Visualize

Load the library

```
require(ggplot2)
# or
require(tidyverse)
```

Layers

Common layers

- `ggplot` - base
- `aes` - x-axis, y-axis, color, group, size, ...
- Geoms - shapes (points, lines, histogram, ...)
- Scales, axis settings, color palletes, labeling
- Facets
- Themes

Combine layers by adding (`+`), not pipes.

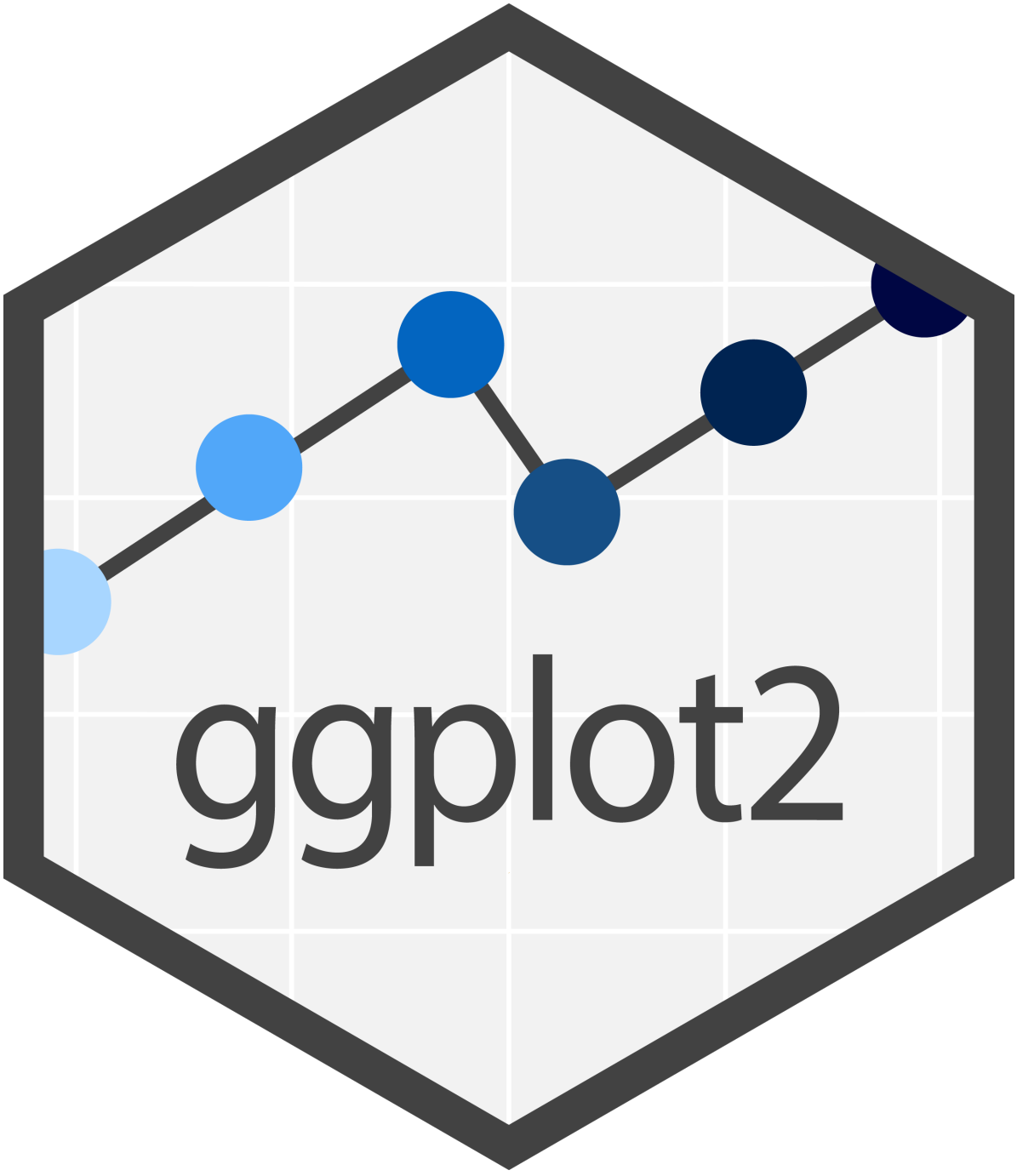

ggplot2

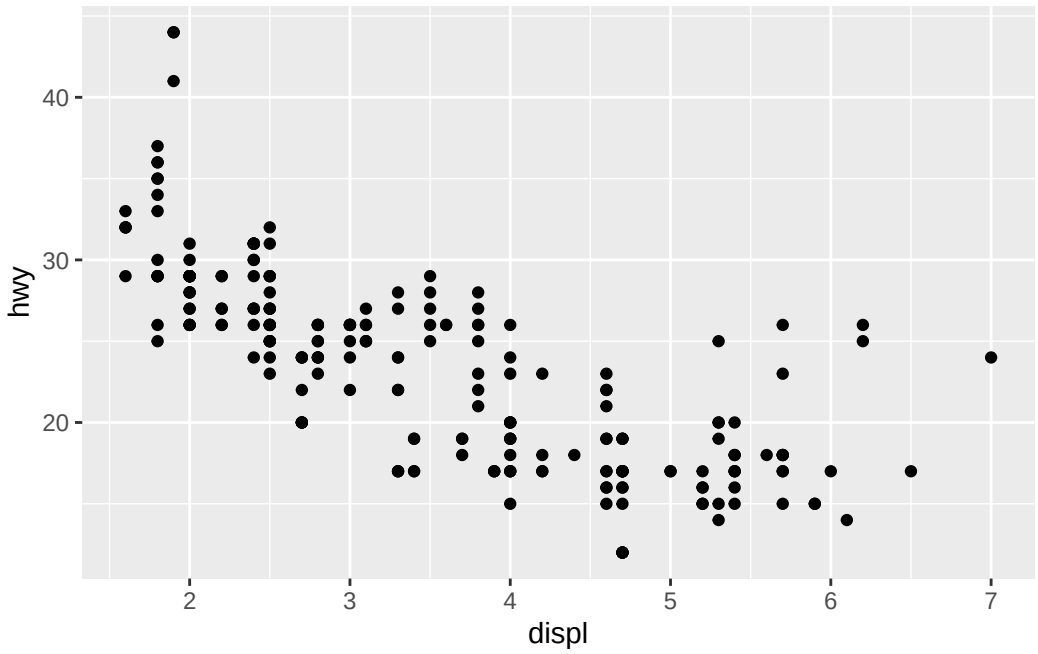
Table `mpg`

```
# A tibble: 234 x 9
   manufacturer model      displ  year   cyl trans      drv     cty   hwy
   <chr>        <chr>      <dbl> <int> <int> <chr>      <chr> <int> <int>
 1 audi         a4           1.8  1999     4 auto(l5)   f        18    29
 2 audi         a4           1.8  1999     4 manual(m5) f        21    29
 3 audi         a4           2    2008     4 manual(m6) f        20    31
 4 audi         a4           2    2008     4 auto(av)   f        21    30
 5 audi         a4           2.8  1999     6 auto(l5)   f        16    26
 6 audi         a4           2.8  1999     6 manual(m5) f        18    26
 7 audi         a4           3.1  2008     6 auto(av)   f        18    27
 8 audi         a4 quattro   1.8  1999     4 manual(m5) 4        18    26
 9 audi         a4 quattro   1.8  1999     4 auto(l5)   4        16    25
10 audi         a4 quattro   2    2008     4 manual(m6) 4        20    28
# ... with 224 more rows
```

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

- use data frame `mpg` as source data
- draw points
- map column `displ` to x-axis
- map column `hwy` to y-axis

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



Exercise 1

Run this code to make a graph. Pay strict attention to spelling, capitalization and parentheses.

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

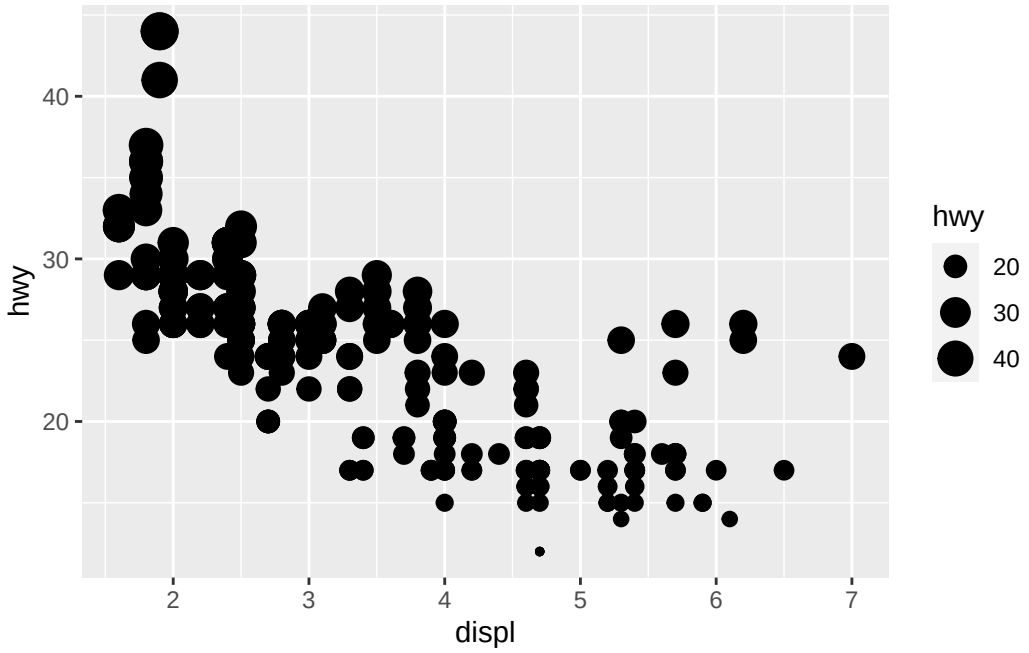
Mappings

Common aesthetics

- size
- color
- shape (e.g. points)
- width (e.g. lines)
- fill (e.g. bars)
- alpha

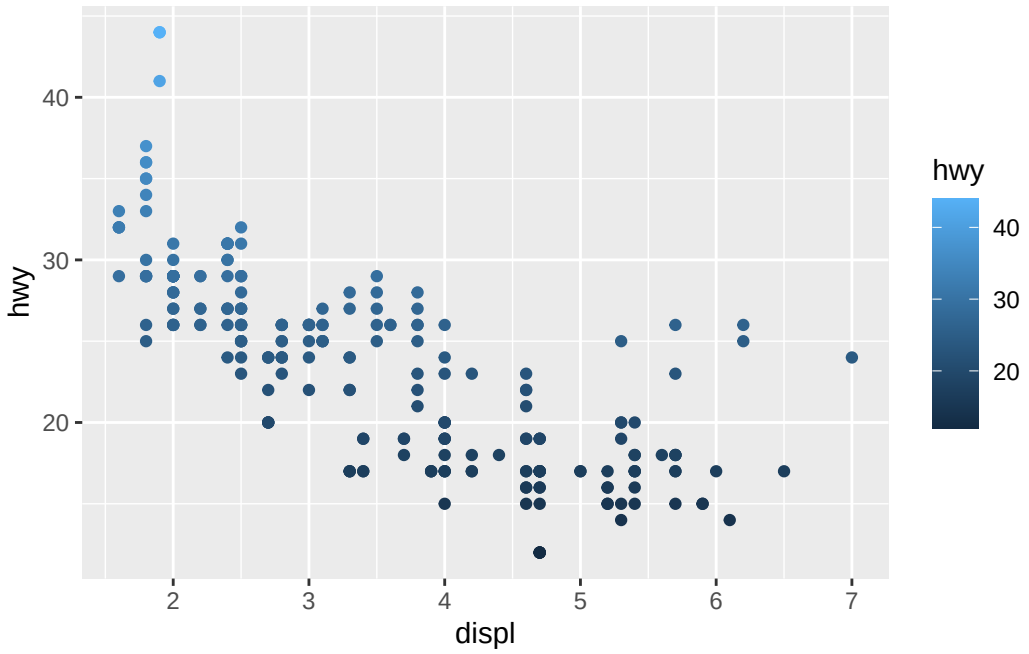
Size - continous

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



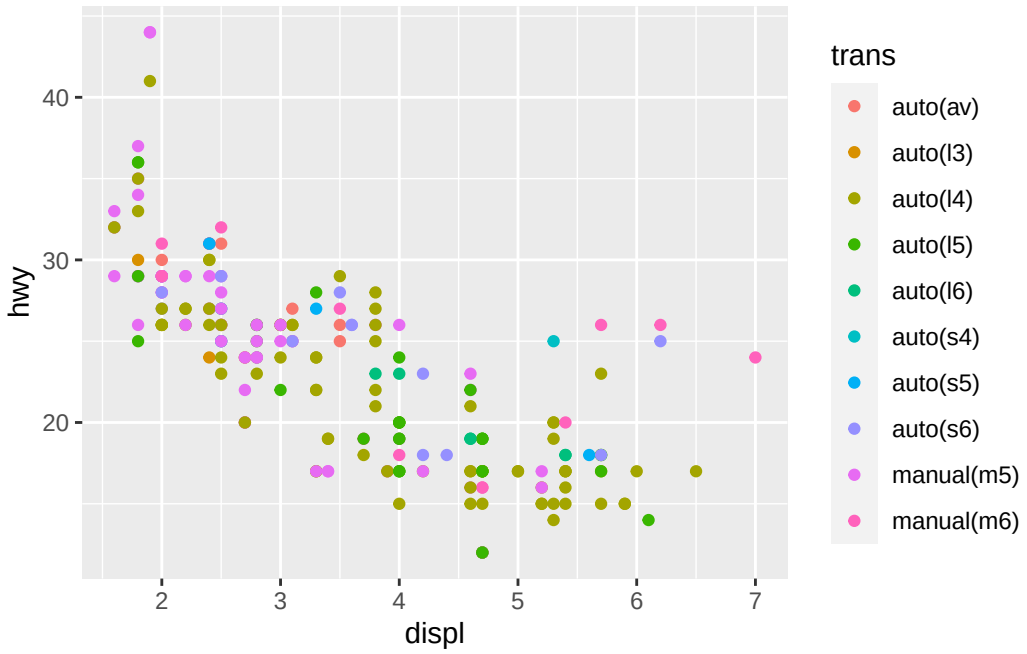
Color - continuous

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



Color - discrete

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

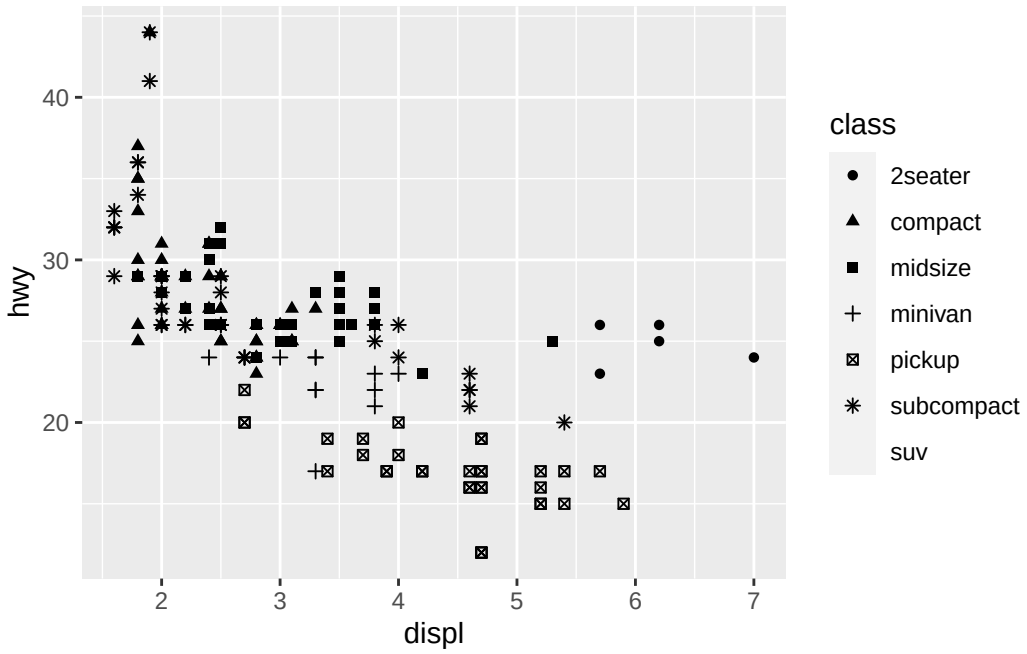


Shape - discrete

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

```
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).
```



note the missing shape for suv

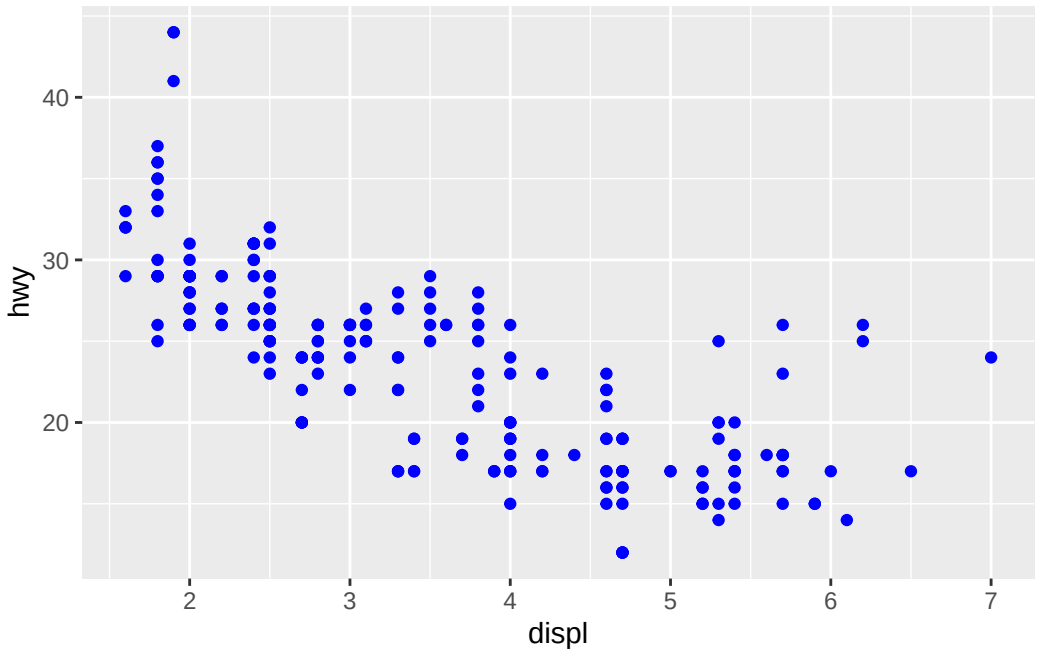
Data vs Explicit

```
geom_SHAPE(mapping = aes(COLUMN NAMES), EXPLICIT VALUES)
```

- use column names inside `aes()`
- use values after comma

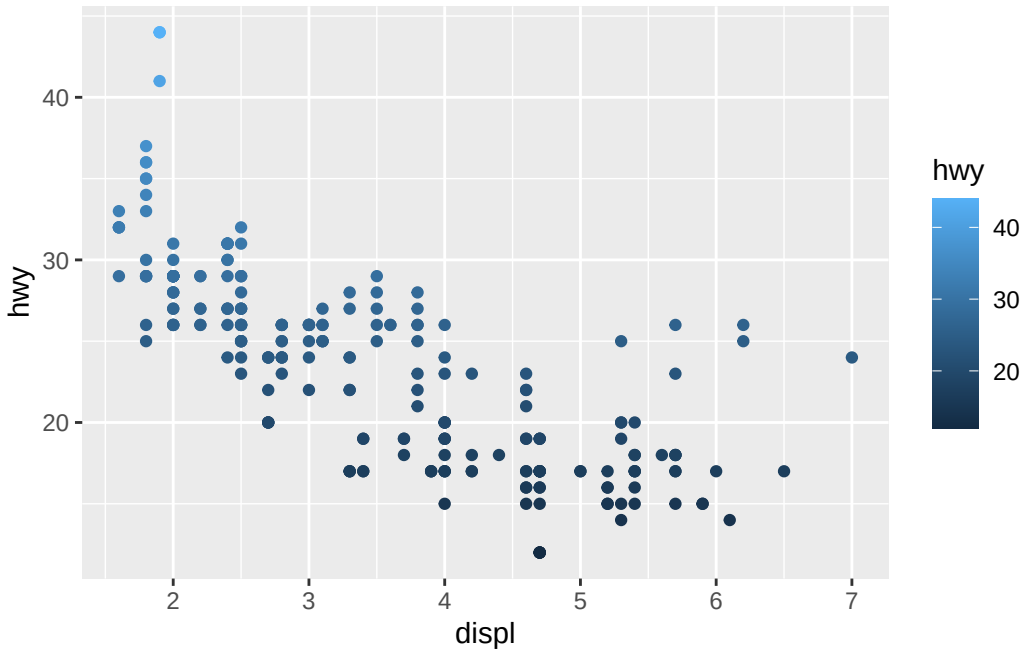
Explicit color

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



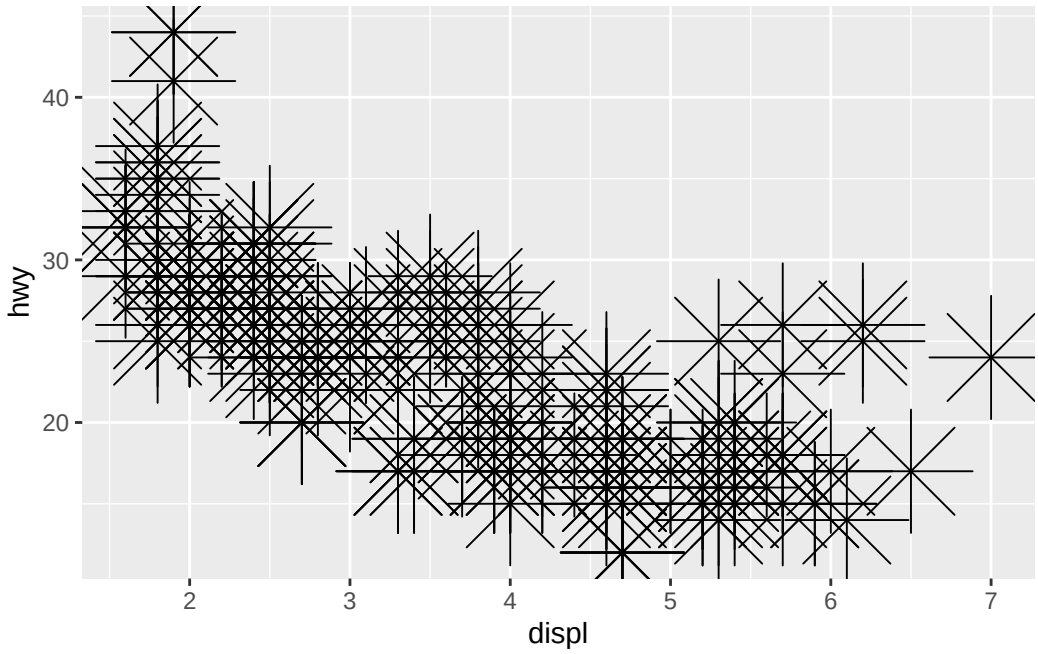
Color inferred from data

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



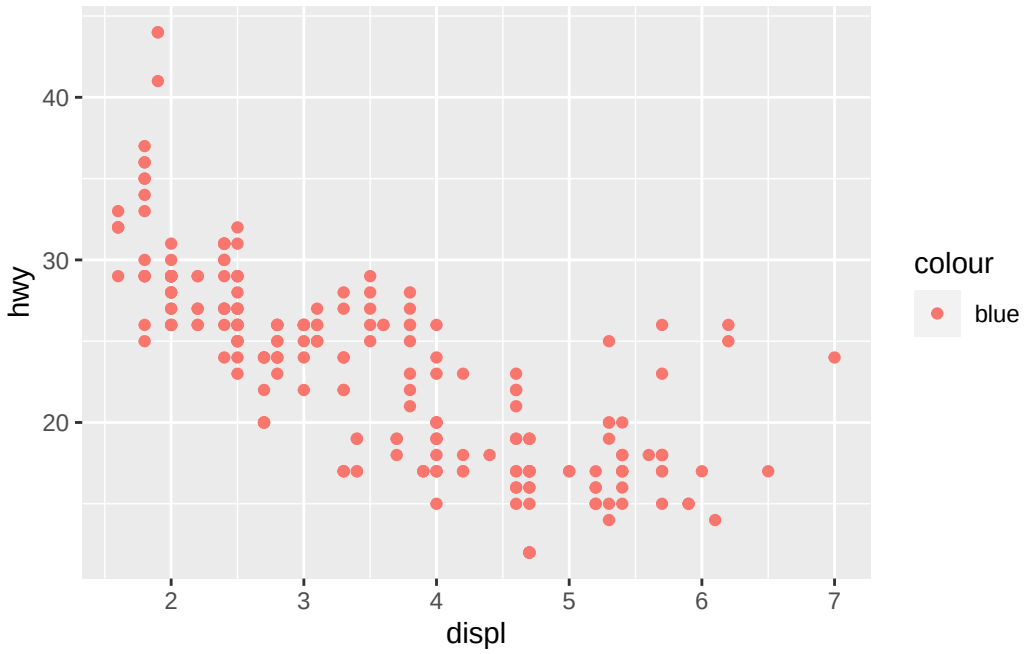
Explicit shape & size

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), shape = 8, size = 15)
```



Common mistake

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



Exercise 2

In the next chunk, add color, size, alpha, and shape aesthetics to your graph. Experiment. Do different things happen when you map aesthetics to discrete and continuous variables? What happens when you use more than one aesthetic?

Facets

Exercise 3

What is the difference between `facet_grid` and `facet_wrap`?

Hint: use `ctrl+shift+c` to comment / uncomment a line.

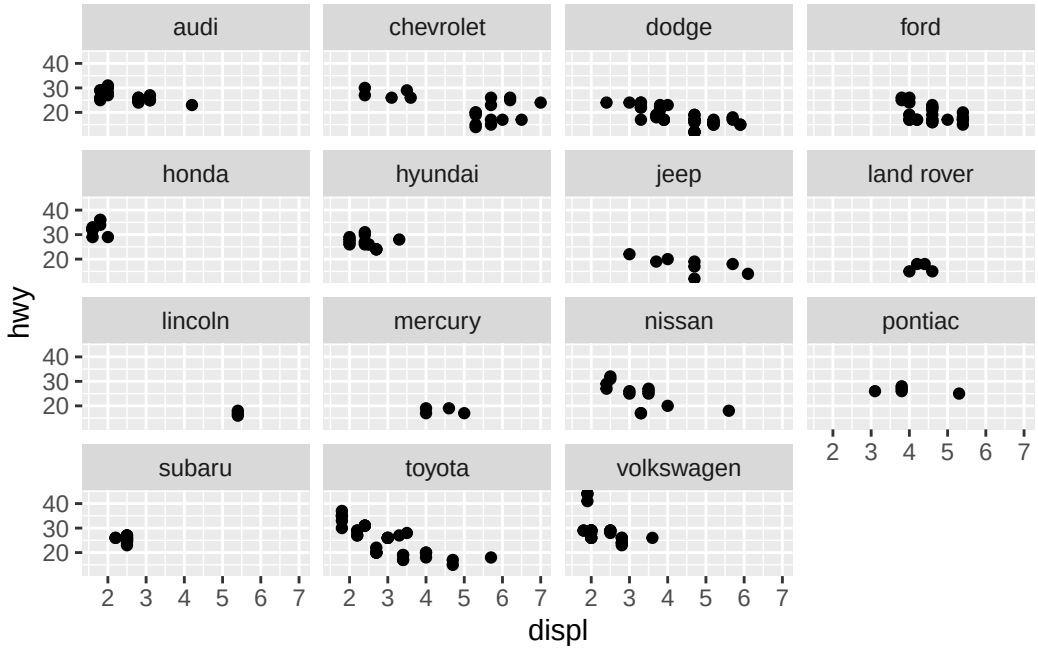
`facet_grid`:

- 2D grid (rows and/or columns)

`facet_wrap`:

- 1D ribbon wrapped into 2D
- can specify dimensions

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



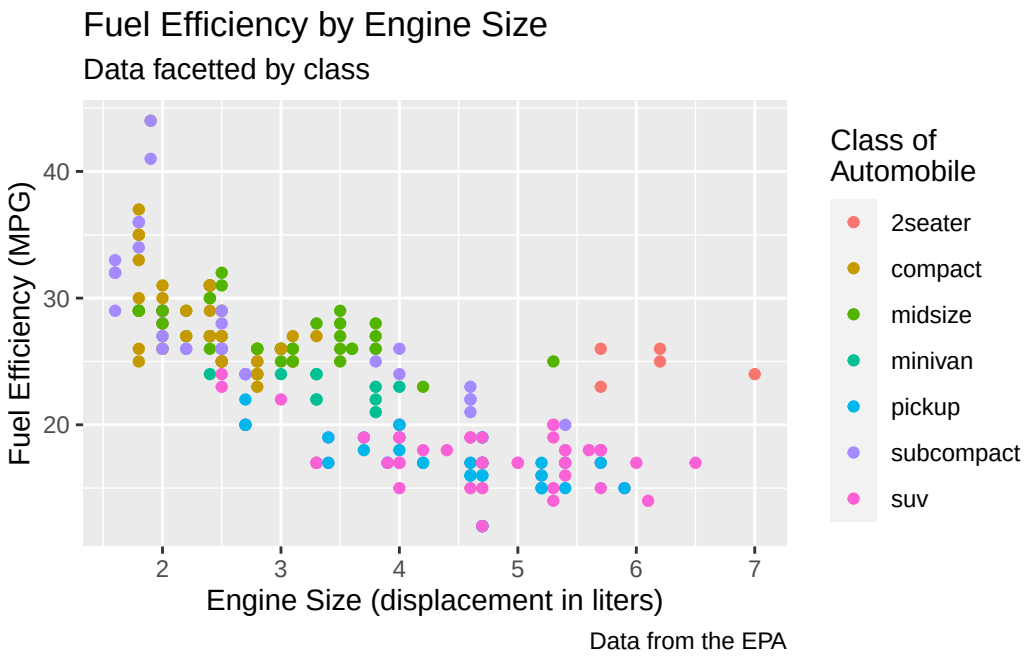
Labels

Exercise 4

Add labels to your graph.

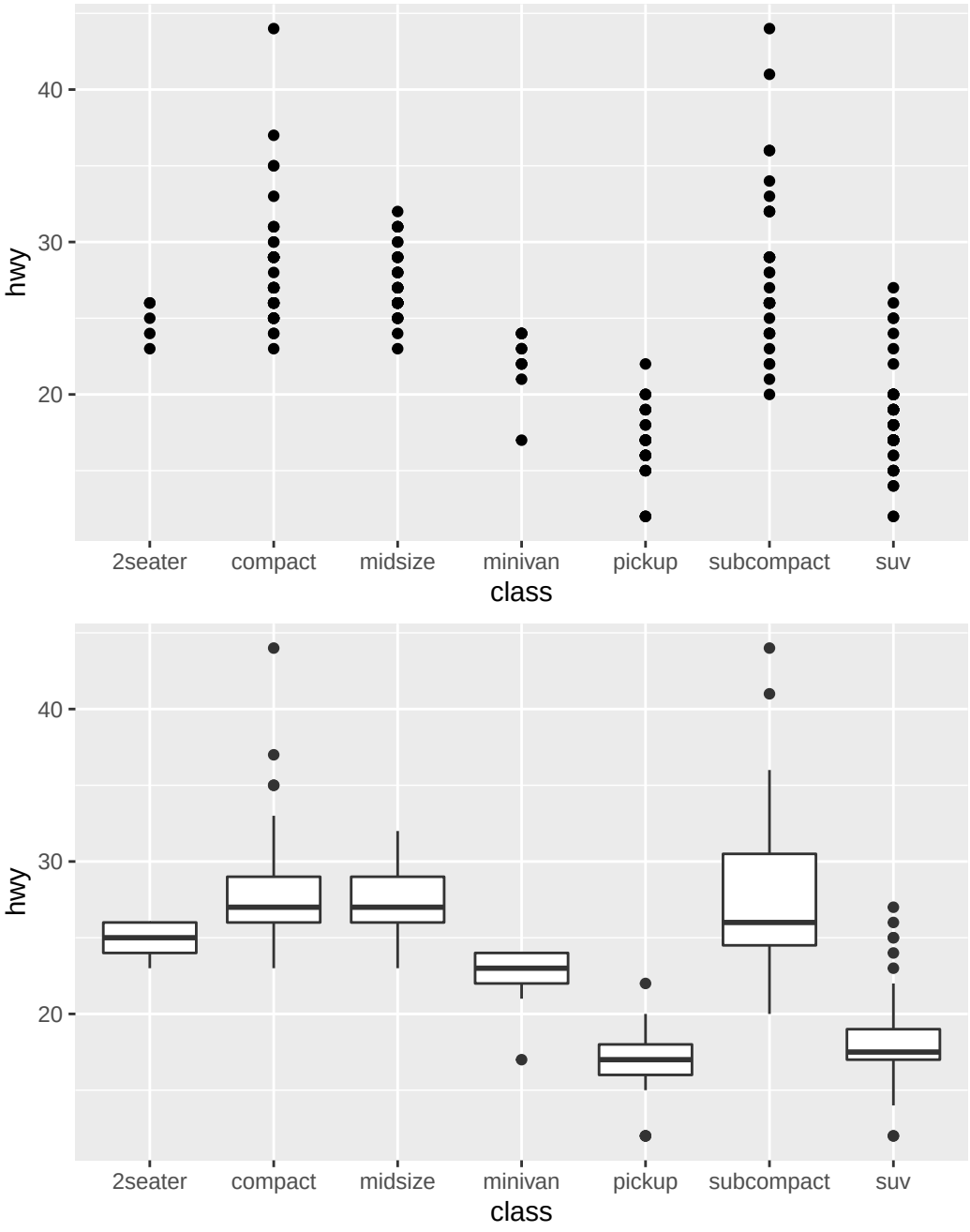
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class)) +
  labs(title = "Fuel Efficiency by Engine Size",
       subtitle = "Data facetted by class",
       x = "Engine Size (displacement in liters)",
       y = "Fuel Efficiency (MPG)",
```

```
        color = "Class of\nAutomobile",
        caption = "Data from the EPA")
```



Geoms

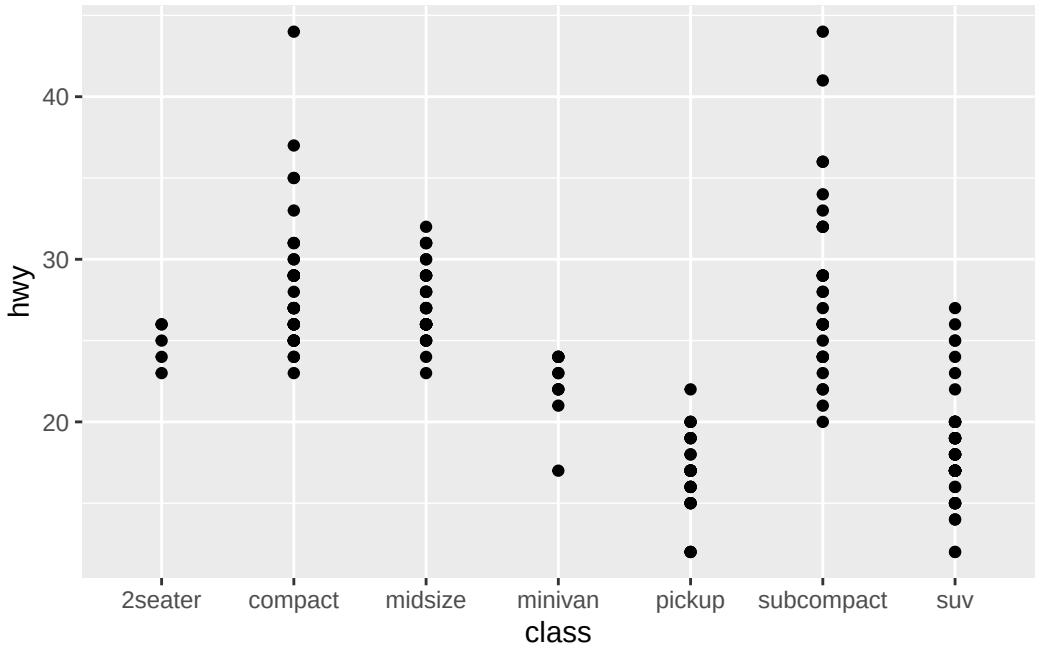
What is the difference?



Same x-axis and y-axis; different chart

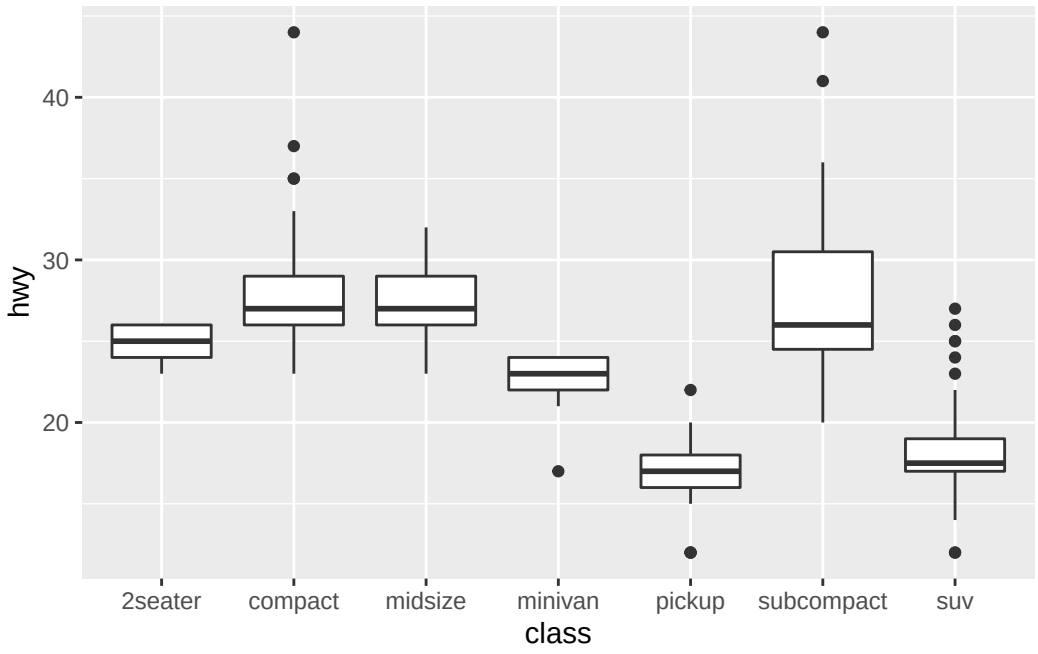
geom_point

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



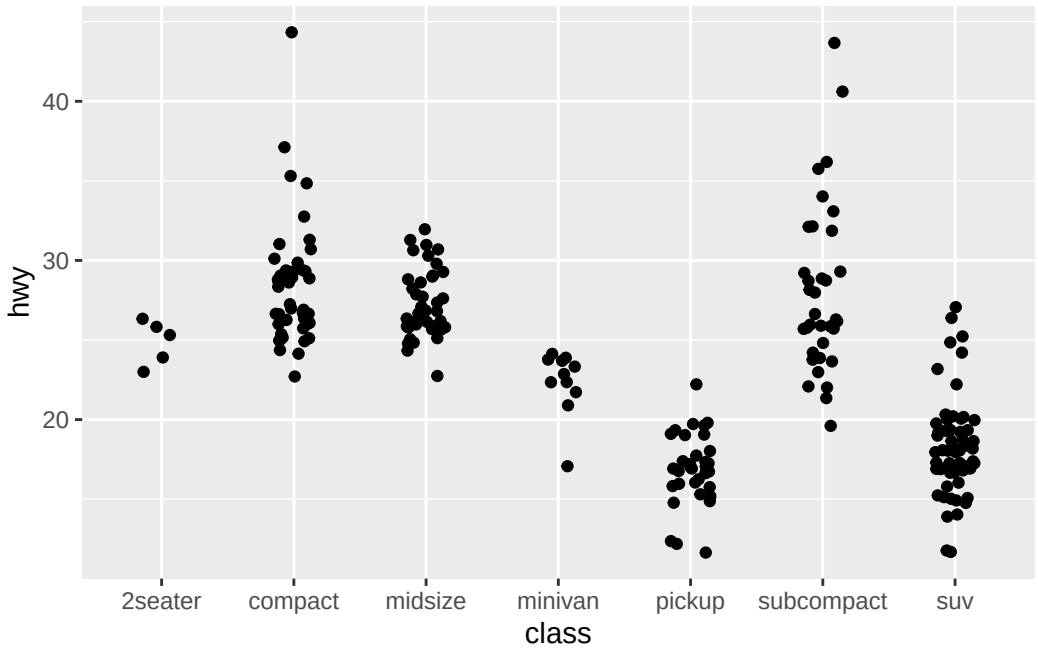
geom_boxplot

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



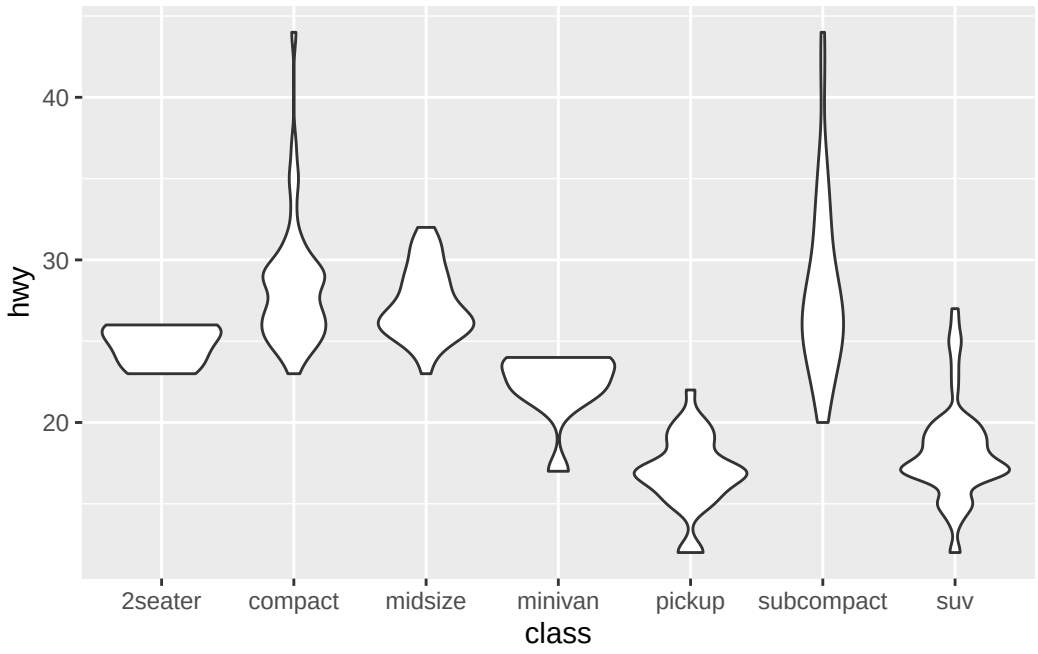
geom_jitter

```
ggplot(data = mpg) +
  geom_jitter(mapping = aes(x = class, y = hwy), width = 0.15)
```



geom_violin

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



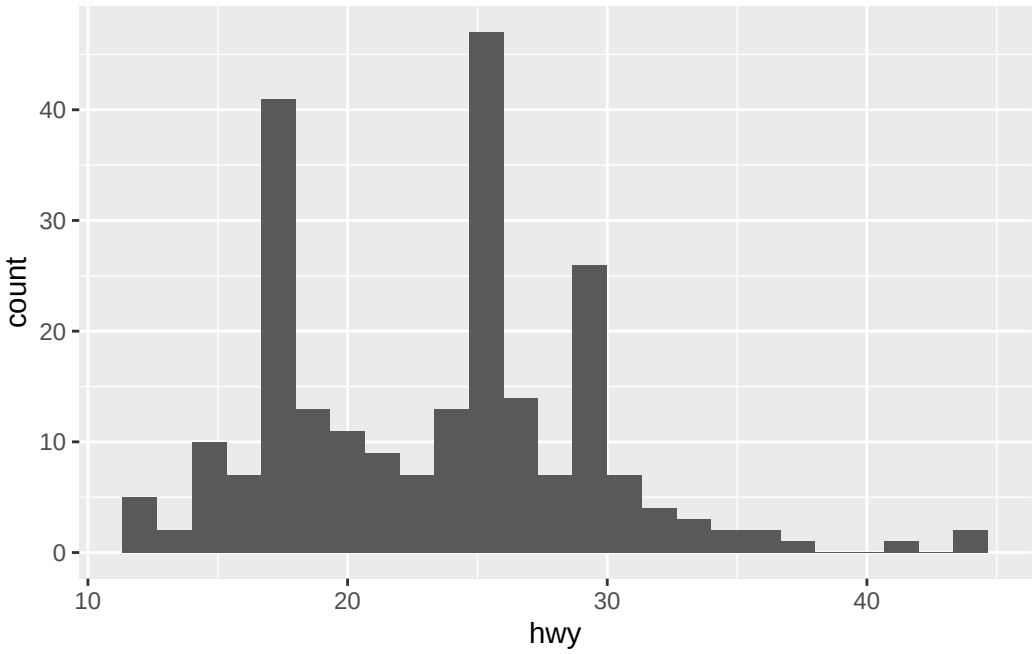
Common geom functions

- `geom_point`
- `geom_line`
- `geom_bar`
- `geom_histogram`
- `geom_boxplot`

Not all geoms use the same mappings!

`geom_histogram`

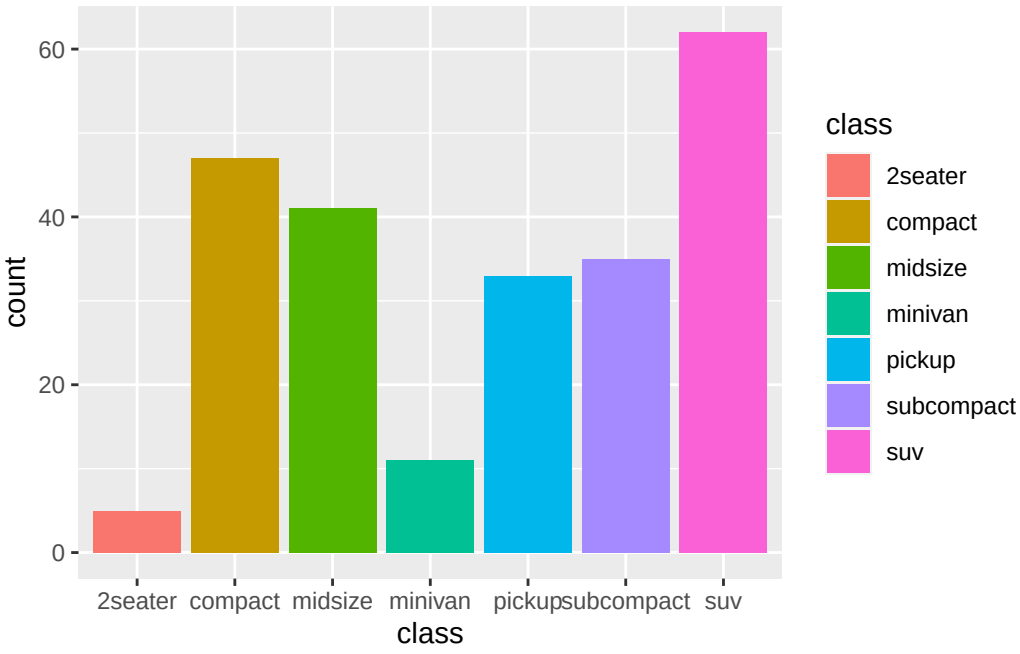
```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = hwy), bins=25)
```



Note that `y` is missing

Exercise 5

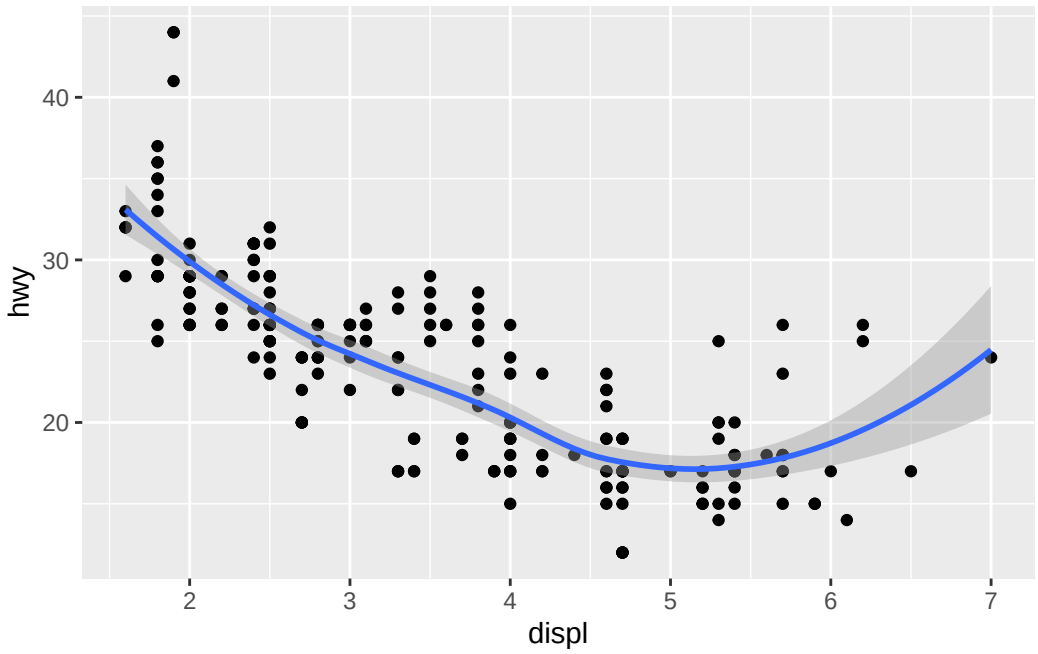
Make a chart that shows number of vehicles per class. Use different colors for each bar.



Combine geoms I

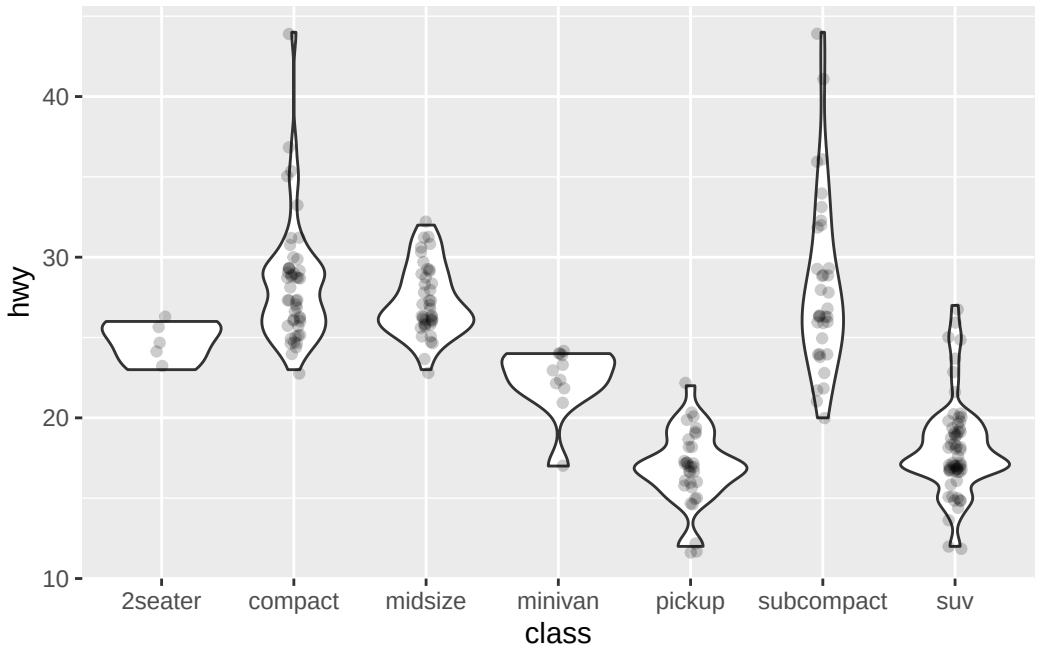
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

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



Combine geoms II

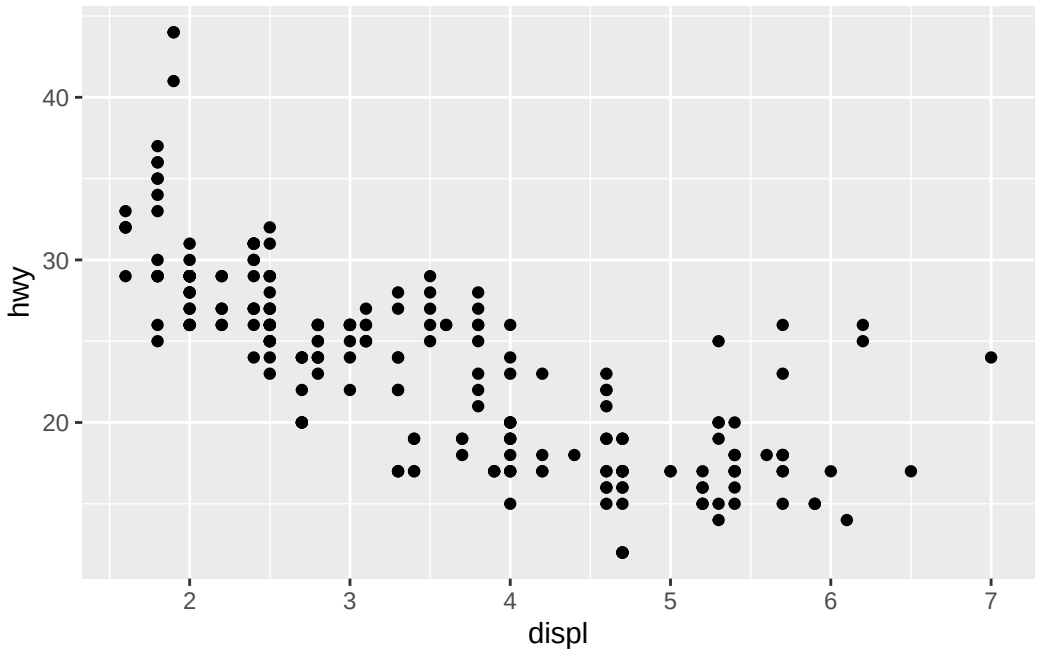
```
ggplot(data = mpg, ) +
  geom_violin(mapping = aes(x = class, y = hwy)) +
  geom_jitter(mapping = aes(x = class, y = hwy), width = 0.05, alpha = 0.2)
```



Global vs. Local

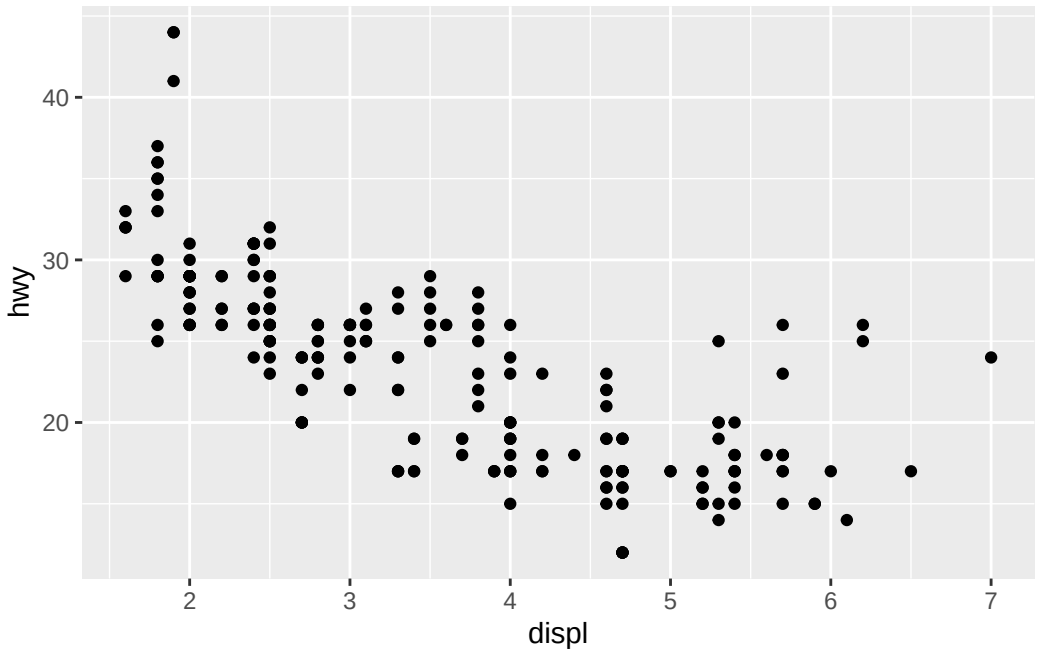
Local mapping - within `geom_point`

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



Global mapping - within `ggplot`

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



Mapping types

Global

- All geoms use it

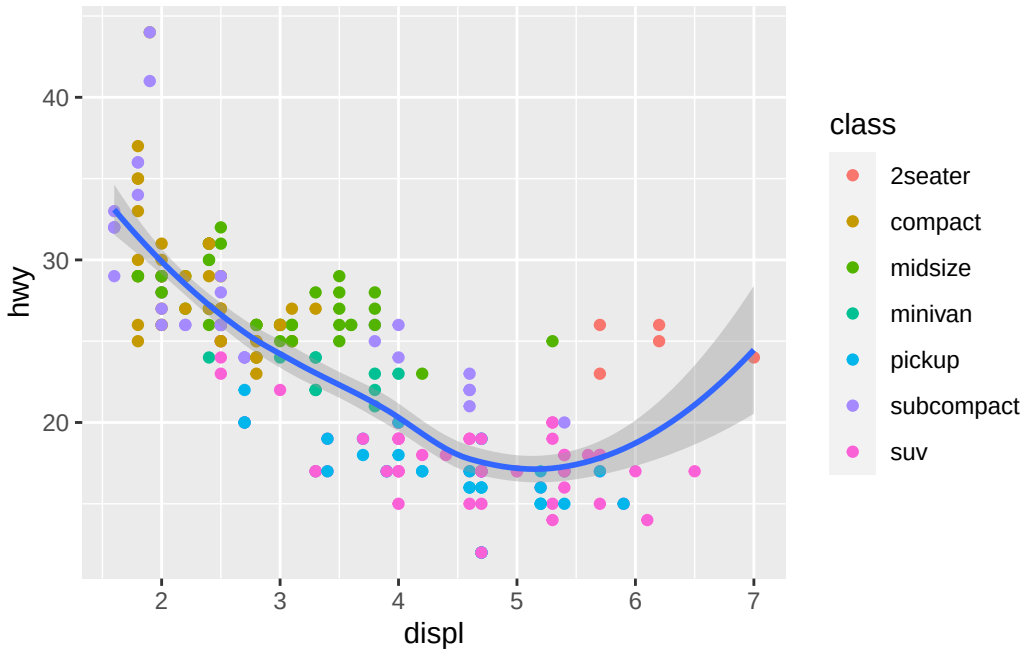
Local

- Only specified geom uses it

Mix global & local

```
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'
```



x, y shared; color only for points; single smoothing line

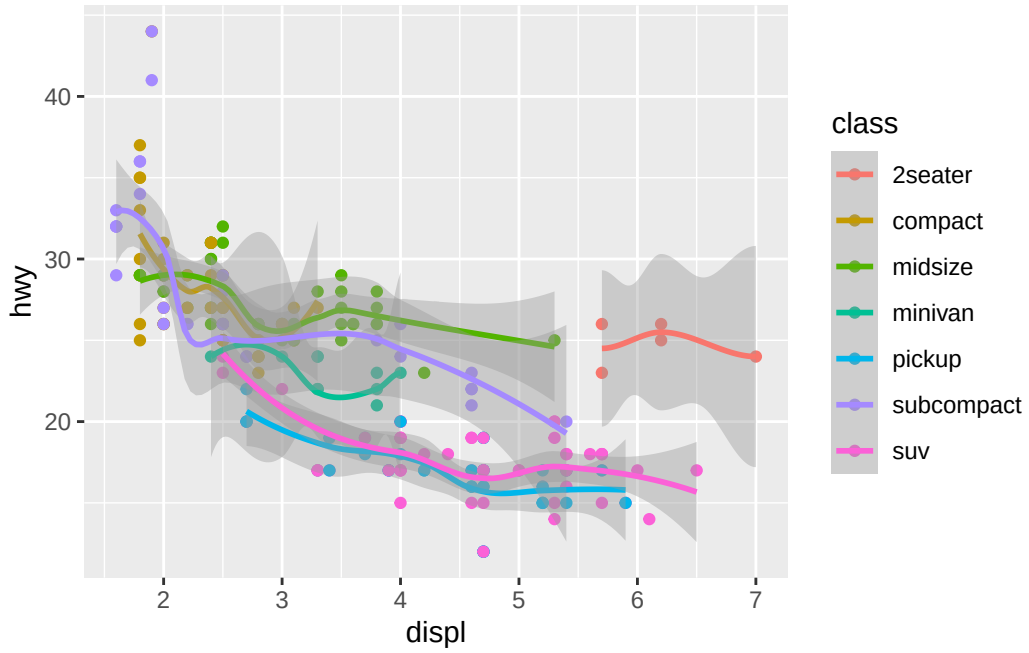
Using only global

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

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

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : span too small. fewer data values than degrees of freedom.

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : pseudoinverse used at 5.6935

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : neighborhood radius 0.5065

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : reciprocal condition number 0
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : There are other near singularities as well. 0.65044

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : span too small. fewer
data values than degrees of freedom.

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
5.6935

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
0.5065

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
number 0

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : There are other near
singularities as well. 0.65044

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : pseudoinverse used at 4.008

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : neighborhood radius 0.708

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : reciprocal condition number 0

Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, : There are other near singularities as well. 0.25
```

```
Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
4.008

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
0.708

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
number 0

Warning in predLoess(object$y, object$x, newx = if
(is.null(newdata)) object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), : There are other near
singularities as well. 0.25
```



x, y and color shared; several smoothing lines

Tips

Unnamed arguments

No need to name all arguments

```
ggplot(mpg, aes(displ, hwy, color = class)) +
  geom_point()
```

- first argument to `ggplot` is `data`
- second argument to `ggplot` is `mapping`
- first two arguments to `aes` are `x` and `y`
- first argument to `geom_*` is `mapping`

Save plots

```
p <- ggplot(mpg, aes(displ, hwy)) + geom_point()

# default size (7x7 inches)
ggsave("my_mpg_chart.png", p)

# specify size in pixels
ggsave("my_mpg_chart.png", p,
       width = 640, height = 480, units = "px")
```

... or right-click the image and select "Save image as..."

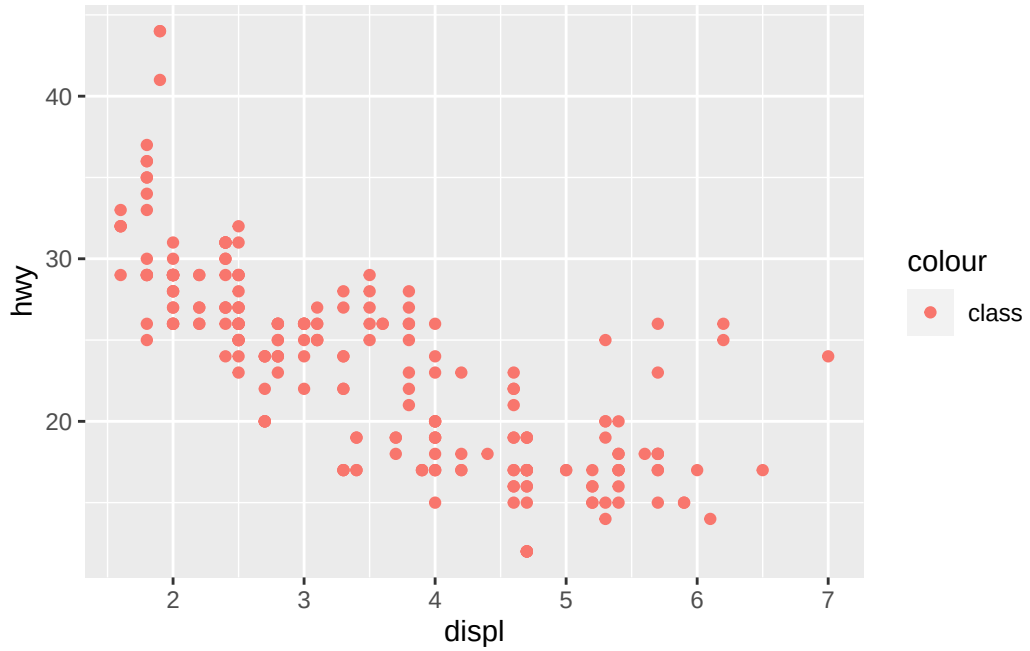
Interactive plots

```
# install.packages("plotly")
require(plotly)

p <- ggplot(mpg, aes(displ, hwy)) + geom_point()
ggplotly(p)
```

Themes

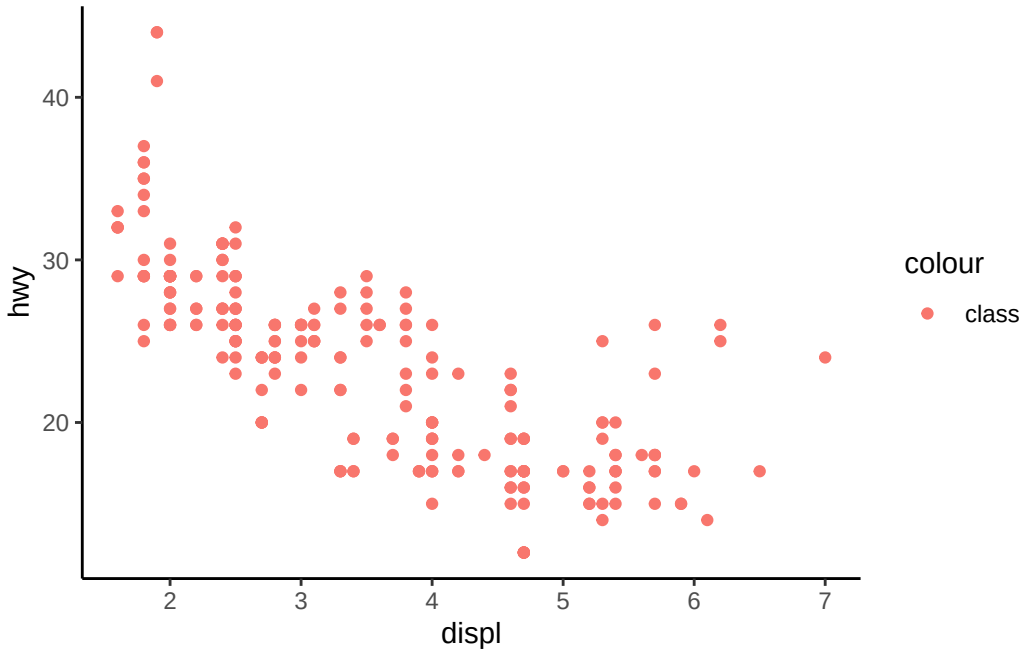
Default theme

```
ggplot(mpg, aes(displ, hwy, color = "class")) +
  geom_point() +
  theme_gray()  # default, changes nothing
```



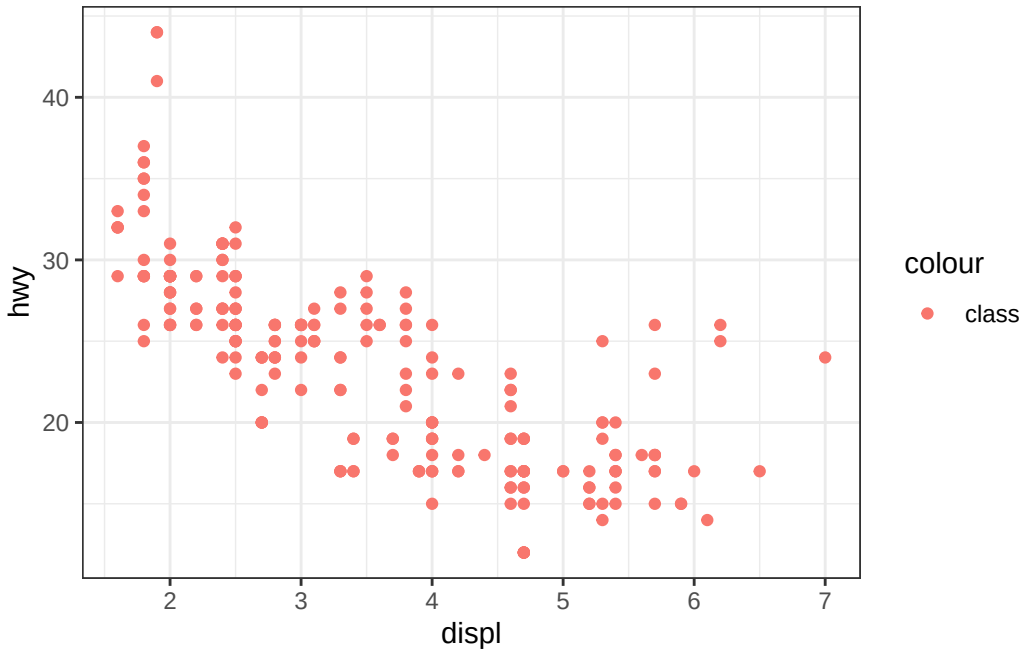
Change theme for a chart

```
ggplot(mpg, aes(displ, hwy, color = "class")) +
  geom_point() +
  theme_classic()
```



Change theme for entire notebook

```
theme_set(theme_bw())

ggplot(mpg, aes(displ, hwy, color = "class")) +
  geom_point()
```



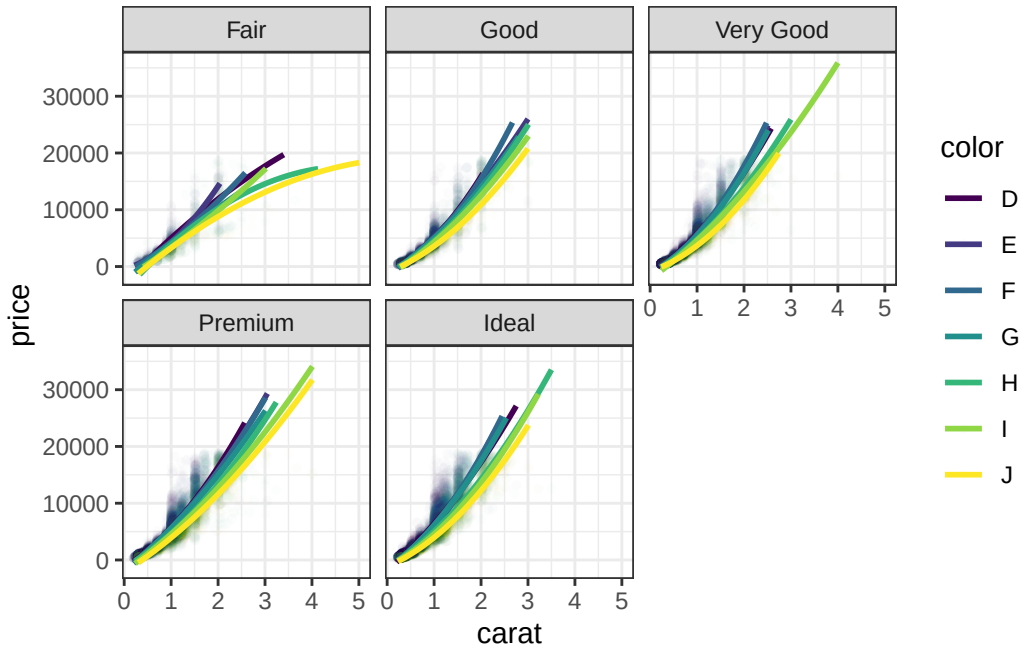
Warning: All following charts have this theme.

More themes

- Type `theme_` to see the list
- Use `theme()` to define manually
- Package `tvthemes` (Simpsons, Game of Thrones, ...)
- Write your own theme

What else?

```
ggplot(diamonds, aes(carat, price, color = color)) +
  geom_point(alpha = 0.01, shape = 16) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE) +
  facet_wrap(~ cut)
```



```
ggplot(iris, aes(Sepal.Length, Sepal.Width)) +
  geom_hex(bins=10) +
  coord_fixed()
```