

Métodos Quantitativos

Aula 05. Estatísticas descritivas no R

Pedro H. G. Ferreira de Souza

pedro.ferreira@ipea.gov.br

Mestrado Profissional em Políticas Públicas e Desenvolvimento

Instituto de Pesquisa Econômica Aplicada (Ipea)

17 out. 2022

Recapitação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Aula passada

- Boas práticas para organização de projetos
- Scripts, pacotes e funções no R
- Classes mais importantes de objetos atômicos
 - Character, logical, integer e numeric
- Classes mais comuns de coleções de objetos
 - Vetores, matrizes e data frames (ou tibbles)
- Manipulação de dados com o `dplyr`
 - filter, select, arrange, mutate
 - (**Atividade #3**) summarise e group_by

Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Introdução

Objetivos de hoje

- Consolidar lições da aula passada
- Apresentar (brevemente) mais funções úteis do R
- Estatísticas descritivas uni- e bivariadas no R

Preliminares

- No **RStudio**, criar novo projeto em uma pasta vazia
- No **Github**, baixar o zip com o material de apoio da aula 04 e descompactar o arquivo na pasta do projeto

Pacotes que usaremos hoje

Já conhecidos: here, tidyverse, summarytools e gapminder

Pacotes novos: nycflights13, scales, writexl

Exercício: carregar os seis pacotes no script de vocês.

Pacotes que usaremos hoje

Já conhecidos: here, tidyverse, summarytools e gapminder

Pacotes novos: nycflights13, scales, writexl

Exercício: carregar os seis pacotes no script de vocês.

```
library(here)
library(tidyverse)
library(summarytools)
library(gapminder)
library(nycflights13)
library(scales)
library(writexl)
Sys.setlocale("LC_ALL","pt_BR.UTF-8") # para corrigir acentos
```

Lembrem-se que os pacotes precisam ser instalados antes do primeiro uso!

Data frames que usaremos hoje

Usaremos vários bancos de dados diferentes. Dois deles nós já vimos:

- Oxfam/Datafolha, que vimos na aula passada
- Gapminder, que vimos na atividade #3

Além disso, usaremos quatro data frames do pacote `nycflights13`:

- `flights`
- `weather`

Os dicionários de dados dessas bases podem ser acessados do console com `?NOME` ou `help(NOME)`

Data frames que usaremos hoje

Exercício: carregar logo os data frames no workspace para poupar trabalho depois.

Data frames que usaremos hoje

Exercício: carregar logo os data frames no workspace para poupar trabalho depois.

```
oxfam.df <- read_csv(...)
```

```
gapminder.df <- ...
```

```
voos.df <- ...
```

```
clima.df <- ...
```

Data frames que usaremos hoje

Exercício: carregar logo os data frames no workspace para poupar trabalho depois.

```
oxfam.df <- read_csv(here("dados", "brutos",
                            "datafolha_oxfam.csv"))

gapminder.df <- gapminder

voos.df <- flights

clima.df <- weather
```

Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Mais sobre o dplyr

Na aula passada, exploramos quatro das principais funções do pacote `dplyr`, incluído no pacote `tidyverse`:

`filter()` para selecionar linhas

`select()` para selecionar colunas

`arrange()` para reordenar os dados de acordo com uma ou mais colunas

`mutate()` para criar ou modificar colunas

Também vimos como encadear funções com o operador `pipe %>%`.

Na atividade #3, vocês aprenderam mais dois comandos do `dplyr`:

`group_by()` para agrupar linhas por categorias de uma coluna

`summarise()` para calcular estatísticas de colunas selecionadas

O `group_by` no `dplyr`

Exemplo: `filter`, `group_by` e `mutate` para calcular a renda média, mínima e máxima por continente e a renda relativa dos países em 2007

O group_by no dplyr

Exemplo: **filter**, **group_by** e **mutate** para calcular a renda média, mínima e máxima por continente e a renda relativa dos países em 2007

```
renda.relativa.df <-  
  gapminder.df %>%  
    filter(...) %>%  
    group_by(...) %>%  
    mutate(media_continente = ....,  
          minimo_continente = ....,  
          maximo_continente = ....,  
          relativa = ...) %>%  
    arrange(...)
```

O `group_by` no `dplyr`

Exemplo: `filter`, `group_by` e `mutate` para calcular a renda média, mínima e máxima por continente e a renda relativa dos países em 2007

```
renda.relativa.df <-  
  gapminder.df %>%  
    filter(year == 2007) %>%  
    group_by(continent) %>%  
      mutate(media_continente = mean(gdpPercap),  
             minima_continente = min(gdpPercap),  
             maxima_continente = max(gdpPercap),  
             relativa = gdpPercap /  
                         media_continente) %>%  
    arrange(continent, desc(relativa))
```

O **summarise** no **dplyr**

Exemplo: use **filter**, **group_by** e **summarise** para obter o número de países, a expectativa de vida média e os valores máximo e mínimo por continente em 2007

O summarise no dplyr

Exemplo: use **filter**, **group_by** e **summarise** para obter o número de países, a expectativa de vida média e os valores máximo e mínimo por continente em 2007

```
expvida.df <-  
  gapminder.df %>%  
    filter(year == 2007) %>%  
    group_by(continent) %>%  
    summarise(n = n(),  
              media = mean(lifeExp),  
              min = min(lifeExp),  
              max = max(lifeExp))
```

O summarise no dplyr

Exemplo: use **filter**, **group_by** e **summarise** para obter o número de países, a expectativa de vida média e os valores máximo e mínimo por continente em 2007

```
print(expvida.df)

## # A tibble: 5 x 5
##   continent     n  media    min    max
##   <fct>     <int> <dbl> <dbl> <dbl>
## 1 Africa       52  54.8  39.6  76.4
## 2 Americas     25  73.6  60.9  80.7
## 3 Asia          33  70.7  43.8  82.6
## 4 Europe        30  77.6  71.8  81.8
## 5 Oceania       2  80.7  80.2  81.2
```

Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Variáveis categóricas

Tabelas de Frequência

`oxfam.df$p14`: Considerando sua renda e padrão de vida, VOCÊ se considera em qual dos seguintes grupos?

Como montar esse código?

```
oxfam.df %>% ...
```

Dica: podemos usar as funções `freq` ou `summarise`, mas a primeira é mais simples.

Variáveis categóricas

Tabelas de Frequência

```
oxfam.df %>% freq(p14)
```

##	##	Freq	% Valid	% Valid Cum.	% Total
<hr/>					
<hr/>					
##	Classe média	681	32.79	32.79	32.65
##	Classe média alta	45	2.17	34.95	2.16
##	Classe média baixa	1014	48.82	83.77	48.61
##	Pobre	331	15.94	99.71	15.87
##	Rico	6	0.29	100.00	0.29
##	<NA>	9			0.43
##	Total	2086	100.00	100.00	100.00

Variáveis ordinais

Problema: as categorias saíram em ordem alfabética, não na ordem natural. Para resolver, é só transformar a variável character em factor:

```
ordem <- c('Pobre', 'Classe média baixa',
          'Classe média', 'Classe média alta', 'Rico')

oxfam.df %>%
  mutate(p14f = factor(p14, levels = ordem,
                       ordered = TRUE)) %>%
  freq(p14f)
```

Variáveis ordinais

		Freq	% Valid	% Valid Cum.	% Total
##					
##					
##					
##	Pobre	331	15.94	15.94	15.87
##	Classe média baixa	1014	48.82	64.76	48.61
##	Classe média	681	32.79	97.54	32.65
##	Classe média alta	45	2.17	99.71	2.16
##	Rico	6	0.29	100.00	0.29
##	<NA>	9			0.43
##	Total	2086	100.00	100.00	100.00

Variáveis contínuas discretizadas

```
cat <- c("Até 18", "18-64", "65 ou mais")
oxfam.df %>%
  mutate(faixas = ifelse(idade<18, 1, ifelse(idade<65, 2, 3)),
         faixas = factor(faixas, labels = cat,
                          ordered = TRUE)) %>%
  freq(faixas)
```

Variáveis contínuas discretizadas

```
cat <- c("Até 18", "18-64", "65 ou mais")
oxfam.df %>%
  mutate(faixas = ifelse(idade<18, 1, ifelse(idade<65, 2, 3)),
         faixas = factor(faixas, labels = cat,
                           ordered = TRUE)) %>%
  freq(faixas)
```

##

	Freq	% Valid	% Valid Cum.	% Total	% Total
<hr/>					
<hr/>					
##	Até 18	82	3.93	3.93	3.93
##	18-64	1816	87.06	90.99	87.06
##	65 ou mais	188	9.01	100.00	9.01
##	<NA>	0			0.00
##	Total	2086	100.00	100.00	100.00

Função ifelse()

Sintaxe é: “ifelse(TESTE, A, B)”, em que:

- Se o teste lógico for verdadeiro, então atribui-se o valor **A**
- Se o teste lógico for falso, então atribui-se o valor **B**

Testes aninhados:

```
faixas =  
# Idade < 18? Se for, faixas == 1, senao continua...  
ifelse(idade < 18, 1,  
# Novo ifelse, somente para maiores de 18:  
# Idade < 65? Se for, faixas == 2, senao faixas == 3.  
ifelse(idade < 65, 2, 3))
```

Exercício

Nos dados do Oxfam/Datafolha, discretize a variável **p13** em grupos 0-20, 21-40, 41-60, 71-80, 81-100, converta em factor e peça a tabela de frequências. Interprete.

```
quintos <- c('0-20', '21-40', '41-60', '61-80', '81-100')
oxfam.df %>%
  mutate(... = ifelse(...,
                     ...,
                     ...,
                     ...),
         ... = factor(..., labels = quintos)) %>%
  ...(...)
```

Exercício

Nos dados do Oxfam/Datafolha, discretize a variável **p13** em grupos 0-20, 21-40, 41-60, 71-80, 81-100, converta em factor e peça a tabela de frequências. Interprete.

Exercício

Nos dados do Oxfam/Datafolha, discretize a variável **p13** em grupos 0-20, 21-40, 41-60, 71-80, 81-100, converta em factor e peça a tabela de frequências. Interprete.

```
quintos <- c('0-20', '21-40', '41-60', '61-80', '81-100')
oxfam.df <-
  oxfam.df %>%
  mutate(p13d = ifelse(p13 <= 20, 1,
                      ifelse(p13 <= 40, 2,
                            ifelse(p13 <= 60, 3,
                                  ifelse(p13 <= 80, 4, 5)))),
        p13d = factor(p13d, labels = quintos,
                      ordered = TRUE))
oxfam.df %>% freq(p13d)
```

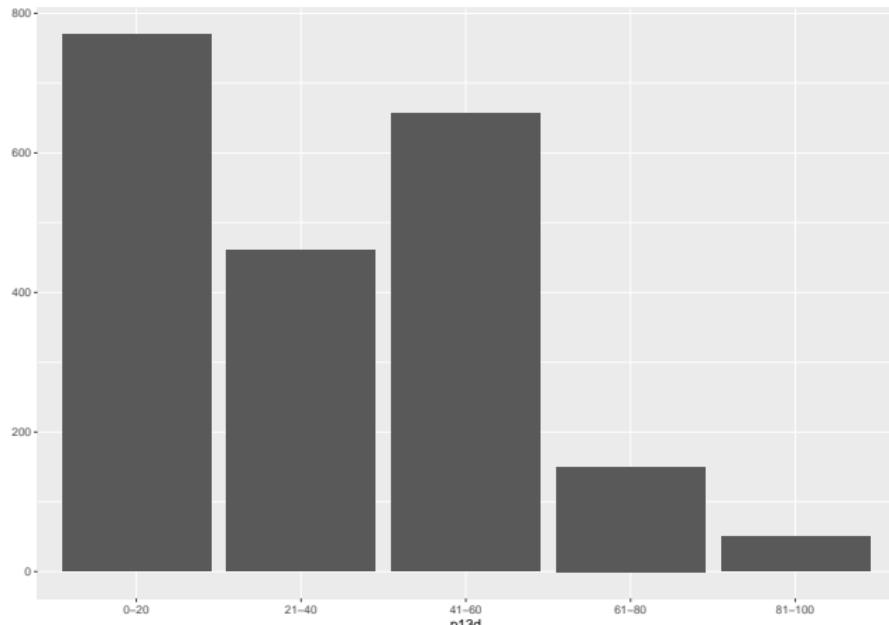
Exercício

##	##	Freq	% Valid	% Valid Cum.	% Total	% To
##	##	-----	-----	-----	-----	-----
##	##	-----	-----	-----	-----	-----
##	0-20	770	36.91	36.91	36.91	36.91
##	21-40	460	22.05	58.96	22.05	
##	41-60	656	31.45	90.41	31.45	
##	61-80	150	7.19	97.60	7.19	
##	81-100	50	2.40	100.00	2.40	
##	<NA>	0			0.00	
##	Total	2086	100.00	100.00	100.00	

Gráficos de barras simples

Frequência absoluta

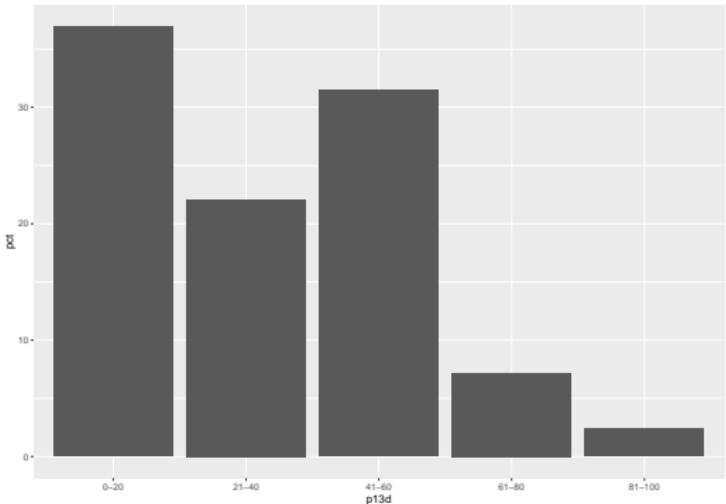
```
qplot(data = oxfam.df, x = p13d, geom="bar")
```



Gráficos de barras simples

Frequência relativa com summarise

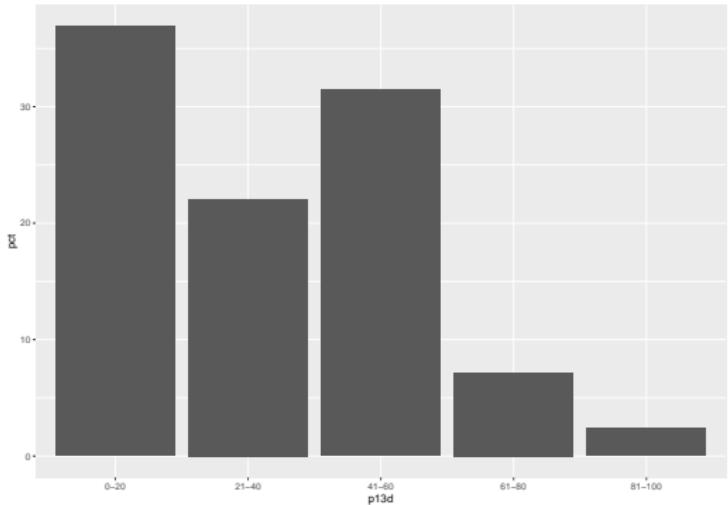
```
p13d_pct.df <- oxfam.df %>% group_by(p13d) %>%  
  summarise(n = n()) %>% mutate(pct = 100 * n / sum(n))  
qplot(data = p13d_pct.df, x = p13d, y = pct, geom="col")
```



Gráficos de barras simples

Frequência relativa com freq

```
p13d_pct.df <- oxfam.df %>% freq(p13d) %>% tb (na.rm = TRUE)  
qplot(data = p13d_pct.df, x = p13d, y = pct, geom="col")
```



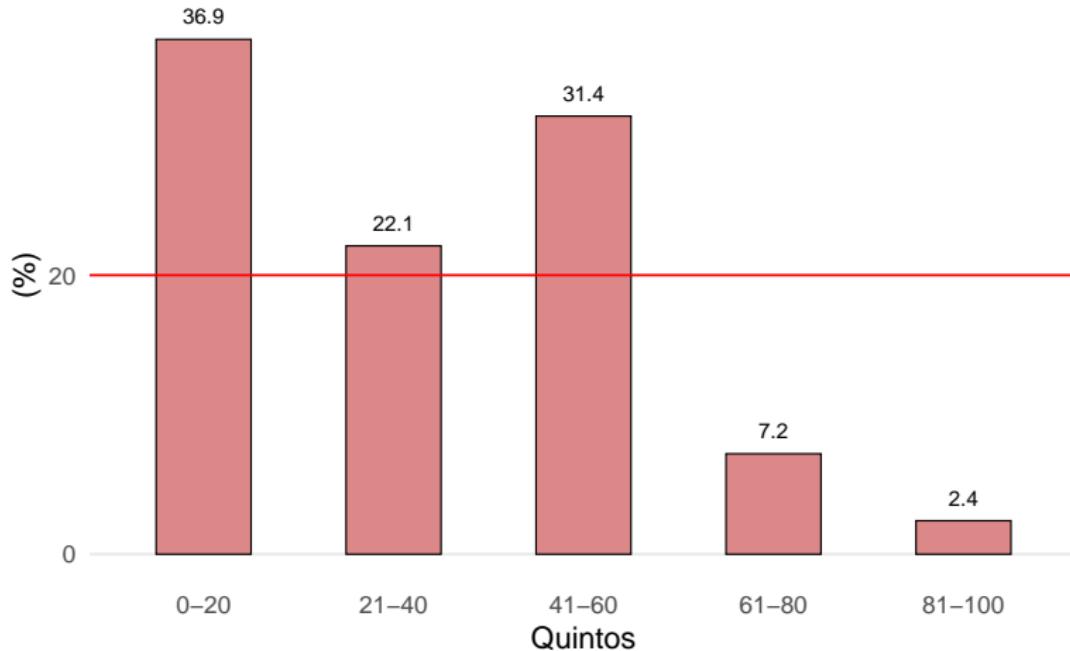
Gráficos de barras melhores (opcional)

Frequência relativa

```
oxfam.df %>%
  group_by(p13d) %>%
  summarise(n = n()) %>%
  mutate(pct = round(100 * n / sum(n), 1)) %>%
  ggplot(aes(x = p13d, y = pct)) +
  ggttitle("Frequencias relativas (%)") +
  geom_col(width = 0.5, color = 'black', fill = "#DD8888") +
  geom_text(aes(label = pct, vjust = -1), size = 5) +
  geom_hline(yintercept = 20, color = 'red', size = 0.75) +
  scale_x_discrete(name = "Quintos") +
  scale_y_continuous(name = "(%)", limits = c(0,40),
                     breaks = c(0,20), minor_breaks = NULL) +
  theme_minimal(base_size = 20) +
  theme(panel.grid.major.x = element_blank())
```

Gráficos de barra melhores (opcional)

Frequencias relativas (%)



Para quem preferir gráficos no Excel...

Alternativa #1

```
oxfam.df %>%  
  group_by(p13d) %>%  
    summarise(n = n()) %>%  
    mutate(pct = 100 * n / sum(n), 1) %>%  
    write_xlsx(here('graficos', 'grafico1.xlsx'))
```

Alternativa #2

```
oxfam.df %>%  
  freq(p13d) %>%  
  tb() %>%  
  write_xlsx(here('graficos', 'grafico1.xlsx'))
```

Exercício

oxfam.df: recodifique a variável **p20** em factor e faça um gráfico de barras com a frequência relativa das respostas.

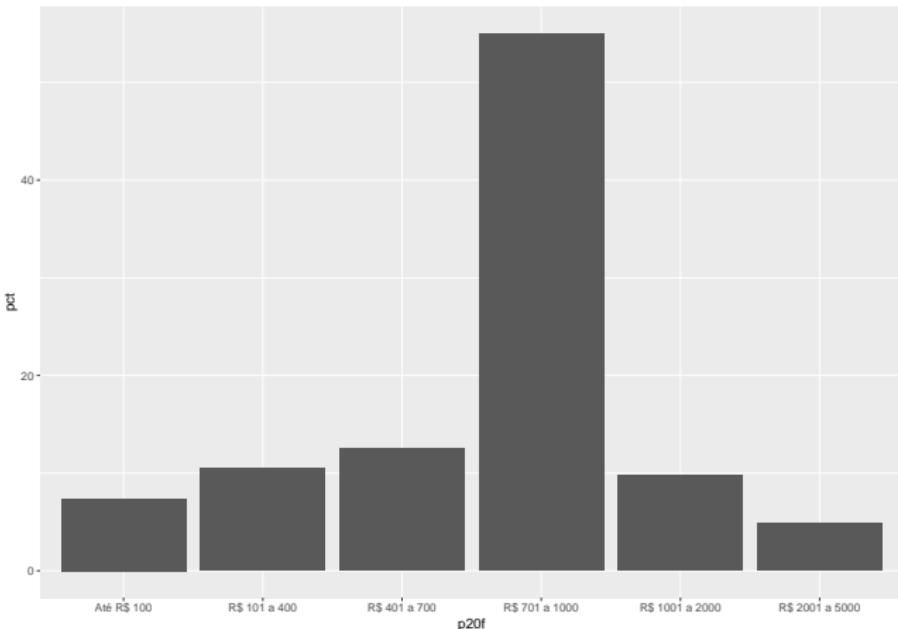
Exercício

oxfam.df: recodifique a variável **p20** em factor e faça um gráfico de barras com a frequência relativa das respostas.

```
cat <- c('Até R$ 100', 'R$ 101 a 400', 'R$ 401 a 700',
       'R$ 701 a 1000', 'R$ 1001 a 2000', 'R$ 2001 a 5000')
p20f.pct.df <- oxfam.df %>%
  mutate(p20f = factor(p20, levels = cat, ordered = TRUE)) %>%
  freq(p20f) %>% tb(na.rm = TRUE)
qplot(data = p20f.pct.df, x = p20f, y = pct, geom = 'col')
```

Exercício

oxfam.df: recodifique a variável **p20** em factor e faça um gráfico de barras com a frequência relativa das respostas.



Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Como representar uma distribuição contínua?

Tabela de frequências?

```
gapminder.df %>% filter(year == 2007) %>% freq(lifeExp, headings = FALSE)
```

	Freq	% Valid	% Valid	Cum.	% Total	% Total	Cum.
<hr/>							
##	39.613	1	0.70	0.70	0.70	0.70	0.70
##	42.082	1	0.70	1.41	0.70	0.70	1.41
##	42.384	1	0.70	2.11	0.70	0.70	2.11
##	42.568	1	0.70	2.82	0.70	0.70	2.82
##	42.592	1	0.70	3.52	0.70	0.70	3.52
##	42.731	1	0.70	4.23	0.70	0.70	4.23
##	43.487	1	0.70	4.93	0.70	0.70	4.93
##	43.828	1	0.70	5.63	0.70	0.70	5.63
##	44.741	1	0.70	6.34	0.70	0.70	6.34
##	45.678	1	0.70	7.04	0.70	0.70	7.04
##	46.242	1	0.70	7.75	0.70	0.70	7.75
##	46.388	1	0.70	8.45	0.70	0.70	8.45
##	46.462	1	0.70	9.15	0.70	0.70	9.15
##	46.859	1	0.70	9.86	0.70	0.70	9.86
##	48.159	1	0.70	10.56	0.70	0.70	10.56
##	48.303	1	0.70	11.27	0.70	0.70	11.27
##	48.328	1	0.70	11.97	0.70	0.70	11.97
##	49.339	1	0.70	12.68	0.70	0.70	12.68

Medidas-resumo mais comuns

Medidas de tendência central

- Média aritmética $\rightarrow \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
- Mediana $\rightarrow x_{\frac{n+1}{2}}$

Medidas-resumo mais comuns

Medidas de tendência central

- Média aritmética → $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
- Mediana → $x_{\frac{n+1}{2}}$

Medidas de dispersão *

- Variância → $var = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
- Desvio padrão → $sd = \sqrt{var} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$
- Coeficiente de variação → $CV = \frac{sd}{\bar{x}}$
- Desvio médio absoluto → $mad = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$
- Amplitude → $range = max(x) - min(x)$
- Amplitude interquartil → $IQR = P75 - P25$

Medidas-resumo mais comuns

Medidas de formato da distribuição *

■ Assimetria (*Skewness*) → $\frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{s^3}$

Assimetria positiva

Simetria

Assimetria negativa



Medidas-resumo mais comuns

Medidas de formato da distribuição *

- Assimetria (*Skewness*) → $\frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{s^3}$

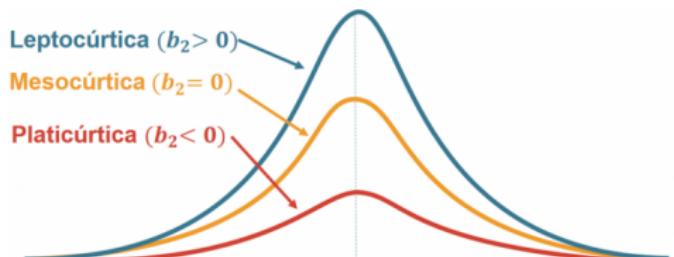
Assimetria positiva

Simetria

Assimetria negativa



- Curtose → $\frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{s^4} - 3$



Computando medidas-resumo no R

Vamos calcular estatísticas de temperatura no aeroporto JFK em Nova York.

Primeiro vamos converter a temperatura de Fahrenheit para Celsius e criar objeto só com os vôos do JFK:

```
jfk.df <- clima.df %>%  
  filter(origin == "JFK") %>%  
  mutate(temp_celsius = (temp - 32) * 5 / 9)
```

Depois, vamos calcular estatísticas de dois jeitos: com o **descr** e com o **summarise**

Computando estatísticas com descr()

```
jfk.df %>% select(temp_celsius) %>% descr(round.digits = 2)
```

Computando estatísticas com `descr()`

```
jfk.df %>% select(temp_celsius) %>% descr(round.digits = 2)

##
##          temp_celsius
##  -----
##      Mean      12.48
##      Std.Dev     9.48
##      Min     -11.10
##      Q1      4.40
##      Median    12.20
##      Q3      20.60
##      Max      36.70
##      MAD      11.56
##      IQR      16.20
##      CV       0.76
##      Skewness   -0.03
##      SE.Skewness 0.03
##      Kurtosis   -0.97
```

Computando estatísticas com summarise()

```
jfk.df %>% summarise(Mean = mean(temp_celsius),  
                         Std.Dev = sd(temp_celsius),  
                         Min = min(temp_celsius),,  
                         Median = median(temp_celsius),  
                         Max = max(temp_celsius),  
                         MAD = mad(temp_celsius),  
                         IQR = IQR(temp_celsius),  
                         CV = Std.Dev / Mean)
```

Computando estatísticas com summarise()

```
jfk.df %>% summarise(Mean = mean(temp_celsius),  
                         Std.Dev = sd(temp_celsius),  
                         Min = min(temp_celsius),,  
                         Median = median(temp_celsius),  
                         Max = max(temp_celsius),  
                         MAD = mad(temp_celsius),  
                         IQR = IQR(temp_celsius),  
                         CV = Std.Dev / Mean)  
  
## # A tibble: 1 × 8  
##   Mean Std.Dev  Min Median  Max    MAD    IQR     CV  
##   <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1  12.5    9.48 -11.1   12.2  36.7  11.6  16.2  0.759
```

Quantis empíricos

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
<hr/>					
##	-11.1	3	0.034	0.034	0.034
##	-10.6	5	0.057	0.092	0.057
##	-10.5	1	0.011	0.103	0.011
##	-10	12	0.138	0.241	0.138
##	-9.4	10	0.115	0.356	0.115
##	-8.9	7	0.080	0.436	0.080
##	-8.3	11	0.126	0.563	0.126
##	-7.8	18	0.207	0.770	0.207
##	-7.2	17	0.195	0.965	0.195
##	-6.7	20	0.230	1.195	0.230
##	-6.1	20	0.230	1.424	0.230
##	-6	3	0.034	1.459	0.034
##	-5.6	18	0.207	1.666	0.207
##	-5	18	0.207	1.872	0.207
##	-4.4	32	0.368	2.240	0.368
##	-3.9	56	0.643	2.883	0.643
##	-3.3	48	0.551	3.434	0.551
##	-3	9	0.103	3.538	0.103
##	-2.8	69	0.793	4.330	0.793
##	-2.2	79	0.907	5.238	0.907
##	-2	1	0.011	5.249	0.011
Souza, P. H. G. F • Aula 05 • 17 out. 2023	85	0.976	6.226	0.976	6.226 42 / 80

Quantis empíricos

```
jfk.df %>%  
  summarise(p05 = quantile(temp_celsius, prob = .05),  
            p10 = quantile(temp_celsius, prob = .10),  
            p25 = quantile(temp_celsius, prob = .25),  
            p50 = quantile(temp_celsius, prob = .50),  
            p75 = quantile(temp_celsius, prob = .75),  
            p90 = quantile(temp_celsius, prob = .90),  
            p95 = quantile(temp_celsius, prob = .95))  
  
## # A tibble: 1 × 7  
##       p05     p10     p25     p50     p75     p90     p95  
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1    -2.2      0    4.4   12.2   20.6    25     27
```

Exercício

gapminder.df: em **2007**, na **Europa**, qual a **média**, a **mediana**, o **desvio padrão**, e a **amplitude** da expectativa de vida? Quais os percentis **10** e **90**?

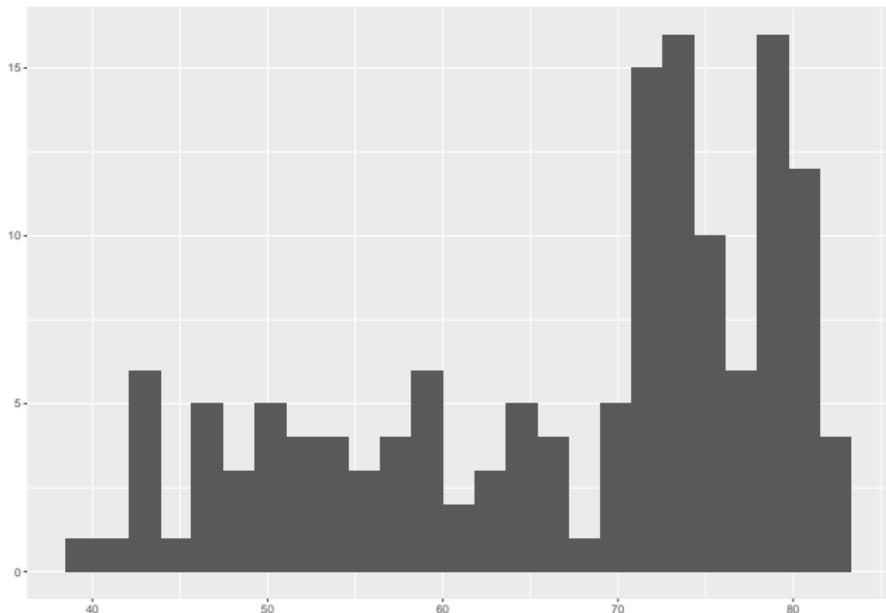
Exercício

gapminder.df: em **2007**, na **Europa**, qual a **média**, a **mediana**, o **desvio padrão**, e a **amplitude** da expectativa de vida? Quais os percentis **10** e **90**?

```
gapminder.df %>%  
  filter(year == 2007 & continent == 'Europe') %>%  
  summarise(media = mean(lifeExp),  
           mediana = median(lifeExp),  
           sd = sd(lifeExp),  
           amplitude = max(lifeExp) - min(lifeExp),  
           p10 = quantile(lifeExp, prob = .10),  
           p90 = quantile(lifeExp, prob = .90))  
  
## # A tibble: 1 x 6  
##   media mediana     sd amplitude    p10    p90  
##   <dbl>   <dbl> <dbl>      <dbl> <dbl> <dbl>  
## 1    77.6    78.6  2.98      9.98  73.3  80.9
```

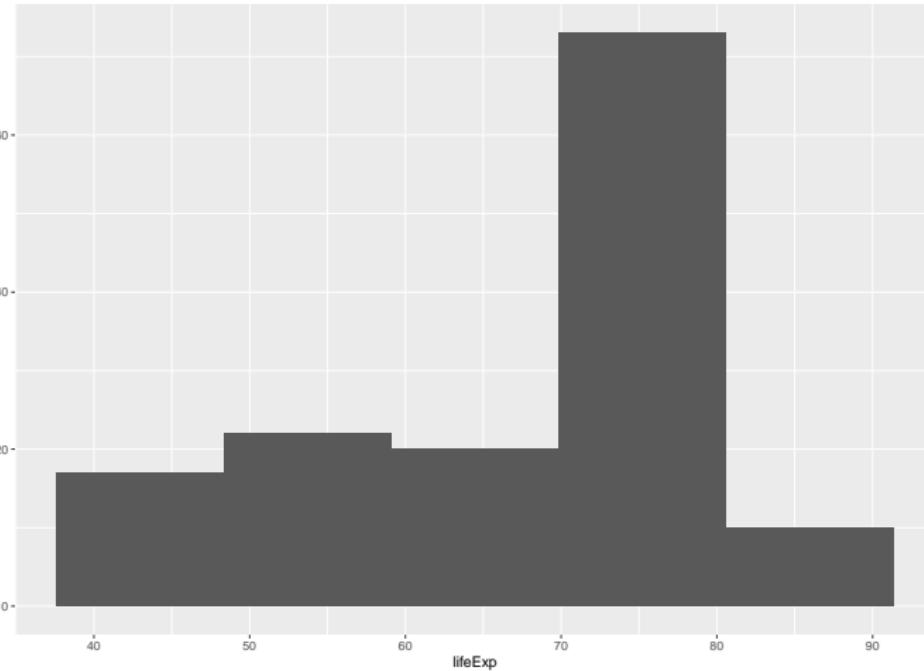
Histograma com 25 bins

```
expvida.df <- gapminder.df %>%
  filter(year == 2007) %>% select(country, lifeExp)
qplot(data=expvida.df, x=lifeExp, geom='histogram', bins=25)
```



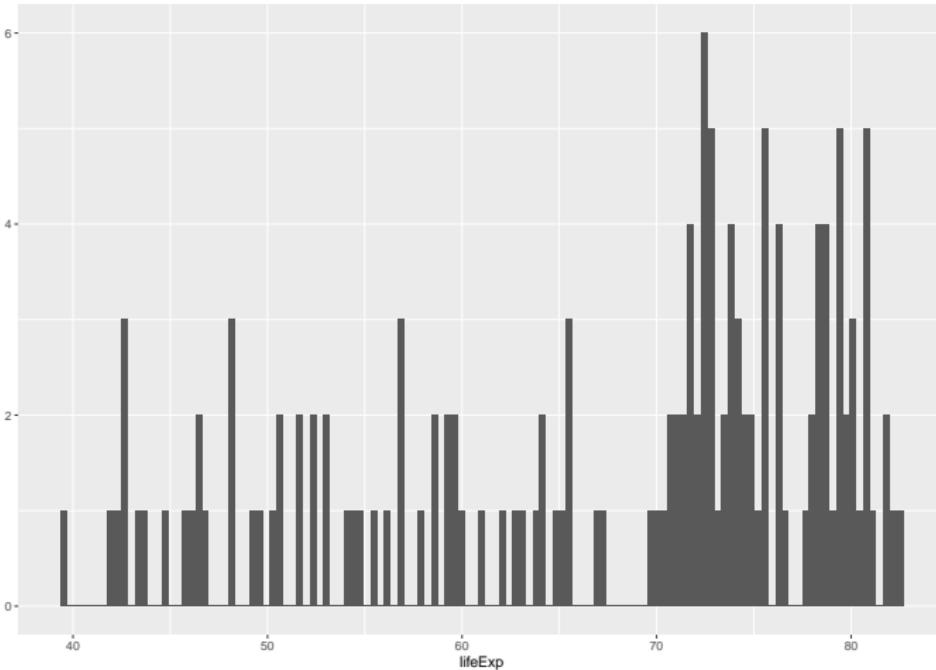
Histograma com 5 bins

```
qplot(data=expvida.df, x=lifeExp, geom='histogram', bins=5)
```



Histograma com 60 bins

```
qplot(data=expvida.df, x=lifeExp, geom='histogram', bins=125)
```



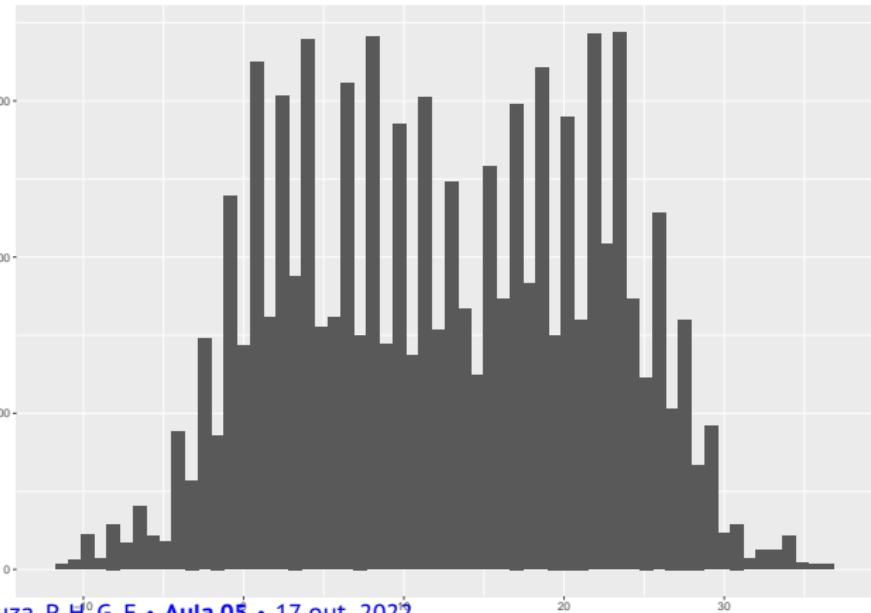
Exercício

clima.df: faça um **histograma** da **temperatura em Celsius** no aeroporto **JFK**. Teste com diferentes números de *bins*. Qual você acha melhor?

Exercício

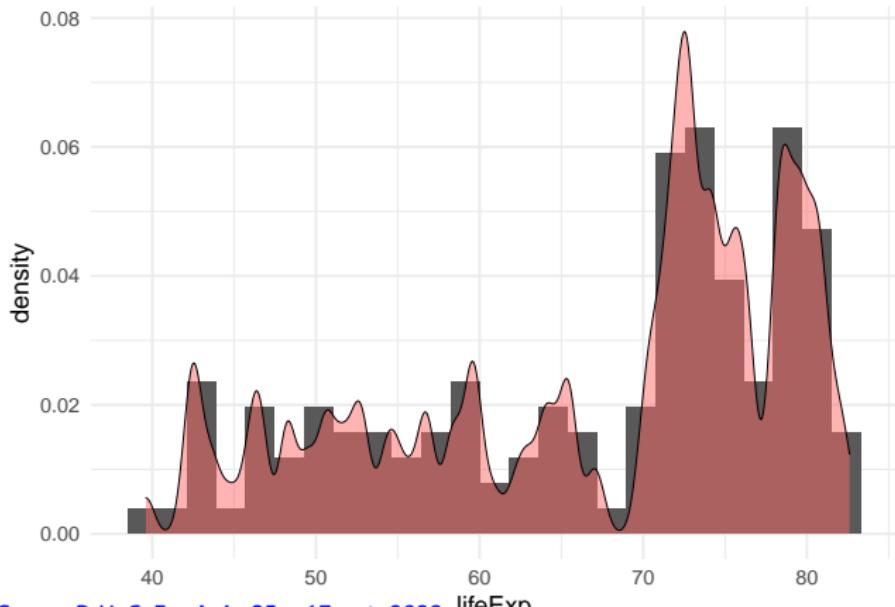
clima.df: faça um **histograma** da **temperatura em Celsius** no aeroporto **JFK**. Teste com diferentes números de *bins*. Qual você acha melhor?

```
qplot(data=jfk.df, x=temp_celsius, geom='histogram', bins=60)
```



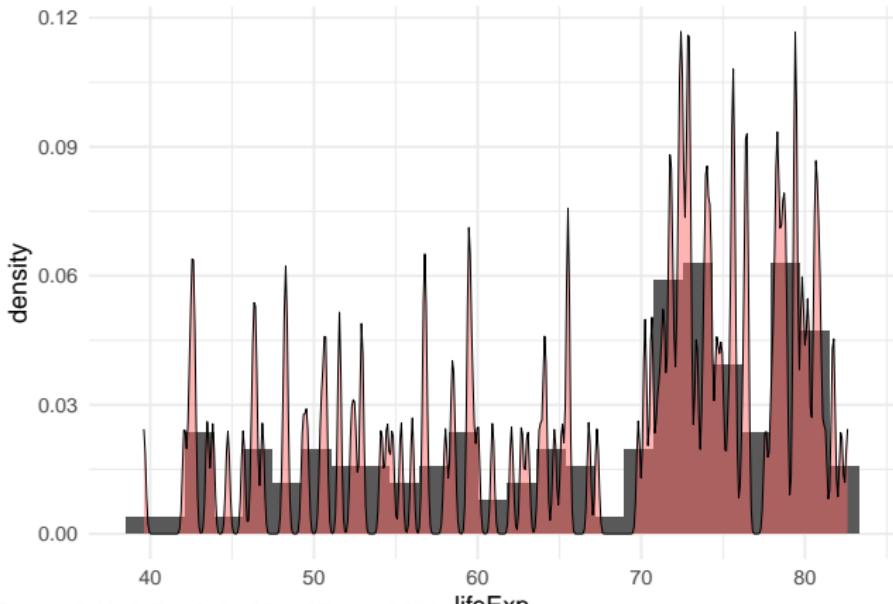
Densidade kernel (bandwidth = 0.5) (opcional)

```
ggplot(data = expvida.df, aes(x = lifeExp)) +  
  geom_histogram(aes(y = ..density..), alpha = 1, bins = 25) +  
  geom_density(fill = "indianred1", alpha = 0.5, bw = 0.5) +  
  theme_minimal(base_size = 20)
```



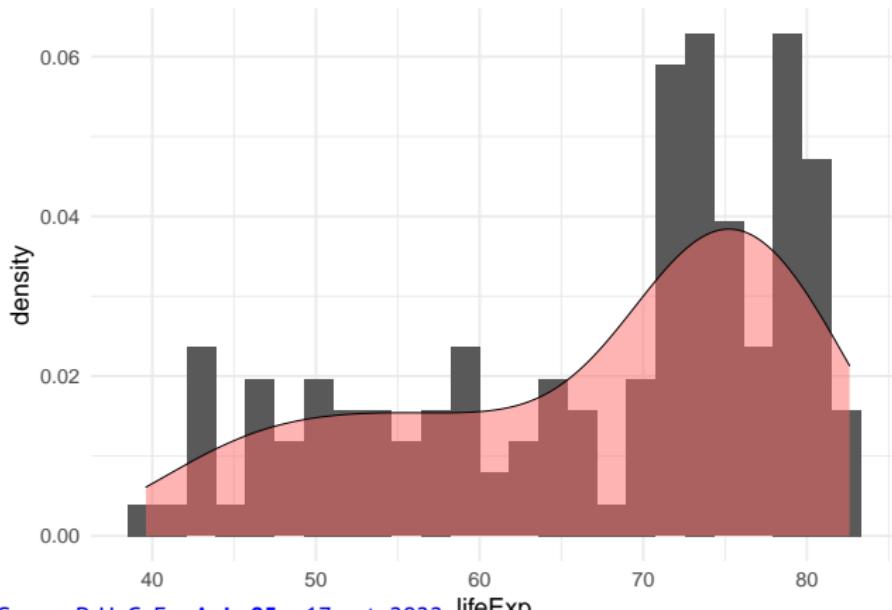
Densidade kernel (bandwidth = 0.1) (opcional)

```
ggplot(data = expvida.df, aes(x = lifeExp)) +  
  geom_histogram(aes(y = ..density..), alpha = 1, bins = 25) +  
  geom_density(fill = "indianred1", alpha = 0.5, bw = 0.1) +  
  theme_minimal(base_size = 20)
```



Densidade kernel (bandwidth = 5.0) (opcional)

```
ggplot(data = expvida.df, aes(x = lifeExp)) +  
  geom_histogram(aes(y = ..density..), alpha = 1, bins = 25) +  
  geom_density(fill = "indianred1", alpha = 0.5, bw = 5.0) +  
  theme_minimal(base_size = 20)
```



Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Tabelas cruzadas

```
ctable(oxfam.df$religiao, oxfam.df$sexo, prop = 'n',
       headings = FALSE)

## -----
##          sexo Feminino Masculino Total
## religiao
##   Católica      567     491 1058
##   Evangélica     334     259  593
##   Outras         89      75  164
##   Sem religião   97     174  271
##   Total          1087    999 2086
## -----
```

Tabelas cruzadas

```
ctable(oxfam.df$religiao, oxfam.df$sexo, prop = 'r',  
       headings = FALSE)
```

```
##
```

```
## -----
```

```
##           sexo      Feminino      Masculino
```

```
##           religiao
```

##	Católica	567 (53.6%)	491 (46.4%)	1058 (100%)
##	Evangélica	334 (56.3%)	259 (43.7%)	593 (100%)
##	Outras	89 (54.3%)	75 (45.7%)	164 (100%)
##	Sem religião	97 (35.8%)	174 (64.2%)	271 (100%)
##	Total	1087 (52.1%)	999 (47.9%)	2086 (100%)

```
## -----
```

```
-----
```

Tabelas cruzadas

```
ctable(oxfam.df$religiao, oxfam.df$sexo, prop = 'c',  
       headings = FALSE)
```

```
##
```

```
## -----  
-----
```

##	sexo	Feminino	Masculino	##
## religiao				
## Católica		567 (52.2%)	491 (49.1%)	1058 (
## Evangélica		334 (30.7%)	259 (25.9%)	593 (
## Outras		89 (8.2%)	75 (7.5%)	164 (
## Sem religião		97 (8.9%)	174 (17.4%)	271 (
## Total		1087 (100.0%)	999 (100.0%)	2086 (
## -----				

Medidas-resumo de associação (bônus)

- Teste do qui-quadrado de Pearson
- Razão de chances
- Coeficiente ϕ
- V de Cramér
- Correlação policórica
- etc

Ver, entre outros, Agresti, *Statistical Methods for the Social Sciences*, cap. 8, 2018

Gráficos de barras

Preparando os dados

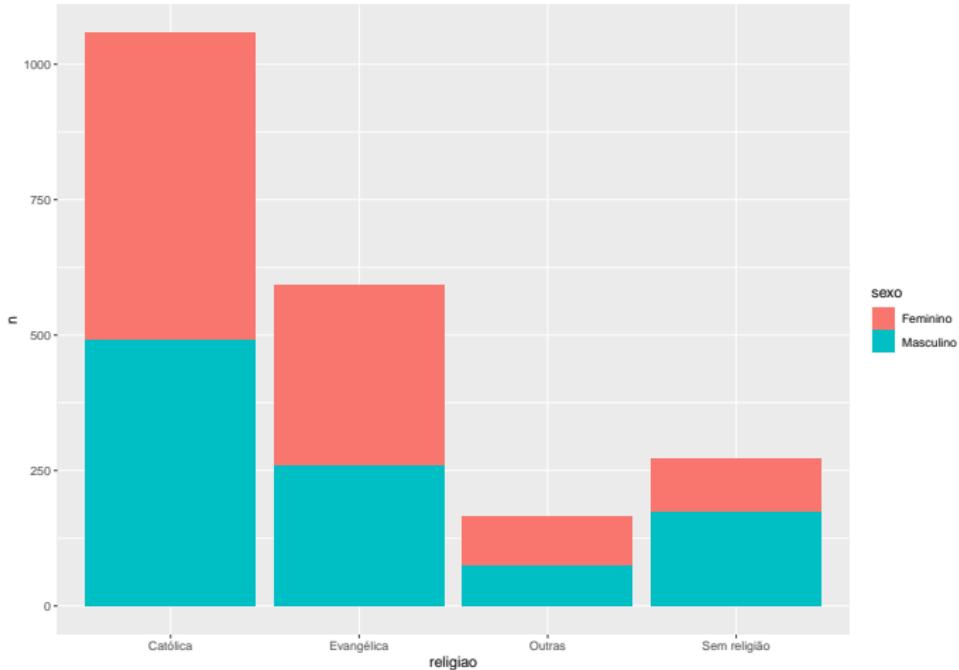
```
r.s.df <-  
  oxfam.df %>% group_by(religiao, sexo) %>% summarise(n = n())  
s.r.df <-  
  oxfam.df %>% group_by(sexo, religiao) %>% summarise(n = n())
```

O que queremos mostrar?

- Frequências absolutas?
- Frequências relativas condicionais à religião?
- Frequências relativas condicoonais ao sexo?

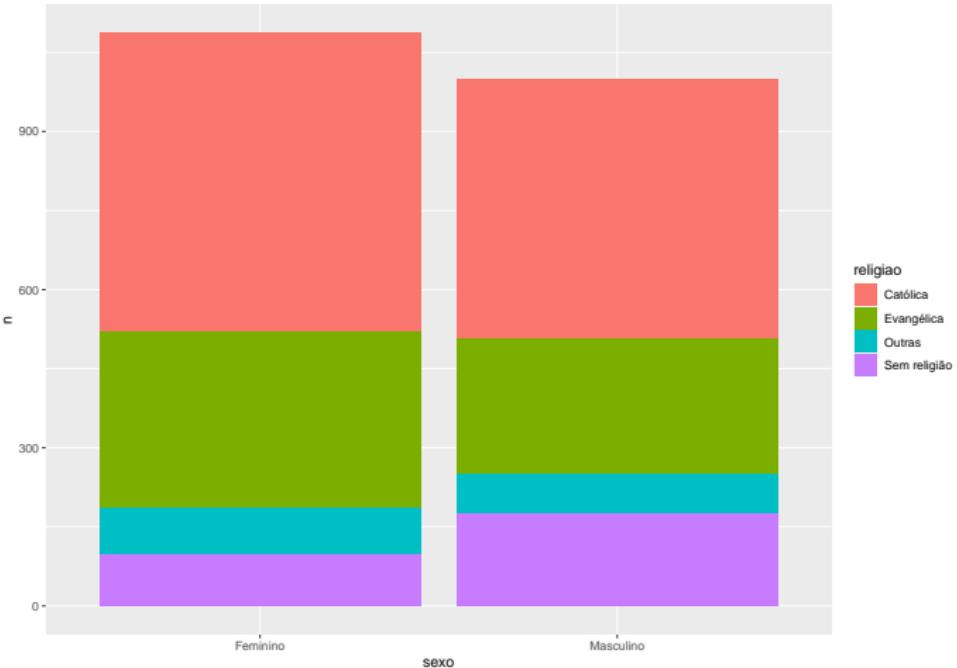
Gráficos de barras #1

```
qplot(data=r.s.df, x=religiao, y=n, fill=sexo, geom='col')
```



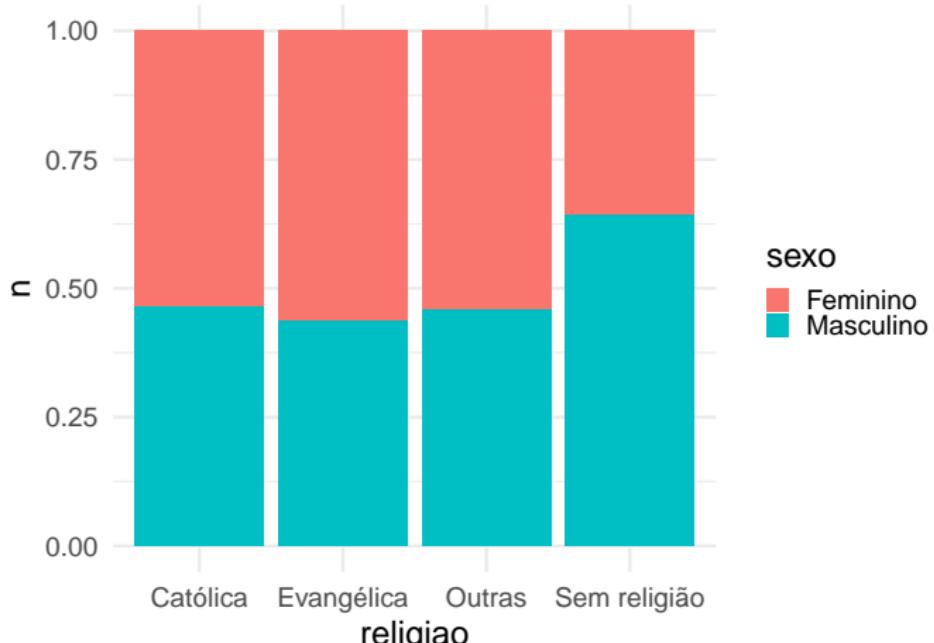
Gráficos de barras #2

```
qplot(data=s.r.df, x=sexo, y=n, fill=religiao, geom='col')
```



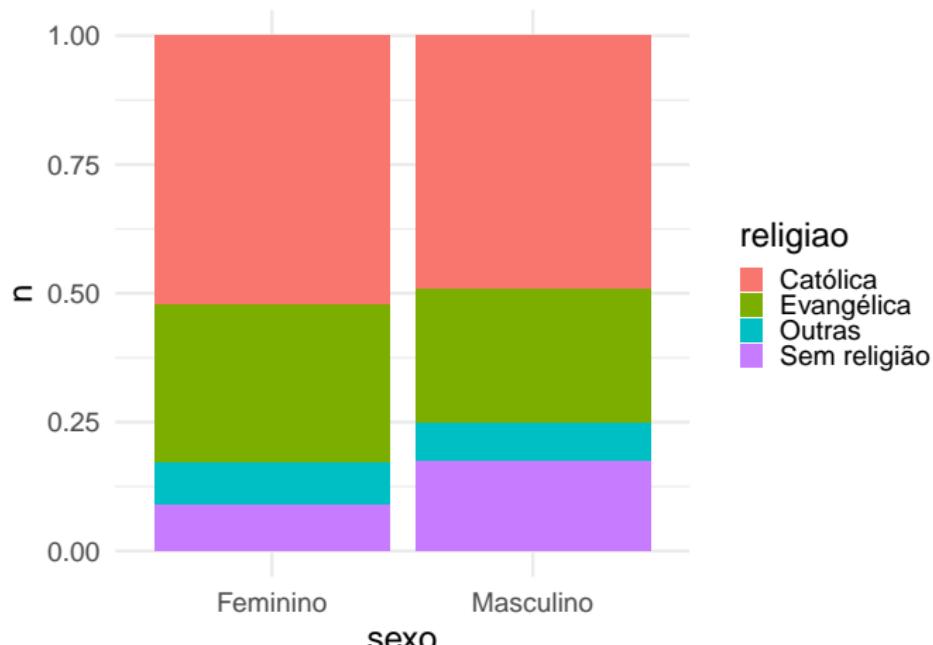
Gráficos de barras #3

```
ggplot(r.s.df, aes(x = religiao, y = n, fill = sexo)) +  
  geom_bar(position = 'fill', stat = 'identity') +  
  theme_minimal(base_size = 25)
```



Gráficos de barras #4

```
ggplot(s.r.df, aes(x = sexo, y = n, fill = religiao)) +  
  geom_bar(position = 'fill', stat = 'identity') +  
  theme_minimal(base_size = 25)
```



Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Covariância e correlação de Pearson

Covariância

$$\text{cov}(X, Y) = \frac{1}{n - 1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Covariância e correlação de Pearson

Covariância

$$\text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Correlação de Pearson

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{sd(X)sd(Y)}$$

- Medida de associação **linear** que varia entre -1 e 1
- Se $\text{corr}(X, Y) = 0$, não há associação linear

Computando essas estatísticas no R

```
jfk.df <- voos.df %>% filter(origin == 'JFK' & month <= 3)

cor(x = jfk.df$dep_delay, y = jfk.df$air_time,
    use = 'complete.obs')
## [1] -0.06898166

cor(x = jfk.df$dep_delay, y = jfk.df$arr_delay,
    use = 'complete.obs')
## [1] 0.9001721
```

Computando essas estatísticas no R

```
cor.gap.df <-  
  gapminder.df %>%  
    select(lifeExp, pop, gdpPercap) %>%  
    mutate(log_gdpPercap = log(gdpPercap))  
  
cor(cor.gap.df)  
  
##           lifeExp      pop  gdpPercap log_gdpPercap  
## lifeExp       1.000   0.065     0.584      0.808  
## pop          0.065   1.000    -0.026     -0.055  
## gdpPercap    0.584  -0.026     1.000      0.798  
## log_gdpPercap 0.808  -0.055     0.798      1.000
```

Exercício

oxfam. df: qual a correlação entre as avaliações sobre o que importa para reduzir a desigualdade no Brasil? (variáveis p27a-p27e)

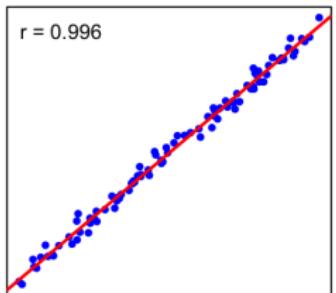
Exercício

oxfam.df: qual a correlação entre as avaliações sobre o que importa para reduzir a desigualdade no Brasil? (variáveis p27a-p27e)

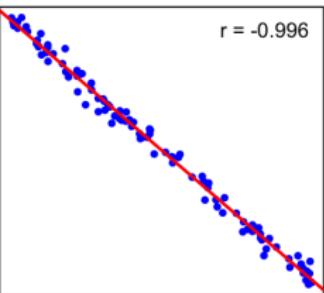
```
avaliacoes.df <- oxfam.df %>% select(contains('p27'))  
cor(avaliacoes.df, use = 'complete.obs')  
  
##          p27a  p27b  p27c  p27d  p27e  
## p27a 1.00 0.13 0.12 0.269 0.107  
## p27b 0.13 1.00 0.42 0.135 0.297  
## p27c 0.12 0.42 1.00 0.104 0.402  
## p27d 0.27 0.13 0.10 1.000 0.032  
## p27e 0.11 0.30 0.40 0.032 1.000
```

Scatterplots

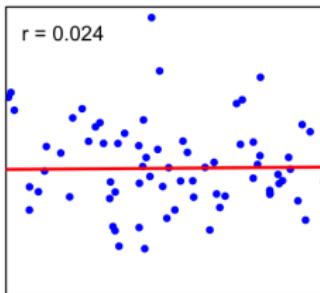
strong positive linear correlation



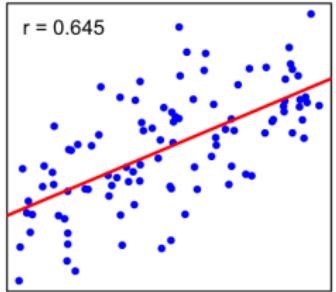
strong negative linear correlation



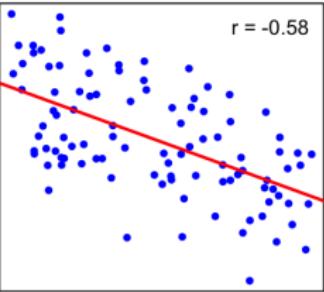
no linear correlation



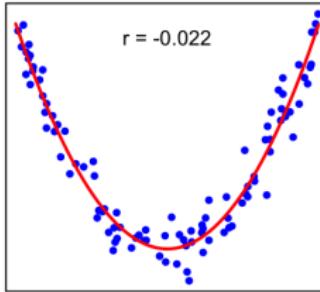
weak to medium positive linear correlation



weak to medium negative linear correlation

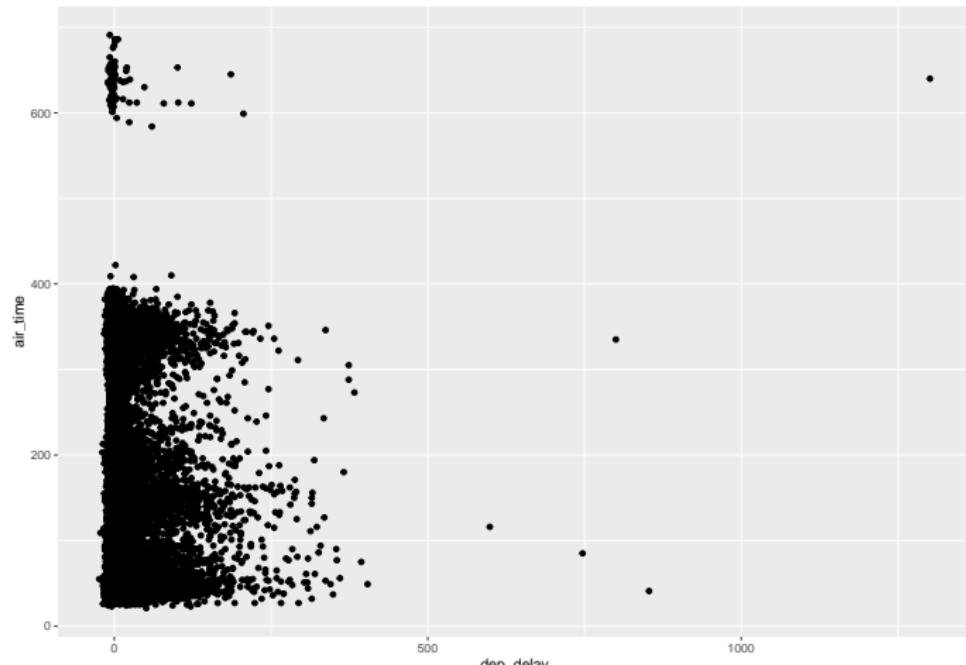


no linear correlation



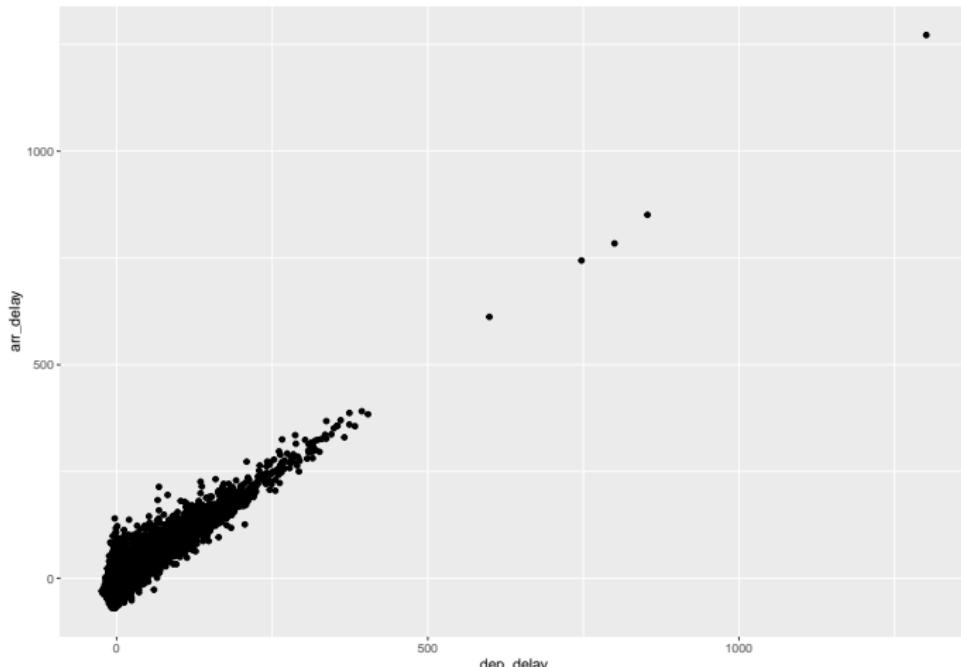
Correlação = -0.07

```
qplot(data = jfk.df, x = dep_delay, y = air_time,  
      geom = 'point')
```



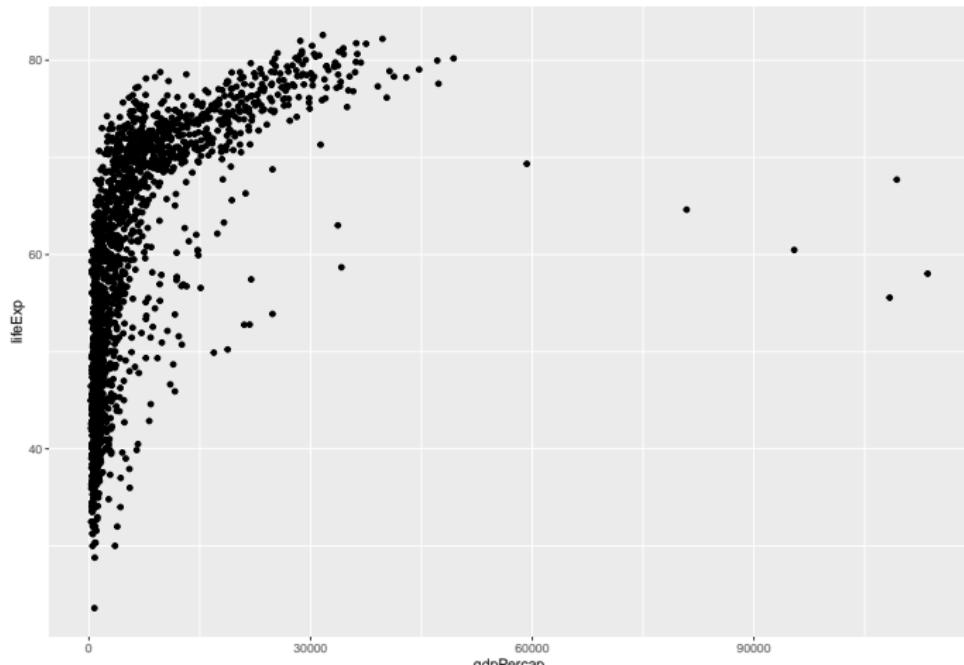
Correlação = 0.90

```
qplot(data = jfk.df, x = dep_delay, y = arr_delay,  
       geom = 'point')
```



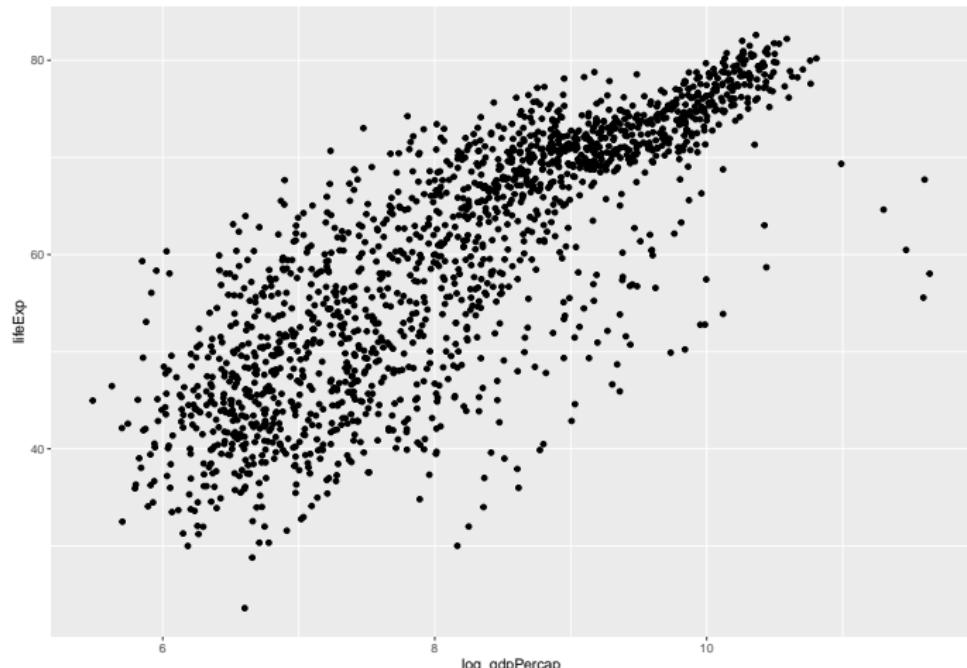
Antes: correlação = 0.58

```
qplot(data = cor.gap.df, x = gdpPercap, y = lifeExp,  
       geom = 'point')
```



Depois: correlação = 0.81

```
qplot(data = cor.gap.df, x = log_gdpPercap, y = lifeExp,  
       geom = 'point')
```



Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Estatísticas condicionais

```
oxfam.df %>%  
  group_by(religiao) %>%  
  summarise(n = n(),  
            p27a_mean = mean(p27a, na.rm = TRUE ),  
            p27d_mean = mean(p27d, na.rm = TRUE ),  
            p27e_mean = mean(p27e, na.rm = TRUE))  
  
## # A tibble: 4 x 5  
##   religiao      n  p27a_mean  p27d_mean  p27e_mean  
##   <chr>     <int>     <dbl>       <dbl>       <dbl>  
## 1 Católica    1058      8.27        7.80       9.64  
## 2 Evangélica   593       8.05        7.93       9.71  
## 3 Outras       164       7.80        7.38       9.85  
## 4 Sem religião 271       7.97        7.51       9.70
```

Estatísticas condicionais

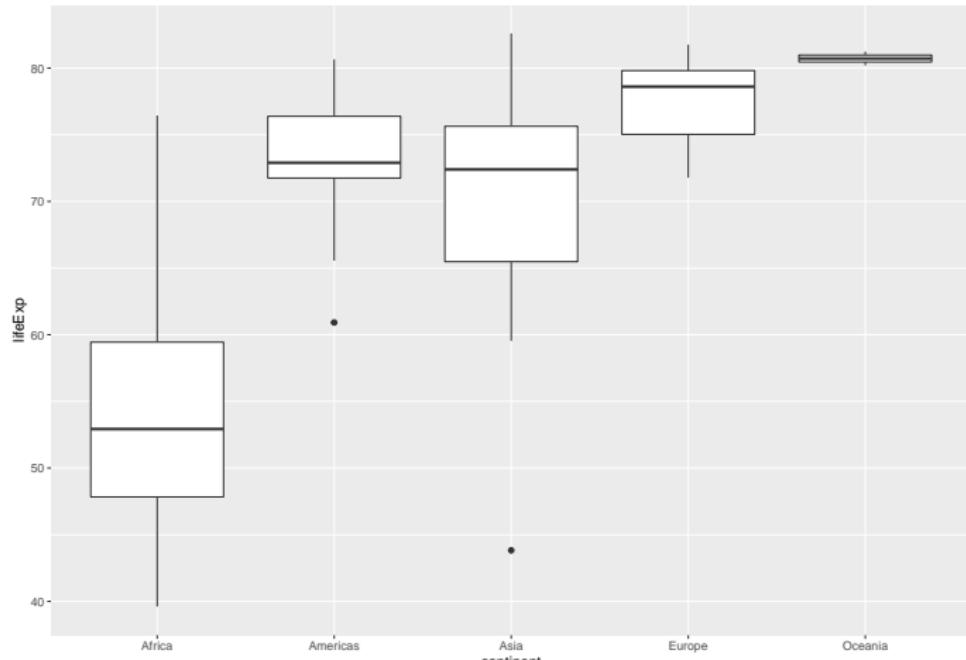
```
voos.df %>% group_by(month) %>% select(dep_delay) %>%
  descr(stats = c('mean', 'med', 'n.valid'), headings = FALSE, transpose)

##
```

		Mean	Median	N.Valid
##	month = 1	10.04	-2.00	26483.00
##	month = 2	10.82	-2.00	23690.00
##	month = 3	13.23	-1.00	27973.00
##	month = 4	13.94	-2.00	27662.00
##	month = 5	12.99	-1.00	28233.00
##	month = 6	20.85	0.00	27234.00
##	month = 7	21.73	0.00	28485.00
##	month = 8	12.61	-1.00	28841.00
##	month = 9	6.72	-3.00	27122.00
##	month = 10	6.24	-3.00	28653.00
##	month = 11	5.44	-3.00	27035.00
##	month = 12	16.58	0.00	27110.00

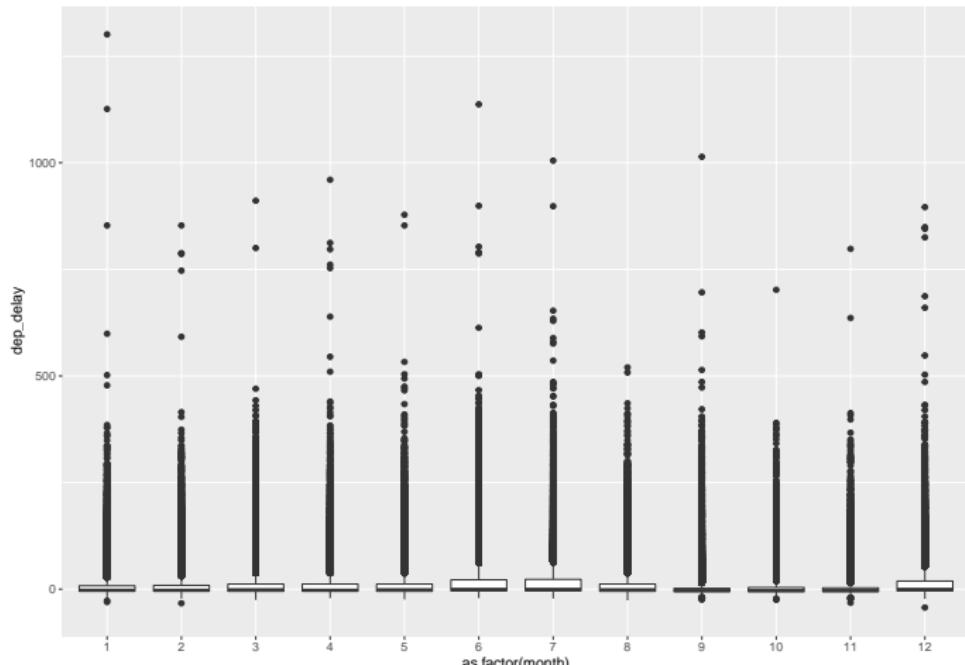
Box plots

```
boxplot.df <- gapminder.df %>% filter(year == 2007)  
qplot(data=boxplot.df, x=continent, y=lifeExp, geom='boxplot')
```



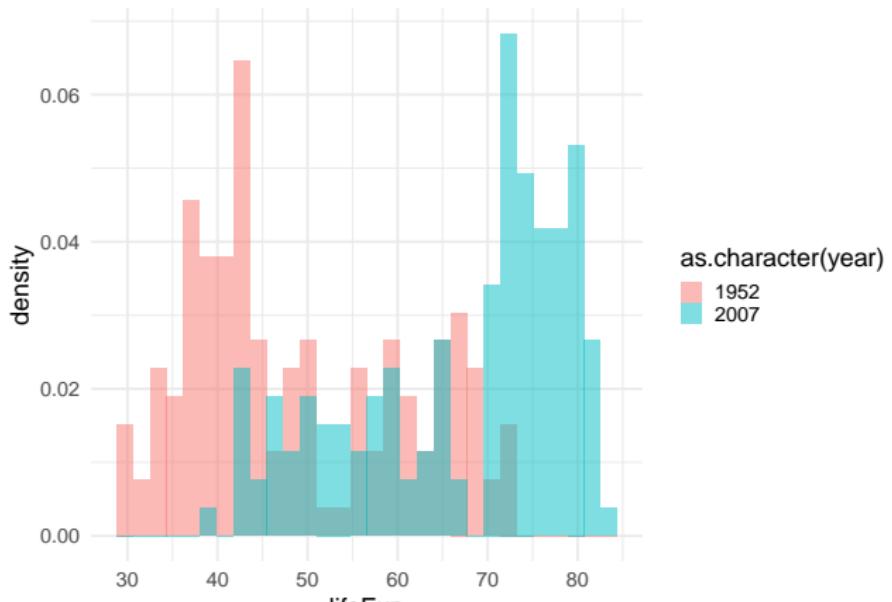
Box plots

```
qplot(data = voos.df, x = as.factor(month), y = dep_delay,  
       geom='boxplot')
```



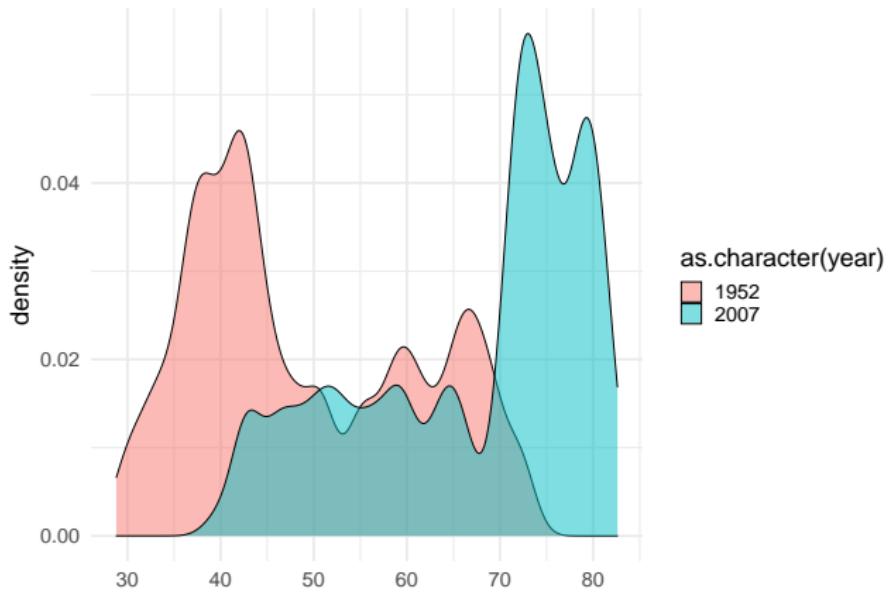
Histogramas sobrepostos (20 bins)

```
ggplot(data = filter(gapminder.df, year %in% c(1952, 2007)),  
       aes(x = lifeExp, fill = as.character(year))) +  
  geom_histogram(aes(y = ..density..), position = 'identity',  
                 alpha = 0.5, bins = 30) + theme_minimal(base_size = 20)
```



Densidades kernel sobrepostas (bw = 1.5)

```
ggplot(data = filter(gapminder.df, year %in% c(1952, 2007)),  
       aes(x = lifeExp, fill = as.character(year))) +  
  geom_density(aes(y = ..density..), position = 'identity',  
               alpha = 0.5, bw = 1.5) + theme_minimal(base_size = 20)
```



Recapitulação

Introdução

Mais funções no R

Estatísticas univariadas para variáveis discretas

Estatísticas univariadas para variáveis contínuas

Estatísticas bivariadas para variáveis discretas

Estatísticas bivariadas para variáveis contínuas

Estatística bivariadas mistas

Próxima aula

Próxima aula

Atividades

Entrega da atividade #4 no Google Classroom

Leituras obrigatórias

Agresti 2018, cap. 2 e 4

Leituras optativas

Bussab e Morettin 2010 cap. 5 a 8

Kellstedt e Whitten 2018, cap. 7