




Monte Carlo Analyses for Single-Case Experimental Designs: An Untapped Resource for Applied Behavioral Researchers and Practitioners

Jonathan E. Friedel¹  · Alison Cox² · Ann Galizio³ · Melissa Swisher⁴ · Megan L. Small¹ · Sofia Perez¹

Accepted: 1 October 2021 / Published online: 24 November 2021
© Association for Behavior Analysis International 2021

Abstract

Group-based experimental designs are an outgrowth of the logic of null-hypothesis significance testing and thus, statistical tests are often considered inappropriate for single-case experimental designs. Behavior analysts have recently been more supportive of efforts to include appropriate statistical analysis techniques to evaluate single-case experimental design data. One way that behavior analysts can incorporate statistical analyses into their practices with single-case experimental designs is to use Monte Carlo analyses. These analyses compare experimentally obtained behavioral data to simulated samples of behavioral data to determine the likelihood that the experimentally obtained results occurred due to chance (i.e., a *p* value). Monte Carlo analyses are more in line with behavior analytic principles than traditional null-hypothesis significance testing. We present an open-source Monte Carlo tool, created in *shiny*, for behavior analysts who want to use Monte Carlo analyses in addition as part of their data analysis.

Keywords Monte Carlo · *Shiny* · statistical analysis · Single-case experimental designs · Visual analysis

✉ Jonathan E. Friedel
jfriedel@georgiasouthern.edu

¹ Department of Psychology, Georgia Southern University, 2670 Southern Drive, Statesboro, GA 30460-8041, USA

² Brock University, St. Catharines, ON, Canada

³ Middle Tennessee State University, Murfreesboro, TN, USA

⁴ Purdue University, Lafayette, IN, USA

Behavior analysis has a rich tradition of single-case experimental designs (e.g., Perone, 1991). Both the experimental and the applied analysis of behavior have heavily favored single-case experimental logic (cf. Sidman, 1960) as the best approach to understand functional relations between environment and behavior. In the past, the predominant data analysis method associated with single-case designs has been visual analysis. In other words, evidence to support some effect exists only if an experimental manipulation leads to a large and orderly change in behavior that is clearly visible. However, behavior analysts have recently been encouraged to use statistical analyses to report their results and thus, bolster their claims about treatment effects. Incorporating statistical analyses may also serve to facilitate easier comparisons with group design results (Crosbie, 1999; Killeen, 2019; Kyonka et al., 2019; Young, 2019). This could mean an increased likelihood that results from single-case experimental designs will be included in meta-analysis evaluating specific phenomenon.

Statistical Analysis Limitations and Resistance to Uptake in Behavior Analysis

Across many disciplines, traditional null-hypothesis significance testing (NHST) is the predominant data analytic technique. In general, the inferential statistical tests featured in undergraduate and early graduate coursework are easy to conduct with off-the-shelf software, offer precise estimates about a sample, and can supposedly comment on the likelihood that an outcome could have been due to chance (Field, 2018). However, these analyses are only valuable if the data adhere to several assumptions about its shape and properties (e.g., normality, homoscedasticity, independent data points; Field, 2018). The benefits of single-case experimental designs and repeated measurement of behavior (Sidman, 1960) are also directly at odds with the assumption of most typical statistical tests. For example, the commonly used independent-samples *t*-tests or ANOVAs require every data point to be an observation that is fully independent of all other observations. Data from single-case experimental designs explicitly violate this assumption because repeated measures of behavior are collected from the participants in the study. A NHST for a relatively simple ABAB design or three-condition multiple baseline design requires a relatively complex mixed-effects general linear model that may or may not require additional caveats (cf. DeHart & Kaplan, 2019). Although it is technically feasible to conduct statistical tests of typical behavior analytic data, a researcher must be cautious in conducting such statistical tests properly with single-case data.

Further, many behavior analysts have mounted strong philosophical and conceptual arguments against a reliance on NHST as the gatekeeper for what is considered “good” behavioral science (Branch, 1999, 2014; Perone, 1999). There are two main problems with NHST if our stated goal as behavior analysts is to determine functional relations between environment and behavior. First, traditional group designs that are used with NHST have been described as inappropriate for studying functional relations between the environment and behavior (Ator, 1999; Branch, 1999, 2014; Perone, 1991; Shull, 1999). As mentioned above, statistical techniques, such

as parametric *t*-tests and ANOVAS, require only single measurements from randomly sampled subjects with completely independent data points.

Second, Branch (2019) has made a compelling argument that studies resulting in a binary yes–no determination of whether an effect exists are the slowest way to map the functional relations that are of interest to behavior analysts. The goal of a NHST analysis is to determine whether there was or was not an effect (e.g., a *p*-value) of some independent variable (IV) on the dependent variable (DV). However, it has been well-established that the *p*-value from typical statistical tests is not useful for determining whether the IV has an effect on the DV (e.g., see Branch, 2014; Trafimow, 2019; Wasserstein et al., 2019).¹ Considering the ubiquity of statistical testing and *p* values, some scientists may be surprised to learn some of these facts. *The American Statistician* recently published a special issue relating only to suggestions on moving past using *p* as a research tool (Wasserstein et al., 2019). Many fields, including behavior analysis, are currently grappling with the conceptual problems with NHST.

Randomization Tests as a Tool for Behavior Analysts

It is reasonable for a behavior analyst to ask why some new tool is necessary to determine if there is a functional relation between an independent variable and behavior when experimental design and visual analysis has served the field so well (cf. Kratochwill et al., 2010). We are not arguing that these sorts of tools should replace visual analysis. Rather, these sorts of analytic processes enable both basic and applied researcher an additional method to validate their results. As described by Cooper et al. (2020) variability is defined as “the extent to which observed values, the data produced by measuring an event, match the true state, or true values, of the event as it exists in nature (p. 786). In some respect, all data analysis techniques are an attempt to accurately describe the relation between behavior and environment. If visual analysis is accurate, then a formalized analytic technique (that is designed to answer a similar question) should provide a similar result. At present, behavior analysts have few ways to examine the accuracy of their results, other than indirectly through reliability measures—which is fraught with limitations.

One analytic method with some of the benefits of traditional NHST that also aligns with behavior analytic principles is the randomization test (Craig & Fisher, 2019; Jacobs, 2019). In a randomization test, results are generated by creating every permutation of each data point to each level of the IV. Randomization tests are the formalized process of examining what some obtained experimental data might look like if “things had turned out a little bit differently.” That is, would a researcher come to a different conclusion if, hypothetically, some of the data points associated with variable A were associated with variable B and vice versa?

¹ For simplicity, we are omitting a discussion of nonparametric tests that do not have the same assumptions as parametric tests and regression techniques that purport to describe whether an independent variable affects the trajectory of the dependent variable.

For example, a behavior analyst who conducts a functional analysis might want to know if their interpretation would change if “data point A from condition X was actually in condition Y.” Comparing experimental data to hypothetical or simulated data in which the results were slightly different will allow a behavior analyst to determine the impact of the change in data on their interpretation. In an ideal case, the results of a formalized randomization test might augment conclusions informed by visual inspection. That is, a randomization test may aid in determining whether a pattern of results was obtained by chance or were due to a genuine relation between the independent variables and behavior. A formal randomization test is compared to the informal process of only considering a few cases for convenience, as is commonplace in single-case experimental research ($n = 3$).

Randomization tests have a long history as a data analytic methodology (Fisher, 1935; Onghena, 2018), and have two main benefits in relation to their use as a statistical test. First, they do not require that data be obtained from a random sample of the population of interest because they rely on a principle of random assignment. This means a given subject, object, or client could have been randomly assigned to any IV or level of an IV (Jacobs, 2019; Onghena, 2018). This is especially relevant for applied behavioral research where random sampling is rare; a common study characteristic undermining the credibility of single-subject experimental design results obtained (e.g., see Long & Hollin, 1995). As described above, randomization tests help with data interpretation (and may offer a way to validate accuracy) by comparing obtained data (e.g., real client or participant data) to hypothetical data (e.g., simulated data) in which the hypothetical data points represent situations wherein participants have been randomly assigned to other IV(s) (e.g., baseline vs. treatment conditions) or levels of the IV (e.g., greater or lesser dosages in medication studies, or greater or lesser percentage correct implementation in treatment fidelity studies).

The second benefit of a randomization test is that the interpretations of experimental data are exclusively about that experiment and data. As described by Jacobs (2019):

We ask, “What if the assignment had been different?” and test that counterfactual by constructing a reference set and superimposing our data onto other possible assignments. If we allow the [randomization test] logic and arrive at $p < .05$, then we can make the inference that the difference in [our experimental treatments] caused a difference in one or more of the responses we observed. (p. 338)

The interpretation from a randomization test about the researcher’s specific experiment is the direct opposite of the more common NHST interpretation, which is about the “unmeasured” population from which the data were sampled. In other words, randomization tests provide information about the experimental effects specific to clients in the study, which is a primary concern for behavior analysts. On the other hand, a NHST presumably provides information about all clients with similar characteristics to the sampled clients but little information about the actual clients in the study due to averaging and obscuring variability for individual clients.

A comprehensive discussion about the many benefits of randomization tests goes beyond the scope of the current article. We strongly encourage interested behavior analysts to read both Jacobs (2019) and Craig and Fisher (2019) because the articles provide a detailed account of the randomization test benefits, how they coincide with behavior analytic principles, and how randomization tests relate to traditional NHST.

An important limitation of formal randomization tests is that it can quickly become impossible to calculate all the permutations of assigning the data points to each independent variable. The formula for the number of combinations (where order of data points does not matter) of n data points to groups with g data points per group is: $n! / ([n - g]! * g!)$. For example, if we had a data set from a 20-session alternating treatments design with behavior assessed in two conditions for an equivalent number of sessions, the randomization test will contain 184,756 possible combinations of those 20 data points. The number of possible data arrangements is even greater if we believe the specific order of data points matter (i.e., permutations). It should be noted that the permutations of data matter when we are interested in a decreasing trend in the rate of self-injurious behavior from baseline to intervention and would conversely expect to see an increasing trend in the rate of behavior from intervention back to baseline. That is, the session order is particularly important (i.e., carryover effects). Thus, for even relatively simple behavior analytic designs, randomization tests might be impractical or impossible.

Monte Carlo Methods Explained

Monte Carlo methods are one way of obtaining all the benefits of randomization tests while sidestepping the work required to create every permutation of data by level of IV(s). In general, modern Monte Carlo methods simulate some object or process and provide information about a system that relies on that object or process (Kroese et al., 2014). For example, Monte Carlo methods are used in the telecommunications industry to determine ideal locations to place wireless phone towers (Baggio & Langendoen, 2008). There are many variables that affect the reliability of a phone network, such as the number of phone users, the services (e.g., talking, text messaging, internet connectivity), and the specific locations of the users. Like with permutation tests, considering the large number of variables that need to be accounted for, it can be difficult or impossible to determine the ideal solution by predicting the effects of every combination of the variables. Therefore, Monte Carlo methods are used to simulate a large number of those combinations of variables to determine ideal phone tower placement. As another example, Monte Carlo analyses are used in business risk management to provide confidence estimates about how likely some job will be completed within a specified time frame (see Kwak & Ingall, 2007). It is common that there a vast array of factors that affect the likelihood a job will be completed on time. Thus, by simulating different scenarios it is possible to obtain information about the likelihood of specific scenarios (i.e., ones where the job is completed on time). In behavior analytic research and practice, Monte Carlo methods can be useful as a bridge towards conducting a randomization test without

having to calculate every permutation of simulated data. In other words, what is the likelihood that my independent variable is related to the behavior of interest?

Behavior analysts could use a Monte Carlo method to randomly sample measures of behavior from across an entire experiment to analyze data in the same way they would use a randomization test. For instance, imagine a scenario in which a behavior analyst is trying to assess client requests for attention across baseline and intervention phases. A behavior analyst might conclude that their intervention was effective if the percent correct, independent requests for attention reached 90% for three consecutive sessions. The Monte Carlo analysis would be useful to determine the likelihood that three consecutive sessions of above 90% accuracy could occur by chance. This is done by simulating randomly selected sets of three data points (from all of the data from that client). Thus, the question answered by a randomization test (Jacobs, 2019) is the same as a Monte Carlo analysis: “Would we make the same decision if our data looked different?”

The key difference between a randomization test and a Monte Carlo analysis is that in the former we check every possible arrangement of data. By contrast, a Monte Carlo analysis simulates a large subset of those possible arrangements. In the example from the preceding section (two conditions and an alternating treatment design), instead of creating all 184,756 permutations of behavior by condition we might have the Monte Carlo simulate a random 1,000 of those permutations. So, we can compare our experimentally obtained data to a subset of simulated (hypothetical) data sets that would have been calculated with a randomization test and obtain a *p* value that is robust and valuable. For instance, if the experimental data shows behavior under condition A (e.g., baseline) is higher than behavior under condition B (e.g., treatment) and the Monte Carlo based randomization test could not replicate that result in 1,000 randomly simulated samples, then we have strong evidence to believe that the target behavior is reliably higher during condition A and the results from our convenience sample are genuine.

There are examples of how a Monte Carlo analysis can be applied to study clinically relevant phenomena, such as the relapse of behavior. Relapse, defined broadly, is the recovery of a previously suppressed response and includes phenomena such as reinstatement, resurgence, renewal, and incubation of craving (see Marchant et al., 2013). Friedel, Galizio, et al. (2019b) used Monte Carlo based analyses to reanalyze previously published data from several single-case, nonhuman animal relapse experiments. In that study, 180,000 simulated samples of data were created from the relevant data to answer 35 research questions. Across the experiments, all of the data from each subject were eligible to be included in the simulated samples. Examples of some of the research questions that were reconsidered were whether there was reinstatement across drug administration groups (i.e., Odum & Shahan, 2004), whether there was a relapse of behavioral variability (i.e., Galizio et al., 2018), and if there were differences in relapse based on the arrangement of stimuli across conditions (i.e., Berry et al., 2014). Friedel, Galizio et al. reported on convergent validity. In particular, the Monte Carlo analysis provided a conclusion that was in line with the NHST reported in the original peer-reviewed studies for 33 (94%) of the research questions identified. The authors also advanced the argument that Monte Carlo methods may be more compatible with behavior analytic methodologies for

various reasons stated earlier in the current article (e.g., flexibility, no distribution assumptions, data informing analysis).

Researchers have also leveraged Monte Carlo analyses to generate ways to handle missing data in single-subject experimental design, which is often overlooked but has been described as “prevalent” in the applied behavioral literature (Peng & Chen, 2018; Smith et al., 2012). It is common for missing data to be handled by deleting cases, omitting missing sessions or intervals, or replacing missing values with 0, thus weakening the generalizability of any single-subject experimental design study (Peng & Chen, 2018). Moreover, reducing the sample may not be representative of the population of interest because participants with missing data are not being excluded randomly.

Thus, it appears that behavior analysts can apply Monte Carlo analyses to a wide variety of questions. This method may also mean behavior analysts need not sacrifice the level of specificity about functional relations afforded by single-subject experimental design to conduct statistical tests on their data. That is, standard NHST use central tendencies to evaluate differences across groups, subjects, and/or conditions obscuring important information contained in the session-by-session data.

Considering that Monte Carlo analyses are a type of tool rather than a specific prescription on how to calculate and interpret a statistic, such as *t*-tests, there are innumerable practical applications for Monte Carlo based analyses. Monte Carlo based analyses are fairly common in the development of new behavior analytic tools for data analysis (e.g., see Ferron et al., 2010; Ferron et al., 2017; Friedel, DeHart, et al., 2019a; Giannakakos & Lanovaz, 2019; Gilroy & Hantula, 2018; Moeyaert et al., 2013). However, uptake as a strategy for making decisions about data (e.g., was treatment effective?) within applied research and practice is largely absent.

Some Barriers to Monte Carlo Method in Behavior Analysis

A major barrier to the wide adoption of Monte Carlo analyses in behavior analysis may be the skill set required to create a Monte Carlo analysis. Statistical software packages used to conduct Monte Carlo simulations usually focus on identifying statistical parameters (e.g., bootstrapping, jackknifing). More diverse statistical packages, such as R (R Core Team, 2020) or MATLAB (MathWorks, 2020), can be used to easily conduct a Monte Carlo analysis for behavior analysts by creating random samples of behavioral data. However, conducting an analysis in R or MATLAB requires knowledge of the environment, skills in programming, and the ability to properly organize data in the environment so that the software can generate randomly sampled data points. Although R is free, and there are free online tutorials to learn both R and MATLAB, learning the foundational skills is still a significant investment of time, which practicing behavior analysts likely do not have. It is also technically possible to conduct Monte Carlo based methods in spreadsheet software such as Microsoft Excel, but Excel does not reliably produce sufficiently random numbers when compared to other software (McCullough & Heiser, 2008). Thus, Microsoft Excel is not typically recommended for large scale uses of randomization functions.

Creating a user-friendly tool capable of automating Monte Carlo analysis for reasonably common behavior analytic data may reduce some of the computational barriers, thus addressing uptake issues. Automation may mean researchers can more easily assess difficult to evaluate phenomenon, such as the impact of psychotropic medication adjustments on behavior function where behavior analytic researchers and practitioners do not have control over the primary independent variable (i.e., medication change). Monte Carlo based methods may also be beneficial in analyzing data from within applied settings where more than one return to baseline cannot be easily implemented (e.g., group home settings with adult clients where caregivers may be more reluctant to withdraw an intervention that they subjectively identify as effective).

log Proportion Responding

In addition to Monte Carlo based analyses, we also believe that behavior analysts may find use in log proportion responding as a helpful measure of behavior (Friedel et al., 2017; Friedel, Galizio, et al., 2019b). log Proportion Responding expresses the rate of behavior of one session divided by the rate of response of the immediately preceding session, followed by calculating the logarithm of that ratio. The concept of a logarithm is not foreign to most behavior analysts, given we have seen semi-logarithmic scales used in specific charting techniques (i.e., standard celeration charts; Lindsley, 1992). The benefits of using a logarithmic (nonlinear) scale on the y-axis is that increments occupy the same space on the graph (e.g., an increase of correct responses from 1 to 2 is the same as an increase of correct responses from 100 to 200). Graphing this way may be less likely to over- or underemphasize DV changes observed in response to shifts in the IV (i.e., suggesting that a change of 1 to 2 is more or less “impressive” than a change of 100 to 200). Logarithmic charting enables practitioners to depict numerical data over a wide range of values in a compact way.

As with semi-logarithmic charting, logarithms are helpful in “leveling the playing field” when calculating summary measures of behavior. In particular, there are benefits of expressing behavior as a proportion of behavior during the preceding session (Friedel et al., 2017; Friedel, Galizio, et al., 2019b) when measuring trends in behavior or potentially transient effects of functional variables on behavior, such as in a multiple-probe design (e.g., Horner & Baer, 1978). For example, in addition to commenting on variability, level or trends using visual analysis (Cooper et al., 2020, pp. 146–150) one may be interested in *quantifying* the difference in variability between two conditions by objectively comparing target responding at baseline versus responding during the intervention condition. In a technical sense, the measure is not a proportion but a ratio (cf. Pustejovsky, 2018), but the terminology has been “log proportion responding” for several years (Friedel et al., 2017).

One inherent issue with simple proportions that log proportion responding transformation may address is that increases in behavior have a greater impact on results than decreases in behavior. For example, if responding during Session 2 is 10 and responding during the first session is 1, this represents a proportion of

0.1. On the other hand, if responding during Session 1 is 1 and responding during Session 2 is 10, that would be a proportion of 10. In comparing interventions that decrease responding to interventions that increase responding, it may appear as though interventions targeting behavior increases have a more profound impact. But this reflects the difference in scale, rather than a *real* difference in behavior. Further, many statically based analyses will weigh proportional increases more heavily. Simply put, an identical change in behavior, but in opposite directions (increase vs. decrease), can have a “bigger impact” on results. Conducting a log transformation of simple proportion would conceptually accomplish the same outcome as generating a semi-logarithmic graph from a linear one. Namely, the log transformation moves proportional changes into the same space, just as the semi-logarithmic graph forces changes to occupy the same space in the graph. Proportional changes are “equalized” by the log transformation process wherein a change of 10 to 1 now equals -1, whereas 1 to 10 now equals +1, and 1 to 1 equals 0, just as 10 to 10 now equals 0 (all of these specific transformations assume a base of the logarithm of 10).

Expressed mathematically, the log proportion rate of response is

$$Y = \log_2 \left(\frac{B_n + c}{B_{n-1} + c} \right) \quad (1)$$

where Y is the log proportion response rate, B is the rate of response during session n , and c is an experimenter determined correction factor (Friedel, Galizio, et al., 2019b). The precise value of the correction factor (c) is determined by the researcher or practitioner. It should be selected to minimize the overall impact of the correction on the measure of behavior (i.e., a c of 1,000 could mask the behavioral effects). Choosing the right correction factor is especially important when there are very low behavior rates because a small c can affect log proportion response rates. Therefore, we recommend a correction value of one because it is the least disruptive to a proportional measure, especially when responding during a given session is zero. It is possible to use any log base when calculating the log proportion, but we recommend a log base of 2 because it is more intuitive in the common ranges of behavior at steady state. That is, a log proportion response rate of 1 indicates a doubling of response rate. Simply put, calculating log proportion of responding represents a standardized way to distinguish between behavior changes achieved through shifts in the independent variable versus natural session-to-session variability without having to obtain larger sample sizes or conduct extensive replications.

Combining log proportion responding and Monte Carlo strategy may enhance the analysis of data from single-case experimental designs, in particular when the question of interest relates to variability in responding or transient changes in behavior. First, by calculating log proportion rate of responding differences across subjects as well as sessions can be expressed on a standardized scale. Then, Monte Carlo analyses can be used to simulate different arrangements of the log proportion rate of responding data to determine whether there are systematic differences in variability or some transient effect on behavior. Thus, log proportion rates of response can extend Monte Carlo analyses utility for some research questions.

This article describes a free, open-access, and online app that we developed specifically to conduct Monte Carlo analyses for single-case experimental design data. In addition, the app has the ability to calculate log proportion responding based on data supplied to the app. The app requires little prior knowledge on how Monte Carlo simulations work, requires no programming on the part of a behavior analyst, and provides an output of the simulated samples based on the behavior analyst’s data. As a basic description of the app: the behavior analyst supplies data, indicates column identifiers (e.g., what column is for sessions, behavior, subjects), and then uses selection boxes to indicate the experimental data they want to test. The most burdensome requirement related to applying this app is that data must be in a long-format (e.g., tidy data; Wickham, 2014) and in a comma-separated file (csv); see Fig. 1 for an example. The Monte Carlo analysis compares the selected experimental data against 1,000 random samples of other data from within the data file (as described above). As an output, the app provides the information necessary for the user to calculate a randomization test p -value and provides a graphical display of how the obtained data compares to the simulated data. We hope the app featured in this article will assist behavior analysts interested in applying these methods and, thus, reaping the benefits of doing so, including: (1) exploring and demonstrating accuracy of results; (2) addressing inherent issues with single-subject experimental design (e.g., convenience sampling); and (3) increasing the likelihood that results obtained via single-subject experimental design may be incorporated into meta-analysis.

Finally, incorporating these tools to augment outcomes supported by visual inspection may have the added benefit of improving behavior analysis visibility across other behavioral health-oriented disciplines, given many of these areas primarily ascribe to group design experimental rationale.

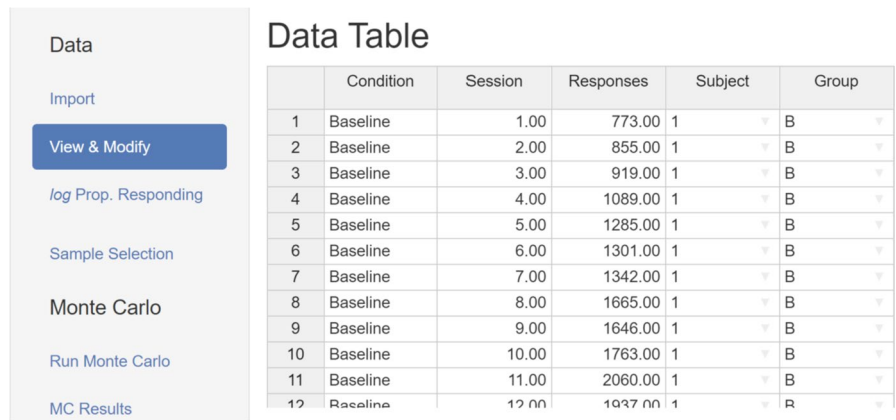


Fig. 1 Long-Format Data in the App. *Note.* Sample data included in the app to demonstrate how users should organize their data in Microsoft Excel before converting to a CSV file. For long-format, data are organized down the columns for each variable rather than across in the rows.

A Monte Carlo App for Behavior Analysts

App Environment

A link to the app can be found in the Availability of Data and Programming note at the end of the paper. The app was developed using *shiny* (Chang et al., 2020). *Shiny* is an open-source package for the statistical software R (R Core Team, 2020). *Shiny* includes methods for building web-based applications that can conduct analyses in R. The *shiny* package has built-in methods and functions to deploy web-based apps for end-user's ease. Some of these methods and functions include preset methods to display html text, having a navigation bar-based menu, having buttons to collect user inputs, and displaying figures calculated from data. *Shiny* apps that are deployed to a server conduct all the analyses on the server and display the results of analyses to any users of the app. A behavior analyst using the app is only required to have access to a modern web browser and not required to have R or *shiny* installed on their local computer (the user does not even need to know what R or *shiny* are). Given the only requirement for using the app is a modern web browser and all of the operations are run on the server, it is possible to use the app via a smart phone. Thus, a behavior analyst who connects to the server (and the app) can conduct any analyses that are built into the *shiny* app.

The core component of the app that was designed to conduct the Monte Carlo analysis via random sampling of data was the *sample_n* function from the *dplyr* package (Wickham et al., 2020). The *sample_n* function is designed to take an *n*-sized random sample from a data set. The app also supports limited graphical plotting of both the data being input and the results of the Monte Carlo analysis. This plotting was made possible with the *ggplot2* R package (Wickham, 2016). An archive of the app and associated files at the time of publication can be found on the Open Science Framework (<https://osf.io/gqtxz/>). A version of the files that will continue to be updated and expanded, based on community input and feedback, can be found at the first author's GitHub repository.

The app produces two main outputs. The first output is a data table of the mean, standard deviation, and sample size of the behavioral measures for each of the simulated samples. As described below, the app has the ability to replicate an analysis across a variable (such as a different Monte Carlo analysis for each subject), in such a case the data table will include the results for all of the replications of the Monte Carlo analysis. The output does not include the specific data records or identifiers that were randomly selected for each simulated sample because the file size is prohibitively large based on our server specifications. The app will also automatically plot a histogram of the means from all 1,000 simulated samples with an indicator for the experimentally obtained mean. The researcher can download the histogram plot and visually compare the experimental data to the simulated samples. If the Monte Carlo analysis was repeated based on a variable within the data (e.g., participants), then a single figure with panels/facets for the specific histogram for each grouping variable will be displayed.

App Specifications

While developing the app, we balanced competing concerns between making the app as widely applicable as possible while also making it easy to use and being compact enough to run on our server. First, the app is limited in that it can only create 1,000 simulated samples per replication (i.e., subject) in a single operation. The simulation procedure and summarizing the resulting simulations can be computationally intensive in terms of how many operations need to be conducted. Further, creating large simulations (e.g., 10,000 simulated samples for six different subjects) can require an extremely large data file very quickly. For that reason, the app needed to be limited to 1,000 samples so that our server is not overtaxed. We also built in a feature that allows a behavior analyst to replicate their analysis if they return to the app at a later date. Computer-based random processes necessarily use deterministic equations to simulate a random process. A random seed value can be used to set the starting point of the random process such that the process will always follow the same path. In that respect, our app allows the user to replicate their results if they use the same data and seed value across analyses, which is especially important when adhering to a commitment to open science and allows other researchers to replicate the analysis (cf. Hales et al., 2019).

As previously mentioned, the app has the ability to repeat a Monte Carlo analysis across participants, clusters, or groups. This functionality was designed to aid in the process of a behavior analyst running an identical Monte Carlo analysis across successive subsets of the data. For example, a behavior analyst might be interested in determining if the rate of behavior is decreasing during an extinction condition for each of five participants. The behavior analyst can indicate that there are subsets/groups of data (in this case participants) and the app will conduct a Monte Carlo analysis for each subset/grouping of data. This grouping approach is preferable to an alternative approach of requiring the behavior analyst to separate files for each subset of data and recreate the analysis five separate times for each subject. We consider this grouping function to be an added convenience for the behavior analyst.

How Samples Are Simulated

There are many different ways to conduct a Monte Carlo analysis. We opted for a process that we believe would reduce potential calculation errors with unknown data sets. The size of each simulated sample is determined by the number of data points in the experimental sample. If the behavior analyst is interested in conducting a Monte Carlo analysis on the first five sessions after a phase change for three subjects, then the experimental sample is 15 data points in total and each Monte Carlo sample will be 15 data points. Data points are randomly sampled only from participants whose data are included in the experimental sample. If the behavior analyst decides to only look at the first five sessions after a phase change for two participants and exclude the data from those same sessions from a third participant's data, then the simulated samples will only be based on the data from the first two participants, excluding data from the third. Finally, each sample is created by random

selection with replacement. In other words, it is possible that data points might be repeatedly sampled for each simulated sample.

Using the App

Orienting Users to the App Interface

Prior to providing a step-by-step task list for readers, there are some important items to note in order to be properly oriented to the app. First, the app has a sidebar navigation panel. This panel has different tabs (or pages) for the different steps in setting up the Monte Carlo analysis. Each tab retains the information and settings entered by the behavior analyst even as they switch between pages. For additional user-ease, the app includes an example data set (found on the “Import Data” tab) depicting how input data should look like for the app to function properly. Clicking on the “View & Modify” tab will permit behavior analysts to use the app with the example data set so they can see how the app is designed to process data. These example data also demonstrate what the output (i.e., end-product) will look like. To gain familiarity with the app, behavior analysts are encouraged to replicate the analysis we have described in this article using the example data set prior to attempting their own analysis. Finally, to reduce constraints on the server, the app will disconnect if the user is idle for 5 min. If the app disconnects the user will simply need to reload and replicate their work. This is a default process in place with *Shiny*.

The following sections provide instructions for using the app to conduct a Monte Carlo analysis. As part of the instructions, we will conduct a Monte Carlo analysis using example data included in the app. The example data are a (simulated) relapse experiment. For each subject there are data for three conditions: baseline, extinction, and relapse. The instructions will walk through how to use the app to conduct a Monte Carlo analysis to determine if behavior during the final “relapse” condition was higher than during the final session of extinction (i.e., Did behavior relapse?). This analysis will replicate some of the research questions and analyses reported in (Friedel, Galizio, et al., 2019b).

Step 1: Importing the Data

Behavior analysts will need to import their data. As mentioned above, the data input needs to be in long-format and “tidy” (Wickham, 2014; see Fig. 1 for an example of tidy data). The data must be uploaded/imported as a long-formatted, tidy CSV file. CSV files are a free, nonproprietary file format in which data are organized as rows and columns are separated by commas. With long-formatted, tidy data, there should only be one measure of the DV per row/record (e.g., problem behavior frequency). Further, any other data columns must be other variables associated with each DV (e.g., session number, participant identified, level of IV). It is likely that behavior analysts will have to do some data preprocessing to accommodate this requirement of importing/uploading a tidy CSV file. Data formatting and processing can be a

complex process, depending on how a user formats their data for clinical or research purposes. However, most behavior analysts use Microsoft Excel (Haddock & Iwata, [in press](#), as cited in Mitteer et al., 2020) and it is relatively easy to save/export an Excel worksheet as a CSV file. There are ample free online resources demonstrating how to save an Excel worksheet as a CSV file, including Microsoft support page for Excel.² The data formatting specifications are in place because it reduces the data processing load on the app. Users can import properly formatted data by clicking on the first tab displayed in the app (“Import” tab; see Fig. 2). Clicking on the “Browse . . .” button will open the user’s local file selection window, enabling them to navigate to the relevant file.

For clarity, we restrict using the term “import” to refer to moving data from the behavior analyst’s computer to the app. When referring to data that is within the app and able to be used by the app, we will refer to that as data that are “loaded” into the app. For demonstration purposes we will use the example data, which can be loaded into the app by clicking on the “Example Data” button.

Step 2: View & Modify

After importing the data, behavior analysts can check if their data is properly formatted (i.e., “tidy”) by clicking on the “View & Modify” tab, which will reveal an interactive data table (see Fig. 1). If the first row (the header) in the data table contains column names and there is only one dependent variable datum per row then the imported data was properly formatted, and users can proceed to the next step. The interactive nature of the data table means behavior analysts can rearrange their data or can modify individual values on the app before running an analysis. The data can be updated by clicking on the cell with the value that is to be modified, typing in the corrected value, and then clicking the “Update” button. Although the data table is interactive, we strongly recommend behavior analysts import/upload a data file that is ready to analyze instead of using the app to modify the data. This recommendation is due to the app not having the depth of functionality of software that can easily manipulate CSV files (like Microsoft Excel or Google Sheets).

The “View & Modify” tab also includes three drop-down boxes for behavior analysts to identify the columns that include the measures of behavior, the sessions, and the participant/subject identifiers (see Fig. 2). The drop-down boxes are populated by the column headers from the loaded data file. For example, the example data set included with the app has five columns: Condition, Session, Responses, Subject, and Group. The three drop-down boxes will only include those five columns (from the sample data) as choices for the behavior analyst to indicate which data correspond to the response measure, time or sessions, and individual participant/subject identifiers (see Fig. 3). The behavior analyst must designate these column identifiers for the next steps. The column identifier for measures of behavior is selected with

² The Microsoft support page for saving an Excel workbook as a CSV file can be found here: <https://support.microsoft.com/en-us/office/save-a-workbook-to-text-format-txt-or-csv-3e9a9d6c-70da-4255-aa28-fcac1f081e6>.

Monte Carlo Analysis for Single-Subject Experimental Designs

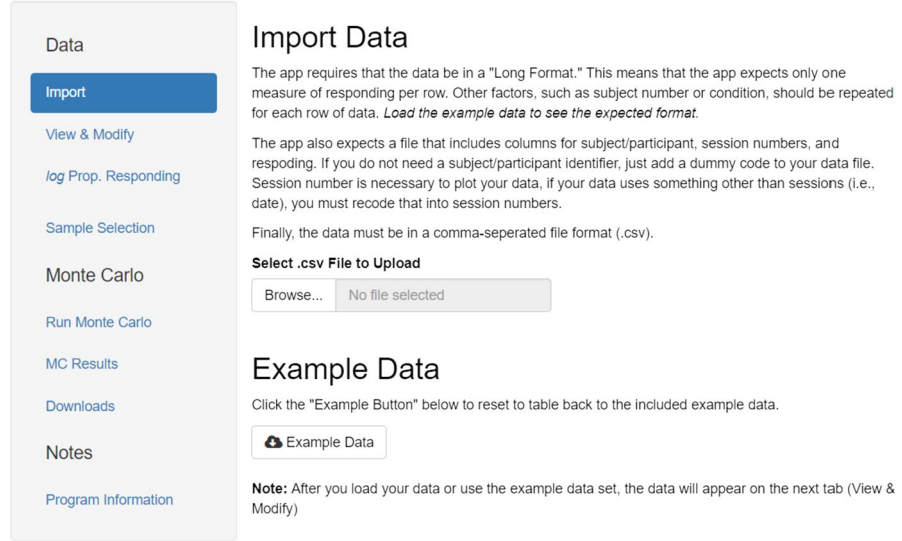


Fig. 2 Import Data Tab in the App. *Note:* The user can either import their own data or use example data embedded in the app

the “column for responding” select box. The column identifiers for session identifiers and subject/participant numbers are the second and third drop down boxes, respectively. To indicate which column header is associated with the relevant data, the user only needs to click on the drop-down box and then click on the relevant labels. In the case of the example data set these columns would be “Responses” for the responding identifier, “Session” for the session identifier, and “Subject” for the subject/participant identifier.

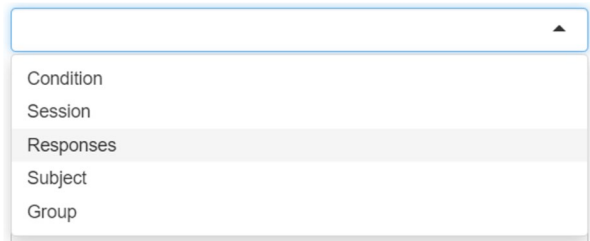


Fig. 3 Column Identification Options from the Example Data. *Note:* The options displayed in the select boxes on the “View & Modify” tab are based on the column headers of the data imported into the app. For this figure, the options were based on the column headers for the example data included with the app

Step 3: log Prop. Responding

The log Prop. Responding tab is where behavior analysts can have the app automatically calculate log proportion responding for the data they have loaded into the app. Calculating log proportion responding is an optional step for the app and Monte Carlo analysis in general. That is, a behavior analyst can conduct a Monte Carlo analysis with or without calculating the log proportion responding. As described in the introduction, there are some potential benefits of calculating log proportion responding based on the type of research question. log Proportion responding can be calculated after the column indicators have been selected, as described in Step 2. The app will not allow the calculation until the column indicators have been selected.

Figure 4 displays the “log Prop. Responding” tab. There are several paragraphs of text included in the app (although, excluded from the figure) that provide brief instructions and reasons why a behavior analyst may wish to calculate and analyze log proportion responding. This text is included as an aid for users. The first step to calculate the log proportion responding is that the behavior analyst must enter their desired base of the logarithm. The default value in the app is for a log base 2, as is specified in Formula 1. Based on rules of logarithms, the main effect of different bases of the logarithm on the final measure of log proportion responding is only in terms of the magnitude of the obtained values. That is, a log base 2 will express changes in behavior based on 2-fold increase (or decrease) in the rate of behavior and a log base 10 will express changes in behavior based on a 10-fold increase (or decrease) in the rate of behavior. To select the desired base

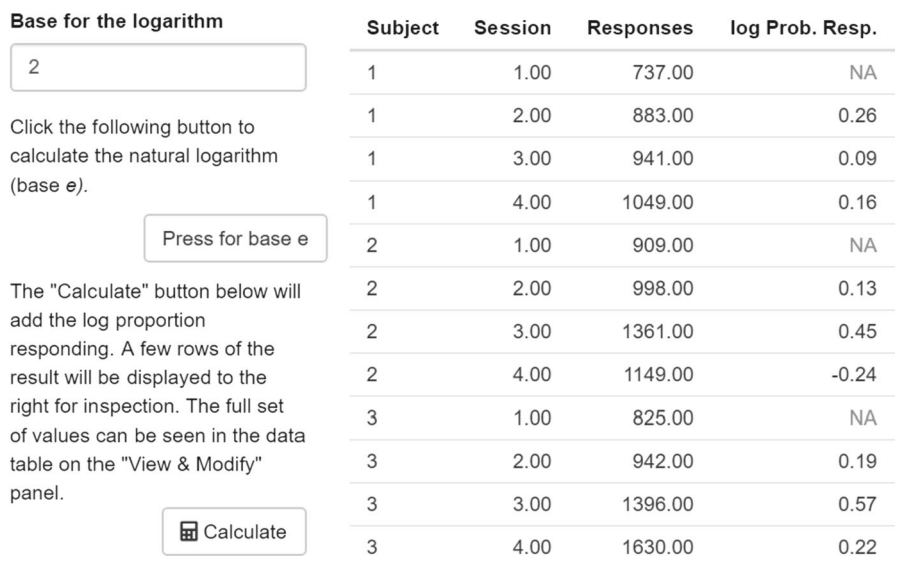


Fig. 4 log Prop. Responding Tab. *Note:* Only the interactive portion of the “log. Prop. Responding” tab is displayed in the figure. The data table will only display after the user clicks the “Calculate” button. Data displayed are from the sample data set included with the app

of the logarithm, the user can either type in the value or click on the up and down arrows on the far right of the entry box labeled “Base for the logarithm.” We also included a button if the user would like to use a base of e for the logarithm (i.e., a natural logarithm). Calculating a natural logarithm is common in many fields, including psychology, so we included this feature as a convenience in case a behavior analyst wishes to calculate the natural logarithm. For the example data, we will use the default base 2 logarithm.

After the base of the logarithm is selected, the behavior analyst only needs to click the “Calculate button” to have the app will calculate the log proportion responding for the currently loaded data. After clicking the button, the app will display a limited data table so that the user can evaluate the log proportion responding values that were calculated. The data table includes the original responding data as well as the newly calculated log proportion responding for only the first four rows of data for the first three subject/participants. After having the app calculate log proportion responding, the user can go back to the “View & Modify” tab to examine the interactive data table that displays the data set. After the app calculates calculating log proportion responding, the measure will be included in a new column on the left of the data that is labeled “log Prop. Resp.”

It is important to note that the app will not use the log proportion responding by default. That is, a behavior analyst may choose to calculate the measure and then decide against using it in the Monte Carlo analysis. If the behavior analyst would like to use the measure in the Monte Carlo analysis, they must change the column that identifies responding to the newly created “log Prop. Resp.” column. If the user wishes to ignore the log proportion responding and conduct the Monte Carlo analysis with the original responding data, then they do not need to take any action.

In the case of the example data, we will change the column that identifies responding to “log Prop. Resp.” for the sake of providing a full demonstration of the app. Demonstrating this optional step will also allow behavior analysts to see how their data might look when using their log proportion data for the Monte Carlo analysis.

Step 4: Sample Selection

Clicking on the “Sample Selection” tab brings the behavior analysts to the next step in using the app to conduct a Monte Carlo analysis. The goal of “Sample Selection” page is to enable the user to indicate which data points are the experimental data they would like to compare against the simulated samples of data that will be created by the Monte Carlo process. For example, if we were comparing behavior during extinction to other conditions then our “experimental sample” would be all the behavior during the extinction condition. It is important to note that this function will not be available to users until they have successfully loaded their data and specified the column labels for the responses, sessions, and participants. The sample selection page is comprised of three sections: (1) a set of responsive selectize boxes to identify the sample data, (2) a figure that serves as a

visual aid to help the behavior analyst determine if the correct data was selected with the filter, and (3) a drop down menu to identify a column for which the data will be treated as “separate groups” and the Monte Carlo analysis will be repeated for each group. The latter section is optional.

Identifying the Sample Data

To begin, behavior analysts should select the relevant data to be included in the sample using the selectize boxes on the lower left-hand side of the page (see Fig. 5). The displayed selectize boxes are based on the columns in the loaded data. There is one selectize box for each column label except the active responding column. The values within each selectize box are all of the unique values within that column. For example, if a condition column has labels for “baseline” and “treatment,” then the associated selectize box for that column label will only have “baseline” and “treatment” as options. After the filter is selected using the selectize boxes, the “Update Filter” button must be clicked so that the app can process which data are included in the sample.

The behavior analyst should treat the selectize boxes as filters they were using to “find” their sample data in the full data set. As an analogy, it is common to have filters on ecommerce sites so that consumers can more easily find products that they are interest in purchasing because those products meet certain criteria. For example, a shopper might only be interested in buying an inexpensive umbrella that is yellow.

Select a grouping factor if you want different Monte Carlo analyses based on that grouping. If no grouping factor is selected, then there will be only one sample. For example, with the Example Data you may wish to have a different Monte Carlo Analysis for each “Group”. Thus, by selecting “Group” as a grouping factor you will get a separate Monte Carlo analysis for each unique group in the data set and the results of each analysis will be independent of one another.

Select grouping factor

Note: The sample selection (to the right) can be difficult to use. If you are struggling to highlight the proper sample using this tool, then consider including an extra column in your data that indicates what data should be in the sample of interest. If you upload the updated data file then you can just use the new column to select/filter your data.

Use the boxes below to select filters for the data you want included in your “real” sample. When completed, click the “Update Filter” button below, which will highlight the selected data in the plot.

Condition

Session

Responses

Subject

Group

 Update Filter

Fig. 5 Sample Selection and Grouping Variable Selection on the “Sample Selection Tab”. *Note:* The values in the grouping factor and the selectize boxes are based on example described using the sample data

Many sites will include the option to filter all the products on the site so that only yellow umbrellas below a specific price point are displayed. In the Monte Carlo app, the selectize boxes on the “Sample selection” tab are designed to be used so that only the experimental sample data are included based on filters selected (like a yellow umbrella). As one example, a researcher would use the selectize box associated with the baseline condition and click or type in “baseline” if they were only interested in baseline sample data. As another example, a behavior analyst would use the selectize box associated with the session number and click or type in each of the values for 16–20 if they were only interested in sample data from sessions 16–20. It is important to note that it is possible to create filters for data that do not exist just like searching for a product that does not exist (e.g., a golf umbrella that costs less than \$1). For example, if a researcher tries to select baseline data for sessions 50–55 but the baseline condition stopped at Session 30, then the behavior analyst is essentially selecting no data.

For the example data set, our proposed research question was about whether behavior was elevated in the relapse condition relative to extinction (i.e., Did relapse occur?). Thus, for our example data the relevant sample for the Monte Carlo analysis data are all in the “Relapse” condition. To include the data from the relapse condition in the sample, the user only needs to click on the selectize box for “Condition” and either click or type “Relapse.” After “Relapse” is in the relevant selectize box, the user must click the “Update Filter” button.

As a note of caution, the interface for selecting the sample data only allows for simple identification of the sample and does not allow for conditionals (these sessions for participant X and these other sessions for participant Y). Behavior analysts often have conditional methods for identifying their sample data of interest. An example of a conditional method for identifying sample data may include examining a sample of behavior from Subject A in which tangible reinforcers were delivered and from Subject B in which edible reinforcers were delivered. Thus, finding the sample data of interest relies on a conditional relationship across the variables (e.g., IF “subject a” THEN “tangible reinforcer,” ELSE IF “subject b” THEN “edible reinforcer”), which the App cannot do. To arrange and complete these types of conditional sample data analyses, we recommend that behavior analysts create the filter in a unique column in the data file prior to uploading the data to the app. A researcher can include an additional column labeled “filter” in the data file with a unique label for the subset of data that should be included in the sample. Then the experimenter can just use the “filter” selectize box and click to include data that was prefiltered. Although we propose this as a solution for complex filters, researchers could also use this technique of creating a column to indicate which are the relevant sample data in general instead of relying on a combination of selectize boxes.

Sample Selection Visual Aid

The next key feature of the “Sample Selection” tab is a simple single-subject style figure displayed at the top of the page in the app (see Fig. 6). The figure has a panel for each subject, with responding plotted against session number. The figure does not include any phase change lines and assumes that there is only one measure of

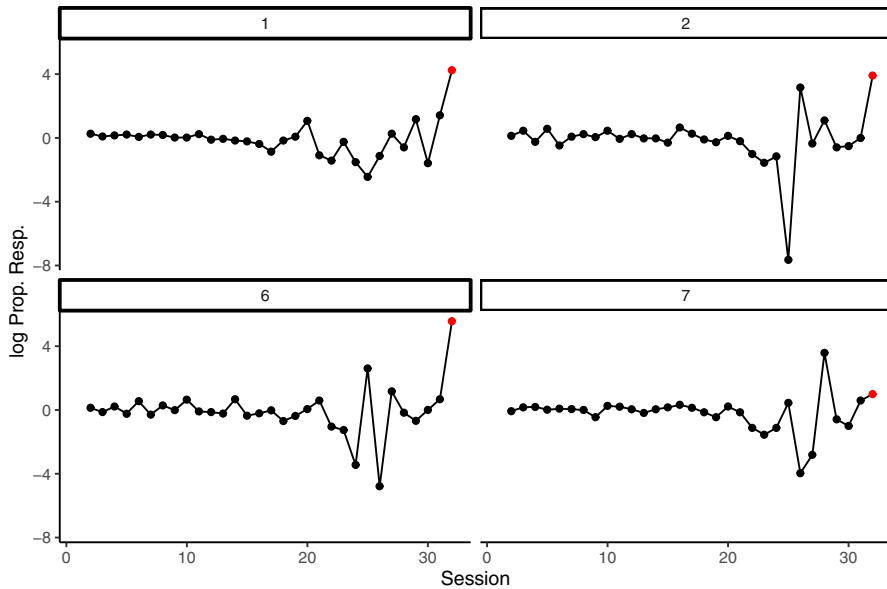


Fig. 6 Sample Selection Visual Aid Figure. *Note:* Example of the sample selection visual aid, for clarity in this figure only the upper left corner of what would be displayed in the app is included (i.e., example data for subjects 1, 2, 6, and 7). The figure was saved after the sample selection was made and the filter was updated, so the selected sample data are highlighted in red. Based on the specification in the example for the sample data (comparing the relapse session to the preceding extinction session), the final data points for each subject are highlighted red

behavior per session. When the responding, session, and subject identifiers have been specified, the figure will be plotted automatically. After the behavior analyst selects the filter for identifying the sample data of interest and clicks on the “Update Filter” button, the identified sample data points will be highlighted in red on the figure. The user should check that the highlighted data points roughly align with their expected sample of interest. In the sample data set, the relapse condition was the last session for each subject. Thus, after the “Relapse” condition under the “Condition” selectize box was selected and the “Update Filter” button was clicked, the last data point for each subject will be highlighted in red on the figure. If, as described above, the filter is created in such a way that none of a subject’s data is included in the sample, then there will be no red data points for that subject.

Iterations of a Monte Carlo Analysis

There are scenarios in which a behavior analyst may wish to conduct an identical Monte Carlo analysis across subsets of data within a large data set. For example, a user might want a Monte Carlo analysis to provide additional evidence of the identified function of behavior from a functional analysis for three different clients. The function of behavior for client A is most likely not relevant to the function of behavior for client B. In other words, the user really has need for three separate Monte Carlo analyses with similar features. The app has the ability to repeat a Monte Carlo

analysis for different subsets/groupings of data as identified by the user. To conduct separate Monte Carlo analyses for each subset/group the user only needs to use the drop-down box to select the column identifier for the grouping variable (see Fig. 5). Following the above example, the behavior analyst would only need to select the column that indicates the client ID to conduct separate Monte Carlo analyses for each client. It should be noted that when grouped simulations are conducted, the Monte Carlo simulation will only randomly sample data included within that group. For example, when conducting the Monte Carlo simulation for Group A, no data from Groups B or C will be included. With the included example data, we can conduct a separate Monte Carlo analysis for each group in the data (arbitrary labels of A, B, and C).

Step 5: Running the Monte Carlo Process

Having the app initiate the Monte Carlo analysis is relatively easy once the data have been loaded, the filter for the sample data has been specified, and—optionally—a grouping variable over which to replicate the analysis has been specified. To have the app conduct the Monte Carlo analysis, the behavior analyst need only click the “Run Monte Carlo Simulation” on the “Run Monte Carlo” tab (see Fig. 7). The app

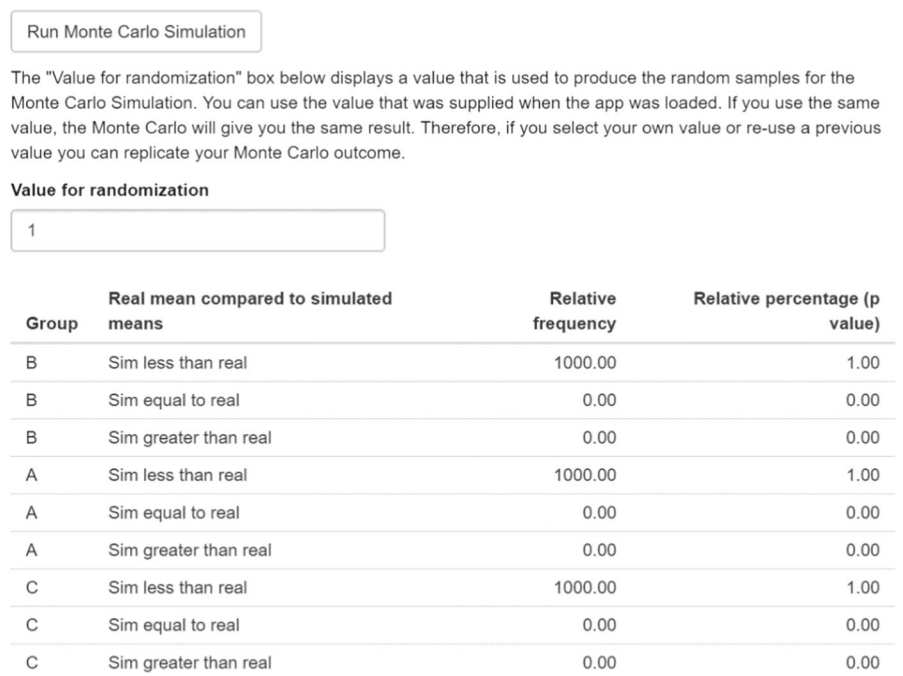


Fig. 7 “Run Monte Carlo” Tab. *Note:* The “Run Monte Carlo Simulation” button, value for randomization, and summary results of the example Monte Carlo analysis for the sample data. The data table is only obtained after clicking the “Run Monte Carlo Simulation” button. The user can create identical results to those reported in the manuscript if the “Value for randomization” is set to 1

may take upwards of a minute or two to calculate all simulated samples depending on the size of the sample and the number of groups/subsets of data. A spinning dial will be displayed to indicate that the app is conducting the Monte Carlo analysis.

The tab also includes a text box for the user to enter a random seed value for the Monte Carlo simulation. The random seed value is an integer that is used to initiate a computerized pseudorandom process. As described above, if the same seed value and the same data are supplied to the app, the Monte Carlo analysis will produce the same results because the pseudorandom selection of data to be included in the various samples will start in the same place and follow the same deterministic path. In this manner, the app can provide a replicable analysis. The initial seed value that is displayed in the box is determined randomly when the app is loaded. If the behavior analyst wishes to use their own seed value or one to replicate a prior analysis, they simply type that seed value into the box. For the example data, we will set the seed value to 1 by typing that value into the textbox so that the results we describe can be replicated.

Step 6: Interpreting the Monte Carlo Results

The first portion of interpreting the Monte Carlo analysis is reported on the “Run Monte Carlo” tab once the analysis is complete. A summary table of the results will be displayed on the “Run Monte Carlo” tab. If the behavior analyst specified that the Monte Carlo analysis should be replicated for different subsets/subjects/groups, then the counts will be repeated per group variable. This summary table is where the relevant p -value for the analysis can be found. The table displays the count of simulated means that are less than, equal to, or greater than the mean of the original sample of data. If the simulated means are reliably lower than the experimentally obtained sample means (and the experimental logic is sound), then Monte Carlo analysis indicates that the sample mean is statistically significantly higher than what would be expected by chance. The reverse is also true if the simulated means are reliably higher than the experimentally obtained sample. The relative percentage column indicates what percentage of the sample is above, equal to, and below the experimentally obtained sample. This value can be taken as the p -value for the comparison of interest. If the Monte Carlo simulation could not simulate a sample more extreme than the experimentally obtained sample (i.e., relative frequency and percentage equal zero), the most conservative interpretation is $p < .001$.

For the sample data, our research question was whether behavior during the one session of the relapse condition was higher than during the last session of extinction. We can potentially answer this question for each group within our data set because we requested that the app replicate the Monte Carlo analysis for each of the groups. The summary table displays the results of the Monte Carlo simulations for each of the three groups. Across each group, the Monte Carlo simulation could only simulate samples that had lower mean log proportion responding than the experimental sample. In other words, if we look at 1,000 random arrangements of data (for each respective group) we cannot create a simulated sample that has a log proportion rate of response as extreme as the rate of responding we obtained through our

experiment. Thus, log proportion responding during the relapse condition was significantly higher than expected by random chance. In particular, a formal written expression of the Monte Carlo analysis for Group A is: “For Group A, log proportion responding was significantly higher ($p < .001$) than expected based on random selections of sessions.” The other groups could be reported in the same manner. Our data are potentially extreme, for the purposes of clarity. If the results were not so clear and the relative percentage reported by the Monte Carlo analysis was 0.042 (42 samples were less than the experimentally obtained sample of log proportion rate of response), for example, then the p -value would be reported as $p = .042$. In addition to reporting the p , users will likely wish to report some relevant summary statistic such as the mean of the experimental sample.

The “MC Results” tab displays a figure of the results from the Monte Carlo analysis (Fig. 8). The figure is supplemental to the table described above. The figure displays a histogram of the means of all 1,000 simulated samples with a different panel for each Monte Carlo analysis associated with a subset/grouping (if specified by the behavior analyst). The different panels might be very small if the user specified a high number of subsets/groupings. For that reason, the bars on the histogram are colored with alternating colors to help easily identify the bins. Presented in addition to the histogram is a vertical dashed line that extends from the x-axis to the top of the panel. The vertical line represents where the sample data (for that subset/group, if specified) occurred in relation to the simulated samples.

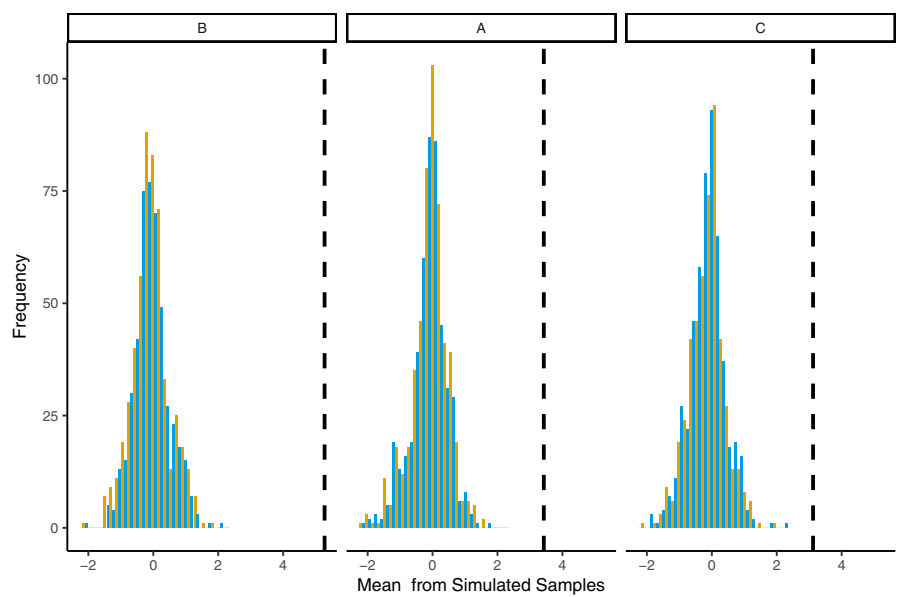


Fig. 8 Monte Carlo Analysis Histogram. *Note:* Histogram from the example Monte Carlo analysis on the sample data. Figure will only display after the Monte Carlo analysis has completed

Step 7: Downloading the Results

The final tab is a “Downloads” tab where the behavior analyst is able to download two files. The first file is a higher resolution copy of the figure displayed on the “MC Results” tab. The second file is the summary data for each simulated sample across the groups/subsets of data. The Monte Carlo analysis can easily produce tens of thousands of rows of data, which is prohibitive based on our server. For that reason, the output only includes the basic summary statistics for each simulated sample. When the users download the files, the default name of the file includes the random seed value labeled as “RV_{seed value}” so that the user can more easily return to the app and replicate the results if they choose.

App Limitations

The app was designed to support a wide variety of single-subject experimental data. That is, the app makes no assumptions about the underlying structure of the experimental design or the resulting data structure. Although there are several benefits to using this app for data analysis, researchers should be aware that there are a few key limitations of our app. First, the app cannot conduct Monte Carlo simulations if the research question is related to paired differences between behavior across conditions because the app treats every measure of behavior as independent of other measures of behavior (in terms of using a response value as a data point). For example, a behavior analyst might be conducting functional communication training and want to compare data on outburst behavior to data of prosocial replacement behavior. The app could conduct a Monte Carlo analysis to determine if prosocial behaviors during a treatment condition were generally higher than outbursts during that condition. However, the app could not conduct a Monte Carlo analysis to determine if the differences between the outbursts and the prosocial behavior each session were reliably of a certain pattern (i.e., assess whether prosocial behavior occurs at a higher rate than outbursts for each session) because the app does not have the ability to track paired data. There are innumerable ways data could be paired or clustered and we could not determine a way to implement the ability to track all possible pairings of data. A behavior analyst could calculate those differences scores and then conduct a Monte Carlo analysis on difference scores (cf. Friedel, Galizio, et al., 2019b).

Another limitation of the app is that it may sometimes incorrectly display data imported into the app on the “Sample Selection” tab. The behavior analyst must supply a filter that will select the data they want to use as their experimental sample. Using the example from the preceding paragraph, the user would have to indicate they are interested in prosocial behavior during a treatment condition. We decided to include a graph of the behavior analyst’s data as a visual aid in selecting the correct filter so that they will be more confident that they are selecting the correct experimental data. We used a simplified graphing method to display a basic summary of the data which only includes a single connected data path for the behavior (one behavior occurring across sessions) and no other indicators (e.g., phase change lines). As mentioned above, the app assumes that all measures of behavior are

independent. If the user has multiple behaviors per session, the app will plot two or more responses for each session with more than one measure and will connect those two data points. If behavior is occurring in multiple conditions, such as with a multielement functional analysis, the app's simplified plot will only show a single data path of behavior. As we consider the graph a visual aid and not a publication worthy figure, we have made no attempt to perfectly plot the user's data.

Finally, it is possible that the app will not perfectly accommodate all possible single-subject experimental designs. There are many common single-subject designs that control for specific confounding variables (cf. Perone, 1991; Sidman, 1960). However, speaking in a broad sense, single-case designs represent a style of experimentation more so than a specific prescriptive set of experimental tools. Thus, single-subject designs are more flexible—which is good for experimental control but bad for designing a one-size-fits-all app—than the clearly predefined group design approaches. For example, a behavior analyst might have a multiple baseline; a multielement design; or a multiple baseline, with a multielement component only during the treatment phase, and also a third phase change that is staggered across subjects. We designed the app so that it can conduct Monte Carlo analyses on data from the most common forms of experimental designs (e.g., withdrawal and reversal designs, multiple baseline designs, alternating treatment designs) as well as derivatives of those designs. That being said, it is possible that a user had specific experimental design that we did not consider or account for, and that this app will not be able to properly analyze data from that design.

Discussion

We have designed an app that can conduct a Monte Carlo analysis on data obtained from a variety of single-subject experimental designs. The app compares the behavior from a sample (as specified by the behavior analyst) and simulates 1,000 samples by randomly selecting—with replacement—measures of behavior from all the other sessions in the data file. The percentage of simulated samples that are above/below the experimental mean can be reported as the p -value of a typical null-hypothesis significance testing. The user can also download a figure that shows a histogram of the means of the simulated samples, with the experimental sample clearly identified, and a file that includes the means and standard deviations for each simulated sample. The app requires no specific expertise in using statistical software nor skills in programming. The two largest potential barriers to using the app are properly formatting the data for input and indicating which data points are the sample of interest through the filtering selectize menus. Even with those potential barriers to using this app, we believe the app makes it far easier for a behavior analyst to conduct Monte Carlo analyses for single-subject experiments.

As with any analytical process there are some limitations. Researchers should be a little more cautious in accepting the results of a randomization test that relies on a Monte Carlo simulation relative to the caution when accepting the results of a true randomization test. With a true randomization test, a researcher can know the exact likelihood of their arrangement of data relative to all possible arrangements. With a

Monte-Carlo-based randomization test, the researcher can only know the likelihood of their arrangement of data relative to the number of samples that were simulated by the Monte Carlo process. For example, if a behavior analyst only simulated 10 samples, they could only draw conclusions about their experimental result relative to those 10 simulated samples. With so few simulations, at best we could say that they could be are 10% sure the results of their experiment were conclusive. Thus, we could not say anything definitive about our research question based on a Monte Carlo analysis with so few simulated samples. It is fortunate that researchers using Monte Carlo based methods can rely on the law of large numbers to increase the likelihood that a Monte Carlo based randomization test will produce a similar interpretation as that of a true randomization test. Instead of simulating 10 samples, we can simulate 1,000. With a large number of simulations and the law of large numbers, we can assume that any interpretation we reach will be the same as if we had compared our experimental result to all possible arrangements of the data (i.e., a true randomization test).

As always, we would caution users of this app that any fancy data analysis tool is no replacement for strong experimental design. The app is blindly comparing data from the sample of interest to sets of randomly sampled data. That is, the app—like all statistical tests—does not differentiate a strong experimental design and, for example, cannot detect that a researcher inappropriately used an A/B design when a multiple baseline was more appropriate. Any data analysis should only be thought of as a tool to determine if an effect on the dependent variable is reliably correlated with an independent variable. The experimental logic is what allows a researcher or practitioner to begin to attribute a causal functional relation between the independent variable and the dependent variable (Sidman, 1960).

Monte Carlo-based analyses that are based on randomization test logic can be a useful tool for behavior analysts conducting data analyses. Monte Carlo analyses can share some of the putative benefits of traditional null-hypothesis significance testing (cf. Craig & Fisher, 2019; Friedel, Galizio, et al., 2019b; Jacobs, 2019) but also can be useful for single-subject experimental designs commonly employed by behavior analysts. In addition, Monte Carlo analyses based on randomization tests are more conceptually sound and align better with behavioristic concepts in general (Jacobs, 2019). Finally, from a pragmatic standpoint, with many single-subject experimental designs, it can be prohibitively difficult for a nonexpert in statistics (see DeHart & Kaplan, 2019) to conduct a null-hypothesis significance test due to the autocorrelated nature of single-subject experiments. Users of this app can gain access to the benefits of Monte Carlo analyses based on randomization test logic. It is our belief that this app can meaningfully eliminate many of the barriers behavior analysts may face if trying to conduct a Monte Carlo analysis on their own to make data-based decisions about treatments.

Acknowledgments The authors thank Kenneth W. Jacobs for productive conversations about Monte Carlo analyses and randomization tests.

Availability of Data and Programming The app can be found at https://shiny.georgiasouthern.edu/BA_Monte_Carlo/. Data and programming code at the time of publication are archived on the Open Science Framework (<https://osf.io/gqtxz/>) and the programming code that is running the app will be maintained on GitHub (https://github.com/jefriedel/BA_Monte_Carlo).

Declarations

Conflict of Interest The authors declare no conflict of interest.

References

- Ator, N. A. (1999). Statistical inference in behavior analysis: Environmental determinants? *The Behavior Analyst*, 22(2), 93–97. <https://doi.org/10.1007/BF03391985>
- Baggio, A., & Langendoen, K. (2008). Monte Carlo localization for mobile wireless sensor networks. *Ad Hoc Networks*, 6(5), 718–733. <https://doi.org/10.1016/j.adhoc.2007.06.004>
- Berry, M. S., Sweeney, M. M., & Odum, A. L. (2014). Effects of baseline reinforcement rate on operant ABA and ABC renewal. *Behavioural Processes*, 108, 87–93. <https://doi.org/10.1016/j.beproc.2014.09.009>
- Branch, M. N. (1999). Statistical inference in behavior analysis: Some things significance testing does and does not do. *The Behavior Analyst*, 22(2), 87–92. <https://doi.org/10.1007/BF03391984>
- Branch, M. (2014). Malignant side effects of null-hypothesis significance testing. *Theory & Psychology*, 24(2), 256–277. <https://doi.org/10.1177/0959354314525282>
- Branch, M. N. (2019). The "reproducibility crisis:" Might the methods used frequently in behavior-analysis research help? *Perspectives on Behavior Science*, 42(1), 77–89. <https://doi.org/10.1007/s40614-018-0158-5>
- Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2020). shiny: Web application framework for R. *R package version 1.5.0*. <https://CRAN.R-project.org/package=shiny>
- Cooper, J. O., Heron, T. E., & Heward, W. L. (2020). *Applied behavior analysis* (3rd ed.). Pearson Education.
- Craig, A. R., & Fisher, W. W. (2019). Randomization tests as alternative analysis methods for behavior-analytic data. *Journal of the Experimental Analysis of Behavior*, 111(2), 309–328. <https://doi.org/10.1002/jeab.500>
- Crosbie, J. (1999). Statistical inference in behavior analysis: Useful friend. *The Behavior Analyst*, 22(2), 105–108. <https://doi.org/10.1007/BF03391987>
- DeHart, W. B., & Kaplan, B. A. (2019). Applying mixed-effects modeling to single-subject designs: An introduction. *Journal of the Experimental Analysis of Behavior*, 111(2), 192–206. <https://doi.org/10.1002/jeab.507>
- Ferron, J. M., Farmer, J. L., & Owens, C. M. (2010). Estimating individual treatment effects from multiple-baseline data: A Monte Carlo study of multilevel-modeling approaches. *Behavior Research Methods*, 42(4), 930–943. <https://doi.org/10.3758/BRM.42.4.930>
- Ferron, J. M., Joo, S. H., & Levin, J. R. (2017). A Monte Carlo evaluation of masked visual analysis in response-guided versus fixed-criteria multiple-baseline designs. *Journal of Applied Behavior Analysis*, 50(4), 701–716. <https://doi.org/10.1002/jaba.410>
- Field, A. (2018). *Discovering statistics using IBM SPSS statistics* (5th ed.). Sage Publications.
- Fisher, R. A. (1935). *The design of experiments*. Oliver & Boyd.
- Friedel, J. E., DeHart, W. B., & Odum, A. L. (2017). The effects of 100 dB 1-kHz and 22-kHz tones as punishers on lever pressing in rats. *Journal of the Experimental Analysis of Behavior*, 107(3), 354–368. <https://doi.org/10.1002/jeab.254>
- Friedel, J. E., DeHart, W. B., Foreman, A. M., & Andrew, M. E. (2019a). A Monte Carlo method for comparing generalized estimating equations to conventional statistical techniques for discounting data. *Journal of the Experimental Analysis of Behavior*, 111(2), 207–224. <https://doi.org/10.1002/jeab.497>
- Friedel, J. E., Galizio, A., Berry, M. S., Sweeney, M. M., & Odum, A. L. (2019b). An alternative approach to relapse analysis: Using Monte Carlo methods and proportional rates of response. *Journal of the Experimental Analysis of Behavior*, 111(2), 289–308. <https://doi.org/10.1002/jeab.489>
- Galizio, A., Frye, C. C. J., Haynes, J. M., Friedel, J. E., Smith, B. M., & Odum, A. L. (2018). Persistence and relapse of reinforced behavioral variability. *Journal of the Experimental Analysis of Behavior*, 109(1), 210–237. <https://doi.org/10.1002/jeab.309>

- Giannakakos, A. R., & Lanovaz, M. J. (2019). Using AB designs with nonoverlap effect size measures to support clinical decision-making: A Monte Carlo validation. *Behavior Modification*, 1–16. Advance online publication. <https://doi.org/10.1177/0145445519860219>
- Gilroy, S. P., & Hantula, D. A. (2018). Discounting model selection with area-based measures: A case for numerical integration. *Journal of the Experimental Analysis of Behavior*, 109(2), 433–449. <https://doi.org/10.1002/jeab.318>
- Haddock, J. N., & Iwata, B. A. (in press). Software for graphing time-series data. *Journal of Applied Behavior Analysis*.
- Hales, A. H., Wesselmann, E. D., & Hilgard, J. (2019). Improving psychological science through transparency and openness: An overview. *Perspectives on Behavior Science*, 42(1), 13–31. <https://doi.org/10.1007/s40614-018-00186-8>
- Horner, R. D., & Baer, D. M. (1978). Multiple-probe technique: A variation on the multiple baseline. *Journal of Applied Behavior Analysis*, 11(1), 189–196. <https://doi.org/10.1901/jaba.1978.11-189>
- Jacobs, K. W. (2019). Replicability and randomization test logic in behavior analysis. *Journal of the Experimental Analysis of Behavior*, 111(2), 329–341. <https://doi.org/10.1002/jeab.501>
- Killeen, P. R. (2019). Predict, control, and replicate to understand: How statistics can foster the fundamental goals of science. *Perspectives on Behavior Science*, 42(1), 109–132. <https://doi.org/10.1007/s40614-018-0171-8>
- Kratochwill, T. R., Hitchcock, J., Horner, R. H., Levin, J. R., Odom, S. L., Rindskopf, D. M., & Shadish, W. R. (2010). Single-case designs technical documentation. *What Works Clearinghouse*. https://ies.ed.gov/ncee/www/Docs/ReferenceResources/www_scd.pdf
- Kroese, D. P., Brereton, T., Taimre, T., & Botev, Z. I. (2014). Why the Monte Carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6), 386–392. <https://doi.org/10.1002/wics.1314>
- Kwak, Y. H., & Ingall, L. (2007). Exploring Monte Carlo simulation applications for project management. *Risk Management*, 9(1), 44–57. <https://doi.org/10.1057/palgrave.rm.8250017>
- Kyonka, E. G. E., Mitchell, S. H., & Bizo, L. A. (2019). Beyond inference by eye: Statistical and graphing practices in JEAB, 1992–2017. *Journal of the Experimental Analysis of Behavior*, 111(2), 155–165. <https://doi.org/10.1002/jeab.509>
- Lindsley, O. R. (1992). Precision teaching: Discoveries and effects. *Journal of Applied Behavior Analysis*, 25(1), 51–57. <https://doi.org/10.1901/jaba.1992.25-51>
- Long, C. G., & Hollin, C. R. (1995). Single case design: A critique of methodology and analysis of recent trends. *Clinical Psychology & Psychotherapy*, 2(3), 177–191. <https://doi.org/10.1002/cpp.5640020305>
- Marchant, N. J., Li, X., & Shaham, Y. (2013). Recent developments in animal models of drug relapse. *Current Opinion in Neurobiology*, 23(4), 675–683. <https://doi.org/10.1016/j.conb.2013.01.003>
- MathWorks. (2020). MATLAB (Version 9.9). The MathWorks, Inc. <https://www.mathworks.com/products/matlab.html>
- McCullough, B. D., & Heiser, D. A. (2008). On the accuracy of statistical procedures in Microsoft Excel 2007. *Computational Statistics & Data Analysis*, 52(10), 4570–4578. <https://doi.org/10.1016/j.csda.2008.03.004>
- Mitteer, D. R., Greer, B. D., Randall, K. R., & Briggs, A. M. (2020). Further evaluation of teaching behavior technicians to input data and graph using GraphPad Prism. *Behavior Analysis: Research and Practice*, 20(2), 81–93. <https://doi.org/10.1037/bar0000172>
- Moeyaert, M., Ugille, M., Ferron, J. M., Beretvas, S. N., & Van den Noortgate, W. (2013). The three-level synthesis of standardized single-subject experimental data: A Monte Carlo simulation study. *Multivariate Behavioral Research*, 48(5), 719–748. <https://doi.org/10.1080/00273171.2013.816621>
- Odom, A. L., & Shahan, T. A. (2004). d-Amphetamine reinstates behavior previously maintained by food: Importance of context. *Behavioural Pharmacology*, 15(7), 513–516.
- Onghena, P. (2018). Randomization tests or permutation tests? A historical and terminological clarification. In V. Berger (Ed.), *Randomization, masking, and allocation concealment* (pp. 209–227). Chapman & Hall/CRC Press. <https://doi.org/10.1201/9781315305110-14>
- Peng, C. Y. J., & Chen, L. T. (2018). Handling missing data in single-case studies. *Journal of Modern Applied Statistical Methods*, 17(1), Article eP2488. <https://doi.org/10.22237/jmasm/1525133280>
- Perone, M. (1991). Experimental design in the analysis of free-operant behavior. In I. H. Iversen & K. A. Lattal (Eds.), *Experimental Analysis of Behavior* (Part 1, pp. 135–171). Elsevier Science.
- Perone, M. (1999). Statistical inference in behavior analysis: Experimental control is better. *The Behavior Analyst*, 22(2), 109–116. <https://doi.org/10.1007/BF03391988>

- Pustejovsky, J. E. (2018). Using response ratios for meta-analyzing single-case designs with behavioral outcomes. *Journal of School Psychology*, 68, 99–112. <https://doi.org/10.1016/j.jsp.2018.02.003>
- R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing <https://www.R-project.org/>
- Shull, R. L. (1999). Statistical inference in behavior analysis: Discussant's remarks. *The Behavior Analyst*, 22(2), 117–121. <https://doi.org/10.1007/BF03391989>
- Sidman, M. (1960). *Tactics of scientific research*. Authors Cooperative.
- Smith, J. D., Borckardt, J. J., & Nash, M. R. (2012). Inferential precision in single-case time-series data streams: How well does the EM procedure perform when missing observations occur in autocorrelated data? *Behavior Therapy*, 43(3), 679–685. <https://doi.org/10.1016/j.beth.2011.10.001>
- Trafimow, D. (2019). A frequentist alternative to significance testing, p-values, and confidence intervals. *Econometrics*, 7(2), 26–40. <https://doi.org/10.3390/econometrics7020026>
- Wasserstein, R. L., Schirm, A. L., & Lazar, N. A. (Eds.) (2019). Special issue on statistical inference in the 21st century: A world beyond $p < .05$. *The American Statistician*, 73(1, suppl. 1).
- Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(10), 1–23. <https://doi.org/10.18637/jss.v059.i10>
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag <https://ggplot2.tidyverse.org>
- Wickham, H., François, R., Henry, L., & Müller, K. (2020). dplyr: A grammar of data manipulation. *R package version 1.0.2*. <https://CRAN.R-project.org/package=dplyr>
- Young, M. E. (2019). Modern statistical practices in the experimental analysis of behavior: An introduction to the special issue. *Journal of the Experimental Analysis of Behavior*, 111(2), 149–154. <https://doi.org/10.1002/jeab.511>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.