

FUNDAÇÃO GETULIO VARGAS

ANALISE EXPLORATÓRIA DE DADOS E VISUALIZAÇÃO

LUIS ANTONIO ALVES

São Paulo
Abril 2021

FAMILIARIZAÇÃO COM O DPLYR

Utilize a base imdb nos exercícios a seguir.

1. Teste aplicar a **função** `glimpse()` do pacote `{dplyr}` à base imdb. O que ela faz?

```
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", ~
```

2. Crie uma tabela com apenas as colunas titulo, diretor, e orcamento. Salve em um objeto chamado `imdb_simples`.

```
imdb_simples <-imdb %>%
  select(titulo, diretor, orcamento)
imdb_simples

## # A tibble: 3,807 x 3
##   titulo                                diretor      orcamento
##   <chr>                                <chr>          <int>
## 1 Avatar                                James Cameron  237000000
## 2 Pirates of the Caribbean: At World's End Gore Verbinski 300000000
## 3 The Dark Knight Rises                 Christopher Nolan 250000000
## 4 John Carter                           Andrew Stanton 263700000
## 5 Spider-Man 3                           Sam Raimi      258000000
## 6 Tangled                               Nathan Greno   260000000
## 7 Avengers: Age of Ultron               Joss Whedon    250000000
## 8 Batman v Superman: Dawn of Justice    Zack Snyder    250000000
## 9 Superman Returns                      Bryan Singer   209000000
## 10 Pirates of the Caribbean: Dead Man's Chest Gore Verbinski 225000000
## # ... with 3,797 more rows
```

3. Selecione apenas as colunas ator_1, ator_2 e ator_3 usando o ajudante `contains()`.

```
imdb %>%
  select(contains("ator"))
```

```
## # A tibble: 3,807 x 3
##   ator_1      ator_2      ator_3
##   <chr>      <chr>      <chr>
## 1 CCH Pounder  Joel David Moore Wes Studi
## 2 Johnny Depp  Orlando Bloom  Jack Davenport
## 3 Tom Hardy    Christian Bale  Joseph Gordon-Levitt
## 4 Daryl Sabara Samantha Morton  Polly Walker
## 5 J.K. Simmons James Franco    Kirsten Dunst
## 6 Brad Garrett Donna Murphy    M.C. Gainey
## 7 Chris Hemsworth Robert Downey Jr. Scarlett Johansson
## 8 Henry Cavill Lauren Cohan     Alan D. Purwin
## 9 Kevin Spacey Marlon Brando    Frank Langella
## 10 Johnny Depp  Orlando Bloom  Jack Davenport
## # ... with 3,797 more rows
```

4. Usando a função `select()` (e seus ajudantes), escreva códigos que retornem a base IMDB sem as colunas ator_1, ator_2 e ator_3. Escreva todas as soluções diferentes que você conseguir pensar:

a)

```
imdb %>%
  select(-contains("ator"))
```

```
## # A tibble: 3,807 x 12
##   titulo      ano diretor duracao cor generos pais classificacao orcamento
##   <chr>      <int> <chr>      <int> <chr> <chr>      <chr> <chr>          <int>
## 1 Avatar      2009 James C~    178 Color Action|A~ USA A partir de ~ 237000000
## 2 Pirates~    2007 Gore Ve~    169 Color Action|A~ USA A partir de ~ 300000000
## 3 The Dar~    2012 Christo~    164 Color Action|T~ USA A partir de ~ 250000000
## 4 John Ca~    2012 Andrew ~    132 Color Action|A~ USA A partir de ~ 263700000
## 5 Spider~~    2007 Sam Rai~    156 Color Action|A~ USA A partir de ~ 258000000
## 6 Tangled     2010 Nathan ~    100 Color Adventur~ USA Livre          260000000
## 7 Avenger~    2015 Joss Wh~    141 Color Action|A~ USA A partir de ~ 250000000
## 8 Batman ~    2016 Zack Sn~    183 Color Action|A~ USA A partir de ~ 250000000
## 9 Superma~    2006 Bryan S~    169 Color Action|A~ USA A partir de ~ 209000000
## 10 Pirates~    2006 Gore Ve~    151 Color Action|A~ USA A partir de ~ 225000000
## # ... with 3,797 more rows, and 3 more variables: receita <int>,
## #   nota_imdb <dbl>, likes_facebook <int>
```

b)

```
imdb %>%  
  select(-starts_with("ator"))
```

```
## # A tibble: 3,807 x 12  
##   titulo      ano diretor duracao cor generos pais classificacao orcamento  
##   <chr>    <int> <chr>    <int> <chr> <chr>    <chr> <chr>          <int>  
## 1 Avatar      2009 James C~    178 Color Action|A~ USA A partir de ~ 237000000  
## 2 Pirates~    2007 Gore Ve~    169 Color Action|A~ USA A partir de ~ 300000000  
## 3 The Dar~    2012 Christo~    164 Color Action|T~ USA A partir de ~ 250000000  
## 4 John Ca~    2012 Andrew ~    132 Color Action|A~ USA A partir de ~ 263700000  
## 5 Spider~~    2007 Sam Rai~    156 Color Action|A~ USA A partir de ~ 258000000  
## 6 Tangled    2010 Nathan ~    100 Color Adventur~ USA Livre          260000000  
## 7 Avenger~    2015 Joss Wh~    141 Color Action|A~ USA A partir de ~ 250000000  
## 8 Batman ~    2016 Zack Sn~    183 Color Action|A~ USA A partir de ~ 250000000  
## 9 Superma~    2006 Bryan S~    169 Color Action|A~ USA A partir de ~ 209000000  
## 10 Pirates~    2006 Gore Ve~    151 Color Action|A~ USA A partir de ~ 225000000  
## # ... with 3,797 more rows, and 3 more variables: receita <int>,  
## #   nota_imdb <dbl>, likes_facebook <int>
```

c)

```
imdb %>%  
  select(-ator_1, -ator_2, -ator_3)
```

```
## # A tibble: 3,807 x 12  
##   titulo      ano diretor duracao cor generos pais classificacao orcamento  
##   <chr>    <int> <chr>    <int> <chr> <chr>    <chr> <chr>          <int>  
## 1 Avatar      2009 James C~    178 Color Action|A~ USA A partir de ~ 237000000  
## 2 Pirates~    2007 Gore Ve~    169 Color Action|A~ USA A partir de ~ 300000000  
## 3 The Dar~    2012 Christo~    164 Color Action|T~ USA A partir de ~ 250000000  
## 4 John Ca~    2012 Andrew ~    132 Color Action|A~ USA A partir de ~ 263700000  
## 5 Spider~~    2007 Sam Rai~    156 Color Action|A~ USA A partir de ~ 258000000  
## 6 Tangled    2010 Nathan ~    100 Color Adventur~ USA Livre          260000000  
## 7 Avenger~    2015 Joss Wh~    141 Color Action|A~ USA A partir de ~ 250000000  
## 8 Batman ~    2016 Zack Sn~    183 Color Action|A~ USA A partir de ~ 250000000  
## 9 Superma~    2006 Bryan S~    169 Color Action|A~ USA A partir de ~ 209000000  
## 10 Pirates~    2006 Gore Ve~    151 Color Action|A~ USA A partir de ~ 225000000  
## # ... with 3,797 more rows, and 3 more variables: receita <int>,  
## #   nota_imdb <dbl>, likes_facebook <int>
```

5. Ordene os filmes em ordem crescente de ano e decrescente de receita e salve em um objeto chamado `filmes_ordenados`.

```
filmes_ordenados <-imdb %>%
  arrange(ano, desc(receita))
filmes_ordenados
```

```
## # A tibble: 3,807 x 15
##   titulo      ano diretor duracao cor generos pais classificacao orcamento
##   <chr>      <int> <chr>    <int> <chr> <chr> <chr> <chr>          <int>
## 1 Intolera~  1916 D.W. Gr~    123 Blac~ Drama|H~ USA   Outros      385907
## 2 Over the~  1920 Harry F~    110 Blac~ Crime|D~ USA   Outros      100000
## 3 The Big ~  1925 King Vi~    151 Blac~ Drama|R~ USA   Outros      245000
## 4 The Broa~  1929 Harry B~    100 Blac~ Musical~ USA   Outros      379000
## 5 Hell's A~  1930 Howard ~     96 Blac~ Drama|W~ USA   Outros     3950000
## 6 A Farewe~  1932 Frank B~     79 Blac~ Drama|R~ USA   Outros      800000
## 7 42nd Str~  1933 Lloyd B~     89 Blac~ Comedy|~ USA   Outros      439000
## 8 She Done~  1933 Lowell ~     66 Blac~ Comedy|~ USA   Outros      200000
## 9 It Happe~  1934 Frank C~     65 Blac~ Comedy|~ USA   Outros      325000
## 10 Top Hat   1935 Mark Sa~     81 Blac~ Comedy|~ USA   Outros      609000
## # ... with 3,797 more rows, and 6 more variables: receita <int>,
## #   nota_imdb <dbl>, likes_facebook <int>, ator_1 <chr>, ator_2 <chr>,
## #   ator_3 <chr>
```

6. Selecione apenas as colunas titulo e orcamento e então ordene de forma decrescente pelo orcamento.

```
imdb %>%
  select(titulo, orcamento) %>%
  arrange(desc(orcamento))
```

```
## # A tibble: 3,807 x 2
##   titulo      orcamento
##   <chr>          <int>
## 1 Pirates of the Caribbean: At World's End 300000000
## 2 John Carter 263700000
## 3 Tangled 260000000
## 4 Spider-Man 3 258000000
## 5 Spider-Man 3 258000000
## 6 The Dark Knight Rises 250000000
## 7 Avengers: Age of Ultron 250000000
## 8 Batman v Superman: Dawn of Justice 250000000
## 9 Pirates of the Caribbean: On Stranger Tides 250000000
## 10 Captain America: Civil War 250000000
## # ... with 3,797 more rows
```

7. Crie um objeto chamado **filmes_pb** apenas com filmes preto e branco.

```
filmes_pb <-imdb %>%
  filter(cor %in% c("Black and White"))
filmes_pb
```

```
## # A tibble: 149 x 15
##   titulo      ano diretor duracao cor   generos pais classificacao orcamento
##   <chr>      <int> <chr>      <int> <chr> <chr>   <chr> <chr>          <int>
## 1 Pearl H~ 2001 Michael~    184 Black~ Action|~ USA   A partir de ~ 140000000
## 2 The Avi~ 2004 Martin ~    170 Black~ Biograp~ USA   A partir de ~ 110000000
## 3 Ali      2001 Michael~    165 Black~ Biograp~ USA   A partir de ~ 107000000
## 4 Eragon  2006 Stefen ~    104 Black~ Action|~ USA   Livre          100000000
## 5 Seabisc~ 2003 Gary Ro~    140 Black~ Drama|H~ USA   A partir de ~ 870000000
## 6 Bewitch~  NA <NA>         25 Black~ Comedy|~ USA   Outros          NA
## 7 Nine     2009 Rob Mar~    118 Black~ Drama|M~ USA   A partir de ~ 800000000
## 8 The Lea~ 2003 Stephen~    110 Black~ Action|~ USA   A partir de ~ 780000000
## 9 War Hor~ 2011 Steven ~    146 Black~ Drama|W~ USA   A partir de ~ 660000000
## 10 Space C~ 2000 Clint E~    130 Black~ Action|~ USA   A partir de ~ 650000000
## # ... with 139 more rows, and 6 more variables: receita <int>, nota_imdb <dbl>,
## # likes_facebook <int>, ator_1 <chr>, ator_2 <chr>, ator_3 <chr>
```

8. Crie um objeto chamado **curtos_legais** com filmes de 90 minutos ou menos de duração e nota no imdb maior do que 8.5.

```
curtos_legais <-imdb %>%
  filter(duracao <= 90, nota_imdb > 8.5) %>%
  select(duracao, nota_imdb)
curtos_legais
```

```
## # A tibble: 13 x 2
##   duracao nota_imdb
##   <int>      <dbl>
## 1     44      8.6
## 2     54      8.8
## 3     44      8.7
## 4     30      8.7
## 5     22      8.8
## 6     55      8.6
## 7     53      9
## 8     88      8.6
## 9     87      8.7
## 10    90      9.1
## 11    87      8.6
## 12    82      8.6
## 13    78      8.7
```

9. Retorne tabelas (**tibbles**) apenas com:

a. filmes coloridos anteriores a 1950;

```
imdb %>%
  filter(cor %in% c("Color"), ano < 1950) %>%
  select(titulo, cor, ano)
```

```
## # A tibble: 14 x 3
##   titulo                cor      ano
##   <chr>                <chr> <int>
## 1 Pinocchio            Color  1940
## 2 Duel in the Sun      Color  1946
## 3 Tycoon               Color  1947
## 4 State Fair           Color  1945
## 5 Gone with the Wind   Color  1939
## 6 The Pirate           Color  1948
## 7 It's a Wonderful Life Color  1946
## 8 Bathing Beauty       Color  1944
## 9 Fantasia             Color  1940
## 10 She Wore a Yellow Ribbon Color  1949
## 11 Spellbound          Color  1945
## 12 Snow White and the Seven Dwarfs Color  1937
## 13 Casablanca          Color  1942
## 14 Bambi               Color  1942
```

b. filmes do “Woody Allen” ou do “Wes Anderson”;

```
imdb %>%
  filter(ator_1 %in% c('Woody Allen', 'Wes Anderson')) %>%
  select(titulo, ator_1, ator_2, ator_3, ano)
```

```
## # A tibble: 10 x 5
##   titulo                ator_1    ator_2    ator_3    ano
##   <chr>                <chr>    <chr>    <chr>    <int>
## 1 The Curse of the Jade Scorpion Woody Al~ Elizabeth ~ Peter Ger~ 2001
## 2 Deconstructing Harry   Woody Al~ Lynn Cohen Judy Davis 1997
## 3 Small Time Crooks      Woody Al~ Jon Lovitz Michael R~ 2000
## 4 Anything Else          Woody Al~ Stockard C~ Fisher St~ 2003
## 5 Hollywood Ending      Woody Al~ Debra Mess~ Treat Wil~ 2002
## 6 New York Stories       Woody Al~ Larry David Mia Farrow 1989
## 7 Annie Hall             Woody Al~ Carol Kane Shelley D~ 1977
## 8 Sleeper               Woody Al~ John Beck  Mews Small 1973
## 9 Everything You Always Wanted to Know ~ Woody Al~ John Carra~ Lynn Redg~ 1972
## 10 Bananas               Woody Al~ Charlotte ~ Louise La~ 1971
```

- c. filmes do “Steven Spielberg” ordenados de forma decrescente por ano, mostrando apenas as colunas título e ano;

```
imdb %>%
  filter(diretor %in% c('Steven Spielberg')) %>%
  arrange(ano) %>%
  select(titulo, ano)
```

```
## # A tibble: 24 x 2
##   titulo                                ano
##   <chr>                                <int>
## 1 Jaws                                1975
## 2 Close Encounters of the Third Kind  1977
## 3 1941                                1979
## 4 Raiders of the Lost Ark            1981
## 5 E.T. the Extra-Terrestrial         1982
## 6 Indiana Jones and the Temple of Doom 1984
## 7 The Color Purple                   1985
## 8 Indiana Jones and the Last Crusade  1989
## 9 Hook                                1991
## 10 Jurassic Park                     1993
## # ... with 14 more rows
```

- d. filmes que tenham “Action” ou “Comedy” entre os seus gêneros;

```
imdb %>%
  filter(generos %in% c('Action', 'Comedy')) %>%
  select(titulo, generos, ano)
```

```
## # A tibble: 196 x 3
##   titulo                                generos  ano
##   <chr>                                <chr>   <int>
## 1 Batman & Robin                       Action  1997
## 2 The Campaign                         Comedy  2012
## 3 The Hangover Part II                 Comedy  2011
## 4 Batman Returns                       Action  1992
## 5 Grown Ups 2                         Comedy  2013
## 6 Jack and Jill                       Comedy  2011
## 7 Grown Ups                           Comedy  2010
## 8 Ted 2                               Comedy  2015
## 9 Dinner for Schmucks                  Comedy  2010
## 10 Last Man Standing                   Comedy   NA
## # ... with 186 more rows
```


e. filmes que tenham “Action” e “Comedy” entre os seus gêneros e tenha nota_imdb maior que 8;

```
imdb %>%
  filter(generos %in% c('Action', "Comedy"), nota_imdb > 8) %>%
  select(titulo, generos, nota_imdb)
```

```
## # A tibble: 5 x 3
##   titulo                                generos nota_imdb
##   <chr>                                <chr>     <dbl>
## 1 Kill Bill: Vol. 1                    Action      8.1
## 2 It's Always Sunny in Philadelphia    Comedy      8.8
## 3 Strangers with Candy                 Comedy      8.1
## 4 Dr. Strangelove or: How I Learned to Stop Worrying and Love~ Comedy      8.5
## 5 Kickboxer: Vengeance                 Action      9.1
```

f. filmes que não possuem informação tanto de receita quanto de orçamento (isto é, possuem NA em ambas as colunas).

```
imdb %>%
  filter(receita %in% c('NA'), orcamento %in% c('NA')) %>%
  select(titulo, receita, orcamento)
```

```
## # A tibble: 0 x 3
## # ... with 3 variables: titulo <chr>, receita <int>, orcamento <int>
```

10. Crie uma coluna chamada **prejuizo=(orcamento - receita)** e salve a nova tabela em um objeto chamado **imdb_prejuizo**. Em seguida, filtre apenas os filmes que deram prejuízo e ordene a tabela por ordem crescente de prejuízo.
 - a) (orcamento - receita)

```
imdb_prejuizo <-imdb %>%
  mutate(prejuizo = (orcamento - receita) ) %>%
  filter((orcamento - receita) >= 0 ) %>%
  arrange(desc(prejuizo)) %>%
  select(titulo, orcamento, receita, prejuizo)
imdb_prejuizo
```

```
## # A tibble: 1,313 x 4
##   titulo                                orcamento receita prejuizo
##   <chr>                                <int>     <int>   <int>
## 1 John Carter                    263700000 73058679 190641321
## 2 The Polar Express              165000000  665426 164334574
## 3 Battleship                     209000000 65173160 143826840
## 4 47 Ronin                       175000000 38297305 136702695
## 5 Jack the Giant Slayer           195000000 65171860 129828140
## 6 Jupiter Ascending               176000000 47375327 128624673
## 7 Mars Needs Moms                 150000000 21379315 128620685
## 8 The Lone Ranger                 215000000 89289910 125710090
## 9 Pan                             150000000 34964818 115035182
## 10 Pan                           150000000 34964818 115035182
## # ... with 1,303 more rows
```

b) (receita - orcamento)

```
imdb_prejuizo <-imdb %>%
  mutate(prejuizo = (receita - orcamento ) ) %>%
  filter((receita - orcamento ) < 0 ) %>%
  arrange(desc(prejuizo)) %>%
  select(titulo, orcamento, receita, prejuizo)
imdb_prejuizo
```

```
## # A tibble: 1,312 x 4
##   titulo                orcamento receita prejuizo
##   <chr>                  <int>    <int>    <int>
## 1 The Horse Boy          160000    155984    -4016
## 2 Newlyweds              9000      4584    -4416
## 3 Shortbus              2000000    1984378   -15622
## 4 Slam                  1000000    982214   -17786
## 5 Pink Narcissus         27000      8231   -18769
## 6 The FP                 60000     40557   -19443
## 7 Assault on Precinct 13 20000000    19976073  -23927
## 8 The Business of Fancydancing 200000    174682  -25318
## 9 Latter Days            850000    819939  -30061
## 10 Happy Christmas       70000     30084  -39916
## # ... with 1,302 more rows
```

11. Fazendo apenas uma chamada da função `mutate()`, crie as seguintes colunas novas na base imdb:

- lucro = receita - orcamento
- lucro_medio
- lucro_relativo = (lucro - lucro_medio)/lucro_medio
- houve_lucro = ifelse(lucro > 0, "sim", "não")

```
imdb %>%
  mutate(lucro = (receita - orcamento ),
         lucro_medio = mean(lucro),
         lucro_relativo = var(lucro - lucro_medio/lucro_medio),
         houve_lucro = ifelse(lucro > 0, "sim", "não")) %>%
  select(titulo, orcamento, receita, lucro, lucro_medio, lucro_relativo, houve_lucro)
```

```
## # A tibble: 3,807 x 7
##   titulo                orcamento receita  lucro lucro_medio lucro_relativo houve_lucro
##   <chr>                  <int>    <int>    <int>    <dbl>    <dbl>    <chr>
## 1 Avatar                237000000  7.61e8  5.24e8      NA      NA sim
## 2 Pirates of ~          300000000  3.09e8  9.40e6      NA      NA sim
## 3 The Dark Kn~          250000000  4.48e8  1.98e8      NA      NA sim
## 4 John Carter          263700000  7.31e7 -1.91e8      NA      NA não
## 5 Spider-Man ~          258000000  3.37e8  7.85e7      NA      NA sim
## 6 Tangled              260000000  2.01e8 -5.92e7      NA      NA não
## 7 Avengers: A~          250000000  4.59e8  2.09e8      NA      NA sim
## 8 Batman v Su~          250000000  3.30e8  8.02e7      NA      NA sim
## 9 Superman Re~          209000000  2.00e8 -8.93e6      NA      NA não
## 10 Pirates of ~          225000000  4.23e8  1.98e8      NA      NA sim
## # ... with 3,797 more rows
```

12. Calcule a **duração média e mediana** dos filmes da base.

```
imdb %>%
mutate(lucro = (receita - orcamento ),
       medio_lucro = mean(lucro, na.rm = TRUE),
       mediana_lucro = median(lucro - medio_lucro/medio_lucro, na.rm = TRUE)) %>%
select(titulo, orcamento, receita, lucro, medio_lucro, mediana_lucro)
```

```
## # A tibble: 3,807 x 6
##   titulo      orcamento receita  lucro medio_lucro mediana_lucro
##   <chr>          <int>   <int>   <int>      <dbl>      <dbl>
## 1 Avatar          237000000  7.61e8  5.24e8  17258230.  3291210.
## 2 Pirates of the Caribbe~ 300000000  3.09e8  9.40e6  17258230.  3291210.
## 3 The Dark Knight Rises  250000000  4.48e8  1.98e8  17258230.  3291210.
## 4 John Carter        263700000  7.31e7 -1.91e8  17258230.  3291210.
## 5 Spider-Man 3         258000000  3.37e8  7.85e7  17258230.  3291210.
## 6 Tangled            260000000  2.01e8 -5.92e7  17258230.  3291210.
## 7 Avengers: Age of Ultro~ 250000000  4.59e8  2.09e8  17258230.  3291210.
## 8 Batman v Superman: Daw~ 250000000  3.30e8  8.02e7  17258230.  3291210.
## 9 Superman Returns     209000000  2.00e8 -8.93e6  17258230.  3291210.
## 10 Pirates of the Caribbe~ 225000000  4.23e8  1.98e8  17258230.  3291210.
## # ... with 3,797 more rows
```

13. Calcule o **lucro médio** dos filmes com **duração menor que 60 minutos**.

```
imdb %>%
mutate(lucro = (receita - orcamento ), medio_lucro = mean(lucro, na.rm = TRUE)) %>%
  filter(duracao < 60) %>%
  select(titulo, receita, lucro, medio_lucro, duracao)
```

```
## # A tibble: 61 x 5
##   titulo      receita lucro medio_lucro duracao
##   <chr>          <int> <int>      <dbl>   <int>
## 1 Hannibal          NA    NA  17258230.    44
## 2 Daredevil          NA    NA  17258230.    54
## 3 Bewitched          NA    NA  17258230.    25
## 4 Constantine        NA    NA  17258230.    43
## 5 Life               NA    NA  17258230.    45
## 6 Sleepy Hollow      NA    NA  17258230.    45
## 7 Last Man Standing  NA    NA  17258230.    30
## 8 Rules of Engagement NA    NA  17258230.    22
## 9 Sabrina, the Teenage Witch NA    NA  17258230.    22
## 10 Sex and the City   NA    NA  17258230.    30
## # ... with 51 more rows
```

14. Apresente na mesma tabela o lucro médio dos filmes com duracao menor que 60 minutos e o lucro médio dos filmes com duracao maior ou igual a 60 minutos.

```
imdb %>%
group_by(duracao < 60, duracao > 60 ) %>%
  mutate(lucro = (receita - orcamento),
         lucro_medio = mean(lucro, na.rm = TRUE)) %>%
  select(titulo, receita, lucro, lucro_medio, duracao)

## Adding missing grouping variables: `duracao < 60`, `duracao > 60`

## # A tibble: 3,807 x 7
## # Groups:   duracao < 60, duracao > 60 [4]
##   `duracao < 60` `duracao > 60` titulo      receita    lucro lucro_medio duracao
##   <lgl>          <lgl>          <chr>         <int>    <int>    <dbl>    <int>
## 1 FALSE         TRUE          Avatar         7.61e8  5.24e8  17274472.    178
## 2 FALSE         TRUE          Pirates of~   3.09e8  9.40e6  17274472.    169
## 3 FALSE         TRUE          The Dark K~   4.48e8  1.98e8  17274472.    164
## 4 FALSE         TRUE          John Carte~   7.31e7 -1.91e8  17274472.    132
## 5 FALSE         TRUE          Spider-Man~   3.37e8  7.85e7  17274472.    156
## 6 FALSE         TRUE          Tangled       2.01e8 -5.92e7  17274472.    100
## 7 FALSE         TRUE          Avengers: ~   4.59e8  2.09e8  17274472.    141
## 8 FALSE         TRUE          Batman v S~   3.30e8  8.02e7  17274472.    183
## 9 FALSE         TRUE          Superman R~   2.00e8 -8.93e6  17274472.    169
## 10 FALSE        TRUE          Pirates of~   4.23e8  1.98e8  17274472.    151
## # ... with 3,797 more rows
```

15. Retorne tabelas (tibbles) apenas com:

a. a nota IMDB média dos filmes por tipo de classificacao;

```
imdb %>%
group_by(classificacao) %>%
  summarise(nota_media=mean(nota_imdb, na.rm = TRUE)) %>%
  select(classificacao, nota_media)

## # A tibble: 4 x 2
##   classificacao      nota_media
##   <chr>            <dbl>
## 1 A partir de 13 anos    6.18
## 2 A partir de 18 anos    6.47
## 3 Livre                6.26
## 4 Outros                6.65
```

b. a receita média e mediana dos filmes por ano;

```
imdb %>%
group_by(ano) %>% summarise(
  receita_media=mean(receita, na.rm = FALSE),
  receita_mediana=median(receita, na.rm = FALSE)) %>%
select(ano, receita_media, receita_mediana)
```

```
## # A tibble: 91 x 3
##   ano receita_media receita_mediana
##   <int>         <dbl>         <dbl>
## 1 1916             NA             NA
## 2 1920      3000000      3000000
## 3 1925             NA             NA
## 4 1929      2808000      2808000
## 5 1930             NA             NA
## 6 1932             NA             NA
## 7 1933             NA             NA
## 8 1934             NA             NA
## 9 1935      3000000      3000000
##10 1936             NA             NA
## # ... with 81 more rows
```

c. apenas o nome dos diretores com mais de 10 filmes.

```
imdb %>%
group_by(diretor)
```

```
## # A tibble: 3,807 x 15
## # Groups:   diretor [1,813]
##   titulo      ano diretor duracao cor generos pais classificacao orcamento
##   <chr>    <int> <chr>    <int> <chr> <chr>    <chr> <chr>          <int>
## 1 Avatar    2009 James C~    178 Color Action|A~ USA A partir de ~ 237000000
## 2 Pirates~  2007 Gore Ve~    169 Color Action|A~ USA A partir de ~ 300000000
## 3 The Dar~  2012 Christo~    164 Color Action|T~ USA A partir de ~ 250000000
## 4 John Ca~  2012 Andrew ~    132 Color Action|A~ USA A partir de ~ 263700000
## 5 Spider~~  2007 Sam Rai~    156 Color Action|A~ USA A partir de ~ 258000000
## 6 Tangled   2010 Nathan ~    100 Color Adventur~ USA Livre        260000000
## 7 Avenger~  2015 Joss Wh~    141 Color Action|A~ USA A partir de ~ 250000000
## 8 Batman ~  2016 Zack Sn~    183 Color Action|A~ USA A partir de ~ 250000000
## 9 Superma~  2006 Bryan S~    169 Color Action|A~ USA A partir de ~ 209000000
##10 Pirates~  2006 Gore Ve~    151 Color Action|A~ USA A partir de ~ 225000000
## # ... with 3,797 more rows, and 6 more variables: receita <int>,
## #   nota_imdb <dbl>, likes_facebook <int>, ator_1 <chr>, ator_2 <chr>,
## #   ator_3 <chr>
```

Incluindo Gráficos