

# Oswaldo Santos

Medico Veterinário graduado pela Universidad Nacional de Colombia. Mestrando em Epidemiologia Experimental Aplicada às Zoonoses. FMVZ - USP

## Exercícios

### Exercícios

## Projeto Final

### Plano A - Amostras complexas: cálculo do tamanho amostral

Parte do meu projeto de mestrado tem a ver com a estimativa de parâmetros populacionais de cães e gatos domiciliados (animais de companhia). Como existem restrições econômicas e operacionais, os planos amostrais baseados em sorteios aleatórios simples são inviáveis na maioria dos casos. Portanto, a opção mais viável e eficiente acaba sendo a amostragem por conglomerados em dois estágios, que consiste em selecionar setores censitários do município em questão e nesses setores, sortear domicílios. No entanto, a determinação do tamanho amostral implica em cálculos às vezes trabalhosos que se fossem simplificados facilitariam o trabalho relacionado com pesquisas em dinâmica populacional de animais de companhia. O objetivo deste plano A é construir uma função que permita calcular o tamanho amostral necessário para implementar planos amostrais complexos. Eu já tenho um código (que dá para melhorar bastante) mais ou menos pronto para fazer esse cálculo e gostaria muito que virasse uma função.

### Plano B - Simulação de populações para avaliar o efeito do desenho de planejamentos amostrais complexos

Esta segunda opção também pretendo realizá-la, no entanto, acredito que o prazo de entrega do trabalho final não é muito favorável. A ideia é construir uma distribuição empírica do número de cães por domicílio para depois criar populações por simulação e posteriormente classificá-las de acordo com a variância de cães entre conglomerados (setores censitários) e dentro dos conglomerados (domicílios), para implementar diferentes planos de amostragem complexa e avaliar os efeitos do desenho.

### Comentários das propostas (Leo)

As duas propostas parecem úteis e factíveis. Apesar da tarefa a ser executada estar clara (especialmente na proposta A), senti falta de uma maior clareza de quais cálculos serão feitos, ou seja, como será calculado o tamanho amostral mínimo/ótimo? Quais os dados de entrada (qual informação será fornecida, em qual formato)? E quais as informações de saída (um número mágico de amostras, número mínimo de amostras por setor, um gráfico contendo o tamanho da amostra associada à alguma precisão de estimativa)?

A estratégia de transformar o código em função é pertinente, mas lembre que a função é mais que um código/script. A função generalizará seu código e permitirá que a tarefa seja executada por outras pessoas também, com um problema parecido mas em outros contextos e situações diferentes da sua particular.

## Projeto Final Principal

### Página de Ajuda

cong2

package:nenhum

R Documentation

Cálculo do tamanho amostral e estimativa do total de  $x$ , para um planejamento por conglomerados em

dois estágios.

#### Description:

Calcula o número mínimo de unidades primárias de amostragem (UPA) e unidades secundárias de

amostragem (USA), para estimar o total de uma variável  $x$ , com um nível de confiança e precisão

especificados. A estimativa do total de  $x$  é calculada, junto com os respectivos erro padrão,

nível de confiança, coeficiente de variação e efeito do desenho. Medidas de variabilidade entre

e dentro dos conglomerados também podem ser obtidas.

#### Usage:

```
cong2(upa_usa, upa_x, confianca = 0.95, erro = 0.1, custo = 12, saida = 1)
```

#### Arguments:

upa\_usa: data frame numérico com duas colunas. Primeira coluna com valores únicos em cada fila

para identificar as UPA da população. Segunda coluna com o número de USA na UPA correspondente

(ver Details).

upa\_x: data frame numérico com duas colunas. Segunda coluna com o valor observado da variável x,

em cada USA do estudo piloto. Primeira coluna com a UPA correspondente à USA examinada (ver Details).

confiança: numérico. Nível confiança desejado para calcular o tamanho amostral e para construir

o intervalo de confiança da estimativa. 0.95 (95 %) por default.

erro: numérico. Erro máximo ou diferencia absoluta (expressada como proporção) entre a estimativa

e o parâmetro populacional. 0.1 (10 %) por default.

custo: numérico. Função de custo. 12 por default (ver Details).

saida: numérico. Especifica os cálculos a serem retornados (ver Details).

Details:

Os cálculos são válidos para amostras autoponderadas.

Em upa\_x, as UPA devem ter sido selecionadas com probabilidade proporcional ao tamanho e com

reposição. Nas UPA seleccionadas, deve ter sido sorteado o mesmo número de USA de forma aleatória

e com reposição.

Se upa\_usa ou upa\_x têm número de colunas diferente de 2 ou se a ordem das colunas não é a

especificada, os cálculos não serão executados ou não corresponderão à descrição.

custo é uma função que expressa o quociente do custo que implica a inclusão de uma UPA dividido

pelo custo de realização de cada entrevista. 0 default equivale a dizer que o custo de incluir uma

UPA é 12 vezes o custo de incluir uma USA.

saida: se saida = 1, o tamanho amostral mínimo será retornado, especificando-se o número de UPA

e USA a serem examinadas (para upa\_x de amostra piloto). Se saida = 2, as variancias e desvios

padrões entre e dentro dos conglomerados e o coeficiente de correlação dentro dos conglomerados

serão adicionados à saida = 1 (para upa\_x de amostra piloto). saida = 3 retorna a estimativa de x,

o erro padrão, os limites inferior e superior de confiança, o coeficiente de variação e o efeito

do desenho (para upa\_x de amostra final). saida = 4 retorna a combinação de saida = 2 e saida = 3.

Value:

Tamanho amostral (total de USA)

Numero de UPA a serem examinadas

Numero de USA a serem examinadas em cada UPA

Variancia entre conglomerados

Variancia dentro dos conglomerados

Desvio padrao entre conglomerados

Desvio padrao dentro dos conglomerados

Coeficiente de correlacao dentro dos conglomerados

Estimativa de x

Erro padrao de x

Limite inferior para uma confianca de 95 %

Limite superior para uma confianca de 95 %

Coeficiente de variacao

Efeito do desenho

**Author(s):**

Oswaldo Santos

oswaldo@vps.fmvz.usp.br

**References:**

Levy, P., Lemeshow, S. (1999) Sampling of Populations - Methods and Applications. Third edition.

Wiley.

**See Also:**

'svytotal', 'confint' e 'cv' no pacote survey.

**Examples:**

```
data1 # Arquivo data1 (ver Arquivos).
```

```
data2 # Arquivo data2 (ver Arquivos).
```

```
cong2(data1, data2)
```

## Código da Função

```
cong2 = function(upa_usa, upa_x, confianca = 0.95, erro = 0.1, custo = 12,
saida = 4) {

  upa_usa_x = merge(upa_usa, upa_x, by = 1)

  M = nrow(upa_usa) # Numero de UPA na populacao.

  N = sum(upa_usa[, 2]) # Numero de USA na populacao.

  Ni = upa_usa[, 2] # Numero de USA na UPA i.

  Nip = tapply(upa_usa_x[, 2], upa_usa_x[, 1], unique) # Numero de USA na
UPA i do piloto.

  Nb = mean(Ni) # Numero medio de USA nas UPA.

  nip = tapply(upa_usa_x[, 2], upa_usa_x[, 1], length) # Numero de USA
amostradas na UPA i do

  # piloto.
```

```
nbp = mean(nip) # Numero medio de USA por UPA no piloto

mp = length(unique(upa_usa_x[ , 1])) # Numero de UPA no piloto.

np = nrow(upa_usa_x) # Numero de USA no piloto.

w = N / np # Peso amostral global.

w2 = Nip / nip # Peso amostral do segundo estagio.

xi = tapply(upa_usa_x[ , 3], upa_usa_x[ , 1], sum) # Total de x na amostra
da UPA i do piloto.

Xi = xi * w2 # Total de x na UPA i.

vec = sum((Xi - mean(Xi)) ^ 2) / mp # Variancia entre conglomerados.

vdc = sum((upa_usa_x[ , 3] - mean(upa_usa_x[ , 3])) ^ 2) * w / N #
Variancia dentro dos
#
conglomerados.

dpec =sqrt(vec) # Desvio padrao entre conglomerados.

dpdc = sqrt(vdc) # Desvio padrao dentro dos conglomerados.

d = (((M / (M - 1)) * vec) - (Nb * vdc)) / (((M / (M - 1)) * vec) + (Nb *
(Nb - 1) * vdc)) # Coef.
# de correlacao dentro das UPA.

d = if(d < 0) {- d} else {d = d} # Para evitar valores negativos que
impedem o calculo de nb.

d = if(d == 0) {1e-02} else {d = d} # Para evitar valores negativos que
impedem o calculo de nb.

nb = sqrt(custo * ((1 - d) / d)) # Numero de USA a serem amostradas por
UPA.

X = w * sum(upa_usa_x[ , 3]) # Estimativa de total de x.

z = abs(round(qnorm((1 - confianca) / 2, 0, 1), 2)) # Nivel de confianca.

m = ((z ^ 2) * sum((((N * xi) / nbp) - X) ^ 2)) / ((erro ^ 2) * (X ^ 2) *
(mp - 1)) # Numero de
#
UPA a serem amostradas.

ep = sqrt(sum((((N * xi) / nbp) - X) ^ 2) / (mp*(mp-1)))
```

```
li = X - z * ep # Limite inferior para confianca de 95 %.
ls = X + z * ep # Limite superior para confianca de 95 %.
cv = ep / X # Coeficiente de variacao.
deff = 1 + (nbp - 1) * d # Efeito do desenho.

if(saida == 1) {
  amo = matrix(c(m * nb, m, nb), ncol = 1)

  rownames(amo) = c("Tamanho amostral (total de USA)", "Numero de UPA a
serem examinadas",
  "Numero de USA a serem examinadas em cada UPA")

  colnames(amo) = "Valor"

  return(amo)
}

if(saida == 2) {
  amol = matrix(c(m * nb, m, nb, vec, vdc, dpec, dpdc, d), ncol = 1)

  rownames(amol) = c("Tamanho amostral (total de USA)", "Numero de UPA a
serem examinadas",
  "Numero de USA a serem examinadas em cada UPA", "Variancia entre
conglomerados", "Variancia
dentro dos conglomerados", "Desvio padrao entre conglomerados", "Desvio
padrao dentro dos
conglomerados", "Coeficiente de correlacao dentro dos conglomerados")

  colnames(amol) = "Valor"

  return(amol)
}

if(saida == 3) {
  amo_est = matrix(c(X, ep, li, ls, cv, deff), ncol = 1)

  rownames(amo_est) = c("Estimativa de x", "Erro padrao de x", "Limite
inferior para uma
confianca de 95 %", "Limite superior para uma confianca de 95 %",
"Coeficiente de variacao",
"Efeito do desenho")

  colnames(amo_est) = "Valor"
```

```
    return(amo_est)

}

if(saida == 4) {

    amo_est = matrix(c(m * nb, m, nb, vec, vdc, dpec, dpdc, d, X, ep, li,
ls, cv, deff), ncol = 1)

    rownames(amo_est) = c("Tamanho amostral (total de USA)", "Numero de UPA
a serem examinadas",
    "Numero de USA a serem examinadas em cada UPA", "Variancia entre
conglomerados", "Variancia
    dentro dos conglomerados", "Desvio padrao entre conglomerados", "Desvio
padrao dentro dos
    conglomerados", "Coeficiente de correlacao dentro dos conglomerados",
"Estimativa de x",
    "Erro padrao de x", "Limite inferior para uma confianca de 95 %",
"Limite superior para uma
    confianca de 95 %", "Coeficiente de variacao", "Efeito do desenho")

    colnames(amo_est) = "Valor"

    return(amo_est)

}

}
```

## Arquivos

[data1.csv](#)

[data2.csv](#)

[função\\_cong2.r](#)

## Projeto Final Complementar

## Página de Ajuda

spptr

package:nenhum

R Documentation

Seleção com probabilidade proporcional ao tamanho e com



reposição.

#### Description:

Seleciona x Unidades Primárias de Amostragem (UPA) com probabilidade proporcional ao tamanho

(número de Unidades Secundárias de Amostragem - USA) e com reposição.

#### Usage:

```
spptr(upa_usa, x)
```

#### Arguments:

upa\_usa: data frame numérico com duas colunas. Primeira coluna com valores únicos em cada fila

para identificar as UPA da população. Segunda coluna com o número de USA (tamanho das UPA) na

UPA correspondente (ver Details).

x: numérico. Número de UPA a serem selecionadas.

#### Value:

UPA selecionadas.

#### Author(s):

Oswaldo Santos

oswaldo@vps.fmvz.usp.br

#### References:

Levy, P., Lemeshow, S. (1999) Sampling of Populations - Methods and Applications. Third edition.

Wiley.

#### See Also:

'ppswr' no pacote pps.

Examples:

```
data1 # Arquivo data1 (ver Files).
```

```
spptr(data1, 10)
```

## Código da Função

```
spptr = function (upa_usa, x)
{
  M <- nrow(upa_usa) # Total de UPA.
  cum <- cumsum(upa_usa[ , 2]) # Acumulado de USA em cada UPA.
  N <- cum[M] # Total de USA.
  upa <- numeric(x) # Objeto para armazenar as UPA selecionadas.
  for (i in 1:x) { # Loop para selecionar x UPA.
    a <- runif(1, 0, N) # Sorteio equiprobabilístico de um número entre 0 e N.
    j <- 1
    while (cum[j] < a) { # Seleção do menor dos acumulados dentre os acumulados maiores ao
      # número sorteado.
      j <- j + 1
    }
    upa[i] <- upa_usa[j , 1] # Armazenamento da UPA correspondente ao acumulado selecionado.
  }
  upa = matrix(upa, ncol = 1) # UPA selecionadas
```

```
colnames(upa) = "UPA selecionadas"  
  
return(print(upa, digits = 15))  
  
}
```

## Arquivos

[data1.csv](#)

[função\\_spptr.r](#)

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2013:alunos:trabalho\\_final:oswaldo:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:oswaldo:start) 

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