

Nome: \_\_\_\_\_

**Você tem total liberdade para resolver os exercícios utilizando a estratégia que quiser: pode utilizar loop for, while ou repeat... ou então programação funcional, como a estratégia de split-apply-combine... pode utilizar qualquer função disponível no R (ou em seus pacotes adicionais)... pode ainda utilizar qualquer tipo de vetorização disponibilizado pelo R. Aliás, que tal tentar resolver os exercícios utilizando o paradigma imperativo com os loop's e também com o paradigma funcional (ou matricial vetorizado). Aproveite a prova como uma oportunidade para consolidar o conhecimento e aprender mais. Fique a vontade para fazer pesquisas na internet e trocar ideia com os colegas (é isso mesmo que está escrito!), mas evite buscar soluções prontas com os colegas, pois assim você irá perder uma ótima oportunidade para estimular seu processo criativo para resolver problemas. E lembre-se a prova é individual**

**1) (8 pontos)** Se listarmos todos os números naturais abaixo de 10 que são múltiplos de 3 ou 5, obtemos 3, 5, 6 e 9. A soma desses múltiplos é 23 e a média é 5,75.

Encontre a média e soma de todos os múltiplos de 3 ou 5 abaixo de 1000. Faça uma função

**2) (8 pontos)** Obter um resumo de determinadas variáveis de um banco de dados. Seguir os passos descritos abaixo.

a. Executar os comandos abaixo:

```
# carregar pacote para leitura dos dados
library(readxl)
# especificar o caminho dos dados de seu computador
dados <- read_excel("C:/aulas/dados_prog/dados.xlsx")
```

Acesse os dados em: <https://gmarcatti.github.io/dados/dados.xlsx>

b. Obter a média e o desvio padrão (funções: `mean` e `sd`) para as variáveis **volume** e **hd** (altura dominante) para cada um dos talhões (variável `chave_talhao`) disponíveis na base de dados. Detalhe importante: o resumo deve ser restrito apenas a última observação (medição) de cada parcela (variável `parc`), isto é, observação de maior idade de cada parcela (medição mais atual).

**3) (8 pontos)** Ajustar um modelo, de volume em função da idade (escolha o modelo de sua preferência), para cada uma das classes `prec_mm` (Precipitações) utilizando um loop de sua preferência. O resultado final deverá ser uma `data.frame` (ou `tibble`) com a quantidade de linhas igual ao número de classes de precipitações e contendo as seguintes colunas: **prec\_mm**, **b0**, **b1**, **R2**. Esta última coluna é o coeficiente de determinação (medida de qualidade ajuste). Faça a plotagem dos ajustes também, pode ser algo semelhante ao apresentado em aula.

**4) (8 pontos)** Para ser número primo, um número deve ser divisível apenas por 2 números: 1 e por ele mesmo. Caso um número seja múltiplo de qualquer outro, ele não é primo. Em outras palavras, se um número é primo, o resto da divisão por qualquer outro número é diferente de zero, com exceção do número 1 e ele mesmo.

Os números primos entre 1 e 100 são: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.

Formule um algoritmo e implemente uma função em código R para retornar os números primos de 1 a n. Sua função deverá apresentar apenas o **n** como argumento, e deverá retornar um vetor com os números primos de 1 a n.

Dicas: (1) Para obter o resto da divisão utilize %%. (2) Tente subdividir o problema em duas partes: 1º crie uma função que identifica se um número é primo ou não. 2º depois utilize essa função em um processo repetitivo para avaliar se os elementos de um vetor de 1:n são primos... os que forem primos devem ser armazenados em um vetor. Provavelmente as duas funções devem ter um loop.

**5) (8 pontos)** Discorra sobre os paradigmas de programação apresentados no curso: imperativo, funcional e matricial/vetorizado. Exemplo: fale um pouco sobre cada um, vantagens, desvantagens, usos, etc.