tabelados (após as interpolações para as lacunas) não seja uma potência de 2, use o valor médio destes para completar a tabela até a próxima potência de 2. (se o número de elementos for ligeiramente superior a uma potência de 2, você poderia também descartar os pontos que estão a mais).

3 Oscilações Econômicas: introdução

A modelagem de preços de ativos, como ações e *commodities* (café, metais, petróleo, grão, etc), é um tema complexo que sofre interferência de diversos fatores. Independentemente dos fatores que influenciam os preços, as séries históricas de preços de ativos geralmente possuem variações de curto prazo (pequenas oscilações diárias, por exemplo) e de longo prazo (variações sazonais ou anuais, por exemplo). O objetivo deste exercício programa é a análise de séries históricas de ativos e *commodities* do ponto de vista de sua decomposição espectral, isto é, a análise das diversas oscilações presentes nas séries. Para isso usaremos como ferramenta básica a Transformada de Fourier.

Análise de oscilações

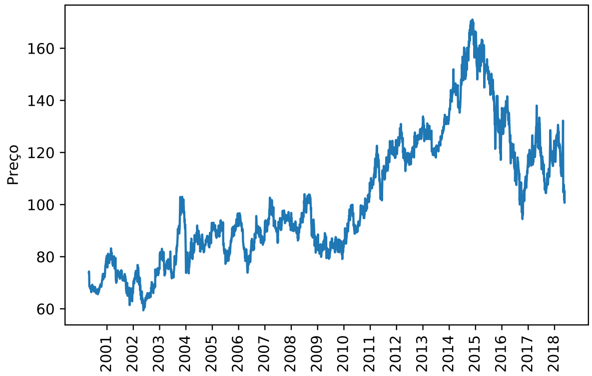

Figura 1: Evolução do preço diário do Boi Gordo desde 20/04/2000 até 21/05/2018, em US$.

Vamos analisar um exemplo. Na Figura 3 temos a série histórica do preço diário do Boi Gordo obtido da bolsa americana considerando o período de 20/04/2000 até 21/05/2018 (`www.investing.com`), totalizando 6602 dias (cerca de 18 anos) - este será o nosso T.

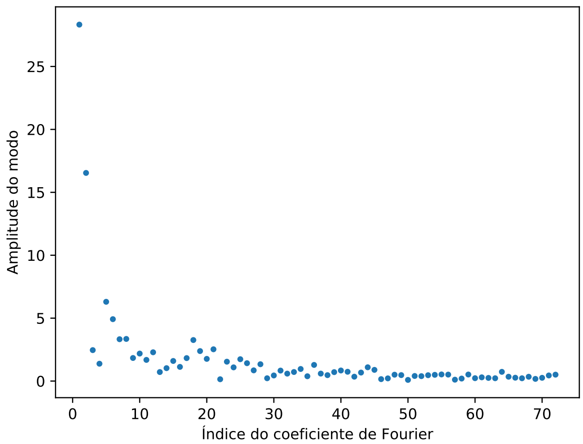

Figura 2: Espectro de Fourier dado pelo valor absoluto dos coeficientes de Fourier (amplitude dos harmônicos) para o preço diário do Boi Gordo. O coeficiente de índice zero não foi apresentado na figura para facilitar a visualização das demais frequências que tem ordem de grandeza menores. Os índices maiores que 75 também não foram apresentados nessa figura para facilitar a visualização dos termos de frequências baixas.

Aparentemente é difícil analisar ciclos de oscilações desta série, mas podemos ver no espectro de (Figura 2) as frequências dominantes. Note que as frequências de números 1 e 2 dominam, pois são ciclos muito longos de 18 e 9 anos, capturando uma tendência de longo prazo da série. Posteriormente, temos um pico no índice 5. Esse pico no índice 5 indica uma oscilação forte com período de cerca de 3.6 anos. Perceba também que há um leve aumento no índice 18, relativo a uma oscilação de período $6602/18 \approx 366$, ou 1 ano. É de se esperar que haja uma certa sazonalidade anual nos preços de ativos, por conta da sazonalidade natural induzida pelos eventos anuais.

Se considerarmos como oscilações longas quaisquer oscilações com períodos maiores que 6 meses podemos definir uma frequência de corte de $6602/180 \approx 36$. Zerando as frequências com índices mais altos que 36, e depois reconstruindo a série temporal apenas com oscilações de períodos longos, obtemos a curva apresentada na Figura 3.

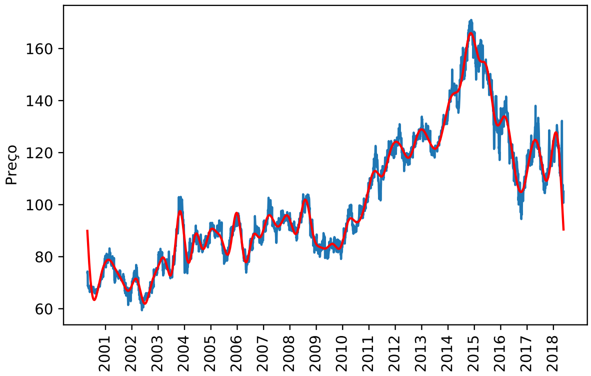

Figura 3: Evolução do preço diário do Boi Gordo desde 20/04/2000 até 21/05/2018, em US$. Em vermelho é apresentada a curva de preços reconstruída a partir de oscilações lentas (índices de Fourier menores que 36).

Podemos analisar os detalhas da oscilação anual reconstruindo apenas o modo espectral relativo ao período de 1 ano (neste caso, de índice 18). Na figura 4 mostramos a oscilação deste modo. Veja que anualmente temos um período de alta no começo do primeiro semestre de cada ano e um período de baixa no começo de cada segundo semestre. Através da série observamos que realmente há periodicamente pequenas altas no começo do primeiro semestre de cada ano.

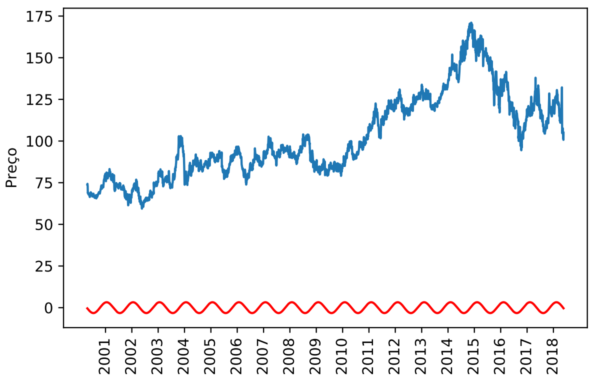

Figura 4: Evolução do preço diário do Boi Gordo desde 20/04/2000 até 21/05/2018, em US$. Em vermelho é apresentada a curva de preços reconstruída a partir de oscilações com período de 1 ano (índice de Fourier 18).

Na figura 5 apresentamos a reconstrução usando apenas as frequências mais altas, com períodos menores que 6 meses. Note na figura que há saltos no início e fim da série, o que se deve à não periodicidade da série original. Esses saltos iniciais e finais podem ser desconsiderados para análise das oscilações rápidas, se desejado.

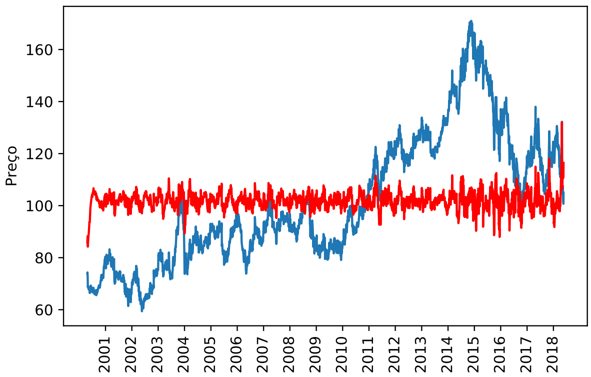

Figura 5: Evolução do preço diário do Boi Gordo desde 20/04/2000 até 21/05/2018, em US$. Em vermelho é apresentada a curva de preços reconstruída a partir de oscilações rápidas (índices de Fourier maiores que 36).

Volatilidade

A volatilidade é uma medida de dispersão dos retornos de um índice. Há muitas formas de se medir a volatilidade, mas a grosso modo ela deve medir a variabilidade do preço do ativo e fornecer uma medida que possa ser usada associada ao risco de investimento no ativo. Assim, se a volatilidade for alta, o risco é maior, se for baixa, o risco é menor.

Uma métrica popularmente usada para o cálculo de volatilidade é a variação média ao longo de um período determinada, por exemplo, pelo desvio padrão temporal sobre o período,

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - \bar{p})^2},$$

onde n indica o número de dias analisados e $\bar{p} = \sum_{i=1}^{n} p_i$ é a média do preço neste período.

Se calcularmos a volatilidade com os dados originais, o desvio padrão será influenciado pela tendência da série e seu comportamento sazonal. É interessante obtermos a volatilidade medida apenas pela variação local do preço do ativo, e para tanto podemos calcular a volatilidade com base na série filtrada, contendo apenas modos de período mais curtos.

No exemplo do Boi Gordo a volatilidade estimada via desvio padrão considerando os 30 dias mais atuais com a série original é de US$ 9.5, ou, percentualmente considerando o preço médio do período, de 8.5%. Retirando as flutuações sazonais, incluindo apenas frequências relativas a oscilações com períodos menores que 6 meses, temos uma volatilidade de US$ 4.5, ou 4.6%. Portanto, em média, as variações recentes de preço do Boi Gordo giram em torno de 4.6%.

4 Desafios

Para chegar às interpretações discutidas na seção anterior tivemos que superar alguns desafios, que descreveremos a a seguir.

Interpolações

Tivemos que completar as séries históricas, uma vez que estas têm dados faltantes. Por exemplo, em séries do mercado financeiro não temos valores nos finais de semana e feriados. Se simplesmente juntarmos os dados ignorando os buracos, a nossa noção de tempo e períodos de oscilações seria afetada, portanto precisamos de alguma forma preencher os buracos. Veja na Figura 6 um exemplo de como os dados tem “buracos”. Para preenchê-los usamos interpolação como já descrito anteriormente.

Tamanho da série

Para a obtenção de uma potência de 2 podemos cortar dados ou adicionar valores ao final (por exemplo, valores iguais ao valor médio do ativo no período).

5 Sua Tarefa

Dados

Fornecemos um conjunto de dados para as análises propostas neste exercício. Os arquivos estão no formato .csv, o que significa que temos dados separados por vírgulas. No exemplo do Boi Gordo o início do arquivos de dados é como o apresentado na Figura 6.

Coluna 1: Data, no formado ANO-MÊS-DIA, sempre com 8 caracteres

Coluna 2: Índice que conta o número de dias desde o início da série. No exemplo da Figura 6 o início se deu em 20/04/2000, portanto, na data mais recente da série, o dia 21/05/2018, estaremos 6601 dias da data de início. O índice da data de início é sempre zero, por isso temos na verdade 6602 dias no total.

Coluna 3: Valor em dólares ou reais (dependendo do ativo) para aquele dia específico.

Note que nos dados apresentados aqui como exemplo temos valor no dia 21/05/2018, mas não temos em 20/05/2018 e 19/05/2018, pois tivemos um fim de semana. Sendo assim, apesar do arquivo original Cattle.csv fornecido ter apenas 4571 linhas (ou seja, 4571 dias), o seu programa deve interpolar os dias faltantes para que tenhamos a série completa com os 6602 dias que existem entre a data do valor mais antigo e do mais recente.

Entrada/Saída do EP

Entrada do EP:

Nome do arquivo contendo os dados no formato .csv, no formato descrito acima.

Parâmetro de corte para os filtros (K)

Opção do tipo de transformada de Fourier a ser usada (Normal, Rápida Recursiva, Rápida Fornecida no Numpy).

Período para cálculo da volatilidade.

O que deve ser implementado:

Método de interpolação cúbica para preenchimento dos dados faltantes.

Transformada rápida de Fourier recursiva.

Transformada normal de Fourier com N^2 operações.

Filtros passa alta/baixa.

Cálculo de volatilidade com base no resultado do filtro passa alta.

```
20180521, 6601, 105.000000
20180518, 6598, 102.400000
20180517, 6597, 103.050000
20180516, 6596, 101.825000
20180515, 6595, 102.975000
20180514, 6594, 104.625000
20180511, 6591, 107.625000
20180510, 6590, 107.525000
20180509, 6589, 105.675000
20180508, 6588, 106.300000
20180507, 6587, 105.175000
20180504, 6584, 106.050000
20180503, 6583, 106.525000
20180502, 6582, 104.850000
20180501, 6581, 105.825000
20180430, 6580, 123.750000
20180427, 6577, 124.450000
20180426, 6576, 122.525000
20180425, 6575, 121.925000
20180424, 6574, 121.100000
20180423, 6573, 121.725000
20180420, 6570, 119.350000
20180419, 6569, 117.800000
20180418, 6568, 118.975000
20180417, 6567, 118.050000
20180416, 6566, 117.150000
20180413, 6563, 116.550000
20180412, 6562, 115.750000
20180411, 6561, 113.500000
20180410, 6560, 112.800000
20180409, 6559, 112.025000
20180406, 6556, 112.225000
20180405, 6555, 114.300000
-:---  Cattle.csv     Top L1
```

Figura 6: Exemplo de dados do preço diário do Boi Gordo, em US$.

Saída do EP:

Arquivos análogos ao de entrada com os dados filtrados (com passa alta e baixa) relativos ao parâmetros de corte (.csv).

Arquivo com 3 colunas, contendo na primeira coluna o número de onda k do harmônico de Fourier, na segunda o coeficiente complexo c_k, e na terceira o valor absoluto desse número complexo (amplitude do harmônico), com cada coluna separada da outra por vírgula (formado .csv).

Valores dos 10 coeficientes de Fourier mais significativos, indicando para cada um o período de oscilação.

Tempo gasto no cálculo das transformadas.

No caso de dados de entrada que não são potências de 2 você deve:

Para o algoritmo recursivo implementado: Ajustar o tamanho dos dados para potência de 2 usando corte de dados ou completamento com a média.

Algoritmo "Lento" (N^2): Não ajustar o tamanho da série.

Algoritmo FFT do numpy/Python: Não ajustar o tamanho da série.

Testes

Você deve montar uma relatório explorando o conjunto de dados fornecidos, apresentando gráficos e discutindo os resultados. Você pode explorar dados de outras fontes se desejar (veja por exemplo os dados de `www.investing.com` ou da base histórica da BMF-BOVESPA [1]).

Testes obrigatórios:

1. Dados sem buracos com tamanho sendo potência de 2. Analise os dados Alum.csv (Alumínio em US$).
2. Dados sem buracos com tamanho que não é potência de 2. Analise os dados Wheat.csv (Trigo em US$) que possuem um número primo de dias!
3. Dados com buracos e tamanho arbitrário.

    Analise os dados de commodities (obtidas em `www.investing.com` em US$) : Rice.csv (Arroz), Copper.csv (Cobre).

    Analise os dados de preços de ações na BMF/BOVESPA: CAML3.csv (CAMIL em R$), ELET3.csv (Eletrobras), SBSP3.csv (SABESP).

Alguns tópicos a serem discutidos no relatório:

Características do espectro, considerando oscilações dominantes de curto e longo período.

Características dos filtros passa alta e passa baixa. Com base nas séries filtradas é possível prever algum padrão futuro para o ativo?

Volatilidade e risco do ativo.

Desempenho (tempo computacional) de cada método implementado frente aos dados analisados.

Para as análises acima escolha o parâmetros de corte do filtro com base nas características da série e de seu espectro.

[1] `http://www.bmfbovespa.com.br/pt_br/servicos/market-data/historico/mercado-a-vista/cotacoes-historicas/`