Universidade de Brasília

DEPARTAMENTO DE ESTATÍSTICA

23 de junho de 2022

Lista 1: Computação eficiente (dados em memória)

Prof. Guilherme Rodrigues

Computação em Estatística para dados e cálculos massivos

Tópicos especiais em Estatística 1

1. **As questões deverão ser respondidas em um único relatório *PDF* ou *html*, produzido usando as funcionalidades do *Rmarkdown* ou outra ferramenta equivalente.**
2. **O aluno poderá consultar materiais relevantes disponíveis na internet, tais como livros, *blogs* e artigos.**
3. **O trabalho é individual. Suspeitas de plágio e compartilhamento de soluções serão tratadas com rigor.**
4. **Os códigos *R* utilizados devem ser disponibilizados na integra, seja no corpo do texto ou como anexo.**
5. **O aluno deverá enviar o trabalho até a data especificada na plataforma Microsoft Teams.**
6. **O trabalho será avaliado considerando o nível de qualidade do relatório, o que inclui a precisão das respostas, a pertinência das soluções encontradas, a formatação adotada, dentre outros aspectos correlatos.**
7. **Escreva seu código com esmero, evitando operações redundantes, visando eficiência computacional, otimizando o uso de memória, comentando os resultados e usando as melhores práticas em programação.**

Nessa lista, utilizamos os pacotes `vroom` e `data.table` para analisar, com rapidez computacional e eficiente uso de memória, dados públicos sobre a vacinação contra a Covid-19.

Questão 1: leitura eficiente de dados

a) Utilizando códigos R, crie uma pasta (chamada *dados*) em seu computador e faça o *download* de todos os arquivos disponíveis no endereço eletrônico a seguir. *https://opendatasus.saude.gov.br/dataset/covid-19-vacinacao/resource/5093679f-12c3-4d6b-b7bd-07694de54173?inner_span=True*

b) Usando a função `p_load` (do pacote `pacman`), carregue o pacote `vroom` (que deve ser usado em toda a Questão 1) e use-o para carregar o primeiro dos arquivos baixados para o R. Descreva brevemente o banco de dados.

Extra: explore essa amostra sem o comando explícito de download.

c) Quantos arquivos totalizam nossos dados? Qual é o tamanho total (em Megabytes) de todos os arquivos?

d) Repita o procedimento do item **b)**, mas, dessa vez, carregue para a memória apenas os casos em que a vacina aplicada foi a Astrazeneca. Para tanto, faça a filtragem usando uma conexão `pipe()`. Observe que a filtragem deve ser feita durante o carregamente, e não após ele.

Quantos megabites deixaram de ser carregados para a memória RAM (ao fazer a filtragem durante a leitura, e não no próprio `R`)?

e) Carregue para o `R` **todos** os arquivos da pasta de uma única vez (usando apenas um comando `R`, sem métodos iterativos).

Questão 2: manipulação de dados

a) Utilizando o pacote `data.table`, repita o procedimento do item **1e)**, agora mantendo, durante a leitura, apenas as 3 primeiras colunas. Use o pacote `geobr` para obter os dados sobre as regiões de saúde do Brasil (procure as funções do `geobr`). Junte (*join*) os dados da base de vacinações com o das regiões de saúde.

Descreva brevemente o que são as regiões (use documentação do governo, não se atenha à documentação do pacote).

b) No *datatable* obtido no item **a)**, crie as variáveis descritas abaixo considerando **apenas os pacientes registrados para a segunda dose**:

1. Quantidade de vacinados por região de saúde;
2. Condicionalmente, a *faixa de vacinação* por região de saúde (alta ou baixa, em relação à mediana da distribuição de vacinações).

Crie uma tabela com as 5 regiões de saúde com menos vacinados em cada *faixa de vacinação*.

Observação: os itens **a)** e **b)** podem ser executados de modo encadeado, usando o operador de pipe.

c) Utilizando o pacote `dtplyr`, repita o procedimento dos itens **a)** e **b)** (lembre-se das funções `mutate`, `group_by`, `summarise`, entre outras). Garanta que você conseguiu criar um objeto com *lazy evaluation* e outro resgatado todos os dados para a memória. Exiba os resultados.

d) Com o pacote `microbenchmark`, comparare o tempo de execução do item **c)** quando se adota as funções do `dtplyr` e do `dplyr`.