



API dos módulos Velc

Criadores : Cleiton da Silva Gomes
Vanessa Barbosa Martins

1. ArquivoForex

1.1 stringFromFloat

stringFromFloat :: [String] -> [Float]

Faz o parse de uma lista de Strings para uma lista de números flutuantes

1.2 obtemTipoGrafico

obtemTipoGrafico :: IO String

Lê o arquivo que contém os dados do robô e recupera o tipo de gráfico que o robô analisa. Pode haver três retornos : "M1", "M5", "H1".

1.3 lerCotacoes

lerCotacoes :: IO [Float]

Retorna uma lista com números flutuantes que foram pegos no arquivo. o Arquivo que é lido depende do tipo de gráfico que o robô analisa.

1.4 detectaQuantidadeCandle

detectaQuantidadeCandle :: IO Int

Lê o arquivo que contém os dados do robô e recupera a quantidade de *candles* que o robô usará para os cálculos , ou seja, a quantos preços serão usados para análise

1.5 cotacoes

cotacoes :: [Float]

Retorna uma lista de números flutuantes, com a quantidade exata de termos que o robô necessita para fazer suas análises (*candles*).



2. CorrelacaoDePearson

2.1 calculaMedia

calculaMedia :: Fractional a => [a] -> a

Faz o cálculo da média de uma lista que possui elementos do tipo ponto flutuante.

2.2 vetorX

vetorX :: [a] -> [a]

Cria um vetor com os número de posição pares do vetor original que foi passado.

2.3 vetorY

vetorY :: [a] -> [a]

Cria um vetor com os número de posição impares do vetor original que foi passado.

2.4 covarianciaPearson

covarianciaPearson :: Num a => [a] -> [a] -> a -> a

Faz o cálculo da co-variância de Pearson , que a é dada pela equação:

$$\sum_{i=1}^n (x_i - \bar{x}) * (y_i - \bar{y})$$

2.5 varianciaPearson

varianciaPearson :: Floating a => [a] -> a -> a

Faz o cálculo da variância de Pearson , que a é dada pela equação:

$$\sum_{i=1}^n (x_i - \bar{x})^2$$

2.6 correlacaoDePearson

correlacaoDePearson :: (Floating a, Monad m) => [a] -> m a

Calcula a correlação de Pearson que é dado por $\frac{co-variância(X,Y)}{\sqrt{variância(X) * variância(y)}}$

3. Fibonacci

3.1 verificaCotacaoListaFibonacci

verificaCotacaoListaFibonacci :: (Eq a, Num a) => a -> Boo

Verifica se um número pertence à sequência de Fibonacci

3.2 encontrarFibonacciAnterior

encontrarFibonacciAnterior :: (Eq a, Num a) => a -> a

Busca o número mais próximo na sequência de Fibonacci, dado o parâmetro.



4. MediaMovel

4.1 mediaMovelInferior

mediaMovelInferior :: Float

Calcula a média do vetor considerando considerando como tamanho o número anterior a quantidade de *candles*, desde que seja um número de fibonacci.

4.2 mediaMovelSuperior

mediaMovelSuperior :: Float

Calcula a média da lista de cotações, de tamanho *através da quantidade de candles*.

5. MetodosNumericos

5.1 main

main :: IO ()

Escreve em arquivo o resultado dos métodos numéricos : correlação de Pearson, Média Móvel Inferior, Média Móvel Superior.