HUMAN PERCEPTION OF EXPONENTIALLY INCREASING DATA DISPLAYED ON A LOG SCALE

by

Emily Anna Robinson

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Statistics

Under the Supervision of Professors Susan VanderPlas and Reka Howard

Lincoln, Nebraska

August, 2022

Table of Contents

List of Figures

List of Tables

Chapter 1

Literature Review

Editing text colors: Emily's editing color. Emily may also use mostly black text as well. Susan's editing color. Reka's editing color.

1.1 Introduction to Graphics

Advanced technology and computing power has promoted data visualization as a central tool in modern data science (Unwin, 2020). Data visualization is defined by whom? as the art of drawing graphical charts in order to display data. Fig. 1.1 illustrates the process of creating a graphic from a data set through the use of variable mapping, data transformations, coordinate systems, and aesthetic features (Vanderplas, Cook, & Hofmann, 2020) This is more of a grammar of graphics flowchart, which isn't *quite* the same thing. I would honestly lead with this part... Graphics are useful for data cleaning, exploring data structure, and have been an essential component in communicating information for the last 200 years (Lewandowsky & Spence, 1989; Unwin, 2020; Vanderplas, Cook, & Hofmann, 2020) During the 20th century, companies began utilizing graphics to understand their mechanics and support business decisions; news sources began displaying graphics of weather forecasts as a means to communicate critical information and aid in

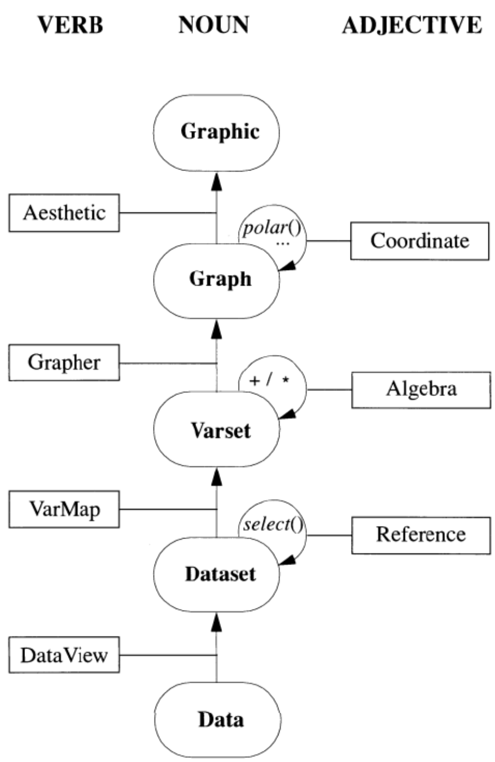

Figure 1.1: Graphic flowchart

decision-making (Vanderplas, Cook, & Hofmann, 2020), and countries began using graphics in order to better understand their population and economic interests Might want to add a set of pictures showing the different types of graphics from this sentence - playfair's balance of trade, statistical atlas, and something from e.g. early IBM?. Today, we encounter data visualizations everywhere, researchers include graphics to communicate their results in scientific publications and mass media relies on graphics to convey news stories to the public through newspapers, TV, and the Web (Unwin, 2020).

In general, the public holds the self perception that numbers are difficult to understand and that they did not perform well in mathematics during school (Unwin, 2020). In contrary, there tends to be a positive self perception when it comes to graphics as they are viewed more as illustrations and not as critical parts of an argument. While tables can be burdensome to readers, graphics can improve the interpretation and representation of the same data (Uri & Haemer, 1948). Shah, Mayer, & Hegarty (1999) presents an opposing argument claiming there are still difficulties interpreting and explaining quantitative information depicted in graphs. I think Shah's argument is that some forms of graphics are too complicated, but that doesn't mean she's arguing that graphics aren't useful. One explanation for the opposing views is that the general public is untrained in the ability required for the evaluation of graphic material. The complexity of graphic devices is directly related to the degree and formality of training necessary for understanding (Haemer & Kelley, 1949). This is 1. a bit hard to understand, and 2. seems to be a slightly different tone than before. Probably not the ideal way to end a paragraph.

Although statistical graphics have become widely used and valued in science, business, and in many other aspects of life, as creators of graphics, we are too accepting of them as default without asking critical questions about the graphics we create or view (Unwin, 2020). Vanderplas, Cook, & Hofmann (2020) poses the general question we must ask ourselves, "how effective is this graph at communicating useful information?" Higher quality of technology has influenced the creation, replication, and complexity of graphics as there are an infinitely many number of graphical displays and design choices that can be implemented at faster speeds with more flexibility. The creator of a graphic makes decisions about the variables displayed, the type of graphic, the size of the graphic and the aspect ratio, the colors and symbols used, the scales and limits, the ordering of categorical variables, and the ordering of variables in multivariate displays (Unwin, 2020). In response to the increasing number of design choices, consistent themes and higher standards are being placed on graphics. Selecting from an extensive list of styles and choices of graphics in order to effectively communicate insights into the data is a challenging task. A consistent concern is the lack of theory of graphics available to build on; better theory should result in better graphics. Creators of graphics need an established set of concepts and terminology to build their graphics from so they can actively choose which of many possible graphics to draw in order to ensure their charts are effective.

Many efforts have been made to provide guidelines for graphical designs including Wilkinson's Grammar of Graphics (Wilkinson, 2012). These guidelines provide the ground work necessary for data plots to be depicted and interpreted as statistics (Buja et al., 2009; Majumder, Hofmann, & Cook, 2013). Vanderplas, Cook, & Hofmann (2020) define a statistic as, "a functional mapping of a variable or set of variables." The grammar of graphics constructs visual statistics through the use of "tidy data," characterized as a data set in which each variable is in its own column, each observation is in its own row, and each value is in its own cell (Hadley Wickham & Grolemund, 2016). The grammar allows variables, as designated in columns, to be mapped to different elements of the graphic such as the axes, colors, shapes, or facets. Software, such as Hadley Wickham's ggplot2 (Hadley Wickham, 2011), aims to implement these guidelines recommended through the grammar of graphics.

Later, it is illustrated how the structure of "tidy data" and the construction of graphics as statistics aid themselves for easy experimentation which allows researchers to compare the effectiveness and understand the perception of different types of charts (Vanderplas, Cook, & Hofmann, 2020). It is important to consider the purpose and motivation behind the generation of the chart as well as the complexity and intended audience you intend to view the chart. This is circling around to a different paragraph – might be a sign that you need to tighten your argument a bit? A general guideline when generating graphics is to keep it familiar in order to not intimidate your audience and to encourage further interaction from the viewers (Unwin, 2020). Despite past attempts to improve the use of graphics in science, Gordon & Finch (2015) evaluated 97 graphs for overall quality, based on five principles of graphical excellence, and found there is still an astonishing lack in the quality of graphics. More startling is the fact that the source of the graphic from an applied science or a statistics graphic had no effect on the quality of the graphic. The improvement in graphics depends on both a better definition of variables, units of measurements, scales, and other graphical elements as well as a typical use of grid lines on an accepted set of graphical forms. With the support of changes in software defaults, future work must be done in order to implement the academic research being conducted in graphics into practice for both academics and non-academics in order to achieve a higher standard of the graphics being presented (Vanderplas, Cook, & Hofmann, 2020).

1.2 Perception and Psychophysics

In order to develop guiding principles for generating graphics effective in communication, we must first understand the basic mechanics of the human perceptual system and the biases we are vulnerable to (Goldstein & Brockmole, 2017). The perceptual process is complex and involved (Fig. 1.2).

This process is broken down into sensation - involving simple processes that occur right at the beginning of a sensory system - and perception - involving higher-order mechanisms and identified with more complex processes. The perceptual process begins when there is a stimulus in the environment and light is reflected and focused back into the viewer's eyes. Within the eye, the light reflected

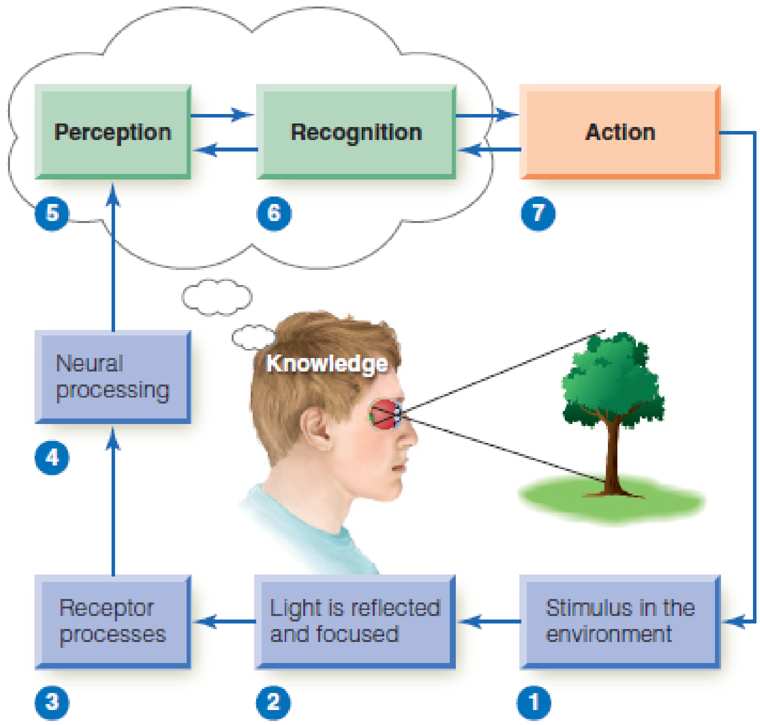

Figure 1.2: Perceptual process

is transformed and focused by the eye's optical system and an image is formed on the receptors of the viewer's retina. It is important to note that everything a person perceives is based not on direct contact with stimuli but on representations of stimuli that are formed on the receptors and the resulting activity in the person's nervous system. Once light is reflected and focused, our visual receptors respond to the light and transform the light energy into electrical energy through a process called transduction. Signals from the receptors are then transmitted through the retina, to the brain, and then within the brain where perception - what do you see? - and recognition - what is it called? - occur. After recognition, viewers take some sort of motor action - for example, move closer to the object. The perceptual process is not direct and instead takes on more of a cyclic nature where a person may go through many iterations of stimuli, perception, recognition, and action before the final image is identified and understood.

When perception occurs, we first experience the **preattentive stage** in which we observe color, shape, size, and other basic information about the stimuli being perceived. Preattentive perception effects are automatically processed within the first 500 millisectonds of viewing and do not depend on sustained cognitive attention (Vanderplas, Cook, & Hofmann, 2020). Following the preattentive stage, **direct attention** is required for additional processing to allow us to draw connections between components that assist in our interpretation of the stimuli. When viewing a chart or graph, most insights we gain are due to the cognitive processes that occur after attention is focused on specific aspects of the graph. The knowledge a viewer brings to the situation influences their cognitive evaluation of the stimuli. Citation? You should probably also provide a bit more context for this piece of the paragraph instead of ending with dropping this one fact...

The relationship between physiology and perception can provide us information about how graphics may be understood and interpreted. According to a cognitive analysis, graph interpretation involves (a) relatively simple pattern perception and association processes in which viewers can associate graphic patterns to quantitative referents and (b) more complex and error-prone inferential processes in which viewers must mentally transform data (Shah, Mayer, & Hegarty, 1999). Pyschophysics, the branch of psychology that deals with the relationships between physical stimuli (e.g. light) and mental phenomena, aims to provide explanations of this relationship and point out human perceptual biases. By examining both behavior and physiology together, we are able to understand the mechanisms responsible for perception.

1.3 Graphical Experiments

One way in which we determine the relationship between behavior and physiology is through the use of graphical tests. These tests may take many forms: identifying differences in graphs, reading information off of a chart accurately, using data to make correct real-world decisions, or predicting the next few observations. All of these types of tests require different levels of use and manipulation of the information presented in the chart. The initial push to develop classification and recommendation systems for charts was grounded on heuristics rather than on experimentation (Vanderplas, Cook, & Hofmann, 2020). Request were made for the validation of the perception and utility of statistical charts through graphical experiments. Initial experiments struggled with methodological issues (Croxton & Stein, 1932; Croxton & Stryker, 1927; Eells, 1926) with most early experimentation stemming from psychophysics research on the perception of size and shape (Teghtsoonian, 1965). While a typical psychophysics experiment focuses on whether an effect is detectable and whether the magnitude of the effect can be accurately estimated, these early experiments instead depended on speed and accuracy for plot evaluation (Lewandowsky & Spence, 1989; Spence, 1990; Teghtsoonian, 1965).

Cognitive psychologists and statisticians made progress by conducting experiments to identify perceptual errors associated with different styles of graphics and charts (Cleveland & McGill, 1985; Spence, 1990). These later experiments relied on similar methodology as early studies by relying on participants directly reading information from the charts to provide a quantitative estimate or answering a predefined question; as with the early studies, accuracy and response time being evaluated (Amer, 2005; Broersma & Molenaar, 1985; Cleveland & McGill, 1984; Dunn, 1988; Peterson & Schramm, 1954; Tan, 1994). Too much use of ”rely/relied”, provide some examples of quantitative estimation questions – this will better support Ch4 Cleveland & McGill (1984) provide a basis for perceptual judgment, still utilized today, by examining six basic stimuliNot really stimuli, so much as concepts or aesthetics if you want to go there...: position along a common scale, position along nonaligned scales, length, angle, slope, and area. Other experiments established the notion that redesigning graphs can result in the improvement of the viewer's interpretation (Shah, Mayer, & Hegarty, 1999). This is done by relying on gestalt principles to minimize the inferential processes and maximize the pattern association processes required to interpret relevant information. This next bit seems like a different paragraph that should probably come a bit earlier... The viewer must first encode the visual array by identifying meaningful visual features, such as a straight light slanting downward. Next, the viewer must classify the quantitative measures and relationships in which those visual features illustrate, such as a decreasing linear relationship between x and y. The last step involves translating the quantitative measures and relationships to the variables defined in the data set, such as a population decreasing over years. These studies establish the process in which viewers interact with charts by first perceptually observing the visual features and later translating to cognitive processing of the information depicted by those features (Shah & Carpenter, 1995).

In recent years, there have been advancements in the methodology used to investigate the effectiveness of statistical charts (Vanderplas, Cook, & Hofmann, 2020). Some of the new methodsLineups have been around since 2009, they're not quite *new* so much as newer than the other options?, such as the lineup protocol (Buja et al., 2009), utilize the grammar of graphics designation of a data plot as a statistic through the functional mapping of variable(s). Lineups don't depend on grammar of graphics, though grammar of graphics makes them exponentially more powerful. This allows the data plot to be tested similar to other statistics, by comparing the actual data plot to a set of plots with the absence of any data structure we can test the likelihood of any perceived structure being significant (Vanderplas, Cook, & Hofmann, 2020). While the methodology of these recent experiments differs from earlier studies, the focus is still on initial perception and graph comprehension with a relatively small amount of work conducted to understand the effect of design choices on higher cognitive processes such as learning or analysis (Green & Fisher, 2009). This is in part because lineups only ask about perceived differences and do not test the actual ability to read information off of a graph accurately or the conclusions drawn from the graph. They are a lower-level tool, which makes them powerful because they can eliminate ambiguous questions, but also limits their real-world application in the absence of complex experimental designs (feature hierarchy, etc.). Most recent graphics experiments have utilized tools such as Amazon Turk, Prolific, Reddit, and other crowd sourcing websites to evaluate the psychophysics and patterns associated with design choices (VanderPlas & Hofmann, 2017)You need to cite more than one experiment if you want to make this claim, and they can't all be lineup experiments. Really, you'd need a very long list of experiments and would ideally be able to show which used online vs. in-person testing.... Such studies tend to be limited by the questions asked while other methods such as Thinking aloud and Eye Tracking allow researchers to evaluate the overall perception of the graphic (Vanderplas, Cook, & Hofmann, 2020).

1.4 Logarithmic Scales and Mapping

We have recently experienced the impact graphics and charts have on a large scale through the SARSNCOV-2 pandemic (COVID-19). At the beginning of 2020, we saw an influx of dashboards being developed to display case counts, transmission rates, and outbreak regions (GmbH, 2020); mass media routinely showed charts to share information with the public about the progression of the pandemic (Romano, Sotis, Dominioni, & Guidi, 2020). People began seeking out graphical displays of COVID-19 data as a direct result of these pieces of work (GmbH, 2020); providing increased and ongoing exposure to these graphics over time. It would be appropriate to add some of those graphics in here, if you wanted to... Many of these graphics helped guide decision makers to implement policies such as shut-downs or mandated mask wearing, as well as facilitated communication with the public to increase compliance (Bavel et al., 2020). As graphics began to play an important role in the communication of the pandemic, creators of graphics were faced with design choices in order to ensure their charts were effective.

When faced with data which spans several orders of magnitude, we must decide whether to show the data on its original scale (compressing the smaller magnitudes into relatively little area) or to transform the scale and alter the contextual appearance of the data. One common solution is to use a log scale transformation to display data over several orders of magnitude within one graph. Logarithms turn multiplicative relationships additivephrasing?, showing elasticities and other proportional changes, and also linearize power laws (Menge et al., 2018). They also have practical purposes, easing the computation of small numbers such as likelihoods and transforming data to fit statistical assumptions. When presenting log scaled data, it is possible to use either un-transformed scale labels (for example, values of 1, 10 and 100 are equally spaced along the axis) or log transformed scale labels (for example, 0, 1, and 2, showing the corresponding powers of 10).In the grammar of graphics/ggplot2 formulation, this is the difference between a scale and a transformation... it might be worth seeing if that is a hadley-ism or if that is a wider convention. We have recently experienced the benefits and pitfalls of using log scales as COVID-19 dashboards displayed case count data on both the log and linear scale (Burn-Murdoch et al., 2020; Fagen-Ulmschneider, 2020). In spring 2020, during the early stages of the COVID-19 pandemic, there were large magnitude discrepancies in case counts at a given time point between different geographic regions (e.g. states and provinces as well as countries and continents). During this time, we saw the usefulness of log scale transformations showing case count curves for areas with few cases and areas with many cases within one chart. As the pandemic evolved, and the case counts were no longer spreading exponentially, graphs with linear scales seemed more effective at spotting early increases in case counts that signaled more localized outbreaks. This is only one recent example of a situation in which both log and linear scales are useful for showing different aspects of the same data. There are long histories of using log scales to display results in ecology, psychophysics, engineering, and physics (Heckler, Mikula, & Rosenblatt, 2013; Menge et al., 2018) elaborate on this a bit - when are they used, and where are they effective? Can you draw more comprehensive conclusions about the use of log scales?. The usefulness of the log scale in science is exemplified in Fig. 1.3 (Munroe, 2005).

You might start this out by pointing out that while logarithms seem unnatural at first, we learn to count by ones, tens, and hundreds – that is, in a base10 order of magnitude system. Then you can transition to research showing one, two, 10 and

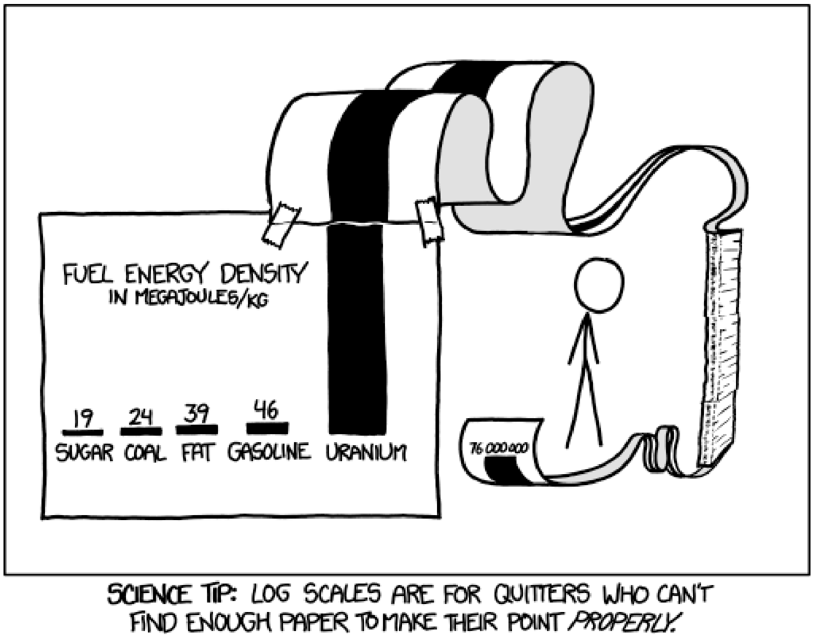

Figure 1.3: Log scale comic

log-number scales.

Research suggests our perception and mapping of numbers to a number line is logarithmic at first, but transitions to a linear scale later in development, with formal mathematics education (Dehaene, Izard, Spelke, & Pica, 2008; Siegler & Braithwaite, 2017, 2017; Varshney & Sun, 2013). For example, a kindergartner asked to place numbers 1-10 along a number line would place 3 close to the middle, following the logarithmic perspective (Varshney & Sun, 2013). Might be worth showing this graphically if you can reproduce pics from the paper... With basic training, members of remote cultures with a basic vocabulary and minimal education understood the concept that numbers can be mapped into a spacial space (Dehaene, Izard, Spelke, & Pica, 2008). There was a gradual transition from logarithmic to linear scale as the mapping of whole number magnitude representations transitioned from a compressed (approximately logarithmic) distribution to an approximately linear one. These results indicate the universal and cultural-dependent characteristics of the sense of number.

Weber's law isn't quite the same thing as the numerical cognition stuff on log scales - it's a different perceptual phenomenon that is definitely related, but distinct in causal mechanism (as far as I know).

This phenomenon was first discovered by Ernst Weber, an early psychophysics researcher, by determining the relationship between the difference threshold (smallest detectable difference between two sensory stimuli; known as the "Just Noticable Difference") and the magnitude of a stimulus. This holds true for a variety of stimuli such as weight, light, and sound as well as for a range of magnitudes; larger numbers require a proportional larger difference in order to remain equally discriminate (Dehaene, Izard, Spelke, & Pica, 2008). Known as **Weber's law**, it was established that we do not notice absolute changes in stimuli, but instead that we notice the relative change (Sun, Wang, Goyal, & Varshney, 2012). Numerically, Weber's Law is defined as

$$\frac{\Delta S}{S} = K \tag{1.1}$$

where ΔS represents the difference threshold, S represents the initial stimulus intensity, and K is called Weber's contrast which remains constant as the magnitude of S changes. Gustav Fechner, a founder of psychophysics, provided further extension to Weber's law by discovering the relationship between the perceived intensity is logarithmic to the stimulus intensity when observed above a minimal threshold of perception (Sun, Wang, Goyal, & Varshney, 2012). Formally known as the Weber-Fechner law is derived from Weber's law as

$$P = K \ln \frac{S}{S_0} \tag{1.2}$$

where P represents the perceived stimulus, K represents Weber's contrast, S represents the initial stimulus intensity, and S_0 represents the minimal threshold of perception.

Assuming there is a direct relationship between perceptual and cognitive processes, it is reasonable to assume numerical representations should also be displayed on a nonlinear, compressed number scale. Therefore, if we perceive logarithmically by default, it is a natural (and presumably low effort) way to display information and should be easy to read and understand/use. The idea is compression enlarges the coding space, thus increasing the dynamic range of perception and firing neurons within our visual system (Nieder & Miller, 2003). Similar to the training and education required to transition from logarithmic mapping to linear mapping, there is also necessary training required in the assessment of graphical displays associated with logarithmic scales. Haemer & Kelley (1949) identify semi-logarithmic charts for temporal series as requiring a certain degree of technical training.

1.5 Underestimation of Exponential Growth

Start out by showing exponential growth e.g. 3 stages (early, mid, late) to explain why this is a hard problem to solve... This reads like a bulleted list turned into a paragraph, which isn't bad, but it does need some work - turn it into a narrative, explain weaknesses in these studies, etc.

Early studies explored the estimation and prediction of exponential growth, finding that growth is underestimated when presented both numerically and graphically but that numerical estimation is more accurate than graphical estimation for exponential curves (Wagenaar & Sagaria, 1975). One way to improve estimation of increasing exponential trends is to provide immediate feedback to participants about the accuracy of their current predictions (Mackinnon & Wearing, 1991). While prior contextual knowledge or experience with exponential growth does not improve estimation, instruction on exponential growth reduces the underestimation: participants adjust their initial starting value but not their perception of growth rate (Jones, 1977; Wagenaar & Sagaria, 1975). Our inability to accurately predict exponential growth might also be addressed by log transforming the data, however, this transformation introduces new complexities; most readers are not mathematically sophisticated enough to intuitively understand logarithmic math and translate that back into real-world effects.

In Menge et al. (2018), ecologists were surveyed to determine how often ecologists encounter log scaled data and how well ecologists understand log scaled data when they see it in the literature. Participants were presented two relationships displayed on linear-linear scales, log-log scales with untransformed values, or log–log scales with log transformed values. The authors propose three types of misconceptions participants encountered when presented data on log-log scales: 'hand-hold fallacy,' 'Zeno's zero fallacy,' and 'watch out for curves fallacies.' These misconceptions are a result of linear extrapolation assuming that a line in log-log space represents a line instead of the power law in linear-linear space. The study found that in each of these scenarios, participants were confident in their incorrect responses, indicating incorrect knowledge rather than a lack of knowledge.

You might take a whack at illustrating some of these fallacies

The 'hand-hold fallacy' stems from the misconception that steeper slopes in log-log relationships are steeper slopes in linear-linear space. In fact, it is not only the slope that matters, but also the intercept and the location on the horizontal axis since a line in log-log space represents a power law in linear-linear space (i.e. linear extrapolation). Emerging from 'Zeno's zero fallacy' is the misconception that positively sloped lines in log-log space can imply a non-zero value of y when x is zero. This is never true as positively sloped lines in log-log space actually imply that y = 0 when x = 0. This misconception again is a result of linear extrapolation assuming that a line in log-log space represents a line instead of the power law in linear-linear space. The last misconception, 'watch out for curves fallacies' encompasses three faults: (1) lines in log-log space are lines in linear-linear space, (2) lines in log-log space curve upward in linear-linear space, and (3) curves in log-log space have the same curvature in linear-linear space. Linear extrapolation is again responsible for the first and third faults while the second fault is a result of error in thinking that log-log lines represent power laws (which are exponential relationships), and all exponential relationships curve upward; this is only true when the log-log slope is greater than 1. Menge et al. (2018) found that in each of these scenarios, participants were confident in their incorrect responses, indicating incorrect knowledge rather than a lack of knowledge.

1.6 Research Objectives

In my research, I conduct three graphical experimental tasks to evaluate the impact our choice of scale (log/linear) has on human perception of exponentially increasing trends. The first experiment evaluates whether our ability to perceptually notice differences in exponentially increasing trends is impacted by the choice of scale. I conducted a visual inference experiment in which participants were shown a series of lineup plots and asked to identify the panel that was most different from the others. The other experimental tasks focus on determining whether there are cognitive disadvantages to log scales: do log scales make it harder to make use of graphical information? I conducted a graphical task similar to the New York Times "You Draw It" page to test an individual's ability to use and make predictions for exponentially increasing data. Participants were asked to draw a line using their computer mouse through the increasing exponential trend shown on both scales. In addition to differentiation and prediction of exponentially increasing data, an experimental task was conducted to test an individuals' ability to translate a graph of exponentially increasing data into real value quantities and extend their estimations by making comparisons. The results of the three experimental tasks provide guidelines for readers to actively choose which of many possible graphics to draw, according to some set of design choices, to ensure that their charts are effective.

Things that seem to be missing: 1) Cleveland and McGill - hierarchy of accuracy in plot objects. You can think of e.g. exponentials as a series of tangential angles, which might help with explaining underestimation? 2) some literature explaining the different levels of complexity between perception, reading information off of a graph, predicting information, etc. - you should find this if only because it grounds the whole experiment in psychological theory (but you don't have to find it for your prelim)

Chapter 2

Perception through lineups

2.1 Introduction

Previously, we saw how a data plot can be evaluated and treated as a visual statistic, a numerical function which summarizes the data. To evaluate a graph, we have to run our statistic through a visual evaluation - a person. If two different methods of presenting data result in qualitatively different results when evaluated visually, then we can conclude that the visual statistics are significantly different. Recent graphical experiments have utilized statistical lineups to quantify the perception of graphical design choices (Hofmann, Follett, Majumder, & Cook, 2012; Loy, Follett, & Hofmann, 2016; Loy, Hofmann, & Cook, 2017; VanderPlas & Hofmann, 2017). Statistical lineups provide an elegant way of combining perception and statistical hypothesis testing using graphical experiments (Majumder, Hofmann, & Cook, 2013; Vanderplas, Cook, & Hofmann, 2020; H. Wickham, Cook, Hofmann, & Buja, 2010). 'Lineups' are named after the 'police lineup' of criminal investigations where witnesses are asked to identify the criminal from a set of individuals. Similarly, a statistical lineup is a plot consisting of smaller panels; the viewer is asked to identify the plot of the real data from among a set of decoy null plots. A statistical lineup typically consists of 20 panels - 1 target panel and 19 null

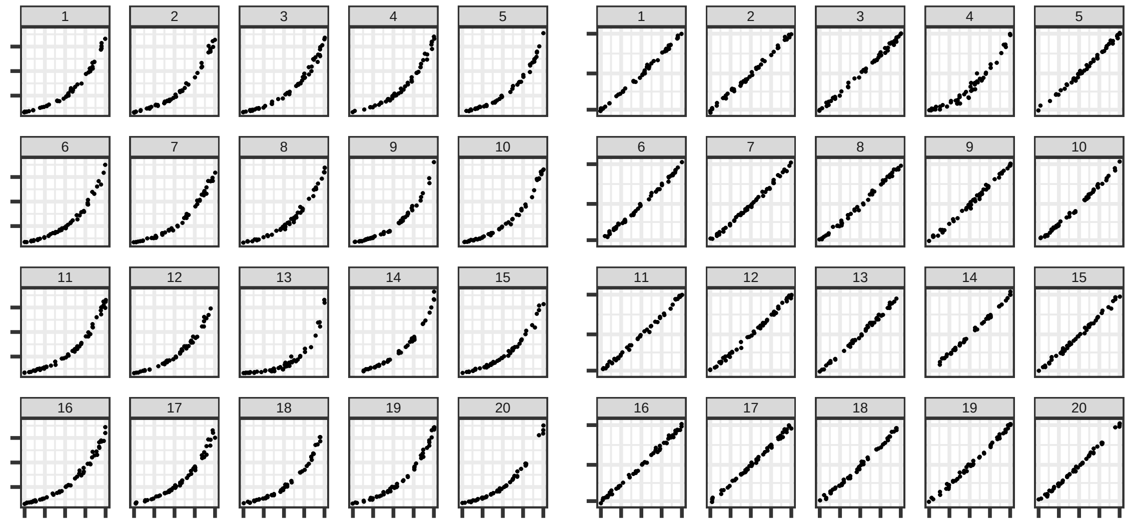

Figure 2.1: Lineup example

panels. Figure If the viewer can identify the target panel embedded within the set of null panels, this suggests that the real data is visually distinct from data generated under the null model. Fig. 2.1 provides example's of a statistical lineup's. The lineup plot on the left displays increasing exponential data on a linear scale with panel (2 x 5) + 3 as the target. The lineup plot on the right displays increasing exponential data on the log scale with panel 2 x 2 as the target. Crowd sourcing websites such as Amazon Mechanical Turk, Reddit, and Prolifc allow us to collect responses from multiple viewers.

While explicit graphical tests direct the participant to a specific feature of a plot to answer a specific question, implicit graphical tests require the user to identify both the purpose and function of the plot in order to evaluate the plots shown (Vanderplas, Cook, & Hofmann, 2020). Implicit graphical tests, such as lineups, have the advantage of simultaneously visually testing for multiple visual features including outliers, clusters, linear and nonlinear relationships. To lay a foundation for future exploration of the use of log scales, we begin with the most fundamental ability to identify differences in charts; this does not require that participants understand exponential growth, identify log scales, or have any mathematical training. Instead, we are simply testing the change in perceptual sensitivity resulting from visualization choices. The study in this chapter is conducted through visual inference and the use of statistical lineups to differentiate between exponentially increasing curves with differing levels of curvature, using linear and log scales.

2.2 Data Generation

In this study, both the target and null data sets were generated by simulating data from an exponential model; the models differ in the parameters selected for the null and target panels. In order to guarantee the simulated data spans the same domain and range of values, we implemented a domain constraint of $x \in [0, 20]$ and a range constraint of $y \in [10, 100]$ with $N = 50$ points randomly assigned throughout the domain and mapped to the y-axis using the exponential model with the selected parameters. These constraints provide some assurance that participants who select the target plot are doing so because of their visual perception differentiating between curvature or growth rate rather than different starting or ending values.

Data was simulated based on a three-parameter exponential model with multiplicative errors:

$$y_i = \alpha \cdot e^{\beta \cdot x_i + \epsilon_i} + \theta \tag{2.1}$$

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

The parameters α and θ are adjusted based on β and σ^2 to guarantee the range and domain constraints are met. The model generated $N = 50$ points $(x_i, y_i), i = 1, ..., N$

where x and y have an increasing exponential relationship. The heuristic data generation procedure is described below:

Algorithm 2.1.1: Paremeter Estimation

Input Parameters: domain $x \in [0, 20]$, range $y \in [10, 100]$, midpoint x_{mid}.

Output: estimated model parameters $\hat{\alpha}, \hat{\beta}, \hat{\theta}$

1. Determine the $y = -x$ line scaled to fit the assigned domain and range.
2. Map the values $x_{mid} - 0.1$ and $x_{mid} + 0.1$ to the $y = -x$ line for two additional points.
3. From the set points (x_k, y_k) for $k = 1, 2, 3, 4$, obtain the coefficients from the linear model $\ln(y_k) = b_0 + b_1 x_k$ to obtain starting values - $\alpha_0 = e^{b_0}, \beta_0 = b_1, \theta_0 = 0.5 \cdot \min(y)$
4. Using the `nls()` function from the `stats` package in Rstudio and the starting parameter values - $\alpha_0, \beta_0, \theta_0$ - fit the nonlinear model, $y_k = \alpha \cdot e^{\beta \cdot x_k} + \theta$ to obtain estimated parameter values - $\hat{\alpha}, \hat{\beta}, \hat{\theta}$.

Algorithm 2.1.2: Exponential Simulation

Input Paremeters: sample size $N = 50$, estimated parameters $\hat{\alpha}$, $\hat{\beta}$, and $\hat{\theta}$, σ standard deviation from the exponential curve.

Output Parameters: N points, in the form of vectors $\mathbf{x}$ and $\mathbf{y}$.

1. Generate $\tilde{x}_j, j = 1, ..., N \cdot \frac{3}{4}$ as a sequence of evenly spaced points in $[0, 20]$. This ensures the full domain of x is used, fulfilling the constraints of spanning the same domain and range for each parameter combination.

2. Obtain $\tilde{x}_i, i = 1, ...N$ by sampling $N = 50$ values from the set of $\tilde{x}_j$ values. This gaurantees some variability and potential clustring in the exponential growth curve disrupting the perception due to continuity of points.
3. Obtain the final x_i values by jittering $\tilde{x}_i$.
4. Calculate $\tilde{\alpha} = \frac{\hat{\alpha}}{e^{\sigma^2/2}}$. This ensures that the range of simulated values for different standard devaition parameters has an equal expected value for a given rate of change due to the non-constant variance across the domain.
5. Generate $y_i = \tilde{\alpha} \cdot e^{\hat{\beta} x_i + e_i} + \hat{\theta}$ where $e_i \sim N(0, \sigma^2)$.

2.3 Parameter Selection

For each level of difficulty, we simulated 1000 data sets of (x_{ij}, y_{ij}) points for $i = 1, ..., 50$ and $j = 1...10$. Each generated x_i point from *Algorithm 2.1.2* was replicated 10 times. Then the lack of fit statistic (LOF) was computed for each simulated data set by calculating the deviation of the data from a linear line. Plotting the density curves of the LOF statistics for each level of difficulty choice allows us to evaluate the ability of differentiating between the difficulty levels and thus detecting the target plot. In Fig. 2.2, we can see the densities of each of the three difficulty levels. While the LOF statistic provides us a numerical value for discriminating between the difficulty levels, we cannot directly relate this to the perceptual discriminability; it serves primarily as an approximation to ensure that we are testing parameters at several distinct levels of difficulty. Final parameter estimates are shown in Table 2.1.

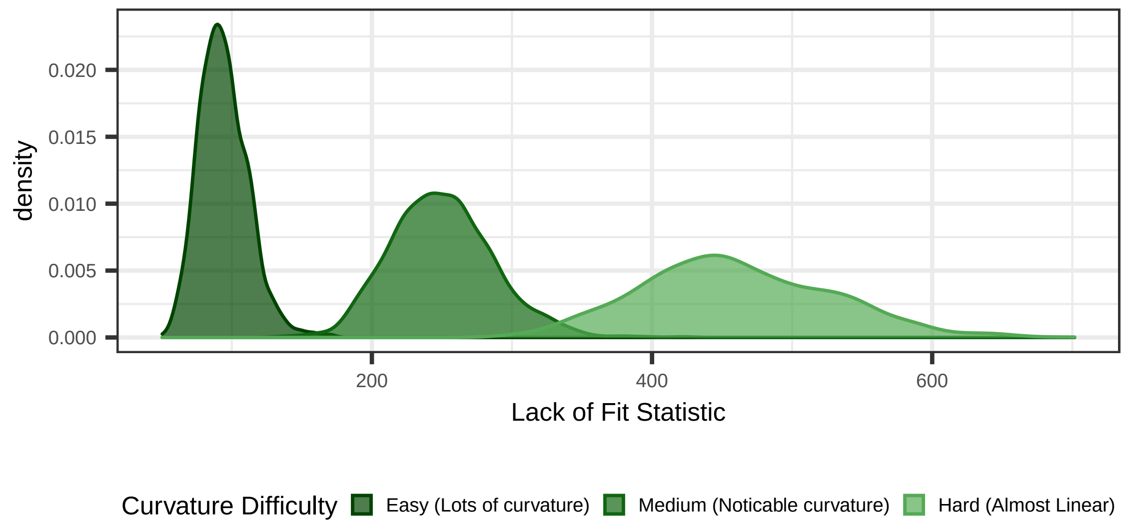

Figure 2.2: Lack of fit statistic density curves

Table 2.1: Lineup data simulation final parameters

	x_{mid}	$\hat{\alpha}$	$\tilde{\alpha}$	$\hat{\beta}$	$\hat{\theta}$	$\hat{\sigma}$
Easy	14.5	0.91	0.88	0.23	9.10	0.25
Medium	13.0	6.86	6.82	0.13	3.14	0.12
Hard	11.5	37.26	37.22	0.06	-27.26	0.05

2.4 Lineup Setup

Lineup plots were generated by mapping one simulated data set corresponding to difficulty level A to a scatter plot to be identified as the target plot while multiple simulated data sets corresponding to difficulty level B were individually mapped to scatter plots for the null plots. For example, a target plot with simulated data following an increasing exponential curve with obvious curvature is embedded within null plots with simulated data following an increasing exponential trend that is almost linear (i.e. Hard Null - Easy Target). By our constraints, the target plot and null plots will span a similar domain and range. There are a total of six (i.e. $3! \cdot 2!$) lineup parameter combinations. Two sets of each lineup parameter combination were simulated (total of 12 test data sets) and plotted on both the linear scale and the log scale (total of 24 test lineup plots). In addition, there are three parameter combinations which generate homogeneous "Rorschach" lineups, where all panels are from the same distribution. Each participant evaluated one of these lineups, but for simplicity, these evaluations are not described in this paper.

2.5 Study Design

Each participant was shown a total of thirteen lineup plots (twelve test lineup plots and one Rorschach lineup plot). Participants were randomly assigned one of the two replicate data sets for each of the six unique lineup parameter combinations. For each assigned test data set, the participant was shown the lineup plot corresponding to both the linear scale and the log scale. For the additional Rorschach lineup plot, participants were randomly assigned one data set shown on either the linear or the log scale. The order of the thirteen lineup plots shown was randomized for each participant.

Participants above the age of majority were recruited from Reddit's Visualization and Sample Size communities. Since participants recruited on Reddit were not compensated for their time, most participants have an interest in data visualization research. Previous literature suggests that prior mathematical knowledge or experience with exponential data is not associated with the outcome of graphical experiments (VanderPlas & Hofmann, 2015). Participants completed the experiment using a Shiny applet (https://shiny.srvanderplas.com/log-study/).

Participants were shown a series of lineup plots and asked to identify the plot that was most different from the others. On each plot, participants were asked to justify their choice and provide their level of confidence in their choice. The goal of this experimental task is to test an individuals ability to perceptually differentiate exponentially increasing trends with differing levels of curvature on both the linear and log scale.

2.6 Results

Participant recruitment through Reddit occurred over the course of two weeks during which 58 individuals completed 518 unique test lineup evaluations. Previous studies have found that results do not differ on lineup-related tasks between Reddit and e.g. Amazon Mechanical Turk (VanderPlas & Hofmann, 2017). Participants who completed fewer than 6 lineup evaluations were removed from the study (17 participants, 41 evaluations). The final data set included a total of 41 participants and 477 lineup evaluations. Each plot was evaluated by between 18 and 28 individuals (Mean: 21.77, SD: 2.29). In 67% of the 477 lineup evaluations, participants correctly identified the target panel.

Target plot identification was analyzed using the lme4 R package and glmer function (Bates, Mächler, Bolker, & Walker, 2015). Estimates and odds ratio comparisons were calculated using the emmeans R package (Lenth, 2021). Each lineup plot evaluated was assigned a value based on the participant response (correct = 1, not correct = 0). Define Y_{ijkl} to be the event that participant l correctly identifies the target plot for data set k with curvature j plotted on scale i. The binary response was analyzed using generalized linear mixed model following a binomial distribution with a logit link function following a row-column blocking design accounting for the variation due to participant and data set respectively as

$$\text{logit } P(Y_{ijk}) = \eta + \delta_i + \gamma_j + \delta\gamma_{ij} + s_l + d_k \tag{2.2}$$

Table 2.2: Lineup ANOVA table for fixed effects.

Effect	Chisq	Df	Pr..Chisq.
Curvature	19.96	5	0.00127
Scale	14.43	1	0.00015
Curvature x Scale	33.65	5	<0.0001

where

- η is the baseline average probability of selecting the target plot
- δ_i is the effect of the log/linear scale
- γ_j is the effect of the curvature combination
- $\delta\gamma_{ij}$ is the two-way interaction effect of the scale and curvature
- $s_l \sim N(0, \sigma^2_{\text{participant}})$, random effect for participant characteristics
- $d_k \sim N(0, \sigma^2_{\text{data}})$, random effect for data specific characteristics.

We assume that random effects for data set and participant are independent.

The analysis of variance table shown in Table 2.2 indicate a significant interaction between the curvature combination and scale. Variance due to participant and data set were estimated to be $\sigma^2_{\text{participant}} = 2.79$ (s.e. 1.67) and $\sigma^2_{\text{data}} = 0.44$ (s.e. 0.66) respectively.

On both the log and linear scales, the highest accuracy occurred in lineup plots where the target model and null model had large curvature differences (Easy Null - Hard Target; Hard Null - Easy Target). There is a decrease in accuracy on the linear scale when comparing a target plot with less curvature to null plots with more curvature (Easy Null - Medium Target; Medium Null - Hard Target). Best, Smith, & Stubbs (2007) found that accuracy of identifying the correct curve type was higher when nonlinear trends were presented indicating that it is hard to say something is linear (i.e. something has less curvature), but easy to say that it is not linear; our results concur with this observation. Overall, there are no significant differences in accuracy between curvature combinations when data is presented on a log scale indicating participants were consistent in their success of identifying the target panel on the log scale. Fig. 2.3 displays the estimated (log) odds ratio of successfully identifying the target panel on the log scale compared to the linear scale. The choice of scale has no impact if curvature differences are large (Hard Null - Easy Target; Easy Null - Hard Target). However, presenting data on the log scale makes us more sensitive to slight changes in curvature (Medium Null - Easy Target; Medium Null - Hard Target; Easy Null - Medium Target). An exception occurs when identifying a plot with curvature embedded in null plots close to a linear trend (Hard Null - Medium Target), again supporting the claim that it is easy to identify a curve in a bunch of lines but much harder to identify a line in a bunch of curves (Best, Smith, & Stubbs, 2007).

2.7 Discussion and Conclusion

The overall goal of this chapter is to provide basic research to support the principles used to guide design decisions in scientific visualizations of exponential data. In this study, we explore the use of linear and log scales to determine whether our ability to notice differences in exponentially increasing trends is impacted by the choice of scale. Our results indicated that when there was a large difference in curvature between the target plot and null plots, the choice of scale had no impact and participants accurately differentiated between the two curves on both the linear

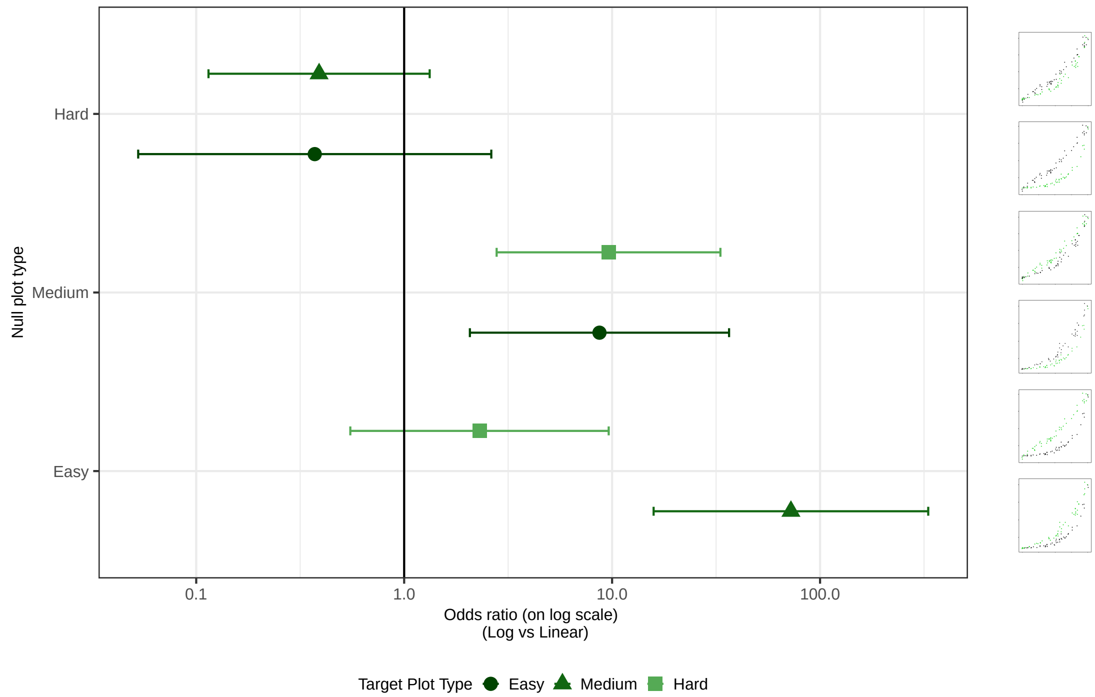

Figure 2.3: Lineups log(odds) results

and log scale. However, displaying exponentially increasing data on a log scale improved the accuracy of differentiating between models with slight curvature differences. An exception occurred when identifying a plot with curvature embedded in surrounding plots closely relating to a linear trend, indicating that it is easy to identify a curve in a group of lines but much harder to identify a line in a group of curves. The use of visual inference to identify these guidelines suggests that there are *perceptual* advantages to log scales when differences are subtle. What remains to be seen is whether there are cognitive disadvantages to log scales: do log scales make it harder to make use of graphical information?

Chapter 3

Prediction with you draw it

3.1 Introduction

In Chapter 2, a base foundation for future exploration of the use of log scales was established by evaluating participants ability to identify differences in charts through the use of lineups. This did not require that participants were able to understand exponential growth, identify log scales, or have any mathematical training; instead, it simply tested the change in perceptual sensitivity resulting from visualization choices. In order to determine whether there are cognitive disadvantages to log scales, we utilize interactive graphics to test an individual's ability to make predictions for exponentially increasing data. In this study, participants are asked to draw a line using their computer mouse through the exponentially increasing trend shown on both the log and linear scale.

3.1.1 Past Methodology

Initial studies in the 20th century explored the use of fitting lines by eye through a set of points (Finney, 1951; Mosteller, Siegel, Trapido, & Youtz, 1981). Common methods of fitting trends by eye involve maneuvering a string, black thread, or ruler until the fit is suitable, then drawing the line. In Finney (1951), it was of interest to

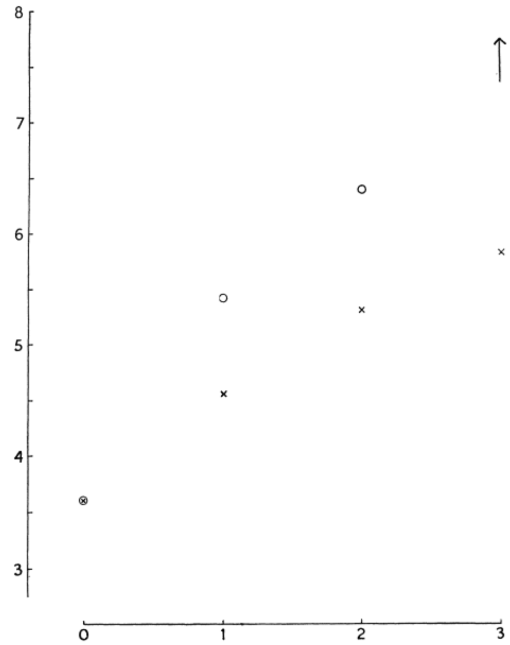

Figure 3.1: Subjective Judgement in Statistical Analysis (1951) Parallel Probits

determine the effect of stopping iterative maximum likelihood calculations after one iteration. Twenty-one scientists were recruited via postal mail and asked to "rule two lines" in order to judge by eye the positions for a pair of parallel probit regression lines in a biological assay Fig. 3.1. Results indicated that one cycle of iterations was sufficient based on the starting values provided by eye from the participants.

Thirty years later, Mosteller, Siegel, Trapido, & Youtz (1981), sought to understand the properties of least squares and other computed lines by establishing one systematic method of fitting lines by eye. The authors recruited 153 graduate students and post doctoral researchers in Introductory Biostatistics. Participants were asked to fit lines by eye to four sets of points Fig. 3.2 using an 8.5 x 11 inch transparency with a straight line etched completely across the middle. A latin square design with packets of the set of points stapled together in four different

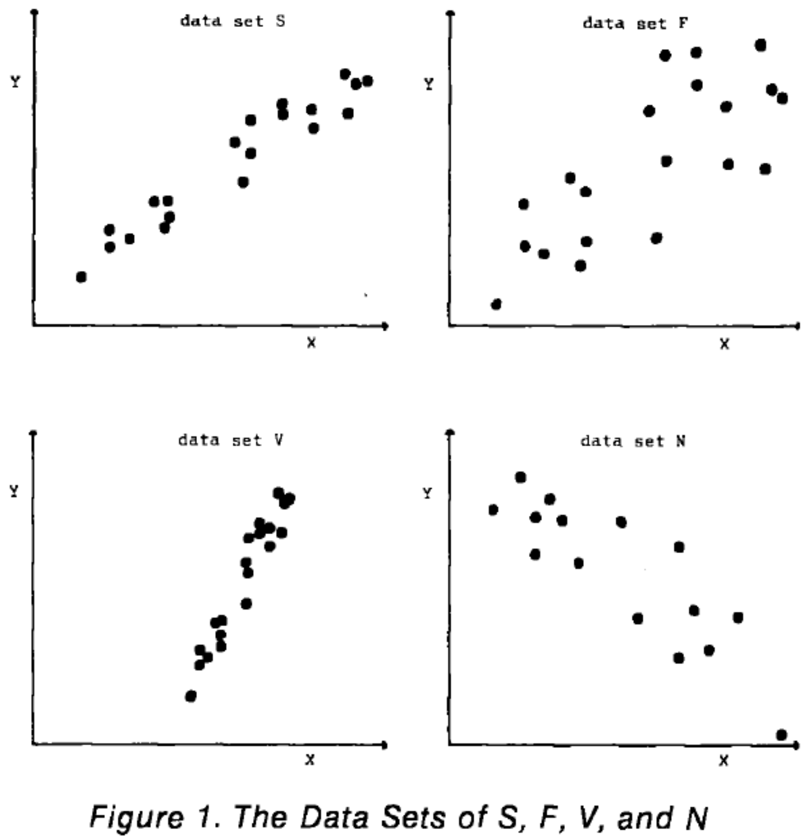

Figure 3.2: Eye Fitting Straight Lines (1981) Data Sets

orders was used in order to determine if there is an effect of order of presentation. It was found that order of presentation had no effect and that participants tended to fit the slope of the first principal component over the slope of the least squares regression line.

In 2015, the New York Times introduced an interactive feature, called You Draw It (Aisch, Cox, & Quealy, 2015; Buchanan, Park, & Pearce, 2017; Katz, 2017) Readers are asked to input their own assumptions about various metrics and learning how these assumptions relate to reality. The NY Times team utilizes Data Driven Documents (D3) that allows readers to predict these metrics through the use of drawing a line on their computer screen with their computer mouse. Fig. 3.3 is

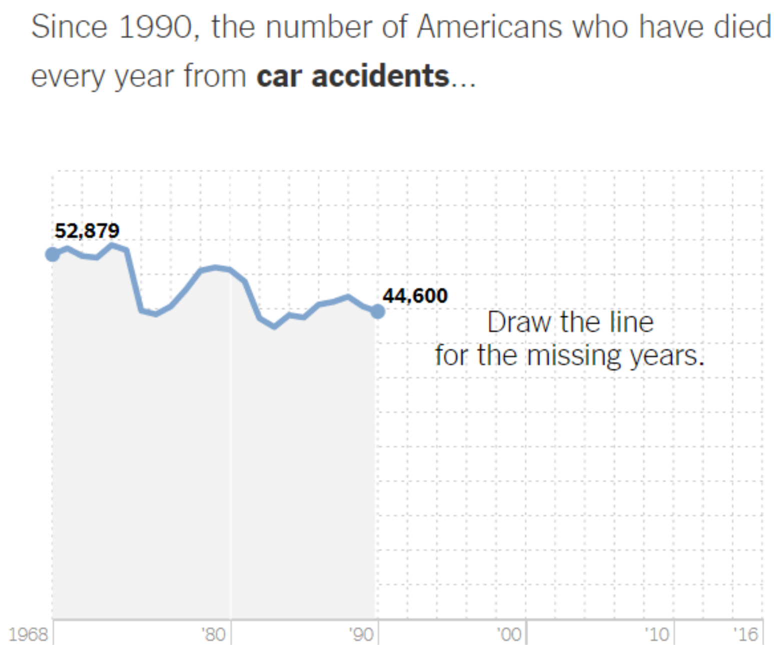

Figure 3.3: New York Times You Draw It Feature

one such example in which readers are asked to draw the line for the missing years providing what they estimate to be the number of Americans who have died every year from car accidents, since 1990. After the reader has completed drawing the line, the actual observed values are revealed and the reader may check their estimated knowledge against the actual reported data (Katz, 2017).

3.1.2 Data Driven Documents

Major news and research organizations such as the NY Times, FiveThirtyEight, Washing Post, and the Pew Research Center create and customize graphics with Data Driven Documents. In June 2020, the NY Times released an front page displaying figures that represent each of the 100,000 lives lost from the coronavirus pandemic at this point in time (Barry et al., 2020). During 2021 March Maddness, FiveThirtyEight created a roster-shuffling machine which allowed readers to build their own NBA contender through interactivity (Ryanabest, 2021). Data Driven

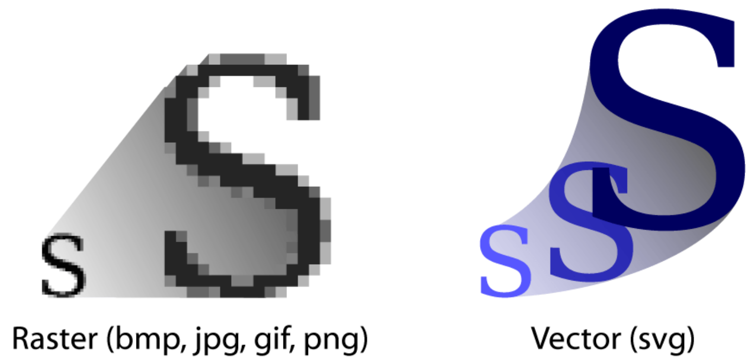

Figure 3.4: SVG vs Raster

Documents (D3) is an open-source JavaScript based graphing framework created by Mike Bostock during his time working on graphics at the NY Times. For readers familiar with R, it is notable to consider D3 in JavaScript equivalent to the ggplot2 package in R. Similar to geometric objects and style choices in ggplot2, the grammar of D3 also includes elements such as circles, paths, and rectangles with choices of attributes and styles such as color and size. Data Driven Documents depend on Extensible Markup Langage (XML) to generate graphics and images by binding objects and layers to the plotting area as Scalable Vector Graphics (SVG) in order to preserve the shapes rather than the pixels (Fig. 3.4) CITE: https://martech.zone/vecteezy-svg-editor-online/. Advantages of using D3 include animation and allowing for movement and user interaction such as hovering, clicking, and brushing.

A challenge of working with D3 is the environment necessary to display the graphics and images. The r2d3 package in R provides an efficient integration of D3 visuals and R by displaying them in familiar HTML output formats such as RMarkdown or Shiny applications (Luraschi & Allaire, 2018). The creator of the graphic applies D3.js code to visualize data which has previously been processed

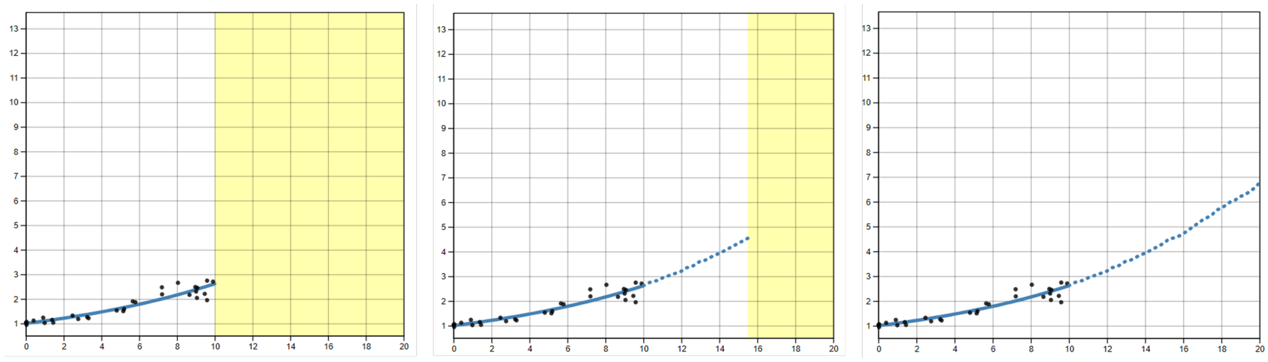

Figure 3.5: You Draw It Example

within an R setting.

```
r2d3(data = data, script = "d3-source-code.js",
    d3_version = "5")
```

The example R code illustrates the structure of the r2d3 function in R which includes specification of a data frame in R (converted to a JSON file), the D3.js source code file, and the D3 version that accompanies the source code. A default SVG container for layering elements is then generated by the r2d3 function which renders the plot using the source code. Appendix A outlines the development of the you draw it study interactive plots through the use of r2d3 and R shiny applications. Fig. 3.5 provides an example of a you draw it interactive plot. The first frame shows what the participant sees along with the following prompt: *Use your mouse to fill in the trend in the yellow box region.* Next, the yellow box region moves along as the participant draws their trend-line until the yellow region disappears indicating the participant has filled in the entire domain.

3.2 Study Design

This chapter contains two sub studies which aim to establish you draw it as a tool for measuring predictions of trends and then apply this as a method of testing graphics. Mosteller, Siegel, Trapido, & Youtz (1981) was replicated as part of the study data collection in order to validate you draw it as a method for testing graphics, referred to as Eye Fitting Straight Lines in the Modern Era. This method is then used to test an individual's ability to make predictions for exponentially increasing data on both the log and linear scales, referred to as Prediction of Exponential Trends. Data for both sub-studies were collected in conjunction with one another over the same study participant sample.

A total of 6 data sets - 4 Eye Fitting Straight Lines in the Modern Era and 2 Prediction of Exponential Trends - are generated for each individual the start of the experiment. The 2 data sets corresponding to to the data used in the Prediction of Exponential Trends are then plotted a total of 4 times each by truncating the points at both 50% and 75% of the domain as well as on both the log and linear scales for a total of 8 task plots. Participants in the study are first shown 2 you draw it plot practice plots followed by 12 you draw it task plots. The order of all 12 task plots were randomly assigned for each individual in a completely randomized design where users saw the 4 task plots from the Eye Fitting Straight Lines in the Modern Era simulated data interspersed with the 8 task plots from the Prediction of Exponential Trends simulated data.

3.3 Eye Fitting Straight Lines in the Modern Era

Finney (1951) and Mosteller, Siegel, Trapido, & Youtz (1981) use methods such as using a ruler, string, or transparency sheet to fit straight lines through a set of points. This section replicates the study found in Mosteller, Siegel, Trapido, & Youtz (1981) in order to establish you draw it as a tool and method for testing graphics.

3.3.1 Data Simulation

All data processing was conducted in R before being passed to the D3.js code. A total of $N = 30$ points $(x_i, y_i), i = 1, ...N$ were generated for $x \in [x_{min}, x_{max}]$ where x and y have a linear relationship. Data was simulated based on linear model with additive errors:

$$y_i = \beta_0 + \beta_1 \cdot x_i + e_i \tag{3.1}$$

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

The parameters β_0 and β_1 are selected to replicate (Mosteller, Siegel, Trapido, & Youtz, 1981) with e_i generated by rejection sampling in order to guarantee the points shown align with that of the fitted line. An ordinary least squares regression is then fit to the simulated points in order to obtain the best fit line and fitted values in 0.25 increments across the domain, $(x_k, \hat{y}_{k,OLS}), k = 1, ..., 4 \cdot x_{max} + 1$. The function then outputs a list of point data and line data both indicating the parameter identification, x value, and corresponding simulated or fitted y value. The data simulation procedure is described below:

Algorithm: Eye Fitting Straight Lines in the Modern Era Data Generation

In parameters: $y_{\bar{x}}$ for calculating the y-intercept, β_0; slope β_1; standard deviation from line σ; sample size $N = 30$; domain x_{min} and x_{max}; fitted value increment $x_{by} = 0.25$.

Table 3.1: Eye Fitting Straight Lines in the Modern Era simulation model parameters

Parameter Choice	$y_{\bar{x}}$	β_1	σ
F	3.90	0.66	1.98
N	4.11	-0.70	2.50
S	3.88	0.66	1.30
V	3.89	1.98	1.50

Out: List of point data and line data each indicating the parameter identification, x value, and corresponding simulated or fitted y value.

1. Randomly select and jitter N = 30 x-values along the domain, $x_{i=1:N} \in [x_{min}, x_{max}]$.
2. Determine the y-intercept, β_0, at x = 0 from the provided slope (β_1) and y-value at the mean of x ($y_{\bar{x}}$) using point-slope equation of a line.
3. Generate "good" errors, $e_{i=1:n}$ based on $N(0, \sigma)$ by setting a constraint requiring the mean of the first N/3 errors $< |2\sigma|$.
4. Simulate point data based on $y_i = \beta_0 + \beta_1 x_i + e_i$
5. Obtain ordinary least squares regression coefficients, $\hat{\beta}_0$ and $\hat{\beta}_1$, for the simulated point data using the `lm` function in the base stats R package.
6. Obtain fitted values every 0.25 increment across the domain from the ordinary least squares regression $\hat{y}_{k,OLS} = \hat{\beta}_{0,OLS} + \hat{\beta}_{1,OLS} x_k$.
7. Output data list of point data and line data each indicating the parameter identification, x value, and corresponding simulated or fitted y value.

Simulated model equation parameters were selected to reflect the four data sets (F, N, S, and V) used in Mosteller, Siegel, Trapido, & Youtz (1981) **??**. Parameter choices F, N, and S simulated data across a domain of 0 to 20. Parameter choice F produces a trend with a positive slope and a large variance while N has a negative

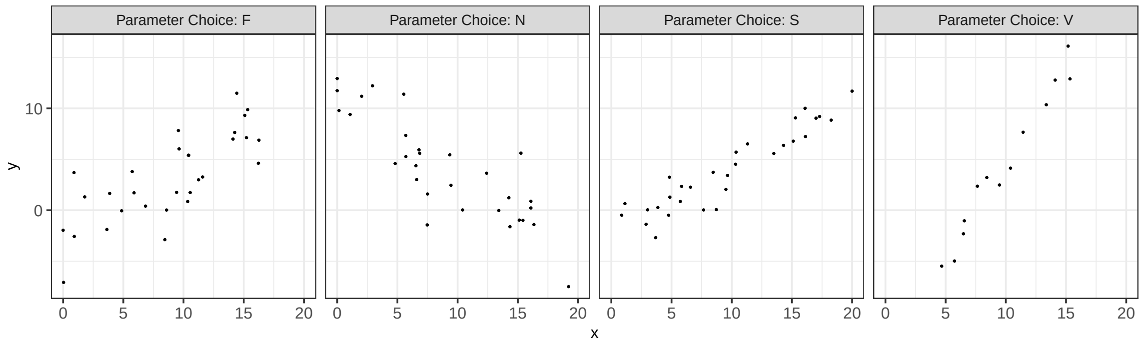

Figure 3.6: Eye Fitting Straight Lines in the Modern Era Simulated Data Example

slope and a large variance. In comparison, S shows trend with a postie slope but a small variance and V yields a steep positive slope over the domain of 4 to 16. Fig. 3.6 illustrates an example of simulated data for all four parameter choices. Within the interactive plot code, the aspect ratio defining the x to y axis ratio was set to 1 with a consistent y range buffer of 10% to allow for users to draw outside of the provided range.

3.3.2 Results

Participants were recruited through Twitter, Reddit, and direct email during May 2021. A total of X individuals completed Y unique you draw it task plots. All completed you draw it task plots were included in the analysis.

In addition to the participant drawn points, (x_k, y_{drawn}), and the ordinary least squares (OLS) regression fitted values, $(x_k, y_{k,OLS})$, a regression equation based on the first principal component (PCA) was used to calculate fitted values, $x_k, y_{k,PCA}$. For each set of simulated data and parameter combination, the PCA regression equation was determined by using the princomp function in the base R stats package to obtain the rotations of the first principal component for x and y. The estimated slope, $\hat{\beta}_{1,PCA}$, is determined by the ratio of the y rotation and x rotation

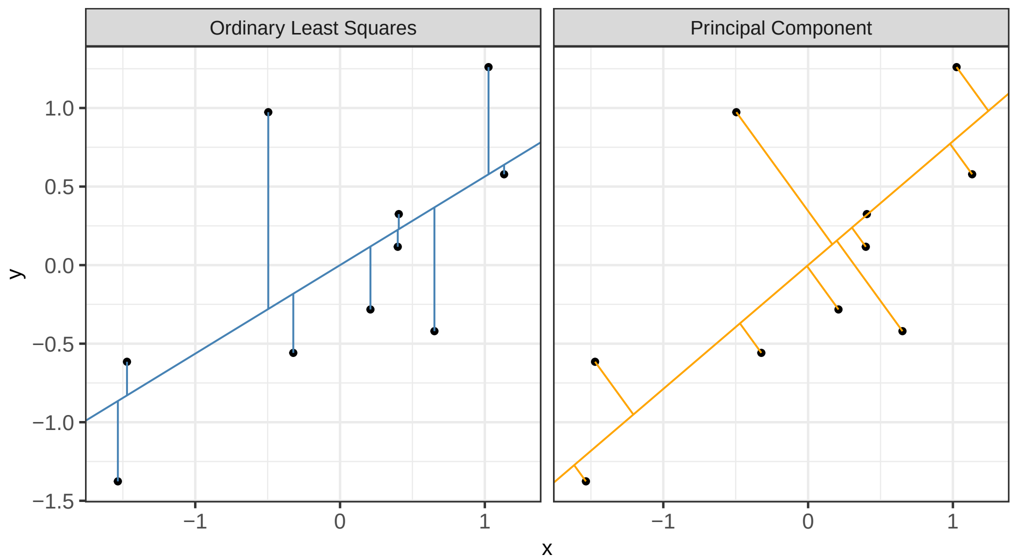

Figure 3.7: OLS vs PCA Regression Lines

of the first principal component with the y-intercept, $\hat{\beta}_{0,PCA}$ calculated by the point-slope equation of the line using the mean of of the simulated points, $(\bar{x}_i, \bar{y}_i)$. Fitted values, $y_{k,PCA}$ are then obtained every 0.25 increment across the domain from the PCA regression equation, $\hat{y}_{k,PCA} = \hat{\beta}_{0,PCA} + \hat{\beta}_{1,PCA}x_k$. Fig. 3.7 illustrates the difference between an OLS regression equation which minimizes the vertical distance and a regression equation with a slope calculated by the first principal component which minimizes the smallest distance (both horizontal and vertical direction).

For each participant, the final data set used for analysis contains $x_{ijk}, y_{ijk,drawn}, \hat{y}_{ijk,OLS}$, and $\hat{y}_{ijk,PCA}$ for parameter combination $i = S, V, F, N$, j = $1, ...N_{participant}$, and x_k value $k = 1, ..., 4 \cdot x_{max} + 1$. Using both a linear mixed model and a generalized additive mixed model, comparisons of vertical residuals in relation to the OLS fitted values ($e_{k,ols} = y_{k,drawn} - \hat{y}_{k,ols}$) and PCA fitted values ($e_{k,pca} = y_{k,drawn} - \hat{y}_{k,pca}$) were made across the domain of x. **??** displays an example of all three fitted trend lines for parameter choice F.

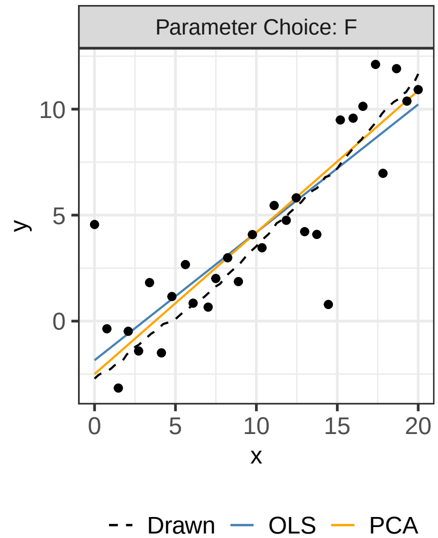

Figure 3.8: Eye Fitting Straight Lines in the Modern Era Example

Using the lmer function in the lme5 package (Bates, Mächler, Bolker, & Walker, 2015), a linear mixed model (LMM) is fit separately to the OLS and PCA residuals, 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:

$$y_{ijk,drawn} - y_{ijk,fit} = e_{ijk,fit} = [\gamma_0 + \alpha_i] + [\gamma_1 x_{ijk} + \gamma_{2i} x_{ijk}] + p_j + \epsilon_{ijk} \tag{3.2}$$

where

- γ_0 is the overall intercept
- α_i is the effect of the parameter combination on the intercept
- γ_1 is the overall slope for x
- γ_{2i} is the effect of the parameter combination on the slope
- $p_j \sim N(0, \sigma^2_{participant})$ is the participant error due to participant variation

- $\epsilon_{ij} \sim N(0, \sigma^2)$ is the residual error.

Eliminating the linear trend constraint, the bam function in the mgcv package (S. N. Wood, 2003, 2004, 2011; S. N. Wood, 2017; S. N. Wood, N., Pya, & S"afken, 2016), a generalized additive mixed model (GAMM) is fit separately to the OLS and PCA residuals to allow for estimation of smoothing splines. Parameter choice was treated as a fixed effect with no extimated 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 the each fit (OLS and PCA) residuals is given by:

$$y_{ijk,drawn} - y_{ijk,fit} = e_{ijk,fit} = \alpha_i + s_i(x_{ijk}) + p_j + s_j(x_{ijk}) \tag{3.3}$$

where

- α_i is the intercept for the parameter choice i
- s_i is the smoothing spline for the i^{th} parameter combination
- $p_j \sim N(0, \sigma^2_{participant})$ is the error due to participant variation
- s_j is the random smoothing spline for each participant.

Fig. 3.9 and Fig. 3.10 show the estimated trend from both the LMM and GAMM. These results indicate participants provided a trend line closer in relation to the estimated PCA regression line than the estimated OLS regression line. These results are consistent to those found in (Mosteller, Siegel, Trapido, & Youtz, 1981).

In addition to fitting trend lines over the residuals between the drawn values and fitted values, a OLS sum of squares and PCA sums of squares measure was

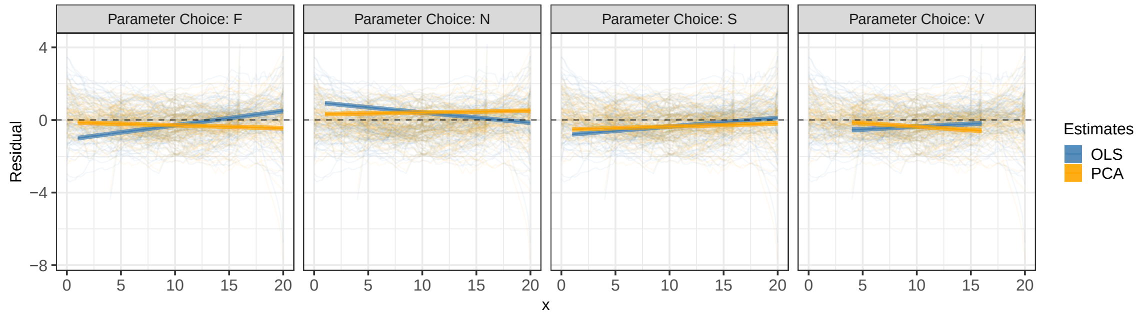

Figure 3.9: Eye Fitting Straight Lines in the Modern Era LMM results

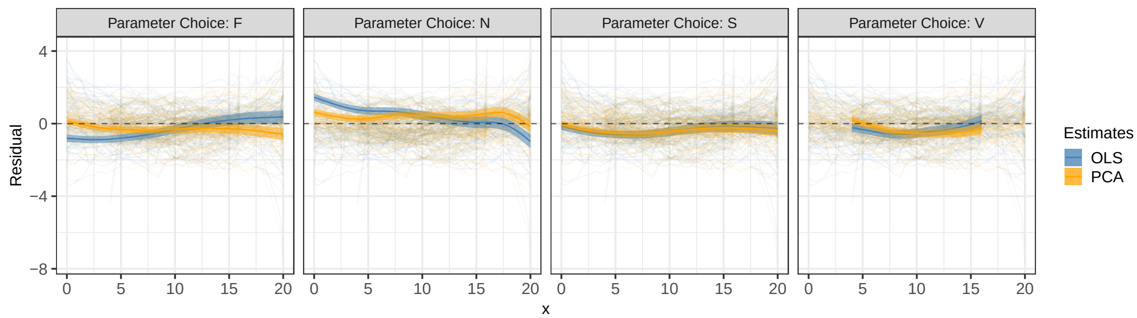

Figure 3.10: Eye Fitting Straight Lines in the Modern Era GAMM results

calculated for each you draw it plot. Sums of squares between fits were compared using lmer (Bates, Mächler, Bolker, & Walker, 2015) to run a linear mixed model (LMM) with a log transformation. Parameter combination, fit, and the interaction between parameter combination and fit were treated as fixed effects with a random participant effect. Define SS_{ijk} as the sums of squares for parameter choice $i = 1, 2, 3, 4$, fit $j = 1, 2$, and participant $k = 1, ..., N_{participant}$. The LMM equation is given by:

$$\log\left(SS_{ijk}\right) = \alpha_i + \beta_j + \alpha\beta_{ij} + p_j + \epsilon_{ij} \tag{3.4}$$

- α_i denotes the effect of the i^{th} parameter (S, F, V, N)
- β_j denotes the effect of the j^{th} fit (OLS, PCA)

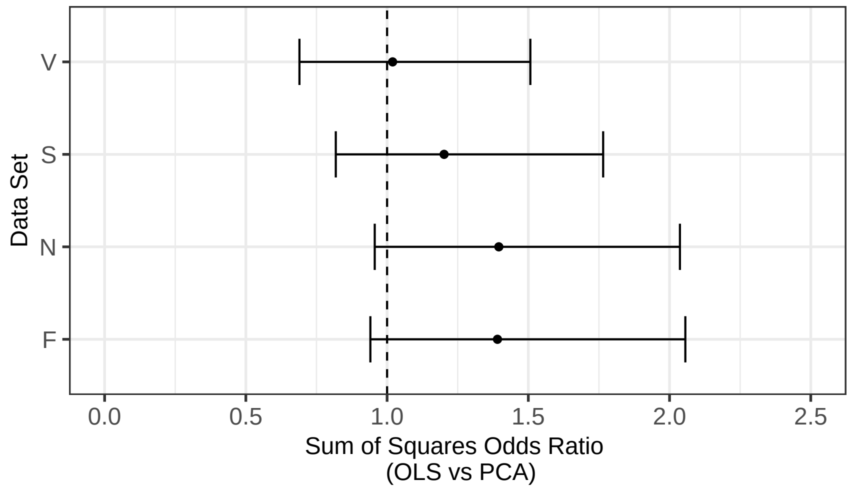

Figure 3.11: Eye Fitting Straight Lines in the Modern Era Sum of Squares Results

- $\alpha\beta_{ij}$ denotes the interaction between the parameter combination and fit
- $p_j \sim N(0, \sigma^2_{participant})$ is the error due to participant variation
- $\epsilon_{ij} \sim N(0, \sigma^2)$ is the residual error.

Fig. 3.11 displays estimated odds ratios between the OLS fit and PCA fit for each parameter choice. While there is no significant effect of fit for any parameter choices, there is indication of the trend found above.

3.4 Prediction of Exponential Trends

3.4.1 Data Simulation

- aspect ratio = 1; linear = T/F; start drawing at x = 10; xrange = (0,20), yrange = range(points)*c(0.5,2)

Algorithm 3.2: Exponential Data Generation

In parameters: `beta, sd, points_choice = "partial", points_end_scale, N = 30, xmin = 0, xmax = 20, xby = 0.25`

Out: data list of point data and line data

1. Randomly select and jitter N = 30 x-values along the domain.
2. Generate "good" errors based on N(0,sd). Set constraint of the mean of the first N/3 = 10 errors less than |2*sd|
3. Simulate point data based on: `y = exp(x*beta + errorVals)`
4. Obtain starting value for beta: `lm(log(y) ~ x, data = point_data)`
5. Use NLS to fit a better line to the point data: `nls(y ~ exp(x*beta), data = point_data, ...)`
6. Simulate nonlinear least squares line data: `y = exp(x*betahat)`
7. Output data list of point data and line data

3.4.2 Parameter Selection

- Visit You Draw It Development - parameter selection for examples.
- Exponential (Linear/Log): 2 x 2 x 2 Factorial
  - Beta: 0.1 (sd. 0.09); 0.23 (0.25)
  - Points End: 0.5; 0.75
  - Scale: Linear; Log

3.4.3 Results

- advocate smoothing of scatterplots to assist in detecting the shape of the point cloud in situations where the error in the data is substantial, or where the density of points changes along the abscissa Cleveland & McGill (1984)

- Twitter/Reddit/Direct Email Pilot Study (05/03/2021): Exponential Prediction
- https://shiny.srvanderplas.com/you-draw-it/
- Twitter/Reddit/Direct Email Pilot Study (05/03/2021): Exponential Prediction

3.5 Discussion and Conclusion

Chapter 4

Numerical Translation and Estimation

4.1 Introduction

- Such complex inferential processes involve quantitatively transforming the information in the display (e.g., mentally transforming from a linear to logarithmic scale or calculating the difference between two or more data points; Cleveland, 1984, 1985).
- Estimation Questions: Timmers & Wagenaar (1977)
  - If nothing will stop this decreasing trend, what is your prediction for 1980? (Group 1)
  - When will the process reach … (a certain level)? (Group 2)

Chapter 5

Conclusion

Appendix A

You Draw It Setup with Shiny

Interactive plots for the study were created using the r2d3 package and integrating D3 source code with an R shiny application. In this section I outline the data simulation process conducted in R, constructing interactive plots with D3.js code and R shiny

Fig. A.1

A.1 You draw it D3.js source code

```
# Define variable margins
const margin = {left: 55,
                right: 10,
                top: options.title ? 40: 10,
                bottom: options.title? 25: 55};

# Define variable default line attributes
const default_line_attrs = Object.assign({
  fill: "none",
  stroke: options.data_line_color || 'steelblue',
  strokeWidth: 4,
```

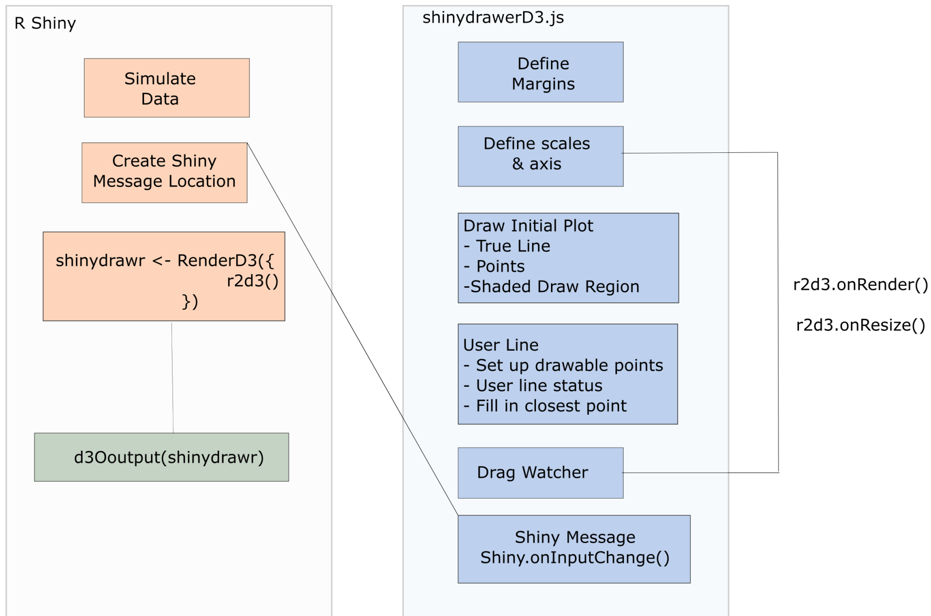

Figure A.1: Interactive plot development

```
  strokeLinejoin: "round",
  strokeLinecap: "round",
}, options.line_style);

# defines a changing variable called state
let state = Object.assign({
  line_data: data.line_data,
  point_data: data.point_data,
  svg: svg.append('g').translate([margin.left, margin.top]).attr("class", "wrapper"),
  w: height*options.aspect_ratio - margin.left - margin.right,
  h: height - margin.top - margin.bottom,
}, options);

# To distinguish between code that runs at initialization-time only and
# code that runs when data changes, organize your code so that the code
# which responds to data changes is contained within the r2d3.onRender()
r2d3.onRender(function(data, svg, width, height, options) {
```

```
  state.line_data = data.line_data;
  state.point_data = data.point_data;

  state = Object.assign(state, options);
  state.options = options;
  state.w = height*options.aspect_ratio;

  start_drawer(state);

});

# An explicit resize handler
r2d3.onResize(function(width, height, options) {
  state.w = height*state.options.aspect_ratio - margin.left - margin.right;
  state.h = height - margin.top - margin.bottom;

  start_drawer(state, reset = false);

});

# Main function that draws current state of viz
function start_drawer(state, reset = true){
  const scales = setup_scales(state);

  if(!state.free_draw){
    draw_true_line(state, scales, state.draw_start);
  }

  # Cover hidden portion with a rectangle (disabled)
  # const line_hider = setup_line_hider(state.svg, state.draw_start, scales);
```

```
# if we reset (these are points that can be drawn) remove what user has drawn.
if(reset){
  state.drawable_points = setup_drawable_points(state);
}

# if we have points, we draw user's line.
draw_user_line(state, scales);
draw_rectangle(state, scales);
draw_finished_line(state, scales, state.draw_start);

  # draw points for initial portion
if(state.points != "none"){
  draw_points(state, scales);
}

# invert from pixle to data scale when they draw their points
const on_drag = function(){
  const drag_x = scales.x.invert(d3.event.x);
  const drag_y = scales.y.invert(d3.event.y);
  fill_in_closest_point(state, drag_x, drag_y);
  draw_user_line(state, scales);
  draw_rectangle(state, scales);
};

# line_status is set by draw watcher - get user status line
# if some points missing - in progress
# complete line - done
const on_end = function(){
  # Check if all points are drawn so we can reveal line
  const line_status = get_user_line_status(state);
```

```
  if(line_status === 'done'){
    # User has completed line drawing
    # if(state.show_finished) line_hider.reveal();
    # if(!state.free_draw)  line_hider.reveal();
    if(state.show_finished){
      draw_finished_line(state, scales, state.draw_start);
    }
    if(state.shiny_message_loc){
      # Make sure shiny is available before sending message
      if(typeof Shiny !== 'undefined'){
        # Send drawn points off to server
        Shiny.onInputChange(
          state.shiny_message_loc,
          state.drawable_points.map(d => d.y)
        );
      } else {
        console.log(`Sending message to ${state.shiny_message_loc}`);
      }
    }
  }
};

setup_draw_watcher(state.svg, scales, on_drag, on_end);

# Do we have a title?
  if(state.title){
    state.svg.append('text')
    .at({
      y: -margin.top/2,
      dominantBaseline: 'middle',
      fontSize: '1.5rem',
```

```
      })
      .style('font-family', system_font)
      .text(state.title);
    }
}

function setup_drawable_points({line_data, free_draw, draw_start}){
  if(free_draw){
    return line_data.map(d => ({x: d.x, y: null}));
  } else {
    return line_data
    .filter(d => d.x >= draw_start)
    .map((d,i) => ({
      x: d.x,
      y: i === 0 ? d.y: null
    }));
  }
}

function get_user_line_status({drawable_points, free_draw}){
  const num_points = drawable_points.length;
  const num_filled = d3.sum(drawable_points.map(d => d.y === null ? 0: 1));
  const num_starting_filled = free_draw ? 0: 1;
  if(num_filled === num_starting_filled){
    return 'unstarted';
  } else if(num_points === num_filled){
    return 'done';
  } else {
    return 'in_progress';
  }
}
```

```
# Draw visable portion of line
function draw_true_line({svg, line_data, draw_start}, scales){
  var df = line_data.filter(function(d){ return d.x<=draw_start})
  state.svg.selectAppend("path.shown_line")
  .datum(df)
  .at(default_line_attrs)
  .attr("d", scales.line_drawer);
}

function draw_points({svg, point_data, points_end, points}, scales){
    if(points == "partial"){
      var df = point_data.filter(function(d){return (d.x<=points_end)});
    } else {
      var df = point_data;
    }
  const dots = state.svg.selectAll("circle").data(df)
  dots
    .enter().append("circle")
    .merge(dots)
    .attr("cx", d => scales.x(d.x))
    .attr("cy", d => scales.y(d.y))
    .attr("r", 2)
    .style("fill", "black")
    .style("opacity", 0.8)
    .style("stroke", "black")
}

function draw_rectangle({svg, drawable_points, line_data, draw_start, width, height, free_draw, x
    if(get_user_line_status(state) === 'unstarted'){
      if(free_draw){
```

```
        var xmin = line_data[0].x
        var len  = line_data.length - 1
        var xmax = line_data[len].x
        var drawSpace_start = scales.x(xmin)
        var drawSpace_end   = scales.x(xmax)
      } else {
        var drawSpace_start = scales.x(draw_start)
        var drawSpace_end   = state.w
      }
  } else {
    if(get_user_line_status(state) === 'done'){
      var drawSpace_start = scales.x(100)
      var drawSpace_end = scales.x(110)
    } else {
      var drawSpace_start = drawable_points.filter(function(d){return (d.y === null)});
      var xmin = df[0].x - x_by;
      var len  = line_data.length - 1
      var xmax = line_data[len].x
      var drawSpace_start = scales.x(xmin)
      var drawSpace_end   = scales.x(xmax)
    }
  }
  const draw_region = state.svg.selectAppend("rect");
  draw_region
    .attr("x", drawSpace_start)
    .attr("width",drawSpace_end - drawSpace_start)
    .attr("y", 0)
    .attr("height", state.h)
    //.style("fill", "#e0f3f3")
    .style("fill-opacity", 0.4)
    .style("fill", "rgba(255,255,0,.8)")
```

```
}

function draw_user_line(state, scales){
  const {svg, drawable_points, drawn_line_color} = state;
  const user_line = state.svg.selectAppend("path.user_line");
  # Only draw line if there's something to draw.
  if(get_user_line_status(state) === 'unstarted'){
    user_line.remove();
    return;
  }
  # Draws the points the user is drawing with their mouse
  user_line
      .datum(drawable_points)
      .at(default_line_attrs)
      .attr('stroke', drawn_line_color)
      .attr("d", scales.line_drawer)
      .style("stroke-dasharray", ("1, 7"));
}

function draw_finished_line({svg, line_data, draw_start, free_draw}, scales){
  if(!free_draw){
    var df = line_data.filter(function(d){ return d.x >= draw_start})
  } else {
    var df = line_data
  }
  const finished_line = state.svg.selectAppend("path.finished_line")
    # Only draw line if there's something to draw.
  if(get_user_line_status(state) === 'unstarted'){
    finished_line.remove();
    return;
  }
```

```
  finished_line
  .datum(df)
  .at(default_line_attrs)
  .attr("d", scales.line_drawer)
  .attr("opacity", 0.5)
}

# from state we need drawable_points
function fill_in_closest_point({drawable_points, pin_start, free_draw}, drag_x, drag_y){
  # find closest point on data to draw
  let last_dist = Infinity;
  let current_dist;
  # If nothing triggers break statement than closest point is last point
  let closest_index = drawable_points.length - 1;
  const starting_index = free_draw ? 0 : (pin_start ? 1: 0);
  # for loop to check where closest point to where I am
  for(i = starting_index; i < drawable_points.length; i++){
    current_dist = Math.abs(drawable_points[i].x - drag_x);
    # If distances start going up we've passed the closest point
if(last_dist - current_dist < 0) {
  closest_index = i - 1;
  break;
}
last_dist = current_dist;
}

drawable_points[closest_index].y = drag_y;
}

function setup_draw_watcher(svg, scales, on_drag, on_end){
  svg.selectAppend('rect.drag_watcher')
```

```
  .at({
    height: scales.y.range()[0],
    width: scales.x.range()[1],
    fill: 'grey',
    fillOpacity: 0,
  })
  .call(
    d3.drag()
    .on("drag", on_drag)
    .on("end", on_end)
  );
}

function add_axis_label(label, y_axis = true){
  const bump_axis = y_axis ? 'x': 'y';
  const axis_label_style = {
    [bump_axis]: bump_axis == 'y' ? 3: -2,
    textAnchor: 'end',
    fontWeight: '500',
    fontSize: '0.9rem',
  };
  return g => {
    let last_tick = g.select(".tick:last-of-type");
    const no_ticks = last_tick.empty();
    if(no_ticks){
      last_tick = g.append('g')
      .attr('class', 'tick');
    }
    last_tick.select("line").remove();
    last_tick.selectAppend("text")
    .at(axis_label_style)
```

```
    .html(label);
  };
}


# Setup scales for visualization
function setup_scales(state){
  # multi-assign: x_range, etc. coming from options
  const {w, h, line_data, x_range, y_range, x_name, y_name, linear} = state;
  # convert x from data scale to pixle scale
  const x = d3.scaleLinear()
  .domain(x_range || d3.extent(line_data, d => d.x))
  .range([0, w]);
  //console.log(linear);
  if (linear == 'true') {
    //console.log('in linear block');
    # converts from data linear scale to pixle scale
    var y = d3.scaleLinear()
    .domain(y_range || d3.extent(line_data, d => d.y))
    .range([h, 0]);
  } else {
    //console.log('in log block');
    # converts from data log scale to pixle scale
    var y = d3.scaleLog()
    .domain(y_range || d3.extent(line_data, d => d.y))
    .range([h, 0]).base(10);
  }

  const xAxis = d3.axisBottom().scale(x).tickSizeOuter(0);
  const yAxis = d3.axisLeft().scale(y).tickFormat(d3.format(".4")).tickSizeOuter(0);
  const xAxisGrid = d3.axisBottom().scale(x).tickSize(-h).tickFormat('');
  const yAxisGrid = d3.axisLeft().scale(y).tickSize(-w).tickFormat('');
```

```
state.svg.selectAll("g.x_grid").remove()
state.svg.selectAll("g.y_grid").remove()
# could call axis-grid "fred"
state.svg.selectAll("g.axis-grid").remove()

state.svg.selectAll("path.shown_line").remove()
state.svg.selectAll("circle").remove()

state.svg.selectAppend("g.x_grid")
.attr('class', 'x axis-grid')
.translate([0,h])
.call(xAxisGrid);

state.svg.selectAppend("g.y_grid")
.attr('class', 'y axis-grid')
.call(yAxisGrid);

state.svg.selectAll(".axis-grid .tick")
.style("stroke", "light-grey")
.style("opacity", ".3");

state.svg.selectAppend("g.x_axis")
.translate([0,h])
.call(xAxis);

state.svg.selectAppend("g.y_axis")
.call(yAxis);

const line_drawer = d3.line()
.defined(d => d.y !== null)
```

```
  .x(d => x(d.x))
  .y(d => y(d.y));

  return {
    x,
    y,
    line_drawer,
  };
}
```

References

10 Aisch, G., Cox, A., & Quealy, K. (2015, May). You draw it: How family income predicts children's college chances. *The New York Times*. The New York Times. Retrieved from https://www.nytimes.com/interactive/2015/05/28/upshot/you-draw-it-how-family-income-affects-childrens-college-chances.html

Amer, T. (2005). Bias due to visual illusion in the graphical presentation of accounting information. *Journal of Information Systems*, *19*(1), 1–18.

Barry, D., Buchanan, L., Cargill, C., Daniel, A., Delaquérière, A., Gamio, L., … al., et. (2020, May). Remembering the 100,000 lives lost to coronavirus in america. *The New York Times*. The New York Times. Retrieved from https://www.nytimes.com/interactive/2020/05/24/us/us-coronavirus-deaths-100000.html

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. http://doi.org/10.18637/jss.v067.i01

Bavel, J. J. V., Baicker, K., Boggio, P. S., Capraro, V., Cichocka, A., Cikara, M., … Willer, R. (2020). Using social and behavioural science to support

COVID-19 pandemic response. *Nature Human Behaviour*, *4*(5), 460–471. http://doi.org/10.1038/s41562-020-0884-z

Best, L. A., Smith, L. D., & Stubbs, D. A. (2007). Perception of Linear and Nonlinear Trends: Using Slope and Curvature Information to Make Trend Discriminations. *Perceptual and Motor Skills*, *104*(3), 707–721. http://doi.org/10.2466/pms.104.3.707-721

Broersma, H., & Molenaar, I. (1985). Graphical perception of distributional aspects of data. *Computational Statistics Quarterly*, *2*(1), 53–72.

Buchanan, L., Park, H., & Pearce, A. (2017, January). You draw it: What got better or worse during obama's presidency. *The New York Times*. The New York Times. Retrieved from `https://www.nytimes.com/interactive/2017/01/15/us/politics/you-draw-obama-legacy.html`

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E.-K., Swayne, D. F., & Wickham, H. (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *367*(1906), 4361–4383. http://doi.org/10.1098/rsta.2009.0120

Burn-Murdoch, J., Nevitt, C., Tilford, C., Rininsland, A., Kao, J. S., Elliott, O., … Stabe, M. (2020). Coronavirus tracked: Has the epidemic peaked near you? *Coronavirus chart: see how your country compares*. Financial Times. Retrieved from `https://ig.ft.com/coronavirus-chart/?areas=eur`

Cleveland, W. S., & McGill, R. (1984). Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. Retrieved from `http://euclid.psych.yorku.ca/www/psy6135/papers/`

ClevelandMcGill1984.pdf

Cleveland, W. S., & McGill, R. (1985). Graphical Perception and Graphical Methods for Analyzing Scientific Data. *Science, New Series*, *229*(4716), 828–833. Retrieved from http://www.jstor.org/stable/1695272

Croxton, F. E., & Stein, H. (1932). Graphic comparisons by bars, squares, circles, and cubes. *Journal of the American Statistical Association*, *27*(177), 54–60.

Croxton, F. E., & Stryker, R. E. (1927). Bar charts versus circle diagrams. *Journal of the American Statistical Association*, *22*(160), 473–482.

Dehaene, S., Izard, V., Spelke, E., & Pica, P. (2008). Log or linear? Distinct intuitions of the number scale in western and amazonian indigene cultures. *Science*, *320*(5880), 1217–1220.

Dunn, R. (1988). Framed rectangle charts or statistical maps with shading: An experiment in graphical perception. *The American Statistician*, *42*(2), 123–129.

Eells, W. C. (1926). The relative merits of circles and bars for representing component parts. *Journal of the American Statistical Association*, *21*(154), 119–132.

Fagen-Ulmschneider, W. (2020). 91-DIVOC. *An interactive visualization of COVID-19*. University of Illinois. Retrieved from https://91-divoc.com/pages/covid-visualization/

Finney, D. J. (1951). Subjective Judgment in Statistical Analysis: An Experimental Study. *Journal of the Royal Statistical Society. Series B (Methodological)*, *13*(2), 284–297. Retrieved from https://www.jstor.org/stable/2984070

GmbH, D. (2020, July). You've informed the public with visualizations about the coronavirus. Thank you. *Chartable*. Retrieved from https://blog.datawrapper.de/datawrapper-effect-corona/index.html

Goldstein, E. B., & Brockmole, J. R. (2017). Sensation and Perception, 480.

Gordon, I., & Finch, S. (2015). Statistician Heal Thyself: Have We Lost the Plot? *Journal of Computational and Graphical Statistics*, *24*(4), 1210–1229. http://doi.org/10.1080/10618600.2014.989324

Green, T. M., & Fisher, B. (2009). The personal equation of complex individual cognition during visual interface interaction. In *Workshop on human-computer interaction and visualization* (pp. 38–57). Springer.

Haemer, K. W., & Kelley, T. L. (1949). Presentation Problems: Suiting the Chart to the Audience: Common Graphic Devices Classified According to Ease of Reading. *The American Statistician*, *3*(5), 11–11. http://doi.org/10.1080/00031305.1949.10501608

Heckler, A. F., Mikula, B., & Rosenblatt, R. (2013). Student accuracy in reading logarithmic plots: The problem and how to fix it. In *2013 IEEE Frontiers in Education Conference (FIE)* (pp. 1066–1071). http://doi.org/10.1109/FIE.2013.6684990

Hofmann, H., Follett, L., Majumder, M., & Cook, D. (2012). Graphical Tests for Power Comparison of Competing Designs. *IEEE Transactions on Visualization and Computer Graphics*, *18*(12), 2441–2448. http://doi.org/10.1109/TVCG.2012.230

Jones, G. V. (1977). Polynomial perception of exponential growth. *Perception & Psychophysics*, *21*(2), 197–198. http://doi.org/10.3758/BF03198726

Katz, J. (2017, April). You draw it: Just how bad is the drug overdose epidemic? *The New York Times*. The New York Times. Retrieved from `https://www.nytimes.com/interactive/2017/04/14/upshot/drug-overdose-epidemic-you-draw-it.html`

Lenth, R. V. (2021). *Emmeans: Estimated marginal means, aka least-squares means*. Retrieved from `https://CRAN.R-project.org/package=emmeans`

Lewandowsky, S., & Spence, I. (1989). The Perception of Statistical Graphs. *Sociological Methods & Research*, *18*(2-3), 200–242. http://doi.org/10.1177/0049124189018002002

Loy, A., Follett, L., & Hofmann, H. (2016). Variations of $q - q$ Plots: The Power of Our Eyes! *The American Statistician*, *70*(2), 202–214. http://doi.org/10.1080/00031305.2015.1077728

Loy, A., Hofmann, H., & Cook, D. (2017). Model Choice and Diagnostics for Linear Mixed-Effects Models Using Statistics on Street Corners. *Journal of Computational and Graphical Statistics*, *26*(3), 478–492. http://doi.org/10.1080/10618600.2017.1330207

Luraschi, J., & Allaire, J. (2018). *r2d3: Interface to 'D3' visualizations*. Retrieved from `https://CRAN.R-project.org/package=r2d3`

Mackinnon, A. J., & Wearing, A. J. (1991). Feedback and the forecasting of exponential change. *Acta Psychologica*, *76*(2), 177–191. http://doi.org/10.1016/0001-6918(91)90045-2

Majumder, M., Hofmann, H., & Cook, D. (2013). Validation of Visual Statistical Inference, Applied to Linear Models. *Journal of the American Statistical Association*, *108*(503), 942–956. http://doi.org/10.1080/01621459.2013.808157

Menge, D. N. L., MacPherson, A. C., Bytnerowicz, T. A., Quebbeman, A. W., Schwartz, N. B., Taylor, B. N., & Wolf, A. A. (2018). Logarithmic scales in ecological data presentation may cause misinterpretation. *Nature Ecology & Evolution*, *2*(9), 1393–1402. http://doi.org/10.1038/s41559-018-0610-7

Mosteller, F., Siegel, A. F., Trapido, E., & Youtz, C. (1981). Eye Fitting Straight Lines. *The American Statistician*, *35*(3), 150–152. http://doi.org/10.1080/00031305.1981.10479335

Munroe, R. (2005). Log scale. *xkcd: A webcomic of romance, sarcasm, math, and language*. Creative Commons. Retrieved from `https://xkcd.com/1162/`

Nieder, A., & Miller, E. K. (2003). Coding of cognitive magnitude: Compressed scaling of numerical information in the primate prefrontal cortex. *Neuron*, *37*(1), 149–157.

Peterson, L. V., & Schramm, W. (1954). How accurately are different kinds of graphs read? *Audio Visual Communication Review*, 178–189.

Romano, A., Sotis, C., Dominioni, G., & Guidi, S. (2020). *The Scale of COVID-19 Graphs Affects Understanding, Attitudes, and Policy Preferences* ({SSRN} {Scholarly} {Paper} No. ID 3588511). Rochester, NY: Social Science Research Network. Retrieved from `https://papers.ssrn.com/abstract=3588511`

Ryanabest. (2021, June). Build an NBA contender with our roster-shuffling machine. *FiveThirtyEight*. Retrieved from https://projects.fivethirtyeight.com/nba-trades-2021/

Shah, P., & Carpenter, P. A. (1995). Conceptual limitations in comprehending line graphs. *Journal of Experimental Psychology: General*, *124*(1), 43.

Shah, P., Mayer, R. E., & Hegarty, M. (1999). Graphs as aids to knowledge construction: Signaling techniques for guiding the process of graph comprehension. *Journal of Educational Psychology*, *91*(4), 690.

Siegler, R. S., & Braithwaite, D. W. (2017). Numerical Development. *Annual Review of Psychology*, *68*(1), 187–213. http://doi.org/10.1146/annurev-psych-010416-044101

Spence, I. (1990). Visual psychophysics of simple graphical elements. *Journal of Experimental Psychology: Human Perception and Performance*, *16*(4), 683–692. http://doi.org/10.1037/0096-1523.16.4.683

Sun, J. Z., Wang, G. I., Goyal, V. K., & Varshney, L. R. (2012). A framework for Bayesian optimality of psychophysical laws. *Journal of Mathematical Psychology*, *56*(6), 495–501. http://doi.org/10.1016/j.jmp.2012.08.002

Tan, J. K. (1994). Human processing of two-dimensional graphics: Information-volume concepts and effects in graph-task fit anchoring frameworks. *International Journal of Human-Computer Interaction*, *6*(4), 414–456.

Teghtsoonian, M. (1965). The judgment of size. *The American Journal of Psychology*, *78*(3), 392–402.

Timmers, H., & Wagenaar, W. A. (1977). Inverse statistics and misperception of exponential growth. *Perception & Psychophysics*, *21*(6), 558–562. http://doi.org/10.3758/BF03198737

Unwin, A. (2020). Why is Data Visualization Important? What is Important in Data Visualization? *Harvard Data Science Review*. http://doi.org/10.1162/99608f92.8ae4d525

Uri, G., & Haemer, K. (1948). Presentation Problems: Graphic Representation for Multiple Linear Correlation. *The American Statistician*, *2*(6), 19–19. http://doi.org/10.1080/00031305.1948.10483416

Vanderplas, S., Cook, D., & Hofmann, H. (2020). Testing Statistical Charts: What Makes a Good Graph? *Annual Review of Statistics and Its Application*, *7*(1), 61–88. http://doi.org/10.1146/annurev-statistics-031219-041252

VanderPlas, S., & Hofmann, H. (2015). Spatial reasoning and data displays. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 459–468.

VanderPlas, S., & Hofmann, H. (2017). Clusters Beat Trend!? Testing Feature Hierarchy in Statistical Graphics. *Journal of Computational and Graphical Statistics*, *26*(2), 231–242. http://doi.org/10.1080/10618600.2016.1209116

Varshney, L. R., & Sun, J. Z. (2013). Why do we perceive logarithmically? *Significance*, *10*(1), 28–31. http://doi.org/10.1111/j.1740-9713.2013.00636.x

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

Wickham, Hadley. (2011). ggplot2. *Wiley Interdisciplinary Reviews: Computational Statistics*, *3*(2), 180–185.

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2010). Graphical inference for infovis. *IEEE Transactions on Visualization and Computer Graphics*, *16*(6), 973–979. http://doi.org/10.1109/TVCG.2010.161

Wickham, Hadley, & Grolemund, G. (2016). *R for data science: Import, tidy, transform, visualize, and model data*. " O'Reilly Media, Inc.".

Wilkinson, L. (2012). The grammar of graphics. In *Handbook of computational statistics* (pp. 375–414). Springer.

Wood, S. N. (2003). Thin-plate regression splines. *Journal of the Royal Statistical Society (B)*, *65*(1), 95–114.

Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, *99*(467), 673–686.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, *73*(1), 3–36.

Wood, S. N. (2017). *Generalized additive models: An introduction with r* (2nd ed.). Chapman; Hall/CRC.

Wood, S. N., N., Pya, & S"afken, B. (2016). Smoothing parameter and model selection for general smooth models (with discussion). *Journal of the American Statistical Association*, *111*, 1548–1575.