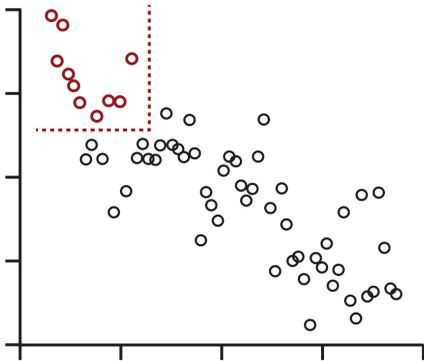


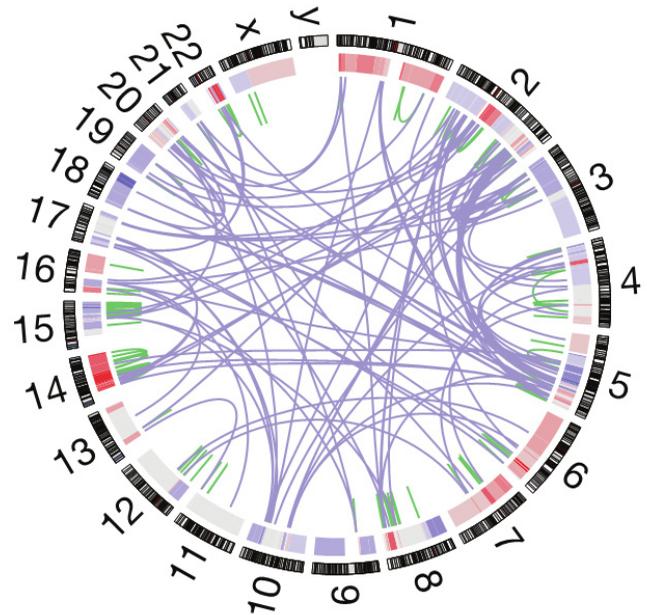
# Graphics

Our interest here is exploratory graphics.

That means not [that] pretty.



vs.



To display data, you encode the values to a visual property.

Example	Encoding	Ordered	Useful values	Quantitative	Ordinal	Categorical	Relational
	position, placement	yes	infinite	Good	Good	Good	Good
1, 2, 3; A, B, C	text labels	optional alpha or num	infinite	Good	Good	Good	Good
	length	yes	many	Good	Good		
	size, area	yes	many	Good	Good		
	angle	yes	medium	Good	Good		
	pattern density	yes	few	Good	Good		
	weight, boldness	yes	few		Good		
	saturation, brightness	yes	few		Good		
	color	no	few (<20)			Good	
	shape, icon	no	medium			Good	

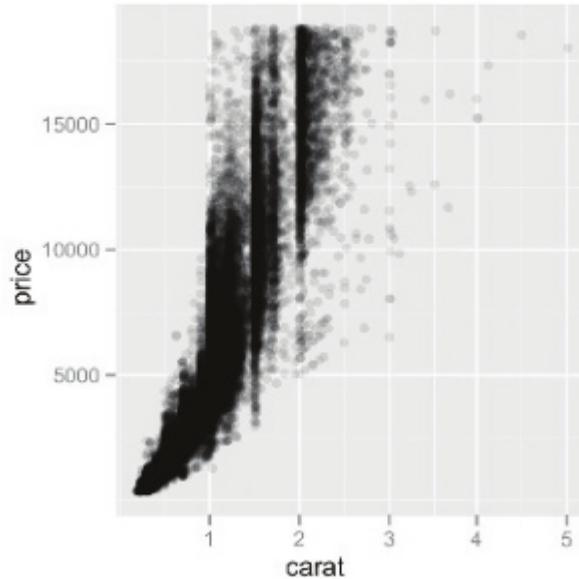
# Not all visual properties are born equal.

Example	Encoding	Ordered	Useful values	Quantitative	Ordinal	Categorical	Relational
	position, placement	yes	infinite	Good	Good	Good	Good
1, 2, 3; A, B, C	text labels	optional alpha or num	infinite	Good	Good	Good	Good
	length	yes	many	Good	Good		
	size, area	yes	many	Good	Good		
	angle	yes	medium	Good	Good		
	pattern density	yes	few	Good	Good		
	weight, boldness	yes	few		Good		
	saturation, brightness	yes	few		Good		
	color	no	few (<20)			Good	
	shape, icon	no	medium			Good	

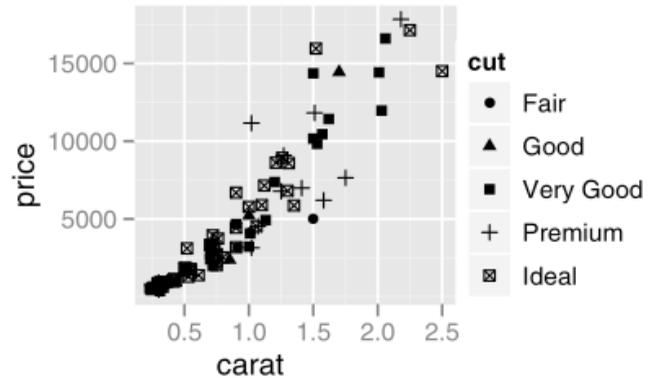
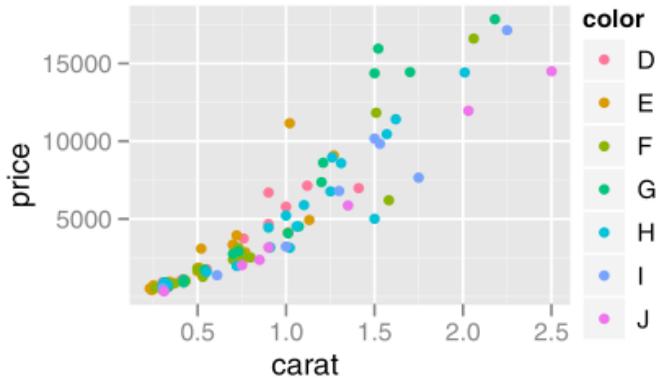
## **“Grammar of graphics”**

is a realization that you can decompose common plot types into a combination of these encodings

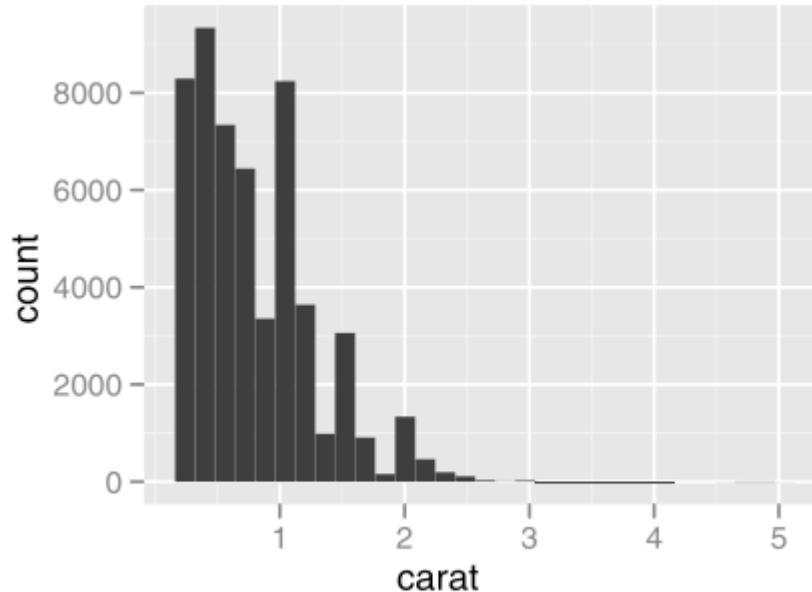
Scatter plot is a combination of two positional encodings.



Scatter plot is a combination of two positional encodings. Additional encodings are possible.



Bar chart is a combination of positional and length encodings.



## ggplot2

is a language for specifying the encodings in R

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

- maps `d$carat` to `x`, `d$price` to `y`
- adds a layer with points at `x`, `y`

## ggplot2

is a language for specifying the encodings in R

```
> ggplot(d, aes(x=carat, y=price, colour=color))  
> + geom_point()
```

- maps `d$carat` to `x`, `d$price` to `y`, `d$color` to `colour`
- adds a layer with points at `x`, `y`, colored by `colour`

## ggplot2

is a language for specifying the encodings in R

```
> ggplot(d, aes(x=carat))  
> + geom_histogram()
```

- it has shortcuts for some plot types
- maps `d$carat` to `x`
- summarizes `x` as a histogram
- adds a layer with bars at histogram bins

**ggplot2** works best with one particular way of organizing data in the tables: “**tidy data**”

- variables in columns
- observations in rows

usually if you feel like something cannot be done with ggplot2, you can solve it by reshaping your data

	Pregnant	Not pregnant
Male	0	5
Female	1	4

There are three variables in this data set.  
What are they?

pregnant	sex	n
no	female	4
no	male	5
yes	female	1
yes	male	0

from Hadley Wickham's talk on Tidy data, June 2012

## The **tidyr** package

- `gather()` gathers more columns into one
- `extract()` splits one column into more
- `spread()` complements gather

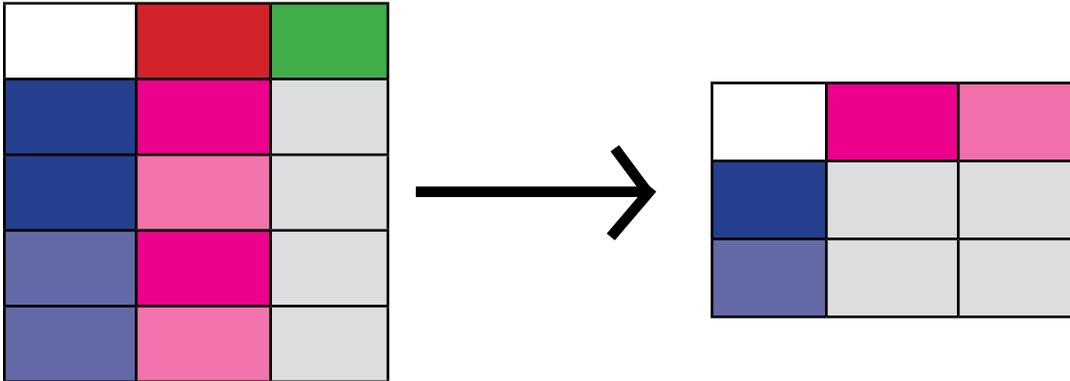
**gather(key, value, columns)**  
gathers more columns into one





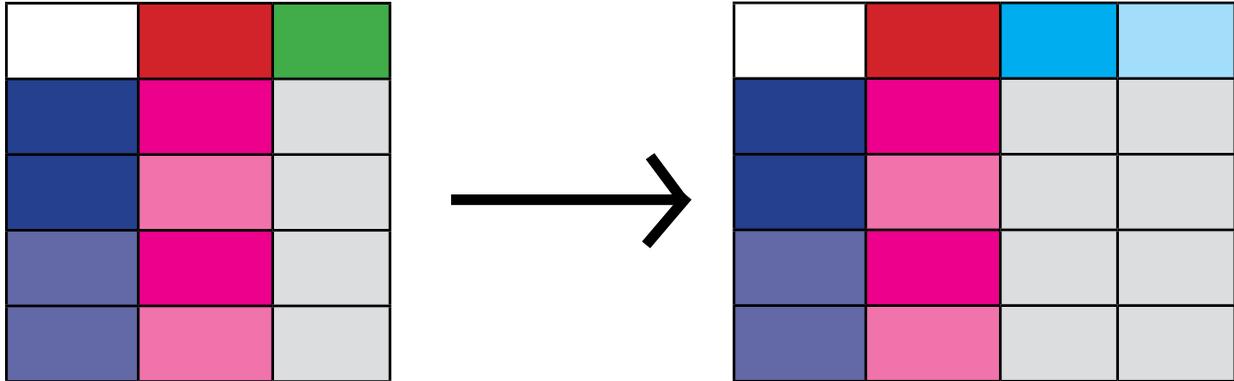
**spread(key, value)**

is the counterpart of gather



**extract(col, into, regex)**

splits a column using a regular expression



e.g. "Feb 2019" into `month` and `year` columns

more colorful explanations at:

<https://www.rstudio.com/resources/cheatsheets/>

(just google for “rstudio cheatsheets”)

# Tidying up the 'pregnancy' example

```
library(tidyverse)

data.frame(
  row.names=c('Male', 'Female'),
  Pregnant=c(0, 1),
  `Non Pregnant`=c(5, 4)) -> dd

dd %>%
  mutate(sex=row.names(.)) %>%
  gather(status, count, 1:2)
```