APRESENTAÇÃO

Doutorando em Zoologia, Instituto de Biociências/USP Aluno do Laboratório de Ecologia Sensorial e Comportamento de Artrópodes (LESCA) Orientado por: Prof. Dr. Rodrigo Hirata Willemart

Trabalho com aprendizado em opiliões

MEUS EXERCÍCIOS RESOLVIDOS

1.r

4.r

5.r

7-2.r

7b2.r

8.r

9-2.r

PROPOSTA DE TRABALHO FINAL

Proposta1

Esforço amostral

Função: esfor.amos ()

Contextualização

Um problema cotidiano de pessoas que precisam planejar viagens de campo para coletar organismos vivos é otimizar o tempo de coleta. Nesse sentido, uma função que possa automatizar esse planejamento pode ser bastante funcional. Além do número de pessoas coletando, outras variáveis são importantes para se planejar como por exemplo o local que se pretende coletar e o período do dia, bem como a quantidade de organismos a ser coletado. Essa função pretende de maneira simples buscar em uma base de dados pré-estabelecida, fornece um output de nº de pessoas sugeridas para um determinado tempo (em horas) que se gastará na tarefa. Para gerar essa base de dados, vou consultar registros de coletas de opiliões, anotando o tempo de coleta/ quantidade de indivíduos coletados de 3 localidades bastante corriqueiras na coleta de opiliões. Em seguida (utilizando o próprio R :D) vou plotar esses dados em gráficos e comparar a distribuição com distribuições já conhecidas (normal, binomial, poisson). Após essa análise vou estabelecer para cada uma das localidades analisadas as distribuições que mais se assemelham as distribuições dos dados do registro utilizando no entanto os valores de media e desvio padrão dos originais. Assim, cada localidade terá um distribuição conhecida no qual será baseado o resultado dessa função. A função será capaz de fornecer uma estimativa do número de horas que será dispendido na coleta de acordo com o número de pessoas.

Planejamento da função

Entrada: esfor.amos (N, loc, per)

- N= quantidade de indivíduos pretendidos na coleta (classe: int, N>0)
- Loc = Local de coleta. Locais antropomorfizados (city), remanescentes de mata (mata), áreas de preservação ambiental (pres)
- Per = período do dia que será feita a coleta. Dia (“D”), noite (“N”) ou ambos (“D/N”)

Verificando parâmetros:

- N deve ser numérico e maior que 0. Se não, volta uma mensagem de erro “ N precisa ser numérico e > 0.”
- Loc deve ser definido. Se não, volta mensagem de erro “defina o local de coleta: “city”, “mata”, “pres”
- Per deve ser um caractere D, N, D/N. Se não, função continua trabalhando, mas devolve uma mensagem de alerta” per não discriminado, calculando para dia”

Pseudo-código

1. Cria um objeto N;

2. De acordo com o o objeto inserido no argumento loc, condiciona a três possíveis distribuições(D, N ou D/N) probalísticas que serão definidas no próximo argumento;

3. De acordo com o objeto inserido no argumento per, direciona a função para as distribuições já pré-estabelecidas pelo argumento anterior;

4. Cria o data.frame;

5. Salva em um documento PDF.

Saída:

Um data frame mostrando uma perspectiva de horas gastas (coluna 1) de acordo com a quantidade de pessoas em campo (coluna 2).

Proposta 2

Desempenho dos estudantes Função: desemp()

Independente da disciplina ministrada a maioria dos professores utiliza métodos de avaliação que envolvem notas ou conceitos. Por mais diferente que sejam esses métodos, geralmente existe uma média final que determina se o aluno atingiu ao não o parâmetro esperado. Fica como papel da coordenação de muitas escolas a conversa com pais e responsáveis ao final de cada bimestre letivo e também ao final do ano, para apresentar o rendimento escolar dos estudantes. Pensando nisso, muitas vezes pode ser extremamente trabalhoso para um coordenador, ficar buscando notas de alunos, em cada planilha, de cada disciplina, dentro dos diferentes bimestres. Se não bastasse, muitas vezes para facilitar a explicação do rendimento dos alunos para os pais cabe ao coordenador gerar gráficos comparativos ou mostrar tabelas gigantescas de difícil interpretação. Nesse sentido, a presente função seria uma ferramenta funcional no intuito de facilitar essa tarefa do coordenador. A função retornaria um boletim com as médias do aluno por disciplina e também dois gráficos: 1-um resumo da sua evolução ao longo dos bimestres em cada disciplina e 2- um histograma que compara as médias do aluno em relação as médias da sala. (isso poderia ser pensado por exemplo para sub

áreas como: media nas disciplinas de humanas, ou exatas...)

Planejamento da função

Entrada: desemp (x dados, aluno)

- dados = deve existir um data.frame para cada disciplina. Esses devem ser data.frames com 1ª coluna destinada a nome dos alunos (class: char) e as seguintes com as notas de cada bimestre 1-4(class: numeric, com 1 casa decimal)
- aluno = nome do aluno que se deseja obter o boletim e os gráficos (classe: Char)

Verificando os parâmetros

- dados deve ser um data frame. Se não, devolve a mensagem de erro “dados não é um data frame”
- aluno deve ser composto apenas por caracteres. Se não, devolve a mensagem de erro “nomes não pode conter dados numéricos”

Pseudo-código:

(ainda me parece muito difícil de escrever, preciso de ajuda :D)

Saída:

Data frame (título =nome do aluno, 1º coluna serão as disciplinas e nas colunas seguintes os bimestres de 1-4 e uma 5ª coluna média final) com o boletim e uma mensagem da situação do aluno (aprovado, em recuperação, reprovado) esse documento será automaticamente salvo em pdf. Além disso, serão disponibilizados dois gráficos resumindo essa situação: 1º um histograma que mostra a nota do aluno em relação a nota dos demais alunos da sala; 2º um gráfico stripchart que mostra a média final do aluno por bimestre em cada uma das disciplinas.

Comentários Melina

Olá Guilherme, as ideias parecem interessantes, mas suas propostas ainda estão confusas. O principal que me ajudaria a entende-las é o pseudocódigo que não foi feito na proposta B e está mal feito na proposta A. O pseudocódigo é basicamente a descrição verbal do passo-a-passo da sua função, se você não conseguiu escreve-lo para proposta B é porque você ainda não tem clareza do que sua função vai fazer e como vai fazer, né? E o pseudocódigo da proposta A deveria estar detalhado o suficiente para indicar cada passo.

PROPOSTA A

Eu não entendi como você vai fazer para usar o banco de dados existente e gerar uma distribuição para cada combinação de parâmetros. Eu não entendi que distribuição é essa.

Pelo que entendi, você vai usar um banco de dados já conhecido de tempo de coleta, esforço e número de indivíduos para cada combinação dos seus parâmetros local (city, mata, pres) e período do dia (dia, noite, dia-noite). Mas o que me parece é que este dados vão te fornecer o número de indivíduos médio que se consegue coletar por esforço (horas-homem) em cada combinação. Por exemplo, voce tem dados de 20 coletas em city durante o dia, com o tempo gasto e o número de

pessoas. basta você padronizar os dados para X indivíduos/hora/pessoa, certo? Daí, se você quer saber o esforço de coletar 100 indivíduos em city durante o dia, vc só precisaria fazer uma regra de 3, se eu coleto X indivíduos em 1 hora com 1 pessoa quantas horas preciso para coletar 100 indivíduos com 1 pessoa? Se a relação for direta, com duas pessoas vc precisa dividir esse tempo por 2. É mais ou menos isso que você quer fazer?

Se sua proposta for mais ou menos essa que eu descrevi, ela está muito simples. porque basta você ter dentro da função os valores de média e desvio padrão do número de indivíduos/hora/pessoa para cada conjunto de parâmetros que a função vai estar pronta.

Acho que se quiser ficar com essa proposta, seria bom pensar melhor em coisas que ela possa fazer de mais desafiador. Ou eu que não entendi bem o que ela faz, e daí preciso que você me ajude e explique melhor.

PROPOSTA B

Pelo que entendi, essa função é bem simples e pode ser feita com uma simples indexação das planilhas de entrada buscando a linha que corresponde ao aluno, separando ela e juntando as colunas para gerar os dados de notas, certo? Depois você faz gráficos para apresentar melhor os resultados. Isso poderia ser feito com poucas linhas de código.

Talvez você devesse pensar em problemas nestas planilhas de disciplinas que possam ser resolvidos ou pelo menos detectados em sua função. Por exemplo, para trazer flexibilidade à função o número de planilhas de disciplinas não deve ser fixo, ou seja, para a turma do 6o ano são 7 disciplinas, mas para o 9o ano são 10, e a função precisa ser capaz de lidar com esse número variável de data.frames que ela vai receber.

Outro problema pode ser nome digitado errado em alguma das planilhas de disciplinas. A sua função vai procurar o nome de “fulaninho” em todas as planilhas, mas em uma vai estar escrito “flaninho”, será que sua função retornará uma mensagem de erro por não encontrar fulaninho em uma das planilhas ou ela é capaz de procurar algo que se pareça com fulaninho assumir que é o mesmo aluno e trazer uma mensagem de aviso sobre esse problema?

E se não houver nota, ou se a nota estiver em uma classe diferente em planilhas diferentes?

Enfim, achei que essa função mesmo sendo simples, pode ter incorporada soluções ou avisos de problemas encontrados nas planilhas das notas, para que ela seja mais bem elaborada e se torne algo mais útil do que fazer um script de manipulação de dados.

Por fim, acho que a função mais desafiadora pra você seria a B, tendo incorporado algumas das sugestões que fiz, ou outras coisas que você veja que podem ser interessantes para tornar a função mais útil para o coordenador. Mas para isso, seria muito importante que você pudesse escrever o pseudocódigo dela!!

Aguardarei nova versão Ok? Qualquer coisa, só me escrever melina.leite@ib.usp.br.

Bom trabalho!

— *Alexandre Adalardo de Oliveira 2018/05/11 09:18*

Olá Guilherme,

Não vou falar muito sobre a proposta 2. Por não ter conseguido fazer o pseudo-código, meu entendimento, é que ela se torna inviável. De qualquer forma, a Melina aponta como sendo promissora se houver mudanças e vc. conseguir imaginar o pseudo-código dela.

A proposta 1 precisa ser melhor definida. Muitos pontos estão obscuros e fica dificil entender como irá fazer. O que é esse registro de coletas de opiliões? Quais as variáveis que estão nesse registro? Imagino que tenha o horário da coleta, número de pessoas e tipo de ambiente (não me parece que seja localidade o que define). Então, se houver esses dados tabulados de maneira correta é possível fazer um modelo para dele tirar previsões. Ou como a Melina coloca, ao menos a predição de opiliões coletados em média em cada uma das situações em homens/horas de trabalho. Não é uma questão de “distribuições conhecidas” e sim de ter um modelo adequado para fazer predições. Entretanto, depois de ter o modelo ou as médias extraídas da base de dados, calcular o predito é muito simples. Uma complicação é trocar o período de coleta pelo horário de inicio e final do trabalho de campo, mas mesmo assim, não vejo um desafio interessante de programação...

Pedimos então, que repense e ajuste as duas propostas, para que possamos fechar o quanto antes o que será sua função final. No momento que tiver isso pronto, mande uma ms para a Melina para que possamos avaliar.

RESPOSTA AOS COMENTÁRIOS:

Oi Melina e Alê,

obrigado pelos comentários. Para ser sincero eu só consegui ter clareza do que eu poderia fazer conforme fui aprendendo novas funcionalidades do R. Por isso precisei ir montando a função para entender como eu poderia prosseguir no passo a passo. Decidi por seguir a sugestão da Melina e desenvolvi a proposta B acrescentando as funcionalidades apontadas por ela e outras que pensei que poderiam ser úteis para o usuário.

Acredito que agora a função está bem flexível, por exemplo: 1- recebe quantas planilhas forem necessárias; 2- recebe valores de médias diferentes, o que pode ser utilizado para disciplinas que usam diferentes médias de aprovação ou mesmo diferentes escolas; 3- pode rodar para uma turma toda de alunos ou para alunos específicos; 4- gera documentos em pdf por aluno que podem auxiliar no entendimento do rendimento escolar do aluno e também da turma a qual ele pertence). Enfim, o script está comentado linha a linha com maiores detalhes das funcionalidades da função.

Não posso negar que foi uma experiência desafiadora. Ainda me sinto Pink, mas sem dúvida com autonomia para continuar aprendendo visando uma transformação para Cérebro, rs.

Não sei se esse é o local e o momento apropriado, mas não posso deixar de registrar publicamente meu profundo agradecimento a monitora Rena e a companheira de turma Marianne Azevedo que me ajudaram muuuito, inclusive psicologicamente, quando estava prestes a desistir :D

CÓDIGO DA FUNÇÃO - NOTAS

```
#Pacotes requeridos:

install.packages("dplyr")
install.packages("memisc")
install.packages("grid")
install.packages("gridExtra")

# código da função:

notas <- function(media, aluno="todos") {  # cria a funcao notas com os
argumentos "media" e "aluno" cujo padrao e "todos"
  #verificando os parametros comuns
  if(missing(media)) stop("insira a media utilizada pela escola para
criterio de aprovacao") # Se nao existir dados para o argumento media,
interrompe a funcao e exibe a mensagem entre aspas
  if (media < 0 | media > 10 ) stop("media deve ser um valor entre 0 e 10")#
se o valor de media nao estiver entre o intervalo 0 e 10, interrompe a
função e exibe a mensagem entre aspas
  if(class(media)!="numeric")stop ("media precisa ser da classe numérica") #
se os dados utilizados no argumento media não pertencerem a classe numeric,
interrompe a funcao e exibe a mensagem entre aspas
  if(class(aluno) != "character") stop("aluno precisa ser da classe
character") # se as informacoes utilizadas no argumento aluno nao
pertencerem a classe character, interrompe a funcao e exibe a mensagem entre
aspas
  #Abrindo pacotes necessario na funcao
  library("grid") # torna disponivel para uso o pacote "grid"
  library("gridExtra")# torna disponivel para uso o pacote "gridExtra"
  library("memisc")# torna disponivel para uso o pacote "memisc"
  library("dplyr")# torna disponivel para uso o pacote "dplyr"
#### importando tabelas das disciplinas: todas precisam estar em um mesmo
diretorio e o diretorio precisa estar indicado como diretorio de trabalho
  aa <- list.files(path=getwd(), full.names=TRUE, pattern = ".csv")#
listando todos os arquivos tipo tabela com extensao .csv
  tabelas <- lapply(aa, read.table, header=TRUE, sep=";") #lendo todas as
tabelas do diretorio
  tabelas <- as.array(tabelas) #transformando em array para facilitar as
```

```
operações
    for (i in 1:length(tabelas)){ # criando um contador de 1 ate o numero de
disciplinas que eu tenho
    nome <- paste0("Disciplina", i) # criando objeto nome que e formado por
um unico item (char) composto por"Disciplina" e o numero da posicao do
contador
    assign(nome, as.data.frame(tabelas[i])) # usa o objeto nome e aplica a
funcao as.data.frame nos dados "tabelas" na posicao [i]
  }
  notas <- to.data.frame(tabelas)# cria o objeto notas que é a transformação
do objeto tabelas em um data.frame
  notas$media <-apply(notas[,4:6],MARGIN =1 , FUN=mean)# cria uma nova
coluna que traz o resultado da media das notas ao longo dos bimestres
  realizados <-data.frame(c(length(Disciplina1$nome),
length(tabelas)),row.names = c("numero de alunos analisados", "numero de
disciplinas compuatadas"))# cria o objeto realizados que será utilizado como
return da funcao, exibindo a quantidade de alunos e disciplinas computados
pela funcao
  notasgeral <- as.data.frame(notas[,-2]) # cria o objeto "notasgeral"
tranformando o objeto notas sem a coluna das disciplinas em data.frame
  if(aluno == "todos") #se argumento aluno for igual a "todos"
  {
    b <- 1:length(unique(notas[,3])) # faz o for para cada aluno
  }
  else                                # caso contrário (ou seja, se alguns
alunos forem selecionados)
  {
    if(sum(aluno == levels(notasgeral$nome)) == 0) #primeiro testa se os
nomes colocados estão entre alunos da turma
    {
      stop("aluno(s) não econtrado(s). Rever nome(s) incluído(s) no
argumento aluno") # Senao, interrompe a funcao e exibe a mensagem entre
aspas
    }
    else #se o nome colocado no argumento for correspondente aos nomes dos
alunos, segue
    {
      b <- grep(paste(aluno,collapse="|"),levels(notasgeral$nome))
#seleciona níveis de notasgeral$nome correspondentes aos alunos escolhidos
para gerar boletins só pra esses alunos dentro do for
    }
  }
  for(a in b){ #cria um loop de "a" em "b"
    pdf(file = paste0("aluno",a,".pdf"))# estabele acriação de arquivos .pdf
para cada aluno dentro do contador "a"
    par(mfrow=c(1,1)) # estabele os parametros graficos de 1 unico graifico
por tela
    hist(notasgeral$media,col = "gray",xlab = "media final",ylab= "N de
alunos",breaks= 10, main= "Desempenho geral dos alunos", freq = FALSE, xlim
= c(0,10))# gera um histograma que apresent a frequencia em que foram
obtidas determinadas notas pelos alunos.
```

```
    abline(v=(media), col="red", lwd=5)#Acrescenta uma linha vertical ao
histograma de acordo com o valor inserido argumento "media"
    aluno <-filter(notas,nome==unique(notas[,3])[a])#  gera um nome objeto
apenas com as informacoes referentes ao nome do aluno
    notasaluno <-as.matrix(aluno[,4:8])#cria o objeto notasaluno que
trasnforma em matrix o objeto aluno mantando apenas as colunas numericas par
permitir a operação seguinte
    dotchart(notasaluno,labels = row.names(notasaluno), main =
unique(notas[,3])[a], xlab ="notas", color=
c("red","black","darkgreen","purple","darkorange","gray"),cex = 0.7, pch=2,
xlim =range(0:10))# cria um dotchart da matrix "notasalunos" que apresenta a
evolução do aluno em cada disciplina para todos os bimestres
    abline(v=(media), col="red", lwd=3, lty=3)# Acrescenta uma linha ao
dotchart de acordo com o valor inserido no argumento "media"
    notasaluno2 <- as.data.frame(aluno)# cria o objeto "notasaluno2" que
retorna o objeto aluno para o formato data.frame facilitando a próxima
operação
    legend("bottomleft",legend =notasaluno2$disciplina, cex=0.6,bty="L",
fill= c("red","black","darkgreen","purple","darkorange","gray"))# Acrescenta
uma legenda ao dotchart de acordo com os nomes das disciplinas do objeto
"notasaluno2"
    grid.newpage()# Permite a criação de uma nova página para a plotagem da
tabela a seguir
    grid.table(aluno[,-3])# plota uma tabela do objeto aluno excluíndo o
nome do mesmo, uma vez que esse já estará no documento
    dev.off() ## fecha o dispositivo pdf
  }
    return(realizados) #retorna o objeto "realizados" que informa ao usuário
quantas disciplinas e quantos alunos foram utilizados para realizar a função
}
```

HELP DA FUNÇÃO - NOTAS

```
notas                 package:unknown                 R Documentation


  Análise de desempenho escolar


Description:


  Função utilizada para verificar desempenho escolar de turmas e/ou
desempenho individual de alunos a partir de
  planilhas de notas de diferentes disciplinas. A função gera documentos em
.pdf de acordo com a quantidade de
  aluno que se deseja verificar o desempenho. Esse documento possuí gráficos
```

```
(histograma e dotchart) e uma tabela
  que auxiliam na análise dos resultados escolares ao longo dos bimestres.
em cada disciplina inserida na função.

Usage:

  notas(media, aluno)

Arguments:

  media: Número pertencente ao intervalo (0-10) que adiciona uma linha
vertical nos gráficos gerados

  aluno: Um caráter que deve corresponder aos nomes utilizados nas planilhas
de entrada. Argumento tem por padrão
  “todos” que utiliza todos os nomes de alunos de acordo com as planilhas de
entrada.

Value:

  Retorna um data.frame e (n)documentos em pdf:

  Data frame:

  Linha 1-mostra a quantidade de alunos que foi utilizado para os cálculos
de comparação de notas (“desempenho dos
  alunos”)

  Linha 2- mostra a quantidade de disciplinas que foram utilizadas para
realizar os calculos

  Documento pdf:
  Será salvo no diretório de trabalho do usuário

Warning:
  Para que a função rode perfeitamente é recomendado que as planilhas de
entrada tenham comprimentos de colunas
  iguais. O modelo ideal têm 6 colunas, sendo a primeira com o nome da
disciplina, a segunda com nome dos alunos e
  as 4 seguintes com notas de cada bimestre.
  As planilhas devem estar na pasta que está selecionada como diretório de
trabalho atual da sessão do R
  Para quantidades grandes de planilhas ou planilhas com muitos nomes de
alunos a função pode requerer alguns
  minutos até ser finalizada.

Details

  Os arquivos que servirão de entrada devem estar no diretório de trabalho.
Os arquivos devem estar na extensão
  .CSV e utilizar o espaçador ”;”.
```

```
  Os arquivos resultantes da operação da função serão salvos no formato .pdf
na mesma pasta que estavam as
  planilhas de entrada. Serão nomeadas de acordo com o número do aluno na
planilha inicial, que deve corresponder
  ao número de chamada.

  A primeira página contendo um histograma de desempenho geral da turma em
relação à média escolar.

  A segunda página, o desempenho do aluno específico com os resultados
escolares de todas as disciplinas analisadas
  em 4 bimestres e também a média final de cada disciplina.

  A terceira página com uma tabela, que equivale ao boletim escolar
sumarizando o desempenho das disciplinas ao
  longo dos bimestres.

  Se o usuário utilizar mais de um nome de aluno ao mesmo tempo a função irá
apresentar um aviso que diz que o
  argumento “media” não está rodando para o padrão “todos” mas utilizou
apenas o primeiro nome inserido na função.
  Os documentos serão gerados normalmente.

Author:

  Guilherme Ferreira Pagoti
  e-mail: guilherme.pagoti@usp.br

Examples:

# exemplo com um valor para média, mas sem especificar o argumento aluno.
  notas(5)

# Exemplo com um valor para média, especificando um aluno que se deseja
obter o documento
  notas(7,”Ana”)

# Exemplo com um valor para média, especificando um grupo de aluno que se
deseja obter os documentos
  notas(6, c(“Breno”,”Gustavo”,”Silvia”))
```

Planilhas exemplo para teste da função:

espanhol.csv

filosofia.csv

historia.csv

matematica.csv

quimica.csv

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:guilherme.pagoti:start

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