Automatic Y-axis Rescaling in Dynamic Visualizations

Jacob Fisher* Columbia University
Remco Chang† Tufts University
Eugene Wu‡ Columbia University

(a) Rescale whenever date range is moved

(b) Rescale when clicking on buttons (red)

(c) No rescaling

(d) A single rescale to transition from historical to predicted future weather

Figure 1: Four interactive visualizations that have different rescaling policies for the y-axis. (a) COVID-19 infection statistics rescales whenever the chart data changes in response to range slider interactions. (b) UK Balance of Trade on Tableau Public rescales whenever the user clicks a different year or statistic. Note chart is flipped, so the "y-axis" is horizontal. (c) Gapminder visualization uses fixed axes that are not rescaled during the animation and slider interactions. (d) New York Times animates in temperature over time as the user scrolls down, and rescales the y-axis to transition from historical to future temperatures.

2021 IEEE Visualization Conference (VIS) | 978-1-6654-3335-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/VIS49827.2021.9623319

Abstract

Animated and interactive data visualizations dynamically change the data rendered in a visualization (e.g., bar chart). As the data changes, the y-axis may need to be rescaled as the domain of the data changes. Each axis rescaling potentially improves the readability of the current chart, but may also disorient the user. In contrast to static visualizations, where there is considerable literature to help choose the appropriate y-axis scale, there is a lack of guidance about how and when rescaling should be used in dynamic visualizations. Existing visualization systems and libraries adapt a fixed global y-axis, or rescale every time the data changes. Yet, professional visualizations, such as in data journalism, do not adopt either strategy. They instead carefully and manually choose when to rescale based on the analysis task and data. To this end, we conduct a series of Mechanical Turk experiments to study the potential of dynamic axis rescaling and the factors that affect its effectiveness. We find that the appropriate rescaling policy is both task- and data-dependent, and we do not find one clear policy choice for all situations.

Index Terms: Human-centered computing—Visualization—Empirical studies in visualization; Human-centered computing—Visualization—Visualization theory, concepts and paradigms

1 Introduction

Scales and axes provide powerful contextual cues to help users interpret data visualizations. Considerable prior work has established guidelines to choose the appropriate scale transformation [2,13] (e.g., linear, log), set the domain of the scale based on the data [12], choose ticks and labels [9, 11, 13], and maintain consistency of scales between multiple views [7]. In addition, automatic visualization tools can help recommend the appropriate scale transformations [4–6]. These studies were primarily in the context of static visualizations.

In contrast, dynamic visualizations—such as animated or interactive visualizations—change the rendered data as the animation progresses, or in response to user interactions. Each frame of the visualization presents a different set of data in the chart. A naive approach is to use existing guidelines to use a separate y-axis scale for each frame. (This paper uses the term "y-axis" to refer to the encoding channel for the visualization's measure attribute.) However, a y-axis that changes every frame removes a shared reference frame and can disorient the user. At the other extreme, the visualization may use a single global y-axis whose domain contains the minimum and maximum of all possible data that can be rendered in the visualization. However, this can obscure the data in any given frame. The following is an example of this trade-off:

Example 1 (Interactive COVID-19 Visualization) *Figure 1(a) is an interactive visualization that depicts the number of COVID-19 infections over a 4 day window. The window can be moved by dragging the date range an the bottom of the visualization. The figure depicts three adjacent windows in February when the number of cases spikes from 3500 (bottom frame) to 18000 to 2500 (top frame). In this visualization, the y-axis rescales whenever the data changes (when the slider moves). This always ensures that the data uses the full height of the chart, but can disorient the user. However, a fixed scale of* $[0, 18000]$ *makes it difficult to perceive the changes in data in the bottom and top frames.*

Qu et al. [7] proposed a consistency constraint model to evaluate how authors maintain encoding and scale consistency across multiple static views. They propose a consistency criteria that the same data field should be encoded using the same scales (the same data domain and retinal range). If directly applied to dynamic visualizations, where the retinal range and the data fields remain constant, this criteria implies that the domain of the scales should be fixed across the frames. On the other hand, their participants tended to ignore this consistency criteria when it would lead to "*too much whitespace*" that would result in "*leaving details and trends harder to see in that view*". This suggests situations where rescaling may be desirable.

Choosing the appropriate rescaling strategy is challenging because there are many possible rescaling policies. For a chart parameterized by a slider with n slider values, there are 2^{n-1} possible rescaling policies: each frame be rescaled (which we call a *breakpoint*) or reuse the previous frame's scale.

In practice, existing visualization systems such as Tableau [1], and libraries such as Vega-lite [8], use one of two extreme policies: rescale on every change, or never rescale by using a fixed axis. However, these policies may not be optimal. Professionally

*e-mail: jacob.fisher@columbia.edu
†e-mail: remco@cs.tufts.edu
‡e-mail: ewu@cs.columbia.edu

978-1-6654-3335-8/21/$31.00 ©2021 IEEE
DOI 10.1109/VIS49827.2021.00031

designed interactive visualizations often use intermediate policies. For instance, Figure 1(d) is a New York Times climate article that depicts temperature from 1960 to (projected) 2090. As the user scrolls down, data is animated in from 1960 to 2017; the y-axis is then rescaled to signify the contrast between historical and predicted future temperatures.

In this paper, we conduct a preliminary crowdsourced study on rescaling policies. Participants perform visual analysis tasks by interacting with a slider widget (Figure 2). Our motivation is to develop rescaling guidelines for designers that are building dynamic visualizations or for automatic interactive visualization design systems [1, 8, 14].

Our main hypothesis is that the effects of a given rescaling policy depends on the type of analysis task (H1). Specifically, rescaling on every frame may improve the accuracy of tasks that compare data within individual frames, whereas using a fixed global scale is beneficial for tasks that compare across frames. Our second hypothesis is that an "intermediate" policy—where it only rescales when the frame's y-axis data domain changes considerably—is beneficial across different analysis tasks (H2).

Regarding H1, our studies find that axis rescaling affects participant task accuracy and/or latency depending on the dataset and task, at least in the line chart examples examined. For datasets that do *not* exhibit large variations, global rescaling is generally effective across tasks, while per-chart rescaling negatively affects latency and accuracy. However, when the variation is larger, per-chart rescaling can benefit tasks that require comparing marks within a chart. Regarding H2, breakpoint policies appear to strike a robust middle-ground across data and tasks: although they do not clearly out-perform global nor per-chart, they also do not clearly perform worse. These findings suggest that axis rescaling is a promising tool when designing interactive visualizations, and that automatic methods for choosing breakpoints can be helpful.

2 Studies Overview

We conducted a series of three Mechanical Turk user studies to compare different rescaling policies in an interactive slider-based visualization: a single global scale, per-chart scaling method, and breakpoint policies that vary the number of slider positions where the y-axis is rescaled (a breakpoint). All three studies were designed to test both H1 and H2. After each study, we re-assessed participants' reactions to our data, tasks, and conditions, and altered them accordingly for later studies. Specifically, we modify our data to amplify extremities and remove a breakpoint condition for the second study, as it did not provide new information from another condition. For the third study, we again modify our data, removed a task, and replaced one breakpoint condition with a breakpoint condition that includes markers for breakpoints, in order to test if these markers make breakpoints easier to use. Additionally, we run the third study as within-subjects.

Shared Protocols In all studies, participants were presented with a line chart and a slider that controls the chart data (Figure 2). The chart displays monthly crude oil prices for a given year, and the slider varies the year. We manually selected the breakpoints so that the variation in the data along the y-axis was minimized for the frames between the breakpoints. To familiarize participants with the interface, they first complete 4 qualification tasks where they are asked to manipulate the slider and state whether the y-axis scale changes at $1-5$ times, >5 times, or does not change. In each task, the participant clicks a start button to load the visualization and start the timer. She then clicks a "ready to submit" to hide the chart and enter the answers to the task. We further ask for qualitative feedback and comments on the task itself. At the end of the study, participants are asked a brief demographic survey (age group and education level). We paid participants equivalent to $15/hr for finishing all tasks. This came out to $1.50 for studies 1 and 2 and $4 for study 3.

Figure 2: The user interface for the pilot study.

2.1 Pilot Study

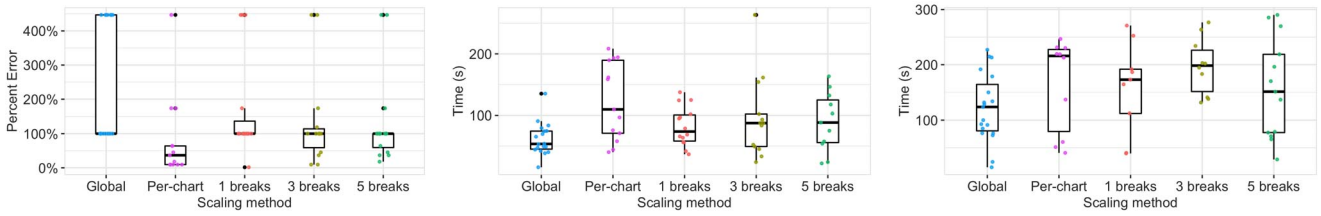
We first ran a between-subjects pilot study to understand the extent different policies affect a participant's task performance, and to assess three types of tasks. We used 5 policies: global, per-chart, and either 1, 3, or 5 breakpoints. This study was run as a between-subject study. Participants were randomly assigned to one policy. Participants were first given a training/qualification task to familiarize them with the interface and their assigned rescaling policy. The task asked "Is there a month in 1999 where the oil price was above $75?" and is unrelated to the main tasks.

Participants then completed three tasks in random order. (T1) "What year had the highest March price? What is its price in that year?"; (T2) "What is the difference in prices between April 1998 and September 1998?"; (T3)"In how many years was the price for August within $5 of the price for February?" Task 1 compares data across frames, and we hypothesized the global policy would be best (H1.1). Task 2 compares within a frame, and we hypothesized per-chart would be best (H1.2). Task 3 compares both within and between frames, and we hypothesized the breakpoint policies would be best (H1.3), though we were not certain which variant. We used the Brent Oil Prices dataset from 1988 to 2018 because of large oil price fluctuations in the dataset. In particular, the oil prices start relatively low, rise by an order of magnitude, and then fluctuate quite a bit. Our motivation for using this type of dataset was that the global policy could perform poorly because prices in earlier years would appear "squashed".

2.1.1 Results

We collected data from 88 participants. We removed the top and bottom 10% of task answers, and responses completed within 5 seconds. We report one-way ANOVA tests for latency and relative response error, where the factor is the scaling policy. p-values under the 0.05 level of significance are in bold. Note that Task 1 asks participants to estimate the price and year, though the middle 80% of responses for year were all correct, so ANOVA was not run for that data point.

Table 1 shows that the rescaling policy was not significant for latency nor error in Task 1, due to the simplicity of the task. We found that latency, error, and latency were significant for tasks 1, 2, and 3 respectively. Figure 3a shows, as expected from H1.2, that error under global performs worse due to the large variation



(a) Response accuracy, Task 2. (b) Response latency, Task 1. (c) Response latency, Task 3.

Figure 3: Results from first Pilot study.

Study	Task	Latency	Error (price)	Error (year)
1	1	**0.017**	0.626	N/A
1	2	0.831	**0.045**	-
1	3	**0.016**	0.868	-
2	1	**0.025**	0.405	N/A
2	2	0.702	**0.000**	-
2	3	0.337	0.768	-

Table 1: P-values from ANOVA in first and second pilot studies.

in the data across the years, whereas the breakpoint conditions appear slightly worse than per-chart. Figure 3b and Figure 3c show that for these tasks, per-chart indeed slows participants down, and breakpoints are in between global and per-chart.

Overall, we found that the rescaling policy can affect task completion time and accuracy, and it depends on the task type. However, the range of the oil prices is only between $9.82 to $132.7, and we decided to evaluate data with larger variations to more clearly assess the differences between the policies (if any).

2.2 Second Pilot Study

We followed the procedure of the Pilot study, but removed the 1 breakpoint policy as it appeared identical to the 3 breakpoint policy. We again ran this study as a between-subjects study. We modified the dataset to greatly accentuate oil pricing differences—for each price p, we replaced its value with 1.15^p. Based on this data, we updated Task 2 to "What is the difference in prices between April 2000 and September 2000?", and Task 3 to "In how many years was the price for August within $10 of the price for February?".

2.2.1 Results

We collected data from 46 participants and ran one-way ANOVA on the 10% trimmed means for each task. Table 1 shows that the rescaling policy has a significant effect on latency in Task 1, but not accuracy. Figure 4 shows, as expected, that global is fastest, followed by 3 and 5 breakpoints, and finally per-chart. Per-chart is slowest because the participant must repeatedly assess the y-axis because it changes every frame. We again found a significant effect for error on Task 2 (Figure 5), and the difference between global and the other policies is more pronounced, as expected due to the data amplification.

We were surprised that the breakpoint policies had no effect despite designing Task 3 to favor them. Participant comments consistently stated Task 3 as too difficult ("I found it difficult to keep count of just how many years fit the criteria."), and so we removed it in the final study. Participants also commented that breakpoints were challenging because the rescaling was unexpected when scrubbing the slider, and could disrupt their analysis: ("The scale change was sometimes hard to notice. It is not something you would expect to change so I feel it is easy to miss.") Thus, we added design considerations in the final study.

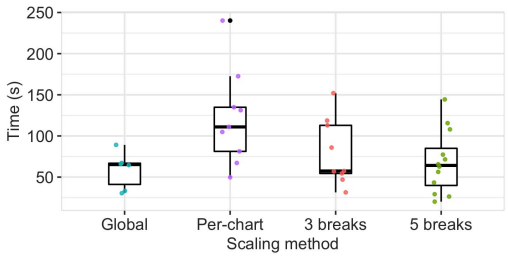


Figure 4: Response latency, Task 1, Second Pilot.

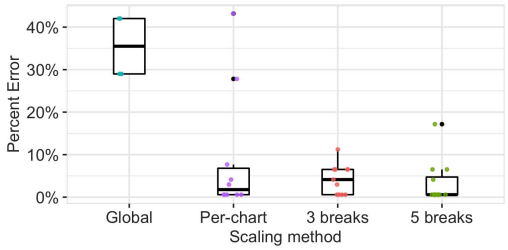
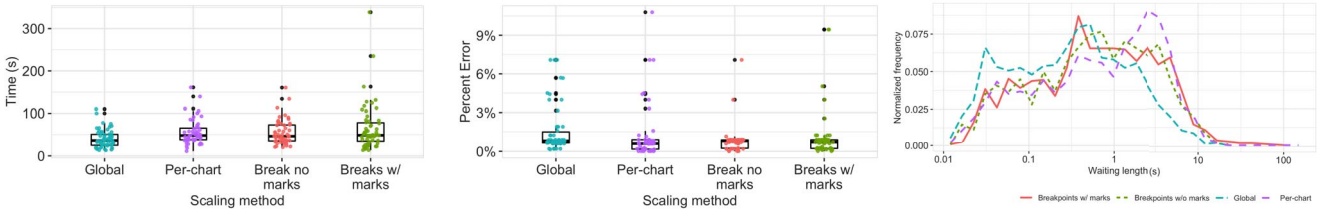


Figure 5: Response accuracy, Task 2, Second Pilot.

2.3 Third Study, With Markers

Based on the prior studies, we ran a final study, now using a within-subjects design. Participants completed 8 tasks in total: they completed Tasks 1 and 2 (we removed Task 3) using the following policies: global, per-chart, 3 breakpoints without markers, and 3 breakpoints with markers, all in a randomized order. Figure 7 shows the markers placed at each breakpoint location between the two adjacent frames. The specific task questions were: (T1) "What was the highest price for October?" and (T2) "What is the difference in prices between May 2005 and June 2005?". Note that we only ask for the highest price in October, rather than the price and year.

Since participants complete the same tasks under multiple policies, using the same data would be subject to learning effects. Thus, we generated synthetic data for 10 years (2000 to 2009) that had similar characteristics as the oil dataset, but where the specific fluctuations were randomly generated. We also updated the qualification tasks so participants were familiarized with each of the scaling methods and asked to answer "At how many positions on the slider, if any, does the y-axis range on the graph change?" To better understand how the policies affected how participants interact with the visualization, we logged interaction traces by saving the timestamp and location of every change to the slider. Finally, we asked participants to simply rank the policies from 1 (easiest) to 4 (hardest to use).



(a) Response latency, Task 1.

(b) Response accuracy, Task 1.

(c) Distribution of waiting lengths, Task 1.

Figure 6: Results from third study

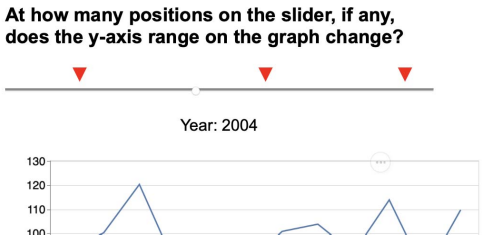


Figure 7: Visual indicators on the slider denoting the locations of breakpoints.

Task	Latency	Error
1	**0.034**	**0.001**
2	0.168	0.530

Table 2: Results of ANOVA tests in the third study.

2.3.1 Results

We collected data from 66 participants, and report one-way within-subjects ANOVA on the 10% trimmed means (Table 2). Both latency (Figure 6a) and error (Figure 6b) were significant in Task 1. Although latency was consistent with the pilots, error was unexpected, as per-chart performed significantly worse than the alternatives. We conjecture that an explanation could be that per-chart requires more effort, participants may have chosen an incorrect year and thus made a mistake. Finally, although we added visual markers for the breakpoints based on participant feedback in the pilots, we did not find a significant effect in performance, however participants ranked visual markers higher on average (2.24) than without markers (3.03).

We also studied whether the rescaling policies affect *how* participants interact with the visualization in Task 1. Specifically, the breakpoint methods would stabilize the y-axes in between the breakpoints, and thus may invite faster scrubbing for analyses like Task 1. To do so, we measure the time between interaction events, and compute the distribution of these wait times (Figure 6c). If participants scrubbed more quickly when the axis scale doesn't change, then we would expect long waiting lengths to be less frequent under the global and breakpoint policies. We in fact see that global has a higher frequency of very short wait lengths, while participants with per-chart spent the greatest portion of their analysis waiting between 1 – 10 seconds. Breakpoints were between the two conditions. We ran pair-wise Kolmogorov–Smirnov tests between the four distributions, and used Bonferroni correction with a factor of 6 (significance threshold is 0.0083). Table 3 shows that all pairs were significant, aside from the two breakpoint policies.

Rescaling Policy 1	Rescaling Policy 2	p-value
Breakpoints w/ marks	Breakpoints no marks	0.397
Breakpoints w/ marks	Global	**0.000**
Breakpoints w/ marks	Per-chart	**0.002**
Breakpoints no marks	Global	**0.000**
Breakpoints no marks	Per-chart	**0.000**
Global	Per-chart	**0.000**

Table 3: Results of Kolmogorov–Smirnov test comparing distribution of interaction wait lengths in third study Task 1.

3 Conclusion and Future Directions

This paper conducted an evaluation of dynamic y-axis rescaling strategies for interactive line chart visualizations, where the user sees a sequence of "frames" in response to user interactions. In addition to commonly used existing strategies that use a fixed y-axis (global) or rescale the y-axis whenever the chart data updates (per-chart), we study a *breakpoints* strategy that rescales the y-axis based on the magnitude that the data changes between successive frames. We find that global and per-chart are respectively well-suited for tasks that compare data across frames and within a single frame, however they perform poorly for the other tasks and depend on how the data's scale changes. Further, *breakpoints* is a robust strategy that performs comparably to the best strategy across tasks.

A promising future direction is an algorithm to automatically determine breakpoint placement. Towards this goal, we have developed such an algorithm motivated by the Resultant Vector algorithm [3, 10] for line chart aspect ratio selection, which can be found as an Observable notebook[1].

Acknowledgments

Support for the research is partially provided by NSF 1564049, 1845638, 2008295, 2106197, 452977, 1939945, 1940175; Amazon and Google research awards, and a Columbia SIRS award.

References

[1] Tableau. https://www.tableau.com, 2020.
[2] W. S. Cleveland. The elements of graphing data. 1985.
[3] S. Guha and W. Cleveland. Perceptual, mathematical, and statistical properties of judging functional dependence on visual displays. Technical report, Technical report, Purdue University Department of Statistics, 2011.
[4] J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5:110–141, 1986.
[5] J. D. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13, 2007.
[6] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics*, 25:438–448, 2019.
[7] Z. Qu and J. Hullman. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 24:468–477, 2018.

[1] https://observablehq.com/d/b1f550d3bba19837

[8] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23:341–350, 2017.
[9] W. D. Stirling. Scale selection and formatting. *Journal of the Royal Statistical Society. Applied Statistics*, 1981.
[10] J. Talbot, J. Gerth, and P. Hanrahan. Arc length-based aspect ratio selection. *IEEE Transactions on Visualization and Computer Graphics*, 17:2276–2282, 2011.
[11] J. Talbot, S. Lin, and P. Hanrahan. An extension of wilkinson's algorithm for positioning tick labels on axes. *IEEE Transactions on Visualization and Computer Graphics*, 16:1036–1043, 2010.
[12] Y. Wang, Z. Wang, L. Zhu, J. Zhang, C.-W. Fu, Z. Cheng, C. Tu, and B. Chen. Is there a robust technique for selecting aspect ratios in line charts? *TVCG*, 2017.
[13] L. Wilkinson. The grammar of graphics. In *Handbook of Computational Statistics*, pp. 375–414. Springer, 2012.
[14] Q. Zhang, H. Zhang, T. Sellam, and E. Wu. Mining precision interfaces from query logs. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 988–1005, 2019.