Visualisation in R with ggplot2 and plyr

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

September 2011

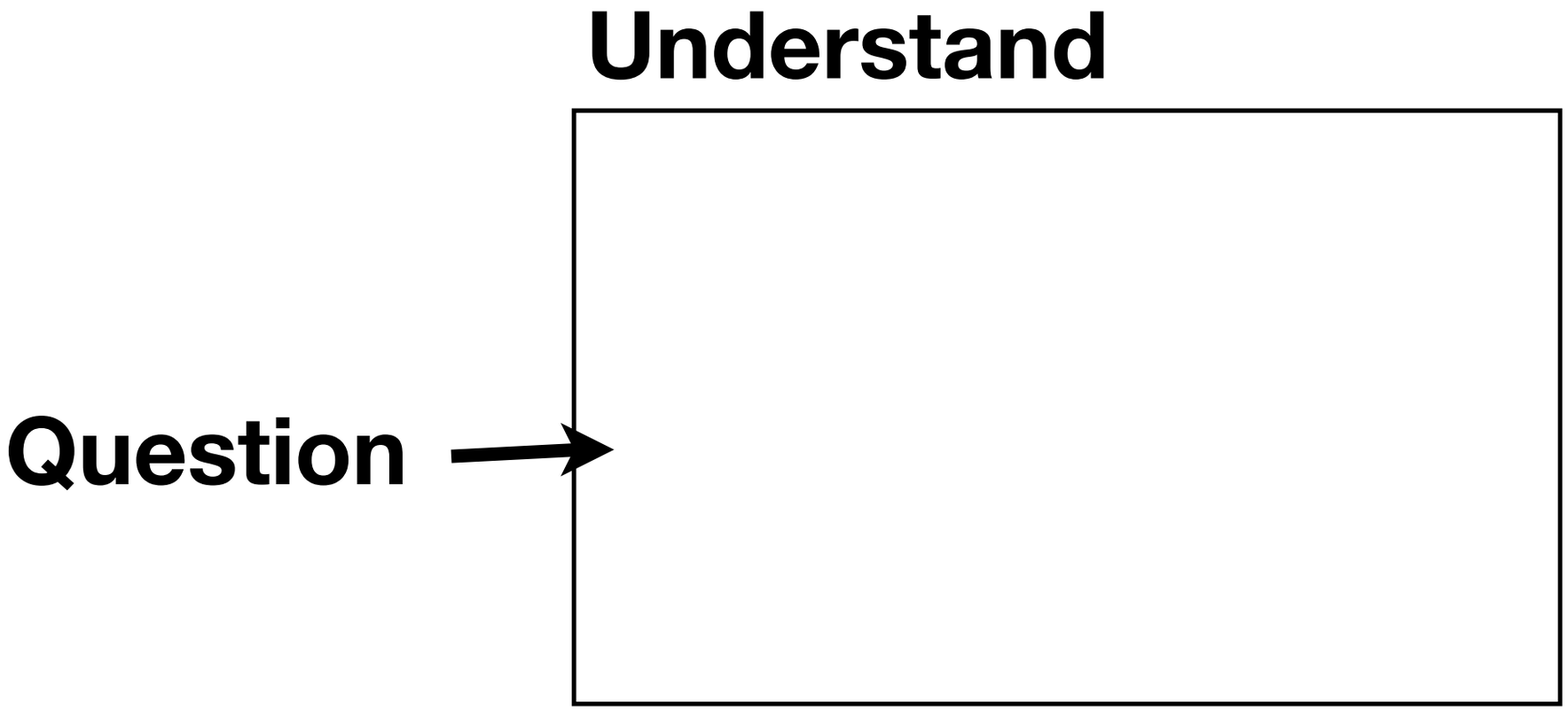
Outline

Why? Data analysis

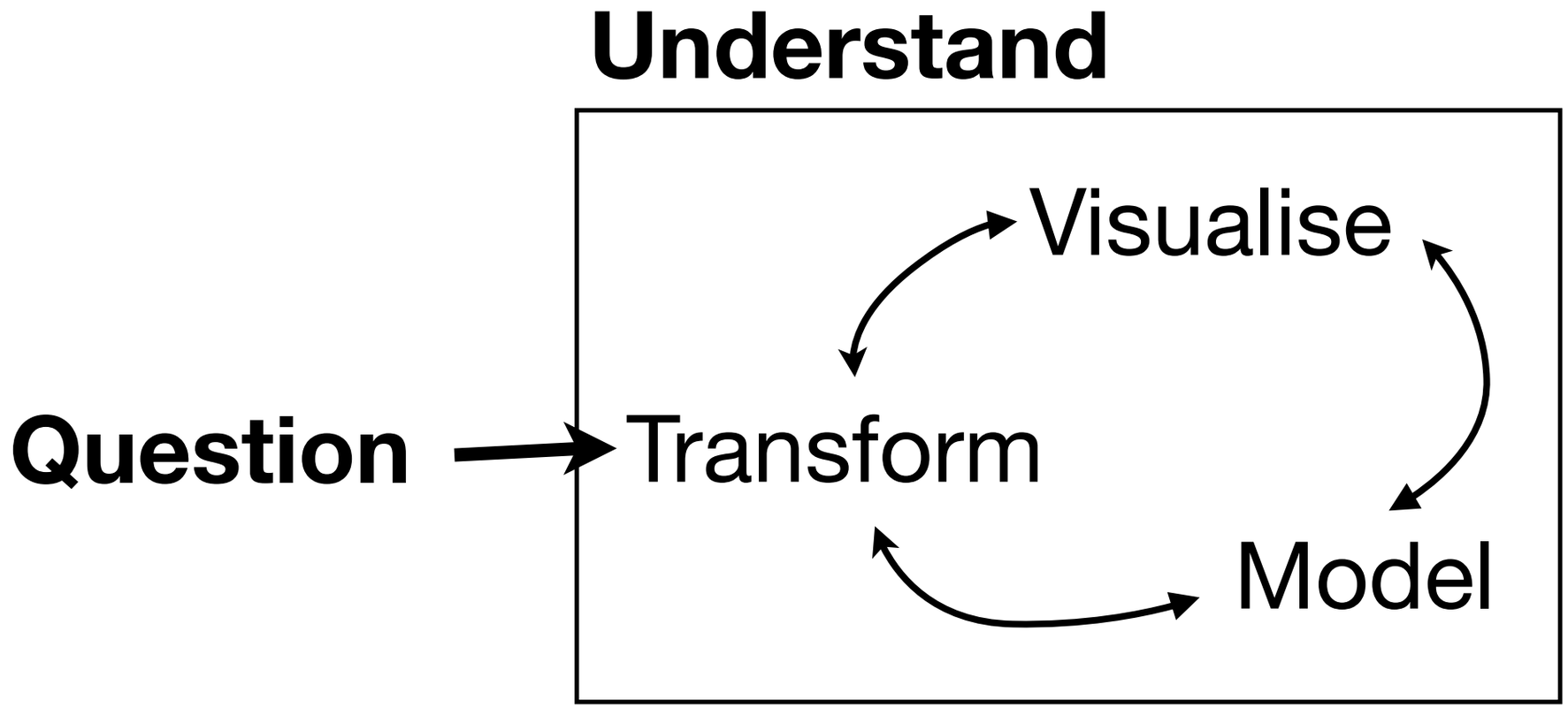
Why? Data analysis

Question

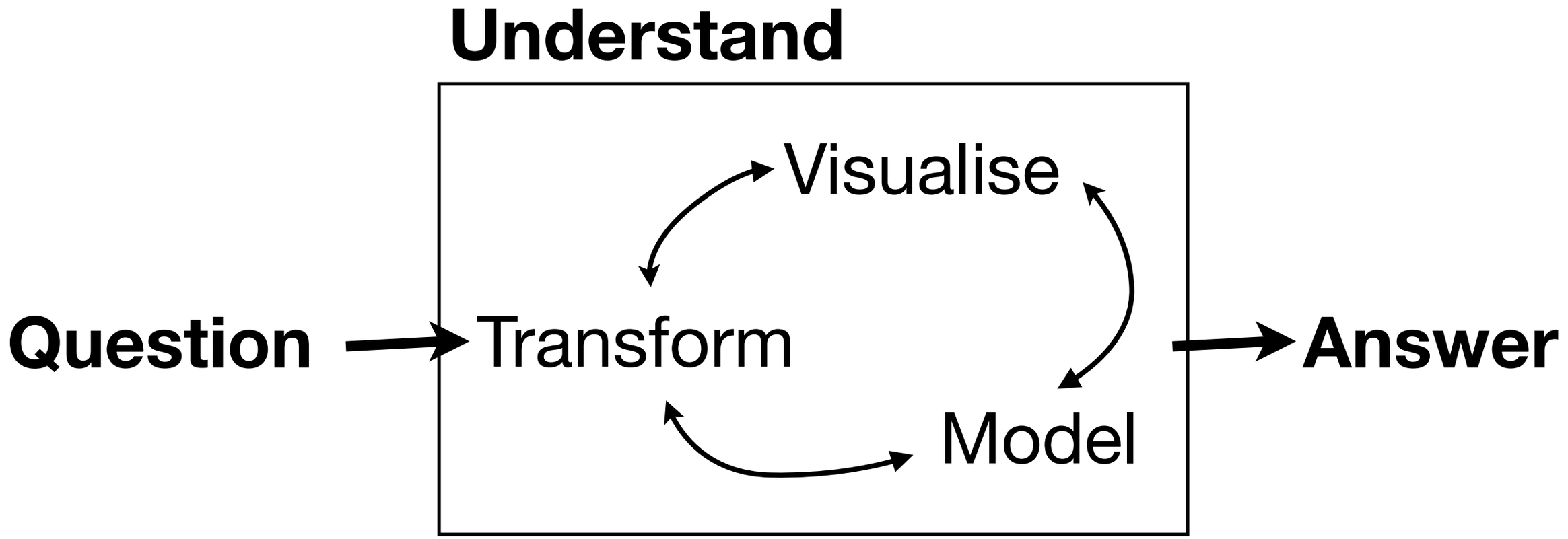
Why? Data analysis



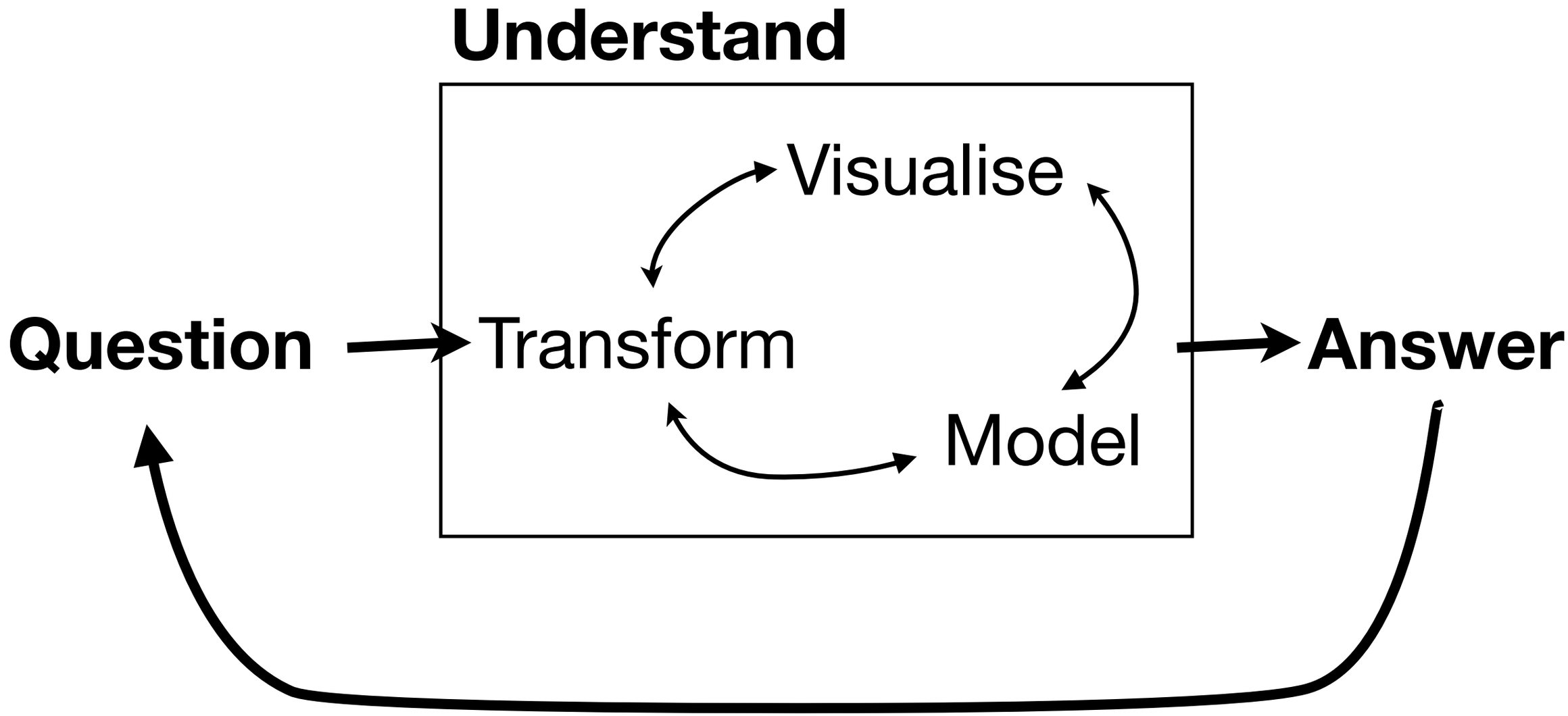
Why? Data analysis

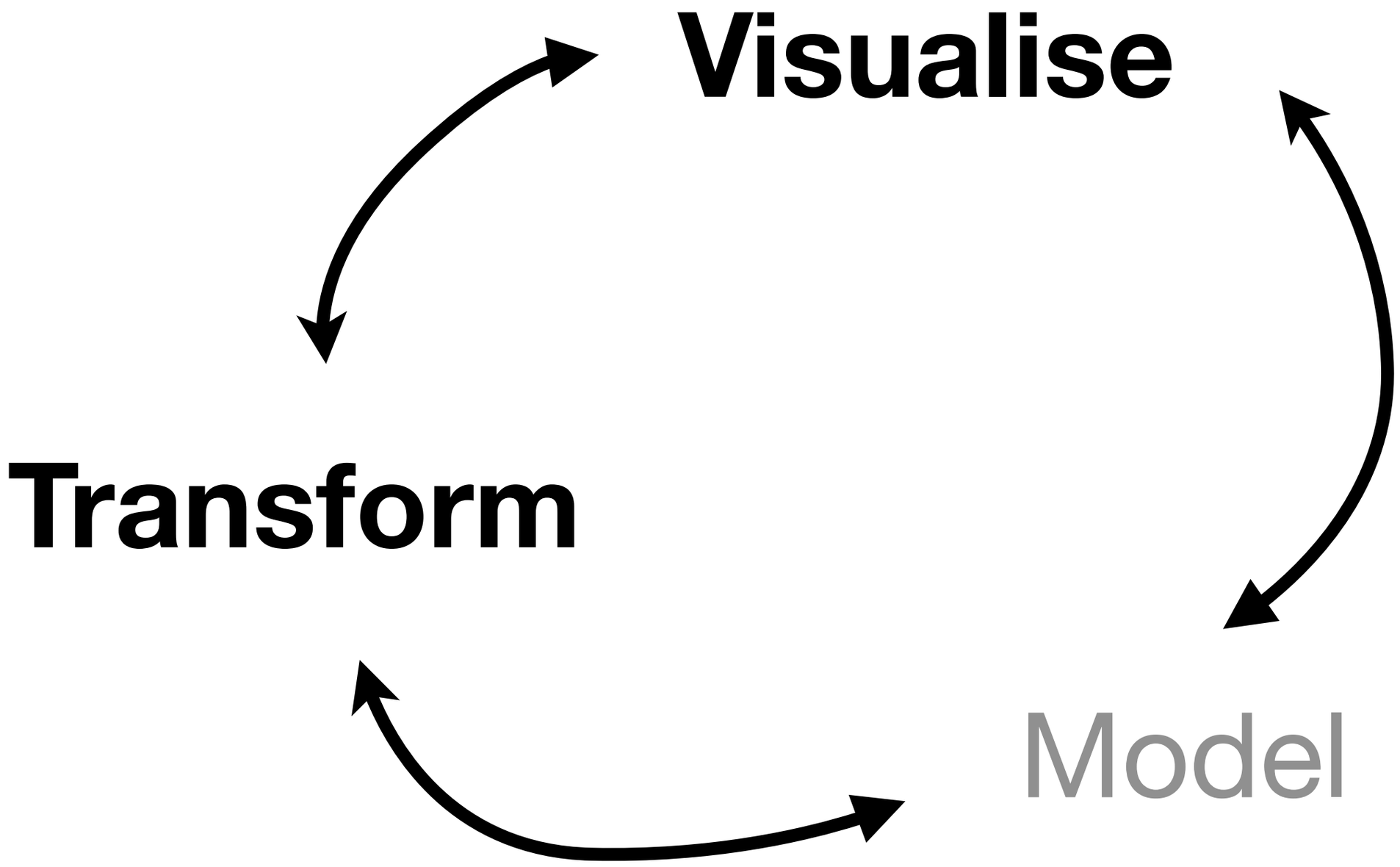


Why? Data analysis

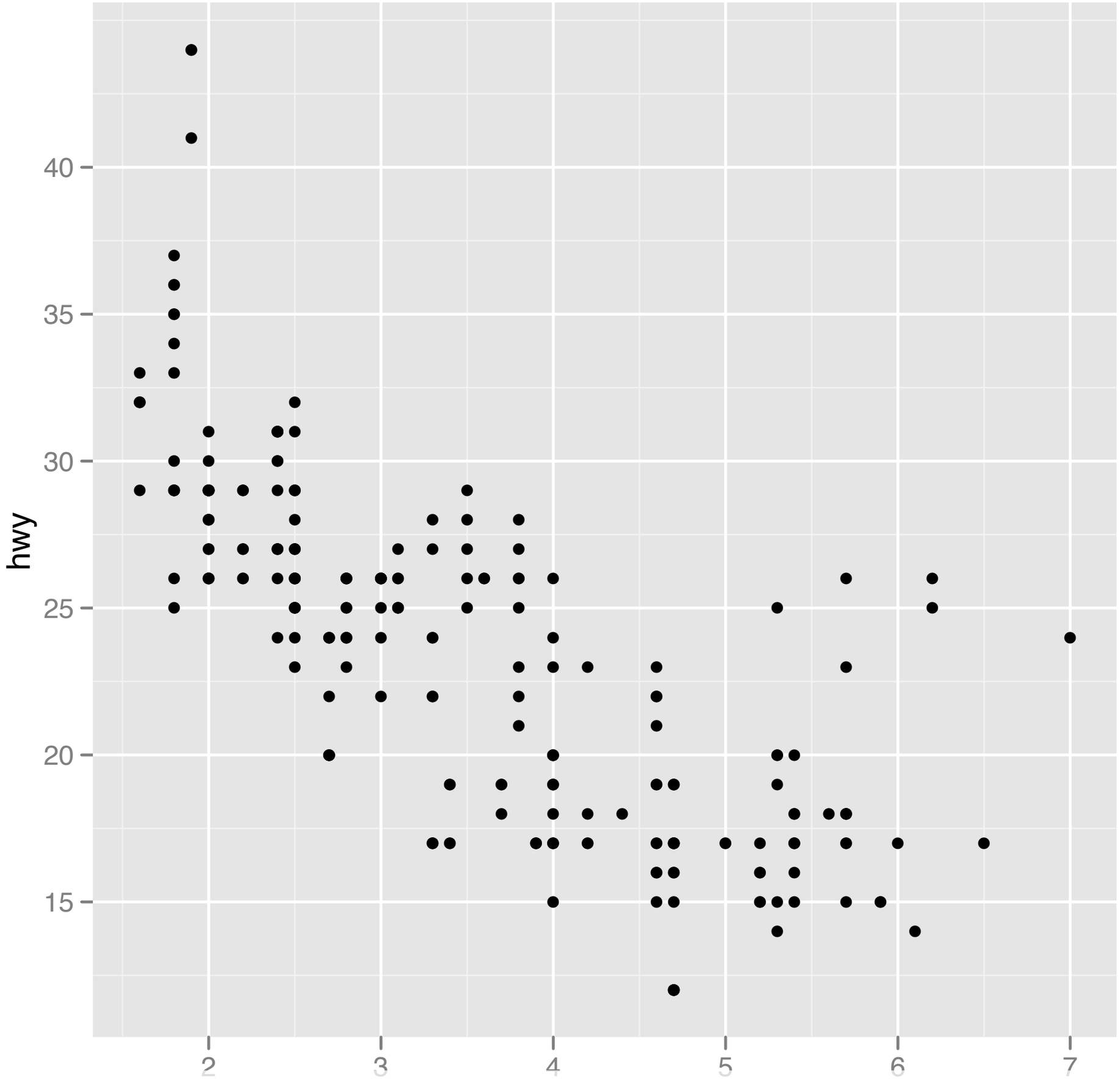


Why? Data analysis

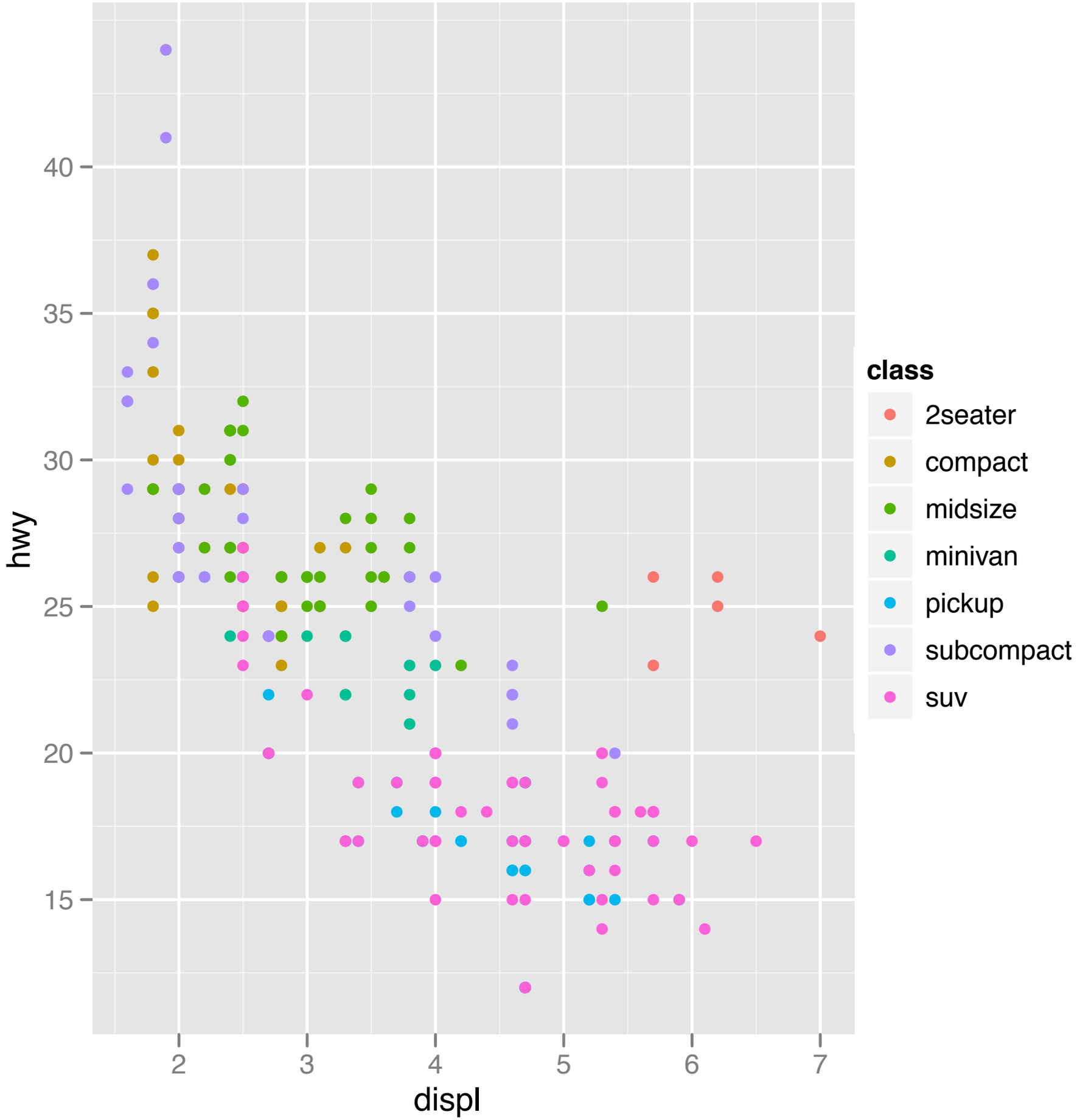


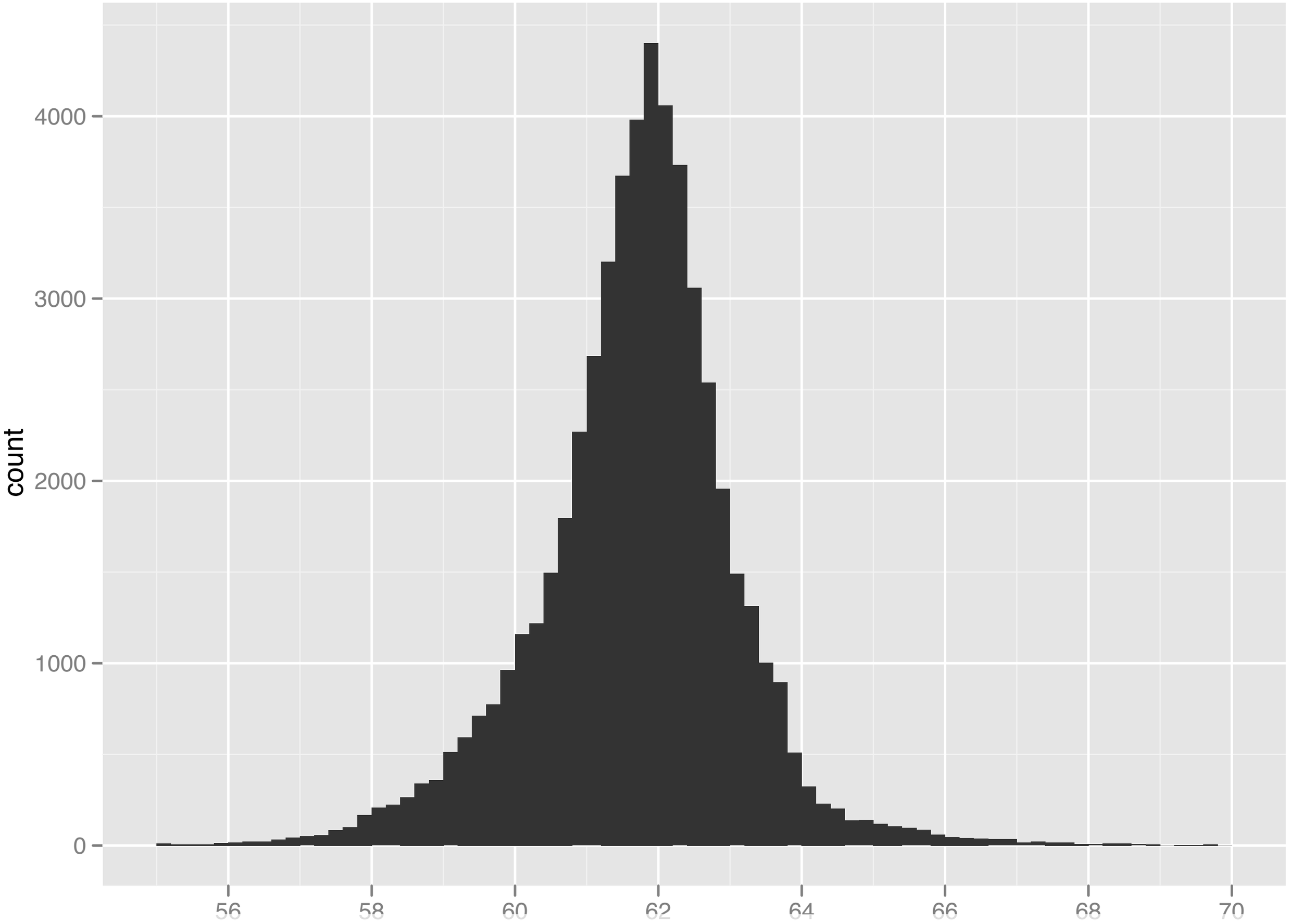

Visualise
Transform
Model

ggplot2 basics

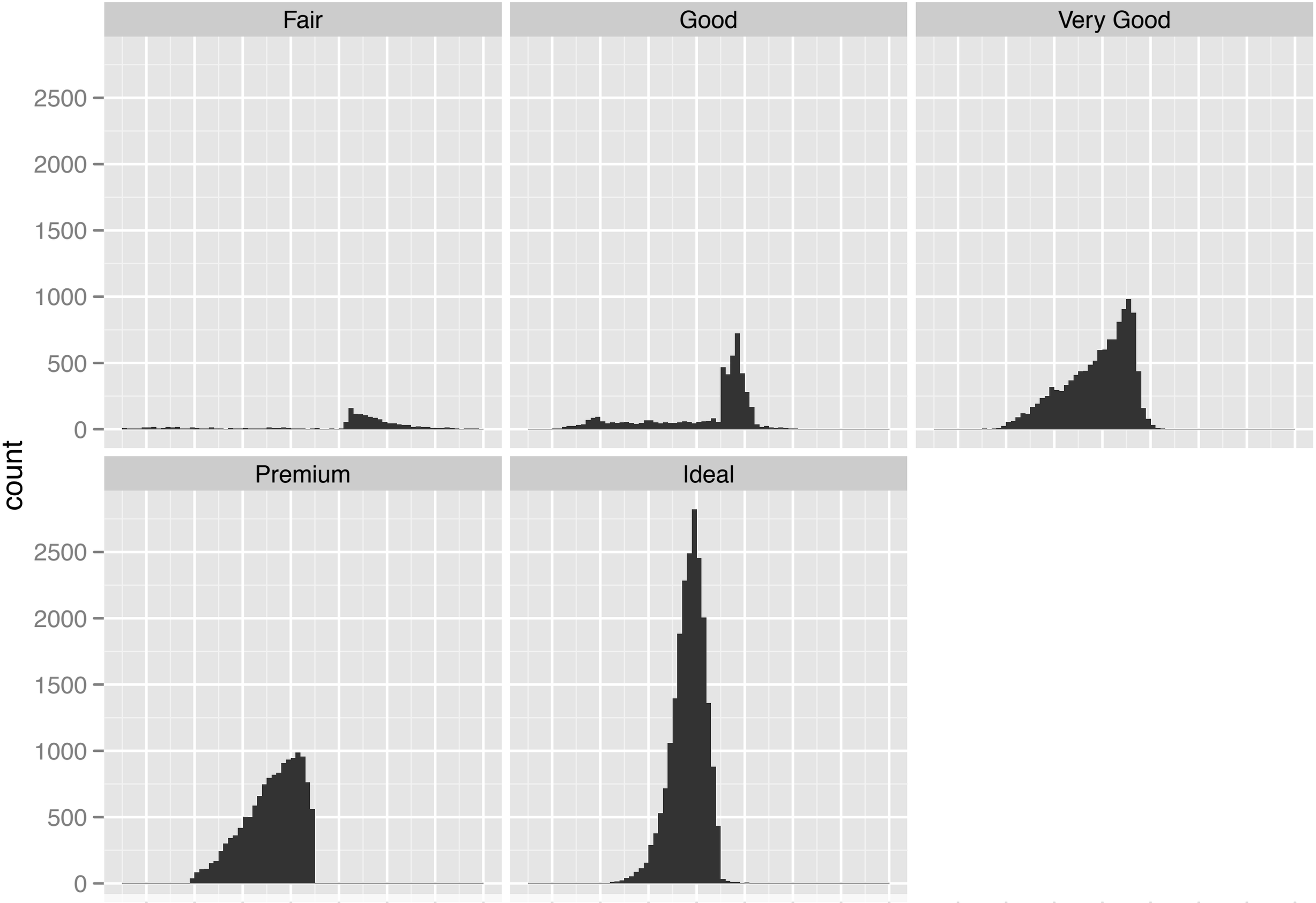



```
qplot(displ, hwy, data = mpg)
```


hwy
40
35
30
25
20
15
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv




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

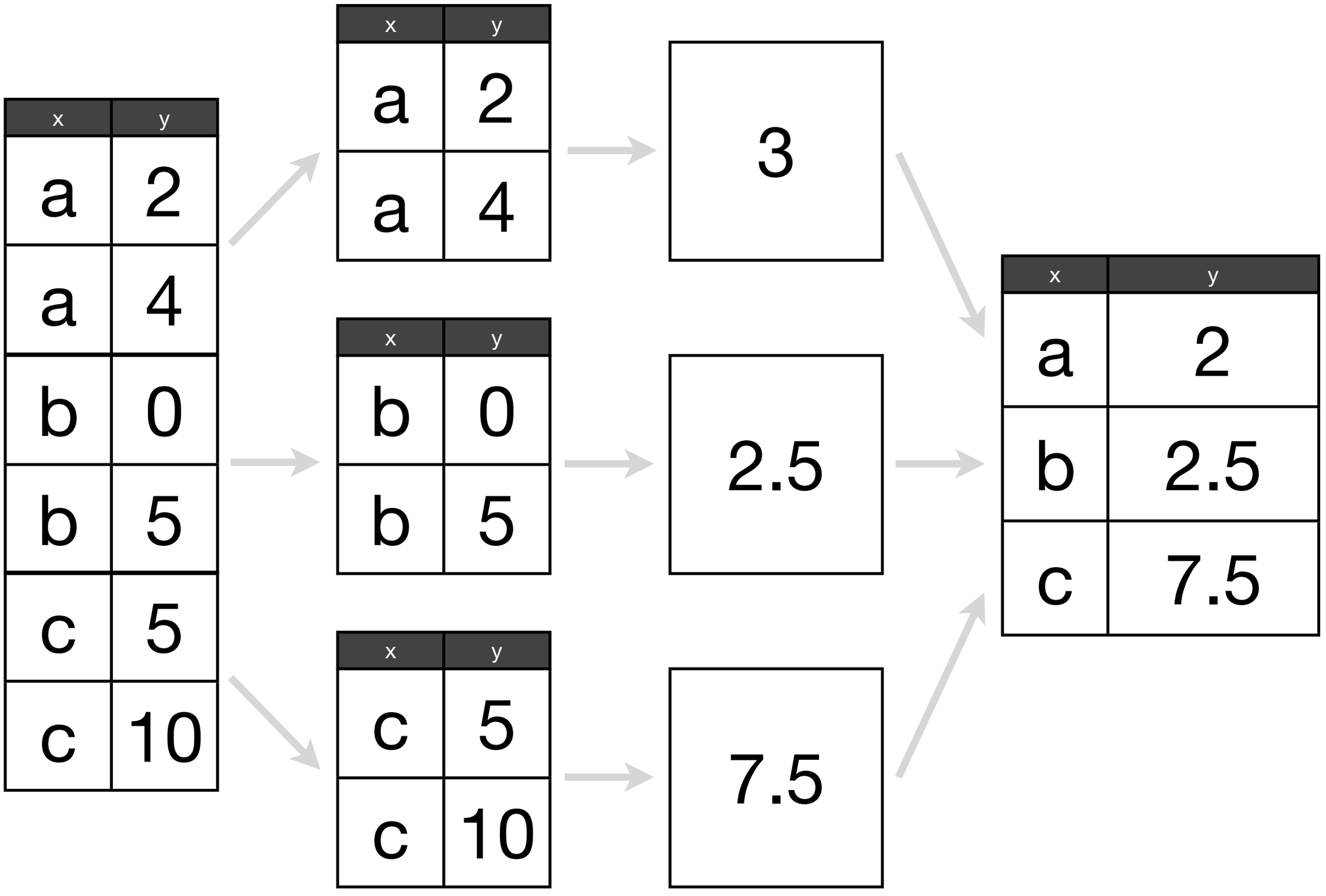


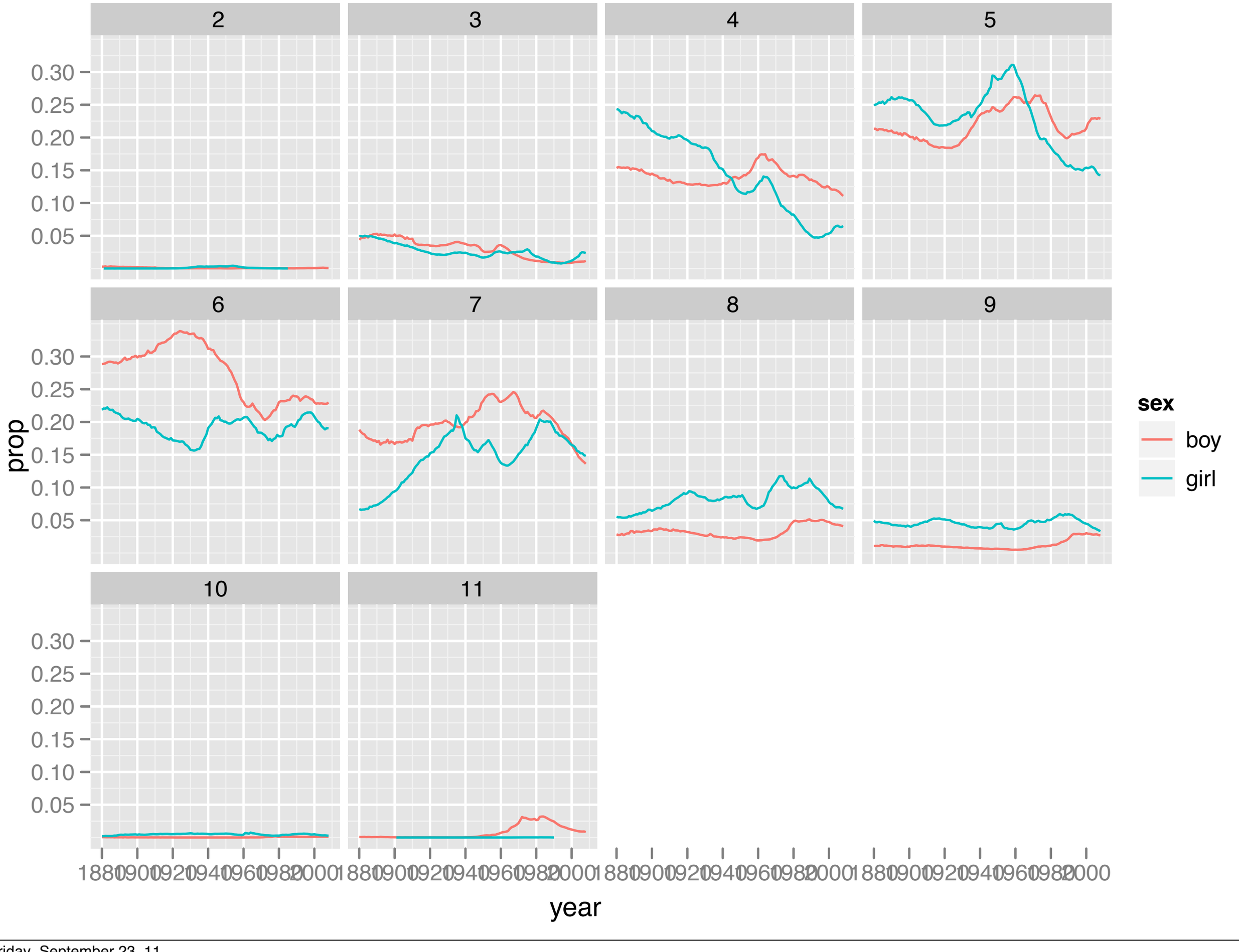

```
qplot(depth, data = diamonds, binwidth = 0.2) +
  xlim(55, 70) + facet_wrap(~cut)
```

Data manipulation

Function	Package
`subset`	`base`
`summarise`	`plyr`
`mutate`	`plyr`
`arrange`	`plyr`

Split Apply Combine

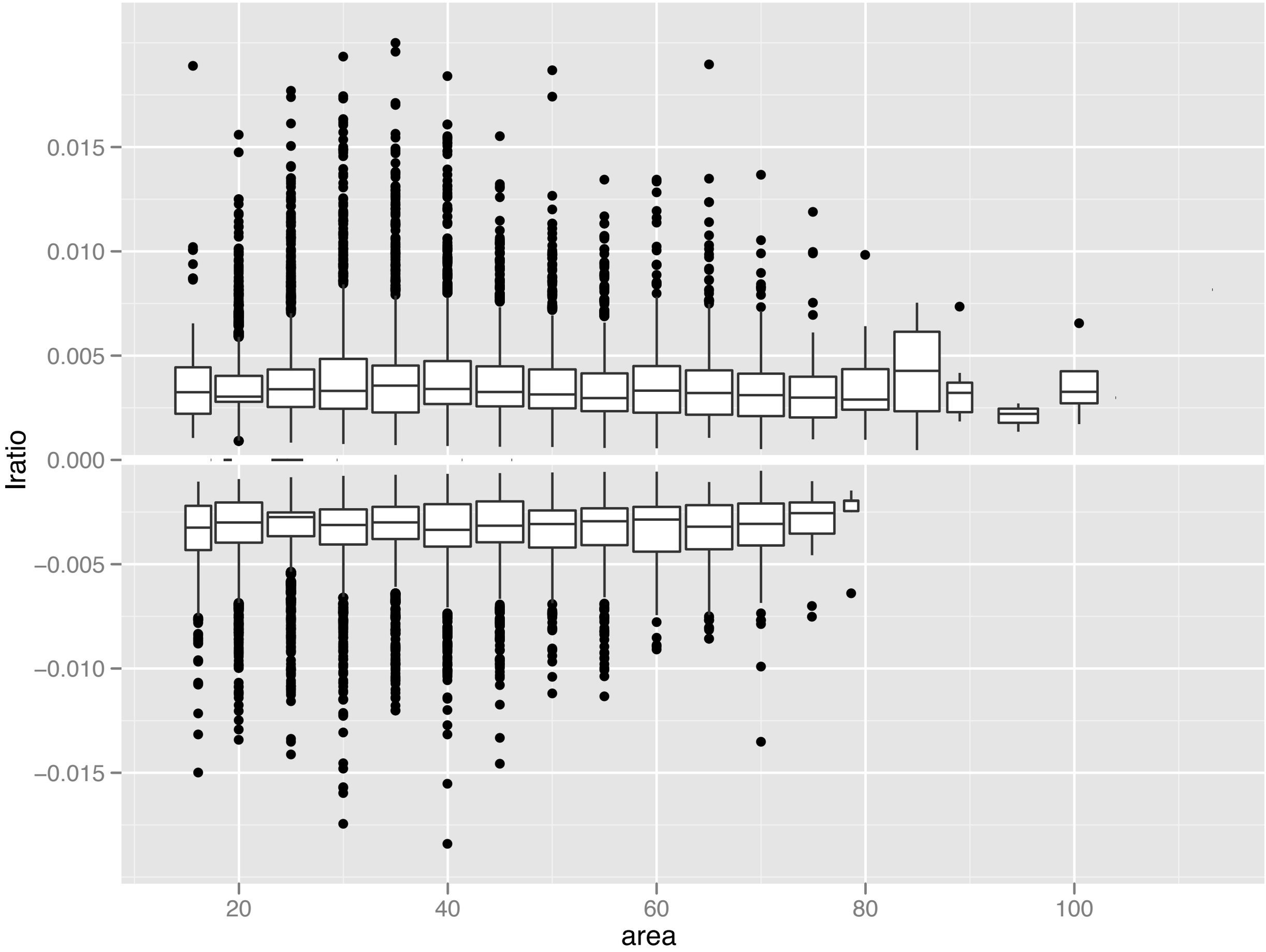



2
3
4
5
6
7
8
9
10
11
0.30
0.25
0.20
0.15
0.10
0.05
prop
1880
1900
1920
1940
1960
1980
2000
year
sex
boy
girl

Graphics: critique and creation

What should I plot?

How can I plot it?


0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015
lratio
20
40
60
80
100
area

Polishing plots for presentation

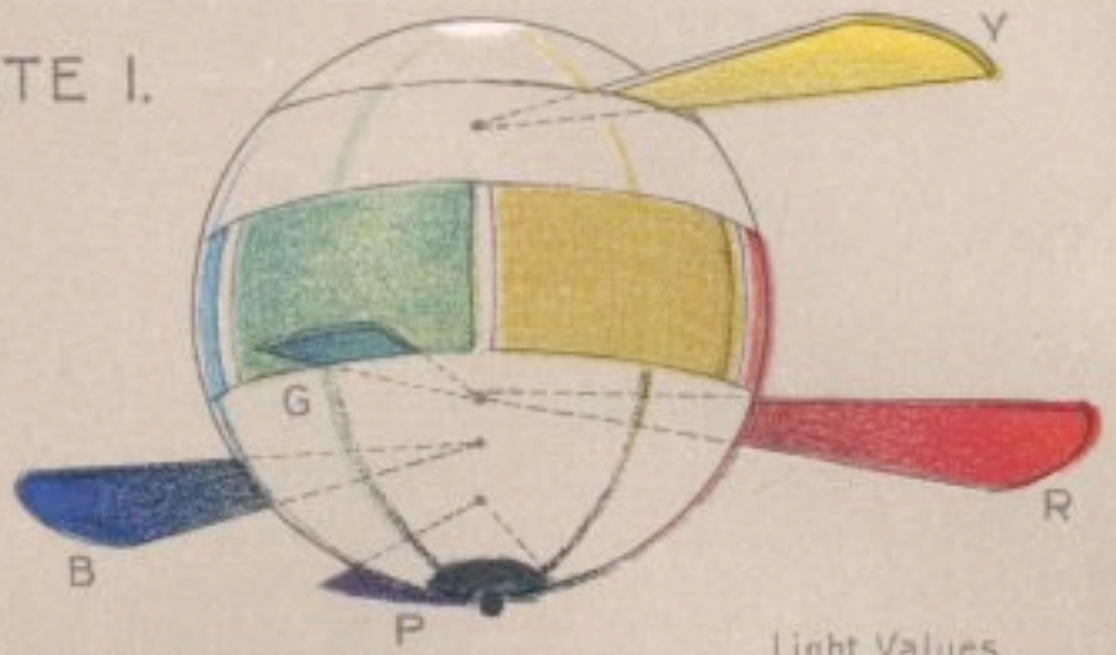
1. **Scales**: used to override default perceptual mappings, and tune parameters of axes and legends.

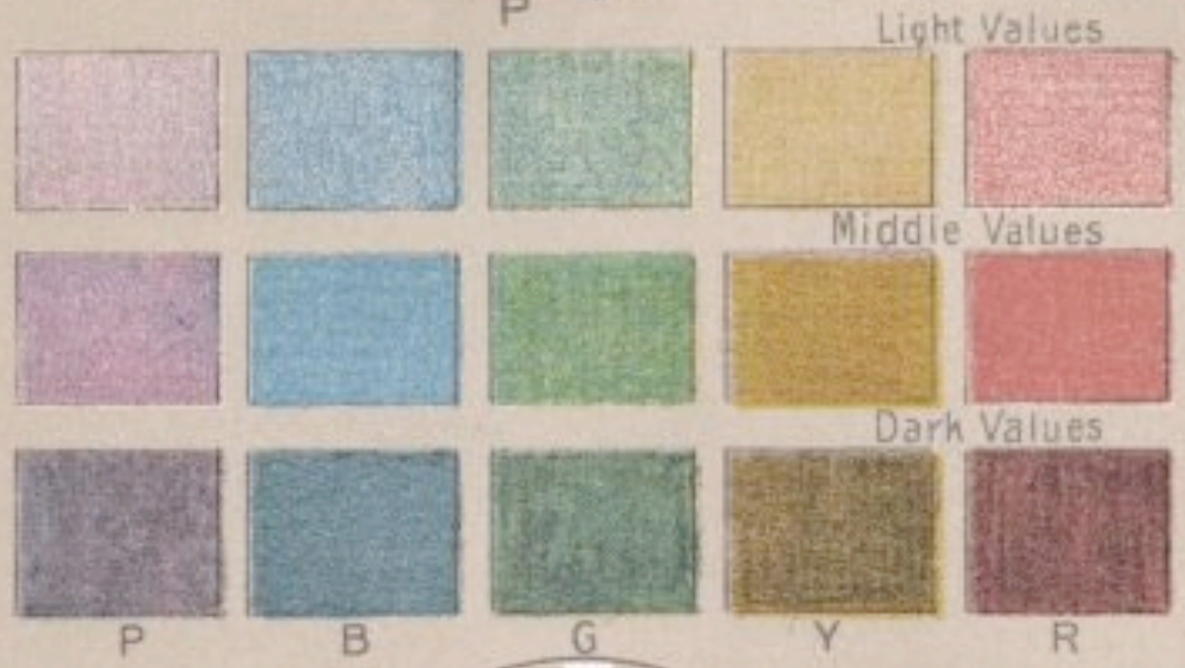
2. **Coordinate systems**: override default Cartesian coordinate system

3. **Themes**: control presentation of non-data elements.

4. **Saving your work**: to include in reports, presentations, etc.

PLATE I.






Learning a new
language is hard!

ggplot2 basics

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

September 2011

1. Diving in: scatterplots & aesthetics
2. Facetting
3. Geoms
4. Histograms and barcharts
5. Scatterplots for large data

Diving in

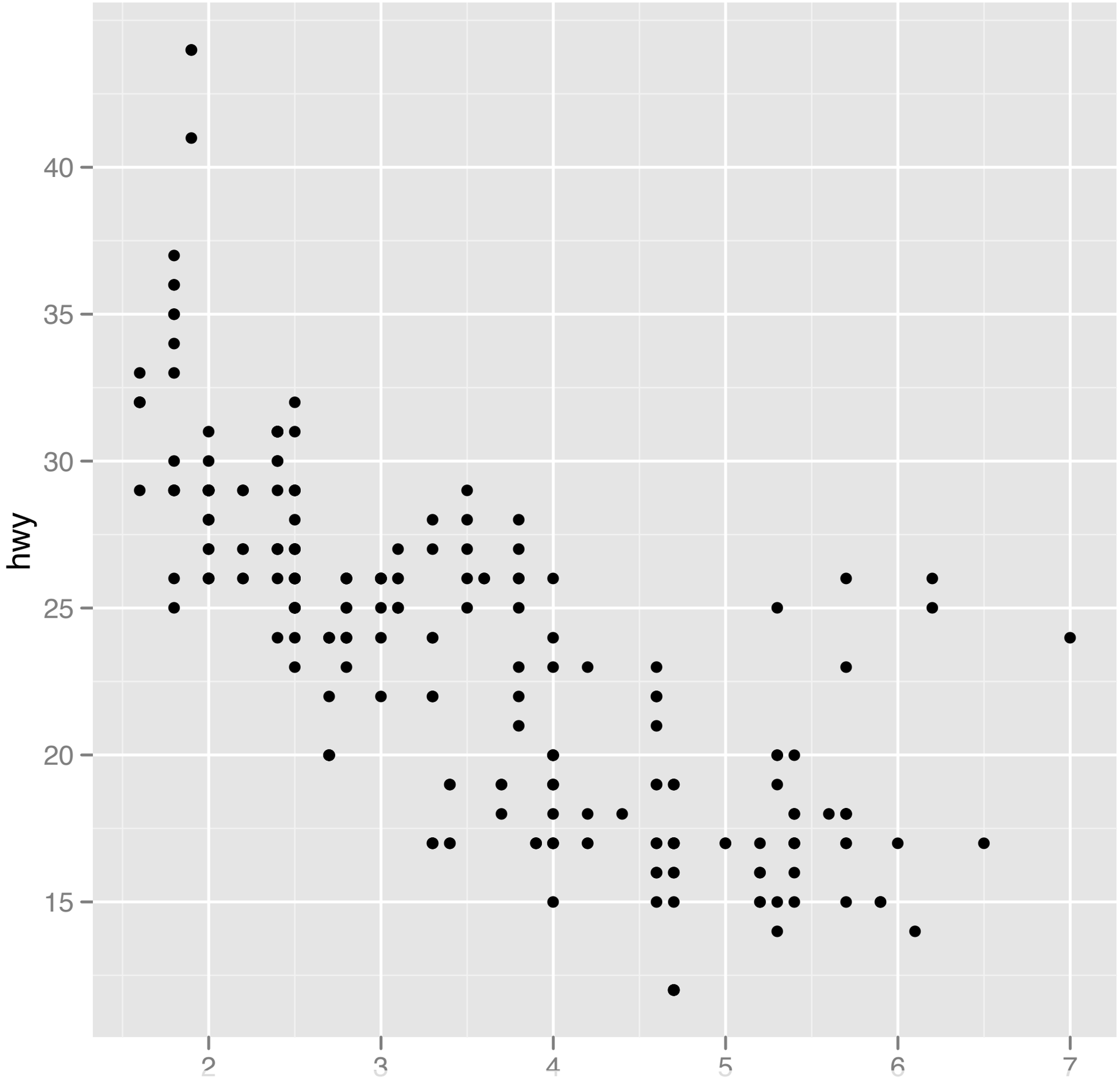
Scatterplot basics

```
install.packages("ggplot2")
library(ggplot2)

?mpg
head(mpg)
str(mpg)
summary(mpg)

qplot(displ, hwy, data = mpg)
```

Scatterplot basics

```
install.packages("ggplot2")
library(ggplot2)

?mpg
head(mpg)
str(mpg)
summary(mpg)

qplot(displ, hwy, data = mpg)
```

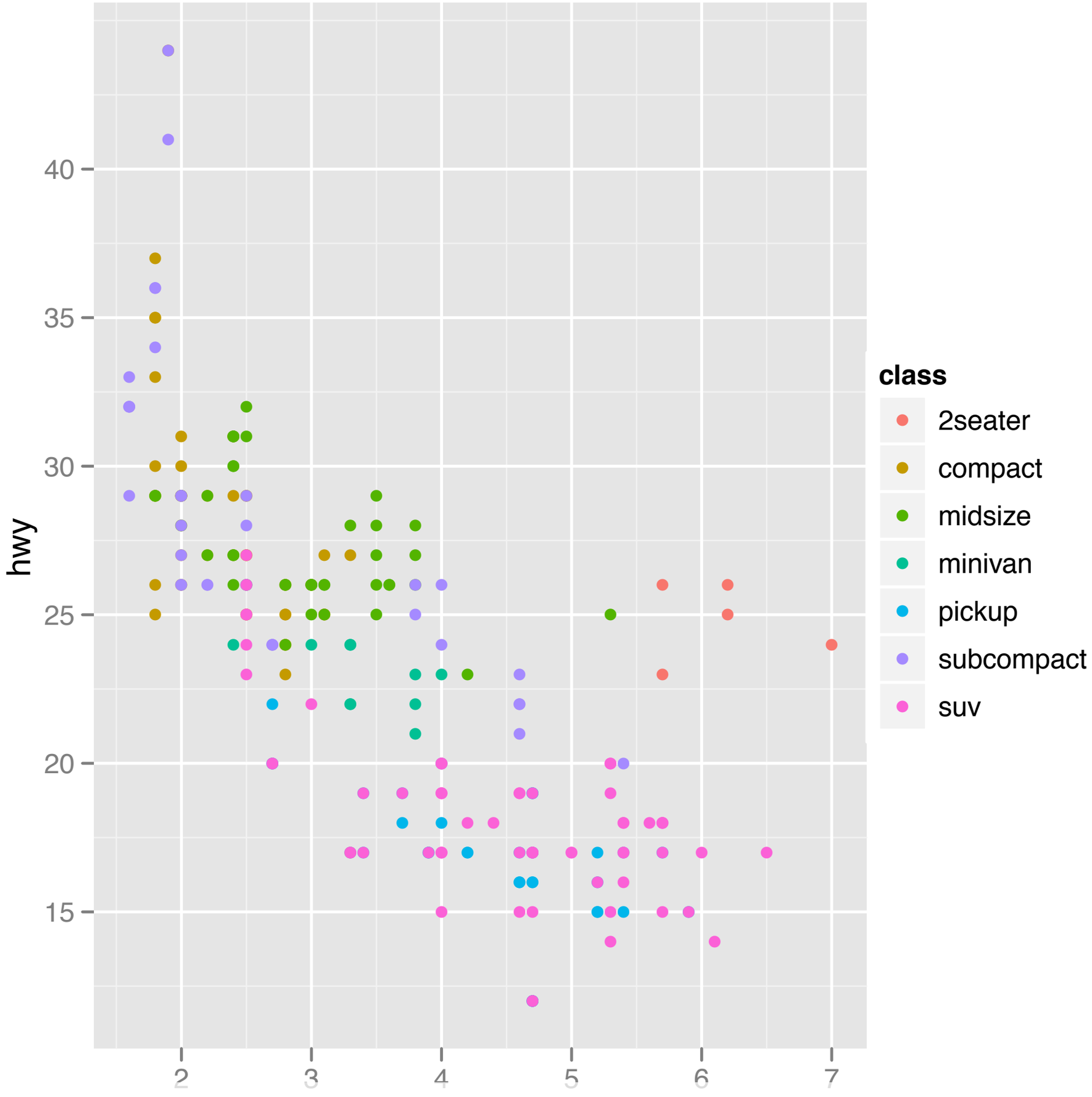
Always explicitly specify the data



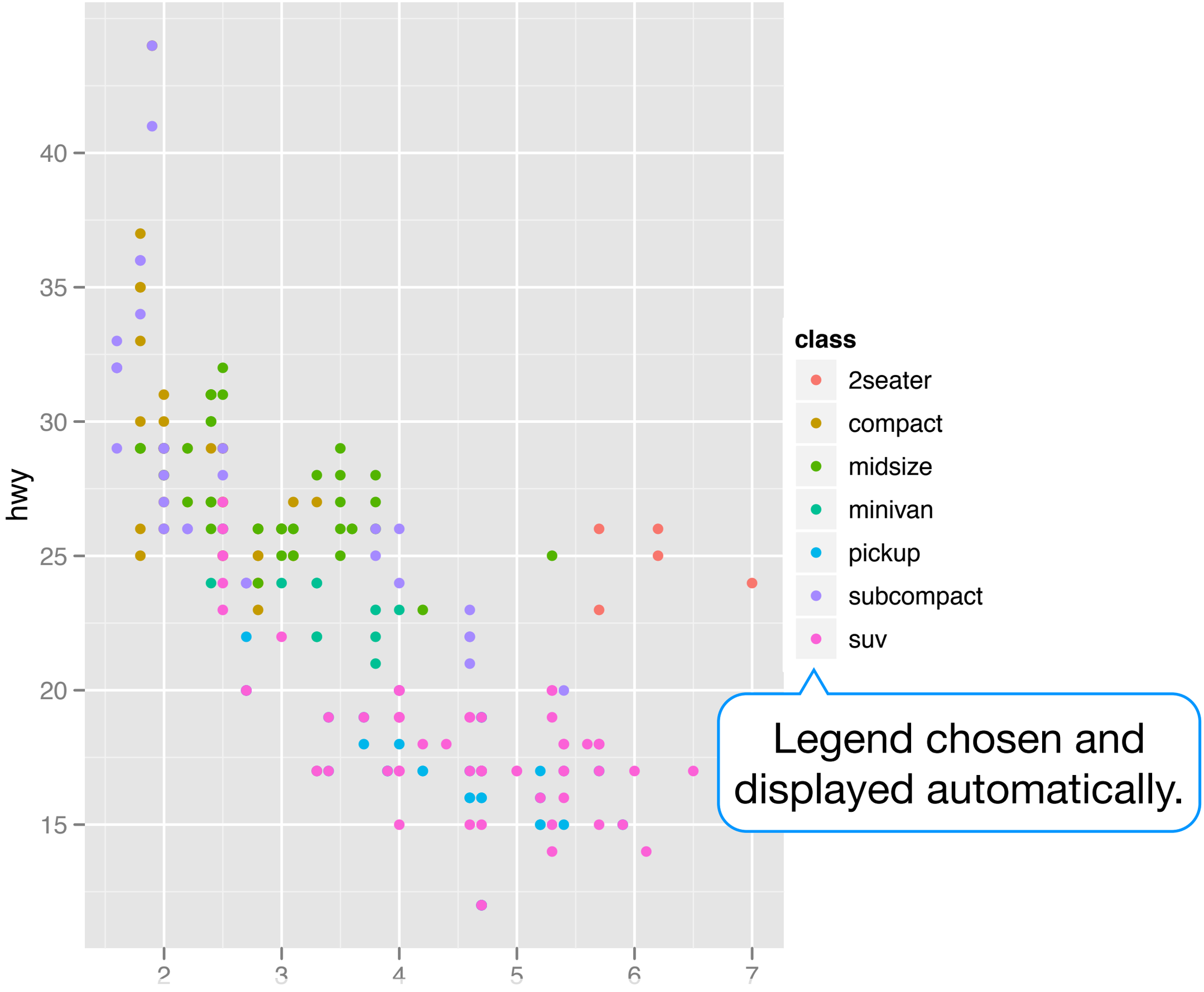

```
qplot(displ, hwy, data = mpg)
```

Additional variables

Can display additional variables with **aesthetics** (like shape, colour, size) or **facetting** (small multiples displaying different subsets)




```
qplot(displ, hwy, colour = class, data = mpg)
```




```
qplot(displ, hwy, colour = class, data = mpg)
```

Your turn

Experiment with colour, size, and shape aesthetics.

What's the difference between discrete or continuous variables?

What happens when you combine multiple aesthetics?

	Discrete	Continuous
Colour	Rainbow of colours	Gradient from red to blue
Size	Discrete size steps	Linear mapping between radius and value
Shape	Different shape for each	Shouldn’t work

Facetting

Faceting

Small multiples displaying different subsets of the data.

Useful for exploring conditional relationships. Useful for large data.

Your turn

```
qplot(displ, hwy, data = mpg) +
facet_grid(. ~ cyl)

qplot(displ, hwy, data = mpg) +
facet_grid(drv ~ .)

qplot(displ, hwy, data = mpg) +
facet_grid(drv ~ cyl)

qplot(displ, hwy, data = mpg) +
facet_wrap(~ class)
```

Summary

`facet_grid()`: 2d grid, rows ~ cols, . for no split

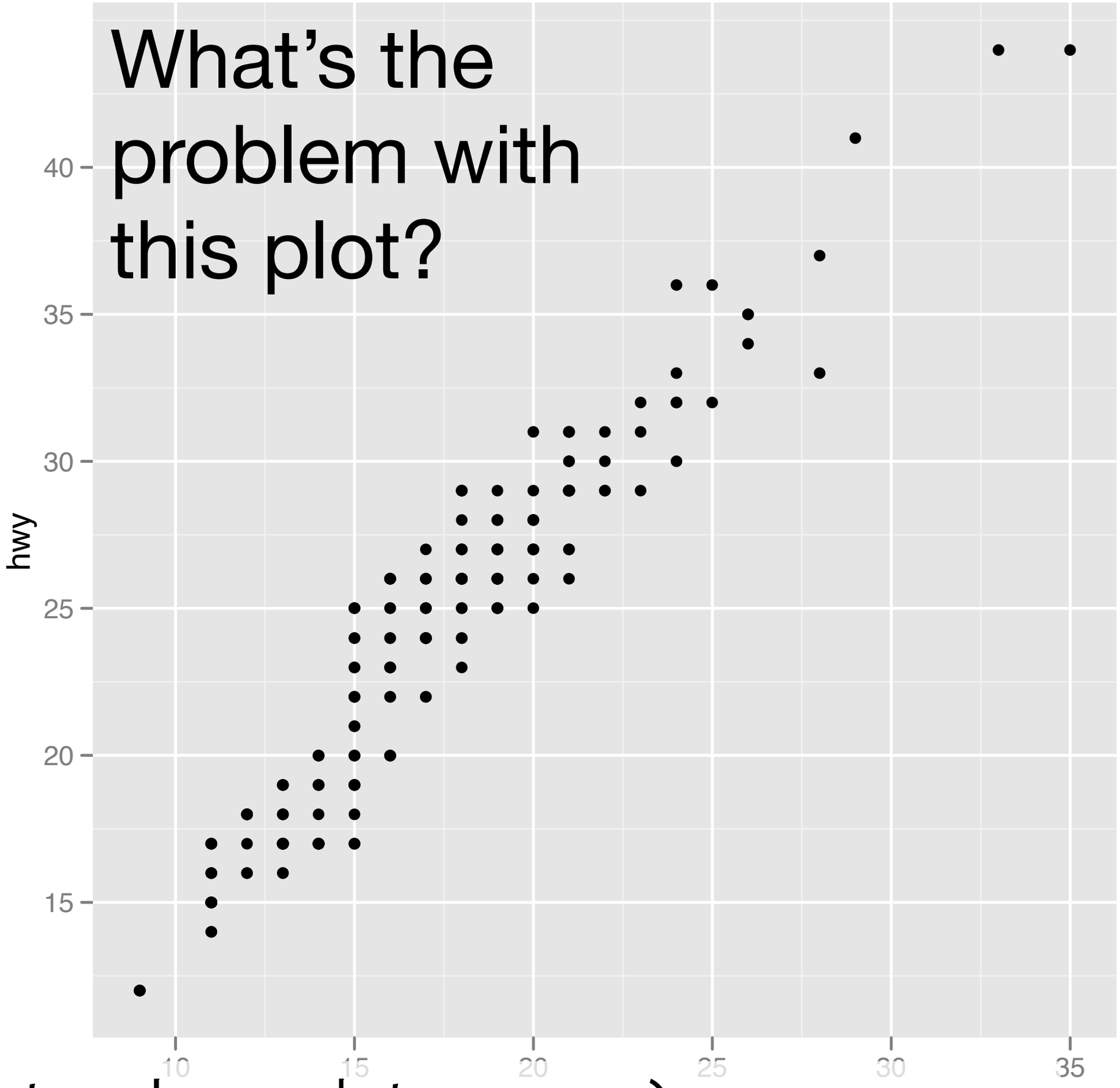
`facet_wrap()`: 1d ribbon wrapped into 2d

Aside: workflow

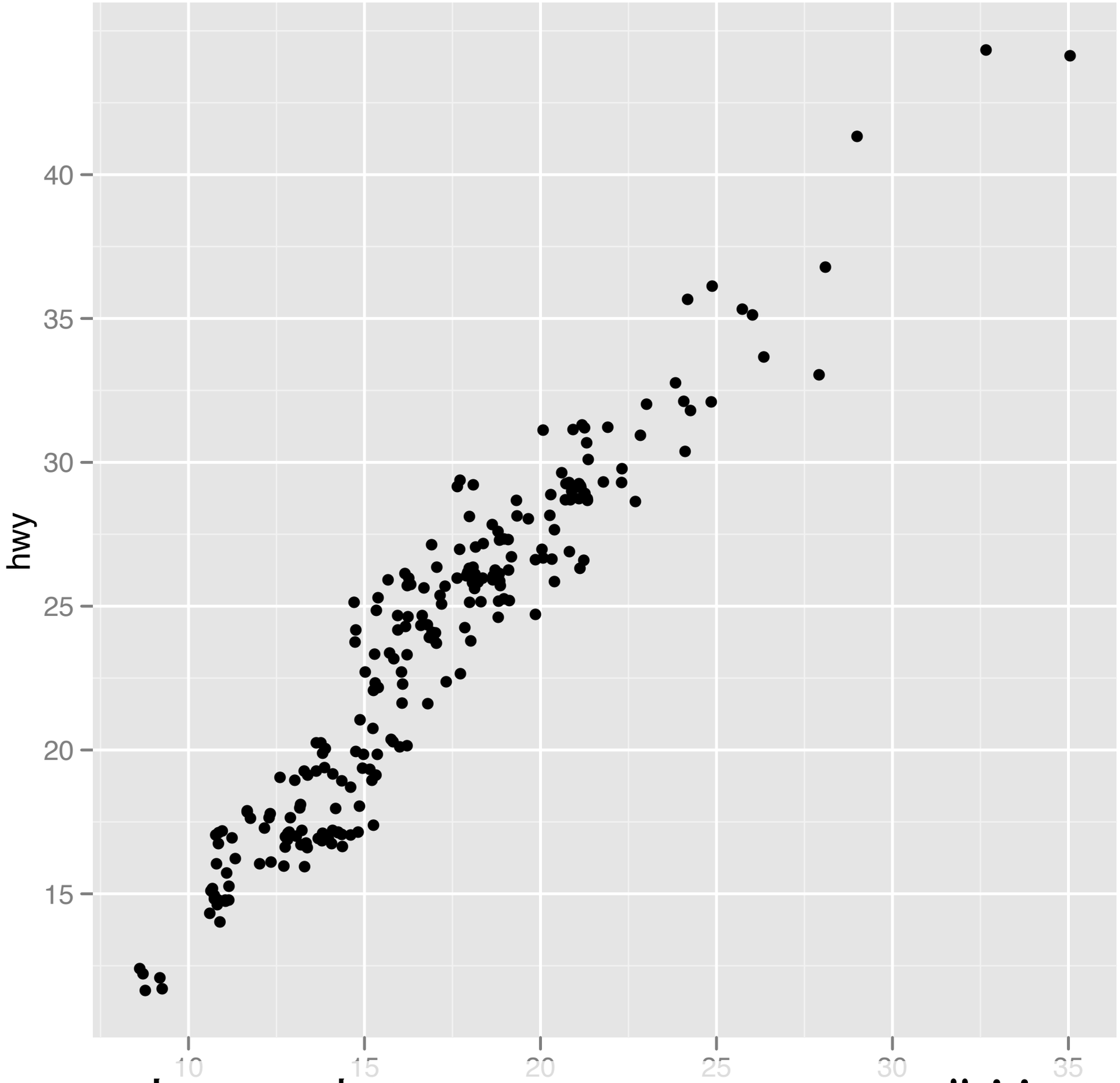
Keep a copy of the slides open so that you can copy and paste the code.

Geoms

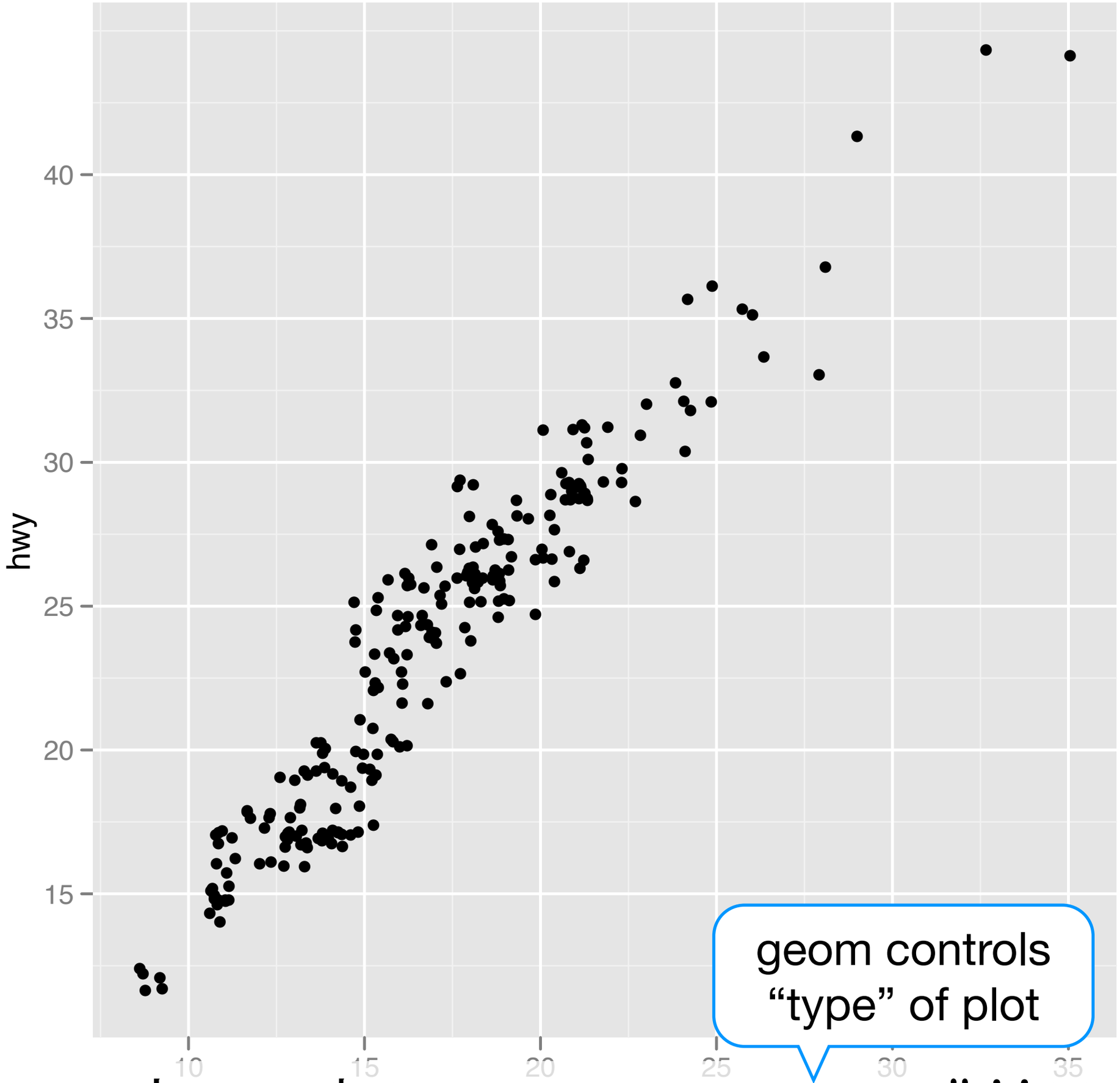
What's the problem with this plot?



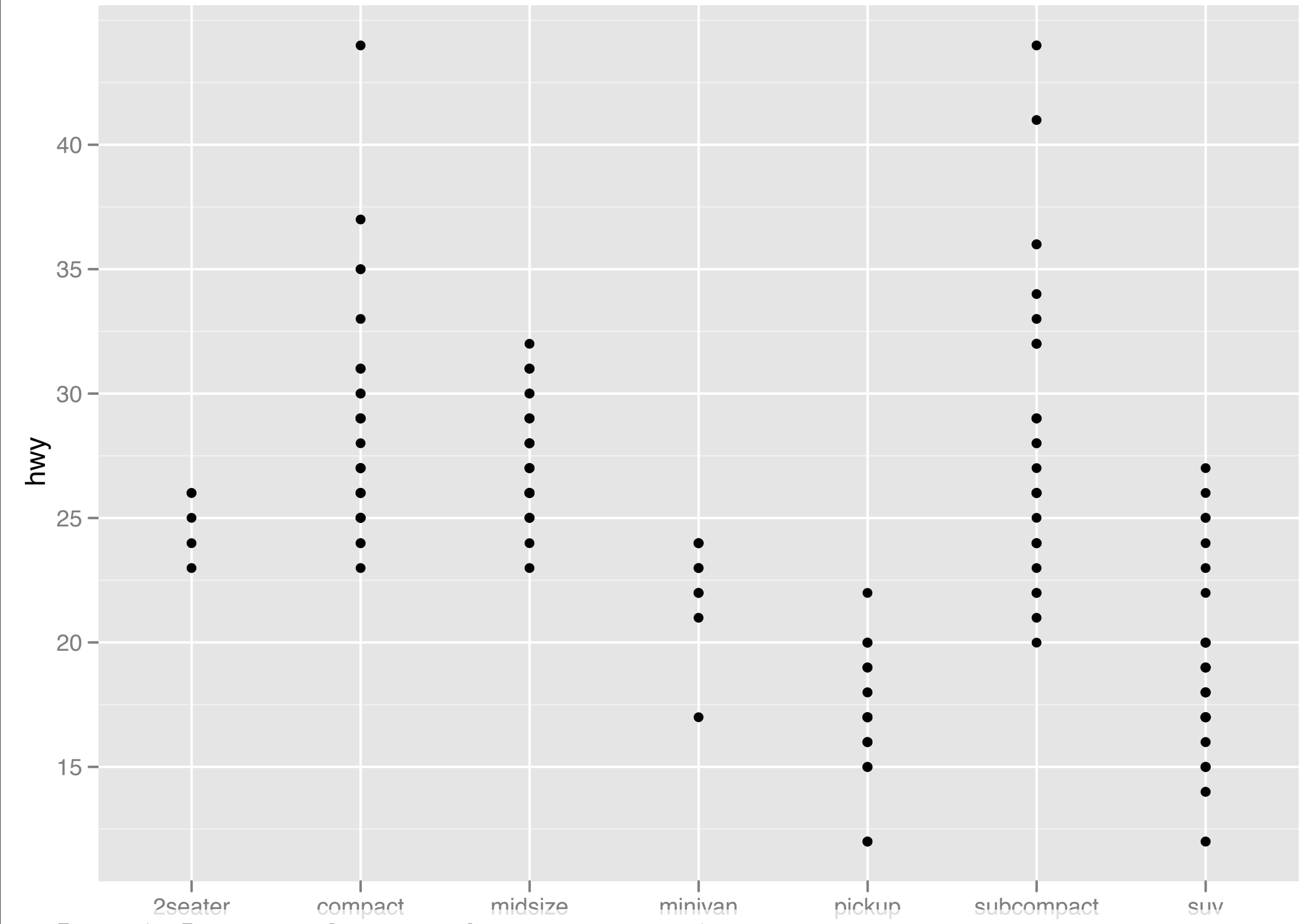

```
qplot(cty, hwy, data = mpg)
```




```
qplot(cty, hwy, data = mpg, geom = "jitter")
```



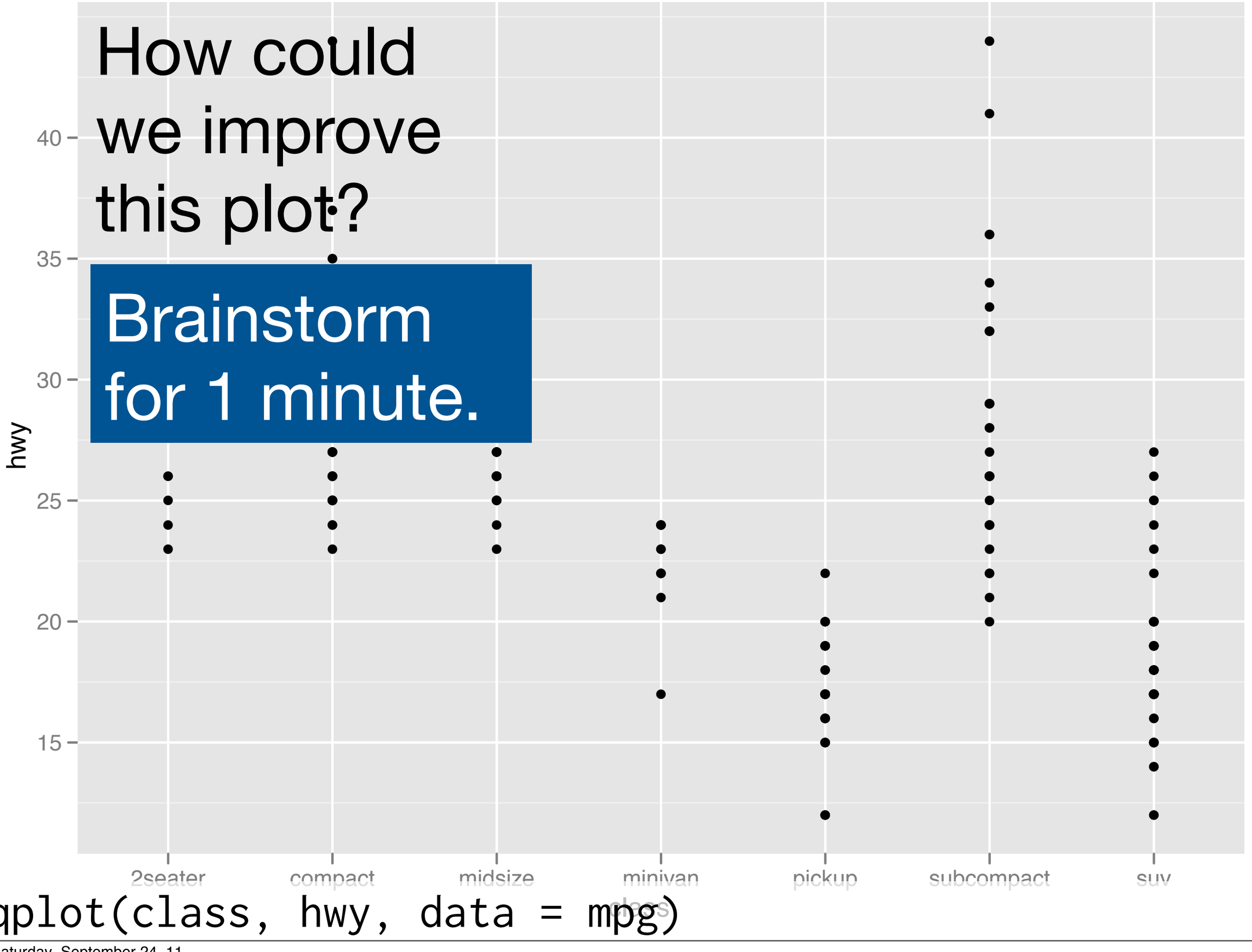

```
qplot(cty, hwy, data = mpg, geom = "jitter")
```



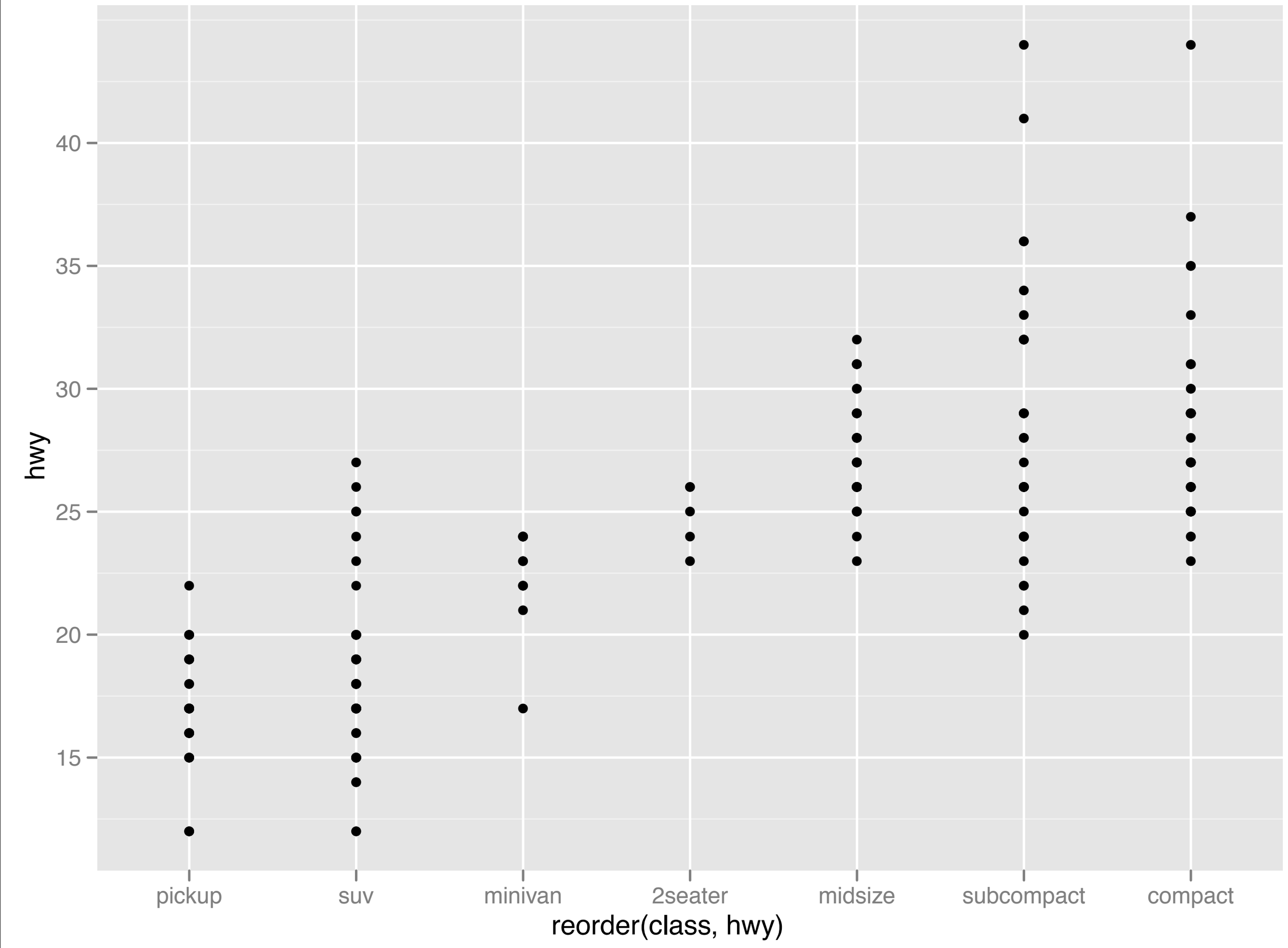

```
qplot(class, hwy, data = mpg)
```

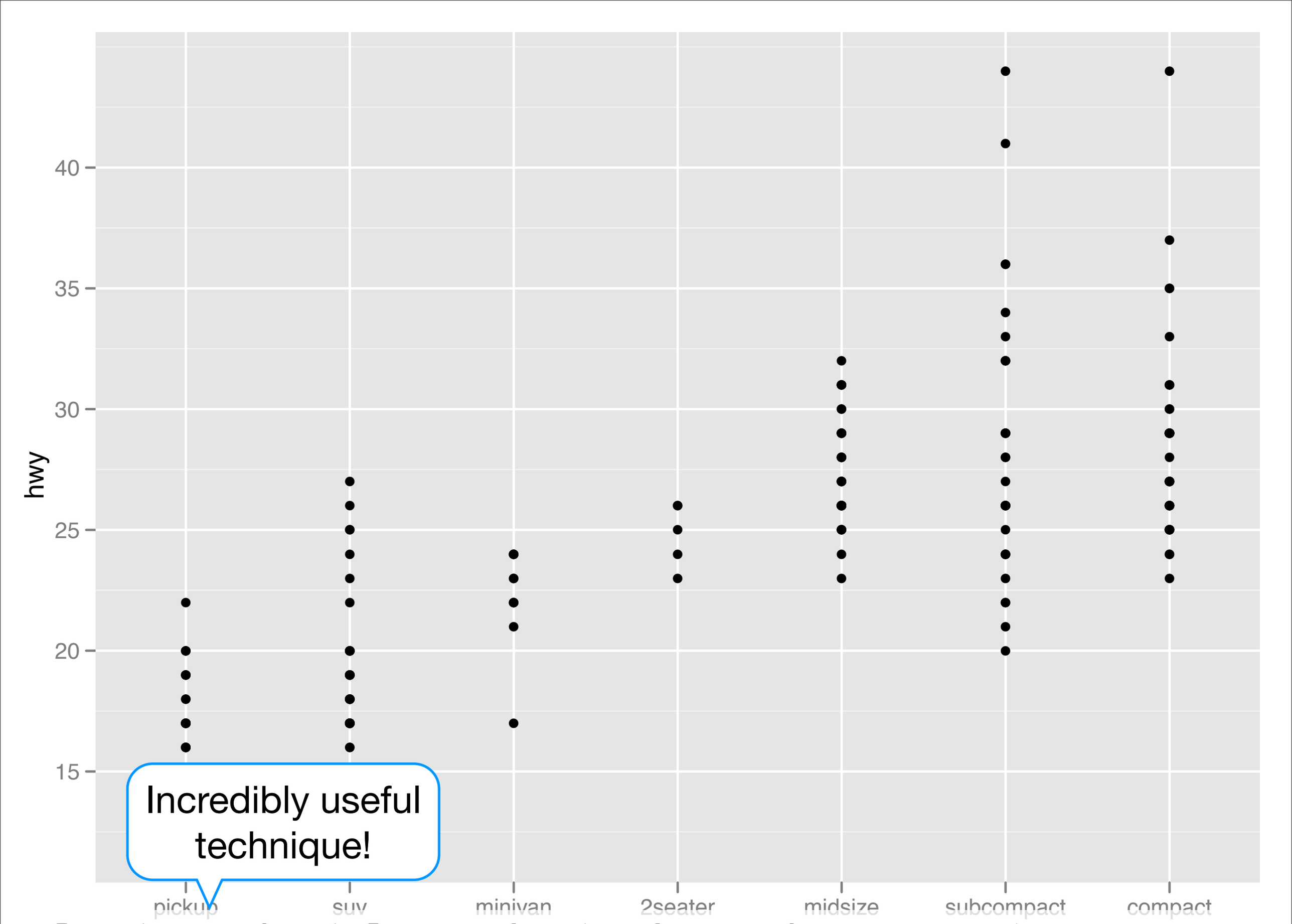
How could we improve this plot?

Brainstorm for 1 minute.

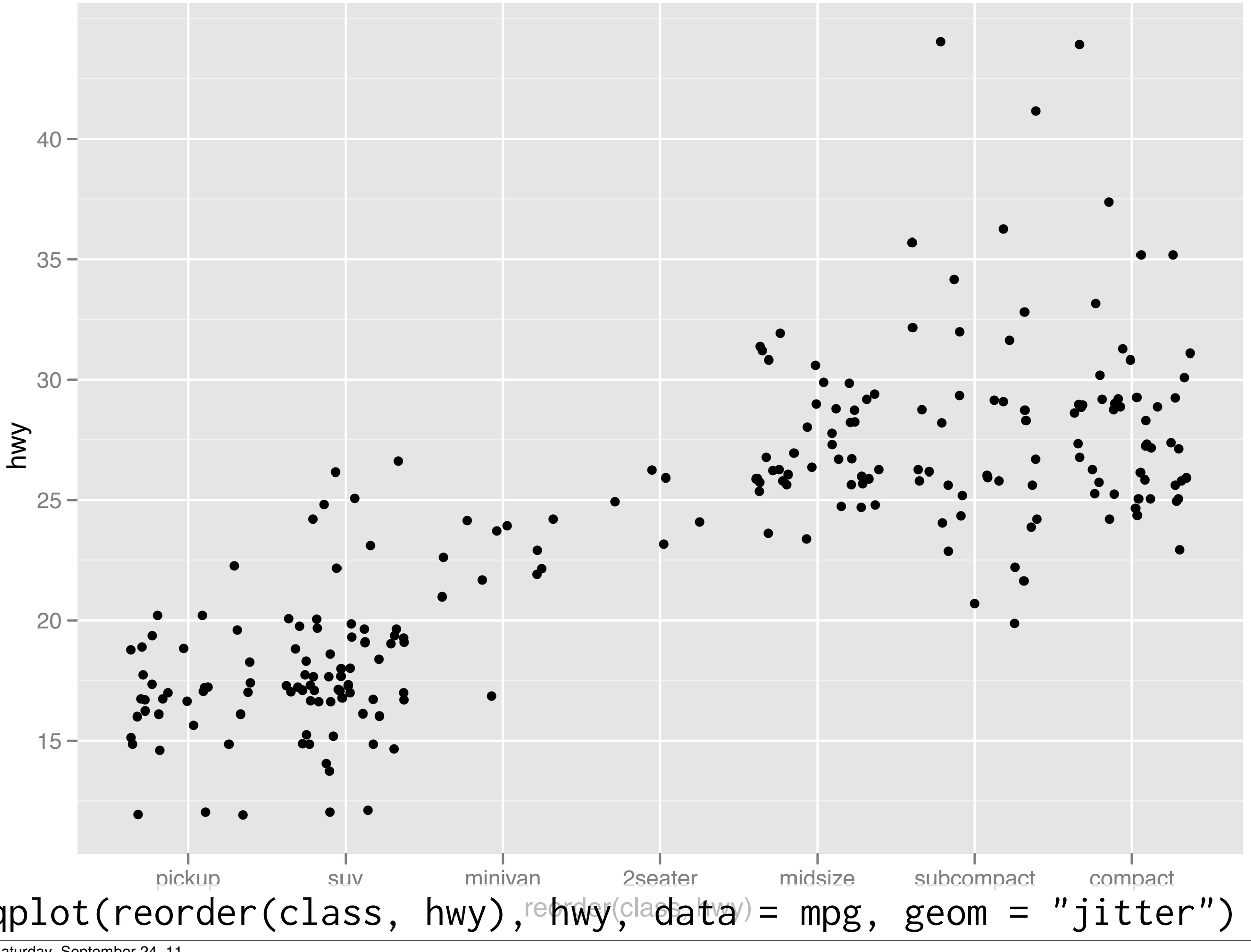



```
qplot(class, hwy, data = mpg)
```

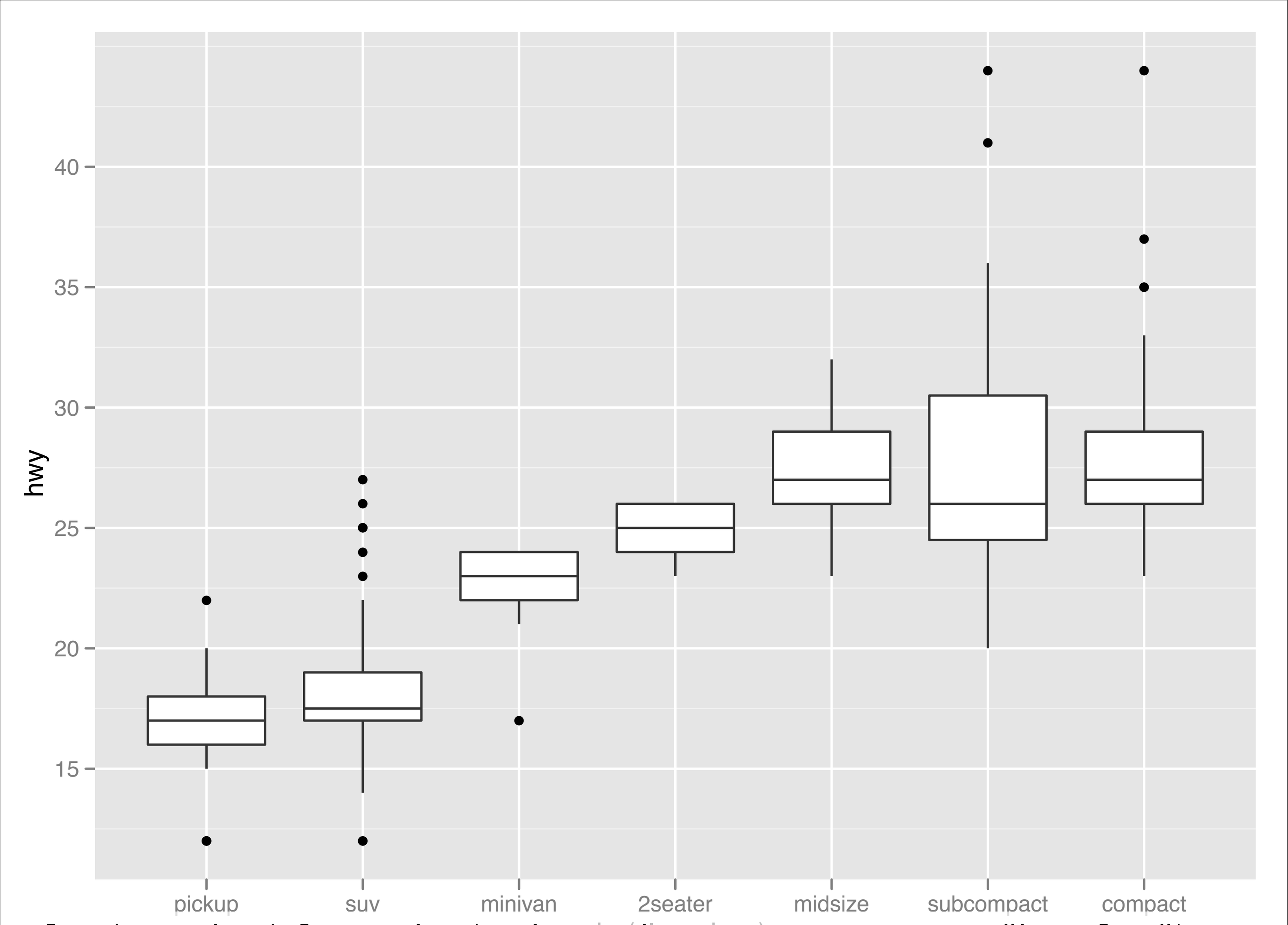

hwy
40
35
30
25
20
15
pickup
suv
minivan
2seater
midsize
subcompact
compact
reorder(class, hwy)



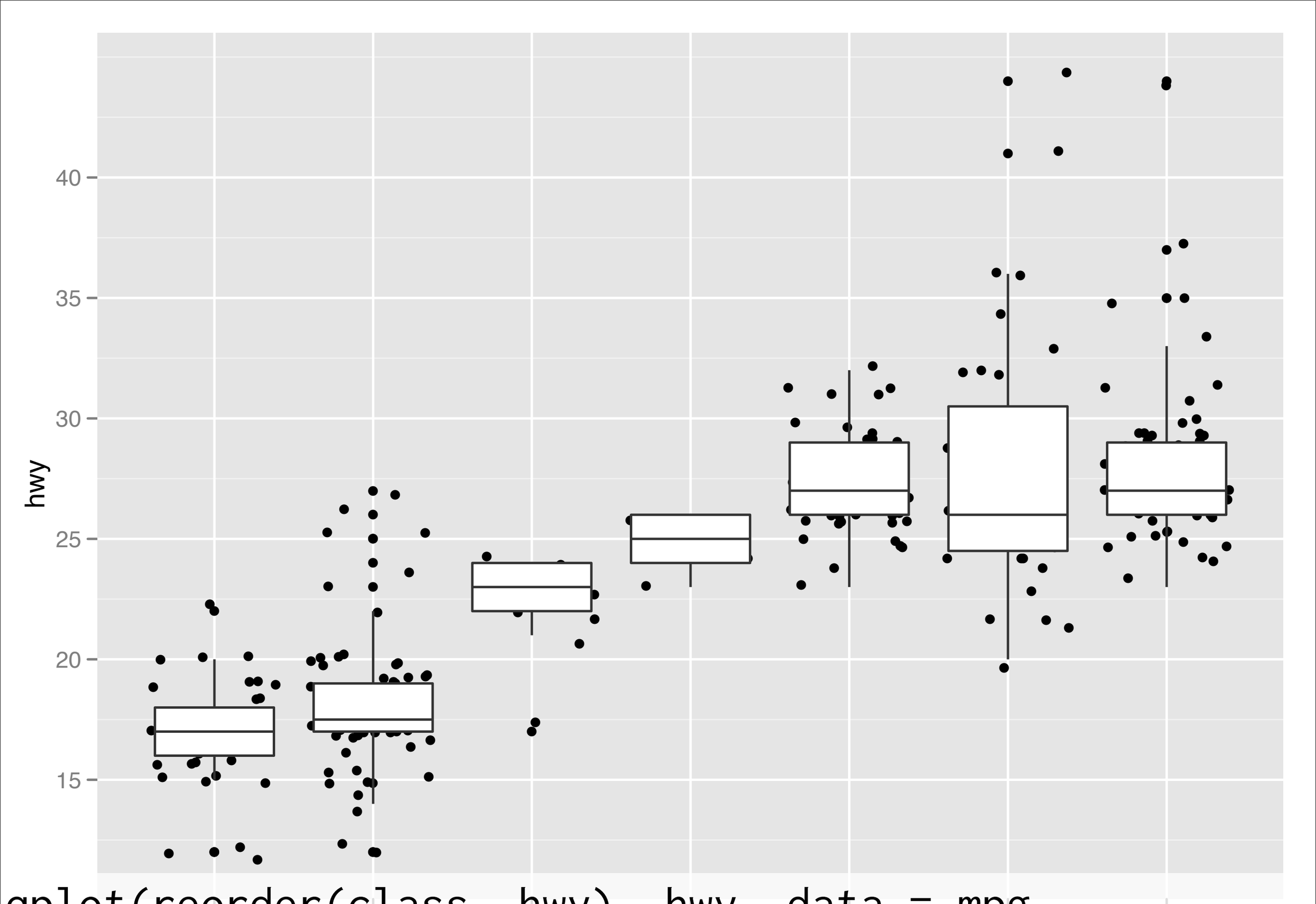

```
qplot(reorder(class, hwy), hwy, data = mpg)
```




```
qplot(reorder(class, hwy), hwy, data = mpg, geom = "jitter")
```




```
qplot(reorder(class, hwy), hwy, data = mpg, geom = "boxplot")
```




```
qplot(reorder(class, hwy), hwy, data = mpg,
  geom = c("jitter", "boxplot"))
```

Your turn

Read the help for reorder. Redraw the previous plots with class ordered by median hwy.

How would you put the jittered points on top of the boxplots?

Diamonds

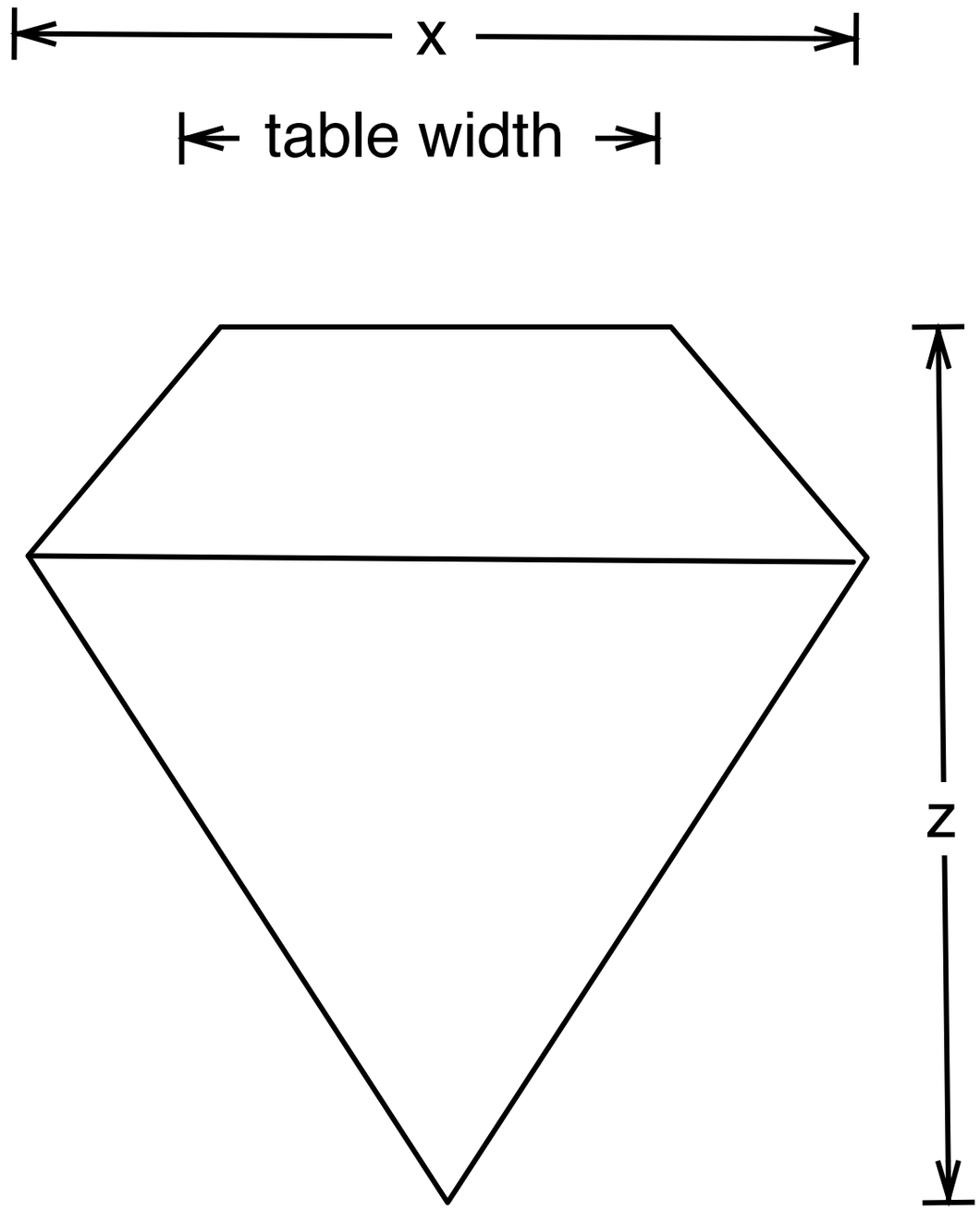
Diamonds data

~**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)
resolution(diamonds$carat)

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

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                         rat)

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

Common ggplot2 technique: adding together plot components

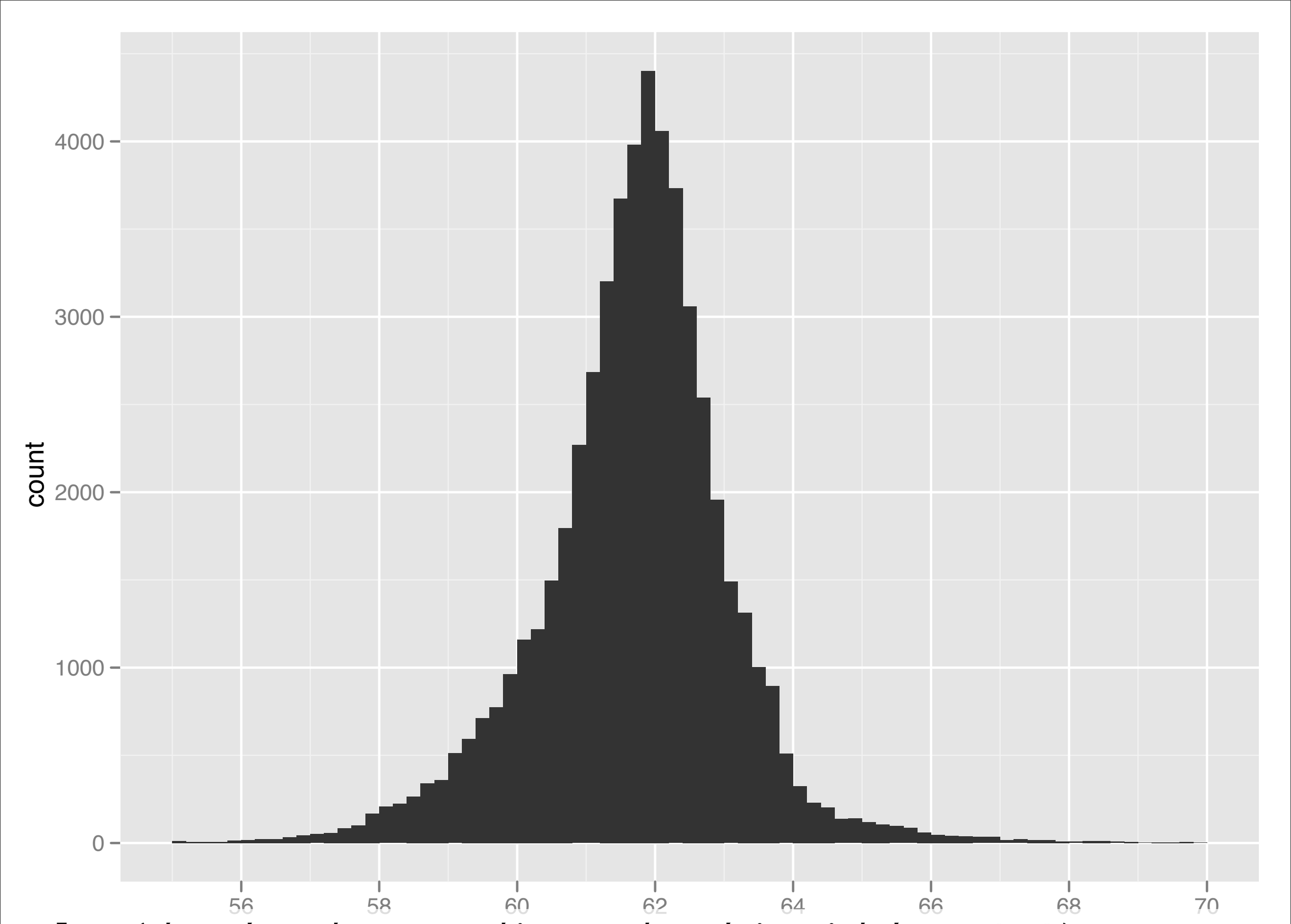
Always experiment with the bin width!

```
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
```

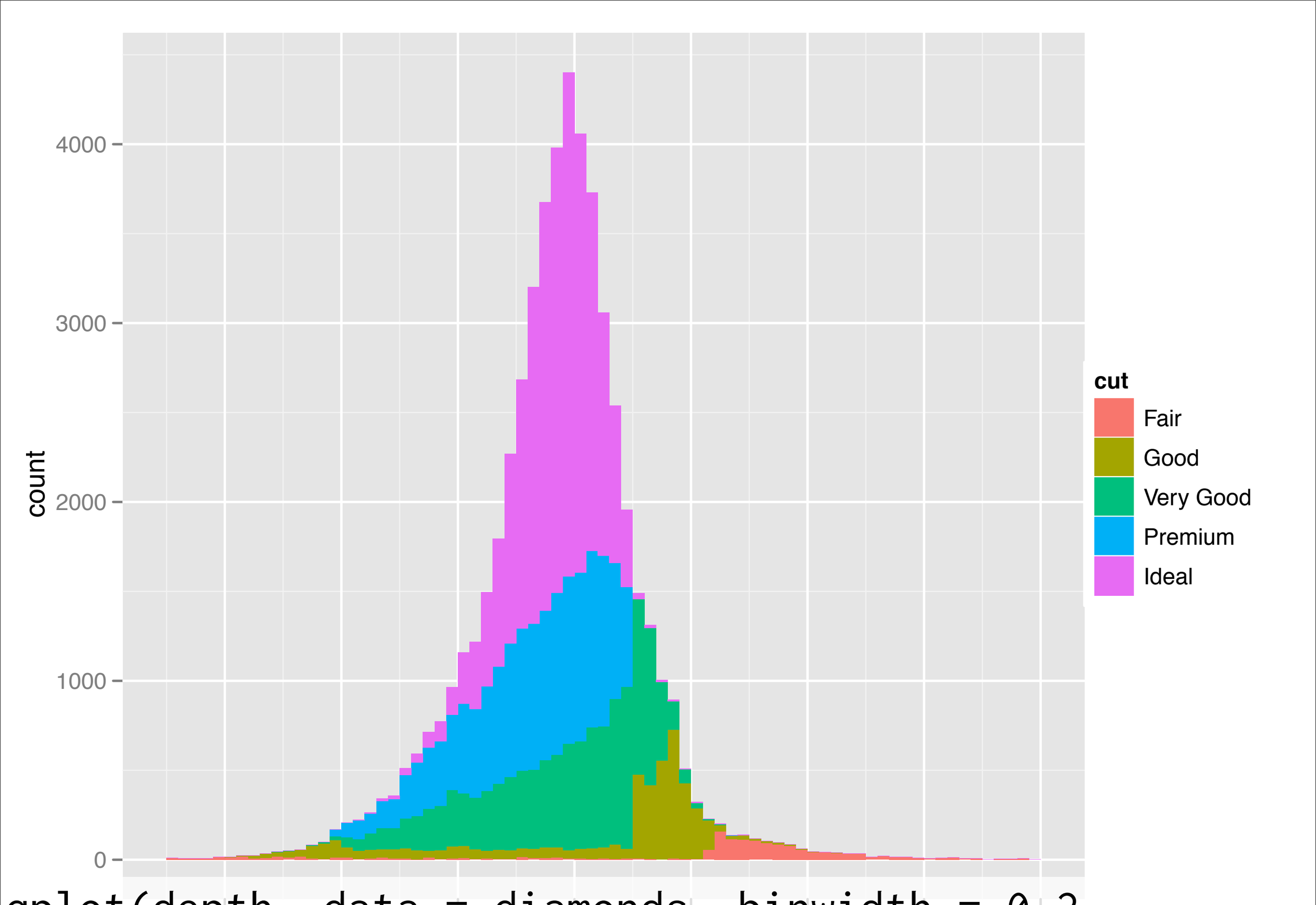
Additional variables

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

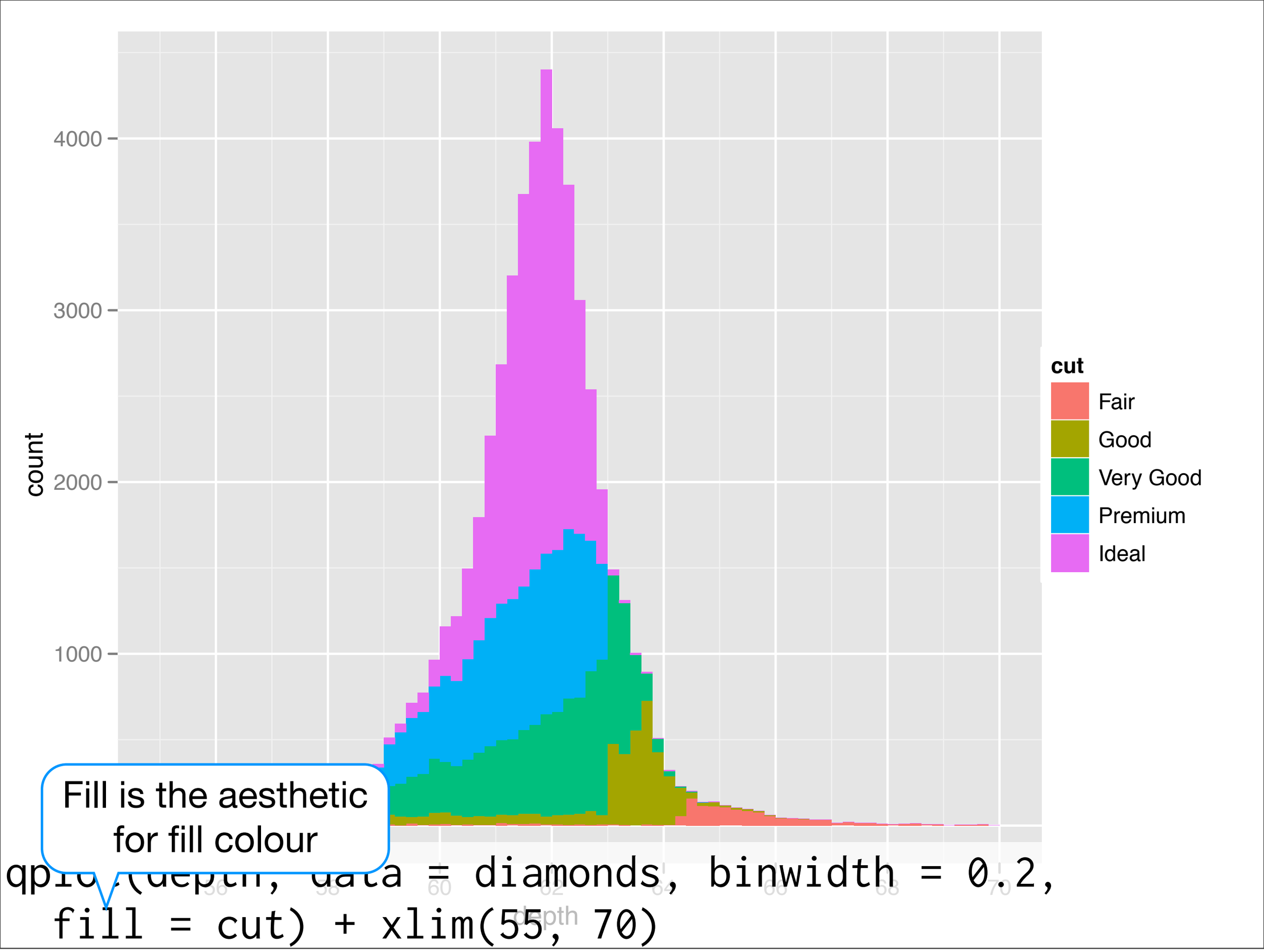
The following examples show the difference, when investigation the relationship between cut and depth.



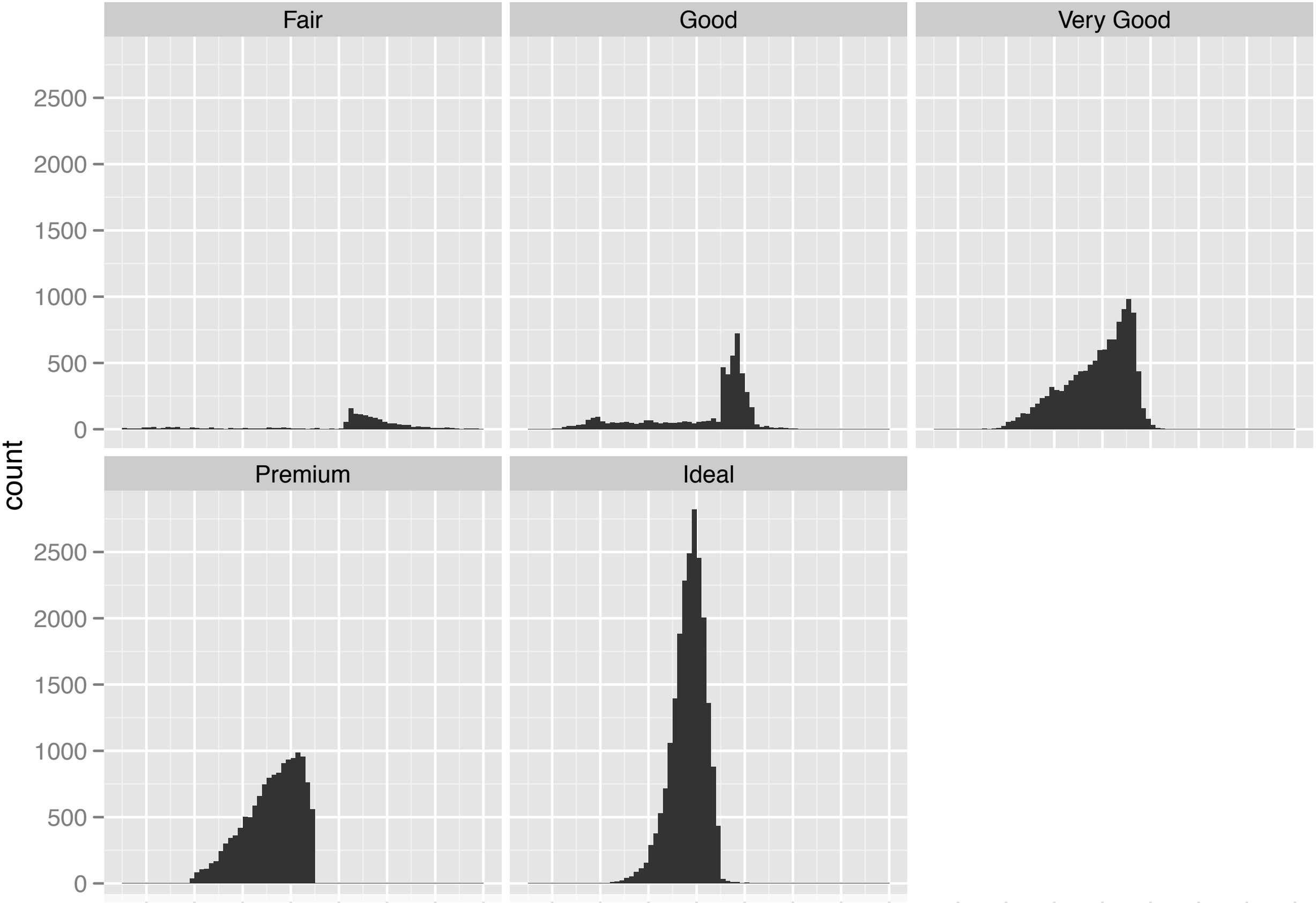

```
qplot(depth, data = diamonds, binwidth = 0.2)
```




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




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



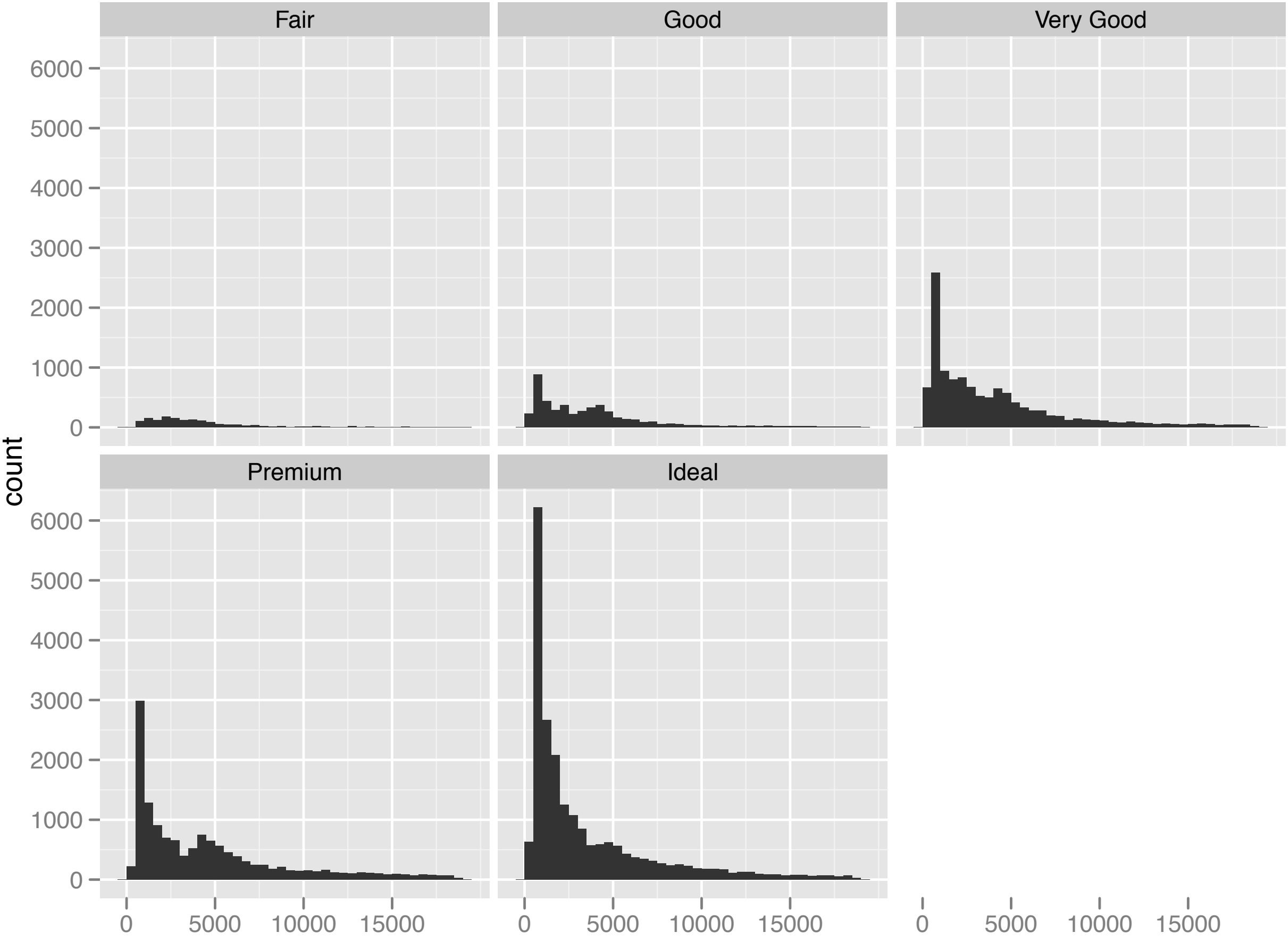

```
qplot(depth, data = diamonds, binwidth = 0.2) +
  xlim(55, 70) + facet_wrap(~cut)
```

Your turn

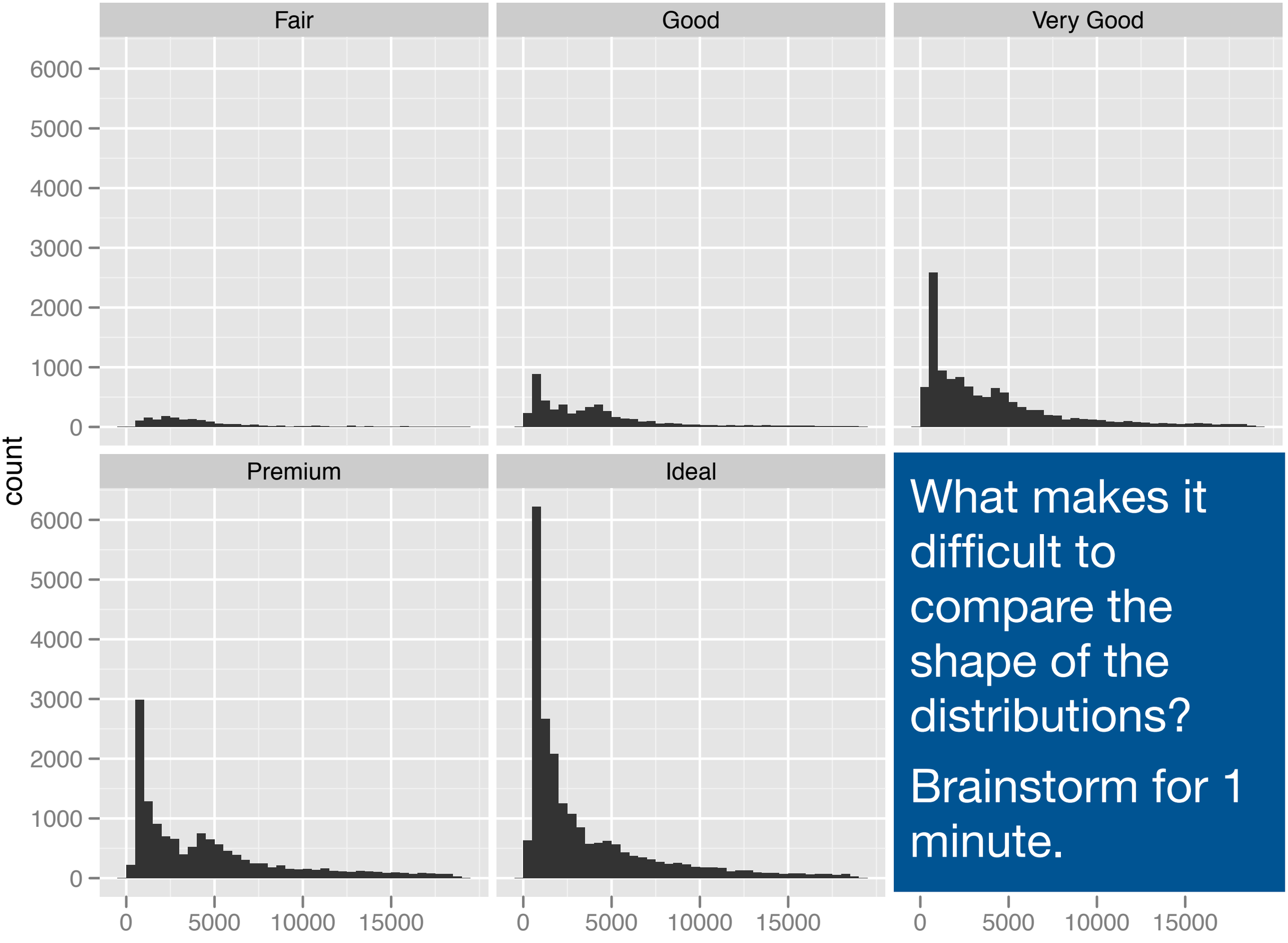
Explore the distribution of price.

How does it vary with colour, or cut, and clarity?

Practice zooming in on regions of interest.




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



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

Brainstorm for 1 minute.

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

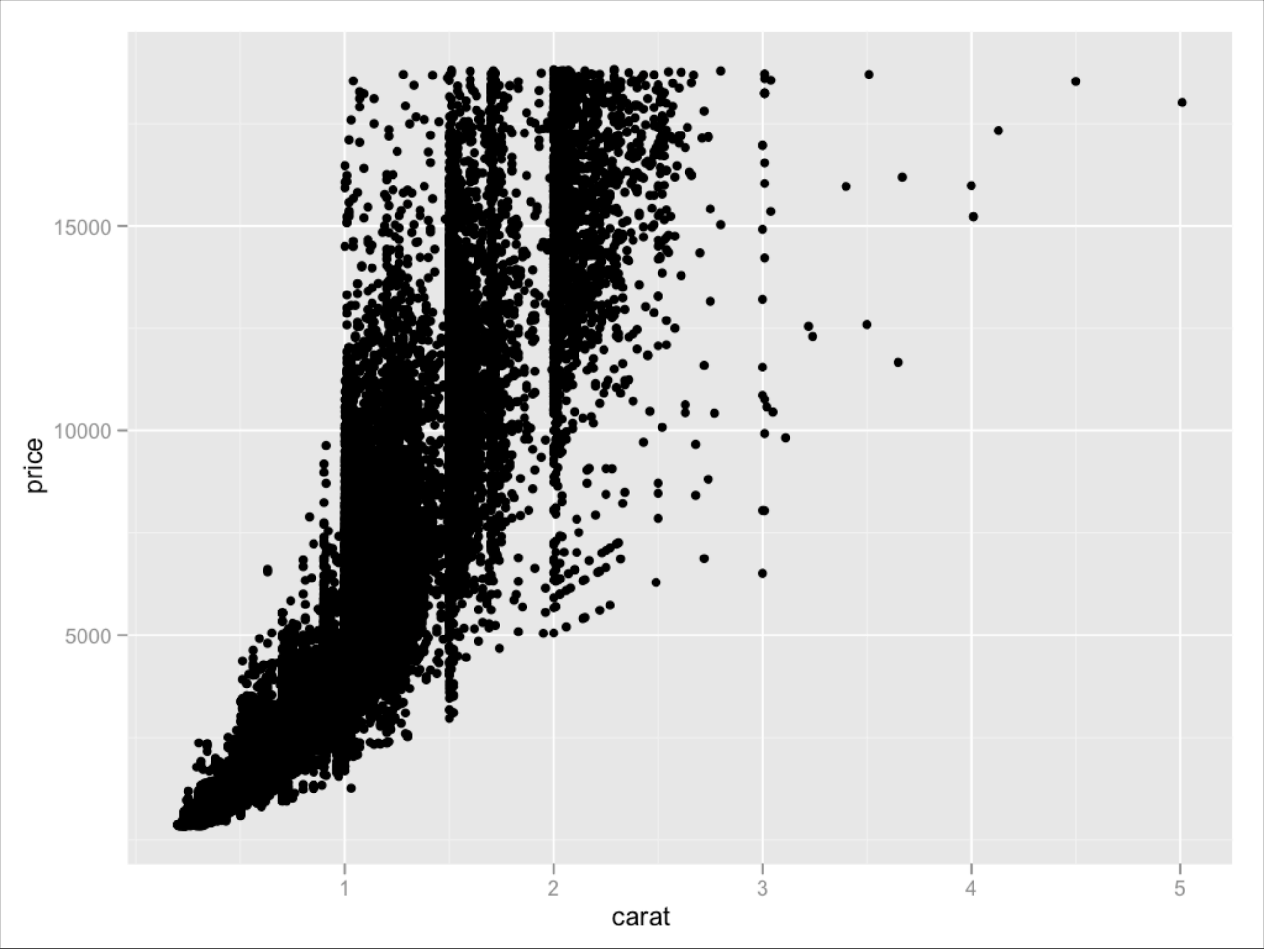
Problems

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)
```

Big scatterplots


15000
10000
5000
price
1
2
3
4
5
carat

Your turn

Take two minutes to brainstorm possible solutions to the overplotting problem.

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
```

```
# 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 = clarity)
```

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(~ clarity)
qplot(log10(carat), log10(price), data = diamonds,
  colour = clarity, geom = "smooth")
```

Workflow

Coding strategy

At the end of each interactive session, you want a summary of everything you did. Two options:

1. Save everything you did with `savehistory()` then remove the unimportant bits.

2. Build up the important bits as you go. (this is how I work)

Working directory

Set your working directory to specify where files will be loaded from and saved to – all paths are relative to the working directory.

From the terminal (linux or mac): the working directory is the directory you're in when you start R

On windows: File | Change dir.

On the mac: ⌘-D

In one directory

Data (.csv)

+

Code (.r)

+

Graphics (.png, .pdf)

+

Written report (.tex)

Graphics: Critique & creation

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

September 2011

Exploratory graphics

Are for **you** (not others). Need to be able to create rapidly because your first attempt will never be the most revealing.

Iteration is crucial for developing the best display of your data.

Gives rise to two key questions:

What should I plot?

How can I plot it?

Two general tools

Plot critique toolkit:
"graphics are like pumpkin pie"

Theory behind ggplot2:
"A layered grammar of graphics"

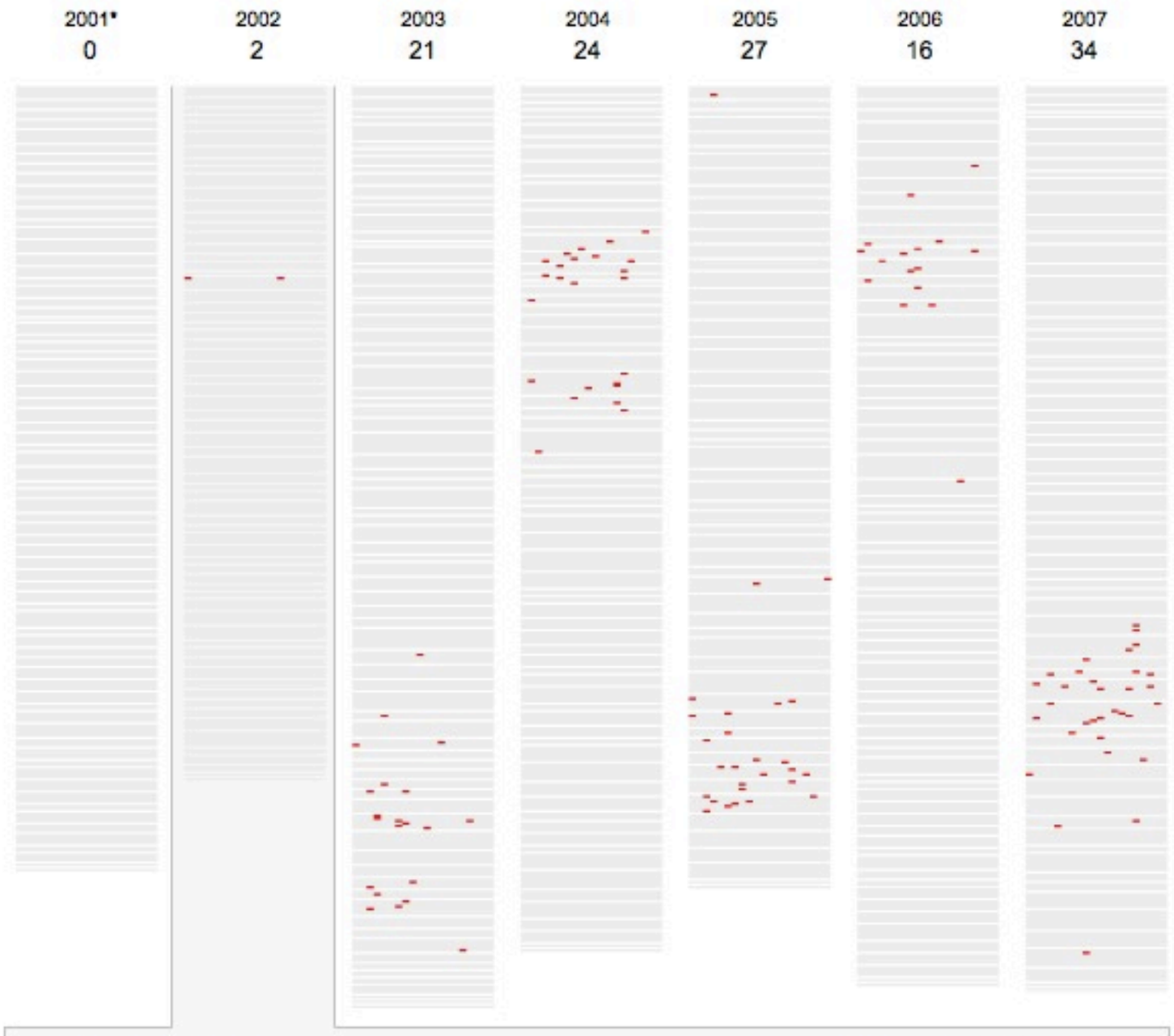
plus lots of practice...

What should I plot?

Critique

- State of the union: http://nyti.ms/r8KdvU
- How different groups spend their day: http://nyti.ms/np29Yk
- CA primary results: http://nyti.ms/r8Sh8N (Click margin of victory)

Use of the phrase "Iraq" in past State of the Union Addresses



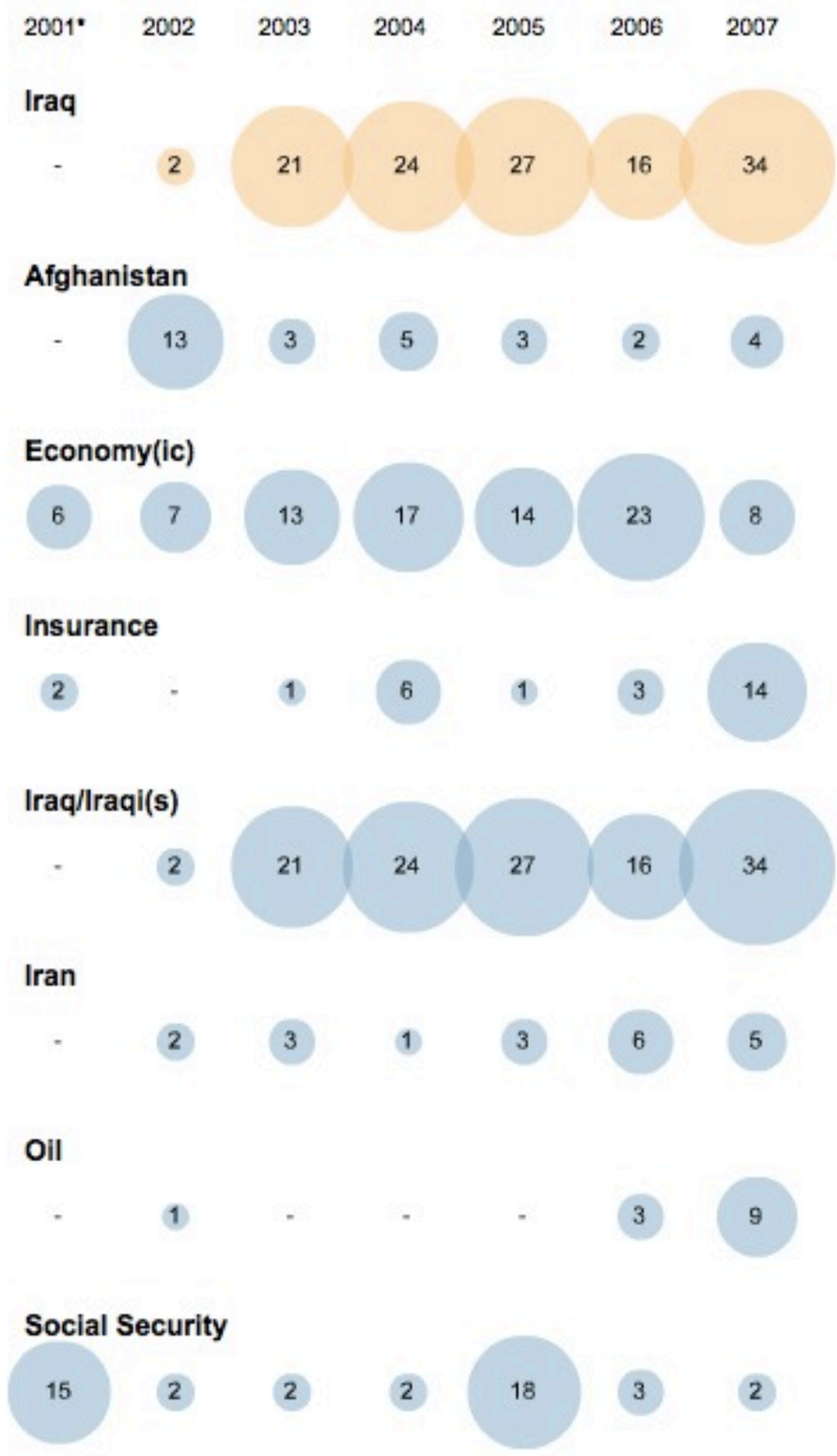
Next Instance of 'Iraq'

The word in context

IRAQ continues to flaunt its hostility toward America and to support terror. The Iraqi regime has plotted to develop anthrax, and nerve gas, and nuclear weapons for over a decade. This is a regime that has already used poison gas to murder thousands of its own citizens -- leaving the bodies of mothers huddled over their dead children. This is a regime that agreed to international inspections -- then kicked out the inspectors. This is a regime that has something to hide from the civilized world.

-- 2002 (Paragraph 20 of 67)

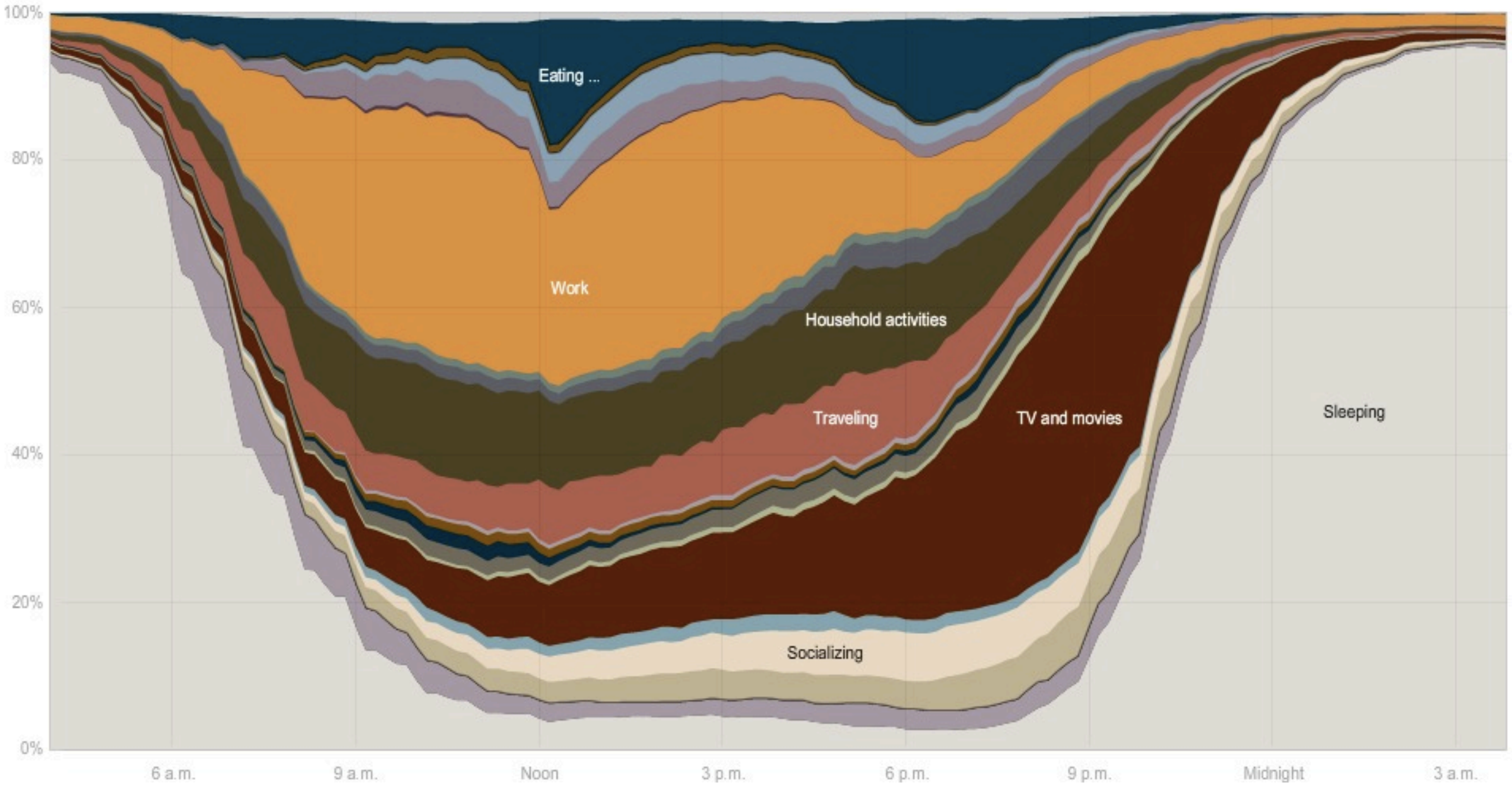
Compared with other words



Everyone

Sleeping, eating, working and watching television take up about two-thirds of the average day.

Everyone	Employed	White	Age 15-24	H.S. grads	No children
Men	Unemployed	Black	Age 25-64	Bachelor's	One child
Women	Not in lab...	Hispanic	Age 65+	Advanced	Two+ children

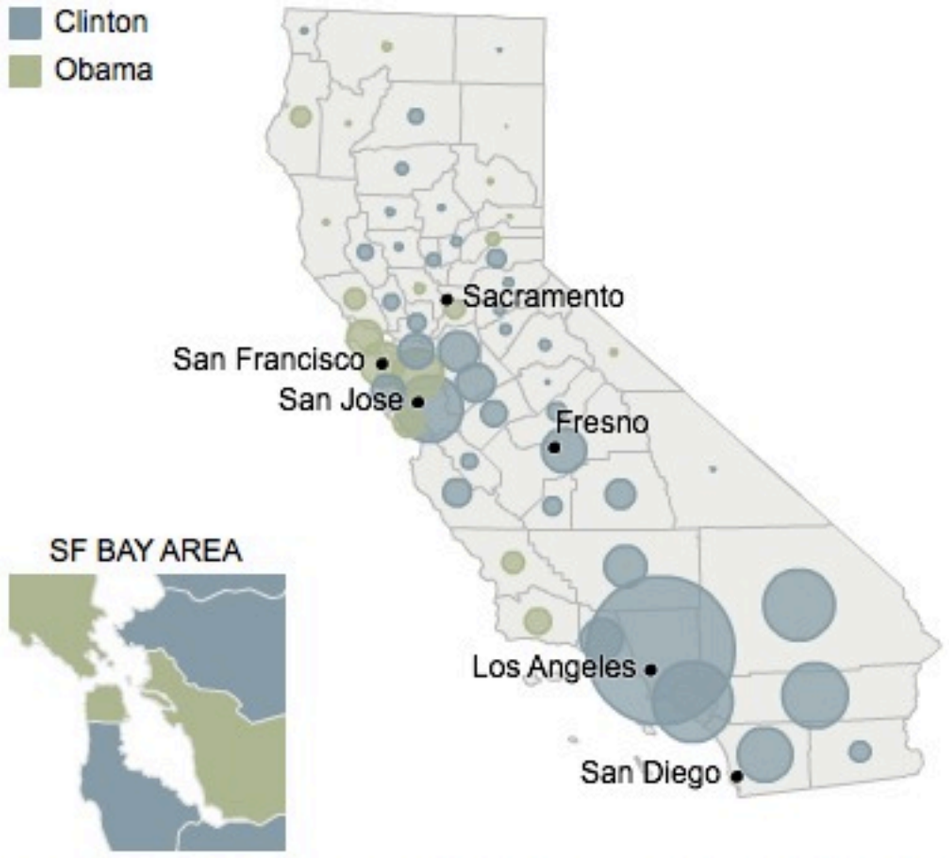


Results by County | **Margin of Victory**

Circles are proportional to the amount each county's leading candidate is ahead.

DEMOCRATS

Clinton
Obama



REPUBLICANS

McCain
Romney

Note: Maps show election returns as reported by The Associated Press.

Graphics are like pumpkin pie

The four C's of critiquing a graphic

Content

Construction


Context


Consumption

Content

What data (variables) does the graph display?

What non-data is present?

What is **pumpkin** (essence of the graphic) vs what is **spice** (useful additional info)?

Your turn

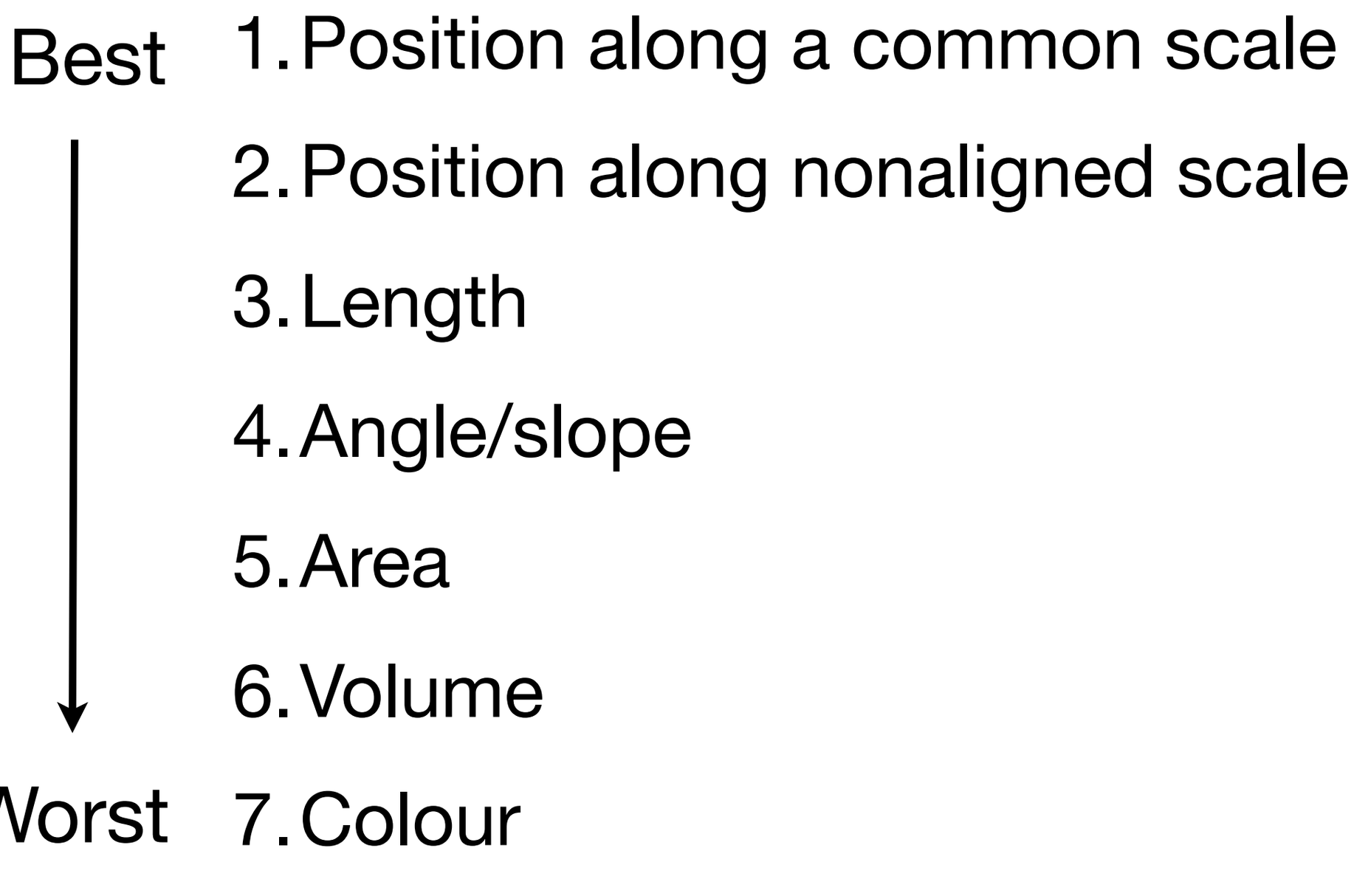
Pair up and identify the data and non-data in each of the three plots. Which features are the most important? Which are just useful background information?

Construction

How many layers are on the plot?

What data does each layer display? What sort of geometric object does it use? Is it a summary of the raw data? How are variables mapped to aesthetics?

Perceptual mapping



Your turn

Answer the following questions for each of the three plots:

How many layers are on the plot?

What data does the layer display? How does it display it?

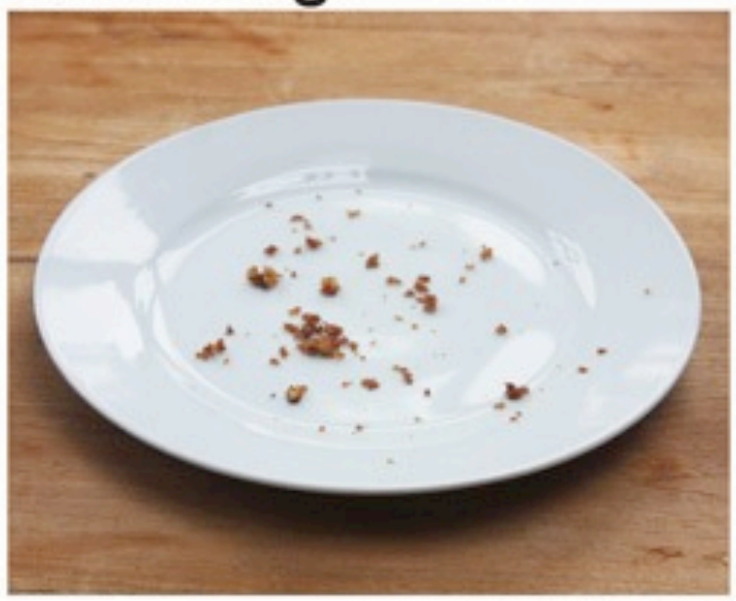
Another metaphor:

Data

Information

Presentation

Knowledge



http://epicgraphic.com/data-cake/

Can the explain composition of a graphic in words, but how do we create it?

How can I plot it?

"If any number of magnitudes are each the same multiple of the same number of other magnitudes, then the sum is that multiple of the sum."

Euclid, ~300 BC

"If any number of magnitudes are each the same multiple of the same number of other magnitudes, then the sum is that multiple of the sum."

Euclid, ~300 BC

$$m(\Sigma x) = \Sigma(mx)$$

The grammar of graphics

An abstraction which makes thinking about, reasoning about and communicating graphics easier.

Developed by Leland Wilkinson, particularly in “The Grammar of Graphics” 1999/2005

You’ve been using it in ggplot2 without knowing it! But to do more, you need to learn more about the theory.

What is a layer?

- Data
- Mappings from variables to aesthetics (**aes**)
- A geometric object (**geom**)
- A statistical transformation (**stat**)
- A position adjustment (**position**)

```
layer(geom, stat, position, data, mapping, ...)

layer(
  data = mpg,
  mapping = aes(x = displ, y = hwy),
  geom = "point",
  stat = "identity",
  position = "identity"
)

layer(
  data = diamonds,
  mapping = aes(x = carat),
  geom = "bar",
  stat = "bin",
  position = "stack"
)
```

```
# A lot of typing!

layer(
  data = mpg,
  mapping = aes(x = displ, y = hwy),
  geom = "point",
  stat = "identity",
  position = "identity"
)

# Every geom has an associated default statistic
# (and vice versa), and position adjustment.

geom_point(aes(displ, hwy), data = mpg)
geom_histogram(aes(carat), data = diamonds)
```

```r
# To actually create the plot
ggplot() +
  geom_point(aes(displ, hwy), data = mpg)
```

```r
ggplot() +
  geom_histogram(aes(displ), data = mpg)
```

```
# Multiple layers
ggplot() +
  geom_point(data = mpg, aes(displ, hwy)) +
  geom_smooth(data = mpg, aes(displ, hwy))

# Avoid redundancy:
ggplot(aes(displ, hwy), data = mpg) +
  geom_point() +
  geom_smooth()
```

```
# Different layers can have different aesthetics
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(colour = class)) +
  geom_smooth()

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

```
ggplot(mpg, aes(displ, hwy, group = class)) +
  geom_point(aes(colour = class)) +
  geom_smooth(method = "lm", se = F)

ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(colour = class)) +
  geom_line(aes(group = class), stat = "smooth",
    method = "lm", se = F)
```

	stat	geom
histogram	bin	bar
smooth	smooth	ribbon
boxplot	boxplot	boxplot
density	density	line
freqpoly	bin	line

Your turn

For each of the following plots created with qplot, recreate the equivalent ggplot code.

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

```
qplot(hwy, cty, data = mpg, geom = "jitter")

qplot(reorder(class, hwy), hwy, data = mpg,
  geom = c("jitter", "boxplot"))

qplot(log10(carat), log10(price),
data = diamonds, colour = color) +
geom_smooth(method = "lm")
```

```
ggplot(diamonds, aes(carat, price)) +
  geom_point()

ggplot(mpg, aes(hwy, cty)) +
  geom_jitter()

ggplot(mpg, aes(reorder(class, hwy), hwy)) +
  geom_jitter() +
  geom_boxplot()

ggplot(diamonds, aes(log10(carat), log10(price),
  colour = color)) +
  geom_point() +
  geom_smooth(method = "lm")
```

More geoms & stats

See http://had.co.nz/ggplot2 for complete list with helpful icons:

Geoms: (0d) point, (1d) line, **path**, (2d) boxplot, bar, **tile**, **text**, polygon, linerange.

Stats: bin, density, summary, sum

Advanced layering

Layering

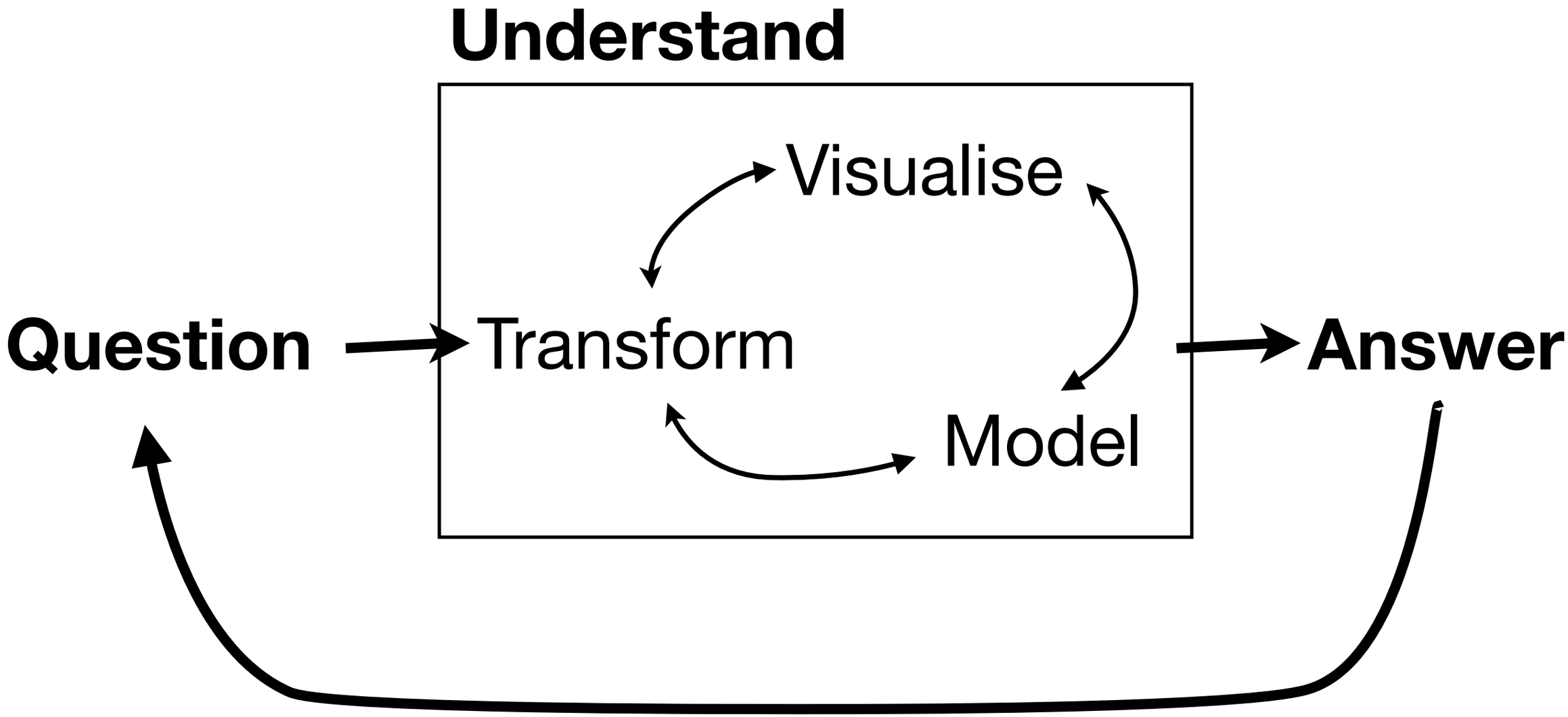
Key to rich graphics is taking advantage of layering.

Three types of layers: context, raw data, and summarised data

Each can come from a different dataset.

Iteration

- First plot is never the best. Have to keep iterating to understand what's going on.
- Don't try and do too much in one plot.
- Best data analyses tell a story, with a natural flow from beginning to end.


Understand
Visualise
Question
Transform
Answer
Model

```
qplot(x, y, data = diamonds)
diamonds$x[diamonds$x == 0] <- NA
diamonds$y[diamonds$y == 0] <- NA
diamonds$y[diamonds$y > 20] <- NA
```

```
diamonds <- mutate(diamonds,
  area = x * y,
  lratio = log10(x / y))

qplot(area, lratio, data = diamonds)
diamonds$lratio[abs(diamonds$lratio) > 0.02] <- NA
```

```
ggplot(diamonds, aes(area, lratio)) +
  geom_point()

ggplot(diamonds, aes(area, lratio)) +
  geom_hline(yintercept = 0, size = 2, colour = "white") +
  geom_point() +
  geom_smooth(method = lm, se = F, size = 2)

ggplot(diamonds, aes(area, abs(lratio))) +
  geom_hline(yintercept = 0, size = 2, colour = "white") +
  geom_point() +
  geom_smooth(se = F, size = 2)
```

```
ggplot(diamonds, aes(area, abs(lratio))) +
  geom_hline(yintercept = 0, size = 2, colour = "white") +
  geom_boxplot(aes(group = round_any(area, 5))) +
  geom_smooth(se = F, size = 2)

ggplot(diamonds, aes(area, abs(lratio))) +
  geom_hline(yintercept = 0, size = 2, colour = "white") +
  geom_boxplot(aes(group = round_any(area, 5)))

ggplot(diamonds, aes(area, lratio)) +
  geom_hline(yintercept = 0, size = 2, colour = "white") +
  geom_boxplot(aes(group = interaction(sign(lratio),
   round_any(area, 5))), position = "identity")
```

Polishing plots for presentation

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair
Department of Statistics / Rice University

September 2011

Communication graphics

When you need to **communicate** your findings, you need to spend a lot of time polishing your graphics to eliminate distractions and focus on the story.

Now it’s time to pay attention to the small stuff: labels, colour choices, tick marks...

1. **Scales**: used to override default perceptual mappings, and tune parameters of axes and legends.

2. **Coordinate systems**: override default Cartesian coordinate system

3. **Themes**: control presentation of non-data elements.

4. **Saving your work**: to include in reports, presentations, etc.

Scales

Scales

Control how data is mapped to perceptual properties, and produce **guides** (axes and legends) which allow us to read the plot.

Important parameters: **name**, **breaks** & **labels**, **limits**.

Naming scheme: scale_*aesthetic_name*.
All default scales have name continuous or discrete.

```
# Default scales
scale_x_continuous()
scale_y_discrete()
scale_colour_discrete()

# Custom scales
scale_colour_hue()
scale_x_log10()
scale_fill_brewer()

# Scales with parameters
scale_x_continuous("X Label", limits = c(1, 10))
scale_colour_gradient(low = "blue", high = "red")
```

```
p <- qplot(cyl, displ, data = mpg)

# First argument (name) controls axis label
p + scale_y_continuous("Displacement (l)")
p + scale_x_continuous("Cylinders")

# Breaks and labels control tick marks
p + scale_x_continuous(breaks = c(4, 6, 8))
p + scale_x_continuous(breaks = c(4, 6, 8),
  labels = c("small", "medium", "big"))

# Limits control range of data
p + scale_y_continuous(limits = c(1, 8))
# same as:
p + ylim(1, 8)
```

Your turn

```
qplot(carat, price, data = diamonds, geom = "bin2d")
```

Manipulate the fill colour legend to:

- Change the title to “Count"
- Display breaks at 1000, 3500 & 7000
- Add commas to the keys (e.g. 1,000)
- Set the limit for the scale from 0 to 8000.

```
p <- qplot(carat, price, data = diamonds, geom = "hex")

# First argument (name) controls legend title
p + scale_fill_continuous("Count")

# Breaks and labels control legend keys
p + scale_fill_continuous(breaks = c(1000, 3500, 7000))
p + scale_fill_continuous(breaks = c(0, 4000, 8000))

# Why don't 0 and 8000 have colours?
p + scale_fill_continuous(breaks = c(0, 4000, 8000),
  limits = c(0, 8000))

# Can use labels to make more human readable
breaks <- c(0, 2000, 4000, 6000, 8000)
labels <- format(breaks, big.mark = ",")
p + scale_fill_continuous(breaks = breaks, labels = labels,
   limits = c(0, 8000))
```

```
p <- qplot(color, carat, data = diamonds)

# Basically the same for discrete variables
p + scale_x_discrete("Color")

# Except limits is now a character vector
p + scale_x_discrete(limits = c("D", "E", "F"))

# Should work for boxplots too
qplot(color, carat, data = diamonds,
  geom = "boxplot") +
  scale_x_discrete(limits = c("D", "E", "F"))
```

Alternate scales

Can also override the default choice of scales. You are most likely to want to do this with **colour**, as it is the most important aesthetic after position.

Need a little background to be able to use colour effectively: colour **spaces** & colour **blindness**.

Colour spaces

Most familiar is **rgb**: defines colour as mixture of **red**, **green** and **blue**. Matches the physics of eye, but the brain does a lot of post-processing, so it's hard to directly perceive these components.

A more useful colour space is hcl: **hue**, **chroma** and **luminance**

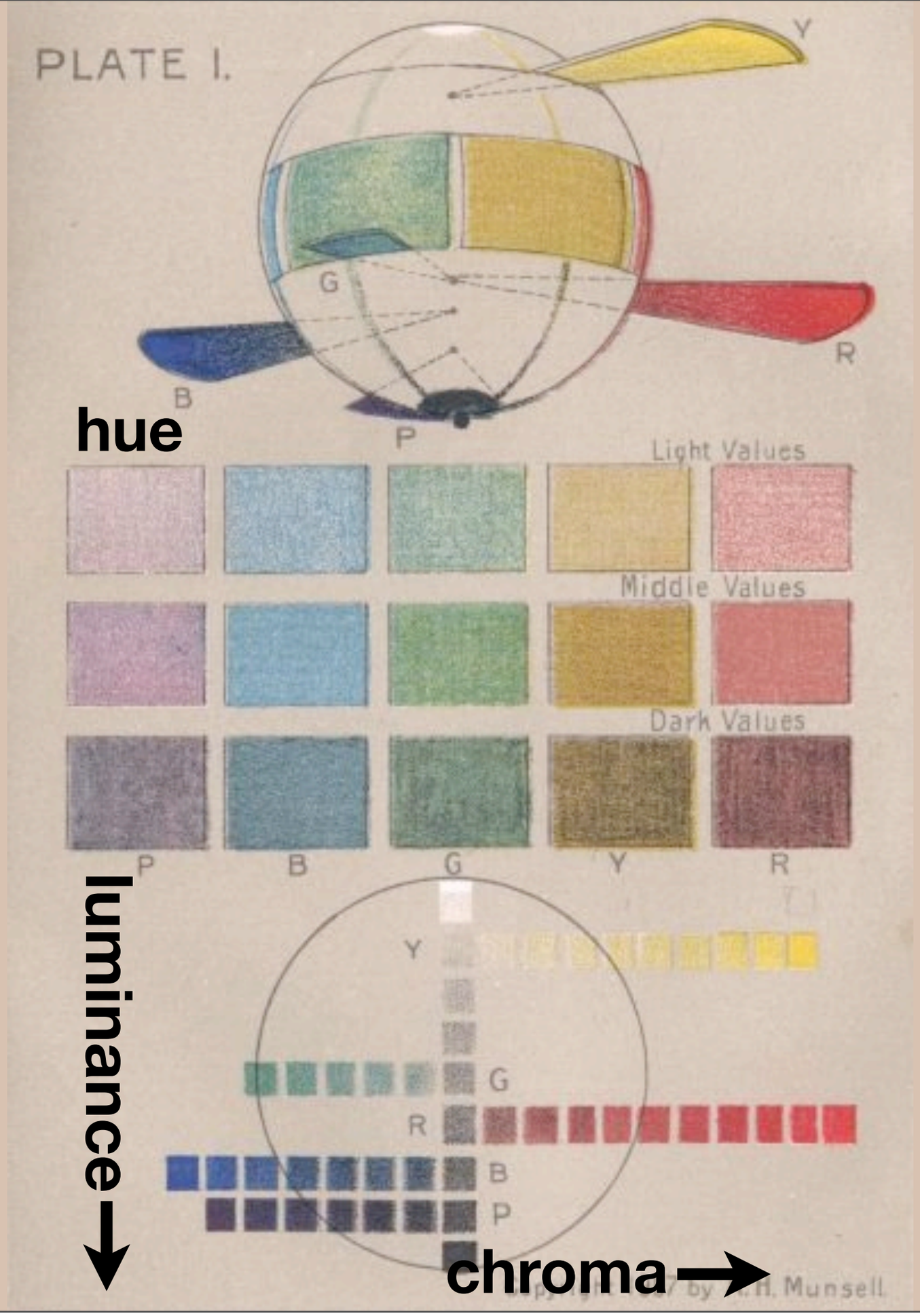


PLATE I.
Y
G
R
B
P
hue
Light Values
Middle Values
Dark Values
P
B
G
Y
R
Y
G
R
B
P
luminance
chroma
by A. H. Munsell

Default colour scales

Discrete: evenly spaced hues of equal chroma and luminance. No colour appears more important than any other. Does not imply order.

Continuous: evenly spaced hues between two colours.

Alternatives

Discrete: brewer, grey

Continuous: gradient2, gradientn

Color brewer

Cynthia Brewer applied basics principles and then rigorously tested to produce selection of good palettes, particularly tailored for maps: http://colorbrewer2.org/

Can use `cut_interval()` or `cut_number()` to convert continuous to discrete.

Colour blindness

7-10% of men are red-green colour “blind”. (Many other rarer types of colour blindness)

Solutions: avoid red-green contrasts; use redundant mappings; **test**. I like color oracle: http://colororacle.cartography.ch

Your turn

Read through the examples for `scale_colour_brewer`, `scale_colour_gradient2` and `scale_colour_gradientn`.

Experiment!

Coordinate systems

```
# For spatial data:
coord_map()

# To zoom in on plot
coord_cartesian(xlim = ..., ylim = ...)

# Polar coordinates
coord_polar()

# Equal coordinates
coord_equal()
```

Themes

Visual appearance

So far have only discussed how to get the data displayed the way you want, focussing on the essence of the plot.

Themes give you a huge amount of control over the appearance of the plot, the choice of background colours, fonts and so on.

```
# Two built in themes.  The default:
qplot(carat, price, data = diamonds)

# And a theme with a white background:
qplot(carat, price, data = diamonds) + theme_bw()

# Use theme_set if you want it to apply to every
# future plot.
theme_set(theme_bw())

# This is the best way of seeing all the default
# options
theme_bw()
theme_grey()
```

Elements

You can also make your own theme, or modify and existing.

Themes are made up of elements which can be one of: `theme_line`, `theme_segment`, `theme_text`, `theme_rect`, `theme_blank`

Gives you a lot of control over plot appearance.

Elements

Axis: axis.line, axis.text.x, axis.text.y, axis.ticks, axis.title.x, axis.title.y

Legend: legend.background, legend.key, legend.text, legend.title

Panel: panel.background, panel.border, panel.grid.major, panel.grid.minor

Strip: strip.background, strip.text.x, strip.text.y

```
p <- qplot(displ, hwy, data = mpg) +
  opts(title = "Bigger engines are less efficient")

# To modify a plot
p
p + opts(plot.title =
  theme_text(size = 12, face = "bold"))
p + opts(plot.title = theme_text(colour = "red"))
p + opts(plot.title = theme_text(angle = 45))
p + opts(plot.title = theme_text(hjust = 1))
```

Your turn

Fix the overlapping y labels on this plot:

```
qplot(reorder(model, hwy), hwy,
data = mpg)
```

Rotate the labels on these strips so they are easier to read.

```
qplot(hwy, reorder(model, hwy), data =
mpg) + facet_grid(manufacturer ~ .,
scales = "free", space = "free")
```

Saving your work

```
qplot(price, carat, data = diamonds)
ggsave("diamonds.png")

# Selects graphics device based on extension
ggsave("diamonds.png")
ggsave("diamonds.pdf")
```

```
# Uses on-screen device size, or override with
# width & height (to be reproducible)
ggsave("diamonds.png", width = 6, height = 6)

# Outputs last plot by default, override
# with plot:
dplot <- qplot(carat, price, data = diamonds)
ggsave("diamonds.png", plot = dplot)

# Defaults to 300 dpi for png
ggsave("diamonds.png", dpi = 72)
```

Raster	Vector
pixel-based	instruction-based
png	pdf
for plots with many points	for all other plots
ms office, web	latex

Your turn

Save a pdf of a scatterplot of price vs carat. Open it up in adobe acrobat.

Save a png of the same scatterplot and embed it into a word or latex document.

Where next?


Learning a new
language is hard!

Learning more

ggplot2 mailing list:
http://groups.google.com/group/ggplot2

ggplot2: Elegant Graphics for Data Analysis
http://amzn.com/0387981403

Introduction to Statistics with R
http://amzn.com/0387954759

Data manipulation with R
http://amzn.com/0387747303