

# Introduction

**Hadley Wickham**

**October 2009**



[http://lookinatdata.com/  
ismi-2009](http://lookinatdata.com/ismi-2009)

1. Preview of today

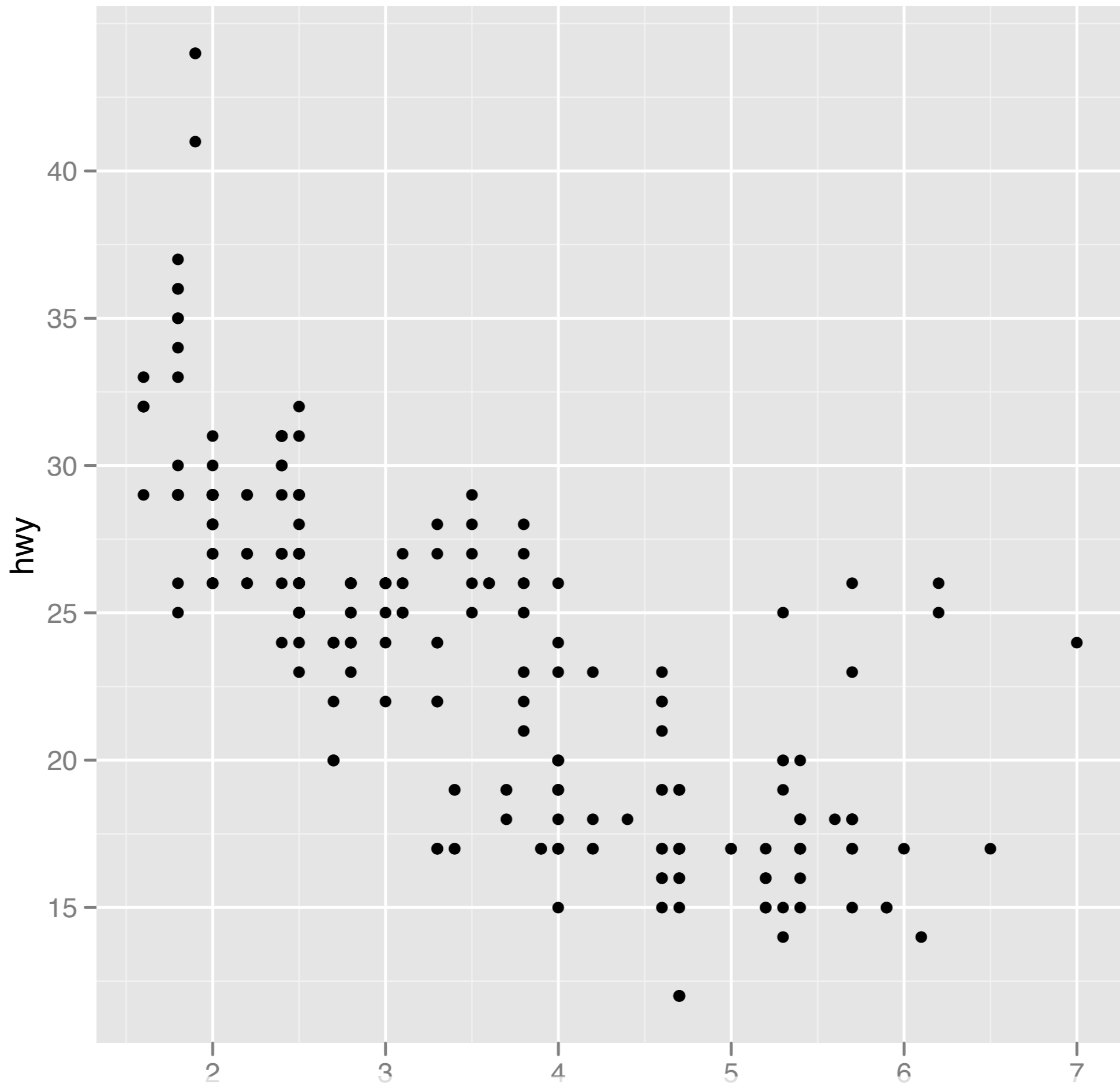
2. About ggplot2

3. More resources

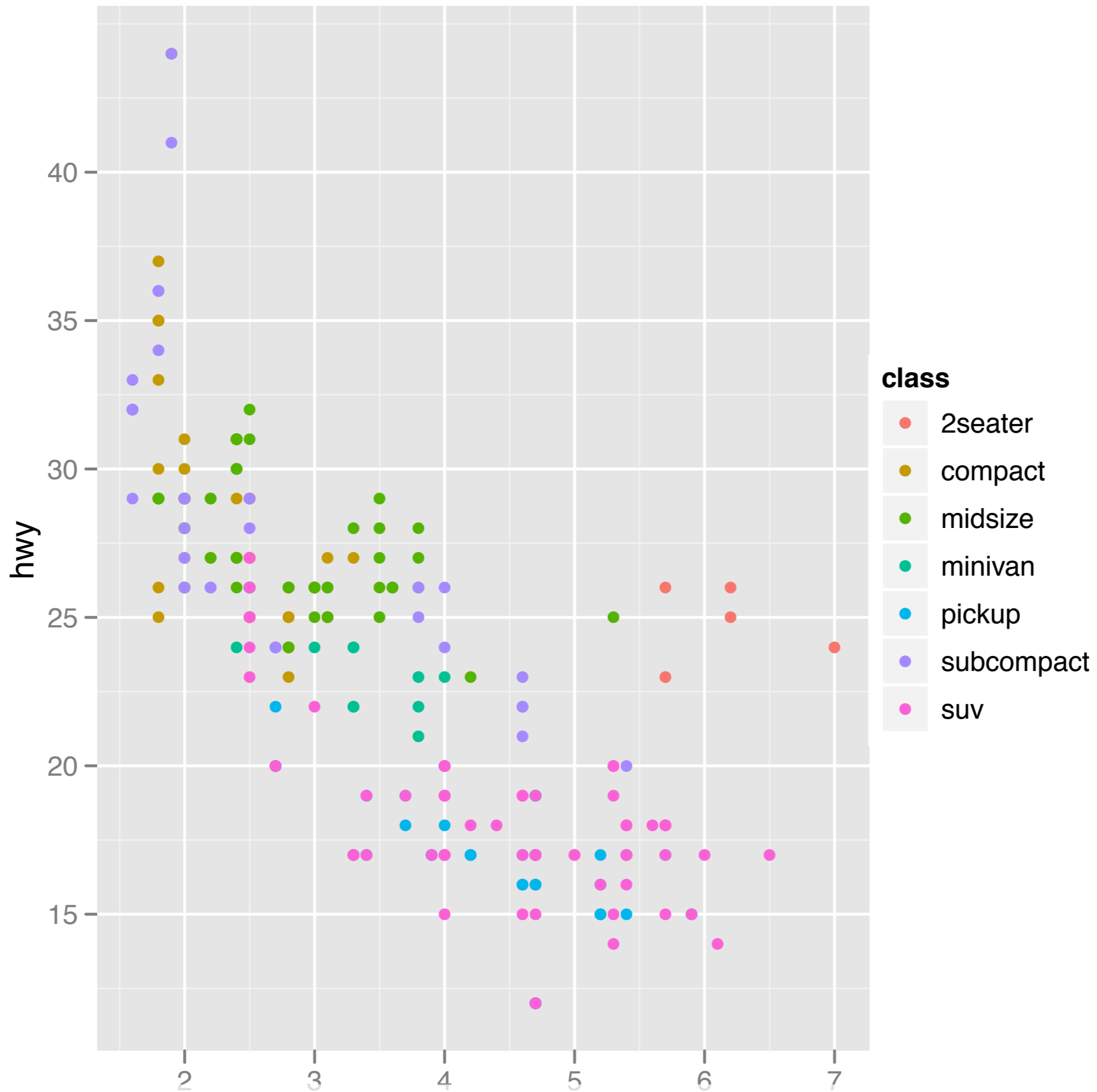
4. Diving in

# Fuel economy

## Basic graphics



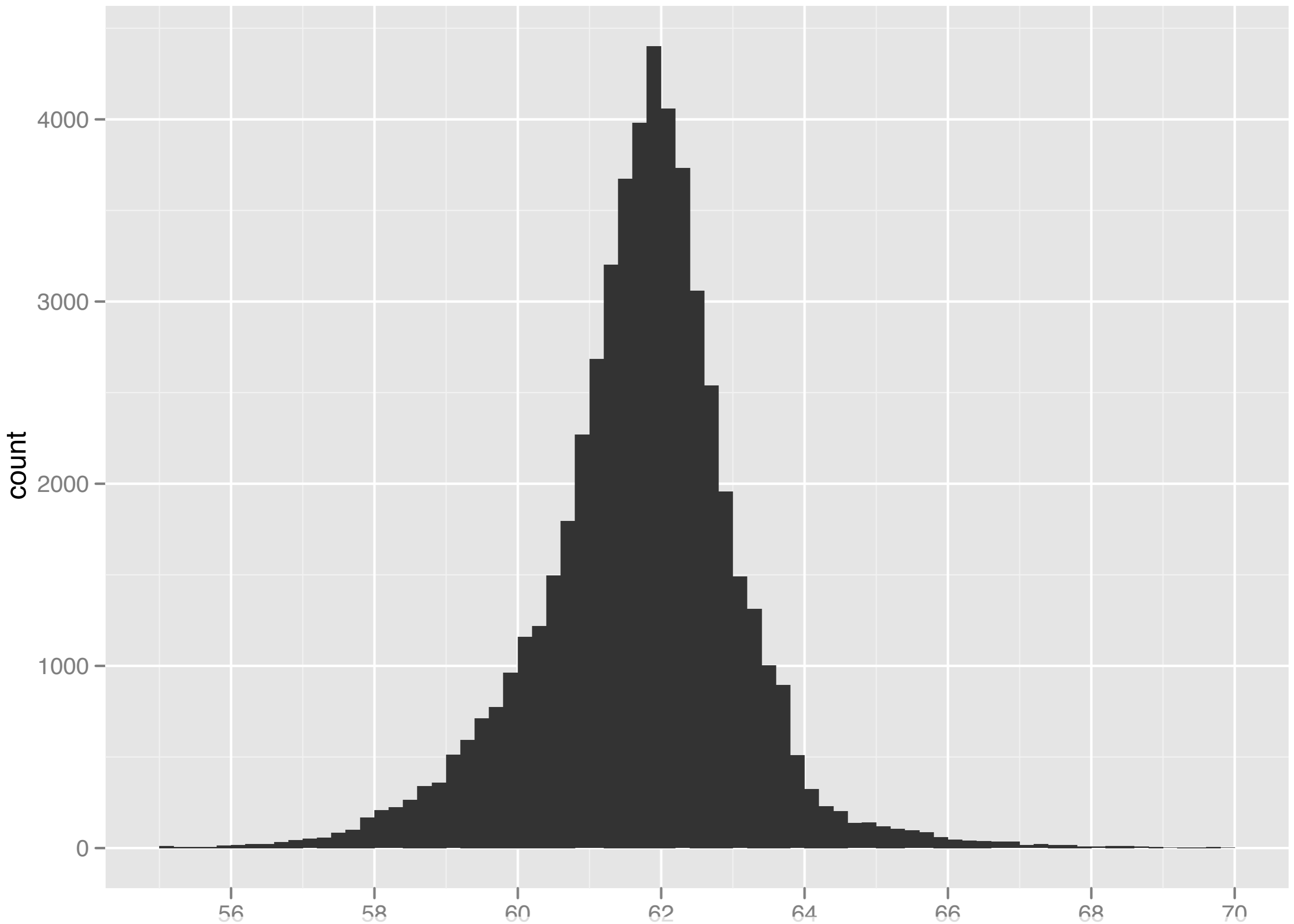
```
qplot(displ, hwy, data = mpg)
```



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

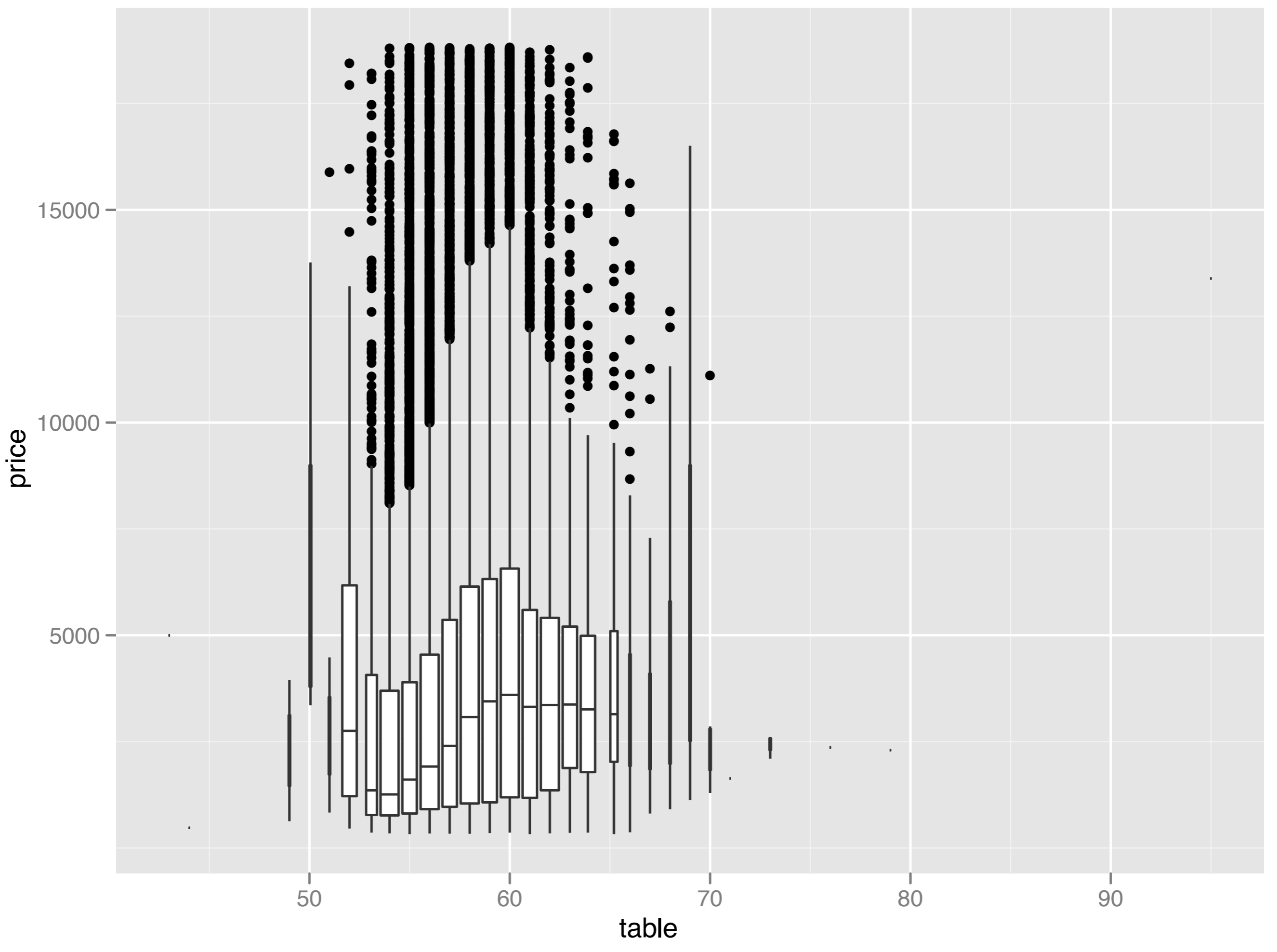
# **Diamond prices**

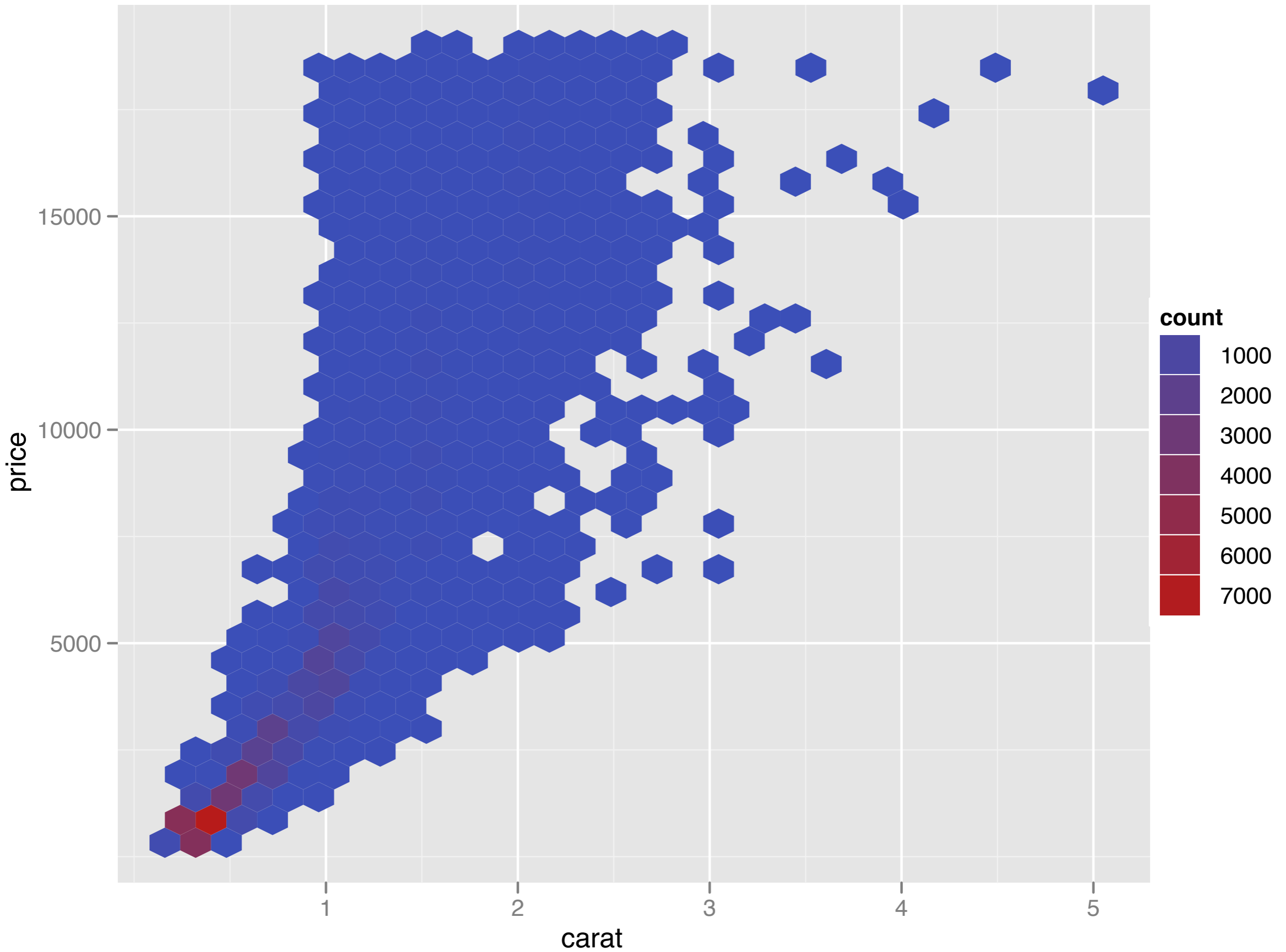
## Displaying large data



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

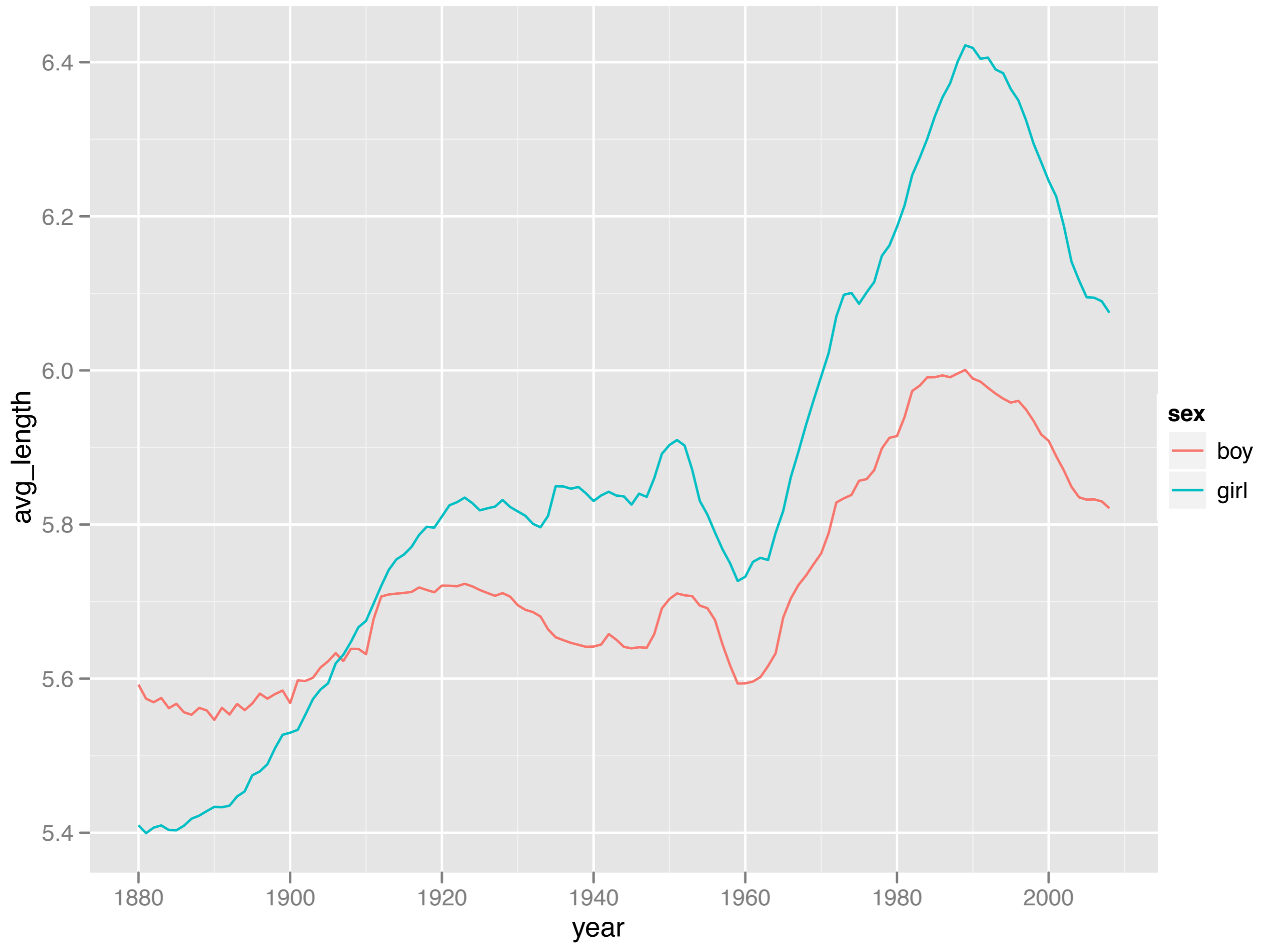


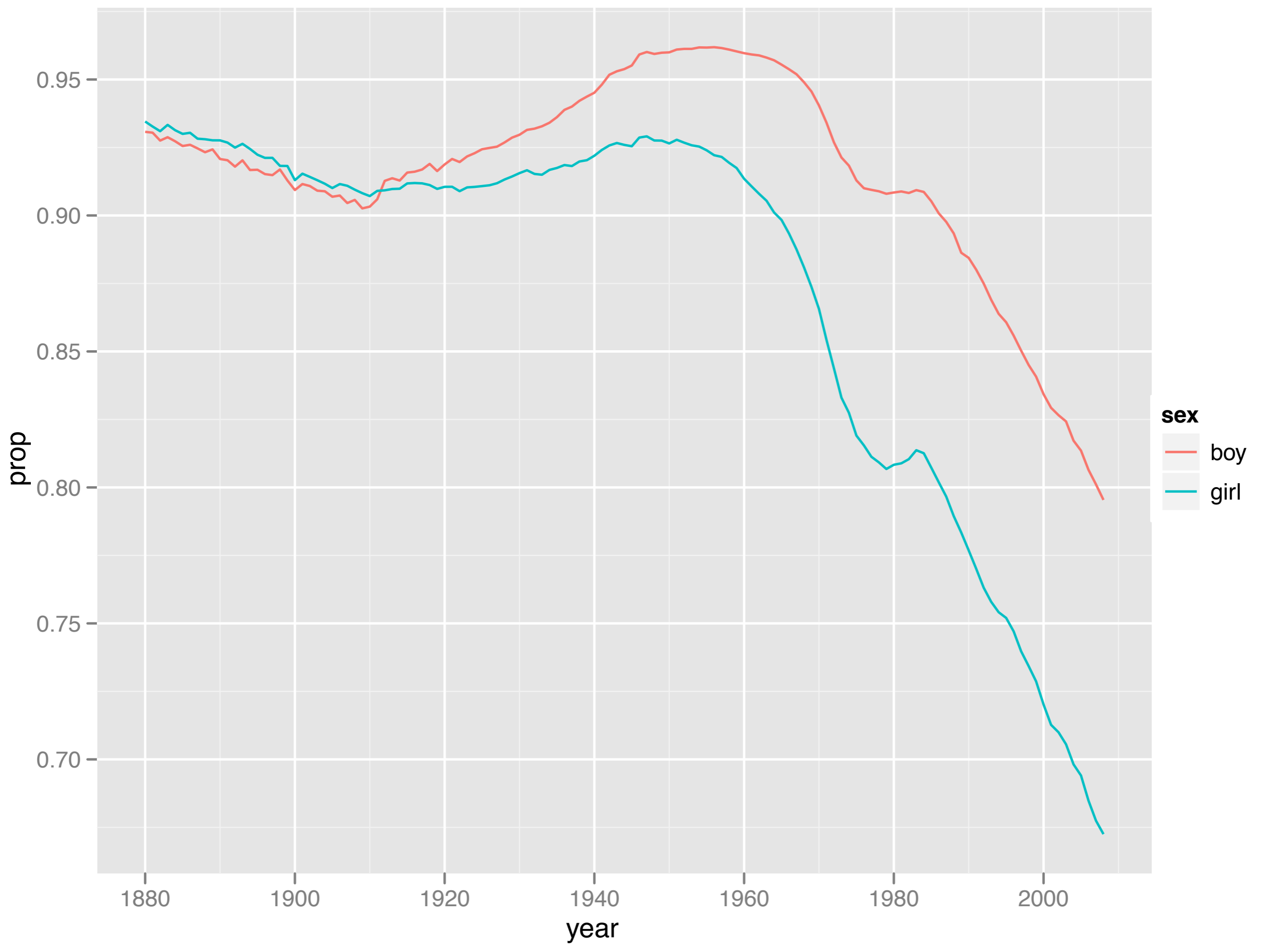




# US baby names

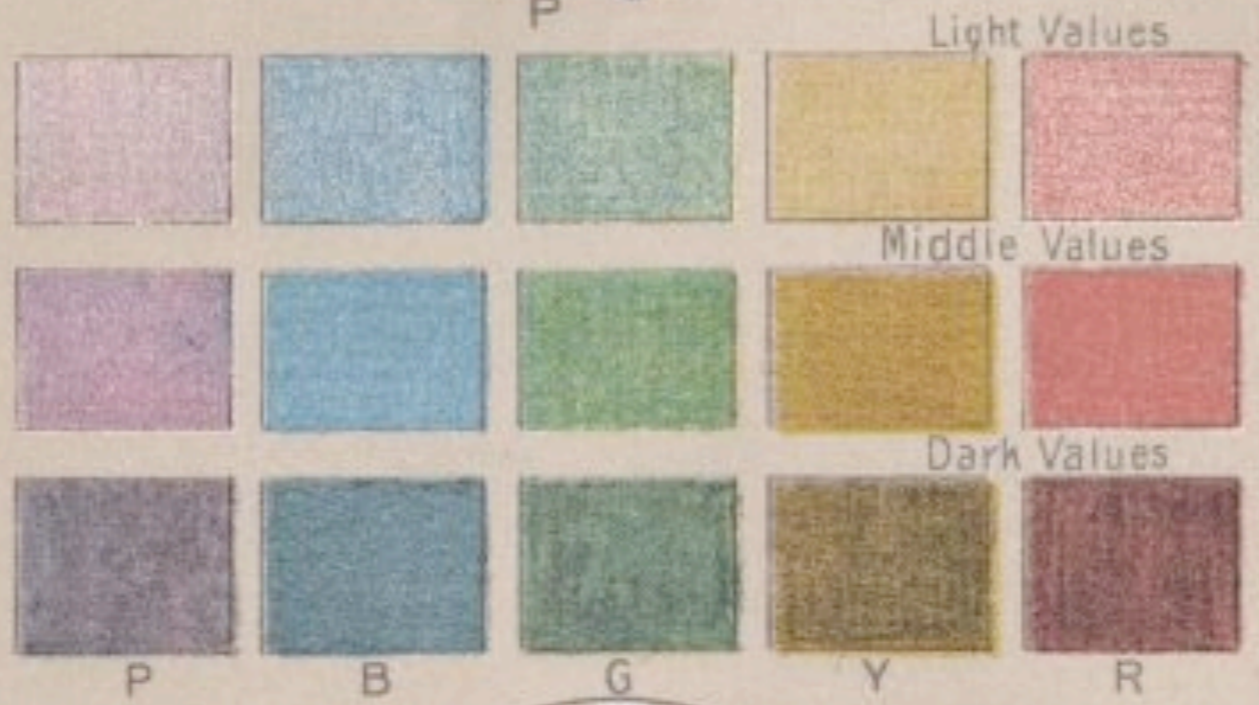
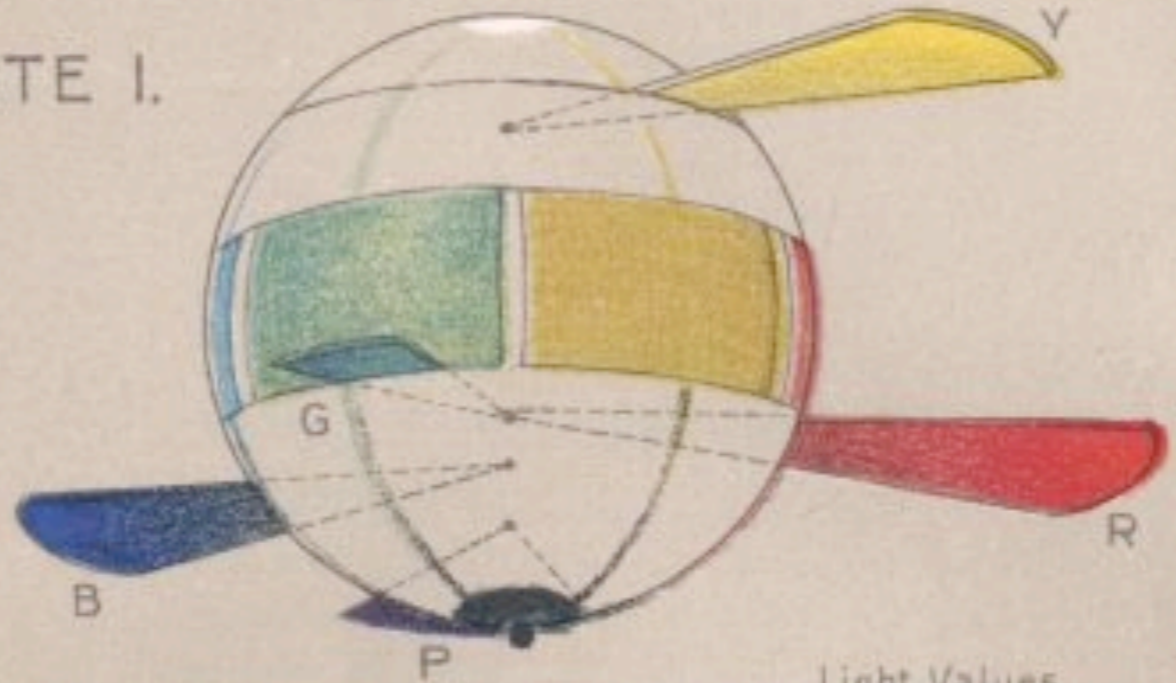
## Transforming data





# Polishing your plots

PLATE I.



Copyright 1907 by A. H. Munsell.

1. Saving your work

2. Labels & ticks

3. Themes



# 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>

# 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
```

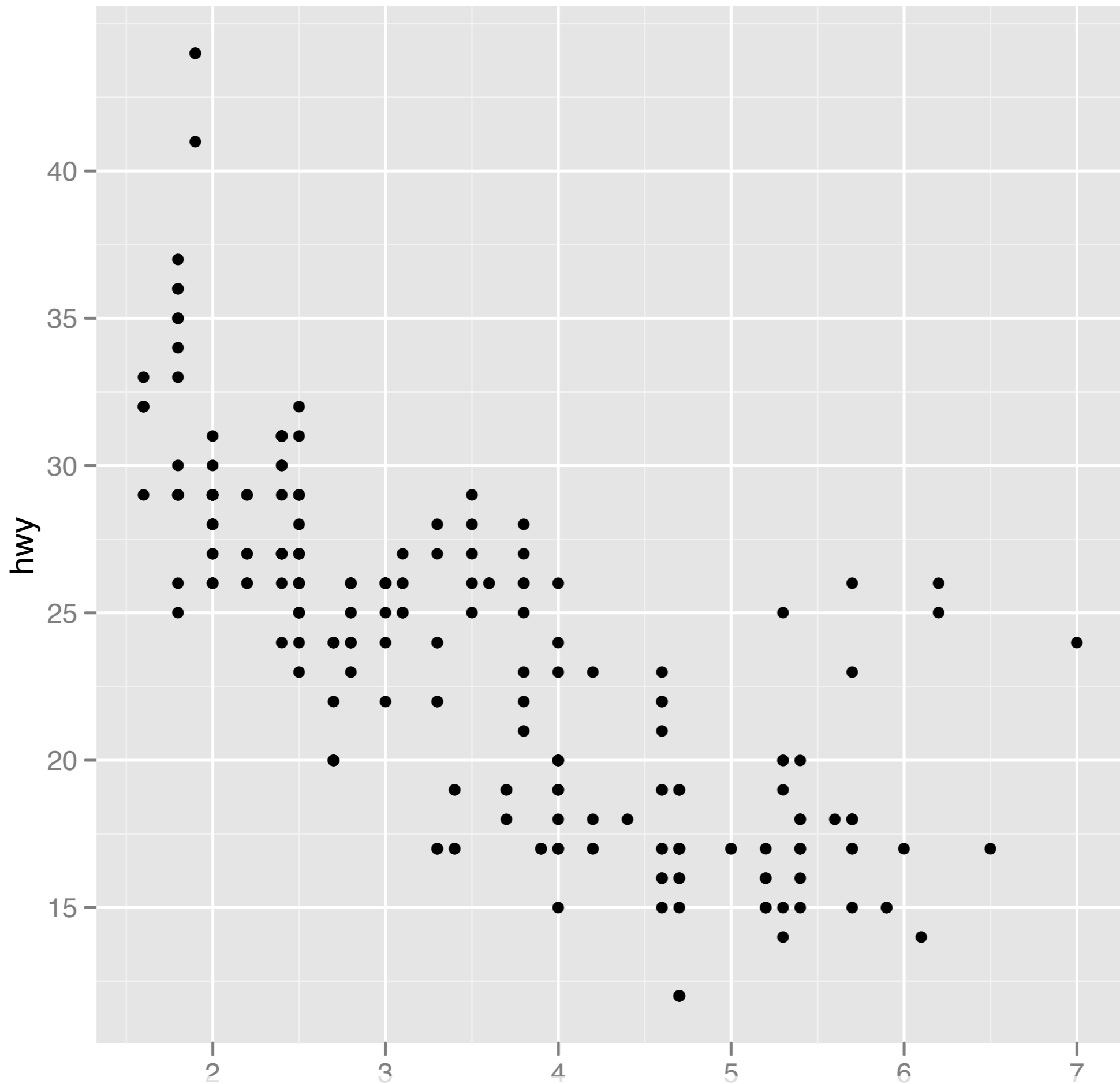
```
head(mpg)
```

```
str(mpg)
```

```
summary(mpg)
```

In ggplot2, we  
always explicitly  
specify the data

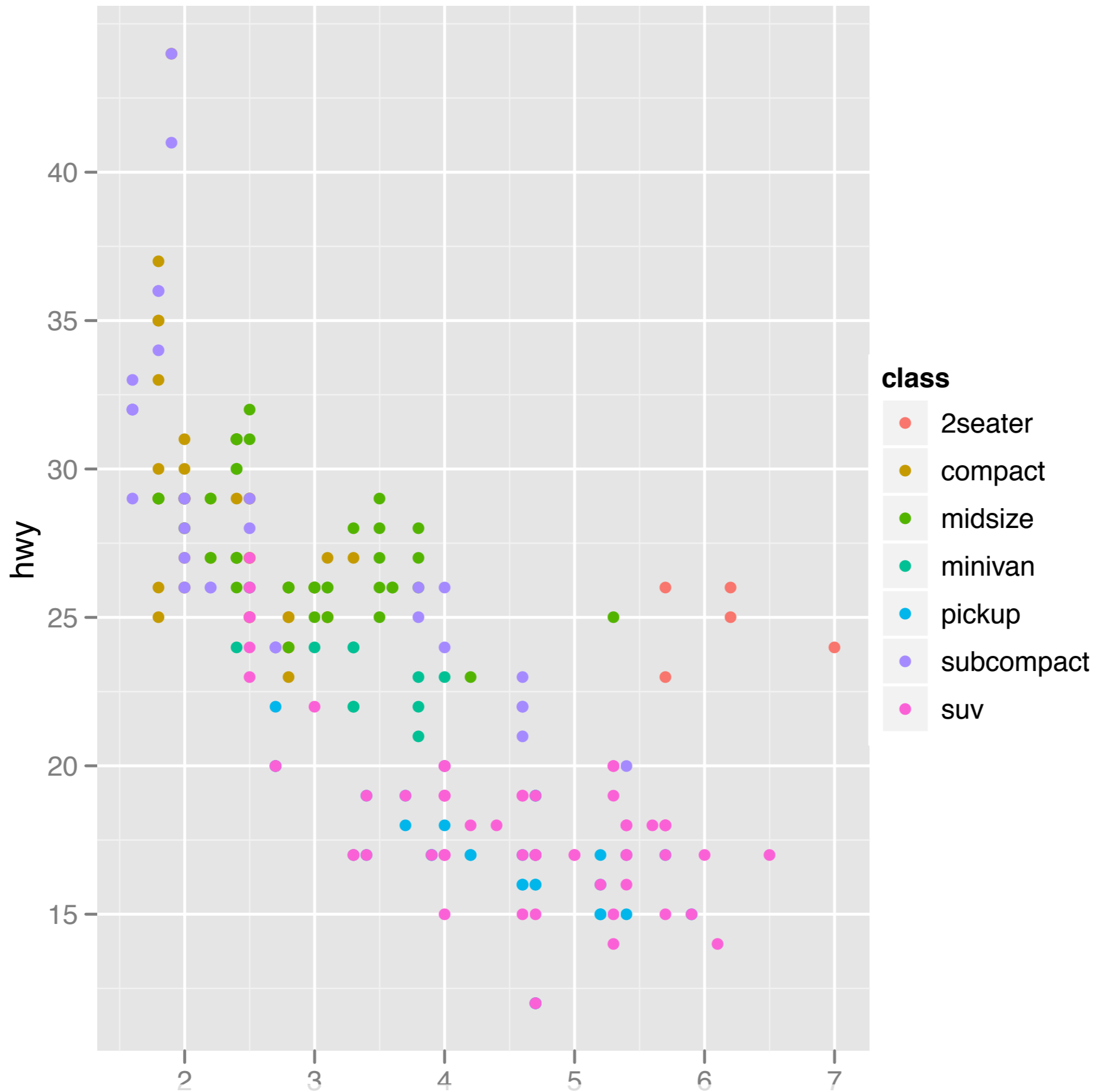
```
qplot(displ, hwy, data = mpg)
```



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

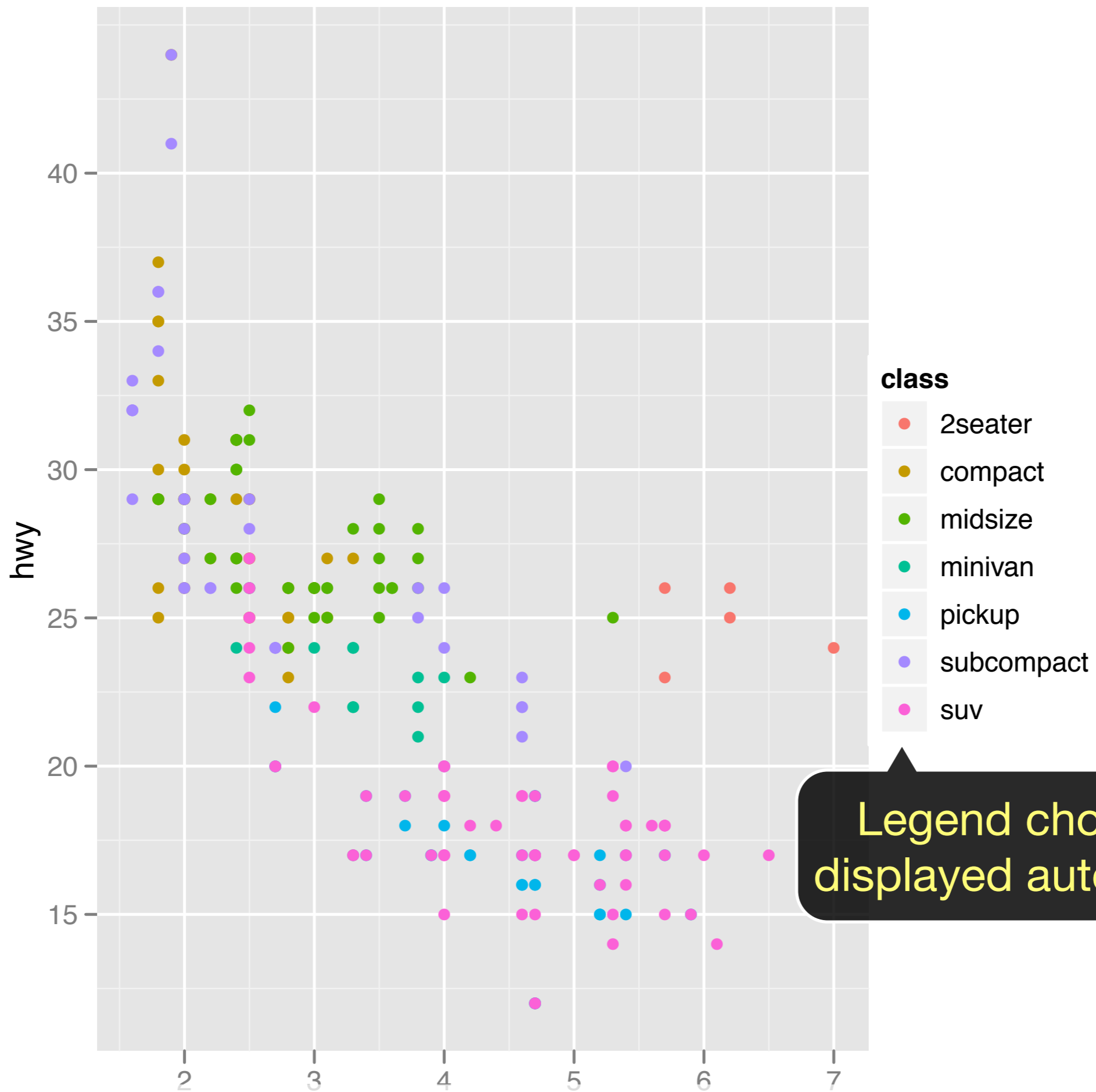
# Additional variables

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)
```

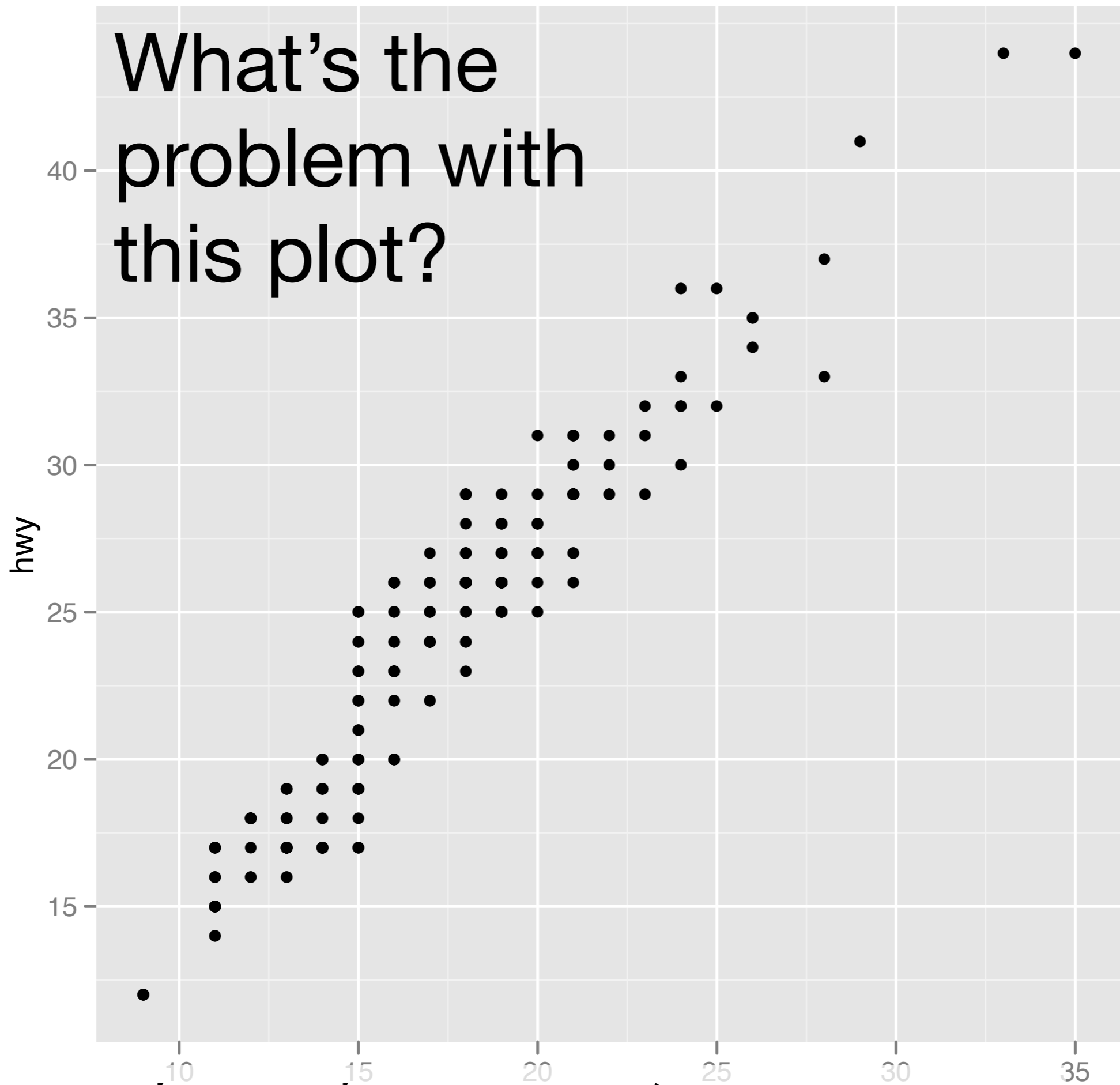
# Summary

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

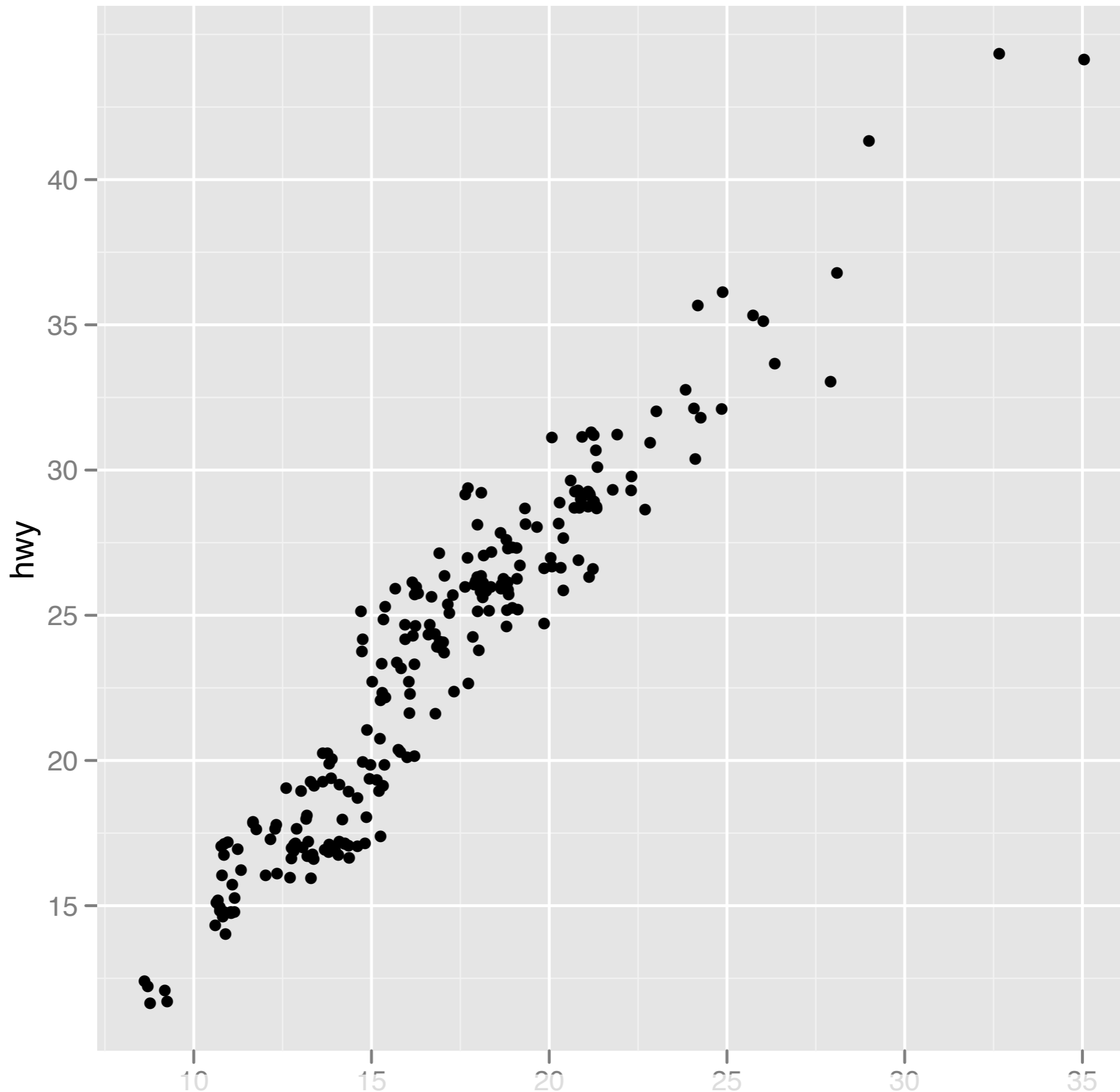
`facet_wrap()`: 1d ribbon wrapped into 2d

`Scales` argument controls whether position scales are fixed or free.

What's the  
problem with  
this plot?

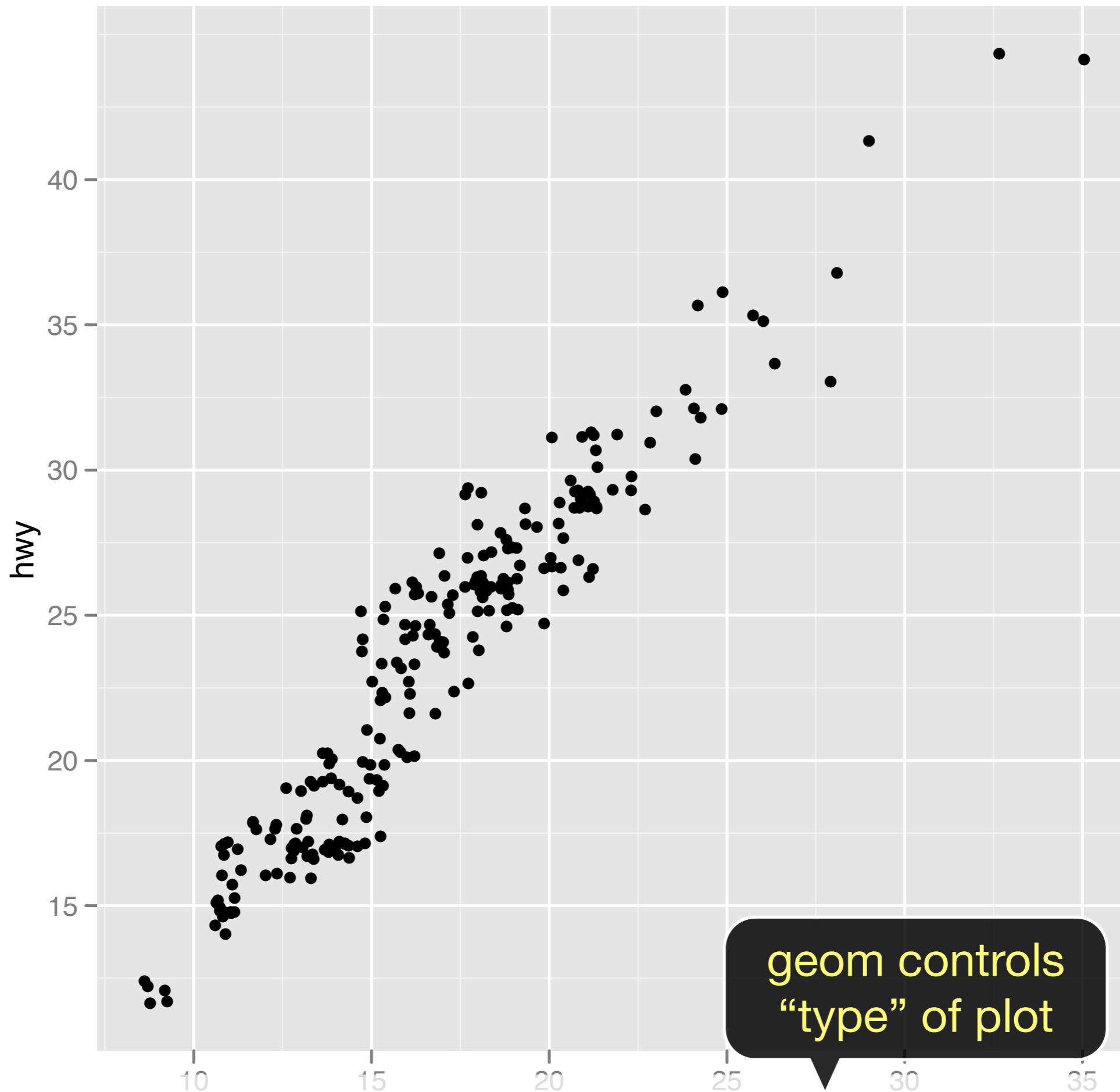


```
qplot(cty, hwy, data = mpg)
```

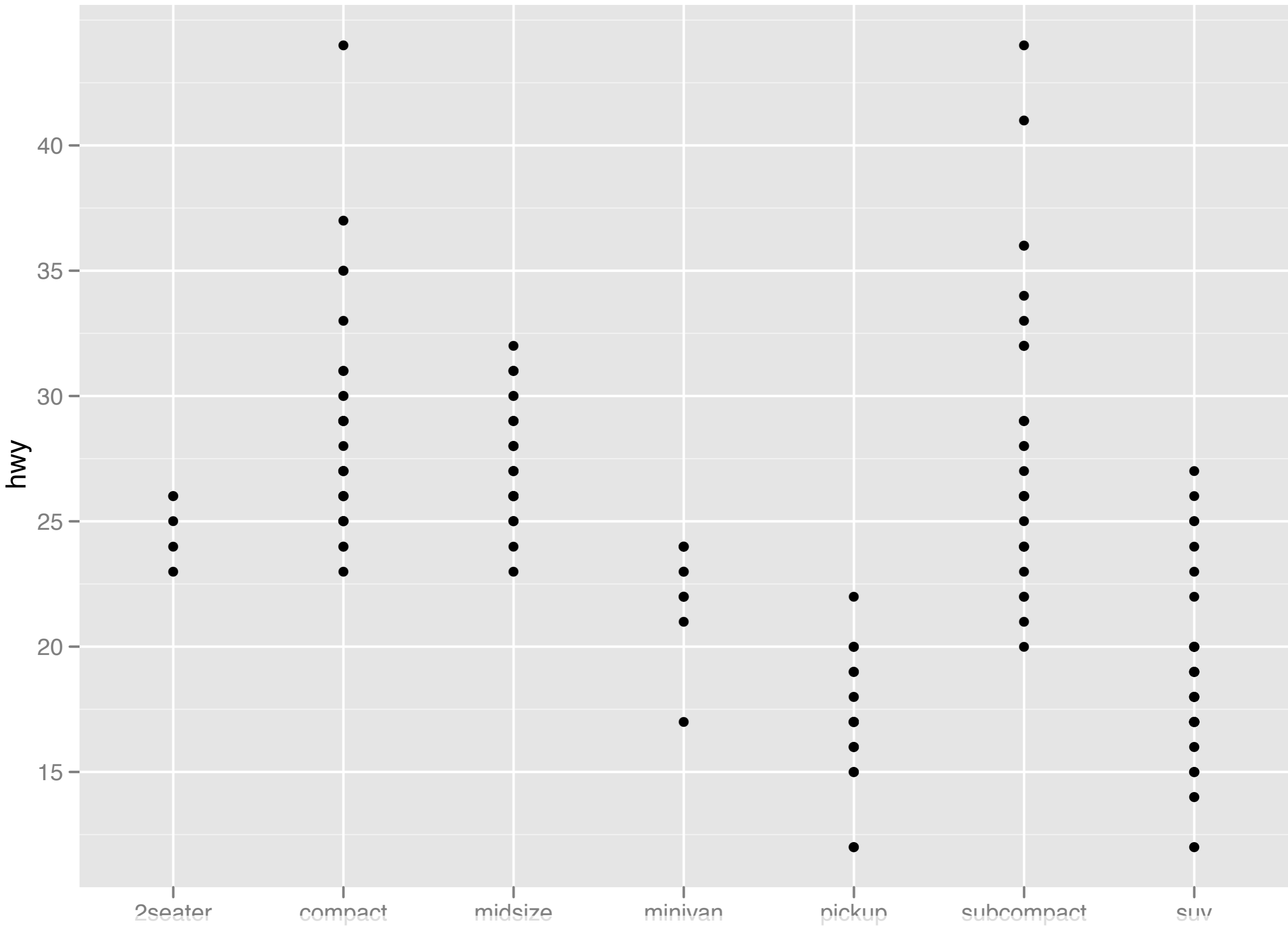


```
qplot(cty, hwy, data = mpg, geom = "jitter")
```





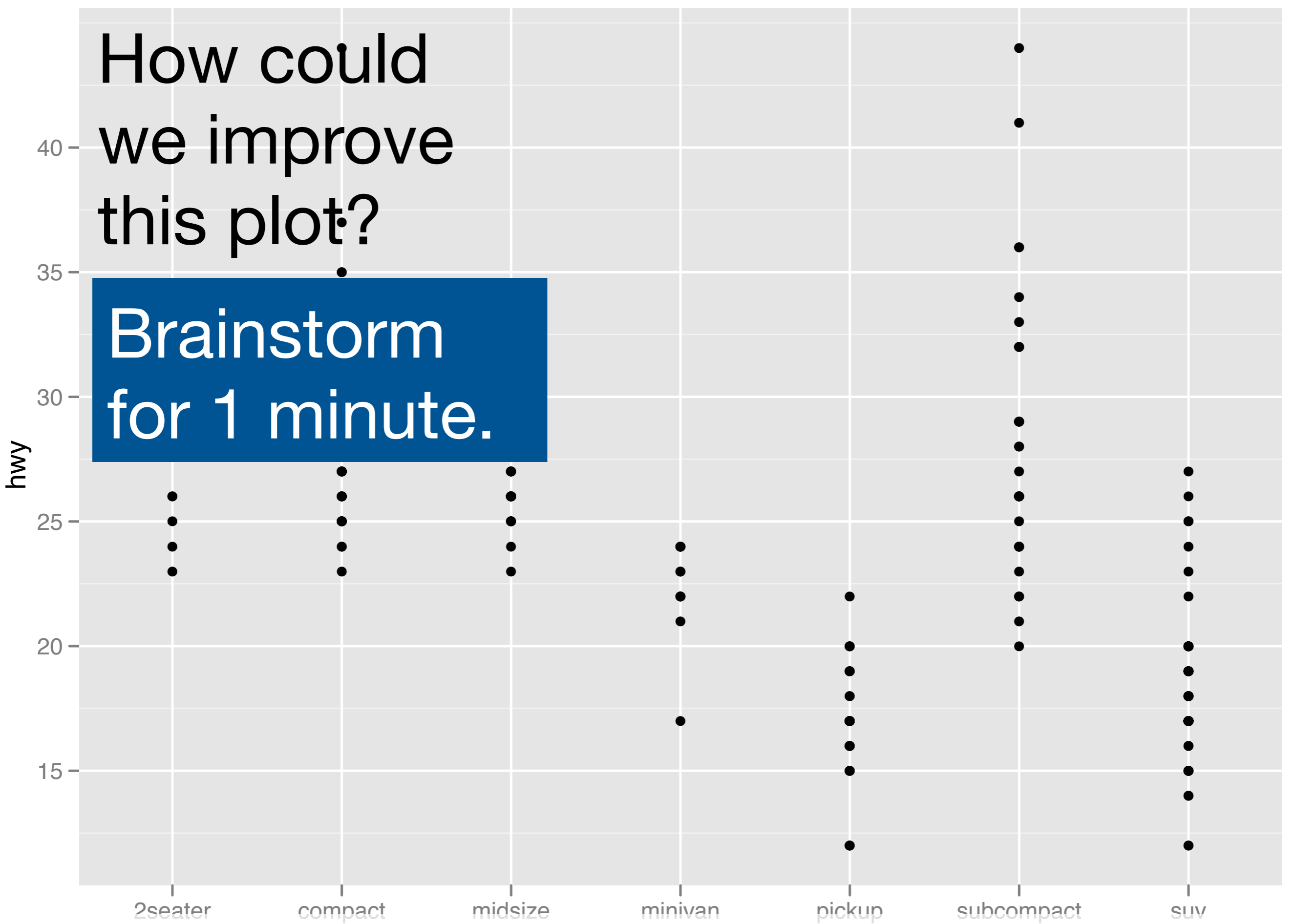
```
qplot(cty, hwy, data = mpg, geom = "jitter")
```



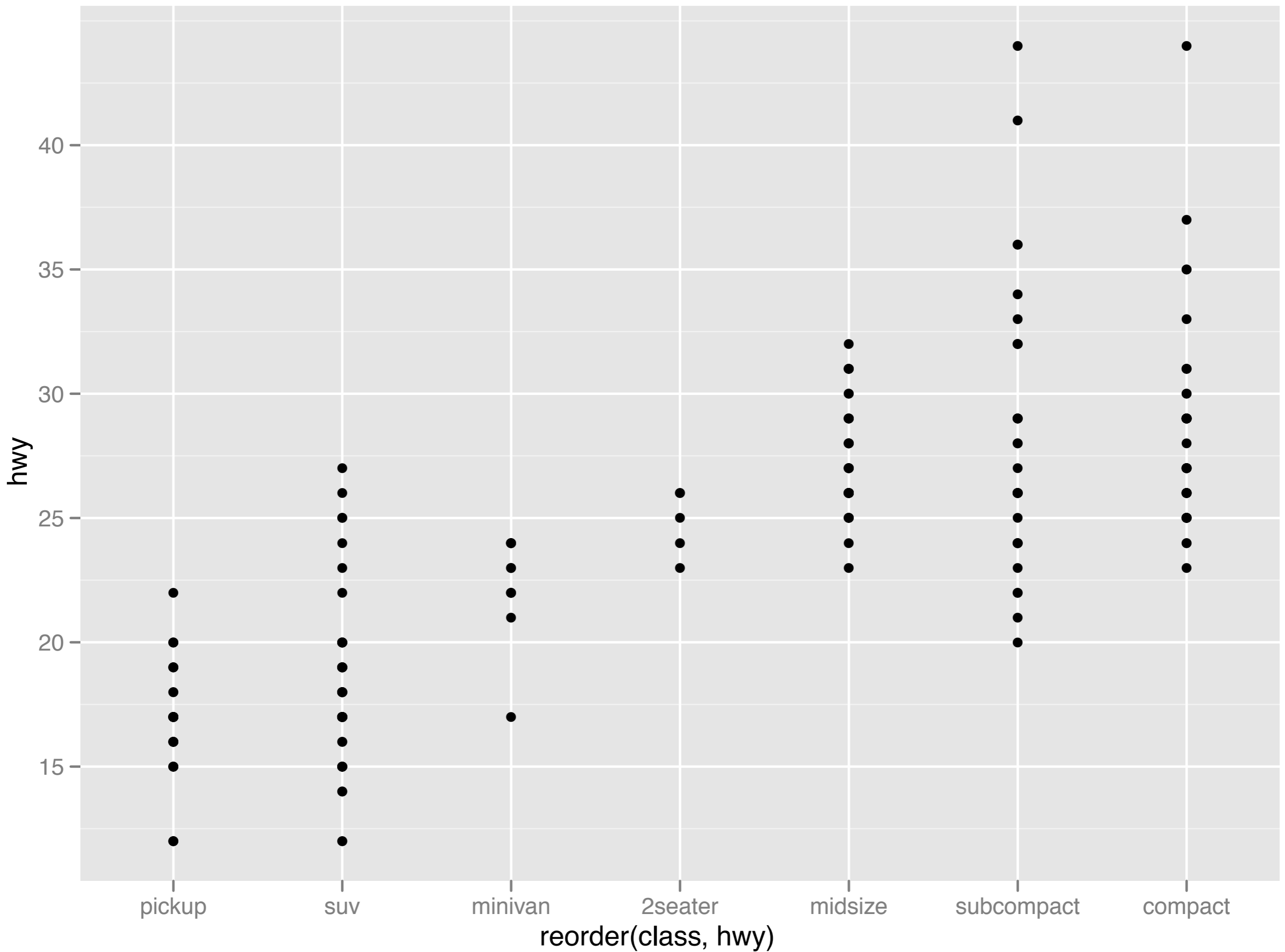
`qplot(class, hwy, data = mpg)`

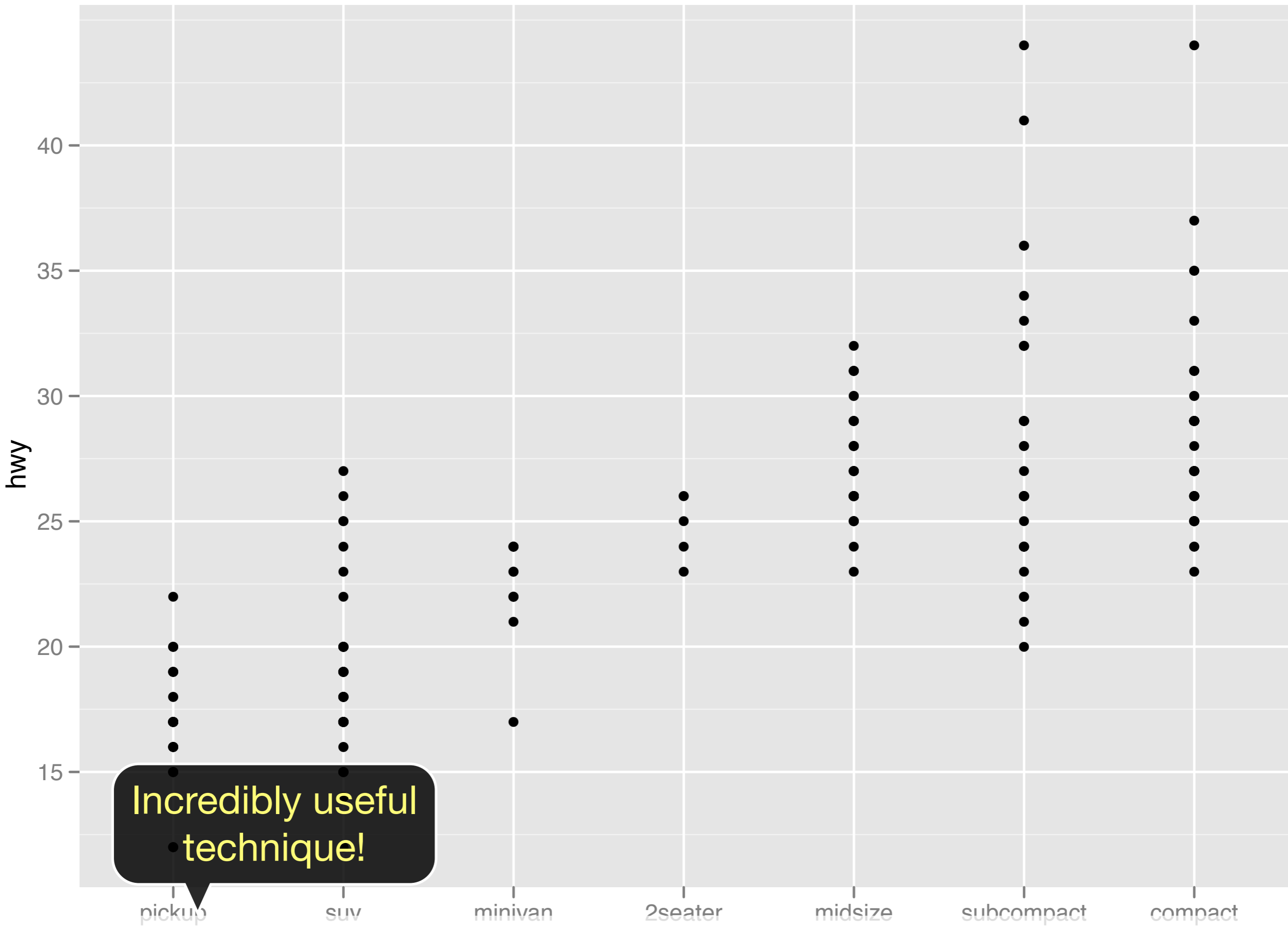
How could we improve this plot?

Brainstorm for 1 minute.



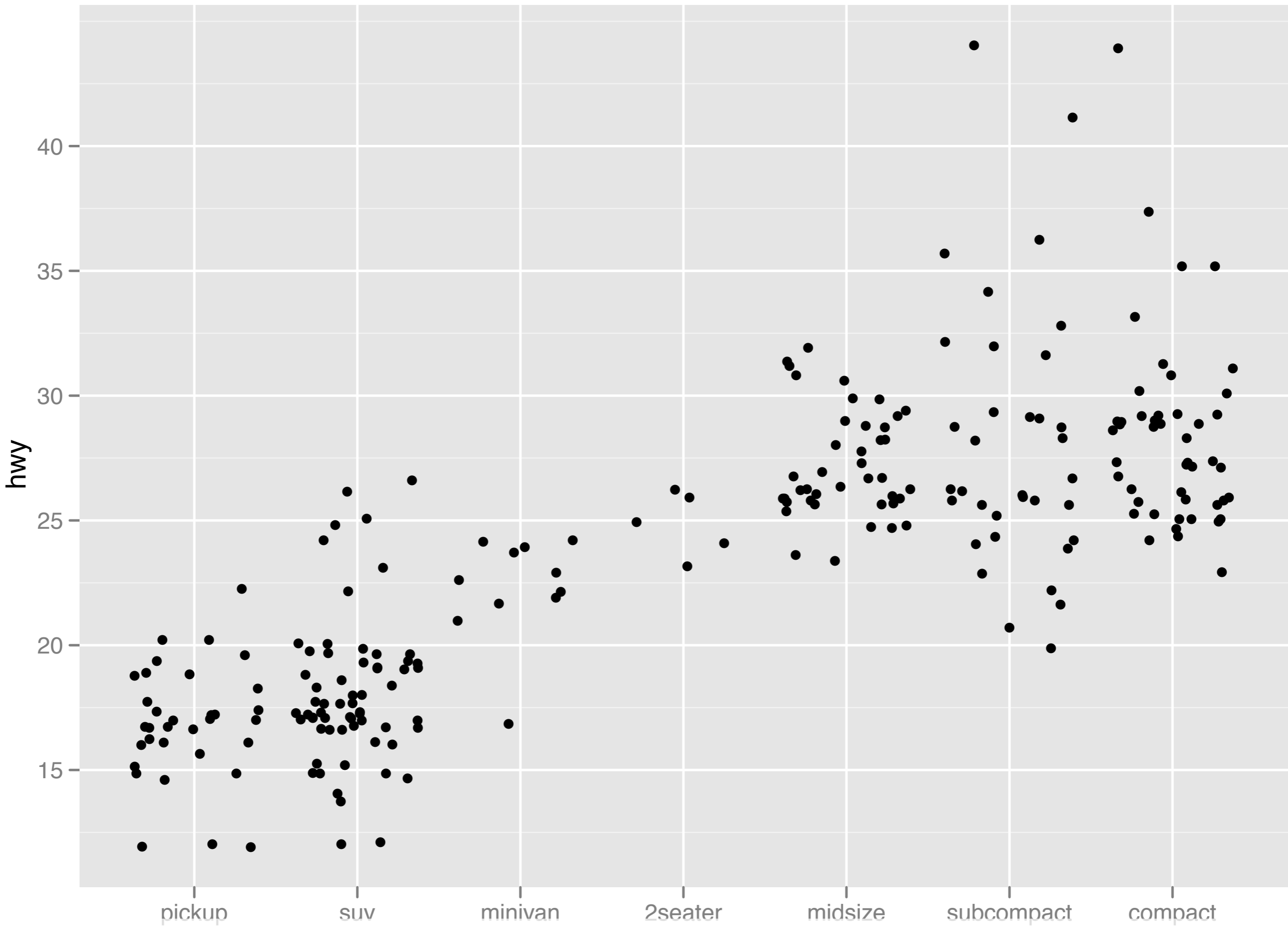
```
qplot(class, hwy, data = mpg)
```



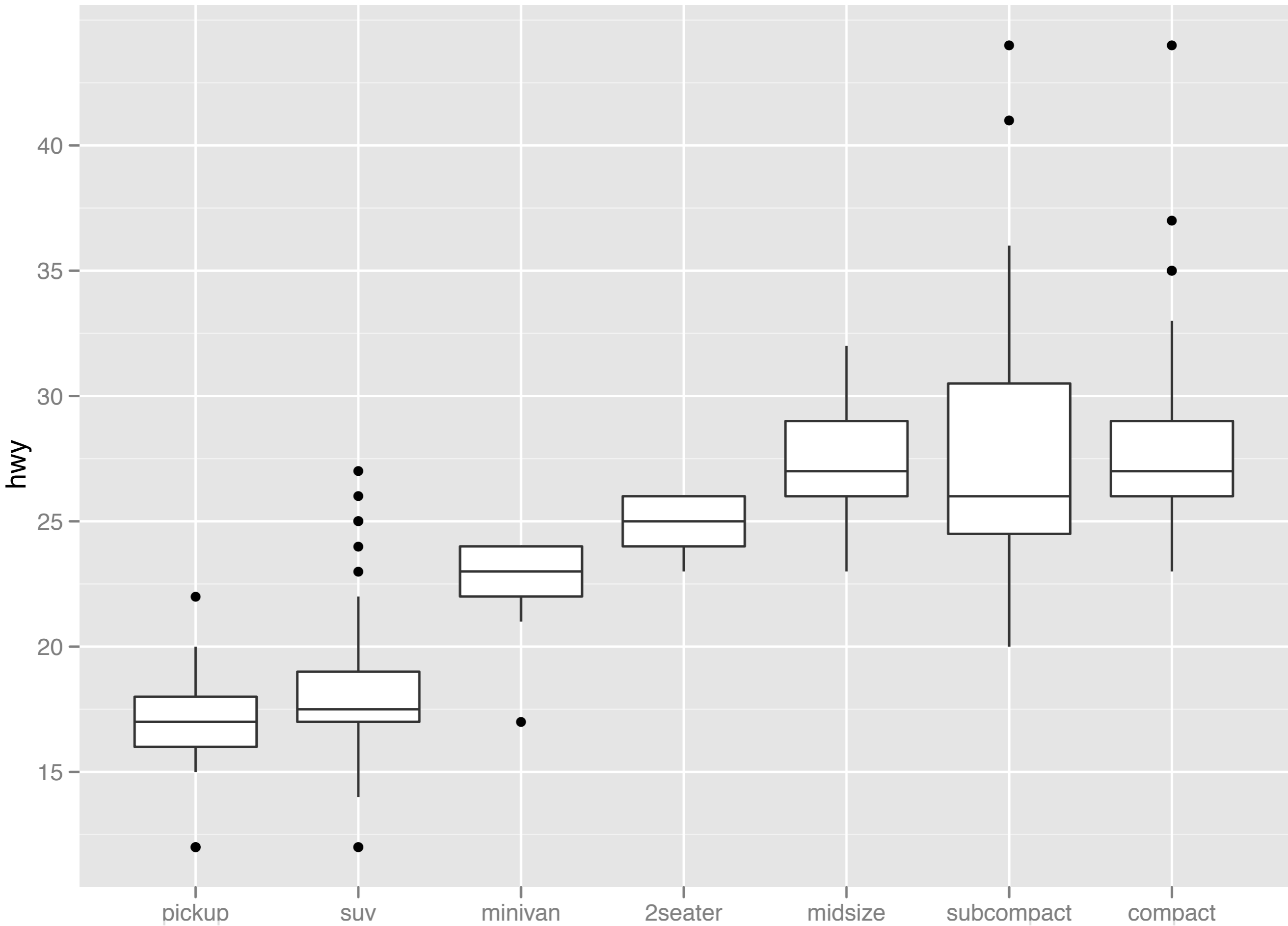


Incredibly useful  
• technique!

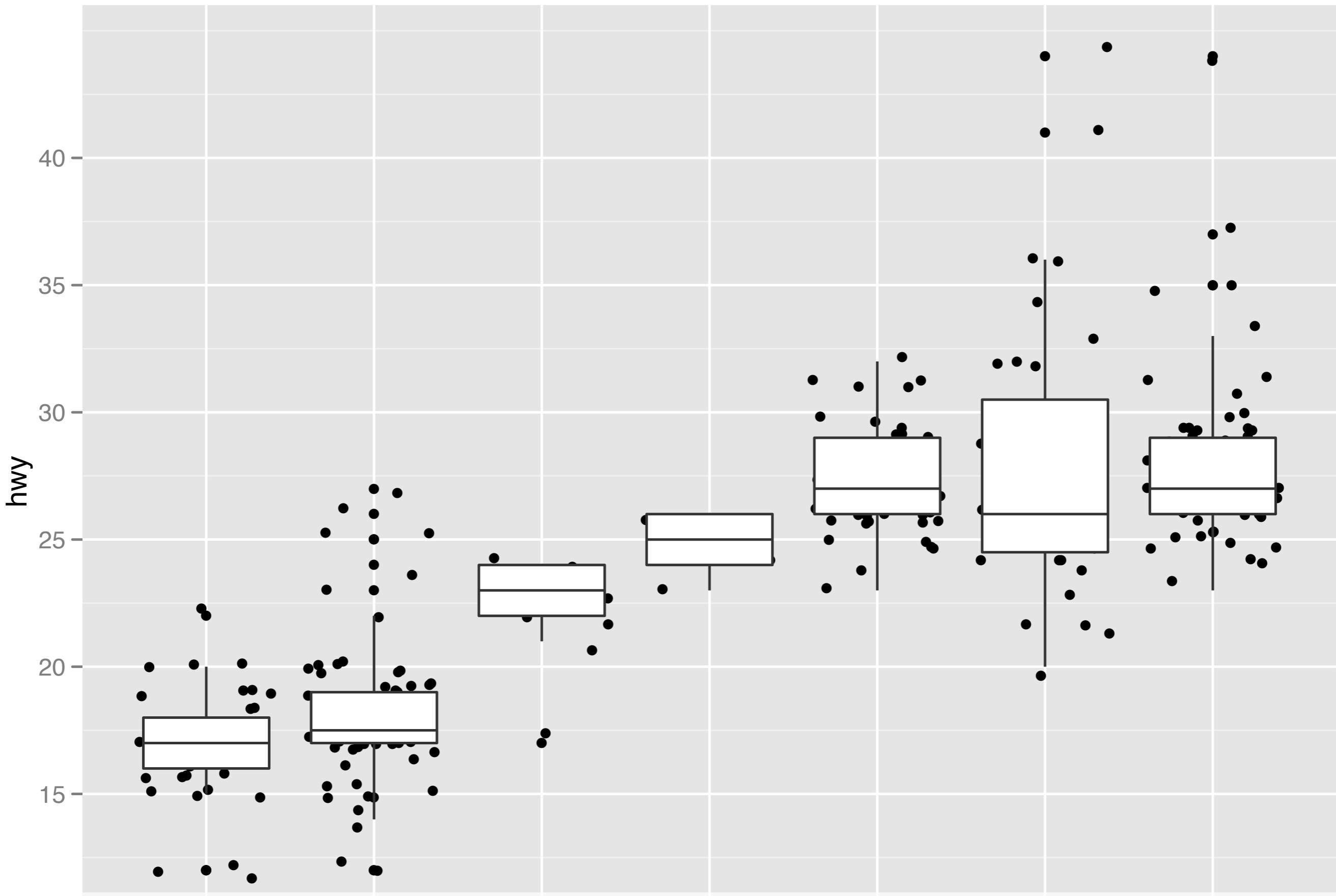
```
qplot(reorder(class, hwy), hwy, data = mpg)
```



```
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(reorder(class, hwy))"))

```



# 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)



This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.