MY472 – Data for Data Scientists
Week 3: Data Visualisation

Thomas Robinson

AT 2023

https://lse-my472.github.io/

Course outline

1. Introduction
2. Tabular data
3. Data visualisation
4. Textual data
5. HTML, CSS, and scraping static websites
6. (Reading week)
7. XML, RSS, and scraping non-static website
8. Working with APIs
9. Creating and managing databases
10. Interacting with online databases
11. Cloud computing

Outline

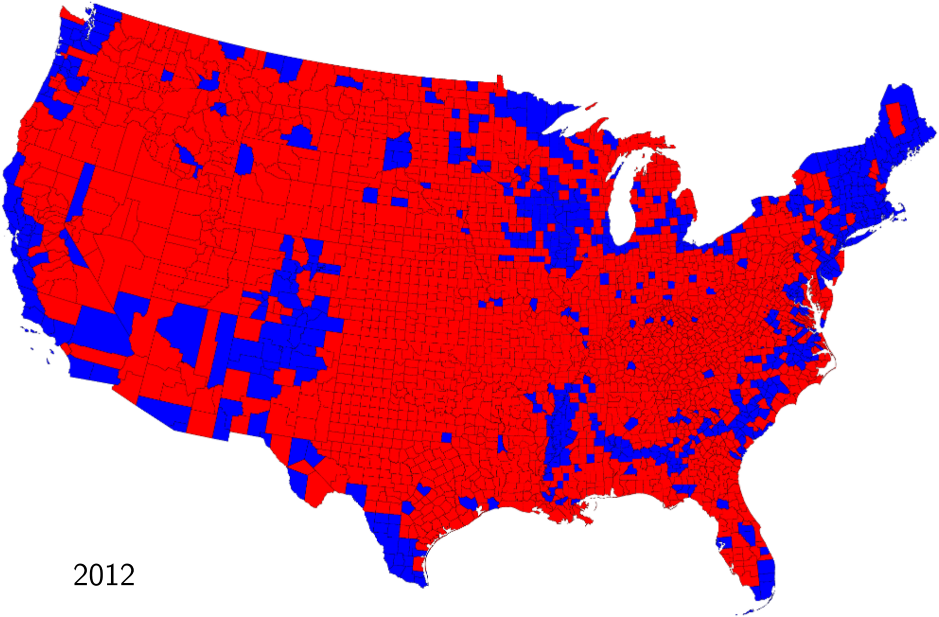
1. Introduction
2. Some principles of data visualisation
3. Grammar of graphics and **ggplot**
4. Coding

Outline

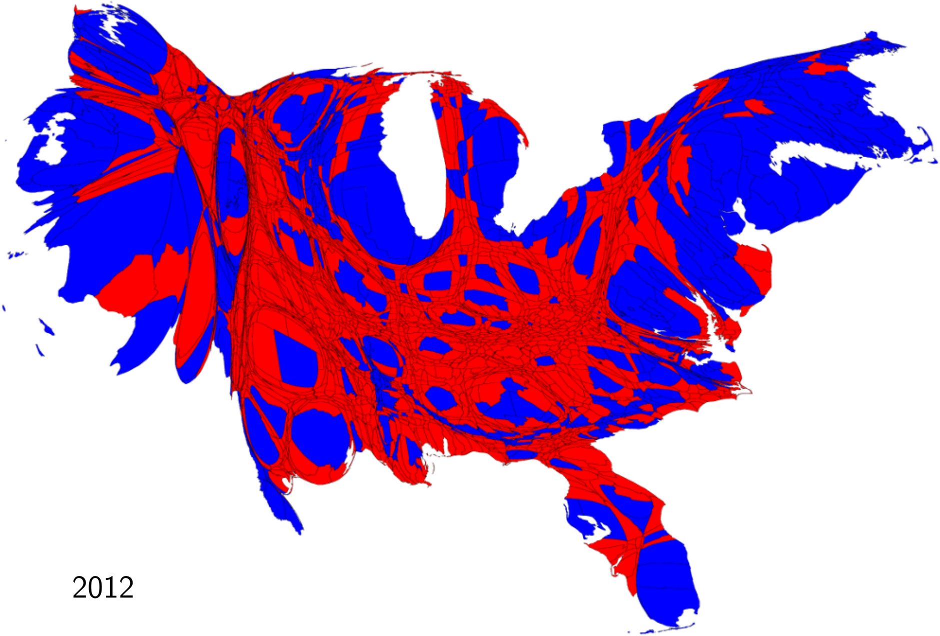
1. Introduction
2. Some principles of data visualisation
3. Grammar of graphics and **ggplot**
4. Coding

Why visualisation can be helpful: Anscombe examples

`01-anscombe.Rmd`



Source: Mark Newman (Michigan)



2012

Source: Mark Newman (Michigan)

Outline

1. Introduction
2. Some principles of data visualisation
3. Grammar of graphics and **ggplot**
4. Coding

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)

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

Outline

1. Introduction
2. Some principles of data visualisation
3. Grammar of graphics and **ggplot**
4. Coding

A grammar for visualization?

- Linguistic grammar provides structure to words that help us convey more complex meaning (information)
- Leland Wilkinson (1999) argued graphics also have a deep structure–a “grammar”–that:
  - “Take us beyond a limited set of charts (words) to an almost unlimited world of graphical forms (statements)” (p.1).
- By combining various “aesthetics” we can reliably make meaningful *visual* representations of data

Fast forward a decade:

> **The grammar of graphics.**
>
> *A statistical graph is a mapping from data to aesthetic attributes (color, shape, size) of geometric objects (points, lines, bars). The plot may also contain statistical transformations of the data and is drawn on a specific coordinate system. Faceting can be used to generate the same plot for different subsets of the data. It is the combination of these independent components that make up a graphic.*
>
> **Hadley Wickham**, *ggplot2, page 3*

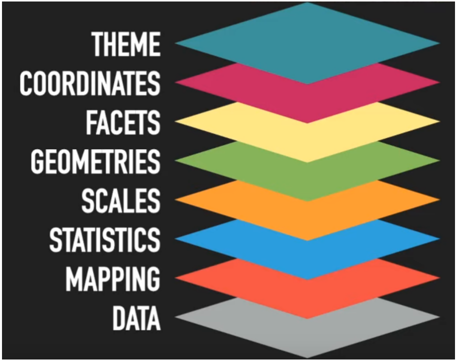
- Layered version of Wilkinson's framework introduced as R package **ggplot2**
- Similar implementation in **plotnine** for Python

Data visualisation with ggplot2

Why **ggplot2**?

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

The grammar



Source: Thomas Lin Pedersen [link]

Grammar

data Data to visualise, for ggplot2 in a ‘tidy‘ format

(aesthetic) mapping Linking variables in the data to components of the graphic

stats Statistical transformations of the data, e.g. binning or averaging

scales Translation between variable ranges and graphical properties, e.g. linking values to colours/shapes

geom Geometric objects that are drawn to represent the data: bars, lines, points, etc. (plots can have multiple geometries)

facets Breaking up the data into subsets e.g. to be displayed independently on a grid

coordinates Coordinate system that e.g. provides axes and gridlines

theme Parts that do not follow from the data: Background colours, fonts, etc.

Layer = Data + Mapping + Statistics + Geom + Position

A layer contains (some) visual information we see on the graphic:

- Without **data**, we have an empty plot!
- **Mapping** links variables in the data to visual properties
- **Statistics** allows us to transform our input data
- A **geom** controls the type of plotting object
- A **position adjustment** allows us to, .e.g., prevent perfectly overlapping points

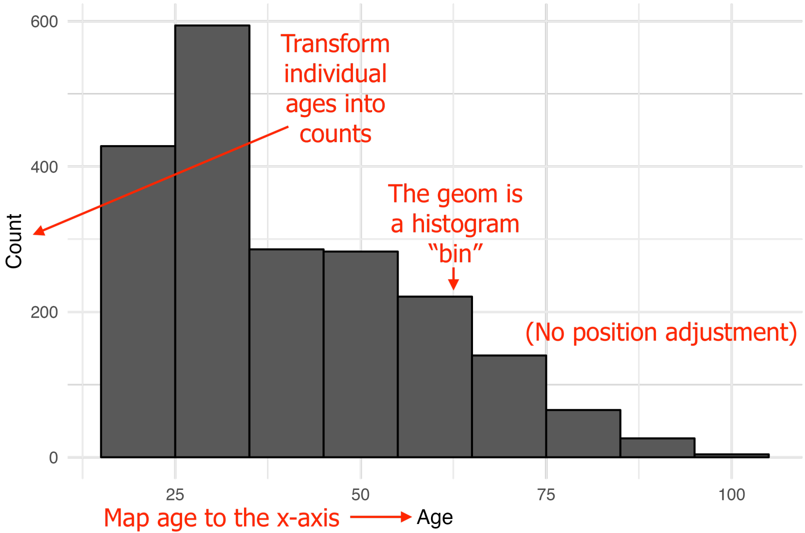
Example: distribution of age

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



Transform individual ages into counts

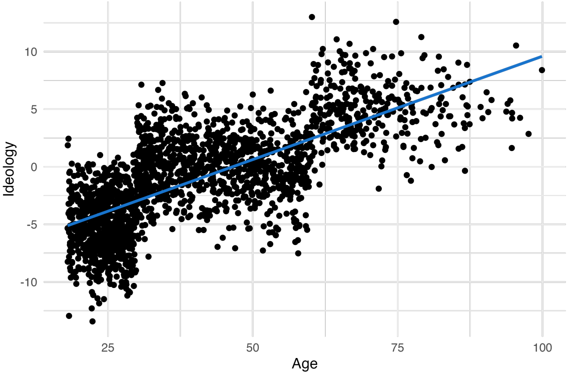
The geom is a histogram "bin"

(No position adjustment)

Map age to the x-axis

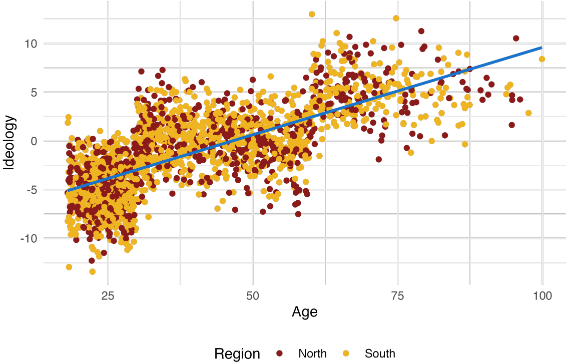
Layering

- Because layers are contained, we can overlay multiple layers at once
- This strategy is very common
  - A scatterplot + line of best fit
  - Coefficient estimates (points) + confidence intervals (errorbars)

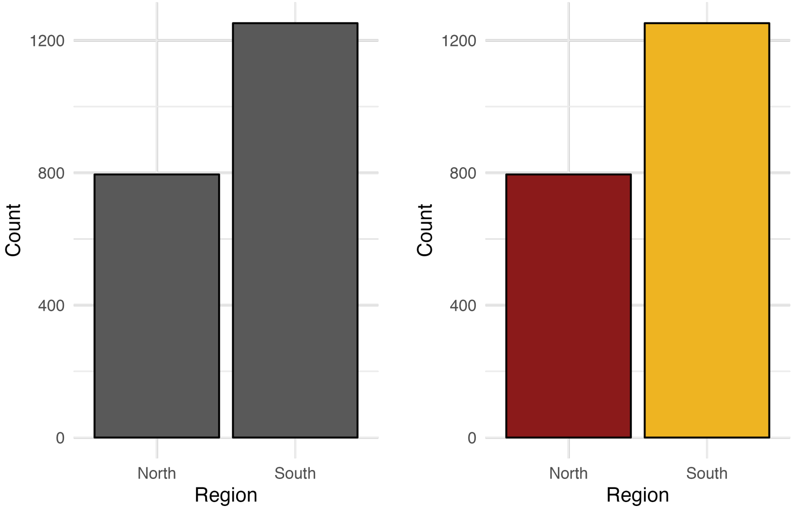


Scales

- 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

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 should we abide by the grammar of graphics?

- The system is very flexible
- Allows us to describe how to go from data to visuals
- Reduces the complexity and verbosity of graph construction
- Forces you to think about *what* information you want to convey

Online resources

- 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

Outline

1. Introduction
2. Some principles of data visualisation
3. ggplot2
4. Coding

Coding

`ggplot-walkthrough.Rmd`

For your reference:

`03a-ggplot2-basics.Rmd`

`03b-scales-axes-legends.Rmd`