

Cap1

```
#VERIFICAR SE O PACOTE ESTÁ CARREGADO
pacotesRequisitados = c("tidyverse")
for(p in pacotesRequisitados){
  if(!require(
    p, character.only = TRUE
  ))
    install.packages(p)
  library(p,character.only = TRUE)
}
```

Carregando pacotes exigidos: tidyverse

-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --

v dplyr 1.1.4 v readr 2.1.4

v forcats 1.0.0 v stringr 1.5.1

v ggplot2 3.4.4 v tibble 3.2.1

v lubridate 1.9.3 v tidyr 1.3.0

v purrr 1.0.2

-- 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

Carregamento dos Dados.

mpg

A tibble: 234 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~

7 audi a4 3.1 2008 6 auto~ f 18 27 p comp~

8 audi a4 quattro 1.8 1999 4 manu~ 4 18 26 p comp~

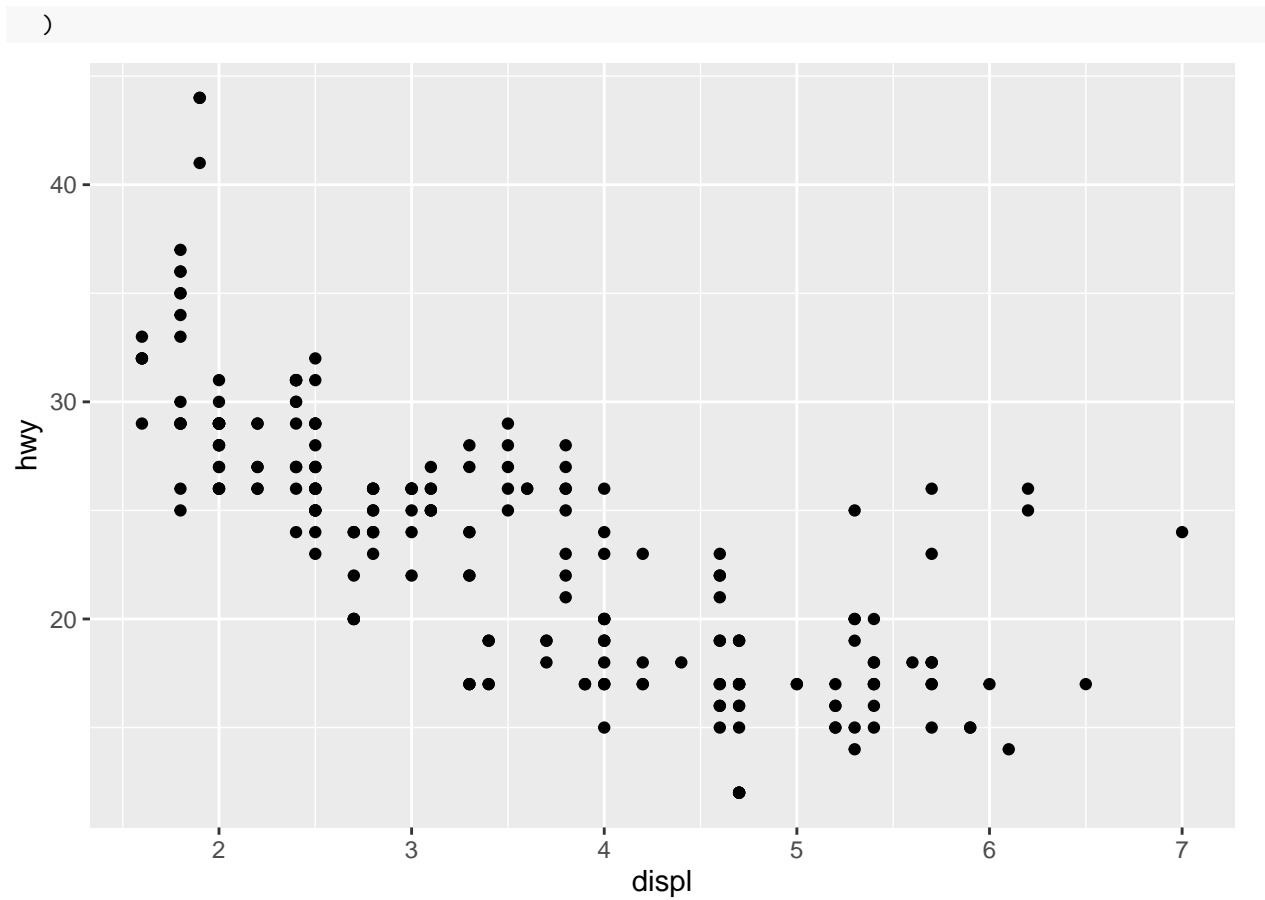
9 audi a4 quattro 1.8 1999 4 auto~ 4 16 25 p comp~

10 audi a4 quattro 2 2008 4 manu~ 4 20 28 p comp~

i 224 more rows

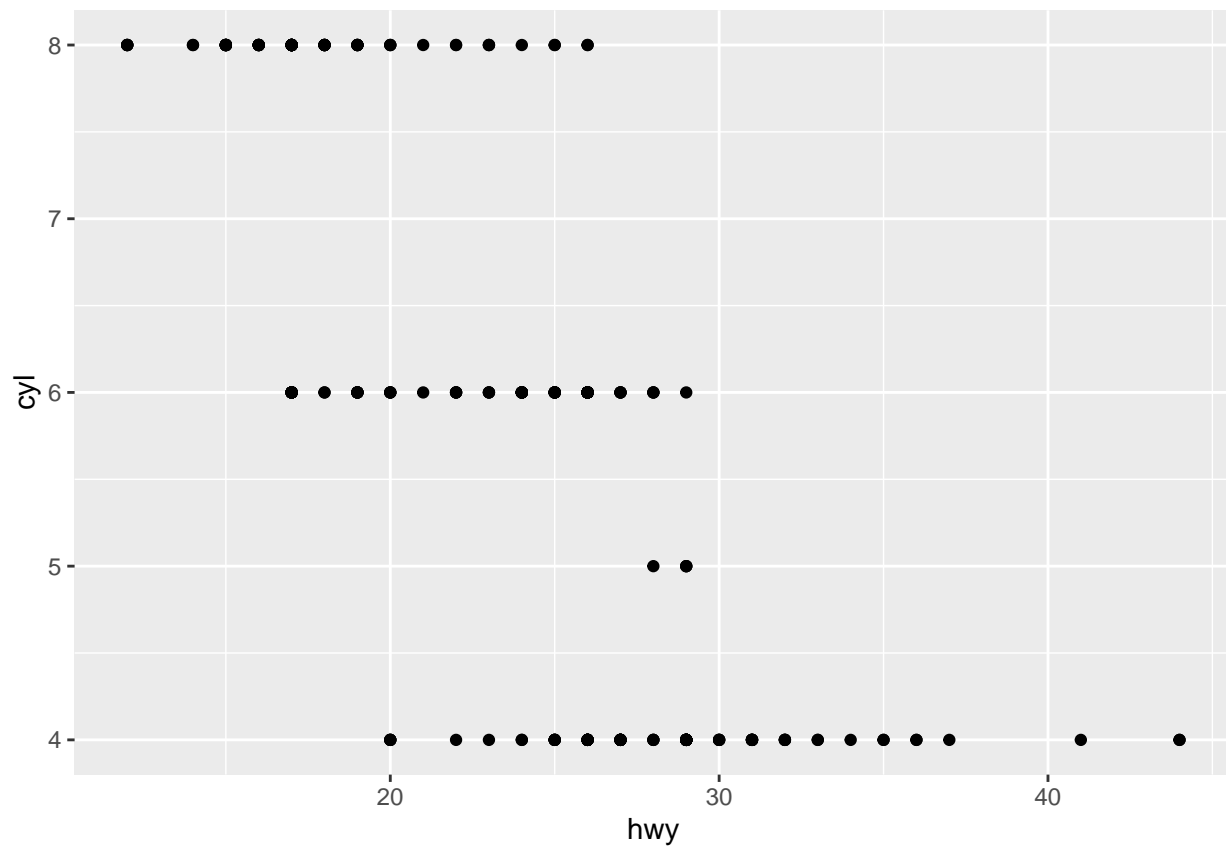
Criação de um ggplot.

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



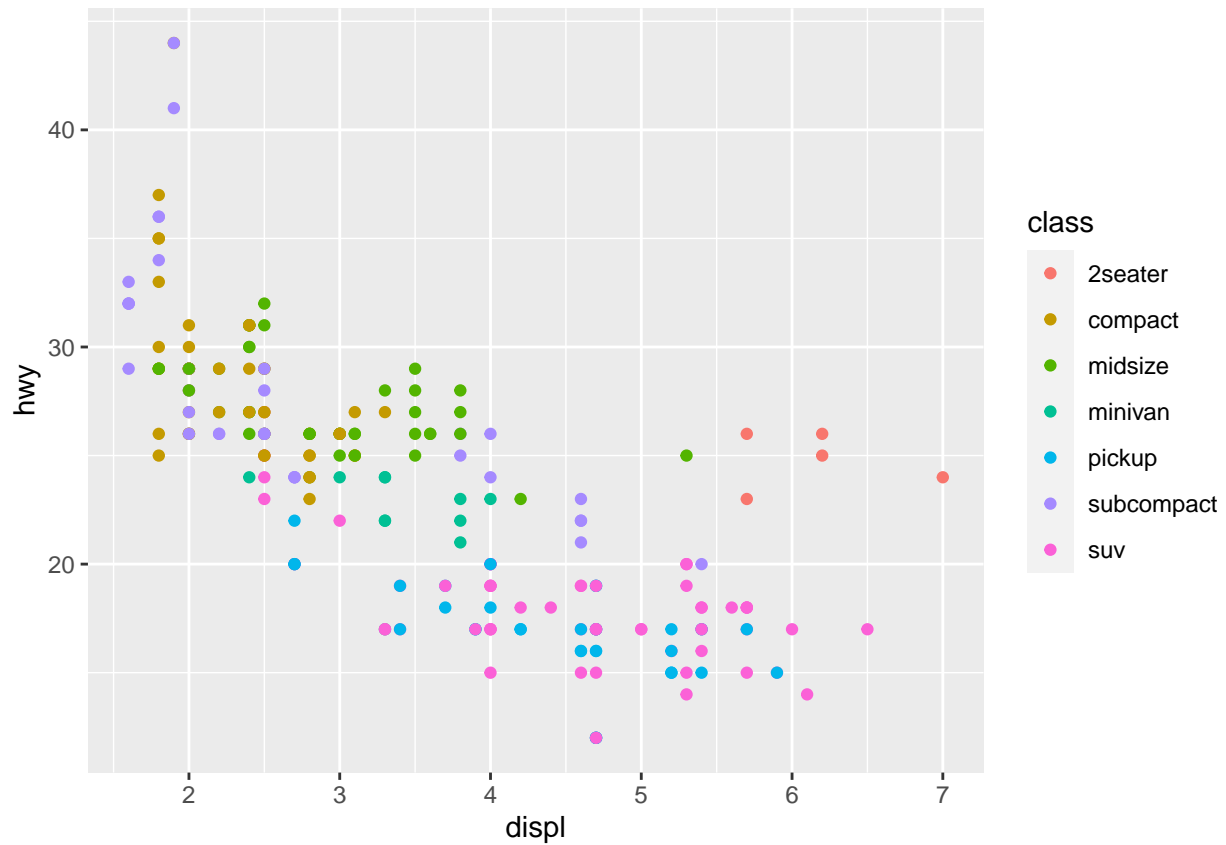
Pagina 6 - Exercício 4

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



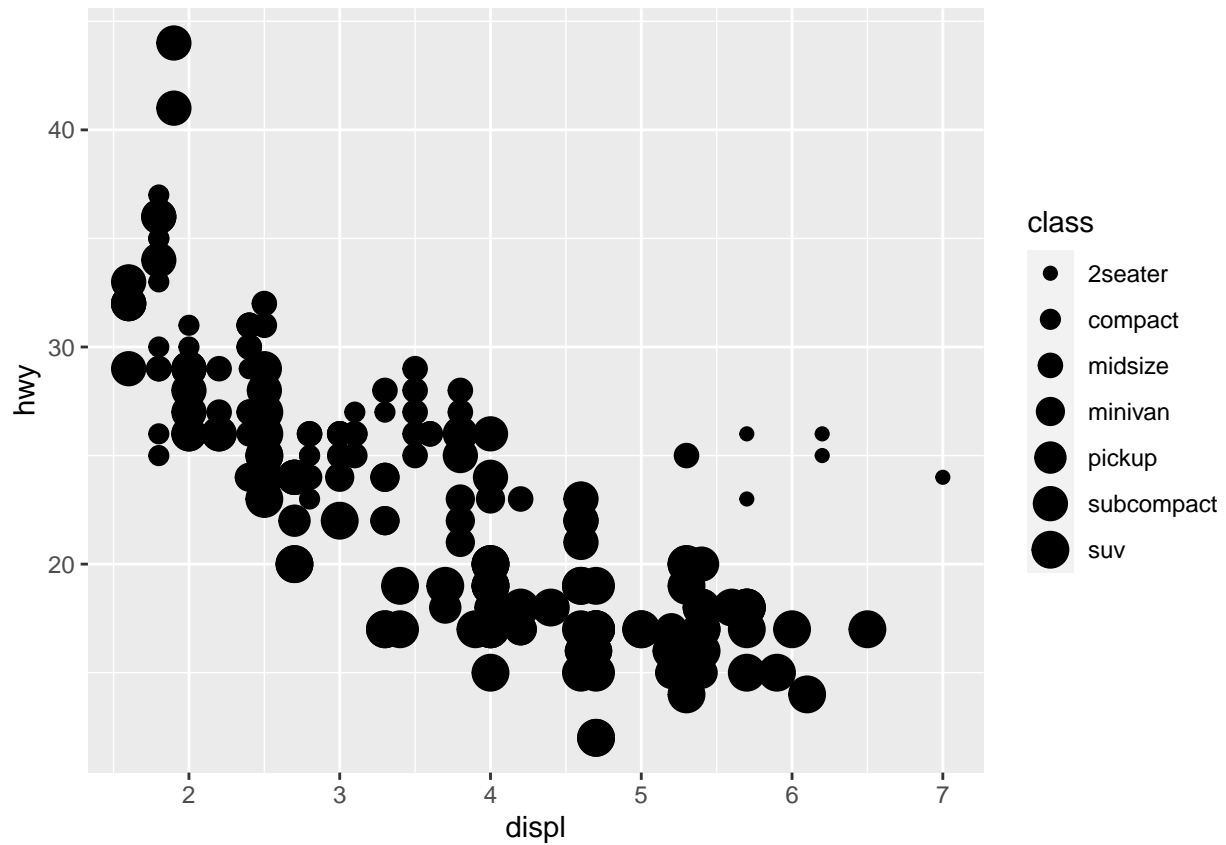
Pagina 8

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



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

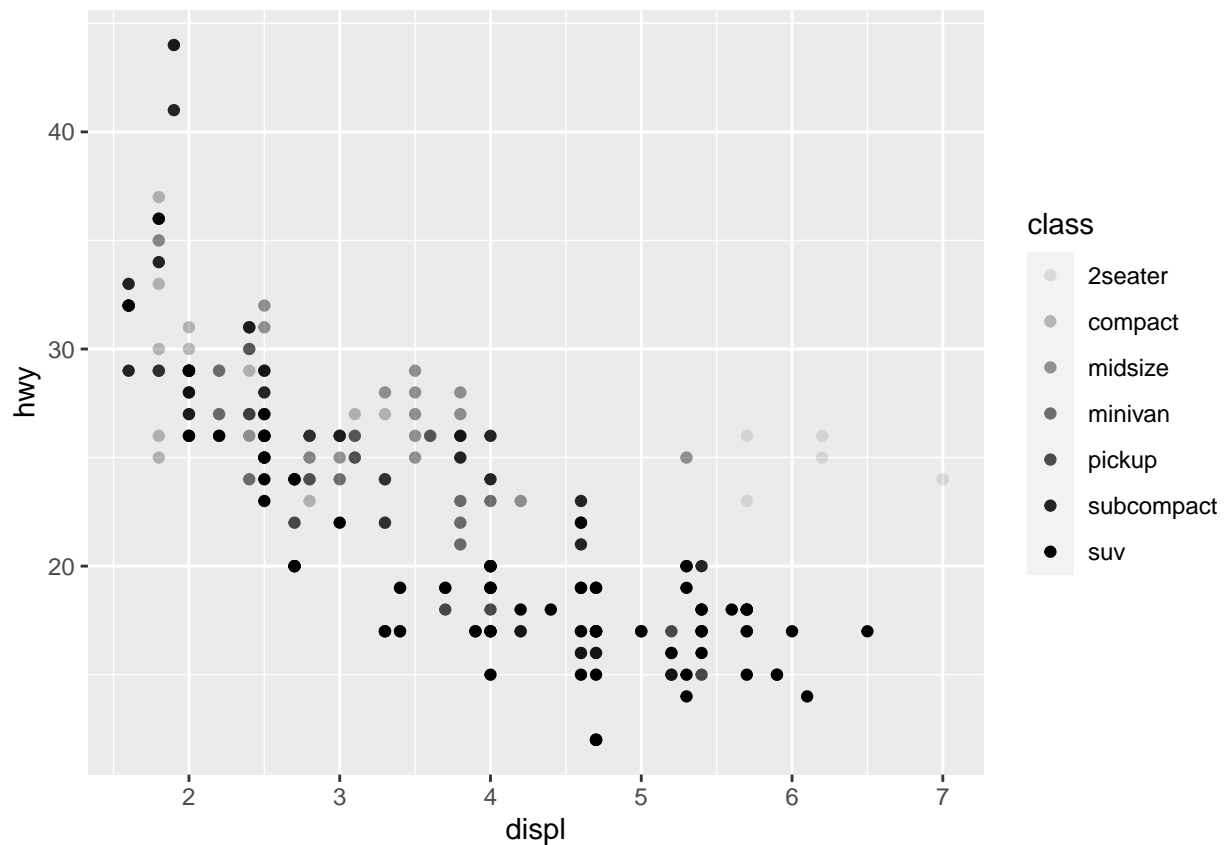
Warning: Using size for a discrete variable is not advised.



Top

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

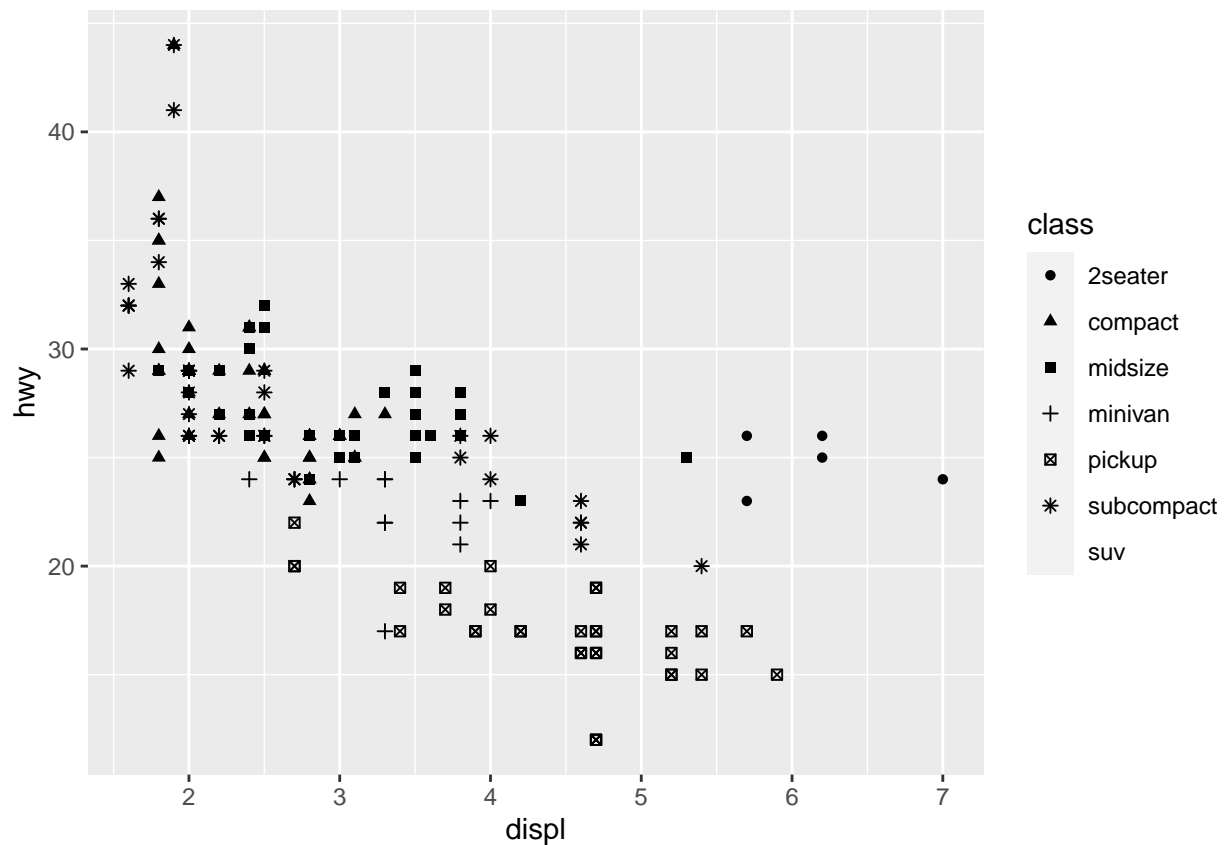
Warning: Using alpha for a discrete variable is not advised.



Bottom

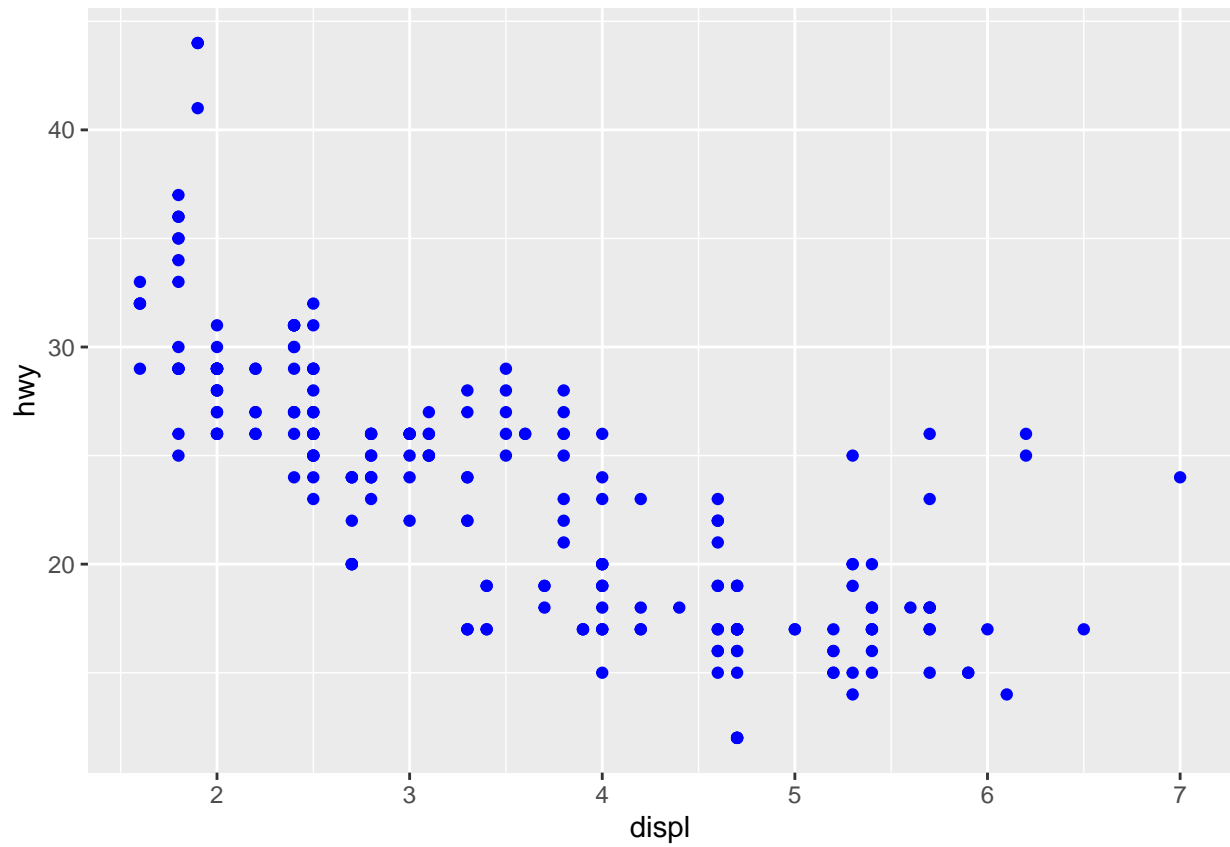
```
ggplot(data=mpg)+
  geom_point(
    mapping = aes(
      x=displ, y=hwy, shape=class #utilização de 6 classes, se houver mais categorias fica fora
    )
  )
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
## Warning: Removed 62 rows containing missing values (`geom_point()`).
```



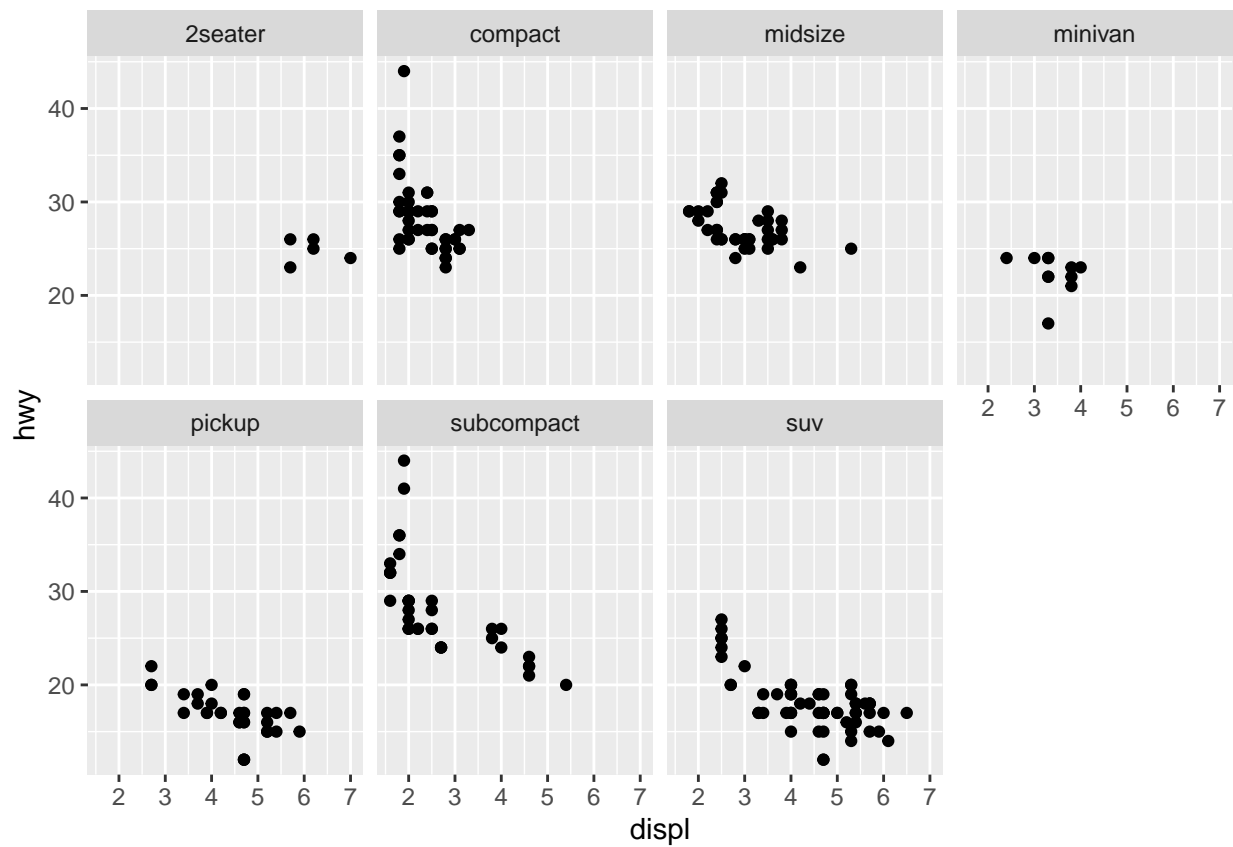
Podemos configurar as propriedades manualmente deixando todos os pontos azuis por exemplo

```
ggplot(
  data=mpg
) +
  geom_point(
    mapping = aes(
      x=displ, y=hwy
    ), color="blue" # a cor não transmite a informação sobre uma variável
  )
```



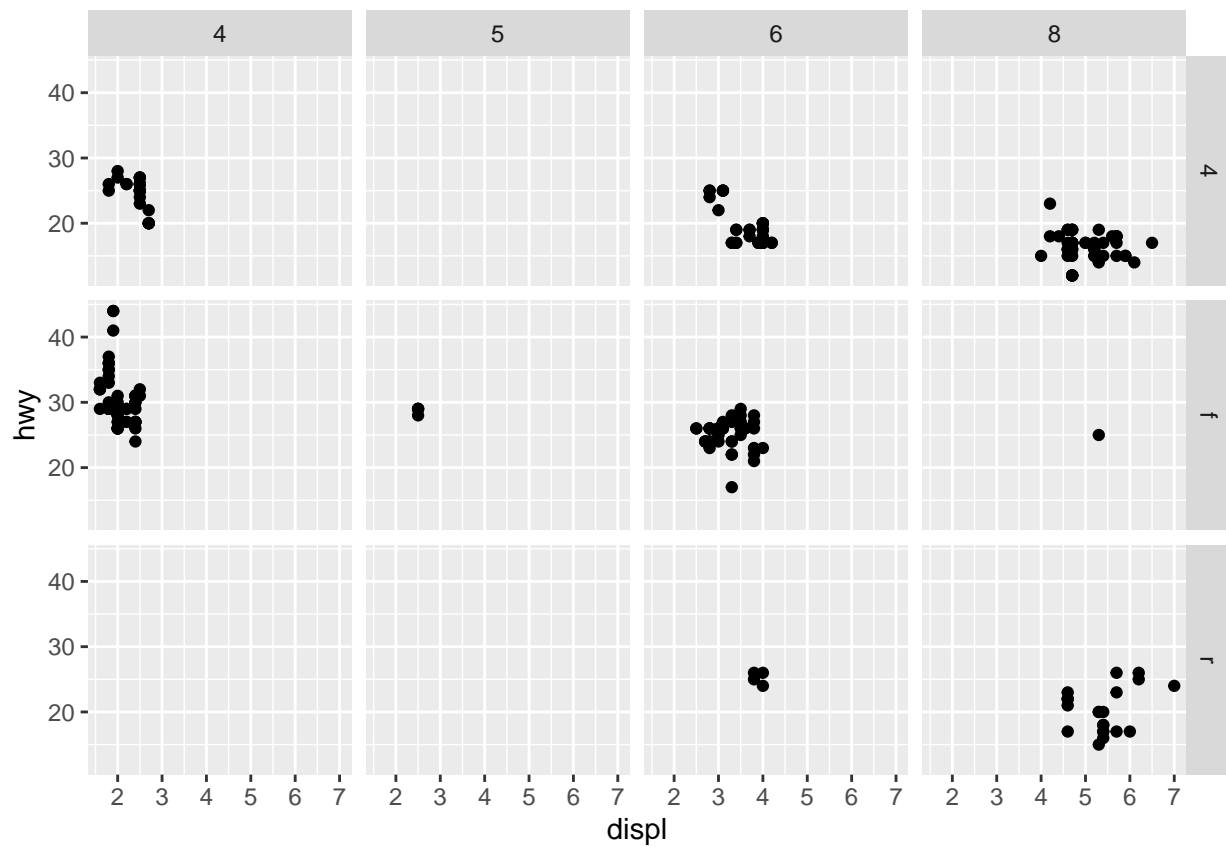
Facetas

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

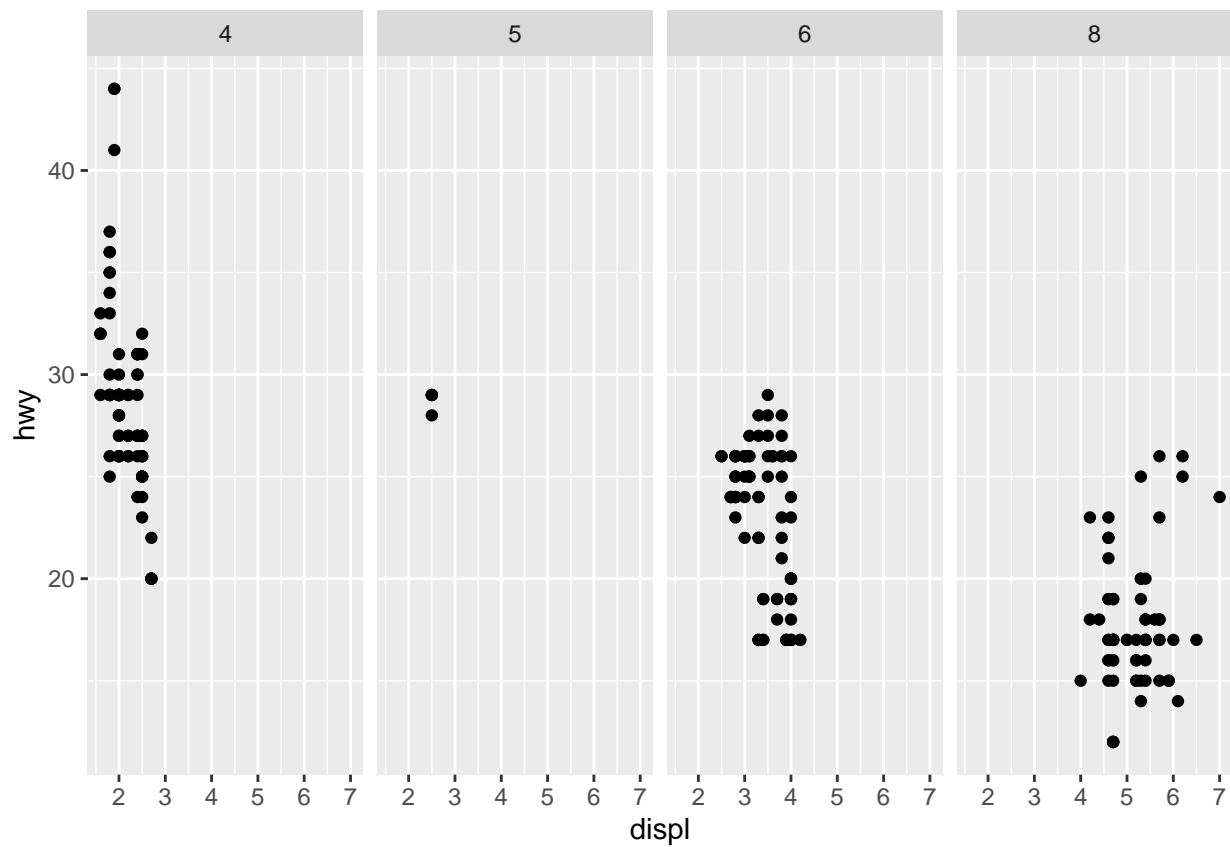



Combinação com mais variáveis

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

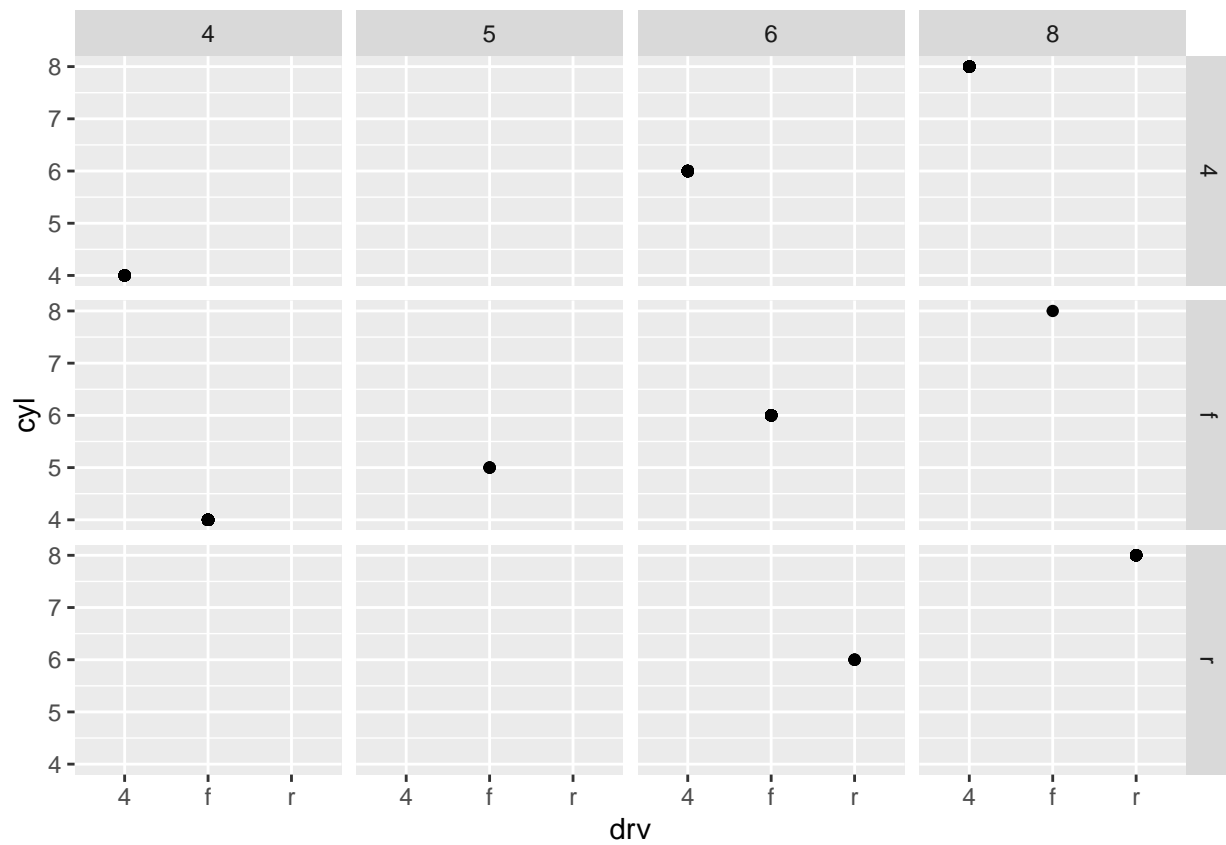


```
ggplot( data = mpg ) +
  geom_point(
    mapping = aes(
      x = displ, y = hwy
    )
  ) +
  facet_grid( . ~ cyl)
```



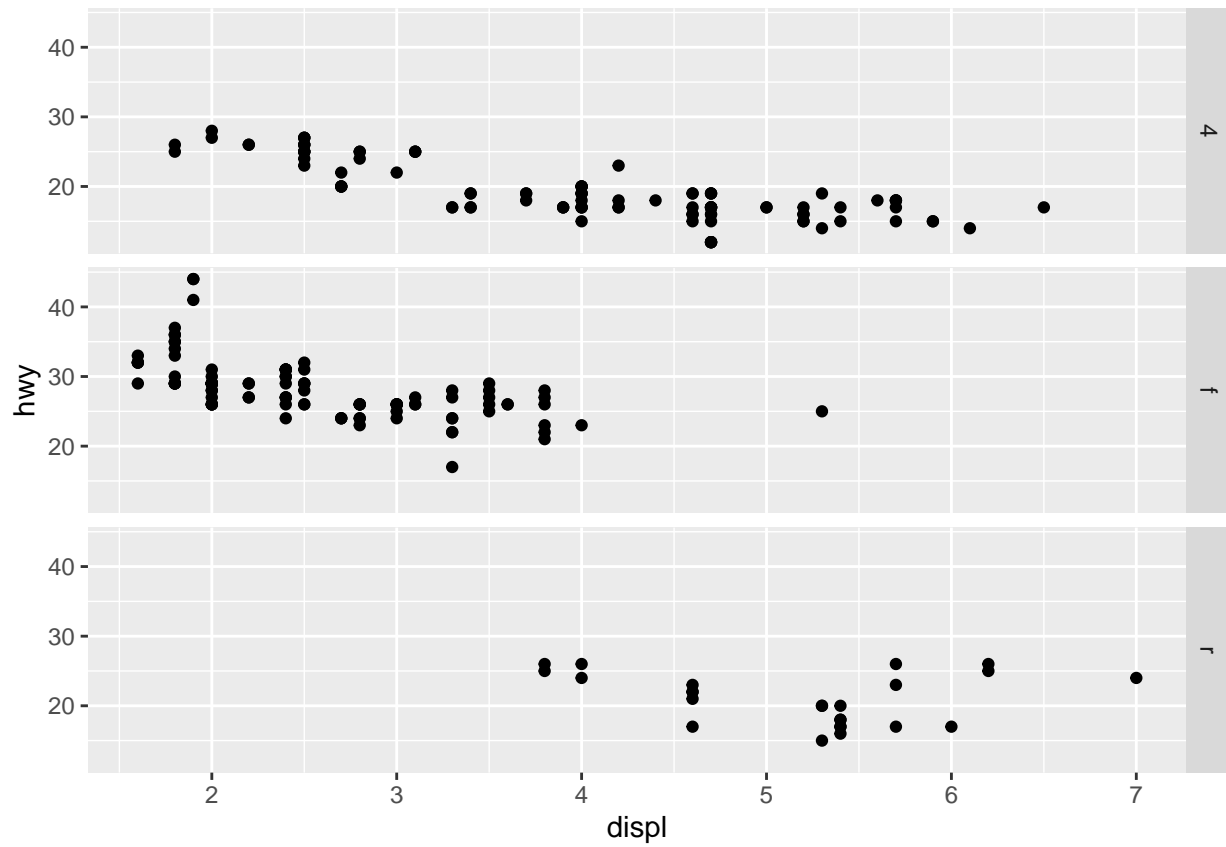
Exercicio 2

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

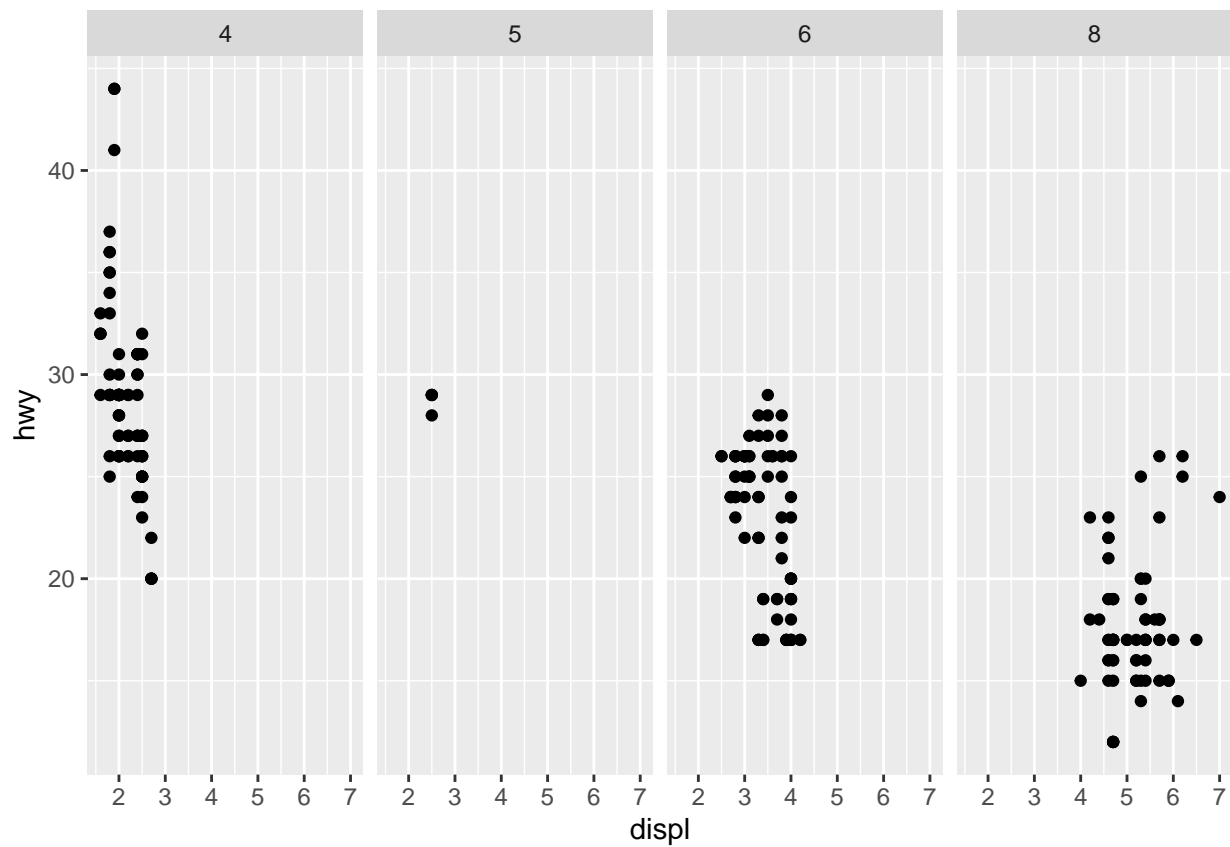


Exercício 3

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



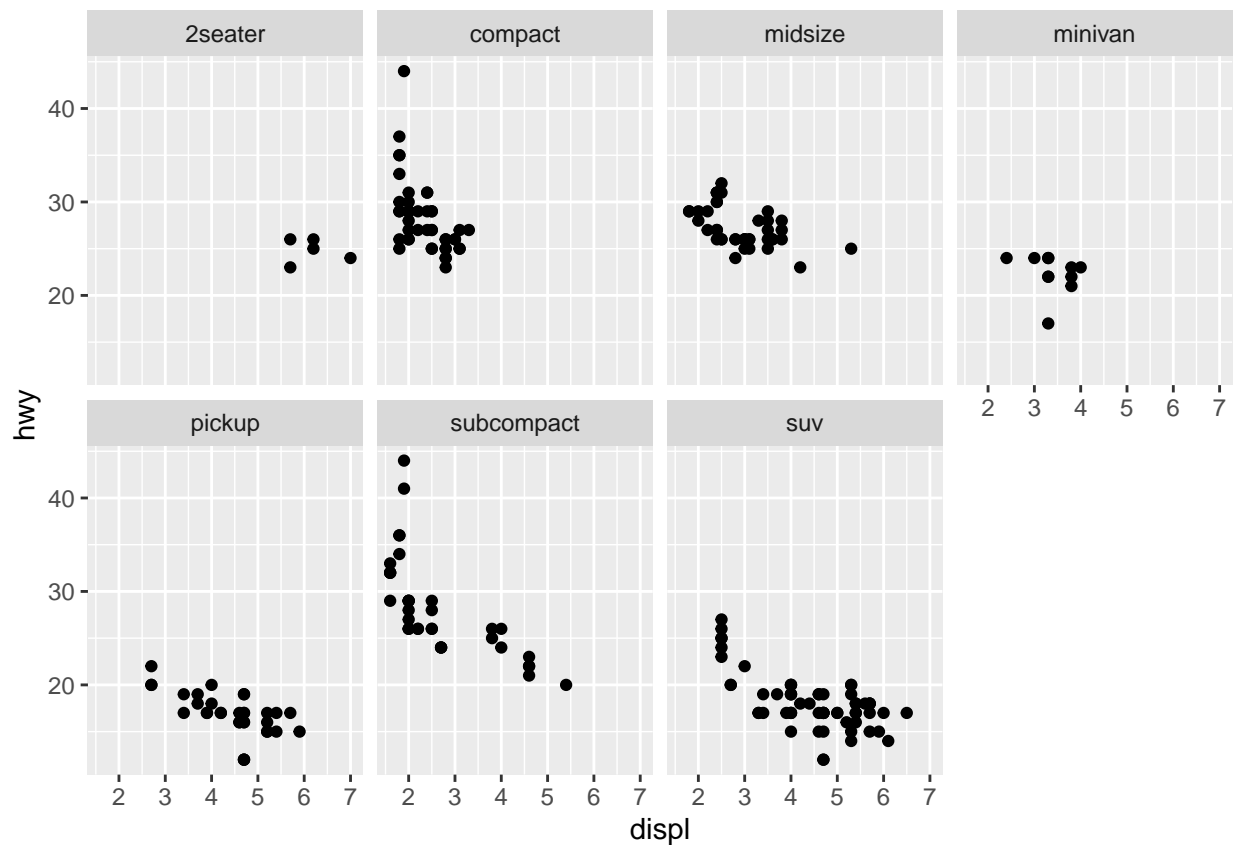
```
ggplot( data = mpg ) +
  geom_point(
    mapping = aes(
      x = displ, y = hwy
    )
  ) +
  facet_grid(
    . ~ cyl
  )
```



Fazendo de acordo com a variável.

Exercicio 4

```
ggplot(data = mpg) +
  geom_point(
    mapping = aes(
      x = displ, y = hwy
    )
  ) +
  facet_wrap( ~ class, ncol = 4)
```



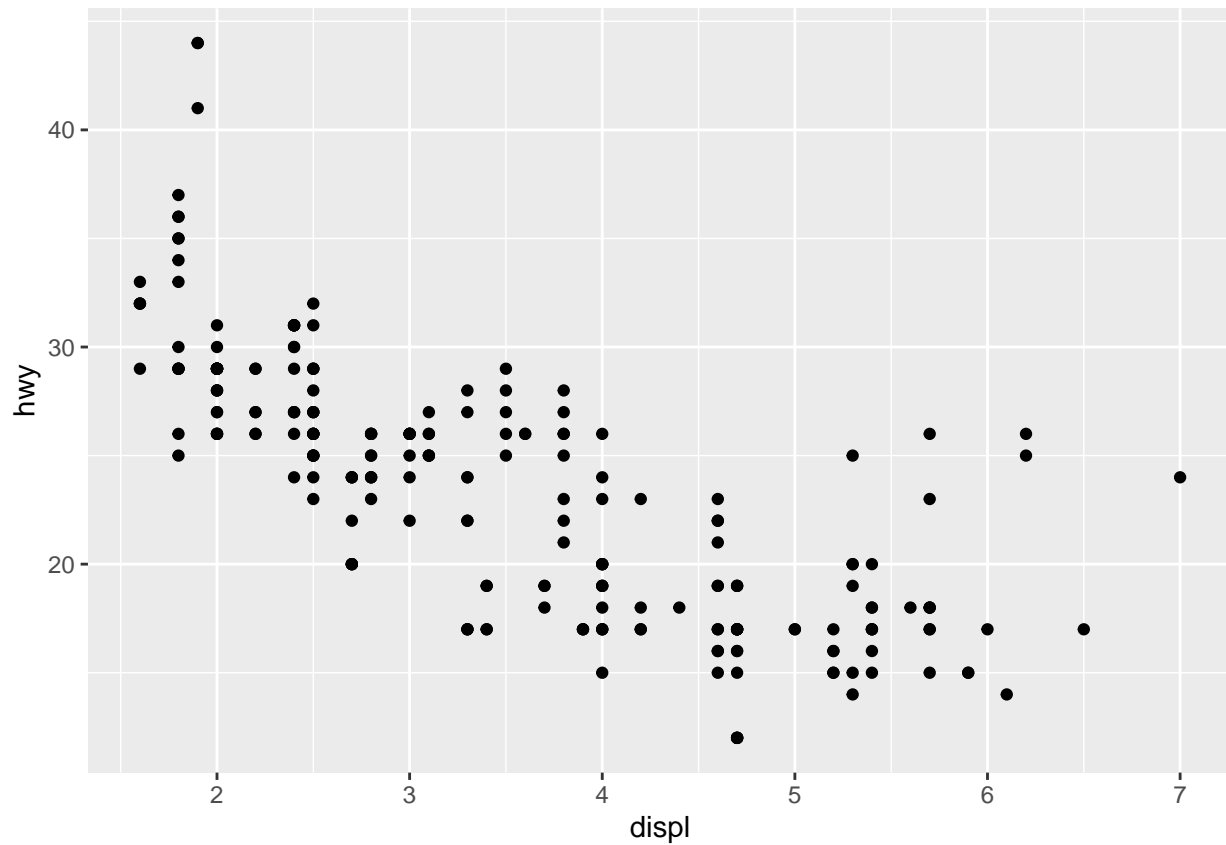
As variáveis ficam isoladas, cada gráfico para cada variável, ao invés de um gráfico para muitas variáveis.

Exercício 5 nrow dividi-se em linhas ncol dividi-se em colunas facet_grid não tem variáveis nrow e ncol pois são de acordo com variáveis as quais você insere de acordo

Objetos geométricos pag 16

left

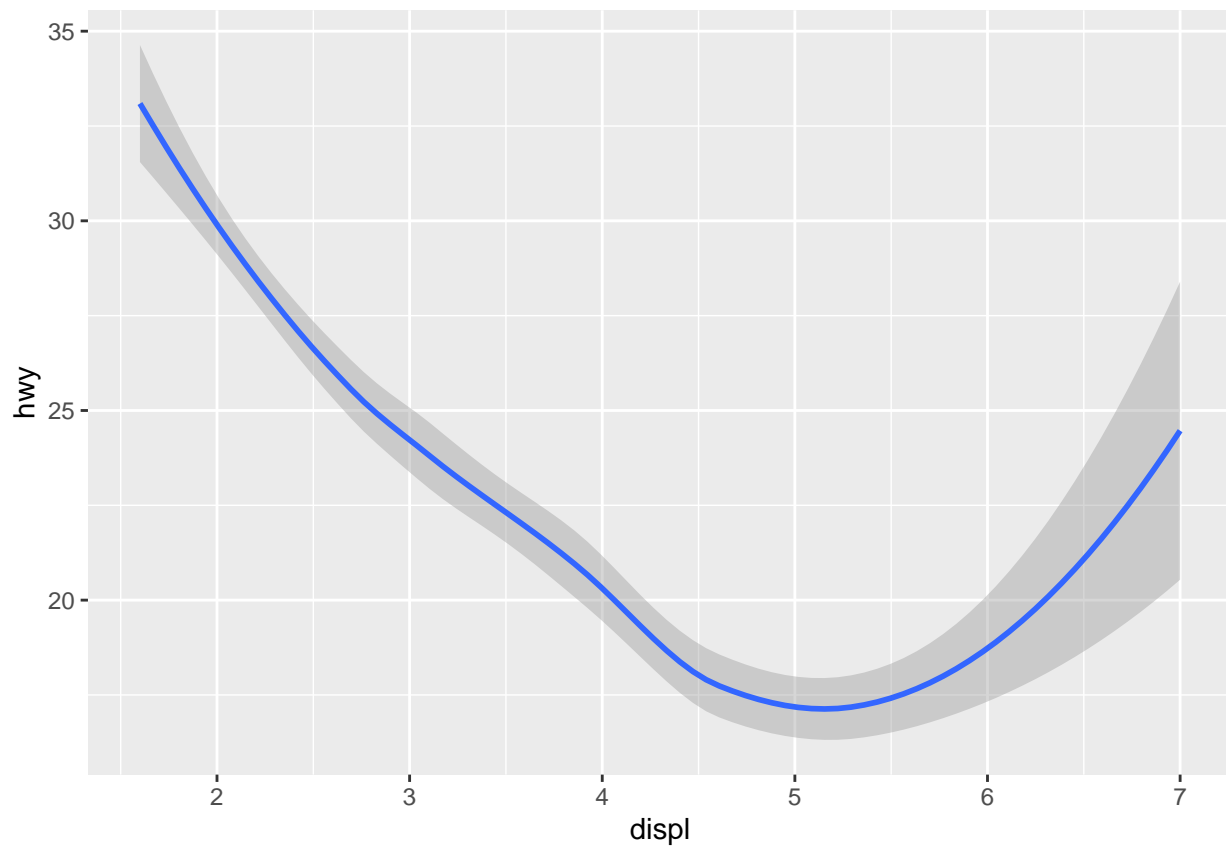
```
ggplot( data = mpg ) +
  geom_point(
    mapping = aes(
      x = displ, y = hwy
    )
  )
```



right

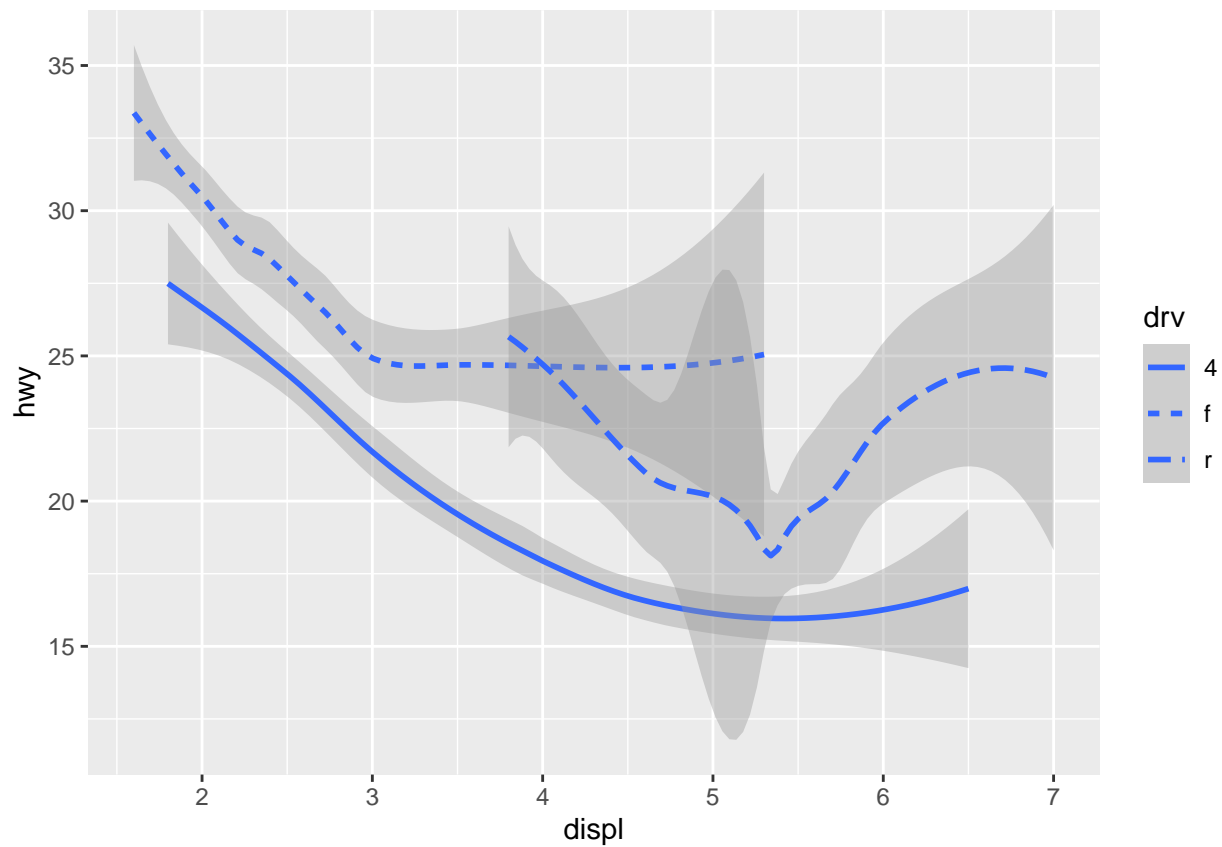
```
ggplot( data = mpg ) +  
  geom_smooth(  
    mapping = aes(  
      x = displ, y = hwy  
    )  
  )  
)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

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

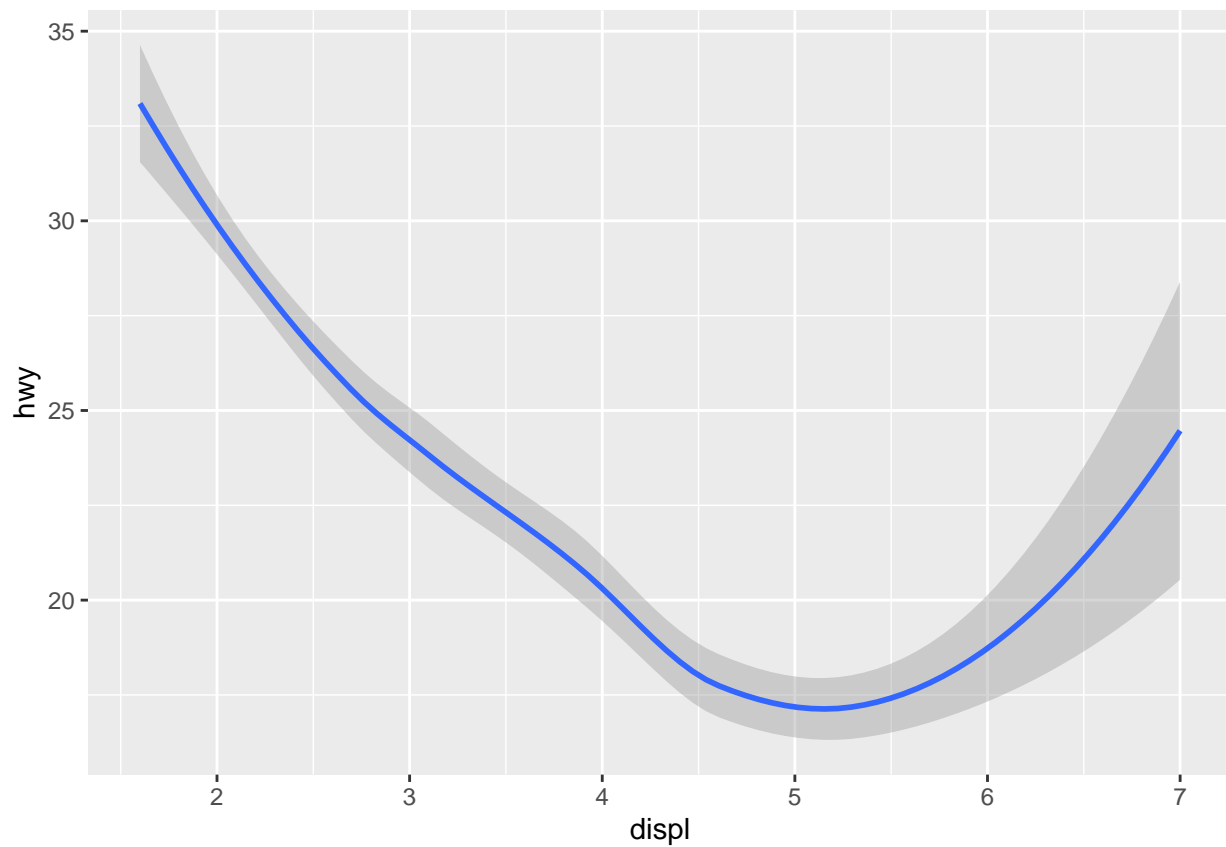
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



4 quer dizer tração nas quatro rodas f tração dianteira r tração traseira pag 18

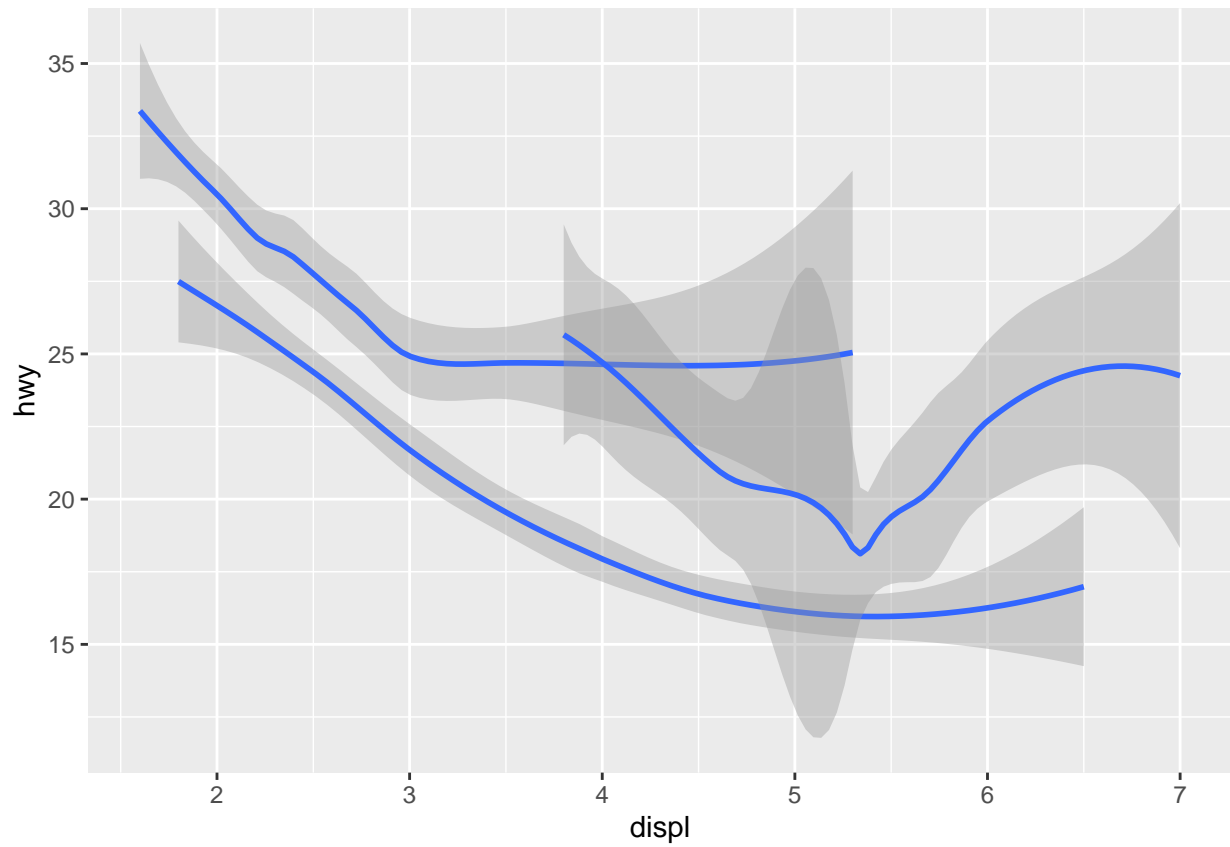
```
ggplot(data = mpg) +
  geom_smooth(
    mapping = aes(
      x = displ, y = hwy
    )
  )
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



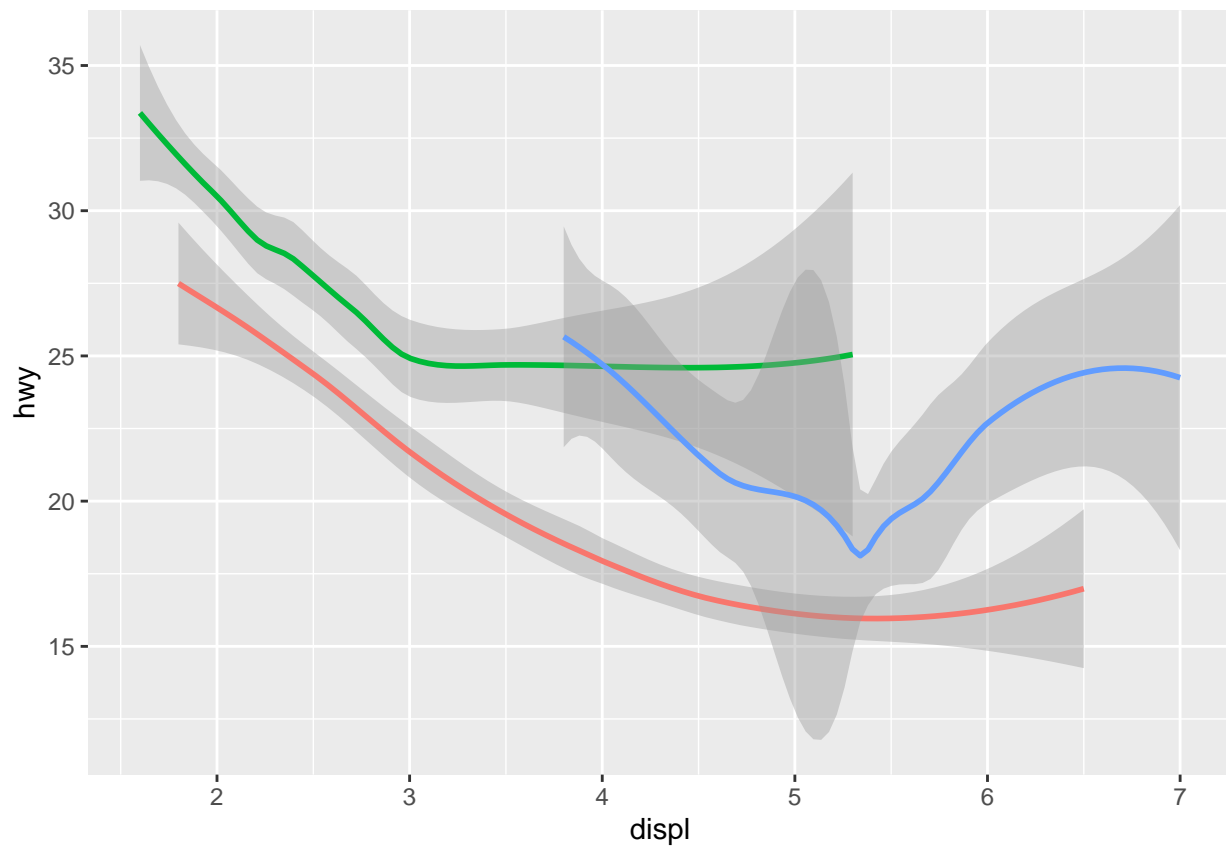
```
ggplot( data = mpg ) +  
  geom_smooth(  
    mapping = aes(  
      x = displ, y = hwy, group = drv  
    )  
  )  
)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
ggplot( data = mpg) +  
  geom_smooth(  
    mapping = aes(  
      x = displ, y = hwy, color = drv  
    ),  
    show.legend = FALSE  
  )
```

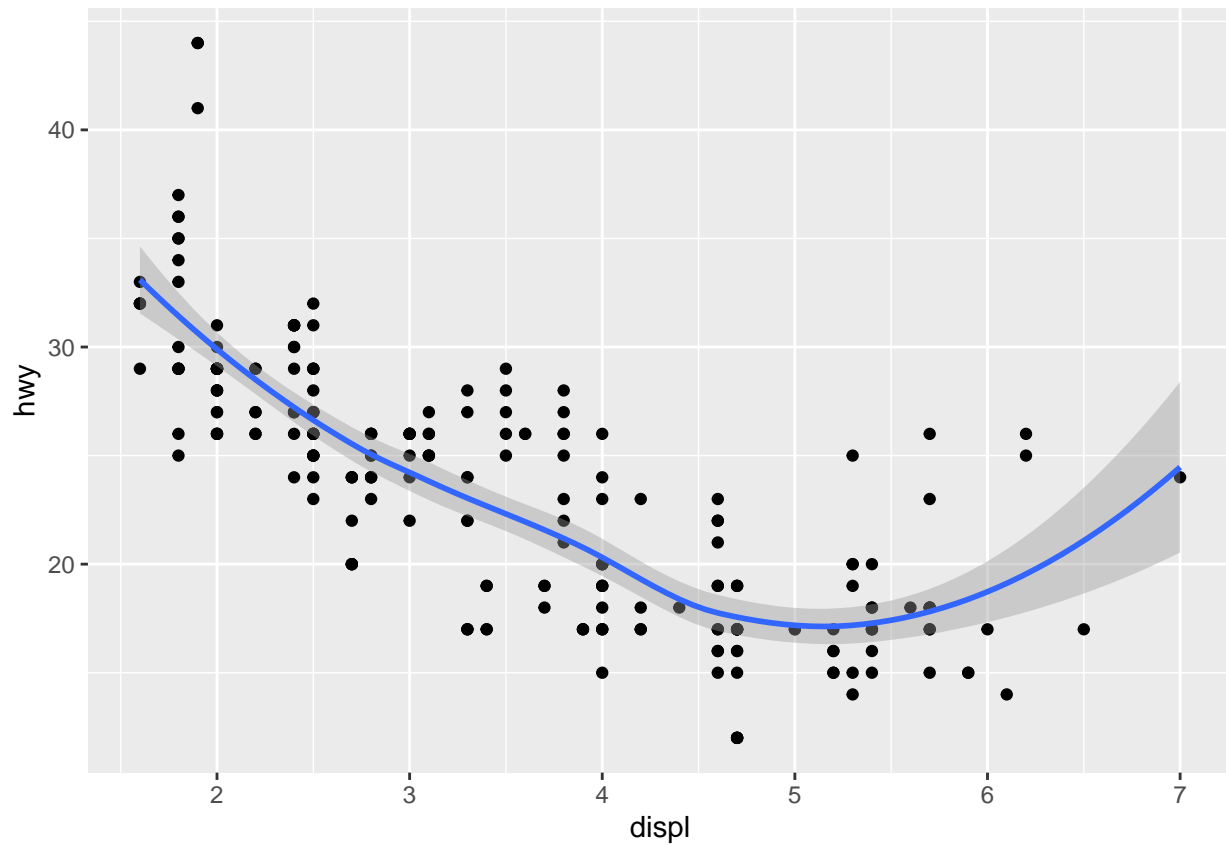
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Para exibir vários geoms no mesmo gráfico

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

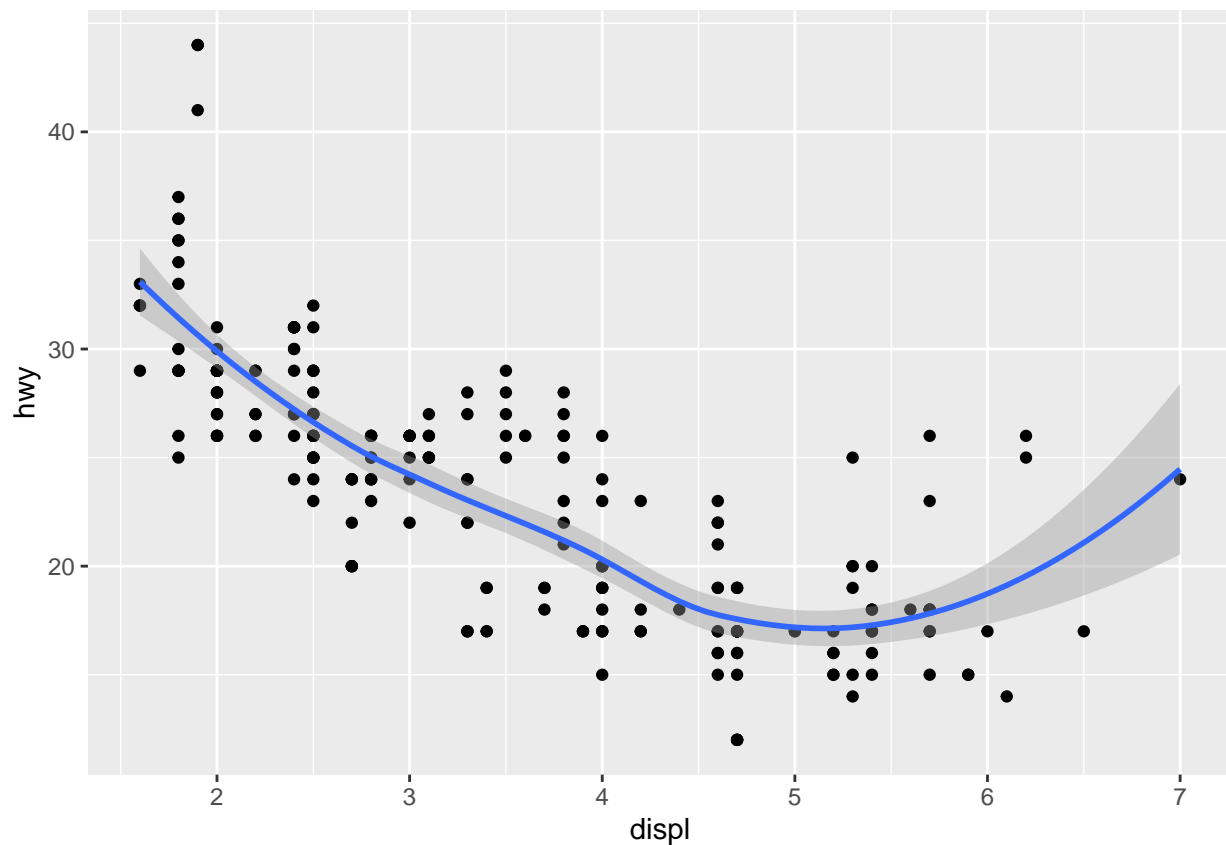
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



duplicação de código, pois estou plotando os mesmo dados em dois tipos de gráficos diferentes péssimo para manutenção do código.. o mais correto seria.

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

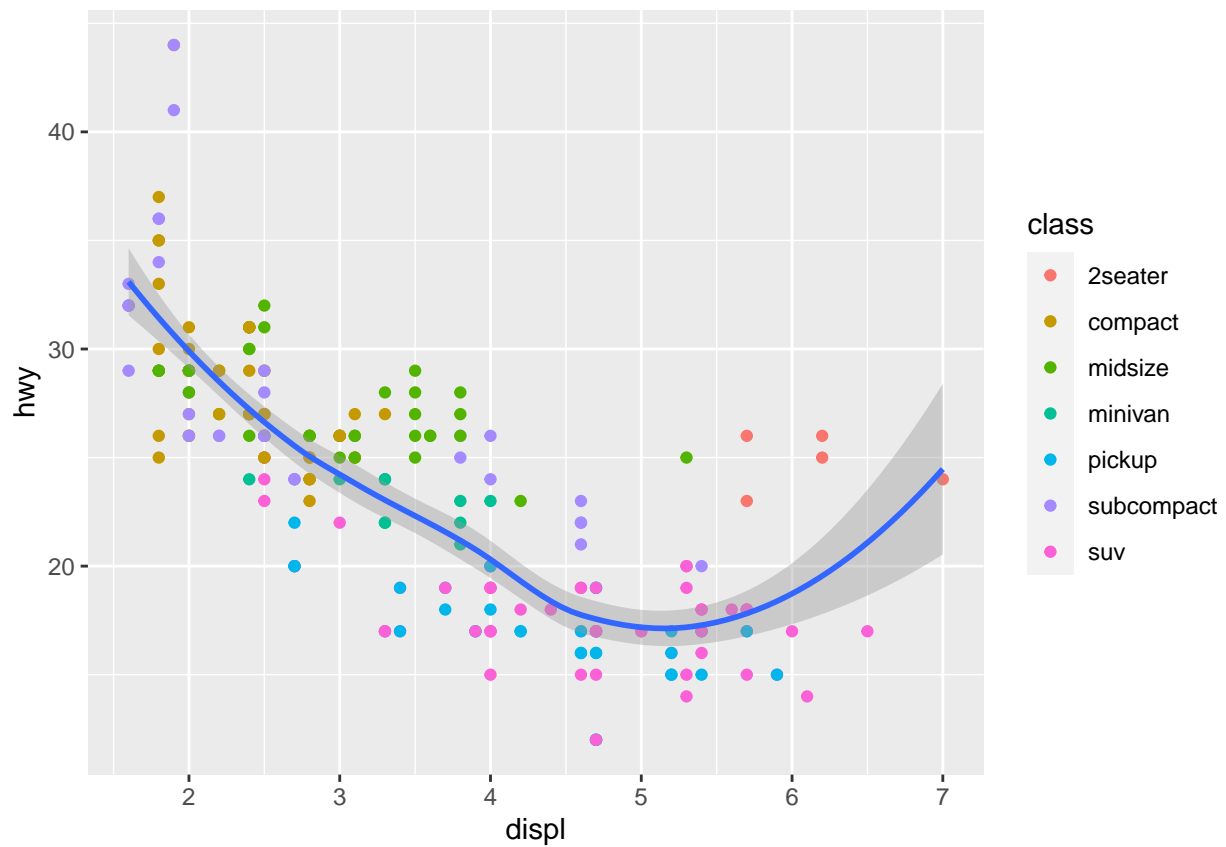
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



para limpeza do código, é o mais correto..podemos modificar umas das propriedades

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

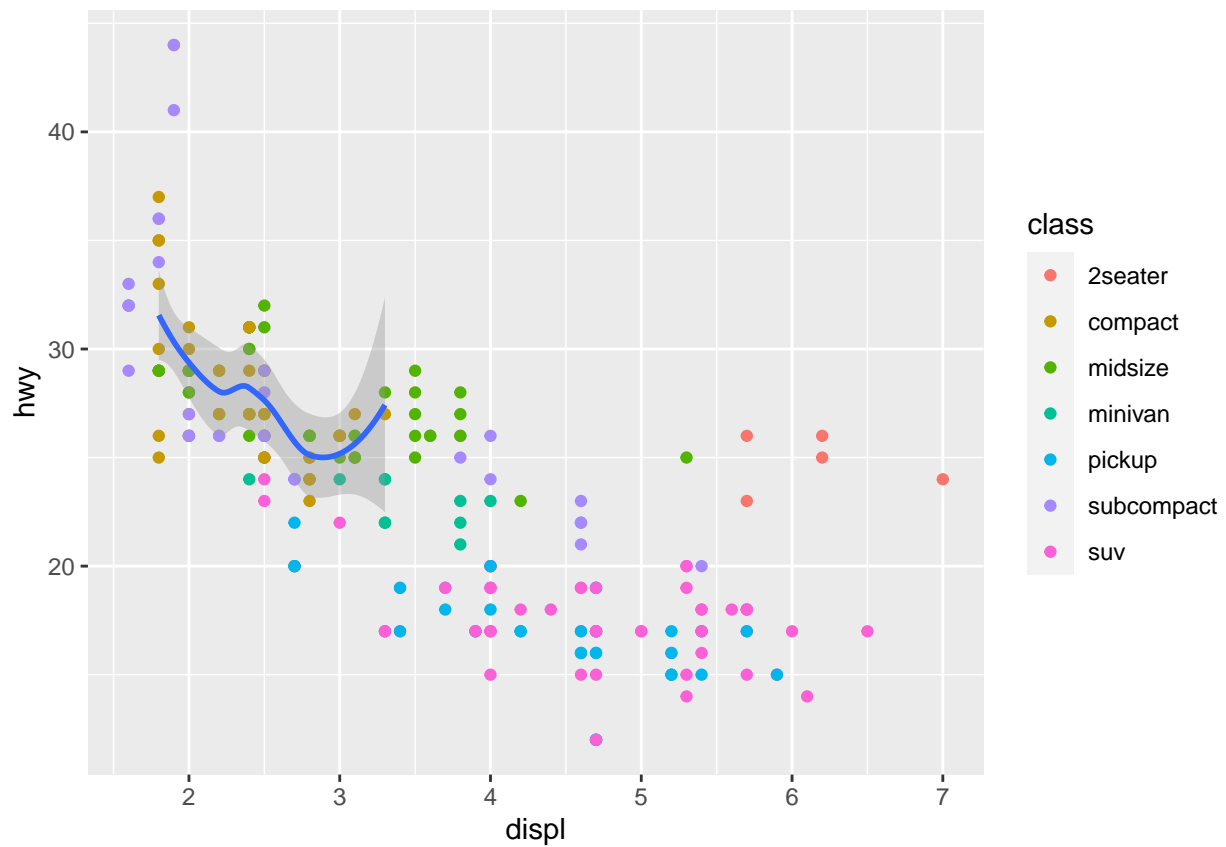
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



pag 20 usando filter

```
ggplot(
  data = mpg, mapping = aes(
    x = displ, y = hwy
  )
) +
  geom_point(
    mapping = aes(
      color = class
    )
  ) +
  geom_smooth(
    data = filter(
      mpg, class == "compact"
    ), se = TRUE
  )
```

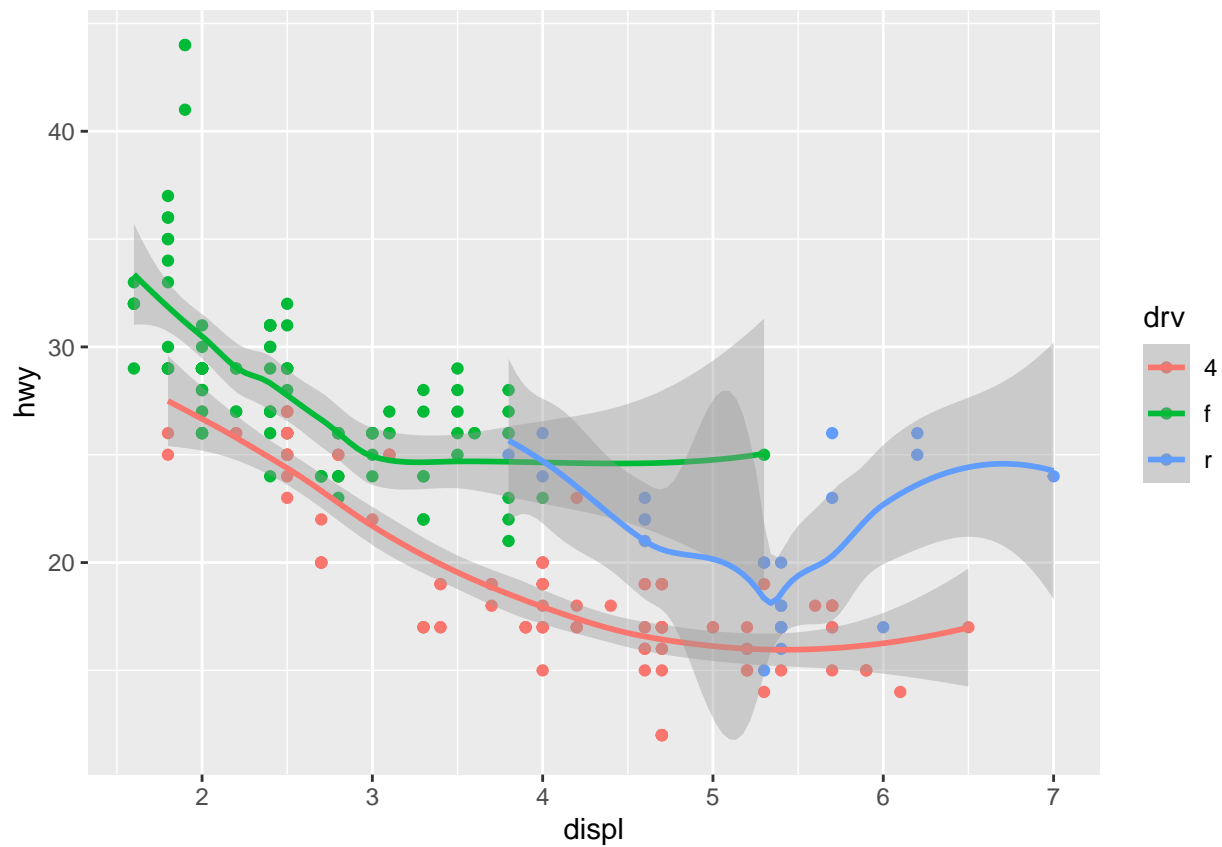
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'



exercício 2

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

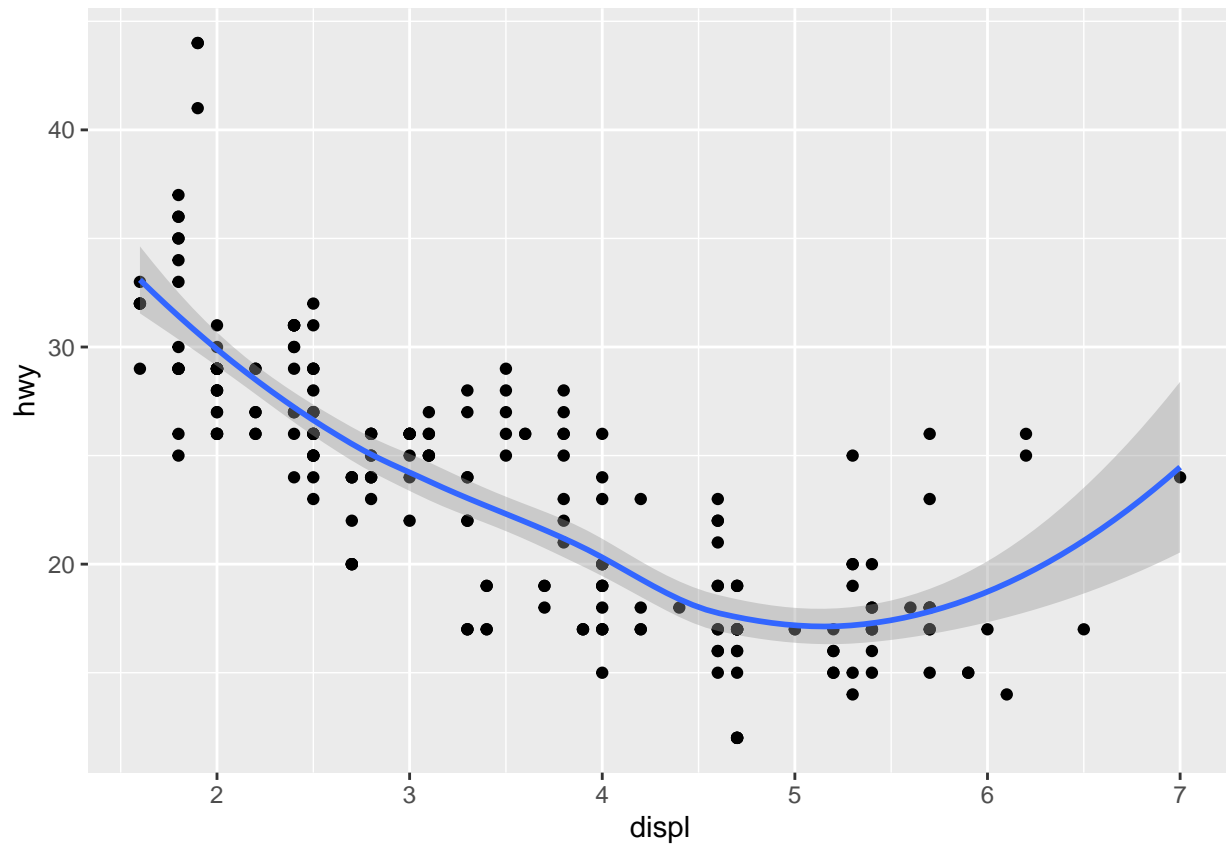
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'



exercício 5

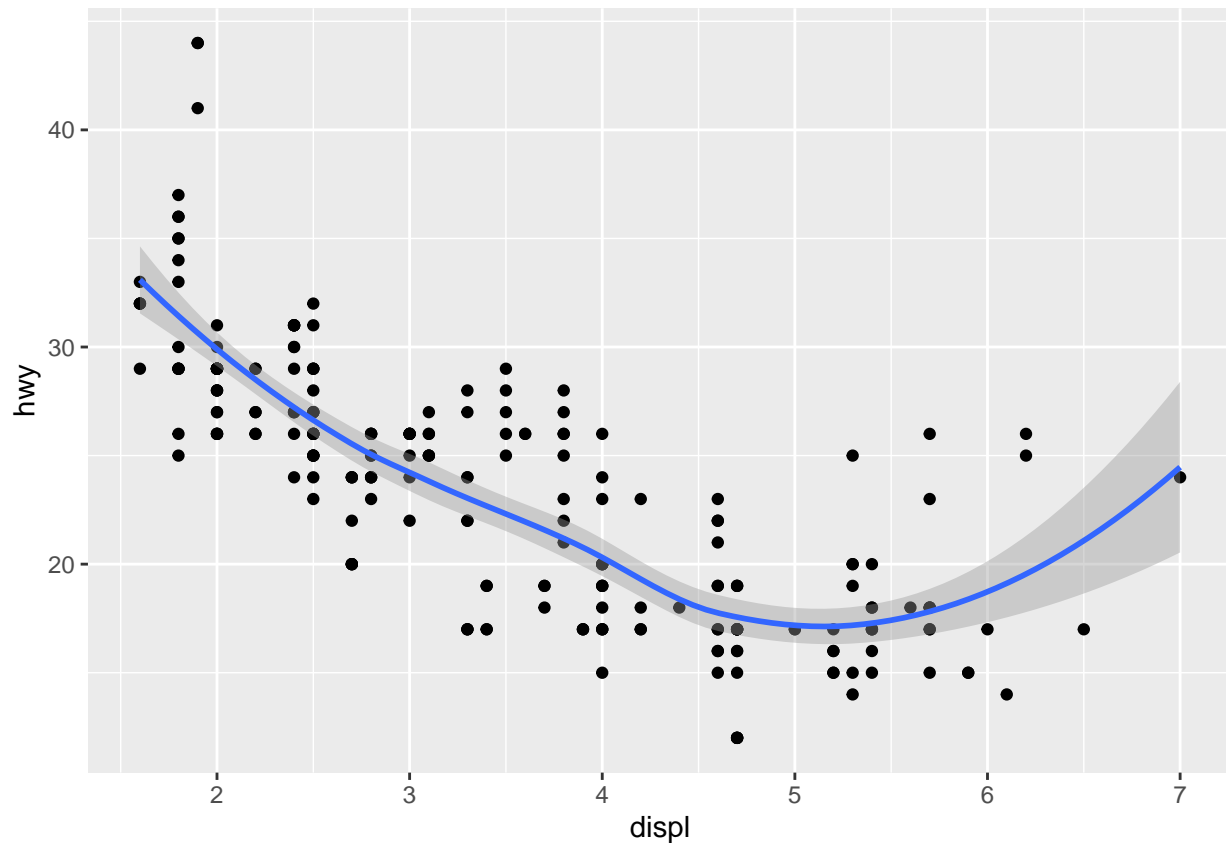
```
ggplot(
  data = mpg, mapping = aes(
    x = displ, y = hwy
  )) +
  geom_point() +
  geom_smooth()

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



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

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



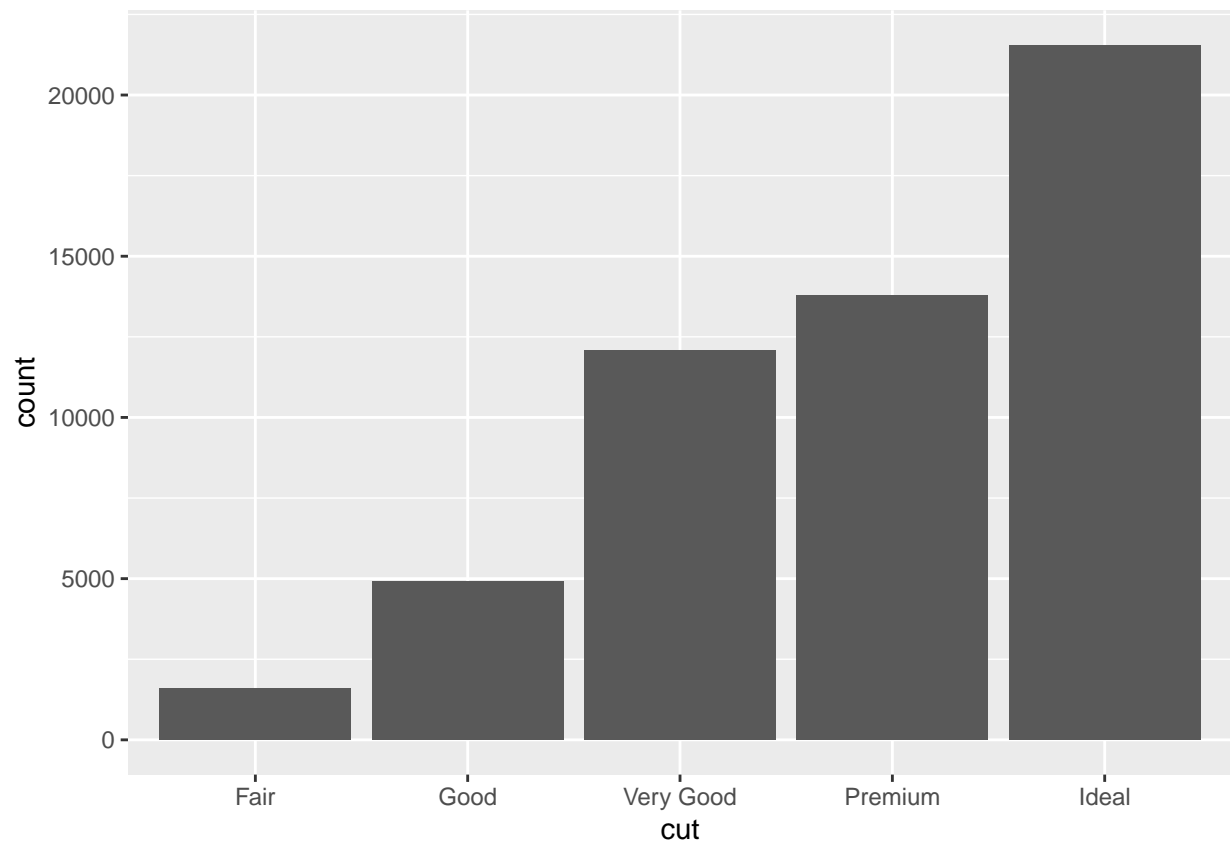
não existe diferença nos dois gráficos.

transformações estatísticas pag 22

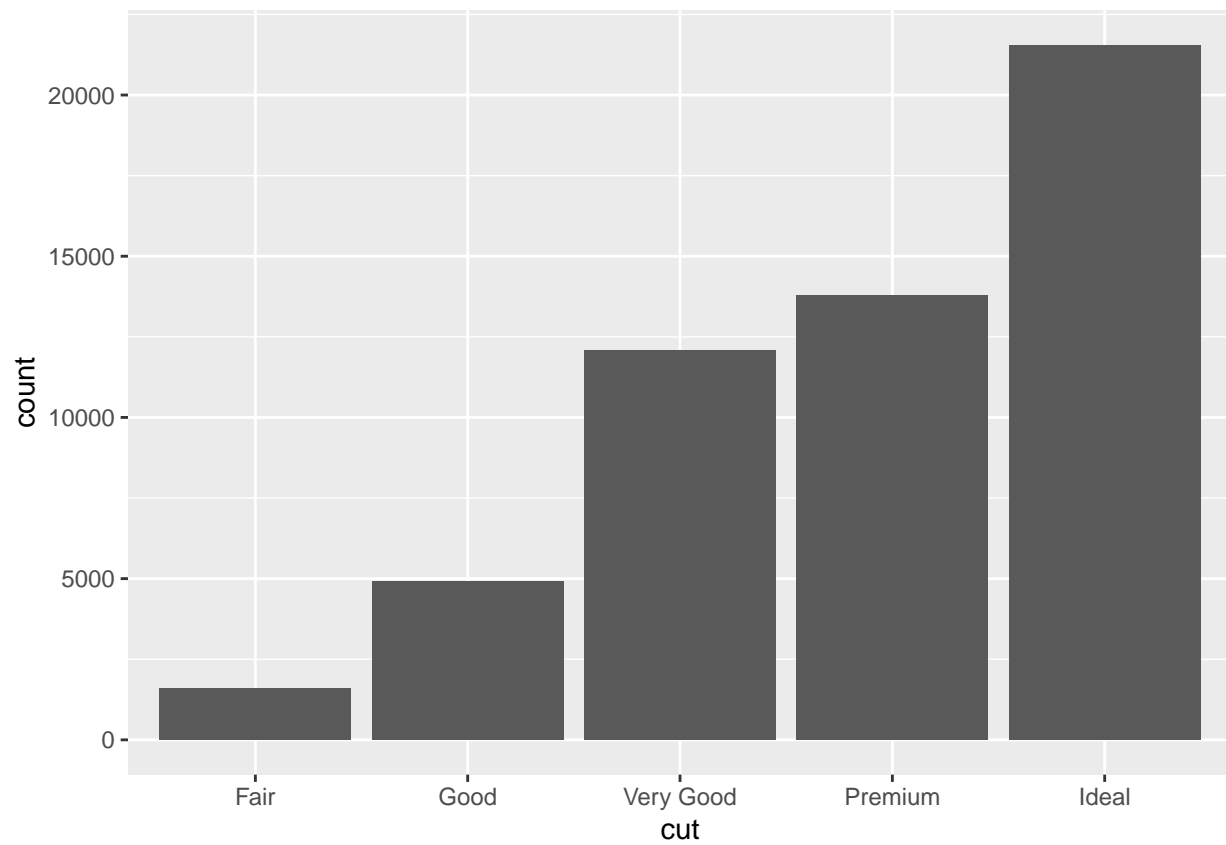
diamonds

```
## # A tibble: 53,940 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal    E     SI2     61.5   55   326  3.95  3.98  2.43
## 2  0.21 Premium E     SI1     59.8   61   326  3.89  3.84  2.31
## 3  0.23 Good    E     VS1     56.9   65   327  4.05  4.07  2.31
## 4  0.29 Premium I     VS2     62.4   58   334  4.2   4.23  2.63
## 5  0.31 Good    J     SI2     63.3   58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2    62.8   57   336  3.94  3.96  2.48
## 7  0.24 Very Good I     VVS1    62.3   57   336  3.95  3.98  2.47
## 8  0.26 Very Good H     SI1     61.9   55   337  4.07  4.11  2.53
## 9  0.22 Fair    E     VS2     65.1   61   337  3.87  3.78  2.49
## 10 0.23 Very Good H     VS1     59.4   61   338  4     4.05  2.39
## # i 53,930 more rows

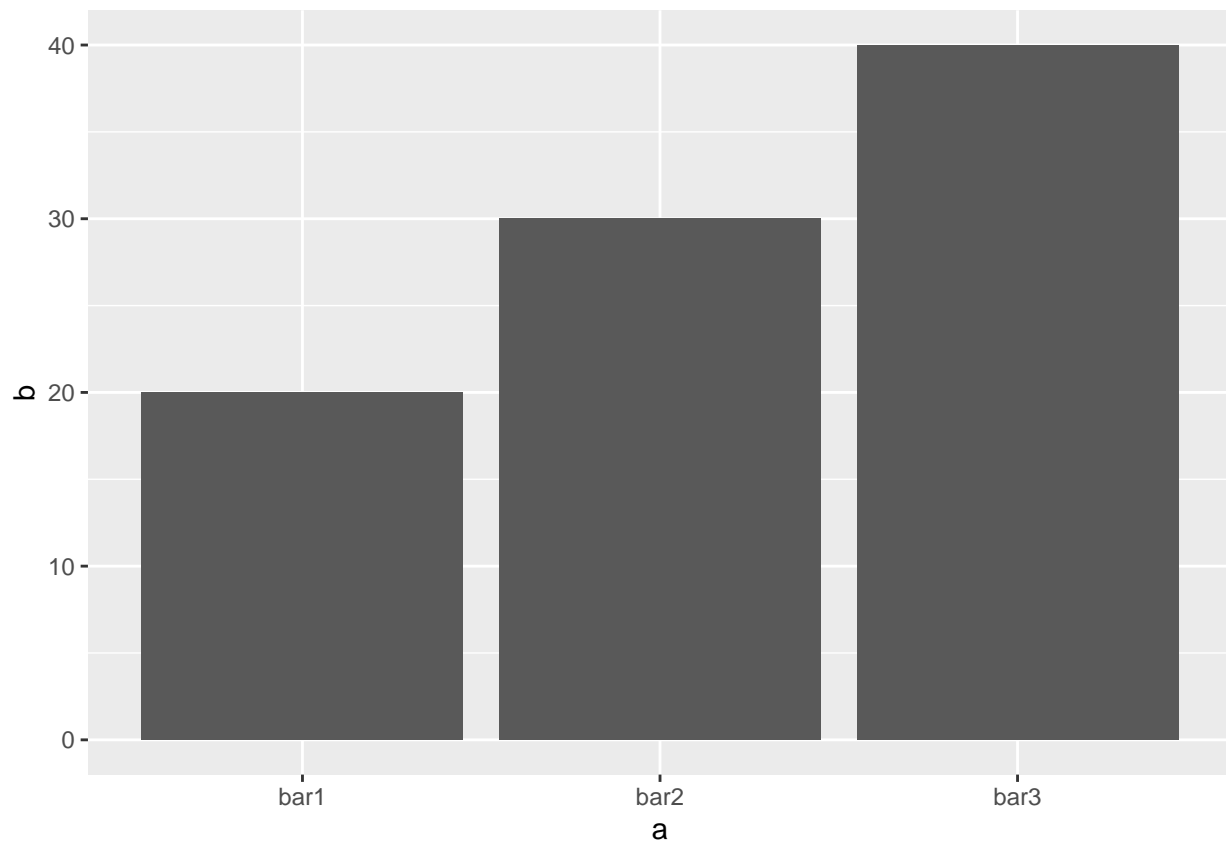
ggplot(data = diamonds) +
  geom_bar(mapping = aes(
    x = cut
  ))
```



```
ggplot(  
  data = diamonds  
) +  
  stat_count(mapping = aes(  
    x = cut  
  ))
```

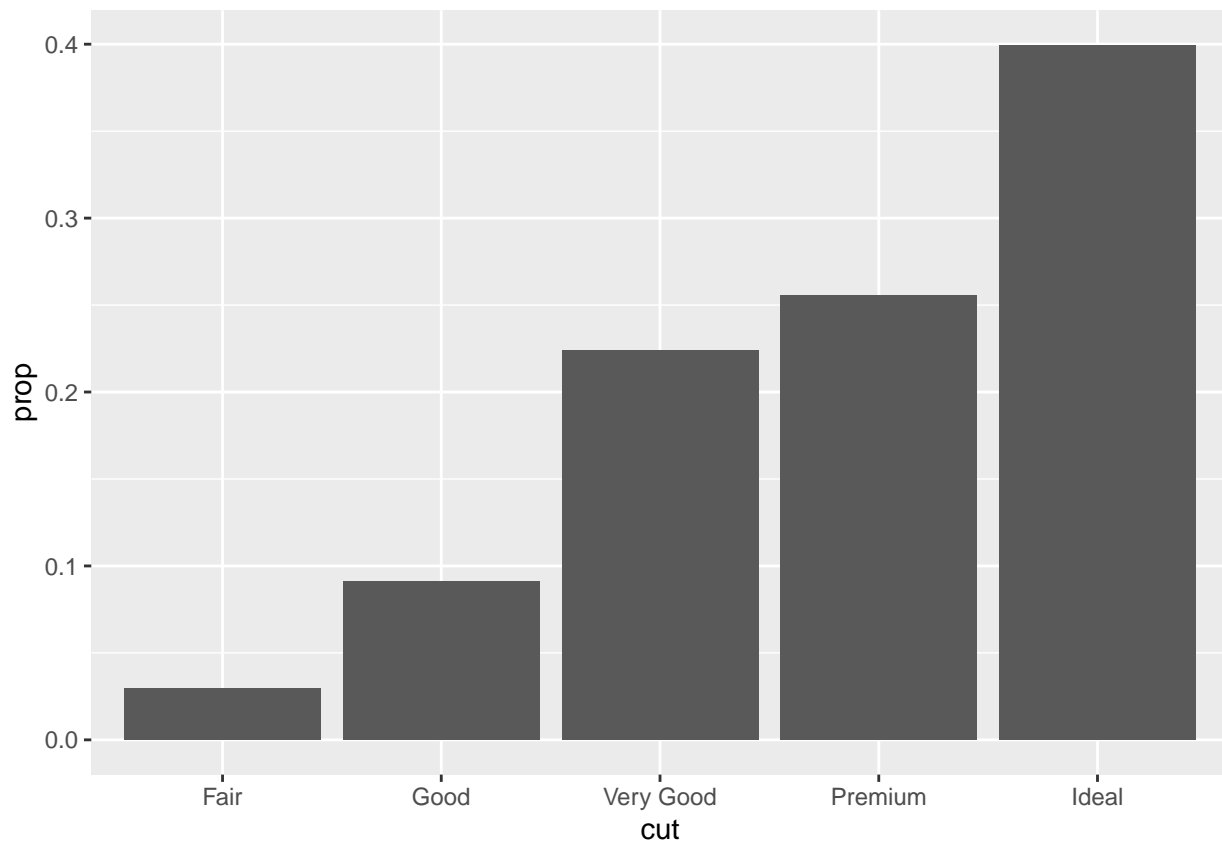


```
demo <- tribble(  
  ~a, ~b,  
  "bar1", 20,  
  "bar2", 30,  
  "bar3", 40  
)  
ggplot(data = demo) +  
  geom_bar(  
    mapping = aes(  
      x = a, y = b  
    ), stat = "identity"  
  )
```



```
ggplot(
  data = diamonds
) +
  geom_bar(
    mapping = aes(
      x = cut, y = ..prop.., group = 1
    )
  )
)
```

```
## Warning: The dot-dot notation (`..prop..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(prop)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

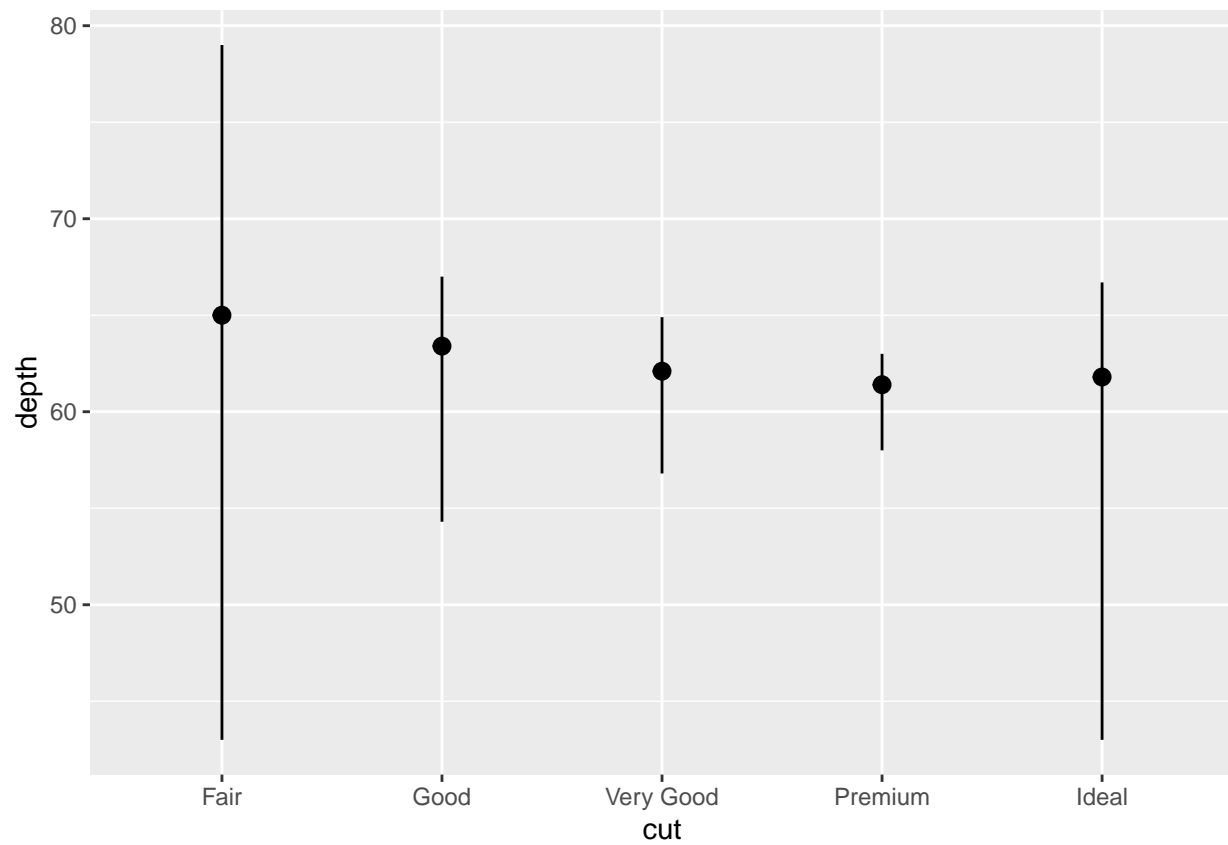


```
ggplot(data = diamonds) +
  stat_summary(
    mapping = aes(
      x = cut, y = depth
    ),
    fun.ymin = min,
    fun.ymax = max,
    fun.y = median
  )
```

```
## Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
## i Please use the `fun` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

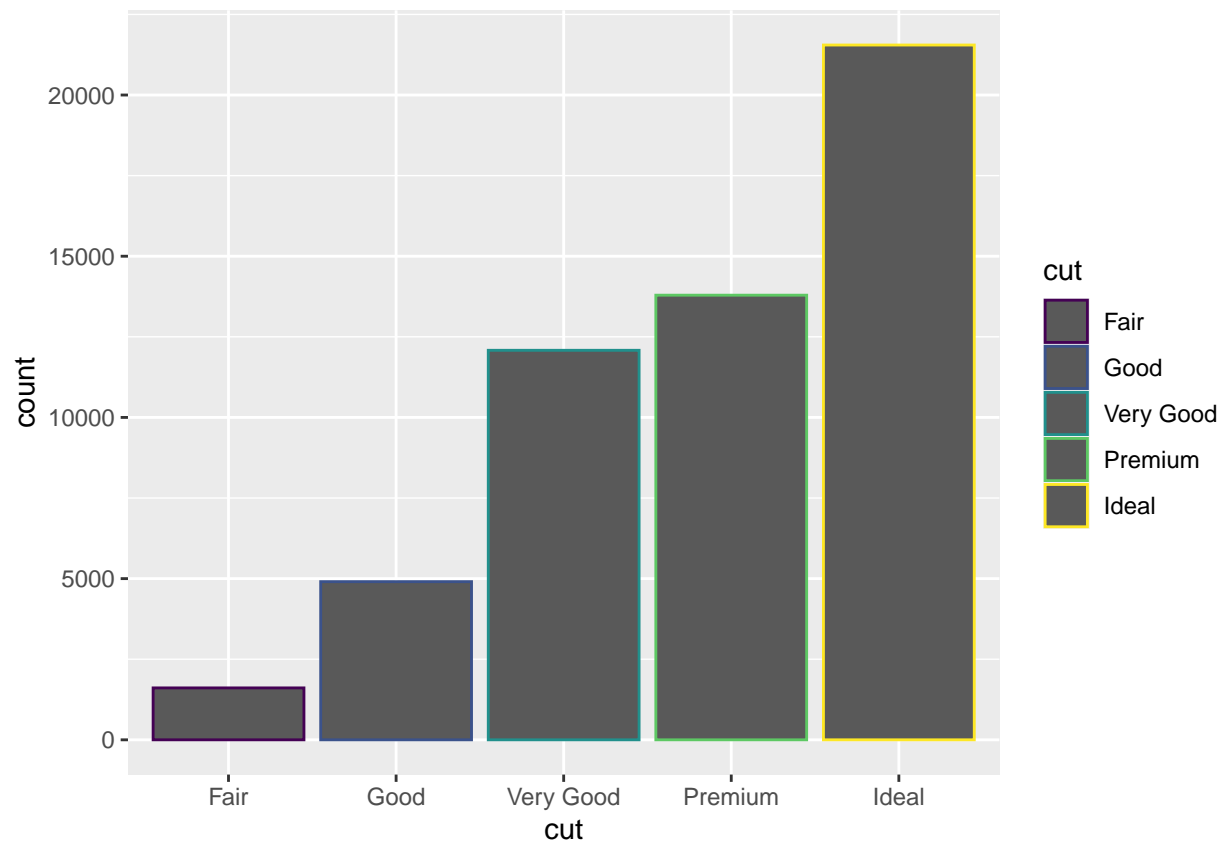
```
## Warning: The `fun.ymin` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
## i Please use the `fun.min` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `fun.ymax` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
## i Please use the `fun.max` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

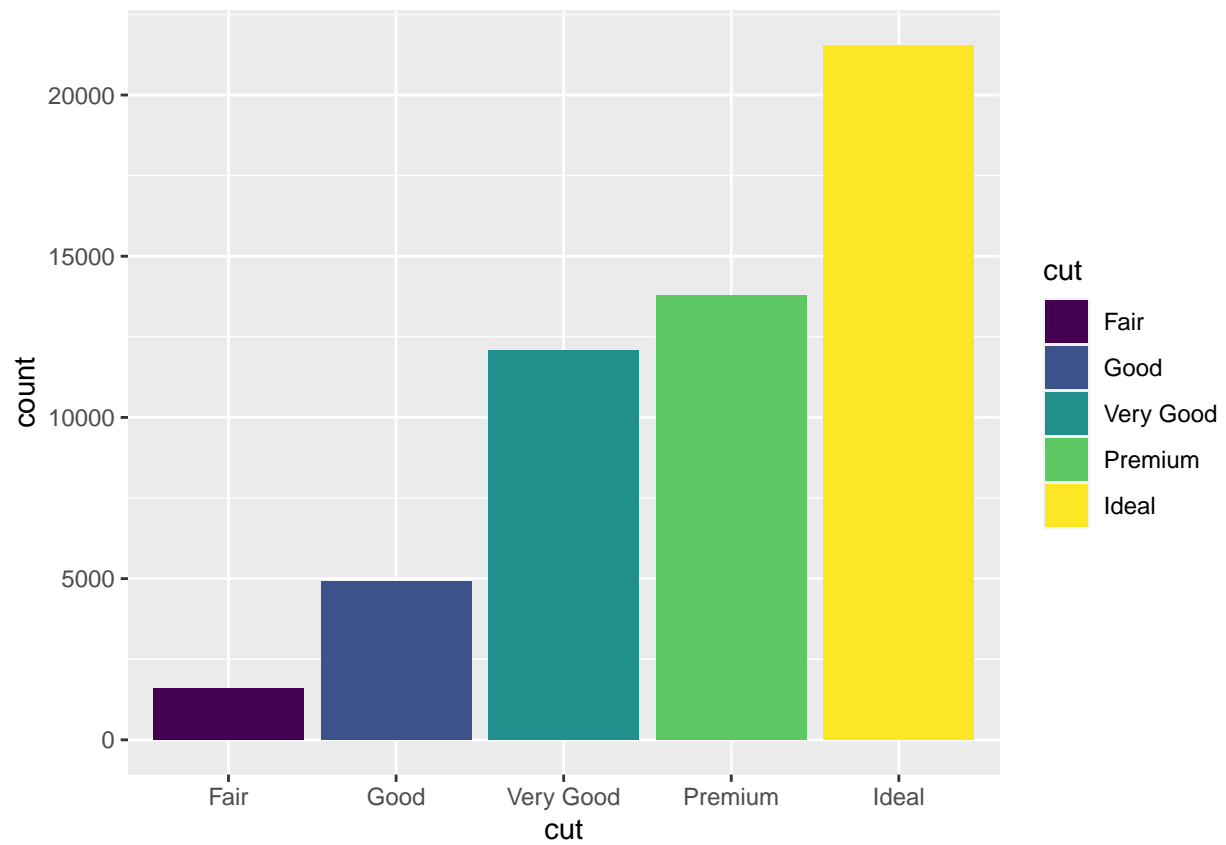



ajustes de posição pg 27

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

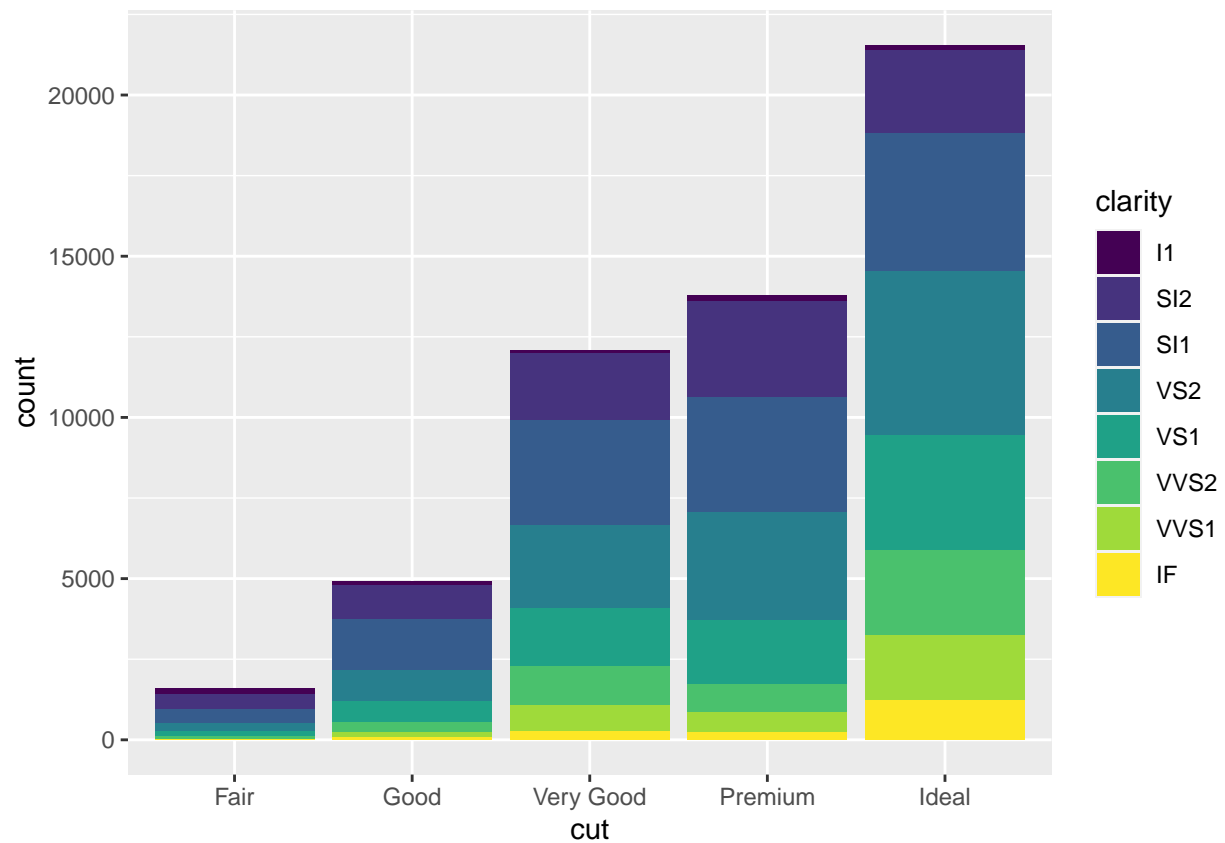


```
ggplot(data = diamonds) +  
  geom_bar(  
    mapping = aes(  
      x = cut, fill = cut  
    )  
  )  
)
```

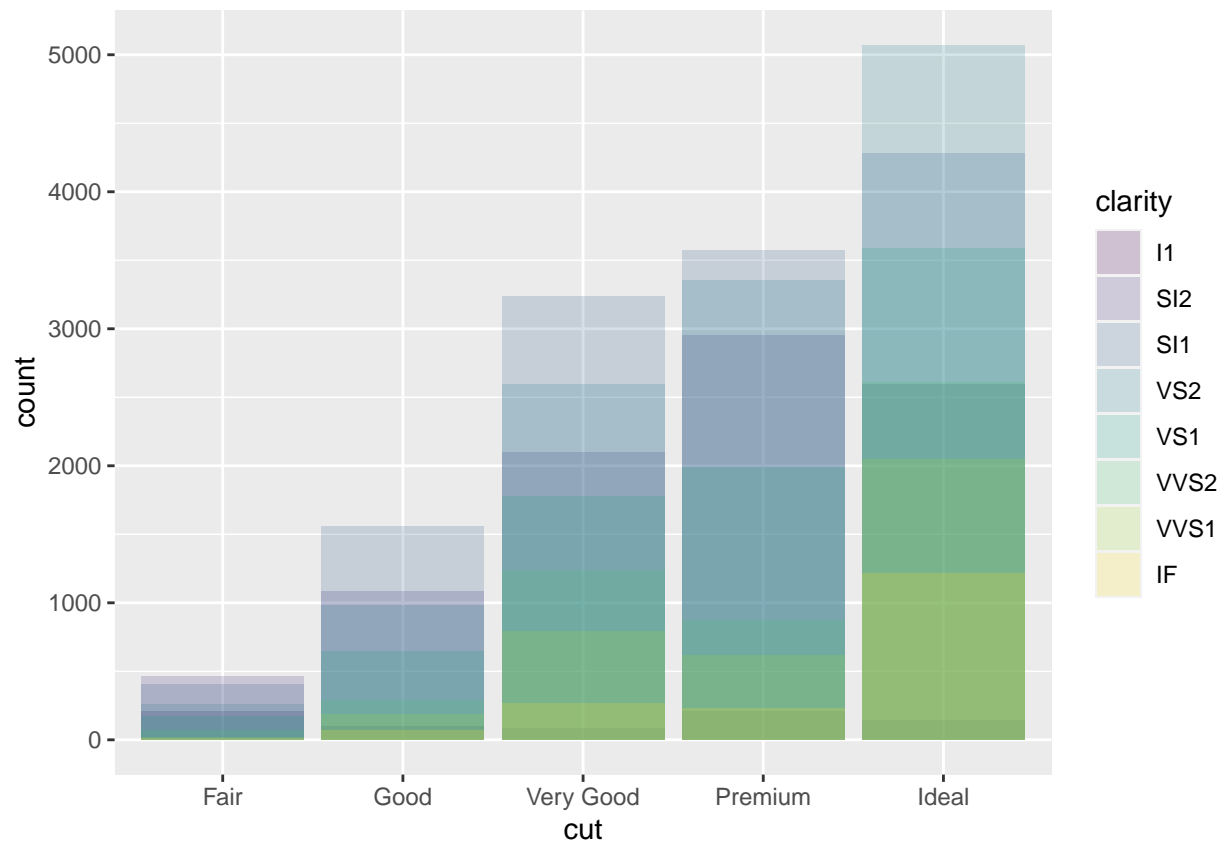


não consigo exibir os dois graficos ao mesmo tempo

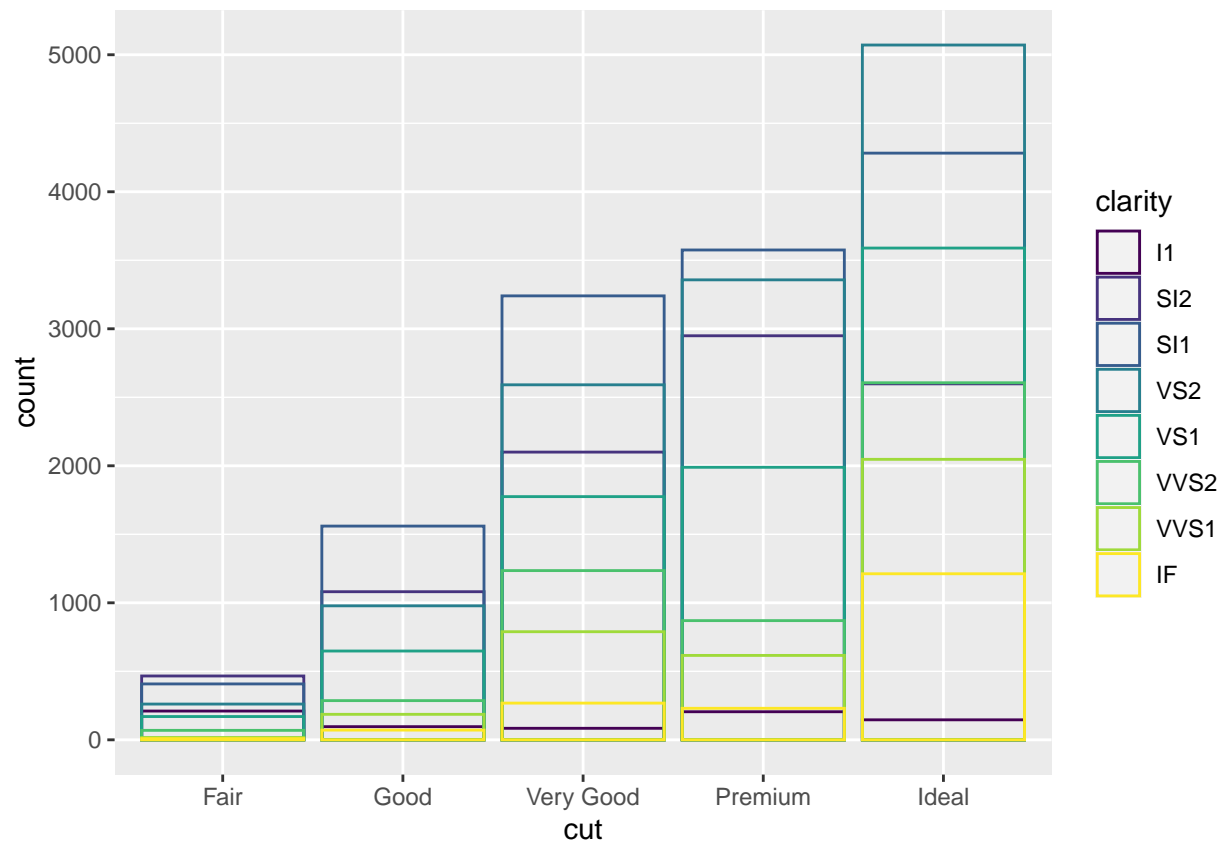
```
ggplot( data = diamonds ) +  
  geom_bar(  
    mapping = aes(  
      x = cut, fill = clarity  
    )  
  )  
)
```



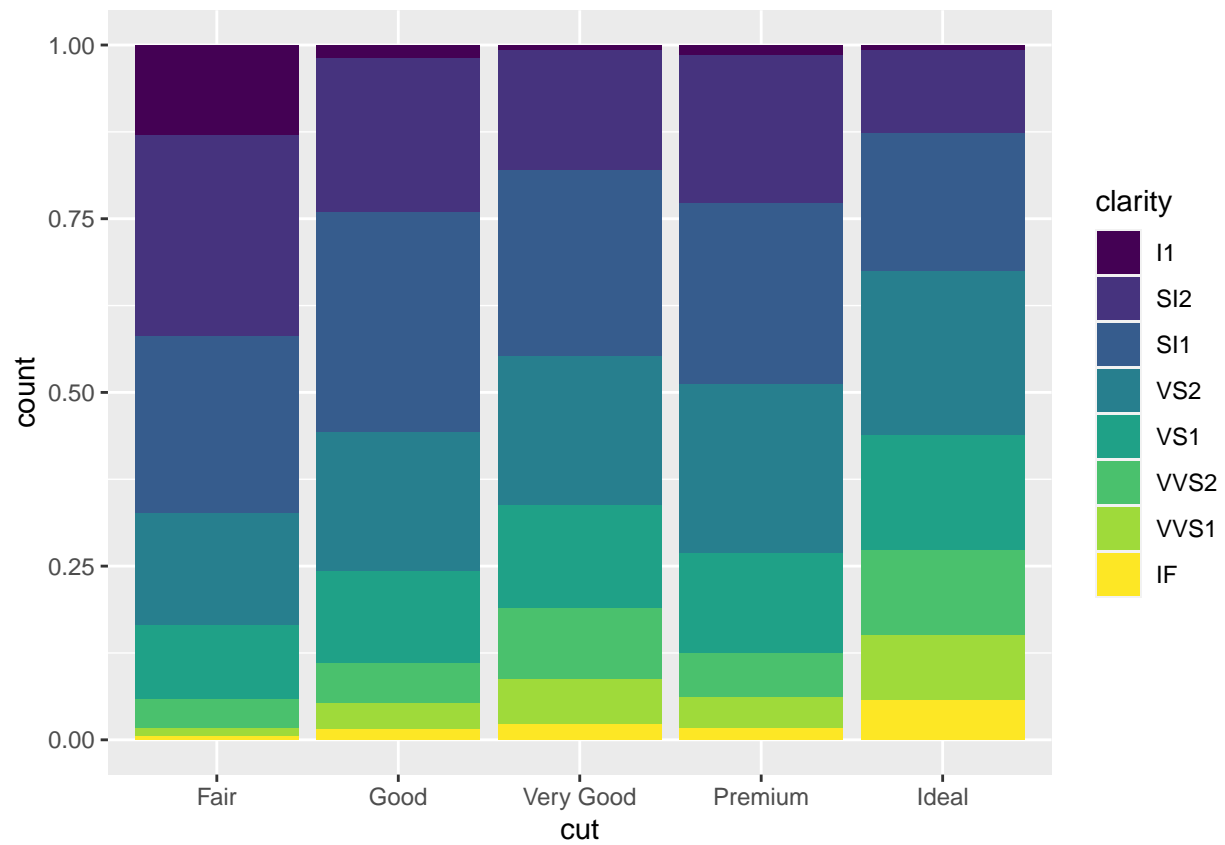
```
ggplot(
  data = diamonds,
  mapping = aes(
    x = cut, fill = clarity
  )
) +
  geom_bar(
    alpha = 1/5, position = "identity" #inutil para grafico de barras, pois sobrepe
  )
```



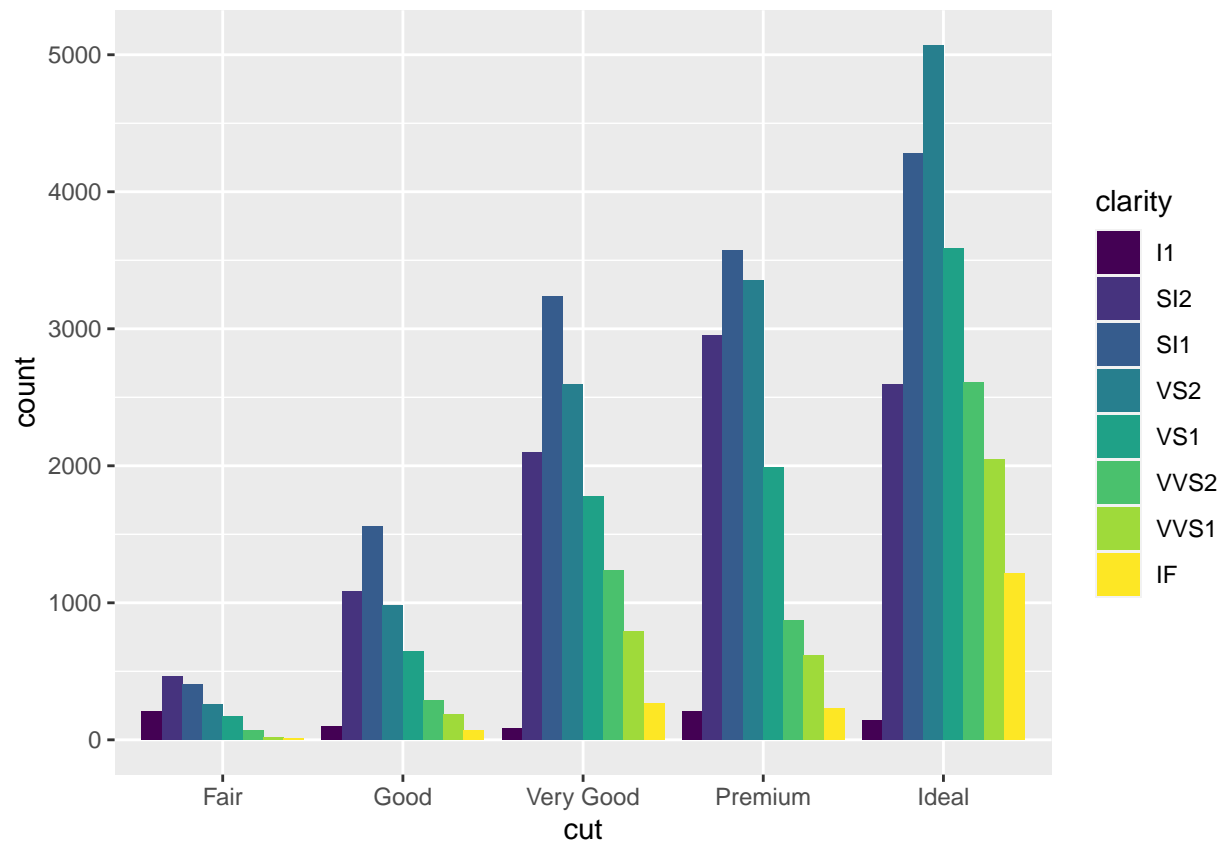
```
ggplot(
  data = diamonds,
  mapping = aes(
    x = cut, color = clarity
  )
) +
  geom_bar(
    fill = NA, position = "identity" #fill = NA, torna-o completamente transparente
  )
```



```
ggplot(
  data = diamonds
) +
  geom_bar(
    mapping = aes(
      x = cut, fill = clarity
    ), position = "fill" # position fill fica todo preenchido
    #fill funciona como empilhamento da mesma altura, facilita comparar proporções
  )
```

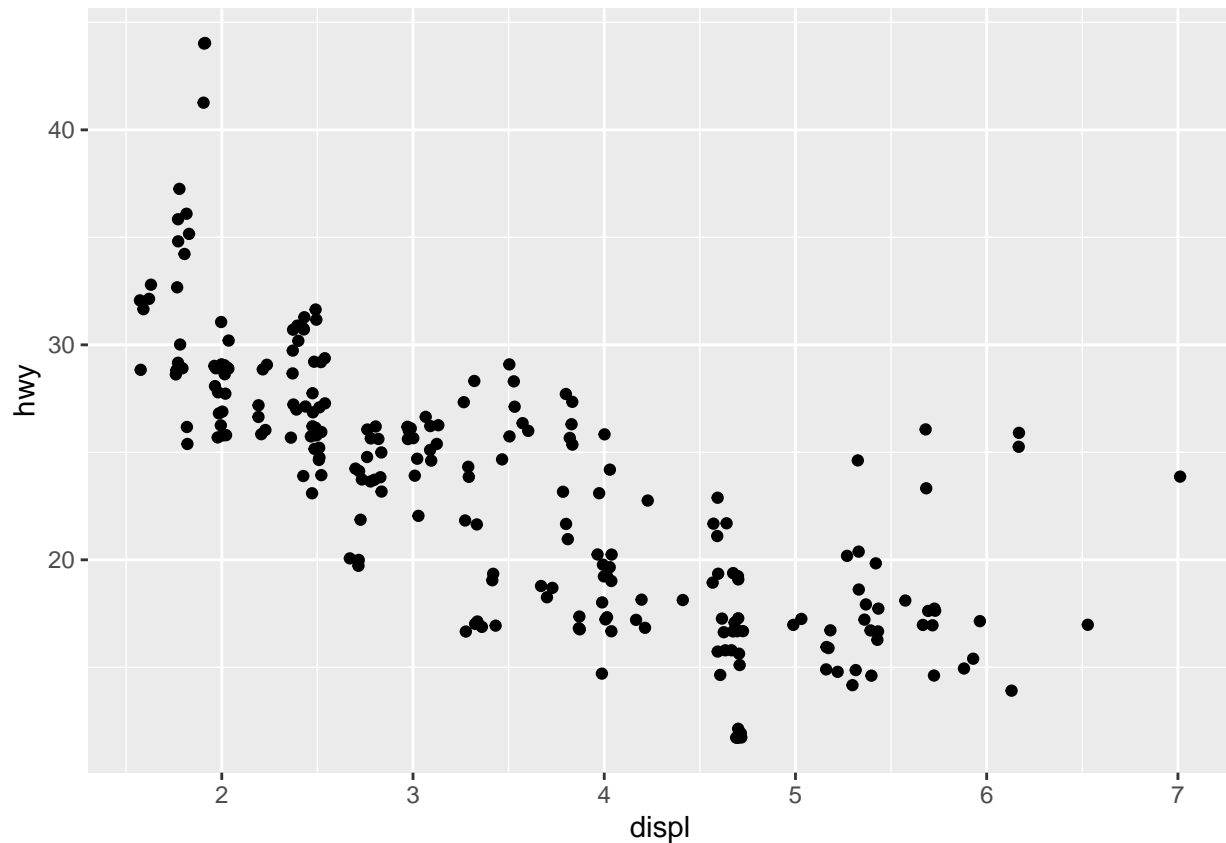


```
ggplot(
  data = diamonds
) +
  geom_bar(
    mapping = aes(
      x = cut, fill = clarity
    ), position = "dodge"
  )
```



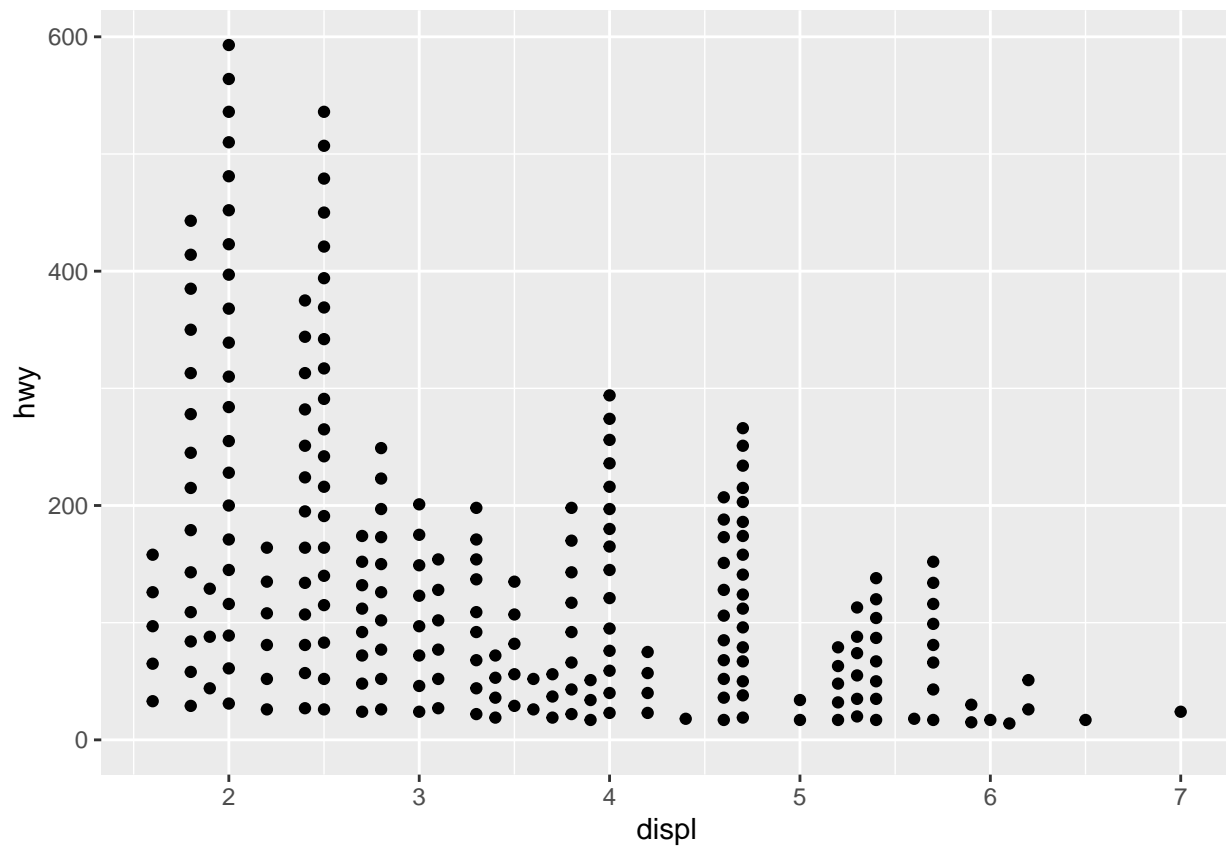
dodge coloca objetos sobrepostos diretamente um ao lado do outro, facilita a comparação de valores individuais

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

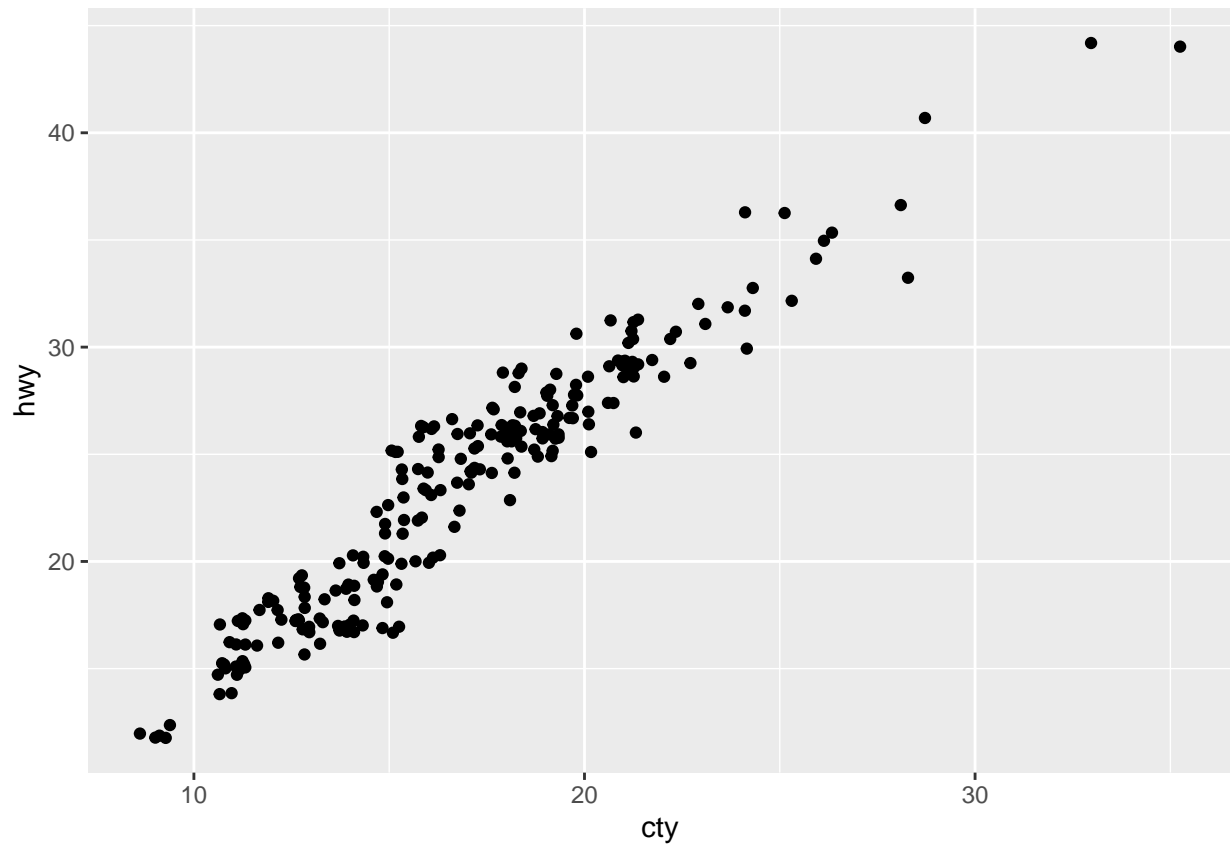
para evitar sobreposição de pontos no gráficos, podemos utilizar o `position = jitter` pois ele adiciona uma pequena quantidade de ruídos aleatórios a cada ponto, isso espalha os pontos, porque não é provável que dois pontos quaisquer receba a mesma quantidade de ruídos aleatório,

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



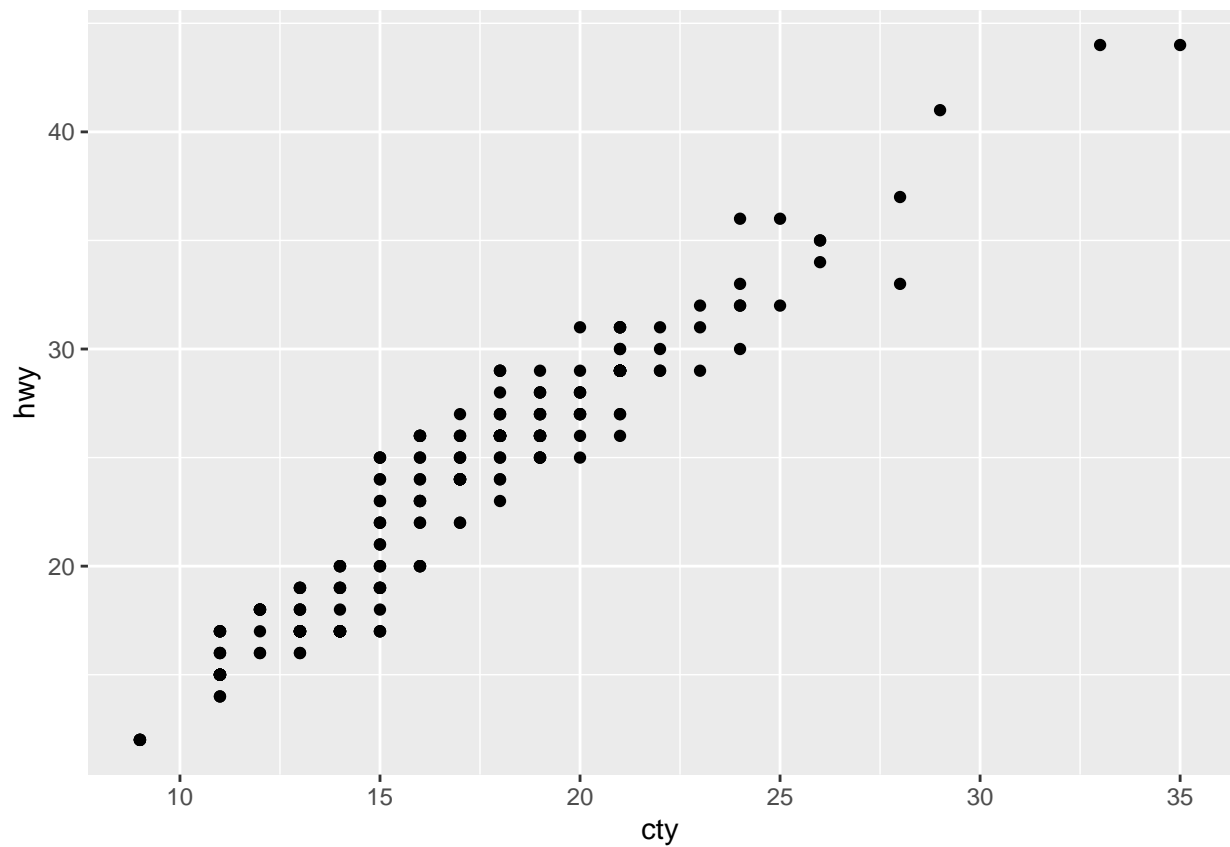
exer 1 pg 31 descreve a sobreposição dos dados, é necessário adicionar o position jitter

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



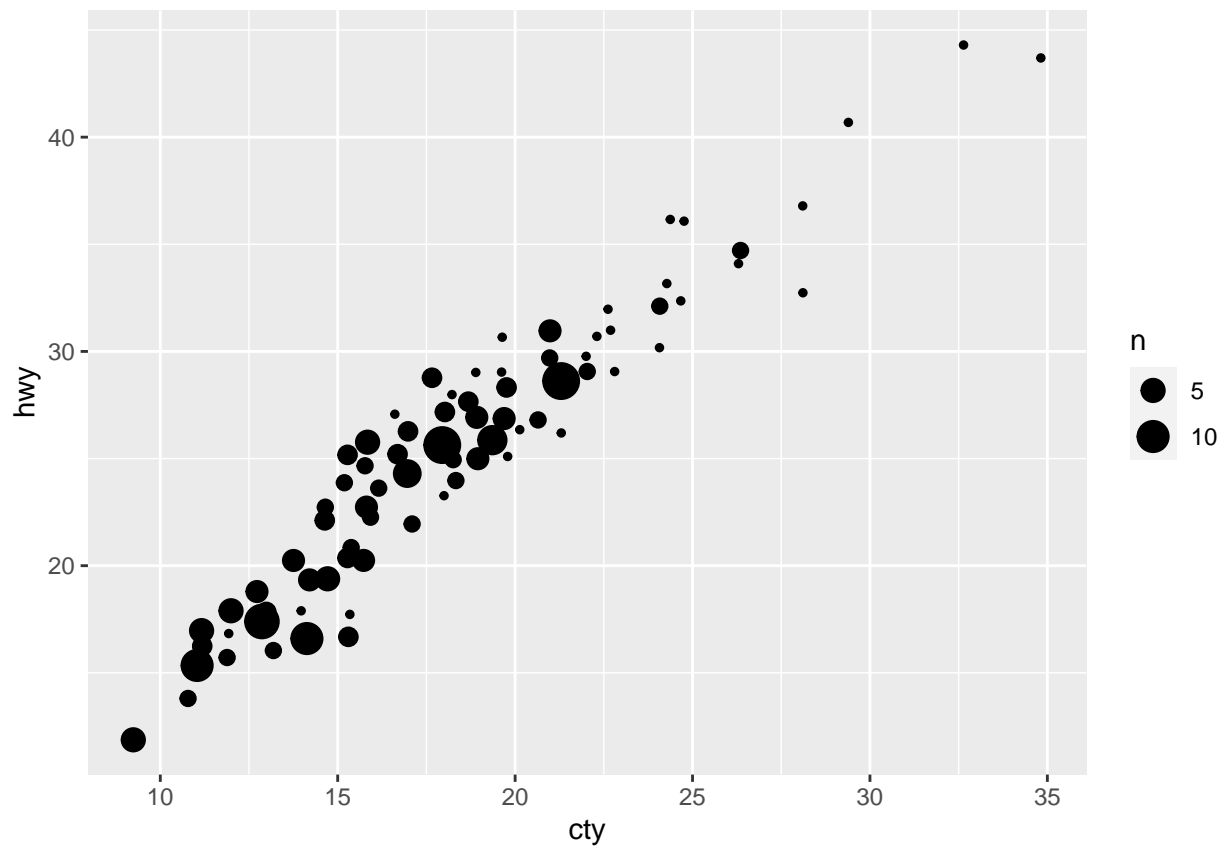
exer 2

```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +  
  geom_jitter(height = 0, width = 0)
```



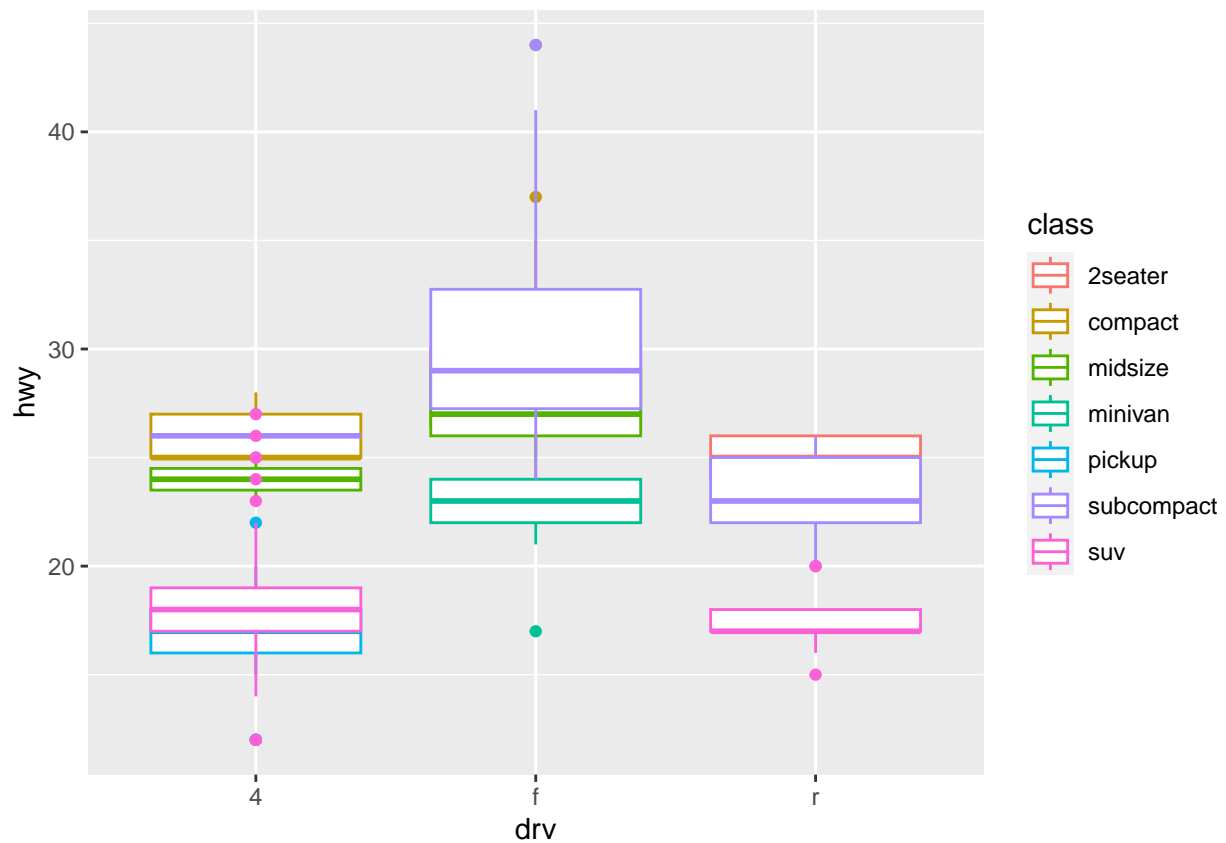
exer 3

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



exer 4

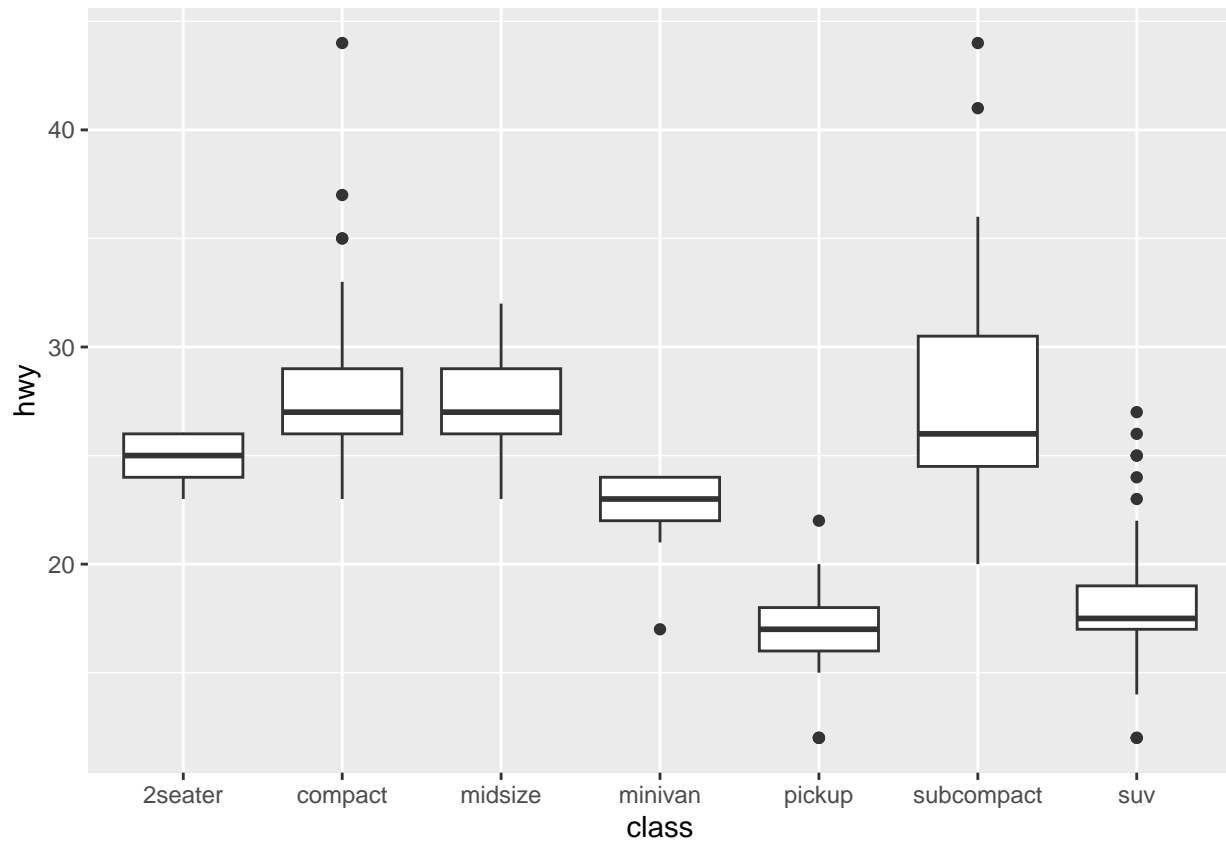
```
ggplot(data = mpg, aes(x = drv, y = hwy, colour = class)) +  
  geom_boxplot(position = "identity")
```



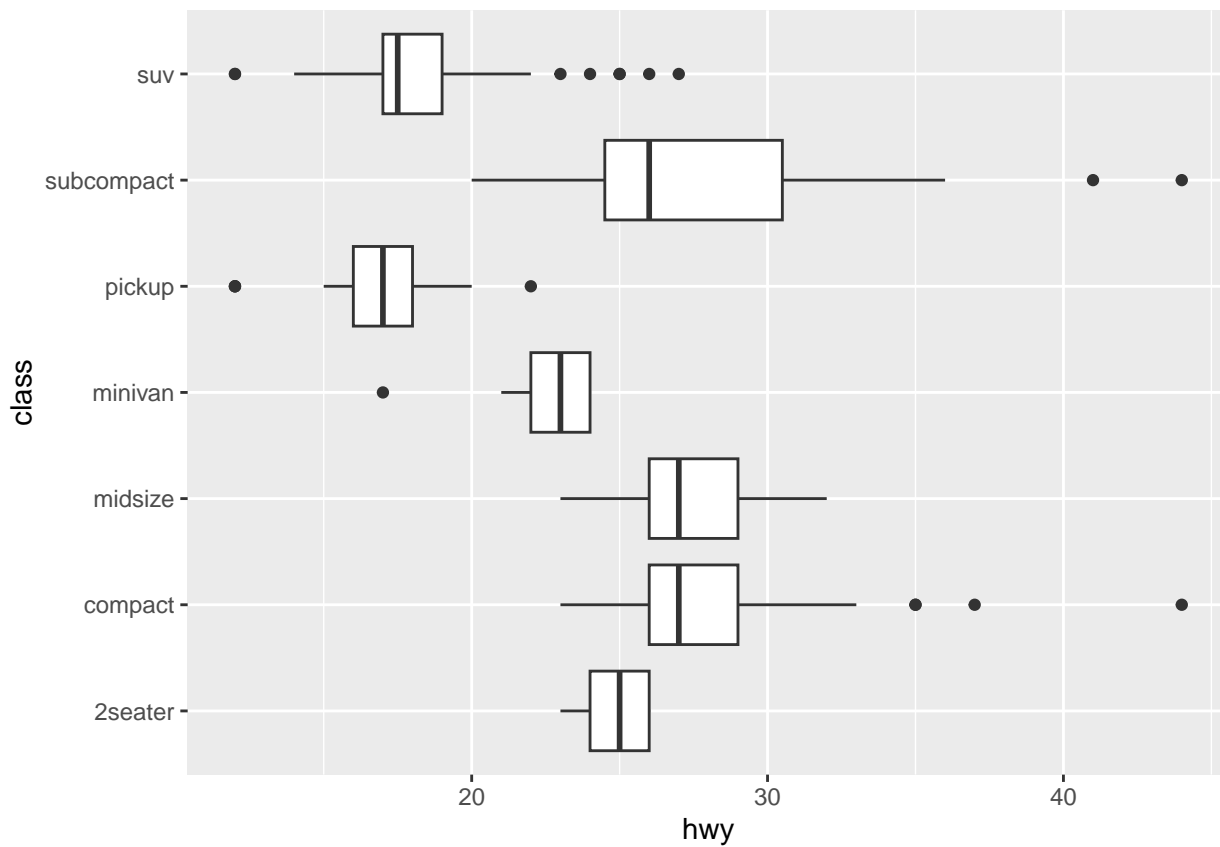
sistemas de coordenadas

coord_flip() troca os eixos x e y, é bom para rótulos longos

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

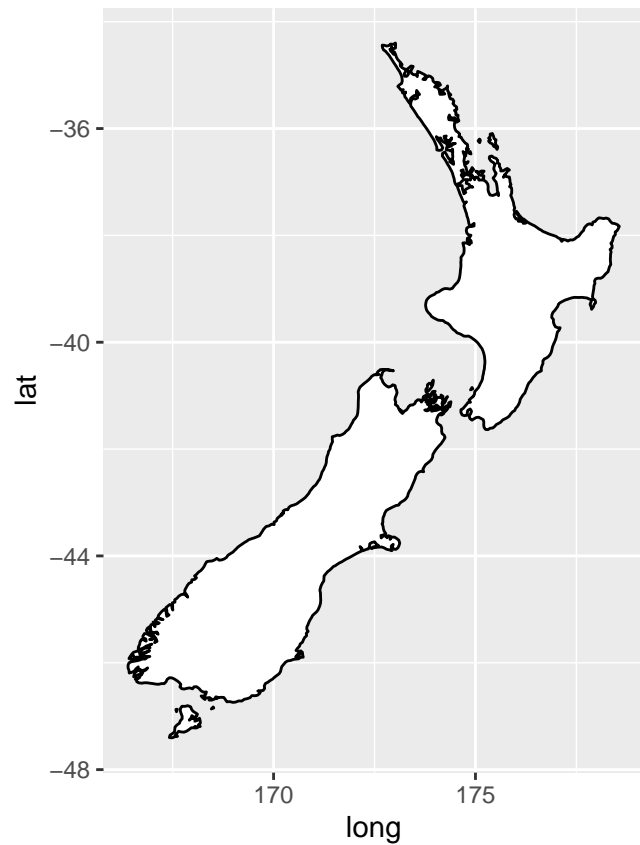


```
ggplot(  
  data = mpg, mapping = aes(  
    x = class, y = hwy  
  )  
) +  
  geom_boxplot() +  
  coord_flip()
```



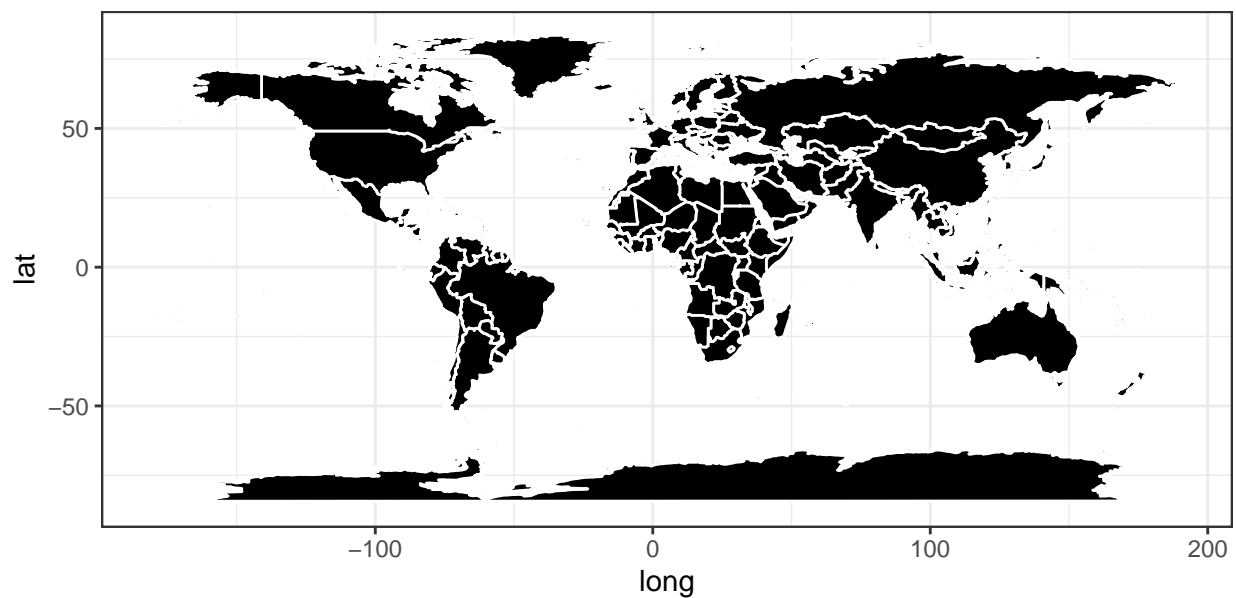
```
nz <- map_data("nz")

ggplot(nz, aes(
  long, lat, group = group
)) +
  geom_polygon(fill = "white", color = "black") +
  coord_quickmap()
```

`coord_quickmap` configura a proporção de tela corretamente

```
world <- map_data("world")
ggplot(world, aes(
  long, lat, group = group
)) +
  geom_polygon(fill = "black", color = "white") +
  coord_quickmap() + theme_bw()
```

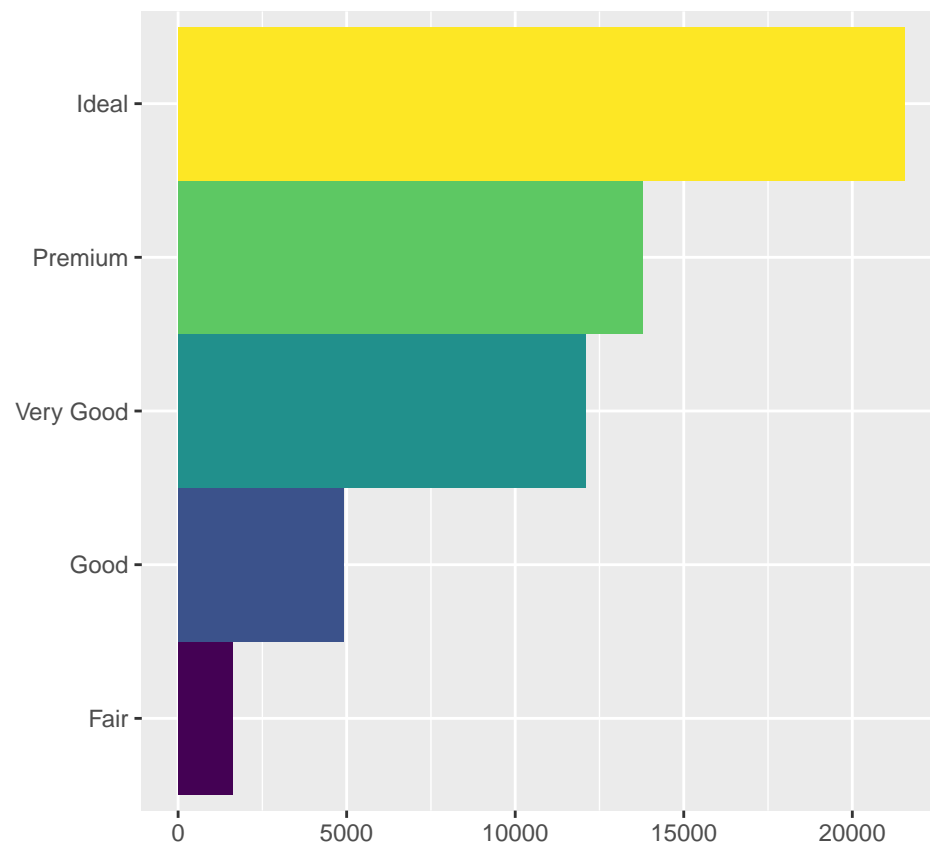


```

bar <- ggplot(data = diamonds) +
  geom_bar(
    mapping = aes(
      x = cut, fill = cut
    ), show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)

bar + coord_flip()

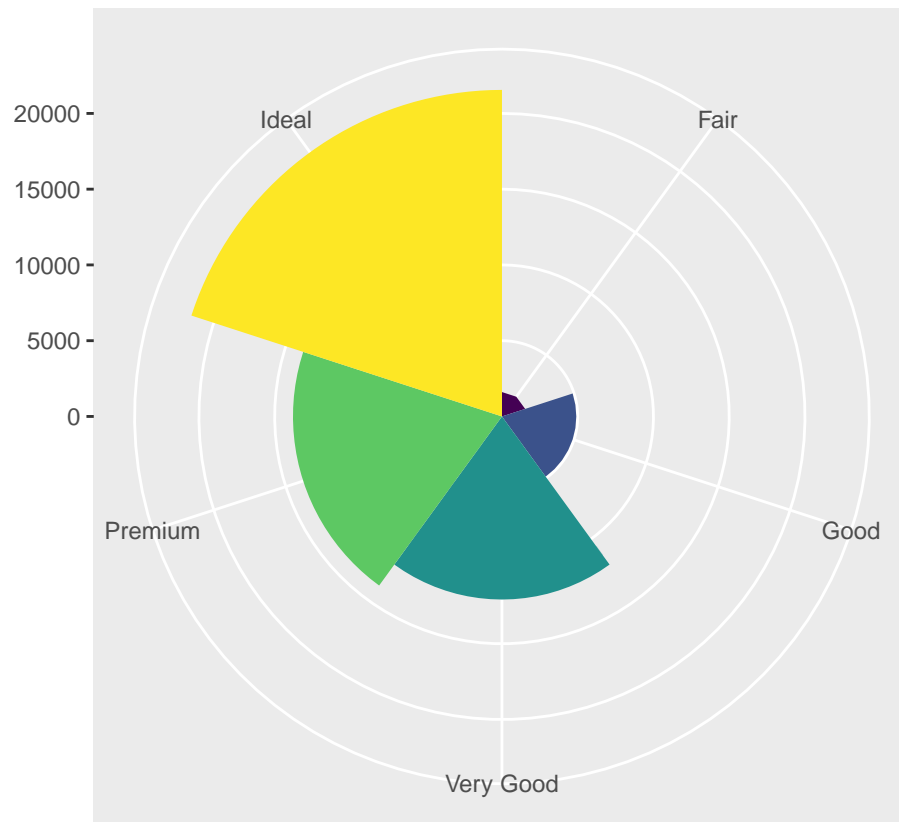
```



```

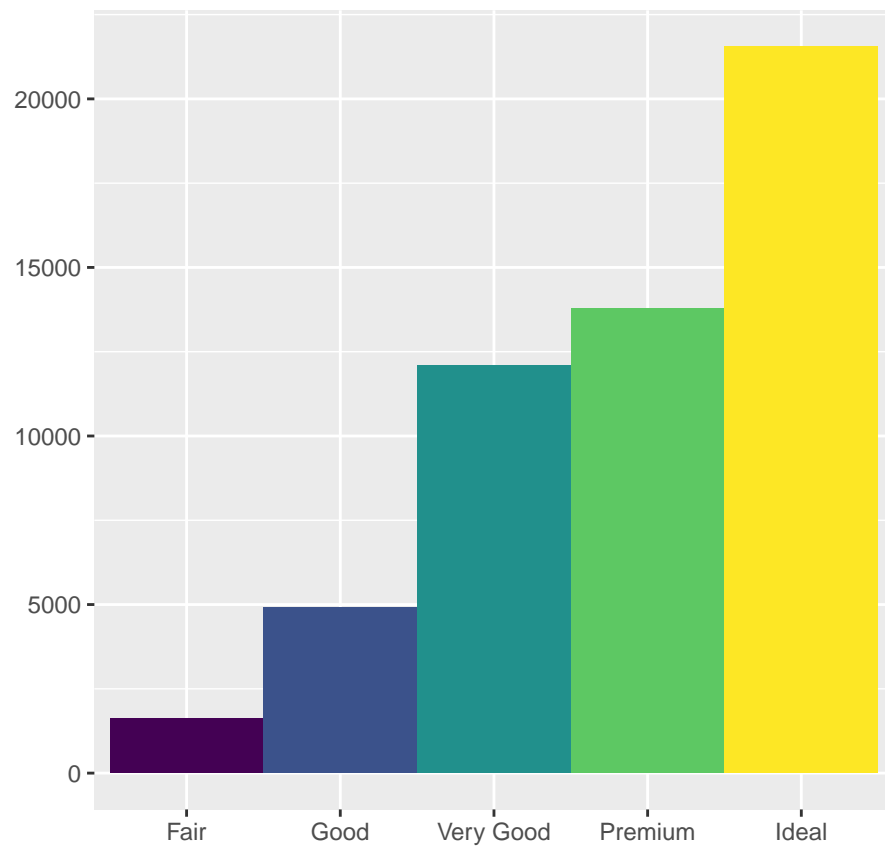
bar + coord_polar()

```



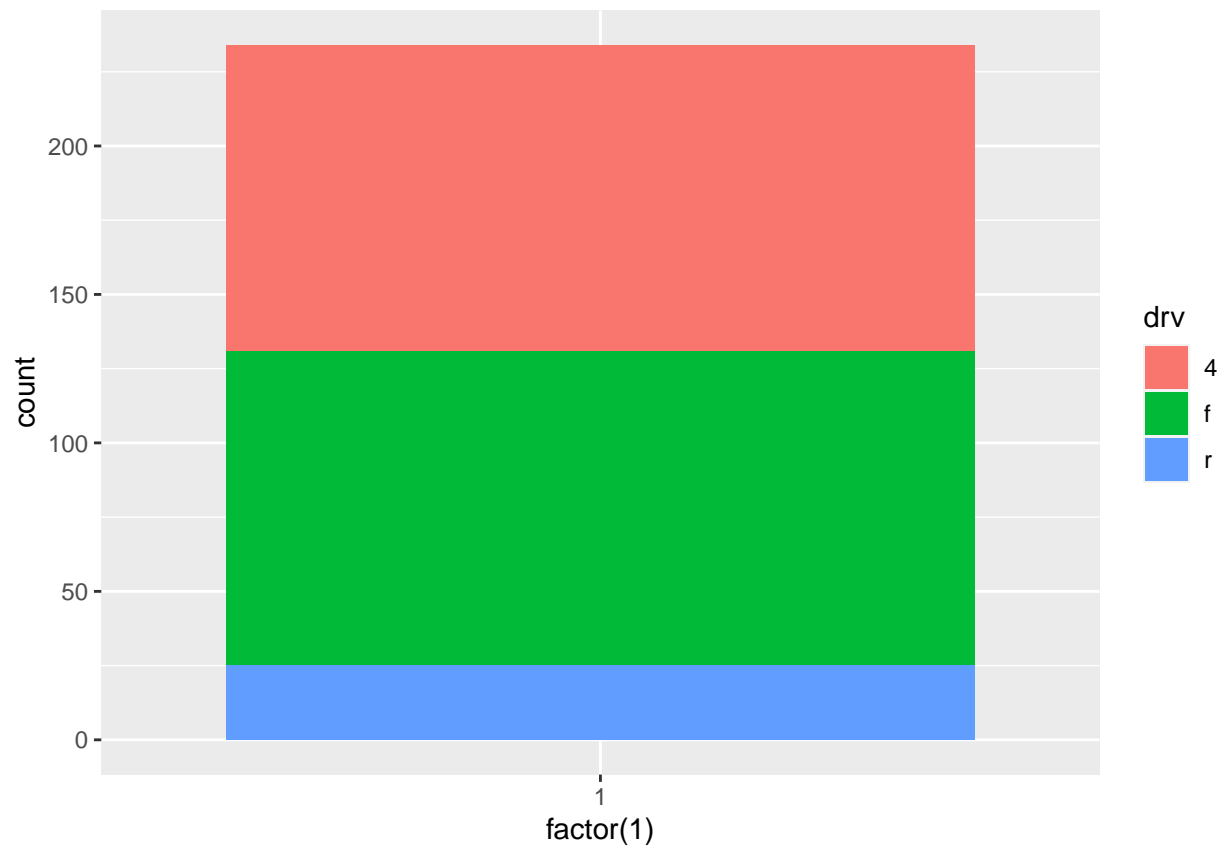
coord_polar usa coordenadas polares, que revelam uma conexão interessante entre um gráfico de barras e gráfico de setores.

bar

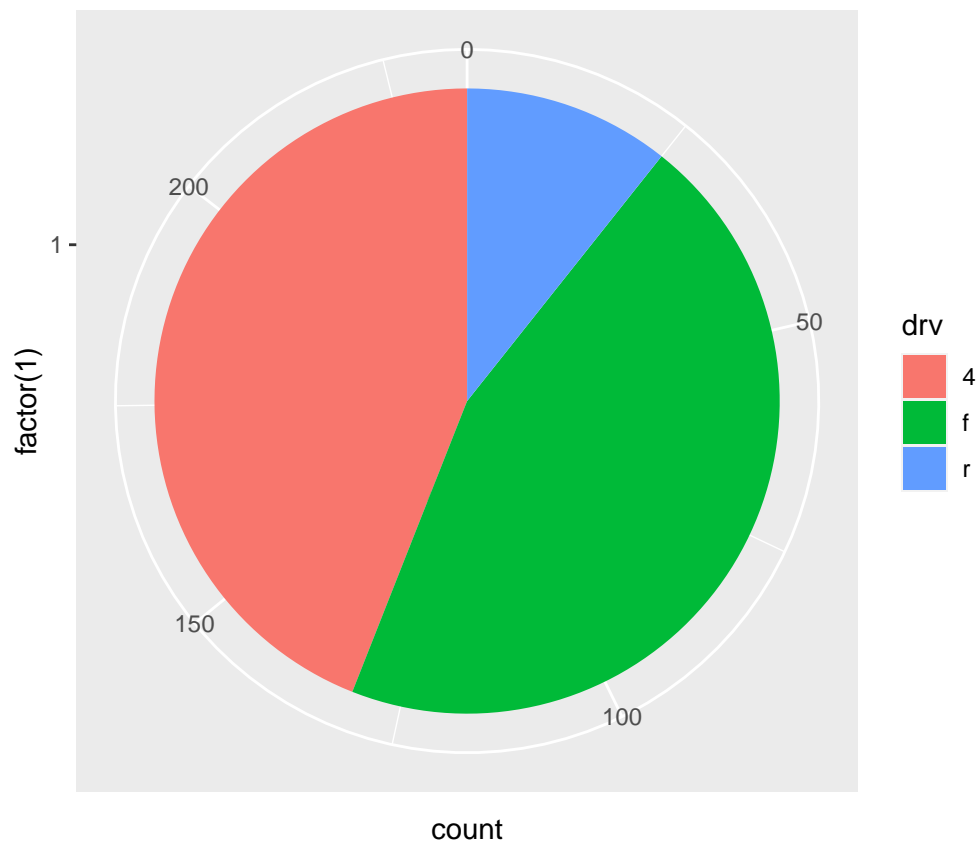


exer 1 pg 33

```
ggplot(  
  mpg, aes(  
    x = factor(1), fill = drv  
  )  
) +  
  geom_bar()
```

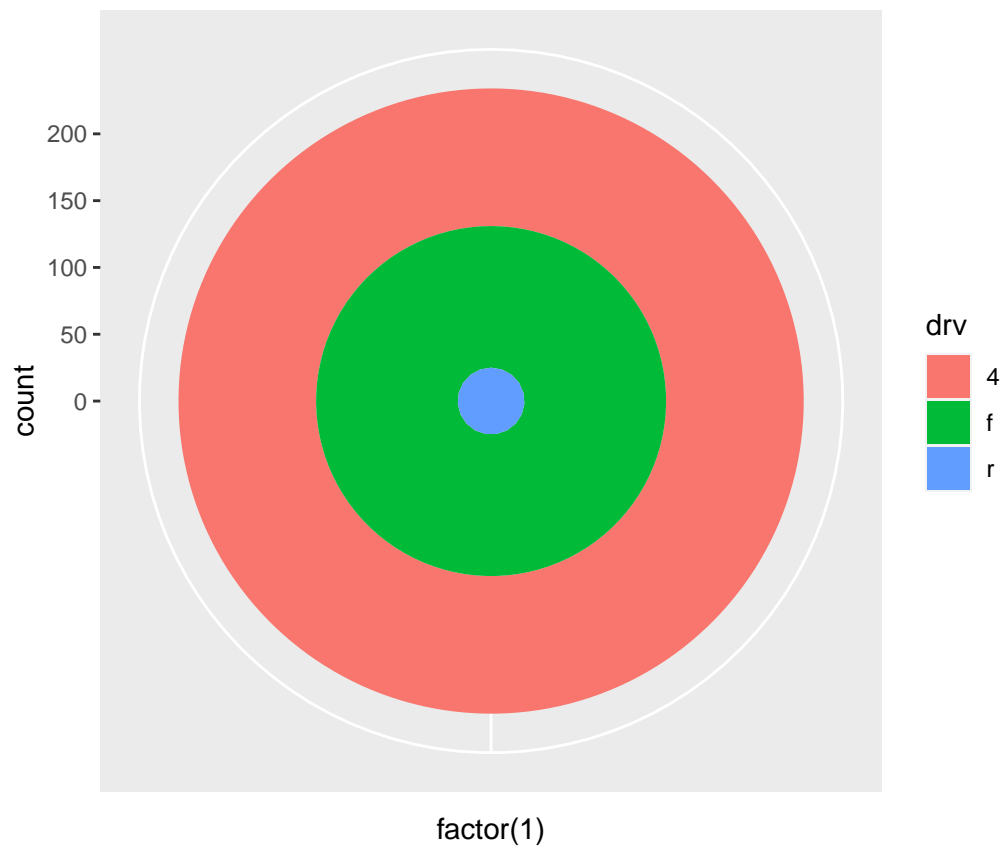


```
ggplot(  
  mpg, aes(  
    x = factor(1), fill = drv  
  )  
) +  
  geom_bar(width = 1) +  
  coord_polar(theta = "y")
```



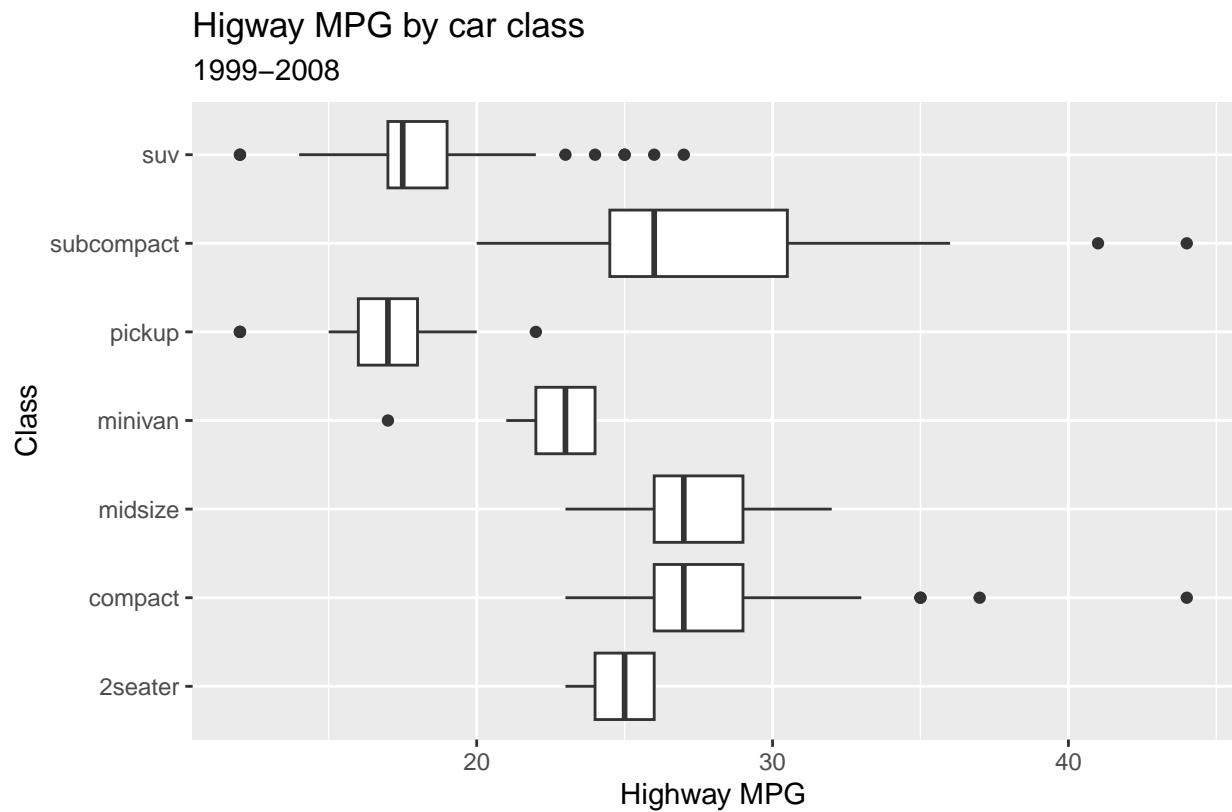
o argumento `theta=y` é o ângulo de início da seção
se deixar em branco

```
ggplot(
  mpg, aes(
    x = factor(1), fill = drv
  )
) +
  geom_bar(width = 1) +
  coord_polar()
```



exer2 labs adicionda titulos, titulos plots, capitulos

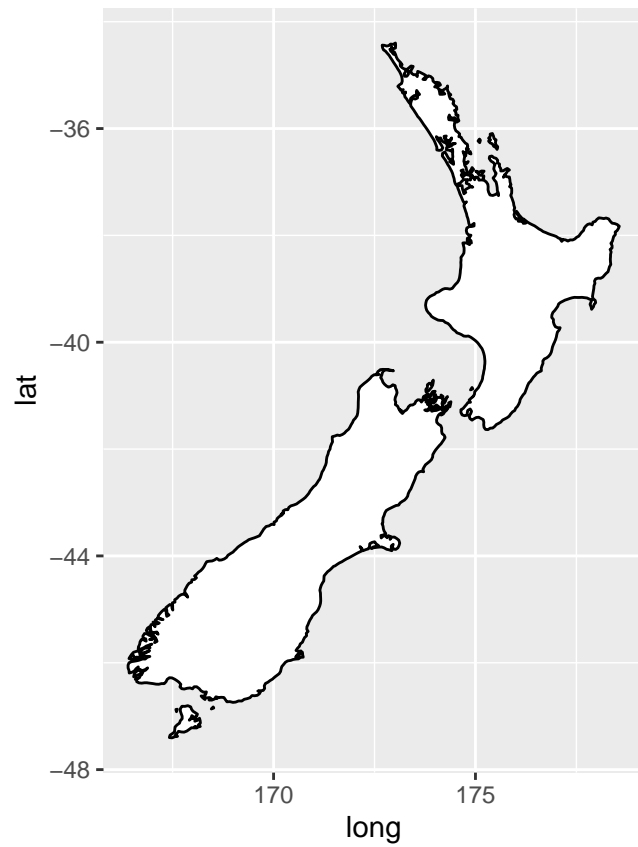
```
ggplot(
  data = mpg,
  mapping = aes(
    x = class, y = hwy
  )
) +
  geom_boxplot() +
  coord_flip() +
  labs(y = "Highway MPG",
       x = "Class",
       title = "Higway MPG by car class",
       subtitle = "1999-2008",
       caption = "Source: http://fueleconomy.gov")
```



exer 3 coord_quickmap X coord_map — se pegarmos o mapa da NZ.

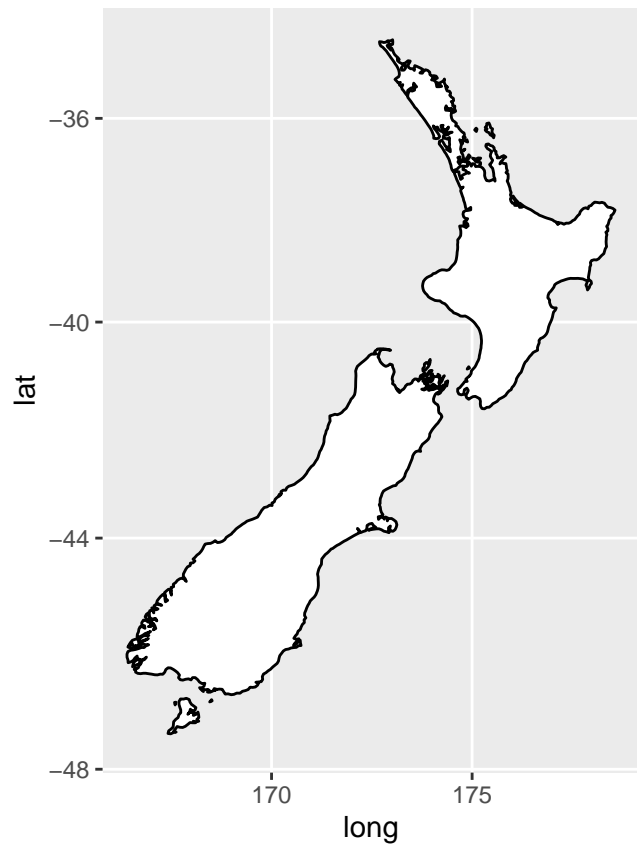
```
nz <- map_data("nz")

ggplot(nz, aes(
  long, lat, group = group
)) +
  geom_polygon(fill = "white", color = "black") +
  coord_quickmap()
```

quickmap mantém a proporção correta dos mapas.

```
ggplot(nz, aes(  
  long, lat, group = group  
) +  
  geom_polygon(fill = "white", color = "black") +  
  coord_map()
```



coord_map utiliza-se mapas de projeção de 3 dimensões.. projeção de mercator

coord_quickmap é a plotagem mais rapida, porem ela ignora a curvatura da terra e ajusta o mapa para lat/long ratio

exer 4

```
p<- ggplot(  
  data = mpg, mapping = aes(  
    x = cty, y = hwy  
  )  
) + geom_point() +  
  geom_abline()  
p + coord_fixed()
```

