

Ramon Vitor Cortez de Godoy



Doutorando na Faculdade de Ciências Farmacêuticas da USP

O título do meu projeto é Efeito da Colostroterapia no Estabelecimento da Microbiota de Recém-Nascidos Prematuros.

[Lista de Exercícios](#)

TRABALHO FINAL

Proposta A: Escolha de Universidade na cidade de São Paulo

Descrição: Todos os anos, diversos jovens que terminam o ensino médio ficam na dúvida de qual faculdade escolher. A cidade de São Paulo, sendo a maior cidade do Brasil e também abarcando uma grande oferta de diversas faculdades públicas e particulares, se torna um dos principais locais de procura, tanto para jovens que residem na cidade, mas também de pessoas de todo o país. Neste contexto, a busca por qual universidade cursar não é algo simples, dentre tantas opções. Minha função tem o objetivo de auxiliar jovens na procura de qual universidade cursar, de acordo com parâmetros de busca que retornem quais faculdades possuem o curso de interesse e que também atendam demais parâmetros na avaliação das universidades que sejam importantes nessa escolha.

Planejamento da função

Entrada: universidades.sp (categoria.univ, cursos.pret, periodo, aval.mec, mensalidade)

categoria.univ = categoria da faculdade escolhida pelo usuário (“pública” ou “particular”) - (classe: character);

cursos.pret = curso que o usuário pretende cursar (classe: character);

periodo = período que o usuário pretende cursar - “matutino”, “vespertino”, “integral” ou “noturno”

(classe: character);

`aval.mec` = nota que o curso recebeu na última avaliação feita pelo Ministério da Educação (classe: numeric)

`mensalidade` = valor médio que o usuário pretende pagar no curso (no caso de escolher cursar uma universidade particular) - (classe: numeric).

Verificando os argumentos da função:

`categoria.univ` é igual a “pública” ou “particular”? Se não, retornar mensagem: “A categoria da universidade deve ser pública ou particular”;

`curros.pret` consta no banco de dados? Se não, retorna a mensagem: “Curso não cadastrado no banco de dados”;

`periodo` é igual a “matutino”, “vespertino”, “integral” ou “noturno”? Se não, retornar mensagem: “O período deve ser igual a matutino, vespertino, integral ou noturno”;

`aval.mec` é um número inteiro com um único argumento maior do que zero e menor ou igual a 5? Se não, retornar: “aval.mec deve conter apenas um número inteiro maior do que 0 e menor ou igual a 5”;

`mensalidade` é um número inteiro maior do que zero? Se não, retornar mensagem: “mensalidade deve ser um número inteiro e >0”.

Pseudo-código:

1- Criar um data frame chamado `universidade`, que irá compor a saída da função;

2- Ler os bancos de dados “`Cursos_SP.xls`” e o banco de dados “`Notas_MEC.xls`”;

3- Receber e verificar os argumentos de entrada do usuário;

4- De acordo com os dados recebidos nos argumentos `categoria.univ`, `curros.pret`, `periodo` e `mensalidade`, realizar uma busca em “`Cursos.xls`” para verificar dados compatíveis com a exigência dos argumentos;

5 - De acordo com o dado recebido no argumento `aval.mec`, realizar uma busca em “`Notas_MEC.xls`” para verificar dados compatíveis com a exigência dos argumentos;

6 - Caso existam universidades que correspondam a todos os argumentos pedidos, retornar o data frame `universidade`, com a lista das universidades selecionadas e as informações sobre as mesmas;

7 - Caso não existam universidades que correspondam a todos os argumentos pedidos, retornar uma mensagem: “Não foi encontrada nenhuma universidade de acordo com os parâmetros de busca”.

Saída:

- Um data frame com as universidades filtradas pela função.

Proposta B: Economizando para a faculdade

Descrição: O final da adolescência e o início da vida adulta são períodos de muitas mudanças na vida dos jovens, onde muitas decisões precisam ser feitas em um curto espaço de tempo. Um dos principais acontecimentos dessa fase é a escolha da graduação, onde o jovem precisa escolher o curso que pretende cursar, além da universidade e da cidade mais apropriada às suas necessidades. Neste contexto, muitas pessoas não se preparam financeiramente para este período e, diante do panorama atual da educação, a chance de ingressar em uma faculdade pública está cada vez mais distante (ou demanda muitos anos em um cursinho). Minha função pretende auxiliar jovens a economizar previamente para o período da graduação e/ou para o período do cursinho preparatório a partir de alguns parâmetros de busca, onde o usuário poderá escolher o tipo de universidade que pretende cursar (pública ou particular), o curso pretendido e o tempo que tem para juntar este valor.

Planejamento da função

Entrada: `investi.facul (categoria.univ, curso.pret, tempo)`

`categoria.univ` = categoria da faculdade escolhida pelo usuário (“pública” ou “particular”) - (classe: character);

`cursos.pret` = curso que o usuário pretende cursar (classe: character);

`tempo` = tempo em meses até o início da faculdade (classe: integer, >0);

Verificando os argumentos da função:

`categoria.univ` é igual a “pública” ou “particular”? Se não, retornar mensagem: “A categoria da universidade deve ser pública ou particular”;

`cursos.pret` consta no banco de dados? Se não, retorna a mensagem: “Curso não cadastrado no banco de dados”;

`tempo` é um número inteiro e >0? Se não, escrever: “tempo precisa ser um número inteiro e >0”;

Pseudo-código:

1- Ler os bancos de dados “Cursos.xls”, “Valor_Grad.xls” e “Valor_Cursinho.xls”;

2- Receber e verificar os argumentos de entrada do usuário;

3- De acordo com os dados recebidos nos argumentos `categoria.univ` e `cursos.pret`, realizar uma busca em “Cursos.xls” para verificar dados compatíveis com a exigência dos argumentos;

4 - Se `categoria.univ` = “particular”, fazer uma busca no banco de dados “Valor_Grad.xls”. O valor retornado ficará armazenado no objeto `valor.grad` = valor total do curso (valor mensal x tempo do curso);

5 - Se `tempo` >0, fazer `valor.grad / tempo` e retornar o valor para o usuário;

6 - Se `categoria.univ` = “pública”, fazer uma busca no banco de dados “Valor_Cursinho.xls”. O valor retornado ficará armazenado no objeto `valor.cursinho` = valor anual médio do cursinho (valor anual da lista de cursinhos cadastrados / total de cursinhos cadastrados);

7 - Se `tempo` >0, fazer `valor.cursinho / tempo` e retornar o valor para o usuário;

Saída:

- O valor que o usuário terá que guardar mensalmente, de acordo com a busca realizada.

Oi Ramon, achei suas duas ideias legais e recomendo seguir com o plano A mesmo.

Sobre o plano A, o grande limitador parece ser a disponibilidade dos dados. Mas se vc já tiver um conjunto de dados com algumas universidades, já dá pra continuar. Tenho uma sugestão de mudança, entretanto. Ao invés de só filtrar as universidades, eu sugiro fazer um ranking das universidades que mais se ajustam aos critérios e das que menos se ajustam. Os critérios do ranking podem ser arbitrários, mas tem que ser explicados no help, ou podem ser determinados pelo usuário (seria super legal, mas fica apenas como sugestão). Seria legal também se todos os critérios pudessem ser informados como vetores, aí a função poderia computar o ranking das universidades para mais de um perfil de estudante.

Sobre o plano B, minha sugestão principal seria incluir um argumento de rendimento do dinheiro que vai ser guardado. Afinal, se o jovem deixar o dinheiro em algum tipo de investimento e tiver alguns anos para guardar o dinheiro, isso pode fazer diferença.

Danilo G Muniz

Olá Danilo, muito obrigado pelas sugestões. Vou seguir com o plano A mesmo. Gostei da ideia de fazer um ranking das universidades de acordo com os critérios de busca, até porque o jovem que busca uma universidade gostaria de saber quais atendem melhor as suas necessidades. Só uma coisa, qual o melhor meio para tirar dúvidas sobre a função? Mando mensagem por aqui mesmo, no fórum ou entro em contato por e-mail?

Obrigado

Ramon Cortez

Links para o trabalho final:

Para o trabalho final, segui com a proposta A, como sugerido. No caso das universidades particulares, adicionei um ranking que mostra as universidades do menor para o maior valor de mensalidade, de acordo com a busca realizada pelo usuário.

- Link para a página da minha função: [universidades.sp](#)
- Link para a página da ajuda da função: [help universidades.sp](#)
- Banco de Dados Cursos_SP: [cursos_sp.xlsx](#)
- Banco de Dados Notas_MEC: [notas_mec.xlsx](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:ramon_vcg89:start 

Last update: **2020/09/23 17:15**