ggplot2: a philosophy and a package

Peter D Smits

Committee on Evolutionary Biology
University of Chicago

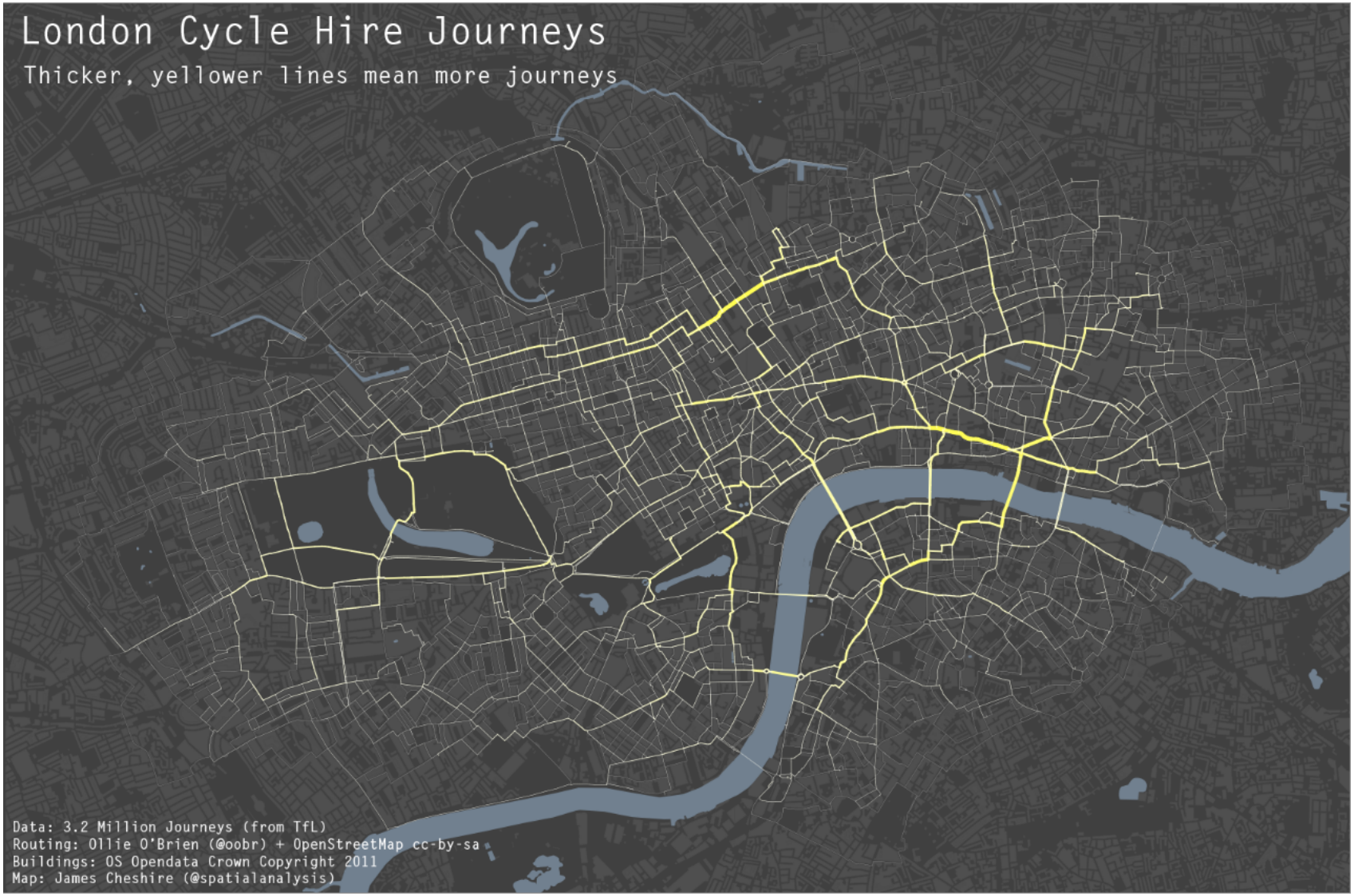
January 14, 2013

Introduction

there are three main ways of making graphics in R:
base graphics, lattice, grid, ggplot2.

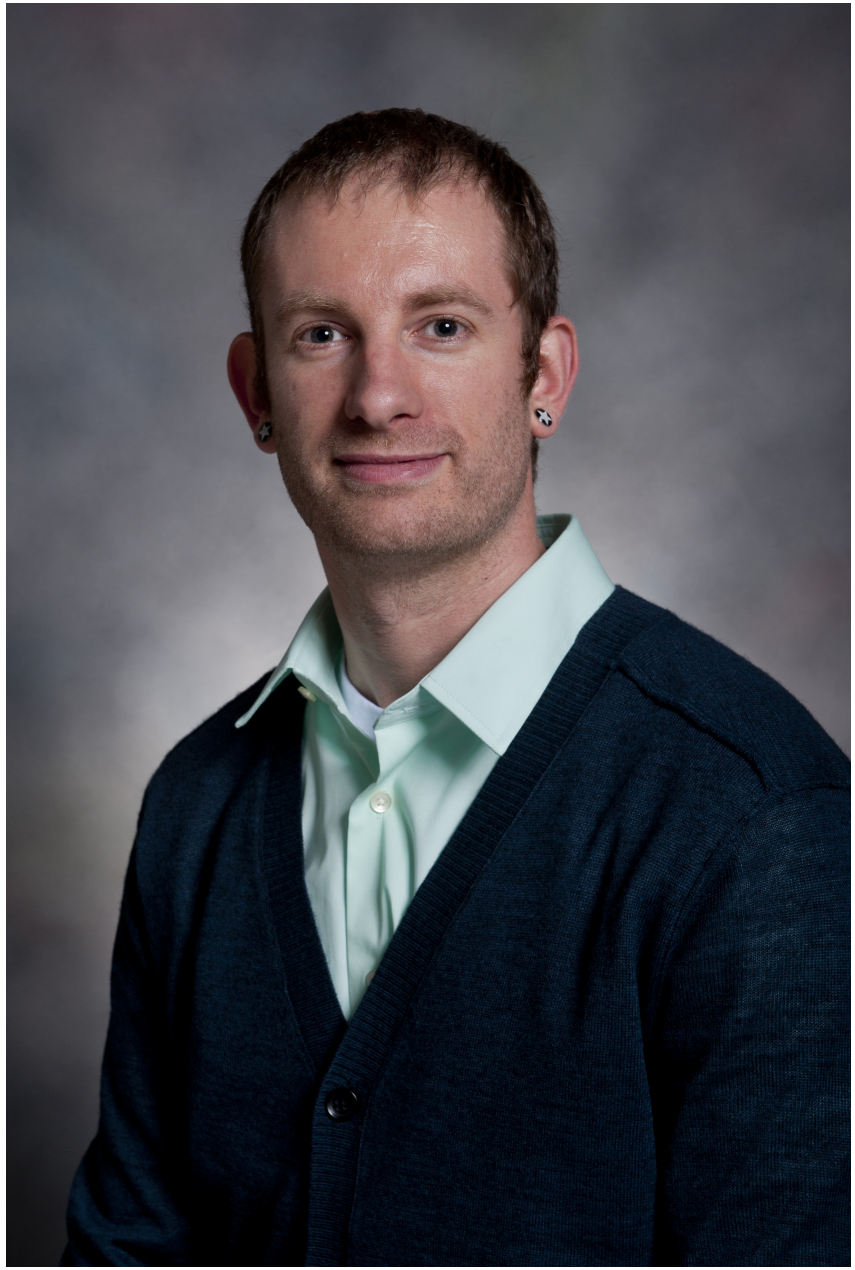
ggplot2 is an implementation of the Grammar of Graphics by Leland Wilkinson.

extremely popular, huge community, extremely powerful, slow.


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)

Hadley Wickham

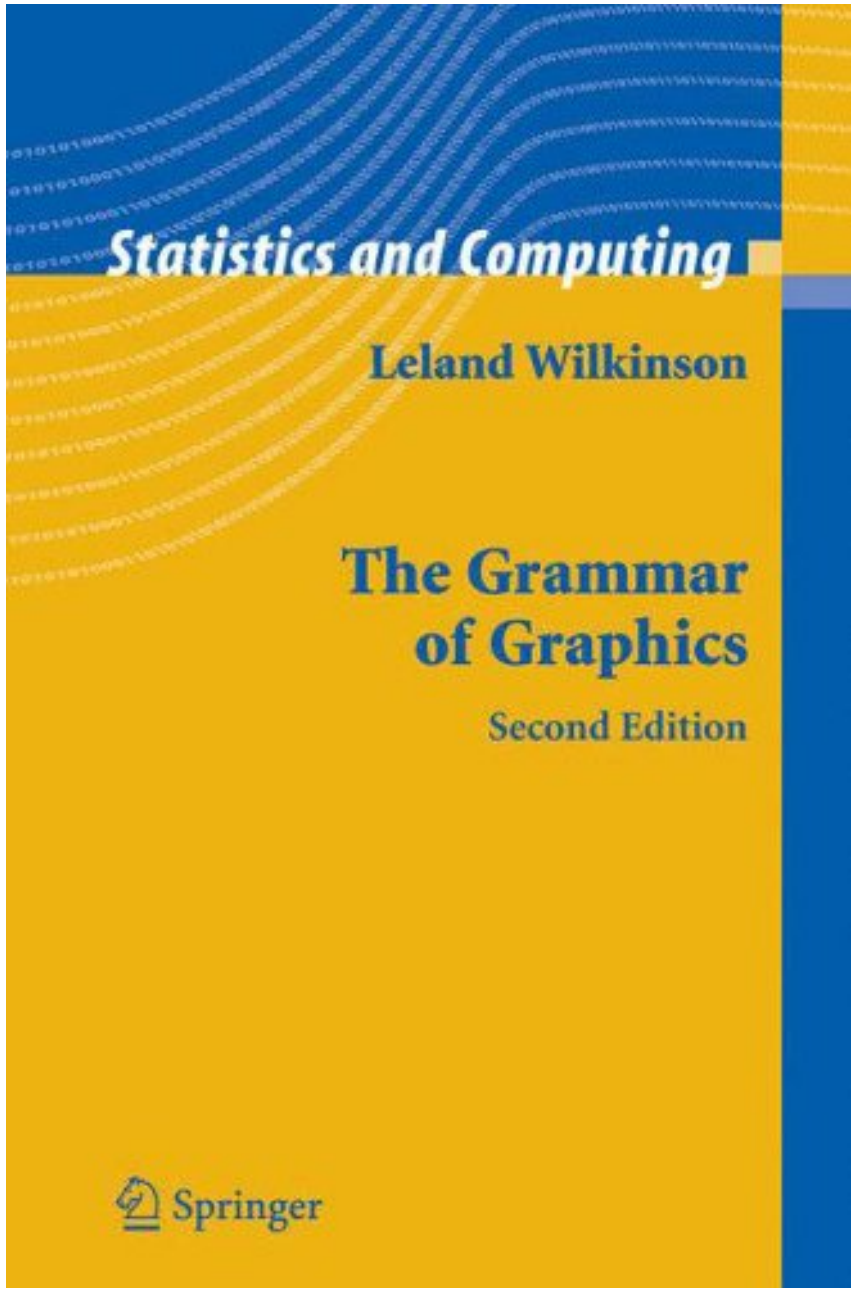
- statistics professor at Rice University
- from New Zealand (oddly common in statistics)
- author of many R packages (ggplot2, reshape2, plyr, devtools, and more)
- ggplot and reshape made up most of his PhD thesis



Grammar of Graphics

object-oriented approach to graphics where we have our graph "object" and we add to and modify the state of it

- variables
- aesthetics
- geometry
- statistics
- facets
- scales, etc.



Basics

`ggplot()` is the basic call, but does not produce a meaningful graph.

`aes()` is for aesthetics (x value, y value, group, colour).

`geom_*()` are geometries (points, bars, etc.).

`stat_*()` are statistics (densities, smooths, functions, etc.).

`facet_*()` faceting options and types.

`theme()` is for various theming options (used to be called `opts()`).

`scale_*()` are various scaling options (see scales package for better selection).

Basics

standard invocations of `ggplot()`

```
ggplot(df)
ggplot(df, aes(x = x, y = y))
ggplot()
```

df **must** be a data frame. x and y are columns of df.

creates an object of class "ggplot" which you probably want to assign (<-) to something.

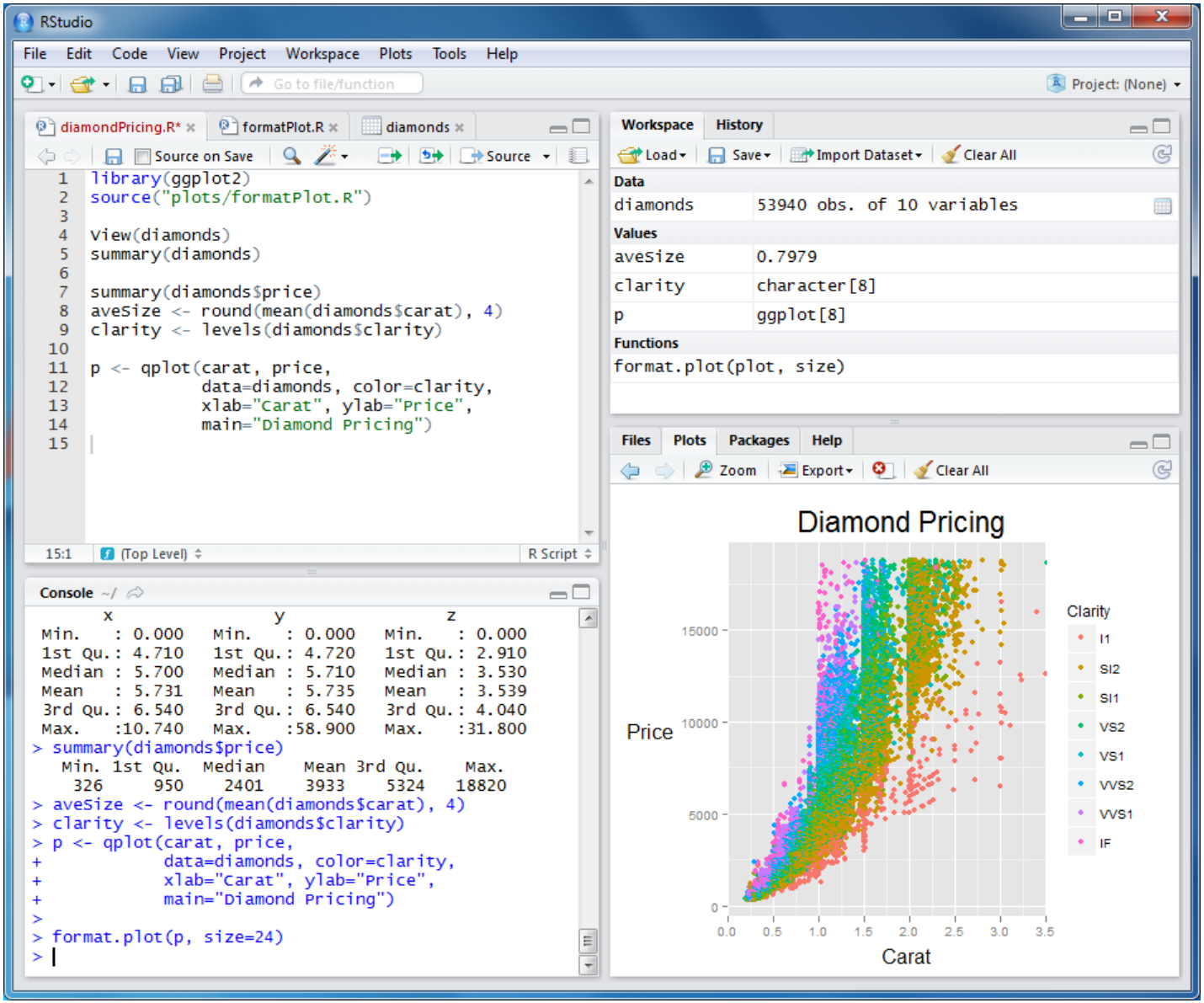
we can then modify this object to give it geometries, statistics, themes, etc.

Adding to a ggplot object

```
g <- ggplot(df, aes(x = x, y = y)) + geom_point()
g <- g + stat_smooth(method = "lm")
```

using the "+" operator, we can add attributes to a ggplot object.

Before we go to far, RStudio



Today's first data

the muscle car data set!

32 cars for 11 variables.

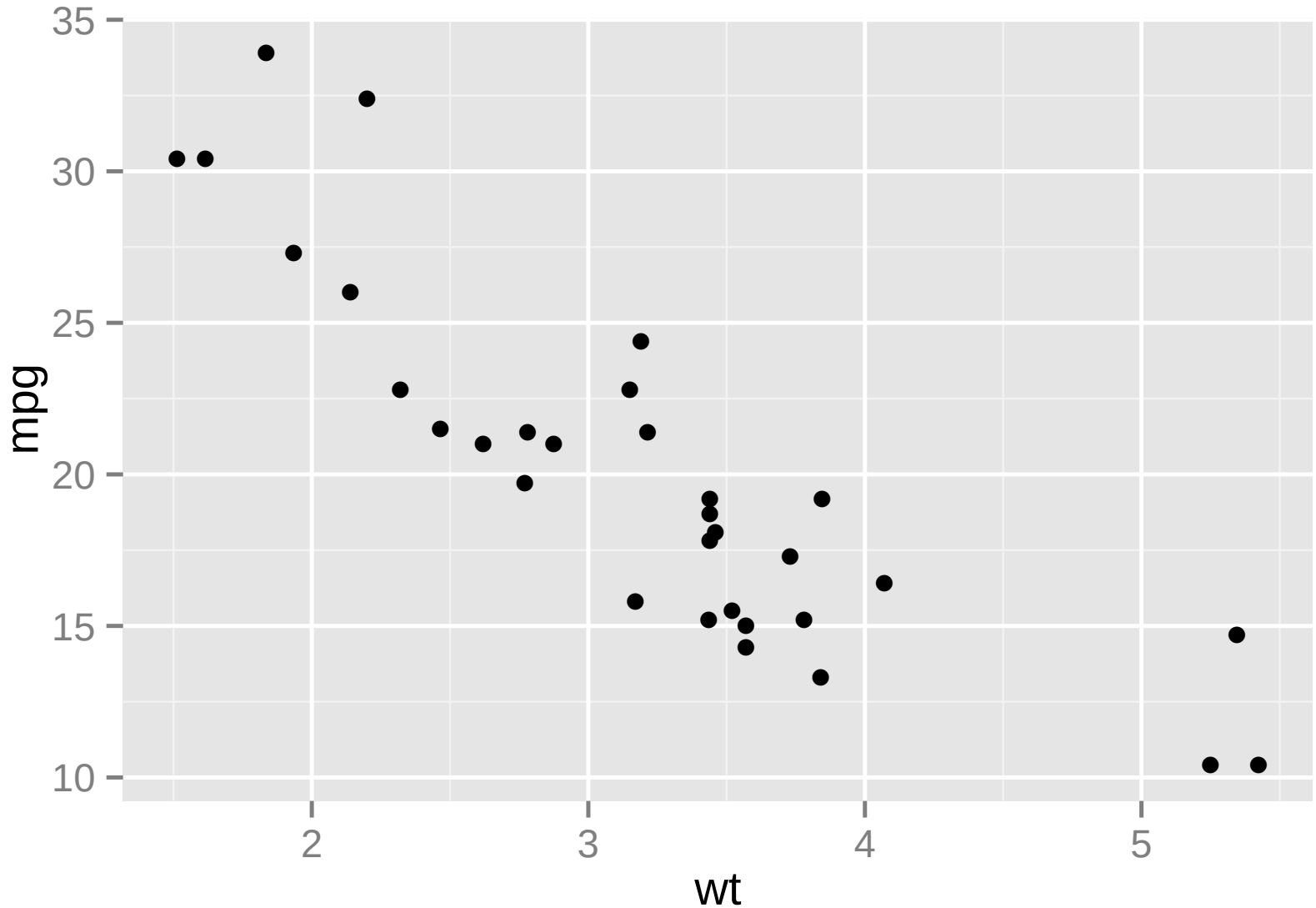
	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.00	6.00	160.00	110.00	3.90	2.62
Mazda RX4 Wag	21.00	6.00	160.00	110.00	3.90	2.88
Datsun 710	22.80	4.00	108.00	93.00	3.85	2.32
Hornet 4 Drive	21.40	6.00	258.00	110.00	3.08	3.21
Hornet Sportabout	18.70	8.00	360.00	175.00	3.15	3.44
Valiant	18.10	6.00	225.00	105.00	2.76	3.46

Our first graph

```
g1 <- ggplot(mtcars, aes(wt, mpg)) + geom_point()
```

Our first graph

```
g1
```

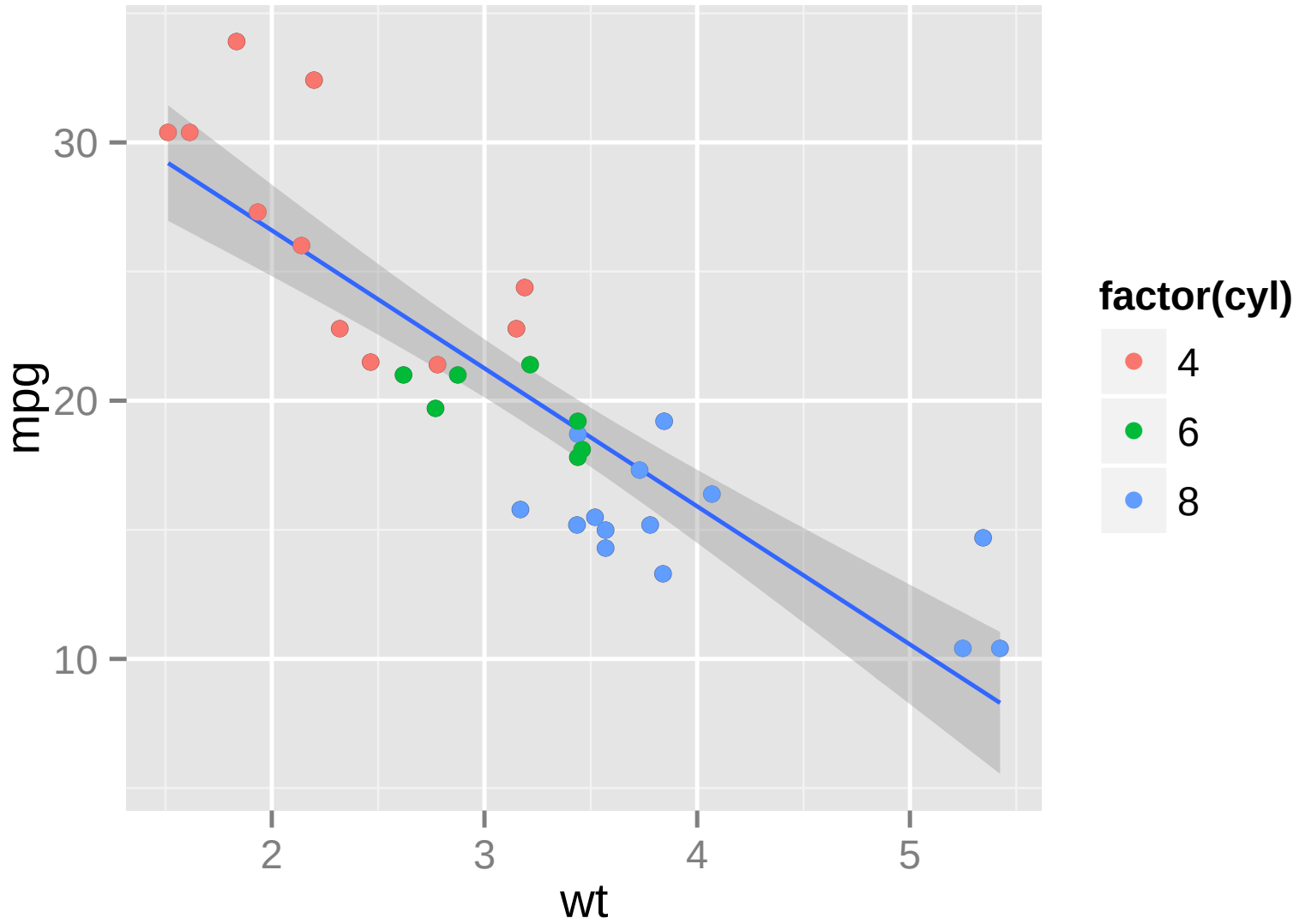


Adding a stat and modifying our first graph

```
g1 <- g1 + stat_smooth(method = "lm")
g1 <- g1 + geom_point(aes(colour = factor(cyl)))
```

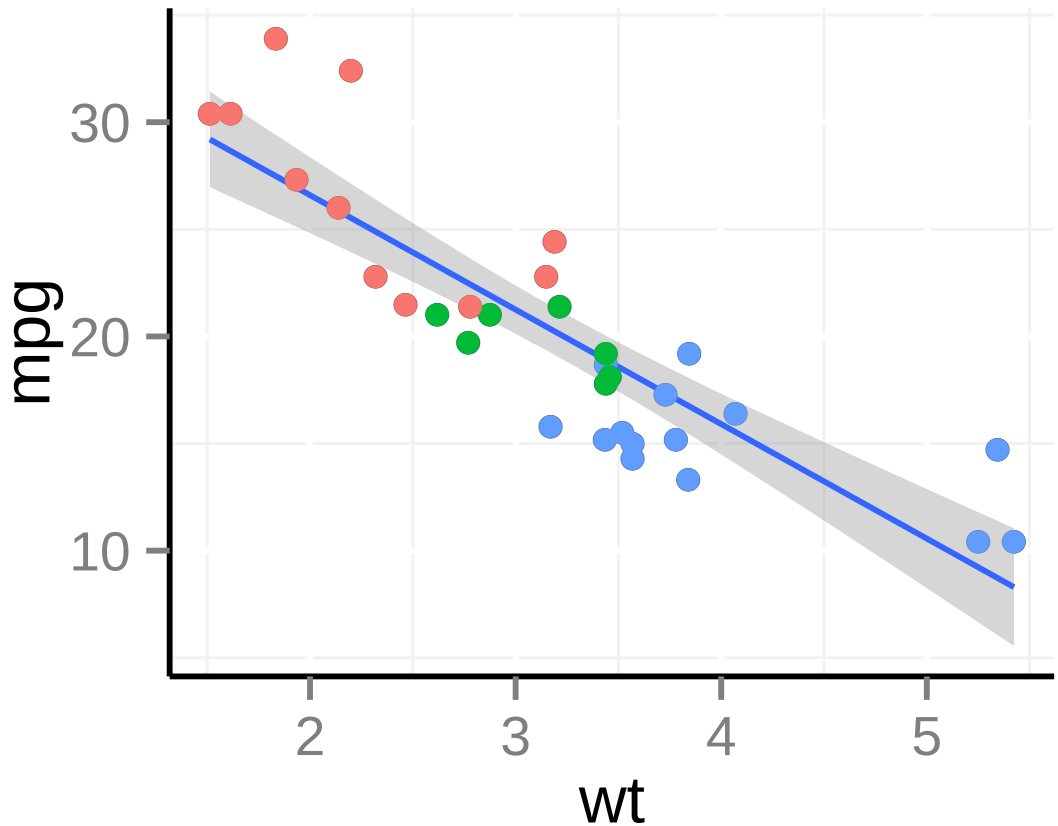
Adding a stat and modifying our first graph

```
g1
```



Try and make it look prettier

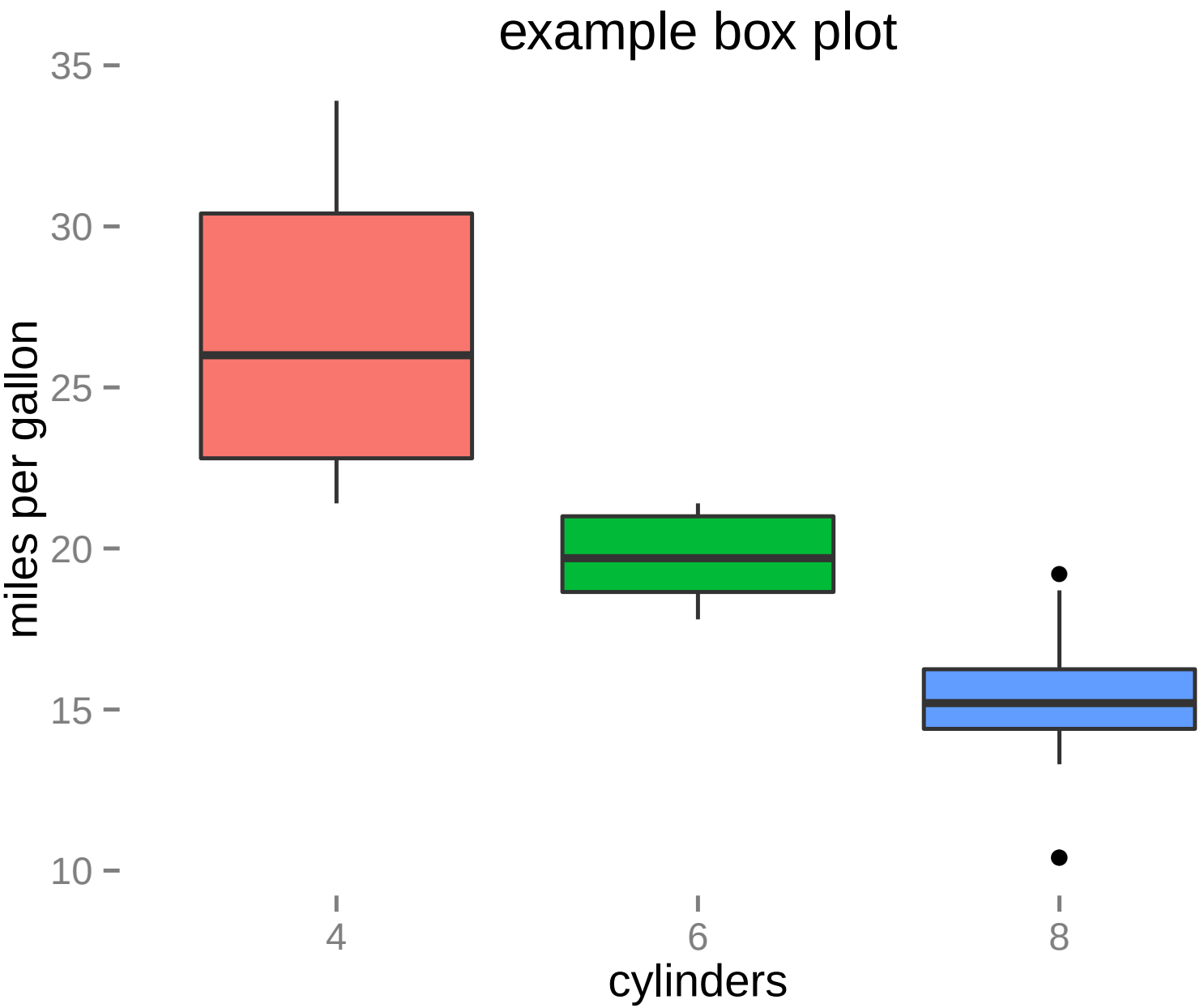
```
g1 <- g1 + theme(legend.position = 'none',
                 axis.line = element_line(colour = 'black'),
                 panel.background = element_blank())
g1
```



Our second graph

```
g2 <- ggplot(mtcars,
             aes(x = factor(cyl),
                 y = mpg,
                 fill = factor(cyl)))
g2 <- g2 + geom_boxplot()
g2 <- g2 + theme(legend.position = 'none',
                 panel.background = element_blank(),
                 panel.grid = element_blank())
g2 <- g2 + labs(x = 'cylinders',
                y = 'miles per gallon',
                title = 'example box plot')
```

Our second graph



Today's second dataset

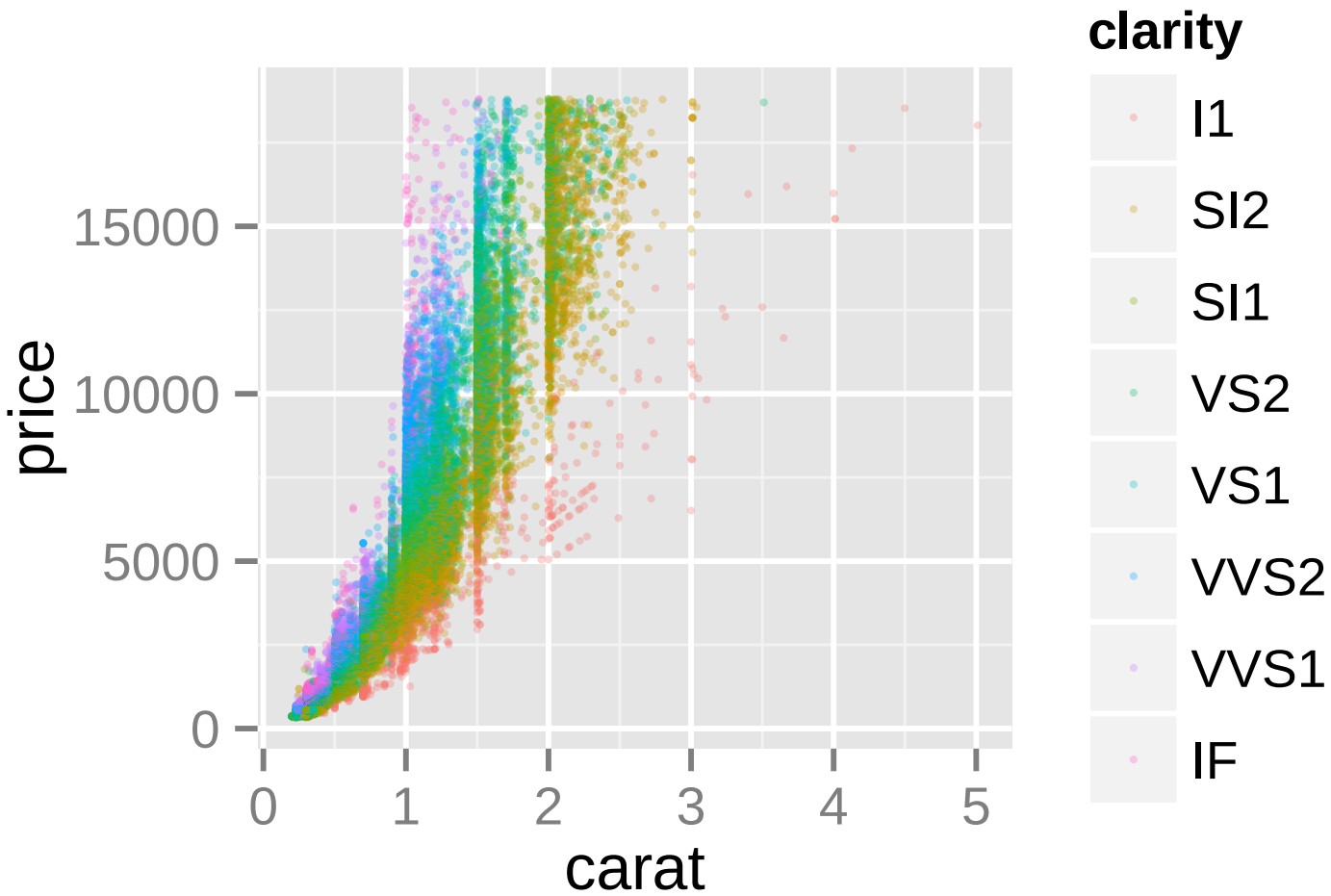
diamonds!

53940 samples for 10 variables.

	carat	cut	color	clarity	depth	table	price
1	0.23	Ideal	E	SI2	61.50	55.00	326
2	0.21	Premium	E	SI1	59.80	61.00	326
3	0.23	Good	E	VS1	56.90	65.00	327
4	0.29	Premium	I	VS2	62.40	58.00	334
5	0.31	Good	J	SI2	63.30	58.00	335
6	0.24	Very Good	J	VVS2	62.80	57.00	336

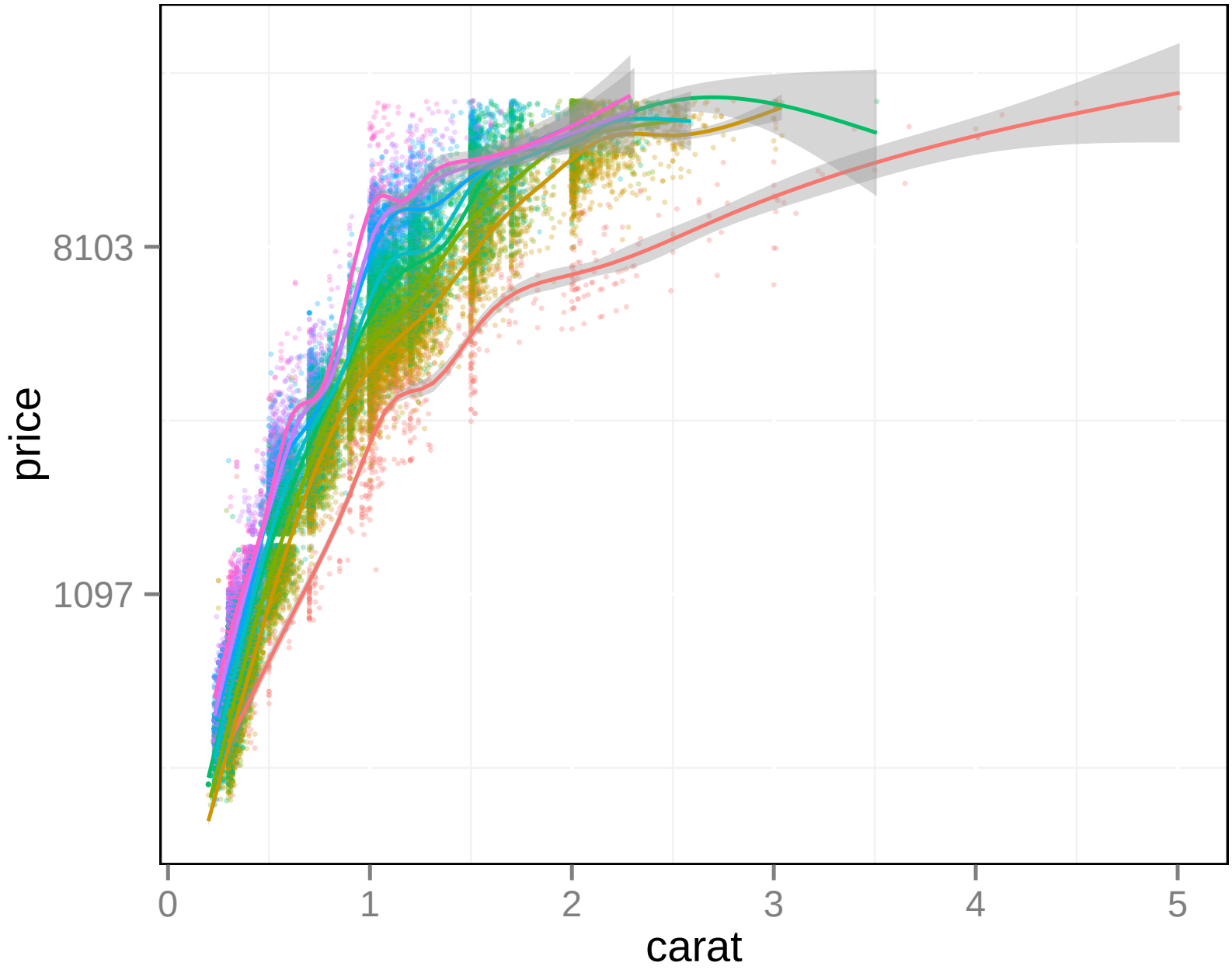
Looking at diamonds

```
d <- ggplot(diamonds, aes(x = carat, y = price,
                          colour = clarity))
d <- d + geom_point(alpha = 0.3, size = 0.7)
d
```



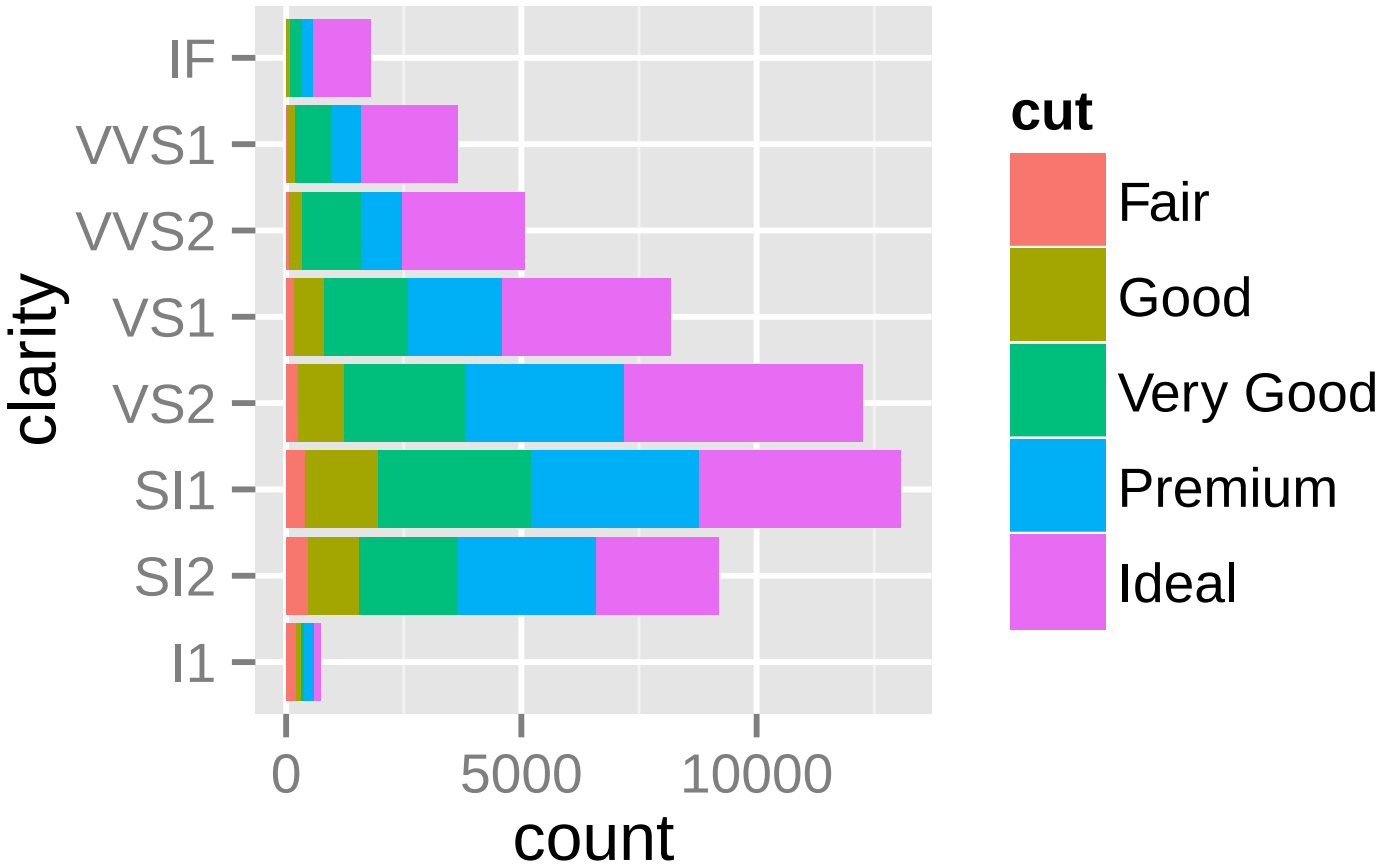
Make that look better

```
library(scales)
d <- d + scale_y_continuous(trans = log_trans())
d <- d + stat_smooth()
d <- d + theme(panel.background = element_blank(),
               panel.border = element_rect(colour = 'black',
                                           fill = NA),
               legend.position = 'none')
```


8103
1097
price
0
1
2
3
4
5
carat

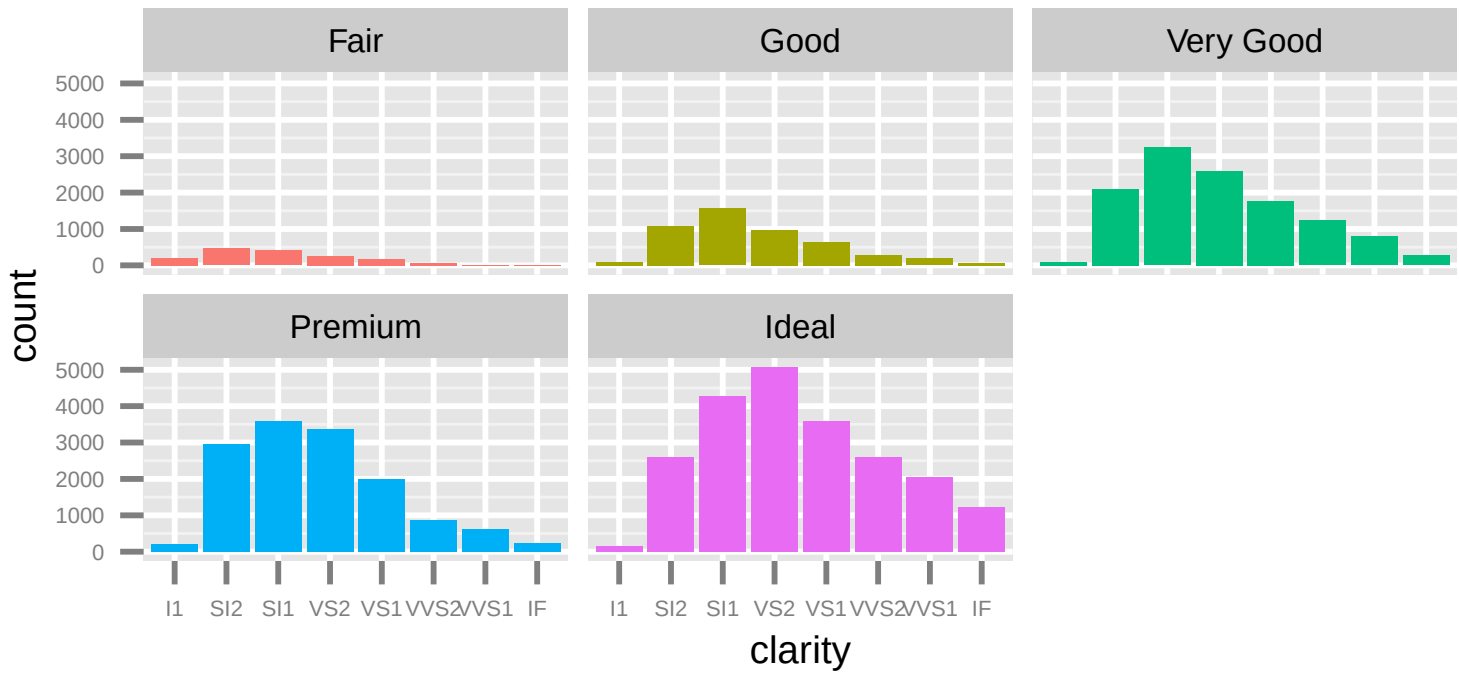
More diamonds

```
d2 <- ggplot(diamonds,
             aes(x = clarity, fill = cut))
d2 <- d2 + geom_bar()
d2 + coord_flip()
```



Make pretty

```
d2 <- d2 + facet_wrap(~ cut)
d2 <- d2 + theme(legend.position = 'none',
                 axis.text = element_text(size = 4),
                 axis.title = element_text(size = 7),
                 strip.text = element_text(size = 6))
d2
```



Other useful packages

GGally: matrix plots.

ggthemes: various canned themes to make your plots prettier (or hilariously ugly).

plyr: generalized apply type functions. split-apply-combine approach to data munging.

reshape2: manipulate the layout and structure of a matrix/data.frame. more data munging.

Useful websites

`http://docs.ggplot2.org/current/` : current ggplot documentation (very good)

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

`http://wiki.stdout.org/rcookbook/Graphs` : various tips and tricks to get over problems

`http://stackoverflow.com/` : coding question/answer site

`http://stats.stackexchange.com/` : statistics question/answer site

`http://www.r-bloggers.com/` : R blog aggregator