



Mestranda em Fisiologia Geral, aluna do Laboratório de Cognição do IB-USP. Pesquisa a organização temporal coletiva e individual de colônias de formigas cortadeiras.

[Exercícios](#)

# TRABALHO FINAL: PROPOSTAS

## Plano A

### BandejaR

#### Contextualização:

O tempo tem se tornado uma moeda cada vez mais importante na nossa sociedade. No contexto da universidade, é comum que alunos de graduação e pós conciliem aulas, atividades de extensão e de laboratório, bem como hobbies e suas vidas pessoais. É comum termos aulas, palestras, monitorias, reuniões e até mesmo experimentos que cubram parte do horário de almoço. Justamente por isso, nem sempre o estudante sabe se será possível almoçar no restaurante universitário, já que o tempo gasto nessa atividade pode depender de vários fatores, como o cardápio, a distância do bandejão, o clima e até mesmo o tempo de espera pelo ônibus circular. O objetivo dessa função, aqui chamada de BandejaR é, portanto, fornecer ao estudante uma estimativa melhor do tempo gasto para almoçar em cada um dos quatro bandejões da cidade universitária da USP.

#### Planejamento da função

A função bandejaR terá como base de dados um dataframe com o tempo estimado a pé de cada instituição de ensino superior até os quatro bandejões da cidade universitária. Não será considerada a ida ao bandejão através de ônibus circulares, uma vez que é sabido que eles são completamente imprevisíveis. Obs: Talvez, ela inclua também o tempo de carro, permitindo ao usuário que ele selecione um novo parâmetro "carro" que seria uma entrada de True or False e retornaria o tempo considerando esses dois meios de transporte. Uma vez que estamos considerando o tempo e a fila de cada bandejão, a função também já incluirá uma estimativa de fluxo de usuários, considerando o tamanho de cada bandejão, e uma estimativa categorizada de tamanho de fila para cada unidade. Outro parâmetro importante para essa função considera os cardápios de cada bandejão. Uma vez que é muito difícil considerar toda a variedade de opções que os bandejões podem oferecer, o parâmetro cardápio (subdividido em cada um dos bandejões) deverá categorizar o cardápio "global" de cada bandejão como "Bom", "Médio" e "Ruim". Nesse contexto, o usuário deverá considerar a opção não-PVT, uma vez que a maior parte das pessoas não é vegetariana/vegana e, portanto, a maior parte da fila deve ser de indivíduos que comem carne. Embora a classificação do cardápio seja subjetiva, acredito que grande parte dos usuários devem considerar mais ou menos as mesmas coisas como "Bom", "Médio" e "Ruim", dando à função uma estimativa consideravelmente acurada. A caracterização do cardápio, para essa função, implica na quantidade de fila que haverá naquele bandejão. Assim, um cardápio "Bom" terá uma fila maior, um cardápio "Médio" terá uma fila média e um cardápio "Ruim" terá uma fila pequena.

## Entrada:

bandejaR (IES\_origem, chuva = False, card\_pref, card\_fis, card\_quim, card\_cen)  
IES\_origem: instituto de origem dentro da cidade universitária Chuva: pode ser True or False. Em caso de True, a função já considera uma fila menor, uma vez que a maior parte dos bandejões da USP não possuem cobertura para dias de chuva. Card\_pref: avaliação geral do cardápio não-PVT do bandejão da Prefeitura (“Bom”, “Medio” e “Ruim”). Card\_fis: avaliação geral do cardápio não-PVT do bandejão da Física (“Bom”, “Medio” e “Ruim”). Card\_quim: avaliação geral do cardápio não-PVT do bandejão da Química (“Bom”, “Medio” e “Ruim”). Card\_cen: avaliação geral do cardápio não-PVT do bandejão da Central (“Bom”, “Medio” e “Ruim”).

## Verificando os parâmetros

IES\_origem é uma unidade da cidade universitária? Se não, escreve: “IES\_origem precisa ser o código da unidade de ensino: ex: IB e não bio ou biologia. Está chovendo? Se não, escreve: “O default de chuva é false. Se estiver chovendo, você deve inserir “chuva = True” Card\_pref é uma das três categorias? Se não, escreve: Card\_pref deve ser uma das três categorias “Bom” ou “Medio” ou “Ruim”. Card\_fis é uma das três categorias? Se não, escreve: Card\_fis deve ser uma das três categorias “Bom” ou “Medio” ou “Ruim”. Card\_quim é uma das três categorias? Se não, escreve: Card\_quim deve ser uma das três categorias “Bom” ou “Medio” ou “Ruim”. Card\_cen é uma das três categorias? Se não, escreve: Card\_cen deve ser uma das três categorias “Bom” ou “Medio” ou “Ruim”.

## Pseudo-código

- Item de lista ordenada Cria objeto deslocamento com o dataframe com as unidades de ensino nas linhas e os bandejões nas colunas, com o tempo de deslocamento entre elas
- Item de lista ordenada Cria objeto fluxo com o fluxo “pessoas/minuto” que cada bandejão é capaz de servir.
- Item de lista ordenada Cria o objeto fila que é um dataframe como número de pessoas na fila de cada bandejão, considerando uma fila “pequena”, “média” ou “grande” (o cardápio permite inferir o tamanho da fila).
- Item de lista ordenada Se chuva = TRUE, entra em um ciclo em que todas as filas serão consideradas pequenas
- O tempo final de cada bandejão é a estimativa de tempo da fila mais 2x o deslocamento (ida e volta)
- Item de lista ordenada Se chuva = False, entra em um ciclo em que o cardápio de cada bandejão insere o tamanho da fila.
- Item de lista ordenada Com base no tamanho da fila, estima-se o tempo gasto para servir a fila.
- Item de lista ordenada O tempo final de cada bandejão é a estimativa de tempo da fila mais 2x o deslocamento (ida e volta)
- Item de lista ordenada Retorna a estimativa de tempo para cada bandejão

## Saída

Tabela com o instituto de origem, os 4 bandejões e a estimativa de tempo para cada um deles.

## Plano B

### Função ConfAtletas

#### Contextualização:

O esporte universitário tem crescido bastante nos últimos anos e com isso, tem crescido as competições esportivas dentro da universidade. Um grande desafio nesse contexto sempre foi fazer a conferência dos atletas regulares (aqueles com vínculo com sua respectiva atlética e escritos na competição) e dos atletas irregulares (aqueles sem vínculo com sua atlética ou não inscritos na competição). Geralmente, a conferência é feita de forma manual, através de documentos oficiais com foto e buscando os atletas em alguma base de dados, que pode ser digital ou não. A ideia por trás da função ConfAtletas é facilitar esse processo, tornando a conferência automática com base na pré-súmula (documento de pré-inscrição de atletas ao início de cada partida).

#### Planejamento da função

Entrada: ConfAtletas (pre\_sumA, pre\_sumB) pre\_sumA: dataframe com os nomes, numero usp e cpf do atleta da equipe A (mandante) pre\_sumB: dataframe com os nomes, numero usp e cpf do atleta da equipe B (visitante)

#### Verificando os parâmetros:

pre\_sumA é um dataframe? Se não, escreve “pre\_sumA deve ser um dataframe com nome, número usp e cpf dos atletas da equipe A” pre\_sumB é um dataframe? Se não, escreve “pre\_sumB deve ser um dataframe com nome, número usp e cpf dos atletas da equipe B”

#### Pseudo-código:

- Cria um objeto ListAtletas que é um dataframe com nome, atlética, número usp e cpf dos atletas inscritos na competição
- Cria um objeto AtletasA que é a busca com base no cpf de todos os atletas da pre\_sumA em ListAtletas.
- Cria um objeto AtletasB que é a busca com base no cpf de todos os atletas da pre\_sumB em ListAtletas
- Retorna AtletasA e AtletasB

#### Saída:

Retorna AtletasA e AtletasB com: Nome do atleta, nusp, cpf e atletica

### Comentário da Débora Brandt

Mila,

A proposta A é simples, mas criativa, e está bem descrita. Gosto da ideia de incluir a opção de ir de carro. Sobre a imprevisibilidade do circular: que tal se você amostrar o tempo de espera para o circular de uma distribuição aleatória (exponencial). Esse link fala sobre isso e pode ajudar:

<https://jakevdp.github.io/blog/2018/09/13/waiting-time-paradox/> A ideia não seria dar uma estimativa precisa do tempo de espera do circular que a pessoa de fato vai ter naquele dia, mas poderia ser uma “aposta” que a função gera sobre se fale a pena ou não esperar o circular. Seria uma maneira divertida de incluir a imprevisibilidade na função, e de tornar a função um pouco mais complexa. Se essa sugestão não estiver fazendo nenhum sentido para você, não precisa incluir isso.

A proposta B não está muito bem desenvolvida e me parece simples demais. Sugiro seguir com a A. Na proposta B, o objeto ListAtletas precisaria ser diferente para cada competição, o que torna a função muito pouco generalizável.

## TRABALHO FINAL

Código da função: [bandejar.r](#)

Help:[helpbandejar.txt](#)

Anexol: Planilha de deslocamentos a pé [deslocamentos\\_bandejoes\\_andando.csv](#)

Anexoll: Planilha de deslocamentos de carro [deslocamentos\\_bandejoes\\_carro.csv](#)

Anexolll: Estimativa de espera do circular [waitingtimeparadox.txt](#)

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2019:alunos:trabalho\\_final:mila.barbosa:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:mila.barbosa:start) 

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