# The Grammar of Graphics and ggplot2

## Pittsburgh useR Group

Jerzy Wieczorek

12/8/15

# Grammar of Graphics

Why bother?

- Framework flexibility (not a premade chart zoo)
- Automatic guides (legends, axes, etc.)
- Facets (consistent small-multiples of data subsets)

Challenges

- Framework limitations
  (GoG intentionally won't do everything you may want)
- Implementation
  (finicky control over legends etc.)
- Interaction (though see `animint`, `ggvis`, Tableau)

# Grammar of Graphics: examples

Let's demonstrate on a small subset of `diamonds` dataset
that comes with `ggplot2`
(with black-and-white theme, and larger font,
and darker color palette)

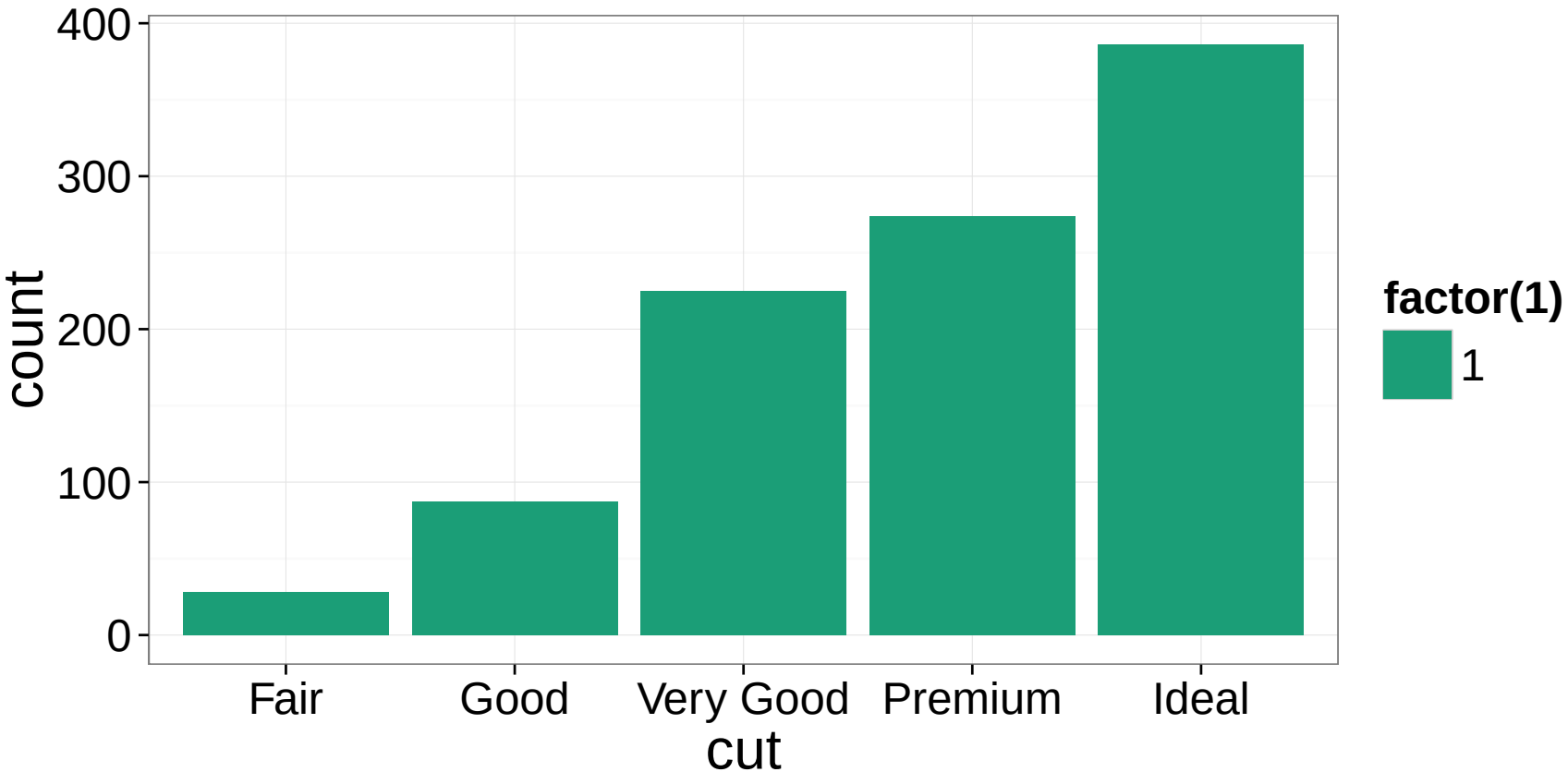
# Grammar of Graphics: examples

```
library(ggplot2)
dsmall = diamonds[sample(nrow(diamonds), 1000), ]
theme_set(theme_bw() +
            theme(text = element_text(size = 24)))
scale_fill_discrete <- function(...) {
  scale_fill_brewer(... , palette = "Dark2")
}
```

# Grammar of Graphics: examples

"Bar chart": map discrete variable to x-axis; use constant fill color; compute counts-by-category, and map them to bar heights

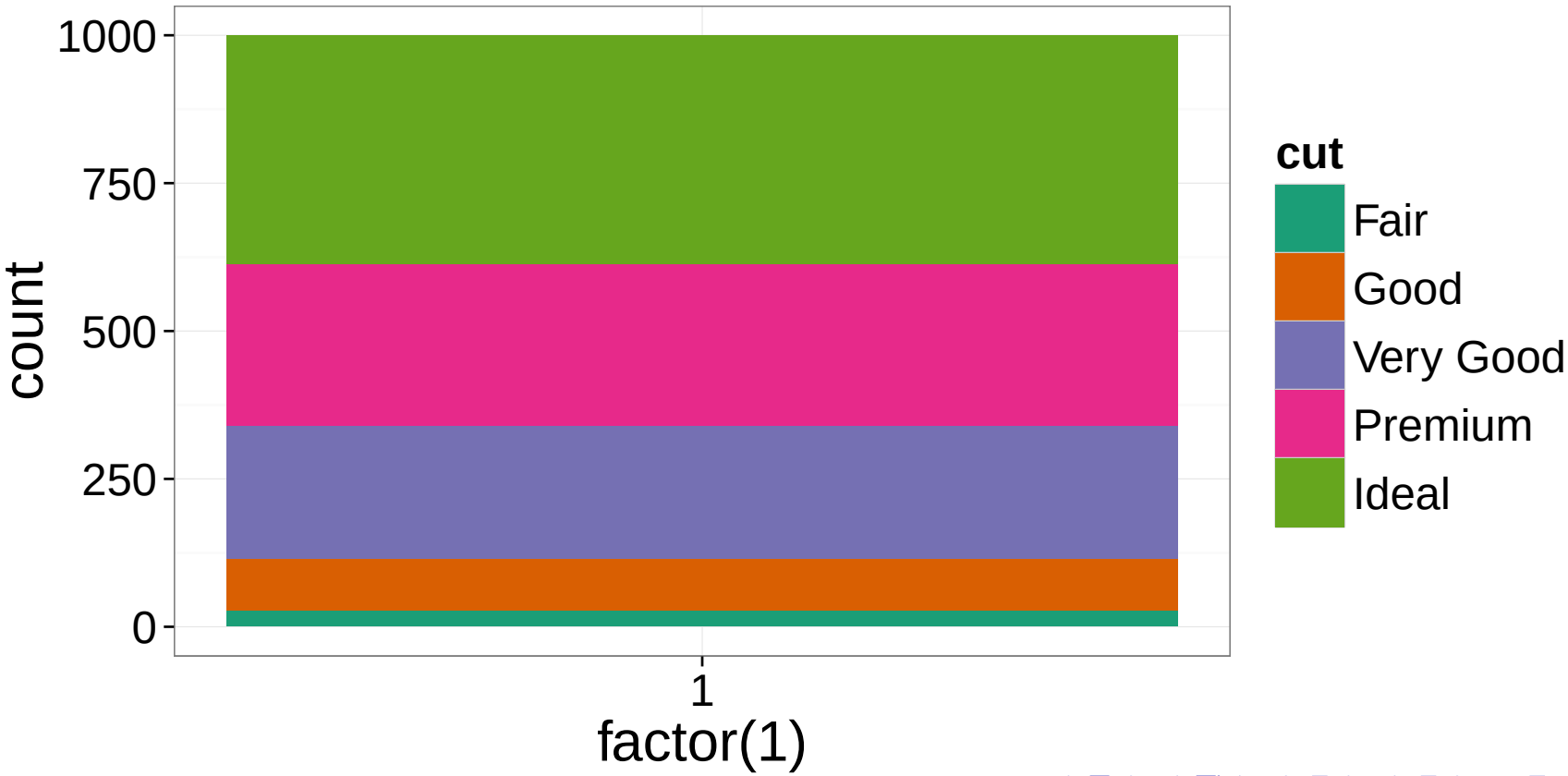
```
ggplot(data = dsmall, aes(x = cut, fill = factor(1))) +
  geom_bar(stat = "bin") + coord_cartesian()
```



# Grammar of Graphics: examples

"Stacked bar chart": set constant x-axis value; map discrete variable to fill color; map counts-by-category to bar-segment heights

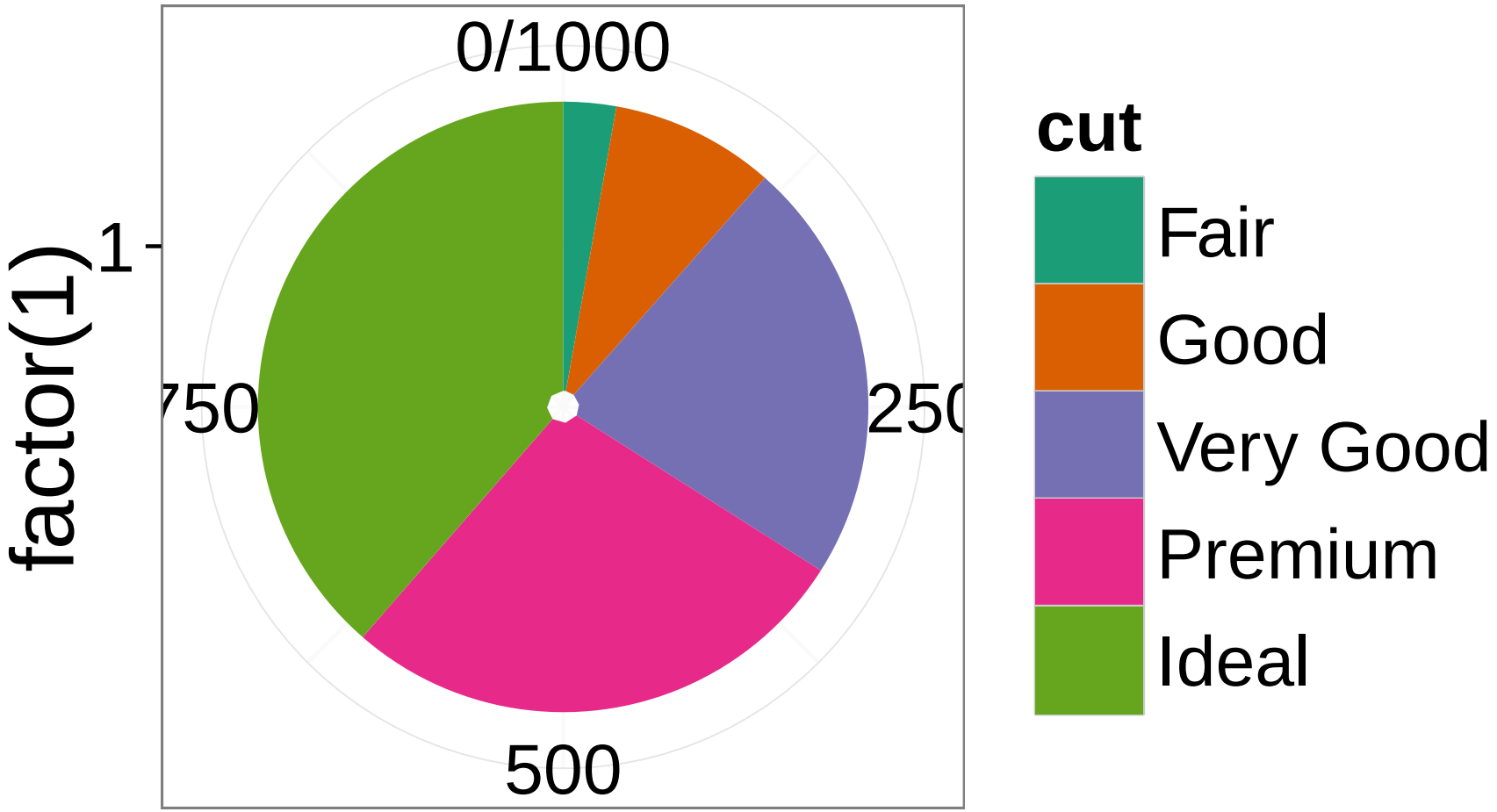
```
ggplot(data = dsmall, aes(x = factor(1), fill = cut)) +
  geom_bar(stat = "bin") + coord_cartesian()
```



# Grammar of Graphics: examples

"Pie chart": switch to polar coordinates, with constant radius 1, and map counts to angles

```
ggplot(data = dsmall, aes(x = factor(1), fill = cut)) +
  geom_bar(stat = "bin") + coord_polar(theta = "y")
```

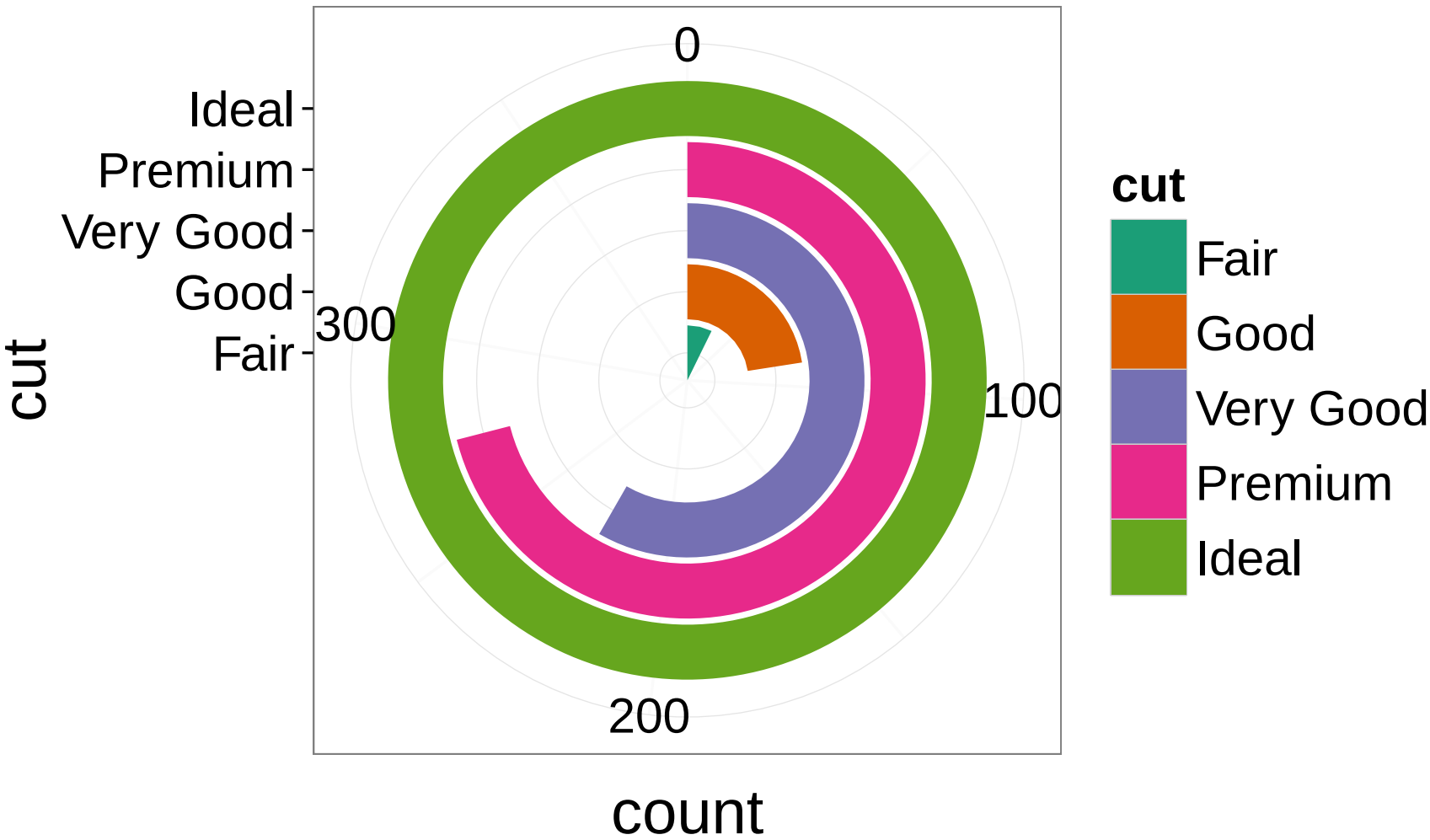


# Grammar of Graphics: examples

So. . . what if we map discrete variable to color **and radius** instead?

# Grammar of Graphics: examples

```
ggplot(data = dsmall, aes(x = cut, fill = cut)) +
  geom_bar(stat = "bin") + coord_polar(theta = "y")
```



# Grammar of Graphics: examples

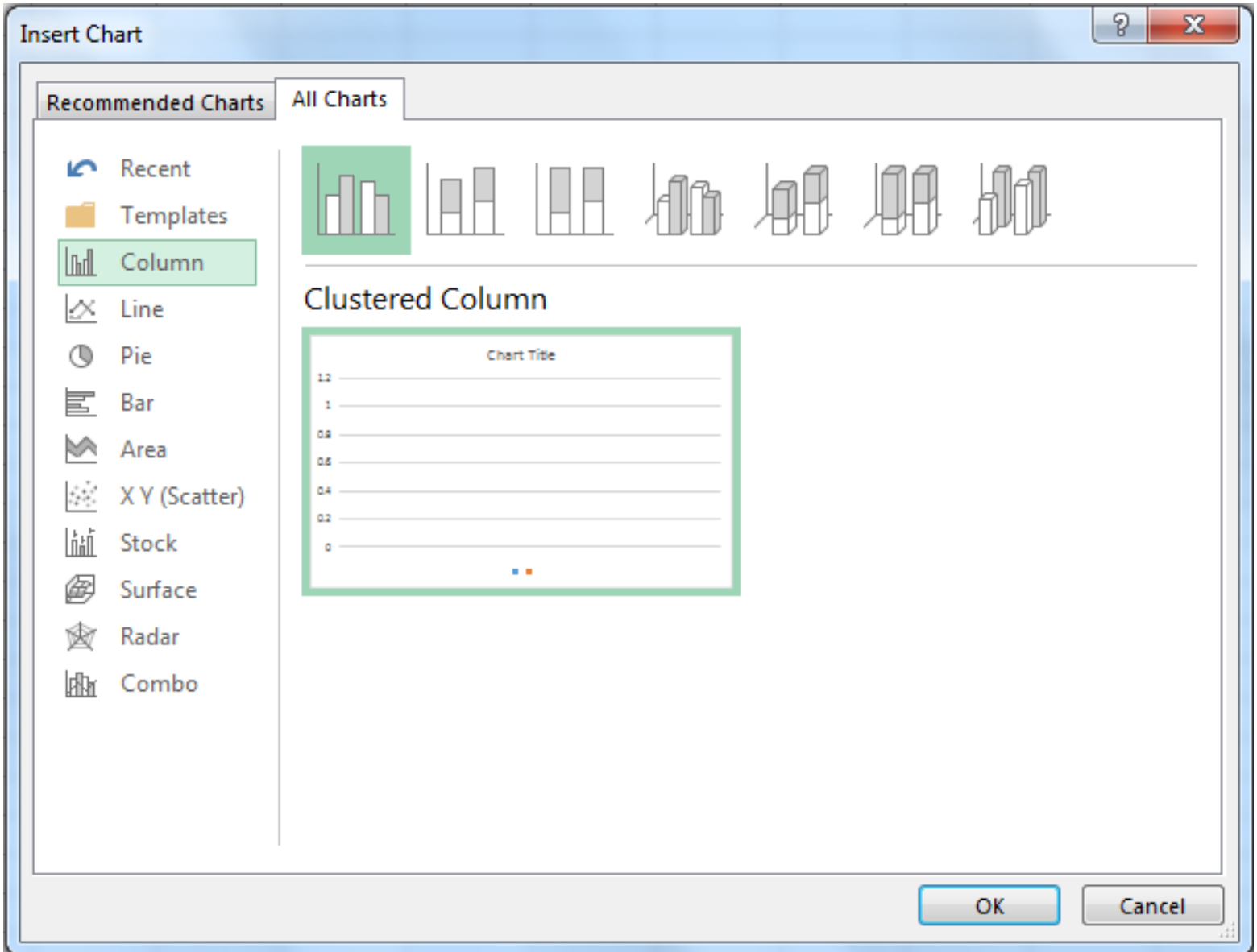
"Race track plot"?
Terrible idea :) but nifty example of GoG's flexibility.

"This system is capable of producing some hideous graphics . . .
This system cannot produce a meaningless graphic, however."
–Leland Wilkinson, *The Grammar of Graphics*

We'll see nicer examples in R demo soon!

# Grammar of Graphics: why bother?

Expressing a graph from the ground up is more flexible than "chart zoo" approach (like Excel's chart wizard)



# Grammar of Graphics: why bother?

"The grammar is useful for you both as a user and as a potential developer of statistical graphics. As a user, it 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: history and influence

- Leland Wilkinson, *The Grammar of Graphics*
- R's `ggplot2` (Hadley Wickham)
  - yeroon.net/ggplot2, web GUI for `ggplot2`
  - `animint` and `ggvis`, extensions that add interactivity
- Tableau (Wilkinson now works there)
- SPSS Graphics Production Language (GPL) and Visualization Designer
- IBM VizJSON
- SAS JMP Graph Builder
- D3.js and Vega
- Python's `ggplot`

# Grammar of Graphics: components

Wilkinson's grammar:

- `data`
- `trans`: variable transformation
  (identity, bin, smooth, quantile...)
- `scale`: scale transformation
  (axis limits, log scale, color mapping...)
- `coord`: Cartesian, polar, map projection...
- `element`: graphic element (point, line, bar...)
  with attributes (color, symbol, length...)
- `guide`: axes, legends, titles...

# Grammar of Graphics: components

`ggplot2` specifications:

- `data`
- `aes`: aesthetic attributes (position, length, color, symbol...)
- `stat`: statistical variable transformation
  (identity, bin, smooth, quantile...)
- `geom`: geometric element (point, line, bar...)
- `scale`: scale transformation
  (axis limits, log scale, color mapping...)
- `coord`: Cartesian, polar, map projection...
- `facet`: divide into small multiples using a discrete variable

# Grammar of Graphics: components

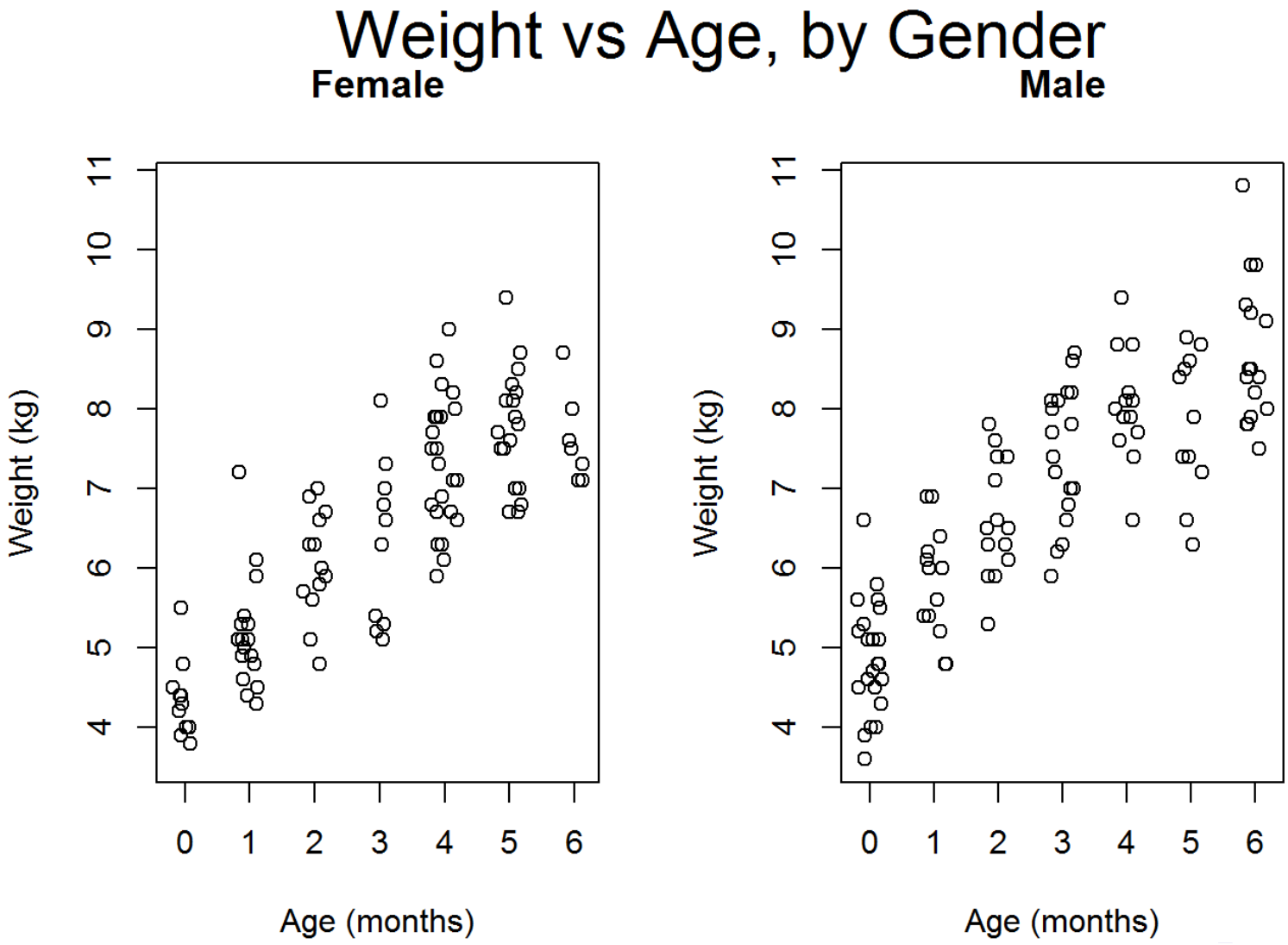
More on `ggplot2` specifications:

- Each layer has its own `data`, `aes`, `stat`, and `geom` ... then the `scale` and `coord` are coordinated across `facets`
- Sometimes can also specify `position` adjustments
- Finer control over stat summaries with `group`: see documentation, `Oxboys` example
- Of course can also control guides (axes, legends, titles...)

# Grammar of Graphics: practice

Example base R plot, NHANES data:

What `data` map to which `aes` here? What `stat`, `geom`, `scale`, `coord` are used? Any `facet`?



# Grammar of Graphics: practice

WHO Child Growth Standards,
charts of Length-for-age, percentiles, by gender

What `data` map to which `aes` here? What `stat`, `geom`, `scale`, `coord` are used? Any `facet`?
(Consider Boy and Girl plots side-by-side.)

# Tableau

Polished implementation of Wilkinson's "graphboard" idea

Student license (1 year free)

# ggplot2

Follow along:

- Editable code in `GoG_code.R`
- Code with output examples in `GoG_code.html`

We won't cover `qplot()`, a `ggplot2` wrapper function that acts more like base R, because it doesn't help explain the GoG concept.

# Grammar of Graphics and ggplot: more resources

- `ggplot2` official documentation
- `ggplot2` cheat sheet
- StackOverflow help for `ggplot2`
- A nice `ggplot2` tutorial
- Wickham's book `ggplot2`, especially Ch 3-4
  [Amazon, or free on Springer Link through CMU or Pitt]
- Chang's *R Graphics Cookbook*, mostly on `ggplot2` [Amazon]
- Wilkinson's book *The Grammar of Graphics*, esp. "Coda"
  [Amazon, or free on Springer Link through CMU or Pitt]