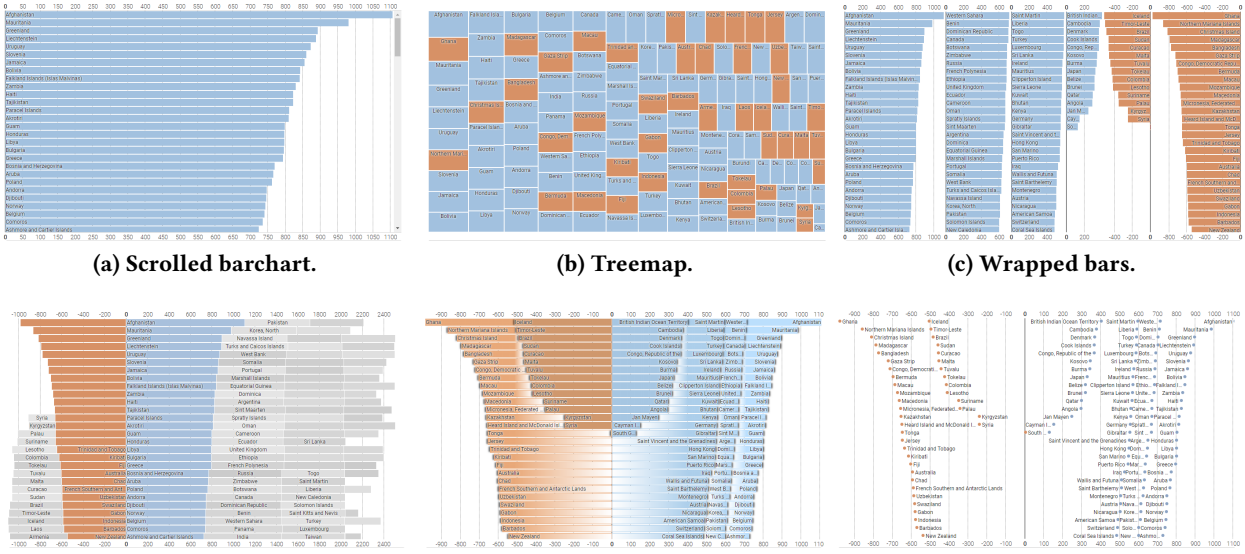
Ranked-List Visualization: A Graphical Perception Study

Pranathi Mylavarapu
Information Studies
University of Maryland
College Park, MD, USA
pranathi@umd.edu

Adil Yalçin
Keshif LLC
Alexandria, VA, USA
adil.yalcin@gmail.com

Xan Gregg
SAS Institute, Inc.
Cary, NC, USA
xan.gregg@jmp.com

Niklas Elmqvist
Information Studies
University of Maryland
College Park, MD, USA
elm@umd.edu



(a) Scrolled barchart. **(b) Treemap.** **(c) Wrapped bars.**

(d) Packed bars. **(e) Piled bars.** **(f) Zvinca plot.**

Figure 1: Six ranked-list visualizations showing the same dataset of 150 values. Blue values are positive, whereas negative values are red. In this paper, we begin to quantify the strengths and weaknesses of each variation with a crowdsourced visual perception study using unlabeled versions of these charts (with no negative values).

ABSTRACT

Visualization of ranked lists is a common occurrence, but many in-the-wild solutions fly in the face of vision science and visualization wisdom. For example, treemaps and bubble charts are commonly used for this purpose, despite the fact that the data is not hierarchical and that length is easier to perceive than area. Furthermore, several new visual representations have recently been suggested in this area, including wrapped bars, packed bars, piled bars, and Zvinca plots. To quantify the differences and trade-offs for these ranked-list visualizations, we here report on a crowdsourced graphical perception study involving six such visual representations, including the ubiquitous scrolled barchart, in three tasks: ranking (assessing a single item), comparison (two items), and average (assessing global distribution). Results show that wrapped bars may be the best choice for visualizing ranked lists, and that treemaps are surprisingly accurate despite the use of area rather than length to represent value.

CHI 2019, May 4–9, 2019, Glasgow, Scotland, UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300422

CCS CONCEPTS

• **Human-centered computing** → **Information visualization**; **Empirical studies in visualization**; **Visualization design and evaluation methods**.

KEYWORDS

Data visualization, ranked lists, graphical perception.

ACM Reference Format:

Pranathi Mylavarapu, Adil Yalçin, Xan Gregg, and Niklas Elmqvist. 2019. Ranked-List Visualization: A Graphical Perception Study. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2019), May 4–9, 2019, Glasgow, Scotland, UK*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300422

1 INTRODUCTION

William Playfair (1759–1823) invented the barchart in 1786 [25] to help members of the British parliament—many of them illiterate—understand political and economic data without the need for actual numbers and text [12, 13]. Barcharts convey values for items using the length or width of a rectangle as visual marks, one per item. The barchart has since become one of the most prolific and familiar types of statistical data graphics [3], and is a staple in virtually any visualization tool and toolkit. One common use of the barchart is to visualize the relative values of specific entities, such as the gross domestic product of countries, the unemployment rate in U.S. states, or the enrollment in different academic units. Such lists are often sorted based on values, and we thus refer to them in this paper as "ranked lists" and their visualization as "ranked-list visualization."

Horizontal barcharts are the dominating ranked-list visualization [10, 14], but recent years has seen an increasing focus on improving the utility of even this basic visual representation. The main criticism is that for lists spanning more than a few dozen items, the entire barchart will not fit on one screen, and thus the list must be scrolled in order to view all of the items [10]. As a result, practitioners and academics alike have proposed alternatives to the scrolled barchart: Figure 1 gives an overview. Each of these representations have their own strengths and weaknesses. For example, treemaps [20] were originally designed for hierarchical data, but has seen common use in practice for ranked lists even if the representation is arguably not ideal for this purpose. Packed bubble charts [28] (Figure 2) use circular marks packed into tight configurations, their area conveying value. The wrapped bars technique [10], proposed by Stephen Few, addresses the scrolling problem by splitting the bars into columns on the same screen, but this makes comparison harder and reduces the horizontal 'data resolution. Even more recent techniques include packed bars [14, 15], piled bars [30], and Zvinca plots [11]. Given this bewildering array of ranked-list visualization techniques, the question for designers is which one is best for which specific task?

In this paper, we begin to answer this question by performing a crowdsourced graphical perception experiment evaluating the completion time and accuracy of these ranked-list visualizations for three different tasks: ranking one item, comparing two items, and averaging all items. We are particularly curious about the impact of interaction for scrolled barcharts, as well as the performance of treemaps for flat ranked lists. While our three tasks are low-level and not fully representative of the realistic use of these chart types, we argue that they are fundamental building blocks of higher-level tasks, such as determining the distribution, finding the extents and variance, and detecting anomalies, correlations, and trends in the data. Following in the grand tradition of graphical perception experiments in data visualization (e.g., [1, 4]), our purpose is thus to provide empirical findings on low-level perceptual aspects of these chart types.

To this end, we recruited 222 participants on Amazon Mechanical Turk and tested their performance for these three tasks and six of the ranked-list visualizations. Our results are mixed, but they do vindicate the use of treemaps, as that chart type did not perform consistently worse than some other chart types. Furthermore, our conclusion is that wrapped bars provide a familiar, compact, and interaction-friendly visual representation for ranked-lists that have the most balanced performance of charts studied in our experiment.

2 BACKGROUND

There is a long history of perceptual experiments in the area of statistical graphics, dating back to early work by Eells et al. [8] from 1926, well before computers were able to generate such graphics. Other early efforts include Croxton et al., who compared barcharts with circle diagrams and piecharts in 1927 [6], as well as investigated the effectiveness of various shapes for comparison in 1932 [5]. Peterson et al. [24] in 1954 measured the accuracy for eight different statistical graphs, providing some guidelines on their relative effectiveness.

Later, Cleveland and McGill [4] collected results from a large number of studies to rank visual variables in their order of effectiveness. These so-called *graphical perception* studies measure the ability for a person to retrieve the data presented in the chart by decoding the visual representation [22]. Representative such studies include work on simple charts by Simkin and Hastie [27], size and layering in horizon graphs [17], and perception for a range of time-series charts [19]. Some efforts have attempted to measure graphical perception based on a cognitive approach [18, 21].

While graphical perception studies are typically costly and time-consuming to perform, results have suggested that such studies can be easily crowdsourced using online marketplaces such as Amazon Mechanical Turk [16]. Such crowdsourcing methods, while not always ideal for general visualization evaluation due to the relative low expertise of typical crowdworkers, have been found to match laboratory

studies for graphical perception tasks, which merely rely on low-level visual machinery that any person possesses.

3 DESIGN SPACE: RANKED-LIST VISUALIZATION

Here we survey the design space of ranked-list visualization, first by delineating the basic requirements for what we consider a ranked-list visualization, and then by presenting a mini-taxonomy of such techniques. We then review each relevant technique and discuss its properties. This design space thus serves as a justification for which chart types were included and excluded, respectively, in this study.

Basic Requirements

Similar to prior work by Yalçin et al. [30], we consider only ranked-list visualizations that fulfill the following criteria:

- **No aggregation:** Each individual item in the list must be distinguishable, and this cannot be grouped together or summarized; in other words, the visual representation must be a *unit visualization* [23]. While aggregated ranked-list visual representations exist, we consider them outside the scope of this work since we regard each individual item as significant.
- **Value representation:** In addition to the identity (label) of the data item, the representation must be able to visually convey a value for each of the items (such as population, age, or income).
- **Overlap avoidance:** To enable visibility of all items, we require that the chart does not allow overdraw. (While piled bars technically involve overdraw, and Zvinca plots can yield overdraw in pathological situations, both charts are designed to minimize overlap.)

Taxonomy of Ranked-List Visualization

We derive the following properties that we can use to classify a ranked-list visualization:

- **Visual mark:** Graphical shape representing items.
- **Encoding:** Visual channel used for value.
- **Baseline:** Whether the technique has one or more common baselines for comparing visual marks.
- **Layout:** Algorithm for determining mark position.
- **Space utilization:** How well available space is used.
- **Resolution:** Screen resolution devoted to conveying item values. The more chart space is allocated to shapes for conveying item values, the higher the discriminability of values. Inspired by the resolution measure proposed by Heer er al. [17].

See Table 1 for our classification of relevant ranked-list visualizations. Table 2 covers the labeling strategy for each technique; while we do not include labels in our graphical perception study, this is an important consideration for any realistic use of a ranked-list visualization.

Barcharts

The most straightforward way to represent a ranked list is through a list of horizontal bars with a common baseline, where each bar represents an item and its length encodes the value (Figure 1a). Negative values can either be represented by bars that go left from a common origin, or communicated using a divergent color. Labeling is trivial, as the label can simply be drawn on top of or next to each bar.

Because the number of items to display may be more than can be contained on the screen, barcharts generally need to support scrolling, where the viewport can be moved up and down; hence we use the term *scrolled barcharts* in this paper. This is a drawback, as interaction will consume time and effort. However, since the chart uses the full width of the available space, its accuracy is high. On the other hand, skewed data distributions may result in wasted display space.

Treemaps

Treemaps were originally proposed by Johnson and Shneiderman [20] in 1991 to represent hierarchical data, such as a computer file system, ontology, or organizational chart, using the principle of *space enclosure* (Figure 1b). Under this principle, children are entirely enclosed by (and packed into) their parents, typically represented using rectangular shapes. Furthermore, the size of each shape is often used to convey a secondary value, such as a file size, the number of children, or stock market performance. However, in recent practice, treemaps are increasingly being used for non-hierarchical data, where there is no space enclosure and thus only organized using the packing layout algorithm. For a ranked list, sophisticated algorithms such as squarified treemap layouts [2] (which are now defaults in visualization software) yield a deterministic layout that encodes the value ranking in an accessible pattern.

Treemaps are *space-filling*, i.e., they use the full 2D space of the chart with no wasted space. Thus, they are not restricted to horizontal bars, and can therefore generally scale to a large number of items. However, the drawback is that the encoded value is conveyed using the *area* of the rectangles representing the items. Seminal results in graphical perception [4] hold that assessing area is significantly more difficult than assessing length. For this reason, a treemap should be less well suited for understanding ranked values than bars, which use length. However, we also speculate that a deterministic layout (as mentioned above) may assist perceptual tasks.

Packed Bubble Chart

Packed bubble charts [28], sometimes just called packed bubbles or bubble charts, is similar to treemaps in that they use the area of their visual marks—circles rather than the

Technique	Visual mark	Encoding	Baseline	Layout	Space util.	Resolution
scrolled barchart	horizontal bar	length	common	row-major	poor	full chart width
treemap [20]	rectangle/square	area	–	space-filling	optimal	full chart area
packed bubbles [28]	circle	area	–	packing	poor	half chart area†
wrapped bars [10]	horizontal bar	length	per column	rows + columns	suboptimal	chart width / #cols
piled bars [30]	horizontal bar	position	common	cycling rows	suboptimal	full chart width
packed bars [14, 15]	horizontal bar	length	varying	packing rows	optimal*	full chart width*
Zvinca plots [11]	dot	position	common	cycling rows	suboptimal	full chart width

∗ = depends on data distribution. † = from numerical approximation.

Table 1: Classification of ranked-list visualizations that we consider in our study.

Technique	Labeling strategy	Clipped	Static visibility
scrolled barchart	on axis or left-aligned inside bar	no	all (subject to scrolling)
treemap [20]	inside rectangle	yes	most
packed bubbles [28]	inside bubble or with tag-lines	yes	most
wrapped bars [10]	left-aligned on axis	no	largest value group, on-demand for others
piled bars [30]	right-aligned inside bar	yes	most
packed bars [14, 15]	left-aligned baseline, others centered	yes	baseline bars and largest others
Zvinca plots [11]	left-aligned	no	smallest value group, on-demand for others

Table 2: Labeling strategies for ranked-list visualizations.

rectangles used in treemaps—to convey the encoded values (Figure 2). However, unlike treemaps and as the name suggests, packed bubble charts are generated by "packing" the circles together as closely as possible without overlapping. Most packed bubble layouts are based on placing each circle and then using collision detection to shrink the chart.

Not surprisingly, packed bubble charts share many of the same strengths and weaknesses as treemaps. However, the actual placement of each bubble on the chart means little.

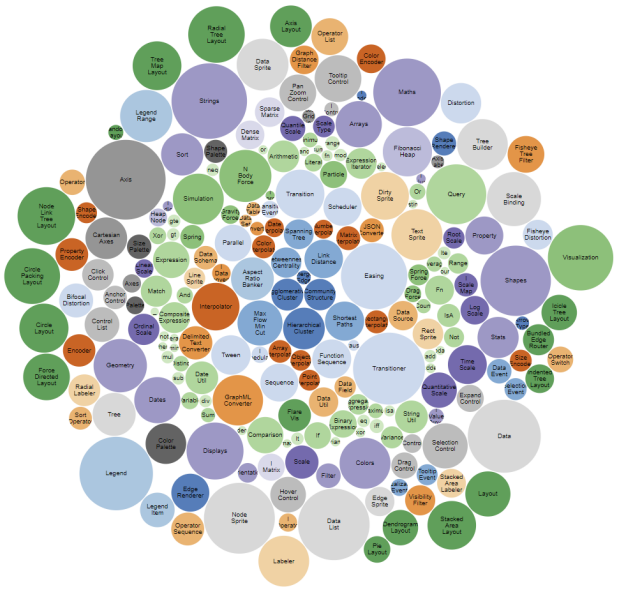


Figure 2: Packed bubble chart for a software class hierarchy. Image from D3 implementation by Mike Bostock (https://blocks.org/mbostock/4063269).

Wrapped Bars

Proposed by Stephen Few in 2013 [10], the design of *wrapped bars* is based on the observation that it is not necessary to use the full chart width for each bar. Instead, by splitting the list of N items into C columns, each with N/C items, we can organize each column horizontally to fit on screen (Figure 1c), thus eliminating the need for scrolling. Furthermore, because the list is sorted, the width of each individual column can be adapted to fit only the range of values it contains, and adapted scales can be shown for each column.

In terms of strengths and weaknesses, wrapped bars have the benefit of still using the length of horizontal bars to convey item values. Furthermore, while there is no longer a single common baseline for the entire chart, bars in each column share the same baseline (one per column). This, of course, makes it more challenging to directly compare items occupying different columns. The upshot is that the introduction of multiple columns means that the chart space can be better utilized than for single-column barchart lists, as

columns will get narrower as a side effect of the ranked order and the width of each column can be fitted to the size of the contained items. However, the columns cause the visual resolution for item values to be reduced since the horizontal chart space used to convey these values has been subdivided. This may make it harder to distinguish minute differences.

Packed Bars

The *packed bars* chart type was proposed by Xan Gregg [14, 15] in 2017, and essentially takes the bars of a scrolling barchart and packs them into a rectangular area (Figure 1d). In other words, instead of introducing multiple columns to avoid scrolling, packed bars add items as horizontal bars in sorted order until they fill the available rows on the screen. Then the technique uses a greedy layout algorithm to pack all of the remaining bars by placing them, one at a time, on the row with the most available horizontal space.

Packing has the benefit of resulting in efficient usage of the available screen space in most situations (although extremely skewed value distributions may result in lopsided layout with significant wasted space). However, packing means losing some of the order information of all bars except the first few rows that fit on the screen (typically the largest values). These first few rows will also have a common baseline, whereas all other bars will have no common baseline by virtue of being packed next to previously packed bars. While packed bars may provide high visual accuracy, this depends on the data distribution; for example, if the distribution causes bar of the largest item value to span the entire chart width, the visual accuracy will also be the full chart width. However, the pathological case here is where all item values are the same (or almost the same), as this will essentially reduce packed bars to wrapped bars, with its corresponding decreased visual resolution (but with no common column baselines).

Piled Bars

The *piled bars* technique [29, 30] builds on wrapped bars by splitting the items into columns, but instead of organizing the columns side-by-side in a horizontal layout, each subsequent column is *piled* on top of the previous column and thus uses the same common baseline (Figure 1e). This can be done without occlusion—i.e. without bars hiding each other—because items in the ranked list are sorted by the item values, which means that one column contains item with values that are guaranteed to be larger or equal than the values in the following column. To visually convey the piled behavior, the technique uses color gradients and shadows to suggest that a bar actually continues "underneath" smaller bars.

This approach combines the advantage of wrapped bars of fitting all items on a single screen while retaining the common baseline of standard scrolled barcharts. The chart can thus also use a common horizontal scale and grid lines, and tick marks. This makes it easier to compare items, even across columns, and it also results in higher visual resolution than for wrapped bars, since bars can use the full chart width. However, despite the gradients and shadows, the visual encoding is not trivial, as viewers may easily believe the bars are stacked instead of piled, i.e., that bars use the preceding bar as a baseline. Furthermore, the pathological case for piled bars is when all values in the list are the same (or almost the same), resulting in all bars having similar widths and thus being hard to distinguish. Finally, while we do not particularly focus on labeling in this design space treatment, similar bar widths will make labeling challenging.

Zvinca Plots

The last chart type we include in this discussion is *Zvinca plots* (Figure 1f), which was proposed in 2017 by Stephen Few based on an idea introduced by Daniel Zvinca (hence the name). While invented independently from Yalçin's piled bars [30], the techniques share the same basic idea: instead of using spatially separate columns, items are subdivided into groups to fit on the screen, and then the groups are drawn using a common baseline. However, rather than using horizontal bars, Zvinca plots merely use dots to signify the item values on the provided scale. This means that Zvinca plots entirely bypass the occlusion concern for piled bars, and have no need for color gradients or shadows to disambiguate between stacking and piling.

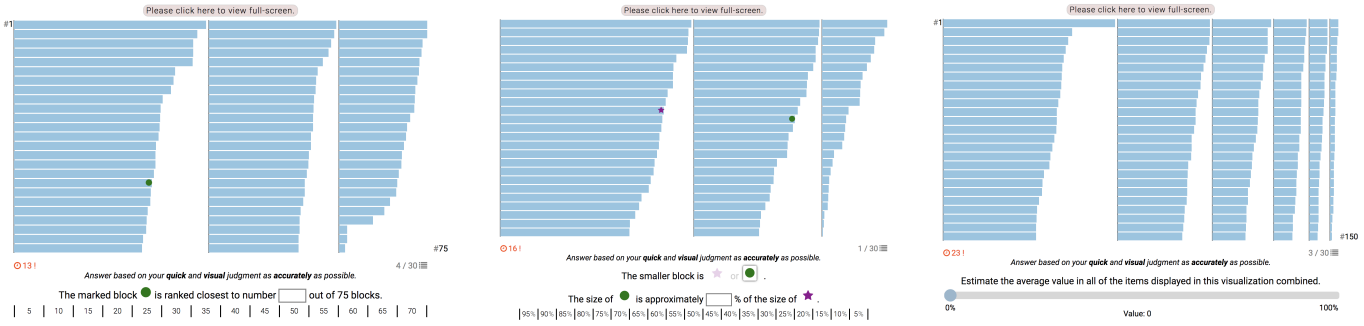
The relative strengths and weaknesses between Zvinca plots and piled bars are more or less arguable. Even if position is nominally the strongest visual channel [1], there is generally no significant advantage to using position rather than length with a common baseline [4], making Zvinca plots and piled bars approximately equivalent in this regard. The chart types share the same advantages for visual resolution, baselines, and space utilization. Zvinca plots manage occlusion and uniform data slightly more gracefully, and are easier to decode without the need for color gradients and shadows. Nevertheless, the two techniques are quite similar.

4 METHOD

To determine the optimal visual representation for ranked lists, we conducted a crowdsourced graphical perception study evaluating low-level visual performance involving six visualizations. We chose three tasks designed to test the gamut of low-level visual tasks. Finally, as we posit that different visual representations may scale differently depending on dataset size; for this reason, we also included three representative dataset sizes. Here we review our methods, and in the next section, we present our results.

Tasks and Data

Our focus in this work was to determine the perceptual characteristics of existing ranked-list visualizations. For this



(a) Rank task (one item). **(b) Comp task (two items).** **(c) Mean task (all items).**

Figure 3: Experimental interface for the three tasks Rank (left), Comp (center), and Mean (right).

reason, we wanted to choose low-level tasks restricted solely to visual perception rather than high-level tasks that are more relevant to data visualization. Our argument is that such low-level visual tasks are building blocks in higher-level tasks, which means that they will be reasonable indicators of the performance of these high-level tasks. This has the benefit of enabling us to recruit any participant with normal vision for our experiment. Furthermore, it also means we can disable labels and scales for our experiment, sidestepping legibility concerns altogether.[1] Nevertheless, we believe that, as with any graphical perception experiment, a study of high-level visualization tasks will eventually be necessary to provide ecological validity to complement our findings. That is outside the scope of the present study, however.

In determining representative low-level visual tasks to focus on, we based our selection on the *cardinality* of data items involved in the task: **one** item, **two** items, and **multiple** (or all) items. Our reasoning is that this data item cardinality yields qualitatively different low-level tasks. This lead us to deriving three concrete tasks as follows:

- T1 **Task 1: Rank (one item):** Given one selected item in a ranked list, determine its rank, i.e., its position in the full list (Figure 3a). We indicate the item using a colored icon centered inside the item's visual mark.
- T2 **Task 2: Compare (two items):** Given two selected items in a ranked list, determine which item is larger, and by how much (Figure 3b). We indicate the items using two colored icons centered inside the marks.
- T3 **Task 3: Mean (all items):** Given a ranked list of items, determine the average value of all items (Figure 3c). Participants respond by moving a slider to the ratio of 0% to 100% of the maximum value.

[1] Zvinca plots do not have an explicit labeling strategy, and packed bars do not label all items. Eliminating labels thus avoids ambiguous comparisons.

We generate datasets using a stochastic algorithm that iteratively perturbs random numbers in the desired direction using a form of simulated annealing (gradually decreasing amplitude) until the average, minimum, and maximum values are within a specific tolerance of the desired values.

Participants

Because this study focused on low-level perceptual tasks that require no specific training or prior data visualization expertise, we conducted our study using Amazon Mechanical Turk. While the use of Mechanical Turk (MTurk) means that we have little control over participant demographics and expertise as well as their computer hardware, prior work has shown that graphical perception tasks such as ours are particularly amenable to this kind of crowdsourced study [16].

In our experiments, each chart type and task combination (6×3) was answered by 10 participants, resulting in us recruiting a total of 180 crowdsourced participants across the three tasks. Each participant could only partake in one experiment, and thus a participant responded to only a single chart type and a single task type. We limited the study to Turkers with a historical performance of at least 90% approval rating as well as at least 1,000 HITs completed to ensure that we recruited only experienced crowdworkers. Furthermore, we limited participation to the United States due to tax and compensation restrictions imposed by our IRB. We screened participants to ensure at least a working knowledge of English; this was required to follow the instructions and task descriptions in our testing platform.

We intentionally did not collect demographic information to minimize the time required to complete an experimental session. The demographics should be consistent with the overall characteristics of the diverse Mechanical Turk worker pool [26]. All participants were ethically compensated at a rate consistent with an hourly wage of at least $10/hour (the

U.S. federal minimum wage in 2018 is $7.25). More specifically, the payout was $2.00 per session, and with a typical completion time of 10 minutes (no participant exceeded 12 minutes), this yielded an hourly wage of $12/hour.

Apparatus

Because of the crowdsourced setting, we were unable to control the devices that participants used to complete the experiment. However, to ensure that participants had a sufficiently large screen to reliably perform the experiment, we rejected participation using devices with less screen resolution than 1280×800 pixels. We maximized the browser window[2] and fixed the viewport size for the testing platform to 920×540 pixels.

Experimental Factors

In addition to the three tasks outlined above, we included two experimental factors:

- **Chart type (C):** The ranked-list visualizations that we wanted to compare. In reference to Section 3, we included scrolled barcharts (SB), treemaps [20] (TM), wrapped bars [10] (WB), packed bars [14, 15] (PaB), piled bars [30] (PiB), and Zvinca plots [11] (ZP). Figure 1 provides an overview.
  We opted to not include packed bubbles (bubble charts) because area-size charts are already represented by treemaps, which also uses a deterministic and sorted layout (whereas the packed bubbles layout is unpredictable and uses collision detection).
- **Dataset Size (D):** It is conceivable that different visual representations will perform differently depending on the number of items being displayed. For this reason, we involve an experimental factor for the number of items to display in the ranked list. Because of the typical intended use-cases of ranked lists in practice [10, 11], we opted to include three levels for this factor: 75 items, 150 items, and 300 items. We also base this choice on the prior evaluation by Yalçin et al. [30], who used these sizes, as well as our pilot studies.

We followed the convention that all bars should have equal height across all chart types (except for treemaps, which do not use bars). This means that the number of columns for wrapped and piled bars depends on the dataset size. Since we do model dataset size in our experiment, the number of columns is indirectly modeled: as low as 3 columns for 75 items, and as high as 10 columns for 300 items.

[2]Unfortunately, this can be blocked by some browsers, and we have no way of ensuring that the user does not change the window size after the fact.

Experimental Design

We used a mixed factorial design, where each participant worked on only one task and visualization, but across all dataset sizes. In other words, the chart C and task T factors were between-participants (BP), whereas data size and repetitions were within-participants (WP). The reason for this was to make each crowdsourced session manageable in duration—in our experience, keeping sessions less than 10 minutes in duration minimizes fatigue and maximizes attention for crowdworkers. This yielded the following design:

	6	**Chart** C (SB, TM, WB, PaB, PiB, ZP) [BP]
×	3	**Task** T (T1 - rank, T2 - comp, T3 - mean) [BP]
×	3	**Data Size** D (75, 150, 300 items) [WP]
×	10	repetitions [WP]
	540	trials (30 per participant)

With 180 participants (10 per each combination of task T and chart C, i.e., 60 per each chart type C), we planned to collect a total of 5,400 trials. For each trial, we also collected the completion time as well as the accuracy. The completion time was measured from the beginning of a trial until the participant submitted an answer. The accuracy measure was defined differently for each task:

- **T1 (rank) - accuracy:** Normalized and absolute difference between the actual rank and the participant response, e.g., $|a - b|/n$, where a was the correct rank, b was the participant answer, and n the number of items in the list (75, 150, or 300).
- **T2 (compare) - accuracy:** Normalized and absolute difference between the actual ratio of the larger value to the smaller value and the participant response, e.g., $|a - b|$, where a was the correct proportion between bars, and b was the response.
- **T3 (mean) - accuracy:** Normalized and absolute difference between the actual average and participant response, e.g., $|a - b|$, where a was the correct average, and b was the response.

Hypotheses

We formulate the following hypotheses for our experiment:

H1 *Scrolled barcharts (SB) will perform significantly slower than all other visualizations.* We believe the necessary interaction to scroll through the list will result in the scrolled barcharts requiring a longer completion time than all other visualizations.

H2 *Treemaps (TM) will yield significantly less accurate performance than all other visualizations for all tasks.* Assessing area is significantly less accurate than assessing lengths or position.

These were formulated prior to running the experiment. They correspond to our motivations for conducting this work

in the first place: our intuition is that (1) the scrolling interaction required for a long list of bars will slow down performance, and (2) that the use of treemaps to represent flat lists of ranked items is inefficient.

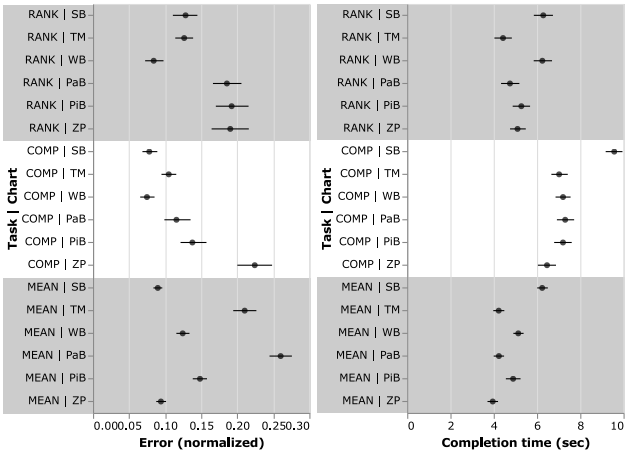


Figure 4: Overall error and completion time for all charts per task type. Error bars show 95% confidence intervals.

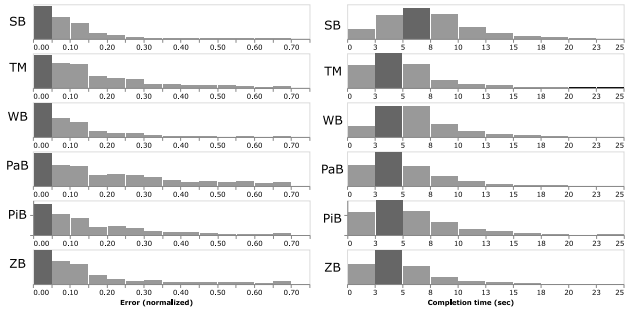


Figure 5: Overall error and completion time distributions.

5 RESULTS

We ran our crowdsourced graphical perception study on Amazon Mechanical Turk and collected a total of 6,684 responses from 222 unique respondents. This was higher than the 180 that we planned, but software errors with the testing platform yielded duplicated trials in the data. We eliminated the extra and incomplete trials. Furthermore, we eliminated completion time outliers that were four times larger than the standard deviation for each task. Following current best practices for fair statistical in HCI, as summarized by Dragicevic [7], we eschewed traditional null hypothesis statistical testing (NHST) in favor of estimation methods to derive 95% confidence intervals (CIs) for all results datasets. More specifically, we employed non-parametric bootstrapping [9] with $R = 1,000$ iterations.

Figure 4 shows the overall error and completion time for all tasks and chart types, whereas Figure 5 show data distributions of the same. We will discuss each task in detail in the following subsections, but we can make a few observations already from this overview. For example, there is good evidence to suggest that SB (scrolled barchart) is overall the most accurate condition, except for the Rank task, where WB (wrapped bars) is more accurate. On the other hand, the results suggest little differentiation between PaB and PiB (packed and piled bars, respectively), except for the Mean task, where packed bars seem to have the most errors, and ZP (Zvinca plots) are similarly accurate as SB. Zvinca plots in general show uneven performance, with seemingly the least accurate of all charts for Comp, likely comparable to PaB and PiB for Rank, and likely comparable to SB for Mean, as mentioned above. Treemaps (TM) did surprisingly well, with only Mean exhibiting what seems to be lower accuracy than all but PaB (packed bars), otherwise yielding good accuracy.

As for completion time, there is evidence that SB (scrolled barchart) is slower than alternatives for all tasks. It is only for the Rank task that WB (wrapped bars) somewhat surprisingly seem to perform comparable than SB and slower than all other charts. Beyond these observations, PaB and PiB seem to perform comparably well for all tasks. ZP (Zvinca plots) shows completion times comparable to the other techniques for Comp and Rank, but seem to outperform the others for the Mean task. Finally, treemaps (TM) do surprisingly well, particularly for the Rank task.

Task 1: Ranking (Single Item)

The left columns of Figure 6 shows the error for the Rank task. As observed above, wrapped bars (WB) overall exhibits the most accurate performance, whereas the advanced techniques—PaB, PiB, and ZP—overall seem to perform poorly. In particular, PiB has high variance in error for 300 records, and ZP also shows a similar trend. The most surprising finding here is that TM does not nearly perform the least accurate, and what's more, there is an inverse linear trend for increasing number of items in the list.

For completion time in the left part of Figure 7, a point of note is that SB seems to perform more slowly than other techniques. Curiously, ZP exhibits an inverse linear completion time trend for increasing number of items. This is also the task where WB overall performs relatively poorly.

Task 2: Comparison (Two Items)

The center column of Figure 6 give the error for the Comp task. Most techniques perform accurately here, with TM even seeming to outperform PaB and PiB. Evidence suggests that Zvinca plots had the lowest accuracy for all sizes.

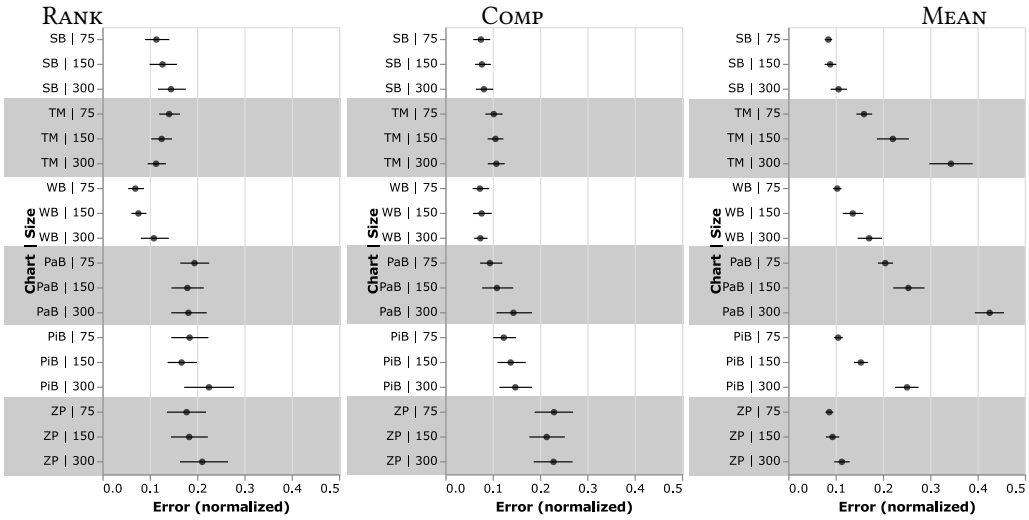


Figure 6: Error for all charts for all tasks across list sizes. Error bars show 95% confidence intervals.

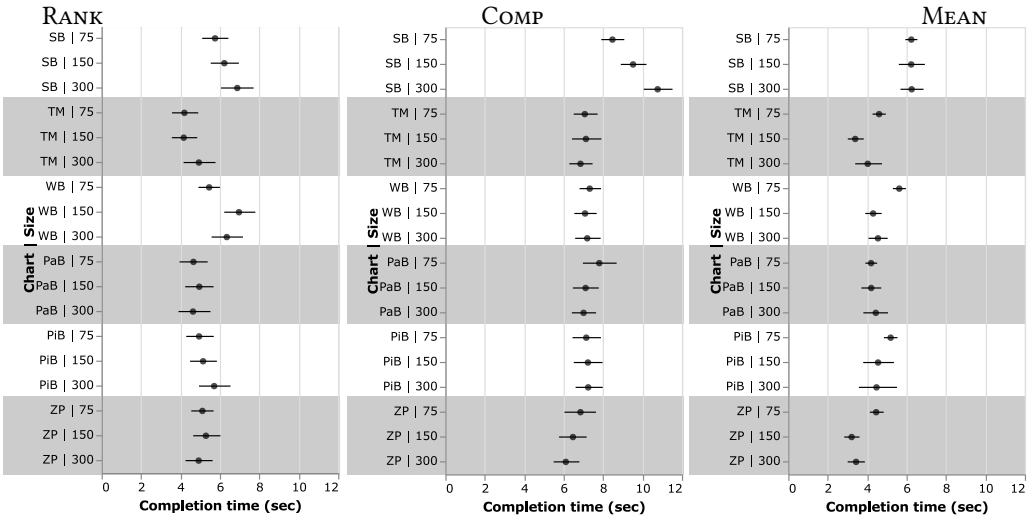


Figure 7: Completion time for all charts for all tasks across list sizes. Error bars show 95% confidence intervals.

The Comp task also gave rise to the longest completion times (Figure 7), particularly for SB (scrolled barchart). All other charts seem to have comparable performance.

Task 3: Average (All Items)

Finally, the results for the Mean task is shown in the right column of Figure 6. This was overall a difficult task, with many techniques yielding high error rates—particularly PaB, TM, and to some extent PiB. These three techniques were particularly sensitive to increasing sizes, as the error rate went up significantly for higher list sizes. The findings may indicate that ZP performed the most accurate here, with SB as the second most accurate, followed by WB.

This task also yielded the most varied completion times, as evidenced by Figure 7. Interestingly, ZP here exhibits an inverse completion time trend; it seems participants were able to respond faster with increasing list sizes.

6 DISCUSSION

Based on our results, we can make the following conclusions about our hypotheses (Section 4):

- Scrolled barcharts performed slower for the Comp and Mean tasks, but evidence suggests it outperformed wrapped bars for the Rank tasks. This is evidence partially in favor of **H1**.

- Surprisingly, our findings suggest that treemaps were never the least accurate of the chart types, and in fact outperformed several charts for both the RANK and COMP tasks. This does not support **H2**.

In the below sections, we will first attempt to explain these results, and then we will discuss their generalizations.

Explaining the Results

There are several findings from our study—some surprising, some not—that require further explanation. First of all, on the matter of scrolled barcharts, which all of the competing techniques were designed to beat, the picture is mixed. While the technique is mostly slower than other charts, it does provide the highest accuracy. The reason for its slow speed is obviously that scrolled barcharts—unlike the other techniques, where the entire dataset is visible on the screen at the same time—requires scrolling (i.e., user interaction) to see the full data. Conversely, the highest accuracy is likely due to its simple, uncluttered, and familiar representation. On the other hand, our scrolled barchart implementation saves horizontal space by folding the labels on top of the bars (Figure 1a), whereas many practical implementations dedicate horizontal space to the left of the axis for labels.

Treemaps perform surprisingly well, which goes against visualization wisdom, which tends to promote length over area judgment [4]. It is also not consistent with recent findings from Yalçin et al. [30]. While treemaps did not ever perform the best in completion time or accuracy, it also never performed the worst. In fact, for the mean task, where it arguably performed the worst, you could argue that the conversion from an area mark to a slider when answering the average size question was potentially problematic for the treemap condition. One potential explanation may be that the squarified treemap layout [2] organizes rectangles in a way such that the position is an indicator of rank, which may be helping the treemap representation. Other layouts may not exhibit the same helpful property.

Save for wrapped bars, the more advanced techniques that rely on creative layouts to keep all bars on a single screen performed relatively poorly. This is surprising, but may partially be explained by unfamiliarity compared to scrolled barcharts, as well as arguably wrapped bars, which retain many familiar features of the former. However, that argument holds less water when considered against treemaps, which are not known to be familiar to a lay audience. Instead, this may stem from the complex layouts of piled bars, where longer bars are overlapped by shorter bars, as well as packed bars, where bars are packed in an unpredictable manner. Finally, Zvinca plots use dot position rather than bar length, and overplotting may potentially be a factor.

One point about Zvinca plots stand out, however: for the MEAN task, ZP performed both the fastest and had the lowest error rate. This is remarkable, and could be explained by the fact that the smaller amount of pixels associated with dots than with bars simply affords easier visual estimation. Another way to look at this task for Zvinca plots is to determine the geometric center for the plots, which is different from the other representations and possibly easier. Alternatively, it may just be an corollary from known graphical perception results, such as that of Cleveland and McGill [4], which states that position is a stronger visual cue than length.

Generalizing the Results

What do these results say about the state of ranked-list visualization? First of all, we think that our treemap findings should be seen as a result cautiously in favor of continuing to use treemaps for flat ranked lists, which is already prevalent in practice. While this representation was never intended for flat lists, our study indicates that treemap layouts can also be utilized to great effect even without a hierarchy.

Having said that, there are better alternatives for ranked lists than treemaps; for example, wrapped bars seem to have comparable accuracy to scrolled barcharts for most settings, and is faster to use in the majority of cases. For this reason, wrapped bars may be the overall most balanced choice.

There are two potential weaknesses that we have not considered in this work: scalability and ecological validity. For the former, it is important to note that we only considered lists of up to 300 items. While many datasets that are viewed as ranked lists commonly only have a few hundred items, these are clearly still small. When looking for a technique that scales to large datasets, many of the design considerations and results discussed here fade. Instead, a designer may pick a technique that uses space optimally—e.g., treemaps—or utilizes less ink—e.g., Zvinca plots. Investigating such scalability issues is left for future work.

As for the ecological validity concern, our stated goal in this work has always been to study low-level perceptual aspects of ranked list visualization. Our argument is similar to most perception studies in that performance for these perceptual aspects will combine into higher-level compound tasks. Of course, high-level analytical tasks actually used in practice may look very different compared to the three tasks studied here. First of all, tasks with completion times on the order of a few seconds are rarely significant in sensemaking practice, where other, more intangible factors come into play. For example, packed bars promote the primary bars (the first column) over secondary bars, and piled bars optimize the horizontal resolution and discriminability, both properties that may be important for a specific task. Second, these high-level analytical tasks are conducted by experts with long experience and training in sensemaking, and thus their

needs, requirements, and wishes may be very different from the casual users we surveyed in our crowdsourced study. However, just as for matters of scale, studying high-level analytical practice for ranked-list visualization is a question we have to leave open for future research.

7 CONCLUSION AND FUTURE WORK

We have presented results from a crowdsourced graphical perception on low-level tasks for ranked-list visualization: ranking an item in a list, comparing two items, and estimating the average value of all of the items in the list. In conducting this work, we involved all of the primary chart types that are typically used for such data in practice: scrolled lists of barcharts, treemaps, wrapped bars, piled bars, packed bars, and Zvinca plots. While no single effect can be found in our results, we do find evidence that each chart type has strengths and weaknesses depending on the task, data, and user. However, our results do indicate that barchart lists provide high accuracy at the cost of scrolling, that treemaps are not nearly as inaccurate as their reputation suggests, and that wrapped bars may provide a powerful middle ground in mitigating the interaction costs associated with long lists.

Our future work will involve both studying the scalability aspects of ranked-list visualization, as well as exploring high-level analytical tasks conducted by data scientists. We are curious to see if any of our recommendations will change as an effect of these changing parameters, both in terms of the number of items in the list, as well as in terms of the skill level, task type, and unique needs of an expert audience.

ACKNOWLEDGMENTS

This work was supported by U.S. National Science Foundation award IIS-1539534 (http://www.nsf.org/). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

[1] Jacques Bertin. 1983. *Semiology of Graphics.* University of Wisconsin Press, Madison, Wisconsin.

[2] Mark Bruls, Kees Huizing, and Jarke J. van Wijk. 2000. Squarified Treemaps. In *Proceedings of the Joint Eurographics/IEEE VGTC Symposium on Visualization*. Eurographics Association, Geneva, Switzerland, 33–42. https://doi.org/10.1007/978-3-7091-6783-0_4

[3] William S. Cleveland. 1994. *Visualizing Data.* Hobart Press, Summit, NJ, USA.

[4] William S. Cleveland and Robert McGill. 1984. Graphical Perception: Theory, Experimentation and Application to the Development of Graphical Methods. *J. Amer. Statist. Assoc.* 79, 387 (Sept. 1984), 531–554. https://doi.org/10.2307/2288400

[5] Frederick E. Croxton and Harold Stein. 1932. Graphic Comparisons by Bars, Squares, Circles, and Cubes. *J. Amer. Statist. Assoc.* 27, 177 (1932), 54–60. https://doi.org/10.2307/2277880

[6] Frederick E. Croxton and Roy E. Stryker. 1927. Bar charts versus circle diagrams. *J. Amer. Statist. Assoc.* 22, 160 (1927), 473–482. https://doi.org/10.2307/2276829

[7] Pierre Dragicevic. 2016. Fair Statistical Communication in HCI. In *Modern Statistical Methods for HCI*, Judy Robertson and Maurits Kaptein (Eds.). Springer, Berlin, Heidelberg, Germany, 291–330. https://doi.org/10.1007/978-3-319-26633-6_13

[8] Walter C. Eells. 1926. The relative merits of circles and bars for representing component parts. *J. Amer. Statist. Assoc.* 21, 154 (1926), 119–132. https://doi.org/10.2307/2277140

[9] Bradley Efron. 1992. Bootstrap methods: another look at the jackknife. In *Breakthroughs in Statistics*. Springer, Berlin, Heidelberg, Germany, 569–593.

[10] Steven Few. 2013. Wrapping Graphs to Extend Their Limits. https://www.perceptualedge.com/articles/visual_business_intelligence/wrapping_graphs_to_extend_their_limits.pdf. In *Visual Business Intelligence Newsletter*.

[11] Stephen Few. 2017. The Journey to Zvinca. https://www.perceptualedge.com/articles/visual_business_intelligence/journey_to_zvinca.pdf. In *Visual Business Intelligence Newsletter*.

[12] Paul J. FitzPatrick. 1960. Leading British Statisticians of the Nineteenth Century. *J. Amer. Statist. Assoc.* 55, 289 (March 1960), 38–70. https://doi.org/10.2307/2282178

[13] Michael Friendly. 2007. A Brief History of Data Visualization. In *Handbook of Computational Statistics: Data Visualization*, Vol. III. Springer, 15–56. https://doi.org/10.1007/978-3-540-33037-0_2

[14] Xan Gregg. 2017. Introducing packed bars, a new chart form. https://community.jmp.com/t5/JMP-Blog/Introducing-packed-bars-a-new-chart-form/ba-p/39972.

[15] Xan Gregg. 2017. Introducing the Packed Bars Chart Type. In *Poster Proceedings of IEEE VIS*.

[16] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using Mechanical Turk to assess visualization design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 203–212. https://doi.org/10.1145/1753326.1753357

[17] Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. 2009. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualization. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1303–1312. https://doi.org/10.1145/1518701.1518897

[18] Weidong Huang, Peter Eades, and Seok-Hee Hong. 2008. Beyond time and error: a cognitive approach to the evaluation of graph drawings. In *Proceedings of BELIV*. 1–8.

[19] Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. 2010. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 927–934. https://doi.org/10.1109/TVCG.2010.162

[20] Brian Johnson and Ben Shneiderman. 1991. Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. In *Proceedings of the IEEE Conference on Visualization*. IEEE, Piscataway, NJ, USA, 284–291. https://doi.org/10.1109/VISUAL.1991.175815

[21] Gerald L. Lohse. 1993. A cognitive model for understanding graphical perception. *Human-Computer Interaction* 8, 4 (1993), 353–388. https://doi.org/10.1207/s15327051hci0804_3

[22] Jerry Lohse. 1991. A Cognitive Model for the Perception and Understanding of Graphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 137–144. https://doi.org/10.1145/108844.108865

[23] Deok Gun Park, Steven M. Drucker, Roland Fernandez, and Niklas Elmqvist. 2018. ATOM: A Grammar for Unit Visualization. *IEEE*

Transactions on Visualization & Computer Graphics 24, 12 (2018), 3032–3043. https://doi.org/10.1109/TVCG.2017.2785807

[24] Lewis V. Peterson and Wilbur Schramm. 1954. How accurately are different kinds of graphs read? *Educational Technology Research and Development* 2, 3 (June 1954), 178–189. https://doi.org/10.1007/BF02713334

[25] William Playfair. 1786. The Commercial and Political Atlas: Representing, by Means of Stained Copper-Plate Charts, the Progress of the Commerce, Revenues, Expenditure and Debts of England during the Whole of the Eighteenth Century.

[26] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who are the crowdworkers?: shifting demographics in Mechanical Turk. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2863–2872. https://doi.org/10.1145/1753846.1753873

[27] David Simkin and Reid Hastie. 1987. An Information-Processing Analysis of Graph Perception. *J. Amer. Statist. Assoc.* 82, 398 (June 1987), 454–465. https://doi.org/10.1080/01621459.1987.10478448

[28] Weixin Wang, Hui Wang, Guozhong Dai, and Hongan Wang. 2006. Visualization of large hierarchical data by circle packing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 517–520. https://doi.org/10.1145/1124772.1124851

[29] Mehmet Adil Yalçın, Niklas Elmqvist, and Benjamin B Bederson. 2017. Piled Bars: Dense Visualization of Numeric Data. In *Poster Proceedings of the Graphics Interface Conference*.

[30] Mehmet Adil Yalçin, Niklas Elmqvist, and Benjamin B. Bederson. 2017. Raising the Bars: Evaluating Treemaps vs. Wrapped Bars for Dense Visualization of Sorted Numeric Data. In *Proceedings of the Graphics Interface Conference*. ACM, New York, NY, USA, 41–49. https://doi.org/10.20380/GI2017.06