

Hugo Renan Bolzani



Doutorando em Engenharia Civil e Ambiental pela UNESP. Atualmente, é Professor Efetivo do Instituto Federal do Sul de Minas Gerais (IFSULDEMINAS) - Campus Poços de Caldas.

Meus exercícios

[Exercícios](#)

Trabalho Final

Proposta A: Treino de musculação

Contextualização:

Para quem vive uma rotina de musculação é fundamental ter em mãos uma sequência de treinos de acordo com o seu nível, mas as vezes fica difícil e confuso selecionar quais os próximos exercícios que serão mais adequados após a finalização do protocolo de treino que atualmente está sendo executado. O tempo adequado de recuperação entre as sessões de treinamento é a variável que vai determinar a melhor forma de dividir o programa de musculação. A divisão depende do número de séries e repetições utilizadas por grupo muscular na sessão de treinamento e é essa diferença no tempo de recuperação que vai determinar a forma que o programa de musculação será dividido.

Exemplos de divisão de treino:

- 1AB - 2X na semana - Membros Superiores A | Membros Inferiores B;
- 2AB - 2X na semana - Perna, Costa, Bíceps, Abdômen A | Peito, Ombro, Tríceps, Gêmeos B;
- AB - 3X na semana - Membros Superiores A | Membros Inferiores B;

- ABC – 2X na semana – Peito, Ombro, Tríceps A | Costa, Bíceps, Abdômen B | Perna completa C.

Para cada grupo, existe uma listagem de exercícios, tais como:

- desenv. fechado na máq.|exc|3|12+3 rep|90°;
- desenv. unil. com halter|axt|3|10" + 10|90°;
- elevação lateral|drop|2|6+6+6|90°;
- crucifixo inverso |ox|2|12|45°;
- sup. fechado barra w|ox|3|15|30°;
- pulley barra v|ss|2|8|;
- francês com corda|ss|2|8|60°;
- tríceps banco pé alto|ox|2|8 até pf|45°;
- gêmeos na leg reto,abd,adz|drop|3|12+12+12|30°.

Sendo assim, a função irá auxiliar as pessoas que desejam ter um plano para o próximo protocolo de treino de musculação.

Planejamento da Função:

Entrada:

- Data.frame com os diferentes tipos de treino.
- Data.frame com o treino anterior.

Pseudo-código:

1. Carregar o arquivo contendo os diferentes tipos de treino.
2. Carregar o arquivo com o treino anterior.
3. Verificar quantos dias na semana o usuário irá treinar.
4. Verificar que tipo de treino o usuário quer (AB, ABC...).
5. Com base no tipo de treino escolhido, selecionar os exercícios que devem ser realizados, bem como o tempo de descanso (não deve ter repetição com o protocolo anterior ou poderá ter um número mínimo de repetição).
6. Gerar um data.frame com o novo protocolo de treino.

Saída:

Data.frame como protocolo de treino.

Referência:

MUSCULAÇÃO ONLINE. Divisão das séries de musculação, guia completo! Disponível em: <https://musculacaoonline.com.br/hipertrofia-muscular/>. Acessado em: 01/05/2018.

Proposta B: Índice de Qualidade de Esgoto Tratado (IQET)

Contextualização:

Na área de tratamento de esgotos é necessário mecanismos para avaliar a eficiência de uma estação de tratamento, consequentemente ter conhecimento se a estação está ou não causando um impacto ambiental no corpo d'água receptor do esgoto tratado. O Índice de Qualidade do Esgoto Tratado é um instrumento capaz de realizar a função supracitada. O índice é obtido por meio de um cálculo matemático que avalia a qualidade do efluente final das estações de tratamento de esgoto. Cada parâmetro físico-químico considerado para o cálculo do índice possui um peso percentual de representação que são respectivamente: Demanda Bioquímica de Oxigênio (DBO₅), Demanda Química de Oxigênio (DQO), Sólidos Sedimentáveis (SS), Óleos e graxas (OG) e Potencial Hidrogeniônico (pH). Para cada parâmetro são definidas faixas que definem o valor para o parâmetro analisado, conforme a Tabela a seguir.

Tabela - Parâmetros e pesos para o IQET

Pontuação	Parâmetros				
	S.Sed (mg/L)	DQO (mg/L)	DBO ₅ (mg/L)	OG (mg/L)	pH
100	0 – 1	0 – 150	0 – 60	0 – 70	5 a 9
75	1,1 – 2	151 – 200	61 – 80	71 – 90	-
50	2,1 – 3	201 – 250	81 – 100	91 – 110	-
25	3,1 – 5	251 – 300	101 – 180	111 – 130	-
0	> 5	> 300	> 180	> 130	< 5 ou > 9
Peso	0,20	0,20	0,40	0,10	0,10

Considerando o peso de cada parâmetro, há uma avaliação dos níveis de atendimento. Se o parâmetro DBO₅ apresentar valores acima de 180 mg/L, o IQET é automaticamente zero, independentemente dos resultados dos outros parâmetros. Este índice tem sua classificação assim distribuída:

- 0 - 40: qualidade precária;
- 41-70: qualidade inadequada;
- 71 - 90: qualidade aceitável;
- 90 - 100: qualidade ótima.

O cálculo é realizado com base na seguinte Equação:

$$\text{IQET} = \text{SSed} \cdot 0,20 + \text{DQO} \cdot 0,20 + \text{DBO}_5 \cdot 0,40 + \text{OG} \cdot 0,10 + \text{pH} \cdot 0,10$$

Sendo assim, a função irá apresentar o IQET de determinado período de uma estação de tratamento de esgoto e auxiliar as companhias de saneamento no planejamento e tomada de decisões com base no resultado instantâneo da eficiência de tratamento de suas estações.

Planejamento da Função:

Entrada:

Data.frame com duas colunas contendo os parâmetros e seus respectivos valores.

Verificando os Parâmetros:

DBO é um número maior que 180? Se sim, escreve: "DBO elevada = IQET com qualidade precária."

Pseudo-código:

1. Carregar o arquivo contendo os parâmetros e seus respectivos valores.
2. Verificar se o valor da DBO é maior que 180.
3. Em caso positivo, encerrar o processo indicando "qualidade precária".
4. Em caso negativo, continuar o processo.
5. Atribuir a pontuação dos parâmetros de acordo com a faixa em que os valores se encontram.
6. Realizar o cálculo do IQET utilizando a pontuação e os pesos dos parâmetros.
7. Apresentar a qualidade em que se encontra a ETE com base no resultado obtido no cálculo.

Saída:

Informação sobre o Valor do IQET: qualidade da eficiência da ETE com base nos valores dos parâmetros apresentados. Exemplo: "Resultado do IQET: qualidade inadequada".

Referência:

SANEPAR. Sanare: Revista técnica da Sanepar.

<http://www.sanepar.com.br/sanepar/sanare/v14/html/Pagina1.htm>. Acesso: 01/05/2018.

A função da proposta A não parece bem clara. Seu pseudo-código mostra o que vai entrar e o que vai sair, mas pelo que está escrito eu não consigo avaliar a dificuldade dos cálculos e operações que a função vai fazer, então seria legal vc detalhar seu pseudo-código com essas etapas de cálculos e manipulação das entradas da função. Outra coisa (menos trabalhosa) que também seria legal é deixar claro de que forma a função vai "verificar" os dias e o tipo de treino (por exemplo: isso será estipulado através de argumentos, ou estimado a partir da tabela de entrada?).

Achei a proposta B simples demais, pois a única decisão da função é saber se $DBO > 180$ e a partir daí aplicar uma única fórmula.

Meu conselho é que vc detalhe mais o plano A e realize esta mesma proposta. Opcionalmente você pode aumentar a complexidade da proposta B consideravelmente, mas eu não sei se existe tanta margem pra isso nesse assunto.

Para ajudar um pouco mais nesse detalhamento, no dia 15 ou 16 vou entrar aqui de novo pra olhar as modificações do seu pseudo código e te dar mais feedbacks. Depois desse prazo, vc também pode tentar me contatar por whatsapp (por favor não mande áudio) (11) 9-9199-3842

Matheus Januario

Olá Matheus, obrigado pelas considerações. Estou participando de um evento e não conseguirei melhorar o pseudo-código a tempo pra você dar mais sugestões. Acho que irei pela proposta A como aconselhou. Estou pensando em usar o sample para sortear os exercícios que estarão no próximo protocolo. Acho que seria necessário uma função interativa, para o usuário informar algumas coisas (exemplo: quantos dias na semana irá treinar) e obter o protocolo com os exercícios. O que acha?

Função Musculação

```
musculacao <- function(treino,
treino.anterior=data.frame("treinoanterior.csv")) #iniciando a funcao
{

peito <- exercicios[exercicios$grupo=="peito", 2] #criacao de objeto com
exercicios de peito indexados
costas <- exercicios[exercicios$grupo=="costas", 2] #criacao de objeto com
exercicios de costas indexados
abdominal <- exercicios[exercicios$grupo=="abdominal", 2] #criacao de objeto
com com exercicios de abdominal indexados
perna <- exercicios[exercicios$grupo=="perna", 2] #criacao de objeto com
exercicios de perna indexados
biceps <- exercicios[exercicios$grupo=="biceps", 2] #criacao de objeto com
exercicios de biceps indexados
triceps <- exercicios[exercicios$grupo=="triceps", 2] #criacao de objeto com
exercicios de triceps indexados
ombro <- exercicios[exercicios$grupo=="ombro", 2] #criacao de objeto com
exercicios de ombro indexados

treino <- readline("Seu arquivo possui no mínimo 6 exercicios para cada
grupo muscular? Responda 's' para sim e 'n' para nao.") #ocorre uma
```

interacao com o usuario com uma pergunta na tela

```
if(substr(treino, 1, 1)=="n") #se a resposta for 'nao' para a pergunta "Seu data.frame com os exercicios possui no mínimo 6 exercicios para cada grupo muscular?"
```

```
  stop("Voce deve ter ao menos 6 opcoes de exercicios para cada grupo muscular") #interrompe a funcao em caso negativo e avisa o usuario
```

```
if(substr(treino, 1, 1)=="s") #se a resposta for 'sim' para a pergunta "Seu data.frame com os exercicios possui no mínimo 6 exercicios para cada grupo muscular?"
```

```
{
```

```
  treino <- readline("Voce quer treinar um grupo muscular por dia? Responda 's' para sim e 'n' para nao.") #ocorre uma interacao com o usuario com uma pergunta na tela
```

```
  if(substr(treino, 1, 1)=="s") #se a resposta for 'sim' para a pergunta "Voce quer treinar um grupo por dia?"
```

```
  {
```

```
    #a verificacao servira para todas as opcoes de treino que o usuario podera escolher
```

```
    if(length(exercicios[exercicios$grupo=="peito", 2])<6) #verifica se ha no minimo 6 opcoes de exercicio de peito no data.frame
```

```
      stop("Voce deve ter ao menos 6 opcoes de exercicios para o treino de peito.") #interrompe a funcao em caso negativo e avisa o usuario
```

```
    if(length(exercicios[exercicios$grupo=="costas", 2])<6) #verifica se ha no minimo 6 opcoes de exercicio de costas no data.frame
```

```
      stop("Voce deve ter ao menos 6 opcoes de exercicios para o treino de costas.") #interrompe a funcao em caso negativo e avisa o usuario
```

```
    if(length(exercicios[exercicios$grupo=="abdominal", 2])<6) #verifica se ha no minimo 6 opcoes de exercicio de abdominal no data.frame
```

```
      stop("Voce deve ter ao menos 6 opcoes de exercicios para o treino de abdominal.") #interrompe a funcao em caso negativo e avisa o usuario
```

```
    if(length(exercicios[exercicios$grupo=="perna", 2])<6) #verifica se ha no minimo 6 opcoes de exercicio de perna no data.frame
```

```
      stop("Voce deve ter ao menos 6 opcoes de exercicios para o treino de perna.") #interrompe a funcao em caso negativo e avisa o usuario
```

```
    if(length(exercicios[exercicios$grupo=="biceps", 2])<6) #verifica se ha no minimo 6 opcoes de exercicio de biceps no data.frame
```

```
      stop("Voce deve ter ao menos 6 opcoes de exercicios para o treino de biceps.") #interrompe a funcao em caso negativo e avisa o usuario
```

```
    if(length(exercicios[exercicios$grupo=="triceps", 2])<6) #verifica se ha no minimo 6 opcoes de exercicio de triceps no data.frame
```

```
      stop("Voce deve ter ao menos 6 opcoes de exercicios para o treino de triceps.") #interrompe a funcao em caso negativo e avisa o usuario
```

```
    if(length(exercicios[exercicios$grupo=="ombro", 2])<6) #verifica se ha no minimo 6 opcoes de exercicio de ombro no data.frame
```

```
      stop("Voce deve ter ao menos 6 opcoes de exercicios para o treino de ombro.") #interrompe a funcao em caso negativo e avisa o usuario
```

```
    exer.peito <- sample(peito, size=5, replace=F) #sorteio de 5 exercicios sem repeticao
```

```
    exer.costas <- sample(costas, size=6, replace=F) #sorteio de 6
exercicios sem repeticao
    exer.abdominal <- sample(abdominal, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
    exer.perna <- sample(perna, size=5, replace=F) #sorteio de 5 exercicios
sem repeticao
    exer.biceps <- sample(biceps, size=6, replace=F) #sorteio de 6
exercicios sem repeticao
    exer.triceps <- sample(triceps, size=5, replace=F) #sorteio de 5
exercicios sem repeticao
    exer.ombro <- sample(ombro, size=6, replace=F) #sorteio de 6 exercicios
sem repeticao
    DiaA.exercicio.repeticoes.intervalo <- c(exer.peito, exer.abdominal[1])
#objeto treino do dia A eh feito com os exercicios de peito e abdominal na
posição 1
    DiaB.exercicio.repeticoes.intervalo <- exer.costas #objeto treino do dia
B eh feito com os exercicios de costas
    DiaC.exercicio.repeticoes.intervalo <- c(exer.perna,
exer.abdominal[2])#objeto treino do dia C eh feito com os exercicios de
perna e abdominal na posição 2
    DiaD.exercicio.repeticoes.intervalo <- exer.ombro #objeto treino do dia
D eh feito com os exercicios de ombro
    DiaE.exercicio.repeticoes.intervalo <- exer.biceps #objeto treino do dia
E eh feito com os exercicios de biceps
    DiaF.exercicio.repeticoes.intervalo <- c(exer.triceps,
exer.abdominal[3]) #objeto treino do dia F eh feito com os exercicios de
triceps e abdominal na posição 3
    treino.novo <- data.frame(DiaA.exercicio.repeticoes.intervalo,
DiaB.exercicio.repeticoes.intervalo,DiaC.exercicio.repeticoes.intervalo,DiaD
.exercicio.repeticoes.intervalo,DiaE.exercicio.repeticoes.intervalo,DiaF.exe
rcicio.repeticoes.intervalo) #cria o data.frame com os exercicios do novo
treino
  }
  if(substr(treino, 1, 1)=="n") #se a resposta for 'nao' para a pergunta
"Voce quer treinar um grupo por dia?"
  {
    treino <- readline("Quantos dias na semana voce ira treinar? Escolha de
3 a 6 dias.") #Ocorre uma interação com o usuario com uma nova pergunta na
tela
    if(substr(treino, 1, 1)=="6") #se a resposta for '6' para a pergunta
"Quantos dias na semana voce ira treinar?"
    {
      exer.peito <- sample(peito, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
      exer.costas <- sample(costas, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
      exer.abdominal <- sample(abdominal, size=2, replace=F) #sorteio de 2
exercicios sem repeticao
      exer.perna <- sample(perna, size=4, replace=F) #sorteio de 4
exercicios sem repeticao
      exer.biceps <- sample(biceps, size=3, replace=F) #sorteio de 3
```

```
exercicios sem repeticao
  exer.triceps <- sample(triceps, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.ombro <- sample(ombro, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  DiaA.exercicio.repeticoes.intervalo <- c(exer.peito, exer.biceps,
exer.abdominal[1]) #objeto treino do dia A eh feito com os exercicios de
peito e biceps e abdominal na posicao 1
  DiaB.exercicio.repeticoes.intervalo <- c(exer.ombro, exer.perna)
#objeto treino do dia B eh feito com os exercicios de ombro e perna
  DiaC.exercicio.repeticoes.intervalo <- c(exer.costas, exer.triceps,
exer.abdominal[2]) #objeto treino do dia C eh feito com os exercicios de
costas e triceps e abdominal na posicao 1
  DiaD.exercicio.repeticoes.intervalo <-
DiaA.exercicio.repeticoes.intervalo #objeto treino do dia D é igual ao A
  DiaE.exercicio.repeticoes.intervalo <-
DiaB.exercicio.repeticoes.intervalo #objeto treino do dia E é igual ao B
  DiaF.exercicio.repeticoes.intervalo <-
DiaC.exercicio.repeticoes.intervalo #objeto treino do dia F é igual ao C
  treino.novo <-
data.frame(DiaA.exercicio.repeticoes.intervalo,DiaB.exercicio.repeticoes.int
ervalo,DiaC.exercicio.repeticoes.intervalo,DiaD.exercicio.repeticoes.interva
lo,DiaE.exercicio.repeticoes.intervalo,DiaF.exercicio.repeticoes.intervalo)
#cria o data.frame com os exercicios do novo treino
}
  if(substr(treino, 1, 1)=="5") #se a resposta for '5' para a pergunta
""Quantos dias na semana voce ira treinar?
  {
    exer.peito <- sample(peito, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
    exer.costas <- sample(costas, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
    exer.abdominal <- sample(abdominal, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
    exer.perna <- sample(perna, size=5, replace=F) #sorteio de 5
exercicios sem repeticao
    exer.biceps <- sample(biceps, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
    exer.triceps <- sample(triceps, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
    exer.ombro <- sample(ombro, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
    DiaA.exercicio.repeticoes.intervalo <- c(exer.peito, exer.biceps)
#objeto treino do dia A eh feito com os exercicios de peito e biceps
    DiaB.exercicio.repeticoes.intervalo <- c(exer.perna,
exer.abdominal[1]) #objeto treino do dia B eh feito com os exercicios de
perna e abdominal na posicao 1
    DiaC.exercicio.repeticoes.intervalo <- c(exer.costas, exer.triceps)
#objeto treino do dia C eh feito com os exercicios de costas e triceps
    DiaD.exercicio.repeticoes.intervalo <- c(exer.ombro, exer.abdominal)
```

```
#objeto treino do dia D eh feito com os exercicios de ombro e abdominal
DiaE.exercicio.repeticoes.intervalo <-
DiaA.exercicio.repeticoes.intervalo #objeto treino do dia E eh feito com os
exercicios do primeiro dia de treino
treino.novo <-
data.frame(DiaA.exercicio.repeticoes.intervalo,DiaB.exercicio.repeticoes.int
ervalo,DiaC.exercicio.repeticoes.intervalo,DiaD.exercicio.repeticoes.interva
lo,DiaE.exercicio.repeticoes.intervalo) #cria o data.frame com os exercicios
do novo treino
}
if(substr(treino, 1, 1)=="4") #se a resposta for '4' para a pergunta
"Quantos dias na semana voce ira treinar?"
{
  exer.peito <- sample(peito, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.costas <- sample(costas, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.abdominal <- sample(abdominal, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.perna <- sample(perna, size=5, replace=F) #sorteio de 5
exercicios sem repeticao
  exer.biceps <- sample(biceps, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.triceps <- sample(triceps, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.ombro <- sample(ombro, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  DiaA.exercicio.repeticoes.intervalo <- c(exer.peito, exer.biceps)
#objeto treino do dia A eh feito com os exercicios de peito e biceps
  DiaB.exercicio.repeticoes.intervalo <- c(exer.perna,
exer.abdominal[1]) #objeto treino do dia B eh feito com os exercicios de
perna nae abdominal na posicao 1
  DiaC.exercicio.repeticoes.intervalo <- c(exer.costas, exer.triceps)
#objeto treino do dia C eh feito com os exercicios de costas e triceps
  DiaD.exercicio.repeticoes.intervalo <- c(exer.ombro, exer.abdominal)
#objeto treino do dia D eh feito com os exercicios de ombro e abdominal
  treino.novo <- data.frame(DiaA.exercicio.repeticoes.intervalo,
DiaB.exercicio.repeticoes.intervalo, DiaC.exercicio.repeticoes.intervalo,
DiaD.exercicio.repeticoes.intervalo) #cria o data.frame com os exercicios do
novo treino
}
if(substr(treino, 1, 1)=="3") #se a resposta for '3' para a pergunta
"Quantos dias na semana voce ira treinar?"
{
  exer.peito <- sample(peito, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.costas <- sample(costas, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.abdominal <- sample(abdominal, size=2, replace=F) #sorteio de 2
exercicios sem repeticao
  exer.perna <- sample(perna, size=4, replace=F) #sorteio de 4
```

```
exercicios sem repeticao
  exer.biceps <- sample(biceps, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.triceps <- sample(triceps, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  exer.ombro <- sample(ombro, size=3, replace=F) #sorteio de 3
exercicios sem repeticao
  DiaA.exercicio.repeticoes.intervalo <- c(exer.peito, exer.biceps,
exer.abdominal[1]) #objeto treino do dia A eh feito com os exercicios de
peito e biceps e abdominal na posicao 1
  DiaB.exercicio.repeticoes.intervalo <- c(exer.ombro, exer.perna)
#objeto treino do dia B eh feito com os exercicios de ombro e perna
  DiaC.exercicio.repeticoes.intervalo <- c(exer.costas, exer.triceps,
exer.abdominal[2]) #objeto treino do dia C eh feito com os exercicios de
costas e triceps e aqbdominal na posicao 2
  treino.novo <-
data.frame(DiaA.exercicio.repeticoes.intervalo,DiaB.exercicio.repeticoes.int
ervalo,DiaC.exercicio.repeticoes.intervalo) #cria o data.frame com os
exercicios do novo treino
}
}
exercicios.repetidos <- intersect(unlist(as.list(treino.anterior)),
unlist(as.list(treino.novo))) #interseccao da transformacao dos treinos em
listas para usar o if else
cat("\nExercicios do treino novo que se repetem com o treino antigo:\n")
#mostra uma mensagem na tela. Ha quebra de linha.
if(length(exercicios.repetidos) > 0) #cria uma condicao: se os exercicios
repetidos forem maior que 0
{
  cat(paste(toString(exercicios.repetidos), "\n", "\n")) #aparecera na
tela os exercicios repetidos dos treinos
}
else{ #se a situacao for diferente da anterior
  cat("Nao ha exercicios repetidos.\n \n") #aparecera na tela essa
mensagem. Ha quebra de linha.
}
cat("O arquivo treionovo.csv com o novo treino foi criado no diretorio
ativo.") #mostra uma mensagem na tela
cat("\n","\n") #quebra de linha
cat("Treino novo:\n") #mostra uma mensagem na tela
cat("\n") #quebra de linha
write.table(treino.novo, file="treionovo.csv", sep=";", row.names=F)
#cria um arquivo cvs no diretorio ativo com o novo treino conforme a escolha
do usuario
  return(treino.novo) #mostra na tela o resultado do novo treino para o
usuario
}
}
```

Help da Função

musculacao package:unknown
R Documentation

MONTAGEM DE TREINOS DE MUSCULAÇÃO

Description:

Função para montar de forma interativa treinos de musculação conforme a disponibilidade de dias do usuário.

A função necessita de um **data.frame** com uma vasta lista de exercícios de musculação, sendo indexados por grupo muscular e, em seguida, sorteados em diferentes tipos de treinos com base nas escolhas do usuário.

Usage:

`musculacao(exercicios, treino.anterior)`

Arguments:

exercicios: **data.frame** com uma lista de exercícios de musculação.

treino.anterior: **data.frame** com os exercícios que constam no último treino do usuário.

Details:

Para a criação do treino de musculação um **data.frame** com uma lista de exercícios (`exercicios.csv`) e um **data.frame** com o último treino do usuário (`treino.anterior.csv`) devem ser fornecidos.

É necessário realizar as leituras desses arquivos no formato `csv`.

Os exercícios são indexados por grupo muscular (peito, costas, abdominal, perna, tríceps, bíceps e ombro). Interações ocorrem com o usuário para verificar se há no mínimo seis exercícios de musculação para cada grupo muscular no `data frame` `exercicios`, se o usuário quer treinar um grupo muscular por dia ou quantos dias da semana será o usuário irá treinar.

A função sorteia os exercícios de cada grupo muscular com base nas escolhas do usuário.

Após as interações, a função mostra na tela quais exercícios do treino novo se repetem com o treino anterior (caso o usuário deixe o **data.frame** `'treino.anterior'` vazio, aparecerá uma mensagem dizendo que não houve repetições).

A função mostra também o treino novo na tela e gera um arquivo no formato `csv` com o treino novo (`novotreino.csv`) no diretório ativo.

O **data.frame** `'exercicios'` fornecido pelo usuário deverá obedecer a seguinte configuração: grupos musculares (exclusivamente: peito, biceps, triceps, perna, costas, ombro, abdominal) na primeira coluna, sendo nomeada como `'grupos'`; exercícios musculares na segunda coluna, sendo nomeado como `título` `'exercicios|repeticoes|intervalo'` (quanto maior a quantidade de exercícios menor é a chance de gerar repetição entre os treinos).

Nesta coluna, os três elementos devem estar discriminados conforme o exemplo: `'P.supino reto|3x15|60'`, sendo `P.supino reto` um exercício de Peito

(P), 3x15 a repetição do exercício e 60 o tempo de intervalo em segundos. Como o grupo e peito e perna possuem a mesma letra inicial, o nome do exercício de perna deve ser iniciado com a letra maiúscula 'I' (de membros inferiores), como exemplo: I.Afundo com halter. O nome das colunas deve estar em letra minúscula e sem acentuação e cedilha.

Value:

Tela: Exercícios repetidos entre o treino antigo e novo e o **data.frame** com o novo treino.

Data.frame: um arquivo 'treinonovo.csv' contendo o novo treino é gerado e armazenado no diretório ativo.

Warning:

Se o **data.frame** contido no 'exercicios.csv' não possuir 6 exercícios para cada grupo muscular, a função não é executada e aparecerá uma mensagem na tela.

Nas interações com o usuário, as respostas devem ser dadas conforme as opções dadas na tela. Caso isso não ocorra, a função não é executada.

Author:

Hugo Renan Bolzani

e-mail: hugo.bolzani@gmail.com

References:

MUSCULAÇÃO ONLINE. Divisão das séries de musculação, guia completo!

Disponível em: <https://musculacaoonline.com.br/hipertrofia-muscular/>.

Acessado em: 01/05/2018.

RUGGIERO, P.G.C. cardapio R Documentação. 2017.

Examples:

#Função

```
musculacao(exercicios,treino.anterior)
```

#Entrada dos data.frame

```
exercicios<-read.csv("exercicios.csv",head=T,sep=";",stringsAsFactors=F)
```

```
treino.anterior<-
```

```
read.csv("treinoanterior.csv",head=T,sep=";",stringsAsFactors=F)
```

Exemplo de data.frame contendo vários exercícios de musculação: [exercicios.csv](#)

Exemplo de data.frame contendo um treino anterior do usuário: [treinoanterior.csv](#)

Arquivo da função: [musculacao.r](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:hugo.bolzani:start

Last update: **2020/09/23 20:15**

