Introduction to Data Science

Session 7: Visualization

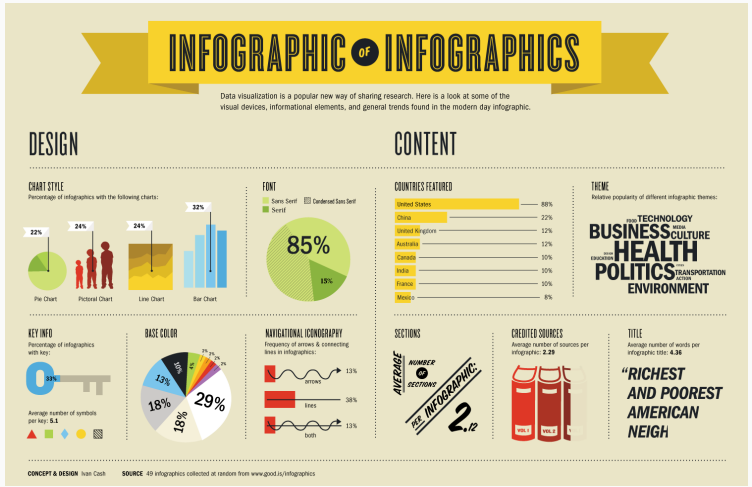
Simon Munzert
Hertie School | GRAD-C11/E1339

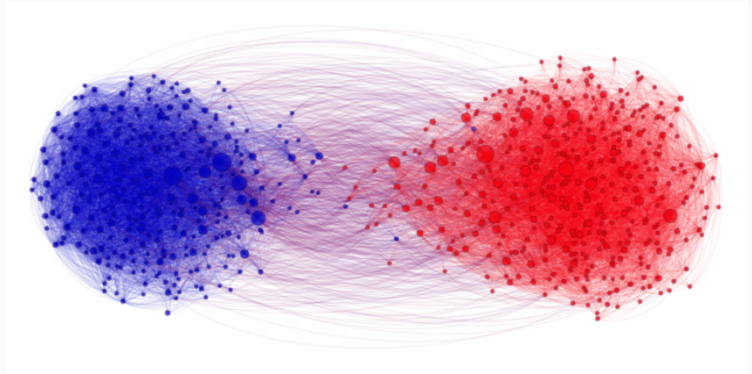
Table of contents

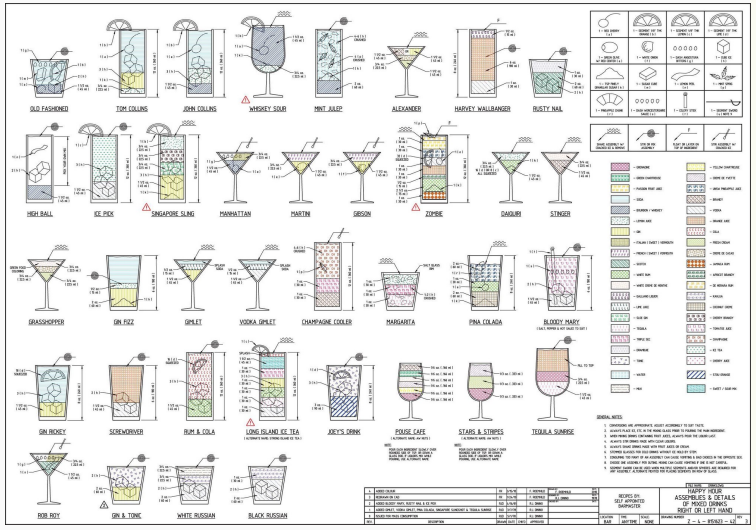
[1] Much of these sections draw on materials from Claus Wilke's excellent book *Fundamentals of Data Visualization*.

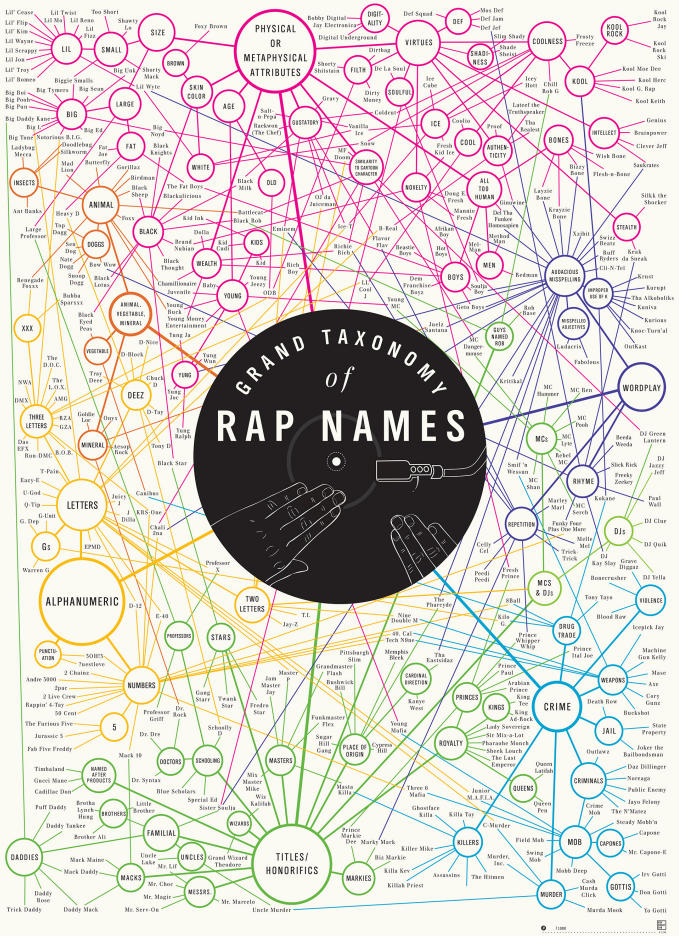
Why data visualization?

You came for this...

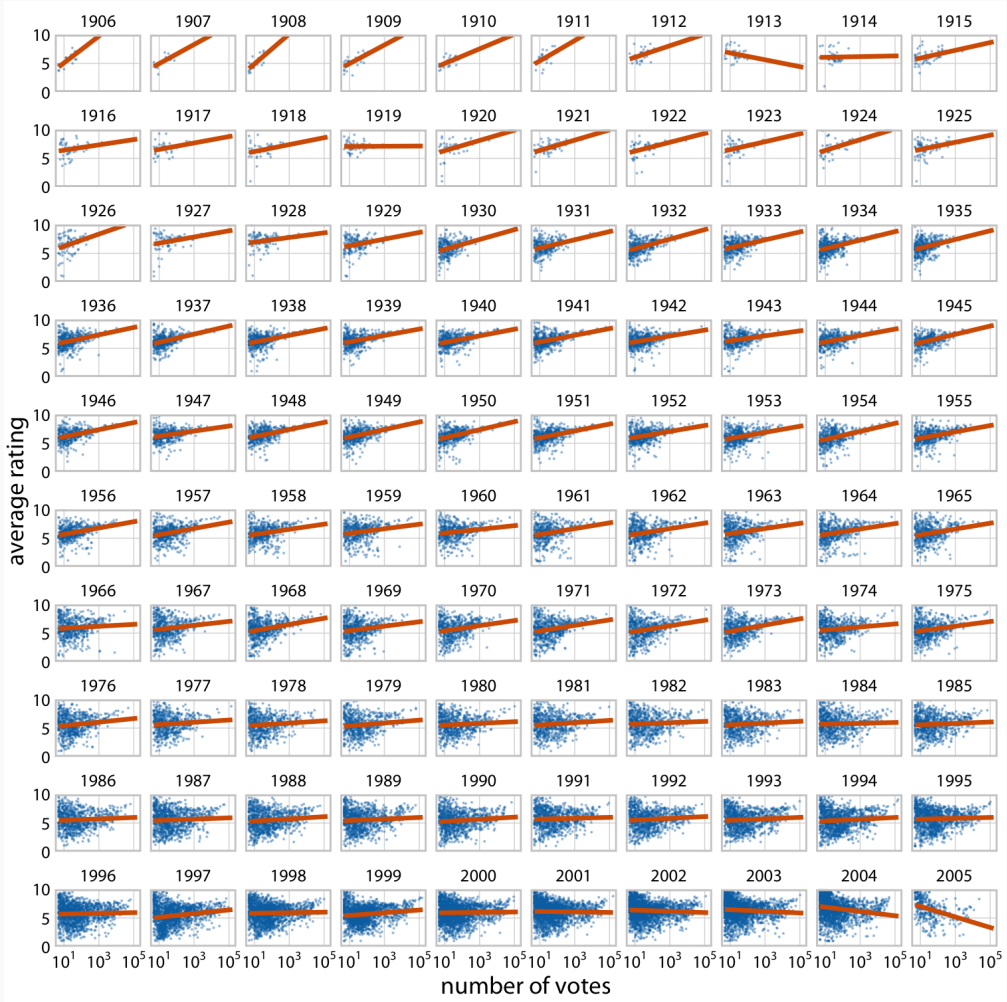


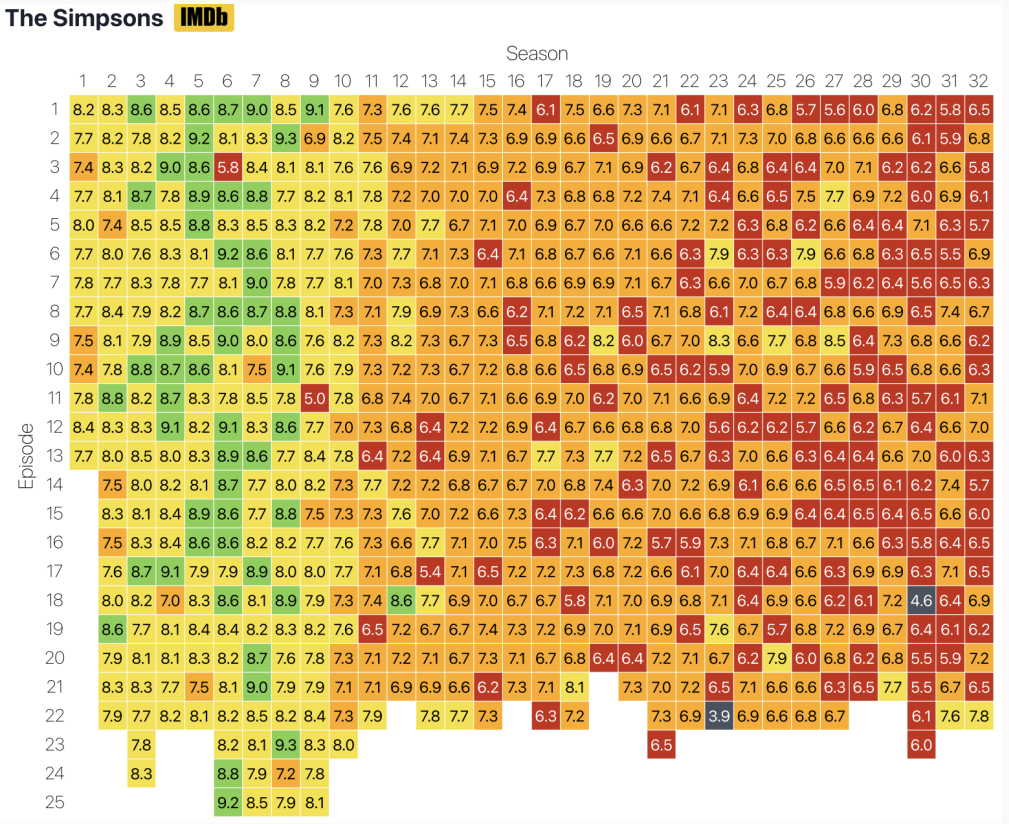






… but you'll be getting this

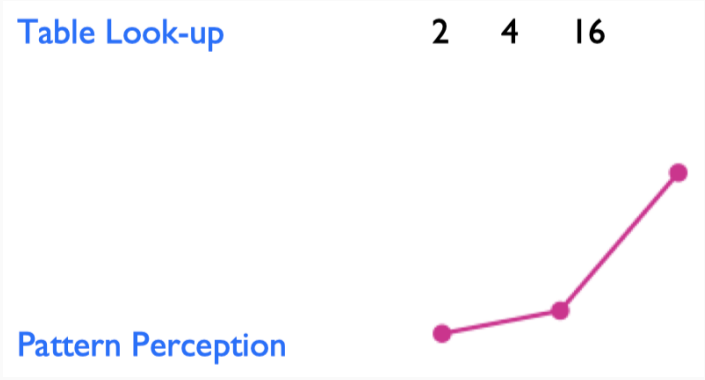




Why data visualization?

A new method for the DS toolbox

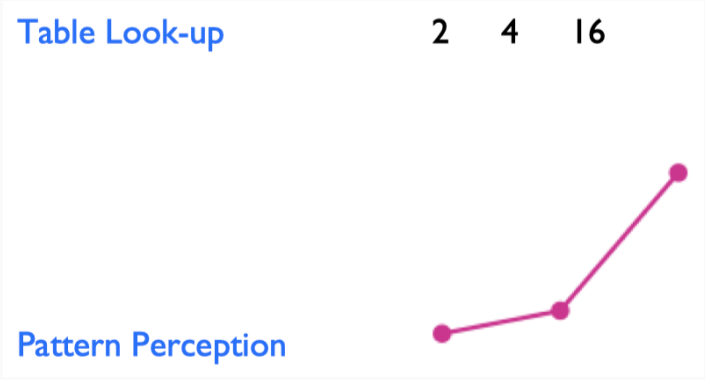
- Data visualization is a method for making sense (and not just pictures) of data.
- Note that this is more than data visualization in the narrow sense, i.e. the act of encoding quantitative information in visual objects.
- Data scientists are mostly interested in patterns, not individual and exact values.
- Two ways to make sense of quantitative information:



Why data visualization?

A new method for the DS toolbox

- Data visualization is a method for making sense (and not just pictures) of data.
- Note that this is more than data visualization in the narrow sense, i.e. the act of encoding quantitative information in visual objects.
- Data scientists are mostly interested in patterns, not individual and exact values.
- Two ways to make sense of quantitative information:

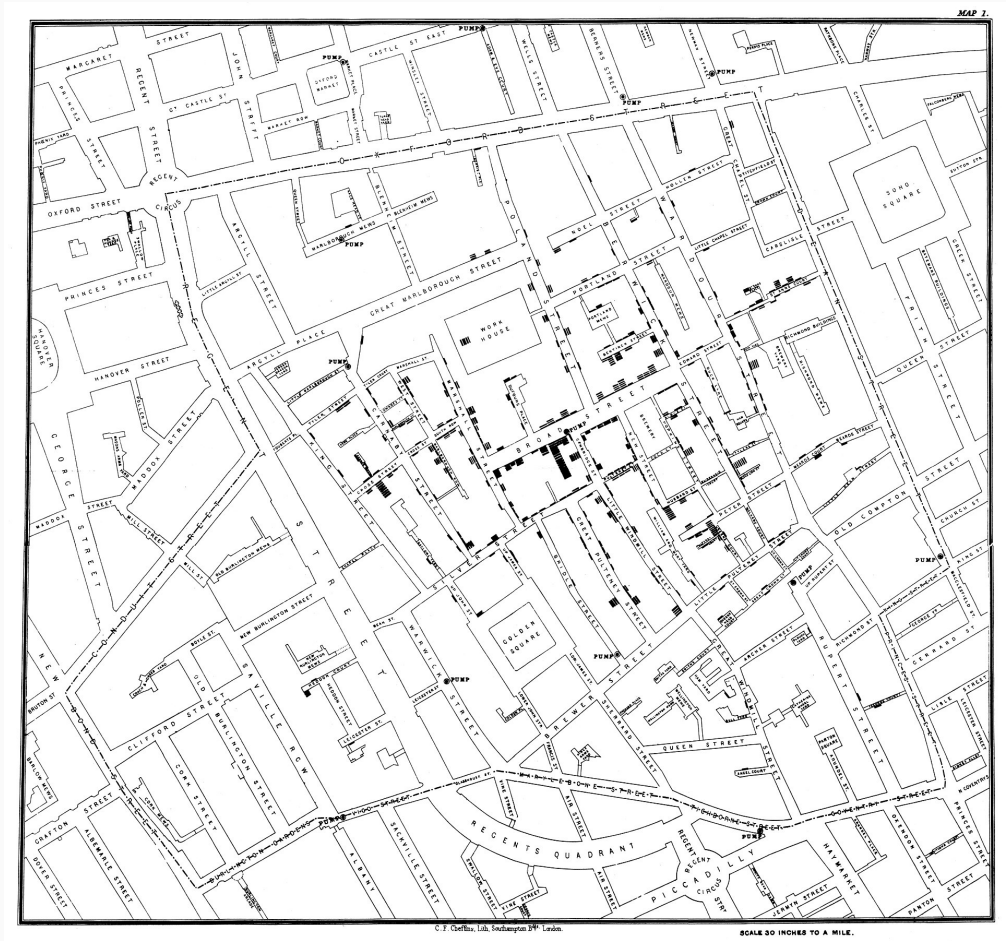


The case for visualization

- Visualization **provides useful summaries** for large, complicated data sets – in fact, the utility of visualization increases with data size.
- Visualization **lets you see things** that would otherwise be invisible, in particular relationships among data (patterns, trends, exceptions).
- Visualization comes with **little or no assumptions** about the nature of the data.
- Visualization facilitates interaction between researcher and data – **it's a hypothesis generating device**.

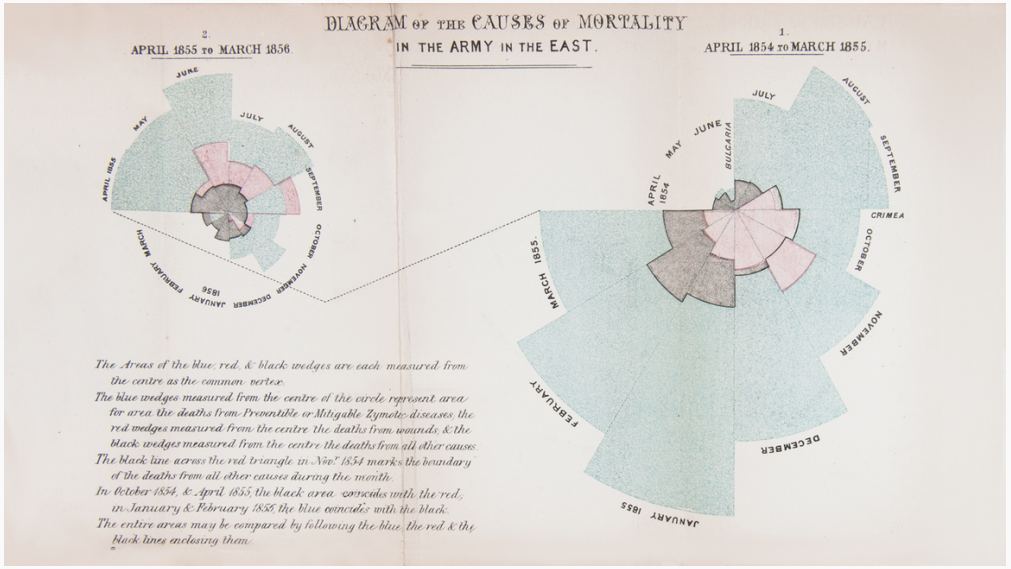
"The critical question is how best to transform the data into something that people can understand for optimal decision making." Colin Ware, 2013

1854 Broad Street cholera outbreak



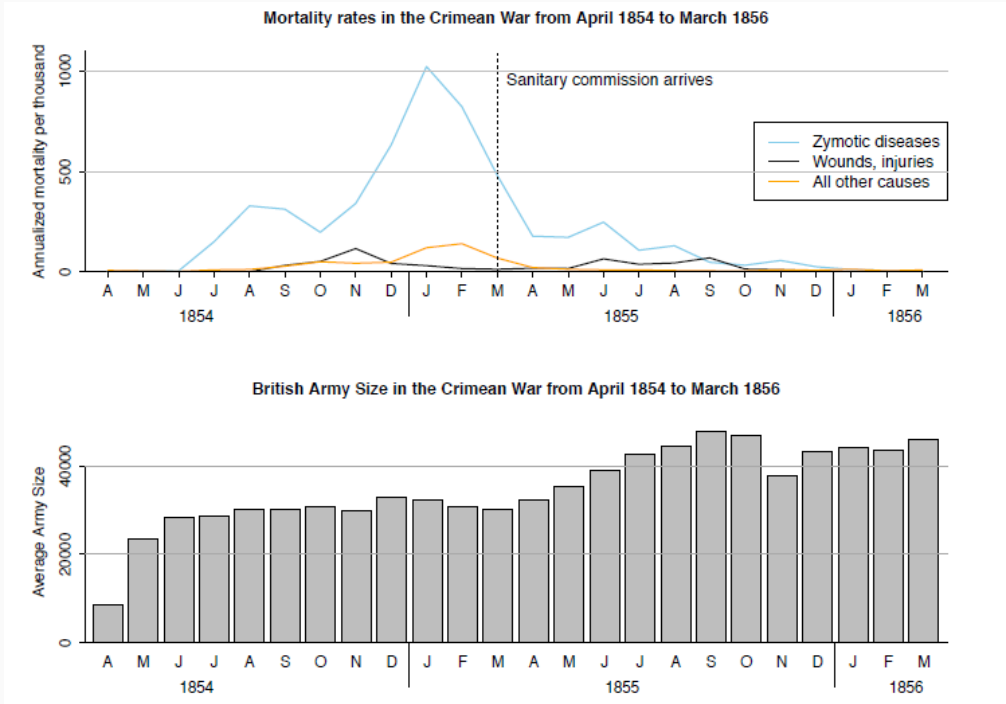
- One of the most famous data viusalizations of all times: John Snow's cholera case map.
- The Broad Street cholera outbreak in in Soho, London in 1854 was studied by physician John Snow to study its causes (rival hypotheses: germ-contaminated water vs. airborne transmission).
- The germ theory was not established at this point but the map helped highlight how cases clustered around a contaminated pump (which was by far not the only source of contaminated water though).
- Fun fact: this is what doing good data viz gives you:

Nightingale's rose



- One of the potentially most influential graphs is by Florence Nightingale, statistician and founder of modern nursing.
- She pioneered in using graphs to communicate data and to ease drawing conclusions.
- Nightingale's Rose, a polar area diagram, illustrates seasonal sources of soldier mortality in the field hospital Nightingale managed during the Crimean War and highlights that epidemic disease are responsible for more deaths than battlefield wounds.
- Nightingale's work revolutionized points hygiene and other practices in hospitals, ultimately saving millions of lives.

Nightingale's rose (cont.)

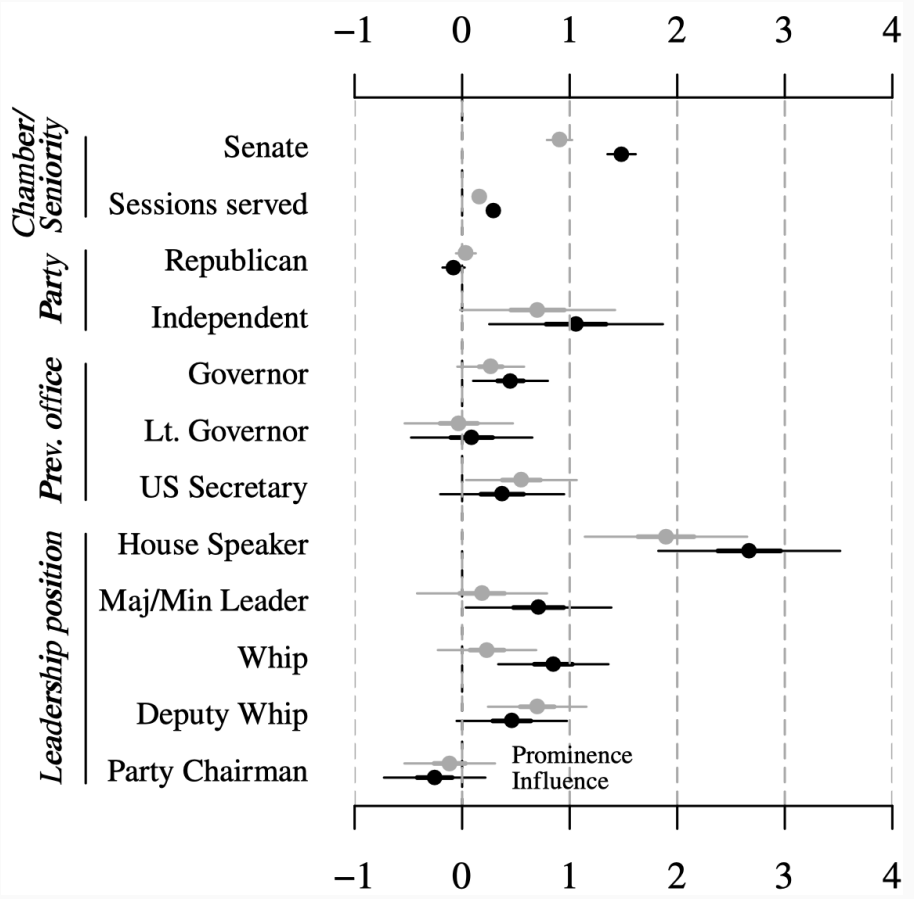


- One of the potentially most influential graphs is by Florence Nightingale, statistician and founder of modern nursing.
- She pioneered in using graphs to communicate data and to ease drawing conclusions.
- Nightingale's Rose, a polar area diagram, illustrates seasonal sources of soldier mortality in the field hospital Nightingale managed during the Crimean War and highlights that epidemic disease are responsible for more deaths than battlefield wounds.
- Nightingale's work revolutionized points hygiene and other practices in hospitals, ultimately saving millions of lives.
- Gelman and Unwin (2012) present an alternative presentation of the same data.
- More historical visualizations here

Graphs vs. tables I: model estimates

	Prominence	Influence
Senate	0.906***	1.483***
	(0.060)	(0.067)
Sessions served	0.163***	0.292***
	(0.016)	(0.017)
Party (Independent)	0.701*	1.059**
	(0.368)	(0.412)
Party (Republican)	0.035	−0.080
	(0.047)	(0.052)
Office: Governor	0.266*	0.450**
	(0.158)	(0.177)
Office: Lt. Governor	−0.031	0.089
	(0.257)	(0.288)
Office: US Secretary	0.551**	0.372
	(0.262)	(0.294)
Position: House Speaker	1.896***	2.670***
	(0.385)	(0.431)
Position: Majority/Minority Leader	0.185	0.711**
	(0.308)	(0.345)
Position: Whip	0.231	0.848***
	(0.233)	(0.261)
Position: Deputy Whip	0.698***	0.462*
	(0.234)	(0.262)
Position: Party Chairman	−0.115	−0.255
	(0.215)	(0.241)
(Intercept)	1.648***	1.527***
	(0.050)	(0.057)
N	492	492
R-squared	0.493	0.694
Adj. R-squared	0.481	0.687
Residual Std. Error (df = 479)	0.505	0.565
F Statistic (df = 12; 479)	38.890***	90.715***

***p < .01; **p < .05; *p < .1



Graphs vs. tables II: amounts

Country	Length of Constitution
Bosnia and Herzegovina	5,230
Montenegro	7,074
Andorra	8,740
Macedonia	9,231
Croatia	10,898
Slovenia	11,410
Italy	11,708
Albania	13,747
Spain	17,608
Serbia	19,891
Greece	27,177
Malta	31,820
Portugal	35,181

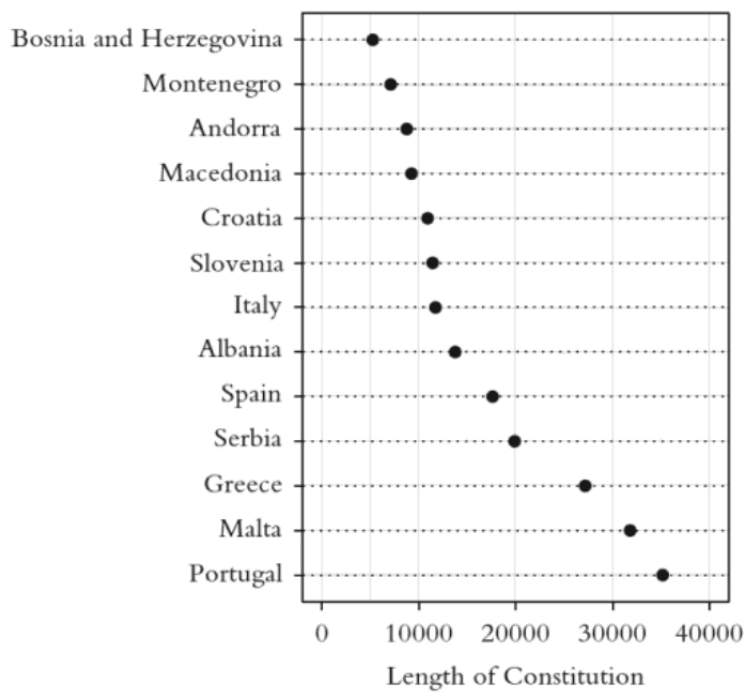


Figure 10.2 Length (in words) of present-day constitutions in countries in Southern Europe. Although it is possible from the table to observe the patterns that jump out in the graph—for example, the large difference between the shortest and longest constitutions—it requires far more (unnecessary) cognitive work.[50]

Graphs vs. tables III: relationships

- The table on the right comprises data sets I through IV, each consisting of eleven (x, y) points.
- Carefully study the table. How do x and y as well as their relationship compare across datasets?

I.		II.		III.		IV.	
y1	*x1*	*y2*	*x2*	*y3*	*x3*	*y4*	*x4*
8.04	10	9.14	10	7.46	10	6.58	8
6.95	8	8.14	8	6.77	8	5.76	8
7.58	13	8.74	13	12.74	13	7.71	8
8.81	9	8.77	9	7.11	9	8.84	8
8.33	11	9.26	11	7.81	11	8.47	8
9.96	14	8.10	14	8.84	14	7.04	8
7.24	6	6.13	6	6.08	6	5.25	8
4.26	4	3.10	4	5.39	4	12.50	19
10.84	12	9.13	12	8.15	12	5.56	8
4.82	7	7.26	7	6.42	7	7.91	8
5.68	5	4.74	5	5.73	5	6.89	8

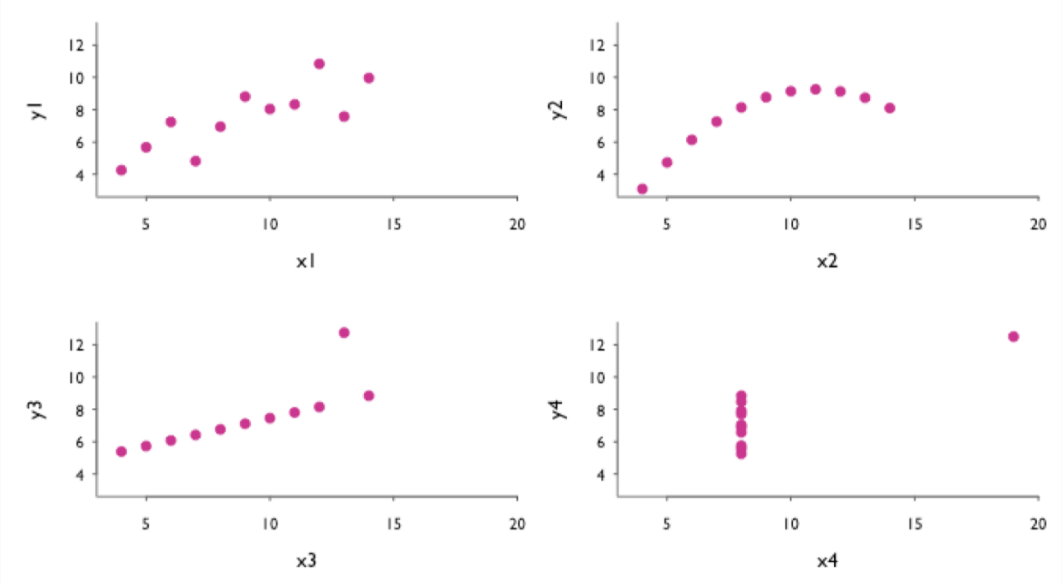
Graphs vs. tables III: relationships (cont.)

- The table on the right comprises data sets I through IV, each consisting of eleven (x, y) points.
- Carefully study the table. How do x and y as well as their relationship compare across datasets?
- It shows that all the data sets have nearly identical simple descriptive statistics in terms of mean, standard deviation, correlation, and linear fit!

	I.		II.		III.		IV.	
	y1	*x1*	*y2*	*x2*	*y3*	*x3*	*y4*	*x4*
	8.04	10	9.14	10	7.46	10	6.58	8
	6.95	8	8.14	8	6.77	8	5.76	8
	7.58	13	8.74	13	12.74	13	7.71	8
	8.81	9	8.77	9	7.11	9	8.84	8
	8.33	11	9.26	11	7.81	11	8.47	8
	9.96	14	8.10	14	8.84	14	7.04	8
	7.24	6	6.13	6	6.08	6	5.25	8
	4.26	4	3.10	4	5.39	4	12.50	19
	10.84	12	9.13	12	8.15	12	5.56	8
	4.82	7	7.26	7	6.42	7	7.91	8
	5.68	5	4.74	5	5.73	5	6.89	8
Mean(*y*)	7.50		7.5		7.50		7.5	
Mean(*x*)	9.0		9.0		9.0		9.0	
SD(*y*)	2.03		2.03		2.03		2.03	
SD(*x*)	3.32		3.32		3.32		3.32	
r(*y*, *x*)	.82		.82		.82		.82	
y = a + b*x*	y = 3 + 0.5*x*		y = 3 + 0.5*x*		y = 3 + 0.5*x*		y = 3 + 0.5*x*	
R^2	.67		.67		.67		.67	

Graphs vs. tables III: relationships (cont.)

- The table on the right comprises data sets I through IV, each consisting of eleven (x, y) points.
- Carefully study the table. How do x and y as well as their relationship compare across datasets?
- It shows that all the data sets have nearly identical simple descriptive statistics in terms of mean, standard deviation, correlation, and linear fit!
- Plotting the data reveals wildly different distributions, countering the impression that "numerical calculations are exact, but graphs are rough" (Anscombe 1973).
- The dataset was constructed by Francis Anscombe and is known as "Anscombe's quartet".
- Graphs 3, Tables 0. Case closed.[1]



[1] That being said, there is a case to be made for tables under certain circumstances. But even tables can (and should) be seen as another form for visualization with clear design principles. To design appealing tables with R, check out the `gt` package.

Data visualization as a method

Different goals, different looks

Exploratory visualization

- "Analytic plots"
- Mostly for ourselves
- Often quick and dirty

Goals

- What's in the data?
- Get a sense of size and complexity of data.
- Explore and interact.
- "Forces us to notice what we never expected to see" (Tukey 1977)

Different goals, different looks

Exploratory visualization

- "Analytic plots"
- Mostly for ourselves
- Often quick and dirty

Goals

- What's in the data?
- Get a sense of size and complexity of data.
- Explore and interact.
- "Forces us to notice what we never expected to see" (Tukey 1977)

Explanatory visualization

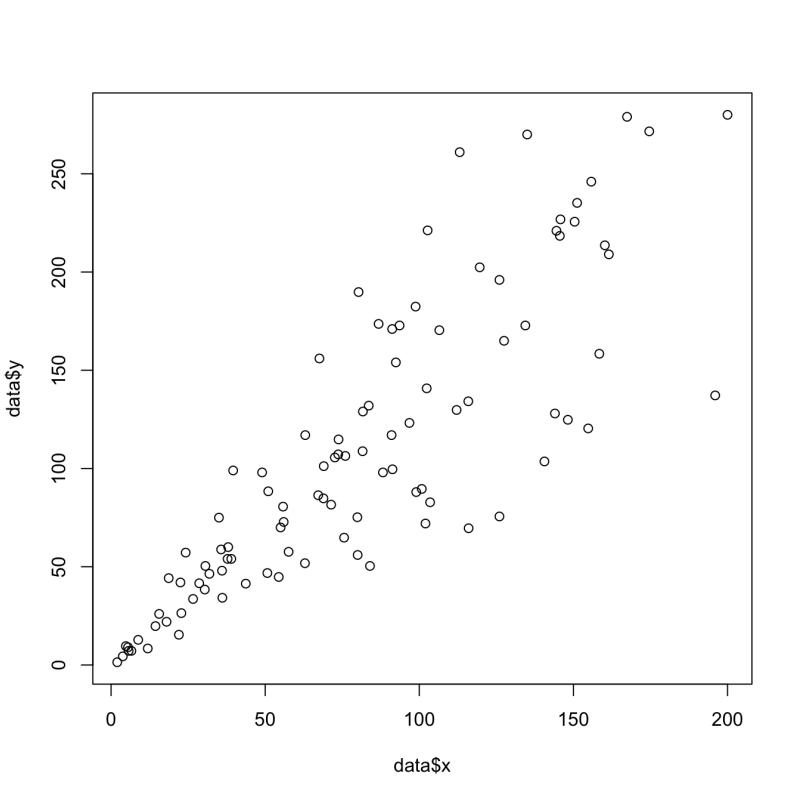
- "Presentation plots"
- For others after the research is completed
- Few, carefully crafted, attractive graphs

Goals

- Communicate content of data.
- Tell a story with data.
- Attract attention and interest.
- "Forces readers to see the information the designer wanted to convey" (Kosslyn 1994)

Different goals, different looks (cont.)

Exploratory visualization



Explanatory visualization

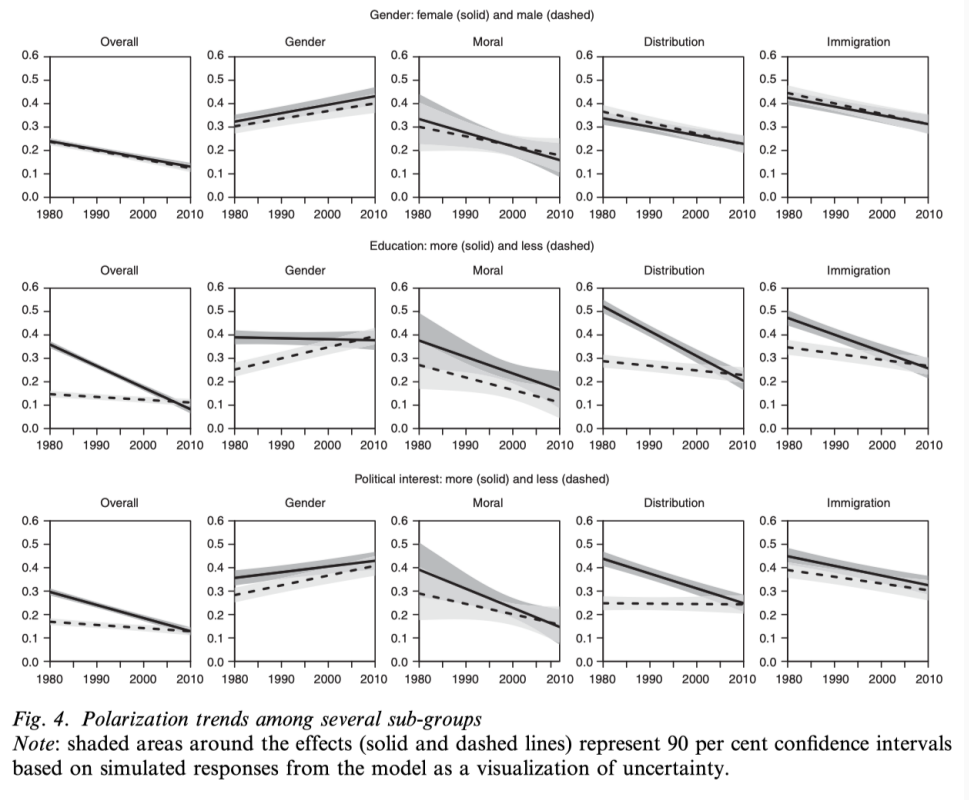


Fig. 4. Polarization trends among several sub-groups
Note: shaded areas around the effects (solid and dashed lines) represent 90 per cent confidence intervals based on simulated responses from the model as a visualization of uncertainty.

The handcraft of visualization

A tool for comparison

- "The fundamental analytical act in statistical reasoning is to answer the question 'compared to what?'
- Whether we are evaluating changes over space or time, searching big data bases, adjusting and controlling for variables, designing experiments, specifying multiple regressions, or doing just about any kind of evidence-based reasoning, the essential point is to make intelligent and appropriate comparisons.
- Thus visual displays, if they are to assist thinking, should show comparisons."

Edward Tufte, Beautiful Evidence, p.127.

The handcraft of visualization

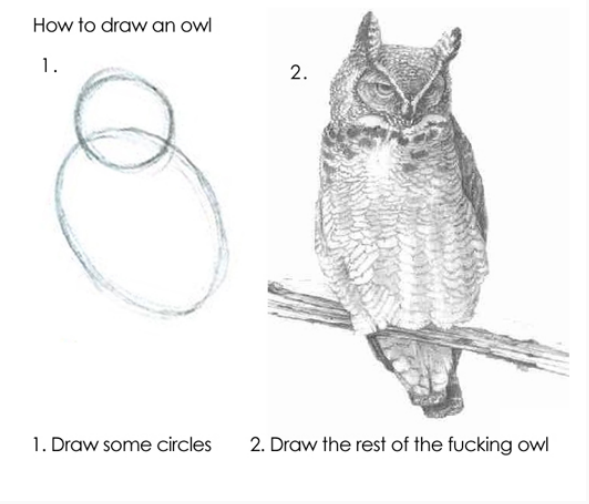
A tool for comparison

- "The fundamental analytical act in statistical reasoning is to answer the question 'compared to what?'
- Whether we are evaluating changes over space or time, searching big data bases, adjusting and controlling for variables, designing experiments, specifying multiple regressions, or doing just about any kind of evidence-based reasoning, the essential point is to make intelligent and appropriate comparisons.
- Thus visual displays, if they are to assist thinking, should show comparisons."

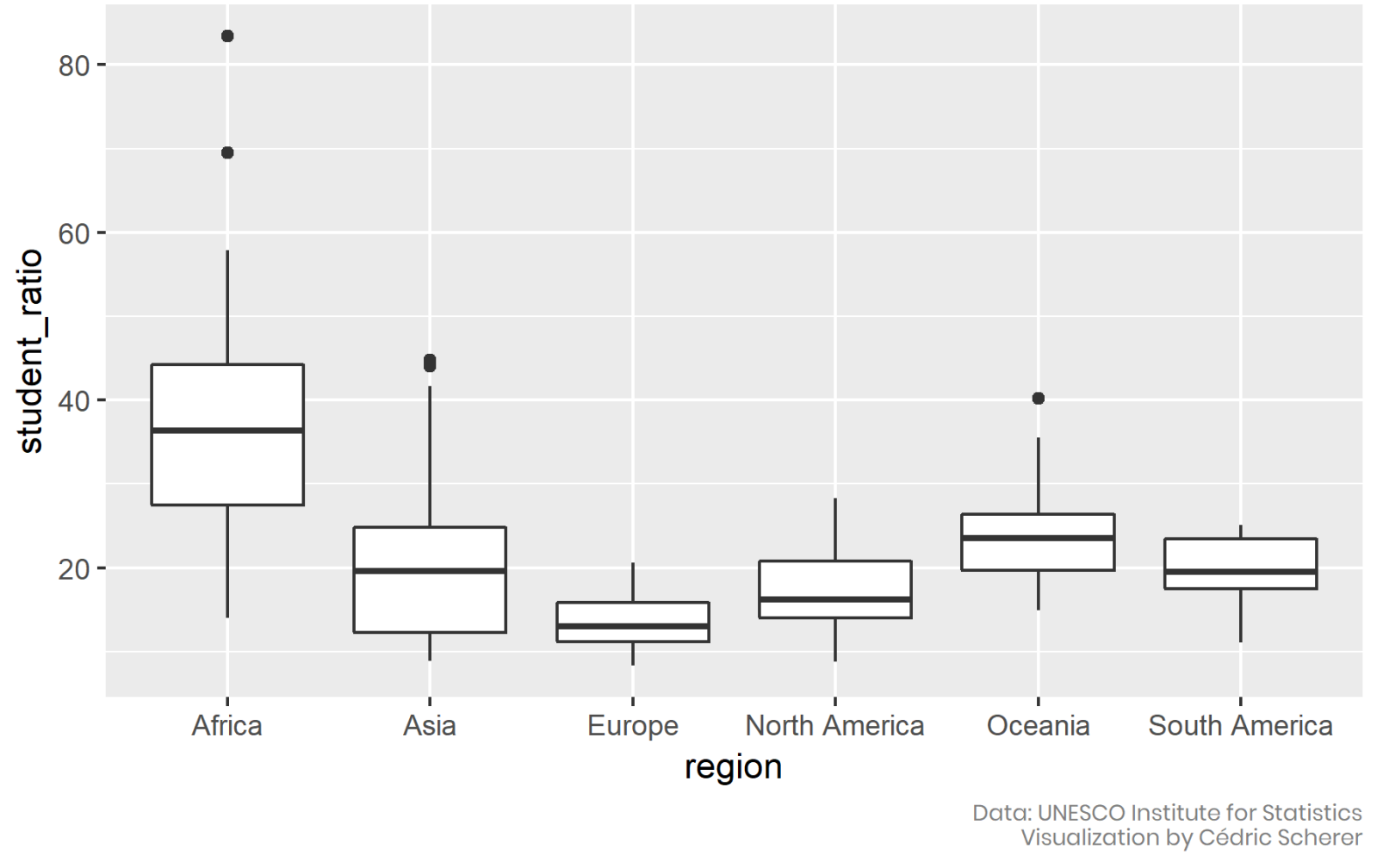
Edward Tufte, Beautiful Evidence, p.127.

Visualization as an iterative process

- The choice of the right graphical format ultimately depends on the task or problem it is trying to solve.
- Always try different graphical formats on the same data – they may reveal different aspects.
- Constructing visualizations is almost always an iterative process – the first graph is rarely also the final one.



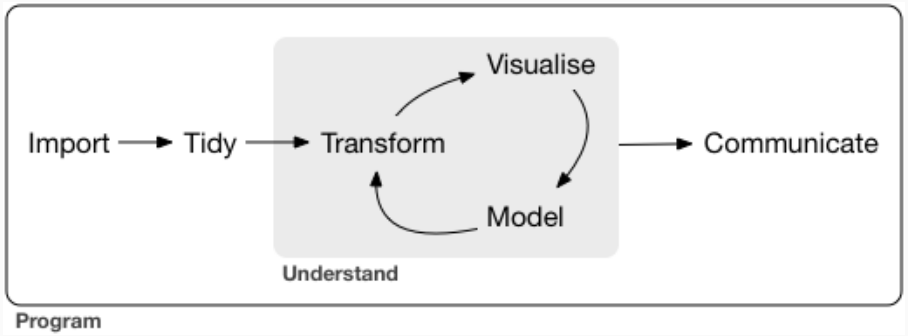
Plotting as an iterative process



`Credit and code` Cedric Scherer

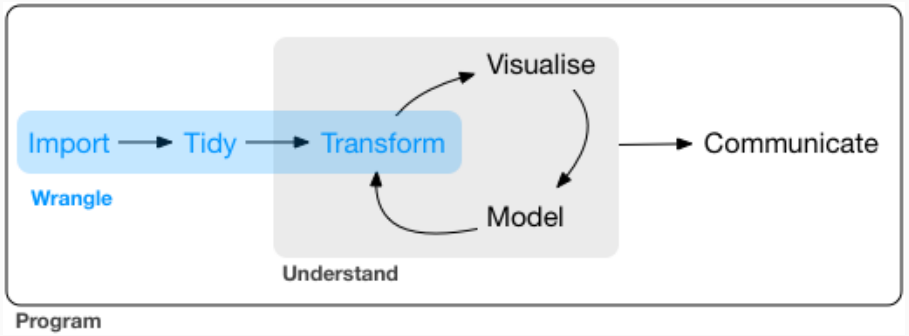
Visualization in the data science workflow

Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.



Visualization in the data science workflow

Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.



Wrangle

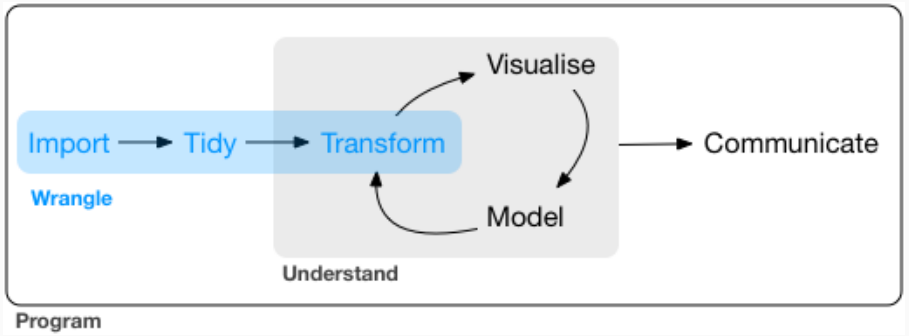
- Sanity checks
- Identification of outliers
- Guidance of recoding operations

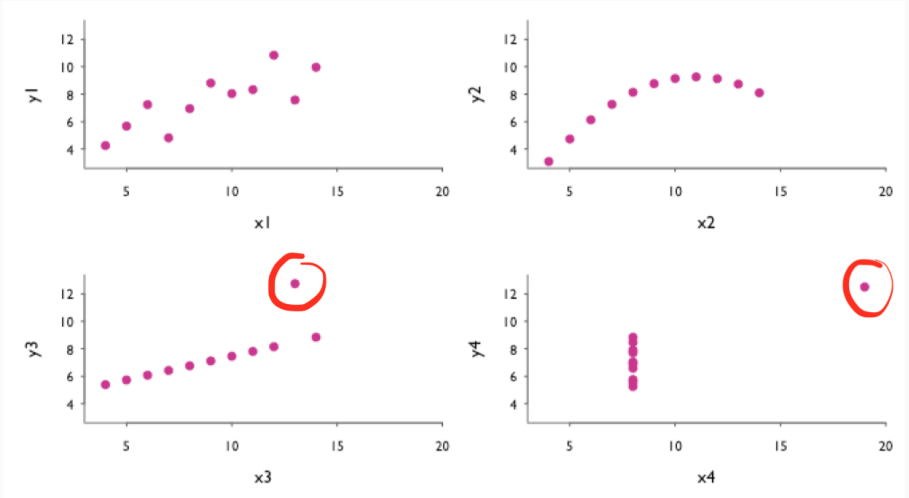
Visualization in the data science workflow

Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.

Wrangle

- Sanity checks
- Identification of outliers
- Guidance of recoding operations

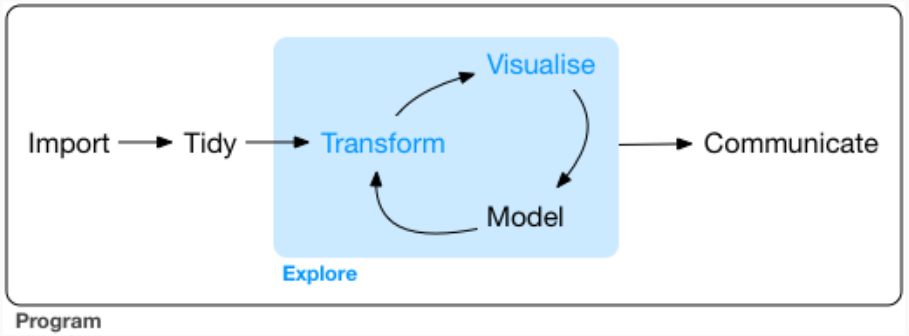




Scatter plots to identify outliers in bivariate relationships.

Visualization in the data science workflow

Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.



Wrangle

- Sanity checks
- Identification of outliers
- Guidance of recoding operations

Explore

- Summarize distributions
- Discover patterns, relationships

Visualization in the data science workflow

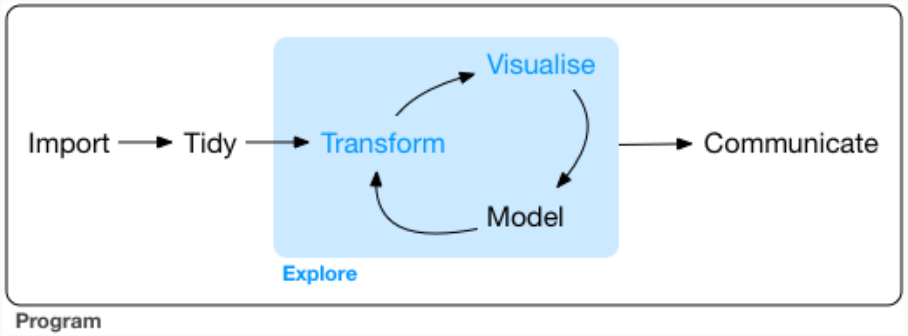
Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.

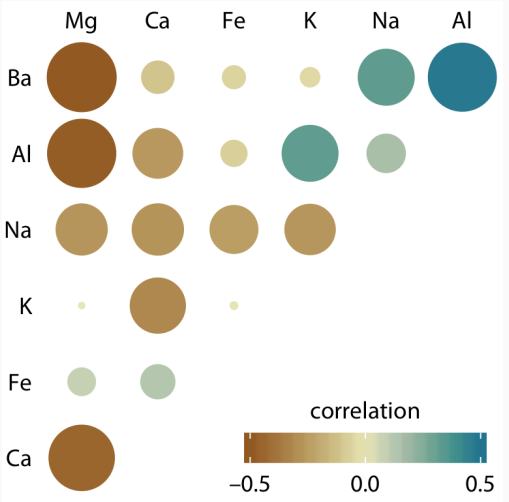
Wrangle

- Sanity checks
- Identification of outliers
- Guidance of recoding operations

Explore

- Summarize distributions
- Discover patterns, relationships





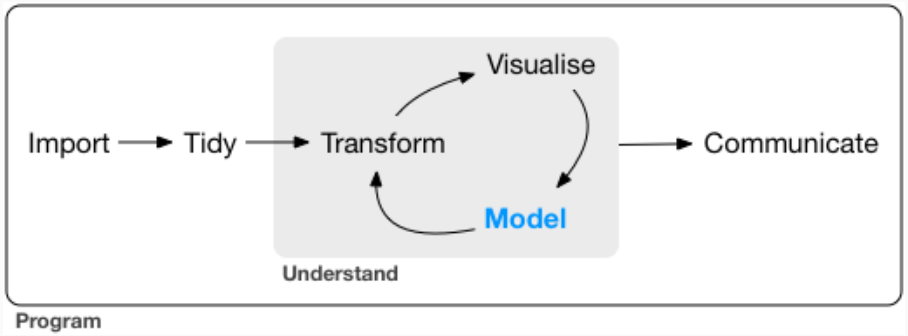
Correlogram to visualize amount of association between pairs of variables

Visualization in the data science workflow (cont.)

Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.

Model

- Test hypotheses
- Summarize (multiple) model estimates
- Visualize uncertainty
- Report robustness/sensitivity analyses

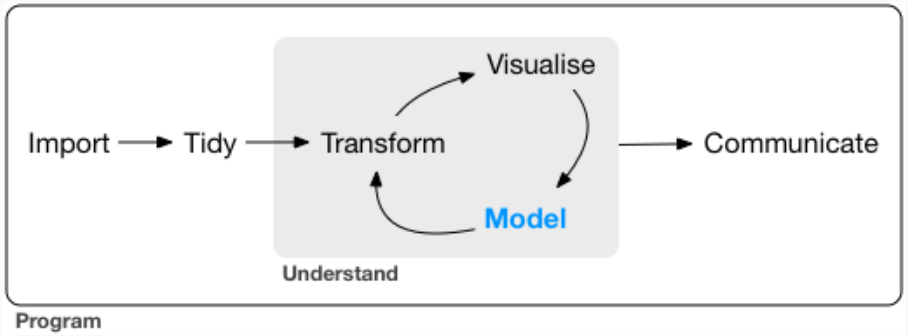


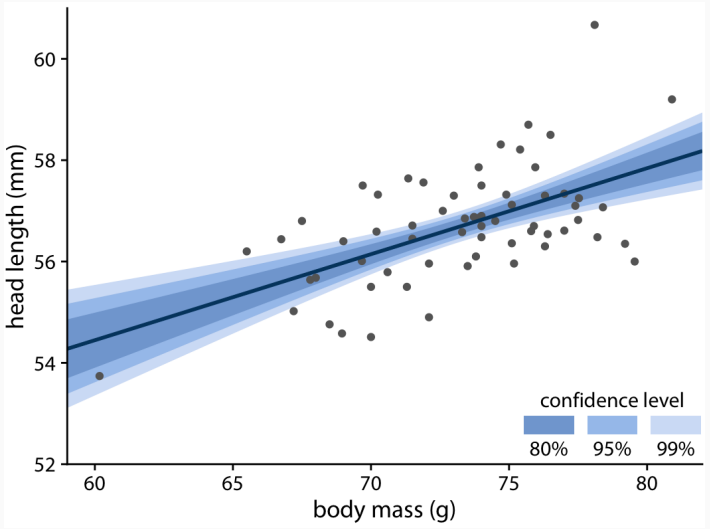
Visualization in the data science workflow (cont.)

Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.

Model

- Test hypotheses
- Summarize (multiple) model estimates
- Visualize uncertainty
- Report robustness/sensitivity analyses

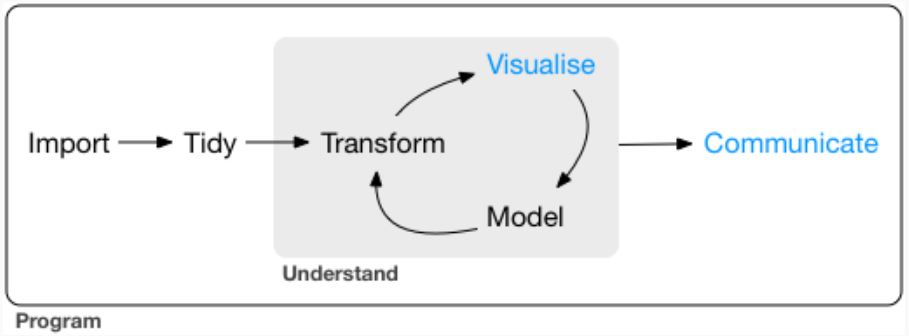




Raw data and trend line with confidence bands to visualize uncertainty of fit

Visualization in the data science workflow (cont.)

Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.



Model

- Test hypotheses
- Summarize (multiple) model estimates
- Visualize uncertainty
- Report robustness/sensitivity analyses

Communicate

- Present raw/cooked data
- Present implications of model results

Visualization in the data science workflow (cont.)

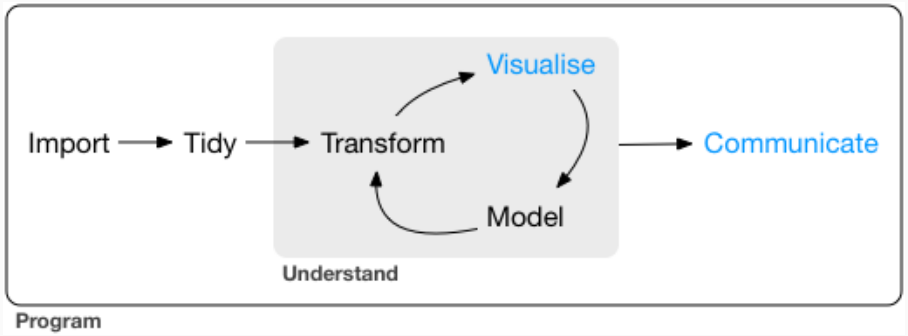
Data visualization is a key skill for data scientists. It is relevant in every step of the workflow.

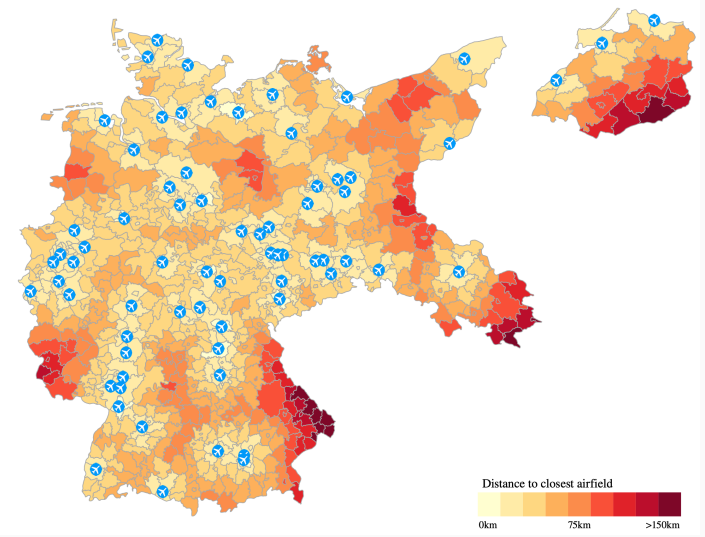
Model

- Test hypotheses
- Summarize (multiple) model estimates
- Visualize uncertainty
- Report robustness/sensitivity analyses

Communicate

- Present raw/cooked data
- Present implications of model results





Choropleth map illustrating the location of civilian airfields in the German Empire, 1932. Administrative counties are shaded according to their centroid's distance to the closest airfield

Visual inference

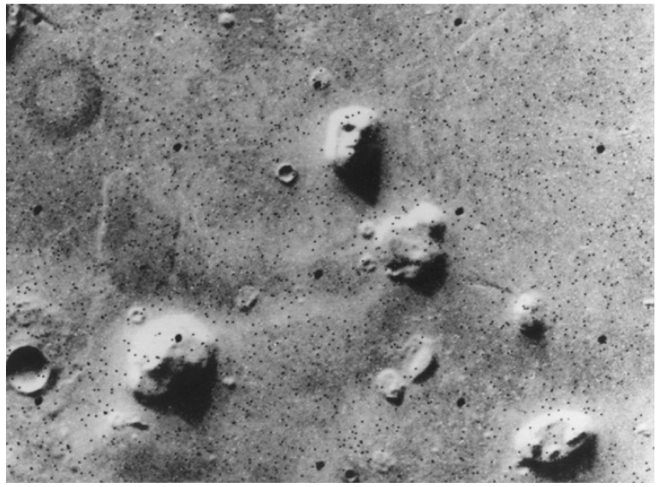
Human talent and weakness

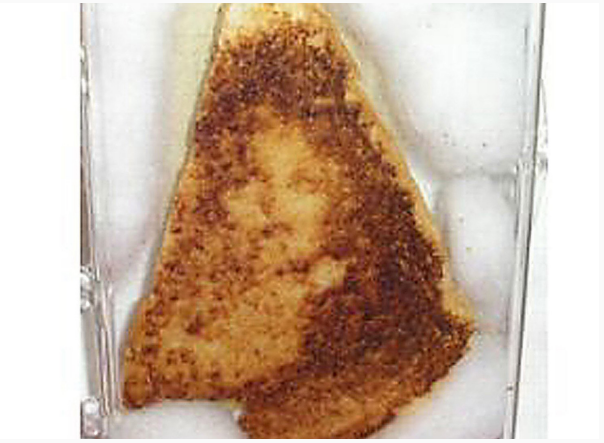
- Humans are extremely good at recognizing patterns.
- At the same time, humans are also extremely good at inferring patterns when there are none (tendency to see patterns in random data = "apophenia").
- This is somewhat linked to the fact that our species is bad at dealing with probability and randomness.

Concerns with exploratory data analysis

- A concern that frequently arises with exploratory analysis is that it lacks the rigor of formal tests in confirmatory analysis or conventional statistical inference.
- Long-standing reservations against visualization as merely "informal" approach to data analysis and the fear that beautiful pictures may in fact not correspond to any meaningful patterns of substantive scientific interest.

Image of Mars taken by NASA's Viking 1 orbiter, in grey scale, on July, 25 1976.





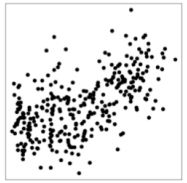
Visual inference (cont.)

Overcoming the exploratory vs. confirmatory visualization divide

- Graphical displays are implicit or explicit comparisons to a reference distribution or baseline model.
- If we discover an interesting pattern in data, this usually means that it looks different from what we expected.
- We usually have implicit models in our mind to which we compare the data ("What do we expect to see?").
- We can make these models explicit and use them to guard against "false discoveries".
- Visual discoveries correspond to the implicit or explicit rejection of null hypotheses (Buja et al. 2009).

Visual inference as an analogue to null hypothesis significance testing

The basic principle of formal testing remains the same in visual inference – with the exception that the test statistic is now a graphical display which is compared to a "reference distribution" of plots showing the null:

Formal Test	Visual Inference
Null hypothesis H_0	Null hypothesis H_0
Test statistic $T = f(x)$	Visual feature in a plot 
Test: Reject? $T(x) > c$?	Human viewer: Discovery?

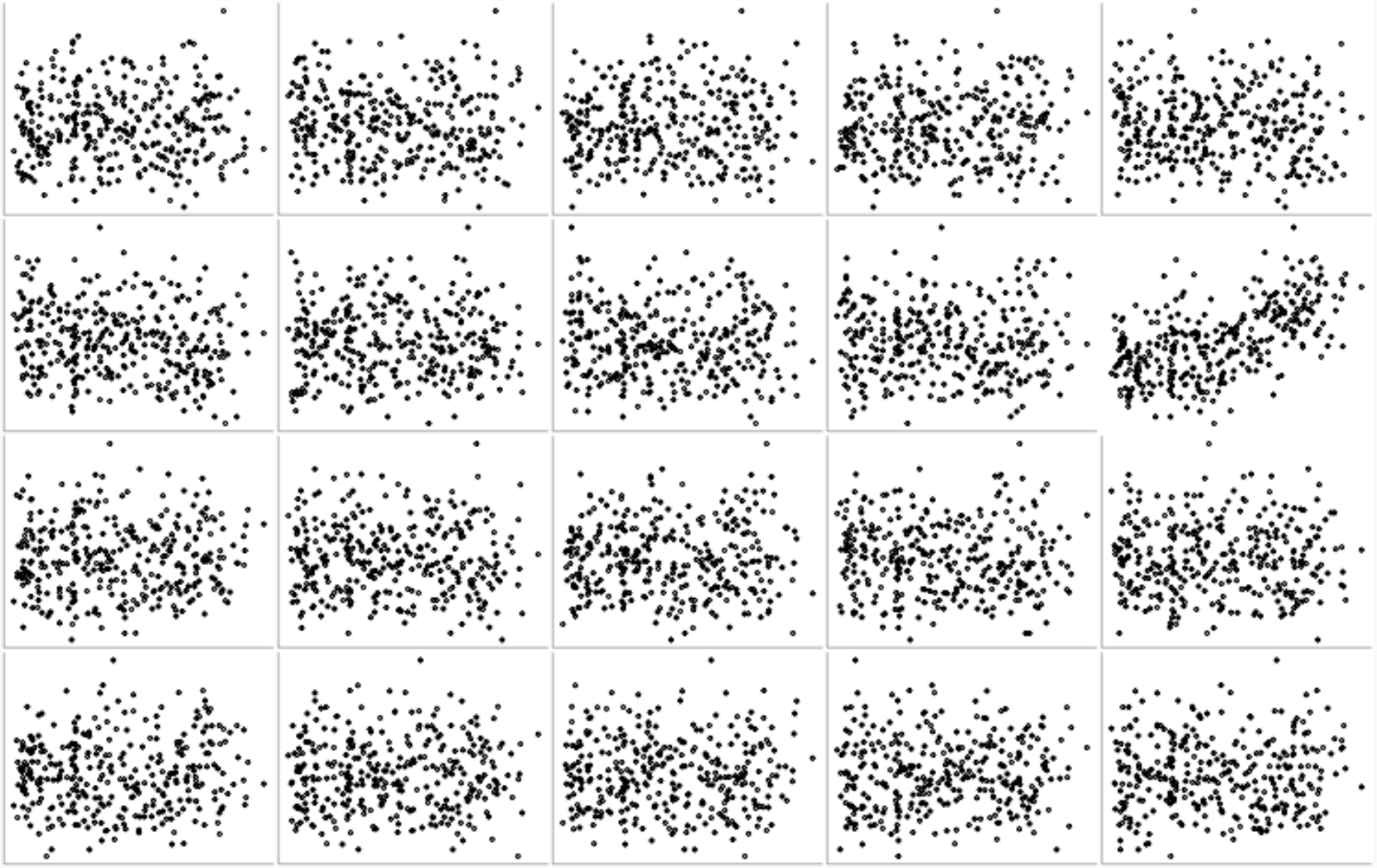
Visual inference: the line-up protocol

This method is called "after the 'police lineup' of criminal investigations [...], because it asks the witness to identify the plot of the real data from among a set of decoys, the null plots, under the veil of ignorance" (Buja et al. 2009).

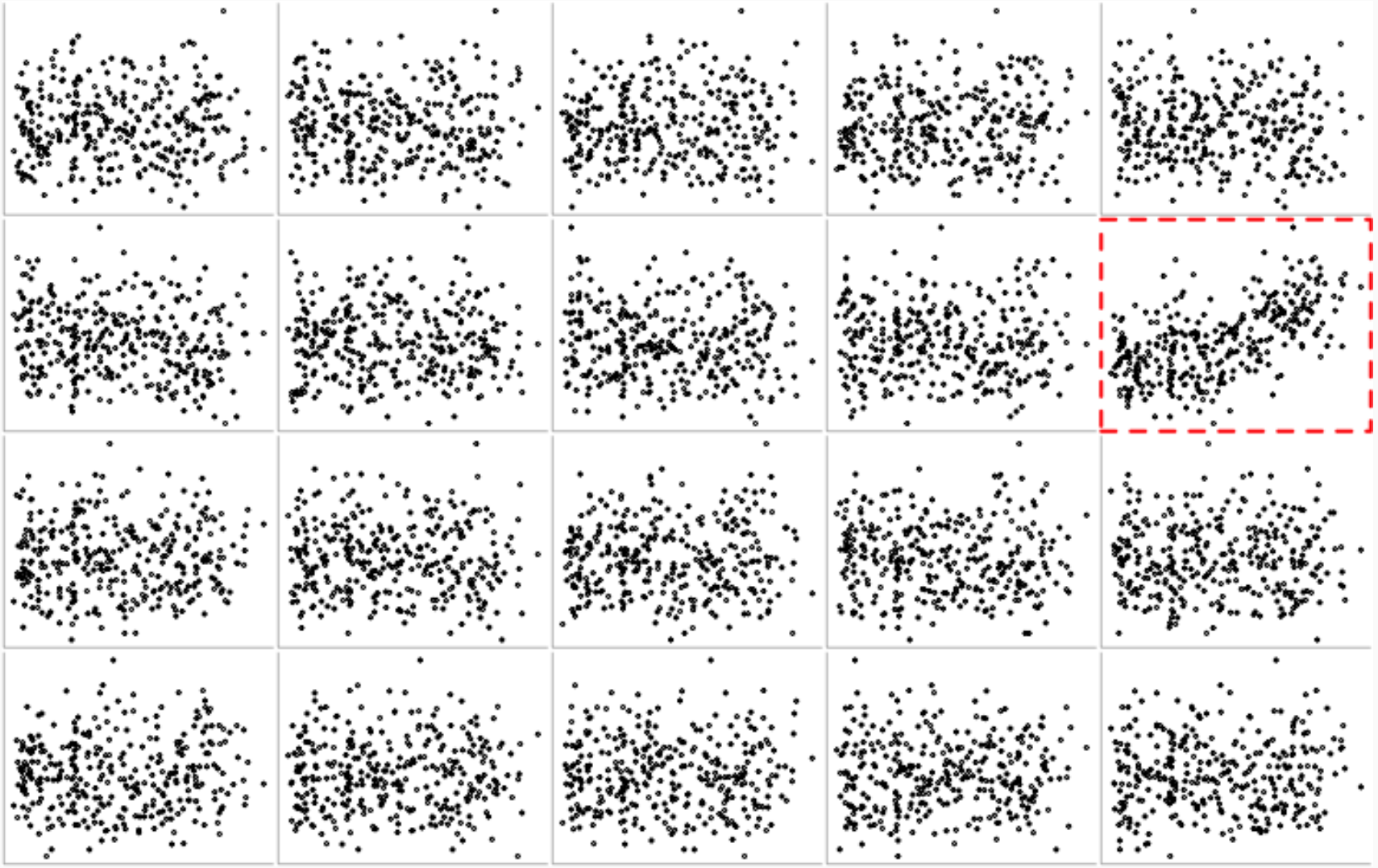
The visual hypothesis test involves the following steps:

1. Simulate data to create $m - 1$ null plots.
2. Randomly place the plot of the real data among them, resulting in a total of m plots.
3. Ask a human viewer to choose the plot that looks the most different from the rest.
4. If the test person succeeds and picks the plot showing the actual data, then this visual discovery can be assigned a p-value of $1/m$. In other words, the probability of picking the true plot just by chance is $1/m$.

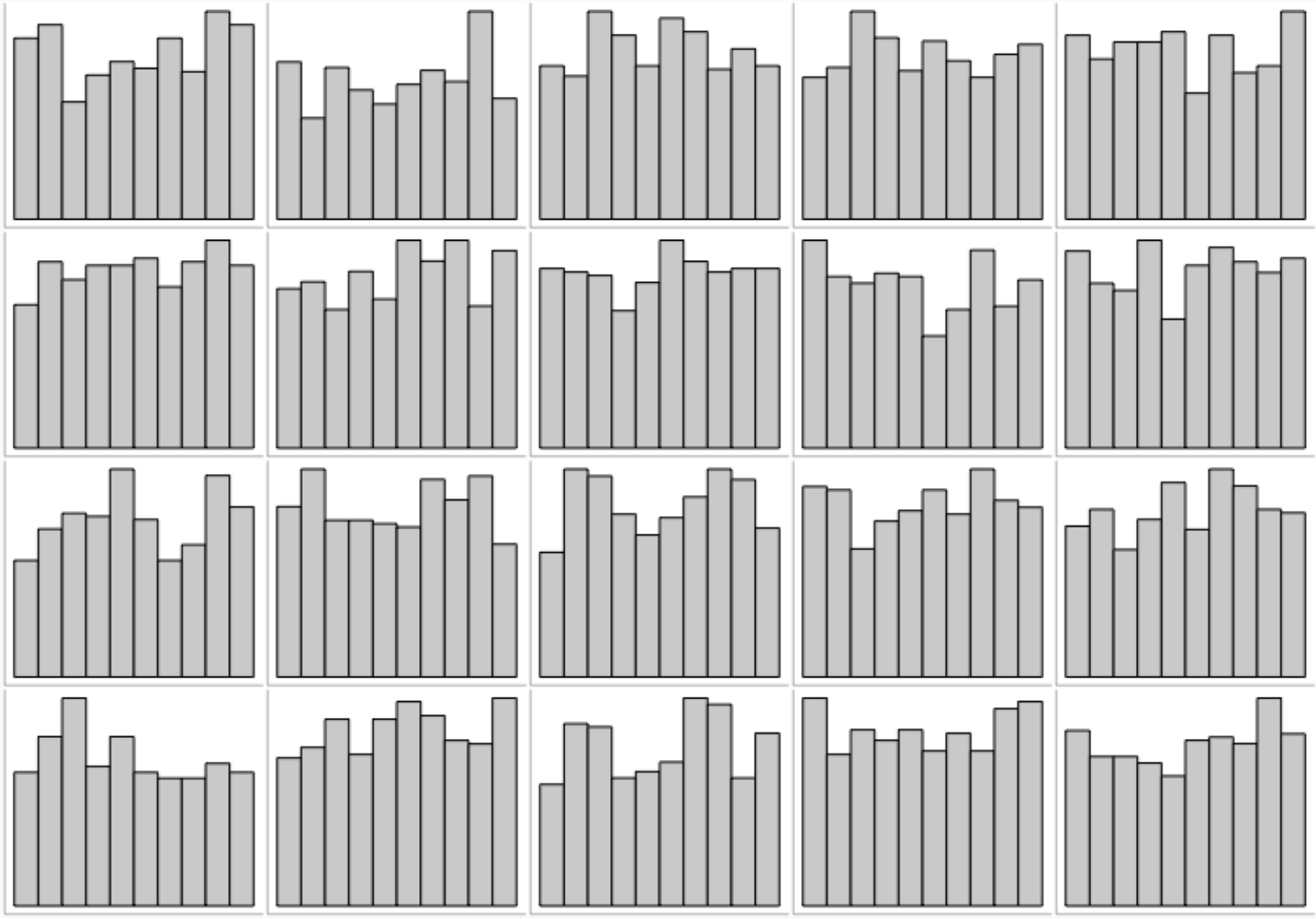
Visual inference: which plot stands out from the rest?



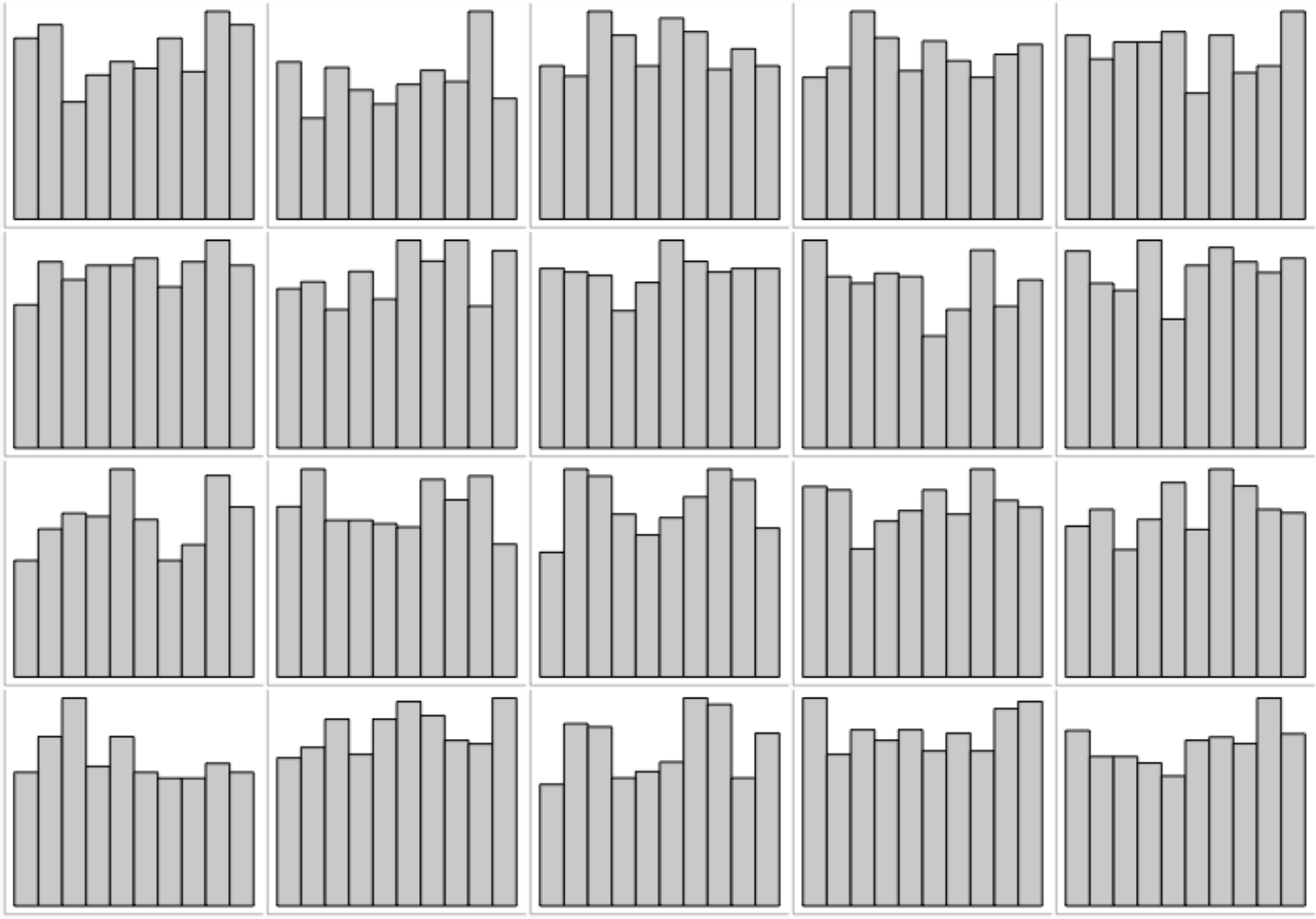
Visual inference: which plot stands out from the rest?



Visual inference: which plot stands out from the rest?



None! All just show runif(500, 0, 1).



Types of data visualization

Different plot types for different purposes

A common mistake in visualization is that plot types are used for purposes they are not meant for. You'll gain more intuition and experience in picking the right types over time.

Before you start plotting, ask yourself:

Different plot types for different purposes

A common mistake in visualization is that plot types are used for purposes they are not meant for. You'll gain more intuition and experience in picking the right types over time.

Before you start plotting, ask yourself:

1. Which quantity do I want to visualize?

- Amounts
- Distributions
- Proportions
- Associations
- Structures
- Trends
- Estimates
- Predictions
- Uncertainty

Different plot types for different purposes

A common mistake in visualization is that plot types are used for purposes they are not meant for. You'll gain more intuition and experience in picking the right types over time.

Before you start plotting, ask yourself:

1. Which quantity do I want to visualize?

- Amounts
- Distributions
- Proportions
- Associations
- Structures
- Trends
- Estimates
- Predictions
- Uncertainty

2. Which question do I want to answer?

- "Is the *distribution* normal (or uniform or...)?" → **Histogram, density plot, Q-Q plot**
- "Are univariate *distributions* across subgroups different?" → **Boxplots, ridgelines**
- "How do *differences in amounts* between groups compare?" → **Barplot, dotplot**
- "What is the *relationship* between x and y?" → **Scatterplot, contour plot, hex bins**
- "What are the *correlations* in a set of variables?" → **Correlogram, small multiples**
- "How did a *trend* develop over time?" → **Line graph, slopegraph**
- "Are the data *clustered* by subgroup?" → **Scatterplot with color**
- "Is there a *spatial pattern*?" → **Choropleth, cartogram heatmap**
- "What are the relative and absolute *effect sizes*? → **Coefficient plot**
- "How uncertain are *estimates*? → **Error bars, confidence bands**

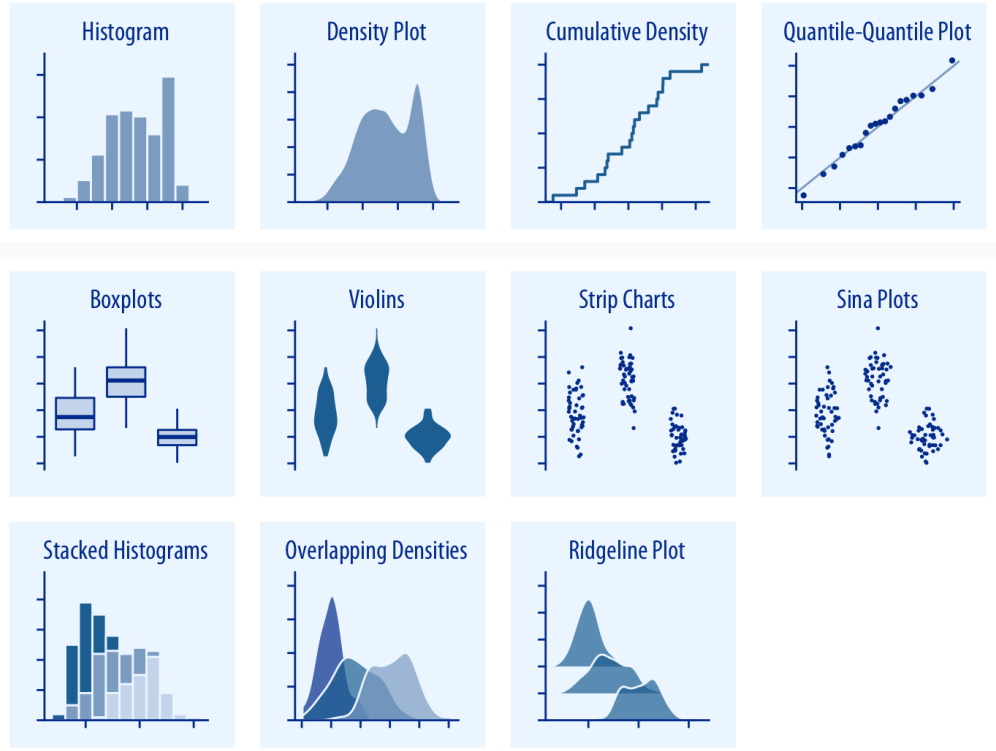
A directory of visualizations

Visualizing amounts

A directory of visualizations

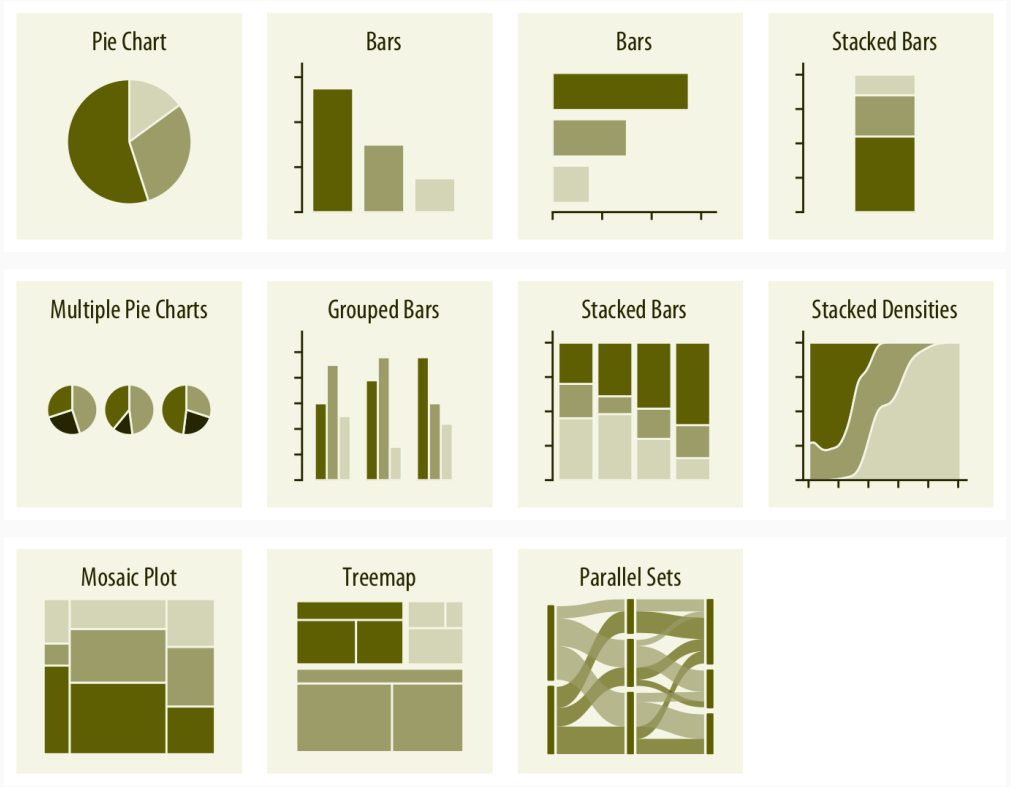
Visualizing amounts

Visualizing distributions



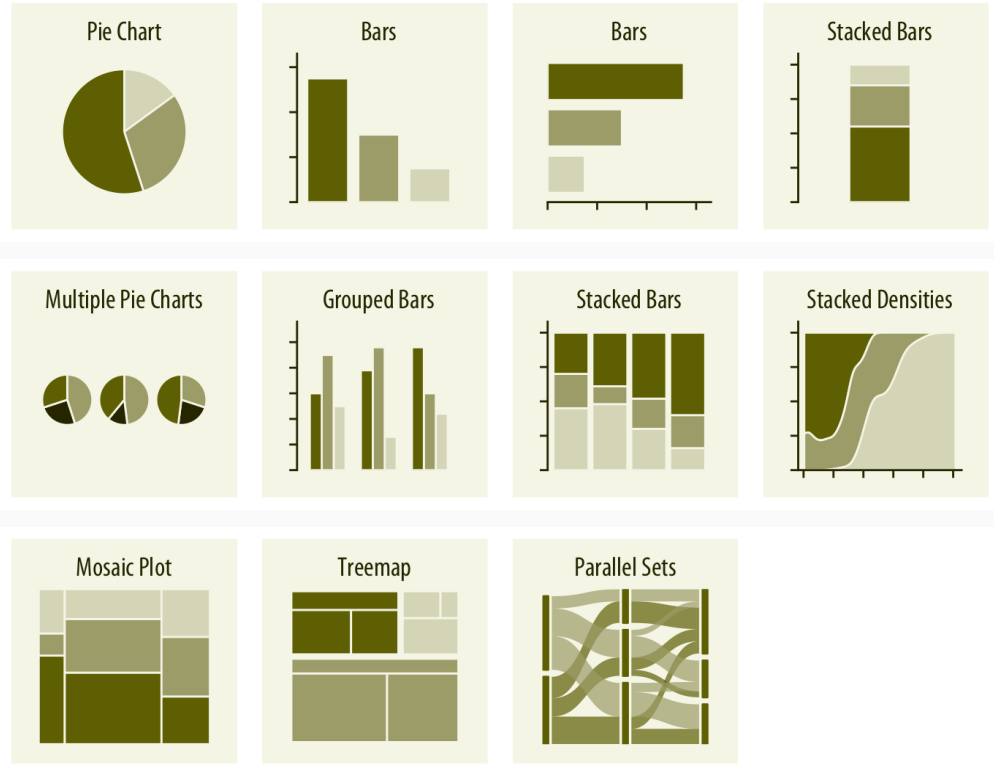
A directory of visualizations (cont.)

Visualizing proportions

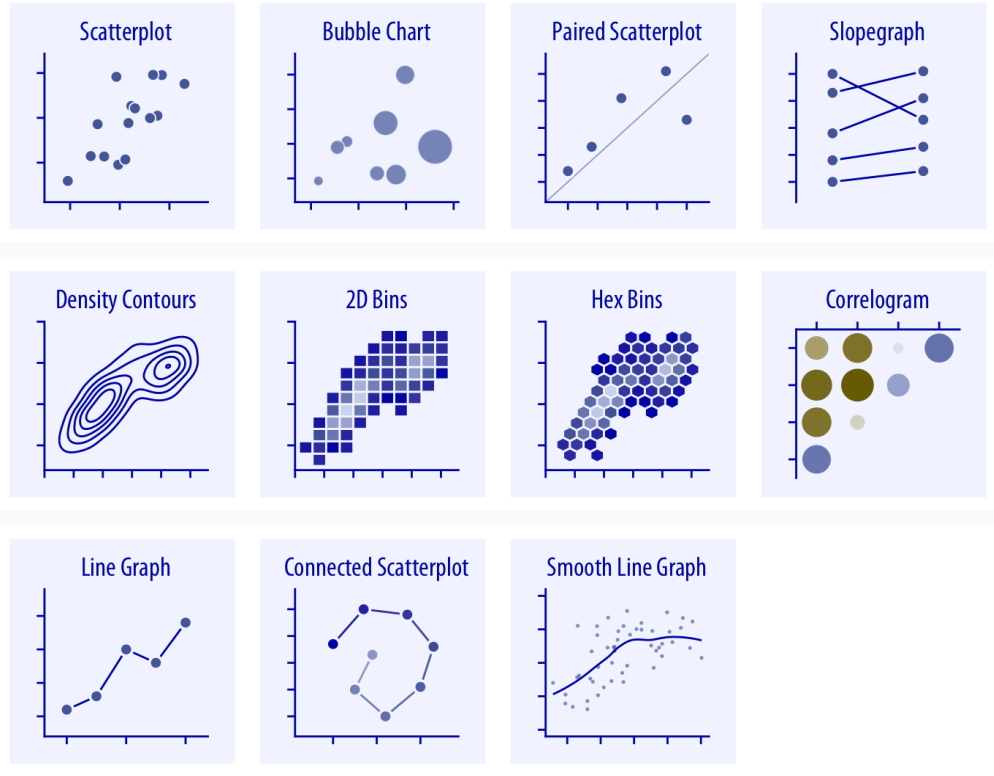


A directory of visualizations (cont.)

Visualizing proportions

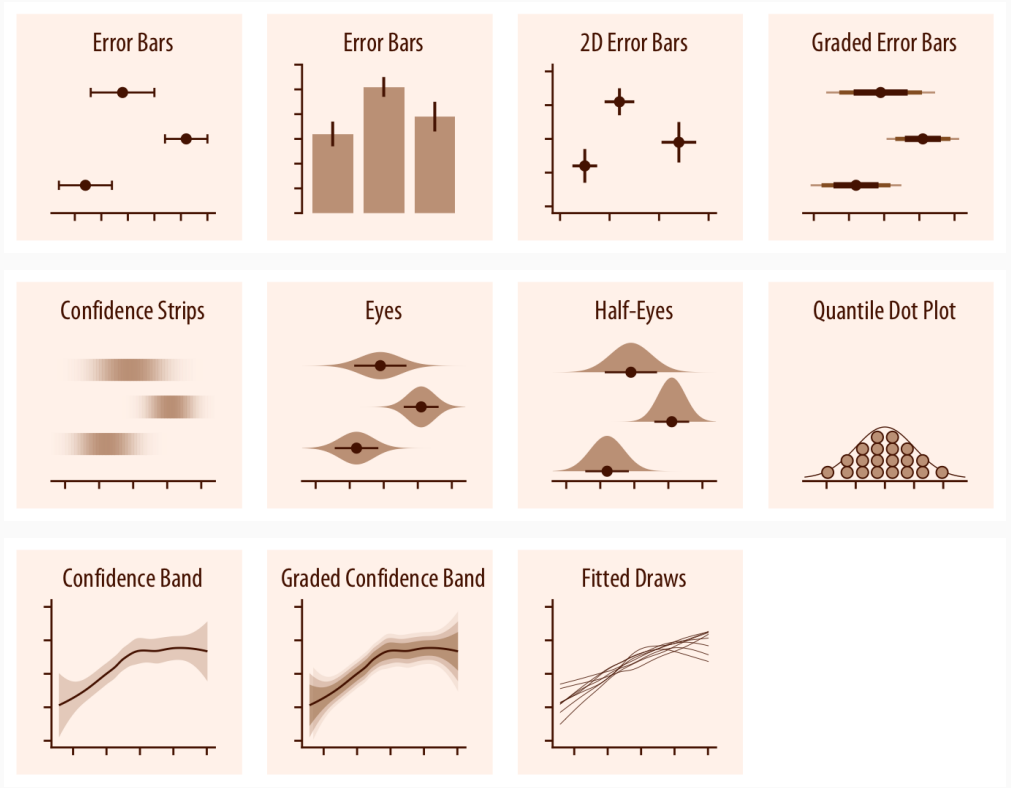


Visualizing x-y relationships



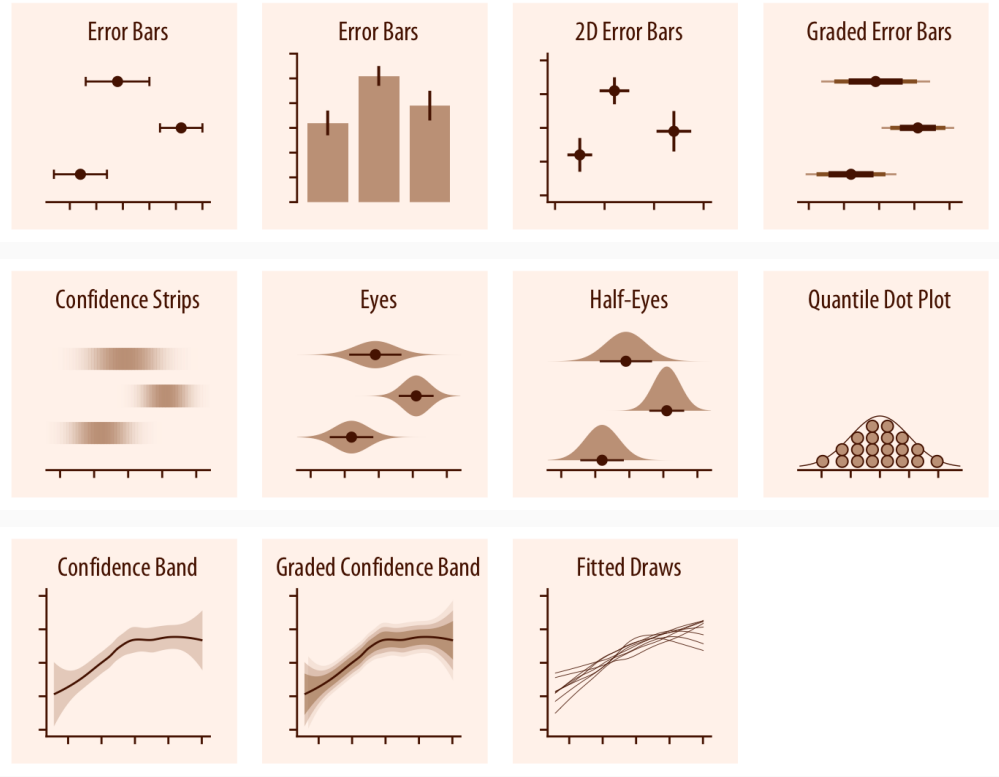
A directory of visualizations (cont.)

Visualizing uncertainty

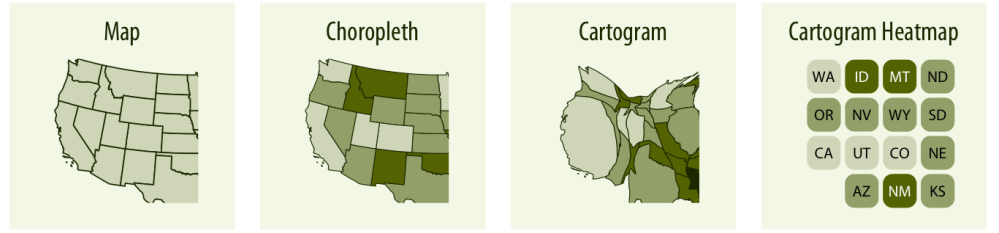


A directory of visualizations (cont.)

Visualizing uncertainty

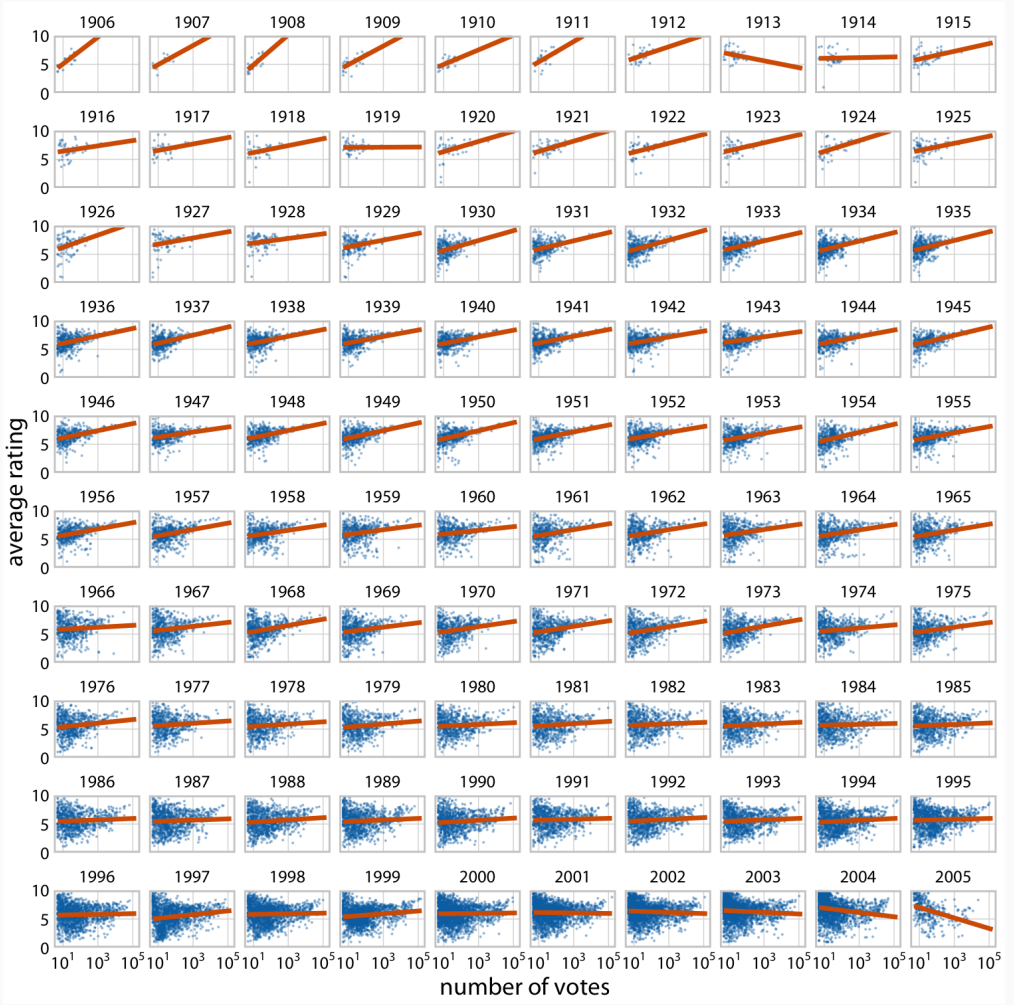


Visualizing geospatial data



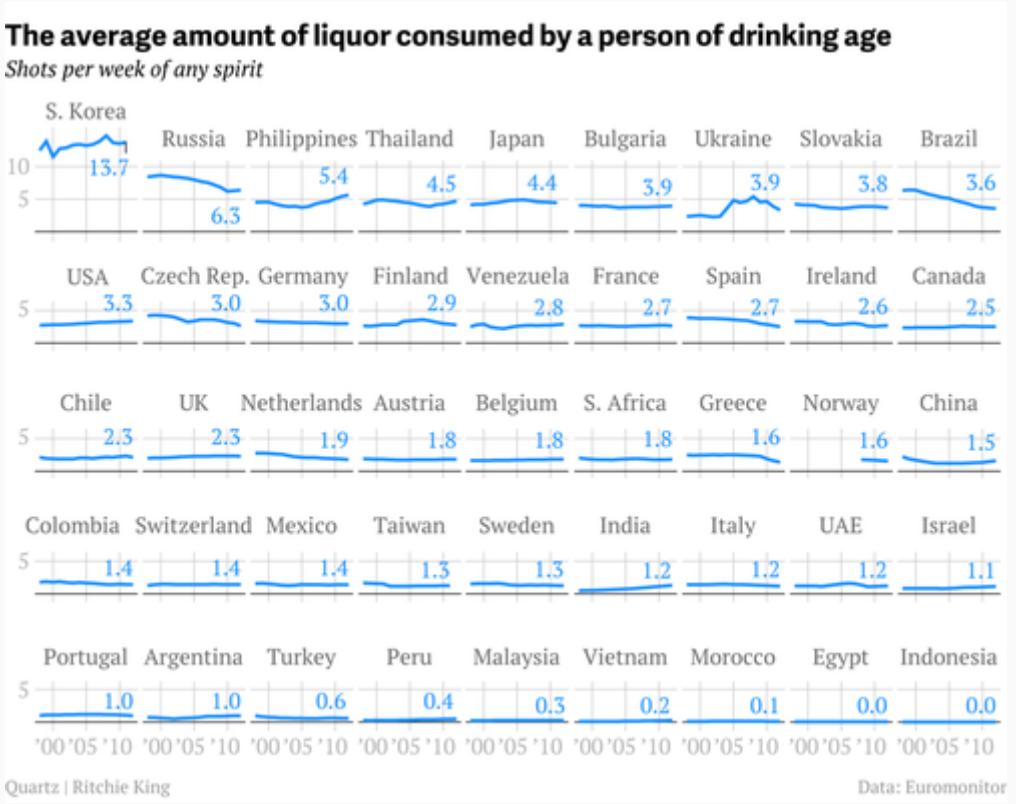
Small multiples

- A powerful yet underestimated visualization strategy is to use multi-panel figures.
- Often we want to compare relationships or trends between groups. With many groups, that's too much information for a single figure panel.
- There are various terms for multi-panel figures, including "small multiples" (Tufte 1990), "trellis plot" (Cleveland 1993) and "faceting" (Wickham 2016)
- In R we can implement this fairly easily with `ggplot`'s `facet_grid()` (or `facet_wrap()`).
- If you do small multiples, make sure to use:
  - common graph size
  - common axis scales
  - helpful alignment and order of panels



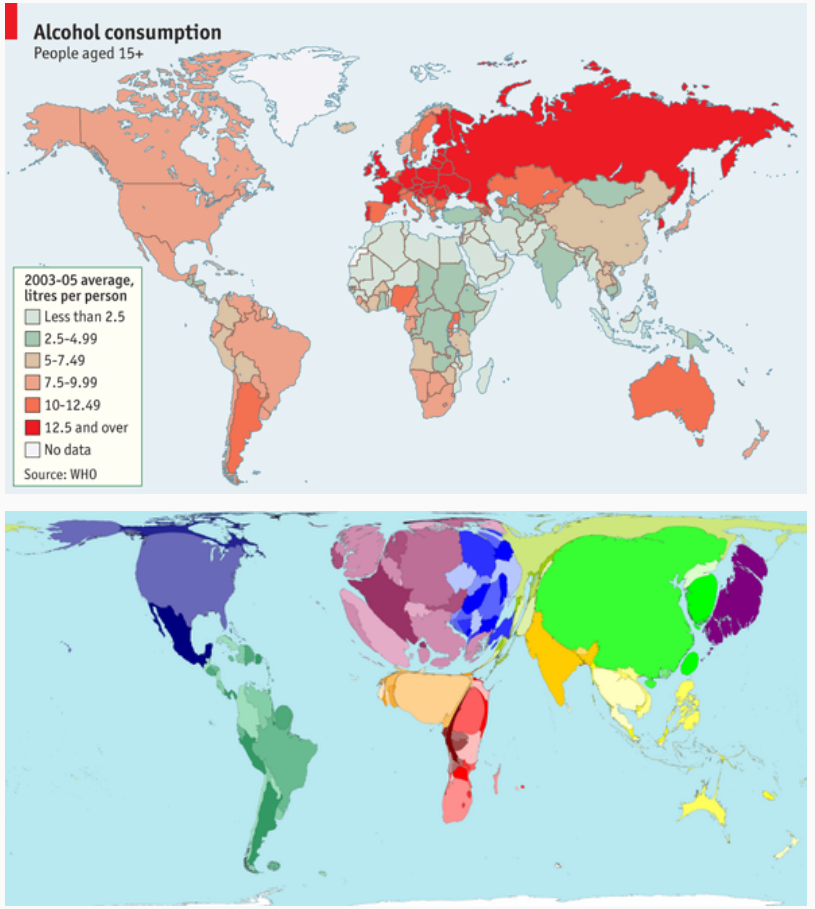
Small multiples (cont.)

- Check out the example on the right, which was also discussed here and there.
- What they did right (by Kaiser Fung):
  - Did not put the data on a map
  - Ordered the countries by the most recent data point rather than alphabetically
  - Scale labels are found only on outer edge of the chart area, rather than one set per panel
  - Only used three labels for the 11 years
  - Did not overdo the vertical scale either
  - The nicest feature was the XL scale applied only to South Korea. This destroys the small-multiples principle but draws attention to the top left corner, where the designer wants our eyes to go.



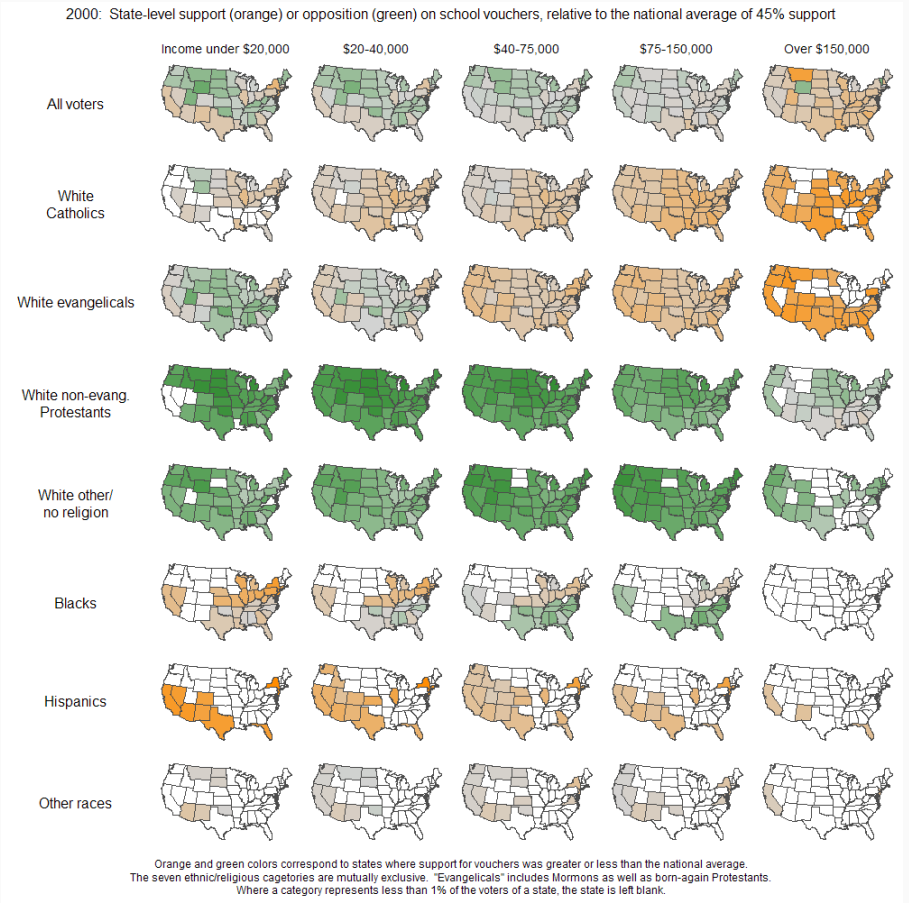
Small multiples (cont.)

- Check out the example on the right, which was also discussed here and there.
- What they did right (by Kaiser Fung):
  - Did not put the data on a map
  - Ordered the countries by the most recent data point rather than alphabetically
  - Scale labels are found only on outer edge of the chart area, rather than one set per panel
  - Only used three labels for the 11 years
  - Did not overdo the vertical scale either
  - The nicest feature was the XL scale applied only to South Korea. This destroys the small-multiples principle but draws attention to the top left corner, where the designer wants our eyes to go.
- Sometimes maps are not a good alternative.



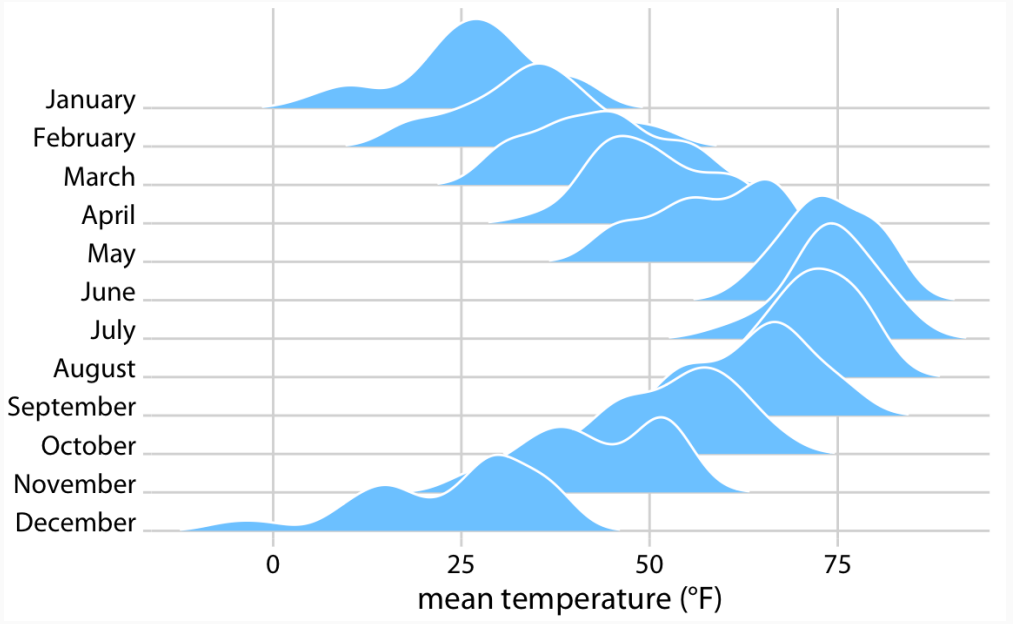
Small multiples (cont.)

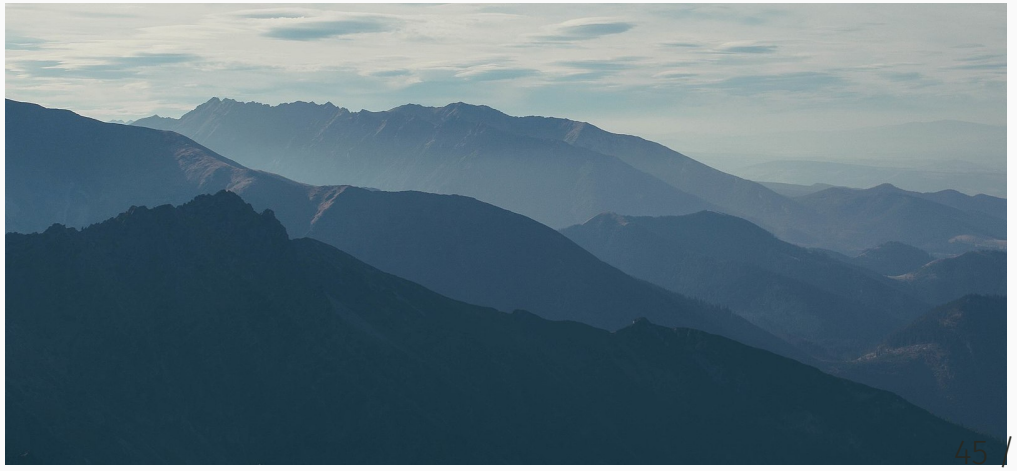
- Check out the example on the right, which was also discussed here and there.
- What they did right (by Kaiser Fung):
  - Did not put the data on a map
  - Ordered the countries by the most recent data point rather than alphabetically
  - Scale labels are found only on outer edge of the chart area, rather than one set per panel
  - Only used three labels for the 11 years
  - Did not overdo the vertical scale either
  - The nicest feature was the XL scale applied only to South Korea. This destroys the small-multiples principle but draws attention to the top left corner, where the designer wants our eyes to go.
- Sometimes maps are not a good alternative.
- But you can do small multiples of maps!



Visualizing many distributions at once

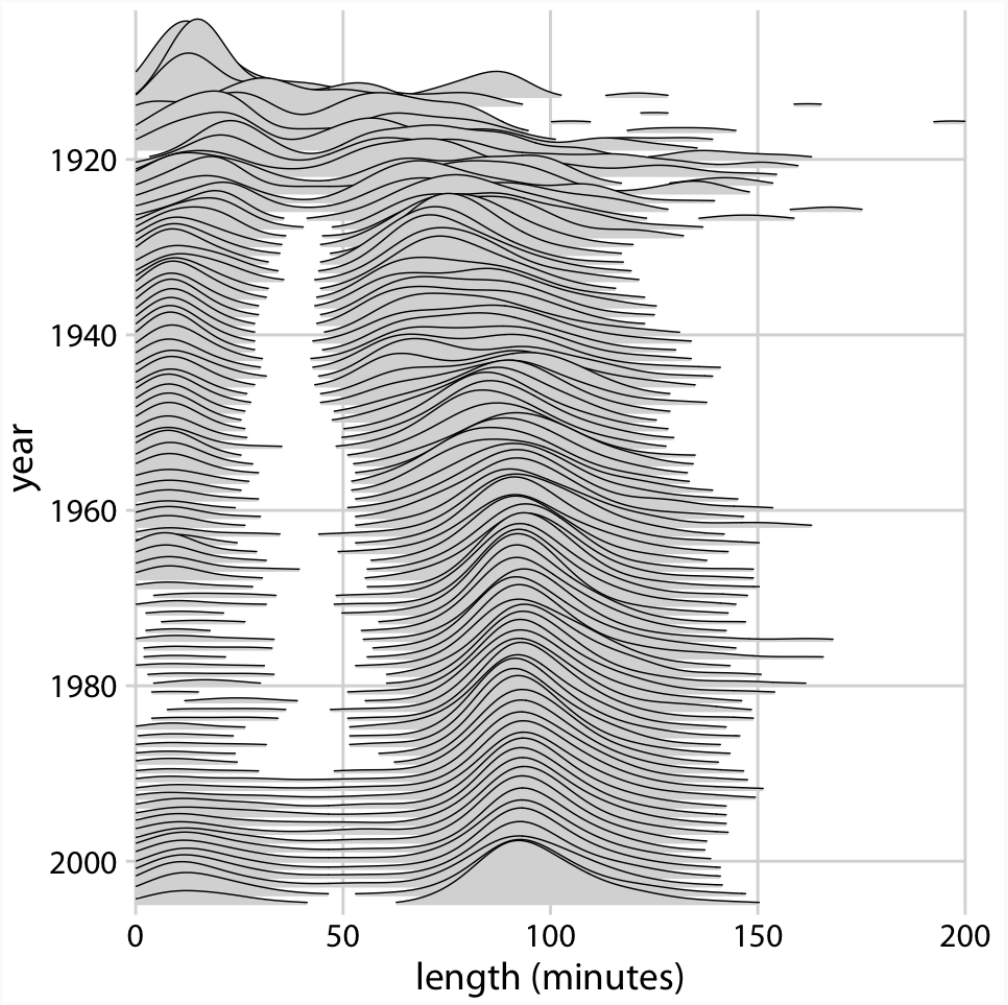
- An increasingly popular, compact variant of small multiples for distributions is the **ridgeline plot** (they look like mountain ridgelines).
- The idea is to staggering distributions plots in the vertical direction (i.e., along the horizontal axis).
- Ridgeline plots tend to work particularly well if want to show trends in distributions over time.





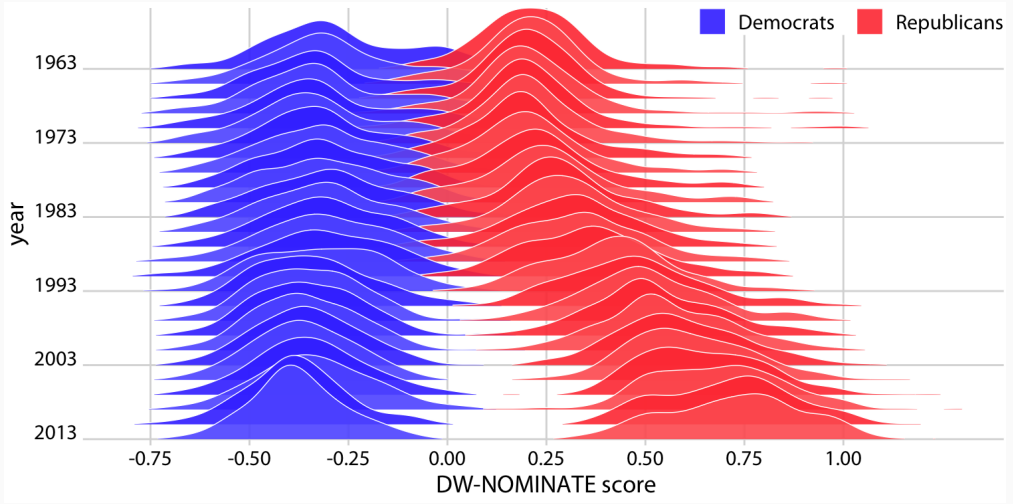
Visualizing many distributions at once (cont.)

- An increasingly popular, compact variant of small multiples for distributions is the **ridgeline plot** (they look like mountain ridgelines).
- The idea is to staggering distributions plots in the vertical direction (i.e., along the horizontal axis).
- Ridgeline plots tend to work particularly well if want to show trends in distributions over time.
- Ridgelines scale to large numbers of distributions.



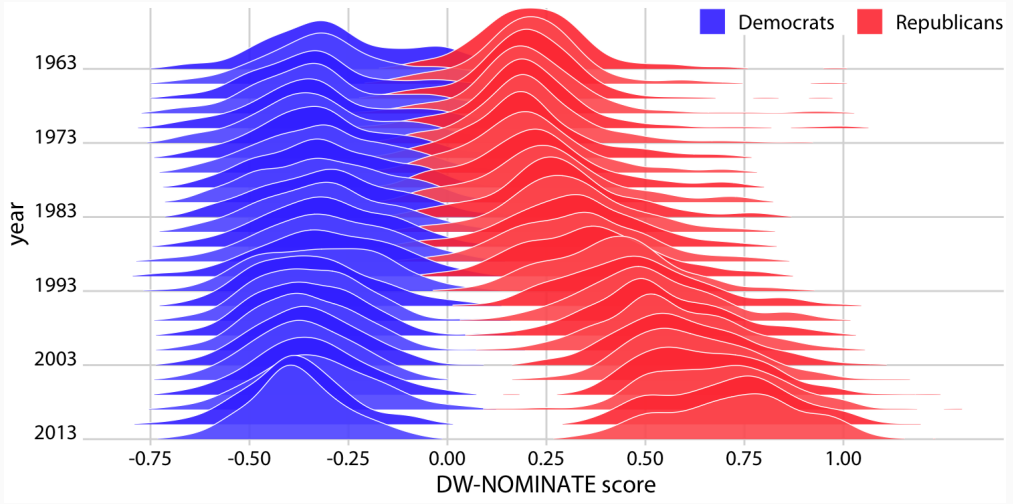
Visualizing many distributions at once (cont.)

- An increasingly popular, compact variant of small multiples for distributions is the **ridgeline plot** (they look like mountain ridgelines).
- The idea is to staggering distributions plots in the vertical direction (i.e., along the horizontal axis).
- Ridgeline plots tend to work particularly well if want to show trends in distributions over time.
- Ridgelines scale to large numbers of distributions.
- Ridgelines can be grouped.



Visualizing many distributions at once (cont.)

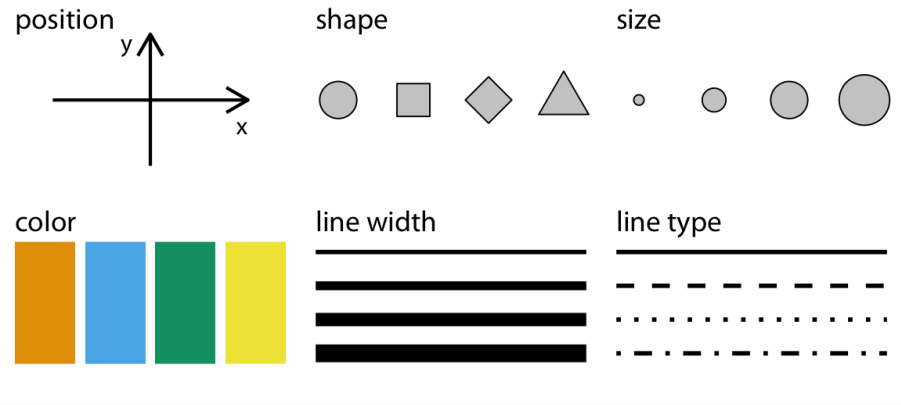
- An increasingly popular, compact variant of small multiples for distributions is the **ridgeline plot** (they look like mountain ridgelines).
- The idea is to staggering distributions plots in the vertical direction (i.e., along the horizontal axis).
- Ridgeline plots tend to work particularly well if want to show trends in distributions over time.
- Ridgelines scale to large numbers of distributions.
- Ridgelines can be grouped.
- They are implemented in R with the `ggridges` package.



Ingredients of data visualization

Mapping data onto aesthetics

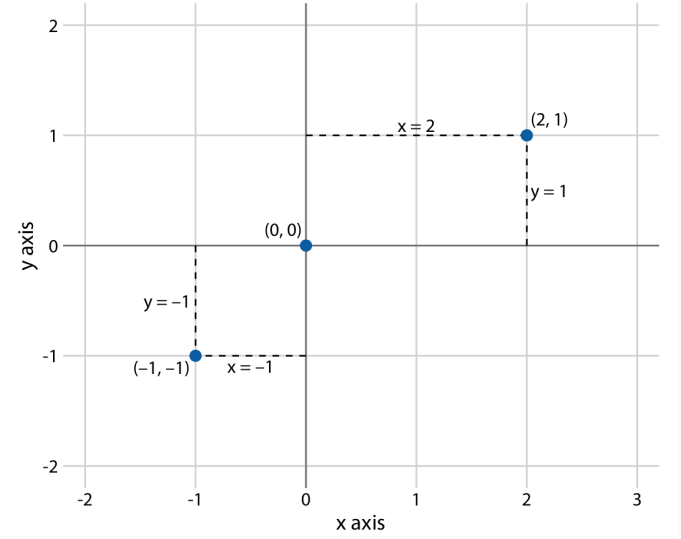
- Whenever we visualize data, we take data values and convert them in a systematic and logical way into the visual elements that make up the final graphic.
- Even though there are many different types of data visualizations, all these visualizations can be described with a common language.
- All data visualizations map data values into quantifiable features of the resulting graphic. We refer to these features as **aesthetics**.
- Key aesthetics are:



- All aesthetics fall into one of two groups: Those that can represent continuous data (e.g., position, size, color) and those that can not (e.g., shape, line type).

Coordinate systems and axes

- Positions of data values matter. Usually, we need two position scales (x and y axis of the plot).
- The combination of a set of position scales and their relative geometric arrangement is called a **coordinate system**.
- Often we have two axes representing two different **units**.
- In a **Cartesian** coordinate system, the grid lines along an axis are spaced evenly both in data units and in the resulting visualization.
- There are scenarios where nonlinear scales are preferred. In a nonlinear scale, even spacing in data units corresponds to uneven spacing in the visualization (e.g., log scales).
- Be sure to **label axes properly**!

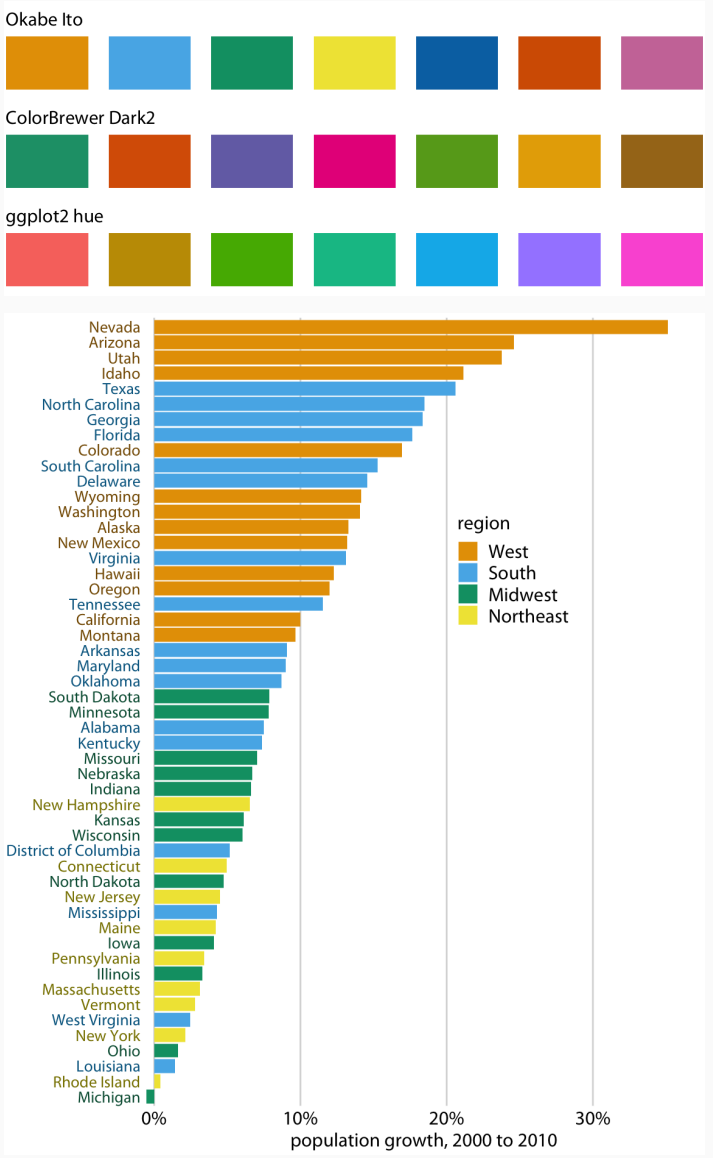


Colors

- There are three fundamental use cases for color in data visualizations:
    1. We can use color to distinguish groups of data from each other;
    2. We can use color to represent data values; and
    3. We can use color to highlight.

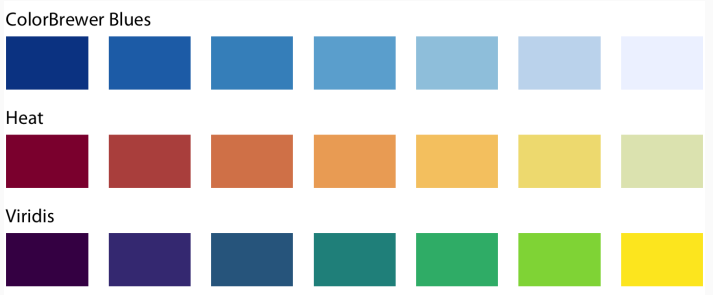
Colors

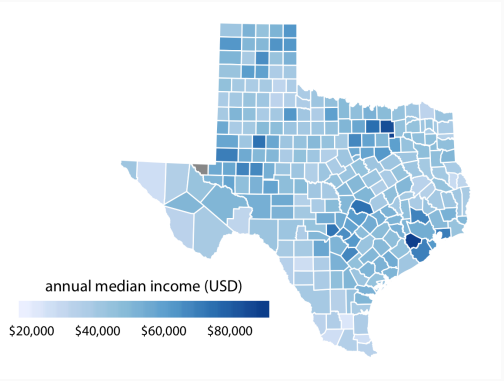
- There are three fundamental use cases for color in data visualizations:
  1. We can use color to **distinguish groups of data from each other**;
  2. We can use color to represent data values; and
  3. We can use color to highlight.

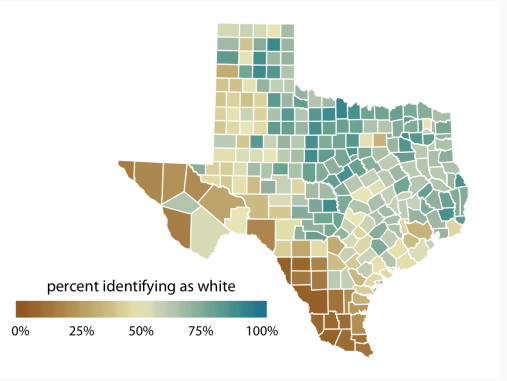


Colors

- There are three fundamental use cases for color in data visualizations:
  1. We can use color to distinguish groups of data from each other;
  2. We can use color to **represent data values**; and
  3. We can use color to highlight.

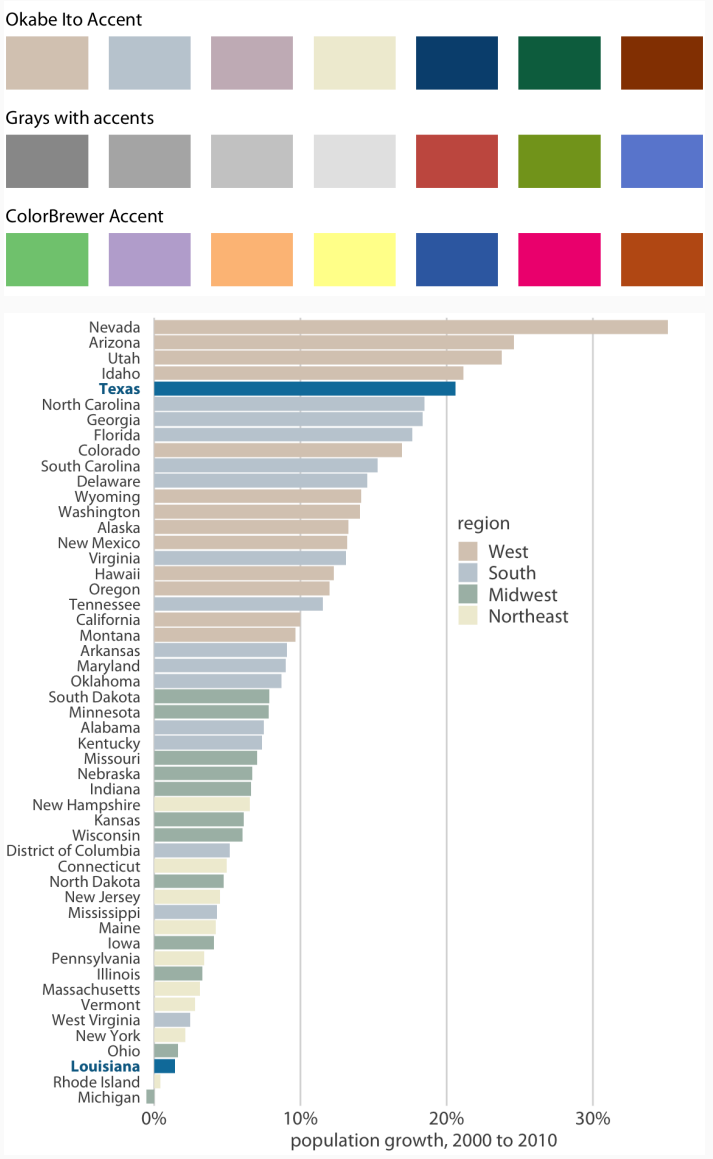






Colors

- There are three fundamental use cases for color in data visualizations:
  1. We can use color to distinguish groups of data from each other;
  2. We can use color to represent data values; and
  3. We can use color to **highlight**.

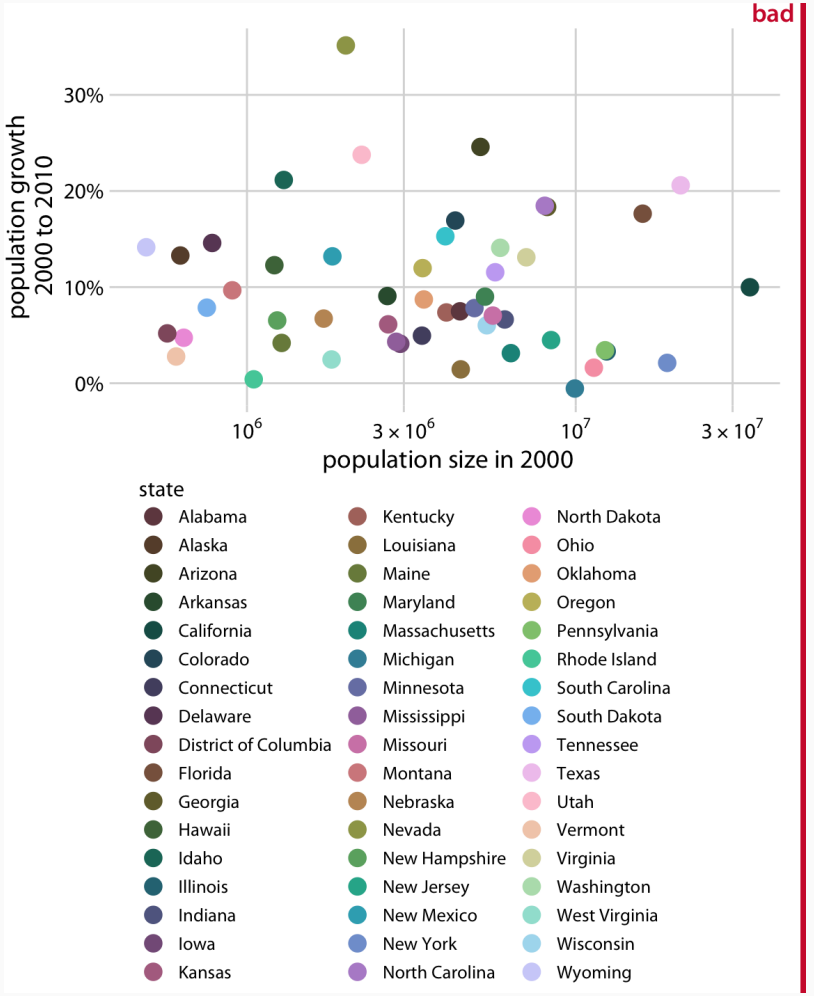


Colors

- There are three fundamental use cases for color in data visualizations:
  1. We can use color to distinguish groups of data from each other;
  2. We can use color to represent data values; and
  3. We can use color to highlight.
- While colors are very powerful aesthetics, try to avoid common pitfalls, such as:

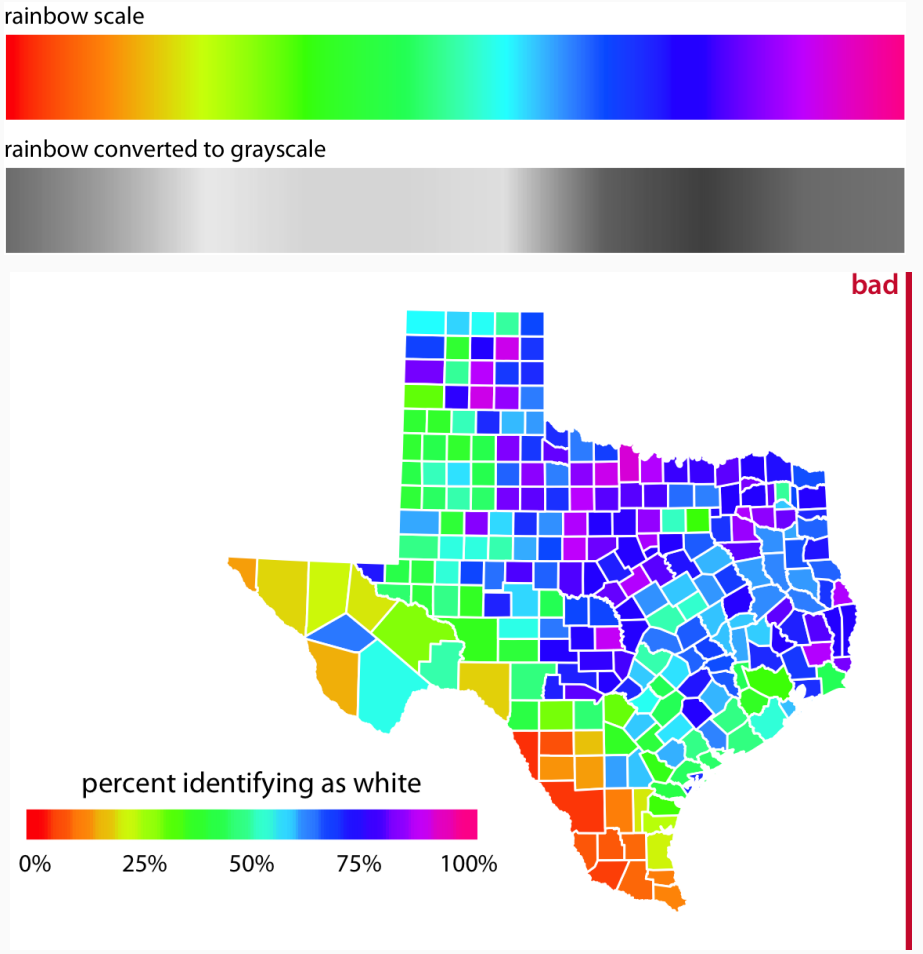
Colors

- There are three fundamental use cases for color in data visualizations:
  1. We can use color to distinguish groups of data from each other;
  2. We can use color to represent data values; and
  3. We can use color to highlight.
- While colors are very powerful aesthetics, try to avoid common pitfalls, such as:
  - Encoding **too much / irrelevant information**



Colors

- There are three fundamental use cases for color in data visualizations:
  1. We can use color to distinguish groups of data from each other;
  2. We can use color to represent data values; and
  3. We can use color to highlight.
- While colors are very powerful aesthetics, try to avoid common pitfalls, such as:
  - Encoding too much / irrelevant information
  - Using **non-monotonic color scales** to encode data values



Colors

- There are three fundamental use cases for color in data visualizations:
  1. We can use color to distinguish groups of data from each other;
  2. We can use color to represent data values; and
  3. We can use color to highlight.
- While colors are very powerful aesthetics, try to avoid common pitfalls, such as:
  - Encoding too much / irrelevant information
  - Using non-monotonic color scales to encode data values
  - Not designing for **color-vision deficiency**

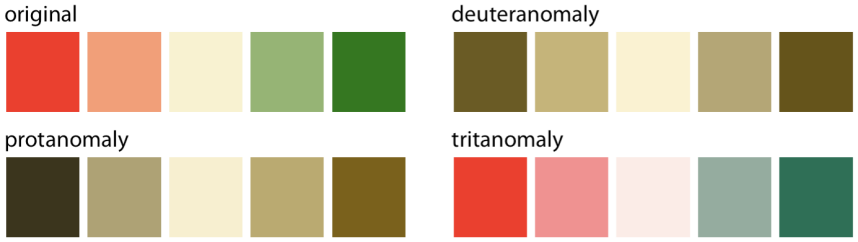


Figure 19.7: A red–green contrast becomes indistinguishable under red–green cvd (deuteranomaly or protanomaly).

Figure 19.8: A blue–green contrast becomes indistinguishable under blue–yellow cvd (tritanomaly).

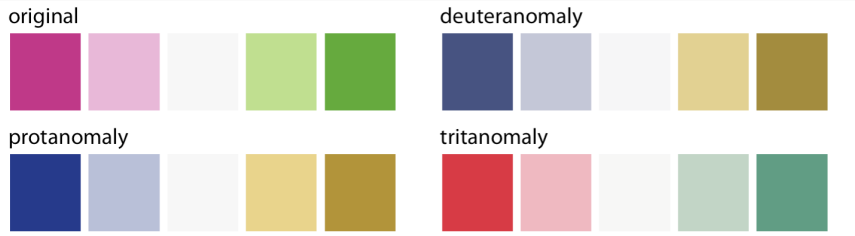


Figure 19.9: The ColorBrewer PiYG (pink to yellow-green) scale from Figure 4.5 looks like a red–green contrast to people with regular color vision but works for all forms of color-vision deficiency. It works because the reddish color is actually pink (a mix of red and blue) while the greenish color also contains yellow. The difference in the blue component between the two colors can be picked up even by deutans or protans, and the difference in the red component can be picked up by tritans.

Colors

- There are three fundamental use cases for color in data visualizations:
  1. We can use color to distinguish groups of data from each other;
  2. We can use color to represent data values; and
  3. We can use color to highlight.
- While colors are very powerful aesthetics, try to avoid common pitfalls, such as:
  - Encoding too much / irrelevant information
  - Using non-monotonic color scales to encode data values
  - Not designing for color-vision deficiency
- There's a whole science around the perception of colors in graphs, and a range of tools that help you select appropriate color schemes. My favorite is ColorBrewer, which is implemented in the `RColorBrewer` package.

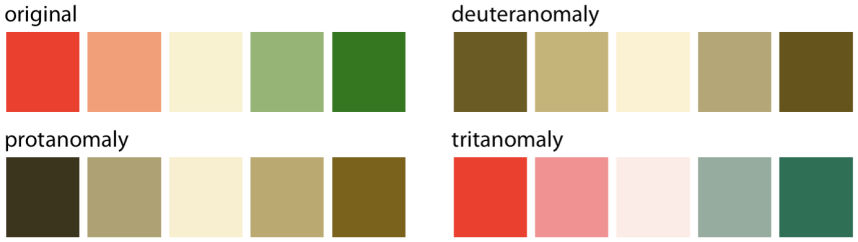


Figure 19.7: A red–green contrast becomes indistinguishable under red–green cvd (deuteranomaly or protanomaly).

Figure 19.8: A blue–green contrast becomes indistinguishable under blue–yellow cvd (tritanomaly).

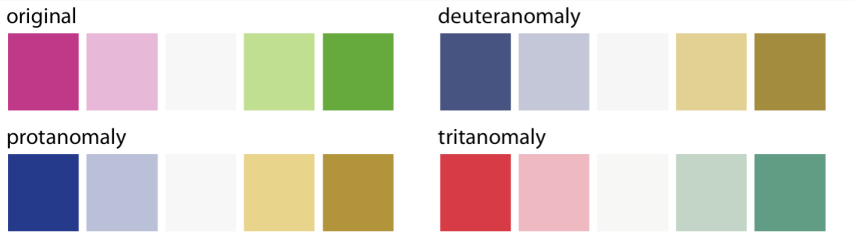


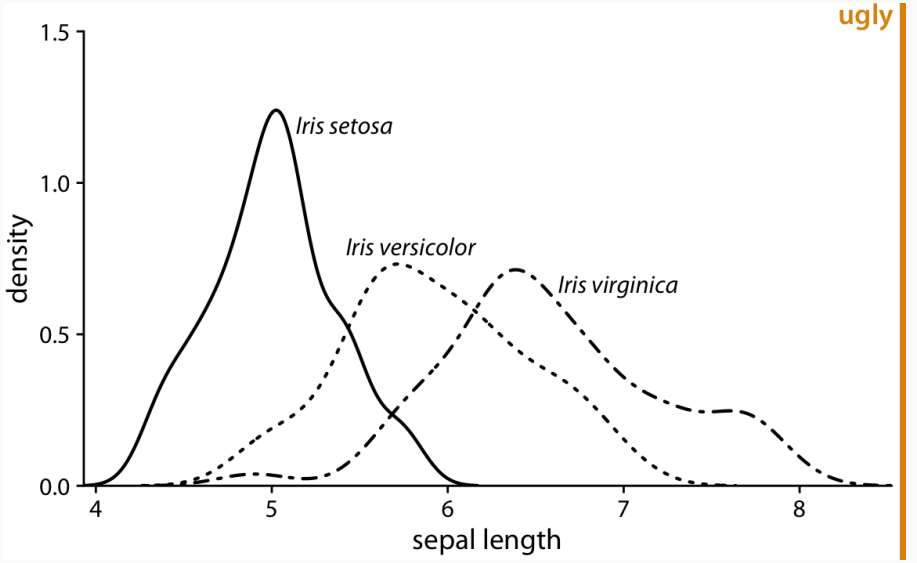
Figure 19.9: The ColorBrewer PiYG (pink to yellow-green) scale from Figure 4.5 looks like a red–green contrast to people with regular color vision but works for all forms of color-vision deficiency. It works because the reddish color is actually pink (a mix of red and blue) while the greenish color also contains yellow. The difference in the blue component between the two colors can be picked up even by deutans or protans, and the difference in the red component can be picked up by tritans.

Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.

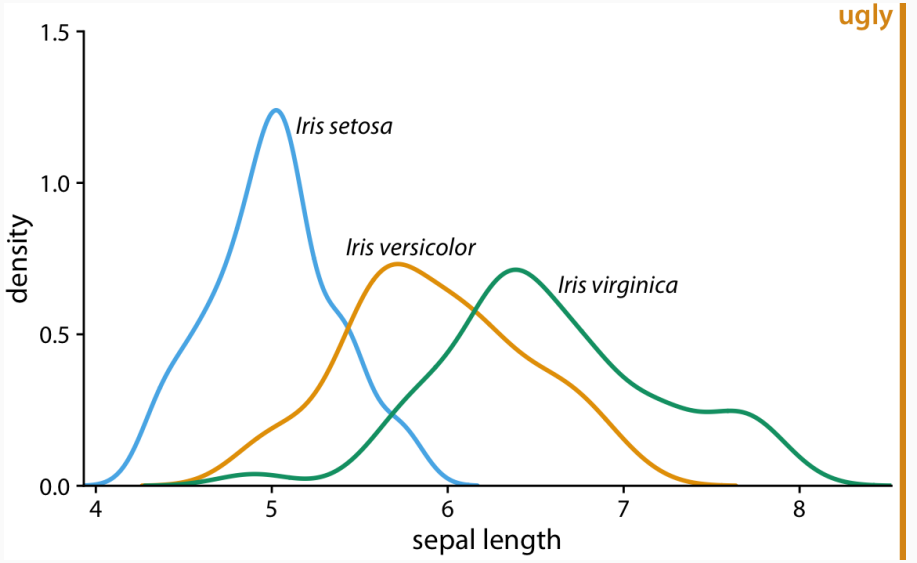
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider colored shapes instead of **different lines**.



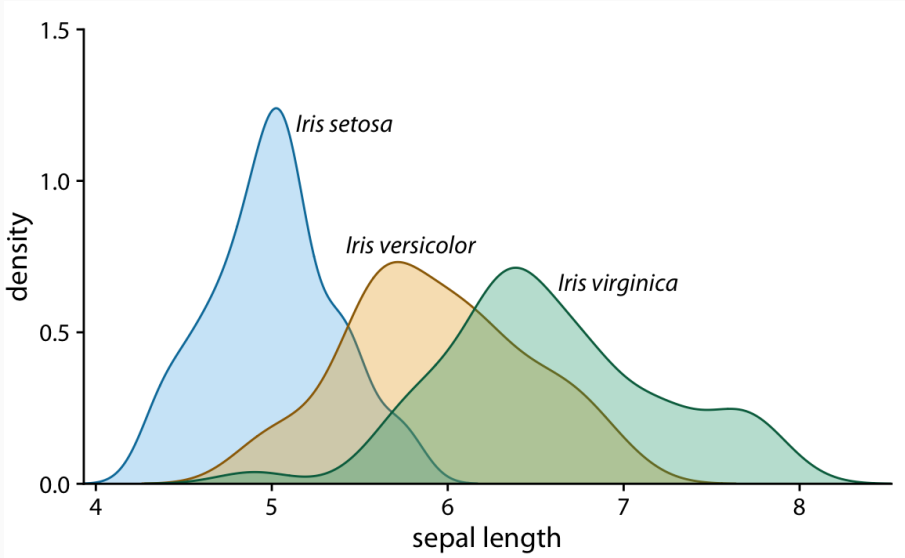
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider **colored shapes** instead of different lines.



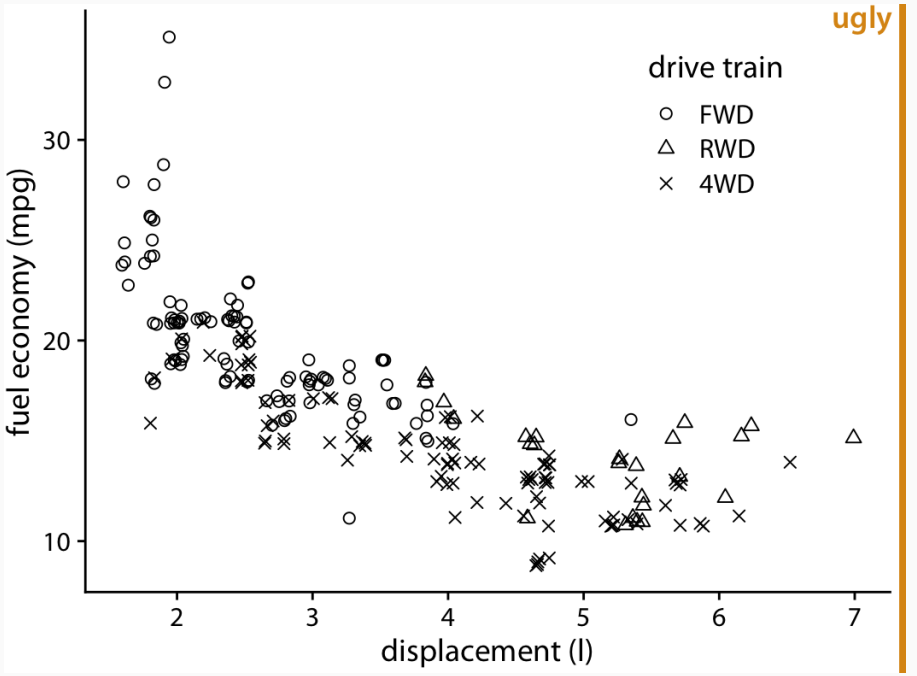
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider **colored shapes** instead of different lines.



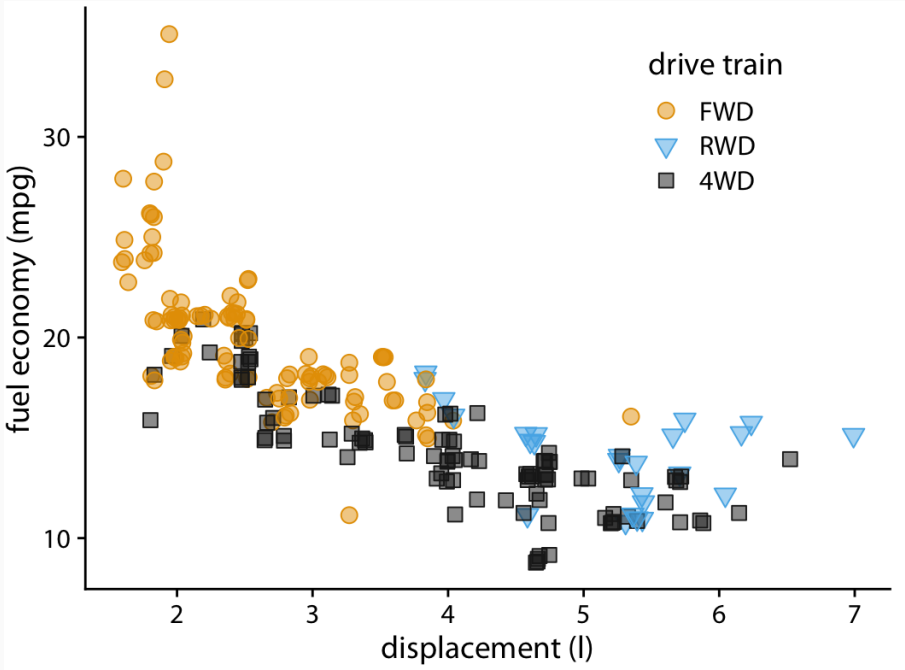
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider colored shapes instead of different lines.
- Consider colored shapes instead of **different points**.



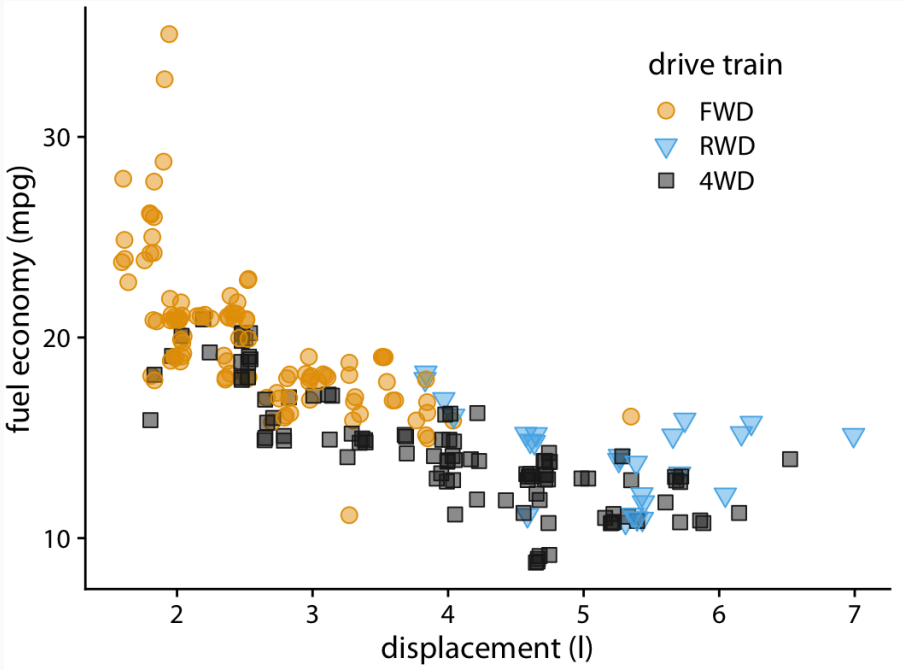
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider colored shapes instead of different lines.
- Consider **colored shapes** instead of different points.



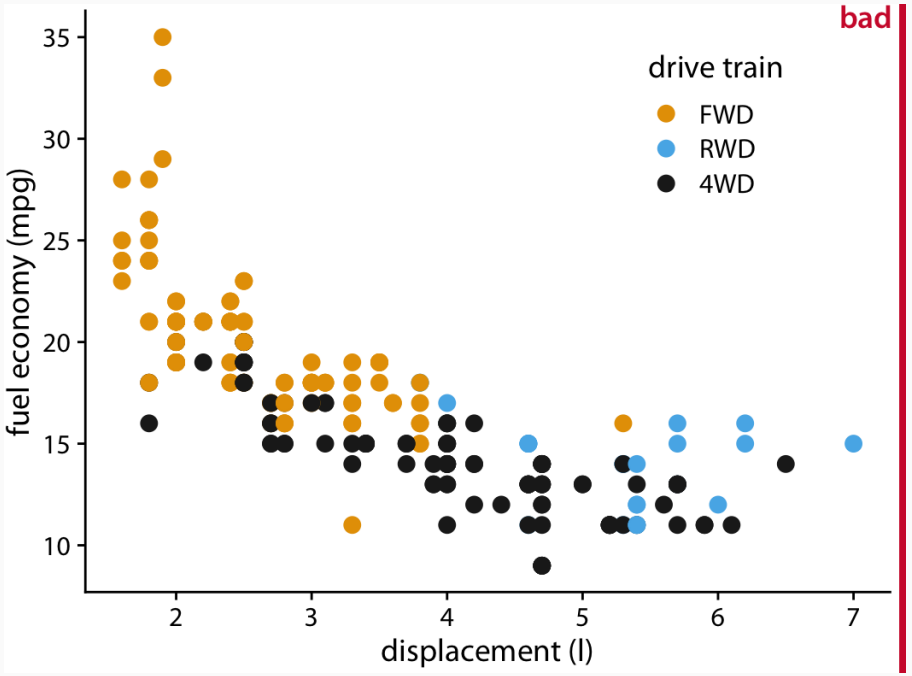
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider colored shapes instead of different lines.
- Consider colored shapes instead of different points.
- Use **redundant coding**, i.e. use color to enhance the visual appearance of the figure without relying entirely on color to convey key information.



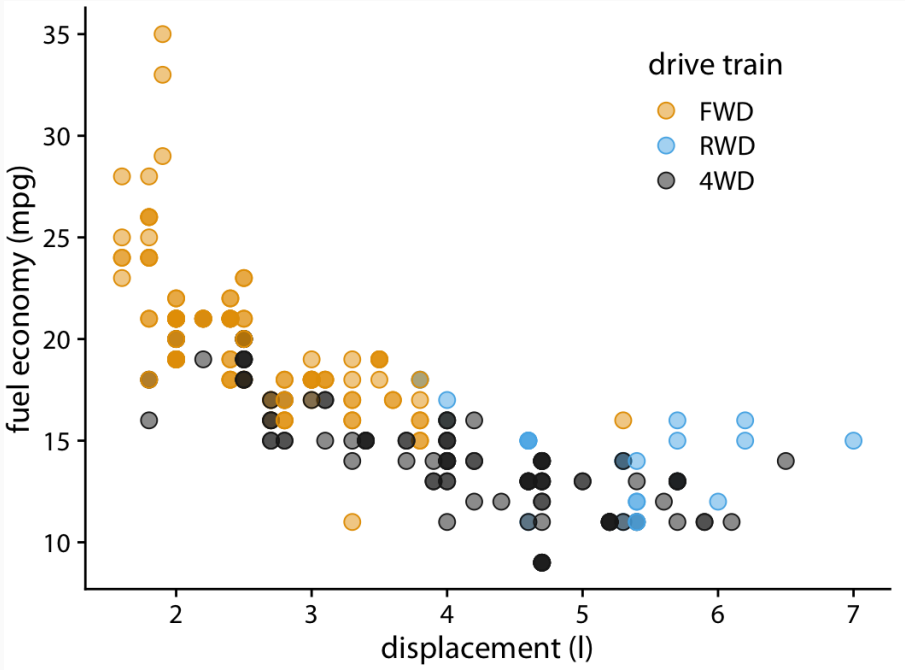
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider colored shapes instead of different lines.
- Consider colored shapes instead of different points.
- Use redundant coding, i.e. use color to enhance the visual appearance of the figure without relying entirely on color to convey key information.
- To tackle **overlapping data**, use partial transparency (alpha blending) and (moderate) jittering.



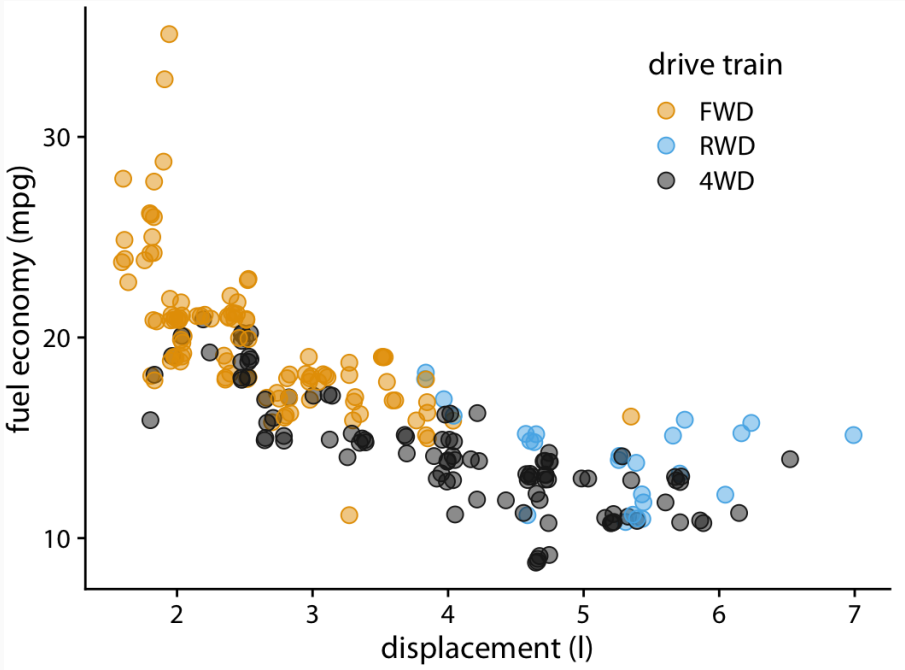
Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider colored shapes instead of different lines.
- Consider colored shapes instead of different points.
- Use redundant coding, i.e. use color to enhance the visual appearance of the figure without relying entirely on color to convey key information.
- To tackle overlapping data, use **partial transparency** (alpha blending) and (moderate) jittering.



Line and point types

- Different line and point types can help distinguish different data types (e.g., subgroups).
- For lines, we can, use solid, dashed or dotted formatting.
- For points, we can use solid dots, open circles, triangles, or really any symbol we can come up with.
- Try to avoid different line and point types. They add a lot of noise and are difficult to read.
- Consider colored shapes instead of different lines.
- Consider colored shapes instead of different points.
- Use redundant coding, i.e. use color to enhance the visual appearance of the figure without relying entirely on color to convey key information.
- To tackle overlapping data, use partial transparency (alpha blending) and **(moderate) jittering**.



Principles of good data visualization

On the shoulder of giants...

"Visualization is surprisingly difficult. Even the most simple matters can easily go wrong."

"No matter how clever the choice of the information, and no matter how technologically impressive the encoding, a visualization fails if the decoding fails."

William Cleveland

On the shoulder of giants...

"Visualization is surprisingly difficult. Even the most simple matters can easily go wrong."

"No matter how clever the choice of the information, and no matter how technologically impressive the encoding, a visualization fails if the decoding fails."

William Cleveland

"Although nothing can replace a good graphical idea applied to an interesting set of numbers, editing and revision are as essential to sound graphical work as they are to writing."

"Design cannot rescue failed content."

"Above all else show the data."

Edward Tufte

Fundamental principles of analytic design

In Chapter 5 of his book "Beautiful Evidence", Edward Tufte outlines six fundamental principles of analytic design:

1. **Comparisons.** Show comparisons, contrasts, differences.
2. **Causality, mechanism, structure, explanation.** Show causality, mechanism, explanation, systematic structure.
3. **Multivariate analysis.** Show multivariate data; that is show more than 1 or 2 variables.
4. **Integration of evidence.** Completely integrate words, numbers, images, diagrams.
5. **Documentation.** Thoroughly describe the evidence. Provide a detailed title, indicate the authors and sponsors, document the data sources, show complete measurement scales, point out relevant issues.
6. **Content counts most of all.** Analytical presentations ultimately stand or fall depending on the quality, relevance and integrity of their content ("What is the problem you want to solve?")

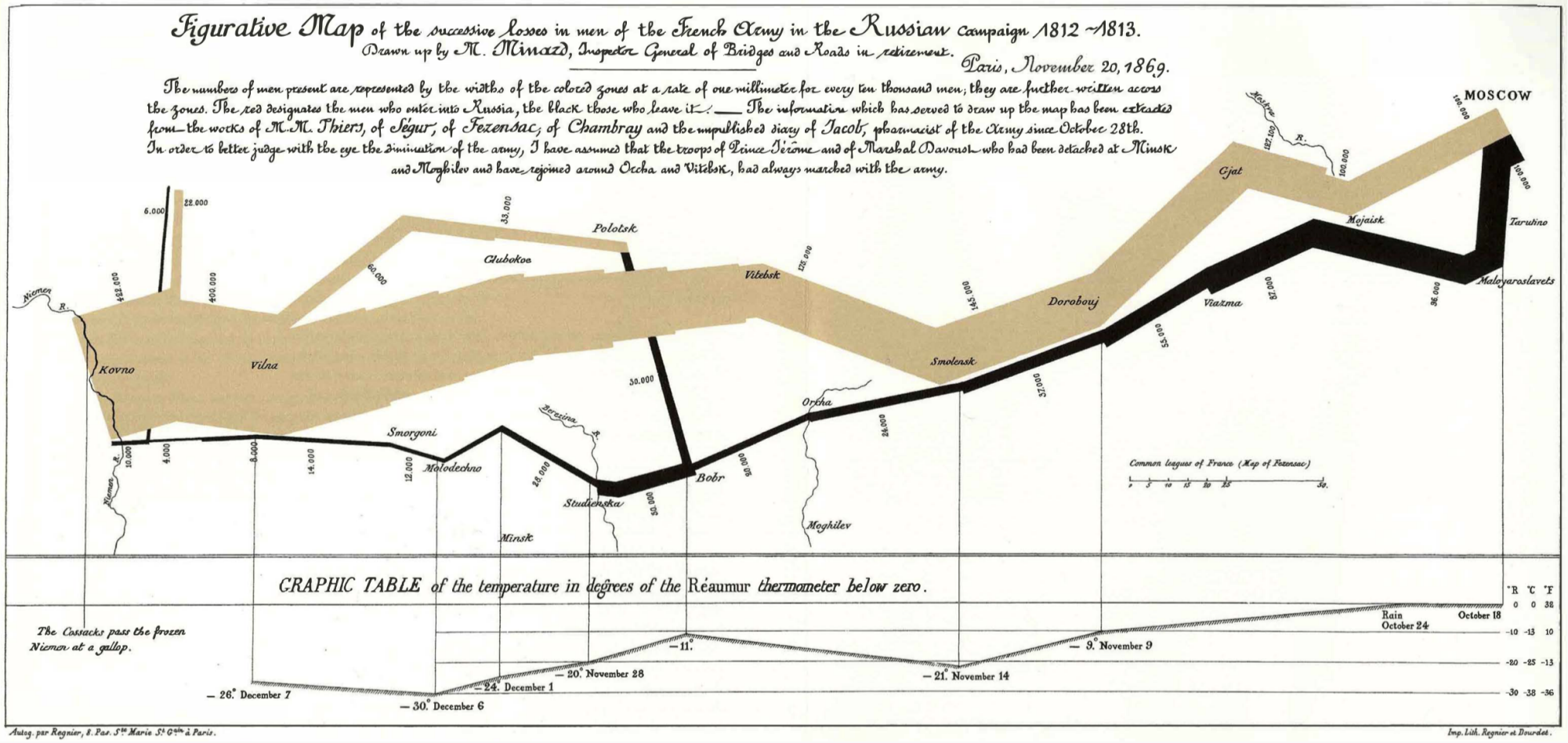
Fundamental principles of analytic design

In Chapter 5 of his book "Beautiful Evidence", Edward Tufte outlines six fundamental principles of analytic design:

1. **Comparisons.** Show comparisons, contrasts, differences.
2. **Causality, mechanism, structure, explanation.** Show causality, mechanism, explanation, systematic structure.
3. **Multivariate analysis.** Show multivariate data; that is show more than 1 or 2 variables.
4. **Integration of evidence.** Completely integrate words, numbers, images, diagrams.
5. **Documentation.** Thoroughly describe the evidence. Provide a detailed title, indicate the authors and sponsors, document the data sources, show complete measurement scales, point out relevant issues.
6. **Content counts most of all.** Analytical presentations ultimately stand or fall depending on the quality, relevance and integrity of their content ("What is the problem you want to solve?")

On the following slide you'll see Minard's famous map of Napoleon's March, praised by Tufte as "one of the best statistical graphs ever". Can you spot how those principles were implemented in Minard's Map? More information on the map here.

Minard's map of Napoleon's March



Dos and Don'ts

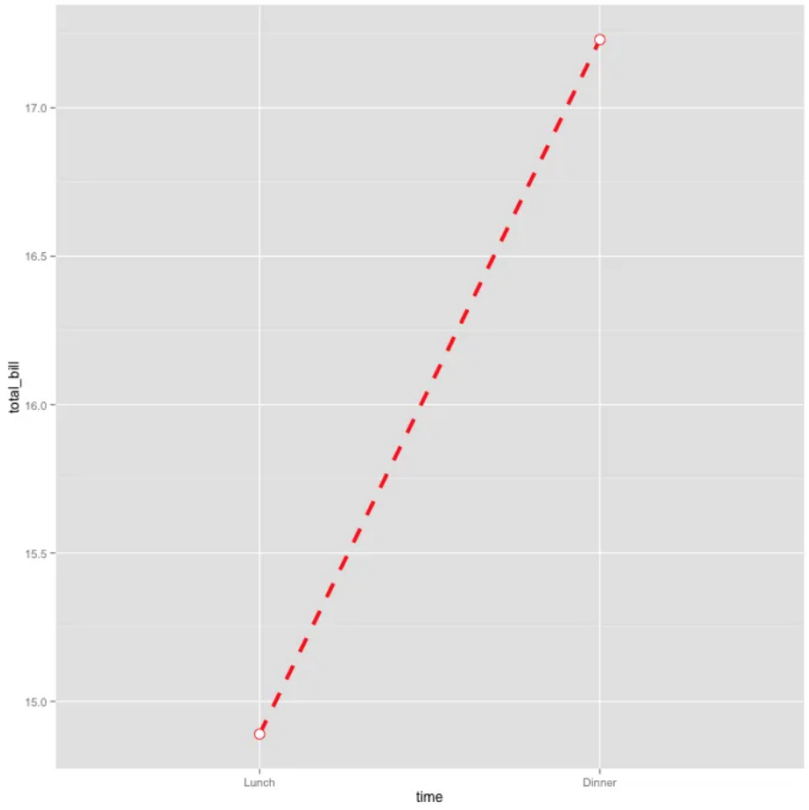
1. Follow the principle of proportional ink.[1]
2. Maximize the data–ink ratio, within reason.
3. Avoid invisible overplotting.
4. Drop all the unimportant stuff.
5. Don't overload graphs. Instead, use several.
6. Use color scales that match the logic of the data scale.
7. Use color-vision deficient-friendly colors.
8. Only use a legend when you need one.
9. Pay attention to legend order.
10. Label axes properly (but avoid trivial information).
11. Use grids and helper lines, within reason.
12. Don't order alphabetically ("Alabama first"). Use natural orders instead.
13. Bar chart axes should include zero.
14. Put the explanatory variable on the x axis, the outcome on the y axis.
15. Axes have canonical directions. Larger values are placed above/right of smaller values.
16. Avoid multiple y axes at all cost.
17. Don't do pie charts. (Or maybe do?)
18. Don't go 3D.
19. Use readable fonts and font sizes.
20. Sometimes a table is just enough.

[1]Follow the links for more information.

Visualization with R

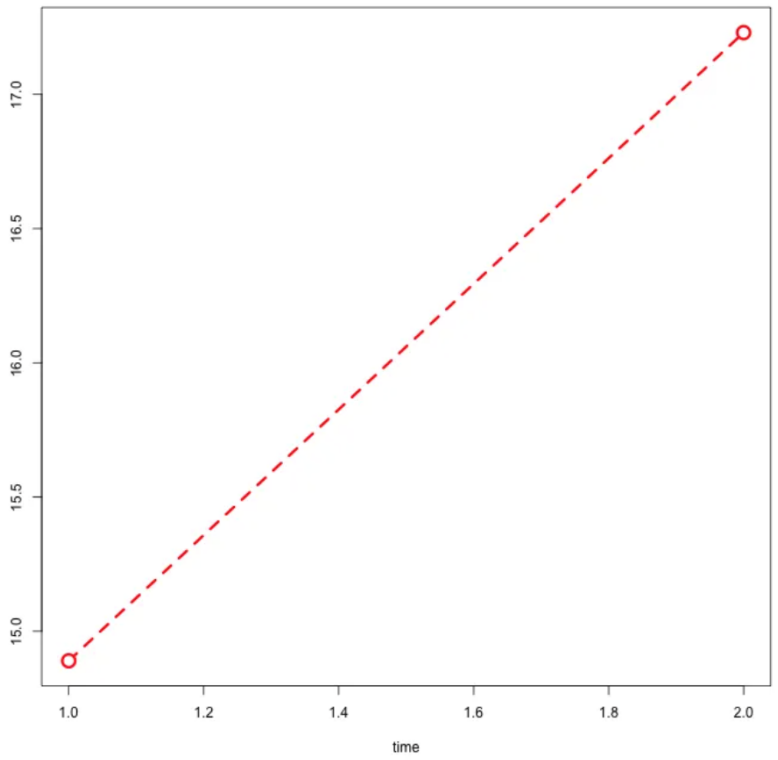
R base graphics vs. ggplot2

ggplot2



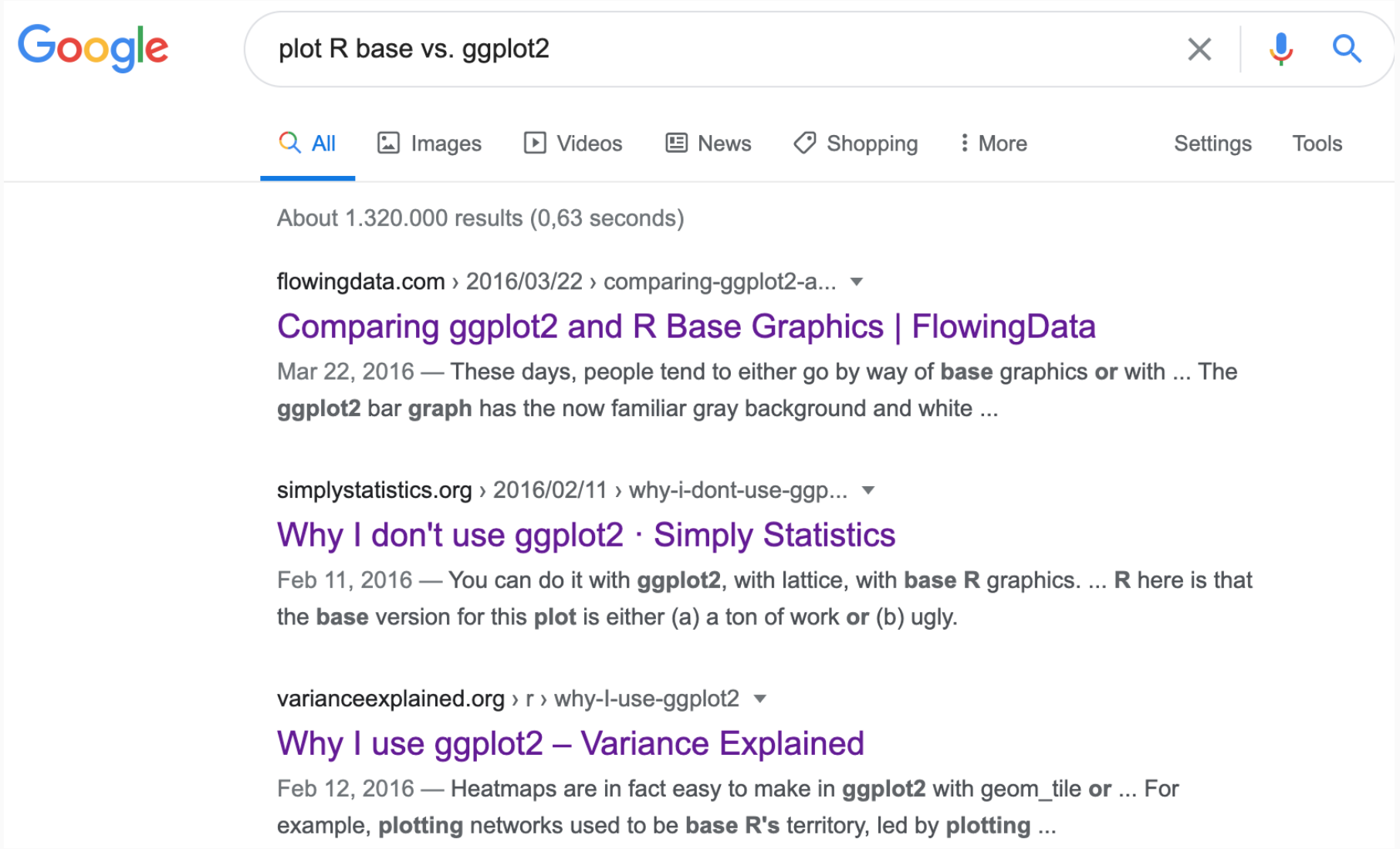

```
ggplot(data=dat, aes(x=time, y=total_bill, group=1)) +
  geom_line(colour="red", linetype="dashed", size=1.5) +
  geom_point(colour="red", size=4, shape=21, fill="white")
```

Base Graphics




```
plot(c(1,2), dat$total_bill, type="l", xlab="time", ylab="",
    lty=2, lwd=3, col="red")
points(c(1,2), dat$total_bill, pch=21, col="red", cex=2,
    bg="white", lwd=3)
```

R base graphics vs. ggplot2 (cont.)

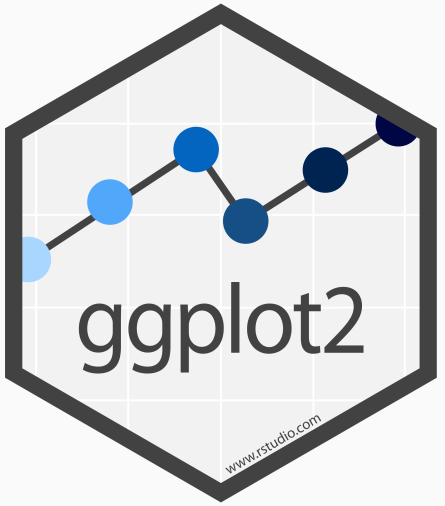


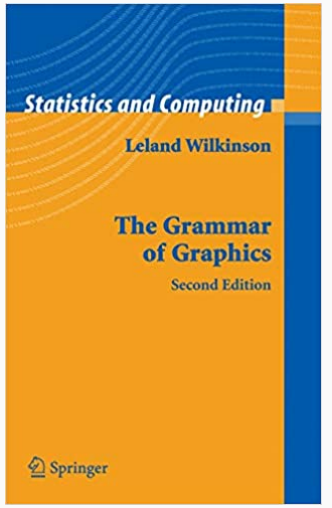
Plotting things in R with ggplot2

`ggplot2` is a system for declaratively creating graphics, based on *The Grammar of Graphics* (Leland Wilkinson). You provide the data, tell `ggplot2` how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details. It is part of the tidyverse.[1]

There are **many** key `ggplot2` verbs that you need to learn, including:

1. `ggplot()`: Begins a plot that you finish by adding layers to.
2. `aes()`: Specify aesthetic mappings that describe how variables are mapped to visuals (e.g., x and y).
3. `geom_*()`: Specify geoms to represent data points in a layer, e.g., as points or lines.





[1] This is R, so you already know that there are multiple ways to do it, and all suck. We will stay in the tidyverse and ignore R base graphics (`plot()`, `barplot()`, `boxplot()`, `hist()`, etc.) for now.

The grammar of graphics

The easy logic of graphical grammar

- Coding a graphic requires us to describe each element of it very precisely.
- To do so, `ggplot` and its creator Hadley Wickham build on and extend a particular grammar - the "Grammar of Graphics" by Wilkinson, Anand, and Grossmann.
- Building plots with `ggplot2` emphasizes the **concept of layers**, which are specified function by function and assembled with `+`.
- With this particular grammar, we talk less about chart *types* and more about specific chart *elements*.
- For instance, we don't say:

  "R, give me a small multiple scatter plot."
- Instead, we say:

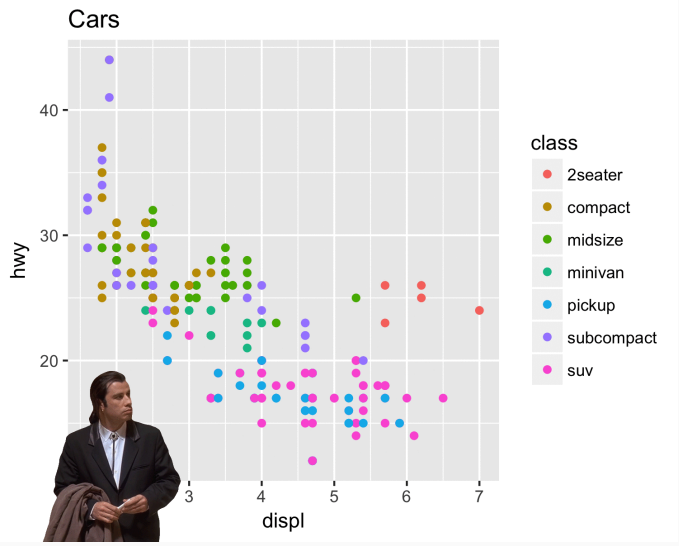
  "Using this dataset, map wealth to the x-axis, health to the y-axis, add points, color by continent, size by population, scale the y-axis with a log, and facet by year."

Grammar components as layers in ggplot2

The grammar of graphics (cont.)

The hard practice of applying graphical grammar

- Building basic plots is fairly straightforward with `ggplot2`.
- There are two aspects that make things complicated:
  1. `ggplot` expects tidy (long) data. That's not always the data structure you have.
  2. It's a comprehensive grammar. All the `ggplot` verbs, arguments and defaults are difficult to learn.
- Things become more complicated (but also exciting) with a multiverse of additional packages that bring additional geometrics, themes, aesthetics, and much more. Check out this overview to learn more.
- We will only touch the basics here. You will learn how to do the first steps in the lab. The rest is learning by doing.



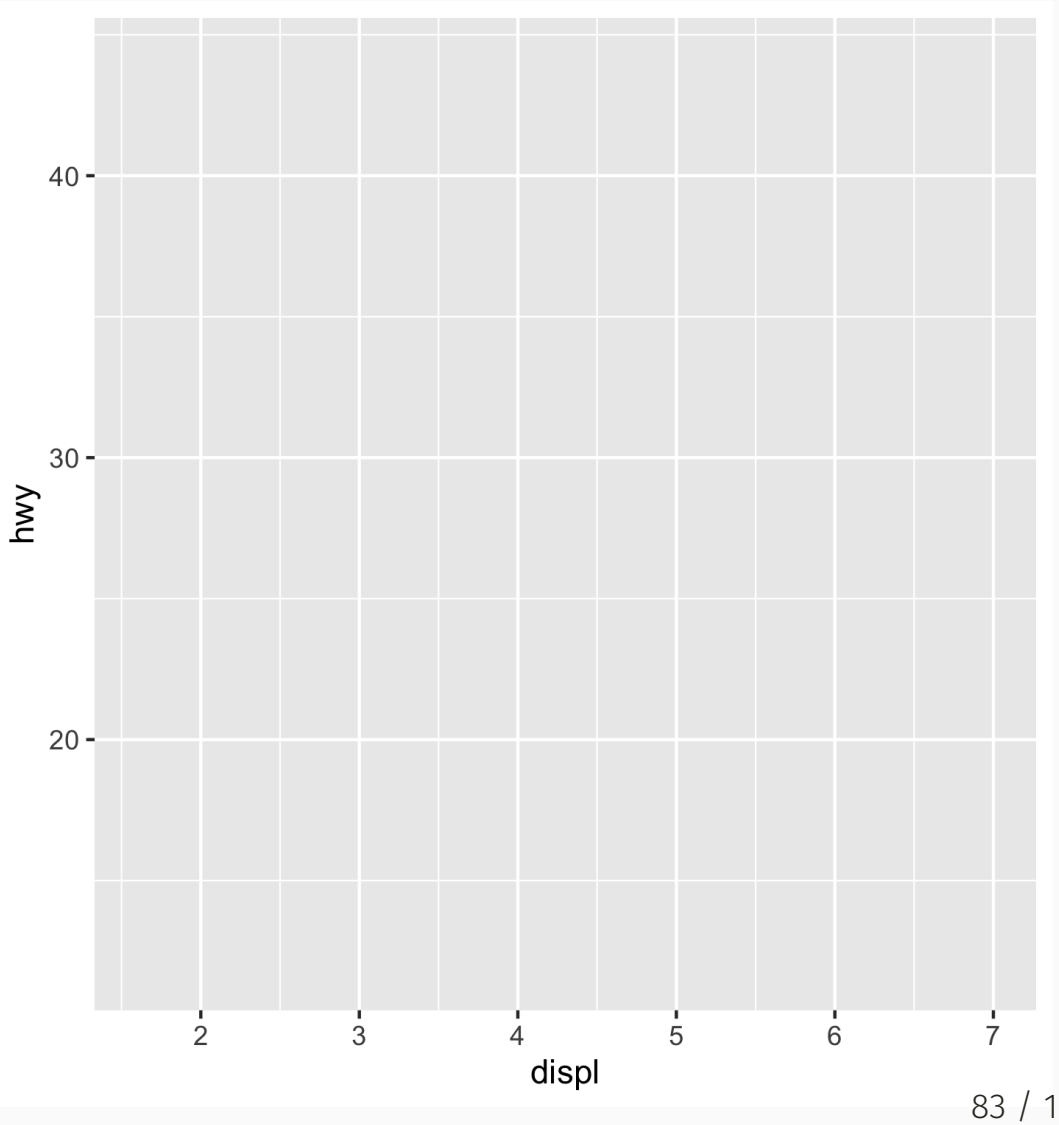
Components of grammar of graphics

Element	Description	Essential?	Example
Data	This is the dataset being plotted containing the variables to be plotted on the graph.	Yes	`ggplot(data = mpg)`
Aesthetics	Aesthetics refers to the scales on which we map our data. Some common aesthetics to consider are axis (x,y), shape, size and color.	Yes	`aes(x = displ, y = hwy, color = drv)`
Geometries	Geom refers to the actual visual elements used for the data in the plot, such as points, lines, and bars.	Yes	`geom_point()`
Scales	Scales map data values to the visual values of an aesthetic. This can be used to change a default mapping	No	`scale_colour_manual(values = c("red", "blue", "green"))`
Facets	Faceting refers to splitting the data into multiple subsets and then displaying plots for the specific subsets in a panel (small multiples).	No	`facet_grid(vars(drv), ncol = 1)`
Statistics	This refers to representing statistical information about the data, such as mean and variance, to help in understanding the data.	No	`stat_count(geom="bar")`
Coordinates	This refers to the space on which the data is plotted (e.g., Cartesian coordinates).	No	`coord_polar()`
Labels	This refers to additional descriptions of your plot, such as title, subtitle, caption, x and y axis label	No	`labs(x = "Height", y = "Weight", title = "Look at my plot")`
Themes	Themes are used to change the appearance of non-data elements, such as fonts, color, or legends.	No	`theme_bw()`

Building a plot step by step with ggplot2

Start with data and aesthetics[1]

```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv))
```

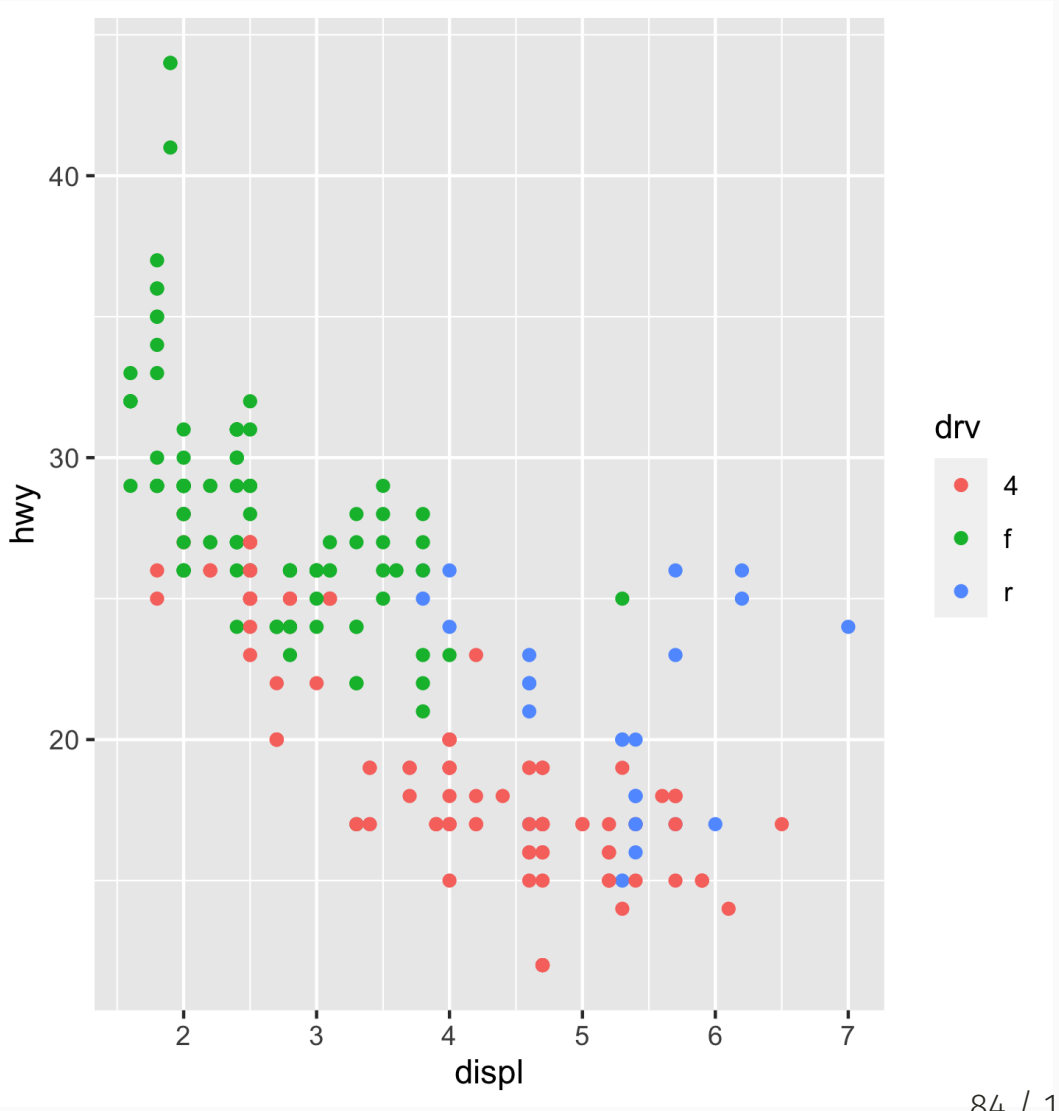


[1]This example is borrowed from Andrew Heiss.

Building a plot step by step with ggplot2 (cont.)

Add a point geom

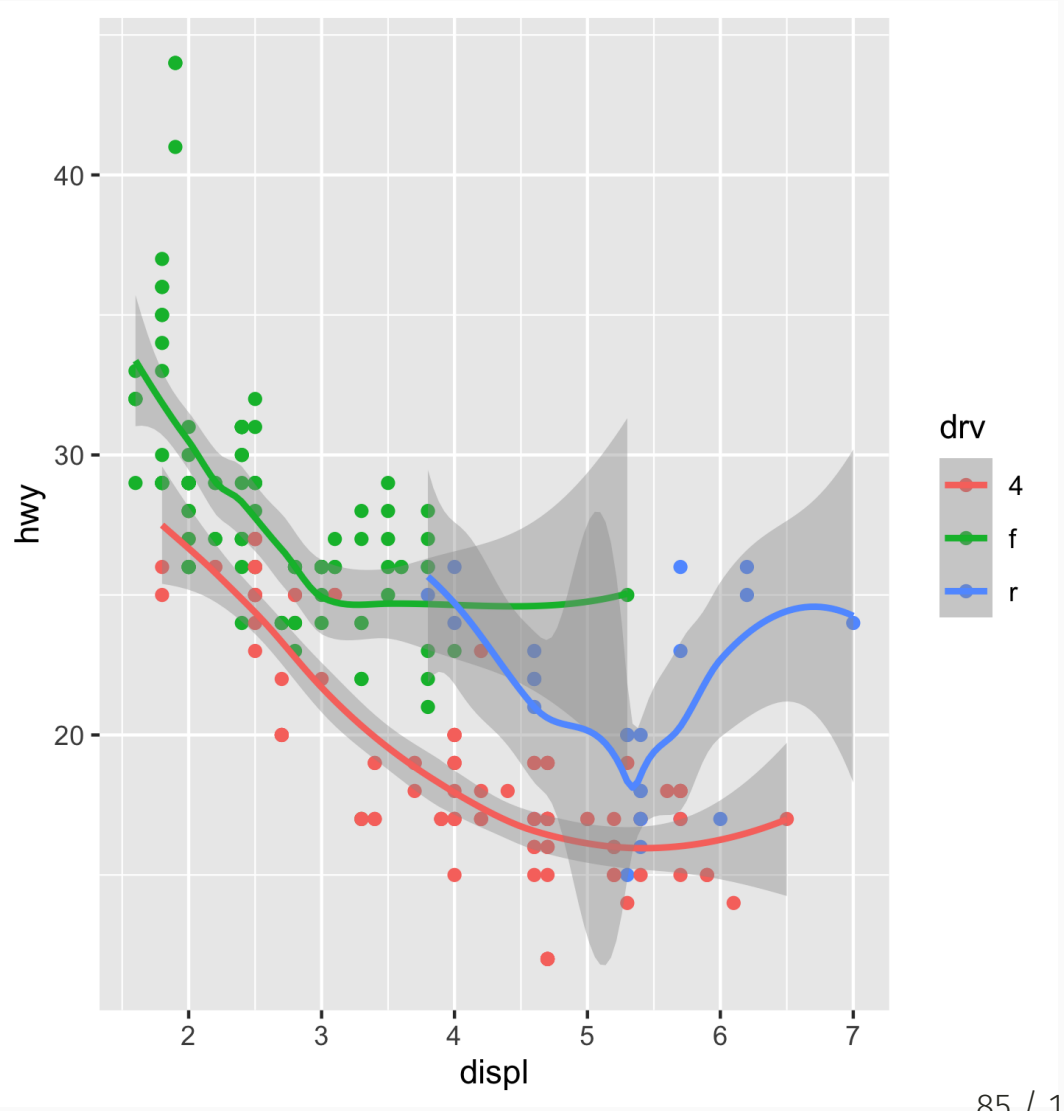
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+   geom_point()
```



Building a plot step by step with ggplot2 (cont.)

Add a smooth geom

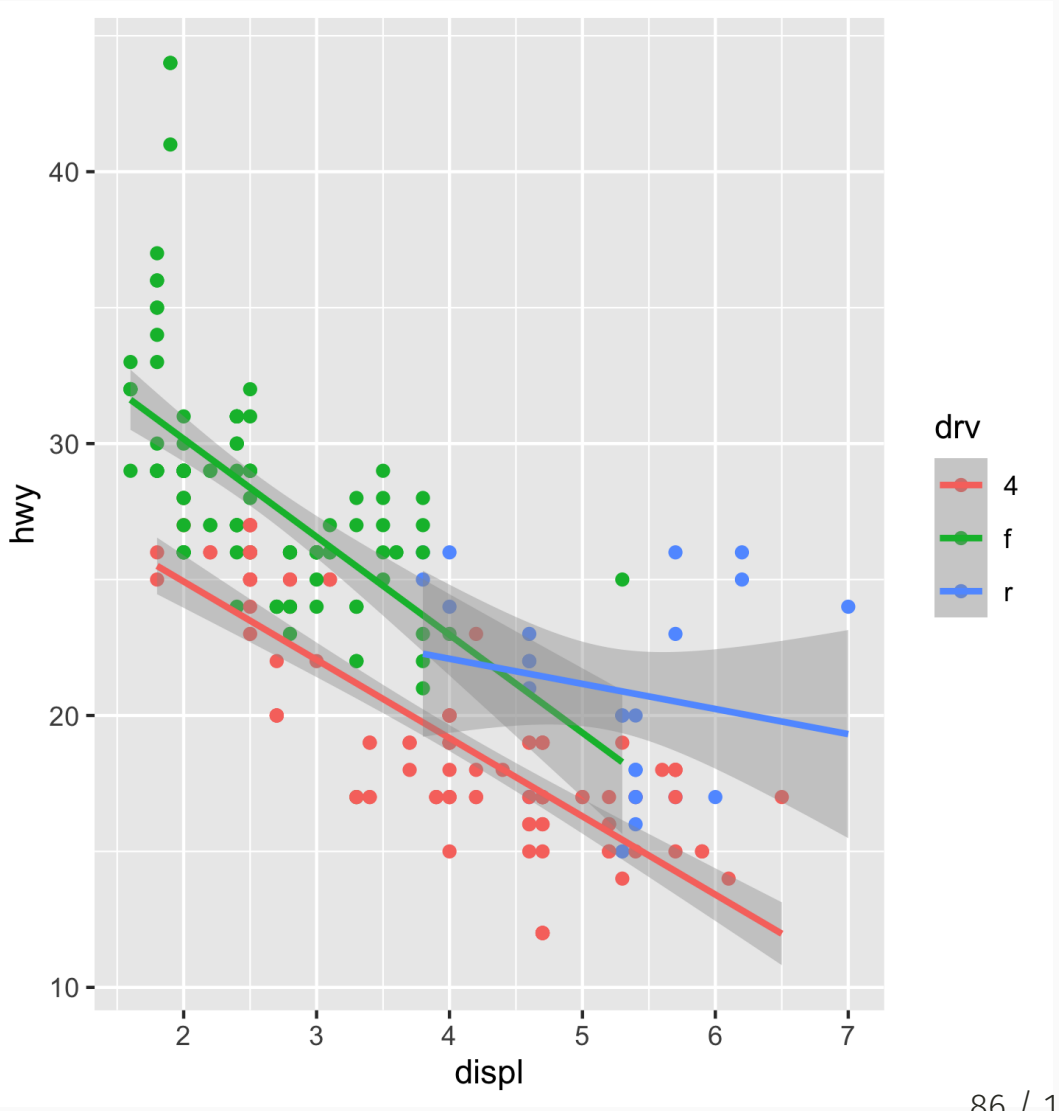
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth()
```



Building a plot step by step with ggplot2 (cont.)

Make it straight

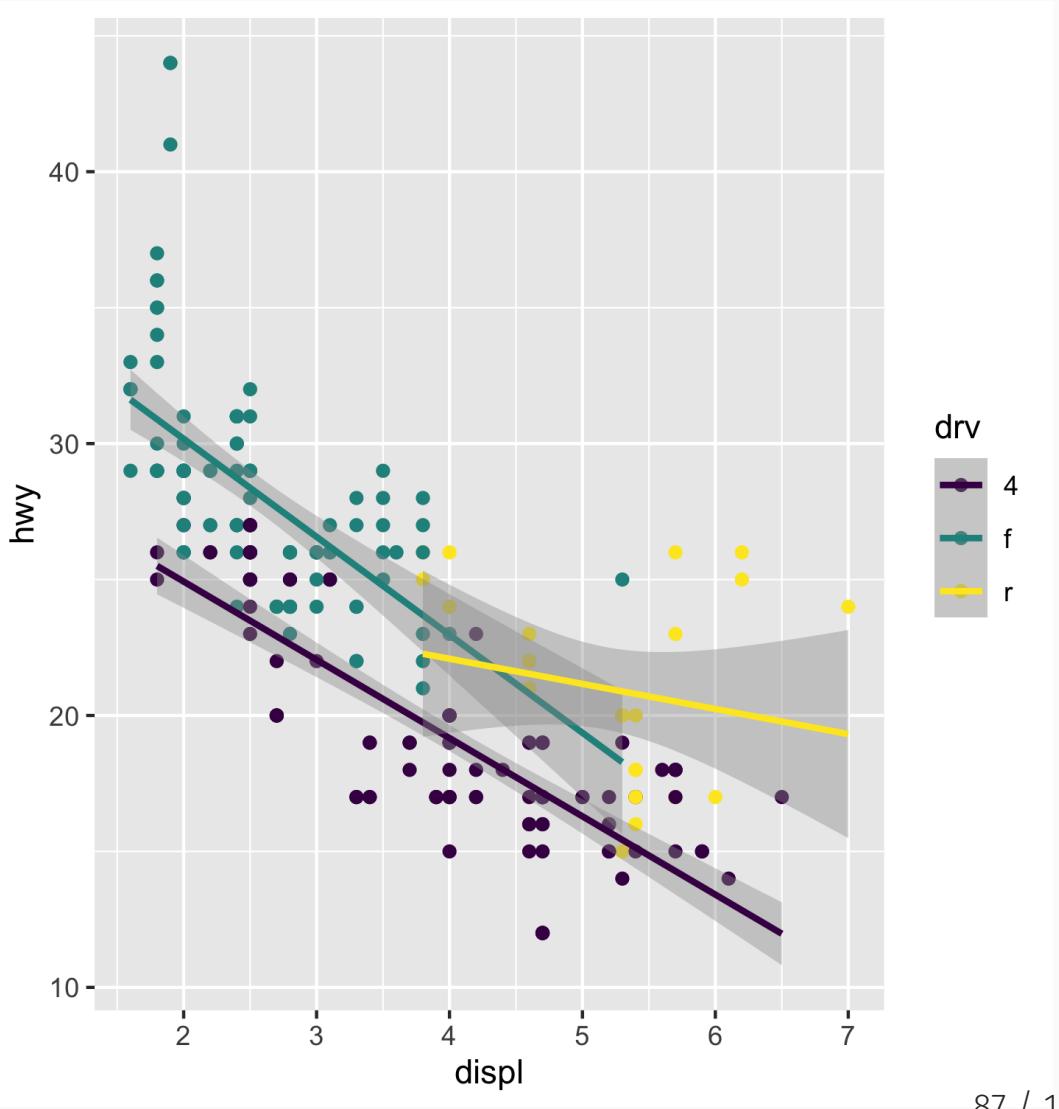
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth(method = "lm")
```



Building a plot step by step with ggplot2 (cont.)

Use a viridis color scale

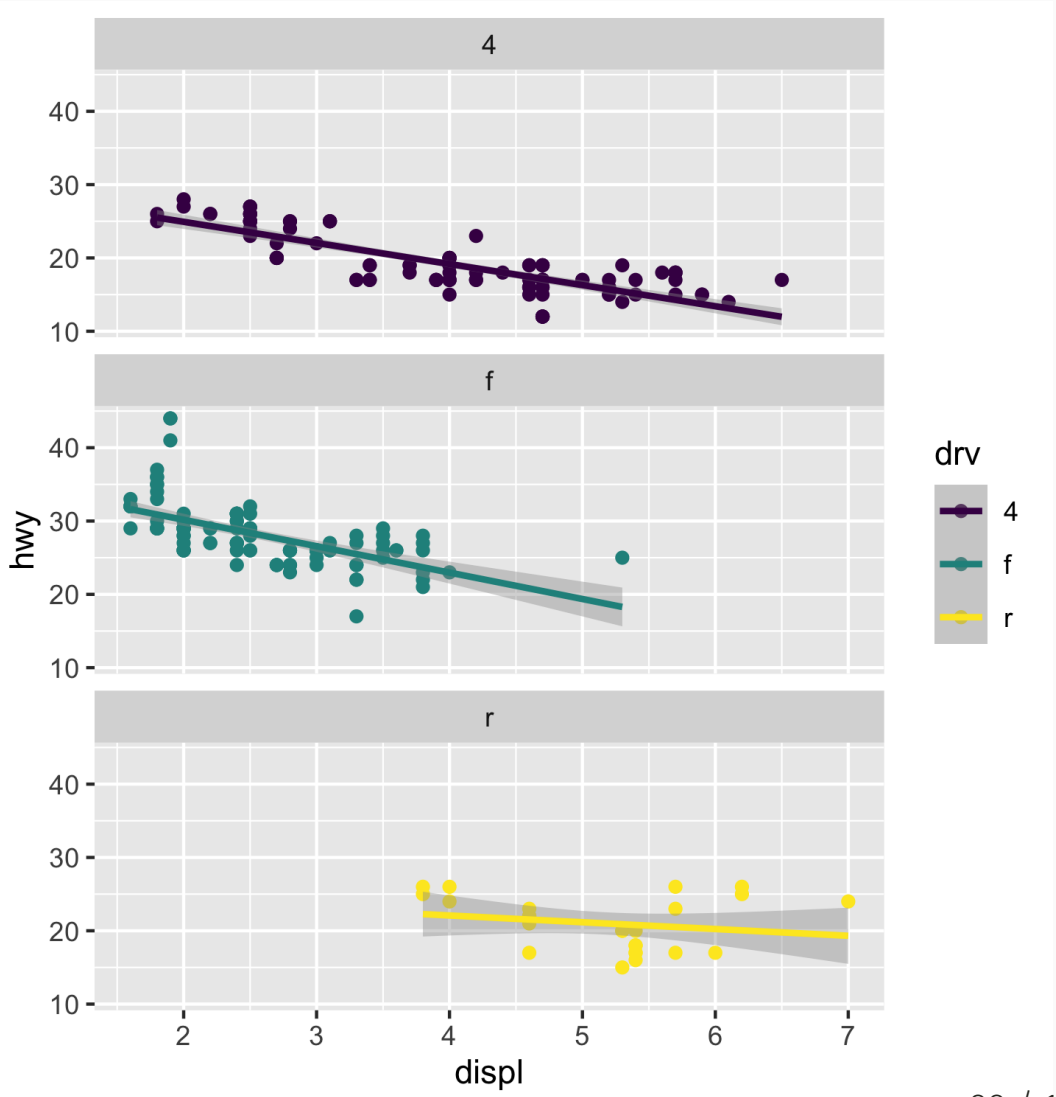
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth(method = "lm") +
+    scale_color_viridis_d()
```



Building a plot step by step with ggplot2 (cont.)

Facet by drive

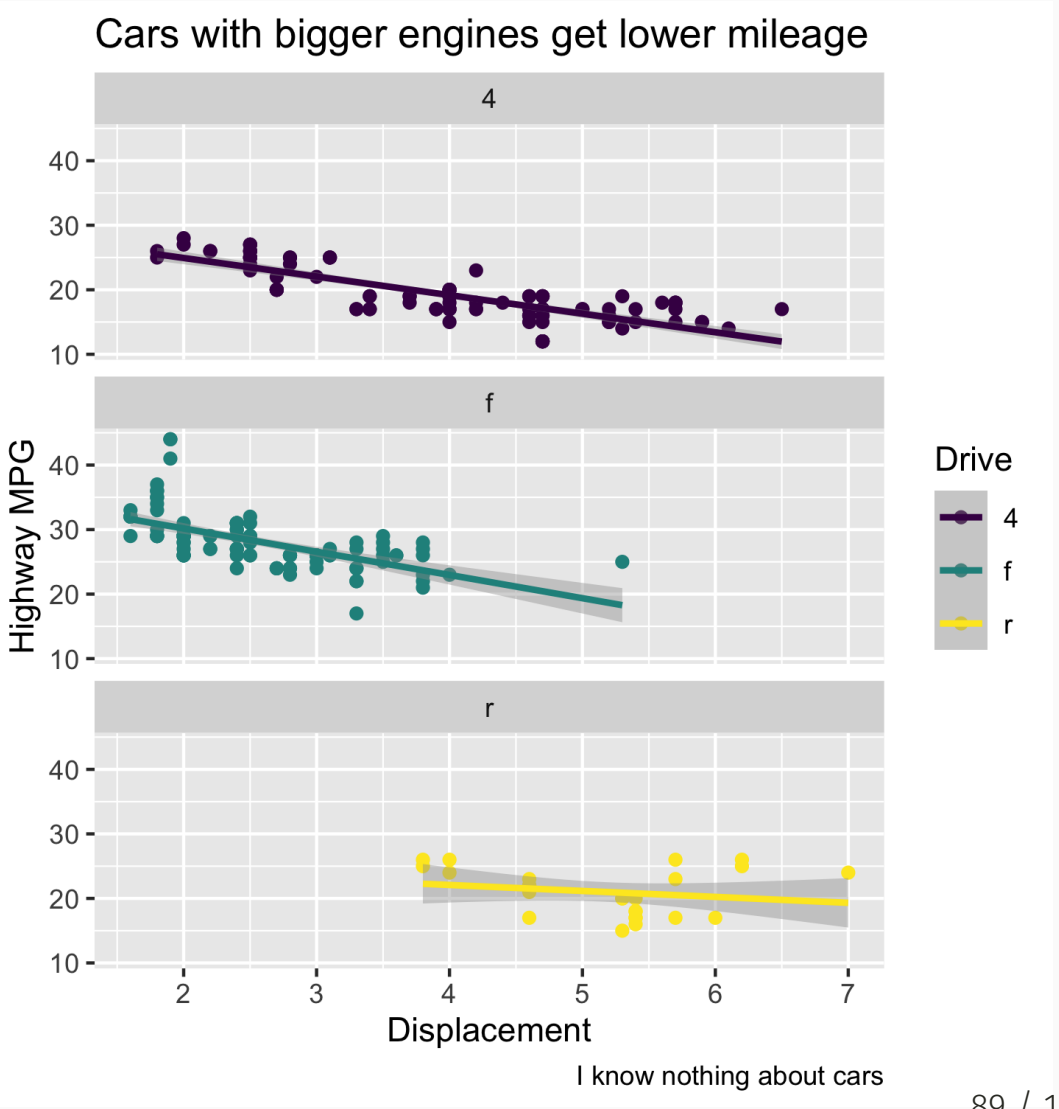
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth(method = "lm") +
+    scale_color_viridis_d() +
+    facet_wrap(vars(drv), ncol = 1)
```



Building a plot step by step with ggplot2 (cont.)

Add labels

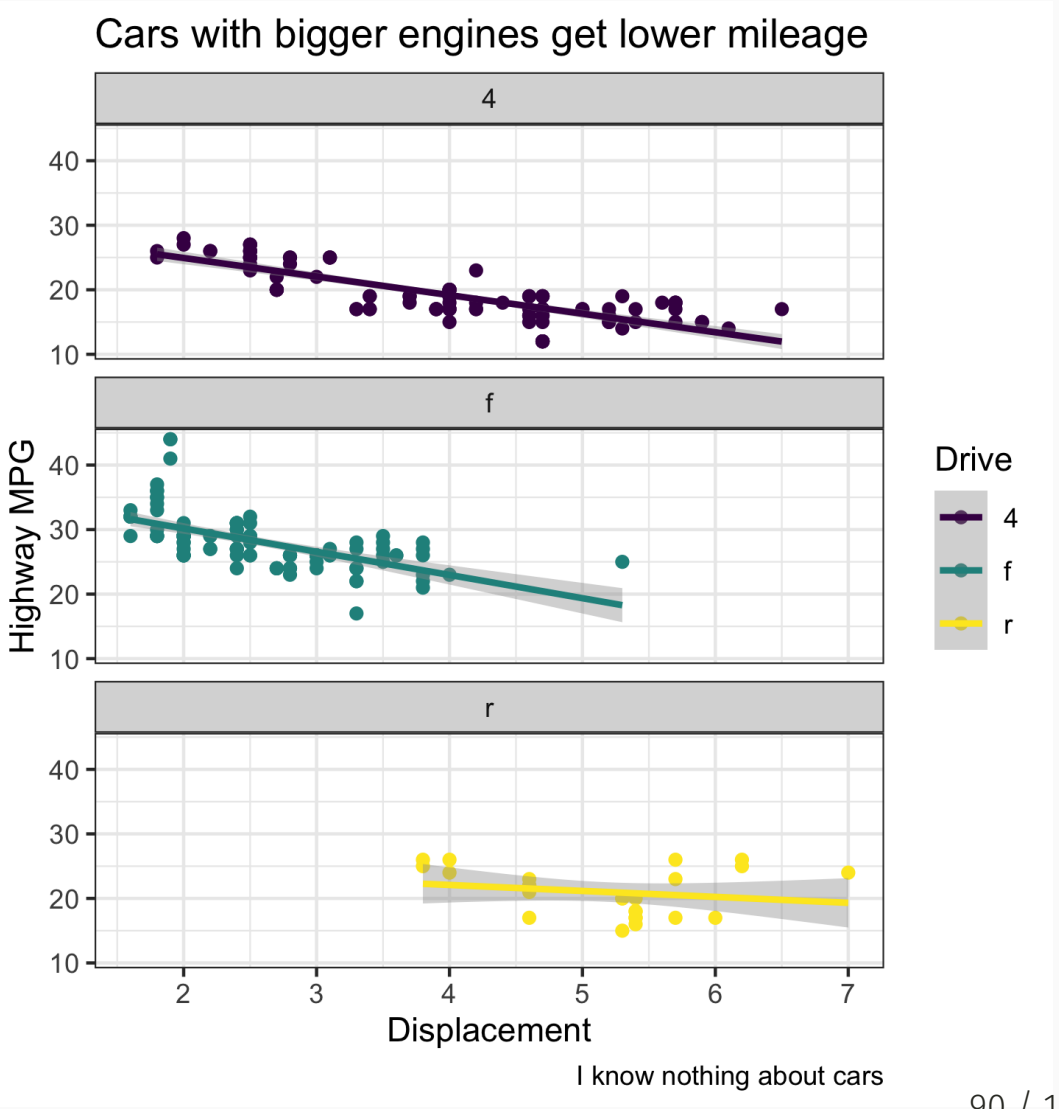
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth(method = "lm") +
+    scale_color_viridis_d() +
+    facet_wrap(vars(drv), ncol = 1) +
+    labs(x = "Displacement", y = "Highway MPG",
+         color = "Drive",
+         title = "Cars with bigger engines get lower
+         caption = "I know nothing about cars")
```



Building a plot step by step with ggplot2 (cont.)

Add a theme

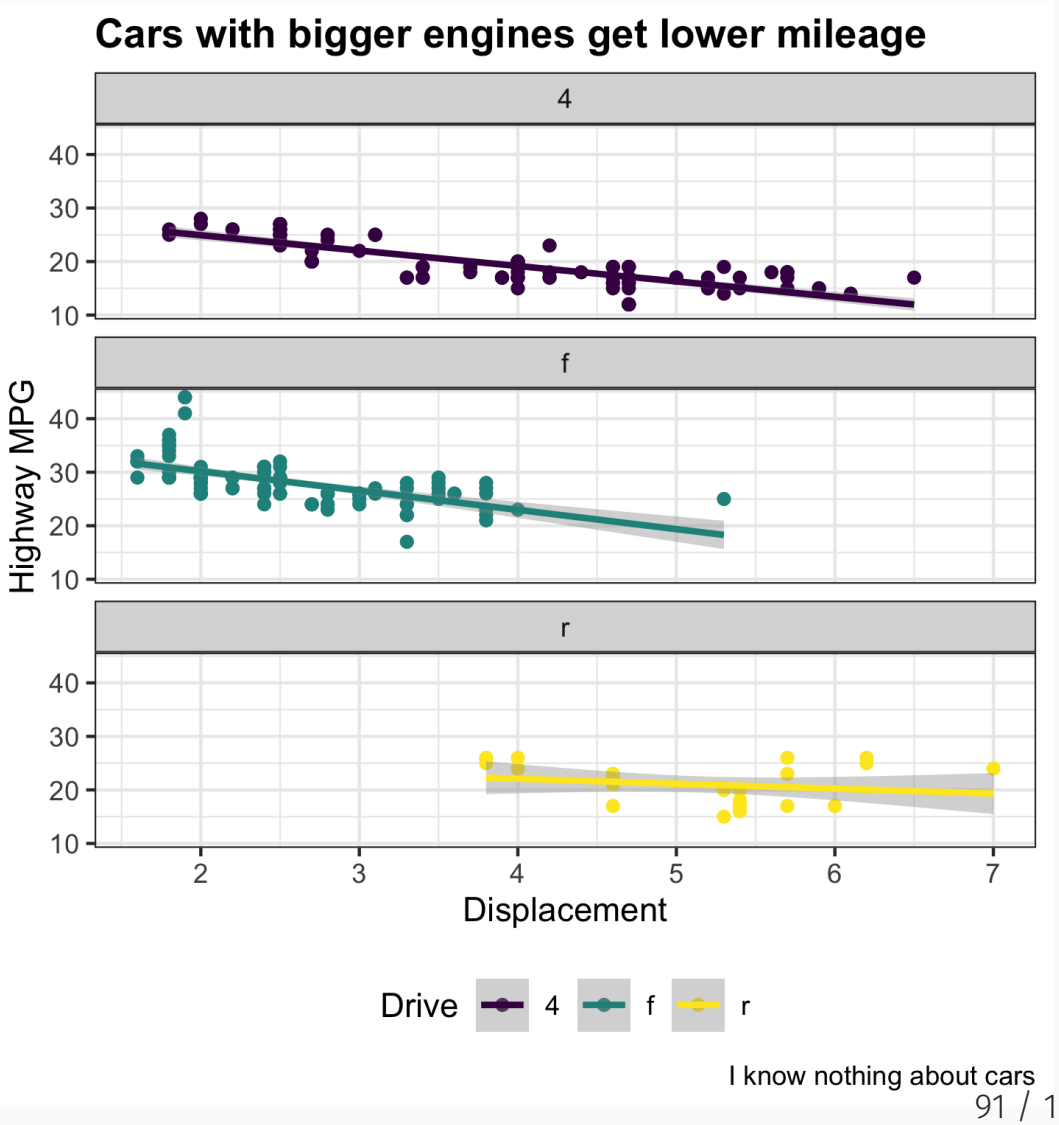
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth(method = "lm") +
+    scale_color_viridis_d() +
+    facet_wrap(vars(drv), ncol = 1) +
+    labs(x = "Displacement", y = "Highway MPG",
+         color = "Drive",
+         title = "Cars with bigger engines get lower
+         caption = "I know nothing about cars") +
+    theme_bw()
```



Building a plot step by step with ggplot2 (cont.)

Modify the theme

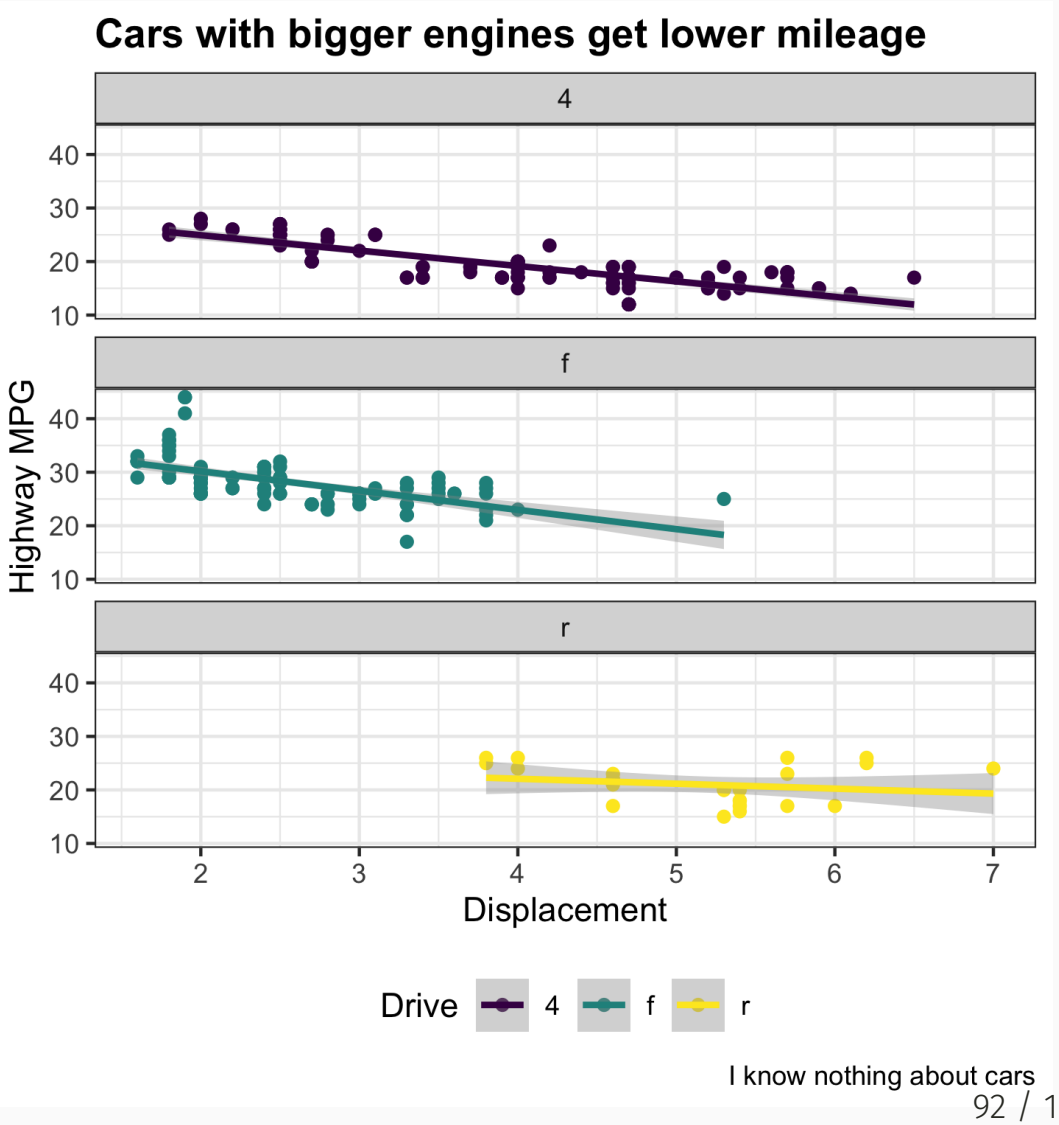
```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth(method = "lm") +
+    scale_color_viridis_d() +
+    facet_wrap(vars(drv), ncol = 1) +
+    labs(x = "Displacement", y = "Highway MPG",
+         color = "Drive",
+         title = "Cars with bigger engines get lower
+         caption = "I know nothing about cars") +
+    theme_bw() +
+    theme(legend.position = "bottom",
+          plot.title = element_text(face = "bold"))
```



Building a plot step by step with ggplot2 (cont.)

Finished!

```
R> ggplot(data = mpg,
+         mapping = aes(x = displ,
+                       y = hwy,
+                       color = drv)) +
+    geom_point() +
+    geom_smooth(method = "lm") +
+    scale_color_viridis_d() +
+    facet_wrap(vars(drv), ncol = 1) +
+    labs(x = "Displacement", y = "Highway MPG",
+         color = "Drive",
+         title = "Cars with bigger engines get lower
+         caption = "I know nothing about cars") +
+    theme_bw() +
+    theme(legend.position = "bottom",
+          plot.title = element_text(face = "bold"))
```



Picking the right image file format

What are the formats?

- At the end of the visualization workflow, you have to decide how to store/export/publish the figures.
- There are many different file formats, but the most important difference is whether they are:
  - **Bitmap (raster graphics)**: store information as a grid of pixels
  - **Vector**: store information as geometric arrangement of individual graphical elements

Table 27.1: Commonly used image file formats

Acronym	Name	Type	Application
pdf	Portable Document Format	vector	general purpose
eps	Encapsulated PostScript	vector	general purpose, outdated; use pdf
svg	Scalable Vector Graphics	vector	online use
png	Portable Network Graphics	bitmap	optimized for line drawings
jpeg	Joint Photographic Experts Group	bitmap	optimized for photographic images
tiff	Tagged Image File Format	bitmap	print production, accurate color reproduction
raw	Raw Image File	bitmap	digital photography, needs post-processing
gif	Graphics Interchange Format	bitmap	outdated for static figures, Ok for animations

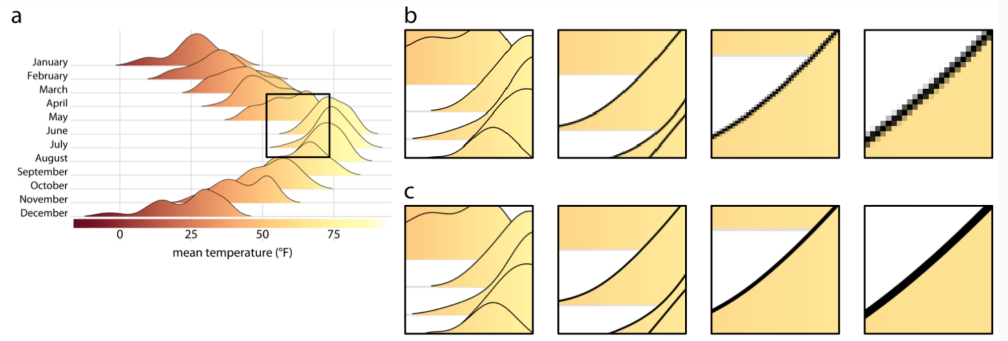


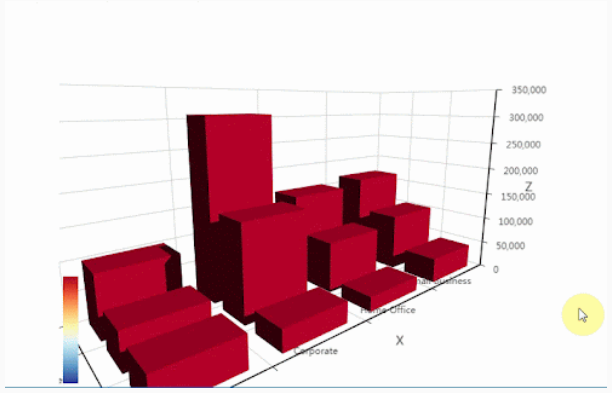
Figure 27.1: Illustration of the key difference between vector graphics and bitmaps. (a) Original image. The black square indicates the area we are magnifying in parts (b) and (c). (b) Increasing magnification of the highlighted area from part (a) when the image has been stored as a bitmap graphic. We can see how the image becomes increasingly pixelated as we zoom in further. (c) Increasing magnification of a vector representation of the image. The image maintains perfect sharpness at arbitrary magnification levels.

Which format to pick?

- Use `pdf/svg` whenever possible.
- Use `png` for online documents/presentations.
- Use `jpeg` as last resort (in particular if pngs are too large and loss by compression is acceptable).

Going interactive

- In online presentations of your data and analyses, you might want to go interactive.
- Interactive webpages are mainly run with JavaScript, and we can use R to draw on JavaScript libraries to create interactive content.
- In particular, the `htmlwidgets` package provides a framework to bind R commands to various JavaScript libraries, including those that create data graphs.
- Many "widgets" are already available - check out http://gallery.htmlwidgets.org/.
- Other JavaScript libraries for interactive graphics can be created with R, too:
  - `leaflet` to connect to the Leaflet library and create interactive maps
  - `plotly` to connect to Plotly and create graphs of all kinds
- For the record: interactive ≠ animated. On the right you see animated charts. Please don't do this at home.



Some notes on the right visualization software

1. Always conceptualize first.

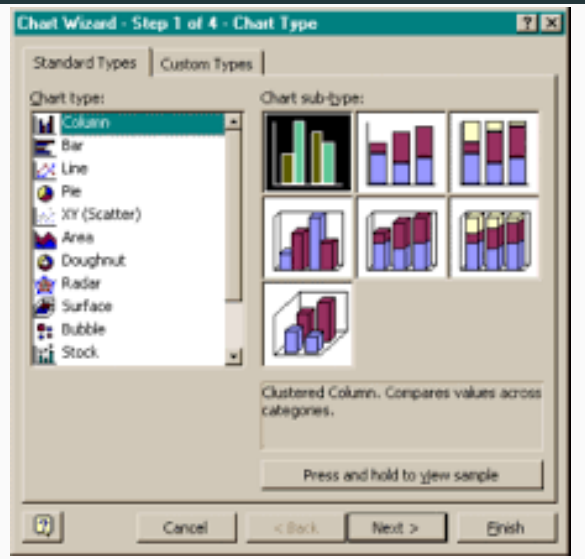
2. Prioritize programmatic solutions to stay reproducible (i.e., R over Excel).

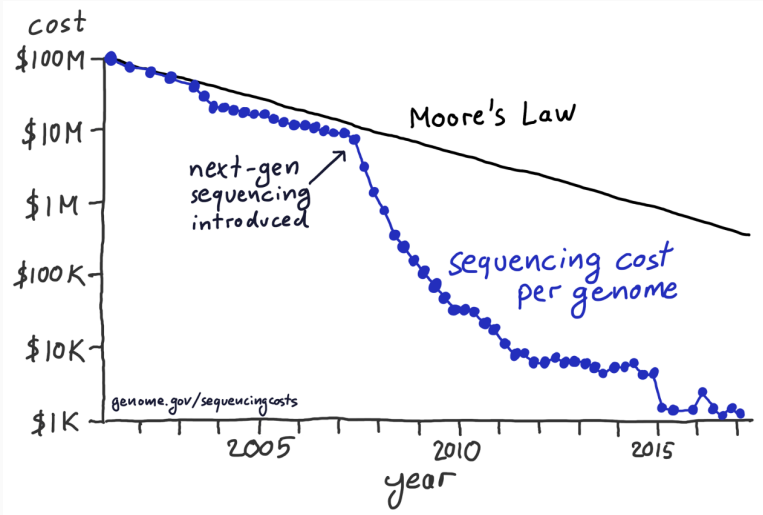
3. For conceptual charts (not: data viz!), other tools might be just fine (e.g., Powerpoint or even hand-drawn figures).

4. Don't be distracted by interactives (as offered by, e.g., highcharts.com, Tableau, and others).

5. Designing good graphs is a learnable skill. Study how others do it in your software of choice!

6. But a good visualization takes time, even if you're experienced. Working a full day on the key plot of your analysis? No problem! If you go the coding route, this is a good investment for Future-You.

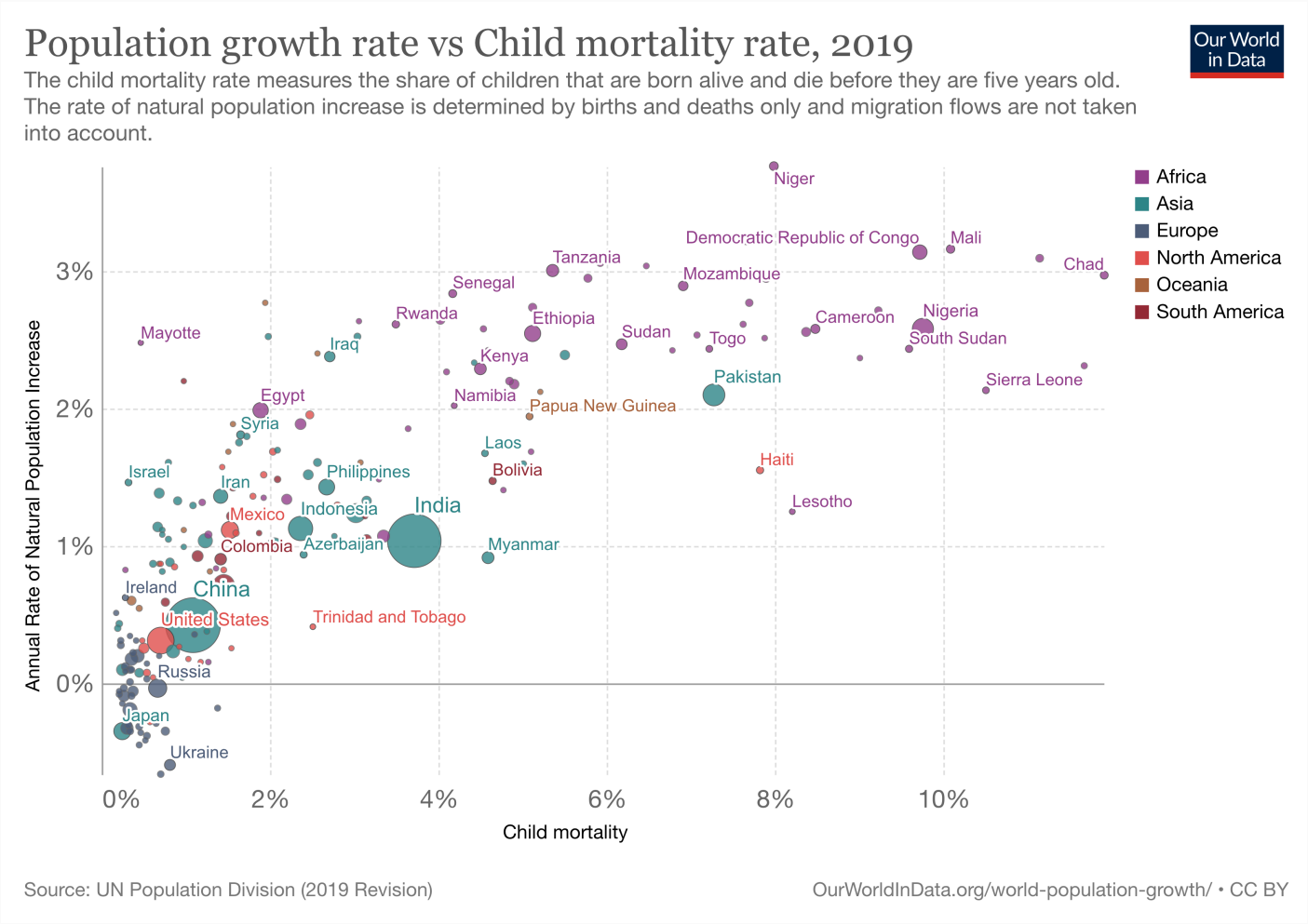




The best statistical graph of all times

The best statistical graph of all times

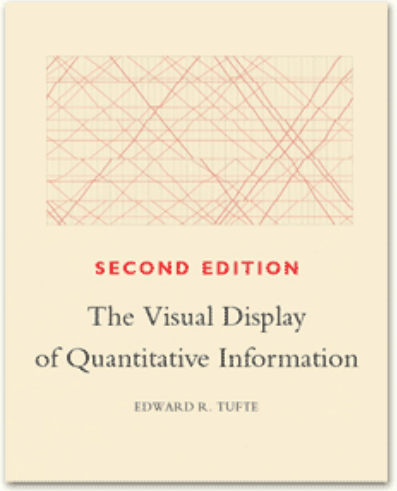
- "Indeed, among all the forms of statistical graphics, **the humble scatterplot** may be considered the most versatile, polymorphic, and generally useful invention in the entire history of statistical graphics." (Friendly & Denis 2005)
- There's another lesson to learn here: **Keep it simple.** Reading visualizations is a skill, and most people exposed to your work will be worse at it than you.

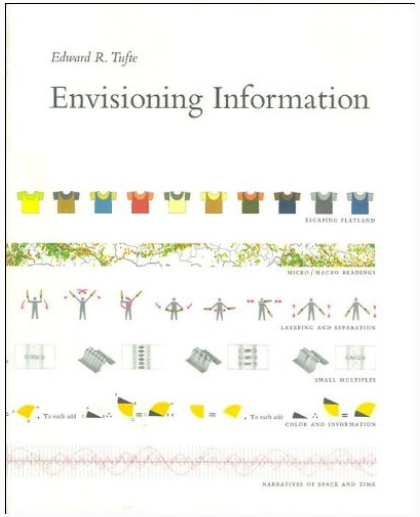


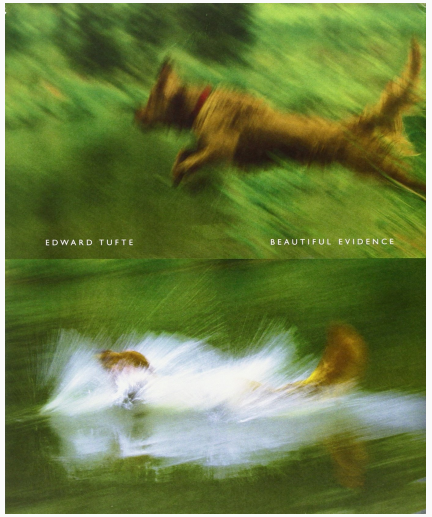
The best statistical graph of all times

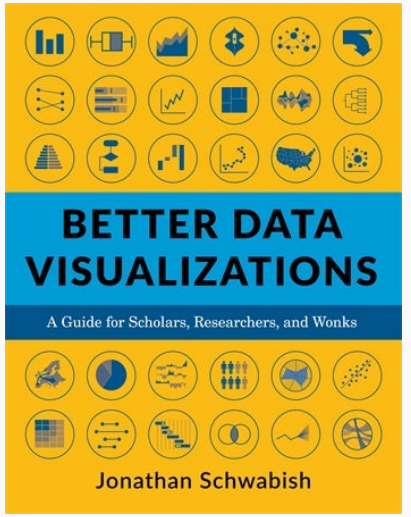
- "Indeed, among all the forms of statistical graphics, **the humble scatterplot** may be considered the most versatile, polymorphic, and generally useful invention in the entire history of statistical graphics." (Friendly & Denis 2005)
- There's another lesson to learn here: **Keep it simple.** Reading visualizations is a skill, and most people exposed to your work will be worse at it than you.

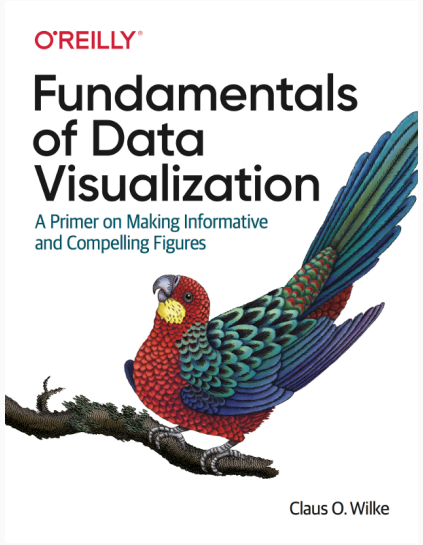
Further reading

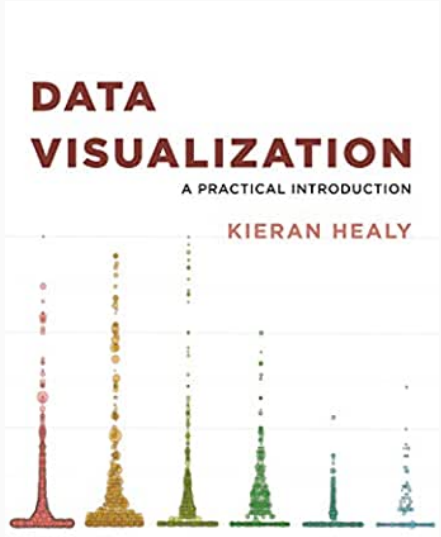


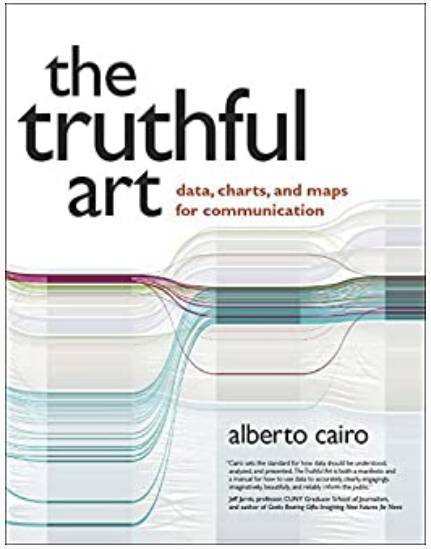


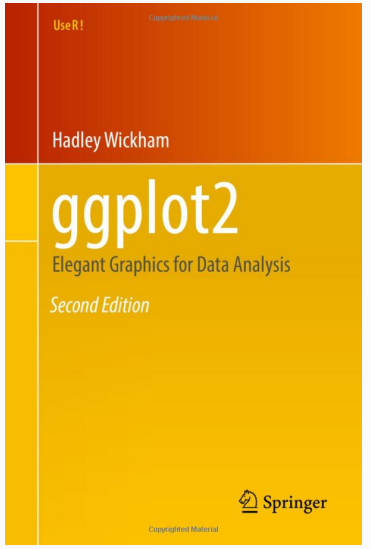












Coming up

Assignment

Assignment 5 is about to go online on GitHub Classroom. Check it out and start plotting!

Next lecture

The workshop! 😱