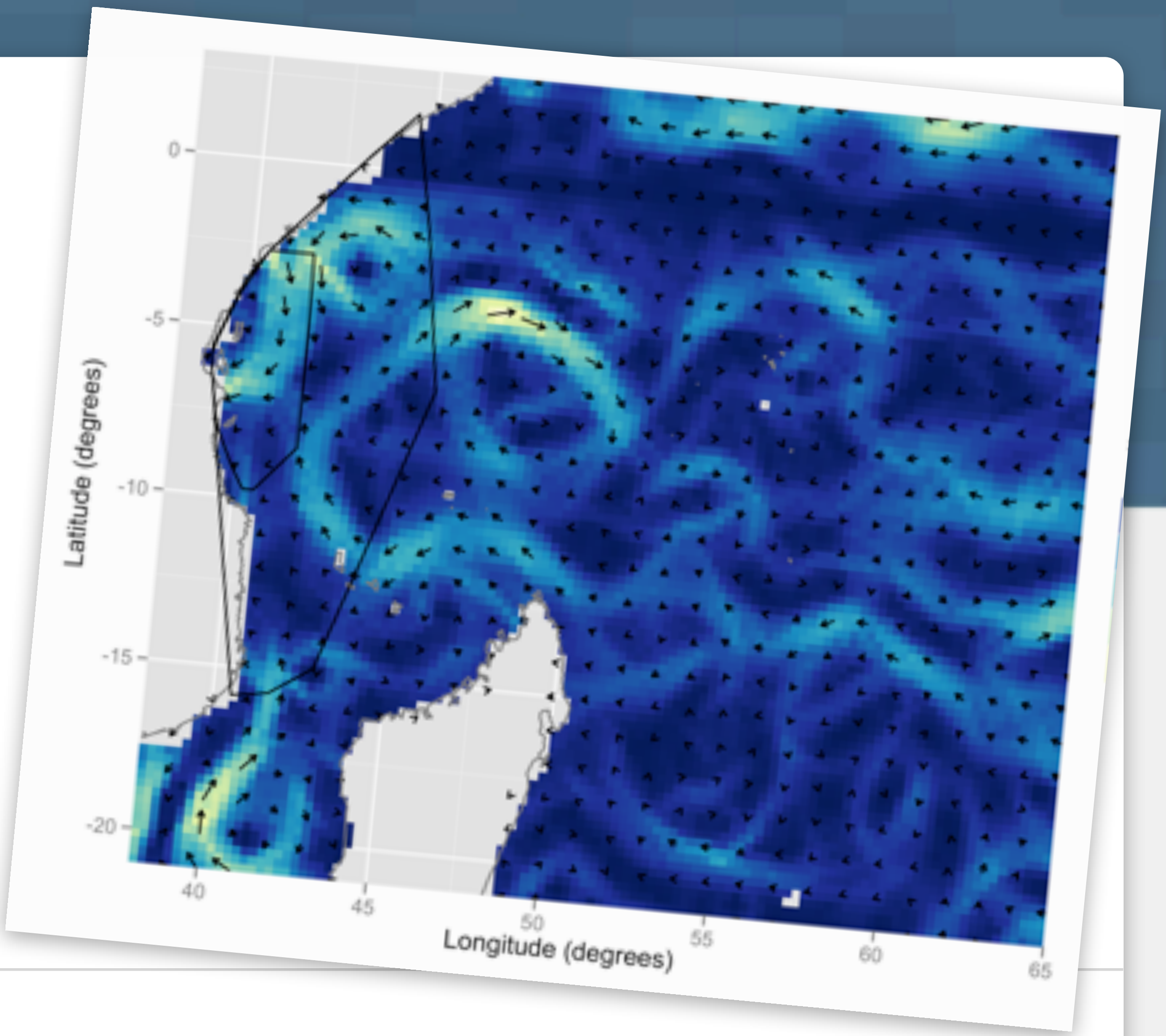
Visualizing Data

Discover the unexpected in your data with ggplot2



Garrett Grolemund

Head Instructor, RStudio

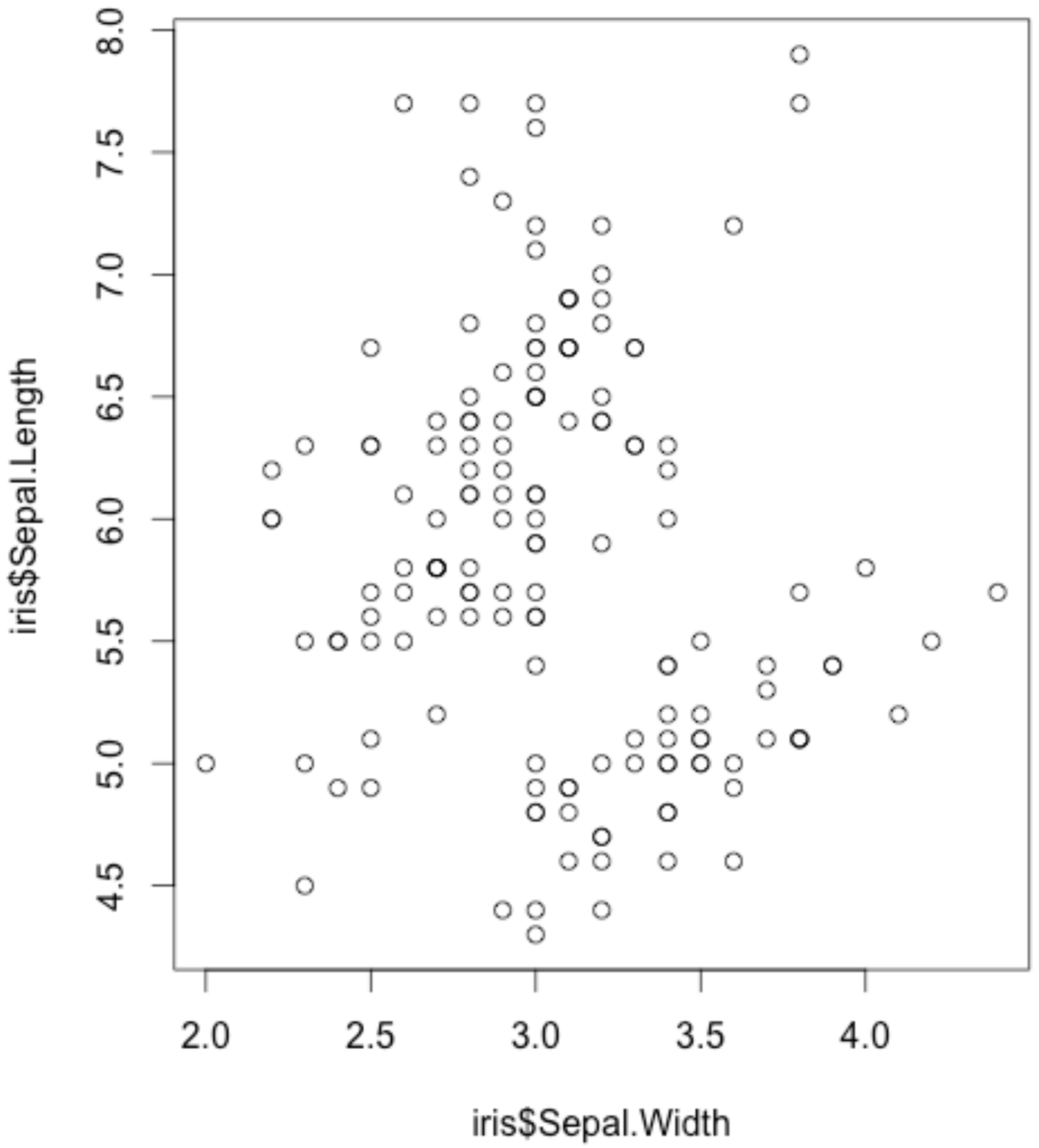
July 2013

1. Scatterplots
    a. Aesthetics, Facets, Geoms
2. Histograms and barcharts
    a. Parameters, Position adjustments
3. Grammar of graphics
4. Layers
5. Customizing graphics

The simple graph has brought more information to the data analyst’s mind than any other device.

– John Tukey

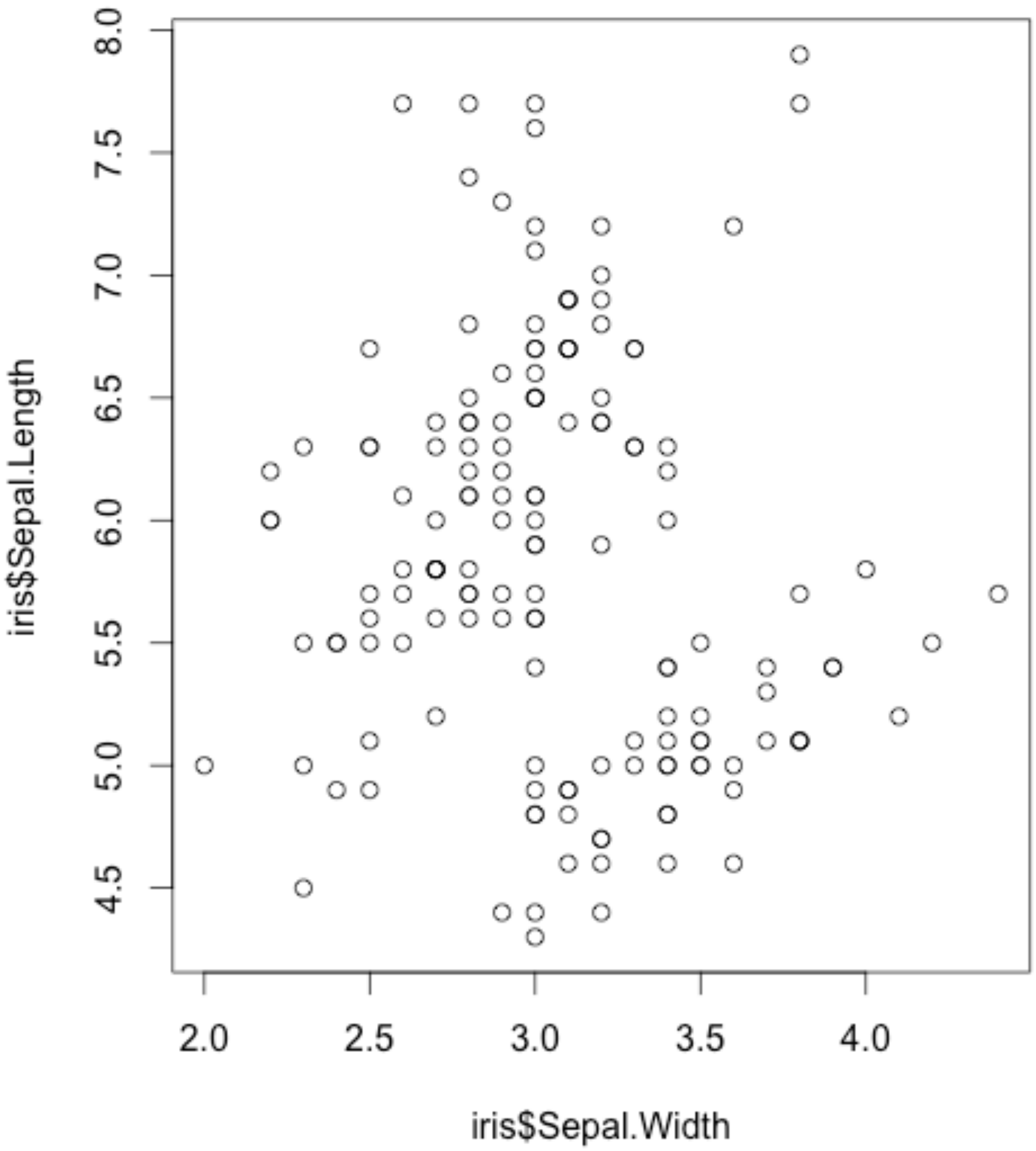
plot

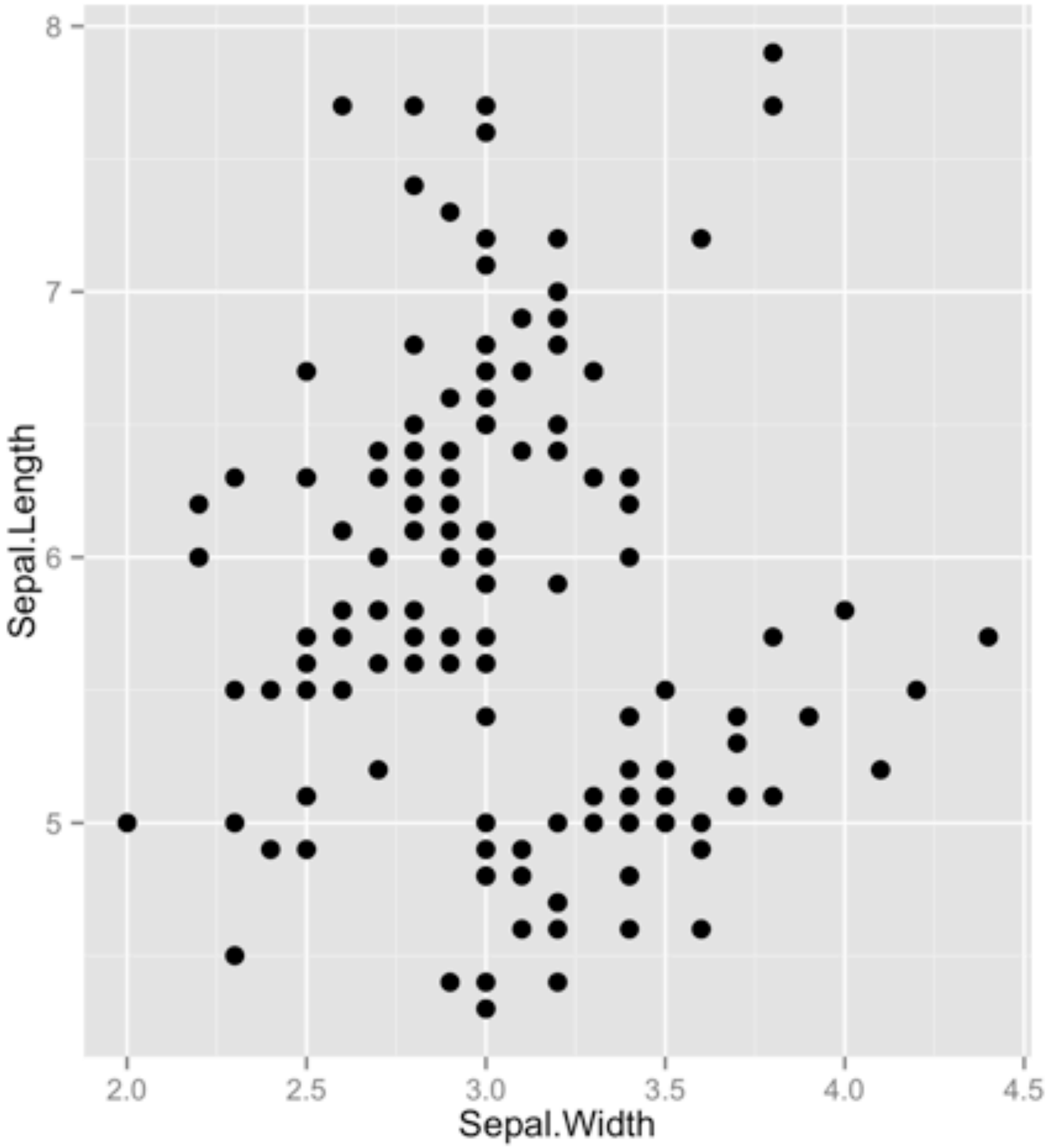


```
plot(iris$Sepal.Width,
     iris$Sepal.Length)
```

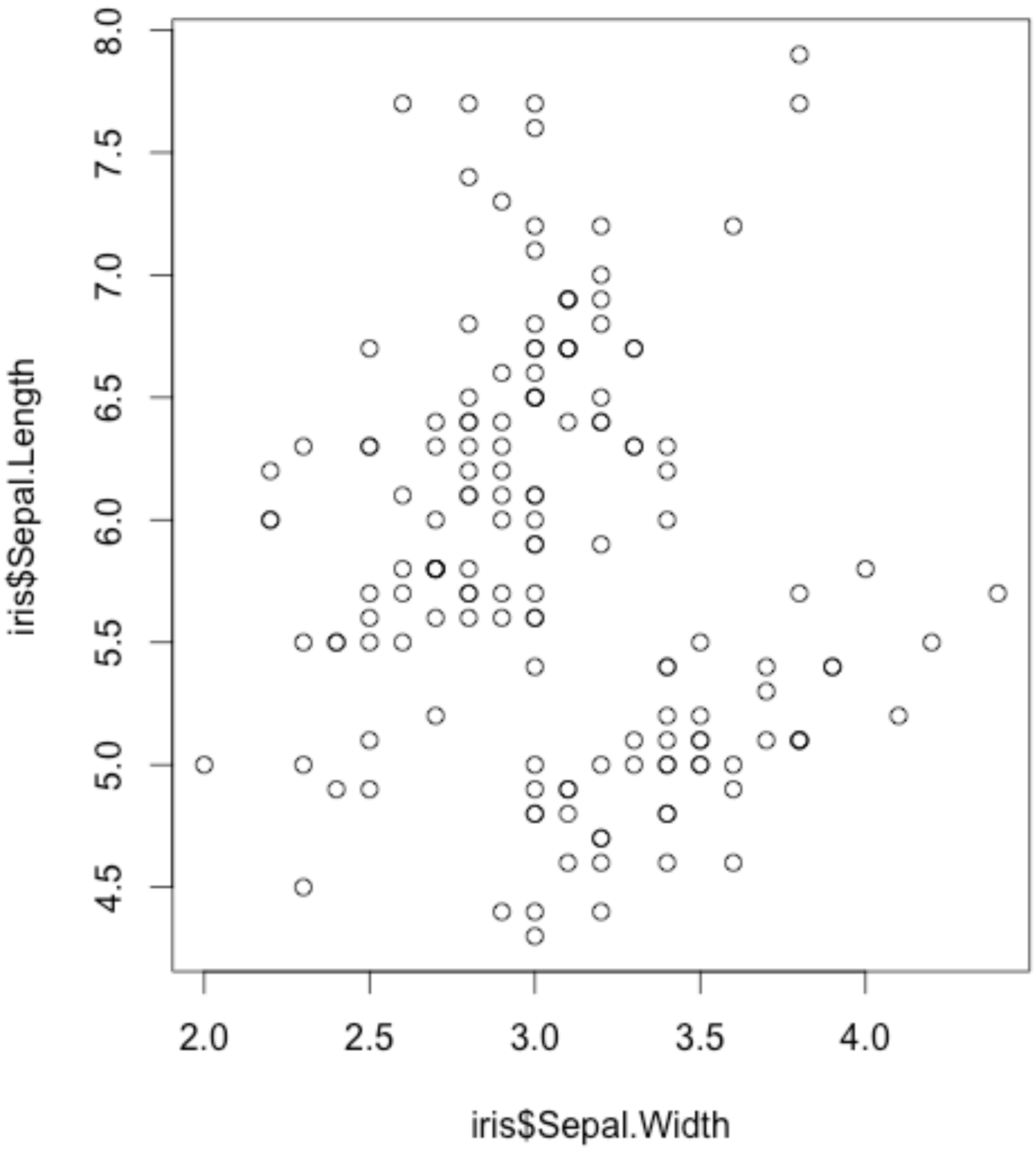
- R's basic plot method
- simple
- does different things in different contexts (usually in a helpful way)
- difficult to customize

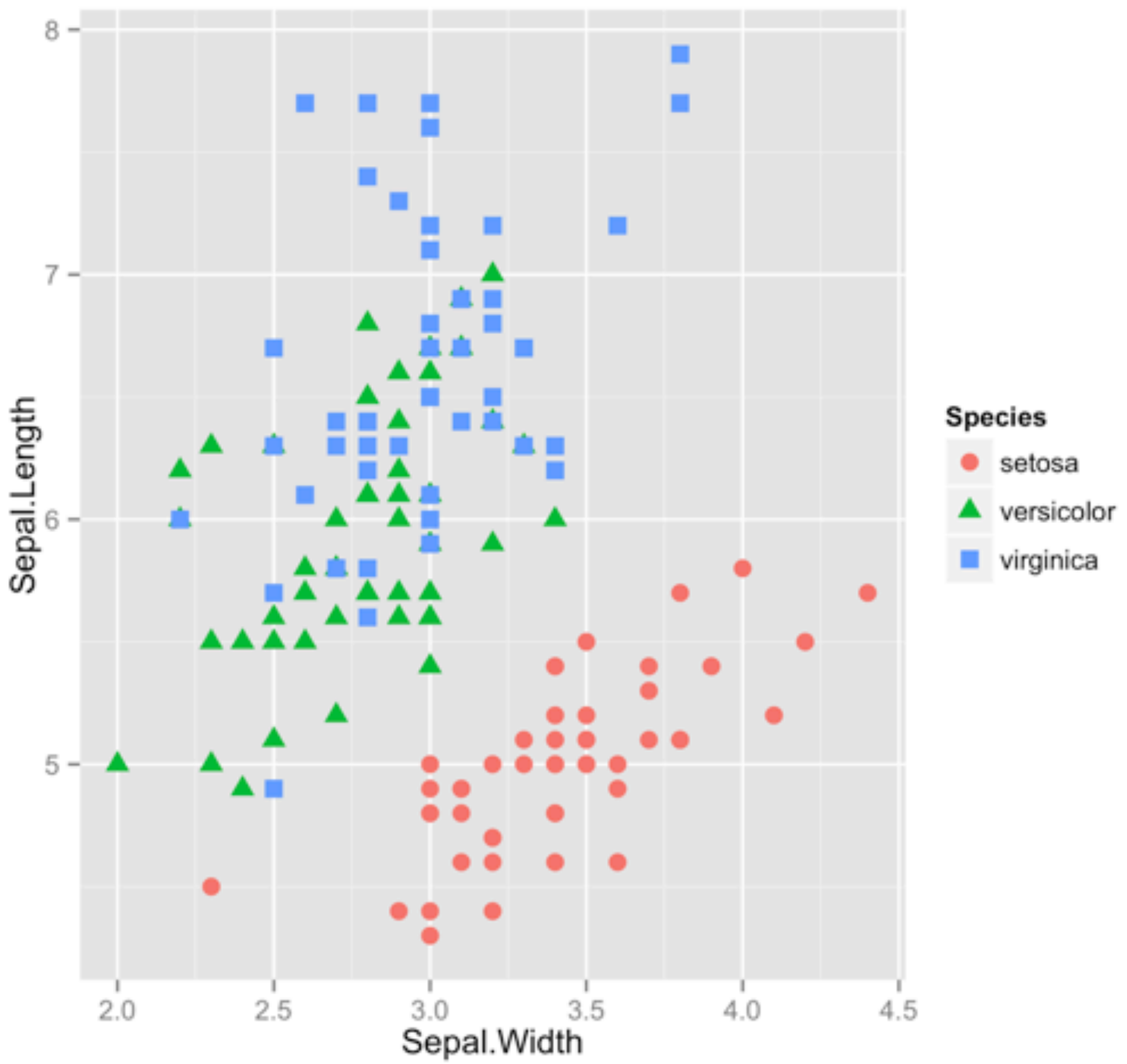
ggplot2


iris$Sepal.Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
2.0
2.5
3.0
3.5
4.0
iris$Sepal.Width

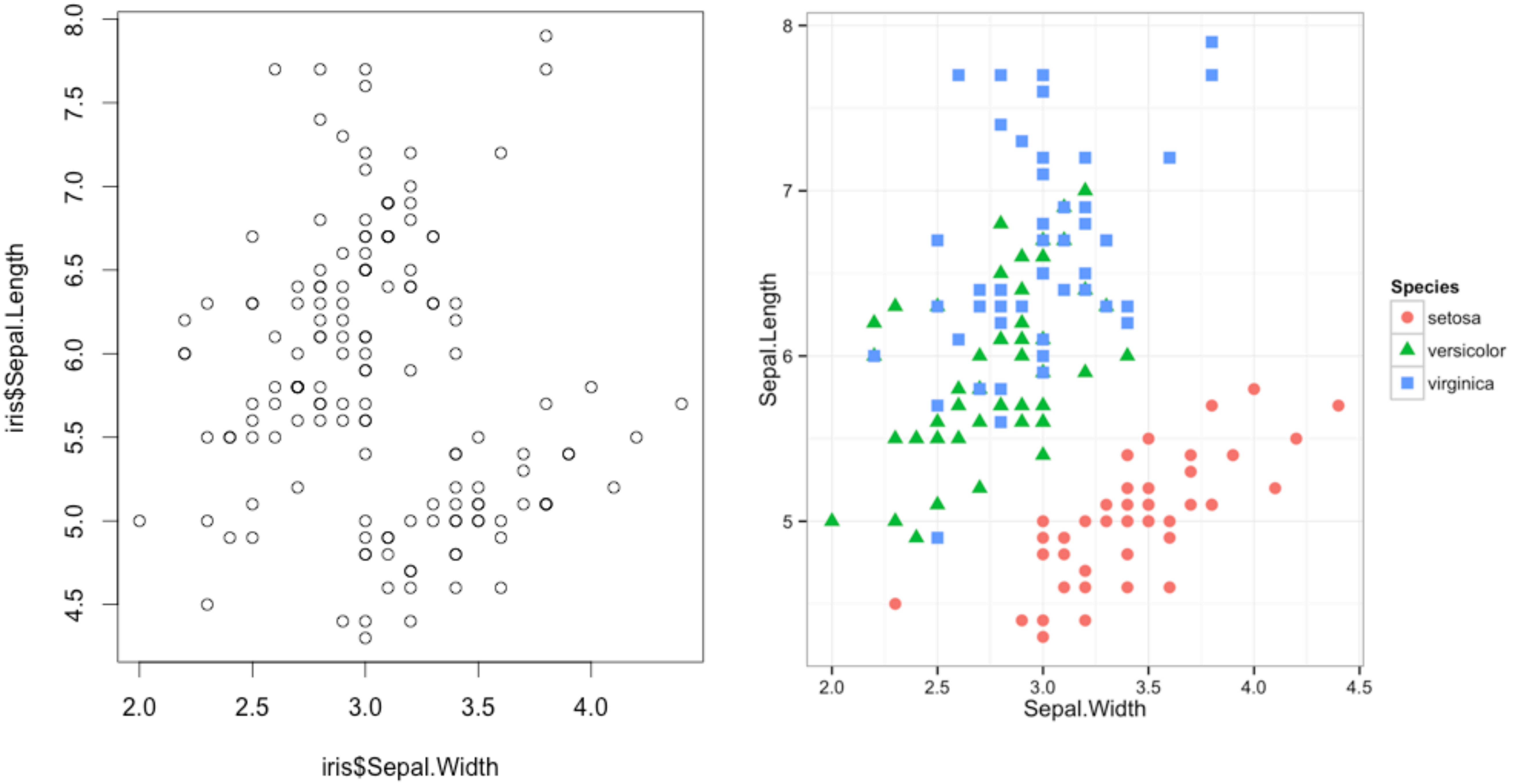

Sepal.Length
5
6
7
8
2.0
2.5
3.0
3.5
4.0
4.5
Sepal.Width

ggplot2

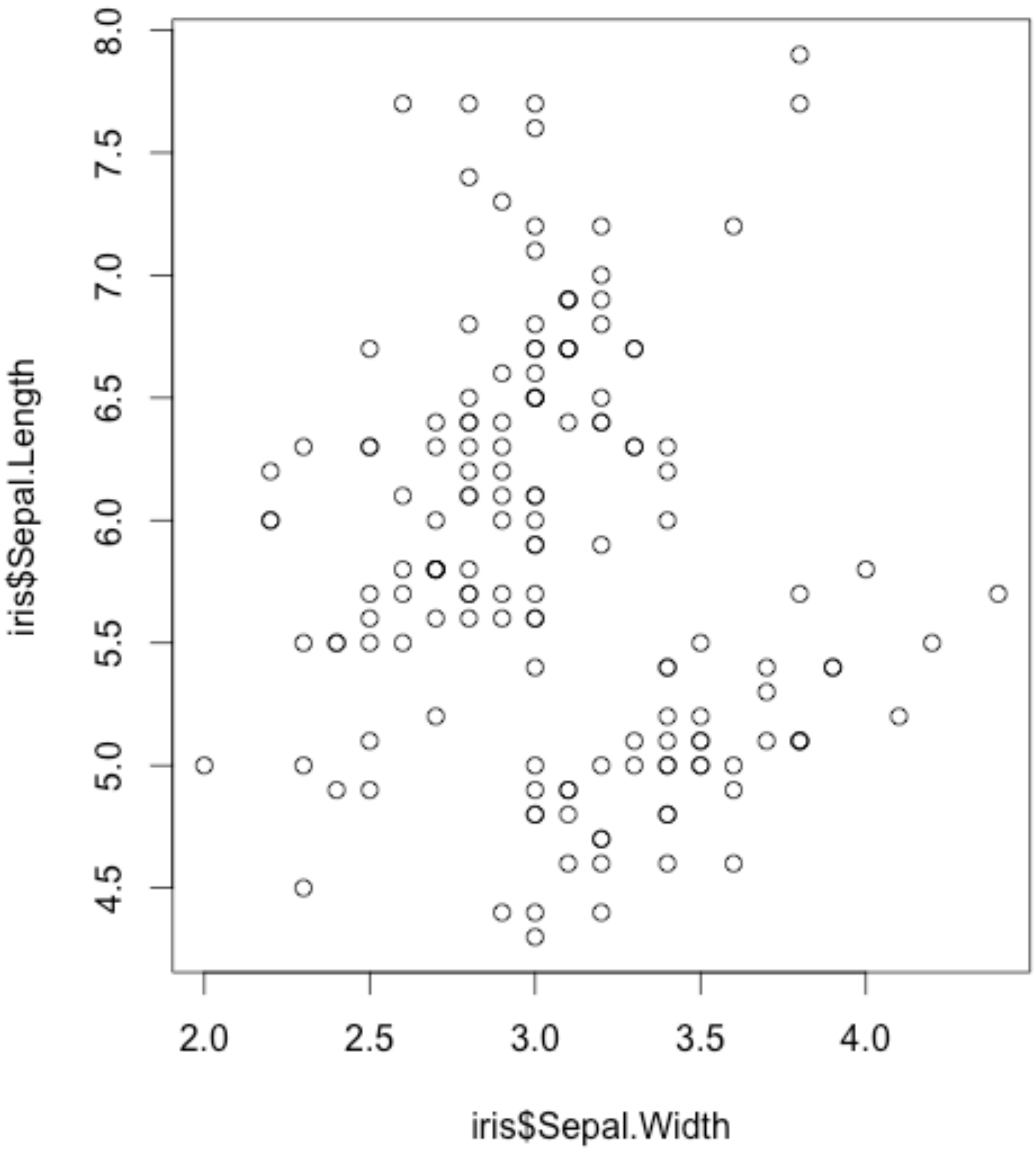

iris$Sepal.Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
2.0
2.5
3.0
3.5
4.0
iris$Sepal.Width

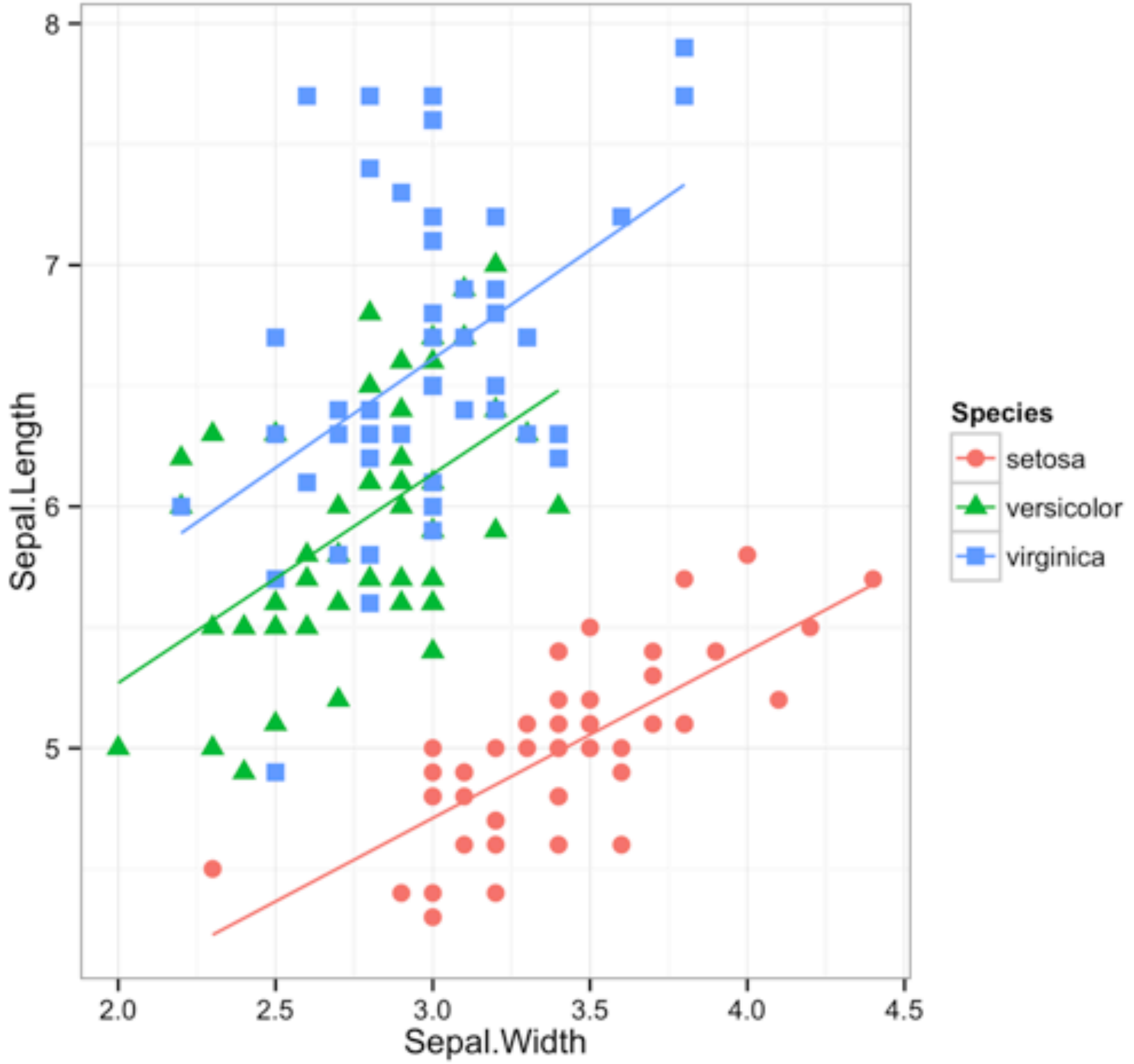

Sepal.Length
5
6
7
8
2.0
2.5
3.0
3.5
4.0
4.5
Sepal.Width
Species
● setosa
▲ versicolor
■ virginica

ggplot2

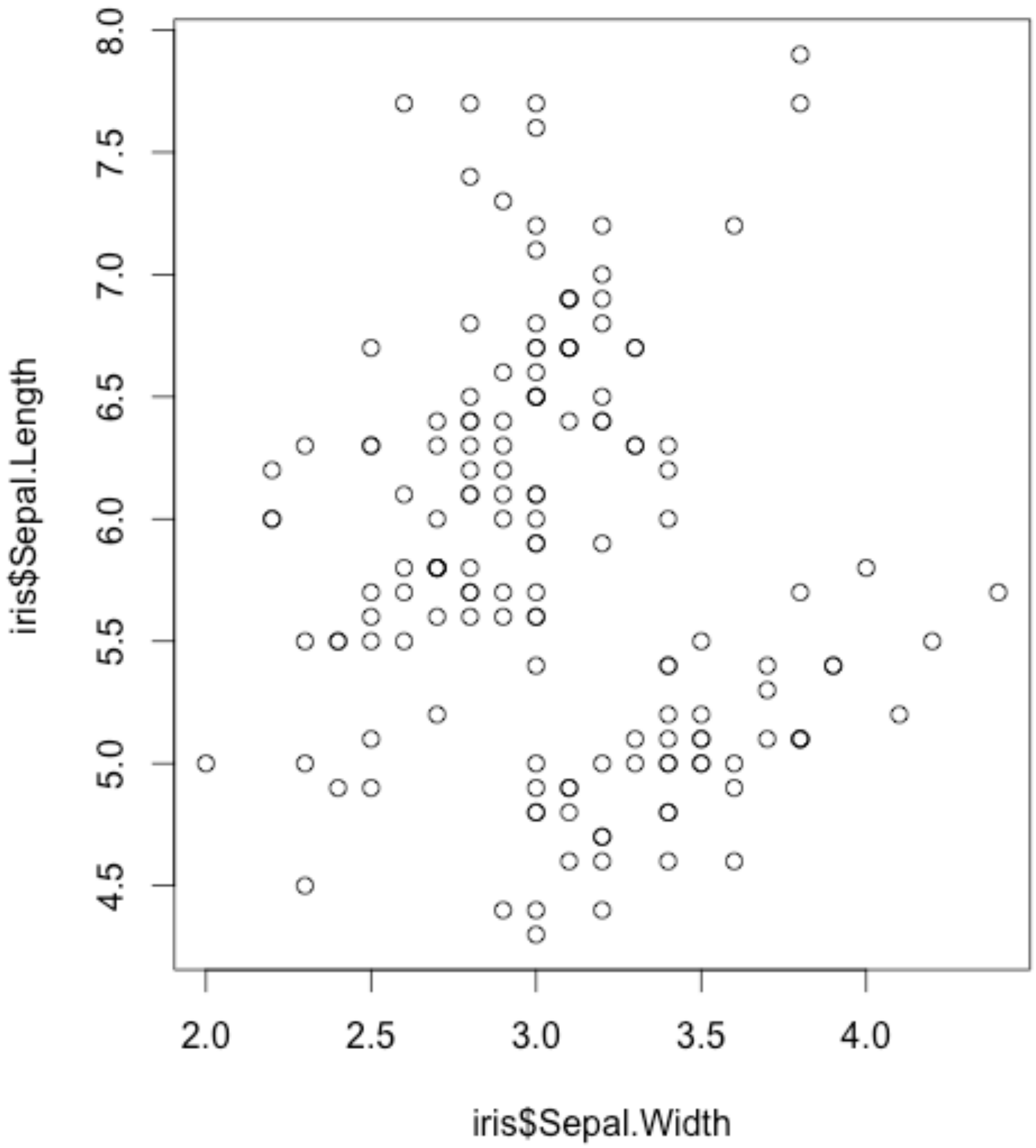

iris$Sepal.Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
2.0
2.5
3.0
3.5
4.0
iris$Sepal.Width
Sepal.Length
5
6
7
8
2.0
2.5
3.0
3.5
4.0
4.5
Sepal.Width
Species
setosa
versicolor
virginica

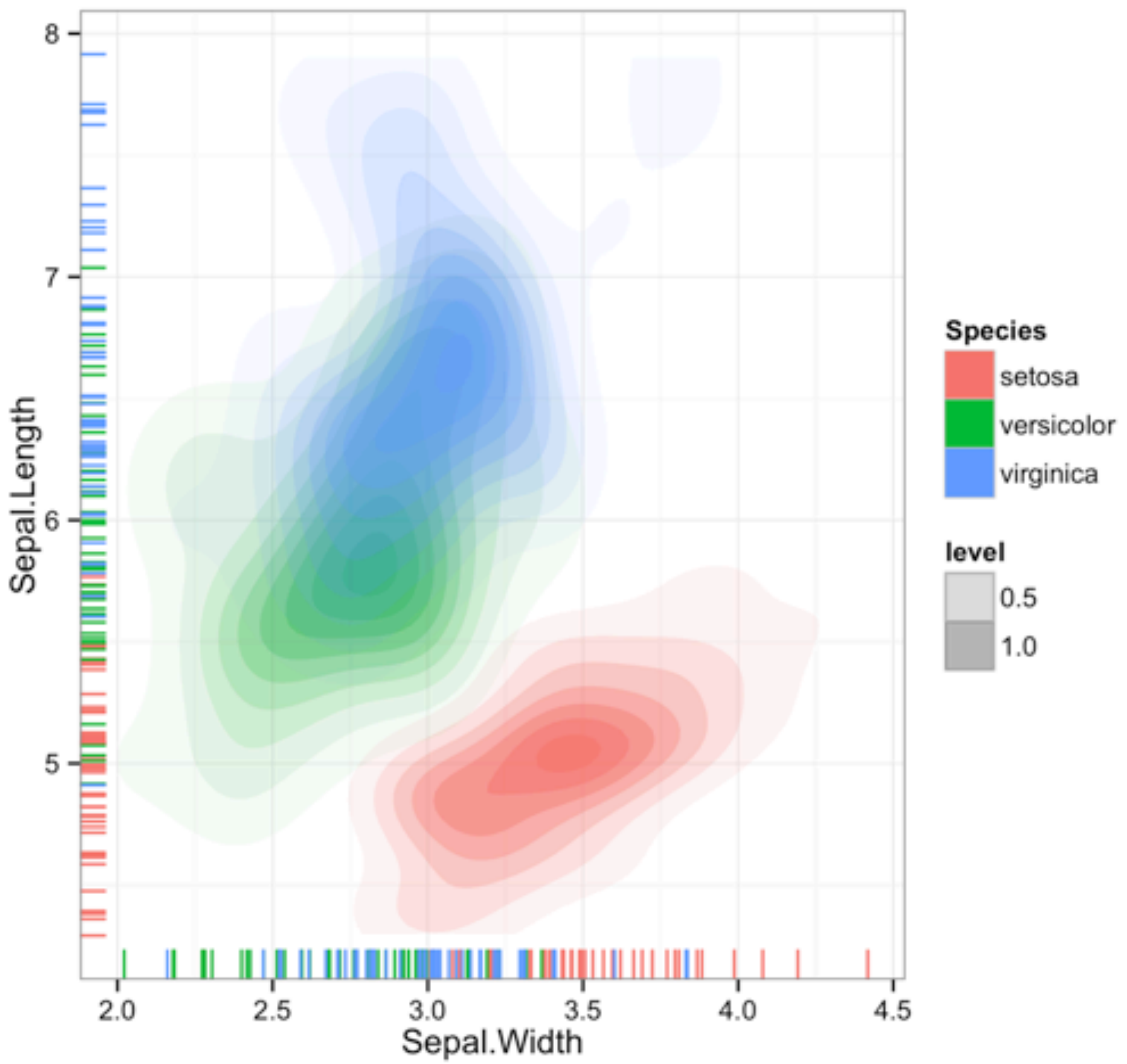
ggplot2


iris$Sepal.Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
2.0
2.5
3.0
3.5
4.0
iris$Sepal.Width

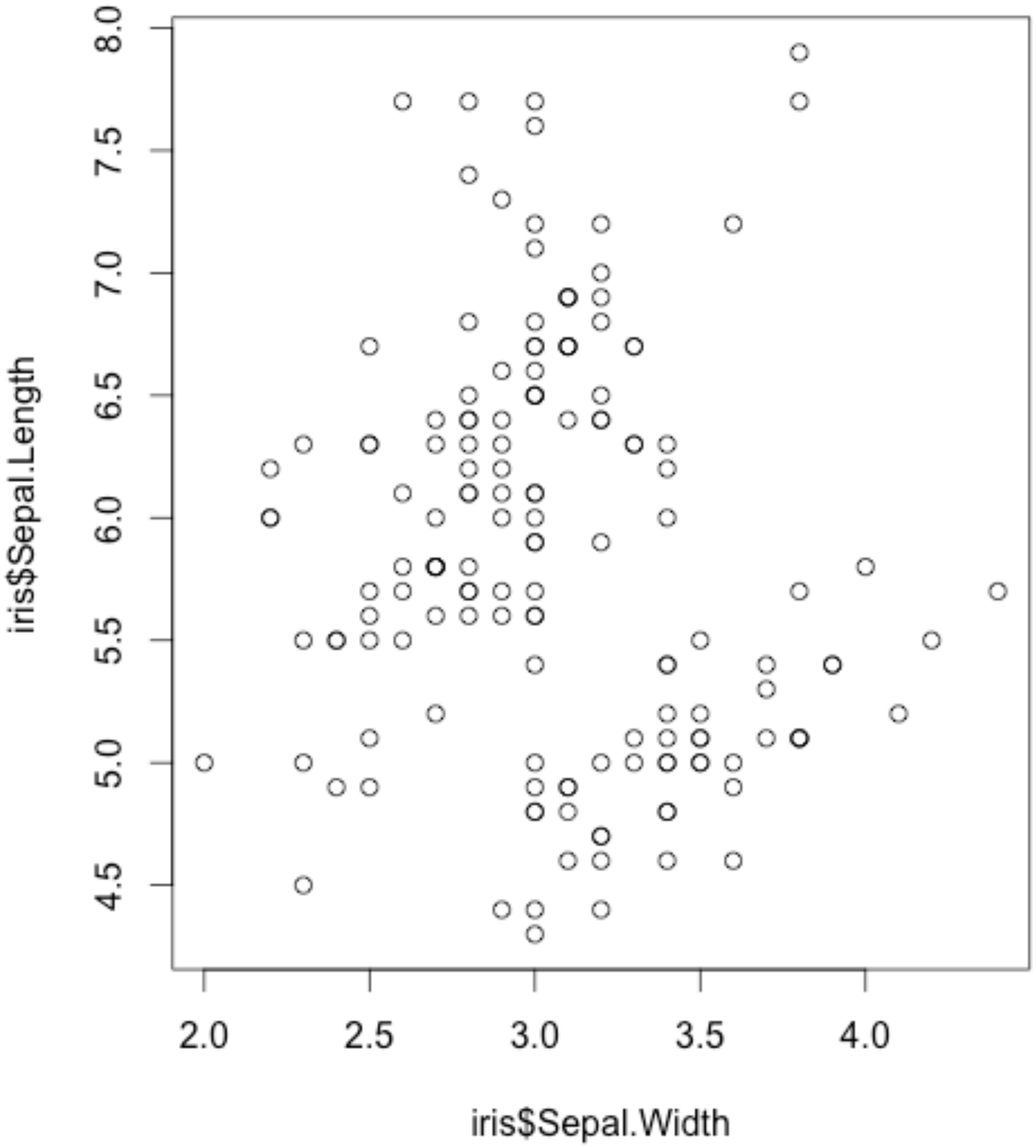

Sepal.Length
5
6
7
8
2.0
2.5
3.0
3.5
4.0
4.5
Sepal.Width
Species
setosa
versicolor
virginica

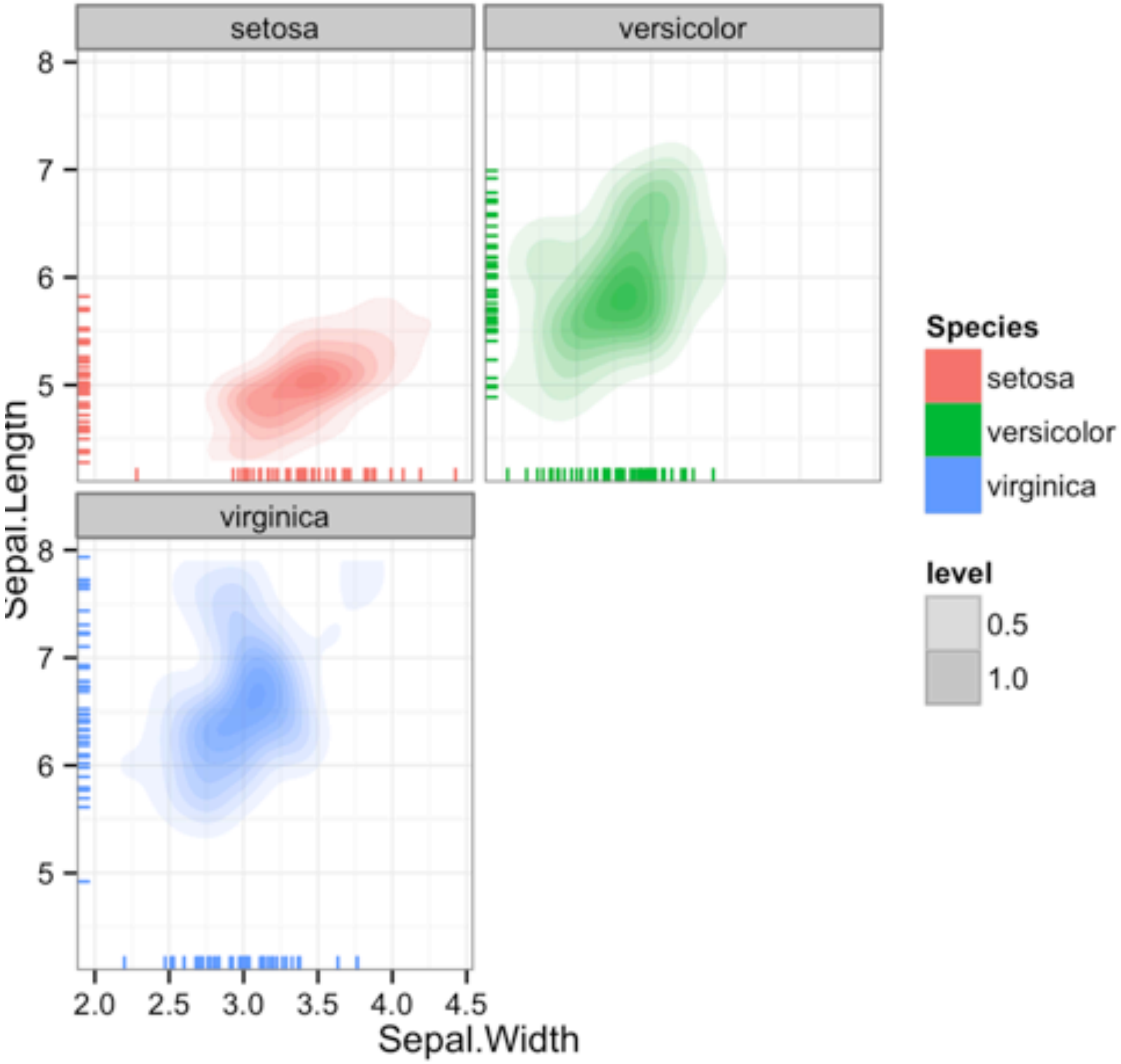
ggplot2

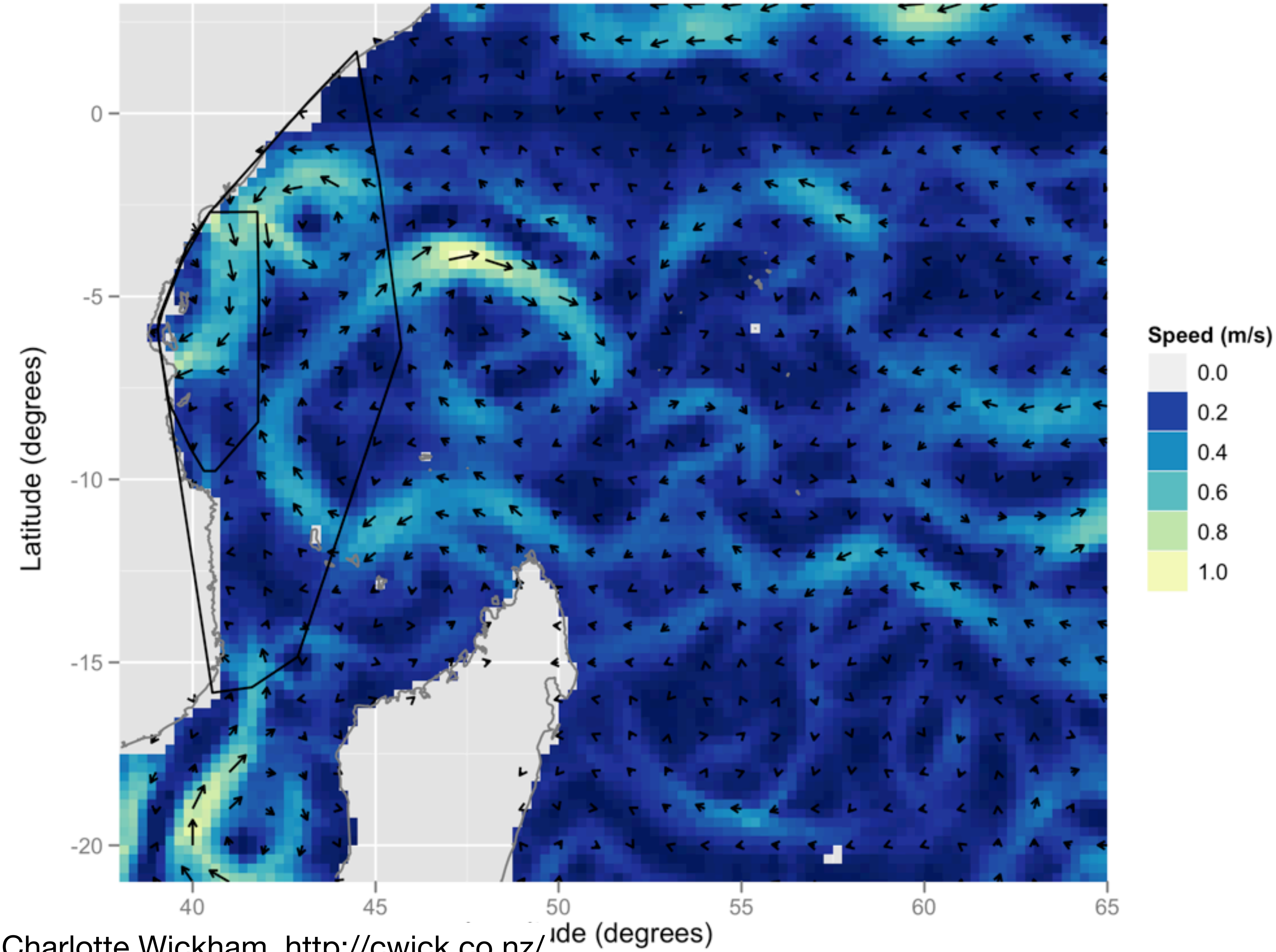

iris$Sepal.Length
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0
iris$Sepal.Width


Sepal.Length
8
7
6
5
2.0
2.5
3.0
3.5
4.0
4.5
Sepal.Width
Species
setosa
versicolor
virginica
level
0.5
1.0

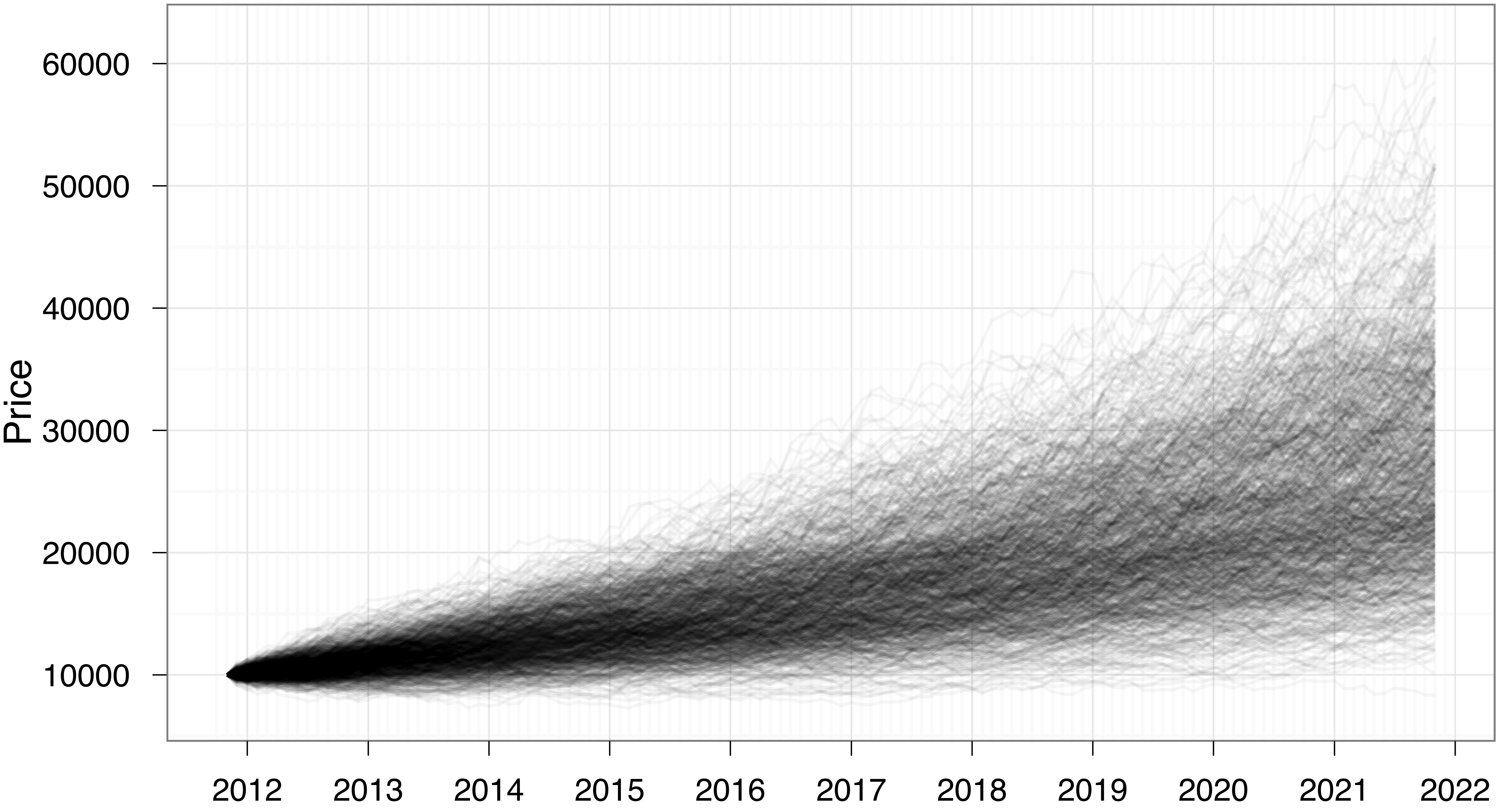
ggplot2

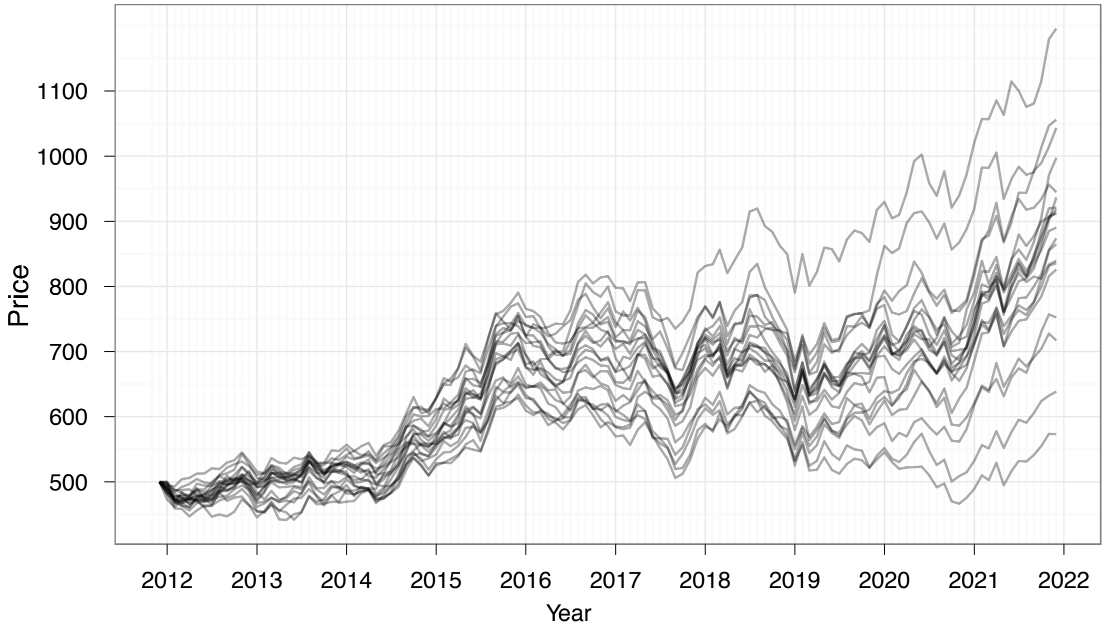

iris$Sepal.Length
iris$Sepal.Width
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0

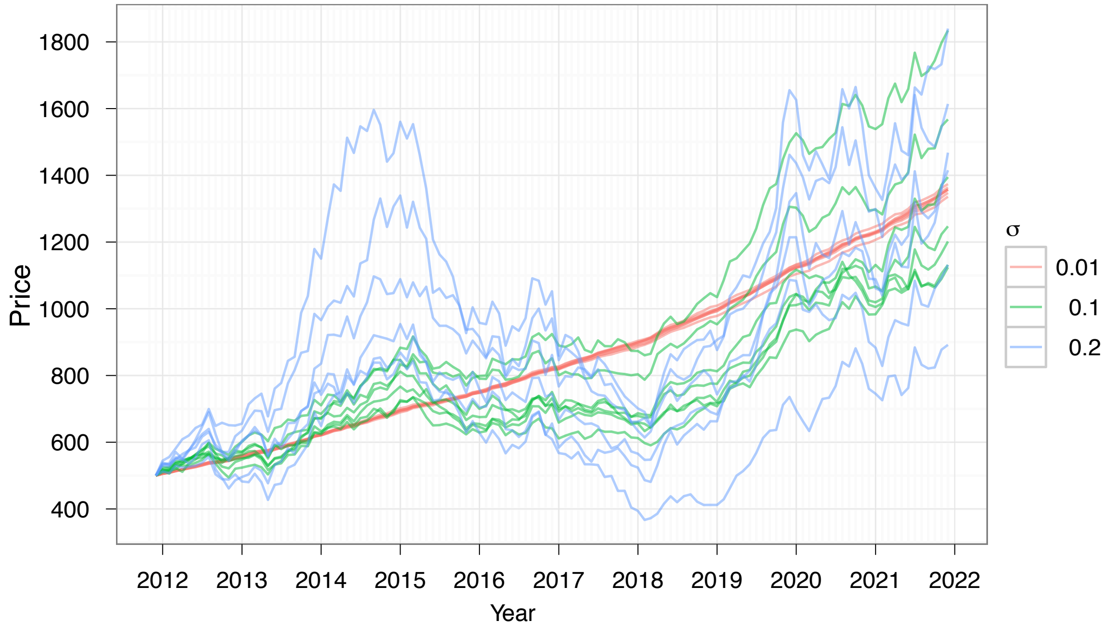

setosa
versicolor
virginica
Sepal.Length
Sepal.Width
8
7
6
5
2.0
2.5
3.0
3.5
4.0
4.5
Species
setosa
versicolor
virginica
level
0.5
1.0

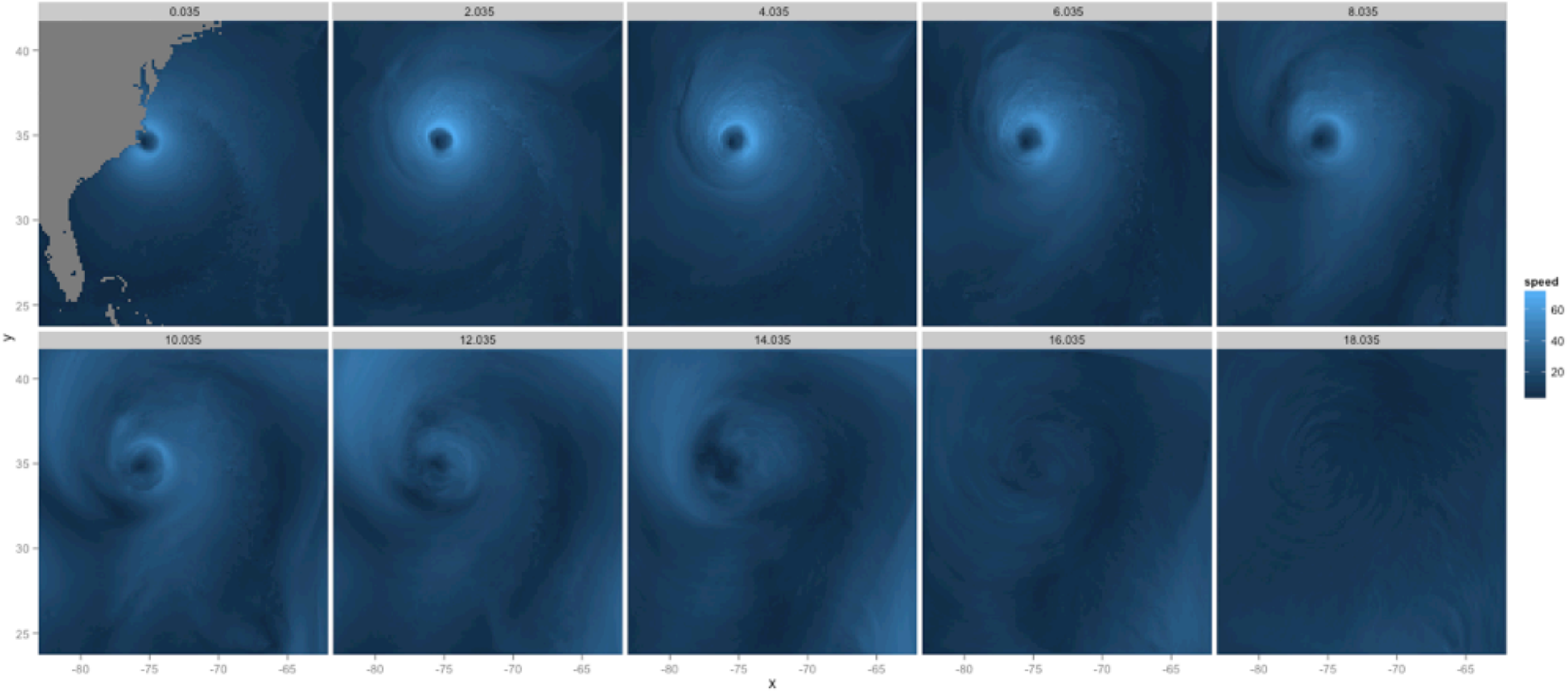


Charlotte Wickham, http://cwick.co.nz/

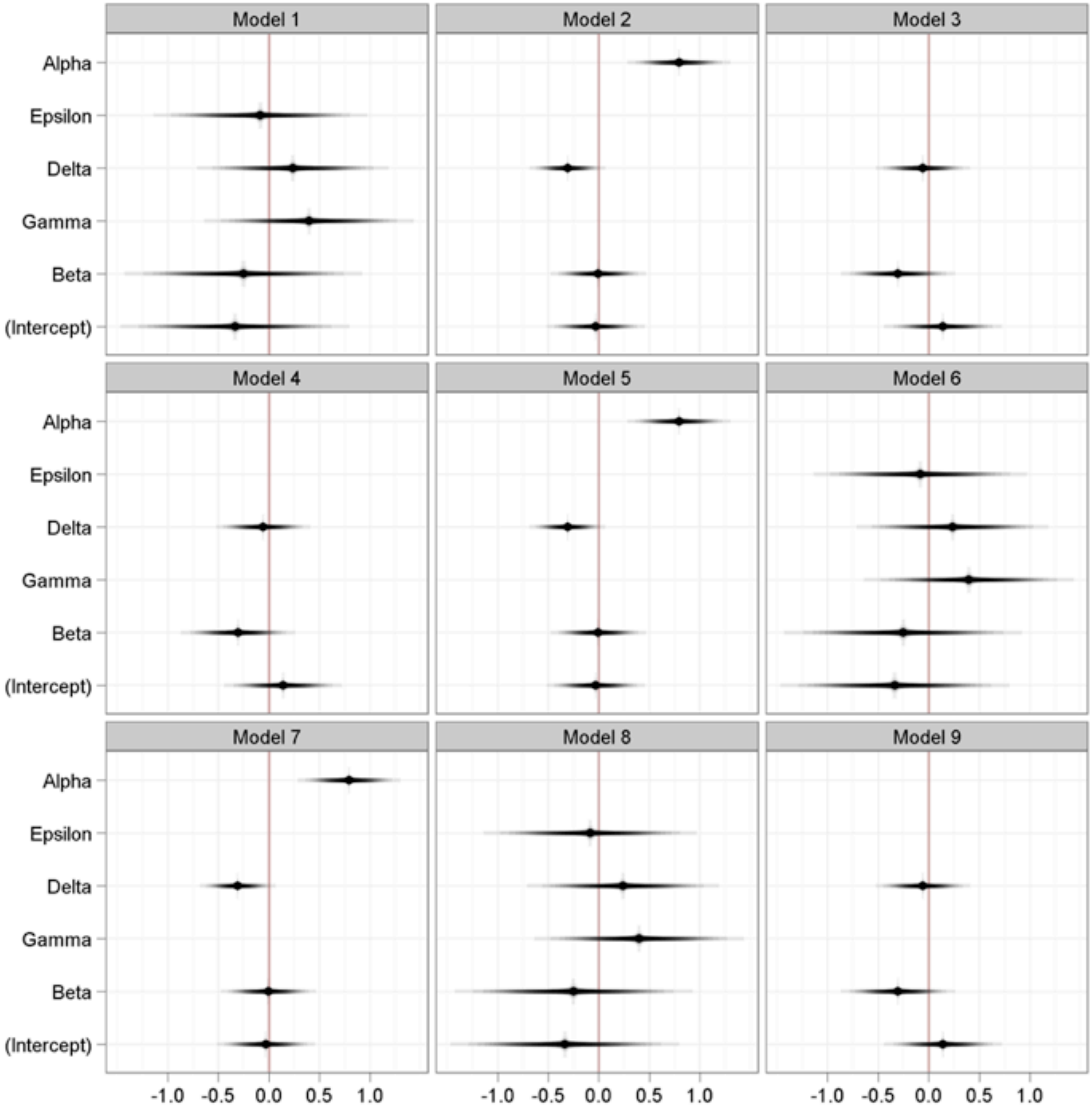

Price
60000
50000
40000
30000
20000
10000
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022


Price
1100
1000
900
800
700
600
500
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Year

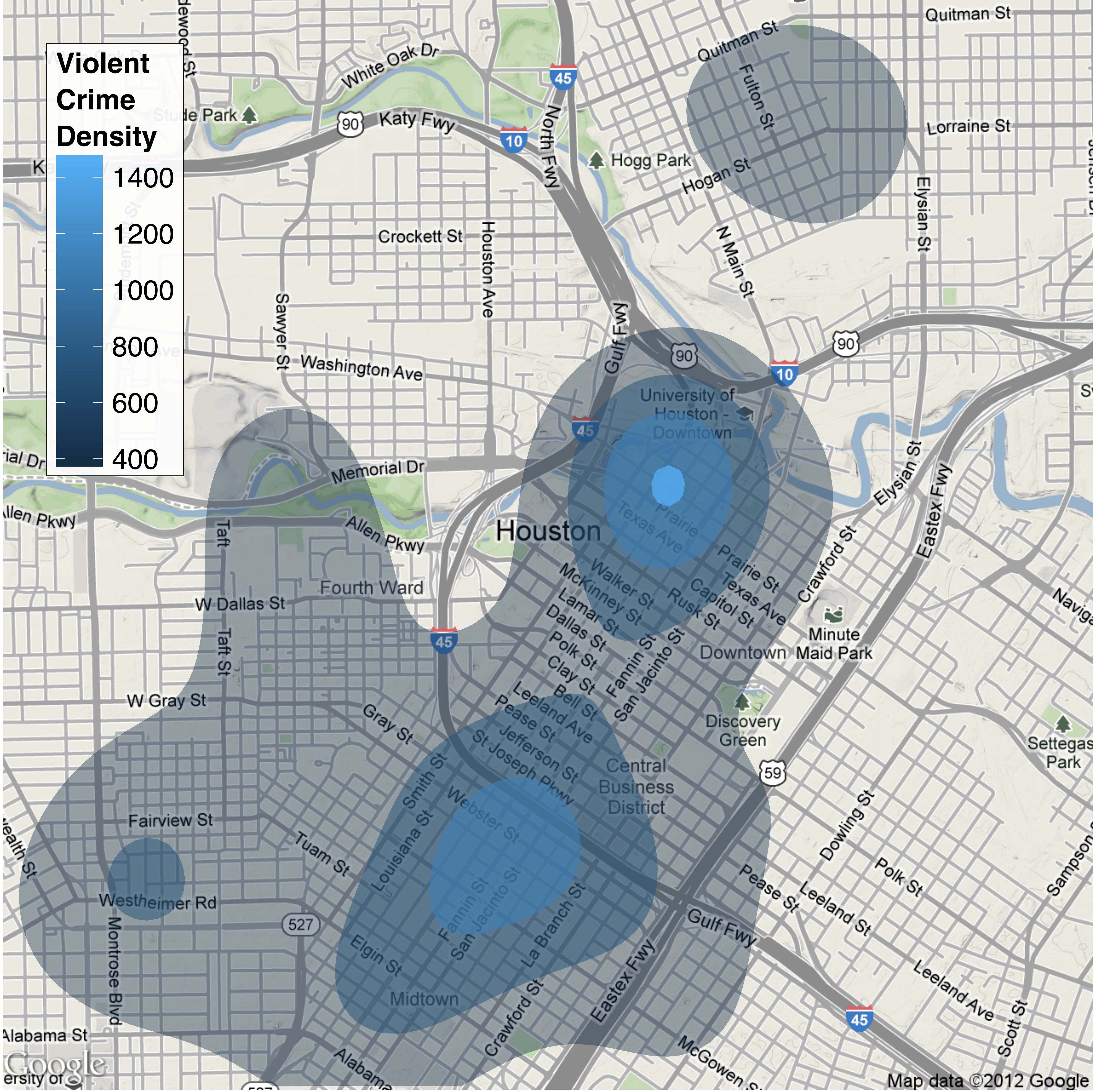

Price
1800
1600
1400
1200
1000
800
600
400
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Year
σ
0.01
0.1
0.2



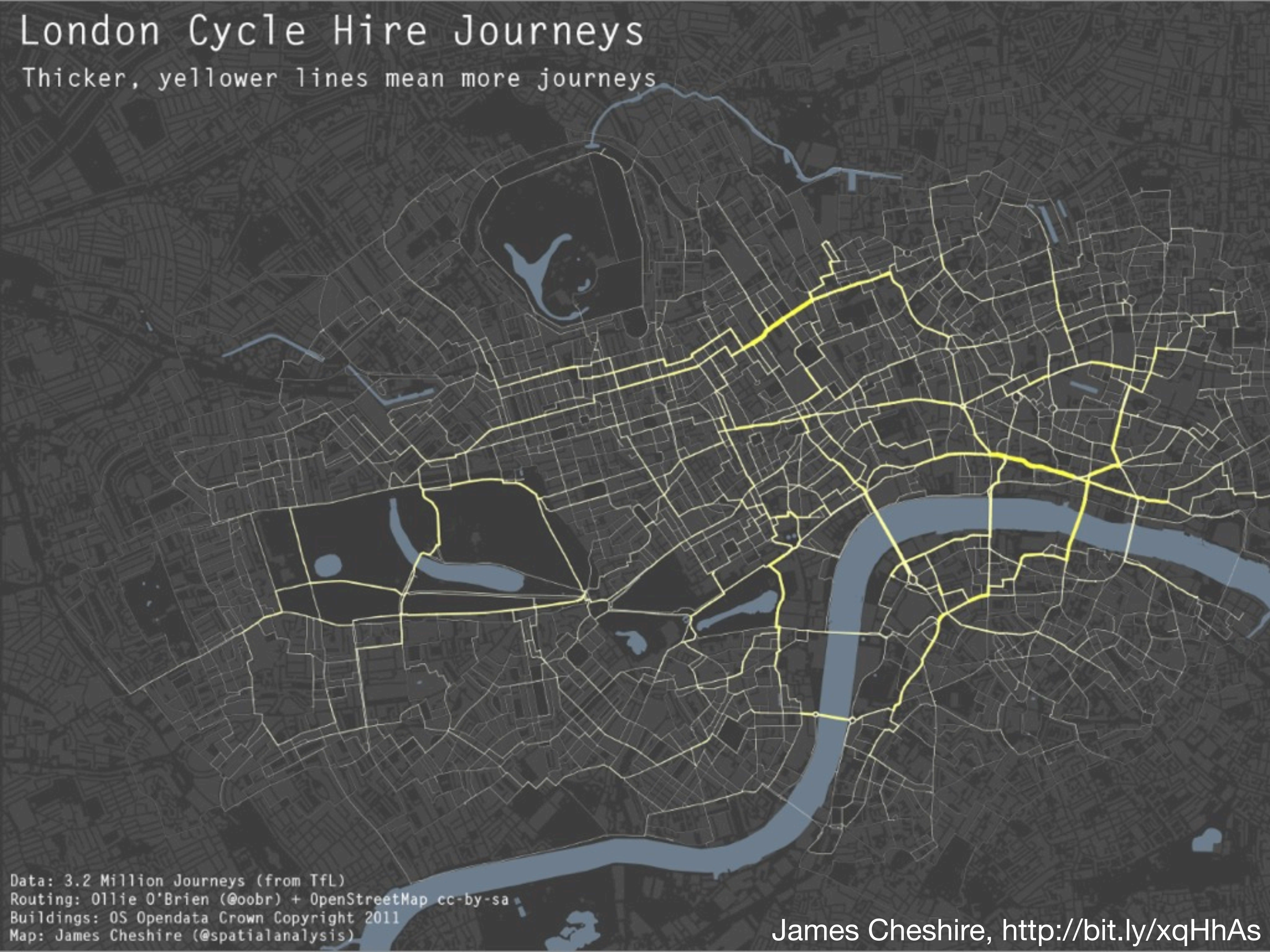
Winston Chang, http://shop.oreilly.com/product/0636920023135.do



David B Sparks, http://bit.ly/hn54NW



David Kahle, https://dl.dropbox.com/u/24648660/ggmap%20useR%202012.pdf



London Cycle Hire Journeys

Thicker, yellower lines mean more journeys

Data: 3.2 Million Journeys (from TfL)
Routing: Ollie O'Brien (@oobr) + OpenStreetMap cc-by-sa
Buildings: OS Opendata Crown Copyright 2011
Map: James Cheshire (@spatialanalysis)

James Cheshire, http://bit.ly/xqHhAs

A picture is not merely worth a thousand words, it is much more likely to be scrutinized than words are to be read.

– John Tukey

Diving in: Scatterplots

Looking at data with R

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

The mpg data set comes in the ggplot2 package

```
?mpg
View(mpg)
```

Always read the help page

Your turn

Make a prediction. What relationship do you expect to see between engine size (displ) and mileage (hwy)?

No peeking ahead!

How can we look at this?

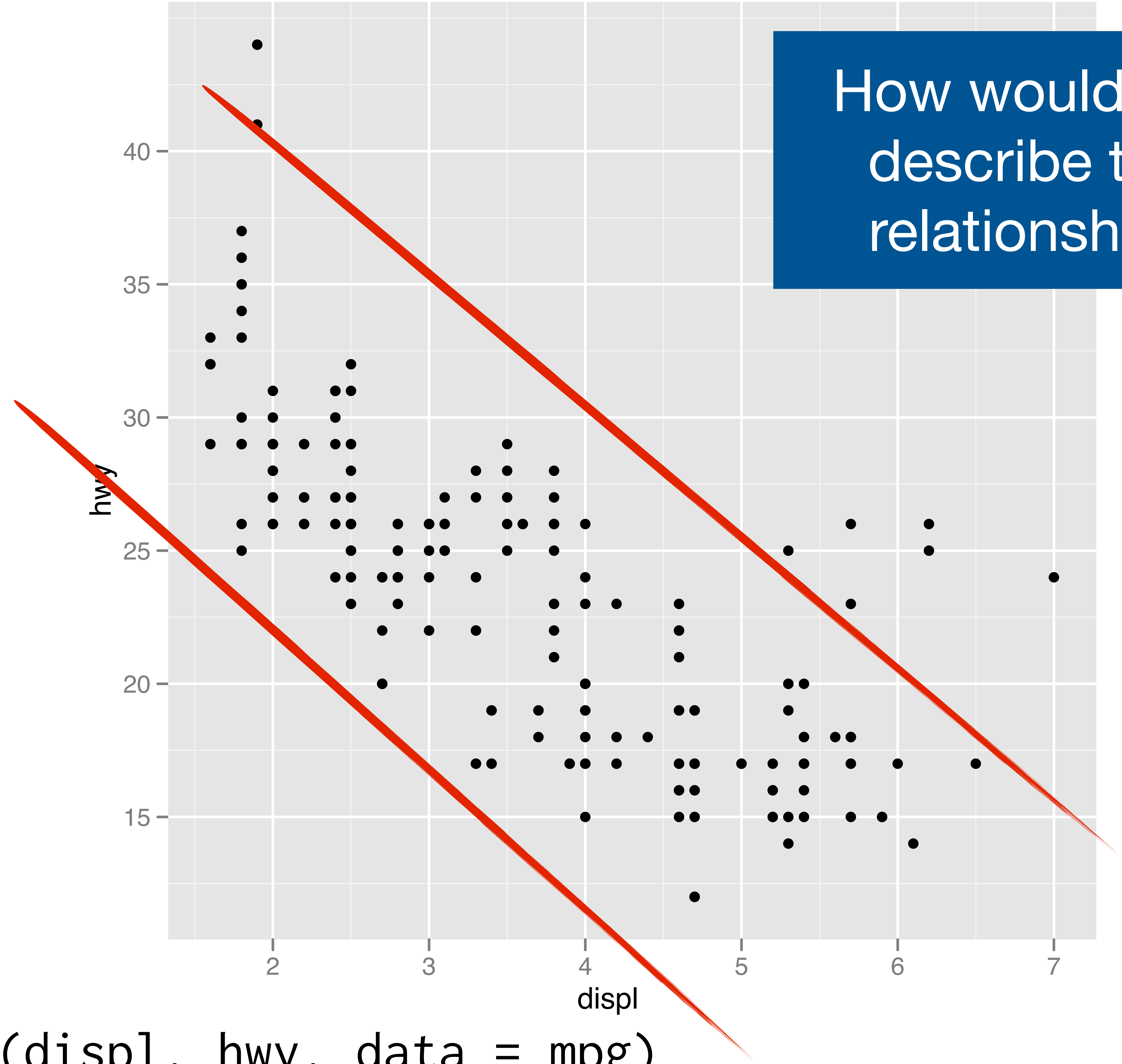
(quick) plots in R

x variable

y variable

data set variables are in

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

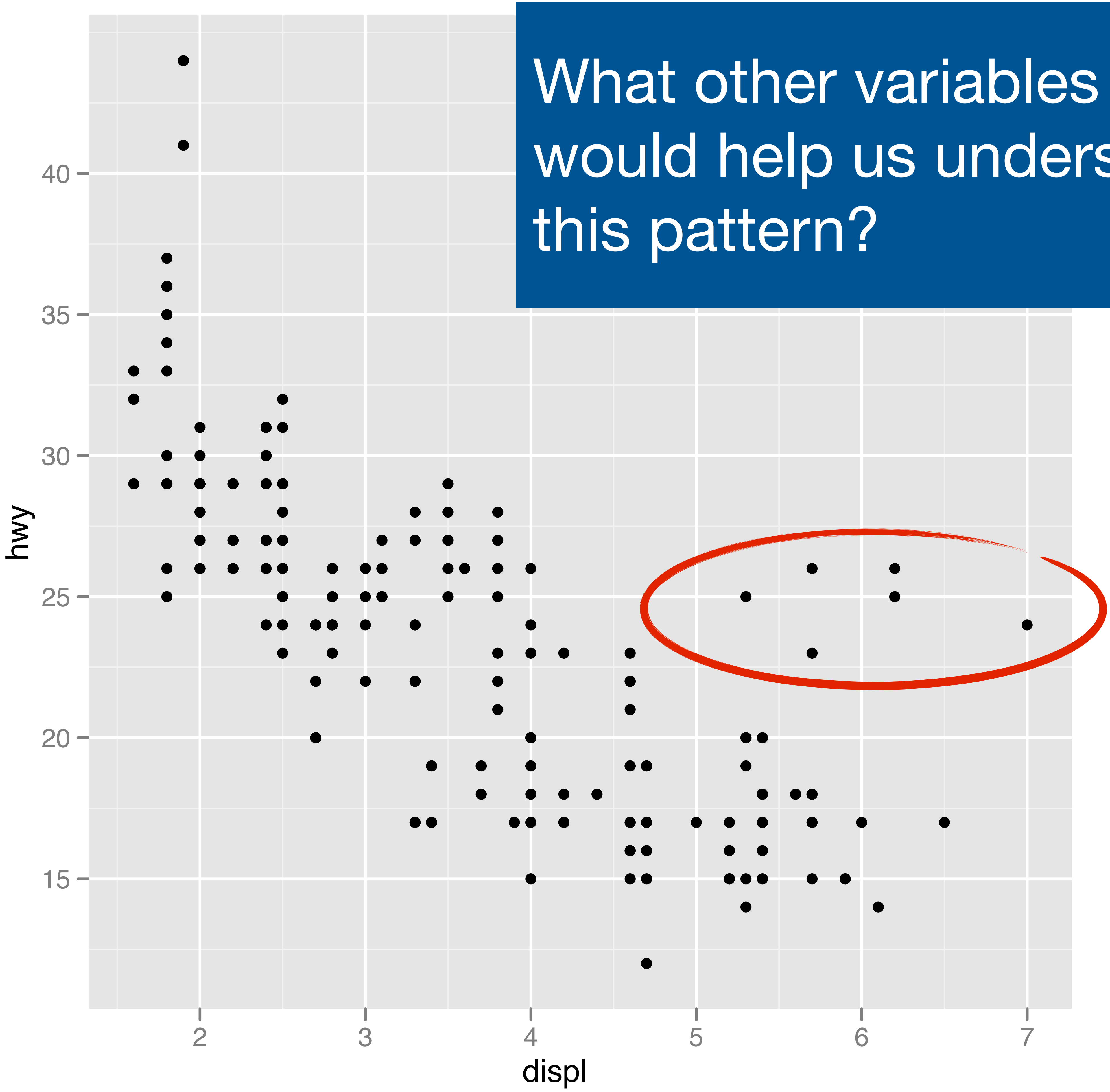


How would you describe this relationship?

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

The greatest value of a picture is when it forces us to notice what we never expected to see.

– John Tukey



What other variables would help us understand this pattern?

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

Additional variables

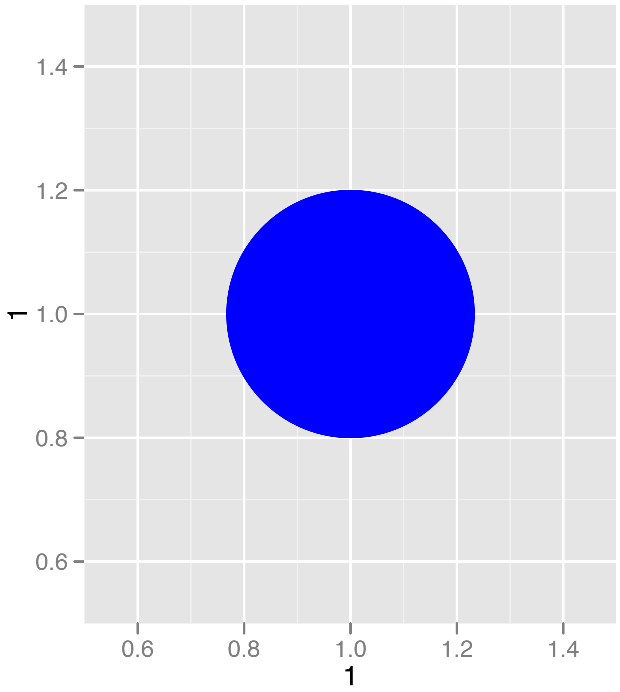
Can display additional variables with

- **aesthetics** (like shape, colour, size), or
- **faceting** (small multiples displaying different subsets)

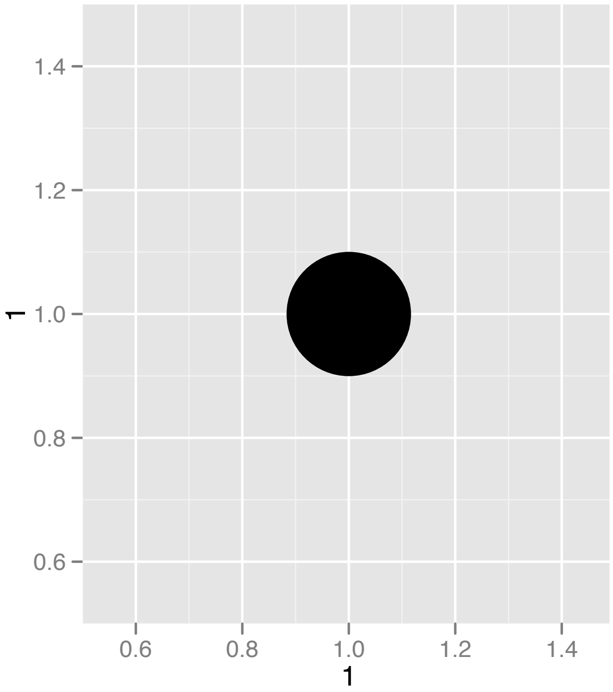
Aesthetics

Aesthetics

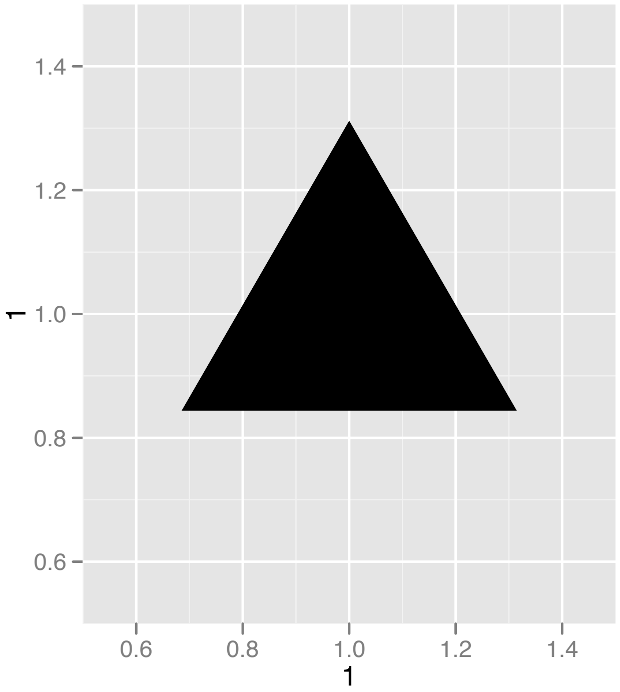
Visual characteristics that can be mapped to data



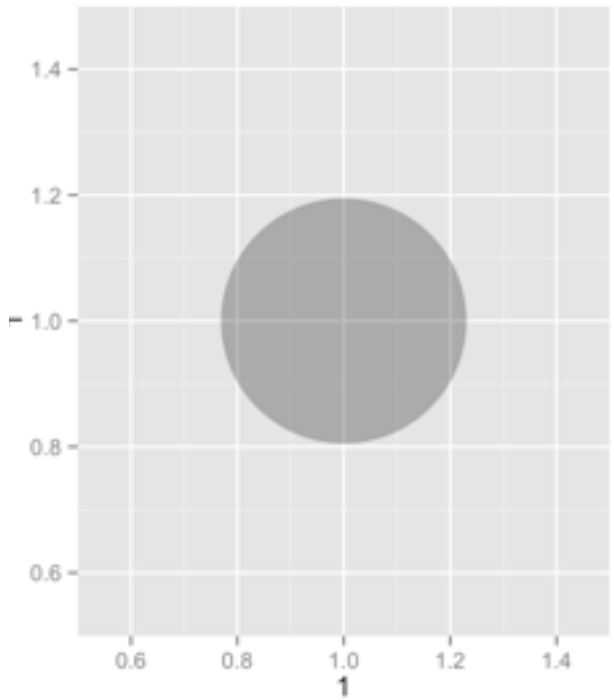
`color`



`size`



`shape`



`alpha`

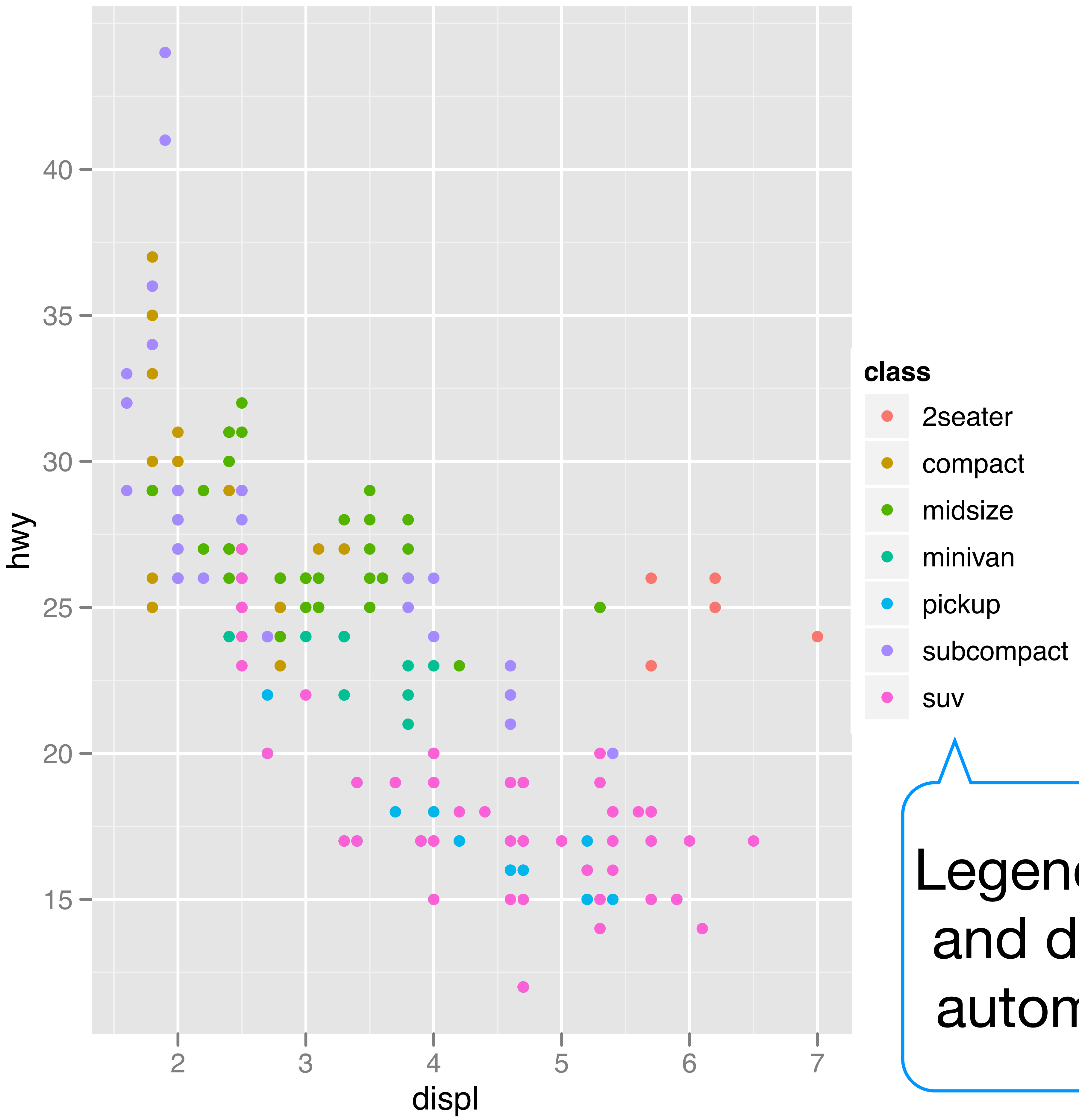
(transparency)

Aesthetics

aesthetic feature

variable to map it to

```
qplot(displ, hwy, data = mpg, color = class)
qplot(displ, hwy, data = mpg, size = class)
qplot(displ, hwy, data = mpg, shape = class)
qplot(displ, hwy, data = mpg, alpha = class)
```



Legend chosen and displayed automatically.

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

Your turn

Add color, size, and shape aesthetics to your graph. Experiment.

Do different things happen for discrete and continuous variables?

What happens when you use more than one aesthetic?

	Discrete	Continuous
Color	Rainbow of colors	Gradient from light blue to dark blue
Size	Discrete size steps	Linear mapping between radius and value
Shape	Different shape for each	Shouldn’t (and doesn’t) work

Faceting

Faceting

Smaller plots that display different subsets of the data.

Also 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

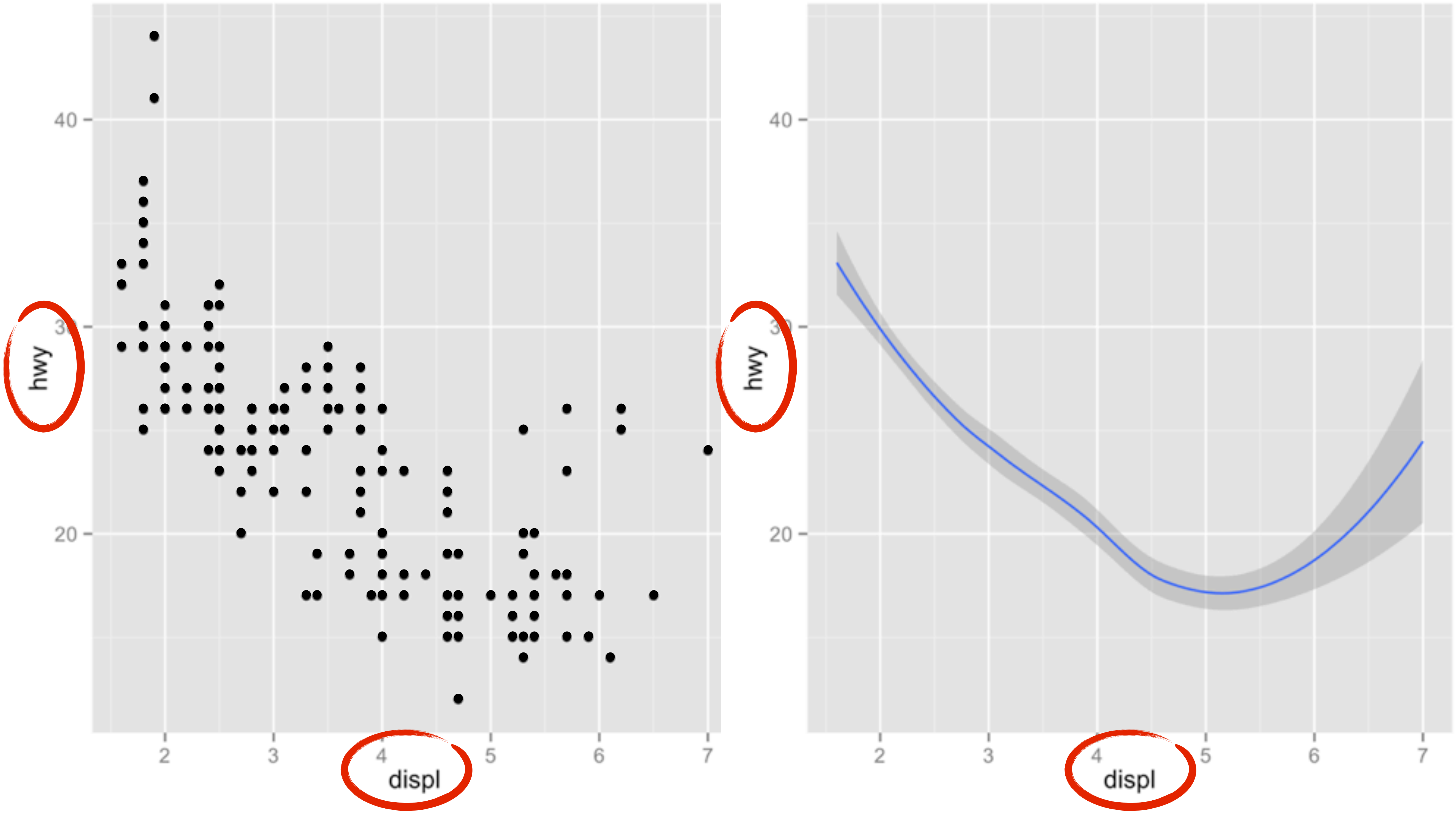
`facet_grid()`: 2d grid, rows ~ cols, . for no split

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

Geoms

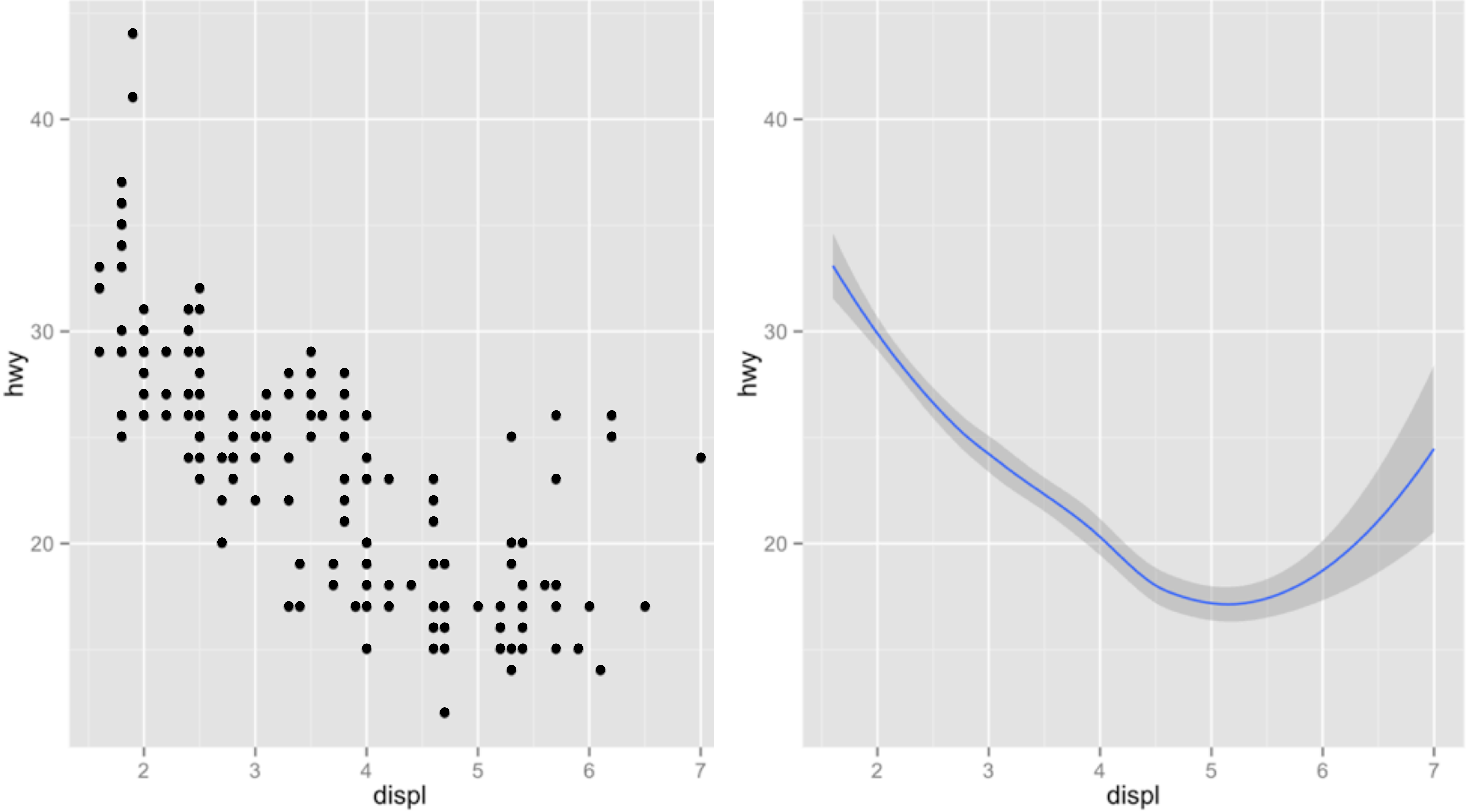
How are these plots similar?

Same: x var, y var, data



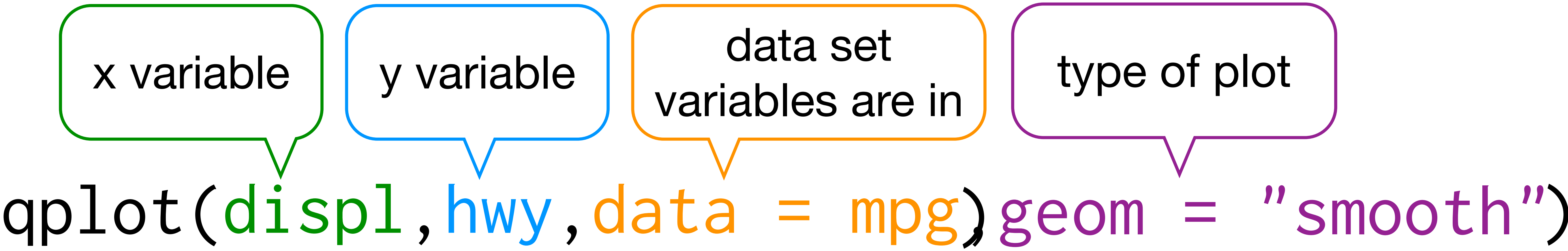
How are these plots different?

Different: "type" of plot
i.e, what plot draws
i.e, geometric object



Geometric object

the "type" of graph, or
what the graph draws



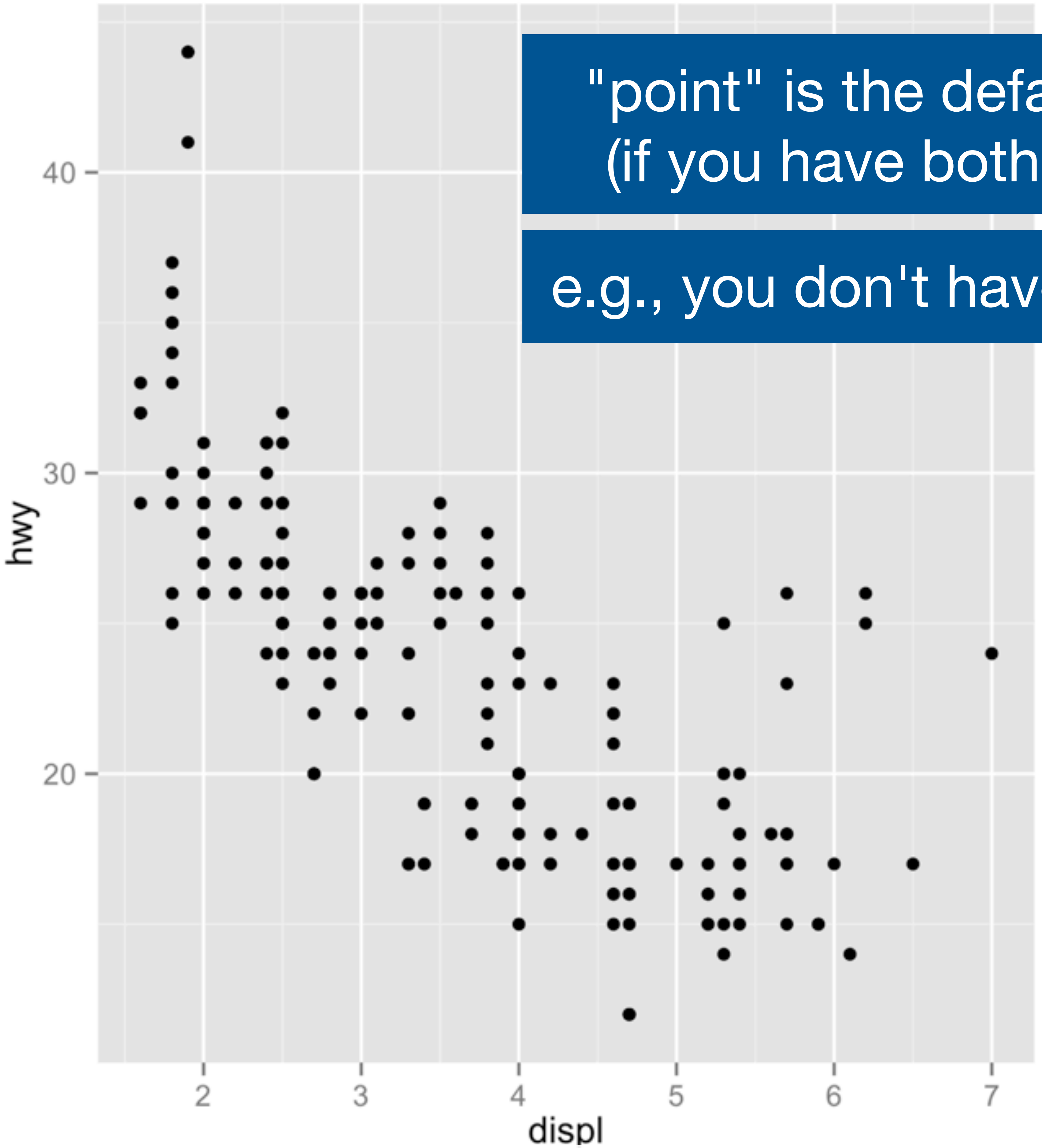
x variable

y variable

data set variables are in

type of plot

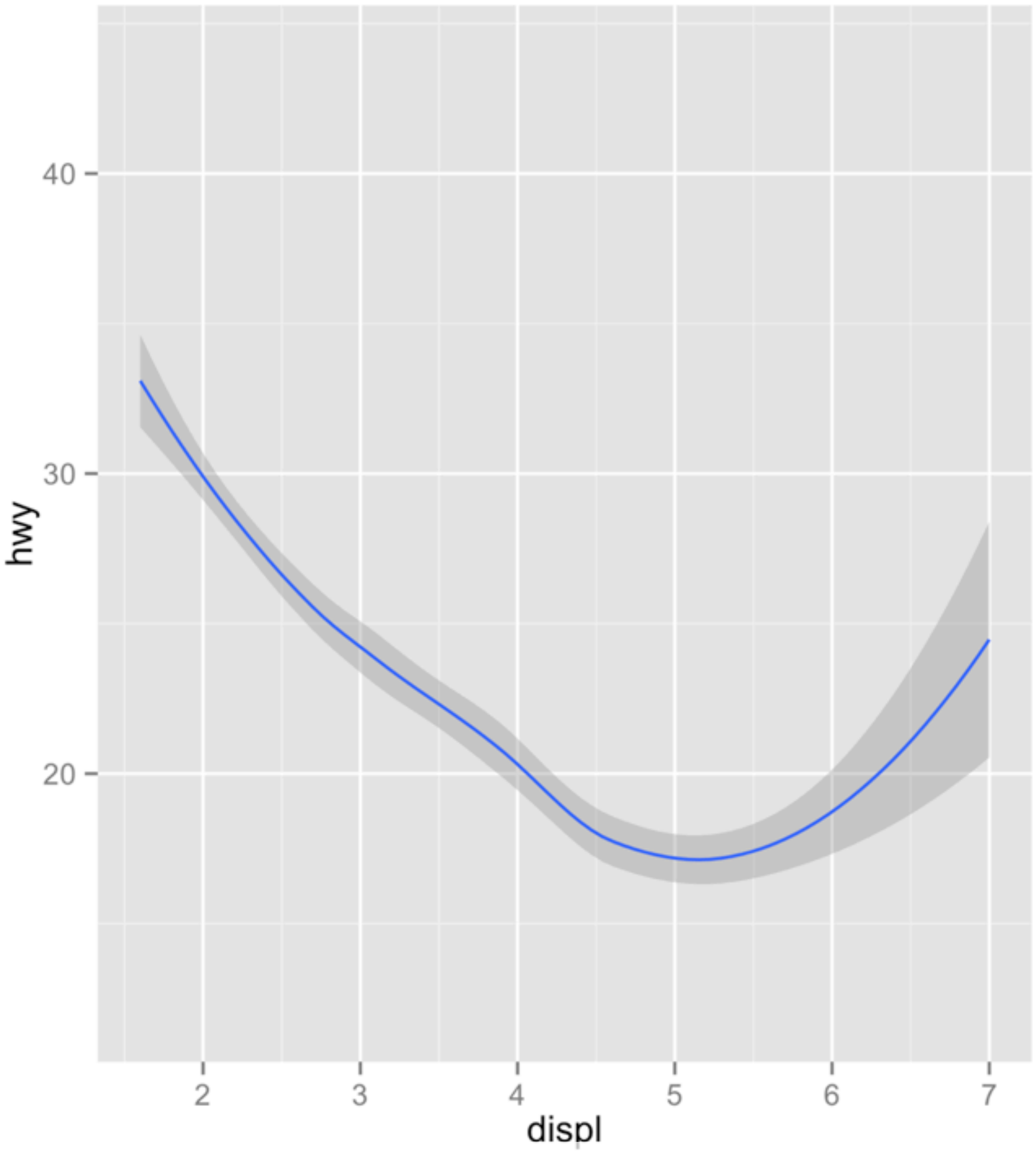
```
qplot(displ,hwy,data = mpg),geom = ”smooth”)
```



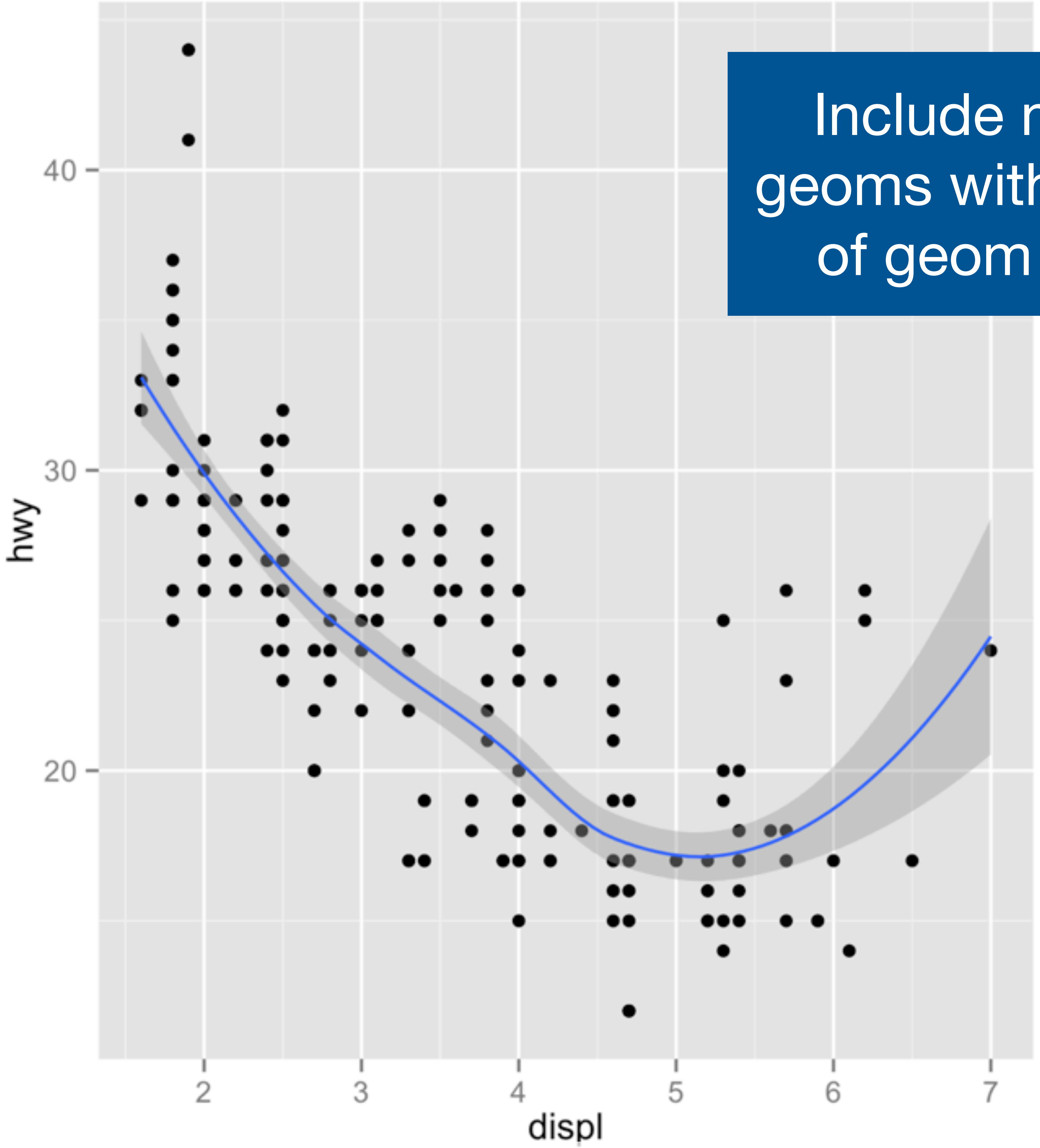
"point" is the default geom
(if you have both x and y)

e.g., you don't have to type it

```
qplot(displ, hwy, data = mpg), geom = ”point”)
```



```
qplot(displ, hwy, data = mpg, geom = "smooth")
```



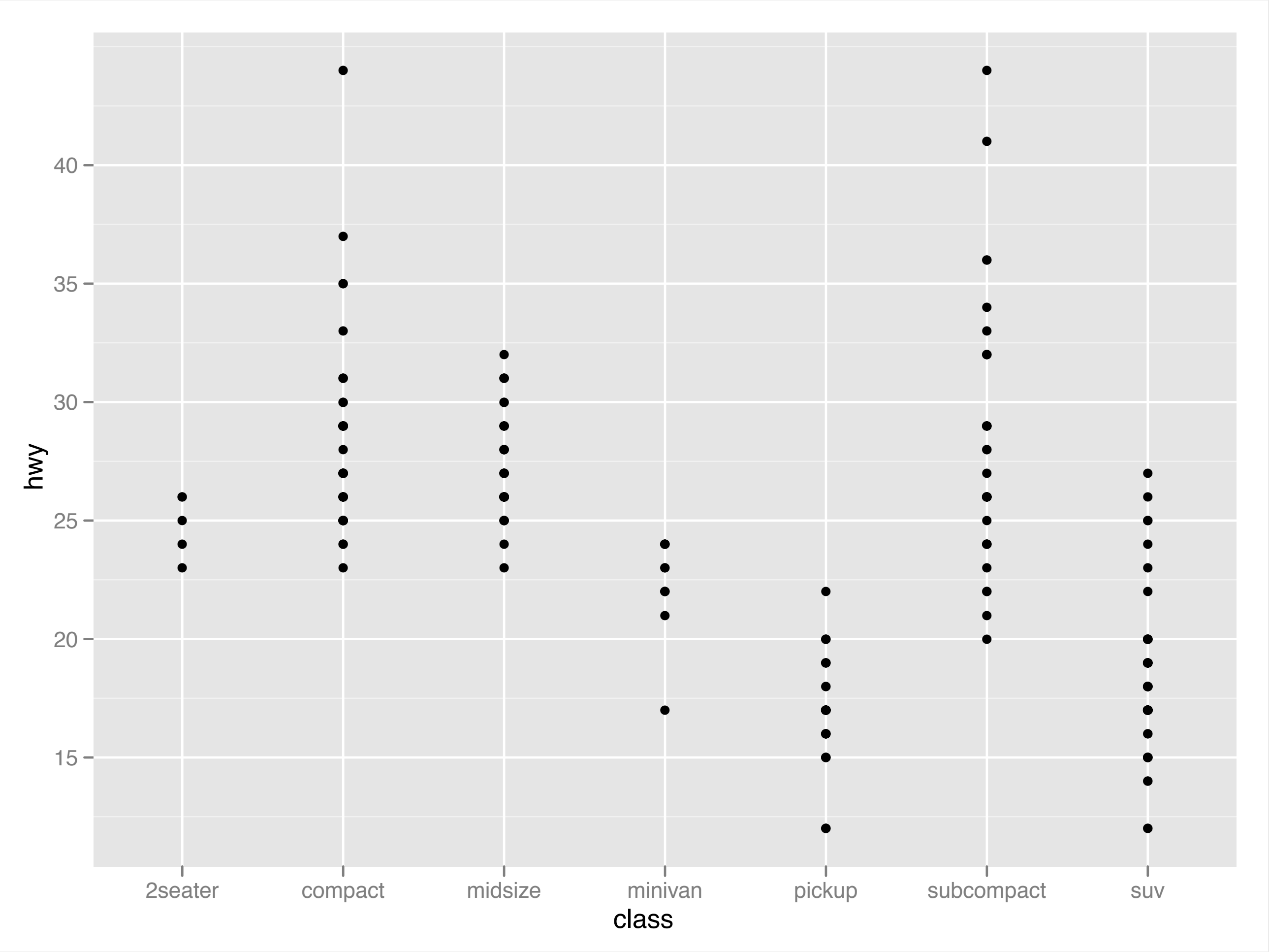
Include multiple geoms with a vector of geom names

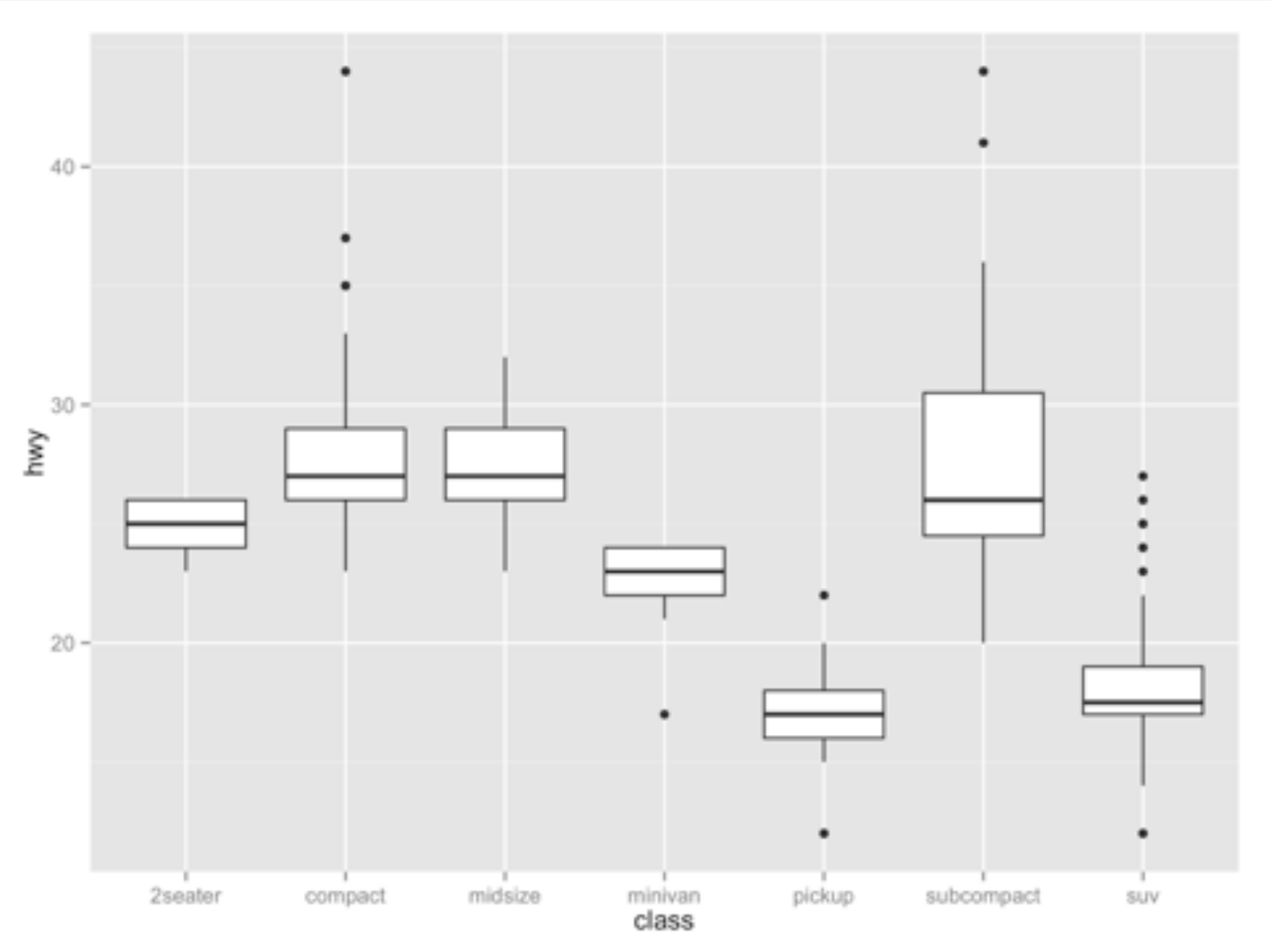
```
qplot(displ, hwy, data = mpg, geom = c("point", "smooth"))
```

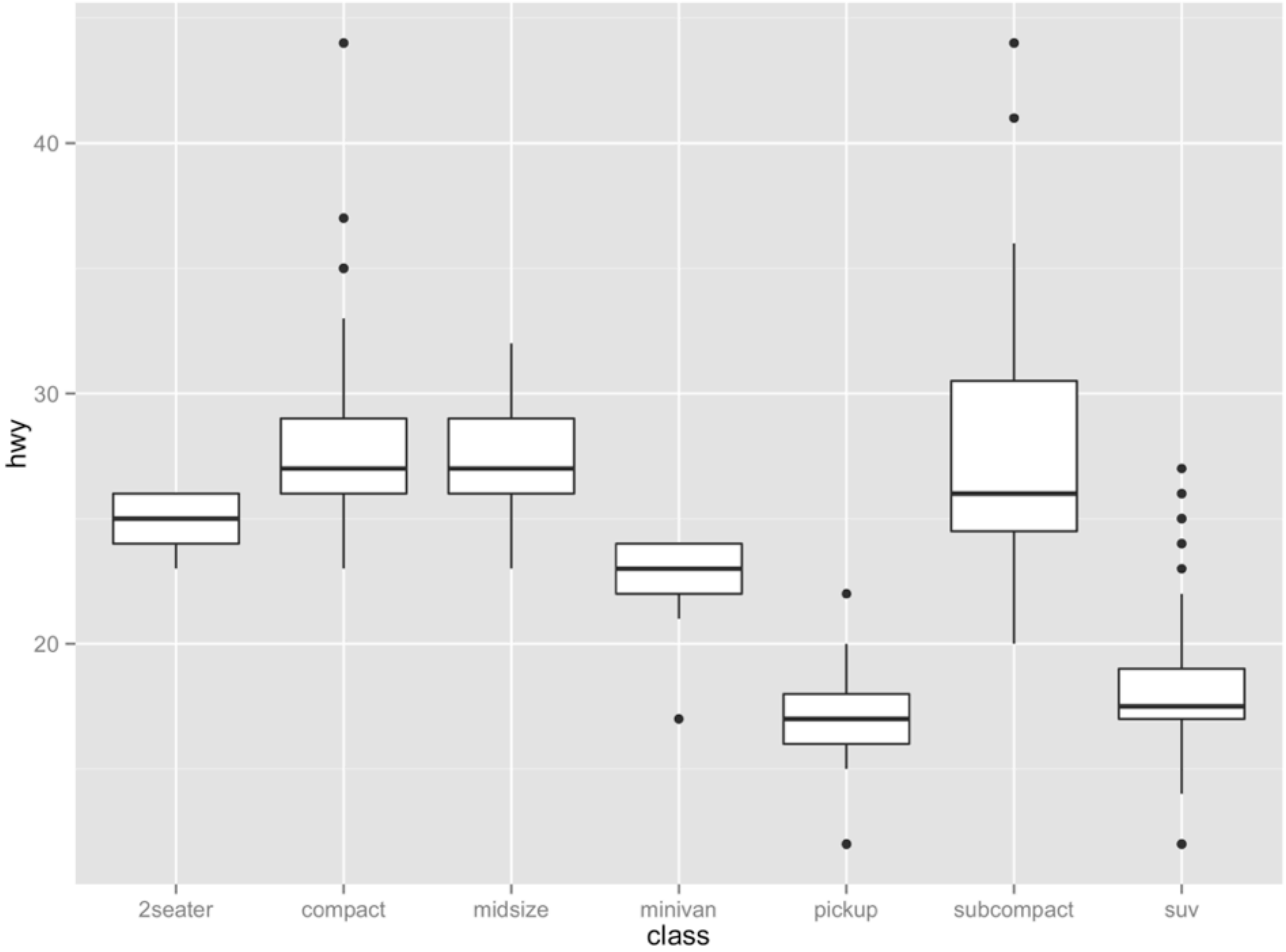
Your turn

How would you replace this scatterplot with one that draws boxplots? Try out your best guess.

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





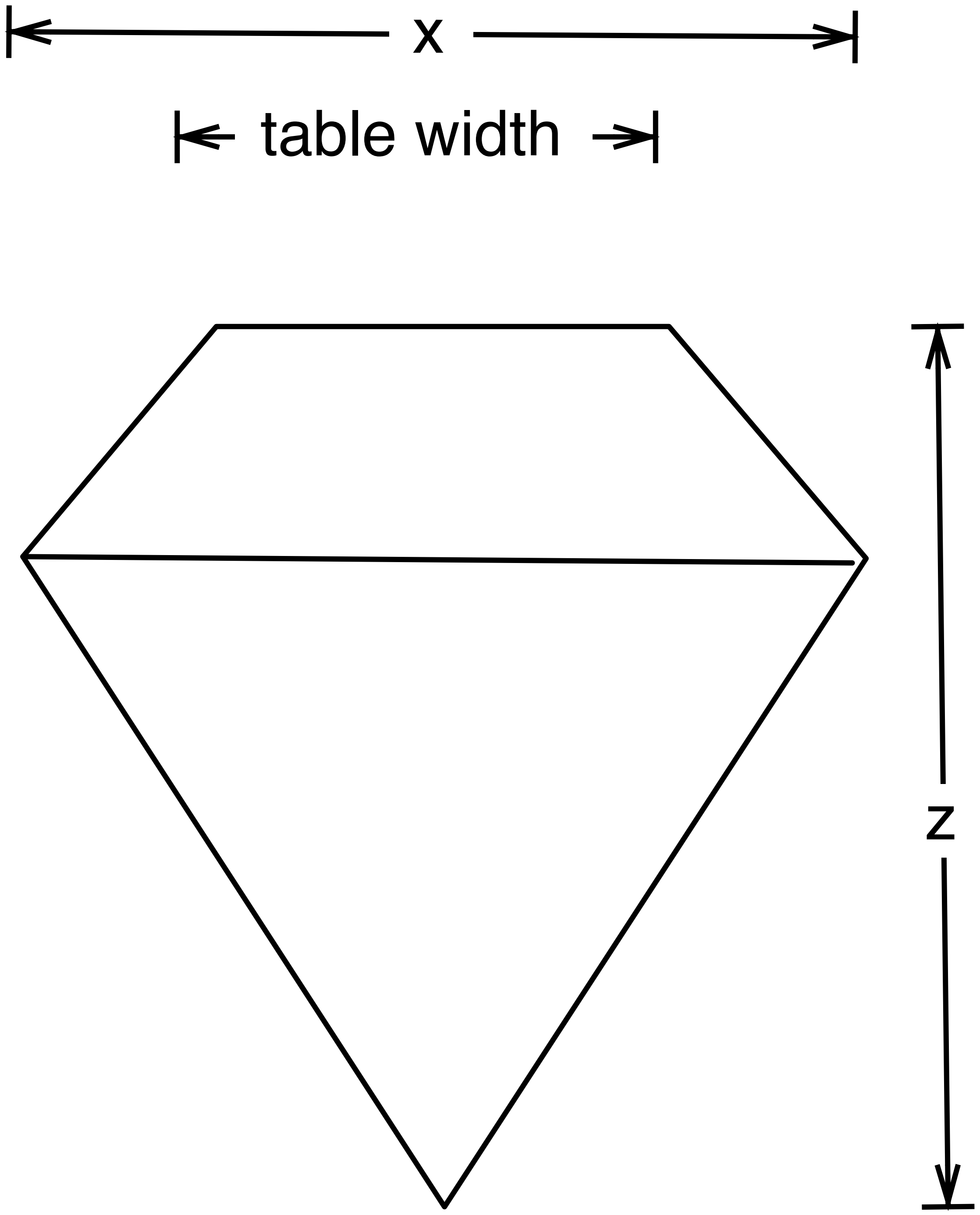


```
qplot(class, hwy, data = mpg, geom = "boxplot")
```

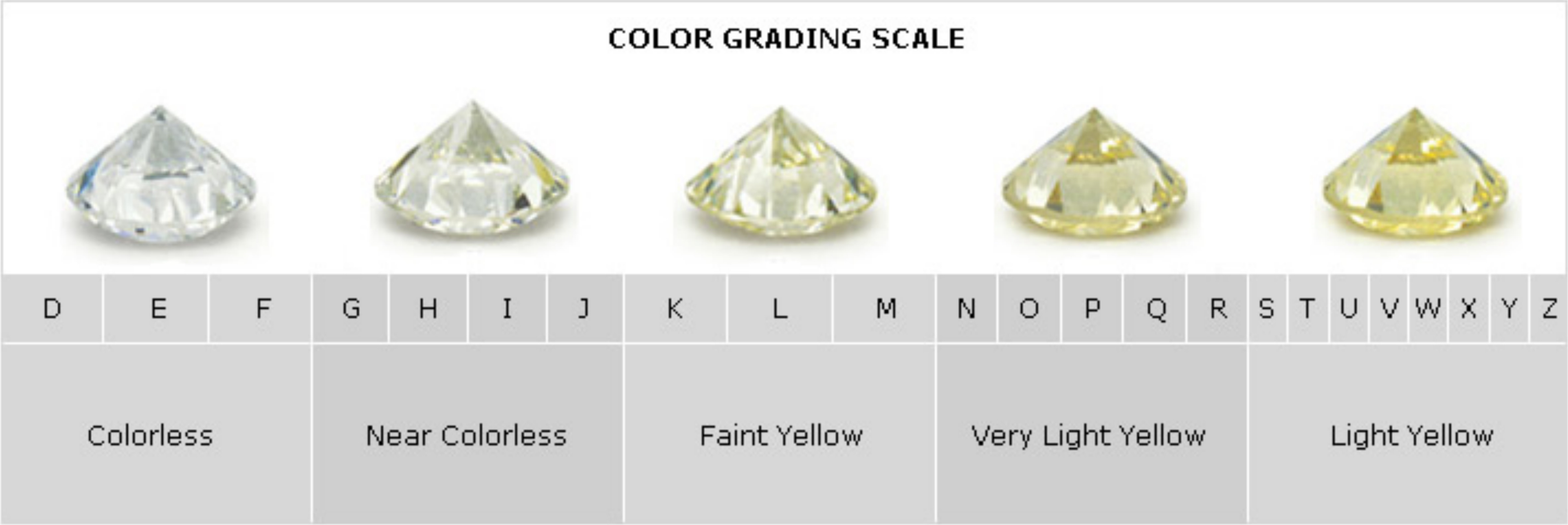
Diamonds

Diamonds data

- ~**54,000** round diamonds from http://www.diamondse.info/
- Carat, color, clarity, cut
- Total depth, table, depth, width, height
- Price



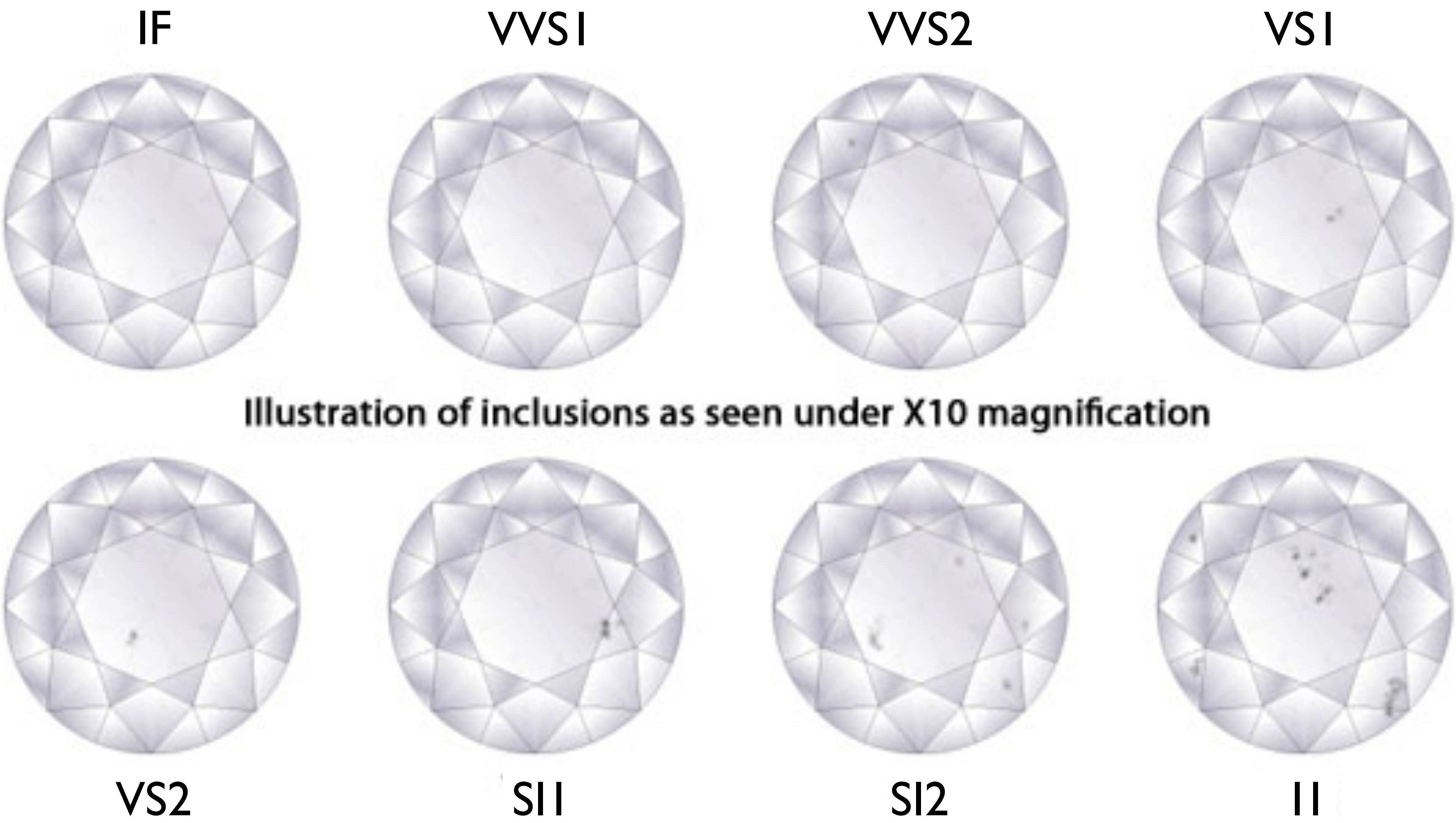
depth = z / diameter
table = table width / x * 100



COLOR GRADING SCALE

Colorless	Near Colorless	Faint Yellow	Very Light Yellow	Light Yellow
D E F	G H I J	K L M	N O P Q R	S T U V W X Y Z

http://www.mlcld.com/colour.aspx



IF VVS1 VVS2 VS1

Illustration of inclusions as seen under X10 magnification

VS2 SI1 SI2 I1

http://www.thediamondsexperts.com/diamonds-guide

Histogram
&
bar charts

Your turn

What types of plots do the following lines of code return?

```
qplot(x, z, data = diamonds)

qplot(x, data  = diamonds)

qplot(cut, data = diamonds)
```

Default geoms for qplot

Two variables → scatterplot (point)

One continuous variable → histogram

One categorical variable → bar chart

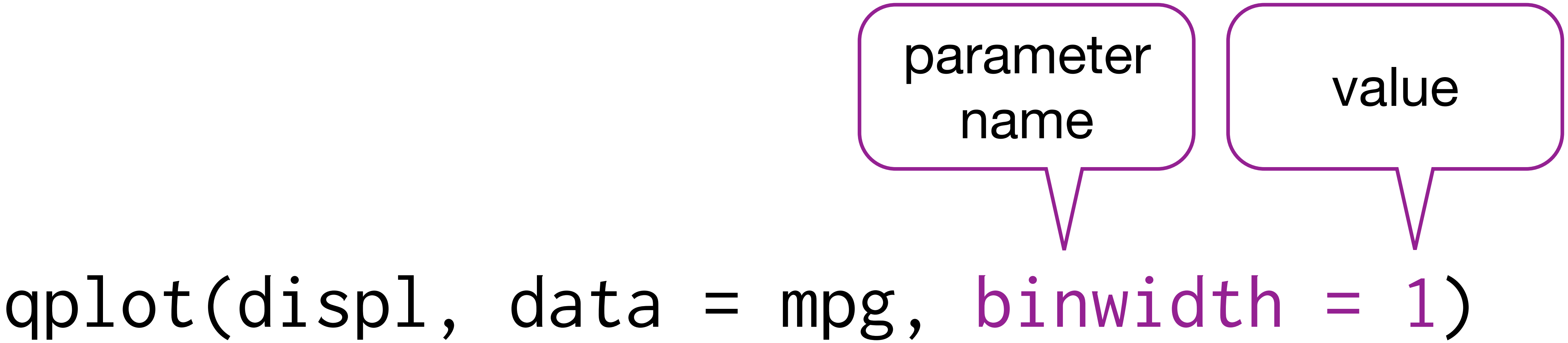
Parameters

Similar to aesthetics.

A parameter is input that controls the appearance of the graph, *but does not map appearance to data.*

e.g. binwidths in a histogram

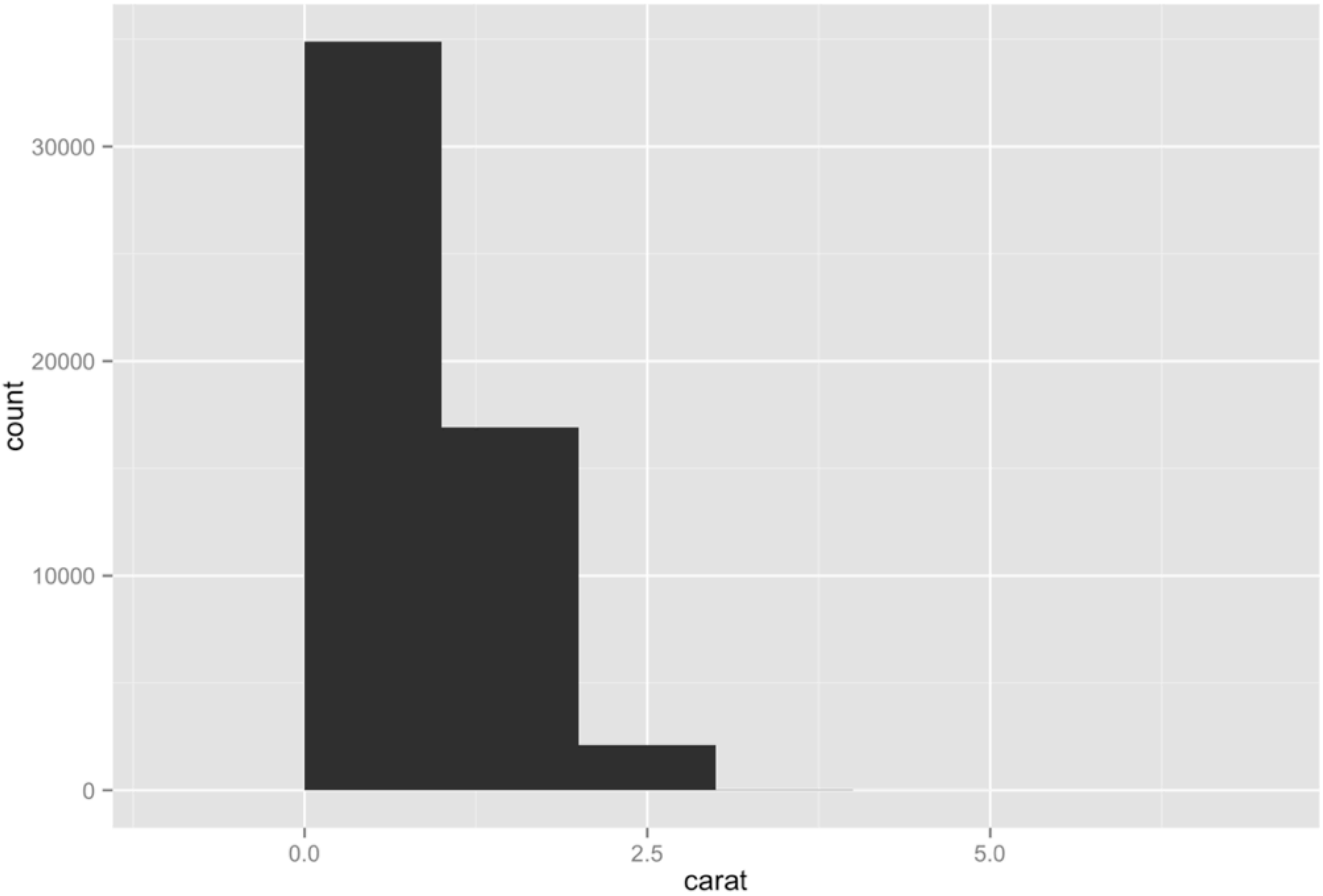
Parameters



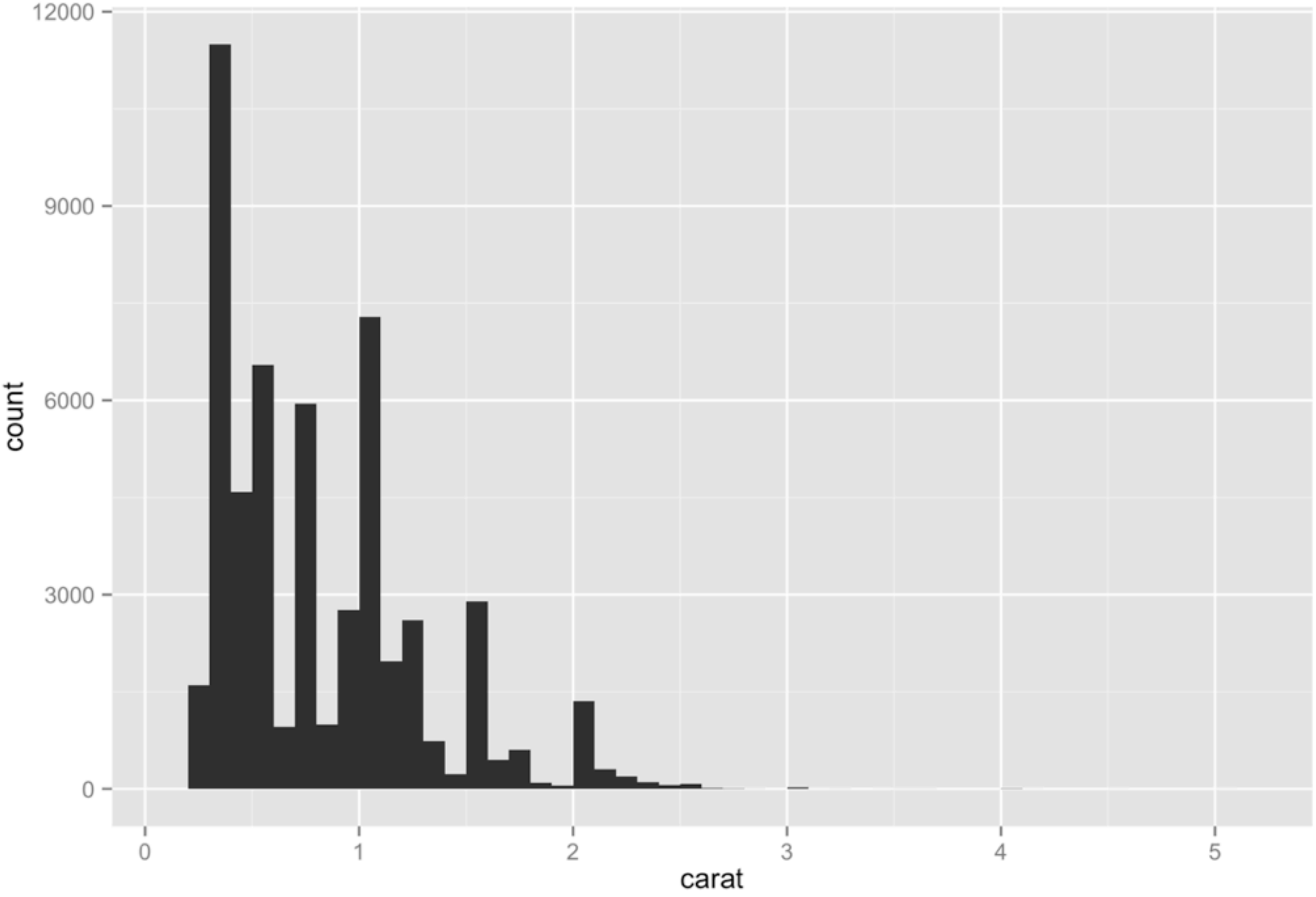
parameter name

value

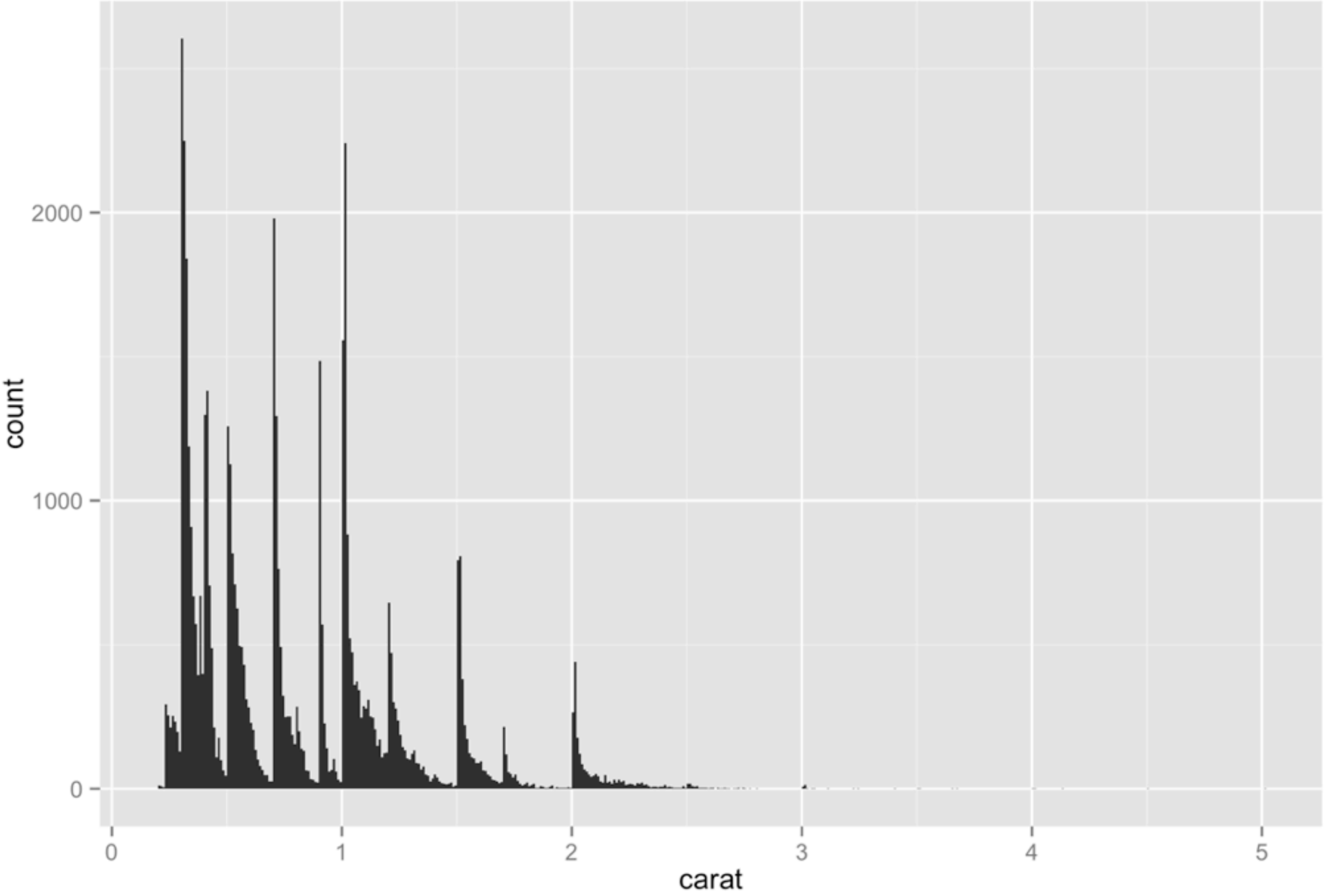
```
qplot(displ, data = mpg, binwidth = 1)
```



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

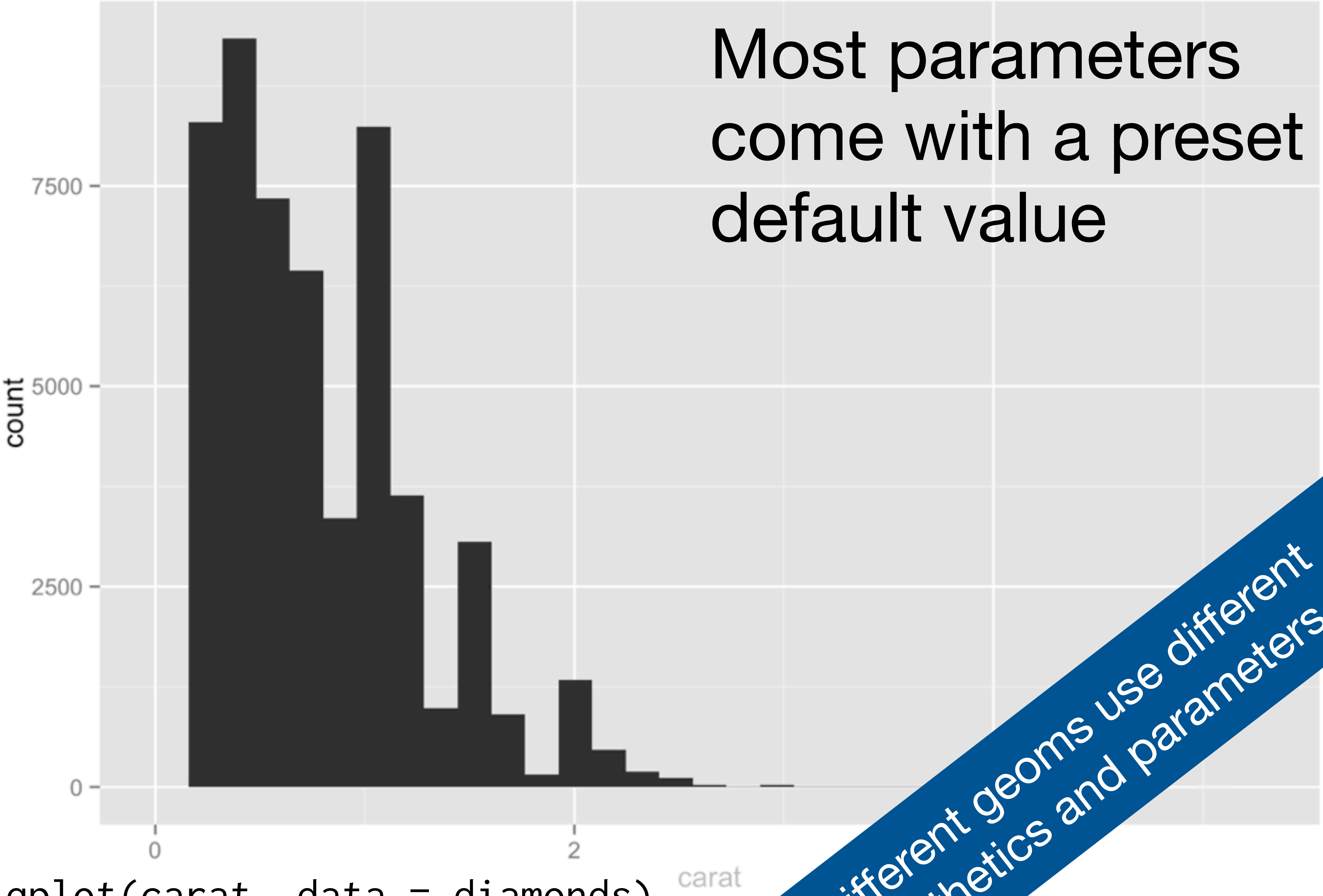


```
qplot(carat, data = diamonds, binwidth = 0.1)
```



```
qplot(carat, data = diamonds, binwidth = 0.01)
```

Most parameters come with a preset default value



```
qplot(carat, data = diamonds)
stat_bin: binwidth defaulted to
```

Different geoms use different aesthetics and parameters

Your turn

Examine the distribution of price at different binwidths.

Do you spot anything odd?

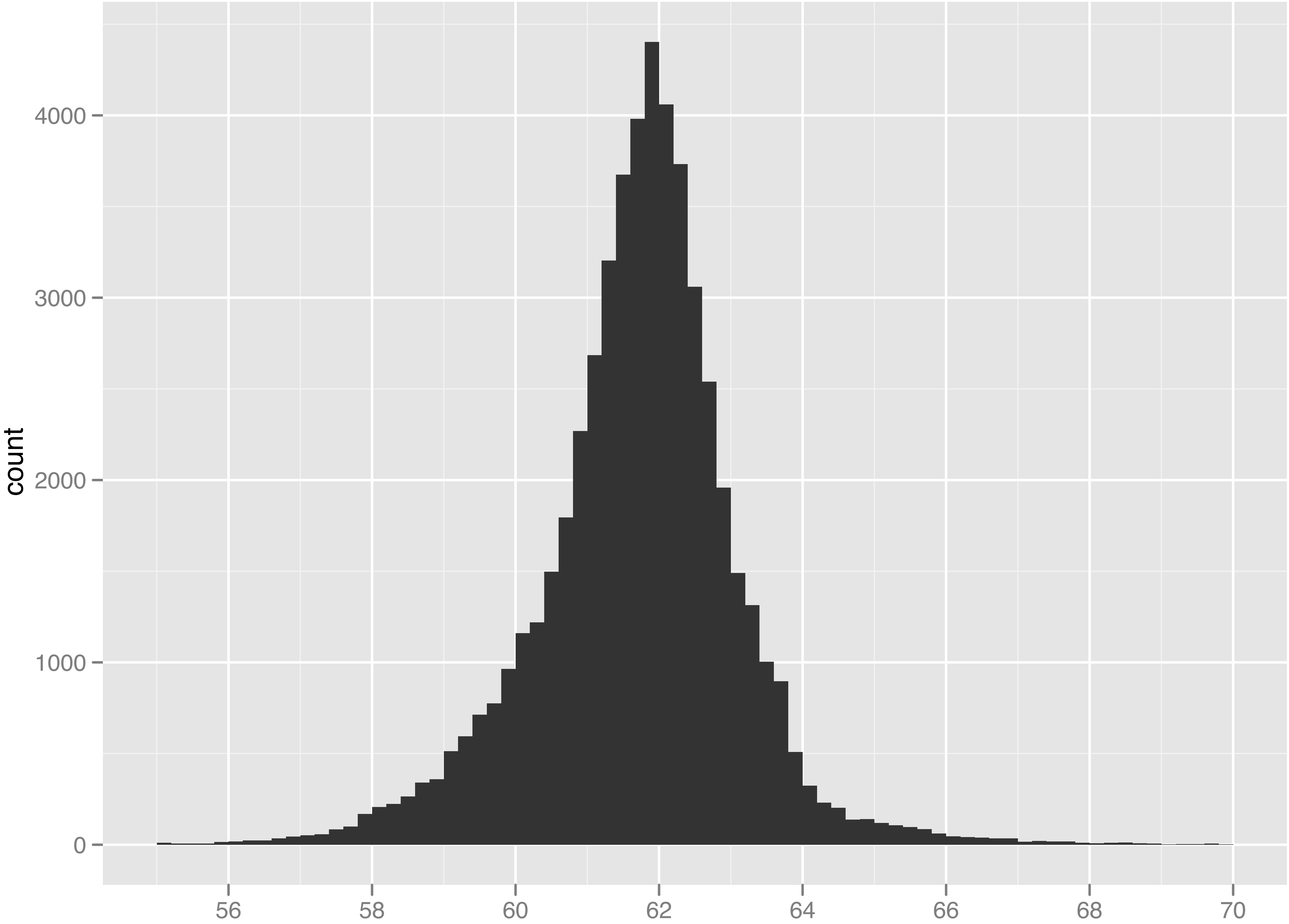
```
qplot(price, data = diamonds)
```

Hint: 0 < price < 18823.
Do not set binwidth = 1!

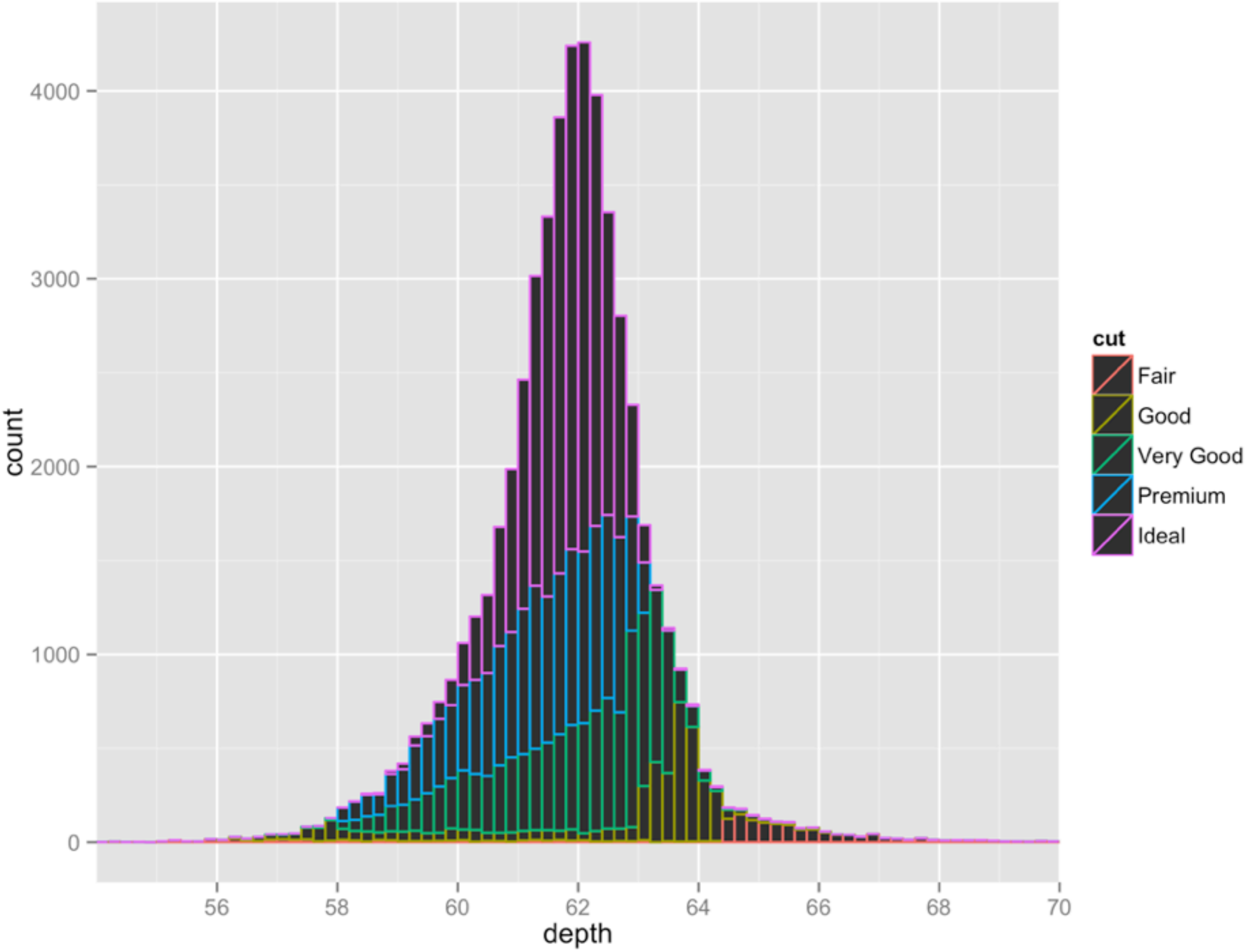
```
qplot(price, data = diamonds)
qplot(price, data = diamonds, binwidth = 500)
qplot(price, data = diamonds, binwidth = 100)
qplot(price, data = diamonds, binwidth = 50)
```

Additional variables

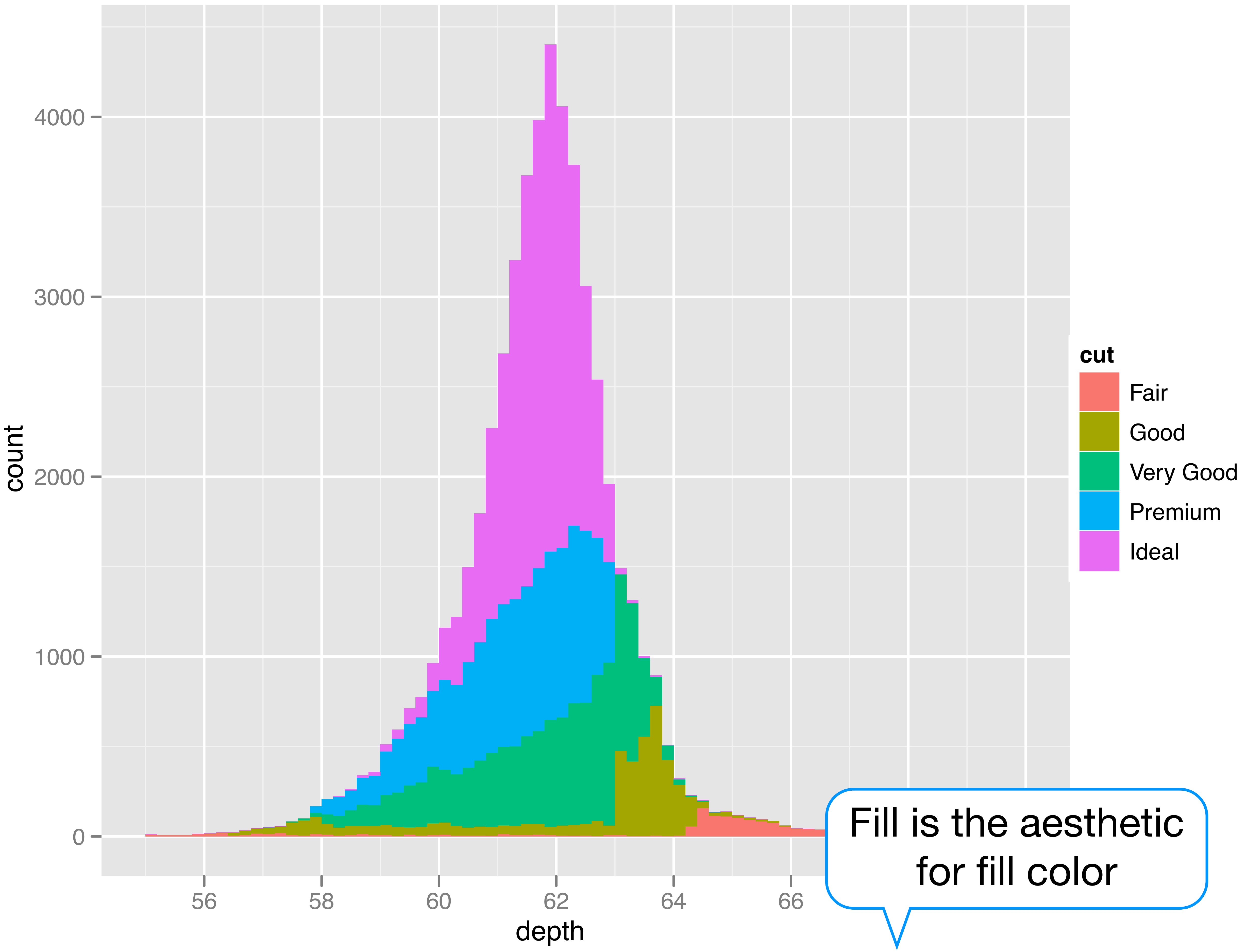
Often switching geoms is more effective than adding aesthetics or faceting to a histogram



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

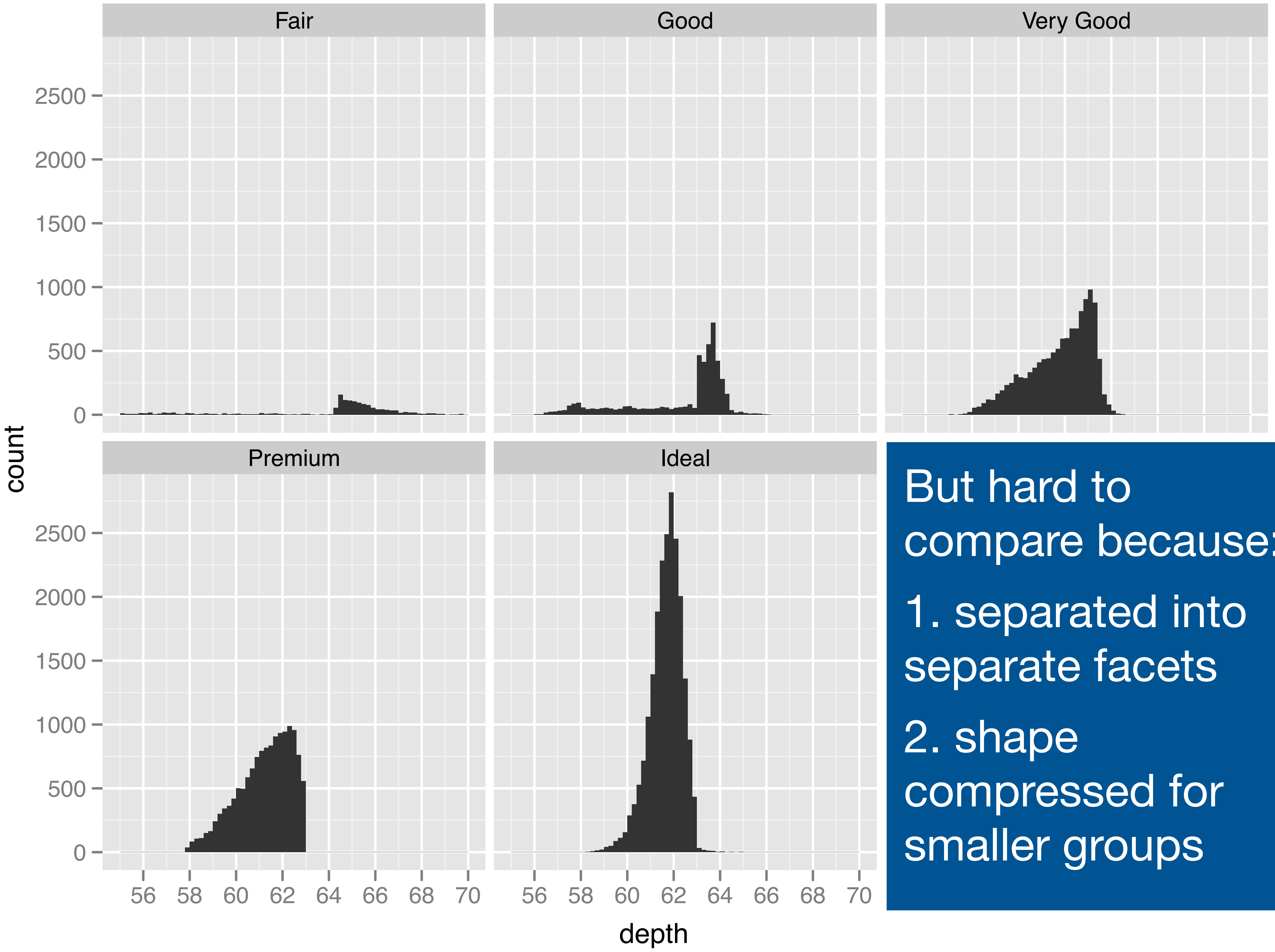


```
qplot(depth, data = diamonds, binwidth = 0.2, color = cut)
```



Fill is the aesthetic for fill color

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

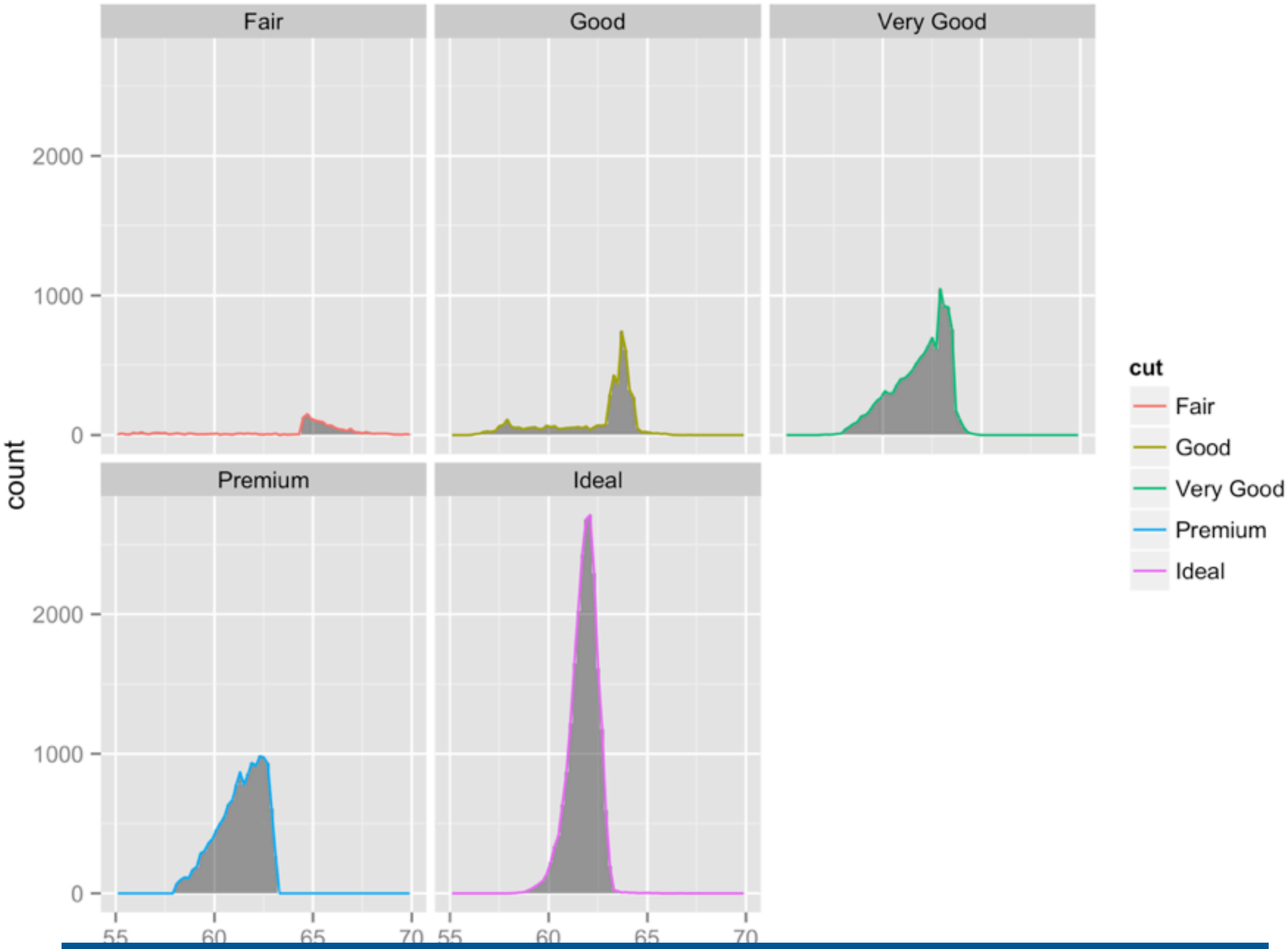


But hard to compare because:

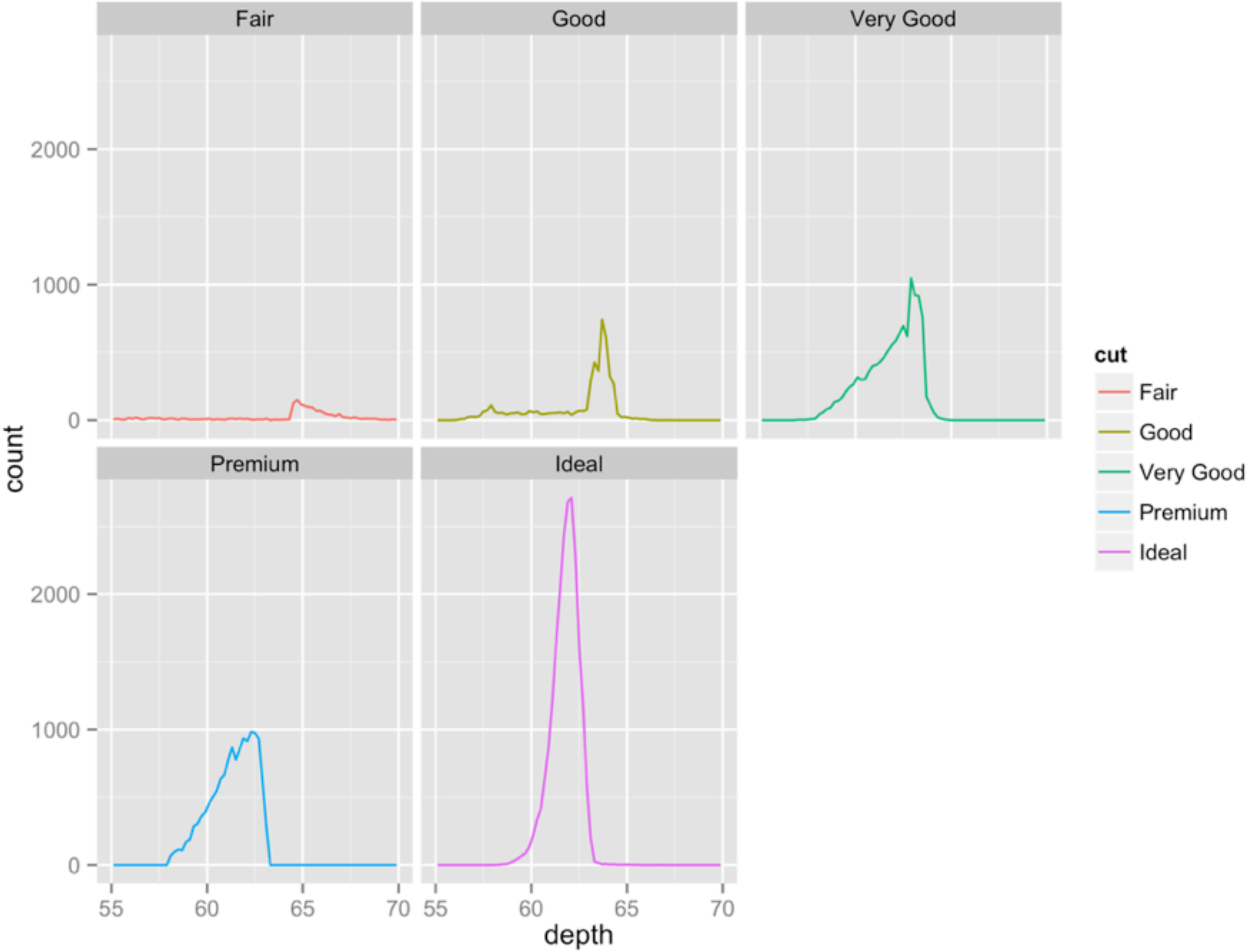
1. separated into separate facets

2. shape compressed for smaller groups

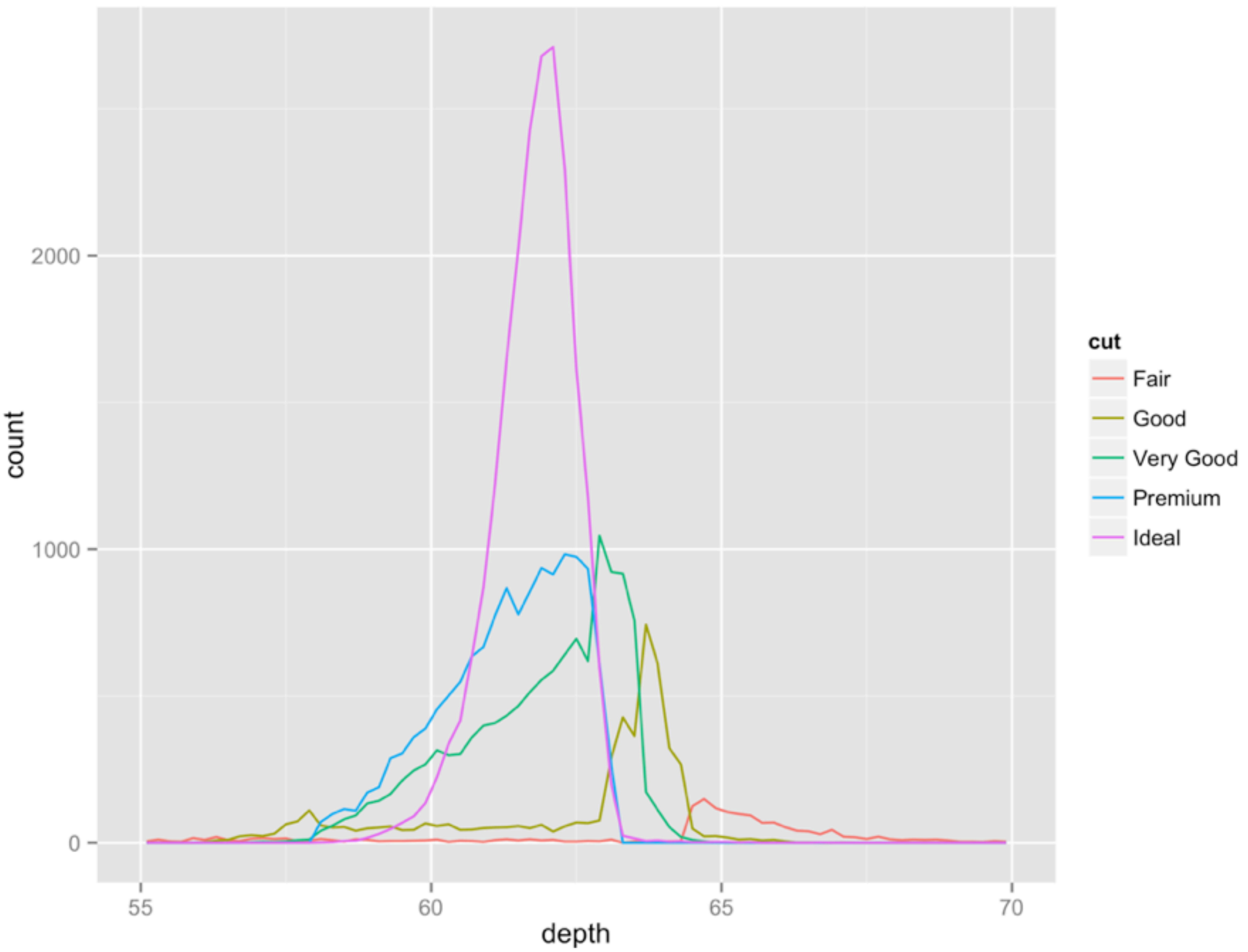
```
qplot(depth, data = diamonds, binwidth = 0.2) +
  facet_wrap(~ cut)
```



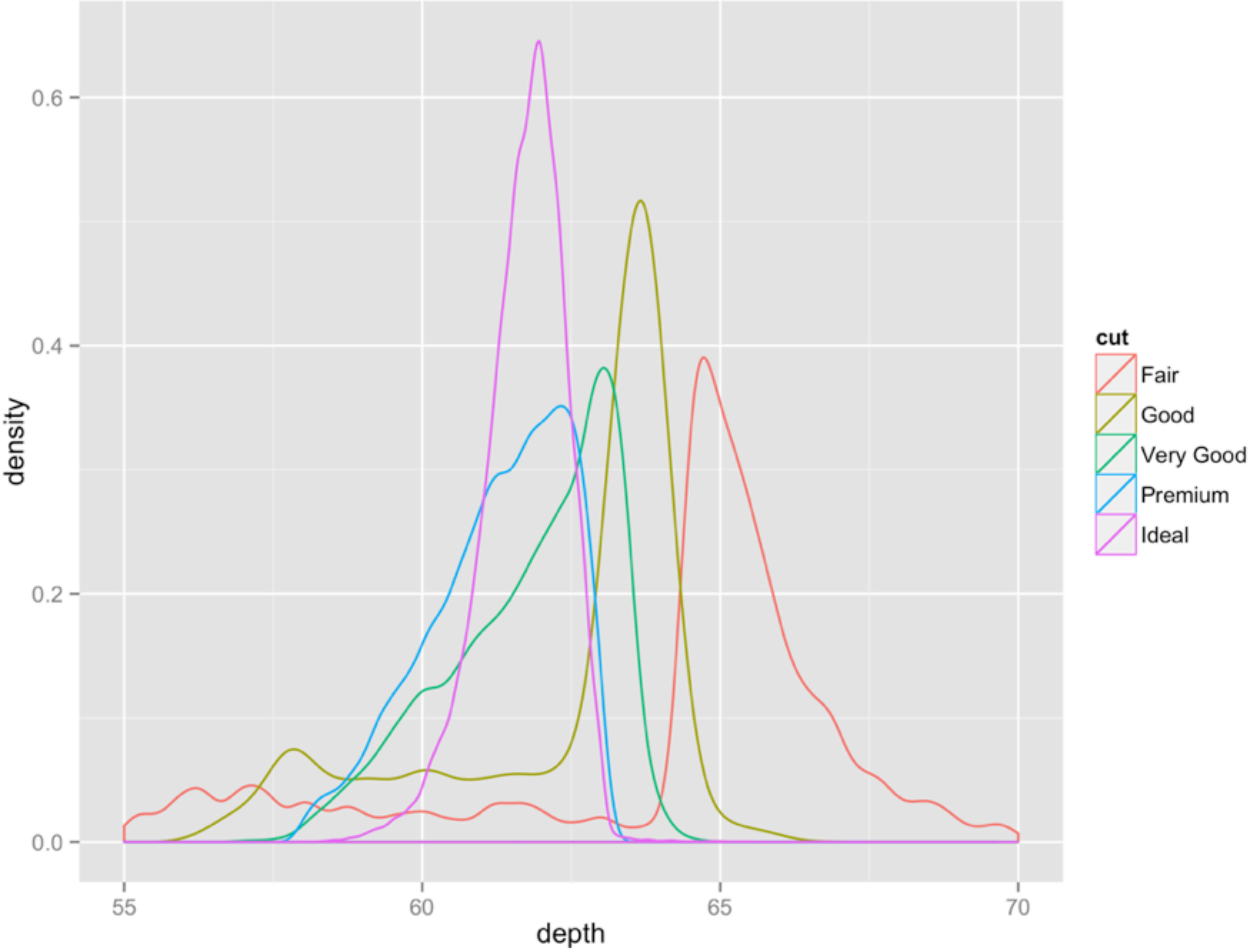
What if we just drew a line along the tops of the histograms, and threw away the bars?



```
qplot(depth, data = diamonds, geom = "freqpoly", color = cut,
  binwidth = 0.2) + facet_wrap(~ cut)
```



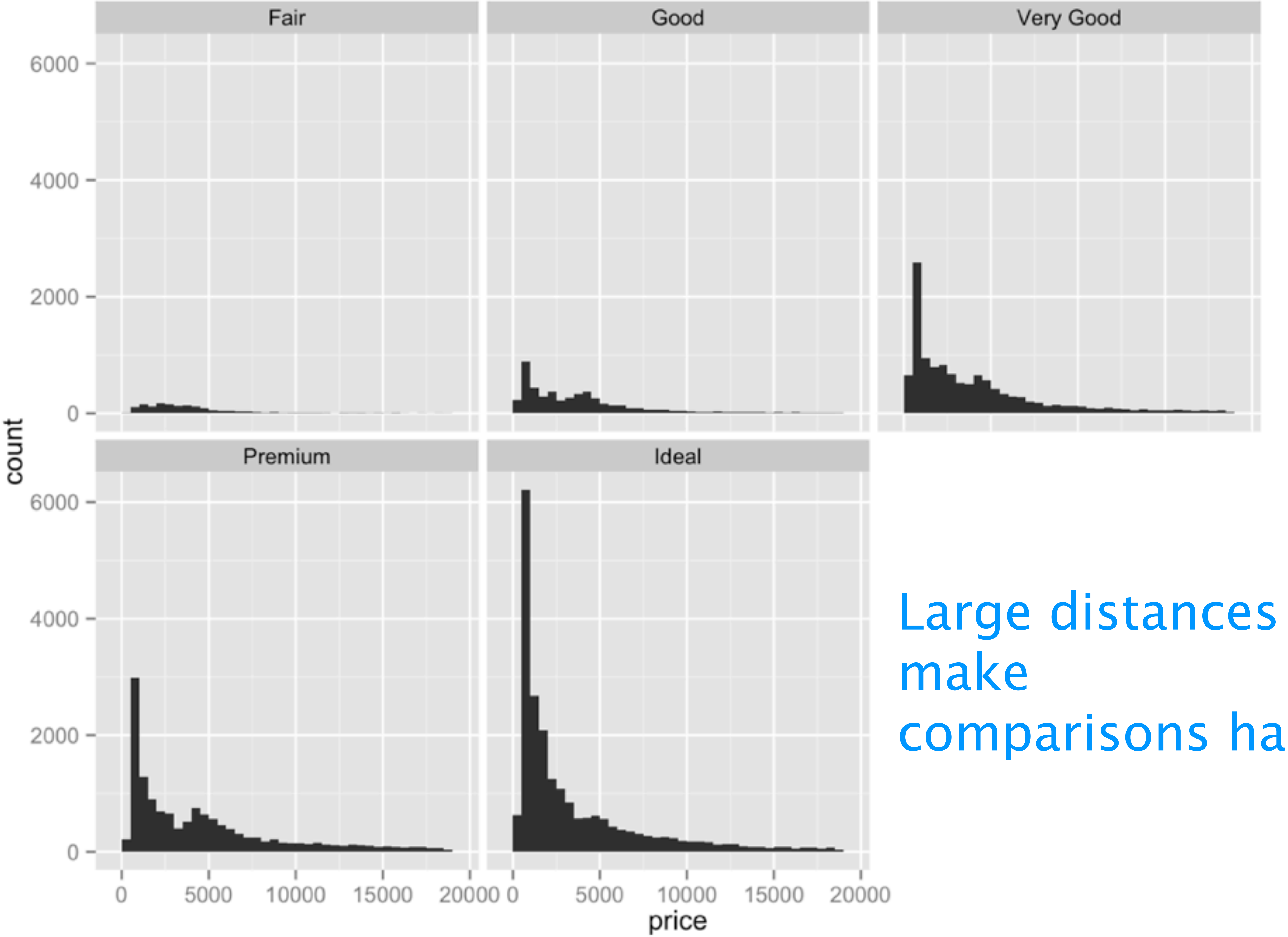
```
qplot(depth, data = diamonds, geom = "freqpoly",
  color = cut, binwidth = 0.2)
```



```
qplot(depth, data = diamonds, geom = "density",
  color = cut)
```

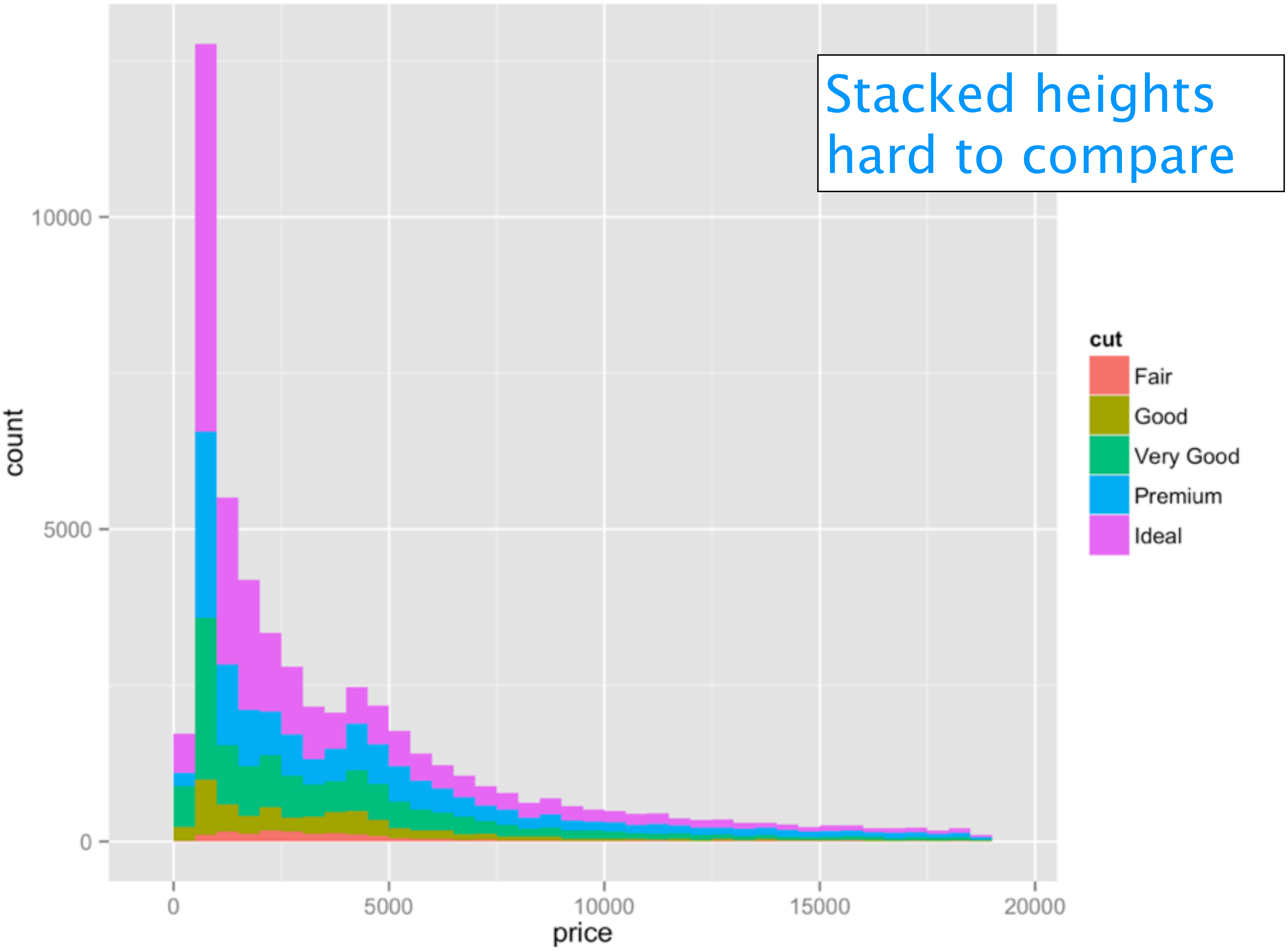
Your turn

Compare the distribution of price for the different cuts. Does anything seem unusual?



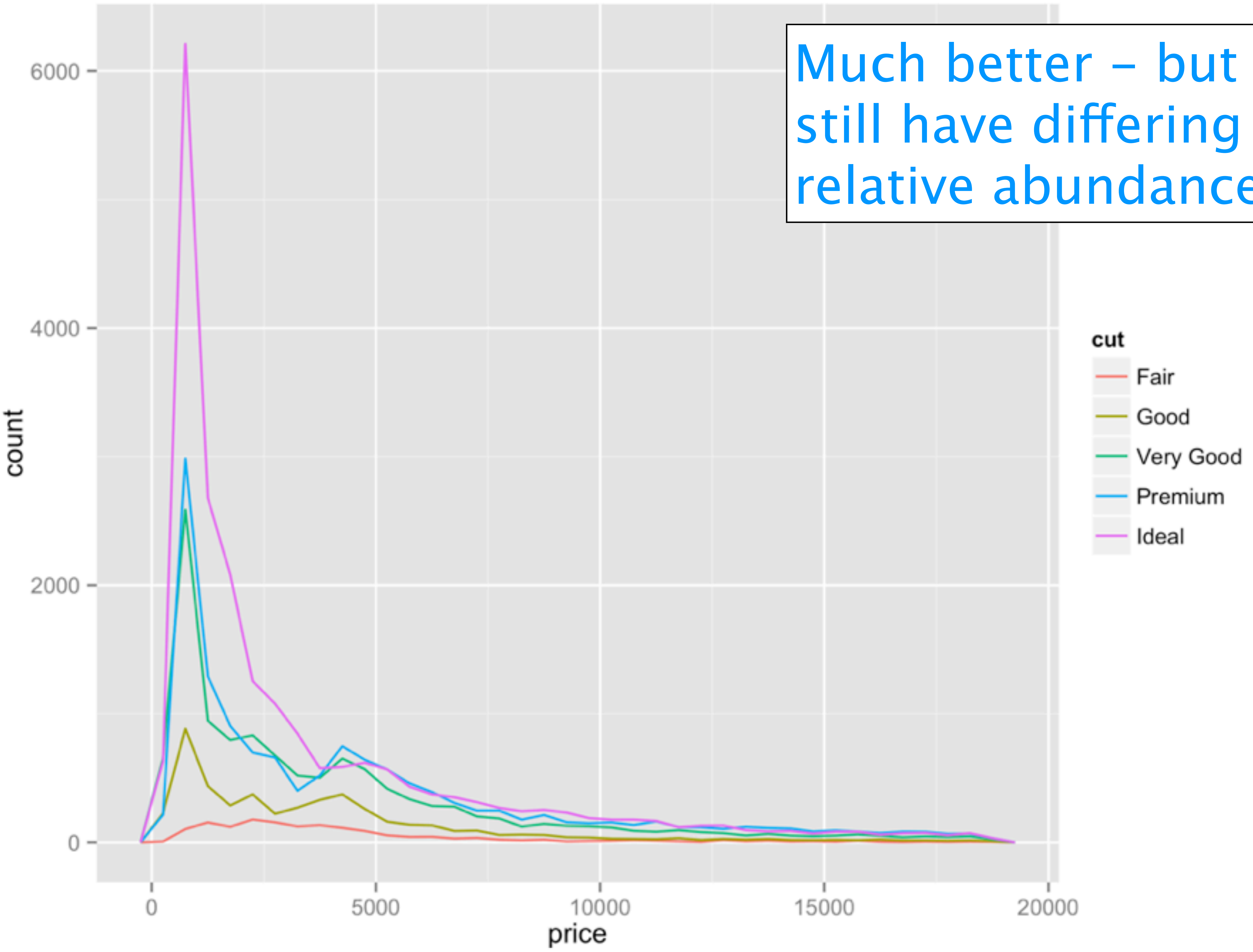
Large distances
make
comparisons hard

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



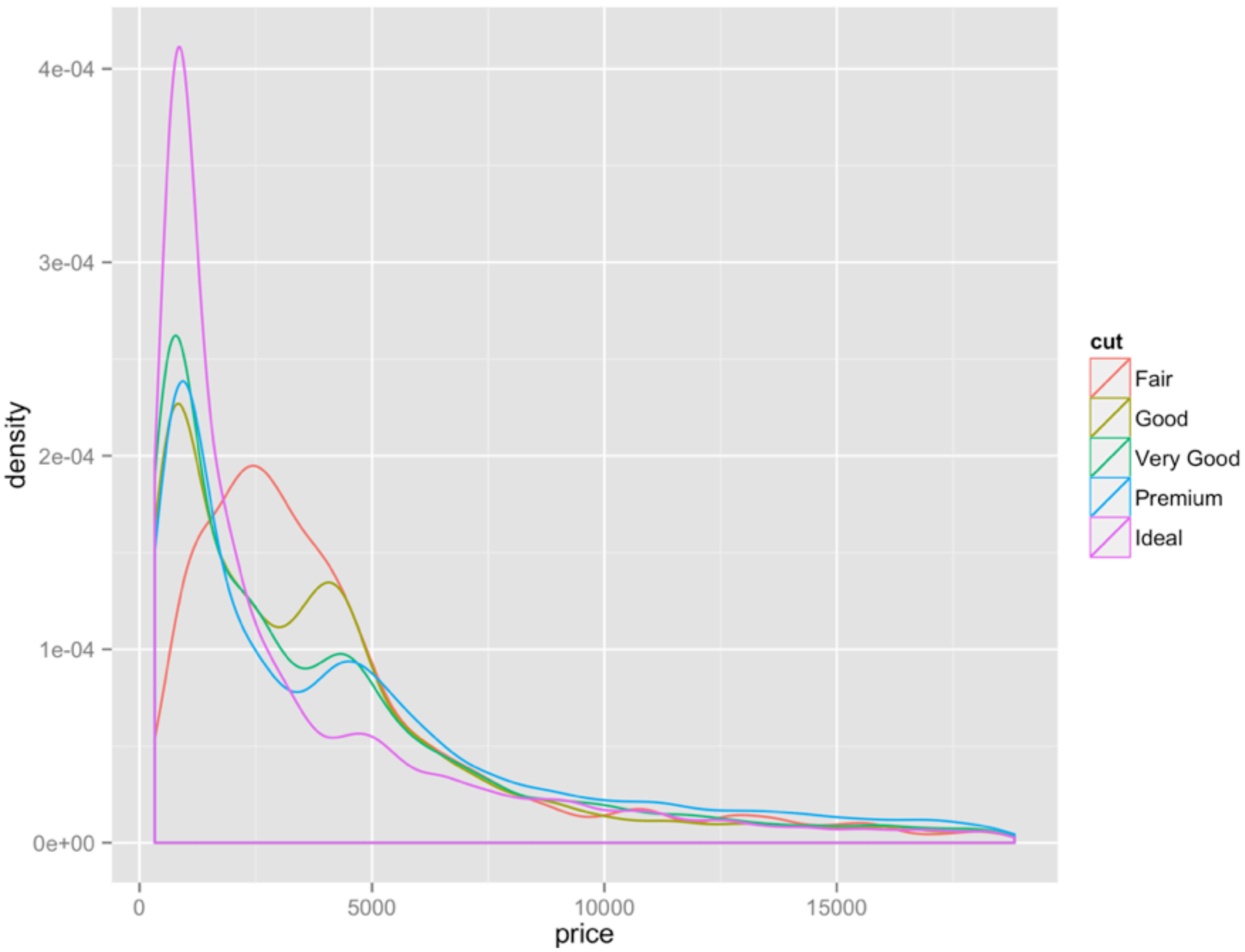
Stacked heights hard to compare

```
qplot(price, data = diamonds, binwidth = 500,
  fill = cut)
```



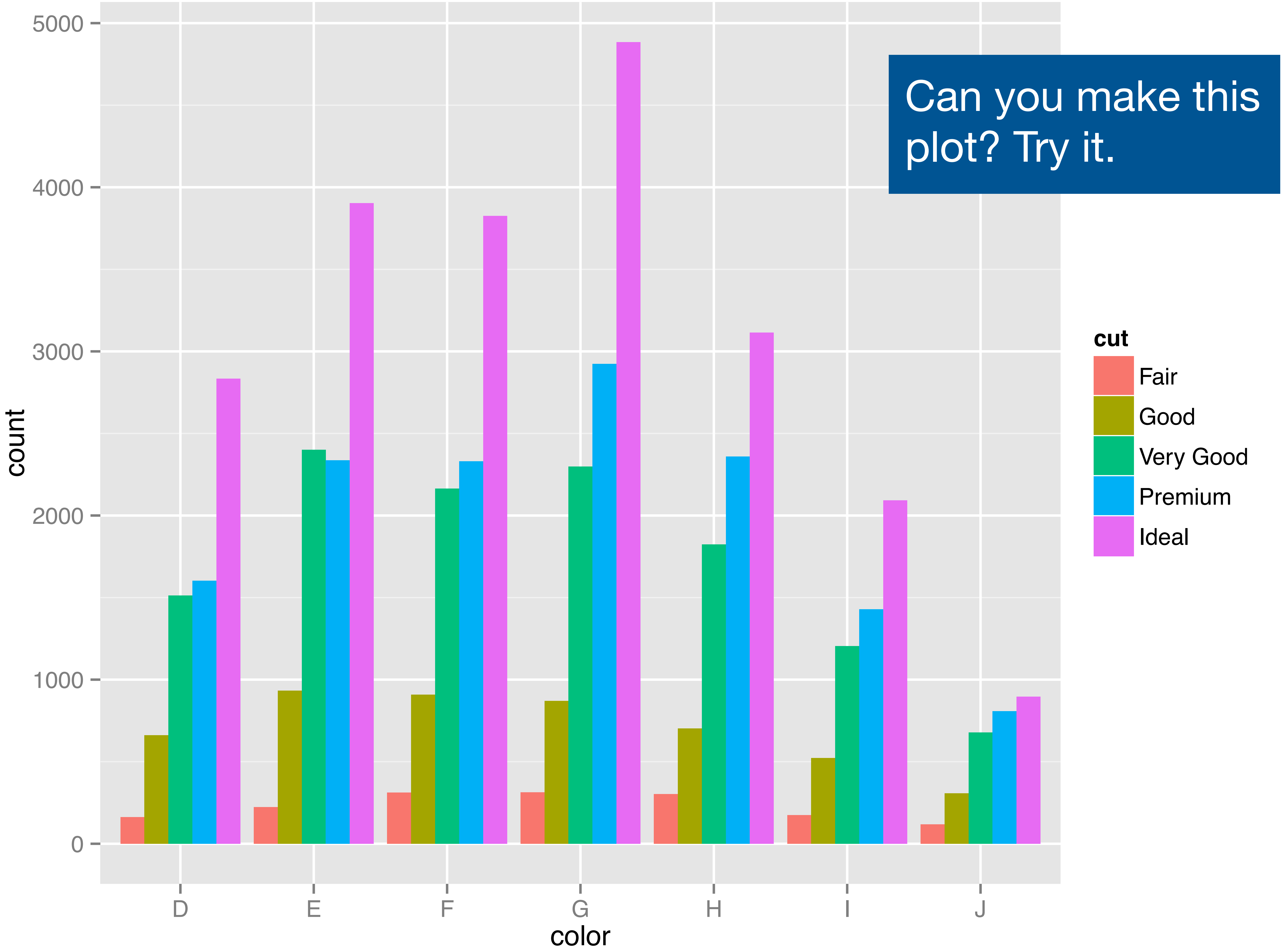
Much better – but still have differing relative abundance

```
qplot(price, data = diamonds, binwidth = 500,
  geom = "freqpoly", color = cut)
```



```
qplot(price, data = diamonds, geom = "density",
  color = cut)
```

Position adjustments



Can you make this plot? Try it.

Your turn

What do each of the position adjustments below do?

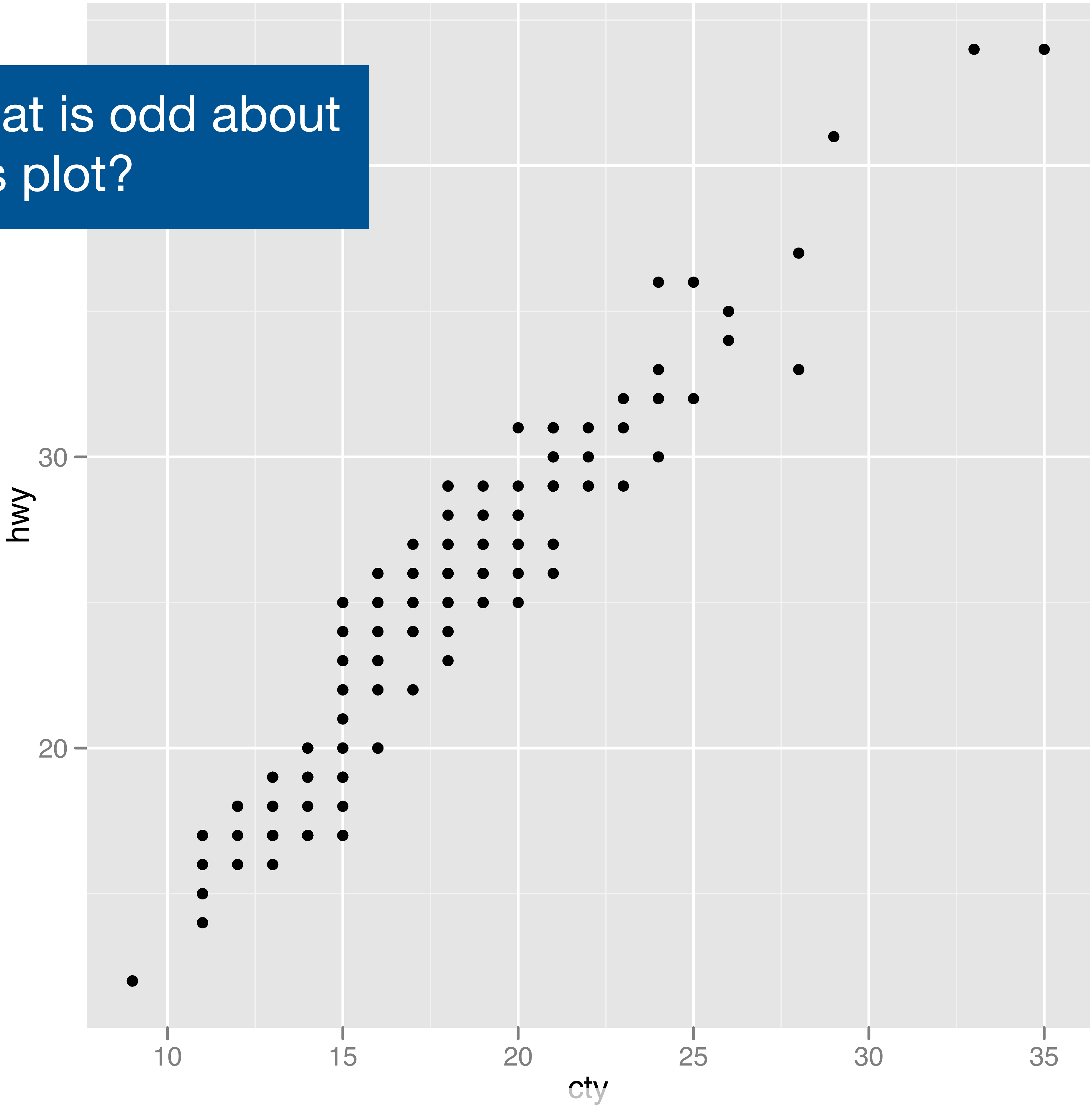
```
qplot(color, data = diamonds, fill = cut,
position = "stack")
```

```
qplot(color, data = diamonds, fill = cut,
position = "dodge")
```

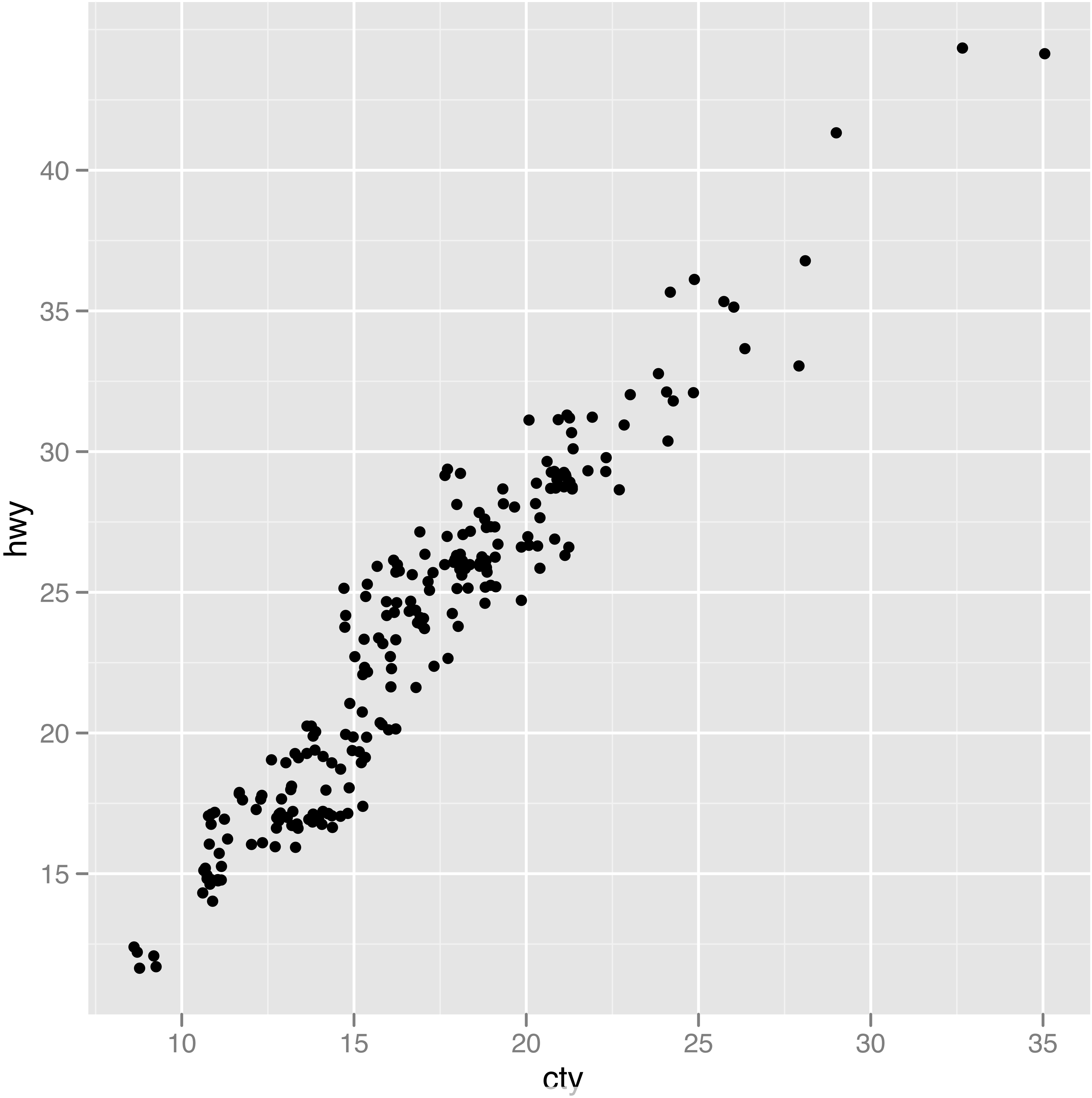
```
qplot(color, data = diamonds, fill = cut,
position = "identity")
```

```
qplot(color, data = diamonds, fill = cut,
position = "fill")
```

What is odd about this plot?



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



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

Position adjustment	effect
identity	no adjustment
stack	colliding objects plotted *above* each other
dodge	colliding objects plotted *beside* each other
fill	available space divided proportionately
jitter	random noise added to placement of each object

Grammar of graphics

Summary

qplot + aesthetics = 10’s of plots

qplot + geoms = 100’s of plots

qplot + geoms + aesthetics = 1000’s of plots

qplot + geoms + aesthetics + position adj = 100,000’s

“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

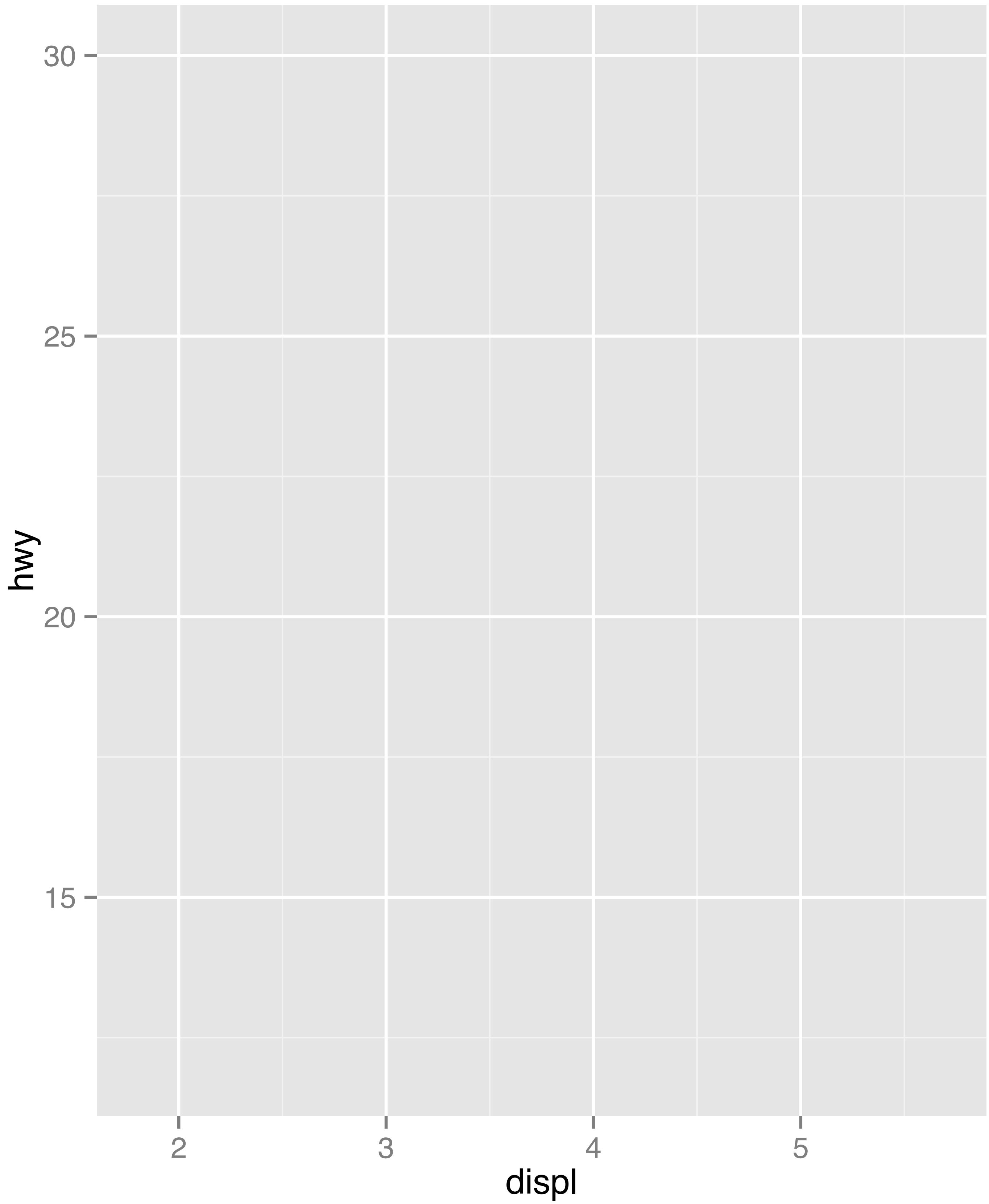
$$m \sum x_i = \sum m x_i$$

The grammar of graphics

An abstraction which makes it easier to describe, make, and reason about graphics.

Developed by Leland Wilkinson, particularly in "*The Grammar of Graphics*" 1999/2005

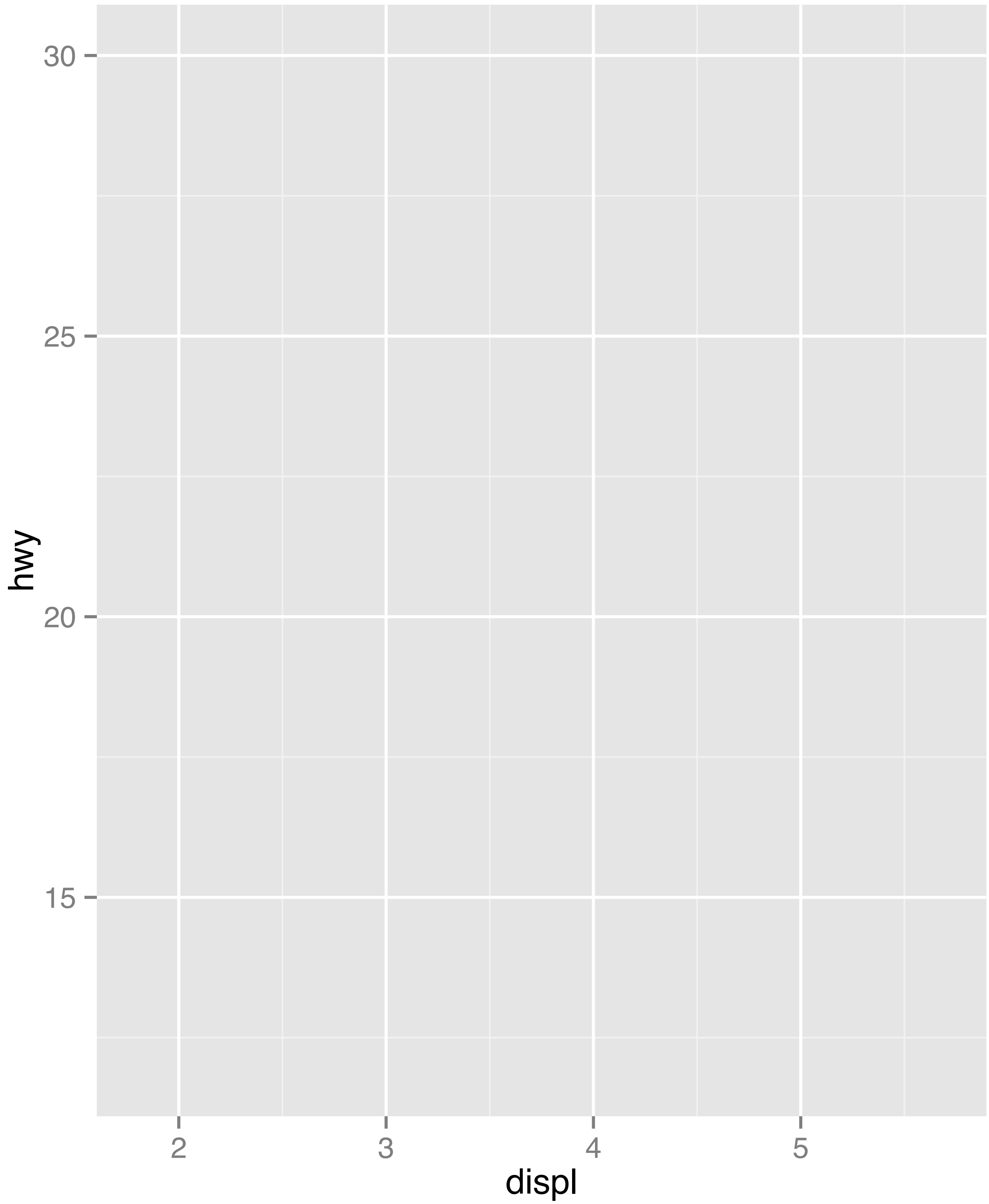
How to build a plot



Coordinate system

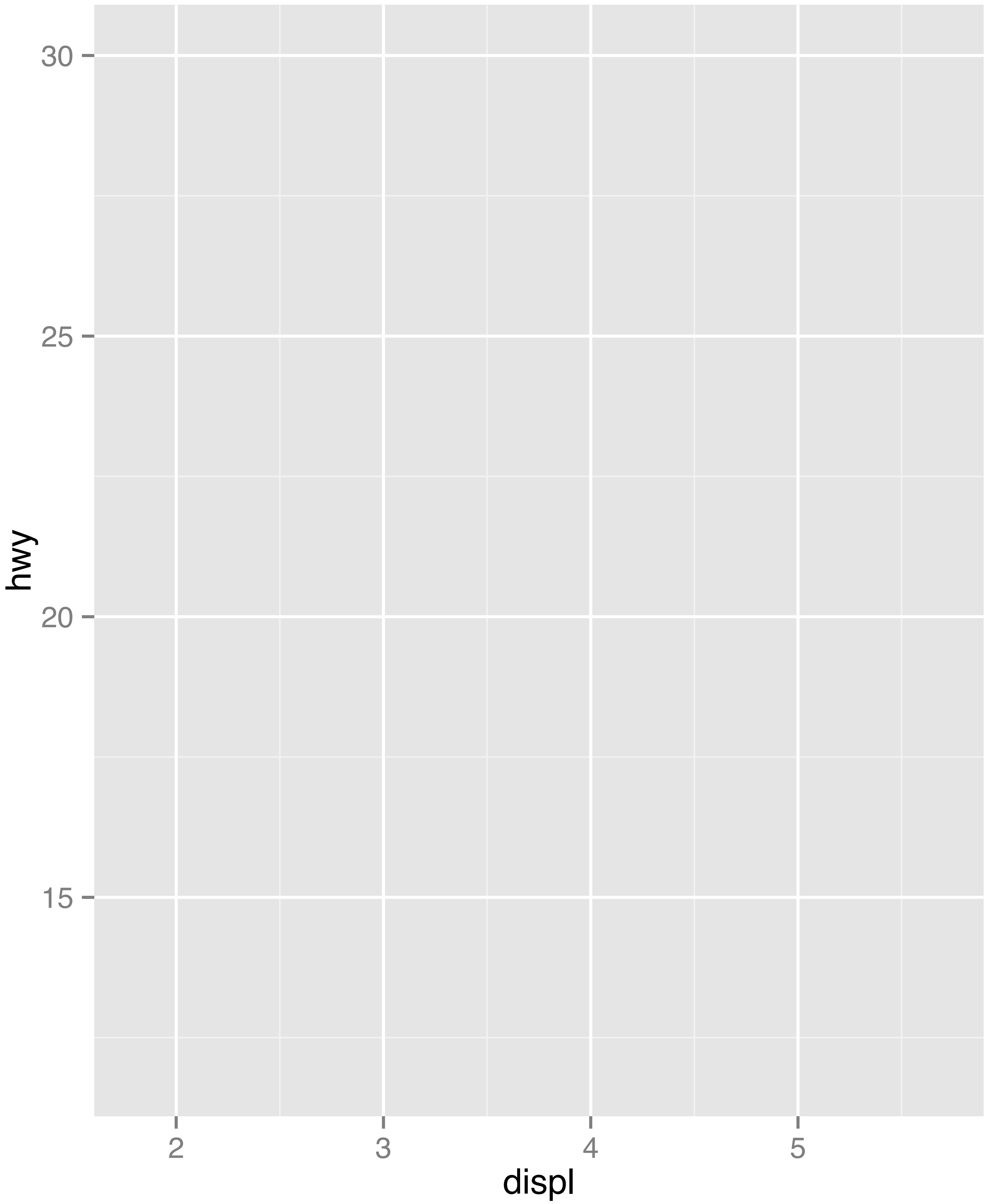
hwy	disp	cyl	class
17	5.0	8	suv
20	2.7	4	pickup
17	4.0	6	suv
25	2.8	6	compact
27	3.1	6	compact
30	2.0	4	compact
25	2.8	6	compact
23	2.8	6	compact
26	3.0	6	midsize
17	5.4	8	pickup
28	2.5	5	subcompact
29	3.5	6	midsize
26	2.4	4	midsize
29	2.0	4	midsize
15	5.4	8	pickup
29	1.8	4	compact
18	5.7	8	suv
12	4.7	8	pickup
26	2.8	6	compact
24	3.3	6	minivan

Data



Coordinate system

hwy	disp	cyl	class	
17	5.0	8	suv	●
20	2.7	4	pickup	●
17	4.0	6	suv	●
25	2.8	6	compact	●
27	3.1	6	compact	●
30	2.0	4	compact	●
25	2.8	6	compact	●
23	2.8	6	compact	●
26	3.0	6	midsize	●
17	5.4	8	pickup	●
28	2.5	5	subcompact	●
29	3.5	6	midsize	●
26	2.4	4	midsize	●
29	2.0	4	midsize	●
15	5.4	8	pickup	●
29	1.8	4	compact	●
18	5.7	8	suv	●
12	4.7	8	pickup	●
26	2.8	6	compact	●
24	3.3	6	minivan	●



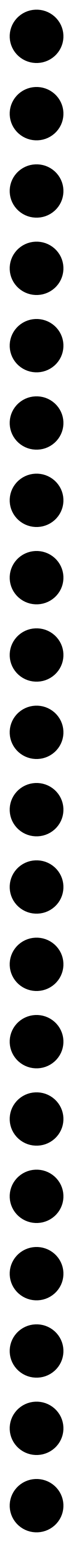
Data

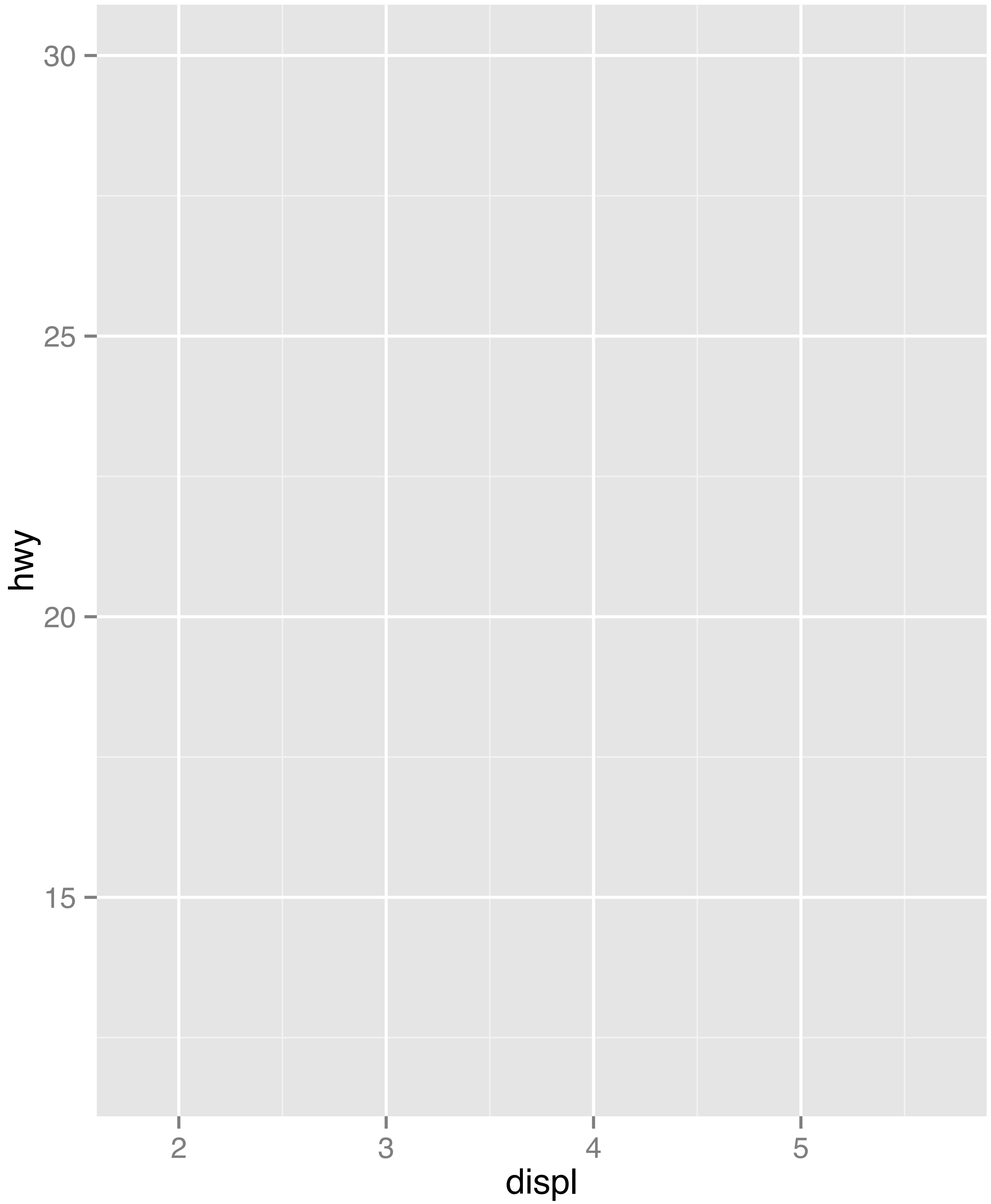
Geom

Coordinate system

Aesthetic mappings

hwy	disp	cyl	class
17	5.0	8	suv
20	2.7	4	pickup
17	4.0	6	suv
25	2.8	6	compact
27	3.1	6	compact
30	2.0	4	compact
25	2.8	6	compact
23	2.8	6	compact
26	3.0	6	midsize
17	5.4	8	pickup
28	2.5	5	subcompact
29	3.5	6	midsize
26	2.4	4	midsize
29	2.0	4	midsize
15	5.4	8	pickup
29	1.8	4	compact
18	5.7	8	suv
12	4.7	8	pickup
26	2.8	6	compact
24	3.3	6	minivan





Data

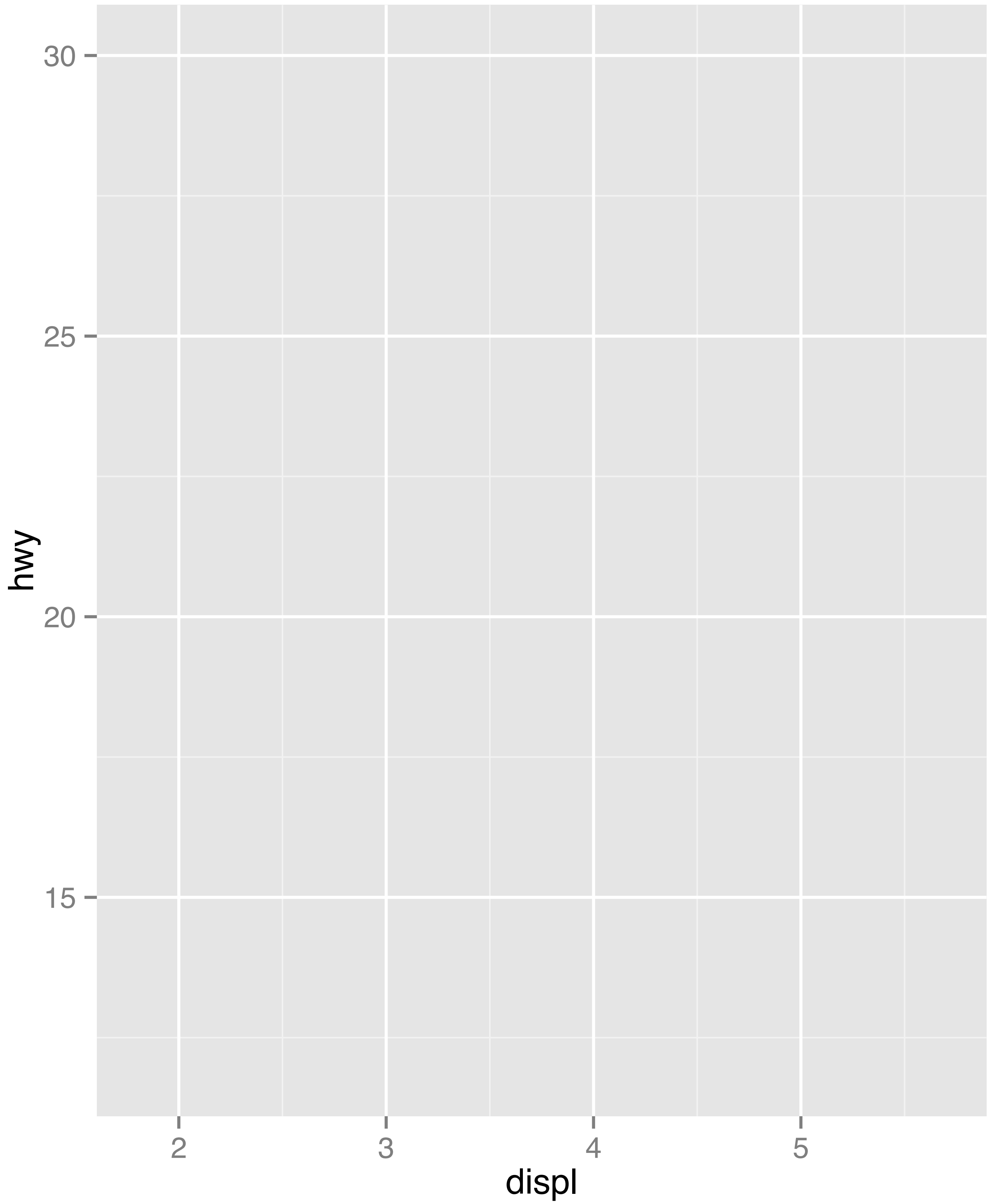
Geom

Coordinate system

Aesthetic mappings

color

hwy	disp	cyl	class	
17	5.0	8	suv	●
20	2.7	4	pickup	●
17	4.0	6	suv	●
25	2.8	6	compact	●
27	3.1	6	compact	●
30	2.0	4	compact	●
25	2.8	6	compact	●
23	2.8	6	compact	●
26	3.0	6	midsize	●
17	5.4	8	pickup	●
28	2.5	5	subcompact	●
29	3.5	6	midsize	●
26	2.4	4	midsize	●
29	2.0	4	midsize	●
15	5.4	8	pickup	●
29	1.8	4	compact	●
18	5.7	8	suv	●
12	4.7	8	pickup	●
26	2.8	6	compact	●
24	3.3	6	minivan	●



Data

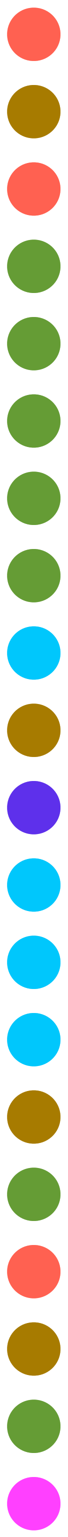
Geom

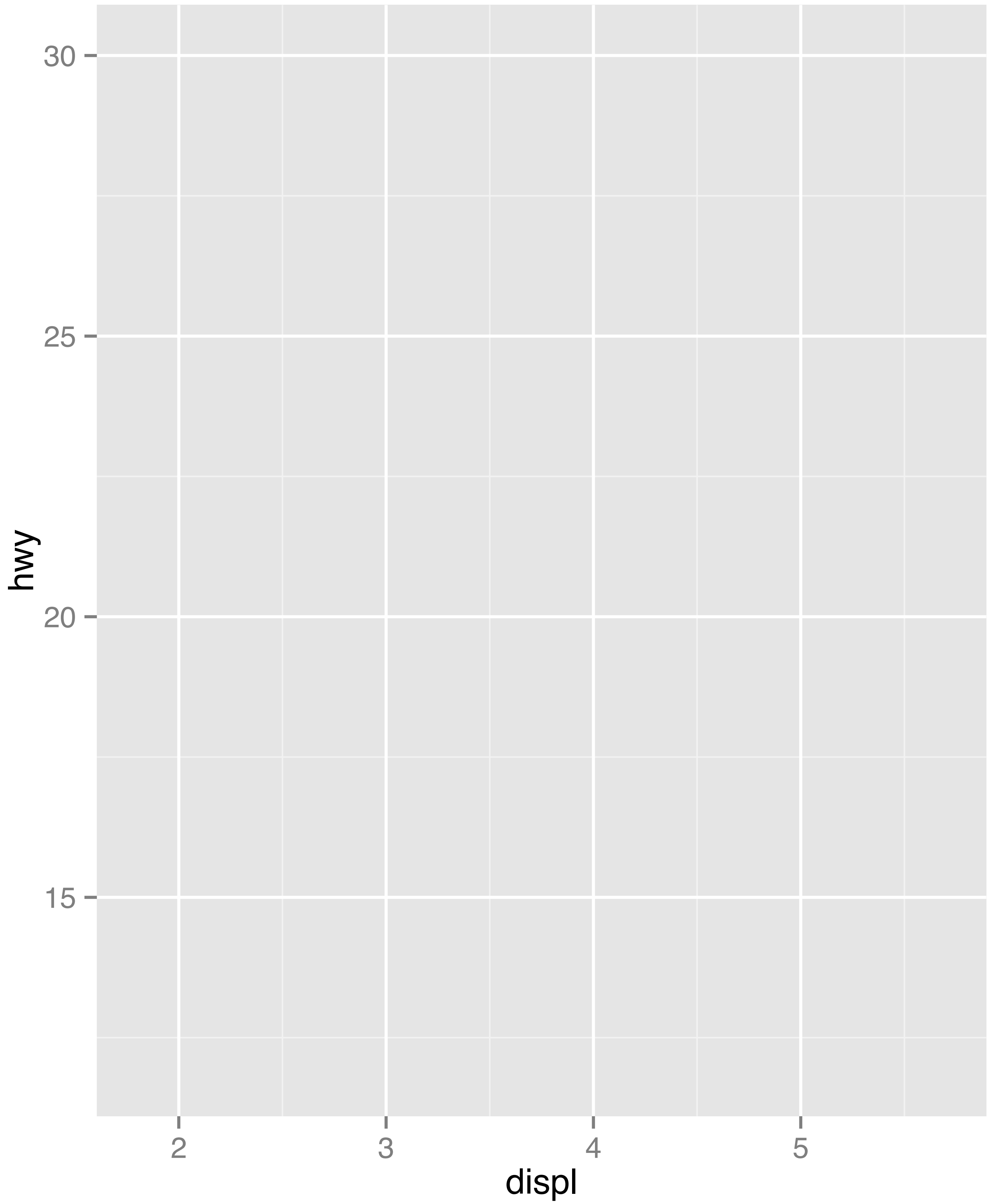
Coordinate system

Aesthetic mappings

y x color

hwy	disp	cyl	class
17	5.0	8	suv
20	2.7	4	pickup
17	4.0	6	suv
25	2.8	6	compact
27	3.1	6	compact
30	2.0	4	compact
25	2.8	6	compact
23	2.8	6	compact
26	3.0	6	midsize
17	5.4	8	pickup
28	2.5	5	subcompact
29	3.5	6	midsize
26	2.4	4	midsize
29	2.0	4	midsize
15	5.4	8	pickup
29	1.8	4	compact
18	5.7	8	suv
12	4.7	8	pickup
26	2.8	6	compact
24	3.3	6	minivan



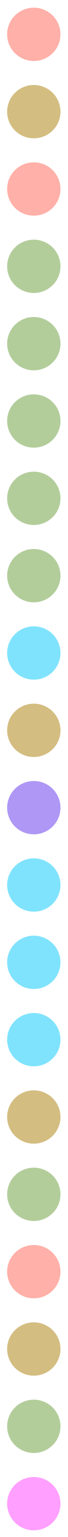


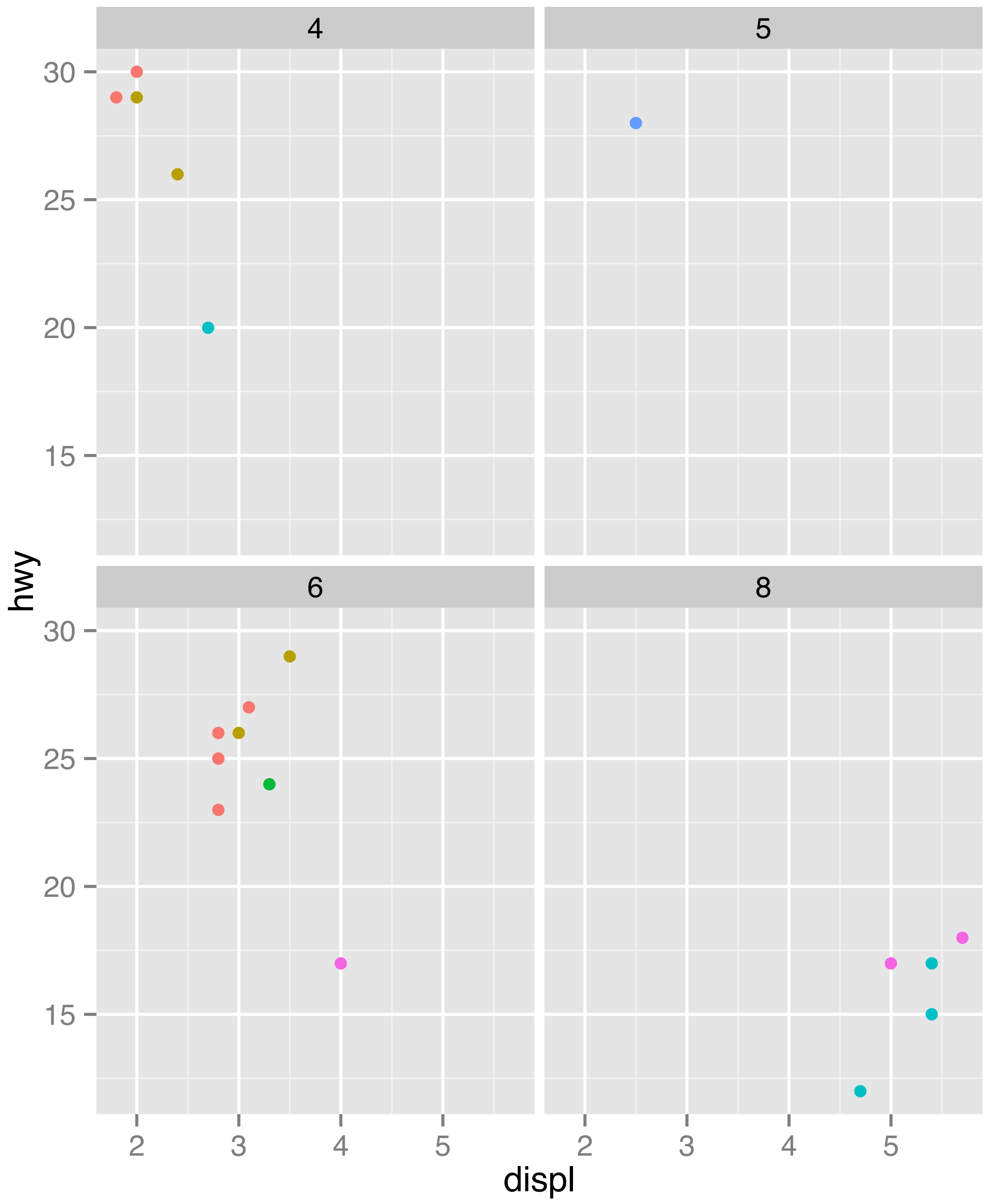
Data Geom Coordinate system

Aesthetic mappings

Facet (or not)

y	x		color
hwy	disp	cyl	class
17	5.0	8	suv
20	2.7	4	pickup
17	4.0	6	suv
25	2.8	6	compact
27	3.1	6	compact
30	2.0	4	compact
25	2.8	6	compact
23	2.8	6	compact
26	3.0	6	midsize
17	5.4	8	pickup
28	2.5	5	subcompact
29	3.5	6	midsize
26	2.4	4	midsize
29	2.0	4	midsize
15	5.4	8	pickup
29	1.8	4	compact
18	5.7	8	suv
12	4.7	8	pickup
26	2.8	6	compact
24	3.3	6	minivan





Data

Geom

Coordinate system

What is a plot?

Coordinate system

+ geom

+ data

+ aesthetic mappings

+ position adjustment

This is the grammar of graphics*

Your turn

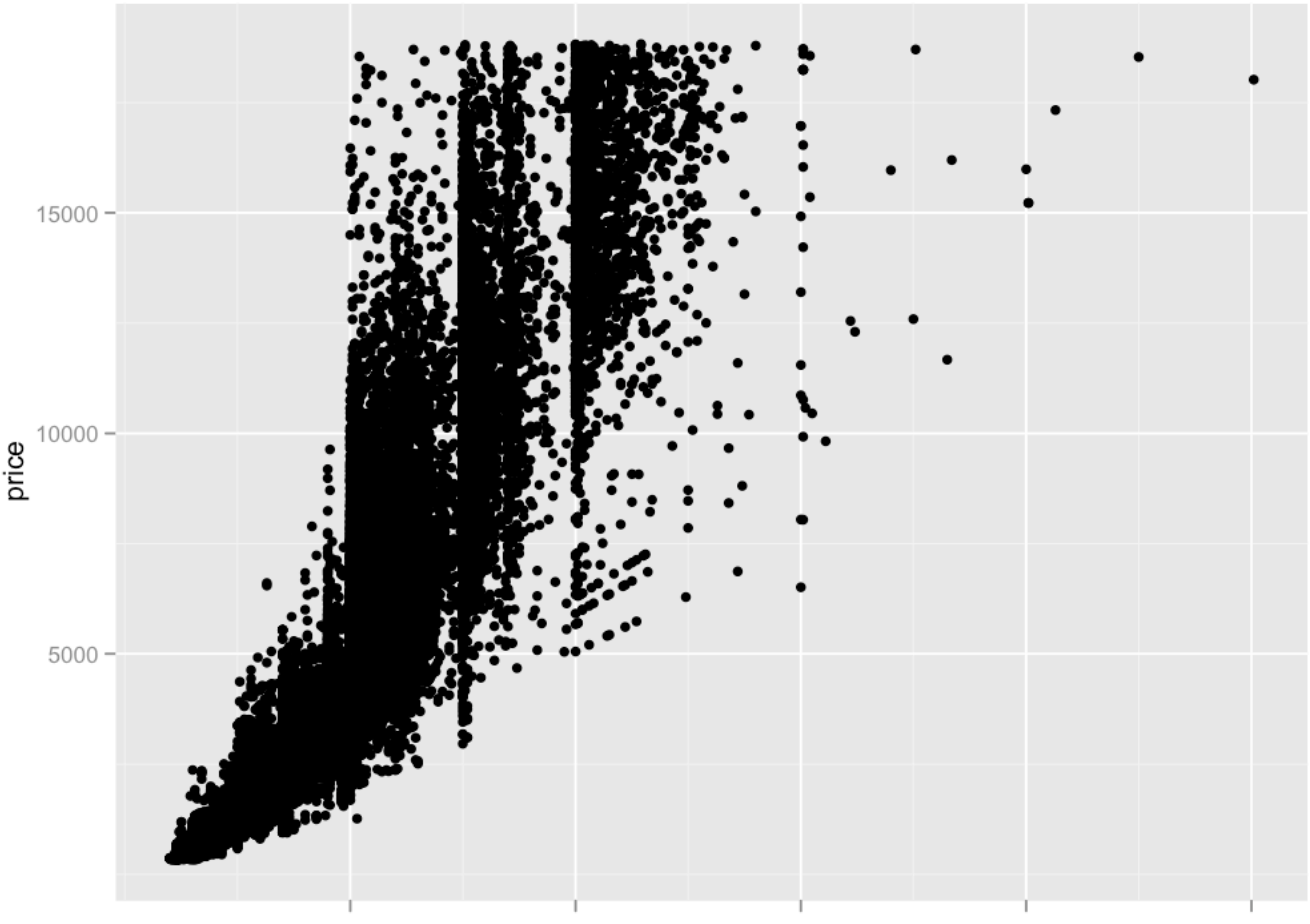
Use qplot to create each of the plots described below

Plot 1
data = economics
geom = line
aesthetic mappings: x = date, y = unemploy

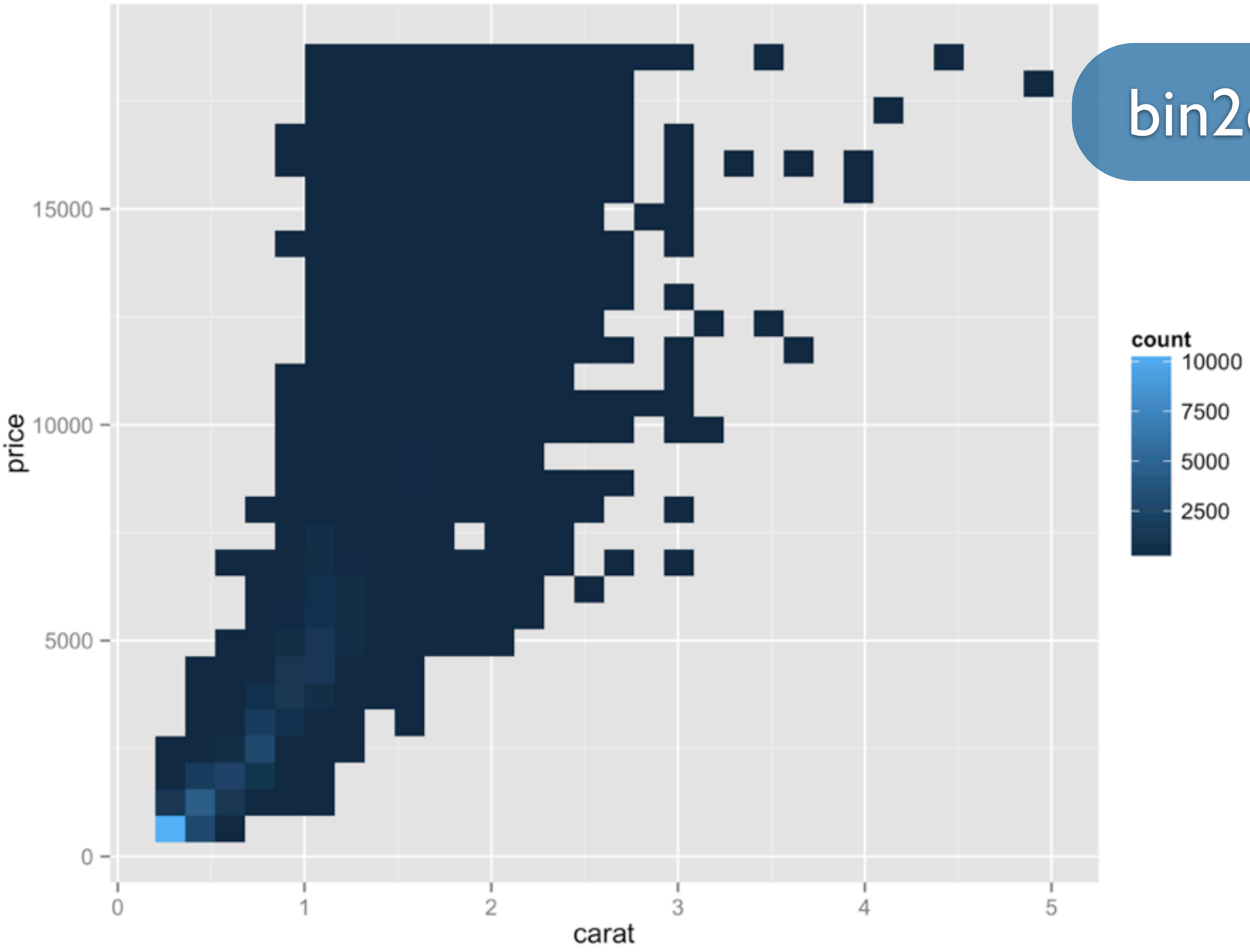
Plot 2
data = mpg
geom = point
position = jitter
aesthetic mappings: x = class, y = hwy, color = class

```
qplot(date, unemploy, data = economics,
  geom = “line”)

qplot(class, hwy, data = mpg,
  position = “jitter”, color = class)
```

Geoms for Big Data

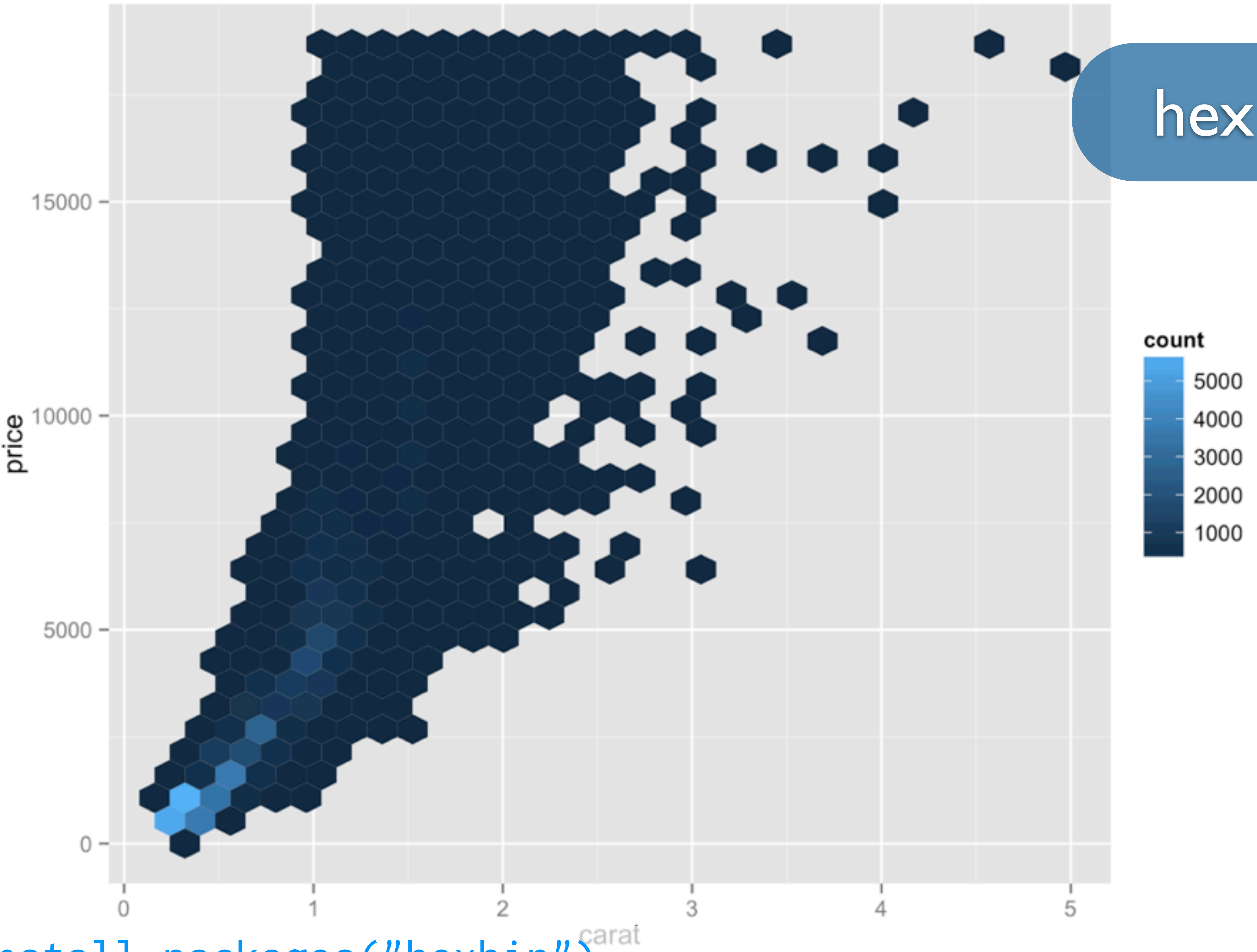


`qplot(carat, price, data = diamonds)`



bin2d

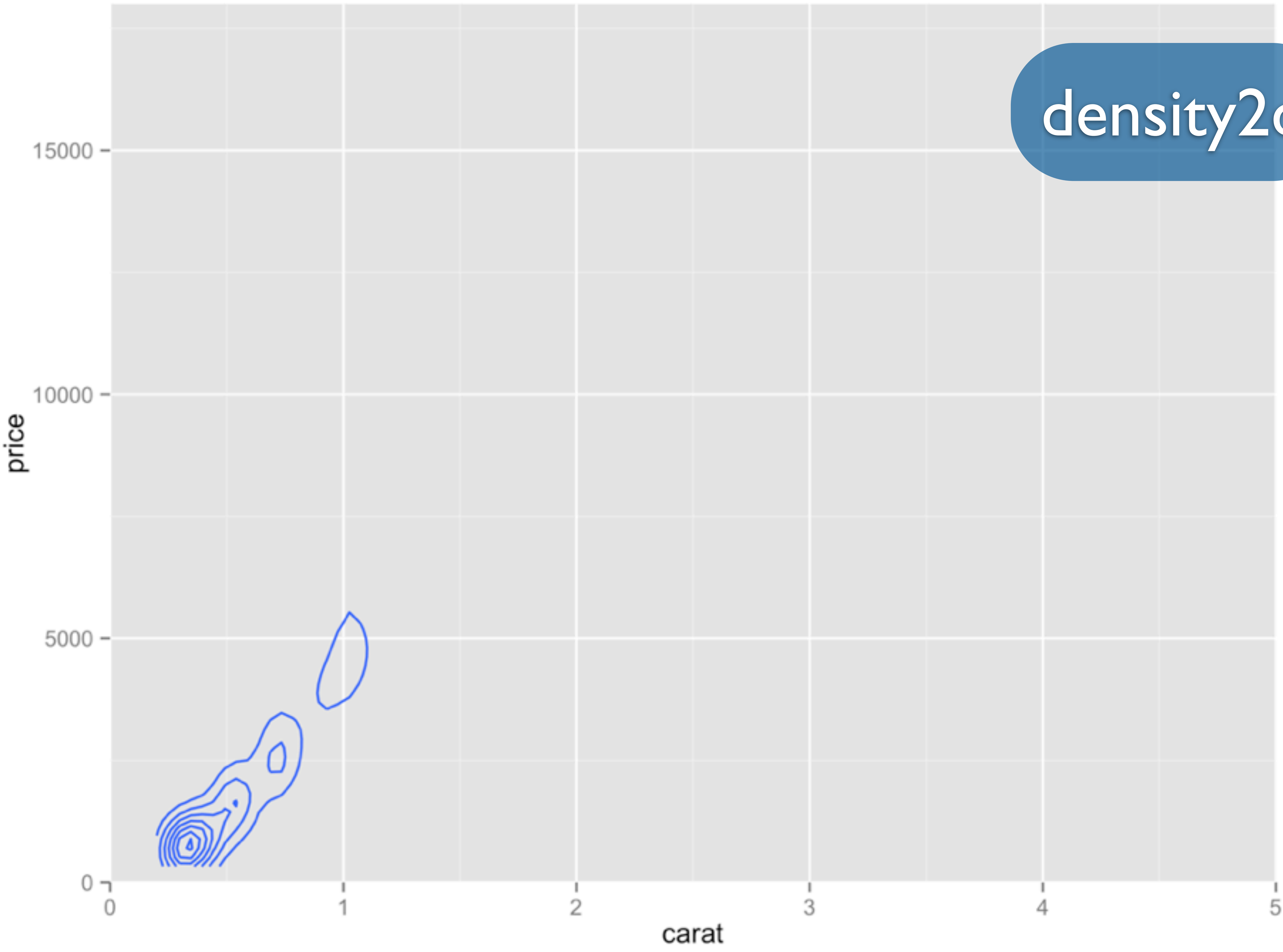
```
qplot(carat, price, data = diamonds, geom = "bin2d")
```



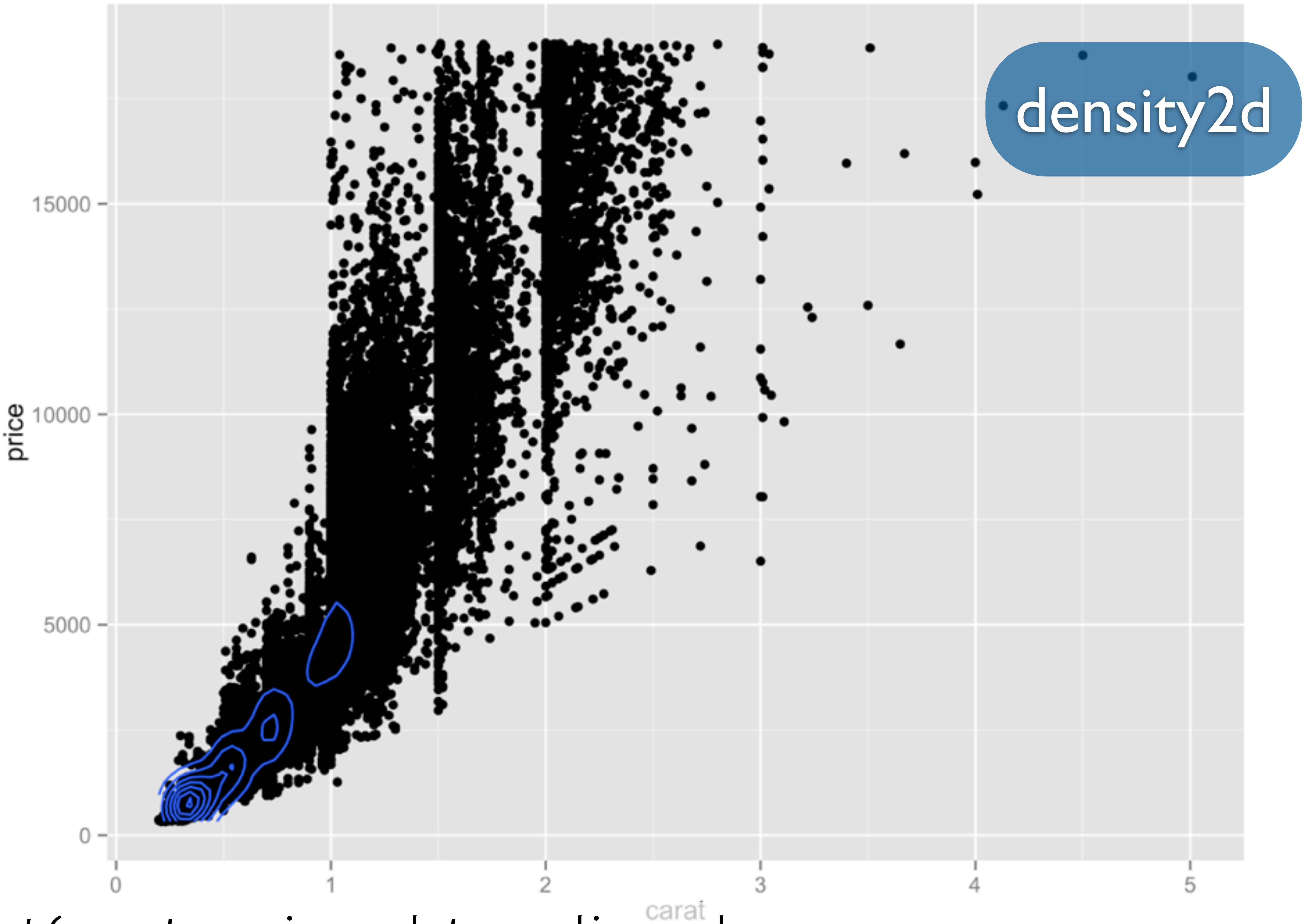
hex

```
# install.packages("hexbin")
qplot(carat, price, data = diamonds, geom = "hex")
```

density2d

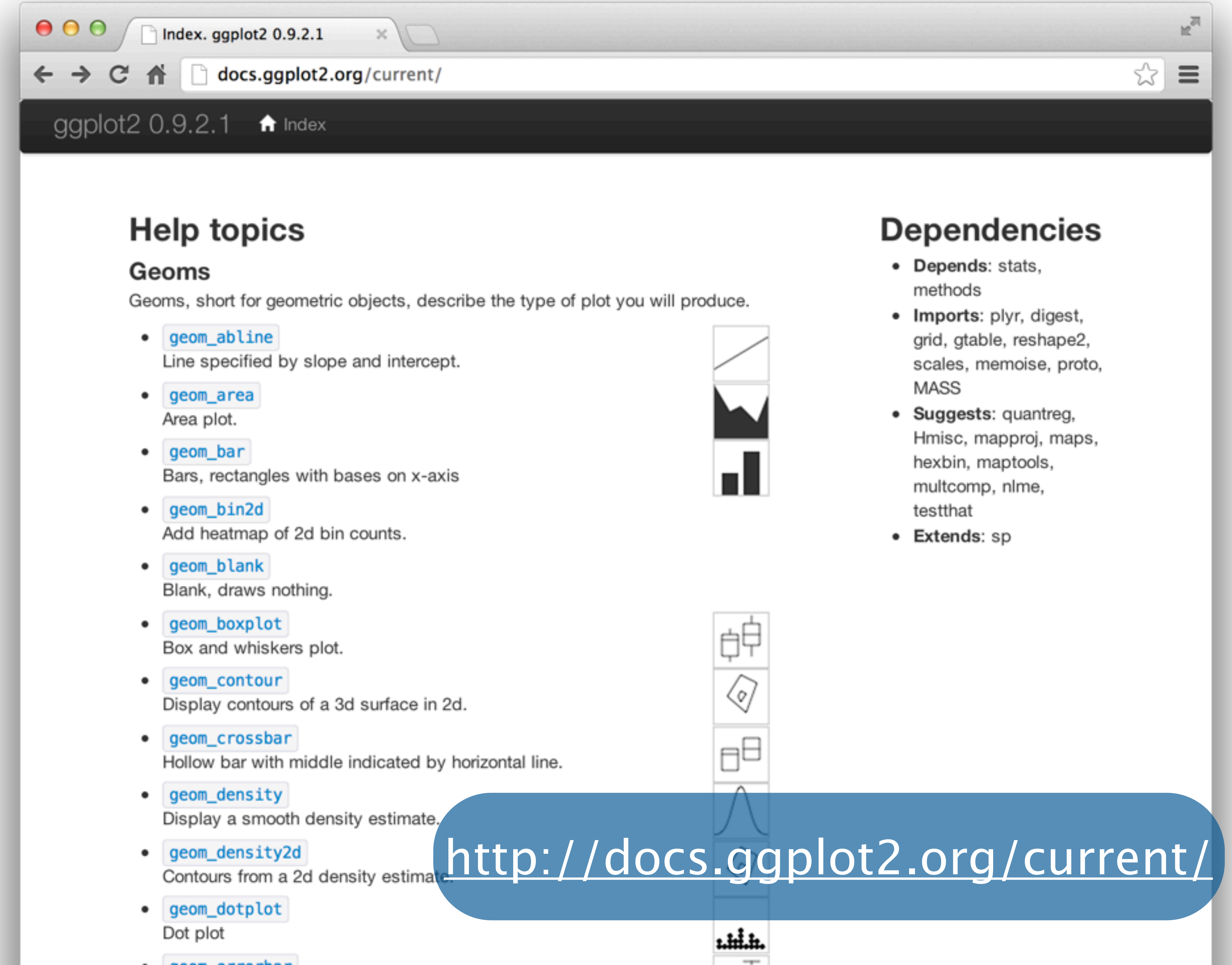


```
qplot(carat, price, data = diamonds, geom = "density2d")
```



density2d

```
qplot(carat, price, data = diamonds,
  geom = c("point", "density2d"))
```



Help topics

Geoms

Geoms, short for geometric objects, describe the type of plot you will produce.

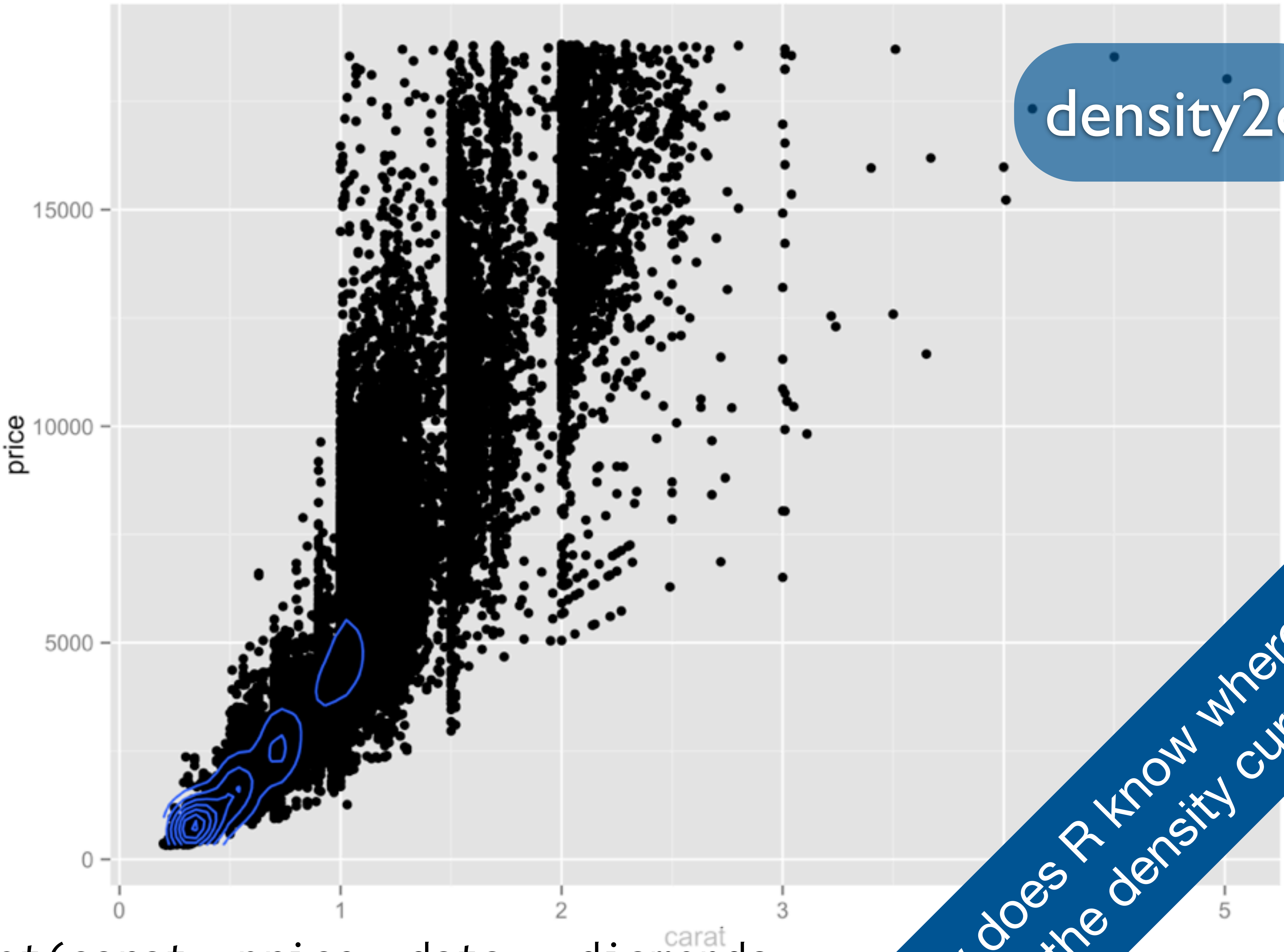
- `geom_abline`
  Line specified by slope and intercept.
- `geom_area`
  Area plot.
- `geom_bar`
  Bars, rectangles with bases on x-axis
- `geom_bin2d`
  Add heatmap of 2d bin counts.
- `geom_blank`
  Blank, draws nothing.
- `geom_boxplot`
  Box and whiskers plot.
- `geom_contour`
  Display contours of a 3d surface in 2d.
- `geom_crossbar`
  Hollow bar with middle indicated by horizontal line.
- `geom_density`
  Display a smooth density estimate.
- `geom_density2d`
  Contours from a 2d density estimate.
- `geom_dotplot`
  Dot plot

Dependencies

- **Depends**: stats, methods
- **Imports**: plyr, digest, grid, gtable, reshape2, scales, memoise, proto, MASS
- **Suggests**: quantreg, Hmisc, mapproj, maps, hexbin, maptools, multcomp, nlme, testthat
- **Extends**: sp

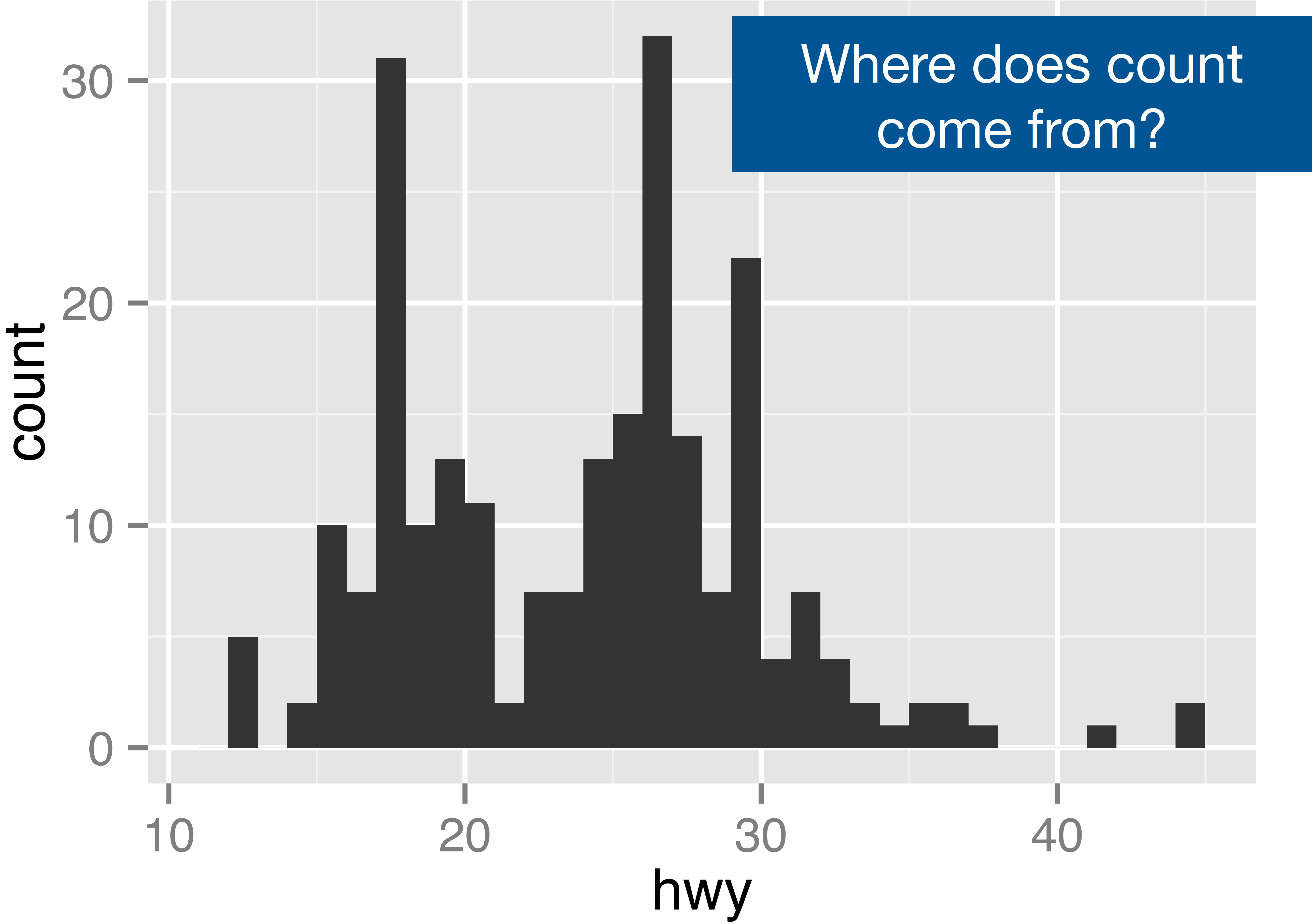
http://docs.ggplot2.org/current/

density2d



```
qplot(carat, price, data = diamonds,
  geom = c("point", "density2d"))
```

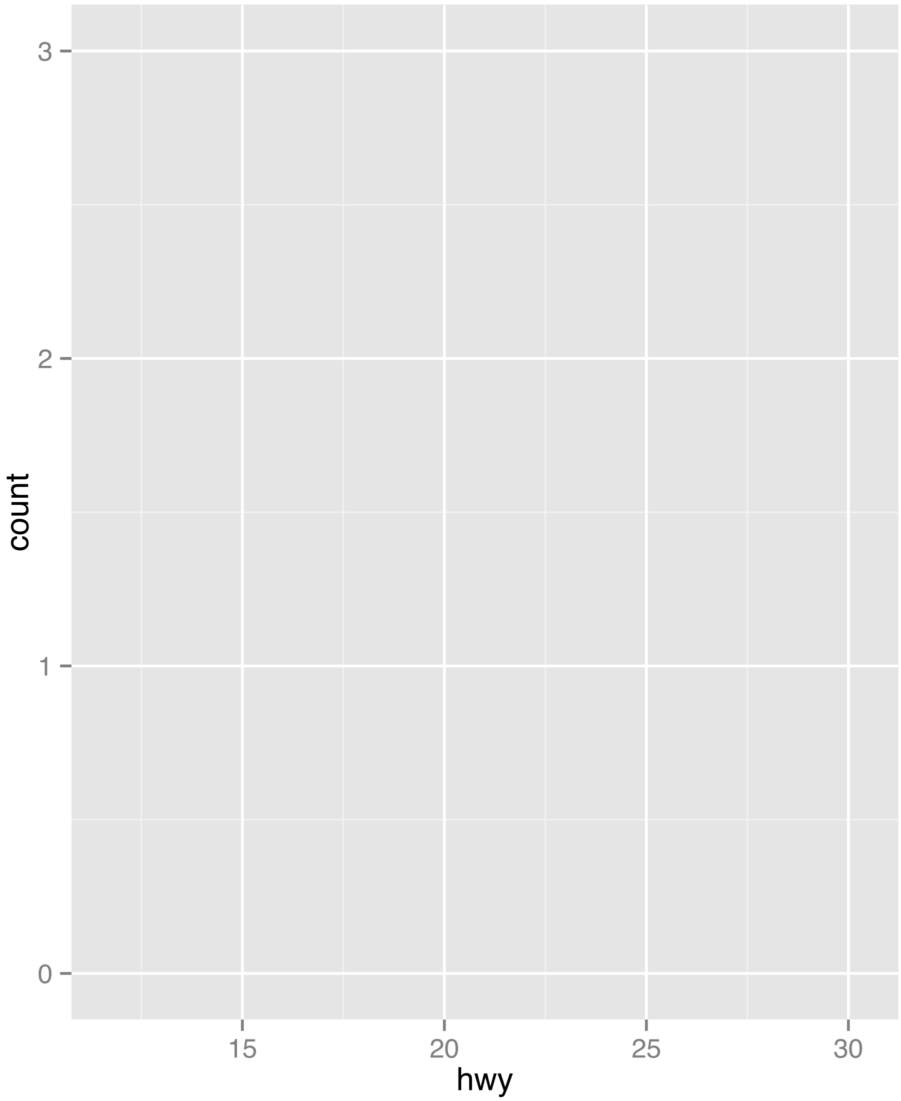
How does R know where to draw the density curves?



Where does count come from?

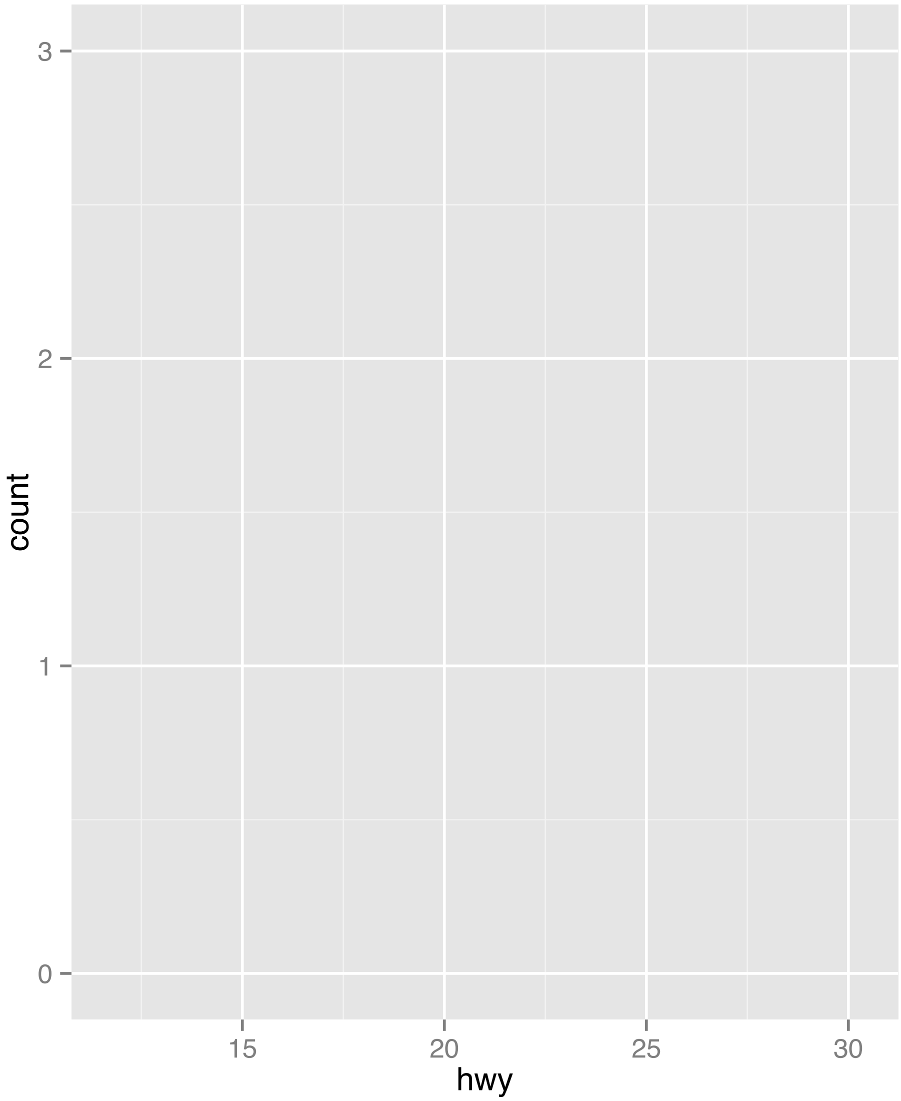
```
qplot(hwy, data = mpg, geom = "histogram", binwidth = 1)
```

How to build a plot 2
Stats



Coordinate system

hwy	displ	cyl	class
12	4.7	8	pickup
15	5.4	8	pickup
17	5.0	8	suv
17	4.0	6	suv
17	5.4	8	pickup
18	5.7	8	suv
20	2.7	4	pickup
23	2.8	6	compact
24	3.3	6	minivan
25	2.8	6	compact
25	2.8	6	compact
26	3.0	6	midsize
26	2.4	4	midsize
26	2.8	6	compact
27	3.1	6	compact
28	2.5	5	subcomp
29	3.5	6	midsize
29	2.0	4	midsize
29	1.8	4	compact
30	2.0	4	compact

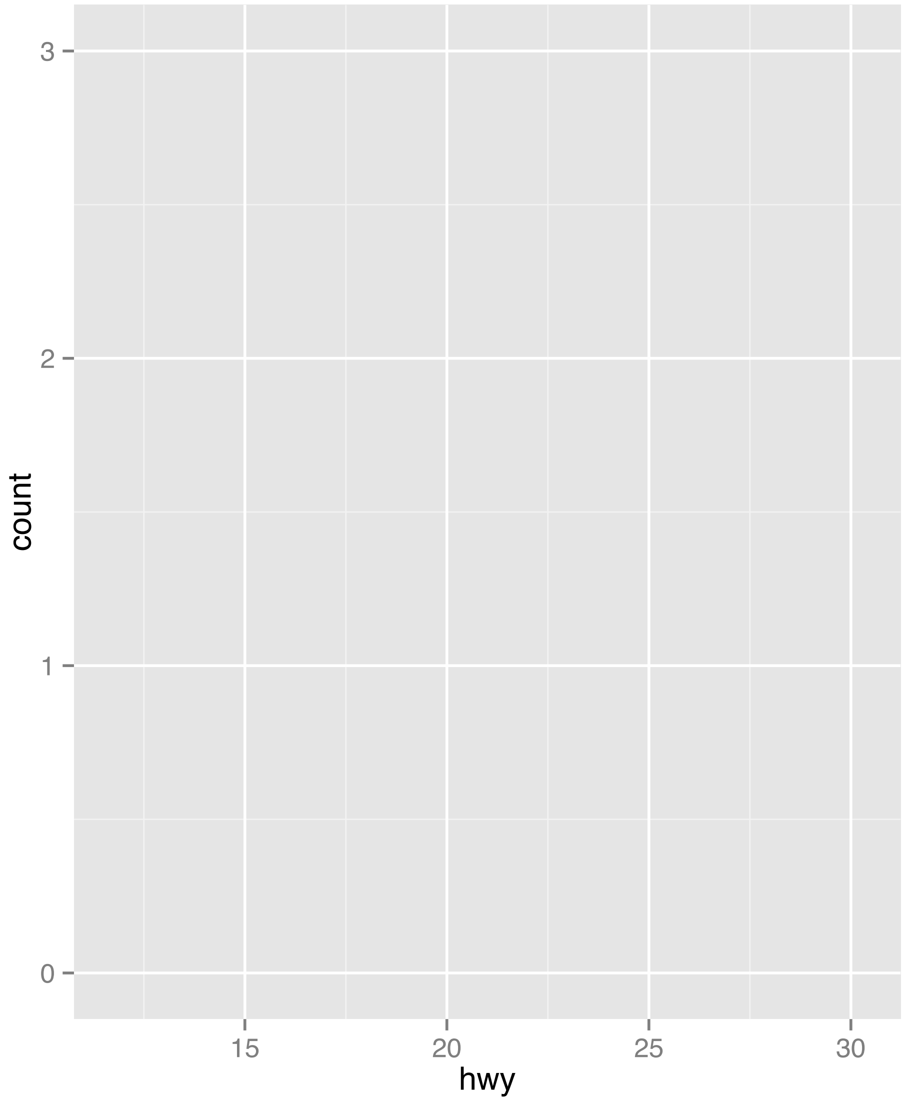


Data

Coordinate system

hwy	displ	cyl	class
12	4.7	8	pickup
15	5.4	8	pickup
17	5.0	8	suv
17	4.0	6	suv
17	5.4	8	pickup
18	5.7	8	suv
20	2.7	4	pickup
23	2.8	6	compact
24	3.3	6	minivan
25	2.8	6	compact
25	2.8	6	compact
26	3.0	6	midsize
26	2.4	4	midsize
26	2.8	6	compact
27	3.1	6	compact
28	2.5	5	subcomp
29	3.5	6	midsize
29	2.0	4	midsize
29	1.8	4	compact
30	2.0	4	compact

bin	hwy	count
1	12	1
2	15	1
3	17	3
4	18	1
5	20	1
6	23	1
7	24	1
8	25	2
9	26	3
10	27	1
11	28	1
12	29	3
13	30	1



Data

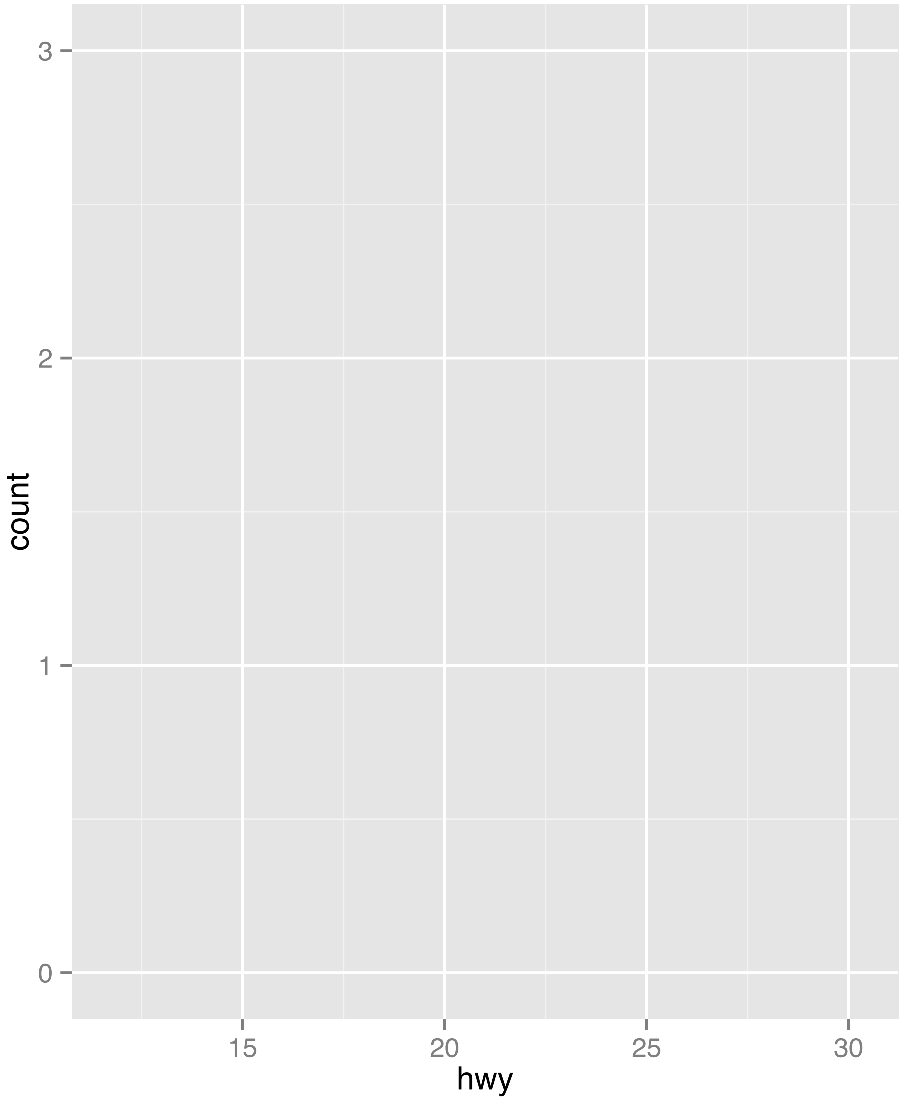
Stat

Coordinate system

hwy	displ	cyl	class
12	4.7	8	pickup
15	5.4	8	pickup
17	5.0	8	suv
17	4.0	6	suv
17	5.4	8	pickup
18	5.7	8	suv
20	2.7	4	pickup
23	2.8	6	compact
24	3.3	6	minivan
25	2.8	6	compact
25	2.8	6	compact
26	3.0	6	midsize
26	2.4	4	midsize
26	2.8	6	compact
27	3.1	6	compact
28	2.5	5	subcomp
29	3.5	6	midsize
29	2.0	4	midsize
29	1.8	4	compact
30	2.0	4	compact

→

bin	hwy	count
1	12	1
2	15	1
3	17	3
4	18	1
5	20	1
6	23	1
7	24	1
8	25	2
9	26	3
10	27	1
11	28	1
12	29	3
13	30	1



Data Stat **Geom** Coordinate system

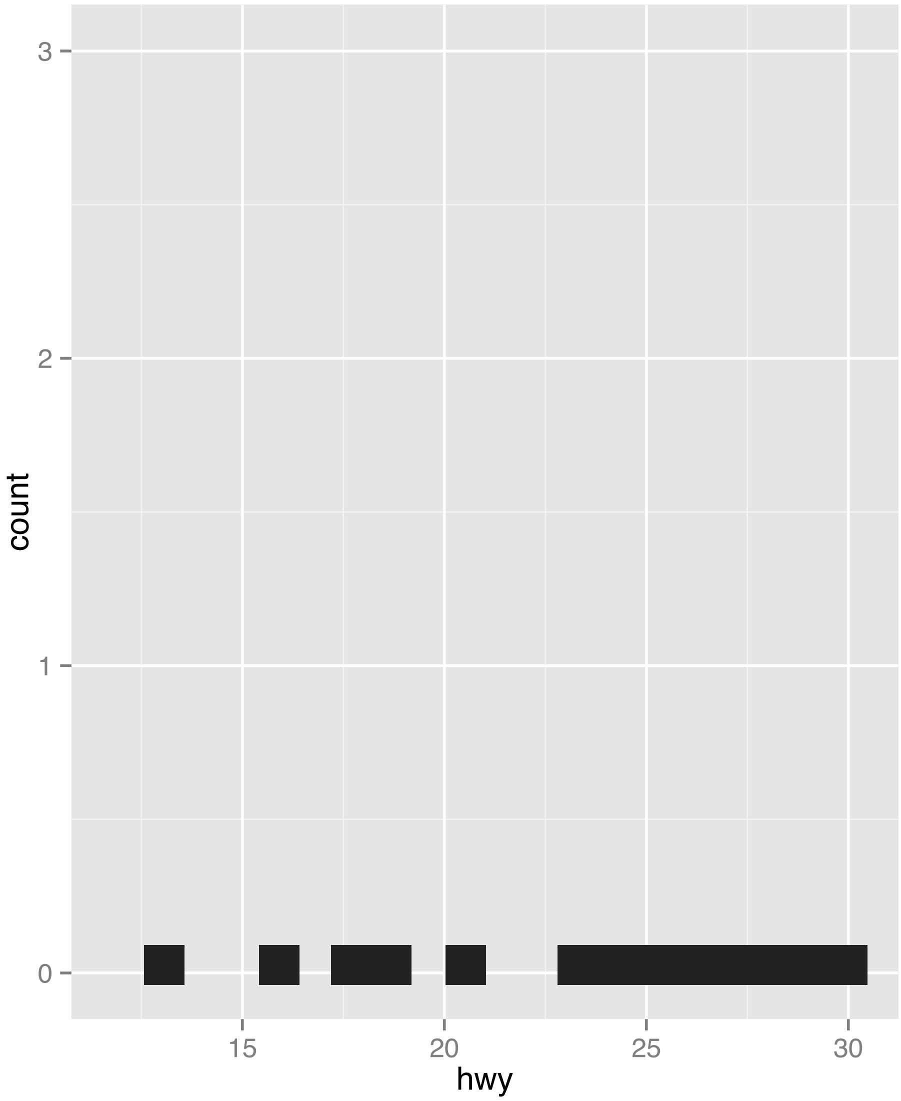
Aesthetic mappings

hwy	displ	cyl	class
12	4.7	8	pickup
15	5.4	8	pickup
17	5.0	8	suv
17	4.0	6	suv
17	5.4	8	pickup
18	5.7	8	suv
20	2.7	4	pickup
23	2.8	6	compact
24	3.3	6	minivan
25	2.8	6	compact
25	2.8	6	compact
26	3.0	6	midsize
26	2.4	4	midsize
26	2.8	6	compact
27	3.1	6	compact
28	2.5	5	subcomp
29	3.5	6	midsize
29	2.0	4	midsize
29	1.8	4	compact
30	2.0	4	compact

→

x

bin	hwy	count
1	12	1
2	15	1
3	17	3
4	18	1
5	20	1
6	23	1
7	24	1
8	25	2
9	26	3
10	27	1
11	28	1
12	29	3
13	30	1



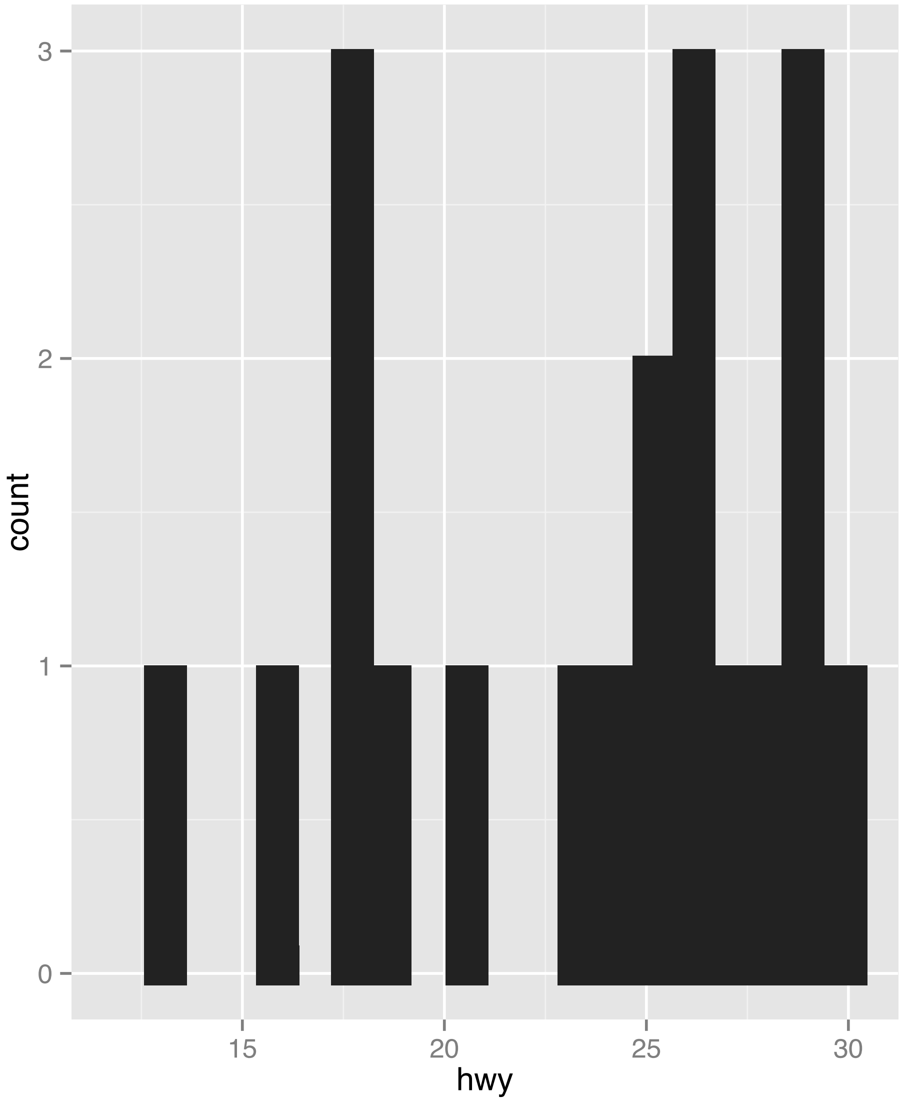
Data Stat Geom Coordinate system

Aesthetic mappings

hwy	displ	cyl	class
12	4.7	8	pickup
15	5.4	8	pickup
17	5.0	8	suv
17	4.0	6	suv
17	5.4	8	pickup
18	5.7	8	suv
20	2.7	4	pickup
23	2.8	6	compact
24	3.3	6	minivan
25	2.8	6	compact
25	2.8	6	compact
26	3.0	6	midsize
26	2.4	4	midsize
26	2.8	6	compact
27	3.1	6	compact
28	2.5	5	subcomp
29	3.5	6	midsize
29	2.0	4	midsize
29	1.8	4	compact
30	2.0	4	compact

x ymax

bin	hwy	count	
1	12	1	■
2	15	1	■
3	17	3	■
4	18	1	■
5	20	1	■
6	23	1	■
7	24	1	■
8	25	2	■
9	26	3	■
10	27	1	■
11	28	1	■
12	29	3	■
13	30	1	■



Data Stat Geom Coordinate system

Stat

A transformation done to the data before plotting it

What is a plot?

Coordinate system

+ geom

+ data

+ aesthetic mappings

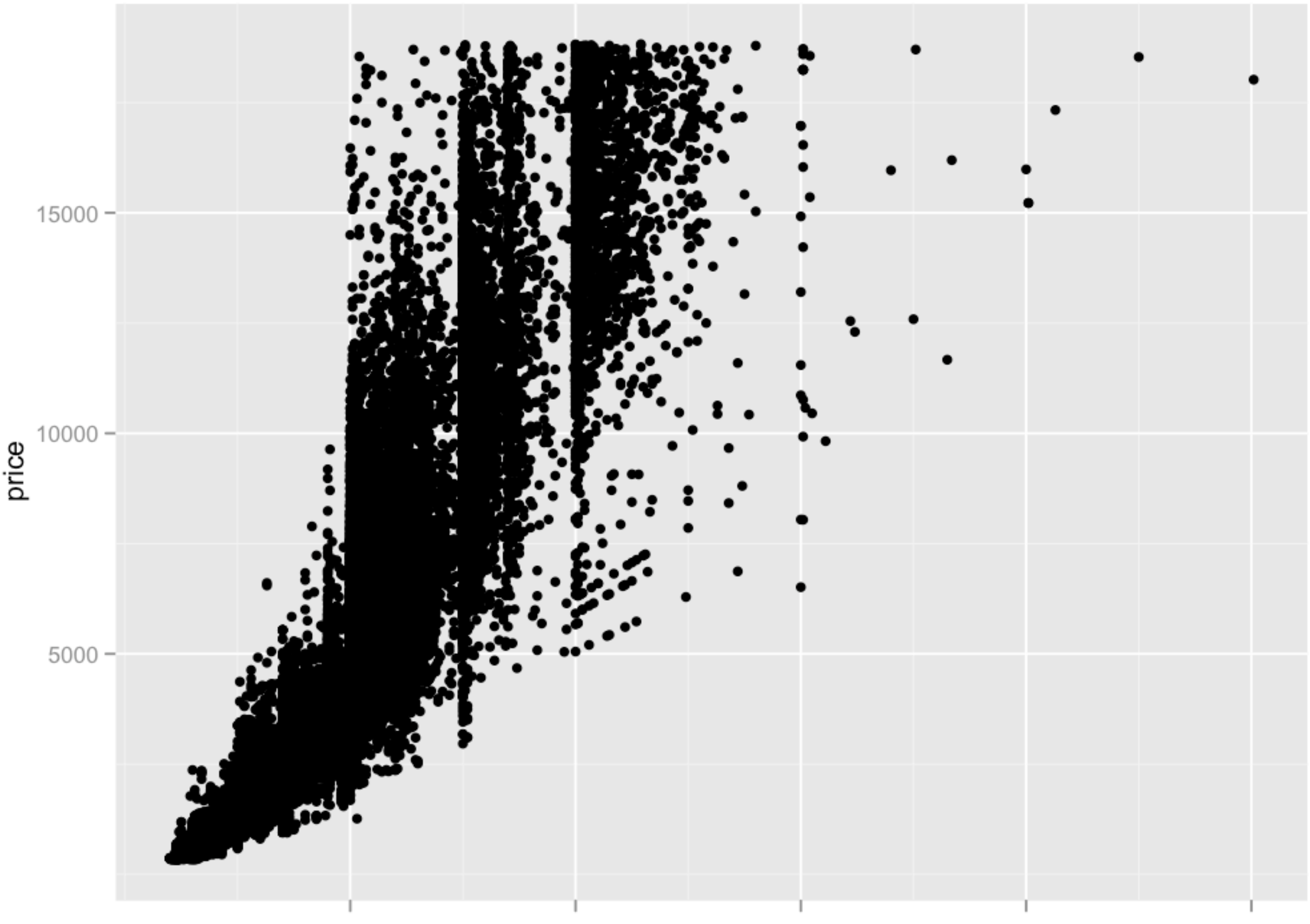
+ position adjustment

+ stat

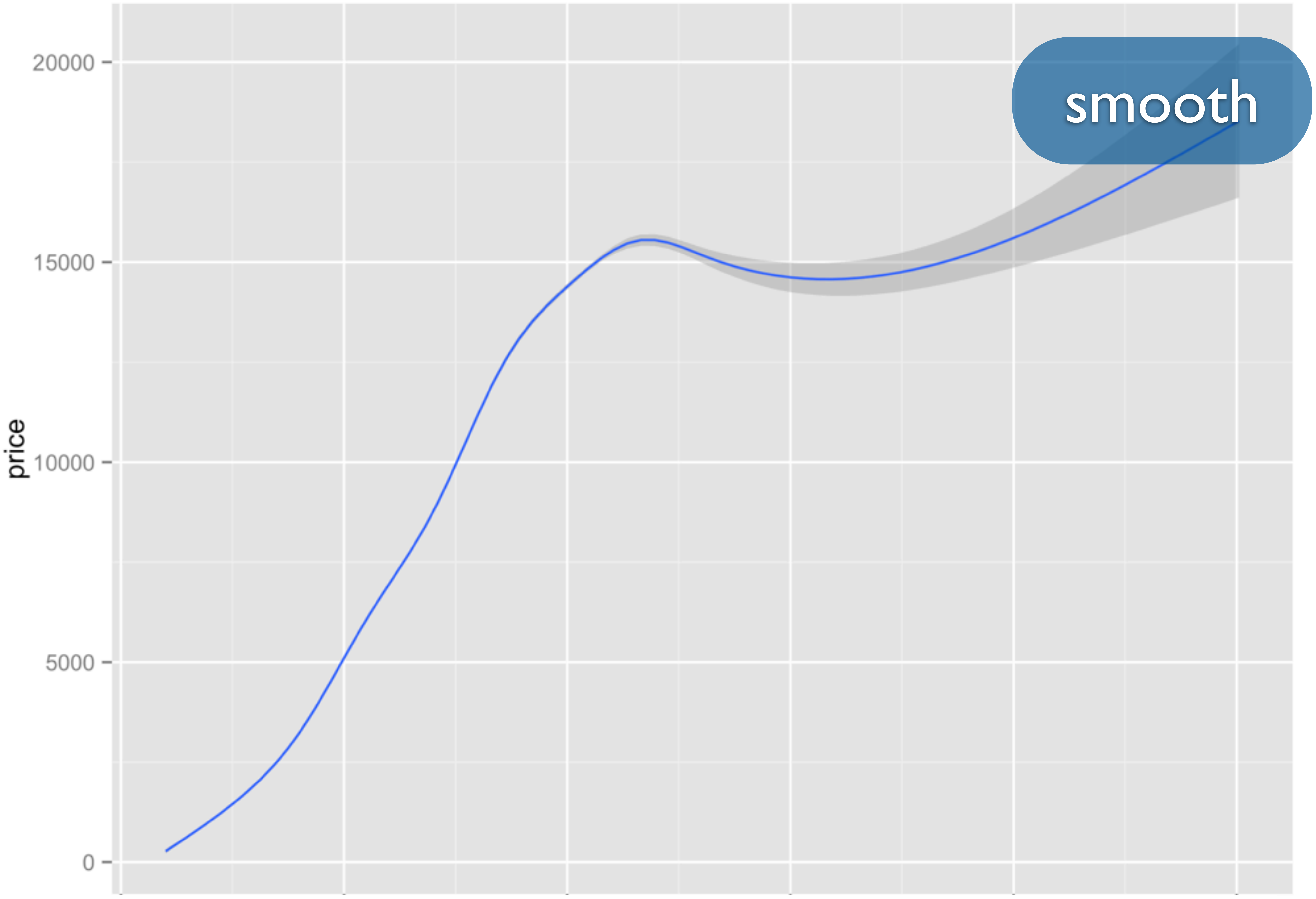
This is the grammar of graphics

geom	base geom	stat
histogram	bar	bin
smooth	line	smooth
boxplot	boxplot	boxplot
density	line	densit
freqpoly	line	

Geoms in ggplot2 know when
and how to use a stat

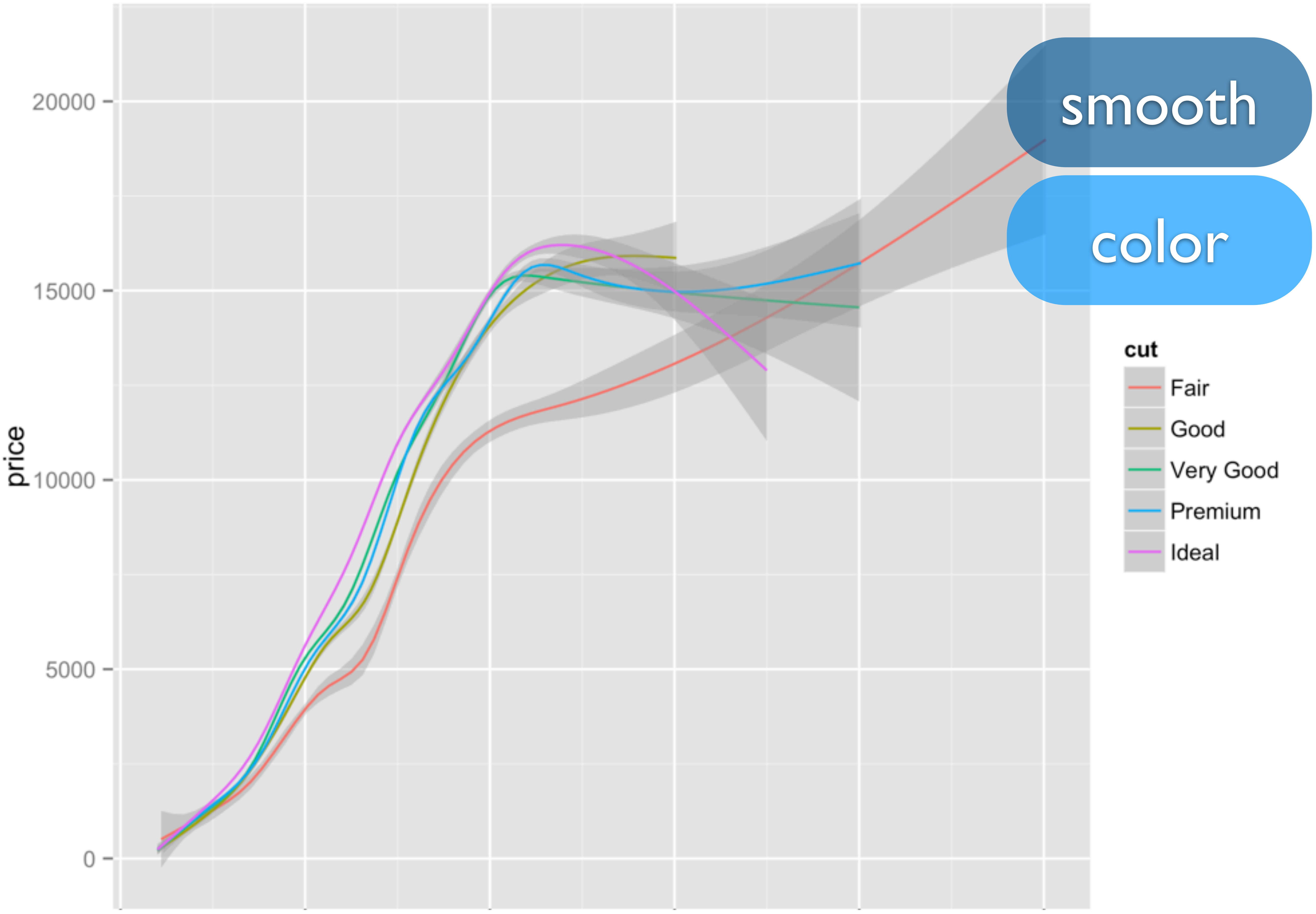


```
qplot(carat, price, data = diamonds)
```

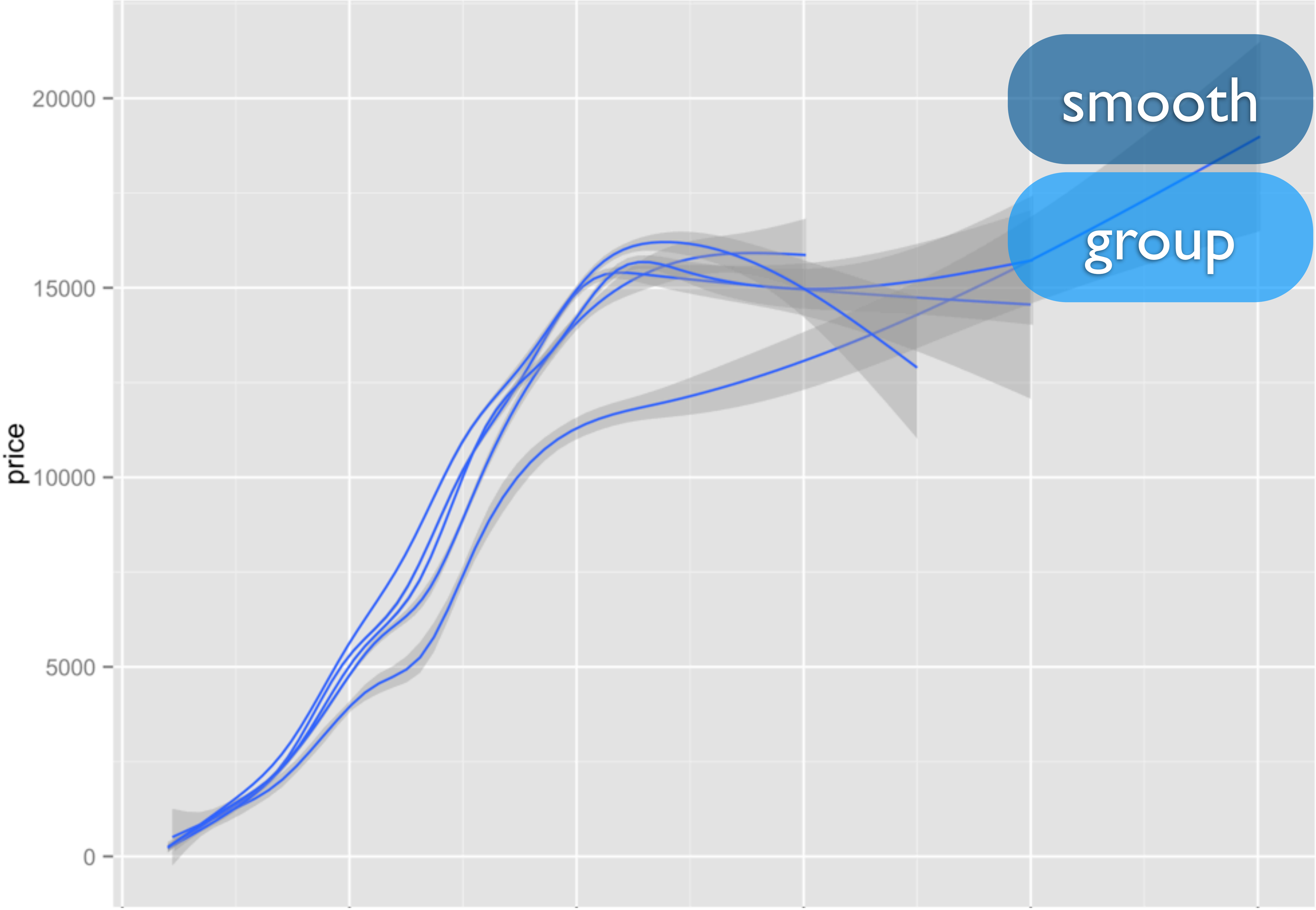


smooth

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



`qplot(carat, price, data = diamonds, geom = "smooth", color = cut)`



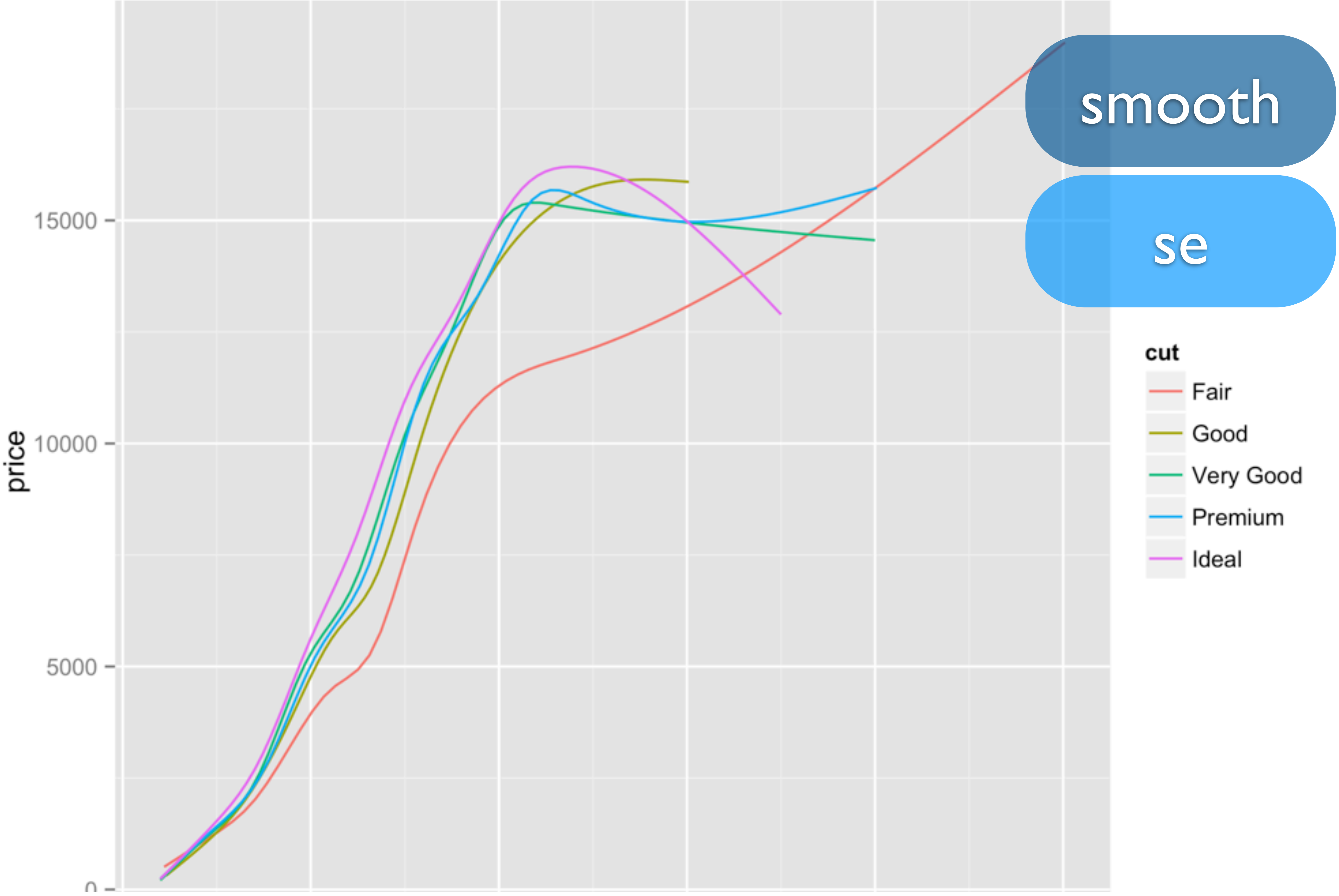
smooth

group

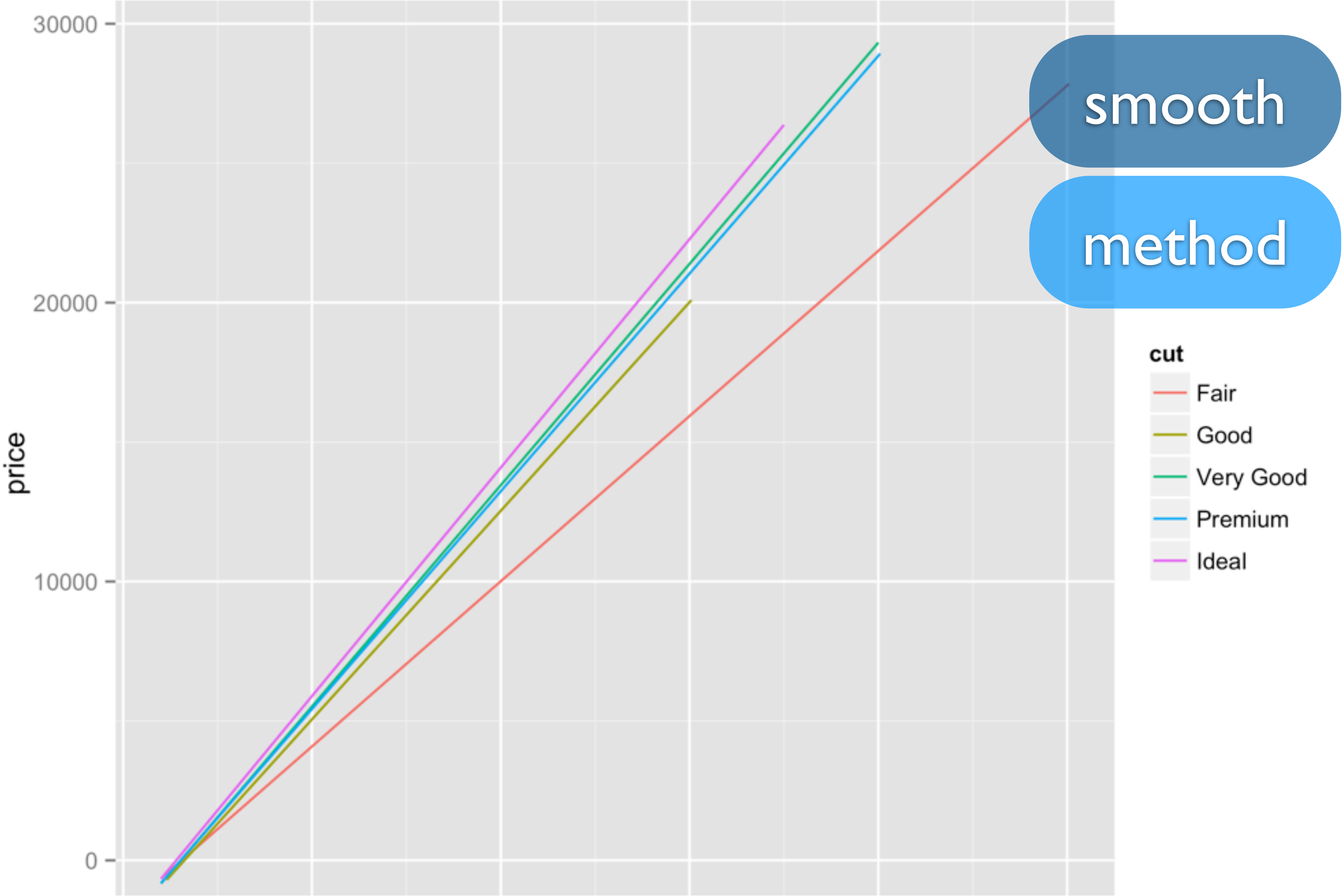
```
qplot(carat, price, data = diamonds, geom = "smooth", group = cut)
```

smooth

se



```
qplot(carat, price, data = diamonds, geom = "smooth",
  color = cut, se = FALSE)
```



smooth

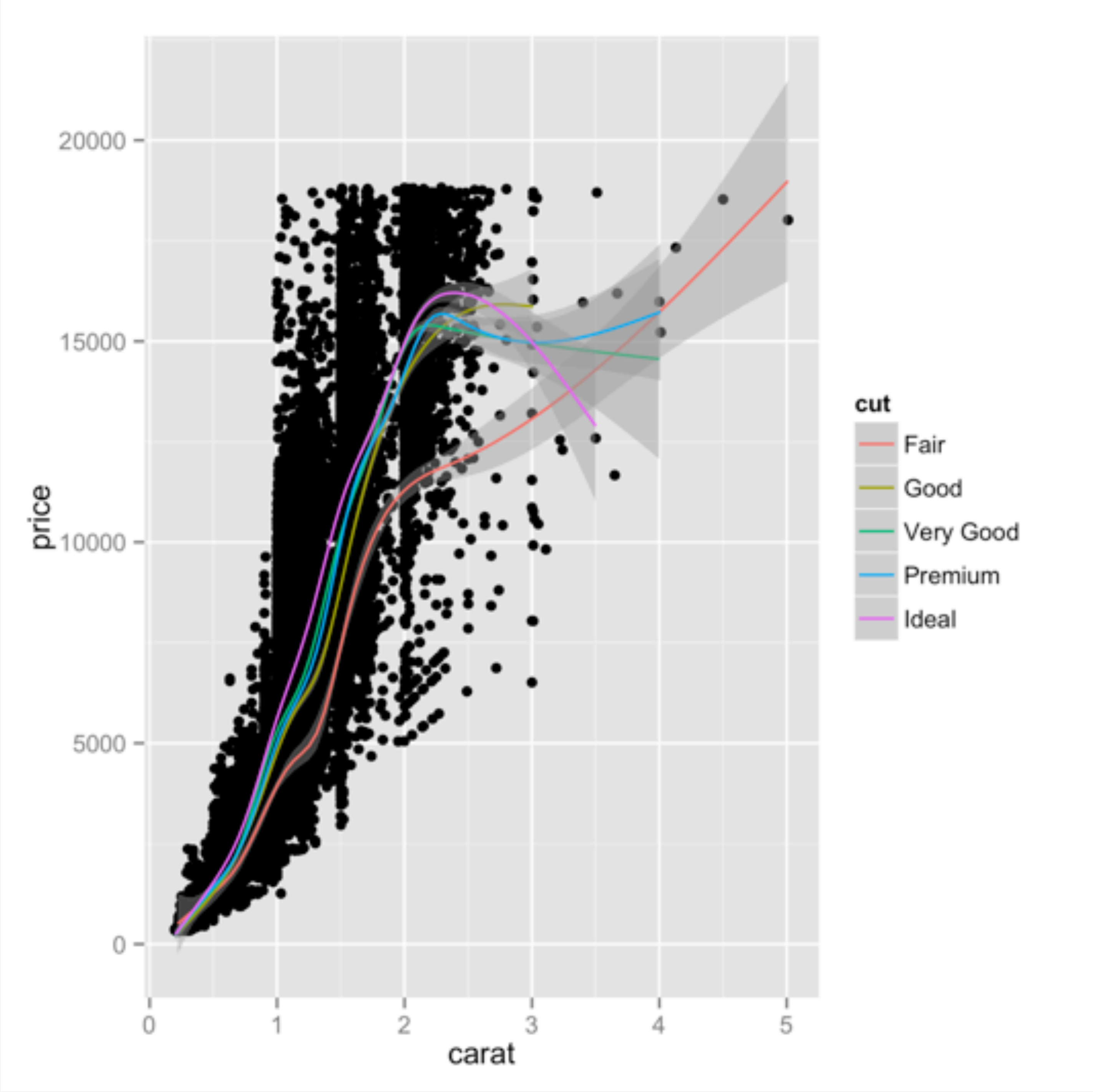
method

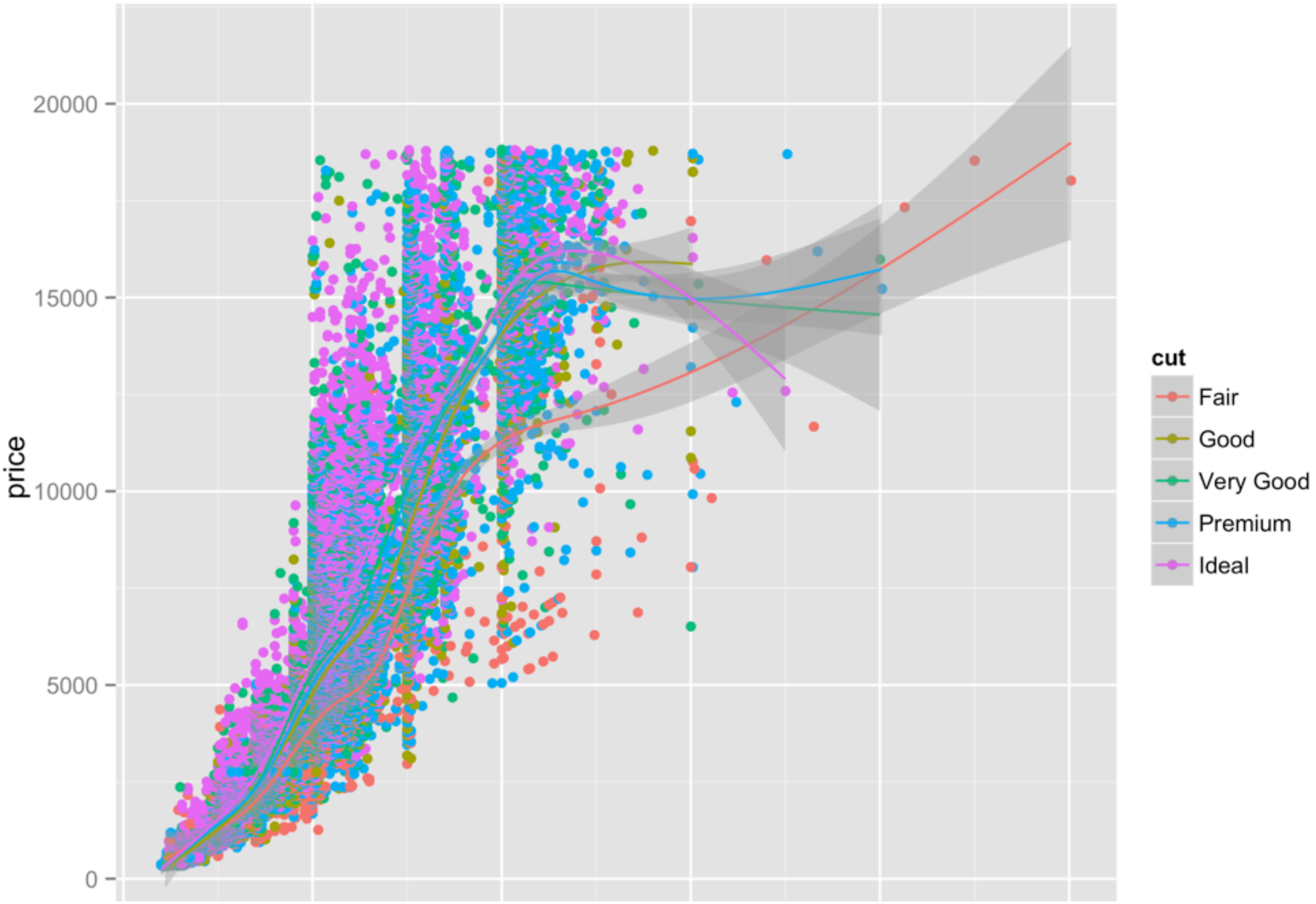
```
qplot(carat, price, data = diamonds, geom = "smooth",
  color = cut, se = FALSE, method = lm)
```

Your turn

It's useful to overlay a summary on top of the raw data.

Can you make this plot?
If not, why not?





```
qplot(carat, price, data = diamonds, color = cut,
  geom = c("point", "smooth"))
```

`qplot` is good for making quick plots, but it fails if you want to assign different aesthetics to different geoms.

The solution? layers

Layers

Geom functions

There are two ways to add additional geoms to a plot

1) A vector of geom names:

```
qplot(carat, price, data = diamonds,
  geom = c(”point”, ”smooth”))
```

2) The geom functions

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

Always begins with geom_

geom's name

open and closed parentheses

```
geom_smooth()
```

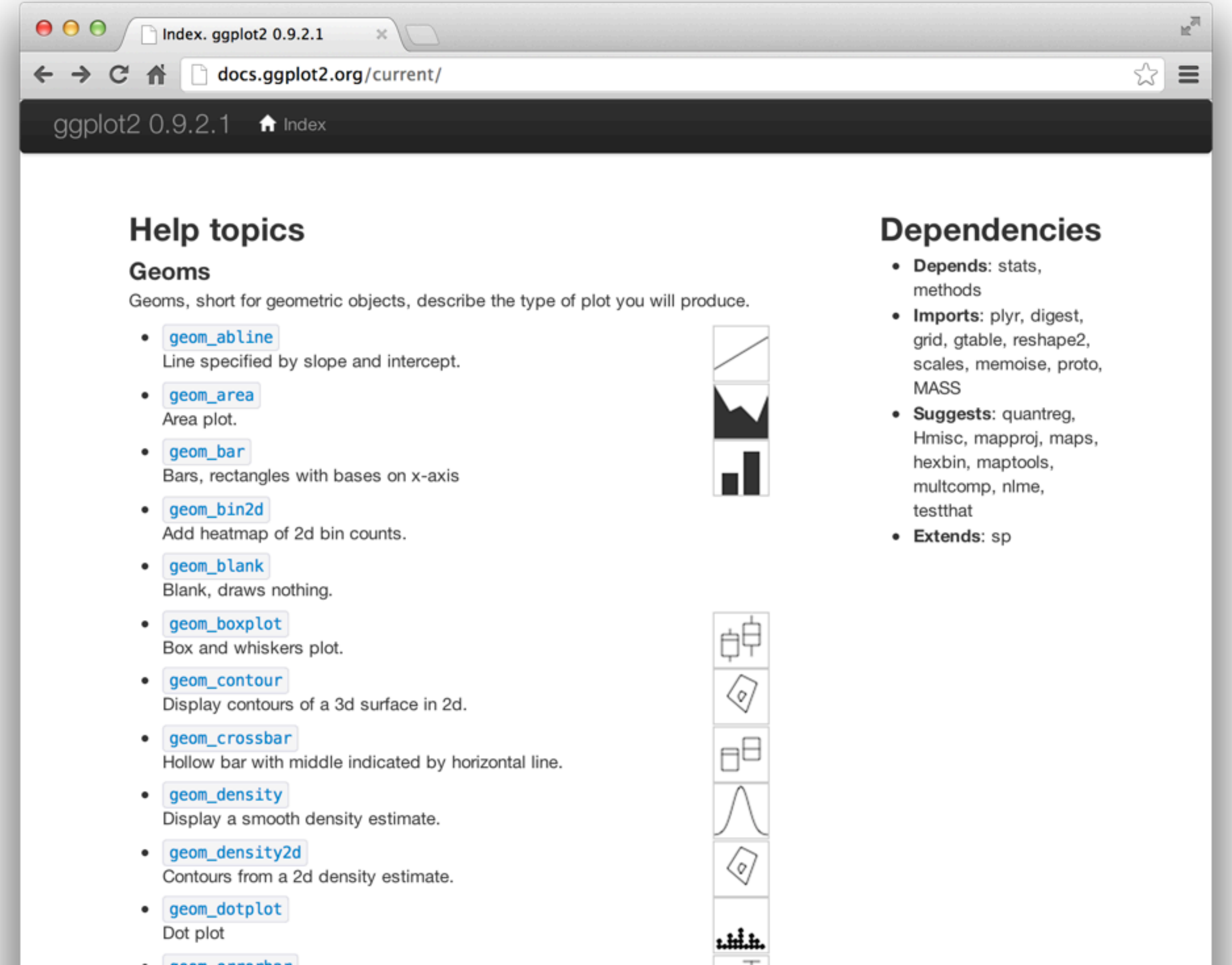
Advantage 1

Geom functions provide a way to look up help pages for specific geoms

`?geom_smooth`

Or even better

http://docs.ggplot2.org/current/



Help topics

Geoms

Geoms, short for geometric objects, describe the type of plot you will produce.

- `geom_abline`
  Line specified by slope and intercept.
- `geom_area`
  Area plot.
- `geom_bar`
  Bars, rectangles with bases on x-axis
- `geom_bin2d`
  Add heatmap of 2d bin counts.
- `geom_blank`
  Blank, draws nothing.
- `geom_boxplot`
  Box and whiskers plot.
- `geom_contour`
  Display contours of a 3d surface in 2d.
- `geom_crossbar`
  Hollow bar with middle indicated by horizontal line.
- `geom_density`
  Display a smooth density estimate.
- `geom_density2d`
  Contours from a 2d density estimate.
- `geom_dotplot`
  Dot plot

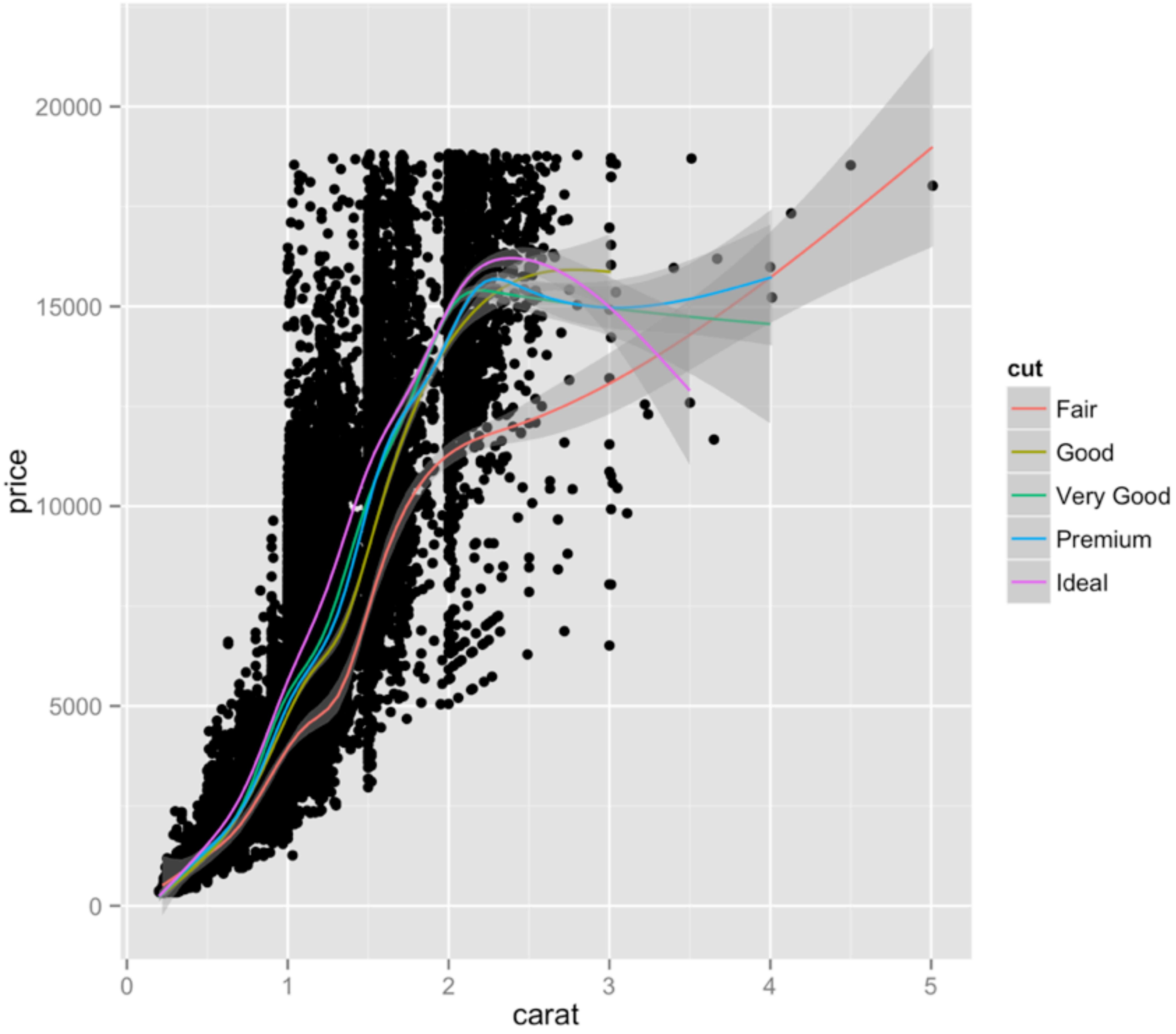
Dependencies

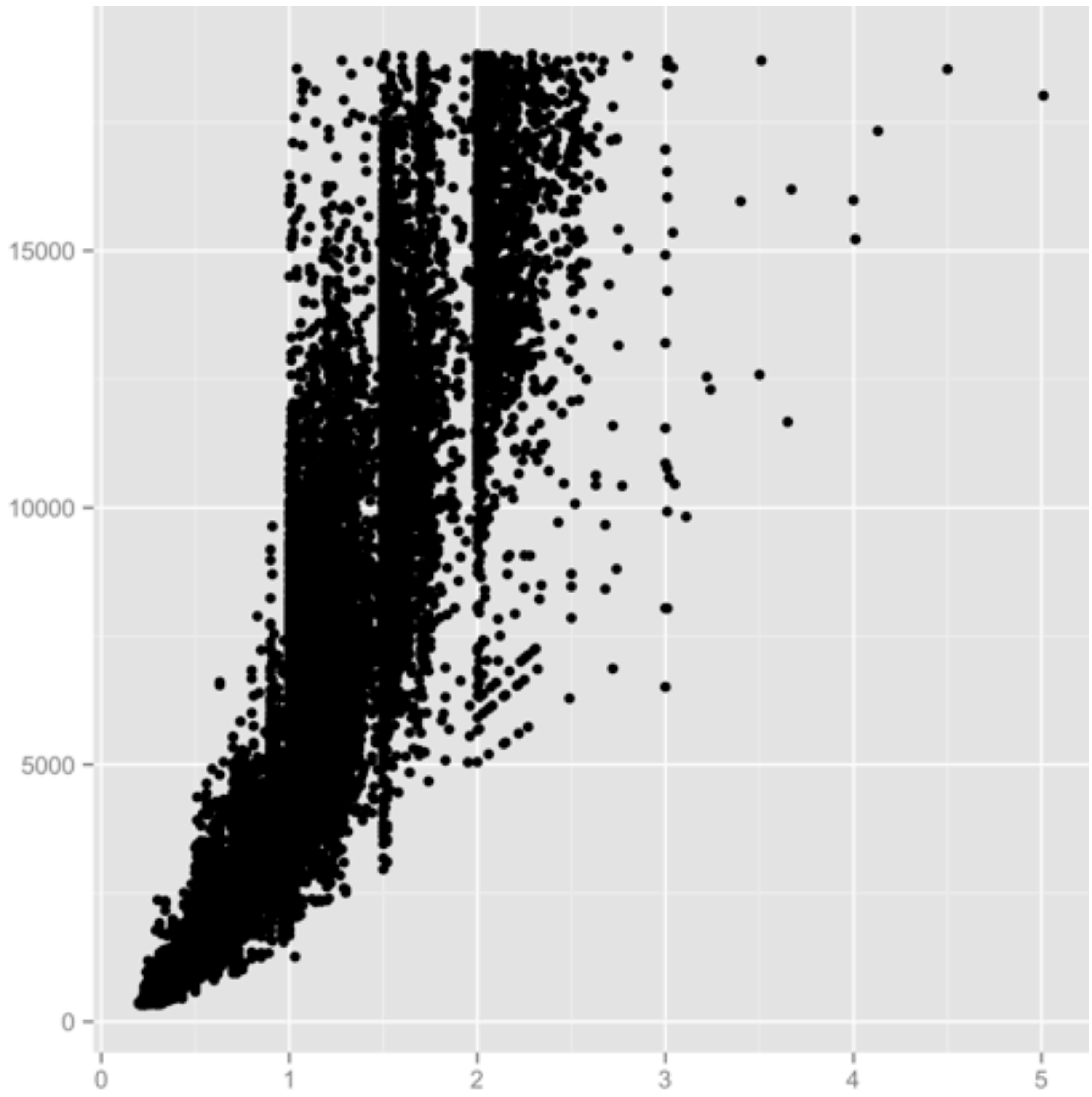
- **Depends**: stats, methods
- **Imports**: plyr, digest, grid, gtable, reshape2, scales, memoise, proto, MASS
- **Suggests**: quantreg, Hmisc, mapproj, maps, hexbin, maptools, multcomp, nlme, testthat
- **Extends**: sp

Advantage 2

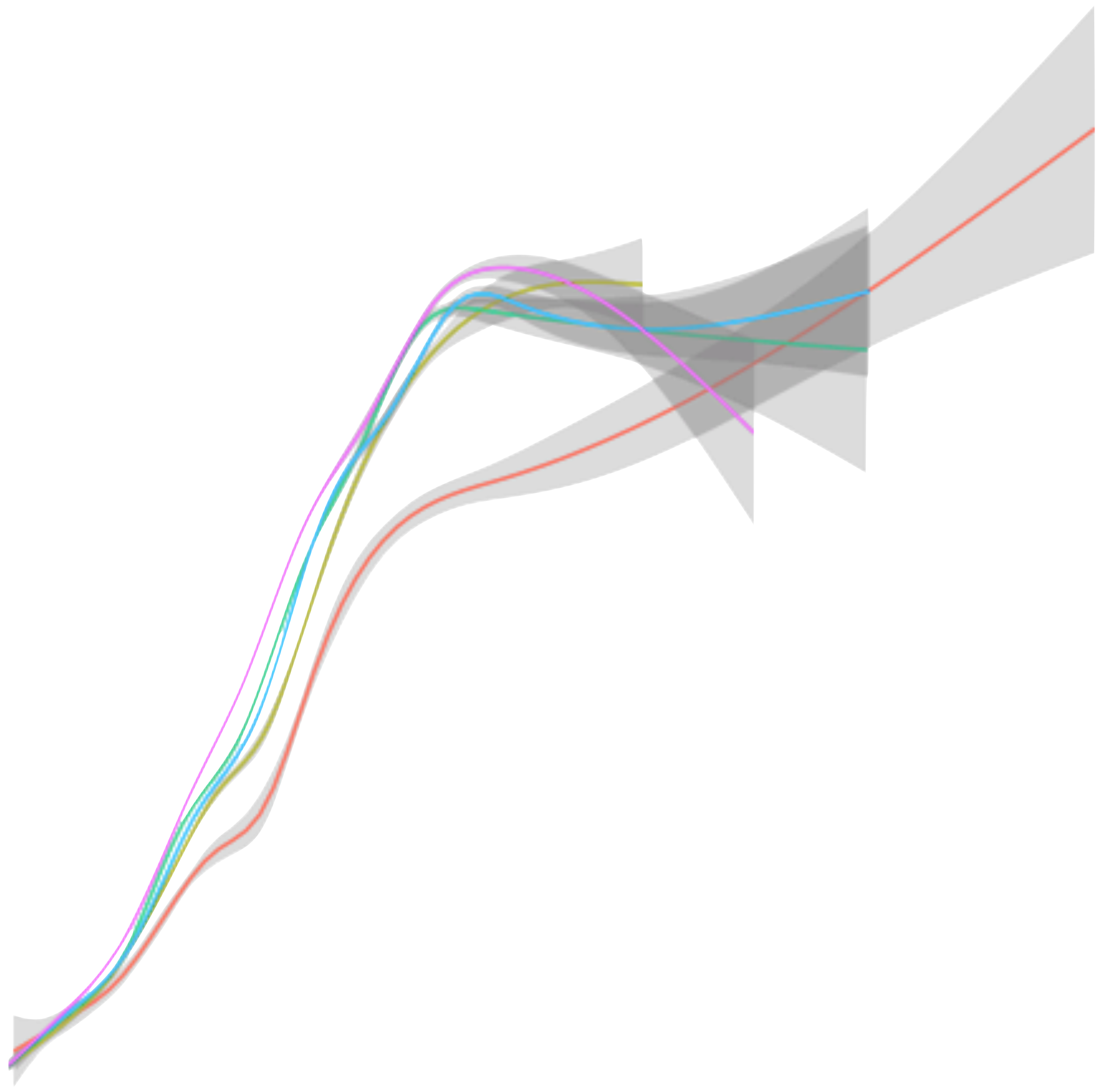
Geom functions provide a way to create separate layers with different aesthetics

What is a layer?

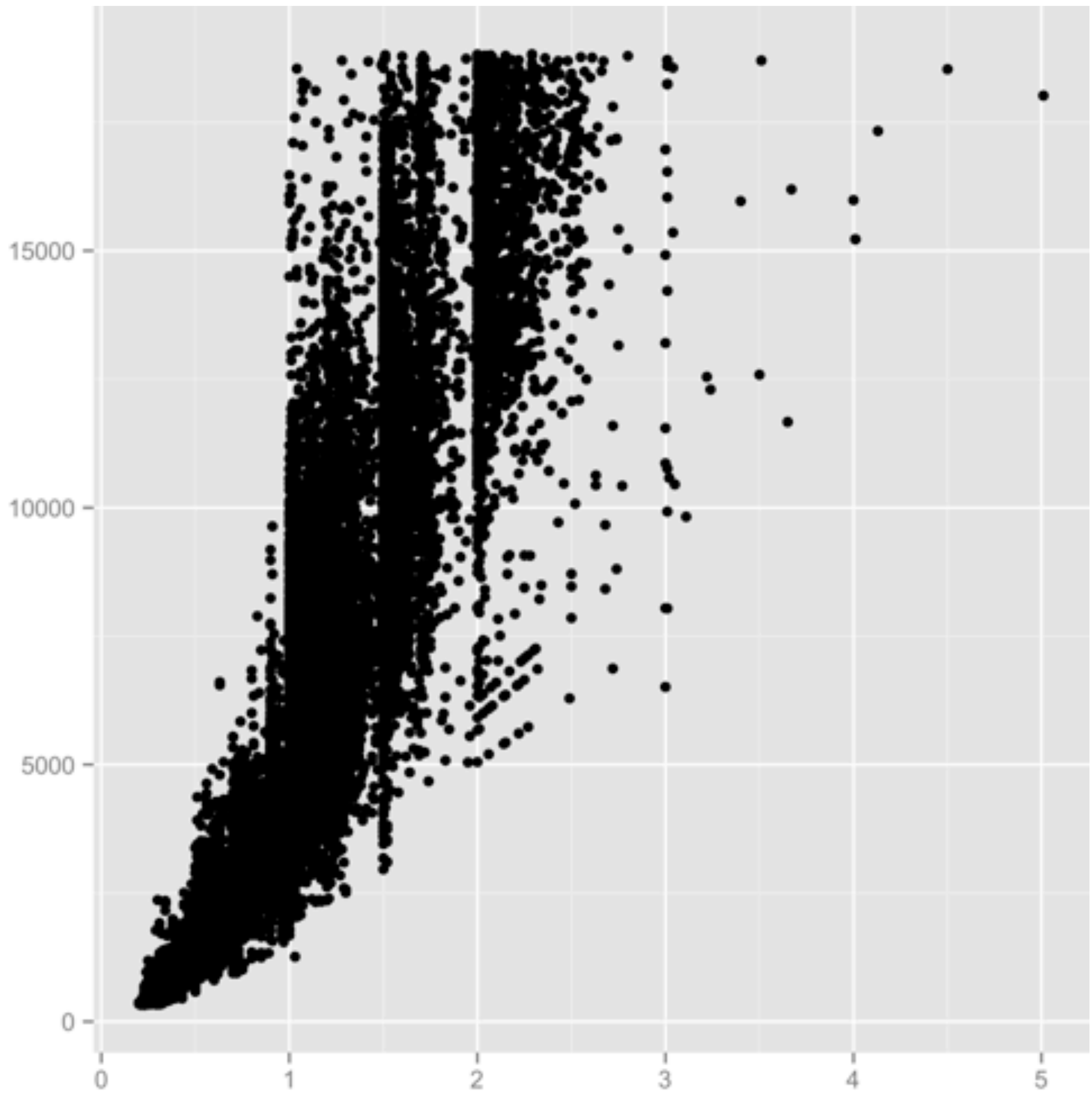

20000
15000
10000
5000
0
price
0
1
2
3
4
5
carat
cut
Fair
Good
Very Good
Premium
Ideal

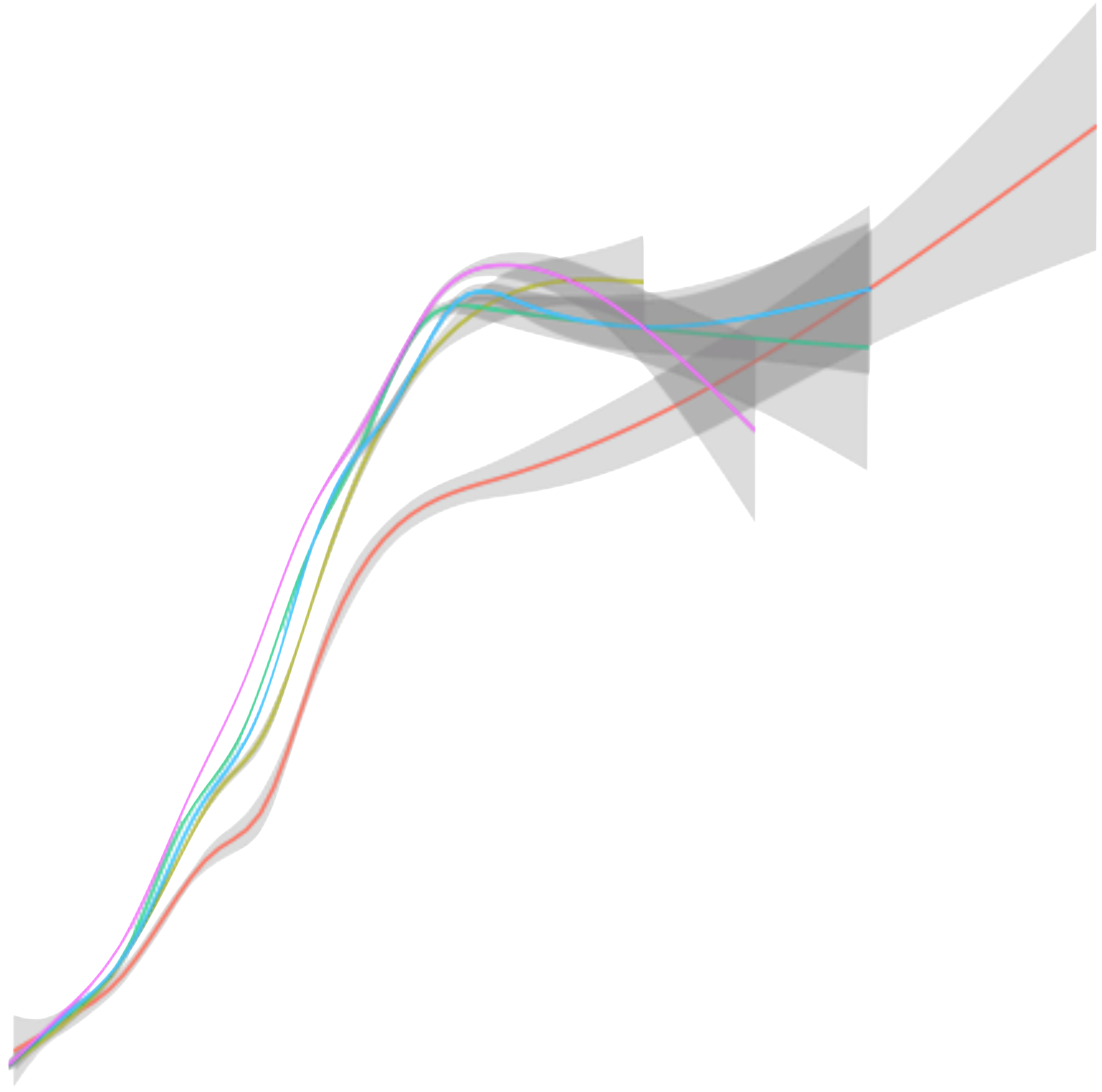


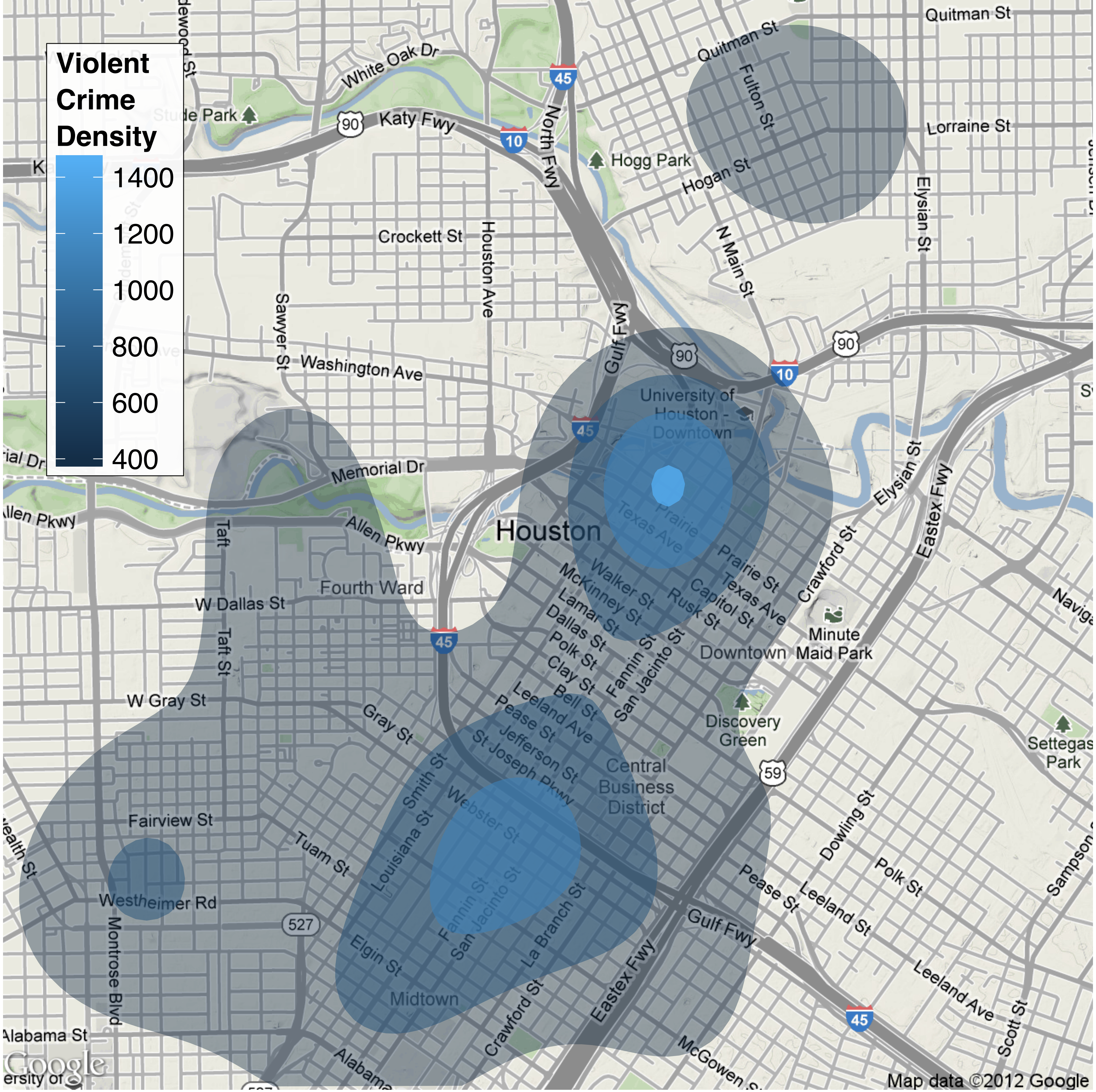
Layer 1



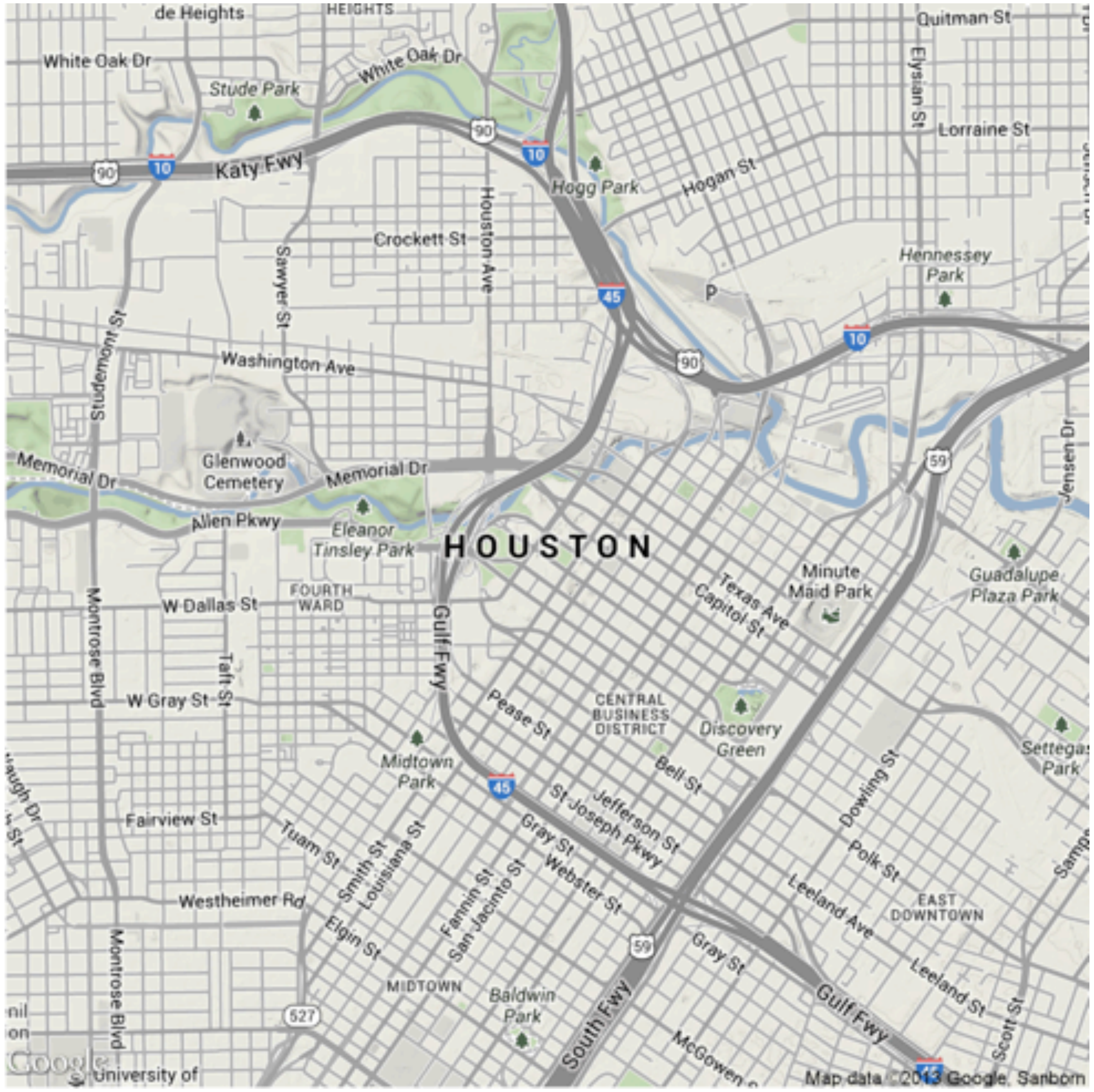
Layer 2

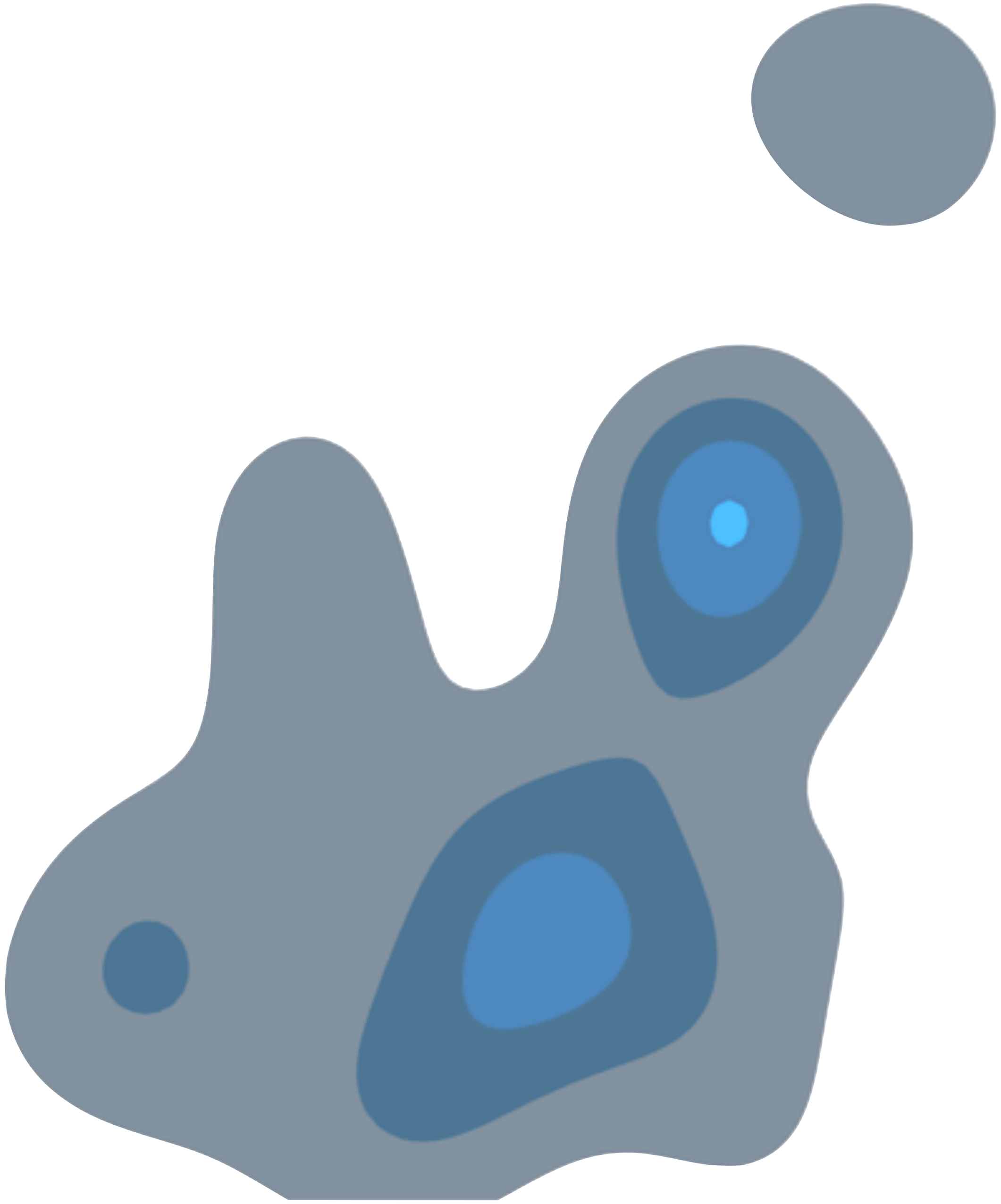

15000
10000
5000
0
0
1
2
3
4
5





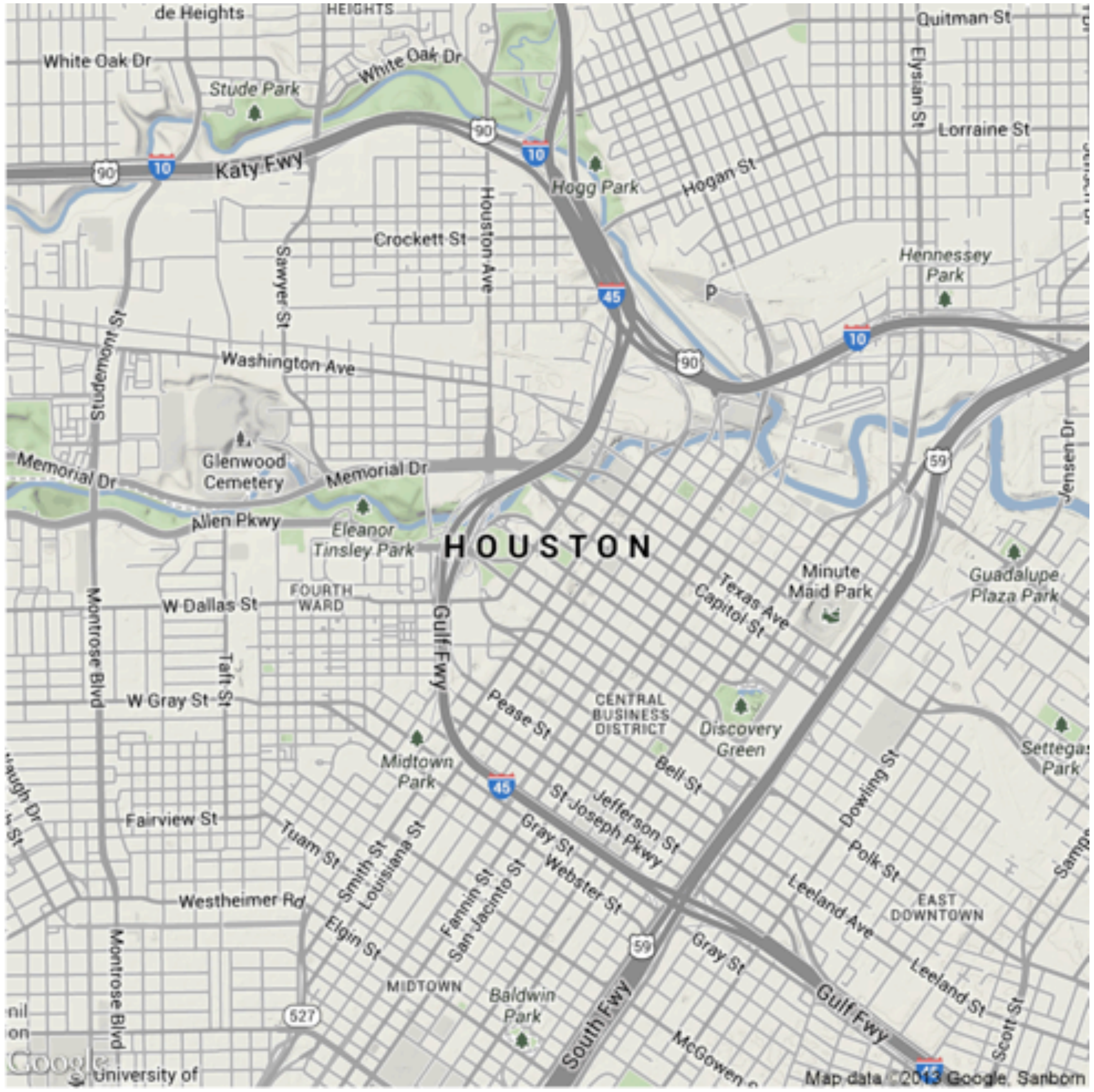
David Kahle, https://dl.dropbox.com/u/24648660/ggmap%20useR%202012.pdf

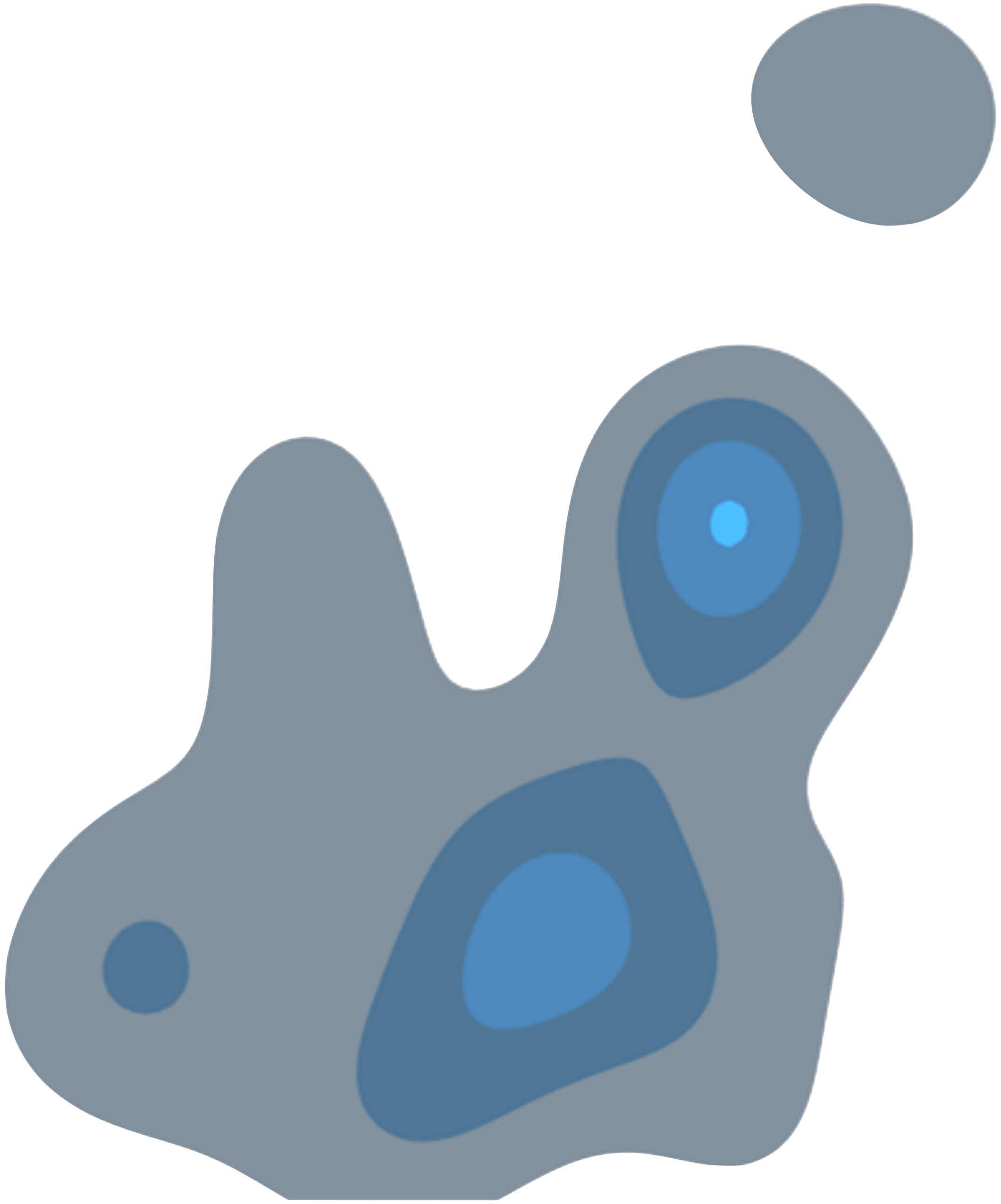




Layer 1

Layer 2


de Heights
HEIGHTS
Quitman St
White Oak Dr
Stude Park
White Oak Dr
Elysian St
Lorraine St
90
10
Katy Fwy
90
10
Hogg Park
Hogan St
Houston Ave
Crockett St
Sawyer St
45
P
Hennessey Park
Studemont St
Washington Ave
90
10
59
Jensen Dr
Glenwood Cemetery
Memorial Dr
Memorial Dr
Allen Pkwy
Eleanor Tinsley Park
HOUSTON
Texas Ave
Capitol St
Minute Maid Park
Guadalupe Plaza Park
FOURTH WARD
W Dallas St
Gulf Fwy
Montrose Blvd
Taft St
W Gray St
Pease St
CENTRAL BUSINESS DISTRICT
Discovery Green
Settegast Park
Midtown Park
Bell St
Dowling St
45
St Joseph Pkwy
Jefferson St
Fairview St
Tuam St
Smith St
Louisiana St
Gray St
Polk St
Webster St
Fannin St
San Jacinto St
Leeland Ave
EAST DOWNTOWN
Westheimer Rd
Elgin St
59
Gray St
Leeland St
MIDTOWN
Baldwin Park
527
Montrose Blvd
South Fwy
Gulf Fwy
Scott St
McGowen St
Google
University of
Map data ©2013 Google, Sanborn



What is a layer?

Coordinate system

+ geom

+ data

+ aesthetic mappings

+ position adjustment

+ stat

A layer is a collection of these

What is a plot?

Coordinate system

+ a layer

+ a layer

+ a layer

+ ...

ggplot

`ggplot` provides an alternative way to build graphs based on this system.

Its more complicated than `qplot`, but gives you more control.

ggplot

Coordinate system

+ a layer

+ a layer

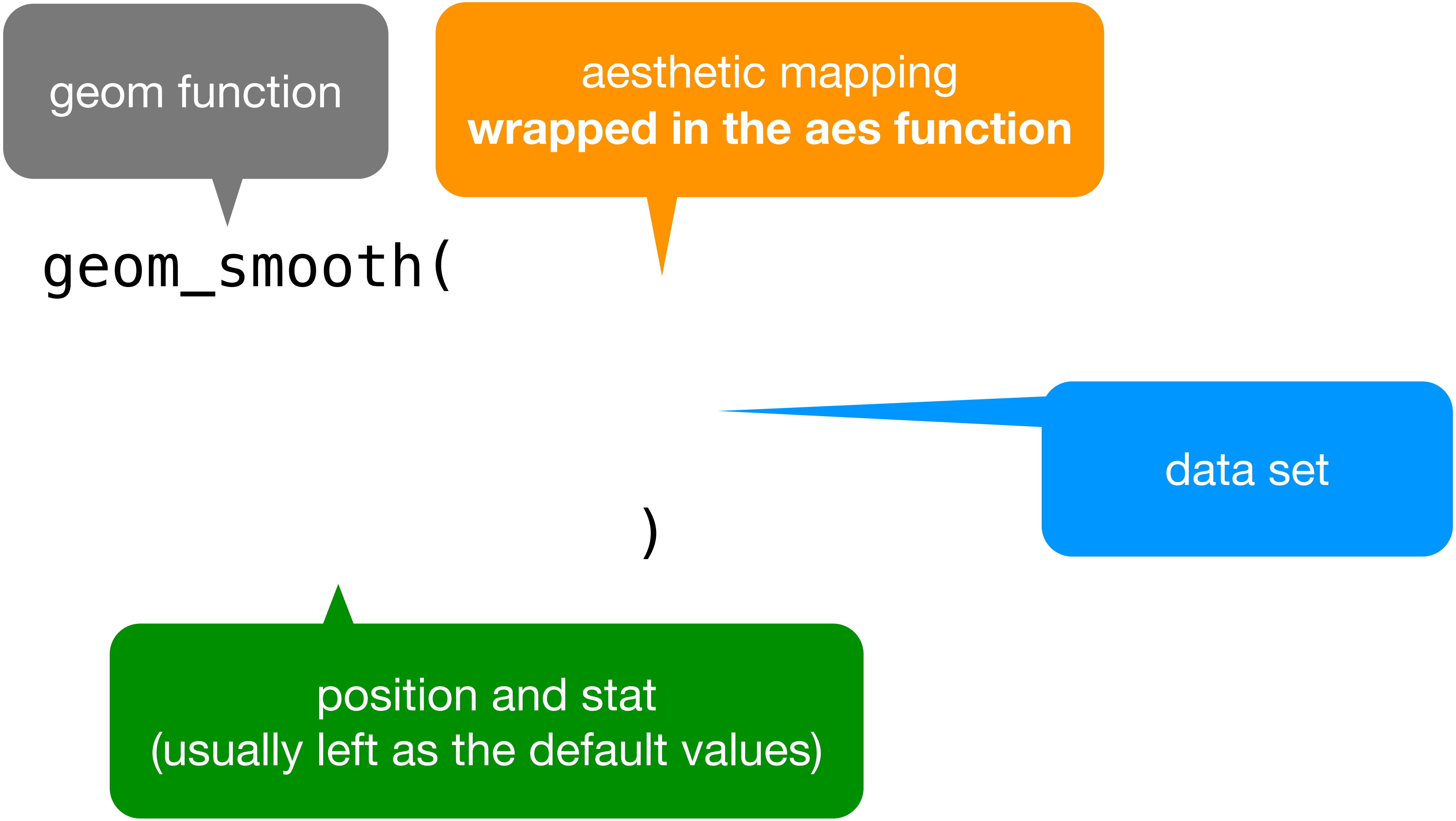
+ a layer

+ ...

```
ggplot() +
  # a layer +
  # a layer +
  # a layer
  # ...
```

But how to build the layers?

Ever geom is a layer



geom function

aesthetic mapping
wrapped in the aes function

```
geom_smooth(



                  )
```

data set

position and stat
(usually left as the default values)

ggplot

Coordinate system

+ a layer

+ a layer

+ a layer

+ ...

```
ggplot() +
  geom_point(
    aes(x = carat, y = price),
    data = diamonds) +
  geom_smooth(
    aes(x = carat, y = price,
        color = cut),
    data = diamonds)
```

```
ggplot() +
  geom_point(aes(x = carat, y = price),
    data = diamonds) +
  geom_smooth(aes(x = carat, y = price,
    color = cut), data = diamonds)
```

A lot of redundant typing

```
ggplot() +
  geom_point(aes(x = carat, y = price),
    data = diamonds) +
  geom_smooth(aes(x = carat, y = price,
    color = cut), data = diamonds)
```

A lot of redundant typing

```
ggplot() +
  geom_point(aes(x = carat, y = price),
    data = diamonds) +
  geom_smooth(aes(x = carat, y = price,
    color = cut), data = diamonds)
```

A lot of redundant typing

set default data and aesthetics with ggplot

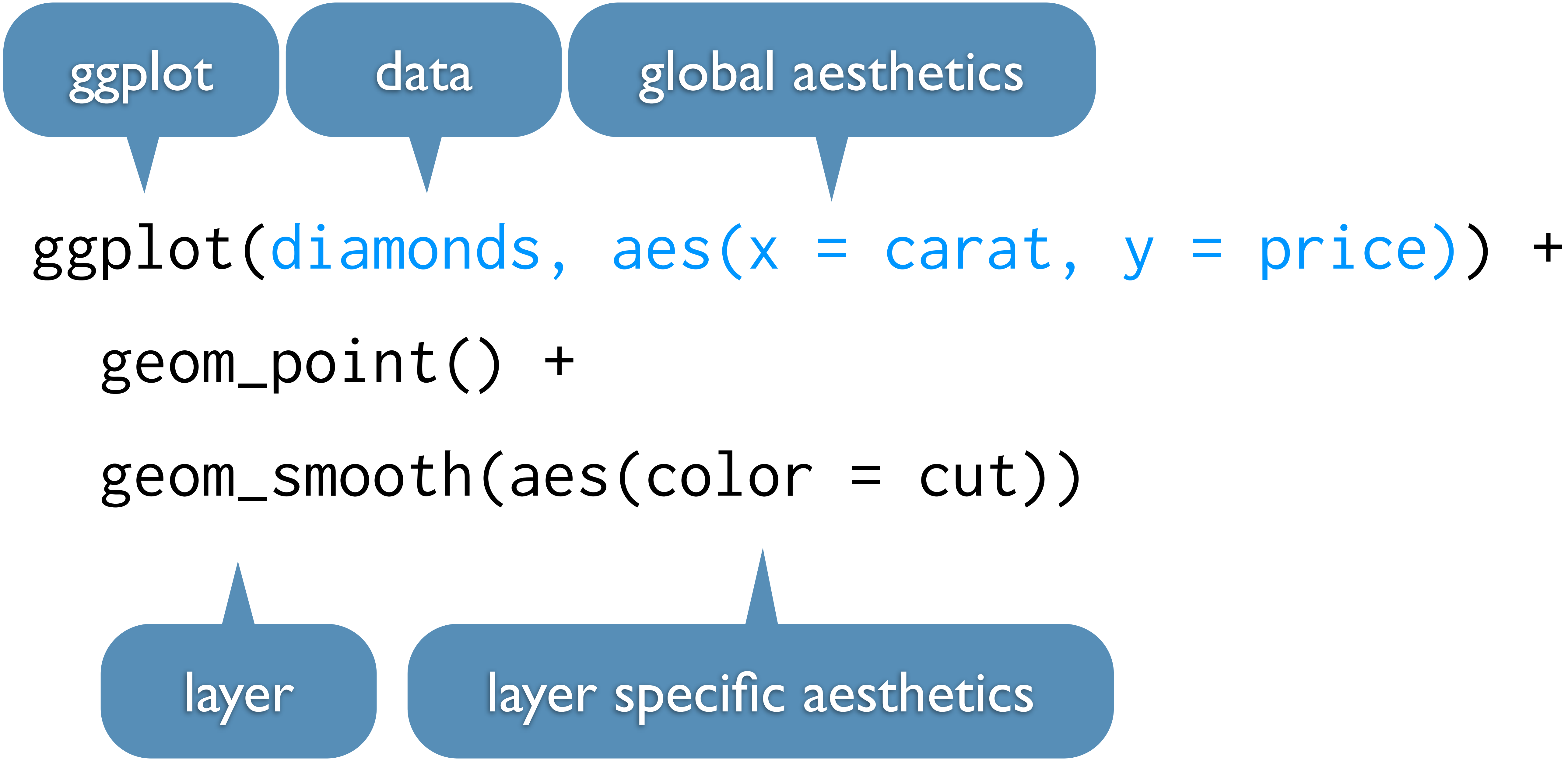
ggplot

default data set

default aesthetic mappings **wrapped in the aes function**

```
ggplot(diamonds, aes(x = carat, y = price))
```

You can overwrite or add to the defaults at the layer level



ggplot

data

global aesthetics

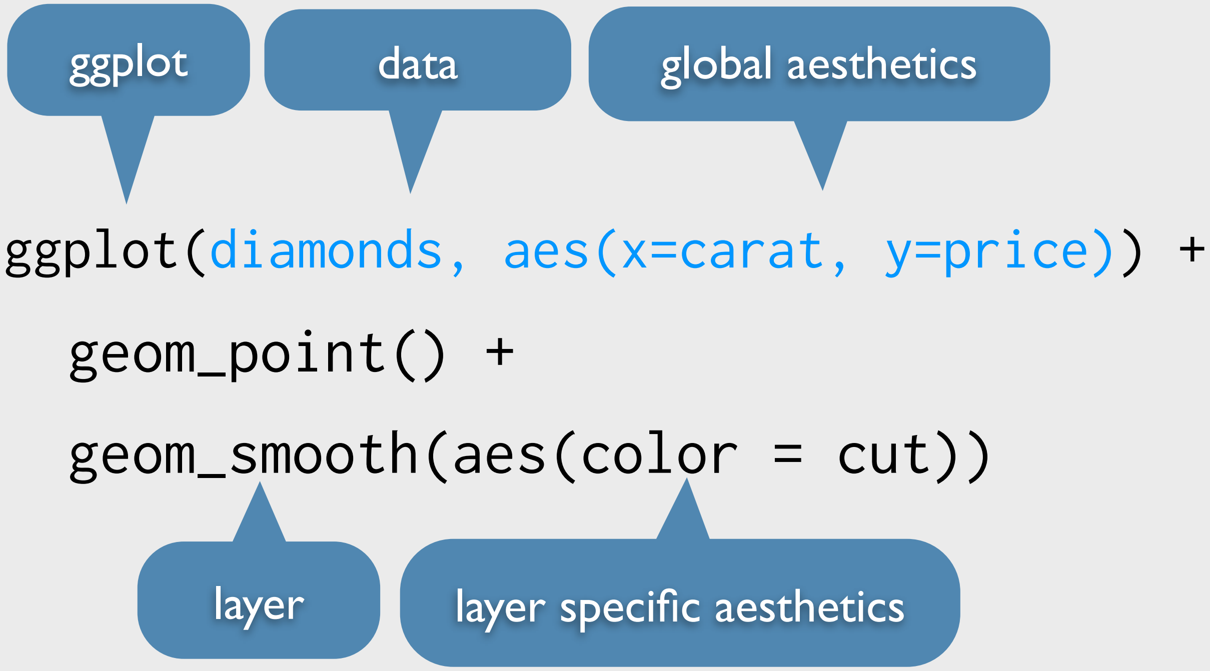
```
ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point() +
  geom_smooth(aes(color = cut))
```

layer

layer specific aesthetics

Your turn

Use `ggplot` to make these graphs.



ggplot

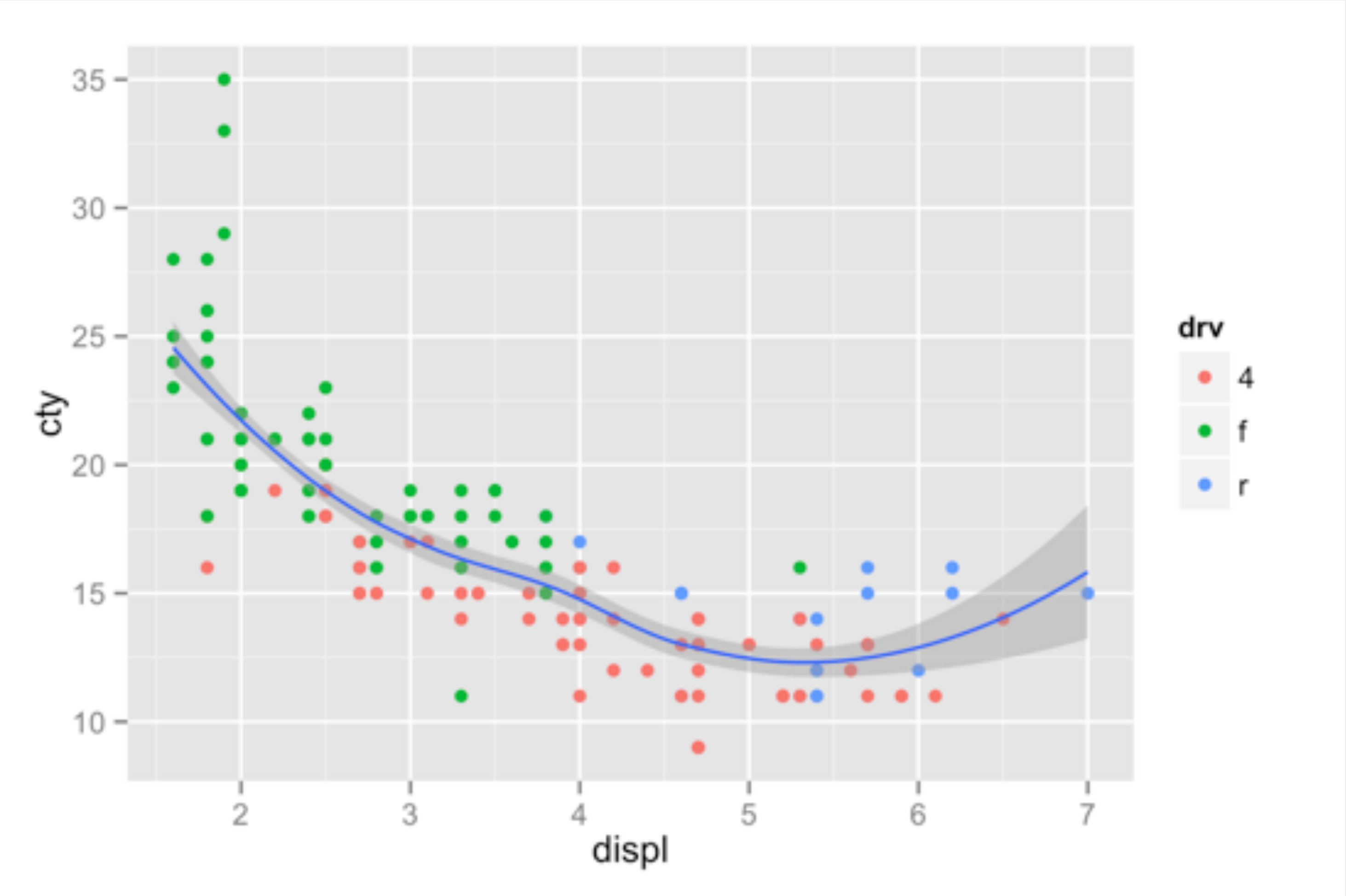
data

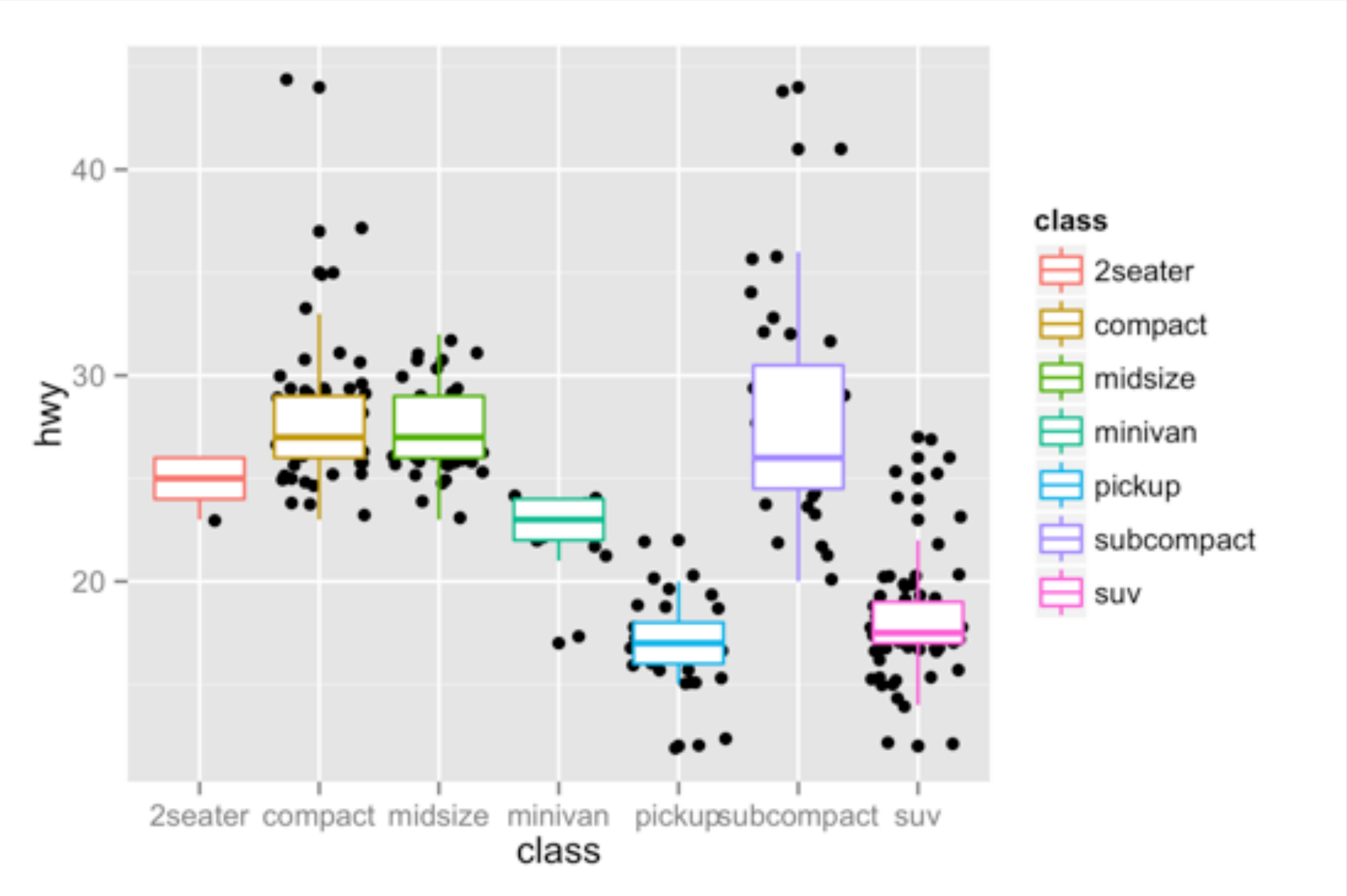
global aesthetics

```
ggplot(diamonds, aes(x=carat, y=price)) +
  geom_point() +
  geom_smooth(aes(color = cut))
```

layer

layer specific aesthetics





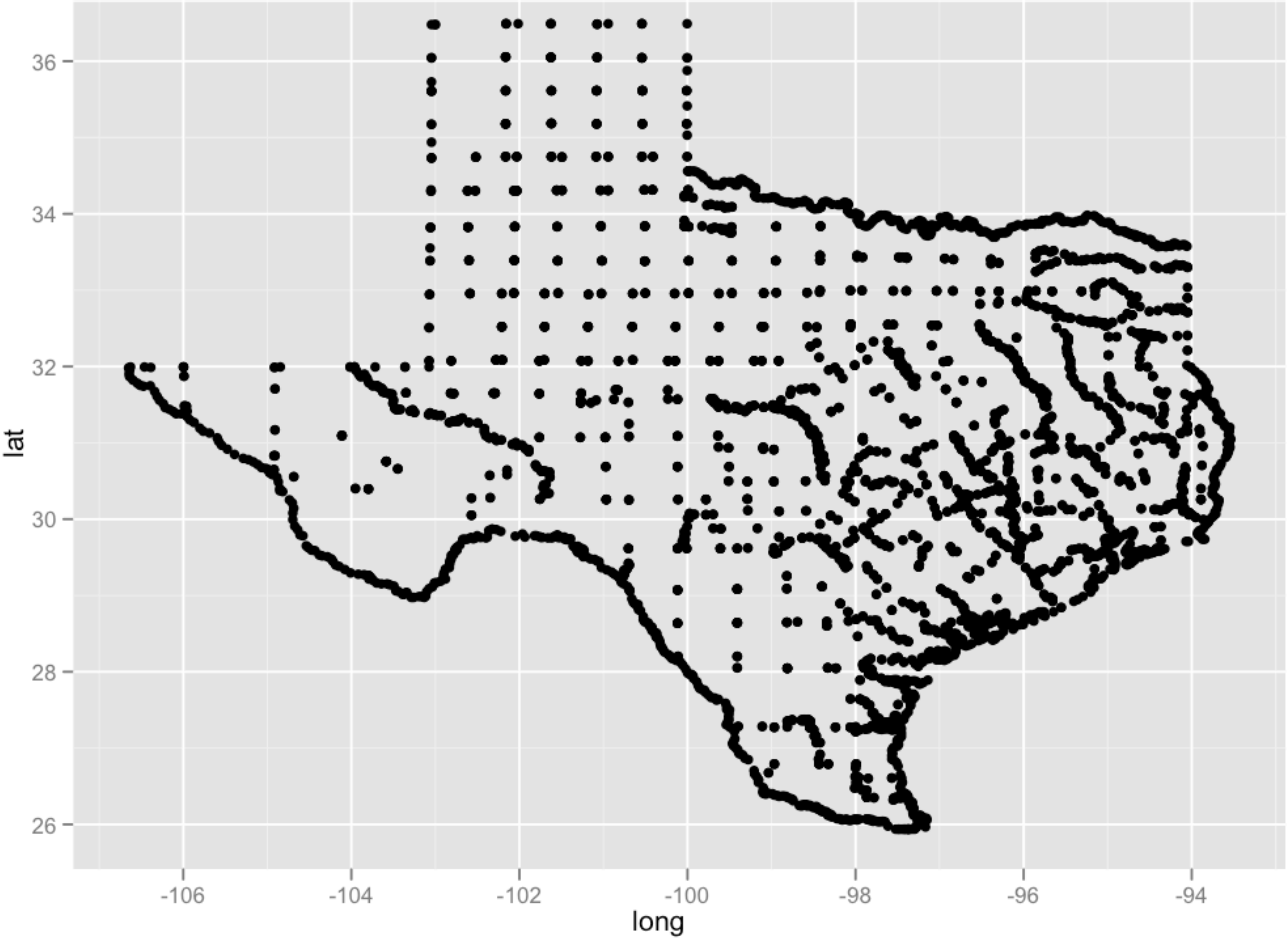
```
ggplot(mpg, aes(displ, cty)) +
  geom_point(aes(color = drv)) +
  geom_smooth()

ggplot(mpg, aes(class, hwy)) +
  geom_point(position = ”jitter”) +
    geom_boxplot(aes(color = class))
```

Customizing graphics

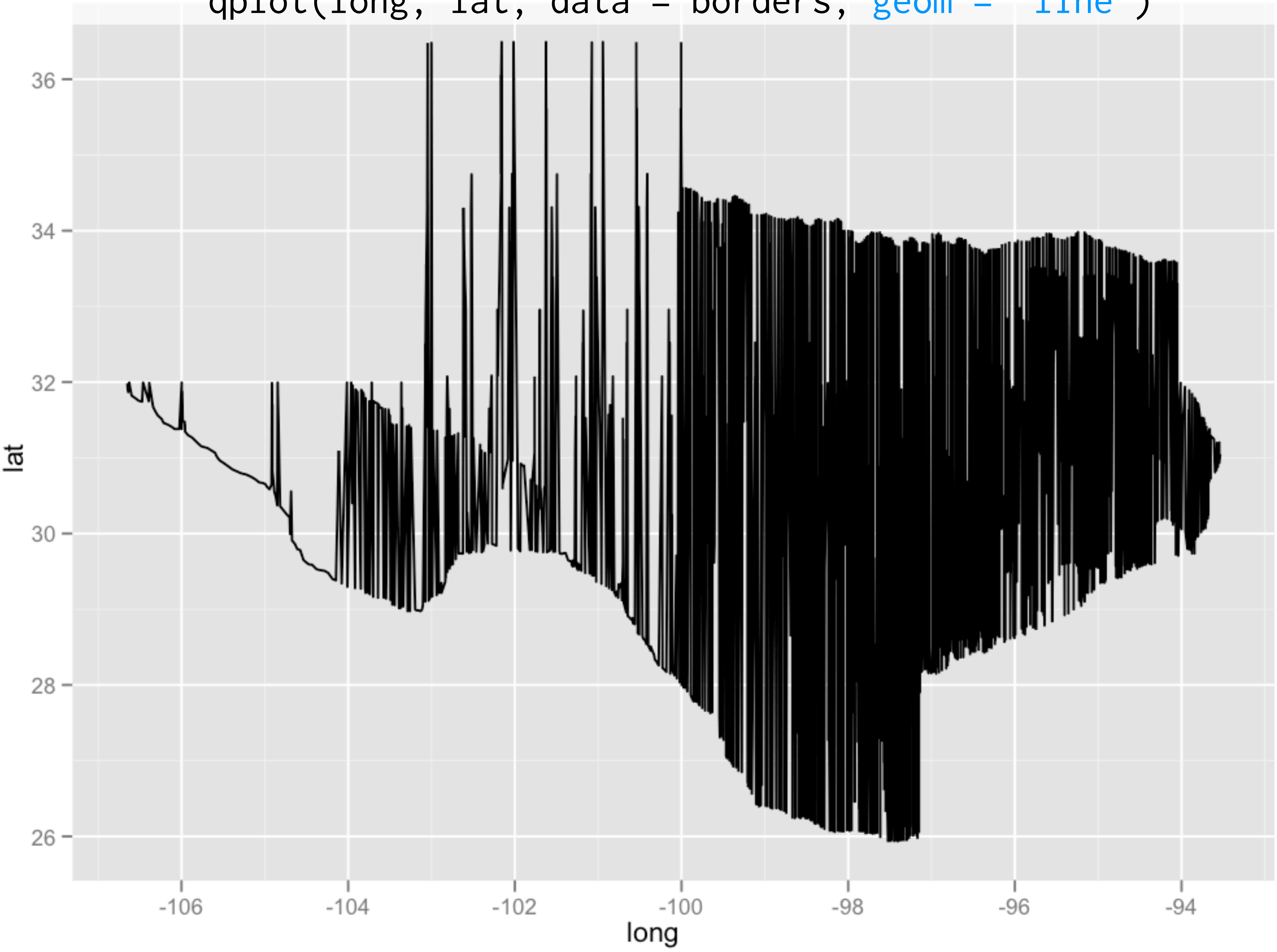
Texas population data

```
borders <- read.csv(”data/texas.csv”)
View(borders)
```

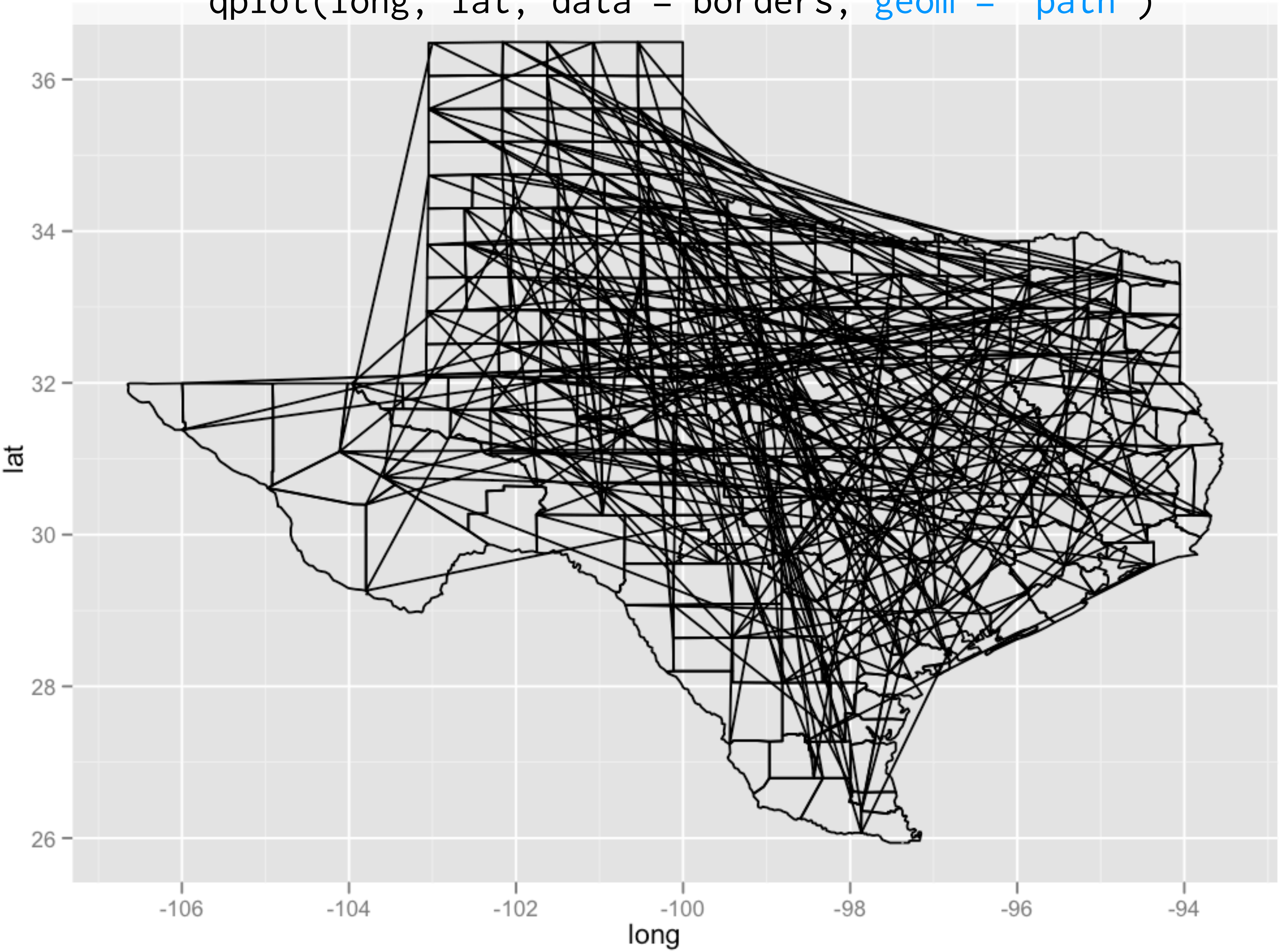
`qplot(long, lat, data = borders)`



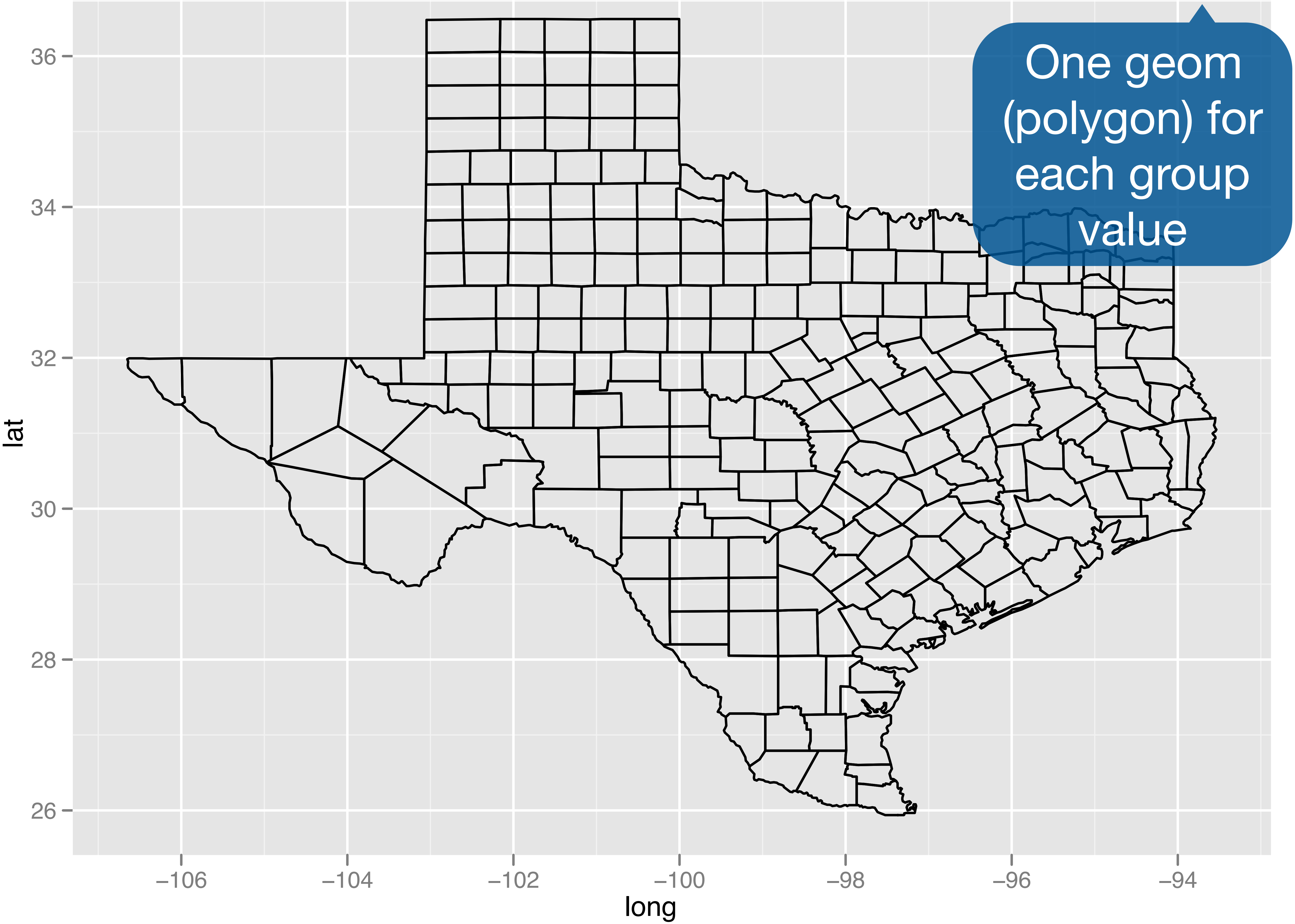
`qplot(long, lat, data = borders, geom = "line")`



```
qplot(long, lat, data = borders, geom = "path")
```

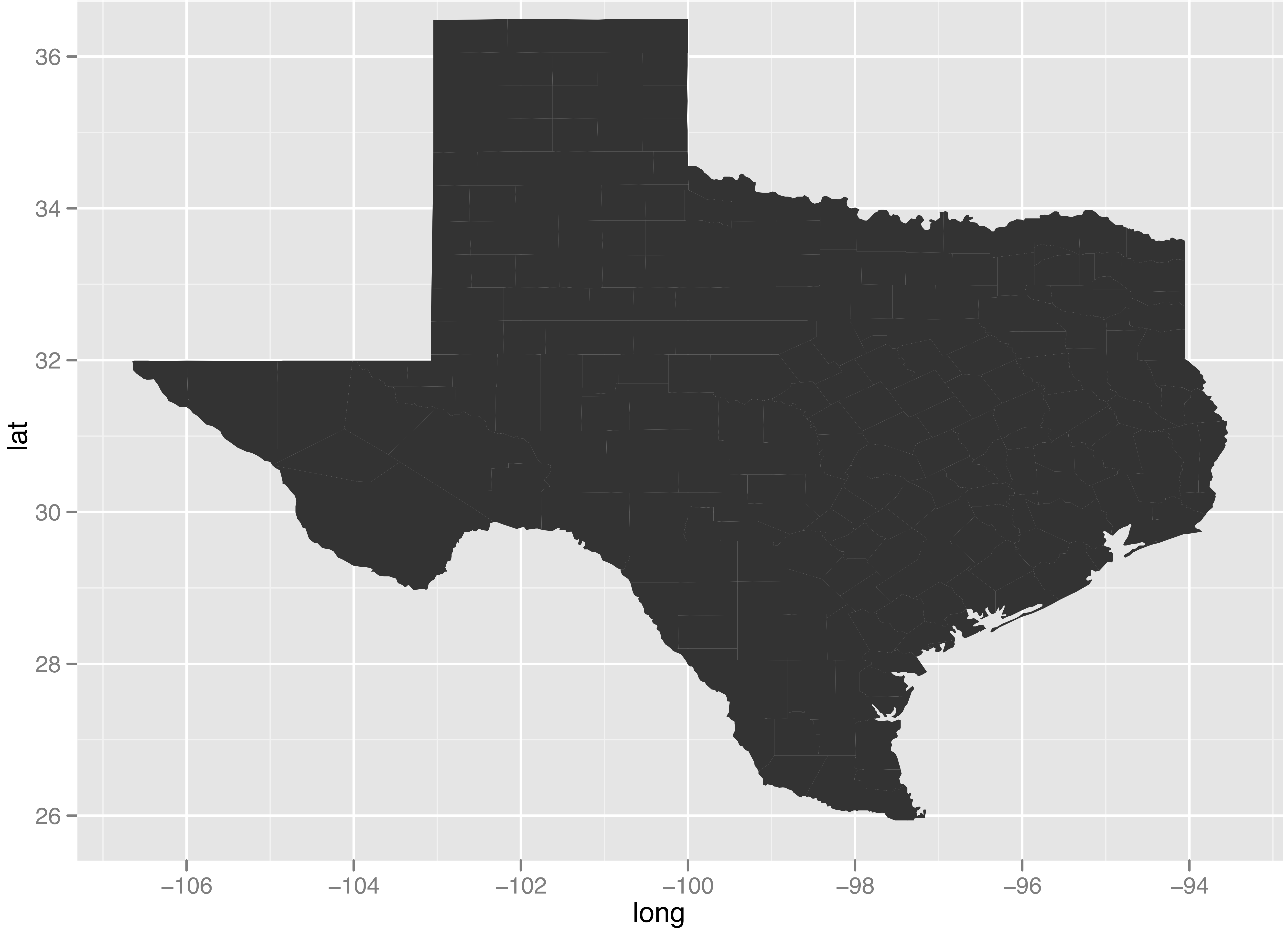


```
qplot(long, lat, data = borders, geom = "path", group = group)
```

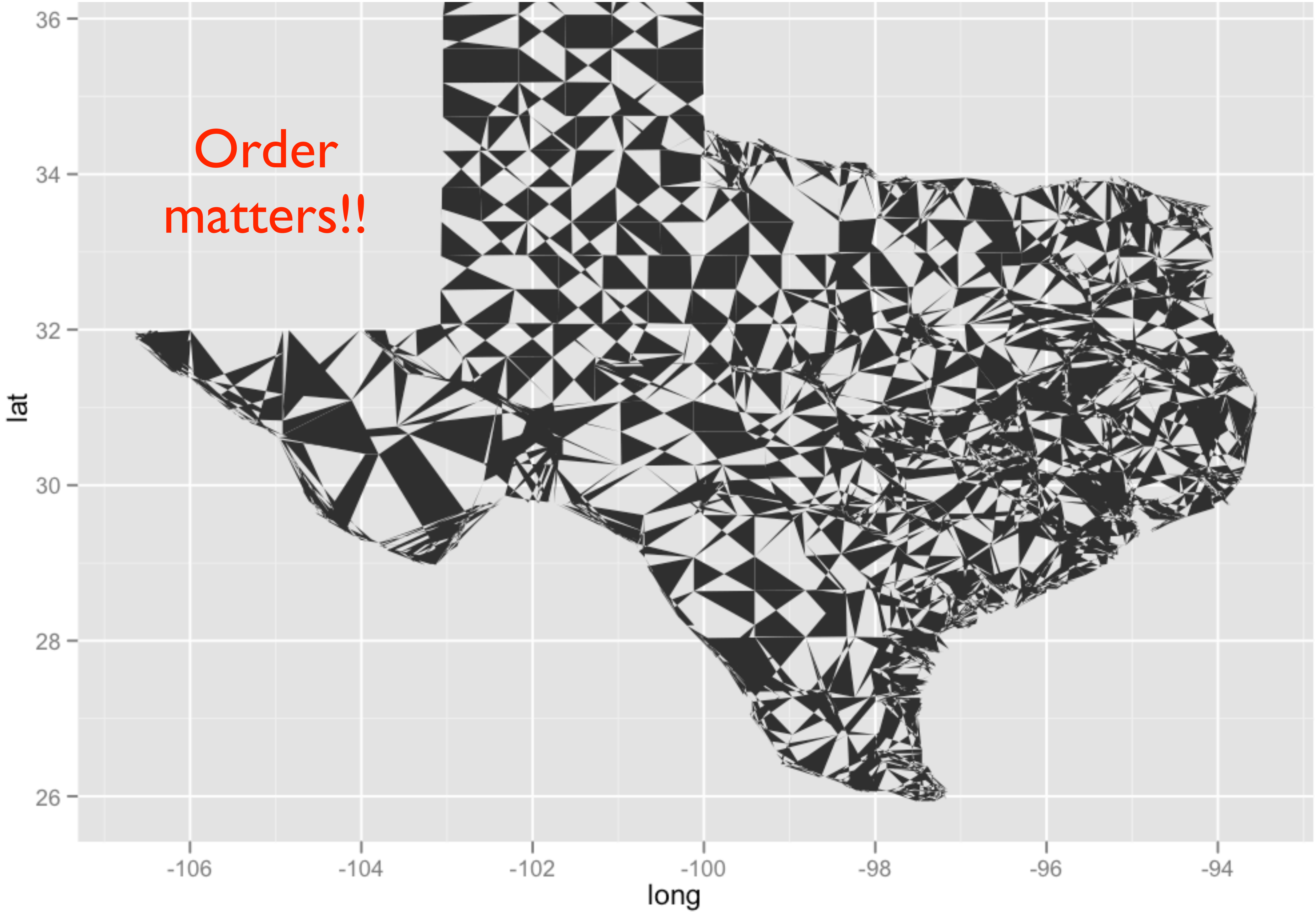


One geom (polygon) for each group value

```
qplot(long, lat, data = borders, geom = "polygon", group = group)
```

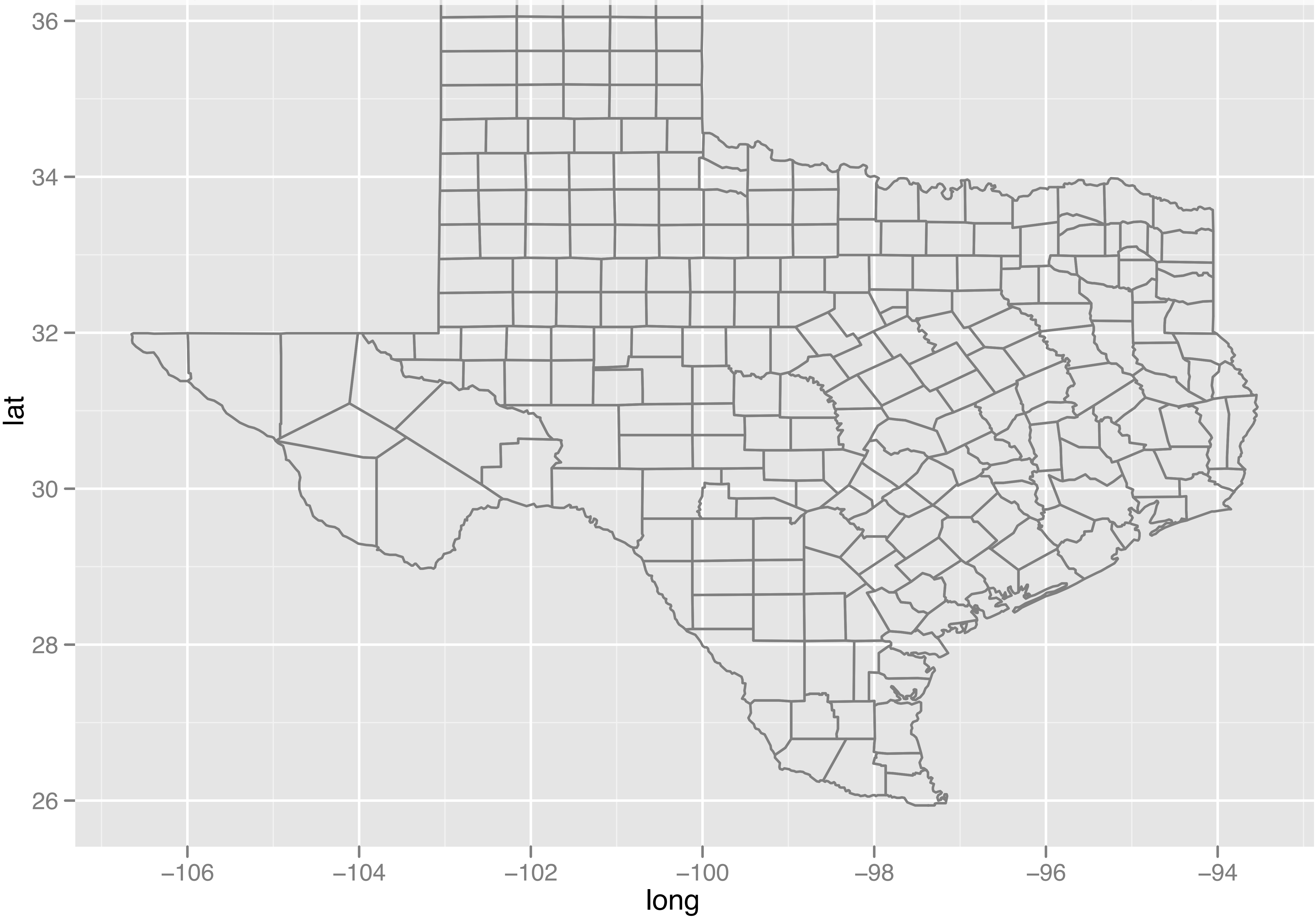


```
borders2 <- borders[sample(nrow(borders)), ]
qplot(long, lat, data = borders2, geom = "polygon", group = group)
```



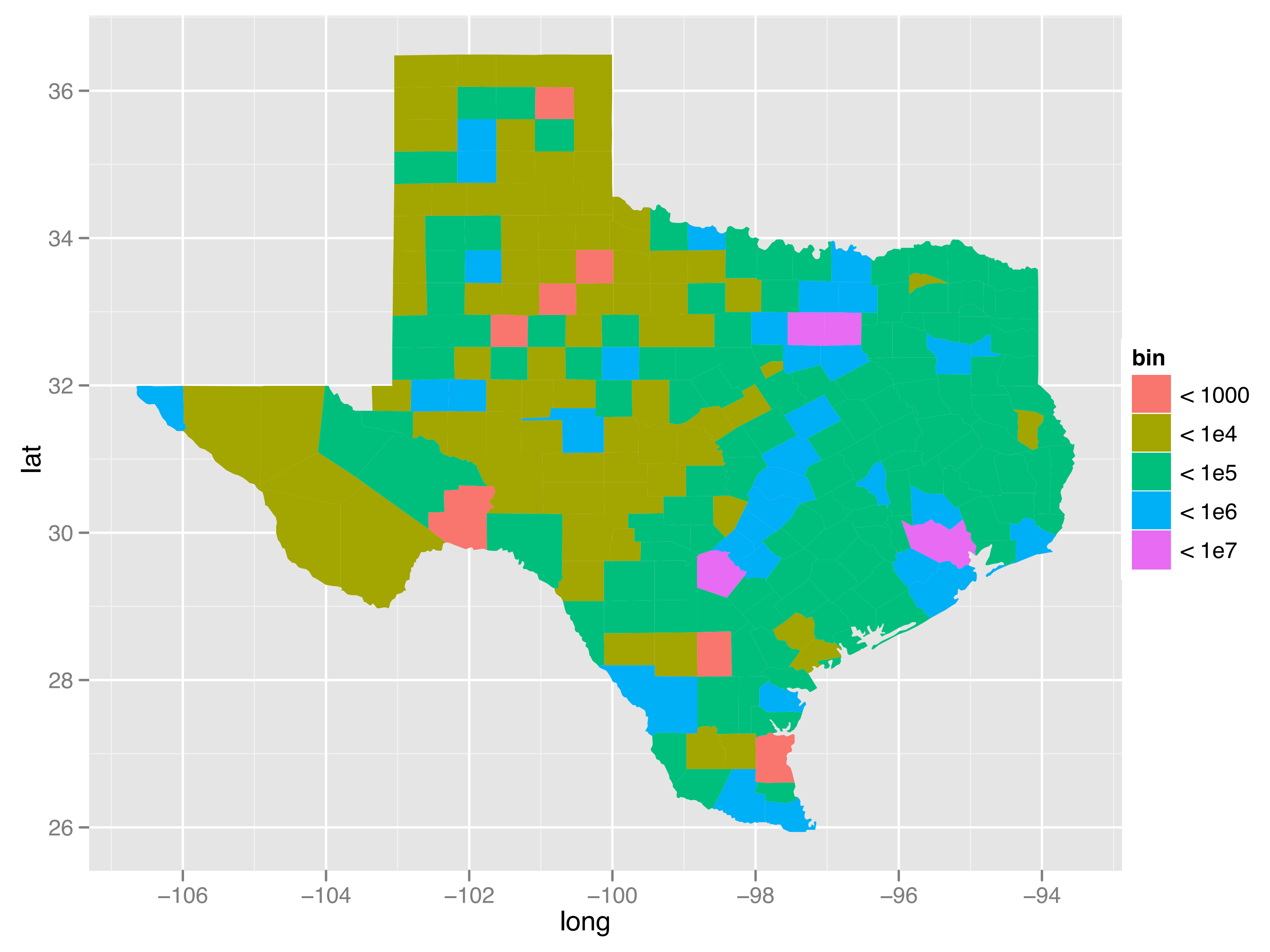
Order matters!!

```
ggplot(borders, aes(long, lat)) +
  geom_polygon(aes(group = group), fill = NA, color = "grey50")
```



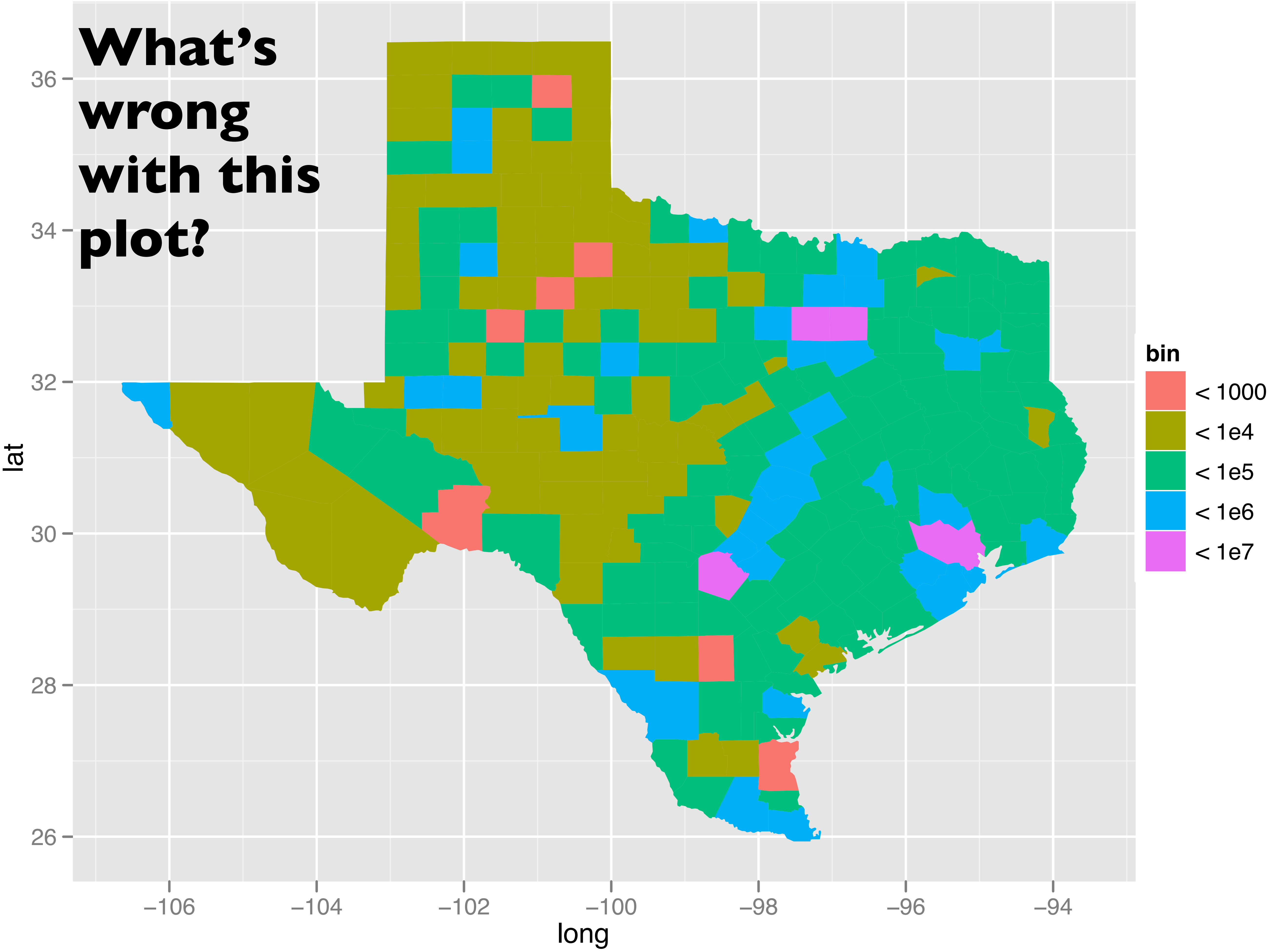
Your Turn

Use `borders` to recreate this map.



```
tx <- qplot(long, lat, data = borders, geom =
"polygon", fill = bin, group = group)

tx
```

What's wrong with this plot?



Some problems

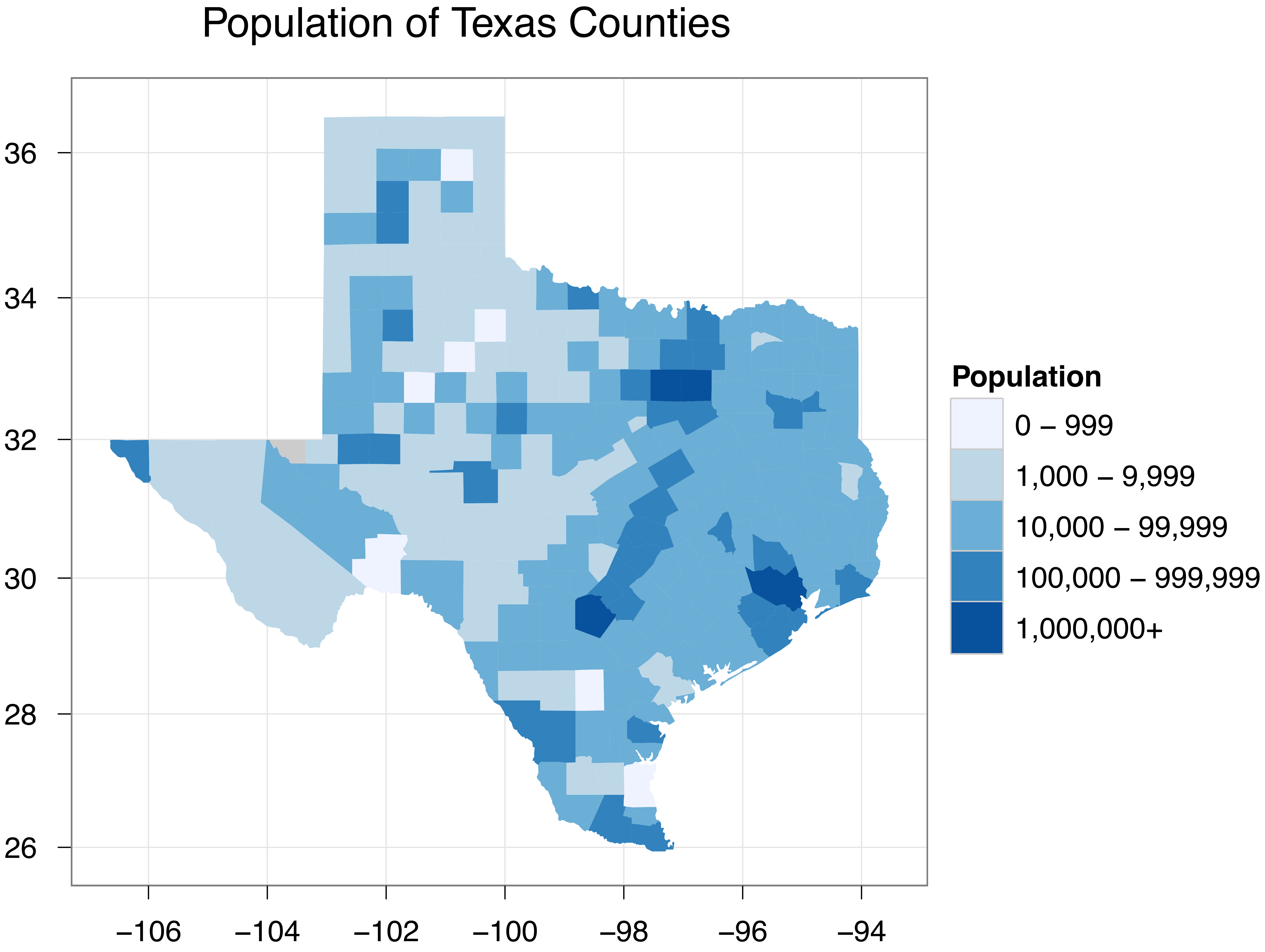
Incorrect aspect ratio

Bad color scheme

Unnecessary axis labels

Legend needs improvement: better title and better key labels

No title



Population of Texas Counties

Title

ggtitle

```
tx + ggtitle(”Population of Texas Counties”)
```

ggplot2 syntax

You modify ggplot2 graphs by adding objects to them.

```
tx + ggtitle(”Population of Texas Counties”)
```

ggplot2 syntax

You modify ggplot2 graphs by adding objects to them.

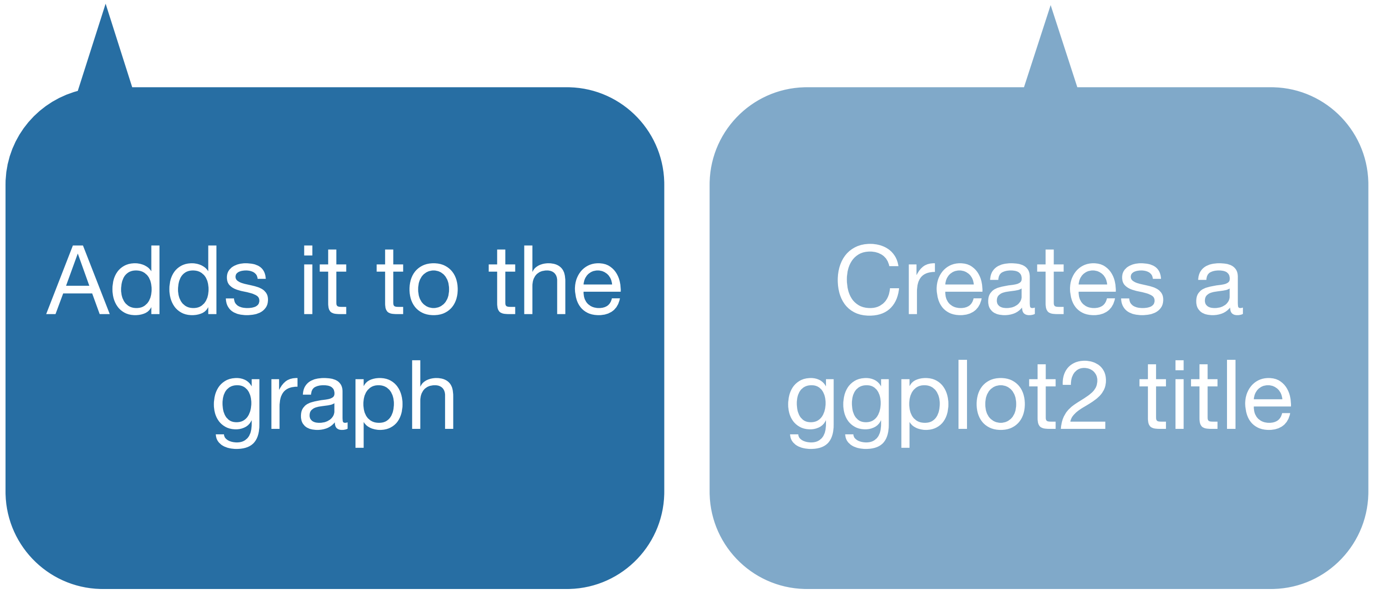
```
tx + ggtitle(”Population of Texas Counties”)
```

Creates a ggplot2 title

ggplot2 syntax

You modify ggplot2 graphs by adding objects to them.

```
tx + ggtitle(”Population of Texas Counties”)
```



Adds it to the graph

Creates a ggplot2 title

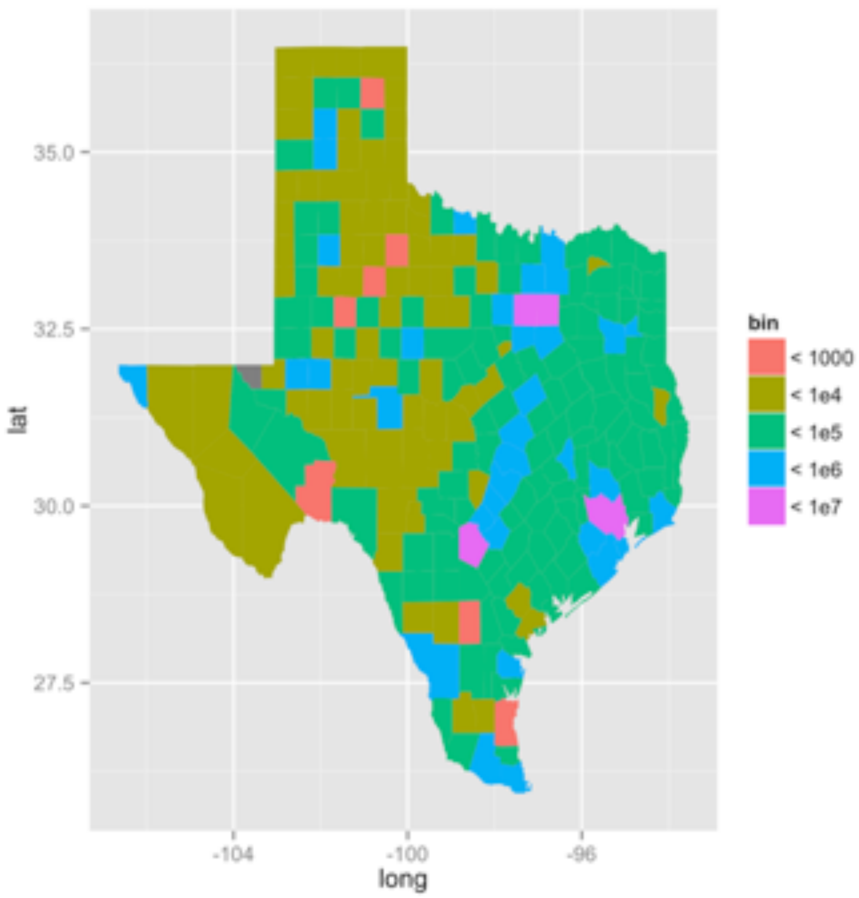
Additions are not permanent. They just affect the current graph being drawn

```
tx + ggtitle(”Population of Texas Counties”)

tx



# to create a new graph that always has a title

tx2 <- tx + ggtitle(”Population of Texas Counties”)

tx2
```

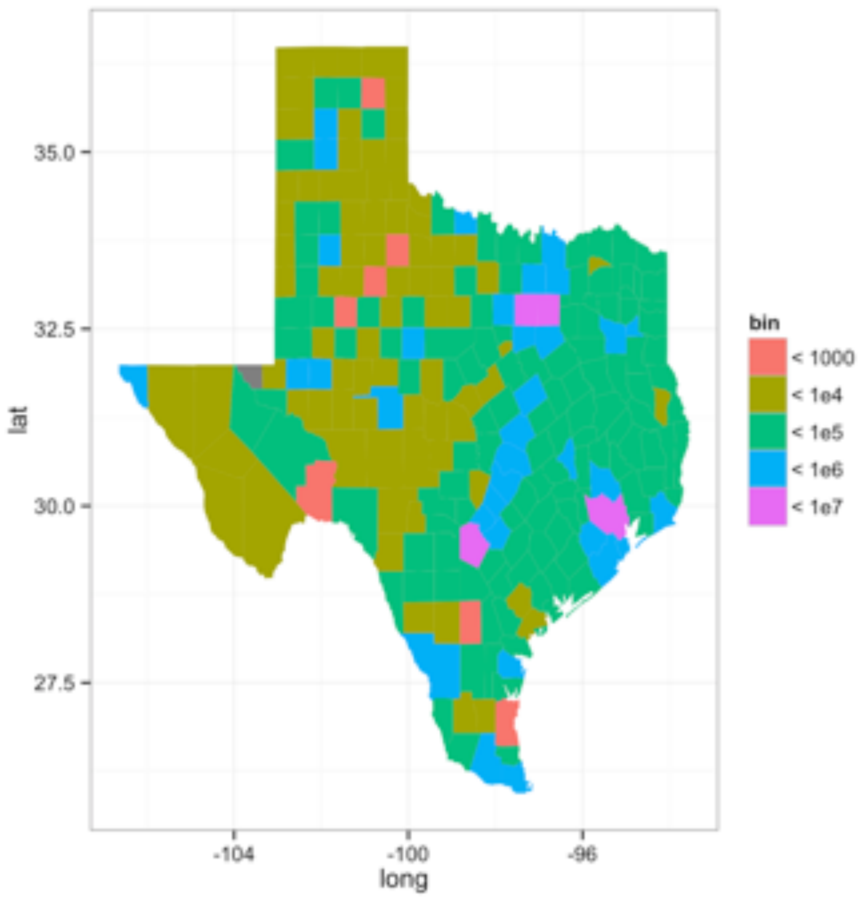
Themes

themes

ggplot2 comes with two pre-loaded themes that control the appearance of non-data elements



`tx + theme_grey()`



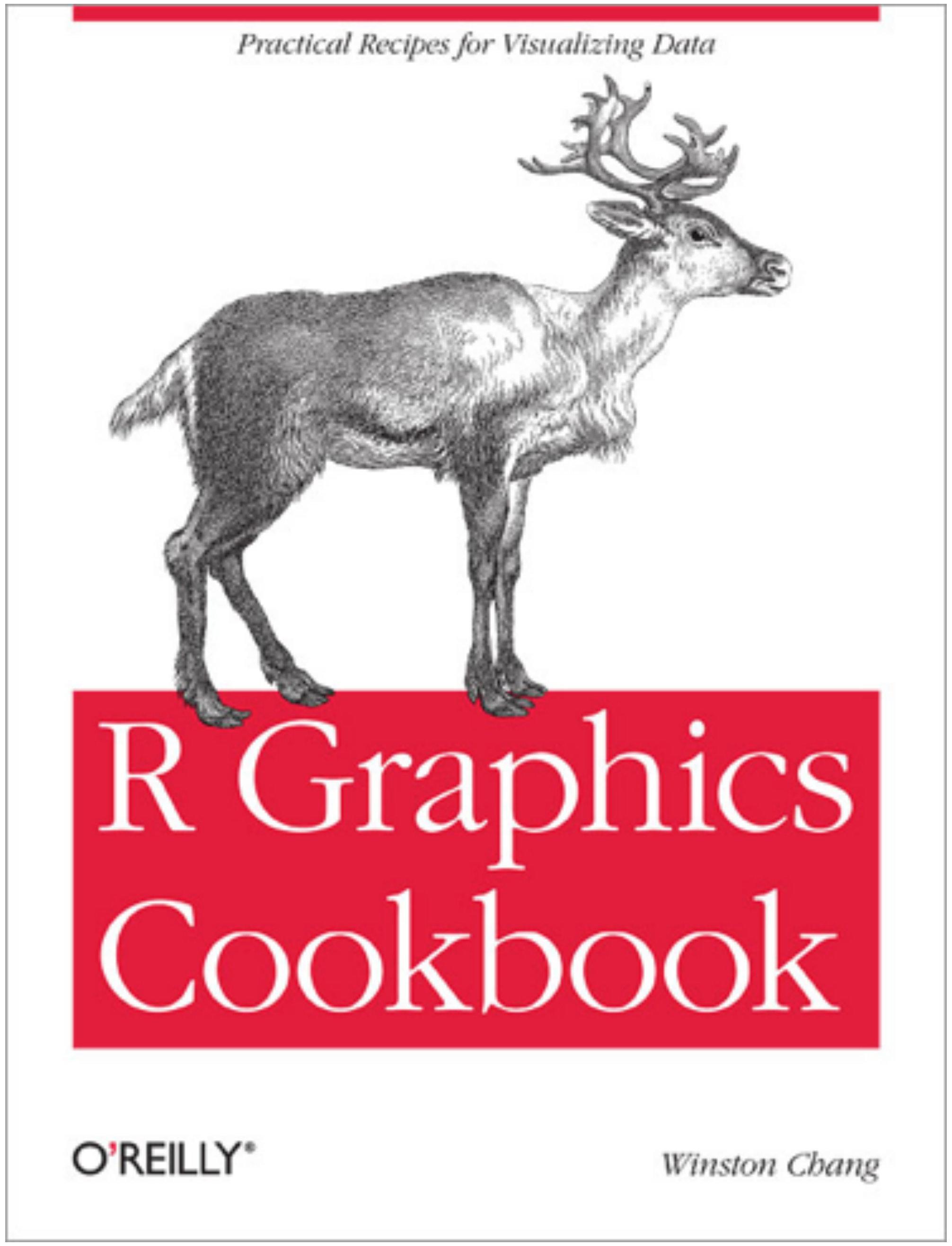
`tx + theme_bw()`

themes

The `ggthemes` package offers other pre-built themes.

To learn how to change individual elements of a theme, I recommend the R Graphics Cookbook by Winston Chang

http://shop.oreilly.com/product/0636920023135.do#



Coordinate systems

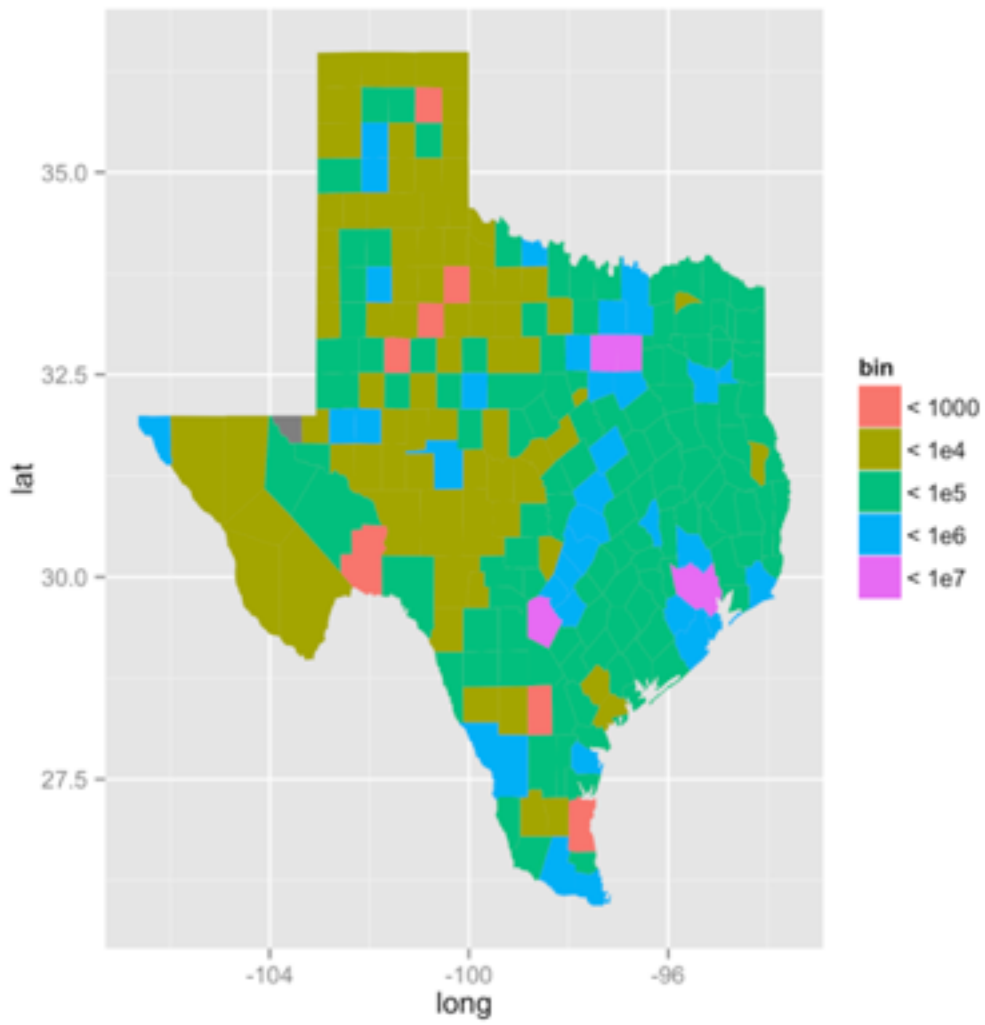
coordinate systems

ggplot2 comes with a few different coordinate systems, but you'll almost always use cartesian coordinates

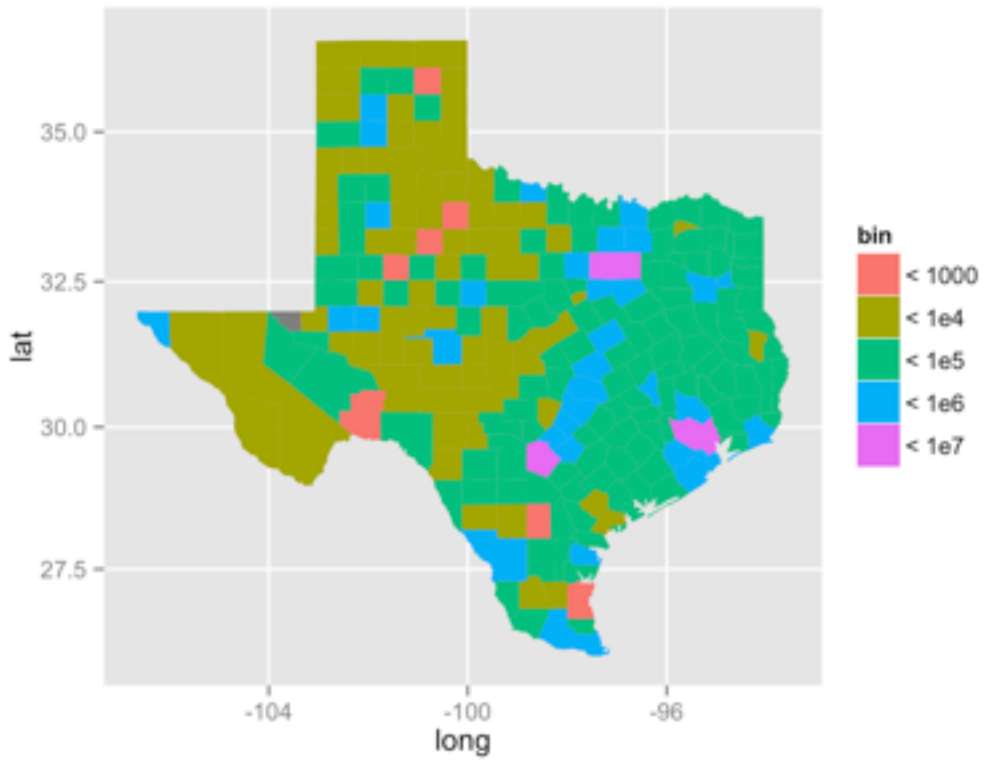
function	system
`coord_cartesian()`	Cartesian coordinates
`coord_fixed()`	Cartesian with a fixed aspect ratio
`coord_polar()`	Polar coordinates
`coord_map()`	A map projection

coordinates

Default



`tx + coord_cartesian()`



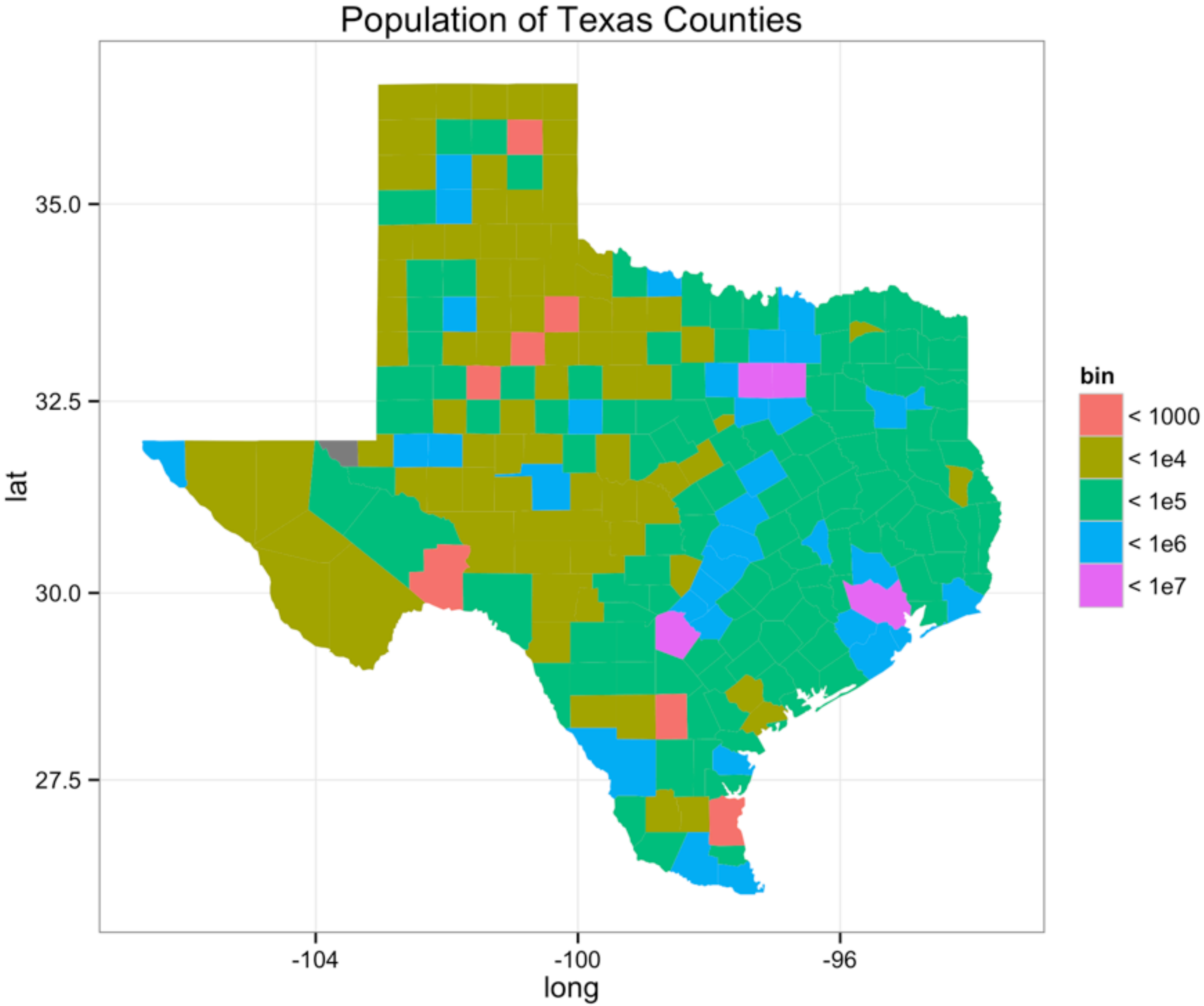
`tx + coord_map("mercator")`

Your Turn

Modify tx to

1. have a title
2. use the black and white theme
3. use a mercator map projection

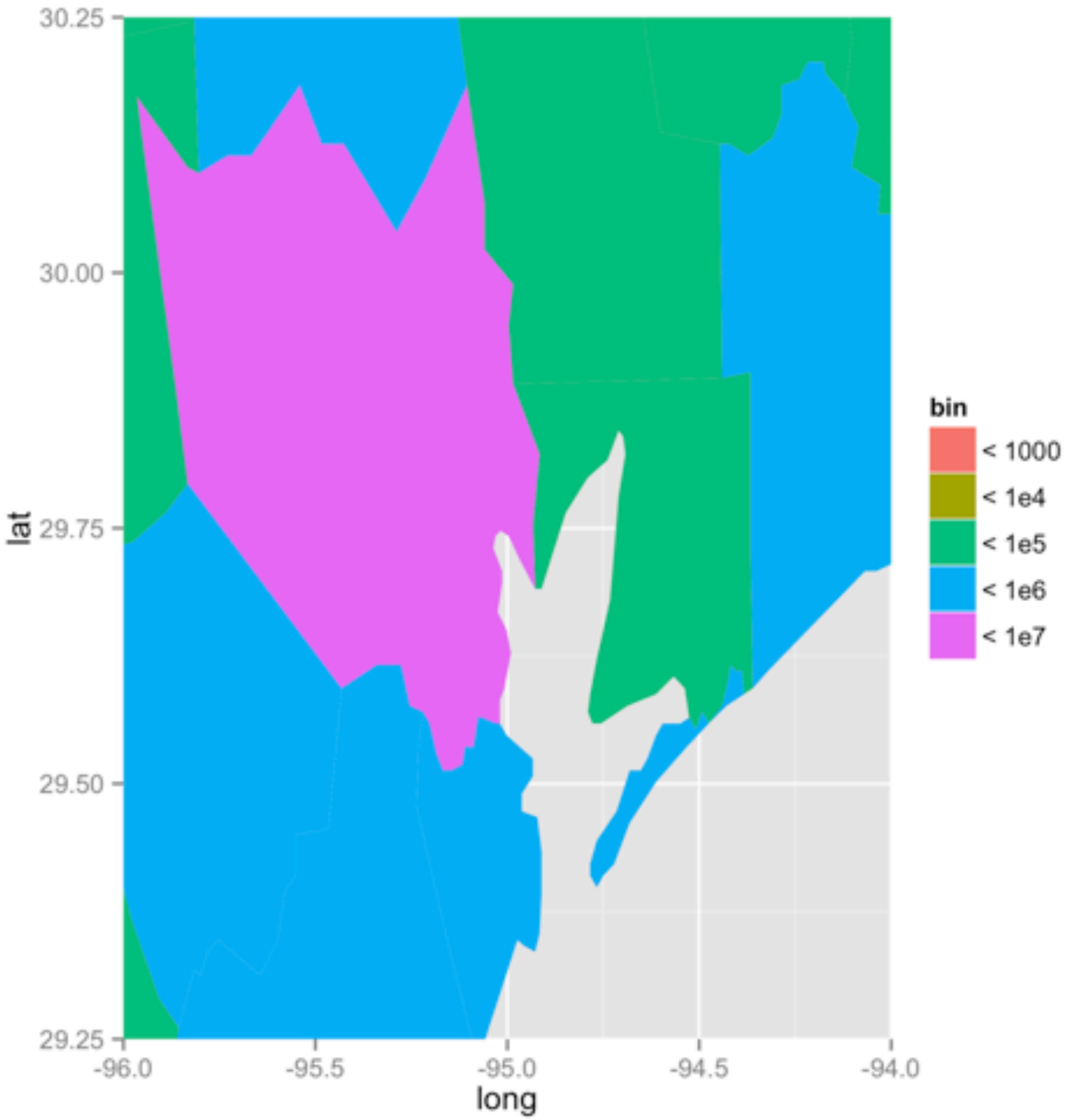
```
tx +
  ggtitle(”Population of Texas Counties”) +
  theme_bw() +
  coord_map(”mercator”)
```

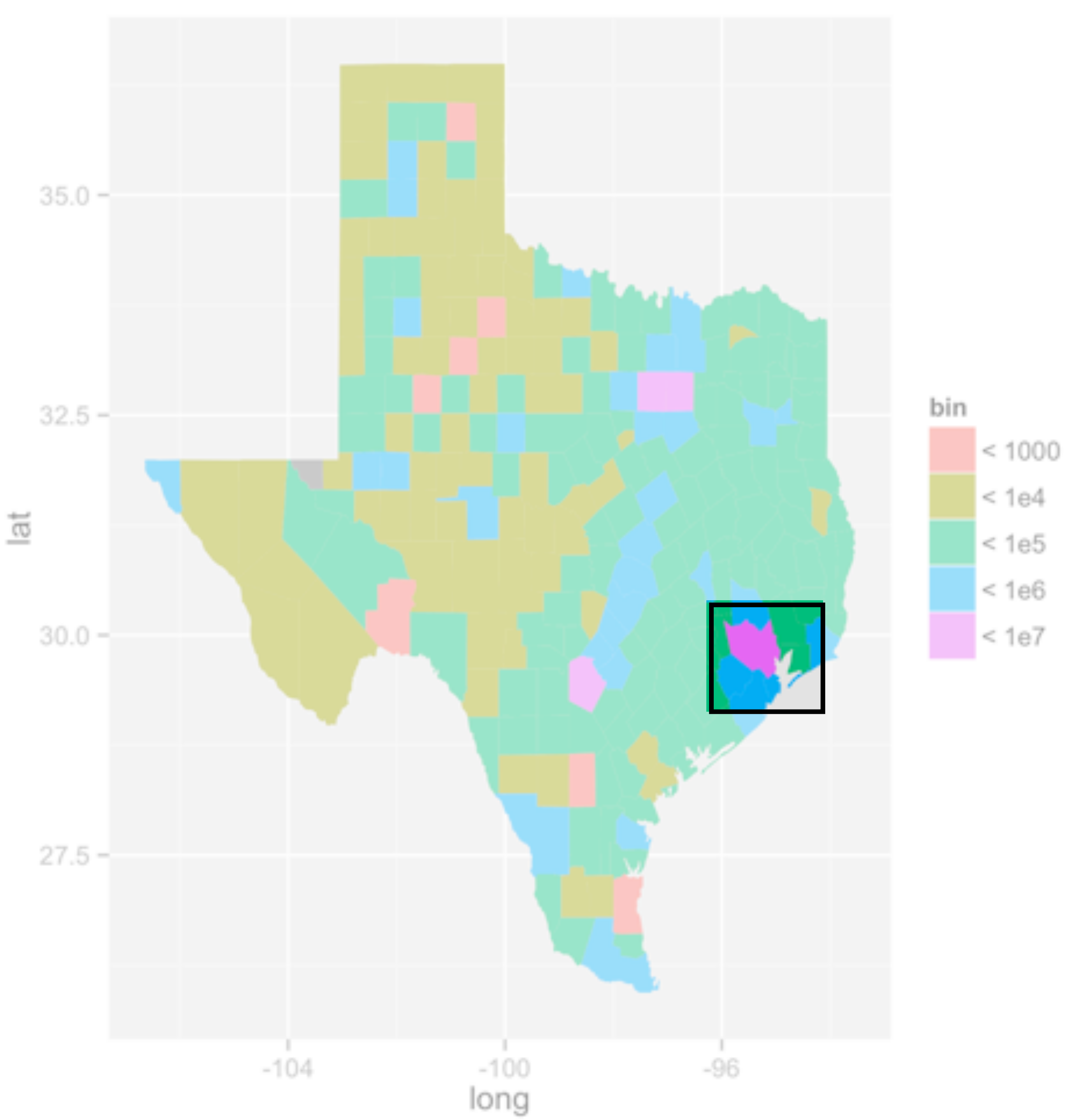

Population of Texas Counties
lat
35.0
32.5
30.0
27.5
long
-104
-100
-96
bin
< 1000
< 1e4
< 1e5
< 1e6
< 1e7

Aside: zooming

The coordinate system provides a convenient way to zoom

```
# Zoomed in on Houston, Texas
tx + coord_cartesian(xlim = c(-96, -94),
  ylim = c(29.25, 30.25))
```


30.25
30.00
29.75
29.50
29.25
lat
-96.0
-95.5
-95.0
-94.5
-94.0
long
bin
< 1000
< 1e4
< 1e5
< 1e6
< 1e7


35.0
32.5
30.0
27.5
lat
-104
-100
-96
long
bin
< 1000
< 1e4
< 1e5
< 1e6
< 1e7

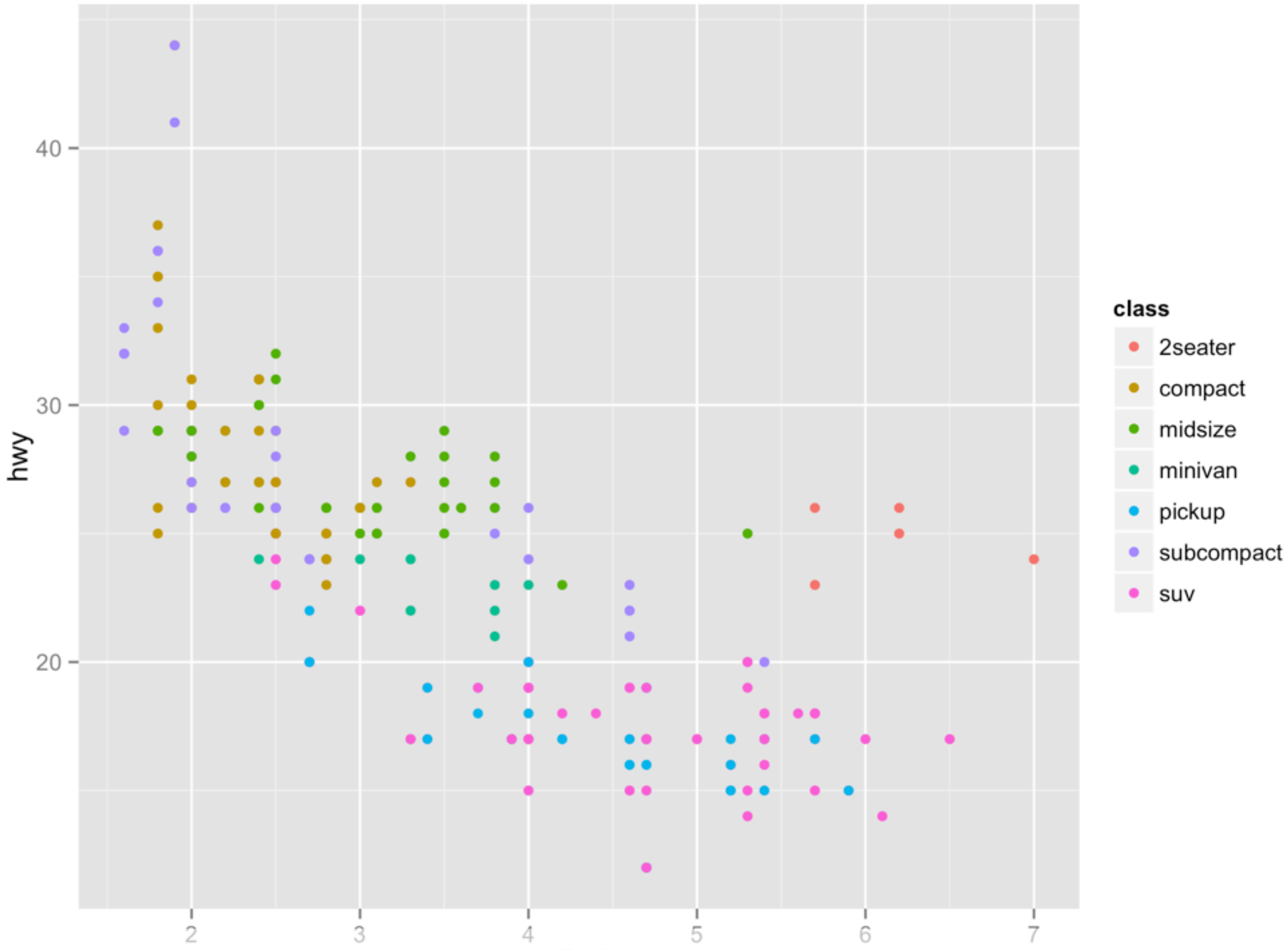
Scales

Aesthetic mapping

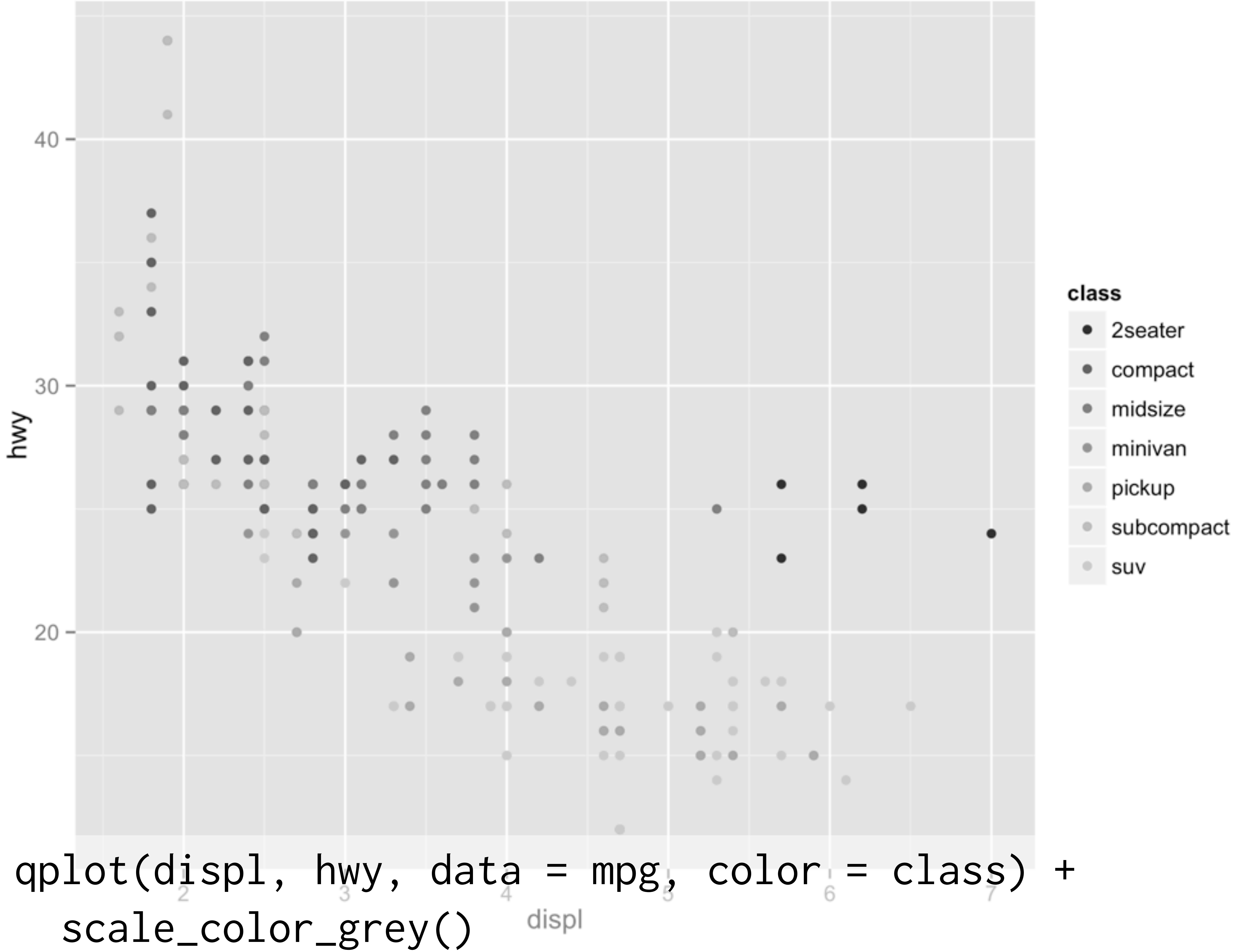
What variable to map to color

Scale

How to map the variable to color



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



```
qplot(displ, hwy, data = mpg, color = class) +
  scale_color_grey()
```

Scales

The details of an aesthetic mapping.

Naming scheme: scale_*aesthetic*_*scalename*

```
qplot(displ, hwy, data = mpg, color = class) +
  scale_color_grey()
```

Scales

The details of an aesthetic mapping.

Naming scheme: scale_*aesthetic*_*scalename*

```
tx + scale_fill_grey()
```

Scales

The details of an aesthetic mapping.

Naming scheme: scale_*aesthetic*_*scalename*

```
tx + scale_fill_grey()
```

Scales

The details of an aesthetic mapping.

Naming scheme: scale_*aesthetic*_*scalename*

```
tx + scale_fill_grey()
```

Scales

The details of an aesthetic mapping.

Naming scheme: scale_*aesthetic*_*scalename*

```
tx + scale_fill_grey()
```

Defaults

By default, a scale is built for every aesthetic that your plot maps to data.

When you add a new scale, you override the default.

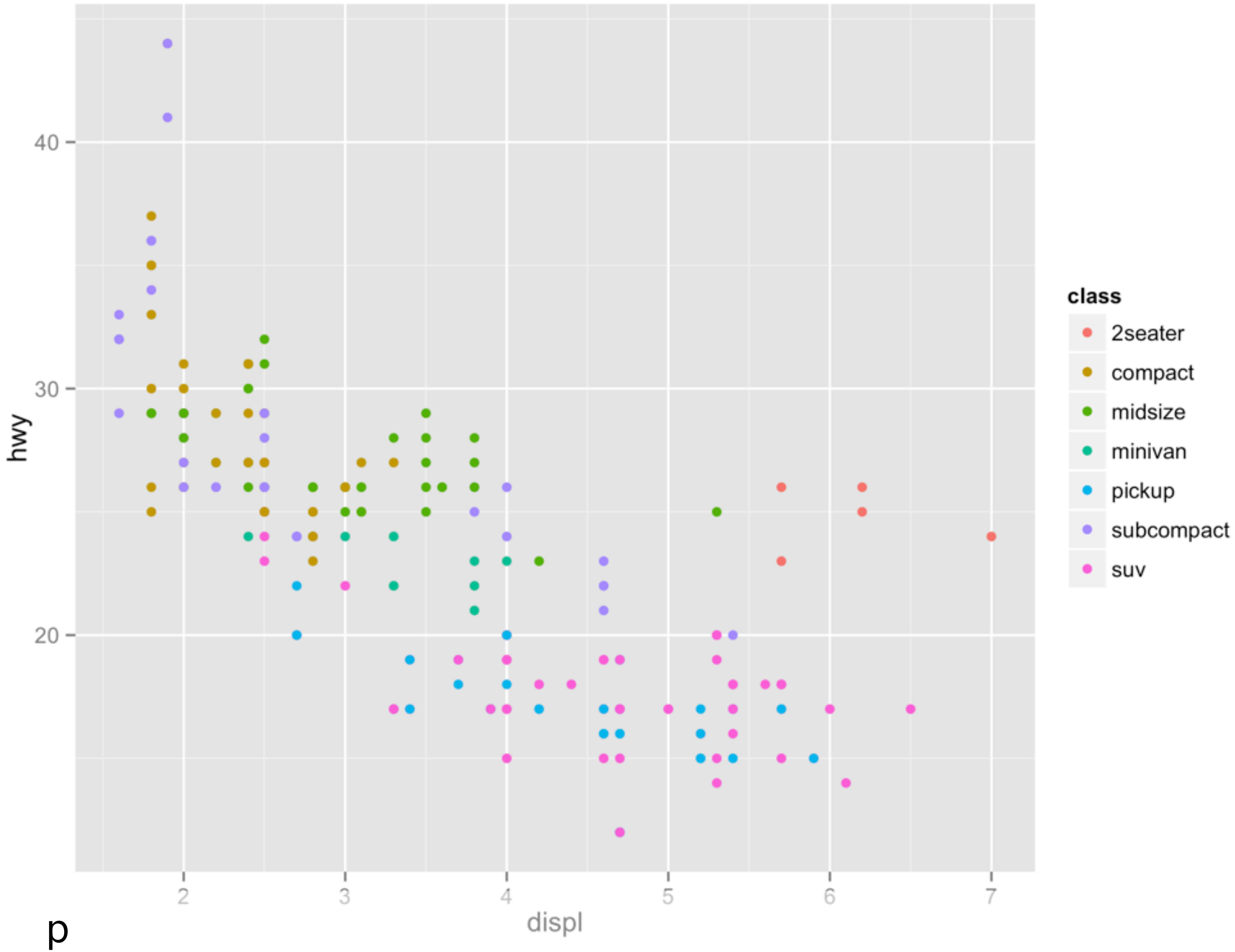
Most common scales to add:

- scale_*aesthetic*_**continuous**, or
- scale_*aesthetic*_**discrete**

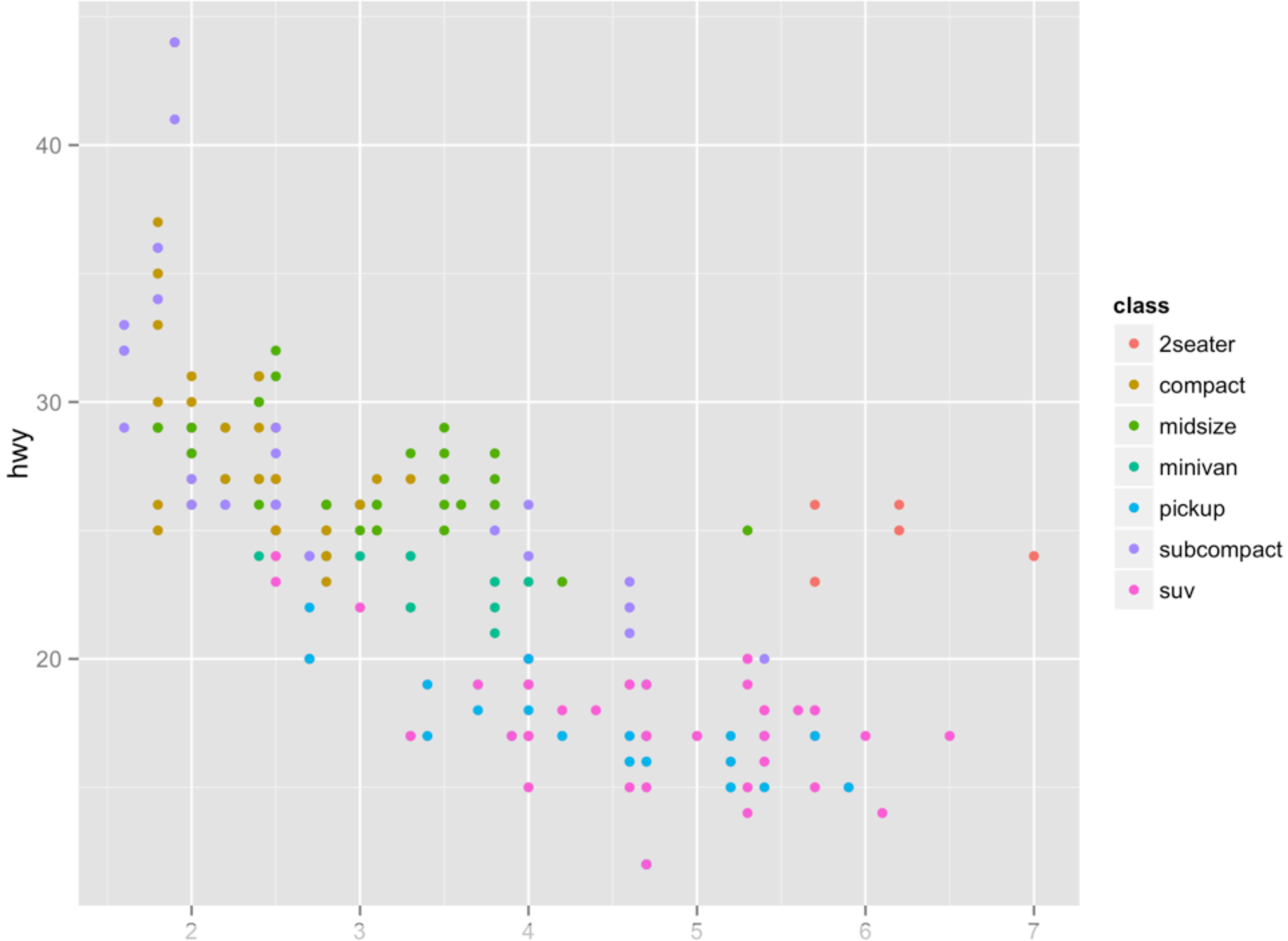
What can you change with a scale?

scale parameter	controls
name	axis labels (x, y) or legend title
breaks	where ticks occur (x, y) or legend entries
labels	tick labels (x, y) or legend labels
limits	The range of data to apply the mapping to

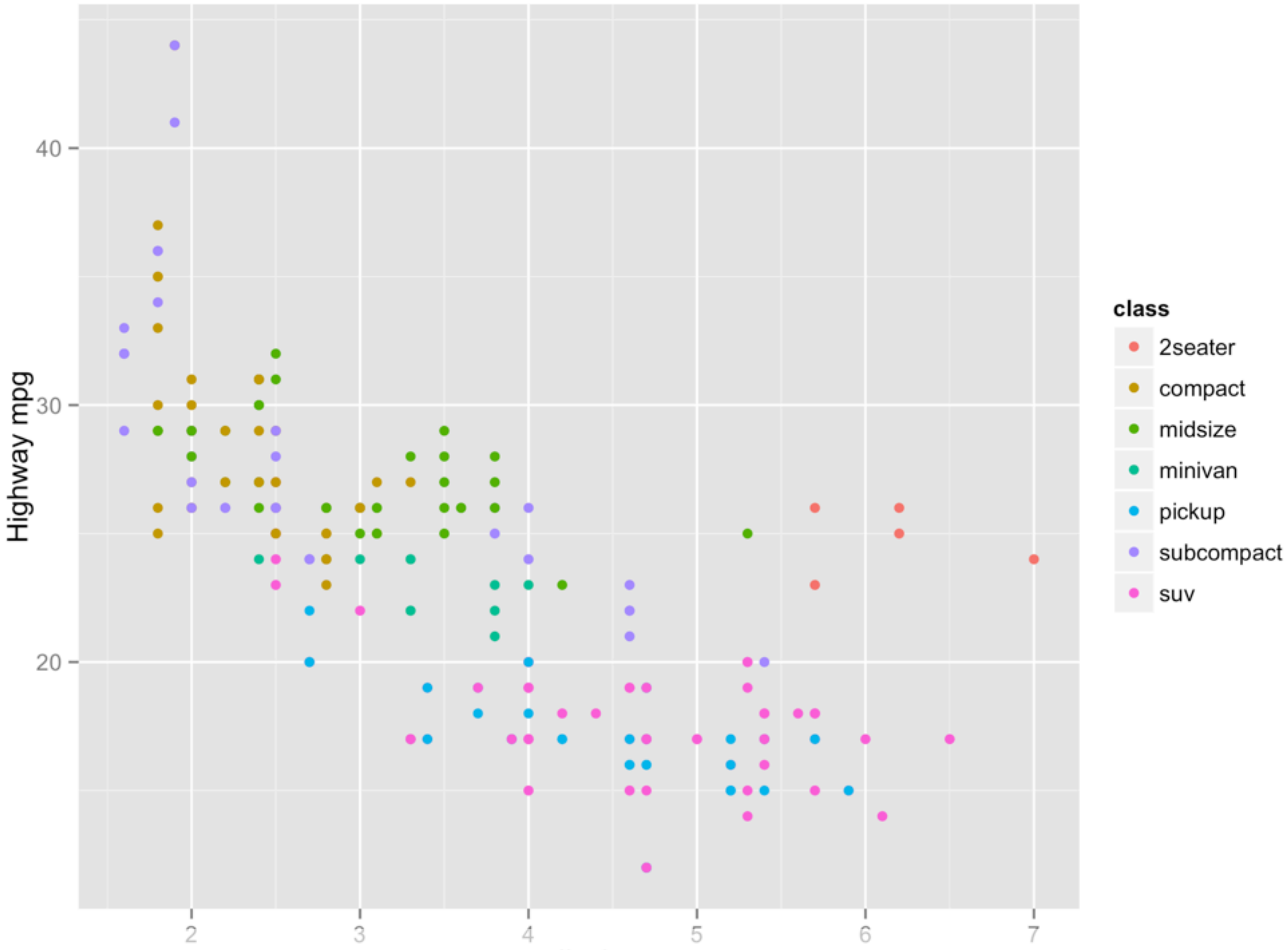
```
p <- qplot(displ, hwy, data = mpg, color = class)
```


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

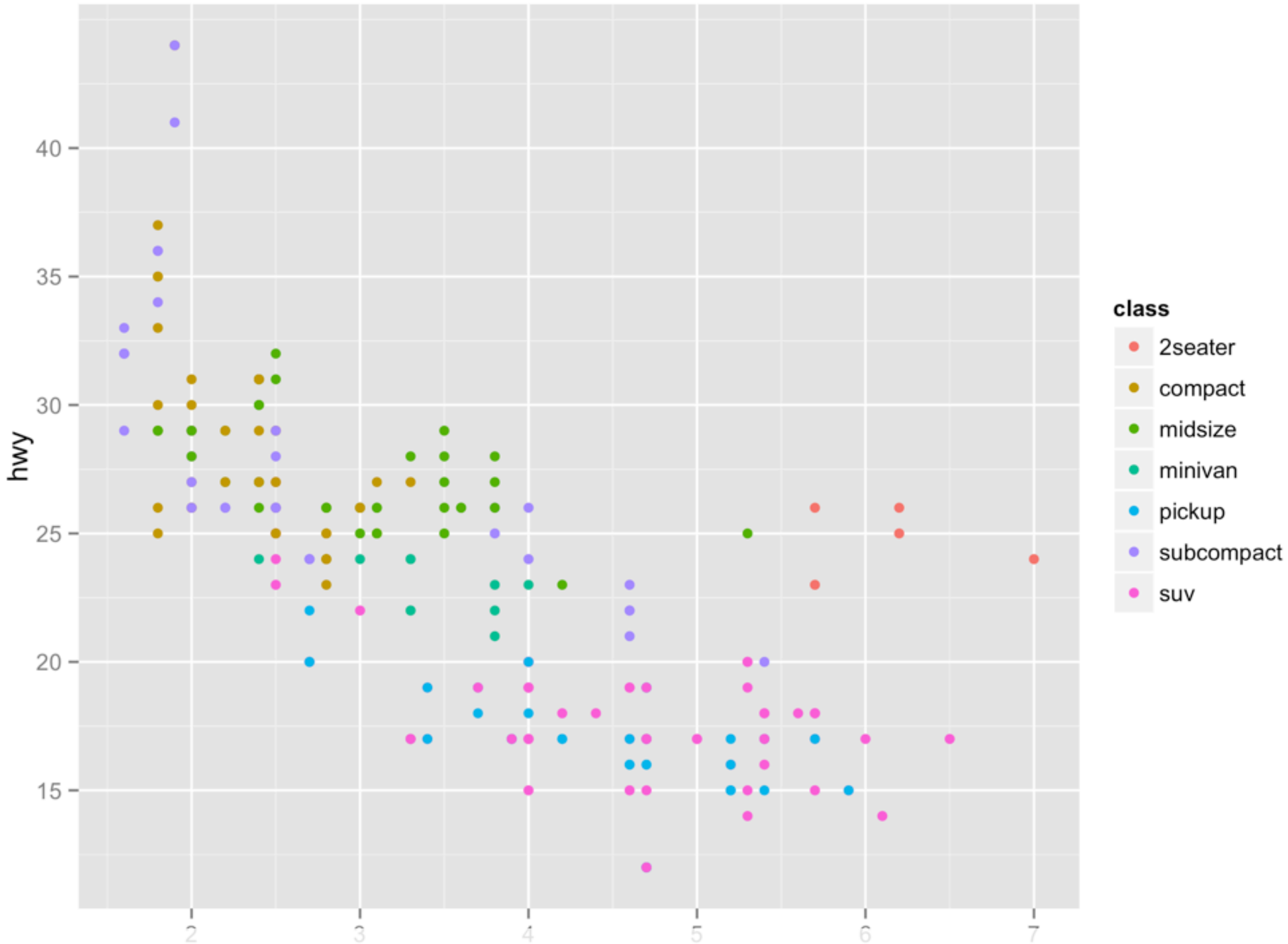
p



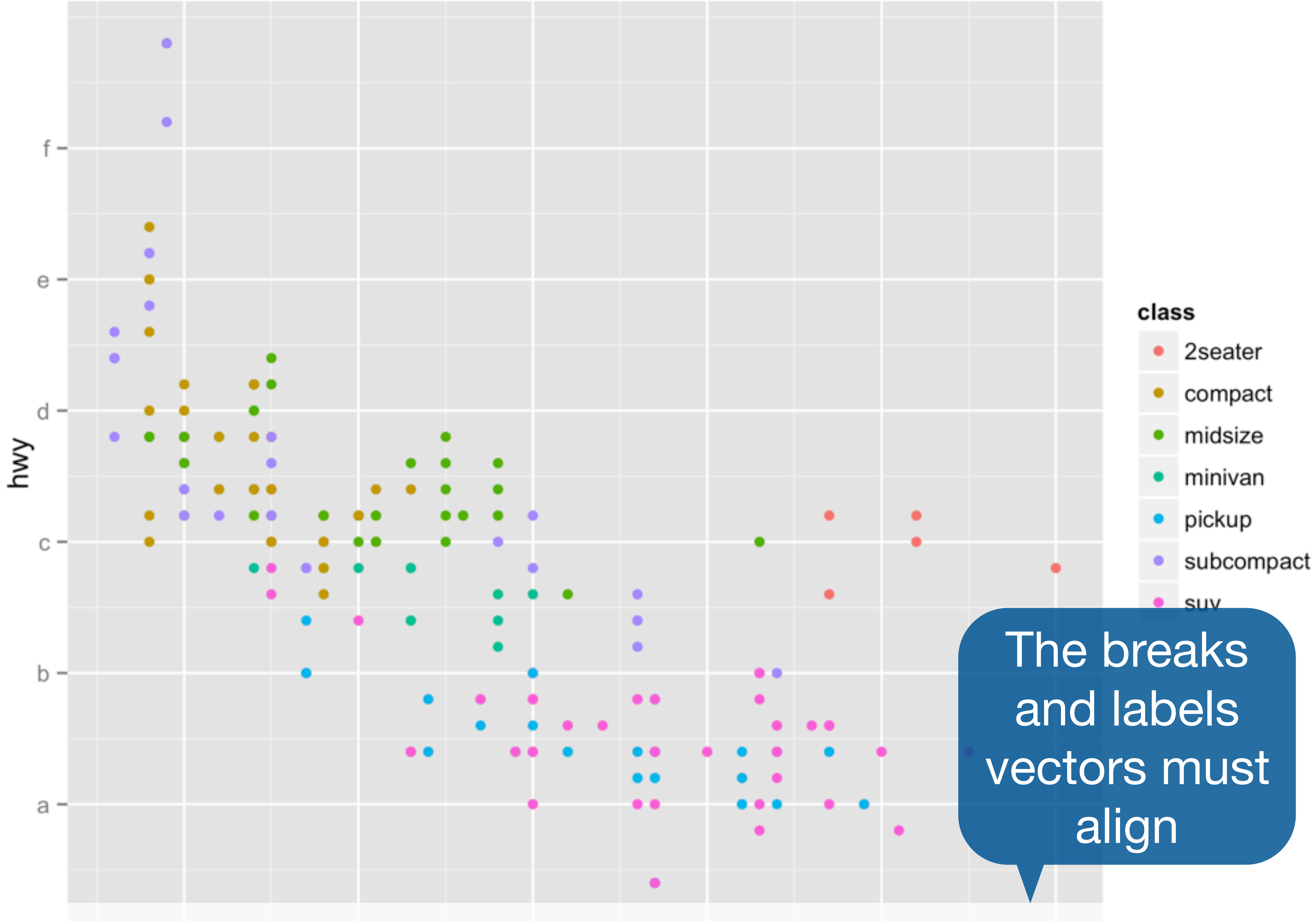
```
p + scale_y_continuous()
```



```
p + scale_y_continuous(name = "Highway mpg")
```

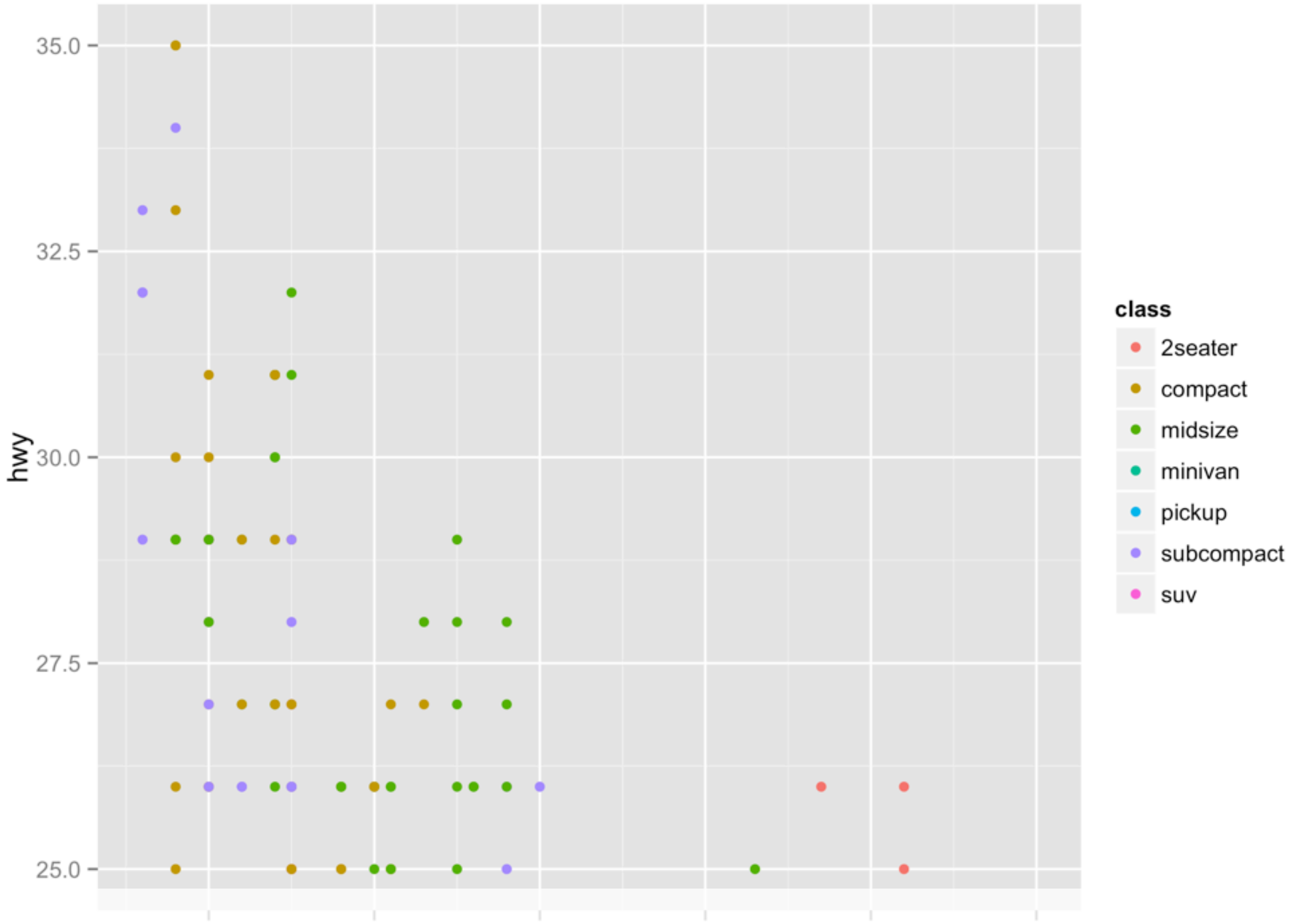


```
p + scale_y_continuous(breaks = c(15, 20, 25, 30, 35, 40))
```



The breaks and labels vectors must align

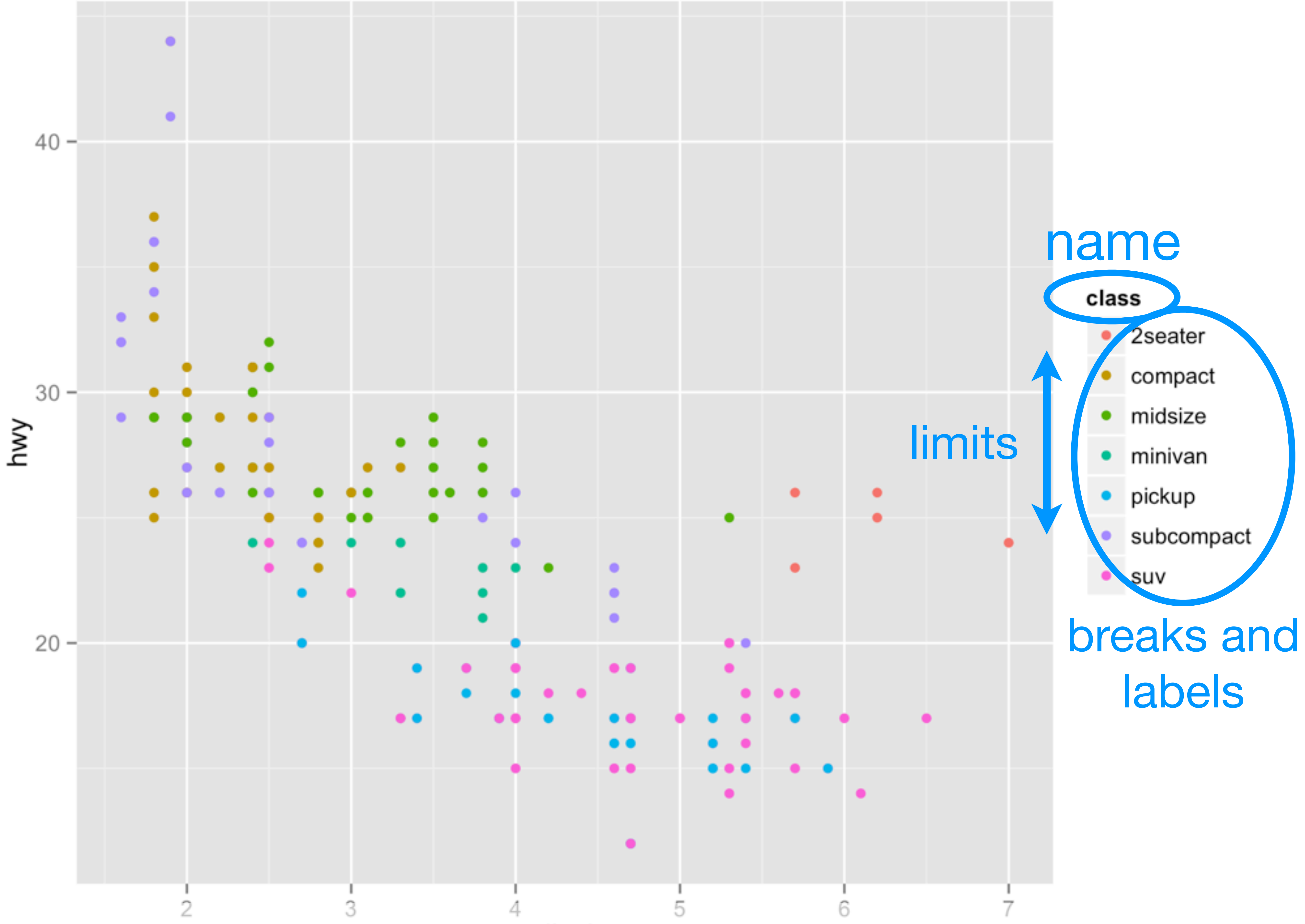
```
p + scale_y_continuous(breaks = c(15, 20, 25, 30, 35, 40),
  labels = c("a", "b", "c", "d", "e", "f"))
```



```
p + scale_y_continuous(limits = c(30, 35))
```

Your turn

What happens if you add a scale for color (hint: `scale_color_discrete`) and change the name, breaks, labels, and limits parameters?



name

limits

breaks and labels

```
p + scale_color_discrete()
```

Your turn

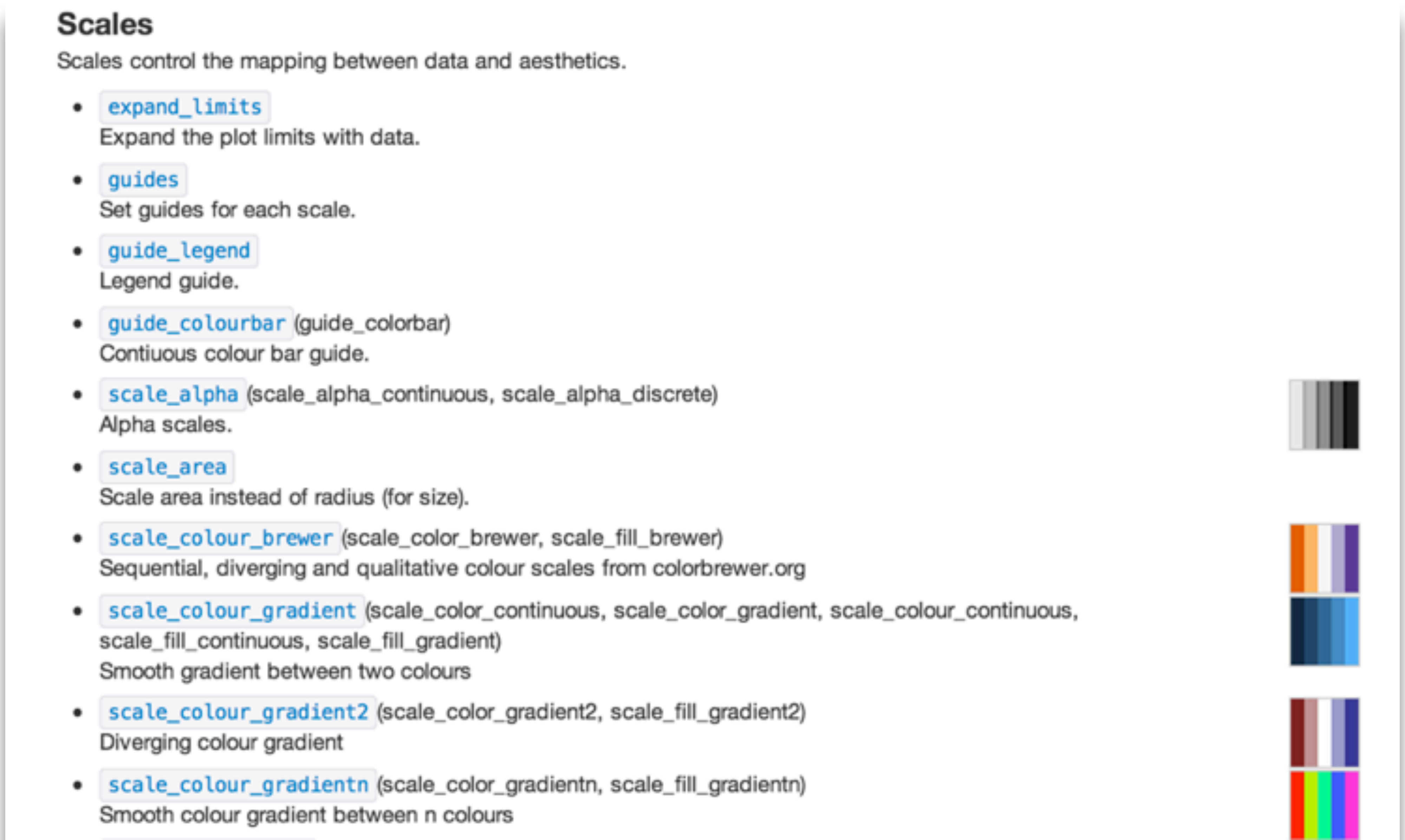
Use scales with `tx` to

1. remove the `long` and `lat` axis labels
2. change the title of the fill legend
3. create more informative legend labels

```
tx + scale_fill_discrete(“Population”,
  labels = c(“0 - 999”, “1,000 - 9,999”,
    “10,000 - 99,999”, “100,000 - 999,999”,
    “1,000,000+”)) +
  scale_x_continuous(“”) +
  scale_y_continuous(“”)
```

More exotic scales

A useful list of scales is available at http://docs.ggplot2.org/current



Scales

Scales control the mapping between data and aesthetics.

- `expand_limits`
  Expand the plot limits with data.
- `guides`
  Set guides for each scale.
- `guide_legend`
  Legend guide.
- `guide_colourbar` (guide_colorbar)
  Contiuous colour bar guide.
- `scale_alpha` (scale_alpha_continuous, scale_alpha_discrete)
  Alpha scales.
- `scale_area`
  Scale area instead of radius (for size).
- `scale_colour_brewer` (scale_color_brewer, scale_fill_brewer)
  Sequential, diverging and qualitative colour scales from colorbrewer.org
- `scale_colour_gradient` (scale_color_continuous, scale_color_gradient, scale_colour_continuous, scale_fill_continuous, scale_fill_gradient)
  Smooth gradient between two colours
- `scale_colour_gradient2` (scale_color_gradient2, scale_fill_gradient2)
  Diverging colour gradient
- `scale_colour_gradientn` (scale_color_gradientn, scale_fill_gradientn)
  Smooth colour gradient between n colours

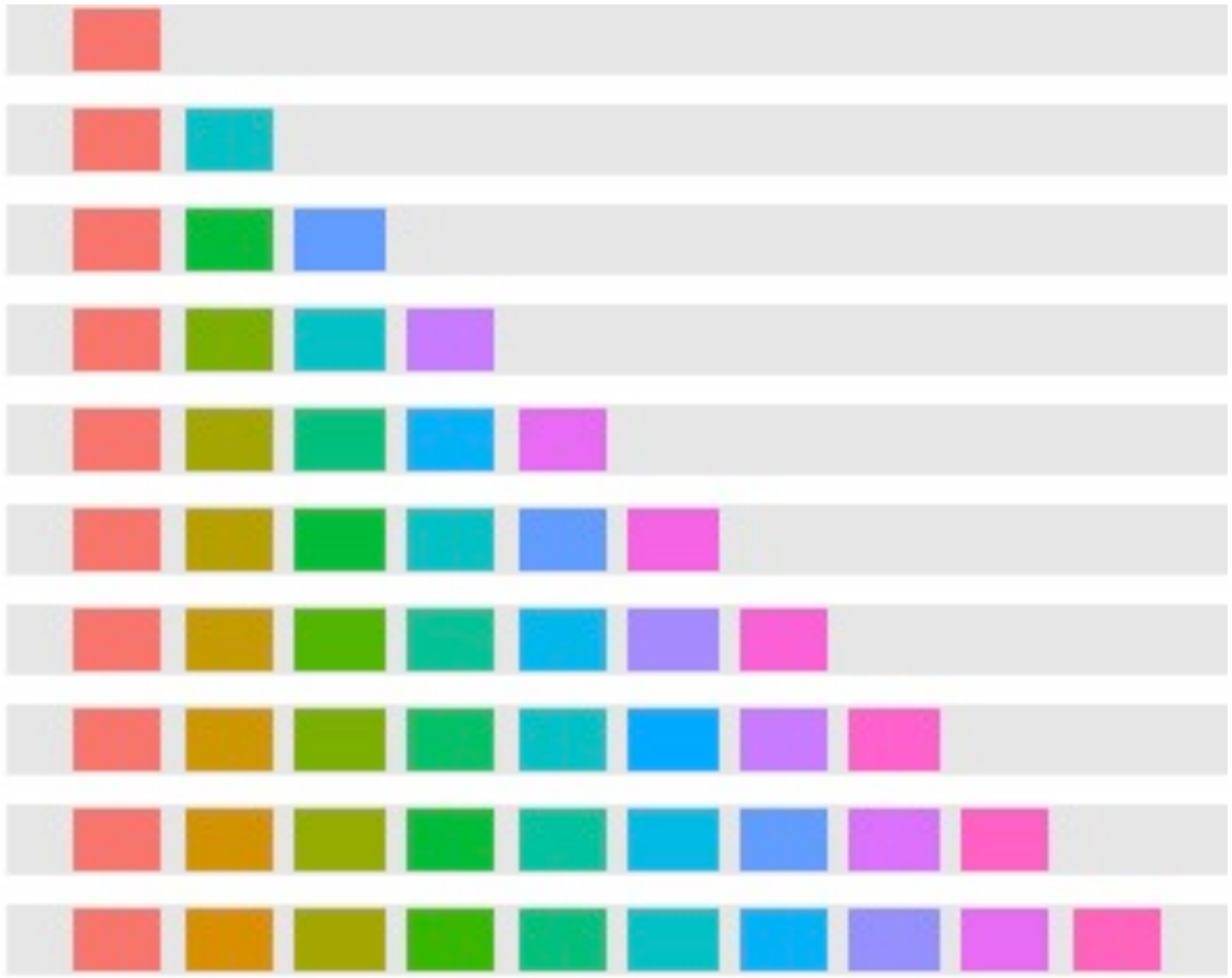
Colors

Colour is the most popular aesthetic after position. It is also the easiest to misuse.

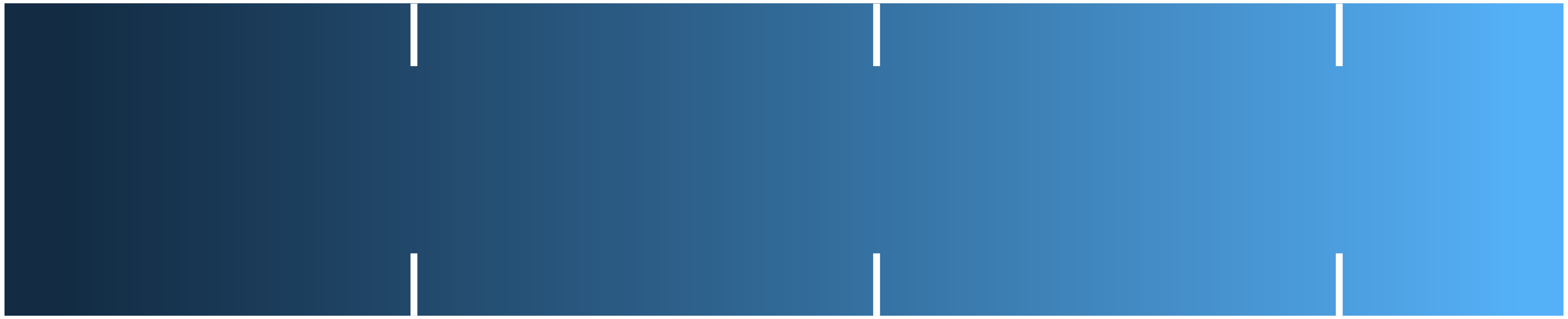
color **spaces**

color **blindness**.

Default discrete palettes



Default continuous palette



Custom color scales

Discrete:

```
scale_fill_manual
scale_fill_brewer
scale_fill_grey
```

Continuous:

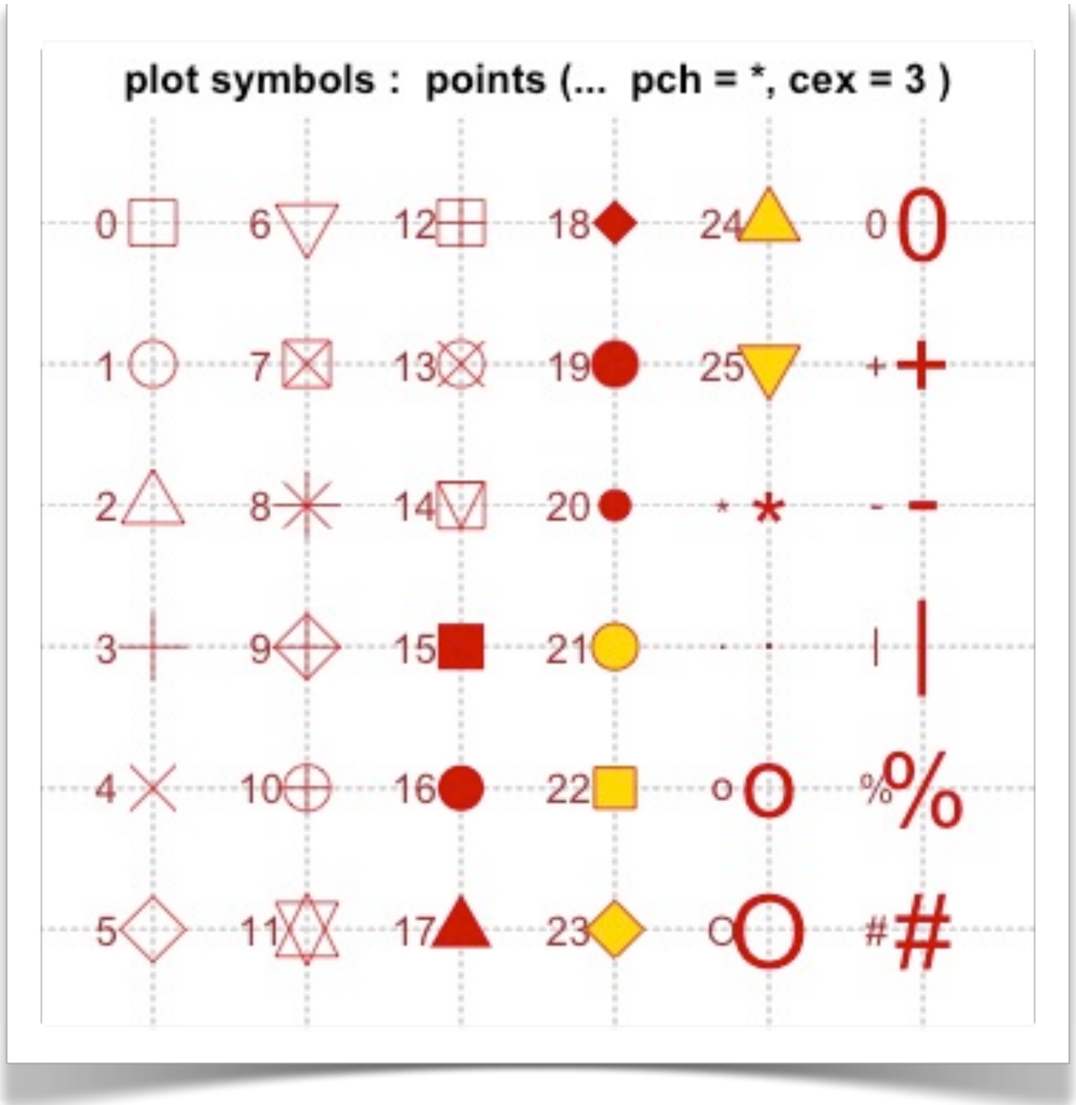
```
scale_fill_gradient
scale_fill_gradient2
scale_fill_gradientn
```

Manual Discrete Scales

```
scale_color_manual
scale_fill_manual
scale_size_manual
scale_shape_manual
```

new parameter	controls
values	values to use in the scale (specific colors, sizes, etc.)

```
r <- qplot(displ, cty, colour = drv, shape = fl,
   data = mpg)

# Specify colors manually
r + scale_color_manual(values = c("red", "black",
   "#3333cc"))

# Specify the shapes manually
r + scale_shape_manual(
   values = c(0, 15, 1, 16, 3))

?pch
```



Manual Scales

Offer complete control

Often look worse than you'd imagine

Brewer Discrete Scales

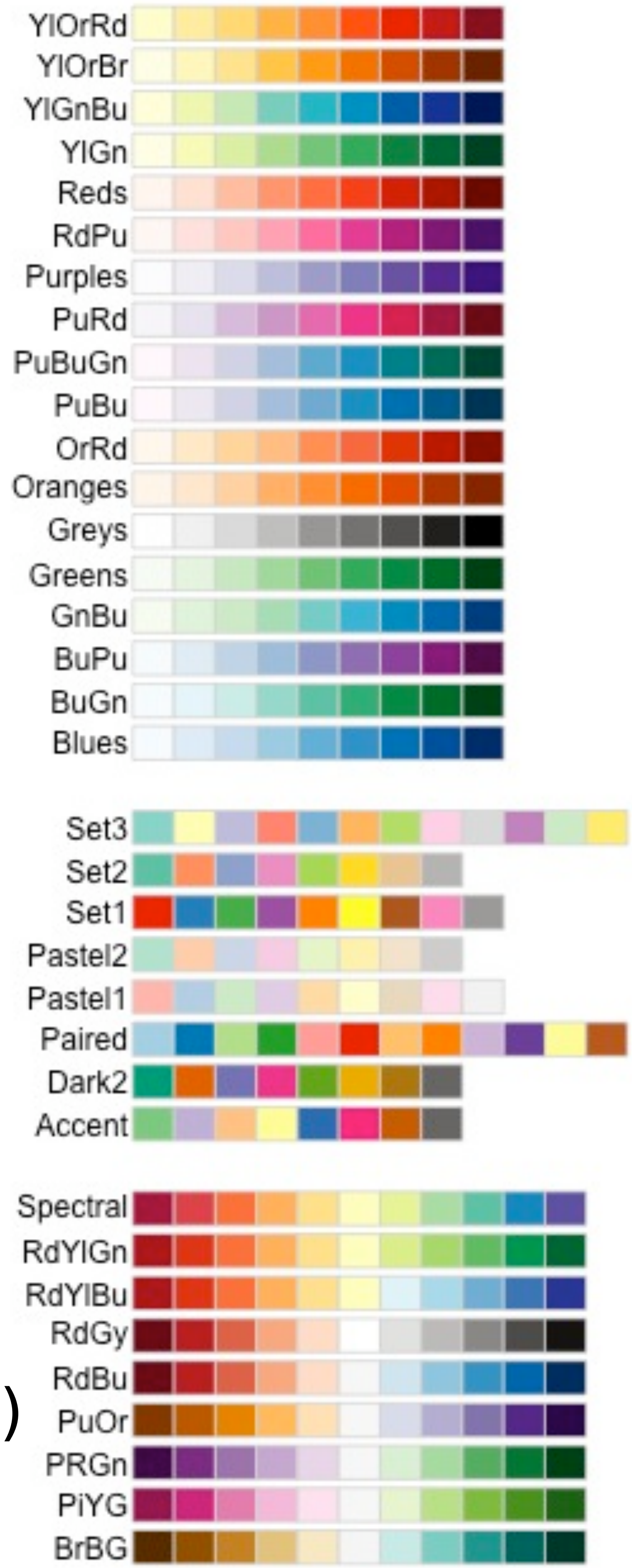
```
scale_color_brewer
scale_fill_brewer
```

new parameter	controls
palette	name of a palette in the RColorBrewer package

Color brewer

Cynthia Brewer developed useful, pleasing palettes, particularly tailored for maps: http://colorbrewer2.org

```
library(RColorBrewer)
RColorBrewer::display.brewer.all()

q + scale_color_brewer(palette="Spectral")
q + scale_color_brewer(palette="Set3")

library(scales)
show_col(brewer_pal(palette = "YlOrRd")(9))
```



Your turn

Modify the `tx` fill scale to use a palette that better maps `population`.

Use `RColorBrewer::display.brewer.all()` if you'd like to see possible brewer palletes.

```
tx + scale_fill_brewer("Population",
  palette = "Blues",
  labels = c("0 - 999", "1,000 - 9,999",
    "10,000 - 99,999", "100,000 - 999,999",
    "1,000,000+")) +
  scale_x_continuous("") +
  scale_y_continuous("")
```

Saving graphics

Your turn

What does this command return?

```
getwd()
```

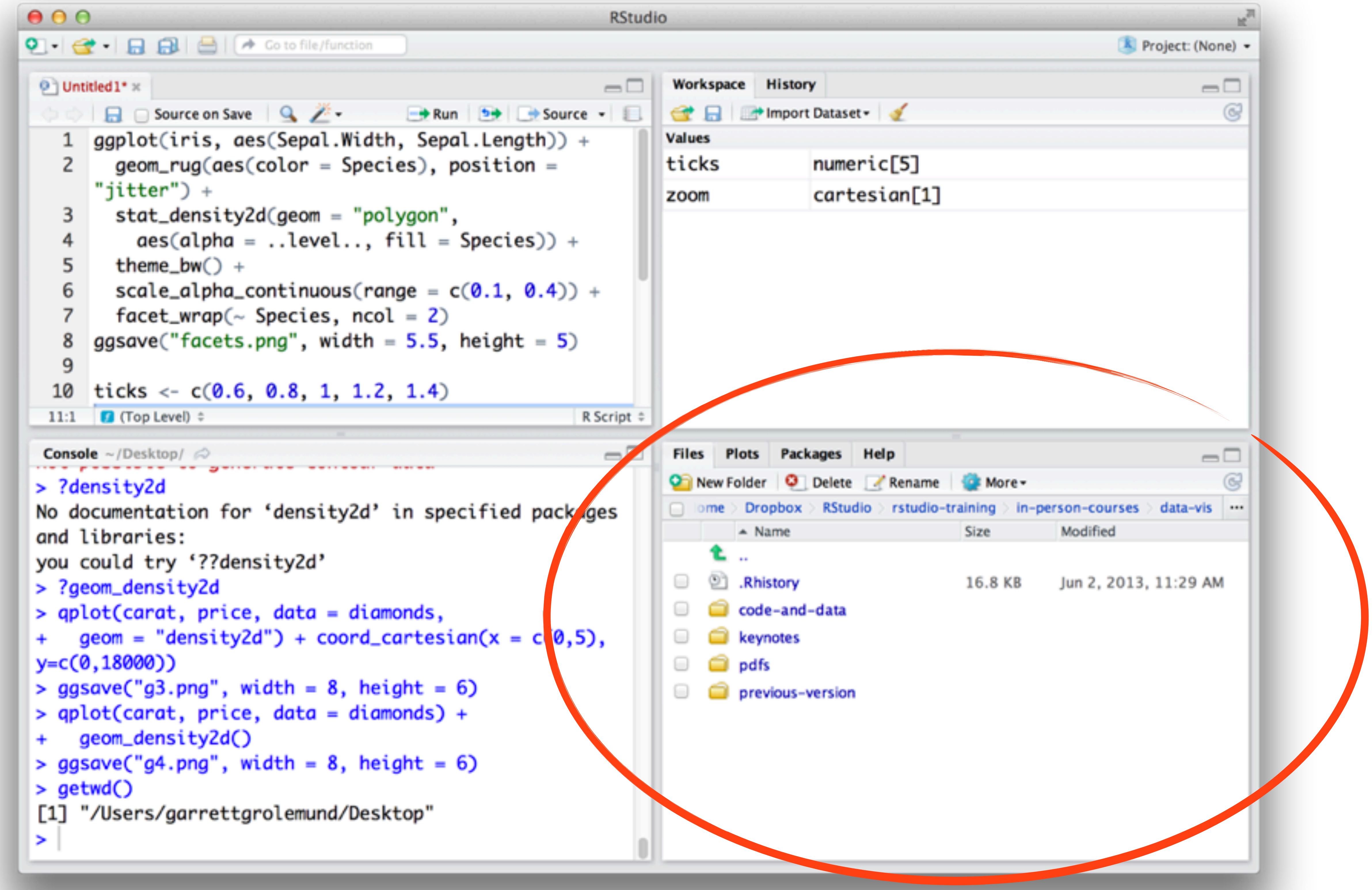
Could you find that file
on your computer?

Working directory

When you start R, it associates itself with a folder (i.e, directory) on your computer.

- This folder is known as your "working directory"
- When you save files, R will save them here
- When you load files, R will look for them here

The files pane of RStudio displays the contents of your working directory



```r
ggplot(iris, aes(Sepal.Width, Sepal.Length)) +
  geom_rug(aes(color = Species), position = "jitter") +
  stat_density2d(geom = "polygon",
    aes(alpha = ..level.., fill = Species)) +
  theme_bw() +
  scale_alpha_continuous(range = c(0.1, 0.4)) +
  facet_wrap(~ Species, ncol = 2)
ggsave("facets.png", width = 5.5, height = 5)

ticks <- c(0.6, 0.8, 1, 1.2, 1.4)
```

Values

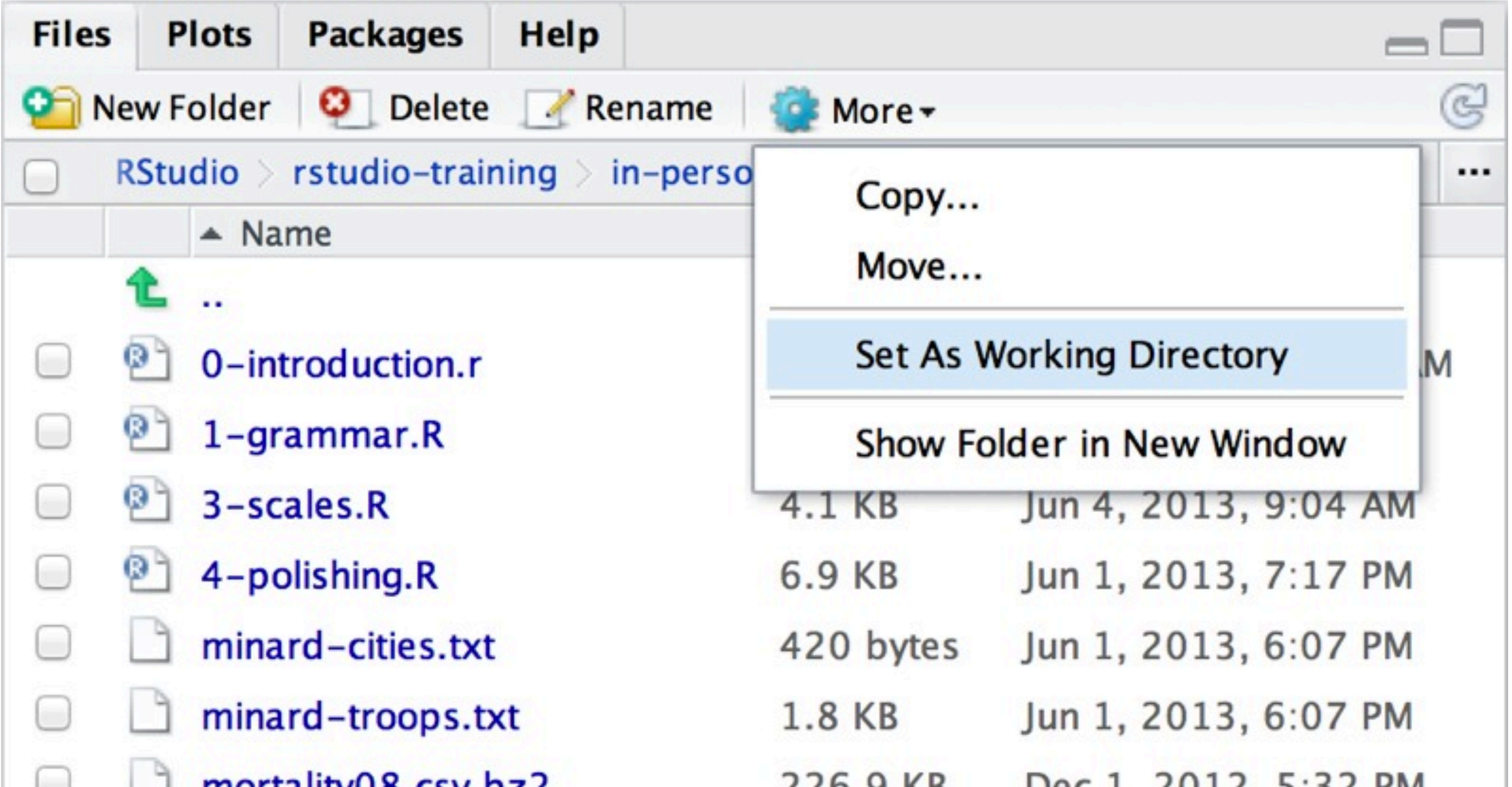
ticks	numeric[5]
zoom	cartesian[1]

```
> ?density2d
No documentation for ‘density2d’ in specified packages
and libraries:
you could try ‘??density2d’
> ?geom_density2d
> qplot(carat, price, data = diamonds,
+   geom = "density2d") + coord_cartesian(x = c(0,5),
y=c(0,18000))
> ggsave("g3.png", width = 8, height = 6)
> qplot(carat, price, data = diamonds) +
+   geom_density2d()
> ggsave("g4.png", width = 8, height = 6)
> getwd()
[1] "/Users/garrettgrolemund/Desktop"
>
```

▲ Name	Size	Modified
..		
☐ .Rhistory	16.8 KB	Jun 2, 2013, 11:29 AM
☐ code-and-data		
☐ keynotes		
☐ pdfs		
☐ previous-version		

Changing the Working directory

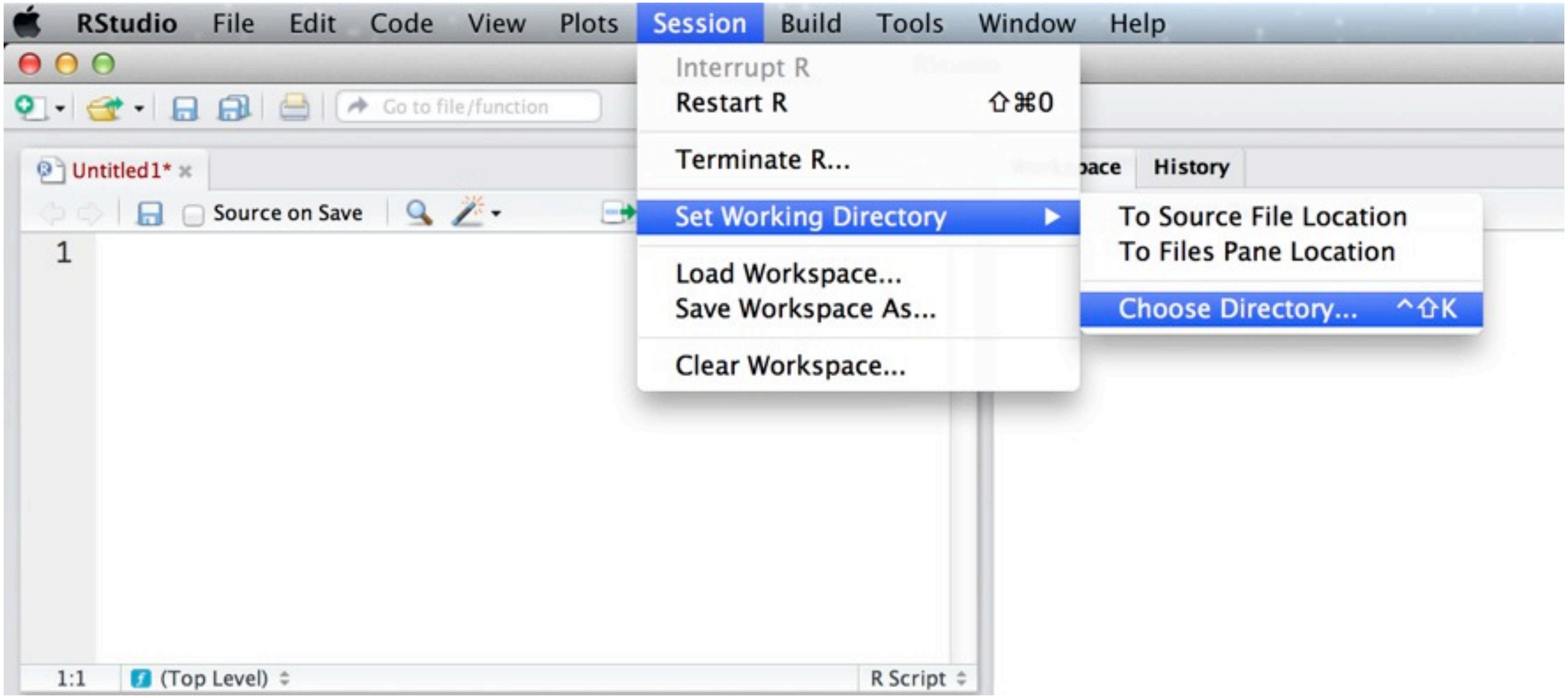
First option: Navigate in the files pane to a new directory. Click More>Set As Working Directory



Name		
0-introduction.r		M
1-grammar.R		
3-scales.R	4.1 KB	Jun 4, 2013, 9:04 AM
4-polishing.R	6.9 KB	Jun 1, 2013, 7:17 PM
minard-cities.txt	420 bytes	Jun 1, 2013, 6:07 PM
minard-troops.txt	1.8 KB	Jun 1, 2013, 6:07 PM

Changing the Working directory

Second option: In the toolbar, go to Session>Set Working Directory>Choose Directory...



Your turn

Change your working directory to the folder you downloaded for today's course.

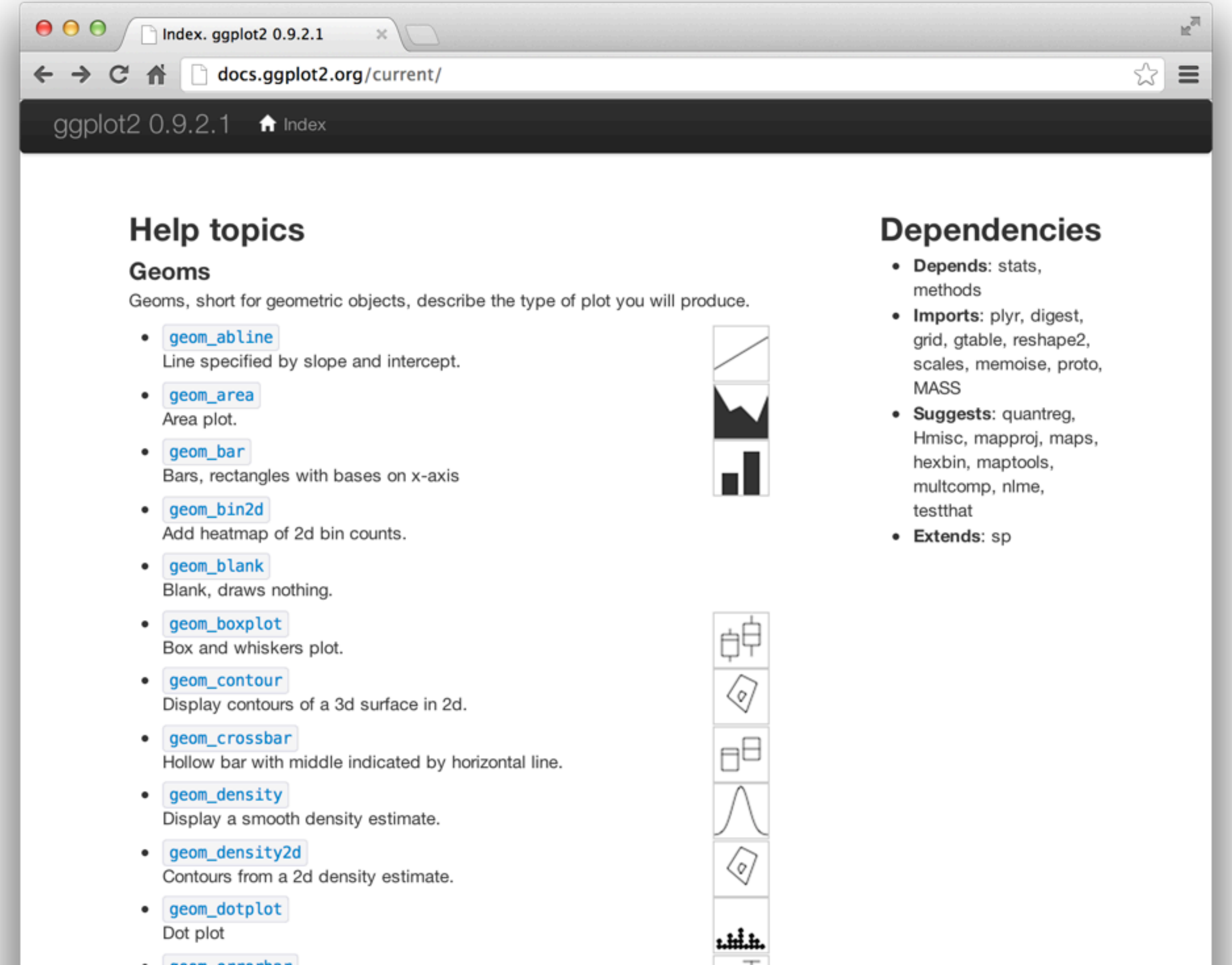
Note: this folder came as a .zip archive. You must extract the .zip file before you can use it as a directory.

Saving plots

```
# Uses size on screen:
ggsave(”my-plot.pdf”)
ggsave(”my-plot.png”)

# Specify size in inches
ggsave(”my-plot.pdf”, width = 6, height = 6)
```

PDF	PNG
Vector based (can zoom in infinitely)	Raster based (made up of pixels)
Good for most plots	Good for plots with thousands of points

Where to go from here



Help topics

Geoms

Geoms, short for geometric objects, describe the type of plot you will produce.

- `geom_abline`
  Line specified by slope and intercept.
- `geom_area`
  Area plot.
- `geom_bar`
  Bars, rectangles with bases on x-axis
- `geom_bin2d`
  Add heatmap of 2d bin counts.
- `geom_blank`
  Blank, draws nothing.
- `geom_boxplot`
  Box and whiskers plot.
- `geom_contour`
  Display contours of a 3d surface in 2d.
- `geom_crossbar`
  Hollow bar with middle indicated by horizontal line.
- `geom_density`
  Display a smooth density estimate.
- `geom_density2d`
  Contours from a 2d density estimate.
- `geom_dotplot`
  Dot plot

Dependencies

- **Depends**: stats, methods
- **Imports**: plyr, digest, grid, gtable, reshape2, scales, memoise, proto, MASS
- **Suggests**: quantreg, Hmisc, mapproj, maps, hexbin, maptools, multcomp, nlme, testthat
- **Extends**: sp

Learning ggplot2

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

stackoverflow
http://stackoverflow.com/tags/ggplot2

Cookbook for common graphics
http://wiki.stdout.org/rcookbook/Graphs/

ggplot2 book
http://amzn.com/0387981403