

# mpg activities

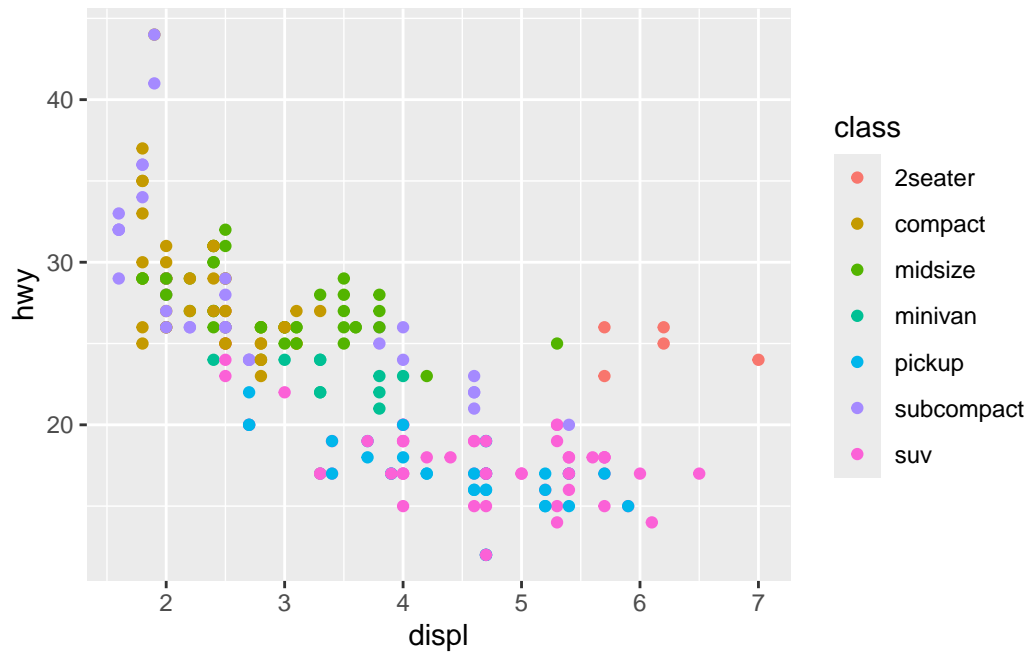
## mpg activities

---

- In this exercise, i'm going to do a two dimensional dispersion graph to visualize the relationship between **highway miles per gallon** x **engine displacement** x **class** from the cars present in the dataframe *mpg*, native to *TidyVerse* library.
- I'll be using the function `ggplot`, from the package **GGPlot2**.
- In my graph, **highway miles per gallon (hwy)** will be represented by the y axis, **engine displacement (displ)** by the x axis and the **class** by the color of the points.

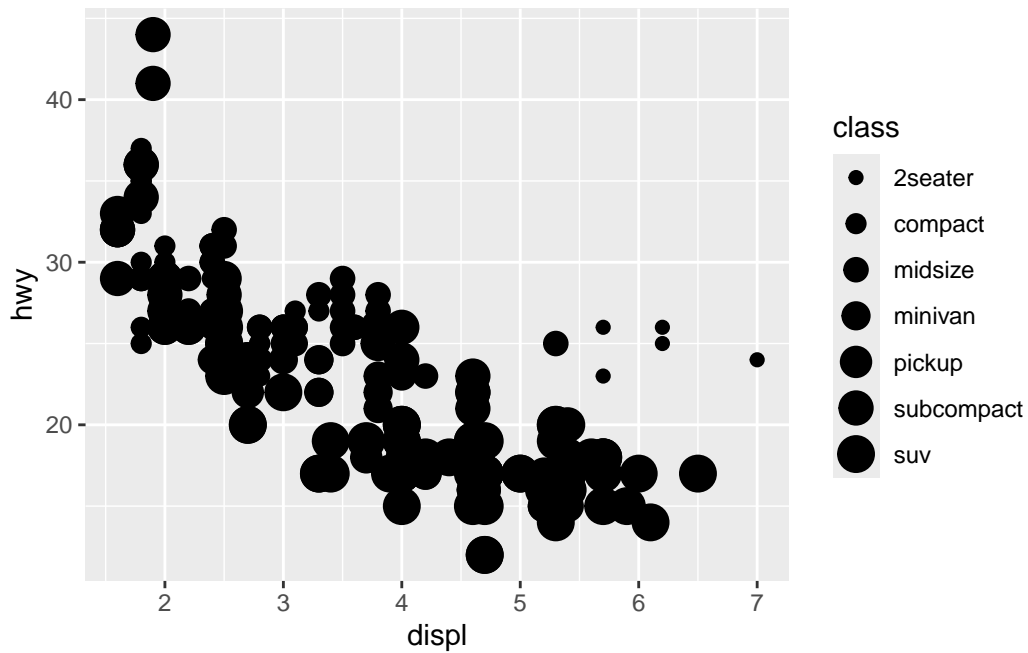
```
library(tidyverse)

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



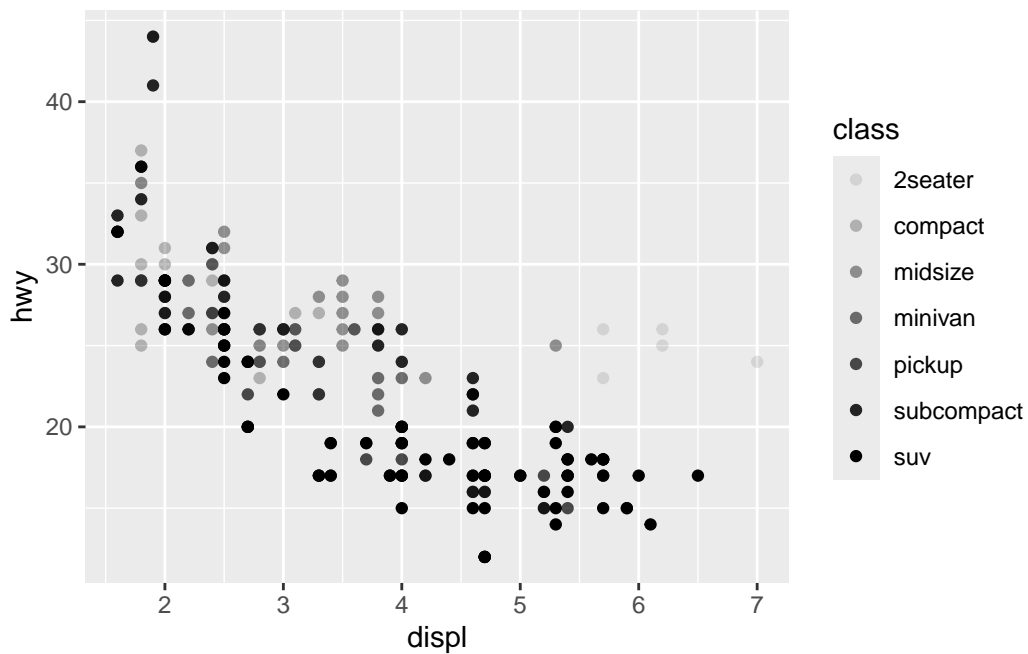
- attributing a variable to a characteristic outside the coordinate axis is called *scaling*.
- I can do scaling with the size of the points:

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



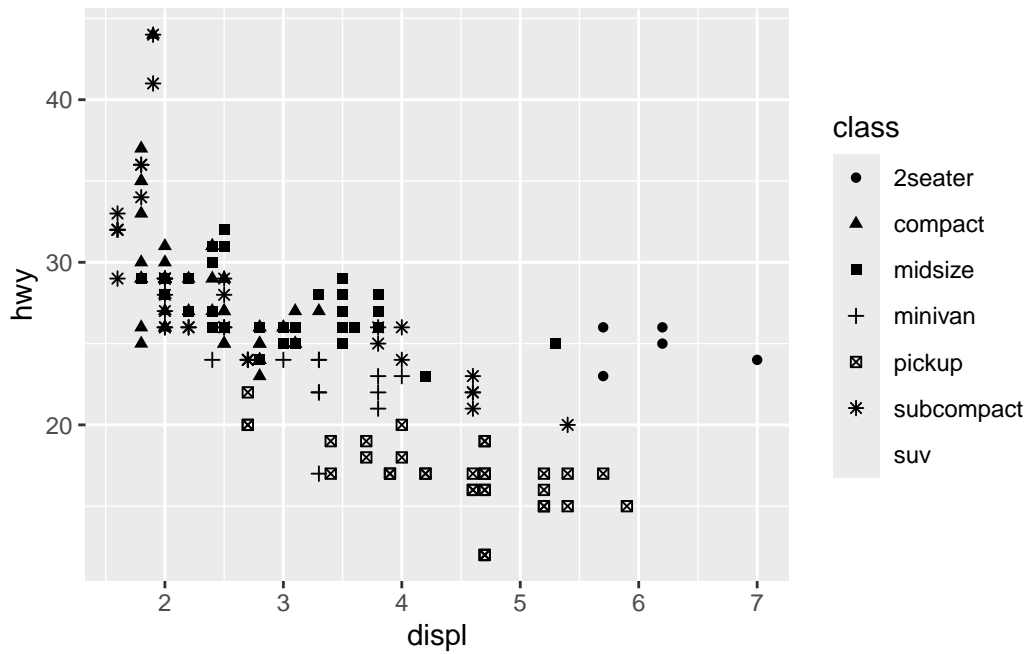
- Or, I can scale to the transparency with alpha:

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



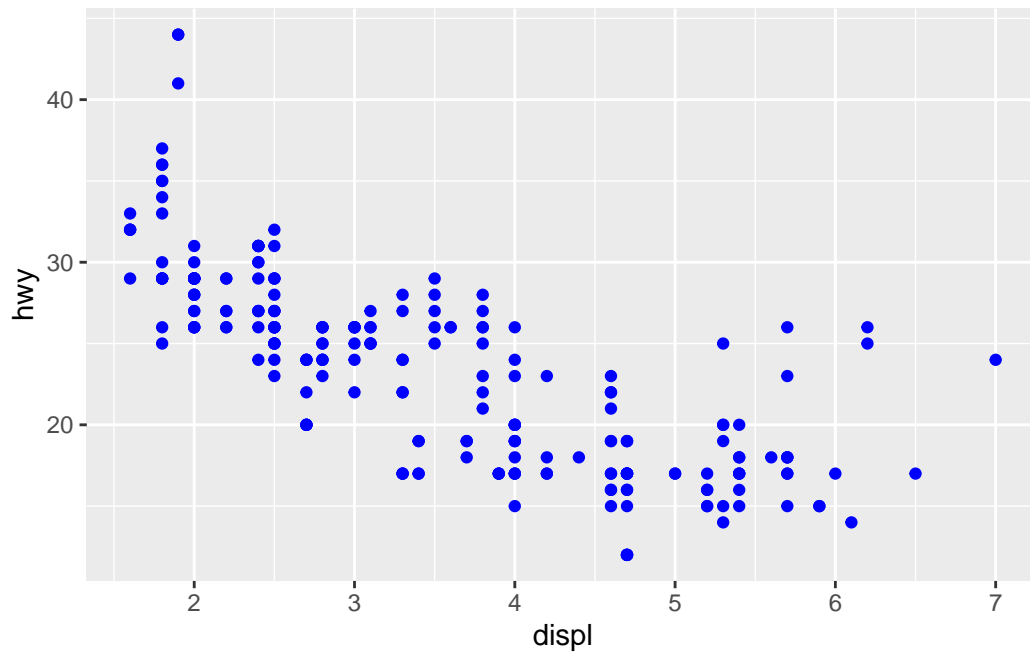
- Or to the shape of the objects:

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



- If it is to my taste, I can choose to color all the points blue (remembering to put the argument `color = "blue"` outside the argument `aes`):

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



## Exercises

---

- Consider the code and the graph:

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



Q: Why aren't the points blue?

A: Because the argument `color` is inside the argument `aes`.

Q: Which **mpg** variables are category? and which are continuous? How can you acquire this information executing **mpg**?

A:

```
head(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(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~
```

Manufacture, model, transmission, drive train, fuel type and class are catategory (considering that by continuous the autor means quantitative, or else only city mpg and higway mpg would be continuous).

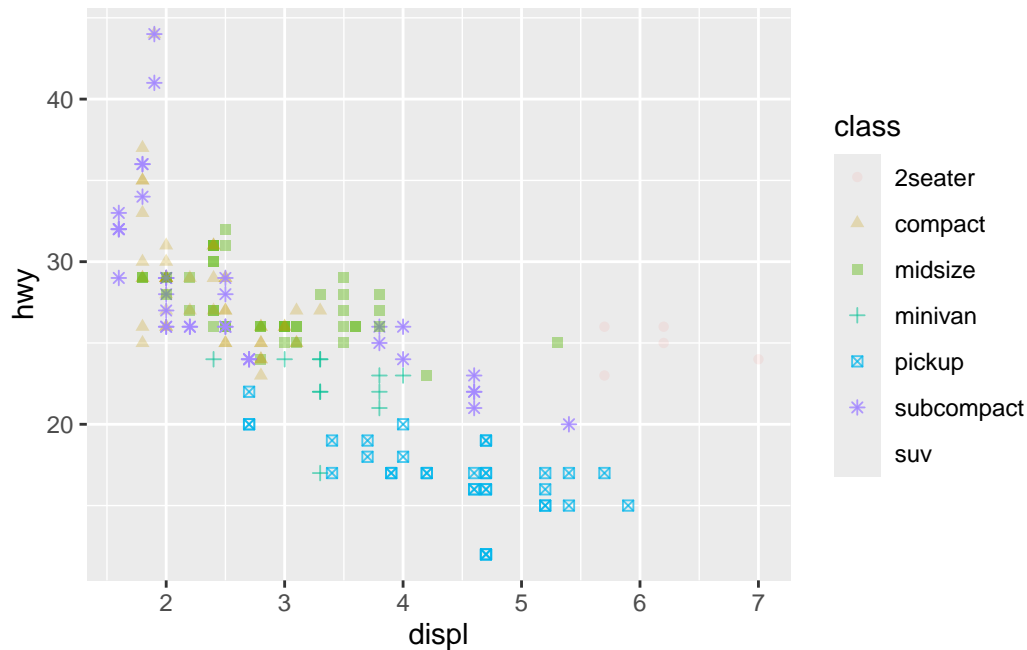
Engine displacement, year, number of cylinders, city mpg and highway mpg are continuous.

When executing `mpg`, it's shown each variable class.

*Q: What happens if you map the same variable to various aesthetics?*

A:

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



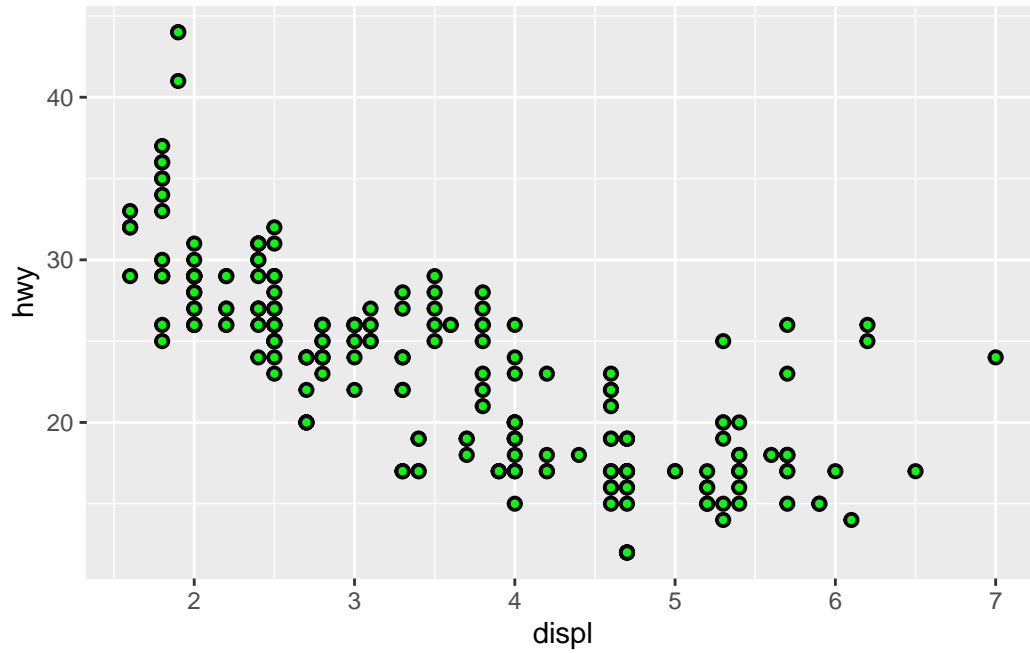
It becomes sort of confuse.

\*Q: How the `stroke` aesthetic works?

A:

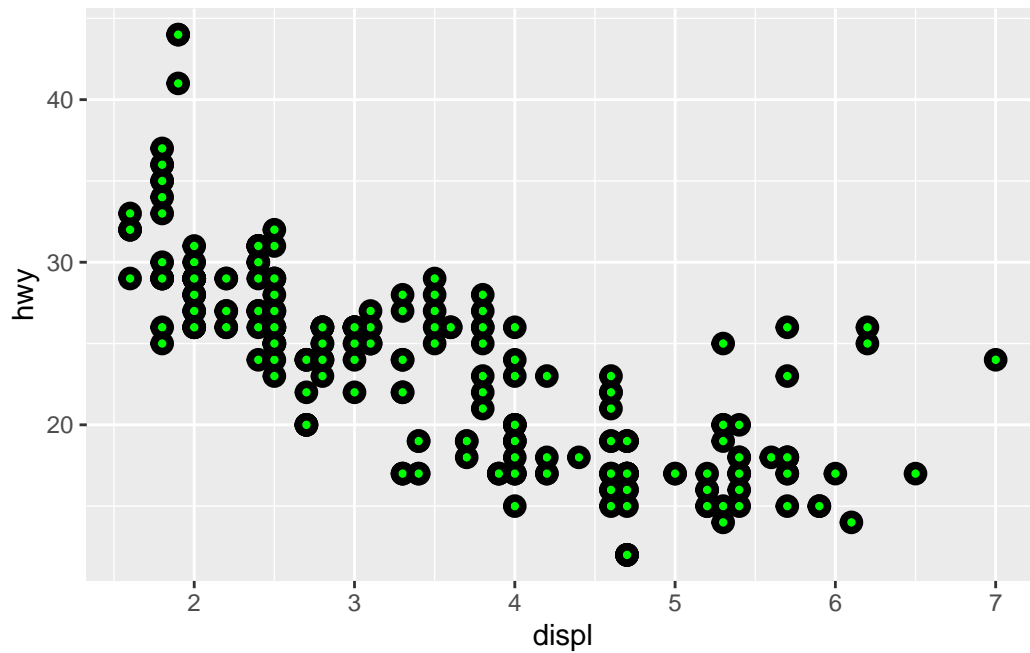
```
vignette("ggplot2-specs") #gives me the aesthetic specification  
  
ggplot(data = mpg) +  
  geom_point(  
    mapping = aes(x = displ, y = hwy),  
    shape = 21,
```

```
fill = "green",  
stroke = 1  
)
```



```
ggplot(data = mpg) +  
  geom_point(  
    mapping = aes(x = displ, y = hwy),  
    shape = 21,  
    fill = "green",  
    stroke = 2  
  )
```



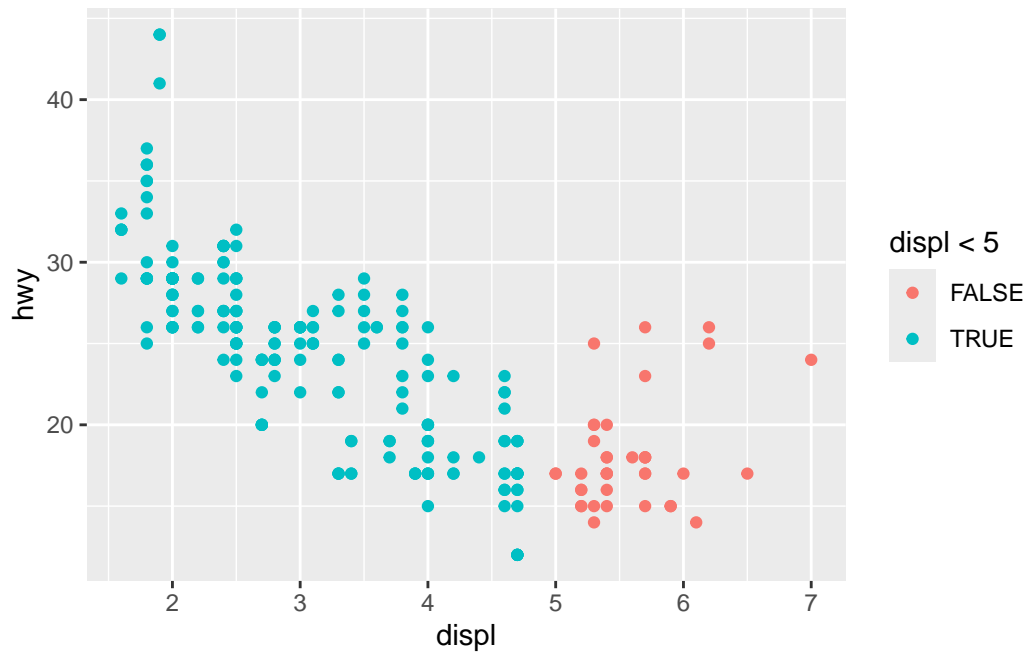


It controls the border width in shapes that possesses it.

*Q: what happens if you map an aesthetic to something other than a variable's name? Such as `aes(color = displ < 5)`*

A:

```
ggplot(data = mpg) +  
  geom_point(  
    mapping = aes(x = displ, y = hwy, color = displ < 5)  
  )
```



It qualifies the points to the values as I specified in the command.