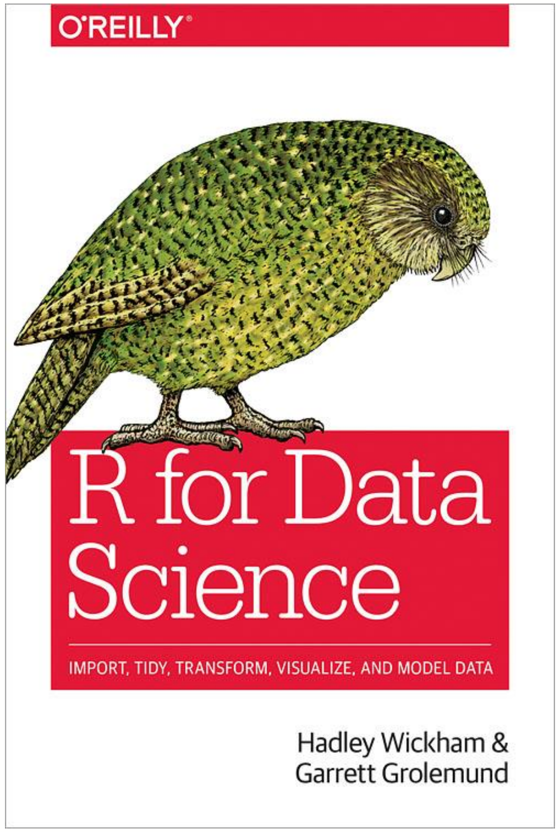
Session 3:
Introduction to ggplot2

Andrew Jones | Strategy Unit

Acknowledgement

This session shadows Chapter 3 of the excellent:

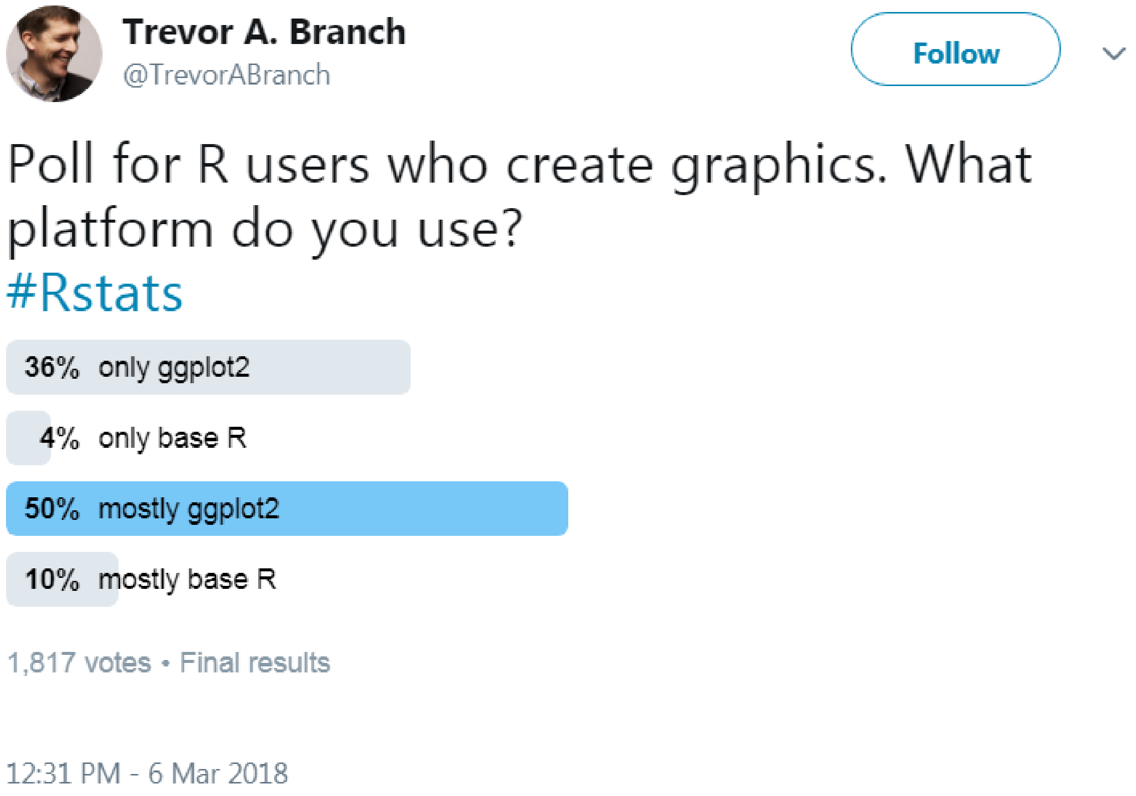

O'REILLY
R for Data Science
IMPORT, TIDY, TRANSFORM, VISUALIZE, AND MODEL DATA
Hadley Wickham & Garrett Grolemund

ggplot2

Is one of several plotting systems in R

ggplot2

Is one of several plotting systems in R



Trevor A. Branch
@TrevorABranch

Poll for R users who create graphics. What platform do you use?
#Rstats

36% only ggplot2

4% only base R

50% mostly ggplot2

10% mostly base R

1,817 votes • Final results

12:31 PM - 6 Mar 2018

Why ggplot2?

1. Highly versatile

2. Relatively easy to make good-looking plots

3. It meshes well with other tools we will be learning

ggplot2

ggplot2 is part of the tidyverse, so:

```
library(tidyverse)
```

mpg data

Data on car efficiency*. 38 models produced in both 1999 and 2008. Please type:

```
test <- mpg
```

I will explain this in detail later

Source: US Environment Protection Agency https://fueleconomy.gov/

mpg data

```
test <- mpg
```

Now, whenever we type, **`test`** , it will refer to the mpg data.

`test` is a data frame.

What is a data frame?

A data frame is a rectangular collection of variables (in columns) and observations (in rows).

id	gender	score
1	F	10.24
2	F	5.98
3	M	7.62

tibble = data frame

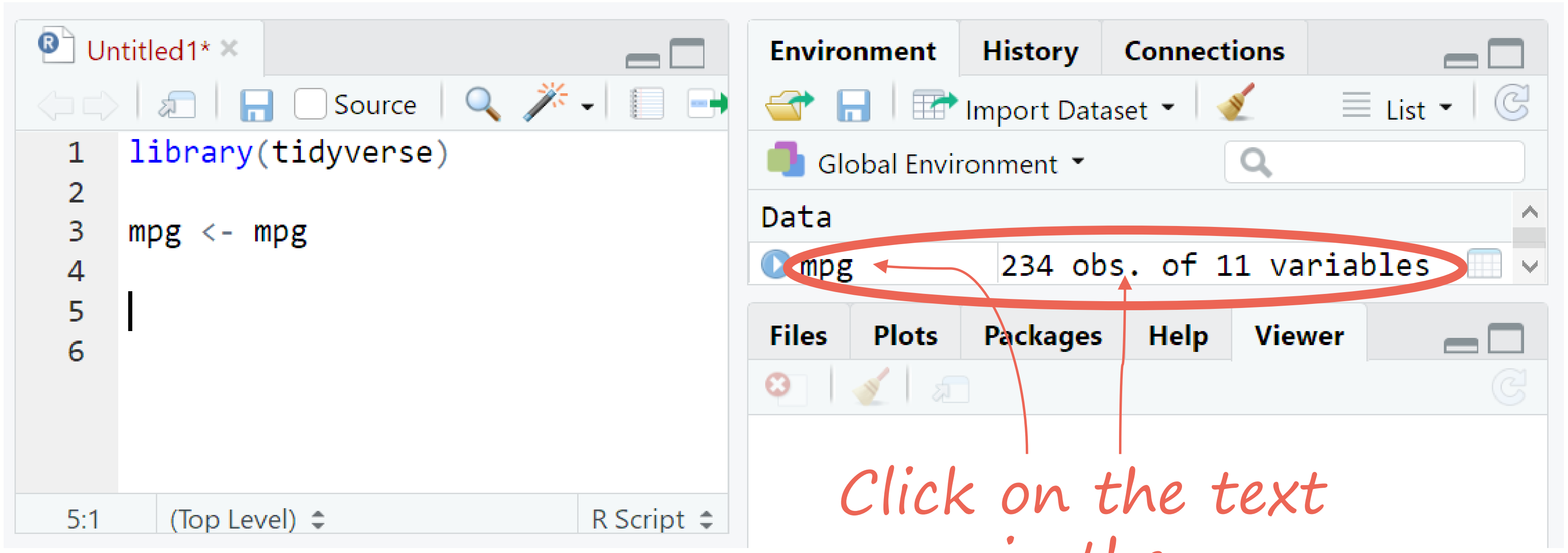
In the tidyverse you will see the term “tibble”.

We’ll take “tibble” to be synonymous with “data frame”.

id	gender	score
1	F	10.24
2	F	5.98
3	M	7.62

Viewing the data

Several ways to examine a data frame. Option 1:

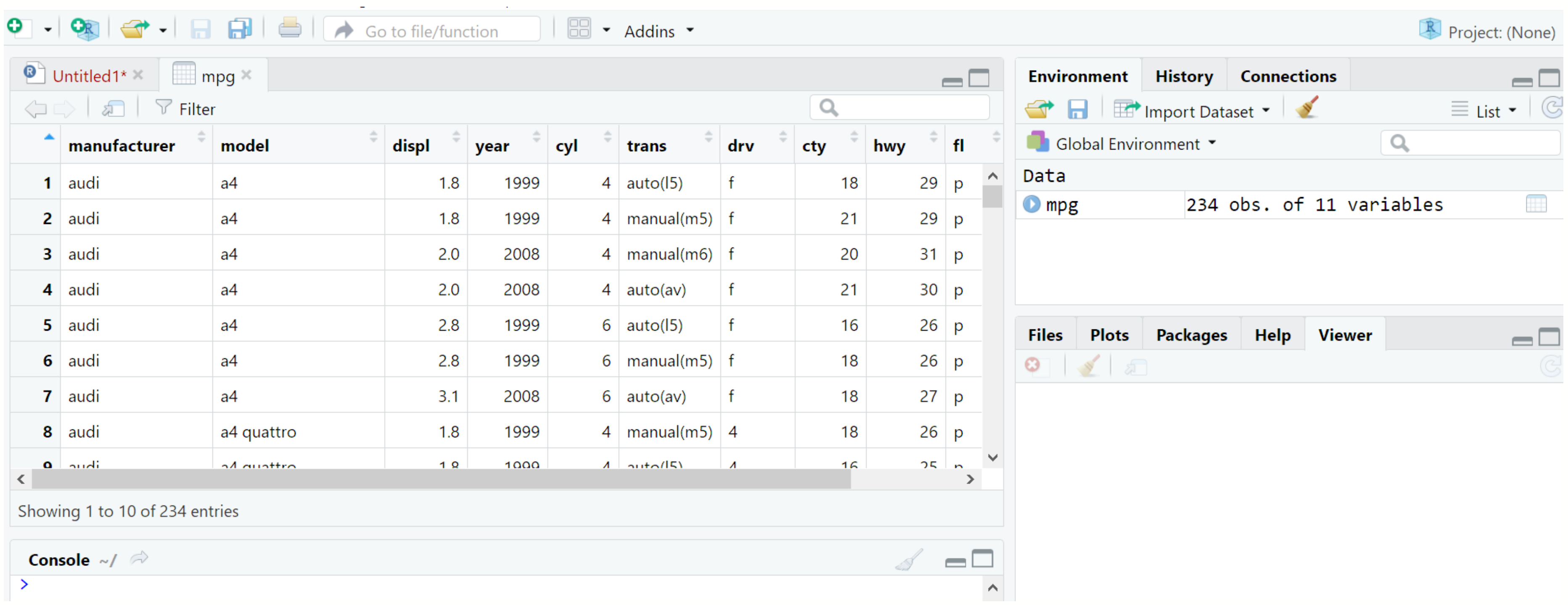


```
library(tidyverse)

mpg <- mpg

```

Click on the text in the Environment pane

Viewing the data

This brings up a view of the data in a new tab:



	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p

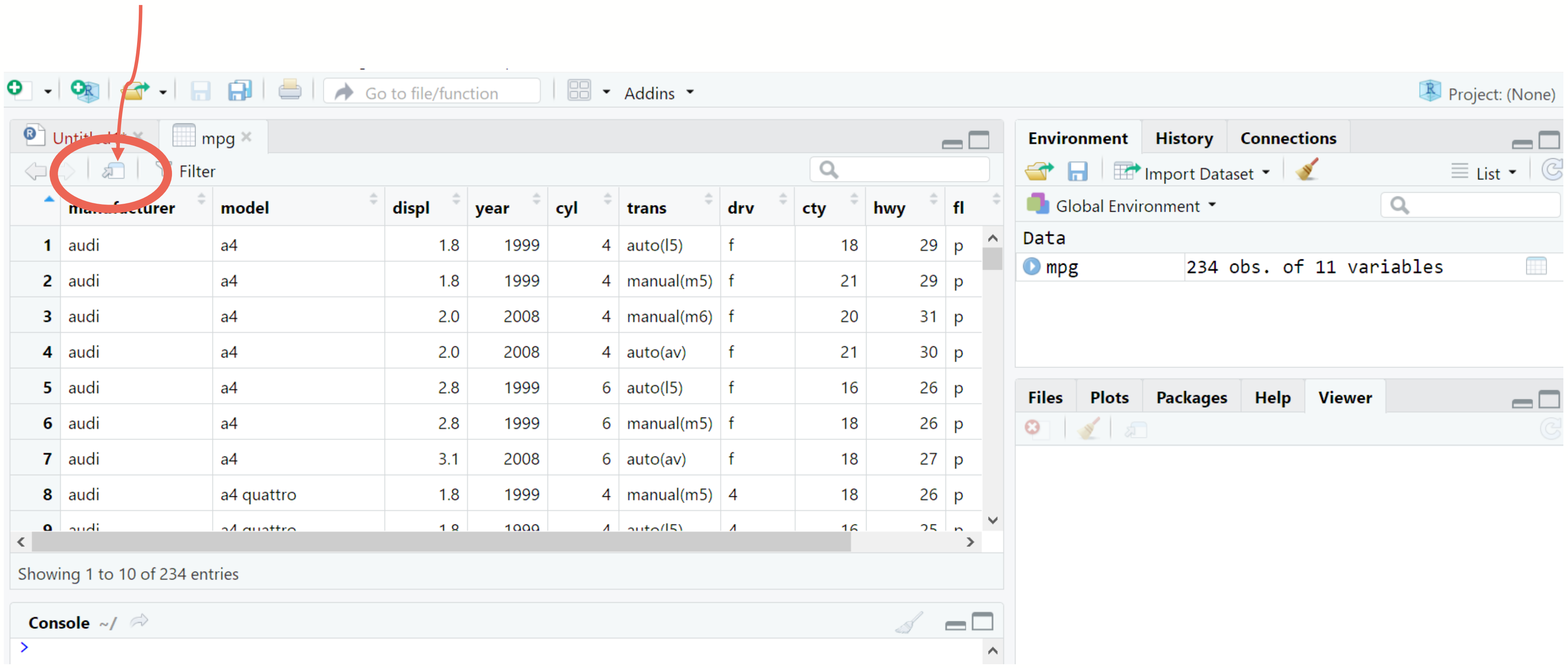
Showing 1 to 10 of 234 entries

```
Data
mpg    234 obs. of 11 variables
```

```
>
```

Viewing the data

Click here to show the data frame in a new window*



	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p

Showing 1 to 10 of 234 entries

Data

mpg 234 obs. of 11 variables

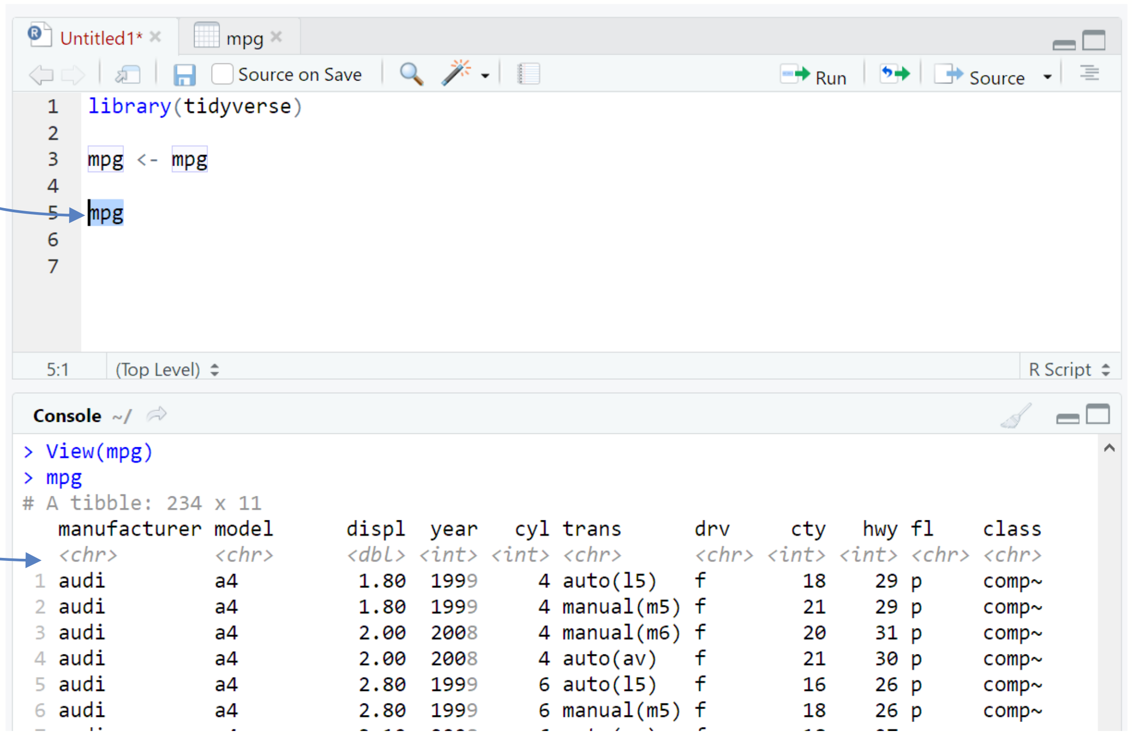
*Very useful with multiple monitors

Option 2: Preview in Console

Type the name of the dataset into editor/console, and run the line (Ctrl + Enter).

Run mpg

Prints data frame to console



```
library(tidyverse)

mpg <- mpg

mpg
```

```
> View(mpg)
> 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          1.80  1999     4 auto(l5)   f        18    29 p     comp~
2 audi         a4          1.80  1999     4 manual(m5) f        21    29 p     comp~
3 audi         a4          2.00  2008     4 manual(m6) f        20    31 p     comp~
4 audi         a4          2.00  2008     4 auto(av)   f        21    30 p     comp~
5 audi         a4          2.80  1999     6 auto(l5)   f        16    26 p     comp~
6 audi         a4          2.80  1999     6 manual(m5) f        18    26 p     comp~
```

Q. How many cars?
What variables do we have?

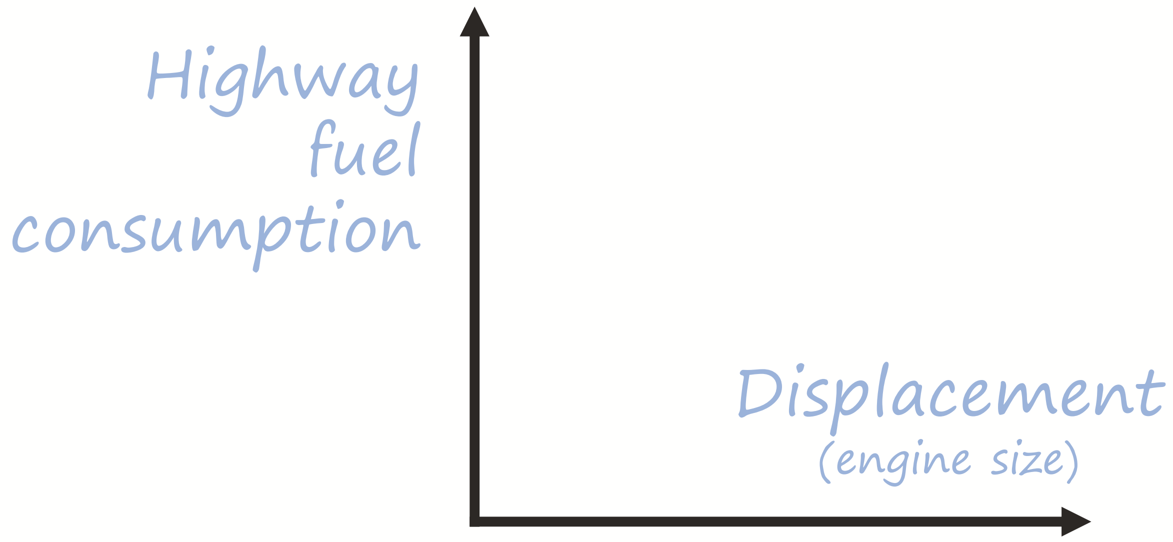
Graphics with ggplot2

The simple graph has bought more information to the data analyst's mind than any other device

- John Tukey

Q. Do cars with large engines (displ) use more fuel than cars with small engines?

Q. Do cars with large engines (displ) use more fuel than cars with small engines?



Q. Do cars with large engines (displ) use more fuel than cars with small engines?

Note that R is case sensitive

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

Breakdown

1. We begin our plot with ggplot()

2. Inside ggplot() we name our dataset

3. Next, we add layer(s) with +

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

How do we move from data to graphic?

Pen and paper exercise: Create a graphic from the data below.

year	time (sec)
1930	12.0
1960	11.3
1990	10.5

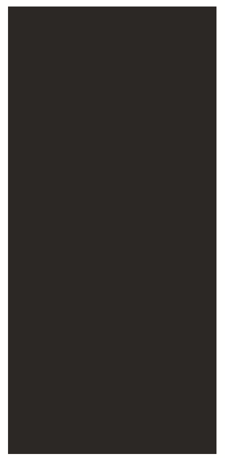
Pen and paper exercise: Create a graphic from the data below.

Now note down all the subtle (unconscious?) choices you made when creating the graphic.

year	time (sec)
1930	12.0
1960	11.3
1990	10.5

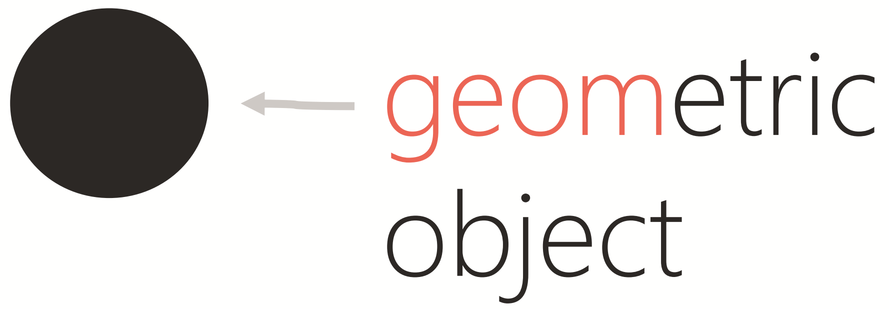
Choices

1. What shape will represent the data?



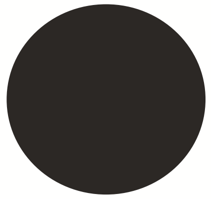
Choices

1. What shape will represent the data?



Choices

1. What shape will represent the data? (geom)
2. What visual (aesthetic) attributes do we give to the geom?



Choices

1. What shape will represent the data? (geom)
2. What visual (aesthetic) attributes do we give to the geom?

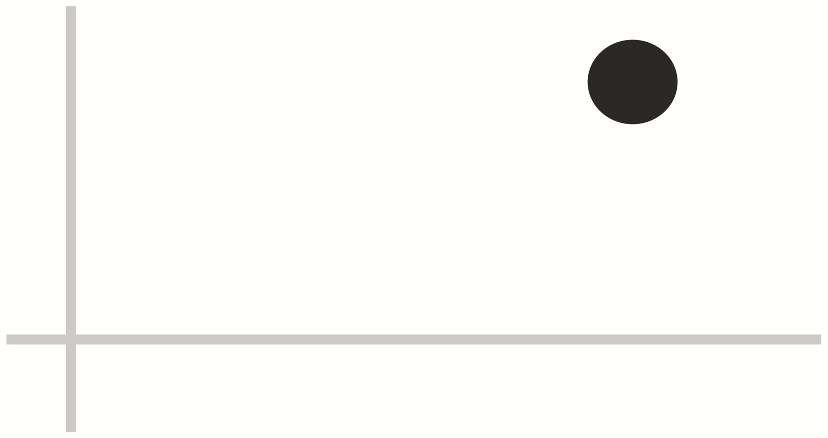
Choices

1. What shape will represent the data? (geom)
2. What visual (aesthetic) attributes do we give to the geom?

position (*x*)

Choices

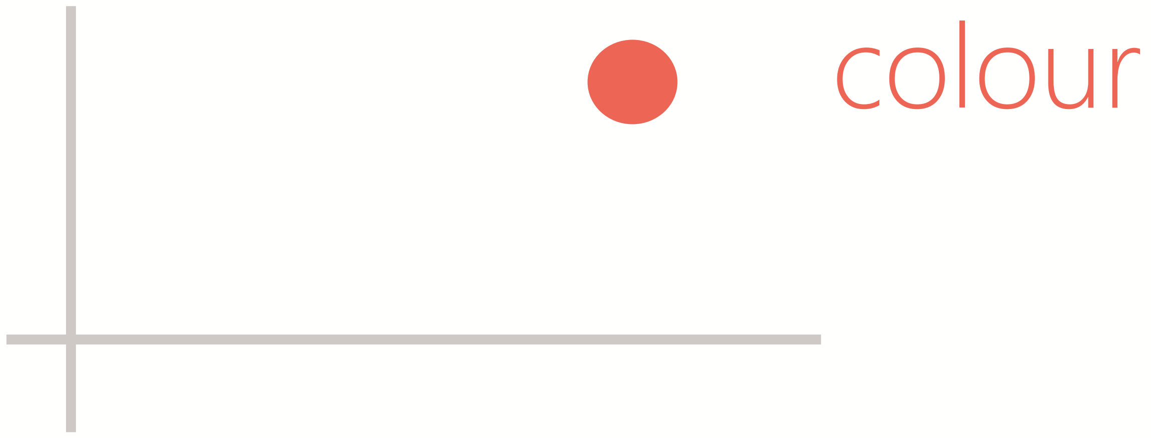
1. What shape will represent the data? (geom)
2. What visual (aesthetic) attributes do we give to the geom?



position (*y*)

Choices

1. What shape will represent the data? (geom)
2. What visual (aesthetic) attributes do we give to the geom?



A statistical graphic

Maps data variables to $\downarrow$ geometric objects.

aesthetic attributes of

A statistical graphic

Maps data variables to $\downarrow$ geometric objects.

aesthetic attributes of

```
ggplot(data = test) +

  geom_point(aes(x = displ, y = hwy))
```

Here, other aes() properties: size, colour, etc. are set by default

Functions ( )

`ggplot()`, **`geom_point()`**, and **`aes()`** are functions.

Arguments (inputs) in a function are separated by commas

Functions ( )

`ggplot()`, **`geom_point()`**, and **`aes()`** are functions.

Arguments (inputs) in a function are separated by commas

Here, we provide geom_point() with one argument : aes()

We give aes() two (explicit) arguments

```
ggplot(test) +
        geom_point(aes(displ, hwy))
```

Unspecified arguments revert to default values

Shorthand

As ggplot knows the order of essential arguments, I will use this convention from now on:

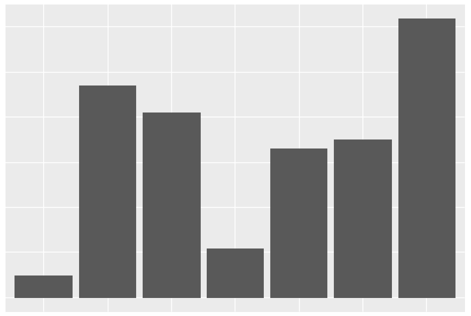
No need for "data ="

```
ggplot(test) +
     geom_point(aes(displ, hwy))
```

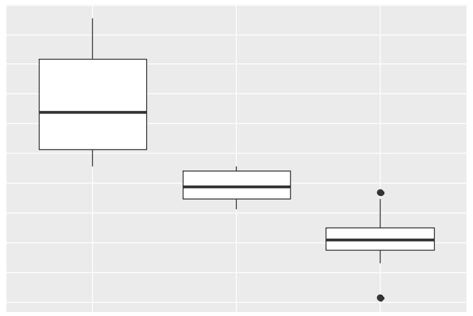
x goes first, y goes second

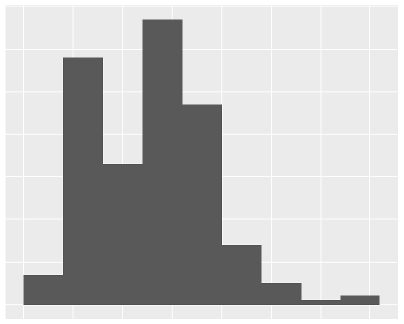
geoms

We tend to describe plots in terms of the geom used:


`geom_bar()`

`geom_line()`


`geom_boxplot()`


`geom_histogram()`

Layering geoms

We can display more than one geom in a plot:

```
ggplot(test) +
  geom_point(aes(displ, hwy)) +
  geom_line(aes(displ, hwy))
```

Note: Nonsense graphic: used to illustrate principle only

Layering geoms

We can display more than one geom in a plot:

```
ggplot(test) +
  geom_point(aes(displ, hwy)) +
  geom_line(aes(displ, hwy))
```

duplication!

Note: `geom_line` *used to illustrate principle only*

Layering geoms

To avoid duplication, we can pass the local **`aes()`** to **`ggplot()`**. This will make it a global value:

```
ggplot(test, aes(displ, hwy)) +
  geom_point() +
  geom_line()
```

In ggplot aes() goes second

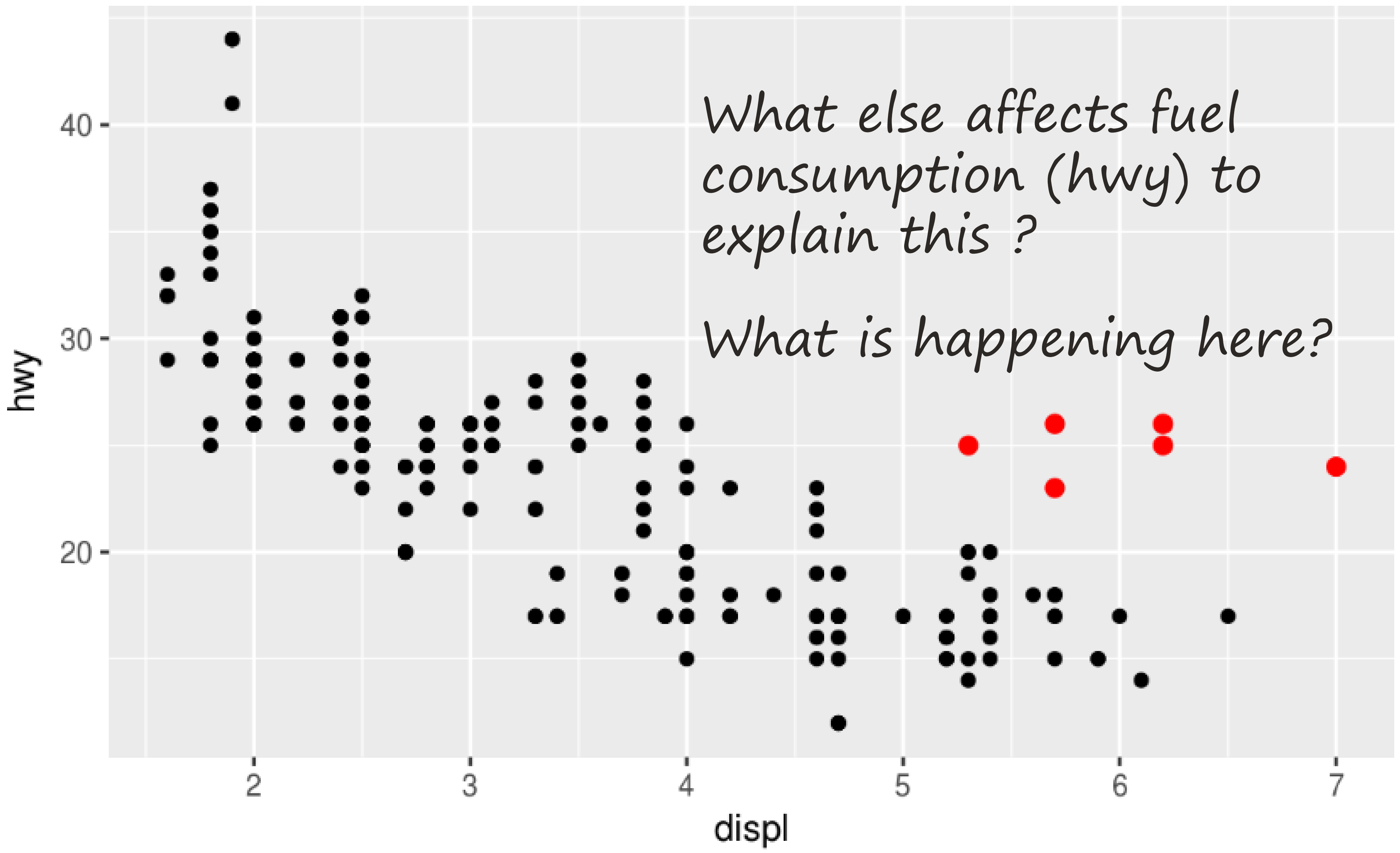
Note: `geom_line` used to illustrate principle only

Your turn

A **`geom_smooth()`** layer can help us identify patterns. Add geom_smooth to our original plot:

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

And (if you like) re-write in shorthand

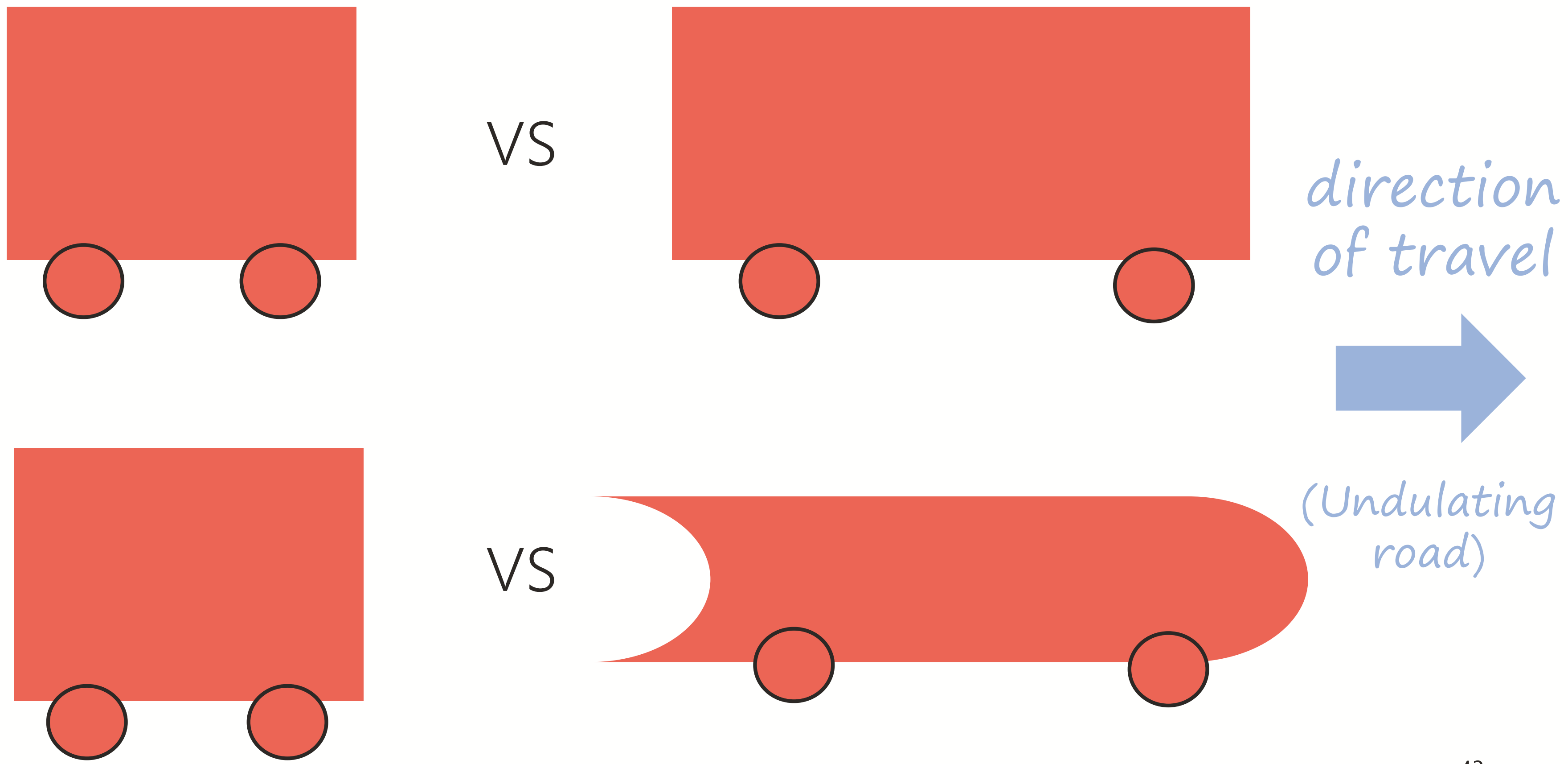


What else affects fuel consumption (hwy) to explain this ?

What is happening here?

Play your cars right

Same engine, material, speed. Which is more fuel efficient?



Hypothesis

Anomalous cars are lighter and/or aerodynamic.

Are they sports cars?

aesthetic attribute

We can map point colour to the class variable

- so a different colour for each class - to find out.

Adding another variable

Remember: Arguments within the **aes**thetic wrapper describe how variables are mapped:

```
ggplot(test) +
  geom_point(aes(displ, hwy, colour = class))
```

We could have chosen
size or shape
– but less clear graphic

Outcome

The anomalous points are (mostly) two-seater cars.

Likely to be sports cars, therefore more aerodynamic and lighter.

All red

If you wished to apply the same colour to all points, the colour **does not vary** *so* argument goes outside `aes()`:

```
ggplot(test) +
  geom_point(aes(displ,  hwy), colour = “red")
```

Small multiples

An alternative way to display additional variables is with small multiples. We do this with `facet_wrap()`

```
ggplot(test) +
  geom_point(aes(displ, hwy)) +
  facet_wrap(vars(class))
```

Small multiples

An alternative way to display additional variables is with small multiples. We do this with **`facet_wrap()`**

```
ggplot(test) +
  geom_point(aes(displ, hwy)) +
  facet_wrap(vars(class), nrow = 2)
```

facet_wrap is used with categorical variables

Demonstrating geoms:

(note these are simple, unpolished graphics)

Q. How are “cty” values distributed?

Histogram

```
ggplot(test, aes(cty)) +
  geom_histogram()
```

Q. How are “cty” values distributed?

Histogram

```
ggplot(test, aes(cty)) +
  geom_histogram(binwidth = 4)
```

Q. Distribution of engine size in each class?

Box plot

```
ggplot(test, aes(class, displ)) +
  geom_boxplot()
```

Q. Number of models by manufacturer?

Bar plot

```
ggplot(test, aes(manufacturer)) +
  geom_bar()
```

Q. Number of models by manufacturer?

Bar plot - flipped

```
ggplot(test, aes(manufacturer)) +
   geom_bar()+
   coord_flip()
```

Two variable bar plot

(more common than `geom_bar`)

```
ggplot(test, aes(manufacturer, hwy)) +
         geom_col()
```

Reorder a two variable bar plot:

Name of variable by which to reorder x

```
ggplot(data, aes(reorder(x, a), y)) +
  geom_col()
```

Plot labels

```
ggplot(mpg, aes(class, displ)) +
  geom_violin()+
  labs(title = "Displacement by class",
       subtitle = “Any subtitle",
       y = "Displacement",
       caption = "Source: US EPA")
```

or: pdf, jpg as
you wish

ggsave("plot_name.png")

By default:

- saves most recent ggplot to your working directory
- saves a plot in the same dimensions as plot window

Tip for now: adjust dimensions of plot pane in RStudio as you wish, then save.

Save your script!

Think of your script as the “real” part of your analysis.

File → Save As… → ggplot_intro.R

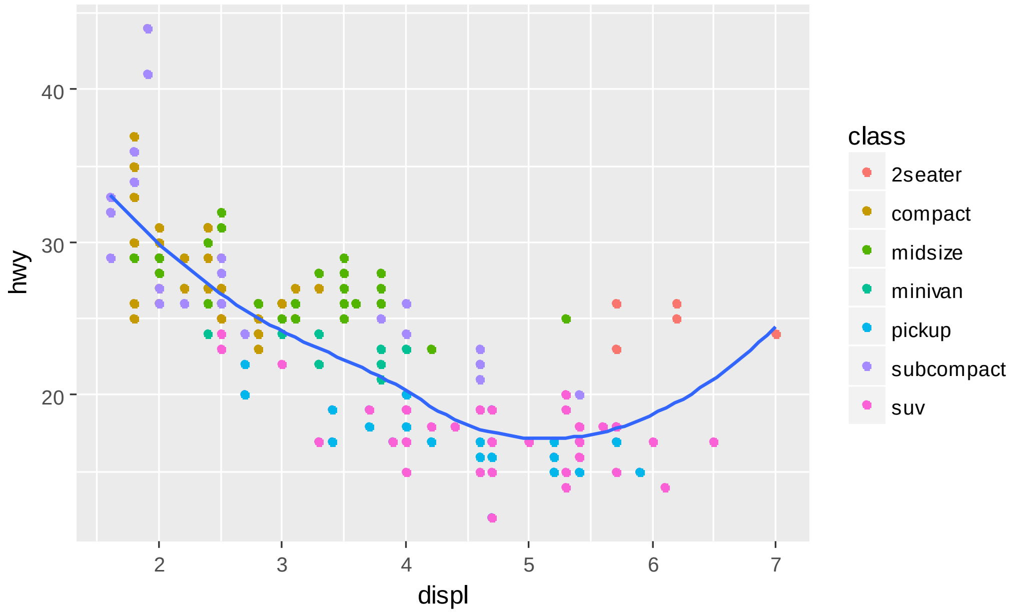
accidental aRt

https://twitter.com/accidental__aRt

Addendum:
A review of local and global aesthetics

Two layers:

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



Duplicate aes attributes:

Ideally, extract similar elements and put them in the global aes()

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

Global and local

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

Note that `colour = class` *must remain within aes() of geom_point, or it will be applied to geom_smooth*

End