A visual history of women’s tennis

Serena Williams was toppled from the number one spot following the US Open, just as she matched Steffi Graf’s all-time record of 186 consecutive weeks atop the WTA rankings. Shown below is every player ever to have appeared at number one at least once since the rankings were created in November 1975.

For players ranked below number one before 1983, rankings are based on year-end standings.

Sources: WTA, FT research

Graphic by John Burn-Murdoch / @burnmurdoch
Quiz

Average score: 72

Lowest score: 23

Highest score: 111

3 people got full points for the extra credit question.
Project 1

9 people got an A
21 people got a B
9 people got a C
1 person got a D
3 people got an F

Shiwangi will send you grades and very brief comments via Piazza by tomorrow
Validation

For this first project, Shiwangi and I served as the experts who evaluated your project.

But in actuality, you should be able to evaluate your own projects.

Chapter 4 provides an overview of ways to validate different aspects of your project.
Aspects of the Project

- Domain situation
- Data/task abstraction
- Visual encoding/interaction idiom
- Algorithm
Domain Situation

- community of users / experts
- specific vocabulary
- specific questions that are important
- particular types of data
- particular methodologies
- specific workflows that evolved for historical reasons to address important problems in that area
- often times experts are used to a particular way of doing things, but open to something that makes it easier
Task and Data Abstraction

- What visualization tasks would support the communities questions and would: make their lives easier, or help them to ask deeper questions about their data?
- What are the important aspects of the data?
- Could their data be simplified, transformed into another format?
- What are they hoping to see in their data? If they saw it, what would that let them do? What decisions could they make?
Encoding and Idiom

- How can you present their data – or some subset of or transformation of their data – so that the visualization supports their tasks?
- Can you make it easy to show all of the important data, while hiding all of the unimportant data?
- Can you make it is easy to find and analyze interesting patterns in the data?
Algorithm

- Is your software fast enough to support complex analysis tasks using your visualization encodings and interaction idioms?
“A Nested Model”

The success of each aspect depends on the how well it fulfills the expectations of the outer level, and inspires expectations of the inner level.

Answers to questions asked at one level are inputs to the next, inner level.
Domain -> Task

Domain – What questions are being asked, or should be being asked, by experts working with a particular types of datasets? What problems are they trying to solve?

Accurately determining these questions will lead to appropriate definitions of specific visualization tasks.

Task – “Your goal is to determine which data type supports a visual representation that addresses the user’s problem.”
Tasks -> Encodings / idioms

**Task** – What information is useful, is salient, is necessary to help user’s think about and/or answer their questions. What visualization tasks will help them do this.

**Visual Encoding / Interaction Idiom** – What’s a good way to present this information so that the visualization tasks can be effectively carried out?
Encodings / idioms

Visual Encoding / Interaction Idiom

Choose effective marks and channels for your dataset type.

- “Intuitive” – but what exactly does this mean?
- nothing “gets in the way”
- “affordance” – it’s obvious what to do
- no visual clutter to confuse the user or obscure the data itself or the execution of the task
- easy to perceive the meaning of the data and to find patterns in the data
Problematic encodings I noticed

- blinking
- abrupt changes or too much motion
- surprises / unexpected visualizations
- lack of legend or instructions
- “chart junk”
- mouseovers caused overlapping tooltips
- mouseouts caused information to vanish
- upside-down text
- distracting colors, shapes
Encodings / idioms -> Algorithms

Make sure that you can quickly visualize and effectively enable the encodings and idioms you’ve designed.

- If your visual encoding and interaction idiom involves letting the user click on a hierarchy of circles and then zoom into more information in each successful circle, then make sure you can program that interaction.

- If your visualization involves 3D volume rendering, then make sure it’s computationally efficient enough to render on a desktop, etc.
Domain situation
Observe target users using existing tools

Data/task abstraction

Visual encoding/interaction idiom
Justify design with respect to alternatives

Algorithm
Measure system time/memory
Analyze computational complexity

Analyze results qualitatively
Measure human time with lab experiment (lab study)

Observe target users after deployment (field study)

Measure adoption
Why does it go wrong?

- **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
Threats and Validations

If it *does* go wrong...

Maybe need to re-think your design...

- Maybe have a great idea for some complex interaction idiom, but it is too computationally difficult to develop in time for a release data...
- Maybe have a nifty way to define the user’s data, but they are used to thinking of it in a certain way and so reject your task definitions.
Threats and Validations

We’re mostly going to look at levels 2 and 3 in this class: defining the visualization task and choosing the visual encoding and interaction idiom.
Threats and Validations

**TASK**

**threat:** You’ve chosen the wrong task or data abstraction. That is, solving a problem that doesn’t need to be solved.

**validation:** ask an expert (or teacher) for feedback; conduct a field study to see if this task is indeed being attempted or is important (evaluation can be done before or after your work)
Threats and Validations

ENCODING / IDIOM

threat: You’ve designed a visualization that doesn’t make sense or isn’t effective.

validation:

Justify the design – why this way and not some other way?

Conduct a usability study – do people understand the visualization?

Conduct a quantitative study – prove its effectiveness...
“Effectiveness”

- Accuracy
- Speed
- Enjoyment

Can a user perform data manipulation tasks quickly and accurately?  
*low level...*

- Supports task

Can a user use visualization to help answer a domain specific question?  
*high 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
  - **Validate**: 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
For Thursday

Continue to practice D3.js

Will introduce Project 2 in class on Thursday

Shiwangi will lead programming lab in class on Tuesday