Acknowledgements and caveat

These slides draw liberally, with permission, from the ICS 104 slides of Prof. Alfred Kobsa, available in their original form at http://www.ics.uci.edu/~kobsa/courses/ICS104/_notes/start.htm

Caveat (beware): At best, PowerPoint slides are only a pale imitation of the entirety of a class meeting. In ICS 104 in particular, the lectures will cover topics beyond what appears in these slides. Don’t rely on them as a substitute for attending class.
Why are we here?

- UI horror stories
- Further examples

What is HCI?

- Narrowly: 1 user, 1 computer
  - Focus on software, layout and operation of UI
- Broadly: people and computers
  - Users’ mental processes, work practices
  - Training; collaboration; management
  - Social/organizational/health issues
Five aspects of HCI

- Human abilities (perception, memory, …)
- Technologies (windowing, mouse, VR, …)
- Design methods (prototyping, lifecycles, …)
- Evaluation methods (experiments, observation, …)
- Implementation tools and techniques (ICS 105)

ICS 104’s place in the spectrum

- ICS 161 [eternal]
- ICS 151-2-3, 141-2-3 [verifiable but subject to technology changes]
- ICS 121 [software focus]
- ICS 104 [user focus]
- ICS 131 [social focus, few experiments possible]
How did we get here?

- Once, just working (usually) was enough
- Once, the burden was on the user
- Today, you have to care: Success of a product depends on good UI
- Terry Winograd’s notes on the evolution of HCI

How do we know a UI is good?

- Analyze using “common sense”
- Develop a theory of “human cognitive processing” to predict users’ problems
- Test the UI on actual users; problems arise
  - Generalize findings, develop guidelines for avoiding problems: Usability Engineering
- But how can we know it’s problem-free?
The star life cycle (adapted from Hix and Hartson, 1993).
Usability goals

- Satisfaction
- Safety/robustness
- Efficiency (time, movement)
- Learnability
- Memorability

Usability principles/heuristics (Jakob Nielsen)

- Visibility of system status
- Match between system and real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition over recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
Conceptual/mental models

- Model: abstraction, simplification
- How user thinks of system/device/product
- Functional (how it works, how to use)
  - Should match the task
- Structural (how it’s organized, built)
  - Harder to acquire from experience
  - Useful for extension, integration
- May not match reality
  - Maybe that’s okay

Metaphors

- A package of elements
  - Analogy with real-world items
  - Draw on user’s knowledge, experience
- Metaphors only go so far
- Typewriter, desktop, book, filing cabinet, office, library, building, city, agent
Affordances: giving a clue

- What the user can see that an object does
- Chairs afford sitting; handles afford pulling
- On screen, affordances are just perceived
  - E.g., buttons and scrollbars—clear to novices?
  - Learned conventions
  - Metaphors, e.g. play/pause button

Conceptual models for activities

- Giving instructions
- Conversing
- Manipulating, navigating
- Exploring, browsing
Cognition

- Attention
- Perception/recognition
- Memory
- Learning
- Reading, speaking, listening
- Problem-solving, planning, reasoning, decision-making

Pre-attentive processing

- Before you get to conscious attention
- [Examples](#)
Learning modes (sensory input)

- Visual
- Auditory
- Kinesthetic

- Exercise from Saundra Sparling
Gestalt laws

- Whole picture, perception in context
- Proximity
- Similarity
- Closure
- Continuity
- Symmetry

Graphical coding

If we use differences in color / shape / words / line width / size / … to distinguish objects or convey meaning, what are the strengths and limitations?

Table from Maguire (1987)
Memory

• Sensory, pre-attentive
• Short-term
• Long-term (with practice, repetition)
• Recognition vs. recall
• Active vs. passive learning

Interface hardware (I/O devices)

• Appropriate for users’ tasks
• Suitable for intended work environment
• Match user’s physical characteristics
  – age, dexterity, impairments, injury avoidance
• Match user’s psychological characteristics
  – computer skills, capacity for learning
Survey of output devices

- Displays (CRT, LCD, …)
  - Wearable •• or room-scale •
- Audio (speech or non-speech)
- Tactile
- Olfactory
- Specialized for disabilities (e.g., Braille)

Survey of input devices

- Keyboards
  - QWERTY, Dvorak, chording •, thumb, numeric, arrows
- Pointers
  - Mouse, trackball •, trackpad, joystick, pen
- Touchscreen
- Speech input
- Handwriting, gestures, “Graffiti”
- Data gloves •, data suits, 3D trackers
- Specialized for disabilities
Communication & collaboration

- People work in a social context
- Rules and conventions for social interaction
  - Conversation (facilitate flow)
    - Synchronous, asynchronous
  - Coordination (facilitate action)
  - Awareness of status
- Computer-supported cooperative work
- Ethnography: Observe people and describe

Ways to look at social interaction

- Language/Action (speech acts)
  - Assert, commit, declare, direct, express
  - Winograd/Flores system
- Distributed cognition
  - Information flow between people, artifacts
How interfaces affect users

• Design to evoke positive responses
  – Feel at ease, enjoy experience, trust system
• Avoid user frustration
  – Gimmicks, error messages, overburdened users
• Anthropomorphism

Agents

• Another metaphor
• Abstract, animated, embodied/physical
• Credibility comes from
  – Agency: take action, deliver results
  – Responsiveness: infer goals, learn about user
  – Predictability on basis of character
  – Trustworthiness: consistent actions for character
HP Cooltown (ubiquitous comp.)

- What inferences does the system make?
- What connections are necessary?
- What are possible pitfalls?
- What’s your (emotional) reaction?

Interaction styles

- Command entry
- Menus
- Direct manipulation
- Form fill-in
- Natural language
Why is natural language hard?

- It’s hard to recognize speech
  - Continuous
  - Individual differences
  - Rapid speech

- Understanding (even written language) is harder still
  - Paraphrase
  - Ambiguity (disambiguate by providing context)

HCI design

- Many roles (HCI designers, graphic designers/artists, tech writers, user reps, management reps, programmers)
- Determining users’ needs, requirements
- Must precede coding
- Guidelines to follow
- Evaluation throughout process
ID process (Preece)

- Identify needs, establish requirements
- Develop alternative designs (unlike software design)
- Build interactive versions of designs (prototypes)
- Evaluate designs
- Iteration is inevitable

ID process model (Preece)
Identify all the stakeholders

- Primary (directly interacting) users, but also:
- Managers
- Recipients of product’s results
- Purchasing decision makers
- Competitors’ users

Human users are diverse

- Physically (hand size, height, strength, coordination, disabilities)
- Cognitively
- Culturally
- Experientially
Design principles (indep. of style)

- Consistency (internal, external)
- **Advance information**
- **Immediate feedback**
- Easy reversal (undo)
- **Error prevention, help**
- Minimal short-term memory
- User control
- User diversity, **personalization**
- Shortcuts for experts
- Online help
- Learning aids

Advance information

- What’s possible now? What will happen next? What can I do now?
- Prevent errors, unexpected results
- Guidelines
  - Give visual indicators, not just text •
  - Distinguish unselectable menu items, objects
  - Change cursor shape •
  - Show submenus on rollover •
  - Show data entry format • •
  - Indicate long operations, ask permission •
Feedback

• User action, system reaction (ideally < 0.1 sec)
• Guidelines
  – Highlight items on rollover
  – Mark selected items
  – Show path in navigation hierarchy
  – Report errors immediately
  – Use status or progress indicators
  – Use visual, auditory, and tactile modes
  – Make reaction time uniform

Undo

• Encourages users to explore functionality
• Guidelines
  – Special-purpose undo (e.g., backspace) supplements general
  – Try to make everything undoable (external effects clearer to users than internal)
  – Multiple undo (undo/redo or linear sequence)
Error avoidance

- Provide advance information
- Keep dangerous controls away from frequently used ones
- Warn users of irreversible effects; don’t make them the default; request confirmation
- Turn safety options on by default
- Recognize errors and react ASAP

Error messages and actions

- Explain nature of problem, how user can solve it (at least with correct examples)
- Describe in terms of user’s task
- Use polite language
- On crash, give opportunity to save
- Support force quit and relaunch
Shneiderman’s error message guidelines

- Avoid “fatal,” “invalid,” “bad”
- Avoid ALL CAPS, cryptic numbers
- Give control over audio feedback
- Give precise messages
- Provide context-sensitive help

Adaptable vs. adaptive menus

- Adaptable: user (or admin) can change (shortcuts, hide/delete/move/duplicate items)
- Adaptive: automatic change (e.g., based on usage frequency) violates constancy
Online help

- Available, consistent for all system functions
- Including currently unavailable options
- Situation-sensitive and concrete
- Written in terms of user’s task
- Not obscuring relevant items; movable
- Initially short with details on request
- Good ID reduces need for explicit help

Tours, tutorials, manuals

- Tour should be short, hit highlights
- Encourage active learning (e.g., user actions, quizzes), address users directly, give examples
- Manuals are last-resort, comprehensive sources
- Use tech writing specialists where possible
Command interface guidelines

• Use action words, verb first (move a b), direct object as first argument
• Use congruent names (advance/retreat, not move/back)
• Allow abbreviations, syntax flexibility, aliases
• Provide command history (edit, re-enter recent commands)
• Multiple args, wildcards, macros, scripts

Menu interaction overview

• Activities: navigation, selection, activation
• Selection: mouse, keys, key + return, touch
• Types of menus:
  – Text, graphical, combination
  – Linear, tabular
  – Static (e.g., menu bar), pull-down, pop-up
  – Isolated, connected (hierarchical)
  – Pie menus
Menu item guidelines

- Head/title: short, meaningful, centered, upper/lower case, clean design
- Show: selectable items, non-selectable items, already-selected items, submenu availability, how to select (besides mouse)
- Entries: short, meaningful, distinguishable (most significant word first)
- Shortcuts (first letter); external consistency

Menu length, item order guidelines

- Keep short for beginners
- Group according to task
- Put frequent items near top for beginners
- When multiple selection allowed, group frequent combinations
- Separate dangerous items from frequent ones
- As last resort, use alpha, time, numerical order
Menu selection time

- Selection time = search time ("S") +
  - if using keyboard, time to press key(s)
  - if using pointer, positioning time ("P") +
    activation time (e.g., to click mouse)
- For beginners searching menu size $n$:
  - if label unknown, examine all items, $S \sim n$
  - if label known, search linearly or randomly; in either case, $S \sim n$
- Experts can remember position; $S$ is constant

Menu positioning time ("P")

- Fitts’ Law: $P = C1 + C2(\log2(2D/W))$
  - $C1$ and $C2$ are constants depending on device
  - $D$ is distance to the target
  - $W$ is size (width) of the object
- The time to acquire a target is a function of the distance to and size of the target
- Screen edge: no chance to overshoot
Menu dynamics guidelines

- Highlight item under cursor
- Show submenus of item under cursor
- Maintain indication of selected items
- Allow leaving without any selection
- Maintain positional constancy (grey out)
- Maintain visibility against all backgrounds

Menu hierarchy guidelines

- Avoid deep nesting
- Top, bottom level menus can be longer
- Longer menus better when under pressure
- Avoid scrolling
- Construct hierarchy by theme
- As before: show submenus, moderate length, external consistency, shortcuts to deeper items
Graphical menu guidelines

- Make items (icons) recognizable, distinct
- Emphasize global properties (form, color, size) over fine details
  - Abstract icons faster than concrete, text
- Give similar icons to similar objects/functions
- Use easily understandable (or learnable) icons
- Textual labels help beginners, infrequent users

Needs and requirements

- Want to understand users, task, context
- Kinds of requirements
  - Functional: what it does
  - Non-functional: e.g., memory reqts, delivery time
  - Data: what info is stored, in what form
  - Environmental: physical, social, org’l context
  - User: what users will be like
  - Usability: what balance of factors
Gathering requirements data

- Questionnaires
- Interviews
- Workshops and focus groups
- Naturalistic observation
- Studying documentation
- Choose based on kind of task, on data provided, cost, time required, what analyst needs to know

Problems gathering data

- Identifying, involving stakeholders
- Availability of key stakeholders
- Ownership of reports, versions
- Communication (with users, within team)
- Domain info hard to get or articulate
- Political problems in organization
- Changes in economic or business situation
Data gathering guidelines

- Focus on identifying stakeholder needs
- Involve all stakeholder groups, more than one person from each
- Combine techniques; use props, prototypes
- Run a pilot session (user testing!)
- Decide how to record data (audio, video, notes)

Data analysis

- Don’t let data get stale
- Do this iteratively, too
- Decide which tools, how much formalism
  - Scenarios (narrative)
  - Use cases (describe interaction with system, alternative paths)
  - Essential use cases (more abstract: user intention, system responsibility, …)
  - Hierarchical task analysis
Prototyping

• Present ideas for evaluation without getting in too deep (in time, money, commitment)
• Use sketches, storyboards, slide shows, video simulations, physical objects, mock-ups, skeleton software
• Build model of work flow, task design, screen layout, information display, difficult or critical aspects

Low-fidelity prototyping

• Unlike the final form
• Quick, cheap, easily changeable
• Examples
  – Sketches
  – Index cards
  – Storyboards
  – Sticky notes
High-fidelity prototyping

- Same materials as final product
- Realistic-looking results
- Tools include MacroMedia Director, Dreamweaver, VB, …
- Users’ expectations and focus?

Prototyping considerations

- Models necessarily omit detail
- Horizontal vs. vertical approach
- Other tools
  - Denim system (sketches with hyperlinks)
  - Scripting languages (e.g., Tcl/Tk)
User-centered design

• Affects product acceptance and success
• Makes users active stakeholders
• Manages expectations
• Gets head start on training
• Communicates without sales hype
• Provides vital information about needs, requirements, usability

Aspects of user-centered approach

• Early focus on users (cognitive, behavioral, attitudinal characteristics) and tasks
• Actual measurement: observe, record, analyze users’ reactions and performance
• Iterative design: find problems, fix them, test again
Time to try it

- Design a web-based system for reserving movie or theater tickets
- Don’t be constrained by existing systems
- Pair up: you will both be users and both be designers
- Determine your context, requirement, tasks
- Design two alternatives with (low-fidelity) prototypes

Evaluation

- Formative vs. summative
- Four paradigms
  - Informal feedback
  - Walkthroughs
  - Field studies
  - Predictive evaluation
- Goals: find problems or new opportunities, check conformity with guidelines, standards, requirements, …
Evaluation planning

- Determine high-level goals
- Explore questions to be answered
- Choose evaluation paradigm and techniques
- Identify practical issues
- Decide how to handle ethical issues
- Evaluate, interpret, and present results

Designing a study

- Reliability: results are repeatable
- Validity: measuring what you want to measure
- Biases: not introduced by process
- Scope: breadth of findings’ applicability
Interviews

- Structured/scripted vs. unstructured/open-ended
- Avoid long, compound questions
- Avoid unfamiliar terms
- Avoid questions that embody assumptions
- Avoid biases
- Intro, warm-up, main body, cool-off, closing

Questionnaire development

- Paper vs. electronic, closed vs. open-ended
- Checkboxes, rating scales, prose responses
- Design
  - Start off-line even if goal is electronic
  - Questions all positive, all negative, mixed
  - Pilot-test questions for clarity, sufficient space
  - Consider analysis
Increasing questionnaire response

- Expect 20%–40% rate (less online)
- Make purpose clear
- Promise anonymity
- Design well
- Offer short version
- Provide stamped return envelope
- Follow up
- Provide incentive

Expert critiques

- Heuristic evaluation w/ guidelines (Nielsen)
  - Brief 3–5 experts
  - Each works separately 1–2 hours, two passes
  - Debrief experts together
- Cognitive walkthrough
  - Tell expert assumptions, context, task
  - Expert walks through prototype w/ usage scenarios
  - Will user know what to do? Will user see correct action is available? Will user understand response?
Usability walkthroughs

• Make an explicit test scenario (test plan)
• Test the test (pilot study)
• Recruit subjects
• Conduct test
• Debrief subjects

Roles in walkthroughs

• Greeter gets user settled
• Facilitator talks to user during testing
• Computer (a person) manipulates interface elements
• Observer(s) take notes
Le mieux est l’ennemi du bien.
(The best is the enemy of the good.)

— Voltaire
[“Dramatic Art” in *Philosophical Dictionary*, 1764]

Observing users

• In the lab
  – Walkthroughs with low-fi prototypes
  – Instrumented sessions with higher-fi systems
• In the field
• Consider, as always, who’s involved, their goals, their actions, their feelings, the relevant objects and events
User testing

- A part of usability testing
- Smaller-scale, less formal, more focused than full-blown usability research
- Can be quantitative: time to complete, number of errors, number of help requests, number of users completing task successfully
- Can include keystroke-level monitoring
- How many users?

Direct manipulation

- GUI objects representing task objects/funcs
- Pointing device
- Based on consistent metaphor
- Congruent operations, always available
- Immediate feedback
- Form of icon, cursor on rollover indicates possible operations
Form design guidelines

• Allow entry in tables or labeled data fields •
• Left-align labels, fields, columns in tables •
• Arrange sequences in columns •
• Use meaningful, unambiguous labels
• Mirror layout of paper source document
• Use adequate white space • •
• Tell user expected form of data; indicate if required
• Allow enough space for expected data

Data entry in forms

• Tab or return should move to next field
• Fill fields with default/most recent/inference
• Allow entry in arbitrary order
• Allow abbreviation/expansion
• Show alternatives if entry not unique •
• Don’t supply dangerous values as auto entries
• Detect, indicate, explain errors; allow multiple corrections
• Don’t require re-entry of correct data
Evaluation exercise

• Start with your ticket reservation system
• Decide on the one aspect that most needs testing/evaluation
• Design and describe an evaluation plan to test that aspect (goals, questions, techniques)
• Outline/sketch on one page
• Present to another group for critique
• Turn in page with names of (new) group

Form design exercise

• In your reservation system:
  – Identify at least one place where the user needs to enter data
  – Carefully design a form for entering that data, using what you now know
  – Sketch that form on paper, informally but cleanly
  – Turn in the sketch with your group’s names
General screen guidelines

- Reflect structure of task, not of implementation
- Group info for coherent subtask on one screen
- With multiple, related screens
  - Use same headlines
  - Present necessary info on each screen in same place
  - Allow navigation to previous screen, access to help, ability to exit subtask or whole program

Special screen areas

- Title at top, distinguished
- Can use bottom for status info, explanation, warnings
- Logos typically upper left or upper right
- Clocks (no seconds, no ticking)
Screen layout

- Use proximity, alignment, consistency, contrast
- Use adequate whitespace (60%–80%)
- Alternatives to whitespace for grouping:
  - Lines
  - Boxes
  - Colored/shaded backgrounds

Guidelines for windows

- Windows make it easy to distinguish
  - applications
  - info or objects within applications
  - events stacked in time (e.g., errors, dialogs)
- Tiled windows easiest for beginners, but overlapping ones are far more flexible
- Signal which window is on top or active
  - partial occlusion
  - 3D effects (shadow, lighting) or graying out
Typography

• HCI for documents, affects effectiveness
• Display type vs. body type
  – Quick recognition of letters, words, lines
• Great control now in user’s hands
  – “With more power comes the power to mess up in new and more spectacular ways.” —DGK
• Differences between displays and paper
• Less designer control for text on the WWW

Typography terms

From McCracken and Wolfe, *User-Centered Website Development*
Font size: baseline to baseline

From McCracken and Wolfe, *User-Centered Website Development*

Line spacing (leading) matters

- The point size is also the distance between successive baselines, assuming no space is added between lines as is commonly done. It is clear that without the "little bit," the letters in successive lines would touch, seriously hampering legibility.

From McCracken and Wolfe, *User-Centered Website Development*
Serif vs. sans serif

Times New Roman
Georgia
Arial
Verdana

Monospace vs. proportional spacing

f("It’s")-1; // Courier
f("It’s")-1; // Andale
f("It’s")-1; // Times
f("It’s")-1; // Georgia
Typeface guidelines (characters)

- Mix upper and lower case
- Choose proportional spacing over monospace
- Use fonts with varying stroke width
- Choose serif over sans-serif fonts
  - But on the web …

Margin justification

Rag-right

Flush right
Typography/text guidelines

- Keep lines short (max chars 35? 50? 70?)
- Don’t justify margins
  - Extra white space
  - Justification and monospace fonts
- Consider extra leading
- Minimize number of fonts
- Use emphasis minimally

Hypermedia and the WWW

- Nodes (info in many media)
- Visible links to other nodes
- HCI view: navigation between pages, information presentation, multimedia layout
- More than just HCI design • • •
Hypermedia organization

- Shape can be hierarchical, linear/multipath, network/web, matrix
- Unrestricted linking makes orientation hard
- Network/web structures hard for beginners

HCI for Web (Farkas & Farkas)

- 1.1: All links indicate they’re links
- 1.2: Help viewers notice links
- 1.3: Links clearly indicate destinations
- 2.1: Effective breadth and depth in hierarchies
- 2.2: Add secondary/shortcut links where approp.
- 2.3 Allow branches to converge where approp.
- 2.4 Reveal underlying information structure
HCI for Web (Farkas & Farkas) 2

• 3.1: Clear, conspicuous orientation at top
• 3.2: Support exploration
• 4.1: Use site maps for structure and direct access
• 4.2: Provide search facility or index for direct access
• 4.3: Provide links to home page throughout

“Information scent”

• Link should “smell right” to user: confidence before clicking, feel closer afterwards
• Practical measure (Spool 1998):
  – Ask users before clicking what they think they’ll get
  – Ask how confident they are (–2 to +2)
  – Ask users after clicking if they felt closer (–2 to +2)
  – Add the two figures
  – Accumulate those sums as you go from link to link; the result should keep increasing
Human vision for color

• About 180° of arc
• Fovea (highest-resolution area) just 2° of arc; 75% of human visual operations
• Light reception happens in retina (back of eye); two kinds of photoreceptors
  – Rods: Degrees of brightness; not in fovea
  – Cones: Colors; in fovea mainly.

Three kinds of cones

• Red-sensitive (64%): many in fovea
• Green-sensitive (32%): many in fovea
• Blue-sensitive (2%): not in fovea; evenly distributed over retina
Color depends on context

• What other colors are nearby
• Cultural connotations

Guidelines based on physiology

• Avoid blue for small objects
• Blue is good for background
• Neighboring objects should not differ just by amount of blue
• Put small red and green objects in center
• Add other emphasis to red and green warning signals on the periphery
• Black, white, yellow, blue OK on periphery
Recommended uses of color

- Emphasis, grouping (especially as background)
- Coding discrete or continuous data
- Distinguishing window types
- Visual separation of overlapping graphics
- Depth in 3D graphics (red closer, blue farther)
- Warnings, status reports
- Increasing attractiveness (within guidelines)

Users with disabilities

- Manual/dexterity; visual; auditory; cognitive
- Various legal requirements to make software and websites accessible:
  - Americans with Disabilities Act (ADA)
  - Institutional directives (e.g., UCI’s Electronic Communications Policy)
Considerations/guidelines (manual)

• Provide access by keyboard and by pointer
• Provide alternative to simultaneous keystrokes
• Provide alternatives to voluminous data entry (defaults, completion, aliases/shortcuts, cycling through until user hits key to stop)
• Provide special devices (head mouse, foot mouse, suction tube, speech recognition)

Considerations/guidelines (visual)

• Allow magnification
• Color-code only large areas
• Avoid frequent color switches
• Color deficiencies (“color blindness”): 8% of European-descended males (0.5% of females) see red/green as medium gray
  – Design for monochrome first; add color for redundancy; at least don’t just differ by red vs. green
• Special devices, software, design guidelines
Hand-held devices

- Often used without watching (“eye-less”), so highly tactile keypads necessary
- Highly targeted info (personalization)
- New interaction paradigms:
  - Motion-invariant displays •
  - Touchscreen dragging (page-flipping) •
  - Hand mirror metaphor •
  - Keyhole/flashlight metaphor

Virtual reality

- Three-dimensional objects and environments
- Multi-sensory input (visual, auditory, haptic)
- User feels immersed: user controls scene movement, receives feedback
- Integrated technologies: displays, position sensing for head/hand, force feedback, audio input/output
Virtual reality applications

- Scientific exploration
- Architectural exploration
- Augmented reality
- Training
- Virtual co-presence (meetings, entertainment)

Improving immersion

- Match input from at least two sensors
- Provide high refresh rate
- Minimize response time
- Provide stereoscopic vision
- Provide three-dimensional sound
Information visualization

- Allows understanding of huge amounts of data
- Allows perception of unanticipated properties
- Reveals problems with data itself
- Facilitates understanding of large- and small-scale features of data
- Facilitates hypothesis formation

Ware, *Information Visualization*

---

Edward Tufte (Yale)

- *The Visual Display of Quantitative Information*
- *Envisioning Information*
- *Visual Explanations*
Tufte on graphical integrity

- Make physically measured representation proportional to quantity being represented
- Use clear, detailed, thorough labeling
- Show data variation, not design variation
- Deflate and standardize monetary figures
- Dimensions in representation ≤ dimensions in the data
- Don’t quote data out of context

Tufte on producing data graphics

- Above all else show the data
- Maximize the data-ink ratio
- Erase non-data-ink
- Erase redundant data-ink
- Revise and edit
Tufte on graphical excellence

- Well-designed presentation of interesting data—substance, statistics, design
- Complex ideas communicated with clarity, precision, efficiency
- Greatest number of ideas in the shortest time with the least ink in the smallest space
- Multiple variables presented
- Data represented truthfully

What would Tufte say about reducing his principles to bullet points?