



Crash Course de R

Medidas Resumo e Manipulação de dados

Prof. Carlos Trucíos

 ctruciosm.github.io

 carlos.trucios@facc.ufrj.br

Faculdade de Administração e Ciências Contábeis,
Universidade Federal de Rio de Janeiro

ctruciosm.github.io — Carlos Trucíos (FACC/UFRJ)

Medidas resumen



Medidas resumo

```
max(variavel)           # máximo
min(variavel)           # mínimo
mean(variavel)          # média
median(variavel)        # mediana
quantile(variavel, prob = k/100) # k-ésimo percentil
IQR(variavel)           # amplitude inter-quartil
var(variavel)           # variância amostral
sd(variavel)            # desvio padrão amostral
cov(variavel_1, variavel_2) # covariância
cor(variavel_1, variavel_2) # correlação de Pearson
summary(dataset_ou_variavel) # Algumas estatística resumo
table(variavel_categorica) # Frequências absolutas
prop.table(table(var_categorica)) # Frequências relativas
```

```
boxplot(variavel)       # Boxplot
hist(variavel)          # Histograma
barplot(table(variavel)) # Gráfico de barras
plot(variavel_1, variavel_2) # Gráfico de dispersão
```

Medidas resumo

Hands-on:



Utilizaremos o *dataset* `penguins` do pacote `palmerpenguins` para fazer uma análise exploratória de dados.

ctruciosm.github.io — Carlos Trucíos (FACC/UFRJ)

Manipulação de dados

Manipulação de dados



O pacote dplyr

Comando básicos:

- `%>%`: Pipe
- `mutate()`: cria novas variáveis.
- `select()`: seleciona um conjunto de variáveis.
- `filter()`: filtra casos.
- `arrange()`: ordena os dados.
- `glimpse()`: parecido com `head()`

Manipulação de dados

Importando os dados

```
uri <- "https://raw.githubusercontent.com/ctruciosm/ISLR/master/dataset/Advertising.csv"  
dados_advertising <- read.csv(uri)
```

Carregando o pacote + glimpse()

```
library(dplyr)  
glimpse(dados_advertising)
```

```
## Rows: 200  
## Columns: 5  
## $ X      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1...  
## $ TV     <dbl> 230.1, 44.5, 17.2, 151.5, 180.8, 8.7, 57.5, 120.2, 8.6, 199...  
## $ Radio  <dbl> 37.8, 39.3, 45.9, 41.3, 10.8, 48.9, 32.8, 19.6, 2.1, 2.6, 5...  
## $ Newspaper <dbl> 69.2, 45.1, 69.3, 58.5, 58.4, 75.0, 23.5, 11.6, 1.0, 21.2, 2...  
## $ Sales  <dbl> 22.1, 10.4, 9.3, 18.5, 12.9, 7.2, 11.8, 13.2, 4.8, 10.6, 8.6...
```

Manipulação de dados

Suponha que estamos interessados apenas em TV e Sales

```
dados_advertising %>%  
  select(TV,Sales)
```

```
##      TV Sales  
## 1  230.1  22.1  
## 2   44.5  10.4  
## 3   17.2   9.3  
## 4  151.5  18.5  
## 5  180.8  12.9  
## 6    8.7   7.2  
## 7   57.5  11.8  
## 8  120.2  13.2  
## 9    8.6   4.8  
## 10 199.8  10.6  
## 11  66.1   8.6  
## 12 214.7  17.4  
## 13  23.8   9.2  
## 14  97.5   9.7  
## 15 204.1  19.0  
## 16 195.4  22.4  
## 17  67.8  12.5  
## 18 281.4  24.4  
## 19  69.2  11.3  
## 20 147.3  14.6  
## 21 218.4  18.0  
## 22 237.4  12.5  
## 23  13.2   5.6  
## 24 228.3  15.5  
## 25  62.3   9.7  
## 26 262.9  12.0  
## 27 142.9  15.0  
## 28 240.1  15.9  
## 29 248.8  18.9
```


Manipulação de dados

Suponha que estamos interessados em TV, Sales e Sales²

```
dados_advertising %>%  
  select(TV,Sales) %>%  
  mutate(Sales2 = Sales^2)
```

##	TV	Sales	Sales2
## 1	230.1	22.1	488.41
## 2	44.5	10.4	108.16
## 3	17.2	9.3	86.49
## 4	151.5	18.5	342.25
## 5	180.8	12.9	166.41
## 6	8.7	7.2	51.84
## 7	57.5	11.8	139.24
## 8	120.2	13.2	174.24
## 9	8.6	4.8	23.04
## 10	199.8	10.6	112.36
## 11	66.1	8.6	73.96
## 12	214.7	17.4	302.76
## 13	23.8	9.2	84.64
## 14	97.5	9.7	94.09
## 15	204.1	19.0	361.00
## 16	195.4	22.4	501.76
## 17	67.8	12.5	156.25
## 18	281.4	24.4	595.36
## 19	69.2	11.3	127.69
## 20	147.3	14.6	213.16
## 21	218.4	18.0	324.00
## 22	237.4	12.5	156.25
## 23	13.2	5.6	31.36
## 24	228.3	15.5	240.25
## 25	62.3	9.7	94.09
## 26	262.9	12.0	144.00
## 27	142.9	15.0	225.00
## 28	240.1	15.9	252.81

Manipulação de dados

E se quisermos TV, Sales e Sales2 para os valores nos quais as vendas foram (>15)?

```
dados_advertising %>%  
  select(TV,Sales) %>%  
  mutate(Sales2 = Sales^2) %>%  
  filter(Sales>15)
```

```
##      TV Sales Sales2  
## 1  230.1  22.1 488.41  
## 2  151.5  18.5 342.25  
## 3  214.7  17.4 302.76  
## 4  204.1  19.0 361.00  
## 5  195.4  22.4 501.76  
## 6  281.4  24.4 595.36  
## 7  218.4  18.0 324.00  
## 8  228.3  15.5 240.25  
## 9  240.1  15.9 252.81  
## 10 248.8  18.9 357.21  
## 11 292.9  21.4 457.96  
## 12 265.6  17.4 302.76  
## 13 266.9  25.4 645.16  
## 14 228.0  21.5 462.25  
## 15 202.5  16.6 275.56  
## 16 177.0  17.1 292.41  
## 17 293.6  20.7 428.49  
## 18 239.9  23.2 538.24  
## 19 216.4  22.6 510.76  
## 20 182.6  21.2 449.44  
## 21 262.7  20.2 408.04  
## 22 198.9  23.7 561.69  
## 23 210.8  23.8 566.44  
## 24 210.7  18.4 338.56  
## 25 261.3  24.2 585.64  
## 26 239.3  15.7 246.49  
## 27 131.1  18.0 324.00
```

Manipulação de dados

E se quisermos os valores ordenados (de menor a maior) por Sales?

```
dados_advertising %>%  
  select(TV,Sales) %>%  
  mutate(Sales2 = Sales^2) %>%  
  filter(Sales>15) %>%  
  arrange(Sales)
```

```
##      TV Sales Sales2  
## 1  193.2  15.2 231.04  
## 2  123.1  15.2 231.04  
## 3   93.9  15.3 234.09  
## 4  228.3  15.5 240.25  
## 5  184.9  15.5 240.25  
## 6  141.3  15.5 240.25  
## 7  187.8  15.6 243.36  
## 8  239.3  15.7 246.49  
## 9  240.1  15.9 252.81  
## 10 209.6  15.9 252.81  
## 11 125.7  15.9 252.81  
## 12 286.0  15.9 252.81  
## 13 110.7  16.0 256.00  
## 14 280.7  16.1 259.21  
## 15 202.5  16.6 275.56  
## 16 197.6  16.6 275.56  
## 17 109.8  16.7 278.89  
## 18 163.3  16.9 285.61  
## 19 213.4  17.0 289.00  
## 20 177.0  17.1 292.41  
## 21 215.4  17.1 292.41  
## 22 135.2  17.2 295.84  
## 23 191.1  17.3 299.29  
## 24 149.7  17.3 299.29  
## 25 214.7  17.4 302.76  
## 26 265.6  17.4 302.76
```

Manipulação de dados

E se quisermos ordenados de maior a a menor?

```
dados_advertising %>%  
  select(TV,Sales) %>%  
  mutate(Sales2 = Sales^2) %>%  
  filter(Sales>15) %>%  
  arrange(desc(Sales))
```

```
##      TV Sales Sales2  
## 1  276.9  27.0 729.00  
## 2  287.6  26.2 686.44  
## 3  283.6  25.5 650.25  
## 4  266.9  25.4 645.16  
## 5  289.7  25.4 645.16  
## 6  243.2  25.4 645.16  
## 7  220.3  24.7 610.09  
## 8  281.4  24.4 595.36  
## 9  261.3  24.2 585.64  
## 10 210.8  23.8 566.44  
## 11 296.4  23.8 566.44  
## 12 198.9  23.7 561.69  
## 13 239.9  23.2 538.24  
## 14 216.4  22.6 510.76  
## 15 205.0  22.6 510.76  
## 16 195.4  22.4 501.76  
## 17 216.8  22.3 497.29  
## 18 250.9  22.2 492.84  
## 19 230.1  22.1 488.41  
## 20 241.7  21.8 475.24  
## 21 213.5  21.7 470.89  
## 22 228.0  21.5 462.25  
## 23 292.9  21.4 457.96  
## 24 182.6  21.2 449.44  
## 25 273.7  20.8 432.64  
## 26 293.6  20.7 428.49
```

Manipulação de dados

- O `%>%` nos ajuda a fazer nosso código mais fácil de entender.
- Se formos quebrar o código em várias linhas, o `%>%` deve ir **sempre** no final da linha.

Podemos também calcular algumas estatísticas

```
dados_advertising %>%  
  select(TV,Sales) %>%  
  mutate(Sales2 = Sales^2) %>%  
  filter(Sales>15) %>%  
  arrange(desc(Sales)) %>%  
  summarise(media_TV = mean(TV), media_Sales = mean(Sales))
```

```
##   media_TV media_Sales  
## 1 213.9013    19.61067
```

Manipulação de dados

Hands-on

Utilize o dataset `dados_advertising`, filtre os dados para considerarmos unicamente os casos em que `Sales <= median(Sales)`, selecione apenas as variáveis `TV`, `Radio` e `Newspaper`, calcule a média e desvio padrão desses dados.

Gabarito

```
dados_advertising %>%  
  filter(Sales<= median(Sales)) %>%  
  select(TV, Radio, Newspaper) %>%  
  summarise(media_tv = mean(TV), sd_tv = sd(TV),  
            media_radio = mean(Radio), sd_radio = sd(Radio),  
            media_paper = mean(Newspaper), sd_paper = sd(Newspaper))
```

```
##   media_tv   sd_tv media_radio sd_radio media_paper sd_paper  
## 1 94.07843 75.12462   15.53627 13.58036   27.88529 20.76878
```

Cuidado com a ordem dos comandos!



Manipulação de dados

Gabarito

```
# Este código da erro! Por quê?  
dados_advertising %>%  
  select(TV, Radio, Newspaper) %>%  
  filter(Sales<= median(Sales)) %>%  
  summarise(media_tv = mean(TV), sd_tv = sd(TV),  
            media_radio = mean(Radio), sd_radio = sd(Radio),  
            media_paper = mean(Newspaper), sd_paper = sd(Newspaper))
```

Por quê?



Manipulação de dados

Mais comandos:

- summarise()
- top_n()
- group_by()
- contains()
- rename()

Manipulação de dados

Podemos calcular estatísticas por grupos.

```
dados_advertising %>%  
  select(TV, Sales, Radio, Newspaper) %>%  
  group_by(Sales > median(Sales)) %>%  
  summarise(media_TV = mean(TV),  
            media_Radio = mean(Radio),  
            mean_Newspaper = mean(Newspaper))
```

```
## # A tibble: 2 × 4  
##   `Sales > median(Sales)` media_TV media_Radio mean_Newspaper  
##   <lgl>                <dbl>      <dbl>          <dbl>  
## 1 FALSE                 94.1       15.5           27.9  
## 2 TRUE                  202.       31.3           33.3
```

Manipulação de dados

Se quisermos as 5 lojas com mais vendas?

```
dados_advertising %>%  
  select(TV, Sales, Radio, Newspaper) %>%  
  top_n(5, Sales)
```

##	TV	Sales	Radio	Newspaper
## 1	266.9	25.4	43.8	5.0
## 2	289.7	25.4	42.3	51.2
## 3	243.2	25.4	49.0	44.3
## 4	276.9	27.0	48.9	41.8
## 5	287.6	26.2	43.0	71.8
## 6	283.6	25.5	42.0	66.2

Manipulação de dados

Se quisermos as 5 lojas com menos vendas?

```
dados_advertising %>%  
  select(TV, Sales, Radio, Newspaper) %>%  
  top_n(-5, Sales)
```

##	TV	Sales	Radio	Newspaper
## 1	8.6	4.8	2.1	1.0
## 2	5.4	5.3	29.9	9.4
## 3	13.1	5.3	0.4	25.6
## 4	0.7	1.6	39.6	8.7
## 5	4.1	3.2	11.6	5.7

Manipulação de dados

Podemos selecionar variáveis por alguma característica em especial:

```
dados_advertising %>%  
  select(contains("a")) %>%  
  head()
```

```
##   Radio Newspaper Sales  
## 1  37.8         69.2  22.1  
## 2  39.3         45.1  10.4  
## 3  45.9         69.3   9.3  
## 4  41.3         58.5  18.5  
## 5  10.8         58.4  12.9  
## 6  48.9         75.0   7.2
```

Manipulação de dados

Podemos renomear as variáveis

```
dados_advertising %>%  
  rename(radio_gastos = Radio,  
         newspaper_gastos = Newspaper,  
         tv_gastos = TV) %>%  
  head()
```

```
##   X tv_gastos radio_gastos newspaper_gastos Sales  
## 1 1   230.1      37.8          69.2      22.1  
## 2 2    44.5      39.3          45.1      10.4  
## 3 3    17.2      45.9          69.3       9.3  
## 4 4   151.5      41.3          58.5      18.5  
## 5 5   180.8      10.8          58.4      12.9  
## 6 6     8.7      48.9          75.0       7.2
```

Manipulação de dados

Hands-on

Utilize o *dataset* `dados_advertising` e:

1. Apague a coluna X (Dica: `select(-X)`)
2. Calcule a média (TV, Radio e Newspaper) e o número de elementos por grupo (`Sales > mean(Sales)`). Dica: para calcular o número de elementos use `n()`.

Gabarito

```
dados_advertising %>%  
  select(-X) %>%  
  group_by(Sales > mean(Sales)) %>%  
  summarise(m_tv = mean(TV),  
            m_radio = mean(Radio),  
            m_newspaper = mean(Newspaper),  
            Total = n())
```