Using Gap Charts to Visualize the Temporal Evolution of Ranks and Scores

Charles Perin, Jeremy Boy, Frédéric Vernier

HAL Id: hal-01435503
https://inria.hal.science/hal-01435503

Submitted on 14 Jan 2017

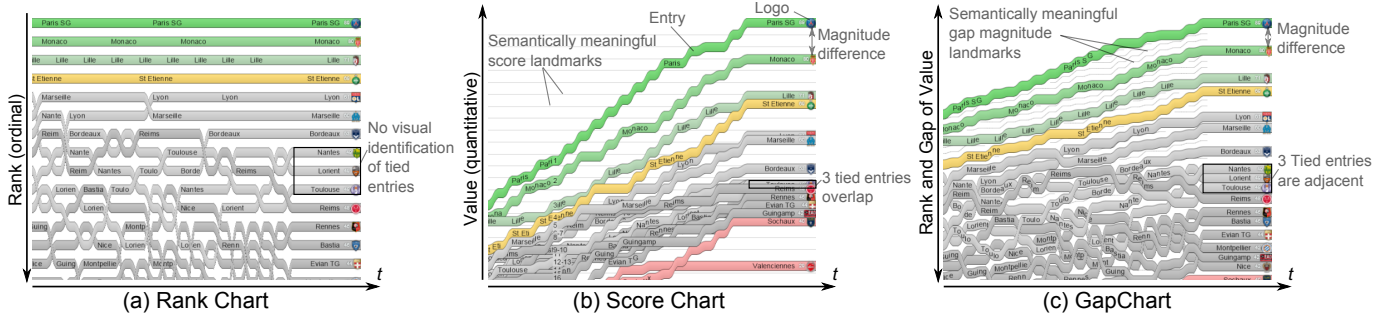# Using Gap Charts to Visualize the Temporal Evolution of Ranks and Scores

Figure 1: The teams participating in a soccer championship visualized using (a) a Rank Chart, (b) a Score Chart, and (c) a Gap Chart.

Charles Perin, University of Calgary. *charles.perin@ucalgary.ca*
Jeremy Boy, United Nations. *myjyby@gmail.com*
Frédéric Vernier, University Paris-Sud. *frederic.vernier@limsi.fr*

Abstract

We present Gap Charts, a novel class of line charts designed for visualizing the evolution of rankings over time, with a particular focus on sports data. Gap Charts show entries, e. g., teams participating in a competition, that are ranked over time according to a performance metric like a growing number of points or a score. The main advantages of Gap Charts are that 1) tied entries never overlap—only changes in rank generate limited overlap between time-steps; and 2) gaps between entries show the magnitude of their score difference. We evaluate the effectiveness of Gap Charts for performing different types of tasks, and find that they outperform standard time-dependent ranking visualizations for tasks that involve identifying and understanding evolutions in both ranks and scores. Finally, we show that Gap Charts are a generic and scalable class of line charts by applying them to a variety of different datasets.

1. Introduction

In this article, we are interested in visualizing the evolution of rankings over time, with a particular focus on sports data. Most athletic competitions rely on the ranking of individuals, teams, or countries to determine outcome. These rankings are usually based on performance metrics like scores, and can evolve over time. Here, we chose to focus primarily on soccer competitions, or *championships*, as they provide a realistic mainstream application case for rank-based visualizations—soccer team rankings are regularly published in newspapers or on the web, and broadcasted on television.

During a championship, team rankings are updated after each match-day, or *time-step*; each team's rank depends on a *score*, which reflects its total number of wins, draws, and losses. If two teams have the same score at a given time-step, their rank is determined by their goal difference, i. e., the difference between goals scored and conceded. Every championship further uses a specific ranking formula to ensure that every rank is occupied by one (and only one) team (e. g., points, then goal difference, then number of goals scored, then game result between the two tied teams, then random). As a result, the ranking is tie-free: while scores may be identical, ranks are always unique.

The most common representation for championship rankings in mass media is a table. Tables can show teams ordered by rank and display scores in cells [1]. However, they cannot convey the magnitude of score differences, or the temporal evolution of both ranks and scores [1, 2]. They also need to be updated or re-created after each time-step.

A simple alternative is to use line charts. Here, we make a distinction between Rank Charts (RC), which use the *y* axis to encode ranks from top to bottom on an ordinal scale, similarly to tables; and Score Charts (SC), which use the *y* axis to encode scores that increase from bottom to top. Generally speaking, RC have the advantage of being overlap-free (Figure 1(a)): teams, or *entries*, are visually distinct at each time-step—although they may intersect between time-steps as rankings change. This makes ranks easy to distinguish. However, RC do not convey the magnitude of scores, nor do they show *ties* (equal scores). This can be problematic when trying to predict future rankings, and ultimately a championship's outcome. Conversely, SC convey the magnitude of scores. This can help make predictions. However, SC are not overlap-free when scores are tied, which makes it impossible to determine unique ranks (Figure 1(b)).

To address these respective limitations, we introduce Gap Charts (GC), a novel class of line charts which use the *y* axis to encode both ranks from top to bottom, and scores from bottom to top. The main originality of GC is that they use the gaps between lines, i. e. the white space, to encode score magnitude. This makes GC overlap-free (Figure 1(c)), as tied scores are simply shown by adjacent entries, i. e., lines that are not separated by a gap at a given time-step.

We evaluate the effectiveness of GC for performing rank-, score-, and rank-and-score related tasks, by comparison with RC and SC. We chose to focus primarily on static renderings of GC, i. e., without possible additional interactive features, as static charts are most common in mass-media publications on soccer championships (and on athletic competitions in general). Our results show that GC are most effective for rank-and-score related tasks, and that they are a good tradeoff between RC and SC for rank-alone and score-alone related tasks.

Finally, we extend our initial design of GC by exploring possible interactive techniques that can alleviate our class of line charts for tasks for which they are least effective. We also assess the scalability of GC by applying them to a range of other datasets. In summary, our main contributions are:

1. a formal distinction between RC and SC, two existing classes of line charts;
2. the design and implementation of GC, a novel class of line charts;
3. a comparative evaluation of GC, RC, and SC using real data from the French soccer *Ligue 1*; and
4. the extension and generalization of GC using other/larger datasets.

2. Background

Sports enthusiasts are generally accustomed to seeing sports rankings in tables. Several studies have shown that tables are effective for simple tasks like value retrieval (see [2] for a review). However, other previous work has shown that people interested in soccer championships usually seek to perform complex synoptic tasks like analyzing trends and comparing patterns over time [1]; and line charts have been found more efficient than tables for performing such tasks [1, 2].

2.1. Formalizing the distinction between Rank Charts and Score Charts

Slope graphs are a particular type of line chart that *"compare changes over time for a list of nouns located on an ordinal or interval scale"* (Figure 2(a)) [3]. They map the values of *n* entries on *m* axes (or *time-steps*—*m* = 4 in Figure 2(a)), and draw a connection between each entry's values. Values are displayed in *boxes*, i. e., horizontal white spaces or segments, at each time-step. Connections show magnitudes of score differences. As entries are plotted on an ordinal scale, values encoded along the *y* axis can be either ranks or scores [1][2][3][4][5]. We define Rank Charts (RC) as a class of slope graphs that uses the *y* axis to encode ranks, and Score Charts (SC) as a class of slope graphs that uses the *y* axis to encode scores.

---

[1] http://www.citylab.com/design/2014/06/a-brilliantly-restored-19th-century-visualization-of-us-city-population-shifts/373386/
[2] http://junkcharts.typepad.com/junk_charts/2010/12/be-guided-by-the-questions.html
[3] http://charliepark.org/slopegraphs/
[4] http://charliepark.org/a-slopegraph-update/
[5] http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0003nk&topic_id=1
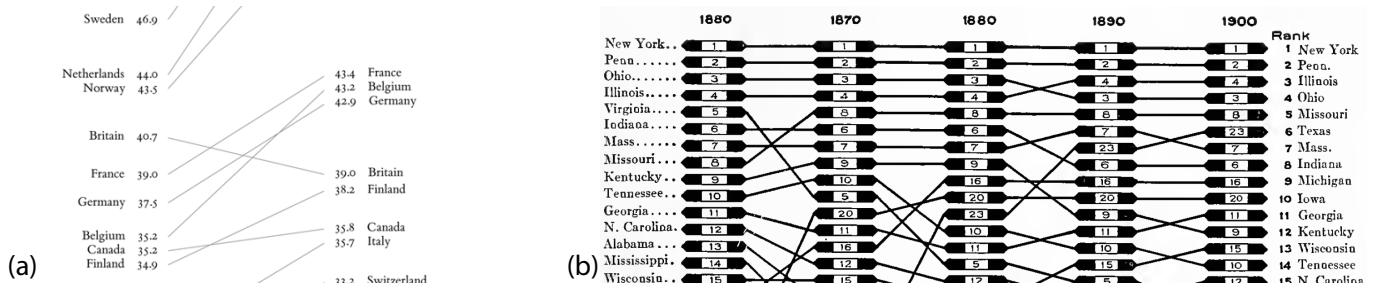
Figure 2: (a) Tufte's Slope Graph [3] (used with permission, cropped) and (b) Brinton's Rank Chart [4] (cropped).

RC (Figure 1(a)) show each of the *n* entries at a unique rank at every time-step *m*. The main advantages of RC are: 1) they are overlap-free—assuming two entries cannot have the same rank; and 2) they scale well, both for *n* and *m*. Figure 2(b) shows an early use of RC from the beginning of the 19th century [4]. RC are also extensively used on the web [6], and come with many variations. While RC clearly show changes in rank, they do not show the magnitudes of score differences, nor do they show tied entries—an important feature according to [5]. That said, some designs have attempted to address the latter limitation by grouping entries at a same rank on the ordinal axis [7]. However, this breaks the expected bijection mapping of entries to ranks.

SC (Figure 1(b)) show the scores for each of the *n* entries at every time-step *m*. The main advantage of SC is that ranks can be inferred from scores, as long as these are distinct (no ties). If they are not, entries overlap, and ranks are impossible to determine. SC also waste a great amount of white-space—particularly in the top-left corner when all scores are low, if scores start from 0 and increase monotonously—, and suffer from scaling: the higher *n*, the more overlap.

Overall, we consider RC useful for showing ranks, and SC useful for showing scores and the magnitude of their differences.

2.2. Other Ways of Visualizing Rankings

RankExplorer [6], which uses stacked area charts, and LineUp [5], which enables interactive analysis of multidimensional ranked entries, were developed to visualize time-dependent rankings. Although they are powerful tools for performing advanced queries, their visual and interaction complexity is high, and we believe untrained people may find it difficult to interpret and interact with them. Sports enthusiasts cannot be expected to be visualization experts [7]; they may lack [1] visualization literacy [8] and interaction propensity [9]. Hence we focus primarily on static representations in this article.

Another recent work [10] has explored the combination of tables and line charts for manipulating ranking tables. However, as for the aforementioned tools, the benefits of this technique lie in relatively complex interaction.

3. GapChart

We introduce Gap Charts (GC) as a novel class of slope graphs which combines the advantages of both RC and SC. GC simultaneously show the temporal evolution of both ranks *and* scores, as they are derived from a specific state in the continuous transition between an 'ordinal rank' *y* axis (RC) and a 'continuous score' *y* axis (SC). Entries are represented by *lines*, composed of *boxes* and *links*. Boxes show the value (rank and/or score) of the entry at a given time-step, and links connect consecutive boxes. *Labels* designate entries' names.

By exploring the continuum between RC and SC, and by tuning each component, we identified the following defining characteristics for GC:

C1 the *y* axis encodes both ranks and score, and the score magnitudes are shown in the gaps between lines;

C2 tied entries are made visually distinct by keeping equal scores at a given time-step adjacent, i. e., with no gap between lines, which keeps the chart overlap-free;

C3 the result is less compact than RC, but more than SC.

Figure 3 shows a gap chart of the Spanish *Liga* 2013–2014. The chart illustrates how three teams are clearly better than the others, most of which simply struggle to avoid being demoted. There is a clear gap between the top-three teams that qualify for the *Champions League* and the rest. This gap appears early in the championship, and continuously increases until the end. There is some competition among the top-three, as they change ranks several times during the second half of the championship. They also finish with very small gaps between them—in fact, FC Barcelona and Real Madrid are tied, i. e., adjacent, at the end of the championship. Their final rank was determined by their goal difference. A second group of teams, from 4th to 7th, fights for 4th place, and Athletic Bilbao is seen to resist until the end. Betis Sevilla however seems destined to be demoted early on, while the two other teams of the bottom trio remain very uncertain throughout the championship.
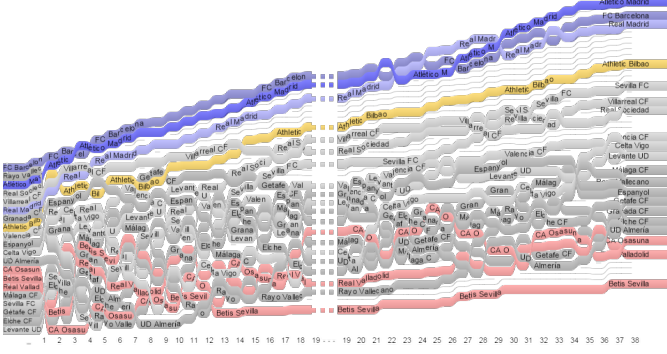
Figure 3: The Spanish *Liga* 2013–2014, visualized using GC.

Figure 4: Main rendering parameters.

3.1. Layout Algorithm

We created a simple algorithm for describing RC, SC, and GC in a unified way. Our generic layout function takes a factor $F \in [0,1]$ as a parameter to compute the vertical position of boxes over time. F expresses the continuum between RC and SC: $F = 0$ creates a RC, and $F = 1$ creates a SC. Given:

H	the height of the canvas
E	the set of entries
T	the set of time-steps
RANK(e_n, t_m)	the rank of e_n at t_m
SCORE(e_n, t_m)	the score of e_n at t_m
$m = Min(\text{SCORE}(E_i, T_j))$	$i \in |E|, t \in |T|$
$M = Max(\text{SCORE}(E_i, T_j))$	$i \in |E|, t \in |T|$
h	the height of entries

The vertical position y of an entry $E_n, n \in |E|$ at each time-step $t_m, m \in |T|$ is computed as follows:

$$y_R = H \frac{\text{RANK}(e,t)}{|E|}$$
$$y_S = H(1 - \frac{1}{|E|h})(1 - \frac{\text{SCORE}(e,t) - m}{M - m})$$
$$y = (1 - F)y_R + F y_S$$

For a soccer championship, $m = 0$ and M is the score of the team ranked 1st at the last time-step. y_R is the y position according to ranks; y_S is the y position according to scores. Setting $F = 0$ implies $y = y_R$ and produces RC; setting $F = 1$ implies $y = y_S$ and produces SC; and setting $F = 1 - \frac{1}{n}$ and $h = \frac{H}{n|E|}$ ensures **C1**, **C2**, and **C3** and produces GC.

3.2. Rendering Rationale

RC, SC and GC share several adjustable rendering parameters. These are: box *width* and *height*, link *curvature* (from a straight line to a "S" shape), link *height at extremities* and *at inflection point* (from very thin to same as box height), *shading*, and *labels positioning*. Each parameter is described in Figure 4.

According to Tufte, data graphics should maximize their data-ink ratio [3]. This suggests that any class of slope graph should avoid large l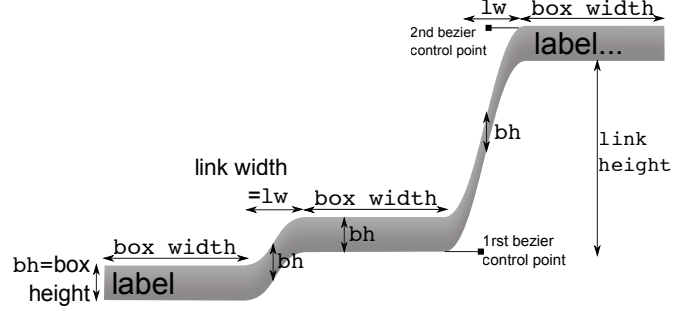ink heights and colors—as *a priori* these do not encode information. Figure 2(a) respects this principle: links are rendered as straight lines (no curvature) with a continuously thin height at extremities and inflection point. There is no shading, and the labels (numeric values) are shown in every box.

Our choice of rendering for GC (see Figure 3) clearly violates the data-ink ratio principle. However, it is based on a number of application-domain-driven rationales, as well as on a short qualitative evaluation, which we describe below.

An important specificity of soccer championship rankings is that scores never decrease—they only increase at a steady, pre-defined pace. Teams gain 3 points for winning a match, 1 for drawing, and 0 for losing. As such, focusing on a unique team's score variations, i. e., on the slope of connections, is of little interest. However, being able to follow the evolution of a team as it changes ranks is crucial. This is why we display entries as continuous lines, where links and boxes smoothly alternate.

In addition, we follow standard color-coding for soccer championships: we use blue for the three top-ranked teams at the end of the championship—as they qualify for the European *Champion's League*—, yellow for the fourth team—as it qualifies for the *Europa League*—, and red for the three bottom-ranking teams—as they are demoted to the minor league.

To further tune the different rendering parameters described in Figure 4, we asked six volunteers to order 3 classes of charts (RC, SC, and GC) ∗ 5 rendering parameters = 15 series of three parameter variations according to aesthetic preference and legibility. We wanted to ensure that the charts would be appealing to unfamiliar viewers (e. g., sports enthusiasts). Preference orders were given on 1–3 Likert-like scales, with the possibility of giving equal preferences. Variations were:

- *link shape*: straight line, weak curvature, strong curvature;
- *link height at extremities*: very thin, medium, same height as boxes;
- *link height at inflection point*: very thin, medium, continuous (i. e. same height as boxes);
- *shading:* none, low, strong; and
- *labels*: origin/middle/end of chart, full name where boxes are wide enough, repeated initials in every box.

For legibility, we instructed participants to focus on how easy/difficult each rendering variation made it to identify:

- a lumped group of teams;
- the gap between two or more teams;
- the evolution of a team's rank over time; and
- the stability of a team's score over time [8].

We first inspected the legibility scores. We then confronted the most legible variations with their aesthetic preference scores. Because both orders were similar for all three classes, i.e., for RC, SC, and GC, we decided to keep one rendering configuration: weak curvature, stroke widths at extremities the same size as the boxes, medium stroke widths at link inflection points, low level of shading, and full name labels where boxes are wide enough. Generally speaking, we found that thick lines were preferred over thin ones. Thus, we stress that although the data-ink ratio principle suggests the use of thin lines, thick lines and repeated labels were perceived as more effective.

Some of these design takeaways may seem counter-intuitive. In particular, existing charts usually emphasize the use of thin lines. However, thick lines and repeated labels make it easier to follow entries over time, especially when the number of entries is high. In addition, for GC, line thickness has to be a ratio of the chart height to ensure that entries with tied scores are adjacent— one of the main features of GC. While the line thickness / chart height could be reduced, it would be at the cost of legibility. Finally, comparing ranks is the main focus of our paper, and existing representations of ranked entries often use similarly thick lines [9] [10].

4. Evaluation

We evaluated the effectiveness of GC for performing rank-, score-, and rank-and-score related tasks, by comparing them with RC and SC. We used real soccer data form 33 seasons of the French *Ligue 1*, a national championship in which 20 teams (*entries*) confront each other twice, over a period of 38 match-days (*time-steps*).

Comparing GC with RC and SC may not seem ideal, since RC are specifically designed to show only ranks, SC to show only scores, and GC to show both. However, the lack of existing alternative designs to GC for showing both ranks and scores simultaneously prevents us from establishing a proper baseline condition. By default, it is impossible to perform score-related tasks using RC, and it can be very difficult to perform rank-related tasks using SC. To alleviate these limitations, we designed a set of 'hover' inspectors. While our main focus is on static representations—which cannot include such 'lightweight' interactions—we wanted to make sure score-related tasks would

be possible to perform with RC, and that rank-related tasks could be performed without too much effort with SC.

We then selected four tasks based on different visual properties of each class of charts, which we found to be frequent and/or important in related work. We also made sure these would be meaningful for sports rankings. Tasks were:

T1 determine the longest period during which teams T_i and T_j have the same score;

T2 determine when the score difference between the teams ranked r_i and r_{i+1} was the highest;

T3 determine how many changes occur at rank r_i; and

T4 determine the longest period during which the score of team T_i stays the same.

Based on Andrienko et al.'s task taxonomy for time-dependent data [11], Perin et al. [1] have found that comparing teams (T1, T2) and detecting trends (T3, T4) are important tasks in the analysis of a soccer championship. Similarly, Gratzl et al. [5] have found that comparing entries' scores and slopes over time (T1, T2, T4), as well as retrieving specific ranks and tied ranks (T3) are frequent tasks related to ranked entries.

4.1. Inspector Designs

As previously discussed, both RC and SC have several limitations. Score-related tasks are impossible to perform with RC, as scores are not indicated. Likewise, rank-related tasks are difficult to perform with SC, as entries with identical scores overlap. Thus, although our primary goal is to assess the efficiency of each technique for static representations, we designed a set of inspectors to enhance RC and SC for our experiment. The rationale was that: 1) the experiment would be extremely frustrating for participants if one third of the tasks were impossible to perform; and 2) GC would be optimal for static representations if the technique outperformed the enhanced/interactive versions of RC and SC. By adding inspectors to each class, we expected to maximize accuracy at the expense of increasing the amount of time spent to perform tasks.

We first designed a basic inspector for all classes of charts— the *tooltip inspector*—to display an entry's full name, rank, score, and goal difference on hover. This inspector also shows the time-step corresponding to the mouse cursor's *x* coordinate. As RC do not show scores, we then designed a specific *scores inspector* to display the scores of all entries at a time-step. Similarly, as SC do not show ranks, we also designed a specific *ranks inspector* to displays the ranks of all teams at a time-step. The scores and ranks inspectors enable quick comparison of the values of several entries at a time. Meanwhile, the tooltip inspector only allows for inspection of one entry at a time. Figure 5 shows the different inspectors.

---

[8] These features are individually inspired by the tasks we later used in our evaluation of GC, described in Section 4.

[9] http://www.javiertordable.com/interesting-visualizations-changes-over-time/

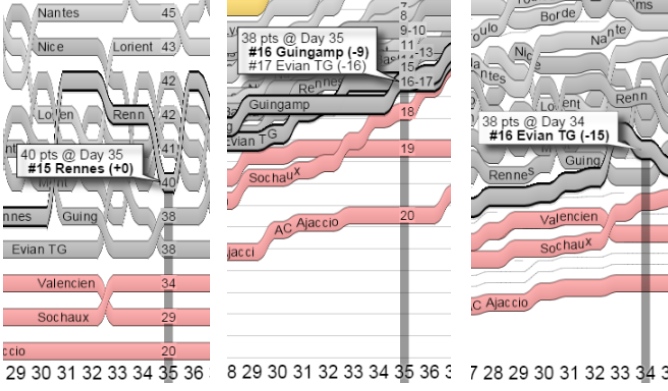[10] http://in.somniac.me/2010/01/fortune-500-visualization/

Figure 5: The different inspectors. For RC (left), both the tooltip inspector and the scores inspector. For SC (center), both the tooltip inspector and the ranks inspector. For all, the tooltip inspector.

4.2. Hypotheses

For each task $\mathbf{T}_i$, we formulated two hypotheses $\mathbf{H}_i\mathbf{a/b}$. Here, we refer to each task in the form of a question (which we asked participants), and propose the following coding for hypotheses: $A > B$ means that we hypothesize participants will be more accurate using A than B, and will respond more quickly using A than B if the accuracy rate is similar. All tasks involved analyzing one season of the championship, i. e., 20 entries over 38 time-steps. T1 and T2 required focusing both on ranks and scores; T3 only on ranks; and T4 only on scores.

T1 *What is the longest period during which teams T_i and T_j had the same score?*

H1a GC > RC

H1b GC > SC

Rationale: using GC, T1 simply consists in finding the longest segment during which the two entries have no gap between them (adjacent entries); using RC, T1 requires the scores inspector, as RC do not convey scores; and using SC, T1 requires the tooltip inspector, as entries with the same score overlap.

T2 *When is the score difference between the team ranked r_i and r_{i+1} the highest?*

H2a GC > RC

H2b GC = SC

Rationale: using GC, T2 simply consists in finding the biggest gap between the two entries; using RC, T2 requires using the scores inspector; and using SC, T2 should require the same strategy as using GC.

T3 *How many changes occurred at rank r_i?*

H3a GC < RC

H3b GC > SC

Rationale: using GC, T3 consists in following the line ending at the given rank (starting from the end), and counting and

moving up one line every time there is a crossing; using RC, T3 simply consists in horizontally following the rank, and counting the number of times it is crossed; and using SC, T3 requires the ranks inspector.

T4 *What is the longest period during which the score of team T_i stayed the same?*

H4a GC > RC

H4b GC < SC

Rationale: using GC, T4 requires the tooltip inspector, as entries with invariant scores may change ranks; using RC, T4 requires the scores inspector; and using SC, T4 simply consists in finding the entry's longest horizontal segment.

4.3. Procedure

We recruited 12 unpaid participants (1 female), aged 19–39 (mean=28), who were not involved in the pre-study. All were students or university staff, and had at least some basic knowledge of soccer championships and of line charts. The experiment was conducted on a desktop computer equipped with a mouse, a keyboard, and a 30" LCD display with a resolution of 2560x1600 pixels—charts were 1272 pixels wide and 750 pixels high. Participants first filled out a short background survey, and answered general questions about soccer. They were given the correct answer for each question, to ensure full comprehension of the ranking process (e. g., number of teams in the championship, number of points gained for winning, drawing, or losing a match, etc.). They were then given a sheet of instructions.

The experiment was blocked by class of charts (RC, SC, GC). Each block consisted of four trials—one for each task (T1, T2, T3, T4)—repeated five times. Blocks and trials were counterbalanced. Dependent variables were *accuracy* (number of correct answers) and *efficiency* (time spent answering). The full factorial design was as follows:

	12	*participants*
×	3	*classes* (RC, SC, GC)
×	4	*tasks* $(T1, T2, T3, T4)$
×	5	*repetitions*
	720	*trials*

Before each block, participants were prompted with the new class, and were invited to ask any questions they might have about the visual representation. Before each trial, a toy example explained the task. All tasks were restricted to the second half of a championship, i. e., to the right half of the charts (see Figure 3), as ranks vary too much during the first half—because all teams start with 0 points. This ensured a consistent difficulty level across datasets and comparable trials. Repetitions were limited to 30 seconds, and participants were allowed to "skip" a repetition if it was considered too difficult. The use of inspectors was allowed, and responses were submitted using a text field. The correct response was displayed after each repetition. Finally, after each trial, participants were asked to evaluate on
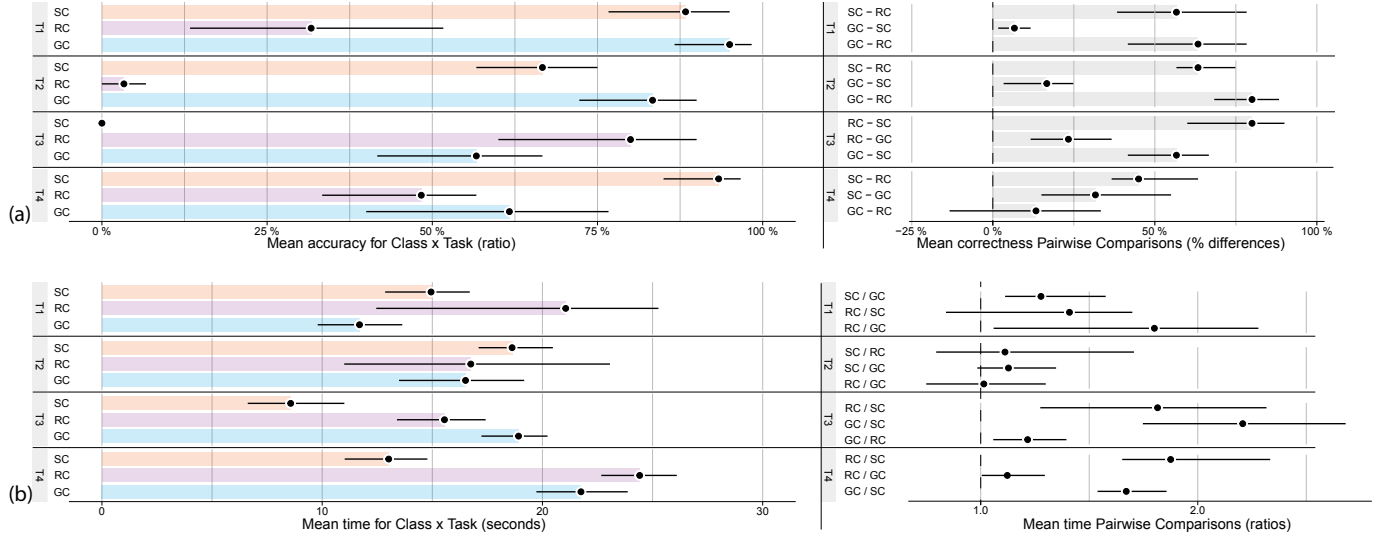
6

Figure 6: Mean (a) *accuracy*, (b) *efficiency*, and mean (a) *accuracy* difference, (b) mean *efficiency* difference as a function of visualization technique and task. Error bars are 95% bootstrapped confidence intervals.

1–5 Likert scales the difficulty of the task, their confidence in their responses, and the suitability of the class of charts for the task. Overall the experiment lasted approximately 45 minutes.

4.4. Results

In order to assess the accuracy and time of performing tasks with the three classes of charts, we base our analyses on *estimation*, i.e., effect sizes with bootstrapped [12] confidence intervals [13] (see Figure 6). Using effect sizes with confidence intervals is recommended by the APA [14], and for reporting statistical results in HCI [15] over the traditional null hypothesis significance testing.

We first compared accuracy rates between classes of charts. If we found no clear difference, we removed timed out and skipped trials and inspected efficiency. Note that in the experiment design, we had highlighted entries that participants were required to focus on in T1 and T4 to avoid measuring any extra time spent finding those entries.

For T1, there is strong evidence that GC are roughly twice as accurate as RC (**H1a**); and while there is little evidence that GC are more accurate that SC, there is evidence that they are more efficient (**H1b**), i. e., the task was completed on average 1.2 times faster using GC.

For T2, there is strong evidence that GC are roughly 75% more accurate than RC (**H2a**); and there is evidence that GC are more accurate than SC (**H2b**), although the effect size is less important.

For T3, there is evidence that GC are less accurate and efficient than RC (**H3a**); but there is strong evidence that GC are more accurate than SC (**H3b**).

Finally for T4, there is no evidence that GC are more accurate than RC, and only little evidence that they are more efficient (**H4a**); and there is strong evidence that GC are less accurate

and efficient than SC (**H4b**).

Figure 7 shows the questionnaire results formatted using Bertifier [16]. These are consistent with the accuracy and efficiency analyses. GC were considered well-suited for T1 and T2, which were easy to perform. Responses were more mitigated for T3 and T4, although there is no strong trend against GC. RC were considered ill-suited T1 and T2, well-suited for T3, and responses were mitigated for T4. SC received enthusiastic feedback for T1 and T4, more mitigated responses for T2, and were considered ill-suited for T3.

4.5. Discussion

H1a is clearly supported by our results. While **H1b** is also supported, we had expected to see a bigger difference between GC and SC. We attribute the little evidence of a difference in accuracy to the fact that we had initially highlighted entries for comparison. In a less artificial setting with no highlights, SC would suffer from visual clutter, and the potentially overlapping entries of interest would be harder to find and follow. This would not be the case with GC, as they are overlap-free.

H2a is also clearly supported by our results. **H2b** however is not, as we had expected results to be similar for GC and SC. We attribute the evidence that GC are more accurate than SC to the fact that GC show *score magnitude landmarks* (the lightgray horizontal lines in Figure 1(c)). These convey the relative score differences between consecutively ranked entries. In contrast, SC show *absolute landmarks* (the lightgray horizontal lines in Figure 1(b)). These only convey the absolute scores of entries, and not the differences between them.

H3a is supported. We expected RC to outperform GC because they use stable ordinal vertical positions, which result in generally horizontal lines—except between time-steps when rankings change. Using RC, T3 simply consisted in following a
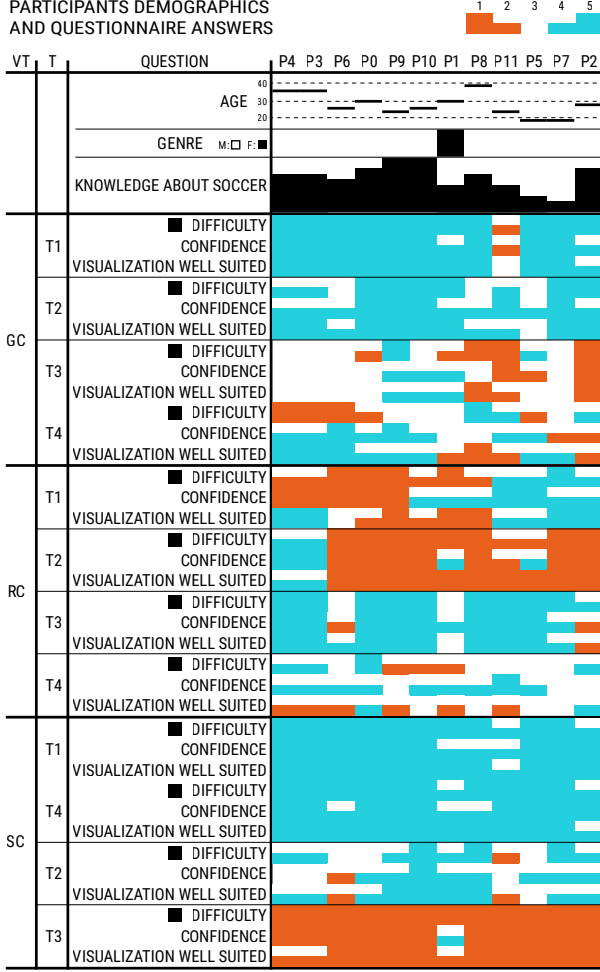
Figure 7: Answers on 1–5 likert scales are mapped on a dual-color scale, 3 being the neutral value. Difficulty scores are inverted to be congruent with other questions, as indicate the black squares. Knowledge about soccer is the average score of 7 questions.

period may be challenged by another team B that obtains the same score after winning a match. If team B's goal difference is higher than team A's, then team A will be reclassified at a rank below team B. Visually, this would result in a break in the horizontal line representing the team, even though the team's score remains unchanged.

Overall, our results show that RC are best suited for rank-related tasks, SC for score-related tasks, and GC for rank-and-score related tasks. This makes sense, as each class of charts is designed to facilitate its respective type of tasks. However, we have found that GC are a good tradeoff for performing all types of tasks—especially if interaction is disabled. Although scores cannot be precisely determined using static GC, we argue that score-values are generally not a first class characteristic of sports rankings—ranks come first, gaps between consecutive entries come second, and scores come last (usually to discriminate ties). Finally, it is interesting to point out that while the scores inspector worked very well for T4 using RC, the ranks inspector was ineffective for T3 using SC (see Figure 6).

Based on our results, we have established the following list of recommendations:

- GC should be used for rank-and-score related tasks. RC should not.
- RC are most effective for rank-related tasks, but GC provide a good alternative. SC should not be used.
- SC are most effective for score-related tasks, but GC provide an alternative. RC may be used if a scores inspector is implemented.

On a higher level, we recommend using GC over the other classes for static representations, as GC are more generic. However, interactively transitioning between RC, GC, and SC using our layout algorithm should prove optimal in situations where interaction is possible.

5. Extensions and Future Work

Participants in our experiment—especially the soccer enthusiasts—suggested developing a way to visualize championships with a focus on a particular team. To address this, we extended our initial design of GC to include an advanced *entry focus* interaction (Figure 8).

We also used GC to visualize a range of other datasets, including data from other sports, academia, economics, and politics. All examples are available at http://newcol.free.fr/. This variety allowed us to assess the scalability of static renderings of GC, as well as the importance of finding ways to visualize missing data.

5.1. Entry Focus Interaction

For interactive versions of GC, double clicking on an entry *e* sets it as the *focus entry*. The chart animates to a 'focus-centered' state, where the baseline for the *y* scale transitions from the minimal value in the dataset to the origin of *e*. The

straight horizontal line along the provided rank, and counting the number of times it was crossed. Vertical positions are not as stable in GC, as they also continuously increase to show scores. This is even more true for SC, which is why **H3b** is so clearly supported by our results.

H4a is only partially supported, and the effect is small. We attribute the lack of evidence of a difference in accuracy between GC and RC to the efficacy of the scores inspector. It enabled score extraction in RC, which would otherwise have been impossible—unlike in GC. **H4b** however is clearly supported. We expected SC to outperform GC because using the former class, T4 simply consisted in finding the longest horizontal segment for the provided entry. The task was not as straightforward using GC, as entries with invariant scores may change vertical position according to changes in rank. For example, a soccer team A that has a stable score over a certain
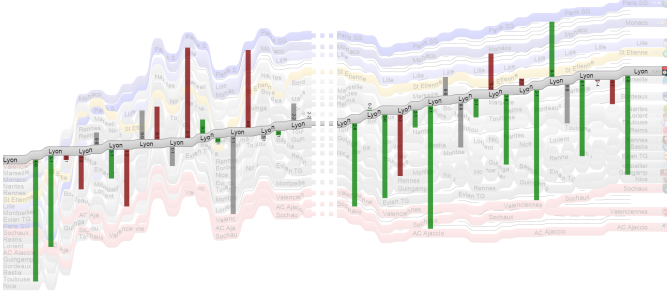
Figure 8: Illustration of the entry focus interaction. The entry is set as the visualization's baseline, and the causes of change are shown.

line showing *e* is thus represented as it would be in a slope graph, with the vertical position of boxes showing *e*'s score at each time-step. This facilitates tasks like T4, as they can be performed like using SC. All other entries are laid out according to the baseline, and the vertical distance between *e* and any other team encodes the score difference between them.

As several participants in our experiment later asked to visualize the causes of change, we also added visual cues (green and red vertical bars) for each game played by *e*. Each bar links *e* with its opponent at the corresponding time-step. Green bars encode wins, gray bars draws, and red bars losses. Figure 8 shows this information. We see that *e* won most of its games against teams ranked lower, and lost to most teams ranked higher. *e* also had difficulties winning several games in a row.

5.2. Generalization and Scalability

As GC are a generic class of slope graphs, we were able to apply them to various other time-dependent ranking datasets. Here, we discuss the generalization and scalability of static renderings of GC through two examples.

Figure 9 shows the evolution of 198 cyclists' (*entries*) rankings after each of the 21 stages (*time-steps*) of the *Tour de France*. The magnitude landmarks (thin gray lines) represent 1 minute gaps between cyclists. Colors encode cyclists' nationality, and stage miniatures provide context at the top. The chart clearly shows that many changes occurred in the rankings during the second stage of the race, which means this stage was key. After that, rankings remained stable for three stages, before changing dramatically once again. *Tour de France* enthusiasts will also see that 'flat terrain stages' generally do not impact rankings or gap magnitudes, whereas 'mountain stages' strongly impact both.

Figure 10 shows the evolution of the top 100 universities' (*entries*) rankings over 10 years (*time-steps*), according to the ARWU Shanghai University Ranking. Colors encode world regions (North America in purple, Europe in orange, Asia in green, and unclassified in dark gray). The most immediate observation is that the entry ranked first (Harvard University) is far above all others. Below that, several universities struggle

Figure 9: *Le Tour de France at a Glance* [17].

for the top-5 ranks, while competition is fierce at the bottom of the ranking. Nearly half the top-100 universities are North American; only five European and two Asian universities are among the first third. It is also interesting to point out that rankings varied a lot between 2003 and 2005. This was simply due to modifications of the ranking formula.

A specific issue raised by this latter example is that data are sometimes incomplete for certain entries (e. g., the universities shown in dark gray). This is typically the case when entries enter and/or leave the ranking at a given time-step. We can think of several 'off the bat' strategies to visualize the presence of miss-
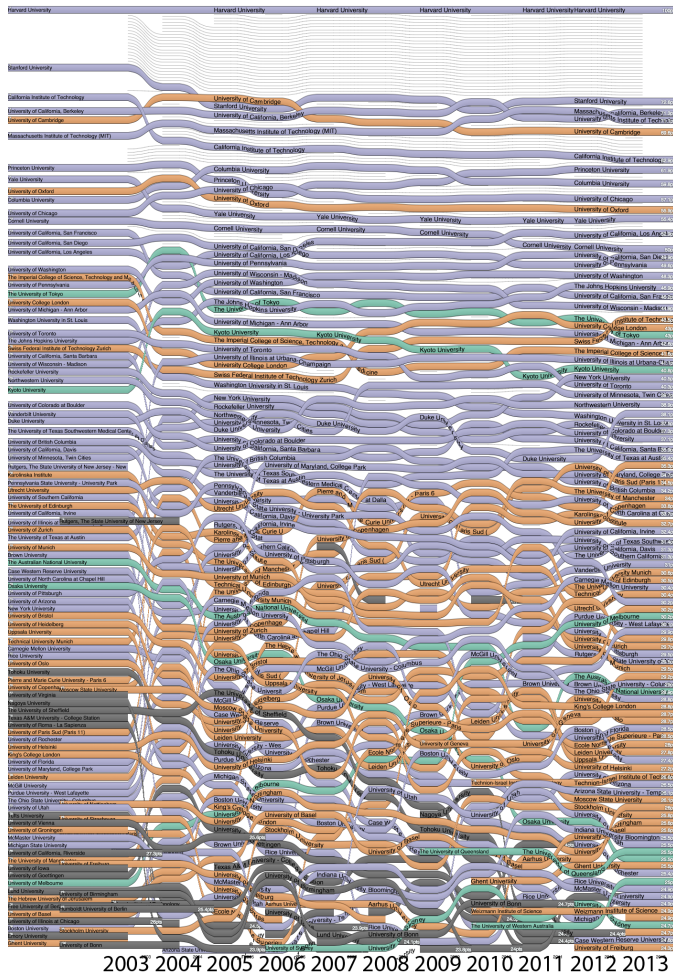
Figure 10: The Shanghai University Ranking.

and scores, ensure no overlap at each time-step, and make it possible to identify tied entries.

We have evaluated the effectiveness of GC for performing rank-, score-, and rank-and-score related tasks. Our results show that GC are most effective for latter type of tasks, and that they are a good tradeoff between RC and SC for performing rank-alone and score-alone related tasks.

Using GC to visualize a wide range of different datasets has raised a set of new challenges and opportunities. Considering interaction, we have explored the possibility of focusing on a particular entry. This should alleviate GC for score-related tasks, which by default are most difficult.

Overall, we believe this work provides evidence that semi space-filling visualizations have unique properties, which raises the question as to whether they can be applied to other families of data graphics.

References

[1] Charles Perin, Romain Vuillemot, and Jean-Daniel Fekete. A table!: Improving temporal navigation in soccer ranking tables. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 887–896, New York, NY, USA, 2014. ACM.

[2] Sirkka L. Jarvenpaa and Gary W. Dickson. Graphics and managerial decision making: Research-based guidelines. *Commun. ACM*, 31(6):764–774, 1988.

[3] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1983, 2001. ISBN 0-9613921-0-X.

[4] William C. Brinton. *Graphic Methods for Presenting Facts*. Industrial management Library. Engineering Magazine Company, 1914.

[5] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19 (12):2277–2286, Dec 2013.

[6] Conglei Shi, Weiwei Cui, Shixia Liu, Panpan Xu, Wei Chen, and Huamin Qu. Rankexplorer: Visualization of ranking changes in large time series data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12): 2669–2678, December 2012.

[7] Charles Perin, Romain Vuillemot, and Jean-Daniel Fekete. Soccerstories: A kick-off for visual soccer analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19 (12):2506–2515, December 2013.

[8] J. Boy, R. A. Rensink, E. Bertini, and J. D. Fekete. A principled way of assessing visualization literacy. *IEEE Transactions on Visualization and Computer Graphics*, 20 (12):1963–1972, Dec 2014.

[9] J. Boy, L. Eveillard, F. Detienne, and J. D. Fekete. Suggested interactivity: Seeking perceived affordances for

ing data, including changing the color (as we have done here), using dashed-lines, or applying sketchy renderings. However, we stress more work is needed to determine best practices.

Overall, these two examples show that although individual entries may be hard to discern and follow when their number increases, interesting and important trends can still be detected. Ultimately, the most limiting factor in the scaling of static GC is the size of the screen or or of the paper on which they are displayed/printed.

6. Conclusions

In this article, we have proposed a formal distinction between Rank Charts (RC) and Score Charts (SC), two previously existing classes of slopegraphs for visualizing time-dependent rankings. Rank Charts are effective for visualizing ranks, and Score Charts for visualizing scores. However, both techniques are limited when it comes to visualizing the other dimension.

We have introduced Gap Charts (GC) as a novel class of slopegraphs to address these limitations. GC show both ranks

information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):639–648, Jan 2016.

[10] Romain Vuillemot and Charles Perin. Investigating the direct manipulation of ranking tables for time navigation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2703–2706, New York, NY, USA, 2015. ACM.

[11] Natalia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., 2005.

[12] K. N. Kirby and D. Gerlanc. BootES: An r package for bootstrap confidence intervals on effect sizes. *Behavior research methods*, 45(4):905–927, 2013.

[13] Geoff Cumming and Sue Finch. Inference by eye: Confidence intervals and how to read pictures of data. *American Psychologist*, 60(2):170, 2005.

[14] American Psychological Association. *The Publication manual of the American psychological association*. Washington, DC, 6th edition, 2013.

[15] Pierre Dragicevic. Fair statistical communication in hci. In *Modern Statistical Methods for HCI*, pages 291–330. Springer, 2016.

[16] C. Perin, P. Dragicevic, and J. D. Fekete. Revisiting bertin matrices: New interactions for crafting tabular visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2082–2091, Dec 2014.

[17] Charles Perin, Jeremy Boy, and Frédéric Vernier. Le tour de france at a glance. In *IEEE VGTC/VPG International Data Visualization Contest*, 2014. `http://vgtc.org/content/ieee-vgtc-vpg-international-data-visualization-contest`.