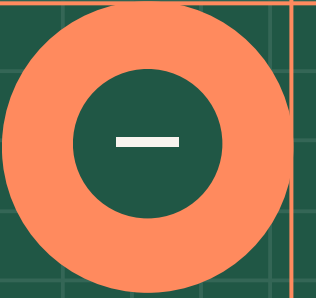
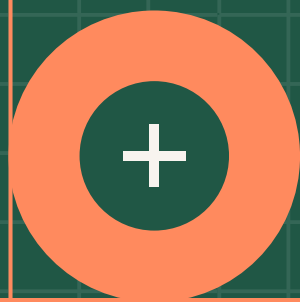


BRI0021-4 - Estatística Aplicada I

# Introdução ao

The R logo, consisting of a blue 'R' inside a grey circle.

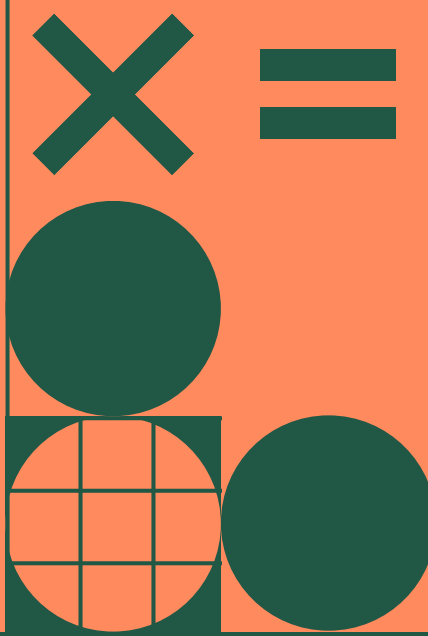
Marislei Nishijma  
Gabriela

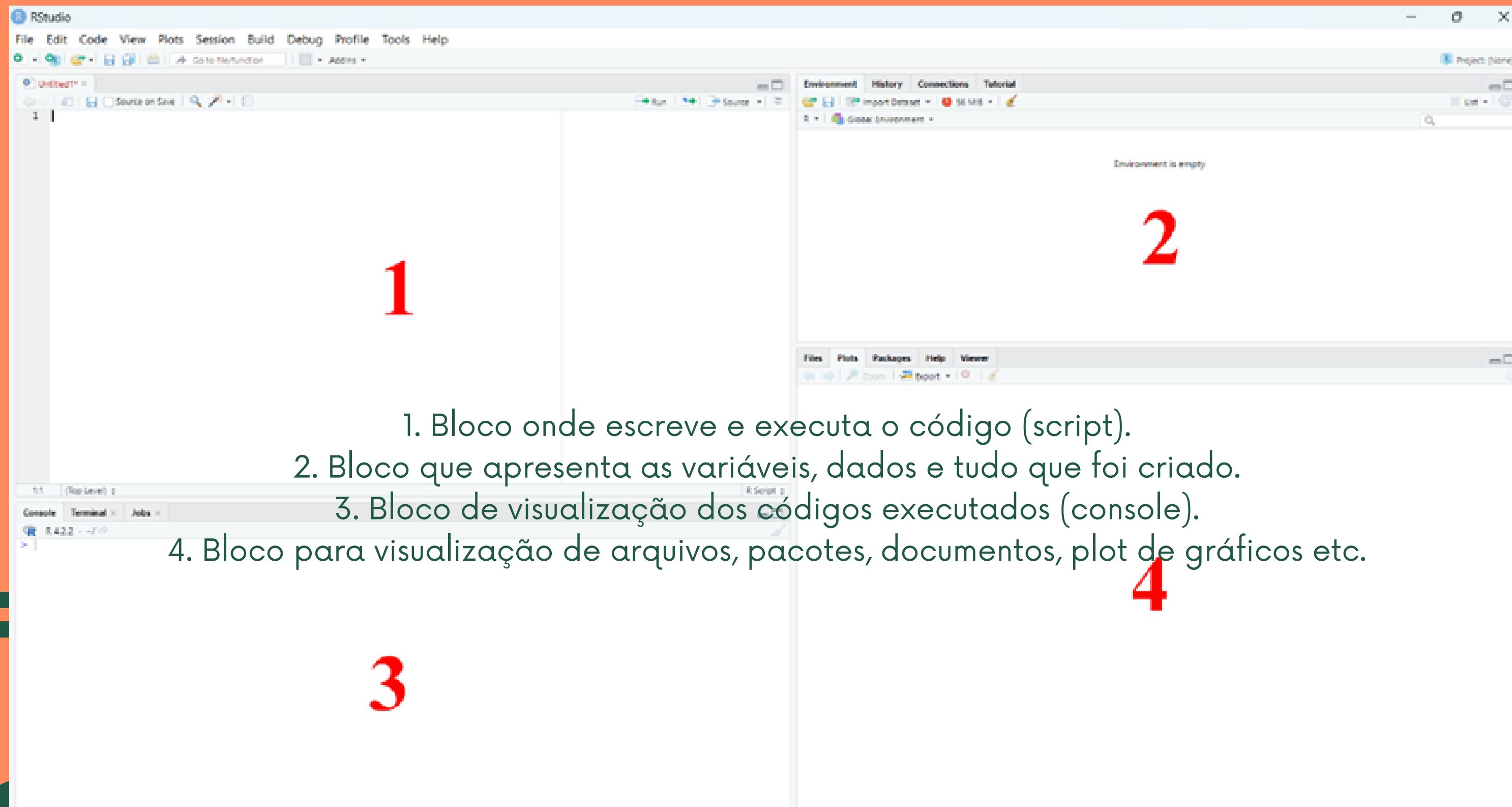


# Conhecendo o Rstudio

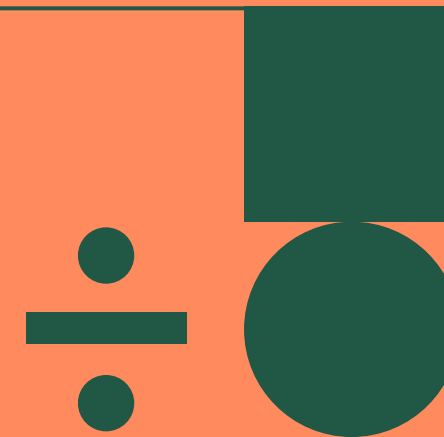


O RStudio é um ambiente de desenvolvimento integrado para R, que é uma linguagem de programação para gráficos e cálculos estatísticos

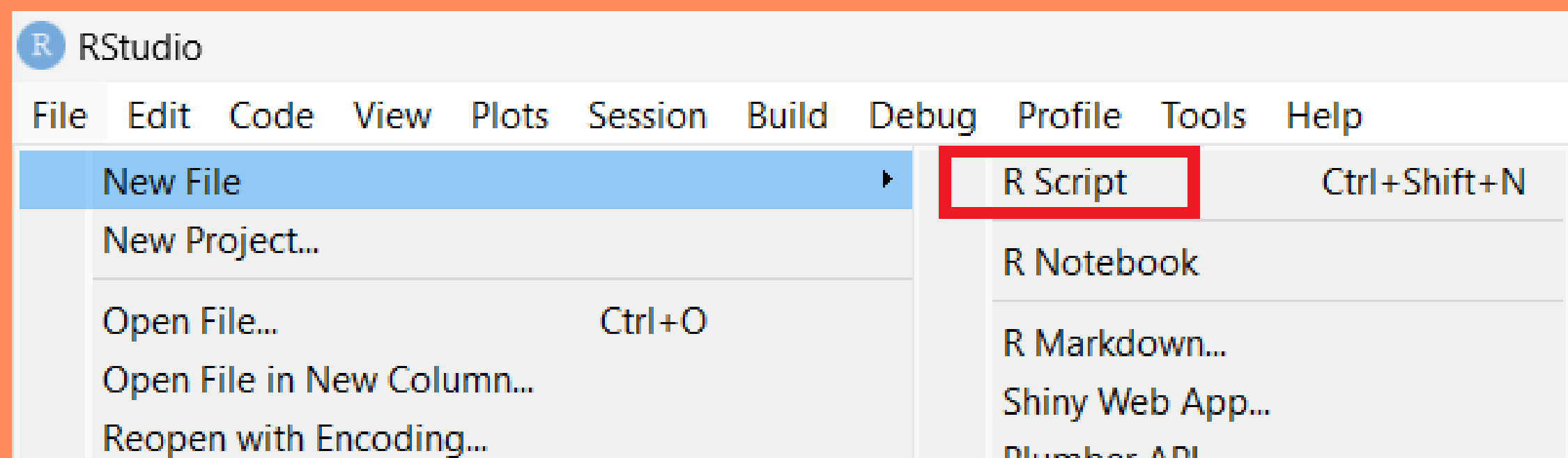
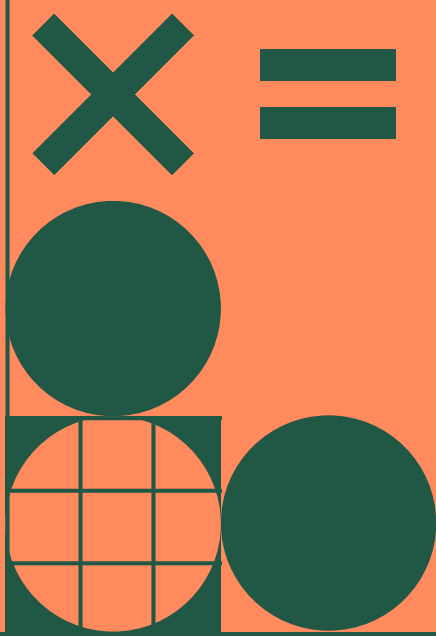


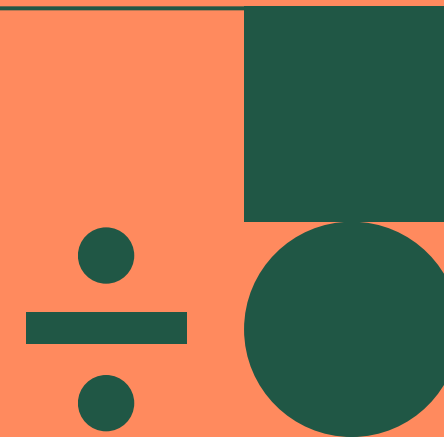


1. Bloco onde escreve e executa o código (script).
2. Bloco que apresenta as variáveis, dados e tudo que foi criado.
3. Bloco de visualização dos códigos executados (console).
4. Bloco para visualização de arquivos, pacotes, documentos, plot de gráficos etc.

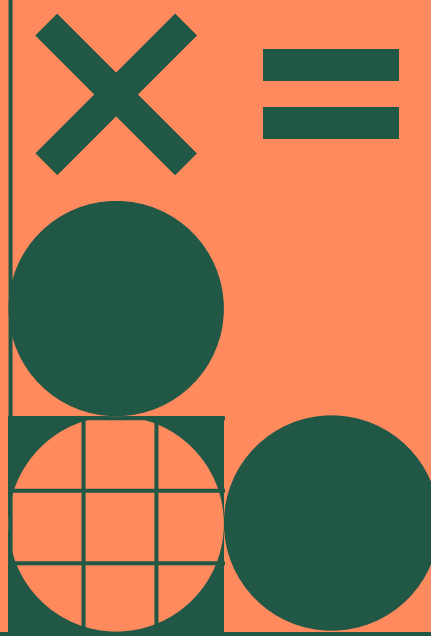


No **script** você pode digitar comandos a serem executados e também comentários, tudo que é escrito após # (jogo da velha/cerquilha) são considerados apenas como comentários.





Versão online: [posit.cloud](https://posit.cloud) (limitado)



# CÓDIGOS BÁSICOS

```
# print - imprime/retorna um argumento  
print('olá mundo!')  
## Funciona como calculadora  
# soma  
a = 2 + 2  
print(a)  
# subtração  
8 - 3  
# multiplicação  
3 * 8  
# potências  
2 ^ 8  
# prioridade de solução  
(2 + 4) / 7
```

# BIBLIOTECAS E PACOTES

```
# inclui funções como filter, select, group by, summarise, arrange, etc.  
install.packages("dplyr") # instala  
library(dplyr) # carrega
```

```
# organização dos dados  
install.packages("tidyr")  
library(tidyr)
```

```
# leitura e importação de dados  
install.packages("readr")  
library(readr)
```

```
# manipulação de strings e textos  
install.packages("stringr")  
library(stringr)
```

```
# geração de gráficos  
install.packages("ggplot2")  
library(ggplot2)
```

# FUNÇÕES

```
# logaritmo  
log(4)  
log10(4) # log na base 10  
  
# raiz quadrada  
sqrt(4)  
  
# repetir um valor  
b <- rep(x = 1, times = 4)  
  
# soma de vários valores  
sum(1, 8, 79)  
sum(b)
```



# VETOR & DATA

```
# vetor
v1 <- c(10, 20, 15, 40)
v2 <- c("a", "b", "c", NA)
v3 <- c(TRUE, FALSE, TRUE, TRUE)

# data frame = combinar dados
data1 <- data.frame(v1, v2, v3)

mean(data1$v1) # media
median(data1$v1) # mediana
sd(data1$v1) # desvio padrão
var(data1$v1) # variância
summary(data1$v1) # sumário estatístico
```

# MERGE

```
# merge - conexão entre dois conjuntos de dados
df1 <- data.frame(id = c(1, 2, 3),
                  nome = c("João", "Maria", "Pedro"))
df2 <- data.frame(id = c(2, 3, 4),
                  idade = c(25, 30, 35))

merged <- merge(df1, df2, by = "id")
View(merged)
print(merged)
```

# ARQUIVOS

```
# header em relação ao título das variáveis, por isso o TRUE,  
# e sep é o tipo de separador do arquivo (. , ; | " ")  
meus_dados <- read.csv(file = "meus_dados.csv", header=T, sep=",")  
meus_dados <- read.csv(file = "C:/Users/bibis/Downloads/dataset/movies.csv",  
                        header=T, sep=",")  
  
# remove linhas com dados faltantes/nulos  
meus_dados <- na.omit(meus_dados)  
  
sum(meus_dados$votes)  
mean(meus_dados$budget)  
min(meus_dados$budget) # mínimo  
max(meus_dados$budget) # máximo
```

# NOVOS DADOS VIA GROUP BY

```
filmes_por_ano <- meus_dados %>%  
  group_by(year) %>% # agrupamento dos dados por ano  
  summarise(total = n()) # realiza a somatória conforme o ano
```

```
filmes_por_genre <- meus_dados %>%  
  group_by(genre) %>% # agrupamento dos dados por gênero  
  summarise(total = n()) %>%  
  arrange(desc(total))
```

```
dados_agrupados <- meus_dados %>%  
  group_by(year, genre) %>% # agrupamento por ano e gênero  
  summarise(contagem = n()) # realiza a somatória de gêneros por ano
```

```
write.csv(filmes_por_genre, file = "filmes_genero.csv", row.names = FALSE)  
#salvar o arquivo
```

# GRÁFICOS

```
filmes_por_ano %>%
  ggplot(aes(x=year, y=total)) +
  geom_line(color="#836FFF") # gráfico de linha na cor escolhida (nome, hexa)

ggplot(filmes_por_genre, aes(x=genre, y=total, fill=genre)) +
  # fill é a identificação da legenda
  geom_bar(stat="identity") + # gráfico de barras
  xlab("Gêneros") +
  ylab("Número de filmes") +
  ggtitle("Filmes por gêneros")

grafico <- dados_agrupados %>%
  filter(year >= 2015) %>% # filtrar apenas filmes lançados a partir de 2015
  filter(genre %in% c("Comedy", "Animation", "Action")) %>% # apenas dos gêneros comédia,
  #animação e ação
  ggplot(aes(x = year, y = contagem)) +
  geom_line(aes(color = genre)) +
  scale_color_brewer(palette = "Set1")

plot(grafico) # plotar o gráfico

ggsave("nome_do_grafico.png", plot = grafico, dpi = 300) # salvar o gráfico
# pode escolher o formato que desejar, jpg, jpeg, pdf, entre outros
```

# Dúvidas?!

[gabriela-duarte@usp.br](mailto:gabriela-duarte@usp.br)

