Exercício

Maurício Bueno

04/09/2021

Table of Contents

[Ativar os pacotes 1](#_Toc83633390)

[Definir o diretório de trabalho 1](#_Toc83633391)

[Importar arquivo do Excel para o R 1](#_Toc83633392)

[Informações sobre o banco de dados 2](#_Toc83633393)

[Exercícios 5](#_Toc83633394)

[Exercícios usando Tidyverse 10](#_Toc83633395)

# Ativar os pacotes

Sempre, antes de começar a trabalhar com o R, precisamos ativar os pacotes (bibliotecas) que vamos usar. Neste caso, são as seguintes:

library(tidyverse)

# Definir o diretório de trabalho

Após ativar os pacotes, ainda é preciso definir o diretório de trabalho no R. Fazemos isso usando a função setwd(). Dentro dos parênteses, temos que informar o caminho até o diretório que iremos usar. Por exemplo, no computador que estou usando neste momento, o diretório está em *C:\Users\mauricio\OneDrive\Documentos\R\R\_basico\Intro\_R.* Então, o código para definição do diretório ficaria assim:

setwd("C:/Users/jmhbu/OneDrive/Documentos/R/R\_basico/Intro\_R")

# Importar arquivo do Excel para o R

Nesta atividade, nós vamos um banco de dados público, que contem algumas informações sobre os respondentes e diversos itens de um teste de personalidade. Os itens estão na seguinte ordem: extroversão (E), neuroticismo (N), amabilidade (A), conscienciosidade (C) e Abertura (O). Copie o arquivo *big\_five.xlsx* para o diretório de trabalho (definido no item anterior).

Esse é um banco de dados estrangeiro, está codificado no sistema europeu. Vamos usar a função read\_excel() para importá-lo. Dentro dos parênteses informamos o arquivo (entre aspas e com a extensão).

big\_five <- readxl::read\_excel("big\_five.xlsx")  
  
# ao rodar o comando, o arquivo será importando e salvo como dataframe no R. Você verá um objeto com esse nome, no quadro superior à direita.  
  
# use um comando de visualização dos dados, para ver se o arquivo está ok.   
view(big\_five)

# Informações sobre o banco de dados

O banco de dados utilizado neste exercício (baixado da internet) foram coletados (c. 2012) por meio de um teste de personalidade online interativo. Os itens foram avaliados em uma escala de cinco pontos, em que 1 = Discordo, 3 = Neutro, 5 = Concordo (0 = não respondido). Os itens foram os seguintes:

E1 - I am the life of the party.

E2 - I don’t talk a lot.

E3 - I feel comfortable around people.

E4 - I keep in the background.

E5 - I start conversations.

E6 - I have little to say.

E7 - I talk to a lot of different people at parties.

E8 - I don’t like to draw attention to myself.

E9 - I don’t mind being the center of attention.

E10 - I am quiet around strangers.

N1 - I get stressed out easily.

N2 - I am relaxed most of the time.

N3 - I worry about things.

N4 - I seldom feel blue.

N5 - I am easily disturbed.

N6 - I get upset easily.

N7 - I change my mood a lot.

N8 - I have frequent mood swings.

N9 - I get irritated easily.

N10 - I often feel blue.

A1 - I feel little concern for others.

A2 - I am interested in people.

A3 - I insult people.

A4 - I sympathize with others’ feelings.

A5 - I am not interested in other people’s problems.

A6 - I have a soft heart.

A7 - I am not really interested in others.

A8 - I take time out for others.

A9 - I feel others’ emotions.

A10 - I make people feel at ease.

C1 - I am always prepared.

C2 - I leave my belongings around.

C3 - I pay attention to details.

C4 - I make a mess of things.

C5 - I get chores done right away.

C6 - I often forget to put things back in their proper place.

C7 - I like order.

C8 - I shirk my duties.

C9 - I follow a schedule.

C10 - I am exacting in my work.

O1 - I have a rich vocabulary.

O2 - I have difficulty understanding abstract ideas.

O3 - I have a vivid imagination.

O4 - I am not interested in abstract ideas.

O5 - I have excellent ideas.

O6 - I do not have a good imagination.

O7 - I am quick to understand things.

O8 - I use difficult words.

O9 - I spend time reflecting on things.

O10 - I am full of ideas.

**race** (raça): Foi escolhida pelos participantes em um menu suspenso.

0 = Dados perdidos

1 = Raça Mista

2 = Ártico (Siberiano, Esquimó)

3 = Caucasiano (Europeu)

4 = Caucasiano (Indiano)

5 = Caucasiano (Médio Leste)

6 = Caucasiano (Norte Africano, Outro)

7 = Indígena Australiano

8 = Nativo Americano

9 = Nordeste Asiático (Mongol, Tibetano, Japonês Coreano, etc)

10 = Pacífico (Polinésia, Micronésia, etc.)

11 = Sudeste Asiático (Chinês, Tailandês, Malaio, Filipino, etc.)

12 = Oeste da África, Bosquímanos, Etíope

13 = Outro

**age**: inserida como texto (indivíduos que relataram idade <13 anos não foram registrados)

**engnat**: Resposta a “o inglês é sua língua nativa?”: 0 = dados perdidos, 1 = sim, 2 = não

**gender**: Escolhido em um menu suspenso. 0 = dados perdidos, 1 = Masculino, 2 = Feminino, 3 = Outro

**hand**: “Com que mão você escreve?”. 0 = dados perdidos, 1 = Direita, 2 = Esquerda, 3 = Ambos

**country**: Código ISO do país.

**source**: Como o participante soube do teste. 1 = de outra página no site de teste 2 = do google 3 = do facebook 4 = de qualquer url com “.edu” em seu nome de domínio (por exemplo, xxx.edu, xxx.edu.au), 5 = não informado no arquivo 6 = outra fonte ou HTTP Referer não fornecido.

# Exercícios

1. Visualise o banco de dados com as funções names() e view() para checar se tudo foi importado corretamente.

cópia\_de\_segurança <- big\_five  
# names(big\_five)  
# View(big\_five)

1. Quantas variáveis existem no banco de dados? (igual ao número de colunas)
2. Quantos sujeitos constam no banco de dados? (igual ao número de linhas)
3. Como as variáveis estão codificadas no banco de dados (character, numeric, factor, etc…)? Obs: uma variável numérica pode estar como número inteiro (int) ou como número decimal (dbl).
4. Renomeie as variáveis, traduzindo-as para o português: raça = race, idade = age, ing\_nat = engnat, gênero = gender, mão\_dom = hand, fonte = source, país = country. Salve essa modificação no dataframe *big\_five*.

big\_five <- rename(big\_five,c(raça = race, idade = age, ing\_nat = engnat, gênero = gender, mão\_dom = hand, fonte = source, país = country))

1. Crie uma coluna **id** para identificar os sujeitos com um “s” seguido do número do sujeito. Por exemplo, s1, s2, s3… sn. Posicione essa coluna como a primeira do banco de dados.

# para não ter que reordenar as variáveis depois que id estiver no final do dataframe, pode-se criar um vetor com os ids e depois fazer um cbind com o big\_five.  
  
id <- paste("s",1:nrow(big\_five),sep = "")  
big\_five <- cbind(id,big\_five)  
  
# essa função do dplyr reloca colunas para o início do dataframe  
# big\_five %>% relocate(id) %>% names()

1. As variáveis raça, ing\_nat, gênero e mão\_dom estão codificadas como número inteiro (integer, abreviado como int). Transforme-as em fatores.

# str(big\_five)  
big\_five$raça <- as.factor(big\_five$raça)  
big\_five$ing\_nat <- as.factor(big\_five$ing\_nat)  
big\_five$gênero <- as.factor(big\_five$gênero)  
big\_five$mão\_dom <- as.factor(big\_five$mão\_dom)  
big\_five$"país" <- as.factor(big\_five$"país")

1. Agora é necessário atribuir os níveis de cada variável que foi transformada em fator. Os níveis para cada variável estão descritos no início deste documento. Lembrete: os níveis têm que ser atribuídos na mesma ordem dos valores. Por exemplo, na variável *gênero* o número 1 foi atribuído a *Masculino*, 2 a *Feminino* e 3 a *Outros.* Além disso, o valor 0 foi atribuído a respostas faltantes. Essas precisam ser transformadas em NA no R. Portanto, temos 0, 1, 2 e 3 no banco de dados, e os nomes desses níveis devem aparecer nessa ordem na linha de comando dessa variável.

levels(big\_five$raça) <- c(NA,  
 "Raça Mista",  
 "Ártico (Siberiano, Esquimó)",  
 "Caucasiano (Europeu)",  
 "Caucasiano (Indiano)",  
 "Caucasiano (Médio Leste)",  
 "Caucasiano (Norte Africano, Outro)",  
 "Indígena Australiano",  
 "Nativo Americano",  
 "Nordeste Asiático (Mongol, Tibetano, Japonês Coreano, etc)",  
 "Pacífico (Polinésia, Micronésia, etc.)",  
 "Sudeste Asiático (Chinês, Tailandês, Malaio, Filipino, etc.)",  
 "Oeste da África, Bosquímanos, Etíope",  
 "Outro"  
 )

levels(big\_five$ing\_nat) <- c(NA,"Sim","Não")

levels(big\_five$gênero) <- c(NA,"Masculino","Feminino","Outro")

1. Rode a função count(big\_five,idade) para contar quantas pessoas de cada idade há no banco de dados. Observe se há algo estranho.
2. Selecione apenas os participantes com idades até 80 anos (inclusive) e provenientes dos Estados Unidos (US). Salve esses participantes em um novo dataframe chamado **bf80us**.

bf80us <- big\_five[c(big\_five$idade <= 80 & big\_five$país == "US"), ]

1. Quantos sujeitos compõem esse novo banco de dados?
2. Exporte o scritp deste exercício no formato .rds usando a função saveRDS(bf80us,"bf80us.rds") e envie pelo google classroom para cumprir a atividade assíncrona dessa semana.

saveRDS(bf80us,"bf80us.rds")

1. Voltando a trabalhar com o arquivo big\_five, criar uma nova variável chamada faixa\_etária com as seguintes faixas: até 20 anos (1), de 21 a 30 (2), 31 a 40 (3), 41 a 50 (4), 51 a 60 (5), 61 a 70 (6), 71 a 80 (7) e atribua NA para valores superiores a 80. Salve essa nova variável no dataframe big\_five.

big\_five$faixa\_etária <- ifelse(big\_five$idade <= 20, 1,  
 ifelse(big\_five$idade > 20 & big\_five$idade <= 30,2,  
 ifelse(big\_five$idade > 30 & big\_five$idade <= 40,3,  
 ifelse(big\_five$idade > 40 & big\_five$idade <= 50,4,  
 ifelse(big\_five$idade > 50 & big\_five$idade <= 60,5,  
 ifelse(big\_five$idade > 60 & big\_five$idade <= 70,6,  
 ifelse(big\_five$idade > 70 & big\_five$idade <= 80,7,NA)))))))  
str(big\_five)

## 'data.frame': 19719 obs. of 59 variables:  
## $ id : chr "s1" "s2" "s3" "s4" ...  
## $ raça : Factor w/ 13 levels "Raça Mista","Ártico (Siberiano, Esquimó)",..: 3 13 1 3 11 13 5 4 5 3 ...  
## $ idade : num 53 46 14 19 25 31 20 23 39 18 ...  
## $ ing\_nat : Factor w/ 2 levels "Sim","Não": 1 1 2 2 2 1 1 2 1 1 ...  
## $ gênero : Factor w/ 3 levels "Masculino","Feminino",..: 1 2 2 2 2 2 2 1 2 2 ...  
## $ mão\_dom : Factor w/ 4 levels "0","1","2","3": 2 2 2 2 2 2 2 2 4 2 ...  
## $ fonte : num 1 1 1 1 2 2 5 2 4 5 ...  
## $ país : Factor w/ 159 levels "(nu","A1","A2",..: 150 150 121 128 150 150 150 72 150 150 ...  
## $ E1 : num 4 2 5 2 3 1 5 4 3 1 ...  
## $ E2 : num 2 2 1 5 1 5 1 3 1 4 ...  
## $ E3 : num 5 3 1 2 3 2 5 5 5 2 ...  
## $ E4 : num 2 3 4 4 3 4 1 3 1 5 ...  
## $ E5 : num 5 3 5 3 3 1 5 5 5 2 ...  
## $ E6 : num 1 3 1 4 1 3 1 1 1 4 ...  
## $ E7 : num 4 1 1 3 3 2 5 4 5 1 ...  
## $ E8 : num 3 5 5 4 1 4 4 3 2 4 ...  
## $ E9 : num 5 1 5 4 3 1 4 4 5 1 ...  
## $ E10 : num 1 5 1 5 5 5 1 3 3 5 ...  
## $ N1 : num 1 2 5 5 3 1 2 1 2 5 ...  
## $ N2 : num 5 3 1 4 3 5 4 4 4 2 ...  
## $ N3 : num 2 4 5 4 3 4 2 4 5 5 ...  
## $ N4 : num 5 2 5 2 4 5 4 4 3 2 ...  
## $ N5 : num 1 3 5 4 3 1 2 1 3 3 ...  
## $ N6 : num 1 4 5 5 3 4 2 1 5 4 ...  
## $ N7 : num 1 3 5 5 3 4 3 1 5 3 ...  
## $ N8 : num 1 2 5 5 3 1 2 1 4 2 ...  
## $ N9 : num 1 2 5 4 3 5 2 1 3 3 ...  
## $ N10 : num 1 4 5 5 4 2 2 1 3 4 ...  
## $ A1 : num 1 1 5 2 5 2 5 2 1 2 ...  
## $ A2 : num 5 3 1 5 5 2 5 5 5 3 ...  
## $ A3 : num 1 3 5 4 3 3 1 1 1 1 ...  
## $ A4 : num 5 4 5 4 5 4 5 4 5 4 ...  
## $ A5 : num 2 4 1 3 1 3 1 3 1 2 ...  
## $ A6 : num 3 4 5 5 5 4 5 3 5 4 ...  
## $ A7 : num 1 2 1 3 1 3 1 1 1 3 ...  
## $ A8 : num 5 3 5 4 5 5 5 3 5 3 ...  
## $ A9 : num 4 4 5 4 5 5 4 4 5 3 ...  
## $ A10 : num 5 3 5 3 5 3 5 5 4 2 ...  
## $ C1 : num 4 4 4 3 3 2 2 4 4 5 ...  
## $ C2 : num 1 1 1 3 1 5 4 2 3 2 ...  
## $ C3 : num 5 3 5 4 5 4 3 5 5 4 ...  
## $ C4 : num 1 2 1 5 3 3 3 1 2 2 ...  
## $ C5 : num 5 3 5 1 3 3 3 4 5 3 ...  
## $ C6 : num 1 1 1 4 1 4 3 1 2 2 ...  
## $ C7 : num 4 5 5 5 1 5 3 4 5 4 ...  
## $ C8 : num 1 1 1 4 3 3 3 1 2 2 ...  
## $ C9 : num 4 4 5 2 3 5 3 3 4 4 ...  
## $ C10 : num 5 4 5 3 3 3 3 5 3 4 ...  
## $ O1 : num 4 3 4 4 3 4 3 3 3 4 ...  
## $ O2 : num 1 3 5 3 1 2 1 1 3 2 ...  
## $ O3 : num 3 3 5 5 1 1 5 5 5 5 ...  
## $ O4 : num 1 3 1 2 1 3 1 1 3 2 ...  
## $ O5 : num 5 2 5 4 3 3 4 4 5 4 ...  
## $ O6 : num 1 3 1 2 1 5 1 1 1 1 ...  
## $ O7 : num 4 3 5 5 3 5 4 5 5 4 ...  
## $ O8 : num 2 1 5 2 1 4 3 3 3 3 ...  
## $ O9 : num 5 3 5 5 5 5 3 2 4 4 ...  
## $ O10 : num 5 2 5 5 3 3 4 5 5 4 ...  
## $ faixa\_etária: num 5 4 1 1 2 3 1 2 3 1 ...

1. Atribua labels (apply\_labels()) aos itens do teste de personalidade, usando as afirmações que foram respondidas pelos participantes da pesquisa.

# cópia\_de\_segurança2 <- big\_five  
library(expss)  
big\_five <- apply\_labels(big\_five,  
E1 = "I am the life of the party",  
E2 = "I don't talk a lot",  
E3 = "I feel comfortable around people",  
E4 = "I keep in the background",  
E5 = "I start conversations",  
E6 = "I have little to say",  
E7 = "I talk to a lot of different people at parties",  
E8 = "I don't like to draw attention to myself",  
E9 = "I don't mind being the center of attention",  
E10 = "I am quiet around strangers",  
N1 = "I get stressed out easily",  
N2 = "I am relaxed most of the time",  
N3 = "I worry about things",  
N4 = "I seldom feel blue",  
N5 = "I am easily disturbed",  
N6 = "I get upset easily",  
N7 = "I change my mood a lot",  
N8 = "I have frequent mood swings",  
N9 = "I get irritated easily",  
N10 = "I often feel blue",  
A1 = "I feel little concern for others",  
A2 = "I am interested in people",  
A3 = "I insult people",  
A4 = "I sympathize with others' feelings",  
A5 = "I am not interested in other people's problems",  
A6 = "I have a soft heart",  
A7 = "I am not really interested in others",  
A8 = "I take time out for others",  
A9 = "I feel others' emotions",  
A10 = "I make people feel at ease",  
C1 = "I am always prepared",  
C2 = "I leave my belongings around",  
C3 = "I pay attention to details",  
C4 = "I make a mess of things",  
C5 = "I get chores done right away",  
C6 = "I often forget to put things back in their proper place",  
C7 = "I like order",  
C8 = "I shirk my duties",  
C9 = "I follow a schedule",  
C10 = "I am exacting in my work",  
O1 = "I have a rich vocabulary.",  
O2 = "I have difficulty understanding abstract ideas",  
O3 = "I have a vivid imagination",  
O4 = "I am not interested in abstract ideas",  
O5 = "I have excellent ideas",  
O6 = "I do not have a good imagination",  
O7 = "I am quick to understand things",  
O8 = "I use difficult words",  
O9 = "I spend time reflecting on things",  
O10 = "I am full of ideas")

1. Nos cinco fatores desse instrumento, há itens na direção do fator e itens no sentido inverso. Por exemplo, na escala de extroversão há itens como *“I am the life of the party”*, que descreve uma pessoa extrovertida, e itens como *“I don’t talk a lot”*, que descreve uma pessoa introvertida. Esse problema precisa ser resolvido antes de calcularmos os escores das pessoas nesse fator. Se as pessoas responderam esses itens usando uma escala Likert de 5 pontos, não podemos somar os valores desses itens diretamente porque eles têm significados diferentes. Por exemplo, se uma pessoa atribuiu valor 5 ao item *“I am the life of the party”*, esse 5 indica que ela é muito **ex**trovertida. Mas o mesmo valor 5 atribuído ao item *“I don’t talk a lot”* indica que a pessoa é muito **in**trovertida. Portanto, temos um problema porque o mesmo valor (5) significa coisas diferentes em itens diferentes. Para evitarmos esse problema temos que inverter os valores dos itens que vão na direção da introversão, mudar o 1 para 5, o 2 para 4, o 3 permanece 3, o 4 para 2 e o 5 para 1.

library(expss)  
  
# uma solução seria usar ifelse  
big\_five$E2\_mod <- ifelse(big\_five$E2 == 1,5,  
 ifelse(big\_five$E2 == 2,4,  
 ifelse(big\_five$E2 == 3,3,  
 ifelse(big\_five$E2 == 4,2,  
 ifelse(big\_five$E2 == 5,1,NA)))))  
# no entanto, essa solução exigiria criar um código desse para cada variável.  
# a função recode() do expss facilita a modificação de variáveis em bloco.  
  
recode(big\_five[ ,c('E2','E4','E6','E8','E10','N2','N4','A1','A3','A5','A7','C2','C4','C6','C8','O2','O4','O6')]) <- c(1~5,2~4,3~3,4~2,5~1,NA~NA)

## Exercícios usando Tidyverse

Nos próximos exercícios vamos trabalhar com a base de dados imdb, que apresenta uma série de informações sobre filmes disponibilizados pela Amazon. Baixe o dataframe [aqui](https://1drv.ms/u/s!ApvtT9NG4aYoko4EF7YU3p-l3oOOcw?e=rsU4Ti) (link válido até 30/09/2021) e importe-o para dentro do RStudio e use as funções do tidyverse para resolver os próximos exercícios.

imdb <- readr::read\_rds("imdb.rds")  
glimpse(imdb)

## Rows: 3,807  
## Columns: 15  
## $ titulo <chr> "Avatar ", "Pirates of the Caribbean: At World's End ",~  
## $ ano <int> 2009, 2007, 2012, 2012, 2007, 2010, 2015, 2016, 2006, 2~  
## $ diretor <chr> "James Cameron", "Gore Verbinski", "Christopher Nolan",~  
## $ duracao <int> 178, 169, 164, 132, 156, 100, 141, 183, 169, 151, 150, ~  
## $ cor <chr> "Color", "Color", "Color", "Color", "Color", "Color", "~  
## $ generos <chr> "Action|Adventure|Fantasy|Sci-Fi", "Action|Adventure|Fa~  
## $ pais <chr> "USA", "USA", "USA", "USA", "USA", "USA", "USA", "USA",~  
## $ classificacao <chr> "A partir de 13 anos", "A partir de 13 anos", "A partir~  
## $ orcamento <int> 237000000, 300000000, 250000000, 263700000, 258000000, ~  
## $ receita <int> 760505847, 309404152, 448130642, 73058679, 336530303, 2~  
## $ nota\_imdb <dbl> 7.9, 7.1, 8.5, 6.6, 6.2, 7.8, 7.5, 6.9, 6.1, 7.3, 6.5, ~  
## $ likes\_facebook <int> 33000, 0, 164000, 24000, 0, 29000, 118000, 197000, 0, 5~  
## $ ator\_1 <chr> "CCH Pounder", "Johnny Depp", "Tom Hardy", "Daryl Sabar~  
## $ ator\_2 <chr> "Joel David Moore", "Orlando Bloom", "Christian Bale", ~  
## $ ator\_3 <chr> "Wes Studi", "Jack Davenport", "Joseph Gordon-Levitt", ~

1. Crie uma variável id para todos os sujeitos do dataframe e coloque essa variável na primeira coluna. Não se esqueça de salvar essa transformação no dataframe imdb.

#imdb\_copia <- imdb  
nrow(imdb)

## [1] 3807

imdb$id <- paste("s",1:nrow(imdb),sep = "")  
  
imdb <- imdb %>% select(id,everything())  
names(imdb)

## [1] "id" "titulo" "ano" "diretor"   
## [5] "duracao" "cor" "generos" "pais"   
## [9] "classificacao" "orcamento" "receita" "nota\_imdb"   
## [13] "likes\_facebook" "ator\_1" "ator\_2" "ator\_3"

1. Selecione as variáveis título, duração, cor e gênero dos 1000 primeiros casos do banco de dados. Não precisa salvar.

imdb %>% select(titulo,duracao,cor,generos) %>% slice(1:1000)

## # A tibble: 1,000 x 4  
## titulo duracao cor generos   
## <chr> <int> <chr> <chr>   
## 1 Avatar  178 Color Action|Adventure|Fantasy|Sci-Fi   
## 2 Pirates of the Caribbean: ~ 169 Color Action|Adventure|Fantasy   
## 3 The Dark Knight Rises  164 Color Action|Thriller   
## 4 John Carter  132 Color Action|Adventure|Sci-Fi   
## 5 Spider-Man 3  156 Color Action|Adventure|Romance   
## 6 Tangled  100 Color Adventure|Animation|Comedy|Family|~  
## 7 Avengers: Age of Ultron  141 Color Action|Adventure|Sci-Fi   
## 8 Batman v Superman: Dawn of~ 183 Color Action|Adventure|Sci-Fi   
## 9 Superman Returns  169 Color Action|Adventure|Sci-Fi   
## 10 Pirates of the Caribbean: ~ 151 Color Action|Adventure|Fantasy   
## # ... with 990 more rows

1. Monte um dataframe somente com os filmes do diretor Clint Eastwood. Salve-o como imdb\_clint.

imdb\_clint <- imdb %>% filter(diretor == "Clint Eastwood")

1. Reorganize o dataframe com filmes do diretor Clint Eastwood por ordem decrescente de receita e visualise os dados para ver se funcionou. Salve o dataframe com o mesmo nome.

imdb\_clint <- imdb\_clint %>% arrange(desc(receita)) %>% view()

1. Selecione uma amostra aleatória do dataframe imdb com 45% dos filmes relacionados. Não precisa salvar.

imdb %>% slice\_sample(prop = .45)

## # A tibble: 1,713 x 16  
## id titulo ano diretor duracao cor generos pais classificacao  
## <chr> <chr> <int> <chr> <int> <chr> <chr> <chr> <chr>   
## 1 s1310 The Number~ 2007 Joel Sc~ 98 Color Mystery|~ USA A partir de ~  
## 2 s264 End of Day~ 1999 Peter H~ 121 Color Action|F~ USA A partir de ~  
## 3 s1325 Out of Afr~ 1985 Sydney ~ 161 Color Biograph~ USA Livre   
## 4 s3701 Good Dick  2008 Mariann~ 86 Color Comedy|D~ USA A partir de ~  
## 5 s1122 Delgo  2008 Marc F.~ 94 Color Adventur~ USA Livre   
## 6 s3794 Breaking U~ 2009 Daryl W~ 88 Color Romance USA Outros   
## 7 s2076 Baby Boy  2001 John Si~ 130 Black~ Crime|Dr~ USA A partir de ~  
## 8 s1455 The Assass~ 2007 Andrew ~ 160 Color Biograph~ USA A partir de ~  
## 9 s1673 The Fisher~ 1991 Terry G~ 137 Color Comedy|D~ USA A partir de ~  
## 10 s2515 Cedar Rapi~ 2011 Miguel ~ 87 Color Comedy USA A partir de ~  
## # ... with 1,703 more rows, and 7 more variables: orcamento <int>,  
## # receita <int>, nota\_imdb <dbl>, likes\_facebook <int>, ator\_1 <chr>,  
## # ator\_2 <chr>, ator\_3 <chr>

1. Calcule as médias de orçamento e de receita, por diretor. Salve o resultado num objeto chamado imdb\_lucro. Obs.: Nos resultados vão aparecer sinais como NA ou NaN, porque há informações não disponíveis que acabam não permitindo os cálculos desejados.

view(imdb)  
imdb\_lucro <- imdb %>% group\_by(diretor) %>% summarise(Orçamento\_médio = mean(orcamento, na.rm = TRUE), Receita\_média = mean(receita, na.rm = TRUE))

1. No dataframe anterior, crie a variável lucro, a partir das médias de orçamento e receita. Identifique o diretor mais lucrativo e o que dá mais prejuízo.

# o mais lucrativo  
imdb\_lucro %>% mutate(lucro = Receita\_média - Orçamento\_médio) %>% arrange(desc(lucro))

## # A tibble: 1,813 x 4  
## diretor Orçamento\_médio Receita\_média lucro  
## <chr> <dbl> <dbl> <dbl>  
## 1 Tim Miller 58000000 363024263 305024263   
## 2 George Lucas 70955400 348283696 277328296   
## 3 Richard Marquand 32500000 309125409 276625409   
## 4 Irvin Kershner 18000000 290158751 272158751   
## 5 Kyle Balda 74000000 336029560 262029560   
## 6 Colin Trevorrow 75375000 328092532. 252717532.  
## 7 Chris Buck 150000000 400736600 250736600   
## 8 Joss Whedon 182500000 432721657 250221657   
## 9 Pierre Coffin 72500000 309775640 237275640   
## 10 Lee Unkrich 200000000 414984497 214984497   
## # ... with 1,803 more rows

# o que dá mais prejuízo  
imdb\_lucro %>% mutate(lucro = Receita\_média - Orçamento\_médio) %>% arrange(lucro)

## # A tibble: 1,813 x 4  
## diretor Orçamento\_médio Receita\_média lucro  
## <chr> <dbl> <dbl> <dbl>  
## 1 Carl Rinsch 175000000 38297305 -136702695   
## 2 Hironobu Sakaguchi 137000000 32131830 -104868170   
## 3 Joe Wright 120000000 36285492. -83714508.  
## 4 Sergey Bodrov 95000000 17176900 -77823100   
## 5 Ericson Core 105000000 28772222 -76227778   
## 6 Simon Wells 115000000 39032067 -75967933   
## 7 Cedric Nicolas-Troyan 115000000 47952020 -67047980   
## 8 Oliver Hirschbiegel 80000000 15071514 -64928486   
## 9 Will Finn 90000000 29243672. -60756328.  
## 10 Akiva Goldsman 60000000 22451 -59977549   
## # ... with 1,803 more rows

# identificação, por curiosidade, do lucro obtido por James Cameron  
imdb\_lucro %>% mutate(lucro = Receita\_média - Orçamento\_médio) %>% arrange(lucro) %>% filter(diretor == "James Cameron")

## # A tibble: 1 x 4  
## diretor Orçamento\_médio Receita\_média lucro  
## <chr> <dbl> <dbl> <dbl>  
## 1 James Cameron 123666667. 318287652. 194620985

# Para identificar a posição de um diretor em termos de lucratividade  
imdb\_lucro$posição <- paste(1:nrow(imdb\_lucro),sep = "")  
  
imdb\_lucro %>% filter(diretor == "Daniel Myrick")

## # A tibble: 1 x 4  
## diretor Orçamento\_médio Receita\_média posição  
## <chr> <dbl> <dbl> <chr>   
## 1 Daniel Myrick 60000 140530114 348

1. Voltando ao dataframe completo (imdb). Obtenha um dataframe que contenha somente os diretores que tiverem 3 filmes ou mais no catálogo.

Dica1: A informação de quantos filmes cada diretor tem, não está diretamente disponível no dataframe. O que a gente tem é uma linha por filme. Então, o mesmo diretor pode constar em várias linhas, correspondentes aos filmes que ele dirigiu.

Dica2: nesse caso, temos que usar o operador %in% ao invés de == porque %in% permite comparar objetos de tamanhos diferentes, enquanto o == só permite comparar objetos de tamanhos iguais, ou uma coluna com UMA informação. Veja o exemplo com o dataframe big\_five

# no dataframe big\_five, se quisermos filtrar os participantes estadunidenses, podemos usar o filter.   
big\_five %>% filter(país == "US") %>% glimpse()

## Rows: 8,753  
## Columns: 60  
## $ id <chr> "s1", "s2", "s5", "s6", "s7", "s9", "s10", "s13", "s14", ~  
## $ raça <fct> "Caucasiano (Europeu)", "Outro", "Sudeste Asiático (Chinê~  
## $ idade <dbl> 53, 46, 25, 31, 20, 39, 18, 22, 21, 28, 21, 21, 26, 29, 2~  
## $ ing\_nat <fct> Sim, Sim, Não, Sim, Sim, Sim, Sim, Sim, Sim, Não, Sim, Si~  
## $ gênero <fct> Masculino, Feminino, Feminino, Feminino, Feminino, Femini~  
## $ mão\_dom <fct> 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~  
## $ fonte <dbl> 1, 1, 2, 2, 5, 4, 5, 2, 5, 2, 5, 5, 1, 1, 2, 1, 2, 4, 1, ~  
## $ país <fct> US, US, US, US, US, US, US, US, US, US, US, US, US, US, U~  
## $ E1 <labelled> 4, 2, 3, 1, 5, 3, 1, 3, 1, 3, 2, 4, 2, 2, 2, 3, 5, 3~  
## $ E2 <labelled> 4, 4, 5, 1, 5, 5, 2, 3, 3, 3, 3, 5, 4, 3, 3, 2, 1, 3~  
## $ E3 <labelled> 5, 3, 3, 2, 5, 5, 2, 4, 2, 3, 2, 5, 3, 4, 1, 4, 4, 4~  
## $ E4 <labelled> 4, 3, 3, 2, 5, 5, 1, 4, 1, 2, 3, 4, 3, 4, 2, 2, 5, 3~  
## $ E5 <labelled> 5, 3, 3, 1, 5, 5, 2, 4, 1, 3, 3, 5, 3, 4, 2, 2, 5, 2~  
## $ E6 <labelled> 5, 3, 5, 3, 5, 5, 2, 4, 5, 4, 5, 5, 3, 5, 2, 3, 5, 4~  
## $ E7 <labelled> 4, 1, 3, 2, 5, 5, 1, 2, 1, 2, 1, 5, 1, 3, 1, 3, 5, 3~  
## $ E8 <labelled> 3, 1, 5, 2, 2, 4, 2, 3, 1, 2, 3, 3, 3, 2, 3, 3, 5, 2~  
## $ E9 <labelled> 5, 1, 3, 1, 4, 5, 1, 4, 1, 3, 4, 5, 3, 2, 3, 4, 5, 2~  
## $ E10 <labelled> 5, 1, 1, 1, 5, 3, 1, 3, 1, 1, 2, 5, 3, 3, 2, 1, 4, 2~  
## $ N1 <labelled> 1, 2, 3, 1, 2, 2, 5, 3, 5, 2, 2, 5, 4, 1, 3, 4, 3, 4~  
## $ N2 <labelled> 1, 3, 3, 1, 2, 2, 4, 3, 3, 2, 2, 4, 2, 4, 3, 2, 2, 3~  
## $ N3 <labelled> 2, 4, 3, 4, 2, 5, 5, 3, 5, 4, 2, 5, 3, 2, 5, 4, 4, 4~  
## $ N4 <labelled> 1, 4, 2, 1, 2, 3, 4, 3, 4, 2, 2, 4, 5, 3, 4, 3, 1, 3~  
## $ N5 <labelled> 1, 3, 3, 1, 2, 3, 3, 2, 5, 4, 1, 3, 3, 2, 3, 2, 3, 2~  
## $ N6 <labelled> 1, 4, 3, 4, 2, 5, 4, 2, 5, 4, 2, 3, 2, 1, 3, 2, 3, 2~  
## $ N7 <labelled> 1, 3, 3, 4, 3, 5, 3, 4, 3, 2, 2, 3, 2, 2, 4, 3, 4, 3~  
## $ N8 <labelled> 1, 2, 3, 1, 2, 4, 2, 4, 2, 2, 2, 3, 2, 1, 3, 2, 3, 3~  
## $ N9 <labelled> 1, 2, 3, 5, 2, 3, 3, 2, 5, 3, 2, 4, 4, 2, 3, 1, 3, 2~  
## $ N10 <labelled> 1, 4, 4, 2, 2, 3, 4, 3, 3, 2, 2, 2, 4, 2, 3, 3, 2, 2~  
## $ A1 <labelled> 5, 5, 1, 4, 1, 5, 4, 5, 5, 5, 2, 5, 5, 4, 5, 5, 2, 5~  
## $ A2 <labelled> 5, 3, 5, 2, 5, 5, 3, 4, 1, 4, 2, 5, 3, 4, 4, 4, 5, 2~  
## $ A3 <labelled> 5, 3, 3, 3, 5, 5, 5, 5, 5, 4, 2, 4, 4, 3, 5, 5, 2, 5~  
## $ A4 <labelled> 5, 4, 5, 4, 5, 5, 4, 5, 4, 4, 2, 5, 4, 3, 4, 5, 4, 3~  
## $ A5 <labelled> 4, 2, 5, 3, 5, 5, 4, 5, 4, 4, 1, 4, 3, 5, 4, 3, 2, 4~  
## $ A6 <labelled> 3, 4, 5, 4, 5, 5, 4, 5, 3, 4, 2, 5, 2, 2, 4, 3, 4, 4~  
## $ A7 <labelled> 5, 4, 5, 3, 5, 5, 3, 4, 1, 4, 1, 5, 3, 4, 3, 3, 5, 4~  
## $ A8 <labelled> 5, 3, 5, 5, 5, 5, 3, 4, 4, 4, 2, 5, 3, 4, 5, 3, 2, 4~  
## $ A9 <labelled> 4, 4, 5, 5, 4, 5, 3, 5, 4, 3, 1, 5, 4, 2, 4, 4, 4, 4~  
## $ A10 <labelled> 5, 3, 5, 3, 5, 4, 2, 4, 2, 4, 2, 3, 4, 3, 3, 4, 5, 3~  
## $ C1 <labelled> 4, 4, 3, 2, 2, 4, 5, 4, 5, 4, 4, 3, 1, 4, 4, 3, 2, 3~  
## $ C2 <labelled> 5, 5, 5, 1, 2, 3, 4, 4, 4, 4, 4, 2, 4, 3, 4, 1, 1, 3~  
## $ C3 <labelled> 5, 3, 5, 4, 3, 5, 4, 4, 5, 4, 4, 3, 5, 5, 4, 4, 2, 3~  
## $ C4 <labelled> 5, 4, 3, 3, 3, 4, 4, 4, 5, 4, 5, 3, 4, 5, 4, 3, 1, 3~  
## $ C5 <labelled> 5, 3, 3, 3, 3, 5, 3, 3, 3, 4, 3, 1, 2, 4, 4, 1, 1, 4~  
## $ C6 <labelled> 5, 5, 5, 2, 3, 4, 4, 2, 4, 4, 4, 5, 5, 4, 5, 1, 1, 3~  
## $ C7 <labelled> 4, 5, 1, 5, 3, 5, 4, 4, 5, 4, 4, 3, 4, 4, 5, 4, 3, 3~  
## $ C8 <labelled> 5, 5, 3, 3, 3, 4, 4, 5, 4, 4, 5, 3, 4, 5, 4, 2, 3, 3~  
## $ C9 <labelled> 4, 4, 3, 5, 3, 4, 4, 5, 4, 4, 4, 4, 2, 3, 4, 2, 5, 3~  
## $ C10 <labelled> 5, 4, 3, 3, 3, 3, 4, 4, 4, 4, 5, 3, 2, 4, 4, 4, 5, 2~  
## $ O1 <labelled> 4, 3, 3, 4, 3, 3, 4, 3, 3, 3, 4, 3, 4, 5, 4, 4, 4, 2~  
## $ O2 <labelled> 5, 3, 5, 4, 5, 3, 4, 4, 5, 4, 5, 5, 4, 5, 4, 3, 5, 3~  
## $ O3 <labelled> 3, 3, 1, 1, 5, 5, 5, 4, 3, 4, 5, 3, 5, 4, 5, 4, 5, 4~  
## $ O4 <labelled> 5, 3, 5, 3, 5, 3, 4, 4, 3, 3, 5, 3, 4, 5, 3, 2, 5, 3~  
## $ O5 <labelled> 5, 2, 3, 3, 4, 5, 4, 3, 4, 4, 5, 3, 2, 4, 4, 4, 5, 3~  
## $ O6 <labelled> 5, 3, 5, 1, 5, 5, 5, 4, 5, 4, 5, 3, 5, 5, 4, 4, 5, 4~  
## $ O7 <labelled> 4, 3, 3, 5, 4, 5, 4, 3, 4, 4, 4, 5, 5, 5, 3, 4, 4, 3~  
## $ O8 <labelled> 2, 1, 1, 4, 3, 3, 3, 3, 3, 2, 4, 4, 3, 4, 4, 4, 4, 2~  
## $ O9 <labelled> 5, 3, 5, 5, 3, 4, 4, 5, 5, 4, 5, 3, 4, 4, 5, 5, 2, 4~  
## $ O10 <labelled> 5, 2, 3, 3, 4, 5, 4, 4, 4, 4, 5, 3, 4, 4, 3, 4, 5, 3~  
## $ faixa\_etária <dbl> 5, 4, 2, 3, 1, 3, 1, 2, 2, 2, 2, 2, 2, 2, 2, 5, 4, 2, 1, ~  
## $ E2\_mod <dbl> 4, 4, 5, 1, 5, 5, 2, 3, 3, 3, 3, 5, 4, 3, 3, 2, 1, 3, 2, ~

# porém, se quisermos filtrar os países com mais de 100 participantes, podemos fazer o seguinte:  
# pegue o dataframe big\_five; ENTÃO, conte a frequência dos países; ENTÃO, filtre os países cujo n for maior que 100. Guarde essa tabela num objeto chamado pais.  
pais <- big\_five %>% count(país) %>% filter(n>=100)  
  
# guardar somente a coluna com o nome dos países selecionados.  
lista\_país <- pais$país  
  
# com isso obtemos uma lista dos países (lista\_país) que contêm mais de 100 participantes. Agora, queremos selecionar em big\_five$país somente aqueles que forem iguais a um dos países constantes na lista\_país. Para isso temos que usar o código %in% ao invés de ==   
  
big\_five %>% filter(país %in% lista\_país) %>% glimpse()

## Rows: 17,459  
## Columns: 60  
## $ id <chr> "s1", "s2", "s3", "s4", "s5", "s6", "s7", "s8", "s9", "s1~  
## $ raça <fct> "Caucasiano (Europeu)", "Outro", "Raça Mista", "Caucasian~  
## $ idade <dbl> 53, 46, 14, 19, 25, 31, 20, 23, 39, 18, 17, 15, 22, 21, 2~  
## $ ing\_nat <fct> Sim, Sim, Não, Não, Não, Sim, Sim, Não, Sim, Sim, Não, Nã~  
## $ gênero <fct> Masculino, Feminino, Feminino, Feminino, Feminino, Femini~  
## $ mão\_dom <fct> 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, ~  
## $ fonte <dbl> 1, 1, 1, 1, 2, 2, 5, 2, 4, 5, 1, 1, 2, 5, 2, 5, 2, 5, 5, ~  
## $ país <fct> US, US, PK, RO, US, US, US, IN, US, US, IT, IN, US, US, U~  
## $ E1 <labelled> 4, 2, 5, 2, 3, 1, 5, 4, 3, 1, 1, 3, 3, 1, 3, 2, 1, 4~  
## $ E2 <labelled> 4, 4, 5, 1, 5, 1, 5, 3, 5, 2, 1, 3, 3, 3, 3, 3, 3, 5~  
## $ E3 <labelled> 5, 3, 1, 2, 3, 2, 5, 5, 5, 2, 2, 5, 4, 2, 3, 2, 2, 5~  
## $ E4 <labelled> 4, 3, 2, 2, 3, 2, 5, 3, 5, 1, 1, 3, 4, 1, 2, 3, 2, 4~  
## $ E5 <labelled> 5, 3, 5, 3, 3, 1, 5, 5, 5, 2, 1, 3, 4, 1, 3, 3, 2, 5~  
## $ E6 <labelled> 5, 3, 5, 2, 5, 3, 5, 5, 5, 2, 2, 3, 4, 5, 4, 5, 2, 5~  
## $ E7 <labelled> 4, 1, 1, 3, 3, 2, 5, 4, 5, 1, 1, 2, 2, 1, 2, 1, 1, 5~  
## $ E8 <labelled> 3, 1, 1, 2, 5, 2, 2, 3, 4, 2, 2, 2, 3, 1, 2, 3, 2, 3~  
## $ E9 <labelled> 5, 1, 5, 4, 3, 1, 4, 4, 5, 1, 1, 3, 4, 1, 3, 4, 3, 5~  
## $ E10 <labelled> 5, 1, 5, 1, 1, 1, 5, 3, 3, 1, 1, 3, 3, 1, 1, 2, 2, 5~  
## $ N1 <labelled> 1, 2, 5, 5, 3, 1, 2, 1, 2, 5, 5, 1, 3, 5, 2, 2, 4, 5~  
## $ N2 <labelled> 1, 3, 5, 2, 3, 1, 2, 2, 2, 4, 3, 1, 3, 3, 2, 2, 4, 4~  
## $ N3 <labelled> 2, 4, 5, 4, 3, 4, 2, 4, 5, 5, 5, 3, 3, 5, 4, 2, 3, 5~  
## $ N4 <labelled> 1, 4, 1, 4, 2, 1, 2, 2, 3, 4, 3, 3, 3, 4, 2, 2, 4, 4~  
## $ N5 <labelled> 1, 3, 5, 4, 3, 1, 2, 1, 3, 3, 2, 2, 2, 5, 4, 1, 1, 3~  
## $ N6 <labelled> 1, 4, 5, 5, 3, 4, 2, 1, 5, 4, 5, 3, 2, 5, 4, 2, 3, 3~  
## $ N7 <labelled> 1, 3, 5, 5, 3, 4, 3, 1, 5, 3, 3, 2, 4, 3, 2, 2, 1, 3~  
## $ N8 <labelled> 1, 2, 5, 5, 3, 1, 2, 1, 4, 2, 3, 3, 4, 2, 2, 2, 2, 3~  
## $ N9 <labelled> 1, 2, 5, 4, 3, 5, 2, 1, 3, 3, 4, 2, 2, 5, 3, 2, 2, 4~  
## $ N10 <labelled> 1, 4, 5, 5, 4, 2, 2, 1, 3, 4, 3, 4, 3, 3, 2, 2, 3, 2~  
## $ A1 <labelled> 5, 5, 1, 4, 1, 4, 1, 4, 5, 4, 4, 2, 5, 5, 5, 2, 2, 5~  
## $ A2 <labelled> 5, 3, 1, 5, 5, 2, 5, 5, 5, 3, 4, 4, 4, 1, 4, 2, 2, 5~  
## $ A3 <labelled> 5, 3, 1, 2, 3, 3, 5, 5, 5, 5, 4, 4, 5, 5, 4, 2, 3, 4~  
## $ A4 <labelled> 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 4, 2, 5, 4, 4, 2, 1, 5~  
## $ A5 <labelled> 4, 2, 5, 3, 5, 3, 5, 3, 5, 4, 5, 1, 5, 4, 4, 1, 2, 4~  
## $ A6 <labelled> 3, 4, 5, 5, 5, 4, 5, 3, 5, 4, 5, 5, 5, 3, 4, 2, 1, 5~  
## $ A7 <labelled> 5, 4, 5, 3, 5, 3, 5, 5, 5, 3, 5, 4, 4, 1, 4, 1, 1, 5~  
## $ A8 <labelled> 5, 3, 5, 4, 5, 5, 5, 3, 5, 3, 4, 5, 4, 4, 4, 2, 3, 5~  
## $ A9 <labelled> 4, 4, 5, 4, 5, 5, 4, 4, 5, 3, 3, 4, 5, 4, 3, 1, 1, 5~  
## $ A10 <labelled> 5, 3, 5, 3, 5, 3, 5, 5, 4, 2, 3, 4, 4, 2, 4, 2, 1, 3~  
## $ C1 <labelled> 4, 4, 4, 3, 3, 2, 2, 4, 4, 5, 3, 3, 4, 5, 4, 4, 4, 3~  
## $ C2 <labelled> 5, 5, 5, 3, 5, 1, 2, 4, 3, 4, 2, 3, 4, 4, 4, 4, 3, 2~  
## $ C3 <labelled> 5, 3, 5, 4, 5, 4, 3, 5, 5, 4, 3, 4, 4, 5, 4, 4, 5, 3~  
## $ C4 <labelled> 5, 4, 5, 1, 3, 3, 3, 5, 4, 4, 2, 3, 4, 5, 4, 5, 4, 3~  
## $ C5 <labelled> 5, 3, 5, 1, 3, 3, 3, 4, 5, 3, 1, 3, 3, 3, 4, 3, 2, 1~  
## $ C6 <labelled> 5, 5, 5, 2, 5, 2, 3, 5, 4, 4, 1, 1, 2, 4, 4, 4, 2, 5~  
## $ C7 <labelled> 4, 5, 5, 5, 1, 5, 3, 4, 5, 4, 3, 3, 4, 5, 4, 4, 3, 3~  
## $ C8 <labelled> 5, 5, 5, 2, 3, 3, 3, 5, 4, 4, 3, 3, 5, 4, 4, 5, 5, 3~  
## $ C9 <labelled> 4, 4, 5, 2, 3, 5, 3, 3, 4, 4, 3, 3, 5, 4, 4, 4, 3, 4~  
## $ C10 <labelled> 5, 4, 5, 3, 3, 3, 3, 5, 3, 4, 3, 4, 4, 4, 4, 5, 5, 3~  
## $ O1 <labelled> 4, 3, 4, 4, 3, 4, 3, 3, 3, 4, 5, 3, 3, 3, 3, 4, 5, 3~  
## $ O2 <labelled> 5, 3, 1, 3, 5, 4, 5, 5, 3, 4, 3, 5, 4, 5, 4, 5, 5, 5~  
## $ O3 <labelled> 3, 3, 5, 5, 1, 1, 5, 5, 5, 5, 5, 4, 4, 3, 4, 5, 4, 3~  
## $ O4 <labelled> 5, 3, 5, 4, 5, 3, 5, 5, 3, 4, 3, 5, 4, 3, 3, 5, 5, 3~  
## $ O5 <labelled> 5, 2, 5, 4, 3, 3, 4, 4, 5, 4, 4, 4, 3, 4, 4, 5, 4, 3~  
## $ O6 <labelled> 5, 3, 5, 4, 5, 1, 5, 5, 5, 5, 5, 1, 4, 5, 4, 5, 5, 3~  
## $ O7 <labelled> 4, 3, 5, 5, 3, 5, 4, 5, 5, 4, 3, 4, 3, 4, 4, 4, 5, 5~  
## $ O8 <labelled> 2, 1, 5, 2, 1, 4, 3, 3, 3, 3, 5, 3, 3, 3, 2, 4, 5, 4~  
## $ O9 <labelled> 5, 3, 5, 5, 5, 5, 3, 2, 4, 4, 4, 4, 5, 5, 4, 5, 5, 3~  
## $ O10 <labelled> 5, 2, 5, 5, 3, 3, 4, 5, 5, 4, 5, 5, 4, 4, 4, 5, 4, 3~  
## $ faixa\_etária <dbl> 5, 4, 1, 1, 2, 3, 1, 2, 3, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, ~  
## $ E2\_mod <dbl> 4, 4, 5, 1, 5, 1, 5, 3, 5, 2, 1, 3, 3, 3, 3, 3, 3, 5, 3, ~

# se rodarmos com == no lugar de %in% vai dar erro porque o comprimento da lista não é igual ao comprimento do dataframe big\_five.  
  
#big\_five %>% filter(país == lista\_país)

# contar a ocorrência de diretores e filtrar os que aparecem 3x ou mais.  
imdb\_diretores\_3mais <- imdb %>% count(diretor) %>% filter(n>=3) %>% arrange(n)  
  
# retirar somente o vetor "diretores" do objeto imdb\_diretores\_3mais   
vetor\_diretores <- imdb\_diretores\_3mais$diretor  
  
# filtrar imdb caso o que estiver em diretor seja igual a um dos diretores constantes no vetor\_diretores.   
imdb\_sub1 <- imdb %>% filter(diretor %in% vetor\_diretores)

1. Entre os diretores com 3 filmes ou mais, qual tem maior média de nota\_imdb?

imdb\_sub1 %>% group\_by(diretor) %>% summarise(média\_imdb = mean(nota\_imdb)) %>% arrange(desc(média\_imdb))

## # A tibble: 411 x 2  
## diretor média\_imdb  
## <chr> <dbl>  
## 1 Christopher Nolan 8.43  
## 2 Pete Docter 8.23  
## 3 Quentin Tarantino 8.2   
## 4 Frank Capra 8.06  
## 5 Tony Kaye 8.03  
## 6 Billy Wilder 7.98  
## 7 Frank Darabont 7.98  
## 8 Joss Whedon 7.92  
## 9 James Cameron 7.88  
## 10 Charles Ferguson 7.87  
## # ... with 401 more rows