



Maria Gabriela Kiss Cornia

Olá galera! 😊

Sou mestranda da Unicamp. Trabalho com Ecologia Florestal.

Meus exercícios

Exercício 1 - Introdução ao R [ex1_introducao_kiss.r](#)

Exercício 4 - Análises exploratórias [ex4_analises.exploratorias_kiss.txt](#)

Exercício 5 - Criação de gráficos [ex5_graficos_kiss.r](#)

Exercício 7a - Regressões Lineares Simples [ex7.2_omodelomaissimples_kiss.txt](#)

Exercícios 7b - Modelos Lineares Múltiplos [ex7b_modelosmultiplos_kiss.r](#)

Exercício 8 - Reamostragem e simulação [ex8_reamostragensesimulacoes_kiss.r](#)

TRABALHO FINAL

- Propostas: [proposta_entregue_kiss.docx](#)

Comentários Lucas Teixeira

Oi, Kiss!

Posta suas propostas aqui na página do Wiki pra gente. Melhor que em arquivo Word. Qualquer dúvida, olha na página do [trabalho final](#) ;)

PROPOSTA A

FUNÇÃO: `label()`

Tarefa executada: geração de textos para servirem de etiquetas em formato PDF. Cada etiqueta agregará informações de planilhas e consultas a dados on line, equivalente a um mecanismo de mala-direta. Como exemplos de etiquetas a serem geradas são as utilizadas em coleções botânicas, zoológicas, geológicas e etc. (vide anexo 1).

Entrada: arquivo em forma de tabela o qual o R lerá como data frame podendo ser dados numéricos, categóricos, ou ambos na mesma coluna, linha ou célula.

Argumentos:

data: leitura dos dados com o separador entre campos que desejar e com título das colunas na primeira linha □ Default: `data = read.table (dados=dados, sep=sep, header=T,...)`

size.p: tamanho do papel - opções: `comprim x larg` em cm ou unidade de linhas, ou "A4", "A3"... Default: `size.p="A4"`

size.lab: tamanho de cada etiqueta - opções: `comprim x larg` em cm ou em unidades de linhas. Default: `size.lab = 10,10`.

n.lab: número de etiquetas por papel. Default: n.lab = 4.

name.col: nome ou posição das colunas dos dados que se deseja que seja transferido para etiqueta. Default: name.col = dados[0,].

ord: ordem das colunas que serão consideradas na confecção do texto da etiqueta. Default: ord= manter a mesma ordem presente nos dados. □ Default: ord = order(name.col).

sep: determinação dos separadores entre os valores de cada linha entre colunas ao concatenar na formação do texto. □ Default: sep="" (pode ser qualquer símbolo que desejar, quebras de linhas, ou linhas). conc.sep = rep(sep, n°colunas - 1), paste(ord, sep).

tit: definição do texto a ser escrito na primeira linha da etiqueta (título), normalmente o nome da instituição. Default: tit = "" □ title()

subtit: definição do texto a ser escrito na segunda linha da etiqueta (subtítulo), normalmente o nome ou número de referência dado ao material a ser etiquetado. Default: subtit = ""

adj: corpo do texto justificado, centralizado,... Default: disp = just.

adj.t: título justificado, centralizado,... Default: disp = center.

consult: Consulta em banco de dados on line para verificação da escrita correta ou atualização de nomes científicos. Opções T ou F. Default: consult = T.

font: estilo da fonte do texto. Default: font = "serif" □ mtext() ou par(font="")

size.f: tamanho da fonte do texto. Default: □ mtext() ou par(font="") (...) Para mais parâmetros ver argumentos da função "par", "text", "mtext" e "title".

Premissas e controle do fluxo da função:

Se tam.p for número muito elevado: parar a função e msg - "Tamanho do papel muito elevado. Verifique se o tamanho de papel informado está em cm."

Se tam.et > tam.p : parar a função e msg - "Tamanho de cada etiqueta maior do que o papel indicado no argumento tam.p".

Se tamanho o texto ficar maior que o tamanho da etiqueta: parar a função e msg - "Tamanho do texto excedeu o tamanho da etiqueta".

Retorno (output) da função: formato PDF do Rmarkdown, como modelo apresentado no anexo 1.

Comentários Lucas Teixeira

Oi, Kiss!

Acho que essa proposta, apesar de diferente, é relativamente pouco desafiadora, porque, se eu entendi bem, ela vai praticamente formatar em "mala direta" as informações que o usuário fornecer. Eu recomendaria você ficar com a proposta B.

PROPOSTA B

FUNÇÃO: count.plants()

Tarefa executada: contagem de indivíduos de diferentes espécies de plantas em diferentes estágios etários que cabem numa área sob determinado espaçamento entre indivíduos. Essa contagem será feita de acordo com o grau de dominância (% da área ocupada) que se deseja para cada espécie.

Entrada: planilha de dados contendo nas linhas as espécies de plantas desejadas e nas colunas as informações de faixa etária e espaçamento desejados entre indivíduos de cada espécie e faixa etária e os demais. O espaçamento consiste na área basal do indivíduo da espécie e faixa etária mais a área entre esse e os demais indivíduos vizinhos. O R lerá a planilha como data frame, as colunas espécie e faixa etária serão lidas como dados categóricos e a coluna espaçamento como numérico na unidade que o usuário desejar.

Argumentos:

data: planilha de dados.

area: área para a qual será feita a contagem. Deve estar na mesma unidade de área que a da coluna espaçamento. Default: area = .

spp.dom: vetor com as porcentagens de área que se deseja que cada espécie ou faixa etária ocupe. Colocar as porcentagens em ordem alfabética de espécies e depois em ordem alfabética de faixas etárias. Default: spp.dom = c()

Retorno (output) da função: retorna um data frame com quantos indivíduos caberiam na área por espécie em cada faixa etária.

Comentários Lucas Teixeira

Oi, Kiss!

Acho essa proposta melhor que a anterior, mas acho que você poderia deixar ela um pouco mais incrementada... O que você acha de colocar alguns controles de fluxo, como você apresentou na proposta anterior?

Estava aqui pensando... Será que teria como a gente fixar alguma espécie (fornecida pelo usuário) para que sua função dê preferência para esta espécie na hora de calcular quantas espécies caberão na área? Imaginando um cenário em que a pessoa tenha X mudas de alguma espécie e ela quer usar todas elas, para depois saber quantas das outras ela precisaria comprar para "preencher" o reflorestamento de uma área, pode ser bastante útil para que não "sobre" mudas de uma espécie. O que você acha?

Ah! A [função do Rodolfo](#) pode te dar algumas dicas também.

Qualquer coisa, tá aí!

Lucas

Oi gente, tudo bem?

Kiss, também gostei mais da proposta B, mas não entendi uma coisa: qual a utilidade dela? É para auxiliar no plantio de mudas seguindo um critério de dominância/área **OU** para o pesquisador coletar dados de contagem de plantas no campo e dar para a função calcular sua dominância e área ocupada?

Se for para plantio, concordo com as colocações do Lucas. Ah, e dar exemplos de como seriam os objetos de entrada e saída ajuda muito na nossa avaliação!

Abraços e boa sorte! — [Rodolfo](#) 2017/06/09 10:09

Anexo 1

HERBÁRIO DOM BENTO JOSÉ PICKEL SPSF - 49278

PASSIFLORACEAE *Passiflora mendoncae* Harms

Det.: L.C. Bernacci 12_01_2015

Brasil, Minas Gerais, Município de Delfim Moreira, Vila de São Francisco dos Campos do Jordão. Beira de estrada., Floresta Ombrófila Mista (WGS84, Zona 23K, 7501282S, 469942W, 1634m), 15_12_2014.

Trepadeira. Brácteas e pétalas rosas.

Col.: M.G.K. Cornia, 118. N.M. Ivanauskas, M.C. Assis e L.N. Gonçalves.

PROPOSTA ESCOLHIDA: A

FUNÇÃO LABEL

Help da função

label package:unknown R Documentation

Consulta dados on line e geracao de arquivos de exibicao grafica de textos em formato PDF. O layout do arquivo gerado e um modelo generico do aplicado em etiquetas de colecoes como as presentes em herbarios.

Description:

A partir de dataframe fornecido, a funcao possui a opcao de realizar uma consulta ao banco de dados on line do Flora do Brasil 2020 (<http://floradobrasil.jbrj.gov.br/>) e gera um arquivo PDF contendo as informacoes do data frame, conferidas ou não com o banco on line. O arquivo e gerado em um layout generico de etiqueta de herbario utilizado na confeccao de exsicatas. Ha opcoes de formatacao do tamanho e orientacao do arquivo para impressao, tamanho de cada etiqueta, tamanho das margens e conteudo do titulo e subtítulo.

Usage:

label (data, sep, consult, paper, size.lab, margin.in, margin.out, tit, subtit, sep.col, ...)

Arguments:

data: caracter com o nome do arquivo dos dados ou do objeto que contem os dados. Os dados devem ser uma planilha com titulo das colunas na primeira linha. As colunas se referem as informacoes que devem constar nas etiquetas sobre cada material. Os materiais sao as

exsicatas, expressas nas linhas. Para cada linha sera gerada um arquivo de etiquetas. O titulo das colunas devem ser, nao necessariamente nessa ordem: herbarium_code, family, genus, epithet, author, determiner, determined, country, state, city, locality, note, datum, zone, UTMx, UTM, altitude, date, observation, main_collector, collector_number, other_collectors. As informacoes que se referem cada coluna: herbarium_code, codigo do herbario que identifica cada material (exsicata); family, nivel taxonomico de familia; genus, nivel taxonomico de genero; epithet, epiteto especifico; author, autor da especie; determiner, quem identificou a especie do material; determined, data em que ocorreu a determinacao; country, pais onde coletou o material; state, estado onde coletou o material; city, municipio onde coletou o material; locality, alguma descricao da localidade onde coletou o material; note, alguma observacao sobre o local de coleta; datum, datum; zone, zona geografica; UTMx; UTM; altitude, altitude do local de coleta em metros; date, data da coleta; observation, descricao do material; main_collector, principal coletor; collector_number, numero que o coletor atribui ao material; other_collectors, demais pessoas que acompanharam a coleta.

sep: caracter utilizado como separador entre dados de colunas diferentes na planilha de dados que deve ser informado pelo usuario.

consult: teste lógico (informar TRUE ou FALSE). Se TRUE a funcao consulta as informacoes das colunas denominadas "genus" e "epithet" no banco de dados on line Flora do Brasil 2020 (<http://floradobrasil.jbrj.gov.br/>). Se FALSE a funcao não executa a consulta e gera a etiqueta da forma como os dados foram informados em data. Default: "TRUE".

paper: caracter que define o tamanho do arquivo desejado. Por exemplo: "a4", "letter", ... Default: "a4r".

size.lab: vetor numerico que determina tamanho da etiqueta em polegadas. As ordem das posicoes no vetor referem-se as seguintes dimensoes da etiqueta c(width, height). Default: c(5,3).

margin.in: vetor numerico que determina a distancia em numero de linhas entre o conteudo de cada etiqueta e a borda da etiqueta. A ordem das posicoes no vetor corresponde as distancias nas seguintes regioes da etiqueta: c(bottom, left, top, right). Default: c(0.5,0.5,0.5,0.5).

margin.out: vetor numerico que determina a distancia em numero de linhas entre a borda do dispositivo grafico, esse que correspondera a borda do papel, e a borda da etiqueta. A ordem das posicoes no vetor corresponde as distancias nas seguintes regioes da etiqueta: c(bottom, left, top, right). Default: c(0.5,0.5,0.5,0.5).

tit: caracter informado pelo usuario que define o titulo localizado na primeira linha da etiqueta, normalmente o nome do herbario.

subtit: caracter informado pelo usuario que define o subtítulo localizado na segunda linha da etiqueta, normalmente a sigla do herbario e o numero de referencia dado ao material a ser etiquetado.

sep.col: caracter informado pelo usuario que define os separadores entre as informacoes expressas por cada coluna em cada linha (material) ao concatenar na formacao do texto da etiqueta. Recomenda-se adicionar o um caracter de espaço (" ") e algum outra caracter como separador.

Details:

Para a execucao da funcao e necessario ter instalado e carregado o pacote "flora".

A consulta aos dados on line pelo argumento "consult = TRUE" é realizada pela execucao da funcao "get.taxa" do pacote "flora" de autoria de Carvalho, G. (2017). Para a execucao da consulta é preciso ter conexao com a Internet. A consulta realiza a verificacao da escrita dos nomes cientificos dos dados presentes nas colunas "genus" e "epithet", além de obter o dado da coluna "author" e corrigir para o nome mais atual da espécie quando for necessário.

Para verificar as opcoes dos tipos de papel e orientacoes ver opcoes do argumento "paper" da funcao pdf{grDevices}.

Nos valores default para paper, margin.in, margin.out e size.lab, recomenda-se o uso maximo de 68 caracteres para os dados presentes nas colunas "observation" e para as colunas "main_collector", "collector_number" e "other_collectors" contabilizados juntos. Recomenda-se, tambem, uso maximo de 136 caracteres para as colunas "country", "state", "city", "locality", "note", "datum", "zone", "UTMx", "UTMy", "altitude" e "date" contabilizados juntos.

Value:

A funcao retorna um arquivo pdf contendo 4 etiquetas para cada material (linha) da planilha de dados no directorio a que o R esta vinculado.

Warning:

A funcao retorna um aviso quando algum dado das colunas "genus" e "epithet" não foram encontrados na consulta realizada ao Flora do Brasil 2020, apontando alguma razao para isso que e o fato do dado não existir no banco de dados on line por se tratar de uma especie exotica.

Author(s):

Maria Gabriela Kiss Cornia maria.cornia@usp.br

References:

Carvalho, G. 2017. Package "flora". Disponível em:
<<https://cran.r-project.org/web/packages/flora/flora.pdf>>. Acesso em: 20 jun 2017.

See Also:

Pacote: "taxize". Funcoes: plot(), pdf(), text(), mtext(), paste().

Examples:

#Exemplo 1

#Criando vetores para compor um exemplo de dataframe de 3 linhas que correspondem a 3 materiais para os quais deseja-se produzir etiquetas.

```
herbarium_code ← c("123", "456", "789") family ← c("Myrtaceae", "Lauraceae",  
"Melastomataceae") genus ← c("Myrceugenia", "Ocotea", "Miconia") epithet ← c("miersiana",  
"pulchella", "cinerascens") author ← c("(Gardner)Legrand et Kausel", "(Nees et Mart.) Mez",  
"Miq.") determiner ← c("Fulano", "Ciclano", "Beltrano") determined ← c("01_02_03", "04_05_06",
```

```

"07_08_09") country ← c("Brasil","Brasil","Brasil") state ← c("SP", "MG", "RJ") city ← c("São
Paulo", "Belo Horizonte", "Rio de Janeiro") locality ← c("aaaaaaa", "bbbbbbb", "ccccccc") note ←
c("ddddddddd", "fffffffff", "ggggggg") datum ← c("WGS84", "WGS84", "WGS84") zone ← c("23K",
"23K", "23K") UTMx ← c("1111111", "2222222", "3333333") UTMx ← c("4444444",
"5555555", "6666666") altitude ← c("777", "888", "999") date ← c("01_02_03", "04_05_06",
"07_08_09") observation ← c("xxxxxxxxxxxxxxxxxxxxxxxx", "yyyyyyyyyyyyyyyy",
"zzzzzzzzzzzzzz") main_collector ← c("Fulano", "Fulano", "Fulano") collector_number ←
c("100", "101", "102") other_collectors ← c("Ciclano, Beltrano", "Ciclano, Beltrano", "Ciclano,
Beltrano")

#criando o dataframe de exemplo dados ← data.frame(herbarium_code, family, genus, epithet,
author, determiner, determined, country, state, city, locality, note, datum, zone, UTMx, UTMx,
altitude, date, observation, main_collector, collector_number, other_collectors)

#executando a funcao label(dados, sep="\t", tit="HERBÁRIO X", subtit="SIGLA ", sep.col= "; ")

#Exemplo 2

#lendo planilha de dados como dataframe e salvando como objeto no R dados ←
read.table("dados.txt", sep="\t", header=TRUE)

#executando a funcao label(dados, sep="\t", consult=TRUE, paper="a4r", size.lab = c(5,3),
margin.in = c(0.5,0.5,0.5,0.5), margin.out = c(0.5,0.5,0.5,0.5), tit="HERBÁRIO X", subtit="SIGLA
", sep.col= "; ")

```

Script da função

```

#funcao com seus argumentos e respectivos valores default label ← function(data, sep="",
consult=TRUE, paper="a4r", size.lab = c(5,3), margin.in = c(0.5,0.5,0.5,0.5), margin.out =
c(0.5,0.5,0.5,0.5), tit="", subtit="", sep.col= "", ... )

{

#leitura dos dados dentro da funcao, caso o usuario nao tenha lido os
dados com funcao read.table ainda
data = read.table(data, header=TRUE, sep=sep, stringsAsFactors = FALSE)

#controle de fluxo - caso o usuario tenha escolhido TRUE para realizar a
consulta ao Flora do Brasil 2020
if(consult==TRUE)
{
  #concatenacao dos dados das colunas "genus" e "epithet" para permitir
a consulta
  data$species_originals <- paste(data$genus, data$epithet, sep=" ")
  tmp <- data
  tmp$species_originals <- gsub("cf\\.", " ", tmp$species_originals)
#retira a abreviacao "cf" geralmente utilizados para permitir consulta
  tmp$species_originals <- gsub("aff\\.", " ", tmp$species_originals)
#retira a abreviacao "aff" geralmente utilizados para permitir consulta

```

```
tmp$species_originals <- gsub("_", " ", tmp$species_originals) #retira
 "_" geralmente utilizados para permitir consulta
nomes <- get.taxa(tmp$species_originals) #funcao do pacote "flora" que
realiza a consulta on line
nomes.join <- data.frame(tmp[setdiff(names(tmp), names(nomes))],
nomes) #faz a comparacao dos nomes da planilha de dados fornecida com os
nomes corretos obtidos pela consulta pela
data$scientific.name <- nomes.join$scientific.name # guardando no
dataframe que contém os dados originais a coluna dos dados corrigidos. A
coluna "scientific.name" e criada pela funcao get.taxa
#criando mensagem de alerta para avisar o usuario caso haja algum nome
de especie que nao foi encontrada no banco de dados on line e aponta uma
possivel razao que consiste no fato da especie ser exotica (nao nativa
do Brasil).
warning("A(s) espécie(s) ", paste(data$species_originals[data$notes ==
"not found"], sep=","), " não foi/foram encontrada(s) na consulta aos
dados on line do Re flora. Verificar possibilidade da(s) espécie(s)
ser/serem exótica(s).", call. = FALSE)
}

#controle de fluxo - caso o usuario tenha escolhido FALSE para realizar
a consulta ao Flora do Brasil 2020
else
{
#realiza a concatenacao com os proprios dados fornecidos, para ficar do
mesmo jeito que e feito pela opcao TRUE de consult
data$scientific.name <- paste(data$genus, data$epithet, data$author,
sep=" ")
}

#executa a funcao para cada linha do dataframe dos dados
for(i in c(1:length(data$herbarium_code)))
{
#gera o arquivo PDF
prod <- pdf(file="labels%0id.pdf", paper=paper, onefile = TRUE, height=
8, width=12)

#formata o output grafico da etiqueta
par(mfrow=c(2,2), mar=margin.in, oma=margin.out, pin=size.lab, cex=1,
family="serif", lwd=0.5)

#funcao de alto nivel para permitir a formacao das novas janelas
graficas e o argumento frame.plot permite a formacao do contorno para a
etiqueta (area de plotagem)
plot(x=NULL, Y=NULL, xlim=c(1,8), ylim=c(1,20), ylab="", xlab="", axes=
FALSE, frame.plot= TRUE)
#comandos que adicionam os textos da etiqueta
mtext(paste(tit, "\n" , subtit, " ", data$herbarium_code[i]), line=-3,
cex= 1.3, font= 2)
mtext(data$family[i], line= -5)
```

```

mtext(paste(data$scientific.name[i]), line= -6, font= 3)
mtext(paste("Det: ", data$determiner[i], ", ", data$determined[i]),
line= -8)
mtext(paste(data$country[i], data$state[i], data$city[i],
data$locality[i], data$note[i],"\n", data$datum[i], data$zone[i],
data$UTMx[i], data$UTMy[i], data$altitude[i], data$data[i],".", sep=
sep.col), line= -10)
mtext(data$observation[i], line= -12)
mtext(paste(data$main_collector[i], data$collector_number[i],
data$other_collectors[i], ".", sep= sep.col), line= -14)

#os comandos se repetem para cada uma das 3 etiquetas da pagina geradas
pelo par(mfrow(2,2))
plot(x=NULL, Y=NULL, xlim=c(1,8), ylim=c(1,20), ylab="", xlab="", axes=
FALSE, frame.plot= TRUE)
mtext(paste(tit, "\n" , subtit, " ", data$herbarium_code[i]), line=-3,
cex= 1.3, font= 2)
mtext(data$family[i], line= -5)
mtext(paste(data$scientific.name[i]), line= -6, font= 3)
mtext(paste("Det: ", data$determiner[i], ", ", data$determined[i]),
line= -8)
mtext(paste(data$country[i], data$state[i], data$city[i],
data$locality[i], data$note[i],"\n", data$datum[i], data$zone[i],
data$UTMx[i], data$UTMy[i], data$altitude[i], data$data[i],".", sep= (";
")), line= -10)
mtext(data$observation[i], line= -12)
mtext(paste(data$main_collector[i], data$collector_number[i],
data$other_collectors[i], ".", sep=", "), line= -14)

#os comandos se repetem para cada uma das 3 etiquetas da pagina geradas
pelo par(mfrow(2,2))
plot(x=NULL, Y=NULL, xlim=c(1,8), ylim=c(1,20), ylab="", xlab="", axes=
FALSE, frame.plot= TRUE)
mtext(paste(tit, "\n" , subtit, " ", data$herbarium_code[i]), line=-3,
cex= 1.3, font= 2)
mtext(data$family[i], line= -5)
mtext(paste(data$scientific.name[i]), line= -6, font= 3)
mtext(paste("Det: ", data$determiner[i], ", ", data$determined[i]),
line= -8)
mtext(paste(data$country[i], data$state[i], data$city[i],
data$locality[i], data$note[i],"\n", data$datum[i], data$zone[i],
data$UTMx[i], data$UTMy[i], data$altitude[i], data$data[i],".", sep= (";
")), line= -10)
mtext(data$observation[i], line= -12)
mtext(paste(data$main_collector[i], data$collector_number[i],
data$other_collectors[i], ".", sep=", "), line= -14)

#os comandos se repetem para cada uma das 3 etiquetas da pagina geradas
pelo par(mfrow(2,2))
plot(x=NULL, Y=NULL, xlim=c(1,8), ylim=c(1,20), ylab="", xlab="", axes=
FALSE, frame.plot= TRUE)

```

```
mtext(paste(tit, "\n" , subtit, " ", data$herbarium_code[i]), line=-3,
      cex= 1.3, font= 2)
mtext(data$family[i], line= -5)
mtext(paste(data$scientific.name[i]), line= -6, font= 3)
mtext(paste("Det: ", data$determiner[i], " ", data$determined[i]),
      line= -8)
mtext(paste(data$country[i], data$state[i], data$city[i],
      data$locality[i], data$note[i], "\n", data$datum[i], data$zone[i],
      data$UTMx[i], data$UTMy[i], data$altitude[i], data$data[i], ".", sep= (";
      ")), line= -10)
mtext(data$observation[i], line= -12)
mtext(paste(data$main_collector[i], data$collector_number[i],
      data$other_collectors[i], ".", sep=", "), line= -14)

#encerra a edicao do grafico(plot), salva no diretorio e permite abrir o
arquivo
dev.off()

#determina o output da funcao
return(prod)

}

}
```

ARQUIVOS DA FUNÇÃO

- Script da função: [funcao.r](#)
- Help da função: [help_label.txt](#)
- Dados para executar exemplo 2 da função: [dados.txt](#)

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:maria.cornia:start

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