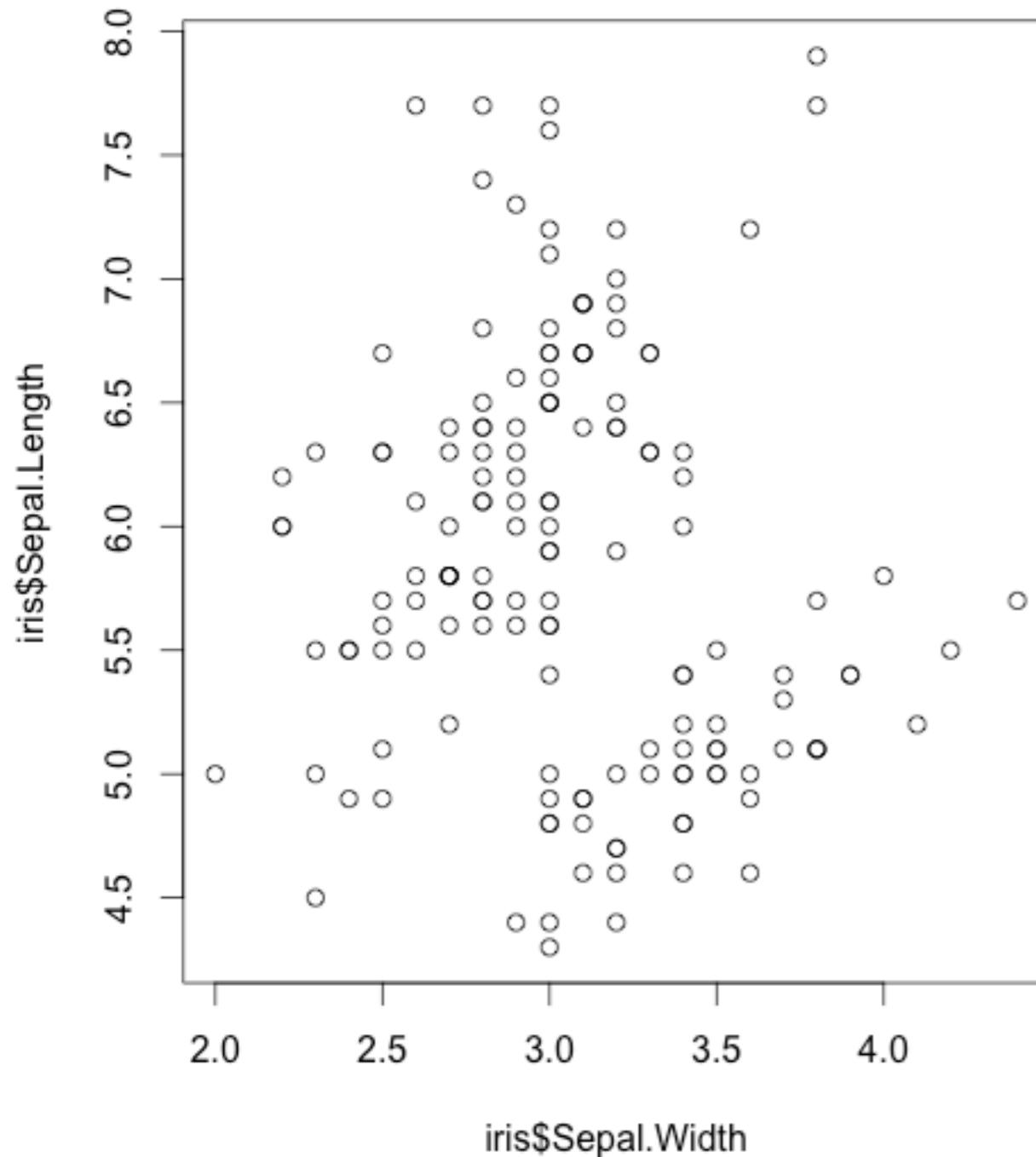


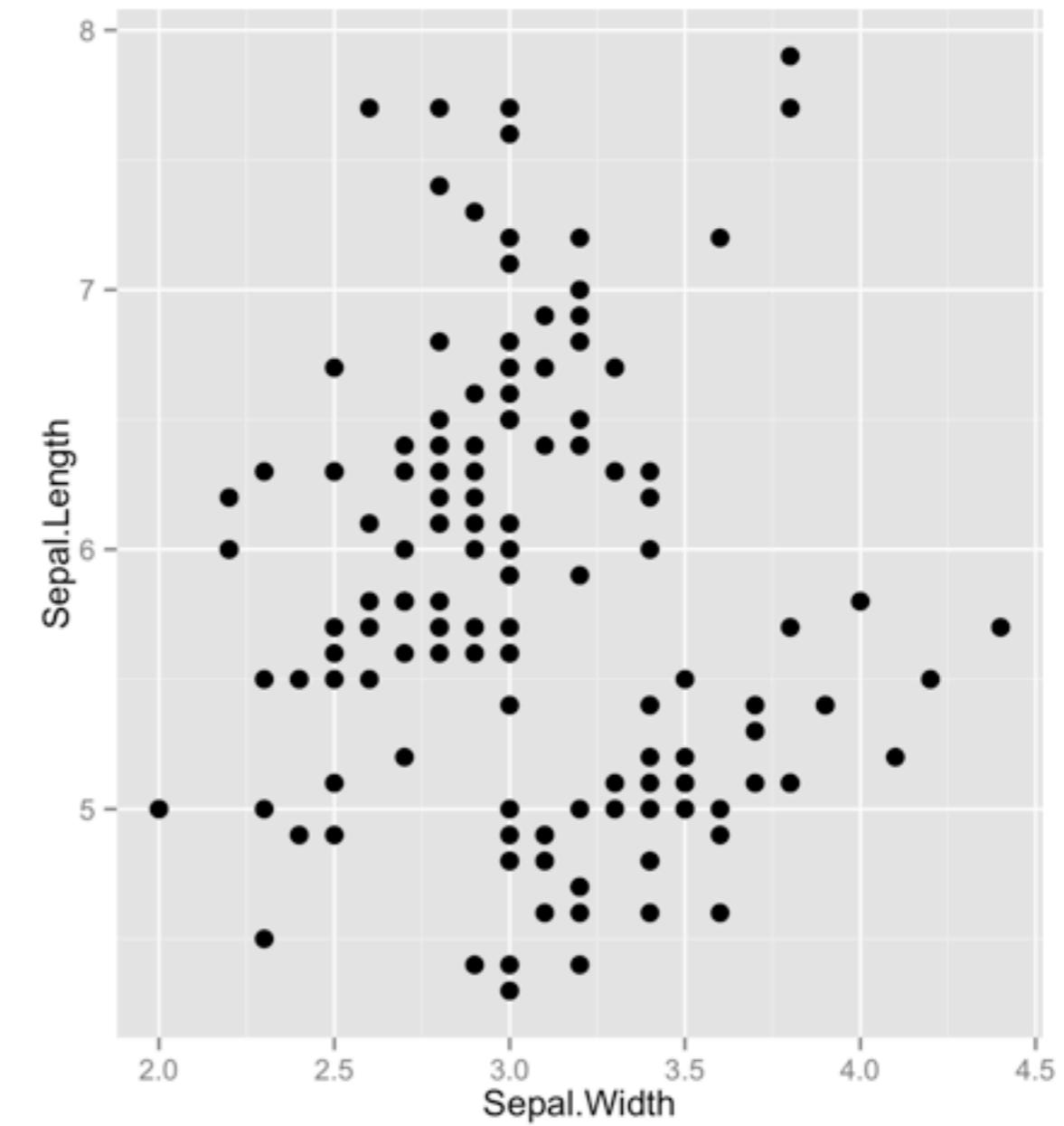
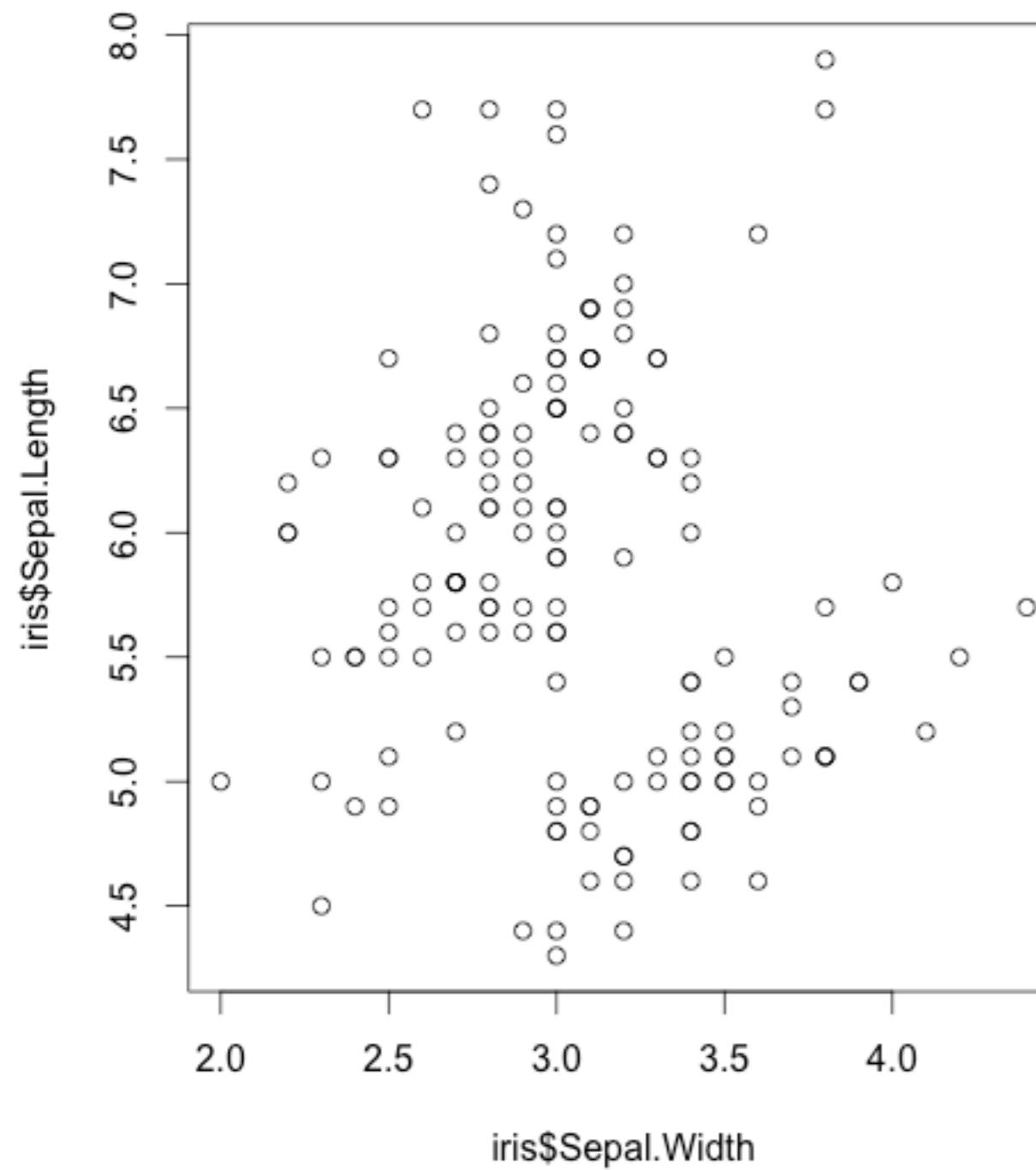
plot



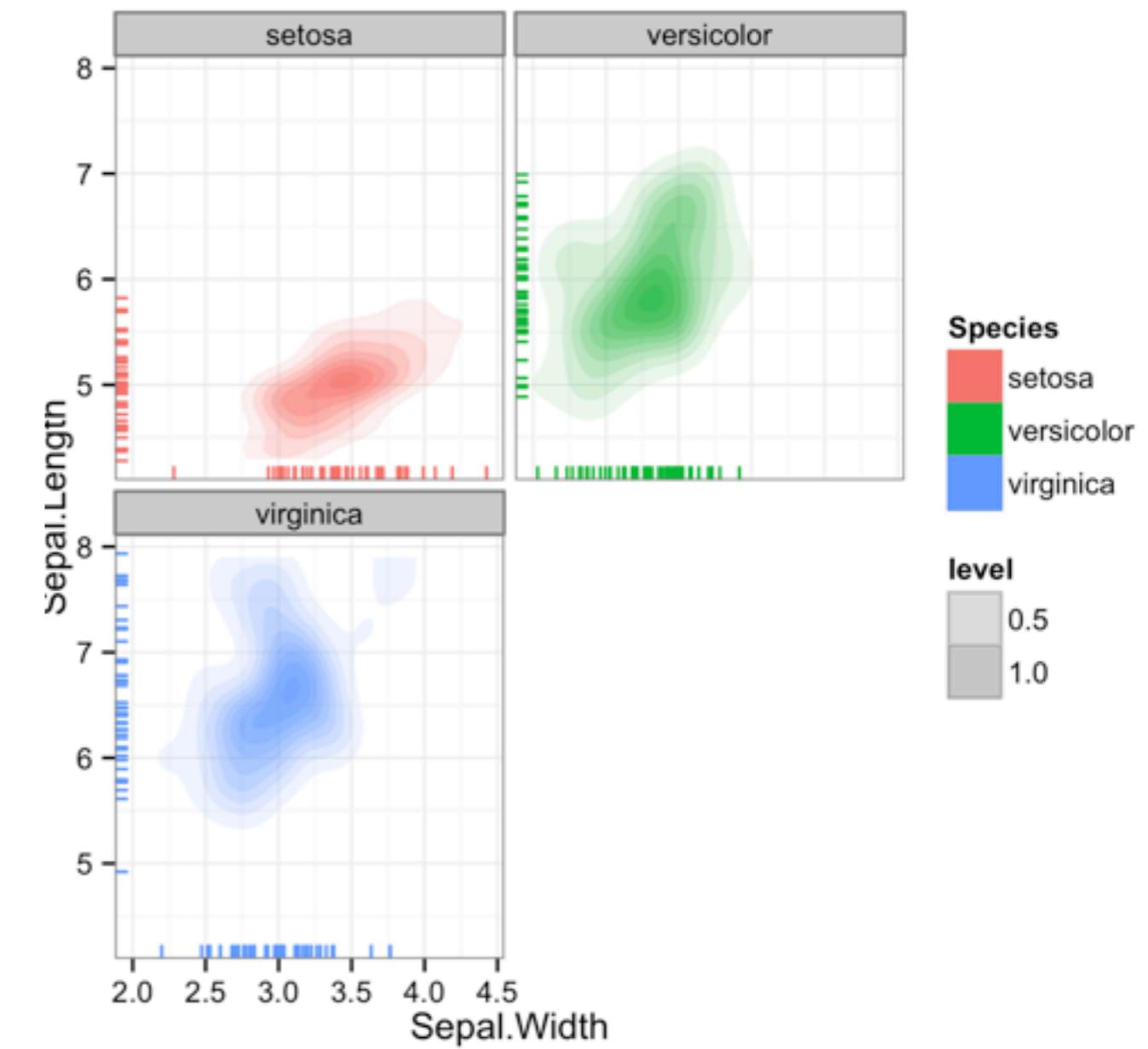
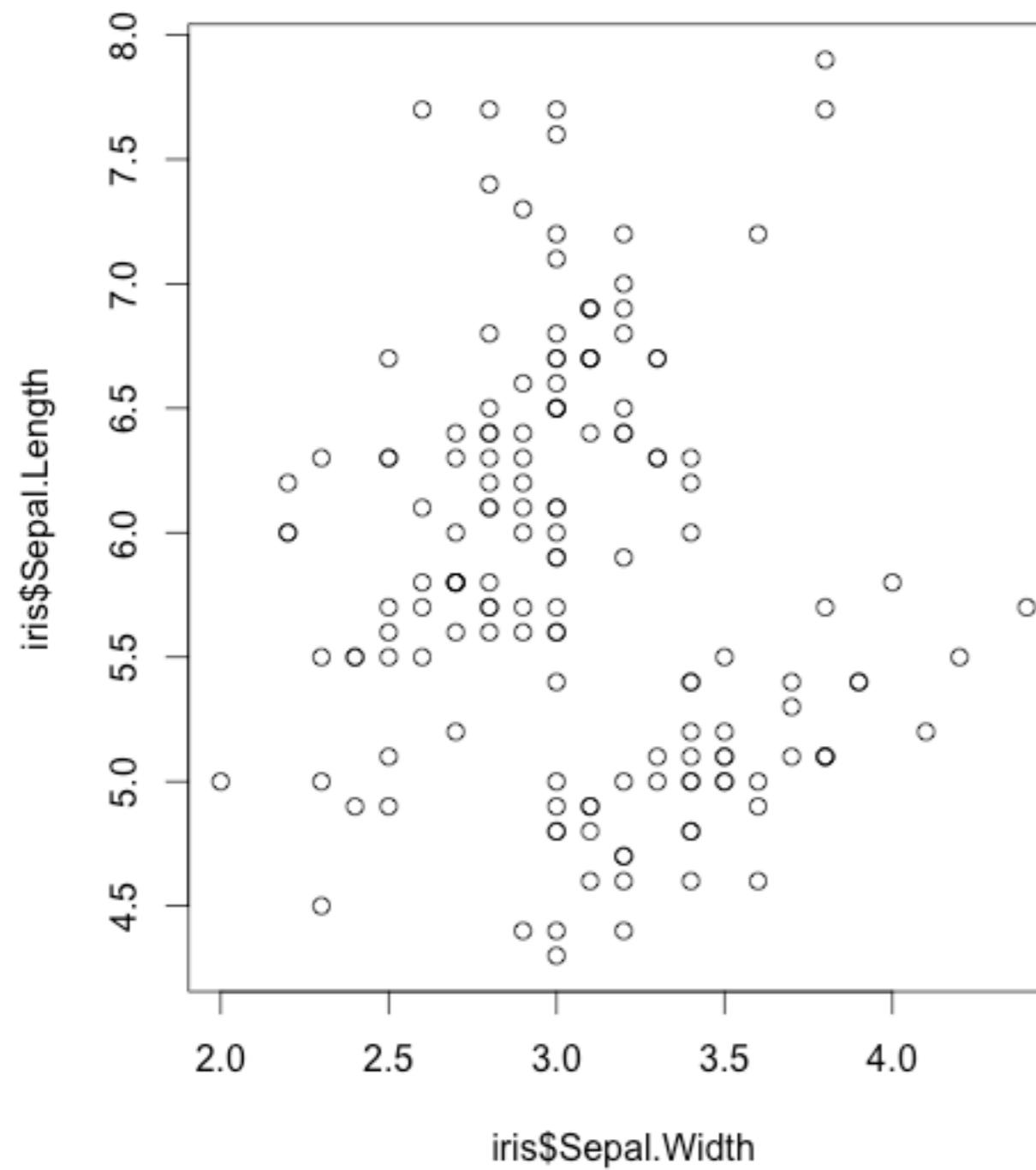
```
plot(iris$Sepal.Length,  
     iris$Sepal.Width)
```

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

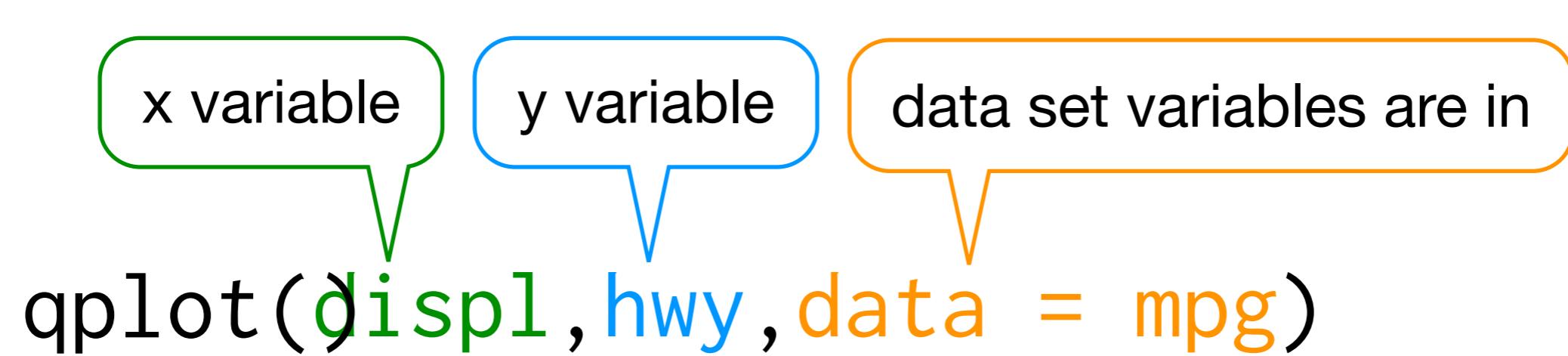
ggplot2



ggplot2



(quick) plots in R



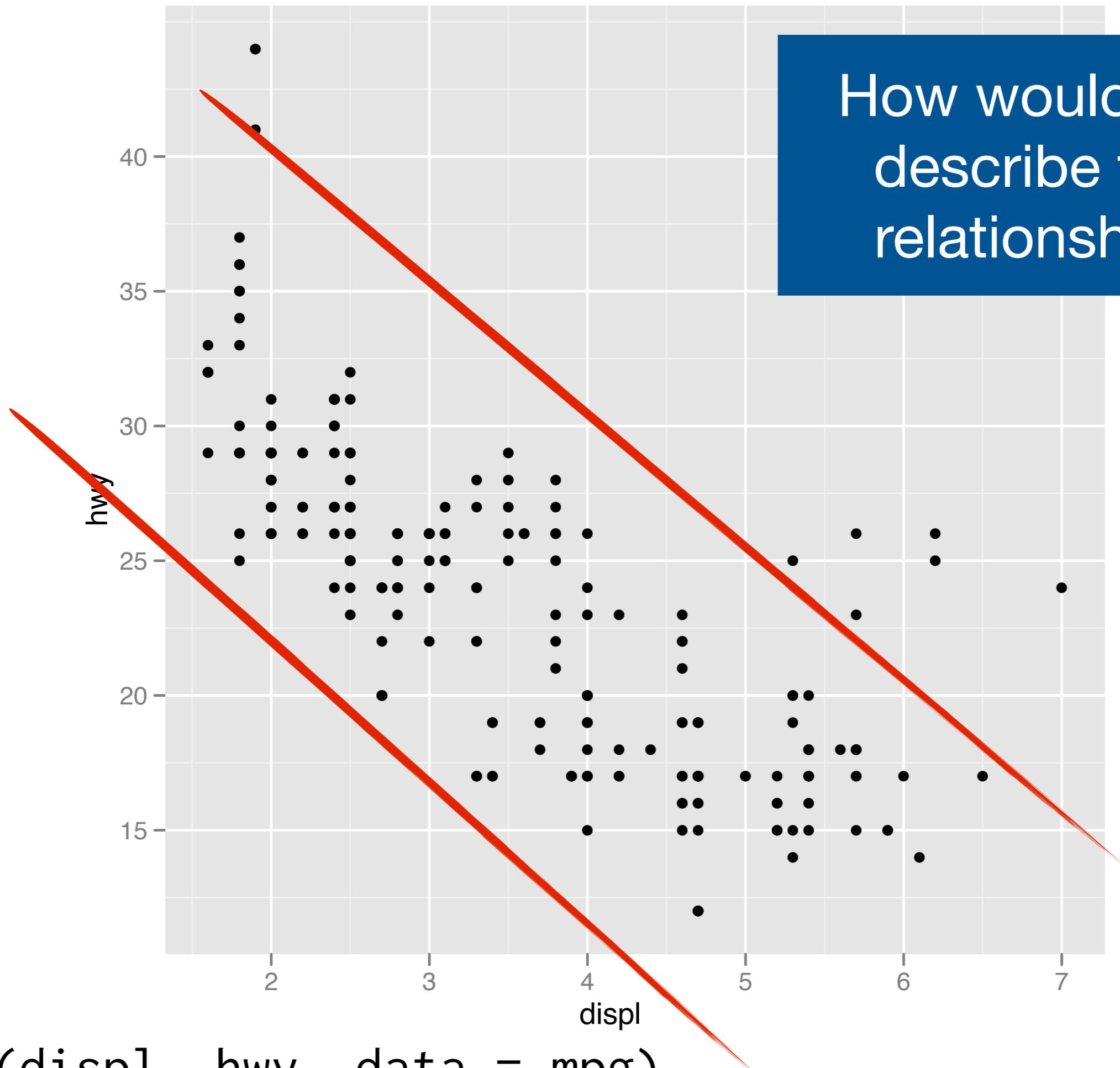
```
qplot(displ, hwy, data = mpg)
```

x variable

y variable

data set variables are in

How would you
describe this
relationship?



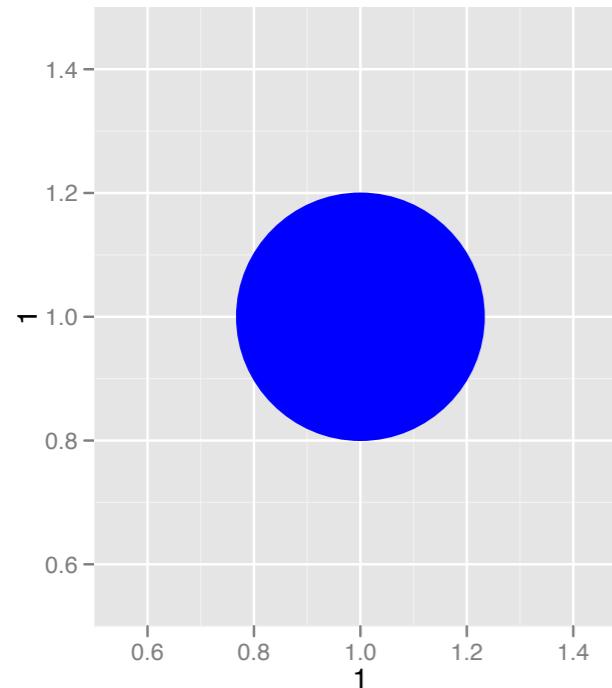
Additional variables

Can display additional variables with

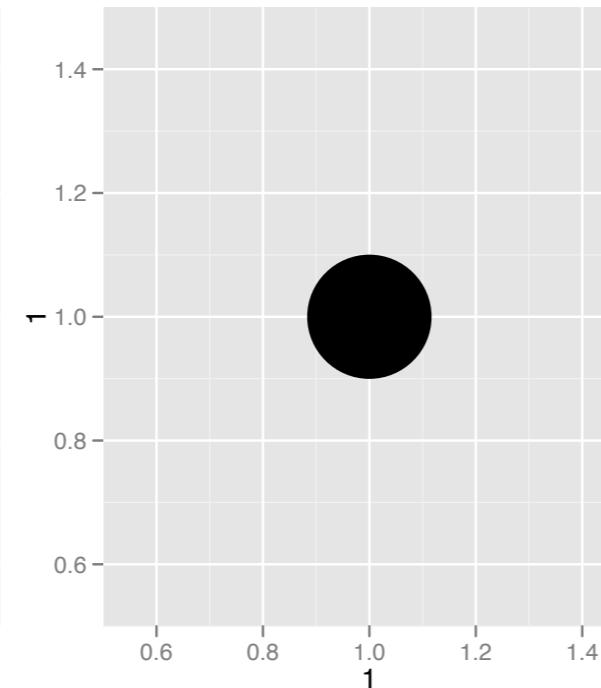
- **aesthetics** (like shape, colour, size), or
- **faceting** (small multiples displaying different subsets)

Aesthetics

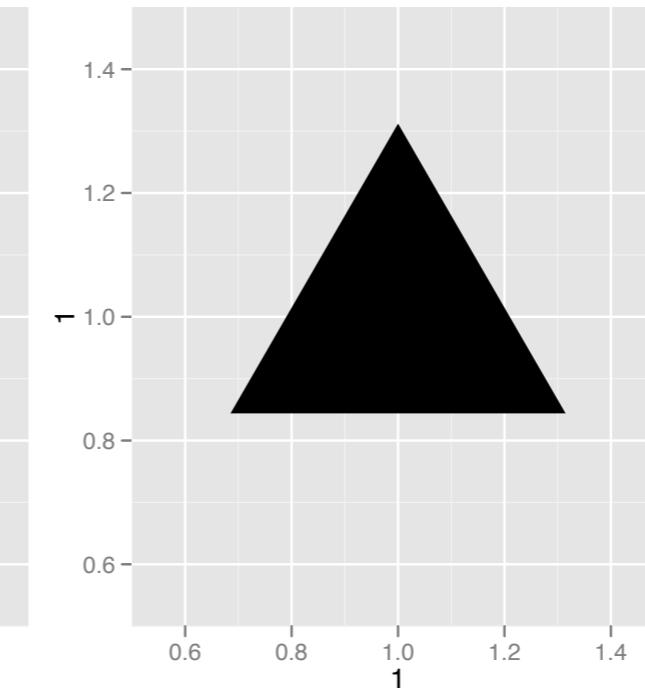
Visual characteristics that can be mapped to data



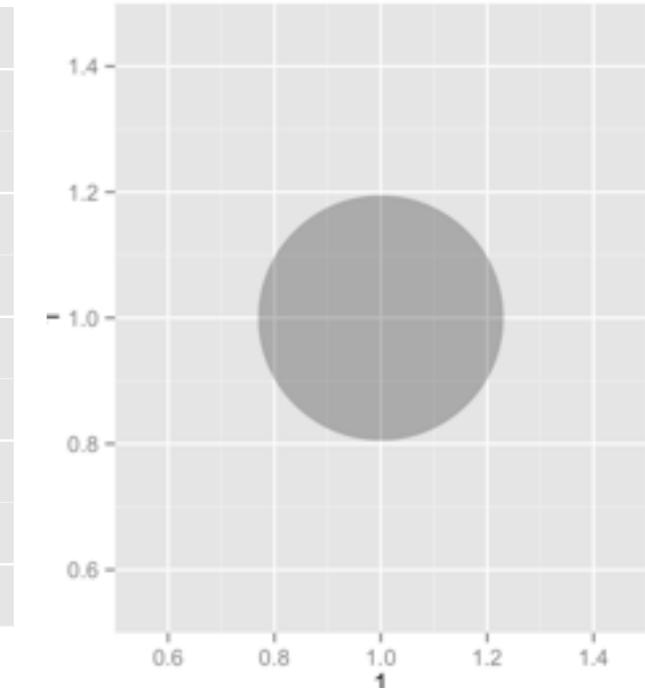
color



size



shape



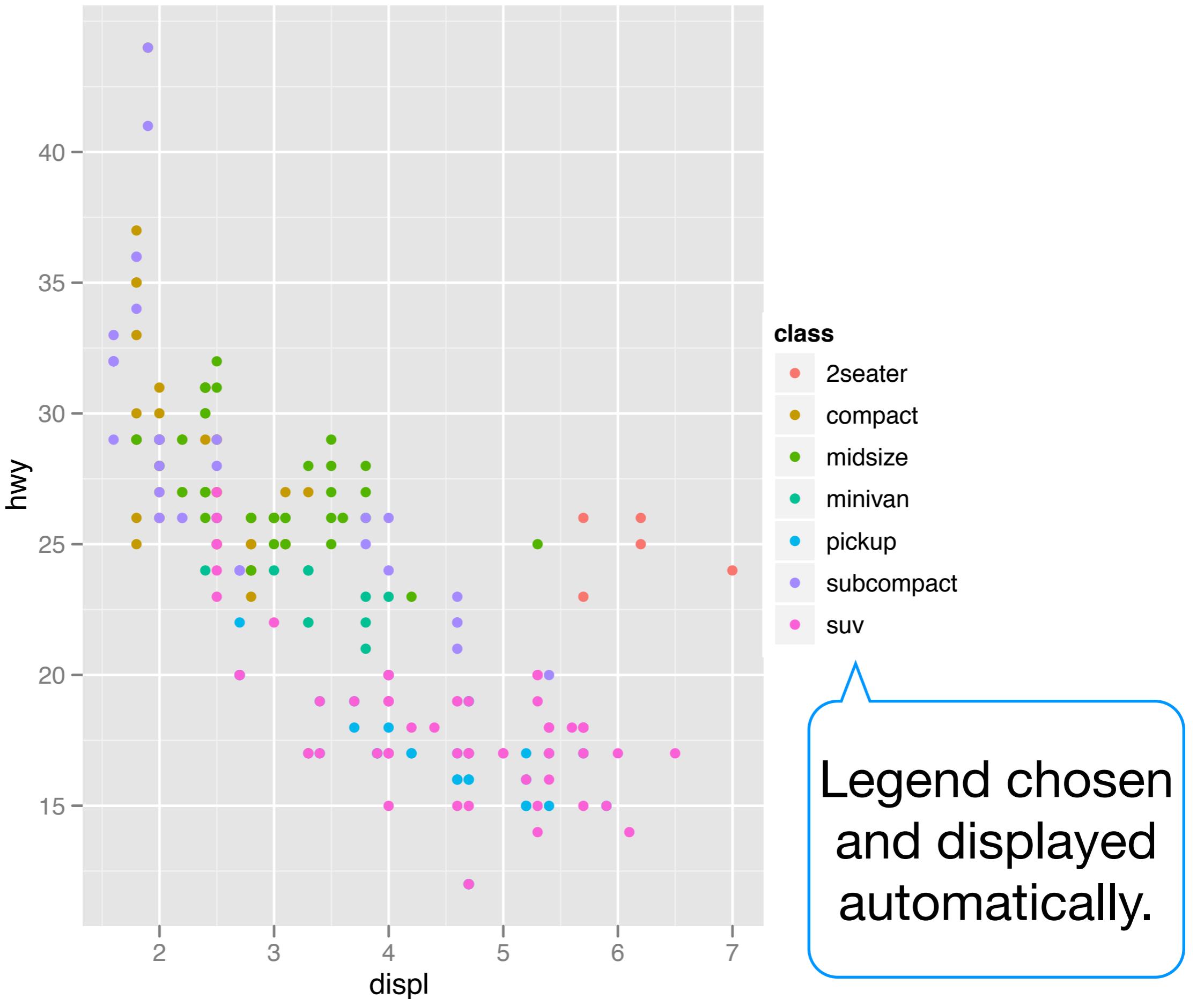
alpha
(transparency)

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)
```



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

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)
```

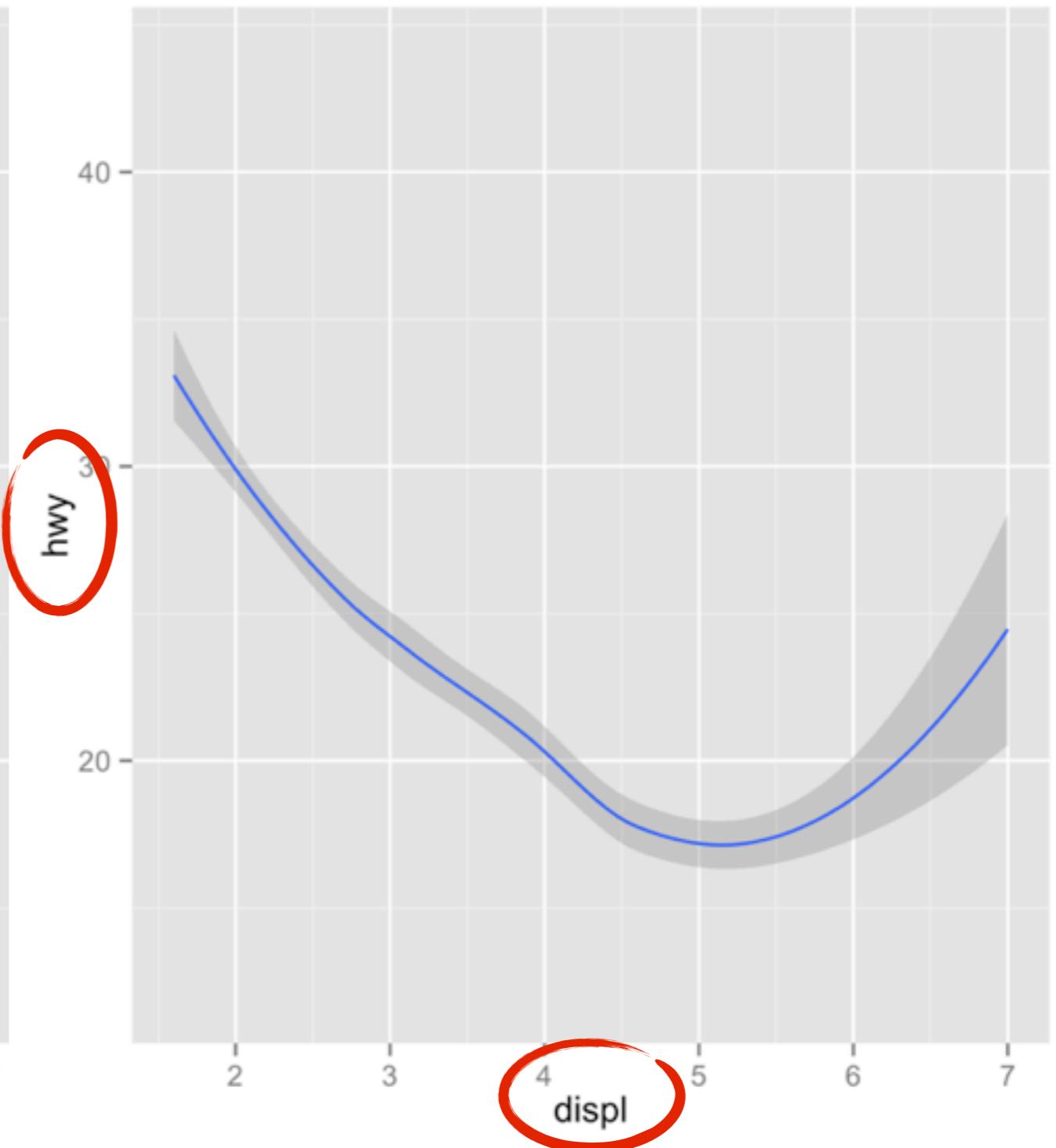
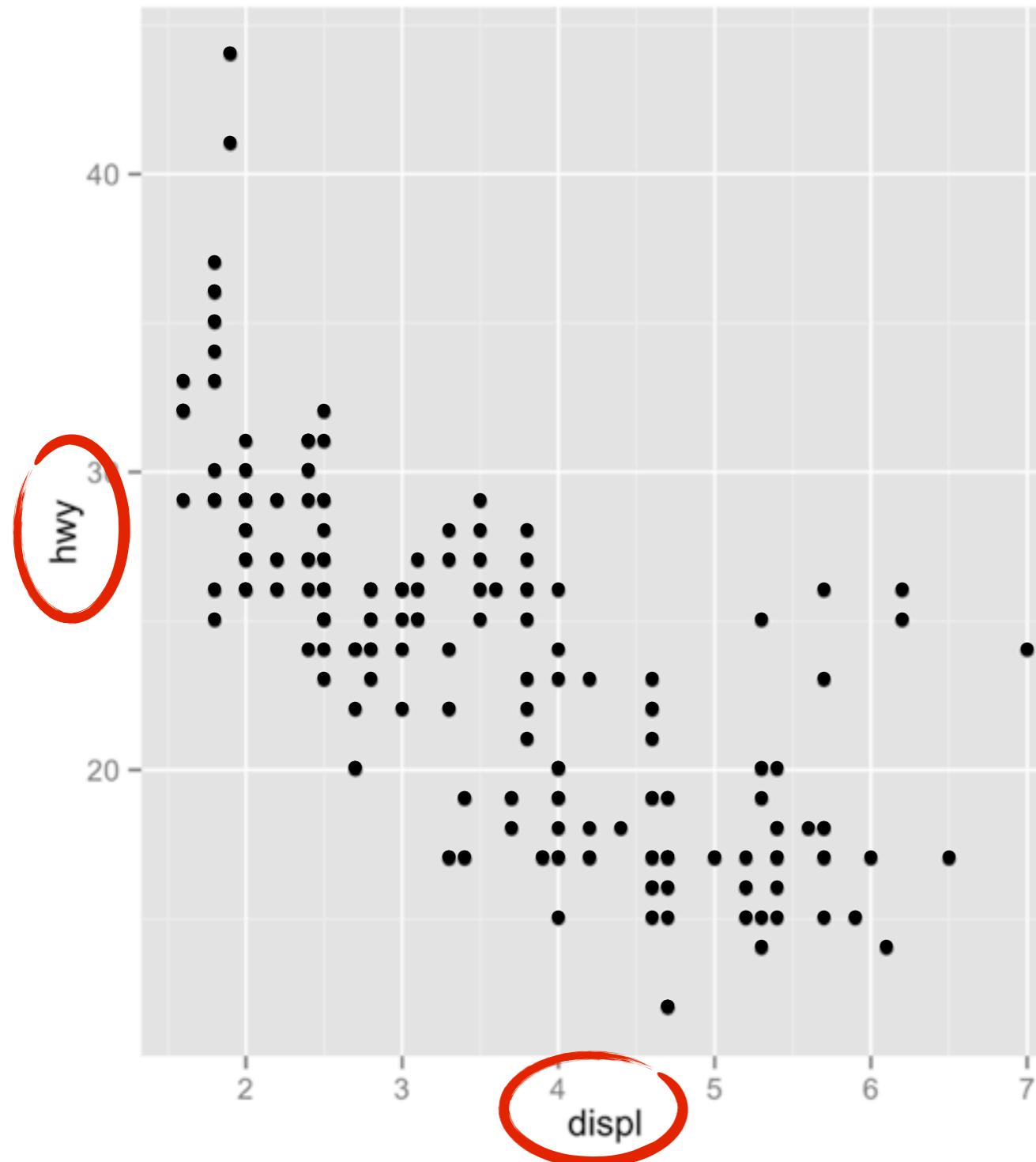
Summary

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

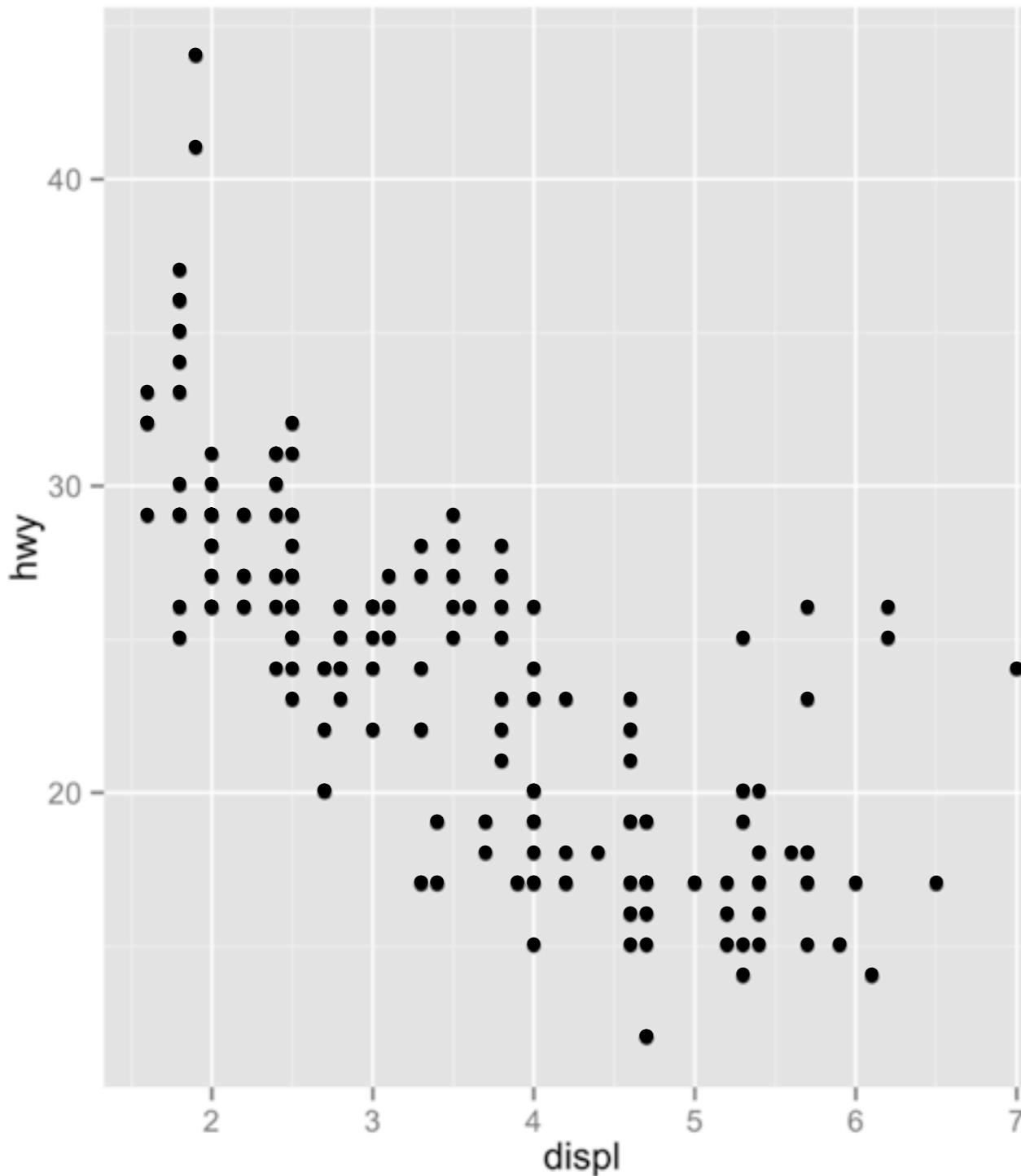
`facet_wrap()`: 1d ribbon wrapped into 2d

How are these plots similar?

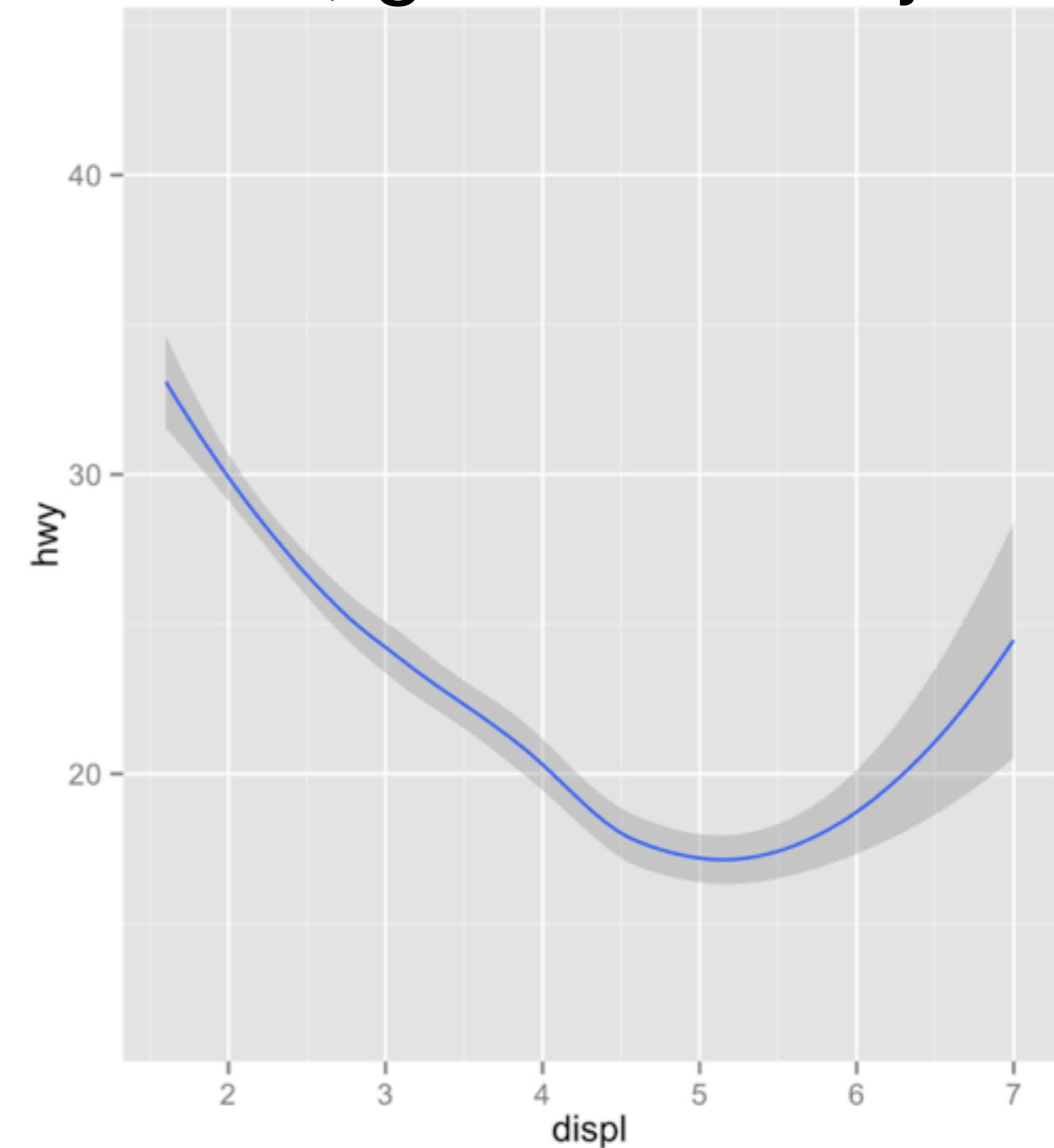
Same: x var, y var, data



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

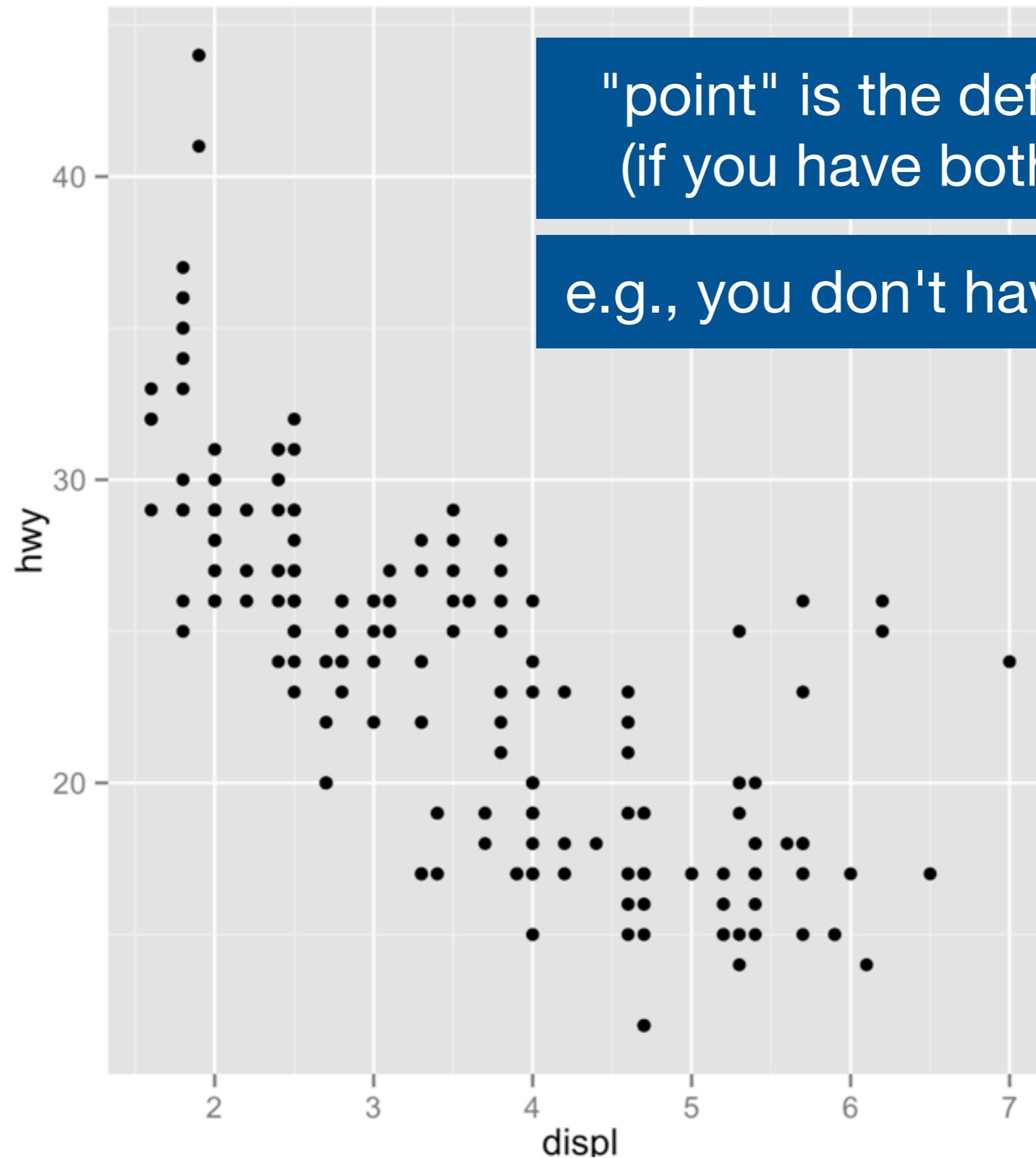
x variable

y variable

data set
variables are in

type of plot

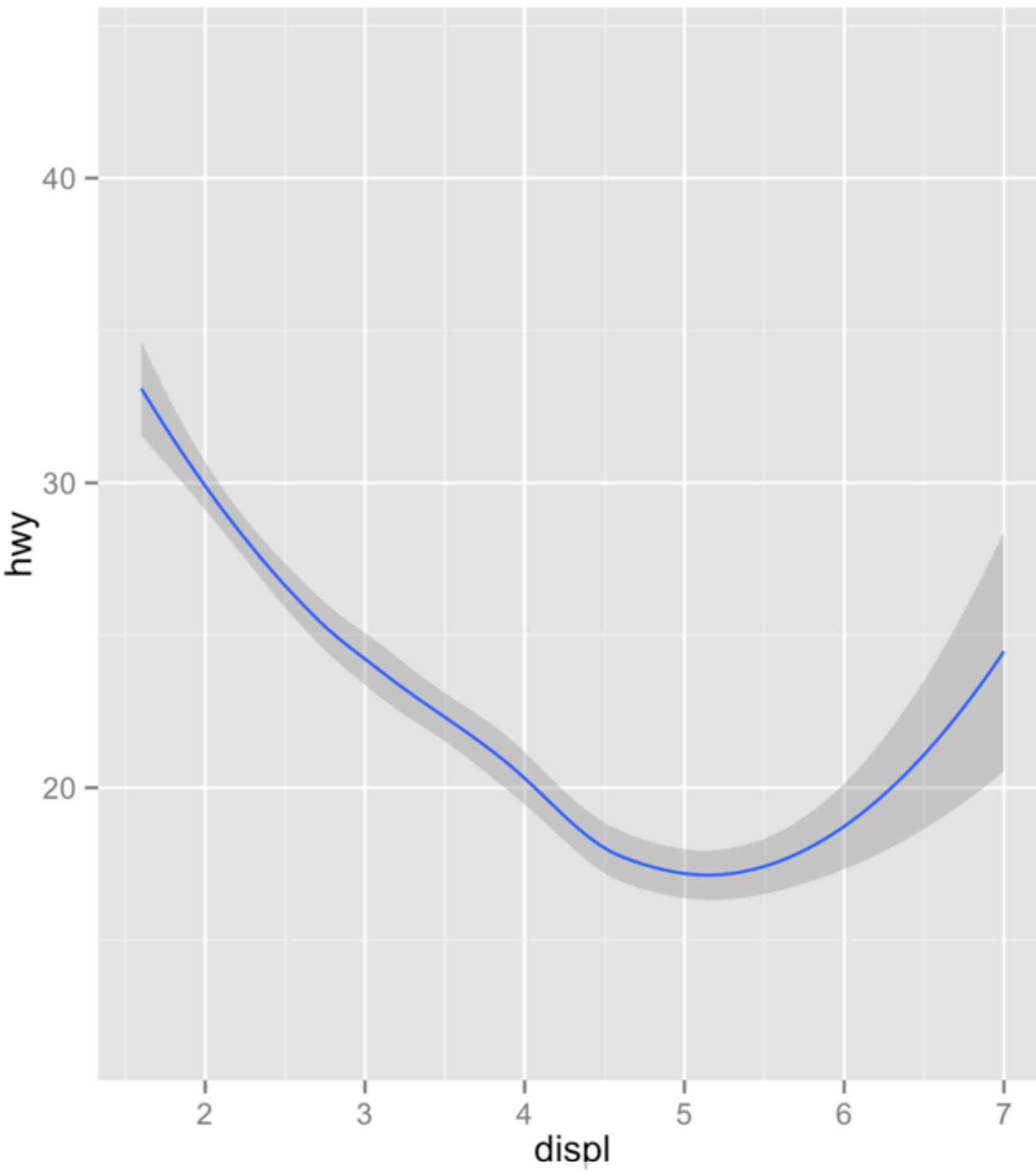
```
qplot(displ, hwy, data = mpg) geom = "smooth")
```



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

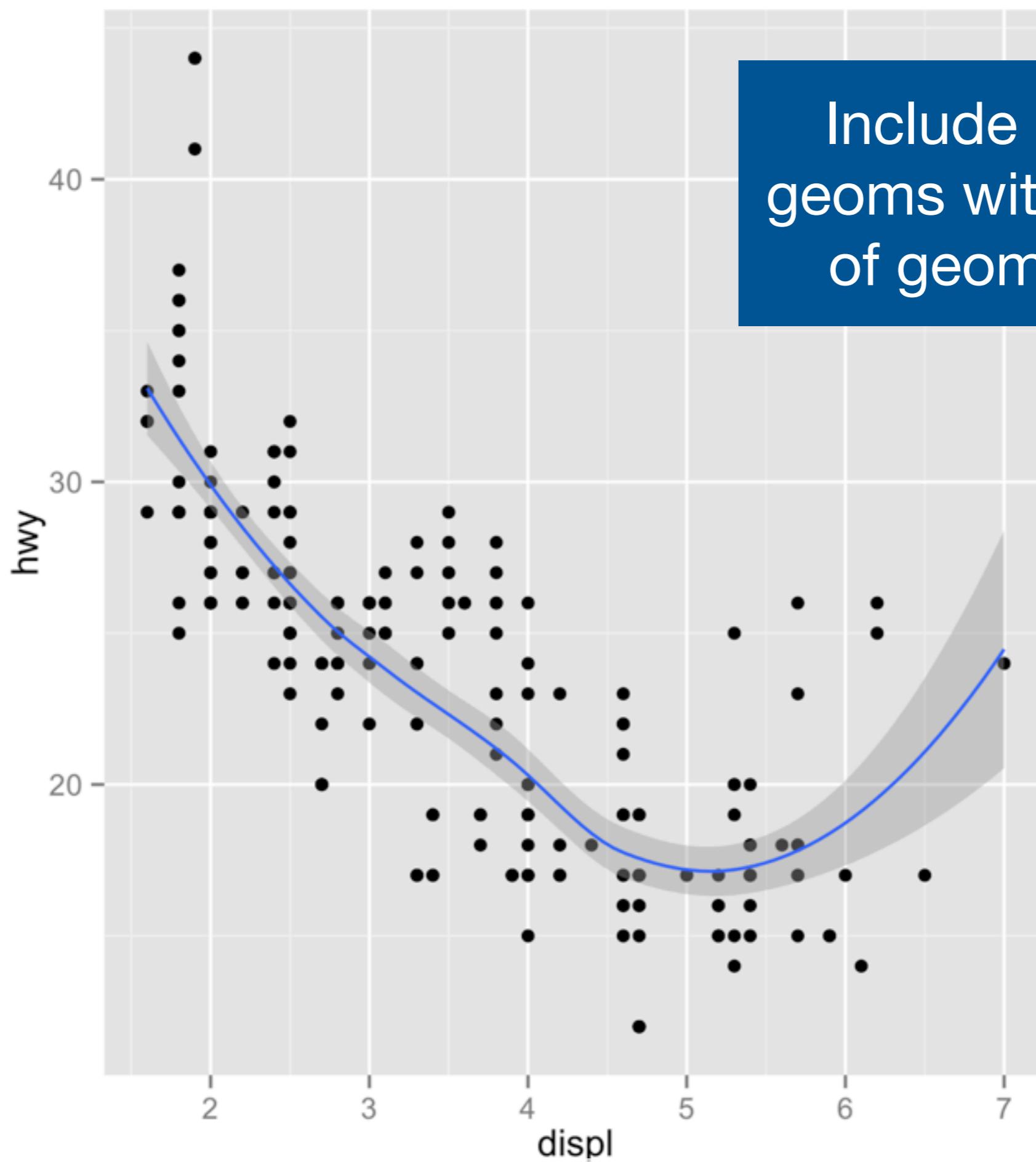
e.g., you don't have to type it

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



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

Include multiple
geoms with a vector
of geom names

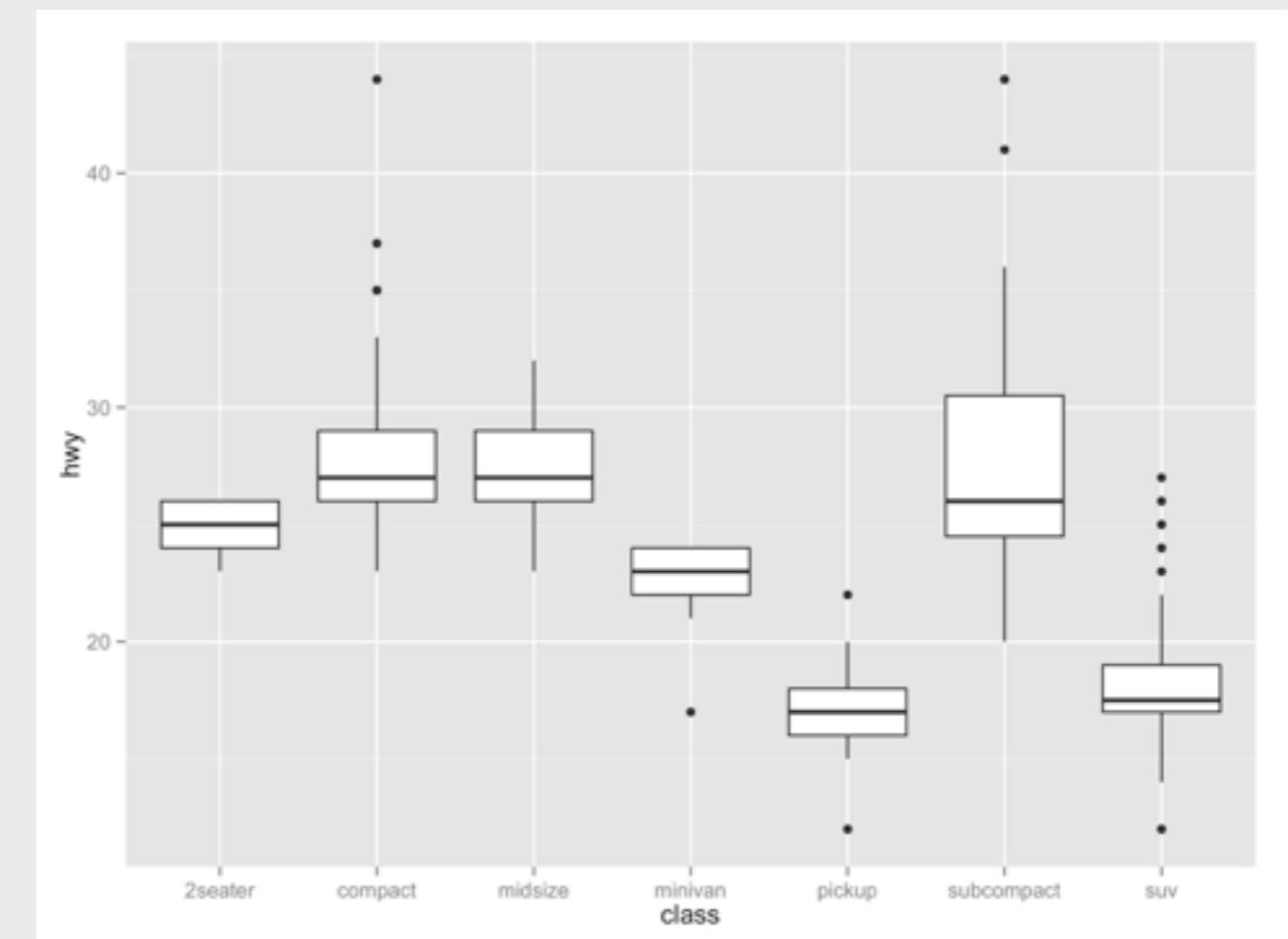
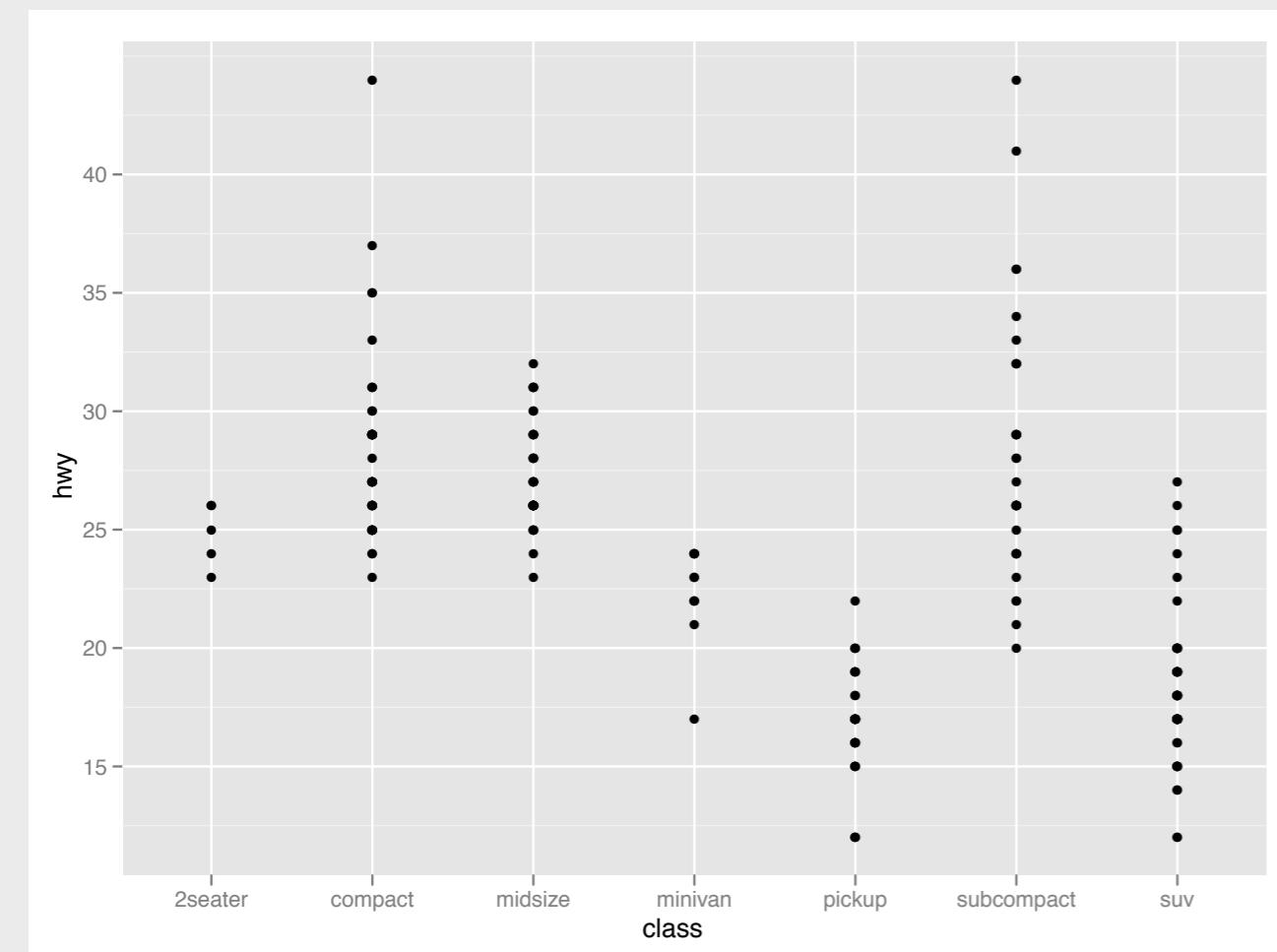


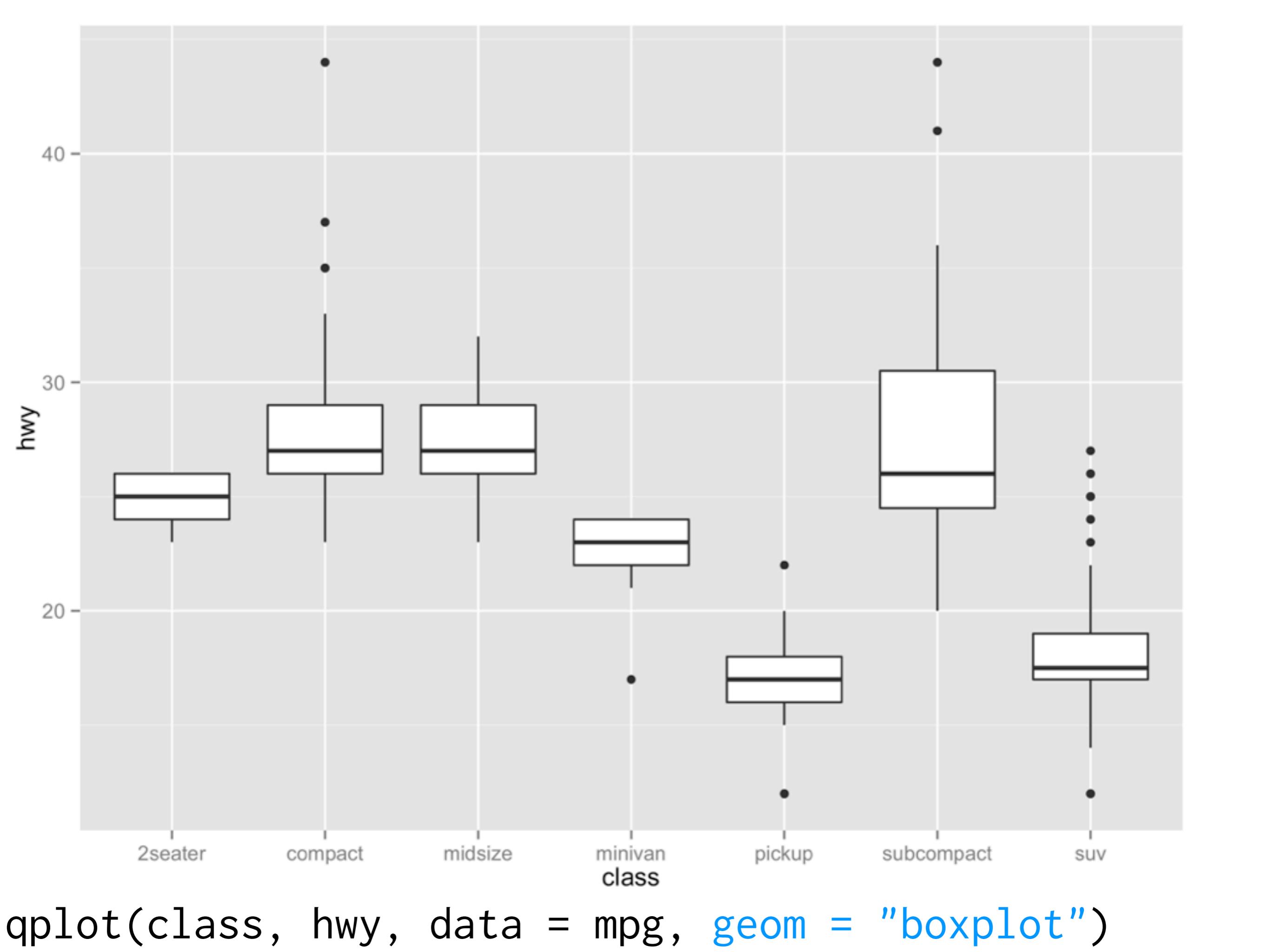
```
qplot(displ, hwy, data = mpg, geom = c("point", "smooth"))
```

Your turn

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

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





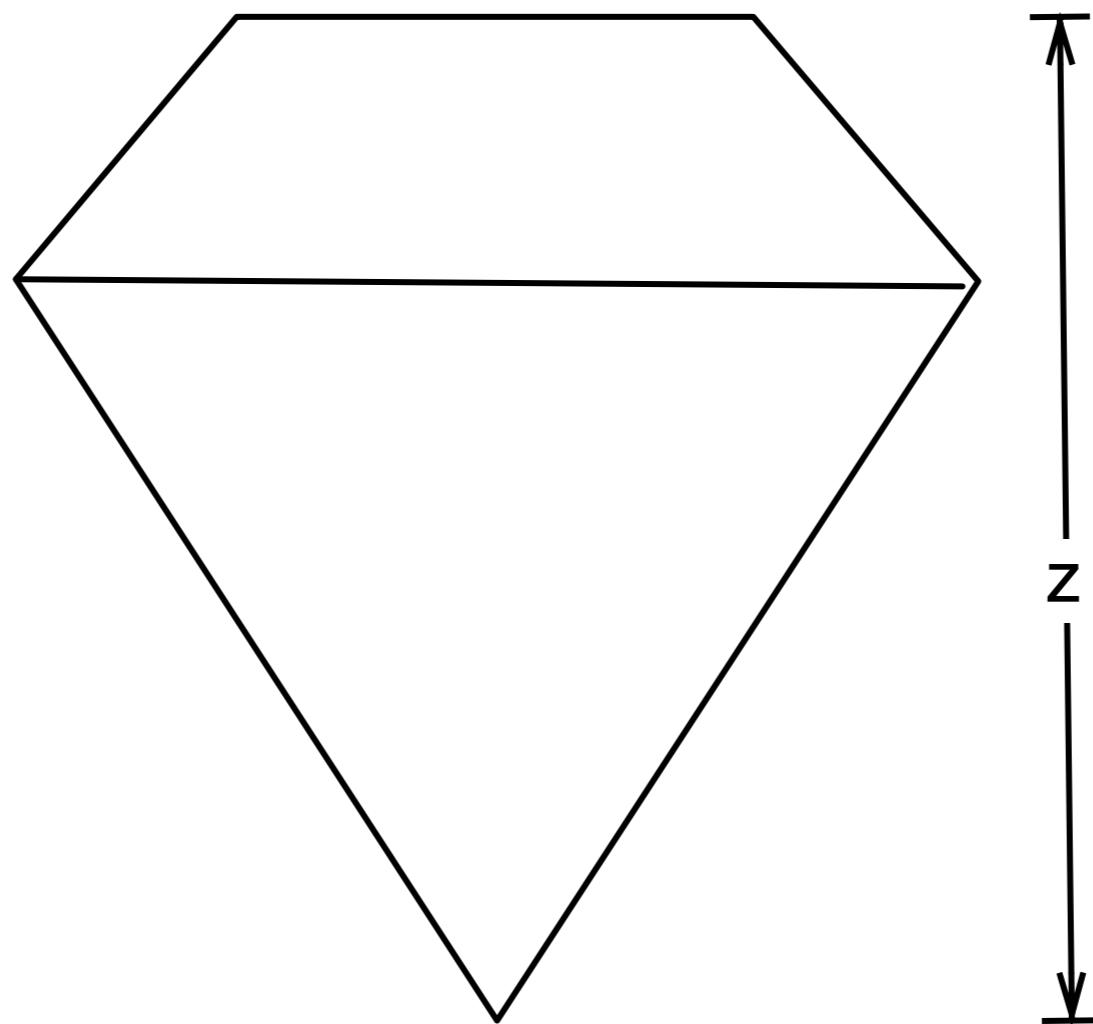
Diamonds data

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



← →
x

← table width →



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

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



IF



VVSI



VVS2



VS1

Illustration of inclusions as seen under X10 magnification



VS2



SI1



SI2



II

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.*

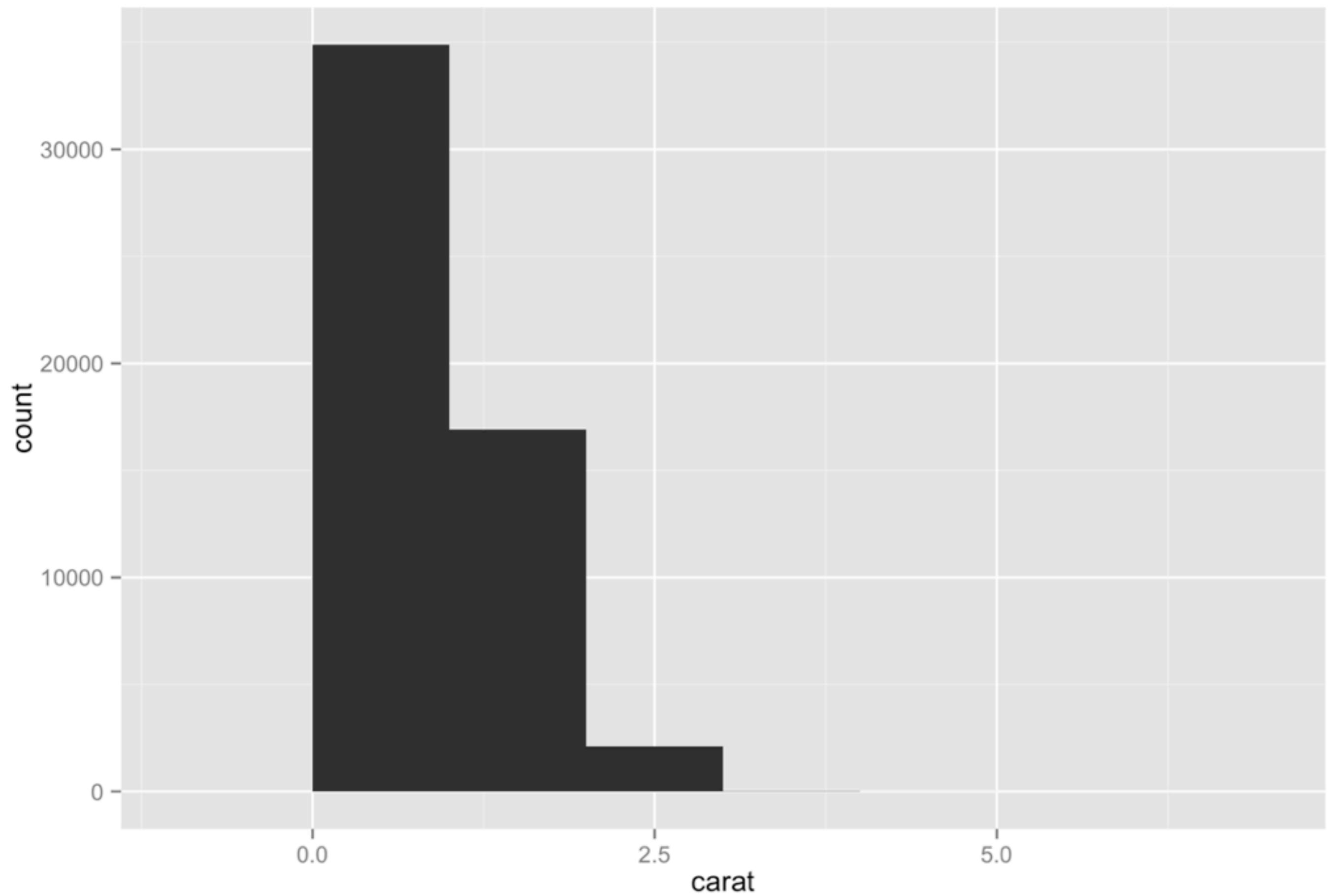
e.g. binwidths in a histogram

Parameters

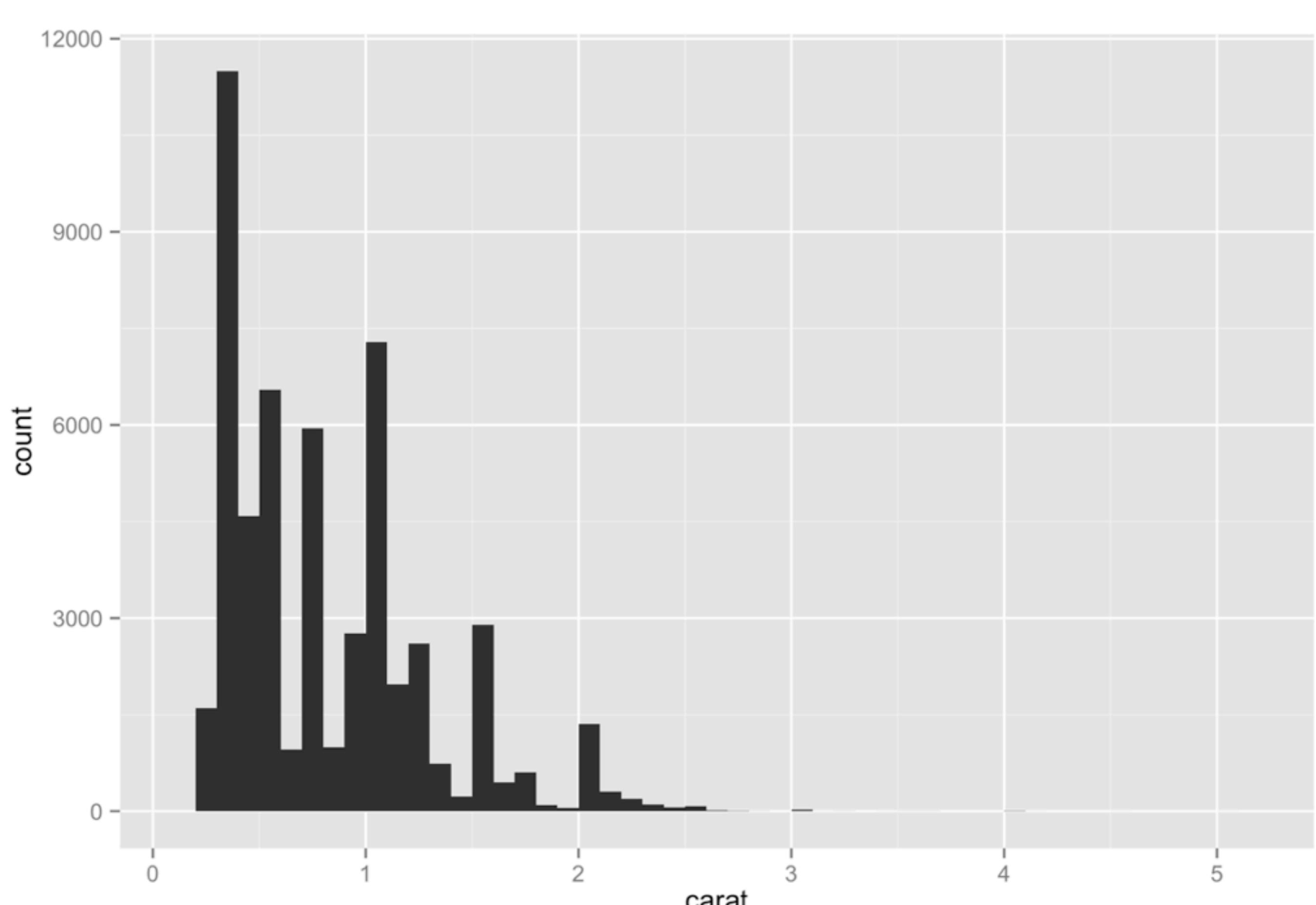
parameter
name

value

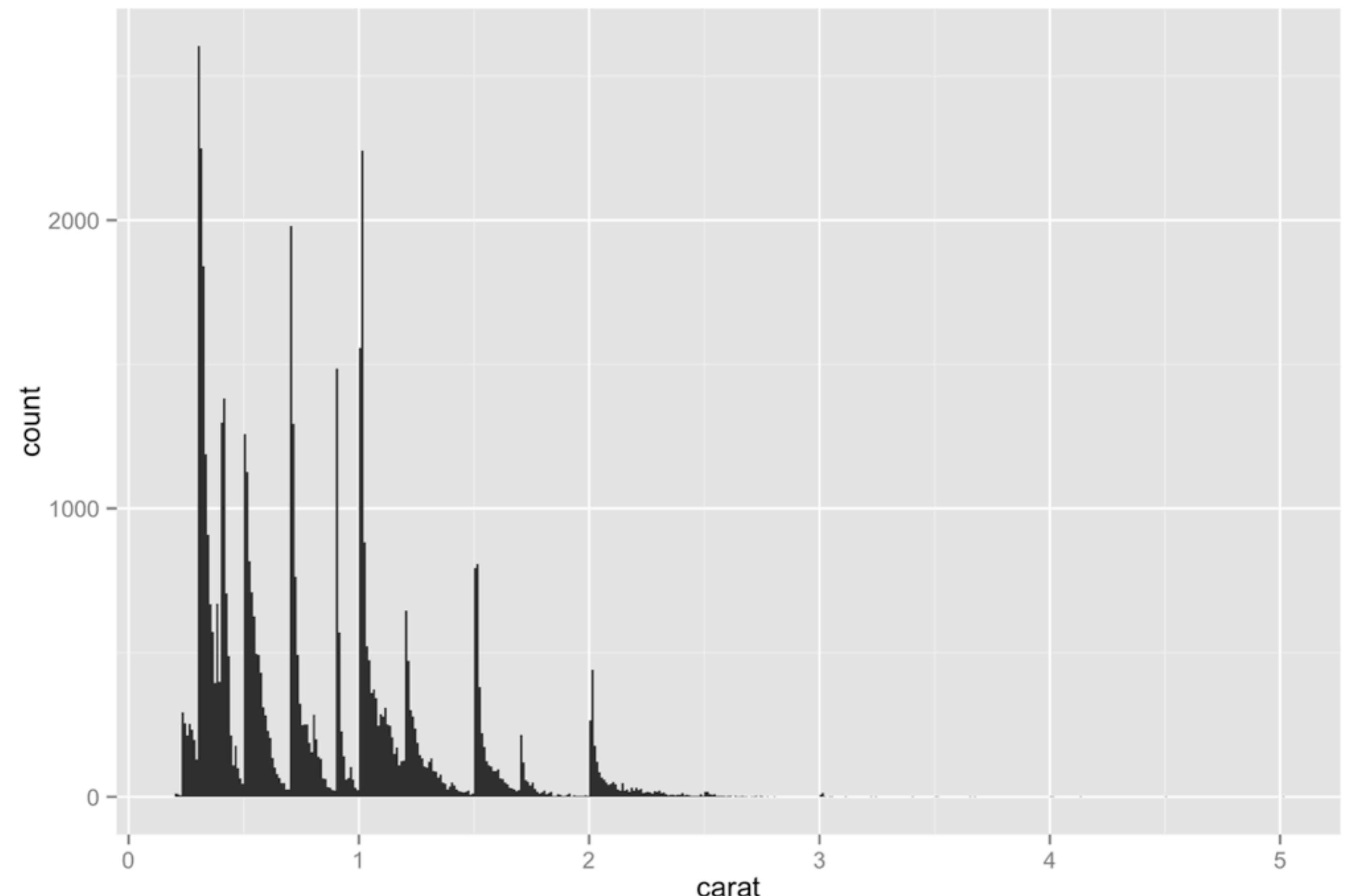
```
qplot(displ, data = mpg, binwidth = 1)
```



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

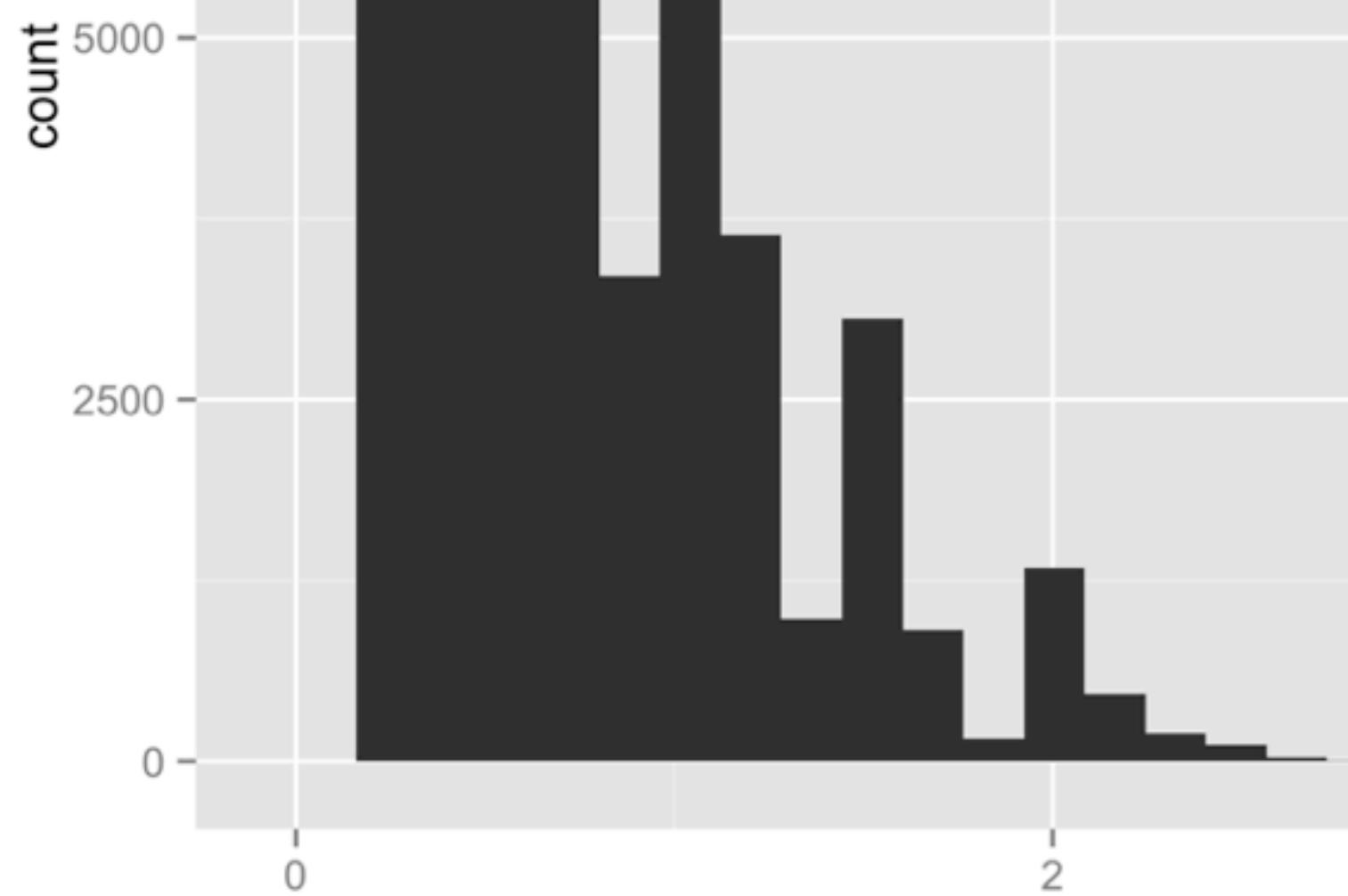


```
qplot(carat, data = diamonds, binwidth = 0.1)
```



```
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?

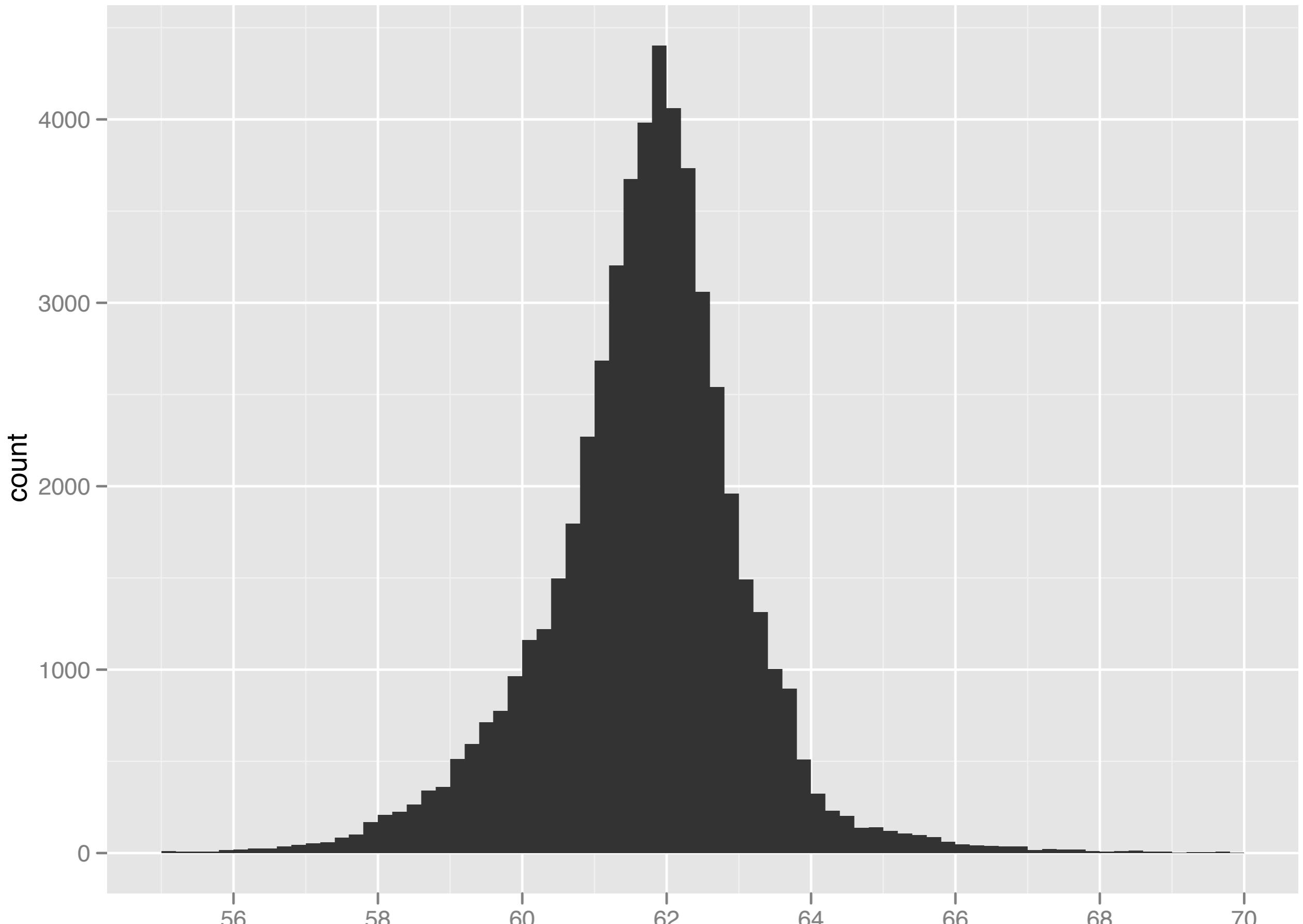
```
qplot(price, data = diamonds)
```

Hint: $0 < \text{price} < 18823$.
Do not set $\text{binwidth} = 1$!

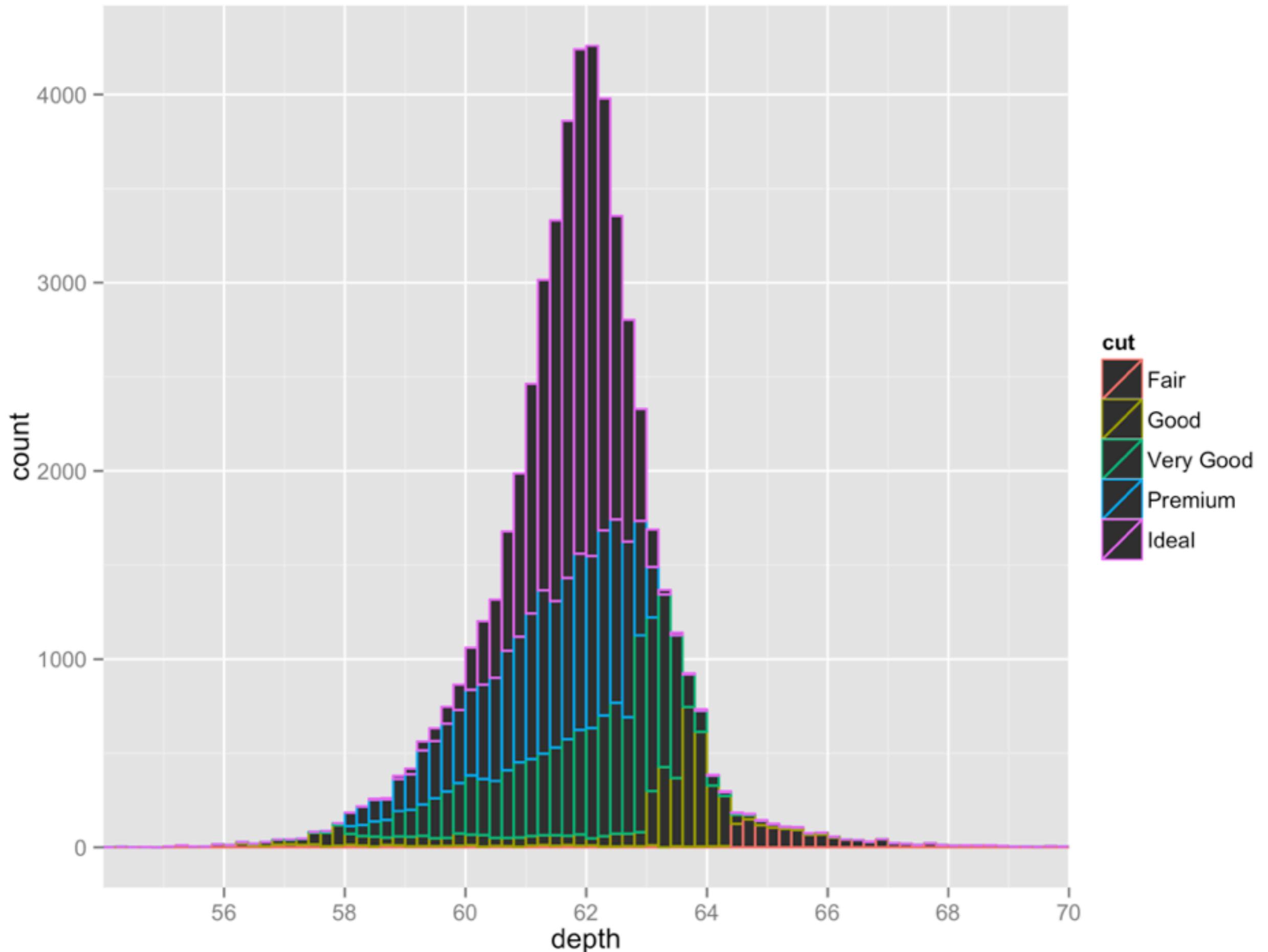
```
qplot(price, data = diamonds)
qplot(price, data = diamonds, binwidth = 500)
qplot(price, data = diamonds, binwidth = 100)
qplot(price, data = diamonds, binwidth = 50)
```

Additional variables

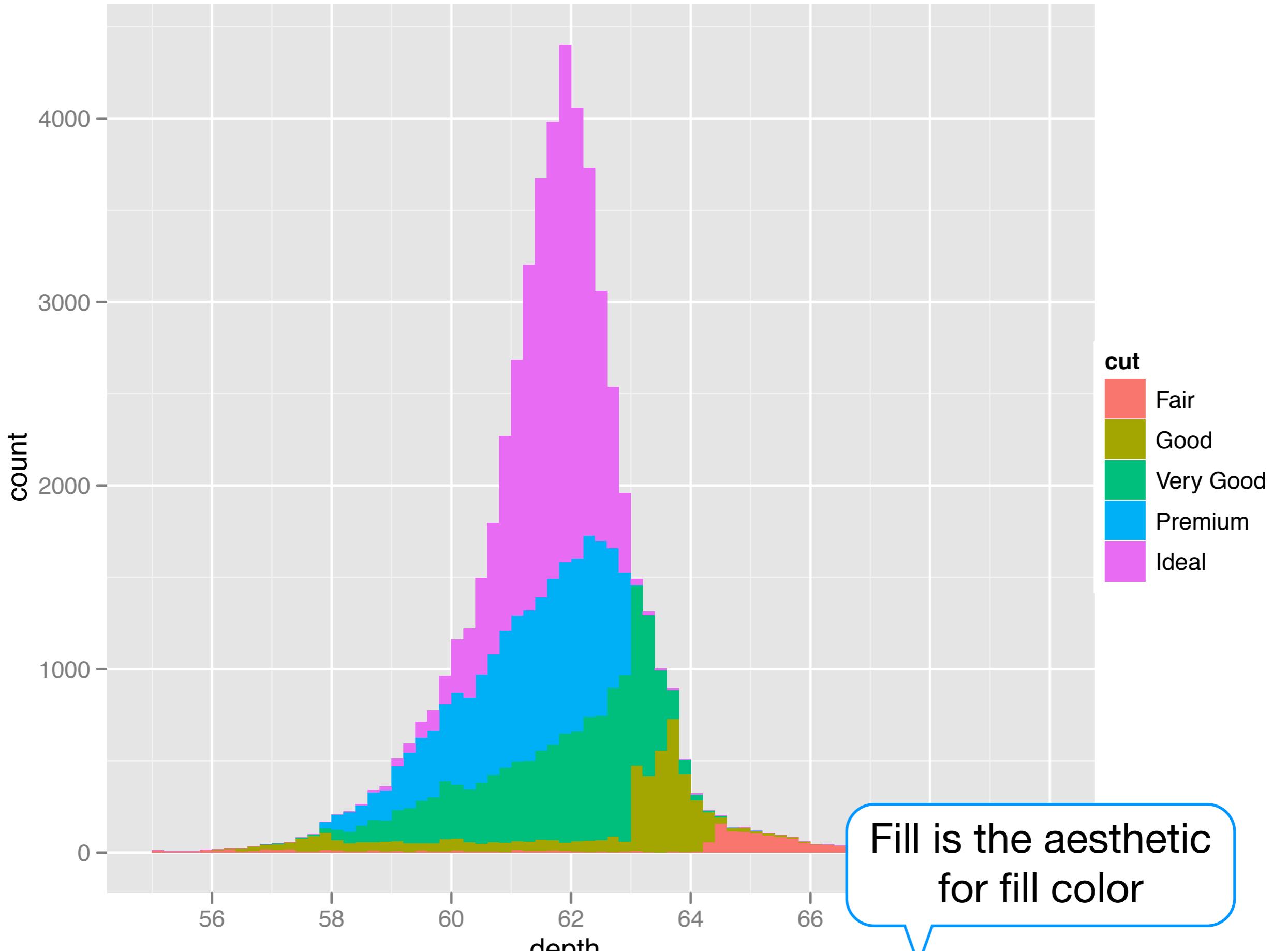
Often switching geoms is more effective than adding aesthetics or faceting to a histogram



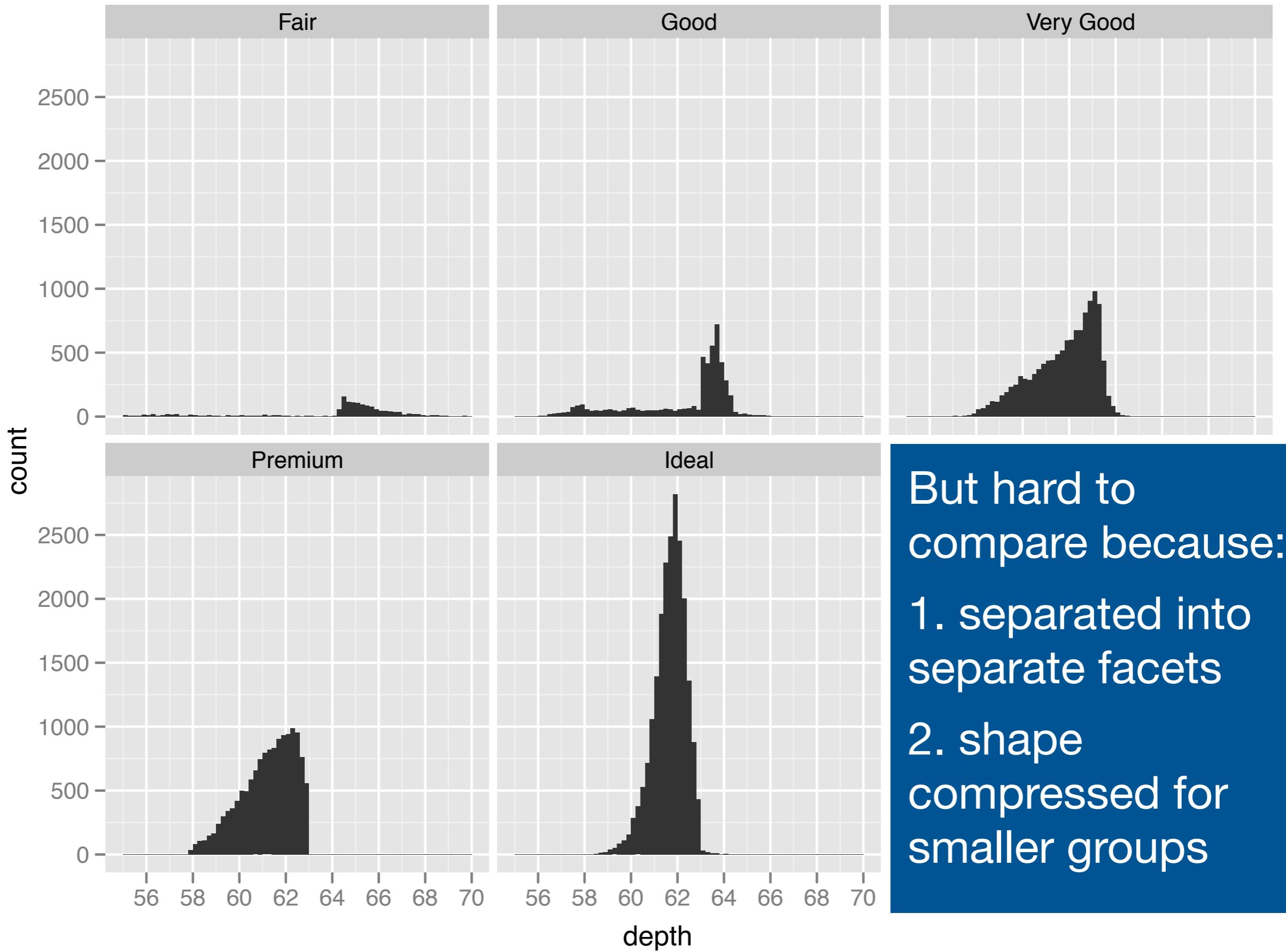
```
qplot(depth, data = diamonds, binwidth = 0.2)
```



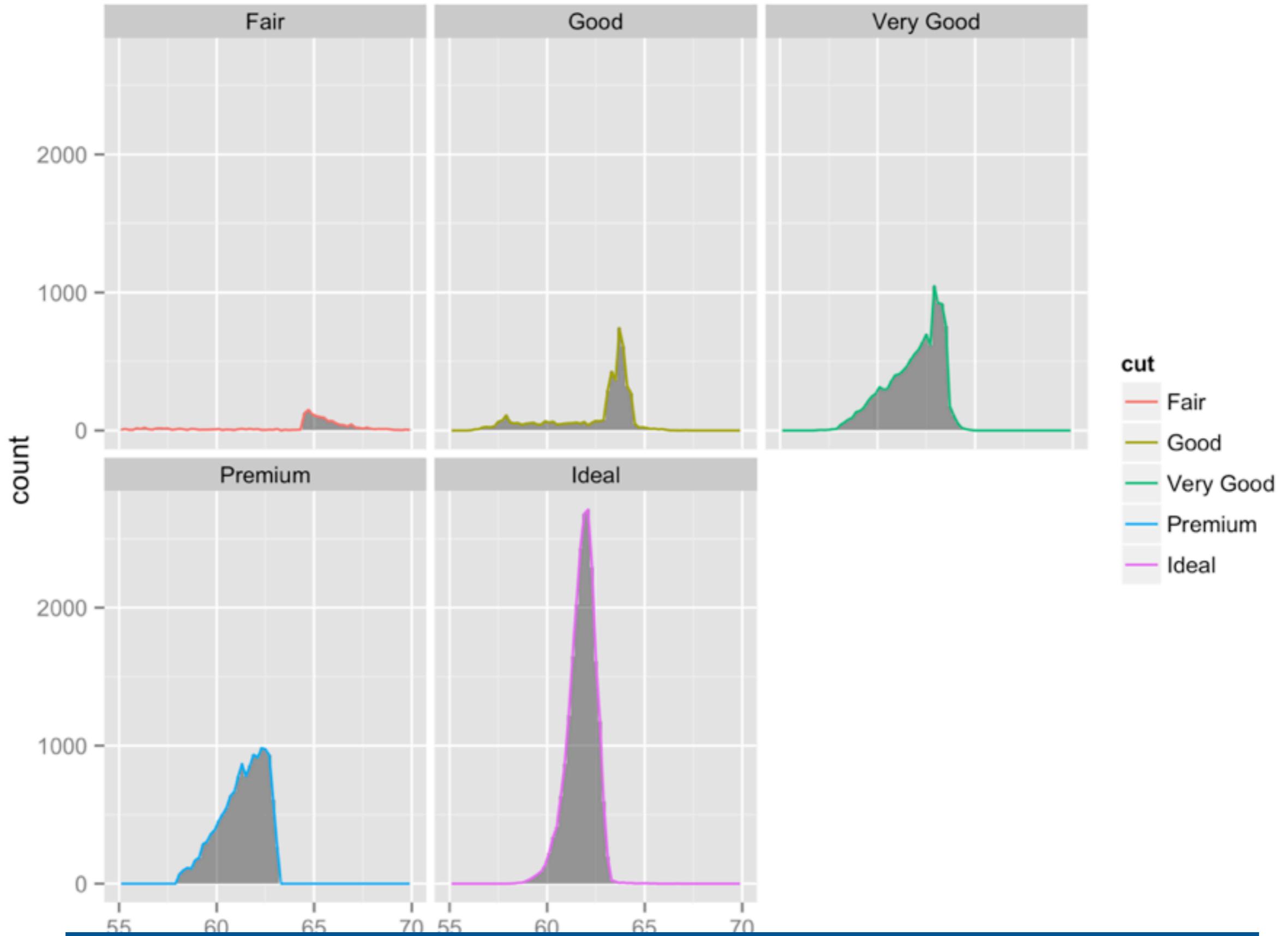
```
qplot(depth, data = diamonds, binwidth = 0.2, color = cut)
```



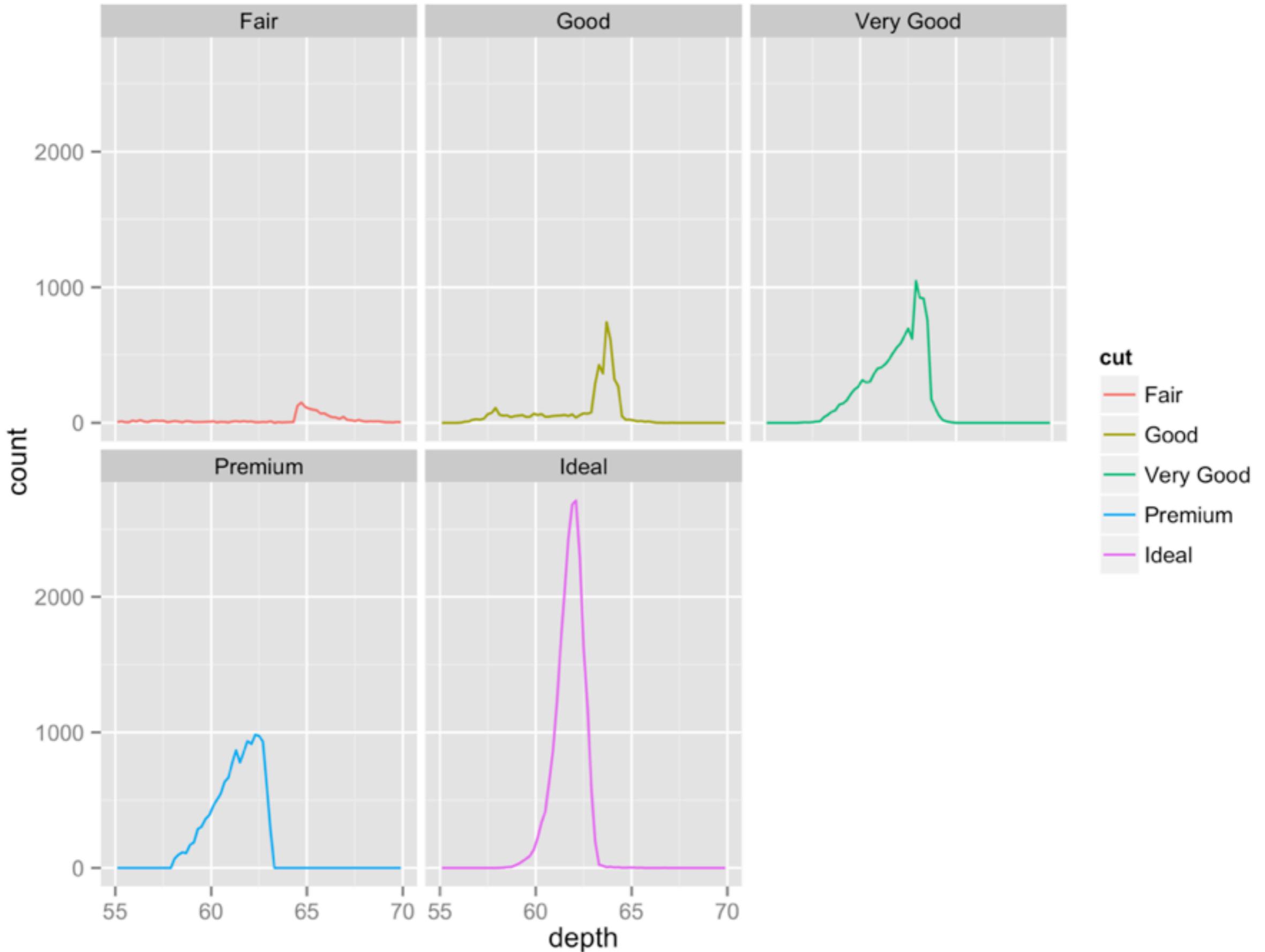
```
qplot(depth, data = diamonds, binwidth = 0.2, fill = cut)
```



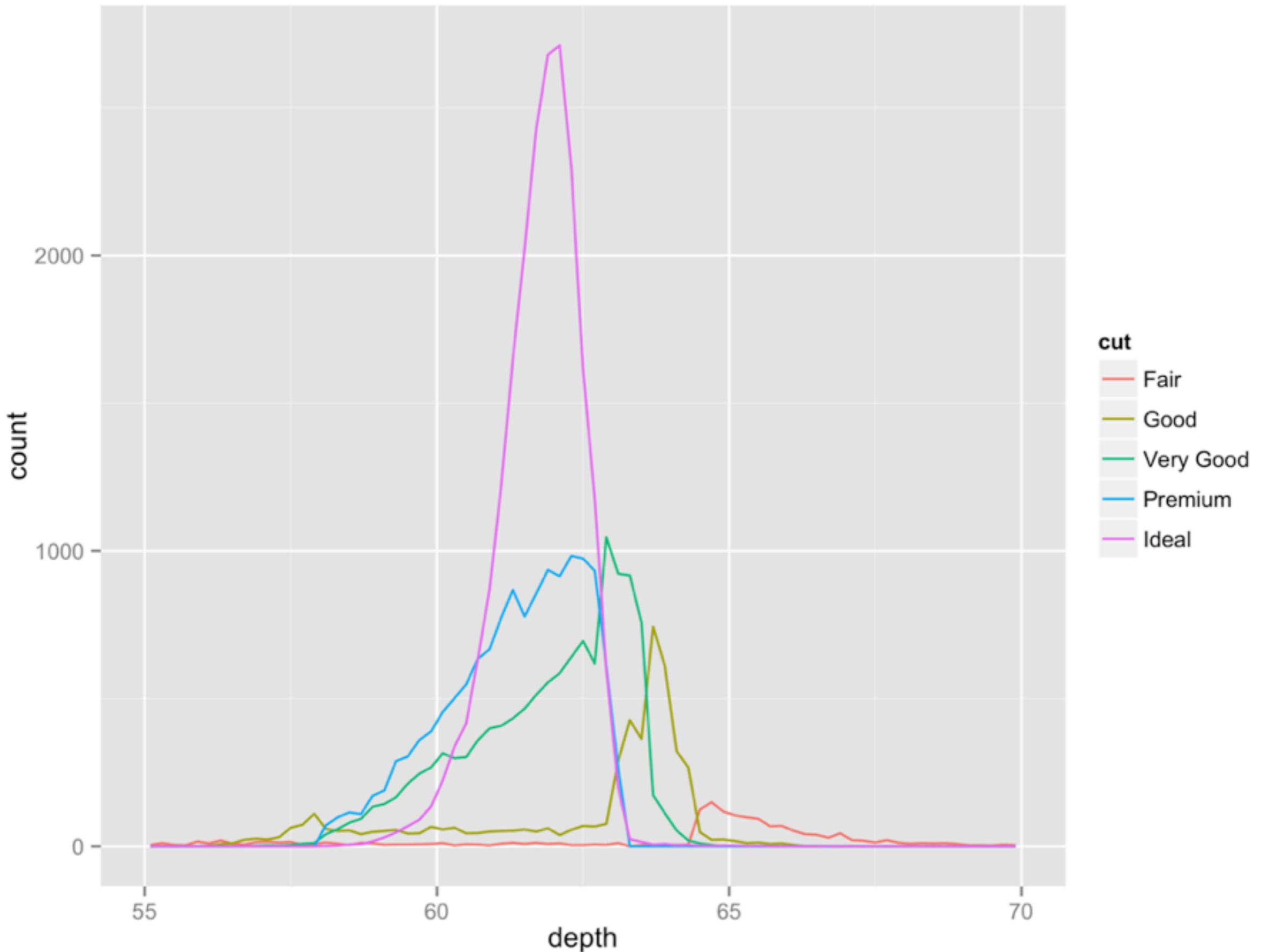
```
qplot(depth, data = diamonds, binwidth = 0.2) +  
  facet_wrap(~ cut)
```



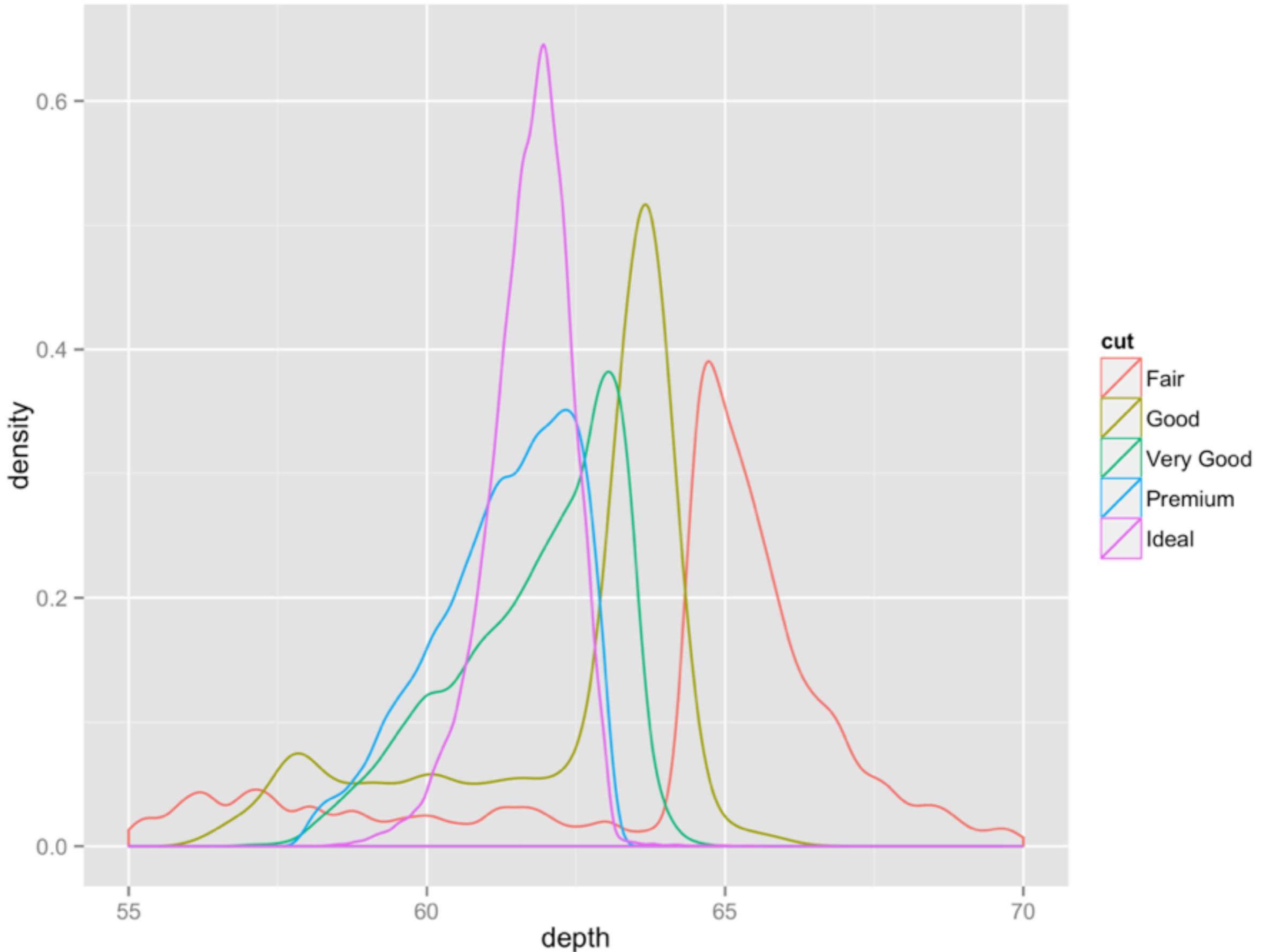
What if we just drew a line along the tops of the histograms, and threw away the bars?



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



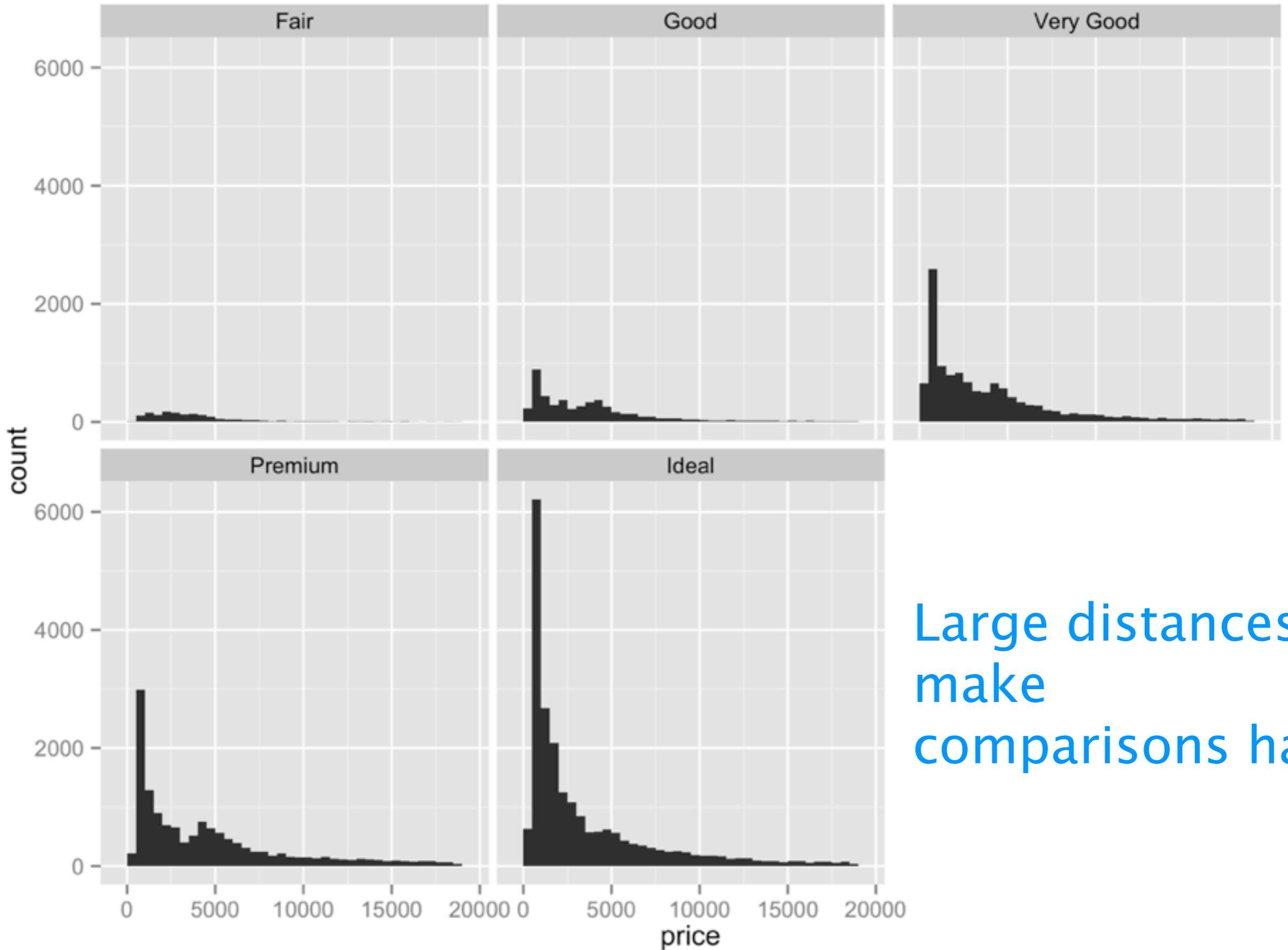
```
qplot(depth, data = diamonds, geom = "freqpoly",
      color = cut, binwidth = 0.2)
```



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

Your turn

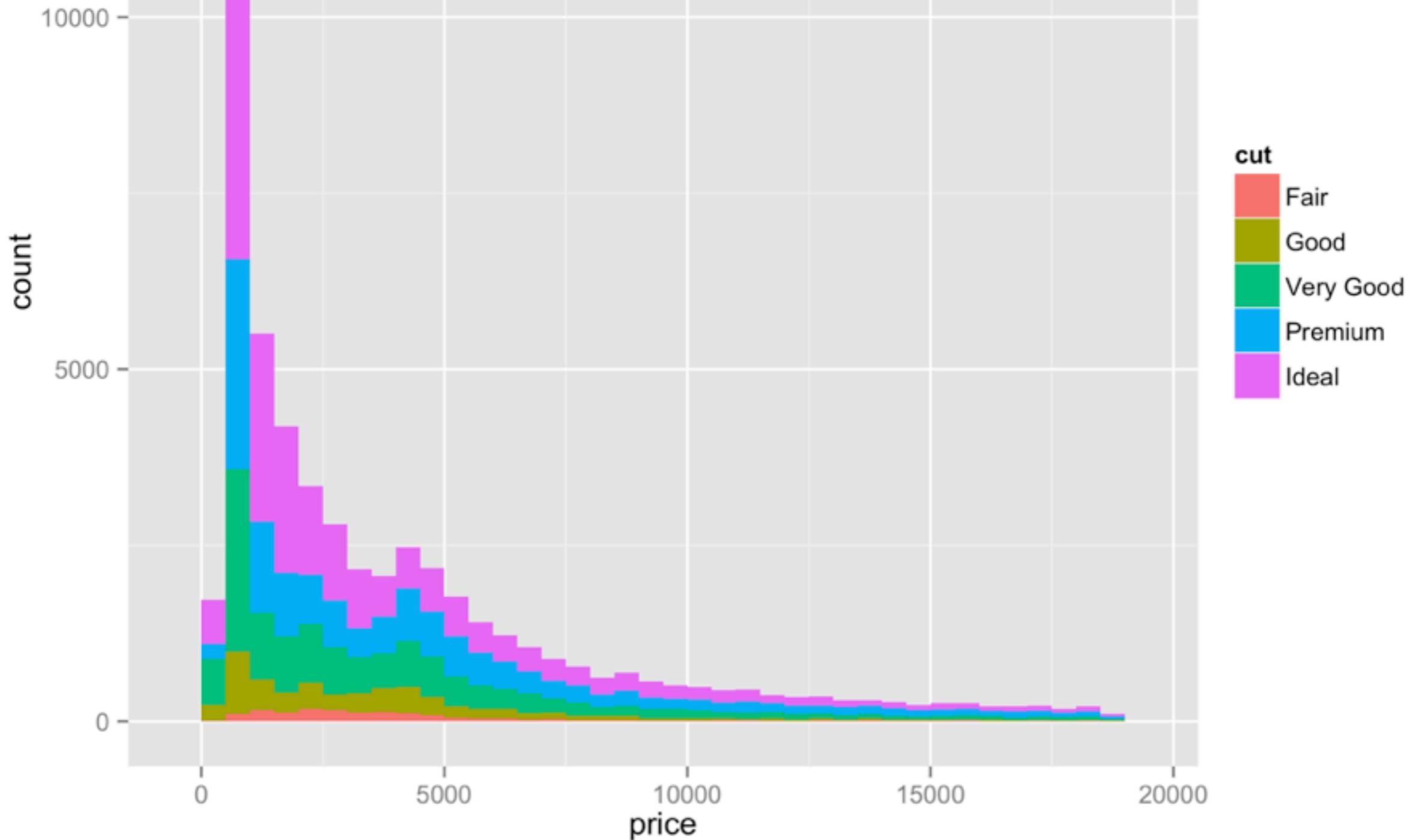
Compare the distribution of price for the different cuts. Does anything seem unusual?



Large distances
make
comparisons hard

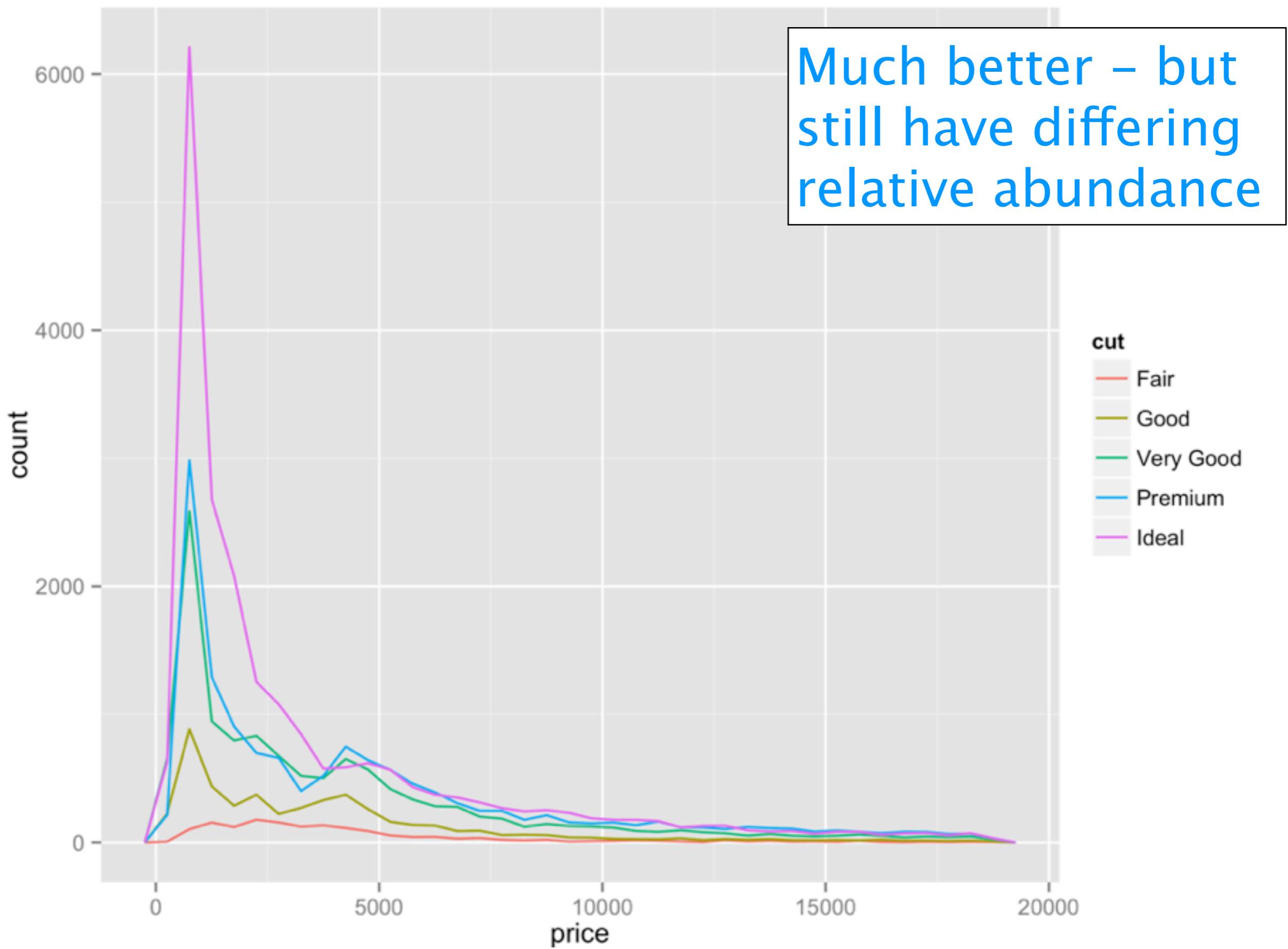
```
qplot(price, data = diamonds, binwidth = 500) +  
  facet_wrap(~ cut)
```

Stacked heights
hard to compare

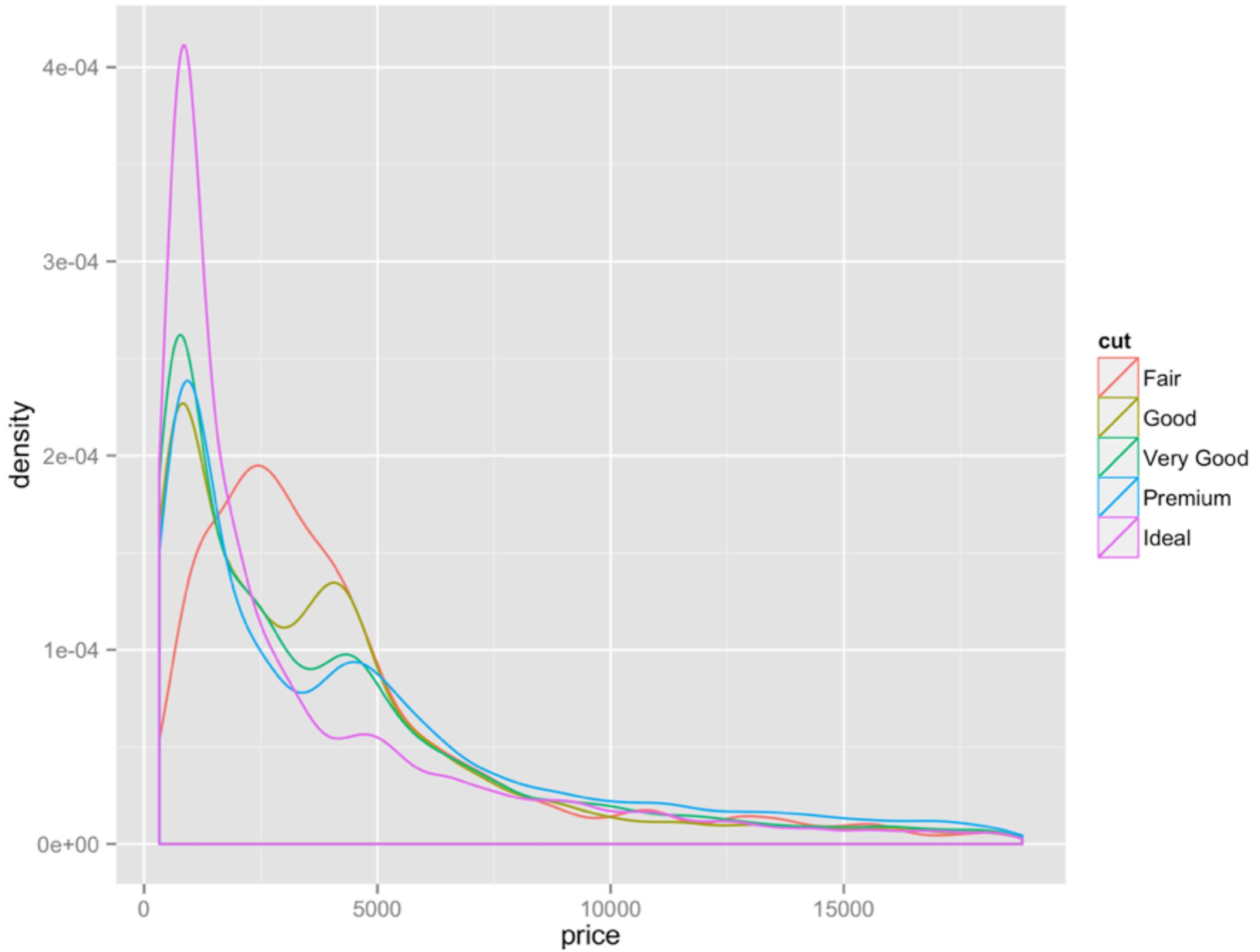


```
qplot(price, data = diamonds, binwidth = 500,  
fill = cut)
```

Much better – but
still have differing
relative abundance



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

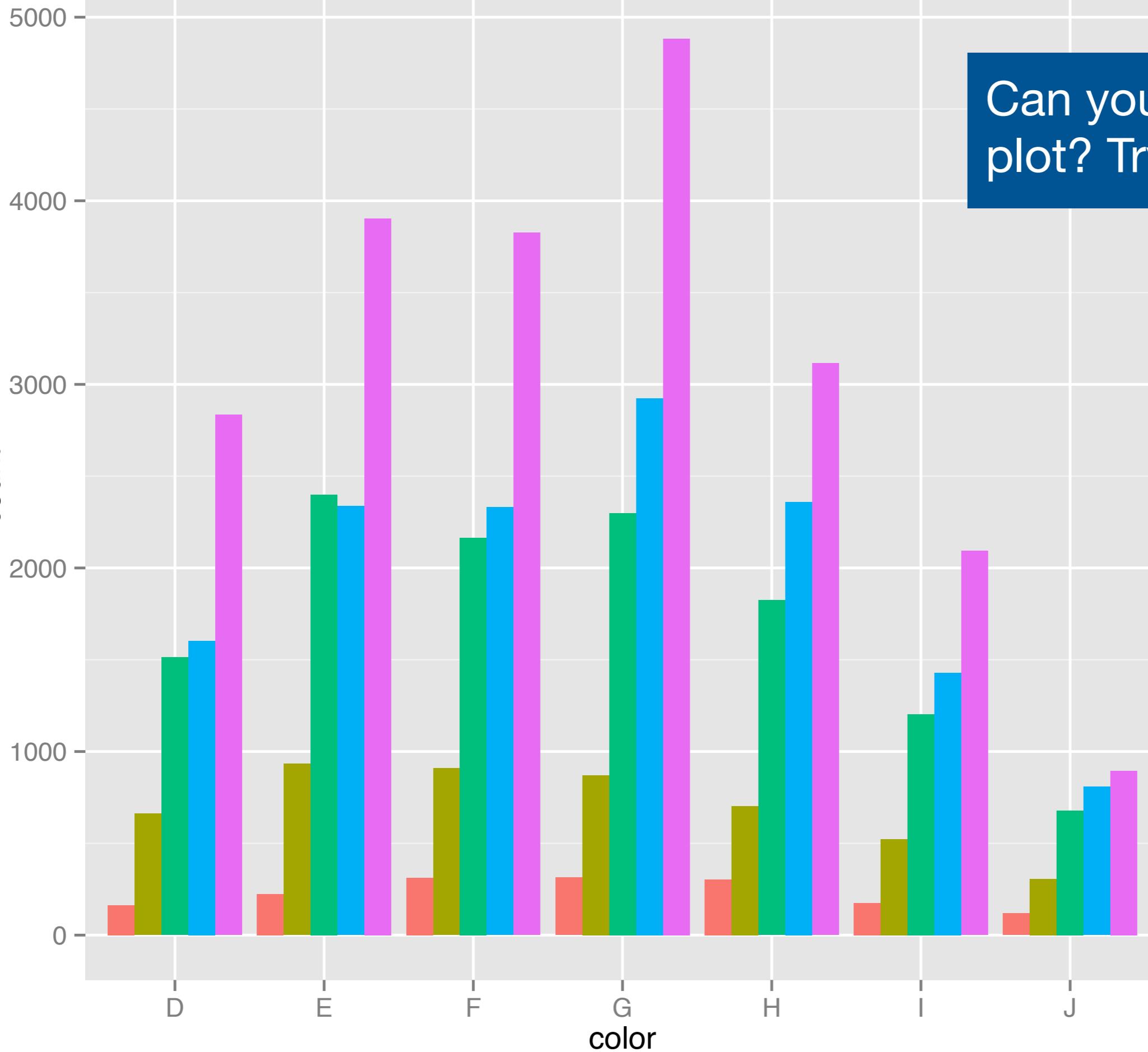


```
qplot(price, data = diamonds, geom = "density",  
color = cut)
```

Position adjustments

Can you make this
plot? Try it.

count



Your turn

What do each of the position adjustments below do?

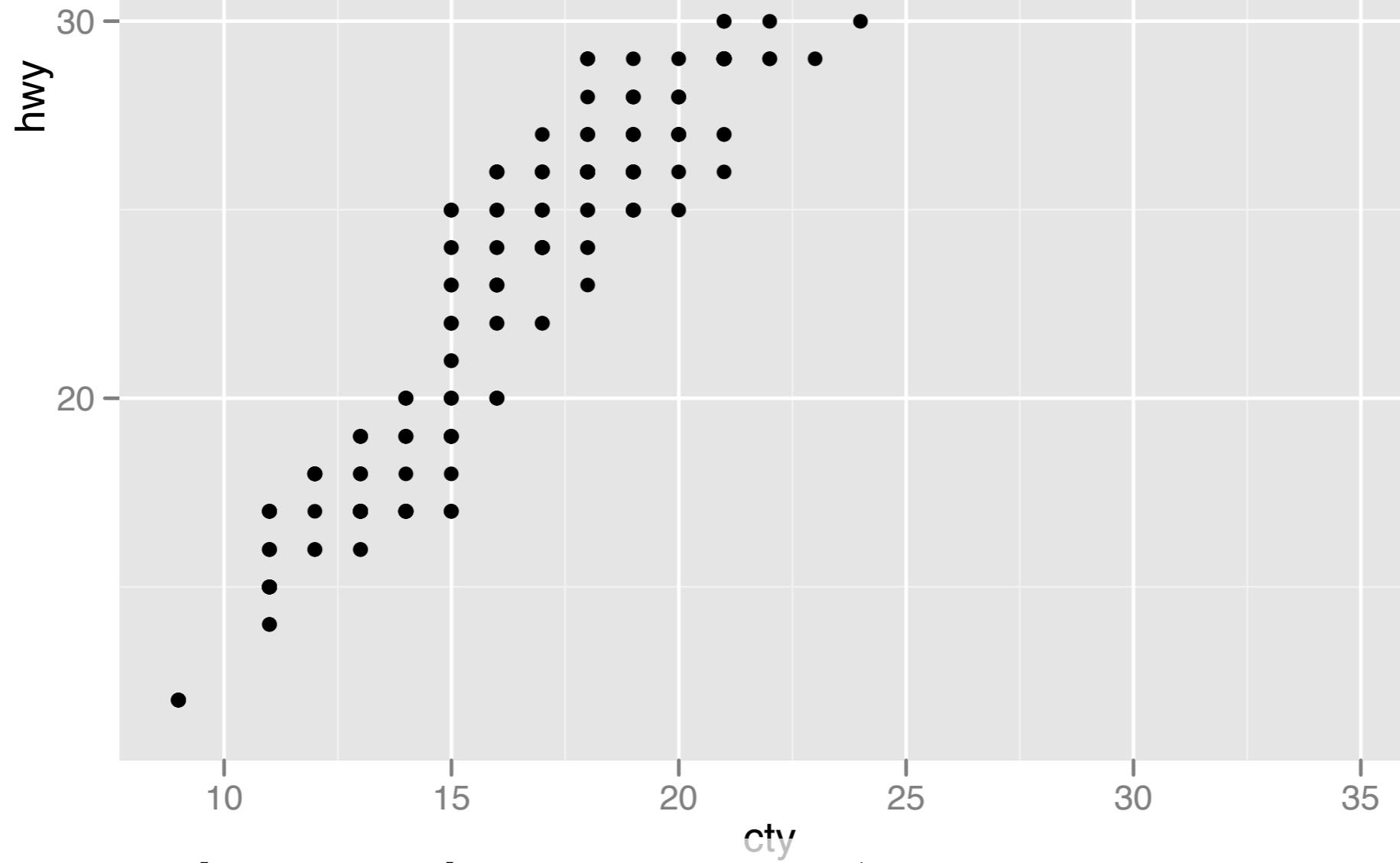
```
qplot(color, data = diamonds, fill = cut,  
position = "stack")
```

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

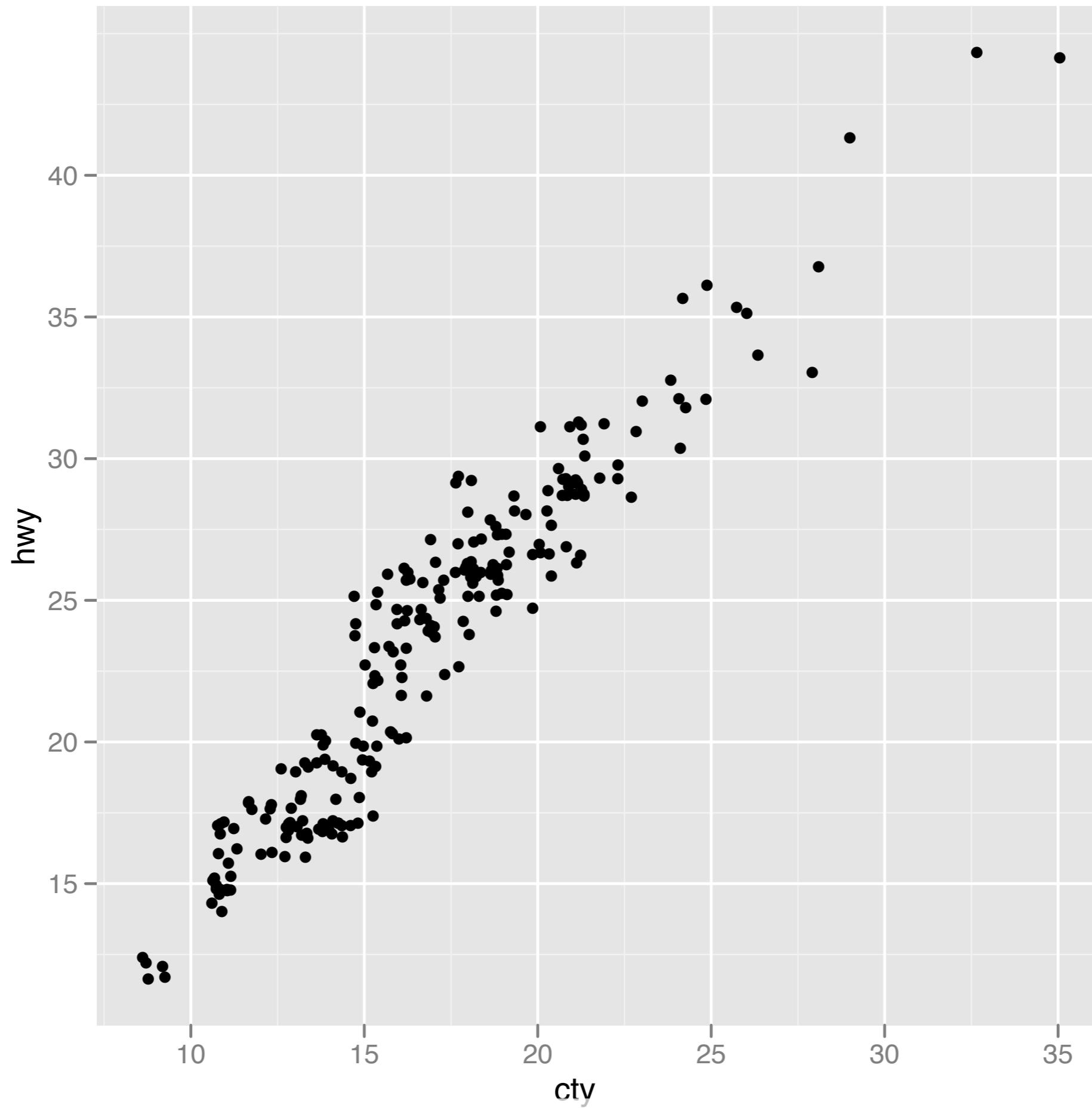
```
qplot(color, data = diamonds, fill = cut,  
position = "identity")
```

```
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 <i>above</i> each other
dodge	colliding objects plotted <i>beside</i> 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

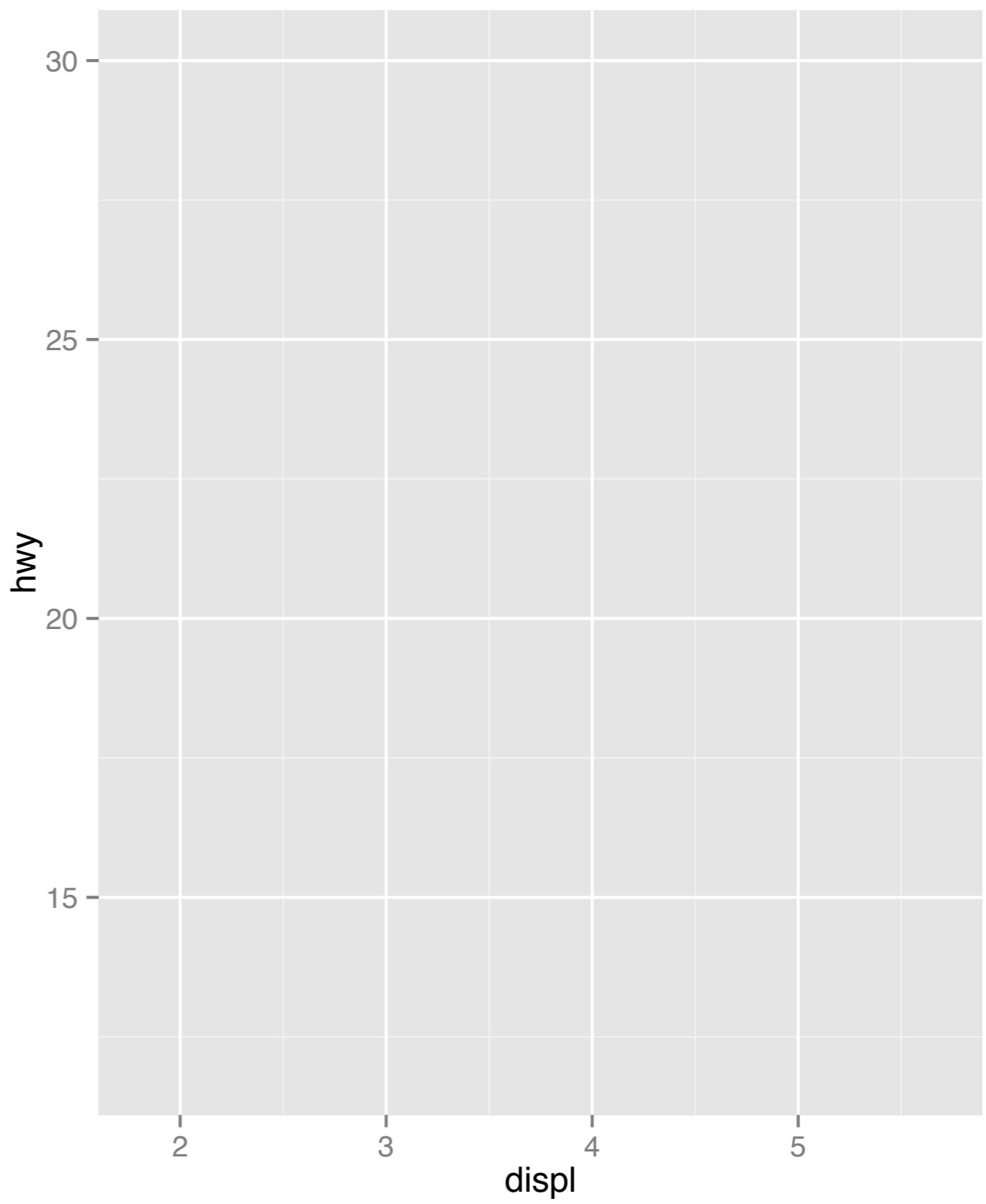
qplot + geoms + aesthetics + position adj = 100,000's

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

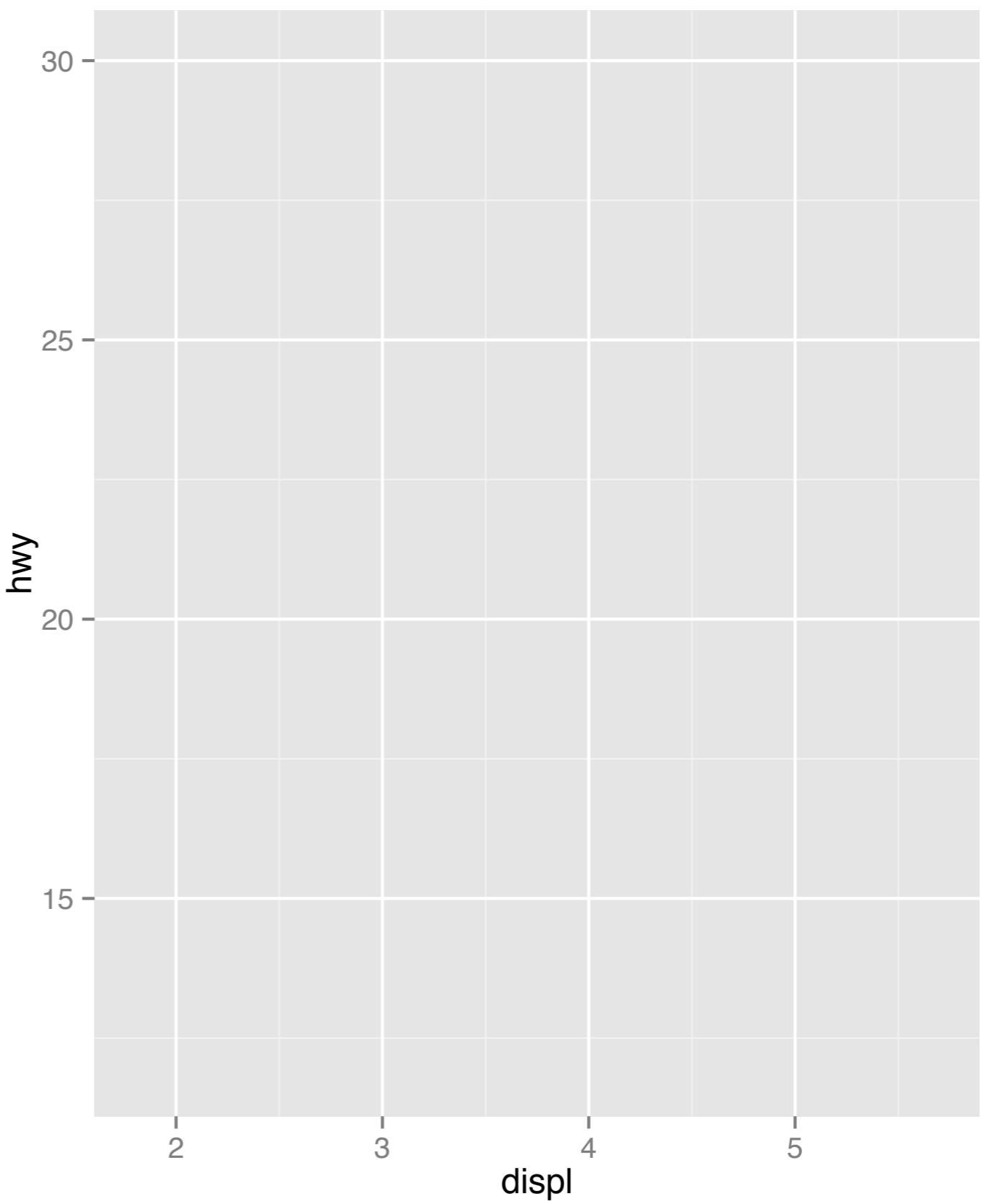
How to build a plot



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

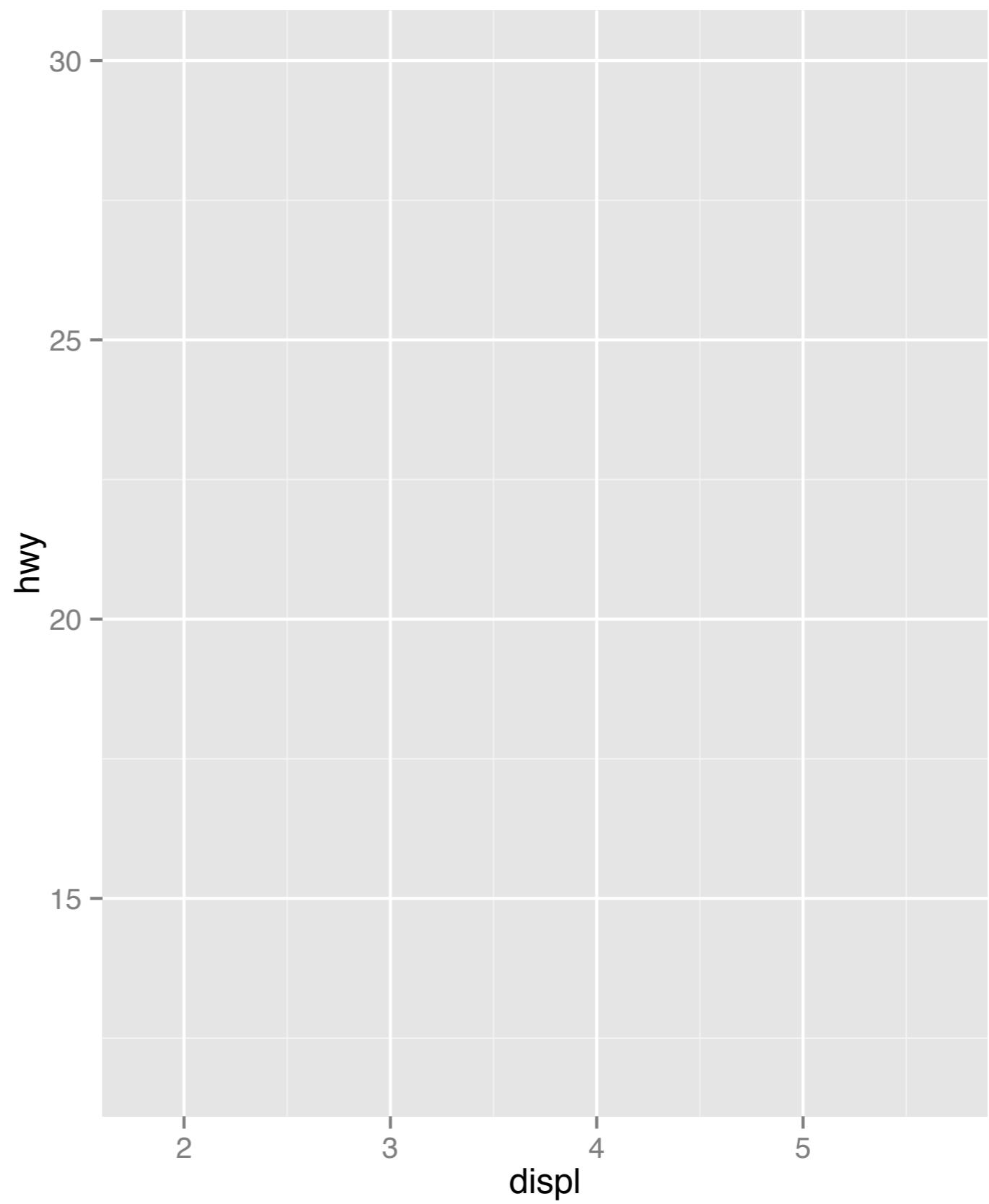


Coordinate system

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

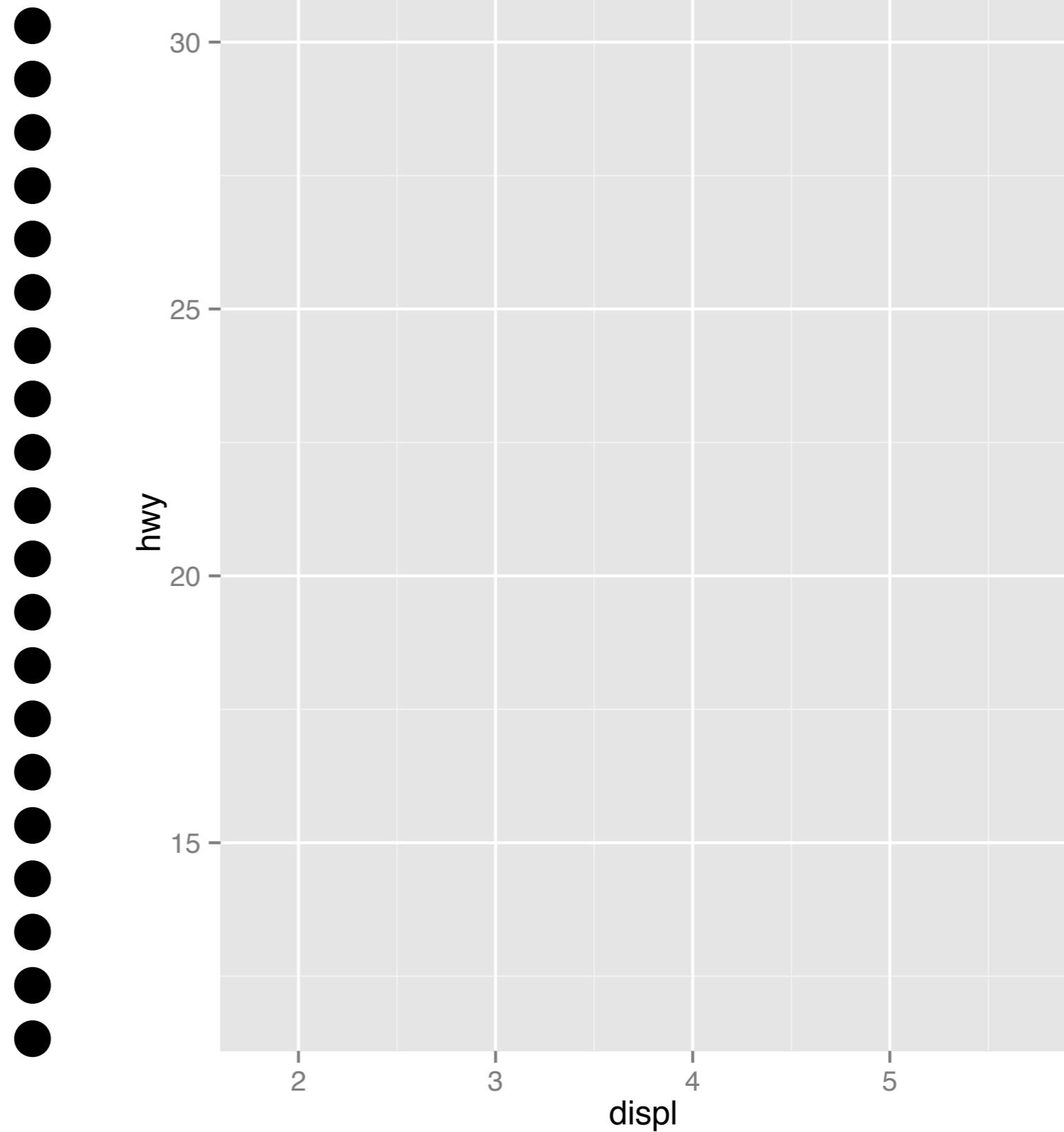
Geom



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



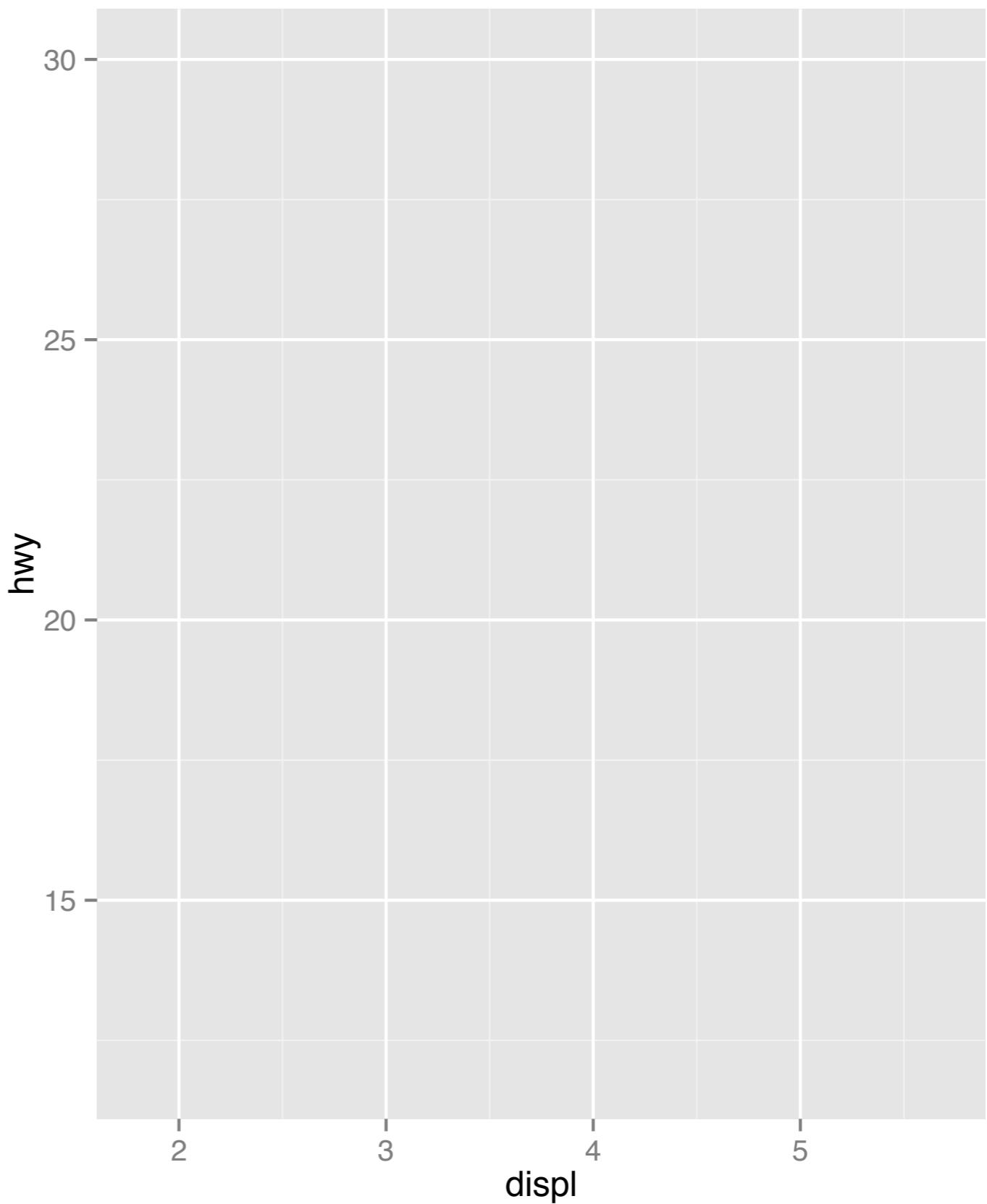
Aesthetic mappings

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



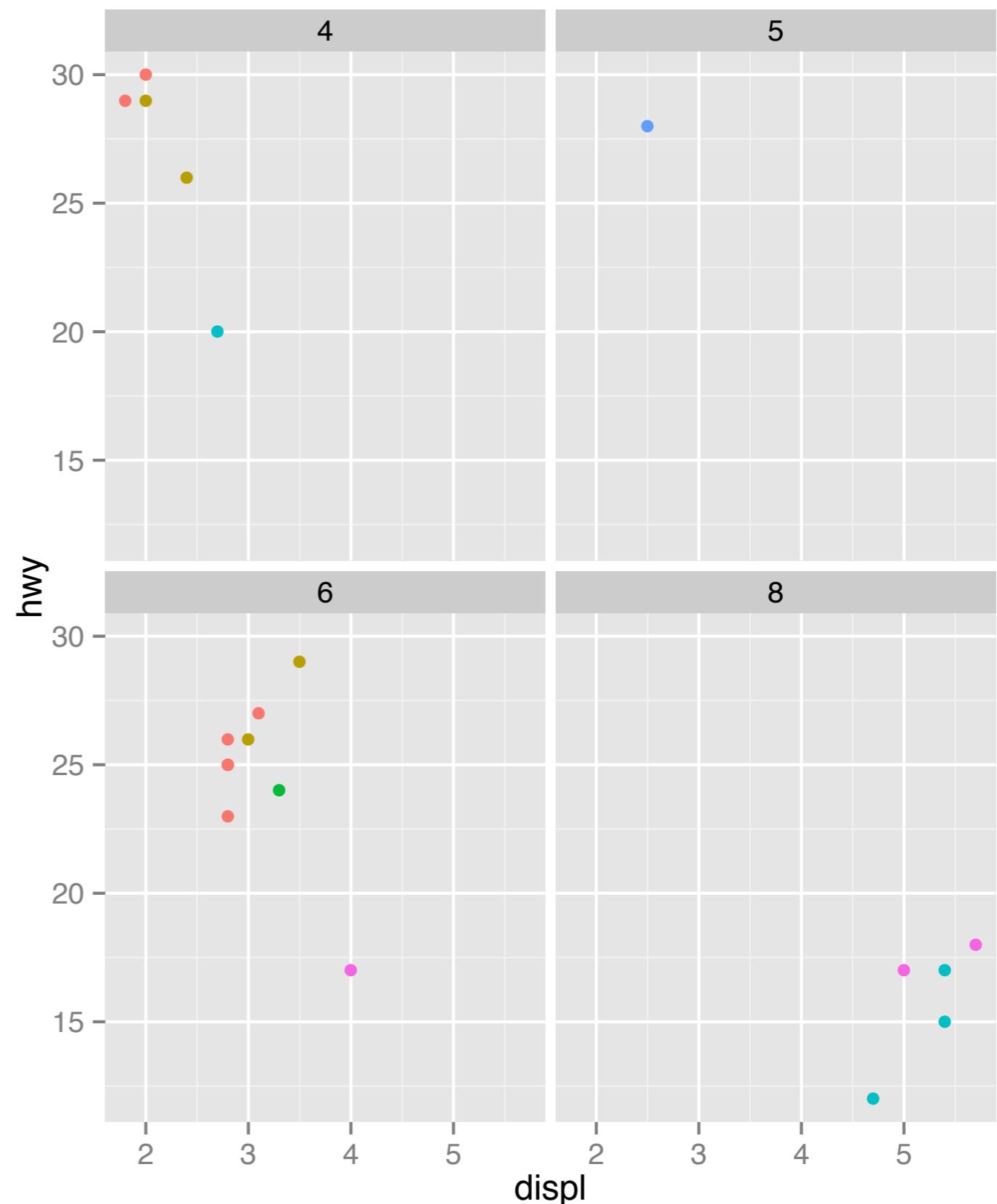
Aesthetic mappings

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

Facet (or not)



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: $x = \text{date}$, $y = \text{unemploy}$

Plot 2

data = mpg

geom = point

position = jitter

aesthetic mappings: $x = \text{class}$, $y = \text{hwy}$, $\text{color} = \text{class}$

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

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

Index. ggplot2 0.9.2.1

docs.ggplot2.org/current/

ggplot2 0.9.2.1 Index

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
- [geom_errorbar](#)

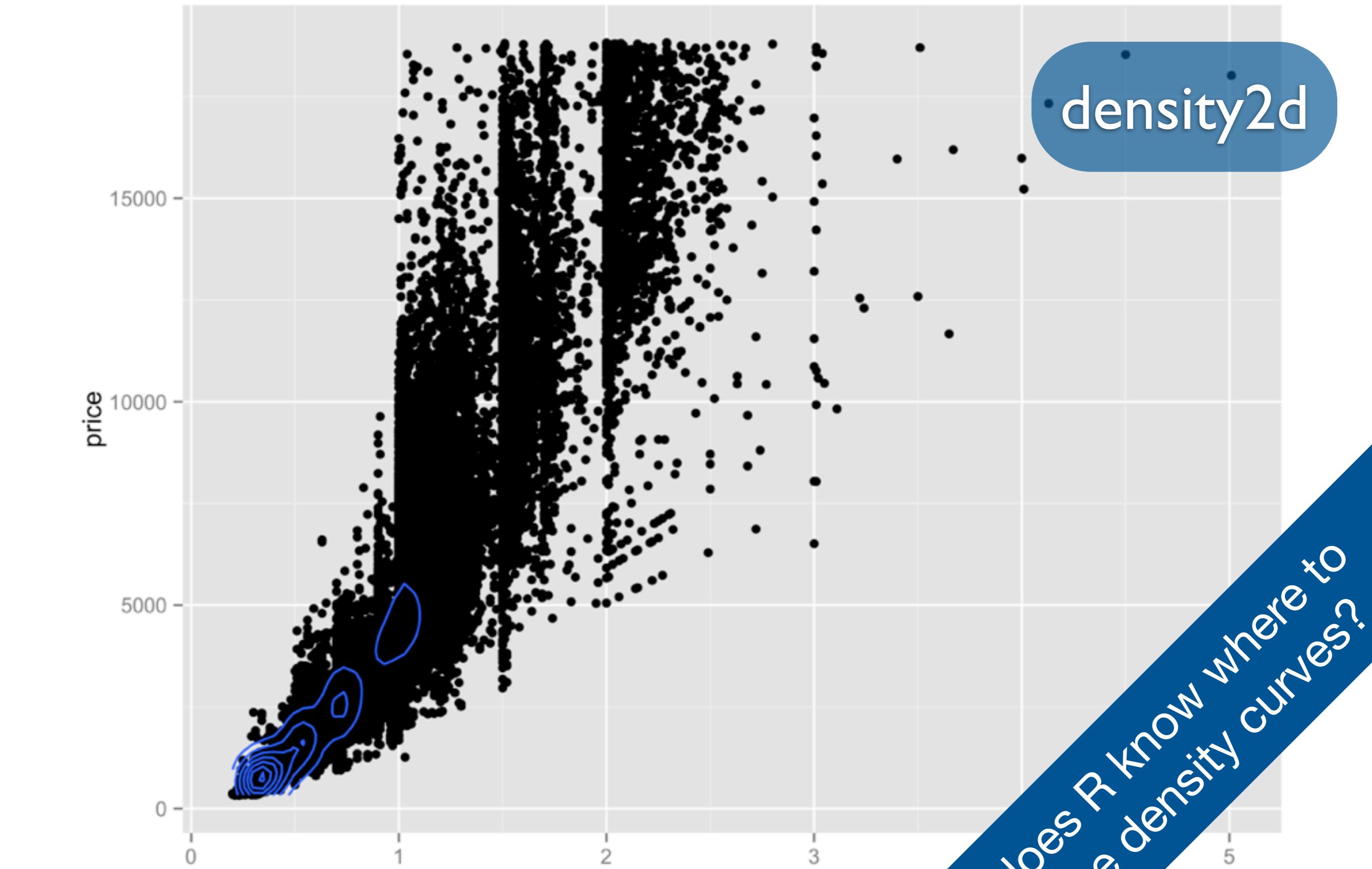


Dependencies

- **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/>

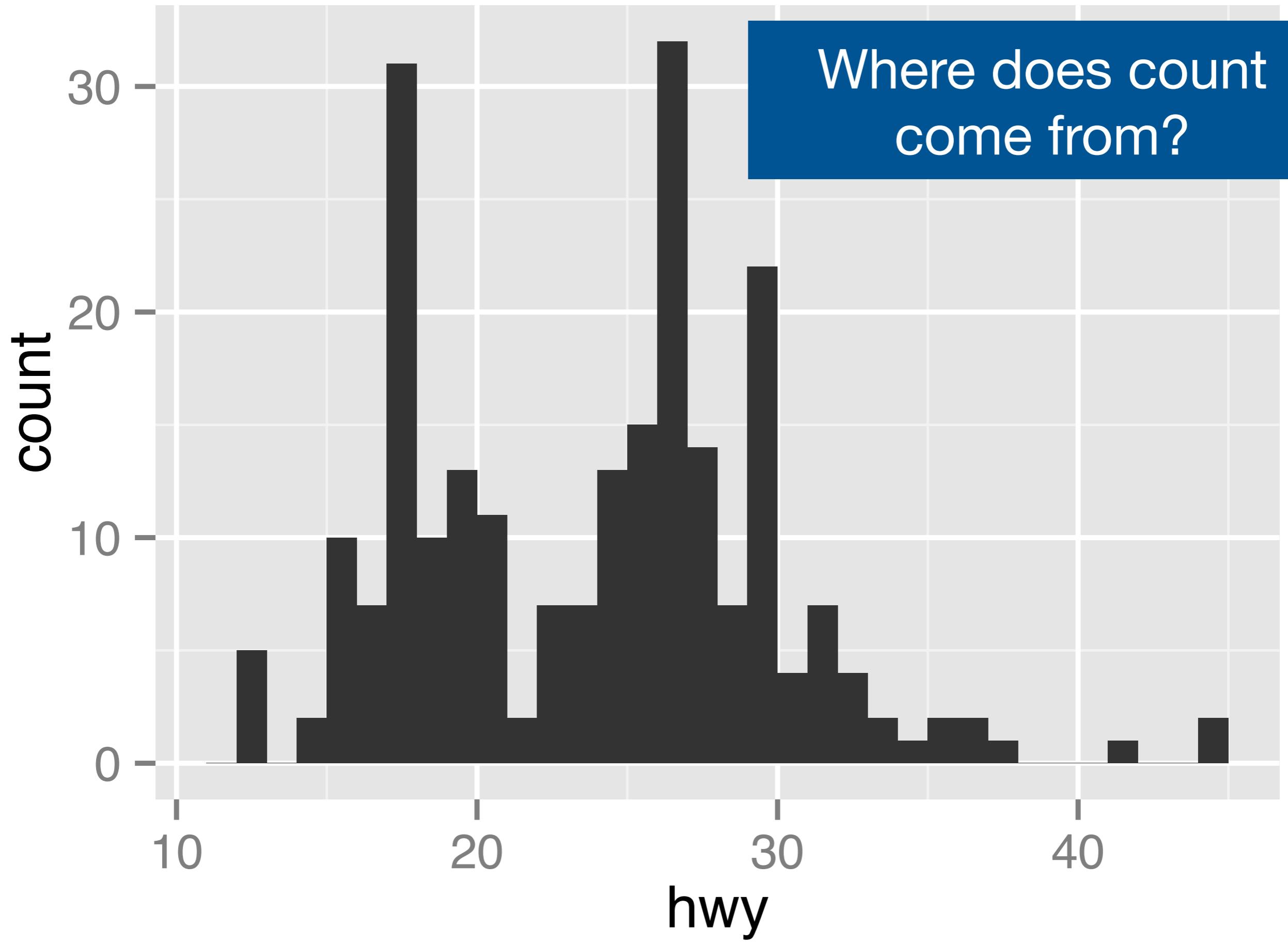


```
qplot(carat, price, data = diamonds,  
geom = c("point", "density2d"))
```

How does R know where to
draw the density curves?

density2d

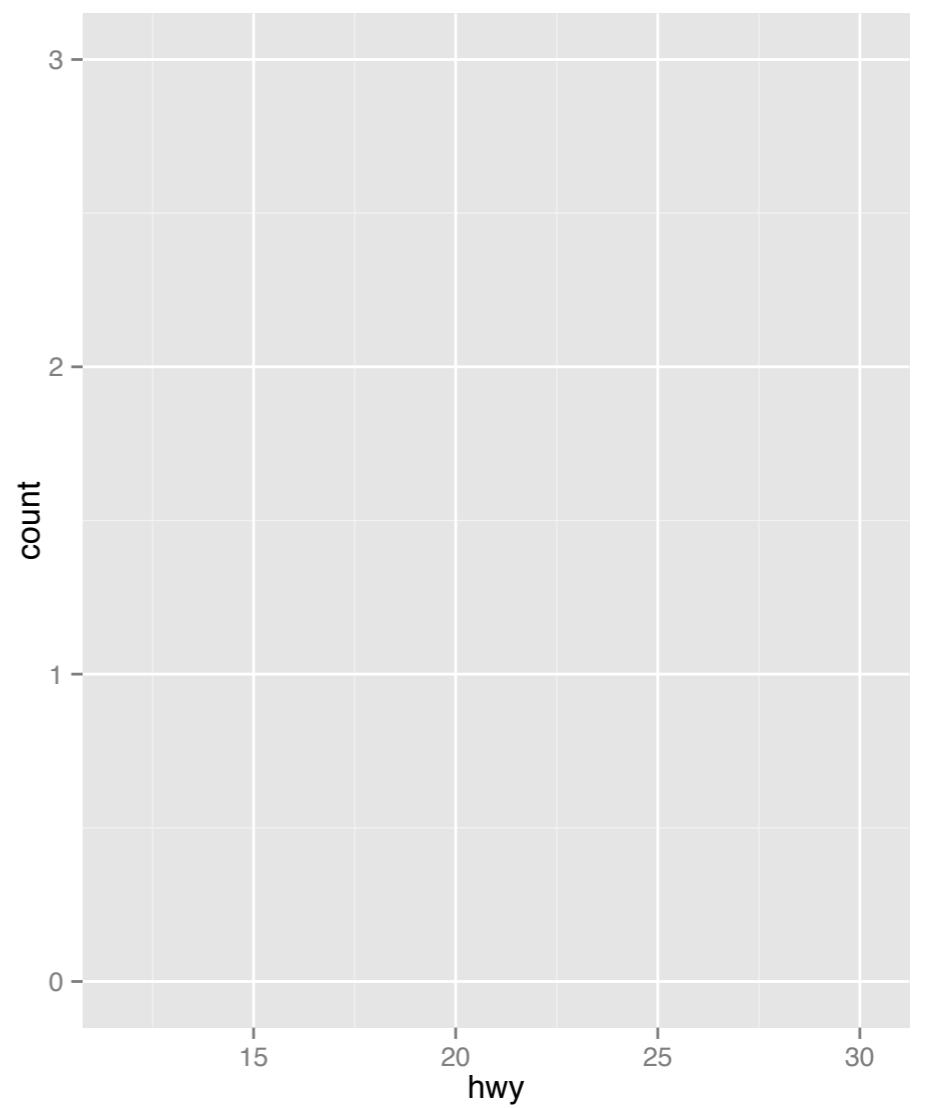
Where does count
come from?



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

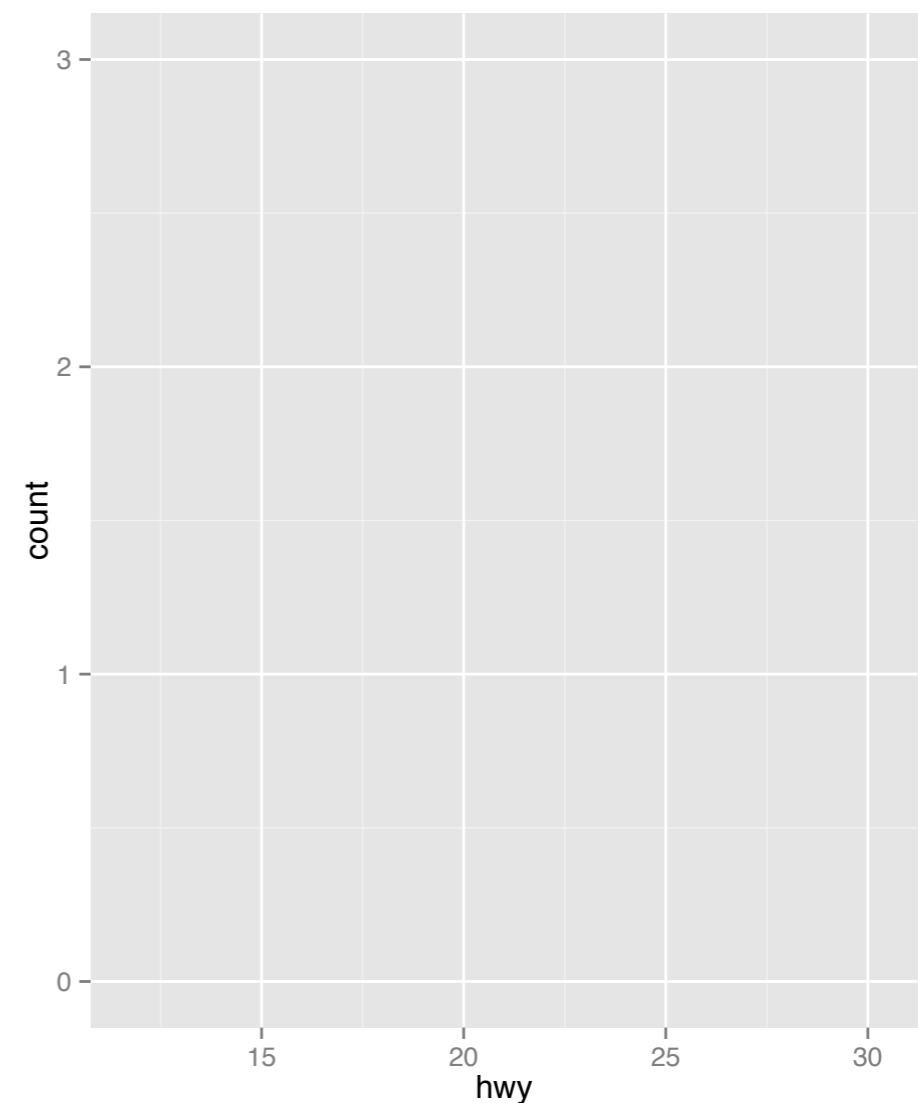
How to build a plot 2

Stats



Coordinate system

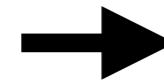
hwy	displ	cyl	class
12	4.7	8	pickup
15	5.4	8	pickup
17	5.0	8	suv
17	4.0	6	suv
17	5.4	8	pickup
18	5.7	8	suv
20	2.7	4	pickup
23	2.8	6	compact
24	3.3	6	minivan
25	2.8	6	compact
25	2.8	6	compact
26	3.0	6	midsize
26	2.4	4	midsize
26	2.8	6	compact
27	3.1	6	compact
28	2.5	5	subcomp
29	3.5	6	midsize
29	2.0	4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



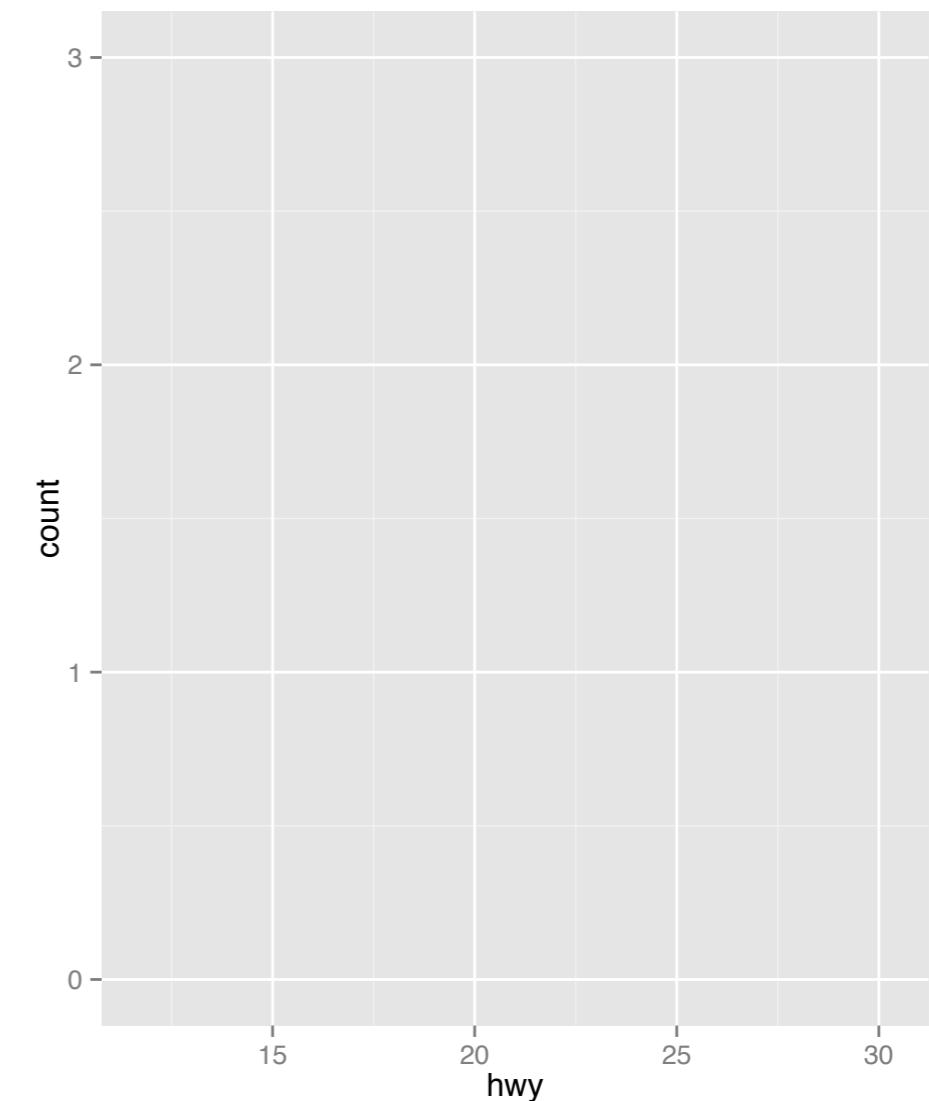
Data

Coordinate system

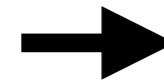
hwy	displ	cyl	class
12	4.7	8	pickup
15	5.4	8	pickup
17	5.0	8	suv
17	4.0	6	suv
17	5.4	8	pickup
18	5.7	8	suv
20	2.7	4	pickup
23	2.8	6	compact
24	3.3	6	minivan
25	2.8	6	compact
25	2.8	6	compact
26	3.0	6	midsize
26	2.4	4	midsize
26	2.8	6	compact
27	3.1	6	compact
28	2.5	5	subcomp
29	3.5	6	midsize
29	2.0	4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



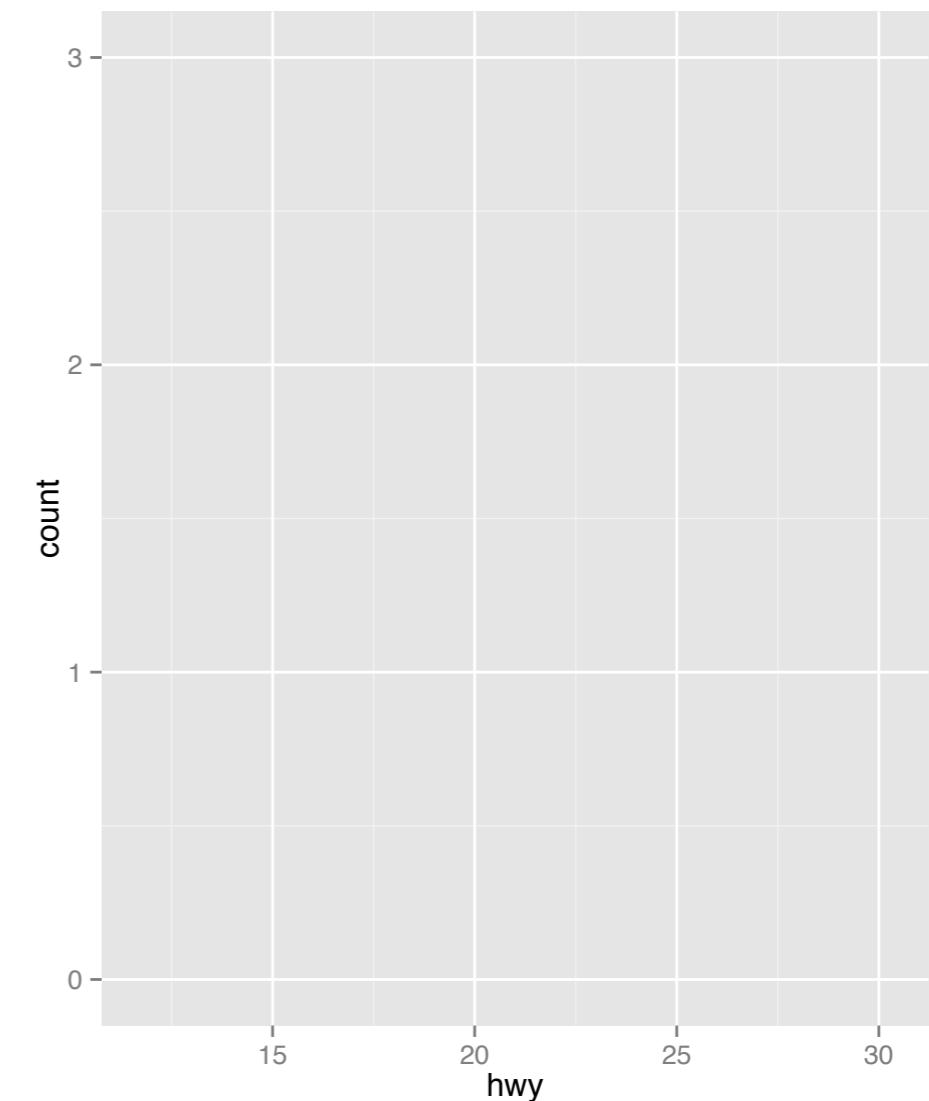
bin	hwy	count
1	12	1
2	15	1
3	17	3
4	18	1
5	20	1
6	23	1
7	24	1
8	25	2
9	26	3
10	27	1
11	28	1
12	29	3
13	30	1



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

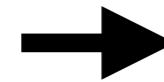
Stat

Geom

Coordinate system

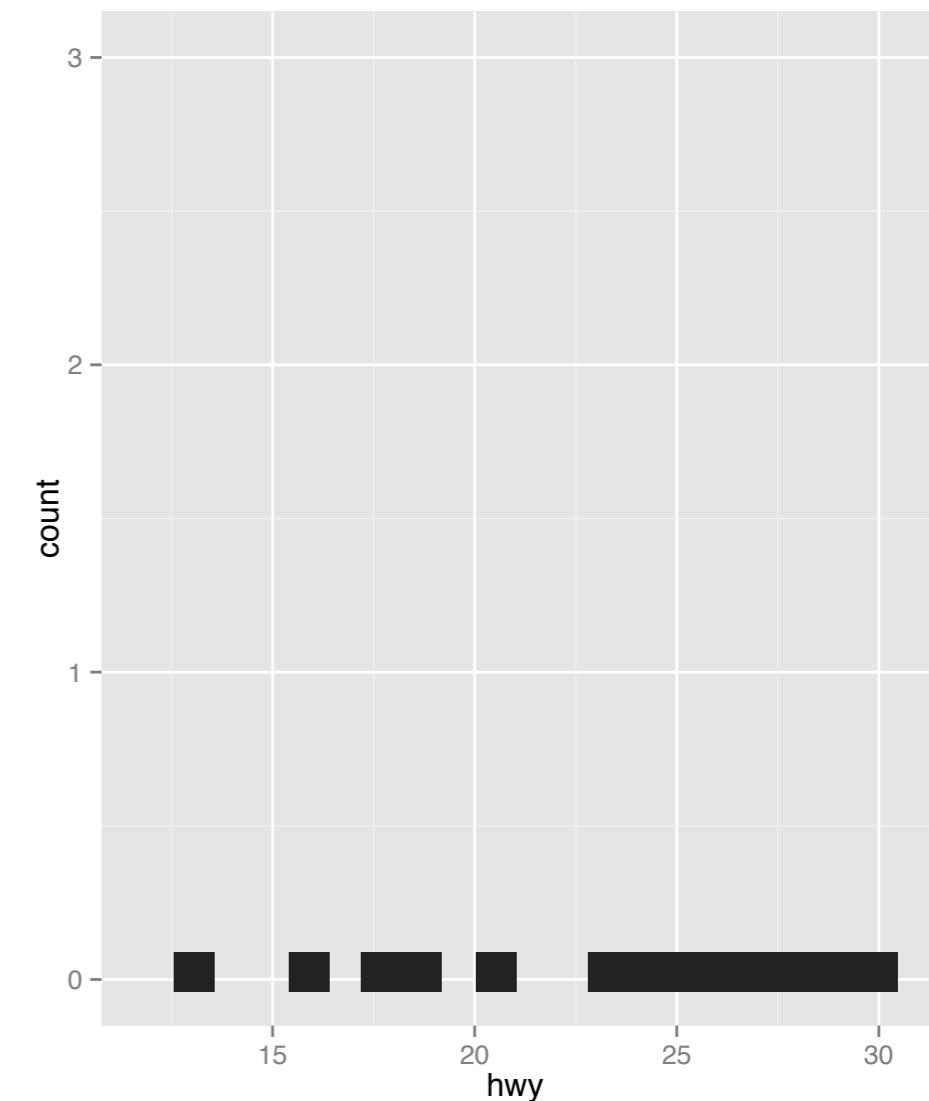
Aesthetic mappings

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

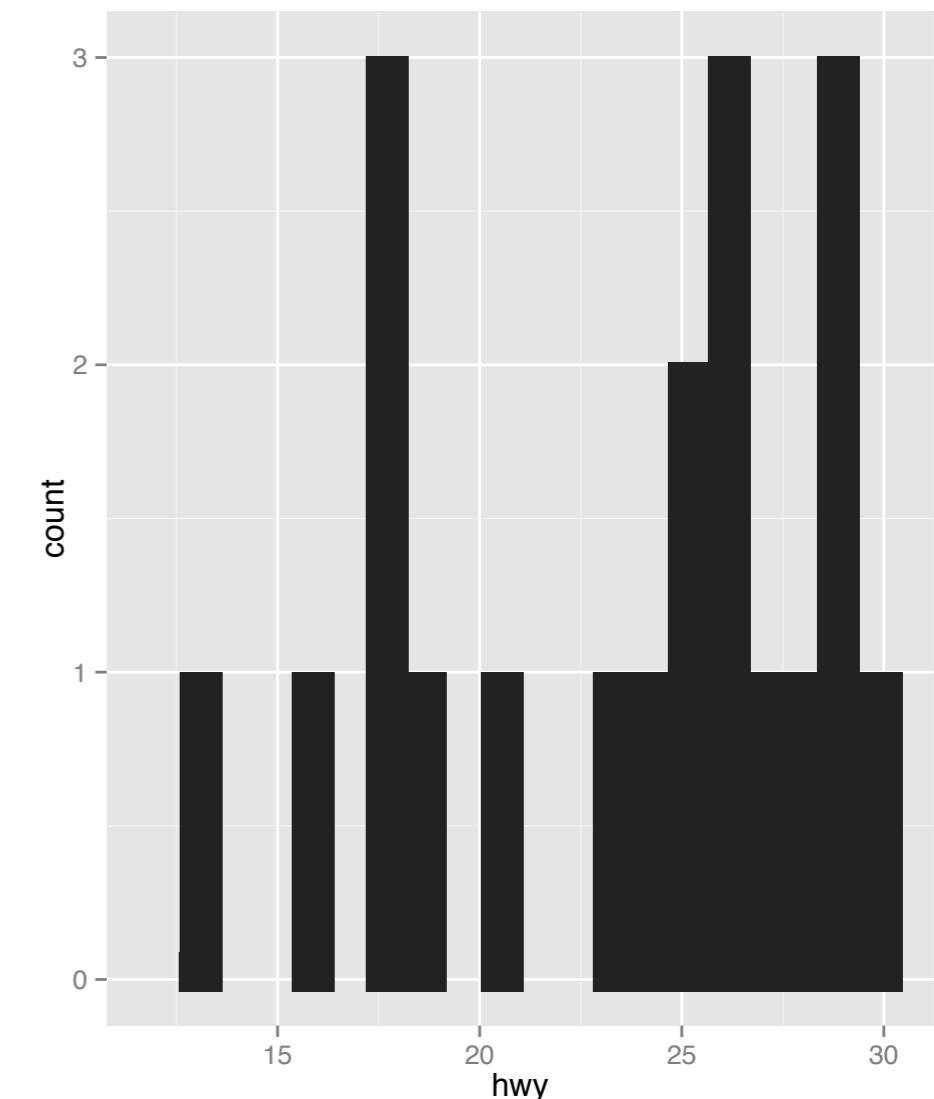
Coordinate system

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		
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

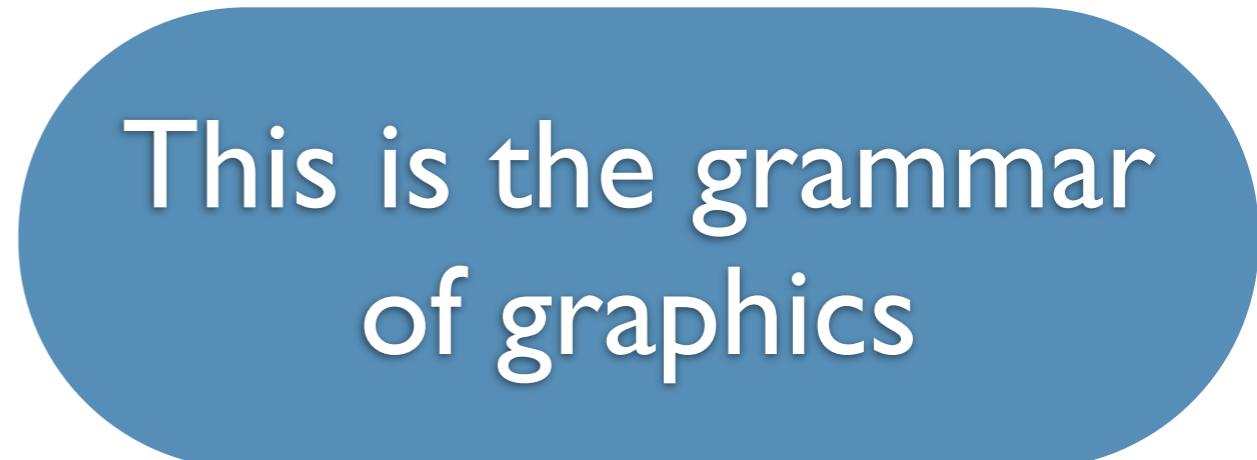
+ geom

+ data

+ aesthetic mappings

+ position adjustment

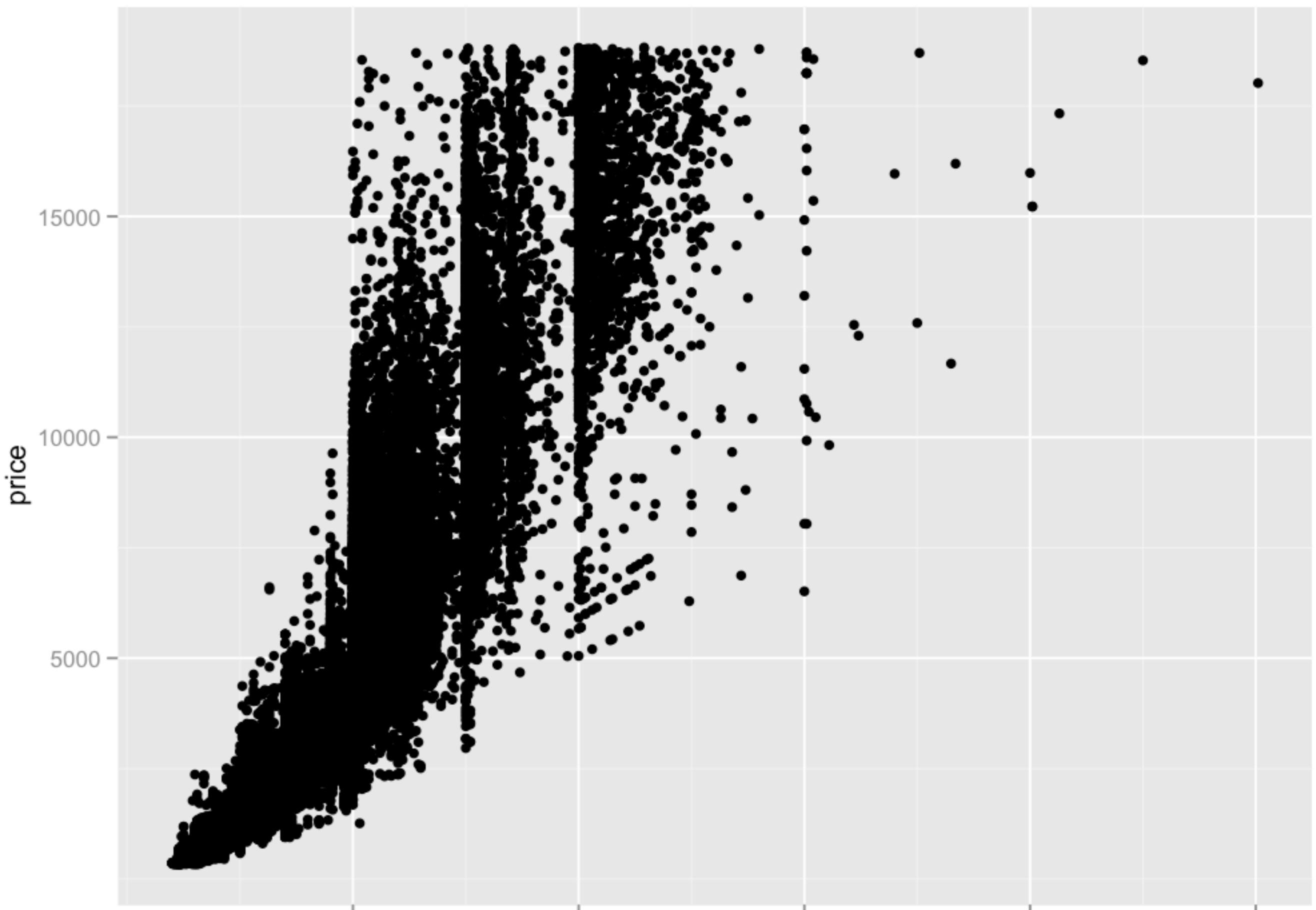
+ stat



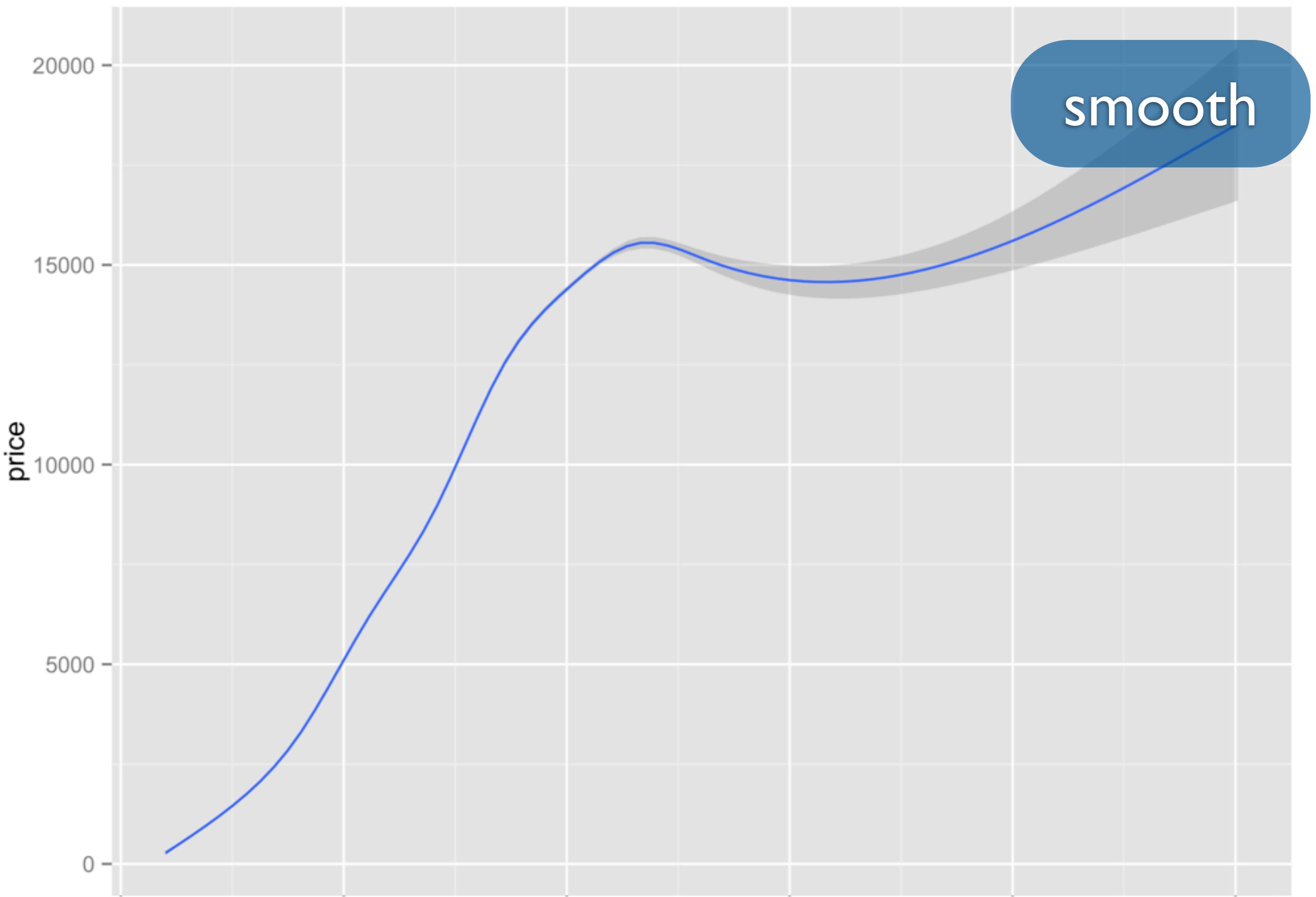
This is the grammar
of graphics

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

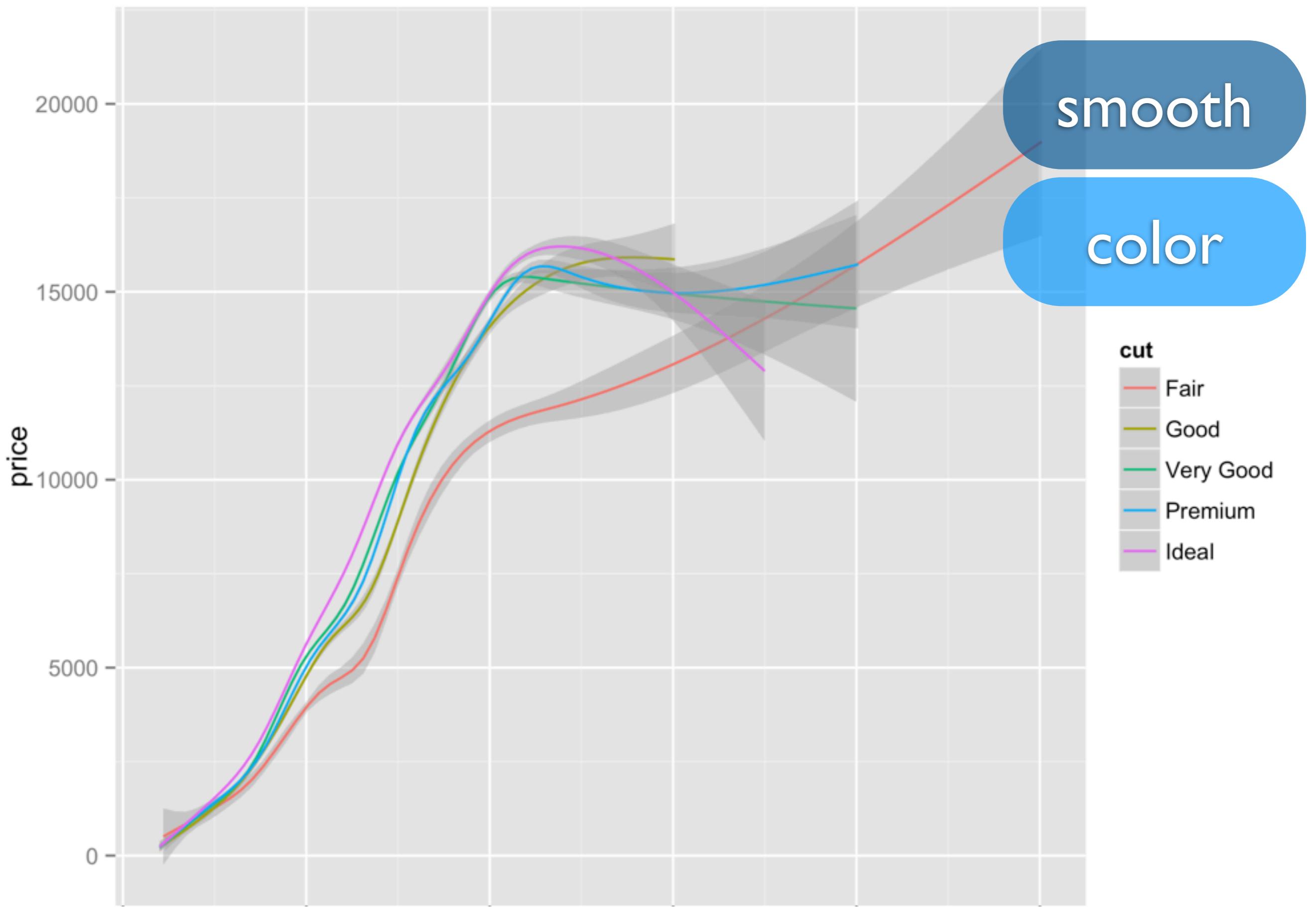
Geoms in ggplot2 know when
and how to use a stat



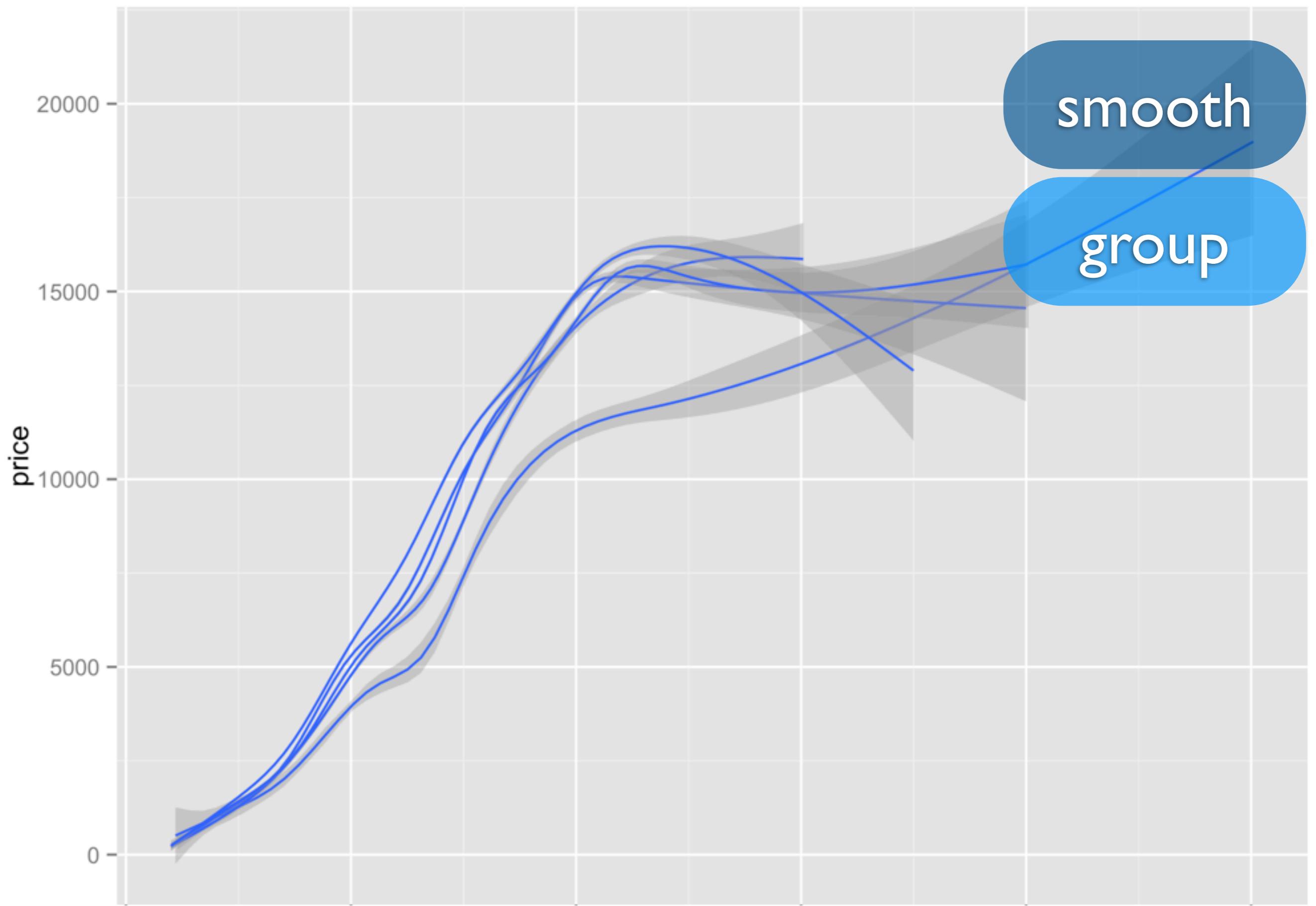
```
qplot(carat, price, data = diamonds)
```



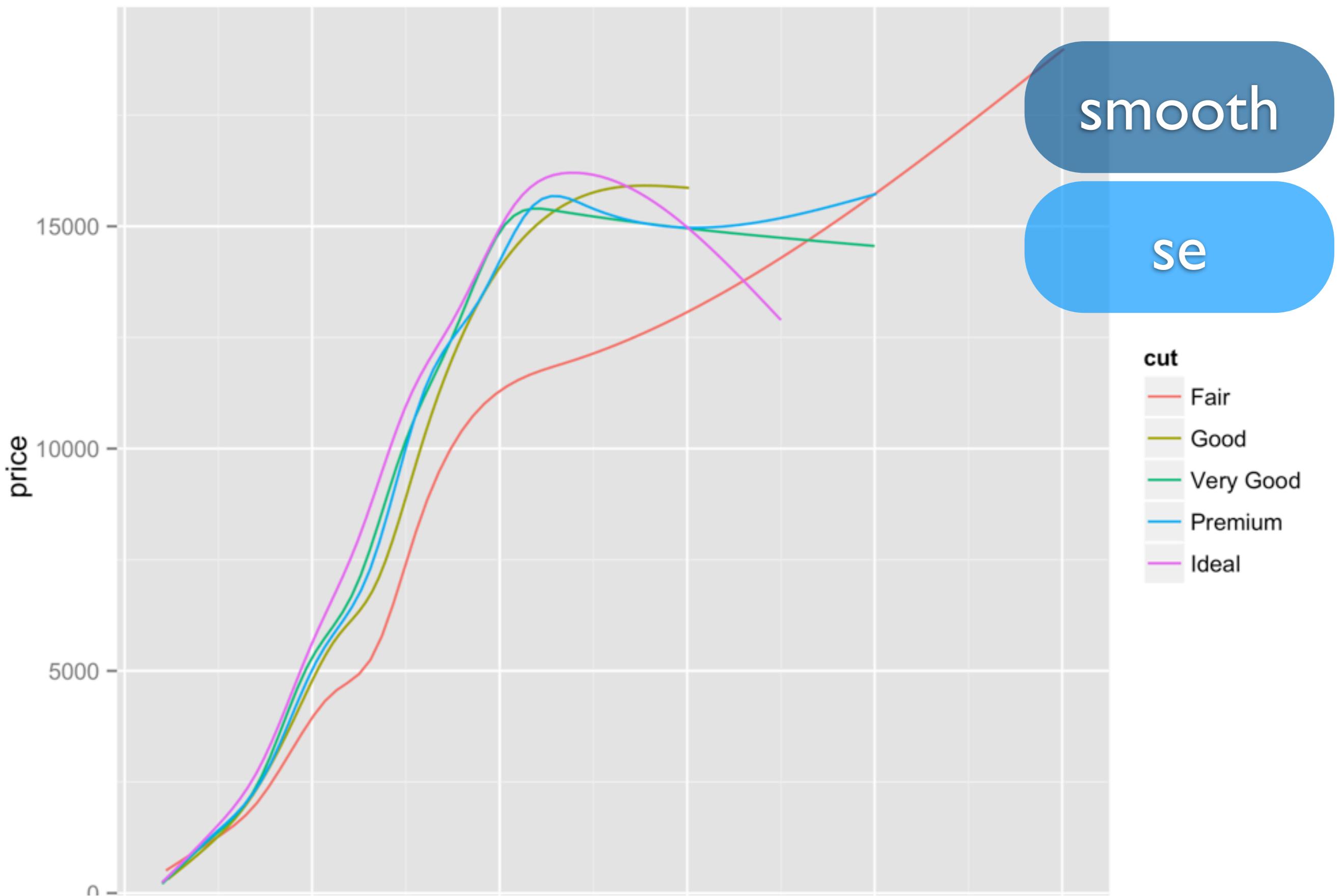
```
qplot(carat, price, data = diamonds, geom = "smooth")
```



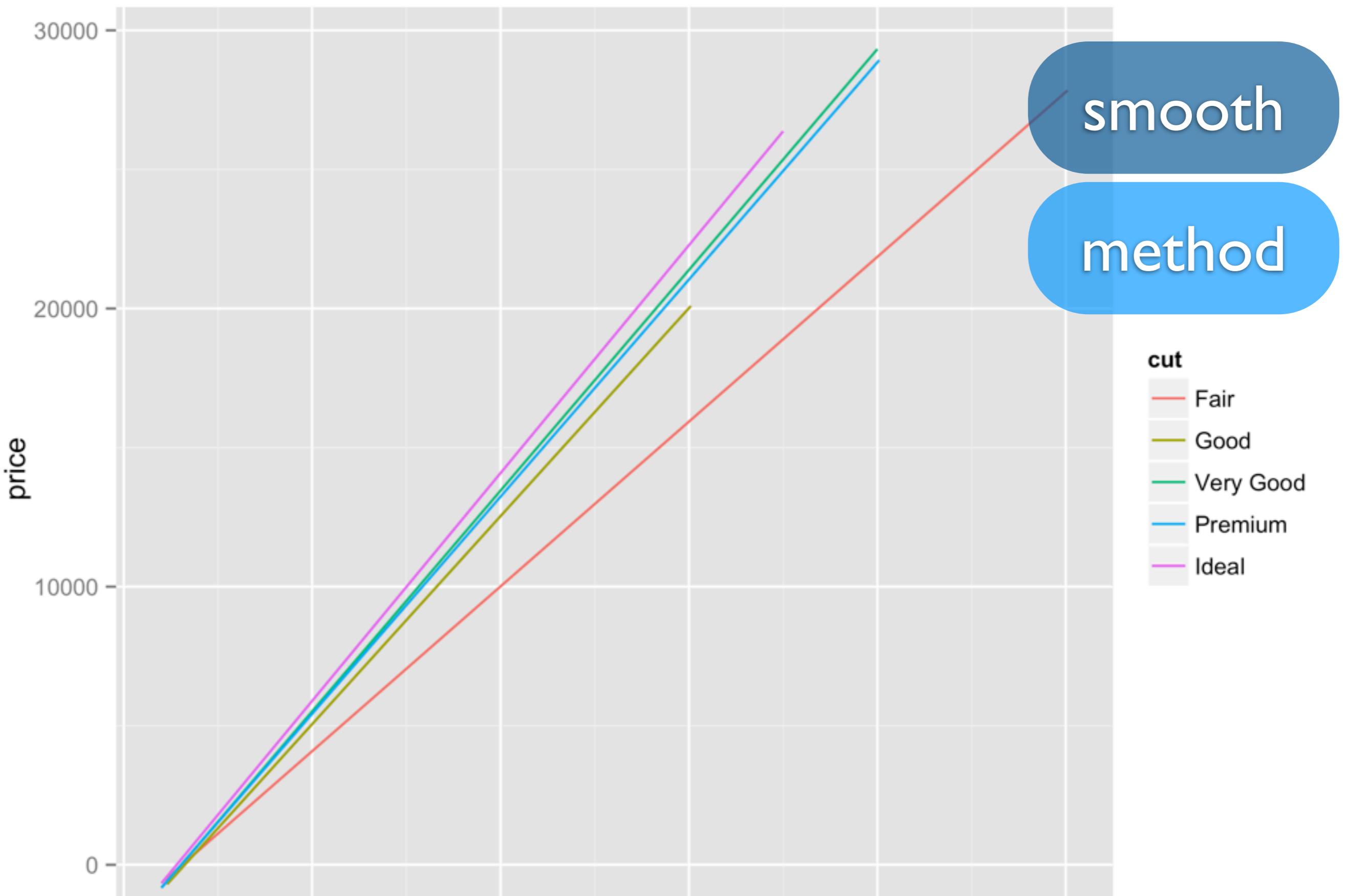
```
qplot(carat, price, data = diamonds, geom = "smooth", color = cut)
```



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



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

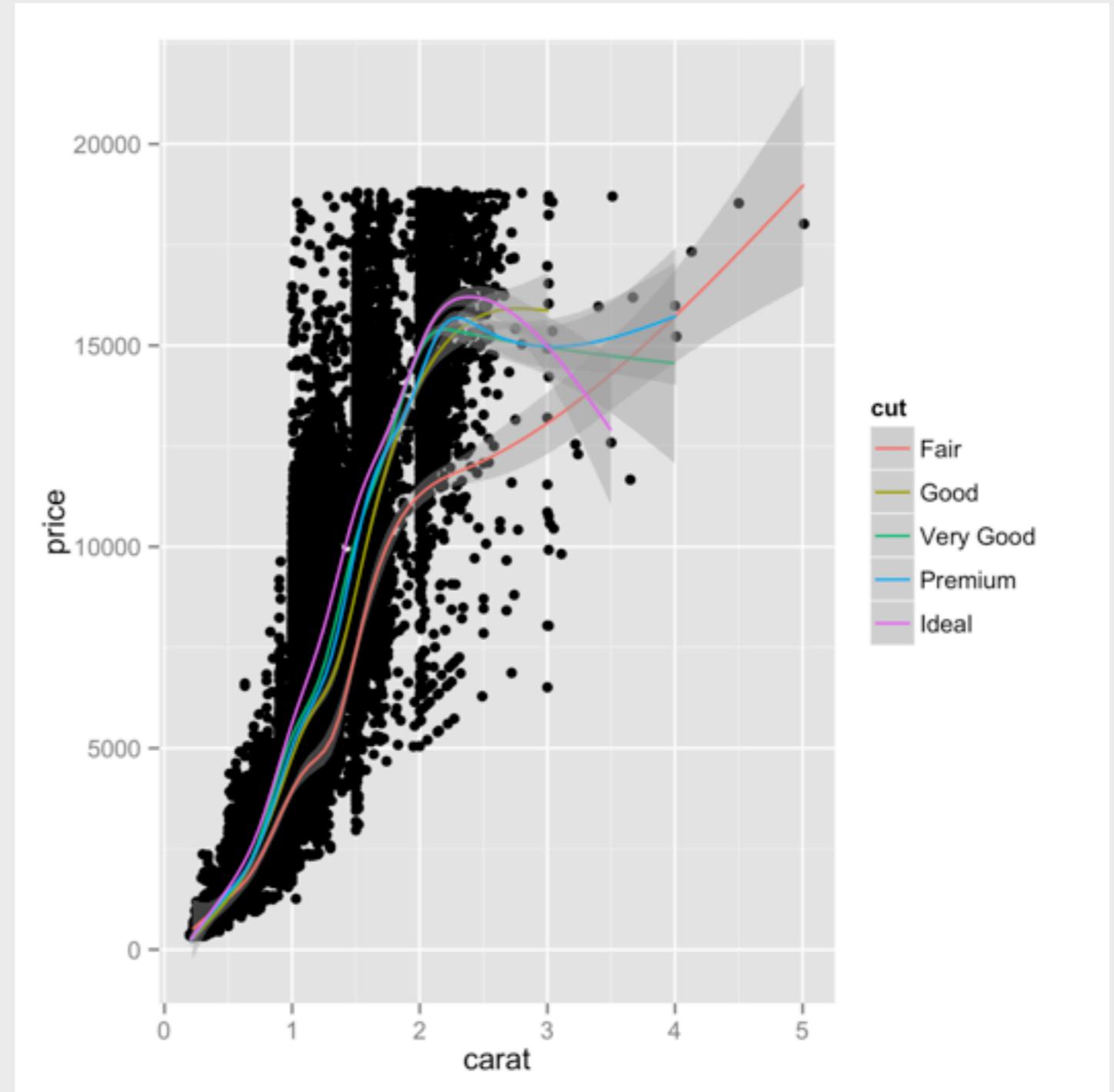


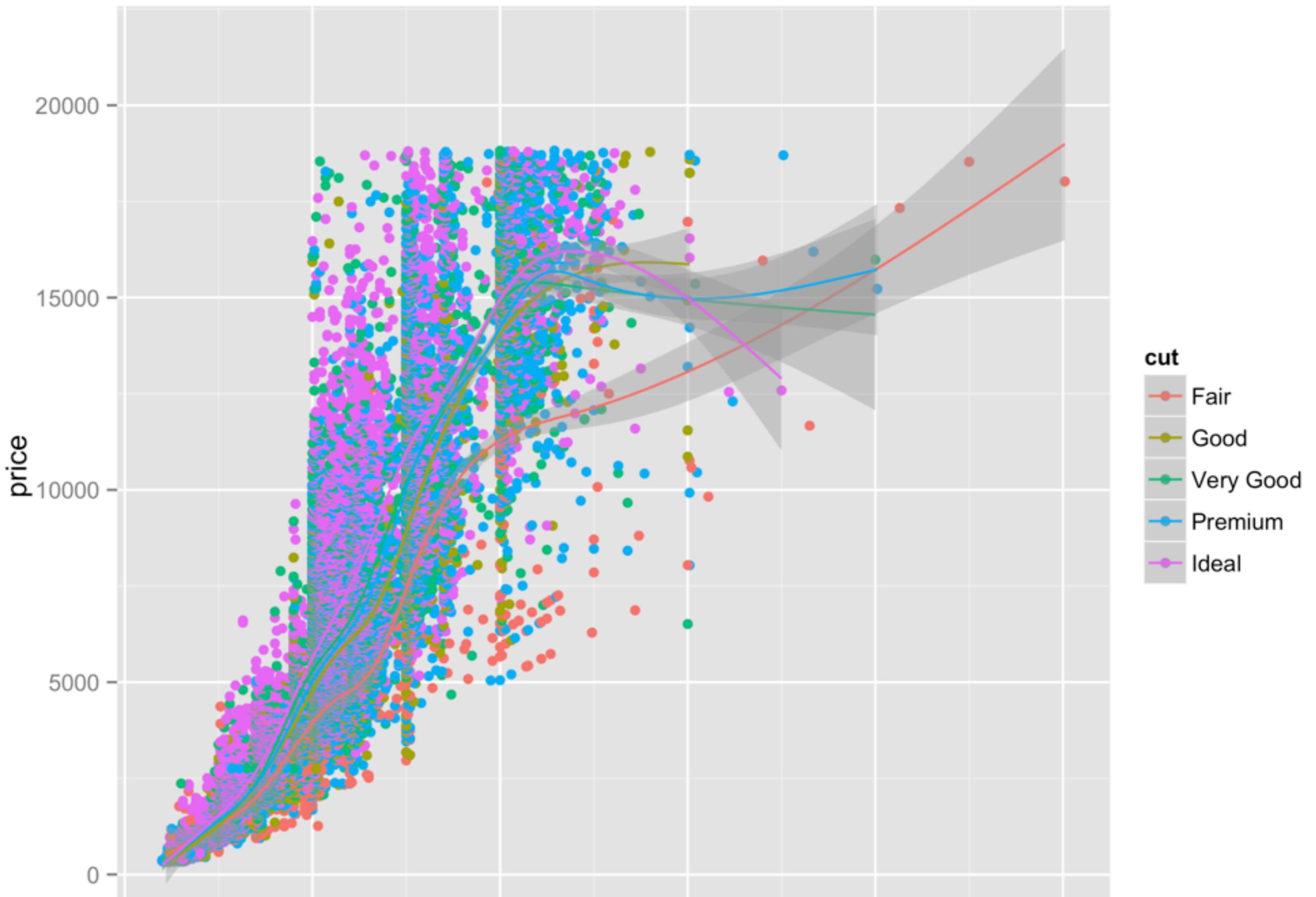
```
qplot(carat, price, data = diamonds, geom = "smooth",
      color = cut, se = FALSE, method = lm)
```

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/>

Index. ggplot2 0.9.2.1

docs.ggplot2.org/current/

ggplot2 0.9.2.1 Index

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
- [geom_errorbar](#)



Dependencies

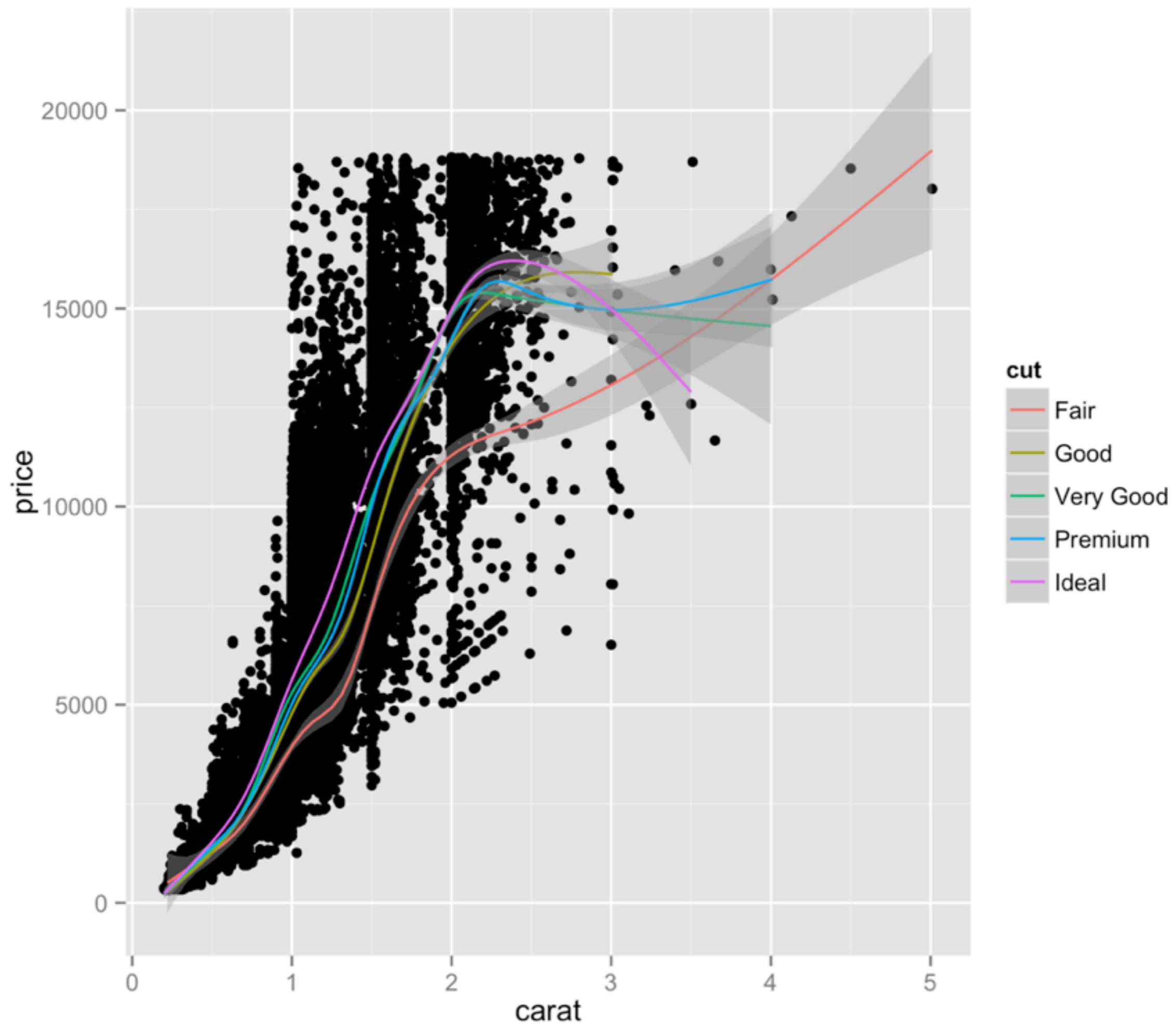
- **Depends:** stats, methods
- **Imports:** plyr, digest, grid, gtable, reshape2, scales, memoise, proto, MASS
- **Suggests:** quantreg, Hmisc, mapproj, maps, hexbin, maptools, multcomp, nlme, testthat
- **Extends:** sp

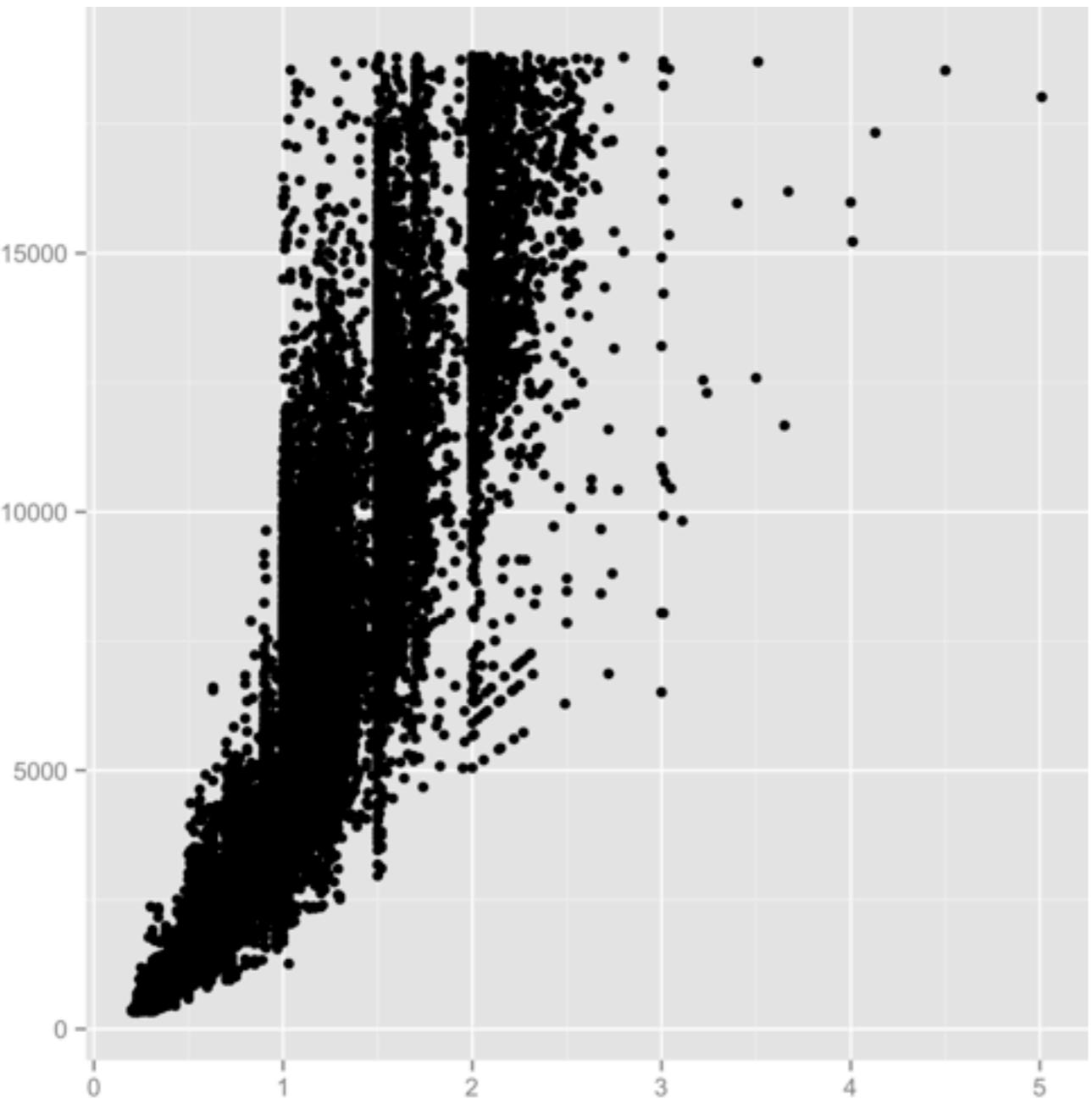


Advantage 2

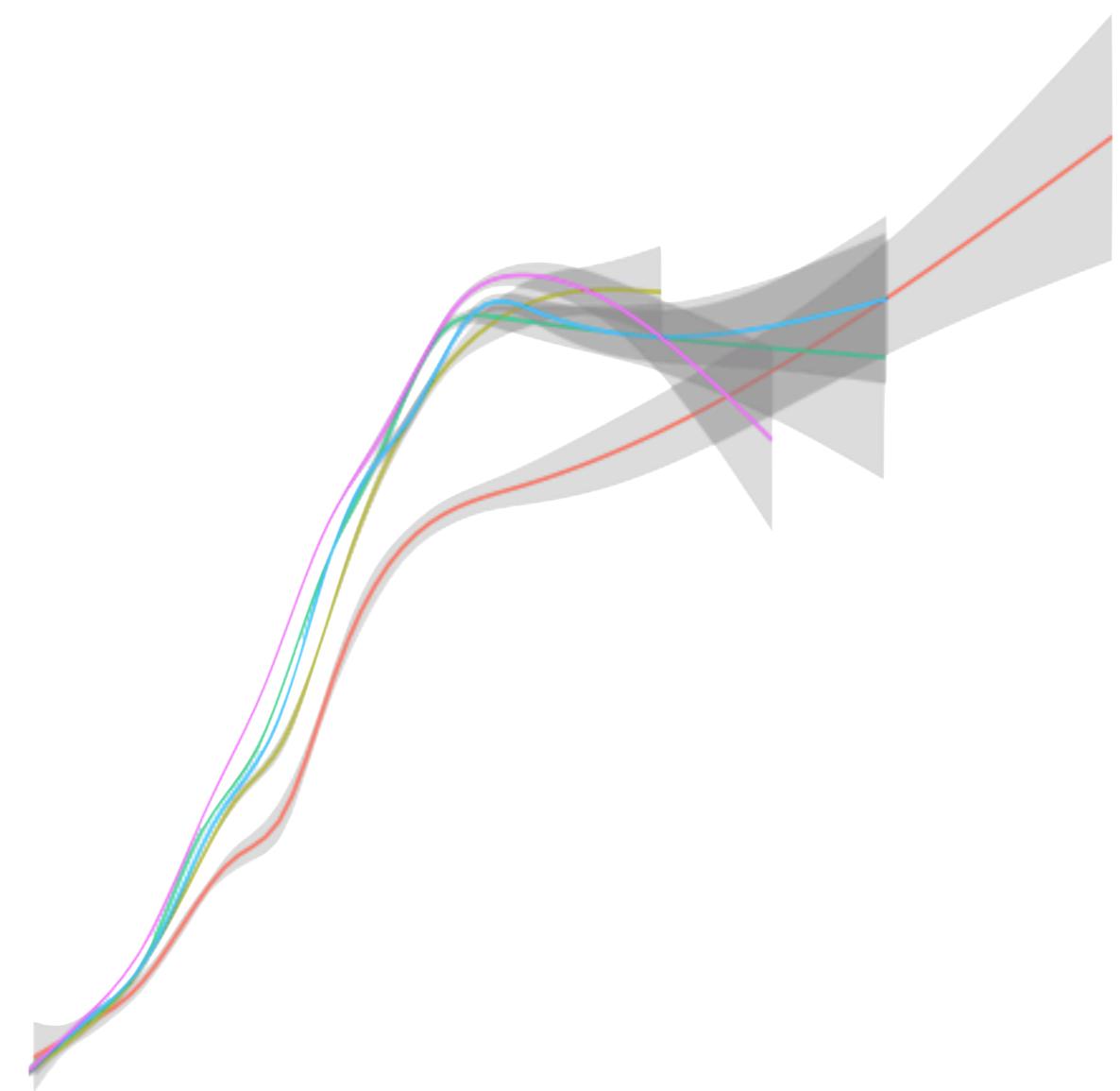
Geom functions provide a way to create separate layers with different aesthetics

What is a layer?

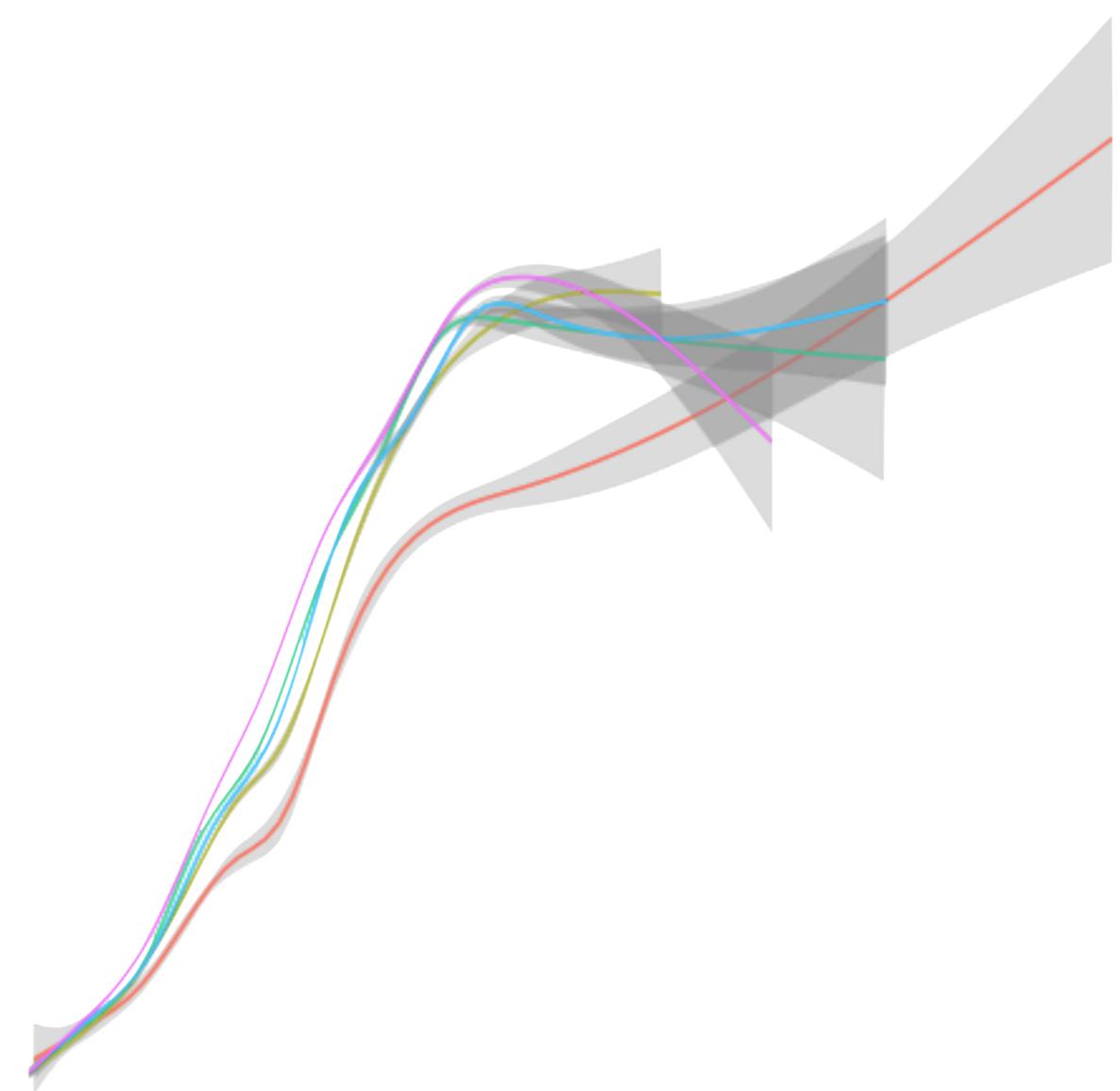
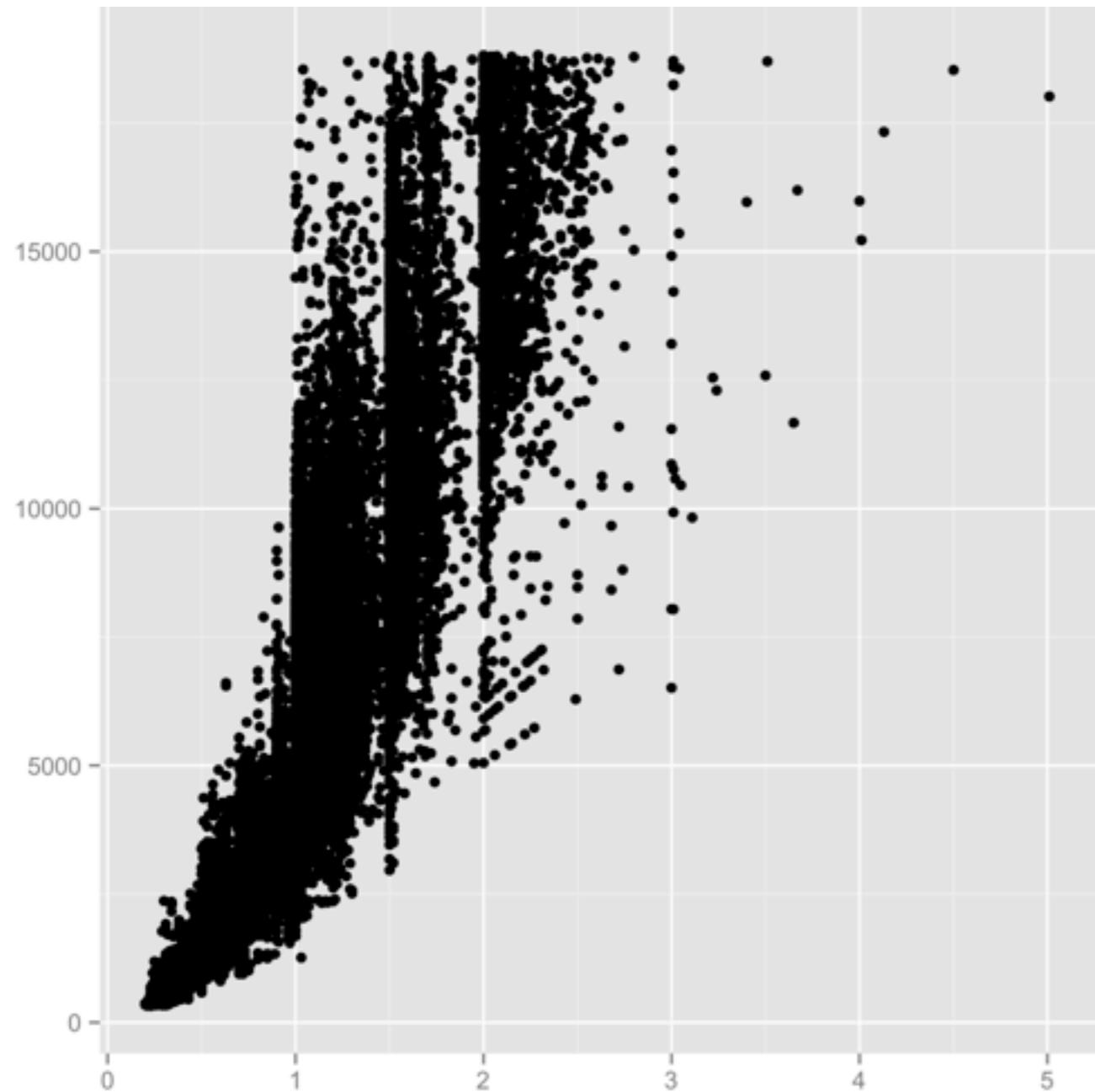


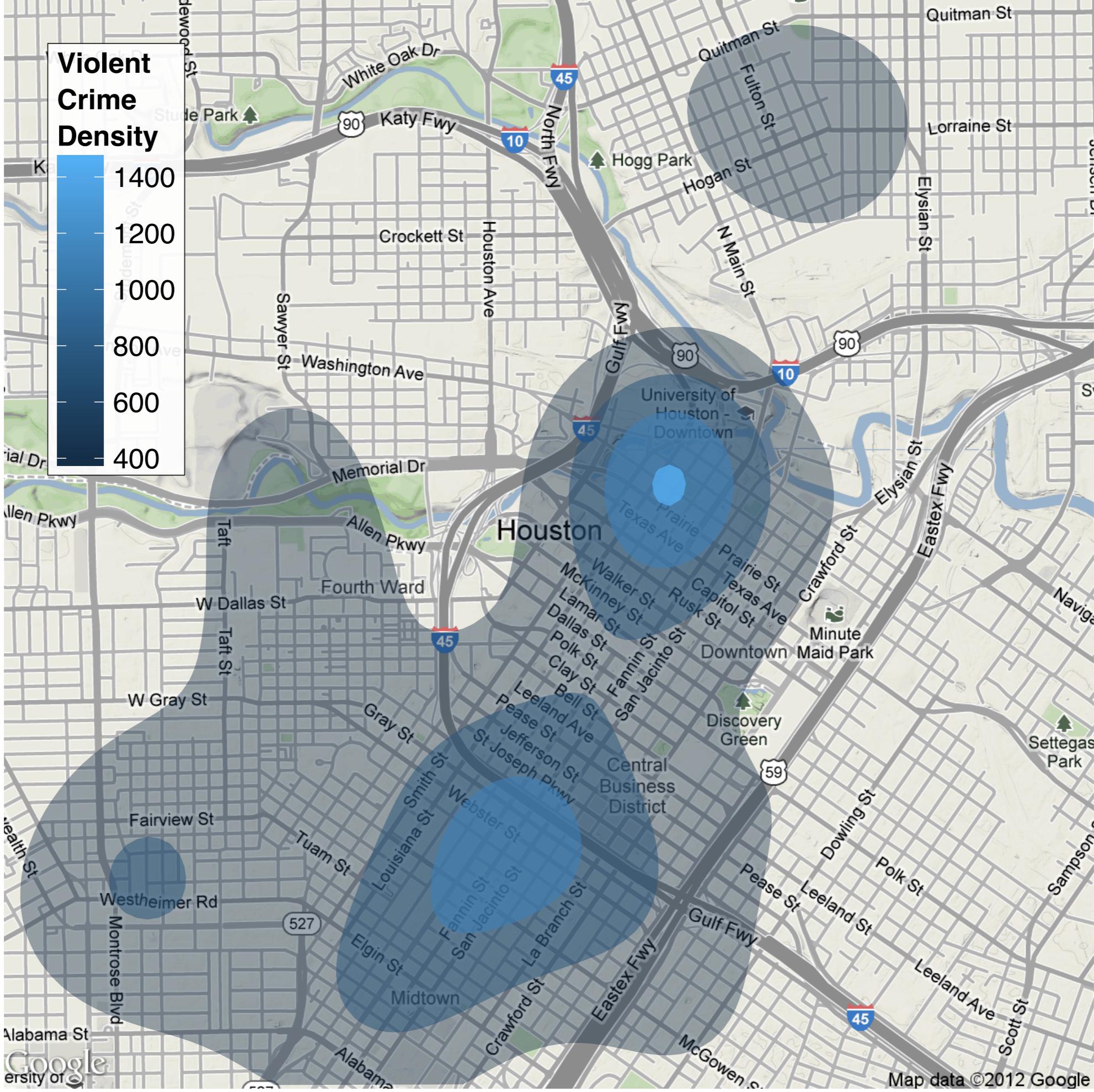


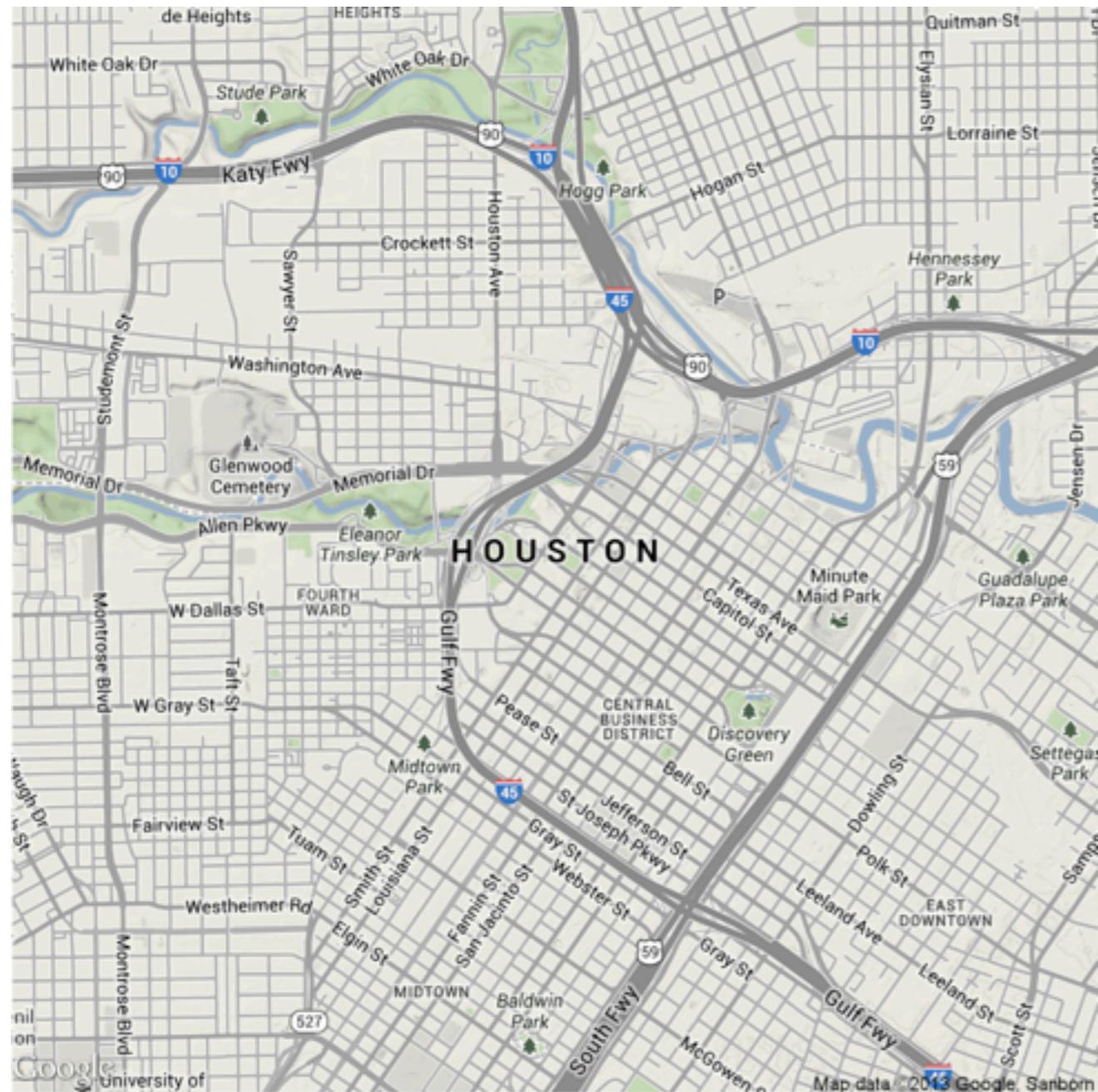
Layer 1



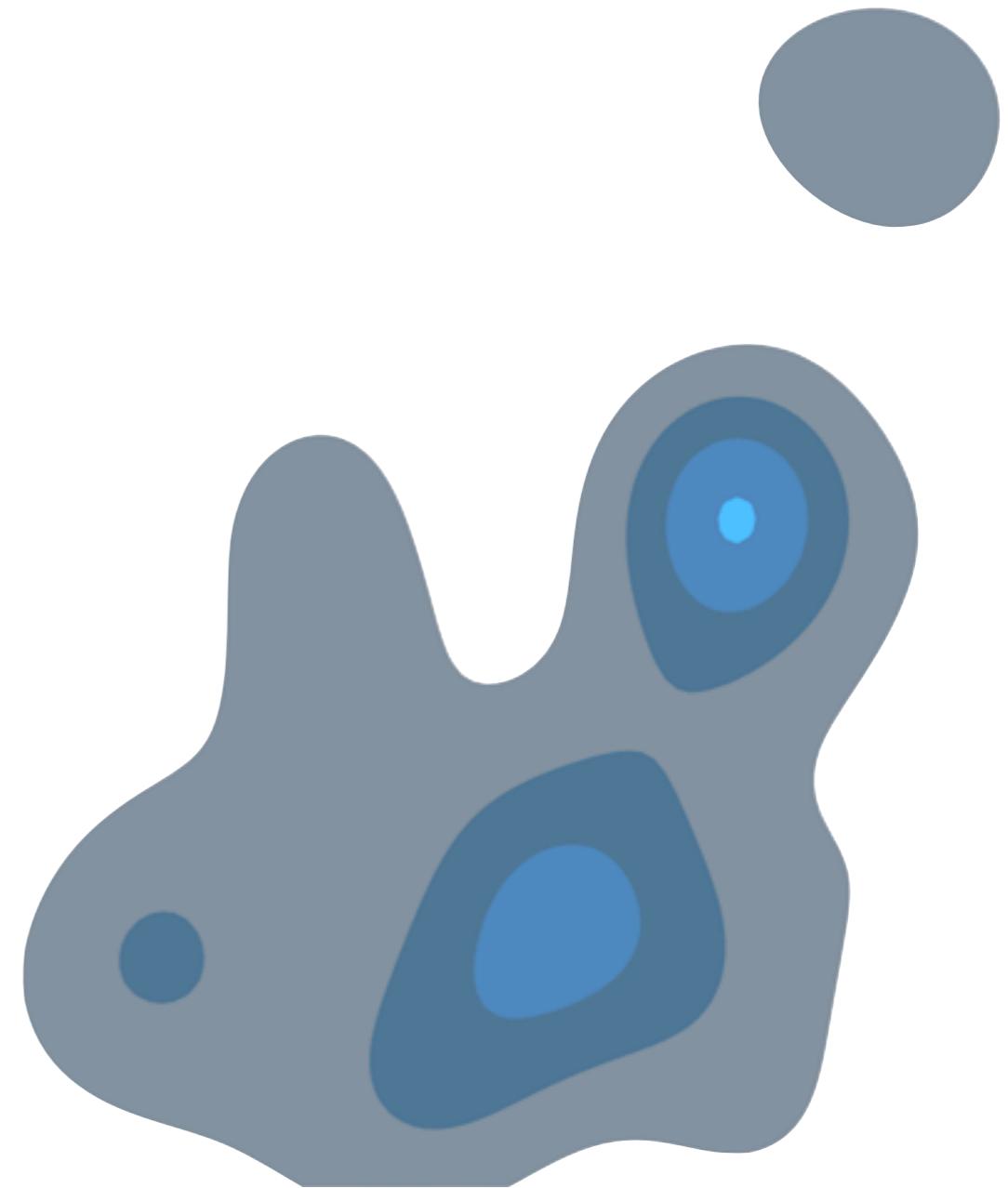
Layer 2



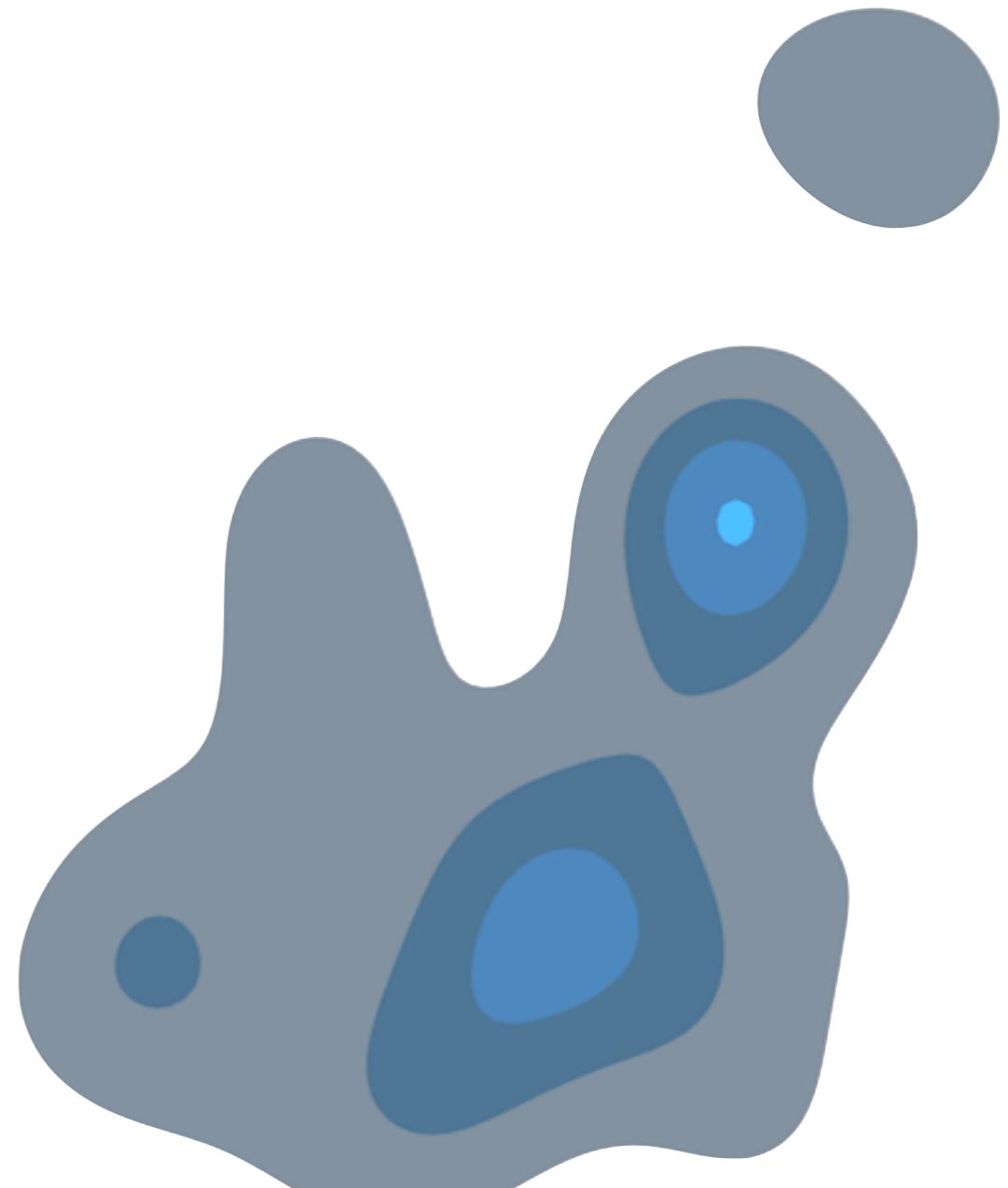
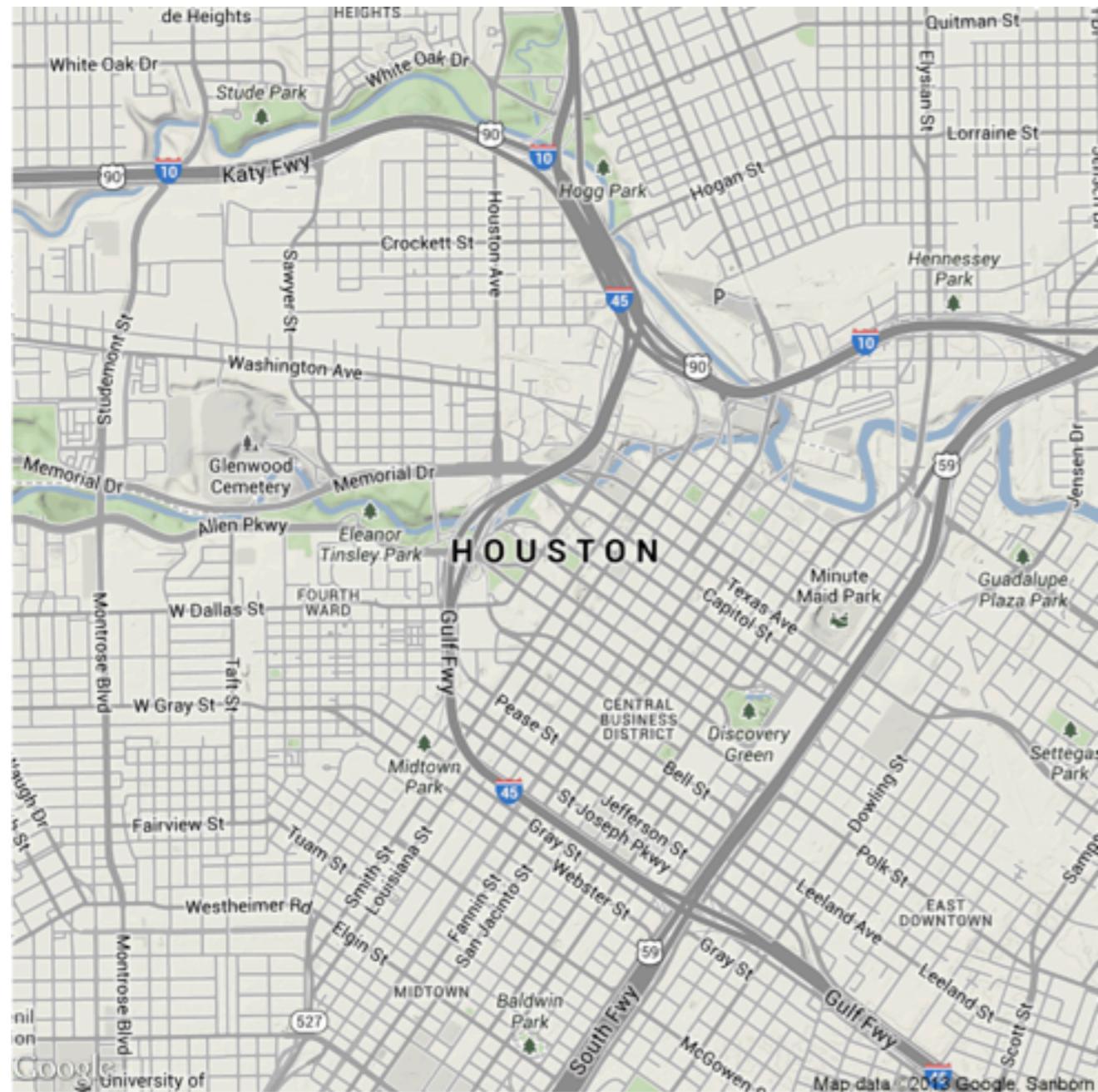




Layer 1



Layer 2



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

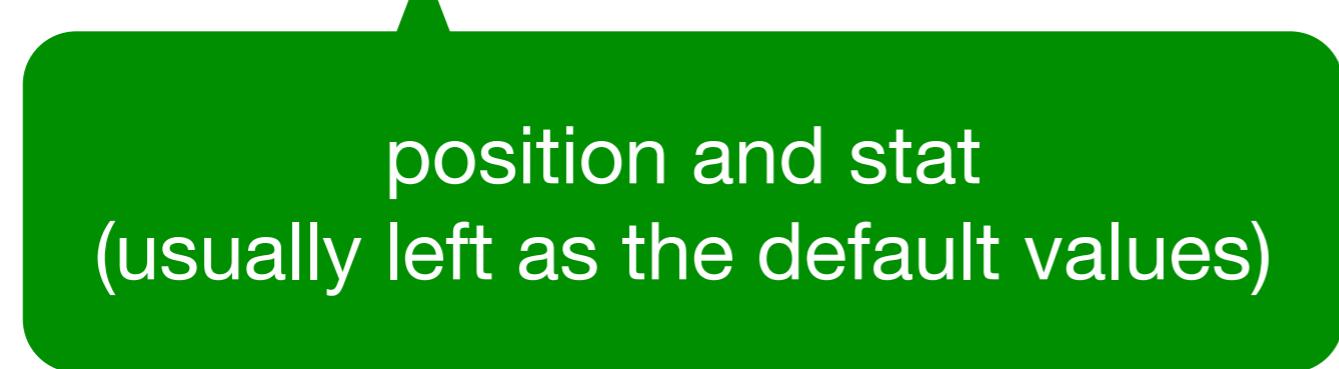
aesthetic mapping
wrapped in the aes function

```
geom_smooth(
```

)



data set



position and stat
(usually left as the default values)

ggplot

```
Coordinate system  ggplot() +  
+ a layer          geom_point(  
+ a layer          aes(x = carat, y = price),  
+ a layer          data = diamonds) +  
+ a layer          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

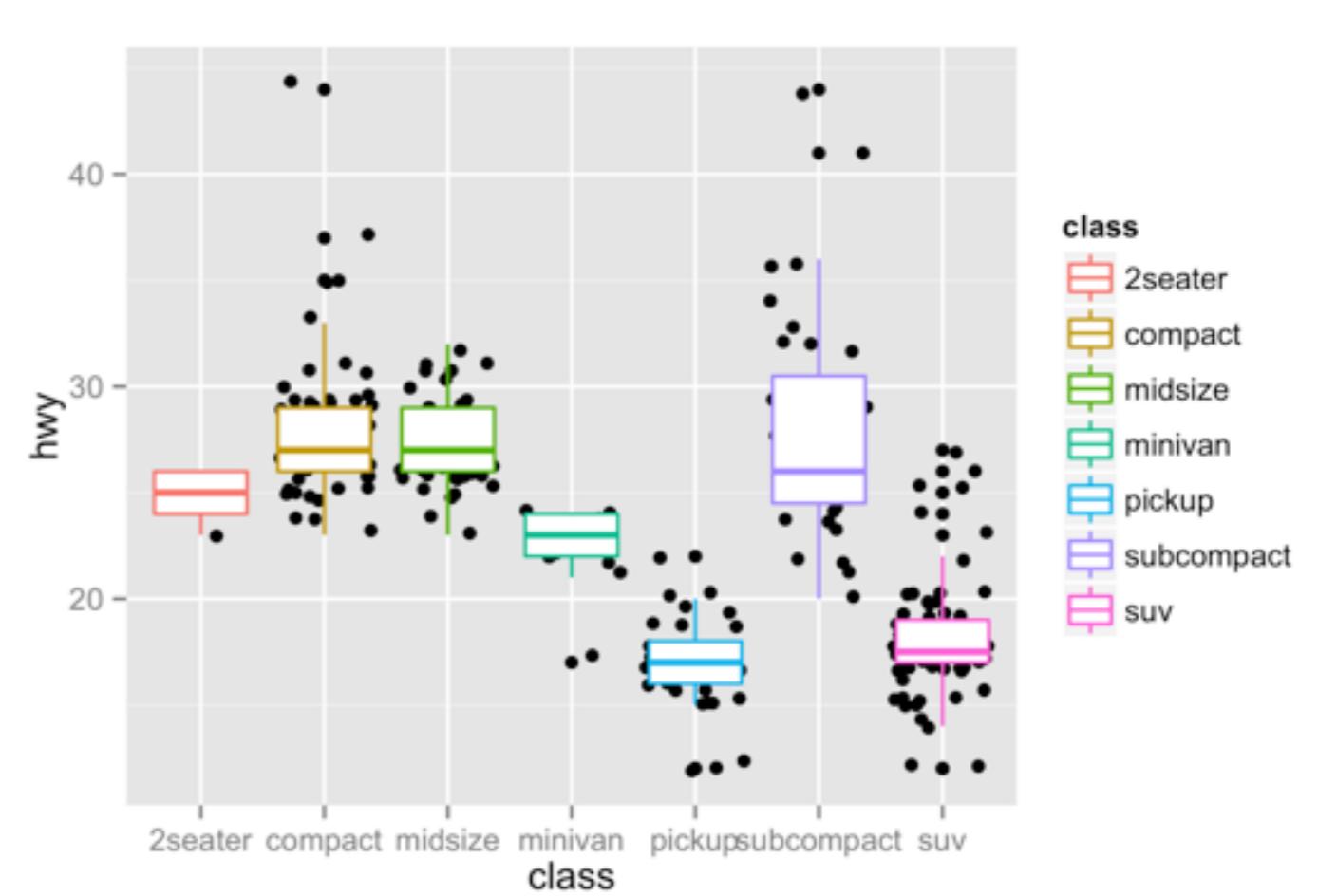
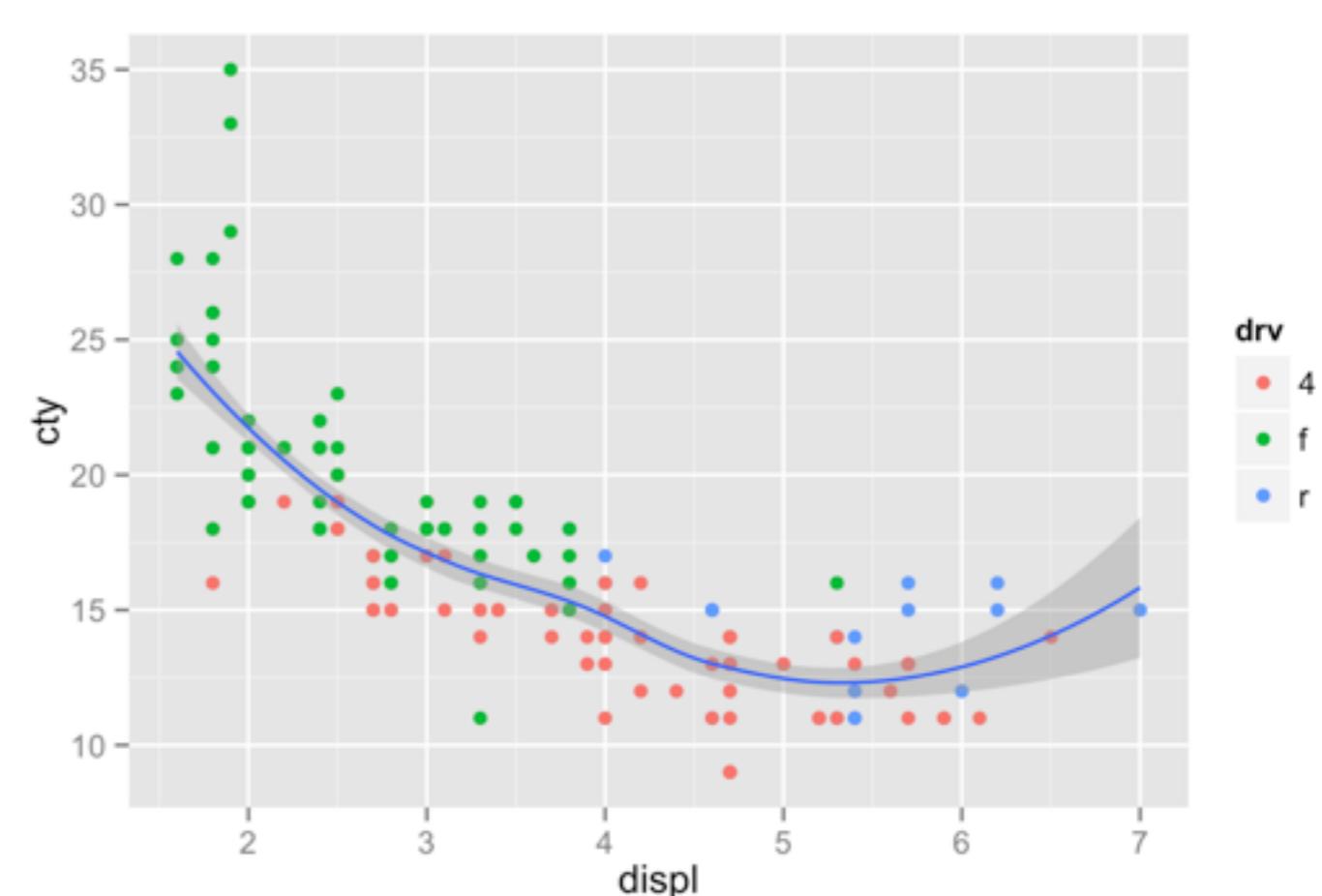
data

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))
```