ILLINOIS

Visualization II

Lecture 18

Dirk Eddelbuettel

STAT 430: Data Science Programming Methods (Fall 2019)

Department of Statistics, University of Illinois

Last lecture

- base graphics and lattice

Today

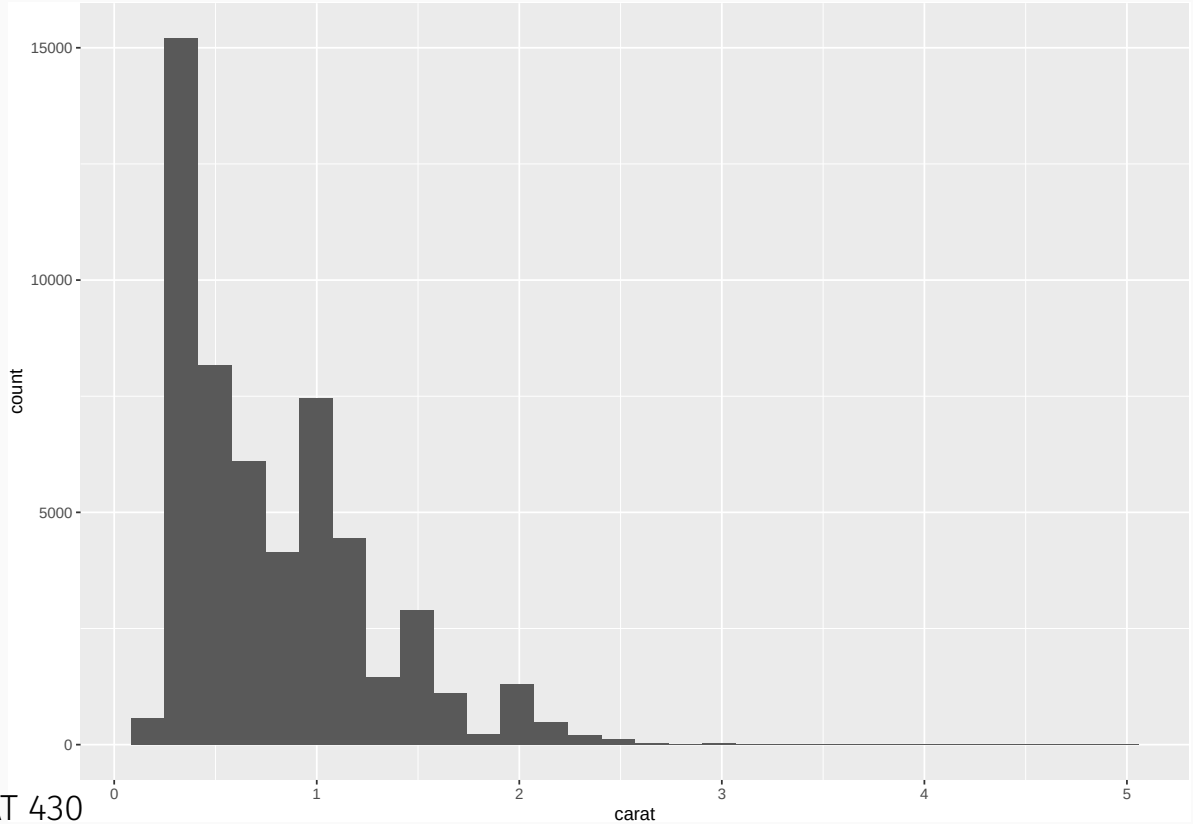
- ggplot2
- Resources:
  - Kieran Healy 'SocViz' book and web site `socviz.co`
  - ggplot2 web site and numerous tutorials on the web
  - Chapter 22 in R for Data Science
  - Chapter 7 in R for Everyone: Adv. Analytics and Graphics, 2nd Ed

GGPLOT2

Grammar of Graphics

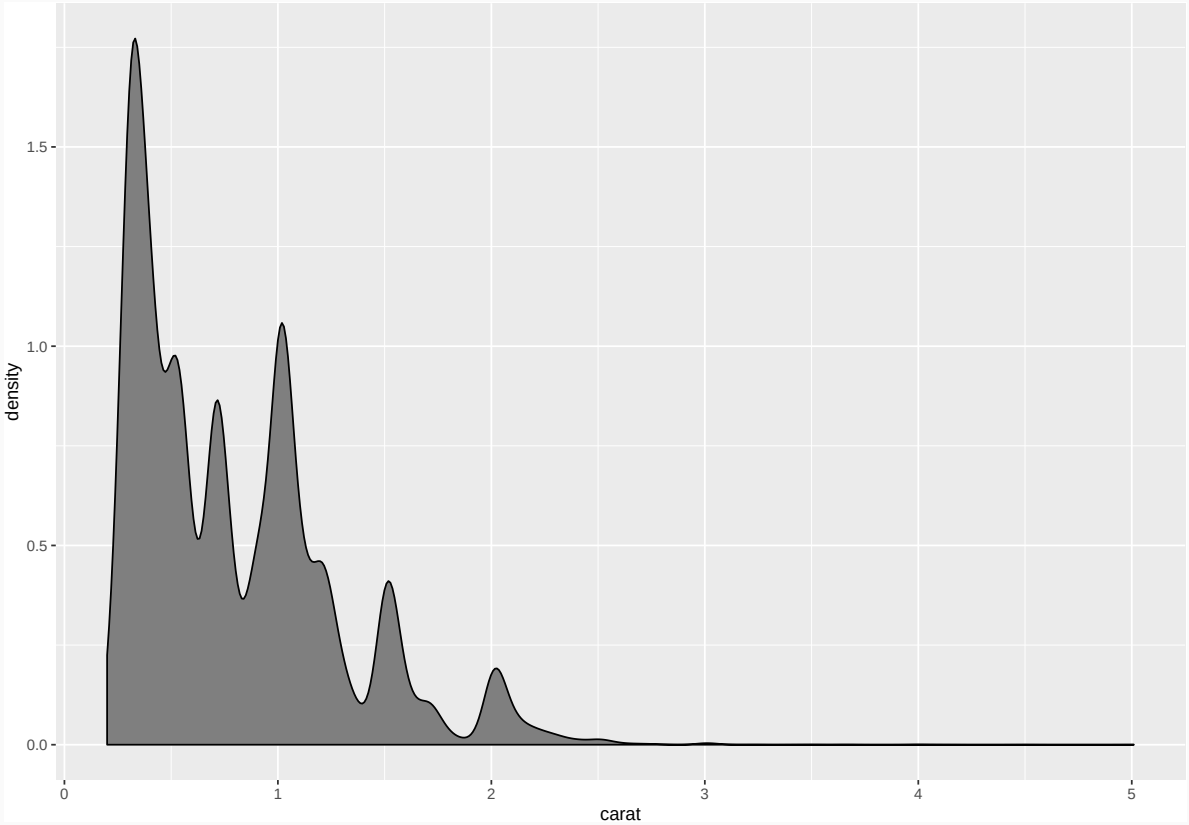
- In his 2005 book Grammar of Graphics Leland Wilkinson formulates a layered framework for graph creation
- Hadley Wickham implemented ggplot2 as part of his PhD work
- The R package has
  - become immensely popular and widely used
  - has seen several important revisions
  - spawned a rich system of add-on packages
  - been reimplemented for Python, Julia and other systems
- Numerous books describe it, we follow Chapter 7 of Lander and Chapter 22 of Wickham and Grolemund

```
suppressMessages(library(ggplot2))
data(diamonds) # data set in package
ggplot(data=diamonds) + geom_histogram(aes(x=carat), bins=30)
```



A layered graph: first `data`, then a `geom`, here for a histogram. We have to assign the number of bins whereas R's `hist()` and others choose this for us.

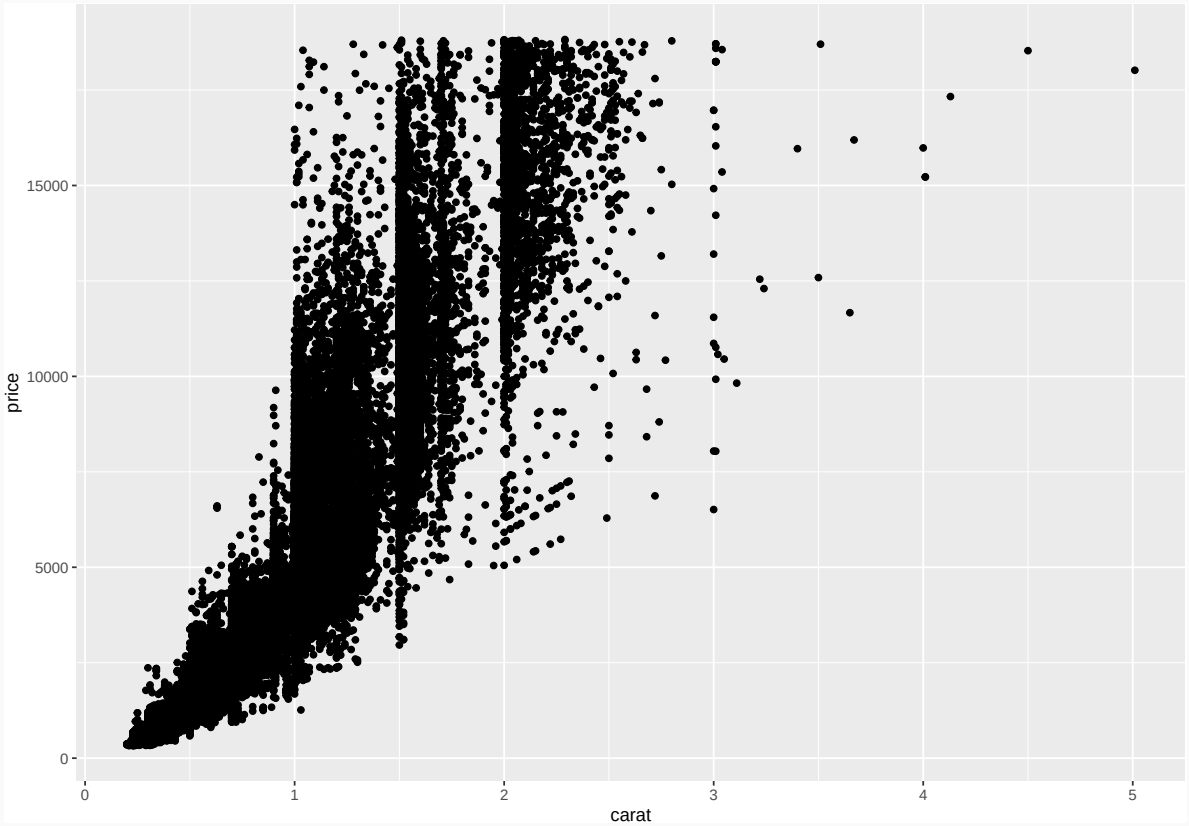
```
ggplot(data=diamonds) +
  geom_density(aes(x=carat), fill="grey50")
```



A density estimate as an alternative.

The bandwidth is chosen for us.

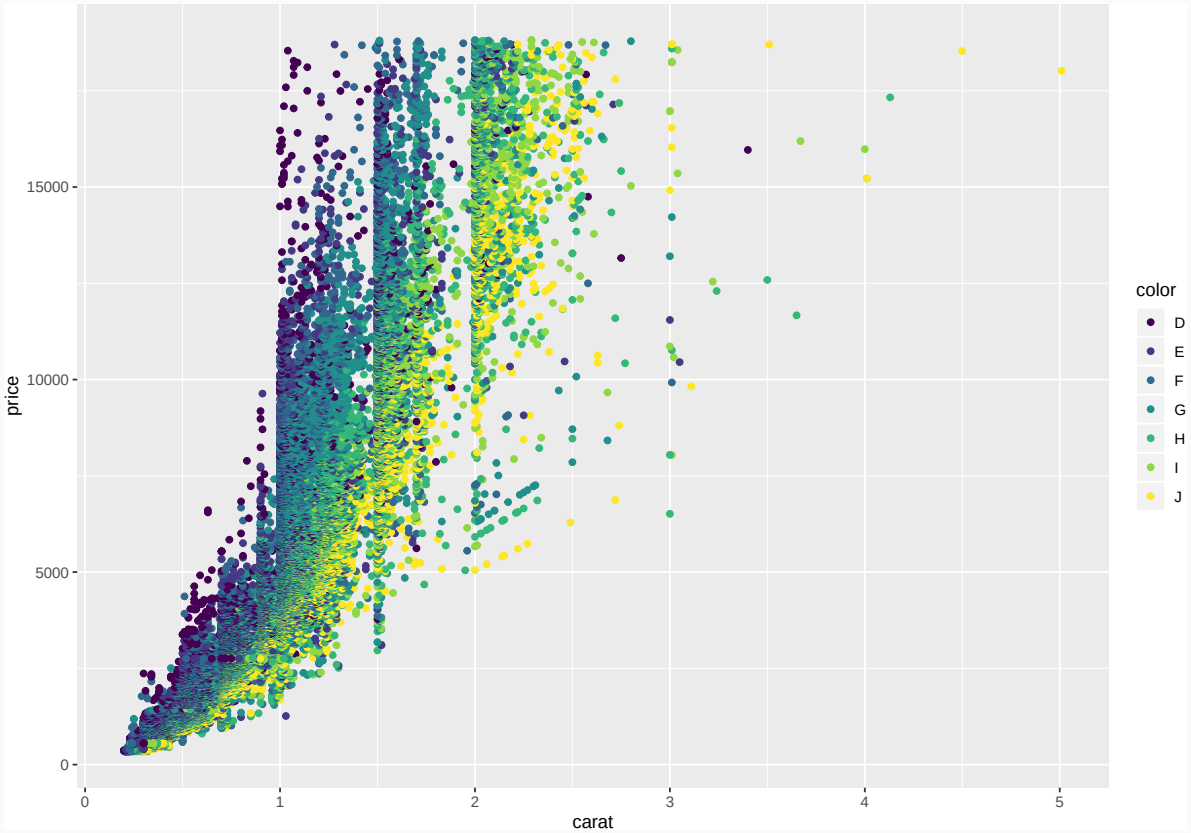
```
ggplot(diamonds, aes(x=carat, y=price)) +
  geom_point()
```



Here we omit the default `data=` argument and move `aes()` into the initial call.

Having given `aes()` the default arguments for `geom_point()`, it produces a scatter plot.

```
g <- ggplot(diamonds, aes(x=carat, y=price))
g +  geom_point(aes(color=color))
```

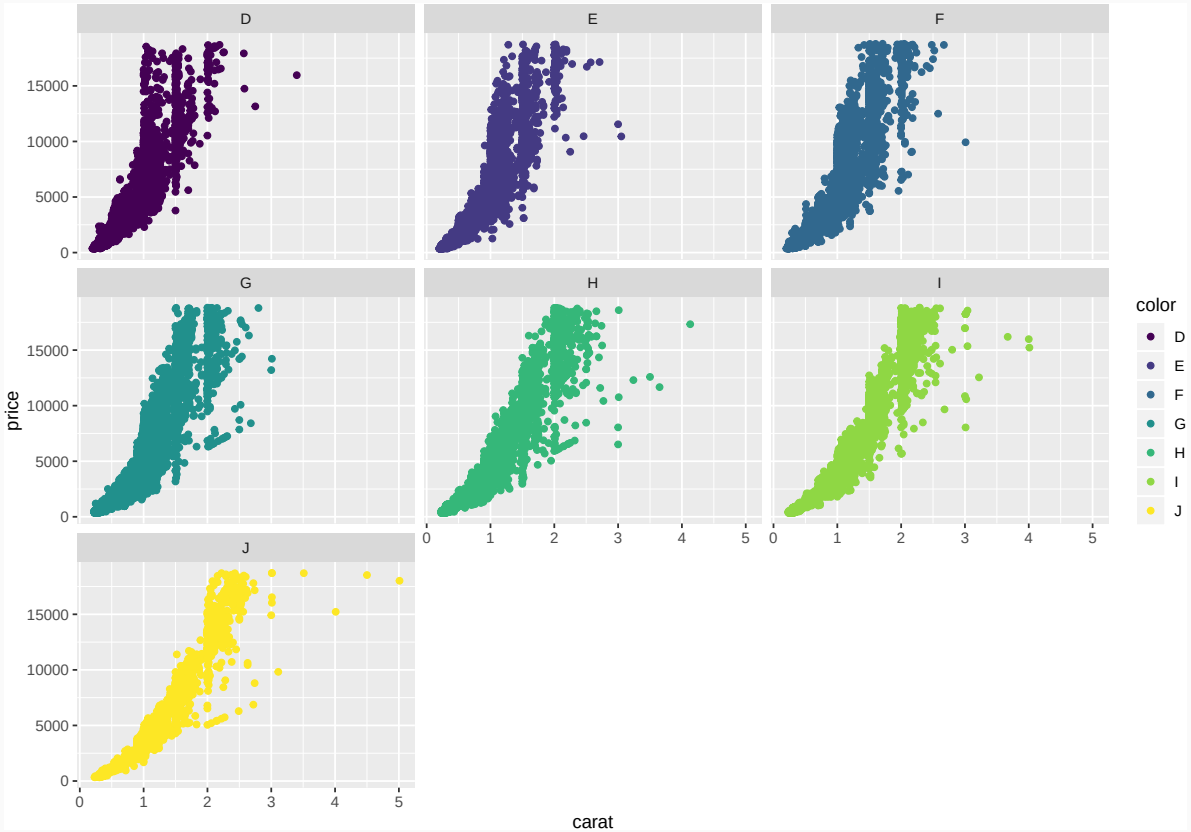


We can assign the initial call and reuse it.

`color` is a factor variable.

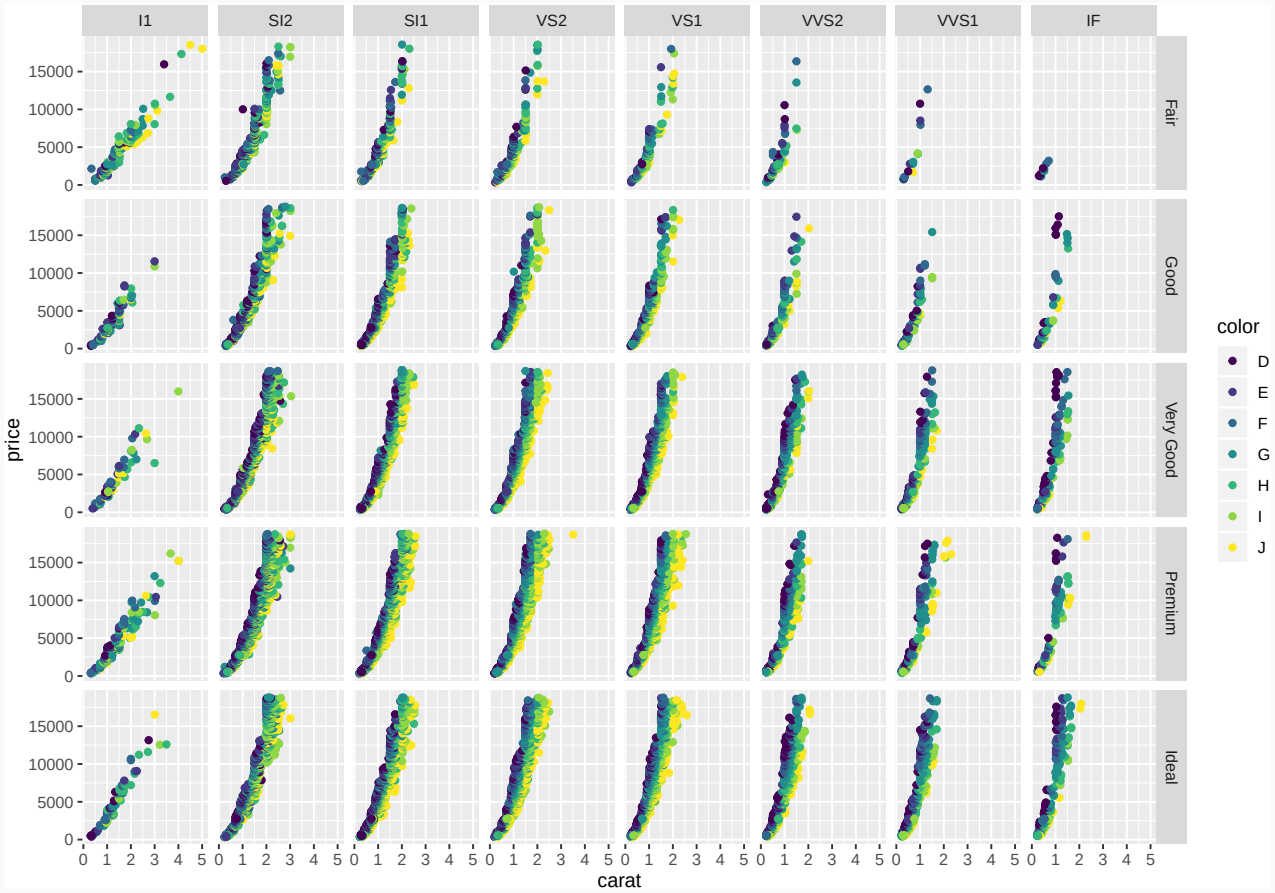
(Default color scheme switched recently and is now based on `viridis`.)

```
g + geom_point(aes(color=color)) +
  facet_wrap( ~ color)
```



Just like the `lattice` package, `ggplot2` can do conditional plots using `facet_wrap()` (and `facet_grid()`).

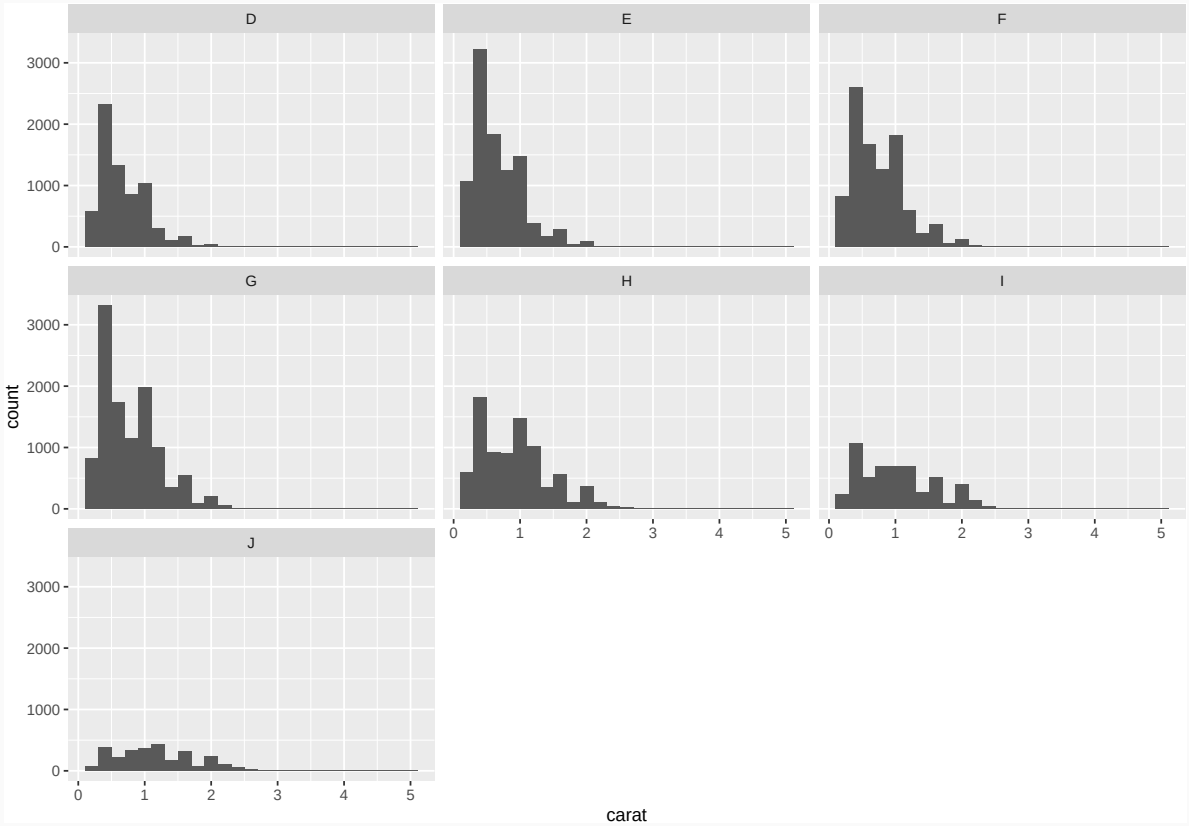
```
g + geom_point(aes(color=color)) + facet_grid(cut ~ clarity)
```



Try reordering:

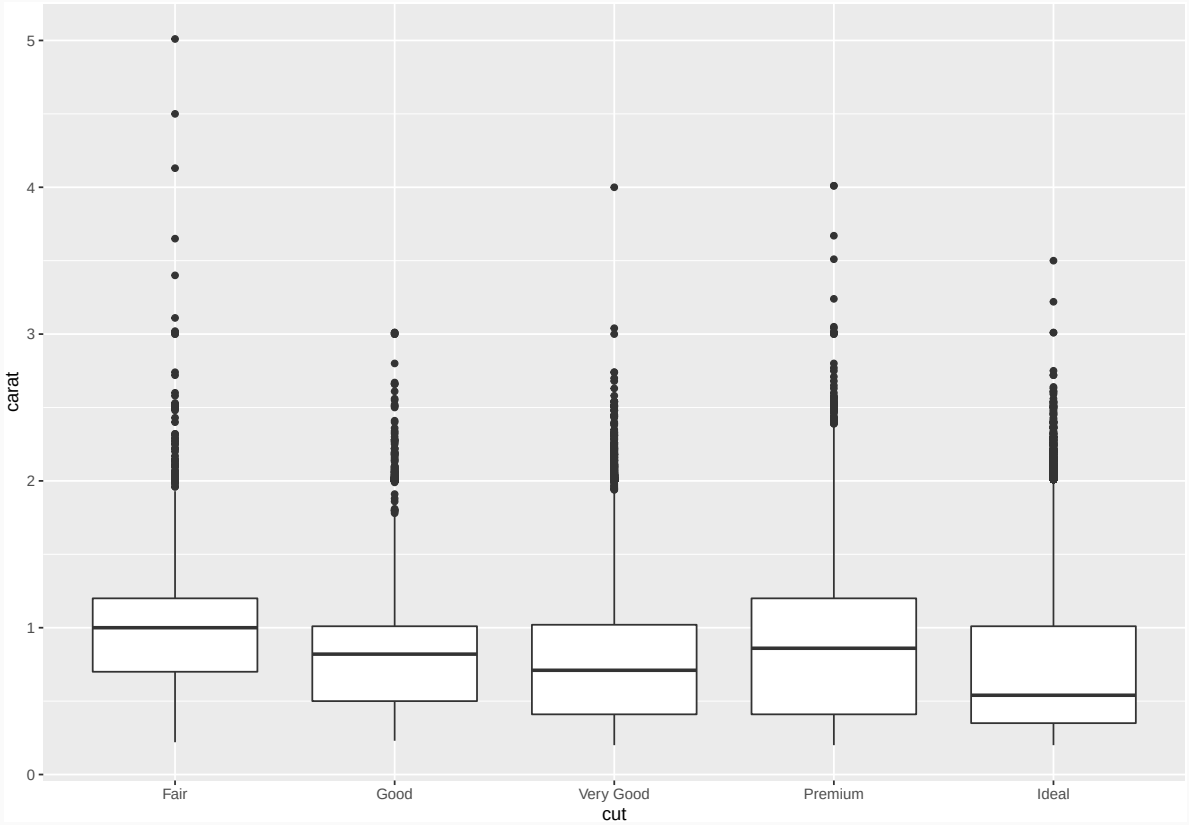
```
clarity ~ cut
```

```
ggplot(diamonds, aes(x=carat)) +
  geom_histogram(bins=25) +  facet_wrap( ~ color)
```



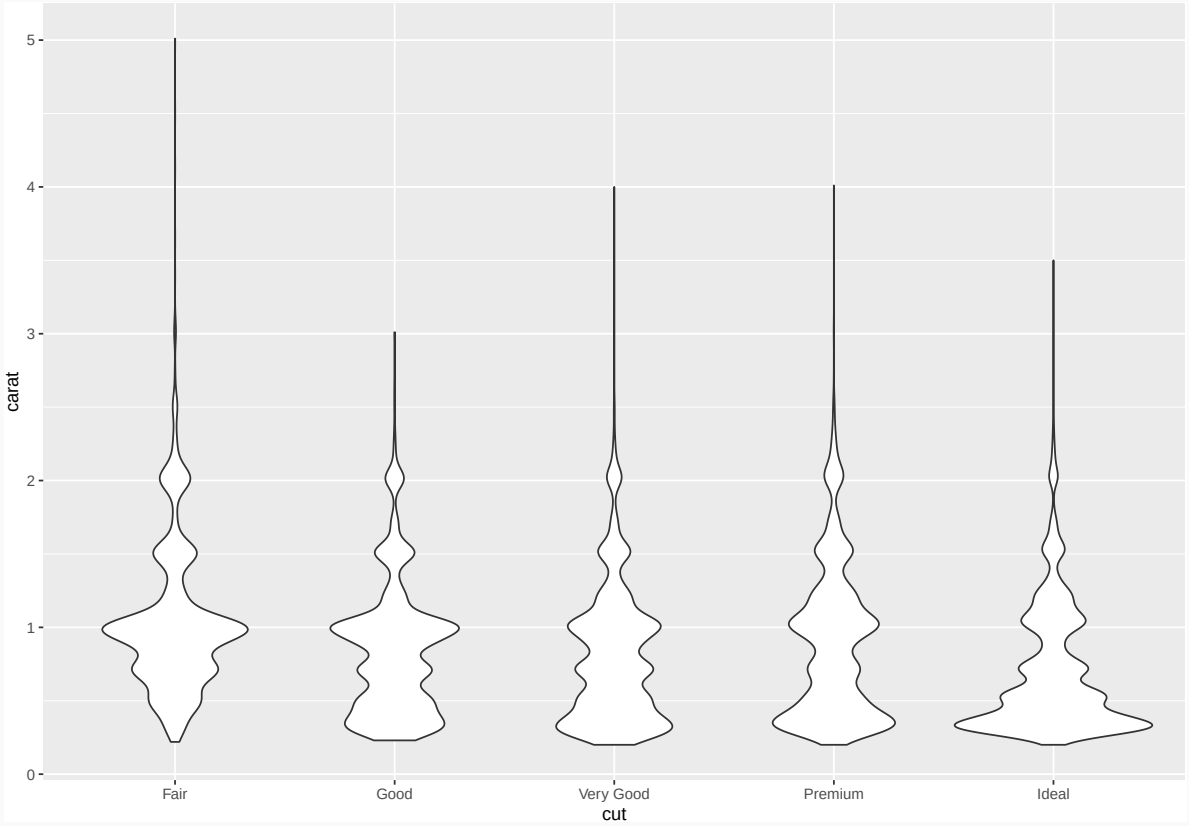
Different variables can be used for the faceting.

```
ggplot(diamonds, aes(x=cut, y=carat)) + geom_boxplot()
```



The `aes()` set `x` for categorical variable `cut`, allowing a boxplot over the numeric variable `carat`.

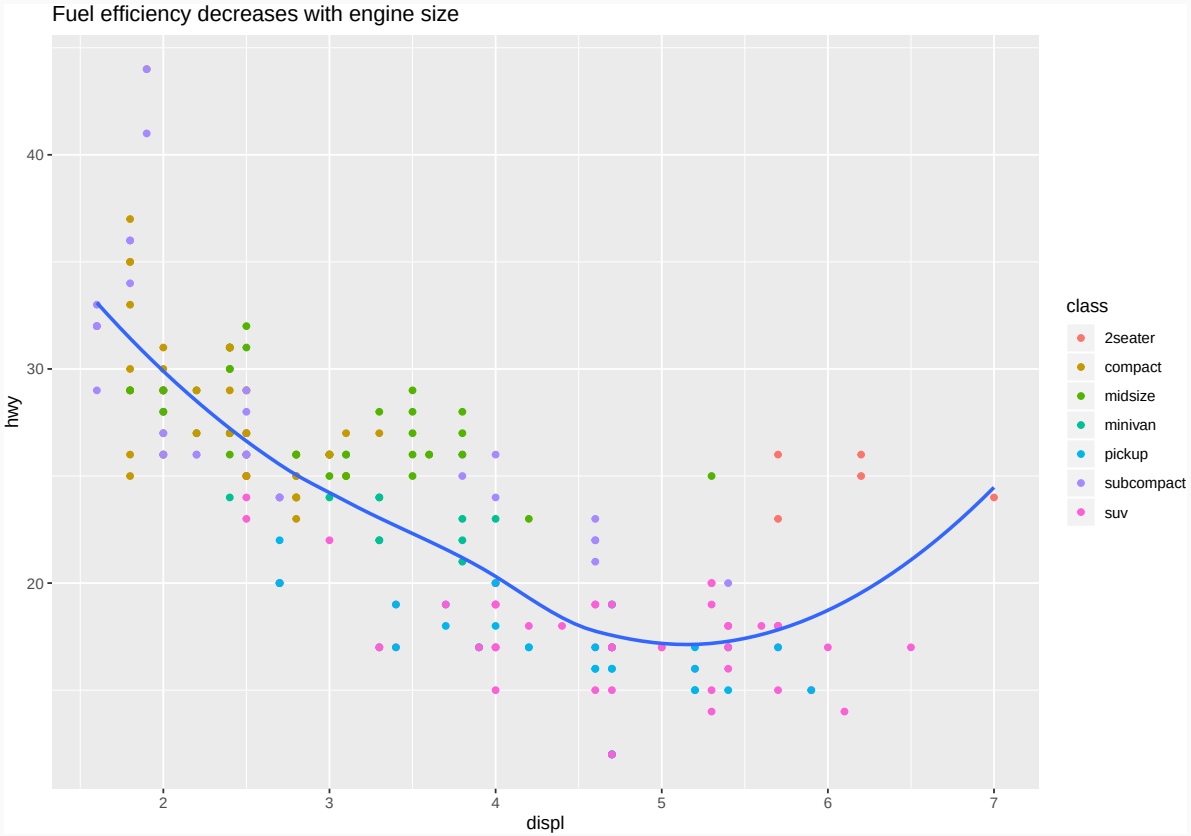
```
ggplot(diamonds, aes(x=cut, y=carat)) + geom_violin()
```



Violin plots are an alternative to boxplots. They do not show quartiles and hinges, but describe the density of the distribution.

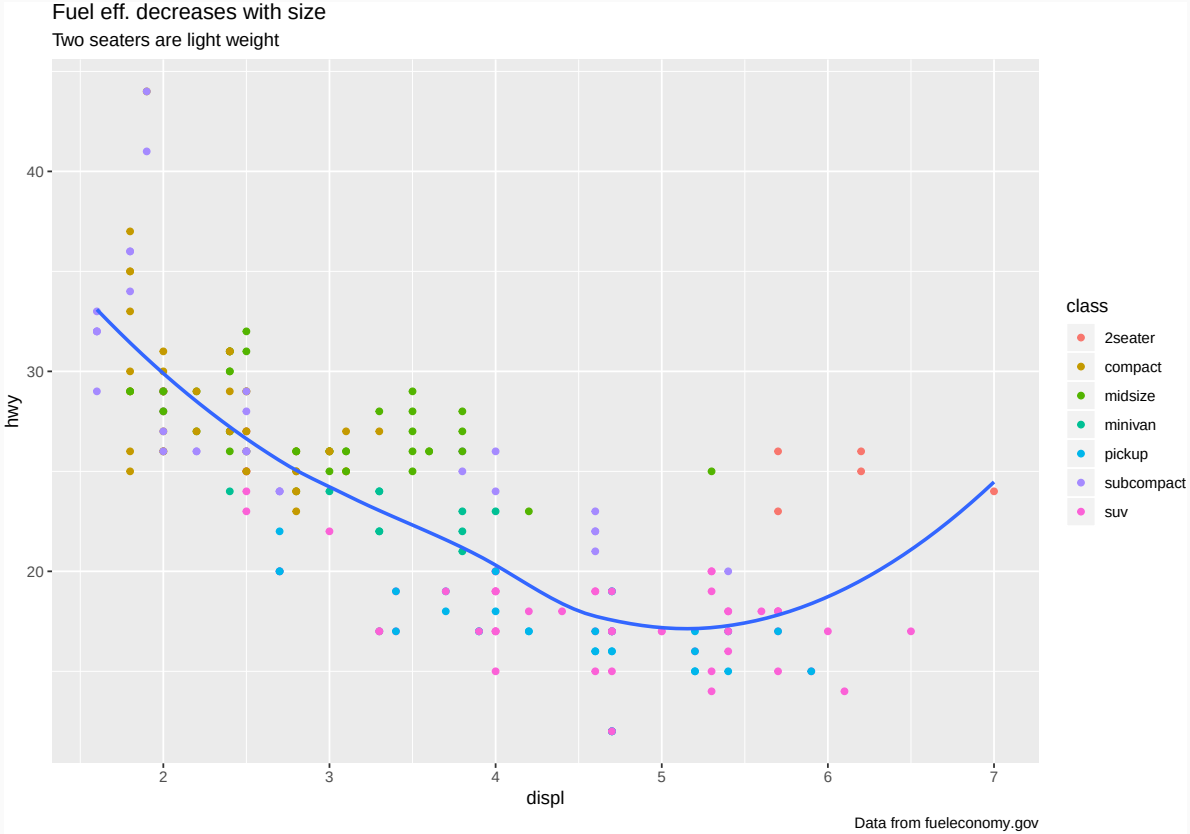
Here, we see multiple modes per cut.

```
ggplot(mpg, aes(displ, hwy)) +  geom_point(aes(color = class)) +
  geom_smooth(method="loess", se = FALSE) +
  labs(title = "Fuel efficiency decreases with engine size")
```



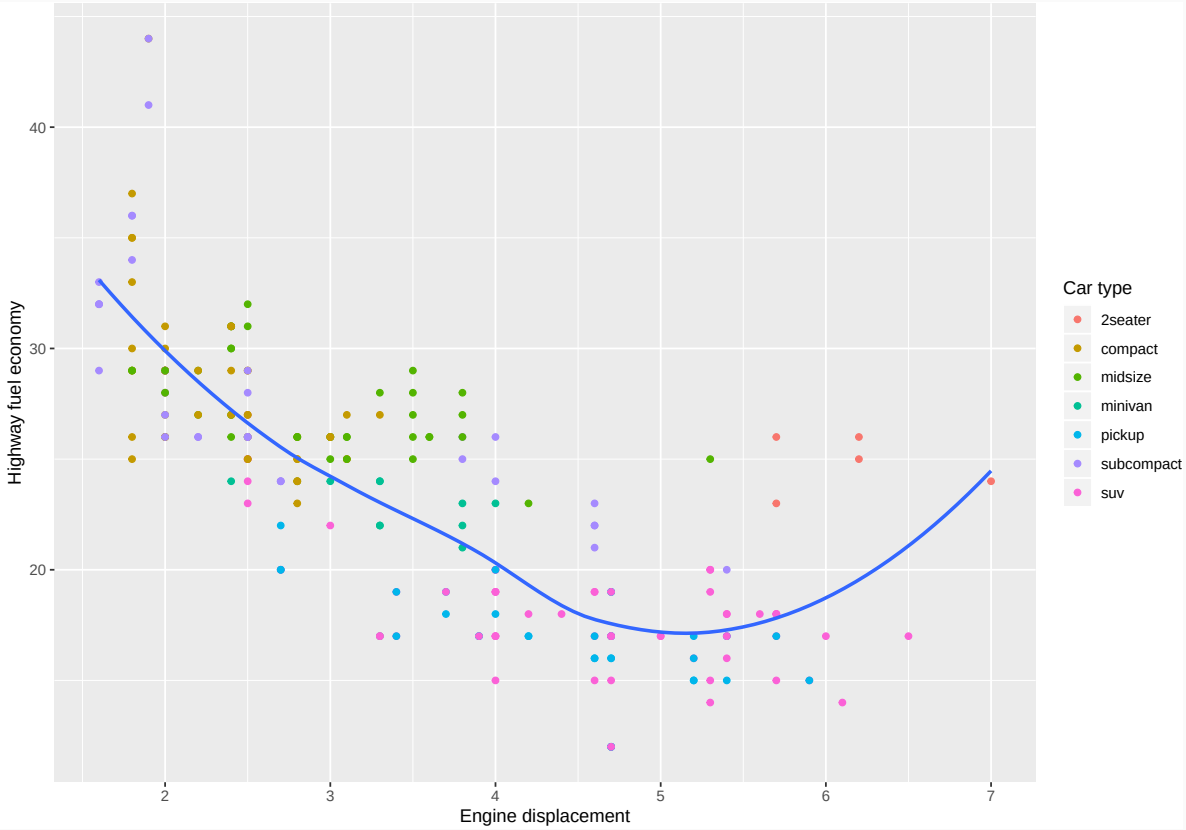
This uses a smoother (without error bands) and sets a title.

```
ggplot(mpg, aes(displ, hwy)) + geom_point(aes(color = class)) +
  geom_smooth(method="loess", se=FALSE) + labs(title="Fuel eff. decreases with size",
    subtitle = "Two seaters are light weight", caption = "Data from fueleconomy.gov")
```



Here we show subtitle and caption

```
ggplot(mpg, aes(displ, hwy)) + geom_point(aes(color = class)) +
    geom_smooth(method="loess", se = FALSE) +
    labs(x = "Engine displacement", y = "Highway fuel economy", colour = "Car type")
```



x and y-axis labels

plus legend

More Settings

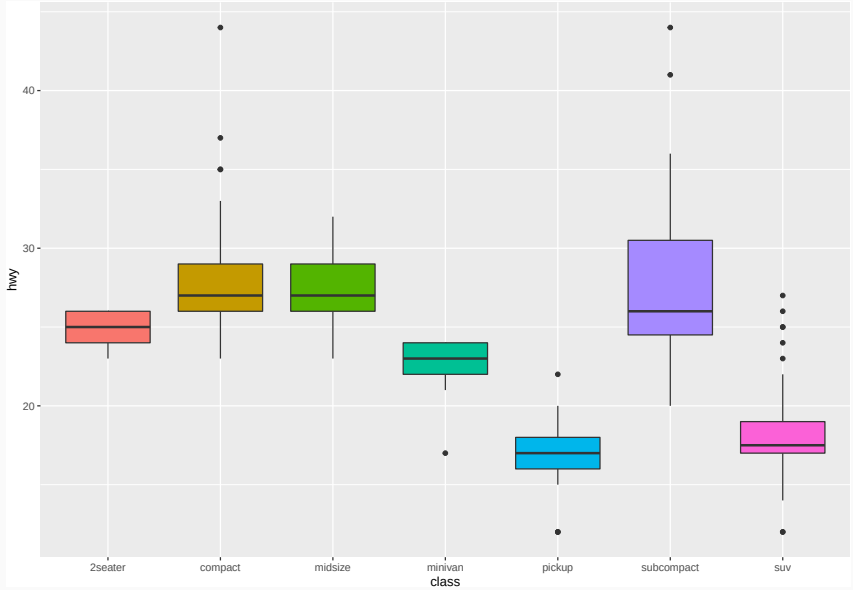
More things to adjust:

- scales (*e.g.* for logarithmic scales)
- axis ticks
- colors
- themes

There are lots of guides and blog posts out there.

Reordering Factor Variables

```
ggplot(mpg,
       aes(x=class, y=hwy, fill=class)) +
    geom_boxplot() + xlab("class") +
    theme(legend.position="none")
```

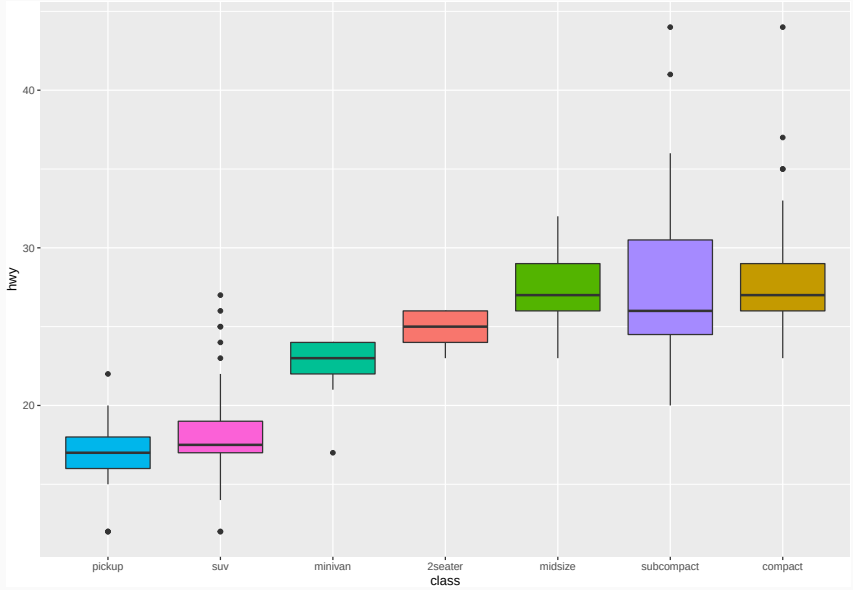


The `class` factor variable shown with a default order: alphabetical.

This is somewhat orthogonal to what we want to show so ...

Reordering Factor Variables (cont.)

```
ggplot(mpg,
       aes(x=reorder(class, hwy),
           y=hwy, fill=class)) +
    geom_boxplot() + xlab("class") +
    theme(legend.position="none")
```

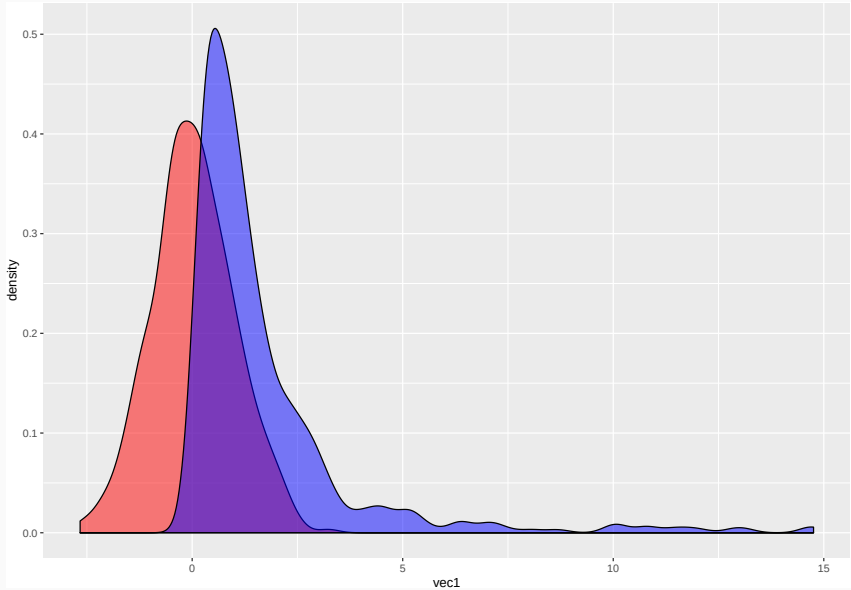


The `class` factor variable is reordered by the values of `hwy` (fuel consumption) making the plot of `hwy ~ class` more structured.

Uses `median` by default, other functions possible – and `factor` variables can of course also be reordered outside of `ggplot2`.

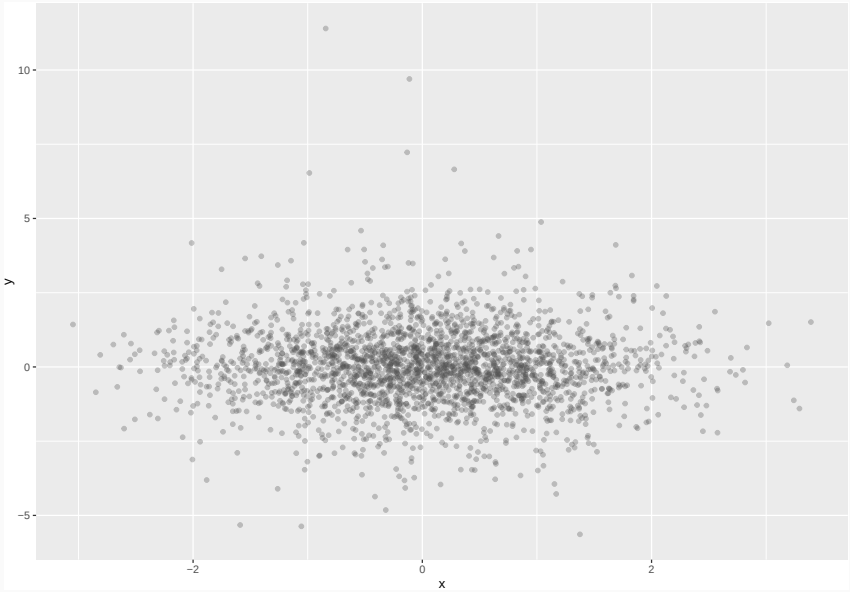
Alpha blending for overplotting

```
set.seed(123)
df <- data.frame(vec1=rnorm(700),
                 vec2=rlnorm(300))
ggplot(df) +
    geom_density(aes(x=vec1),
                 fill="red", alpha=.5) +
    geom_density(aes(x=vec2),
                 fill="blue", alpha=.5)
```

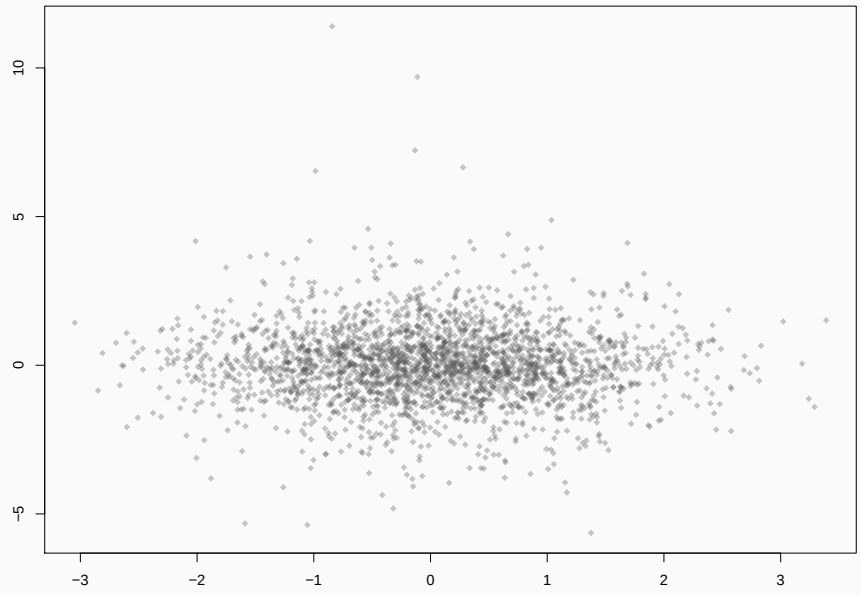


Alpha blending allows for overplotting. Also works directly for color codes as the 'fourth' part of a rgb specification: `col="#44668850"` sets 44, 66, 88 (in hex) for R, G and B – with 50 for alpha blending.

```
library(ggplot2); set.seed(123)
dat <- data.frame(x=rnorm(2500),
                  y=rt(2500,5))
ggplot(dat, aes(x=x, y=y)) +
    geom_point(color="#50505050")
```




```
set.seed(123)
par(mar=c(3,3,1,1))
dat <- data.frame(x=rnorm(2500),
                  y=rt(2500,5))
plot(dat, col="#50505050", pch=18)
```



Alpha blending via color code in **ggplot** and base R.

Additional resources

- Site at https://ggplot2.tidyverse.org/
- Reference https://ggplot2.tidyverse.org/reference/
- ggplot2 book https://www.amazon.com/dp/0387981403/ (but describes older version)
- Many tutorials online
- Kieran Healy 'SocViz' book and web site socviz.co

ggplot2 eco-systems

More ggplot2 'goodies'

- An entire site at https://www.ggplot2-exts.org/
- Some extensions are stylistic
- Some extensions are domain-specific
- But a pretty vibrant ecosystem worth checking out

SocViz and DataViz

Data Visualization: A Practical Introduction

- Recent book by Kieran Healy at Duke
- Inexpensive paperback at Amazon etc
- On-line at `http://socviz.co/`
- Code and data at `https://github.com/kjhealy/socviz`
- A very thorough yet readable introduction to visualization

Fundamentals of Data Visualization

- Recent book by ggplot2 contributor Claus Wilke at U of Texas
- Website at https://serialmentor.com/dataviz/
- Code at https://github.com/clauswilke/dataviz

Summary

Commonalities

- Both base graphics and `ggplot2` plot of a pixel canvas
- Once drawn, a plot is 'fixed' and cannot be altered
- These days, we often want interactive graphs
- That is a whole different topic which we cannot cover
- Several useful packages out there, sometimes used with `shiny` (which we will cover later)