

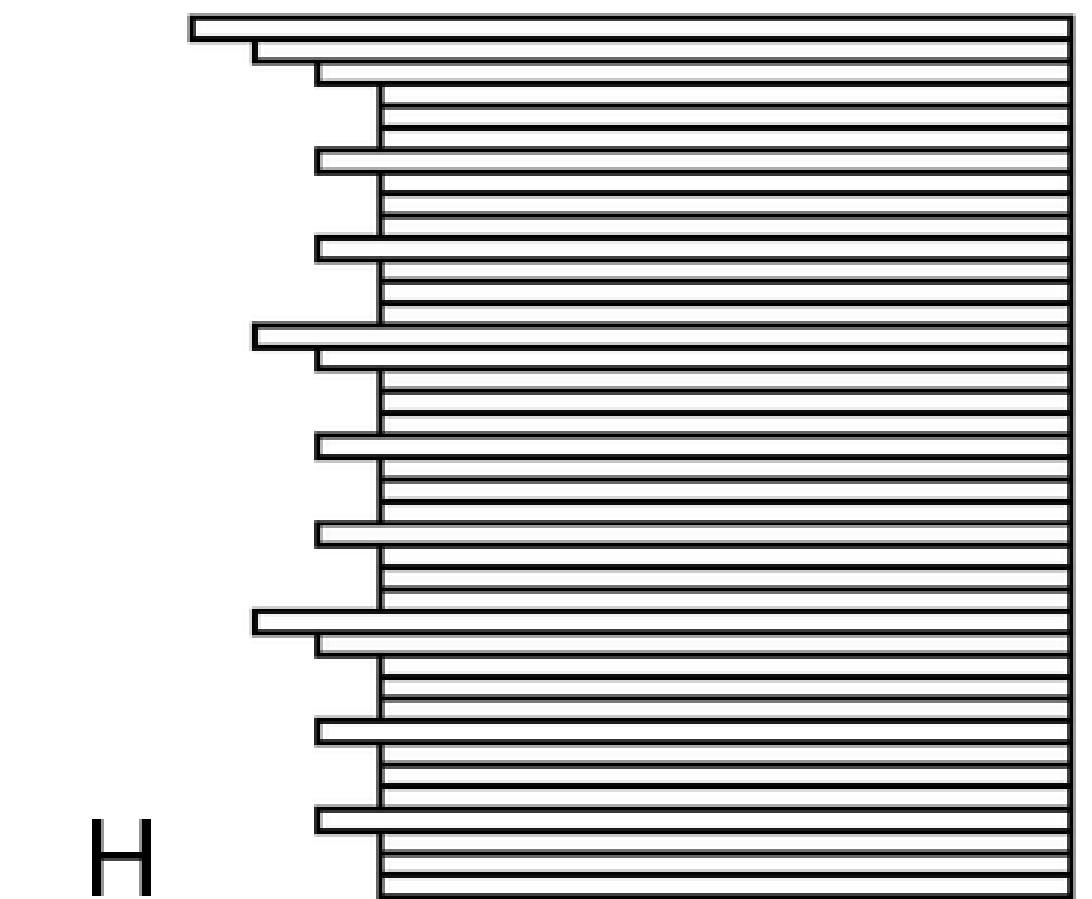
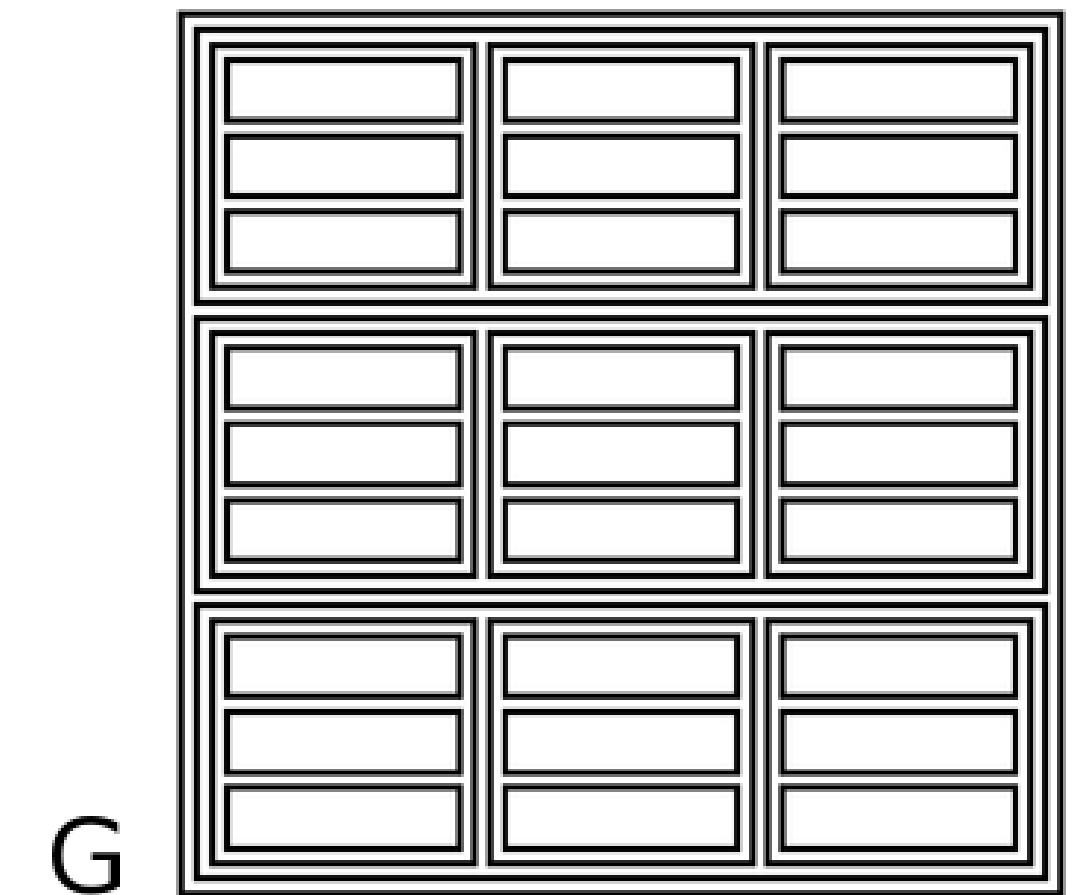
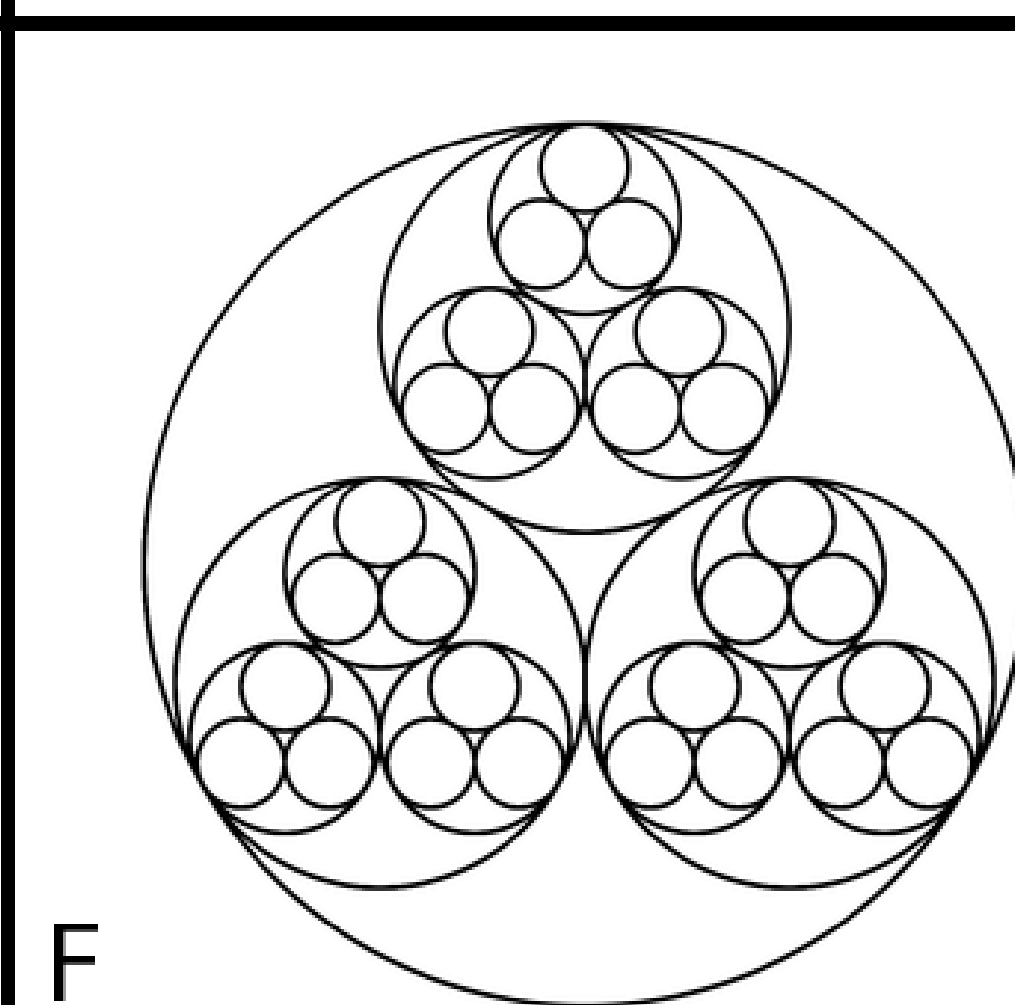
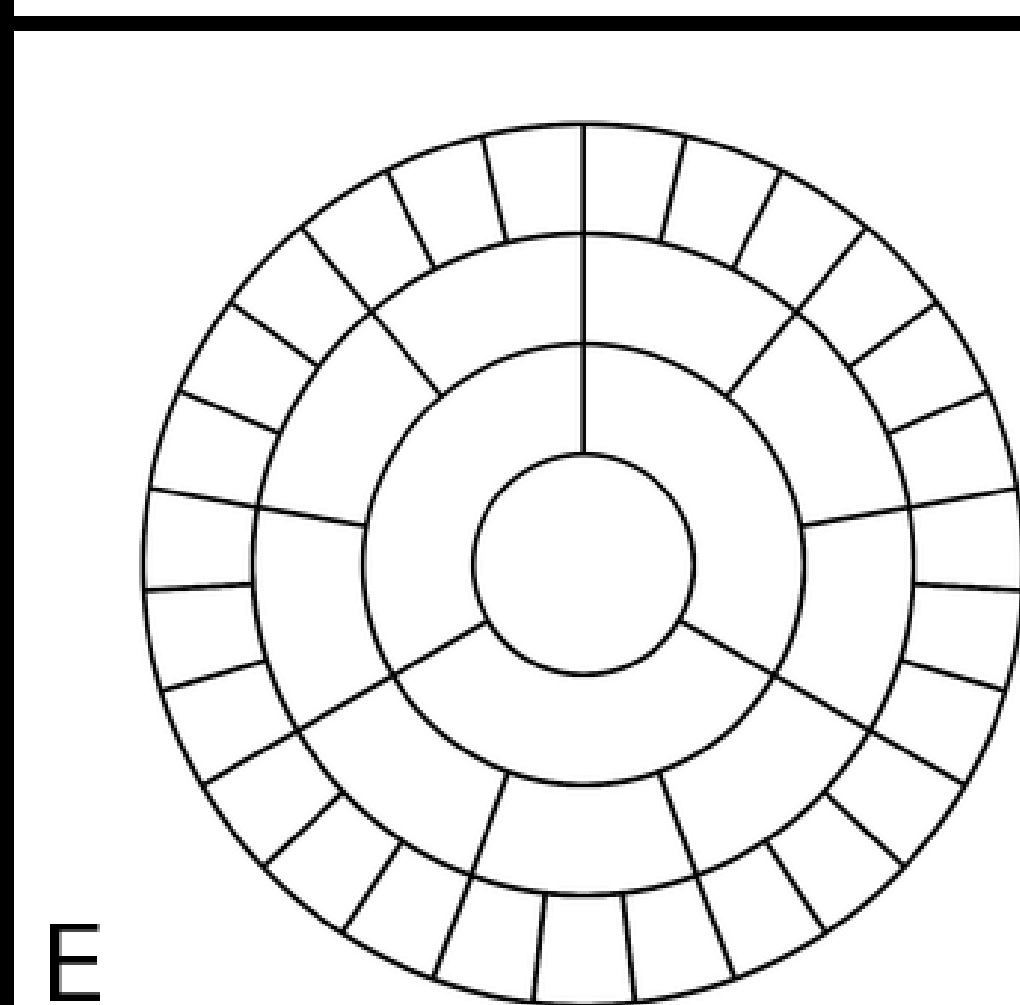
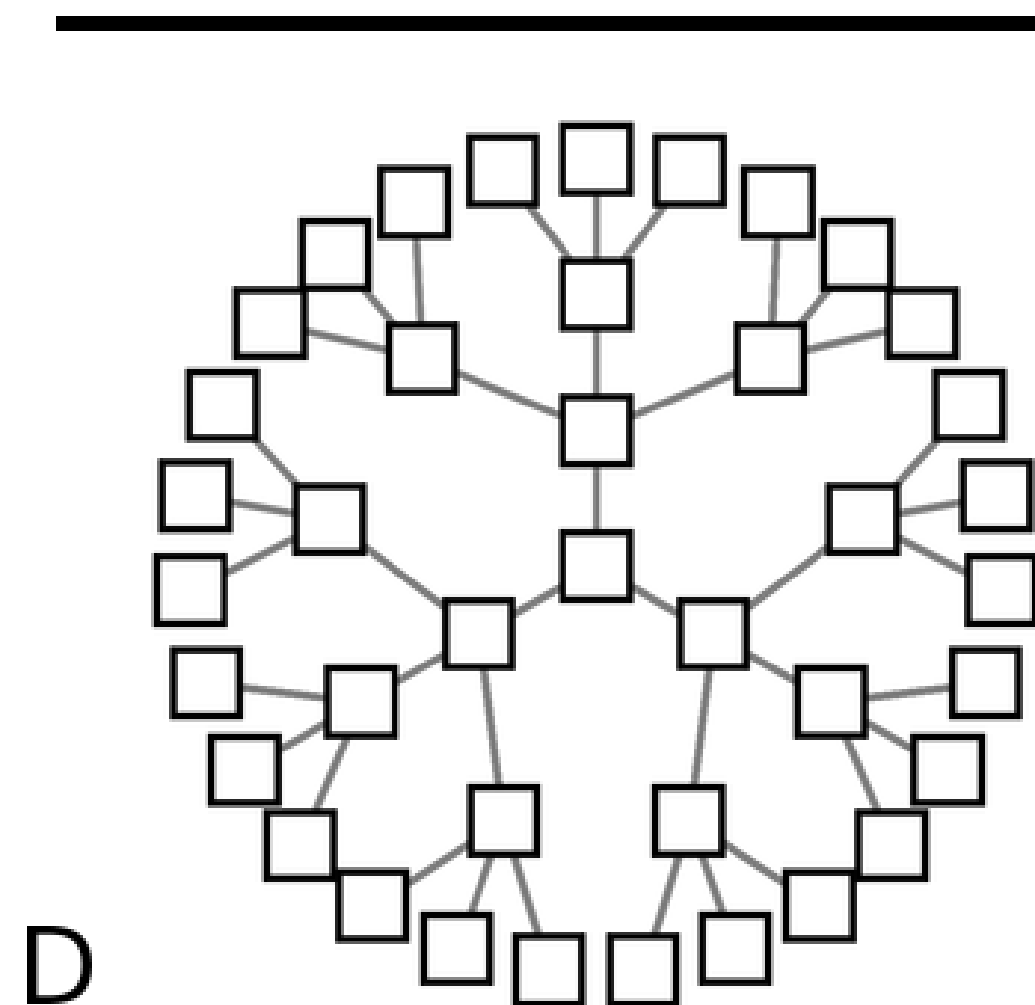
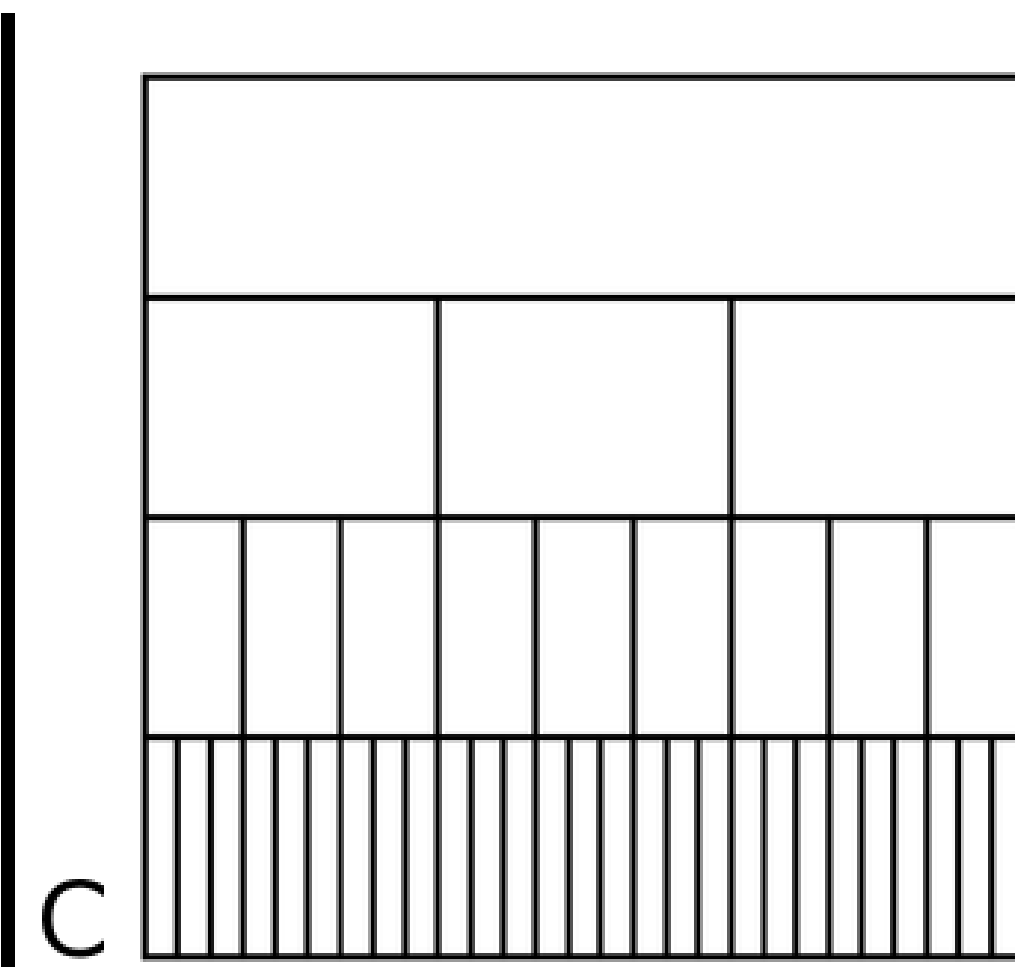
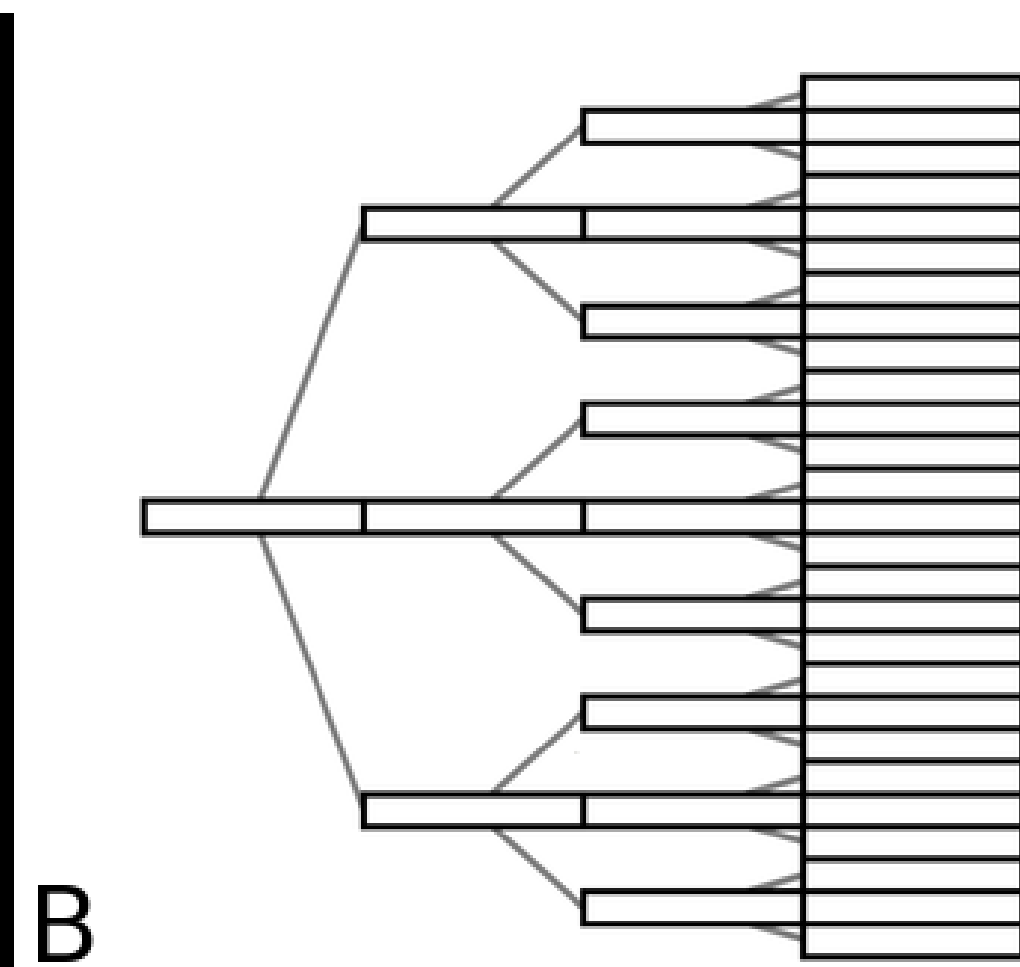
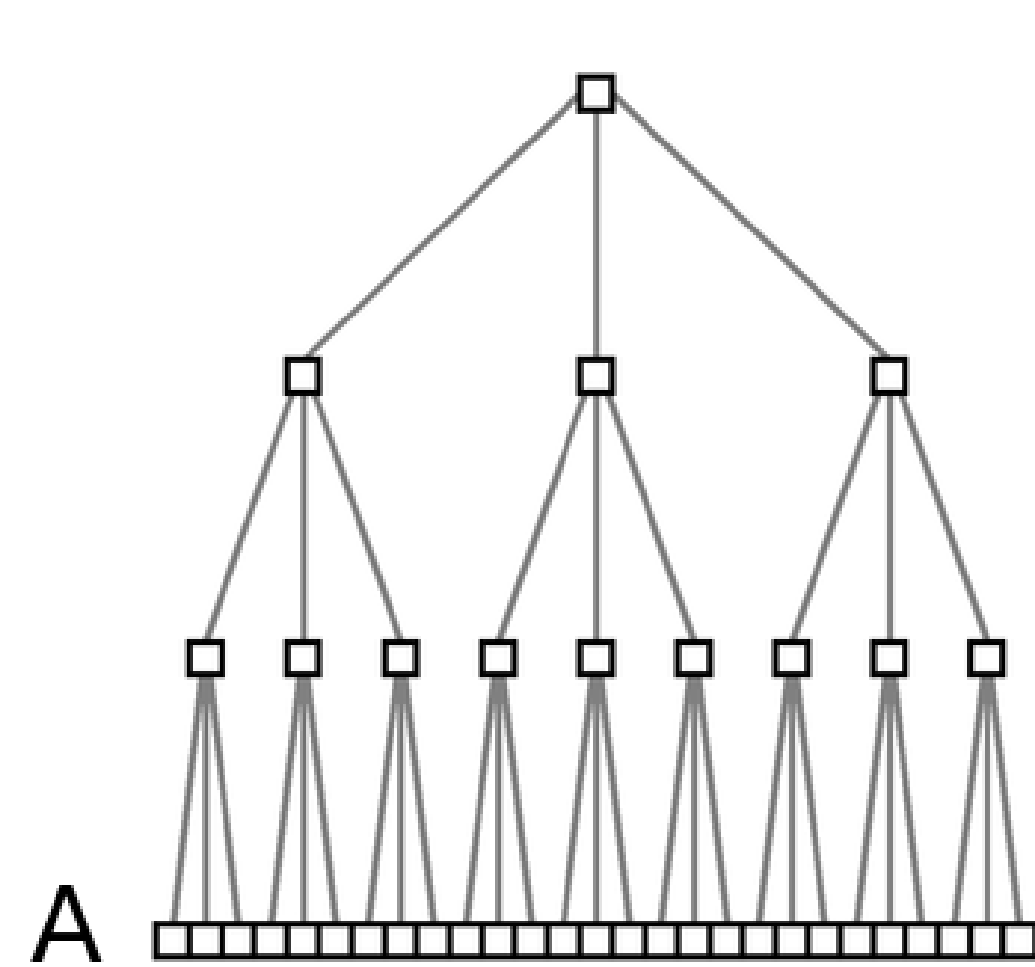
# Data Visualization (CSCI 627/490)

---

Design

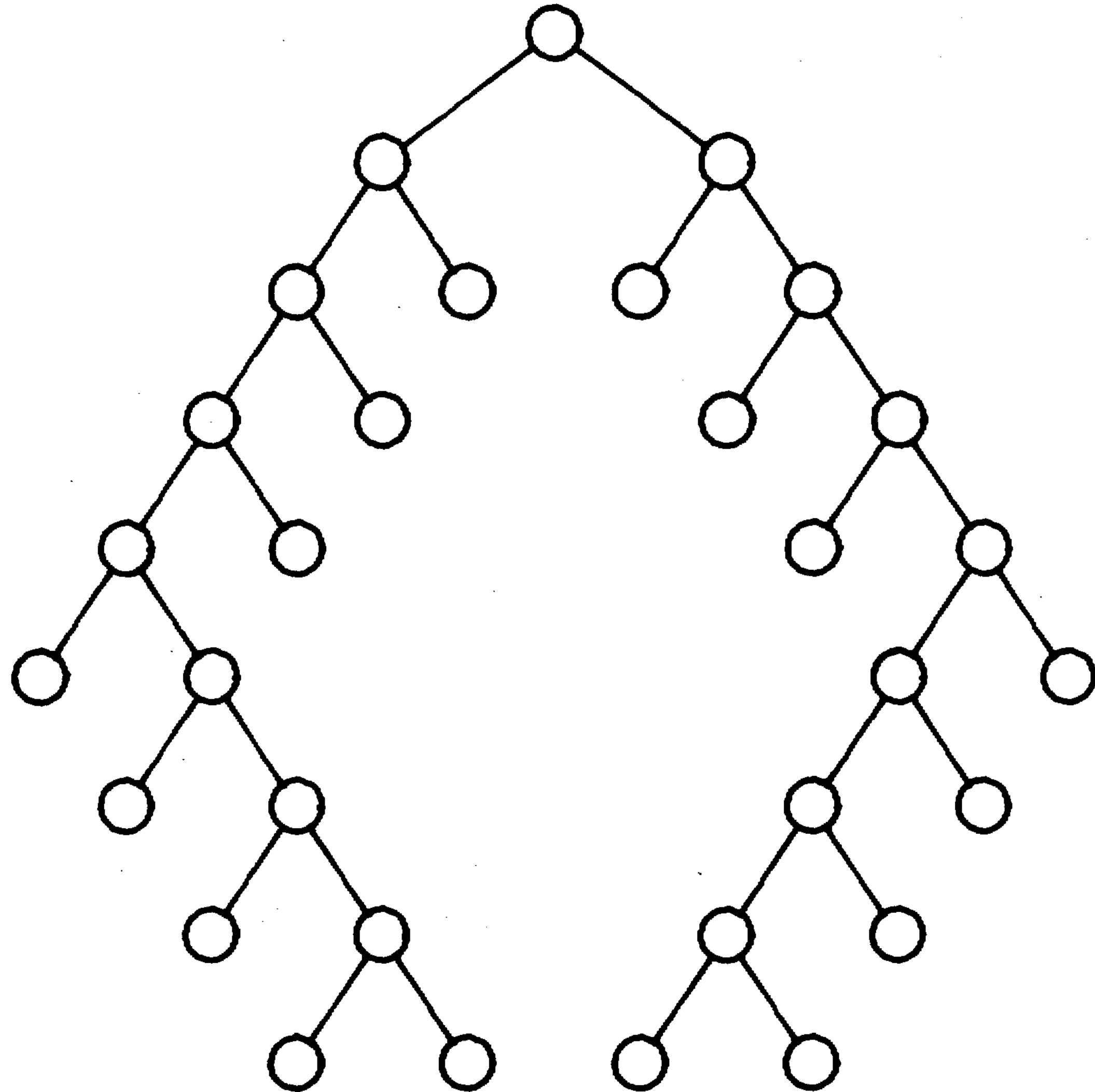
Dr. David Koop

# Tree Visualizations



[McGuffin and Robert, 2010]

# Reingold-Tilford Algorithm



- Recurse on left and right subtrees
- Shift subtree over as long as it doesn't overlap
- Place parent centered above the subtrees
- Originally, only binary trees, extended by Walker

[Reingold and Tilford, 1981]

# Treemap

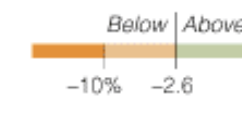
## Truck Sales Slip, Tripping Up Chrysler

Over the past few years, Chrysler executives said they were following the lead of Toyota and Honda, focusing on vehicles that met the needs of their customers. But as American consumers turned away from large trucks and S.U.V.'s in 2006, Chrysler continued to churn out big vehicles, which are now sitting unsold at dealerships across the country.

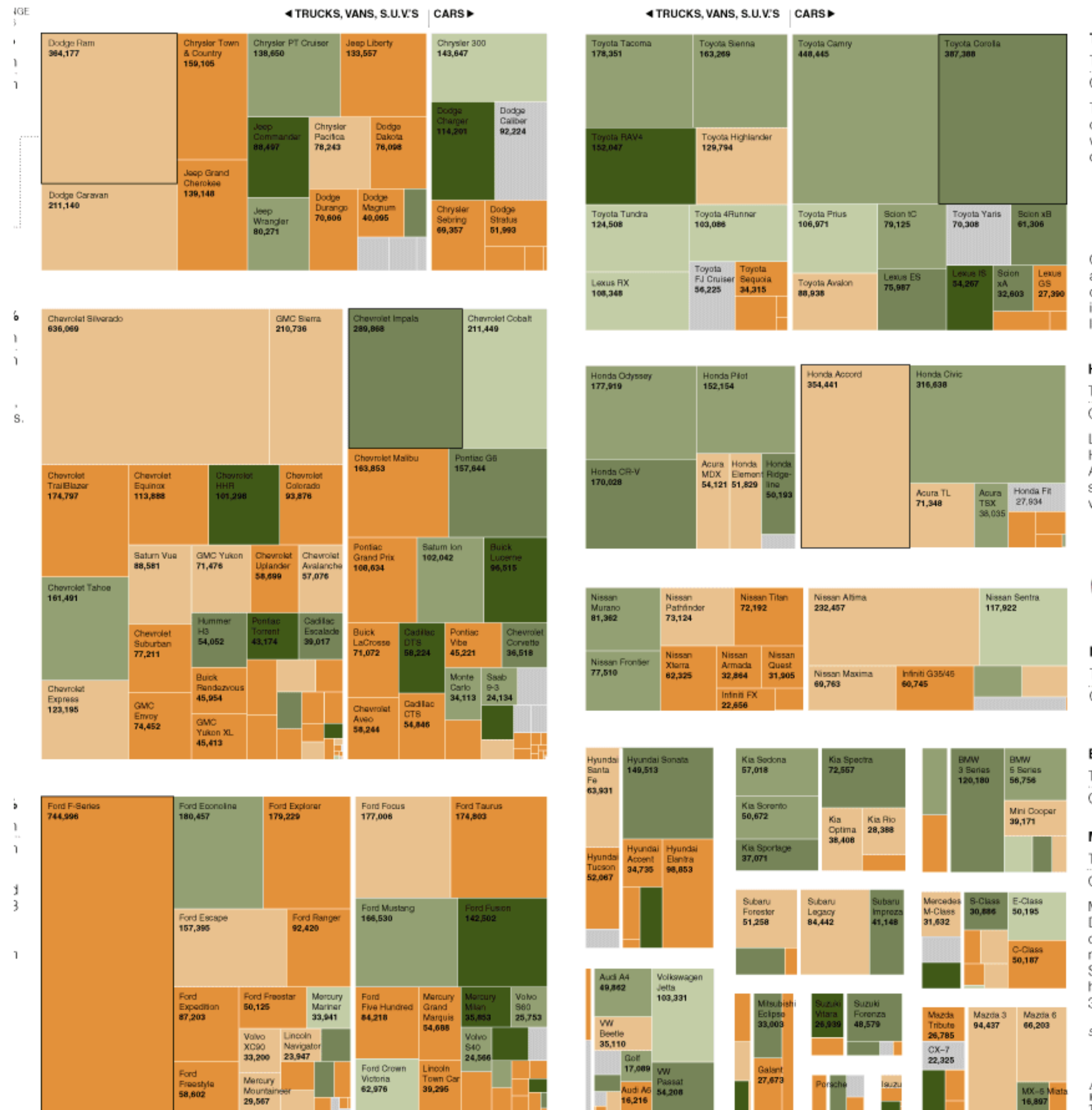
### READING THE CHART

Boxes are scaled proportionally according to number of cars sold in 2006

Change in sales from



- Containment marks instead of connection marks — show hierarchy
- Encodes some quantitative attribute of the items as the **size** of the rectangles
- Not as easy to see the intermediate rectangles (hierarchy)
- Scalability: millions of leaf nodes and links possible

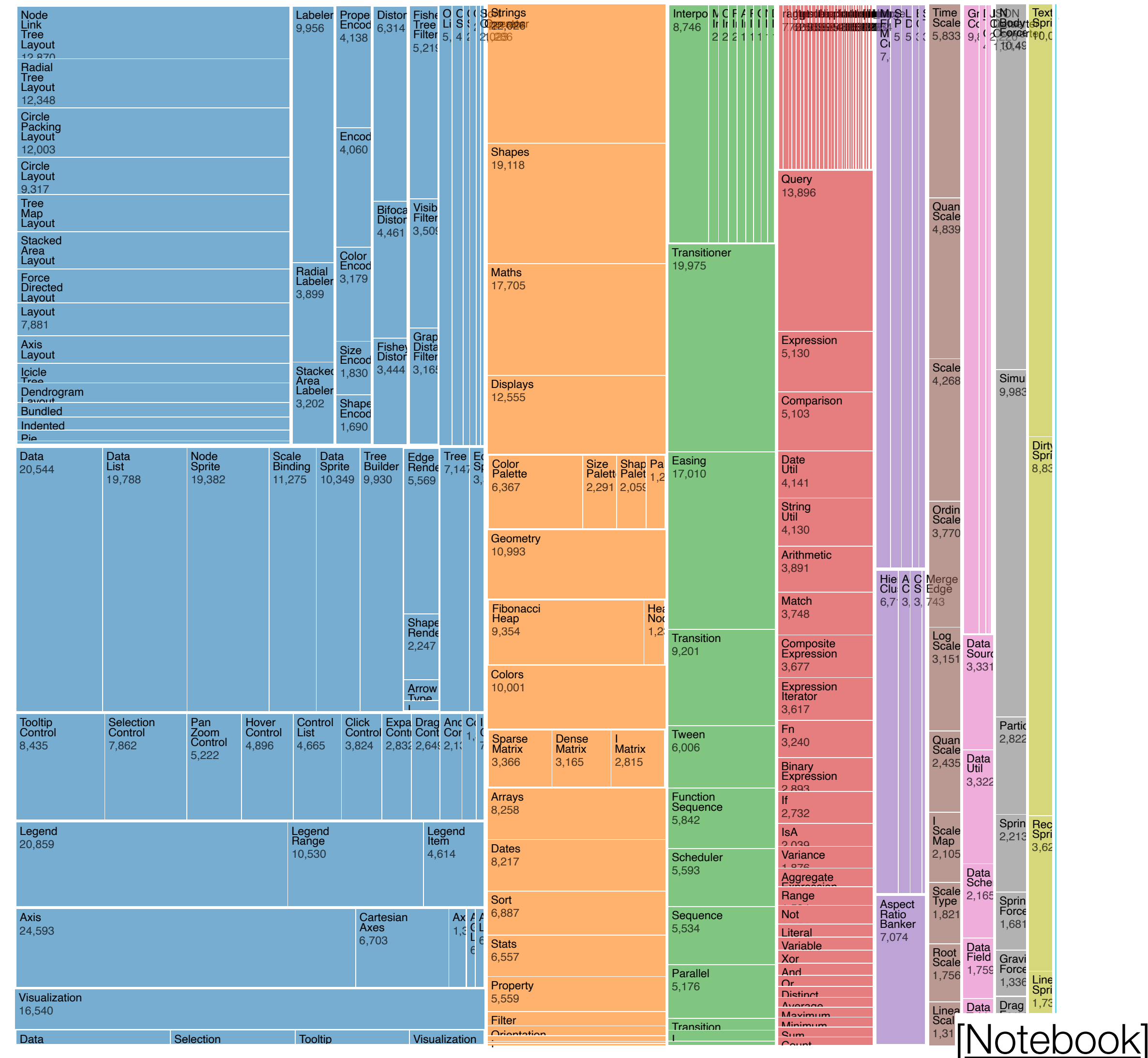


[A. Cox and H. Fairfield, NYTimes, 2012]



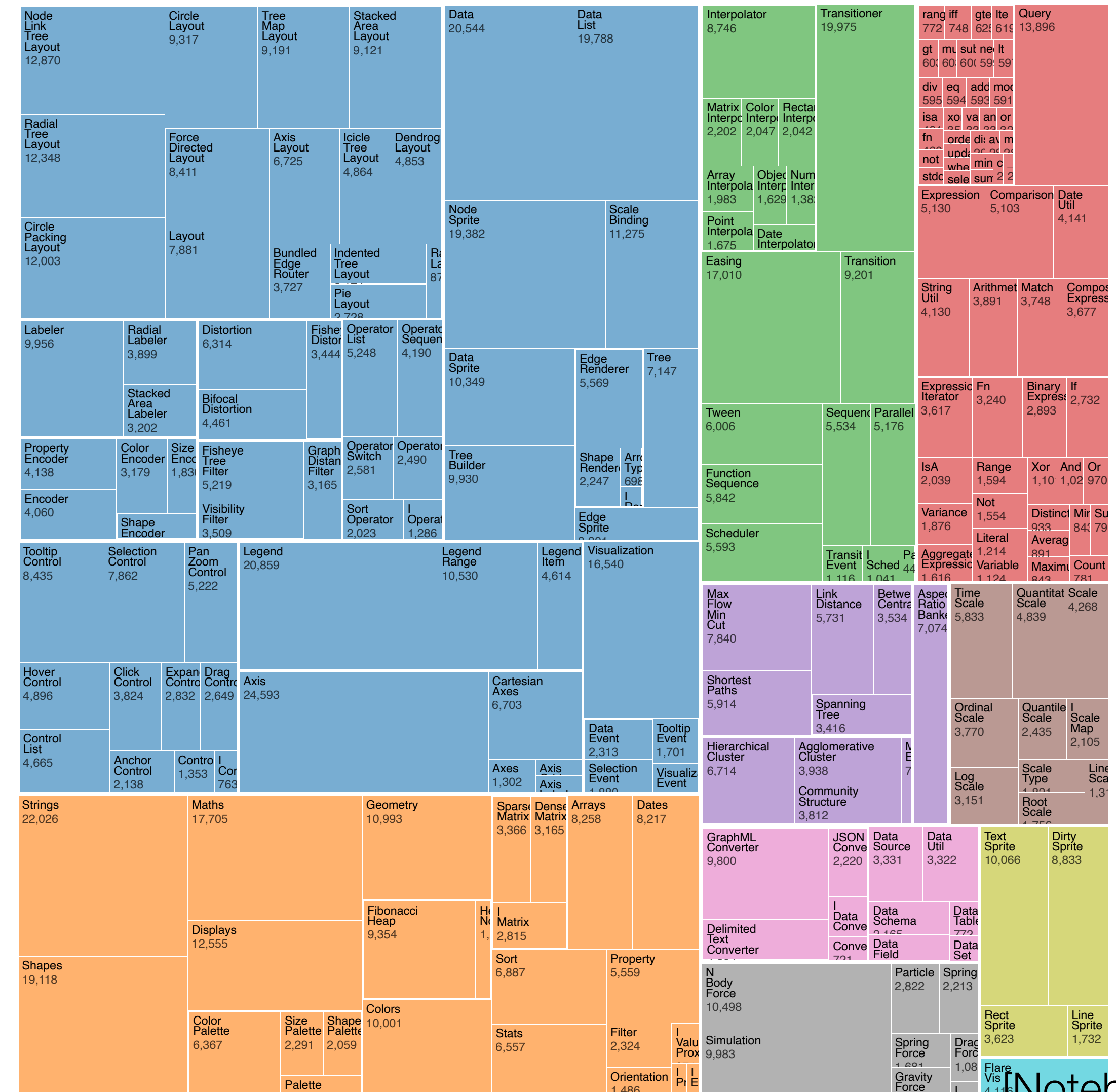
# Treemap Layouts: Slice & Dice

- Split at each level into strips
- At each step, orientation of division (horizontal/vertical) changes
- Better, but some rectangles still have bad aspect ratio



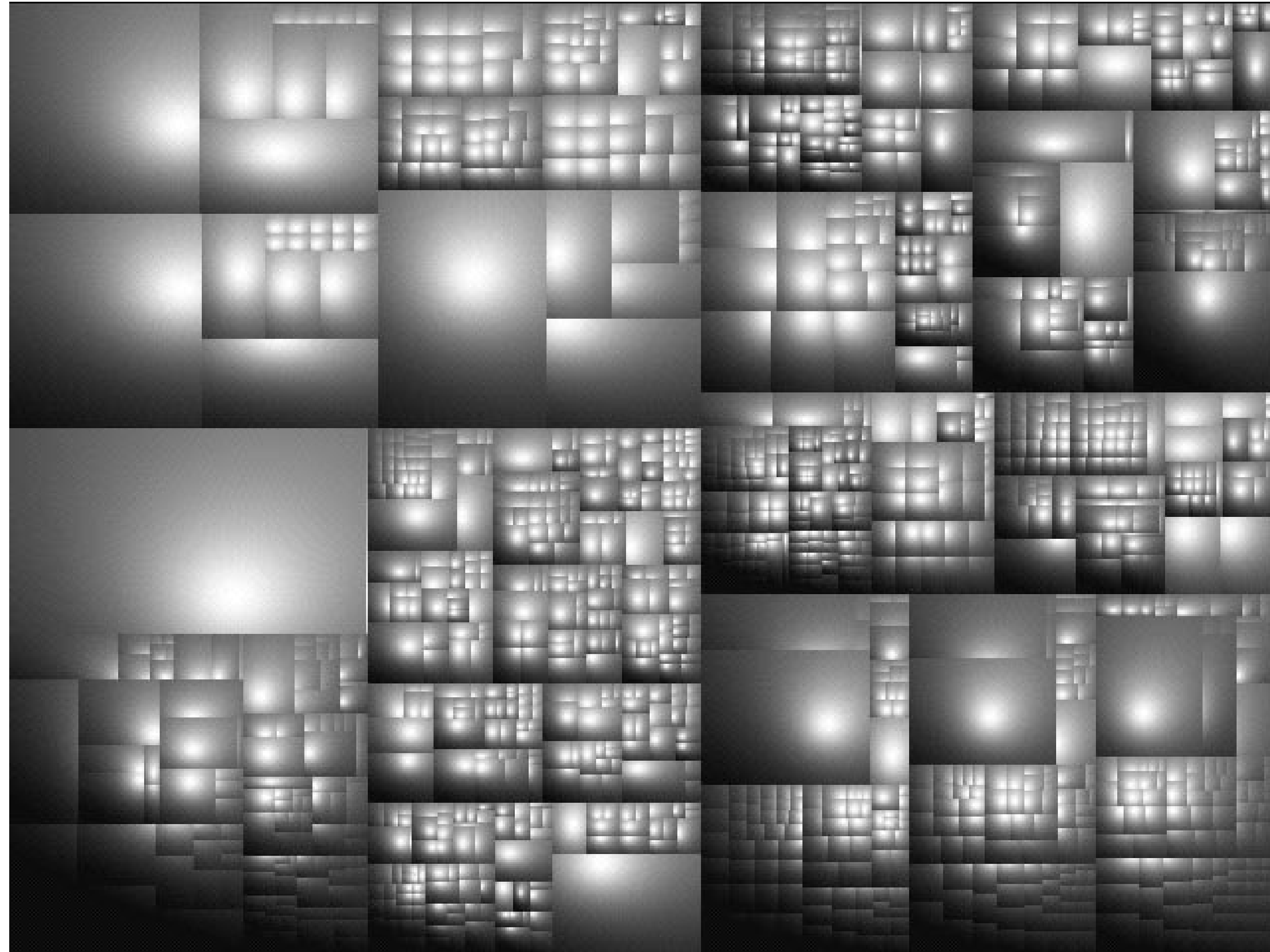
# Treemap Layouts: Squarify

- Slice & Dice and Strip can lead to bad **aspect ratios**
- Solution: Strip only uses rows, allow columns to be used, too
- Choose divisions ( $x/y$ ) based on the width/height of region in order to maintain good aspect ratios
  - Use left and right side
  - Process large rectangles first
- Ordering not preserved which may cause issues if the data is updated

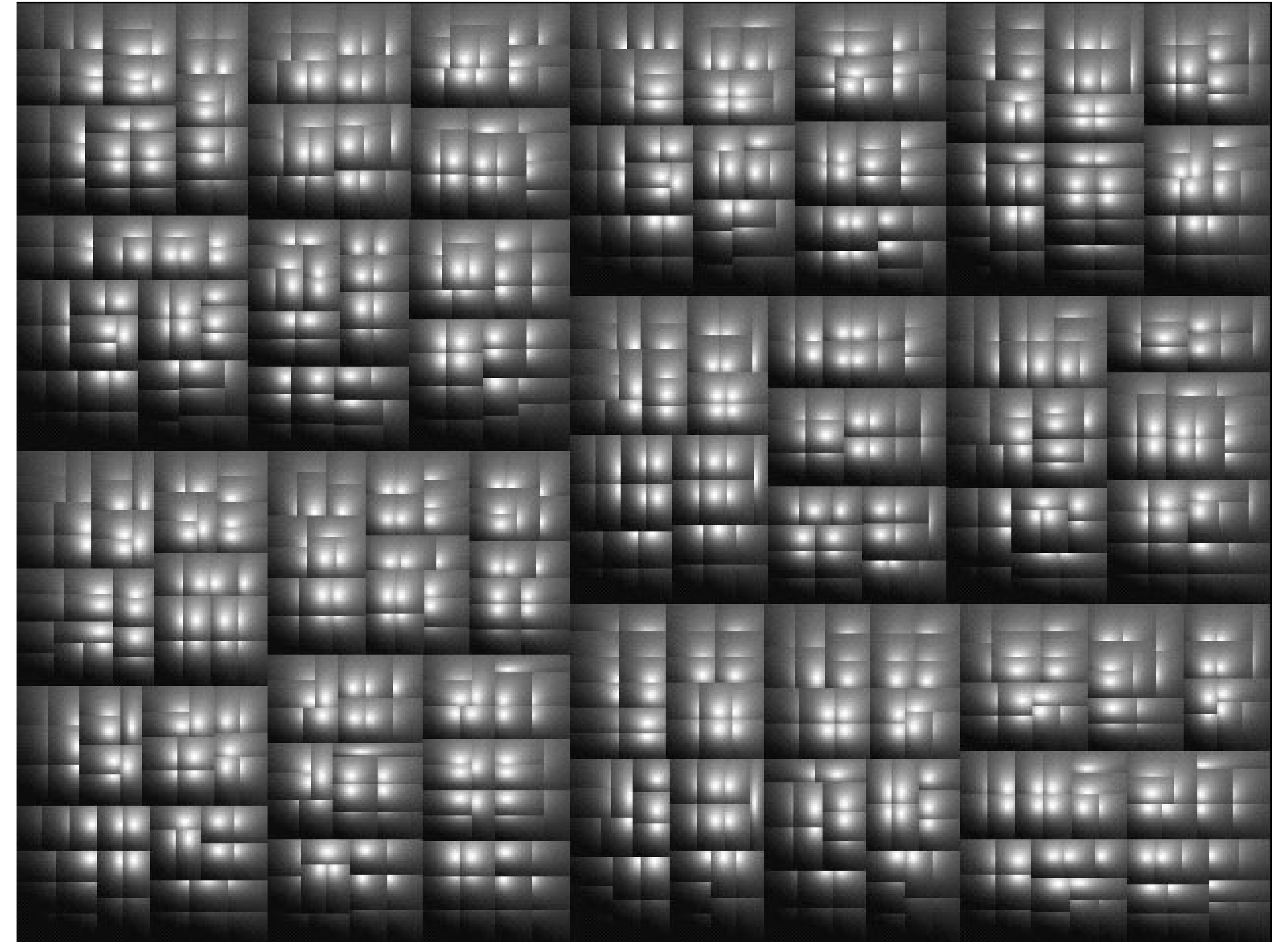


# [Notebook]

# Squarified + Cushioned Treemaps



(a) File system

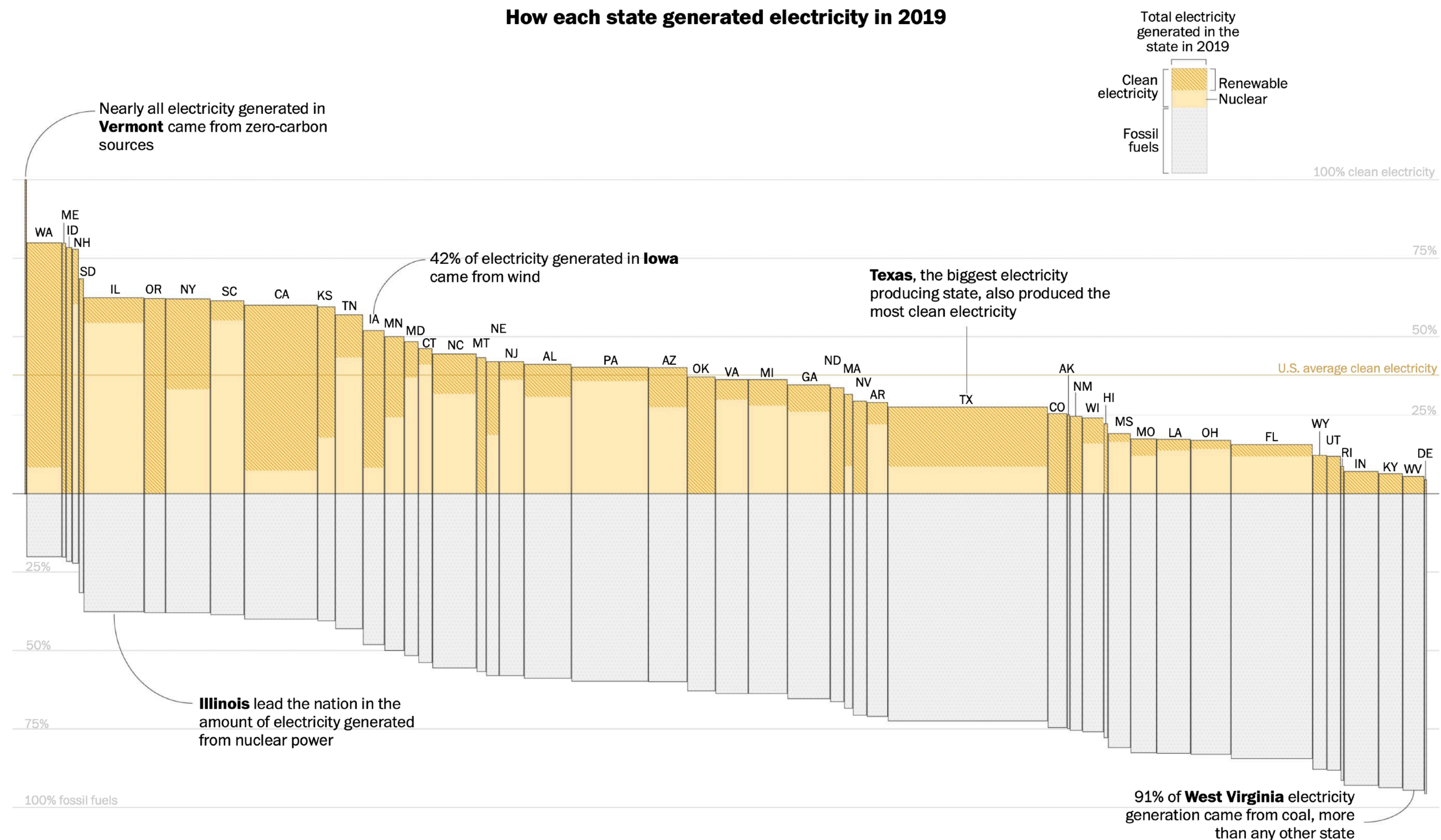


(b) Organization

[Brus et al., 1999]



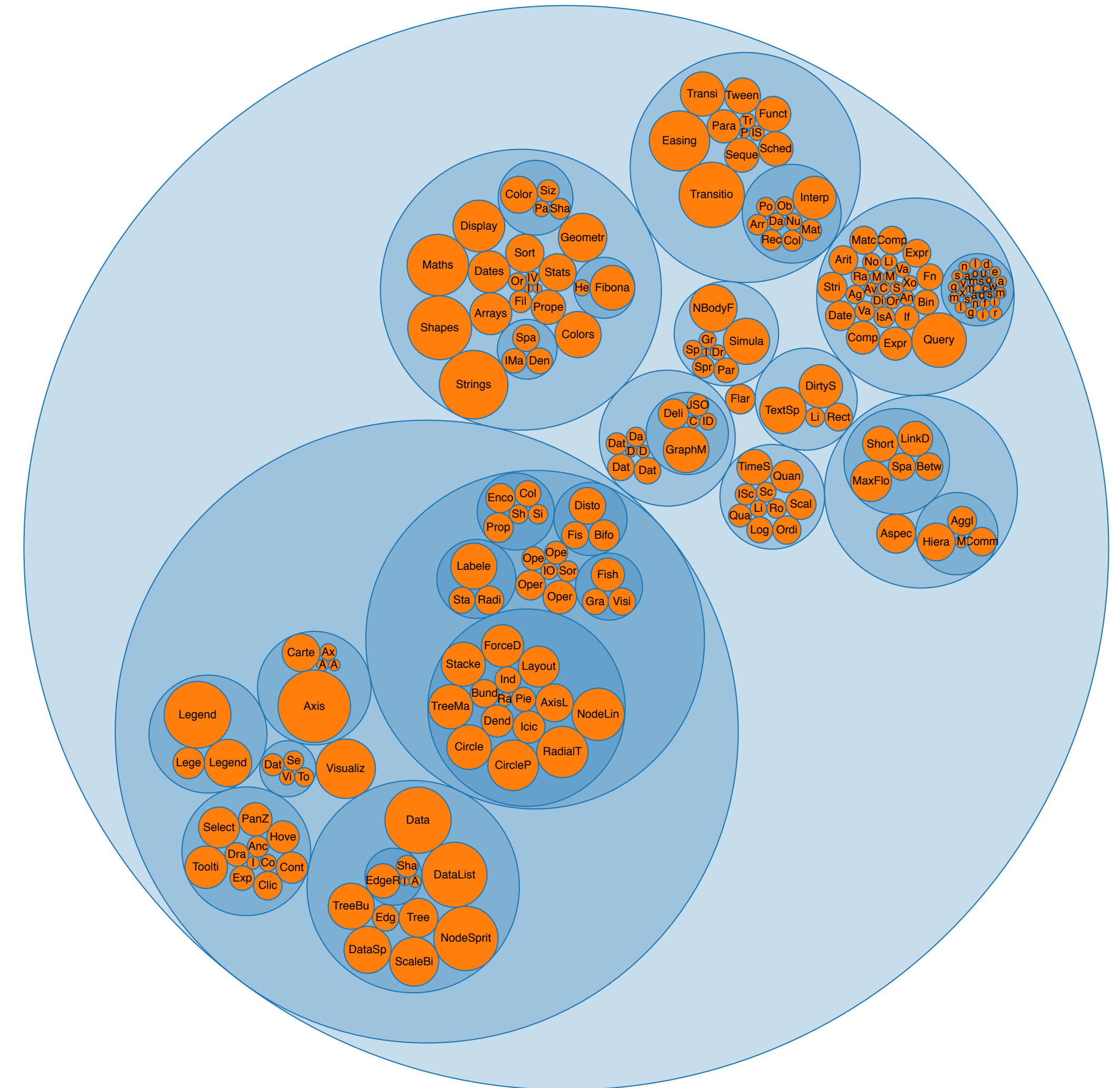
# Variations: Marimekko Chart



[J. Muyskens, [Washington Post](#)]

# Nested Circles

- Looks more like cluster diagram, but shows hierarchy
- Containment shown by the layering of semi-transparent circles
- Labeling becomes more difficult

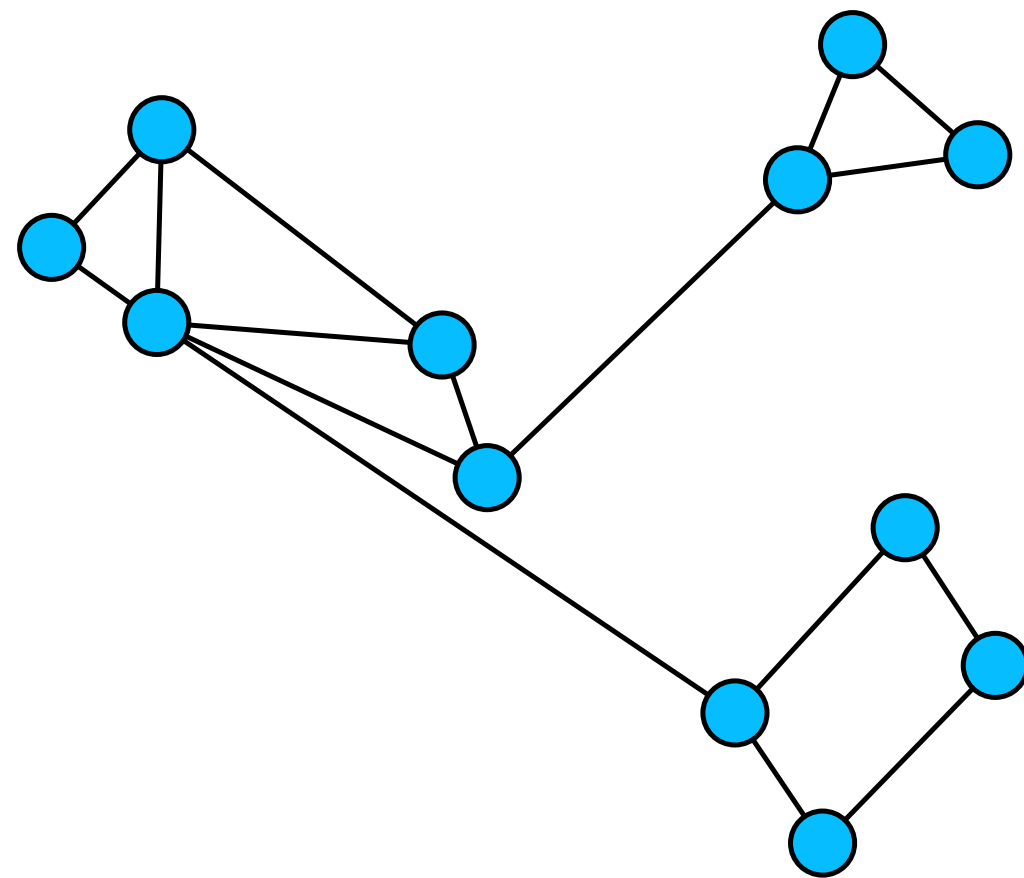


[Bostock, 2012]

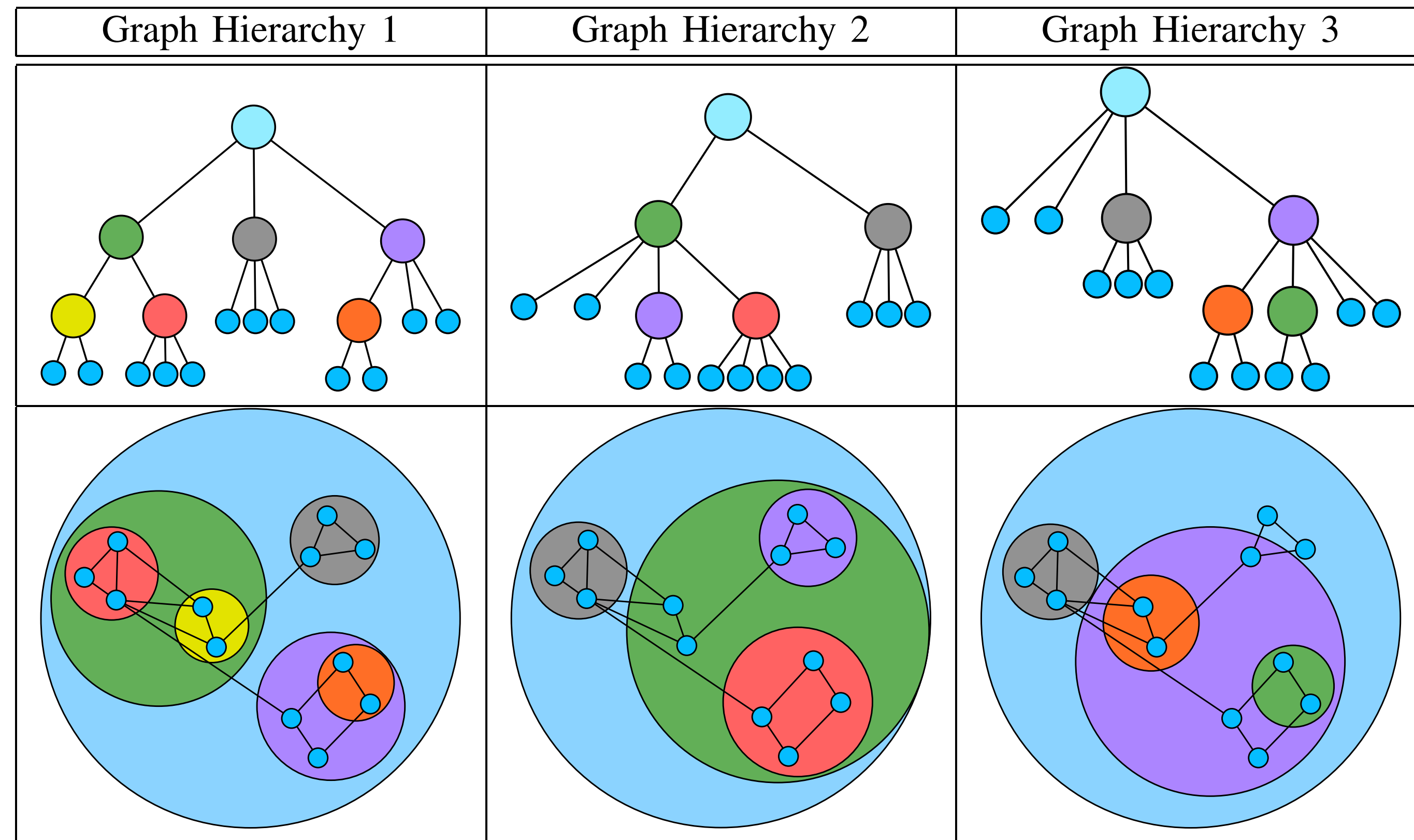


# Compound Networks

- Add a hierarchy to the network (e.g. from clustering)
- GrouseFlocks: uses nested circles with colors



(a) Input Graph



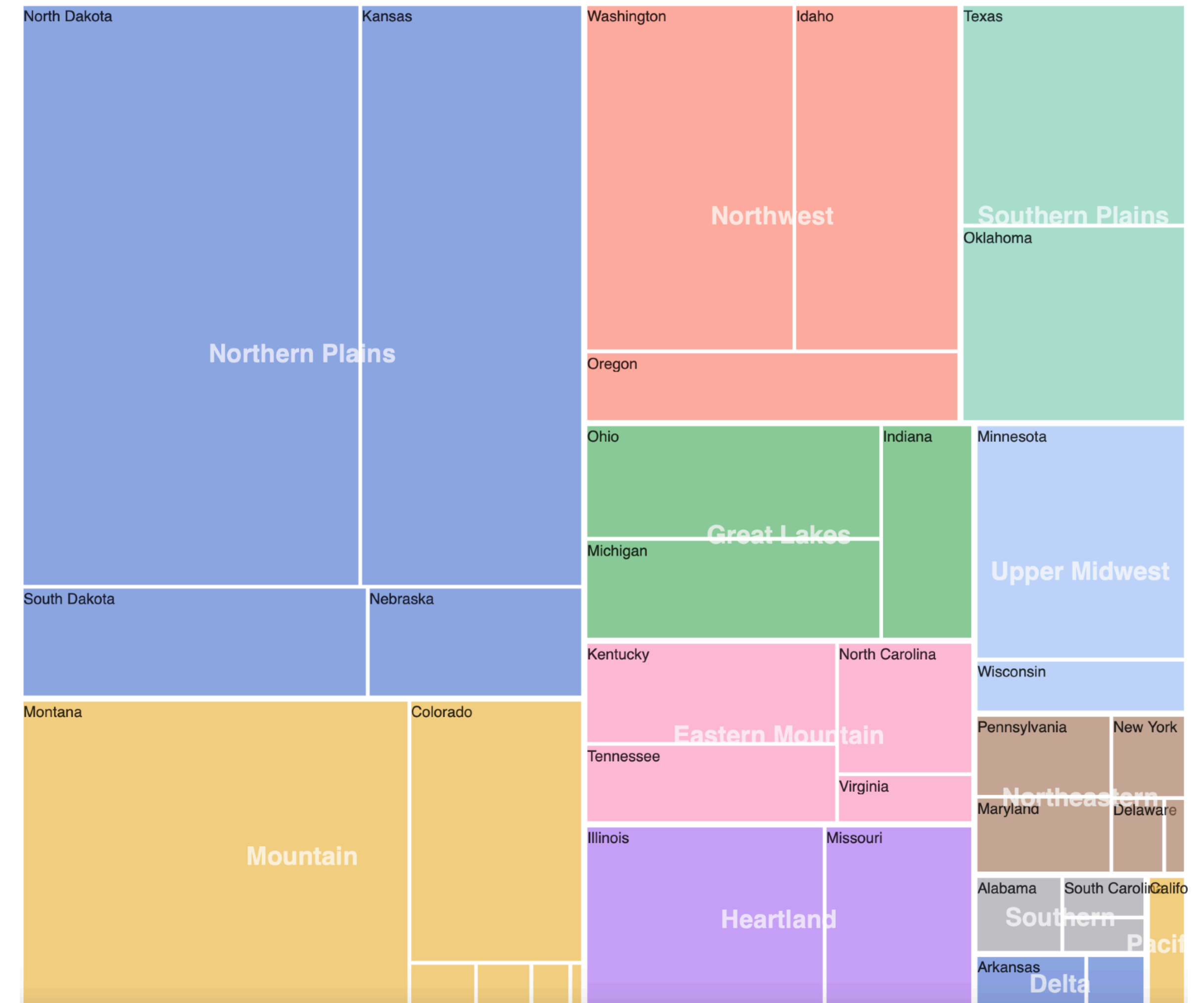
(b) Graph Hierarchies

[Archambault et al., 2008]



# Assignment 4

- Crop Production in the US
- Geospatial Visualizations & Treemap
  - Choose colormaps carefully
  - Add legend
- You may use D3 or Observable Plot
  - Part 1a: D3
  - Part 3 will require some D3 for treemap layout



# Project

---

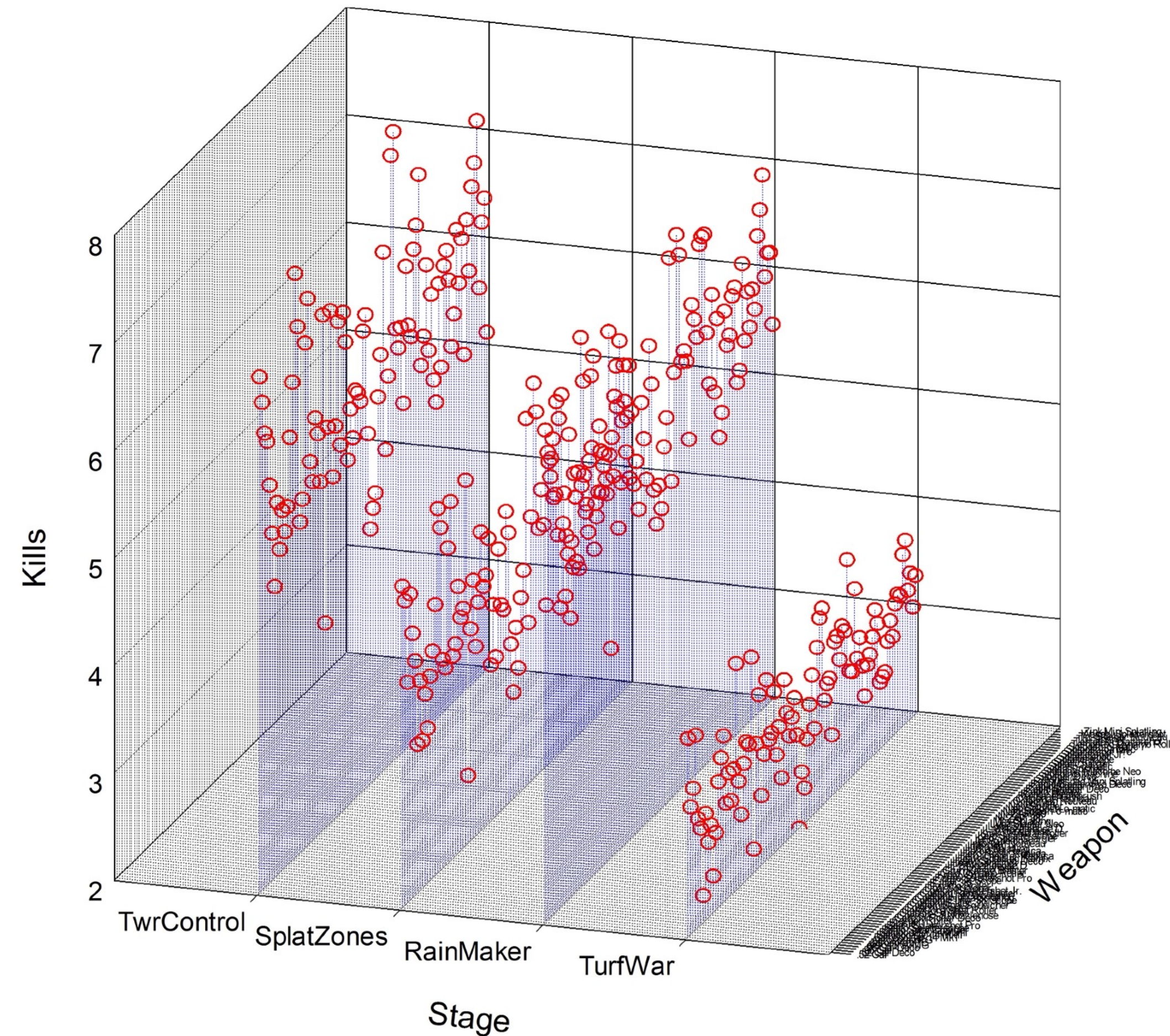
- Next steps:
  - Start thinking about the designs that help answer the questions
  - Tasks should drive your design
  - Different designs are great
    - Multiple views
    - Single view with details on demand
    - Interaction design (linked highlighting, navigation)
    - In general, don't force the user to make choices without first seeing an overview

# Guidelines for Visualization Design



# WTF Visualizations ([wtfviz.net](http://wtfviz.net))

3D Category Scatter

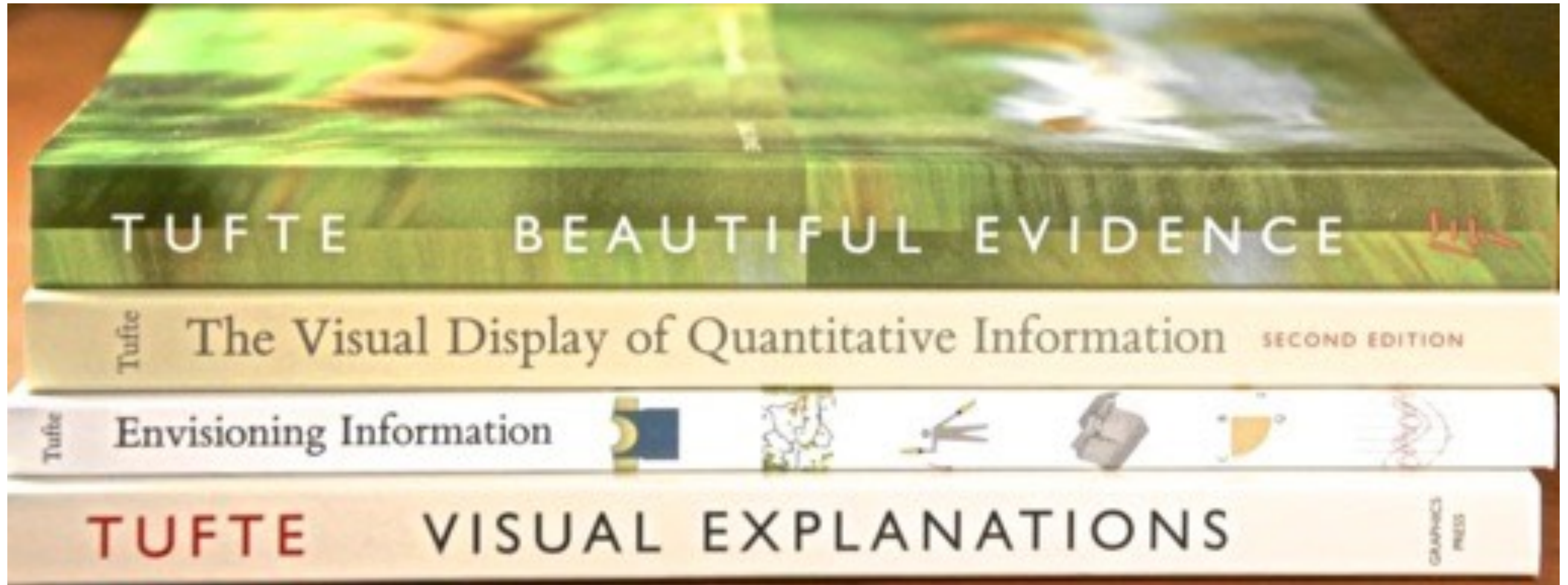


[WTF Visualizations, 2017]



# Tufte: "The da Vinci of Data" — NYTimes

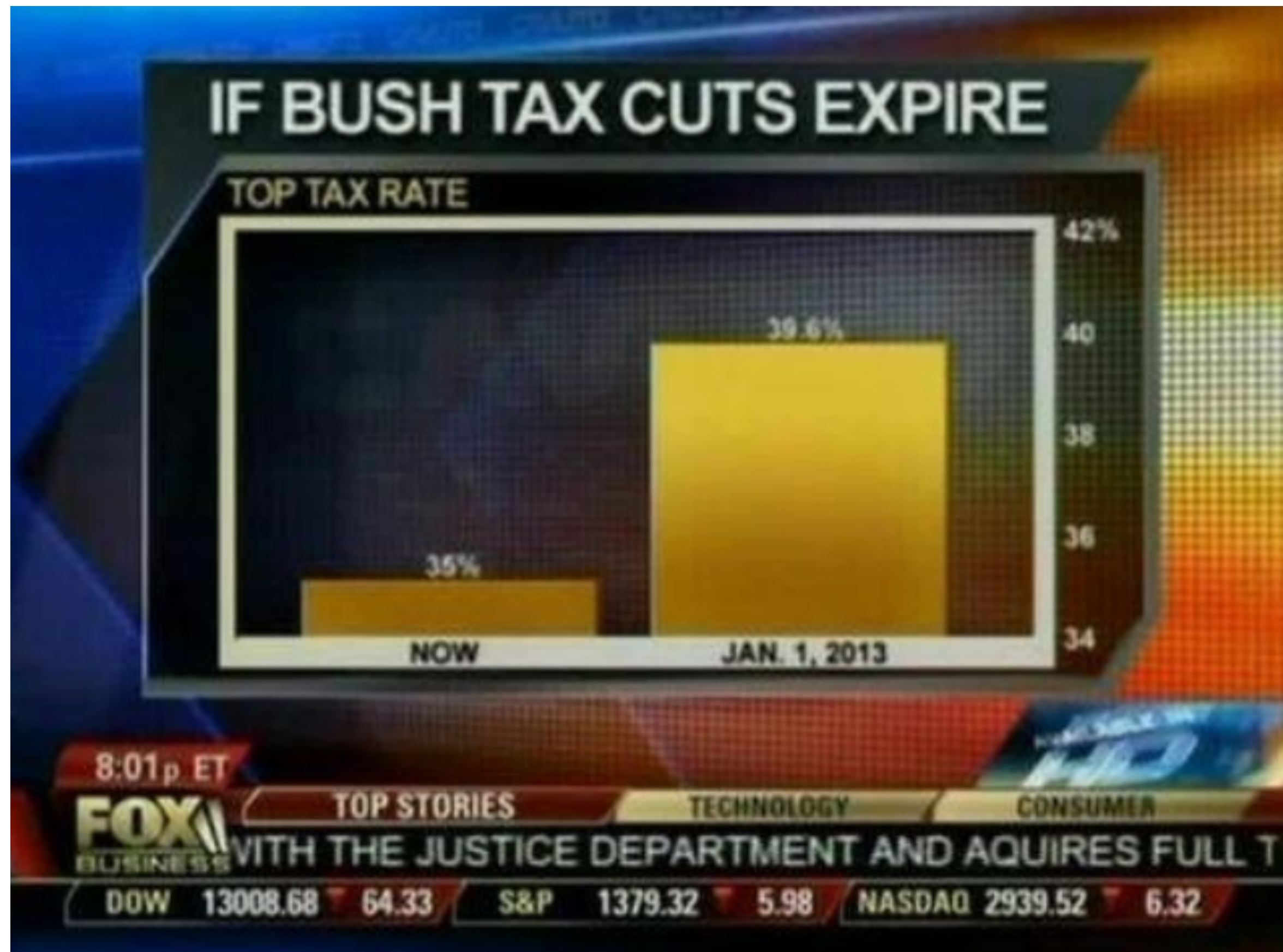
---



[<https://www.edwardtufte.com/tufte/>, 2017]



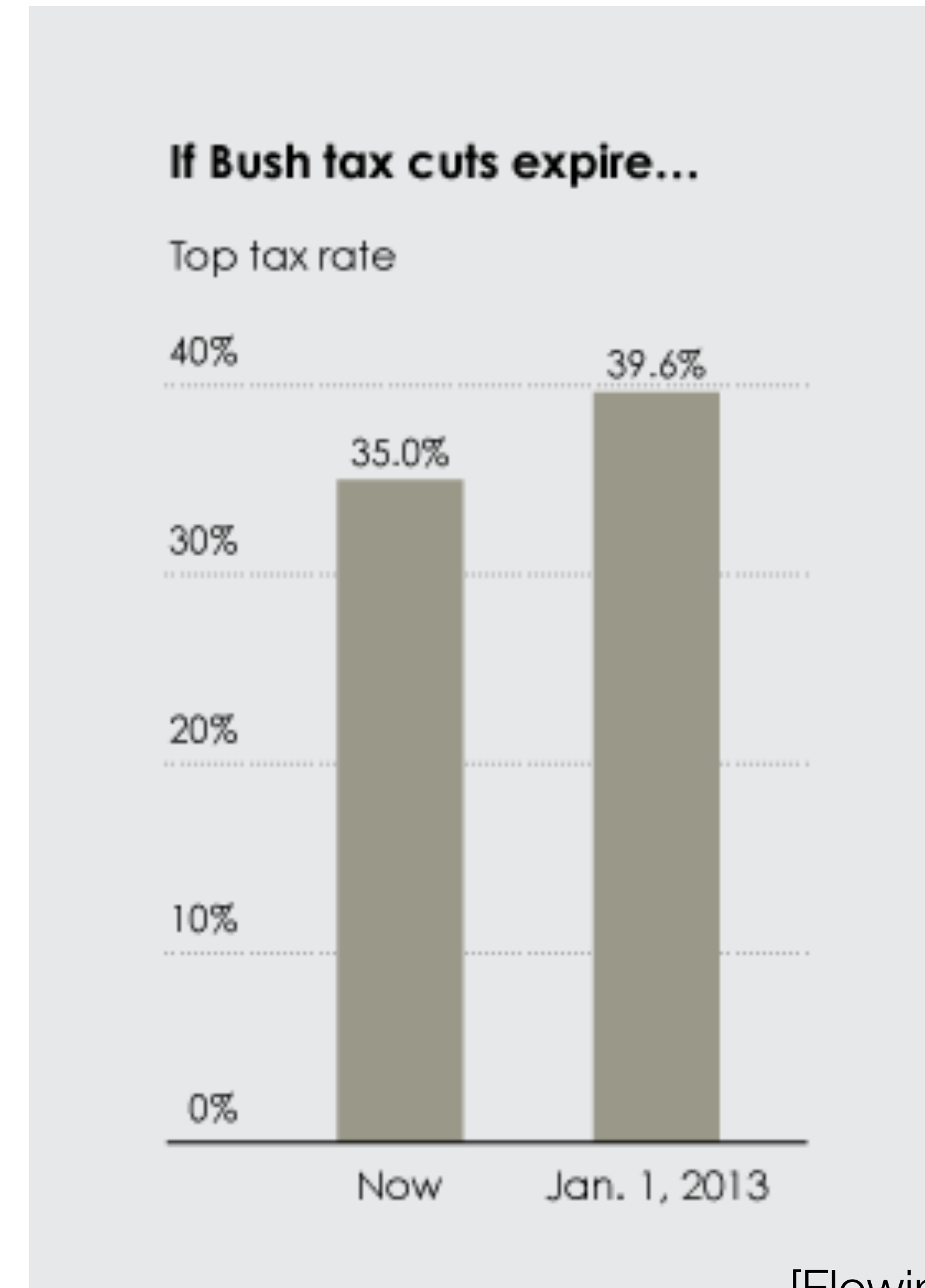
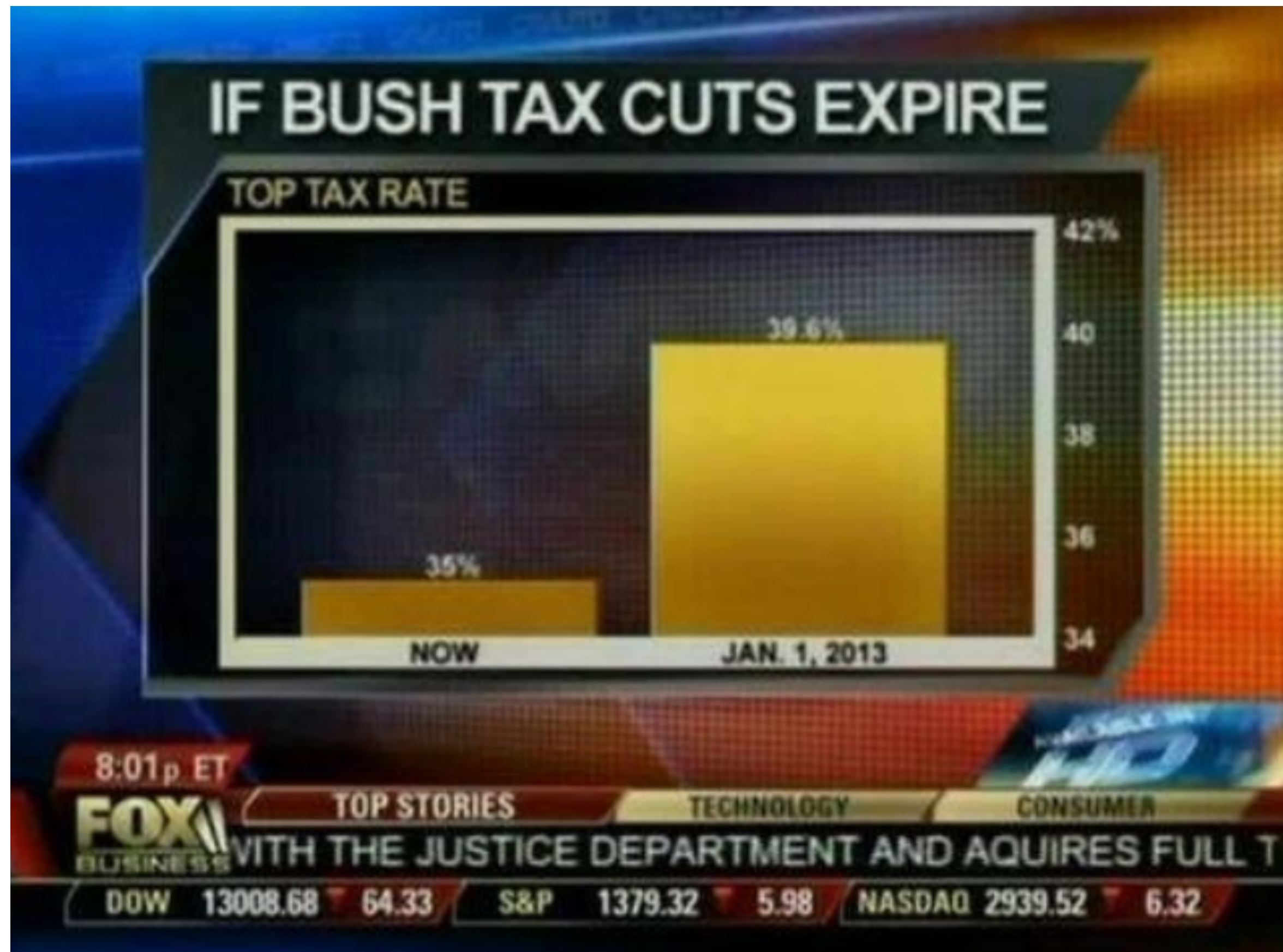
# Bad: Data magnitude $\neq$ Mark magnitude



[Flowing Data, 2012]

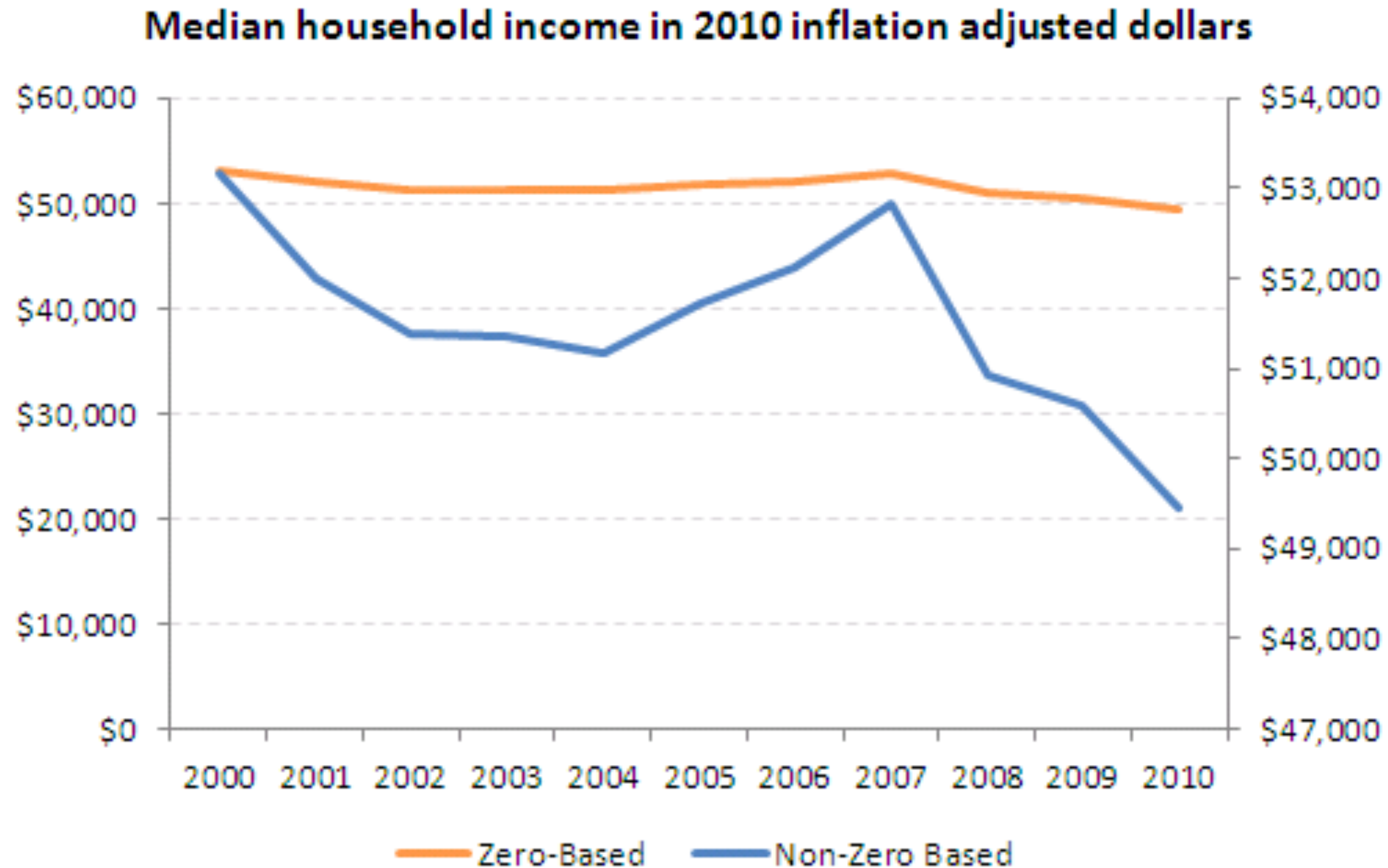


# Good: Data magnitude $\Leftarrow \Rightarrow$ Mark magnitude



[Flowing Data, 2012]

# Starting Scales at Zero?

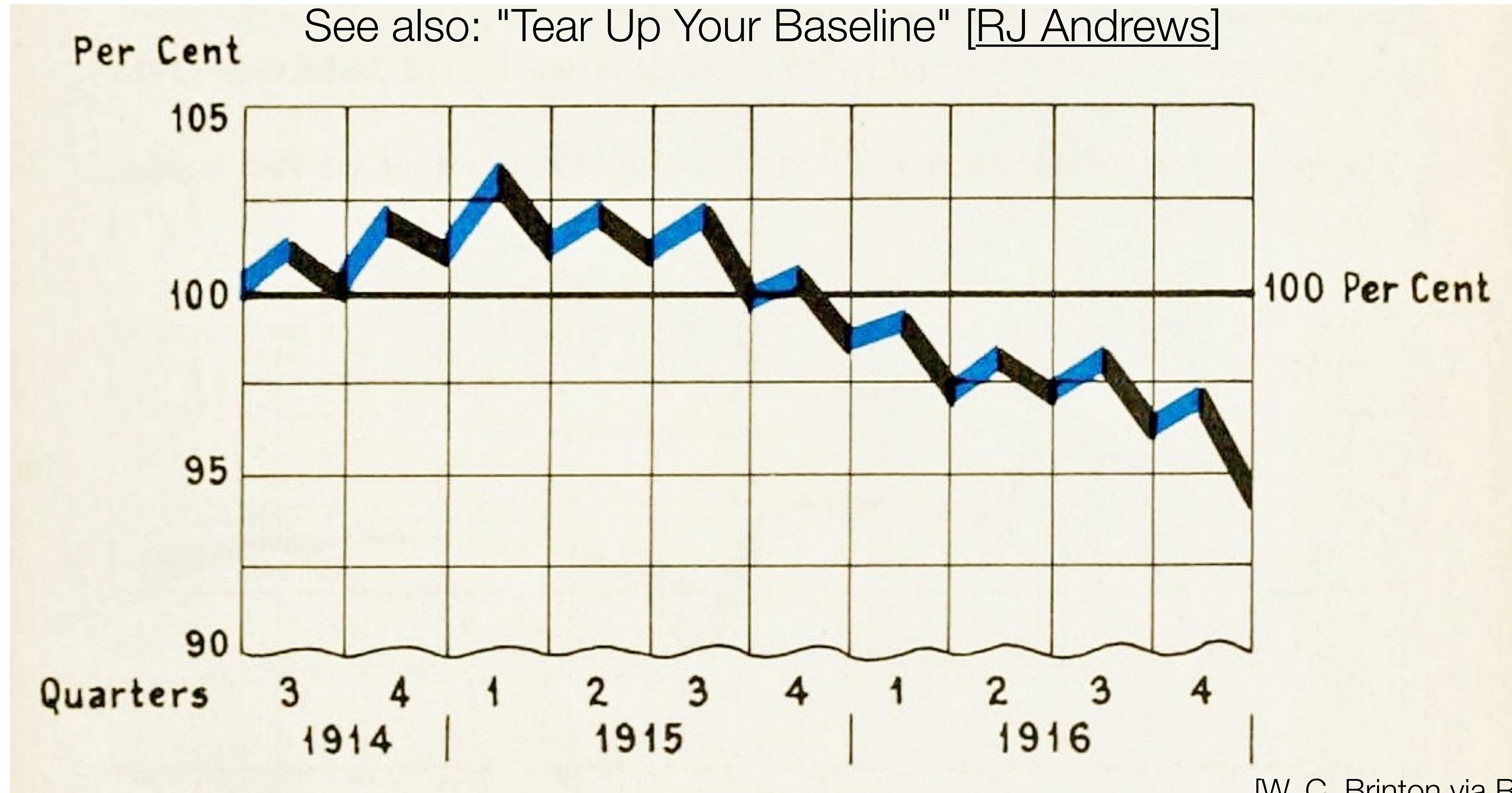


[A. Kirebel, VizWiz]



# Wavy baselines for non-zero starts

See also: "Tear Up Your Baseline" [[RJ Andrews](#)]



[W. C. Brinton via [RJ Andrews](#)]



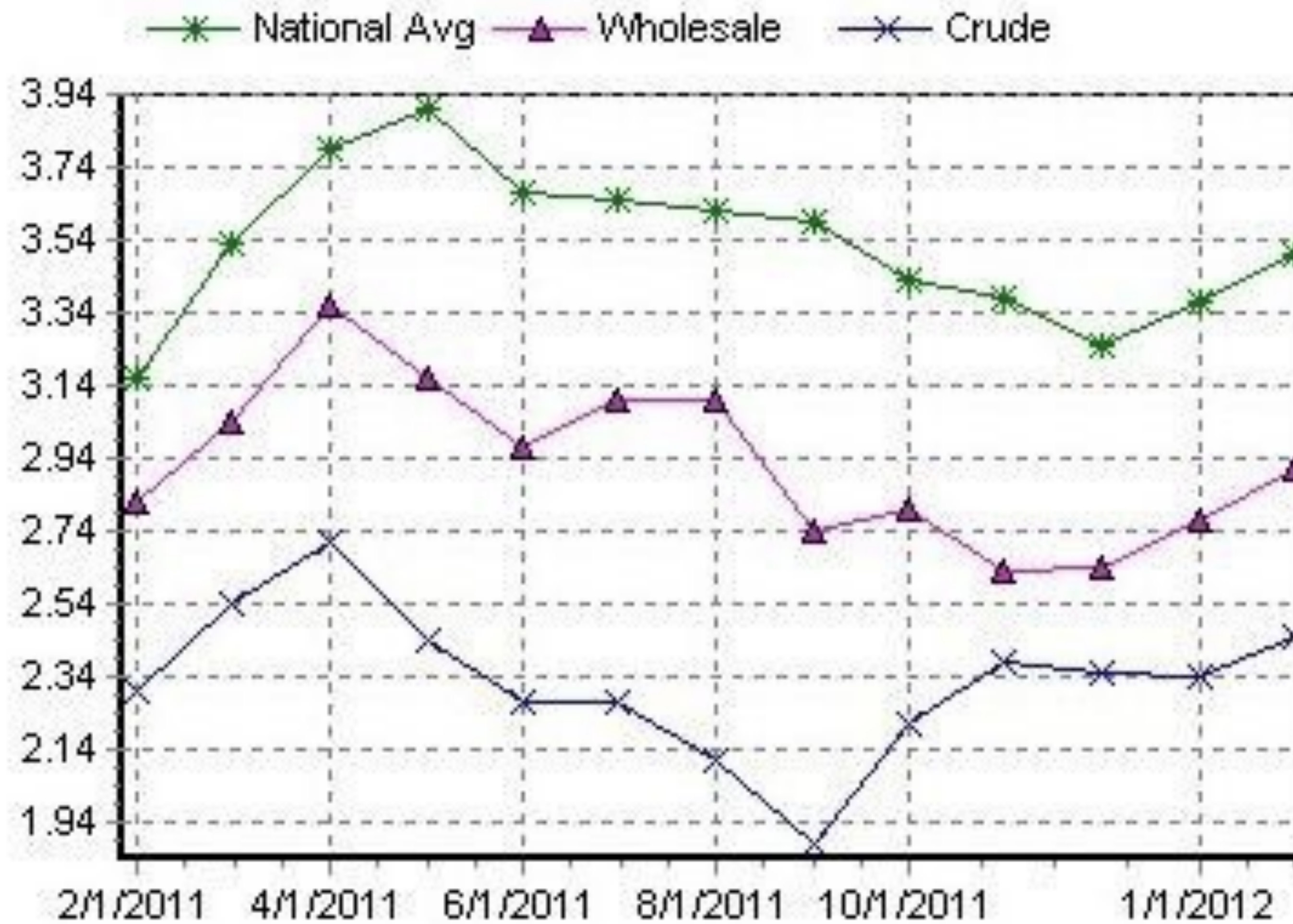
# Cherry-picking data



[Fox News via [Media Matters](#), 2012]

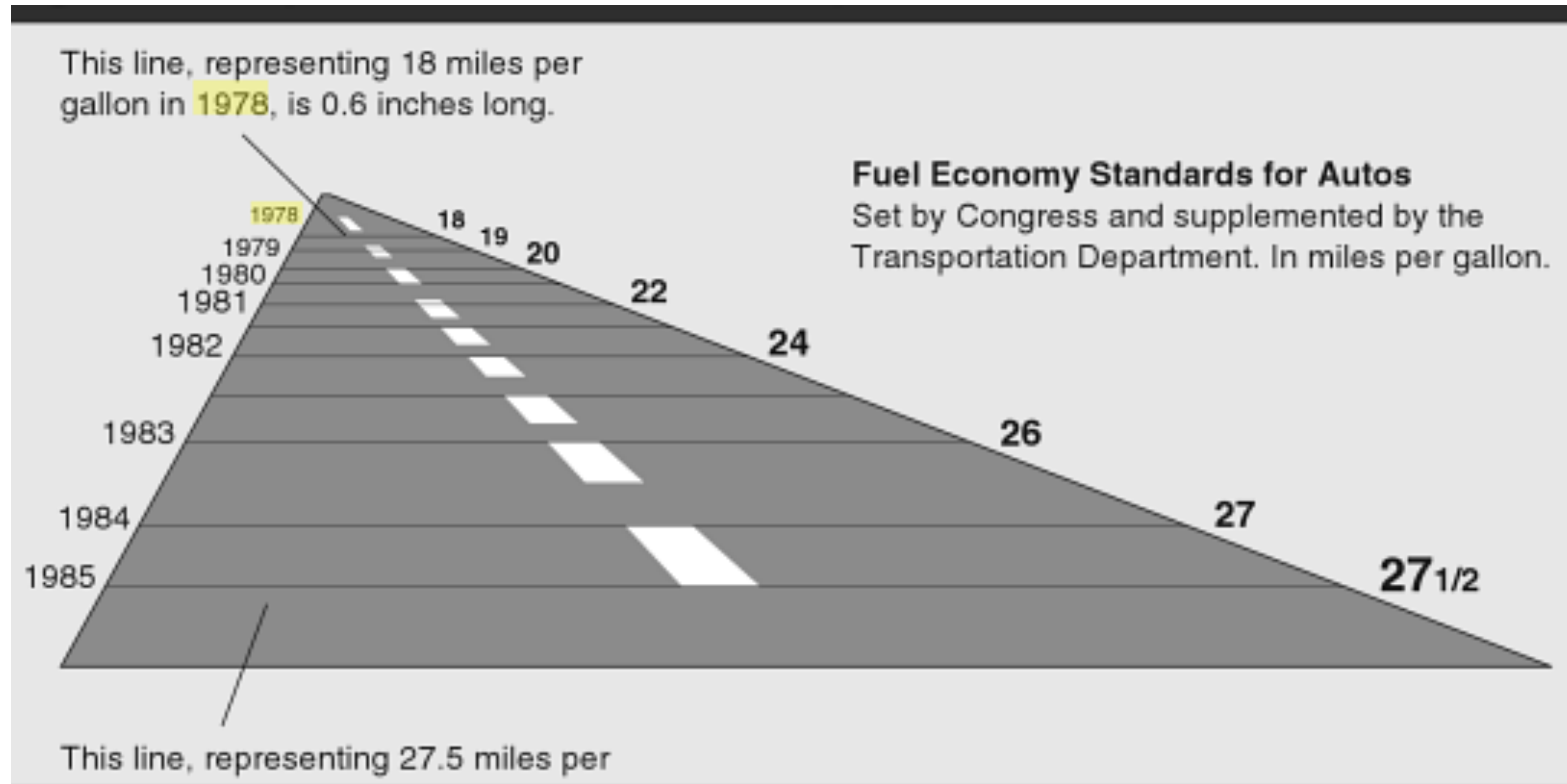


# Show **all** the data



[AAA via [Media Matters](#), 2012]

# Tufte's Lie Factor



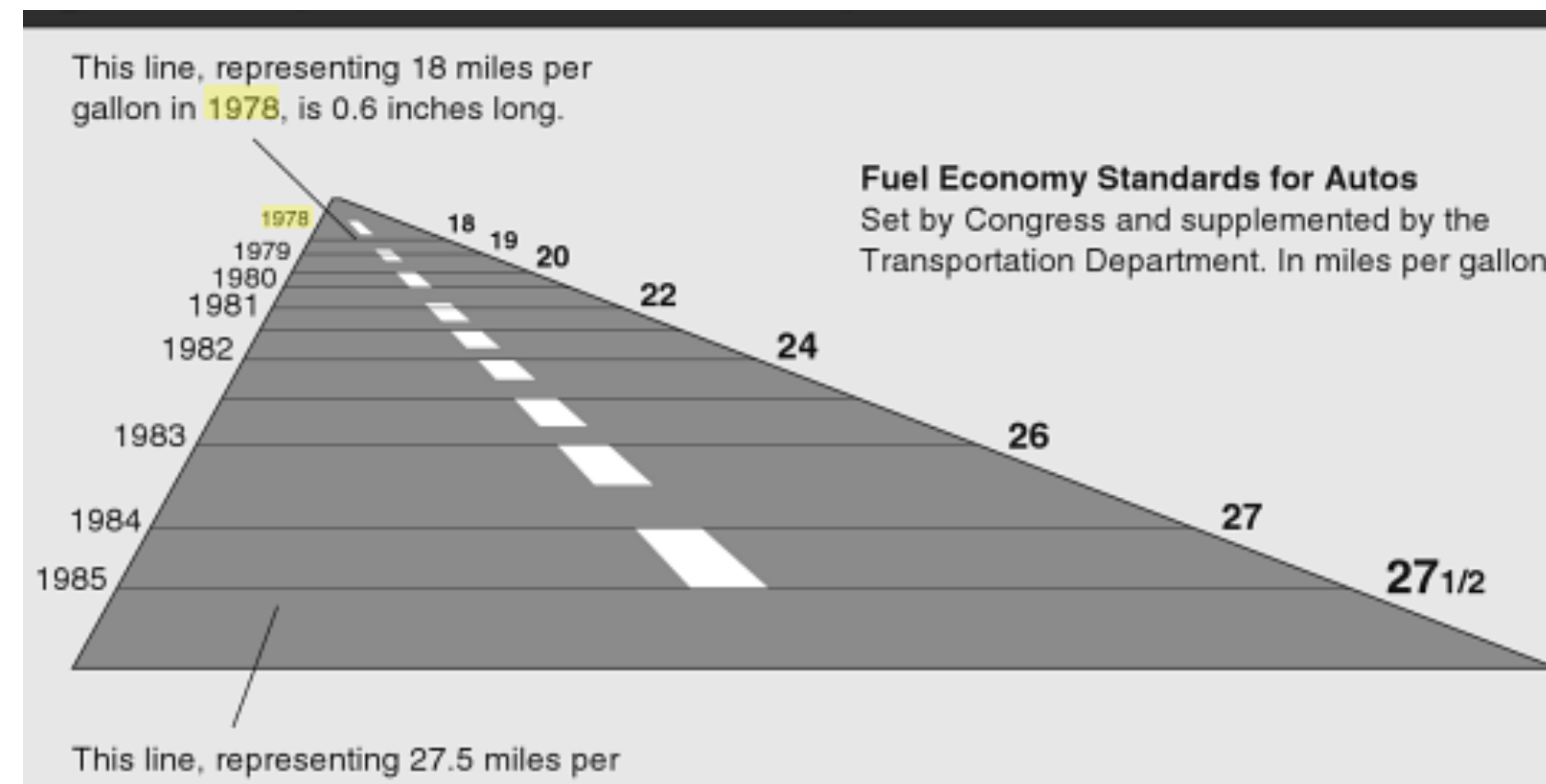
[NYTimes via Tufte, 1991]



# Tufte's Lie Factor

- Size of effect = (2nd value - 1st value) / (1st value)
- Lie factor = (size of effect in graphic) / (size of effect in data)
- In the graphic:

$$\text{Lie Factor} = \frac{\frac{5.3 - 0.6}{0.6}}{\frac{27.5 - 18}{18}} = 14.8$$

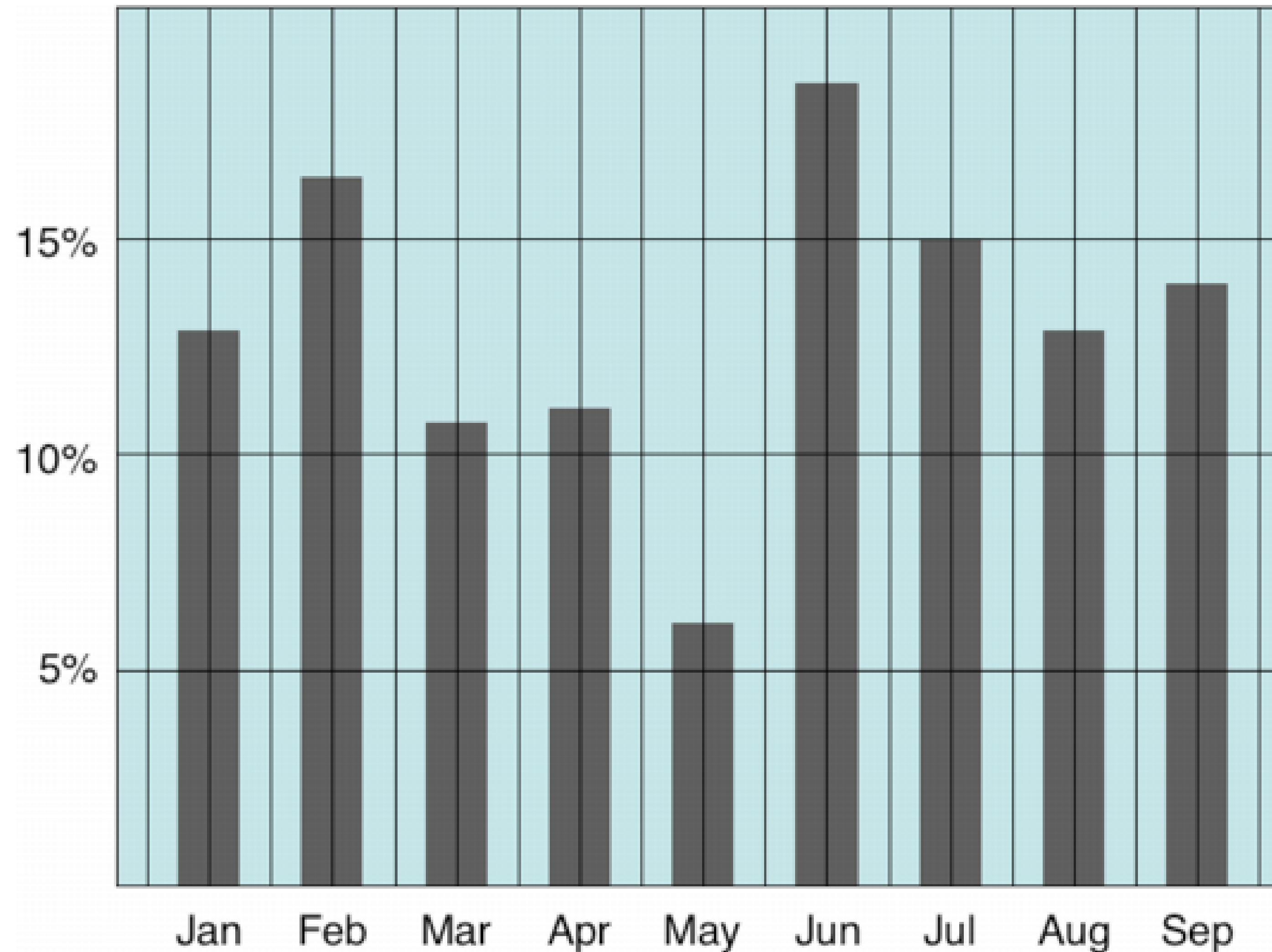


# (Some of) Tufte's Integrity Principles

---

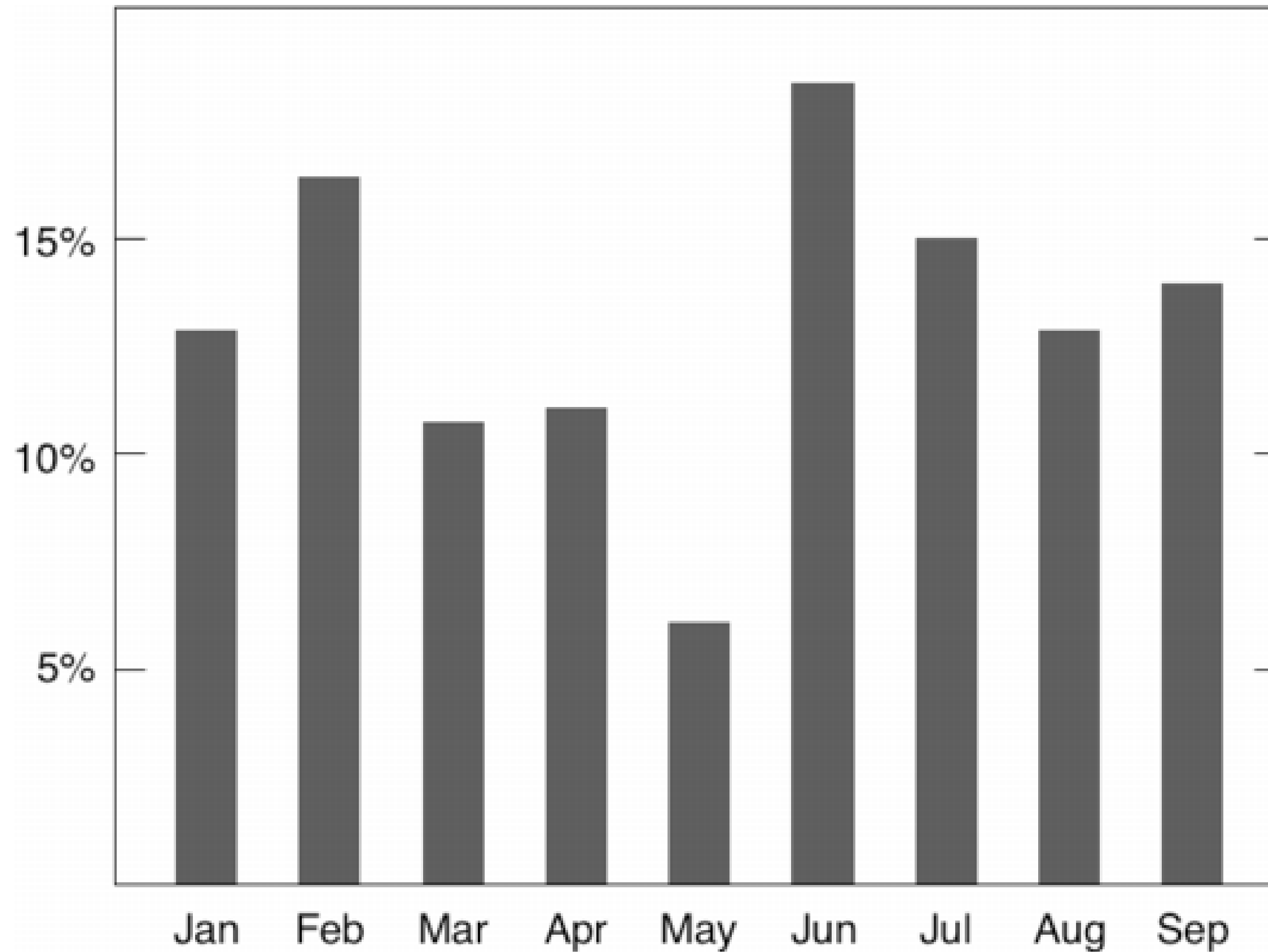
- Show data variation, not design variation
- Clear, detailed, and thorough labeling and appropriate scales
- Size of the graphic effect should be directly proportional to the numerical quantities ("lie factor")

# Avoid Chartjunk



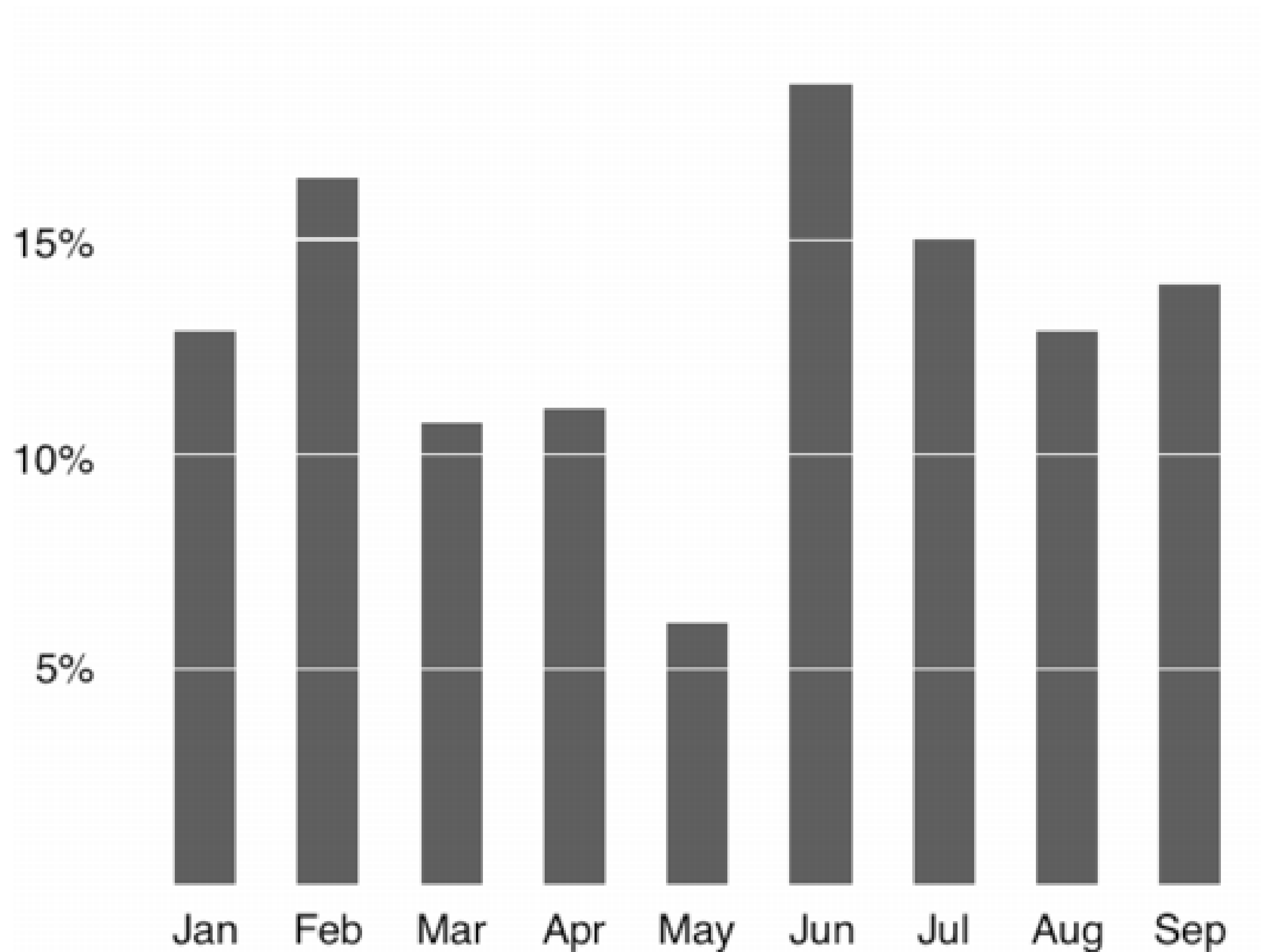
[T. Brey via [A. Lex](#)]

# Avoid Chartjunk



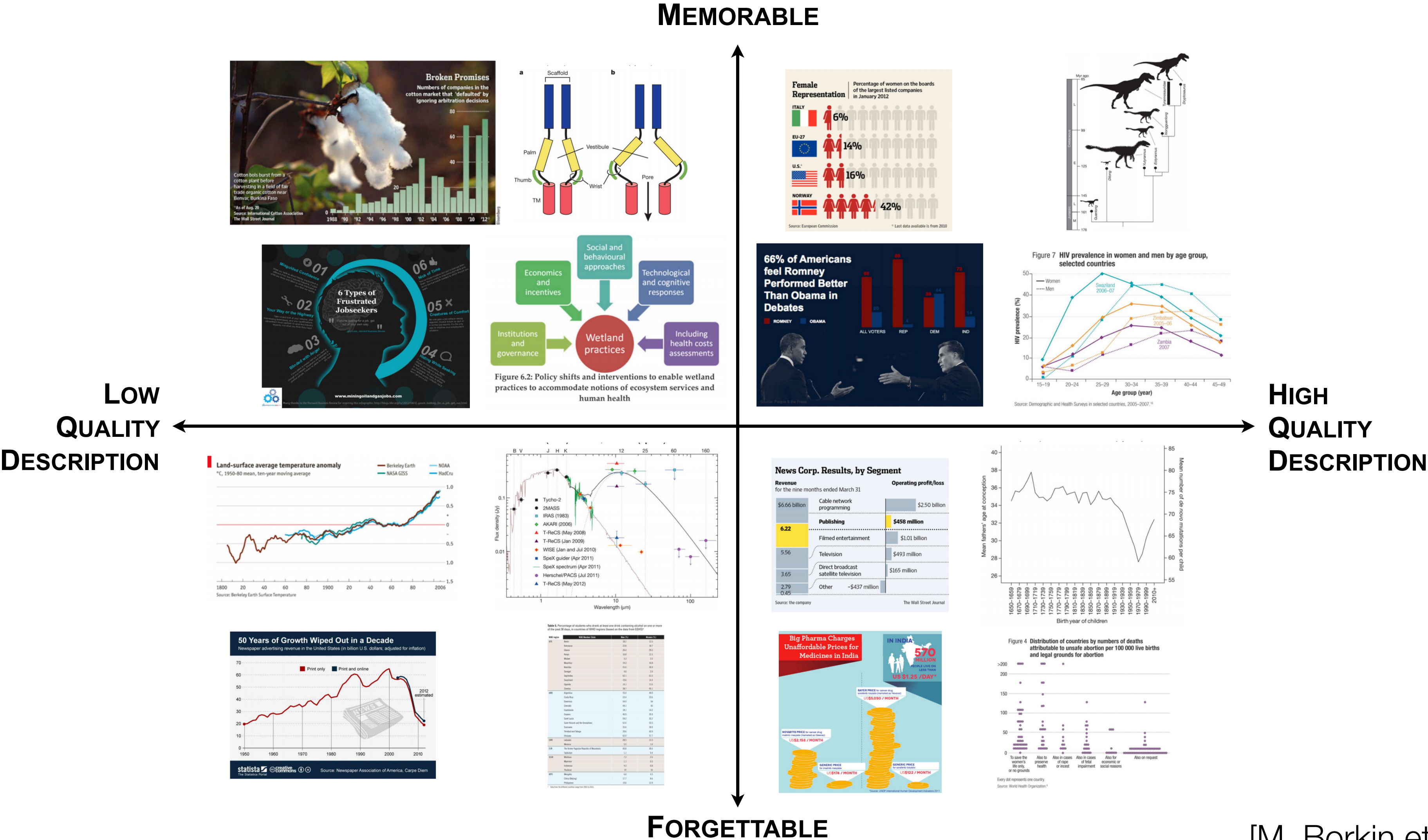
[T. Brey via [A. Lex](#)]

# Avoid Chartjunk



[T. Brey via [A. Lex](#)]

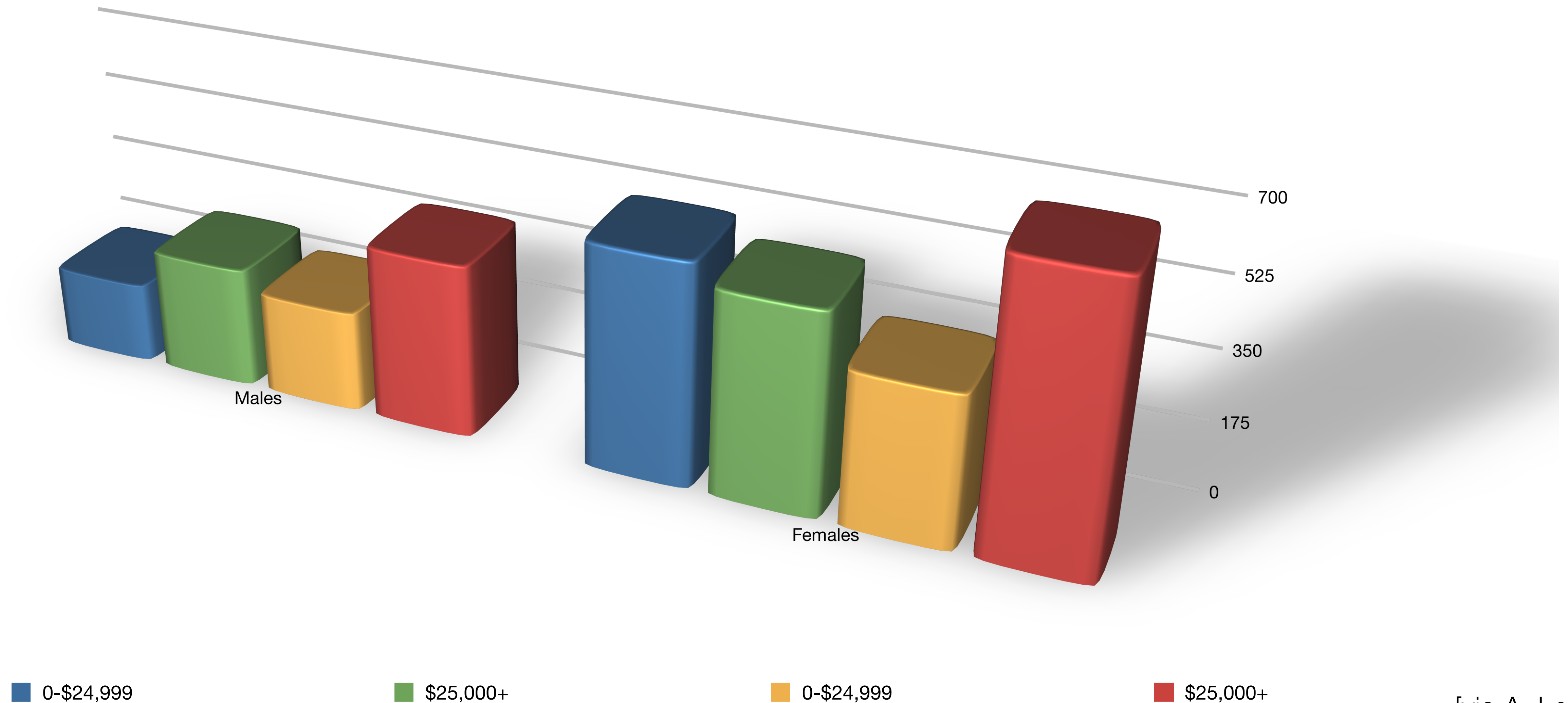
# Avoid Chartjunk?



[M. Borkin et al., InfoVis 2015]

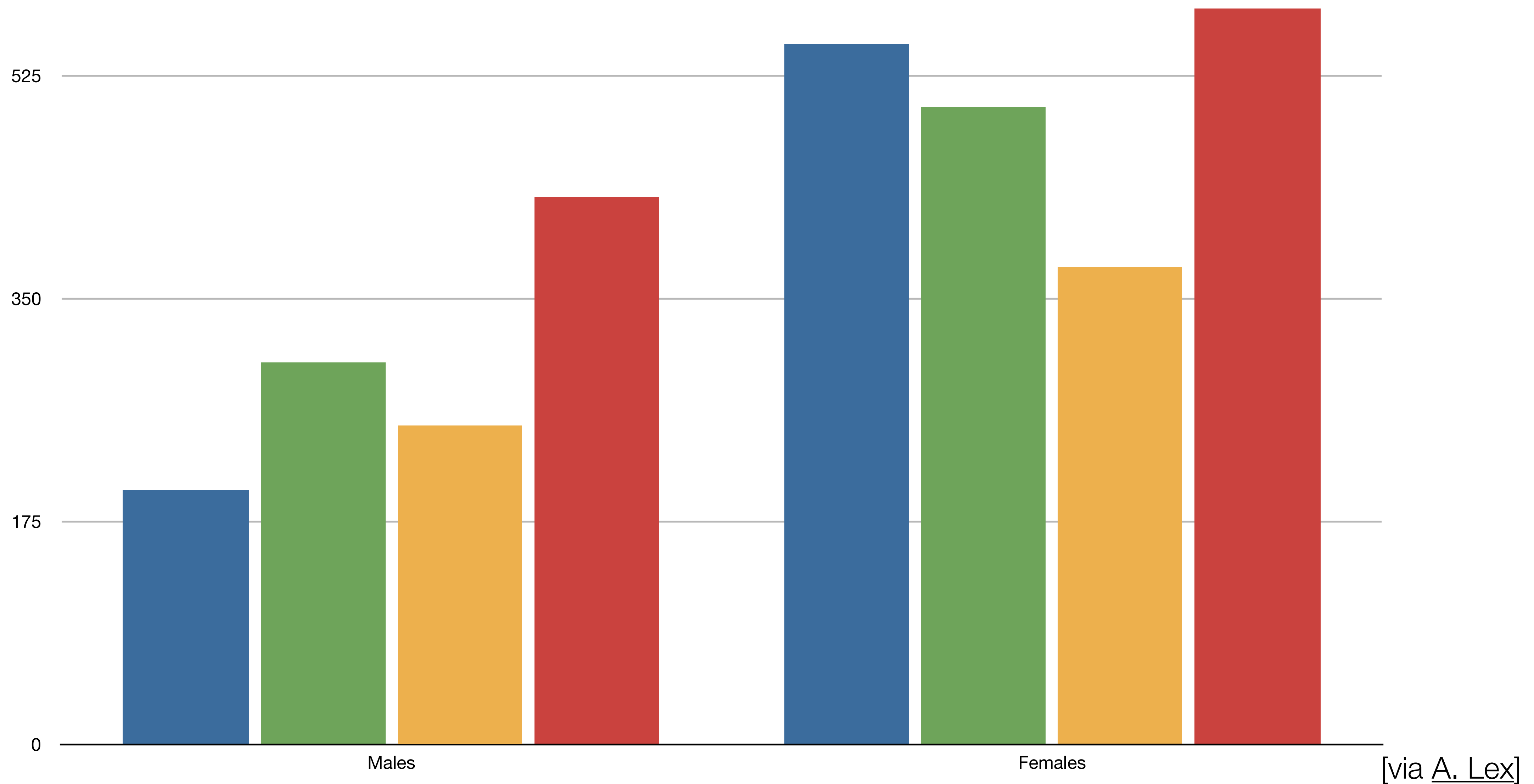


# Data-to-Ink Ratio (Also Unjustified 3D)

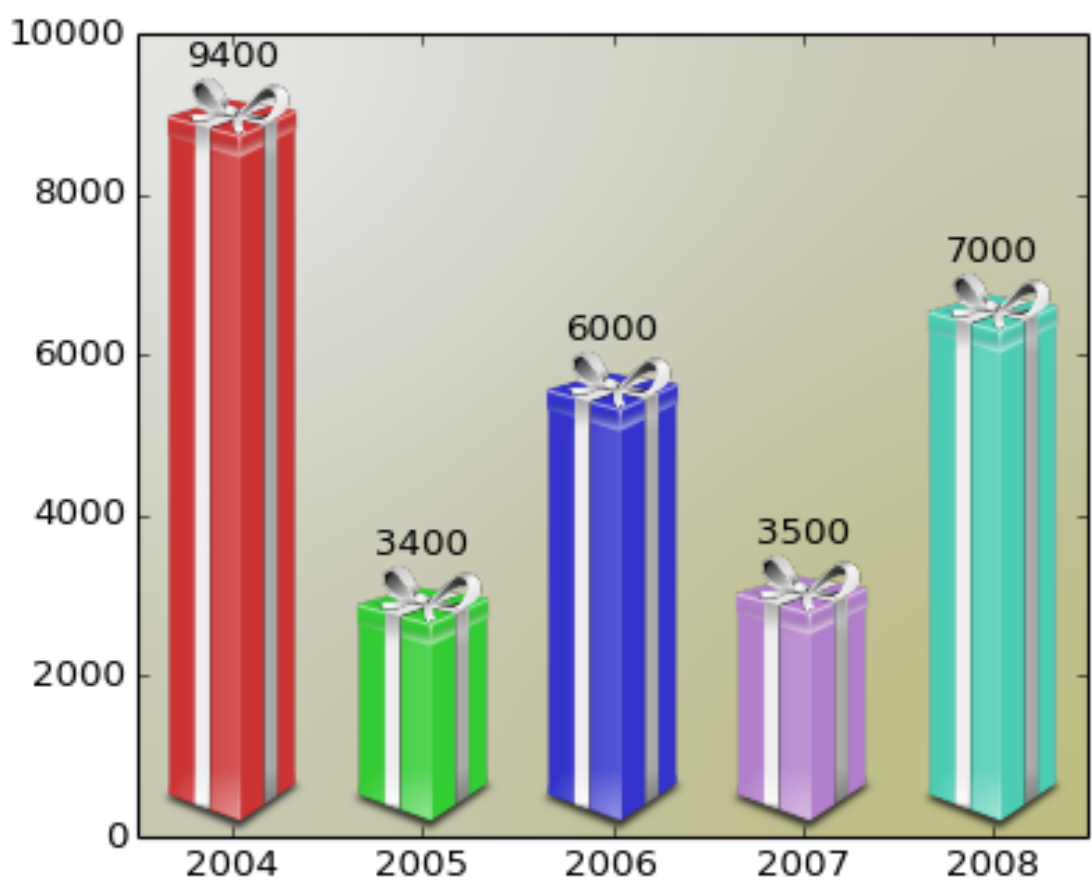
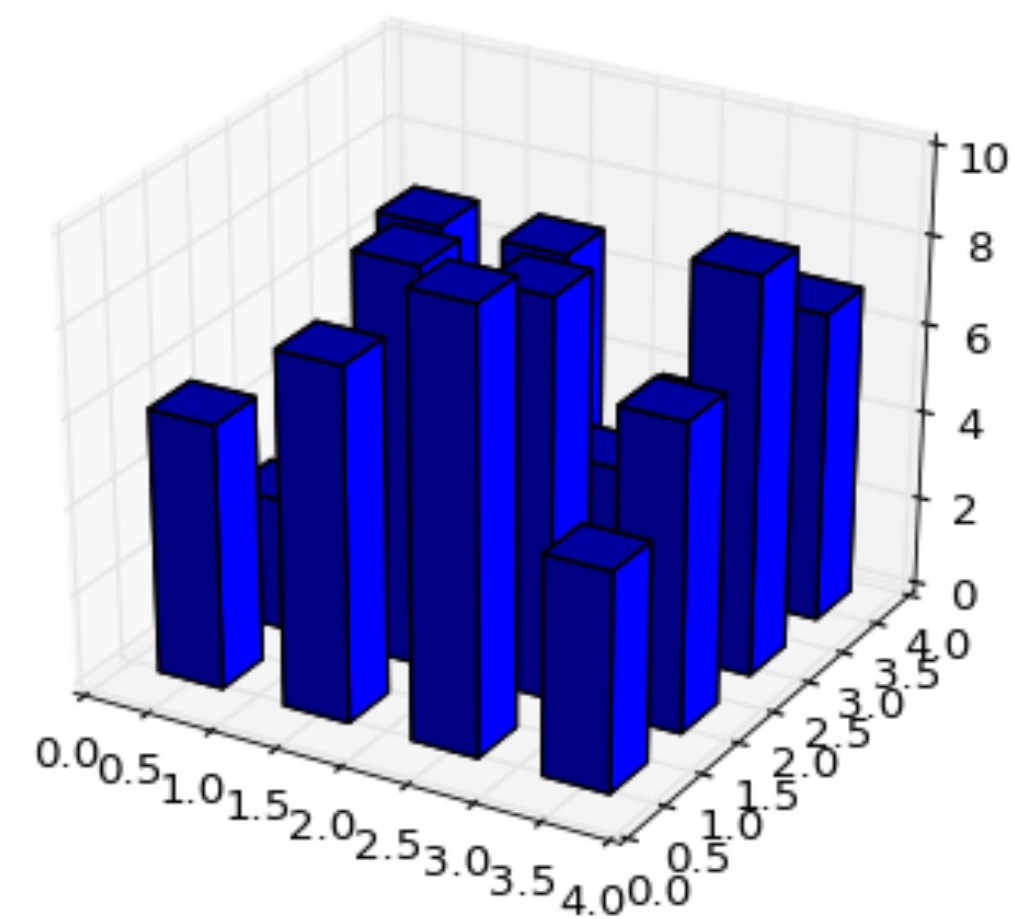


[via A. Lex]

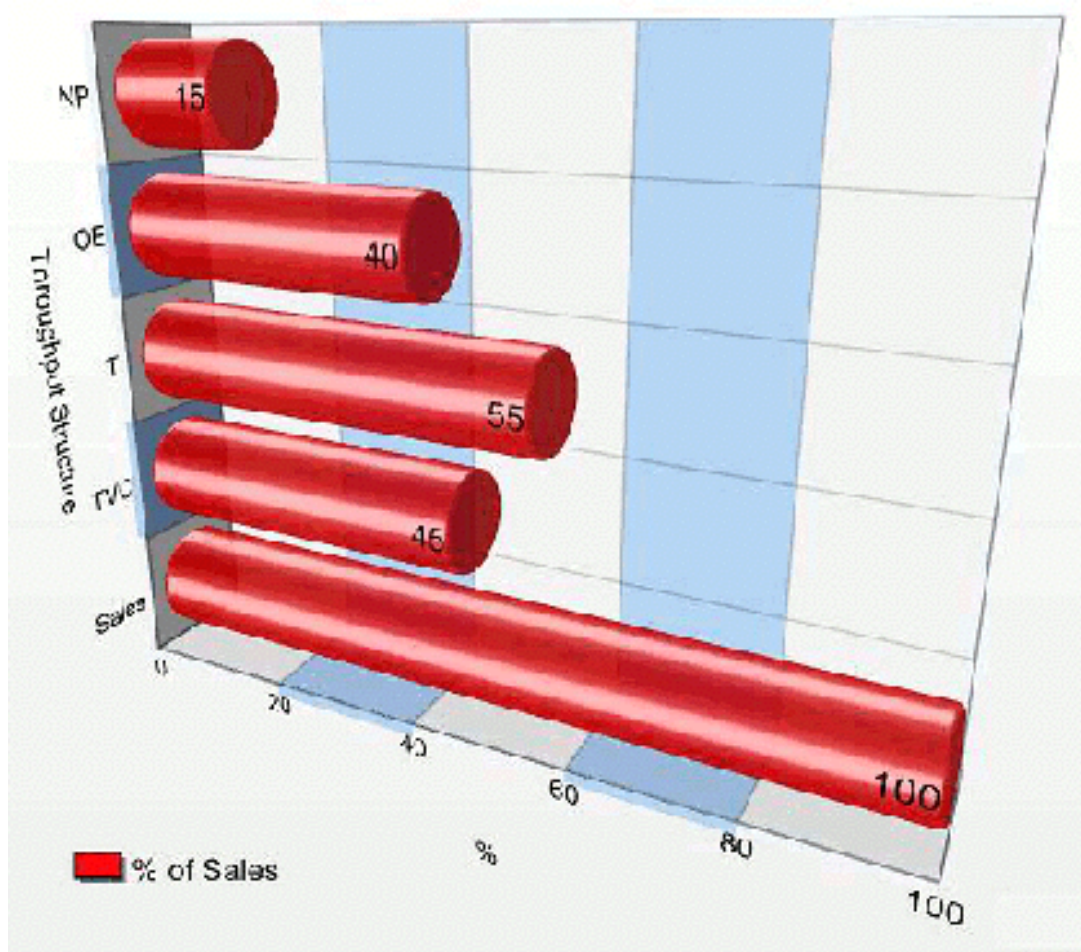
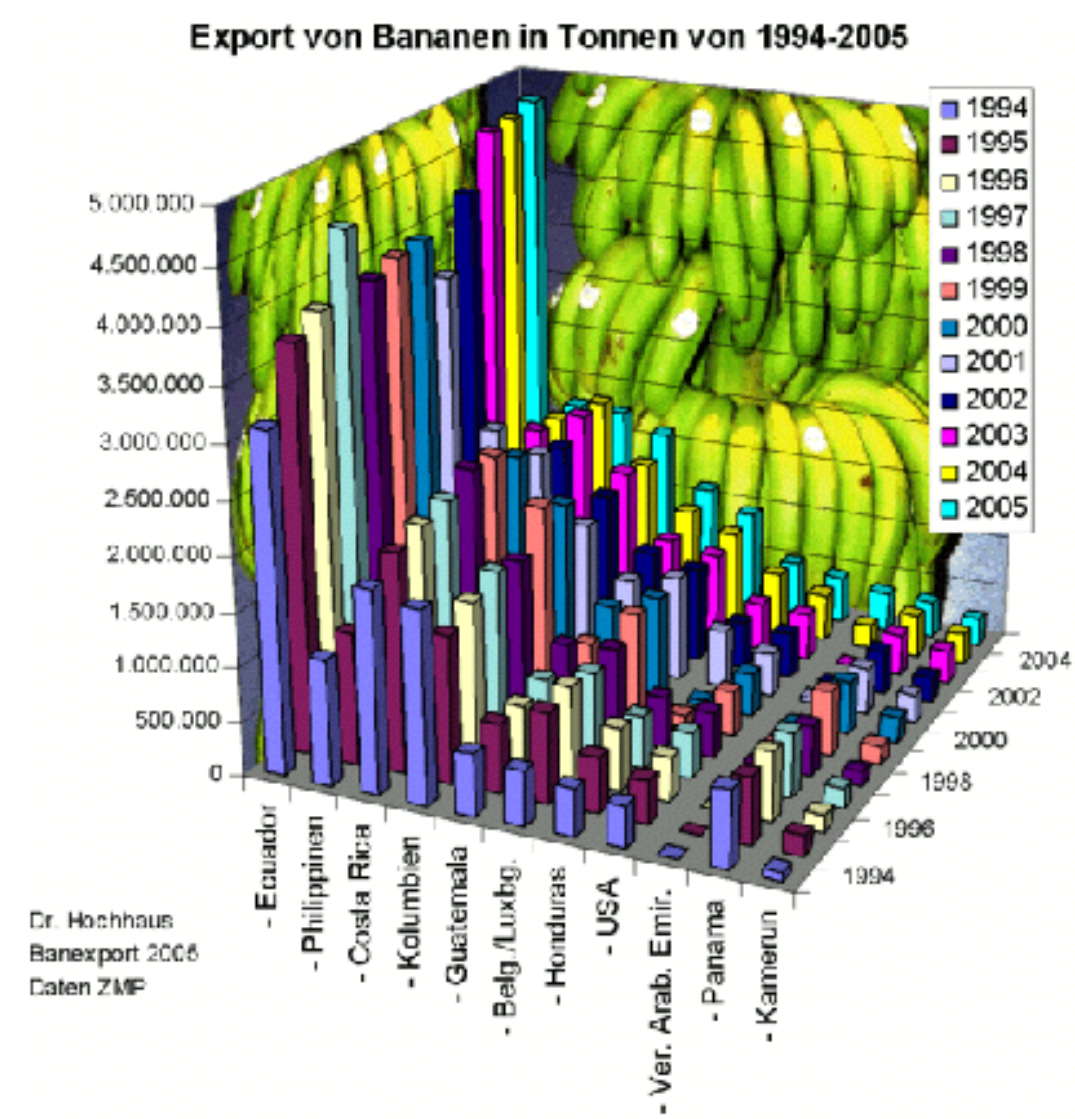
# Maximize Data-to-Ink Ratio



# No Unjustified 3D



matplotlib gallery



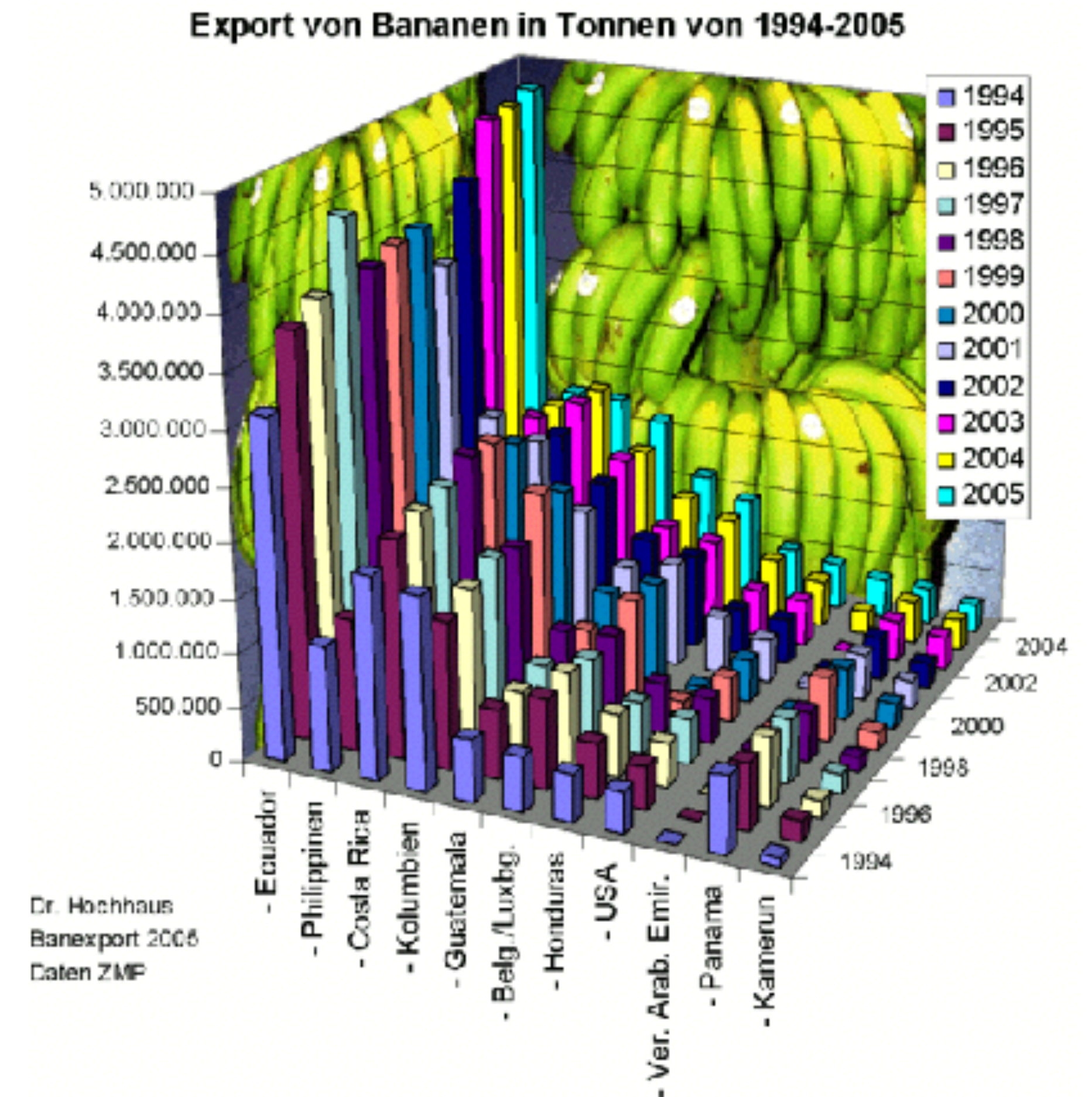
Excel Charts Blog

[via A. Lex]



# No Unjustified 3D

- Occlusion hides information
- Perspective distortion dangers
- Tilted text isn't legible
- Can **help** with shape perception



[via A. Lex]

# Eyes Beat Memory

---

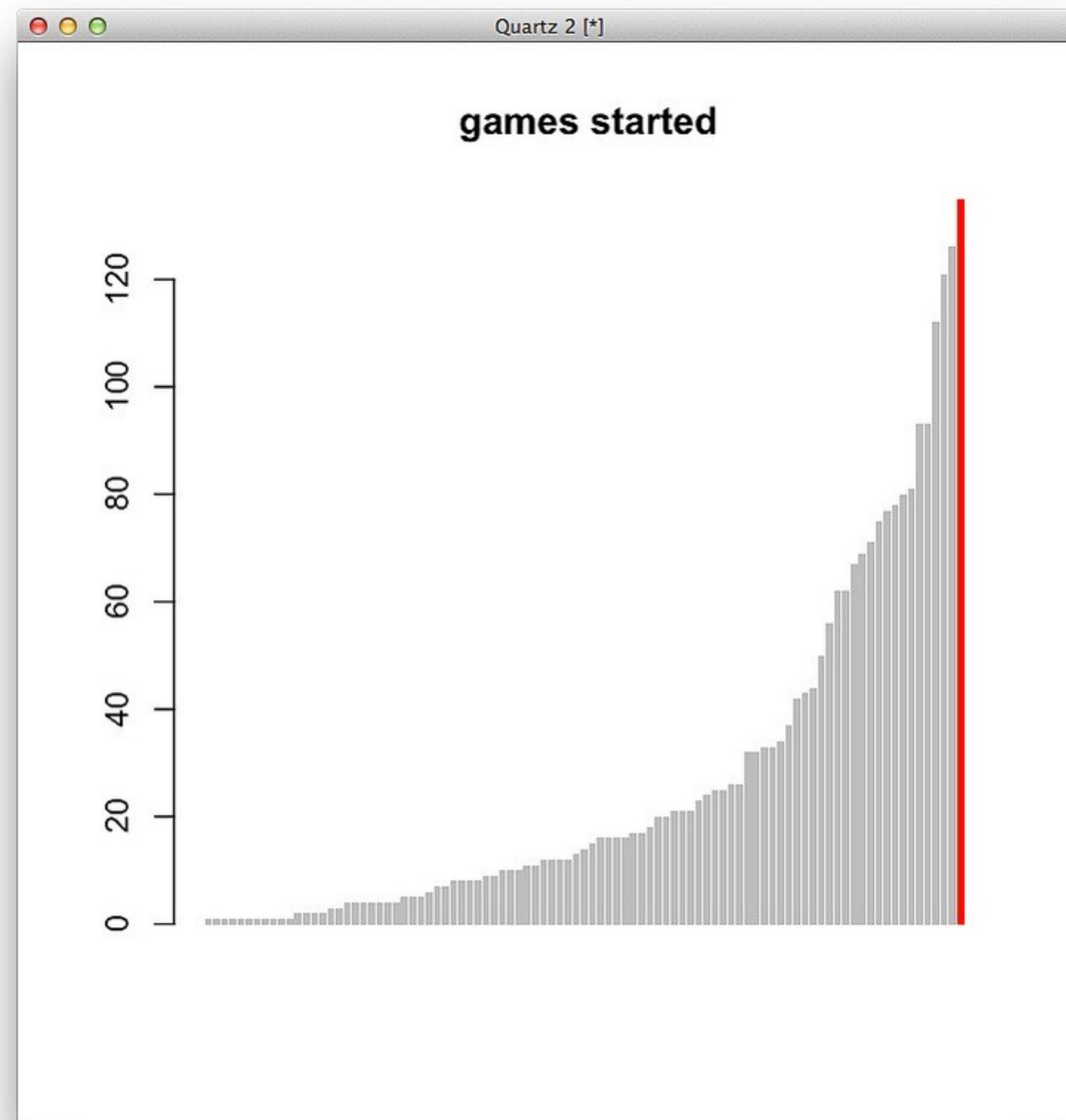
- Reduce cognitive load (using up working memory)
- Animation versus side-by-side views
- Change blindness

“Computer-based visualization systems provide visual representations of datasets **designed** to help people carry out tasks more effectively.”

— T. Munzner



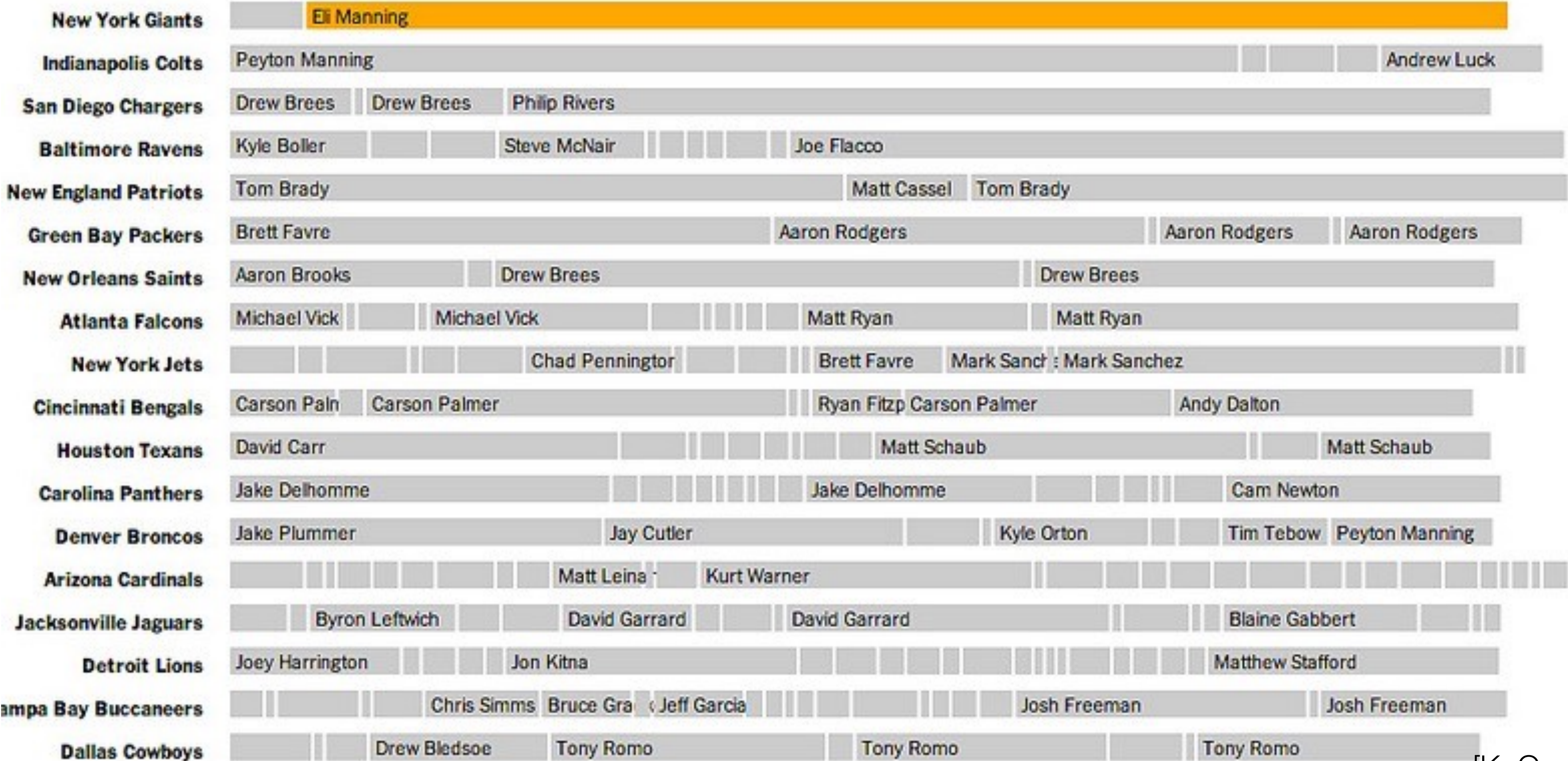
# Design Iteration



[K. Quealy, 2013]



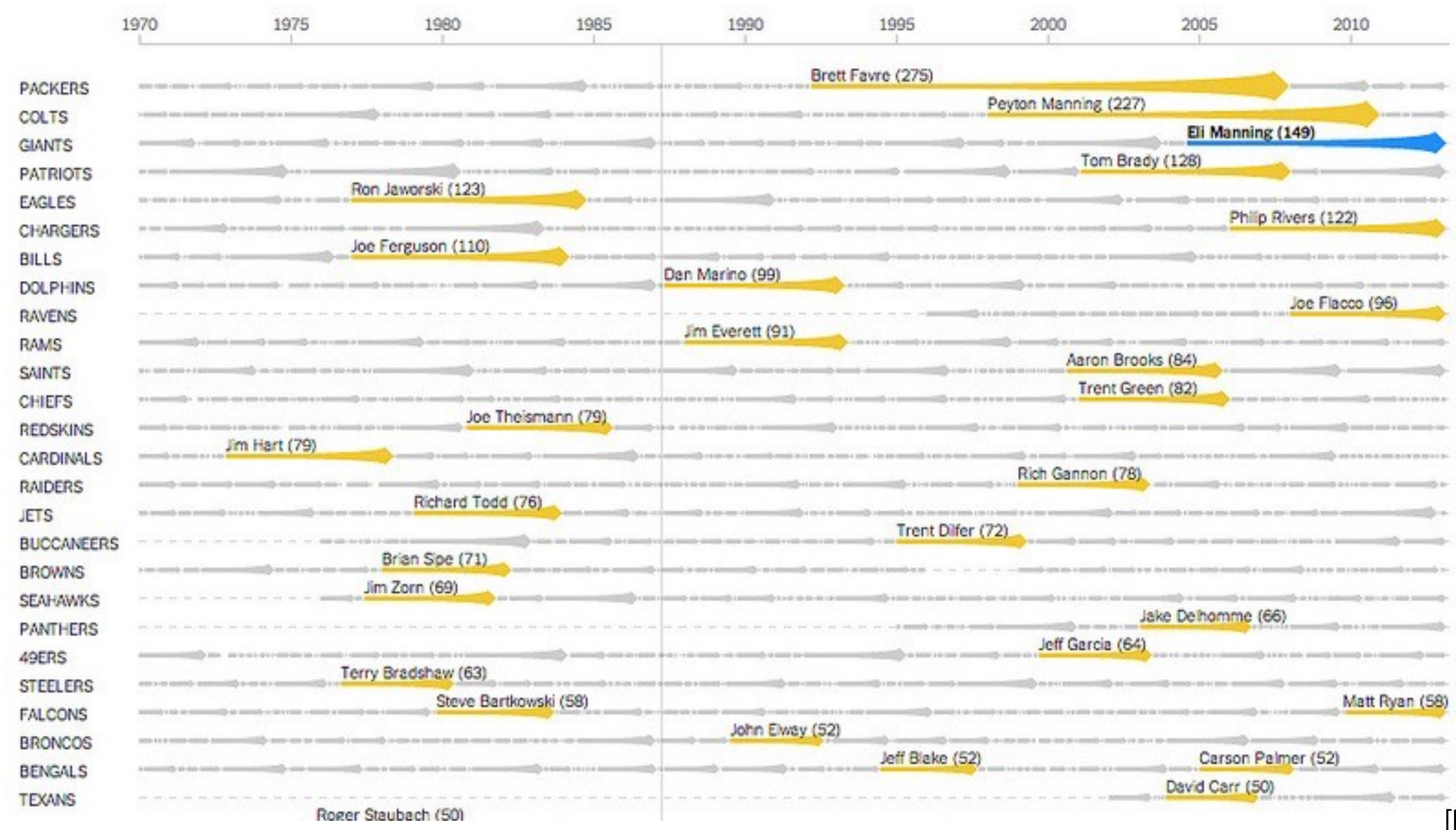
# Design Iteration



[K. Quealy, 2013]



# Design Iteration



[K. Quealy, 2013]





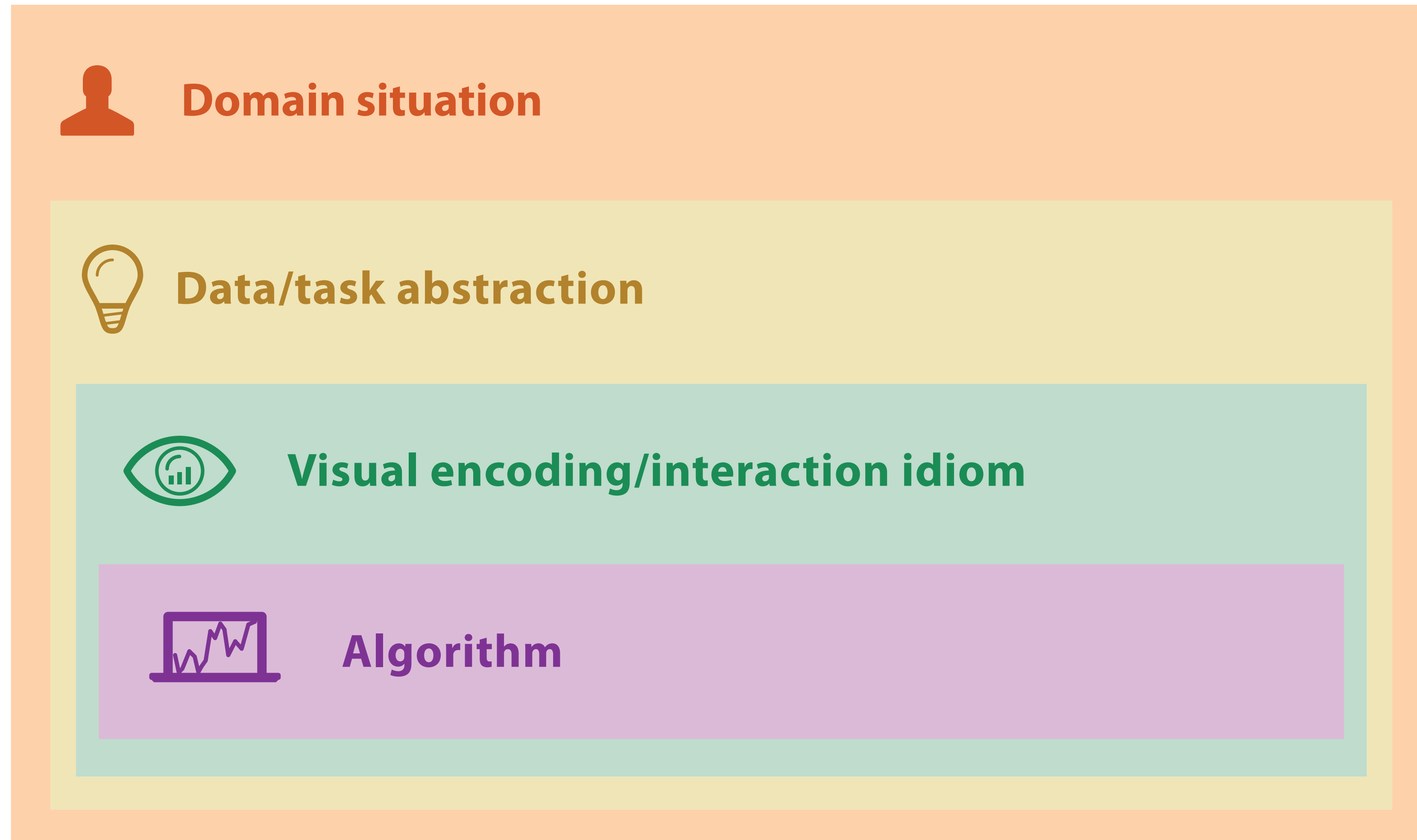
# Design

---

- Unlike a math problem, there are many different approaches for the visualization of some data
- Need to have some way to discuss how to determine whether a visualization is doing what we want
- Validation: Understand why a design is effective
  - What problems can be effective
  - Do this at different levels

# Four Nested Levels of Design


---




[Munzner, 2014]



# Potential problems at each level

 **Domain situation**  
You misunderstood their needs

 **Data/task abstraction**  
You're showing them the wrong thing

 **Visual encoding/interaction idiom**  
The way you show it doesn't work

 **Algorithm**  
Your code is too slow

[Munzner, 2014]

# Validation at each level

❗ **Threat** Wrong problem

☑ **Validate** Observe and interview target users

❗ **Threat** Wrong task/data abstraction

❗ **Threat** Ineffective encoding/interaction idiom

☑ **Validate** Justify encoding/interaction design

❗ **Threat** Slow algorithm

☑ **Validate** Analyze computational complexity

Implement system

☑ **Validate** Measure system time/memory

☑ **Validate** Qualitative/quantitative result image analysis

*Test on any users, informal usability study*

☑ **Validate** Lab study, measure human time/errors for task

☑ **Validate** Test on target users, collect anecdotal evidence of utility

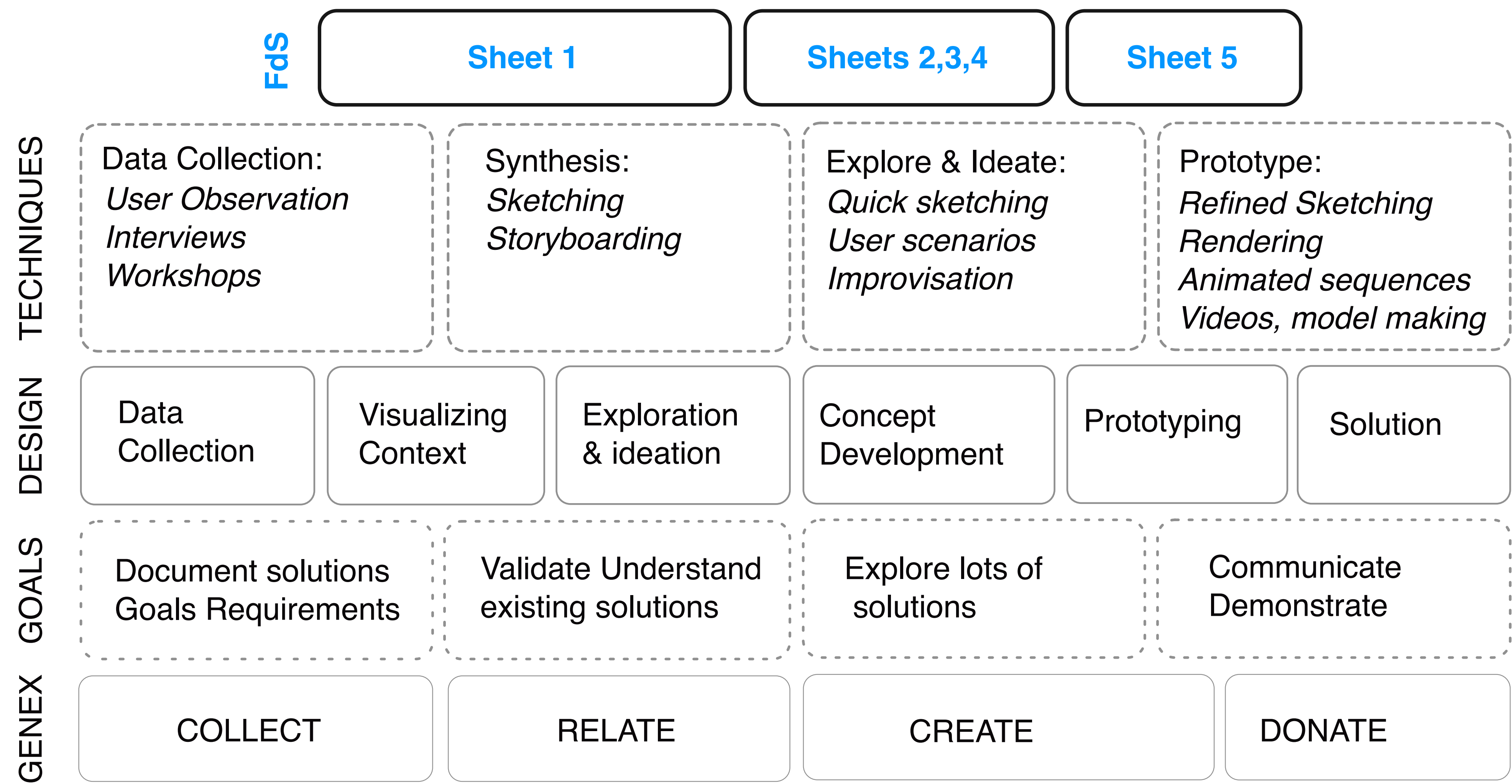
☑ **Validate** Field study, document human usage of deployed system

☑ **Validate** Observe adoption rates

[Munzner, 2014]



# Five Design-Sheet Methodology



[J. Roberts et al., 2016]

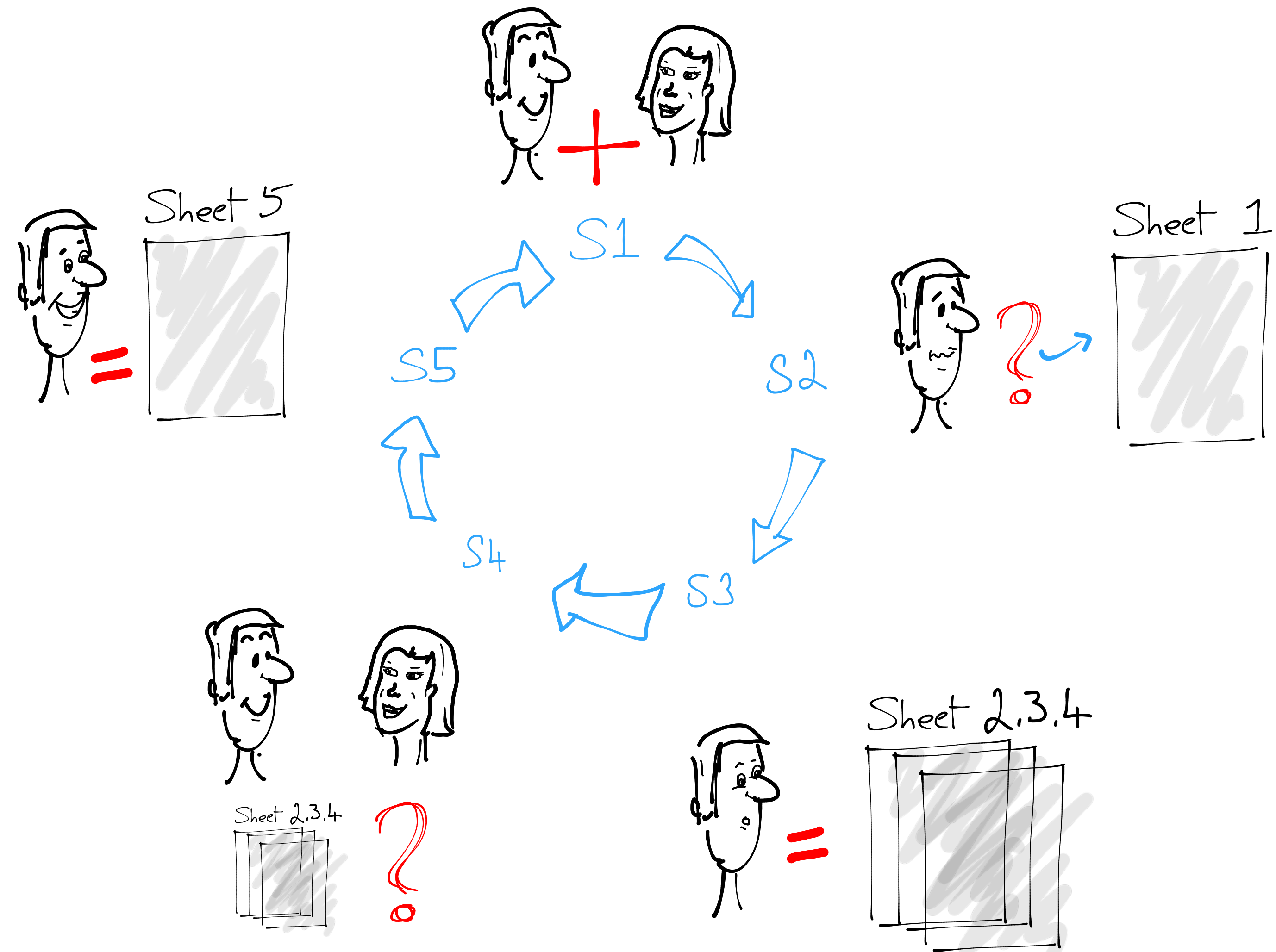
# Five Stages

---

1. Meet with client and consider task; or contemplate task on own.
2. Ideate and sketch small ideas.
3. Sketch and plan three alternative designs.
4. Consider solutions with client; or deliberate on own.
5. Generate realization sheet, and implement prototype. Discuss with client and re-iterate if necessary.

[J. Roberts et al., 2016]

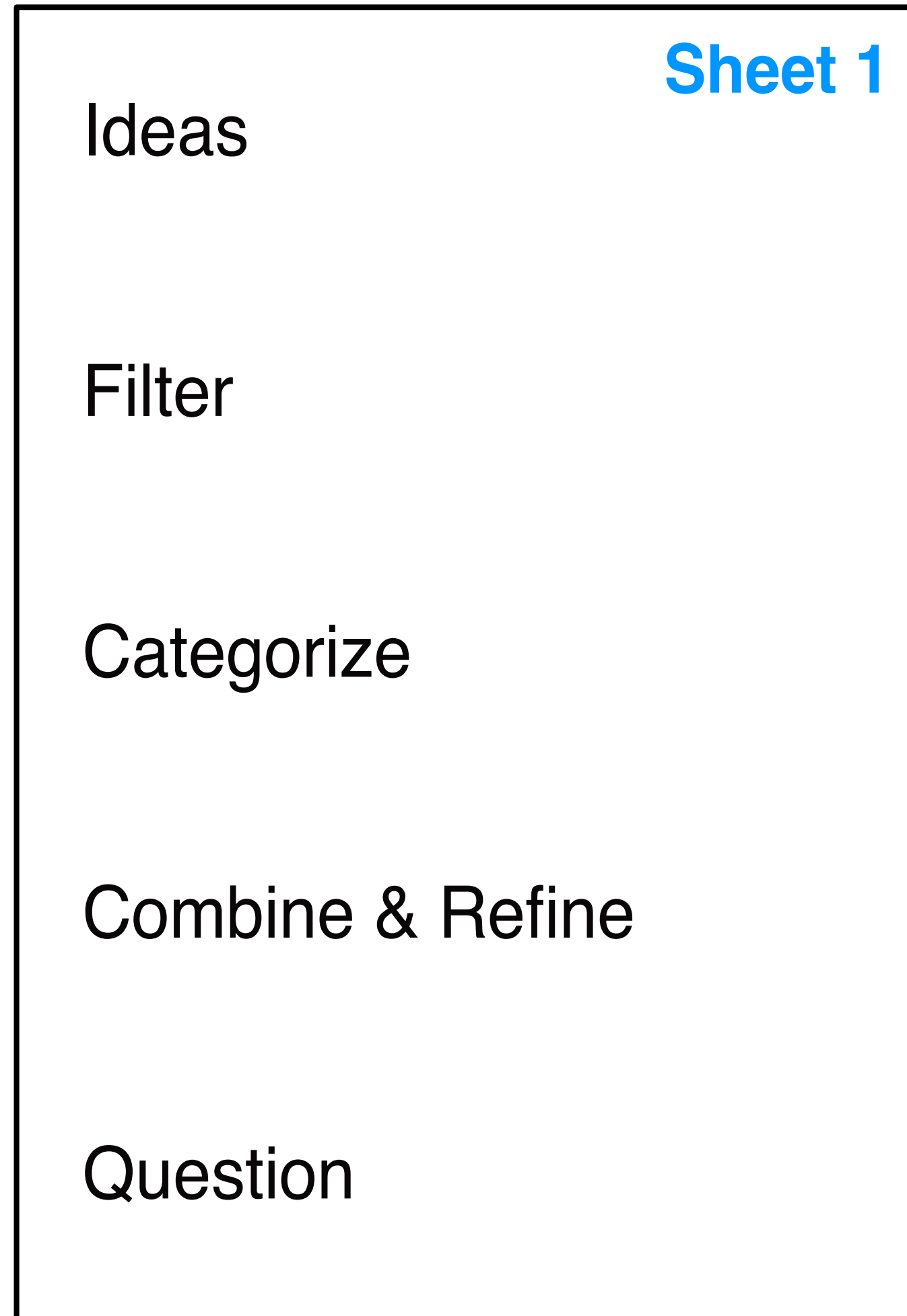
# Five Stages



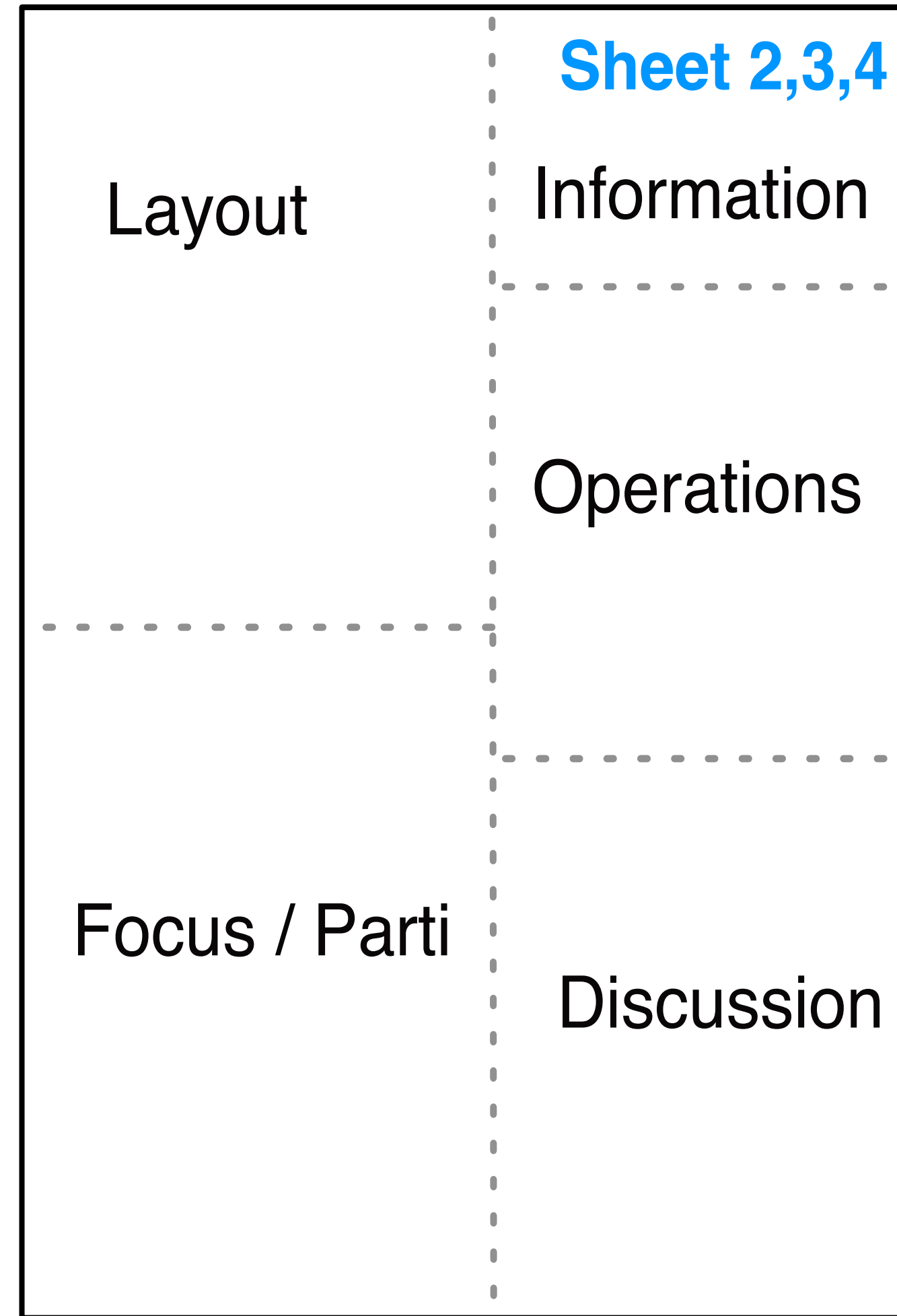
[J. Roberts et al., 2016]



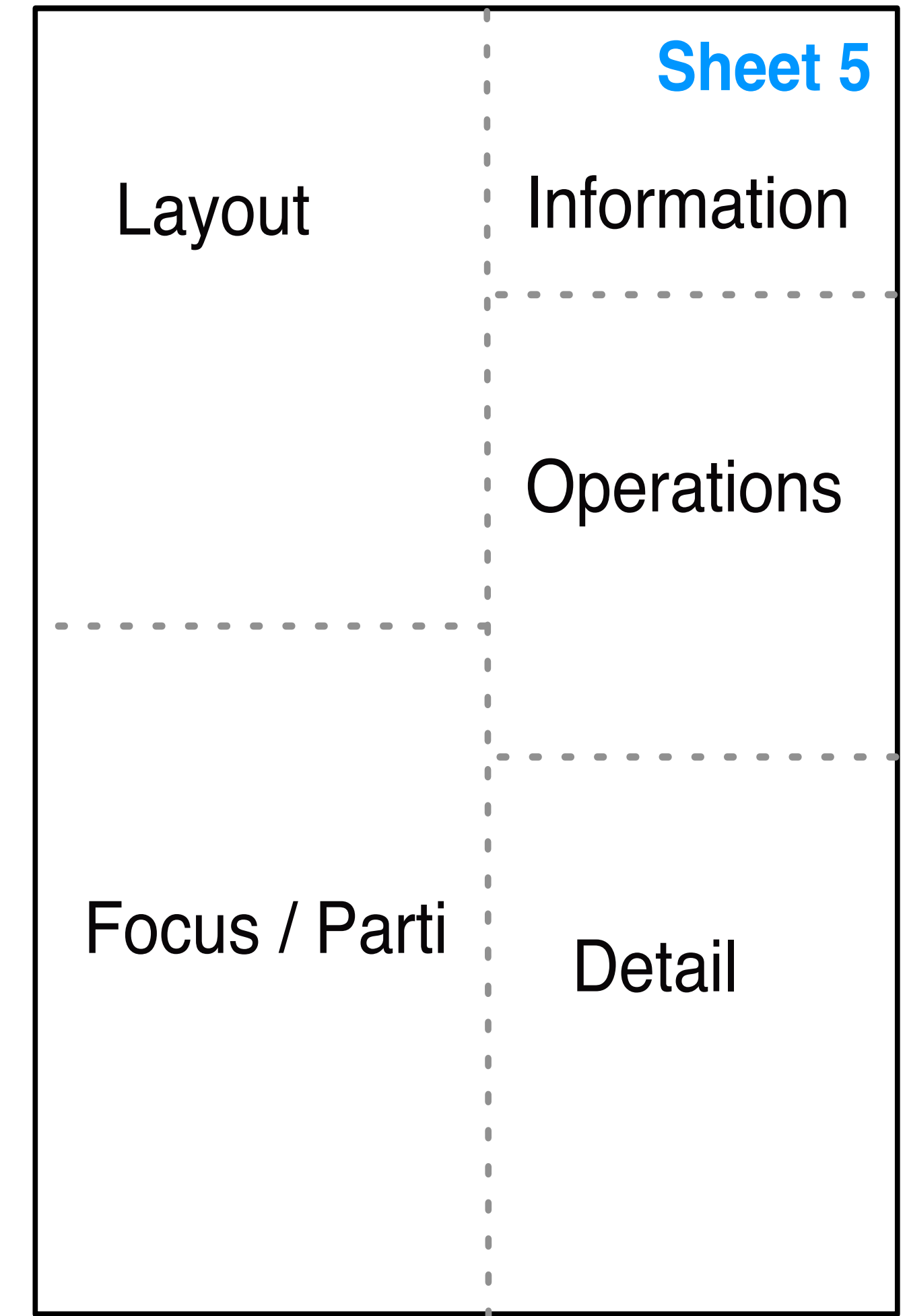
# The Five Sheets



Ideation



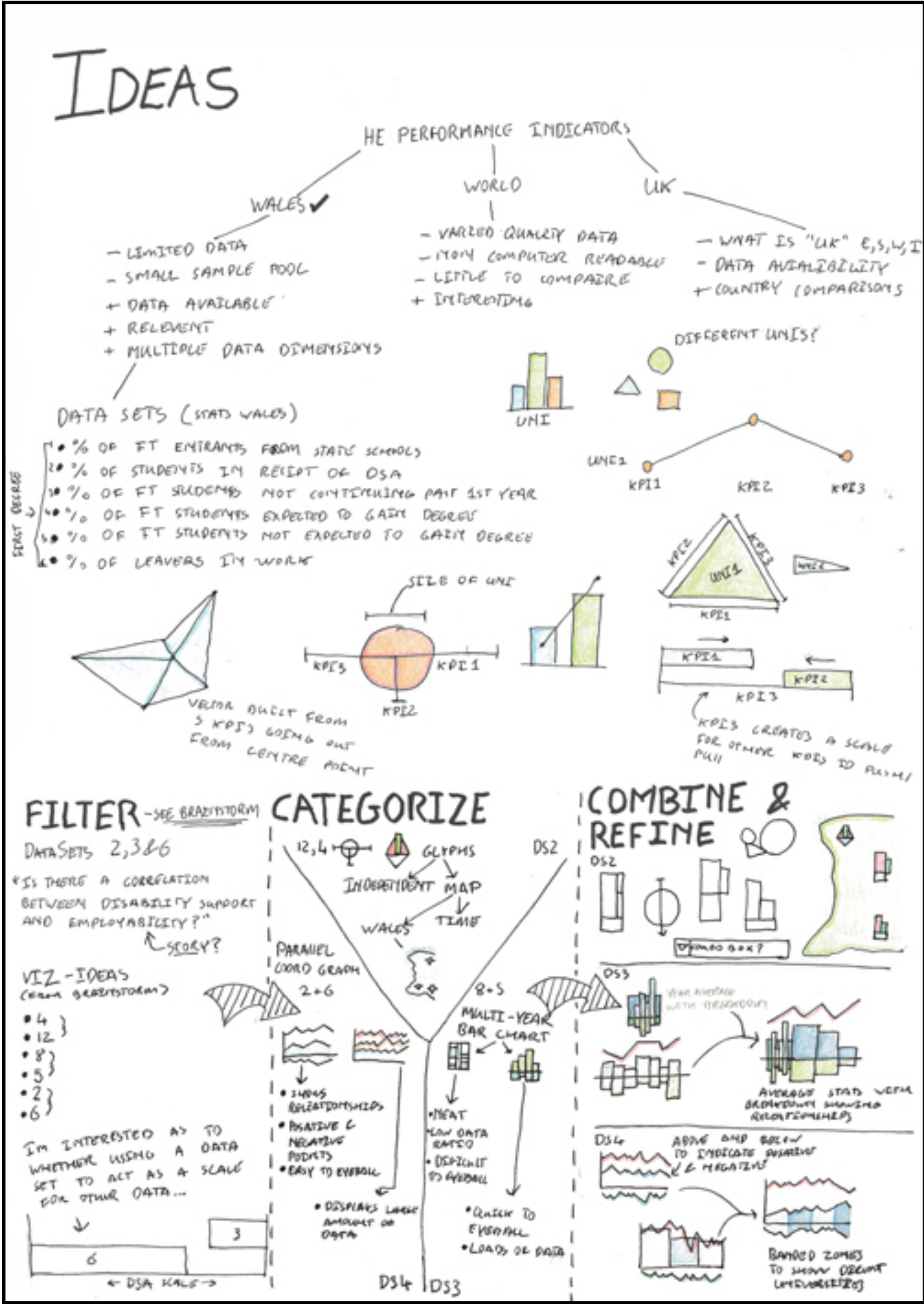
Alternative Designs



Realization

[J. Roberts et al., 2016]

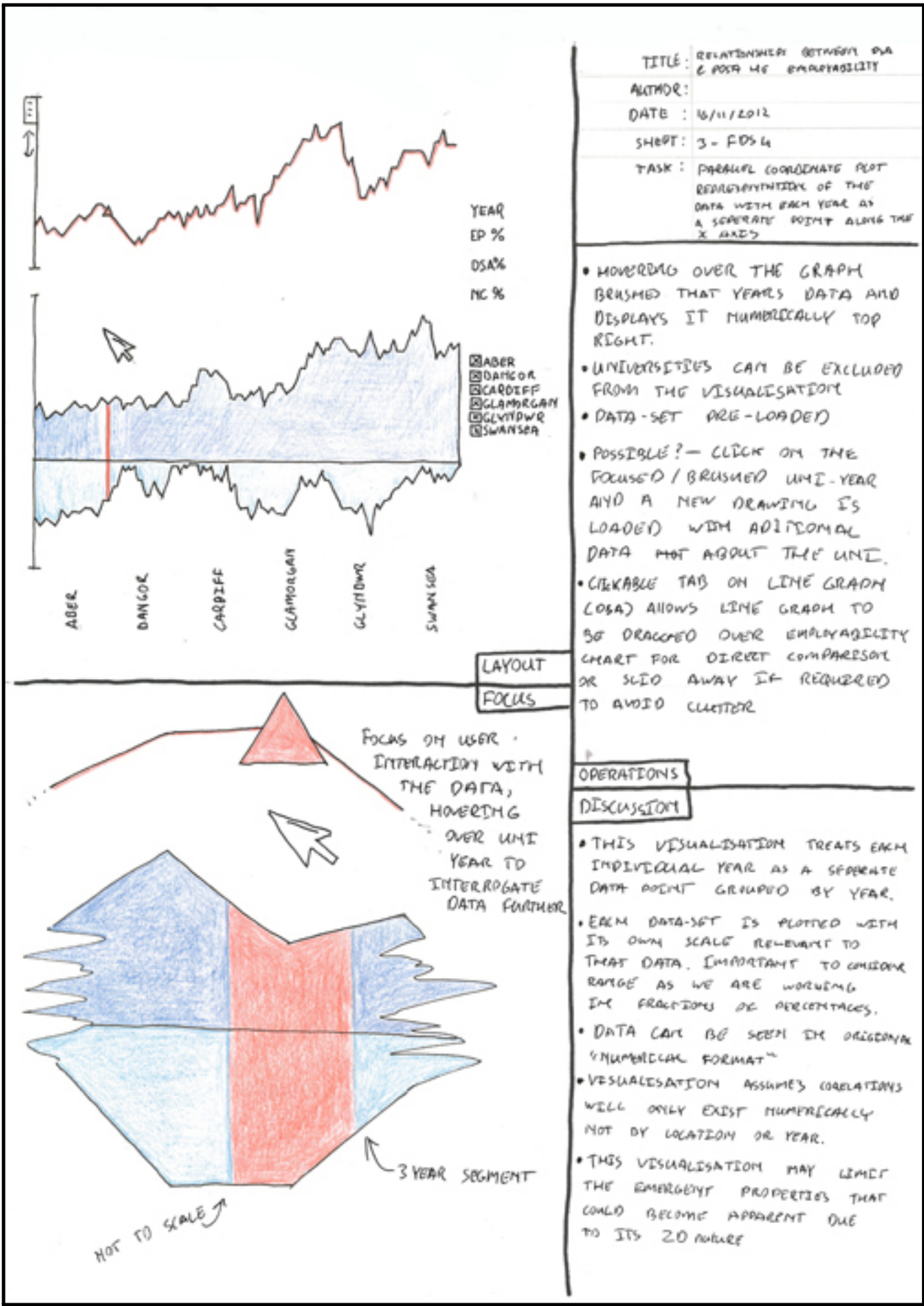
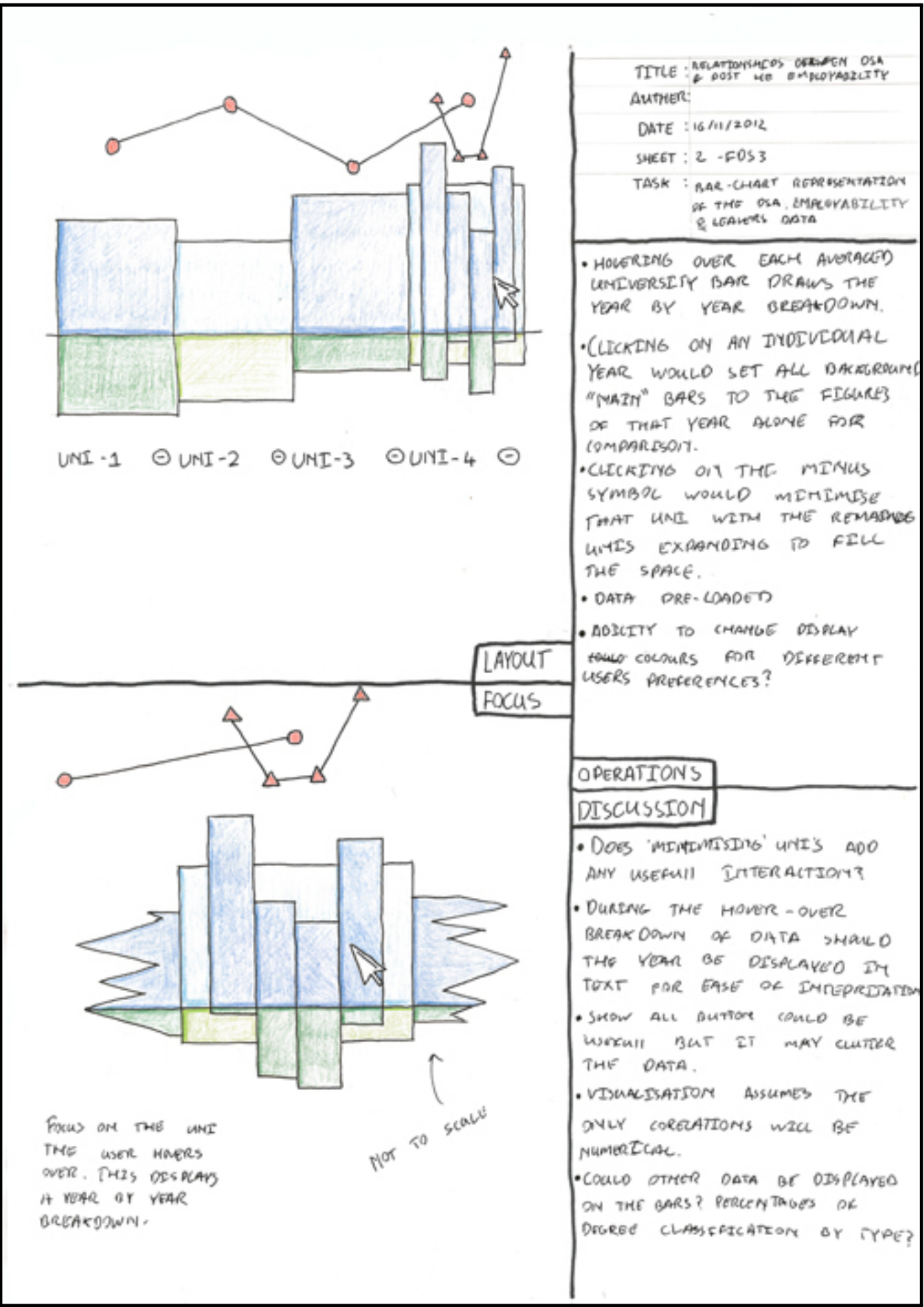
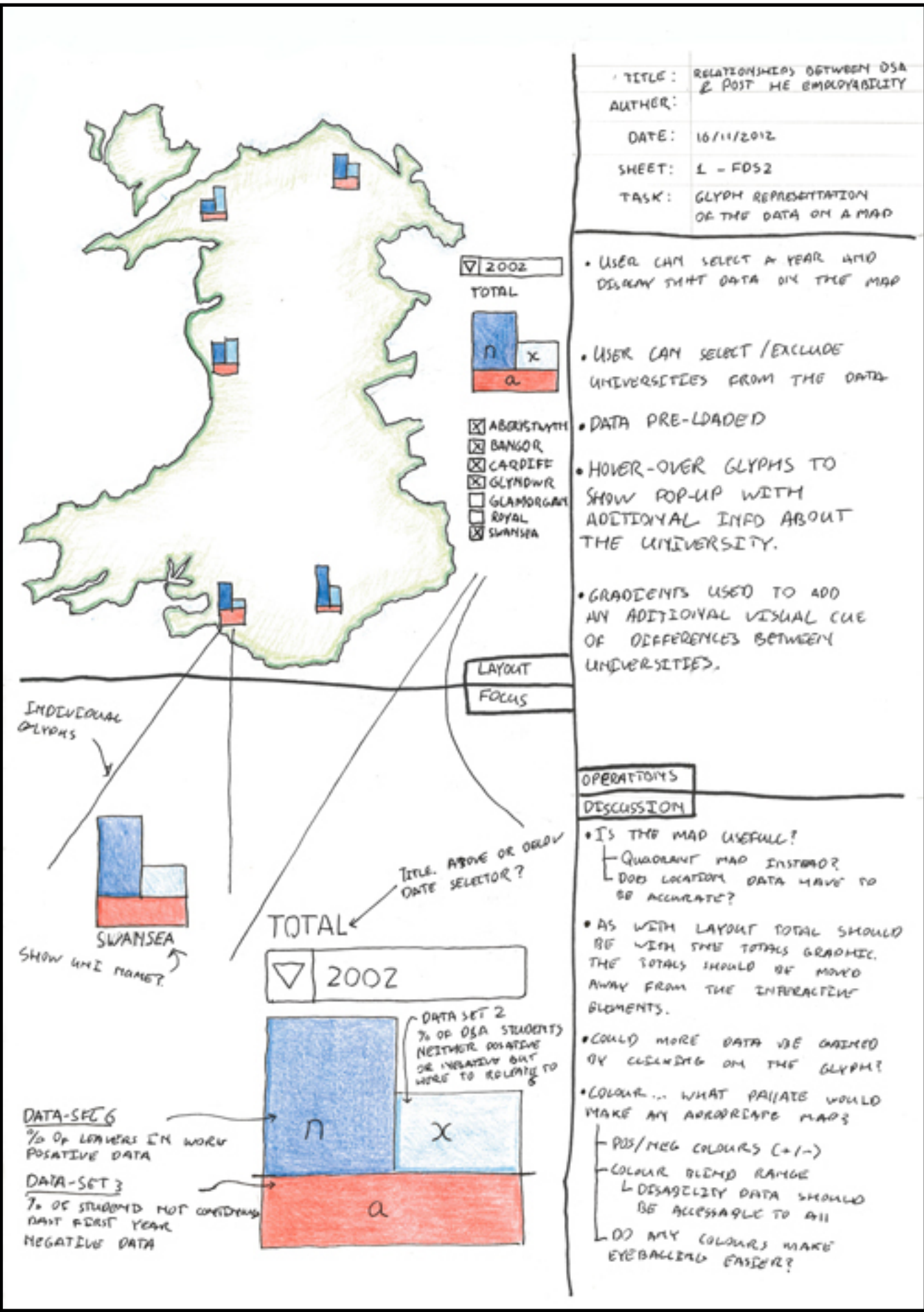
# Example: University Access for Disabled Students



[J. Roberts et al., 2016]



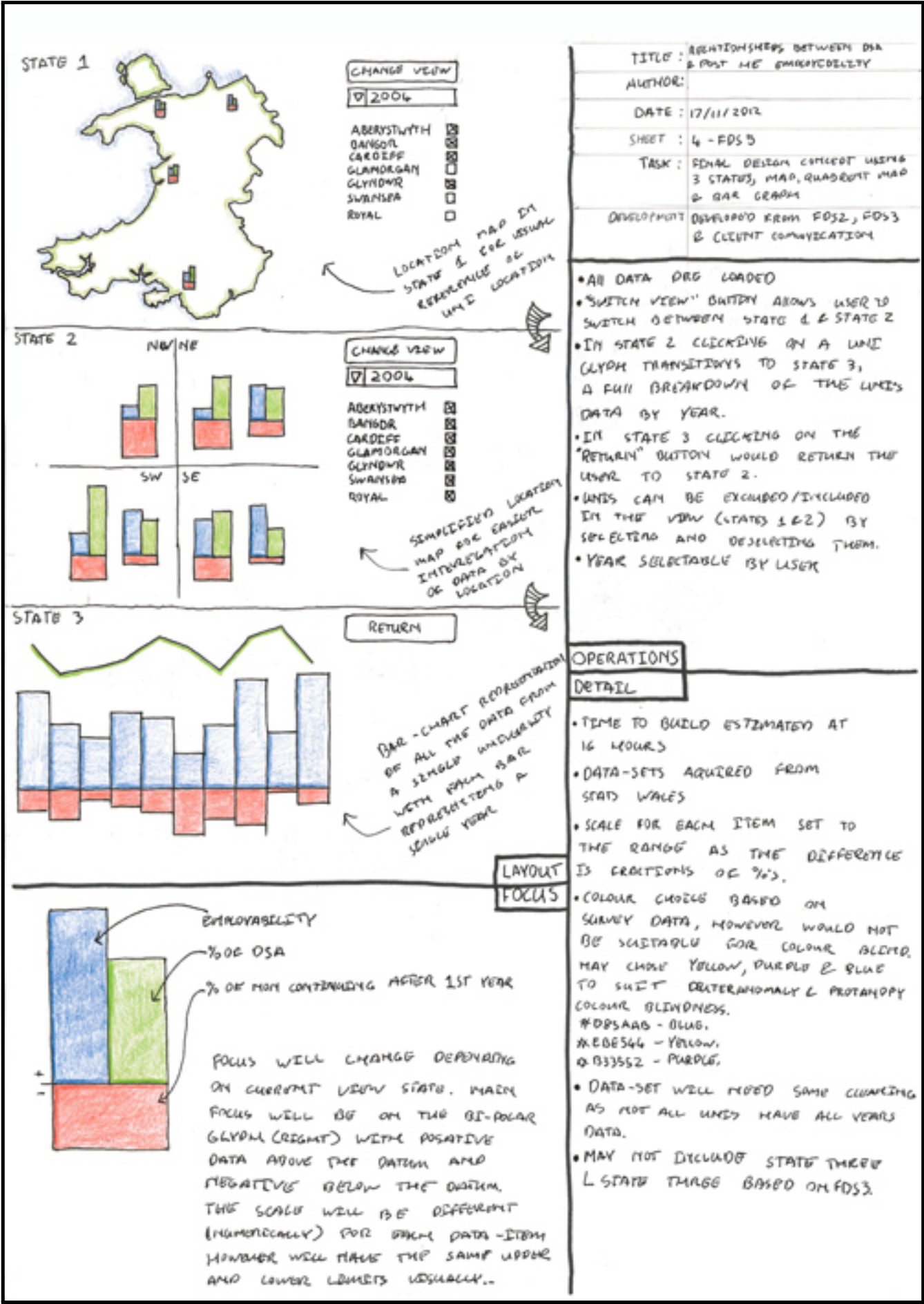
# Sheets 2-4



[J. Roberts et al., 2016]

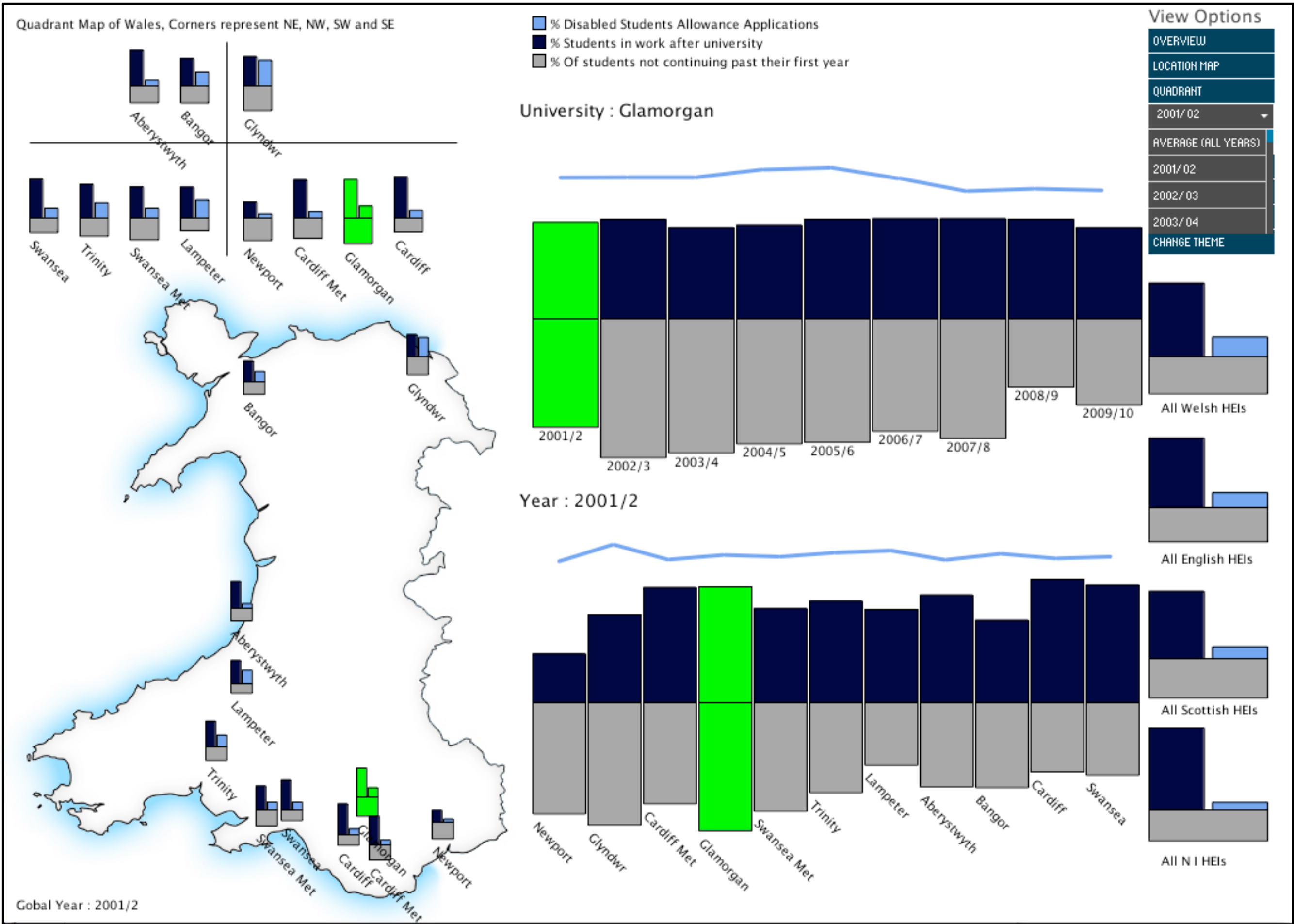


# Sheet 5



[J. Roberts et al., 2016]

# Prototype



[J. Roberts et al., 2016]