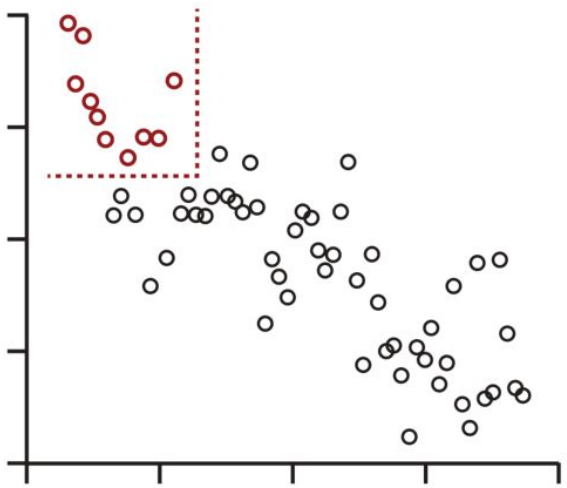
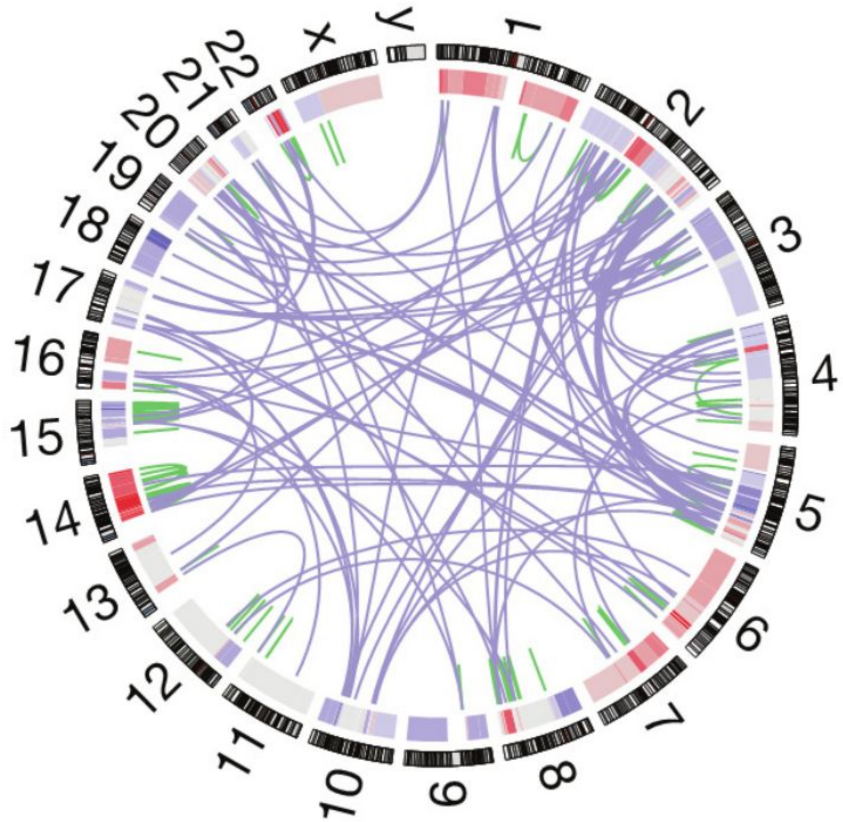
Graphics

Exploratory

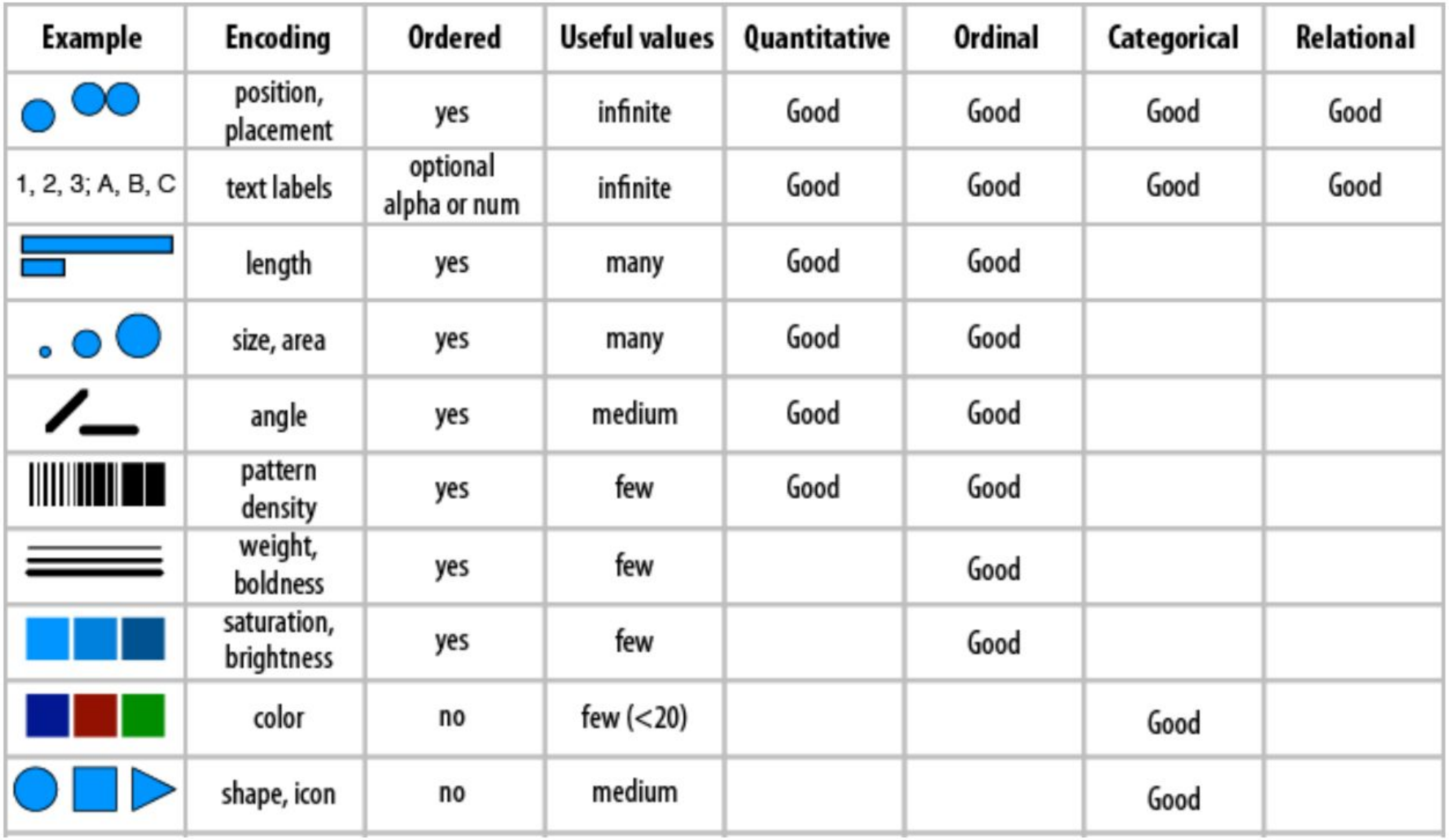
Explanatory



vs.



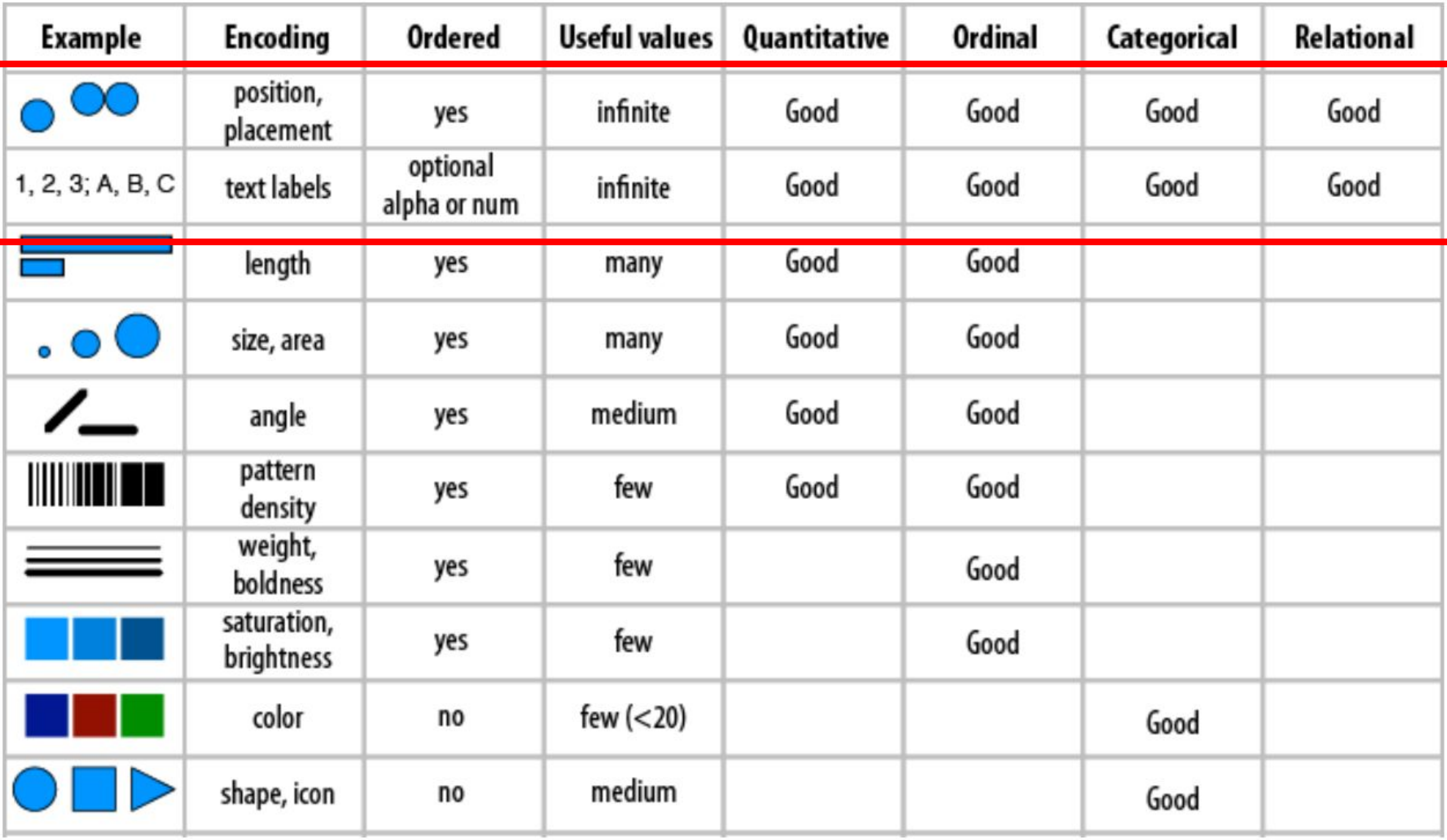
To display data, you encode the values to a visual property.



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	

Ware, Information Visualization: Perception for Design (Morgan Kaufmann), p. 179.

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



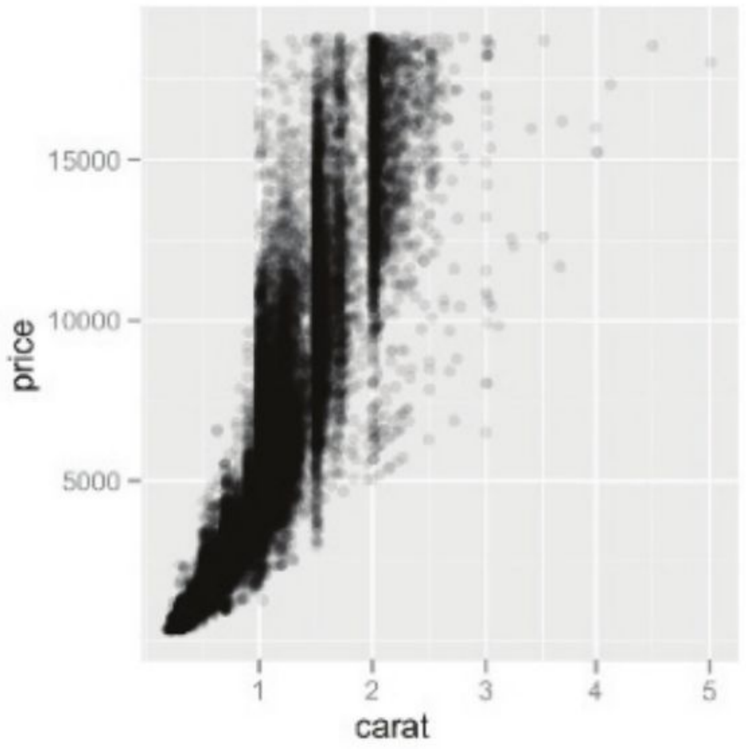
Example	Encoding	Ordered	Useful values	Quantitative	Ordinal	Categorical	Relational
● ●●	position, placement	yes	infinite	Good	Good	Good	Good
1, 2, 3; A, B, C	text labels	optional alpha or num	infinite	Good	Good	Good	Good
▬	length	yes	many	Good	Good		
• ● ⬤	size, area	yes	many	Good	Good		
∠	angle	yes	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	

Ware, Information Visualization: Perception for Design (Morgan Kaufmann), p. 179.

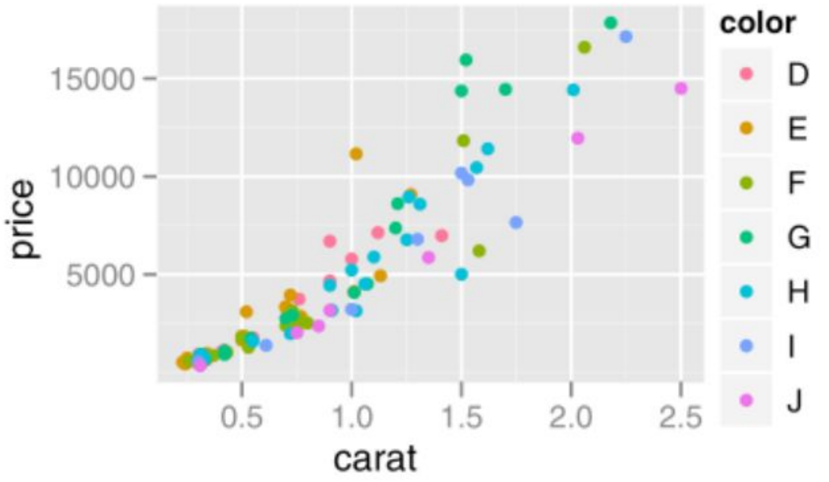
“Grammar of graphics”

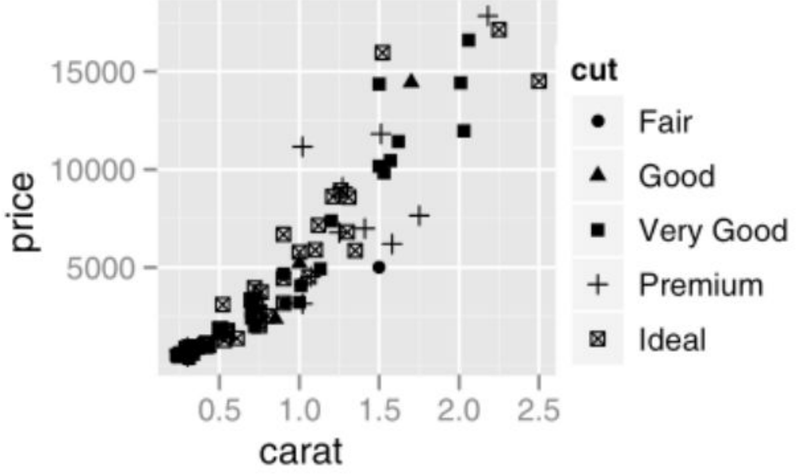
*You can decompose common plot types into a **combination of visual encodings**.*

Scatter plot is a combination of two **positional** encodings.

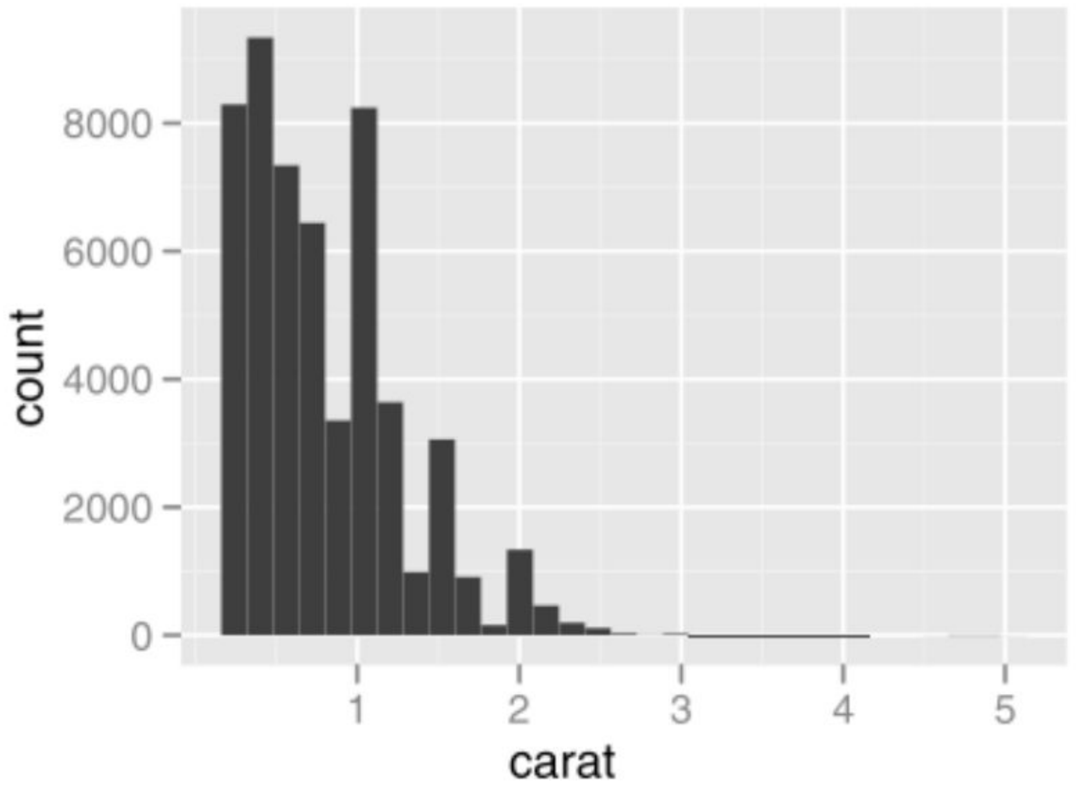


Scatter plot is a combination of two positional encodings. We can add **colour** or **shape.**





Bar chart is a combination of **positional** and **length** encodings.



ggplot2

is a language for specifying the encodings in R

ggplot2

is a language for specifying the encodings in R

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

map d$carat to x, d$price to y
+ add 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()
```

map d$carat to x, d$price to y, d$color to colour
+ add a layer with points at x, y

ggplot2 has some **shortcuts**

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

- map d$carat to x
- create bins over values of x
- count values in bins to get a histogram
- add a layer with bars at histogram bins

ggplot2

works best with

“**tidy data**”

If you feel like something cannot be done in ggplot2, the fix is often to reshape your data.

“Tidy data” is a way of organizing data in tables

- variables in columns
- observations in rows
- but what is an observation sometimes depends on your task 😉

What are the three variables in this data set?

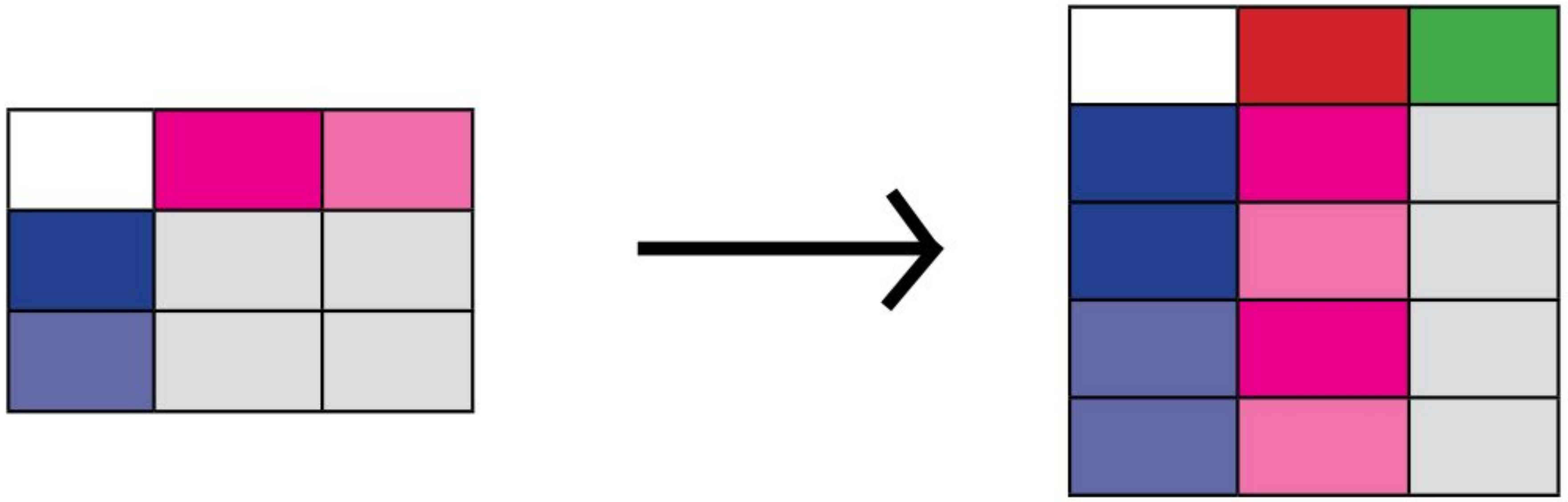
	Pregnant	Not pregnant
Male	0	5
Female	1	4

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

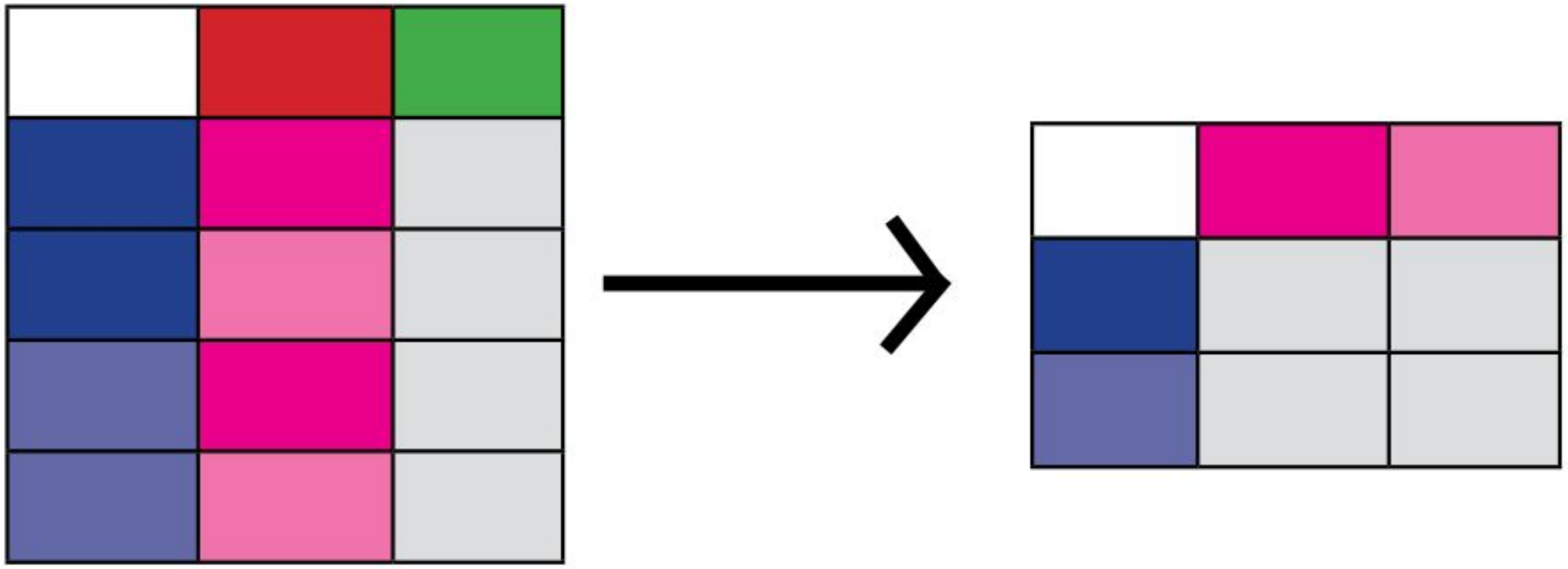
Enter the **tidyr** package

- `pivot_longer()`
  - gathers more columns into one
  - result has more rows
- `pivot_wider()`
  - spreads one column into more
  - result has fewer rows
- `extract()`
  - splits one column into more
  - keeps the same number of rows

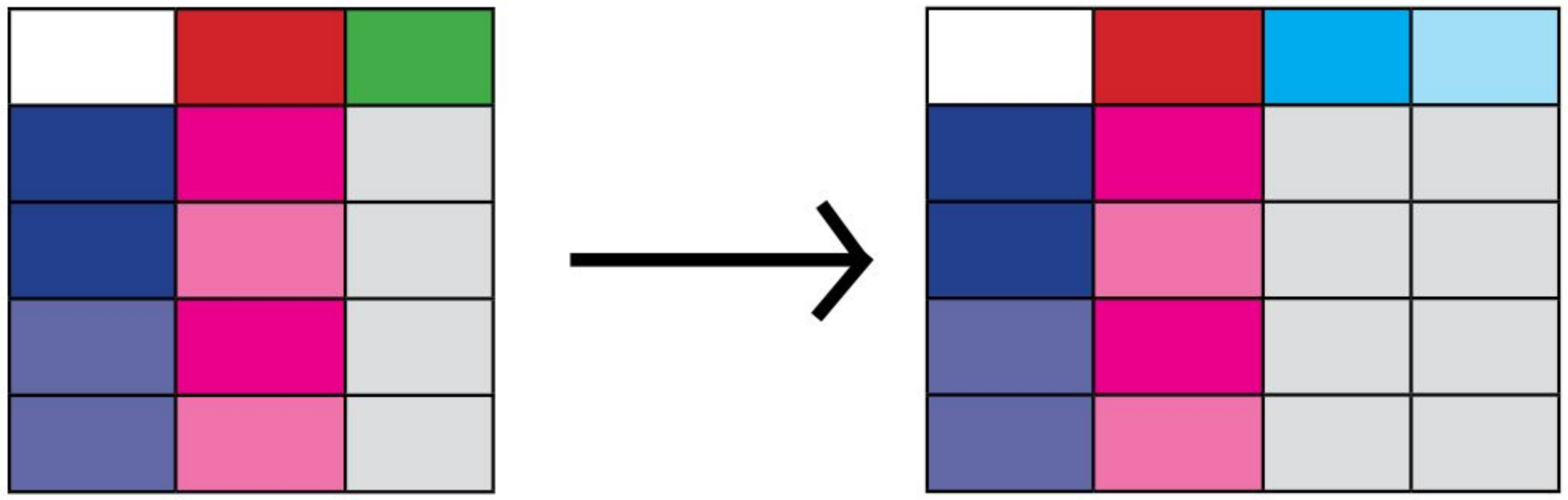
```
pivot_longer(cols,
  names_to=colname, values_to=colname)
```



```
pivot_wider(names_from=col,values_from=col)
```



`extract(col, into, regex)`



e.g. “Feb 2019” into `month` and `year` columns

For more colorful explanations

Google for “RStudio Cheatsheets”

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