Crash Course de R

Visualização de dados

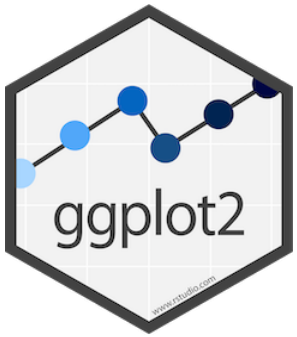
Prof. Carlos Trucíos
 ctruciosm.github.io
 carlos.trucios@facc.ufrj.br

Faculdade de Administração e Ciências Contábeis,
Universidade Federal de Rio de Janeiro

Introdução

Introdução

Data Visualization: interdisciplinary field that deals with the graphic representation of data.



O pacote ggplot2

Comandos básicos:

- `ggplot()`: prepara o ambiente para o gráfico
- `geom_***()`: define o gráfico
- `+`: adicionar *camadas* ao gráfico

ggplot2

ggplot2

```
# install.packages("ggplot2")  # Instalando o ggplot2
library(ggplot2) # Carregando o ggplot2
head(mpg)  # mpg é um dataset do pacote ggplot2
```

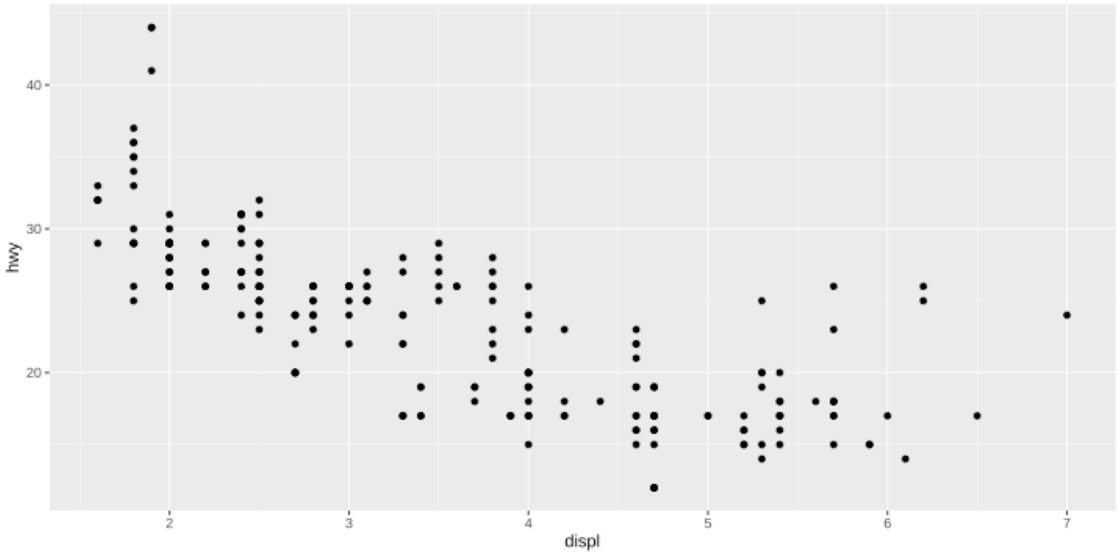
```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa…
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa…
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa…
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa…
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa…
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa…
```

Se quiser conhecer mais do *dataset*, basta digitar `?mpg` ou `help(mpg)`

> Se você tiver instalado o pacote `tidyverse`, basta carregar ele diretamente `library(tidyverse)`. O pacote `tidyverse` já contem o pacote `ggplot2`

ggplot2: estruturas básicas

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



ggplot2: estruturas básicas

- `ggplot()`: crea o arcabouço do gráfico onde iremos adicionar (+) as diferentes *camadas*, o argumento *data* recebe nosso conjunto de dados.
- `ggplot()`: sozinho não traz nada interessante, mas prepara o caminho para construirmos os gráficos.
- `geom_point()`: vai incluir a nuvem de pontos (scatter plot) no nosso gráfico. O argumento `aes()` dentro da função `geom_point()` nos diz quais variáveis serão mapeadas para fazer o gráfico (qual variável estará no eixo x e qual no eixo y)

ggplot2: Hands-on

Hands-on

1. Rode o seguinte código

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```

1. Subtitua *color* = *class* por *colour* = *class*
2. Subtitua *color* por *shape*
3. Subtitua *color* por *size*
4. Subtitua *class* por *trans*
5. O que acontece se fizermos (dentro de `geom_point()`)

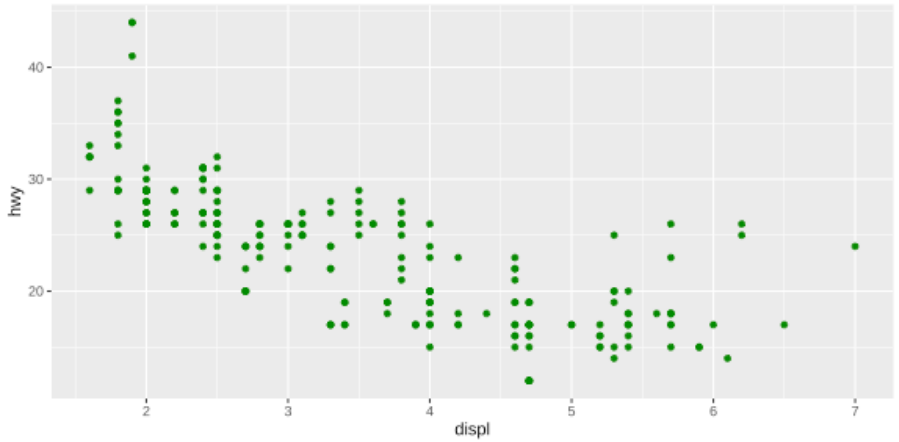
```
mapping = aes(x = displ, y = hwy, color = trans, size = trans)
```

> Com `aes()` podemos incluir tamanho (size), forma (shape), color (color/colour), transparência (alpha),...

ggplot2: estruturas básicas

- Às vezes queremos apenas escolher uma unica cor para os pontos.

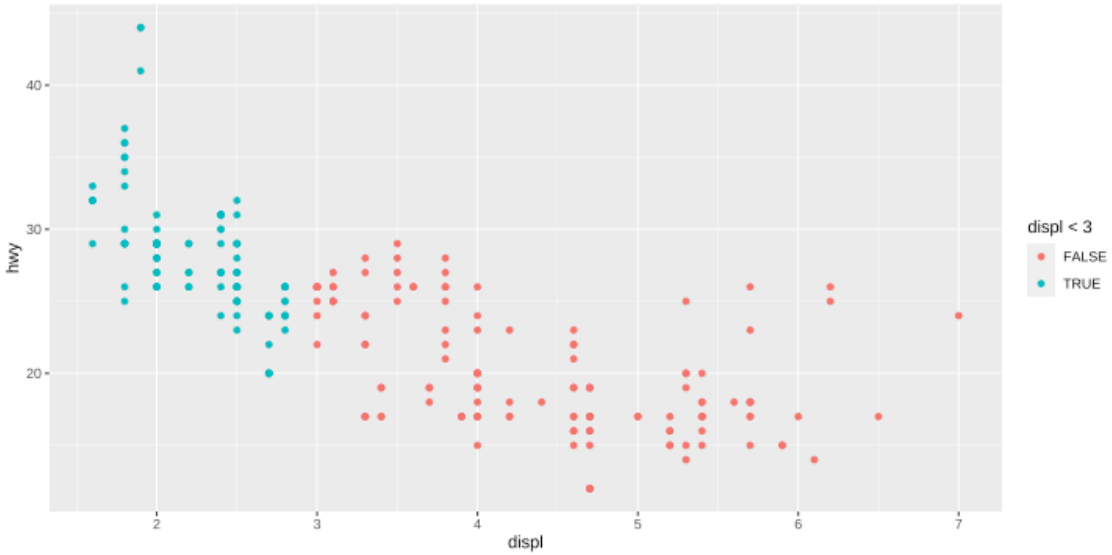
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy),  color = "green4")
```



> Repare que definimos *color* = *"green4"* fora de `aes()`, como mais um argumento no `geom_point()`.

ggplot2: estruturas básicas

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy,
                           color = displ < 3))
```



ggplot2: facets

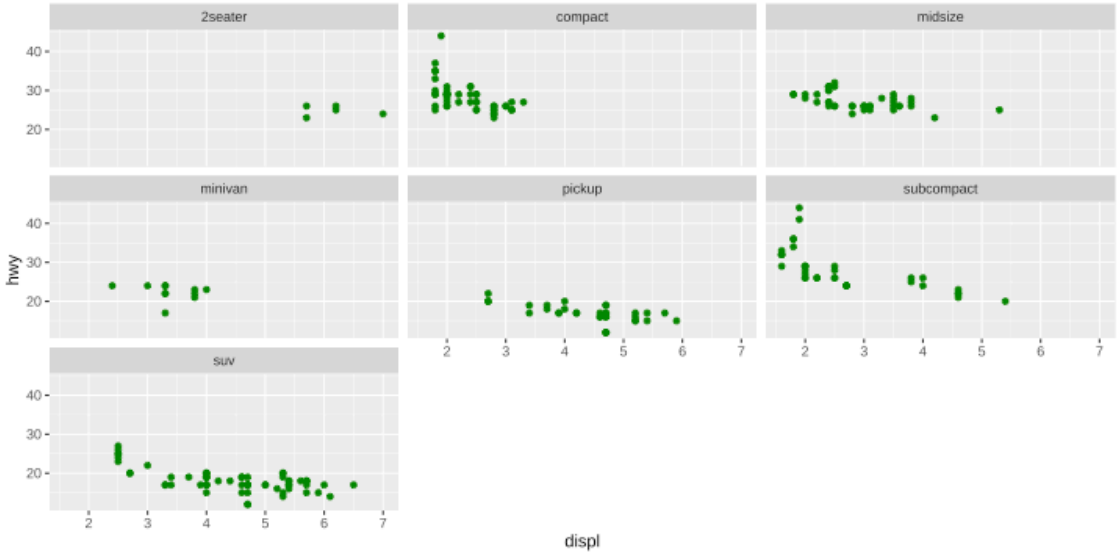
> Cuidado onde colocarmos o sinal "+", "+" deve ir no final de uma linha e não no começo!

Às vezes estamos interessados em visualizar os dados por categoria:

- Uma forma é visualizar tudo em um único gráfico e separar as categorias por cor/tamanho/forma.
- Outra forma é separar o gráfico em *facets* (sub-figuras em que cada sub-figura represente um grupo ou categoria). Para isto, usaremos `facet_wrap(~var_cat)`

ggplot2: facets

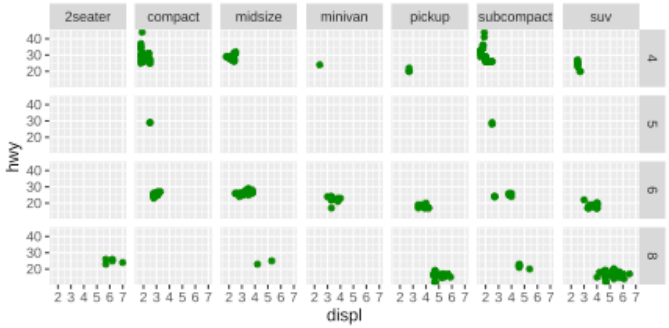
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "green4") +
  facet_wrap(~class)
```



ggplot2: facets

Se quisermos separar as figuras em dois tipos de categorias (2 variáveis categóricas) diferentes, usamos `facet_grid(var_cat1 ~ var_cat2)`

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "green4") +
  facet_grid(cyl~class)
```



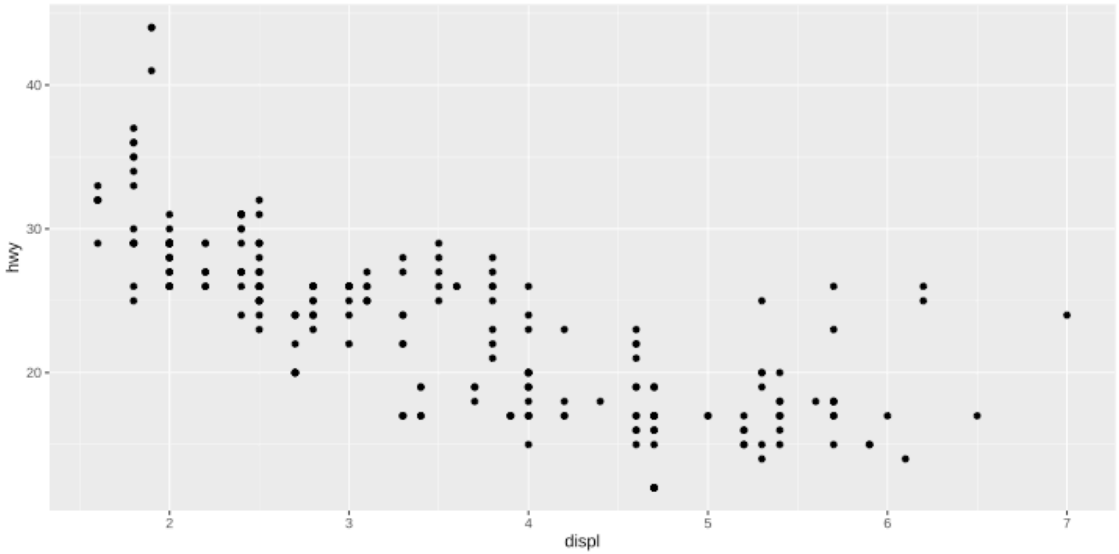
ggplot2: geom_*()

Além de `geom_point()` existem outros tipos de gráficos que podem ser feitos com ggpplot2, por exemplo: `geom_line()`, `geom_smooth()`, `geon_bloxplot()`, `geom_histogram()`, etc.

- `geom_point()`: gráfico de dispersão
- `geom_line()`: grafico de sequência (séries temporais)
- `geom_smooth()`: fará uma curva de suavização (pode fazer uma reta de regressão se incluirmos `method = "lm"`)
- `geom_bloxplot()`: boxplots
- `geom_histogram()`: histogramas

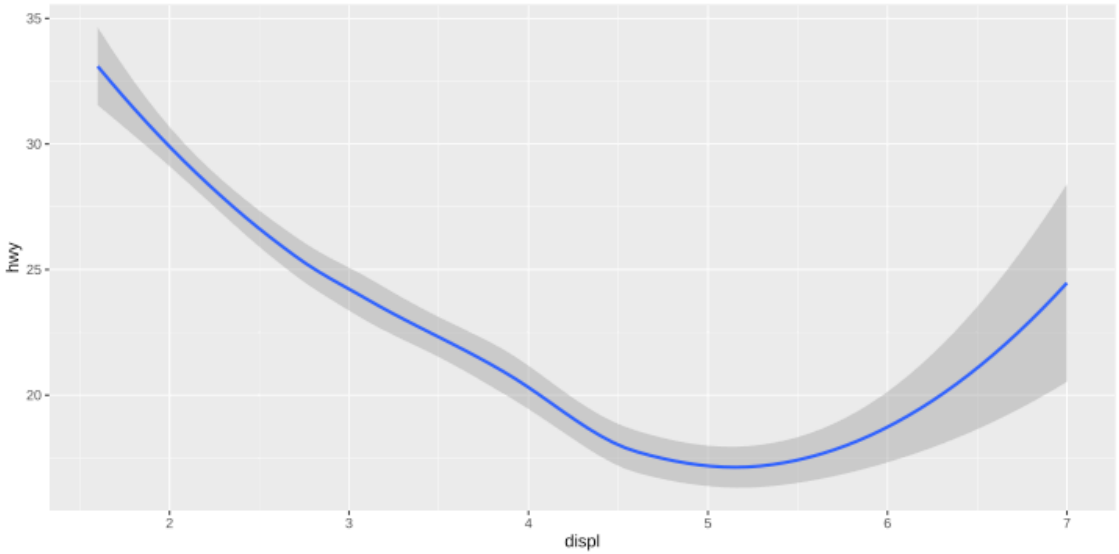
ggplot2: geom_*()

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



ggplot2: geom_*()

```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



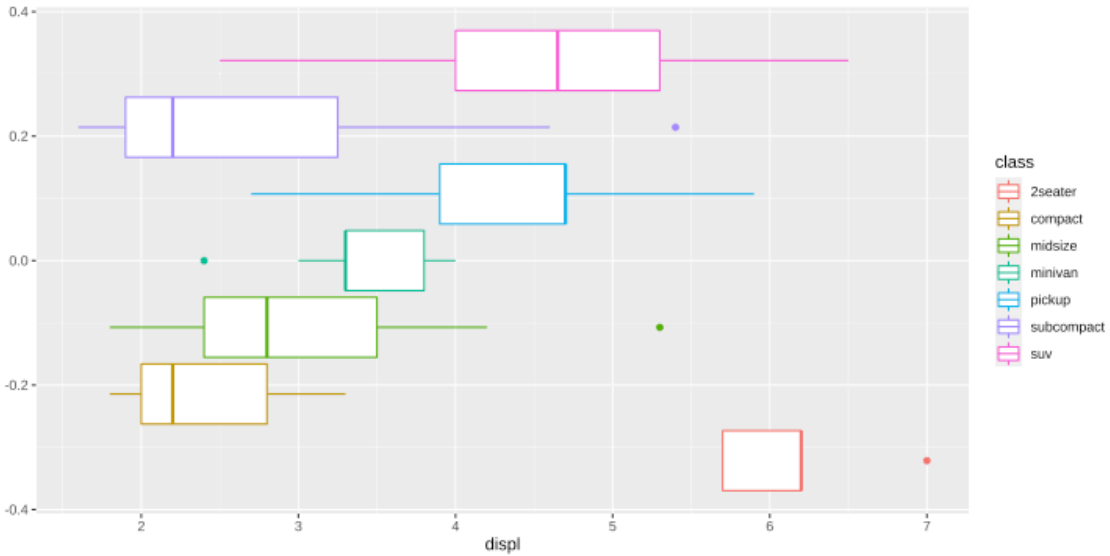
ggplot2: geom_*()

R Code | Plot

```
url <- "https://raw.githubusercontent.com/ctruciosm/ctruciosm.github.io/master/datasets/BTCUSDT.csv"
btc <- read.csv(url)
head(btc)
btc$timestamp <- as.Date(btc$timestamp)
ggplot(data = btc) +
  geom_line(mapping = aes(x = timestamp, y = close))
```

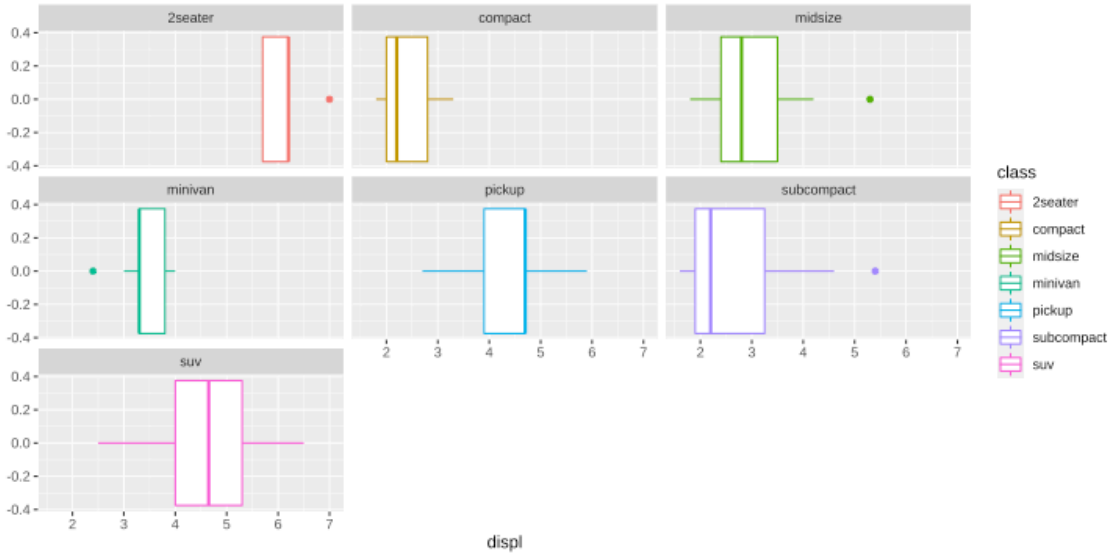
ggplot2: geom_*()

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = displ, color = class))
```



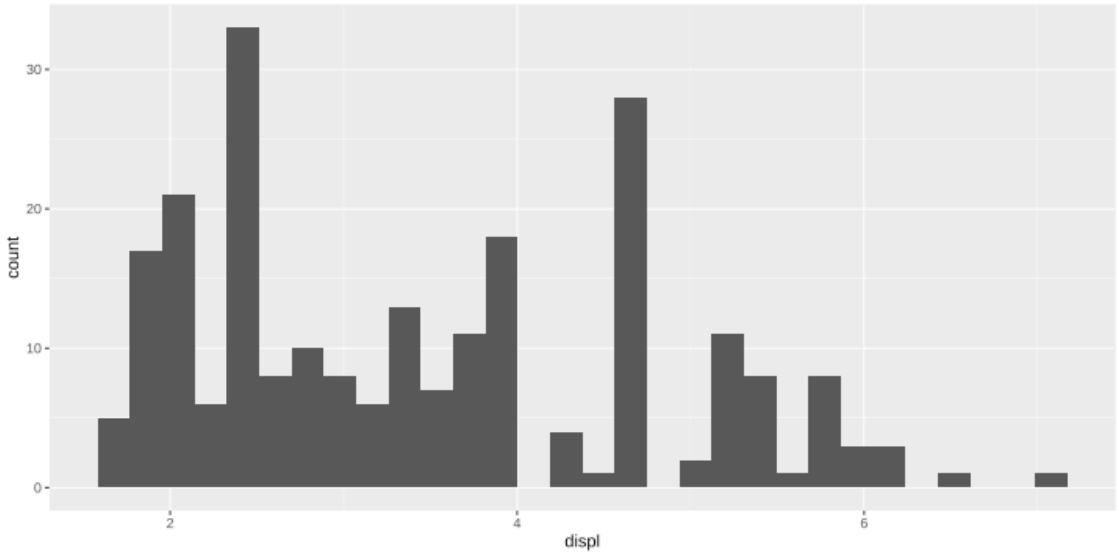
ggplot2: geom_*()

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = displ, color = class)) +
  facet_wrap(~class)
```



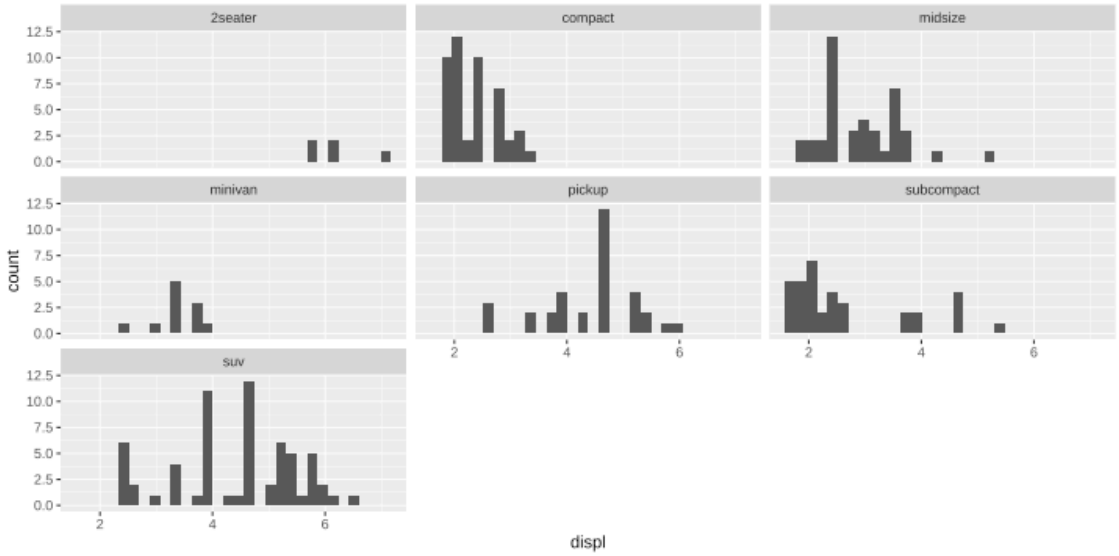
ggplot2: geom_*()

```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = displ))
```



ggplot2: geom_*() + facet

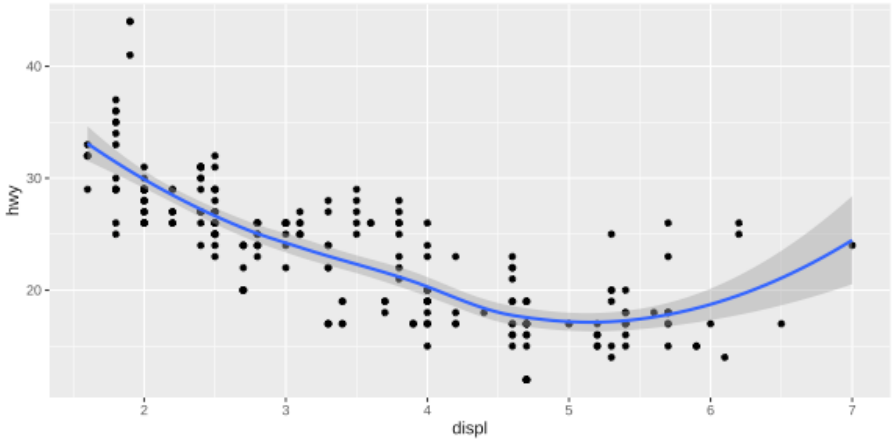
```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = displ)) +
  facet_wrap(~class)
```



ggplot2: geom_*() + facet

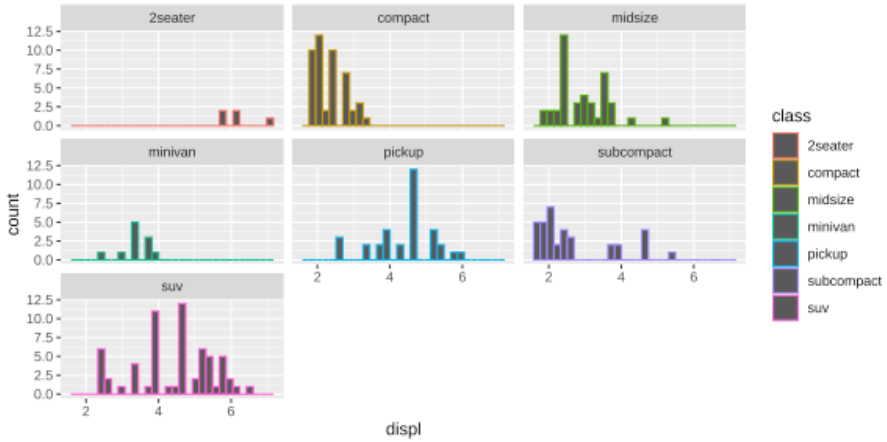
- Se formos utilizar sempre as mesmas variáveis mapeadas, podemos definir o `aes()` dentro do `ggplot()`.
- Também podemos incluir várias *camadas de gráficos* e misturar com `facet_grid()`/`facet_wrap()`

```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```



ggplot2: geom_*() + facet

```
ggplot(data = mpg, aes(x = displ, color = class)) +
  geom_histogram() +
  facet_wrap(~class)
```



O que acontece de mudarmos `color = class` por `fill = class`?

ggplot2: Hands-on

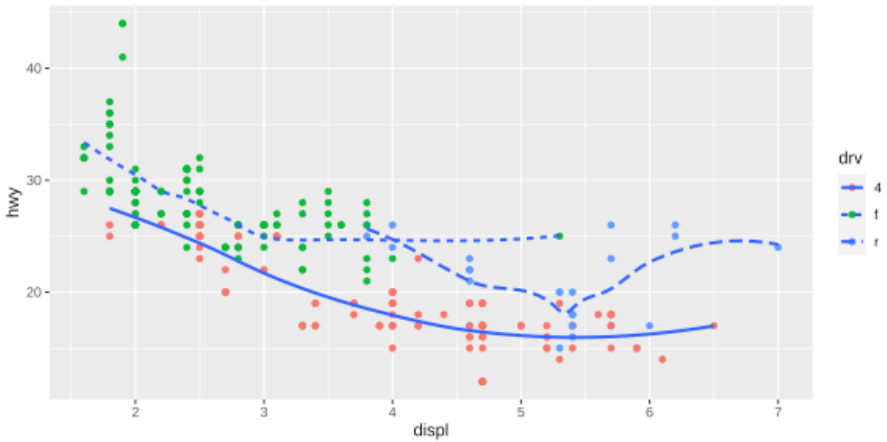
Hands-on (utilizando o *dataset* `mpg`)

1. Faça um gráfico de dispersão com x = displ, y = hwy
2. No mesmo gráfico, utilize geom_smooth() para incluir uma curva suavizada
3. Que os pontos no gráfico estejam de cores diferentes segundo drv
4. O que acontece se incluirmos linetype = drv dentro do aes() referente à suavização?
5. Substitua o linetype por color
6. dentro do geom_smooth() mas fora do aes(), escreva se = FALSE

ggplot2: mix

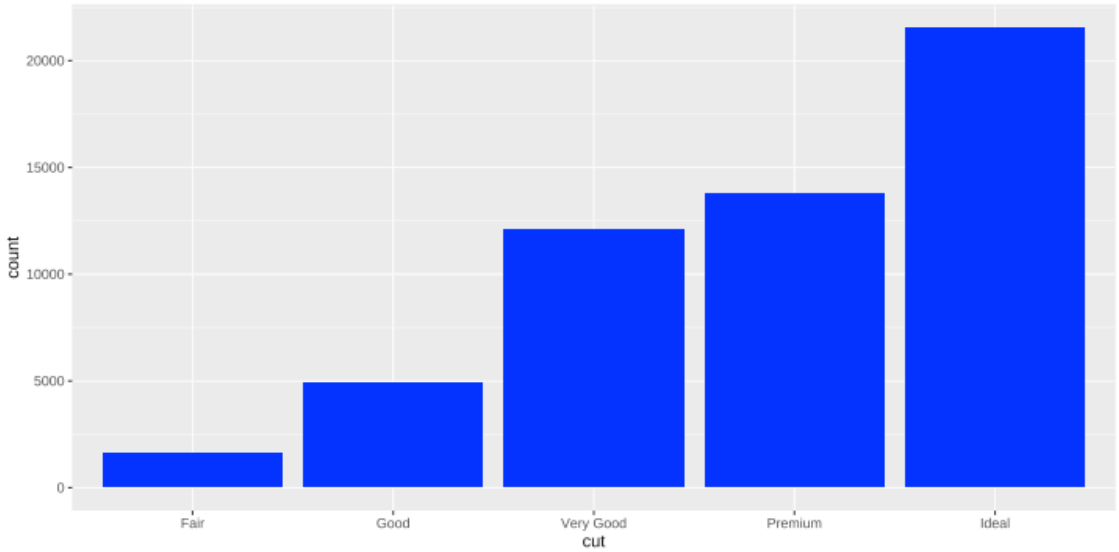
Gabarito

```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy, color = drv)) +
  geom_smooth(aes(x = displ, y = hwy, linetype = drv),
              se = FALSE)
```



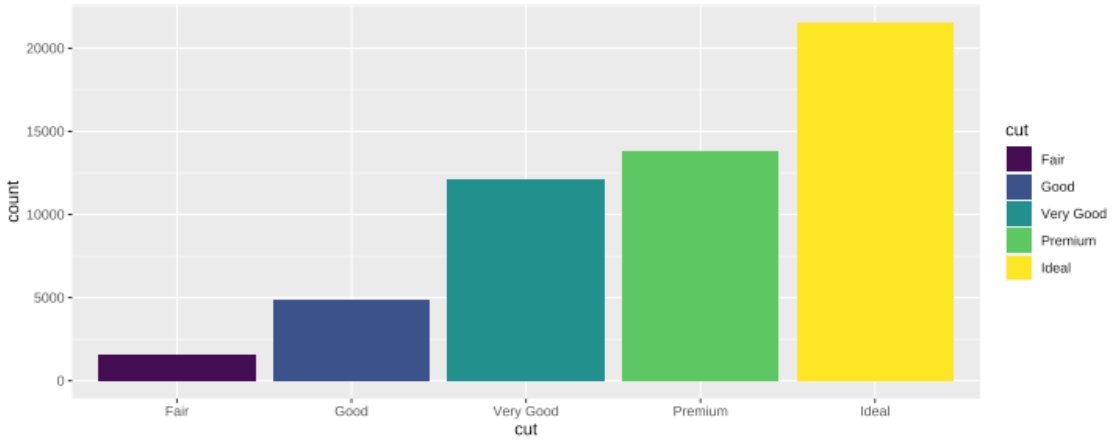
ggplot2: gráfico de barras

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut), fill = "blue")
```



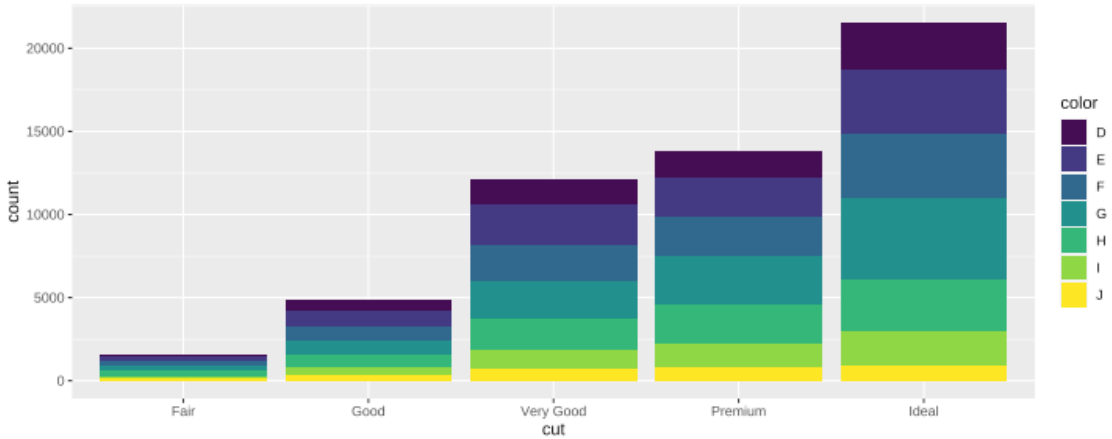
ggplot2: gráfico de barras

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut))
```



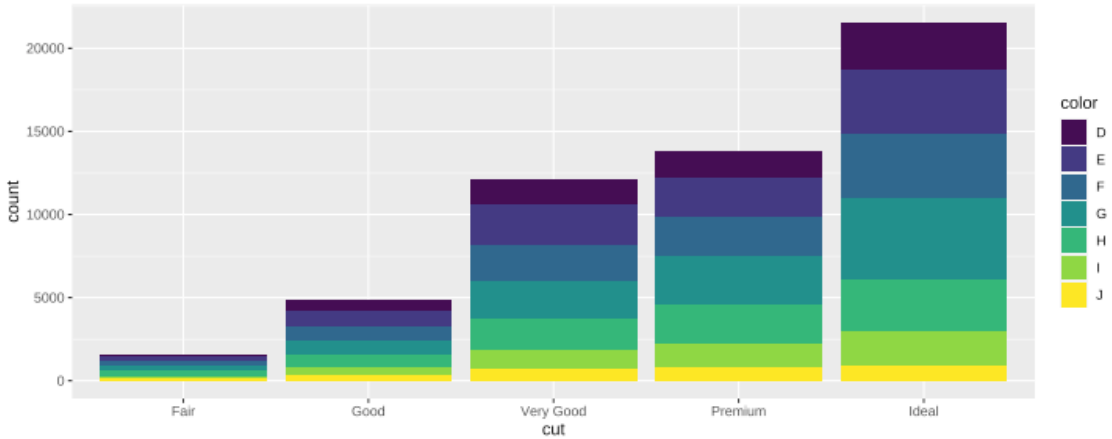
ggplot2: gráfico de barras

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = color))
```



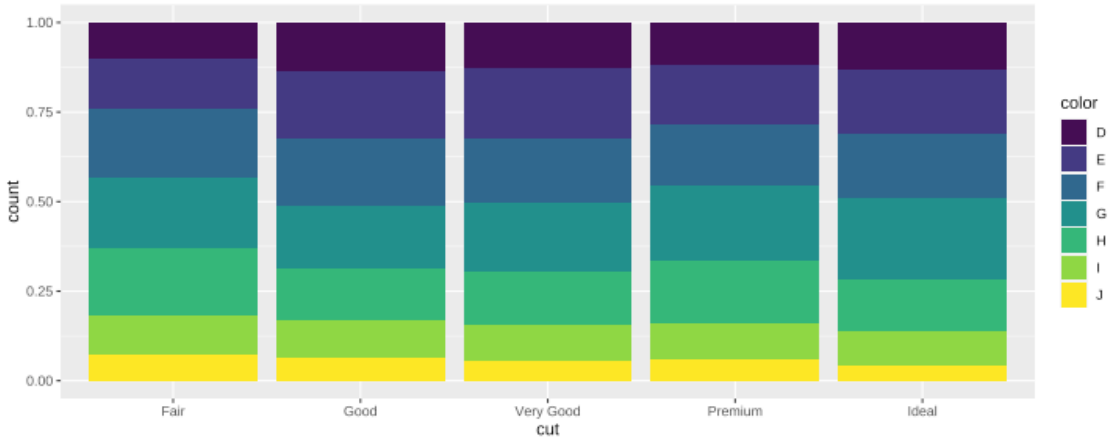
ggplot2: argumento position

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = color), position = "stack")
```



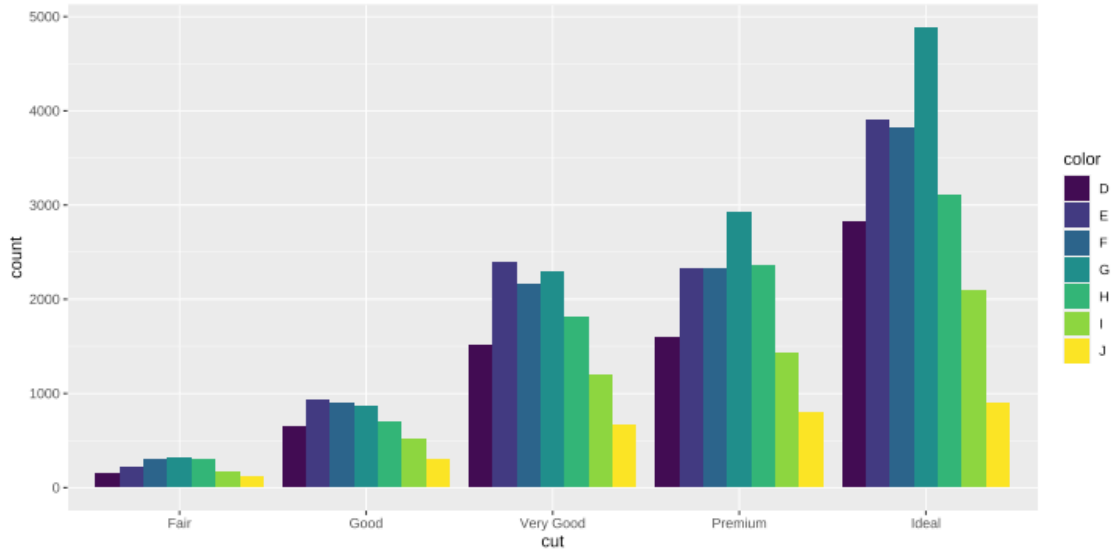
ggplot2: argumento position

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = color), position = "fill")
```



ggplot2: argumento position

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = color), position = "dodge")
```

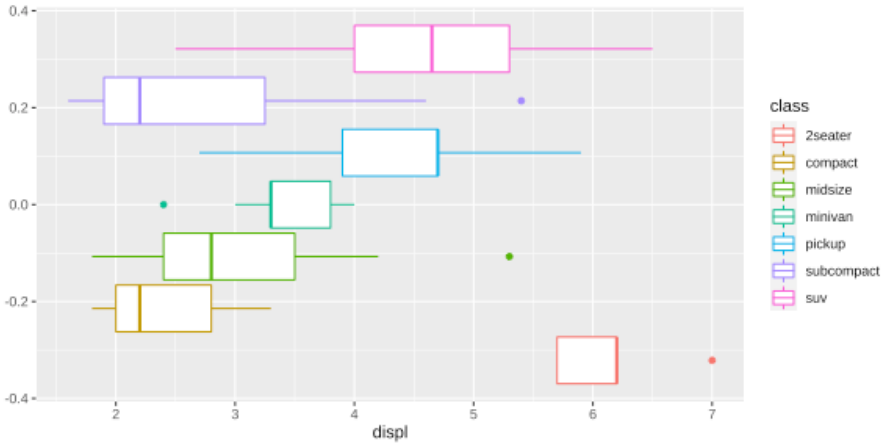


> O argumento position nos ajuda a fazer ajustes da posição: "identity" ,"stack", "fill", "dodge", "jitter"

ggplot2: sistema de coordenadas

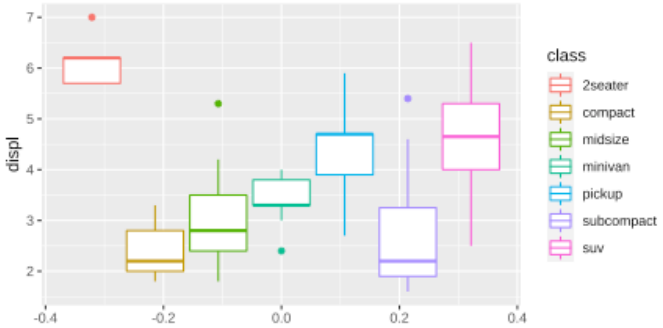
Podemos tambem mudar o sistema de coordenadas

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = displ, color = class))
```



ggplot2: sistema de coordenadas

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = displ, color = class)) +
  coord_flip()
```



- coord_quickmap(): para dar as corretas proporções em mapas (sim, podemos fazer mapas no R!)
- coord_polar(): coordenadas polares

ggplot2: Hands-on

Hands-on

Escreva o seguinte código:

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut))
```

1. Mude os eixos do gráfico (utilize coord_flip())
2. Transforme o gráfico de barras para coordenadas polares (utilize coord_polar())
3. inclua dentro do geom_bar() o argumento width = 1

ggplot2: Hands-on

Gabarito

```
p1 = ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut), width = 1)
p1 + coord_flip()
p1 + coord_polar()
```

ggplot2: Resumo

```
ggplot(data = <dataset>) +
  <geom_tipo>(
     mapping = aes(x = <variavel_x>, y = <variavel_y>, fill = <variavel_fill>),
     stat = <stat_tipo>,
     position = <tipo>
  ) +
  <coord_tipo> +
  <facet_tipo>
```

Existem mais de 40 geom dentro do ggplot2 e muitas extensões

- Preocupado com lembrar como usar o ggplot2? imprima o cheatsheet do ggplot2
- Para ver mais das extensões entre aqui