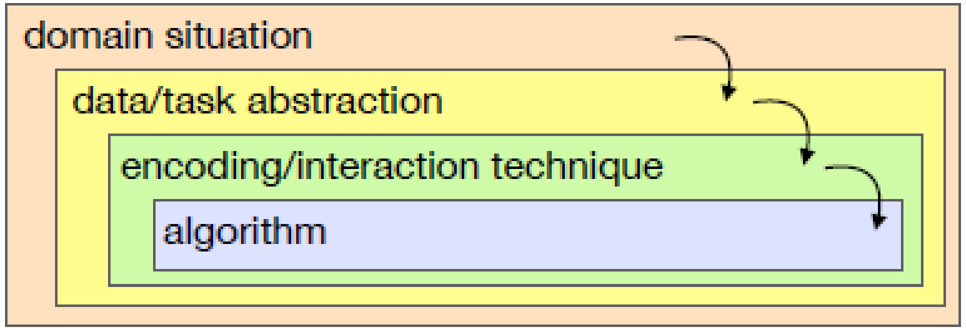
Lecture 4

9/15/2020

Analysis: Four Levels for Validation

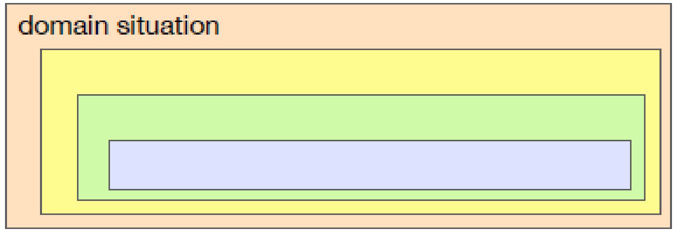
Four Nested Levels of Vis Design

- Design model: describes levels of design inherent to, and that should be considered in, the creation of a visualization.
- Four cascading levels are nested;
  - Output from an upstream level above is input to the downstream level below
- Block
  - Outcome of the design process at that level
  - Choosing a wrong block at an upstream level inevitably cascades to all downstream levels
    - Abstracting the wrong data will not solve the intended problem even if the idiom and algorithm levels are perfect
- Highly iterative refinement process
  - Principles of design as redesign



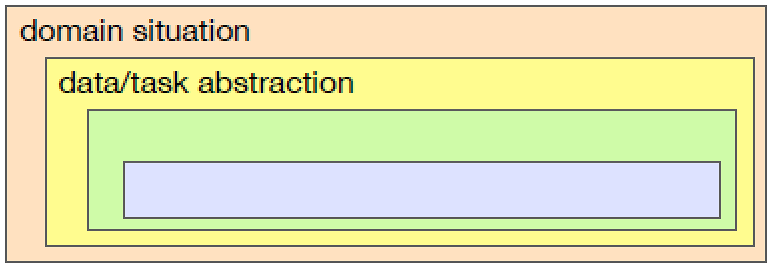
Domain Situation

- Describing a group of target users, their domain of interest, their questions, and their data
- Domain
  - particular field of interest of the target users of a vis tool
  - Has own vocabulary for describing its data and problems
  - Users: as small as 5 people of a department to as big as the general public (ex scientific community)…
- At this level, **situation blocks are identified**
  - **Outcome** of the design process is an understanding that the designer reaches about the needs of the user
  - **Methods** typically used to identify domain situation blocks: interviews, observations, careful research about target users within a specific domain
  - Common pitfall: designers cut corners by making assumptions and not engaging with target users
- User-centered design or human-centered design
  - Working closely with a specific target audience to iteratively refine a design



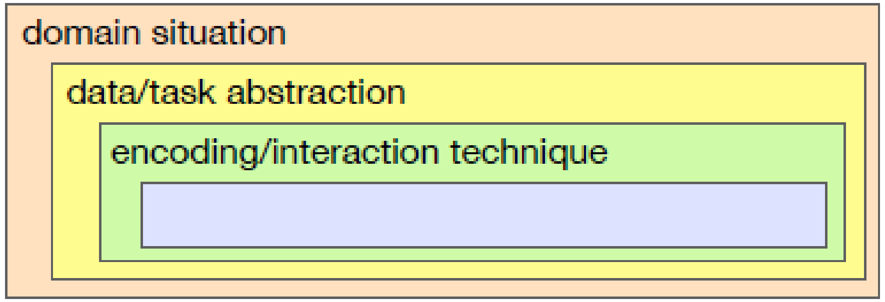
Data/Task Abstraction

- Abstracting the specific domain questions and data from the domain-specific form into a generic, computational form
- Questions from very different domain situations can map to the same abstract vis tasks
  - Ex: browsing, comparing, summarizing
- **Abstract data blocks are designed**
  - Selecting a data block is a creative design step
  - Original data may work in some cases
  - More often, designer chooses to transform the original data from its upstream form to something quite different
  - Many vis idioms are specific to a particular data type
  - Goal: determine which data type would support a visual representation of it that addresses the user's problem



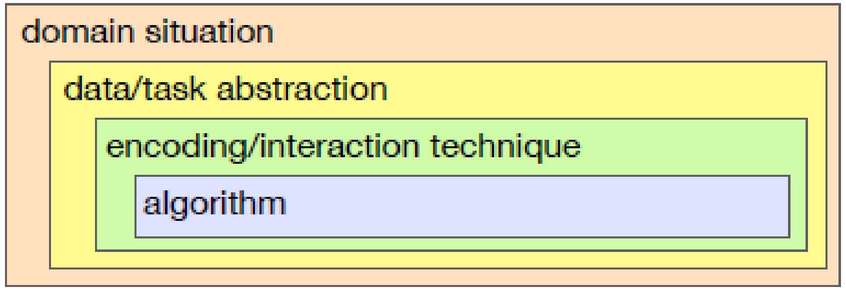
Encoding/Interaction Technique

- Decide on the specific way to create and manipulate the visual representation of the abstraction
- Idiom – each distinct possible approach
- Two concerns:
  - How to **create** a single picture of the data
    - **Visual encoding** idiom controls exactly what users see
  - How to **manipulate** that representation **dynamically**
    - **Interaction** idiom controls how users change what they see
- Good and bad matches
  - Human abilities: visual perception and memory



Algorithm

- Crafting a detailed procedure that allows a computer to automatically and efficiently carry out the desired visualization goal
- Efficiently handle the visual encoding and the interaction idioms chosen in the previous level
- **Algorithm blocks are designed**
- Different algorithms to achieve an idiom
  - Speed of computation, computer memory required, is resulting image an exact match or an approximation of specified visual encoding idiom
- Nested model emphasized separating algorithm design from idiom design
  - Algorithm design concerns are primarily about computational issues
  - Idiom design concerns are more about human perception issues



Angles of Attack

Problem-driven work	Technique-driven work

• Start at top domain situation level and work down through abstraction, idiom, and algorithm decisions	• Bottom two levels: idiom or algorithm design
	• Goal:
	  • Invent new idioms that better support existing abstractions
	  • New algorithms that better support existing idioms

Problem-Driven Work

- Address problems of real-world users and attempt to design a solution that helps them work more effectively
  - Aka design study
- Usually existing visual encoding and interaction idioms solve the problem
- Challenge usually lies at the abstraction level
- Sometimes, the problem motivates the design of new idioms (if existing ones don’t adequately solve the abstracted design problem)
- Design process for problem-driven work
  - Almost never strictly linear
  - Involves iterative refinement at all of the levels

Technique-Drive Work

- Starting point
  - New visual encoding or interaction idiom
  - New algorithm

Threats to Validity

- Fundamental reasons why you might have made the wrong choices
- Different set of threats for each of the four levels
- “they” = target users
- “you” = vis designer

Validation Approaches

- Immediate
- Downstream
- Nested levels
  - Most kinds of validation for outer levels are not immediate because they require results from the downstream levels nested within them
- Examples:
  - poor algorithm design may cast doubt when testing an interaction technique
  - Poor visual encoding choice may cast doubt when testing a legitimate abstraction choice
- Get feedback from target users about abstraction and encoding designs before diving into designing or implementing any algorithms

Validation Approaches: Domain

- Primary threat
  - problem is mischaracterized; target users do not in fact have these problems
- Validation
  - Immediate
    - Interview and observe the target audience to verify the characterization (don't rely on assumptions)
    - Common approach: field study
      - Investigate people in real world settings (not laboratory)
      - Gathering qualitative data from semi-structured interviews
  - Downstream
    - Report rate at which the tool has been adopted by the target audience, but this does not always tell the whole story!
    - A tool that is actually used by its intended users has reached a different level of success than one that has only been used by its designers

Validation Approaches: Abstraction

- Threat
  - identified task abstraction blocks and designed data abstraction blocks do not solve the characterized problems of the target audience
- Key aspect of validation:
  - System must be tested by target users doing their own work (not an abstract task specified by the designers of the vis system)
  - Downstream approach:
    - Have a member of the target user community try the tool
    - Hopes to collect anecdotal evidence (insights found, hypotheses confirmed) that the tool is in fact useful
    - All three of the other levels have to be fully addressed first
    - Usually qualitative, but quantitative methods have been developed
    - Important to distinguish between discovery made by target user vs designer; former is compelling argument for utility of the vis tool
  - Rigorous validation approach: field study
    - Observe and document how target audience uses the deployed system as part of their real-world work-flow
    - ** Field study difference for abstraction vs domain:
      - Domain field study – documents existing work practices
      - Abstraction field study – observing how behavior changes after intervening with the deployment of a vis tool

Validation Approaches: Idiom

- Threat
  - Chosen idioms are not effective at communicating the desired abstraction
- Immediate validation approach:
  - Justify the design of the idiom with respect to known perceptual and cognitive principles
  - Evaluation methods to systematically ensure that no known guidelines are being violated by the design
    - Heuristic evaluation
    - Expert review
- Downstream approach
  - Lab study
    - Quantitative, qualitative, objective measurements by carrying out tasks, subjective measurements such as preferences, mouse clicks, eye movement
  - Presentation of and qualitative discussion of results in the form of still images or video
    - Why is this considered downstream?
  - Quality metrics - quantitative measurements of results images created by the implemented system
    - Ex: measurable layout metrics such as # of edge crossings and edge bends used to assess drawings of node-link networks

Validation Approaches: Algorithm

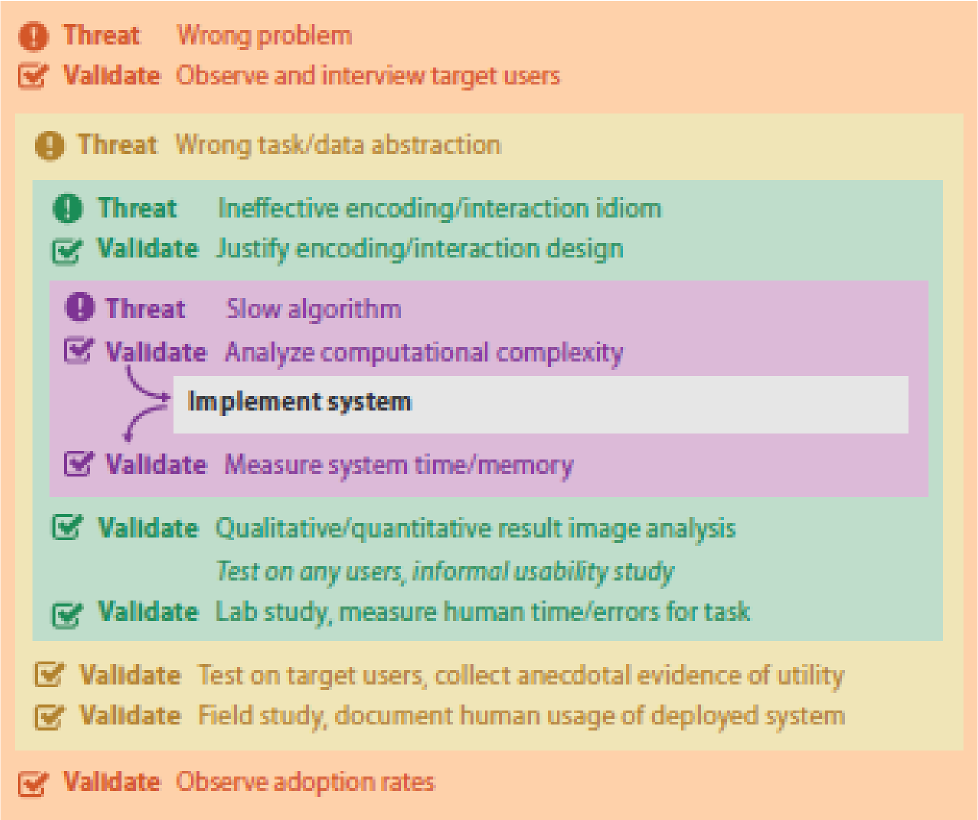
- Threat
  - Algorithm is suboptimal in terms of time or memory performance (theoretically or compared to a previously proposed algorithm)
- Immediate validation approach
  - Analyze the computational complexity of the algorithm (# of items in dataset, # of pixels in the display)
- Downstream validation approach
  - Measure the wall-clock time and memory performance of the implemented algorithm
    - Scalability (how dataset size affects algorithm speed, datasets to use – benchmark and broad set)

Validation Approaches: Algorithm

- Threat
  - Incorrectness at algorithm level where implementation does not meet the specification from the idiom level above (poor algorithm design or bugs)
- Implicit validation approach
  - Presenting still images/videos created by the implemented algorithm, where the reader of a paper can directly see that the algorithm correctness objectives have been met
- Explicit validation approach
  - Qualitative discussion of why these images show that the algorithm is in fact correct

Nested Model: Threats and Validations

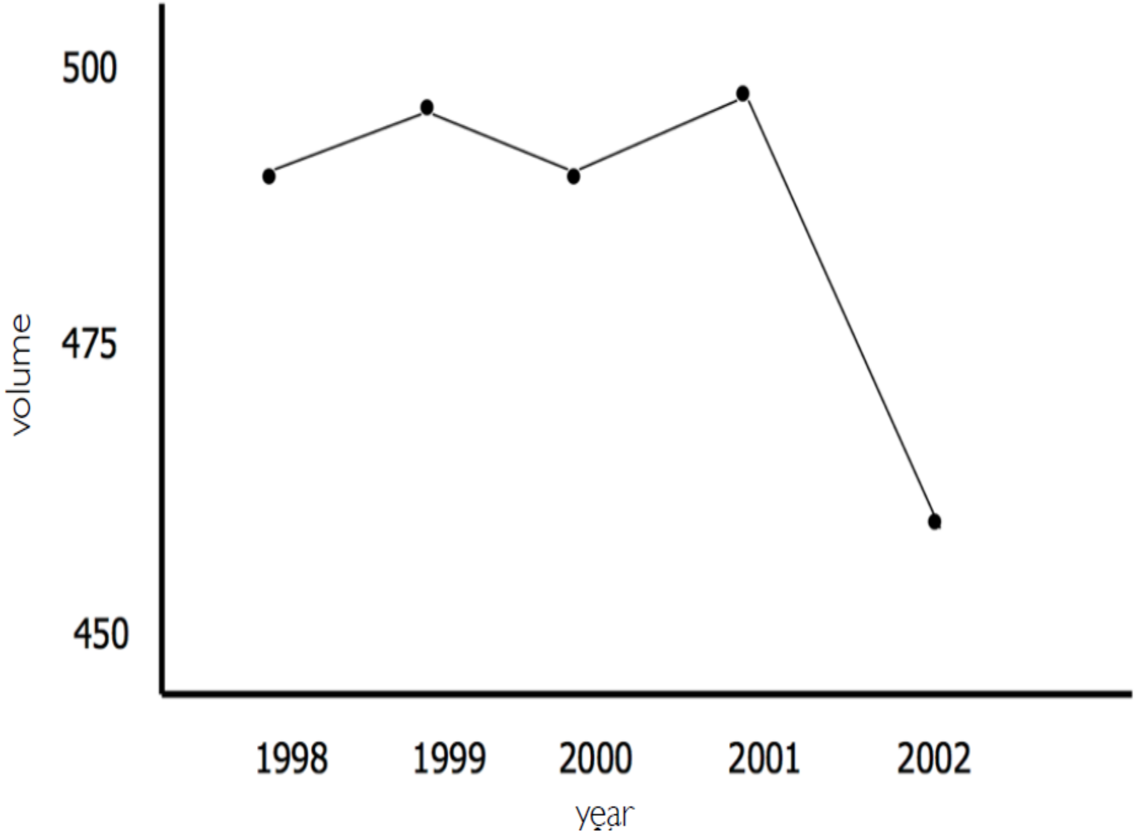
The nested model explicitly separates the vis design problem into levels in order to guide validation according to the unique threats at each level.



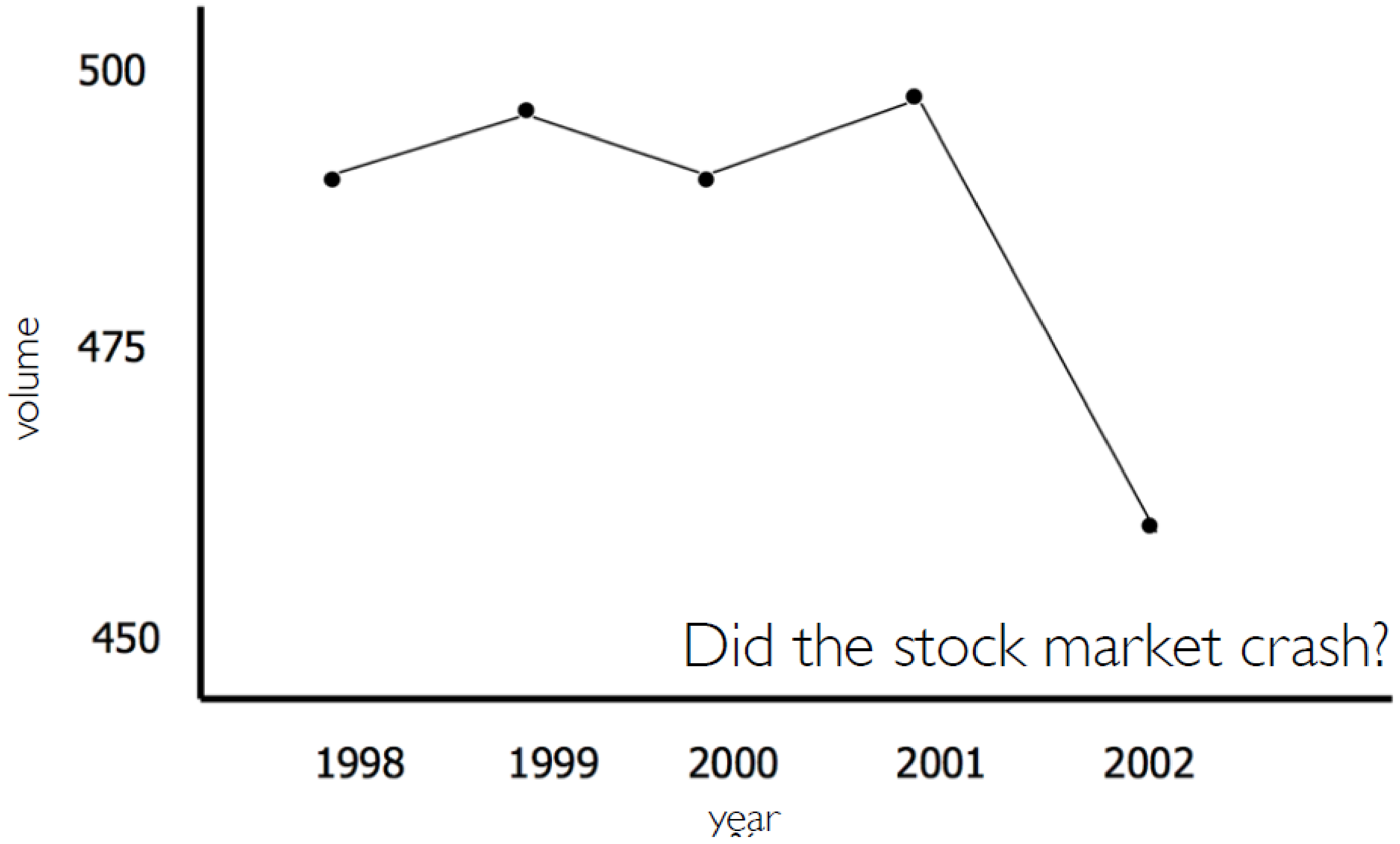
Tufte: Graphical Integrity

- Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity.

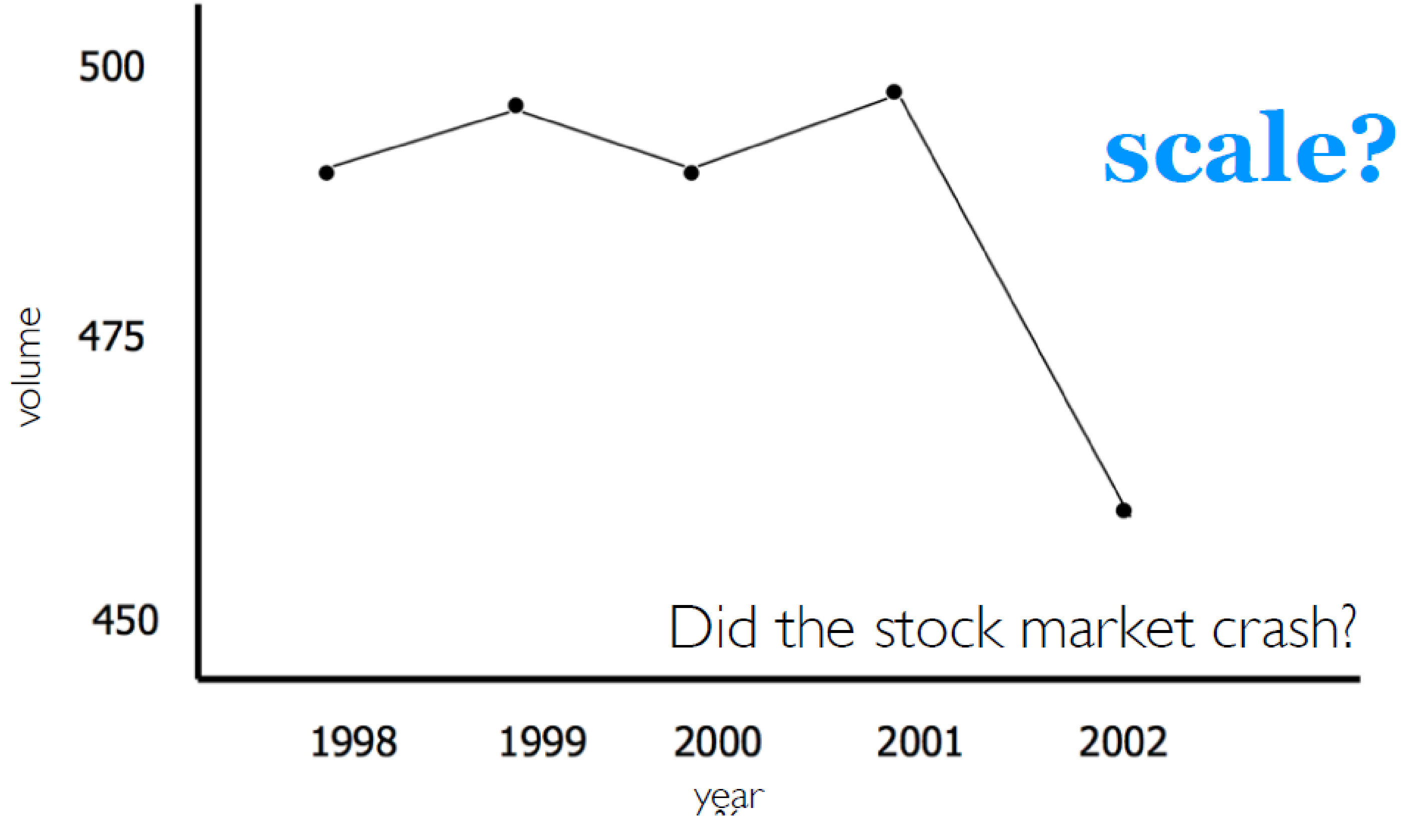
SCALE DISTORTION



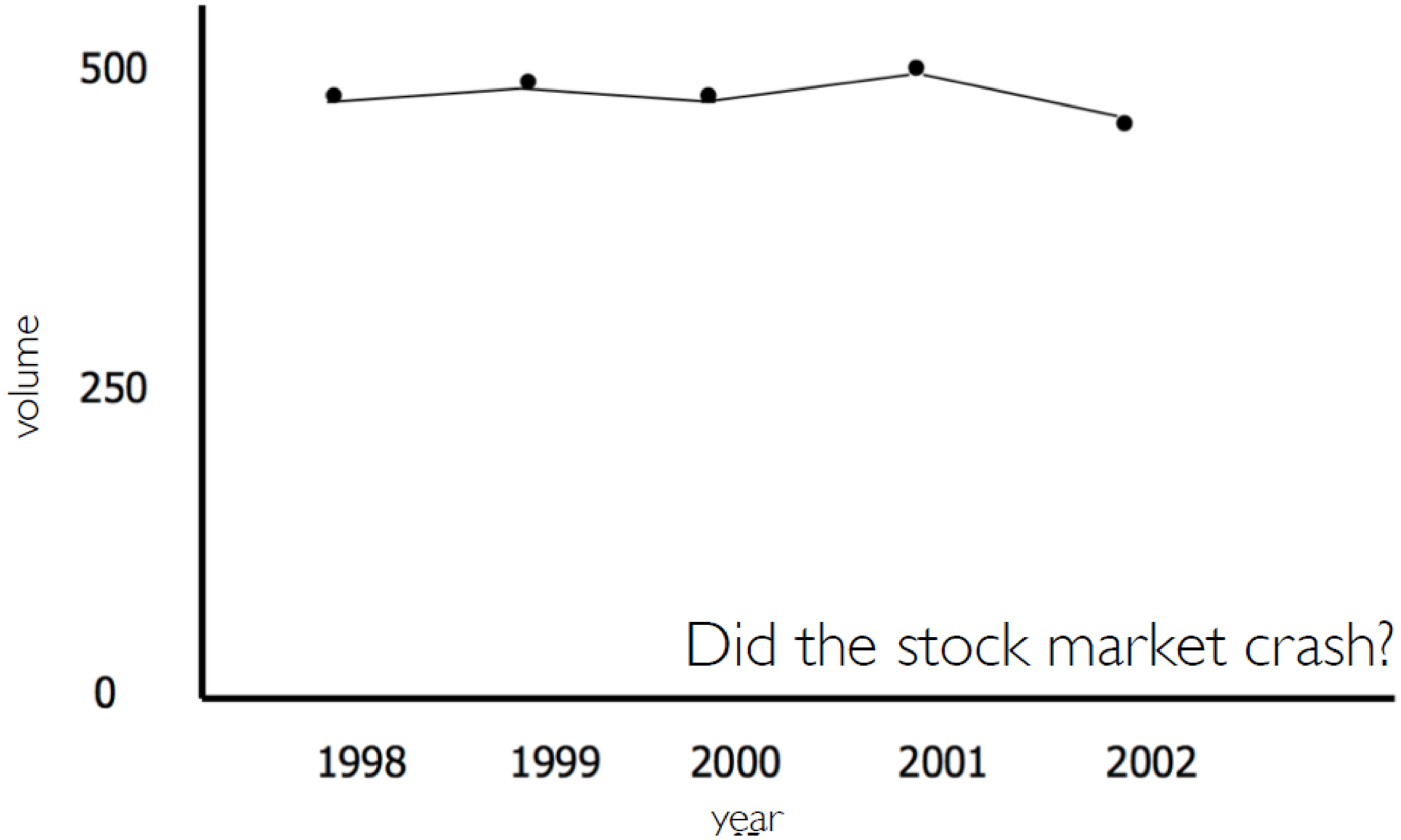
SCALE DISTORTION



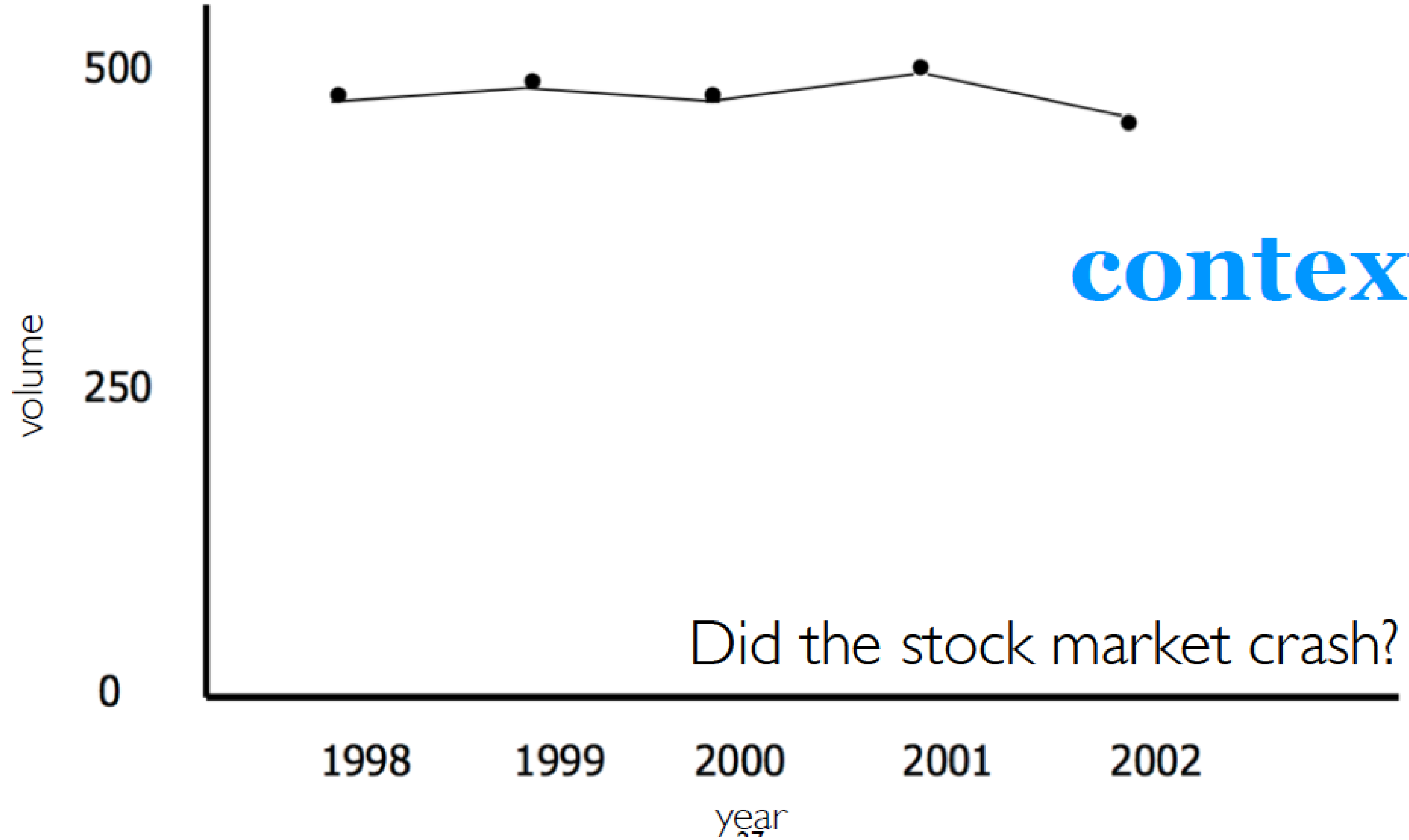
SCALE DISTORTION



SCALE DISTORTION

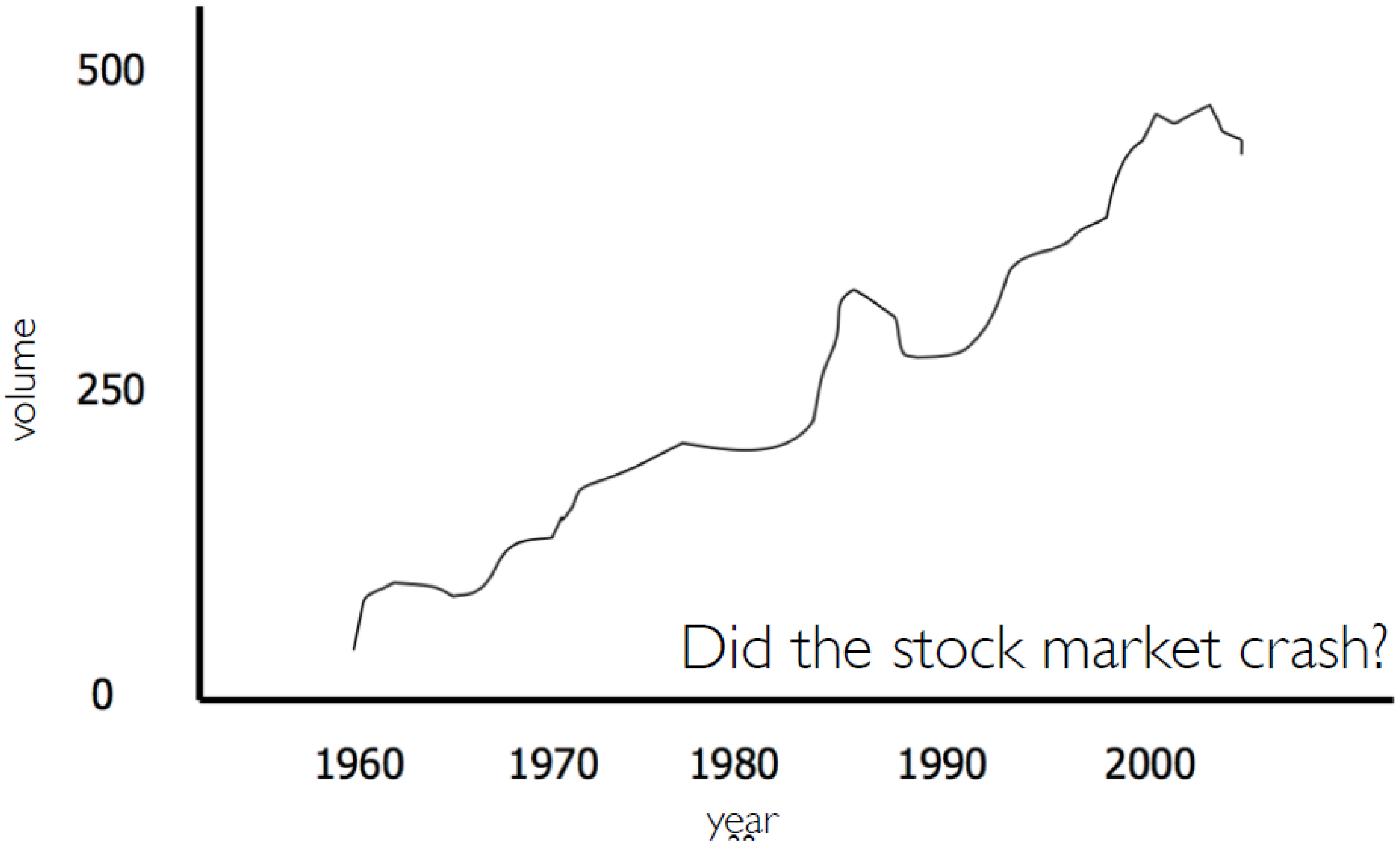


SCALE DISTORTION

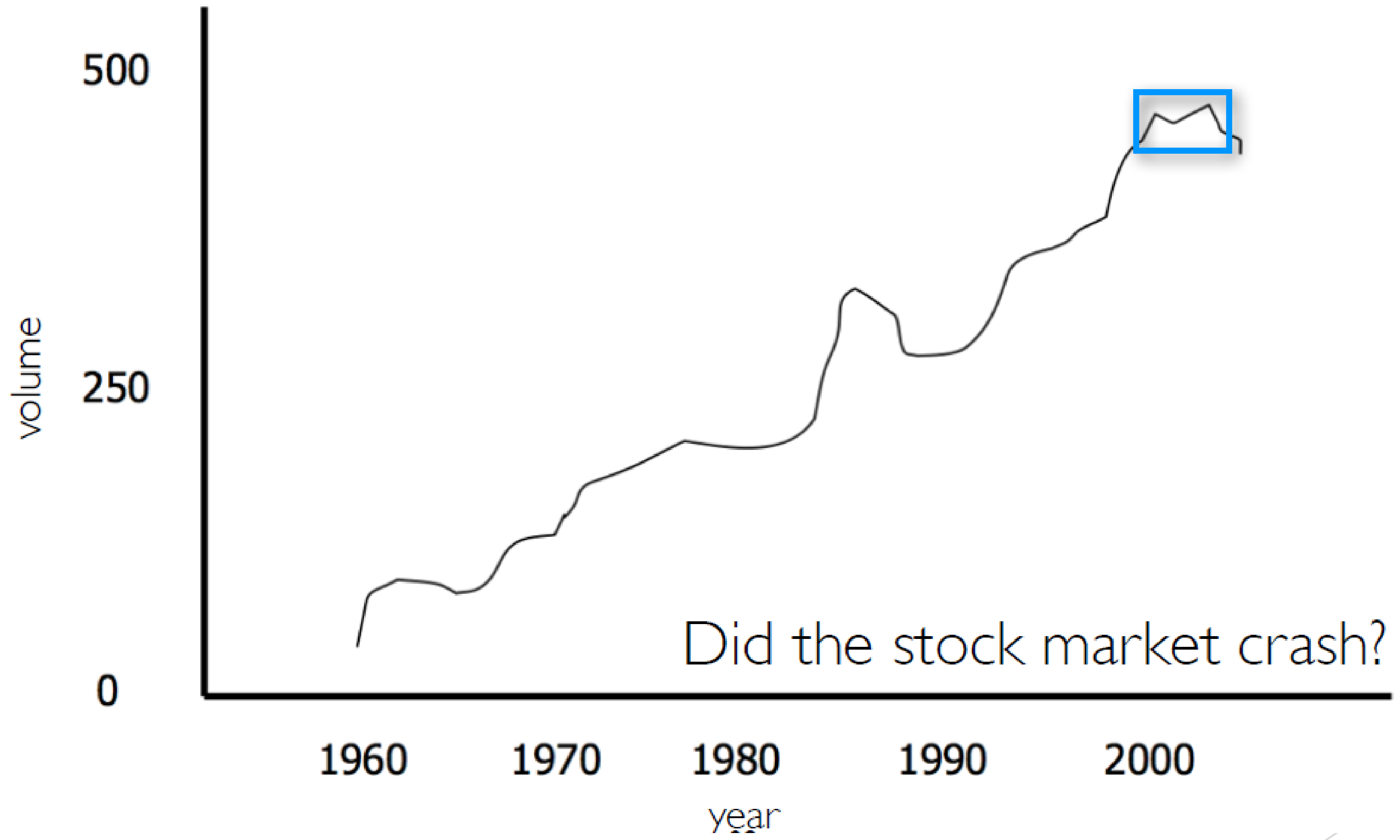


context?

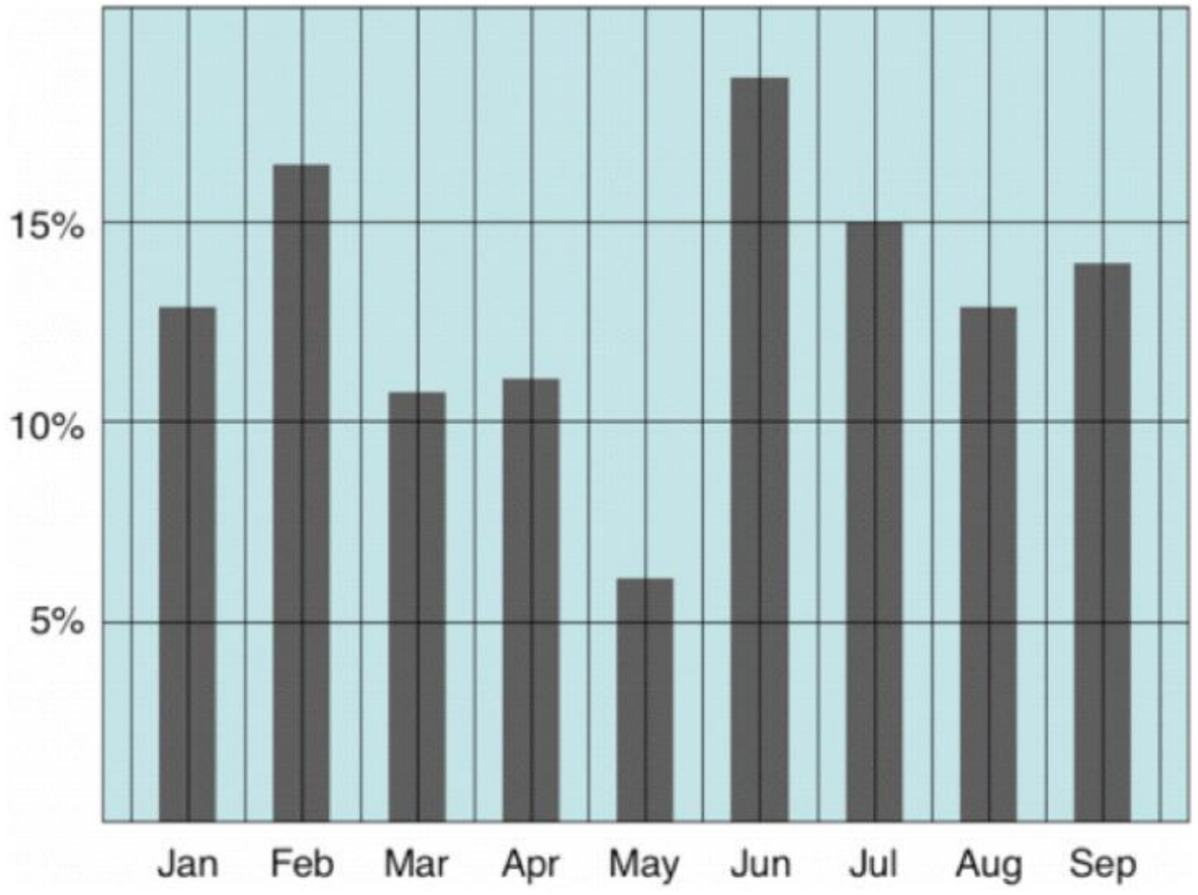
SCALE DISTORTION



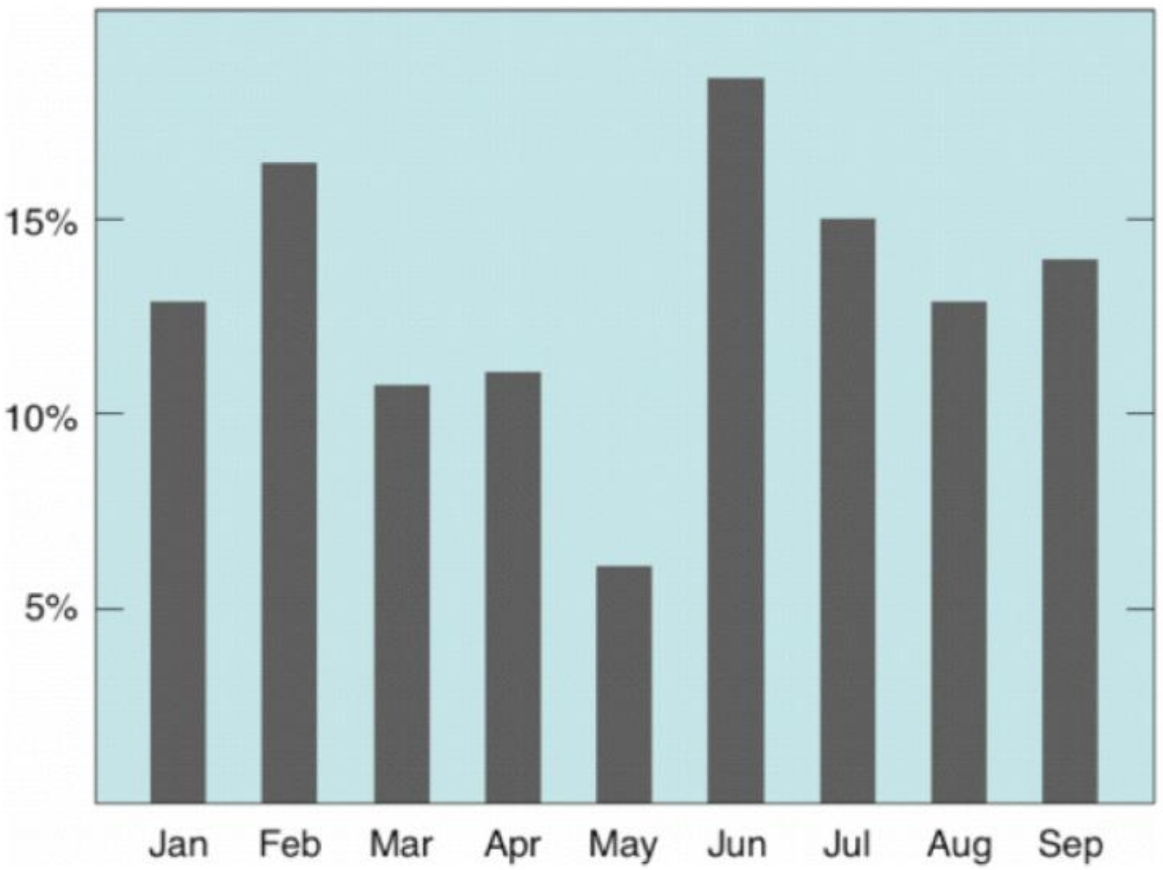
SCALE DISTORTION



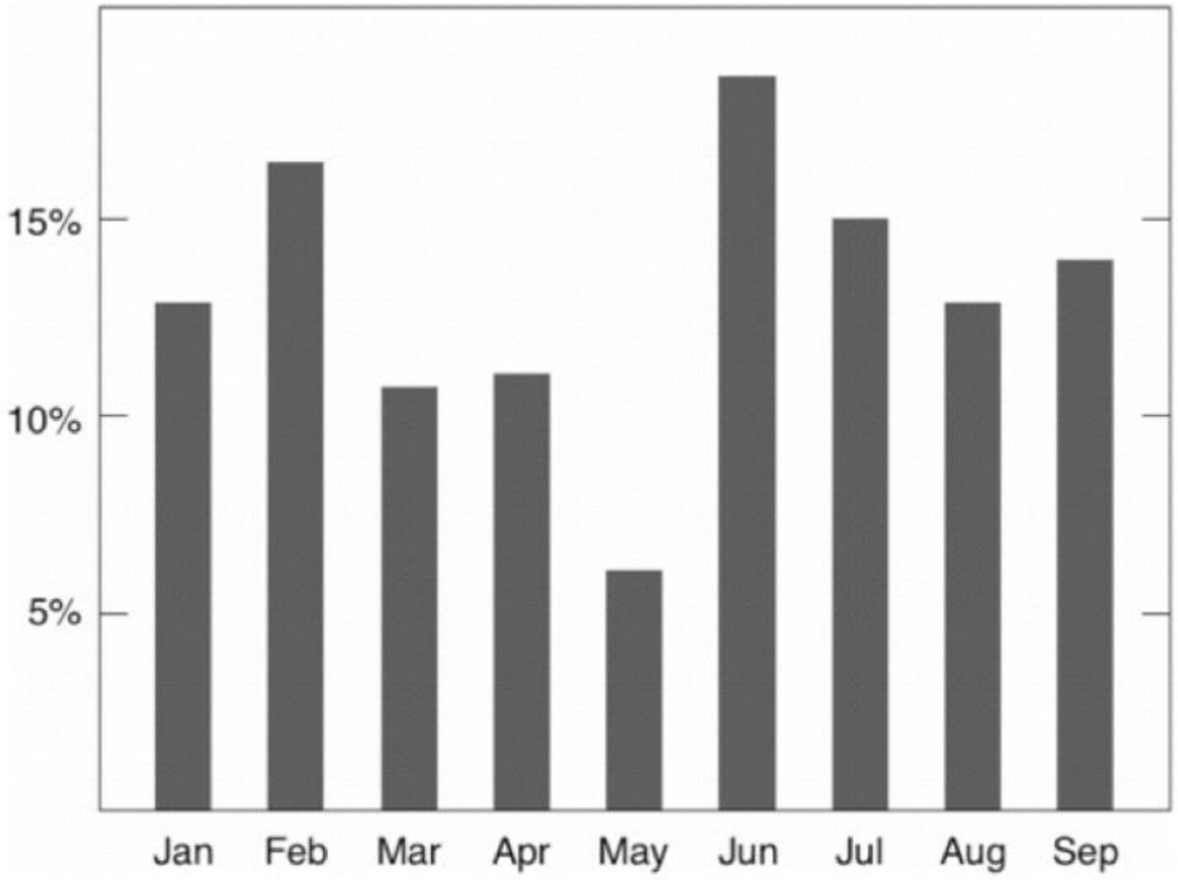
Avoid Chart Junk



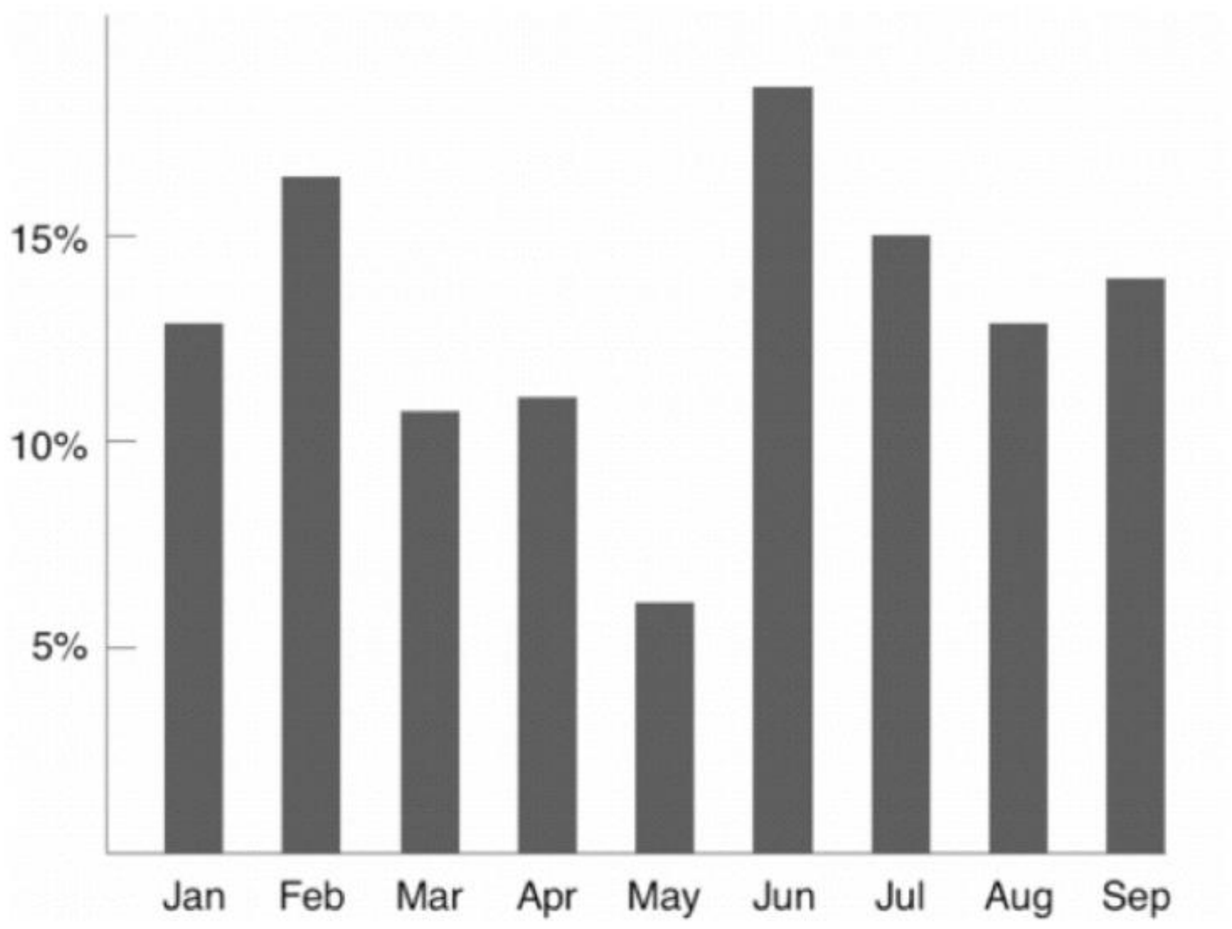
Avoid Chart Junk



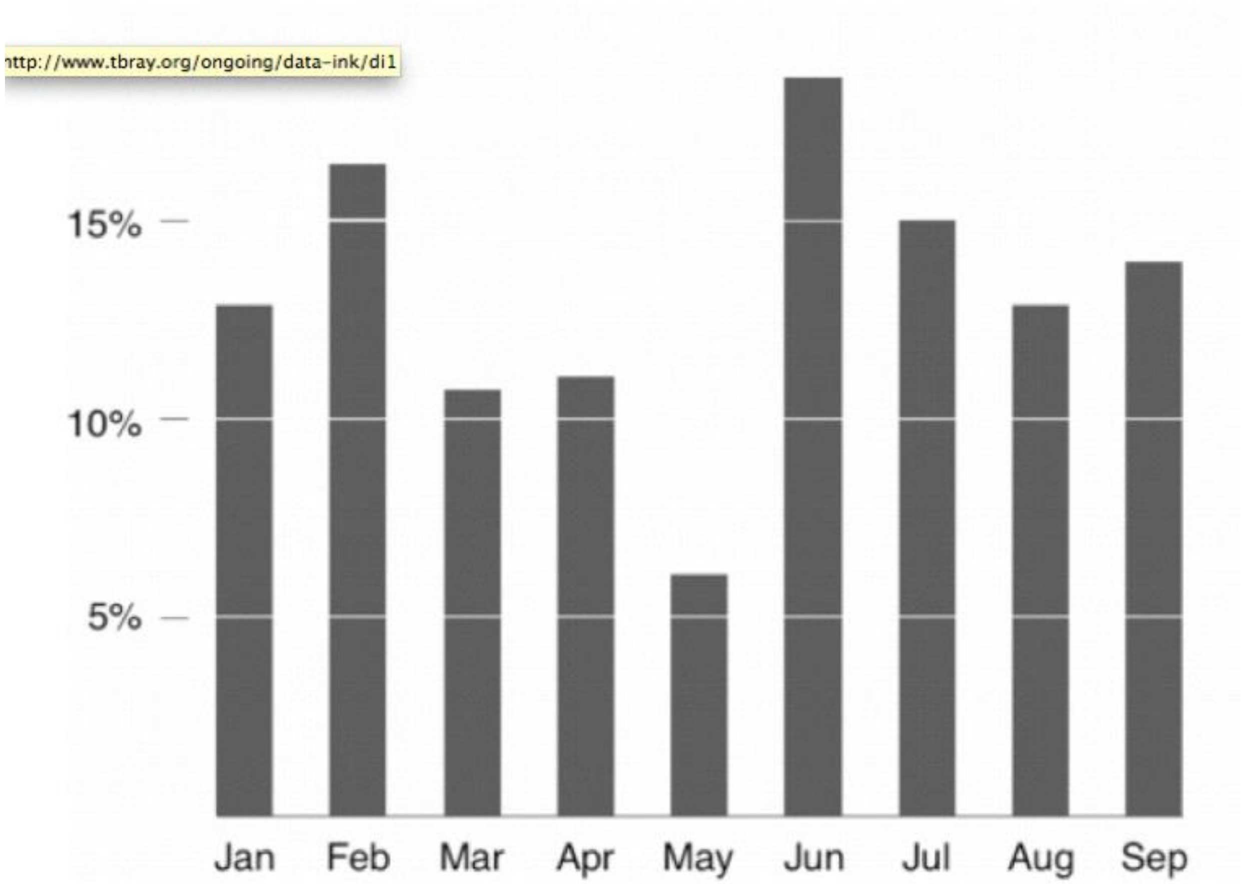
Avoid Chart Junk



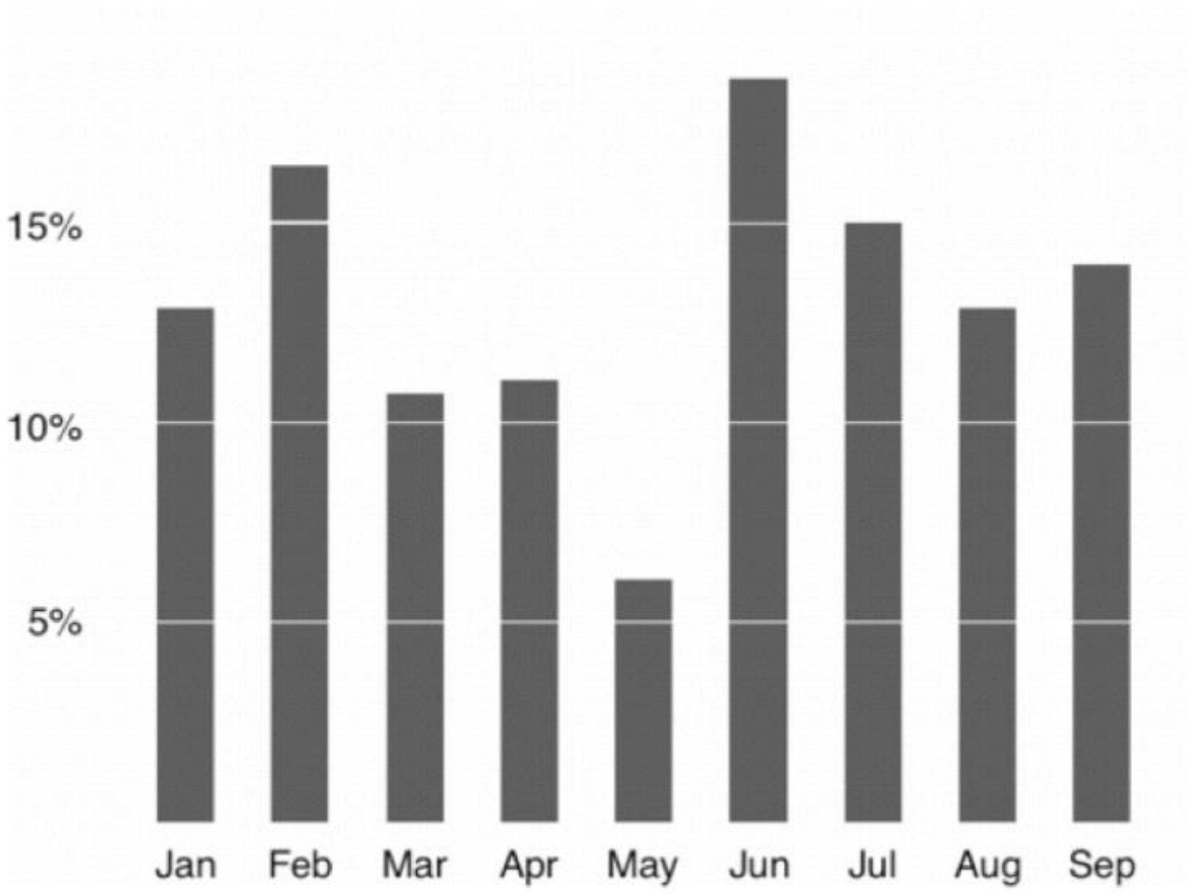
Avoid Chart Junk



Avoid Chart Junk



Avoid Chart Junk



Sources/Credits

- Tamara Munzner, Visualization Analysis & Design, A K Peters Visualization Series, CRC Press, 2014.
- Utah, Miriah Meyer, Visualization (2014).
- UMass Dartmouth, David Koop, Data Visualization (2015).