Visualisation in R

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

February 2010

HELLO

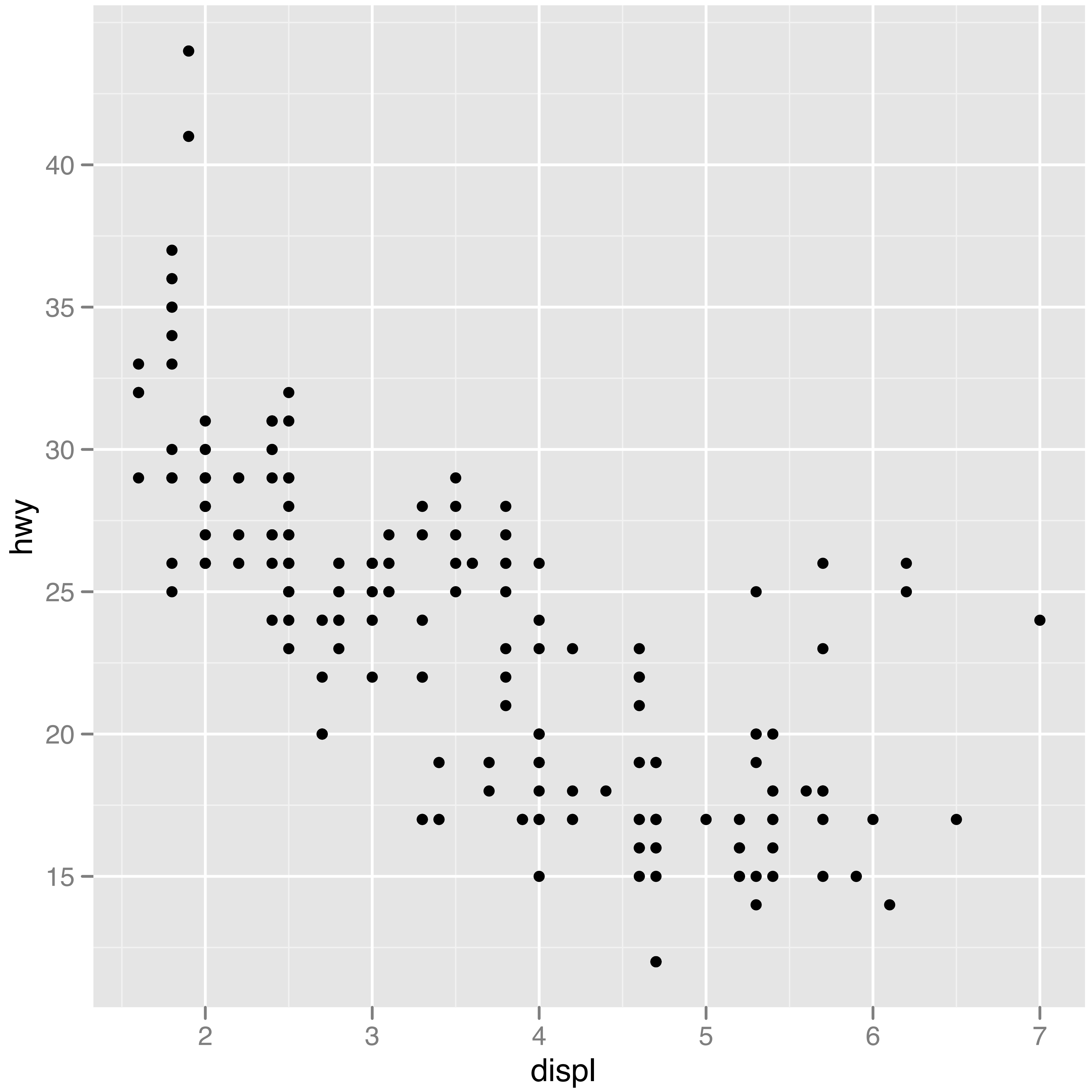
my name is

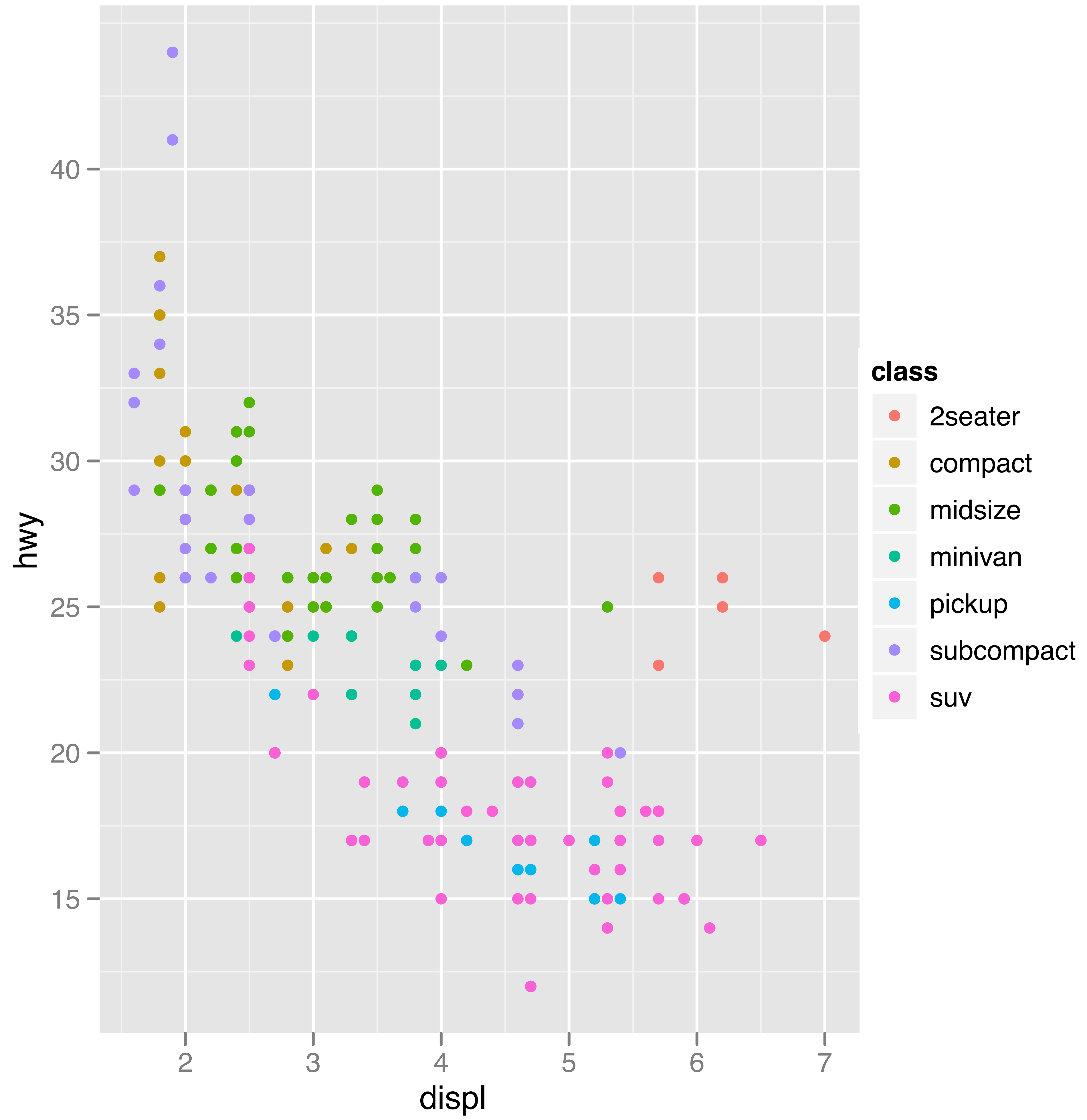
Hadley

http://had.co.nz/rice-vis

1. Preview of today
2. About ggplot2
3. More resources
4. Diving in

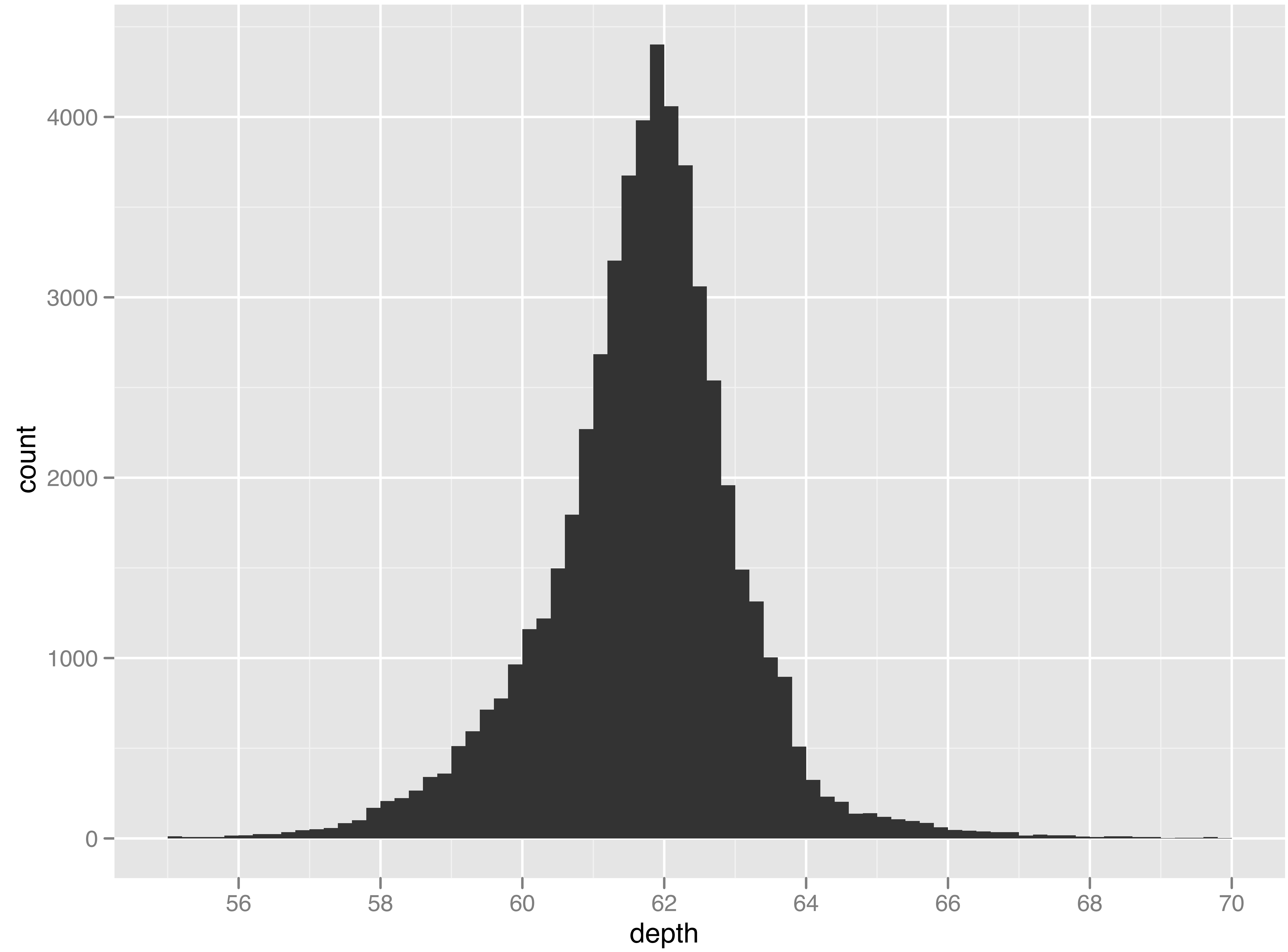
Fuel economy
Basic graphics

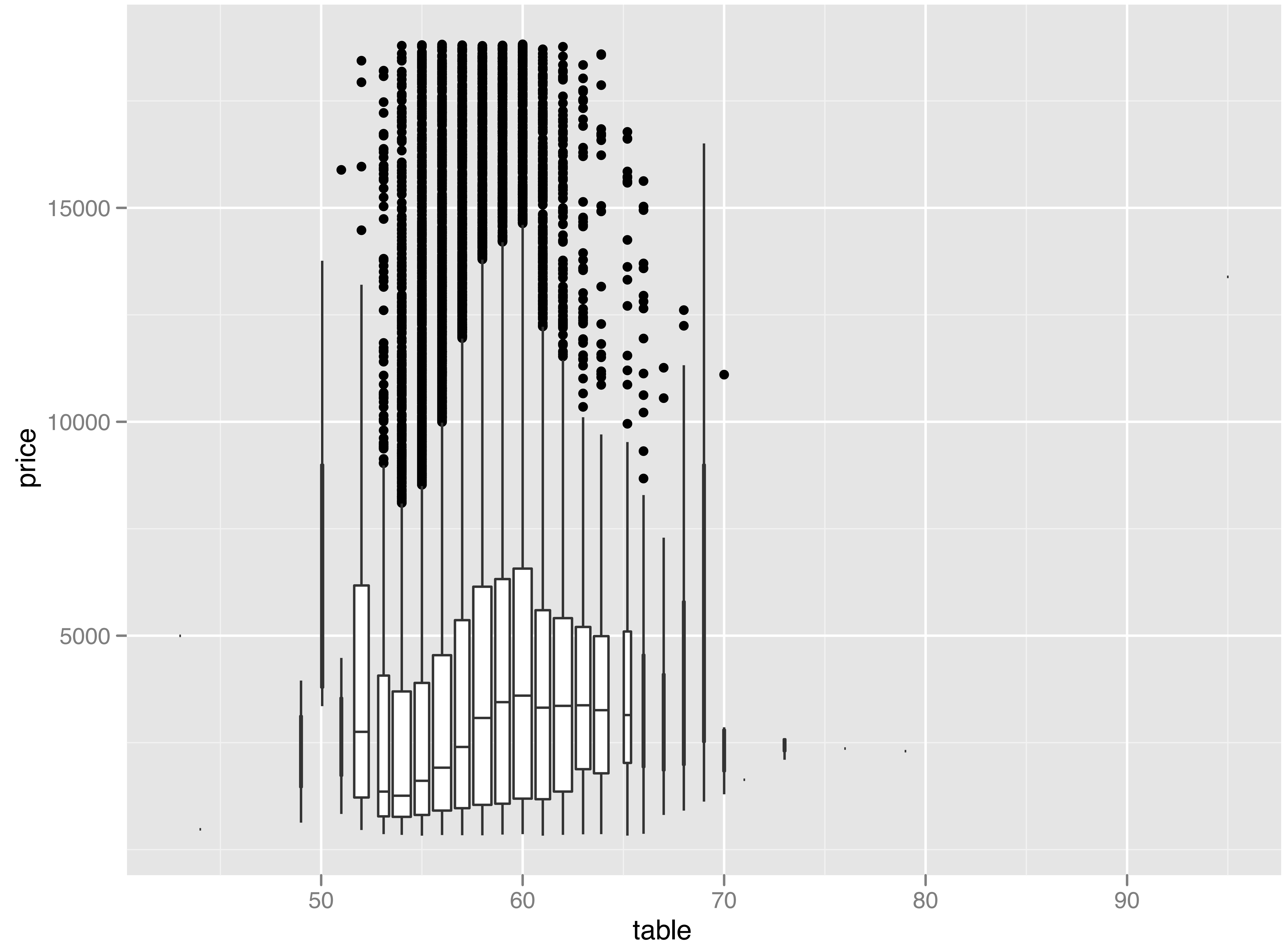

hwy
displ
40
35
30
25
20
15
2
3
4
5
6
7


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

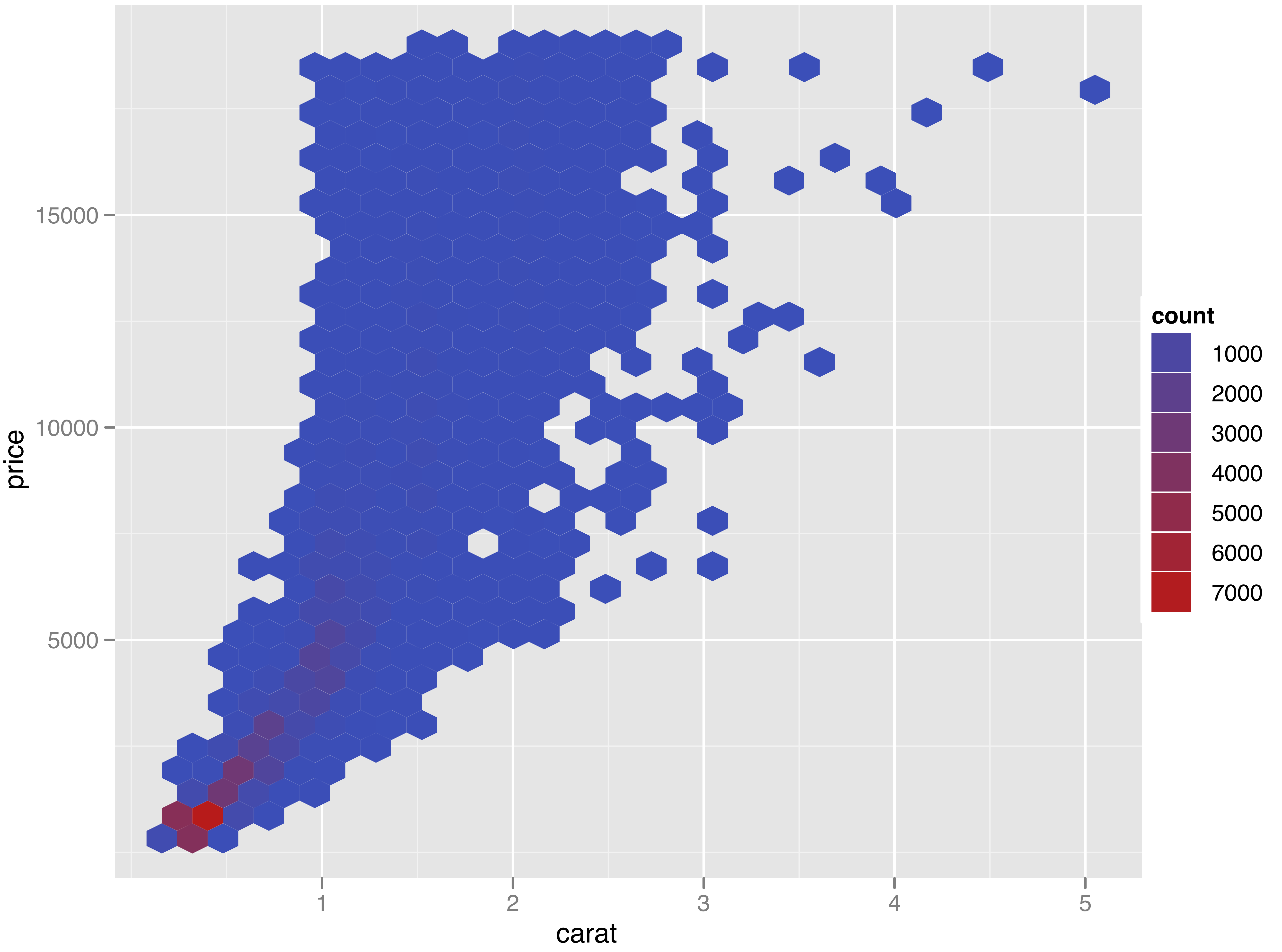
Diamond prices

Displaying large data


count
4000
3000
2000
1000
0
56
58
60
62
64
66
68
70
depth

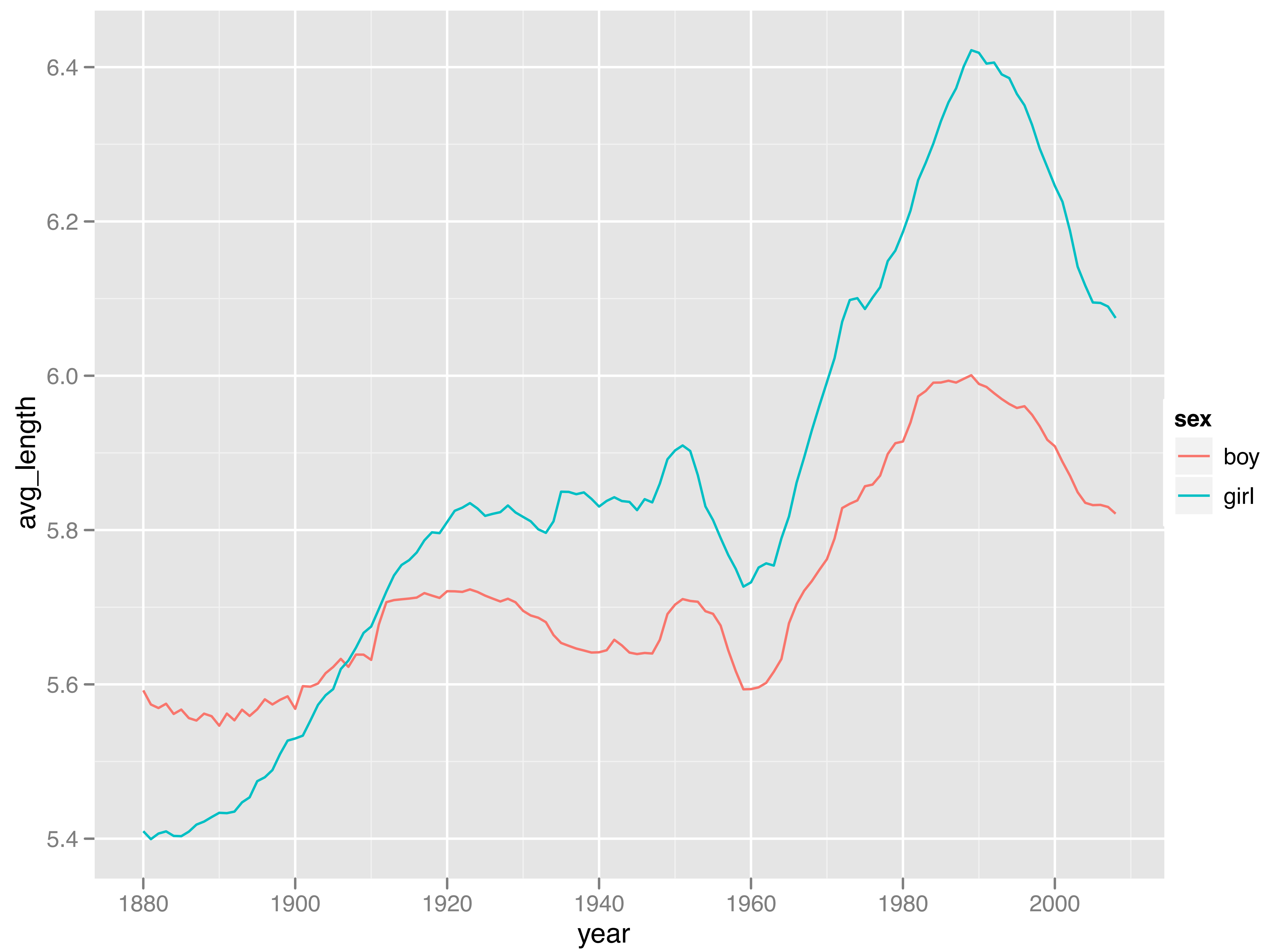


price
15000
10000
5000
table
50
60
70
80
90

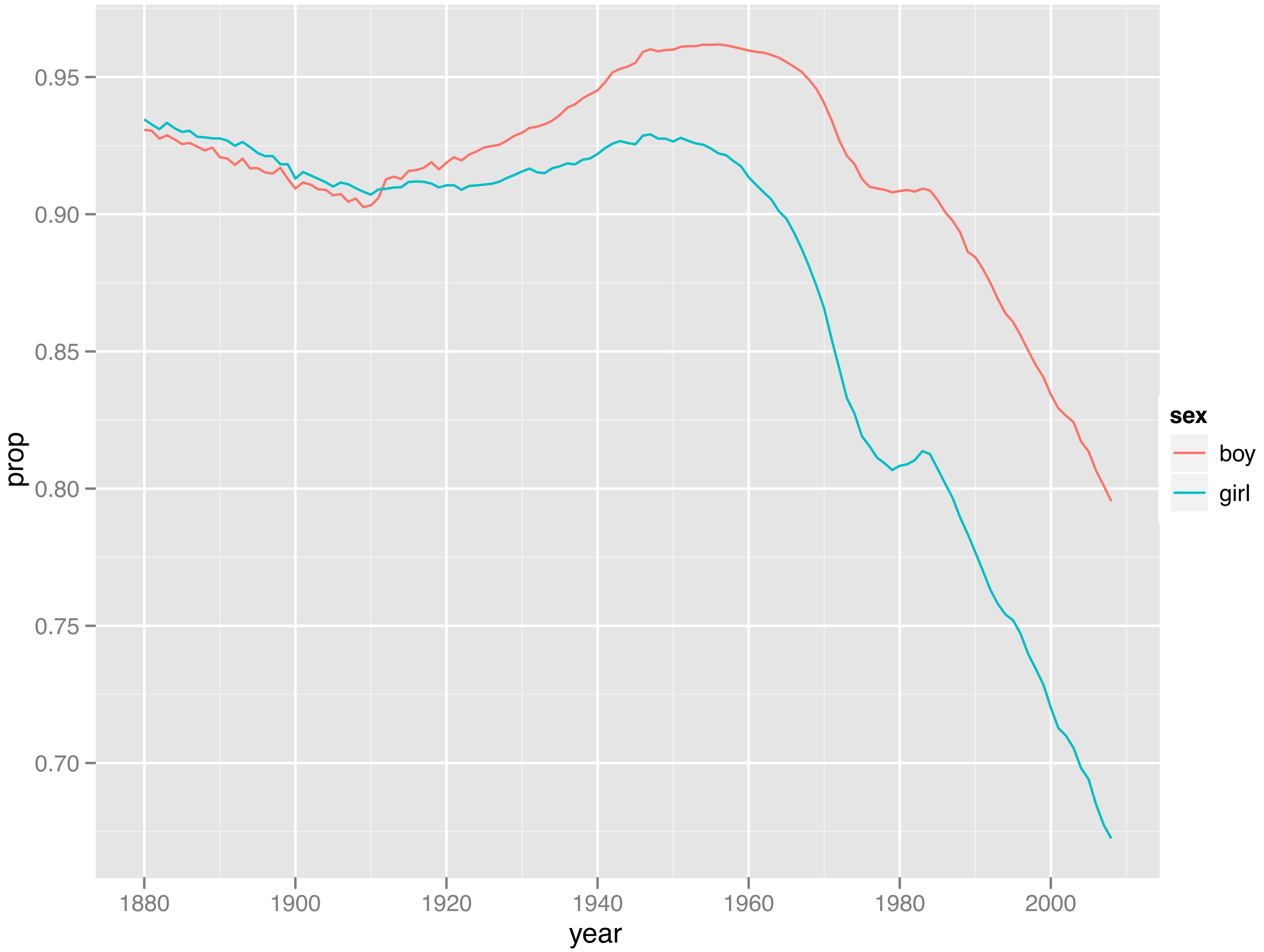

price
15000
10000
5000
carat
1
2
3
4
5
count
1000
2000
3000
4000
5000
6000
7000

US baby names

Data manipulation and transformation



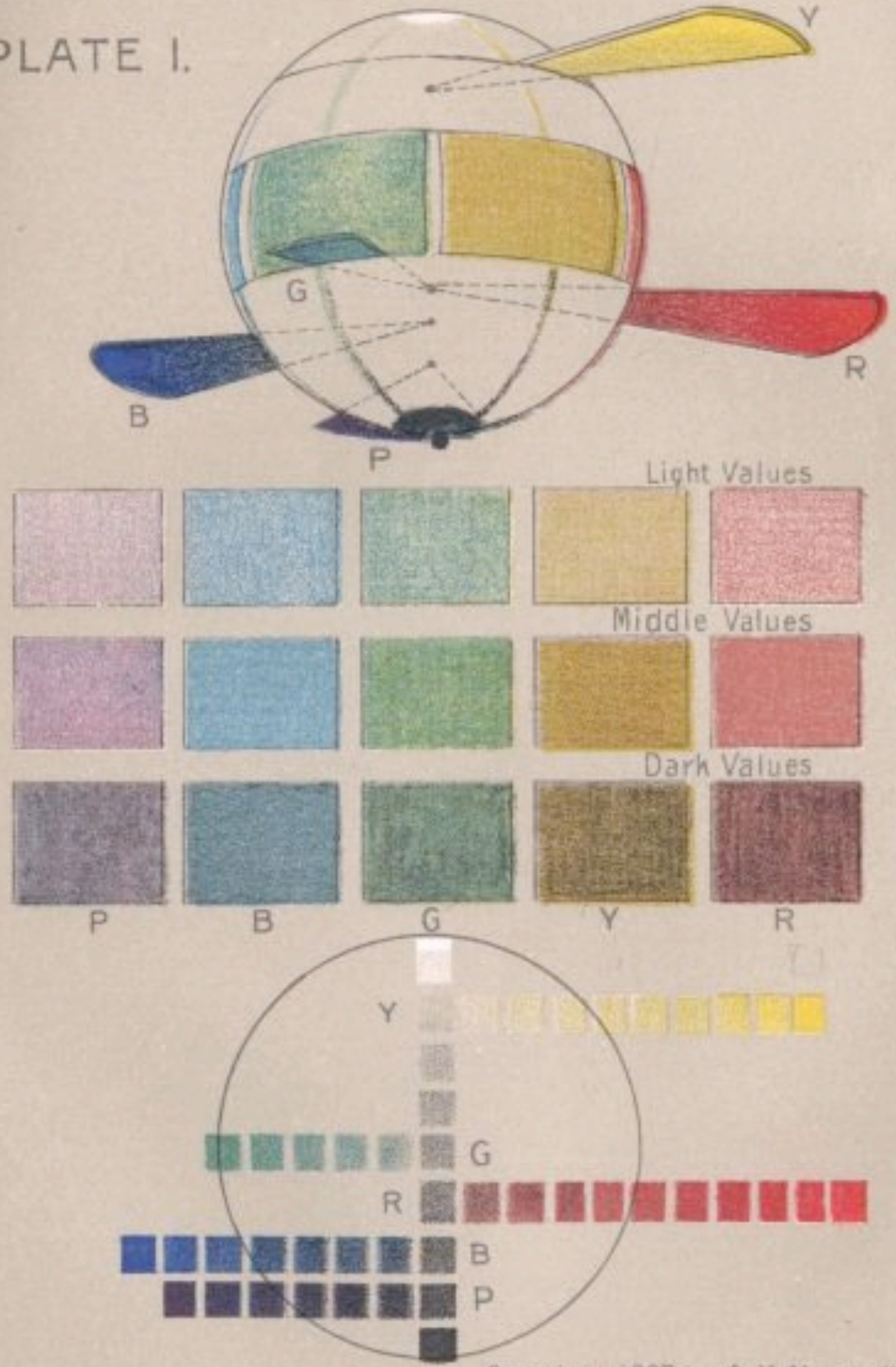
6.4
6.2
6.0
5.8
5.6
5.4
avg_length
1880
1900
1920
1940
1960
1980
2000
year
sex
boy
girl


0.95
0.90
0.85
0.80
0.75
0.70
prop
1880
1900
1920
1940
1960
1980
2000
year
sex
boy
girl

Polishing your plots

1. **Scales**: used to override default perceptual mappings, and tune parameters of axes and legends.
2. **Themes**: control presentation of non-data elements.
3. **Saving your work**: to include in reports, presentations, etc.

PLATE I.



ggplot2

About ggplot2

Graphical grammar (domain specific language), based on “The Grammar of Graphics” by Leland Wilkinson.

Specify what you want, not how to create it. Many fiddly details taken care of.

“Instead of spending time making your graph look pretty, you can focus on creating a graph that bests reveals the messages in your data.”

Useful resources

http://had.co.nz/ggplot2

http://had.co.nz/ggplot2/book

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

http://learnr.wordpress.com

http://ggplot2.wik.is

Learning a new language is hard!

Scatterplot basics

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

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

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

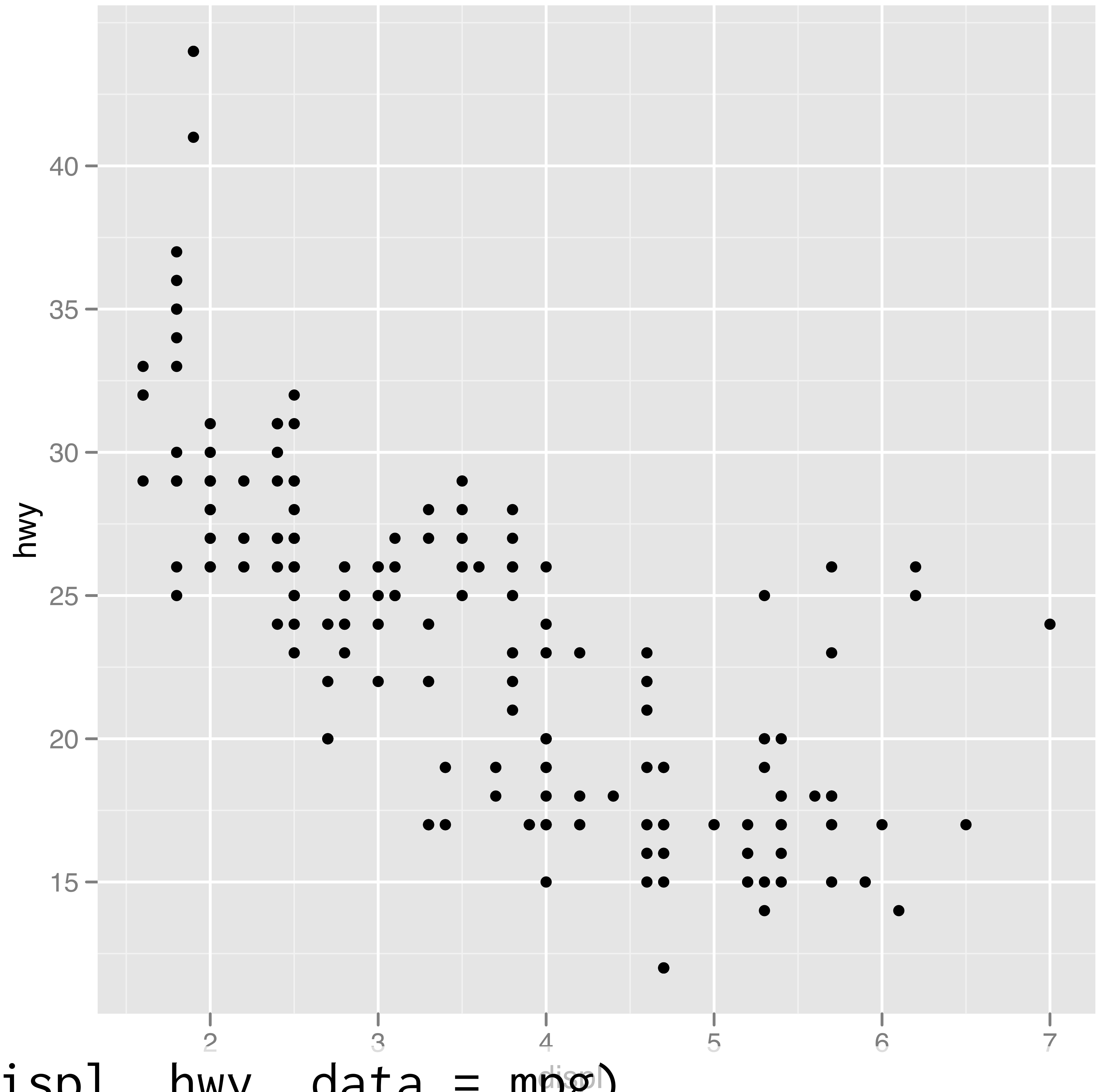
Scatterplot basics

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

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

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

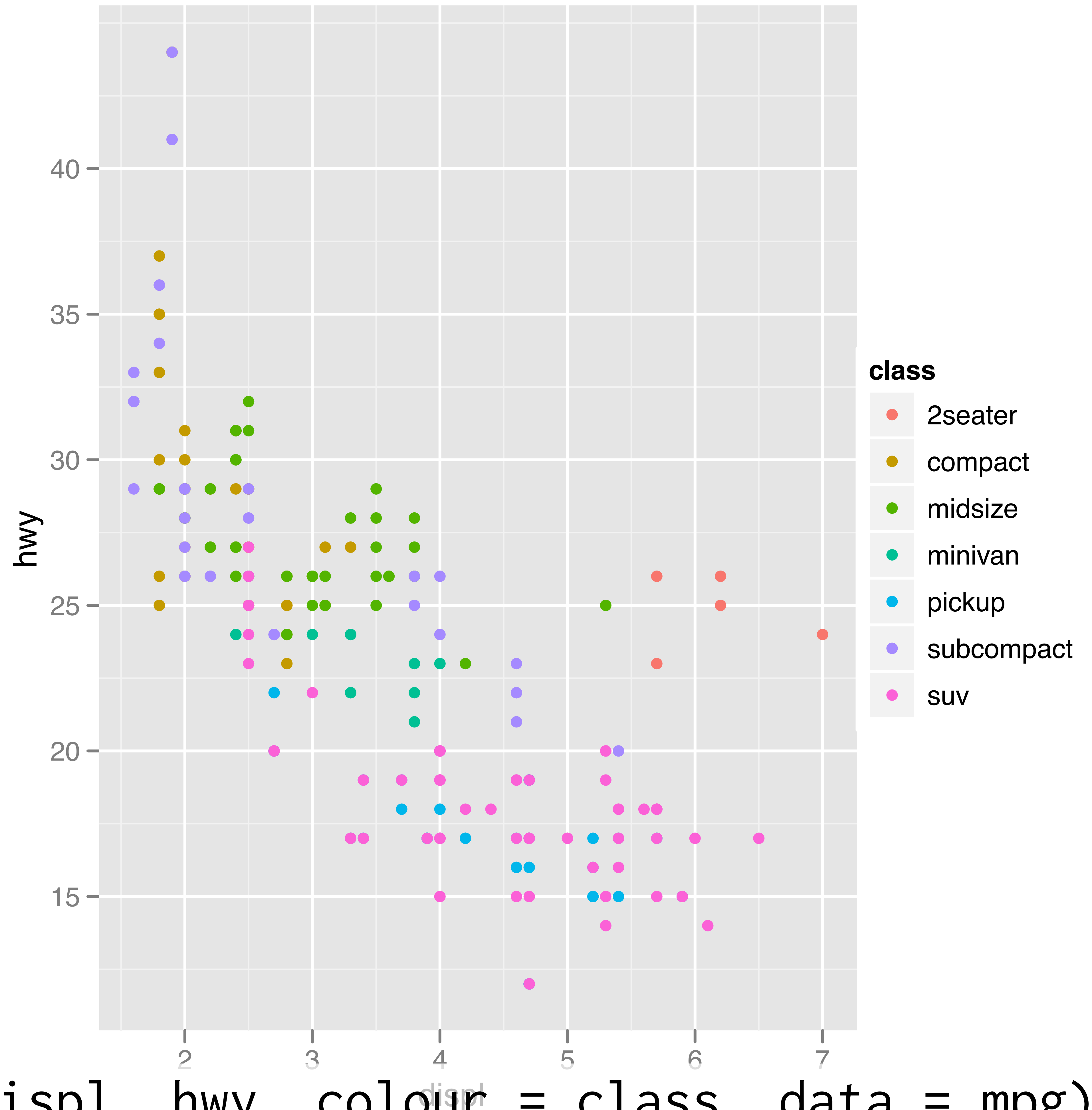
In ggplot2, we always explicitly specify the data



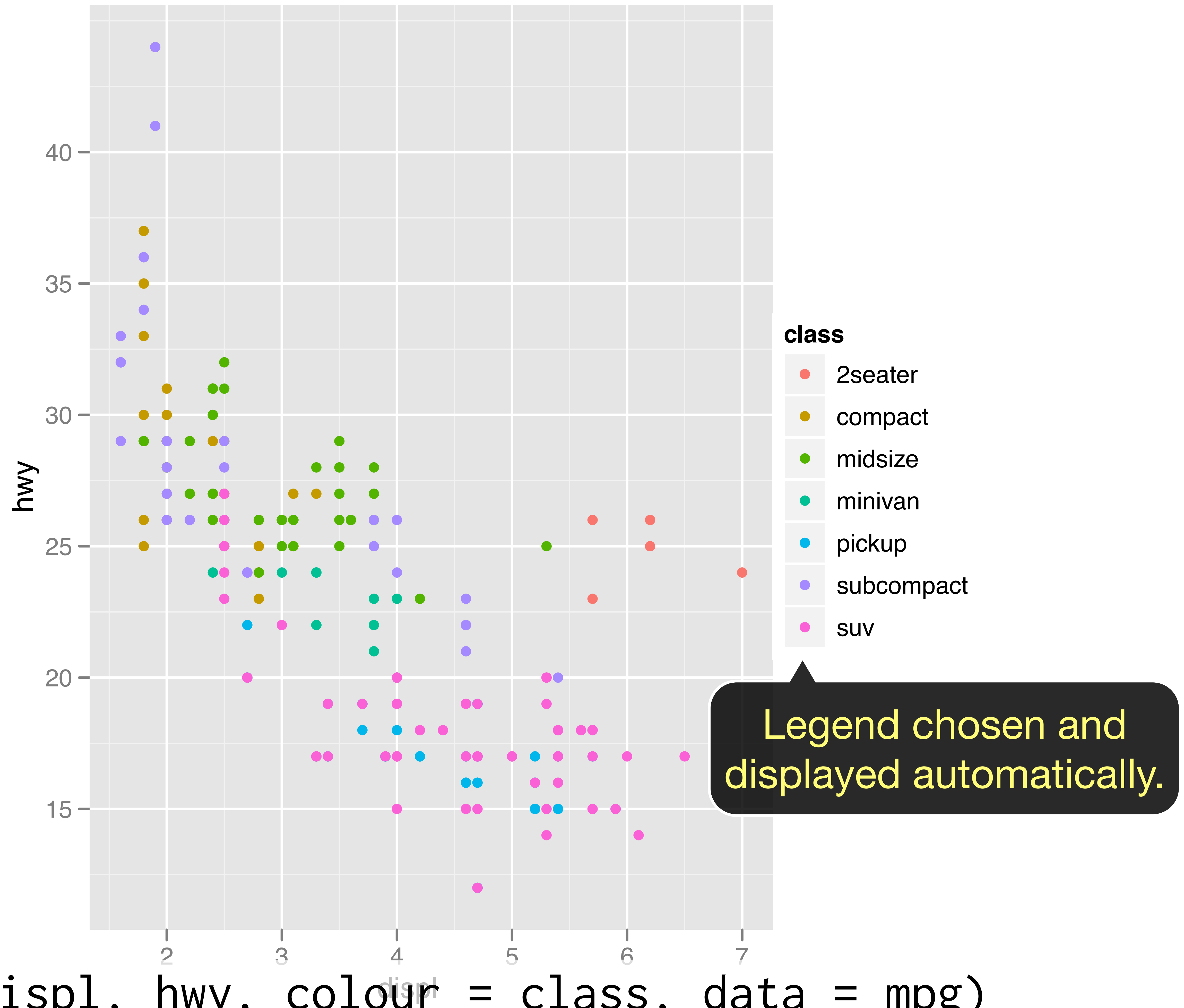
qplot(displ, hwy, data = mpg)

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)
```



Legend chosen and displayed automatically.

```
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	Doesn’t work

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)
```

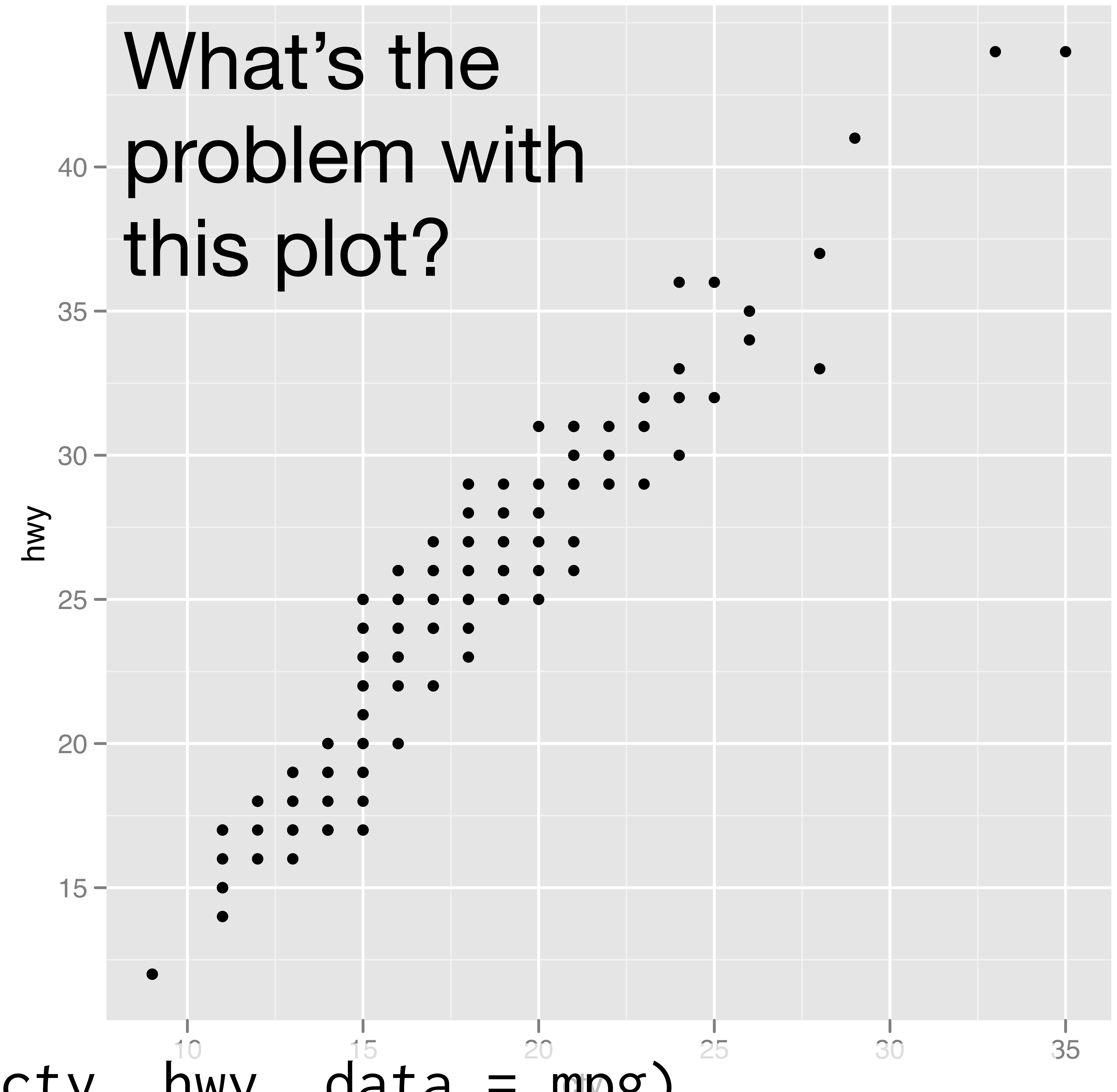
Summary

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

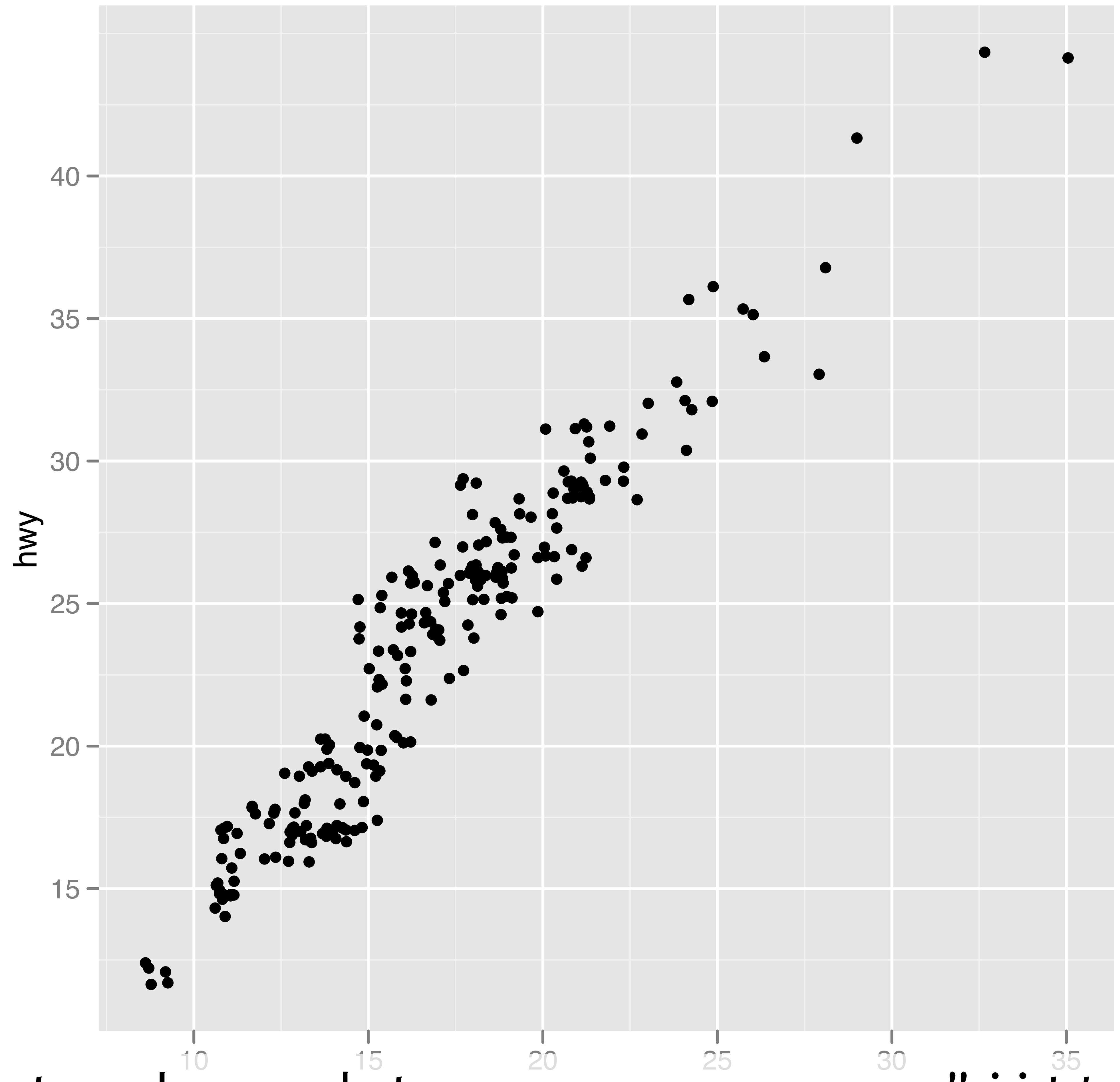
`facet_wrap()`: 1d ribbon wrapped into 2d

Scales argument controls whether position scales are fixed or free.

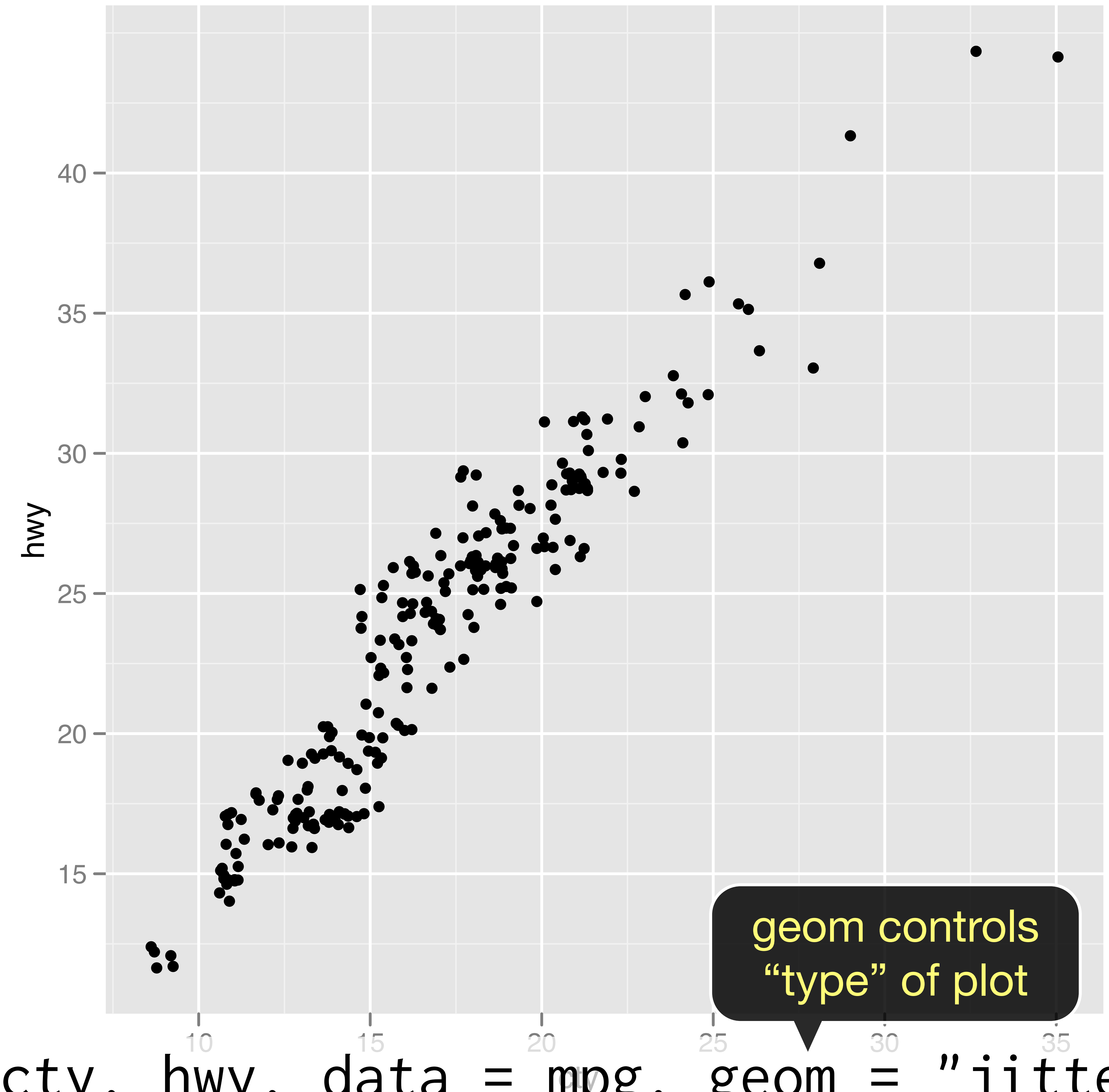
What's the problem with this plot?



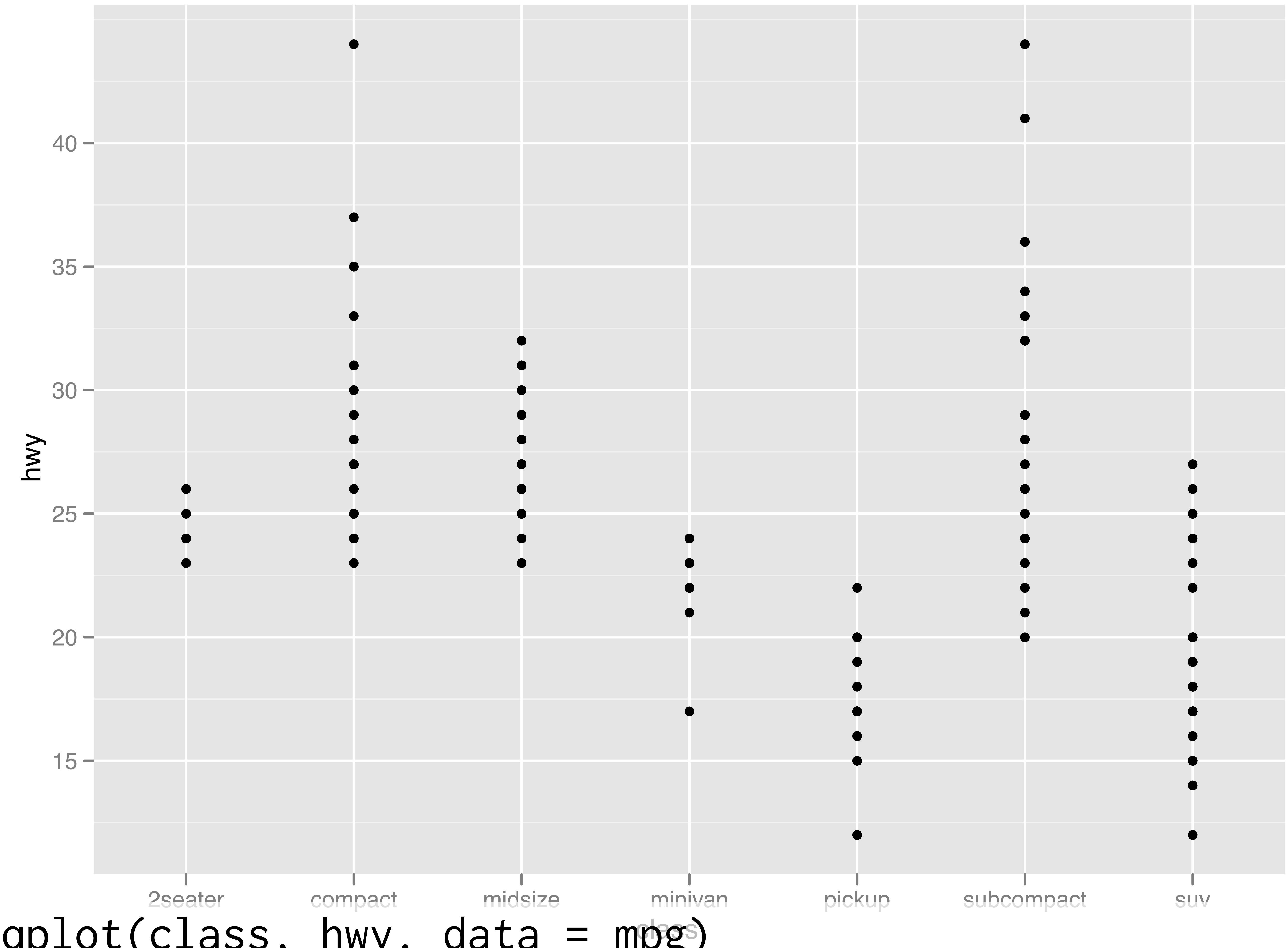
```
qplot(cty, hwy, data = mpg)
```



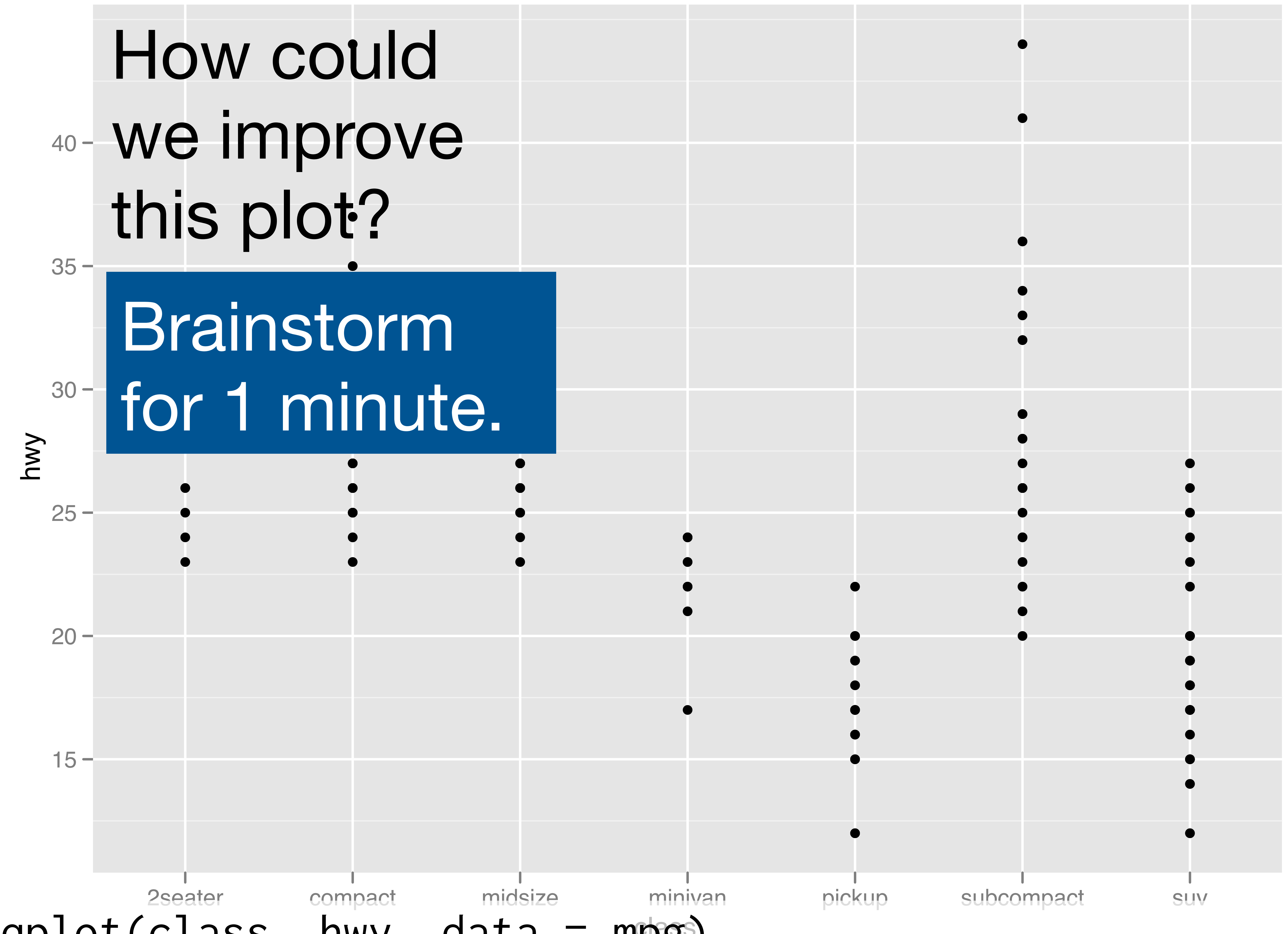
```
qplot(cty, hwy, data = mpg, geom = "jitter")
```



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



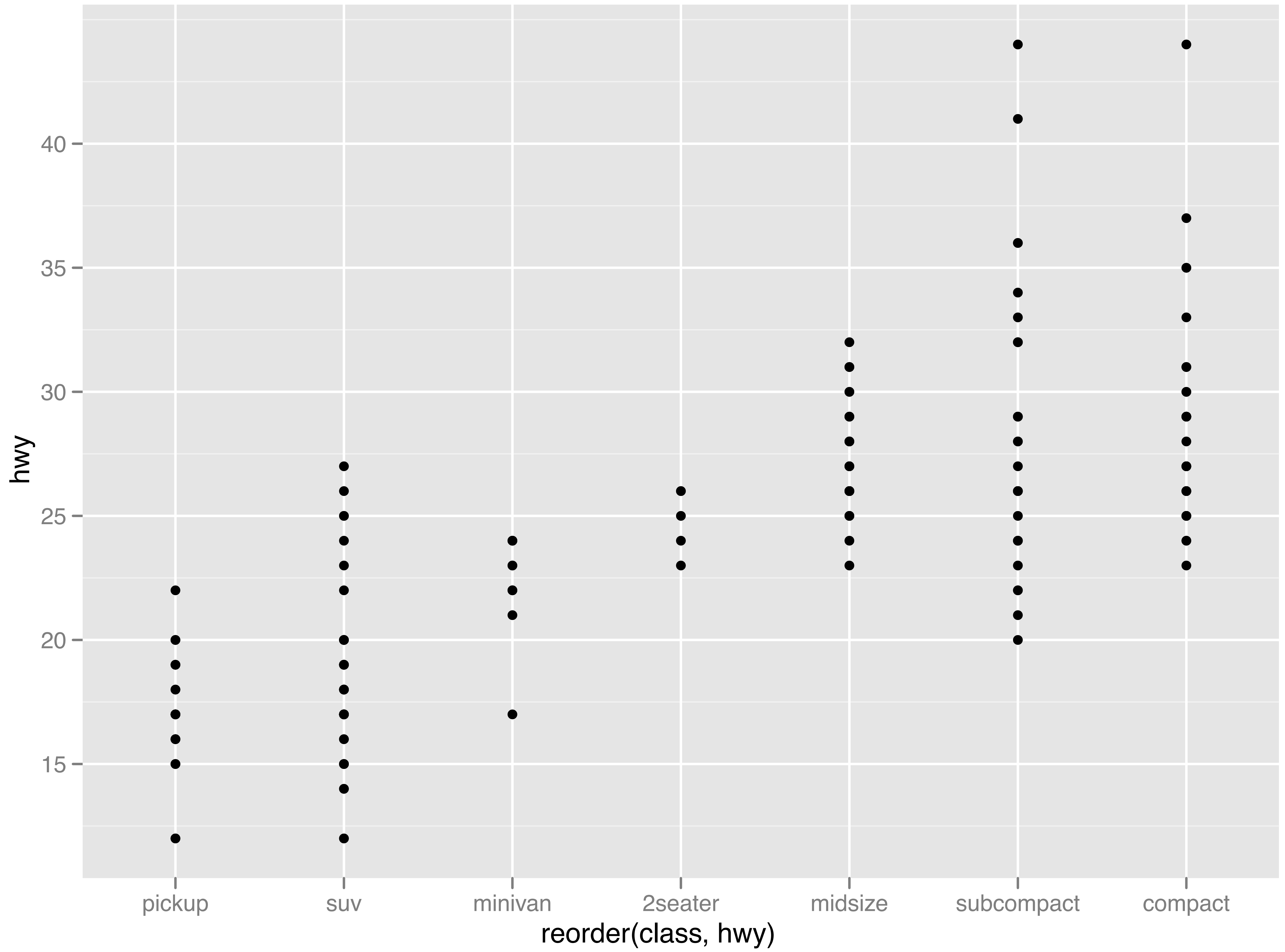
```
qplot(class, hwy, data = mpg)
```



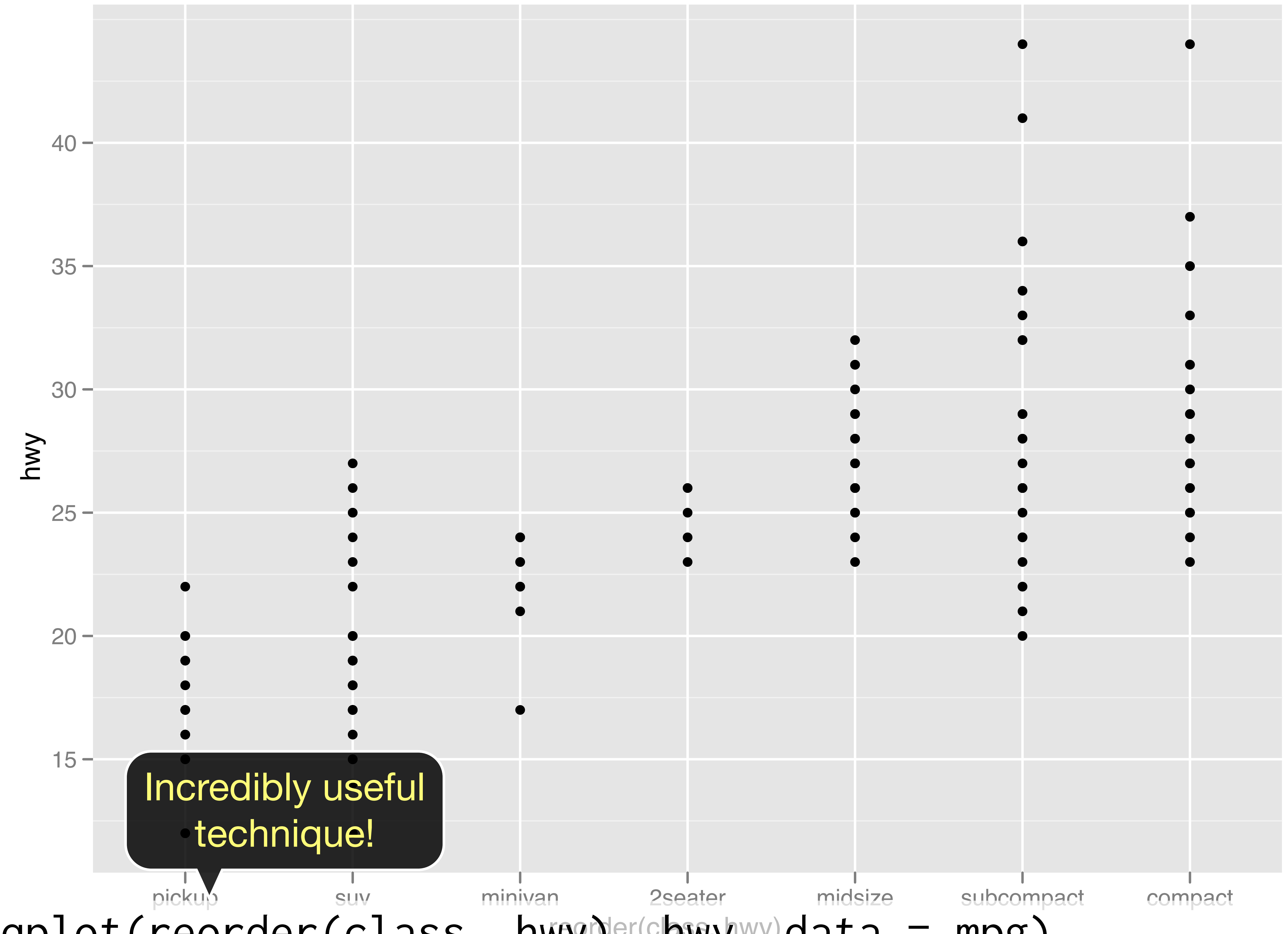
How could we improve this plot?

Brainstorm for 1 minute.

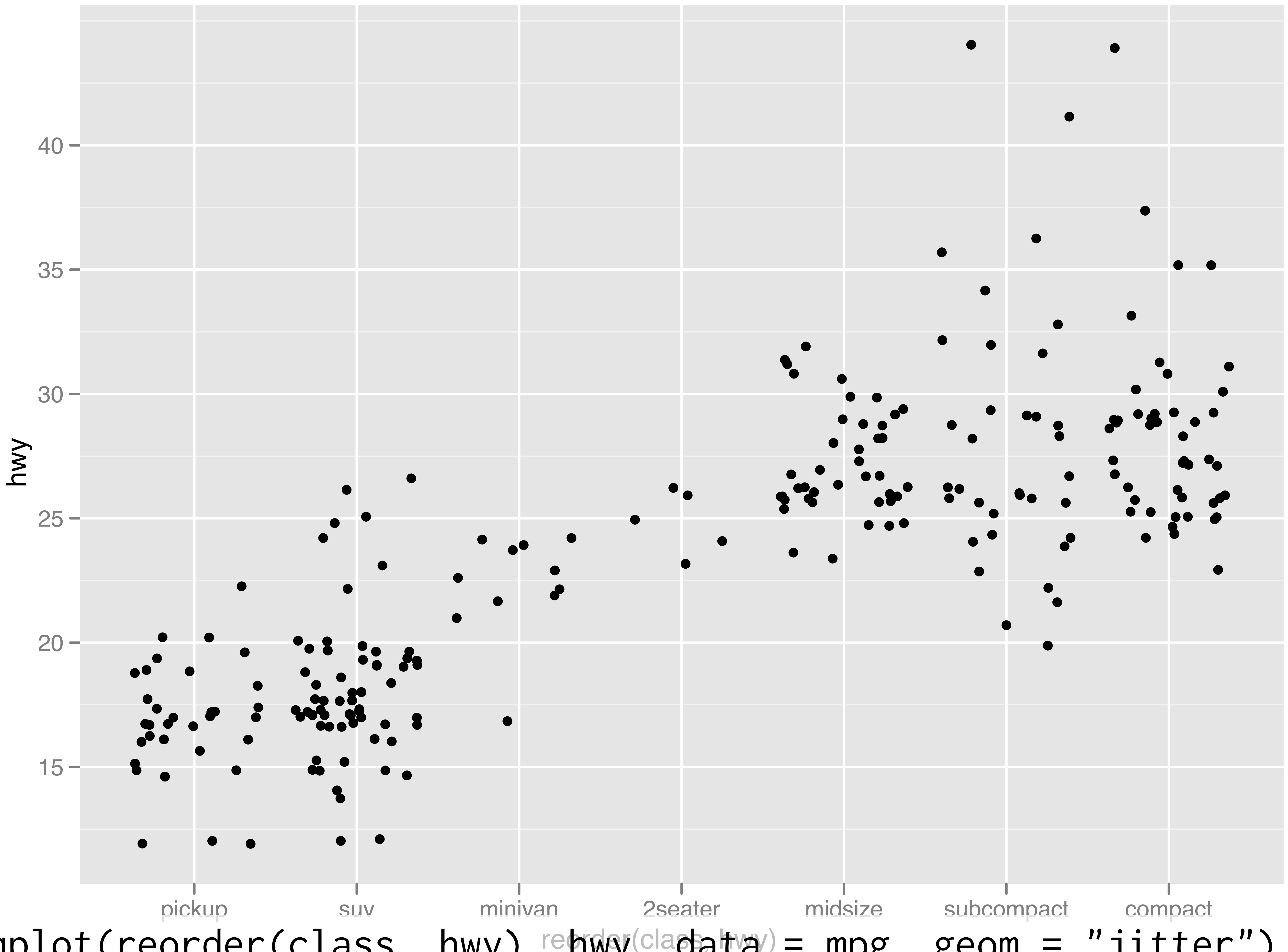
```
qplot(class, hwy, data = mpg)
```



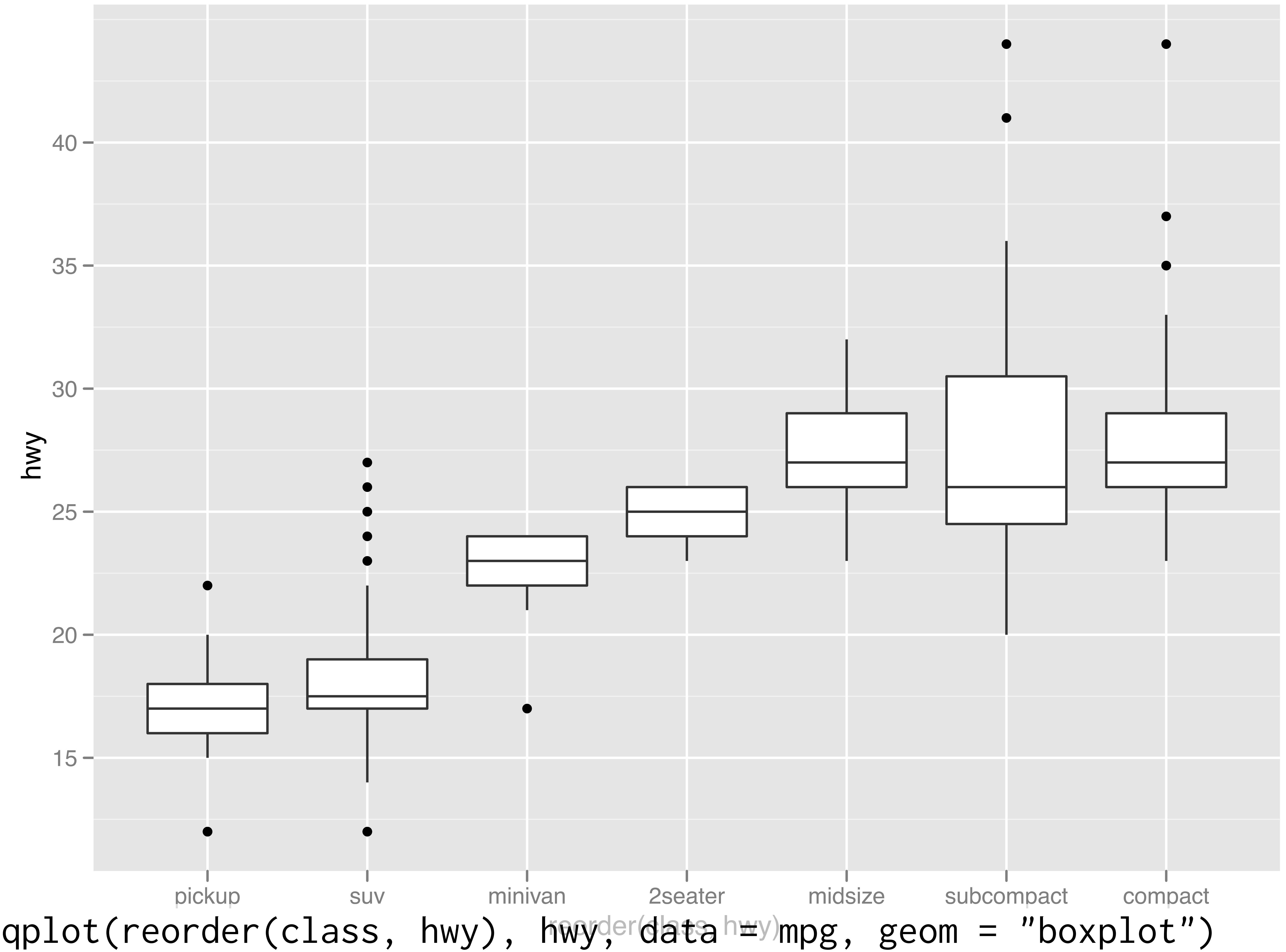
hwy
40
35
30
25
20
15
pickup
suv
minivan
2seater
midsize
subcompact
compact
reorder(class, hwy)



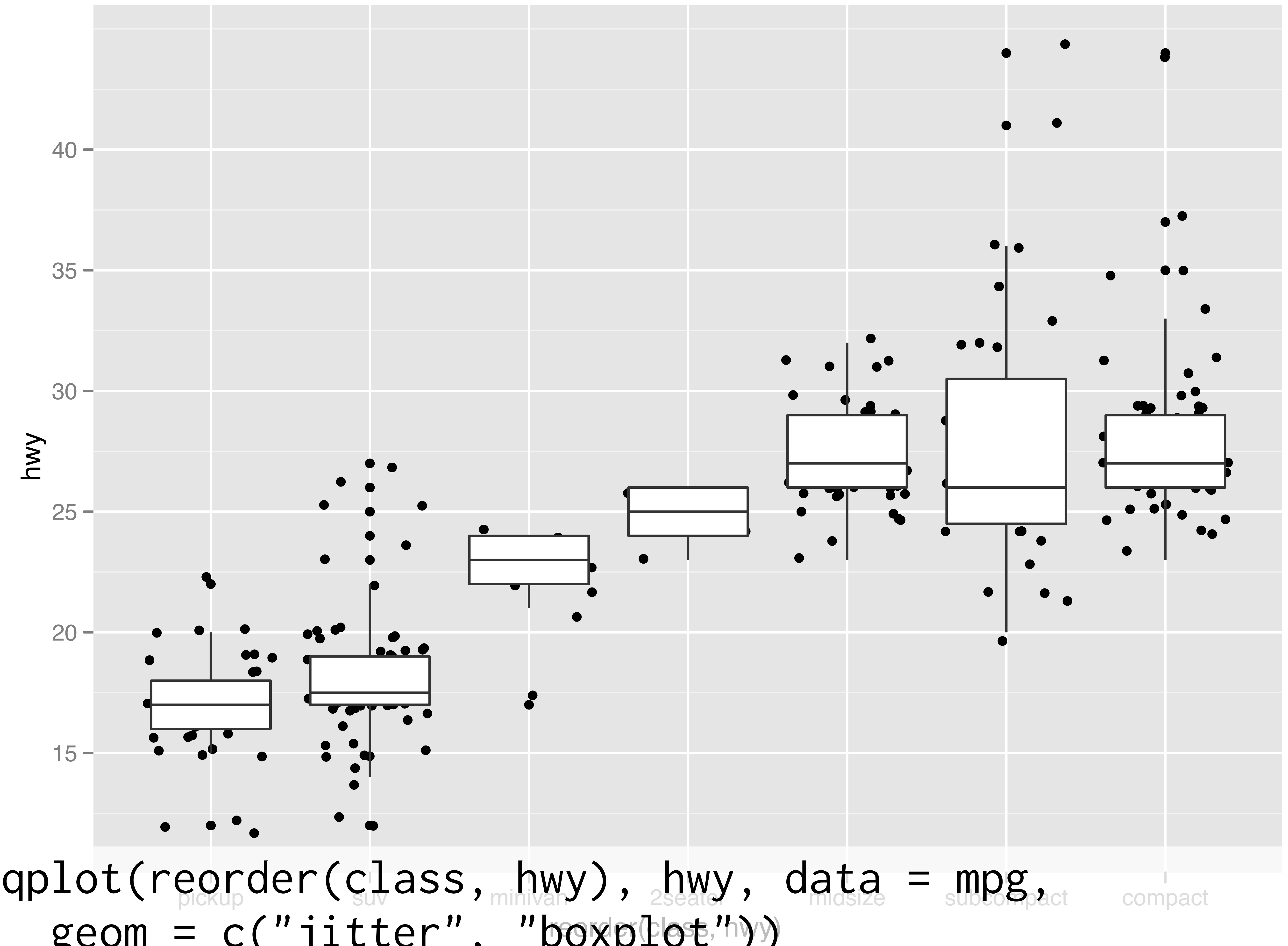
```
qplot(reorder(class, hwy), hwy, data = mpg)
```



```
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 previously plots with class ordered by median hwy.

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

Aside: 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)