

```

install.packages("tidyverse")
library("tidyverse")
tidyverse_update

install.packages(c("nycflights13", "gapminder", "Lahman"))
install.packages("stringr")
library("tidyverse")
install.packages("ggplot2")
library("ggplot2")
library("dplyr")
library(nycflights13)

# histograma para verificação da incidência de cada variável categórica na
# amostra
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut)))

ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)

# histograma com menor binwidth
smaller <- diamonds %>%
  filter(carat < 3)
ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.1)

# histograma com linhas ao invés de colunas para uma melhor visualização da
# sobreposição da frequência das variáveis
ggplot(data = smaller, mapping = aes(x = carat, colour = cut)) +
  geom_freqpoly(binwidth = 0.1)

# histograma com menor binwidth, porém há algum erro no comando que faz com que
# a largura da coluna permaneça em 0.1
ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.01)

ggplot(data = faithful, mapping = aes(x = eruptions)) +
  geom_histogram(binwidth = 0.25)

# identificando outliers através de histogramas
ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y, binwidth = 0.5))

# aumentando a visualização do histograma pra visualizar observações com poucas
# ocorrências
ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  coord_cartesian(ylim = c(0, 50))

unusual <- diamonds %>%
  filter(y < 3 | y > 20) %>%
  select(price, x, y, z) %>%
  arrange(y)
unusual

# missing values

# substituindo outliers por NA
diamonds2 <- diamonds %>%
  mutate(y = ifelse(y < 3 | y > 20, NA, y))

# plotando os dados sem os outliers
ggplot(data = diamonds2, mapping = aes(x = x, y = y)) +
  geom_point()

# apagando a mensagem de alerta no console
ggplot(data = diamonds2, mapping = aes(x = x, y = y)) +
  geom_point(na.rm = TRUE)

# observações de partida dos voos
# voos cancelados (NA) separados da contagem
nycflights13::flights %>%
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %/% 100,
    sched_min = sched_dep_time %% 100,
    sched_dep_time = sched_hour + sched_min / 60
  ) %>%
  ggplot(mapping = aes(sched_dep_time)) +
  geom_freqpoly(mapping = aes(colour = cancelled), binwidth = 1/4)

# 7.5 covariation

ggplot(data = diamonds, mapping = aes(x = price)) +
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)

ggplot(diamonds) +
  geom_bar(mapping = aes(x = cut))

# eixo y = densidade (contagem padronizada)
ggplot(data = diamonds, mapping = aes(x = price, y = ..density..)) +
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)

# boxplot com a distribuição dos preços dos diamantes
ggplot(data = diamonds, mapping = aes(x = cut, y = price)) +
  geom_boxplot()

# highway mileage variation
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot()

# boxplot reordenado com base no valor da mediana das classes denominadas no
# eixo y (highway)
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy))

```

```

# girando o boxplot em 90°
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy)) +
  coord_flip()

# visualizando covariação entre variáveis categóricas
ggplot(data = diamonds) +
  geom_count(mapping = aes(x = cut, y = color))

# observando o número de observações de outras formas
library("dplyr")
diamonds %>%
  count(color, cut)

diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = color, y = cut)) +
  geom_tile(mapping = aes(fill = n)) +

install.packages("seriation")
library("seriation")

# covariação entre variáveis contínuas
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price))

# visualizando as regiões em que ocorrem "overplot" com maior frequência de
# observações
ggplot(data = smaller) +
  geom_bin2d(mapping = aes(x = carat, y = price))

install.packages("hexbin")
library("hexbin")
ggplot(data = smaller) +
  geom_hex(mapping = aes(x = carat, y = price))

# boxplot com o binwidth variando de acordo com o número de observações em cada
# caixa
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_width(carat, 0.5)), varwidth = T)

ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_number(carat, 20)))

# 7.6 patterns and models
ggplot(data = faithful) +
  geom_point(mapping = aes(x = eruptions, y = waiting))

library("modelr")

mod <- lm(log(price) ~ log(carat), data = diamonds)

diamonds2 <- diamonds %>%
  add_residuals(mod) %>%
  mutate(resid = exp(resid))

ggplot(data = diamonds2) +
  geom_point(mapping = aes(x = carat, y = resid))

ggplot(data = diamonds2) +
  geom_boxplot(mapping = aes(x = cut, y = resid))

# ggplot2 calls
ggplot(data = faithful, mapping = aes(x = eruptions)) +
  geom_freqpoly(binwidth = 0.25)

ggplot(faithful, aes(eruptions)) +
  geom_freqpoly(binwidth = 0.25)

diamonds %>%
  count(cut, clarity) %>%
  ggplot(aes(clarity, cut, fill = n)) +
  geom_tile()

# questão 13

# extraindo os valores das colunas 1 a 15, correspondentes aos anos de 1876 a
# 1932
bd
bd[1 : 15,]
bd13 <- bd[1 : 15,]

# regressão utilizando os valores extraídos
reg13 <- lm(Vote ~ Growth, data = bd13)
summary(reg13)

# representação gráfica dos parâmetros da regressão
library(ggplot2)
ggplot(data = reg13, aes(x = Growth, y = Vote))
ggplot(data = reg13, aes(x = Growth, y = Vote)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x)

ggplot(data = reg13, aes(x = Growth, y = Vote)) +
  geom_point() +
  geom_abline(data = reg13, slope = 0.5336, intercept = 51.9850)

View(reg13)

# regressão com dados populacionais
reg <- lm(Vote ~ Growth, data = bd)
summary(reg)

# gráfico dos parâmetros da regressão
ggplot(data = reg, aes(x = Growth, y = Vote)) +

```

```
  geom_point() +  
  geom_abline(data = reg, slope = 0.62, intercept = 51.5082)  
ggplot(data = reg, aes(x = Growth, y = Vote)) +  
  geom_point() +  
  geom_smooth(method = "lm", formula = y ~ x)
```