USCOTS Breakout 3G:
Fundamentals of data visualization for education

Part 1: The Grammar of Graphics

Silas Bergen, Chris Malone, and Jerzy Wieczorek
Winona State University and Colby College

6/30/2021

Plan for today's workshop

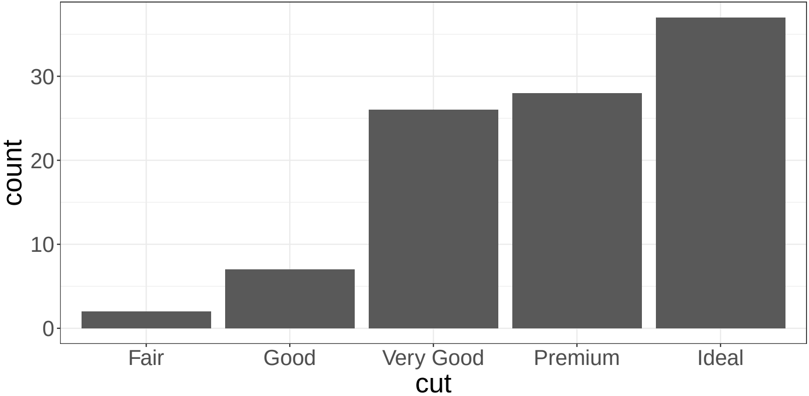
- ∼30 minutes: Jerzy introduces Grammar of Graphics
- ∼5 minute break
- ∼30 minutes: Silas introduces Gestalt principles
- ∼10 minutes for questions

Grammar of Graphics: graphic forms from the ground up

Using a subset of `diamonds` dataset from R's `ggplot2` package.

"Bar chart":

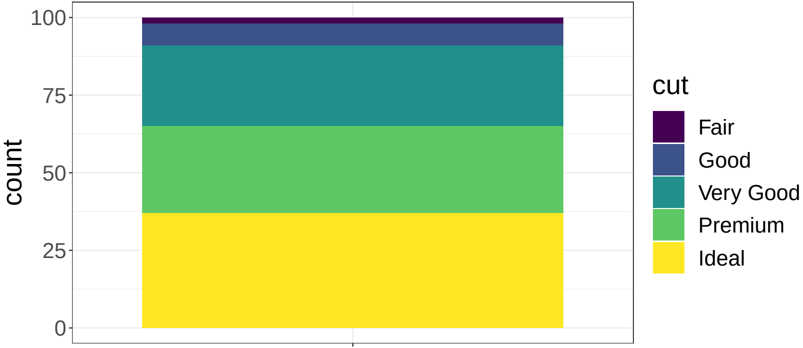
- use a different x-value for each category of `cut`
- compute counts for each category and show with bar heights



Grammar of Graphics: graphic forms from the ground up

"Spine chart" or "stacked bar chart":

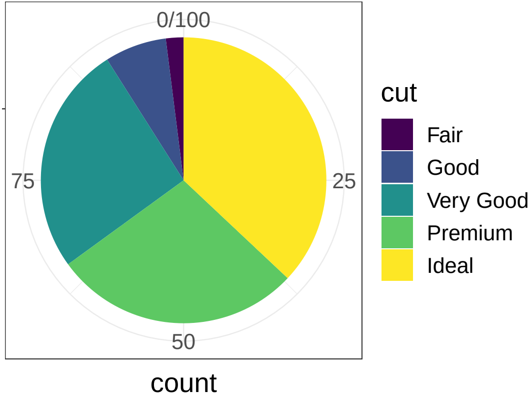
- same as before, but stack bars at **same** x-value and use different fill color for each `cut`



Grammar of Graphics: graphic forms from the ground up

"Pie chart":

- same as spine chart, but in **polar** coordinates (angle vs. radius), with counts mapped to **angle** (instead of to height) and nothing mapped to radius



Grammar of Graphics concept

Think of a data visualization or graph as a mapping

- **from variables** in the dataset, or statistics computed from the data
- **to visual attributes** (or "aesthetics") of marks (or "geometric elements") on the page/screen

The Grammar of Graphics (GoG) is a way of specifying exactly how to create a particular graph from a given dataset. It helps us to see connections between apparently unrelated graphs and to systematically design new graphs.

Grammar of Graphics specification for the “bar chart”

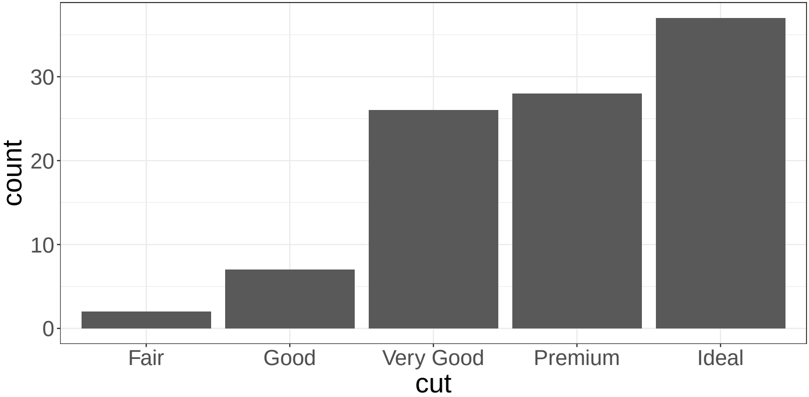
dataset: `diamonds`

“geometric element” or `geom`: bars

“statistic” or `stat`: count

“aesthetic mappings” or `aes`:

- x-axis position to show the `cut` variable
- y-axis to show the counts



Grammar of Graphics specification for the “spine chart”

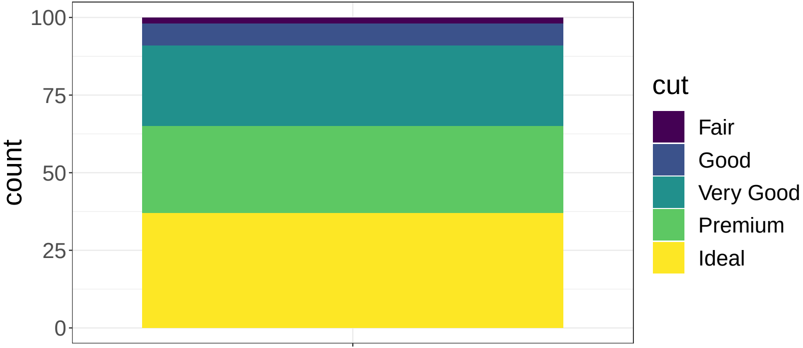
dataset: `diamonds`

“geometric element” or `geom`: bars (stacked, not overlaid)

“statistic” or `stat`: count

“aesthetic mappings” or `aes`:

- ~~x-axis position~~ **color** to show the `cut` variable
- y-axis to show the counts



Grammar of Graphics specification for the “pie chart”

dataset: `diamonds`

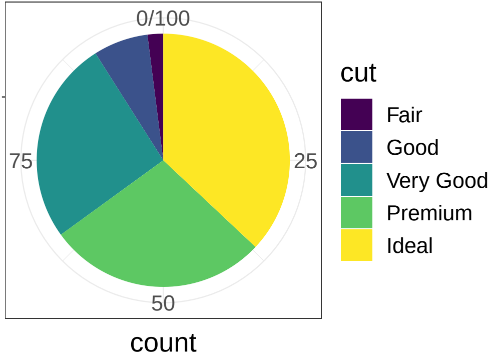
“geometric element” or `geom`: bars

“statistic” or `stat`: count

“aesthetic mappings” or `aes`:

- ~~x-axis position~~ color to show the `cut` variable
- ~~y-axis~~ **angle** to show the counts

coordinates or `coord`: polar



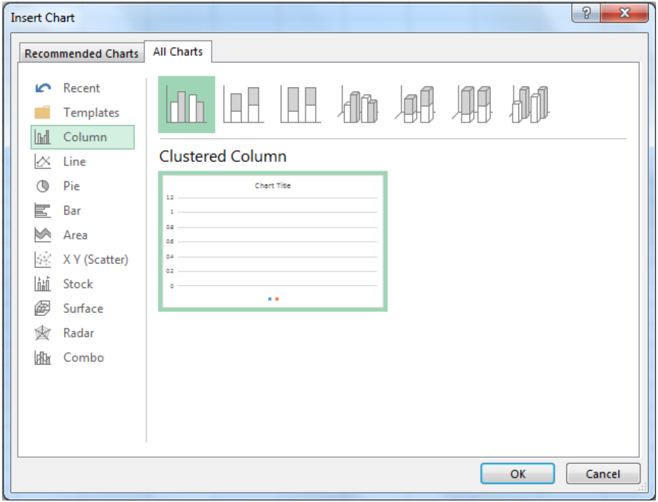
Grammar of Graphics: why bother?

It's not just a neat party trick!

- More flexible than "chart zoo" of named graphs
- Software understands the structure of your graph
  → easily automate small multiples for data subsets

Grammar of Graphics: why bother? (1) Flexibility!

Flexibly design a graph from the ground up using a grammar: compare to a fixed "chart zoo" like Excel's chart wizard



Grammar of Graphics: why bother? (1) Flexibility!

Example: What if we transform the pie chart spec further...
mapping counts to **radius** not angle, and
mapping `cut` to angle?

Grammar of Graphics: why bother? (1) Flexibility!

"Coxcomb chart" – rarely recommended, but demos GoG's power!

`dataset`: `diamonds`

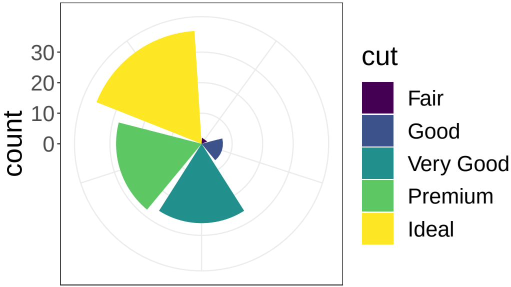
`geom`: bars

`stat`: count

`aes`:

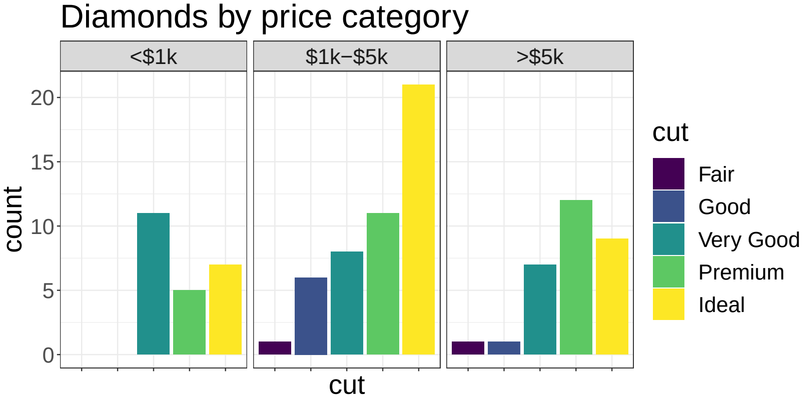
- **angle** and color to show the `cut` variable
- ~~angle~~ **radius** to show the counts

`coord`: polar



Grammar of Graphics: why bother? (2) Automation!

“Facets” are helpful: divide into sub-plots by values of a variable.
Automatically makes consistent scales and a common legend.
Faster to make, with less scope for human error.



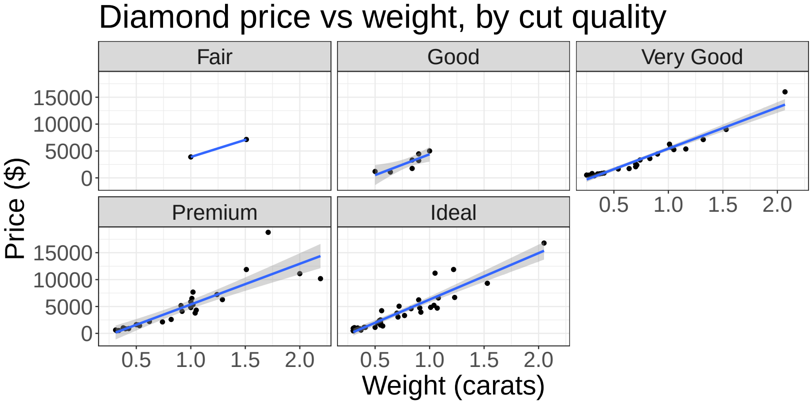
Grammar of Graphics: why bother? (2) Automation!

"Facets" are helpful: divide into sub-plots by values of a variable.

Automatically makes consistent scales and a common legend.

Faster to make, with less scope for human error.

Also can compute **stat. summaries** on-the-fly for each subgroup.



Grammar of Graphics: why bother?

"[The grammar] makes it easier for you to iteratively update a plot, changing a single feature at a time. The grammar is also useful because it suggests the high-level aspects of a plot that *can* be changed, giving you a framework to think about graphics, and hopefully shortening the distance from mind to paper. It also encourages the use of graphics customised to a particular problem, rather than relying on generic named graphics."
–Hadley Wickham, `ggplot2`

Grammar of Graphics: components

GoG components, as specified in R's `ggplot2`:

- `data`
- `aes`: aesthetic mappings (position, length, color, symbol. . .)
- `geom`: geometric element (point, line, bar. . .)
- `stat`: statistical variable transformation (identity, count, linear model, quantile. . .)
- `scale`: scale transformation (log scale, color mapping. . .)
- `coord`: Cartesian, polar, map projection. . .
- `facet`: divide into subplots / small multiples using a categorical variable

Of course, we can also control axes, legends, titles. . . ("`guides`")

Exercise: from a Grammar of Graphics spec to a graph

`data`: ages and lengths of a random sample of US children,
aged 0 to 6 months

```
## # A tibble: 209 x 4
##    ID    SEX    AGE_MO LENGTH_CM
##    <chr> <fct>   <dbl>     <dbl>
##  1 62207 Male        0      57.2
##  2 62216 Male        6      70.5
##  3 62238 Female      4      66.7
##  4 62246 Female      1      55.5
##  5 62358 Female      3      61
##  6 62438 Male        0      58.6
##  7 62451 Female      4      65.7
##  8 62490 Male        6      69.1
##  9 62520 Male        0      59.9
## 10 62540 Female      3      66.2
## # ... with 199 more rows
```

Exercise 1: from a Grammar of Graphics spec to a graph

Sketch what this plot might look like:

`data`: ages and lengths of a random sample of US children,
ages 0 to 6 months

`aes`: `x = AGE_MO`, `y = LENGTH_CM`

`geom`: point

`stat`: identity (no transformation)

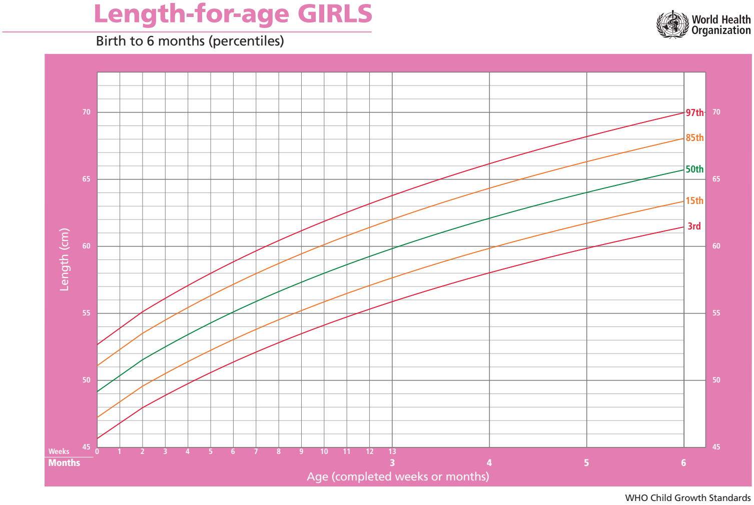
`scales`: identity

`coord`: cartesian

`facet`: by `SEX`

Exercise 2: from a graph to a Grammar of Graphics spec

WHO Child Growth Standards chart: Length-for-age %iles for girls.

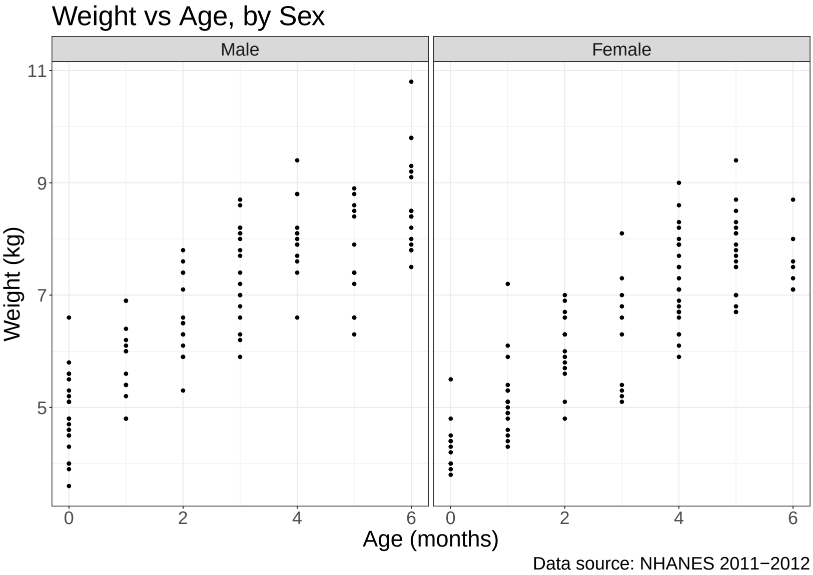


Exercise 2: from a graph to a Grammar of Graphics spec

WHO Child Growth Standards chart: Length-for-age %iles for girls.

`data`: ?
`aes`: ?
`geom`: ?
`stat`: ?
`scales`: ?
`coord`: ?
`facet`: ?

Exercise 1: from a Grammar of Graphics spec to a graph



Exercise 2: from a graph to a Grammar of Graphics spec

WHO Child Growth Standards chart: Length-for-age %iles for girls.

APPROACH 1:
Use the original data,
and calculate summaries as part of plotting.

`data`: ages and lengths of a random sample of US children,
ages 0 to 6 months
`aes`: `x = AGE_MO`, `y = LENGTH_CM`, `color = percentile`
`geom`: line
`stat`: quantile regression of `y` on `x`
`scales`: manual x-axis and color scales
`coord`: cartesian
`facet`: none

Exercise 2: from a graph to a Grammar of Graphics spec

WHO Child Growth Standards chart: Length-for-age %iles for girls.

APPROACH 2:
First pre-calculate the percentiles of length at each age.
Then plot this data summary.

`data`: ages and length-percentiles for US children,
ages 0 to 6 months
`aes`: `x = AGE_MO`, `y = LENGTH_CM`, `color = percentile`
`geom`: line
`stat`: identity
`scales`: manual x-axis and color scales
`coord`: cartesian
`facet`: none

Grammar of Graphics: history and influence

- Leland Wilkinson, *The Grammar of Graphics* book (1st ed. 1999)
- Hadley Wickham, `ggplot2` in R (2005)
- Tableau
- SPSS Graphics Production Language (GPL) and Visualization Designer
- IBM VizJSON
- D3.js
- Python's `seaborn`, `plotnine`, etc.
- and many others. . .

Grammar of Graphics: more resources

- Wilkinson's book *The Grammar of Graphics*, especially last chapter "Coda"
- Wickham's book `ggplot2`, especially Ch 3-4

Next:

Take a 5 minute break!

When we return:

Now you know how the Grammar of Graphics concept can help you think flexibly about graphs.

So how do you actually make those decisions?
(i.e. what goes on x- vs y-axis,
or what goes on facets vs color?)

Silas will discuss Gestalt principles to help you make those choices.