Plotting

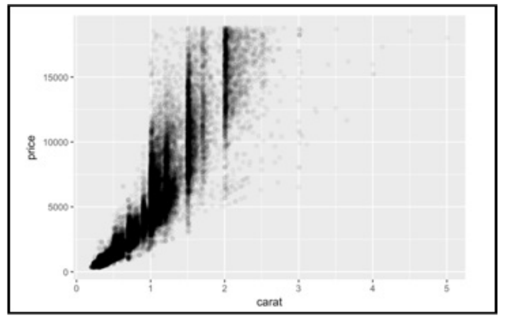
R for Data Science

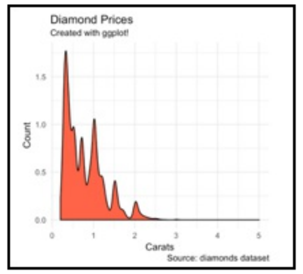
Basel R Bootcamp

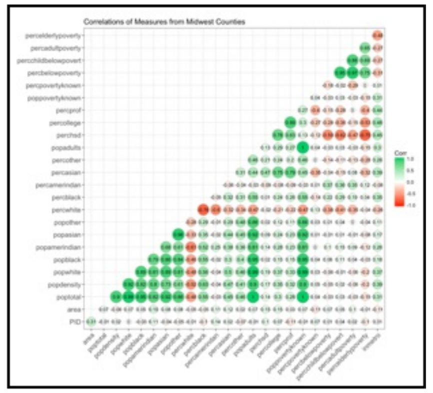
🕒 🏠 ✉ in

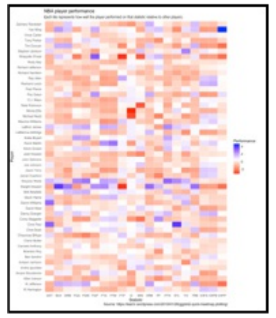
February 2019

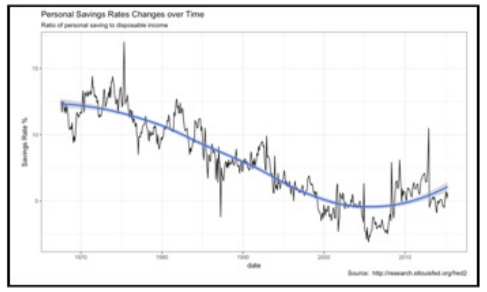
> As good as R is for statistics, it's as good if not better for data visualisation
>
> Nathaniel D. Phillips

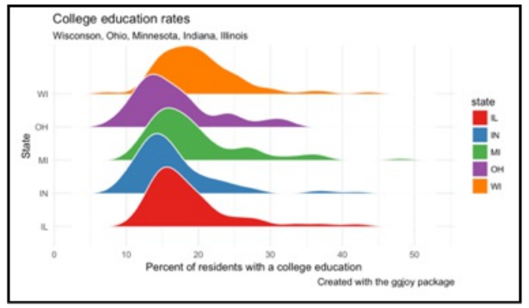


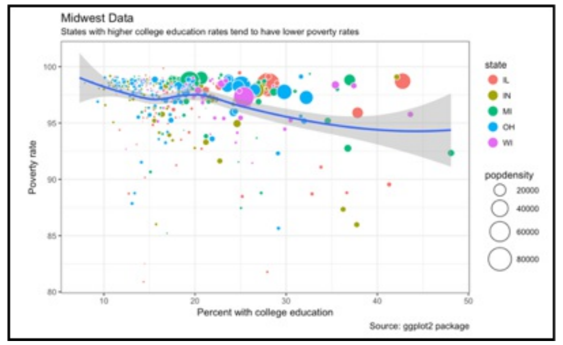












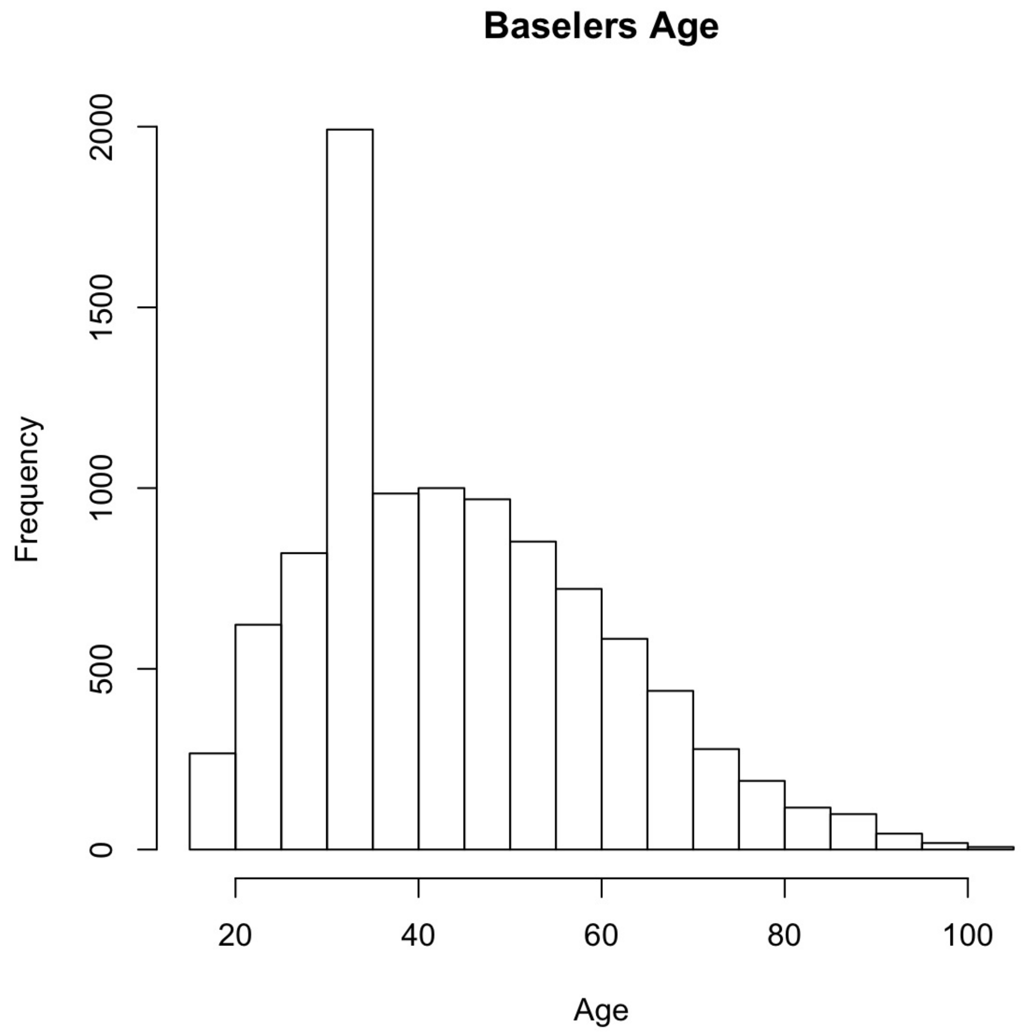
Base R Plotting

The **classic framework** of plotting.

Contains separate **function for each 'type'** of plot.

E.g. `barplot()` for a bar plot, `boxplot()` for a box plot, and `plot()` for a scatterplot.

```r
# Histogram in base R
hist(x = baselers$age,
     xlab = "Age",
     ylab = "Frequency",
     main = "Baselers Age")
```



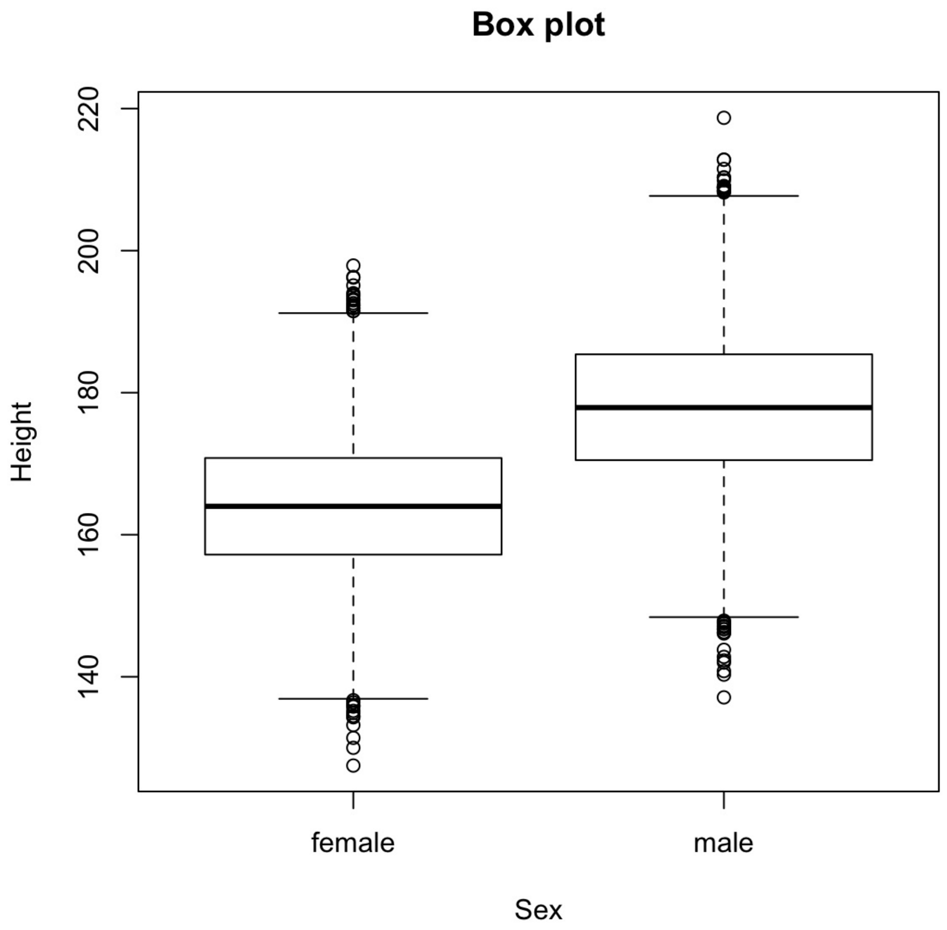
Base R Plotting

The **classic framework** of plotting.

Contains separate **function for each 'type'** of plot.

E.g. `barplot()` for a bar plot, `boxplot()` for a box plot, and `plot()` for a scatterplot.

```
# Boxplot in base R
boxplot(formula = height ~ sex,
        data = baselers,
        xlab = "Sex",
        ylab = "Height",
        main = "Box plot")
```



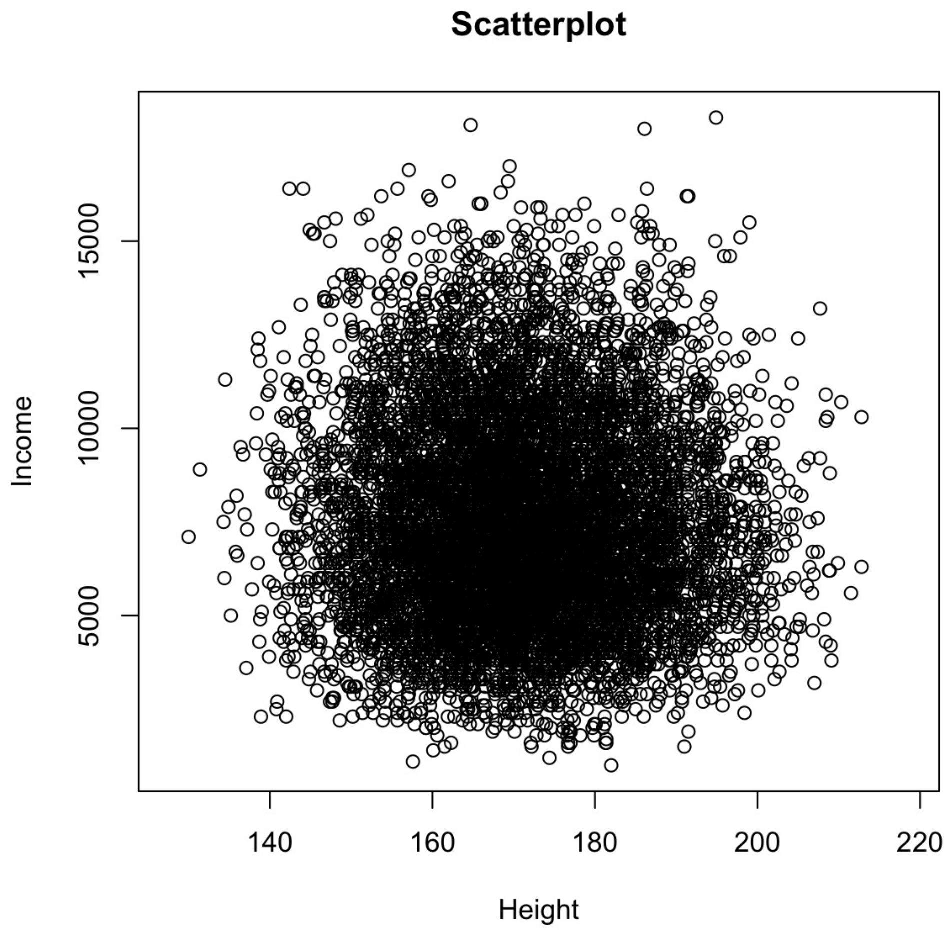
Base R Plotting

The **classic framework** of plotting.

Contains separate **function for each 'type'** of plot.

E.g. `barplot()` for a bar plot, `boxplot()` for a box plot, and `plot()` for a scatterplot.

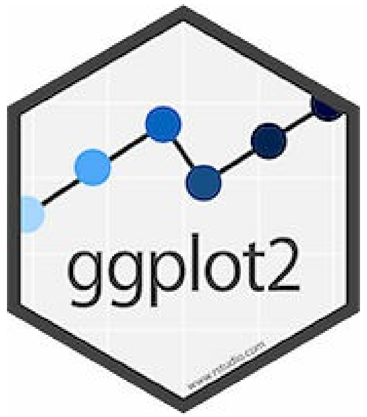
```
# Scatterplot in base R
plot(x = baselers$height,
     y = baselers$income,
     xlab = "Height",
     ylab = "Income",
     main = "Scatterplot")
```



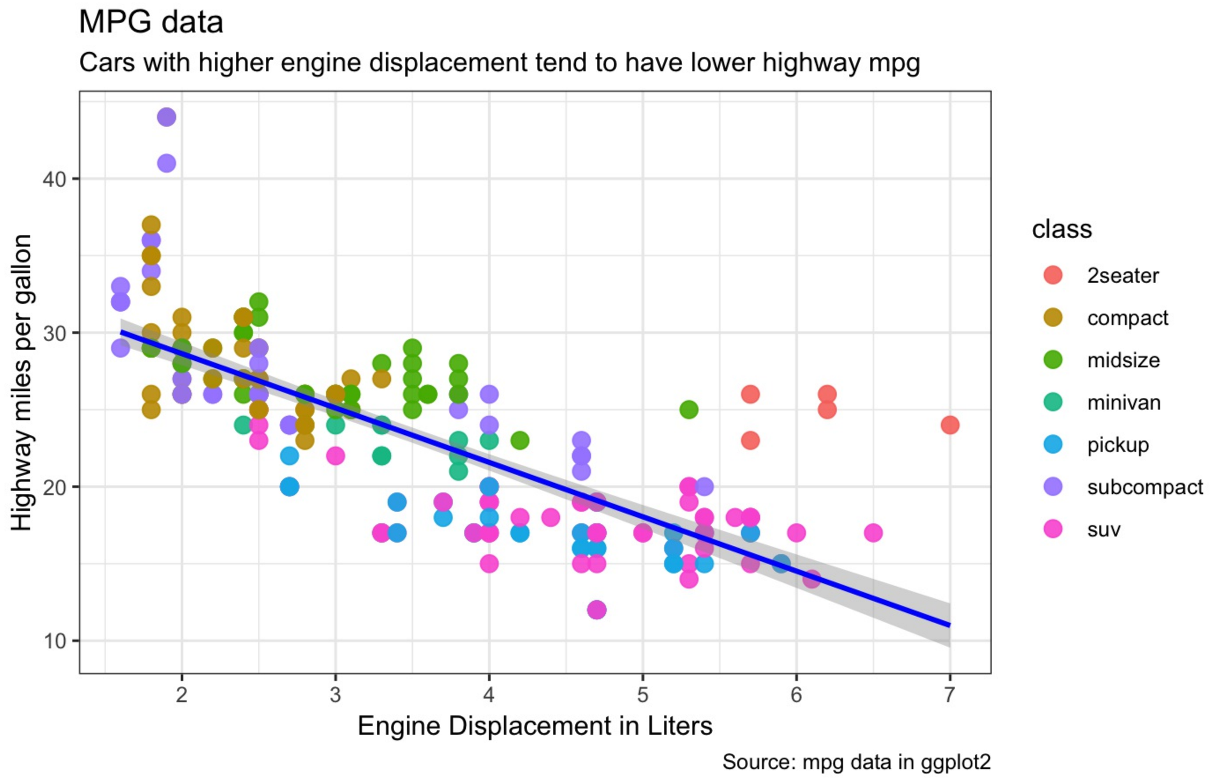
Problems with Base R plotting

- Default plots look pretty **outdated**.
- Plots can quickly require a **LOT of code**.
- Can't store plots as **objects** to reference and update later

Solution: `ggplot2`



This plot would take **a lot of code in Base R** but **just 10 lines of code**, 5 of which controlling the labels, in `ggplot2`.


MPG data

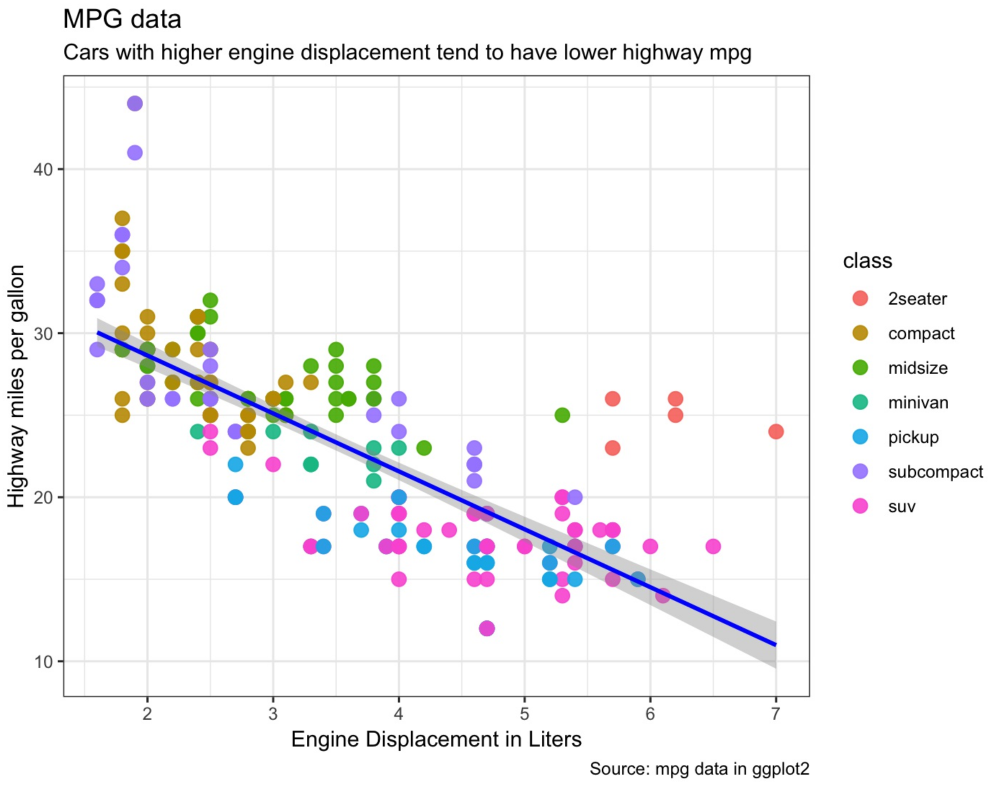
Cars with higher engine displacement tend to have lower highway mpg

Source: mpg data in ggplot2

Grammar of Graphics in `ggplot2`

The **Grammar of graphics** breaks down plots into several key pieces:

Aesthetics	Description
Data	What dataframe contains the data?
axes	What does the x-axis, y-axis, color (etc) represent?
color	What does color represent?
size	What does size represent?
geometries	What kind of geometric object do you want to plot?
facets	Should there be groups of plots?



MPG data

Cars with higher engine displacement tend to have lower highway mpg

Source: mpg data in ggplot2

Our goal: Creating this plot

Data

- Use the `mpg` tibble

Aesthetics

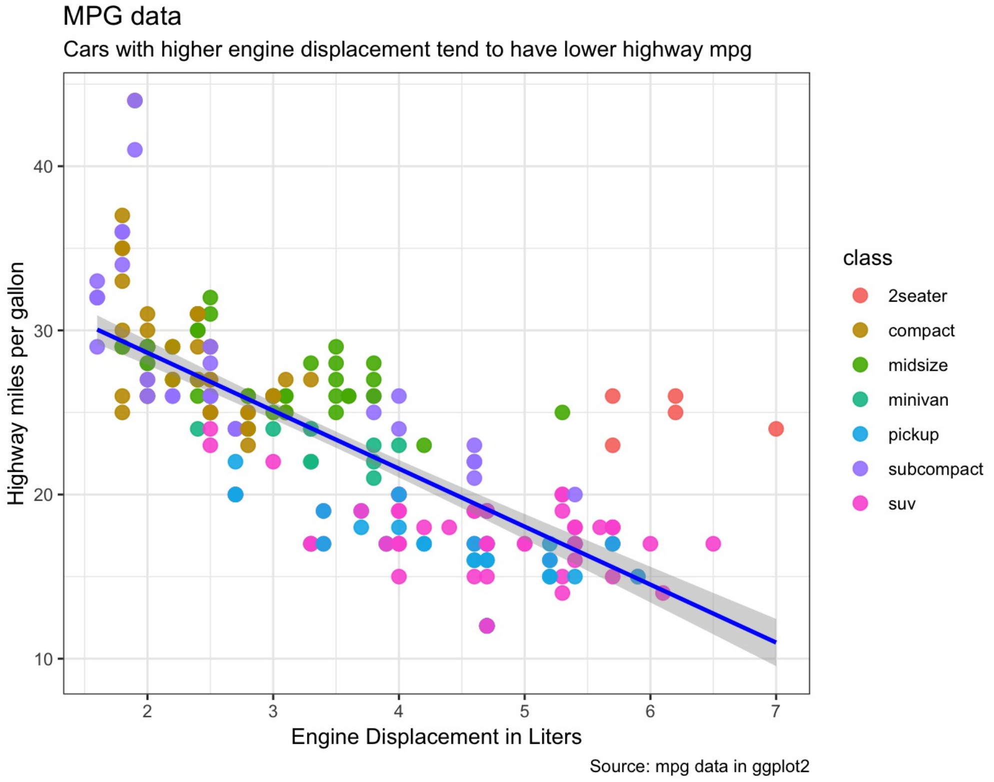
- Engine displacement (`disp`) on the x axis
- Highway miles per gallon (`hwy`) on the y-axis
- Color plotting elements by the `class` of car

Geometric objects

- Show data as points
- Add a regression line

Labels and themes

- Add plotting labels
- Use a black and white plotting theme



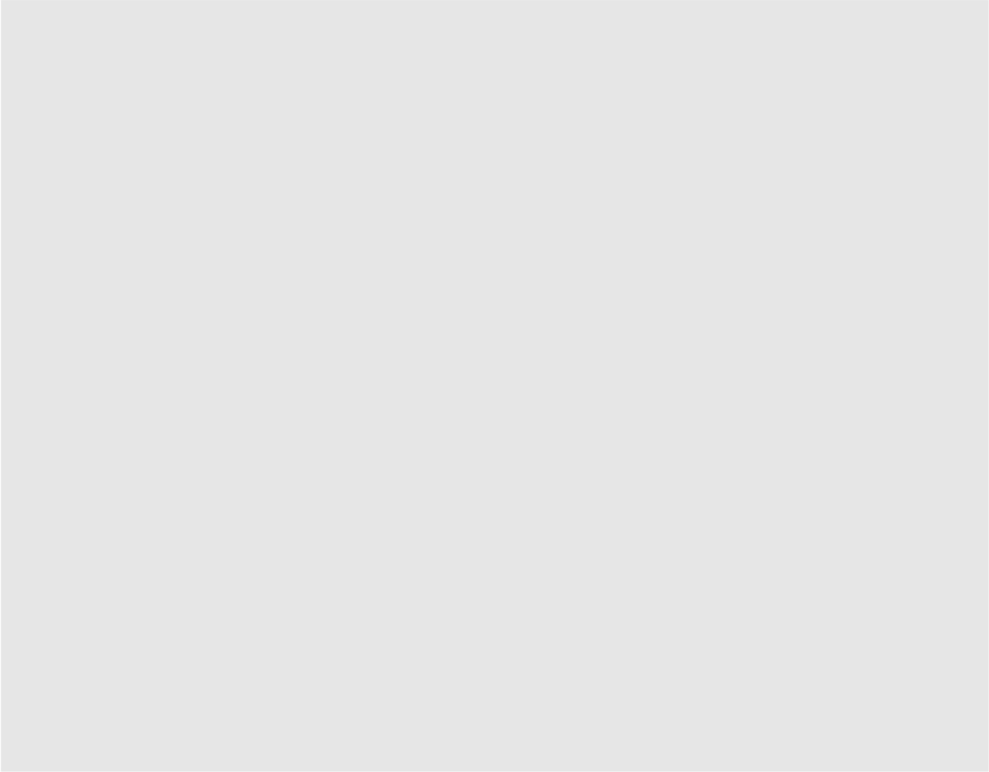
`ggplot()`

To **create a ggplot2 object**, use the `ggplot()` function

`ggplot()` has two main arguments:

- `data` - A data frame (aka `tibble`)
- `mapping` - A call to `aes()`

```
ggplot(data = mpg)
```



ggplot()

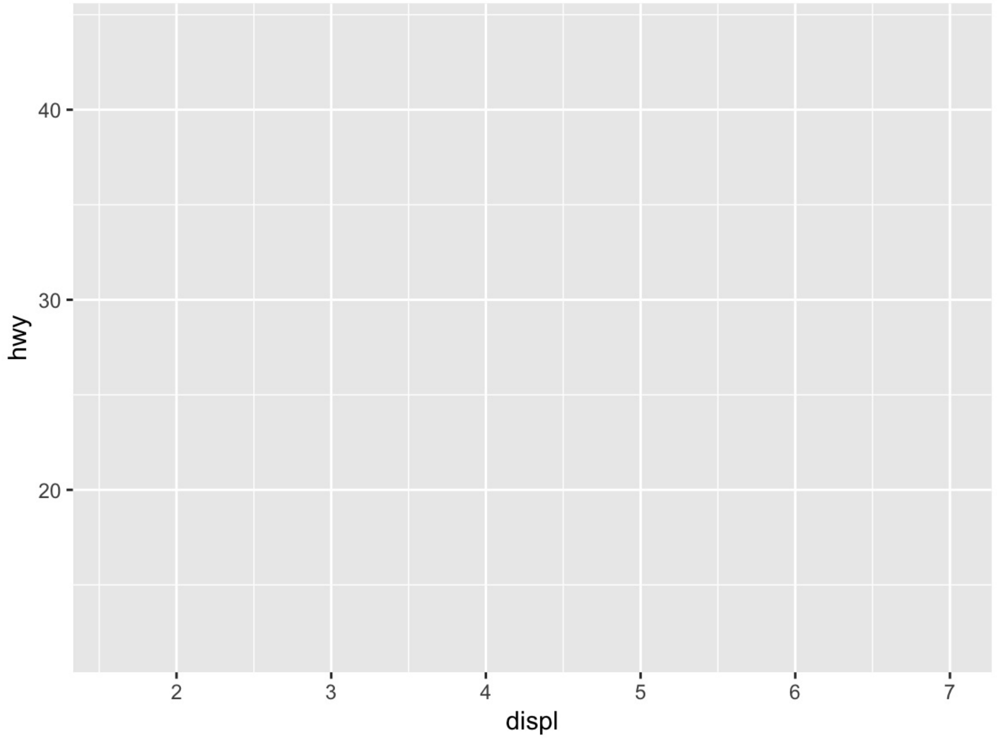
An **aesthetic mapping** is a visual property of the objects in your plot.

Use `aes()` to assign columns in your dataframe to properties in your plot.

Common aesthetics are...

aesthetics	Description
x, y	Data mapped to coordinates
`color`, `fill`	Border and fill colors
`alpha`	Transparency
`size`	Size
`shape`	Shape

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

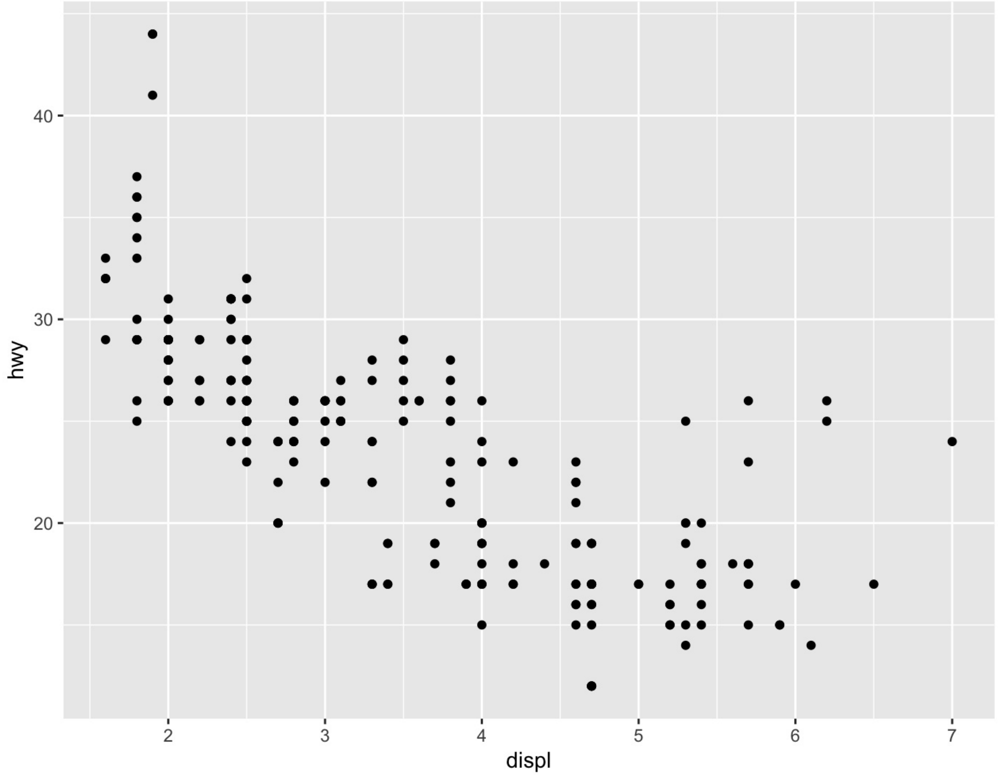


Adding elements to plots with +

Once you have specified the `data` argument, and global aesthetics with `mapping = aes()`, **add additional elements to the plot with +**.

The + operator works just like the pipe %>% in `dplyr`. **It just means "and then..."**

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



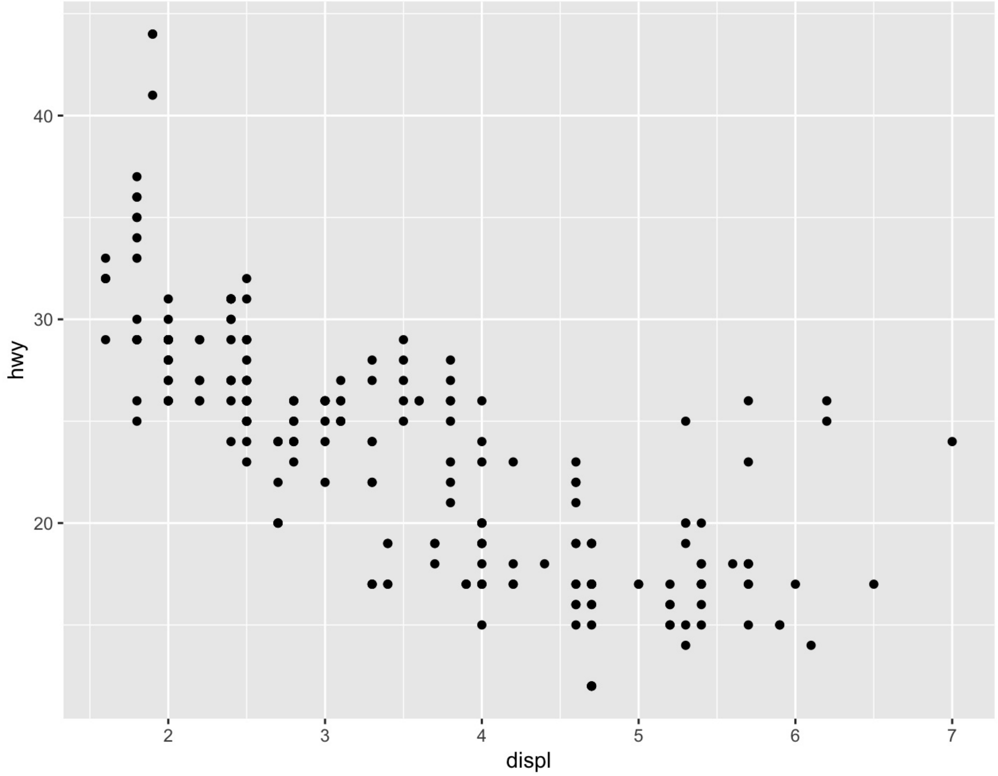
Geometric objects (geom)

A geom is a geometric object in a plot that represents data

To add a geom to a plot, just include + geom_X() where X is the type of geom.

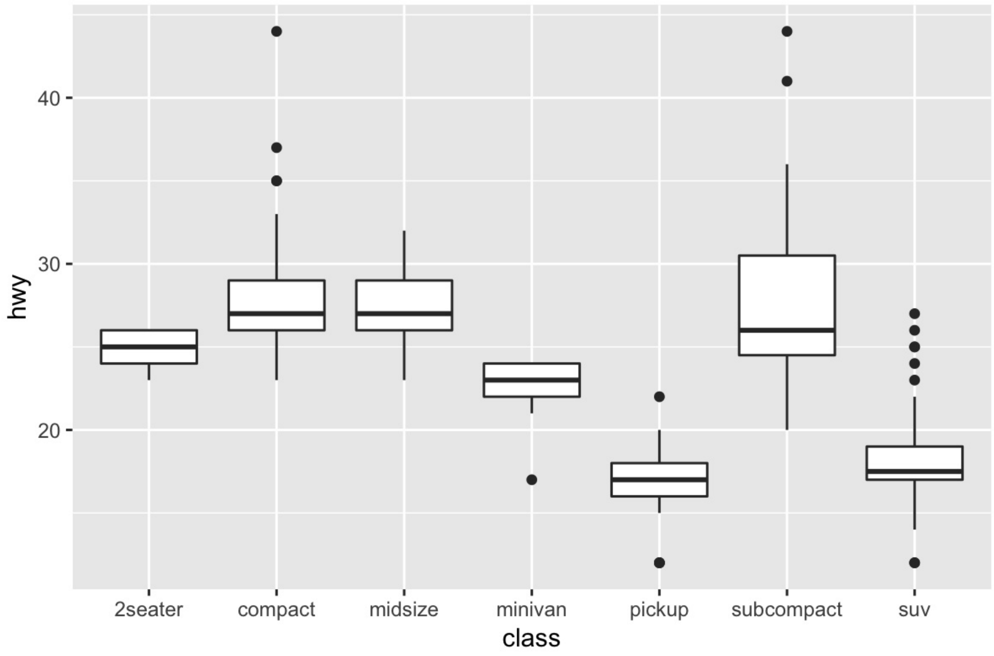
Common geoms are…

geom	output
geom_point()	Points
geom_bar()	Bar
geom_boxplot()	Boxplot
geom_count()	Points with size reflecting frequency
geom_smooth()	Smoothed line



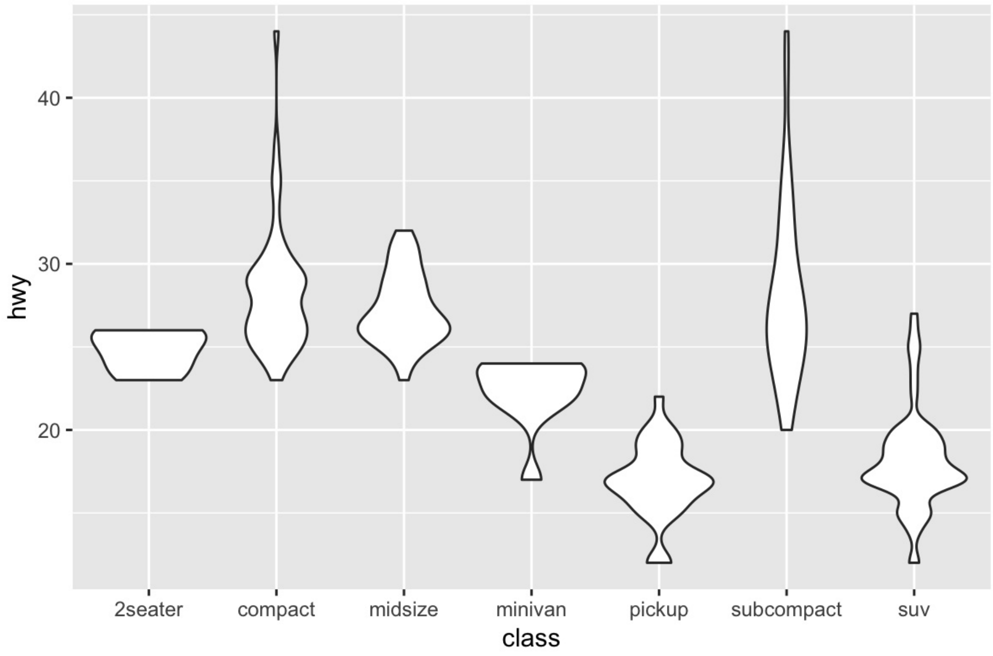
geom_boxplot()

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



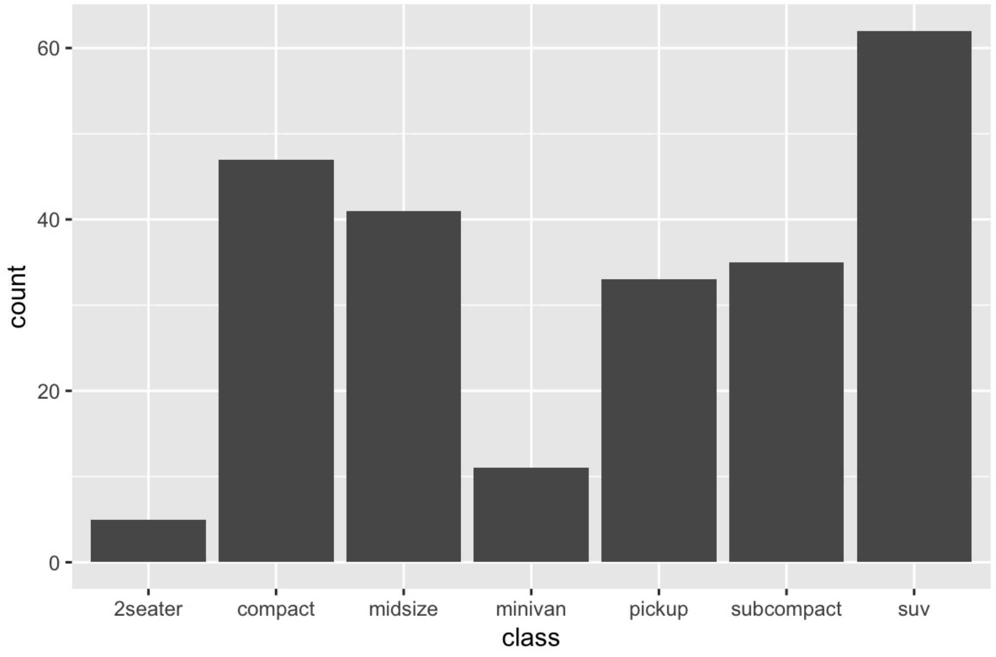
geom_violin()

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



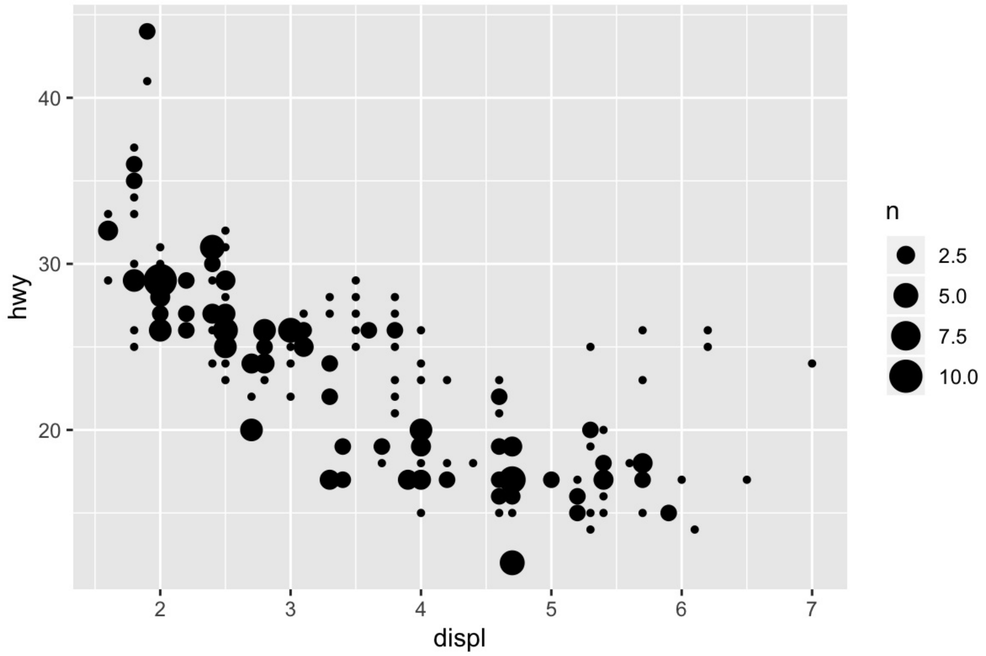
geom_bar()

```
ggplot(data = mpg,
  mapping = aes(x = class)) +
  geom_bar()
```



geom_count()

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



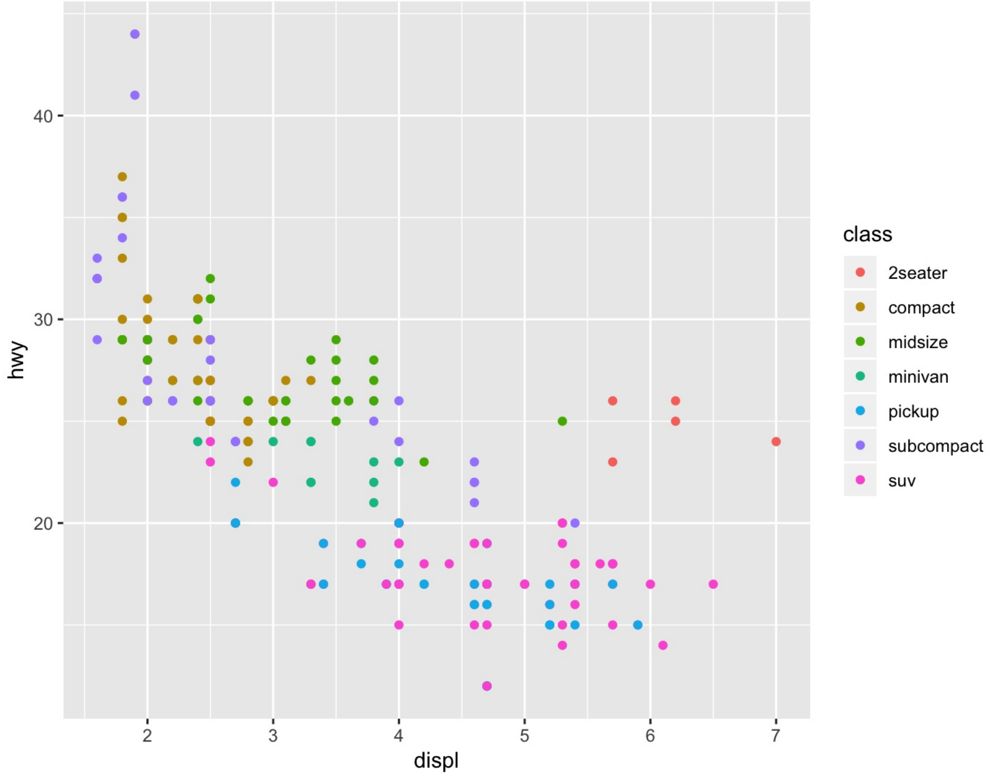
aes()

`color` geoms according to a variable.

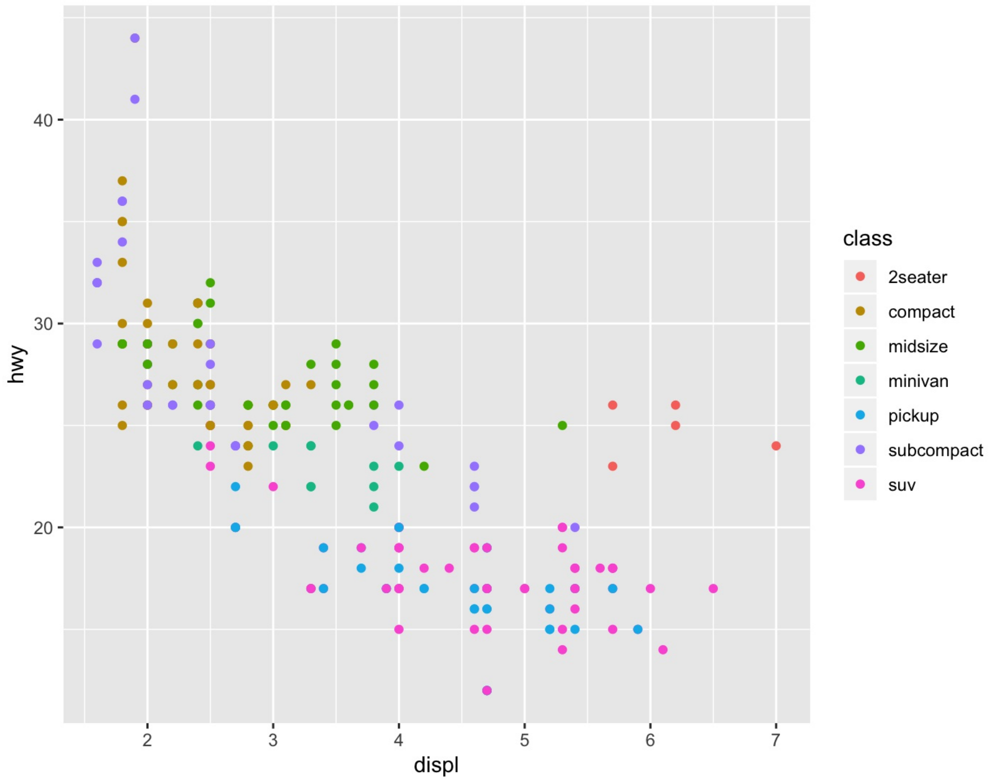
```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = class)) +
  geom_point()
```

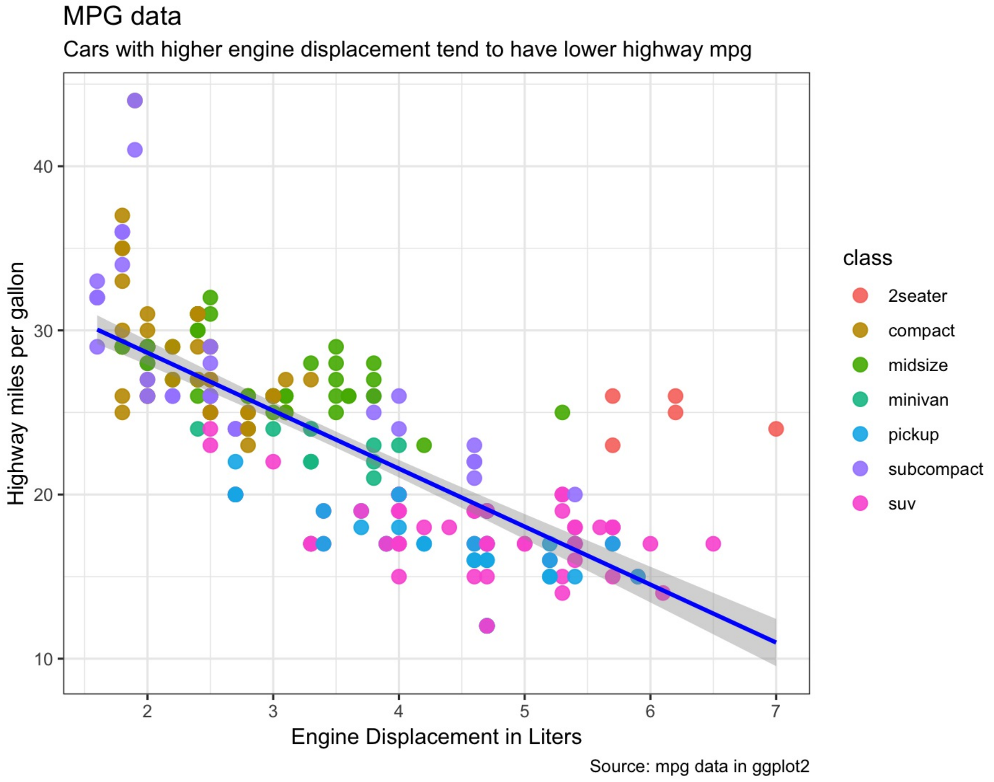
`mpg`

displ	hwy	class	year
4.0	26	subcompact	2008
3.0	26	compact	1999
4.0	18	pickup	2008
5.2	16	suv	1999
1.8	29	midsize	1999



What's next?





MPG data

Cars with higher engine displacement tend to have lower highway mpg

Source: mpg data in ggplot2

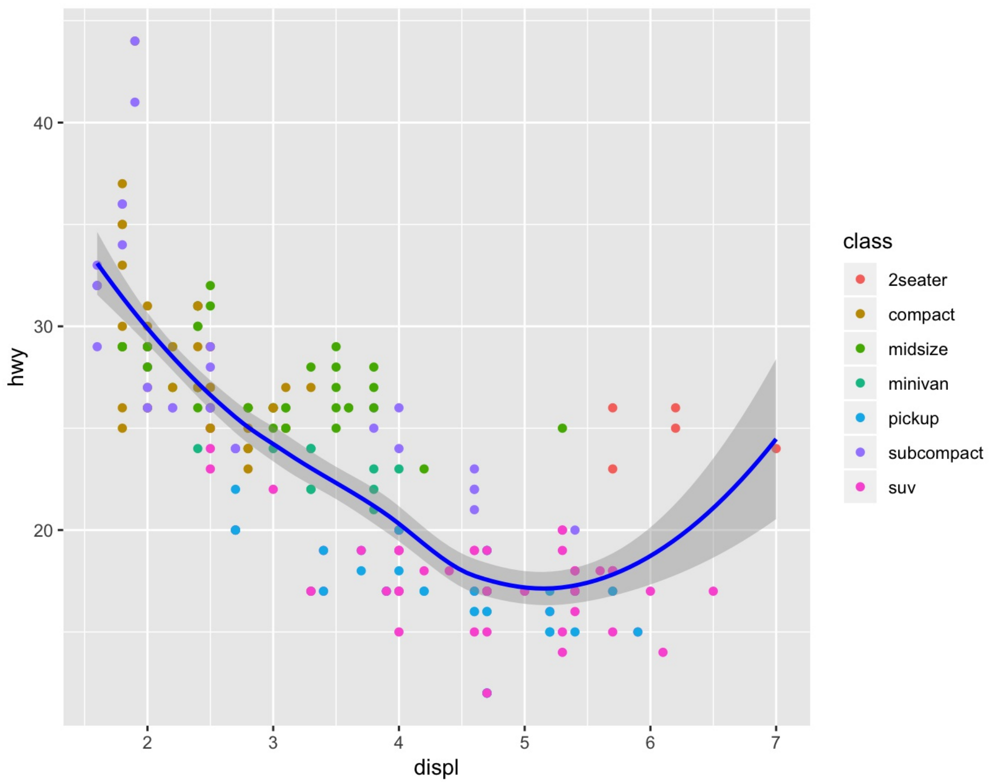
geom_smooth()

geom_smooth() adds a **smoothed line**.

Change how the line is created with method (e.g., method = lm).

Color the line with col.

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



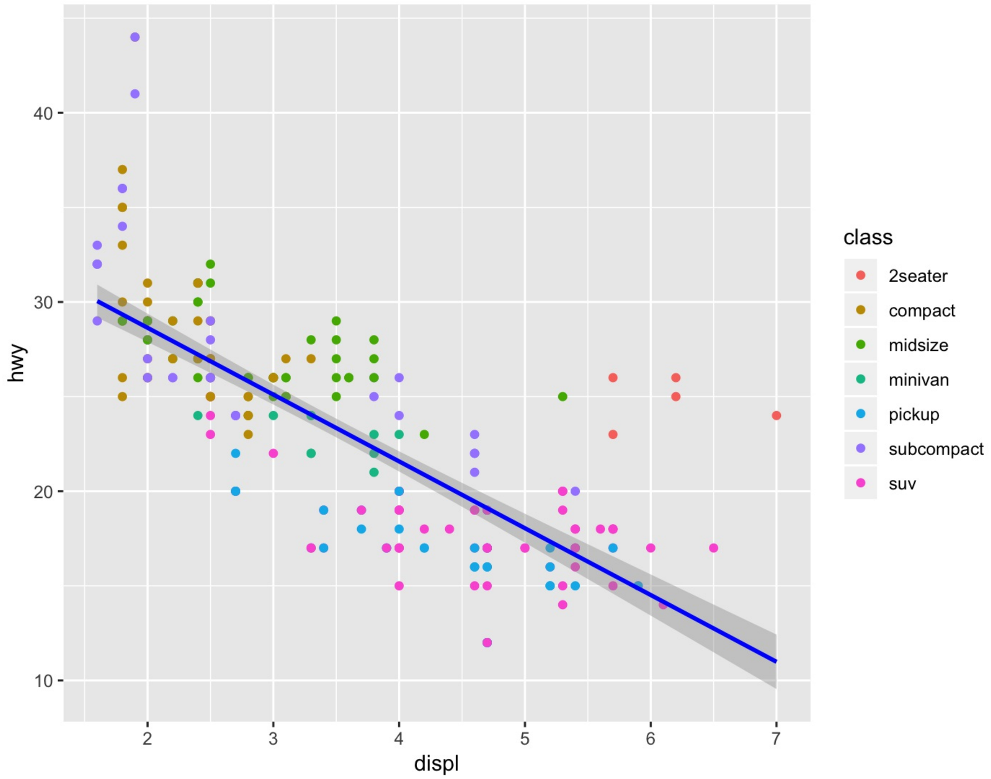
geom_smooth()

geom_smooth() adds a **smoothed line**.

Change how the line is created with method (e.g., method = lm).

Color the line with col

```
ggplot(data = mpg,
       mapping = aes(x = displ, y = hwy,
                     col = class)) +
  geom_point() +
  geom_smooth(col = "blue",
              method = "lm")
```

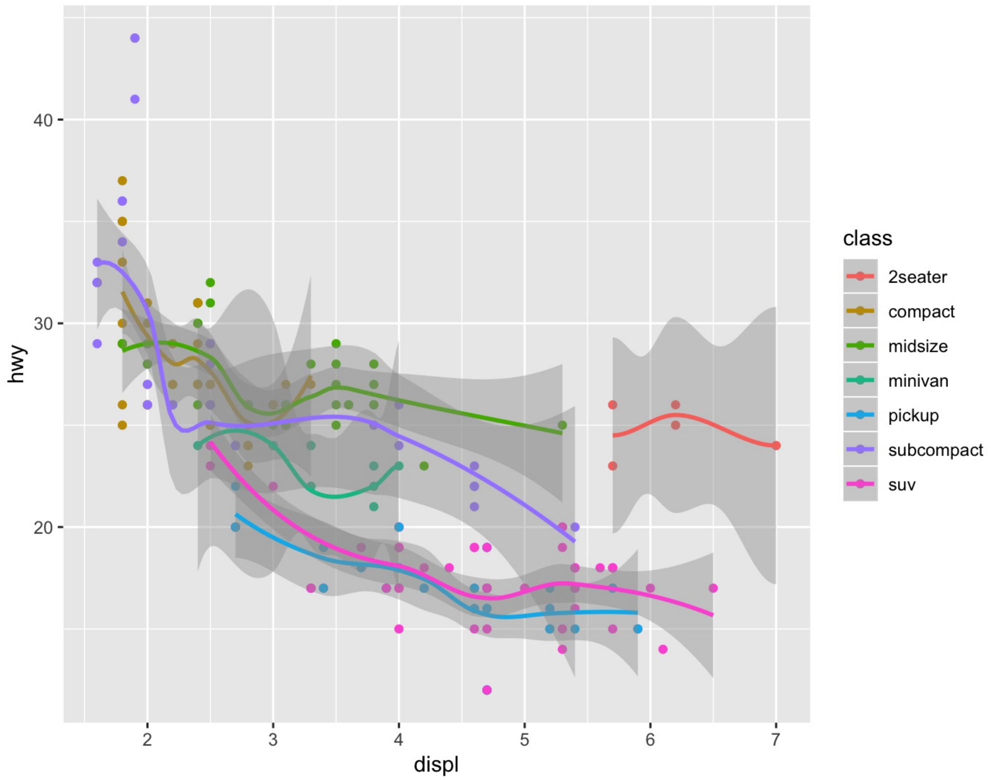


Overriding aesthetics

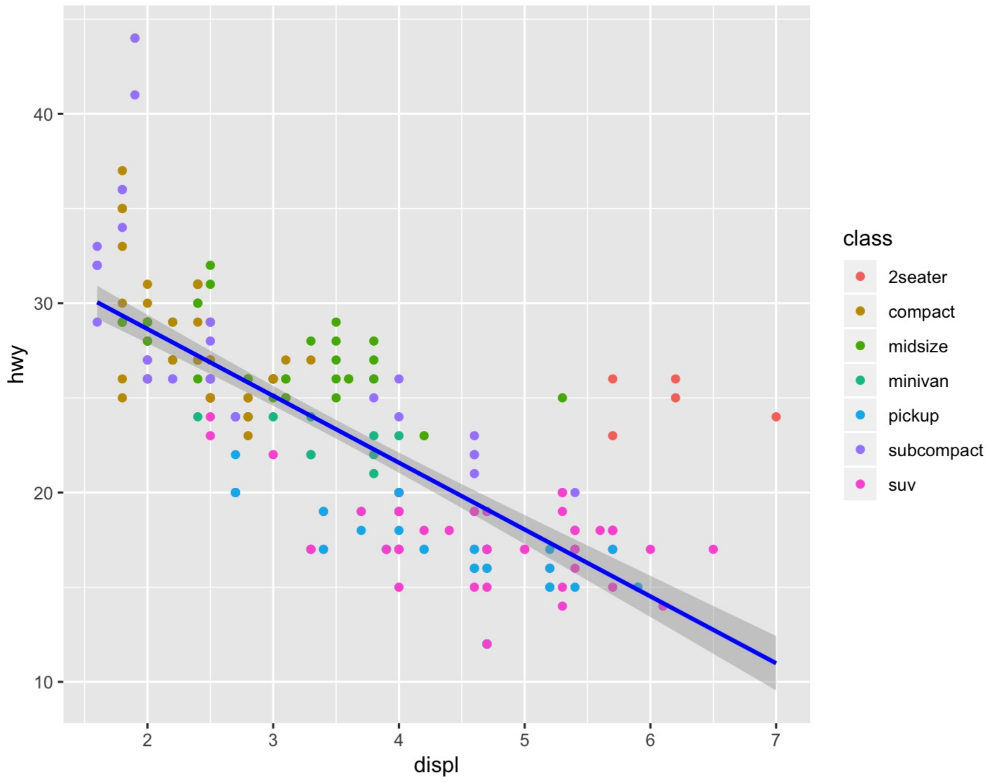
If you add additional plotting aesthetics, they will **override** the general plotting aesthetics.

This is what happens, when you don't override...

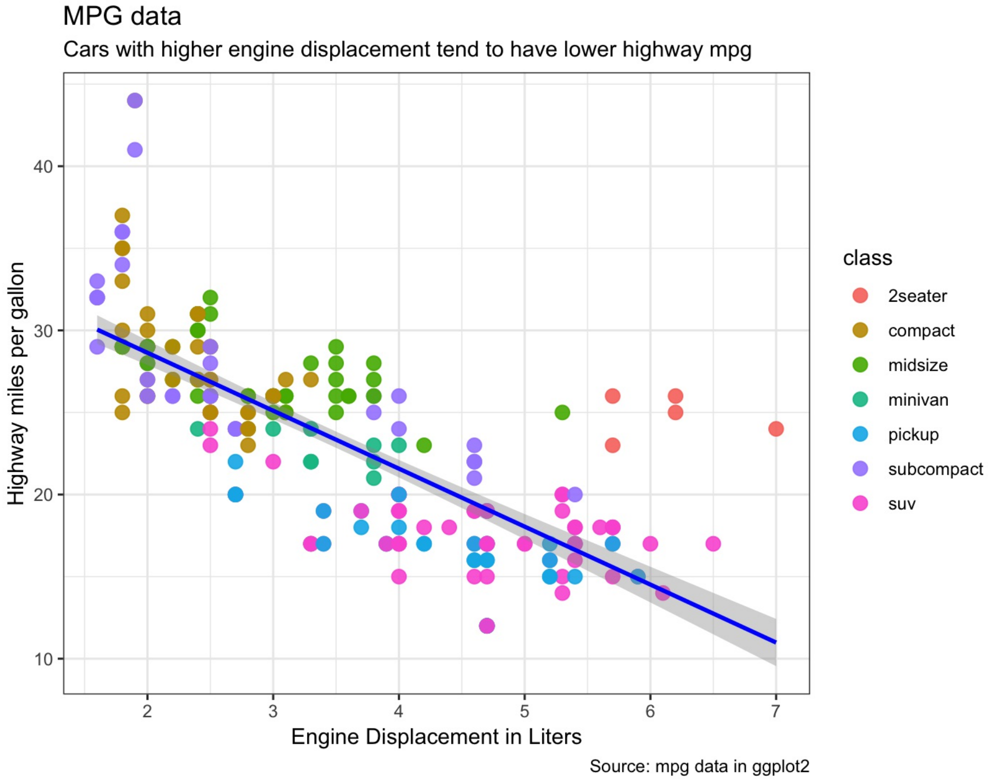
```
ggplot(data = mpg,
       mapping = aes(x = displ, y = hwy,
                     col = class)) +
  geom_point() +
  geom_smooth() # no overriding
```



What's next?



hwy
40
30
20
10
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv



MPG data
Cars with higher engine displacement tend to have lower highway mpg
Highway miles per gallon
40
30
20
10
2
3
4
5
6
7
Engine Displacement in Liters
Source: mpg data in ggplot2
class
2seater
compact
midsize
minivan
pickup
subcompact
suv

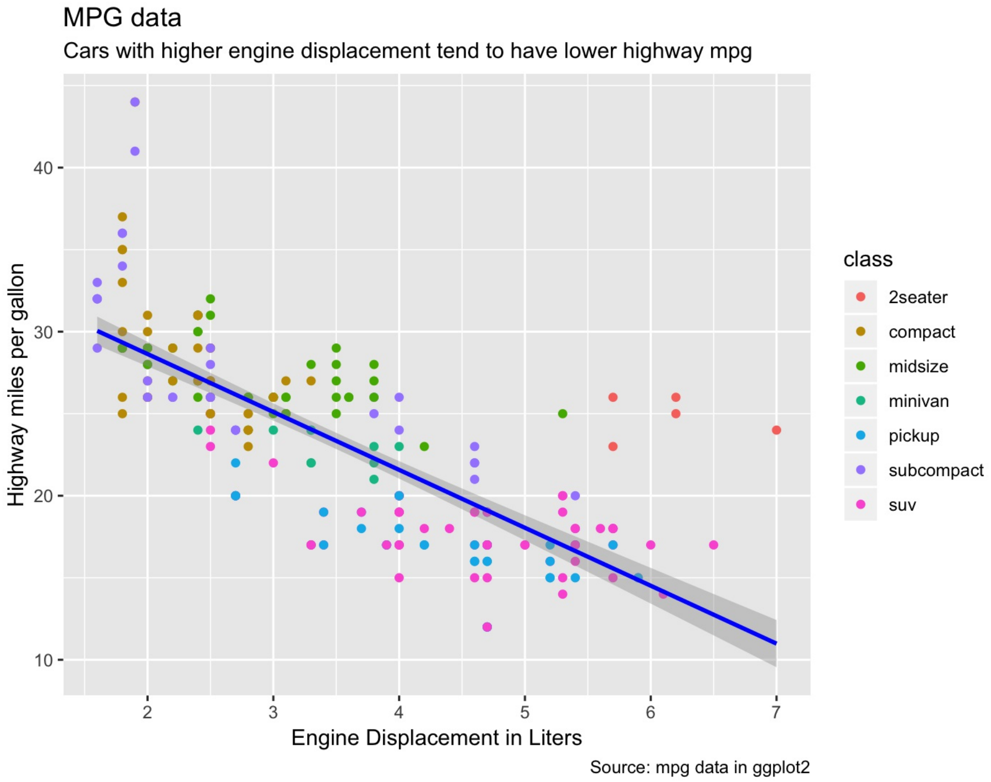
labs()

You can add **labels** to a plot with the `labs()` function

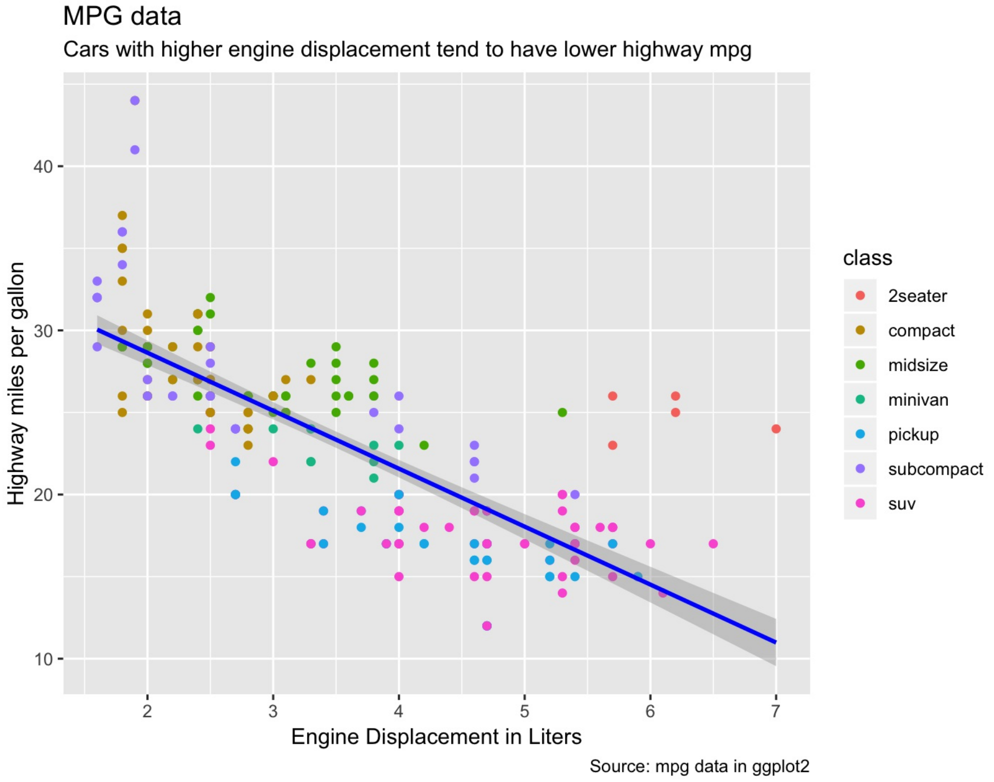
`labs()` arguments are ...

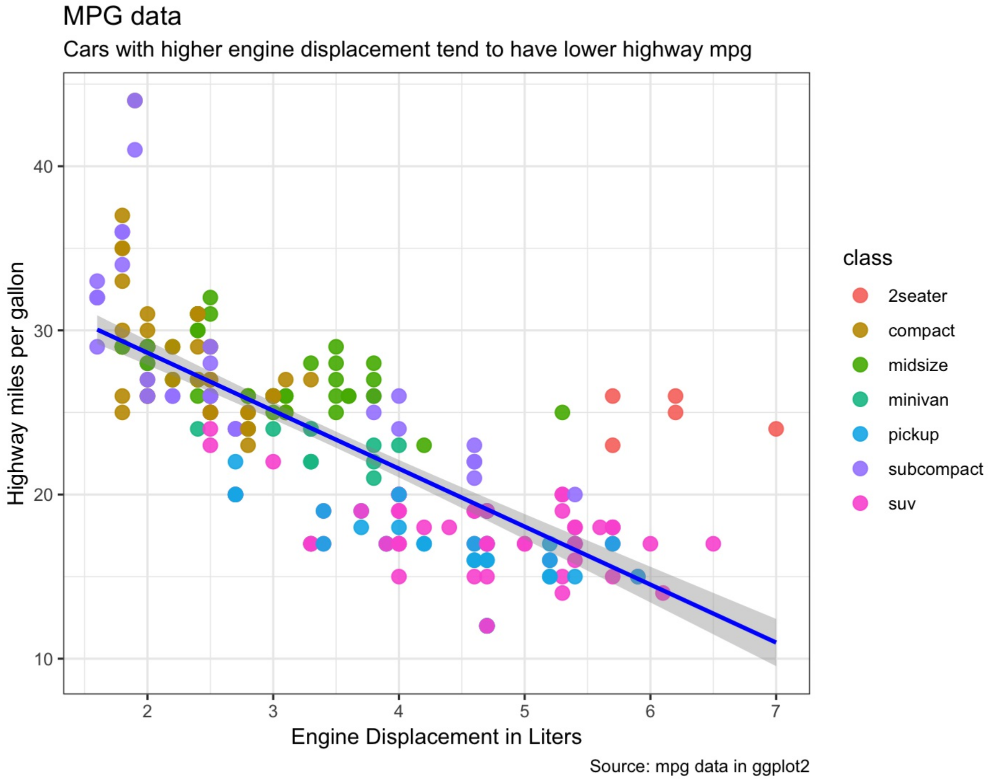
- `title` - Main title
- `subtitle` - Subtitle
- `caption` - Caption below

```
ggplot(...) +
  labs(x = "Engine Displ...",
       y = "Highway miles...",
       title = "MPG data",
       subtitle = "Cars with ...",
       caption = "Source...")
```



What's next?


MPG data
Cars with higher engine displacement tend to have lower highway mpg
Highway miles per gallon
10
20
30
40
2
3
4
5
6
7
Engine Displacement in Liters
class
2seater
compact
midsize
minivan
pickup
subcompact
suv
Source: mpg data in ggplot2


MPG data
Cars with higher engine displacement tend to have lower highway mpg
Highway miles per gallon
10
20
30
40
2
3
4
5
6
7
Engine Displacement in Liters
class
2seater
compact
midsize
minivan
pickup
subcompact
suv
Source: mpg data in ggplot2

Themes with `theme_XX()`

A plotting **theme** controls many aspects of its **overall look**, from the background, to the grid lines, to the label font to the spacing between plot labels and the plotting space.

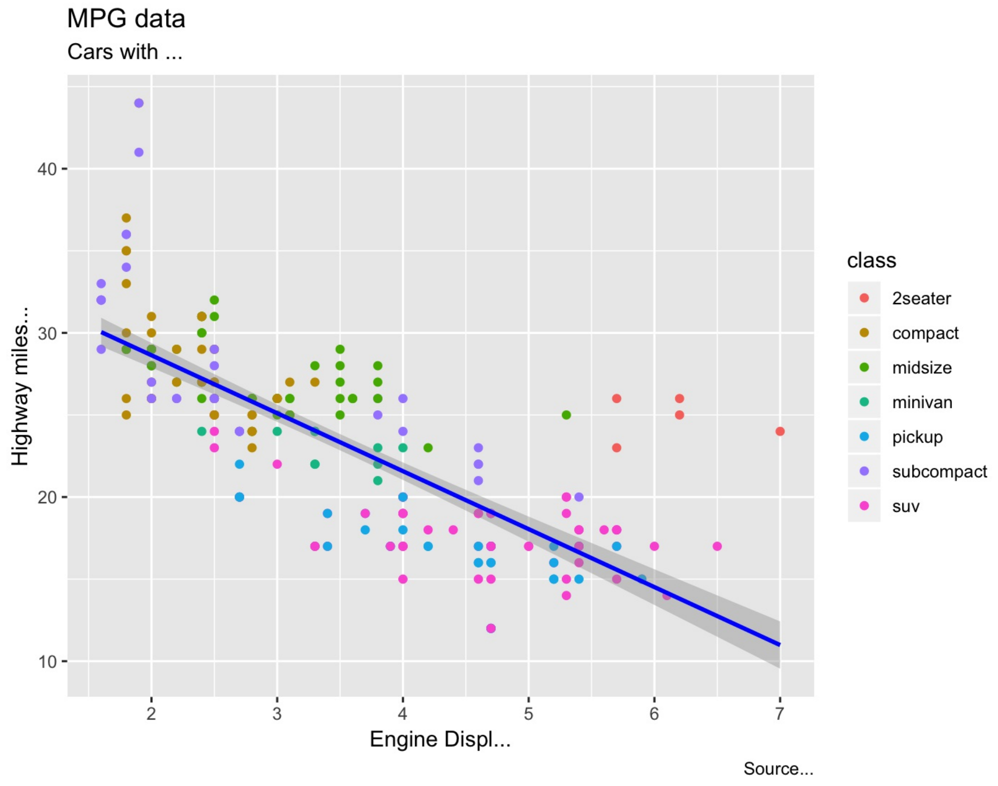
Themes built into `ggplot2`

`theme_bw()`
`theme_minimal()`
`theme_classic()`
`theme_light()`
`theme_void()`

Themes from the `ggthemes` package

`theme_excel()`
`theme_economist()`
etc.

```
ggplot(...) +
  theme_gray() # The Default theme
```



Themes with `theme_XX()`

A plotting **theme** controls many aspects of its **overall look**, from the background, to the grid lines, to the label font to the spacing between plot labels and the plotting space.

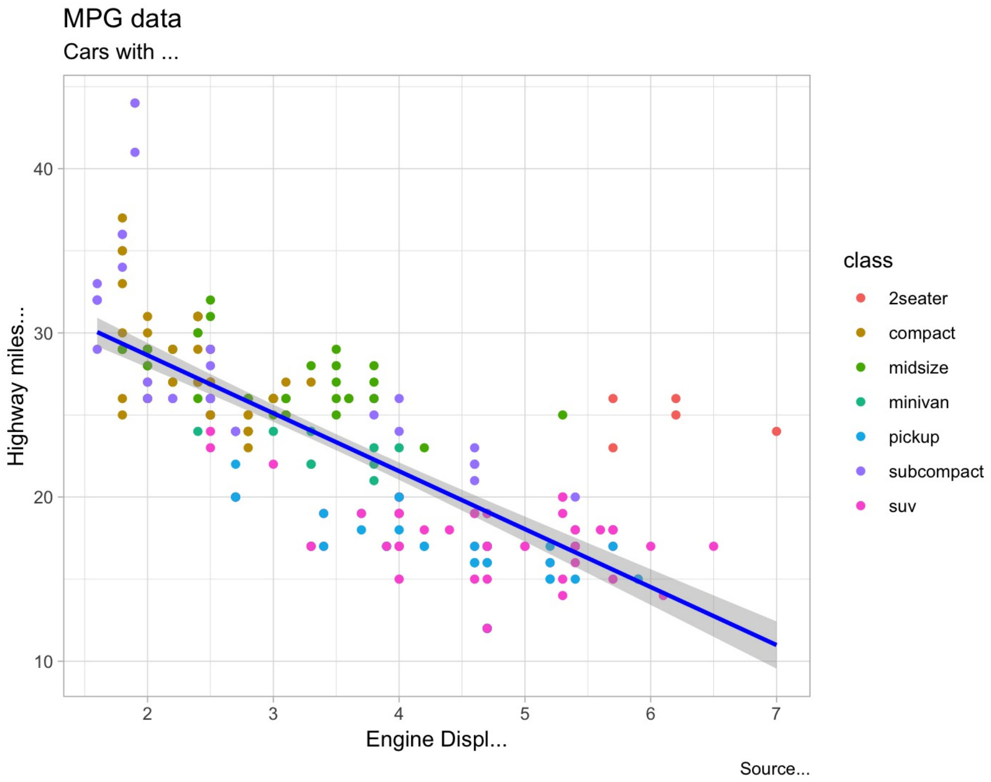
Themes built into `ggplot2`

`theme_bw()`
`theme_minimal()`
`theme_classic()`
`theme_light()`
`theme_void()`

Themes from the `ggthemes` package

`theme_excel()`
`theme_economist()`
etc.

```
ggplot(...) +
  theme_light()
```



Themes with `theme_XX()`

A plotting **theme** controls many aspects of its **overall look**, from the background, to the grid lines, to the label font to the spacing between plot labels and the plotting space.

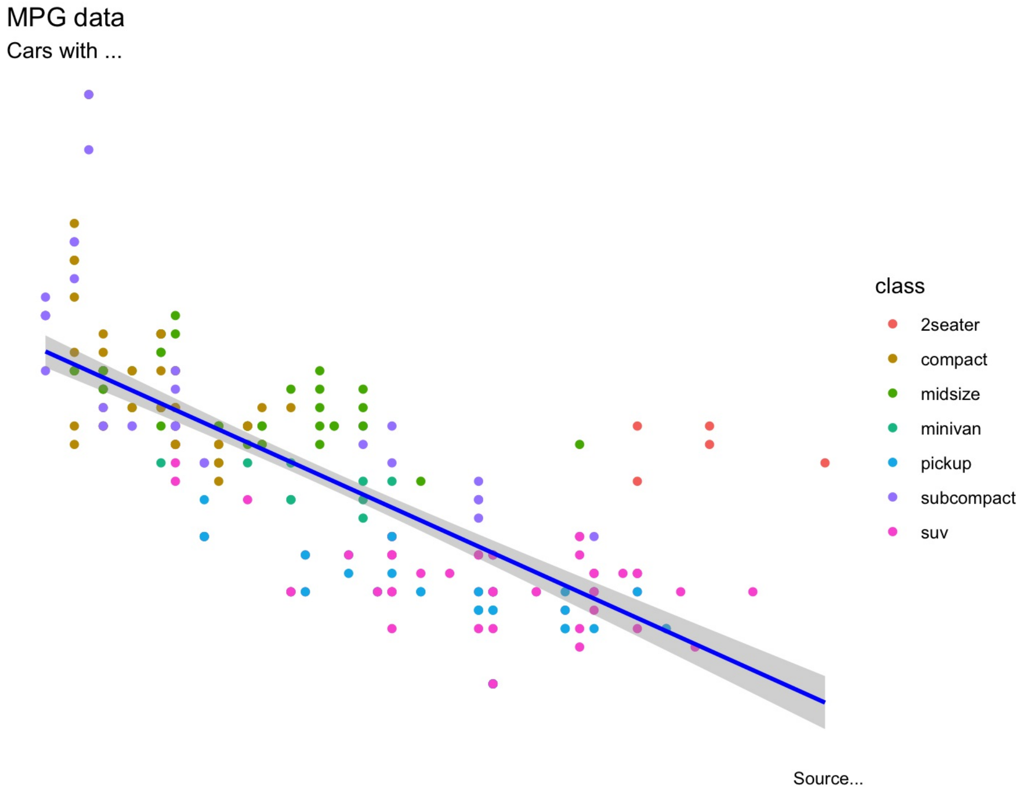
Themes built into `ggplot2`

`theme_bw()`
`theme_minimal()`
`theme_classic()`
`theme_light()`
`theme_void()`

Themes from the `ggthemes` package

`theme_excel()`
`theme_economist()`
etc.

```
ggplot(...) +
  theme_void()
```



Themes with `theme_XX()`

A plotting **theme** controls many aspects of its **overall look**, from the background, to the grid lines, to the label font to the spacing between plot labels and the plotting space.

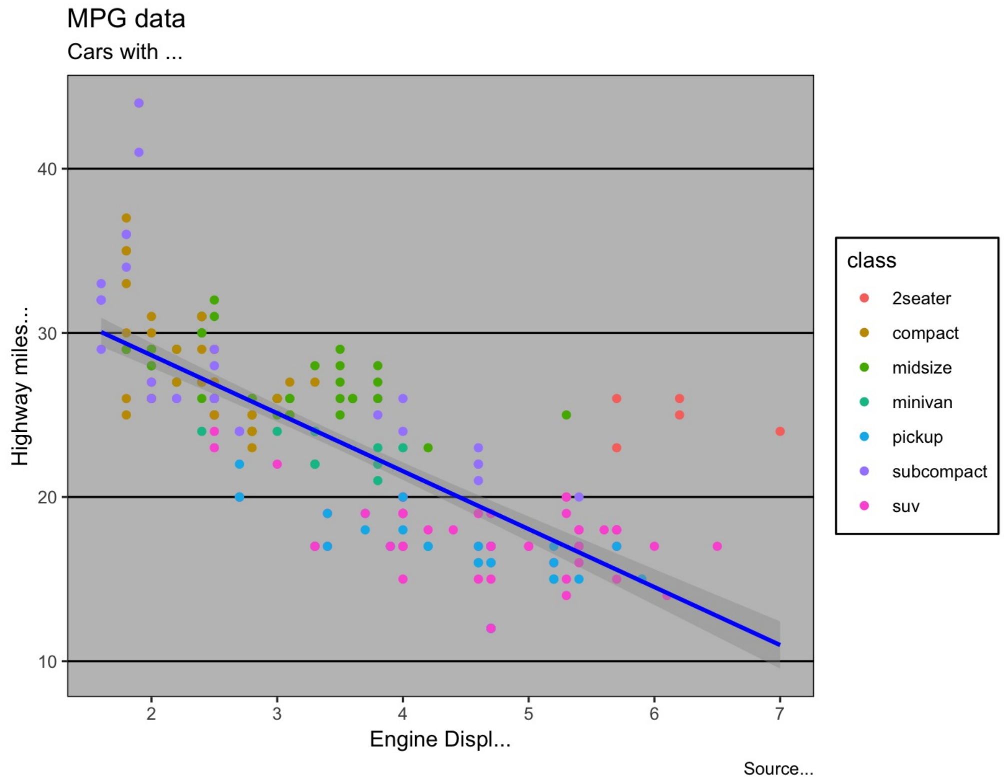
Themes built into `ggplot2`

```
theme_bw()
theme_minimal()
theme_classic()
theme_light()
theme_void()
```

Themes from the `ggthemes` package

```
theme_excel()
theme_economist()
```
etc.

```
ggplot(...) +
  theme_excel()
```



Themes with `theme_XX()`

A plotting **theme** controls many aspects of its **overall look**, from the background, to the grid lines, to the label font to the spacing between plot labels and the plotting space.

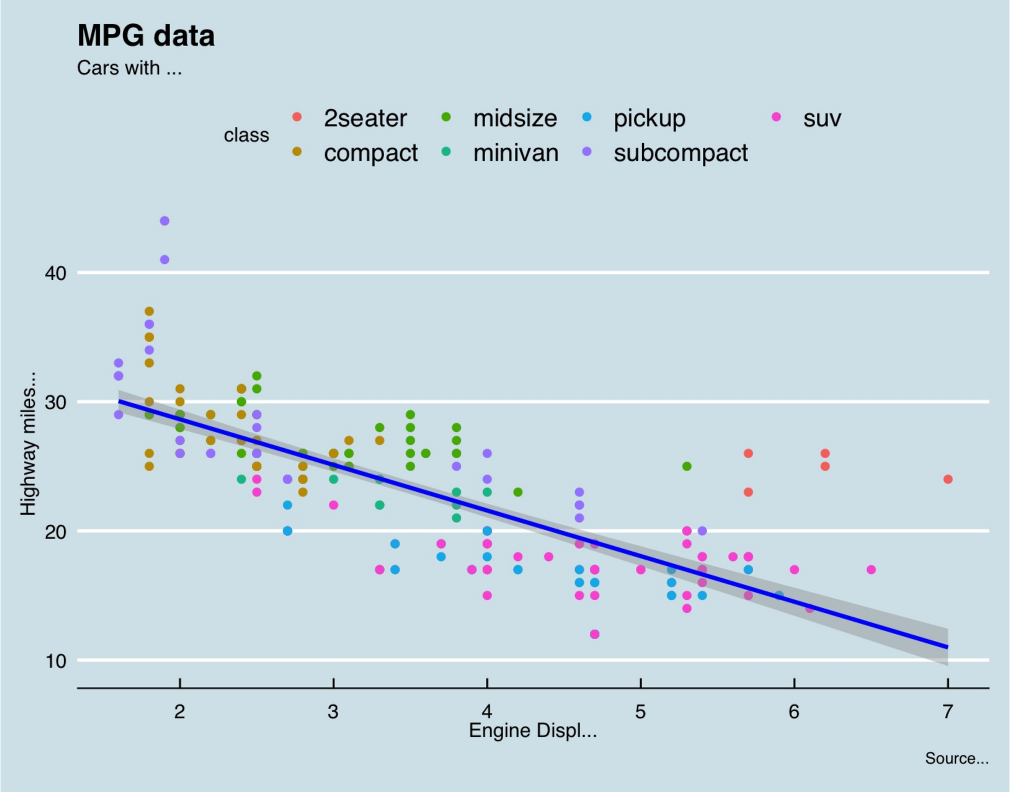
Themes built into `ggplot2`:

```
theme_bw()
theme_minimal()
theme_classic()
theme_light()
theme_void()
```

Themes from the `ggthemes` package

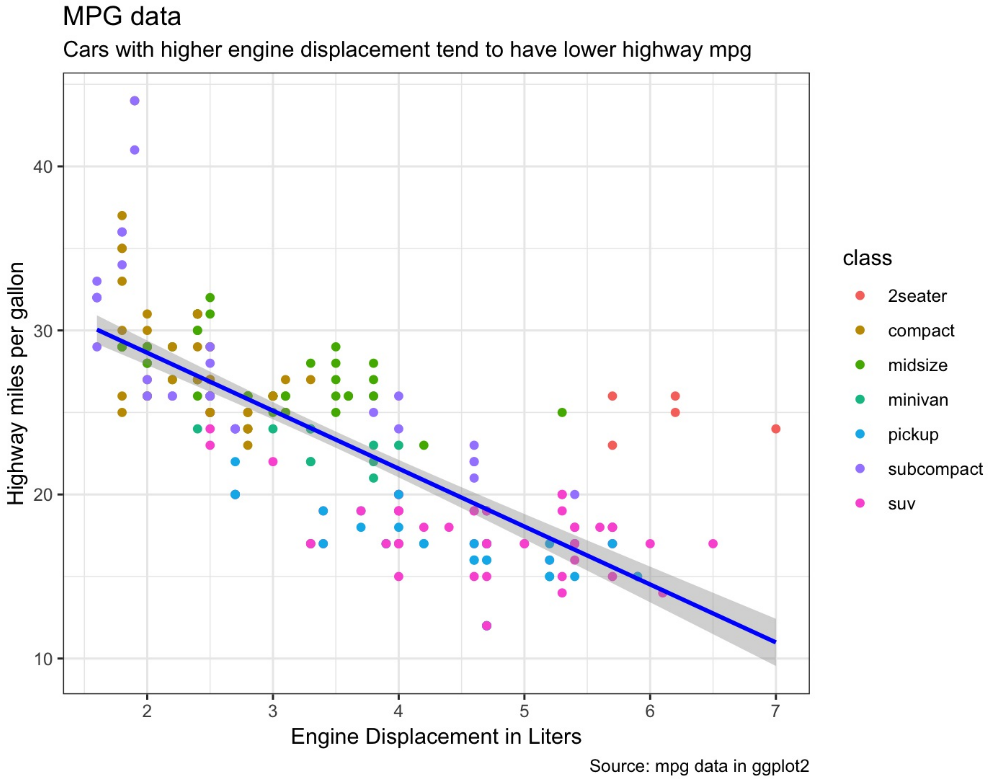
```
theme_excel()
theme_economist()
```
etc.

```
ggplot(...) +
  theme_economist()
```



Final result!

```r
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     col = class)) +
  geom_point() +
  geom_smooth(col = "blue",
              method = "lm")+
  labs(x = "Engine Displ. in Liters",
       y = "Highway miles per gallon",
       title = "MPG data",
       subtitle = "Cars with higher...",
       caption = "Source: mpg data...") +
  theme_bw()
```



MPG data

Cars with higher engine displacement tend to have lower highway mpg

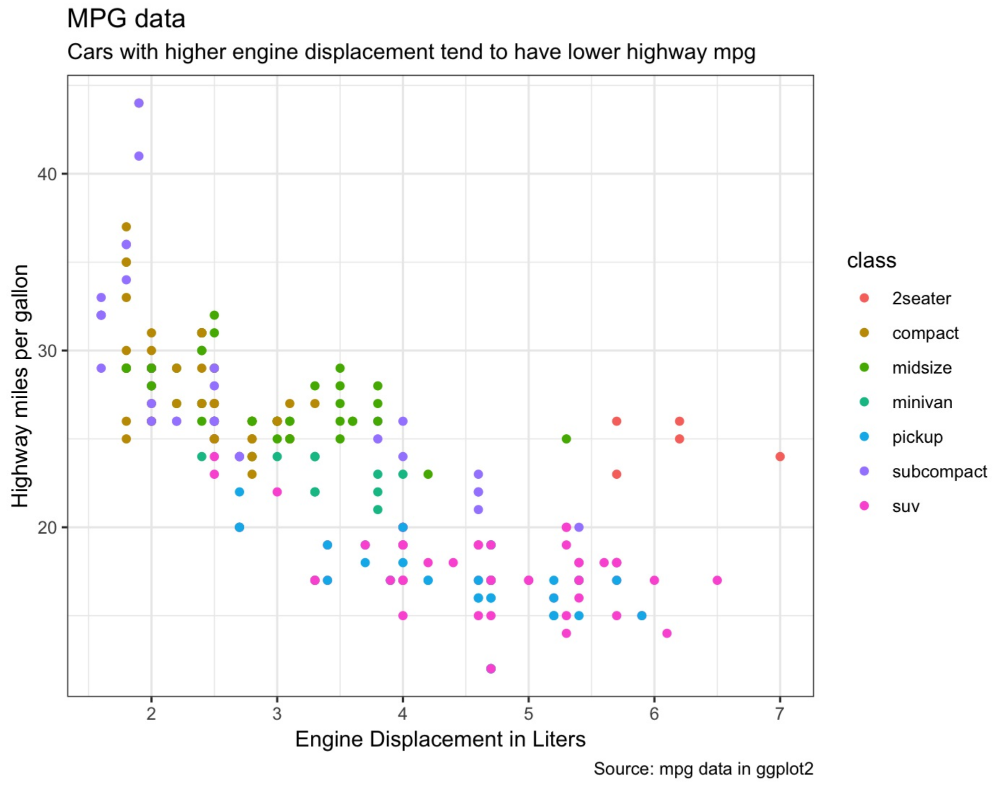
Source: mpg data in ggplot2

The gg object

1 ggplot returns an object of the class *gg*.
2 You can assign the result of `ggplot` to an **object**.
3 Evaluating the object will show the plot.
4 You can even edit existing `ggplot` **objects**.

```
# Create myplot
myplot <- ggplot(data = mpg,
        mapping = aes(x = displ,y = hwy,
                      col = class)) +
  geom_point() +
  labs(x = "Engine Displacement in ...",
       y = "Highway miles per gallon",
       title = "MPG data",
       subtitle = "Cars with higher ...",
       caption = "Source: mpg data ...") +
  theme_bw()
```

```
# Evaluate myplot
myplot
```

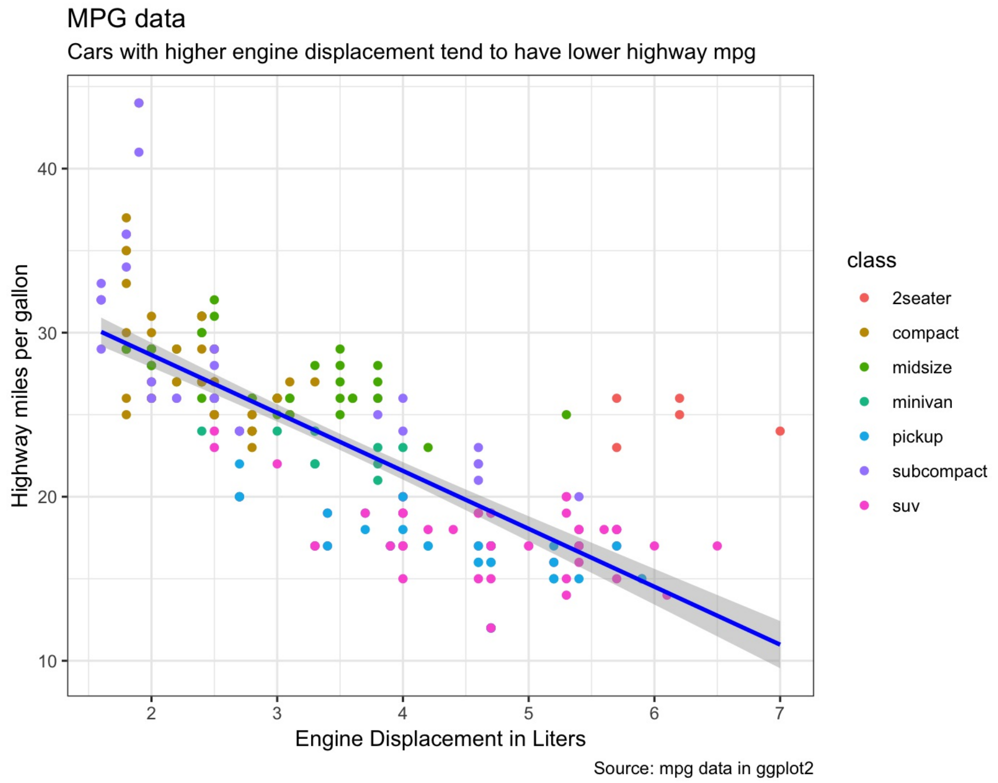


The *gg* object

1 ggplot returns an object of the class *gg*.
2 You can assign the result of `ggplot` to an **object**.
3 Evaluating the object will show the plot.
4 You can even edit existing `ggplot` **objects**.

```
# Create myplot
myplot <- ggplot(data = mpg,
       mapping = aes(x = displ,y = hwy,
                     col = class)) +
  geom_point() +
  labs(x = "Engine Displacement in ...",
       y = "Highway miles per gallon",
       title = "MPG data",
       subtitle = "Cars with higher ...",
       caption = "Source: mpg data ...") +
  theme_bw()
```

```
myplot + # add geom
  geom_smooth(col = "blue", method = "lm")
```

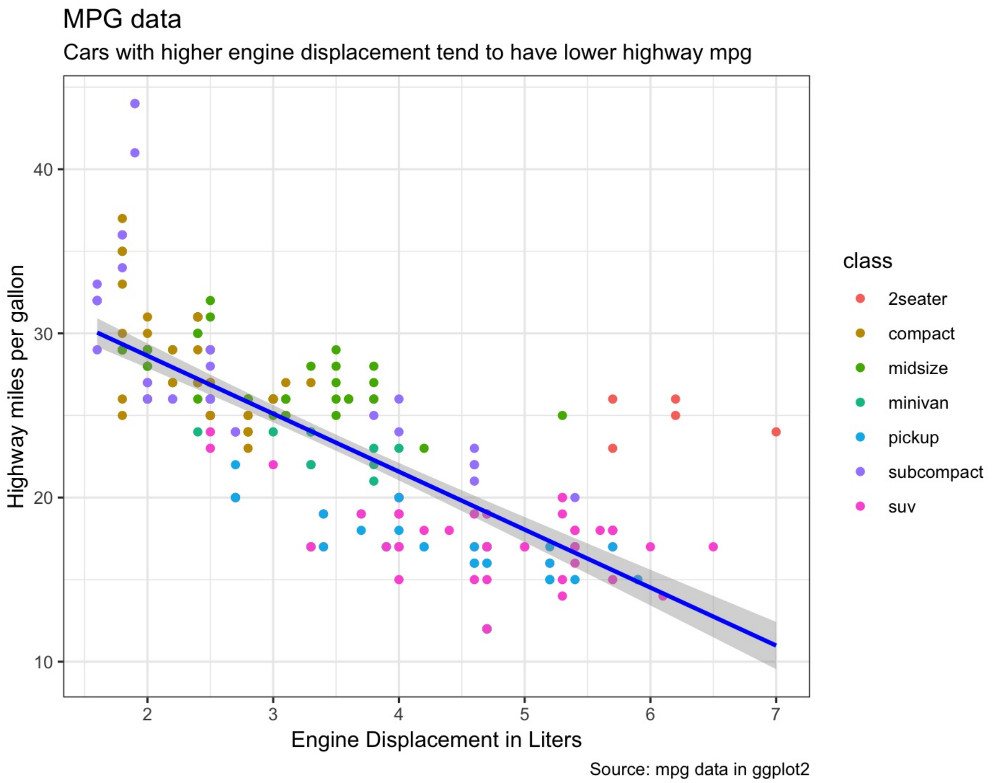


`facet_wrap()`

Faceting = **same plot for different groups**.

To facet plots, use, e.g., `facet_wrap()`.

```
# Without faceting
myplot +
  geom_smooth(col = "blue",
              method = "lm")
```

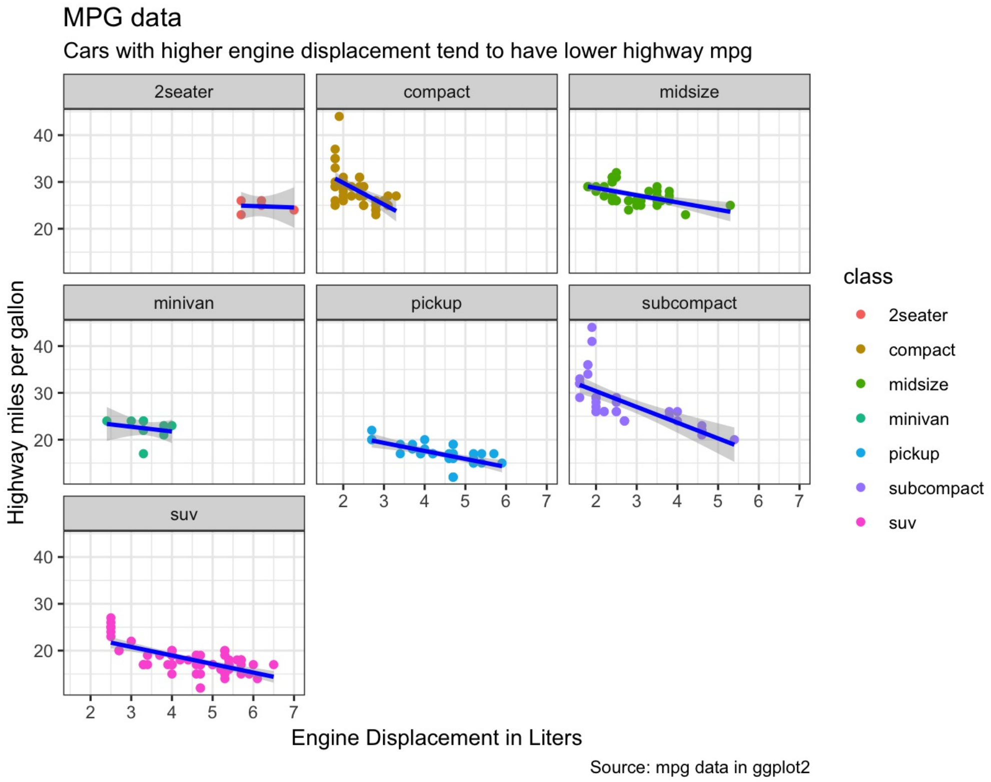


facet_wrap()

Faceting = **same plot for different groups**.

To facet plots, use, e.g., `facet_wrap()`.

```
# With faceting
myplot +
  geom_smooth(col = "blue",
              method = "lm")  +
  facet_wrap(~ class)  # Tilde first
```


MPG data

Cars with higher engine displacement tend to have lower highway mpg

Source: mpg data in ggplot2

ggsave()

To create an **image file** of a plot (e.g., `.jpg`, `.pdf`, `.png`), use the `ggsave()` function.

Arguments

Argument	Description
`filename`	Name of the to-be-created file
`device`	File type (e.g.; "pdf", "jpeg", "png")
`path`	Folder to store image in
`width`, `height`	Plot width, height (e.g., inches)

```r
# Create myplot object
myplot <- myplot +
  geom_smooth(col = "blue", method = "lm") +
  facet_wrap(~ class)

# Create "myplot.pdf", from myplot
ggsave(filename = "myplot",
       plot = myplot,
       device = "pdf",
       path = "figures",
       width = 6,
       height = 4)
```

ggsave()

To create an **image file** of a plot (e.g., `.jpg`, `.pdf`, `.png`), use the `ggsave()` function.

Arguments

Argument	Description
`filename`	Name of the to-be-created file
`device`	File type (e.g.; "pdf", "jpeg", "png")
`path`	Folder to store image in
`width`, `height`	Plot width, height (e.g., inches)

```r
# Create myplot object
myplot <- myplot +
  geom_smooth(col = "blue", method = "lm") +
  facet_wrap(~ class)

# Create "myplot.png", from myplot
ggsave(filename = "myplot",
       plot = myplot,
       device = "png",
       path = "figures",
       width = 6,
       height = 4)
```

Practical