Projeto_Pokemon

Thomas Raphael Zonta

2025-02-20

```
knitr::opts_chunk$set(echo = TRUE)
```

Bibliotecas

Essas foram as bibliotecas importadas para a realização desta análise.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(readr)
library(dplyr)
library(summarytools)
```

```
##
## Anexando pacote: 'summarytools'
##
## O seguinte objeto é mascarado por 'package:tibble':
##
##     view
```

```
library(readxl)
library(knitr)
library(dlookr)
```

```
## Registered S3 methods overwritten by 'dlookr':
##   method          from
##   plot.transform  scales
```

```
##   print.transform scales
##
## Anexando pacote: 'dlookr'
##
## O seguinte objeto é mascarado por 'package:tidyr':
##
##     extract
##
## O seguinte objeto é mascarado por 'package:base':
##
##     transform
```

Base de dados Pokemon

A ideia do projeto é dentro da base de dados estipular quais pokemons dentro das 6 gerações catalogadas são os mais fortes utilizando a coluna que registra os dados de ataque básico. A seguir vamos carregas a nossa base de dados:

```
df <- read.csv("C:/Users/Pichau/Documents/RSTUDIO/Estatistica_para_Ciencia_Dados/pokemon.csv", stringsAs
```

Análises iniciais da tabela

Contagem do número de linhas da tabela.

```
count(df)
```

```
##     n
## 1 800
```

Visualização das colunas da tabela.

Vamos visualizar o topo da base de dados para vermos os dados que teremos para trabalhar.

```
kable(head(df))
```

X.	Name	Type.1	Type.2	HP	Attack	Defense	Sp..Atk	Sp..Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
4	Mega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
5	Charmander	Fire		39	52	43	60	50	65	1	False
6	Charmeleon	Fire		58	64	58	80	65	80	1	False

Tipos de dados da base de dados

Vamos visualizar o tipo de dados de cada coluna da nossa tabela.

```r
df %>% dlookr::diagnose()
```

```
## # A tibble: 12 x 6
##    variables  types     missing_count missing_percent unique_count unique_rate
##    <chr>      <chr>             <int>           <dbl>        <int>       <dbl>
##  1 X.         integer               0               0          800      1
##  2 Name       character             0               0          800      1
##  3 Type.1     character             0               0           18      0.0225
##  4 Type.2     character             0               0           19      0.0238
##  5 HP         integer               0               0           94      0.118
##  6 Attack     integer               0               0          111      0.139
##  7 Defense    integer               0               0          103      0.129
##  8 Sp..Atk    integer               0               0          105      0.131
##  9 Sp..Def    integer               0               0           92      0.115
## 10 Speed      integer               0               0          108      0.135
## 11 Generation integer               0               0            6      0.0075
## 12 Legendary  character             0               0            2      0.0025
```

Frequência por tipo

Vamos visualizar a frequência com que cada tipo.1 aparece na nossa tabela.

```r
df %>% dplyr::select(Type.1) %>% summarytools::freq()
```

```
## Error in table(names(candidates))[["tested"]]: índice fora dos limites
```

```
## Warning in parse_call(mc = match.call(), caller = "freq"): metadata extraction
## terminated unexpectedly; inspect results carefully
```

```
## Frequencies
## df$Type.1
## Type: Character
##
##                  Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## -------------- ------ --------- -------------- --------- --------------
##             Bug     69      8.62           8.62      8.62           8.62
##            Dark     31      3.88          12.50      3.88          12.50
##          Dragon     32      4.00          16.50      4.00          16.50
##        Electric     44      5.50          22.00      5.50          22.00
##           Fairy     17      2.12          24.12      2.12          24.12
##        Fighting     27      3.38          27.50      3.38          27.50
##            Fire     52      6.50          34.00      6.50          34.00
##          Flying      4      0.50          34.50      0.50          34.50
##           Ghost     32      4.00          38.50      4.00          38.50
##           Grass     70      8.75          47.25      8.75          47.25
##          Ground     32      4.00          51.25      4.00          51.25
##             Ice     24      3.00          54.25      3.00          54.25
##          Normal     98     12.25          66.50     12.25          66.50
##          Poison     28      3.50          70.00      3.50          70.00
##         Psychic     57      7.12          77.12      7.12          77.12
##            Rock     44      5.50          82.62      5.50          82.62
```

```
##          Steel     27      3.38          86.00      3.38          86.00
##          Water    112     14.00         100.00     14.00         100.00
##           <NA>      0                               0.00         100.00
##          Total    800    100.00         100.00    100.00         100.00
```

Primeira comparação de poder de ataque

Vamos usar um um código para trazer de cada geração registrada o pokemon mais forte e mais fraco.

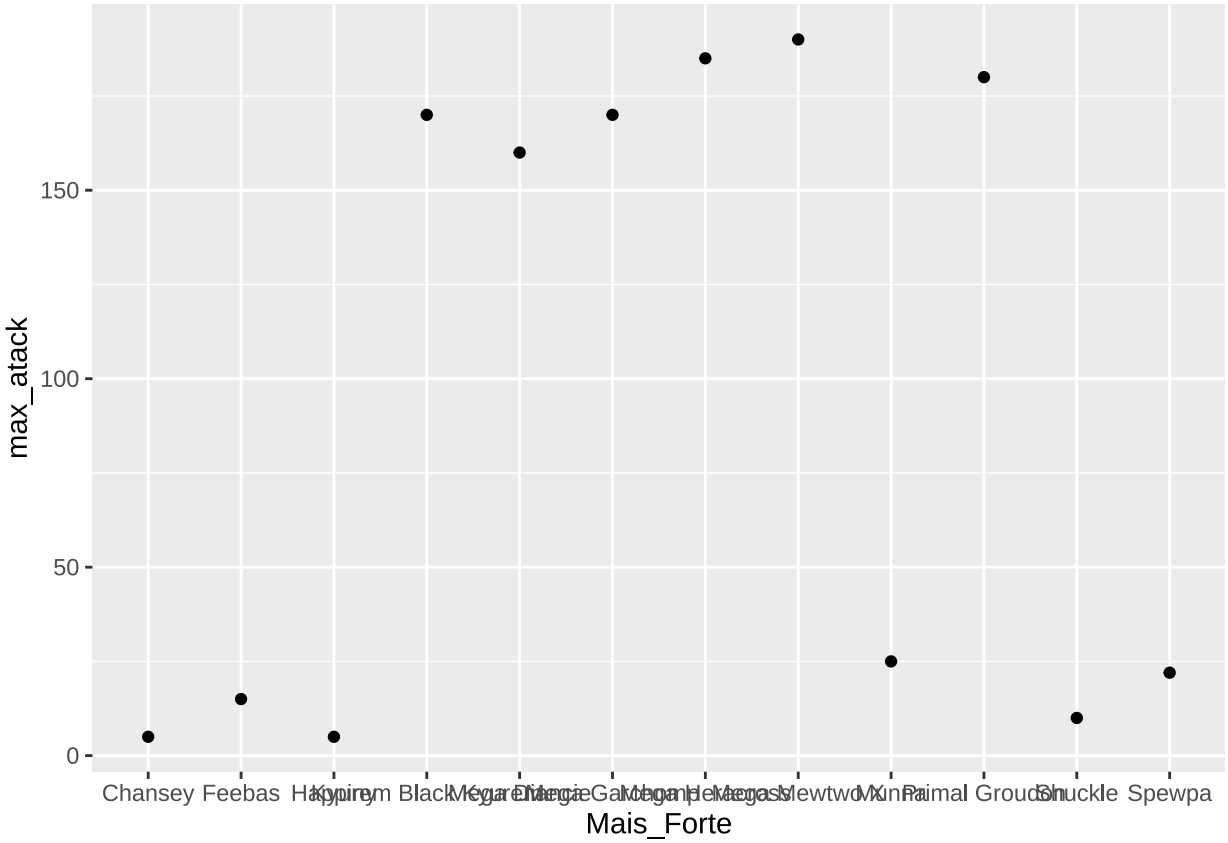
```
stronger <- df %>% group_by(Generation) %>% summarise(Mais_Forte = Name[which.max(Attack)],max_atack = m
                                                      Mais_Fraco = Name[which.min(Attack)], min_atack =

print(stronger)
```

```
## # A tibble: 6 x 5
##   Generation Mais_Forte          max_atack Mais_Fraco min_atack
##        <int> <chr>                   <int> <chr>          <int>
## 1          1 Mega Mewtwo X             190 Chansey            5
## 2          2 Mega Heracross            185 Shuckle           10
## 3          3 Primal Groudon            180 Feebas            15
## 4          4 Mega Garchomp             170 Happiny            5
## 5          5 Kyurem Black Kyurem       170 Munna             25
## 6          6 Mega Diancie              160 Spewpa            22
```

Gráfico de apresentação

Veremos o gráfico apresentando os dados dos pokemons mais fortes e fracos de cada geração.

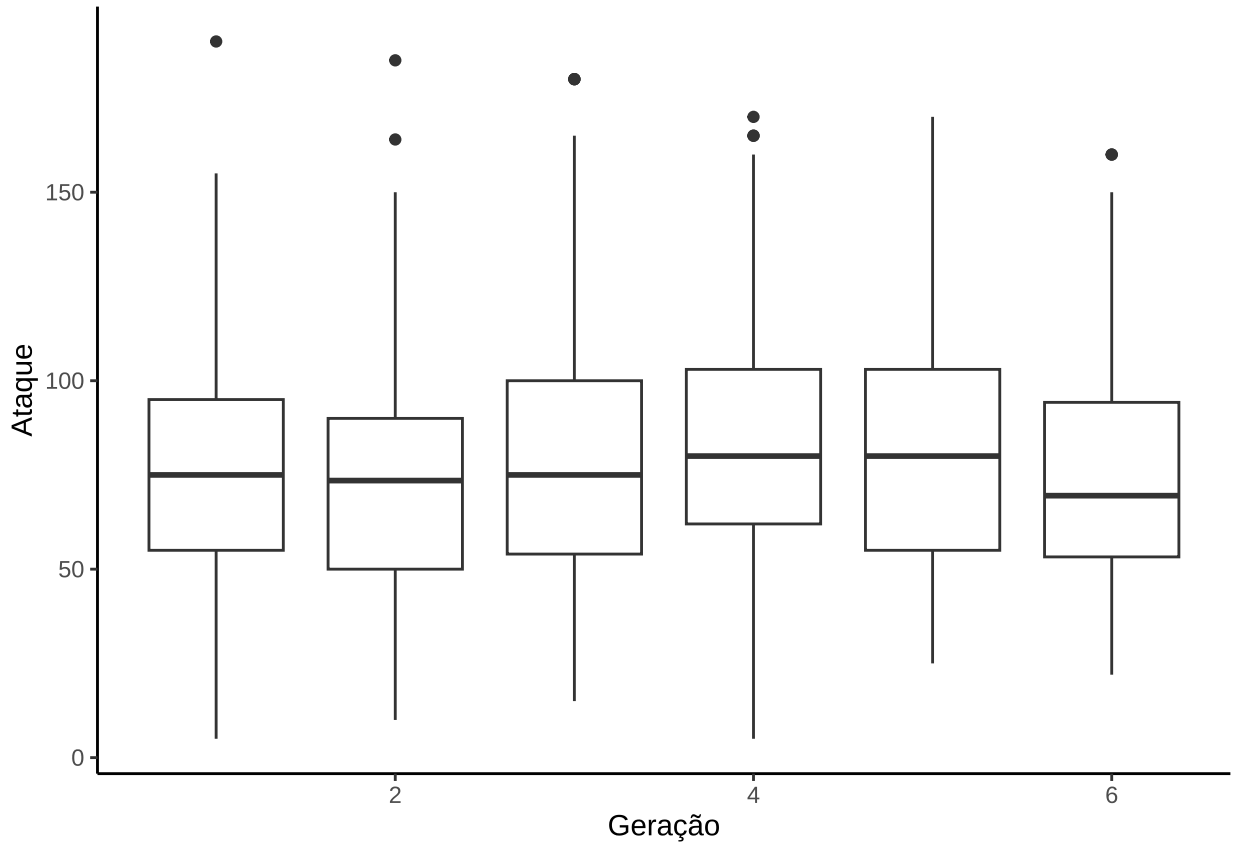
```
ggplot(data = stronger) +
  geom_point(mapping = aes(x = Mais_Forte,  y = max_atack)) +
  geom_point(mapping = aes(x = Mais_Fraco, y = min_atack))
```



Buscando outliers

Poderemos ver se há pokemons que fogem a média dos outros.

```
df %>% dplyr::select(Generation, Attack) %>%
  ggplot(aes(group = Generation, x=Generation, y = Attack)) + geom_boxplot() +
  xlab('Geração') +
  ylab('Ataque') +
  theme_classic()
```



Mais forte por tipo.1

Veremos a lista dos mais fortes pokemons por seu tipo.

```
Mais_forte_Tipo <- df %>% group_by(Type.1) %>% summarise(Mais_forte_tipo1 = Name[which.max(Attack)],
                                                         max_forte = max(Attack, na.rm = TRUE)) %>% arra
Mais_forte_Tipo
```

```
## # A tibble: 18 x 3
##    Type.1   Mais_forte_tipo1         max_forte
##    <chr>    <chr>                        <int>
##  1 Psychic  Mega Mewtwo X                  190
##  2 Bug      Mega Heracross                 185
##  3 Dragon   Mega Rayquaza                  180
##  4 Ground   Primal Groudon                 180
##  5 Ghost    Mega Banette                   165
##  6 Rock     Rampardos                      165
##  7 Fire     Mega Blaziken                  160
##  8 Normal   Slaking                        160
##  9 Water    Mega Gyarados                  155
## 10 Dark     Mega Absol                     150
## 11 Steel    Aegislash Blade Forme          150
## 12 Fighting Mega Lucario                   145
## 13 Grass    Mega Abomasnow                 132
```

```
## 14 Fairy    Xerneas                        131
## 15 Ice      Mamoswine                      130
## 16 Electric Electivire                     123
## 17 Flying   Tornadus Incarnate Forme       115
## 18 Poison   Toxicroak                      106
```

Media de ataque

A média de ataque dos pokemons calculada na geração.

```
media_por_geracao <- df %>% group_by(Generation) %>% summarise(media = mean(Attack, na.rm = TRUE)) %>%
  arrange(Generation)

media_por_geracao
```

```
## # A tibble: 6 x 2
##   Generation media
##        <int> <dbl>
## 1          1  76.6
## 2          2  72.0
## 3          3  81.6
## 4          4  82.9
## 5          5  82.1
## 6          6  75.8
```

Estatísticas de ataque

Estatistica de ataque dos pokemons calculada por geração.

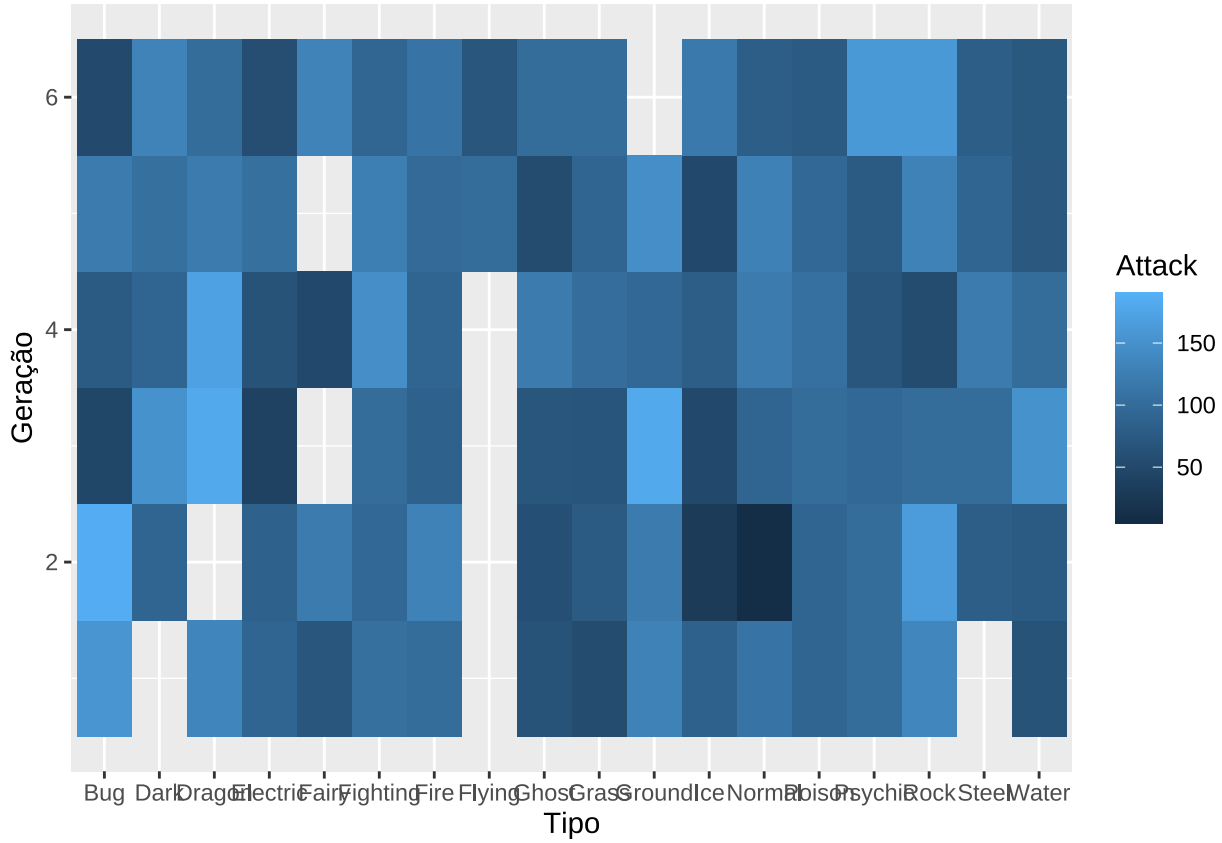
```
estatisticas_attack <- df %>% group_by(Generation) %>%
  summarise(
    media = mean(Attack, na.rm = TRUE),
    desvio = sd(Attack, na.rm = TRUE),
    prim_quartil = quantile(Attack, 0.25, na.rm = TRUE),
    terc_quartil = quantile(Attack, 0.75, na.rm = TRUE)
  )

print(estatisticas_attack)
```

```
## # A tibble: 6 x 5
##   Generation media desvio prim_quartil terc_quartil
##        <int> <dbl>  <dbl>        <dbl>        <dbl>
## 1          1  76.6   30.7         55           95
## 2          2  72.0   32.7         50           90
## 3          3  81.6   36.6         54          100
## 4          4  82.9   32.8         62          103
## 5          5  82.1   30.4         55          103
## 6          6  75.8   29.2         53.2         94.2
```

Comparação de tipo e geração com base no ataque

Gráfico trazendo 3 parametros para demonstração: Geração, Tipo e Ataque

```
df %>% ggplot(aes(x= Type.1, y= Generation, fill = Attack)) +
  geom_tile() +
  xlab("Tipo") +
  ylab("Geração")
```



Função DESC

```
df %>%  group_by(Generation) %>% dplyr::select(Attack) %>% summarytools::descr()
```

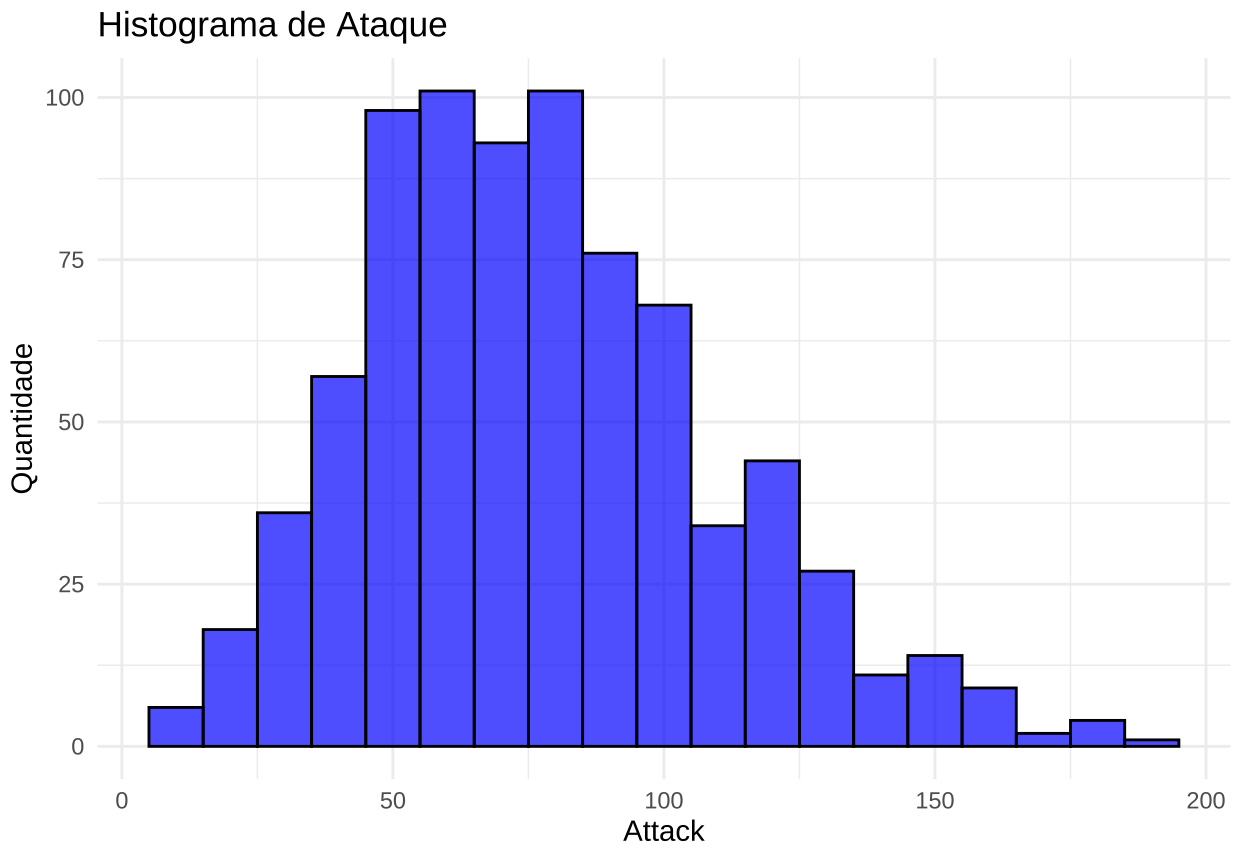
```
## Adding missing grouping variables: `Generation`

## Descriptive Statistics
##  by Generation
## Data Frame: df
## N: 800
##
##                         1        2        3        4        5        6
## ----------------- -------- -------- -------- -------- -------- --------
##              Mean    76.64    72.03    81.62    82.87    82.07    75.80
```

```
##           Std.Dev    30.74    32.71    36.59    32.78    30.37    29.18
##               Min     5.00    10.00    15.00     5.00    25.00    22.00
##                Q1    55.00    50.00    53.00    62.00    55.00    53.00
##            Median    75.00    73.50    75.00    80.00    80.00    69.50
##                Q3    95.00    90.00   100.00   103.00   103.00    95.00
##               Max   190.00   185.00   180.00   170.00   170.00   160.00
##               MAD    29.65    31.88    37.06    29.65    37.06    28.91
##               IQR    40.00    40.00    46.00    41.00    48.00    41.00
##                CV     0.40     0.45     0.45     0.40     0.37     0.38
##          Skewness     0.58     0.63     0.63     0.31     0.34     0.83
##       SE.Skewness     0.19     0.23     0.19     0.22     0.19     0.27
##          Kurtosis     0.55     0.72    -0.14    -0.06    -0.59     0.40
##           N.Valid   166.00   106.00   160.00   121.00   165.00    82.00
##                 N   166.00   106.00   160.00   121.00   165.00    82.00
##         Pct.Valid   100.00   100.00   100.00   100.00   100.00   100.00
```

Histograma de poder de ataque por quantidade

Separação por quantidade de pokemons nivelado por seu poder de ataque com bins = 10

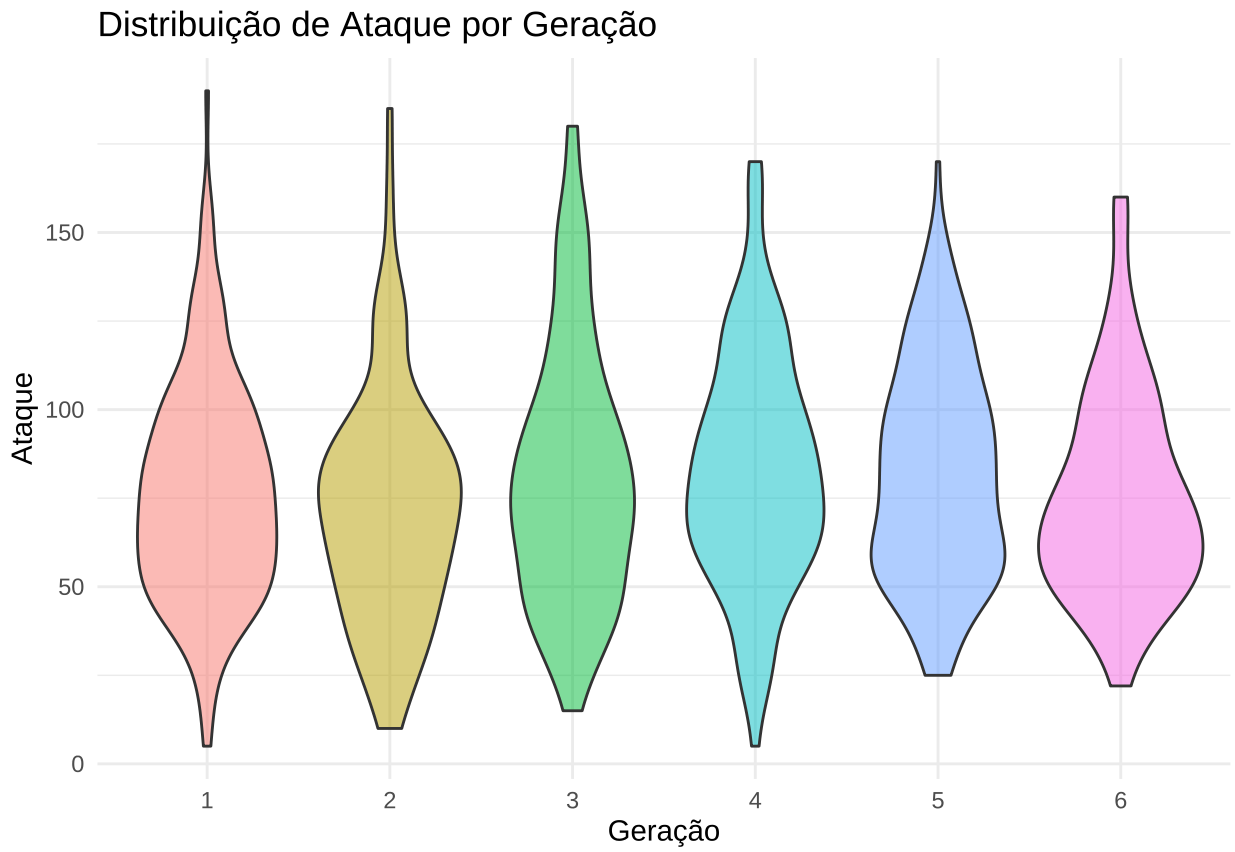
```
ggplot(df, aes(x = Attack)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histograma de Ataque", x = "Attack", y = "Quantidade") +
  theme_minimal()
```



Distribuição de ataque separado por geração

Aqui podemos analisar separadamente cada geração com o gráfico violino o lado para o qual a calda pende, mostrando uma curva assimétrica com calda prolongada para direita.

```
ggplot(df, aes(x = as.factor(Generation), y = Attack, fill = as.factor(Generation))) +
  geom_violin(alpha = 0.5) +
  labs(title = "Distribuição de Ataque por Geração", x = "Geração", y = "Ataque") +
  theme_minimal() +
  theme(legend.position = "none")
```



Scatterplot entre ataque e defesa

Veremos um scatterplot entre ataque e defesa separados por tipo.1 de pokemon vendo se há correlação ao fato de um pokemon com alto ataque também tem um alto índice de defesa ou não.

```
ggplot(df, aes(x = Attack, y = Defense, color = as.factor(Type.1))) +
  geom_point(alpha = 0.6) +
  labs(title = "Scatterplot de Attack vs. Defense",
       x = "Attack", y = "Defense", color = "Tipo de Pokémon") +
  theme_minimal()
```

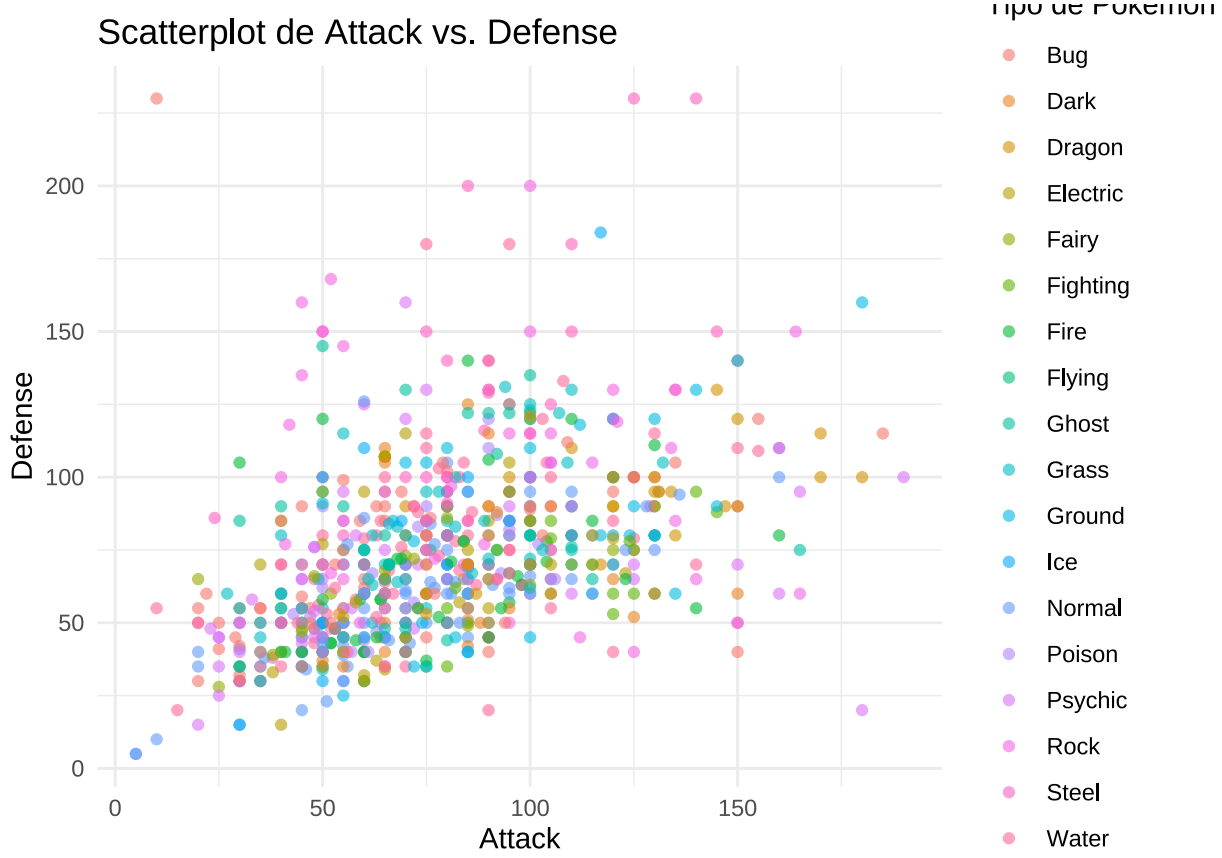
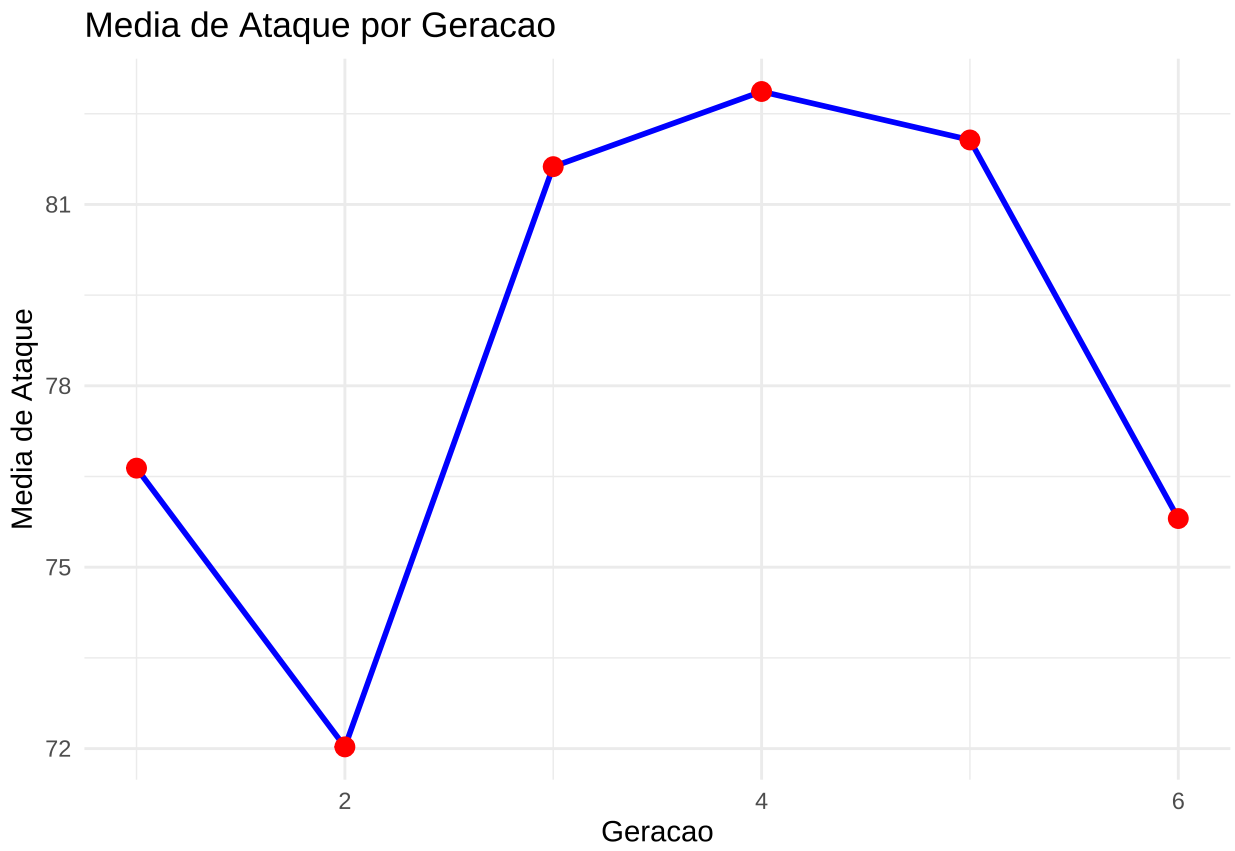


Gráfico de linha

Veremos um gráfico de linha calculando a médita de ataque por geração.

```
df_summary <- df %>%
  group_by(Generation) %>%
  summarise(Media_Attack = mean(Attack, na.rm = TRUE))

ggplot(df_summary, aes(x = Generation, y = Media_Attack)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "red", size = 3) +
  labs(title = "Media de Ataque por Geracao", x = "Geracao", y = "Media de Ataque") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Instalação RStudio

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Figure 1: Programa instalado na máquina

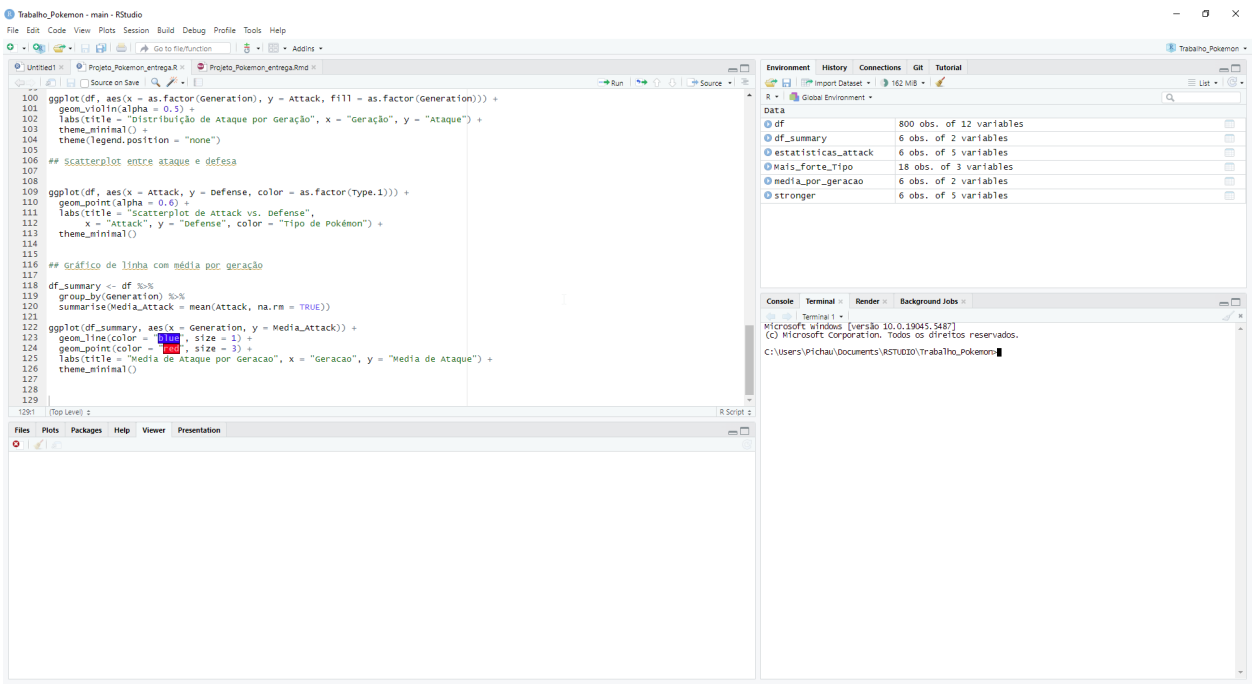


Figure 2: dataset no ambiente do RStudio