Trabalho final: Função NO QUE INVESTIR

Arquivos

Código da função: funcao_final.r

Página de ajuda: help_final.txt

Arquivos para os exemplos:

1. Exemplo 1: planilha_1.csv
2. Exemplo 2: planilha_2.csv

Código da função

```
no_que_investir <- function(dataframe) #cria uma funcao cujos parametros sao
colunas da planilha
{ #inicio da funcao
  ##### Verificacao dos parametros
  if (any(is.na(planilha$investimento)==TRUE)) #verifica se em alguma linha
da coluna "investimento" da planilha inserida ha algum NA
  {
    stop ("A coluna investimento nao esta completamente preenchida.") #Se
sim, interrompe a funcao e exibe a mensagem para o usuario
  }
  if (any(is.na(planilha$quantidade_acoes)==TRUE)) #verifica se em alguma
linha da coluna "quantidade_acoes" da planilha inserida ha algum NA
  {
    stop ("A coluna quantidade_acoes nao esta completamente preenchida.")
#Se sim, interrompe a funcao e exibe a mensagem para o usuario
  }
  if (sum(planilha$objetivo) != 1)  #verifica se a soma de todos os valores
inseridos na coluna "objetivo" eh igual a 1
  {
    stop ("A soma dos objetivos deve ser igual a 1.") #Se nao, interrompe a
funcao e exibe a mensagem para o usuario
  }
  ##### abre o pacote do qual a funcao eh dependente
  require(GetHFData) #abrir o pacote GetHFData dentro da funcao
  ##### selecao das informacoes da planilha importada que sao de interesse
  renda_variavel <- data.frame(planilha$investimento,
planilha$quantidade_acoes, planilha$objetivo) #cria um data frame chamado
"renda_variavel" com as colunas da planilha que interessam para essa analise
  colnames(renda_variavel) <- c("investimento", "quantidade_acoes",
"objetivos") #corrige os nomes das colunas do data frame recem criado
  ##### processo de busca das cotacoes. Inicialmente cria-se um vetor cheio
```

de NAs cujo tamanho corresponde ao numero de colunas do data frame que, por sua vez, corresponde ao numero de empresas das quais queremos as cotacoes. Em seguida, o R consultara um repositorio da Bovespa (ftp://ftp.bmf.com.br/MarketData/), encontrara os valores, minuto a minuto, da cotacao de cada empresa no intervalo de dias indicado (nesse caso, 5), selecionara o ultimo valor disponivel (presente na ultima linha da tabela resultante, que eh o mais recente possivel) e o salvara na posicao correspondente no vetor que estava preenchido com NAs.

  acoes_cotacao <- rep(NA, nrow(renda_variavel)) #cria um vetor chamado "acoes_cotacao" cheio de NAs cujo tamanho equivale ao numero de linhas do data frame "renda_variavel"

  for(i in 1:nrow(renda_variavel)) #abre um ciclo "for" com um contador i que vai de 1 ate o numero de linhas do data frame "renda_variavel"

  {

    cotacao <- ghfd_get_HF_data(my.assets = renda_variavel[i, 1], #inicio do processo de busca das cotacoes das empresas. Em cada ciclo serah feita a busca para as empresas presentes na coluna investimentos do data frame "renda_variavel". Como resultado serao criadas tabelas que serao sobrescritas no objeto cotacao.

                                type.market = 'equity', #o tipo de dados que serao procurados no repositorio

                                first.date = Sys.Date() - 5, #o primeiro dia que serah procurado no repositorio. Eu coloquei uma margem de dias grande para que finais de semana e feriados nao inviabilizem o processo. Nesse caso, a data mais antiga que serah consultada eh de 5 dias anteriores ao dia atual.

                                last.date = Sys.Date() - 1, #a data mais recente que serah procurada no repositorio. Como os dados sao depositados nesse repositorio com atraso de um dia, as informacoes mais recentes sao de 1 dia anterior em relacao ao dia atual

                                first.time = '10:00:00', #o primeiro horario do dia que serah procurado no repositorio. Eu coloquei o horario de inicio de negociacao no mercado de acoes da Bovespa

                                last.time = "17:00:00", #o ultimo horario do dia que serah procurado no repositorio. Eu coloquei o horario aproximado de encerramento de negociacao no mercado de acoes da Bovespa

                                type.output = 'agg', #agrega apenas as informacoes encontradas pelos parametros inseridos. Se nao usar essa opcao, o output apresentarah uma serie de outras informacoes que nao importam para o objetivo dessa funcao

                                agg.diff = '1 min') #o intervalo de tempo em que serao procurados os valores das acoes. Nesse caso, serah feito minuto a minuto nos dias selecionados

    acoes_cotacao[i] <- cotacao[nrow(cotacao), 5] #cria um vetor em que eh salvo o ultimo valor da cotacao de cada empresa no dia selecionado (o ,5 indica que o valor deve ser retirado da quinta coluna da tabela gerada, na qual estao inseridos os valores da cotacao).

  }

  ##### Manipulacao e organizacao dos dados

  renda_variavel <- data.frame(renda_variavel, acoes_cotacao) #acrescenta o

```
vetor "acoes_cotacao" como uma nova coluna do data frame "renda_variavel"
  acoes_valor <-
renda_variavel$quantidade_acoes*renda_variavel$acoes_cotacao #cria um vetor
chamado "acoes_valor" em que sao salvos os resultados da multiplicacao do
valor da cotacao pela quantidade de acoes de cada empresa
  acoes_total <- sum(acoes_valor) #Soma o total investido em acoes e o salva
no objeto acoes_total
  acoes_proporcao <- acoes_valor/acoes_total #Divide cada valor presente no
vetor acoes_valor por acoes_total e salva o resultado em um vetor chamado
acoes_proporcao.
  ##### Criacao do grafico
  plot(x=NULL,y=NULL, #gera um plot vazio
       xlim=round(c(0, nrow(renda_variavel)+1), 0), #os valores do eixo x
variam de 0 ao numero de linhas do data frame "renda_variavel" mais um (para
que a ultima empresa nao fique muito na borda do grafico), sem nenhuma casa
decimal (para que cada empresa seja posicionada acima de um numero inteiro)
       ylim= c(0,1), #os valores do eixo y variam de 0 a 1 (o valor maximo
possivel, ja que estamos tratando de proporcoes e a soma de todas elas
equivale a 1)
       bty = "l", #o grafico possui apenas as margens inferior e esquerda
       xaxt ="n", #nao coloca numeros no eixo x, porque quero que no eixo x
estejam presentes os codigos das acoes
       xlab="Acoes", #titulo do eixo x
       ylab = "Proporcao do capital") #titulo do eixo y
  axis(1, at = 1:nrow(renda_variavel), labels = renda_variavel$investimento)
#adiciona os codigos das acoes de cada empresa na sua posicao correspondente
do eixo x
  for(i in 1:nrow(renda_variavel)) #entra em um ciclo for para adicionar
segmentos cujo tamanho seja determiando pela distancia entre a proporcao do
capital investido em uma determinada empresa e o objetivo do patrimonio que
nela deveria estar alocado
  {
    segments(x0=i, y0=renda_variavel[i,3], x1= i, y1=acoes_proporcao[i], col
= "red") #fornece os pontos do grafico em que uma linha vertical de cor
vermelha devera ser tracada. Essa linha estara posicionada exatamente acima
da posicao do eixo x na qual o codigo da acao foi inserido
  }
  points(renda_variavel$objetivos, #insere pontos no grafico que indicam a
proporcao do capital que o usuario deseja ter alocado em cada empresa (o
objetivo)
         col = "blue", #esses pontos terao a cor azul
         pch = 19, #esses pontos serao bolinhas totalmente preenchidas
         cex = 1.3) #esses pontos serao 1.3 vezes maiores do que o padrao
  points(acoes_proporcao, #insere pontos no grafico que indicam a proporcao
do capital que efetivamente esta investido em cada empresa
         col = "green", #esses pontos terao a cor verde
         pch = 19, #esses pontos serao bolinhas totalmente preenchidas
         cex = 1.3) #esses pontos serao 1.3 vezes maiores do que o padrao
  subtracao <- acoes_proporcao - renda_variavel$objetivos #criacao de um
vetor chamado subtracao no qual estara inserido o resultado da subtracao da
proporcao investida em cada empresa pelo objetivo
```

```
  for(i in 1:nrow(renda_variavel)) #entra em um ciclo for para plotar no
grafico o resultado dessa subtracao. Esse valor possuira tres casas
decimais, cor vermelha, estara inserido um pouco a direita da posicao
mediana de cada linha vertical vermelha.
  {
    text(x=i+0.2, #posicao do eixo x,  um pouco a direita da linha
previamente tracada
         y=((renda_variavel[i,3] + acoes_proporcao[i]) / 2 ), #posicao do
eixo y, na posicao mediana de cada linha
         round(subtracao[i],3), #arredonda o valor em apenas 3 casas
decimais
         col = "red", #o numero serah plotado com a cor vermelha
         cex = 0.9) #o numero tera 90 por cento do o padrao
  }
  ### O plano eh de que um asterisco seja posicionado acima da empresa cuja
diferenca entre a proporcao investida e o objetivo for a minima, pois isso
indica que eh nessa empresa que o aporte mensal devera ser alocado.
Posicionar o asterisco nessa posicao, que eh variavel, foi desafiador.
Inicialmente eh criado um objeto chamado valor_minimo no qual eh inserido o
valor presente na posicao 1 do vetor subtracao. Alem dele, eh criado um
objeto chamado posicao_x no qual sera inserido o numero 1. Em seguida, entra
em um ciclo for para verificar se o valor seguinte eh menor do que o
anterior. Se sim, o valor serah substituido no objeto valor_minimo e sua
posicao serah substituida no vetor posicao_x. Com isso, eh possivel
determinar a posicao no vetor em que se encontra o valor minimo, e eh
possivel usar essa informacao para determinar em qual ponto do eixo x serah
plotado o asterisco, jah que os codigos das acoes estao posicionados acima
de numeros inteiros, que sao correspondentes a ordem em que eles estao
posicionados no vetor subtracao.
  valor_minimo = subtracao[1] #criacao de um objeto em que eh salvo o valor
presente na posicao 1 do vetor subtracao
  posicao_x=1 #objeto contendo o numero 1
  for (i in 2:nrow(renda_variavel)) #entra em um ciclo for que se iniciarah
na segunda posicao do vetor subtracao, jah que o valor da primeira posicao
foi inserida no objeto recem-criado
  {
    if(subtracao[i] < valor_minimo) #se o numero presente na posicao
seguinte do vetor subtracao for menor do que o anterior...
    {
      valor_minimo = subtracao[i] #...esse valor serah sobrescrito no
objeto...
      posicao_x = i #...e sua posicao serah salva no objeto x
    }
  }
  points (x = posicao_x, #um ponto serah inserido na posicao do eixo x em
que estah presente a empresa cuja diferenca entre a proporcao investida e o
objetivo for a minima
          y = 1, #o ponto serah inserido na posicao 1 do eixo y
          pch = 8, #o ponto serah um asterisco
          col = "red", #o ponto possuira a cor vermelha
```

```
          cex = 1.5) #o ponto serah 1.5 vezes maior do que o padrao
  ### inserir o titulo do grafico
  mtext ("Em qual empresa devo investir?", #insere o texto fora da area do grafico
         side = 3, #o eixo serah inserido sobre o eixo superior
         cex = 1.5) #as letras terao 1.5 vezes o tamanho padrao
  ### inserir a legenda do grafico
  legend ((x = nrow(renda_variavel) + 0.5), #posicao da legenda no eixo x
          y = 0.7, # posicao da legenda no eixo y
          c("Objetivo","Investido", "Invista aqui"), # os textos que deverao aparecer na legenda
          border = "white", # eu coloquei a borda branca porque o padrao eh ter caixas pretas ao redor dos simbolos
          cex = 0.8, # a legenda terah 0.8 do tamanho padrao
          pch = c(19, 19, 8), # determina os tipos de pontos que aparecerao na legenda. No caso, dois pontos totalmente preenchidos e um asterisco
          col = c("blue", "green", "red")) # as cores dos pontos
  ##### calculo do numero de acoes que cada empresa estah distante do seu objetivo
  diferenca_total_dinheiro <- subtracao * acoes_total #o objeto subtracao mostra a diferenca entre a proporcao investida e o objetivo de cada empresa. Nesse comando ocorre a multiplicacao da proporcao que cada investimento estah do seu objetivo pelo valor total do patrimonio investido, indicando quanto dinheiro a mais ou a menos estah aplicado em cada empresa
  diferenca_em_numero_de_acoes <- diferenca_total_dinheiro/renda_variavel$acoes_cotacao #como no passo anterior determinamos quantos reais a mais ou a menos estao aplicados em cada empresa, agora dividimos esse valor pela cotacao de cada empresa, para descobrir quantas acoes a mais ou a menos deveriam ser negociadas para que o objetivo fosse atingido
  renda_variavel_final <- data.frame(renda_variavel$investimento, renda_variavel$acoes_cotacao, diferenca_total_dinheiro, round(diferenca_em_numero_de_acoes, 0)) #criacao de um data frame contendo os codigos das acoes, a cotacao de cada uma delas, quantos reais e quantas acoes a mais ou a menos em relacao aos objetivos. Como nao eh posivel negociar fracoes de acoes, o numero de acoes serah arredondado para um numero inteiro.
  colnames(renda_variavel_final) <- c("investimento", "quantidade_acoes", "diferenca_em_reais", "diferenca_em_numero_de_acoes") #alterar os nomes das colunas do data frame que serah exibido ao final
  ##### O que deve ser retornado ao usuario apos a funcao ser executada
  ### Observacao: o desenvolver da funcao "ghfd_get_HF_data" aparentemente nao colocou um \n ao final do codigo, o que fazia com que o dataframe final retornasse distorcido, porque em vez de ter uma quebra de linha, o cabecalho era inserido na mesma linha em que terminavam de aparecer os dados de processamento da funcao mencionada. Para evitar esse problema, eu decidi salvar uma frase em um objeto que, por meio do print abaixo, eh inserido no restante da linha da funcao, o que permite que o dataframe final se inicie corretamente na linha abaixo.
  parabens <- ("Parabens! O pacote GetHFData conseguiu se comunicar com a Bovespa com sucesso!!") #criacao de um objeto contendo uma frase
```

```
  print(parabens) #essa frase aparece antes do dataframe abaixo
  return(renda_variavel_final) #ao final, a funcao retorna o data frame
recem criado para que o usuario possa ler  as informacoes relevantes
} #final da funcao
```

Página de Ajuda

```
no_que_investir                package:unknown                R
Documentation

SISTEMA PARA DECISÃO DE QUAIS AÇÕES DEVEM SER COMPRADAS

Description:

     A função acessa as cotações das ações por meio do pacote “GetHFData”.
Calcula o patrimônio total e a proporção
     do capital investido em cada empresa e em qual delas a diferença entre
o capital investido e o objetivo é a
     mínima, indicando que é onde o usuário deve investir. A função fornece
um gráfico com as proporções do capital
     investido e os objetivos, e destaca a ação que deve ser comprada. Por
fim, exibe um dataframe com os códigos
     das ações, as cotações e a diferença (reais e número de ações) do
capital investido em relação aos objetivos
     para cada empresa.

Usage:

     no_que_investir(dataframe = planilha)

Arguments:

     Dataframe: nome da planilha que possui os dados. O arquivo deve ser uma
tabela contendo, obrigatoriamente, as
     colunas “investimento”, “objetivo” e “quantidade_acoes”.

Value:

     Plot com pontos azuis (representando os objetivos) e verdes (indicando
a proporção do capital efetivamente
     investido) e a diferença entre essas duas variáveis escrita em números
vermelhos, para cada empresa informada
     na planilha. Um asterisco vermelho indica em qual empresa o resultado
dessa subtração é o menor, o que informa
     ao usuário que é nessa empresa que o aporte mensal deve ser alocado.

     Dataframe com os códigos das ações de cada empresa, suas cotações, e a
```

```
diferença tanto em reais quanto em
     números de ações de cada empresa em relação aos seus objetivos.

Warning:

     A função é interrompida quando, nas colunas “investimento” e
“quantidade_acoes” houver algum dado faltante; e
     quando, na coluna “objetivo” a soma de todos os valores não for igual a
1.

Note:

     Detalhes importantes para a formatação correta do arquivo de entrada:
       1 - Antes de dar início à função, é necessário criar um objeto
chamado “planilha” com a sua planilha de
       controle por meio da função “read.table” seguindo os seguintes
parâmetros:
           a - header = TRUE, #a primeira linha da planilha original
corresponde ao cabeçalho
           b - dec = #a separação decimal da planilha original é feita por
ponto
           c - sep = ";", #o espaçamento da planilha original é feito por
pontos-e-vírgulas
           d - na.strings=c("") #preenche os eventuais espaços vazios da
planilha com NAs.
       2 - A tabela precisa conter, obrigatoriamente, as colunas
“investimento”, “objetivo” e “quantidade_acoes”.
       3 - A coluna “investimento” possui os códigos das ações da carteira
de investimentos do usuário (exemplo:
       ABEV3) que serão utilizadas para que a função busque, por meio do
pacote “GetHFData”, as cotações em um
       repositório da Bovespa;
       4 - A coluna “objetivo” possui a proporção do patrimônio que o
usuário deseja ter investido em cada empresa.
       Para indicar a proporção, use números decimais obedecendo a seguinte
grafia: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,
       0.7, 0.8, 0.9 ou 1.0;
       5 - A coluna “quantidade_acoes” possui o número de ações de cada
empresa que o usuário possui;
       6 - Além dessas, recomenda-se que a tabela contenha duas colunas
adicionais: uma que apresente o nome de
       cada empresa por extenso e outra com seus respectivos CNPJ. Embora
esses dados não sejam usados na análise,
       é interessante manter todas as informações organizadas para facilitar
o processo de declaração de imposto de
       renda;
       7 - A ordem das colunas não importa;

     Limitações da função
       1 - A função é dependente do pacote “GetHFData”;
       2 - O R com frequência possui dificuldade em se comunicar com o
```

```
repositório da Bovespa, então seja
       perseverante (tente rodar a função mais de uma vez se a conexão não
for estabelecida com sucesso) e paciente
       (o R importa um grande volume de dados, o que demanda tempo);
       3 - O repositório da Bovespa (ftp://ftp.bmf.com.br/MarketData/) é
atualizado apenas uma vez por dia (em
       horários variáveis), quando são inseridos os valores das cotações do
dia anterior. Sendo assim, as
       informações infelizmente possuem ao menos um dia de atraso, logo, não
fornecem as cotações em tempo real.

Author(s):
     Augusto Santos Rampasso
     augusto.santosr@gmail.com

Examples:

     planilha <- read.table("planilha_1.csv", header = TRUE, dec = ".", sep
= ";", na.strings=c(""))
     no_que_investir(dataframe = planilha)

     planilha <- read.table("planilha_2.csv", header = TRUE, dec = ".", sep
= ";", na.strings=c(""))
     no_que_investir(dataframe = planilha)
```

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:augusto.rampasso:trab_final

Last update: **2020/08/12 06:04**