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

Adicionalmente, carregue o pacote `swirl` *e instale o curso "Getting and Cleaning Data" usando a seguinte função* `install_course("Getting and Cleaning Data")`[1] *e faça as seguintes lições:*

e) 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" ocorrerá um erro e aparecerá uma mensagem informando um endereço de arquivo que precisamos substitutir:

`C:/.../Getting_and_Cleaning_Data/Manipulating_Data_with_dplyr/lesson.yaml`

Copie o endereço até a pasta do arquivo ("C:/.../Manipulating_Data_with_dplyr/") e cole em uma barra de endereço para acessar a pasta. Então, substitua o arquivo "lesson.yaml" pelo disponibilizado em: `https://fhnishida.github.io/fearp/rec2301/lesson.yaml`

Depois, acesse a lição "1: Manipulating Data with dplyr" no `swirl` *novamente.*

Exercício 2. *A base de dados deste exercício vem 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* `outcome-care_modified.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, olhe suas primeiras 10 linhas e observe a sua estrutura:

```
outcome = read.csv2("https://fhnishida.github.io/fearp/eco1/outcome-care_modified.
    csv")
```

[1] Caso tenha problemas, instalar manualmente: `http://swirlstats.com/scn/getclean.html`
[2] Versão resumida da base de dados fornecida no curso da John Hopkins no Coursera

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:*

```
best = function(estado, problema) {
    # Filtre a base de dados 'outcome' pela sigla do estado

    # Reordene a coluna taxa de mortalidade da doença selecionada de
    # forma crescente e, em caso de empate, coloque os nomes dos
    # hospitais de forma alfabética.

    # Retorne o nome do hospital com a menor taxa de mortalidade
    }
```

Como referência, seguem alguns exemplos de output da função:

```
> best("TX", "heart failure")
[1] "FORT DUNCAN MEDICAL CENTER"

> best("MD", "heart attack")
[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"`

Resposta:

```
best = function(estado, problema) {
    # Filtre a base de dados 'outcome' pela sigla do estado
    bd = outcome %>% filter(State == estado)

    # Reordene a coluna taxa de mortalidade da doença selecionada de
    # forma crescente e, em caso de empate, coloque os nomes dos
    # hospitais de forma alfabética.
    if (problema == "heart attack") {
        bd = bd %>% arrange(DeathRatesHeartAttack, Hospital.Name)
    } else if (problema == "heart failure") {
        bd = bd %>% arrange(DeathRatesHeartFailure, Hospital.Name)
    } else if (problema == "pneumonia") {
        bd = bd %>% arrange(DeathRatesPneumonia, Hospital.Name)
    }

    # Retorne o nome do hospital com a menor taxa de mortalidade
    bd$Hospital.Name[1]
}
```

a) "MUSC MEDICAL CENTER"

b) "MAIMONIDES MEDICAL CENTER"

c) "YUKON KUSKOKWIM DELTA REG HOSPITAL"

□

Exercício 3. *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:*

$$mpg = \alpha + \beta wt + \varepsilon.$$

a) Assuma $\hat{\alpha} = 36$ *e* $\hat{\beta} = -4$ *e calcule a soma dos erros quadráticos* $\hat{\varepsilon}^2$*, tal que o desvio é dado por:* $\hat{\varepsilon} = mpg - \widehat{mpg}$*. Note que mpg e* $\widehat{mpg} = \hat{\alpha} + \hat{\beta} wt$ *são, respectivamente, vetores com os valores observados e preditos/ajustados da variável de consumo de combustível.*

b) Crie uma função `erro_quad(alpha, beta)` *que recebe como inputs possíveis valores de* $\hat{\alpha}$ *e* $\hat{\beta}$*, e retorna a soma dos erros quadráticos do modelo acima a partir da base de dados mtcars.*

c) Assuma 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 possível valor de* α *e cada coluna corresponde a um possível valor de* β*. Preencha essa matriz usando repetições (for/while) e a função* `erro_quad()`*. Depois, verifique qual par* (α, β) *minimiza o erro quadrático do modelo acima. (Dica: use a função* `which(..., arr.ind = TRUE)` *na matriz preenchida para obter os índices de linha e coluna.)*

Resposta:

a) Os erros quadráticos são calculados pela diferença entre os valores ajustados com os valores observados de *mpg*, elevando ao quadrado.

```
mpg_ajustado = (36 + -4 * mtcars$wt)
erros = mtcars$mpg - mpg_ajustado
sum(erros^2) # soma dos erros quadráticos
```

```
[1] 627.7735
```

b) A função segue o mesmo cálculo feito no item (a):

```
erro_quad = function(alpha, beta) {
    mpg_ajustado = alpha + beta * mtcars$wt
    erros = mtcars$mpg - mpg_ajustado
    sum(erros^2)
}

erro_quad(36, -4) # Testando para os valores do item (a)
```

```
[1] 627.7735
```

c)

```
# Definindo os possíveis valores de alpha e beta
alpha_grid = seq(34, 38, length=11)
beta_grid = seq(-6, -2, length=11)

# Criando matriz de zeros com dimensão 11 x 11
matriz = matrix(0, nrow=length(alpha_grid), ncol=length(beta_grid))

# Preenchendo a matriz com soma dos erros quadráticos, dadas todas possíveis
    combinações de valores de alpha e de beta
for (i in 1:length(alpha_grid)) {
    for (j in 1:length(beta_grid)) {
        matriz[i, j] = erro_quad(alpha_grid[i], beta_grid[j])
    }
}

head(matriz) # Visualizando 6 linhas da matriz
min(matriz) # Menor soma de erros quadráticos

# Índices de alpha e de beta que minimizam desvio quadrático
indices = which(matriz == min(matriz), arr.ind = TRUE)
indices
```

```
     row col
[1,]   8   3
```

```
# Atribuindo os índices a objetos
i_alpha = indices[1, "row"]
i_beta = indices[1, "col"]

# Buscando valores de alpha e de beta a partir dos índices acima
alpha_grid[i_alpha]
beta_grid[i_beta]
```

```
[1] 36.8
[1] -5.2
```

□

Exercício 4. *Carregue bases de dados de GDP para 190 países ranqueados e de informações educacionais a partir do seguinte código:*

```
gdp = read.csv2("https://fhnishida.github.io/fearp/eco1/data_GDP.csv")
educ = read.csv2("https://fhnishida.github.io/fearp/eco1/EDSTATS_Country.csv")
```

a) Mescle as 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"?

Resposta:

a) 189 correspondências e 13º país é “St. Kitts and Nevis”

```
library(dplyr)
## JUNTANDO BASES E MANTENDO APENAS CASOS QUE APARECEM EM AMBAS
joined = merge(gdp, educ, by.x="Country", by.y="CountryCode", all=FALSE)
nrow(joined) # núm. de correspondencias entre as duas bases (já que ALL=FALSE)
```

```
[1] 189
```

```
# Ordenando GDP de forma crescente
joined = joined %>% arrange(GDP)
joined[13, 1:4] # 13o país na ordem decrescente (e 4 primeiras colunas)
```

```
  CountryCode Rank          Long.Name.x GDP
           KNA  178 St. Kitts and Nevis 767
```

b) 32.9 e 91.9

```
## AVERAGE RANK
high_OECD = joined %>% filter(Income.Group == "High income: OECD")
high_nonOECD = joined %>% filter(Income.Group == "High income: nonOECD")

mean(high_OECD$Rank)
mean(high_nonOECD$Rank)
```

```
[1] 32.96667
[1] 91.91304
```

```
# OU, de forma mais simples (por ‘group_by’ e ‘summarise’):
library(dplyr)
joined %>%
    group_by(Income.Group) %>%
    summarise(meanRank = mean(Rank))
```

```
  Income.Group          meanRank
  <chr>                    <dbl>
1 High income: nonOECD      91.9
2 High income: OECD         33.0
3 Low income               134.
4 Lower middle income      108.
5 Upper middle income       92.1
```

□