Gráficos

Beatriz Magiore

2025-11-13

GGPLOT

O conjunto de dados ***Pokemon*** é composto de variáveis quantitativas e qualitativas. A tabela a seguir mostra o resumo dos dados:

```
##      name             pokedex.id        height           weight
##  Length:890         Min.   :  1.0   Min.   :  1.00   Min.   :   1.0
##  Class :character   1st Qu.:223.2   1st Qu.:  5.00   1st Qu.:  85.0
##  Mode  :character   Median :445.5   Median : 10.00   Median : 268.0
##                     Mean   :445.5   Mean   : 11.82   Mean   : 629.9
##                     3rd Qu.:667.8   3rd Qu.: 15.00   3rd Qu.: 630.0
##                     Max.   :890.0   Max.   :200.00   Max.   :9999.0
##      type           secundary.type          hp            attack
##  Length:890         Length:890         Min.   :  1.0   Min.   :  5.0
##  Class :character   Class :character   1st Qu.: 50.0   1st Qu.: 55.0
##  Mode  :character   Mode  :character   Median : 65.0   Median : 75.0
##                                        Mean   : 68.7   Mean   : 76.3
##                                        3rd Qu.: 80.0   3rd Qu.: 95.0
##                                        Max.   :255.0   Max.   :181.0
##     defense           sp.atk           sp.def           speed
##  Min.   :  5.00   Min.   : 10.00   Min.   : 20.00   Min.   :  5.00
##  1st Qu.: 50.00   1st Qu.: 45.25   1st Qu.: 50.00   1st Qu.: 45.00
##  Median : 67.00   Median : 65.00   Median : 65.00   Median : 65.00
##  Mean   : 71.82   Mean   : 69.55   Mean   : 69.86   Mean   : 65.65
##  3rd Qu.: 90.00   3rd Qu.: 90.00   3rd Qu.: 85.00   3rd Qu.: 85.00
##  Max.   :230.00   Max.   :173.00   Max.   :230.00   Max.   :160.00
```

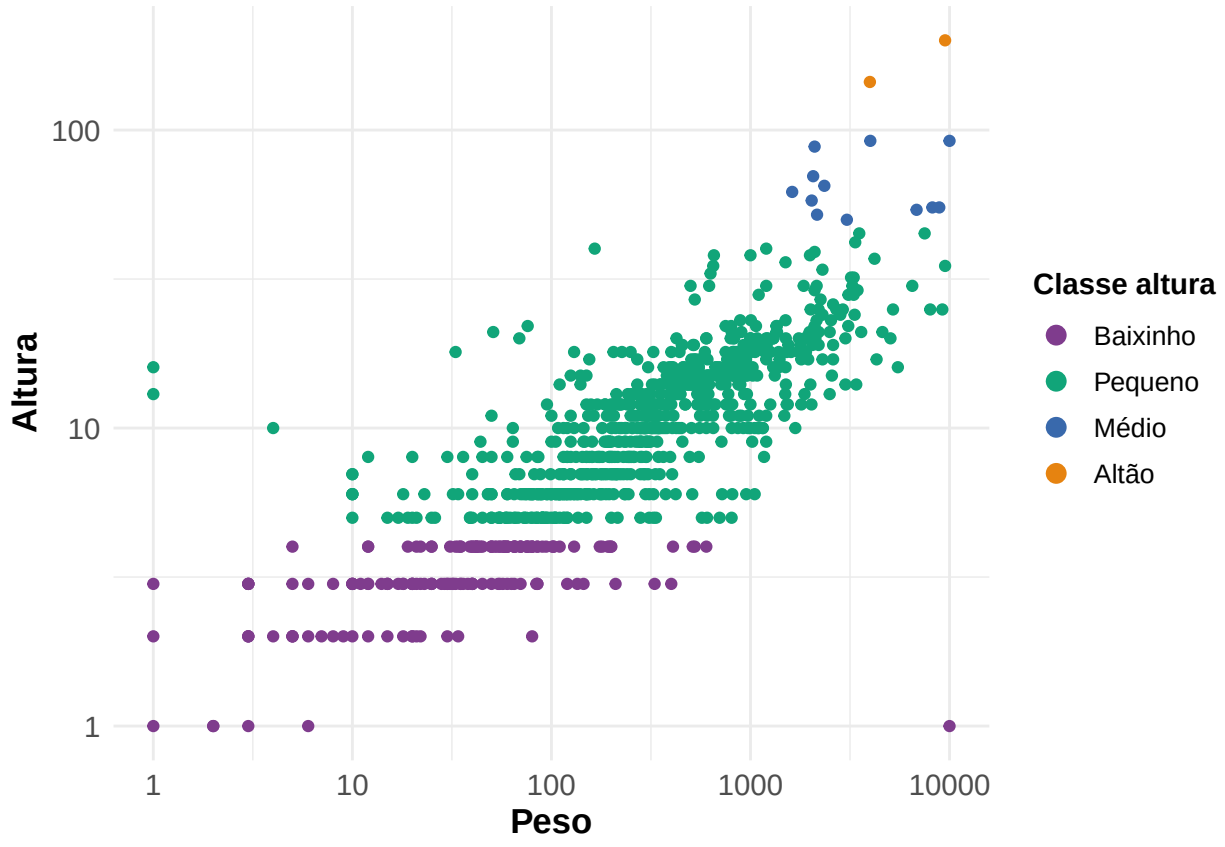
Box-plot

```
df <-  dados %>%
  mutate(
    classe_altura = case_when(
      height < 5 ~ "Baixinho",
      height < 50 ~ "Pequeno",
      height < 100 ~ "Médio",
      TRUE ~ "Altão"
    )
  ) %>%
  mutate(
    classe_altura = factor(classe_altura,
                           levels = c("Baixinho", "Pequeno", "Médio", "Altão"))
  )
```

```r
ggplot(df, aes(x = classe_altura, y = weight, fill = classe_altura))+
  geom_violin()+
  geom_boxplot(width = 0.2, fill = "white")+
    labs(
      x = "Classe altura",
      y = "Peso"
    )+
    scale_y_continuous(trans = "log10")+
    scale_fill_manual(values = cores[c(1,2,3,7)])+
    theme_minimal()+
    theme(
      axis.text = element_text(size = 11),
      axis.title = element_text(size = 13,face = "bold"),
      legend.position = "none"
    )
```

Gráfico de pontos

```
df %>%
  arrange(desc(classe_altura)) %>%
  ggplot(aes(x = weight, y = height, color = classe_altura))+
  geom_point()+
    labs(
      x = "Peso", y = "Altura", color = "Classe altura"
    )+
    guides(color = guide_legend(override.aes = list(size = 3)))+
    scale_x_continuous(trans = "log10")+
    scale_y_continuous(trans = "log10")+
    scale_color_manual(values = cores[c(1,2,3,7)])+
    theme_minimal()+
    theme(
      axis.text = element_text(size = 11),
      axis.title = element_text(size = 13, face = "bold"),
      legend.position = "right",
      legend.text = element_text(size = 10),
      legend.title = element_text(size = 11, face = "bold")
    )
```



Gráficos facetados

```
df %>%
  group_by(classe_altura) %>%
  summarise(
    media_h = mean(height),
    media_w = mean(weight)
  ) %>%
  pivot_longer(cols = c(2,3), values_to = "Valor", names_to = "Tipo") %>%
  mutate(
    Tipo = recode(Tipo,
                  "media_h" = "Média Altura",
                  "media_w" = "Média Peso")
  ) %>%
  ggplot(aes(x = Tipo, y = Valor, fill = classe_altura))+
  geom_col()+
  labs(
      x = "Tipo", y = "Valor", fill = "Classe Altura"
    )+
  facet_wrap(.~classe_altura)+
  theme_minimal()+
  theme(
      axis.text = element_text(size = 11),
      axis.title = element_text(size = 13,face = "bold"),
      legend.position = "none"
    )
```

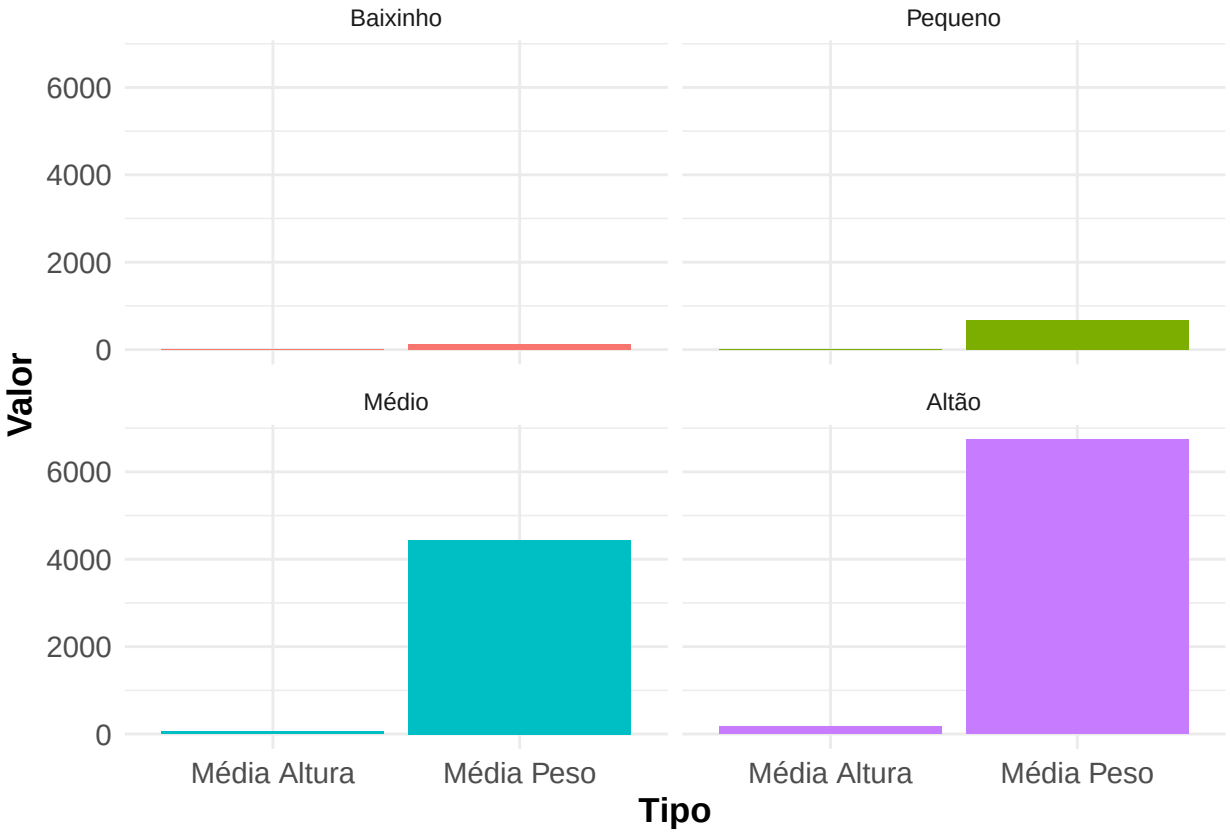


Gráfico de dois eixos

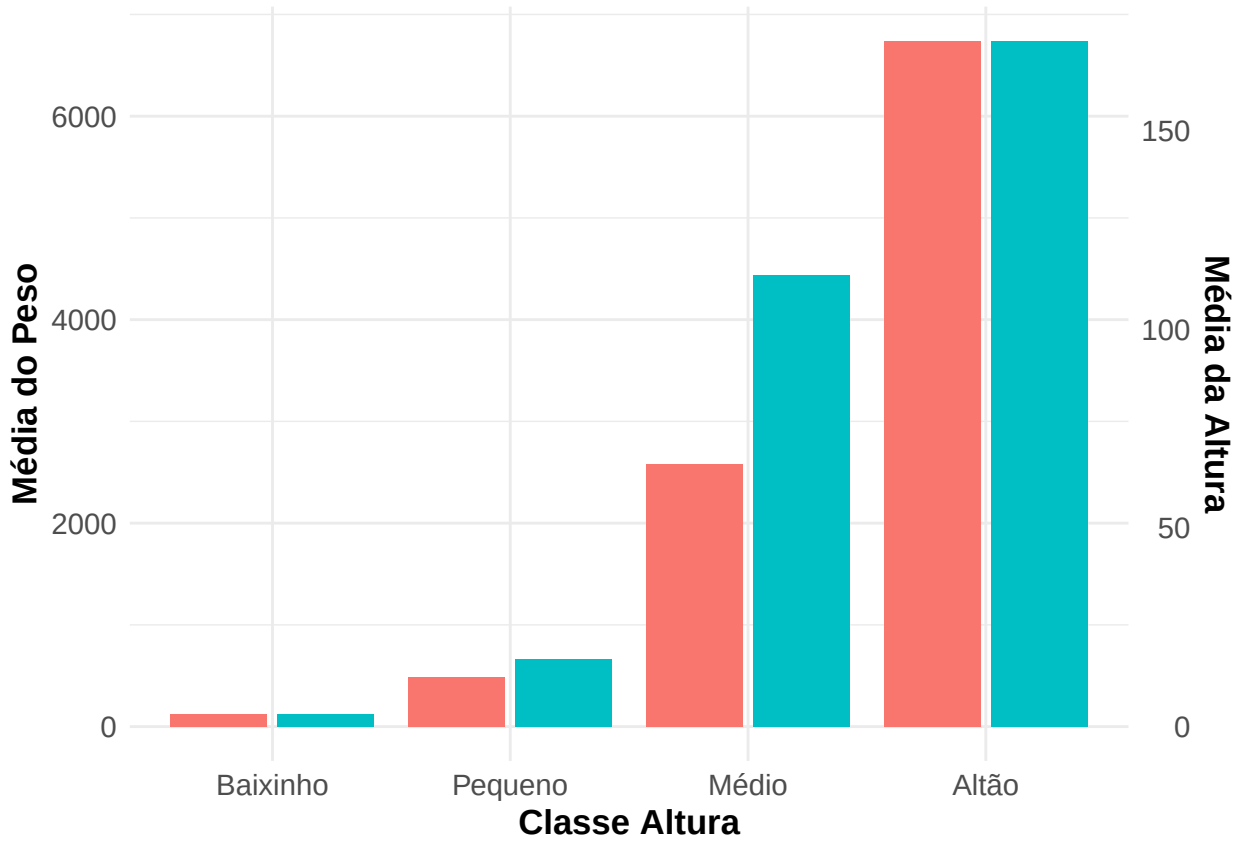
```
df_media <- df %>%
  group_by(classe_altura) %>%
  summarise(
    media_h = mean(height),
    media_w = mean(weight)
  ) %>%
  pivot_longer(cols = c(2,3), values_to = "Valor", names_to = "Tipo") %>%
  mutate(
    Medida = case_when(
      grepl("_h", Tipo) ~ "Altura",
      TRUE ~ "Peso"
    )
  )

max_altura <- df_media %>%
  filter(Medida == "Altura") %>%
  arrange(desc(Valor)) %>%
  distinct(Medida, .keep_all = TRUE) %>%
  pull(Valor)

max_peso <- df_media %>%
  filter(Medida == "Peso") %>%
  arrange(desc(Valor)) %>%
  distinct(Medida, .keep_all = TRUE) %>%
  pull(Valor)

fator = max_peso/max_altura

df_media %>%
  mutate(
    Valor2 = ifelse(Medida == "Altura", Valor*fator, Valor)
  ) %>%
  ggplot(aes(x = classe_altura, y = Valor2, fill = Medida))+
  geom_col(position = position_dodge2())+
    labs(
      x = "Classe Altura", y = "Média do Peso", fill = "Média"
    )+
    scale_y_continuous(sec.axis = sec_axis(~./fator, name = "Média da Altura"))+
    theme_minimal()+
    theme(
      axis.text = element_text(size = 11),
      axis.title = element_text(size = 13,face = "bold"),
      legend.position = "none"
    )
```


6000
4000
2000
0
150
100
50
0
Média do Peso
Média da Altura
Baixinho
Pequeno
Médio
Altão
Classe Altura

Mapa dengue em SP

```
## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
dados <- read.csv2("Dados/dengue_sp.csv")
head(dados)
```

```
##    Município.de.residência X2014 X2015 X2016 X2017 X2018 X2019 X2020 X2021
## 1  MUNICIPIO IGNORADO - SP     1     -     -     -     -     -     -     -
## 2        350010 ADAMANTINA    40  1141    98     8     8  2085  1053    10
## 3            350020 ADOLFO     5   121   115     1   127   131    10   117
## 4             350030 AGUAI   545  2467    26     6    15   677  2128    21
## 5    350040 AGUAS DA PRATA     6   334     2     3     1    29    27     1
## 6  350050 AGUAS DE LINDOIA    11   166    26     2     1    27     4     4
##   X2022 X2023 X2024 Total
## 1     -     -     1     2
## 2  1333  3358   172  9306
## 3   191    19   185  1022
## 4    10    12   598  6505
## 5     -    11   265   679
## 6    22     5  1362  1630
```

```
dados <- dados %>%
  rename(municipio = 1)
```

Garantindo que o conjunto de dados só tenha municípios válidos

```
dados <- dados %>%
  filter(grepl("^\\d{6}", municipio)) %>%
  mutate(across(starts_with("x"), as.integer)) %>%
  replace(is.na(.), 0)
```

```
## Warning: There were 10 warnings in `mutate()`.
## The first warning was:
## i In argument: `across(starts_with("x"), as.integer)`.
## Caused by warning:
## ! NAs introduzidos por coerção
## i Run `dplyr::last_dplyr_warnings()` to see the 9 remaining warnings.
```

```
df <- dados %>%
  pivot_longer(2:13, names_to = "Ano", values_to = "Casos") %>%
  filter(Ano != "Total") %>%
  mutate(Ano = str_remove(Ano, "X")) %>%
  mutate(Cod = str_extract(municipio, "\\d{6}"))
```

Lendo dados da população

```
pop <- read.csv2("Dados/tabela4709.csv", skip = 3)
names(pop) <- c("Cod", "Municipio", "Populacao")

pop <- pop %>%
  mutate(Cod = str_extract(Cod, "\\d{6}"))
```

```
df <- df %>%
  left_join(pop) %>%
  mutate(
    taxa = Casos/Populacao*10000
  )
```

```
## Joining with `by = join_by(Cod)`
```

Lendo o arquivo SHP

```
sp_shp <- read_sf("Dados/SHP/cidades_SP.shp")
```

Juntar os dados

```
df <- sp_shp %>%
  mutate(Cod = str_extract(CD_GEOCODM, "\\d{6}")) %>%
  select(Cod, geometry) %>%
  right_join(df)
```

```
## Joining with `by = join_by(Cod)`
```

Gráfico

```
ggplot(df)+
  geom_sf(aes(color = taxa))+
  scale_colour_viridis_c()+
  labs(colour = "Taxa de incidencia")+
  facet_wrap(.~Ano)+
  theme_minimal()
```