Gabrielle Rizzato

Aluna de mestrado no Programa de Genética e Biologia Evolutiva no IB/USP. Trabalha com genética evolutiva de genes craniofaciais em primatas.

exec

Trabalho Final

Proposta A

Introdução

Professores universitários costumam dividir as notas finais em diversos parâmetros. Os parâmetros mais comuns que eles podem utilizar são: provas, trabalhos, seminários, frequência, participação e aulas práticas, sendo que geralmente cada um recebe um peso diferente na nota final. No fim eles ficam com milhares de planilhas calculando a nota de cada um dos alunos para cada um dos parâmetros e ainda tem que fazer uma conta com pesos diferentes para calcular a nota final. Não é algo super difícil de fazer, mas consome um bom tempo no excel. E como aprendemos em aula, pra que utilizar o excel quando podemos utilizar o R? Minha proposta é uma função que resolva isso tudo e facilite a vida do professor, onde eles só coloquem todas as planilhas com as notas e recebam a nota final.

Input

Nessa função, cada planilha, referente a um parâmetro diferente, vai entrar como um argumento, seguida do argumento referente ao peso daquele parâmetro. Cada planilha entrará como um arquivo csv salvo com tabulação, onde os nomes dos alunos ficam a esquerda e as colunas são as notas que eles receberam em cada prova/trabalho/etc. Em cada uma das planilhas é feita a média das linhas e em seguida a planilha inteira é multiplicada pelo argumento do seu peso na nota. No final todos os valores são somados

Output

No final a função retorna um data.frame com os nomes dos alunos, as médias parciais em cada um dos parâmetros e a nota final deles. Sendo que o data.frame vai ser automaticamente salvo no diretório para o caso do professor ser distraído e fechar o R sem salvar.

Argumentos

Será então um argumento para cada parâmetro e um argumento para cada peso

prov = provas

trab = trabalhos

semi = seminários

freq = frequência

part = participação em aula

aulp = aulas práticas

p1 = peso das provas

p2 = peso dos trabalhos

p3 = peso dos seminários

p4 = peso da frequência

p5 = peso da participação em aula

p6 = peso das aulas práticas

Todos os argumentos terão como padrão FALSE, porque é altamente improvável que todos os argumentos sejam preenchidos, dessa forma se algum parâmetro não estiver sendo avaliado é só não colocar na função.

A função será montada da seguinte maneira:

n.final(prov,p1,trab,p2,semi,p3,freq,p4,part,p5,aulp,p6)

Apesar dela ficar meio grande, como eu disse é muito improvável que todos os argumentos vão ser utilizados.

Eu preferi colocar o argumento do peso em seguida do argumento do parâmetro porque acho que as chances de confusão vão ser menores, mas também é possível colocar todos os argumentos do parâmetros primeiro e em seguida os do peso.

Melhorias

Se sobrar tempo no final, pensei também em junto com as notas a função no final fazer um gráfico com o rendimento dos alunos em cada um dos parâmetros, assim o professor pode ter uma ideia de onde os alunos estão encontrando mais dificuldade.

Comentários Julia

Oi Gabrielle,

Achei legal a proposta e viável. Duas coisas : Acho que poderia sim incorporar esse output gráfico na função. E de repente sua função

poderia fornecer também uma lista com os alunos que iriam para a 'recuperação' de acordo com um argumento que seria a nota mínima necessária. Ou uma lista com os alunos que estão em situação inferior à média da sala ?

Outra questão: Os argumentos estão dando peso as notas mas seria interessante pensar em como trabalhar com argumentos que dessem também flexibilidade ao professor. Caso ele esteja interessado por exemplo em calcular a nota final do aluno de forma relativa a maior nota da sala ou não ? Ou algumas o

Proposta B

Introdução

No processo de fabricação de cerveja artesanal são realizadas diversas etapas onde é necessário muitos cálculos para determinar a quantidade dos ingredientes. A etapas de fabricação são mostura (onde os grãos de malte são colocados em água aquecida para ocorrer a quebra do amido em açúcares menores formando o mosto), clarificação (separação do mosto dos grãos), clarificação (lavagem dos grãos com água quente, possibilitando uma maior extração do açúcar, fervura (mosto é fervido e o lúpulo é adicionado), resfriamento, fermentação (é adicionada a levedura), maturação e envase, onde é adicionado o priming, que é a adição de água com açúcar na cerveja que vai ser envasada, de forma que a levedura produza gás e gaseifique o líquido já dentro da garrafa, processo chamado de carbonatação. Ingredientes como o malte e o lúpulo são difíceis de calcular numa função fechada pois as quantidades variam de acordo com a receita sendo feita, além disso as receitas podem ser mudadas de acordo com o gosto do cervejeiro. Assim como a água, que a quantidade a ser adicionada varia com fatores como quanto é perdido pela evaporação na fervura. A fermentação e maturação das cervejas caseiras são feitas em grandes panelas chamadas fermentadores. Apesar do tamanho total do fermentador ser conhecido, fatores como a evaporação fazem com que o volume final sempre varie. Esses cálculos não dependem de receitas e por isso podem ser transformados numa função simples que ajude o cervejeiro caseiro em seus passos finais para saber quanta cerveja ele produziu, qual a quantidade de priming que ele vai precisar colocar no envase, e quantas garrafas ele vai ter no final.

Input

O input será o raio do fermentador e a altura que o líquido está na panela. O cálculo é feito a partir da fórmula do volume onde:

V = h*pi*r^2

Com o volume é calculado o priming, onde são 7 gramas de açúcar por litro de cerveja, diluídos numa quantidade de água que é 3ml de água pra cada grama de açúcar.

A quantidade de garrafas a ser utilizada é calculada dividindo o total de litros pelo tamanho da garrafa.

Output

O output vai retornar um dataframe com a quantidade de cerveja produzida, a quantidade de açúcar e água do priming a ser adicionado e quantas garrafas irá ser preciso para o envase.

Argumentos

Os argumentos vão ser:

r = raio

h = altura

bottle = volume da garrafa que será utilizada no envase

O raio e altura vão ter como padrão o centímetro. O volume da garrafa vai ter como padrão 600 ml, onde o cervejeiro vai poder mudar pra garrafas de 1L, 500 ml, 355 ml ou 275 ml.

A função vai ser estruturada da seguinte maneira

cerva(r,h,bottle="600")

Comentários Julia

Acho que a ideia de lidar com a fabricação de cervejas abre espaço para muitas funções e muitas combinações possíveis. Porém no final fiquei com a impressão que a função executa a fórmula (h*pi*r^2) e depois uma divisão para calcular a quantidade de açucar e quantas garrafas.

Se for seguir com a ideia de trabalhar com a função voltada para as cervejas, poderia por exemplo incluir uma entrada de dados em que o usuário coloque a quantidade de malte, lupulo, trigo, cevada, frutas (etc como preferir) e a função analise que cerveja poderia ser fabricada (IPA, trigo, larger, stout, etc) ou no caso de mais de uma qual renderia maior quantidade em litros.

Ambas propostas são viáveis. A proposta A exige que alguns detalhes sejam melhor elaborados e a B falta mais corpo para que ganhe um pouco mais de complexidade. Talvez seja mais proveitoso focar na proposta A e dar o capricho final nessa proposta. ;)

Bjs

Oi Júlia! Obrigada pelas dicas. Eu realmente tava querendo fazer a proposta B porque gosto muito do assunto. O que você falou não da pra fazer porque as receitas são muito variáveis e não teria como calcular algo assim. O que eu posso fazer é adicionar o cálculo o IBU (amargor) que é uma conta um pouco mais complicada. Você acha que seria o suficiente pra incrementar a função? Eu to procurando várias outras fórmulas que eu também poderia colocar junto e to tentando deixar mais redondinha a

função. O que você acha?

Comentários Julia

Oi Gabrielle,

Acho que incluir outros cálculos pode ser interessante até porque isso pode ser uma decisão do usuário. Se ele quer amargor a função segue um caminho, se ele nao quer ela segue outro caminho e por ai vai para outras características que você incluir.

Mesmo as receitas variando muito você pode brincar com elas escolhendo uma referência para a sua função e pronto. Pode na documentação deixar claro que a estrutura de uma IPA por exemplo foi pensada de acordo com o site X ou o livro Y. Mas não precisa utilizar essa ideia, foi apenas um pensamento de como dar um pouco mais de corpo a função, para que ela não seja apelas o cálculo (h*pi*r^2) e depois uma divisão para calcular a quantidade de açucar e quantas garrafas que foi o que me preocupou.

Tente pensar nos argumentos que uma pessoa gostaria de utilizar (voce como usuária por exemplo). A função pode ter um cálculo simples, não precisa de multiplas fórmulas, mas é a flexibilidade dos argumentos que torna necessário utilizar 'IFs/ELSE' por exemplo ou a capacidade de processar multiplos dados de uma vez que torna necessário o uso de ciclos de 'FOR' por exemplo, deixando a função mais bonita sem ter que se tornar uma algebra cabeluda ;)

Veja o que acha e consegue aprimorar e vamos nos falando hoje e amanhã!

Comentários Julia 09/junho

Oi Gabrielle, como está indo ? Lembre de atualizar a sua proposta para podermos deixar acordado hoje ;)

Qualquer coisa pode me procurar no juliambmolina@gmail.com

Beijos

Oi Júlia, desculpe não responder na sexta. Olha, eu acho que to indo bem, acho que com o cálculo do IBU e o do priming (aprimorado) a função vai ficar bem legal. Não se preocupe que eu vou conseguir inserir if/else e acho que for também.

Eu adicionei como argumentos:

Gravidade original e gravidade final, que são as densidades da cerveja antes e depois da fervura para cálculo de grau álcoolico e IBU (amargor)

Também coloquei como argumentos o valor de alfa ácidos no lúpulo e peso de lúpulo adicionado: esses valores junto com uma matriz que eu inseri na função (e a gravidade original) são utilizados para calcular o IBU (amargor da cerveja).

Vou colocar um ciclo para utilização de mais de um tipo de lúpulo.

Também vou colocar um gráfico para apoio visual do priming.

O que você acha de todos esses cálculos a mais?

Se eu tiver algum problema te mando a dúvida por email, pode ser?

Muito obrigada

Beijos

Comentários Julia - 18/06

Oi Gabrielle, tudo bem ?

Acho que sua proposta ficou bem formulada no final com os ajustes.
Ter todos estes cálculos não é problema algum, só pode somar ;)
Apenas tome cuidado para não se complicar demais! De toda forma, parece que você está caminhando bem.

Qualquer coisa pode mandar e-mail sim ! Ou manda lá no forum.

Beijão!!

Help da Função

```
breja      package:nenhum          R Documentation

Dados de uma cerveja artesanal

Description

breja() calcula o volume de cerveja produzida em uma brassagem (processo de
fabricação de cerveja), quantas garrafas serão necessárias de acordo com o
tamanho da garrafa utilizada, qual o IBU e o grau alcoólico final da
cerveja, e quanto açuçar e água é necessário no envase para obter a
carbonatação necessária no produto final.

Usage

breja =
function(r,h,OG,FG,bottle,t,alfa1,glup1,alfa2="",glup2="",alfa3="",glup3="",
tlup="f",co2=2,tfer)

Arguments

r             um valor numérico indicando o raio do recipiente onde está a
cerveja após a fermentação
```

```
h             um valor numérico indicando a altura do recipiente onde está a
cerveja após a fermentação
OG            um charactere indicando a Original Gravity (Gravidade
Original), medida com um densímetro a 20ºC antes da fermentação. É a medida
das substâncias fermentáveis e não fermentáveis do mosto antes da
fermentação.
FG            um charactere indicando a Final Gravity (Gravidade Final),
medida com um densímetro a 20ºC após a fermentação. É a medida das
substâncias não fermentáveis, após a fermentação do mosto.
bottle        um valor numérico indicando o volume da garrafa a ser
utilizada no envase da cerveja
t             um valor numérico indicando o tempo de fervura do lúpulo
principal utilizado na cerveja
alfa1         um valor numérico indicando a porcentagem de alfa ácidos em
decimais do lúpulo principal utilizado
glup1         um valor numérico indicando a quantidade em gramas do lúpulo
principal utilizado
t2            um valor numérico indicando o tempo de fervura do segundo
lúpulo utilizado na cerveja (apenas em caso de utilização de dois lúpulos)
alfa2         um valor numérico indicando a porcentagem de alfa ácidos em
decimais do segundo lúpulo utilizado na cerveja (apenas em casos de
utilização de dois lúpulos)
glup2         um valor numérico indicando a quantidade em gramas do segundo
lúpulo utilizado na cerveja (apenas em casos de utilização de dois lúpulos)
t3            um valor numérico indicando o tempo de fervura do terceiro
lúpulo utilizado na cerveja (apenas em casos de utilização de três lúpulos)
alfa3         um valor numérico indicando a porcentagem de alfa ácidos em
decimais do terceiro lúpulo utilizado na cerveja (apenas em casos de
utilização de três lúpulos)
glup3         um valor numérico indicando a quantidade em gramas do terceiro
lúpulo utilizado na cerveja (apenas em casos de utilização de três lúpulos)
tlup          um character indicando o tipo de lúpulo utilizado
co2           um valor numérico indicando o volume final de CO2 desejado na
carbonatação da cerveja
tfer          um valor numérico indicando a temperatura na qual ocorreu a
fermentação da cerveja

Details

No processo de fabricação de cerveja artesanal são realizadas diversas
etapas onde é necessário muitos cálculos para determinar a quantidade dos
ingredientes e resultados finais obtidos. As etapas de fabricação são
mostura (onde os grãos de malte são colocados em água aquecida para ocorrer
a quebra do amido em açúcares menores formando o mosto), clarificação
(separação do mosto dos grãos), lavagem (lavagem dos grãos com água quente,
possibilitando uma maior extração do açúcar), fervura (o mosto é fervido e o
lúpulo é adicionado), resfriamento, fermentação (onde é adicionada a
levedura), maturação e envase, onde é adicionado o priming, que é a adição
de água com açúcar na cerveja que vai ser envasada, de forma que a levedura
utilize esse açucar para produzir gás para gaseificar o líquido já dentro da
garrafa, processo chamado de carbonatação. A fermentação e maturação das
```

cervejas caseiras são feitas em grandes panelas chamadas fermentadores. Apesar do tamanho total do fermentador ser conhecido, fatores como a evaporação na fervura fazem com que o volume final sempre varie.

Para calcular o volume deve ser inserido apenas o raio (r) e a altura (h) OU diretamente o volume (v) caso o usuário tenha transferido o produto do fermentador para algum tipo de recipiente com graduação de volume.

A relação entre a densidade do mosto ou da cerveja e a densidade da água é conhecida como gravidade específica. Assim, uma gravidade específica de 1.010 significa que a solução pesa 10/1000=1/100=1% mais do que o mesmo volume em água. A gravidade específica medida antes da fermentação é conhecida como Original Gravity (OG) e a medida após a fermentação é conhecida por Final Gravity (FG), ambas devem ser medidas com um densímetro antes e depois da fermentação respectivamente. Essa unidade depende da temperatura, porque a densidade da água e de soluções aquosas varia com a temperatura. Em geral, os densímetros estão calibrados para 20ºC. Por causa do cálculo do IBU o valor da gravidade original pode ter que ser aproximado como explicado mais abaixo.

O comparativo entre essas duas medidas, OG e FG, determina a potência alcoólica da cerveja. A medida de álcool na cerveja é dada em Alcohol by Volume (ABV). De maneira geral, são classificadas como cervejas de baixo teor alcoólico as que variam entre 0,5 – 2,0%. De médio teor entre 2,0 – 4,5%. Cervejas acima de 4,5% são consideradas de alto teor alcoólico.

Bottle deve ser escolhido entre os tamanhos de garrafa de volume 10000, 600, 500, 355 e 275, pois esses são os tamanhos padrões comercializados no mercado.

O lúpulo possui uma substância chamada alfa ácido que é encontrada em suas flores e é a fonte de amargor da cerveja. Os alfa ácidos não são hidrossolúveis, eles precisam ser isomerizados e transformados em iso-alfa-ácidos através do calor e por isso são adicionados na fervura. A forma de medir o nível de amargor de uma cerveja é através da unidade padrão de medida chamada “IBU” (International Bitterness Units). Os valores de IBU obtidos em uma cerveja correspondem à concentração de iso-alfa-ácidos diluidos na cerveja, seguindo a regra 1 IBU = 1mg de iso-alfa-ácido por litro. Para se calcular o amargor de uma cerveja é preciso levar em conta a quantidade de lúpulo utilizada, o índice de alfa ácido do lúpulo, a densidade do mosto (OG), o tempo de fervura e o volume final de cerveja. O limite técnico para IBU é de cerca de 100, alguns tentaram ultrapassar este número, mas não há nenhuma medida real depois de 100 IBUs quando se trata de gosto limite. As cervejas comerciais nacionais, do estilo Standard Lager, possuem entre 8 e 15 IBU, enquanto as Premium Lagers possuem entre 15 e 25 IBU. Uma cerveja bem lupulada como um IPA (India Pale Ale) terá uma classificação IBU mais elevada, como um 75, enquanto uma cerveja de malte como uma Stout geralmente tem uma classificação de IBU mais baixo, em torno de 30-40. Cervejas como as Lambics, que são feitas com frutas em sua composição, possuem IBU de no máximo 10.

Existem pelo menos quatro métodos para se calcular o IBU de uma cerveja, elaborados respectivamente por Glenn Tinseth, Jackie Rager, Mark Garetz e Ray Daniels, variando entre eles principalmente na forma como eles avaliam a Utilização (U). Este fator representa a eficiência de isomerização dos alfa ácidos. Dos 4 métodos, o método de Tinseth é o mais utilizado no mundo e considerado por muitos o método mais preciso de todos, por isso é o método utilizado nessa função.

O método de Glenn Tinseth possui uma tabela de Utilização de alfa ácidos que se dá em função da OG (Gravidade Original) e do tempo de fervura do lúpulo. Como a tabela possui valores de temperatura e OG pré-determinados o usuário da função precisa ter o cuidado de manter o tempo de fervura em um desses valores. Caso a OG encontrada seja um valor intermediário entre dois valores pré-determinados é permitido uma aproximação para o valor mais próximo.

O tempo de fervura deve ser um dentre esses valores:
0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,60,70,80,90
A gravidade original (OG) deve ser um dentre esses valores:
"1.030","1.040","1.050","1.060","1.070","1.080","1.090","1.100","1.110","1.120","1.130"

Em caso de adição de mais de um lúpulo eles podem ser adicionados em diferentes tempos de fervura. Nesses casos, o tempo 2 (t2) e o tempo 3 (t3) também devem ser um dentre os valores já citados. Caso todos os lúpulos tenham o mesmo tempo de fervura, esse mesmo tempo tem que ser repetido no t2 e/ou t3.

A porcentagem de alfa ácidos do lúpulo deve ser dada em decimais. Se por exemplo o lúpulo possui 12,5% de álfa ácidos, deve ser colocado 0,125. Da mesma maneira, a quantidada de lúpulo deve ser dada em gramas.

Os valores calculados pelo método de Tinseth consideram lúpulos em flor despejados soltos na panela de fervura. E de acordo com estudiosos, os lúpulos em pellet são cerca de 10% mais eficientes do que o lúpulo em flor. A mesma relação, de cerca de 10% é atribuída para o uso de hopbags, ou seja, se ao invés de jogar o lúpulo direto na fervura colocar na panela dentro de uma bolsa, nesse caso deverá ser considerado um déficit de 10% na eficiência.

Sendo assim no tipo de lúpulo utilizado deve ser escolhido dentre:
"f" para lúpulos em flor direto na panela, ou pellets dentro de uma hopbag
"b" para lúpulos em flor dentro de uma bag
"p" para lúpulos em pellet direto na panela

O priming consiste em adicionar uma quantidade extra de açúcares fermentáveis na cerveja para que as leveduras residuais iniciem uma refermentação na garrafa, gerando CO2 e a consequente carbonatação da cerveja. Esse açúcar tem que ser diluído em água e fervido por cerca de 5 minutos antes de ser adicionado na cerveja que vai ser envasada. Podemos entender por carbonatação a quantidade de dióxido de carbono incorporada a um líquido, nesso caso a cerveja, expressa em volumes de CO2. Esse volume de

CO2 representa a quantidade determinada pelo volume do gás que pode ser retirado de uma determinada litragem de cerveja à temperatura de 20°C em um ambiente com uma atmosfera de pressão. Por exemplo, 1 litro de cerveja com 2,5 volumes eliminaria 2,5 litros de CO2 à temperatura de 20°C em um ambiente com uma atmosfera, isso se absolutamente todo o gás incorporado fosse retirado.

Tabela de CO2 adequado para alguns estilos:

ESTILO DE CERVEJA	VOLUME DE CO2
Ales Inglesas	1,5 – 2,0
Porter, stout	1,7 – 2,3
Ales belgas em geral	1,9 – 2,4
Lagers em geral	2,2 – 2,7
Ales Americanas	2,2 – 2,7
Lambics	2,4 – 2,8
Lambics com frutas	3,0 – 4,5
Cervejas de Trigo	3,3 – 4,5

Value

A função breja retorna um dataframe com as informações:

Volume Total - Volume total de cerveja produzido na brassagem dado em litros
Garrafas - Número de garrafas a ser utilizado no envase de acordo com o volume desejado da garrafa
IBU - International Bitterness Units, o amargor da cerveja, variando de 0 a 100
ABV - Alcohol by Volume, grau alcóolico da cerveja dado em porcentagem
Açucar do Priming - Quantidade de açúcar necessária para o priming dada em gramas
Água do Priming - Volume de água necessário para o priming dado em ml

Warning

As informações necessárias para a função são recolhidas em diferentes etapas da brassagem. É necessário ir anotando ao longo do processo para usá-las todas juntas na função antes do processo final de envase.

Etapas:

Fervura - anotar a quantidade de gramas de cada lúpulo adicionado (valores de glup), o tempo de fervura de cada um (valores de t) e qual o tipo de lúpulo utilizado (valor de tlup). O valor do alfa ácido é encontrado na embalagem (valores de alfa).
Após o resfriamento (mosto a cerca de 20ºC) e antes de inocular o fermento - medir com um densímetro e anotar a OG (Original Gravity)
Fermentação - anotar a temperatura na qual a fermentação está ocorrendo
Após a fermentação - medir com um densímetro e anotar a FG (Final Gravity)
Maturação - medir o raio do recipiente da maturação e a altura que a cerveja

```
se encontra no recipient(ou caso de um recipiente graduado apenas anotar o
volume)
Decidir qual o volume de CO2 desejado

Utilizar todos os dados anotados na função para obter o resultado da
quantidade de priming e de garrafas necessária para o evase, além do IBU e
ABV final de sua cerveja.

Envasar e esperar a fermentação secundária ocorrer.

Beber, cheers!


Author (s):

Gabrielle Azevedo Rizzato
gabriellerizzato@gmail.com


Examples

####  Uma IPA feita com 3 tipos de lúpulos em flores, dois de amargor que
ficaram mais tempo, e um aromático colocado no final, e envasada em garrafas
de 600 ml
breja(v=30,OG="1.050",FG="1.010",bottle=600,t=60,alfa1=0.125,glup1=30,t2=60,
alfa2 = 0.15,glup2 = 20,t3=33, alfa3=0.35,glup3 = 10,tlup="f",co2=2.7,
tfer=17)

### Uma Stout feita com apenas um lúpulo em hopbag e envasada em garrafas de
500 ml
breja(r=15,h=25,OG="1.090",FG="1.010",bottle=500,t=42,alfa1=0.14,glup1=30,tl
up="b",co2=1.9,tfer=18)

####  Uma Pale Ale feita com dois lúpúlo em pellet e envasada em garrafas de
1000 ml
breja(r=20,h=30,OG="1.060",FG="1.010",bottle=1000,t=33,alfa1=0.167,glup1=40,
t2=15,alfa2=0.1,glup2=40,tlup="p",co2=2.2,tfer=20)

####  Uma Lager feita com apenas um lúpúlo em flor e envasada em garrafas de
1000 ml
breja(v=50,OG="1.040",FG="1.010",bottle=1000,t=33,alfa1=0.12,glup1=30,tlup="
f",co2=2.5,tfer=12)
```

Código da Função

```
breja =
function(r=NA,h=NA,v=NA,OG,FG,bottle,t,alfa1,glup1,t2=NA,alfa2=NA,glup2=NA,t
3=NA,alfa3=NA,glup3=NA,tlup="p",co2=2,tfer)
{
```

```
  if(!is.na(v))    #como o r e o h são argumentos mutuamentes exclusivos ao
v, checar se o volume é diferente de NA, ou seja se tem o dado do volume
       {
         if(!is.na(r) | !is.na(h))   #se o volume for diferente de NA,
checar se o raio ou o volume também são diferentes de NA
              {
                stop("nao colocar raio e altura se o volume foi colocado")
#se o raio e o volume também forem diferentes de NA, parar a funcao porque
nao pode ter raio e altura se já tem o volume
              }
         if(class(v)!="numeric")   #checar se a classe do volume e diferente
de classe numerica
              {
                stop("volume precisa ser um valor numerico")  #parar a
funcao e mostrar mensagem de erro
              }
         volume=v   #se o v é diferente de NA então o volume é diretamente o
valor de v
         volume.ml = volume*1000  #volume em ml é o valor do volume
multiplicado por 1000
       }
  else    #se o volume for NA
       {
         if(is.na(r) | is.na(h))  #checar se o raio e a altura tabmém sao
NA, porque como nao foi dado o volume, tem que ter sido fornecido o raio e
altura
             {
               stop("valores de raio e altura ausentes") #se nao tiver raio
e altura parar a funcao e emitir mensagem de erro dizendo que raio e alturas
estao ausentes
             }
         if(class(r)!="numeric")   #checar se o valor do raio e um vetor
numerico
             {
               stop("o raio precisa ser um valor numerico")   #se nao for um
valor numerico parar a funcao e emitir mensagem de erro
             }
         if(class(h)!="numeric")   #checando se o valor da altura e um vetor
numerico
             {
               stop("a altura precisa ser um valor numerico")  #se nao for
um valor numerico parar a funcao e emitir mensagem de erro
             }
         else          #se os valores de raio e altura estiverem corretos
calcula o volume de cerveja que tem na panela
             {
               raio=r   #r e o valor do raio
               altura=h   #h e o valor da altura
               volume = (pi*(raio^2)*altura)/1000  #formula do volume e
igual a pi vezes raio ao quadrado vezes a altura dividido por 1000
```

```
               volume.ml = volume*1000  #volume em ml é o valor do volume
multiplicado por 1000
             }
       }
  if(class(OG)!="character")   #checando se a classe de OG e diferente da
classe caractere
       {
         stop("a gravidade original precisa ser dada como um caractere")
#se for diferente de caractere parar a funcao e emitir mensagem de erro
       }
  gravidades =
c("1.030","1.040","1.050","1.060","1.070","1.080","1.090","1.100","1.110","1
.120","1.130")    #vetor gravidades contem a concatenacao dos valores
permitidos para gravidade original
  if(OG %in% gravidades==FALSE)  #checar se valor da gravidade original e um
valor que dentro do vetor gravidades e igual a falso, ou seja se for uma
gravidade original que nao esta dentro dos valores permitidos
       {
         stop("a gravidade original foi especificada incorretamente, leia o
help da funcao")  #parar a funcao e emitir mensagem de erro pedindo pra ler
o help da funcao onde o usuario encontrara quais sao os valores permitidos
para OG
       }
  if(class(FG)!="character")  #checando se a classe de FG e diferente da
classe caractere
       {
         stop("a gravidade final precisa ser dada como um caractere")   #se
for diferente de caractere parar a funcao e emitir mensagem de erro
       }
  else   #se todas as classes estiverem corretas e o valor de OG esta
correto continuar a funcao
       {
         OG.n= as.numeric(OG)  #transformar a OG em um valor numerico
         FG.n= as.numeric(FG)  #transformar a FG em um valor numerico
         grau = (OG.n-FG.n)*134  #calculo do grau alcoolico a partir da
diferenca da gravidade original e gravidade final como valores numericos
multiplicado por 134
       }

  if(class(bottle)!="numeric")  #checando se a classe do valor das garrafas
e diferente da classe numerico
       {
         stop("bottle precisa ser um valor numerico")  #se nao for um valor
numerico parar a funcao e emitir mensagem de erro
       }
  tamanhos = c(1000,600,500,355,275)  #vetor com a concatenacao dos valores
permitidos para o tamanho de garrafas
  if(bottle %in% tamanhos ==FALSE)  #checar se o tamanho das garrafas e um
valor que dentro do vetor garrafas é igual a falso, ou seja se o tamanho das
garrafas nao esta dentro dos tamanhos permitidos
       {
```

```
         stop("bottle nao e um valor valido de garrafa, leia o help da
funcao")  #parar a funcao e emitir mensagem de erro pedindo pra ler o help
da funcao onde o usuario encontrara os valores permitidos para o tamanho das
garrafas
       }
  else  #se o tamanho das garrafas estiver correto
       {
          if(bottle==1000)   #se o tamanho da garrafa for igual a 1000
               {
                 garrafas=volume.ml/1000  #o numero de garrafas que vao ser
utilizadas e igual ao volume da cerveja em ml dividido por 1000
               }
          if(bottle==600)  #se o tamanho da garrafa for igual a 6000
               {
                 garrafas=volume.ml/600  #o numero de garrafas que vao ser
utilizadas e igual ao volume da cerveja em ml dividido por 600
               }
          if(bottle==500)  #se o tamanho da garrafa for igual a 5000
               {
                 garrafas=volume.ml/500  #o numero de garrafas que vao ser
utilizadas e igual ao volume da cerveja em ml dividido por 500
               }
          if(bottle==355)  #se o tamanho da garrafa for igual a 355
               {
                 garrafas=volume.ml/355  #o numero de garrafas que vao ser
utilizadas e igual ao volume da cerveja em ml dividido por 355
               }
          if(bottle==275)  ##se o tamanho da garrafa for igual a 275
               {
                 garrafas=volume.ml/275  #o numero de garrafas que vao ser
utilizadas e igual ao volume da cerveja em ml dividido por 275
               }
        }
  if(class(t)!="numeric") #checando se a classe do tempo de fervura e uma
classe diferente da classe numerica
       {
         stop("o tempo precisa ser um valor numerico")    #se for diferente
de numerico parar a funcao e emitir mensagem de erro
       }
  tempos =
c(0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,60,70,80,90)
#vetor com a concatenacao dos valores permitidos para o tempo de fervura
  if(t %in% tempos ==FALSE)  #checar se o tempo é um valor que dentro do
vetor tempos e igual a falso, ou seja se for um tempo de fervura que nao
esta dentro dos tempos permitidos
       {
         stop("o tempo de fervura foi especificado incorretamente, leia o
help da funcao")     #parar a funcao e emitir mensagem de erro pedindo pra
ler o help da funcao onde o usuario encontrara os valores permitidos para o
tempo de fervura
```

```
       }
  if(!(class(alfa1)=="numeric" | class(alfa2)=="numeric" |
class(alfa3)=="numeric"))  #checando se a classe dos alfa acidos sao
diferentes da classe numerica
       {
         stop("o alfa acido precisa ser um valor numerico")  #se for
diferente de numerico parar a funcao e emitir mensagem de erro
       }
  if(!(class(glup1)=="numeric" | class(glup2)=="numeric" |
class(glup3)=="numeric"))  #checando se a classe das quantidades de lupulo
sao diferentes da classe numerica
       {
         stop("o peso do lupulo precisa ser valor numerico")  #se for
diferente de numerico parar a funcao e emitir mensagem de erro
       }
  else  #se as classes e valores estiverem corretos continuar a funcao
       {
          tabela = matrix(c(0.000, 0.034, 0.065, 0.092, 0.116, 0.137, 0.156,
0.173, 0.187, 0.201, 0.212, 0.223, 0.232, 0.240, 0.247, 0.253, 0.259, 0.264,
0.269, 0.273, 0.276, 0.285, 0.291, 0.295, 0.301,    #cria a tabela de
utilizacao de alfa acidos para calculo de IBU pelo metodo de Glenn Tinseth,
na qual o valor de utilizacao de alfa acidos e dado considerando a gravidade
original por tempo de fervura
                            0.000, 0.031, 0.059, 0.084, 0.106, 0.125, 0.142,
0.158, 0.171, 0.183, 0.194, 0.203, 0.212, 0.219, 0.226, 0.232, 0.237, 0.241,
0.246, 0.249, 0.252, 0.261, 0.266, 0.270, 0.275,
                            0.000, 0.029, 0.054, 0.077, 0.097, 0.114, 0.130,
0.144, 0.157, 0.168, 0.177, 0.186, 0.194, 0.200, 0.206, 0.212, 0.216, 0.221,
0.224, 0.228, 0.231, 0.238, 0.243, 0.247, 0.252,
                            0.000, 0.026, 0.049, 0.070, 0.088, 0.105, 0.119,
0.132, 0.143, 0.153, 0.162, 0.170, 0.177, 0.183, 0.189, 0.194, 0.198, 0.202,
0.205, 0.208, 0.211, 0.218, 0.222, 0.226, 0.230,
                            0.000, 0.024, 0.045, 0.064, 0.081, 0.096, 0.109,
0.120, 0.131, 0.140, 0.148, 0.155, 0.162, 0.167, 0.172, 0.177, 0.181, 0.184,
0.188, 0.190, 0.193, 0.199, 0.203, 0.206, 0.210,
                            0.000, 0.022, 0.041, 0.059, 0.074, 0.087, 0.099,
0.110, 0.120, 0.128, 0.135, 0.142, 0.148, 0.153, 0.158, 0.162, 0.165, 0.169,
0.171, 0.174, 0.176, 0.182, 0.186, 0.188, 0.192,
                            0.000, 0.020, 0.038, 0.054, 0.068, 0.080, 0.091,
0.101, 0.109, 0.117, 0.124, 0.130, 0.135, 0.140, 0.144, 0.148, 0.151, 0.154,
0.157, 0.159, 0.161, 0.166, 0.170, 0.172, 0.176,
                            0.000, 0.018, 0.035, 0.049, 0.062, 0.073, 0.083,
0.092, 0.100, 0.107, 0.113, 0.119, 0.124, 0.128, 0.132, 0.135, 0.138, 0.141,
0.143, 0.145, 0.147, 0.152, 0.155, 0.157, 0.161,
                            0.000, 0.017, 0.032, 0.045, 0.056, 0.067, 0.076,
0.084, 0.091, 0.098, 0.103, 0.108, 0.113, 0.117, 0.120, 0.123, 0.126, 0.129,
0.131, 0.133, 0.135, 0.139, 0.142, 0.144, 0.147,
                            0.000, 0.015, 0.029, 0.041, 0.052, 0.061, 0.069,
0.077, 0.083, 0.089, 0.094, 0.099, 0.103, 0.107, 0.110, 0.113, 0.115, 0.118,
0.120, 0.121, 0.123, 0.127, 0.130, 0.132, 0.134,
                            0.000, 0.014, 0.026, 0.037, 0.047, 0.056, 0.063,
```

```
0.070, 0.076, 0.082, 0.086, 0.091, 0.094, 0.098, 0.101, 0.103, 0.105, 0.108,
0.109, 0.111, 0.112, 0.116, 0.119, 0.120, 0.123), nrow=25, ncol=11)
          rownames(tabela) =
c(0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,60,70,80,90,120)
#coloca como nome das linhas os tempos determinado pela tabela do metodo
          colnames(tabela) = c("1.030", "1.040", "1.050", "1.060", "1.070",
"1.080", "1.090", "1.100", "1.110", "1.120", "1.130")  #coloca como nome das
colunas as gravidades originais determinadas pela tabela do metodo
          utili = tabela[rownames(tabela)==t, colnames(tabela)==OG]
#calcula a utilizacao de alfa acidos considerando o tempo de fervura e a
gravidade original informados nos argumentos
          if(is.na(t2) & is.na(t3))   #se o tempo2 e o tempo 3 foram iguais
a NA, ou seja se so tem um lupulo e um valor de fervura
                {
                  IBU1=(utili*alfa1*glup1*1000)/volume  #o valor do IBU1 e
calculado com a utilizacao desse lupulo dada pelo tempo de fervura e OG,
vezes o valor de alfa acido, vezes peso em gramas, vezes 1000, dividido pelo
volume
                  IBUtotal=IBU1  #como so tem um lupulo o valor de IBU total
e igual ao valor de IBU1
                }
          if(!(is.na(t2) | is.na(alfa2) | is.na(glup2)))  #se o valor do
tempo2, o valor de alfa2 e o valor do peso em gramas2 forem diferentes de NA
                {
                   if(t2 %in% tempos ==FALSE)  #checar se o t2 e um valor
que dentro do vetor tempos e igual a falso, ou seja se for um tempo de
fervura que nao esta dentro dos tempos permitidos
                      {
                        stop("o tempo de fervura 2 foi especificado
incorretamente, leia o help da funcao")     #parar a funcao e emitir
mensagem de erro pedindo pra ler o help da funcao onde o usuario encontrara
os valores permitidos para o tempo de fervura
                      }
                  utili2 = tabela[rownames(tabela)==t2,
colnames(tabela)==OG]  #calcula a utilizacao de alfa acidos 2 considerando o
tempo de fervura 2 e a gravidade original informados nos argumentos
                  IBU1=(utili*alfa1*glup1*1000)/volume  #o valor do IBU1 e
calculado com a utilizacao desse lupulo dada pelo tempo de fervura e OG,
vezes o valor de alfa acido, vezes peso em gramas, vezes 1000, dividido pelo
volume
                  IBU2=(utili2*alfa2*glup2*1000)/volume  #o valor do IBU2 e
calculado com a utilizacao do segundo lupulo dada pelo tempo de fervura 2 e
OG, vezes o valor de alfa acido 2, vezes peso em gramas 2, vezes 1000,
dividido pelo volume
                  IBUtotal=IBU1+IBU2  #IBU total e igual ao valor de IBU1
mais o valor de IBU2
                }
          if(!(is.na(t3) | is.na(alfa3) | is.na(glup3)))  #se o valor do
tempo3, o valor de alfa3 e o valor do peso em gramas3 forem diferentes de NA
                {
```

```
                    if(t3 %in% tempos ==FALSE)  #checar se o t3 e um valor
que dentro do vetor tempos e igual a falso, ou seja se for um tempo de
fervura que nao esta dentro dos tempos permitidos
                       {
                         stop("o tempo de fervura 3 foi especificado
incorretamente, leia o help da funcao")     #parar a funcao e emitir
mensagem de erro pedindo pra ler o help da funcao onde o usuario encontrara
os valores permitidos para o tempo de fervura
                       }
                    utili3 = tabela[rownames(tabela)==t3,
colnames(tabela)==OG]   #calcula a utilizacao de alfa acidos 3 considerando
o tempo de fervura 3 e a gravidade original informados nos argumentos
                    IBU1=(utili*alfa1*glup1*1000)/volume  #o valor do IBU1 e
calculado com a utilizacao desse lupulo dada pelo tempo de fervura e OG,
vezes o valor de alfa acido, vezes peso em gramas, vezes 1000, dividido pelo
volume
                    IBU2=(utili2*alfa2*glup2*1000)/volume  #o valor do IBU2 e
calculado com a utilizacao do segundo lupulo dada pelo tempo de fervura 2 e
OG, vezes o valor de alfa acido 2, vezes peso em gramas 2, vezes 1000,
dividido pelo volume
                    IBU3=(utili3*alfa3*glup3*1000)/volume  #o valor do IBU3 e
calculado com a utilizacao do terceiro lupulo dada pelo tempo de fervura 3 e
OG, vezes o valor de alfa acido 3, vezes peso em gramas 3, vezes 1000,
dividido pelo volume
                    IBUtotal=IBU1+IBU2+IBU3  #o valor do IBU total e igual ao
IBU1 mais o IBU2 mais o IBU3
                  }
          if(class(tlup)!="character")   #checando se a classe do tipo de
lupulo e diferente da classe caractere
                  {
                    stop("tipo de lupulo especificado incorretamente, leia o
help da funcao")  #parar a funcao e emitir mensagem de erro pedindo pra ler
o help da funcao onde o usuario encontrara os tipos de lupulo permitidos
                  }
          else  #se a classe do tipo de lupulo estiver certa continuar a
funcao
                  {
                      if(tlup=="f")  #se o tipo de lupulo for igual a flor
direta ou o pellet dentro de uma hopbag
                            {
                              IBU=IBUtotal  #o IBU e igual ao IBU total
                            }
                      if(tlup=="b")  #se o tipo de lupulo for igual a flor
numa hopbag
                            {
                              IBU=IBUtotal-(IBUtotal*0.1)  #o IBU e igual ao
IBU total menos 10% porque ha uma perda de amargor com a bag
                            }
                      if(tlup=="p")  #se o tipo de lupulo for igual ao pellet
direto
                            {
```

```
                              IBU=IBUtotal+(IBUtotal*0.1) #o IBU e igual ao
IBU total mais 10% porque o pellet e mais forte que a flor
                            }
                }
       }
  if(class(co2)!="numeric")   #checando se a classe do CO2 e diferente da
classe numerica
       {
         stop("o volume de CO2 final desejado precisa ser um valor numerico,
leia o help da funcao")     #parar a funcao e emitir mensagem de erro
pedindo pra ler o help da funcao onde o usuario encontrara os valores de
volume de CO2 de acordo com os estilos de cerveja
       }
  if(class(tfer)!="numeric")   #checando se a classe da temperatura de
fermentacao e diferente da classe numerica
       {
    stop("a temperatura de fermentacao precisa ser um valor numerico")
#parar a funcao e emitir mensagem de erro
       }
  else  #se a classe do CO2 e tfer estiverem corretas continua a funcao
       {
          CO2.residual=
matrix(c(1.713,1.646,1.527,1.527,1.473,1.424,1.377,1.331,1.282,1.237,1.194,1
.154,1.117,1.083,1.050,1.019,0.985,0.956,0.928,0.902,0.878,0.854,0.829,0.804
,0.781,0.759,0.738,0.718,0.699,0.682,0.655,0.592,0.530,0.479,0.436,0.359))
#matriz com o volume residual de CO2 que sobrou na cerveja apos o final da
fermentacao que depende da temperatura mais alta pela qual a fermentacao
passou porque toda vez que a temperatura e aumentada durante a fermentacao a
cerveja perde CO2. Por outro lado, baixar a temperatura apos a fermentacao
nao criara mais CO2.
          rownames(CO2.residual)=
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,
28,29,30,31,32,33,34,35)   #coloca como nome das linhas as temperaturas de
fermentacao
          colnames(CO2.residual)= "Vol de CO2"  #coloca como nome da coluna
Vol de CO2 que é o volume de CO2 dissolvido por volume de H20"
          carb.desej=co2  #a carbonatacao final desejada e o valor do
argumento volume de co2
          if(tfer<=35)    #se a temperatura de fermentacao for menor ou
igual a 35 graus
                {
                    carb.resi= CO2.residual[rownames(CO2.residual)==tfer,1]
#a carbonatacao residual e igual ao C02 residual dado de acordo com a
temperatura de fervura
                    carb.adic=carb.desej-carb.resi       #o quanto e
necessario de carbonatacao adicional e igual ao quanto e desejado menos o
residual calculado
                    acucar=carb.adic*4     #o acucar necessario para o
priming é igual a carbonatação adicional que precisa colocar vezes quatro,
porque cada 4,0g/l (quatro gramas por litro) de sacarose resulta em 01
```

```
volume de CO2
                }
          if(tfer>35)   #se a temperatura de fermentacao for acima de 35
graus o C02 residual e insignificante
                {
                    acucar= carb.desej*4    #multiplica direto a
carbonatacao desejada por 4, porque cada 4,0g/l (quatro gramas por litro) de
sacarose resulta em 01 volume de CO2
                }
       }
    acucar.total=acucar*volume    #quantidade de acucar a ser utilizada no
priming pro total de cerveja produzida e igual a quantidade de acucar
necessaria para dar o volume desejado de CO2 em um litro multiplicado pelo
volume total de litros que se tem
    agua.dil=acucar*3     #quantidade de agua necessaria pra diluir o acucar
total, antes de adicionar na cerveja, e igual a quantidade de acucar vezes 3
    resultado = data.frame(volume, garrafas, IBU, grau, acucar, agua.dil)
#o resultado e um dataframe com o volume total, o numero de garrafas a ser
utiluzada, o IBU, o grau alcoolico e a quantidade de acucar e agua
necessaria para a carbonatacao da cerveja
    colnames(resultado)= c("Volume Total(l)", "Garrafas", "IBU", "ABV(%)",
"Acucar do Priming(g)", "Agua do Priming(ml)")   #colocando o nome das
colunas no dataframe
    resultado      #produz no resultado o dataframe com o nome das colunas
    return(resultado)  #retorna o resultado no console
}
```

Arquivos da Função

Função breja() Help da função breja()