

Proposta A

Estratificação e amostragem de documentos

O objetivo da função é ajudar pessoas que tem que fazer levantamento e análise de informações presentes em documentos que estão inseridos em grandes bases de dados, mas precisam selecionar apenas alguns documentos de forma aleatória para fazer a análise. Principalmente quando a pesquisa envolve documentação legal ou governamental, pode ser complicado sortear quais os documentos a serem analisados. A função pode ajudar, por exemplo, a estratificar e sortear Termos de Referência de empreendimentos licenciados pelo IBAMA, de forma a obter um número reduzido de documentos que seja o mais representativo possível, abrangendo documentos de anos e regiões diversas, e de tipologias de empreendimentos diferentes.

Função

A função agiria de forma a estratificar e amostrar uma base de dados que contenha informações referentes a documentos que podem ser categorizados. A estratificação e amostragem podem variar de acordo com o desejo do usuário.

Input

O input da função será um data frame, em que as colunas são categorias dos documentos da base de dados (eg. data, tipo, órgão emissor, etc) e cada linha é um documento.

- *categ*: Argumento que permite ao usuário escolher por quais categorias quer estratificar os dados (através da seleção das colunas, ex. dados\$ ano).
- *amostra*: Argumento que permite ao usuário escolher o tamanho da sua amostra final de documentos.
- *espec*: Argumento que possibilita a escolha de quais dados em quais colunas o usuário está interessado (ex. dados\$ ano==2012). Se o usuário não especifica quais dados quer, o default é fazer uma amostragem igualmente distribuída (mesma porcentagem de todos os dados disponíveis).
- *tipo*: Argumento que permite que o usuário escolha se quer amostras igualmente distribuídas (mesma porcentagem de todos os dados disponíveis), se quer uma amostra com números escolhidos por ele de cada categoria específica, ou se irá amostrar os documentos de acordo com a disponibilidade deles (ex. 20% de 1000 documentos totais são de 2013, você quer que sua amostra com 50 documentos possua 20% dos documentos de 2013).
- *relev*: Argumento que possibilita o usuário selecionar quais informações (colunas) ele não quer que apareçam no output.

Output

O output da função será um data frame com os documentos amostrados de acordo com as preferências do usuário.

— *Diogo Melo* 2017/06/05 16:03

Todas as propostas são bacanas! Vou comentar uma de cada vez. A primeira proposta pode ser muito útil pra uma série de coisas, principalmente se vc deixar a função bem geral. Tente fazer uma função que receba qualquer data.frame com colunas agrupadoras e que retorne uma amostra (com ou sem reposição) para qualquer conjunto de colunas que vc passe como argumento.

Pense também como levar em conta a quantidade de linhas em cada categoria (se eu tiver 100 linhas de um tipo e só 10 do outro, faz sentido que amostra respeite essa proporção).

Outro bonus legal seria pensar em como permitir que as categorias sejam aninhadas, caso eu tenha duas categorias que sejam de alguma forma hierárquicas.

No geral, ótima proposta, espaço pra fazer uma função realmente útil, e com complicações possíveis que deixam a coisa interessante.

Oi Diogo,

Obrigada pelos comentários e dicas. Acho que vou fazer a proposta A, pq acredito que ela vai ser mais útil para mim. Quando você diz sobre categorias aninhadas, você quer dizer como subcategorias? Por exemplo, uma coluna com ano e uma coluna com os meses?

Raquel

Proposta B

Mapeamento de ocorrências no campus

Devido à alta incidência de ocorrências policiais dentro do campus da USP, alguns frequentadores do campus se sentiriam mais seguros se pudessem saber a densidade de ocorrências a cada hora do dia em cada instituto. Pensando nisso, a ideia da função é mapear as ocorrências policiais dentro do campus no último semestre de acordo com coordenadas geográficas do tipo X e Y. Os institutos terão uma área de influência e serão representados por pontos no mapa do campus. De acordo com a densidade de ocorrências na área de influência de cada ponto, o ponto aparece no mapa com uma cor. Os mapas de ocorrência também se sobrepõem por horas do dia, gerando um GIF que mostra quais os pontos mais ou menos seguros no campus de acordo com a hora. A função teria como

objetivo auxiliar os frequentadores do campus que dependem do transporte público ou se deslocam a pé, e possibilitar que a guarda universitária se desloque para os pontos mais violentos de acordo com a hora. Pode ser utilizada também para outros tipos de densidade de ocorrências, como a de espécies.

Função

A função iria gerar um mapa único ou GIF, com pontos distribuídos pelo mapa mudando de cor a cada hora. A cor dos pontos mudaria de acordo com a densidade de ocorrências na área de influência do ponto de acordo com coordenadas X e Y.

Input

O input será um data frame com: as coordenadas geográficas, do tipo X e Y, ligadas as ocorrências; a hora de cada ocorrência; as coordenadas de cada ponto (no exemplo seriam as coordenadas dos pontos de cada instituto); e com as coordenadas da área de influência de cada ponto. Os pontos seriam plotados, e conforme o cruzamento da área de influência do ponto e das coordenadas da ocorrência, a ocorrência seria atribuída a um ponto (ex. instituto). Seria gerado um mapa por hora com o ponto de uma cor de acordo com a quantidade de ocorrências por ponto/hora. Por fim, os mapas podem ser sobrepostos gerando um GIF que muda a cor dos pontos de acordo com as ocorrências de hora em hora (24 mapas em seguida).

- *mapa*: Argumento que permite ao usuário escolher se quer que seja gerado um GIF (*gif*), ou se ele está interessado apenas em um mapa juntando todas as ocorrências independente da hora (*total*), ou um mapa de uma hora específica.

Adicional se possível: um argumento para o usuário inserir uma imagem (previamente carregada no R) de um mapa onde os pontos poderão ser plotados (ex. um mapa do campus).

Output

Mapa único ou GIF com os pontos com cores diferentes de acordo com a densidade de ocorrências por hora e localidade.

— [Diogo Melo](#) 2017/06/05 16:17

Essa função é bacana tb, mas deveria ser mais geral. Ela poderia receber um mapa qualquer, um conjunto de “regiões” (os institutos, ou pontos centrais de uma area), e um conjunto de ocorrencias. Dai pra frente fica tudo fácil.

Função bacana tb, e existem pacotes que permitem vc carregar mapas do google maps, por exemplo. Ai vc só precisa receber uma coordenada central do mapa. O ideal é a função ter tanto a opção de usar o google maps quanto receber um shape-file do mapa.

Acho que seria minha opção de função pra fazer nesse momento. É divertida e útil.

Proposta C

Cálculo de quantidades para as refeições no bandeijão

Os restaurantes universitários da USP (mais conhecidos como bandeijões) recebem diariamente centenas de alunos no almoço e na janta. Para que as refeições sejam feitas deve-se fazer o cálculo das quantidades necessárias de cada alimento para preparar cada prato. A função então seria responsável por calcular as quantidades necessárias de cada alimento para 5 pratos pré-estabelecidos de cada (5 pratos de carne, 5 opções vegetarianas, 5 complementos, 5 saladas e 5 sobremesas) para os bandeijões Centra, da Física e da Química. As quantidades para cada prato variam de acordo com a refeição (almoço ou jantar), com o bandeijão, com o período letivo (férias ou não) e se o prato do dia é estrogonofe (aumento significativo de visitantes). O cálculo de pessoas por bandeijão é baseado nas informações do [SAS](#). As informações referentes à quais ingredientes cada prato leva serão retiradas [daqui](#).

Função

Cálculo de quantidades de alimentos necessários para uma refeição dentre locais pré-determinados e com quantidade de visitantes pré-estabelecida. Os pratos podem ser escolhidos dentre 5 opções.

Input

Os inputs da função serão apenas os argumentos dados pelo usuário.

- *band*: argumento que seleciona entre os 3 bandeijões possíveis (*fisica*, *quimica* ou *central*).
- *ref*: argumento que determina a refeição (*almoco* ou *jantar*).
- *periodo*: o default será *letivo*, mas aceita a entrada *ferias*.
- *pratoc*: determina qual o prato de carne dentre 5 opções.
- *prатов*: determina qual o prato vegetariano dentre 5 opções.
- *acompanh*: determina qual o acompanhamento dentre 5 opções.
- *salada*: determina qual a salada dentre 5 opções.
- *sobre*: determina qual a sobremesa dentre 5 opções.

Output

O output da função será um data frame discriminando para quantos visitantes foi calculada a refeição, os ingredientes usados para cada prato e as quantidades necessárias de cada ingrediente.


— *Diogo Melo* 2017/06/05 16:17

Bonitinha, mas conceitualmente simples. Pode ser um plano C, mas é menos interessante que as outras.

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:raquel.monteiro.silva:propostas 

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