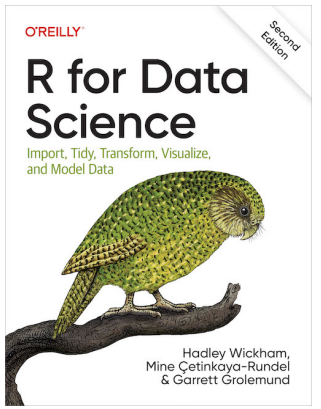
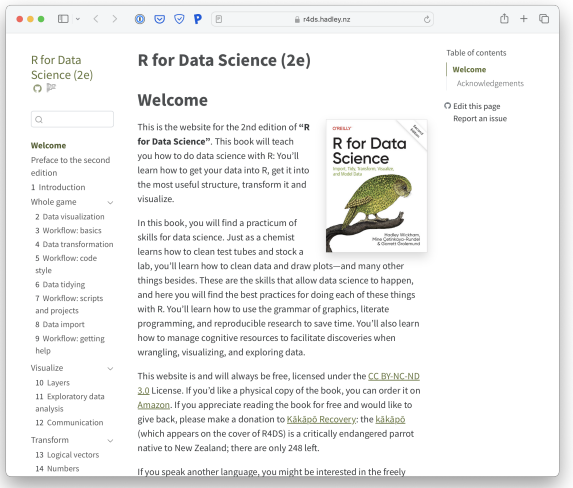
Data wrangling & visualization I — ggplot2

Week 1

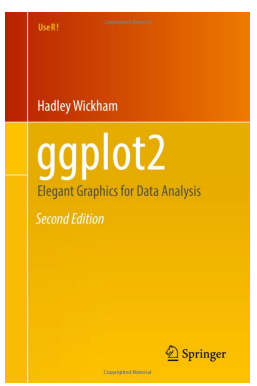
R/RStudio books

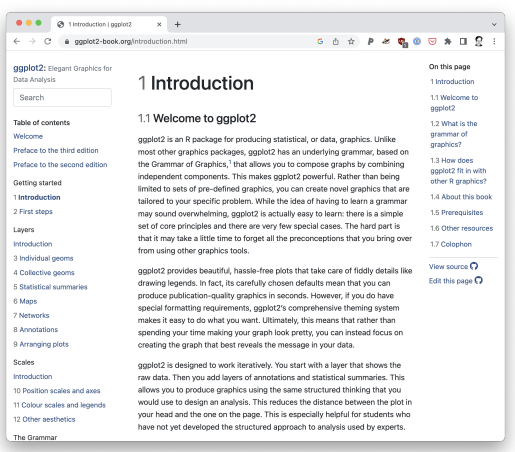
- R for Data Science
  by Hadley Wickham &
  Garrett Grolemund



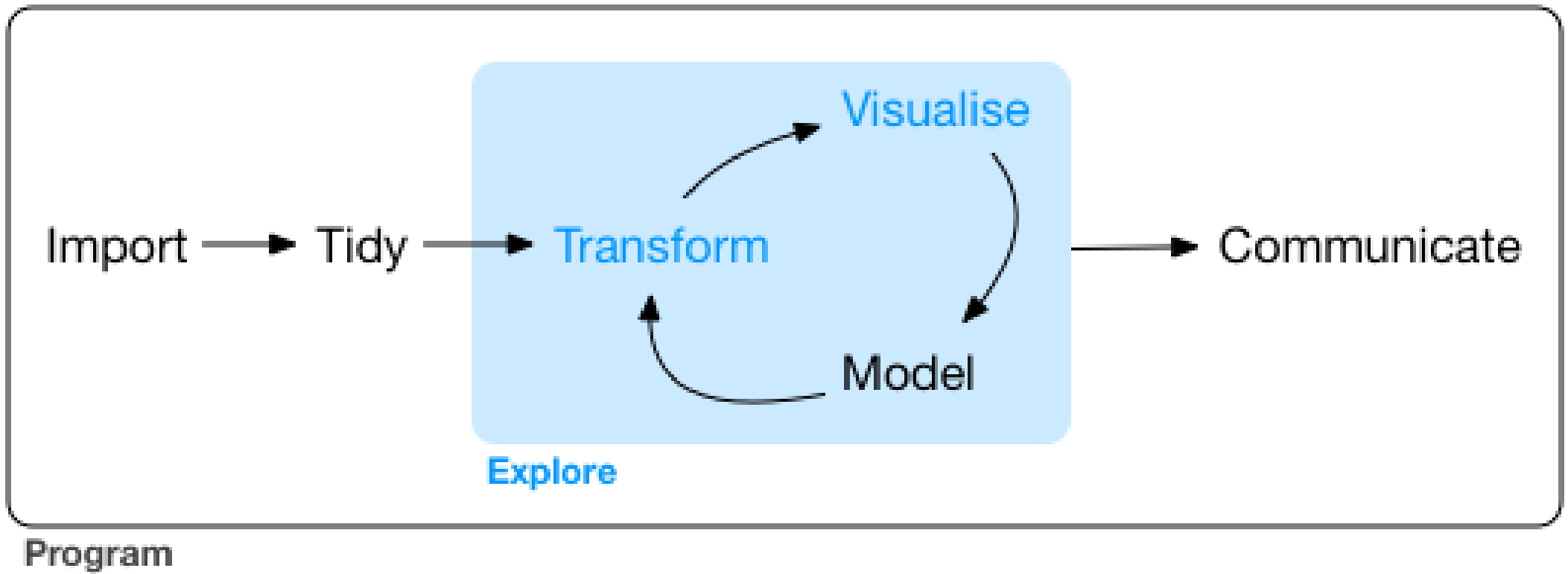


- ggplot2: Elegant Graphics
  for Data Analysis
  by Hadley Wickham



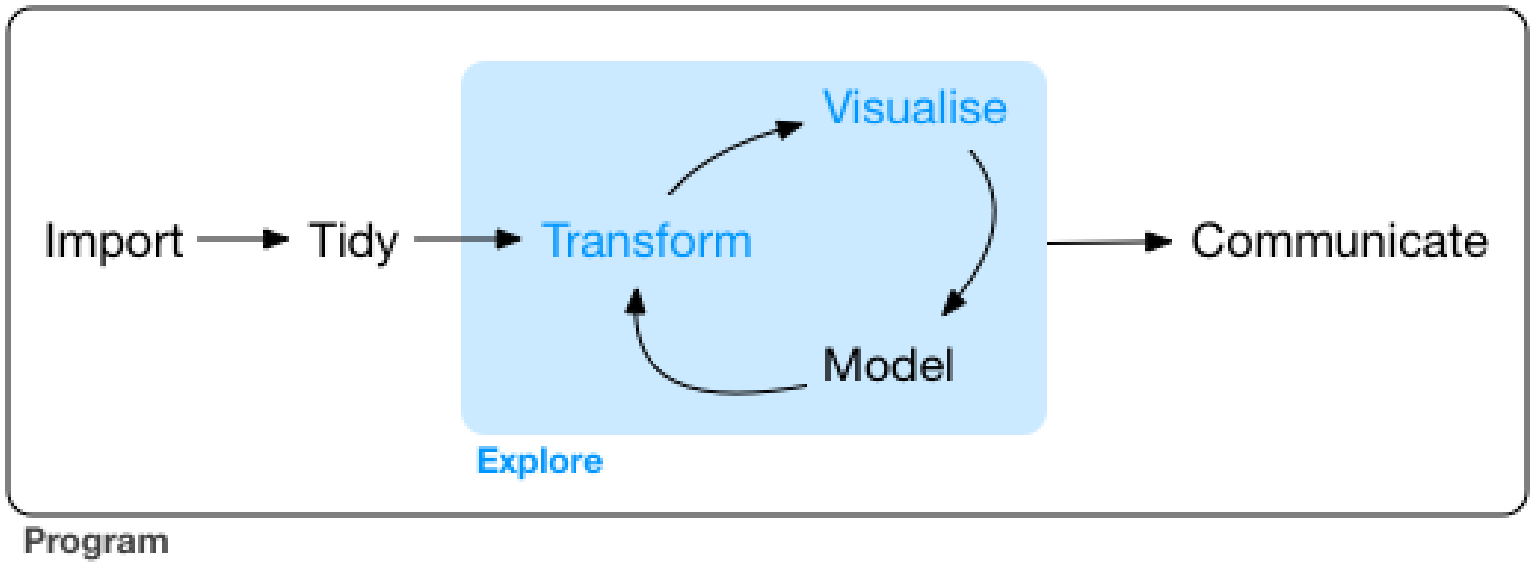


Data Visualization



- **this week**: we will learn the basic structure of a ggplot2 plot

Data Transformation



- **next week**: we will learn the key verbs (R commands) to:
  - select variables
  - filter out observations
  - create new variables
  - compute summaries

Why make plots?

- numeric summaries of data are easy to generate
- mean, sd, correlation, list of t-tests, etc…
- but numerical summaries are just **summaries**
- they can obscure patterns in the underlying data

	fdi	open	gdpgrow	govcon	wages	vinflat	inflat	educat	indust	return	infras	risk	trend	demindex	size
fdi	1.0000														
open	0.4068 0.0001	1.0000													
gdpgrow	0.1115 0.0001	0.1058 0.0001	1.0000												
govcon	0.0575 0.0022	0.2483 0.0001	-0.0510 0.0065	1.0000											
wages	0.0172 0.3571	0.0686 0.0003	0.0876 0.0001	-0.1918 0.0001	1.0000										
vinflat	-0.0239 0.2022	-0.0624 0.0009	-0.1168 0.0001	0.0121 0.5180	-0.1571 0.0001	1.0000									
inflat	-0.0807 0.0001	-0.0559 0.0029	-0.2097 0.0001	0.0134 0.4742	-0.1328 0.0001	0.5591 0.0001	1.0000								
educat	0.1016 0.0001	0.2006 0.0001	-0.0936 0.0001	0.2780 0.0001	-0.5131 0.0001	0.0888 0.0001	0.0496 0.0082	1.0000							
indust	0.0487 0.0094	0.2225 0.0001	0.0321 0.0864	0.1218 0.0001	-0.2658 0.0001	0.1222 0.0001	0.1088 0.0001	0.3099 0.0001	1.0000						
return	0.0623 0.0009	-0.0428 0.0224	0.0900 0.0001	0.0114 0.5438	0.0556 0.0030	-0.1704 0.0001	-0.3142 0.0001	-0.0125 0.5030	-0.0895 0.0001	1.0000					
infras	0.0971 0.0001	0.1421 0.0001	-0.0672 0.0003	0.3421 0.0001	-0.5049 0.0001	-0.0246 0.1891	-0.0513 0.0062	0.8044 0.0001	0.0685 0.0003	0.0166 0.3763	1.0000				
risk	-0.0517 0.0058	-0.0505 0.0070	-0.1140 0.0001	0.0249 0.1845	-0.0742 0.0001	0.3747 0.0001	0.4878 0.0001	0.0204 0.2754	0.0287 0.1259	0.1982 0.0001	-0.0456 0.0150	1.0000			
trend	0.2164 0.0001	0.1236 0.0001	-0.1500 0.0001	0.0289 0.1230	-0.1375 0.0001	0.0702 0.0002	0.0344 0.0661	0.3015 0.0001	0.0388 0.0383	0.1007 0.0001	0.2609 0.0001	0.0263 0.1600	1.0000		
demindex	0.1124 0.0001	0.0944 0.0001	-0.0939 0.0001	0.3340 0.0001	-0.4413 0.0001	-0.0071 0.7047	0.0071 0.7022	0.7320 0.0001	0.0619 0.0010	0.0710 0.0001	0.7750 0.0001	0.0183 0.3278	0.2502 0.0001	1.0000	
size	-0.0246 0.1893	-0.1409 0.0001	-0.0570 0.0024	0.1544 0.0001	-0.3552 0.0001	-0.0211 0.2602	-0.0307 0.1018	0.4178 0.0001	0.0399 0.0333	0.0180 0.3371	0.5044 0.0001	-0.0400 0.0330	0.1092 0.0001	0.4241 0.0001	1.0000

year	nobs	fdi	open	gdpgrow	govcon	wages	vinflat	inflat	educat	tyr	indust	return	infras	risk	ict	fuel	mineral	demindx	size
A. Mean by year of observation:																			
1992	39	1.65	76.42	1.61	17.53	15.55	5255.80	16.62	88.82	8.20	32.61	8.37	299.87	8.40	4.55	0.90	0.72	26.78	300
1993	39	1.88	74.82	1.91	17.93	15.22	5255.80	28.83	92.85	8.20	31.89	7.80	310.15	7.51	4.71	0.92	0.69	27.18	301
1994	39	2.13	76.02	3.86	17.39	14.97	5255.80	18.42	94.21	8.59	31.41	6.22	322.13	7.22	4.74	0.95	0.72	27.60	309
1995	39	2.46	79.20	3.79	16.92	14.86	5255.80	15.34	95.95	8.63	31.41	7.87	335.95	11.45	4.76	0.95	0.72	27.85	317
1996	39	2.38	80.24	3.28	16.82	14.51	5255.80	12.73	96.35	8.62	31.37	7.81	349.93	8.28	4.93	1.00	0.69	28.41	323
1997	39	2.89	83.35	3.84	16.80	14.61	5255.80	24.53	95.74	8.60	31.13	4.50	365.05	5.68	5.19	1.00	0.67	27.94	333
1998	38	3.78	87.84	1.63	16.89	14.10	5393.37	7.88	95.76	8.85	31.02	6.97	374.52	5.01	5.94	0.97	0.68	27.55	345
1999	36	4.96	88.82	2.48	16.88	13.92	5691.71	6.80	94.64	8.69	30.98	6.17	372.96	5.25	6.26	0.92	0.64	26.96	367
B. Mean by region:																			
EastAsPa	32	2.69	104.95	4.41	10.26	24.94	64.97	8.15	59.76	6.21	38.92	6.39	67.43	3.06	3.21	1.00	1.00	8.60	13
EurCenAs	48	2.76	89.07	0.53	16.97	10.06	21288.00	68.11	89.55	10.37	36.26	6.99	231.65	19.79	3.91	0.73	0.50	32.15	9
Europe	172	2.79	85.80	2.94	20.09	12.75	374.67	3.40	111.11	9.17	28.27	6.49	496.53	4.72	6.12	0.99	0.59	33.57	48
LatinACa	40	3.25	43.11	3.52	11.11	18.26	13773.80	20.82	59.71	6.78	32.92	9.21	127.08	8.30	4.22	0.98	1.00	18.73	16
MidENAfr	8	1.27	49.00	4.76	10.27	18.29	37.49	8.59	76.73	4.94	32.34	7.14	50.55	5.16	1.98	1.00	1.00	3.25	6
SubSaAfr	8	0.71	44.91	1.75	19.62	23.14	13.57	8.65	89.48	5.90	34.08	8.01	103.73	4.86	6.30	1.00	1.00	14.31	15
C. Mean by level of income:																			
HighOECD	148	2.66	70.84	2.59	20.18	11.42	22.46	2.60	115.63	9.17	27.23	6.47	514.41	4.31	6.18	0.99	0.63	34.83	54
HighOECD	24	3.61	178.08	5.09	19.55	20.92	2546.63	8.38	83.24	9.17	34.71	6.62	386.33	7.25	5.76	1.00	0.33	25.80	6
Lowincom	8	1.06	59.02	3.76	7.52	13.03	130.59	16.00	49.25	4.52	42.21	6.24	18.79	2.08	1.96	1.00	1.00	4.01	19
Lowmdinc	48	1.85	70.46	1.21	13.15	18.84	16592.80	64.53	71.34	7.34	34.16	9.50	124.43	20.15	2.93	0.79	0.83	17.45	12
Uppmdinc	80	3.34	78.19	3.39	14.19	16.68	9722.08	15.94	76.39	8.32	35.71	6.55	168.39	5.95	4.61	0.95	0.80	22.98	12
D. Mean by indebtedness:																			
Debncla	172	2.79	85.80	2.94	20.09	12.75	374.67	3.40	111.11	9.17	28.27	6.49	496.53	4.72	6.12	0.99	0.59	33.57	48
Leasndeb	48	2.13	73.00	3.03	16.66	15.67	4910.21	13.36	84.49	8.47	34.67	6.66	151.57	6.34	4.23	0.94	0.67	22.66	11
Modindeb	64	3.30	80.94	2.57	11.67	20.40	9783.95	39.28	67.26	7.34	36.65	10.14	131.32	14.62	4.04	0.92	0.88	17.55	12
Sevindeb	24	2.11	59.37	2.08	11.66	11.91	29725.10	56.05	65.37	7.43	34.35	2.56	163.10	9.17	2.66	0.75	1.00	20.70	16
E. Mean by HIPC group:																			
EMU	80	2.65	78.91	2.58	20.33	13.11	26.23	2.54	114.71	8.15	27.86	5.94	467.76	4.59	5.36	0.99	0.61	36.18	70
Others	228	2.78	81.38	2.89	16.04	15.30	7182.50	21.42	87.12	8.68	32.76	7.33	296.39	8.36	5.04	0.94	0.72	24.51	19
E. Sample mean and standard deviation:																			
Mean	308	2.74	80.74	2.81	17.15	14.73	5323.73	16.51	94.28	8.55	31.48	6.97	340.90	7.38	5.12	0.95	0.69	27.54	324
Std De	308	2.96	55.21	3.90	5.67	7.73	14326.2	62.57	26.33	2.12	5.97	9.28	199.45	15.59	2.02	0.22	0.46	12.11	484

Why make plots?

- numeric summaries of data are easy to generate
- mean, sd, correlation, list of t-tests, etc…
- but numerical summaries are just **summaries**
- they can obscure patterns in the underlying data

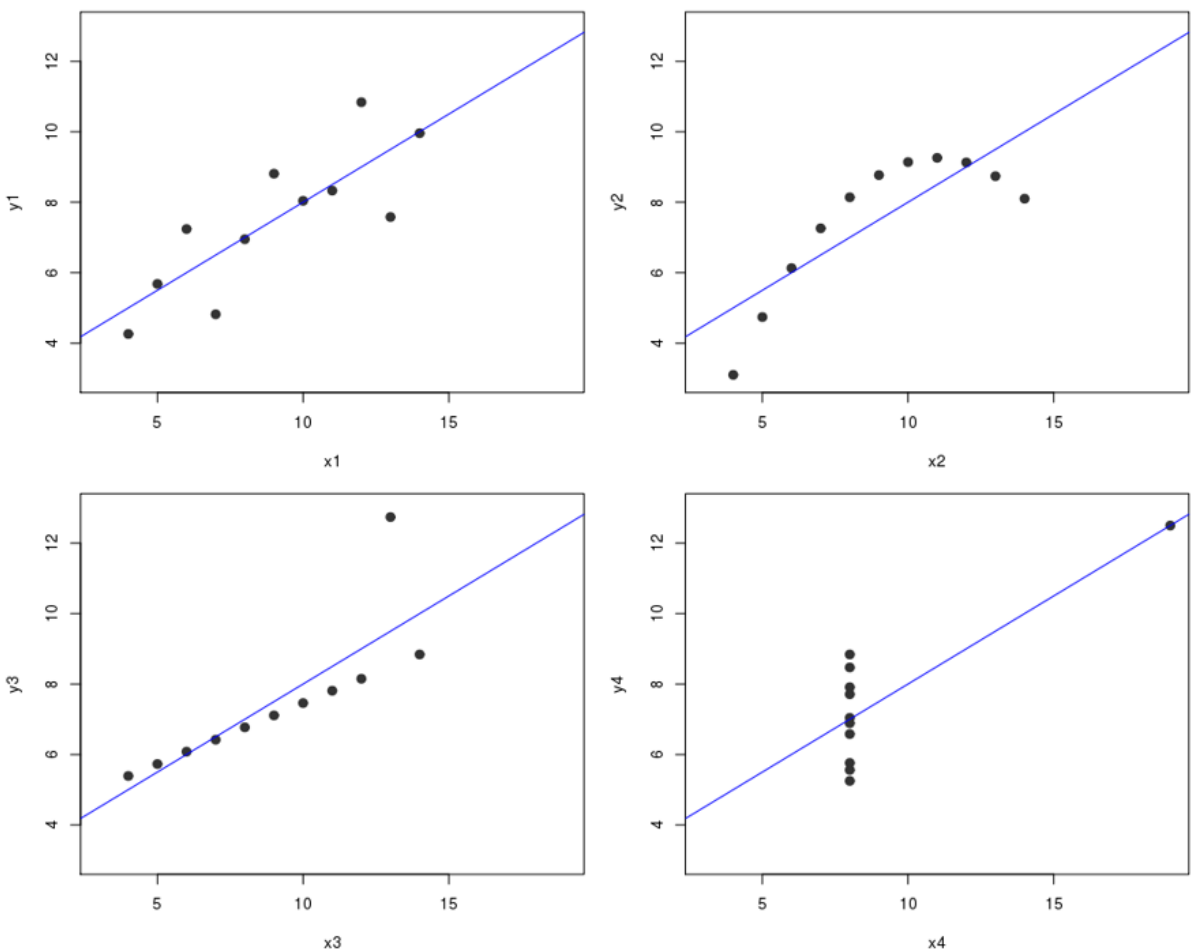
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA
ALWAYS PLOT YOUR DATA

Anscombe's Quartet

Datasets with the same statistical properties (Anscombe's quartet)

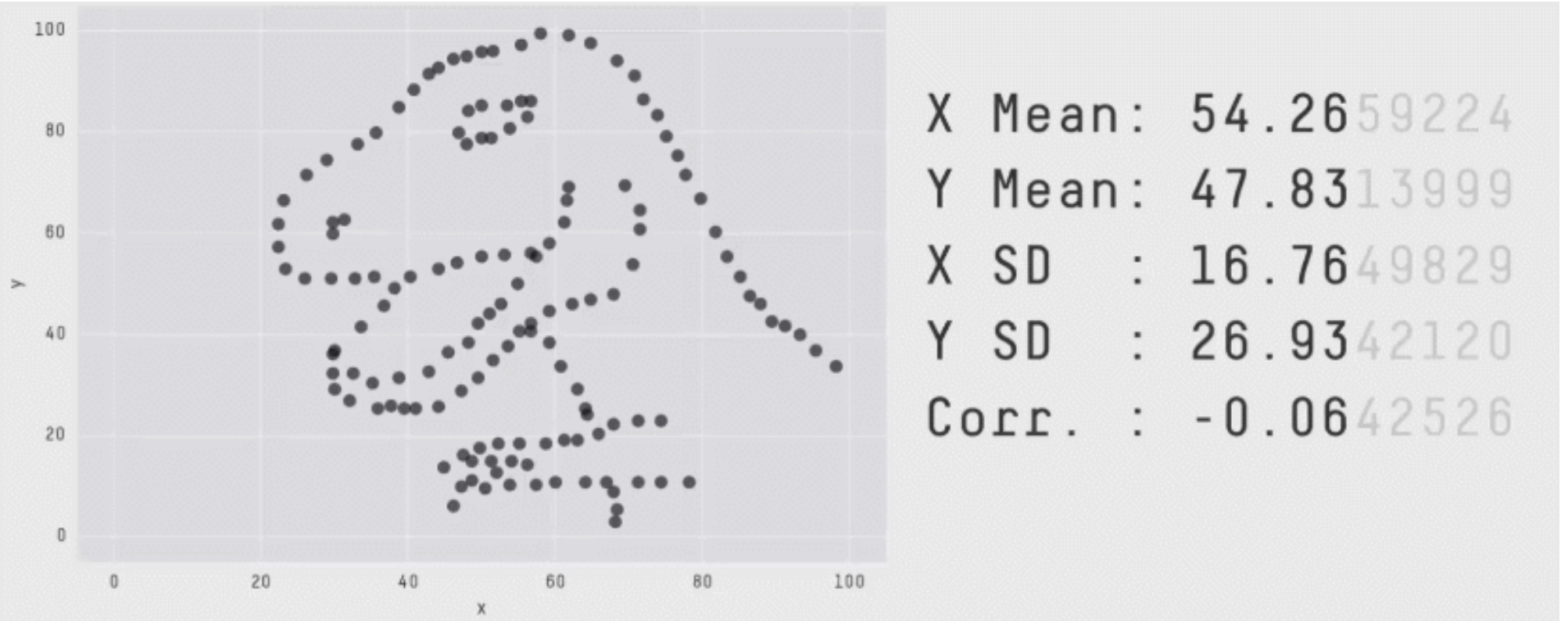
For all 4 of the datasets:

mean of x is 9
mean of y is 7.5
variance is 11
correlation is 0.816
linear regression: y = 3 + 0.5x



Datasaurus Dozen

Datasaurus Dozen



Why make plots using code?

- repeatable
- extensible
- sharable
- durable

Prerequisites

- only once:

```
1 install.packages("tidyverse")
```

- every time you start RStudio:

```
1 library(tidyverse)
```

Sample dataset: mpg

- the `ggplot2` package (which comes with `tidyverse`) includes:
  - a **tibble** called `mpg`
  - type `?mpg` in the RStudio console to get a help page on the `mpg` dataset
  - type `mpg` to see the first few rows

```
1 mpg
```

```
# A tibble: 234 × 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
```

Sample dataset: mpg

- the `ggplot2` package (which comes with `tidyverse`) includes:
  - a **tibble** called `mpg`
  - type `nrow(mpg)` to count the number of rows
  - type `ncol(mpg)` to count the number of cols

```
1 nrow(mpg)
```

```
[1] 234
```

```
1 ncol(mpg)
```

```
[1] 11
```

Sample dataset: mpg

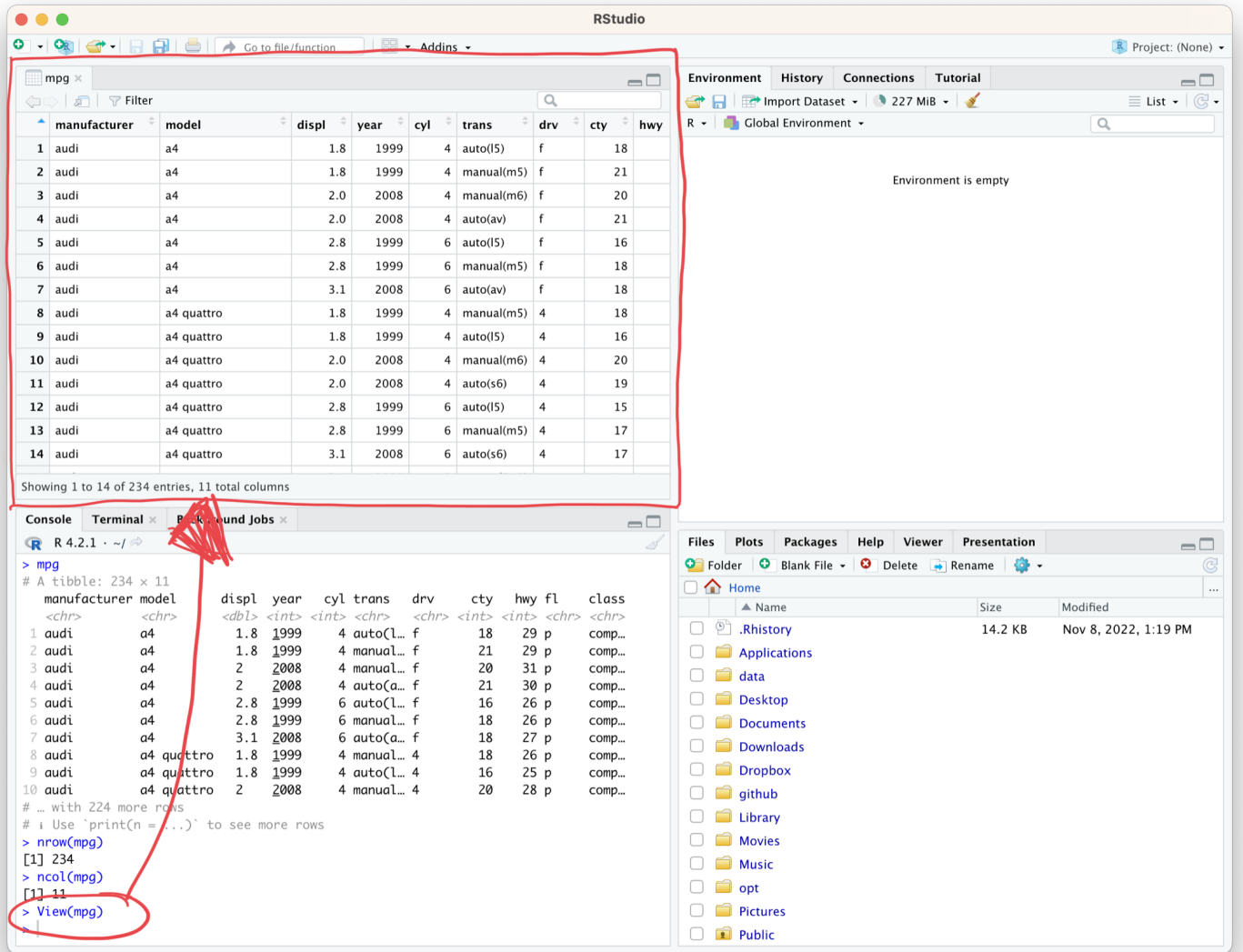
- you can also see a view of a data frame using `glimpse()`:

```
1 glimpse(mpg)
```

```
Rows: 234
Columns: 11
$ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "…
$ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "…
$ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.…
$ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200…
$ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, …
$ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto…
$ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4…
$ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1…
$ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2…
$ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p…
$ class        <chr> "compact", "compact", "compact", "compact", "compact", "c…
```

Sample dataset: mpg

- type `View(mpg)` to bring up a spreadsheet-like view of the data frame



Q: Do big engines use more fuel?

```
1 mpg
```

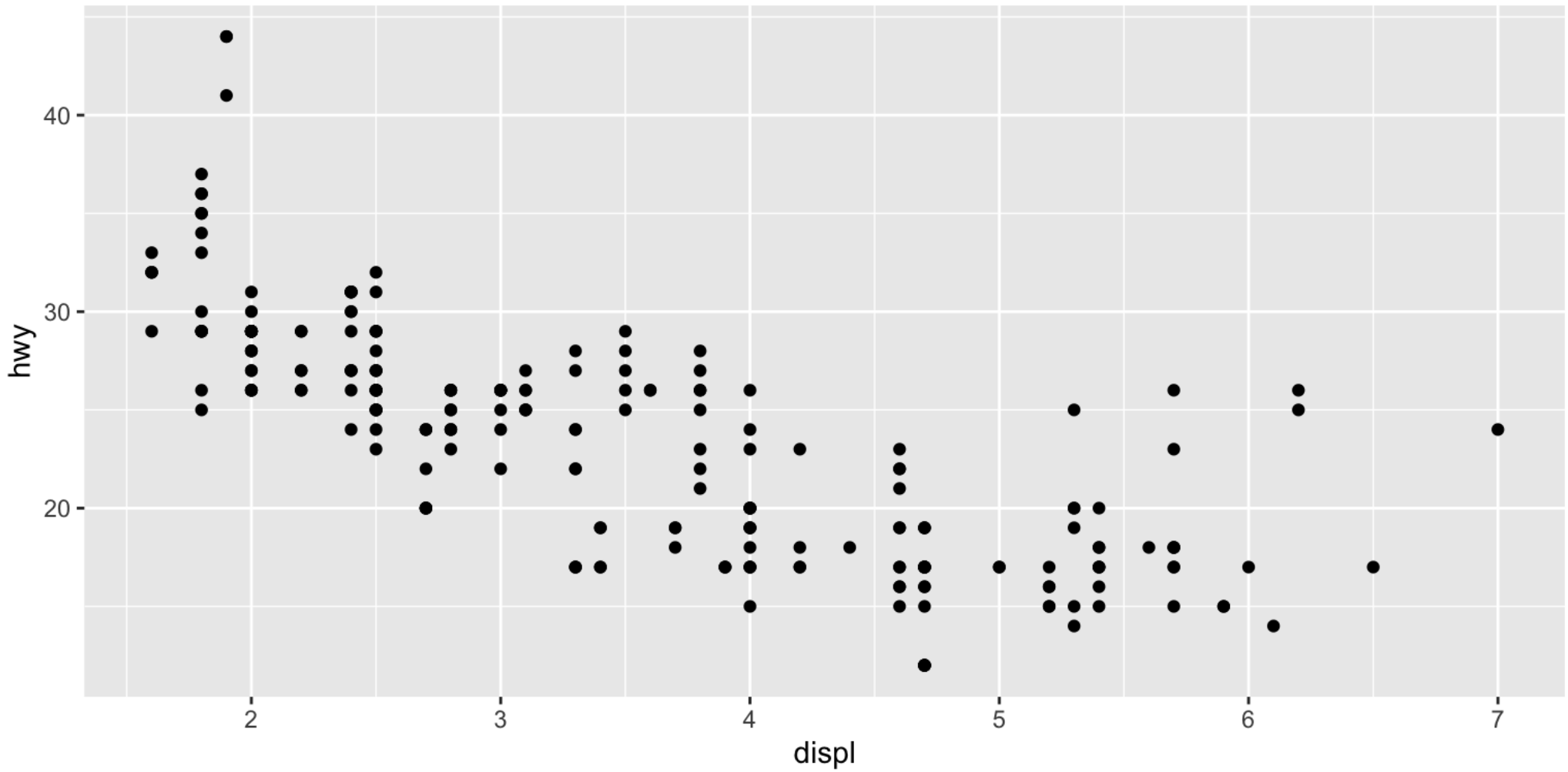
```
# A tibble: 234 × 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
 2 audi         a4           1.8  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…
# … with 224 more rows
```

- `displ`: engine size, in litres
- `hwy`: fuel efficiency (highway), in miles per gallon (mpg)

Creating a `ggplot`

Creating a ggplot

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

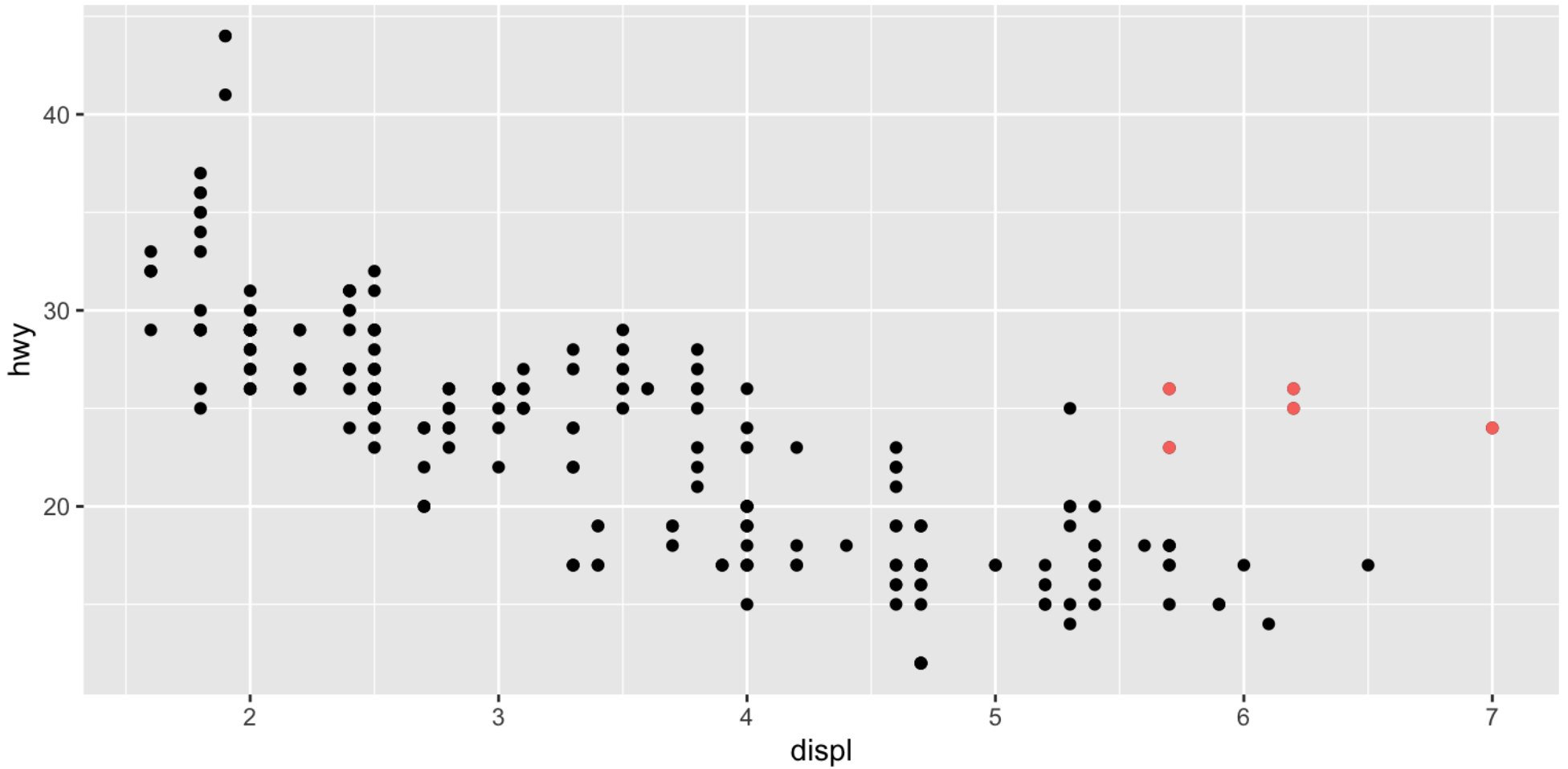


Creating a `ggplot`

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

- `ggplot(data = mpg)`
  - creates a coordinate system you can add layers to
- `geom_point()`
  - adds a layer of **points** to your plot
  - `mapping = aes(x = displ, y = hwy)`
    - tells `geom_point` to map `displ` values on to X-axis and `hwy` values onto Y-axis

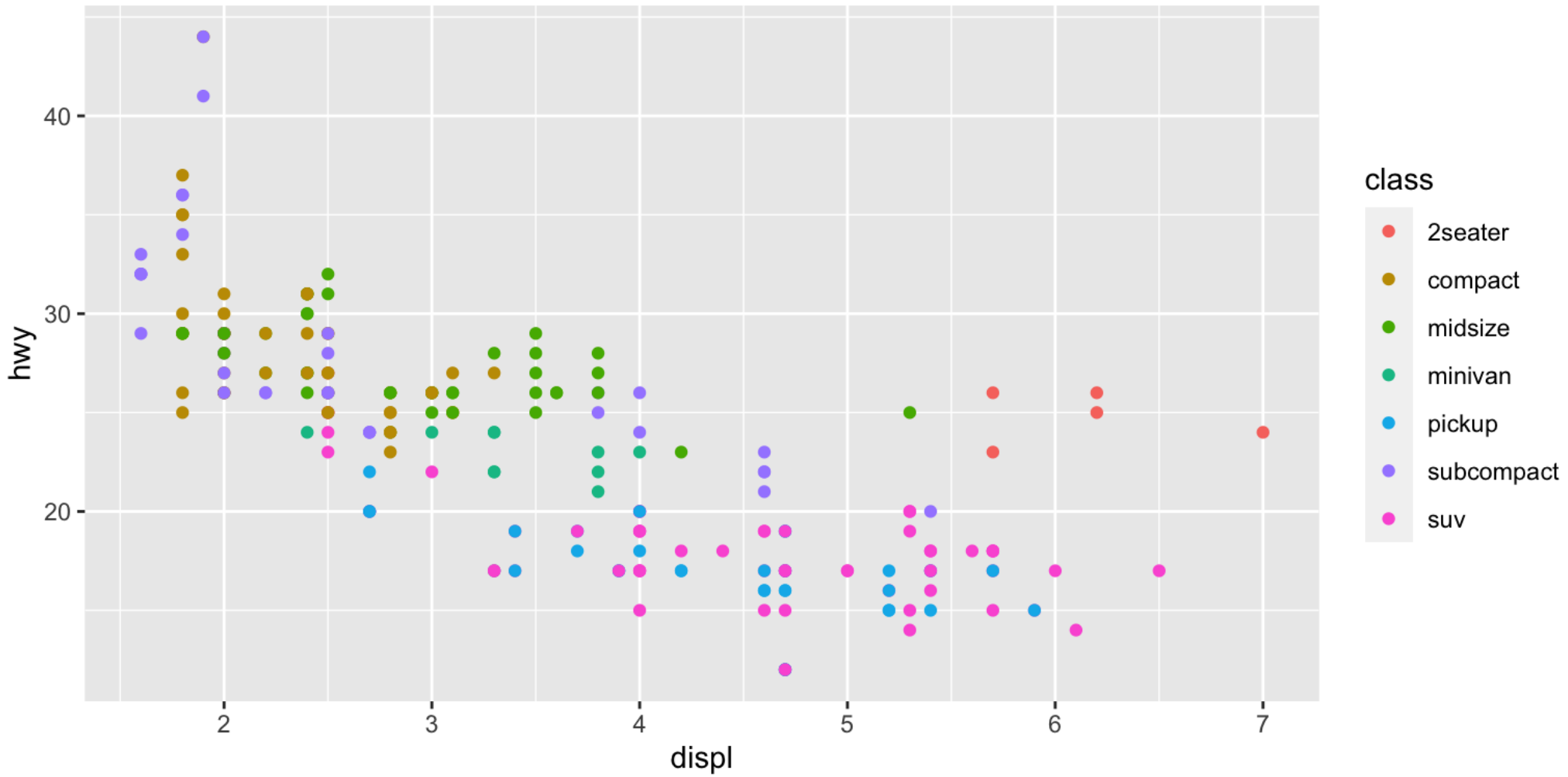
Colour-code by another variable



- can we explain why the cars shown in red don't follow the trend?

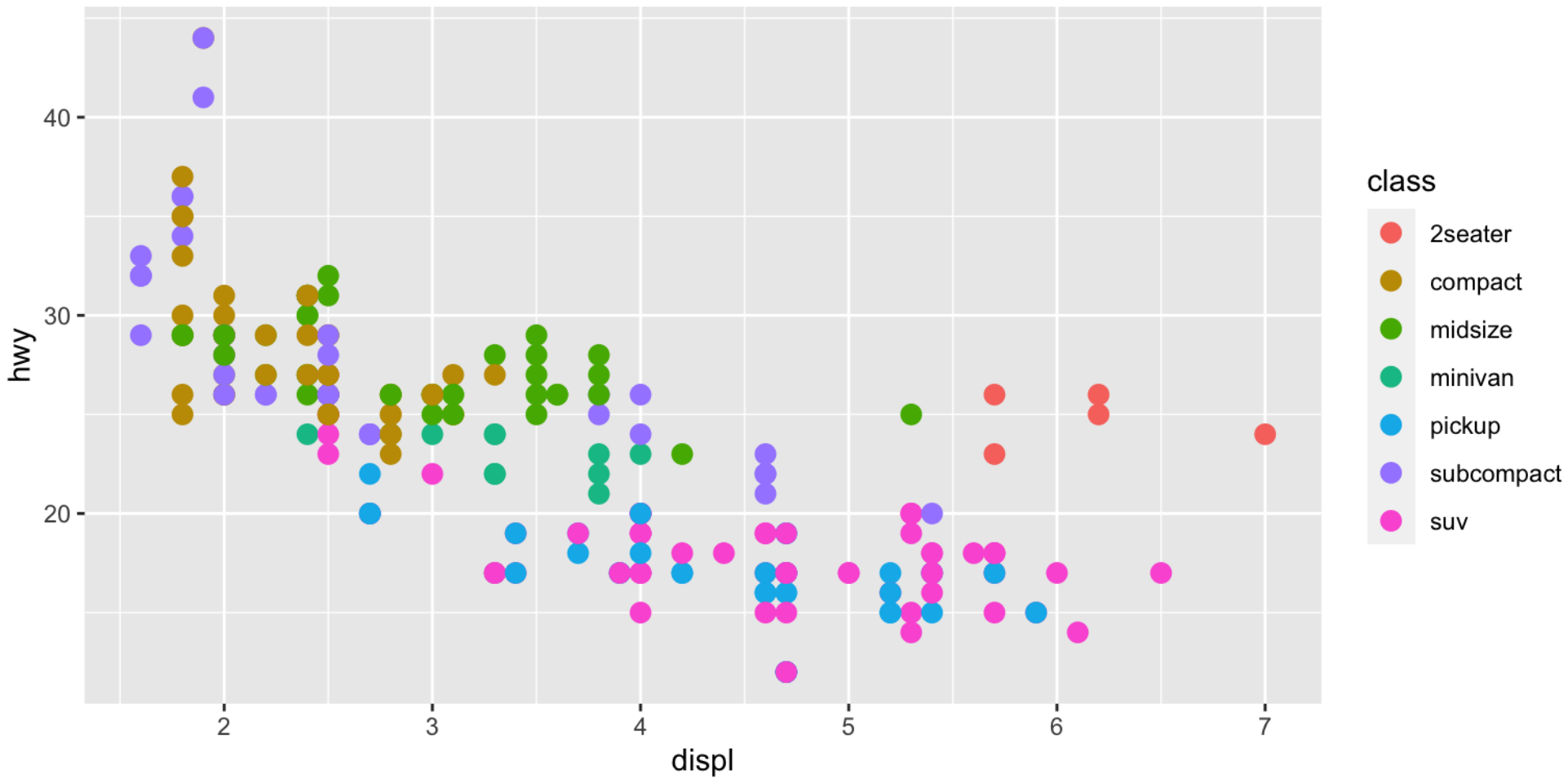
Colour-code by another variable

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



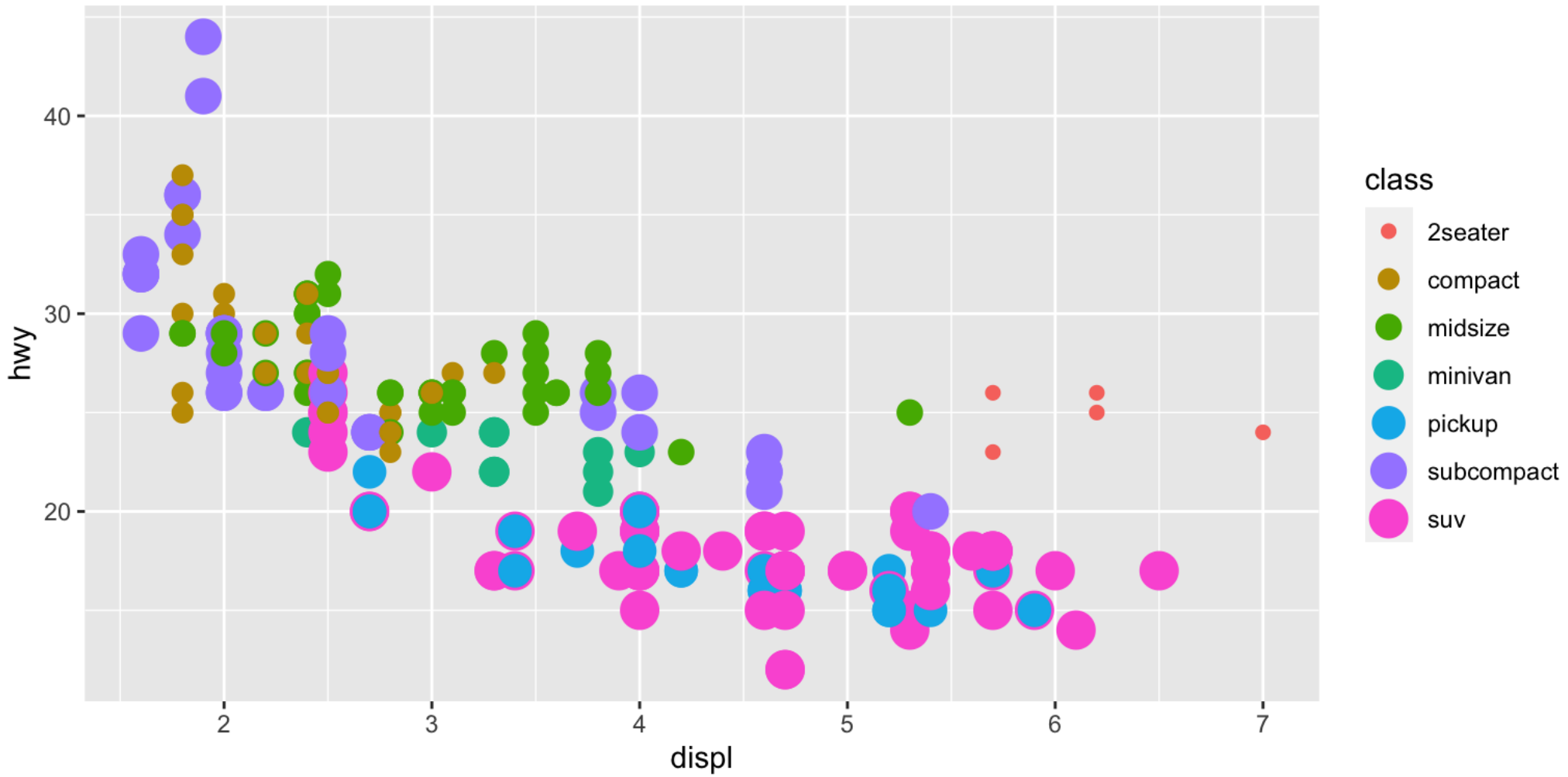
Bigger marker size

```
1  ggplot(data = mpg) +
2    geom_point(mapping = aes(x = displ, y = hwy, color = class),
3               size = 3)
```



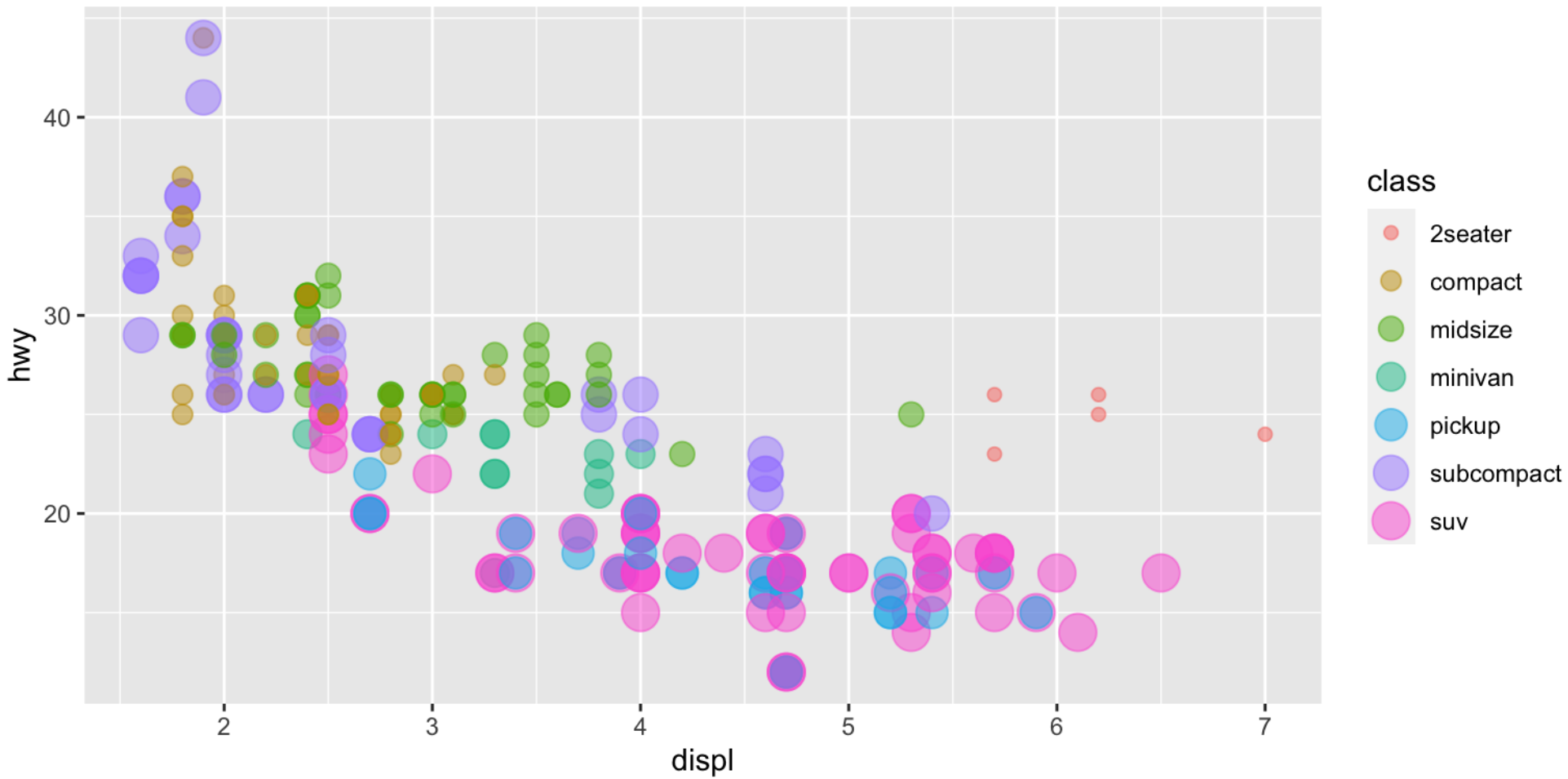
Size-code by another variable

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



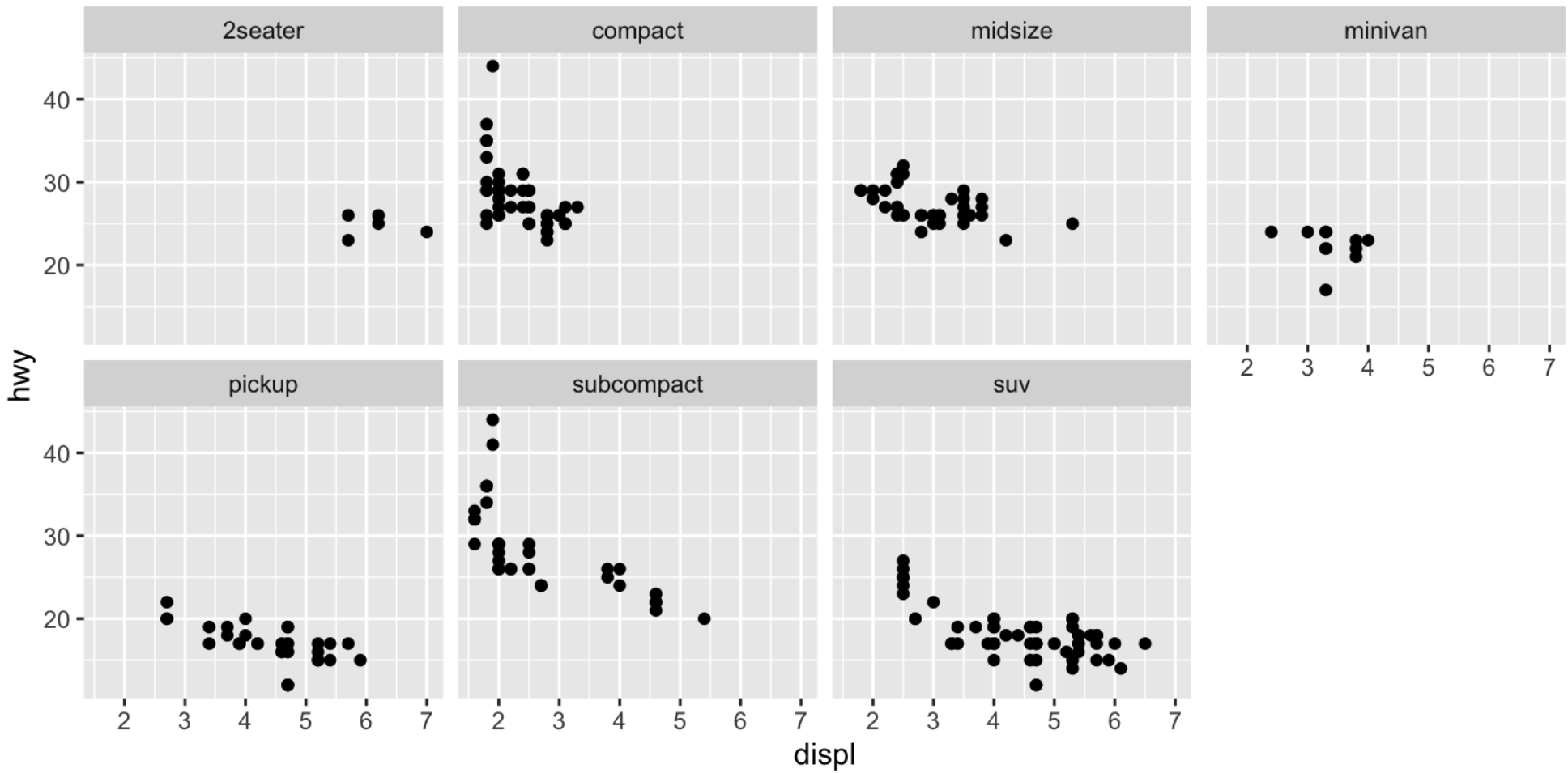
Alpha transparency

```
1  ggplot(data = mpg) +
2    geom_point(mapping = aes(x = displ, y = hwy, color = class, size = class),
3               alpha = 0.5)
```



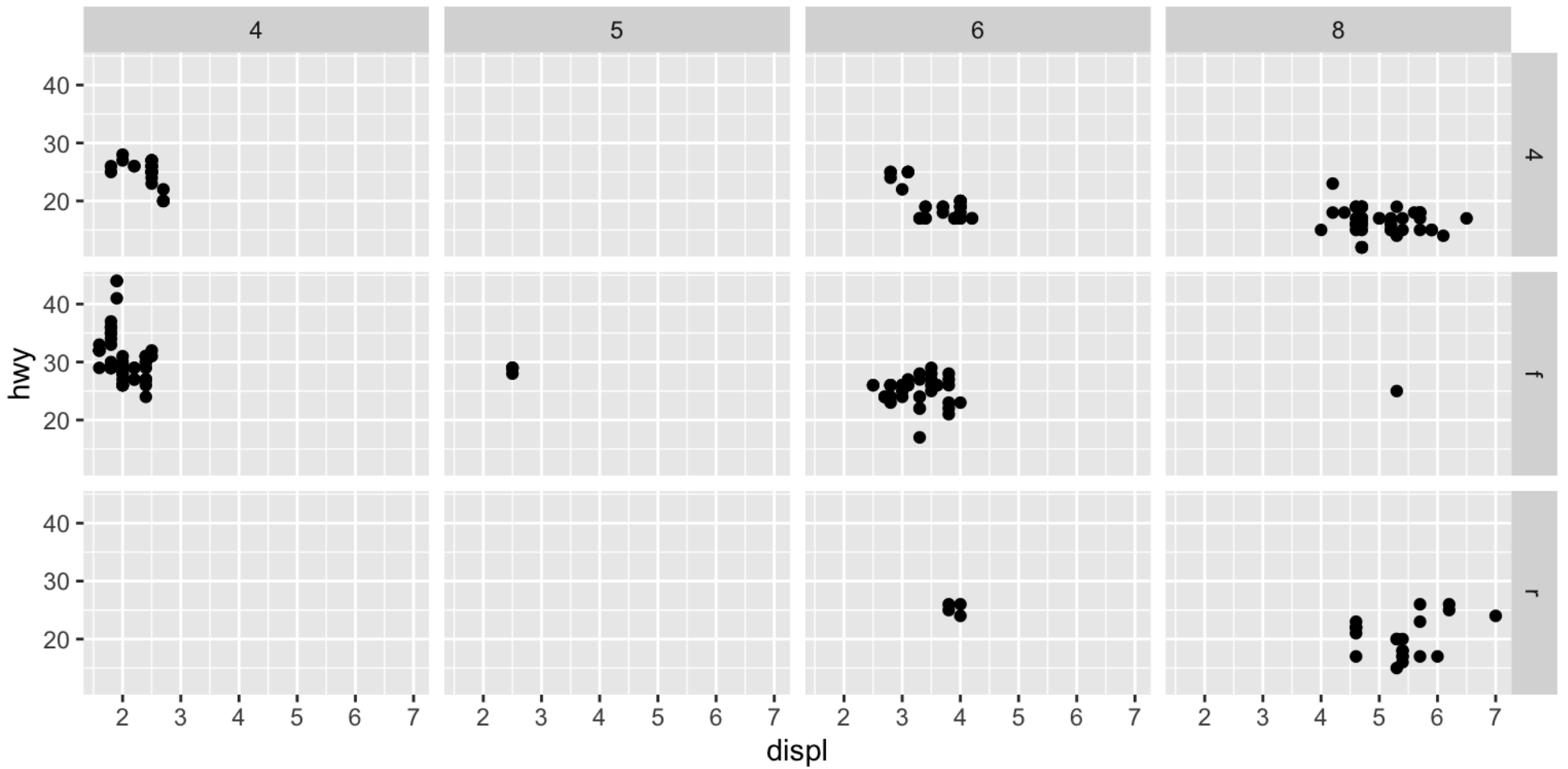
Facets—wrap

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



Facets—grid

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



Facets: wrap vs grid

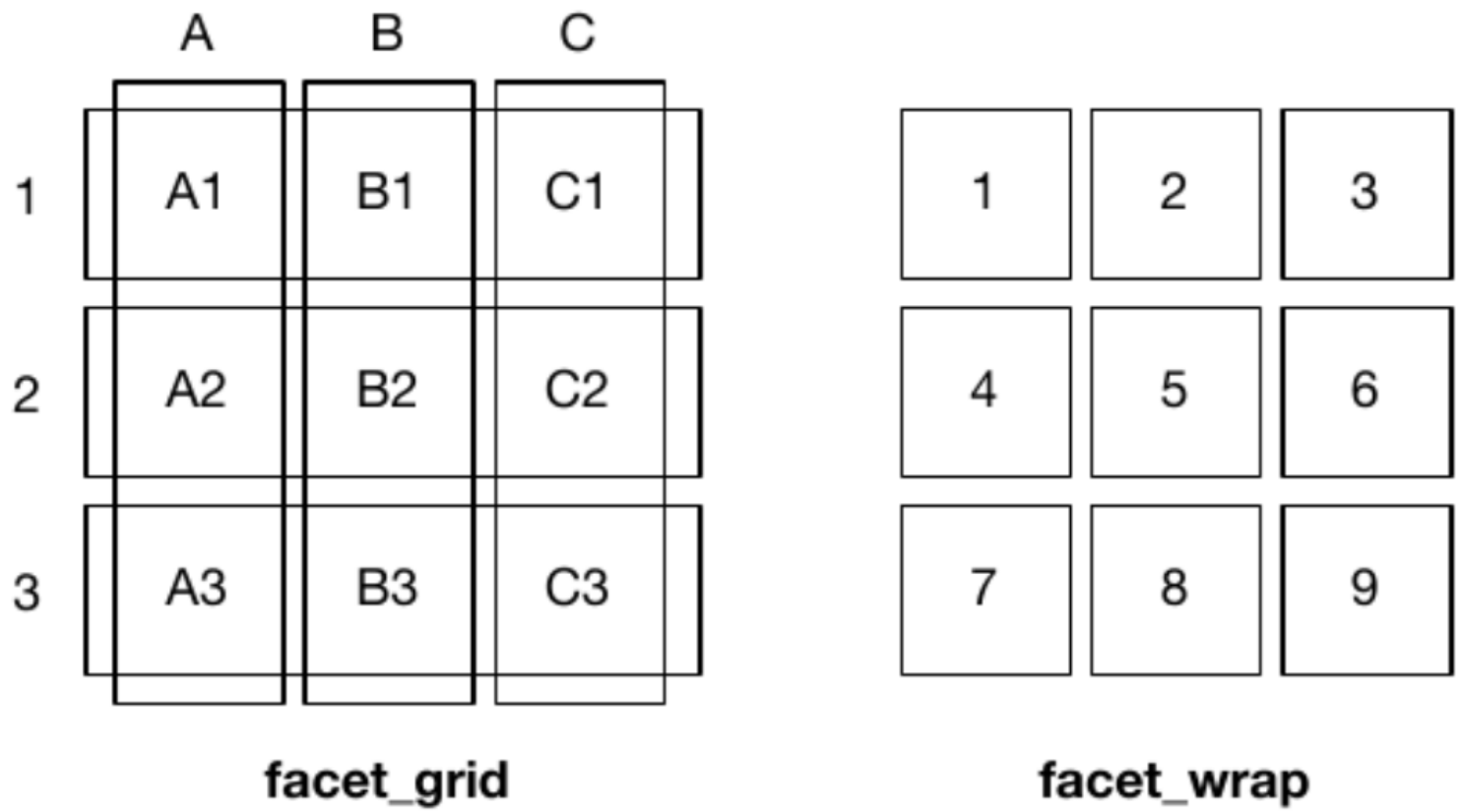
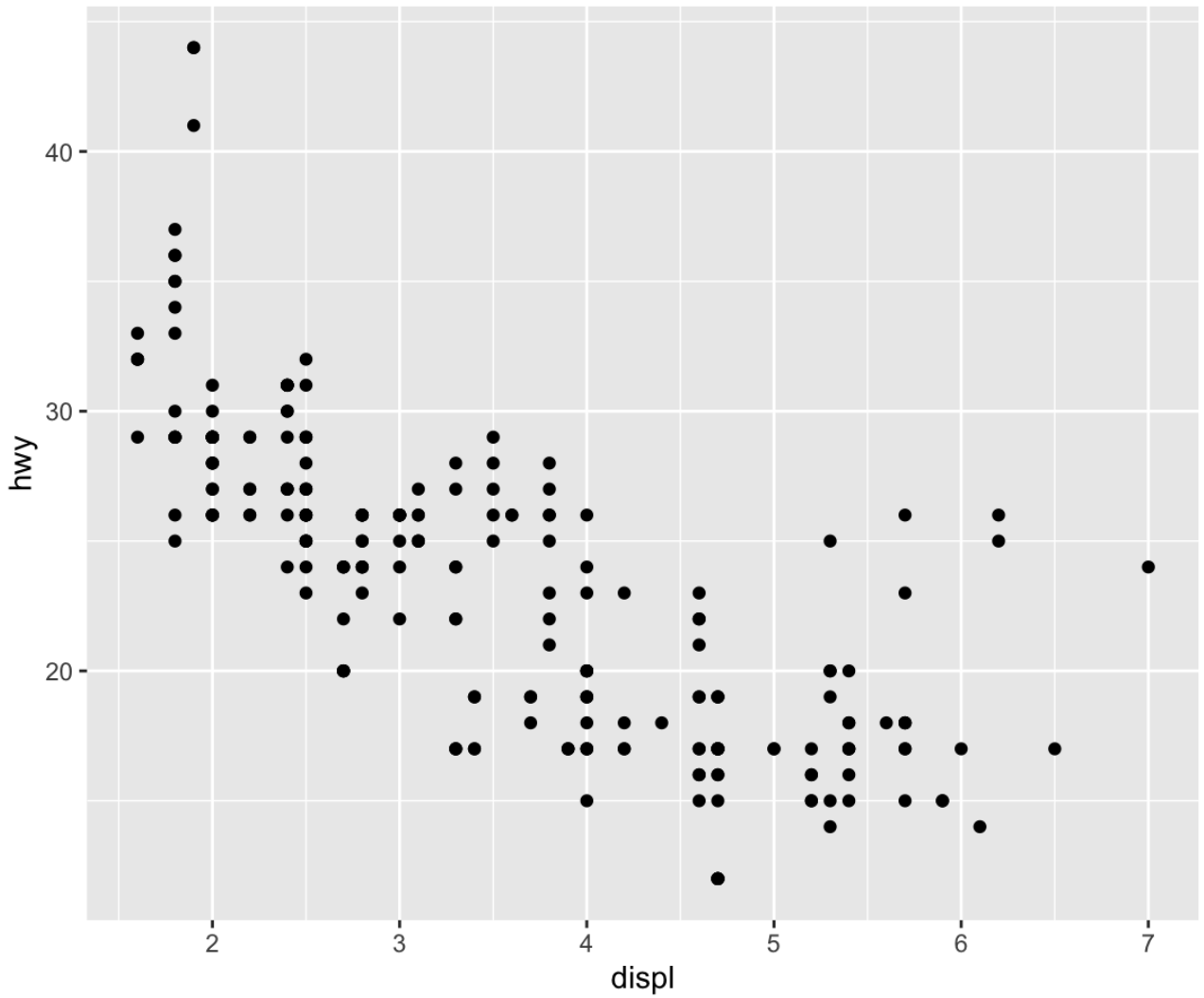


Figure 17.1: A sketch illustrating the difference between the two faceting systems. `facet_grid()` (left) is fundamentally 2d, being made up of two independent components. `facet_wrap()` (right) is 1d, but wrapped into 2d to save space.

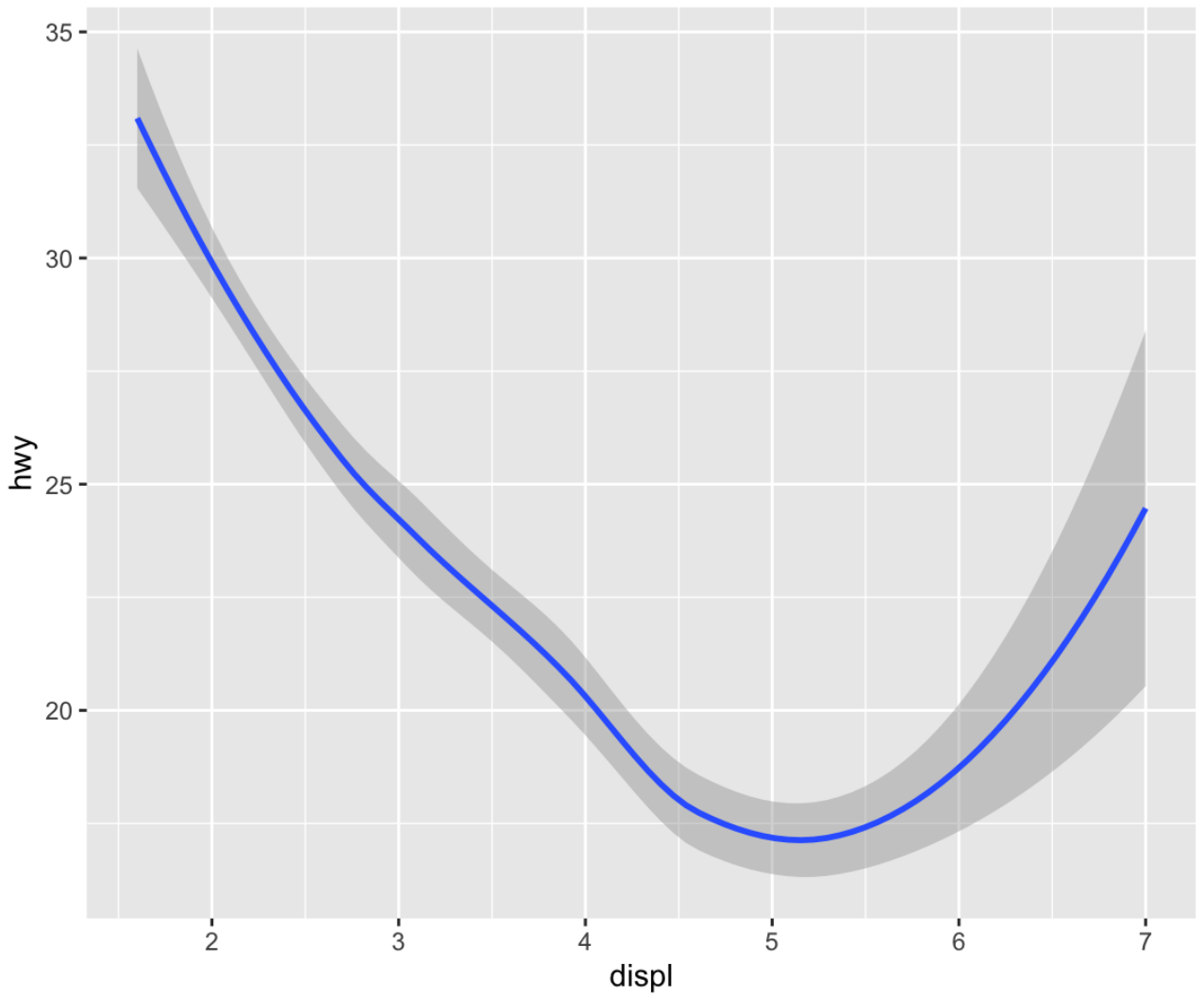
geoms—geom_point()

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



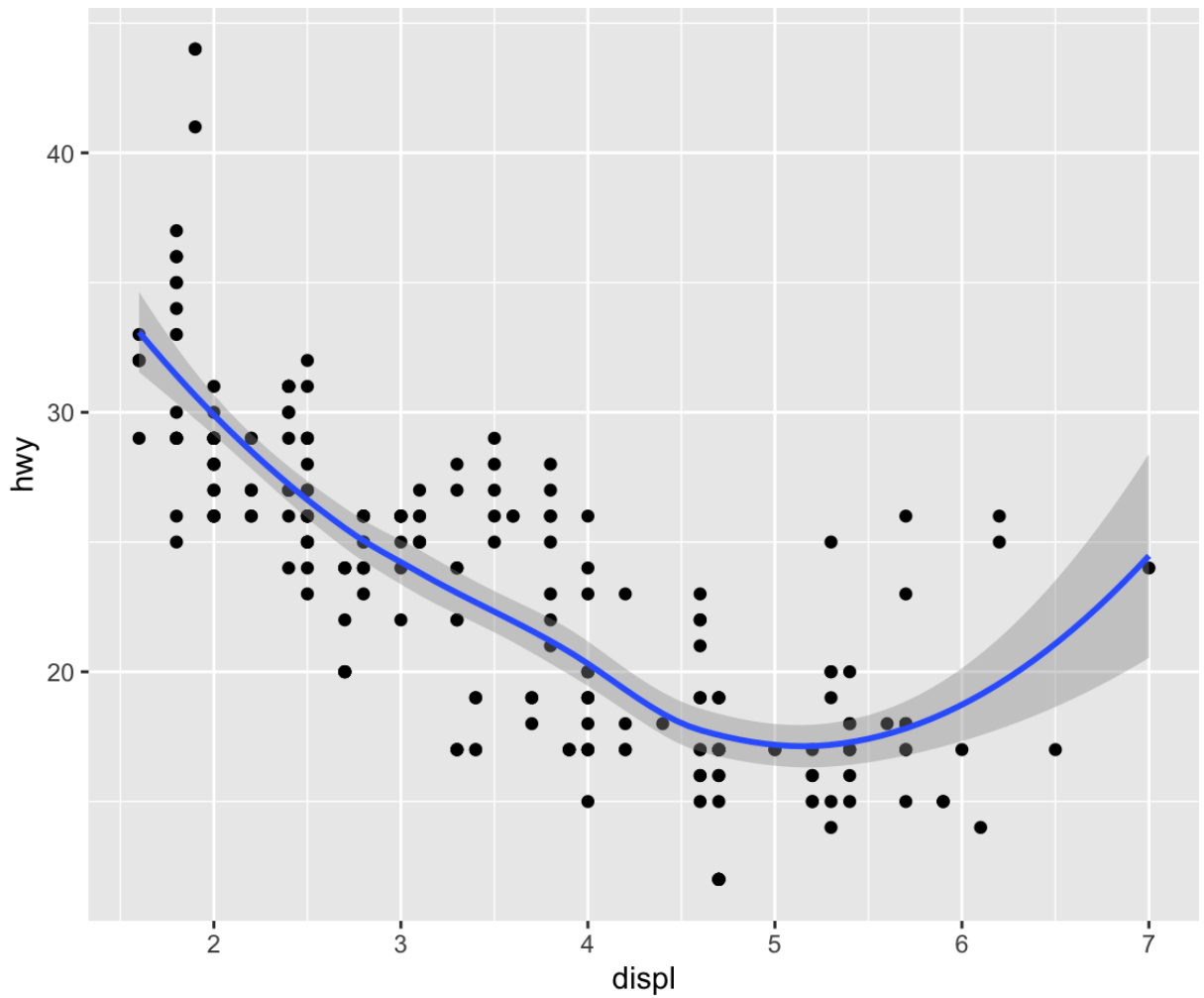
geoms—geom_smooth()

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



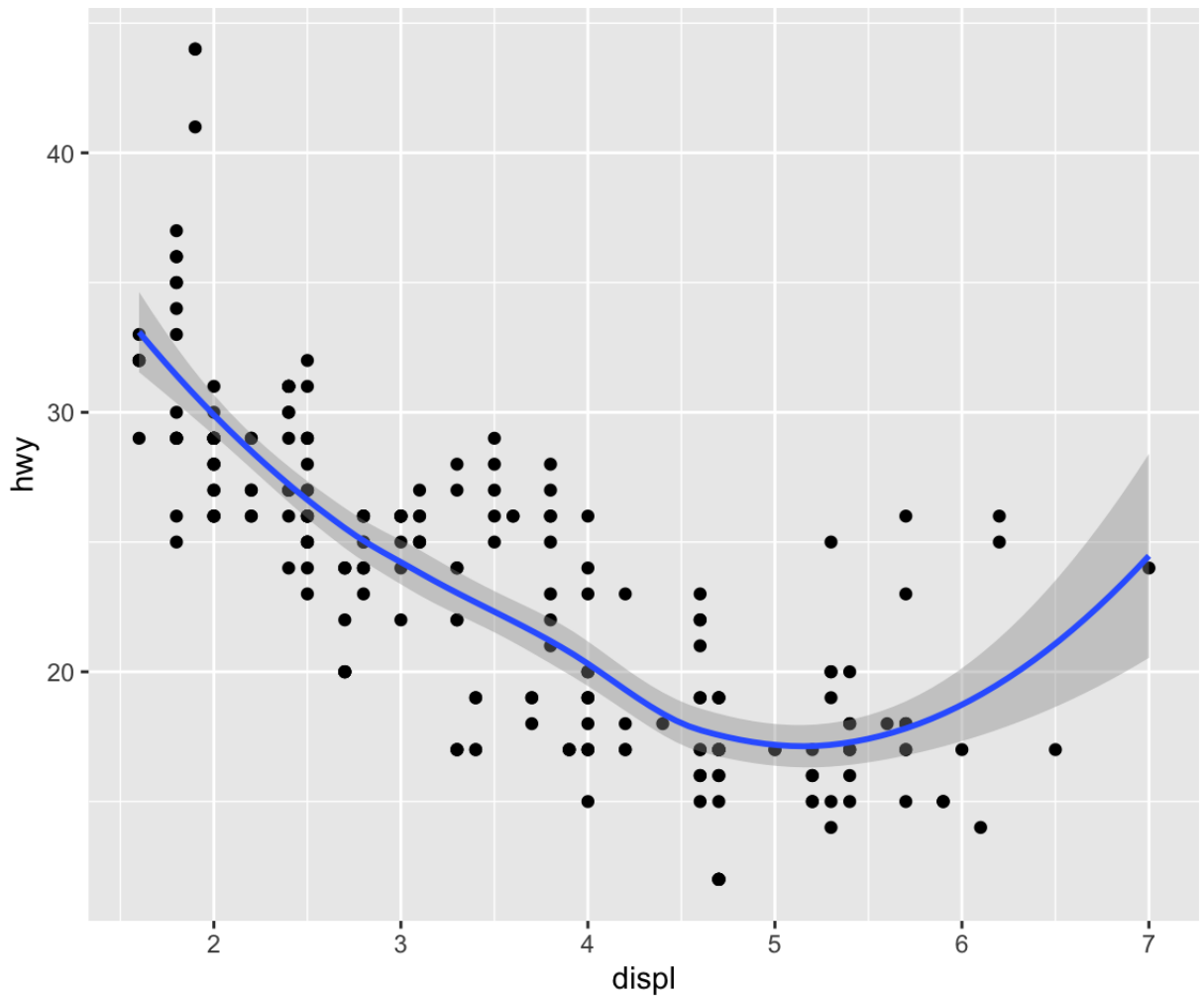
geoms–both

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



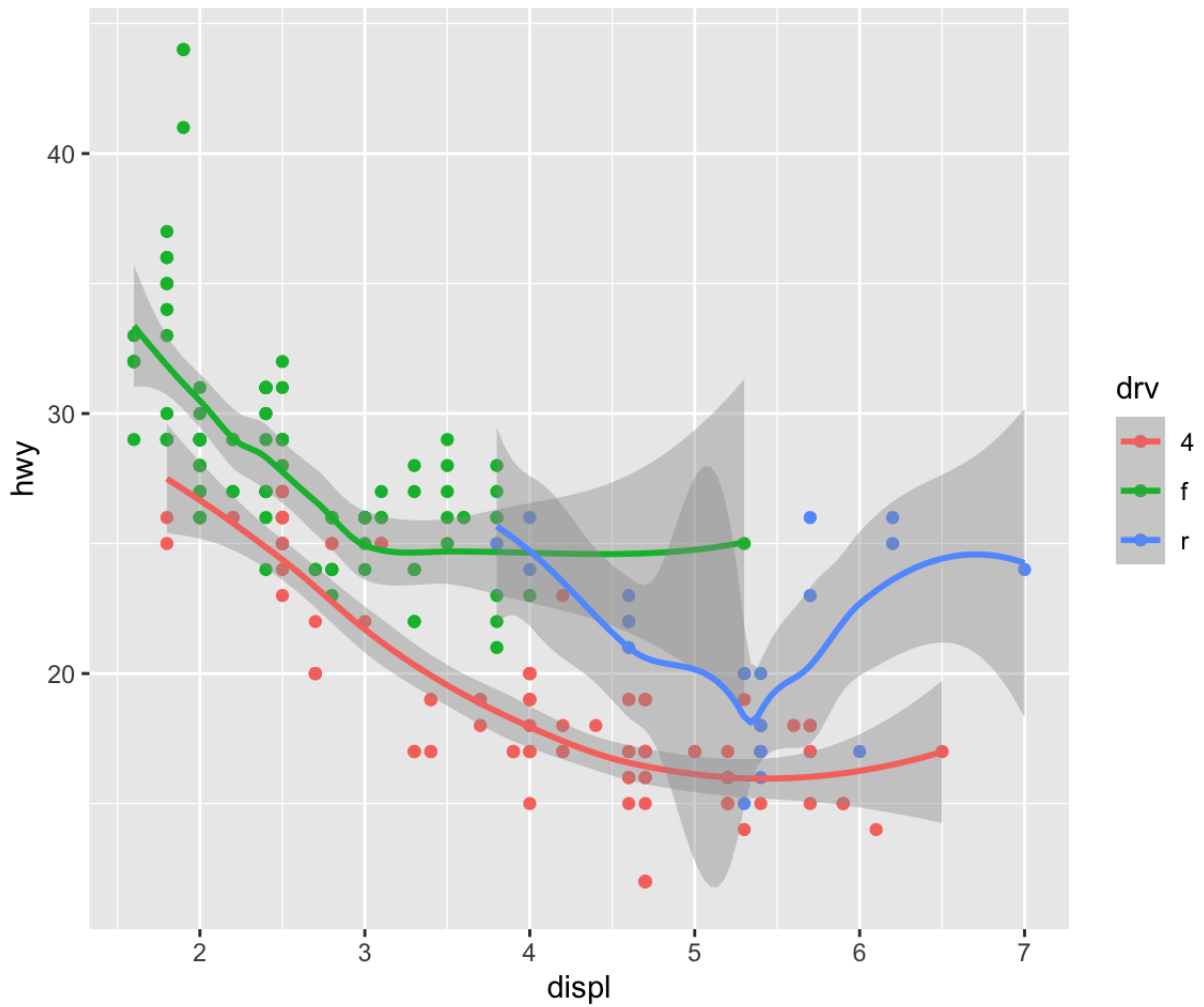
Pair mappings to `ggplot()`

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



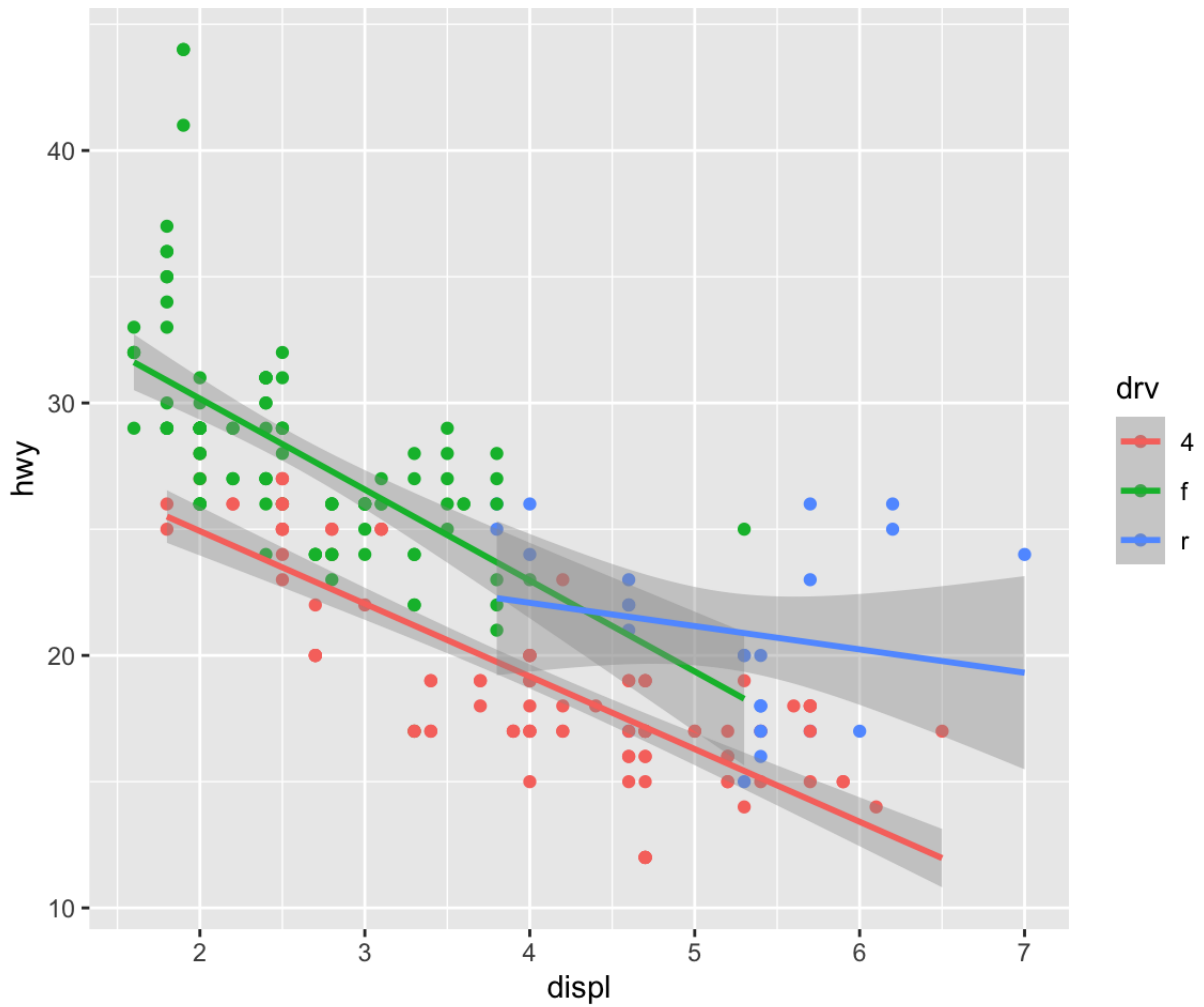
geoms—colors

```
1 ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color=drv)) +
2     geom_point() +
3     geom_smooth()
```



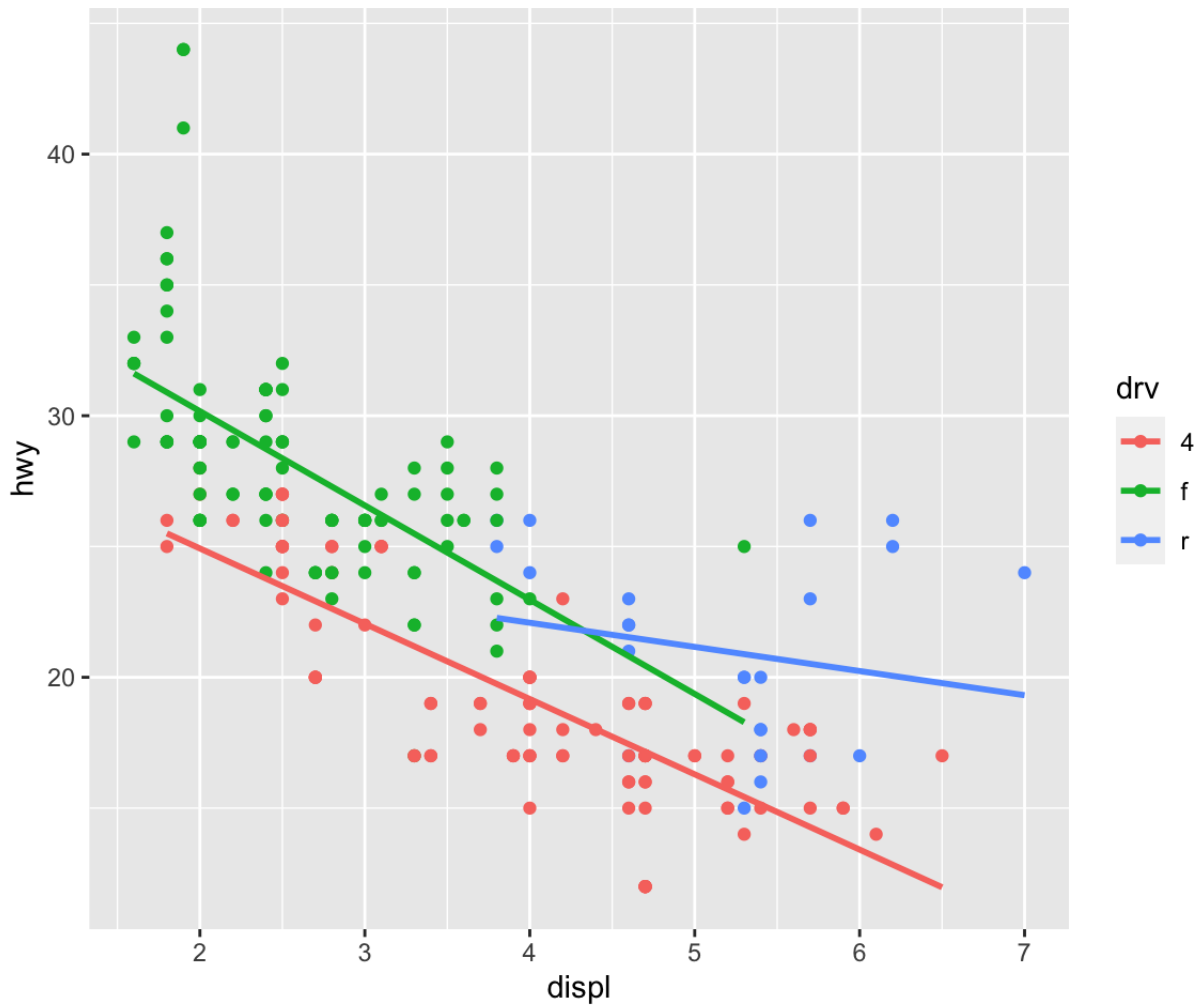
`geom_smooth()` options

```
1 ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color=drv)) +
2     geom_point() +
3     geom_smooth(method="lm")
```



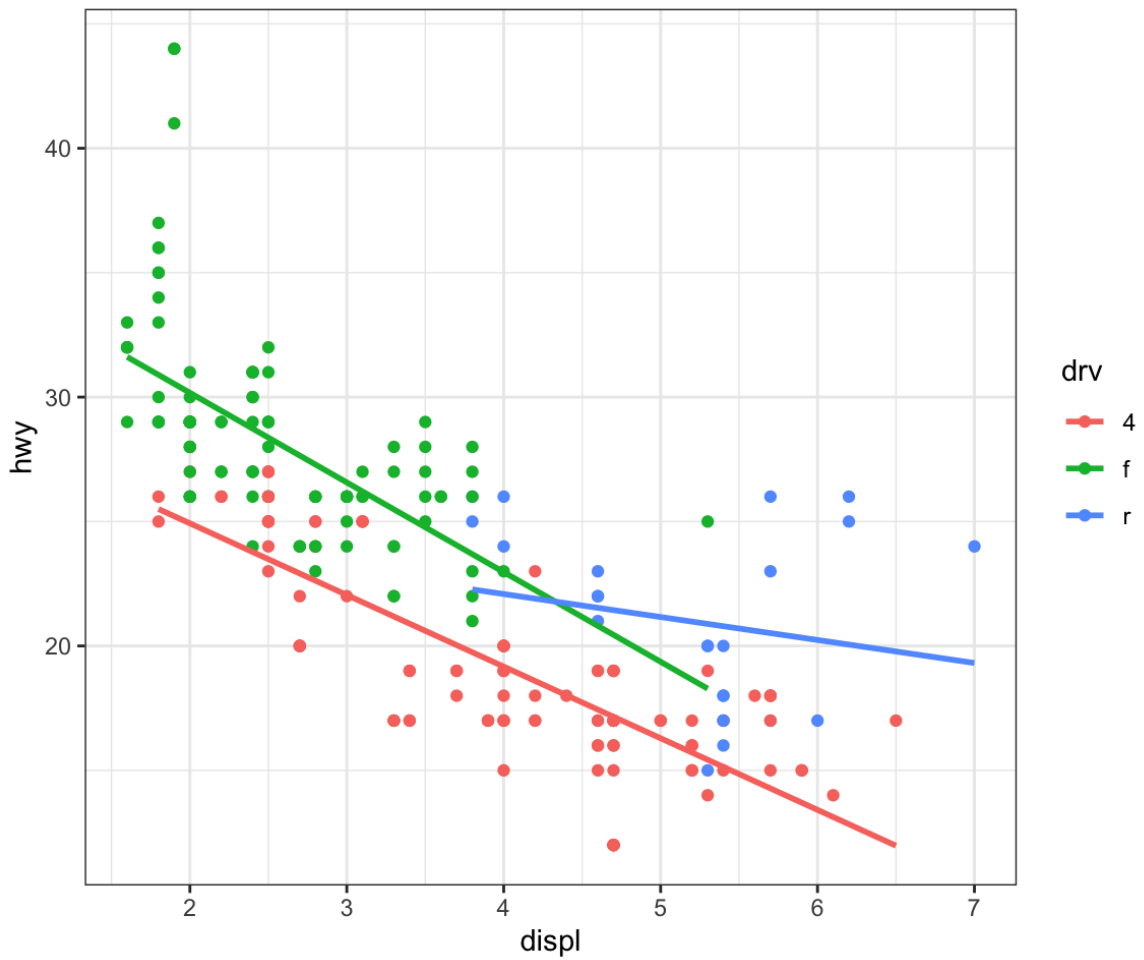
`geom_smooth()` options

```
1 ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color=drv)) +
2     geom_point() +
3     geom_smooth(method="lm", se=FALSE)
```



`theme` options

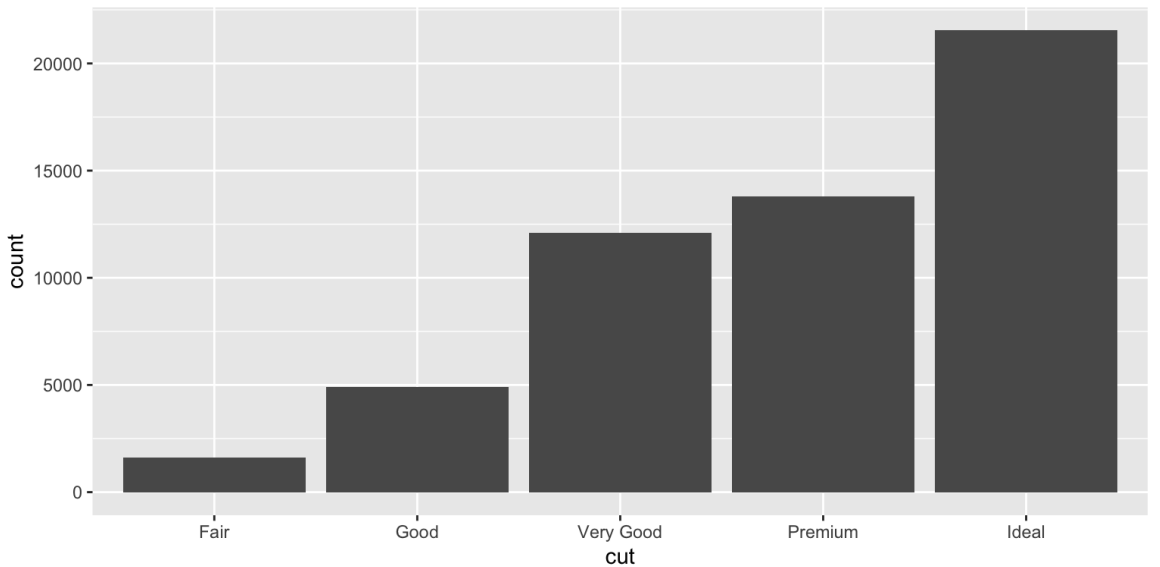
```
1 ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color=drv)) +
2     geom_point() +
3     geom_smooth(method="lm", se=FALSE) +
4     theme_bw()
```



Statistical Transformations

- some graphs (e.g. scatterplots) plot raw values in your dataset
- others (bar charts, histograms, boxplots, smoothers) **calculate new values**
  - called a **stat** for short

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



Statistical Transformations

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

1. **`geom_bar()`** begins with the **`diamonds`** data set

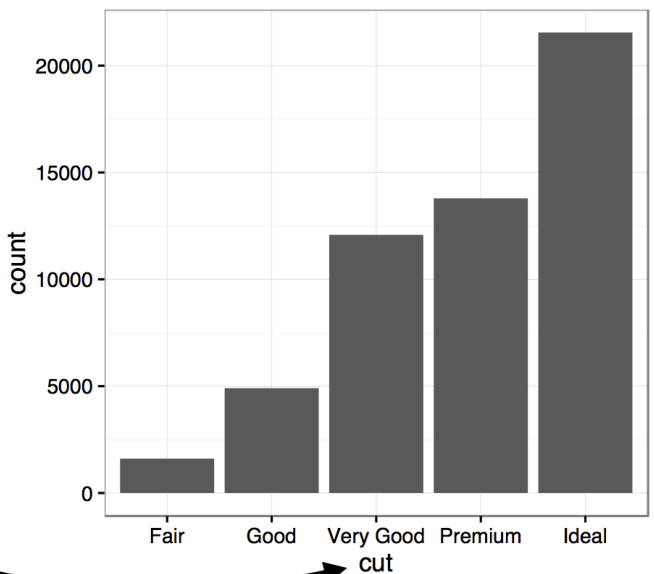
2. **`geom_bar()`** transforms the data with the "count" stat, which returns a data set of cut values and counts.

3. **`geom_bar()`** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.

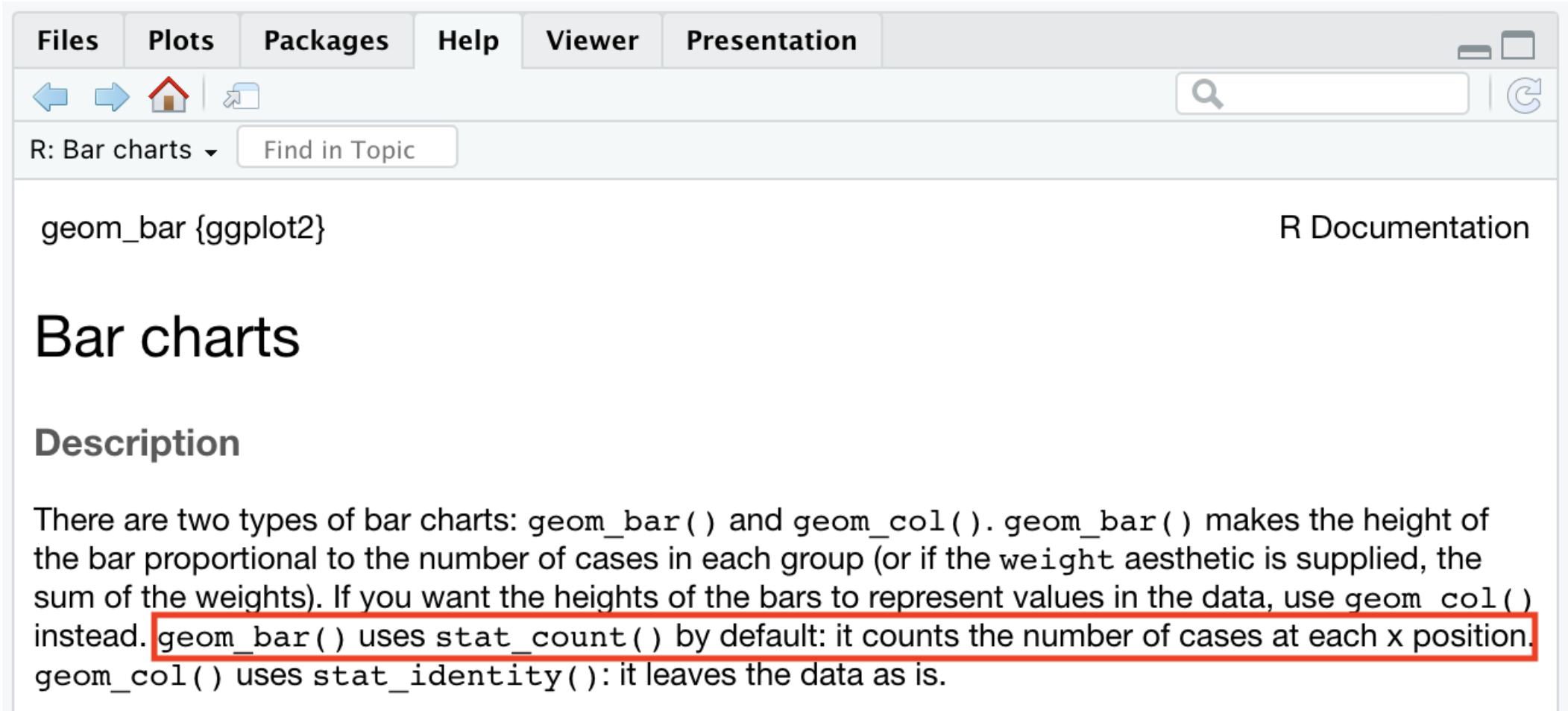
carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1



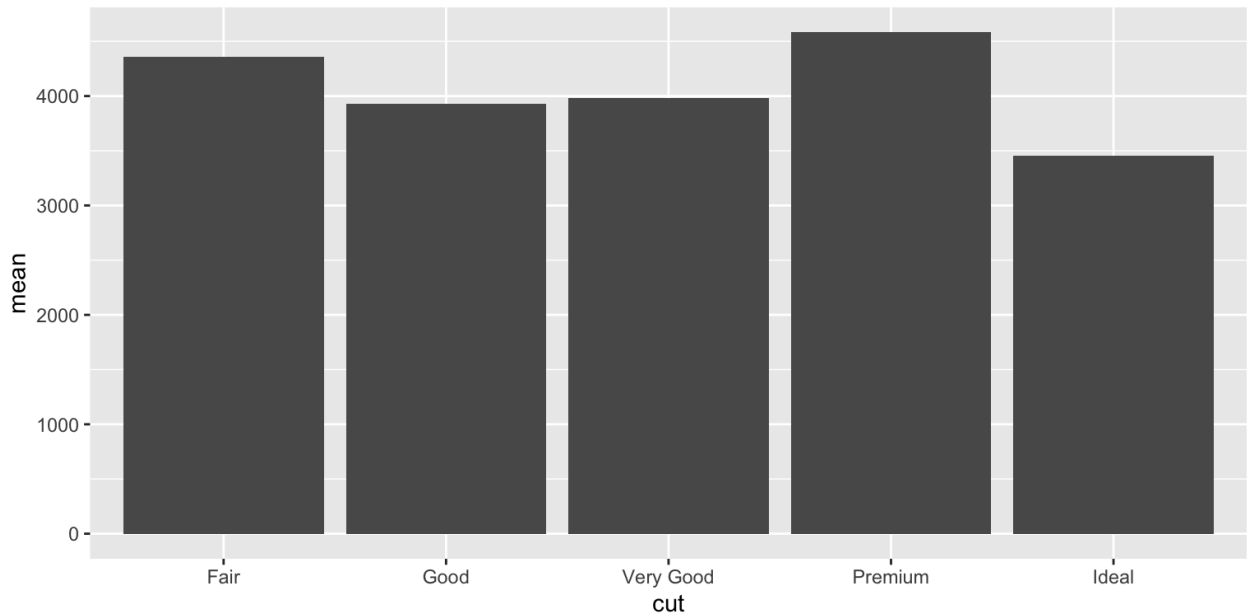
Statistical Transformations



Barplot with raw values not stat

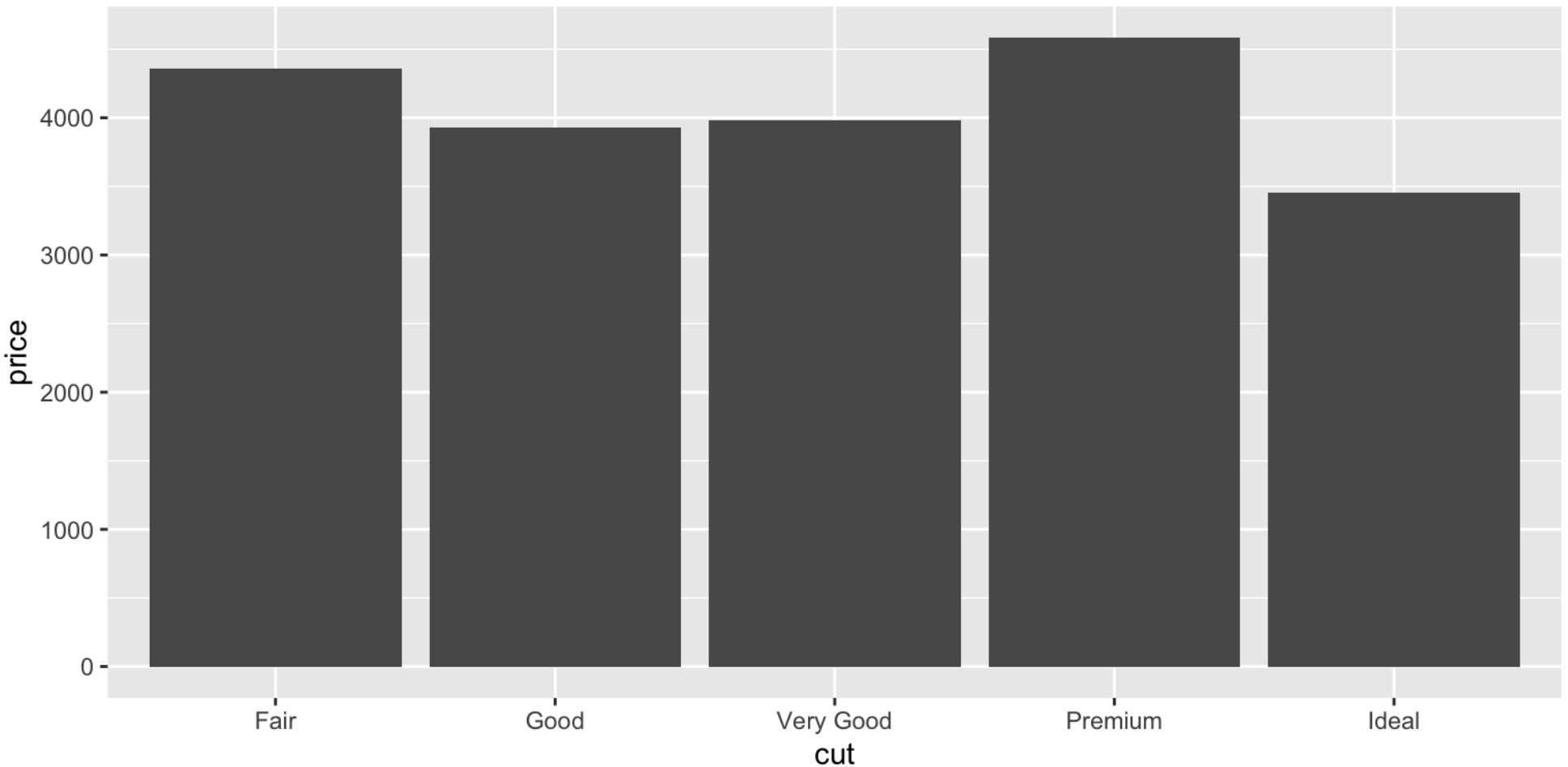
```
# A tibble: 5 × 3
  cut            n  mean
  <ord>      <int> <dbl>
1 Fair        1610 4359.
2 Good        4906 3929.
3 Very Good  12082 3982.
4 Premium    13791 4584.
5 Ideal      21551 3458.
```

```
1 ggplot(data = meanprice) +
2   geom_bar(mapping = aes(x = cut, y = mean),
3            stat = "identity")
```



Barplot with summary (e.g. mean)

```
ggplot(diamonds) +
  stat_summary_bin(aes(x = cut, y = price),
                   fun = "mean", geom = "bar")
```



So many options!

- do the readings
- play with the code
- homeworks will teach you a little bit more
- build on simple examples

The layered grammar of graphics

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION>
```

- you can uniquely describe any plot as a combination of a **dataset**, a **geom**, a set of **mappings**, a **stat**, a **position** adjustment, a **coordinate** system, and a **faceting** scheme

The layered grammar of graphics

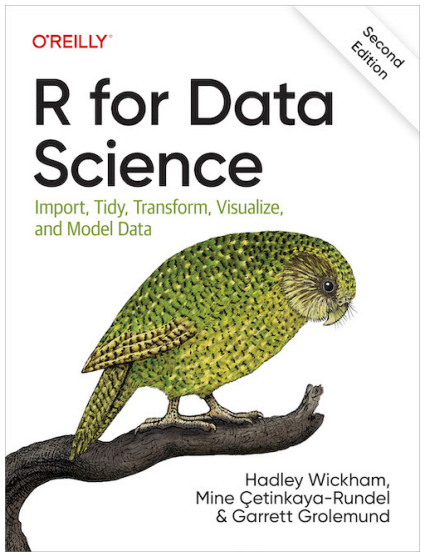
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION>
```

- with these 7 parameters you can make **any** plot

- you rarely need to supply all seven parameters to make a graph
- `ggplot2` will provide useful defaults for everything except the data, the mappings, and the geom function

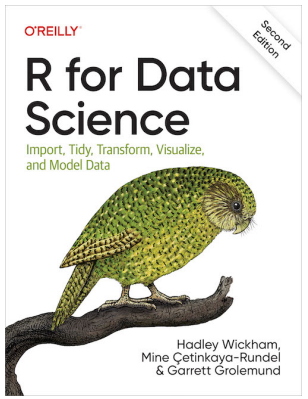
R for Data Science

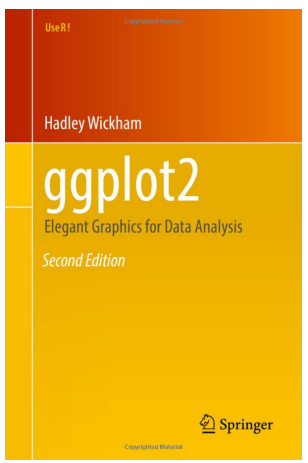
- work through Chapter 3 Data visualisation
- read Chapter 4 Workflow: basics
- next week:
  - Chapter 5 Data transformation
  - Chapter 6 Workflow: scripts
  - Chapter 7 Exploratory Data Analysis



`ggplot`—play around!

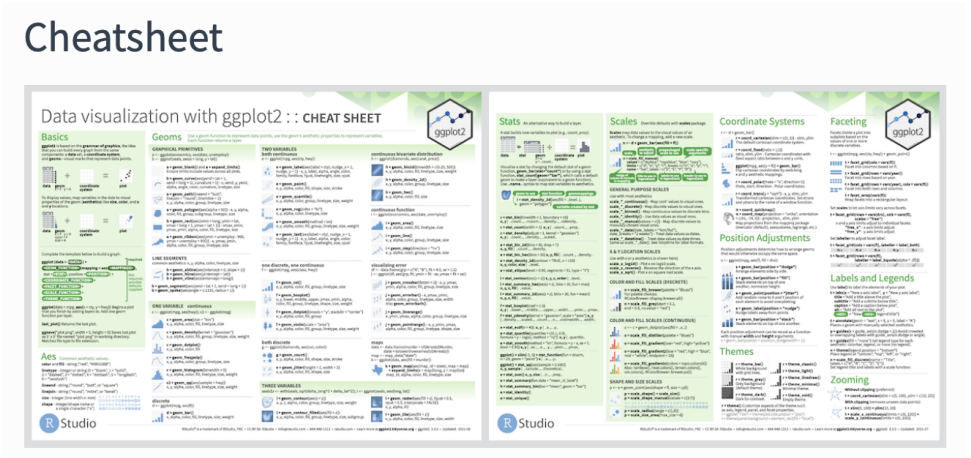
- `ggplot` is simple to get started with
- is as complex as you want it to be
- designed as a “grammar” of graphics—systematic
- many ways of doing the same thing



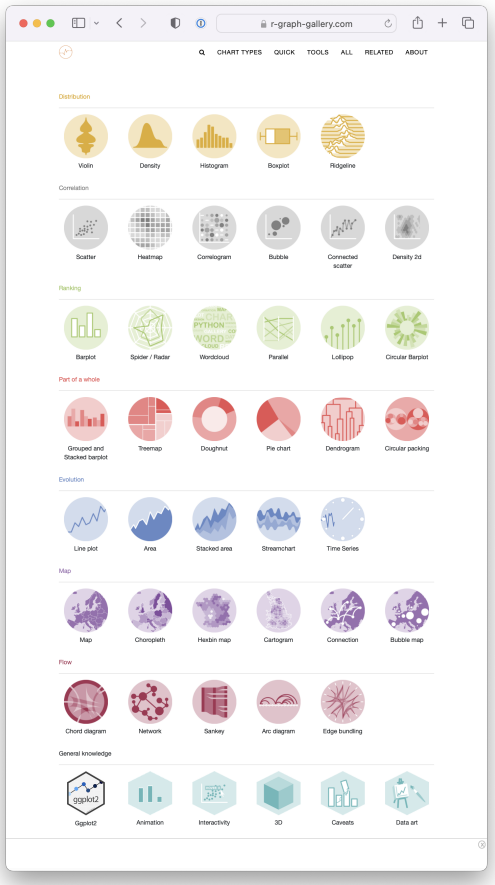


`ggplot`—play around!

ggplot2 cheatsheet



R Graph Gallery



`ggplot`—Homework 1

- some answers are directly in the lecture slides
- some answers are directly in the readings
- some answers require you to apply what you have learned
- TAs are there to help guide you
  - (but not to literally give you the code)

ALWAYS PLOT YOUR DATA

Datasets with the same statistical properties (Anscombe's quartet)

For all 4 of the datasets:

mean of x is 9
mean of y is 7.5
variance is 11
correlation is 0.816
linear regression: y = 3 + 0.5x