

Análise dos Dados com R

Visualização dos Dados e EDA

James R. Hunter, PhD
Retrovirologia, EPM, UNIFESP

2023-10-10



Visualização dos Dados



John Tukey on Visualization

The simple graph has brought more information to the data analyst's mind than any other device.

O gráfico simples trouxe mais informações à mente do analista dos dados do que qualquer outro dispositivo.

Tópico de suprema importância

Mas, Primeiro ...

Mais Duas Funções Importantes de “Data Munging”

Dados Extensos (*Wide*) vs. Profundos (*Long*)

Este slide aborda a diferença entre os formatos de dados extensos e profundos, com foco na sua aplicação em aprendizado de máquina.

O formato *Wide* é mais comumente usado para classificação e regressão, enquanto o formato *Long* é mais adequado para tarefas de agrupamento e previsão de séries temporais.

Os exemplos mostrados incluem:

- Formato *Wide*: Exemplos de classificação e regressão.
- Formato *Long*: Exemplos de agrupamento e previsão de séries temporais.

As diferenças entre os formatos de dados são fundamentais para a eficiência e a eficácia das técnicas de aprendizado de máquina.

As aplicações práticas dos formatos de dados extensos e profundos são amplamente variadas, desde sistemas de recomendação até processos de análise de dados.

É importante entender as nuances entre os dois formatos para escolher a abordagem mais adequada para uma tarefa de aprendizado de máquina.

As técnicas de transformação de dados são cruciais para adaptar os formatos de dados para as necessidades do modelo de aprendizado de máquina.

As aplicações práticas dos formatos de dados extensos e profundos são amplamente variadas, desde sistemas de recomendação até processos de análise de dados.

É importante entender as nuances entre os dois formatos para escolher a abordagem mais adequada para uma tarefa de aprendizado de máquina.

Este Quer Dizer

- Planilhas normalmente apresentam dados no formato **extenso**
 - Cada caso tem um número de variáveis
- Para algumas análises, precisamos combinar algumas das variáveis
 - Esta operação faz o formato **profundo**

Dados de Exemplo

- Vem das bases de dados do Estado de São Paulo (SEADE)
 - Uma tabela de comorbidades
- Conjunto randomizado de 300 casos dos dados demográficos e de comorbidades
- Conjunto já *tidy*

Dados

```
1 sp_comorb <- readRDS(here::here("seade_comorb_sample.rds")) %>%
2   mutate(pacid = 1:nrow(.), .before = 1) # add pacid to make what is happen
3   glimpse(sp_comorb)
```

```
Rows: 300
Columns: 10
$ pacid      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, ...
$ city       <chr> "Itaquaquecetuba", "Sorocaba", "Sao Paulo", "Sao Paulo",
"...
$ age        <dbl> 58, 62, 78, 65, 59, 68, 67, 83, 61, 58, 73, 67, 77, 77,
39...
$ sex        <fct> male, male, female, female, male, female, female, female,
...
$ death      <lgl> FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE,
TRUE...
$ cardiopathy <fct> true, true, true, true, false, true, true, true,
tru...
$ diabetes    <fct> true, NA, true, true, true, NA, NA, NA, false, NA, true,
N
```

Mudar Formato para Análise Desejada

- Para a análise atual, queremos estudar comorbidades como um grupo
 - Não como as condições dos indivíduos
- Neste caso ...
 - Cada comorbidade não é uma variável em si
 - São **valores** de 2 novas variáveis
 - **comorbid**: o nome da comorbidade (a chave - *key*)
 - **value**: presença ou ausência da condição (o valor - *value*)

Função `tidyverse::pivot_longer()`

- `cols` = colunas que seriam combinados em pares *key:value*
- `names_to` = o nome da variável que vai conter as chaves
- `values_to` = o nome da variável que vai conter os valores

Novo Tibble Long

```
1 sp_comorb_long <- sp_comorb %>%
2   pivot_longer(cols = cardiopathy:kidney, names_to = "comorbid",
3                 values_to = "value")
4 glimpse(sp_comorb_long)
```

Rows: 1,500
Columns: 7

```
$ pacid    <int> 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4,
5...
$ city     <chr> "Itaquaquecetuba", "Itaquaquecetuba", "Itaquaquecetuba",
"Ita...
$ age      <dbl> 58, 58, 58, 58, 58, 62, 62, 62, 62, 78, 78, 78, 78,
6...
$ sex      <fct> male, male, male, male, male, male, male, male, male,
f...
$ death    <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE,
TRUE...
$ comorbid <chr> "cardiopathy", "diabetes", "obesity", "neuro", "kidney",
"car...
$ value    <fct> true, true, false, false, true, NA, NA, NA, NA, true,
```

sp_comorb_long sobre Comorbidades - Sim

- Pacientes não um unidade básica deste formato
 - Cada **pacid** aparece 5 vezes
 - 1 para cada comorbidade

```
# A tibble: 3 × 7
  pacid city          age sex   death comorbid  value
  <int> <chr>      <dbl> <fct> <lgl> <chr>    <fct>
1     1 Itaquaquecetuba    58 male FALSE cardiopathy true
2     1 Itaquaquecetuba    58 male FALSE diabetes    true
3     1 Itaquaquecetuba    58 male FALSE obesity    false
```

Pode Inverter o Processo *Long* a *Wide*

- `tidy::pivot_wider()`
- Valores de variável *key* tornam nomes das variáveis no formato *wide*
- Valores de variável *value* tornam valores desses novas variáveis

Exemplo da Inversão

```
1 sp_comorb_wide <- sp_comorb_long %>%
2   pivot_wider(names_from = "comorbid",
3               values_from = "value")
4 glimpse(sp_comorb_wide)
```

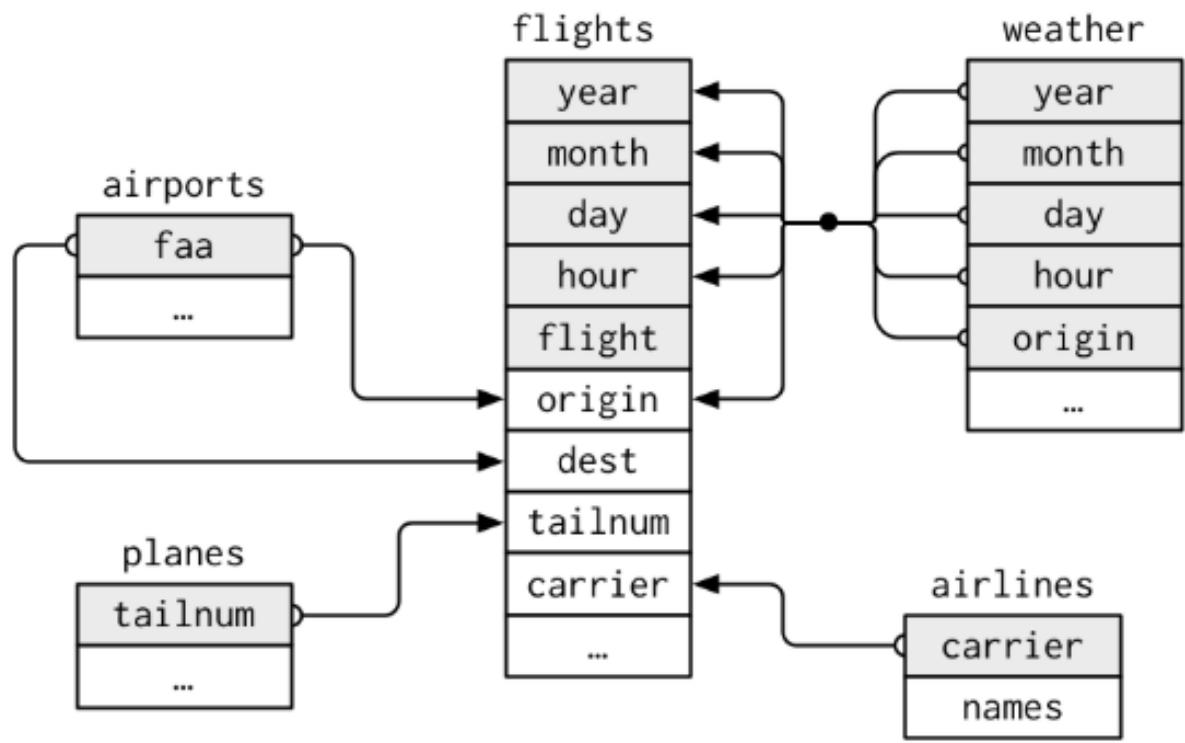
```
Rows: 300
Columns: 10
$ pacid      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17...
$ city       <chr> "Itaquaquecetuba", "Sorocaba", "Sao Paulo", "Sao Paulo",
"...
$ age        <dbl> 58, 62, 78, 65, 59, 68, 67, 83, 61, 58, 73, 67, 77, 77,
39...
$ sex        <fct> male, male, female, female, male, female, female, female,
...
$ death      <lgl> FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE,
TRUE...
$ cardiopathy <fct> true, true, true, true, false, true, true, true,
tru...
$ diabetes    <fct> true, NA, true, true, true, NA, NA, NA, false, NA, true,
N
```

Juntar Dados de *Tibbles* Diferentes

- Dados para uma análise podem estar gravados em mais que uma tabela
- Especialmente quando está trabalhando com `ly true for data from` *bancos de dados relacionais* como SQL
- `join...` funções para integrar *data frames* baseados em chaves comuns

Dados para *Joins*

- Dados sobre voos saindo de qualquer um dos 3 aeroportos de NY em 2013
 - Pacote [nycflights13](#)
 - Tabelas para
 - Nomes das linhas aéreas que servem os aeroportos
 - Aeroportos de destino de voo
 - Aviões - modelos e número de cauda
 - Tempo nos aeroportos
 - Voos - componente central do sistema



Escolhe Amostra de 10 Voos

```
1 library(nycflights13)
2 data(flights)
3 # select a set of 10 flights
4 flights <- flights %>%
5   slice_sample(n = 10) %>%
6   select(year:day, flight, origin, dest, carrier) # select subset of vars
7 flights

# A tibble: 10 × 7
  year month   day flight origin dest carrier
  <int> <int> <int>   <int> <chr> <chr> <chr>
1  2013     2     6     3223  LGA    MDW    WN
2  2013    11    16     2219  LGA    MSP    DL
3  2013     5     5      77  JFK    FLL    B6
4  2013     8    12     4115  JFK    IAD    9E
5  2013     9     3     1511  EWR    RSW    B6
6  2013     7    12      575  JFK    MSY    B6
7  2013     5     5     1561  EWR    SFO    UA
8  2013    12     1     2302  JFK    BUF    B6
9  2013    11     1     3326  JFK    IND    9E
10 2013     7     7     1695  EWR    IAH    UA
```

No flights, Só ID de 2 Letras, Sem Nome Completo ([airlines](#))

```
1 # load the airlines list
2 data(airlines)
3 head(airlines)

# A tibble: 6 × 2
  carrier name
  <chr>   <chr>
1 9E      Endeavor Air Inc.
2 AA      American Airlines Inc.
3 AS      Alaska Airlines Inc.
4 B6      JetBlue Airways
5 DL      Delta Air Lines Inc.
6 EV      ExpressJet Airlines Inc.
```

Juntar Nome de Linha aos Voos

- As 2 tabelas têm variável `carrier`
 - `carrier` - código de 2 dígitos
- `left_join()`
 - Juntar os dados da tabela da RHS para os dados no LHS
 - Usando variáveis em comum
 - Somente mostra colunas relacionados ao problema atual

```
1 # join the airline names to the flights
2 flights_mod <- flights %>%
3   left_join(airlines, by = "carrier")
4 flights_mod[, 4:8]

# A tibble: 10 × 5
  flight origin dest carrier name
  <int> <chr>  <chr> <chr>  <chr>
1 3223  LGA    MDW    WN     Southwest Airlines Co.
2 2219  LGA    MSP    DL     Delta Air Lines Inc.
3 77    JFK    FLL    B6     JetBlue Airways
4 4115  JFK    IAD    9E     Endeavor Air Inc.
5 1511  EWR    RSW    B6     JetBlue Airways
6 575   JFK    MSY    B6     JetBlue Airways
7 1561  EWR    SFO    UA     United Air Lines Inc.
8 2302  JFK    BUF    B6     JetBlue Airways
9 3326  JFK    IND    9E     Endeavor Air Inc.
10 1695  EWR   IAH    UA     United Air Lines Inc.
```

Tipos de Joins

- Joins de mutação como `left_join`
 - Mudar a *data frame* do lado esquerdo
 - Pode até tirar fileiras do *data frame* do lado esquerdo
 - Usar dados do *data frame* do lado direto
 - Mas não mudar o *data frame* do lado direto
- Outros joins de mutação
 - `right_join()`
 - Inversão dos papéis dos *data frames* do esquerdo e direto
 - `full_join()`
 - Mantem todos as fileiras no lado esquerdo se existe a chave correspondente certa ou não
 - `inner_join()` Somente mantém as fileiras com valor de chave nos dois lados

VSS: Chaves em *Joins*

- Se as chaves dos lados esquerdo e direto têm nomes, precisa usar um `by` = diferente
- Caso que tem nome da chave = `a` no esquerdo e `b` no direto
- `by = c("a" = "b")`
 - Uso da função `c()`
 - Uso das aspas

Análise Exploratório dos Dados

Exploração Inicial dos Dados

- Onde queremos tentamos achar o que os dados estão dizendo
- Principal uso de visualizações
- Série de medidas e gráficos que mostram as variáveis
- Exploração das variáveis
 - Uma por vez (univariada)
 - Tabulações cruzadas de conjuntos de variáveis
- Sempre procurando valores de dados estranhos

Dados: fute_mod.rds

- Conjunto dos dados sobre lesões relacionadas ao futebol nos EUA

```
1 library(tidyverse)
2 fm <- readRDS(here::here("fute_mod_2020.rds")) %>%
3   mutate(age_grp = factor(case_when(
4     age < 18 ~ "youth",
5     age < 60 ~ "adult",
6     TRUE ~ "elderly"
7   ))) %>%
8   mutate(age_grp = fct_relevel(age_grp, c("youth", "adult", "elderly")))
9 glimpse(fm)
```

Rows: 7,603
Columns: 10

```
$ case_num      <chr> "160102033", "160106032", "160107304", "160109914", "16011...
$ trmt_date    <date> 2016-01-02, 2016-01-02, 2016-01-01, 2016-01-01, 2016-01-0...
$ age          <dbl> 27, 14, 9, 16, 17, 33, 12, 16, 12, 50, 10, 15, 17, 11, 16, ...
$ sex          <fct> Male, Male, Male, Female, Female, Male, Male, Female, Male...
$ body_part    <fct> Foot, Knee, Toe, Wrist, Wrist, Knee, Finger, Head, Finger, ...
$ diag         <fct> "Contusion Or Abrasion", "Fracture", "Fracture", "Strain, ...
$ disposition  <fct> Released, Released, Released, Released, Released, Released...
$ psu          <fct> 63, 61, 8, 20, 73, 61, 58, 61, 63, 61, 20, 20, 20, 17, 61, ...
$ narrative    <chr> "27YOM PLAYING SOCCER COLLIDED WITH ANOTHER PLAYER CONTUSI...
$ age_grp      <fct> adult, youth, youth, youth, adult, youth, yo...
```

Variável age

```
1 summarytools:::descr(fm$age)
```

Descriptive Statistics

fm\$age

N: 7603

	age
Mean	16.38
Std.Dev	8.92
Min	0.00
Q1	11.00
Median	14.00
Q3	17.00
Max	85.00
MAD	4.45
IQR	6.00
CV	0.54

Min = 0.00 ?

```
summarytools::descr(fm$age)

## Descriptive Statistics
## fm$age
## N: 7603
##
##                               age
## -----
##          Mean      16.38
##          Std.Dev     8.92
##          Min      0.00
##          Q1      11.00
##          Median    14.00
##          Q3      17.00
##          Max      85.00
##          MAD      4.45
##          IQR      6.00
##          CV       0.54
##          Skewness   2.22
##          SE.Skewness 0.03
##          Kurtosis   6.60
##          N.Valid   7603.00
##          Pct.Valid 100.00
```

Quem É Essa Pessoa com age = 0?

- UNK AGE MALE WAS HEADBUTTED BY ANOTHER PLAYER WHILE PLAYING SOCCERDX NOSE FX
- Não é um infante; pessoa de idade desconhecida
- Mudar age = 0 para NA
- Existem outros casos com age = 0 ou próximo?

Quantos Casos Têm Idade Menos de 5 Anos

- Idade em que crianças começam escola

```
1 fm %>%
2   filter(age < 5) %>%
3   summarise(n = n())
# A tibble: 1 × 1
      n
  <int>
1     82
```

Medidas de Tendência Central

Media, Mediana e Moda

Media Aritmética, Geométrica e Harmonica

Mediana e Moda

Medidas de Dispersione

Variancia e Desvio Padrão

Coeficiente de Variação

Relações entre as Medidas de Tendência Central

Exercícios Resolvidos

Exercícios para Praticar

Respostas das Perguntas Finais

Respostas das Exercícios Resolvidos

Respostas das Exercícios para Praticar

Interesse em Pessoas que Jogam Futebol

- Quais tipos de lesões sofrem **amadores** jogando futebol
- Eliminar casos com idades menos de 5 anos

```
1 fm_mk2 <- fm %>%
2   filter(age >= 5)
3 summarytools::descr(fm_mk2$age)
```

Descriptive Statistics
fm_mk2\$age
N: 7521

age	
Mean	16.53
Std.Dev	8.86
Min	5.00
Q1	12.00
Median	14.00
Q3	17.00
Max	85.00
MAD	4.45
IQR	5.00
CV	0.54
Skewness	2.28
SE.Skewness	0.03
Kurtosis	6.78
N.Valid	7521.00
Pct.Valid	100.00

Médias de Duas Distribuições

- Média de `fm` (com pequenas crianças): 16.3786225
- `Média de fm_mk2` (sem pequenas crianças): 16.5261268
- Se removêssemos 82 casos, porque a diferença não é maior?

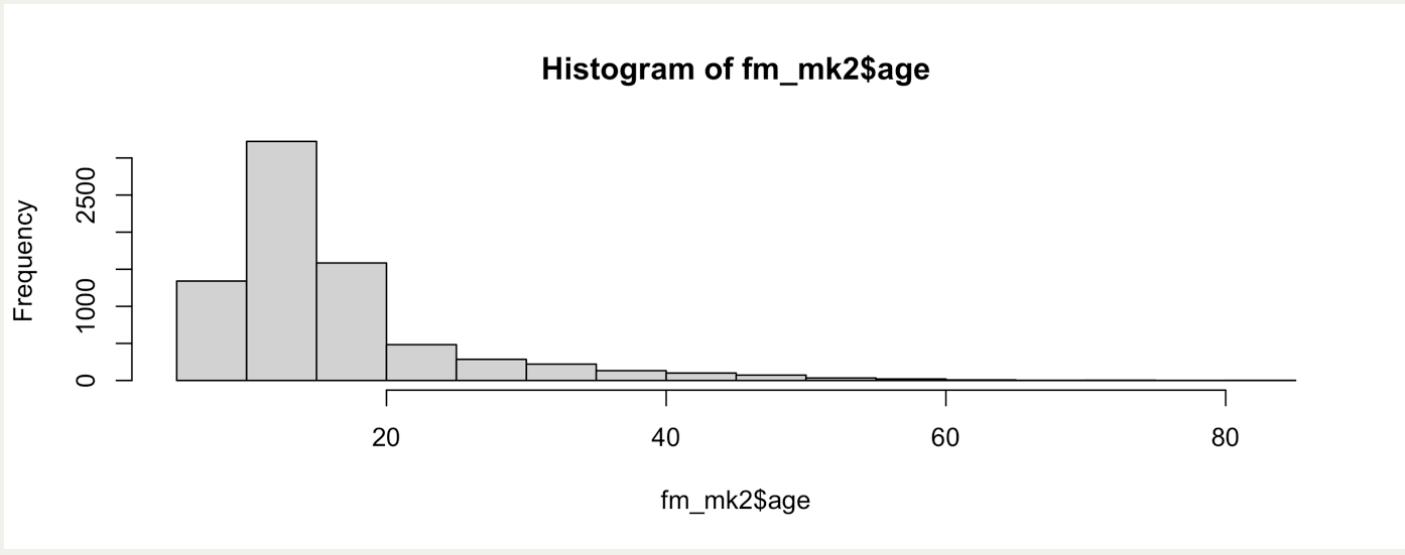
O Que É a Média?

- Um das medidas de **tendência central**
 - Valores que ficam no meio da distribuição
 - Valores populares
- O **centro aritmético** de uma distribuição
- Sensível aos valores extremos

$$\mu_x = \frac{\sum_{i=1}^n x_i}{n}$$

Visualização Clássica da Média de uma Distribuição - Histograma

```
1 hist(fm_mk2$age)
```



Histograma Foi Útil?

- Não deu muita informação
- Problema de **bins**
- Apresentação muito feia

Sistema Gráfico Alternativo

Grammar of Graphics - ggplot2

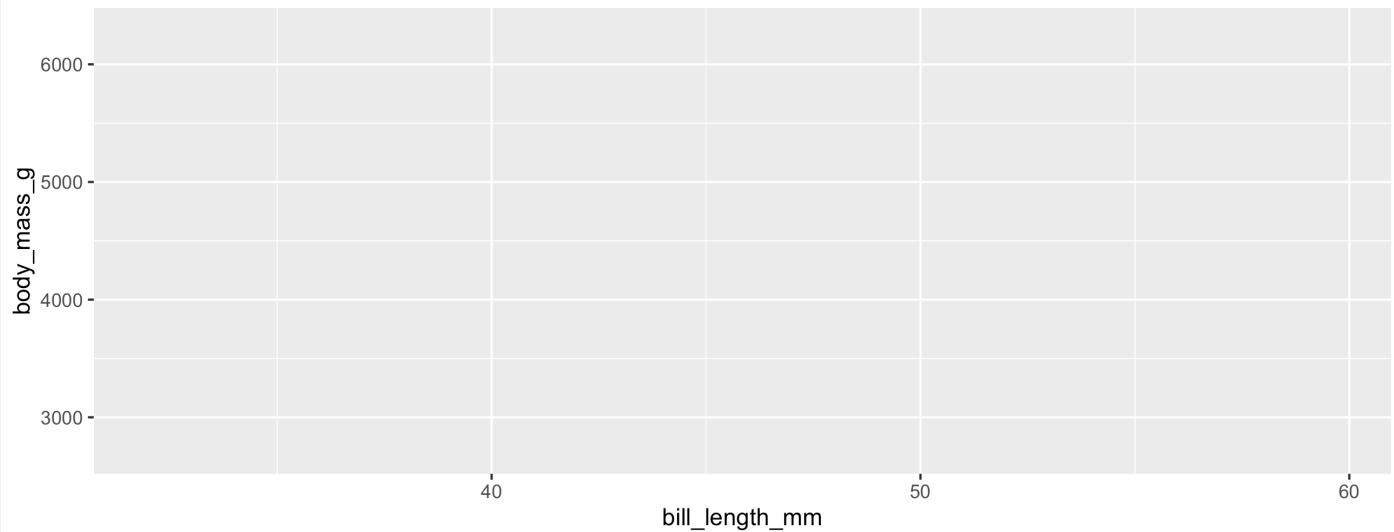
- Um sistema para **construir** gráficos (que se comunicam muito melhor)
- Um dos primeiros produtos de Hadley Wickham
- Construir seu gráfico camada por camada
- Começar por especificar um conjunto de dados: **penguin**
 - Variáveis **bill_length_mm** e **body_mass_g**

```
Rows: 333
Columns: 5
$ bill_length_mm      <dbl> 39.1, 39.5, 40.3, 36.7, 39.3, 38.9, 39.2, 41.1, 38.6...
$ bill_depth_mm       <dbl> 18.7, 17.4, 18.0, 19.3, 20.6, 17.8, 19.6, 17.6, 21.2...
$ flipper_length_mm  <dbl> 181, 186, 195, 193, 190, 181, 195, 182, 191, 198, 18...
$ body_mass_g         <dbl> 3750, 3800, 3250, 3450, 3650, 3625, 4675, 3200, 3800...
$ species             <chr> "Adelie", "Adelie", "Adelie", "Adelie", "Adelie", "A...
```

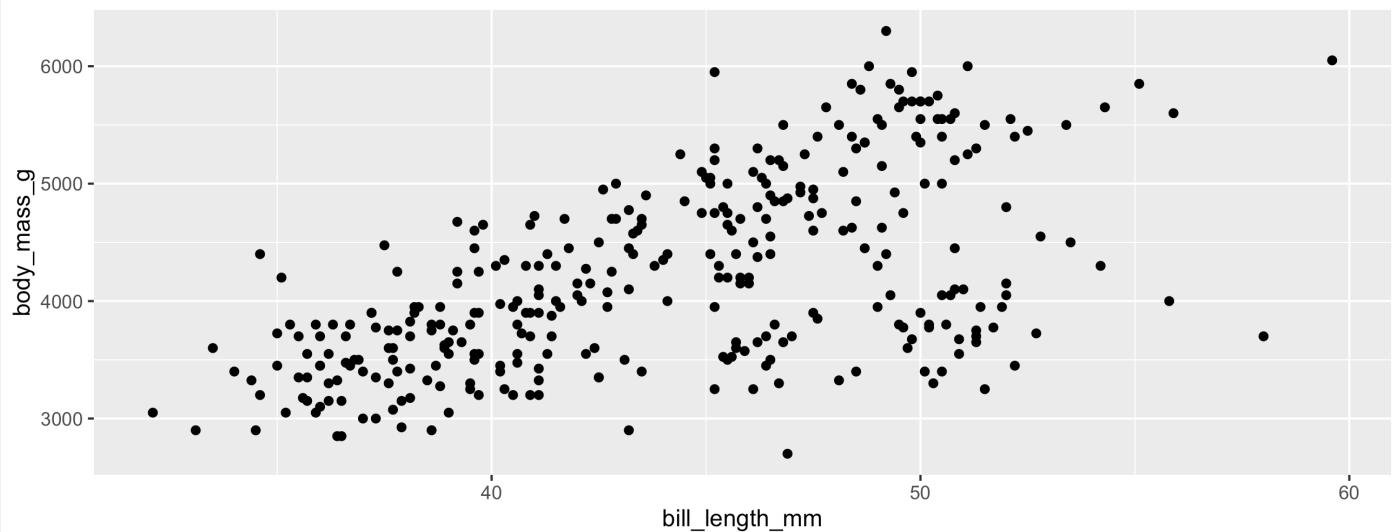
```
1 ggplot()
```

```
1 ggplot(data = pd)
```

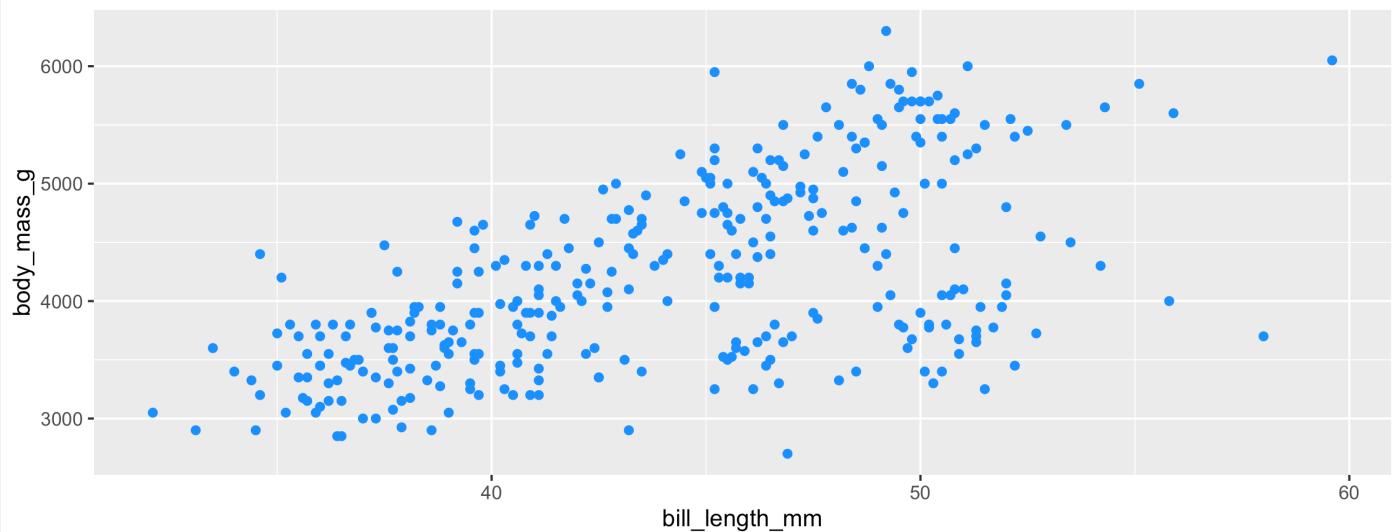
```
1 ggplot(data = pd, aes(x = bill_length_mm, body_mass_g ))
```



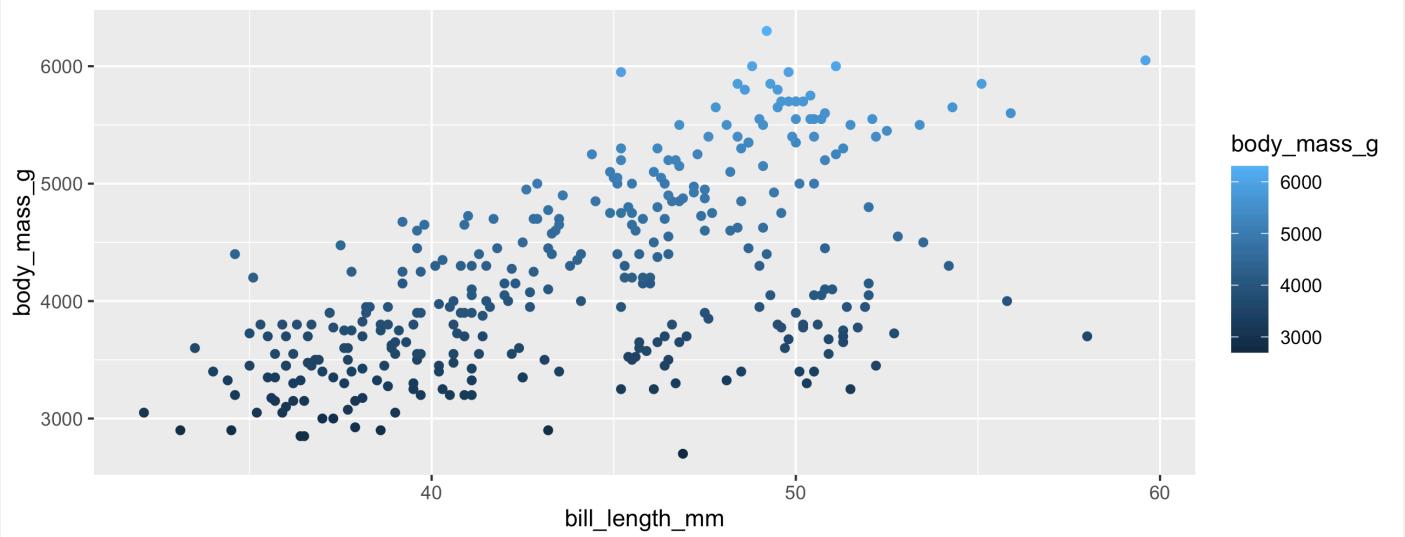
```
1 ggplot(data = pd, aes(x = bill_length_mm, body_mass_g )) +  
2   geom_point()
```



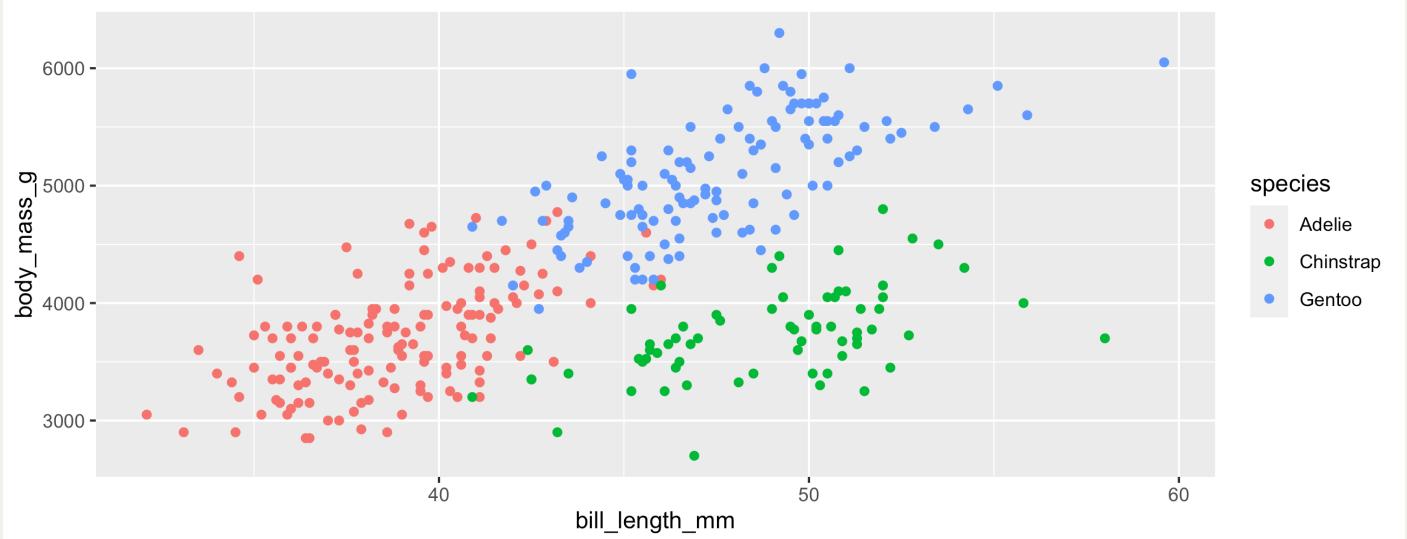
```
1 ggplot(data = pd, aes(x = bill_length_mm, body_mass_g )) +  
2   geom_point(color = "dodgerblue")
```



```
1 ggplot(data = pd, aes(x = bill_length_mm, body_mass_g, color = body_mass_g))  
2 geom_point()
```

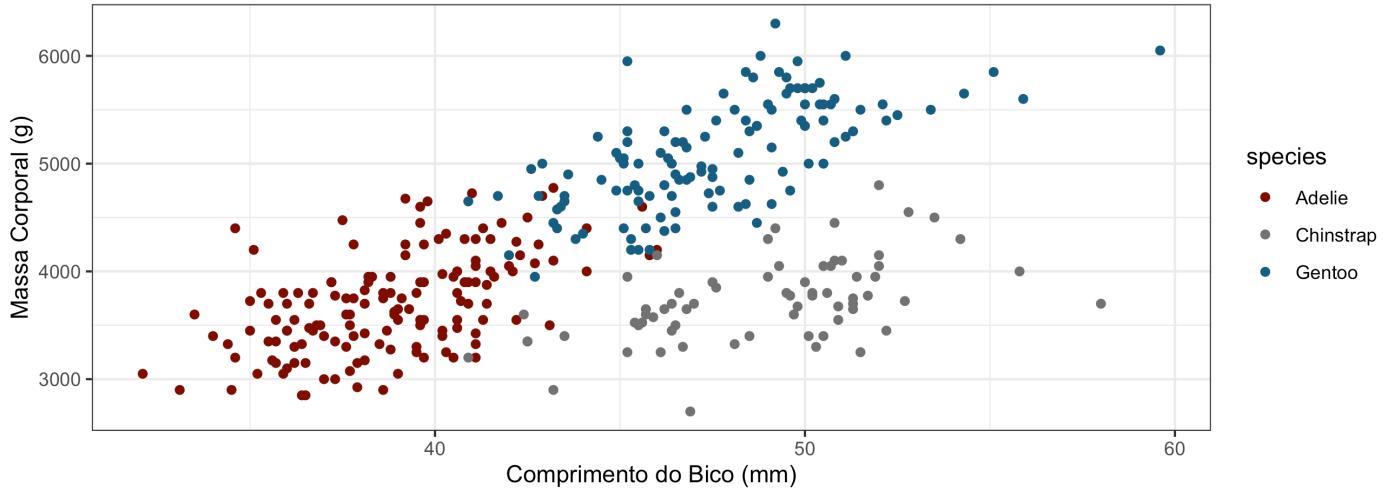


```
1 ggplot(data = pd, aes(x = bill_length_mm, body_mass_g, color = species )) +  
2   geom_point()
```



```
1 ggplot(data = pd, aes(x = bill_length_mm, body_mass_g, color = species)) +
2   geom_point() +
3   labs(title = "Uma Boa Introdução aos Gráficos de Dispersão", x = "Comprimento do Bico (mm)",
4        y = "Massa Corporal (g)") +
5   scale_colour_manual(values = c("#800000FF", "#767676FF", "#155F83FF")) +
6   theme_bw()
```

Uma Boa Introdução aos Gráficos de Dispersão



Recursos - `ggplot`

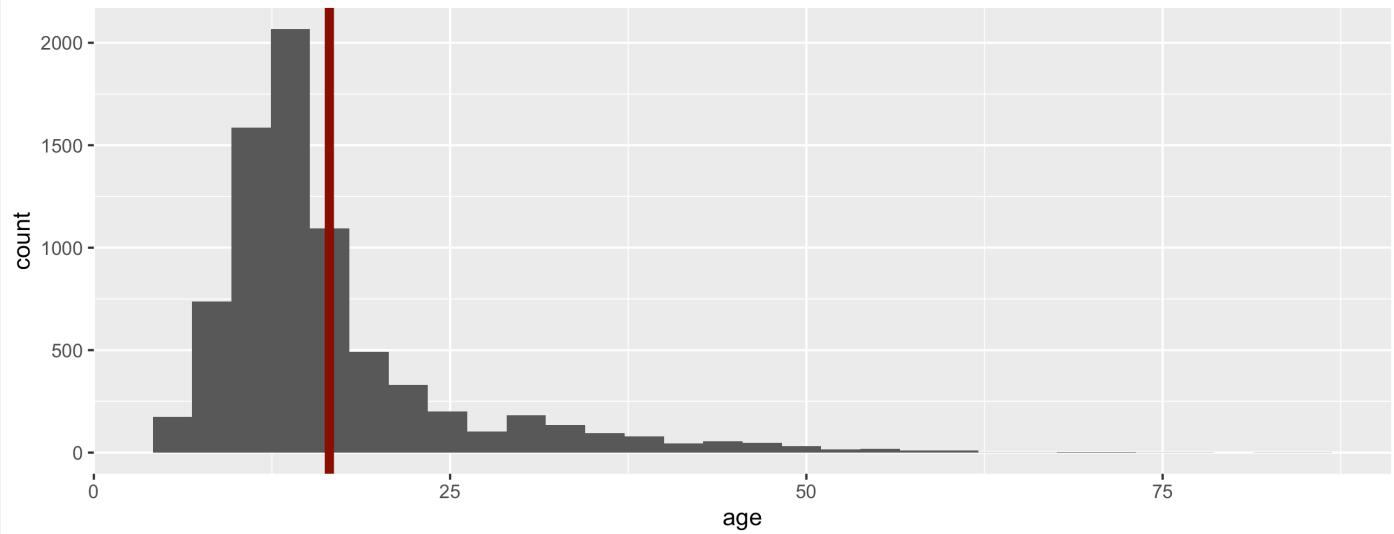
- Winston Chang, **R Graphics Cookbook**, 2Ed., <https://r-graphics.org>
- Kieran Healy, **Data Visualization: A Practical Introduction**, <https://socviz.co>
- [https://r-graph_gallery.com](https://r-graph-gallery.com) - examples of many types of graphs with explanations and code
- `ggplot` cheat sheet

Histograma de **age**

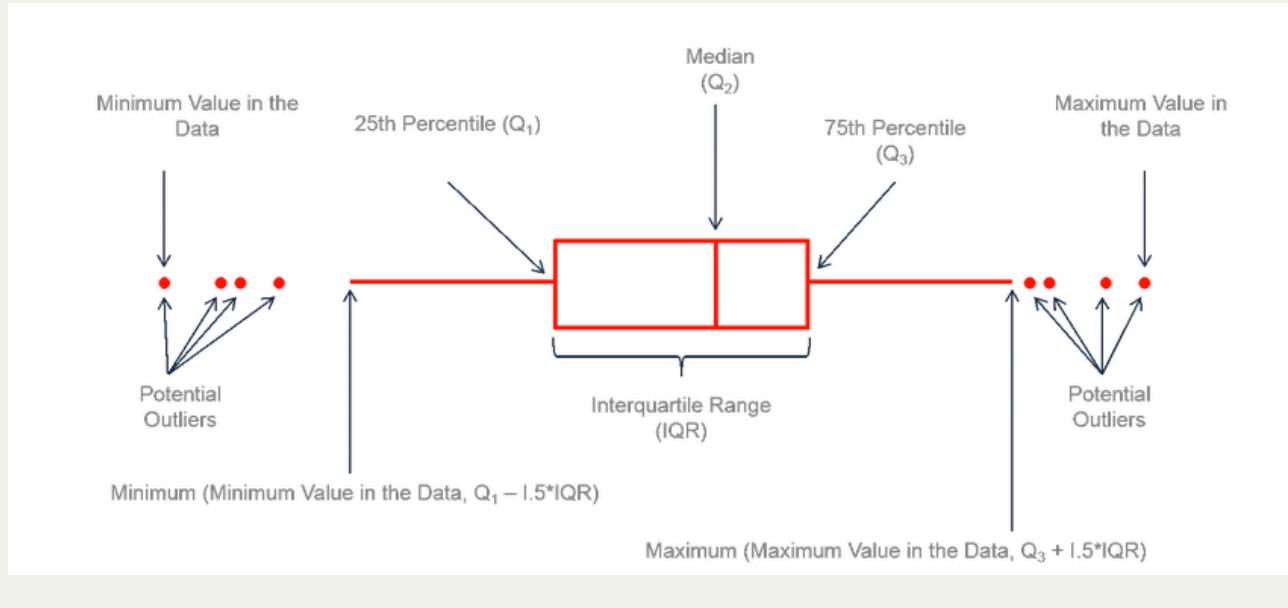
| Live Coding

Histograma de age

```
1 avg_age <- mean(fm_mk2$age)
2 ggplot(data = fm_mk2, aes(x = age)) +
3   geom_histogram(bins = 30) +
4   geom_vline(xintercept = avg_age, colour = "darkred", size = 2)
```



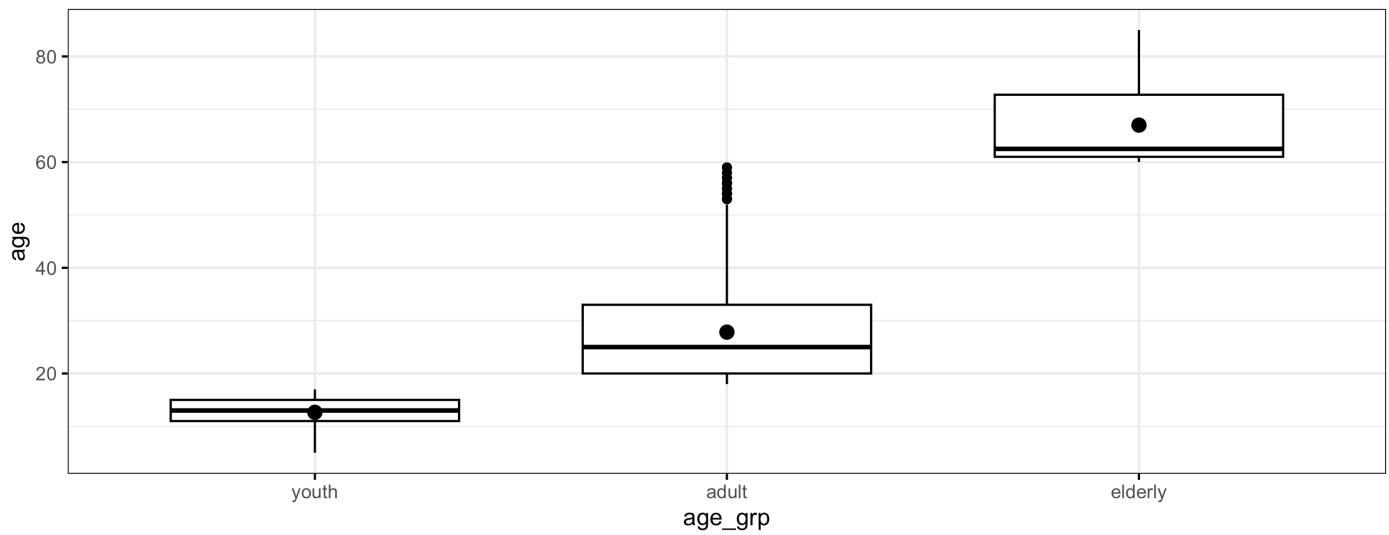
Segundo Gráfico que Mostra Distribuições Bem - Boxplot



- source: <https://r-graph-gallery.com>

Boxplot com os Dados de Futebol

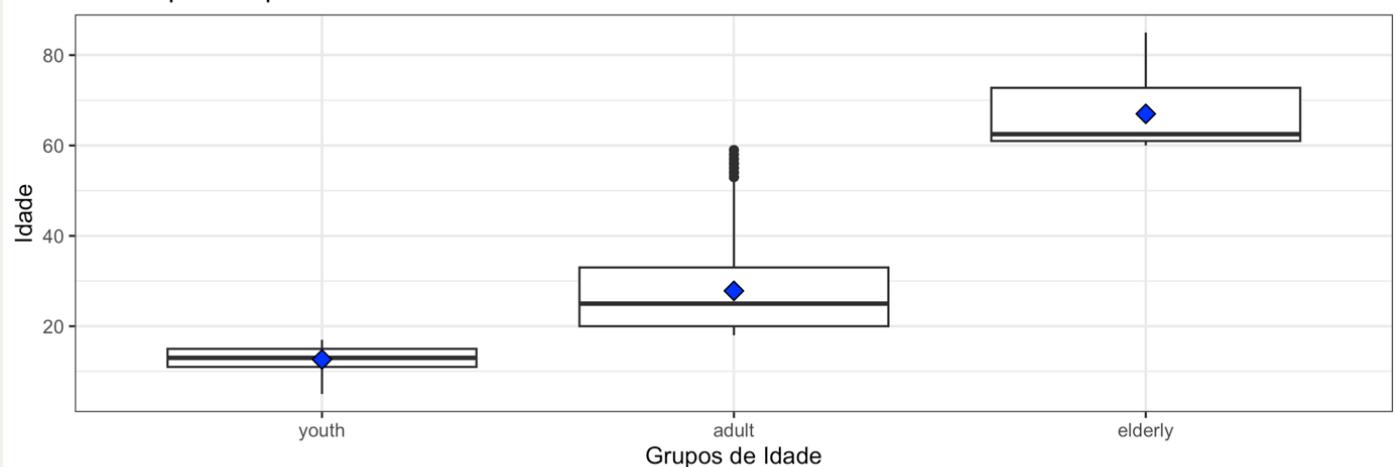
```
1 fm_mk2 |> ggpubr::ggboxplot(x = "age_grp",
2                               y = "age",
3                               add = "mean",
4                               ggtheme = theme_bw())
```



Boxplot de Futebol com ggplot

```
1 fm_mk2 |>
2   ggplot(mapping = aes(x = age_grp,    y = age,)) +
3     geom_boxplot() +
4     stat_summary(fun = "mean", geom = "point", shape = 23,
5                   size = 3, fill = "blue") +
6     labs(title = "Idades por Grupo de Idade",
7           x = "Grupos de Idade",
8           y = "Idade",
9           caption = "Texto que explica o gráfico.") +
10    theme_bw()
```

Idades por Grupo de Idade



Texto que explica o gráfico.

De Onde Veio Esta Informação sobre a Média?

6.8.1 Problem

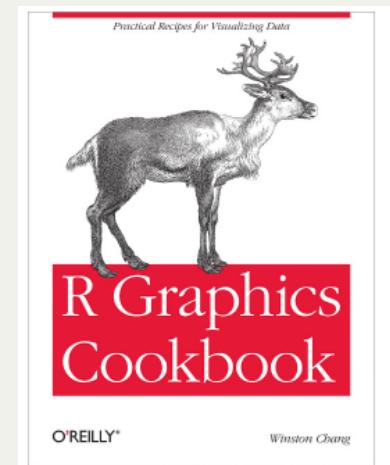
You want to add markers for the mean to a box plot.

6.8.2 Solution

Use `stat_summary()`. The mean is often shown with a diamond, so we'll use shape 23 with a white fill. We'll also make the diamond slightly larger by setting `size = 3` (Figure 6.21):

```
library(MASS) # Load MASS for the birthwt data set

ggplot(birthwt, aes(x = factor(race), y = bwt)) +
  geom_boxplot() +
  stat_summary(fun.y = "mean", geom = "point", shape = 23, size = 3, fill = "white")
#> Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
#> i Please use the `fun` argument instead.
#> This warning is displayed once every 8 hours.
#> Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
#> generated.
```

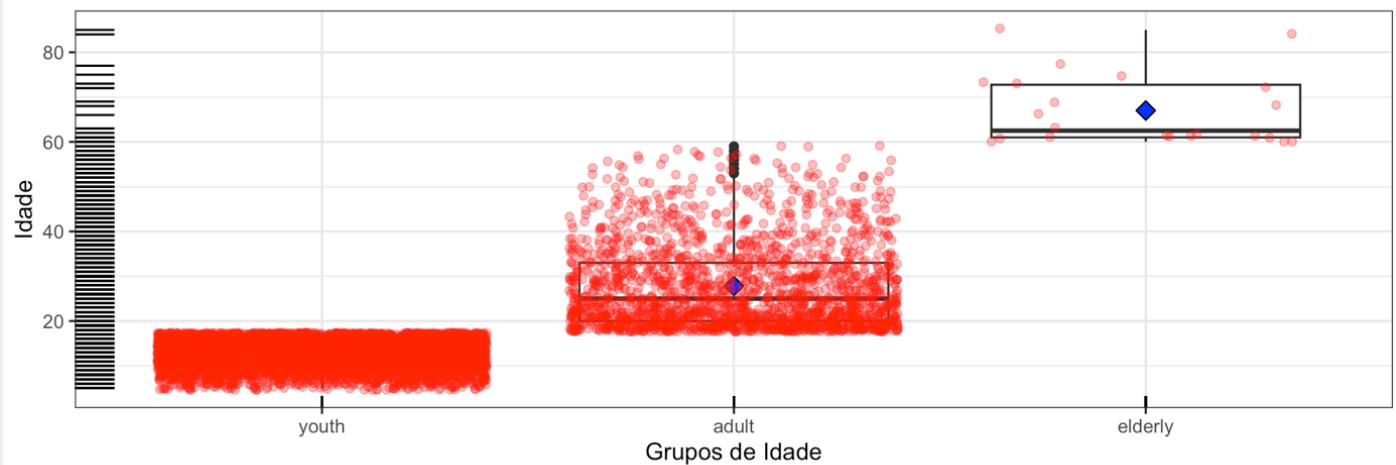


ggplot Boxplot - 2

- Gostaria de saber onde cai os pontos
 - **geom_jitter** - mostra todos os pontos com um pouco de variação
 - **geom_rug** - mostra os casos individuais numa certa dimensão

```
1 fm_mk2 |>
2   ggplot(mapping = aes(x = age_grp, y = age,)) +
3     geom_boxplot() +
4     stat_summary(fun = "mean", geom = "point", shape = 23, size = 3, fill = "blue") +
5     geom_jitter(alpha = .3, color = "red") +
6     geom_rug() +
7     labs(title = "Idades por Grupo de Idade",
8           x = "Grupos de Idade",
9           y = "Idade",
10          caption = "Texto que explica o gráfico.") +
11        theme_bw()
```

Idades por Grupo de Idade

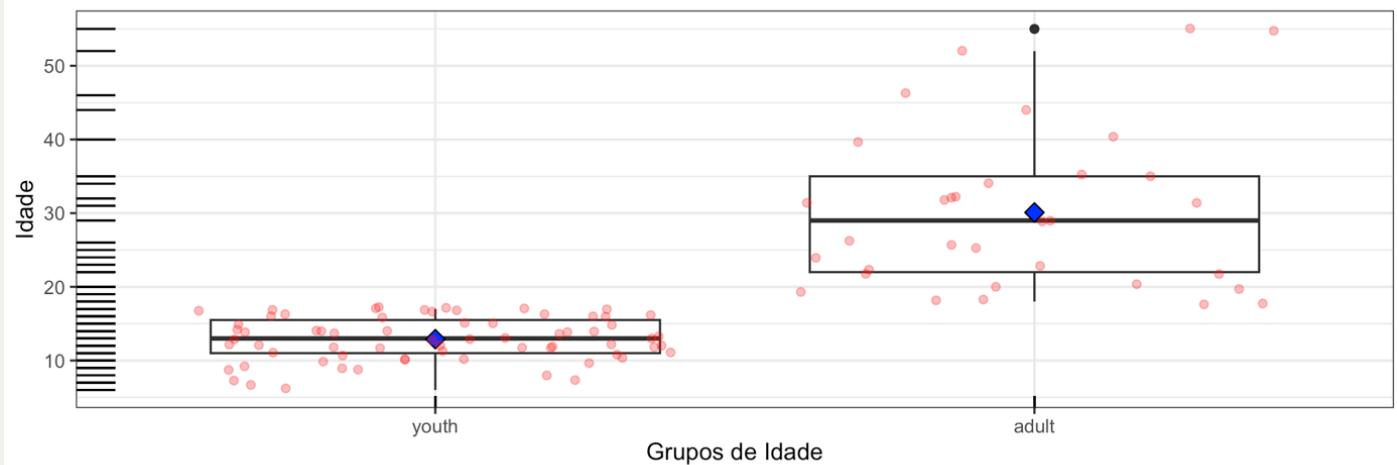


Lição Disso

- Com muitos pontos, impossível ver a distribuição
- Criar uma versão com só 100 pontos

```
1 fm_mk2 |>
2   slice_sample(n = 100) |>
3   ggplot(mapping = aes(x = age_grp,    y = age,)) +
4     geom_boxplot() +
5     stat_summary(fun = "mean", geom = "point", shape = 23, size = 3, fill = "red") +
6     geom_jitter(alpha = .3, color = "red") +
7     geom_rug() +
8     labs(title = "Idades por Grupo de Idade",
9           x = "Grupos de Idade",
10          y = "Idade",
11          caption = "Texto que explica o gráfico.") +
12     theme_bw()
```

Idades por Grupo de Idade



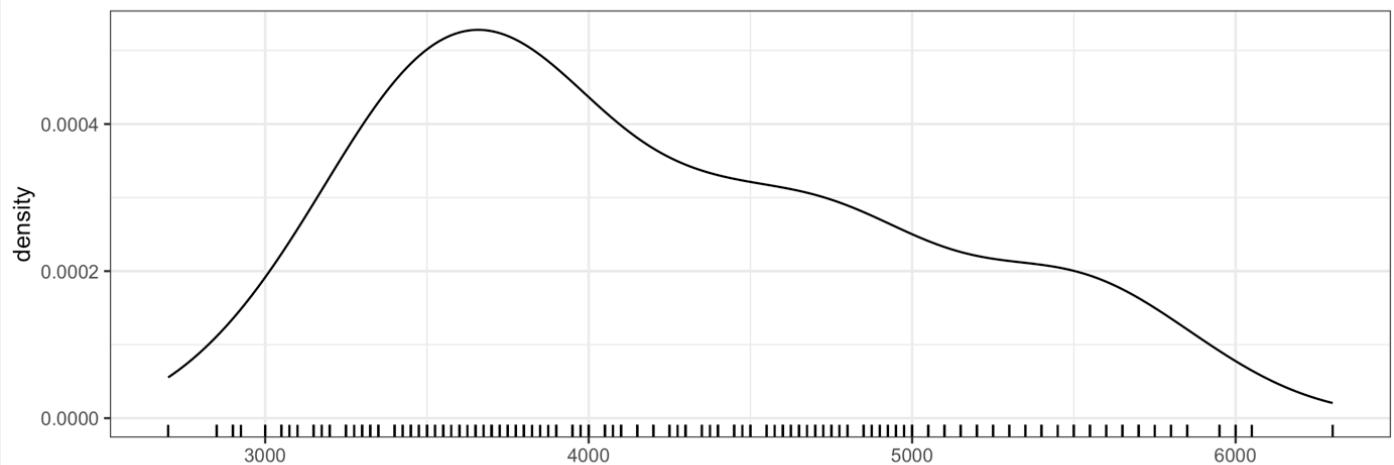
Texto que explica o gráfico.

Terceiro Gráfico das Distribuições - Plotagem de Densidade

- Tecnicamente, uma plotagem de densidades de *kernels*
 - *Kernel Density* divide a distribuição em partes e calcular a densidade em cada região não-linearmente e recombinar elas para compôr uma curva suave
- Usar `body_mass_g` para ilustrar

```
1 set.seed(42)
2 pd |> # dataframe penguins |>
3   ggplot(mapping = aes(x = body_mass_g)) +
4   geom_density() +
5   labs(title = "Massa Corporal",
6        x = "") +
7   geom_rug()+
8   theme_bw()
```

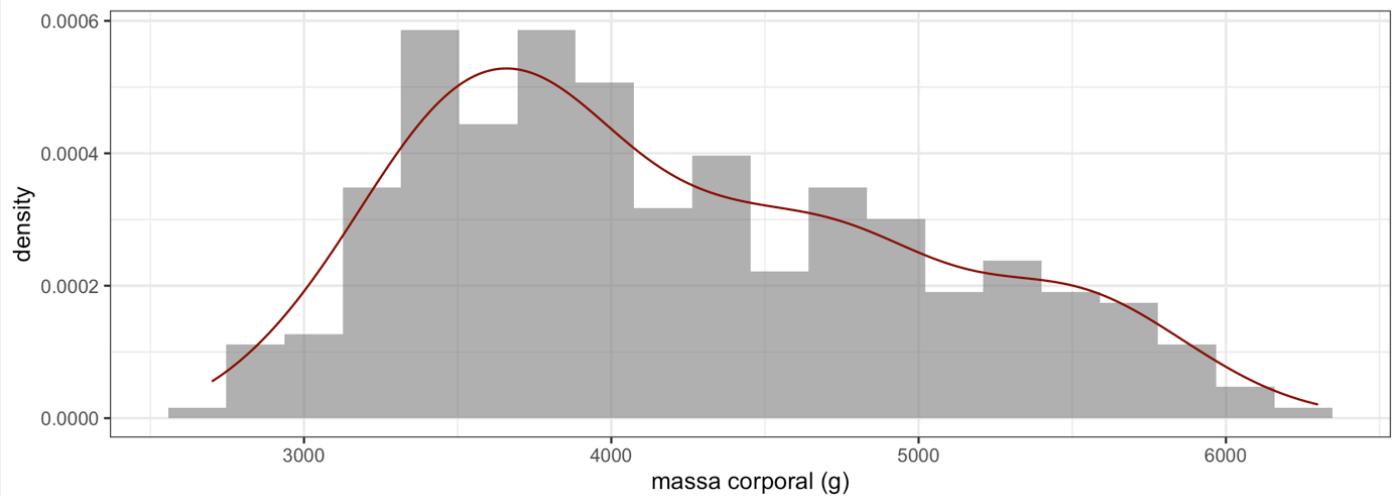
Massa Corporal



Densidade com Histograma

```
1 pd |> # dataframe penguins |>
2   ggplot(mapping = aes(x = body_mass_g)) +
3   geom_histogram(aes(y = ..density..), bins = 20, alpha = 0.5) +
4   geom_density(colour = "darkred") +
5   labs(title = "Massa Corporal",
6        x = "massa corporal (g)") +
7   theme_bw()
```

Massa Corporal



Comparar As Espécies

```
1 set.seed(42)
2 pd |> # dataframe penguins |>
3   group_by(species) |>
4   ggplot(mapping = aes(x = body_mass_g, colour = species, fill = species))
5   geom_density(alpha = 0.5) +
6   labs(title = "Massa Corporal",
7        x = "") +
8   geom_rug()+
9   theme_bw()
```

Massa Corporal

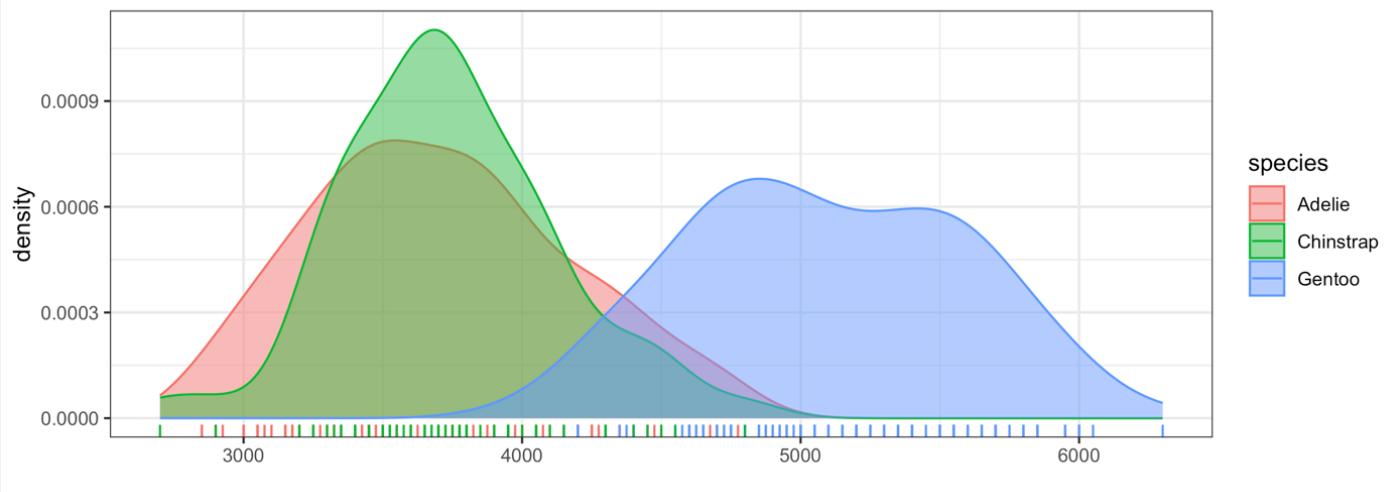
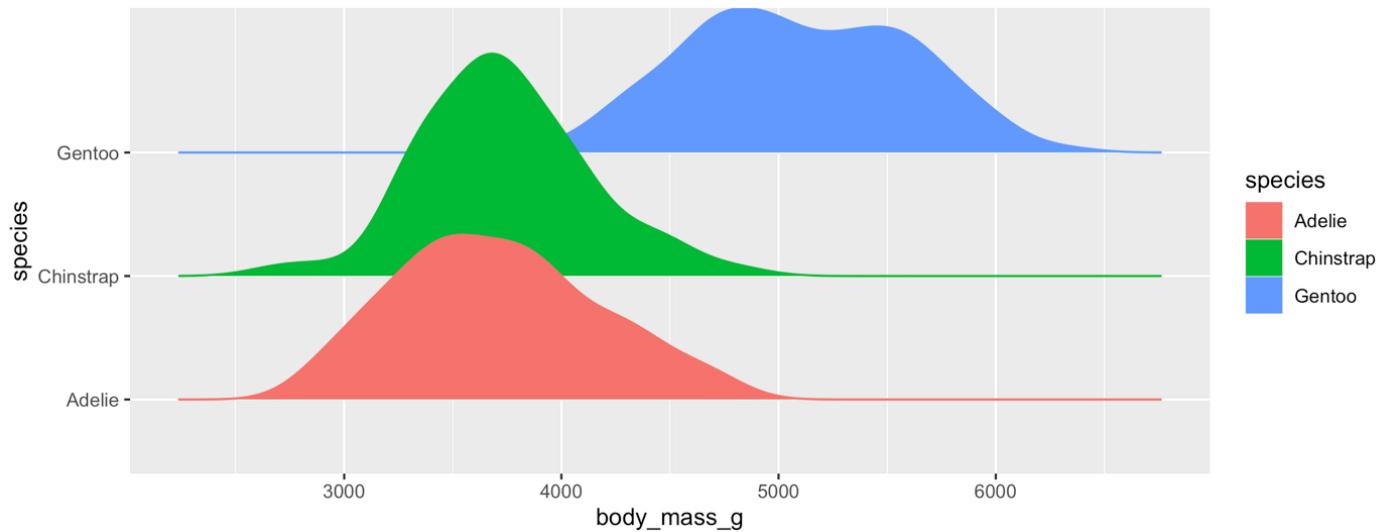


Gráfico *Ridgeline* - Extensão de Densidade

- Maneira fácil para comparar as densidades de várias categorias das variáveis
- Precisa instalar **ggridges** de CRAN

```
1 library(ggridges)
2
3 pd |>
4   ggplot(aes(x = body_mass_g,
5             y = species,
6             colour = species,
7             fill = species)) +
8   geom_density_ridges() +
9   theme_gray() # default theme for ggplot
```



Ridgeline com Mais Ooomph

- Mudar cores para uma paleta mais agradável
 - Usar `ggsci` paleta `uchicago`
- Tirar a legenda - desnecessária
- Reduz o tamanho das caudas
- Mostrar os quartis nas curvas de densidade

```
1 library(ggridges)
2
3 pd |>
4   ggplot(aes(x = body_mass_g,
5             y = species,
6             fill = species)) +
7   stat_density_ridges(quantile_lines = TRUE, rel_min_height = 0.01) +
8   scale_fill_uchicago(palette = "default", alpha = 0.8) +
9   guides(fill = FALSE) +
10  labs(title = "Massa Corporal por Espécie",
11        x = "Massa Corporal (g)",
12        y = "") +
13  theme_gray() # default theme for ggplot
```

Massa Corporal por Espécie

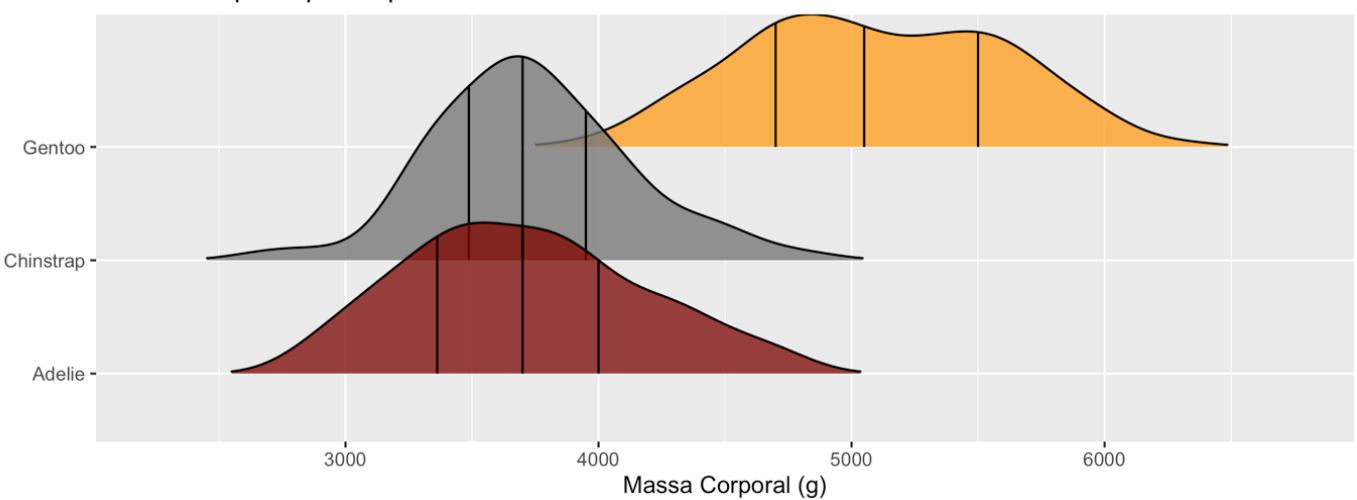


Gráfico Radar - Outra Maneira de Retratar Dimensões

- Cria um campo circular para mostrar um número de dimensões
- Funciona melhor comparando poucas classes
- Precisa preparar os dados para utilizar este tipo de gráfico
 - Comparando ou a média ou mediana das classes em cada dimensão
 - Vai pôr as dimensões na escala de 0 até 1 utilizando `scales::rescale()`
- Usa pacote `ggradar`
 - `remotes::install_github("ricardo-bion/ggradar")`

Radar - Preparação dos Dados

```
1 pacman::p_load(ggradar, scales)
2
3 pd_radar <- pd |>
4   tidyr::drop_na() |>      # NAs can't be processed in ops below
5   group_by(species) |>
6   summarise(
7     avg_bill_length = mean(bill_length_mm),
8     avg_bill_depth = mean(bill_depth_mm),
9     avg_flipper_length = mean(flipper_length_mm),
10    avg_body_mass = mean(body_mass_g)
11  ) |>
12  ungroup() |>
13  mutate_at(vars(-species), rescale)
14 pd_radar
```

A tibble: 3 × 5

	species	avg_bill_length	avg_bill_depth	avg_flipper_length	avg_body_mass
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	Adelie	0	0.979	0	0
2	Chinstrap	1	1	0.211	0.0194
3	Gentoo	0.874	0	1	1

Código do Gráfico

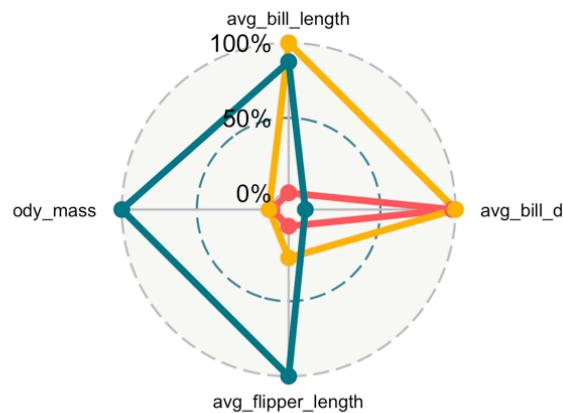
```
1 ggpengrad <- pd_radar %>%
2   ggradar(
3     font.radar = "arial",
4     grid.label.size = 5, # Affects the grid annotations (0%, 50%, etc.)
5     axis.label.size = 3, # Afftects the names of the variables
6     group.point.size = 3, # Simply the size of the point
7     legend.title = "Espécie",
8     plot.title = "Características - Pinguins Palmer",
9   )
```

Resultado

Características - Pinguins Palmer

Espécie

- Adelie
- Chinstrap
- Gentoo



Muitos Outros Tipos dos Gráficos

Live Coding das Outras Opções Destes Tipos