

Isadora Aguirra



Graduada em Gestão Ambiental pela EACH (USP - leste). Atualmente está desenvolvendo um projeto de pesquisa sobre o Bolsa Família e o uso de recursos naturais.

exec

TRABALHO FINAL

Isadora, siga com a proposta A. Mas importante, faça com que sua função opere um data frame com um conjunto de dados de várias famílias. Ou seja, nos dados de entrada as colunas são as variáveis e as linhas as famílias. Como as variáveis podem ser agrupadas em 3 categorias, utilize um outro argumento de entrada que defina a classe de cada variável (um vetor com as 3 categorias na posição de cada coluna do dados de entrada), assim vc. pode operar os dados definido qual tipo de variável deve ser utilizada. Sugiro tb. que calcule o Gini e o Gini decomposto como vc. mesma sugeriu. Arrume a propostas com essas modificações e pode começar a trabalhar.

— [Alexandre Adalardo de Oliveira](#) 2016/04/30 08:52

Resposta Isadora 04/05

Professor, muito obrigada pelas correções. Fiz as alterações sugeridas na proposta A. Mas acredito que seria mais interessante caso o número de categorias em que as variáveis de renda serão agrupadas não fosse fixo (acima você sugeriu 3 categorias), ou seja, que o número de categorias pudesse ser definido pelo usuário. Acredito que dessa forma as possibilidades de aplicação da função seriam mais abrangentes. Pode ser assim ou isto tornaria o desenvolvimento da função muito complicado?

Obrigada,

Isadora

Dificuldade encontrada 12/05

Professor e monitor, estou com problemas, o pacote IC2 do R não tem uma função para calcular a decomposição do coeficiente de Gini por fonte de renda. O pacote traz a função `decompSGini` que realiza a decomposição por subgrupos na população e não é o caso. Depois de uma ampla consulto e de verificar em outros pacotes, conclui que a função ainda não existe. Nesse caso, gostaria da autorização de

vocês para remover o item c) da proposta. O item c é concertar a decomposição por fonte de renda. Certamente é possível operacionalizá-lo no R, contudo uma função que realizasse essa tarefa renderia uma outra proposta de trabalho final.

Obrigada, Isadora

Proposta A

Contextualização

As famílias pobres que habitam as áreas rurais de países em desenvolvimento costumam apresentar formas de geração de renda altamente diversificadas (SUNDERLIN et al, 2005). A sobrevivência desses grupos pode depender de recursos originários de ambientes não cultivados, como florestas e rios (ANGELSEN, 2011). Dessa forma, a compreensão das diferentes dimensões dos meios de vida rurais e da pobreza não deve se restringir a medidas quantitativas de renda em dinheiro (SUNDERLIN et al, 2005). É possível distinguir as fontes de renda de famílias rurais em pelo menos três categorias principais: (i) renda ambiental que diz respeito à contribuição de produtos ambientais provenientes de terras não agrícolas (i.e. madeira, lenha, carne de caça, frutas e animais silvestres); (ii) renda agrícola que refere-se à soma de rendimentos de produção de subsistência e de rendimentos da pecuária e salários de atividades relacionadas; (iii) renda não agrícolas que compreende salários, pensões e transferências do governo.

Utilidade da função

A função proposta tem a finalidade de facilitar a operacionalização de dados referentes aos rendimentos de populações pobres rurais. Para tanto, realizará análises comuns neste tipo de estudo a partir de uma variável categórica ou quantitativa contínua definida pelo usuário. A função permitirá estimar a participação relativa e absoluta de cada fonte de renda e de seus componentes, o grau de desigualdade de renda por meio do coeficiente de Gini e a contribuição de cada fonte de renda para o coeficiente de Gini.

Entrada

Primeiro argumento: um data.frame que contenha um conjunto de dados referente as fontes de renda das amostras (i.e. famílias, unidades domésticas, indivíduos). O data.frame deve estar organizado da seguinte maneira: a) as linhas devem corresponder às amostras, ou seja, famílias, unidades domésticas ou indivíduos para os quais as informações de renda foram coletadas.

b) A primeira coluna deve conter a identificação das amostras.

c) As demais colunas devem corresponder as variáveis de renda já convertidas para uma unidade comum (i.e. reais, dólares).

Segundo argumento: variável categórica ou quantitativa contínua utilizada como critério para apresentação e divisão dos dados.

Terceiro argumento: corresponderá às categorias que o usuário pretende incluir na análise da renda, ou seja, ao total de categorias nas quais as variáveis de renda serão agrupadas. O usuário deverá indicar as colunas do data frame que correspondem a cada categoria de renda e assim definir como o agrupamento deve ser realizado.

Saída

Para cada critério estabelecido no segundo argumento da função (variável categórica ou contínua) a função retornará:

- a) um data frame em que as linhas corresponderão às categorias de renda e seus componentes e as colunas corresponderão ao valor absoluto e relativo para cada categoria de renda e seus componentes.
- b) um data frame em que as linhas corresponderão às categorias de renda, a primeira coluna corresponderá à identificação das categorias de renda e a segunda coluna corresponderá ao coeficiente de Gini¹ para cada categoria.
- c) um data frame em que as linhas corresponderão às categorias de renda, a primeira coluna corresponderá à identificação das categorias de renda e as colunas seguintes corresponderão à decomposição do coeficiente de Gini² para cada categoria de renda.

¹ O coeficiente de Gini reflete o grau de desigualdade de renda através de um grupo e varia de 0 (completa igualdade) a 1 (apenas 1 família tem toda a renda).

² A decomposição do coeficiente de Gini mostra a contribuição de cada fonte de renda para o coeficiente de Gini e o efeito marginal de cada fonte sobre a desigualdade de renda total

Referências

SUNDERLIN, W. D; ANGELSEN, A; BELCHER, B., BURGUES, P.; NASI, R.; SANTOSO, L.; WUNDER, S. Livelihoods, forests, and conservation in developing countries: an overview. *World development*, v. 33, n. 9, p. 1383-1402, 2005.

ANGELSEN, A.; LARSEN, H. O.; LUND, J., F.; SMITH-HALL, C.; WUNDER, S. *Measuring livelihoods and environmental dependence: Methods for research and fieldwork*. London: Earthscan, 2011.

Proposta B

Contextualização

A função será utilizada na etapa de análise exploratória do dados, etapa que costuma consumir grande parte do tempo das análises e na qual os gráficos são muito importantes, pois permitem assimilação clara e rápida das informações coletadas. A ideia é fornecer um panorama geral ao usuário relacionando a variável preditora com as demais.

Função `rela.data()`

A função proposta utilizará técnicas de estatística descritiva para relacionar as variáveis de um data frame aos pares, retornando ao usuário gráficos exploratórios e testes de hipóteses. O usuário deverá indicar a variável preditora que deseja associar com as demais variáveis do data frame, sendo que esta poderá ser categórica ou quantitativa.

Entrada: data frame em que cada coluna refere-se a uma variável.

Saída: a função retornará gráficos e testes diferentes de acordo com os tipos de variáveis envolvidas, conforme descrito abaixo:

- a) Ambas são categóricas – a função retornará um teste Qui-Quadrado e a tabela de análise dos

resíduos, bem como o gráfico de colunas por estratos da segunda variável;

b) Ambas são quantitativas (contínuas ou discretas) - a função retornará um histograma com as frequências absolutas, um histograma com as frequências relativas, um diagrama de dispersão entre as duas variáveis e sua respectiva linha de tendência e o Coeficiente de Correlação de Pearson.

c) Uma variável é categórica e outra quantitativa - a função retornará um gráfico de colunas por estratos da variável categórica e o gráfico BOX-PLOT por categoria.

Comentários Bruno

Oi Isadora,

Na proposta A, gostaria que você me explicasse melhor o objeto de entrada e saída. Pelo que eu entendi o objeto de saída é o objeto de entrada "resumido".

Na proposta B, o que você quer dizer com "relacionar a variável preditora com as demais"? Seria uma preditora e várias respostas? Além disso, acho que você precisa tomar cuidado com o que a função vai fazer com os NAs presentes no conjunto de dados. No objeto de saída b), talvez você possa verificar a normalidade dos seus dados, e caso não sejam normais, usar outros testes de correlação quem lidam com dados não normais.

Bruno

Resposta Isadora

Olá Bruno, obrigada pelos comentários. Quanto a objeto de entrada na proposta A, seria um data frame semelhante ao abaixo para cada fonte de renda:



Logo, não se trata de resumir, mas apresentar os valores absolutos e relativos da soma de todos os tipos de renda (ambiental, agrícola e não agrícola), tendo como critério uma variável categórica ou quantitativa contínua escolhida pelo usuário. No caso da minha pesquisa, por exemplo, seria interessante separar quem recebe renda do programa bolsa família de quem não recebe. Eu gostaria bastante de trabalhar com as rendas na função, caso você tenha achado simples, há também outras análises que seriam interessantes que a função devolvesse, como o desvio padrão, o valor mínimo e máximo para cada fonte de renda. Outro retorno possível é o coeficiente de Gini ou ainda, a decomposição do coeficiente de Gini. O coeficiente de Gini reflete o grau de desigualdade de renda através de um grupo e varia de 0 (completa igualdade) a 1 (apenas 1 família tem toda a renda). Já a decomposição mostra a contribuição de cada fonte de renda para o coeficiente de Gini e o efeito marginal de cada fonte sobre a desigualdade de renda total. Veja exemplos (a) coeficiente de Gini, (b) decomposição do coeficiente. Obs: não inclui por receio de que a execução fosse muito complicada, mas notei que já existe uma função para o cálculo do coeficiente de Gini.

a) 

b) 

Quanto ao plano B, inicialmente pensei que seria interessante uma função que relacionasse uma variável preditora com as variáveis resposta (sempre aos pares), você acha que não poderia ser assim? Quanto aos NAs, acredito que seria interessante incluir um teste lógico e uma mensagem de alerta para a presença de NAs em cada relação (preditora e resposta) desenvolvida. Mas o que você sugere é que eu pense eu como a função tratará os NAs em cada caso? Gostei bastante da sua proposta de verificar a normalidade dos dados, certamente vou incluir caso eu siga com a proposta B.

Obrigada!

Isadora

Função desenvolvida - Proposta A

Página de Ajuda

renda.opera package: nenhum R Documentation

Função para calcular a renda absoluta, a renda relativa e o coeficiente de desigualdade de Gini para três categorias de renda definidas pelo usuário.

Description:

A partir de um dataframe, divide os dados em subconjuntos de acordo com variável categórica binária ou quantitativa contínua. Depois disso, as variáveis de renda que compõe cada subconjunto são agrupadas em três categorias e para cada categoria são calculados a renda relativa, a renda absoluta e o índice de Gini.

Usage:

```
renda.opera <- function(renda, var, cat1, cat2, cat3)
```

Arguments:

renda: Data.frame que deve estar organizado da seguinte maneira: (i) as linhas devem corresponder às amostras, ou seja, famílias, unidades domésticas ou indivíduos para os quais as informações de renda foram coletadas; (ii) a primeira coluna deve conter a identificação das amostras; (iii) as demais colunas devem corresponder as variáveis de renda.

Var: Valor numérico; número que indica a coluna do data.frame correspondente a variável categórica binária ou quantitativa contínua que orientará a divisão do data.frame nos subgrupos da análise.

Cat1: Valor numérico; número ou sequência de números que indicam a(s) coluna(s) do data.frame correspondente(s) a primeira categoria.

Cat2: Valor numérico; número ou sequência de números que indicam a(s) coluna(s) do data.frame correspondente(s) a segunda categoria.

Cat3: Valor numérico; número ou sequência de números que indicam a(s) coluna(s) do data.frame correspondente(s) a terceira categoria.

Details:

As variáveis de renda devem estar convertidas em uma unidade comum (i.e. reais, dólares).

Todos os argumentos da função precisam ser fornecidos.

O pacote 'ineq' deve estar instalado.

O pacote 'data.table' deve estar instalado.

Value:

comp1 : No console, um data frame correspondente a cada subgrupo do data frame "renda" fornecido na entrada dos dados, em que as linhas correspondem às categorias de renda e seus componentes e as colunas correspondem ao valor absoluto e relativo para cada categoria de renda, seus componentes e o total para todas as categorias.

comp2 : No console, também para cada subgrupo do data frame "renda", a função retornara um data frame em que as linhas correspondem as categorias de renda, a primeira coluna corresponde à identificação das categorias de renda e a segunda coluna corresponde ao coeficiente de Gini para cada categoria.
comp3 : retorna uma lista contendo os data.frames criados.

Warning:

A função será interrompida e retornará mensagens de erro caso o objeto

de entrada renda não seja
data.frame; caso a coluna com que compõe a variável que dividirá o
data.frame em subgrupos (var)
não seja indicada corretamente; caso os pacote 'ineq' e data.table não
estejam instalados. Por fim,
na primeira vez em que for executada a função exibirá uma mensagem
referente a instalação do pacote
'ineq' e referente a instalação do pacote 'data.table'.

Note:

Esteja atento a presença de NAs nos dados, pois estes serão excluídos.

Author(s):

Isadora Aguirra
isadora.aguirra@usp.br

References:

Pacote 'ineq': <https://cran.r-project.org/web/packages/ineq/index.html>

Pacote 'data.table':

<https://cran.r-project.org/web/packages/data.table/index.html>

See Also:

Funções: for(), subset(), unique()

Acknowledgments:

Aos monitores Diogo Melo e Vitor Rios pelos esclarecimentos, dicas e sugestões.

Examples:

```
#Exemplo 1:
#criando vetores para incluir no dataframe
familia <- c("a1", "a2", "a3", "a4", "a5", "a6", "a7", "a8", "a9",
"a10" )
var1 <- c(0,1,0,1,1,0,1,1,1,0)
fruta.col <- round((runif(10, min = 15, max = 20)), digits = 0)
caça<- round((runif(10, min = 0, max = 15)), digits = 0)
madeira <- round((runif(10, min = 15, max = 150)), digits = 0)
sementes <- round((runif(10, min = 0, max = 15)), digits = 0)
verduras.quintal <- round((runif(10, min = 15, max = 100)), digits = 0)
verduras.roca <- round((runif(10, min = 10, max = 190)), digits = 0)
bolsa.familia <- round((runif(10, min = 77, max = 175)), digits = 0)
aposentaria <- round((runif(10, min = 678, max = 4159)), digits = 0)
trabalho.externo <- round((runif(10, min = 678, max = 2000)), digits =
```

```
0)
#criando dataframe para exemplo
renda <- data.frame(familia,var1, fruta.col, caça, madeira, sementes,
verdurinhas.quintal,
+ verdurinhas.roca, bolsa.familia, aposentaria, trabalho.externo)
renda.opera (renda, var=2, cat1=3:6, cat2=7:8, cat3=9:11)
renda.opera (renda, var=2, cat1=8:11, cat2=5:7, cat3=3:14)

#Exemplo 2:
#semelhante ao exemplo 1, mas com var não binária.
familia <- c("a1", "a2", "a3", "a4", "a5", "a6", "a7", "a8", "a9",
"a10" )
fruta.col <- round((runif(10, min = 15, max = 20)), digits = 0)
caça<- round((runif(10, min = 0, max = 15)), digits = 0)
madeira <- round((runif(10, min = 15, max = 150)), digits = 0)
sementes <- round((runif(10, min = 0, max = 15)), digits = 0)
var1 <- c(20,10,20,30,40,20,30,30,40,30)
verdurinhas.quintal <- round((runif(10, min = 15, max = 100)), digits = 0)
verdurinhas.roca <- round((runif(10, min = 10, max = 190)), digits = 0)
trabalho.externo <- round((runif(10, min = 678, max = 2000)), digits =
0)

#criando dataframe para exemplo
renda <- data.frame(familia, fruta.col, caça, madeira, sementes, var1,
verdurinhas.quintal,
+ verdurinhas.roca, trabalho.externo)
renda.opera (renda, var=6, cat1=2:5, cat2=7:8, cat3=9)
```

Código da Função

```
renda.opera <-function(renda, var, cat1, cat2, cat3)
{
  ##### Verificando os argumentos fornecidos pelo usuário #####
  #Verifica se renda é um data.frame e em caso negativo exibe mensagem de erro
  if (class (renda)!="data.frame")
    stop("\n renda nao é um dataframe.\n")
  #Verifica se a coluna com os dados da variável que orientará a divisao do
data.frame em subgrupos foi indicada corretamente e em caso negativo exibe
mensagem de erro
  if (class(var) != "numeric")
    stop("\n Coluna nao indicada corretamente.\n")
  #Verifica se o pacote 'data.table' foi instalado e apresenta mensagem de
erro em caso negativo.
  if (require(data.table) == "FALSE")
    stop("\n 0 Pacote 'data.table' não foi instalado corretamente.\n")
  #Verifica se o pacote 'ineq' foi instalado e apresenta mensagem de erro em
caso negativo.
  if (require(ineq) == "FALSE")
    stop("\n 0 Pacote 'ineq' não foi instalado corretamente.\n")
  ##### Arrumando os dados e determinando critérios de divisao #####
  # Os passos a seguir correspondem aos procedimentos para identificar os
```



```

criterios que orientarao: (i) divisao do data.frame em subgrupos; (ii)
variaveis que formarao cada categoria de renda.
renda = na.omit(renda) # remove os NAs
subgrupos <- unique(renda[,var])# identifica os diferentes valores
presentes na coluna "var" e portanto, os valores que a variavel devera
assumir durante os ciclos
list.subgrupos <- length(subgrupos) # conta o numero de subgrupos totais
que serao formados a partir do data.frame renda
saida1 <- vector("list", list.subgrupos)# cria uma lista em que os
data.frames gerados para os diferentes subgrupos serão guardados
library(data.table)# carrega o pacote data.table, o pacote foi necessario
pois apresenta maior facilidade do que a funcao "for" para guardar em uma
lista a sequencia de data.frame gerados nos ciclos.
# # # # Ciclo 1 # # # #
# 0 primeiro ciclo gera um data.frame para cada subgrupo do data.frame
renda que contem os valores absolutos e relativos para cada categoria de
renda e também para o total (categoria1 + categoria2 + categoria2)
cicl01 <- lapply(subgrupos,
                 function(i) # os valores assumidos pela variável i no ciclo
devem corresponder ao vetor de subgrupos correspondentes aos valores do
argumento "var"
                 # Nas linhas abaixo sao selecionadas as colunas que compoe as categorias
1, 2 e 3 segundo os cristerio definidos pelo usuario. Note que sao gerados
subgrupos do data.frame "renda" - subset(renda, renda[,var]==i.
                 {
                   categoria1 <- (subset(renda, renda[,var]==i)[,cat1])
                   categoria2<- (subset(renda, renda[,var]==i)[,cat2])
                   categoria3<- (subset(renda, renda[,var]==i)[,cat3])
                   # # # # # # # # Calculando valores absolutos por Categoria # # # # # # #
#
                   #Categoria 1 - valores absolutos
                   col.sums <- apply(categoria1, 2, sum) # soma total de cada variavel que
compoe a categoria 1
                   a<- ncol(categoria1) # conta as colunas da categoria 1
                   total1 <- sum(col.sums[1:a]) # soma de todas as variaveis que compoe a
categoria 1 e gera um valor total absoluto para a categoria 1
                   #Categoria 2 -valores absolutos
                   col.sums2 <- apply(categoria2, 2, sum)# soma total de cada variavel que
compoe a categoria 2
                   b<- ncol(categoria2)# conta as colunas da categoria 2
                   total2 <- sum(col.sums2[1:b]) # soma de todas as variaveis que compoe a
categoria 2 e gera um valor total absoluto para a categoria 2
                   #Categoria 3 -valores absolutos
                   col.sums3 <- apply(categoria3, 2, sum)# soma total de cada variavel que
compoe a categoria 3
                   c<- ncol(categoria3)# conta as colunas da categoria 3
                   total3 <- sum(col.sums3[1:c])# soma de todas as variaveis que compoe a
categoria 3 e gera um valor total absoluto para a categoria 3
                   # # # # # # # # Calculando os valores relativos por Categoria # # # # # # #
#
                   totalt <- sum (total1,total2,total3) #soma do total absoluto das três

```

```
categorias
#Categoria 1 - valores relativos
col.sumsrelativo <- round((col.sums/totalt*100), digits=1) # gera o
total relativo para cada variavel que compoe a categoria 1 tendo como
referencia o total abslouto das tres categorias
totalrelativo <- sum(col.sumsrelativo[1:a]) # gera o total relativo para
a soma de todos os elementos que compoe a Categoria 1
#Categoria 2 - valores relativos
col.sumsrelativo2 <- round((col.sums2/totalt*100), digits=1)# gera o
total relativo para cada variavel que compoe a categoria 2 tendo como
referencia o total abslouto das tres categorias
totalrelativo2 <- sum(col.sumsrelativo2)# gera o total relativo para a
soma de todos os elementos que compoe a Categoria 2
#Categoria 3 - valores relativos
col.sumsrelativo3 <- round((col.sums3/totalt*100), digits=1)# gera o
total relativo para cada variavel que compoe a categoria 3 tendo como
referencia o total abslouto das tres categorias
totalrelativo3 <- sum(col.sumsrelativo3)# gera o total relativo para a
soma de todos os elementos que compoe a Categoria 3
# # # # # # # # Adequando os dados para o data.frame # # # # # # # #
cat1 <- names(col.sums)#cria um vetor com os nomes das variaveis que
compe a categoria 1
nomes1 <- c("categoria1", cat1) # cria novo vetor que adiciona o termo
"categoria 1" ao vetor cat1
vetorab1<- as.vector (c( total1, col.sums)) # cria um vetor com totais
absolutos de cada variavel que compoe a categoria 1 e concatena com o total
da categoria
vetorrel1<- as.vector(c(totalrelativo, col.sumsrelativo)) # cria um
vetor com totais relativos de cada variavel que compoe a categoria 1 e
concatena com o total relativo da categoria
cat2 <- names(col.sums2)#cria um vetor com os nomes das variaveis que
compe a categoria 2
nomes2 <- c("categoria2", cat2) # cria novo vetor que adiciona o termo
"categoria 2" ao vetor cat2
vetorab2<- as.vector (c( total2, col.sums2)) # cria um vetor com totais
absolutos de cada variavel que compoe a categoria 2 e concatena com o total
da categoria
vetorrel2<- as.vector(c(totalrelativo2, col.sumsrelativo2))# cria um
vetor com totais relativos de cada variavel que compoe a categoria 2 e
concatena com o total relativo da categoria
cat3 <- names(col.sums3)#cria um vetor com os nomes das variaveis que
compe a categoria 2
class(cat3)
nomes3 <- c("categoria3", cat3) # cria novo vetor que adiciona o termo
"categoria 3" ao vetor cat3
vetorab3<- as.vector (c( total3, col.sums3))# cria um vetor com totais
absolutos de cada variavel que compoe a categoria 3 e concatena com o total
da categoria
vetorrel3<- as.vector(c(totalrelativo3, col.sumsrelativo3))# cria um
vetor com totais relativos de cada variavel que compoe a categoria 3 e
```

```

concatena com o total relativo da categoria
  categoria <- c(nomes1,nomes2,nomes3)# cria vetor com o nomes das três
categorias
  absoluto <- c(vetorab1, vetorab2, vetorab3)# cria vetor com os valores
absolutos das três categorias
  relativo <- c(vetorre11, vetorre12,vetorre13)# cria vetor com os valores
relativos das tres categorias
  # # # # # # # # Incluindo o total # # # # # # # #
  totalrel<- sum(totalrelativo, totalrelativo2, totalrelativo3)# soma o
total relativo das três categorias, evidentemente 100%
  vetorabtotal <- as.vector(totalt) # torna a soma absoluta das categorias
um vetor
  vetorre1total <- as.vector(totalrel) # torna a soma relativa das
categorias um vetor
  Categoria <- c(categoria, "total") # cria vetor que inclui os nomes das
categoria, seus elementos e o nome "total"
  Absoluto <- c(absoluto, vetorabtotal) # cria vetor que inclui o valor
absoluto de cada categoria e de seus componentes e o valor absoluto total.
  Relativo <- c(relativo,vetorre1total)# cria vetor que inclui o valor
relativo de cada categoria e de seus componentes e o valor relativo total.
  print(data.frame(Categoria, Absoluto, Relativo))# gera data.frame com o
  })
  result <- do.call(rbind, ciclo1)# guarda o resultado dos ciclos gerados
# retorna uma lista com os data.frames gerados para cada subgrupo
# # # # # # # # Calculando o índice de Gini # # # # # # # #
  library(ineq)#carrega o pacote ineq para acessar função que permite
calcular o índice de Gini
  saida2 <- vector("list", list.subgrupos)# cria lista em que os data.frames
gerados para os diferentes subgrupos serão guardados
  #iniciando novo ciclo: no ciclo a seguir sera calculado o coeficiente de
Gini para cada categoria de renda e tambem para o total (considerando as
tres categorias) para cada subgrupo do data.frame "renda"
  ciclo2 <- lapply(subgrupos,
    function(k)
  {
    # Nas linhas abaixo sao selecionadas as colunas que compoe as categorias
1, 2 e 3 segundo os criterios definidos pelo usuario. Note que sao gerados
subgrupos do data.frame "renda" - subset(renda, renda[,var]==i.
    categorial <- (subset(renda, renda[,var]==k)[,cat1])
    categoria2<- (subset(renda, renda[,var]==k)[,cat2])
    categoria3<- (subset(renda, renda[,var]==k)[,cat3])
    # # # # # # # # Calculando valores absolutos por Categoria # # # # # # #
#
    # Essas operações precisaram ser retormados, pois objetos gerados no
ciclo anterior não ficam registrados.
    #Categoria 1 - valores absolutos
    col.sums <- apply(categorial, 2, sum) # soma total de cada variavel que
compoe a categoria 1
    a<- ncol(categorial) # conta as colunas da categoria 1
    total1 <- sum(col.sums[1:a]) # soma de todas as variaveis que compoe a
categoria 1 e gera um valor total absoluto para a categoria 1

```

```
#Categoria 2 -valores absolutos
col.sums2 <- apply(categoria2, 2, sum)# soma total de cada variavel que
compoe a categoria 2
b<- ncol(categoria2)# conta as colunas da categoria 2
total2 <- sum(col.sums2[1:b]) # soma de todas as variaveis que compoe a
categoria 2 e gera um valor total absoluto para a categoria 2
#Categoria 3 -valores absolutos
col.sums3 <- apply(categoria3, 2, sum)# soma total de cada variavel que
compoe a categoria 3
c<- ncol(categoria3)# conta as colunas da categoria 3
total3 <- sum(col.sums3[1:c])# soma de todas as variaveis que compoe a
categoria 3 e gera um valor total absoluto para a categoria 3
totalt <- sum (total1,total2,total3) #soma do total absoluto das três
# # # # # # # # Categoria 1 # # # # # # # #
renda.familia1 <- rowSums(categoria1[,1:a]) #resgata "a" que é o número
de colunas da categoria1 e obtem a renda total por familia para a categoria
1
G1 <- ineq(renda.familia1, parameter = NULL, type = c("Gini"), na.rm =
TRUE)#usa a função ineq para calcular o gini para a categoria 1
# # # # # # # # Categoria 2 # # # # # # # #
renda.familia2 <- rowSums(categoria2[,1:b])#resgata "b" que é o número
de colunas da categoria 2 e obtem a renda total por familia para a
categoria 2
G2 <- ineq(renda.familia2, parameter = NULL, type = c("Gini"), na.rm =
TRUE)##usa a função ineq para calcular o gini para a categoria 2
# # # # # # # # Categoria 3 # # # # # # # #
renda.familia3 <- rowSums(categoria3[,1:c])#resgata "c" que é o número
de colunas da categoria 3 e obtem a renda total por familia
G3 <- ineq(renda.familia3, parameter = NULL, type = c("Gini"), na.rm =
TRUE)##usa a função ineq para calcular o gini para a categoria 3
# # # # # # # # Somando as três categorias # # # # # # # #
renda.familia.total <- renda.familia1 + renda.familia2 + renda.familia3
#calcula a renda total por familia considerando as tres categorias
Gtotal <- ineq(renda.familia.total, parameter = NULL, type = c("Gini"),
na.rm = TRUE)#usando a função ineq para calcular o Gini total
Categorias <- c("Categoria1", "Categoria2", "Categoria3", "Total")#cria
vetor com nome de todas as categorias
Gini <- c(G1,G2,G3,Gtotal) #cria vetor com os resultados de Gini
correspondentes a cada categoria e tambem com o valor total
print(data.frame(Categorias,Gini)) # cria e registra no console um
data.frame contendo o Gini para cada Categoria e também o total
})
result2 <- do.call(rbind, ciclo2)# armazena o resultado gerado pelo ciclo
lista3 <- list(result,result2)# cria lista com os data.frames gerados
return(lista3)# retorna lista criada
}
```

Arquivos da Função

[função renda.opera](#)

[Help da função](#)

[Scrip dos exemplos](#)

[dados para testar a função](#)

From:

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

Permanent link:

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

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