# Large data

**Hadley Wickham**

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

**November 2010**

1. The diamonds data
2. Histograms and bar charts
3. Frequency polygons
4. Scatterplots for large data

# Diamonds

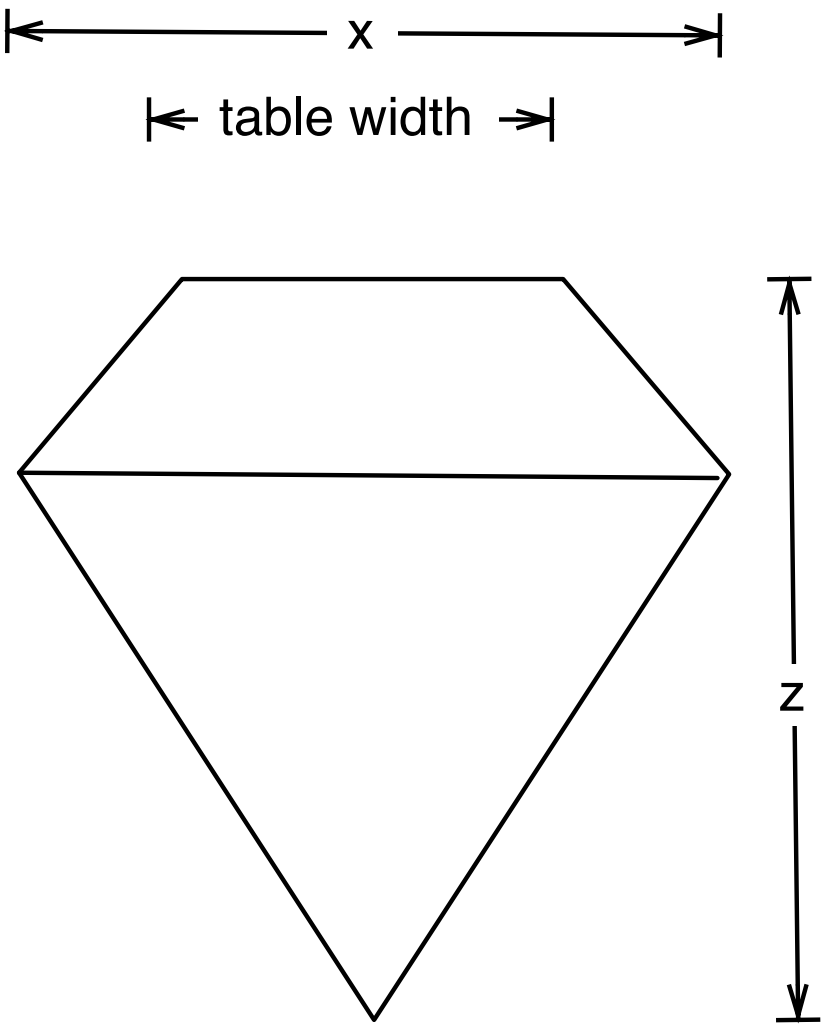
# Diamonds data

~**54,000** round diamonds from http://www.diamondse.info/

Carat, colour, clarity, cut

Total depth, table, depth, width, height

Price




depth = z / diameter
table = table width / x * 100

# Histogram & bar charts

# Histograms and barcharts

Used to display the **distribution** of a variable

Categorical variable → bar chart

Continuous variable → histogram

# Examples

```
# With only one variable, qplot guesses that
# you want a bar chart or histogram
qplot(cut, data = diamonds)

qplot(carat, data = diamonds)
qplot(carat, data = diamonds, binwidth = 1)
qplot(carat, data = diamonds, binwidth = 0.1)
qplot(carat, data = diamonds, binwidth = 0.01)
resol                         arat)

last_plot() + xlim(0, 3)
```

Common ggplot2 technique: adding together plot components

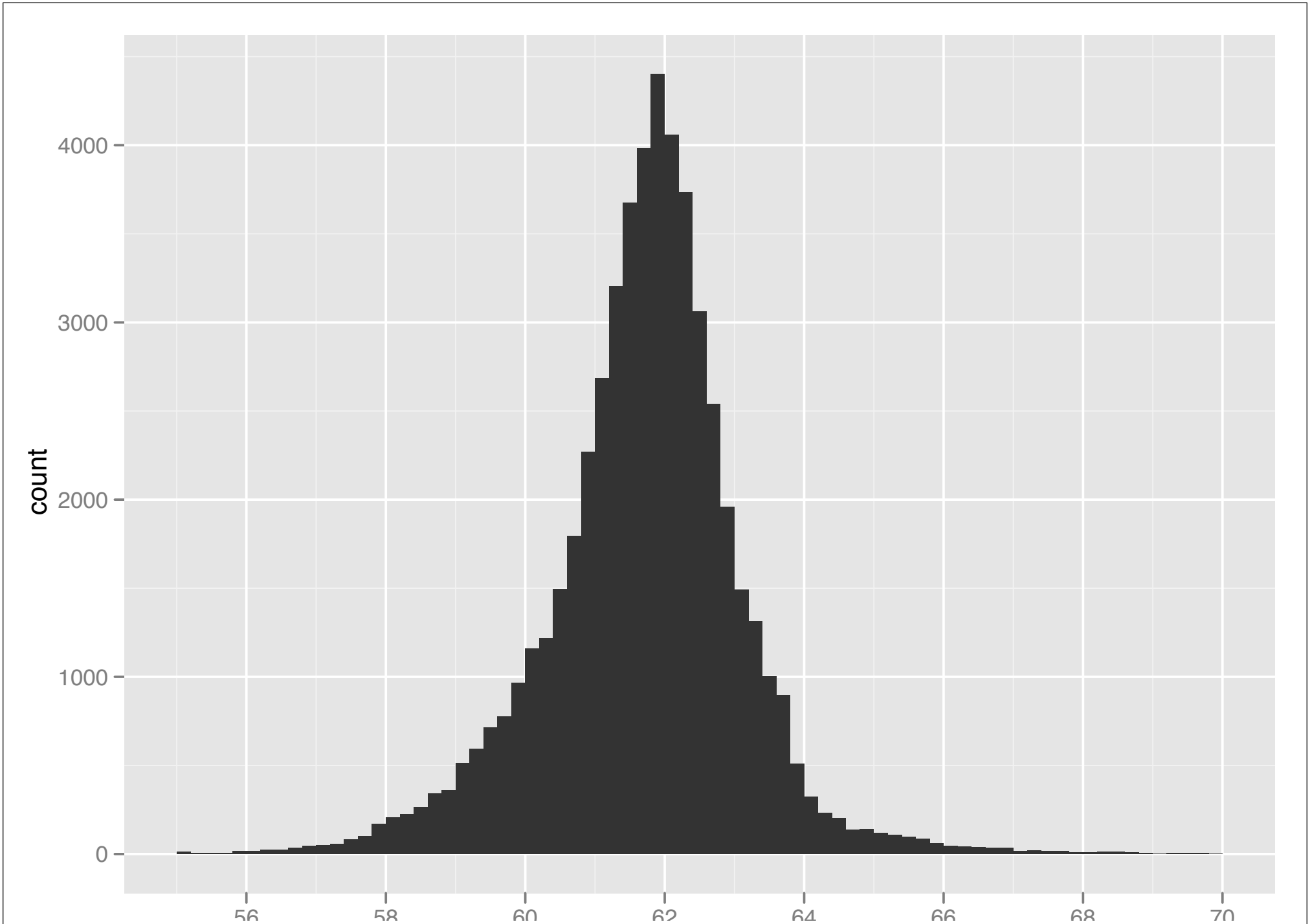
Always experiment with the bin width!

```
qplot(table, data = diamonds, binwidth = 1)

# To zoom in on a plot region use xlim() and ylim()
qplot(table, data = diamonds, binwidth = 1) +
   xlim(50, 70)
qplot(table, data = diamonds, binwidth = 0.1) +
  xlim(50, 70)
qplot(table, data = diamonds, binwidth = 0.1) +
  xlim(50, 70) + ylim(0, 50)

# Note that this type of zooming discards data
outside of the plot regions
# See coord_cartesian() for an alternative
```

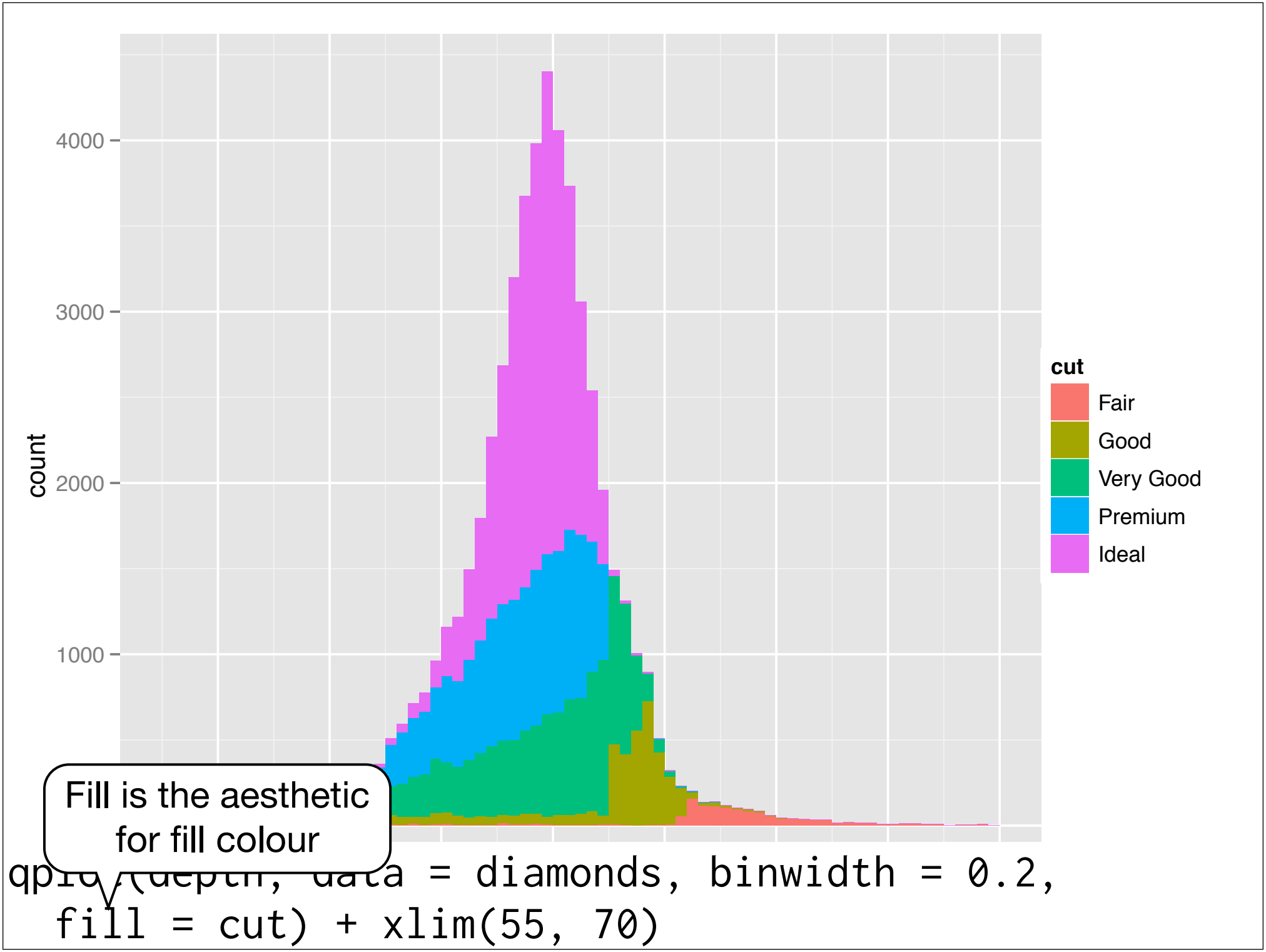
# Additional variables

As with scatterplots can use **aesthetics** or **faceting**.  Using aesthetics creates pretty, but ineffective, plots.

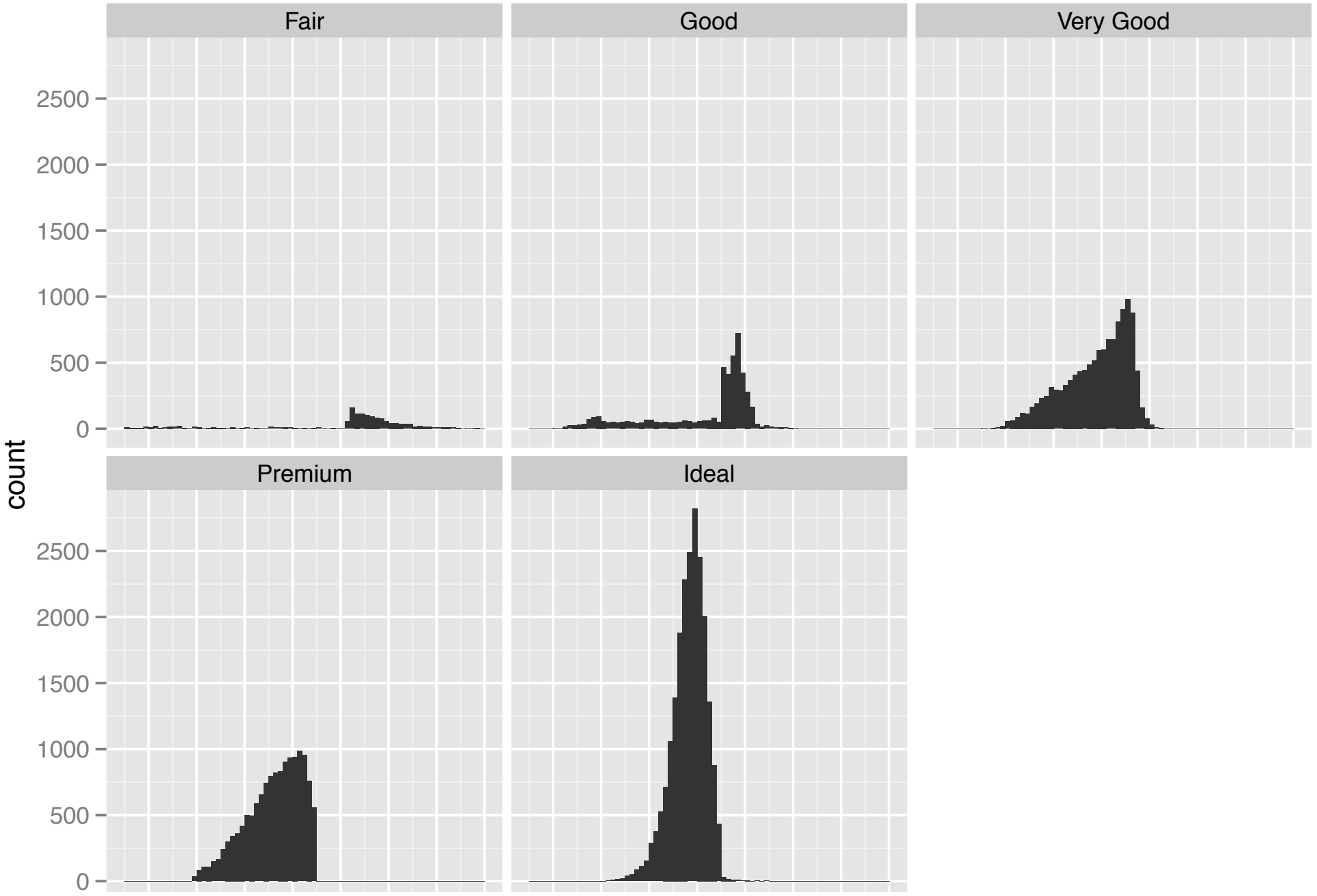
The following examples show the difference, when investigating the relationship between cut and depth.




```
qplot(depth, data = diamonds, binwidth = 0.2)
```




```
qplot(depth, data = diamonds, binwidth = 0.2,
  fill = cut) + xlim(55, 70)
```



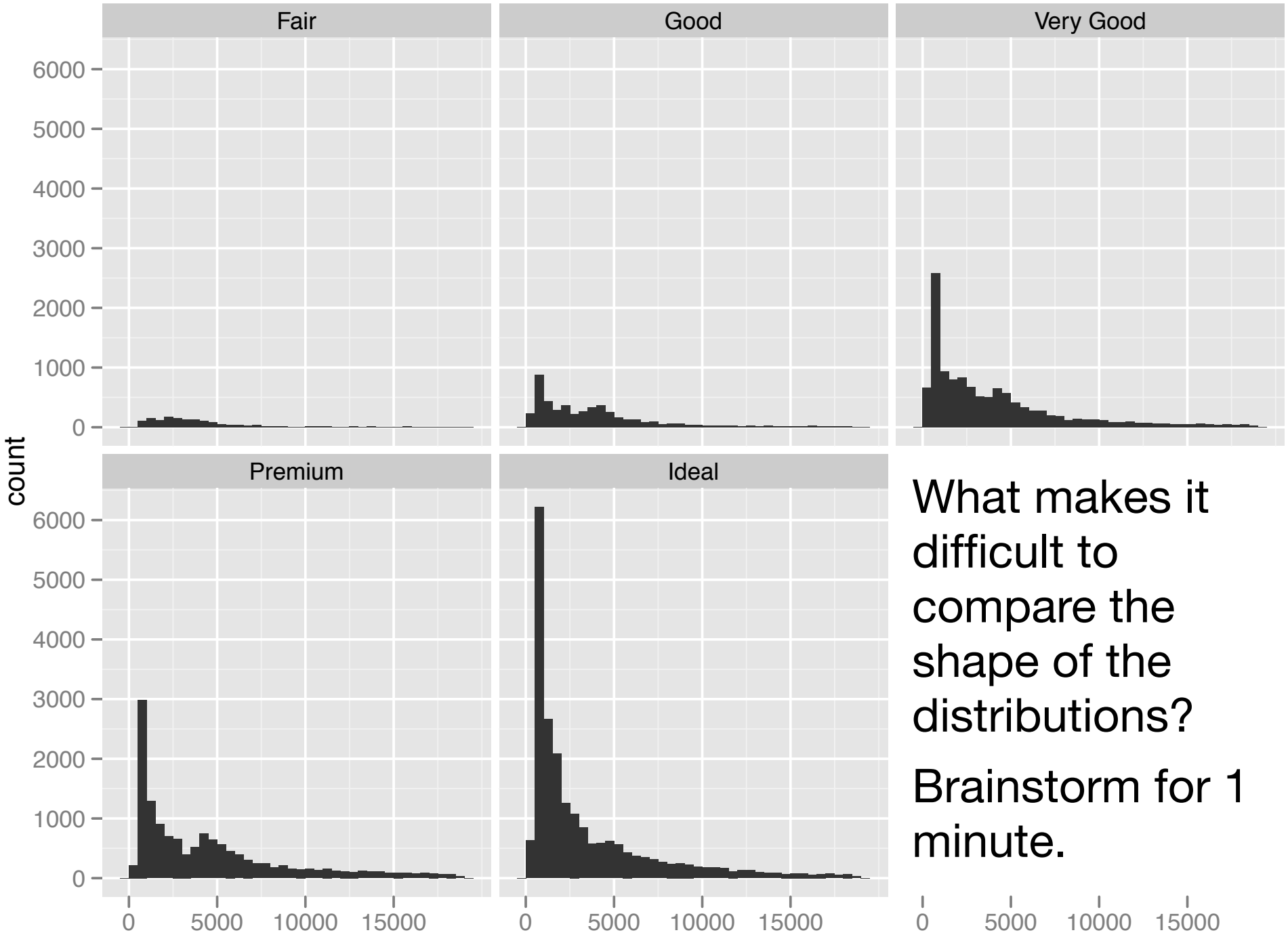

```
qplot(depth, data = diamonds, binwidth = 0.2) +
  xlim(55, 70) + facet_wrap(~ cut)
```

# Your turn

Explore the distribution of price.

How does it vary with colour, or cut?

Practice zooming in on regions of interest.




What makes it difficult to compare the shape of the distributions?

Brainstorm for 1 minute.

```
qplot(price, data = diamonds, binwidth = 500) + facet_wrap(~ cut)
```

# Problems

Each histogram far away from the others, but we know stacking is hard to read → *use another way of displaying densities*

Varying relative abundance makes comparisons difficult → *rescale to ensure constant area*

```
# Large distances make comparisons hard
qplot(price, data = diamonds, binwidth = 500) +
  facet_wrap(~ cut)

# Stacked heights hard to compare
qplot(price, data = diamonds, binwidth = 500, fill = cut)

# Much better - but still have differing relative abundance
qplot(price, data = diamonds, binwidth = 500,
  geom = "freqpoly", colour = cut)

# Instead of displaying count on y-axis, display density
# .. indicates that variable isn't in original data
qplot(price, ..density.., data = diamonds, binwidth = 500,
  geom = "freqpoly", colour = cut)

# To use with histogram, you need to be explicit
qplot(price, ..density.., data = diamonds, binwidth = 500,
  geom = "histogram") + facet_wrap(~ cut)
```

# Your turn

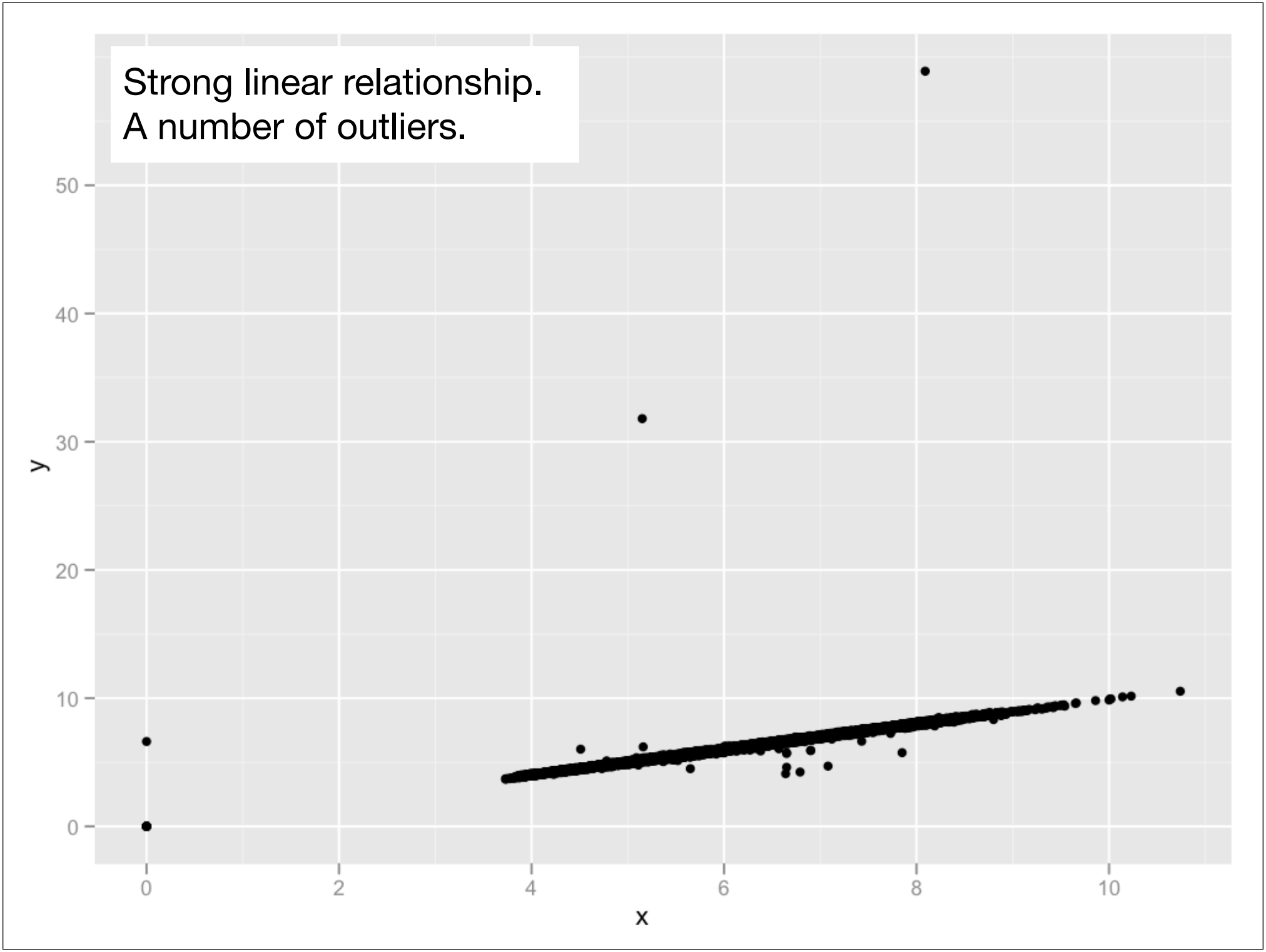
Practice using this technique to explore the relationship between price and cut, and carat and cut.

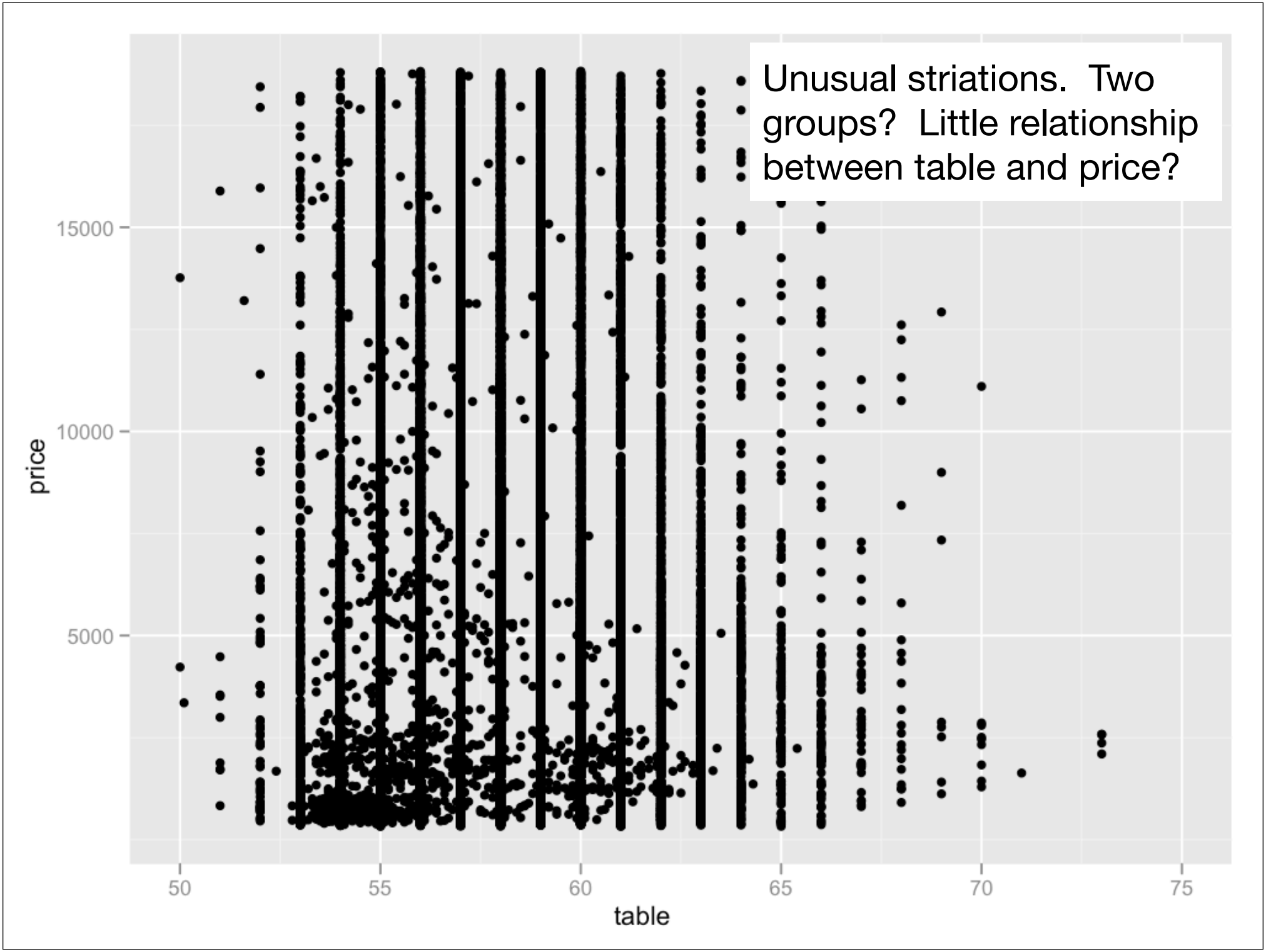
Do you see what you expect?

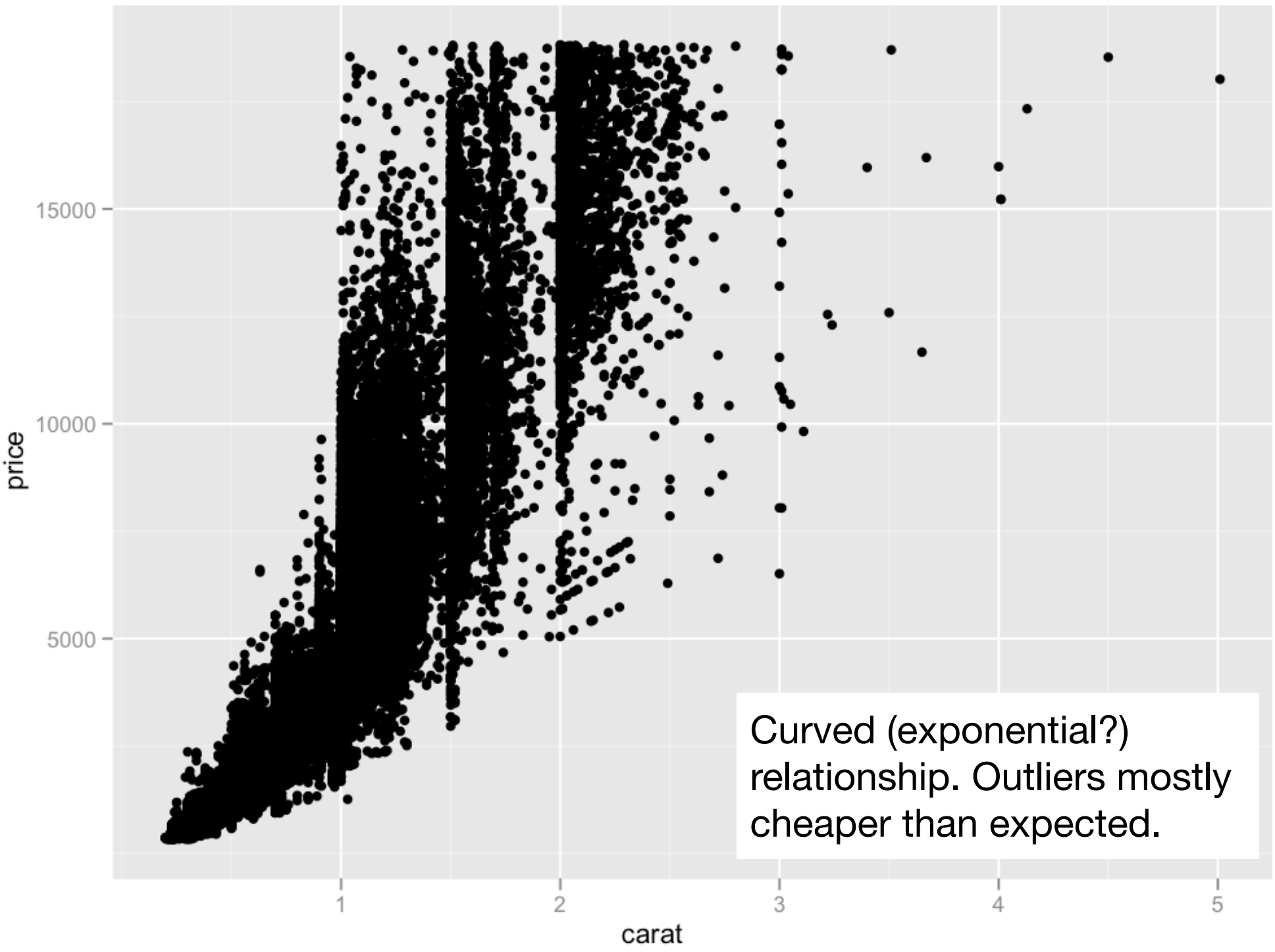
# Scatterplots

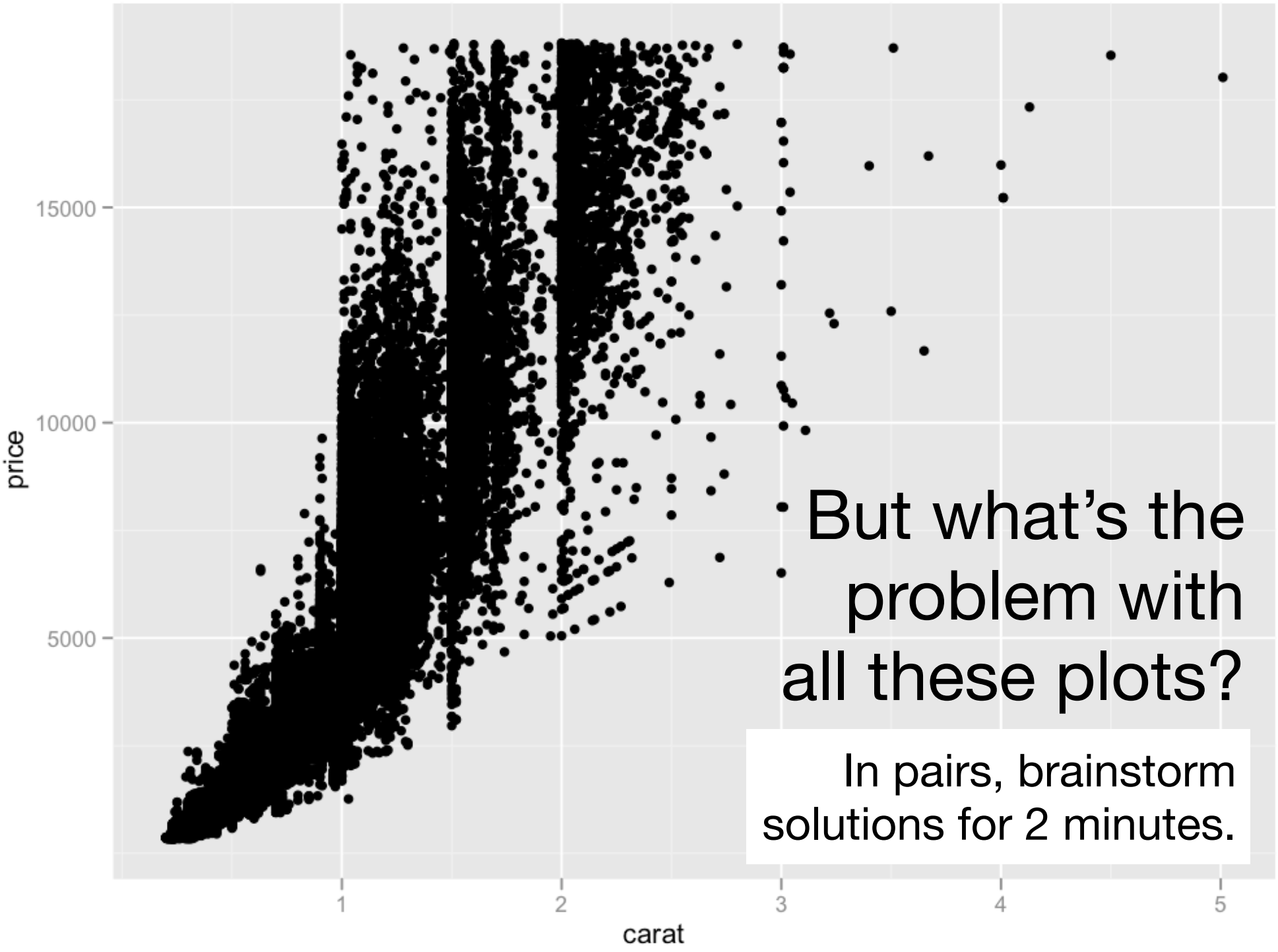
# Revision: interpretation

- Global patterns
- Local patterns
- Deviations


Strong linear relationship.
A number of outliers.
50
40
30
20
10
0
y
0
2
4
6
8
10
x


Unusual striations. Two groups? Little relationship between table and price?
price
15000
10000
5000
50
55
60
65
70
75
table


15000
10000
5000
price
1
2
3
4
5
carat
Curved (exponential?) relationship. Outliers mostly cheaper than expected.




# But what's the problem with all these plots?

In pairs, brainstorm solutions for 2 minutes.

| Idea | ggplot |
| --- | --- |
| Small points | `shape = I(".")` |
| Transparency | `alpha = I(1/50)` |
| Jittering | `geom = "jitter"` |
| Smooth curve | `geom = "smooth"` |
| 2d bins | `geom = "bin2d"` or `geom = "hex"` |
| Density contours | `geom = "density2d"` |

```
# There are two ways to add additional geoms
# 1) A vector of geom names:
qplot(price, carat, data = diamonds,
  geom = c("point", "smooth"))

# 2) Add on extra geoms
qplot(price, carat, data = diamonds) + geom_smooth()

# This how you get help about a specific geom:
?geom_smooth
# or go to http://had.co.nz/ggplot2/geom_smooth.html
```

```
# To set aesthetics to a particular value, you need
# to wrap that value in I()

qplot(price, carat, data = diamonds, colour = "blue")
qplot(price, carat, data = diamonds, colour = I("blue"))

# Practical application: varying alpha
qplot(price, carat, data = diamonds, alpha = I(1/10))
qplot(price, carat, data = diamonds, alpha = I(1/50))
qplot(price, carat, data = diamonds, alpha = I(1/100))
qplot(price, carat, data = diamonds, alpha = I(1/250))
```

# Your turn

Explore the relationship between carat, price and clarity, using these techniques.

(i.e. make this plot more informative:

`qplot(carat, price, data = diamonds, colour = cut)`)

Which did you find most useful?

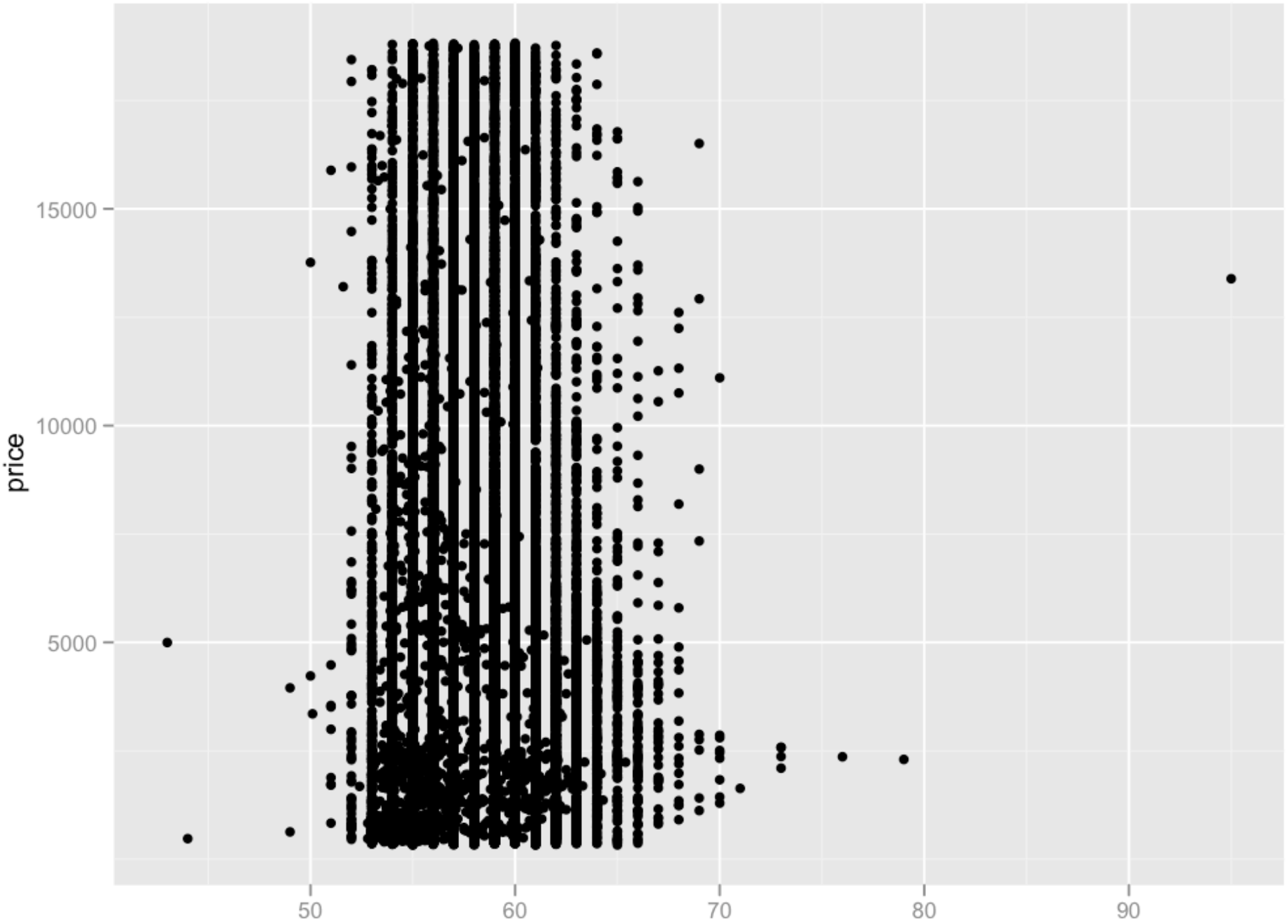
```
qplot(carat, price, data = diamonds,
  colour = clarity)
qplot(log10(carat), log10(price),
  data = diamonds, colour = clarity)
qplot(log10(carat), log10(carat / price),
  data = diamonds, colour = clarity)


qplot(log10(carat), log10(price), data = diamonds,
  geom = "hex", bins = 10) + facet_wrap(~ cut)
qplot(log10(carat), log10(price), data = diamonds,
  colour = cut, geom = "blank") + geom_smooth(method
= "lm")
```

```
mod <- lm(log10(price) ~ log10(carat),
  data = diamonds)

qplot(log10(carat), log10(price), data = diamonds,
  geom = "bin2d") +
  facet_wrap( ~ cut) +
  geom_abline(intercept = coef(mod)[1],
    slope = coef(mod)[2], size = 2)
```

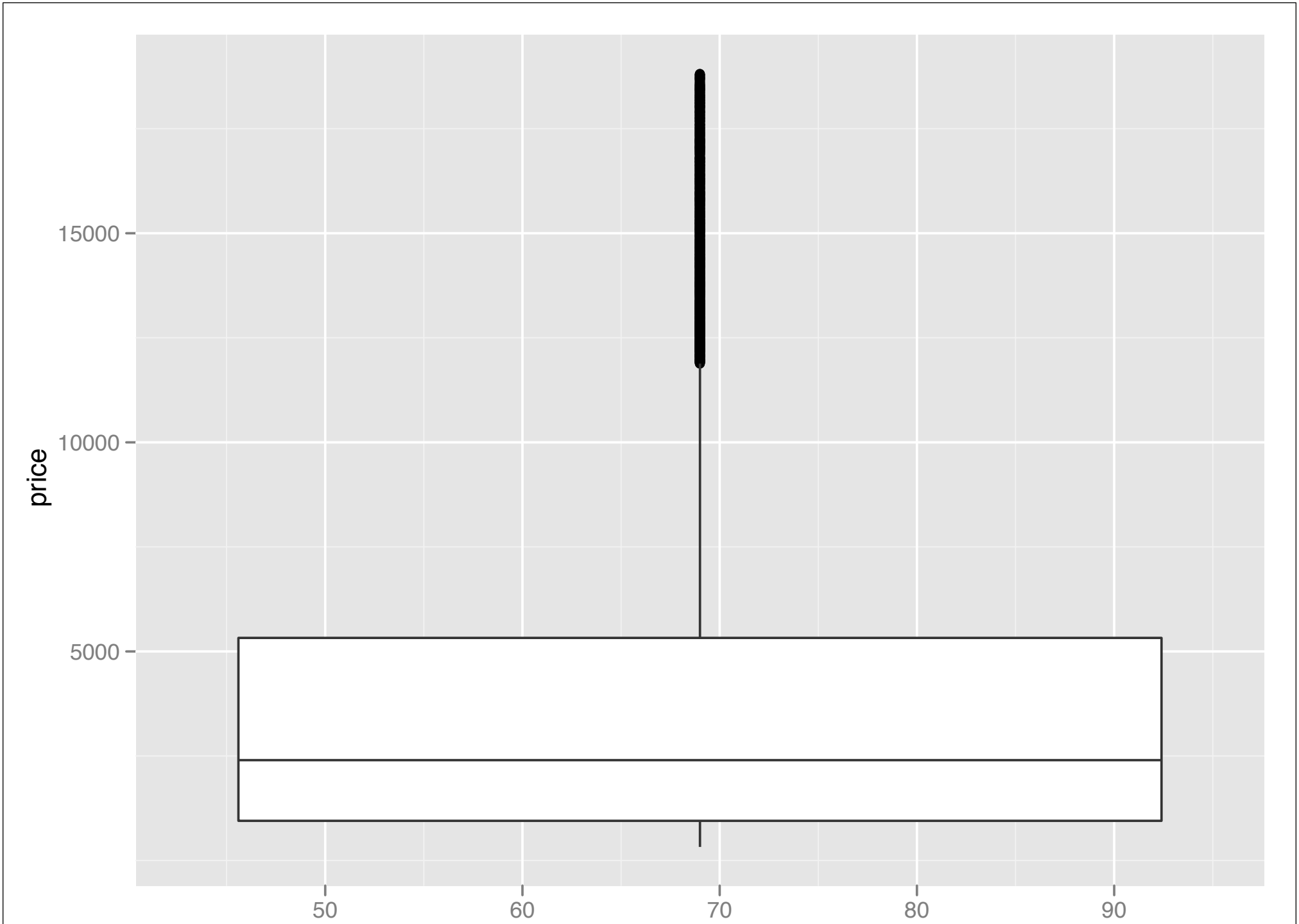
# Boxplots

Less information than a histogram, but take up much less space.

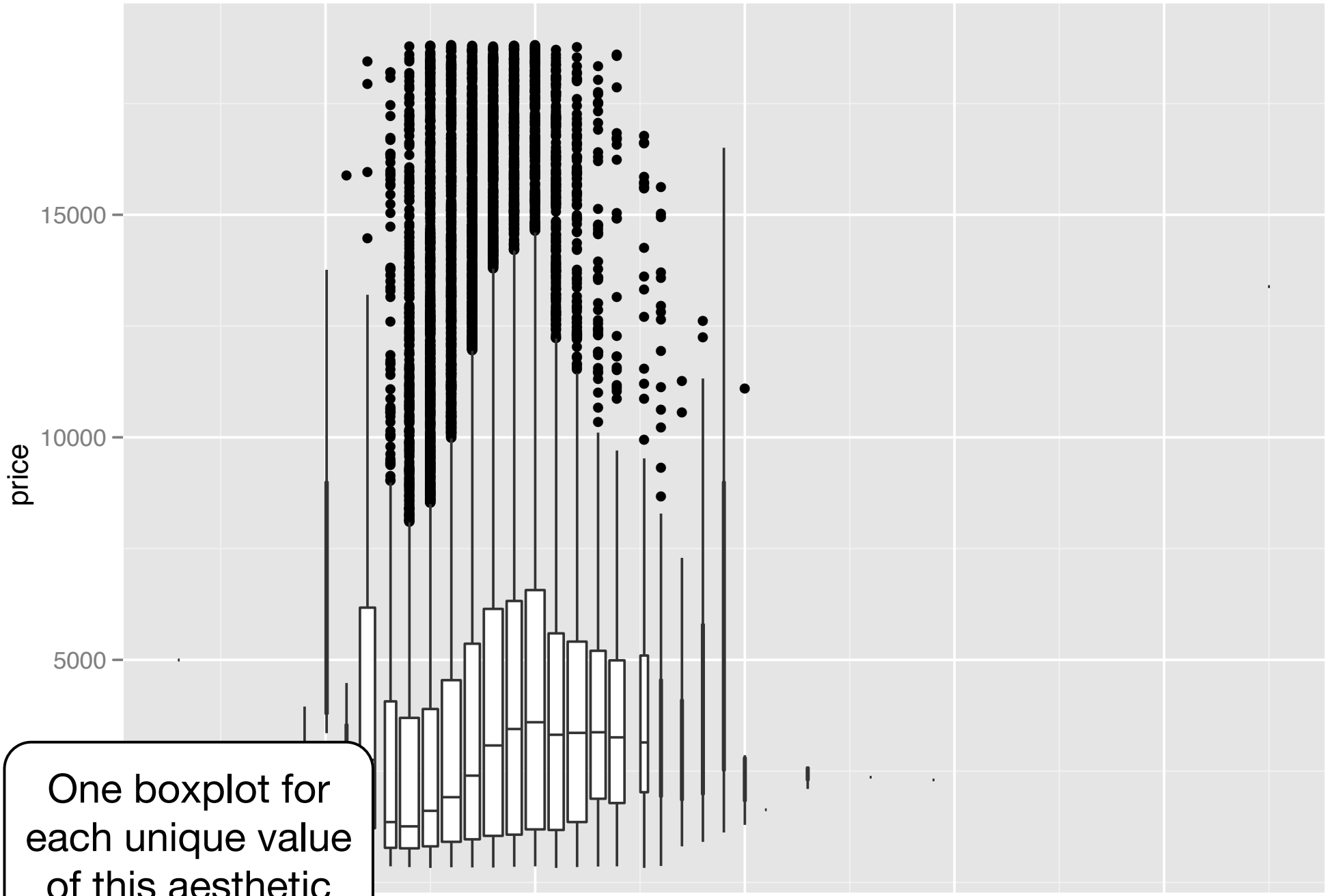
Already seen them used with discrete x values.  Can also use with continuous x values, by specifying how we want the data **group**ed.




```
qplot(table, price, data = diamonds)
```




```
qplot(table, price, data = diamonds, geom = "boxplot")
```




One boxplot for each unique value of this aesthetic

```
qplot(table, price, data = diamonds, geom = "boxplot",
  group = round(table))
```