

# Aula CE1 - Introdução ao Tidyverse

*José Guilherme Lopes*

## Instalando e carregando o tidyverse

## ggplot2: Visualização de Dados

Um exemplo de dataset:

```
head(ggplot2::mpg)
```

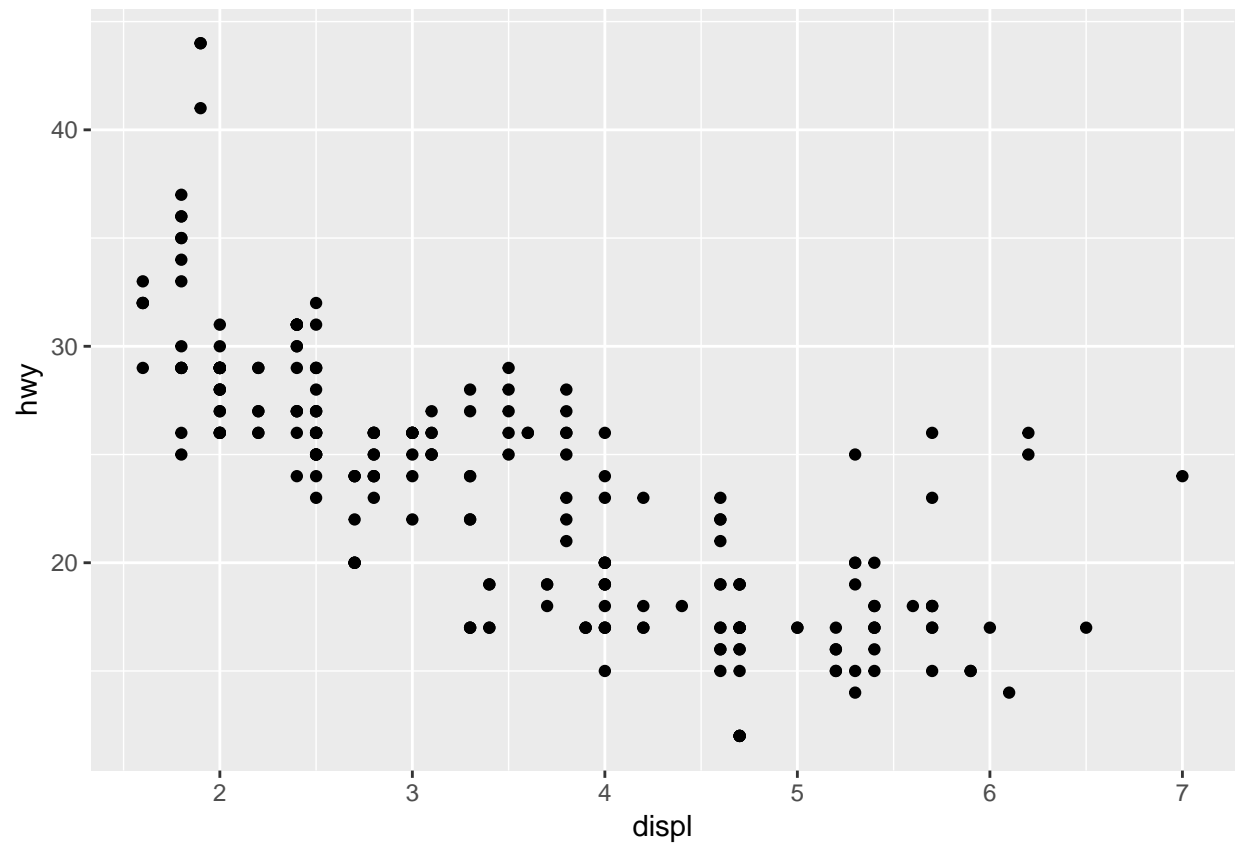
```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans drv     cty   hwy fl      class
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi          a4      1.8  1999     4 auto~ f      18    29 p      comp~
## 2 audi          a4      1.8  1999     4 manu~ f      21    29 p      comp~
## 3 audi          a4      2    2008     4 manu~ f      20    31 p      comp~
## 4 audi          a4      2    2008     4 auto~ f      21    30 p      comp~
## 5 audi          a4      2.8  1999     6 auto~ f      16    26 p      comp~
## 6 audi          a4      2.8  1999     6 manu~ f      18    26 p      comp~
```

### Código básico criar um gráfico no ggplot:

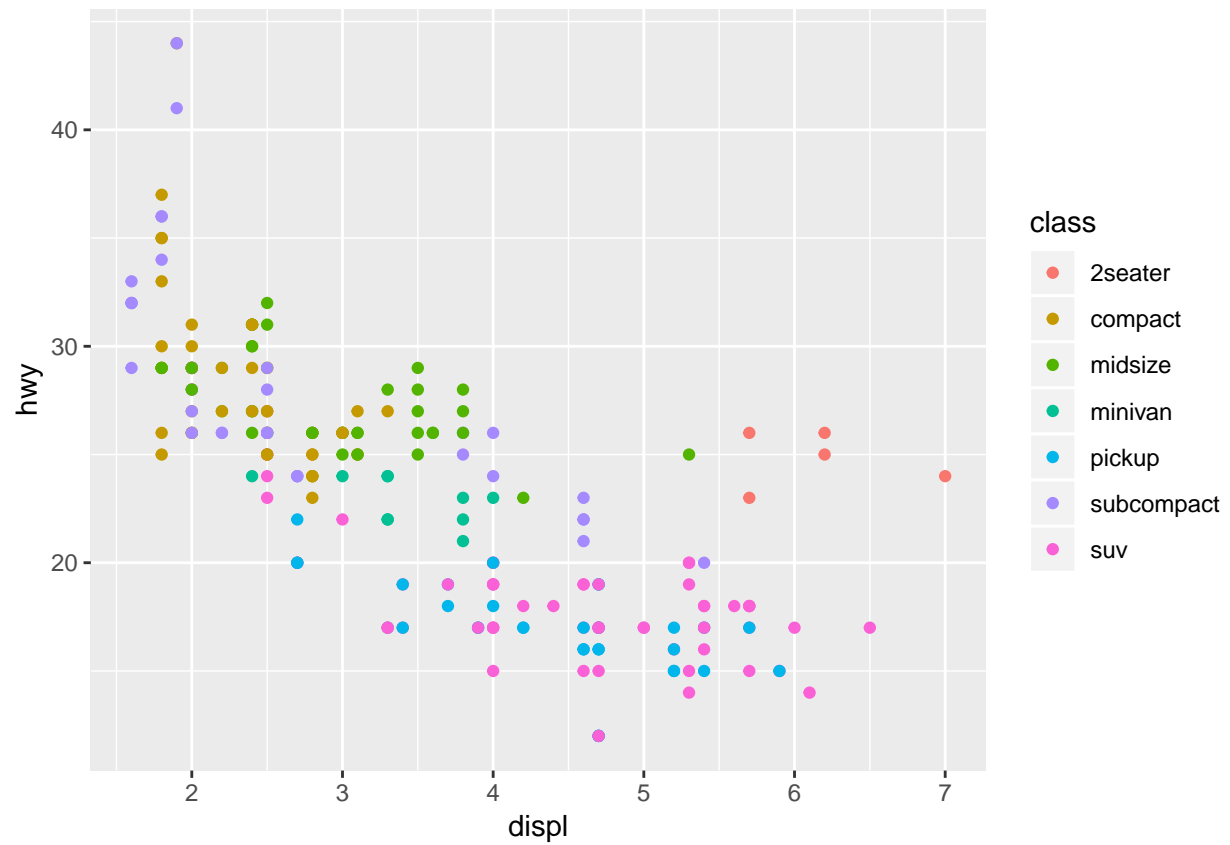
- 1. função `ggplot()` +
- 2. função `geom_XXX()`
- 3. Especificar o dataset com `data =`
- 4. Especificar as variáveis com `mapping = aes()`

Exemplos:

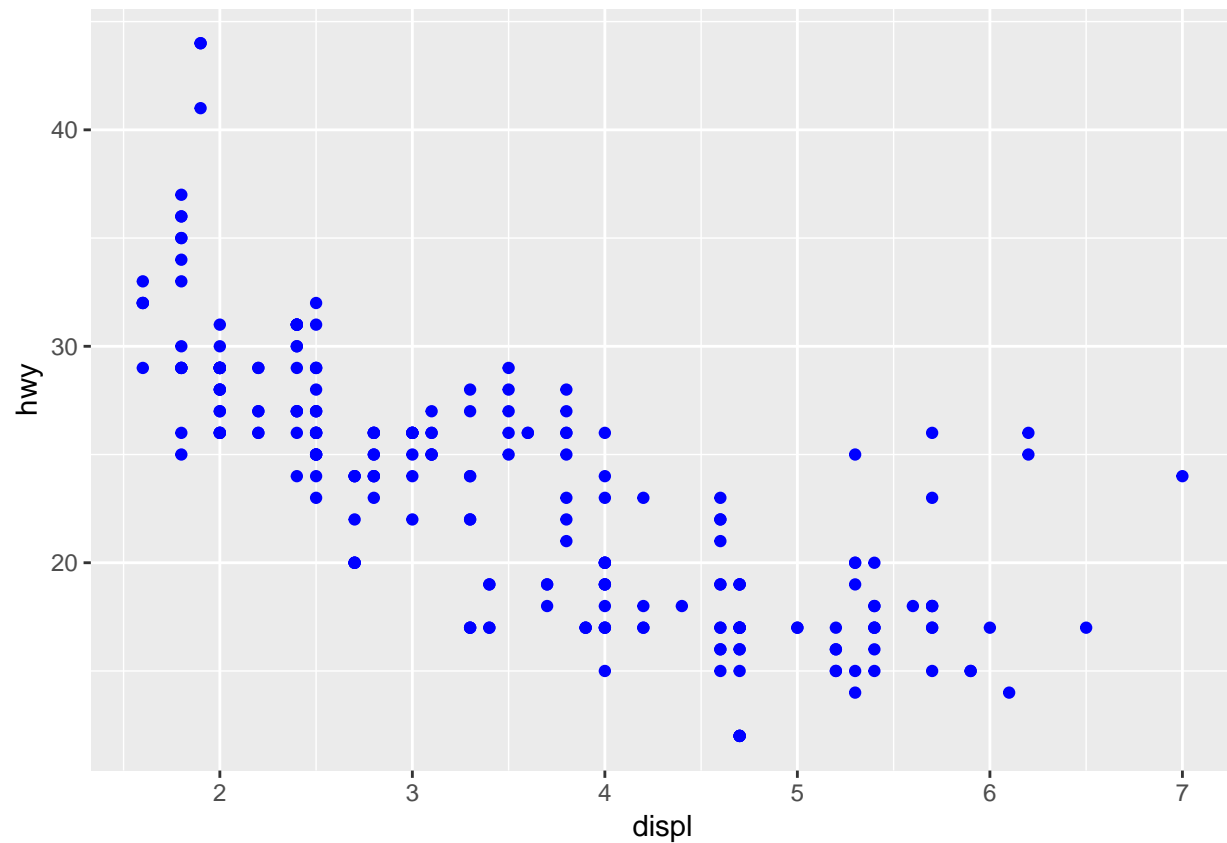
```
ggplot(data=mpg)+
  geom_point(mapping = aes(x=displ, y=hwy))
```



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



```
ggplot(mpg) +
  geom_point(mapping = aes(x=displ, y=hwy), colour="blue")
```

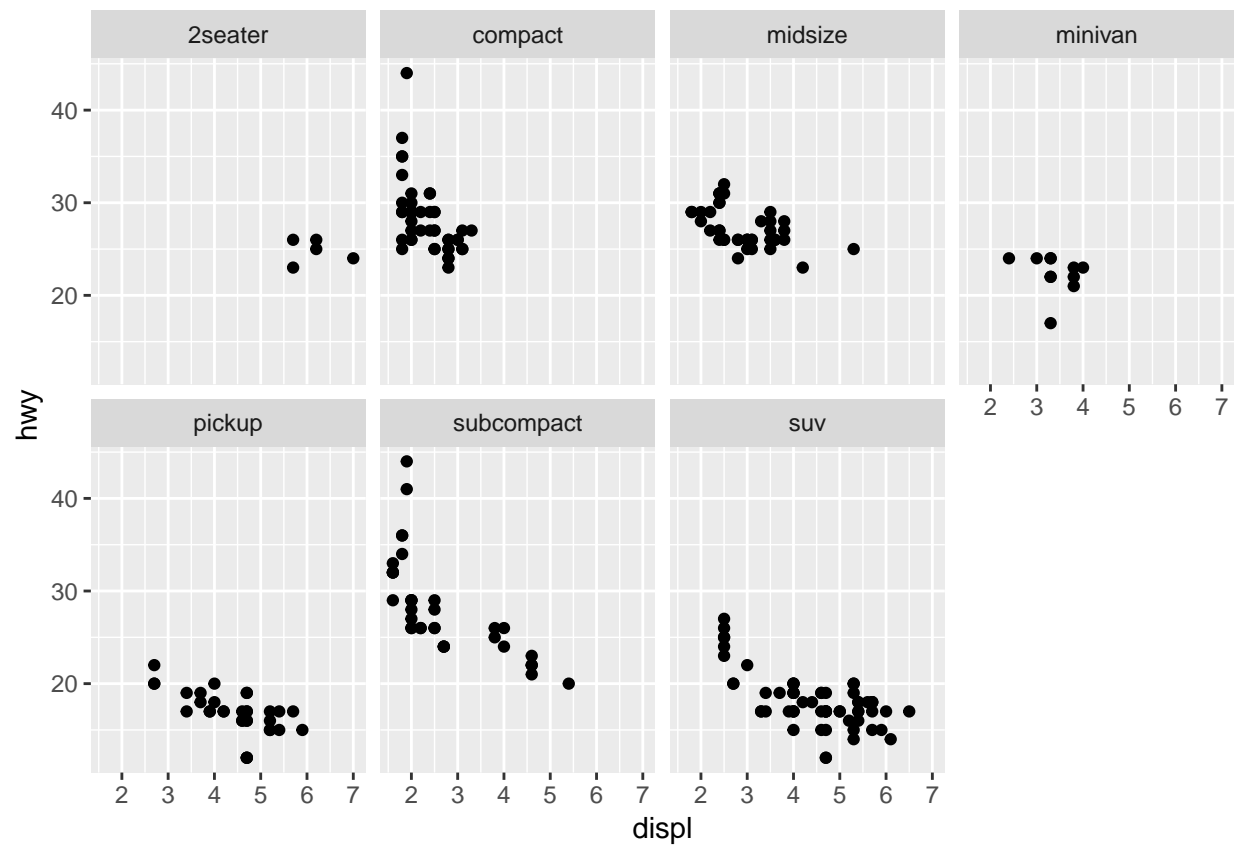


### Facetas:

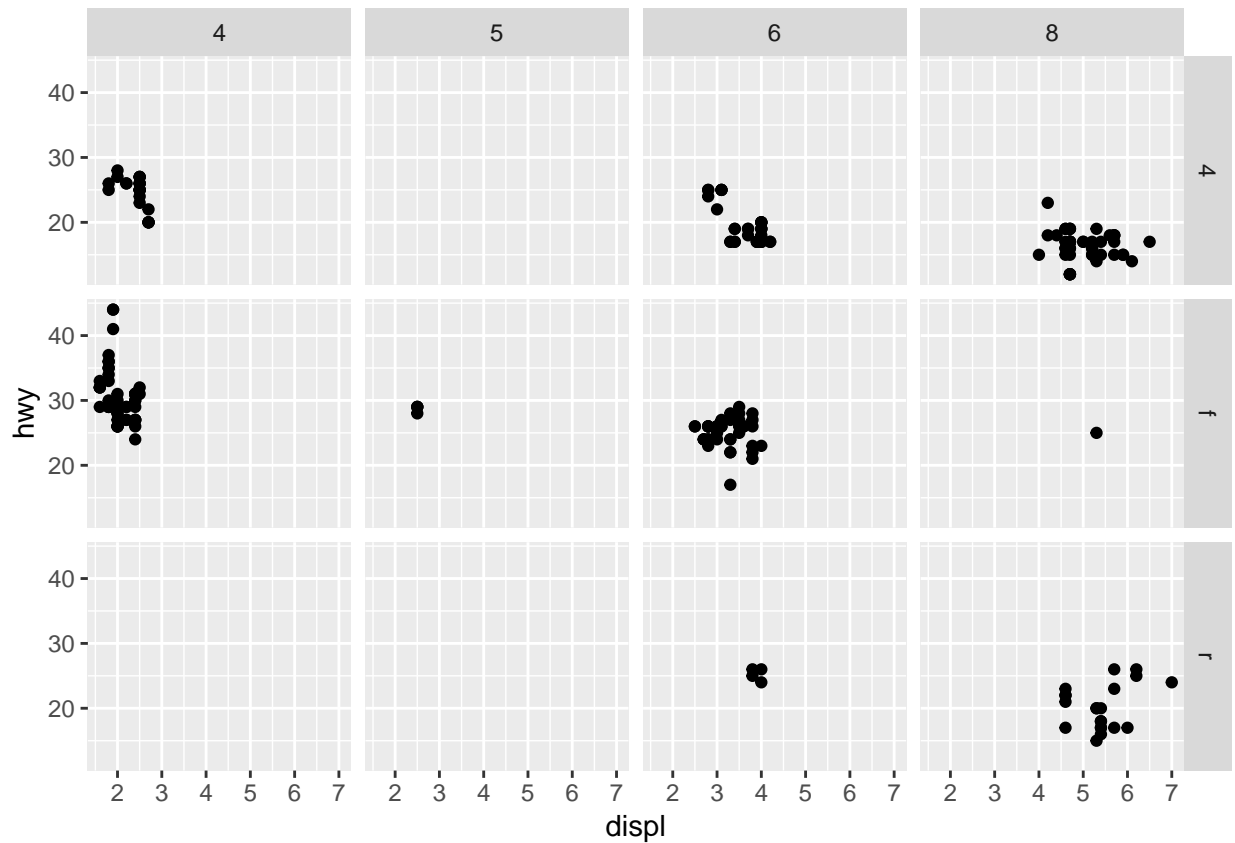
- Particiona o gráfico de acordo com uma variável (de preferência categórica)

Exemplos:

```
ggplot(mpg)+  
  geom_point(mapping = aes(x=displ, y=hwy))+  
  facet_wrap(~ class, nrow=2)
```



```
ggplot(mpg)+
  geom_point(mapping = aes(x=displ, y=hwy))+
  facet_grid(drv ~ cyl)
```



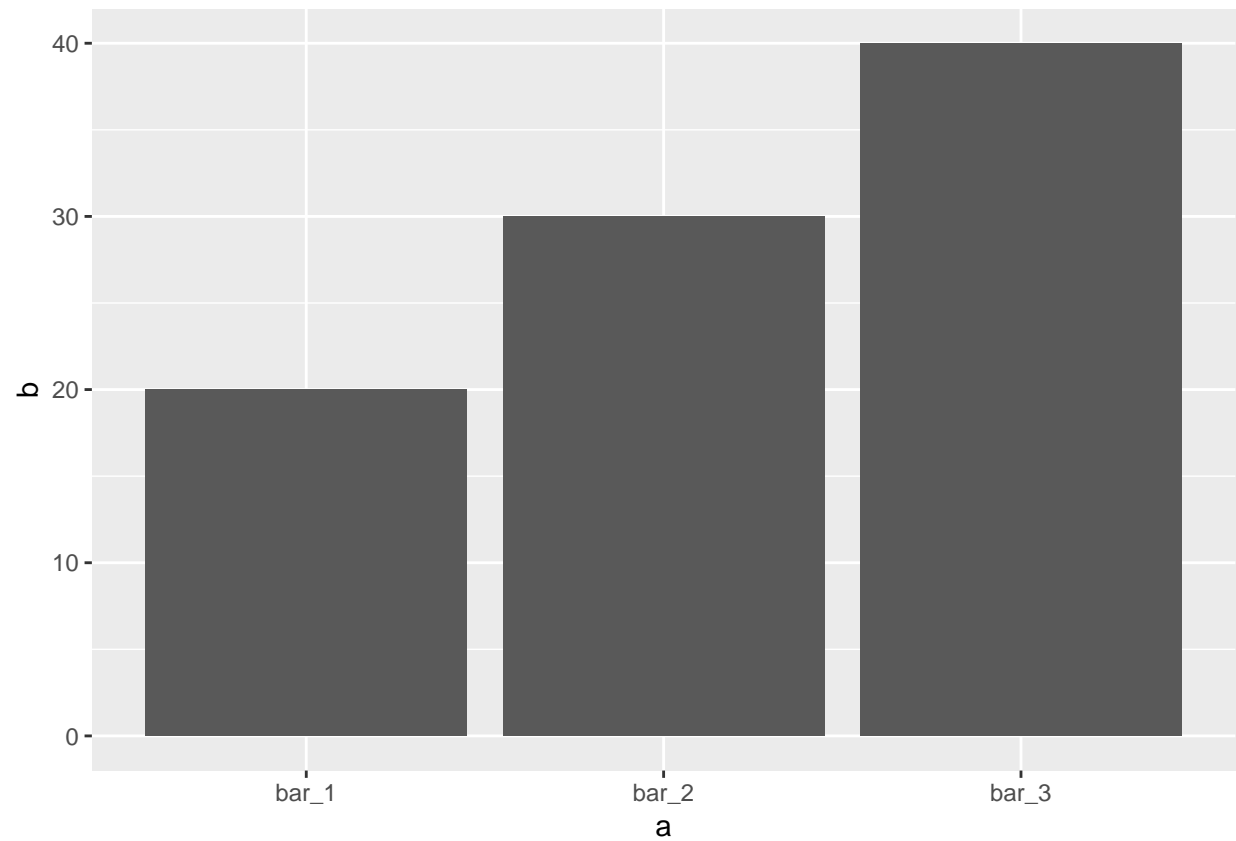
## Transformações Estatísticas

- Permitem alterar a transformação que é feita nos dados para geração do gráfico.

Exemplo:

```
demo <- tribble(~a, ~b, "bar_1", 20, "bar_2", 30, "bar_3", 40)

ggplot(data=demo)+
  geom_bar(mapping = aes(x=a, y=b), stat="identity")
```

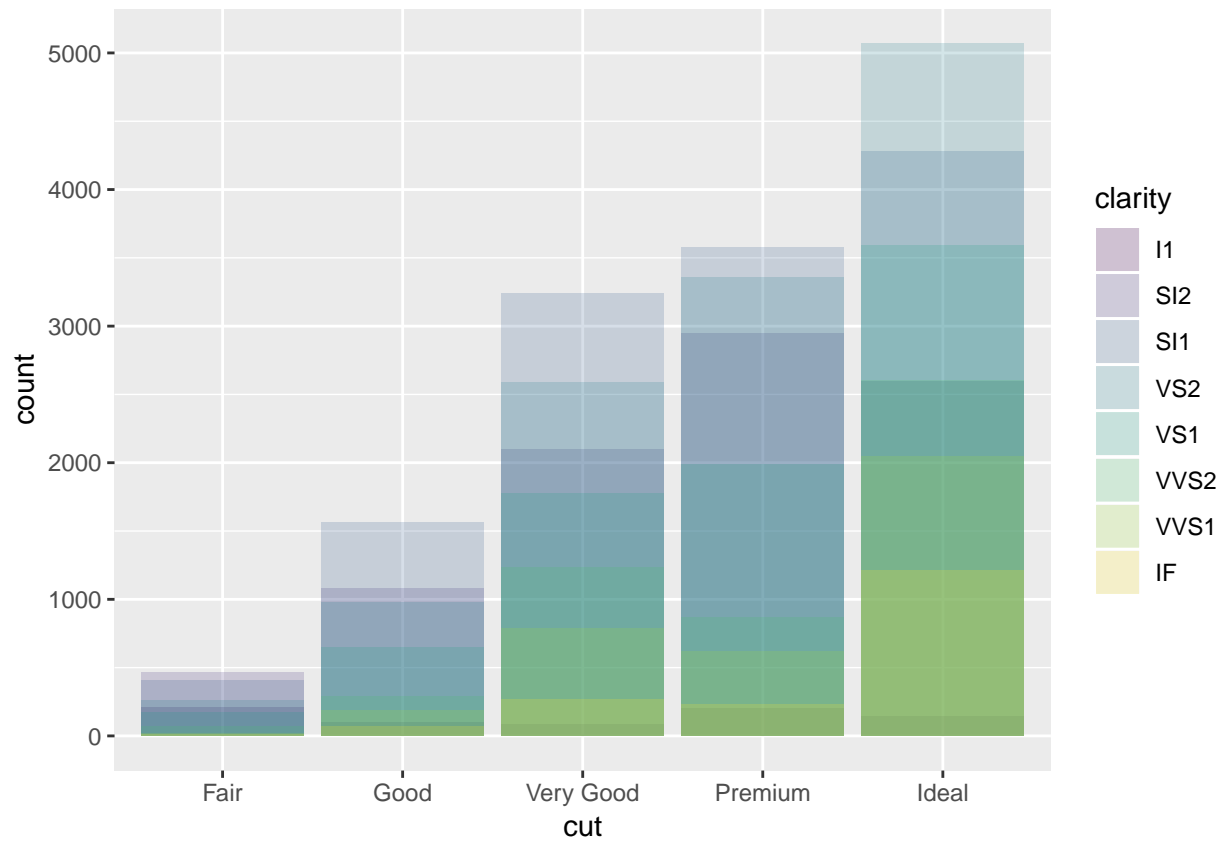


## Ajustes de Posição

- Permitem alterar a posição em que os objetos são inseridos no gráfico.

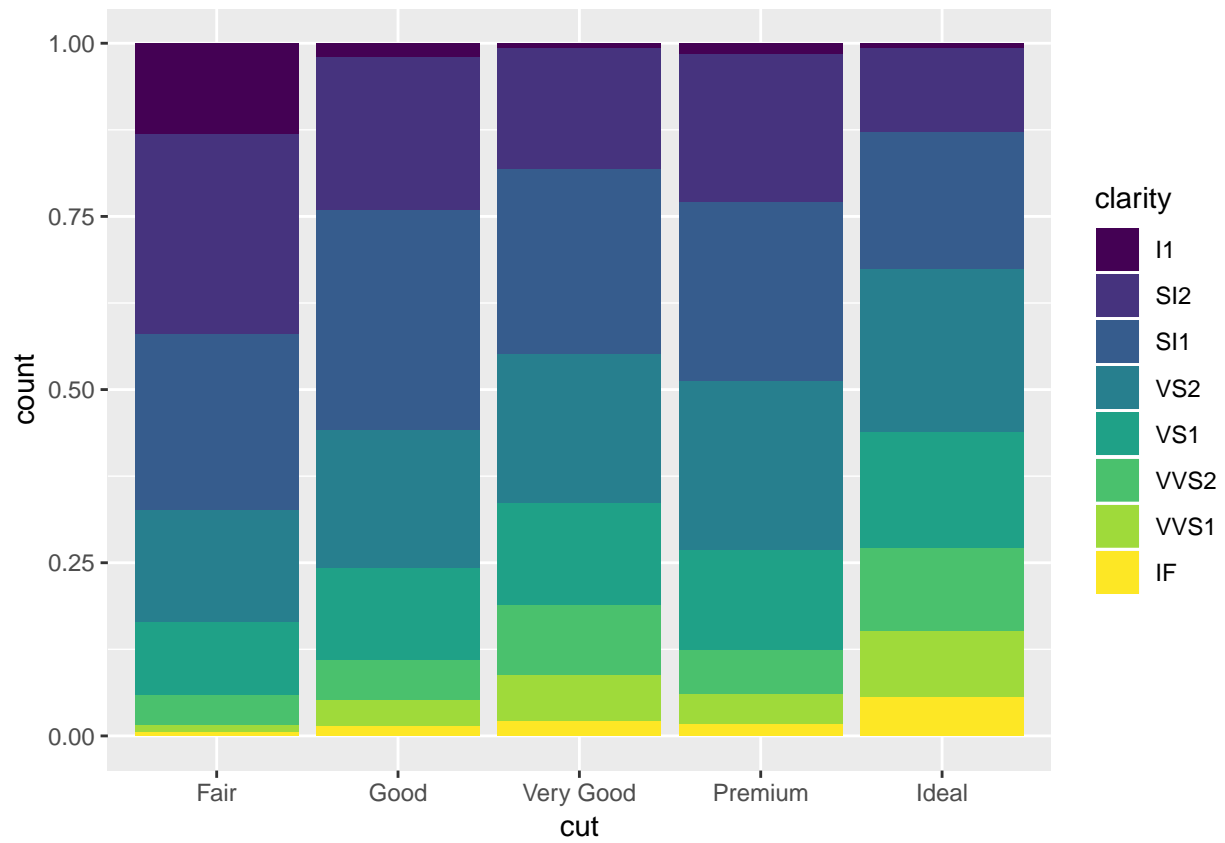
Exemplos:

```
ggplot(data=diamonds, mapping = aes(x=cut, fill=clarity)) +  
  geom_bar(position="identity", alpha=0.20)
```

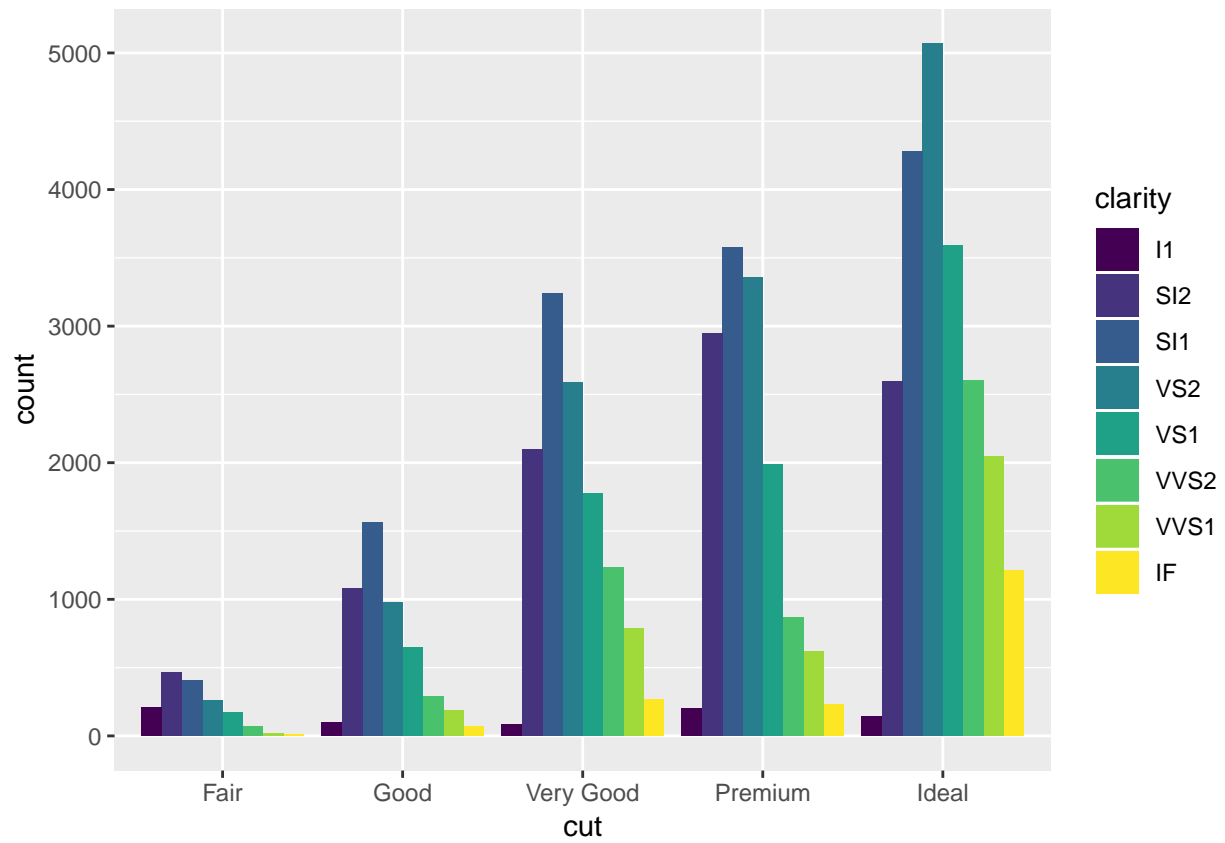


```
ggplot(data=diamonds)+  
  geom_bar(mapping = aes(x=cut, fill=clarity), position="fill")
```

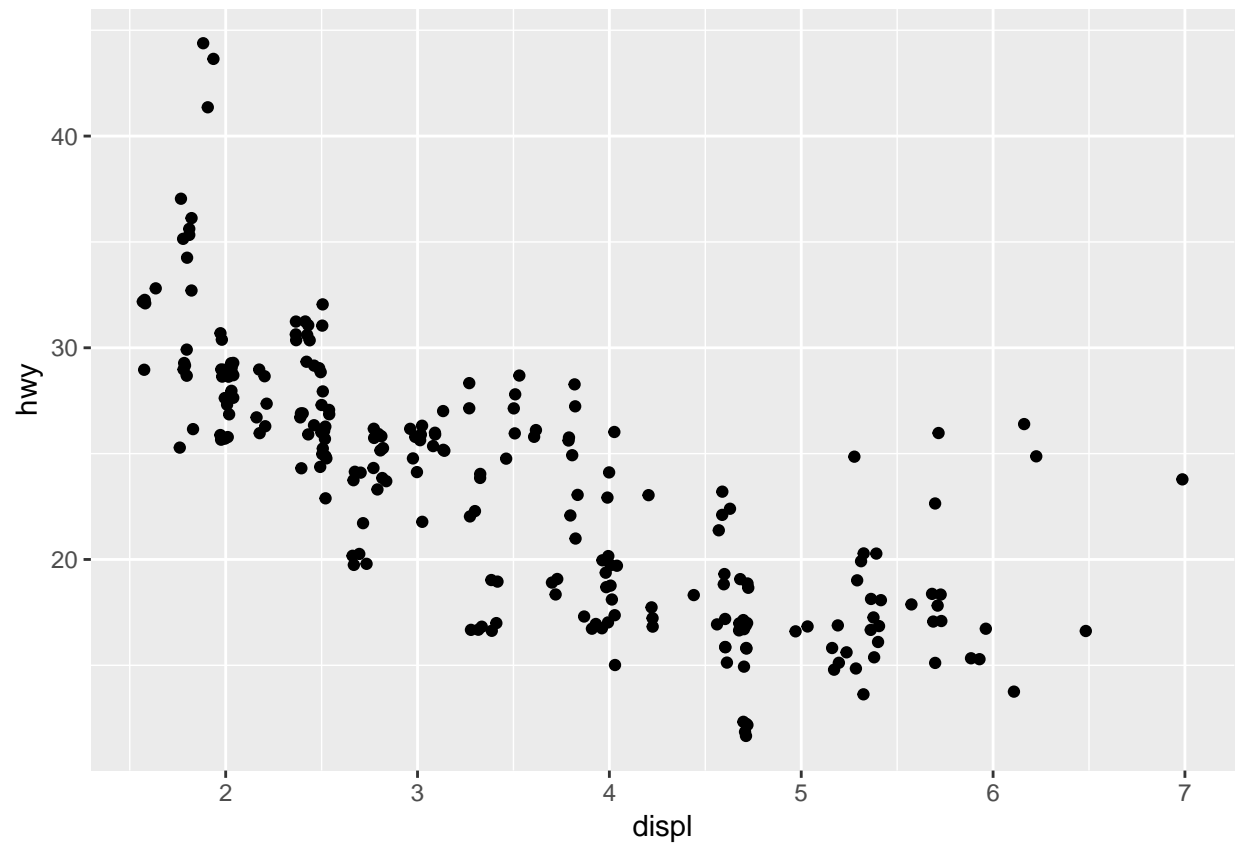




```
ggplot(data=diamonds)+  
  geom_bar(mapping=aes(x=cut, fill=clarity), position="dodge")
```



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

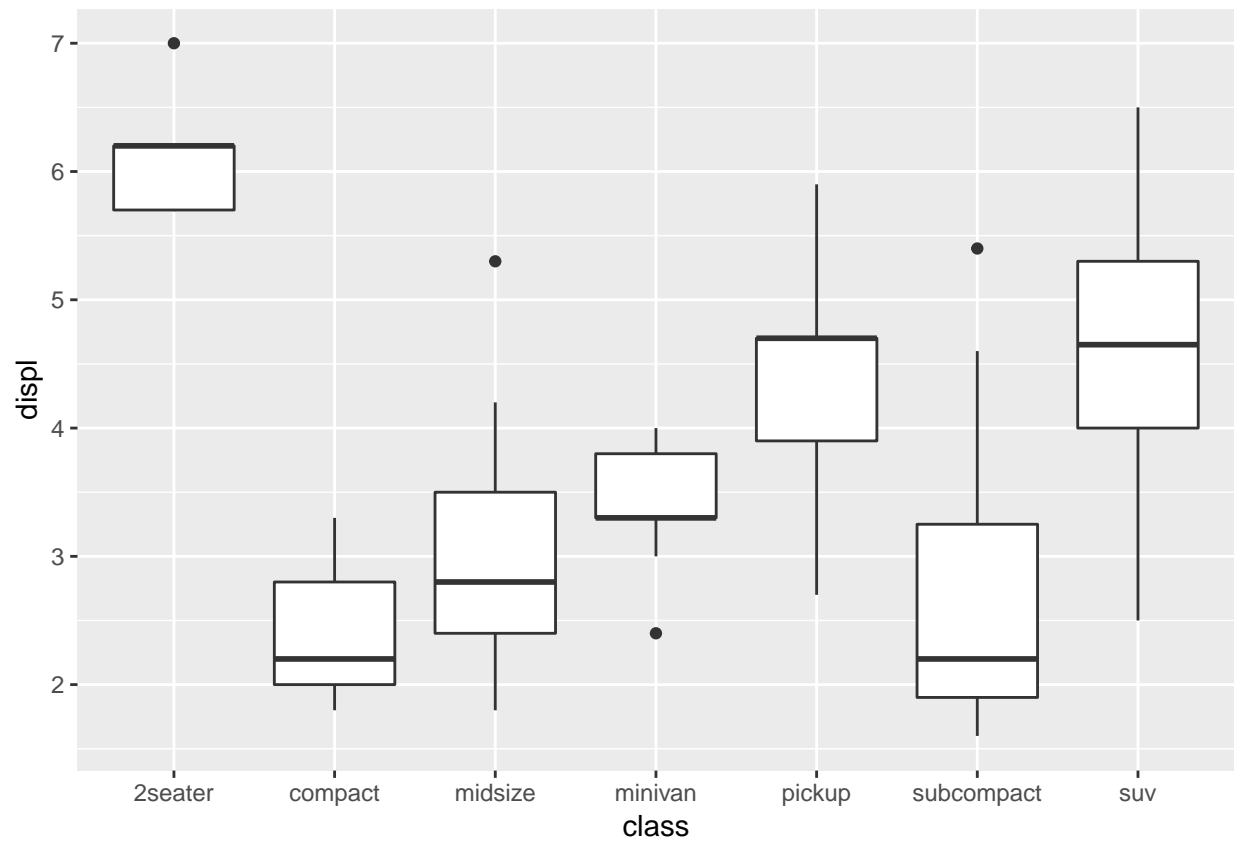


## Sistemas de Coordenadas

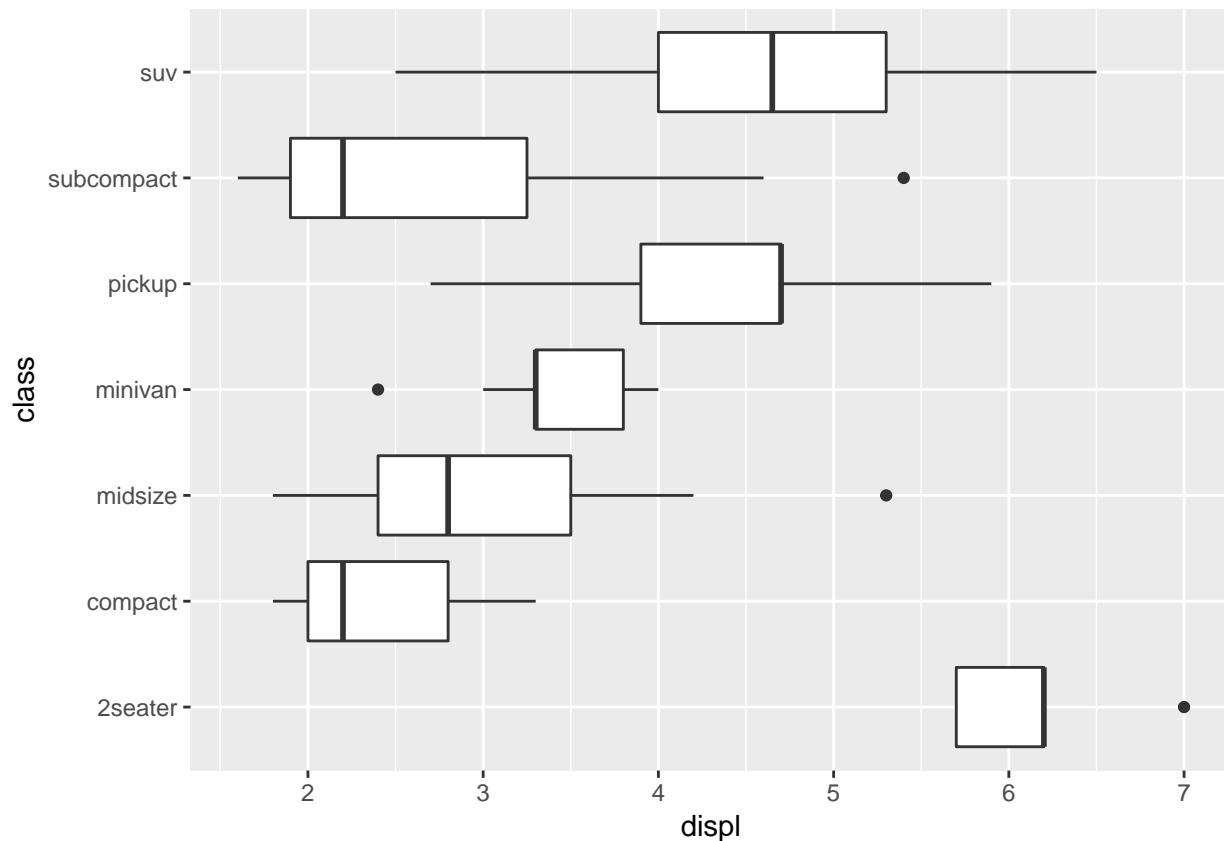
- Permitem ajustar os eixos do gráfico.

Exemplo:

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



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



## dplyr: Transformação e Manipulação de Dados

Funções mais importantes do dplyr:

- **filter()**: filtra observações de acordo com um critério
- **arrange()**: ordena as linhas
- **select()**: seleciona variáveis
- **mutate()**: cria novas variáveis que são funções de outras variáveis
- **summarize()**: resume um grupo de observações
- **group\_by()**: muda o escopo da aplicação de uma função, passando de todo o dataset para grupos definidos.

Recurso bastante útil do dplyr: o **pipe** `%>%`

Operadores de comparação (útil para *filter*):

- `==`
- `<`
- `<=`

- !=
- ==

Operadores lógicos:

- &
- 
- !