Journal of Data Science xx (xx), 1–19 DOI: xx.xxxx/xxxxxxxx
July 2020

'You Draw It': Implementation of visually fitted trends with `r2d3`

Abstract

How do statistical regression results compare to intuitive, visually fitted results? Fitting lines by eye through a set of points has been explored since the 20th century. Common methods of fitting trends by eye involve maneuvering a string, black thread, or ruler until the fit is suitable, then drawing the line through the set of points. In 2015, the New York Times introduced an interactive feature, called 'You Draw It,' where readers are asked to input their own assumptions about various metrics and compare how these assumptions relate to reality. This research is intended to implement 'You Draw It', adapted from the New York Times, as a way to measure the patterns we see in data.

Introduction

We all use statistical graphics, but how do we know that the graphics we use are communicating effectively? Through experimentation, graphical testing methods allow researchers to conduct studies designed to understand how we perceive graphics and perform graphical tasks such as differentiation, prediction, estimation, and extrapolation. Each of these levels of interaction with a graph require a different method of engagement with and use of the information presented in a chart. In this paper, we describe the adaptation of an old tool for graphical testing and evaluation, eye-fitting, for use in modern web-applications suitable for testing statistical graphics. We present an empirical evaluation of this testing method for linear regression, and briefly discuss an extension of this method to non-linear applications.

One of the most common charts created is a scatterplot of points over time; these charts

show up regularly in news articles and in scientific publications alike. These charts rely on our ability to identify and detect trends in data. Our visual system is naturally built to look for structure and identify patterns, including patterns and trends over time; many times we do not even notice this process happening subconsciously. As shown in Figure 1, a viewer engaging with a plot of weekly average gas prices over time may perform several cognitive operations. First, they scan the plot and assesses points to determine if there are any outliers or otherwise remarkable points. Then, they may fit a rough mental smooth/trend line to the points to summarize the useful information and remove variability. Finally, an interested and engaged viewer may pull in additional contextual information from long-term memory, seeking to explain variation in the mental 'trend' with supplemental information, such as COVID lock downs which reduced gasoline demand and the beginning of the war in Ukraine, which wreaked havoc on the supply and demand for global energy sources.

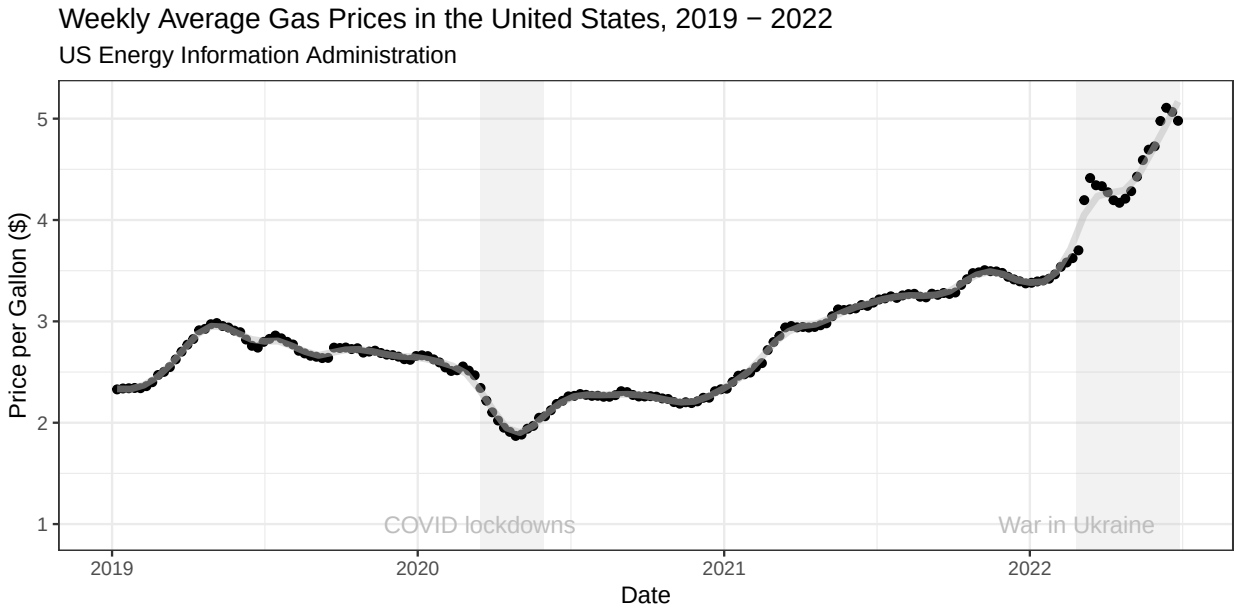

Figure 1: Weekly average gas prices in the United States, 2019-June 2022. Additional mental operations a viewer might perform while looking at the plot are annotated in grey (US Energy Information Administration 2022).

Initial studies in the 20th century explored the use of fitting lines by eye through a set of points (Finney 1951; Mosteller et al. 1981; Unwin and Wills 1988). Common methods

of fitting trends by eye involved maneuvering a string, black thread, or ruler until the fit was suitable, then drawing the line through the set of points. Results from these early studies provided groundwork for visually judging slopes and selecting accurate starting values for common iterative calculations. Recently, Ciccione and Dehaene (2021) conducted a comprehensive set of studies investigating human ability to detect trends in graphical representations from a psychophysical approach. This set of studies asked participants to judge trends, estimate slopes, and conduct extrapolation by using a track-pad to adjust the tilt of a line on the screen. Results indicated the slopes participants reported were always in excess of the ideal slopes, both in the positive and in the negative direction, and those biases increased with noise and with number of points. This supported the results found in Mosteller et al. (1981) and suggested that participants might use Deming regression (Deming 1943), which minimizes the Euclidean distance of points from the line, when fitting a line to a noisy scatterplot.

While psychologists and statisticians have been using eye-fitting techniques to assess our innate perceptual statistical modeling abilities, news organizations have used similar techniques in order to draw readers in and demonstrate the difference between readers' assumptions and reality. In 2015, the New York Times (NYT) introduced an interactive feature, called 'You Draw It' (Aisch, Cox, and Quealy 2015), where readers input their own assumptions about various metrics of news interest and compare these assumptions to reality. The NYT team used Data Driven Documents (D3), a JavaScript library that allows readers to interact with a chart directly by drawing a line on their computer screen with a mouse. Despite the somewhat different purpose behind this feature, the D3 driven method used by the NYT is wonderfully intuitive, and does not require the assumption of linearity, making it much more adaptable to testing how viewers perceive and predict when presented with non-linear data.

We set out to implement 'You Draw It', adapted from the NYT feature, as a way to ex-

perimentally assess the patterns we see in data. Here, we provide technical details of the software development, utilizing interactive graphics in R (R Core Team 2022). We then share results from our study which validates 'You Draw It' as a method for graphical testing on linear trends and apply an appropriate data analysis method to the participant data. We also briefly demonstrate the use of the 'You Draw It' method and analysis on nonlinear data.

Development

'You Draw It' v0

The NYT uses D3, Data Driven Documents (Bostock, Ogievetsky, and Heer 2011), to create many of the interactive data graphics that users rely on, from the famous "election needle" in 2016 to their COVID dashboards. Naturally, this same framework was used to create the 'You Draw It' feature. The NYT code used to create the initial D3 plot and 'You Draw It' interactive feature includes multiple working parts. First, the relationship between coordinate mappings and pixels sets up the chart and the initial data is plotted. Next, the `d3.drag` function adds user interactivity and observes the movement of the user's mouse. Additionally, the `forEach` loop finds the closest point in the coordinate system to the user's mouse. Finally, the code detects if all points are filled by the user and updates the status of the task.

When working with a new programming language, setting up a new development environment can be challenging. As R users, we found the software overhead required to develop straight JavaScript code to be intimidating (Wattenberger 2019), but luckily, the R community anticipated this barrier. In the next section, we will see how the `r2d3` package (Strayer, Luraschi, and Allaire 2020) in R can be used to render the JavaScript code and display D3 visuals in familiar R HTML formats such as Rmarkdown and Shiny.

Generating D3 Plots in R

We leveraged the `r2d3` package to take data randomly generated in R and create a D3 plot that could be modified with the 'You Draw It' script. This step allowed us to more easily connect the data generation process with the resulting D3 code without having to generate new code manually each time we re-generated the data. While not strictly necessary for the validation test of the 'You Draw It' method compared to past linear regression methods presented in this paper, this abstraction makes it much easier to test arbitrary or model-generated data which is unique to each participant.

We conducted all data simulation and processing in R and output two data sets - *point data* and *line data* - containing (x, y) coordinates corresponding to either a simulated point or fitted value predicted by a statistical model respectively. Then, the `r2d3` package converts the data sets in R to JavaScript Object Notation (JSON) to be interpreted by JavaScript code included with the `r2d3` package. Parameters for aesthetic design choices are defined in a list of options in the `r2d3` function call; `r2d3` passes these to the generated JavaScript code. For instance, we can specify the buffer space allowed for the x and y axes to avoid users anchoring their lines to the axes limits.

Adding 'You Draw It' Functionality

In order to make use of the original 'You Draw It' code for perceptual experiments, we first needed to accommodate an additional layer. The original NYT features asked participants to draw on a blank coordinate grid, as shown in Figure 2.

When testing perception of graphics, however, we want to pre-populate the chart with additional information - points, and in some cases, portions of a trend line. This requires that we modify the original JavaScript source code to accommodate these additional elements, which we generate using `r2d3`, as described above.

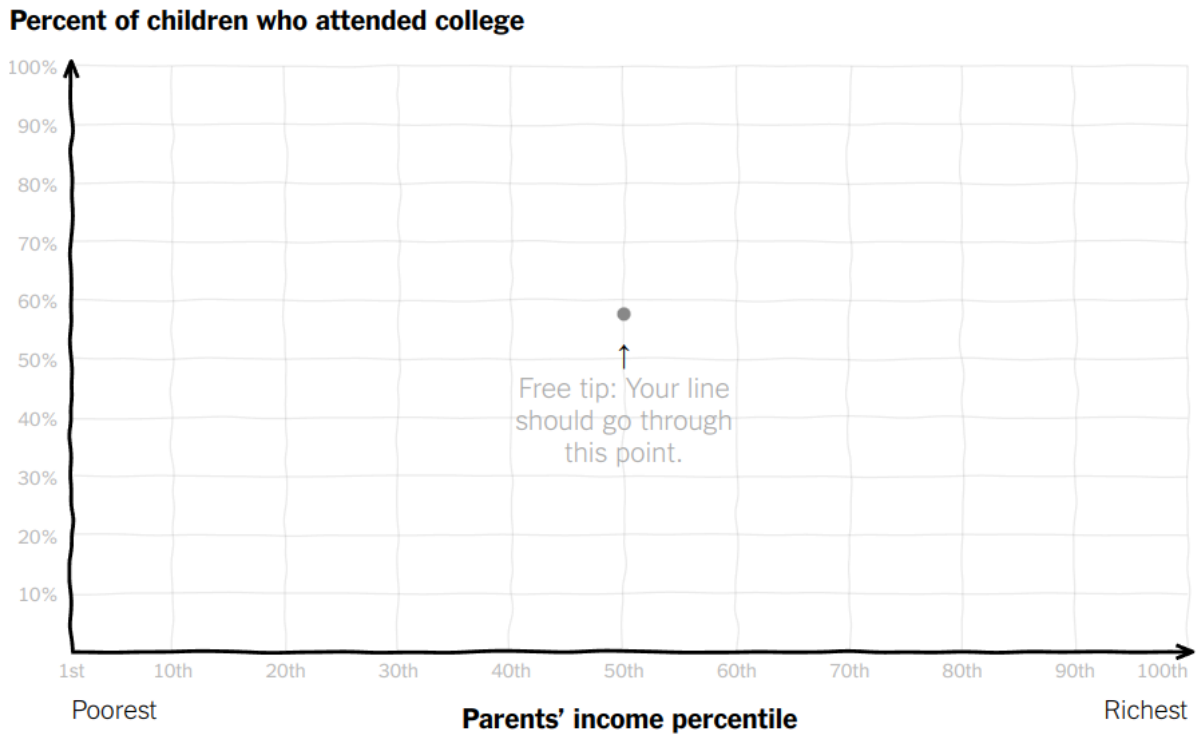

Figure 2: Screenshot of the You Draw It application originally developed by the New York Times.

As previously discussed, we modified `r2d3` code to draw the initial plot and data. Drag events in `D3.js` are utilized to observe and react to user input, as in the original 'You Draw It' script from the NYT. [1] We adapted initial code from Pearce (2019) and Furic (2017).

One constraint we inherited from the NYT code is that users can only draw one-to-one functions; because the code works with drag events, each point in x can correspond to only one point in y, at least as the code is currently written. While this is not particularly problematic for our applications to date, it might limit applications of this type of user interaction when assessing user drawn confidence bands, ribbons, and other situations where two or more vertical points are required.

Visual Cues

During testing of our modifications to the 'You Draw It' script, we discovered that frequently the JavaScript code would "skip" from point to point out of sequence. This resulted in a jagged line (visually) with missing values in the underlying stored array of data; as a result, the for-loop controlling the user-drawn line never exited and the user's data was not recorded. While it would have been possible to fix this using some sort of interpolation algorithm, we did not want to compromise user results by introducing interpolation artifacts, so we opted instead for a visual cue that fixed the problem by modifying the human behavior.

As shown in Figure 3, the user-filled portion of the plot is represented with a yellow rectangle, which adapts to the user's input and spans any missing values. When the box disappears, the array is filled in and the user can submit their response.

One challenge introduced by adding this visual cue is that D3 uses Scalable Vector Graphics (SVG) to render elements. Adding an additional layer meant that we had to ensure that not only the grids, lines, and points of the default scatterplot and trend line rendered in the

[1] The full source code we developed for the 'You Draw It' task is available at https://github.com/earobinson95/presentations/blob/master/can-you-draw-it/www/you-draw-it.js.

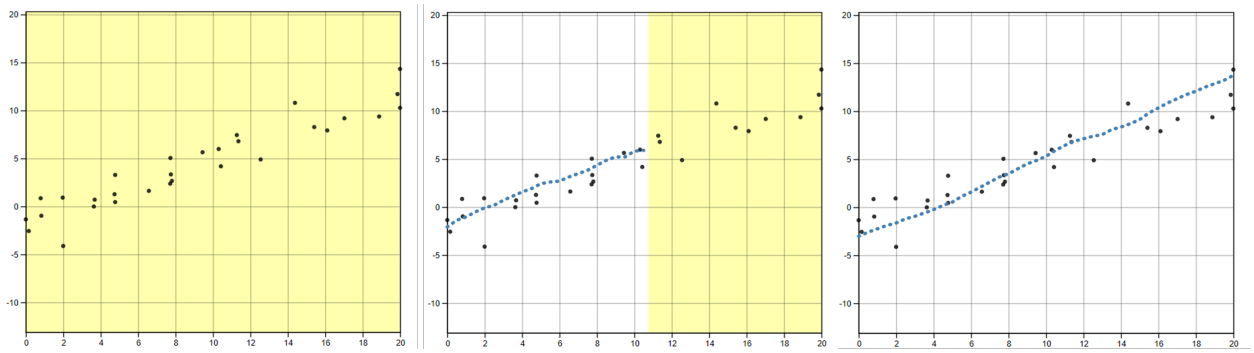

Figure 3: You Draw It task plot as shown to user.
left: Initial state, with instructions *Use your mouse to fill in the trend in the yellow box region.*
middle: User view during task completion.
right: Finished state

correct order, but that the yellow box rendered behind all of these points as well so that no information was masked; and that this layer order updated in real time.

Connecting to Shiny

Shiny Messages are used to communicate the user interaction between the JavaScript code and the R environment. The initial plot is rendered using Shiny's `RenderD3` and `d3Output` functions; subsequent user interactions are controlled via the JavaScript code. Once the user is finished modifying the plot, they can submit their response, so long as they have filled in all of the points. This is enforced using Shiny's message-passing interface and a JavaScript hook that notifies Shiny when the array of user-drawn points is completely filled in. Once the user is done drawing the line, we save the results of the drawn line to a SQLite database, using Shiny to pass data from JavaScript to R. The connections between each portion of the app are shown in Figure 4.

One additional challenge when integrating the D3 graphics into the Shiny application is that by default, most Shiny applications use a reactive framework that adjusts to the user's browser size. When working with D3, however, this can be problematic: most D3 parameters are specified at the pixel level. The discontinuity between the assumptions of Shiny and D3

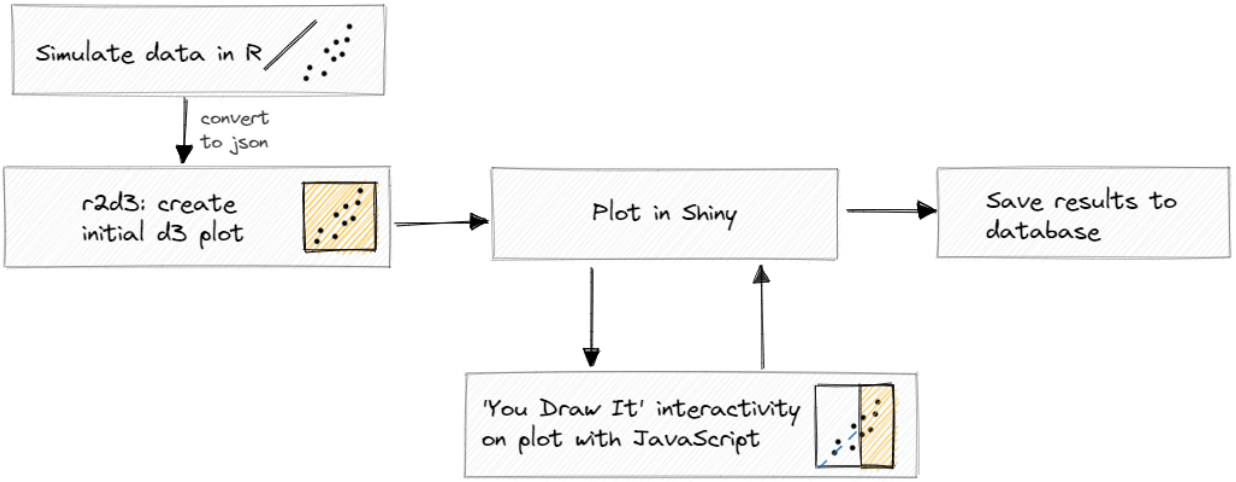

Figure 4: Sketch of underlying code for 'You Draw It', illustrating the data simulation conducted in R, the initial setup of the visual stimuli with D3 source code, along with the iterative process between the user interaction and plotting in Shiny. Once the user is done drawing the line, we saved the results of the drawn line to a database for analysis.

meant that we had to fix the size of the Shiny element, and then perform a check to ensure that the user's screen was sufficiently big to render the plot. While in an ideal world, users could participate using cell phones, tablets, and traditional laptop/desktop computers, functionally our checks limited users' ability to participate using smaller screen sizes found in cell phones and some tablets. Additionally, after some feedback by laptop users during pilot testing, we included an additional requirement that participants have a computer mouse available; the results from touchpad users were qualitatively different (more jagged) in ways suggesting that the recorded data did not adequately describe users' perceptions.

Application

Validation Study

We conducted a study in order to validate 'You Draw It' as a method for graphical testing, comparing results to the less technological method utilized in Mosteller et al. (1981). The original study asked 153 graduate students and postdoctoral researchers in an Introductory

Biostatistics course to fit lines by eye to a set of four points (S, F, V, N), each set with differing slope and variance properties, using an 8.5 x 11 inch transparency with a straight line etched across the middle. Researchers conducted the study with a latin square experimental design to test for a practice effect on the task. Qualitative analysis suggested that participants tended to fit the slope of the first principal component (minimizes Euclidean distance) as opposed to the slope from the ordinary least squares regression (minimizes vertical distance) Figure 5 and found no effect of order.

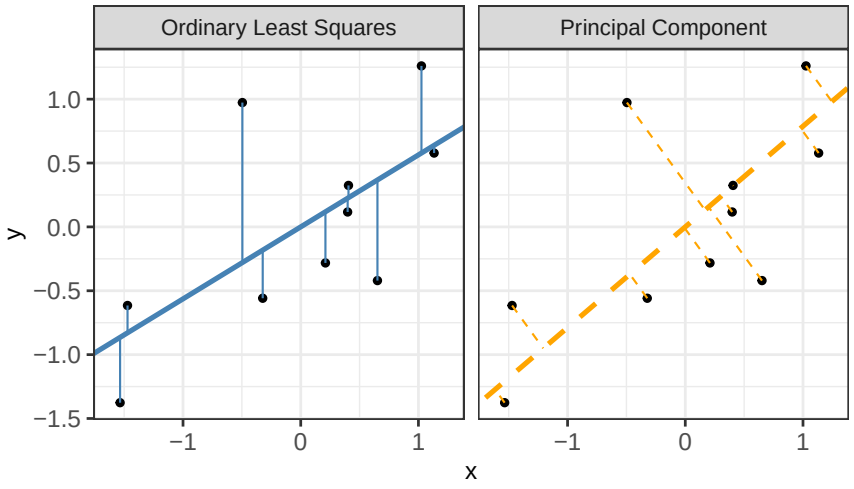

Figure 5: Comparison between an OLS regression equation which minimizes the vertical distance of points from the line and a regression equation with a slope calculated by the first principal component which minimizes the smallest distance of points from the line.

In our study, we replicated Mosteller et al. (1981) using the 'You Draw It' method and simulated data with parameter coefficients selected to reflect those from the four data sets in the original study. Data were simulated based on a linear model with additive errors:

$$y_i = \beta_0 + \beta_1 x_i + e_i \tag{1}$$

$$\text{with } e_i \sim N(0, \sigma^2).$$

For each participant, four unique data sets were randomly and independently generated from

the underlying parameters at the beginning of the study and mapped to a scatterplot graphic. When participants started the study, they were first asked to complete two practice plots - accompanied by instructions and a .gif demonstrating the task - in order to train them in the skills associated with executing the task. The four 'You Draw It' task plots associated with the selected parameters followed the practice plots in a random order for each individual; these plots were interspersed with plots from a different experiment.

Participants were recruited via Prolific in March 2022; while this study does utilize a convenience sample, as this is primarily a perceptual task, previous results have found few differences between expert and non-expert participants in this context (VanderPlas and Hofmann 2015). The study was conducted and distributed via a Shiny application and participants completed the tasks on their own computer, with a computer mouse, and in an environment of their choosing. The data from this study were collected to validate this method of graphical testing, with the hopes of providing a new tool to assess graphical perception interactively.

Data Analysis and Results

As a part of the validation study, we collected data for each participant drawn trend line, allowing us to conduct formal analyses which compared the intuitive visually fit lines to statistical regression results - the original study lacked these formal comparisons, providing only summaries of averages, variances, and actual (least squares) values for the slope and intercept of each data set. The feedback data, stored in a SQLite database, contained the simulated data points} - (x_{ijk}, y_{ijk}) -, the predicted values from the statistical regression models - $\hat{y}_{ijk,OLS}$, and $\hat{y}_{ijk,PCA}$ -, and the predicted values from the user drawn line - $y_{ijk,drawn}$ for parameter choice $i = 1, 2, 3, 4$, $j = 1, ..., N_{\text{participant}}$, and x_{ijk} value $k = 1, ..., 4x_{max} + 1$. Figure 6 displays an example of all three fitted trend lines for parameter choice F.

A unique data set was simulated independently for each participant, therefore, comparisons

of vertical residuals between the user drawn line and the OLS fitted values ($e_{ijk,OLS} = y_{ijk,drawn} - \hat{y}_{ijk,OLS}$) and PCA fitted values ($e_{ijk,PCA} = y_{ijk,drawn} - \hat{y}_{ijk,PCA}$) were used to assess the accuracy of participant drawn lines. We use mixed model methods to analyze residual trends and capture the variability due to participant differences.

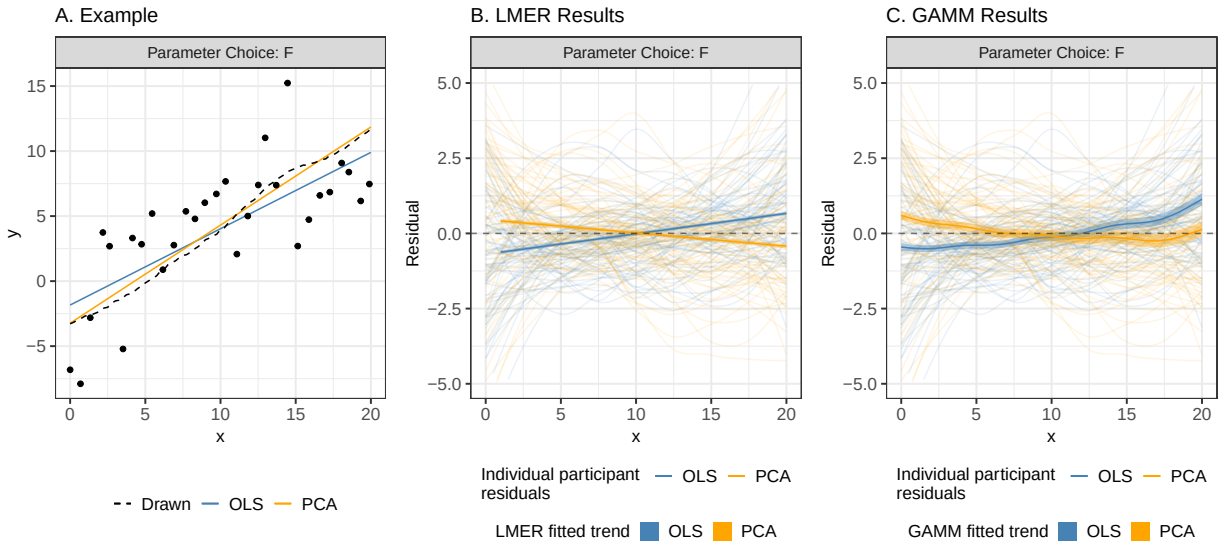

Figure 6: Visual analysis results for parameter choice set F. **A.** Example of validation feedback data where three trend lines show the the OLS fitted, PCA fitted, and participant drawn values overlaid on the simulated data points. **B & C.** Estimated trends of residuals (vertical deviation of participant drawn points from both the OLS (blue) and PCA (orange) fitted points) as fit by the (B) linear mixed model and (C) generalized additive mixed model. A random sample of 75 participants was selected to display the individual participant residuals behind the overall trend.

We first fit a linear mixed model (LMM) to the OLS and PCA residuals separately using the `lmer` function in the `lme4` package (Bates et al. 2015), constraining the fit to a linear trend. Parameter choice, x, and the interaction between x and parameter choice were treated as fixed effects with a random participant effect accounting for variation due to participant. The LMM equation for each fit (OLS and PCA) residuals is given by:

$$y_{ijk,drawn} - \hat{y}_{ijk,fit} = e_{ijk,fit} = [\gamma_0 + \alpha_i] + [\gamma_1 x_{ijk} + \gamma_{2i} x_{ijk}] + p_j + \epsilon_{ijk} \tag{2}$$

where

- $y_{ijk,drawn}$ is the drawn y-value for the i^{th} parameter choice, j^{th} participant, and k^{th} increment of x-value
- $\hat{y}_{ijk,fit}$ is the fitted y-value for the i^{th} parameter choice, j^{th} participant, and k^{th} increment of x-value corresponding to either the OLS or PCA fit
- $e_{ijk,fit}$ is the residual between the drawn and fitted y-values for the i^{th} parameter choice, j^{th} participant, and k^{th} increment of x-value corresponding to either the OLS or PCA fit
- γ_0 is the overall intercept
- α_i is the effect of the i^{th} parameter choice (F, S, V, N) on the intercept
- γ_1 is the overall slope for x
- γ_{2i} is the effect of the parameter choice on the slope
- x_{ijk} is the x-value for the i^{th} parameter choice, j^{th} participant, and k^{th} increment
- $p_j \sim N(0, \sigma^2_{\text{participant}})$ is the random error due to the j^{th} participant's characteristics
- $\epsilon_{ijk} \sim N(0, \sigma^2)$ is the residual error.

Eliminating the linear trend constraint, we can extend the method and analysis beyond ordinary least squares by using generalized additive mixed models (GAMM) to estimate smoothing splines and allow for a more flexible residual trend. The `bam` function in the `mgcv` (Wood 2017) package was used to fit a GAMM separately to the OLS and PCA. Parameter choice was treated as a fixed effect with no estimated intercept and a separate smoothing spline for x was estimated for each parameter choice. A random participant effect accounting for variation due to participant and a random spline for each participant accounted for variation in spline for each participant. The GAMM equation for each fit (OLS and PCA) residuals is given by:

$$y_{ijk,drawn} - \hat{y}_{ijk,fit} = e_{ijk,fit} = \alpha_i + s_i(x_{ijk}) + p_j + s_j(x_{ijk}) \tag{3}$$

where

- $y_{ijk,drawn}$ is the drawn y-value for the i^{th} parameter choice, j^{th} participant, and k^{th} increment of x-value
- $\hat{y}_{ijk,fit}$ is the fitted y-value for the i^{th} parameter choice, j^{th} participant, and k^{th} increment of x-value corresponding to either the OLS or PCA fit
- $e_{ijk,fit}$ is the residual between the drawn and fitted y-values for the i^{th} parameter choice, j^{th} participant, and k^{th} increment of x-value corresponding to either the OLS or PCA fit
- α_i is the intercept for the parameter choice i
- s_i is the smoothing spline for the i^{th} parameter choice
- x_{ijk} is the x-value for the i^{th} parameter choice, j^{th} participant, and k^{th} increment
- $p_j \sim N(0, \sigma^2_{\text{participant}})$ is the error due to participant variation
- s_j is the random smoothing spline for each participant.

This method allows us to quantitatively compare participant lines to the OLS and principal component regression lines, which is an improvement over the qualitative approach in Mosteller et al. (1981). While the statistical models were fit to all four parameter choices, Figure 6 displays the estimated trends of residuals (vertical deviation of participant drawn points from both the OLS and PCA fitted points) as modeled by a LMM and GAMM respectively for parameter choice set F. A random sample of 75 participants was selected to display individual participant residuals behind the overall residual trend. Examining the plots, the estimated trends of PCA residuals (orange) appear to align more parallel and closer to the $y = 0$ horizontal (dashed) line than the OLS residuals (blue). In particular, this trend was more prominent in parameter choices with large variances. Results from our study were consistent with those found in the original study; when shown points following a linear trend, participants tended to fit the slope of the first principal component Our study established 'You Draw It' as a method for graphical testing and reinforced the differences

between intuitive visual model fitting and statistical model fitting, providing information about human perception as it relates to the use of statistical graphics.

Extension to Nonlinear Data

The analysis method presented above extends nicely beyond the linear regressions tested in Mosteller et al. (1981). Here we briefly demonstrate this with an example from a study examining participants ability to make forecasts for exponentially increasing data on log and linear scales. Along with analyzing the feedback data with the GAMM method for flexibility due to nonlinear data, we used spaghetti plots to conduct a visual analysis of participant forecasts compared to the nonlinear least squares statistical model and make comparisons between two chart design features Figure 7. The combination of the GAMM analysis and visual display demonstrates the strength of the 'You Draw It' method for testing statistical graphics.

Conclusion

The data presented in this paper are part of a much broader experiment on the perception of log scales; while this paper focuses primarily on the computational implementation of the 'You Draw It' method and its adaptation to testing graphics, we are currently finishing the analysis and description of the broader experiment, which uses this 'You Draw It' method to assess user prediction of exponential trends.

By introducing an interactive method for assessing eye-fit data summaries, along with an analysis method which allows us to test competing hypotheses to determine whether user responses are similar to particular statistical models, we have provided another tool for experimentally testing statistical charts. In the future, we hope to create an R package to more easily facilitate these types of user experiments, making this technique available to other researchers who may not be willing to tinker with JavaScript directly.

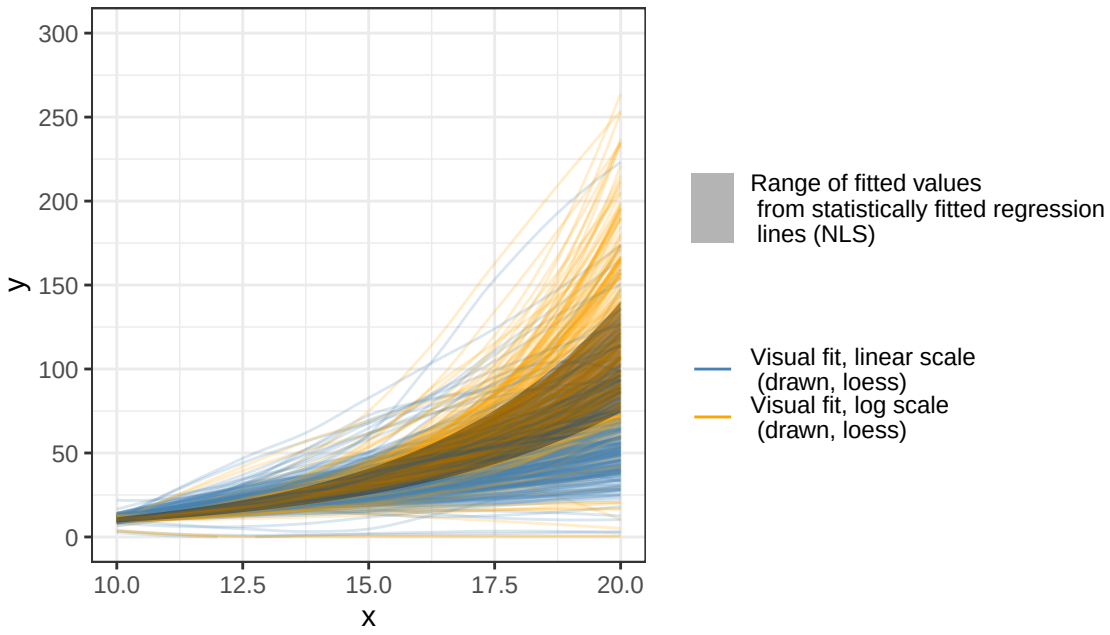

Figure 7: Spaghetti plot of results from a study which asked participants to forecast trends of exponentially increasing data. Participants drawn lines on the linear scale are shown in blue and the log scale are shown in orange. Variability in the statistically fitted regression lines occurred due to a unique data set being simulated for each individual; the gray band shows the range fitted values from the statistically fitted regression lines.

As with any method for testing statistical graphics, there are a host of additional studies which would be useful to understand how users react to the 'You Draw It' method and what parameters are most important for the researcher to control. One avenue of future exploration is to investigate the effect of axis limits on user anchoring. We know that the perceptual experience is heavily affected by anchoring to e.g. axis breaks, and we would expect that a similar anchoring effect might be present based on the limits of the plot and the amount of space in x and y provided for the user to draw.

Supplementary Material

- **'You Draw It' Demonstration Applet:** The shiny app used to demonstrate the 'You Draw It' method can be accessed at emily-robinson.shinyapps.io/you-draw-it-validation-applet.
- **'You Draw It' Script:** The 'You Draw It' JavaScript code can be accessed at github.com/earobinson95/presentations/blob/master/can-you-draw-it/www/you-draw-it.js.
- **Study Applet:** The shiny app used to conduct the study can be accessed at shiny.srvanderplas.com/perception-of-statistical-graphics/.
- **Study Applet Code:** The code used to create the RShiny Applet for data collection can be found at github.com/earobinson95/log-perception-prolific/tree/main/perception-of-statistical-graphics.
- **Participant Data (Linear):** De-identified participant data collected in the study and used for analyses are available to be downloaded from GitHub at github.com/earobinson95/sdss-2022-you-draw-it-manuscript/raw/master/data/eyefitting-model-data.csv.
- **Participant Data (Nonlinear):** De-identified participant data collected in the study and used for analyses are available to be downloaded from GitHub at

github.com/earobinson95/sdss-2022-you-draw-it-manuscript/raw/master/data/youdrawit-model-data.csv.

- **Data Analysis Code:** The code used to replicate the nonlinear study analysis in this paper can be found at earobinson95.github.io/sdss-2022-you-draw-it-manuscript/analysis/data-analysis.htmll.

References

Aisch, Gregor, Amanda Cox, and Kevin Quealy. 2015. "You Draw It: How Family Income Predicts Children's College Chances." *The New York Times* 28.

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48. https://doi.org/10.18637/jss.v067.i01.

Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. 2011. "D^3 Data-Driven Documents." *IEEE Transactions on Visualization and Computer Graphics* 17 (12): 2301–9.

Ciccione, Lorenzo, and Stanislas Dehaene. 2021. "Can Humans Perform Mental Regression on a Graph? Accuracy and Bias in the Perception of Scatterplots." *Cognitive Psychology* 128: 101406.

Deming, William Edwards. 1943. "Statistical Adjustment of Data."

Finney, DJ. 1951. "Subjective Judgment in Statistical Analysis: An Experimental Study." *Journal of the Royal Statistical Society: Series B (Methodological)* 13 (2): 284–97.

Furic, Benoit. 2017. "You-Draw-It-Graph: The User Can Draw His Own Graph Line and Compare with the Reality." *GitHub*. https://github.com/BenoitFuric/you-draw-it-graph.

Mosteller, Frederick, Andrew F Siegel, Edward Trapido, and Cleo Youtz. 1981. "Eye Fitting Straight Lines." *The American Statistician* 35 (3): 150–52.

Pearce, Adam. 2019. "You-Draw-It." *Observable*. https://bl.ocks.org/1wheel/07d9040c3422dac16bd5be741433ff1e.

R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Strayer, Nick, Javier Luraschi, and JJ Allaire. 2020. *R2d3: Interface to 'D3' Visualizations*. https://CRAN.R-project.org/package=r2d3.

Unwin, Antony R, and Graham Wills. 1988. "Eyeballing Time Series." In *Proceedings of the 1988 ASA Statistical Computing Section*, 263–68.

US Energy Information Administration. 2022. "Weekly U.S. All Grades All Formulations Retail Gasoline Prices (Dollars Per Gallon)." *US Energy Information Administration Independent Statistics & Analysis*. https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=PET&s=EMM_EPM0_PTE_NUS_DPG&f=W.

VanderPlas, Susan, and Heike Hofmann. 2015. "Spatial Reasoning and Data Displays." *IEEE Transactions on Visualization and Computer Graphics* 22 (1): 459–68.

Wattenberger, Amelia. 2019. *Fullstack D3 and Data Visualization: Build Beautiful Data Visualizations and Dashboards with D3*. Fullstack.io.

Wood, S. N. 2017. *Generalized Additive Models: An Introduction with r*. 2nd ed. Chapman; Hall/CRC.