Week 3: Data Visualisation

LSE MY472: Data for Data Scientists
https://lse-my472.github.io/

Autumn Term 2024

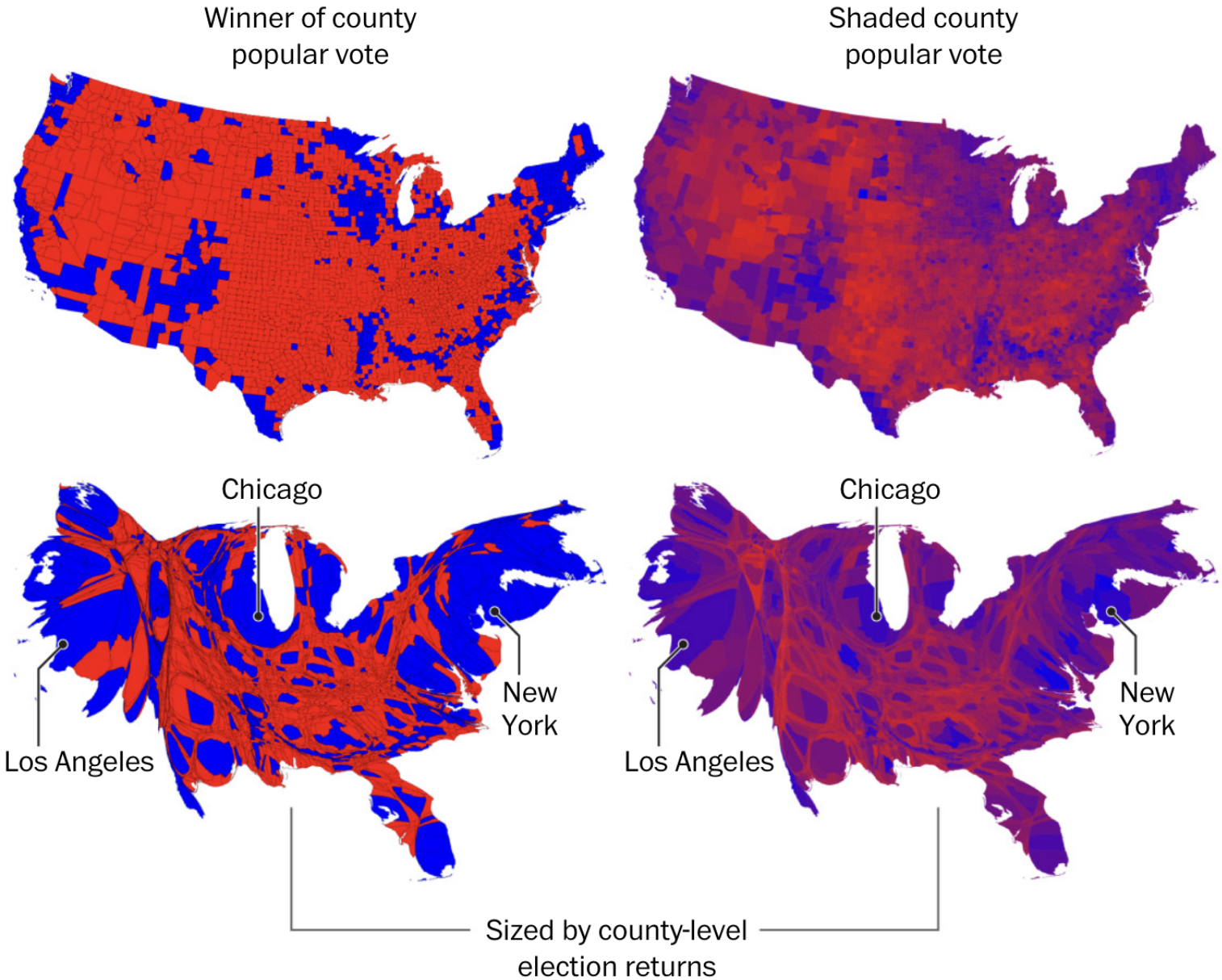
Ryan Hübert

Slides last updated: 16 October 2024

The importance of visualisation

Source: https://x.com/realDonaldTrump/status/1178989254309011456

The importance of visualisation



Source: https://www.washingtonpost.com/graphics/politics/2016-election/how-election-maps-lie/

Plan for today

- Summarising (tabular) data
- Some principles of data visualisation
- Grammar of graphics and `ggplot`
- Coding

Summarising (tabular) data

Reducing complexity to enable *learning* from data

- A tabular dataset is a complex object—lots of observations (rows), lots of information about those observations (columns)
- The typical human cannot look at a raw tabular dataset and draw meaning from it
- Point of data analysis: reduce complexity, enable learning
- To learn, you need to **summarise**, e.g.:
  - Calculate means/medians/counts/etc. of each variable
  - Calculate correlations between multiple variables
  - Make plots of distributions ("shapes") of variables
- Visual communication of data: **visualisation**
  (aka summarising on steroids)
- Warning: *lots* of discretion, need to do this well (LOL!)

What do we *learn* from this?

```
> print(ip_and_unemployment)
# A tibble: 223 × 4
   country date       series       value
   <chr>   <chr>      <chr>        <dbl>
 1 france  01.01.2019 ip           0.973
 2 france  01.01.2019 unemployment 8.7
 3 france  01.02.2019 ip          -0.496
 4 france  01.02.2019 unemployment 8.7
 5 france  01.03.2019 ip          -0.633
 6 france  01.03.2019 unemployment 8.6
 7 france  01.04.2019 ip           0.521
 8 france  01.04.2019 unemployment 8.5
 9 france  01.05.2019 ip           1.64
10 france  01.05.2019 unemployment 8.5
```

Summarising to learn

What was the highest and average unemployment in each country during pandemic onset? (Why is average less useful?)

```
datafr %>%
  filter(series == "unemployment") %>%
  group_by(country) %>%
  summarise(data_beg = min(dmy(date)), data_end = max(dmy(date)),
            max_ue = max(value), average_ue = mean(value)) %>%
  arrange(desc(max_ue))
```

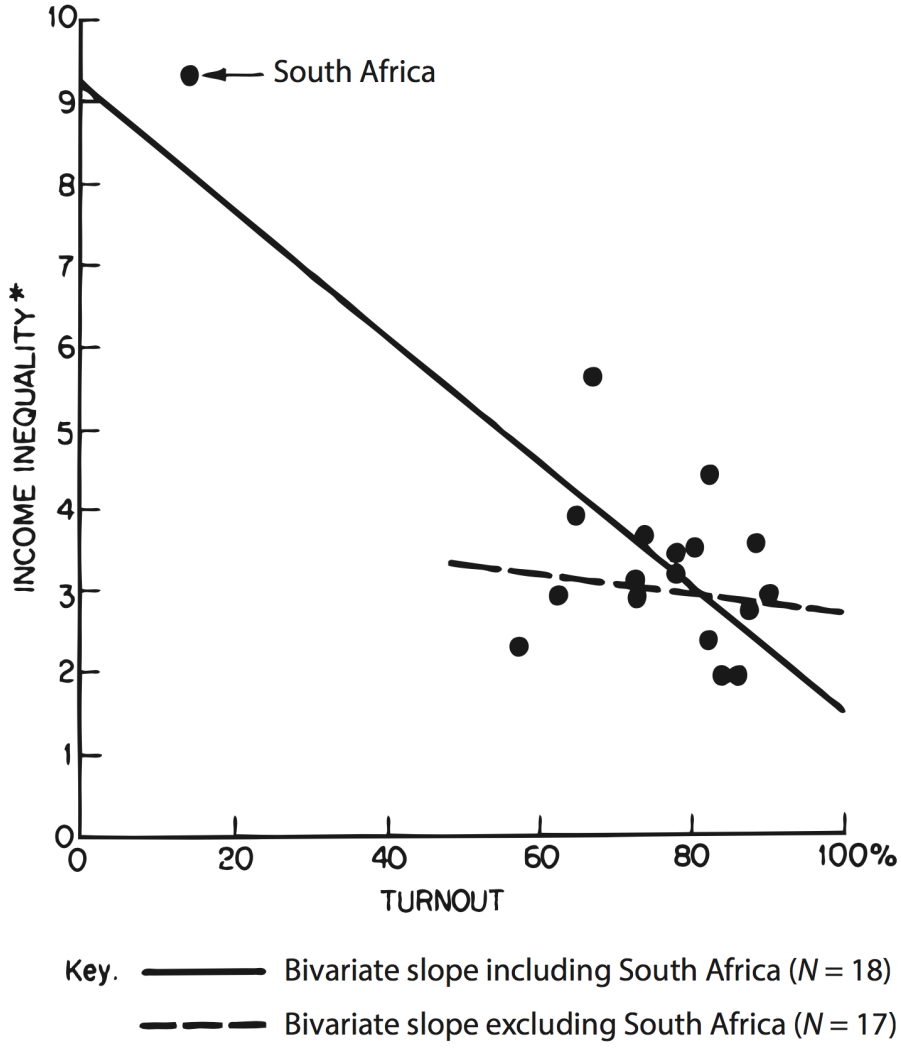
```
# A tibble: 6 × 5
  country data_beg   data_end   max_ue average_ue
  <chr>   <date>     <date>      <dbl>      <dbl>
1 spain   2019-01-01 2020-07-01   15.8       14.4
2 us      2019-01-01 2020-08-01   14.7        5.66
3 italy   2019-01-01 2020-07-01   10.4        9.53
4 france  2019-01-01 2020-07-01    8.7        8.05
5 germany 2019-01-01 2020-07-01    4.4        3.44
6 uk      2019-01-01 2020-05-01    3.9        3.78
```

Some principles of data visualisation

Principles by Edward Tufte

- Show the data
- Avoid distorting what the data have to say
- Allow viewer to compare
- Serve a clear purpose: description, exploration, tabulation or decoration
- Be closely integrated with the statistical and verbal descriptions of the dataset
- Graphics can reveal data (e.g. Anscombe Quartet, which you can replicate with `01-anscombe.Rmd`)

Why you should look at data, an example (from Healy 2019)



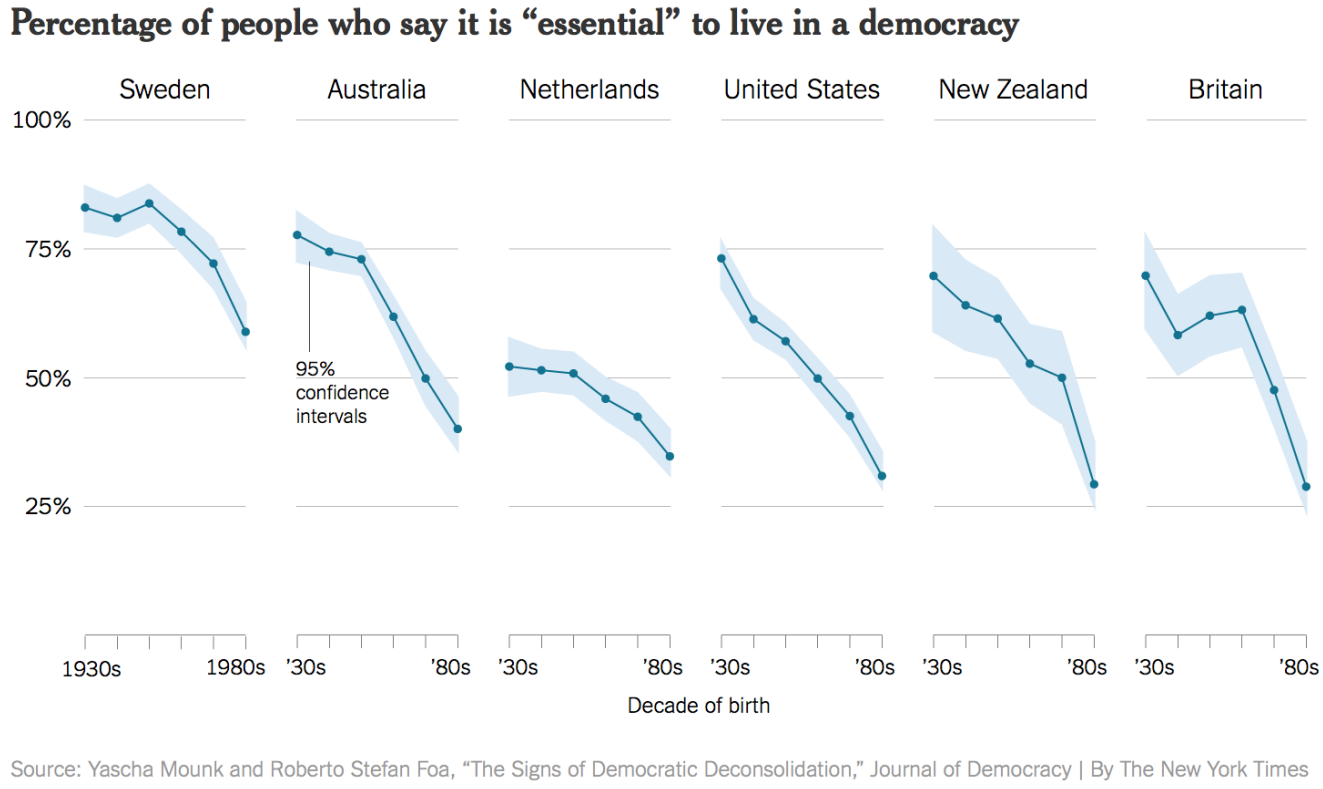
From Jackman, R. M. (1980). "The impact of outliers on income inequality." *American Sociological Review* 45, 344–347.

What makes bad figures bad? (from Healy 2019)

1. Bad taste (e.g., too much "stuff")

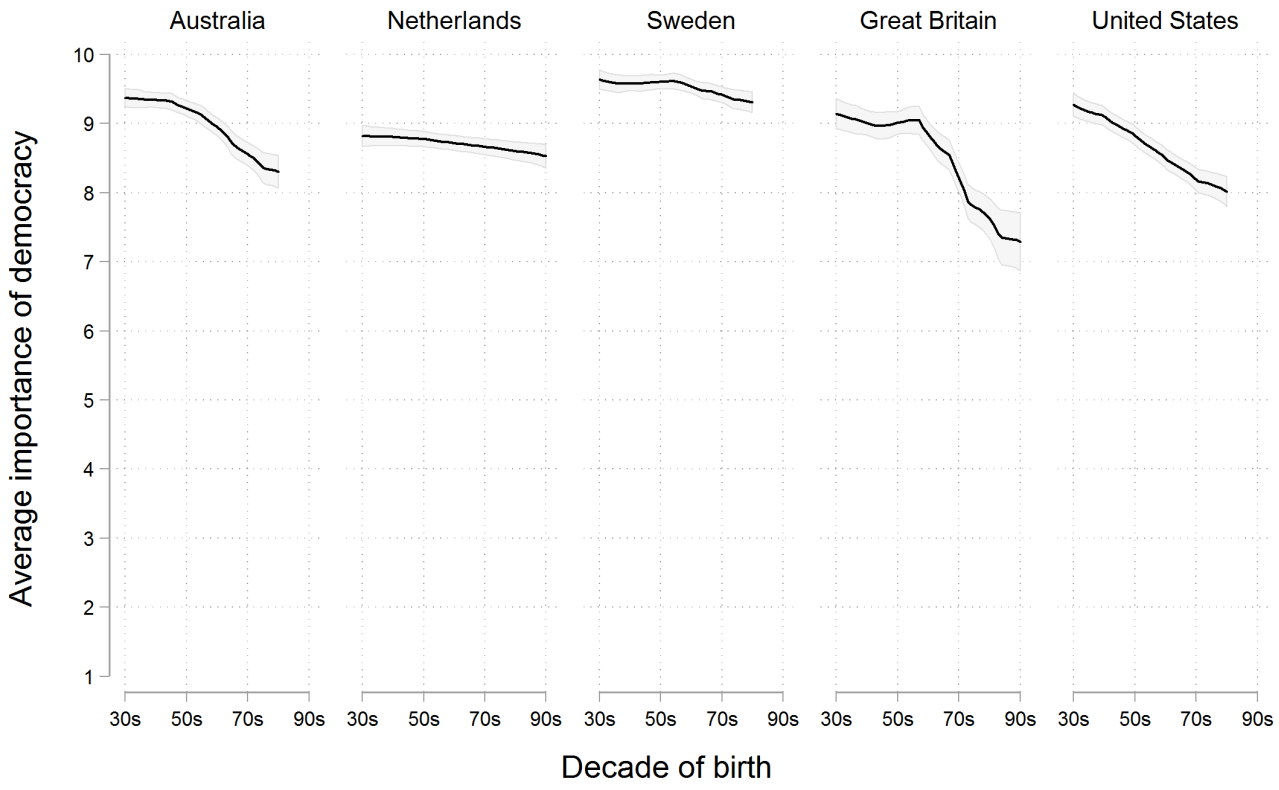
What makes bad figures bad? (from Healy 2019)

2. Bad data (e.g., cherry-picking, misleading)



What makes bad figures bad? (from Healy 2019)

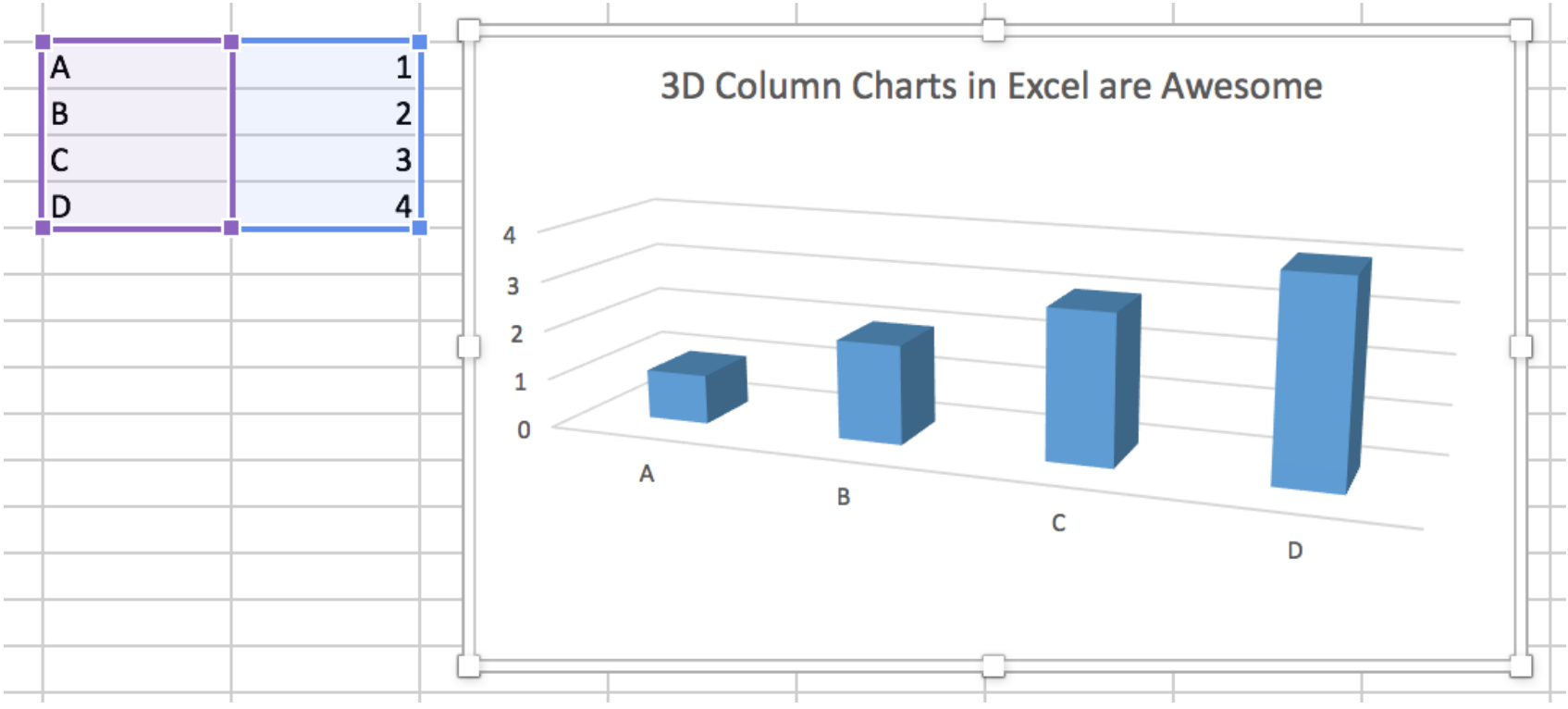
2. Bad data (e.g., cherry-picking, misleading)



Graph by Erik Voeten, based on WVS 5

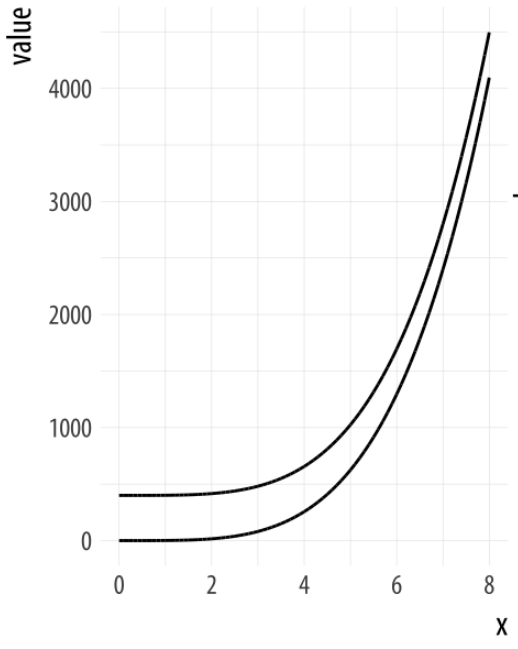
What makes bad figures bad? (from Healy 2019)

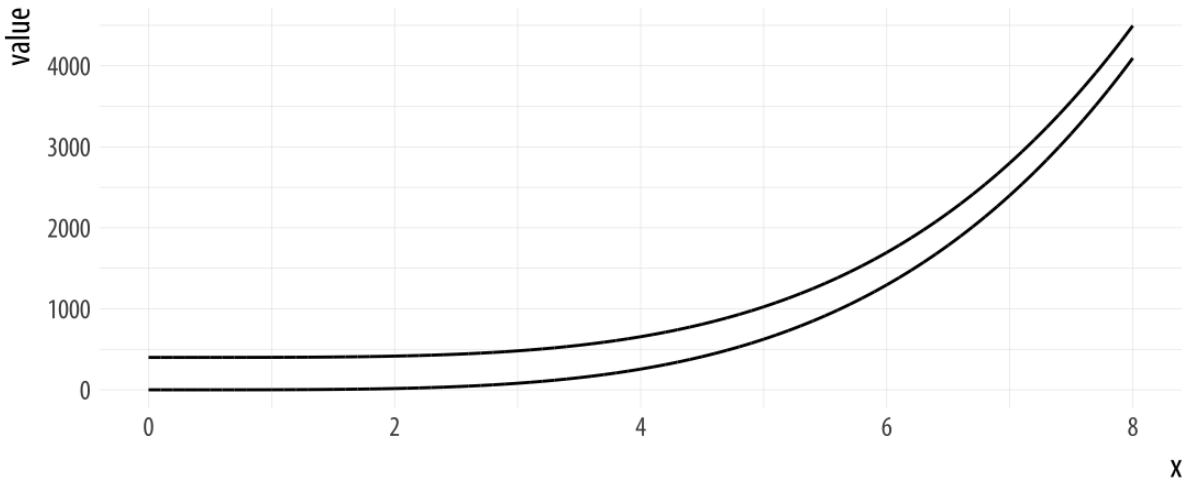
3. Bad perception



What makes bad figures bad? (from Healy 2019)

3. Bad perception





Some general guidelines

- Maximize data-to-ink ratio
- Avoid misleading decisions
  - Y axis starts at 0
  - Comparison of areas is hard
  - Use comparable units
  - Erase chart junk
- Use text to inform and contextualise. Add annotations
- Appropriate use of scales (x/y axes, color, size, shape. . .)
- Use small multiples to facilitate comparisons
- Always cite sources
- Consider accessibility and different use-cases, e.g., sizing, colour-blind palettes, web vs. print (https://colorbrewer2.org/)

Grammar of graphics and `ggplot`

The "grammar of graphics"

- Wilkinson (2005): (statistical) graphics have a "grammar"
  - That is: a set of mathematical and aesthetic rules for creating visual representations from data
- The big (somewhat subtle) idea: data visualisation isn't limited to a constrained set of pre-defined and formulaic "charts"
  - The grammar allows us to innovate and create new kinds of visuals
- `ggplot2`: Hadley Wickham's "layered" version of Wilkinson's grammar of graphics designed for use in R
  - Similar implementation in `plotnine` for Python

The "grammar" of ggplot2

ggplot2 creates visuals from data using **layers**

- ➜ A visual can have more than one layer
- ➜ Intuitively: creating a visual = stacking layers

Each layer contains:

- ➜ `data`: data to visualise (in `tidy` format)
- ➜ `mapping`: links variables in data to visual properties
- ➜ `stat`: statistical transformations of data
- ➜ `geom`: controls the *type* of plotting object (line, point, etc)
- ➜ `position`: adjust overlapping objects

The "grammar" of ggplot2

Layers are the most important component of the grammar, but there are four others major components

- **scales**: translation between variable ranges and graphical properties, e.g. linking values to colours/shapes
- **coordinates**: Coordinate system that e.g. provides axes and gridlines
- **facets**: Breaking up the data into subsets e.g. to be displayed independently on a grid
- **theme**: Parts that do not follow from the data: Background colours, fonts, etc.

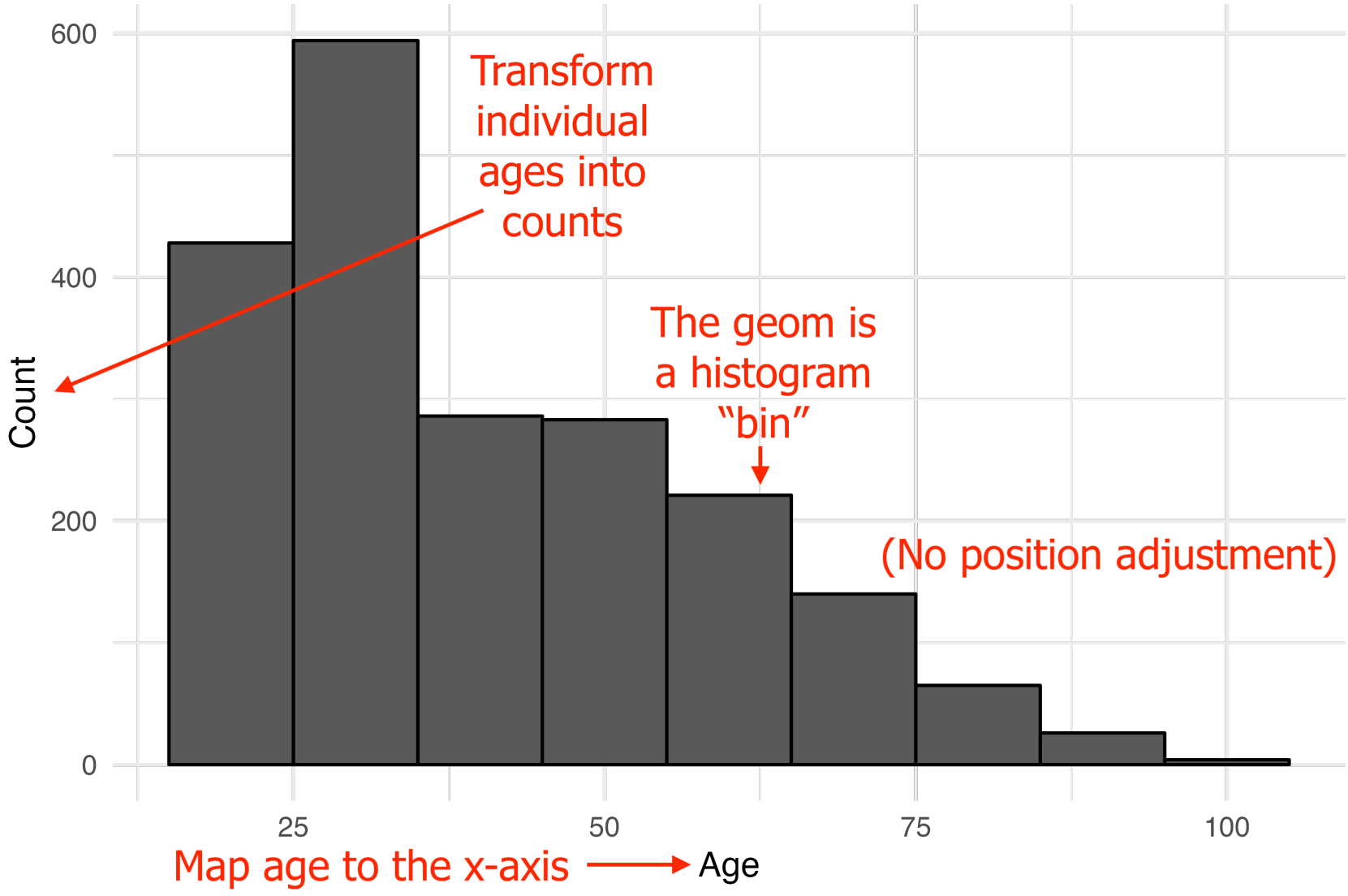
Example: distribution of age

Consider subject-level information about age:

```
#>   age
#> 1  20
#> 2  56
#> 3  40
#> 4  21
#> 5  38
#> 6  39
#> ...
```

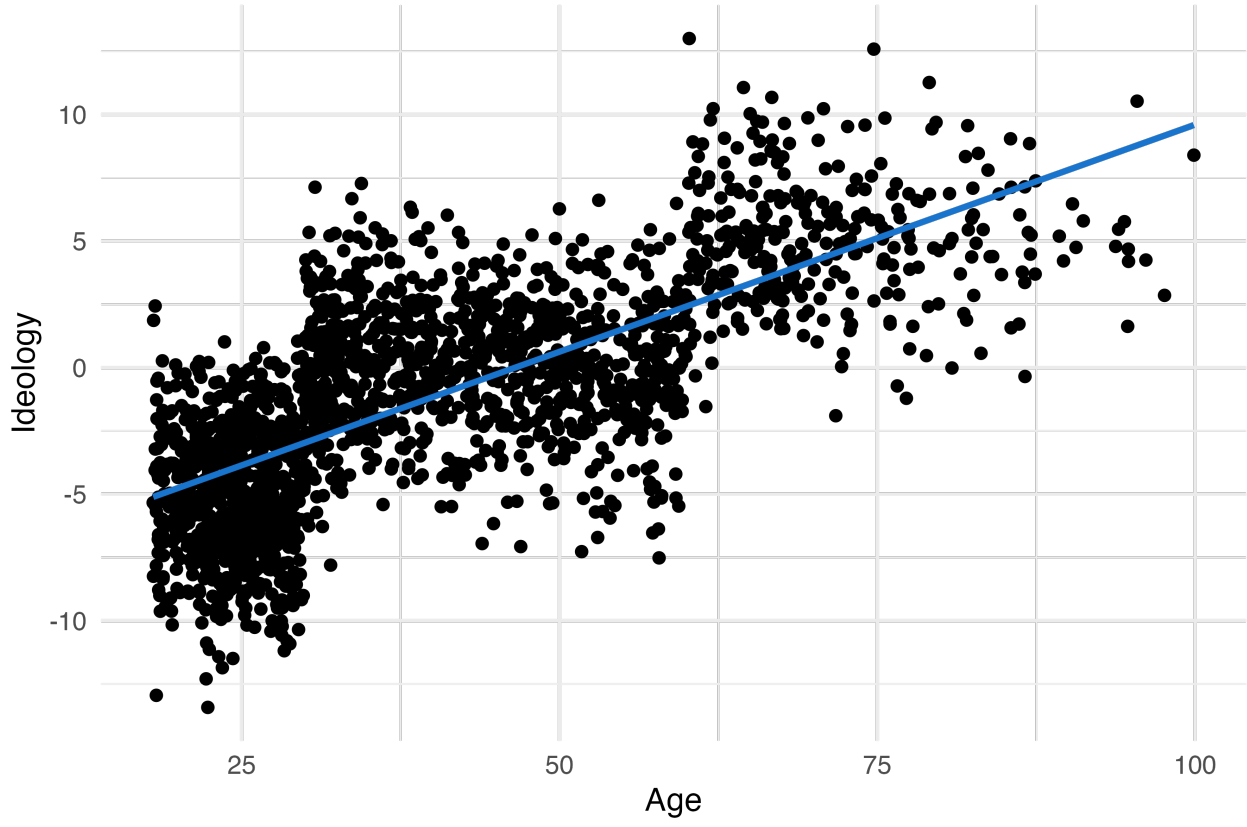
How could we summarise this information visually?

Example: distribution of age



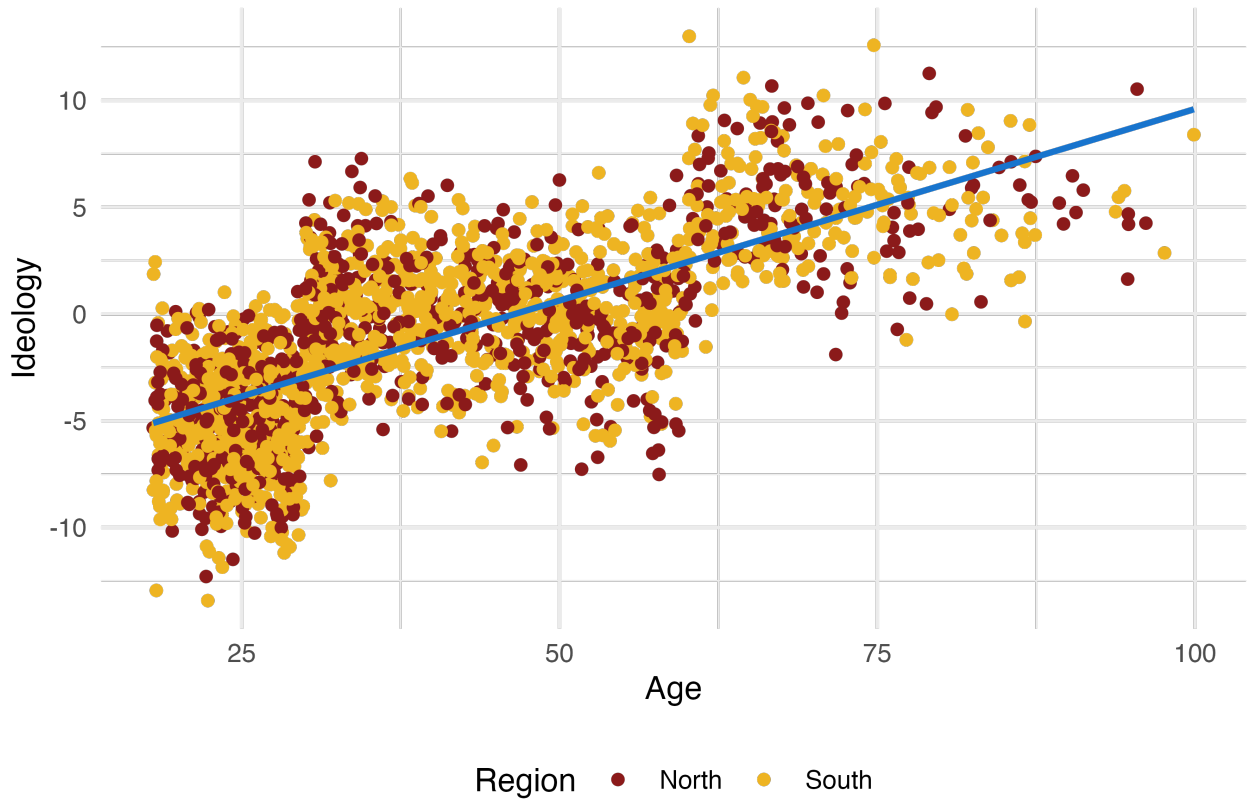
Multiple layers

- Since layers are contained, we can overlay multiple layers
- This strategy is very common
- Example: A scatterplot + line of best fit

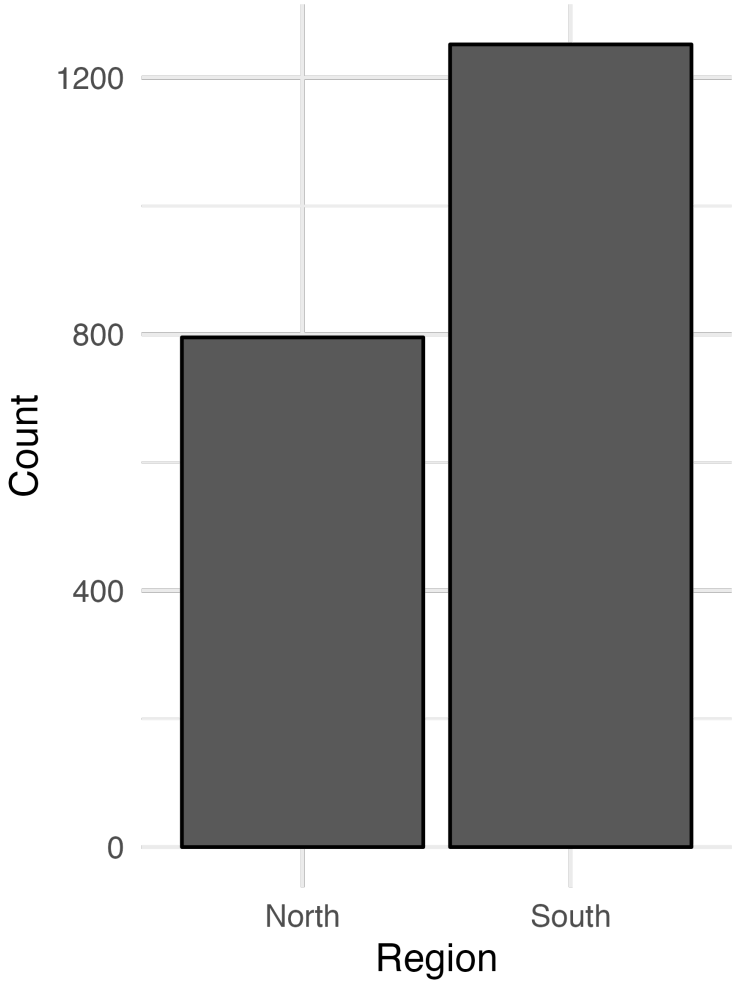


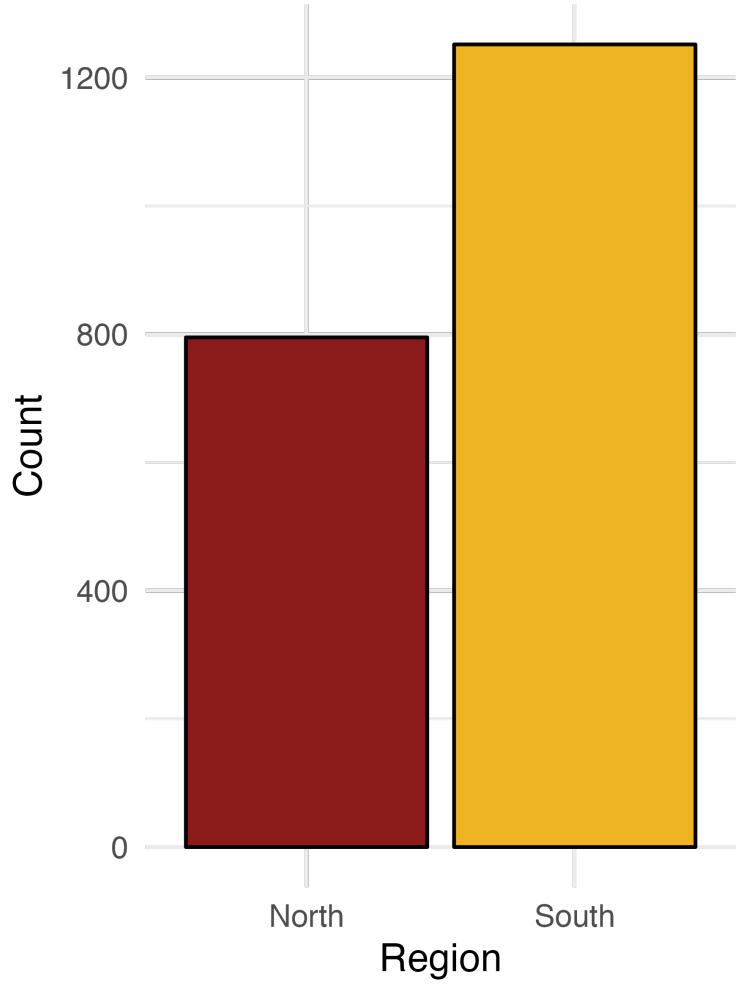
Scales

- Scales "translate" data ranges to property ranges
  - Map continuous numeric data to a color spectrum
  - Translate categorical data to different shapes
  - Map the size of a geom to some value (e.g. frequency)
  - Etc.
- Scales modify the geom object(s)



Which do you prefer?





Redundant scales

In the previous slide:

- ➜ Colouring the bars by region adds **no** new information
- ➜ We call this **redundancy**
  - ➜ When two (or more) scales translate the *same* variable to different aesthetics
- ➜ Redundancy can overly complicate plots. . .
- ➜ . . . but can also add clarity, improve accessibility

Facets and coordinates

Facets allow you to create **multiple** plots by mapping subsets of your data

- E.g. Plotting separate histograms by respondent's country of origin
- When you facet by a single variable we use a *wrap*
- When we facet by two (or more) variables, we use a *grid*

Coordinate systems "map the position of objects onto the plane of the plot" (Wickham 2010, p.13)

- In almost all cases we use **Cartesian coordinates**
  - Two orthogonal dimension (x, y)
- Alternative systems exist, like polar coordinates:
  - Allow you to draw circular distributions like pie-charts (eww!)

Why ggplot2?

- Consistent, modular, and very flexible
- Sensible defaults for quick exploratory plots
- But also easy to customize and extend
- Excellent online resources
- Pretty (publishable) graphics

Online resources

- Kieran Healy's book on data visualisation in R: https://socviz.co/
- Main documentation page: https://ggplot2.tidyverse.org/
- Book by Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen: https://ggplot2-book.org/
- R Graph gallery for `ggplot2` https://www.r-graph-gallery.com/ggplot2-package.html
- Two recent video workshops by Thomas Lin Pedersen, video 1, video 2, and the repo with associated exercises
- StackOverflow, tag: `ggplot2` https://stackoverflow.com/questions/tagged/ggplot2

Coding

Coding

- ➜ `02-ggplot-walkthrough.Rmd`

For your reference:

- ➜ `03a-ggplot2-basics.Rmd`
- ➜ `03b-scales-axes-legends.Rmd`