5. Exploratory Data Analysis - ggplot2

Data Science for OR - J. Duggan

Data Exploration

"Data exploration is the art of looking at your data, rapidly generating hypotheses, quickly testing them, then repeating again and again and again." (Wickham and Grolemund 2017).

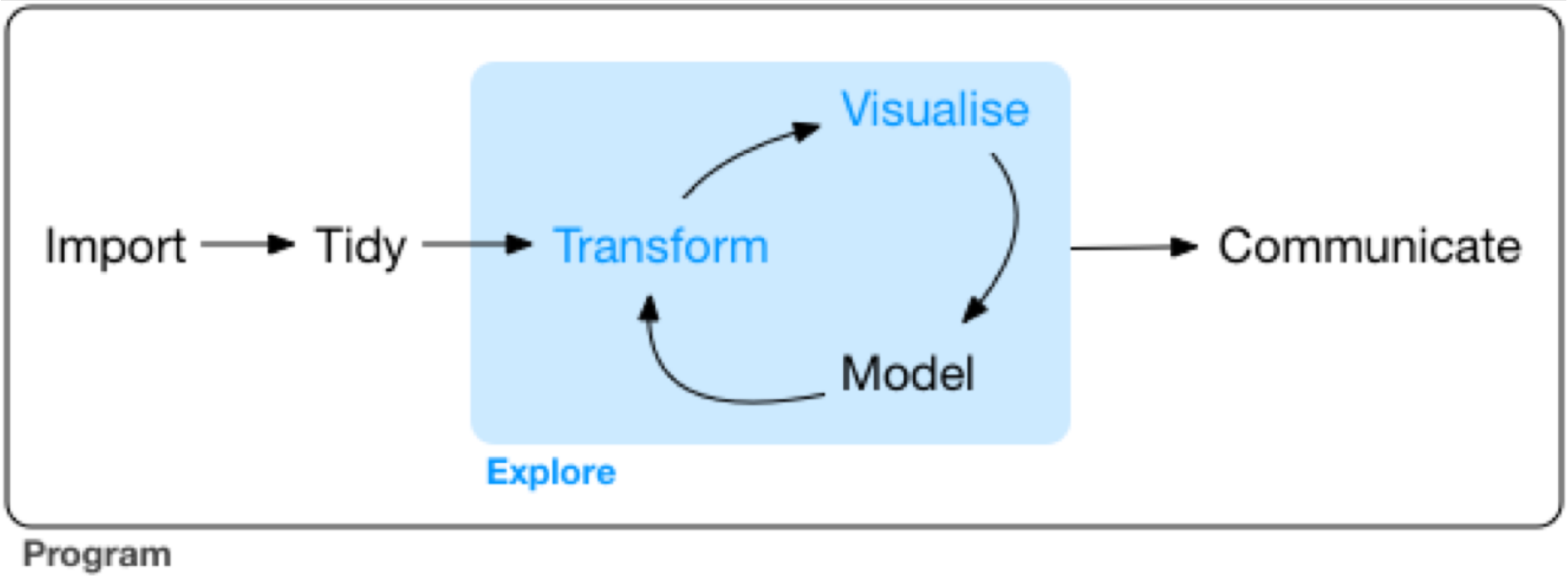


Figure 1: Exploring Data

Data visualisation with ggplot2

"The simple graph has brought more information to the data analyst's mind that any other device." – John Tukey

```
d <- ggplot2::mpg  # get a copy of mpg
glimpse(d)  # show structure and some data
```

```
## Observations: 234
## Variables: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi"
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 20
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6,
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)"
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4"
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28
```

Fuel Economy Data Set (ggplot2::mpg)

This dataset contains a subset of the fuel economy data that the EPA makes available on http://fueleconomy.gov. It contains only models which had a new release every year between 1999 and 2008 - this was used as a proxy for the popularity of the car.

manufacturer	car manufacturer	**drv**	drive type
model	model name	**cty**	city miles per gallon
displ	engine disp (l)	**hwy**	highway miles per gallon
year	year of make	**fl**	fuel type
model	model name	**cty**	city miles per gallon
cyl	number of cylinders	**class**	"type" of car
trans	type of transm.		

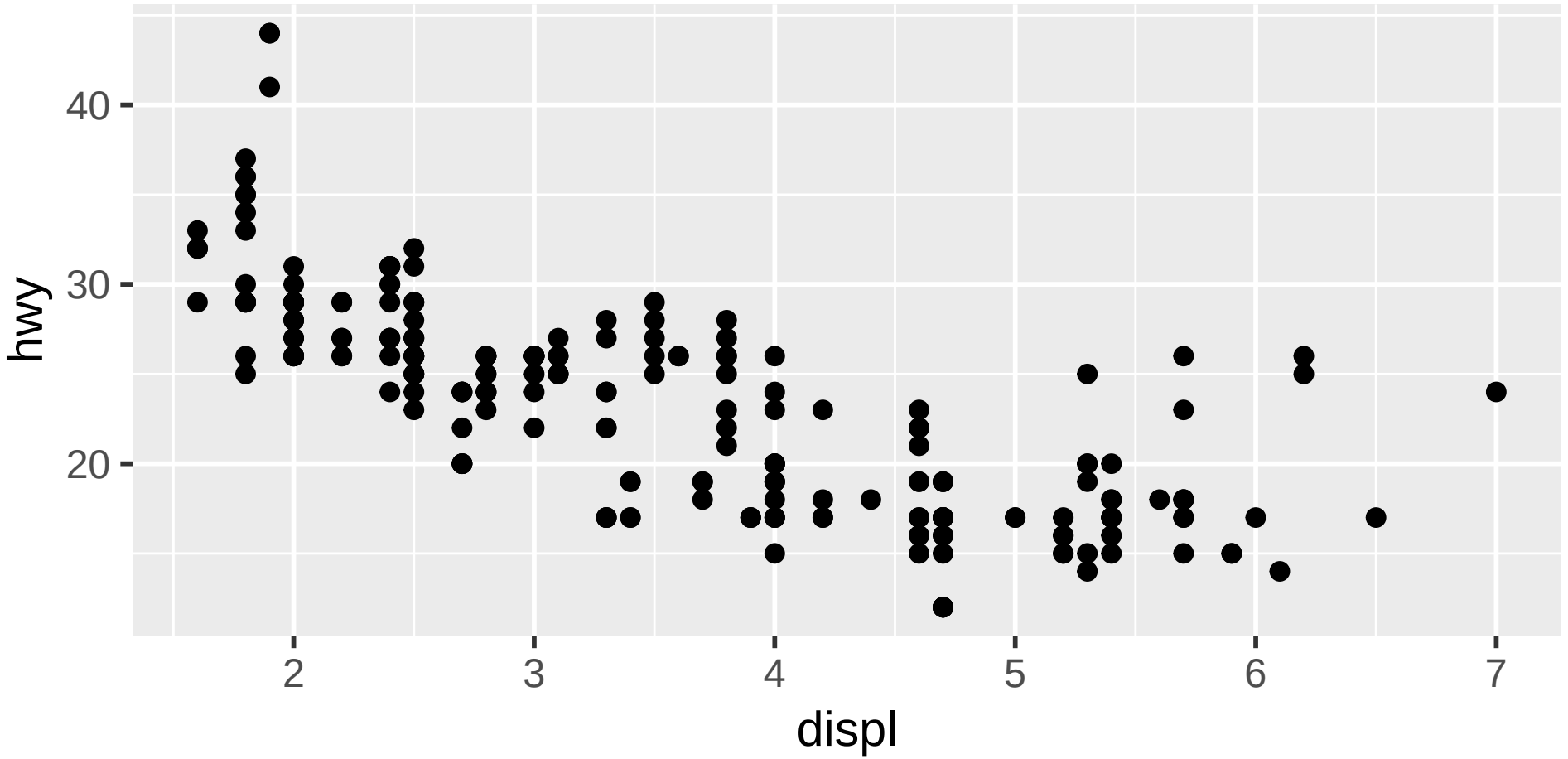
Exploring Data

Generate a first graph to help answer the following question

- Do cars with big engines use more fuel than cars with small engines
- What might the relationship between engine size and fuel efficiency look like?
  - Positive or negative?
  - Linear or non-linear?
- Variable (scatter plot)
  - **displ**, a car engine size in litres (x)
  - **hwy**, a car's fuel efficiency on highway - (y)

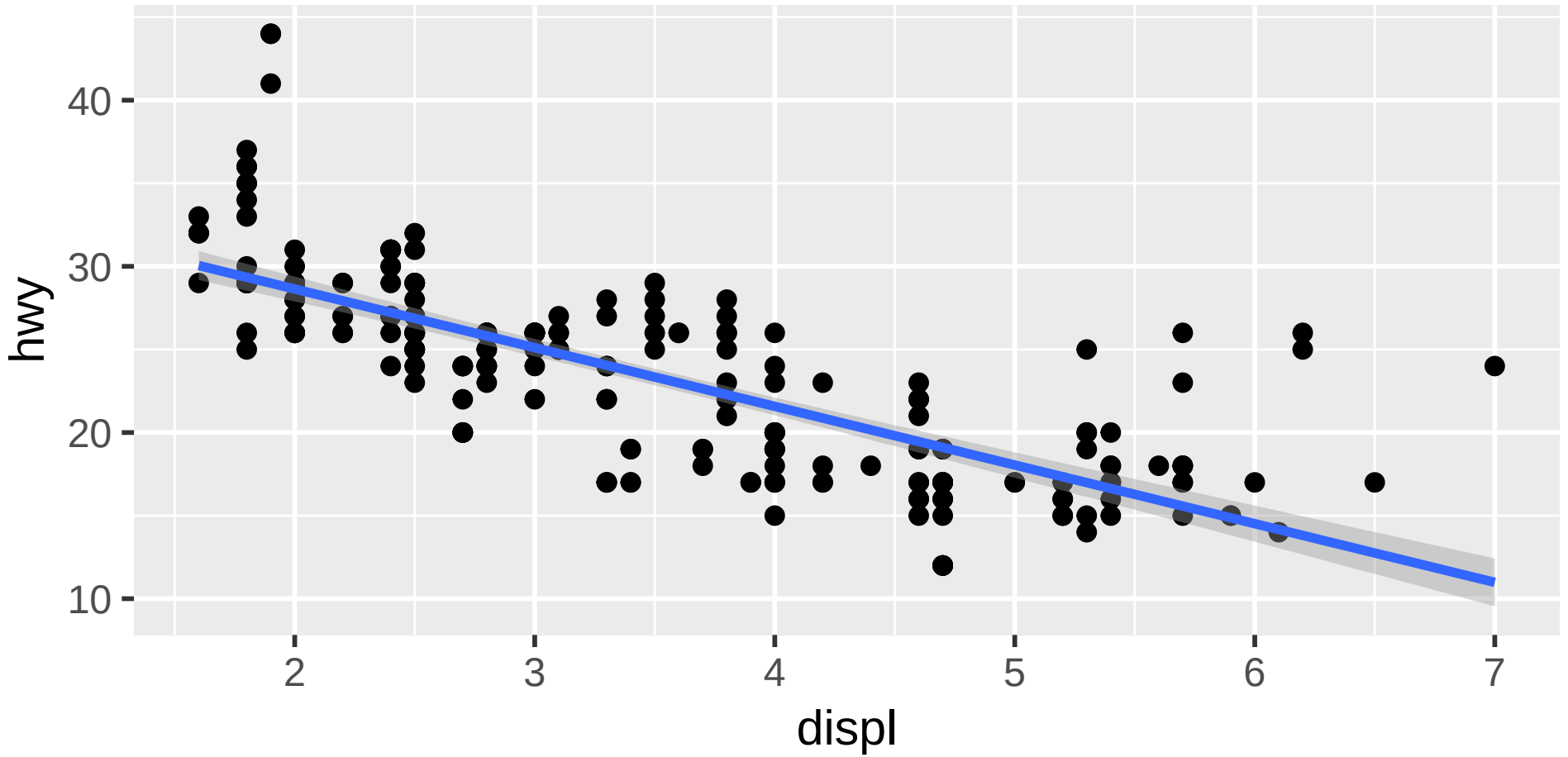
Plotting with ggplot2

```
ggplot(data = d) +    # specify the source tibble
  geom_point(mapping=aes(x=displ,   # map x, y vars
                         y=hwy))
```



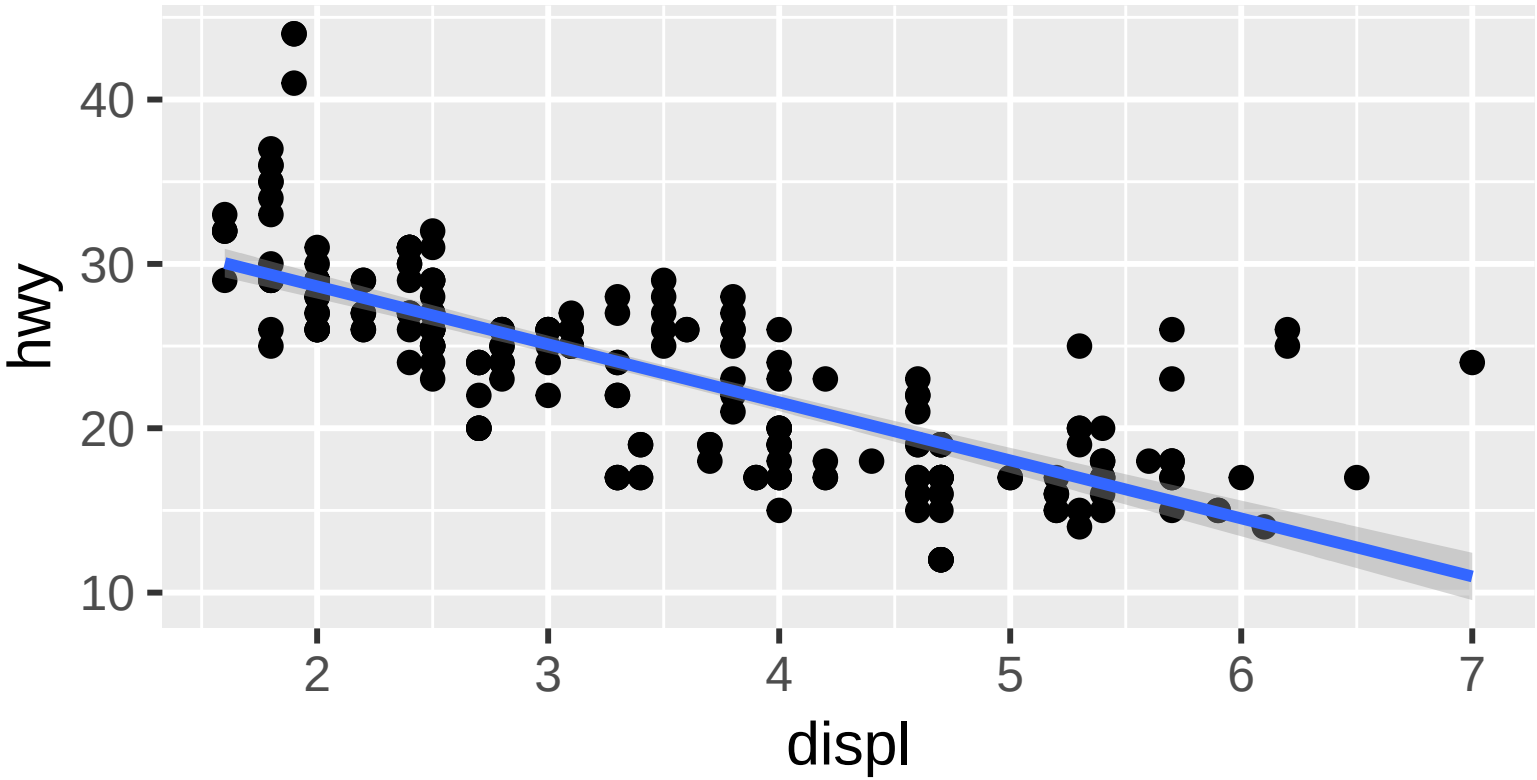
Adding a linear model

```
ggplot(data = d,aes(x=displ, y=hwy)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Interpreting the plot

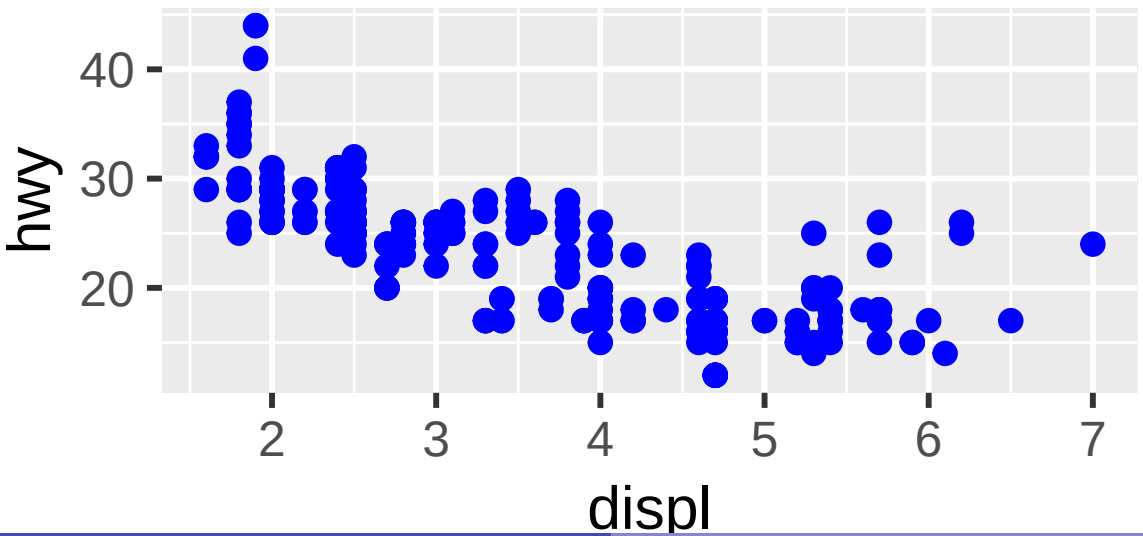
- The plot shows a negative relationship between engine size (displ) and fuel efficiency (hwy)
- Cars with big engines use more fuel
- Does this confirm or refute your hypothesis about fuel efficiency and engine size?



Challenge 2.1

- Explore the hypothesis that city driving is less fuel efficient that highway driving
- Use ggplot to present the points on the same graph, and colour each data set differently
- Does the data confirm or refute your initial hypothesis?

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



Aesthetic Mappings

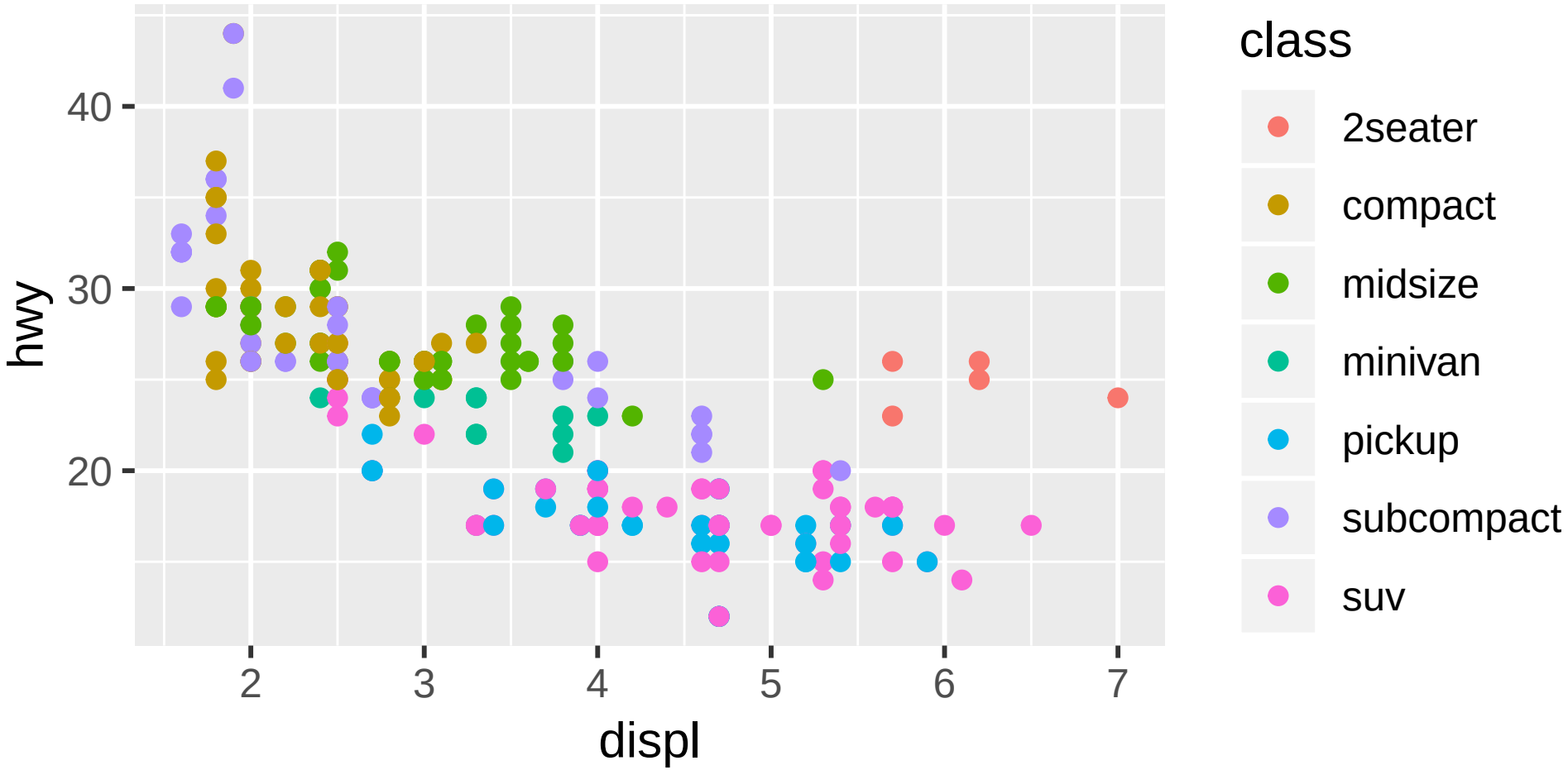
- A third variable can be added to a 2-D plot by mapping it to an aesthetic.
- An aesthetic is a visual property of the plot's objects.
- An aesthetic's level could be colour, size or shape

```
unique(d$class)
```

```
## [1] "compact"    "midsize"    "suv"        "2seater"    "mi
## [6] "pickup"     "subcompact"
```

In ggplot2 - Adding the third variable

```
ggplot(data=d)+
  geom_point(aes(x=displ,y=hwy,colour=class))
```

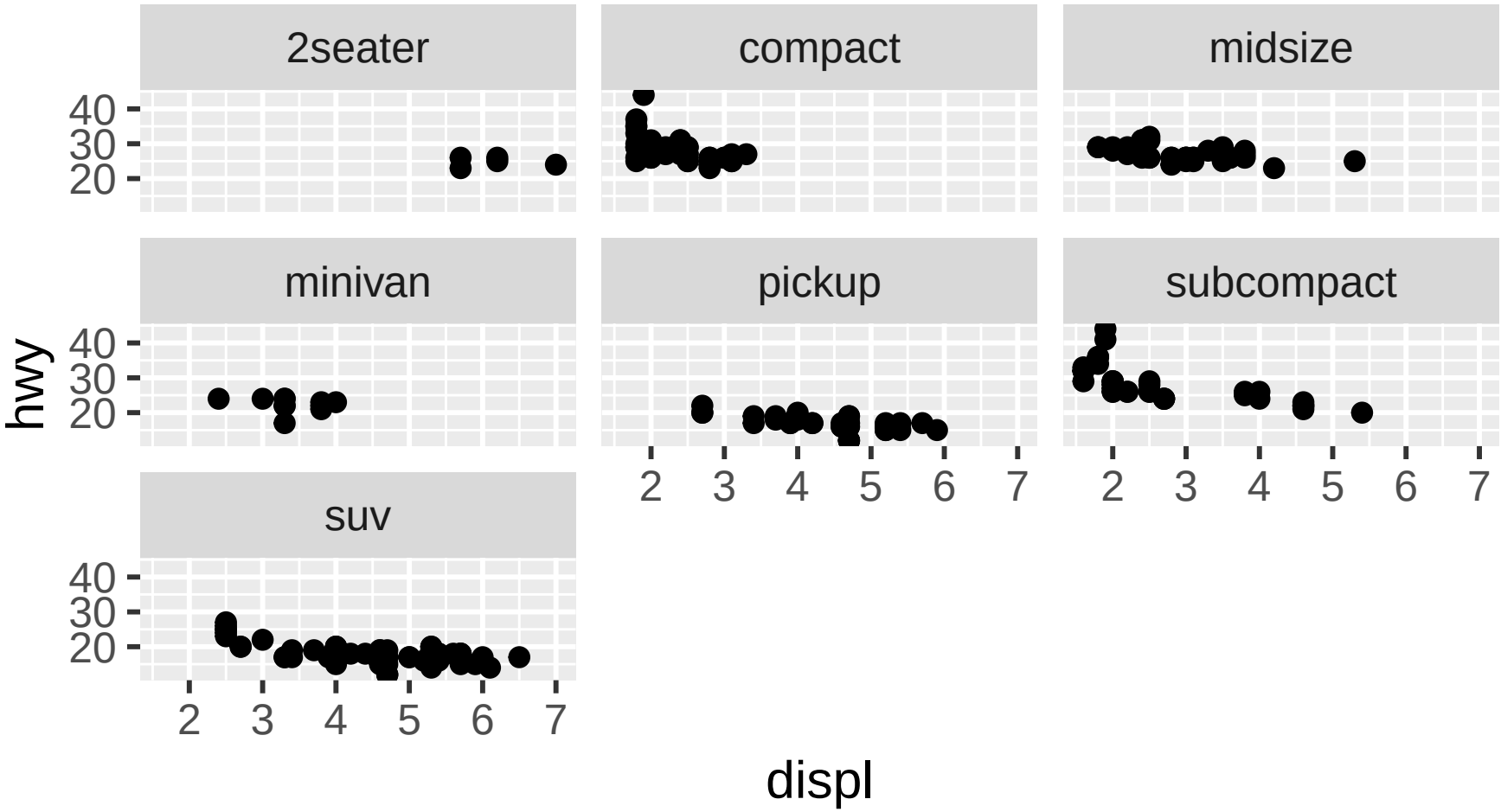


Facets

- Another way to add categorical variables is to split a plot into facets, subplots that display one subset of the data.
- To facet your plot by a single variable, use facet_wrap(), with ~ followed by the variable name
- To facet on the combination of two variables, used facet_grid()

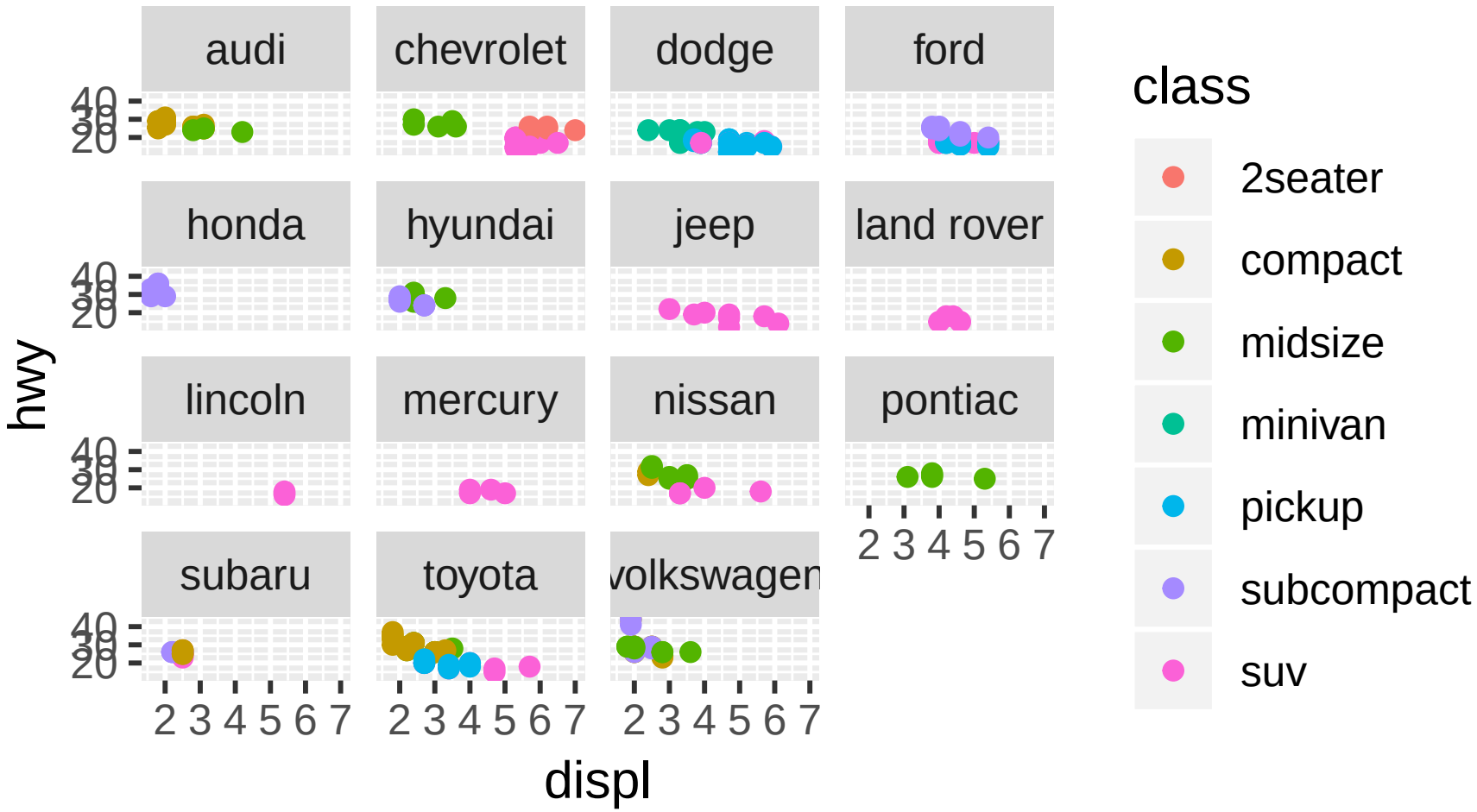
Facet Example 1

```
ggplot(data=d)+
  geom_point(aes(x=displ,y=hwy))+
  facet_wrap(~class)
```



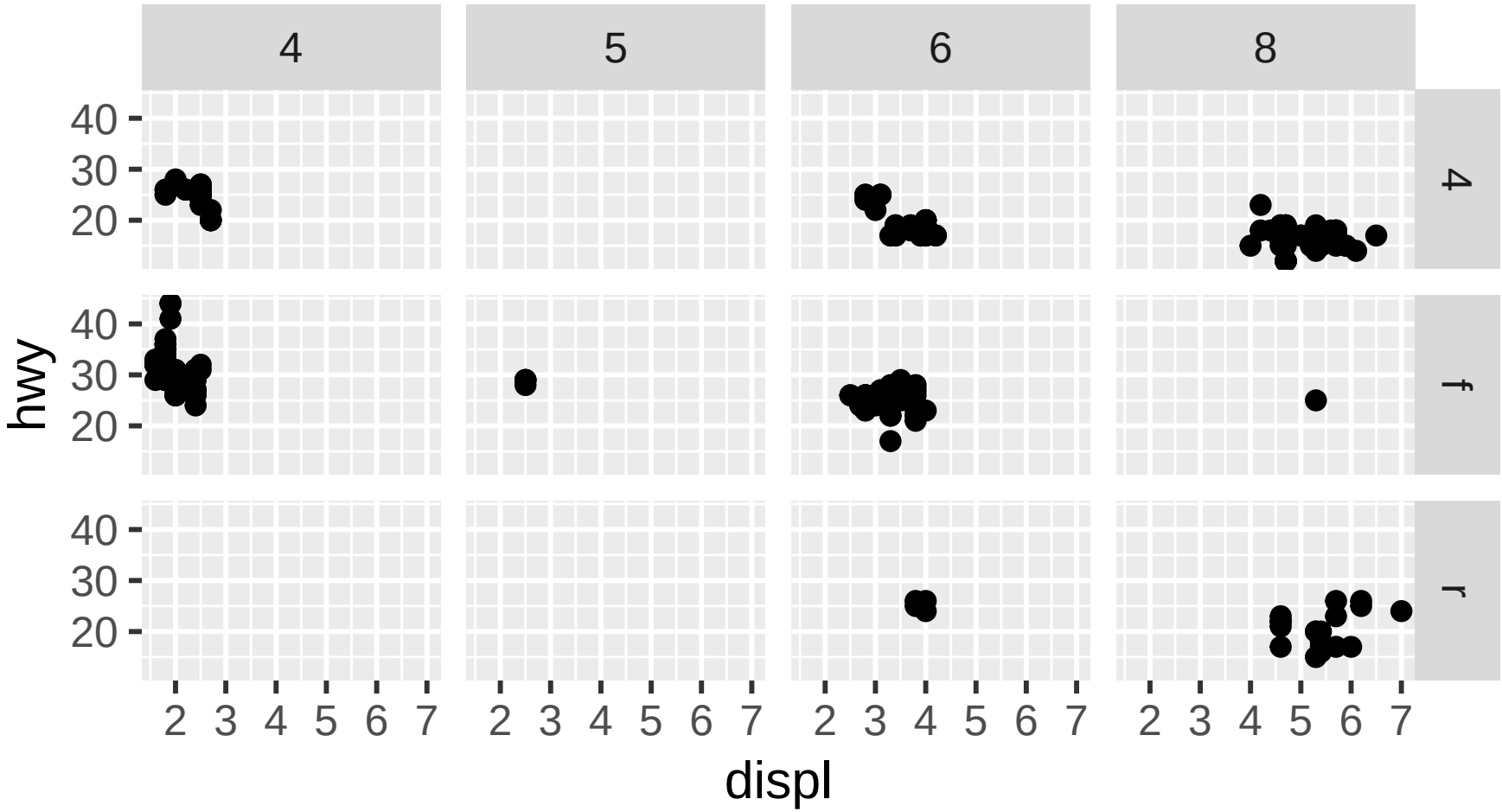
Facet Example 2

```
ggplot(data=d)+
  geom_point(aes(x=displ,y=hwy,colour=class))+
  facet_wrap(~manufacturer)
```



Facet Grid Example

```
ggplot(data=d)+
  geom_point(aes(x=displ,y=hwy))+
  facet_grid(drv~cyl)
```



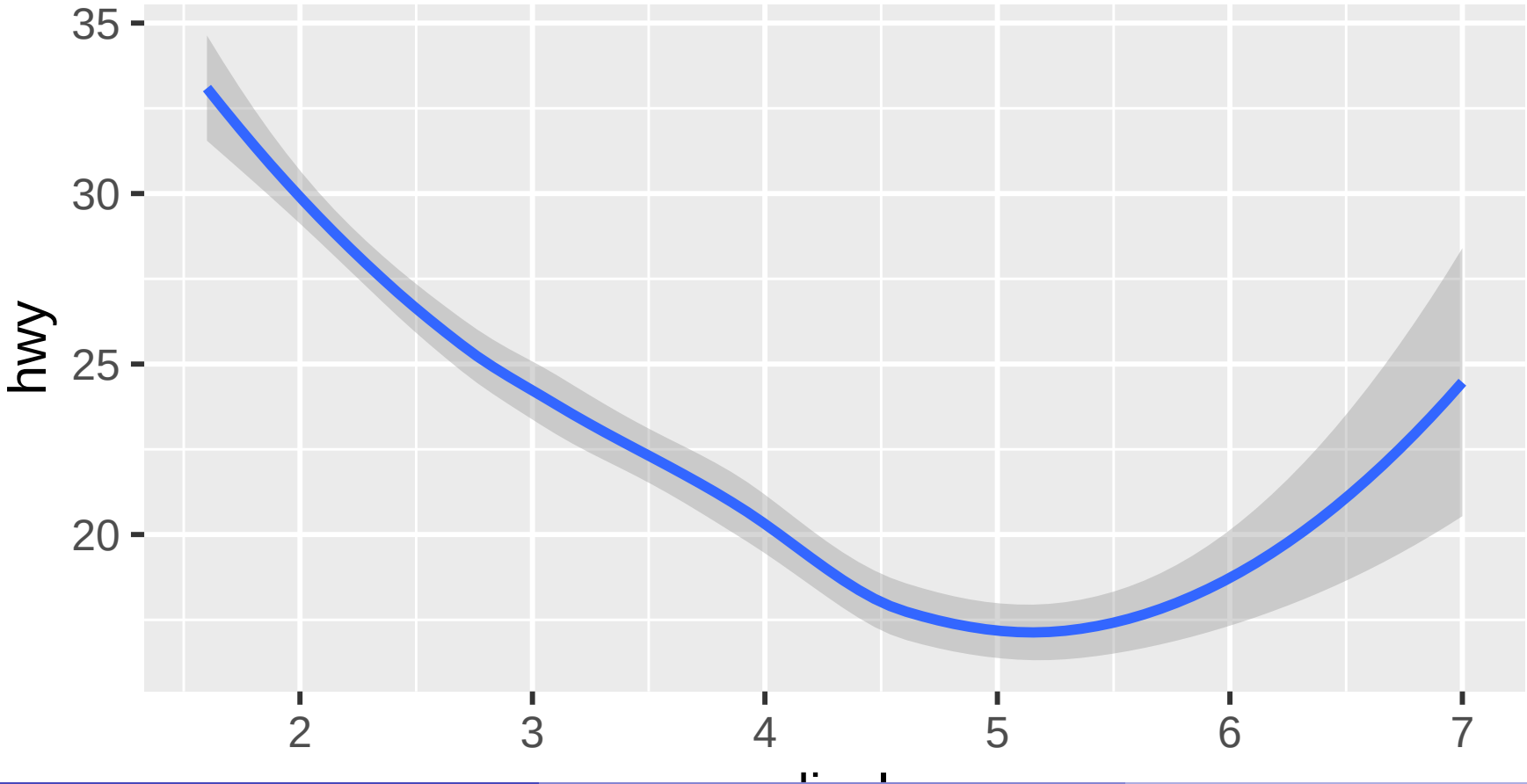
Geoms

- A geom is a geometrical object that a plot uses to represent data
- Bar charts use bar geoms, line charts use line geoms, and scatter plots use the point geom.
- To change the geom in your plot, simply change the geom function that is added to the ggplot call.

Same data - geom 1

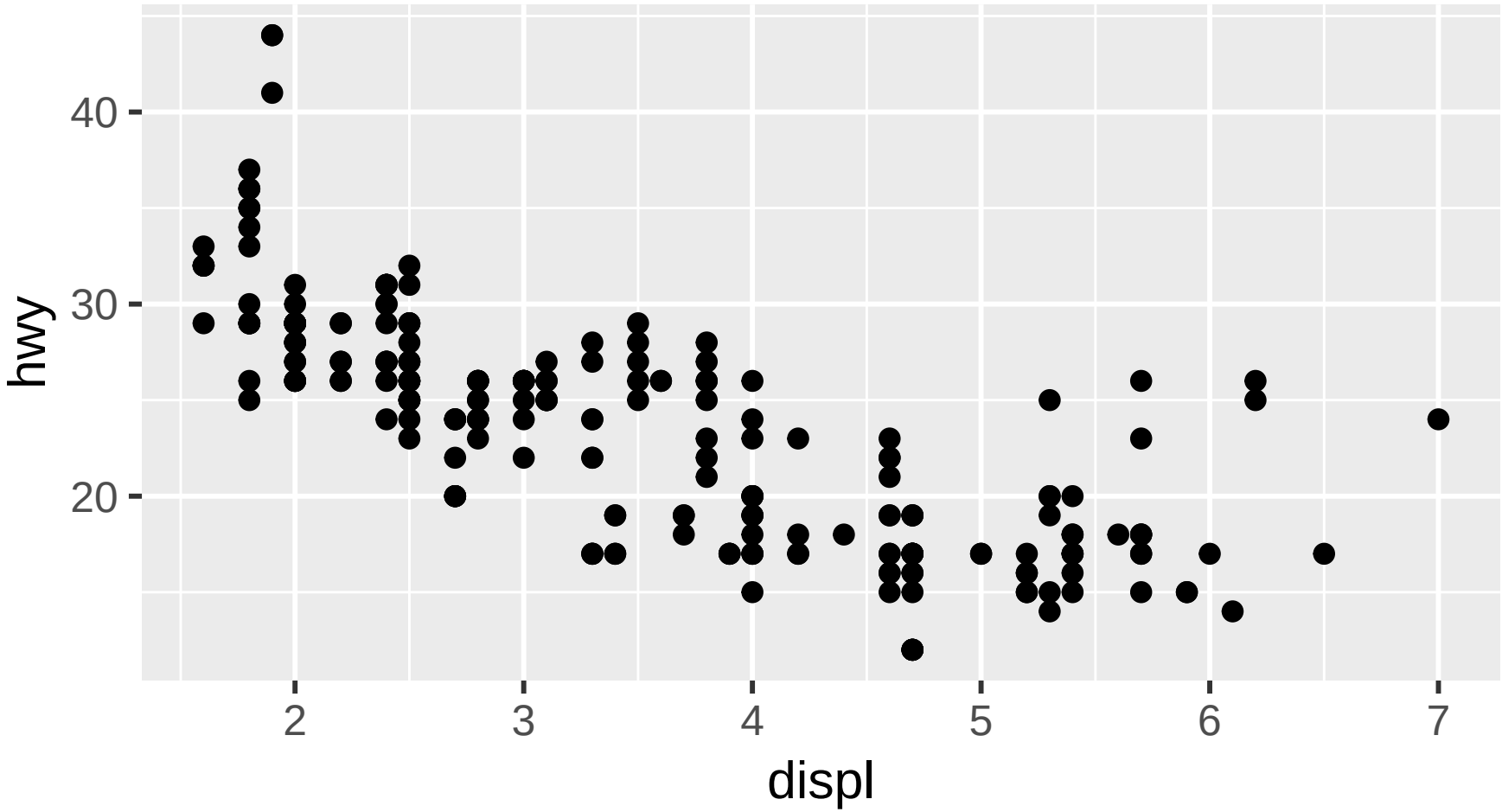
```
ggplot(data=d)+
  geom_smooth(aes(x=displ,y=hwy))
```

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



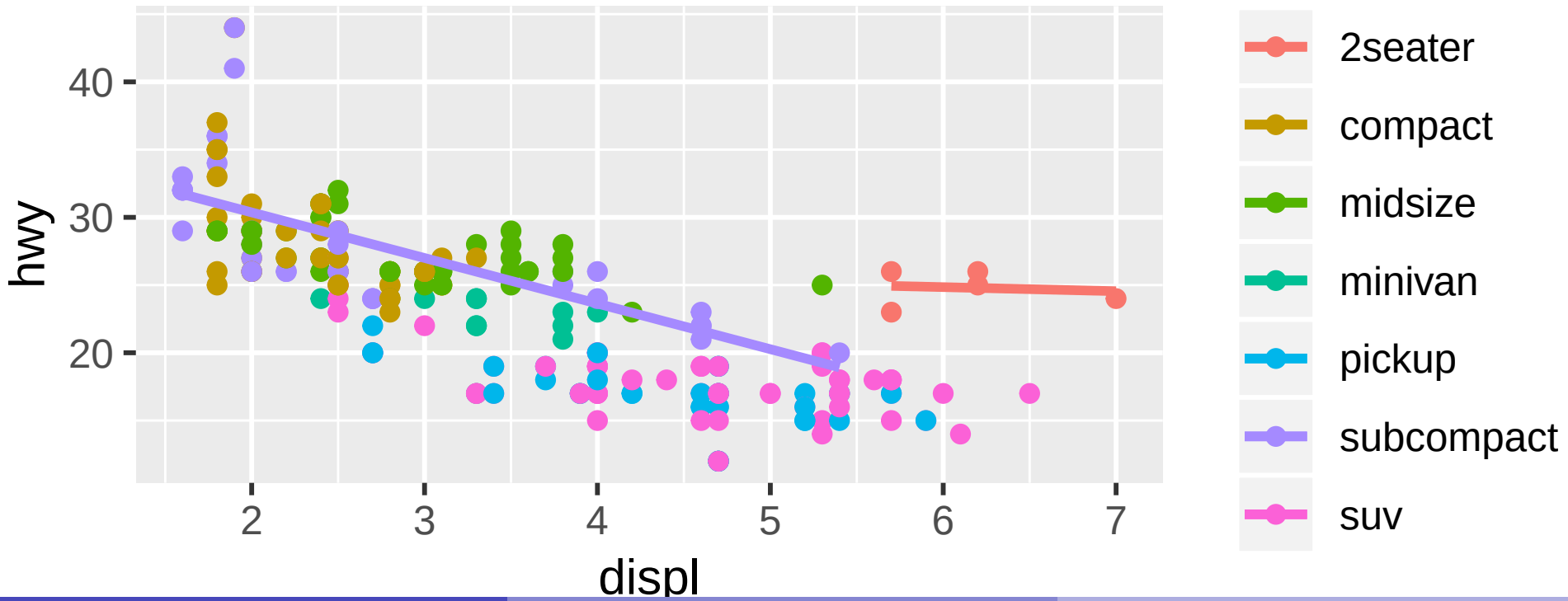
Same data - geom 2

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



Using different data sources

```
ggplot(data=d,aes(x=displ,y=hwy,colour=class))+
  geom_point()+
  geom_smooth(data=filter(d,class=="subcompact"),
              method="lm",se = F)+
  geom_smooth(data=filter(d,class=="2seater"),
              method="lm",se = F)
```



Sample plot geoms

Geom	Purpose
geom_smooth()	Fits a smoother to data and displays the smooth and its standard error
geom_boxplot()	Produces a box-and-whisker plot to summarise the distribution of a set of points
geom_histogram() geom_freqpoly()	Shows the distribution of continuous variables
geom_bar()	Shows the distribution of categorical variables
geom_path() geom_line()	Draws lines between data points
geom_area()	Draws an area plot, which is a line plot filled to the y-axis. Multiple groups will be stacked upon each other
geom_rect() geom_tile() geom_raster()	Draw rectangles
geom_polygon()	Draws polygons, which are filled paths.

Diamonds Data Set

A dataset containing the prices and other attributes of almost 54,000 diamonds

Table 2: Selected sample from diamonds data set

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
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48
0.24	Very Good	I	VVS1	62.3	57	336	3.95	3.98	2.47
0.26	Very Good	H	SI1	61.9	55	337	4.07	4.11	2.53
0.22	Fair	E	VS2	65.1	61	337	3.87	3.78	2.49
0.23	Very Good	H	VS1	59.4	61	338	4.00	4.05	2.39

Explanation of Variables

Feature	Explanation
price	price in US dollars $326–$18,823
carat	weight of the diamond (0.2–5.01)
cut	quality of the cut (Fair, Good, Very Good, Premium, Ideal)
color	diamond colour, from J (worst) to D (best)
clarity	a measurement of how clear the diamond is (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
x	length in mm (0–10.74)
y	width in mm (0–58.9)
z	depth in mm (0–31.8)
depth	total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43–79)
table	width of top of diamond relative to widest point (43–95)

Diamonds summary

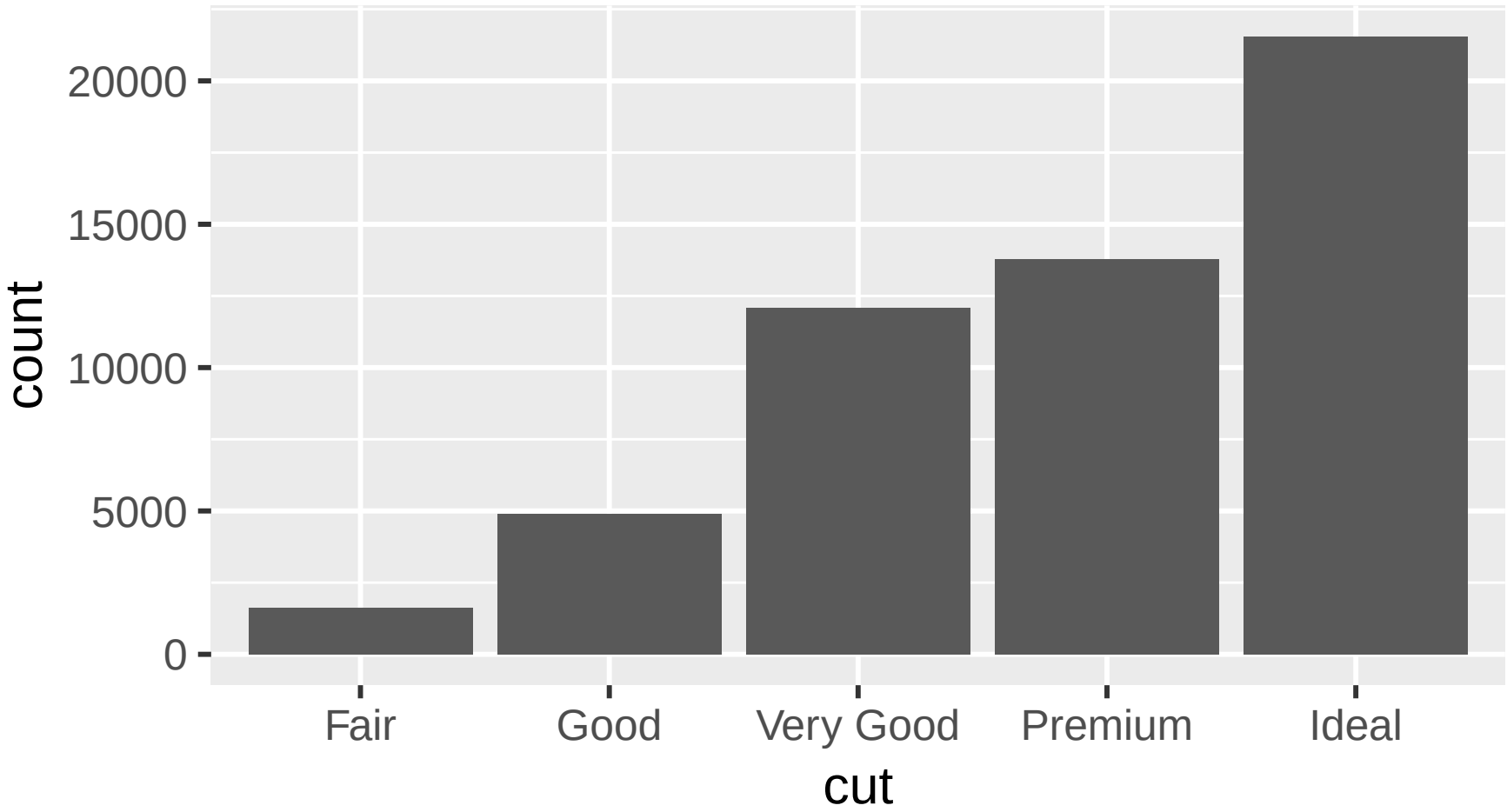
```
> summary(diamonds)
     carat               cut        color        clarity          depth      
 Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065   Min.   :43.00  
 1st Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258   1st Qu.:61.00  
 Median :0.7000   Very Good:12082   F: 9542   SI2     : 9194   Median :61.80  
 Mean   :0.7979   Premium   :13791   G:11292   VS1     : 8171   Mean   :61.75  
 3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2    : 5066   3rd Qu.:62.50  
 Max.   :5.0100                     I: 5422   VVS1    : 3655   Max.   :79.00  
                                    J: 2808   (Other): 2531                  
     table           price             x                y                z          
 Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
 1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910  
 Median :57.00   Median : 2401   Median : 5.700   Median : 5.710   Median : 3.530  
 Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735   Mean   : 3.539  
 3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040  
 Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900   Max.   :31.800  
```

Statistical Transformations

- Many graphs, like scatterplots, plot the raw values of the dataset
- However, other graphs (e.g. bar charts) calculate new values to plot
  - Bar charts, histograms and frequency polygons bin your data and plot bin counts, the number of points that fall in each bin
  - Smoothers fit a model to your data and the plot predictions from the model
  - Boxplots compute a robust summary of the distribution and display a specially formatted box

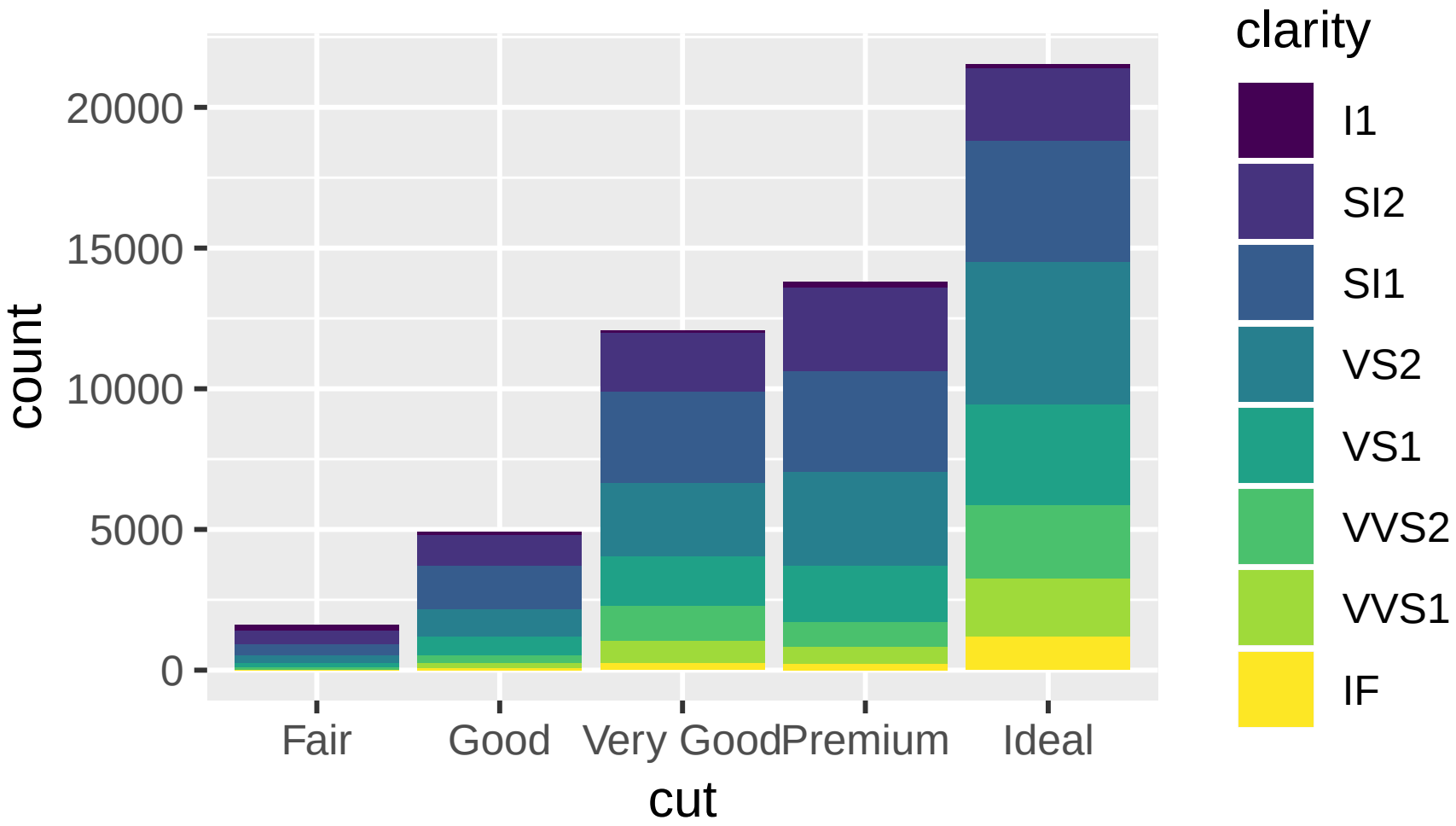
Bar Chart

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



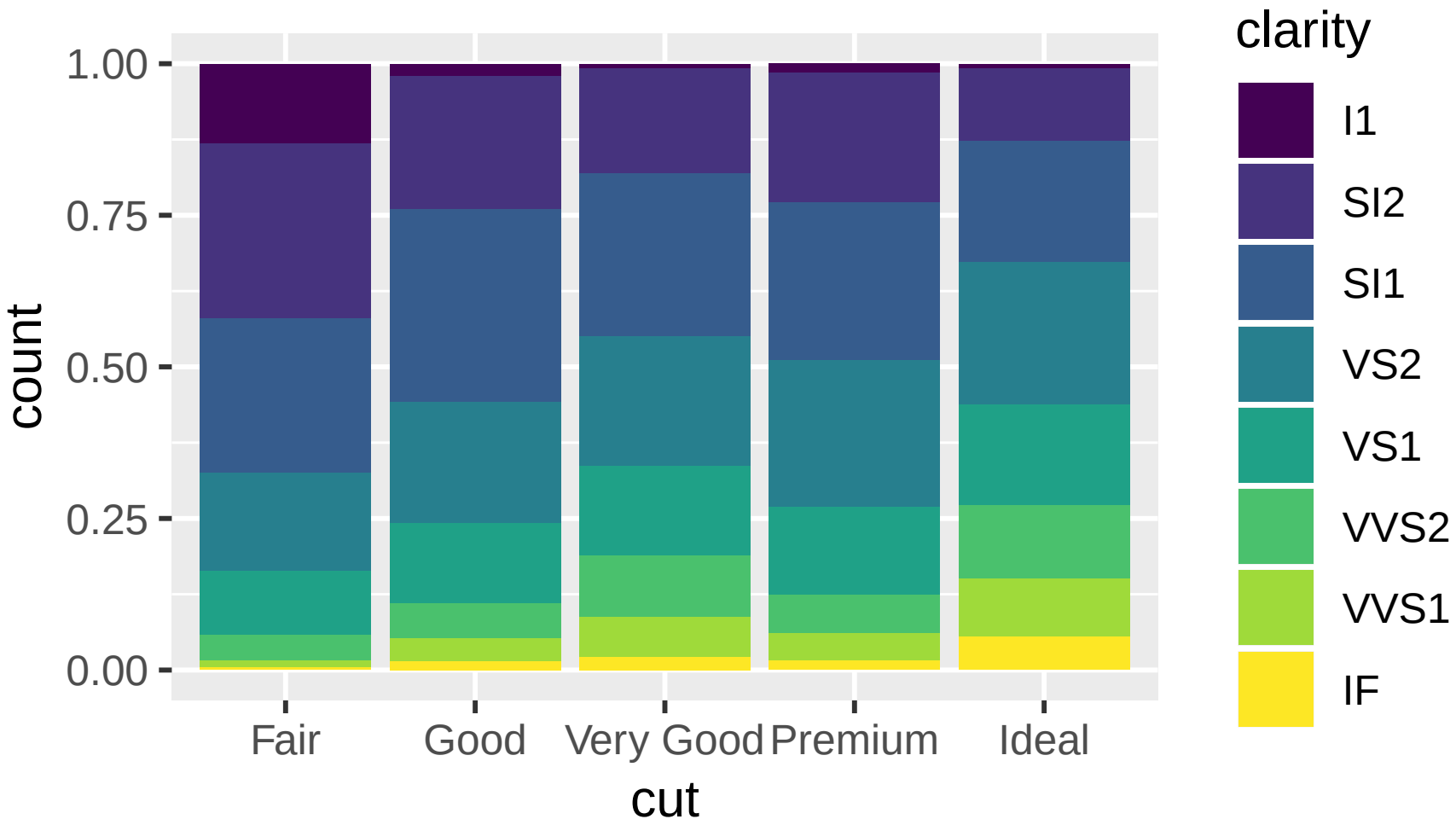
Bar Chart: Adding information with fill

```
ggplot(data=diamonds)+
  geom_bar(aes(x=cut,fill=clarity))
```



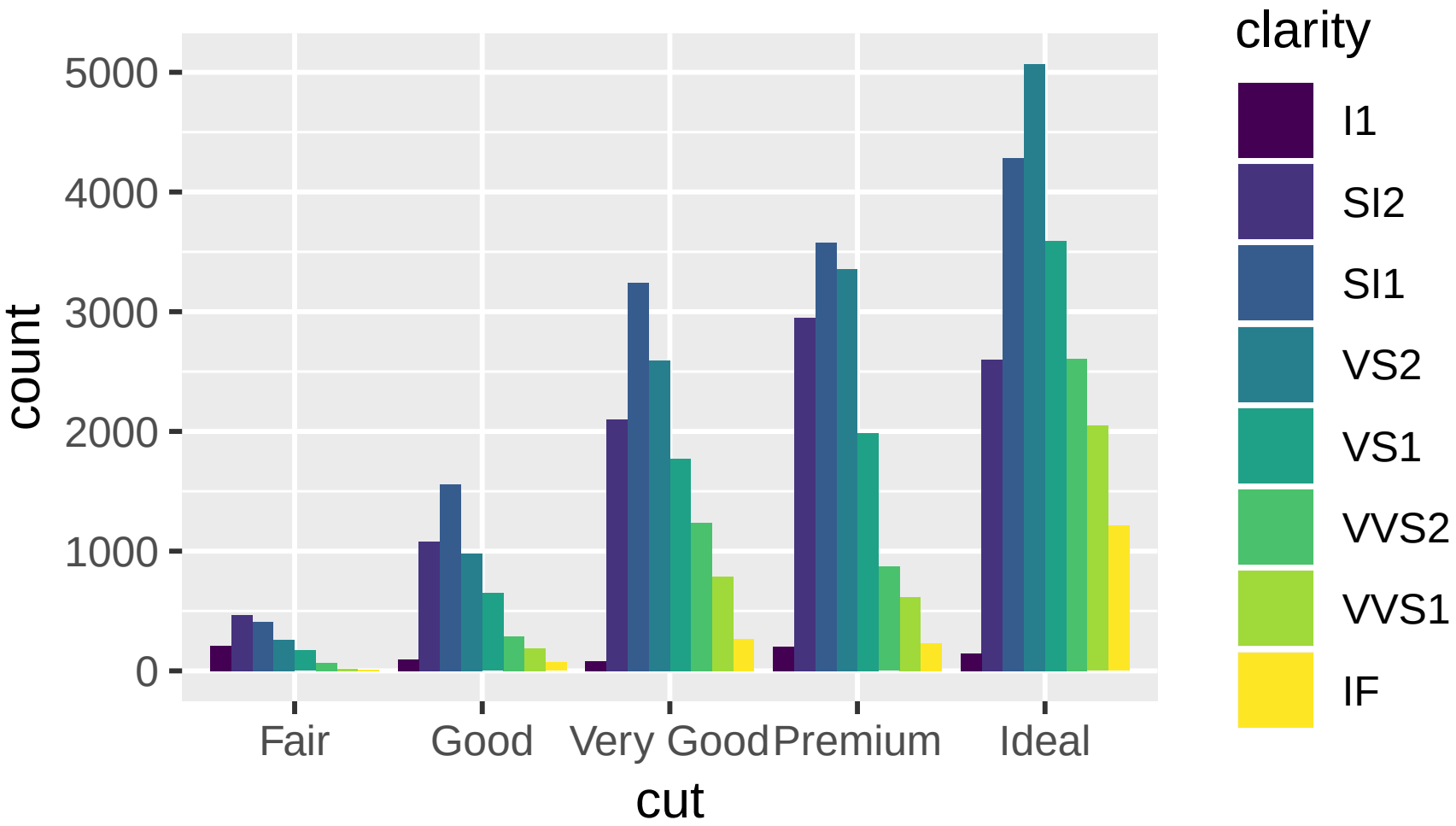
Bar Chart: Normalising Plot

```
ggplot(data=diamonds)+
  geom_bar(aes(x=cut,fill=clarity),position="fill")
```



Bar Chart: side-by-side

```
ggplot(data=diamonds)+
  geom_bar(aes(x=cut,fill=clarity),position="dodge")
```

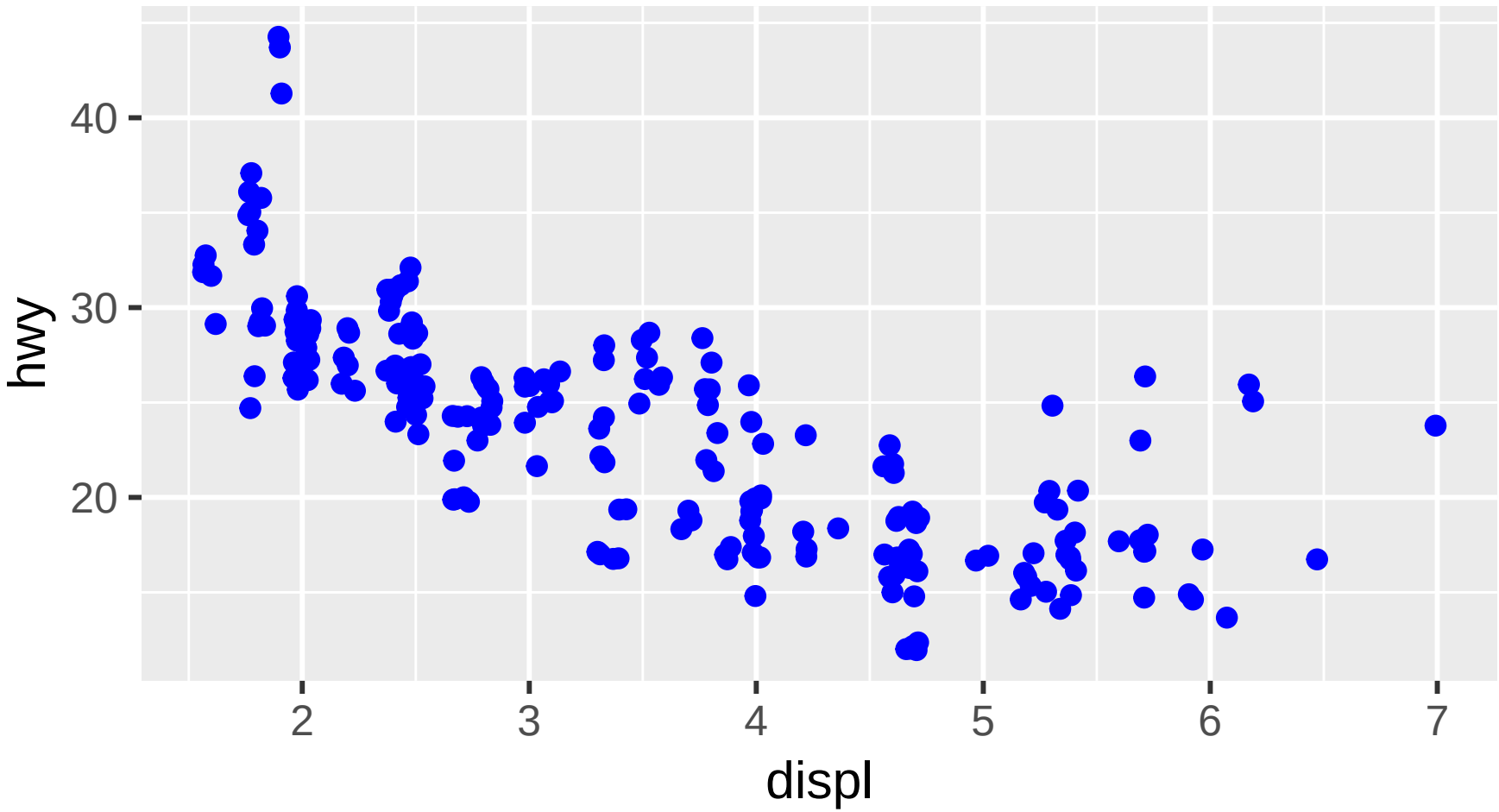


Additional Adjustment

- Recall our first scatterplot
- 126 points displayed, yet there are 234 observations
- Many points can overlap, so it makes it hard to see where the mass of data is
- Are all points spread equally, or is there one special combination that contains 129 values?
- "jitter" adds random noise to each point

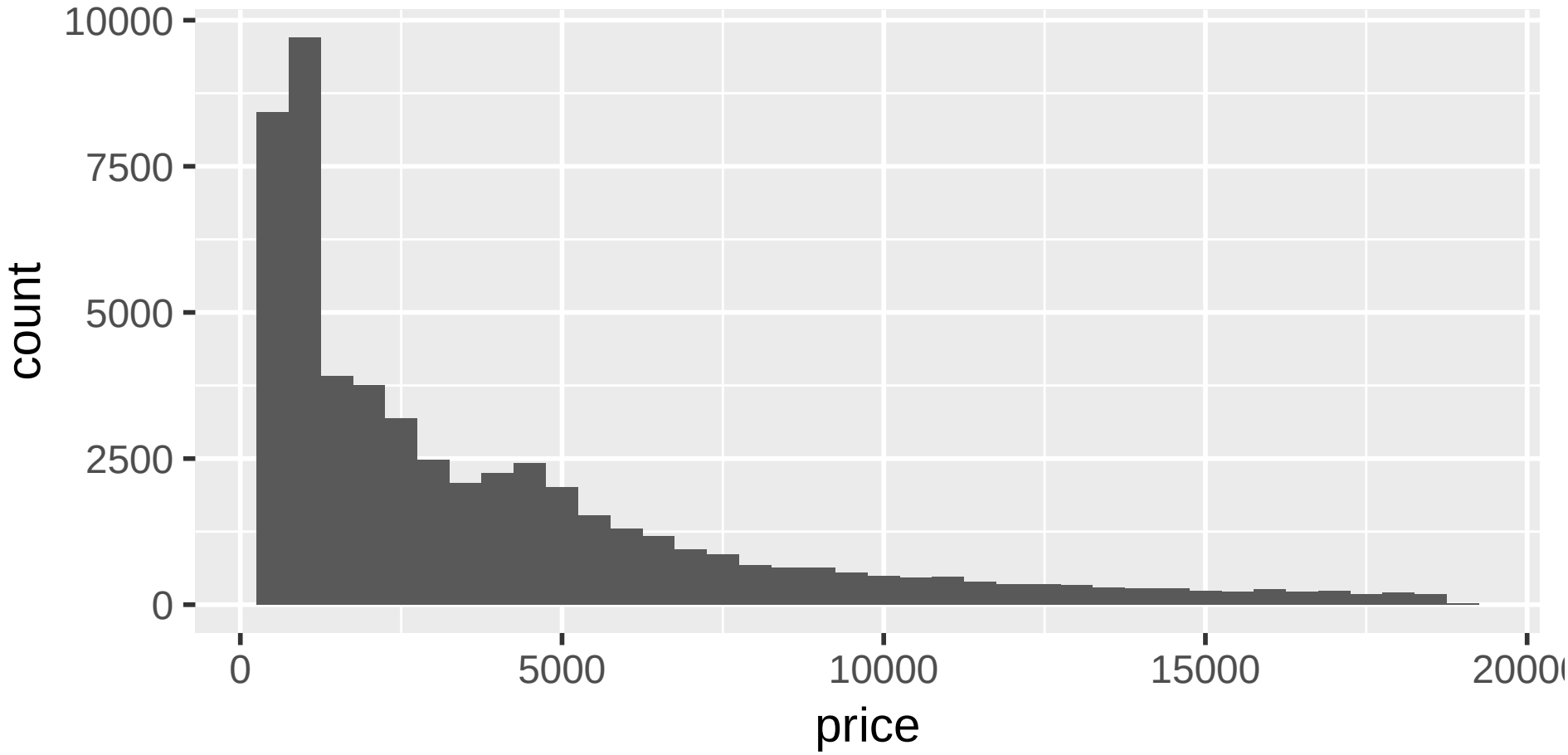
Using jitter

```
ggplot(data=mpg)+
  geom_point(aes(x=displ,y=hwy),
             colour="blue", position="jitter")
```



Histogram

```
ggplot(data=diamonds,aes(x=price))+
  geom_histogram(binwidth = 500)
```

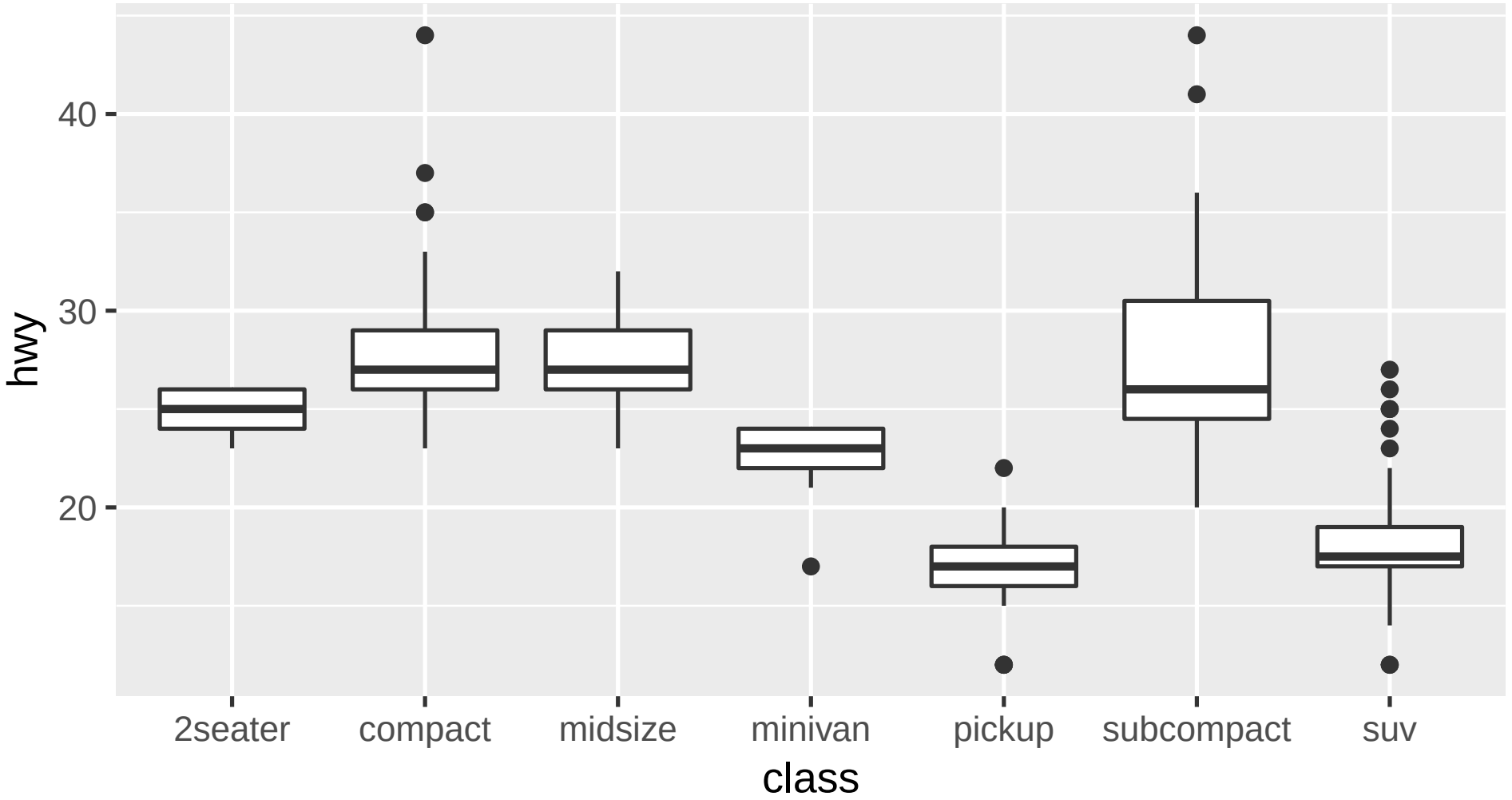


Boxplot

- Display the distribution of a continuous variable broken down by a categorical variable
- Box that stretches from the 25th to 75th percentile a distance known as the interquartile range (IRQ)
- Median in the middle of box
- Points outside more that 1.5 times the IQR from either edge of the box are displayed (outliers)
- Whisker extends to the farthest non-outlier point in the distribution

Boxplot Example

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



Summary

- The ggplot2 approach can be summarised by a template
- It can take seven parameters, but usually not all need to be applied (defaults used)
- These seven parameters comprise the **grammar of graphics**

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