

# Princípios de Visualização de Dados com o R

Curso de Verão IME-USP 2022

Bruna Garbes

03/02/2022

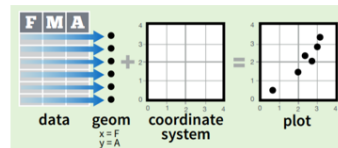


# Programação

- Breve revisão da primeira aula
- Gráfico de histograma.
- Gráfico de barras.
- Gráfico de dispersão.
- Gráfico de bolhas.
- Gráfico de box-plot.
- Gráfico de linhas.
- Gráfico de cascata (waterfall graph).
- Gráfico de mapa de árvore (tree map graph).
- Outros tipos de gráficos.
- Como pedir ajuda na hora de construir um gráfico.
- Combinando mais de um tipo de gráfico.
- Salvando e exportando os gráficos gerados no Ggplot2.
- Exercícios práticos.

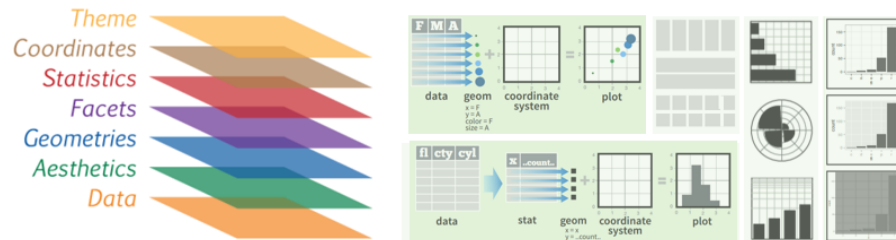
# Revisão da Sintaxe Básica do ggplot2

```
ggplot(data, aesthetics) +  
  geometries()
```



# Sintaxe Básica do ggplot2

```
ggplot(data, aesthetics) +  
  geometries(statistics) +  
  facets +  
  coordinates +  
  theme
```



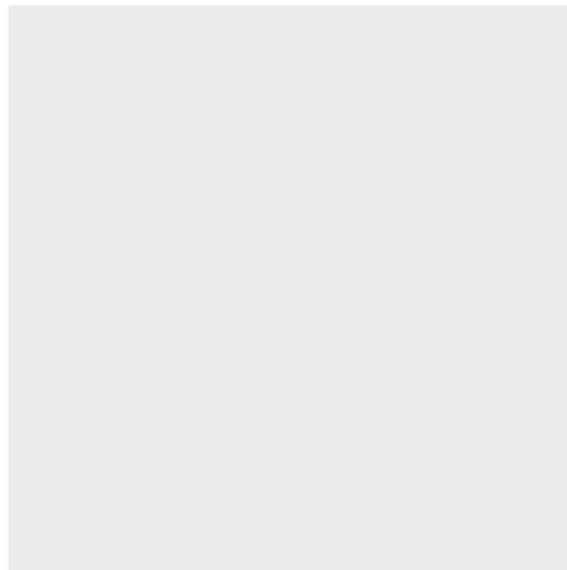
# Sintaxe Básica do ggplot2

```
#Conjunto de bibliotecas para ciência de dados
install.packages("tidyverse")
library(tidyverse)
```

```
dados <- tibble(var_x = 1:4, var_y = seq(2,8,2), var_grupo = c(rep("a", 3), "b"))
summary(dados)
#>   var_x      var_y  var_grupo
#> Min.   :1.00  Min.   :2.0  Length:4
#> 1st Qu.:1.75  1st Qu.:3.5  Class :character
#> Median :2.50  Median :5.0  Mode  :character
#> Mean   :2.50  Mean   :5.0
#> 3rd Qu.:3.25  3rd Qu.:6.5
#> Max.   :4.00  Max.   :8.0
```

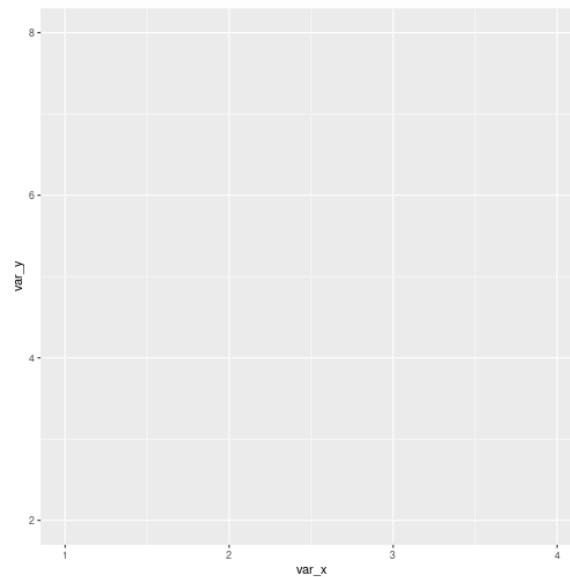
## Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados)
```



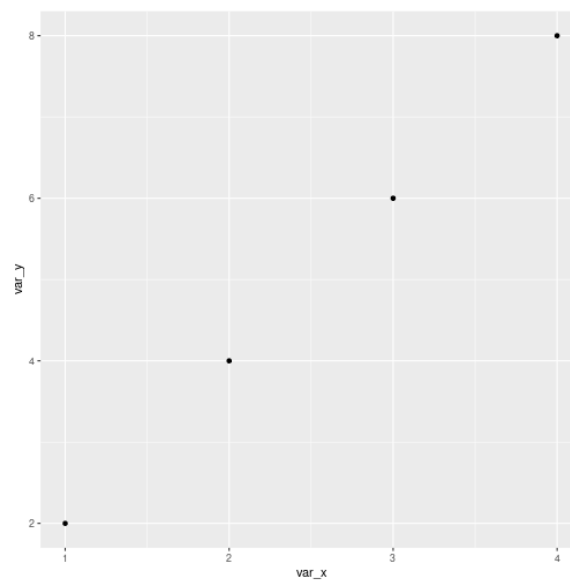
## Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y))
```



## Sintaxe Básica do ggplot2

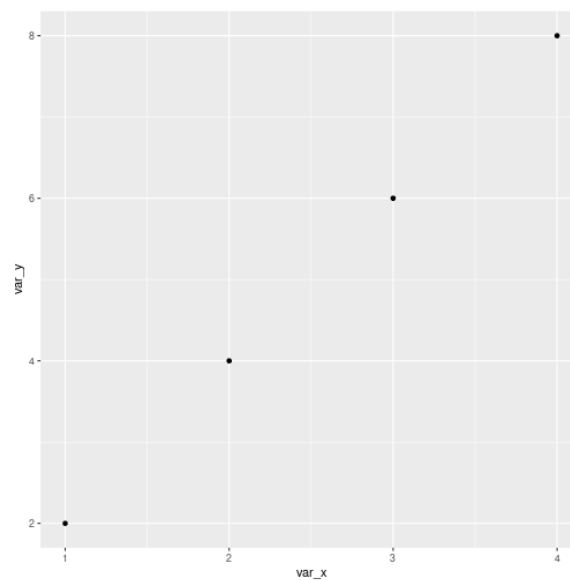
```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y)) +  
  geom_point()
```





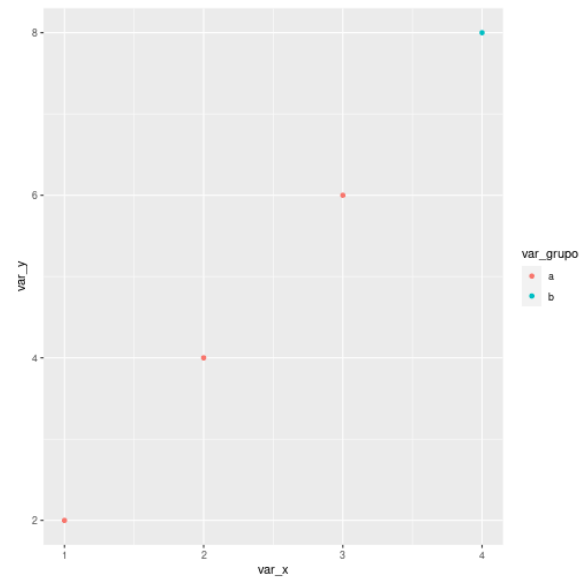
## Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados) +  
  geom_point(aes(x = var_x, y = var_y))
```



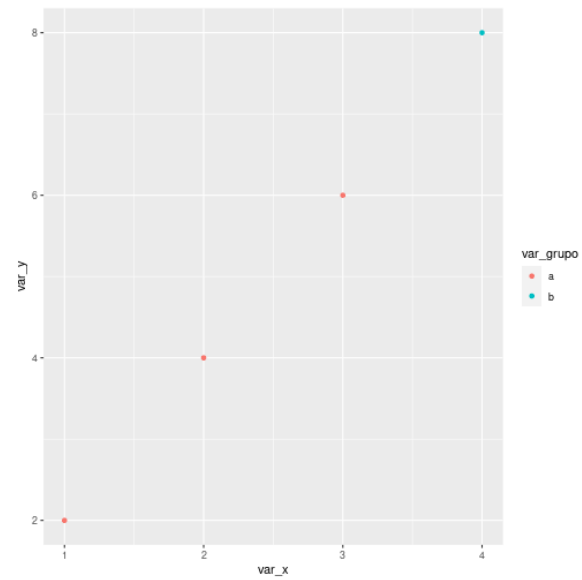
## Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados) +  
  geom_point(aes(x = var_x, y = var_y,  
                 color = var_grupo))
```



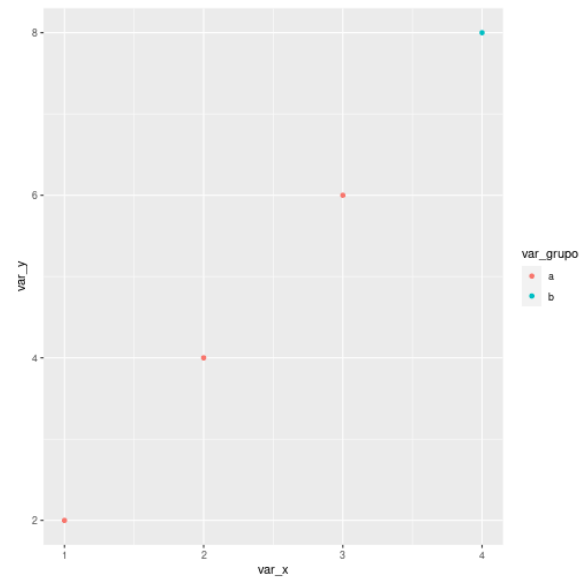
# Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y)) +  
  geom_point(aes(color = var_grupo))
```



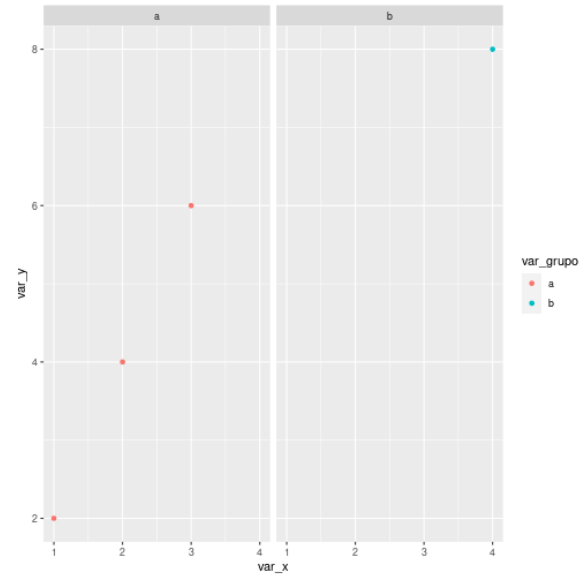
## Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y,  
                  color = var_grupo)) +  
  geom_point()
```



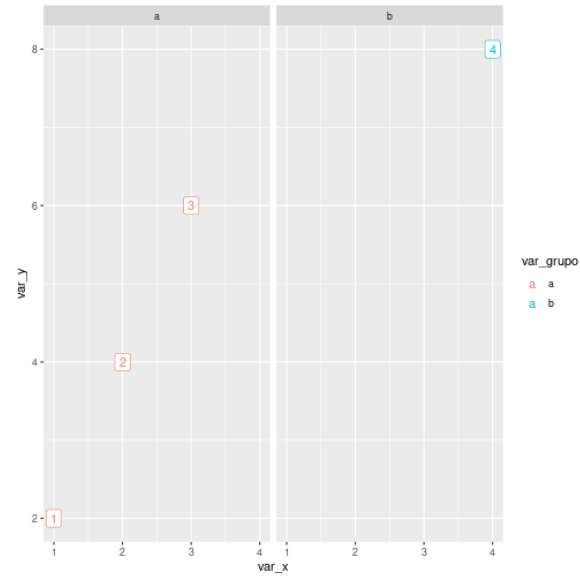
# Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y,  
                  color = var_grupo)) +  
  geom_point() +  
  facet_grid(.~var_grupo)
```



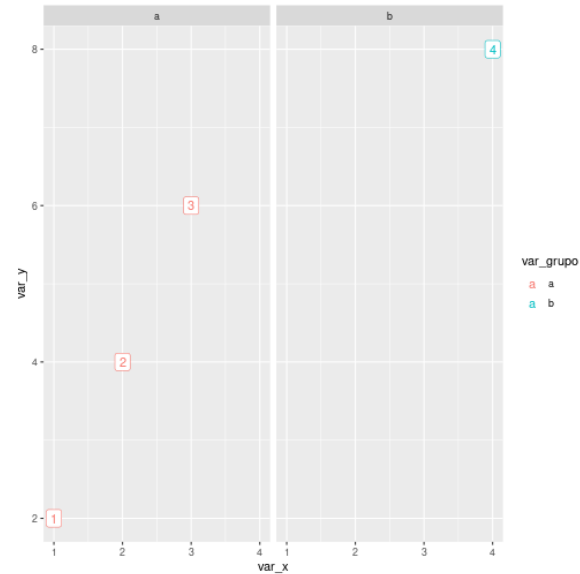
# Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y,  
                 color = var_grupo)) +  
  geom_label(aes(label=var_x)) +  
  facet_grid(.~var_grupo)
```



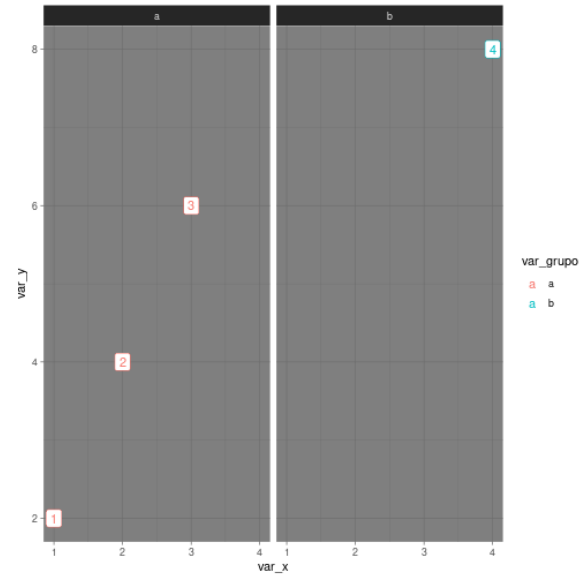
# Sintaxe Básica do ggplot2

```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y,  
                  color = var_grupo,  
                  label = var_x)) +  
  geom_point() +  
  geom_label() +  
  facet_grid(.~var_grupo)
```



# Sintaxe Básica do ggplot2

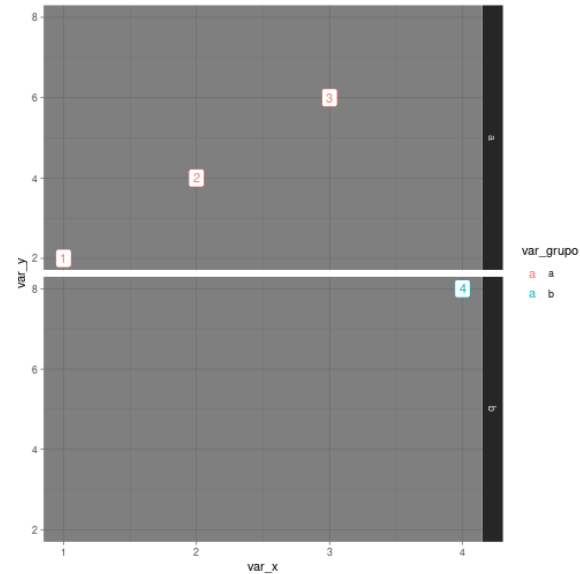
```
#exemplo  
ggplot(dados, aes(x = var_x, y = var_y,  
                 color = var_grupo,  
                 label = var_x)) +  
  geom_point() +  
  geom_label() +  
  facet_grid(~var_grupo) +  
  theme_dark()
```





# Sintaxe Básica do ggplot2

```
#exemplo  
p <- ggplot(dados, aes(x = var_x, y = var_y,  
                        color = var_grupo,  
                        label = var_x)) +  
  geom_point() +  
  geom_label() +  
  facet_grid(~var_grupo) +  
  theme_dark()  
  
p + facet_grid(var_grupo~.)
```



# Aplicações



Ilustração por Allison Horst - Twitter: [@allison\\_horst](https://twitter.com/allison_horst)

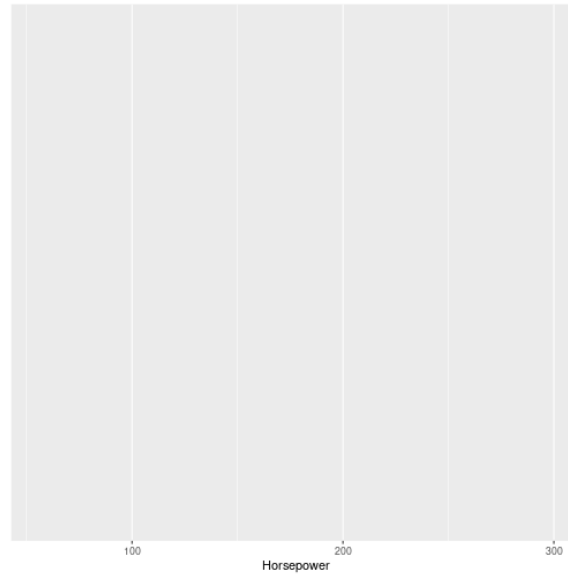
# Base de Dados Cars93

```
#Conjunto de bibliotecas para ciência de dados  
install.packages("tidyverse")  
library(tidyverse)  
  
#Pacote que contém base de dados Cars93  
install.packages("MASS")  
library(MASS)
```

```
df <- Cars93 %>%  
  select(Horsepower, Type, AirBags)  
  
glimpse(df)  
#> Rows: 93  
#> Columns: 3  
#> $ Horsepower <int> 140, 200, 172, 172, 208, 110, 170, 180, 170, 200, 295, 110, ...  
#> $ Type <fct> Small, Midsize, Compact, Midsize, Midsize, Midsize, Large, ...  
#> $ AirBags <fct> None, Driver & Passenger, Driver only, Driver & Passenger, ...
```

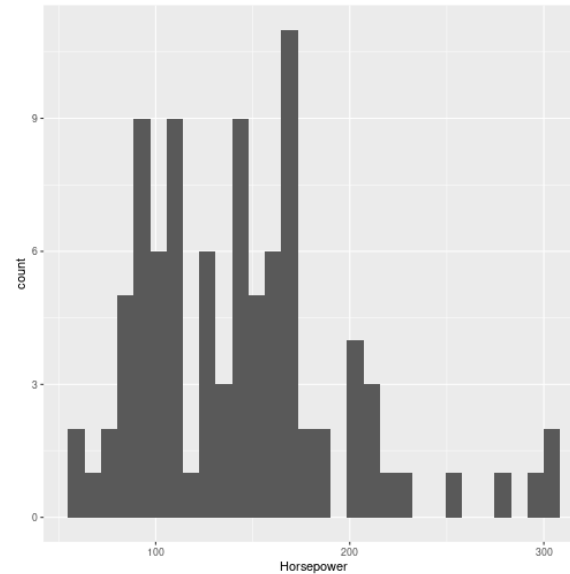
# Base de Datos Cars93

```
ggplot(df, aes(x = Horsepower))
```



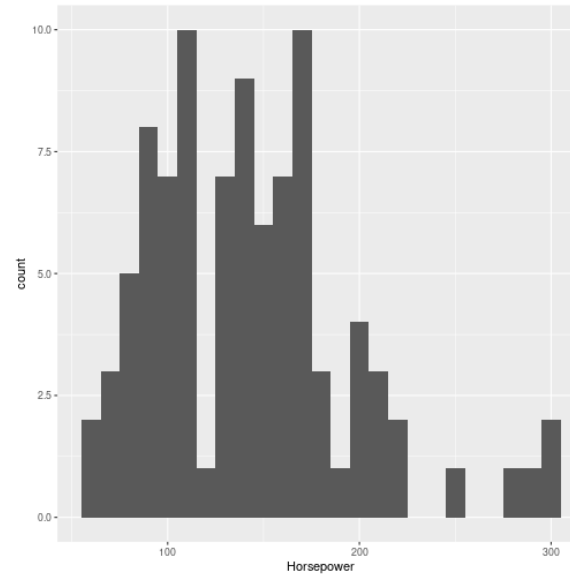
## Base de Datos Cars93 + Histograma

```
ggplot(df, aes(x = Horsepower)) +  
  geom_histogram()
```



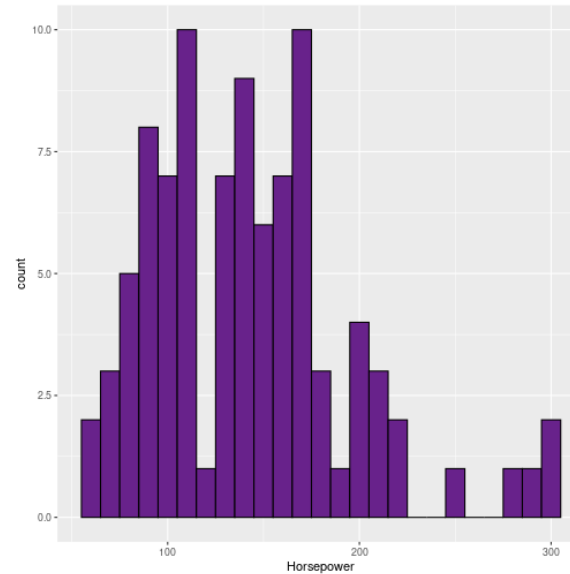
## Base de Datos Cars93 + Histograma

```
ggplot(df, aes(x = Horsepower)) +  
  geom_histogram(binwidth = 10)
```



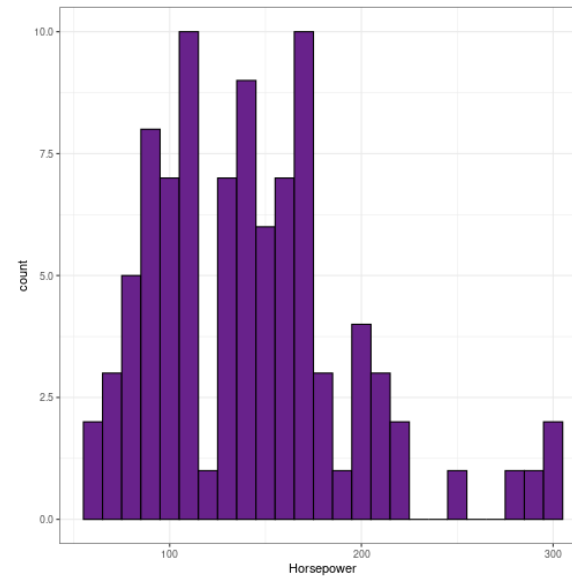
## Base de Datos Cars93 + Histograma

```
ggplot(df, aes(x = Horsepower)) +  
  geom_histogram(binwidth = 10,  
                color = "black",  
                fill = "darkorchid4")
```



## Base de Dados Cars93 + Histograma

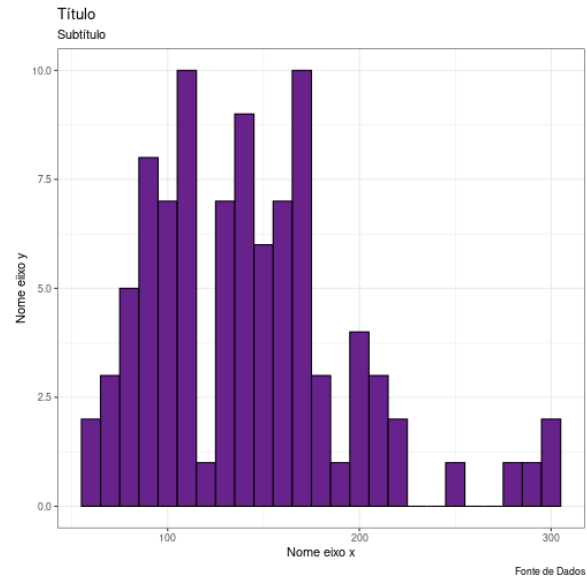
```
ggplot(df, aes(x = Horsepower)) +  
  geom_histogram(binwidth = 10,  
                color = "black",  
                fill = "darkorchid4") +  
  theme_bw()
```





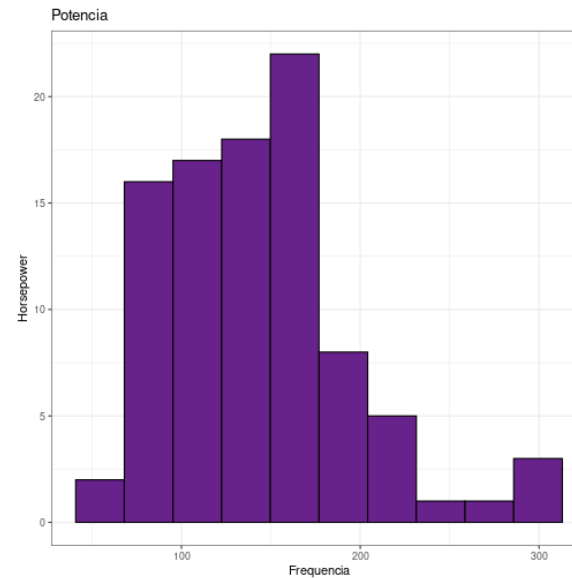
# Base de Dados Cars93 + Histograma

```
ggplot(df, aes(x = Horsepower)) +  
  geom_histogram(binwidth = 10,  
                color = "black",  
                fill = "darkorchid4") +  
  theme_bw() +  
  labs(x = "Nome eixo x",  
       y = "Nome eixo y",  
       title = "Título",  
       subtitle = "Subtítulo",  
       caption="Fonte de Dados")
```



## Base de Datos Cars93 + Histograma

```
ggplot(df, aes(x = Horsepower)) +  
  geom_histogram(bins = 10,  
                color = "black",  
                fill = "darkorchid4") +  
  theme_bw() +  
  labs(x = "Frecuencia",  
       y = "Horsepower",  
       title = "Potencia")
```



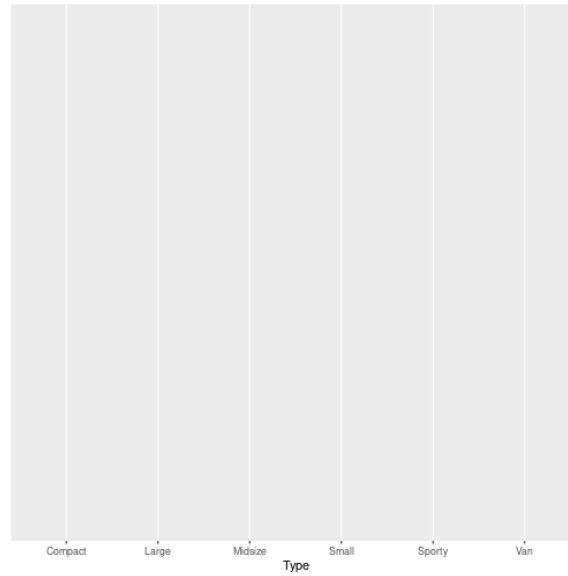
# Base de Dados Cars93 + Histograma

## Exercício

Criar um histograma a partir de alguma variável da base Cars93, com no máximo 5 barras, com título, subtítulo e nome dos eixos x e y.

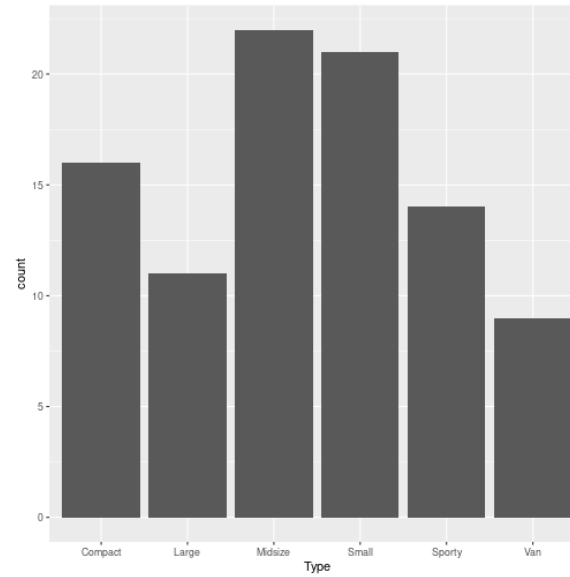
# Base de Datos Cars93

```
ggplot(df, aes(x = Type))
```



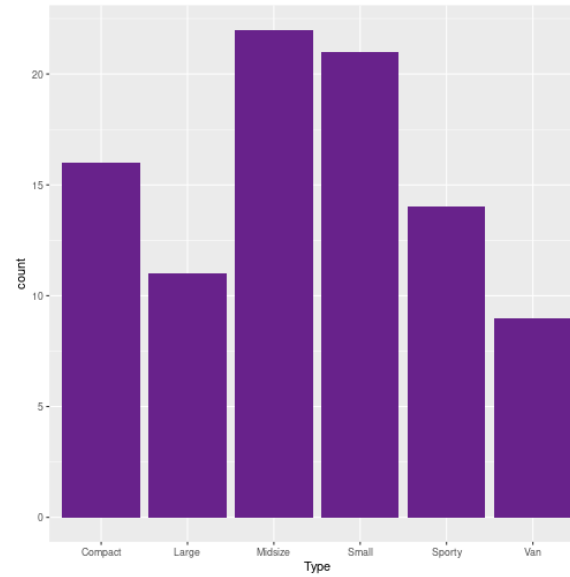
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type)) +  
  geom_bar()
```



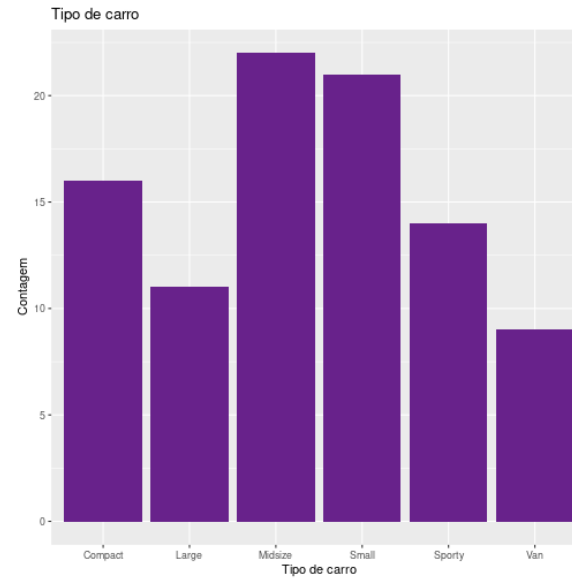
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type)) +  
  geom_bar(fill = "darkorchid4")
```



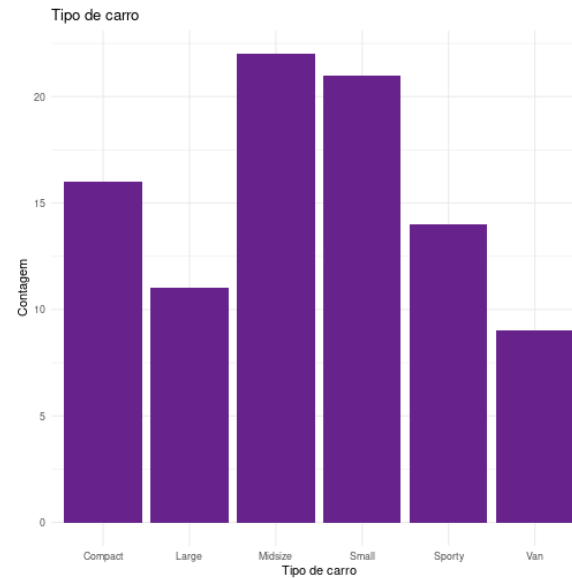
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type)) +  
  geom_bar(fill = "darkorchid4") +  
  labs(x = "Tipo de carro", y = "Contagem",  
       title = "Tipo de carro")
```



## Base de Dados Cars93 + Gráfico de Barras

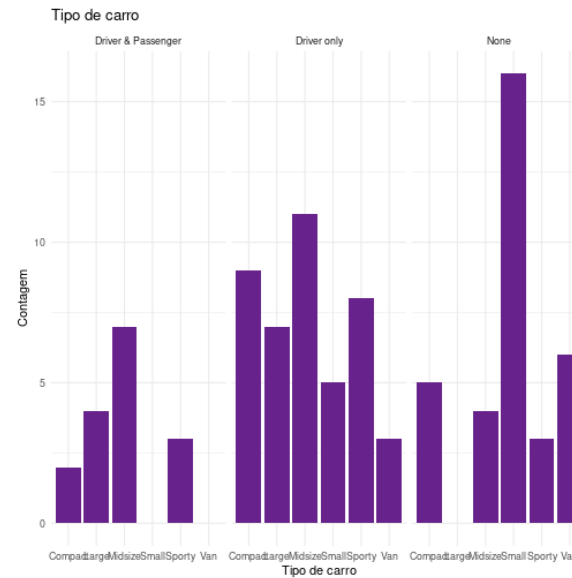
```
ggplot(df, aes(x = Type)) +  
  geom_bar(fill = "darkorchid4") +  
  labs(x = "Tipo de carro", y = "Contagem",  
       title = "Tipo de carro") +  
  theme_minimal()
```





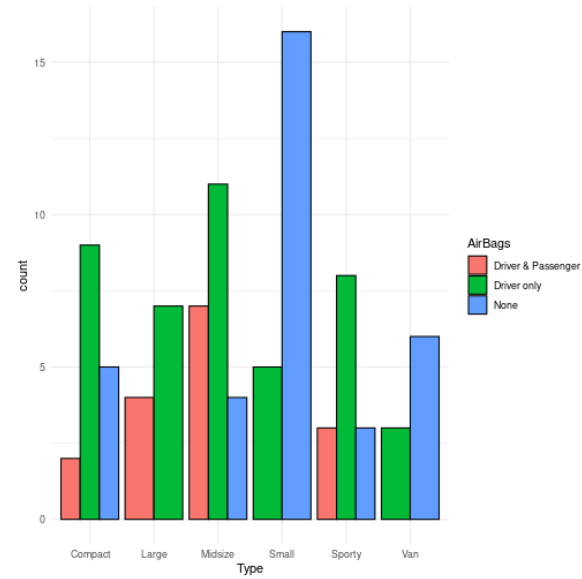
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type)) +  
  geom_bar(fill = "darkorchid4") +  
  labs(x = "Tipo de carro", y = "Contagem",  
       title = "Tipo de carro") +  
  theme_minimal() +  
  facet_grid(~ AirBags)
```



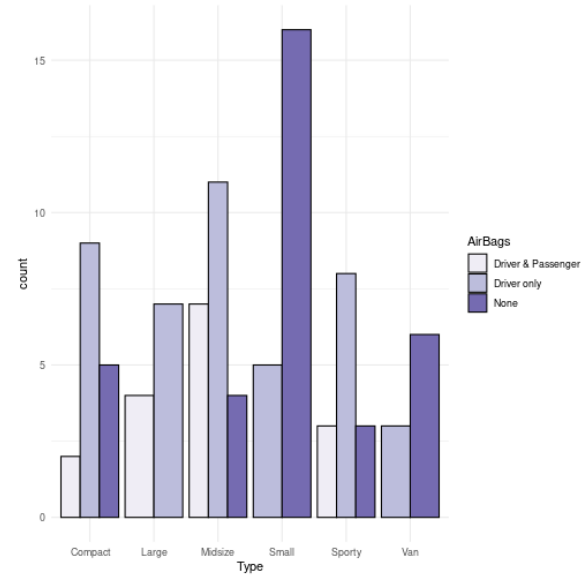
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type, fill = AirBags)) +  
  geom_bar(position = "dodge",  
           color = "black") +  
  theme_minimal()
```



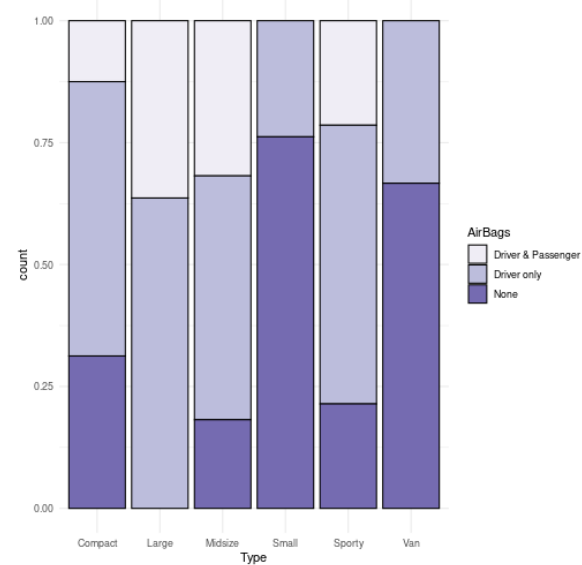
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type, fill = AirBags)) +  
  geom_bar(position = "dodge",  
           color = "black") +  
  theme_minimal() +  
  scale_fill_brewer(palette = "Purples",  
                   direction = 1)
```



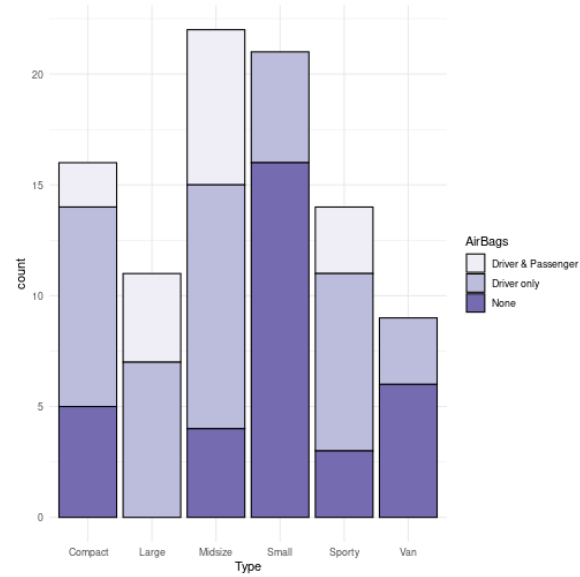
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type, fill = AirBags)) +  
  geom_bar(position = "fill",  
           color = "black") +  
  theme_minimal() +  
  scale_fill_brewer(palette = "Purples",  
                   direction = 1)
```



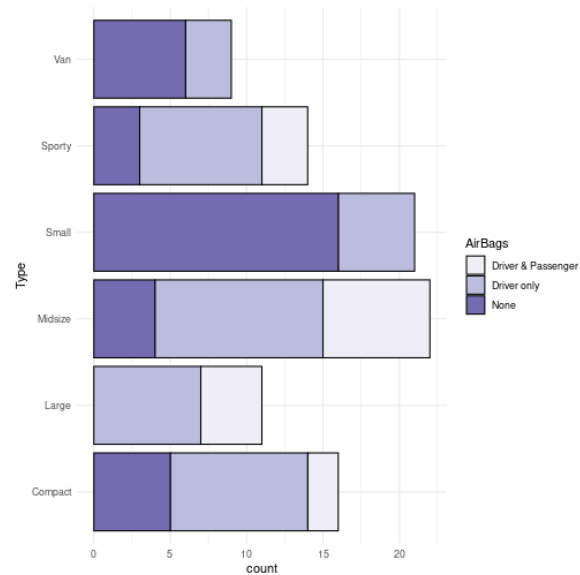
## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type, fill = AirBags)) +  
  geom_bar(position = "stack",  
           color = "black") +  
  theme_minimal() +  
  scale_fill_brewer(palette = "Purples",  
                   direction = 1)
```



## Base de Datos Cars93 + Gráfico de Barras

```
ggplot(df, aes(x = Type, fill = AirBags)) +  
  geom_bar(position = "stack",  
           color = "black") +  
  theme_minimal() +  
  scale_fill_brewer(palette = "Purples",  
                   direction = 1) +  
  coord_flip()
```



# Base de Dados Cars93 + Gráfico de Barras

## Exercício

Criar um gráfico de barras, usando umas das opções 'position' (fill, dodge ou stack), alterando o 'theme' e inserindo título, subtítulo e nomes dos eixos x e y.

# Base de Dados HairEyeColor

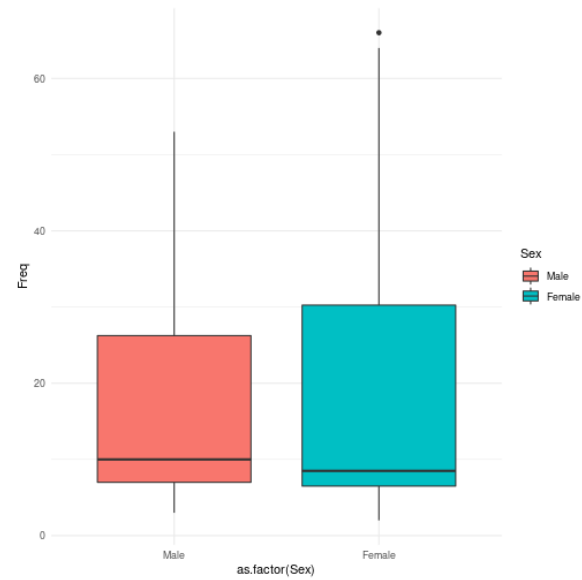
```
#Conjunto de bibliotecas para ciência de dados  
install.packages("tidyverse")  
library(tidyverse)
```

```
dfhair <- data.frame(HairEyeColor)  
glimpse(dfhair)  
#> Rows: 32  
#> Columns: 4  
#> $ Hair <fct> Black, Brown, Red, Blond, Black, Brown, Red, Blond, Black, Brown,...  
#> $ Eye <fct> Brown, Brown, Brown, Brown, Blue, Blue, Blue, Blue, Hazel, Hazel,...  
#> $ Sex <fct> Male, Male, Male, Male, Male, Male, Male, Male, Male, Male, Male,...  
#> $ Freq <dbl> 32, 53, 10, 3, 11, 50, 10, 30, 10, 25, 7, 5, 3, 15, 7, 8, 36, 66,...
```



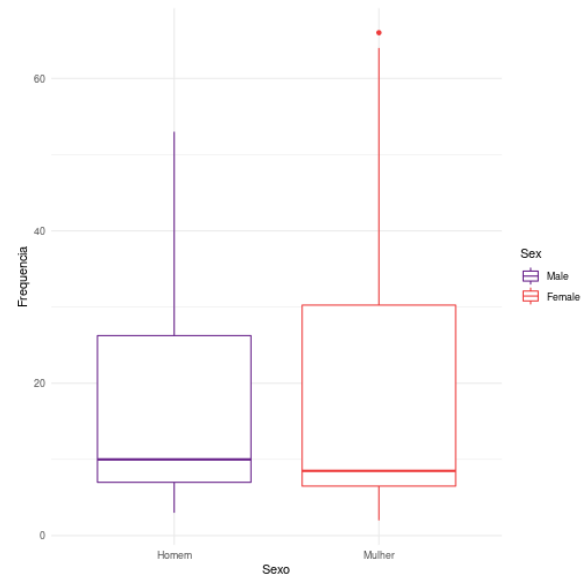
## Base de Dados HairEyeColor + Boxplot

```
ggplot(dfhair, aes(x = as.factor(Sex), y = Freq,  
                  fill = Sex)) +  
  theme_minimal() +  
  geom_boxplot()
```



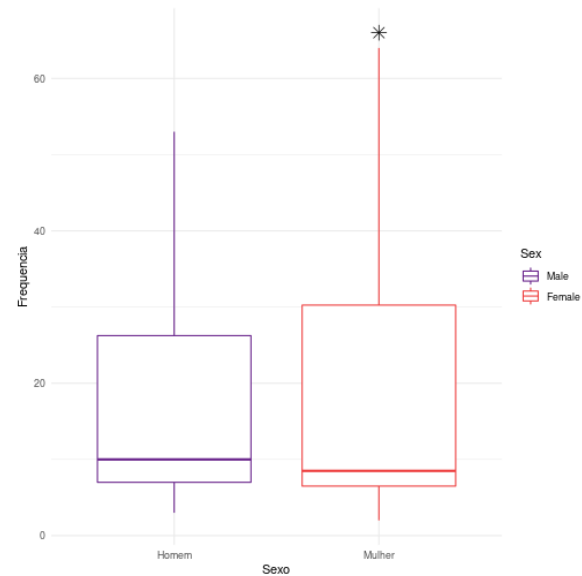
## Base de Dados HairEyeColor + Boxplot

```
ggplot(dfhair, aes(x = as.factor(Sex), y = Freq,  
                  color = Sex)) +  
  theme_minimal() +  
  geom_boxplot() +  
  scale_x_discrete(labels = c("Homem", "Mulher")) +  
  xlab("Sexo") +  
  ylab("Frequencia") +  
  scale_color_manual(values =  
                    c("darkorchid4", "brown2"))
```



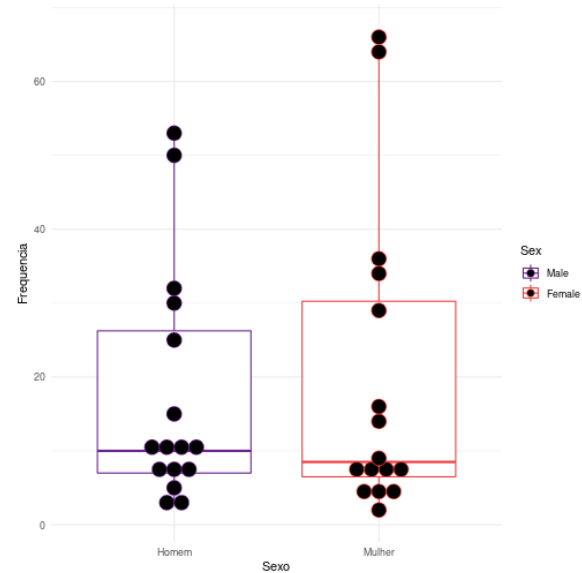
## Base de Dados HairEyeColor + Boxplot

```
ggplot(dfhair, aes(x = as.factor(Sex), y = Freq,  
                  color = Sex)) +  
  theme_minimal() +  
  geom_boxplot(  
    outlier.colour = "black",  
    outlier.shape = 8,  
    outlier.size = 4  
  ) +  
  scale_x_discrete(labels = c("Homem", "Mulher")) +  
  xlab("Sexo") +  
  ylab("Frequencia") +  
  scale_color_manual(values =  
    c("darkorchid4", "brown2"))
```



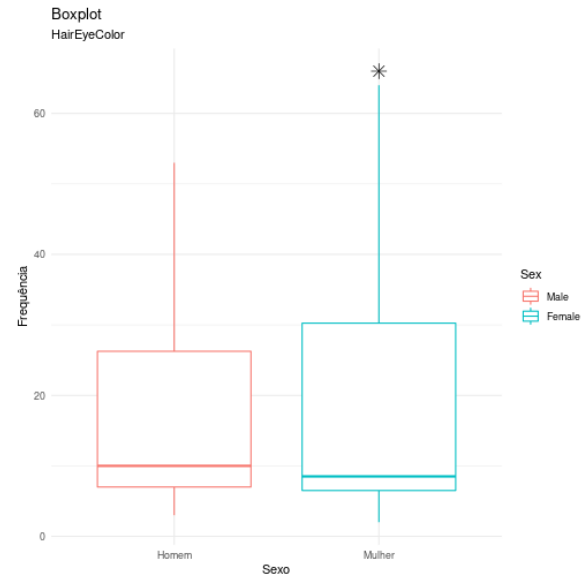
## Base de Dados HairEyeColor + Boxplot

```
ggplot(dfhair, aes(x = as.factor(Sex), y = Freq,  
                  color = Sex)) +  
  theme_minimal() +  
  geom_boxplot() +  
  geom_dotplot(  
    binaxis = 'y',  
    stackdir = 'center',  
    dotsize = 1,  
    binwidth = 2  
  ) +  
  scale_x_discrete(labels = c("Homem", "Mulher")) +  
  xlab("Sexo") +  
  ylab("Frequencia") +  
  scale_color_manual(values =  
    c("darkorchid4", "brown2"))
```



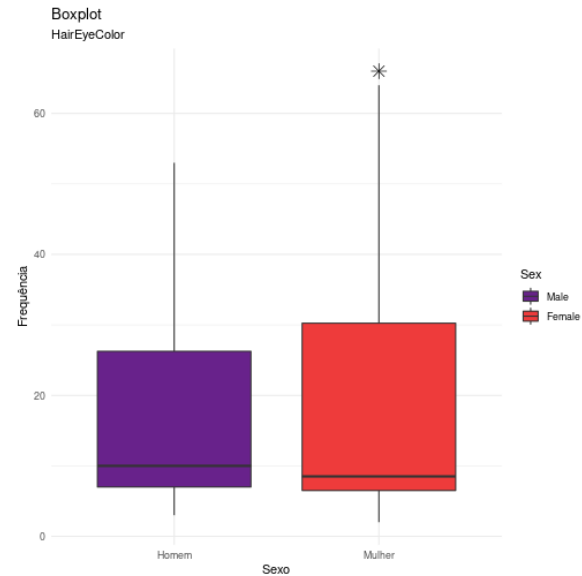
# Base de Dados HairEyeColor + Boxplot

```
ggplot(dfhair, aes(x = as.factor(Sex), y = Freq,  
                  color = Sex)) +  
  theme_minimal() +  
  geom_boxplot(  
    outlier.colour = "black",  
    outlier.shape = 8,  
    outlier.size = 4  
  ) +  
  scale_x_discrete(labels = c("Homem", "Mulher")) +  
  labs(  
    title = "Boxplot",  
    x = "Sexo",  
    y = "Frequência",  
    subtitle = "HairEyeColor"  
  )
```



# Base de Dados HairEyeColor + Boxplot

```
ggplot(dfhair, aes(x = as.factor(Sex), y = Freq,  
                  fill = Sex)) +  
  theme_minimal() +  
  geom_boxplot(  
    outlier.colour = "black",  
    outlier.shape = 8,  
    outlier.size = 4  
  ) +  
  scale_x_discrete(labels = c("Homem", "Mulher")) +  
  labs(  
    title = "Boxplot",  
    x = "Sexo",  
    y = "Frequência",  
    subtitle = "HairEyeColor"  
  ) +  
  scale_fill_manual(values =  
    c("darkorchid4", "brown2"))
```



# Base de Dados HairEyeColor + Boxplot

## Exercício

Criar um boxplot a partir da base de dados 'hairEyeColor', alterando a cor, evidenciando outliers quando houver e inserindo título, subtítulo e nomes dos eixos x e y.

```
#Conjunto de bibliotecas para ciência de dados
install.packages("tidyverse"); library(tidyverse)
```

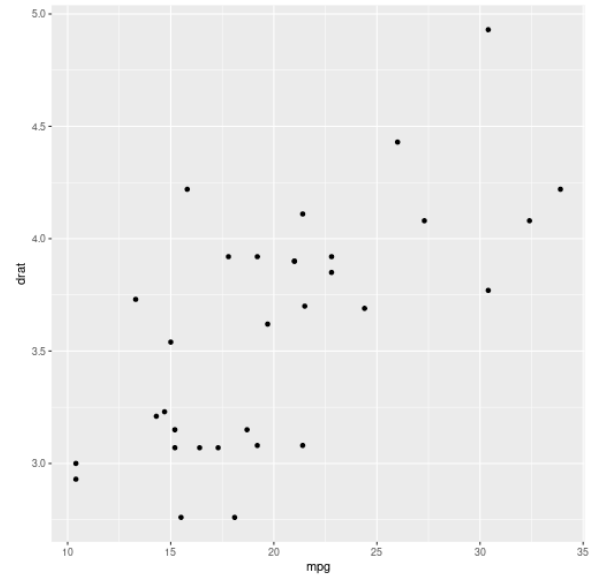
## Base de Dados mtcars

```
data(mtcars)
glimpse(mtcars)
#> Rows: 32
#> Columns: 11
#> $ mpg <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 19.2, 17.8,...
#> $ cyl <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 8,...
#> $ disp <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.7, 140.8, 16...
#> $ hp <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180, 180, 180...
#> $ drat <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3.92, 3.92,...
#> $ wt <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.190, 3.150, 3...
#> $ qsec <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.00, 22.90, 18...
#> $ vs <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,...
#> $ am <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0,...
#> $ gear <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3, 3,...
#> $ carb <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1, 2, 1, 1, 2,...
```



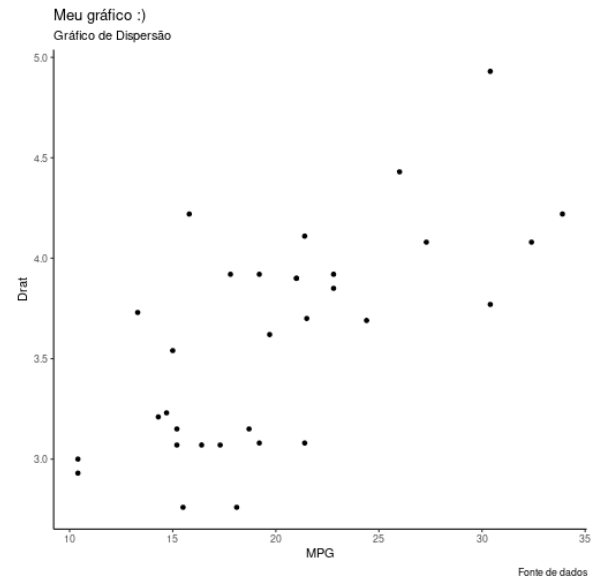
## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat)) +  
  geom_point()
```



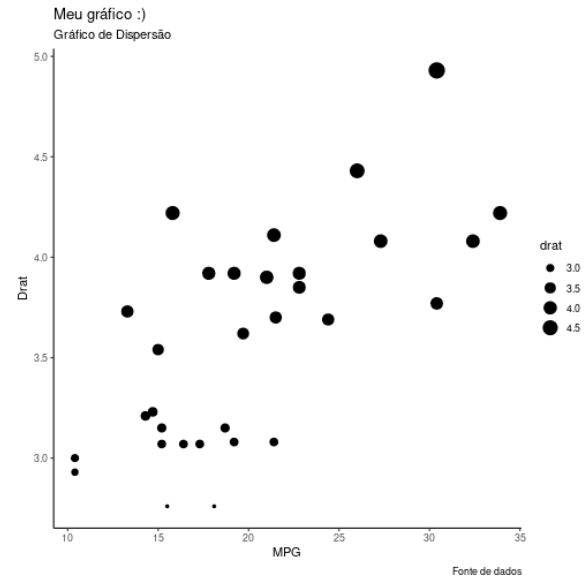
## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat)) +  
  geom_point() +  
  theme_classic() +  
  labs(  
    title = "Meu gráfico :)",  
    subtitle = "Gráfico de Dispersão",  
    x = "MPG",  
    y = "Drat",  
    caption = "Fonte de dados")
```



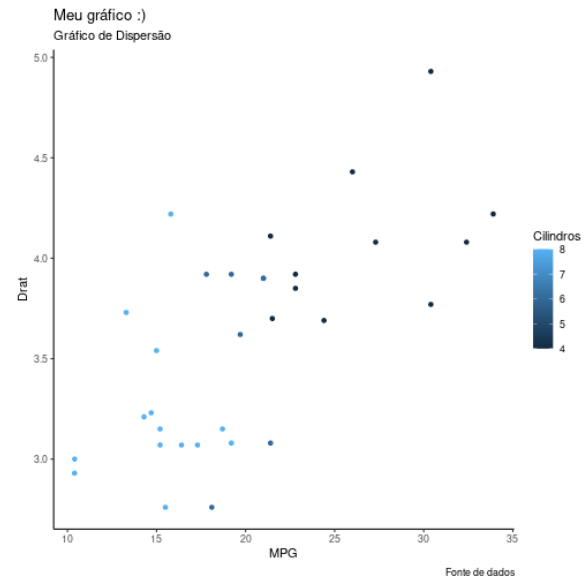
## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat)) +  
  geom_point(aes(size = drat)) +  
  theme_classic() +  
  labs(  
    title = "Meu gráfico :)",  
    subtitle = "Gráfico de Dispersão",  
    x = "MPG",  
    y = "Drat",  
    caption = "Fonte de dados")
```



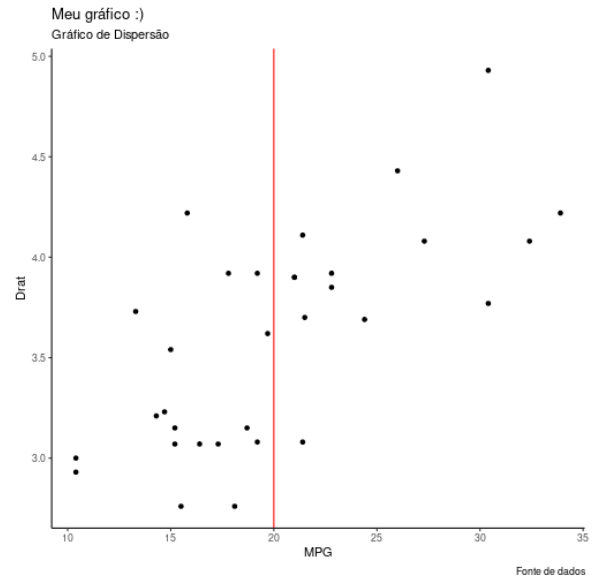
## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat, colour = cyl)) +  
  geom_point() +  
  theme_classic() +  
  labs(  
    title = "Meu gráfico :)",  
    subtitle = "Gráfico de Dispersão",  
    x = "MPG",  
    y = "Drat",  
    caption = "Fonte de dados",  
    colour="Cilindros")
```



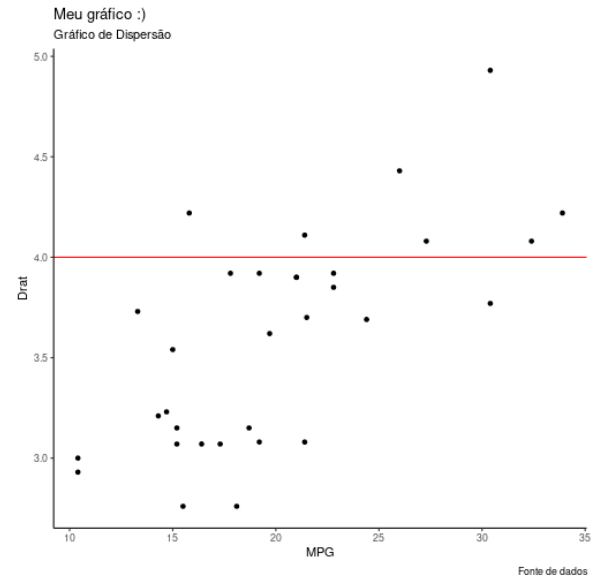
## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat)) +  
  geom_point() +  
  theme_classic() +  
  labs(  
    title = "Meu gráfico :)",  
    subtitle = "Gráfico de Dispersão",  
    x = "MPG",  
    y = "Drat",  
    caption = "Fonte de dados") +  
  geom_vline(xintercept = 20, col="red")
```



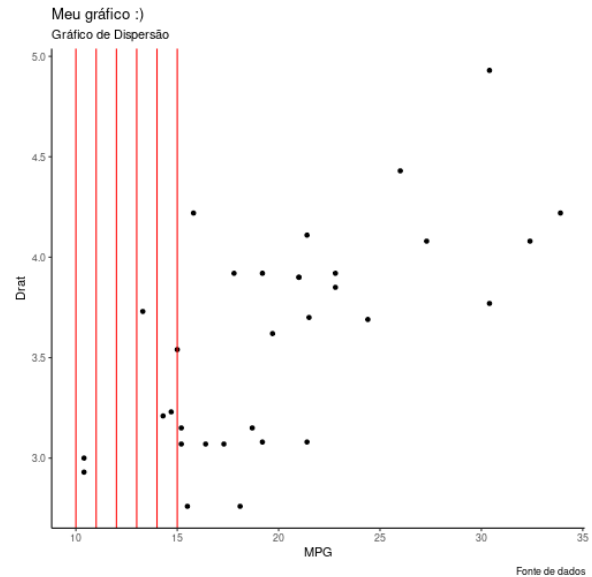
## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat)) +  
  geom_point() +  
  theme_classic() +  
  labs(  
    title = "Meu gráfico :)",  
    subtitle = "Gráfico de Dispersão",  
    x = "MPG",  
    y = "Drat",  
    caption = "Fonte de dados") +  
  geom_hline(yintercept = 4, col="red")
```



## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat)) +  
  geom_point() +  
  theme_classic() +  
  labs(  
    title = "Meu gráfico :)",  
    subtitle = "Gráfico de Dispersão",  
    x = "MPG",  
    y = "Drat",  
    caption = "Fonte de dados") +  
  geom_vline(xintercept = 10:15, col="red")
```

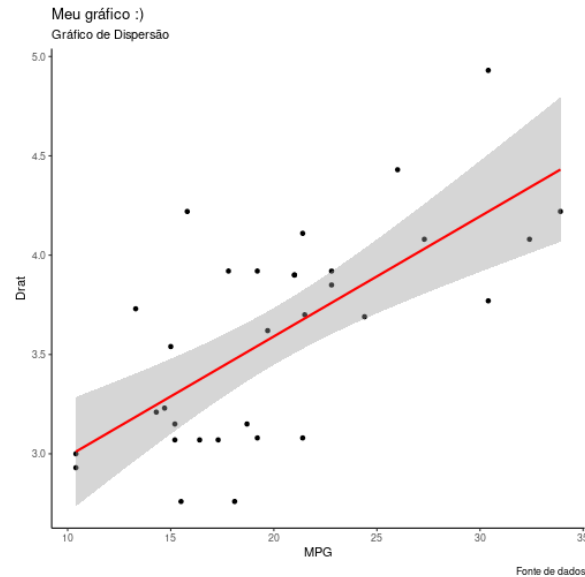






## Base de Dados mtcars + Gráfico de Dispersão

```
ggplot(mtcars, aes(mpg, drat)) +  
  geom_point() +  
  theme_classic() +  
  labs(  
    title = "Meu gráfico :)",  
    subtitle = "Gráfico de Dispersão",  
    x = "MPG",  
    y = "Drat",  
    caption = "Fonte de dados") +  
  geom_smooth(method = "lm", se = TRUE, col="red")
```



# Base de Dados mtcars + Gráfico de Dispersão

## Exercício

Criar um gráfico de dispersão a partir da base 'mtcars' e usando o geom\_smooth

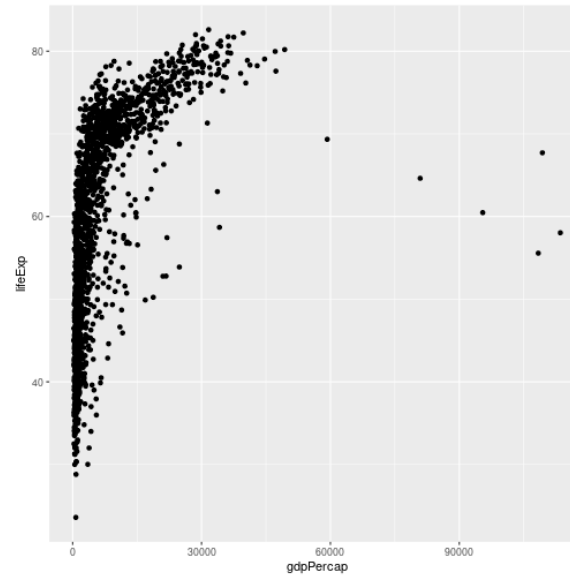
```
#Conjunto de bibliotecas para ciência de dados  
install.packages("tidyverse"); library(tidyverse)
```

## Base de Dados gapminder

```
data(gapminder)  
glimpse(gapminder)  
#> Rows: 1,704  
#> Columns: 6  
#> $ country <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ...  
#> $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ...  
#> $ year <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ...  
#> $ lifeExp <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8...  
#> $ pop <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12...  
#> $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ...
```

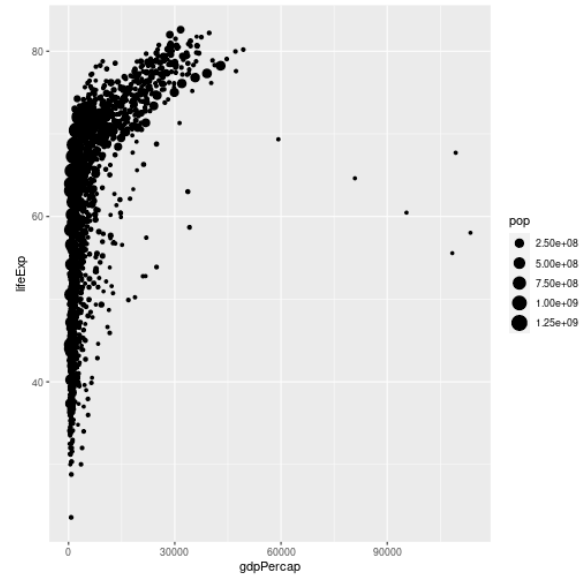
## Base de Datos gapminder + Gráfico de Bolhas

```
ggplot(gapminder, aes(gdpPerCap, lifeExp)) +  
  geom_point()
```



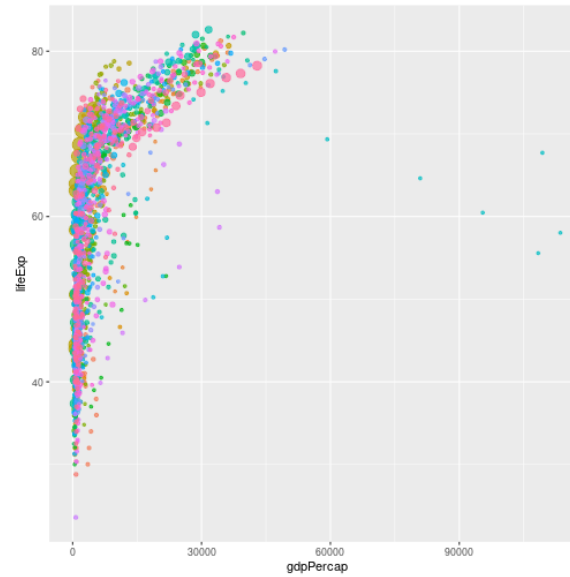
## Base de Datos gapminder + Gráfico de Bolhas

```
ggplot(  
  gapminder,  
  aes(x = gdpPerCap, y=lifeExp, size = pop)) +  
  geom_point()
```



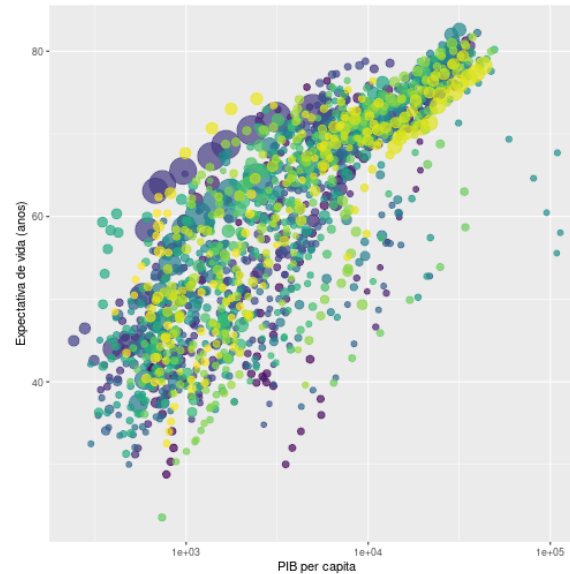
## Base de Dados gapminder + Gráfico de Bolhas

```
ggplot(  
  gapminder,  
  aes(x = gdpPerCap, y=lifeExp, size = pop, colour = c  
  ) +  
  geom_point(show.legend = FALSE, alpha = 0.7)
```




## Base de Datos gapminder + Gráfico de Bolhas

```
ggplot(  
  gapminder,  
  aes(x = gdpPercap, y=lifeExp, size = pop, colour = c  
  ) +  
  geom_point(show.legend = FALSE, alpha = 0.7) +  
  scale_color_viridis_d() +  
  scale_size(range = c(2, 12)) +  
  scale_x_log10() +  
  labs(x = "PIB per capita", y = "Expectativa de vida
```



# Uma colinha para ajudar a lembrar dos comandos

## Data Visualization with ggplot2 : : CHEAT SHEET



Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

### Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geom** objects (maps) that represent data points.

To display values, map variables in the data to visual properties of the geom: **aesthetics**, like **size**, **color**, and **shape**.

Complete the template below to build a graph.

```
ggplot(data = DATA) +
  COORDINATE FUNCTION +
  GEOM FUNCTION +
  AESTHETIC FUNCTION +
  SCALE FUNCTION +
  THEME FUNCTION
```

**ggplot(data = mpg, aes(x = displ, y = hwy))** Begins a plot that you finish by adding layers to. Add one geom function per layer.

**geom(x = displ, y = hwy, data = mpg, geom = "point")** Creates a complete plot with geom data, geom, and mappings. Includes many useful defaults.

**last\_layer()** Returns the last plot

**ggplot("file.png", width = 5, height = 8)** Saves last plot as a file named "file.png" in the working directory. Matches file type to file extension.

### Geoms

**GRAPHICAL PRIMITIVES**  
`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_circle()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`

**LINE SEGMENTS**  
`geom_abline()`  
`geom_hline()`  
`geom_vline()`  
`geom_spoke()`

**ONE VARIABLE** continuous  
`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`

### TWO VARIABLES

continuous x, continuous y  
`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_circle()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`

discrete x, continuous y  
`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_circle()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`

discrete x, discrete y  
`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_circle()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`

### THREE VARIABLES

`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_circle()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`

`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_circle()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`

`geom_area()`  
`geom_bar()`  
`geom_boxplot()`  
`geom_circle()`  
`geom_density_2d()`  
`geom_histogram()`  
`geom_jitter()`  
`geom_line()`  
`geom_point()`  
`geom_rect()`  
`geom_ridge()`  
`geom_segment()`  
`geom_smooth()`  
`geom_text()`  
`geom_tile()`  
`geom_violin()`





# Uma colinha para ajudar a lembrar dos comandos

The image is a collage of cheat sheets for the ggplot2 package in R. It features a central logo for 'ggplot2' and several sections, each with a title and a list of R code snippets. The sections include: 'Stats' (An alternative way to build a layer), 'Scales' (Scale map data values to the visual values of an aesthetic), 'Coordinate Systems' (Scale map data values to the visual values of an aesthetic), 'Faceting' (Facets divide a plot into subplots based on the values of one or more discrete variables), 'Position Adjustments' (Position adjustments determine how to arrange geoms that would otherwise occupy the same space), 'Labels' (Labels are text annotations for a plot), 'Legends' (A legend is a visual key for each aesthetic color, legend, or more than one), 'Themes' (Themes are a collection of styling options for a plot), and 'Zooming' (Zooming is a way to focus on a specific part of a plot). Each section contains several lines of R code, often accompanied by small icons or symbols. The background is a light green color with a subtle pattern.

R Studio

RStudio is a trademark of RStudio, Inc. - CC BY SA RStudio - info@rstudio.com - 866-489-1222 - rstudio.com - Learn more at <http://ggplot2.tidyverse.org> - ggplot2 3.3.0 - Updated: 2019-12

## RStudio CheatSheets: ggplot2

## Próximas aulas

- Gráfico de linhas.
- Gráfico de cascata (waterfall graph).
- Gráfico de mapa de árvore (tree map graph).
- Outros tipos de gráficos.
- Como pedir ajuda na hora de construir um gráfico.
- Combinando mais de um tipo de gráfico.
- Salvando e exportando os gráficos gerados no Ggplot2.
- Exercícios práticos.

## Contatos

- Bruna
  - Instagram: @mardedados
  - LinkedIn: brunagarbes
- R-ladies São Paulo:
  - Website RLadies Global: <https://rladies.org/>
  - MeetUp: <https://www.meetup.com/pt-BR/R-Ladies-Sao-Paulo>
  - Twitter: @RLadiesGlobal, @RLadiesSaoPaulo
  - Instagram: @RLadiesSaoPaulo
  - Facebook: @RLadiesSaoPaulo
  - Github: [https://github.com/rladies/meetup-presentations\\_sao-paulo](https://github.com/rladies/meetup-presentations_sao-paulo)

Até a próxima aula!



Ilustração por Allison Horst - Twitter: @allison\_horst