CS 171: Process Book

Ross Rheingans-Yoo

I. Overview / Motivation

I've been dancing with the Harvard Ballroom Dance Team for a year and a half, and this year, I've served as the team's secretary. In that time, I've traveled to more than a dozen collegiate competitions with HBDT, each of which includes several hundred competitors, hundreds of rounds of competition, and thousands of judging marks (which are, conveniently, posted online).

This year, I've been working with the team's executive board to better understand our team's performance at competitions (to inform coaching/training decisions, mostly), which is no small task, as the Harvard team is sometimes in excess of a hundred dancers! To date, my methods have included taking notes manually during/after a competition, compiling the data in Microsoft Excel or similar, and using rudimentary graphing tools to explore trends.

I plan to apply my newfound skills with D3 to create a few dynamic visualizations that draw from competition results scraped from online sources, aiming to answer a few questions that the HBDT executive board has been asking recently.

II. Related Work

While online reporting of competition results is relatively new in collegiate ballroom dance, ballroom dancer and web developer Cloud Cray has done some work writing a web-scraping toolkit for O2CM[1], one of the most popular registration/results websites. This work[2] will save me a lot of difficulty in compiling /parsing results, and allow me to focus on visualizing and presenting the actual data.

Mr. Cray has been running some statistical analysis and published his findings as an infographic on his blog[3], but has mainly focused on large-scale trends across the US. By contrast, I'd be interested in exploring only a handful of competitions (and a handful of teams), which, nonetheless, will still represent a few thousand judging marks to be wrangled and visualized.

III. Questions

After discussion with a few dancers and other members of the HBDT executive board, as well as Mr. Cray, I've identified a few interesting questions:

- Are the recent changes in coaching working? (Are we getting better between years?)
- What particular styles/dances/events are we best/worst in?
- Which schools are doing better than us, and in which styles/dances?
- What correlation exists between competitive performance and long-term commitment to HBDT? (Is early competitive success a good predictor of which dancers will stay on the team?)

The last, of course, requires cross-comparison of competitive results with outside information (HBDT membership rolls), and is probably outside the scope of this project, but the others could be addressed by a graphical interface for tracking the performance of selected dancing couples across multiple rounds of competition, and comparing results across events/competitions/years.

Additionally, I expect that the visualization might see use by dancers, captains, or coaches

[1] http://www.o2cm.com/

[2] https://github.com/CloudCray/DanceMarkScraper

[3] http://i.imgur.com/gJBUpJJ.jpg

wishing to understand competitors' results in larger context of a competition, and so I plan to provide the ability for users to focus on data detail when desired, down to the level of individual rounds or even individual competitors.

IV. Data

A. Acquisition / Processing

I will be using competition results data scraped from O2CM, using the previously-mentioned DanceMarkScraper. This includes almost all competitions attended by HBDT in the past several years, and is by far the most popular competition results reporting site in the US. I've considered writing additional scraping tools to scrape similar data from other results reporting sites, but weighing the effort required to process their (often fragmentary) data into the forms available fromt O2CM scrapes against the number of additional competitions that it would make available, I've determined that it's not worth the effort, and that I intend to focus solely on data scraped from O2CM, which covers all but a handful of competitions Harvard has attended in the past several years

The scraped results from a single large competition, written out in CSV format, comprise approximately 5MB, and the scraping and writing-out process takes approximately 3-5 minutes using the Dance MarkScraper scripts (written in Python). This is acceptable for my purposes, and I expect scraping a hundred or so competitions to take most of a day and less than 1GB of storage. This 'flat data' represents every judges' mark as a single line of a few dozen values, which is clearly suboptimal, but at least reflects all possibly relevant data in a single line-based format.

From there, I use my own Python scripts to aggregate the flat data into a hierarchical format (events/competitors/rounds/results) and write it out as JSON, which takes negligible time (less than a minute for a large competition), and 2̃00KB per file (so less than 100MB in total for a few hundred such sets). This is an acceptable size and format for use in the web front-end, and all further data processing will be live and ad-hoc within d3.js scripts.

To be clear, I've defined the data structures (both filetypes and in-memory structures) I will be using, and processed one dataset fully into them using automated scripts. I've included a sample in the submitted Git[4][5], which covers one large competition (approximately as large as any other single competition in the final dataset).

B. Format

I'll use some jargon-ish terminology in describing the structure of competition results data, so it might be useful to describe the rough format of ballroom competition, and how results data are structured.

Within one *competition*, several dozen *events* are made up of several (between one and six) *rounds*. Each event may be contested by between one and several dozen *couples* (or competitors). In each round up until the final round, several judges will make a judgement about each couple — voting either to recall that couple to the next round or not. If the number of votes to recall exceeds some threshold (different round-to-round and between events), the couple is recalled to the next round; otherwise, they are eliminated and will stop dancing. In the final round, a more complicated system is used to sort all finalist couples in order.

Thus, couples' results in an individual event are (roughly) based first on the number of rounds reached, and second on the number of recall votes (or *marks*) received in their last round danced. The system, of course, is slightly more detailed than this in practice, but this explanation at least explains the structure of the data I'm visualizing.

V. Exploratory Analysis

I've performed relatively little exploratory data analysis thus far, as my efforts in building

[4] flat_comp.csv

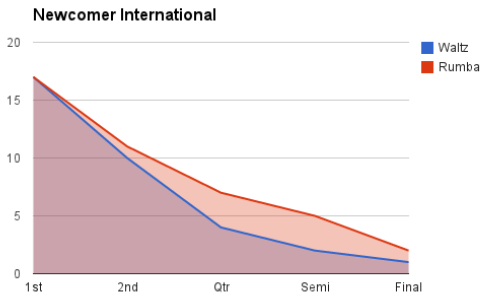
[5] res_data.json

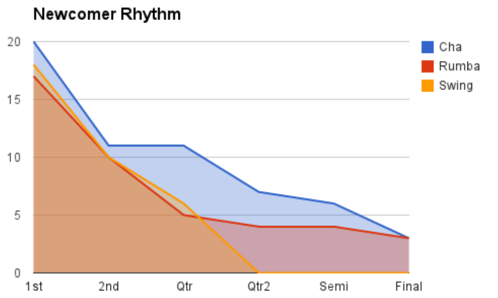
the first prototype have mostly focused on the ability to display results data from a single event in an interactive format. In the next week, I expect to use this single-event view to perform exploratory analysis in comparing different events (across levels, styles, competitions, or years) in order to discover interesting trends to visualize.

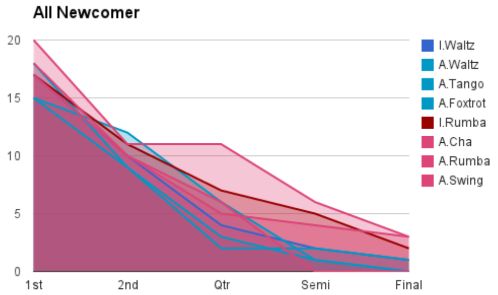
VI. Design Evolution

A. Pre-design

The initial idea for this project began with some rudimentary graphs I made over winter break in Microsoft Excel based on competition data collected manually this past fall: Here, I was graphing the total number of Harvard competitors remaining in the competition after each successive round. (In the second graph, for example, note that Harvard dancers did much better in Rumba than in Swing in the Rhythm style, but only in later rounds, whereas they did better in Cha Cha than Rumba early on, but the two ended near the same in final results.)







Despite (or perhaps, due to) its conceptual simplicity, this can be a powerful visualization for comparing the aggregate results of small groups against each other (either the same group in different events, two groups in the same event, or the same team across different competitions or years), and I've played around with graphing some of these comparisons. However, the labor required to manually look up and input results quickly became prohibitive.

B. Initial Design

When I took up this project for CS 171, I built on some of these preliminary ideas using design principles learned in class. In particular, I needed to be aware of the principle of maximizing data-to-ink ratio, the effective use of color, and the concept of detail-on-demand. While I still liked the overlaid-area-graph concept, I wanted to add a stacked-graph option (for comparing one or more teams against the field at large in a single event).

Of course, the possibility for interactivity represents a significant addition to static graphs, and I wanted to make effective use of this capability in d3. However, I didn't have a concrete idea of how I wanted to incorporate interactive elements until Design Studio; I've described the design evolution of the interactivity dimension below.

C. Design Studio

I completed Design Studio #3 in class on 10 April 2014, with Jack Davison (the rest of his group was otherwise occupied), and peer-reviewed his group's project on visualizing trends in Q guide data. We shared both design sketches and preliminary prototypes for critique and discussion. Several points that we discussed shed light on important design considerations in my project, so I've listed the major ones below:

1) Audience: In discussion, Jack and I realized a fact that might be useful for both of our projects: our prospective userbases were made up of two relatively-distinct types:

- top-down users, who have relatively little domain-specific knowledge, but are browsing for interesting trends.
- bottom-up users, who have domain-specific knowledge and are looking to draw connections between known facts and additional interesting data features.

In my case, specifically, the former corresponds to users who don't know (or don't care) about individual competitors, but are interested in aggregate statistical results, and the latter corresponds to users who know some handful of competitors by name, and want to explore their results in the context of the field at large.

As these two have different interests and needs, it's important to consciously consider and balance the needs of both groups. Specifically, taking this into consideration led me to place a greater focus on interactivity as a method of revealing data to bottom-up users who might be interested in it.

2) Data-to-Ink Ratio: One of the major critiques Jack had of my design sketches was that their data-to-ink ratio was strikingly poor (or perhaps more properly, their data-per-square-inch ratio). He suggested subdividing the stacked area charts (of one or two groups) into smaller regions making use of color scales to increase data density.

A few premilinary computer sketches later revealed that this was a brilliant suggestion, and that a segmented bar-chart design is a fantastic way to display data where an individual competitor's result is made up of temporal data (what round reached) as well as an aggregate of preceding rounds. I expect to use this as a default view, and to use the previously-described area/stream graph as a secondary, toggle-able option.

3) Tabular Detail View: Ironically enough, as we discussed, the high-density, detail-focused view sparks a desire in users for yet more detail data. If a particular round appears unusual or interesting, a user will want the ability to explore it in more depth. Since we have the data already available, it makes sense to add a sidebar-based detail view that allows for comparison of competitors within a round. (Preliminary sketches seem to indicate that this information could be easily conveyed in a tabular format, not unlike Homework 1.) I will be exploring this possibility further in later prototypes.

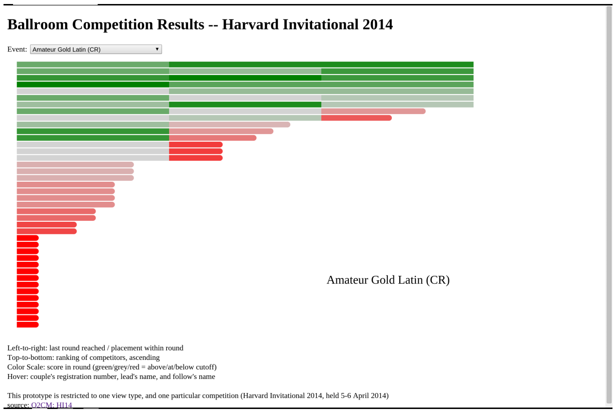
4) Overall area vs. Percentage area: Jack suggested that, in views comparing the number of competitors remaining from a selected group to the total, both absolute-scale and relative-scale measurements might be informative. As such, I decided to add a toggle-option to switch between absolute-scale (number of in-group competitors remaining, stacked with total remaining competitors) and relative-scale (% of in-group competitors, compared to total competitors, per round) in area-graph views of competitors remaining over time.

VII. Implementation

A. *"Competitors" view, single event*

Based on feedback in design studio, I decided to use a relatively information-dense view that displayed all competitors in a particular and their individual round-by-round results for a default view (and also for my first prototype). In particular, the "Competitors" view[6] presents a single event in terms of all of its competitors, their round-by-round results, and overall placement.

[6] prototype: comp_vis.html



Ballroom Competition Results -- Harvard Invitational 2014

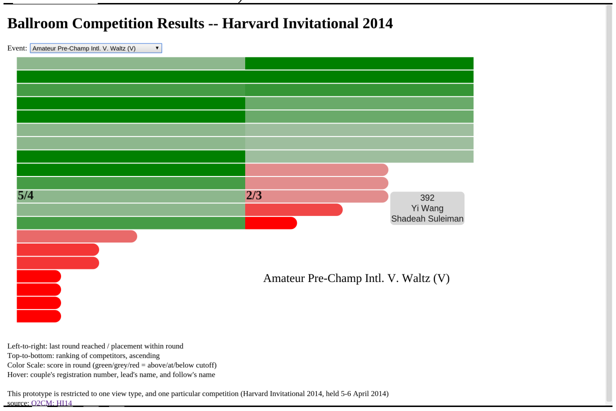
Event: Amateur Gold Latin (CR)

Left-to-right: last round reached / placement within round
Top-to-bottom: ranking of competitors, ascending
Color Scale: score in round (green/grey/red = above/at/below cutoff)
Hover: couple's registration number, lead's name, and follow's name

This prototype is restricted to one view type, and one particular competition (Harvard Invitational 2014, held 5-6 April 2014)
source: O2CM: HI14

Here, each bar represents a single couple competing in the event (here, Amateur Gold Latin Cha/Rumba), the length represents their success (large steps represent the number of rounds recalled; within a round, longer bar means closer to being recalled to the next round), color represents score within a particular round (on a divergent scale, with green/grey/red representing marks above/at/below the cutoff for recall), and vertical position simply sorted by final result.

On mouseover, more detail is revealed:



Ballroom Competition Results -- Harvard Invitational 2014

Event: Amateur Pre-Champ Intl. V. Waltz (V)

Left-to-right: last round reached / placement within round
Top-to-bottom: ranking of competitors, ascending
Color Scale: score in round (green/grey/red = above/at/below cutoff)
Hover: couple's registration number, lead's name, and follow's name

This prototype is restricted to one view type, and one particular competition (Harvard Invitational 2014, held 5-6 April 2014)
source: O2CM: HI14

- tooltip: couple number, lead and follow's names
- round-by-round numeric score, compared to recall cutoff

A dropdown menu (at top of page) allows the selection of any event in the loaded competition (for Harvard Invitational 2014, the competition hardcoded for this prototype, there are several dozen such events), and the graph is redrawn appropriately. All data are being drawn from a single JSON file pre-processed by backend Python scripts.

This view is lacking certain features (detail sidebar, for example), but will likely survive in some form as a way to explore individual competitors' results in individual events, as well as a base for other fine-grained visualizations (*i.e.* ones which present data individual competitors).

B. *Other views*

While the high-density design is often good for diving into a single data set (here, a single dance event), the principle when presenting comparisons is "eliminate irrelevant distractors". Two of the above graph side-by-side would be almost impossible to compare, due to the high-density potentially-interesting visual features which trap the user's attention. In order to effectively comparee even two events (in terms of, say, the Harvard team's relative success), a much simpler view is necessary.

As described in the original project proposal, this view will be a (stacked) area graph. The axes remain approximately the same (x-axis: results, in terms of rounds recalled; y-axis: # of competitors), but the use of area-based streams instead of discrete bars (and the elimination of such fine-grained color scale) draws the attention to the absolute numbers of dancers in certain groups, and their success in being recalled to successive rounds.

Based on paper sketches (which are far too sloppy to show here), the design for the "Teams, single event" will be as follows: A stacked area graph, where each area represents an individual team (by school or other affiliation), and its height at a particular x-value is the number of competitors from that team remaining at that point of the competition. By default, all teams will be light grey, but on hover, each will reveal its name (*e.g.* "Harvard University"), and on click will become 'active', assuming a unique color and rising to the top of the stack (for easy comparison with other teams). On hover over the x-span of a round, details from that round (*e.g.* a listing of individual dancers from selected teams and their numeric scores) will appear as a sidebar. This view aims to answer questions of the form “In this event, how well did Harvard and MIT do, comparatively, and compared to the rest of the field?”

“Teams, multiple events” will be two or more such “Teams” views side-by-side, linked such that teams selected in one graph also become active in the other. This view aims to answer the same questions as “Teams, single event”, but across a few events side-by-side.

The above three views (along with their interactive elements) represent the ‘must-have’ features. The ‘optional features’ are largely additional views, or options to existing views:

“Multiple events, overlaid” will present a single team, stacked with the total field, for a few events, overlaid over each other, as in the above Excel graphs.

For stacked graphs, a toggle-able option between “absolute number” and “relative percentage” switches whether y-axis represents the absolute number of dancers in each group, or its percentage of the total dancers remaining.

For bar-based views, the ability to set up a particular group of dancers the user is interested in (searched by team affiliation or assembled by interactive clicks to add particular individuals), who will be highlighted. This group will *not* be reset as the user changes between events, so that a user can, say, select all of their team’s dancers, and see how they placed within each event at a glance (with all of the detail of the “Competitors” view).

VIII. Evaluation

(in progress, forthcoming)