



Fernando Ravanini Gardon

Mestrando em recursos hídricos energéticos e ambientais pela FEC/UNICAMP, orientado pela Profª Drª Rozely Ferreira dos Santos (USP/IB).

Título da Dissertação: “Contribuição de restauros florestais para serviços ecossistêmicos hídricos”.

exec

Exercícios

F F F F F F FFFF F F

Trabalho Final

Proposta A

`ecoh.i (x, suplevel=, resume=, graphics=, na.rm=)`

Criar uma função (Ecohydrological Index - `ecoh.i`) que categorize áreas de restauração florestal com base nos aspectos estruturais da vegetação e sua efetividade na manutenção de processos ec hidrológicos que garantem a oferta de serviços ecossistêmicos hídricos para a população. A função gera 2 dataframes com o cálculo do índice, dados resumo (média, mediana, min, max, sd, var) e plota gráficos.

A função pode ser aplicada tanto em diferentes parcelas, quanto em uma floresta como um todo. Os parâmetros da estrutura florestal que compõem a função foram selecionados a partir dos protocolos de monitoramento de florestas brasileiras, levando em consideração a influência dos parâmetros sobre os processos do ciclo hidrológico que ocorrem em florestas tropicais.

Os parâmetros selecionados como dados de entrada da função são: cobertura de dossel (%), número de estratos e densidade de indivíduos arbóreos (indivíduos/ha). A entrada de dados é feita por um data frame, sendo que podem ser inseridos dados coletados por parcela ou da floresta como um todo.

Na função, o usuário deverá indicar qual coluna esta o nível superior dos dados (floresta ou parcela), para que a função calcule dados resumo como média, mediana, etc (arg: `suplevel = 1`, indica que o level superior esta na coluna 1).

O dataframe de entrada está exemplificado abaixo:

(concordo com a sugestão da Melina de que não necessariamente o dataframe fornecido pelo usuário deve estar na mesma ordem que o exemplo, porém não sei como fazer isso, se definindo um

argumento ou então modificando os dados de entrada, ao invés de inserir um dataframe o usuário poderia inserir vetores com os dados de cada uma das variáveis e então a função montava o dataframe; estou seriamente pensando em inserir os dados como vetores. - ESTE É UM PONTO QUE ME GERA BASTANTE DÚVIDA)

floresta	parcela	Cobertura de dossel	Nº de estratos	Densidade de indivíduos
1	1	70	3	1052
1	2	75	4	1200
1	3	65	4	950
2	1	65	4	1103
2	2	50	2	880
2	3	70	2	900
3	1	75	1	750
3	2	80	3	689
3	3	40	1	500

Cada um dos parâmetros possui um peso e é valorizado quanto a sua condição levantada em campo (adequado - 1; parcialmente adequado - 0.65; e inadequado - 0). Esta categorização está relacionada aos protocolos de monitoramento florestal que apresentam as características esperadas para florestas restauradas com mais de 5 anos de idade, sendo eles o Pacto pela Restauração da Mata Atlântica (2009) e a publicação do Cadernos da Mata Ciliar - Vol. 4 (2008). A tabela abaixo apresenta os pesos de cada parâmetro e o valor referente a condição observada do parâmetro:

Parâmetros	adequado (valor-1)	parcialmente adequado (valor-0.65)	inadequado (valor-0)
cobertura de dossel (peso - 0.35)	⇒80	⇒50 e <80	<50
nº de estratos (peso - 0.40)	>2	==2	<2
densidade (peso - 0.25)	⇒1250	⇒880 e <1250	<880

Para obter o índice a função multiplica o valor da condição do parâmetro pelo peso do próprio parâmetro e então soma os valores de cada linha do data frame, normalizando-os de 0 a 1. Em seguida gera um novo dataframe (data.frame.ecoh.i) com 7 colunas, que contém os dados fornecidos pelo usuário acrescentado de uma nova coluna com os valores do índice calculado (coluna 6 ← ecoh.i). Os valores de 0 a 1 são divididos em 5 categorias (muito ruim-0 a 0.2; ruim-0.21 a 0.4; intermediário - 0.41 a 0.6; bom - 0.61 a 0.8; e muito bom - 0.81 a 1) e a função acrescenta mais uma coluna ao dataframe contendo a condição da floresta referente ao índice obtido (coluna 7 ← condição ecoh.i).

floresta	parcela	Cobertura de dossel	Nº de estratos	Densidade de indivíduos	ecoh.i	Condição ecoh.i
----------	---------	---------------------	----------------	-------------------------	--------	-----------------

Caso o dataframe seja composto por varias parcelas de diferentes florestas, o usuário poderá pedir a função que gere um segundo dataframe (data.frame.ecoh.i.resume) contendo 8 colunas, que apresenta as diferentes florestas (col 1), os valores de ihev máximo (col 2) e mínimo (col 3), a mediana (col 4), o sd (col 5), a var (col 6), a média (col 7 - ecoh.i média) e a condição média referente ao valor de ihev (col 8 ← Condição ecoh.i média) de cada floresta. Esta opção poderá ser feita indicando no argumento da função determinado "resume=F ou T", sendo o default do argumento FALSE. É muito importante lembrar que o argumento resume= está ligado ao argumento suplevel=

que irá dizer para a função em qual coluna que está o nível superior dos dados para o cálculo dos dados resumo. Abaixo é apresentado o cabeçalho do data frame gerado com as colunas com os valores resumo:

floresta	ecoh.i mínimo	ecoh.i máximo	ecoh.i mediana	ecoh.i.sd	ecoh.i.var	ecoh.i média	Condição ecoh.i média
-----------------	----------------------	----------------------	-----------------------	------------------	-------------------	---------------------	------------------------------

Por opção do usuário, a função pode plotar 1 boxplot e 2 histogramas (frequência e densidade) referentes aos valores de ecoh.i do dataframe 1). Caso o usuário queira plotar os gráficos isso deve ser indicado pelo argumento `graphics=TRUE`. Se o usuário não indicar o argumento `graphics` o default é `FALSE` e a função não plota os gráficos. Os gráficos resumos possíveis serão 3:

1- histograma de frequência de observações geral, com os dados de ecoh.i de cada linha do dataframe 1;

2- histograma com curva de densidade de probabilidade geral com os dados de ecoh.i de cada linha do dataframe 1;

3- boxplot com a representação dos dados de ecoh.i de cada linha do dataframe 1, sendo cada uma das caixas a representação do dados coletados para determinada floresta.

Resumindo, os produtos da função dependem do tipo de entrada de dados pelo usuário, sendo eles o próprio data frame fornecido pelo usuário com novas colunas contendo os valores de ecoh.i e a condição associada a estes valores, e também gráficos que permitem avaliar a variação dos dados com mais clareza.

Proposta B

O volume de chuva precipitado em superfícies florestais é redistribuído nos ambientes atmosférico e terrestre pelos processos de interceptação pelas copas e infiltração, respectivamente. Em florestas tropicais conservadas o volume interceptado atinge cerca 30% do volume precipitado, ou seja, 1/3 da precipitação incidentes nestas florestas não chega ao solo pois é retido pela estrutura florestal, retornando para a atmosfera e contribuindo para a formação de novas chuvas. Somente os 2/3 restantes atravessam a estrutura e contribuem para a recarga dos reservatórios. No entanto, diferentes estágios de desenvolvimento florestal conferem proporções distantes de interceptação (floresta em estágio inicial - 15%; estágio secundário - 20%; e estágio avançado - 30%), com isso os cálculos envolvendo o balanço hídrico de diferentes regiões podem subestimar ou superestimar estes valores, conduzindo a erros no balanço hídricos e nas tomadas de decisões quanto ao uso dos recursos hídricos.

Neste sentido, busca-se criar uma função (`hydro.data`) que agilize a manipulação de dados pluviométricos e informe o volume de precipitação que efetivamente atinge o solo e contribui para a recarga dos reservatórios em bacias florestais. O dado de entrada da função é composto por um data frame contendo os volumes de chuva em mililitros coletados em pluviômetros durante um período, sendo a primeira coluna referente a identificação da parcela ou da floresta, a segunda coluna com os volumes coletados, a terceira com a área de captação dos pluviômetros e a quarta coluna com o estágio de desenvolvimento da floresta (Inicial-1; secundário-2 ou avançado-3).

A função inicialmente acrescenta uma nova coluna ao data frame fornecido pelo usuário contendo o volume precipitado em milímetros de altura (mm)(5ª coluna ← Altura de precipitação). Uma outra coluna é adicionada contendo o volume que em milímetros que efetivamente atinge o solo(mm)(6ª

coluna ← Volume de recarga), ou seja, esta coluna apresenta o volume total precipitado descontado do volume interceptado (definido de acordo com o estágio da floresta). Abaixo está apresentado o cabeçalho do data frame final que retornará para o usuário:

id	Volume precipitado (ml)	Área de captação do pluviômetros (cm ²)	Estágio de desenvolvimento florestal	Altura de precipitação (mm)	Volume de recarga (mm)
----	-------------------------	---	--------------------------------------	-----------------------------	------------------------

A função gera também 2 gráficos, um boxplot contendo os volumes das chuvas em cada período e suas variações (mm), e um outro gráfico de barras contendo os volumes acumulados em cada floresta ou parcela (mm).

COMENTÁRIOS PROPOSTAS MELINA LEITE

EM 28/04/2016

PROPOSTA A

Achei a proposta simples, factível, e um tanto fácil, acredito que você possa fazer mais por ela. A principal crítica é que a forma com que vc vai tratar a presença de dados hierarquizados (ou seja, parcelas dentro florestas, ou florestas dentro de bacias). Aqui nós temos dois níveis, e acho que vc deveria ter um argumento para que a pessoa possa indicar a coluna que descreve qual o nível hierárquico superior (por exemplo, a floresta), para que possa ser feito os cálculos das médias (e porque não mediana? mínimo? máximo?).

Por exemplo:

floresta	parcela
1		1
1		2
1		3

Eu acho que argumento "graphics=TRUE", nao deveria ser usado para dizer se o usuário quer ou não calcular médias de grupos de dados, e sim apenas para a pessoa dizer se quer que apareçam gráficos resumo dos dados ou não.

Outra coisa, não seria mais interessante a função retornar um outro data.frame (separado do conjunto de dados original) com os resultados sumarizados (media, ..) por floresta? Assim o dados sumarizados por floresta estaria mais facilmente visualizável.

E se o meu data.frame estiver com as colunas em alguma ordem diferente da sua coluna de exemplo? isso inviabilizaria o uso da função? como fazer com que isso não aconteça?

Eu acho que a pessoa deveria poder colocar seu data frame na ordem de colunas que quiser, e também pode escolher se quer ter medidas resumo (média, mediana, minimo..) para níveis de uma coluna de fator (floresta por exemplo). Fazendo essa opção, ela pessoa poderia ou não querer os gráficos dos dados e das medidas resumo plotados.

Eu não entendi os gráficos de dispersão, me pareceu com os gráficos feitos para o cálculo da ANOVA na unha, é isso? Imagine se eu tenho uma tabela com 2000 linhas de dados de parcelas em florestas, você acha que esses gráficos seriam informativos? Haveria outros gráficos mais informativos? Não sei, acho que vc deve pensar nas limitações e utilidade desse tipo de gráfico.

PROPOSTA B

Você só vai calcular a chuva em mm (vindo da coluna de cm) e depois calcular o volume de recarga a partir da porcentagem referente a cada estágio? Muito pouco para uma função, isso pode ser facilmente feito em duas linhas de comando.

Sugiro dar ênfase na proposta 1 e deixar ela redondinha o suficiente, dado o baixou grau de dificuldade. Não se esqueça de um bom HELP!

Enfim, mãos à massa!

REFORMULAÇÃO DA PROPOSTA A

Reescrevi a proposta A modificando alguns pontos sugeridos pela monitora Melina. DÚVIDAS - 1- indicando qual coluna está o nível superior (superlevel=) de dados (floresta ou parcelas), os dados resumo serão calculados de acordo com os levels existentes nesta coluna, certo?

2- caso o usuário não tenha dados de diferentes parcelas para uma floresta, ou seja, somente dados de floresta como um todo (um valor por floresta), e este usuário indicar o argumento resume=T, os dados resumo não serão calculados. Se mesmo assim o usuário quiser os dados resumos referentes ao conjunto de dados total e não para cada floresta, como seria possível isso? indicar em um outro argumento?

3- na situação anterior, se o usuário indicar o argumento graphics=T, como prosseguir com a criação do gráfico boxplot? não da para fazer uma caixa para cada floresta sem diferentes amostragens para uma mesma floresta, e gerar um boxplot com apenas uma caixa não sei se é importante.

COMENTÁRIOS 2 MELINA

em 06/05/2016

Respondendo primeiro às tuas perguntas pontuais:

1. sim, essa é a idéia de ter níveis superiores para se ter medidas resumo. nesse caso, se a pessoa já coloca algum número no argumento superlevel (superlevel=1), a função já vai construir o segundo data frame com as medidas resumo. Não precisa e outro argumento pra isso.
2. bom, o que eu entendi do seu argumento resume=T é que ele retorna as medidas resumo de toda a tabela de dados (ou seja uma linha apenas de informação com min, max, med...). Assim, esse argumento é útil tanto para o usuário que tem nível superior nos dados quanto (superlevel=1 - ou qualquer outra coluna indicando o nivel) para o que não tem (superlevel=NULL - esse deveria ser o default da função) e quer a medida resumo da tabela toda. Por exemplo, para o usuário que vai usar superlevel=null, porque tem tabelas separadas

para cada floresta, ele pode usar esse argumento para ter as medidas resumo para suas florestas em separado.

3. acho que nesse caso, você pode ter UM boxplot com os dados de toda a tabela.

Comentários da proposta reformulada:

sobre os data frames em colunas sem ordem:

- fazer em formato de vetores talvez faça o usuário ter um pouco de preguiça de usar sua função.
- você pode fixar a ordem das figuras (mas eu não acho nem um pouco elegante, mas é mais fácil) e informar explicitamente no help da função que o data frame precisa estar nesse formato, ou
- você pode fazer como fez com o argumento `superlevel`. Pelo que entendi, nele vc indica em qual coluna estão os dados do nível superior no qual se quer tirar dados de medidas resumo. Você pode ter argumentos que indicam em qual coluna está o dado necessário, por exemplo, `cobertura.dossel=3` (os dados de cobertura do dossel estão na coluna 3), `n.estratos=2` (os dados de número de estratos na coluna 2).
- Eu fiz uma sugestão de incrementar a função, mas se ficar muito difícil, pode fazer a tabela como fixa.

sobre os gráficos:

- vejo que o histograma 1 e a curva de densidade de probabilidade (hist 2) podem ser desenhados num mesmo plot. Tente fazer assim, pois poupa “tinta” e os dois trazem a mesma informação de forma um pouco diferente. Se você tiver dificuldades (veja na internet, certamente tem roteiros sobre isso), sugiro ficar só com o histograma de frequências.
- Com eu disse antes se a pessoa não tiver nível superior (`superlevel=NULL`), daí pode aparecer um só boxplot com os dados totais.

Tudo certo? Mãos à massa!

Finalização Proposta A

Ok Melina. Consegui plotar os gráficos juntos. Acabei optando por inserir um novo argumento (`per.group=TRUE` ou `FALSE`) à função caso o usuário queira fazer os dados resumo por subset ou com todos os dados. Inseri também duas mensagens de erro. Caso o dado de entrada não seja `data.frame` a função retorna uma mensagem de erro. Caso o usuário queira dados resumo por grupo mas não tenha diferentes grupos no `data.frame` de entrada (por exemplo parcelas de uma floresta) a função retornará uma mensagem de erro e não rodará. Obrigado pela ajuda e até mais.

HELP DA FUNÇÃO A (`ecoh.i`)

```
ecoh.i                package:nenhum                R Documentation

Cálculo de um Índice Ecohidrológico (Ecohydrological Index – ecoh.i).

Description:
```

A função calcula um índice baseado em variáveis da estrutura e composição florestal que determinam a ocorrência de diferentes processos hidrológicos. Os valores de cada variável são multiplicados pelo peso da variável e da condição referente ao valor desta variável. Gráficos boxplot e histogramas de frequência são plotados.

Usage:

```
ecoh.i <- function(x, suplevel, fcover, strata, densityh, resume=F,  
per.group, graphics=FALSE)
```

Arguments:

x Um data.frame contendo os valores das variáveis, podendo as colunas estarem em qualquer ordem. Deve haver ao menos uma coluna referente ao nível de análise (floresta, parcela, etc).

suplevel Indica em qual coluna do data.frame está o level superior de análise, para que a partir desta coluna sejam feitos os cálculos.

fcover Indica qual coluna do data.frame contém os dados de cobertura florestal.

strata Indica qual coluna do data.frame contém o número de estratos florestais.

densityh Indica qual coluna do data.frame contém os valores de densidade de indivíduos arbóreos.

resume Argumento opcional. Opção para calcular dados resumo do índice (max,min,sd,var,média,mediana).

per.group Argumento condicionado ao argumento resume. Indica se os dados resumos serão agrupados por level ou calculado a partir dos dados todos.

graphics Opção para plotar gráficos dos dados.

Details:

A função calcula um índice de qualidade hidrológica baseada em variáveis da estrutura e da composição vegetal de restauros florestais, categorizando o índice de acordo com a sua condição (muito bom, bom, regular, ruim, muito ruim).

As variáveis utilizadas no índice são os argumentos da função, sendo elas: Cobertura florestal (fcover – valores em porcentagem); Número de estratos (strata – contagem do estratos presentes); e Densidade de indivíduos arbóreos por hectare(densityh – contagem de indivíduos arbóreos - extrapolação para uma área de 1ha se necessário).

Para o cálculo do índice os dados são normalizados entre 0 e 1. Pesos são atribuídos a cada variável (fcvoer-0.35; strata-0.40; densityh-0.25) e também a cada valor (cada valor de cada variável é multiplicado por um peso dependendo da sua magnitude (ex: se para a variável fcover um valor maior que 80% de cobertura é considerado uma condição adequada pela literatura, então o peso da variável é multiplicado por 1; um valor intermediário seria multiplicado por 0.65 e um valor inadequado multiplicado por zero).

O data.frame de entrada deve conter valores em todas as linhas de cada variável para que seja possível calcular o índice, caso haja NAs a função não roda.

Value:

Se resume==TRUE, a função calcula dados resumo do índice, porém nesta condição o argumento per.group deve ser especificado para dizer a função para calcular o resumo por subset ou a partir de todos os dados.

Se per.group==TRUE, a função calcula o resumo para cada subset do data.frame, caso per.group==FALSE o resumo é calculado para o conjunto geral de dados.

Se graphics==TRUE, a função plota um gráfico boxplot (se houver levels inferiores de organização) e um histograma de frequências.

Note:

Caso o dado de entrada no argumento x não seja um data.frame, a função retorna uma mensagem de erro.

Caso seja indicado resume==TRUE e per.group==TRUE mas os dados não apresentarem nenhum subset, a função retorna uma mensagem de erro dizendo exatamente isto e não roda.

Author(s):

Fernando Ravanini Gardon
fernadogardon@hotmail.com

References:

Rodrigues et al. (2009). Pacto pela restauração da mata atlântica : referencial dos conceitos e ações de restauração florestal. 256p.
Gandara, F.B; Uehara, T.H.K. (2011). Monitoramento de áreas em recuperação. Cadernos da mata ciliar. SMA/SP. Vol 4. P68.

Examples:

```

plots <- rep(seq(1,15),10)
forest.cover <- round(runif(150,0,100))
ind.density <- round(runif(150,0,1666))
number.strata <- round(runif(150,1,5))
forest <- rep(seq(1,15),each=10)
dados<-data.frame(plots,forest.cover,ind.density,number.strata,forest)

###com dados resumo por subset
indice.ecohidro1 <- ecoh.i(x=dados,suplevel = 5,fcover=2,strata=4,densityh =
3, resume=T, per.group=T, graphics=F)

###resumo dos dados em geral
indice.ecohidro1 <- ecoh.i(x=dados,suplevel = 5,fcover=2,strata=4,densityh =
3, resume=T, per.group=F, graphics=F)

###opção por dados resumo separado por subset retornando erro pois não há
subset
forest <- seq(1,15)
plots <-seq(1,15)
forest.cover <- round(runif(15, 0,100),2)
number.strata <- round(runif(15,1,5),2)
ind.density <- round(runif(15,0,1666),2)
dados<-data.frame(plots,forest.cover,ind.density,number.strata,forest)
indice.ecohidro1 <- ecoh.i(x=dados,suplevel = 5,fcover=2,strata=4,densityh =
3, resume=T, per.group=F, graphics=F)

```

CÓDIGO DA FUNÇÃO ECOH.I

```

ecoh.i <- function(x, suplevel, fcover, strata, densityh, resume=F,
per.group, graphics=FALSE)
{
  if(class(x) != "data.frame"){
    #####identificando o
    tipo de dado de entrada
    stop("The input data is not a data.frame.") ##### caso não seja um
    data.frame a função retorna amensgaem de erro
  }
  else{} ##### caso seja a função
  continua normalmente
  na.excluí(x)
  dataf.var <-data.frame(x[,suplevel],x[,fcover],x[,strata],x[,densityh])
  #####lendo os dados indicados pelo usuário e reorganizando-os no data.frame da
  função
  colnames(dataf.var)<- c("suplevel", "fcover", "strata", "densityh")
  #####mudando os nomes das colunas
  dataf.var$suplevel <- as.factor(dataf.var$suplevel)
  #####transformando a coluna representativa do nível superior em fator
  dataf.var[dataf.var$fcover<50,2]<-0.35*0
  #####transformando os dados das variáveis pelos valores das variáveis
  multiplicados pelo seu respectivo peso e condição
  dataf.var[dataf.var$fcover>=80,2]<-0.35*1

```

```
###idem
  dataf.var[dataf.var$fcover>=1,2]<-0.35*0.65
###idem
  dataf.var[dataf.var$strata<2,3]<-0.40*0
###idem
  dataf.var[dataf.var$strata==2,3]<-0.40*0.65
###idem
  dataf.var[dataf.var$strata>2,3]<-0.40*1
###idem
  dataf.var[dataf.var$densityh<880,4]<-0.25*0
###idem
  dataf.var[dataf.var$densityh>=1250,4]<-0.25*1
###idem
  dataf.var[dataf.var$densityh>1,4]<-0.25*0.65
###idem
  dataf.var$ecoh.i.par <- apply(dataf.var[,2:4],MARGIN=1 ,FUN=sum)
###cria a coluna com o valor do indice (ecoh.i.par) referente a soma
dosvalores das variáveis em cada linha
  dataf.var$ecoh.i.par.cond <- dataf.var$ecoh.i.par
###cria a coluna com os mesmo valores da coluna anterior para transformá-los
em condições
  dataf.var$fcover <- x[,fcover] ### repõe
nas colunas de variaveis os valores fornecidos pelo usuário
  dataf.var$strata <- x[,strata]
###idem
  dataf.var$densityh <- x[,densityh]
###idem
  dataf.var[dataf.var$ecoh.i.par.cond >= 0.8 ,6]<- "muito bom"
###transforma os valores da coluna em condição (muito bom, bom, etc...)
  dataf.var[dataf.var$ecoh.i.par.cond >=0.6 & dataf.var$ecoh.i.par.cond
<0.8,6]<- "bom" ###idem
  dataf.var[dataf.var$ecoh.i.par.cond >= 0.4 & dataf.var$ecoh.i.par.cond
<0.6,6]<- "regular" ###idem
  dataf.var[dataf.var$ecoh.i.par.cond > 0.2 & dataf.var$ecoh.i.par.cond
<0.4,6]<- "ruim" ###idem
  dataf.var[dataf.var$ecoh.i.par.cond < 0.2,6] <- "muito ruim"
###idem
  if(resume==FALSE) ###caso o
usuario não queira dados resumo
  {
    if(graphics==TRUE) ###mas
queira o gráfico gral dos dados
  {
    x11() ###abre janela gráfica
    hist(dataf.var$ecoh.i.par, main="", xlab="ecoh.i") #### plota o
histograma com as frequencias de todos os dados do data.frame de entrada
par(new=T) ###condição para plotar no gráfico anterior
    plot(density(dataf.var$ecoh.i.par),axes=F,ann=F,xlim=c(0,1),col="red")
###plota a linha de densidade em cima do histograma de frequência
    axis(4) ###adiciona a escala de densidade no eixo 4
```

```

    mtext("Density", side=4, padj=3, cex=1) ###adiciona nome ao eixo 4
  }
  else{}      ###caso o usuário não queira gráficos a função segue
  return(dataf.var) ###retorna o data.frame com os índices e as
condições referentes ao índice obtido em cada linha
}
else      ###caso usuário queiro os dados resumo
{
  if(per.group==TRUE)  ###caso queira os dados resumo tem que adicionar
este argumento separados indicando se quer os dados resumo referente a todos
os dados do data.frame ou por subset (parcelas por exemplo).
  {
    dataf.var.med <- data.frame(levels(dataf.var$suplevel))      ###cria
data.frame para os dados resumo
    dataf.var.med$ecoh.i.min <- tapply(dataf.var$ecoh.i.par,
dataf.var$suplevel, min)  #####cria coluna com o valor mínimo do índice para
cada sublevel
    dataf.var.med$ecoh.i.max <- tapply(dataf.var$ecoh.i.par,
dataf.var$suplevel, max)  #####cria coluna com o valor máximo
    dataf.var.med$ecoh.i.sd <- tapply(dataf.var$ecoh.i.par,
dataf.var$suplevel, sd)  #####cria coluna com o valor de sd
    dataf.var.med$ecoh.i.var <- tapply(dataf.var$ecoh.i.par,
dataf.var$suplevel, var)  #####cria coluna com o valor da var
    dataf.var.med$ecoh.i.median <- tapply(dataf.var$ecoh.i.par,
dataf.var$suplevel, median)  #####cria coluna com o valor da mediana
    dataf.var.med$ecoh.i.media <- tapply(dataf.var$ecoh.i.par,
dataf.var$suplevel, mean)  #####cria coluna com o valor da média
    dataf.var.med$condicao.ecoh.i.media <- tapply(dataf.var$ecoh.i.par,
dataf.var$suplevel, mean)  #####cria coluna com o valor da média do índice
para transformar em condição
    dataf.var.med[dataf.var.med$condicao.ecoh.i.media >= 0.8,8]<- "muito
bom"      ###muda os valores do índice na coluna para condição
    dataf.var.med[dataf.var.med$condicao.ecoh.i.media >=0.6 &
dataf.var.med$condicao.ecoh.i.media <0.8,8]<- "bom"      ###idem
    dataf.var.med[dataf.var.med$condicao.ecoh.i.media >= 0.4 &
dataf.var.med$condicao.ecoh.i.media <0.6,8]<- "regular"  ###idem
    dataf.var.med[dataf.var.med$condicao.ecoh.i.media > 0.2 &
dataf.var.med$condicao.ecoh.i.media <0.4,8]<- "ruim"      ###idem
    dataf.var.med[dataf.var.med$condicao.ecoh.i.media < 0.2,8] <- "muito
ruim"      ###idem
    colnames(dataf.var.med) <- c("suplevel","ecoh.i.min","ecoh.i.max",
"ecoh.i.sd","ecoh.i.var","ecoh.i.median","ecoh.i.media","condicao.ecoh.i.med
ia")  ###muda os nomes das colunas do data.frame
    resume.list <- list(dataf.var,dataf.var.med)  ###salva o data.frame
com dados geral (dataf.var) e com dados resumo (dataf.var.mee) em um objeto
da categoria lista
    if(!dataf.var[1,1]==dataf.var[2,1])  ###porém, caso a primeira
coluna do data.frame fornecido pelo usuário não apresente repetições, ou
seja, caso a primeira linha da coluna do level superior indicado pelo
usuário não seja igual a segunda linha da mesma coluna
  {

```

```
stop("There are no subset levels to resume.") ####a função retorna
uma mensagem de erro dizendo que não há como resumir os dados pois não há
repetições de uma mesma área (parcela de uma floresta por exemplo, ou
florestas de uma mesma bacia)
}
else{} ####caso haja repetições a função segue
if(graphics==TRUE) ####caso o usuário queira gráficos
{
  x11() ####abre janela gráfica
  par(mfrow=c(1,2), mar=c(5,5,5,5)) ####ajuste da janela gráfica
  boxplot(dataf.var$ecoh.i.par~dataf.var$suplevel,xlab="Level of
analyses", ylab="ecoh.i", cex=1) ####plota boxplot com os dados de cada
level
  abline(h=mean(dataf.var$ecoh.i.par),col="red") ####adiciona uma
linha com amédia da variável y
  text(x=2.5,y=c(mean(dataf.var$ecoh.i.par)+0.05),labels="mean",col="red")
####adiciona o texto na linha da média
  hist(dataf.var$ecoh.i.par, main="", xlab="ecoh.i") ####plota o
histograma com afrequencia de valores
  par(new=T) ####seleciona para plotar no mesmo gráfico do histograma
plot(density(dataf.var$ecoh.i.par),axes=F,ann=F,xlim=c(0,1),col="red")
####plota a inha com a densidade no histograma de frequência
  axis(4) ####adiciona o eixo 4
  mtext("Density", side=4, padj=3, cex=1,col="red") ####adicono texto
ao eixo 4
}
else{} ####caso não queira gráficos a função não plota nada
return(resume.list) ####retorna no console o objeto já criado contendo
a lista com os data.frames de dados geral e dados resumo
}
else ####caso o usuário não queira os dadosdividios por subset a função
faz o dado resumo geral
{
  general.resume <- 1 ####cria a primeira coluna do data.frame de dados
resumo
  dataf.var.med1 <- data.frame(general.resume) ####cria o data.frame
dedados resumo
  dataf.var.med1$ecoh.i.min1 <- apply(dataf.var[5],MARGIN=2, FUN=min)
####cria coluna com o valor mínimo do índice para os dados em geral
  dataf.var.med1$ecoh.i.max1 <- apply(dataf.var[5],MARGIN=2, FUN=max)
####cria coluna com o valor máximo
  dataf.var.med1$ecoh.i.sd1 <- apply(dataf.var[5],MARGIN=2, FUN=sd)
####cria coluna com o valor de sd
  dataf.var.med1$ecoh.i.var1 <- apply(dataf.var[5],MARGIN=2, FUN=var)
####cria coluna com o valor de var
  dataf.var.med1$ecoh.i.median1 <- apply(dataf.var[5],MARGIN=2,
FUN=median) ####cria coluna com o valor da mediana
  dataf.var.med1$ecoh.i.medial <- apply(dataf.var[5],MARGIN=2, FUN=mean)
####cria coluna com o valor da média
  dataf.var.med1$condicao.ecoh.i.medial <- apply(dataf.var[5],MARGIN=2,
```

```

FUN=mean) #####cria coluna com o valor médio do índice para todo o conjunto
de dados
  dataf.var.med1[dataf.var.med1$condicao.ecoh.i.medial >= 0.8 ,8]<-
"muito bom" #####muda os valores do índice na coluna para condição
  dataf.var.med1[dataf.var.med1$condicao.ecoh.i.medial >=0.6 &
dataf.var.med1$condicao.ecoh.i.medial <0.8,8]<-"bom" #####idem
  dataf.var.med1[dataf.var.med1$condicao.ecoh.i.medial >= 0.4 &
dataf.var.med1$condicao.ecoh.i.medial <0.6,8]<- "regular" #####idem
  dataf.var.med1[dataf.var.med1$condicao.ecoh.i.medial > 0.2 &
dataf.var.med1$condicao.ecoh.i.medial <0.4,8]<-"ruim" #####idem
  dataf.var.med1[dataf.var.med1$condicao.ecoh.i.medial < 0.2,8] <-"muito
ruim" #####idem
  general.resume <- list(dataf.var,dataf.var.med1) #####salva os
data.frames emum objeto da classe lista
  if(graphics==TRUE) #####caso usuário queira gráficos
  {
    x11() #####abre janela gráfica
    par(mfrow=c(1,1), mar=c(5,5,5,5)) #####ajusta janela gráfica
    hist(dataf.var$ecoh.i.par, main="", xlab="ecoh.i") #####plota o
histograma de frequência
    par(new=T) #####seleciona para plotar no mesmo gráfico do
histograma
    plot(density(dataf.var$ecoh.i.par),axes=F,ann=F,xlim=c(0,1),col="red")
#####plota a curva de densidade dos dados no gráfico e histograma
    axis(4) #####adiciona o eixo 4
    mtext("Density", side=4, padj=3, cex=1,col="red") #####adicono
texto ao eixo 4
  }
  else{} #####caso não queira gráficos a função não plota nada
  }
  return(general.resume) #####retorna no console o objeto da classe
lista contendo os data.frames com os dados em geral e o resumo destes dados
}
}

```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:fernandogardon:start

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