

Patricia Guidao Cruz Ruggiero



Doutoranda do Departamento de Ecologia, USP

[patricia_ruggiero.pdf](#) [exec](#)

Meus exercícios

[pruggiero_exer01.r](#) [aula4_exerc01.r](#) [aula4_exerc02.r](#) [aula4_exerc03.r](#) [aula4_exerc04.r](#)
[pruggiero_exerc4.rtf](#) [pruggiero_exerc5.rtf](#) [aula7_exerc02.r](#) [aula7_exerc2-5.r](#) [aula7b_exercicios.r](#)
[aula8_exerc_palmito.r](#) [aula9_exerc02.r](#) [aula9_exerc03.r](#)

Propostas - Trabalho Final

Plano A: Função CARDAPIO

Se você mantém uma casa onde pessoas, e especialmente crianças almoçam e jantam todos os dias, uma das suas funções semanais é montar um cardápio completo e balanceado para cada dia da semana. Isso requer que você pense, organize e seja criativo (!) nas escolhas das receitas TODAS as semanas. Essa função repetitiva toma tempo da sua semana (já ocupada de tarefas!), pode se tornar cansativa e por fim você pode acabar gerando cardápios monótonos, sempre muito parecidos. A função CARDAPIO irá montar um cardápio para cada dia da semana a partir de um dataframe com as receitas elencadas por você. Para que a função CARDAPIO possa rodar, as receitas devem ser indexadas pela posição no cardápio (por exemplo: entrada, prato principal, acompanhamento) e pelo tipo principal de ingrediente (por exemplo: carne vermelha, frango, peixe, ovo, legume verde, legume vermelho/amarelo, carboidrato). Com isso, a função CARDAPIO irá montar cinco combinações (para segunda-feira, terça-feira, quarta-feira, quinta-feira e sexta-feira; sábado e domingo vamos deixar a vida rolar mais solta!) tomando como parâmetros as escolhas que você faz sobre: 1) as posições (quantas entradas, pratos principais e acompanhamentos por dia) e 2) sobre os tipos de ingredientes (quanta carne vermelha, peixe, legumes, carboidrato na semana). Cada receita somente pode aparecer em uma das cinco combinações. A função tem como saída um objeto do tipo lista contendo cinco listas, cada uma apresentando a combinação de cada dia. A lista de saída (com as cinco combinações) será nomeada com a data e a hora em que as combinações foram geradas. Adicionalmente, a função CARDAPIO vai comparar a lista recém gerada com uma lista indicada por você (cardápio da semana passada) para checar se há sobreposição de receitas. A função irá informar o número de receitas que se repetem nos dois cardápios para que você decida se quer gerar novo cardápio ou se está satisfeito com a combinação apresentada. E bom apetite!

Obs: aqui em casa, a função CARDAPIO já está sendo aguardada ansiosamente.

Plano B: Função CAM (Conta de Água Monitorada - uma referência às plantas do tipo CAM, eficientes na economia de água)

Em tempos sombrios de escassez de água, muito estresse social pode ser gerado entre usuários deste recurso. Se você mora em casa ou em prédio que tem contador de água individual, ótimo!, você pode controlar o seu gasto acompanhando o girar do relógio. Porém, se você mora em prédio onde o contador de água é coletivo, é muito difícil saber quanto você gasta e quanto o vizinho gasta e, portanto, discutir e promover a economia de água. A função CAM (Conta de Água Monitorada) irá calcular o gasto de água estimado da sua habitação com base em informações fornecidas por você sobre: 1) o número de pessoas na habitação e o tempo médio de banho de cada pessoa, 2) o tempo total diário de torneira corrente, 3) número de vezes que a máquina lava-roupa é acionada, e 4) número de vezes que a máquina lava-louças é acionada. As informações 1, 2, 3 e 4 serão fornecidas na forma de vetores e a função CAM calculará o gasto estimado de água no mês, considerando o número de dias do mês calculado. A saída, portanto, é um vetor com um valor contendo a estimativa em metros cúbicos para o gasto da sua habitação. Com este valor estimado, e considerando que todas as habitações do seu condomínio economizaram água como você, a função CAM poderá calcular uma estimativa para a conta de água total do condomínio e comparar com o gasto real fornecido por você. A saída desta comparação é um histograma com os dois valores - um excelente material para a sua próxima reunião de condomínio!

Proposta A boa que levava a aluna a trabalhar conceitos de manipulação de dados. A proposta B me parece simples demais. Envolve apenas um cálculo de consumo médio mensal de água.

- Vitor Aguiar

Pode seguir com a proposta A, sugiro que tente fazer as opções da função o mais interativo possível. Veja a função readline e a função simula que foi utilizada em aula para exemplo de como tornar a função interativa.

— [Alexandre Adalardo de Oliveira](#) 2015/03/27 14:25

Obrigada pelo retorno e pelas sugestões. Ao trabalho!

Patricia

Função CARDAPIO

Help da Função

cardapio R Documentação

Sorteio de receitas e Montagem de cardápios

Descrição

A função `cardapio` é uma função interativa que sorteia elementos (receitas) a partir de um `data.frame` e organiza estes elementos em uma lista indexada pelos cinco dias da semana, formando cardápios para cada dia da semana. O sorteio dos elementos do `data.frame` é feito a partir de uma indexação destes elementos pela posição no cardápio e pelo ingrediente principal da receita.

Uso

```
cardapio(receitas, cardapio.anterior)←data.frame
```

Argumentos

`receitas` `data.frame` contendo as receitas. Veja 'Detalhes'.

`cardapio.anterior` lista contendo um cardápio gerado anteriormente

Detalhes

Na função `cardapio`, as receitas são fornecidas em um `data.frame`. As receitas são indexadas pelo tipo (que trata da posição no cardápio: entrada, prato principal, acompanhamento e prato único) e pelo ingrediente principal (carne, ave, peixe, porco, veggie, arroz, grão, verde, outro). Objetos são criados com base em tipo e em ingrediente para a montagem de cardápios. A função sorteia cinco elementos (receitas) a partir desta indexação formando objetos que contêm estes elementos e a partir dos quais são montados os cardápios para os cinco dias da semana (segunda, terça, quarta, quinta e sexta-feira). Na montagem dos cardápios, as receitas nas primeiras posições dos objetos (com os elementos sorteados) vão compor o cardápio da segunda-feira, as receitas nas segundas posições vão compor a terça-feira e assim por diante até a sexta-feira que corresponde às receitas nas quintas posições. Na interação com a função, o usuário pode optar por um cardápio vegetariano ou pode optar, para cada dia da semana, se deseja comer carne, ave, peixe, porco ou uma receita vegetariana (veggie). O usuário também pode optar se deseja que a função retorne uma sugestão de receita para o jantar, que é sorteada a partir das opções de prato único. O `data.frame` com as receitas deve ser fornecido pelo usuário. O `data.frame` deve necessariamente obedecer a seguinte configuração. 1) Conter as receitas na primeira coluna (`head==título` desta coluna deve ser 'receitas'). 2) Conter o tipo na segunda coluna (classificado exclusivamente como: entrada, prato principal, acompanhamento e prato unico); esta coluna deve ser nomeada 'tipo'. 3) Conter o ingrediente na terceira coluna (classificado exclusivamente como: carne, ave, peixe, porco, veggie, arroz, grão, verde, outro); esta coluna deve ser nomeada 'ingrediente'. Receitas do tipo 'entrada' e 'prato unico' não precisam ser classificadas quanto ao ingrediente. É necessário que existam ao menos cinco receitas classificadas como 'prato principal' e 'veggie' e ao menos cinco receitas

classificadas como 'acompanhamento' e 'outro' para a montagem do cardápio vegetariano. Também é necessário que existam ao menos cinco receitas classificadas como 'carne', 'ave' e 'peixe' e ao menos cinco receitas classificadas como 'acompanhamento', 'verde' e 'outro', para a montagem do cardápio não-vegetariano. Os nomes das colunas devem estar necessariamente com letra minúscula e não se deve usar assentos de nenhum tipo (agudo, til etc) e nem cedilha. Um exemplo de data.frame está disponível abaixo.

Data.frame

A função retorna um data.frame contendo os cardápios montados para os cinco dias da semana e a opção para o jantar.

Autor

Patricia G. C. Ruggiero pruggiero@usp.br

Exemplo

```
> cardapio02=cardapio(minhasreceitas, cardapio01)
```

Nota

Recomenda-se que o data.frame das receitas seja importado utilizando-se o argumento `stringsAsFactors=FALSE`, como no exemplo a seguir:

```
> minhasreceitas<-read.csv("receitas de casa.csv", head=T, sep=";",  
stringsAsFactors=F)
```

Recomenda-se que o cardápio gerado seja gravado usando:

```
write.table(cardapio02, file="cardapio02.csv", sep=";", row.names=F)
```

para que ele possa ser buscado posteriormente para preencher o argumento 'cardapio.anterior'.

Código da Função

```
cardapio <- function(receitas, cardapio.anterior=data.frame())  
{  
  #Indexação das receitas do data.frame por tipo e ingrediente  
  carne<-receitas[receitas$tipo=="prato principal" &  
receitas$ingrediente=="carne",1] #cria objeto só com prato principal de  
carne  
  ave<-receitas[receitas$tipo=="prato principal" &  
receitas$ingrediente=="ave",1] #cria objeto só com prato principal de ave  
  peixe<-receitas[receitas$tipo=="prato principal" &  
receitas$ingrediente=="peixe", 1] #cria objeto só com prato principal de  
peixe  
  porco<-receitas[receitas$tipo=="prato principal" &  
receitas$ingrediente=="porco",1] #cria objeto só com prato principal de
```

```
porco
  veggie<-receitas[receitas$tipo=="prato principal" &
receitas$ingrediente=="veggie",1] #cria objeto só com prato principal
vegetariano
  arroz<-receitas[receitas$ingrediente=="arroz", 1] #cria objeto só com os
pratos de arroz
  grao<-receitas[receitas$ingrediente=="grao", 1] #cria objeto só com os
pratos de grao(feijões)
  verde<-receitas[receitas$ingrediente=="verde", 1] #cria objeto só com os
acompanhamentos de ingredientes verdes
  outro<-receitas[receitas$ingrediente=="outro", 1] #cria objeto com
acompanhamentos classificados como 'outro'
  acomp.legs<-c(receitas[receitas$ingrediente=="verde", 1],
receitas[receitas$ingrediente=="outro", 1]) #cria objeto só com os
acompanhamentos 'verde' + 'outro'
  entrada<-receitas[receitas$tipo=="entrada", 1] #cria objeto só com as
entradas
  unico<-receitas[receitas$tipo=="prato unico", 1] #cria objeto só com os
pratos unicos
#Primeira interação com o usuário
cat("Vamos montar seu cardápio!") #começando a interação com uma afirmação
na tela
dieta<-readline("Você deseja um cardápio vegetariano?") #pergunta na tela
if(substr(dieta, 1, 1)=="s") #para o caso do usuário responder 'sim'(ou
qualquer palavra que comece com 's')
{
  #verifica se o usuário tem pelo menos 5 opcoes de prato principal
vegetariano no data.frame das receitas
  if(length(receitas[receitas$tipo=="prato principal" &
receitas$ingrediente=="veggie", 3])<5)
    stop("Você deve ter ao menos 5 opções de receitas veggie como prato
principal.") #interrompe a função e informa o usuário caso a condição não se
cumpra
  #Sorteio das receitas para um cardápio vegetariano
  pp.veggie<-sample(veggie, size=5, replace=F) #sorteio sem repetir o
prato principal
  ac.arroz<-sample(arroz, size=5, replace=T) #sorteio do arroz permite
repetição
  ac.grao<-sample(grao, size=5, replace=T) #sorteio do grao permite
repetição
  ac.outro<-sample(acomp.legs, size=5, replace=F) #sorteio do
acompanhamento sem repetir a receita
  ac.entrada<-sample(entrada, size=5, replace=T) #sorteio da entrada
permite repetição
  #Montagem do cardápio vegetariano
  segunda<-c(ac.entrada[1], pp.veggie[1], ac.arroz[1], ac.grao[1],
ac.outro[1]) #objeto cardapio da segunda-feira é feito com as receitas na
posição[1]
  terca<-c(ac.entrada[2], pp.veggie[2], ac.arroz[2], ac.grao[2],
ac.outro[2]) #objeto cardapio da terça-feira é feito com as receitas na
posição[2]
```

```
quarta<-c(ac.entrada[3], pp.veggie[3], ac.arroz[3], ac.grao[3],
ac.outro[3]) #objeto cardapio da quarta-feira é feito com as receitas na
posição[3]
quinta<-c(ac.entrada[4], pp.veggie[4], ac.arroz[4], ac.grao[4],
ac.outro[4]) #objeto cardapio da quinta-feira é feito com as receitas na
posição[4]
sexta<-c(ac.entrada[5], pp.veggie[5], ac.arroz[5], ac.grao[5],
ac.outro[5]) #objeto cardapio da sexta-feira é feito com as receitas na
posição[5]
cardapio.semana<-data.frame(segunda, terca, quarta, quinta, sexta) #cria
o data.frame com os cardápios dos dias da semana
}
if(substr(dieta, 1, 1)=="n") #para o caso do usuário responder 'não' (ou
qualquer palavra que comece com 'n')
{
#verifica se o usuário tem pelo menos 5 opcoes de prato principal carne,
ave e peixe, 5 acompanhamentos
#verde e outro no data.frame das receitas. Interrompe a função caso uma
destas condições não for cumprida.
#E mostra na tela mensagem alertando o usuário.
if(length(receitas[receitas$tipo=="prato principal" &
receitas$ingrediente=="carne", 3])<5)
stop("Você deve ter ao menos 5 receitas de carne como prato
principal.")
if(length(receitas[receitas$tipo=="prato principal" &
receitas$ingrediente=="ave", 3])<5)
stop("Você deve ter ao menos 5 receitas de ave como prato principal.")
if(length(receitas[receitas$tipo=="prato principal" &
receitas$ingrediente=="peixe", 3])<5)
stop("Você deve ter ao menos 5 receitas de peixe como prato
principal.")
if(length(receitas[receitas$ingrediente=="verde", 3])<5)
stop("Você deve ter ao menos 5 receitas de acompanhamento verde.")
if(length(receitas[receitas$ingrediente=="outro", 3])<5)
stop("Você deve ter ao menos 5 receitas de acompanhamento outro.")
#Sorteio das receitas para um cardápio não-vegetariano
pp.carne<-sample(carne, size=5, replace=F) #sorteio sem repetir o prato
principal de carne
pp.ave<-sample(ave, size=5, replace=F) #sorteio sem repetir o prato
principal de ave
pp.peixe<-sample(peixe, size=5, replace=F) #sorteio sem repetir o prato
principal de peixe
pp.porco<-sample(porco, size=5, replace=T) #sorteio do prato de porco
permite repetição
pp.veggie<-sample(veggie, size=5, replace=T) #sorteio do prato veggie
permite repetição
ac.arroz<-sample(arroz, size=5, replace=T) #sorteio do arroz permite
repetição
ac.grao<-sample(grao, size=5, replace=T) #sorteio do grão permite
repetição
```

```
ac.verde<-sample(verde, size=5, replace=F) #sorteio do verde não permite
repetição
ac.outro<-sample(outro, size=5, replace=F) #sorteio do outro não permite
repetição
ac.entrada<-sample(entrada, size=5, replace=T) #sorteio da entrada
permite repetição
#Segunda interação com o usuário, que só acontece caso ele responda
'não' à primeira pergunta.
cat("Sendo assim, você pode escolher entre as opções carne, ave, peixe,
porco e veggie.") #apresenta esta informação na tela
dia.semana<-c("segunda", "terça", "quarta", "quinta", "sexta") #cria
objeto com os dias da semana (caracteres)
escolha<-list() #cria lista vazia (onde serão colocadas as opções que
usuário fará para cada dia)
for (dia in dia.semana) #cria condição para dia=segunda, depois
dia=terça, depois dia=quarta etc até "sexta"
{
  escolha[dia]<-readline(paste("O que gostaria de comer na", dia, "?"))
#apresenta na tela pergunta para cada dia da semana
  #e lê a resposta do usuário (readline) que é colocada no objeto
'escolha' indexado pelo dia da semana
}
pp<-cbind() #cria objeto onde serão colocadas as opções(nome das
receitas) de prato principal
for(y in 1:5) #cria condição para y=1, depois y=2 etc até 5
{
  #Atribuindo os pratos sorteados conforme a escolha do usuário.
  #A escolha do usuário para segunda-feira está na posição [1] do objeto
escolha
  #a escolha de terça está na posição [2] e assim até a posição [5]
sexta-feira
  #O prato principal sorteado, e de acordo com a escolha do usuário,
será extraído dos objetos pp.carne
  #pp.ave, pp.peixe, pp.porco e pp.veggie, da mesma posição e será
colocado na mesma posição no objeto 'pp' (prato principal)
  if(escolha[[y]]=="carne") pp[y]<-pp.carne[y]
  if(escolha[[y]]=="ave") pp[y]<-pp.ave[y]
  if(escolha[[y]]=="peixe") pp[y]<-pp.peixe[y]
  if(escolha[[y]]=="porco") pp[y]<-pp.porco[y]
  if(escolha[[y]]=="veggie") pp[y]<-pp.veggie[y]
}
#Montagem do cardápio não-vegetariano
segunda<-c(ac.entrada[1], pp[1], ac.arroz[1], ac.grao[1], ac.verde[1],
ac.outro[1]) #objeto cardapio da segunda-feira é feito com as receitas na
posição[1]
terca<-c(ac.entrada[2], pp[2], ac.arroz[2], ac.grao[2], ac.verde[2],
ac.outro[2]) #objeto cardapio da terça-feira é feito com as receitas na
posição[2]
quarta<-c(ac.entrada[3], pp[3], ac.arroz[3], ac.grao[3], ac.verde[3],
ac.outro[3]) #objeto cardapio da quarta-feira é feito com as receitas na
posição[3]
```

```
    quinta<-c(ac.entrada[4], pp[4], ac.arroz[4], ac.grao[4], ac.verde[4],
ac.outro[4]) #objeto cardapio da quinta-feira é feito com as receitas na
posição[4]
    sexta<-c(ac.entrada[5], pp[5], ac.arroz[5], ac.grao[5], ac.verde[5],
ac.outro[5]) #objeto cardapio da sexta-feira é feito com as receitas na
posição[5]
    cardapio.semana<-data.frame(segunda, terca, quarta, quinta, sexta) #cria
o data.frame com os cardápios dos dias da semana
  }
  #Comparando o cardápio criado com o cardápio fornecido pelo usuário no
argumento 'cardapio.anterior'
  cat("Pratos repetidos em relação ao cardápio passado:\n") #explica na tela
o que será apresentado
  cardapiol<-as.list(cardapio.anterior) #converte 'cardapio.anterior' em
lista (antes data.frame)
  cardapio2<-as.list(cardapio.semana) #converte 'cardapio.semana' em lista
(antes data.frame)
  repete<-intersect(unlist(cardapiol), unlist(cardapio2)) #cria o objeto
'repete' que é a intersecção de cardapiol e cardapio2
  #para isso, é necessário simplificar cardapiol e cardapio2 em vetores
  if(length(repete) > 0){ #cria condição sobre o comprimento de 'repete', se
comprimento maior que zero
    cat(paste("\t", toString(repete), "\n")) #mostra na tela as receitas que
se repetem nos dois cardápios
  } else cat("Não há repetições.") #no caso da não haver repetição, ou seja,
'repete' tem tamanho igual a 0, mostra mensagem na tela
  #Terceira(ou segunda) interação com o usuário
  jantar<-readline("Uma sugestão para o jantar? (Esta opção não isola as
receitas veggie.)") #considera que o usuário pode escolher uma (somente uma)
opção de jantar
  if(substr(jantar, 1, 1)=="s") opcao.jantar<-sample(unico, size=1) #se o
usuário responder 'sim' (ou qualquer string que comece com 's')
  #então, uma opção de jantar é sorteada do objeto 'unico', que contém
receitas tipo prato único
  if(substr(jantar, 1, 1)=="n") opcao.jantar<-("Sem opção para jantar.") #se
o usuário responder 'nao' (ou qualquer palavra que comece com 'n')
  #então, a mensagem aparece na tela e também é adicionada ao objeto de
saída
  final<-cbind(cardapio.semana, jantar=c(opcao.jantar, rep("",
length(cardapio.semana$segunda)-1))) #cria objeto final com o cardapio
montado e a opção de jantar
  #ajusta o preenchimento de espaços para compatibilizar a opcao.jantar com
o resto
  return(final) #saída
  #Bon appetit!!!
}
```


****Arquivos da função final****

[cardapio.r](#)

exemplo de data.frame de entrada

[receitas_de_casa.csv](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:pruggiero:start 

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