Grand Challenges: Definitions
- grand challenges in other fields
  - physics: build atom bomb
  - astrophysics: man on the moon
  - biology: cure cancer
- "outward" grand challenges
  - high impact, broadly understandable, inspiring
  - clear milestone to judge success
  - concrete driving problems to galvanize field

Validation Methods - How To Choose?
- unsatisfying flat list of validation methods when writing recent paper

Infovis
- Problem
- Validation Methods - How To Choose?

Inward GC: Towards Science
- "inward" grand challenge for infovis: building it into a science
  - how can we accelerate the transition from a collection of papers to a body of work that constitutes a science?
  - need synthesis at scales larger than a single paper
  - need to add common framework unifying all vi work
  - need for doing good science within single paper
  - guide for creating papers that can teach useful others
  - some current thoughts as concrete example...

Separating Design Into Levels
- multiple levels
  - domain problem characterization
  - data/operation abstraction
  - design
  - encoding/interaction technique
  - design
- three separate design problems
  - not just the encoding level
  - each level has unique threats to validity
  - execute language from security via software engineering
  - dependencies between levels
  - outputs from level above are inputs to level below
  - downstream levels required for validating some upstream threats

Abstraction Design
- for chosen problem, you abstract into operations on specific data types
- often need to derive/transform data type from raw data
- ex: choose coastal-to-coast train route
  - abstraction: path following on node-link graph with initial node positions (lat, lon) and two sets of weights on edges (cost, beauty)
  - can your abstraction solve the problem?
  - threat: bad choice of abstraction not faithful to solving problem
  - downstream validation: observe whether useful with field study

Algorithm Design
- for chosen encoding/interaction, you design computational algorithm
- is your algorithm better than previous approaches?
  - threat: algorithm slower than previous ones
  - immediate validation: analyze computational complexity
  - downstream validation: after implementation, measure wallclock time

Matching Validation To Threats
- threat: wrong problem
  - validate: observe target users
- threat: bad data/operation abstraction
  - validate: justify design
- threat: encoding/interaction technique
  - validate: justify design
- threat: algorithm
  - validate: observe adoption rates

Interlocking Between Papers
- common problem: mismatches between design/threat and validation
  - ex: cannot validate claim of good encoding design with wallclock timings
  - guidance from model:
    - explicit separation into levels with linked threat and validation for each

Problem Characterization
- you assert there are particular tasks of target audience that would benefit from infovis tool support
- did you get the problem right?
  - threat: your target users don’t actually do this
  - immediate validation: you observe/interview target population
  - vs. assumptions or conjectures
  - downstream validation: adaption rates
    - you build tool; they choose to use it to address their needs

Outward Grand Challenge: TPT
- total political transparency
  - goal: reduce government corruption through civilian oversight
  - data: campaign contributions, voting records, redistricting, earmarks, registered lobbyists, military procurement contracts, street repair records, real estate assessment records, ...
  - available in theory, not understandable in practice - yet
  - infovis-complete set of problems
  - implication: need open software for open data
    - concern not only for truth, but also for justice
      - capability for analysis equally distributed in society
  - constraint: need open software for open data

Encoding/Interaction Design
- for chosen abstraction, you design visual encoding, interaction techniques
  - path following ex:
    - visual encoding: maximize angular resolution, minimize edge bends, maintain quasi-geographic constraints
    - interaction: rearrange nodes as selected to make chosen path central
  - can your encoding/interaction communicate your abstraction?
    - threat: design not effective for achieving operations
    - immediate validation: justify that choices do not violate known perceptual/cognitive principles
    - downstream validation: use system to do assigned tasks, measure human time/error costs