**Ch 7/10: Tables, Color**

**Paper: D3**

Tamara Munzner
Department of Computer Science
University of British Columbia

[Paper: D3 System]

**D3**

- **Objectives**
  - **compatibility**
  - **debugging**
  - **performance**
  - **related work typology**
    - **documents transforms**
    - **graphs libraries**
    - **infrovis systems**
  - **general notes about related work sections are a mini-taxonomy/typology**

- **Declerative tools**
  - **imperative tool/libraries**
    - **say exactly how to do it**
    - **familiar programming model**
    - **OpenGL, prefuse**
  - **declarative: other possibility**
    - **just say what to do**
    - **Protovis, D3**

- **WebGL/OpenGL**
  - **graphics library**
    - **pros**
      - **power and flexibility, complete control for graphics**
      - **hardware acceleration**
      - **multiple language bindings (C, C++, Java (w/ JGL))**
      - **cons**
        - **big learning curve if you don't know already**
        - **no vis support, must roll your own everything**
        - **example app: TreeJuxtaposer (OpenGL)**

- **Prefuse**
  - **infovis toolkit, in Java**
  - **fine-grained building blocks for tailored visualizations**
  - **pros**
    - **heavily used (previously)**
    - **accessibility**
    - **power and flexibility, complete control for graphics**
    - **pros**
      - ** stealing the vis idea from previous traits**
  - **cons**
    - **even more different from traditional programming model**
    - **example app: MizBee**

- **Protovis**
  - **declarative infovis toolkit, in Javascript**
  - **Protovis meets Document Object Model**
  - **pros**
    - **seamless integration with DOM**
    - **accessibility**
    - **massive user community, many third-party apps/libraries on top of it, lots of docs**
  - **cons**
    - **even more different from traditional programming model**
    - **example apps: many**

- **InfoVis Reference Model**
  - **conceptual model underneath design of Prefuse and many other toolkits**
  - **heavily influenced of infovis (including nested model)**
    - **aka infovis pipeline, data state model**

- **Declarative tools**
  - **imperative tool/libraries**
    - **say exactly how to do it**
    - **familiar programming model**
    - **OpenGL, prefuse**
  - **declarative: other possibility**
    - **just say what to do**
    - **Protovis, D3**

- **D3**
  - **declarative infovis toolkit, in Javascript**
  - **Protovis meets Document Object Model**
  - **pros**
    - **seamless integration with DOM**
    - **accessibility**
    - **massive user community, many third-party apps/libraries on top of it, lots of docs**
  - **cons**
    - **even more different from traditional programming model**
    - **example apps: many**
D3 features:
- Document transformation as atomic operation
- Scene changes vs representation of scenes themselves
- Immediate property evaluation semantics
- Avoid confusing consequences of delayed evaluation

Validation
- Performance benchmarks
- Page load times
- Accessibility
- Adaptability
- Everybody has voted with their feet by now!

D3 capabilities:
- Query-driven selection
  - Selection: filtered set of elements; queries from the current doc
- Site partitioning/grouping
- Operators act on selections to modify content
- Iterators via or instead transitions with attribute/series interpolations
- Event handlers for interaction
- Data binding to sceneraph elements
  - Data maps bind input data to elements
  - Error, update, exit; substructures
  - Visibility: available for subsequent re-selection
  - Sort, filter

Idiom: Scatterplot
- Express values
- Quantitative attributes
- No keys, only values
- Data
  - 2 quant attribs
  - Mark: points
  - Color: var position
  - Tasks
  - Fisheye, outliers, distribution, correlation, clusters
- Scalability
  - Hundreds of items


Ch 7: Arrange Tables

Encode tables: Arrange space

Idiom: Bar Chart
- One key, one value
  - Data
    - 1 cong attrib, 1 quant attrib
    - Mark: lines
    - Channels
      - Length as express quant value
      - Spatial regions one per mark
      - Separated horizontally aligned vertically
    - Ordered by quant attrib
      - By label (alphabetical), by length attrib (data-driven)
  - Task
    - Compare, lookup values
    - Scalability
      - Dozens to hundreds of levels for key sort

Keys and values
- Key
  - Independence attribute
    - Used as unique index to look up items
  - Simple tables: 1 key
  - Multidimensional tables: multiple keys
- Value
  - Dependent attribute, value of cell
  - Classify arrangements by key count
    - 0, 1, 2, many...

Idiom: Streamgraph
- Generalized stacked graph
  - Emphasizing horizontal continuation
  - As vertical items
  - 2 cong attrib, 1 quant attrib
    - Mark: vertical stack of line marks
    - Channels
      - Length and color bars
      - Spatial regions one per glyph
      - Aligned full glyph, lowest bar component
      - Conditional bar superimpose
    - Task
      - Same-to-whole relationship
      - Scalability
      - Several to one dozen levels for stacked attrib

VAD Ch 7: Arrange Tables

Some keys: Categorical regions
- Separate
  - Order
  - Align

Limitation: Hard to know rank. What’s the 4th most? The 7th?

Idiom: Stacked Bar Chart
- One more key
  - Data
    - 2 cong attrib, 1 quant attrib
    - Mark: vertical stack of line marks
    - Channels
      - Length and color bars
      - Spatial regions one per glyph
      - Aligned full glyph, lowest bar component
      - Conditional bar superimpose
    - Task
      - Same-to-whole relationship
      - Scalability
      - Several to one dozen levels for stacked attrib

Limitation: Hard to make comparisons
### Choose Bar vs Line Charts
- One key, one value
  - Abstract
  - 2 quant attribs
  - Mark points
  - Line connection marks between them
- Channels
  - Aligned lengths to express quant value
  - Separated and ordered by key strikethrough horizontal regions
- Task
  - Find trend
    - Sequential: marks emphasize ordering of items along key axis
  - Scalar
    - The more male a person is, the taller he is.

### Bar Chart Axes
- Labelled axis is crucial
- Avoid cropping y-axis
  - Include 0 at bottom left
  - Or slope increases
- Dual axes controversial
  - Acceptable if comparable
  - Beware, easy to mislead!

### Arrange Tables
- Connection marks
  - One key, two (related) values
  - Axes expressiveness principle
    - Implication of trend so strong that it overrides semantics
    - “The race male a person is, the taller he is.”
- Horiz + vert axes: value attribs
- Depends on type of key attrib
  - Bar charts if categorical
  - Line charts if ordered
- Do not use line charts for categorical key attrib
- Value expressiveness principle
- Connection marks emphasize ordering of items along key axis by explicitly showing relationship between axes and the area
- Task
  - Find trend
    - Sequential: marks emphasize ordering of items along key axis
  - Scalar
    - The more male a person is, the taller he is.
- Scalability
  - Hands of key levels, hands of value levels

### Connections
- Connection marks
  - One key, two (related) values
  - Axes expressiveness principle
- Implication of trend so strong that it overrides semantics
- “The race male a person is, the taller he is.”
- Horiz + vert axes: value attribs
- Depends on type of key attrib
  - Bar charts if categorical
  - Line charts if ordered
- Do not use line charts for categorical key attrib
- Value expressiveness principle
- Connection marks emphasize ordering of items along key axis by explicitly showing relationship between axes and the area
- Task
  - Find trend
    - Sequential: marks emphasize ordering of items along key axis
  - Scalar
    - The more male a person is, the taller he is.
- Scalability
  - Hands of key levels, hands of value levels

### Indexing Line Charts
- Data: 2 quant attributes
  - 1 key: 1 value
  - Derived data: new quant value attrib
  - Index
    - Plot instead of original value
  - Task: change over time
    - Principle: normalized, not absolute
    - Scalability
  - Same as standard line chart

### Gantt Charts
- One key, two (related) values
  - Data
    - 1 chart: 2 quant attribs
    - Marks: line
    - One attribute: value attribs
    - Channels
      - High position: start times
      - High length: duration
    - Task
      - Emphasis: temporal overlap, start/finish dependencies between tasks
    - Scalability
      - Dozens of keys
      - Hundreds of value levels

### Separate, Order, Align Regions
- Axes orientation
  - Rectilinear: scalability wrt #axes
  - 2 axes best
  - 3 problematic
    - More in afternoon
  - 4+ impossible
- Layout density
  - Dense: spacefilling
  - Order: align
  - 1 Key
  - 2 Keys
  - 3 Keys
  - Many Keys
  - List
  - Recursive subdivision
  - Volume
  - Matrix
  - Rectilinear
  - Parallel
  - Radial

### Correlation
- Scattered matrix
  - Positive correlation
  - Diagonal: low-to-high correlation
  - Unrelated
  - Parallel coordinates
  - Positive correlation
  - Parallel line segments
  - Negative correlation
  - All segments cross at baseline
  - Uncorrelated
  - Scalability

### Pie Chart, Polar Area Chart
- Pie chart
  - Area marks with angle channel
    - Accuracy: angle/area less accurate than line length
  - Area length also less accurate than line length
  - Polar area chart
    - Area marks with length channel
    - More direct analog to bar charts
- Data
  - 1 key strikethrough, 1 quant value attrib
  - Task
    - Part-whole judgements

### Normalized Stacked Bar Chart
- Task
  - Part-to-whole judgements
- Normalized stacked bar chart
  - Stacked bar chart, normalized to full width
  - Single stacked bar equivalents to full pie
  - High information density requires narrow rectangle
- Pie chart
  - Information density requires large circle

### Glyph Maps
- Rectilinear good for linear vs nonlinear trends
- Radial good for cyclic patterns

### Orientation Limitations
- Rectilinear: Scalability wrt Axes
  - 2 axes best
  - 3 problematic
  - 4+ impossible
  - Parallel: unfamiliarity, training time
  - Radial: perceptual limits
  - Angles lower precision than lengths
  - Symmetry between angle and length
  - Can be exploited

### Scatterplot Matrix
- Scatterplot with line connection marks
  - popular in journalism
  - Horiz: x-axis value attribs
  - Line connection marks: temporal order
  - Alternative to dual-axis charts
  - Horiz: x-axis value attribs
  - Empirical study
  - Engaging, but correlation unclear
Designing for color deficiency: Avoid encoding by hue alone

Color/Lightness constancy: Illumination conditions
- need luminance for edge detection
- fine-grained detail only visible through luminance contrast
- edge detection through luminance contrast
- 2 chroma channels
- red-green (a*) & yellow-blue axis (b*)
- "color blind": one axis has degraded acuity
- 8% of men are red/green color deficient
- beware: only pseudo-perceptual!

Bezold Effect: Outlines matter
- color constancy: simultaneous contrast effect
- lightness (L) or value (V)
- categorical can show identity
- ordered can show magnitude
- poor for encoding
- 2 chroma channels
- color constancy: simultaneous contrast effect
- perceptual processing before optic nerve
- one achromatic luminance channel (L*)
- L from HLS

Corners of the RGB color cube
- need luminance for edge detection
- fine-grained detail only visible through luminance contrast
- edge detection through luminance contrast
- 2 chroma channels
- red-green (a*) & yellow-blue axis (b*)
- "color blind": one axis has degraded acuity
- 8% of men are red/green color deficient
- blue/yellow is rare

Designing for color deficiency: Blue-Orange is safe
- color constancy: simultaneous contrast effect
- lightness (L) or value (V)
- categorical can show identity
- ordered can show magnitude
- poor for encoding
- 2 chroma channels
- color constancy: simultaneous contrast effect
- perceptual processing before optic nerve
- one achromatic luminance channel (L*)
- L from HLS

Corners of the RGB color cube
- need luminance for edge detection
- fine-grained detail only visible through luminance contrast
- edge detection through luminance contrast
- 2 chroma channels
- red-green (a*) & yellow-blue axis (b*)
- "color blind": one axis has degraded acuity
- 8% of men are red/green color deficient
- blue/yellow is rare

Designing for color deficiency: Check with simulator
- redundancy encode: Avoid encoding by hue alone
- luminance
- change shape

Color/Lightness constancy: Illumination conditions
- need luminance for edge detection
- fine-grained detail only visible through luminance contrast
- edge detection through luminance contrast
- 2 chroma channels
- red-green (a*) & yellow-blue axis (b*)
- "color blind": one axis has degraded acuity
- 8% of men are red/green color deficient
- blue/yellow is rare

Bezold Effect: Outlines matter
- color constancy: simultaneous contrast effect
- lightness (L) or value (V)
- categorical can show identity
- ordered can show magnitude
- poor for encoding
- 2 chroma channels
- color constancy: simultaneous contrast effect
- perceptual processing before optic nerve
- one achromatic luminance channel (L*)
- L from HLS

Corners of the RGB color cube
- need luminance for edge detection
- fine-grained detail only visible through luminance contrast
- edge detection through luminance contrast
- 2 chroma channels
- red-green (a*) & yellow-blue axis (b*)
- "color blind": one axis has degraded acuity
- 8% of men are red/green color deficient
- blue/yellow is rare

Designing for color deficiency: Blue-Orange is safe
- color constancy: simultaneous contrast effect
- lightness (L) or value (V)
- categorical can show identity
- ordered can show magnitude
- poor for encoding
- 2 chroma channels
- color constancy: simultaneous contrast effect
- perceptual processing before optic nerve
- one achromatic luminance channel (L*)
- L from HLS

Corners of the RGB color cube
- need luminance for edge detection
- fine-grained detail only visible through luminance contrast
- edge detection through luminance contrast
- 2 chroma channels
- red-green (a*) & yellow-blue axis (b*)
- "color blind": one axis has degraded acuity
- 8% of men are red/green color deficient
- blue/yellow is rare
Categorical Color: Limited Number of Discriminable Bins

- Human perception built on relative comparisons
- Great if color contiguous
- Surprisingly bad for absolute comparisons
- Noncontiguous small regions of color
- Fewer bins than you want
- Rule of thumb: 6-12 bins, including background and highlights

ColorBrewer

- http://www.colorbrewer2.org
- Saturation and area example: size affects salience!

Ordered Color: Rainbow is Poor Default

- Problems
  - Perceptually unordered
  - Perceptually non-linear
- Benefits
  - Fine-grained structure visible and readable
  - Alternatives
  - Large-scale structure fewer hues
  - Fine structure multiple hues with monotonically increasing luminance [eg viridis R/python]

Viridis

- Colorful, perceptually uniform, colorblind-safe, monotonically increasing luminance

Color Encoding

- Map other channels
  - Size, Angle, Curvature, ...
  - Hue, Saturation, Luminance
  - Direction, Rate, Frequency, ...

Shape

- Cyclic
- Diverging
- Ordered
- Sequential

Next Time

- To read
  - VAD Ch 8: Arrange Spacial Dots
  - VAD Ch 9: Arrange Networks
  - Paper: ALBSE-Explorer - visualizing genome sequence assemblies.
- Prepare
  - Project pitches (3 mins each)

Pitch Help

- Think of it like an "elevator pitch"
- Explain big idea
- Convincing us that it's cool/worthwhile
- Give us a sense of how fleshed out it is
- Convince us that it's cool/worthwhile
- Benefits
  - Perceptually unordered
  - Perceptually non-linear
- Alternatives
  - Fine-grained structure visible and readable

Bivariate

- Ordered
- Categorical

Map Other Channels

- Size
  - Length
- Area
- Curvature
- Volume
- Angle
- Shape
- Motion

Pitch Slides

- Next time (Oct 8) everybody must do a 3-min project pitch
- Slides required by 1pm in PDF format
- Submit to Canvas as "Pitch Slides" Assignment
- Both topic & methods
- Deadline for coming up with some concrete project idea
Projects (Reminder)

- groups of 2, 3, or 4
- permission for solo project granted in exceptional circumstances, by petition
- stages
  - milestones along the way, mix of written & in-class
  - pitches (data/task), proposals, peer project reviews
  - formative feedback
- final versions
- final presentations
- final reports
- summative written feedback for both

Projects (Reminder)

- programming
  - common case (I will only consider supervising students who do these)
  - four types
    - problem-driven design studies (target specific tasks/data)
    - technique-driven (explore design choices space for encoding or interaction idiom)
    - algorithm implementation (as described in previous paper)
    - interactive explainer (like Distill articles)
- analysis
  - use existing tools on dataset
  - detailed domain survey
    - particularly suitable for non-CS students
- survey
  - very detailed domain survey
    - particularly suitable for non-CS students

Projects: Design studies (Reminder)

- BYOD (Bring Your Own Data)
  - you (or your teammates) have your own data to analyze
    - thesis/research topic
    - personal interest
  - dovetail with another course (sometimes works, but timing may be tricky)

- FDOI (Find Data Of Interest)
  - many existing datasets, see resource page to get started
    - http://www.cs.ubc.ca/group/infovis/resources.shtml
  - can be tricky to determine reasonable task

More info

- showcase project examples
  - http://www.cs.ubc.ca/~tmm/courses/547-17F/projectdesc.html#examp
- resources (detailed list from 2015)
  - http://www.cs.ubc.ca/group/infovis/resources.shtml
  - inspiration
  - data repositories
  - data wrangling & EDA
  - visualization design
  - sharing your work
- tools directory (updated regularly)
  - https://www.visualisingdata.com/resources/