

Lista Prática 2

Exercício 1 (*swirl*). Usando o pacote *swirl*, acesse o curso “R Programming” e faça as seguintes lições:

- a) 10: *lapply and sapply*
- b) 12: *Looking at Data*
- c) 13: *Simulation*

Adicionalmente, instale o curso “Getting and Cleaning Data” usando a seguinte função `swirl::install_course("Getting and Cleaning Data")` e faça as seguintes lições:

- d) 1: *Manipulating Data with dplyr*
- e) 2: *Grouping and Chaining with dplyr*

Ao instalar o curso e entrar na lição “1: Manipulating Data with dplyr” aparecerá o erro:

```
Error in yaml.load(readLines(con, warn = readLines.warn), error.label = erro  
r.label, :  
(C:/Users/fhnis_r40seqz/AppData/Local/Programs/R/R-4.3.0/library/swirl/Cour  
ses/Getting_and_Cleaning_Data/Manipulating_Data_with_dplyr/lesson.yaml) Scann  
er error: while scanning a tag at line 205, column 9 did not find expected wh  
itespace or line break at line 205, column 19
```

Copie o endereço até a pasta “Manipulating_Data_with_dplyr” (destacado na imagem acima) e substitua o arquivo “lesson.yaml” pelo arquivo disponível em:

<https://fhnishida.netlify.app/project/rec5004/lesson.yaml>

Por fim, use `swirl::install_course("Exploratory Data Analysis")` para instalar o curso “Exploratory Data Analysis” e faça as seguintes lições:

- f) 5: *Base Plotting System*
- g) 9: *GGPlot2 Part2*¹

Exercício 2. A base de dados deste exercício foi obtida do site *Hospital Compare* (<http://hospitalcompare.hhs.gov>) administrado pelo Departamento de Saúde e Serviços Humanos dos EUA. O propósito deste site é prover dados e informação sobre a qualidade dos tratamentos de mais de 4 mil hospitais certificados pelo Medicare. O arquivo que utilizaremos é o *medicare.csv*², que contém informação sobre taxa de mortalidade de 30 dias para alguns problemas de saúde.

Comece carregando a base de dados com o seguinte código e observe a sua estrutura:

¹O 1º exercício irá pedir para fazer um gráfico usando `qplot()` que é uma função pouco utilizada. Você pode dar `skip()` para avançar para a parte com gráficos usando a função `ggplot()`

²Versão resumida da base de dados fornecida no curso da John Hopkins no Coursera

```
1 outcome = read.csv2("https://fhnishida.netlify.app/project/rec5004/medicare.csv")
```

Como exercício, escreva a função `best()` que terá dois argumentos: (I) a sigla de um Estado americano, e (II) o nome do problema de saúde. Essa função, a partir base de dados `outcome`, retorna um vetor de texto com o nome do hospital que possui a melhor (menor) taxa de mortalidade (Death Rate), dadas sigla de Estado e problema de saúde. O nome do hospital está na coluna “Hospital.Name” e os problemas de saúde podem ser “heart attack”, “heart failure” e “pneumonia”. Se houver empate de melhor hospital, o nome do hospital a ser retornado pela função é dado pela ordem alfabética – se “Hospital A” e “Hospital B” estiverem empatados, deve-se retornar “Hospital A”. A função deve ter a seguinte estrutura:

```
1 best = function(estado, problema) {
2   # Filtre a base de dados pela sigla do estado
3
4   # Reordene a coluna taxa de mortalidade da doença selecionada de forma
   # crescente e, em caso de empate, coloque os nomes dos hospitais de forma alfabé
   # tica.
5
6   # Retorne o nome do hospital com a menor taxa de mortalidade
7 }
```

Como referência, seguem alguns exemplos de output da função:

```
1 > best("TX", "heart failure")
2 [1] "FORT DUNCAN MEDICAL CENTER"
3
4 > best("MD", "heart attack")
5 [1] "JOHNS HOPKINS HOSPITAL, THE"
```

Usando a função `best()` escrita, quais são os resultados retornados para os casos:

- a) estado = "SC" e problema = "heart attack"
- b) estado = "NY" e problema = "pneumonia"
- c) estado = "AK" e problema = "pneumonia"

Exercício 3. Carregue bases de dados de GDP para 190 países ranqueados e de informações educacionais a partir do seguinte código:

```
1 gdp = read.csv2("https://fhnishida.netlify.app/project/rec5004/GDP.csv")
2 educ = read.csv2("https://fhnishida.netlify.app/project/rec5004/EDSTATS.csv")
```

- a) Mescle as duas bases de dados a partir do código de país de 3 dígitos. Quantos códigos tiveram correspondência em ambas bases? Organize a base de dados mesclada de forma decrescente no GDP (de modo que USA fica na última linha). Qual é o país na 13^a linha?
- b) Quais são os rankings médios para os grupos de renda “High income: OECD” e “High income: nonOECD”?
- c) Divida os países em 5 grupos de (quase) mesmo tamanho de acordo com seus GDPs, criando a variável “GDP.Group”. Depois, faça uma tabela cruzada entre ela e “Income.Group”. Quantos países estão entre as nações com maior renda (High: OECD/nonOECD) e estão entre o 1^o e 2^o quintis do GDP?

Exercício 4. Considere a base de dados *mtcars* (nativa no R) que contém o consumo de combustível e mais 10 características automobilísticas para 32 carros (modelos 1973-74). Em particular, queremos analisar a relação entre o consumo de combustível (*mpg*, em milhas por galão) e o peso do automóvel (*wt*, em mil libras) pelo seguinte modelo:

$$mpg_i = \alpha + \beta wt_i + \varepsilon_i$$

- a) Assuma $\hat{\alpha} = 36$ e $\hat{\beta} = -4$ e calcule a soma dos quadrados dos resíduos $\sum_{i=1}^N \hat{\varepsilon}_i^2$, tal que um resíduo é dado por: $\hat{\varepsilon}_i = mpg_i - \widehat{mpg}_i$.
- b) Crie uma função `resid_quad(alpha, beta)` que recebe como inputs possíveis valores de $\hat{\alpha}$ e $\hat{\beta}$, e retorna a soma dos quadrados dos resíduos do modelo acima a partir da base de dados *mtcars*.
- c) Considere os seguintes vetores de possíveis valores de $\hat{\alpha}$ e de $\hat{\beta}$:
`alpha_grid = seq(34, 38, length=11)` e `beta_grid = seq(-6, -2, length=11)`
 Crie uma matriz de dimensão 11×11 em que cada linha corresponde a um valor de $\hat{\alpha}$ e cada coluna corresponde a um valor de $\hat{\beta}$. Preencha essa matriz usando a função `resid_quad()` e verifique qual par $(\hat{\alpha}, \hat{\beta})$ minimiza a soma dos quadrados dos resíduos do modelo acima.

Exercício 5. Faça uma amostragem de 100 números sem reposição do vetor de números inteiros `1:100`. Usando estruturas condicionais e de repetição/loop, reordene o vetor de forma decrescente.

- a) Crie um algoritmo usando os seguintes passos:
 1. Compare o número da 1^a posição com o da 2^a e, caso este seja maior, troque a ordem.
 2. Compare o número da 2^a posição com o da 3^a e, caso este seja maior, troque a ordem.
 - ⋮
 99. Compare o número da 99^a posição com o da 100^a e, caso este seja maior, troque a ordem.
- b) Os passos executados no item anterior foram suficientes? Adapte seu algoritmo para que o vetor esteja totalmente ordenado decrescentemente.