

Minhas propostas

Proposta A

A função irá tratar variáveis quantitativas coletadas ao longo do tempo, em taxas de amostragem diferentes e analisar a existência de correlação entre esses dois dados, sem o efeito do tempo, retornando gráficos exploratórios. Em maior detalhe, a função irá:

- igualar a taxa de amostragem de dois dados (x e y). O usuário poderá escolher entre fazer a soma ou a média dos dados coletados mais frequentemente.
- criar dois modelos lineares (lm_x e lm_y) descrevendo a variação de cada uma das duas variáveis em função do tempo. A função irá comparar modelos preliminares antes de escolher os modelos lm_x e lm_y finais. O usuário poderá escolher dentre os seguintes modelos a serem comparados: polinômio de primeiro grau, polinômio de segundo grau, polinômio de terceiro grau, função senoide. Os modelos finais serão escolhidos pelo valor de p e pela soma dos desvios quadráticos residuais.
- utilizar os resíduos desses dois modelos finais para criar um terceiro modelo linear (lm_3) descrevendo a variação da variável y em função da x
- retornar sumário do terceiro modelo linear lm_3
- retornar uma tabela que contenha os dados já tratados, para ser utilizada pelo usuário. Haverá a opção de adicionar os dados tratados em uma tabela já existente.
- retornar dois gráficos com a variação de x e y em relação ao tempo, para os primeiros dez dias de dados. O usuário terá a opção de sobrepor esses dois gráficos.
- plot $y \sim x$ dos dados brutos
- plot $y \sim x$ dos dados descontando o efeito do tempo (resíduos dos modelos lineares (lm_x e lm_y) da variação de cada variável (x e y) em função do tempo)

Objetos e argumentos de entrada da função:

x - vetor numérico quantitativo

$x.horario$ - vetor contendo os horários de amostragem da variável x

y - vetor numérico quantitativo

$y.horario$ - vetor contendo os horários de amostragem da variável y

$resume.por = "sum", "mean"$ - argumento que define o método a ser utilizado para igualar as taxas de amostragem (respectivamente: soma ou média dos dados coletados mais frequentemente).

$modelos = "p1", "p2", "p3", "fs"$ - argumento para selecionar os modelos preliminares a serem comparados para se obter os modelos finais da variação de x e y em relação ao tempo (respectivamente: polinômio de primeiro grau ($y = a + bx$), polinômio de segundo grau ($y = a + bx + cx^2$), polinômio de terceiro grau ($y = a + bx + cx^2 + dx^3$), função senoide ($y = \sin(x)$))

$tabela = ""$ - argumento que determina como retornar os dados tratados. Se não especificada, a função retornará uma tabela nova. Se o usuário especificar um data frame, os dados serão adicionados como colunas novas no data frame já existente.

sobrep - argumento lógico. Se TRUE, serão sobrepostos os gráficos de linha da variação de x e y em relação ao tempo

xlab="x" - legenda a ser colocada no eixo da variável x nos gráficos

ylab="y" - legenda a ser colocada no eixo da variável y nos gráficos

Objeto de saída:

lista com o sumário do modelo linear de y em função de x e tabela contendo os dados tratados.

Gráficos gerados:

- dois gráficos com a variação de x e y em relação ao tempo, para os primeiros dez dias de dados
- plot $y \sim x$ dos dados brutos
- plot $y \sim x$ dos dados descontando o efeito do tempo (resíduos dos modelos lineares (lm_x e lm_y) da variação de cada variável (x e y) em função do tempo)

Proposta B

A função irá calcular o melhor caminho de um ponto A até B, com possibilidade de paradas, sendo que os pontos estão dentro de um espaço delimitado e a posição deles neste espaço se altera com o tempo.

Em maior detalhe, a função irá:

- calcular a distância do ponto inicial, no horário da partida, até o ponto final (ou parada, se existir) em todos os horários amostrados
- determinar o melhor horário para se chegar no ponto final, levando em consideração:
 - o horário que tenha menor distância
 - o tempo gasto para chegar, de acordo com uma velocidade determinada pelo usuário
 - o tempo de espera entre a chegada e o aparecimento do ponto na posição.
- Será escolhido o horário em que a posição do ponto seja mais próxima do ponto de partida, que seja alcançado com a velocidade pretendida e cujo tempo de espera para o ponto aparecer na posição seja o menor possível.
 - No caso de existirem paradas, o próximo trecho será calculado utilizando a posição final do trecho anterior e o novo horário de partida será o horário em que o ponto apareceu nessa posição.
 - A rota será construída de trecho em trecho, até o ponto final.

Objetos e argumentos de entrada da função:

posx - data frame onde a primeira coluna contém os nomes dos pontos e as colunas seguintes, a posição dos pontos no eixo x, em cada horário amostrado, com taxa de amostragem constante.

posy - data frame onde a primeira coluna contém os nomes dos pontos e as colunas seguintes, a posição dos pontos no eixo y, em cada horário amostrado, com taxa de amostragem constante.

p.partida - caracter correspondente ao nome do ponto de partida

ordem.fixa - argumento lógico. Se FALSE, a ordem dos pontos de parada é escolhida de forma a otimizar o tempo do deslocamento

p.parada - lista de caracteres correspondendo aos nomes dos pontos de parada. Se ordem.fixa=TRUE, colocar os pontos na ordem desejada.

p.destino - caracter correspondente ao nome do ponto de destino

t.partida - valor numérico que indica a coluna de dados correspondente ao horário de partida

vel - valor numérico indicando a velocidade do deslocamento, no formato: unidade de x e y/taxa de amostragem

Objeto de saída: data frame com os pontos determinados pelo usuário, posição x e y de cada ponto, o horário de chegada em cada parada e ponto final, tempo de espera até o ponto aparecer na posição esperada.

Gráfico gerado: plot xy ilustrando a posição dos pontos no espaço. Conterá os pontos determinados pelo usuário, com nome e horário, e uma linha indicando o caminho do ponto inicial até o ponto final

Comentários Danilo (gruingas@gmail.com)

Milene, ambas as suas propostas são interessantes, mas o plano B é MUITO mais legal. O plano A é, no final das contas, uma função que automatiza um tipo de análise. É relativamente comum as pessoas fazerem esse tipo de função, mas é uma coisa pouco desafiadora e de aplicação restrita, afinal, quem mais vai fazer exatamente este mesmo tipo de análise?

Já seu plano B é super desafiador e interessante, a questão é: você sabe por onde começar a escrever esse tipo de algoritmo? Se a resposta for positiva, é só prosseguir. Se for negativa, você pode prosseguir com o plano A ou tentar montar uma versão menos complicada do plano B. Mas, se você prosseguir com o plano B, explique melhor no seu help como tem que ser esses data.frames que a função recebe, porque não ficou muito claro pra mim o que tem nesses data.frames além das coordenadas x e y.

Olá Milene, Sobre o plano A, concordo com o Danilo. Tome cuidado para não ser algo muito específico ao seu caso e à sua estrutura de dados. Isto tem muito mais cara de script do que de função. Como função acaba tendo uma aplicação restrita demais. Sobre o plano B, é mesmo **muito** mais legal! O desafio de implementar o algoritmo é algo bastante interessante e que eu recomendo que você siga em frente.

— Sara

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:milene.jannetti:minhas_propostas 

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