Title
Estimating the growth of functions

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 46(0)

Authors
Marupudi, Vijay
Bye, Jeffrey K.
Varma, Sashank

Publication Date
2024

Peer reviewed

Estimating the growth of functions

Vijay Marupudi (vijaymarupudi@gatech.edu)
School of Interactive Computing,
Georgia Tech
Atlanta, GA 30308, USA

Jeffrey Kramer Bye (jbye@umn.edu)
Department of Educational Psychology,
University of Minnesota,
Minneapolis, MN 55455, USA

Sashank Varma (varma@gatech.edu)
School of Interactive Computing,
Georgia Tech,
Atlanta, GA 30308, USA

Abstract

An important aspect of mathematical and computational thinking is algorithmic thinking—the analysis of systems, algorithms, and natural processes. A fundamental skill in algorithmic thinking is estimating the growth of functions with increasing input size. In this study, we asked 178 participants to estimate values of seven common functions in algorithmic analysis [$\log(n)$, $\sqrt{n}$, $n\log(n)$, n^2, n^3, 2^n, $n!$] to understand their intuitive perception of their growth. Their estimates were fit against the actual values for all functions. Participants showed a linearization bias: sublinear functions were best fit by a linear function, and superlinear functions were best fit by a cubic (i.e., polynomial) function, even those that grow much faster (e.g., $n!$). In addition, participants estimated logarithmic functions least accurately. These results provide insight into how people perceive the growth of functions and set the stage for future studies of how to best improve people's reasoning about functions more generally.

Keywords: mathematical cognition; algorithmic thinking; function learning; exponential growth; logarithmic growth

Introduction

An important part of mathematical and computational thinking is acquiring an intuitive sense of common functions such as $\log(n)$, $\sqrt{n}$, $n\log(n)$, n^2, n^3, 2^n and $n!$. This involves developing an accurate perception of the magnitudes of expressions using these functions and understanding the general shape of these functions. However, people have been shown to struggle with many of these common functions. Research on how students analyze the computational complexity of algorithms in the classroom has suggested that poor understanding of the functions themselves, especially logarithms, might underlie the difficulty of algorithmic analysis (Parker & Lewis, 2014). While students' intuitions about superlinear functions (e.g., exponentials) may be improved by experience with intractable problems (del Vado Vírseda, 2021) and concrete examples like the Tower of Hanoi problem (del Vado Vírseda, 2019; Levitin, 2005), novices in particular still struggle with determining whether a problem of a given level of complexity can be solved in a feasible amount of time (MacCormick, 2018).

These common functions appear in many other sciences, where they have similarly been found to be difficult to teach and intuit. Undergraduate life sciences majors are often unable to identify a logarithmic function from a graph (Confrey & Smith, 1995), despite having taken calculus. Middle school teachers who teach exponential functions themselves have misconceptions about the difference between linear and exponential growth trends (Alagic & Palenz, 2006). Students' early misconceptions about exponential growth often come from misapplying the constant unit rate interpretation of slope from linear functions; they show little evidence of the 'covariational reasoning' that allows for modeling multiplicative relations (Confrey & Smith, 1995; Ferrari-Escolá et al., 2016). Students' understanding of exponential growth has been characterized as a learning trajectory (Ellis et al., 2016), with students gradually moving from conceptualizing exponential growth as a qualitatively superlinear curve, to natural number multiplications (e.g., "the number of cases is doubled five times"), to a correspondence between multiplicative (exponential) growth in the outcome and additive (linear) growth in time (e.g., "over five days, the number of cases has increased tenfold").

Research in psychology has also found that people perform poorly when reasoning about superlinear functions, especially the fastest growing ones. Tversky and Kahneman (1973) investigated the factorial function $n!$, which grows even faster than the exponential function. They found that people's estimate of 8! were an order of magnitude smaller than the actual value. Wagenaar and Sagaria (1975) had college undergraduates view a table of hypothetical pollution levels for the years 1970-1974 that had been generated by an exponential function, and then estimate the pollution level in 1979. Their estimates were also an order of magnitude less than the correct value.

Another line of research in cognitive psychology has shown that people find it easier to learn, from training data, functions that are linear, continuous, monotonic, and noncyclic compared to the alternatives (Busemeyer et al., 1997). Participants learning linear, quadratic, exponential, and periodic functions were better at function interpolation than extrapolation, with increased error the farther a new input value is from previously sampled points (Gelpi et al., 2021). The Population of Linear Experts theory (Kalish et al., 2004) posits that errors in extrapolation occur because humans encode different intervals of input values as following linear functions with different slopes. In these experiments on function learning, participants are unaware of the shape of the functions before prediction, and have to learn them from feedback. However, there are important differences between learning functions and estimating their growth. For example,

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

in mathematics, computer science, and natural science contexts, people are often aware of the definition of the function at hand (e.g., that pandemics explode at an exponential rate), and do not have to learn them.

Research on irrational numbers supports the hypothesis that humans are linear approximators. Patel and Varma (2018) asked participants to compare pairs of $\sqrt{n}$ expressions and found behavioral patterns similar to people's comparisons of natural numbers, i.e., the linear function. On the other hand, when asked to interpolate graphically, people have been shown to employ a quadratic curve to estimate noisy exponential curves (Ciccione et al., 2022). More generally, surprisingly few studies have investigated how polynomials are processed, despite their prevalence in high school mathematics. Research on people's perception of other important functions, such as $\log(n)$ and $n!$, is also limited. Many of these studies focus on the phenomenon of underestimation rather than people's intuitions about the profile, or *shape*, of different functions. In addition, very few studies have investigated how people perceive the symbolic forms of these functions or compared how people perceive the growth of different functions, despite their importance in mathematics, natural science, computer science, and science communication.

To understand how people perceive the growth of functions, we asked participants to make speeded estimations of the values of seven functions common in mathematics and computer science: $\log(n)$, $\sqrt{n}$, $n\log(n)$, n^2, n^3, 2^n, $n!$. They did so for input values appropriate for each function to encourage estimation instead of calculation, in contrast to previous work where all functions were displayed with the same input values (Marupudi et al., 2022). We fit participants' estimates for each function to the actual values of all seven functions (as well as the linear function, n). We predicted that their estimates would be linearized, i.e., functions would be best fit by a function that is closer to a linear growth profile (shape). For example, we predicted 2^n would be best fit by a polynomial function such as n^2 or n^3 (i.e., closer to linear n), and $\log(n)$ would be best fit by a linear function. Finally, given past research, we also predicted participants would perform least accurately for functions involving logarithms (Parker & Lewis, 2014).

Methods

Participants

We recruited 301 participants from the online recruitment platform Prolific (www.prolific.com). The study was made available for participants between 18 and 65 years old. Data for 128 participants was removed due to data quality concerns (see below), resulting in a final dataset of 178 participants. The mean age of participants was 38.47 years old (SD = 11.18). 103 participants identified as men, 69 as women, and 4 as non-binary or gender non-conforming. This task was part of a larger study for which participants received $12 for 45 minutes of their time. The experimental protocol was approved by the university IRB.

Design and Materials

The study followed a 7 (function, within-participants) x 9 (input value, within-participants) x 2 (order of trials within functions, between-participants) design. Each trial consisted of one functional expression with one input value for participants to estimate the value of. We selected seven common functions important for mathematics and computer science: $\log(n)$, $\sqrt{n}$, $n\log(n)$, n^2, n^3, 2^n, $n!$. The range of the input values for each function was varied to prevent participants from using calculation strategies. The input values were between 1 and 100000 for $\log n$, 1 and 10000 for $\sqrt{n}$, 1 and 1000 for $n\log n$, and 1 and 1000 for n^2, 1 and 100 for n^3, 1 and 20 for 2^n, and 1 and 10 for $n!$. To determine the input values for each function, for each participant, nine base values were chosen to evenly span the range, and a random number up to 5% of the upper bound for each function was added to each value to prevent the use of special strategies for specific input numbers (e.g., 503^2 instead of 500^2).

The dependent variable for each trial was the participant's estimate for the functional expression. We also collected their response time (RT) to make each estimate; these data are not analyzed here due to space constraints.

Procedure

Participants completed the experiment online after a different math task. They were instructed to try their best to estimate the values of the mathematical expressions, up to 2 decimal points. Examples of all seven functions (without answers) were provided to familiarize them with the functions that would be presented to them. Additional instructions was presented for logarithms and factorials because pilot testing revealed that some participants needed to be reminded of the syntax of these functions and their definitions.

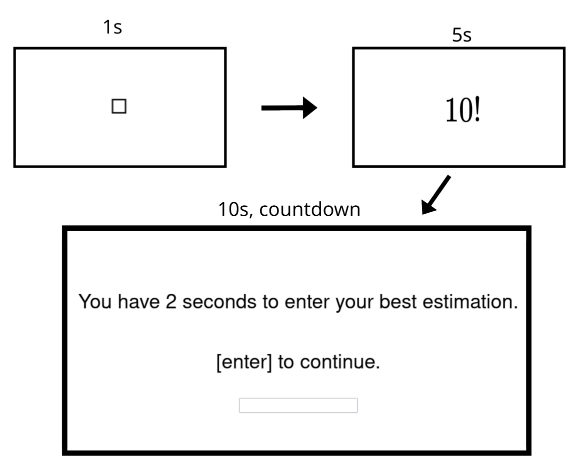

Figure 1: Experimental trial structure.

Participants then completed the 63 estimation trials. The trials were grouped by the function to be estimated, the order of which was randomized. For each function, the order of the input numbers was randomized for half of the participants, and presented in ascending order for the others. On each trial, participants first viewed a fixation square (□) for one second. They were then shown a functional expression (e.g., $\sqrt{2}$). Participants typed in their estimate and pressed ENTER to submit their response. Participants could type any number for their estimate. After 5 seconds, the functional expression was replaced with a timer counting down from 10 seconds, giving participants a total of 15 seconds to provide their estimation. Figure 1 shows an example of a single trial. We still allowed them to answer after the timer ran out, but prompted them to be faster next time. Participants were not provided any feedback for their estimates. At the end of the session, participants were debriefed and compensated.

Exclusion criteria

Of the first 301 participants, we excluded participants who were within 10% of the correct values for 8!, 9!, 10!, and 11! as it was unlikely people were able to estimate those values that accurately in the allotted time without using a calculator (Tversky & Kahneman, 1973). We also removed participants who answered with both the correct first digit and the correct order of magnitude on those trials. Finally, we removed participants who achieved 100% accuracy within 10% of the correct answer on any function or provided the same response to multiple input values of the same function. This left 178 participants.

Results

Accuracy

We used two different metrics to measure accuracy: exact accuracy, which measures whether participants were able to provide the exact value of an expression, and accuracy within 10%, which measures whether participants were able to provide an estimate that was within 10% of the correct value of the expression. Exact accuracy provides a measure of calculation while accuracy within 10% also includes successful estimates. Participants estimated a total of 10.7% of trials with exact accuracy and 19.5% of trials accuracy within 10% of the correct answer. To understand the differences in accuracy between the various functions and input values, we fit two generalized linear mixed effects models with a binomial link function predicting both metrics, using the function participants estimated and rank order of the input value. The rank order of the input value was used instead of the value to facilitate comparisons across the different functions.

We found that exact accuracy was significantly predicted by the function participants estimated ($\chi^2 = 959.30$, $p < .001$; see Figure 2, left panel). Pairwise Tukey contrasts with the Benjamini-Hochberg p-value correction (Benjamini & Hochberg, 1995) revealed that exact accuracy differed between almost all pairs of functions. The adjusted p-values for all pairwise contrasts were less than .001 except for the three functions $\sqrt{n}$, $\log(n)$, and $n\log(n)$, which were comparable.

Similarly, we found that accuracy within 10% was also predicted by the function participants estimated ($\chi^2 = 840.41$, $p < .001$); see Figure 2 (right panel). Pairwise Tukey contrasts with the Benjamini-Hochberg p-value correction revealed that accuracy within 10% differed among almost all pairs of functions: only the p-value for the $\log(n)$ compared to $\log(n)$ contrast ($p = .339$) was greater than .003.

Purely judging from the exact accuracy means (Figure 2, left), a curious pattern emerges: the higher the growth rate of the function, the higher participants' exact accuracy. This is most likely an artifact of exact accuracy being easier for integer versus decimal values. Indeed, looking at accuracy within 10% (Figure 2, right), this pattern mostly disappears. While 2^n and $n!$ have similar means across the measures, the remaining functions have notably higher accuracy within 10% values (which are a superset of the exact accuracy values). This suggests that the accuracy within 10% is a better indicator of estimation in general, and the curious pattern is not particularly meaningful to our research questions.

To visualize how calculation strategies vary between different trials, we separated out the accuracy measures for trials with larger input values, defined as trials where the input rank was 5 or greater. In other words, these were the larger half of trials for each function. We predicted that participants would be less likely to calculate the values of expressions on these larger trials, and that this would especially be the case for 2^n and $n!$, resulting in a drastic reduction in exact accuracy for these functions. We plot the accuracies for the larger input values in Figure 3. Consistent with our prediction, the overall exact accuracies dropped (i.e., compared to Figure 2), suggesting less calculation for these trials (and implying, conversely, more calculation for the easier input values). Additionally, the high accuracy within 10% for the superlinear functions has dropped to the same level as for (or lower than) the sublinear and polynomial functions.

We then evaluated whether the order in which participants saw the input values for a function – ascending vs. random – affected the accuracy of their estimates. Recall that this was a between-participants manipulation. We considered both measures of accuracy, exact accuracy and accuracy within 10%. For each, we fit a generalized linear mixed effects model predicting the metric with the order condition of the participant as a fixed effect and a random intercept per participant. Exact accuracy was similar between both conditions ($\beta = -0.147$, $z = -1.72$, $\chi^2 = 2.967$, $p = .085$). Similarly, accuracy within 10% was similar in both conditions ($\beta = -0.161$, $z = -1.119$, $\chi^2 = 1.25$, $p = .263$). Thus, we exclude this variable in most subsequent analyses.

Monotonicity

Another important property of participants' perceptions of these functions is *monotonicity*: whether they understand that increases in the input value of the function always results in an increase in the magnitude of the expression, as all functions tested in this study were strictly monotonic. A trial was considered monotonically accurate if a participant's estimate for it was greater than their estimate for the trial with the next lower input value for the same function. We fit a

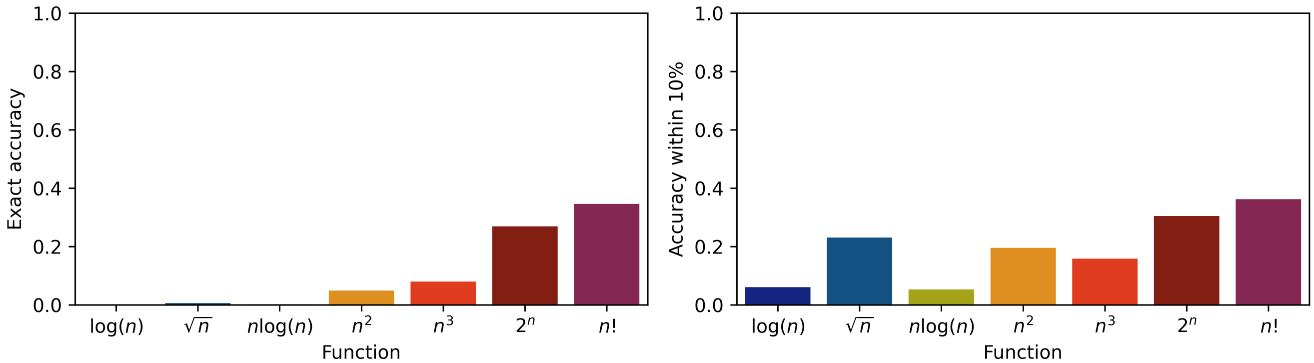

Figure 2: Exact accuracy and accuracy within 10% for all functions. Participants were largely unable to provide accurate estimations for these functions in the ranges presented to them. The similarity between the values of the metrics for $n!$ and 2^n suggests that participants were mostly accurate on items they could calculate exactly. Additionally, people performed poorly at estimating expressions involving the $\log(n)$ function.

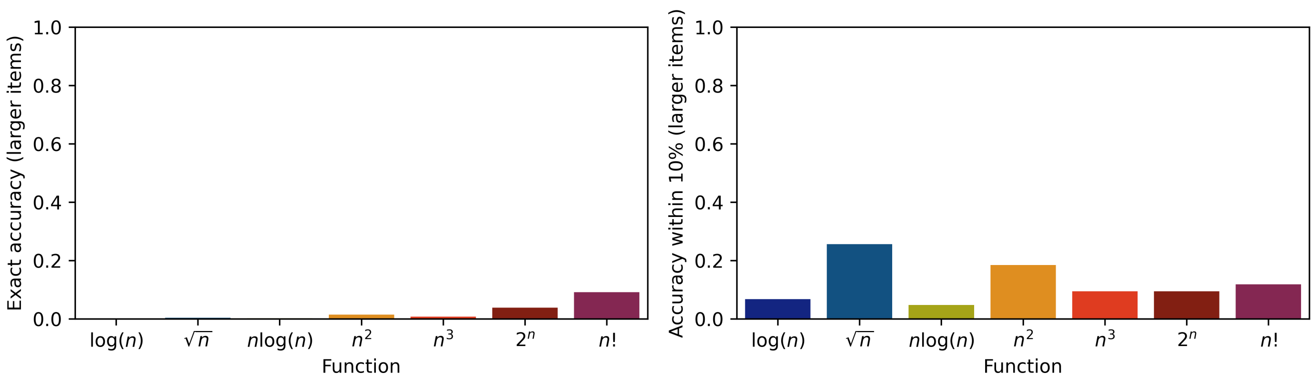

Figure 3: Exact accuracy and accuracy within 10% for larger input values (i.e, of input rank ≥ 5). The reduced exact accuracies for these trials. However, the largely intact accuracy within 10% values suggest that participants were unable to calculate these values and were forced to estimate them.

generalized linear mixed effects model predicting monotonicity accuracy for the function participants estimated, the input rank of the input value, and the estimation order, along with a random intercept per participant. We found a main effect of estimation order condition: participants were 1.86 times as likely to provide a monotonically accurate response in the ascending condition compared to the random condition ($\beta = -0.63$, $\chi^2 = 68.3$, $z = -8.264$, $p < .001$). Monotonicity accuracy also depended on the function participants estimated ($\chi^2 = 293.244$, $p < .001$). Pairwise Tukey contrasts with Benjamini-Hochberg p-value adjustments indicated that the superpolynomial functions 2^n and $n!$ were similar to each other ($p = .16$) and were the most monotonically accurate, likely a consequence of their higher accuracy (explained below). The remaining functions were largely similar to each other in monotonicity accuracy, and significantly different from the superpolynomial functions. Finally, participants were 1.09 times as likely to make monotoniticity errors for each increase in input rank ($\beta = -0.088$, $\chi^2 = 69.083$, $z = -8.312$, $p < .001$). Thus, for example, the expressions with the highest magnitude were 2.022 times more likely to include monotoniticity errors compared to the expressions of the lowest magnitude.

Underestimation

For trials where participants did not answer exactly, we asked whether there was a systematic bias in their errors. For these trials, their bias was for underestimation (Figure 4). Participants underestimated on 60.2% of all trials.

We fit a generalized linear mixed effects model predicting whether participants underestimated with fixed and random effects for the function estimated by participants along with a random intercept per participant. We found a main effect of the function participants estimated ($\chi^2 = 252.101$, $p < .001$). Pairwise Tukey contrasts revealed significant differences in underestimation between $\log(n)$ and all other functions, $\sqrt{n}$ and all other functions, $n\log(n)$ with n^2 and n^3, and n^3 with 2^n. We then conducted a binomial test for each function com-

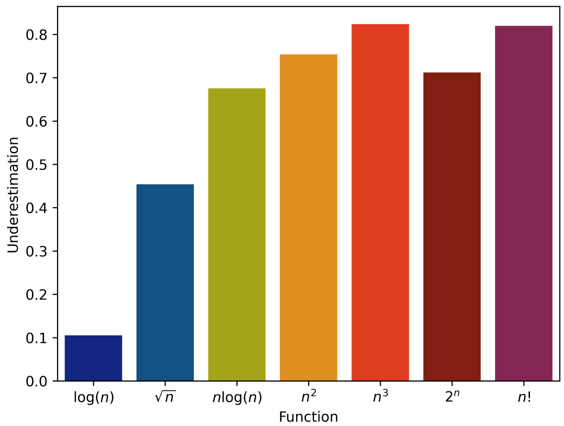

Figure 4: Differences in underestimation between the functions participants estimated. Underestimation is defined as the proportion of trials where the participants' estimate was lower than the correct value. Values greater than 0.5 indicate underestimation and values lower than 0.5 indicate overestimation of the value of functional expressions.

paring the rate of underestimation to chance (i.e., 0.5). All adjusted *p*-values were lower than .001.

As Figure 4 shows, only the two sublinear functions were overestimated; the rest were underestimated. The results are largely consistent with our prediction that participants use qualitative linear and polynomial functions to approximate all functions considered here. That is, *people have a linear bias in their estimation of the growth of functions*, causing sublinear functions to be overestimated and polynomial and superpolynomial functions to be underestimated.

Reaction time

We investigated the time participants took to estimate the value of the functional expressions. We fit a linear mixed effects model predicting the time participants took to estimate a functional expression with fixed effects of the function and the rank of the input value with a random intercept per participant. There was a main effect of the function participants estimated ($\chi^2 = 149.53$, $p < .001$).

Pairwise Tukey contrasts with the Benjamini-Hochberg *p*-value correction found that participants spent more time estimating the $\sqrt{n}$, n^2, and n^3 functions, which were similar to each other and different from other functions in their RT profile ($p < 0.05$). $\log(n)$, $n\log(n)$, and 2^n also followed similar reaction time profiles to each other (*p* ¡ 0.05). The factorial function, however, was in a class by itself: participants estimated it more quickly than all the other functions ($p < .001$). This may be partly driven by the fact that prior estimations can be repurposed on subsequent trials. Additionally, we found evidence for an effect of input rank ($\beta = 124.88$, $\chi^2 = 13.082$, $t = 3.617$, $p = .0029$) such that an increase of one rank was associated with a 125 ms increase in reaction time.

Shape match

Finally, we conducted a shape-fitting analysis to further probe the smaller number of qualitative functions participants might be using to estimate the growth of all functions: sublinear, linear, polynomial, and/or superpolynomial. We first normalized each participant's estimates for each function to the $[0,1]$ range to capture the shape of their estimates instead of their magnitudes. Then, for each function and participant, we predicted their estimates for the function using the normalized values of correct function and those of the other alternative functions (the tested functions and a baseline linear function). This resulted in eight R^2 values per function. A high R^2 value indicates participants' estimates fit the shape (functional form) well.

As can be seen in Figure 5, the shapes of participants' estimates for polynomial and superpolynomial functions were best predicted by a polynomial function, n^3. This result indicates that participants understand superlinear functions relative to a qualitative polynomial function. Additionally, participants' perceptions of the sublinear functions $\log(n)$ and $\sqrt{n}$ as well as the next slowest-growing function, $n\log(n)$, were fit best by the linear function most of the time. We interpret this as evidence that participants use a qualitative linear function when estimating sublinear (and near-linear) functions.

Discussion

An important part of mathematical development is acquiring an intuitive sense of numbers. Functions play an important role in abstracting over these intuitions. However, the field has only recently begun to investigate their psychological underpinnings.

The current study asked people to estimate the growth of seven functions common in algorithmic analysis. Compared to prior work that used smaller input ranges Marupudi et al. (2022), participants' estimates were linearized: their estimates of sublinear functions were best fit by the mathematical values of the linear function, and their estimates of superlinear functions were best fit by polynomial functions. Likely as consequence of this linearization, participants overestimated the sublinear functions: $\log(n)$ and $\sqrt{n}$, and underestimated the polynomial and superlinear functions n^2, n^3, 2^n, and $n!$. Also as predicted, participants made generally poor estimates for functions involving logarithms. This difficulty is consistent with prior research (Confrey & Smith, 1995; Parker & Lewis, 2014). Misperceiving logarithms as linear can potentially have important negative consequences for decision-making and using mathematics to solve problems. It might lead to misconceptions about algorithms like binary search. Additionally, people might wrongly perceive a problem as infeasible due to resource constraints. Despite these errors, the data showed that participants were mostly aware of the monotonic nature of all functions in this study.

These results do come with limitations. While the conclusions here might hold for symbolic perceptions of the func-

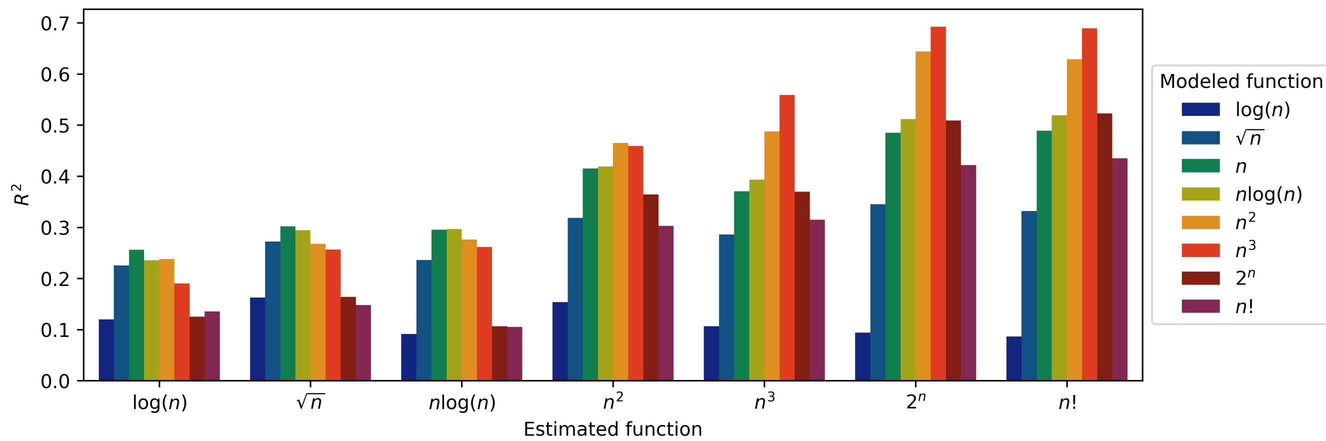

Figure 5: Various shape fits for the functions participants estimated. We find evidence of a linear bias, as a linear model fits participants' estimates of sublinear functions best, and polynomial models fit participants' estimates of superlinear functions best.

tional expressions, it is possible that participants do hold more accurate internal representations. For example, Ciccione et al. (2022) discovered that participants were better at estimating exponential functions when presented in a noise-free graphical plot or when data were presented using a logarithmic scale. An analogous study using graphed curves instead of symbolic expressions can be run to answer that question for the various functions. Another limitation of the current work is that the ranges of the input values provided to participants were based on subjective judgement and pilot testing. Development of an objective criterion to determine equivalent ranges between the functions can help improve measurement of the perceptions of these functions. It is also important to note that the set of functions used in this study is not comprehensive. There are functions that grow much slower than log(n) and much faster than $n!$. For example, Ackermann's function is a function of 2 positive integers for which A(1, 1) = 3, A(2, 2) = 7, and A(3, 3) = 61, and A(4, 4) = $2^{2^{65536}} - 3$. Humans may not be capable of intuiting the growth of such a function.

Finally, certain expressions presented to participants involved large numbers, e.g., log(99789). Since people recruit a logarithmically-compressed representation for numerical magnitudes, they are prone to perceive large numbers as smaller than they are (Izard & Dehaene, 2008). Despite this, participants still overestimated the value of logarithmic functions, underscoring the size of the overestimation effect.

The current results set the stage for future instructional studies of the best ways to build intuitions about the growth of functions in students. The current estimation task can be adapted into a pre-post assessment of reasoning about the growth of functions that can then be used in studies investigating different instructional approaches to teaching algorithmic analysis and big-O notation. This would be an important innovation because many measures of computational thinking lack items relevant for algorithmic thinking (Román-González et al., 2017). Those that do often include items that merely measure people's impressions of algorithms ("I think that I have a special interest in the mathematical processes") rather than their direct knowledge of algorithms (Korkmaz et al., 2017). Also, the estimation task can potentially be adapted into an instructional activity for building students' intuitions about the growth of functions. For example, they could be presented with two functions and asked which would yield the greater value on the same input, or on different inputs (Rittle-Johnson & Star, 2007; Schwartz & Bransford, 1998).

We do not know whether participants can actually translate the descriptions of natural processes or algorithms into symbolic representations, and how these translation processes are associated with each individual's intuitions for the growth of functions. Accurate estimation of the growth of functions is important not just for computer science, but also for computational thinking in all sciences. For example, undergraduates have difficulty reasoning about the explosion and collapse of biological processes even when interacting with a dynamic, graphic simulation—producing estimates that are off by an order of magnitude (Wagenaar & Timmers, 1979). The COVID-19 pandemic revealed that one reason people have difficulty understanding the spread of infectious diseases is their perception of the growth of nonlinear functions over time. The failure to understand the difference between linear and exponential growth might underlie the widespread underprediction of the expected number of cases and deaths during the first wave of the COVID-19 epidemic. The current research can help at educating the public to reason about the spread of infectious diseases, which can be exponential, logarithmic, and polynomial at different phases of pandemics (Keeling & Rohani, 2011; Kermack et al., 1927), and for using computational simulations, analogies, and explanations to develop an understanding of these issues.

References

Alagic, M., & Palenz, D. (2006). Teachers explore linear and exponential growth: Spreadsheets as cognitive tools. *Journal of Technology and Teacher Education*, *14*(3), 633–649.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Busemeyer, J. R., Byun, E., Delosh, E. L., & McDaniel, M. A. (1997). Learning functional relations based on experience with input-output pairs by humans and artificial neural networks.

Ciccione, L., Sablé-Meyer, M., & Dehaene, S. (2022). Analyzing the misperception of exponential growth in graphs. *Cognition*, *225*, 105112. https://doi.org/10.1016/j.cognition.2022.105112

Confrey, J., & Smith, E. (1995). Splitting, covariation, and their role in the development of exponential functions. *Journal for research in mathematics education*, *26*(1), 66–86.

del Vado Vírseda, R. (2019). Computability and Algorithmic Complexity Questions in Secondary Education. *Proceedings of the ACM Conference on Global Computing Education*, 51–57. https://doi.org/10.1145/3300115.3309507

del Vado Vírseda, R. (2021, March 3). Learning from the Impossible: Introducing Theoretical Computer Science in CS Mathematics Courses. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 952–958). Association for Computing Machinery. Retrieved August 11, 2021, from https://doi.org/10.1145/3408877.3432475

Ellis, A. B., Ozgur, Z., Kulow, T., Dogan, M. F., & Amidon, J. (2016). An exponential growth learning trajectory: Students' emerging understanding of exponential growth through covariation. *Mathematical thinking and learning*, *18*(3), 151–181.

Ferrari-Escolá, M., Martinez-Sierra, G., & Méndez-Guevara, M. E. M. (2016). "Multiply by adding": Development of logarithmic-exponential covariational reasoning in high school students. *The Journal of Mathematical Behavior*, *42*, 92–108.

Gelpi, R., Saxena, N., Lifchits, G., Buchsbaum, D., & C. G. Lucas. (2021). Sampling heuristics for active function learning.

Izard, V., & Dehaene, S. (2008). Calibrating the mental number line. *Cognition*, *106*(3), 1221–1247.

Kalish, M. L., Lewandowsky, S., & Kruschke, J. K. (2004). Population of linear experts: Knowledge partitioning and function learning. *Psychological review*, *111*(4), 1072.

Keeling, M. J., & Rohani, P. (2011). *Modeling infectious diseases in humans and animals*. Princeton university press.

Kermack, W. O., McKendrick, A. G., & Walker, G. T. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, *115*(772), 700–721. https://doi.org/10.1098/rspa.1927.0118

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in human behavior*, *72*, 558–569.

Levitin, A. (2005). Analyze that: Puzzles and analysis of algorithms. *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, 171–175. https://doi.org/10.1145/1047344.1047409

MacCormick, J. (2018). Strategies for Basing the CS Theory Course on Non-decision Problems. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 521–526. https://doi.org/10.1145/3159450.3159557

Marupudi, V., Bye, J. K., & Varma, S. (2022). Foundations for Algorithmic Thinking: Estimating the Growth of Functions. *Proceedings of the 2022 Annual Meeting of the American Educational Research Association*.

Parker, M., & Lewis, C. (2014). What makes big-O analysis difficult: Understanding how students understand runtime analysis. *Journal of Computing Sciences in Colleges*, *29*(4), 164–174.

Patel, P., & Varma, S. (2018). How the abstract becomes concrete: Irrational numbers are understood relative to natural numbers and perfect squares. *Cognitive science*, *42*(5), 1642–1676. https://doi.org/10.1111/cogs.12619

Rittle-Johnson, B., & Star, J. R. (2007). Does comparing solution methods facilitate conceptual and procedural knowledge? An experimental study on learning to solve equations. *Journal of Educational Psychology*, *99*(3), 561–574. https://doi.org/10.1037/0022-0663.99.3.561

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, *72*, 678–691.

Schwartz, D. L., & Bransford, J. D. (1998). A time for telling. *Cognition and instruction*, *16*(4), 475–5223.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive psychology*, *5*(2), 207–232.

Wagenaar, W. A., & Timmers, H. (1979). The pond-and-duckweed problem; Three experiments on the misperception of exponential growth. *Acta Psychologica*, *43*(3), 239–251. https://doi.org/10.1016/0001-6918(79)90028-3

Wagenaar, W. A., & Sagaria, S. D. (1975). Misperception of exponential growth. *Perception & Psychophysics*, *18*(6), 416–422. https://doi.org/10.3758/BF03204114