Recovering data from summary statistics: Sample Parameter Reconstruction via Iterative TEchniques (SPRITE)

James Heathers[1], Jordan Anaya[2], Tim van der Zee[3], and Nicholas J. L. Brown[4]

[1]**Northeastern University, Boston, MA**
[2]**Omnes Res, Charlottesville, Virginia**
[3]**Graduate School of Teaching (ICLON), Leiden University, Leiden, The Netherlands**
[4]**University Medical Center, University of Groningen, The Netherlands**

Corresponding author:
James Heathers[1]

Email address: jamesheathers@gmail.com

ABSTRACT

Scientific publications have not traditionally been accompanied by data, either during the peer review process or when published. Concern has arisen that the literature in many fields may contain inaccuracies or errors that cannot be detected without inspecting the original data. Here, we introduce SPRITE (Sample Parameter Reconstruction via Interative TEchniques), a heuristic method for reconstructing plausible samples from descriptive statistics of granular data, allowing reviewers, editors, readers, and future researchers to gain insights into the possible distributions of item values in the original data set. This paper presents the principles of operation of SPRITE, as well as worked examples of its practical use for error detection in real published work. Full source code for three software implementations of SPRITE (in MATLAB, R, and Python) and two web-based implementations requiring no local installation (1, 2) are available for readers.

Keywords: Statistics, Reproducibility, Replication, Reanalysis

INTRODUCTION

Many scientists agree that sharing of data corresponding to published research should be routine (Kuipers and van der Hoeven, 2009). This is a longstanding problem (e.g., Wolins, 1962) which persists to the present; even when researchers commit to future data sharing, requests to do so are typically honored in only around one quarter to one half of cases. For example, Wicherts et al. (2006) and Bakker and Wicherts (2011) reported receiving data from 38 out of 141 (23%) and 21 from 49 (43%) authors respectively, Vines et al. (2014) obtained variously 17% to 29% for papers in cohorts within the previous 10 years, two of us received 9 out of 21 (43%) requested data sets from articles published in the previous 5 years (Brown and Heathers, 2017), and most recently it was found that for a journal with a data availability policy only 65 out of 180 authors provided the requested materials (Stodden et al., 2018).

There are a variety of reasons why researchers may not share data. A researcher may wish to derive further value for the collected data without competition, may be required to invest a large amount of time ensuring data is appropriately collated and externally interpretable, may regard the data as sensitive, protected, or proprietary, may not have ethical or institutional approval to release the data, may be pressed for time, or may regard the request as hostile or vexatious. It is also possible that the data is simply no longer accessible; data preservation requirements for research after publication differ greatly between fields, are inconsistently enforced, and may be preserved in storage no longer accessible to researchers with transient employment. Thus, availability of data decreases with the age of any given publication (Vines et al., 2014).

Access to the data underlying scientific publications is arguably desirable for its own sake, as transparency is a core

scientific value. However, in some circumstances, the need to check the data becomes even more urgent and paramount, such as when a manuscript contains numerical errors or inconsistencies. The frequency of faulty statistical reporting is quite substantial, and includes erroneous means, standard deviations, *p* values, degrees of freedom, and other test statistics (Bakker and Wicherts, 2014, 2011; Brown and Heathers, 2017; Nuijten et al., 2016, 2017; Petrocelli et al., 2013; Veldkamp et al., 2014). When a potential error is detected, there is only so much one can do to verify the veracity of the claims without having access to the data. As discussed above, requesting data from authors has historically not proved to be an effective strategy. In the case of suspected errors, the authors of the problematic paper might be even less willing to share the underlying data if the people making the request are clear about their intention to investigate potential errors. A researcher will especially be reticent to share data when it has been selectively edited, selectively analyzed, falsified, or fabricated. In extreme cases, a researcher might not wish to admit that requested data corresponding to reported summary statistics never existed as described.

Consequently, there is a need to develop techniques that can be deployed to detect anomalies within formal scientific publications without access to the underlying data. Perhaps the most obvious example is the detection of duplication. In the biological and life sciences, duplication is commonly observed in the form of image manipulation, notably of Western blots (e.g., Bik et al., 2016). In the social sciences and humanities, duplication is usually observed as plagiarism, the unauthorized reproduction of text written by oneself or others without attribution (e.g., Friedman et al., 2009). For forensic meta-scientists or other interested parties, detecting duplication has the straightforward advantage over other methods that the anomaly may be demonstrated in absolute terms since the evidence of overlap between sources is usually irrefutable.

However, a variety of investigative techniques exist that are able to detect anomalies in published research via the analysis of expected/observed distributions, means, standard deviations, *F*, *t*, and *p* values, and the relationships between these elements (e.g., Brown and Heathers, 2017; Carlisle, 2012, 2017; Nuijten et al., 2016; Simonsohn, 2013; Sternberg and Roberts, 2006). Perhaps the most significant of these is Carlisle (2012), which was instrumental in the confirmation of fraud in the work of Yoshitaka Fujii, who holds the unfortunate distinction of having more research retracted from the formal scientific record than any other researcher in history (at present, more than 180 papers). Techniques for studying anomalies in descriptive and summary statistics are less powerful (by definition) than the reanalysis of raw data, but can still make significant contributions to improving the scientific literature.

SPRITE (Sample Parameter Reconstruction via Iterative TEchniques) is a technique for reconstructing potential discrete data sets starting only from a few pieces of basic summary information about a sample, namely the mean, the standard deviation (SD), the sample size, and the lower and upper bounds of the range of item values. SPRITE complements the existing tests known as GRIM (Brown and Heathers, 2017) and GRIMMER (Anaya, 2016), which are simple methods for detecting certain inaccuracies in published means (GRIM) and SDs (GRIMMER) using the granularity of their constituent values. Recently, both tests were central to the identification of problems in a series of papers in food psychology (van der Zee et al., 2017; Anaya et al., 2017). While they are theoretically instructive, in practice both GRIM and GRIMMER can only be applied to a restricted subset of published work since they typically require the per-cell sample size to be smaller than N=100, assuming that the descriptive statistics are reported to the usual two decimal places (dp). In contrast, SPRITE does not have the same intrinsic sample size limitations, and can be used with per-cell sample sizes in the thousands. Unlike GRIM and GRIMMER, SPRITE takes into consideration the range of possible values of the raw data. It can also identify cases in which the summary statistics are theoretically possible, but imply a highly skewed or otherwise unusual distribution of individual responses.

Background to Data Reconstruction

Within the social and medical sciences, measurements on a sample or samples are generally described in terms of mean and SD, the sample sizes, the relevant test statistic, and a *p* value derived from that test. While this appears to be widely regarded as an adequate description, it typically leaves us with very little indication of the distribution of the underlying data, because statistics pertaining to modality (e.g., skewness, kurtosis, bimodality) are much less commonly reported. As a consequence, mean and SD pairs that might appear reasonable at first sight may instead correspond to unusual or even impossible distributions.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

It is computationally expensive to fully enumerate a sample space. Figure 1A shows a small window of all mean/SD pairs assuming integer data with no range restrictions and a sample size of 10. The even horizontal spacing (at intervals of 0.1) is due to the granularity of the mean and is the basis of the GRIM test. Any reported means that are not multiples of 0.1 will fail the GRIM test; that is, GRIM inconsistencies occur in the horizontal spaces between dots. Like the means, the SDs also show a pattern, albeit a more complex one. The SD pattern depends on the column (i.e., the mean), but repeats every time the mean reaches the next integer value. For example, the SDs for mean/SD pairs with a mean of 3.0 are the same as the SDs for the mean/SD pairs with a mean of 4.0. These patterns, and their relationships to the means, are the basis of the GRIMMER test. Assuming the mean has passed the GRIM test, the SD can then be checked against the pattern (column of dots) for that mean; GRIMMER inconsistencies occur in the vertical spaces between the dots in the column. Note that the vertical spacing between the dots decreases with larger SDs. This is due to the properties of the square root operator (which converts variance to SD), and has the effect that it is more difficult to find inconsistent SDs when the SD is large.

However, integer variables are typically constrained (i.e., they cannot take on any arbitrary numerical value). Consider, for instance, a Likert-type item asking the opinion of a small sample of participants (N=10), on a scale from 1 to 5 (representing, for example, responses of "strongly disagree", "disagree", "neither agree nor disagree", "agree", "strongly agree"). If we now plot only the possible mean/SD pairs given the bounds of the scale, the shape of the plot resembles an umbrella, and we refer to it as an *umbrella plot*, as seen in Figure 1B. The limited range of values allowed on the scale (1–5) adds a new constraint: not only are only certain means and SDs possible, but also only certain mean/SD *pairs* are possible. The top part of the umbrella consists of samples with larger SDs, meaning that more values at both endpoints of the scale are present (i.e., 1's and 5's, as shown by the bar chart in the top left corner of the pane), while the bottom part consists of samples where only identical or consecutive values are included (e.g., all 3's and 4's, or only 2's). While GRIM and GRIMMER will accurately identify impossible mean/SD pairs within the body of the umbrella, they will not flag values outside of the umbrella or at the edges, as shown by the green box in Figure 1B.

Even if a mean/SD pair is possible, it can still be extremely unlikely that these values would appear in real-world data. The inset to Figure 1B shows the distribution of a mean/SD pair at the umbrella's edge. If a distribution was presumed normal, this distribution may be difficult to believe. Randomly generating one million data sets (with each response equally likely; Figure 1C) supports this notion, as the values towards the middle of the umbrella display reasonable means and SDs, and are the most likely. Others are equally unlikely—for instance, one million simulations under the above parameters produced no data sets consisting only of 1's, 2's, 3's, 4's, or 5's, highlighting just how unlikely it would be to observe them in a real-world sample even though such combinations are technically possible. The fact that SPRITE returns the distribution allows users to statistically or heuristically judge the "weirdness" for themselves. A series of results or articles with possible but highly unlikely distributions may warrant further investigation.

The SPRITE Algorithm

SPRITE can be thought of as a random walk through the possible data distributions in search of the correct SD. SPRITE starts by generating a random sample of item values with the correct mean (rounded to a user-specified precision), but an arbitrary SD. Randomly selected pairs of values within this sample are then adjusted, with one value being increased and the other decreased (a step in the random walk), such that the SD changes in the desired direction towards the target value, but the mean is unaffected. (Some implementations of SPRITE may occasionally adjust only one item value, rather than a pair; this allows the exact mean to vary slightly, provided that its value when rounded to the specified precision remains the same.) This process is iterated until a sample with the desired target SD (again, within the limits of the specified precision) is obtained, or the program reaches a predefined cycle limit without a solution being found. Depending on the implementation, a further "outer" loop may be applied to generate multiple unique solutions (where these exist), which when plotted in the form of bar charts can give a simple visual impression of the possible distributions of the items that match a given mean and SD.

It should be noted that the random nature of the iterative process means that SPRITE functions on a heuristic rather than an analytical basis. This means that any SPRITE result may or may not precisely represent the data set of interest. The solution(s) found by SPRITE are not guaranteed to be the same as the original data set; they merely share the same

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

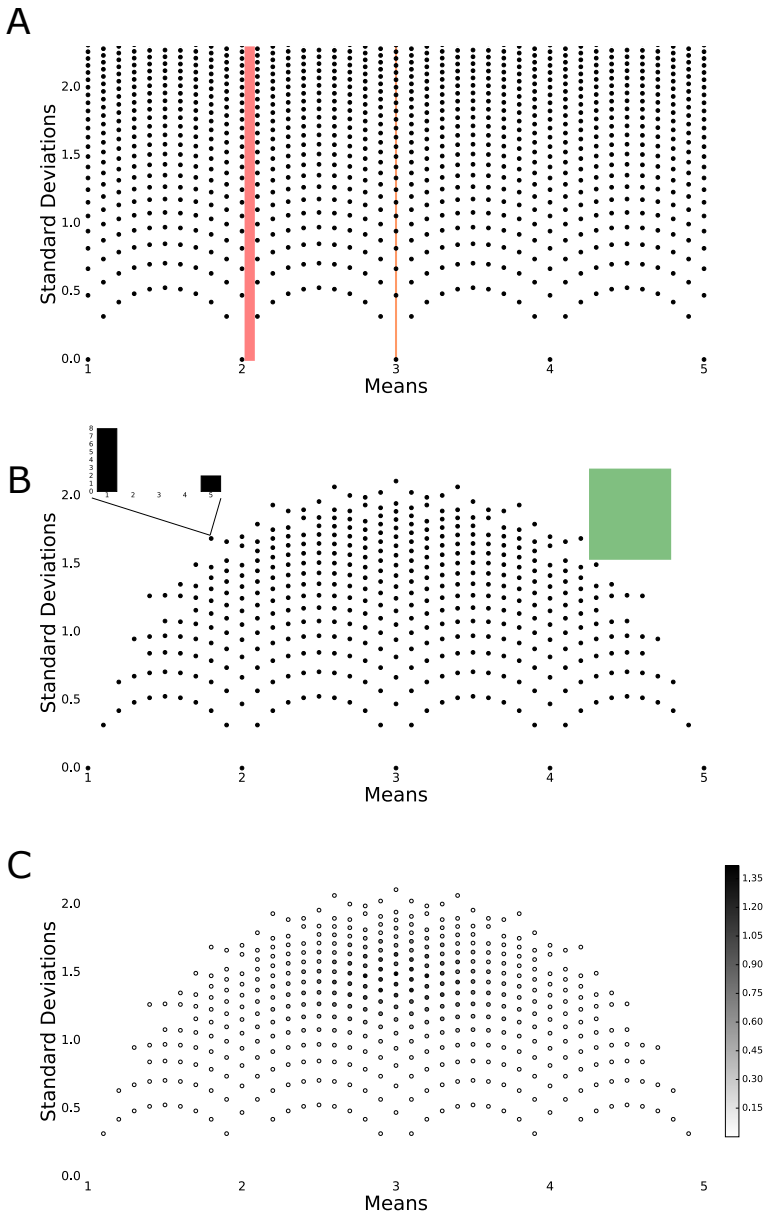

Figure 1. Umbrella plots and types of inconsistencies. (A) A small window of all possible mean/SD pairs for integer data. GRIM errors occur when the mean is impossible, and are indicated by a red bar. GRIMMER errors occur when the mean is possible, but the SD is not, and are indicated by all points not in black on the orange line. (B) Limiting the scale to 1-5 creates the characteristic umbrella plot. Values at the edges have extreme mean/SD values and represent unlikely but possible distributions, as shown in the inset. SPRITE errors include GRIM and GRIMMER errors, but also include values outside of the umbrella (e.g., in green) which would otherwise pass both tests. (C) Density plot of 1 million randomly generated data sets. The mean/SD pairs in the center of the umbrella are much more likely than pairs at the edges.

descriptive statistics (mean and SD). However, SPRITE is sufficiently rapid that if multiple unique solutions exist, generally several can be enumerated quickly. For the kinds of samples typically encountered in the social and behavioral sciences (N less than 1,000, Likert-type items in the range 1–7 or 0–10), SPRITE is quite efficient, typically finding a substantial number of solutions (where these exist) in just a few seconds on an ordinary computer.

The SPRITE algorithm can also be optimized for speed. For example, instead of starting from a random distribution it is possible to start from the distribution with the largest SD, or the distribution with the smallest SD. With large sample sizes, if the solution is highly skewed, starting from a random distribution will invariably require a large number of steps to reach the solution. However, a disadvantage of starting from a skewed distribution is that the results obtained will be biased towards solutions having the same sign of skew if there are multiple possible solutions. Another optimization that can dramatically improve performance with large sample sizes is performing more than one step per cycle; similarly, when dealing with wide scales (e.g., number of minutes spent exercising per week, which might range from 0 to over 1,000), performance can be improved by performing a larger "step", adding and subtracting more than 1 from each of the pair of values selected for modification in each cycle. While both of these optimizations may cause the generated SD to temporarily "overshoot" the target value, this will be naturally corrected by smaller steps in the later cycles. Currently, however, none of

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

the implementations of SPRITE of which we are aware (described later) perform either of these optimizations.

In addition to being inherently fast, with the possibility of improving the speed for edge cases, SPRITE is also extremely flexible. Specific values can be included or excluded (i.e., "no item value is 4") and set amounts or proportions of values can be added (i.e., "there are exactly three 4's"). Furthermore, the derived histograms can be investigated for outliers or normality, the procedure can be modified to include modal or median values, and so on. Even when SPRITE does not find a solution, SPRITE can return the closest distribution it found. This can give the user an idea of just how wrong the reported values are, by distinguishing between rounding errors and mean/SD pairs that are far removed from the possible values in the umbrella plot.

We have found that working through real data sets is the best way to explore possible use cases of SPRITE. The case studies below illustrate some situations where SPRITE proved to be an extremely effective tool.

CURRENT IMPLEMENTATIONS OF SPRITE

3 separate implementations (MATLAB, R, and Python) of SPRITE are presently available. Functions and implementations differ between versions, and are described below. All code described below can be downloaded from this OSF project.

mSPRITE

mSPRITE is a command-line implementation of SPRITE in MATLAB R2016B. It allows the user to specify (A) mean, (B) SD, (C) sample size, (D–E) a lower and upper bound for included values, (F) a rounding factor (i.e. a specified number of dp), (G) a matrix of numbers which must necessarily be included in the answer, (H) a starting distribution (i.e. the initial values are set to minimum, maximum, or pseudorandom SD), (I) the number of iterations the program should perform before timeout, and (J) a flag for diagnostic or error messages. It returns (i) a sample described by the parameters A through F or the closest possible approximation, (ii–iii) the mean and SD of this sample, (iv) the number of iterations run to return the solution, and (v-vi) the incremental SD values, and the number of "swaps/steps" taken to make them. The swapping procedure is random, not directed (see below), and mSPRITE will perform slower than rSPRITE (below) in general and on large samples in particular.

rSPRITE

rSPRITE is an implementation of SPRITE in the R programming environment (R Core Team, 2018). It consists of a single source file that uses the Shiny (Chang et al., 2017) runtime environment to provide the graphical user interface. The source code for rSPRITE can be run directly in RStudio on any computer once the referenced packages have been installed. Additionally, a hosted version of rSPRITE is available, allowing anyone with Internet access to run rSPRITE in a web browser window, even (with a limited number of results displayed) on the screen of a smartphone.

The user interface of rSPRITE requires the following items to be specified: minimum and maximum scale values, sample size, target mean and SD, precision (1, 2, or 3 dp), the maximum number of samples to be produced, and, optionally, a fixed count and item value (which allows the user to specify, for example, that all samples must contain exactly eight items with the value of 4). Additionally, the user can specify that a fixed seed be used for the random number generator, which provides a way of making the results exactly reproducible by others.

The output from rSPRITE consists of one or more bar charts (Figure 2), each of which contains a label that records the input parameters that produced it. The charts are ordered by ascending skewness from top left to bottom right. Additionally, a link is provided to allow the user to download the resulting sets of values in CSV format for subsequent analysis or checking.

pSPRITE

pSPRITE is implemented with Python 2.7. It consists of a single script, with one main function and a few minor helper functions. NumPy is needed for the random choices. In addition to the OSF SPRITE page linked above, the file is available at GitHub, and has also been made into a web application, available at PrePubMed, which is open source. An example of the web app output is shown in Figure 3.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

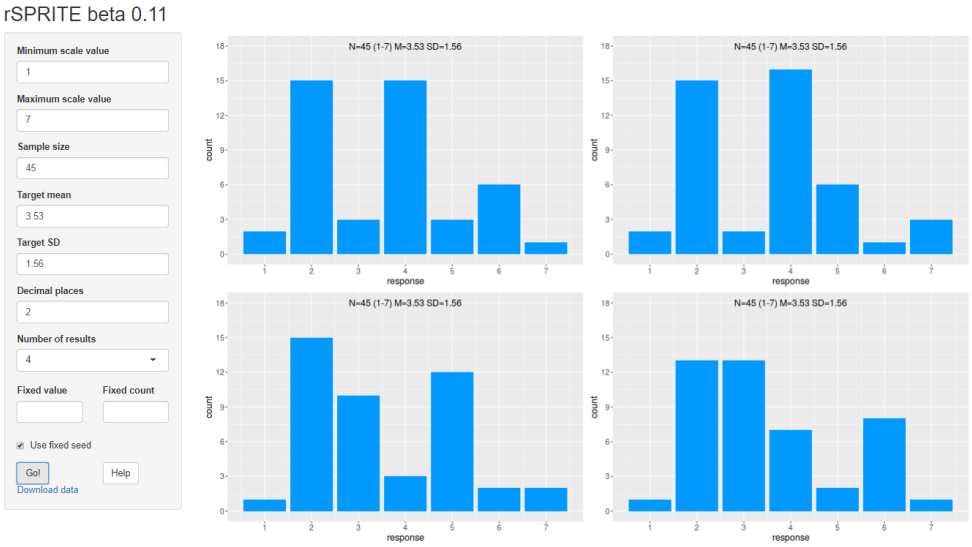

Figure 2. Shiny rSPRITE output. The left panel allows the user to enter the required sample parameters and descriptive statistics; the central panel displays the requested number of unique distributions.

pSPRITE differs from rSPRITE in that the mean and SD can be given different precision levels, the restrictions can be more than one value, and when a solution is not found the closest solution is returned. pSPRITE also has less built in randomness than rSPRITE, and users can decide if they want to start from a random distribution or the closest distribution of random, min, max to the target SD. While rSPRITE allows users to decide how many solutions to return, and the web version graphs these solutions, pSPRITE only returns one solution and if a user wants more they either have to write their own loop, or use the web app multiple times.

The output of pSPRITE is an indication of whether a solution was found, and the solution or closest solution in the form of a dictionary.

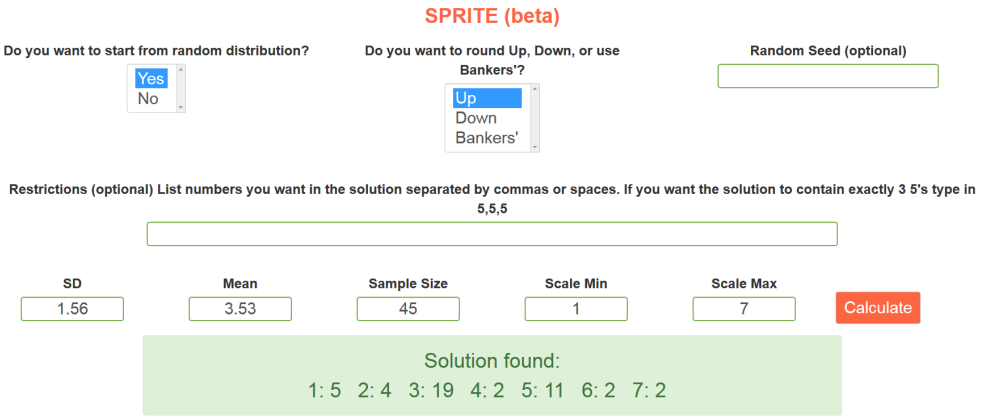

Figure 3. pSPRITE web app output The pSPRITE web app has a few more options than the rSPRITE app, but less output. How a user wants to round can be specified, whether a random distribution or closest to min, max, random to the SD should be used, individual decimal precision for mean and SD is inferred from the input, and restrictions do not have to be the same number.

Performance

SPRITE is a heuristic technique that does not fully explore the parameter space, and thus is not guaranteed to return all solutions. However, it almost always locates a solution when one exists. It is possible for the algorithm to get "stuck" and

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

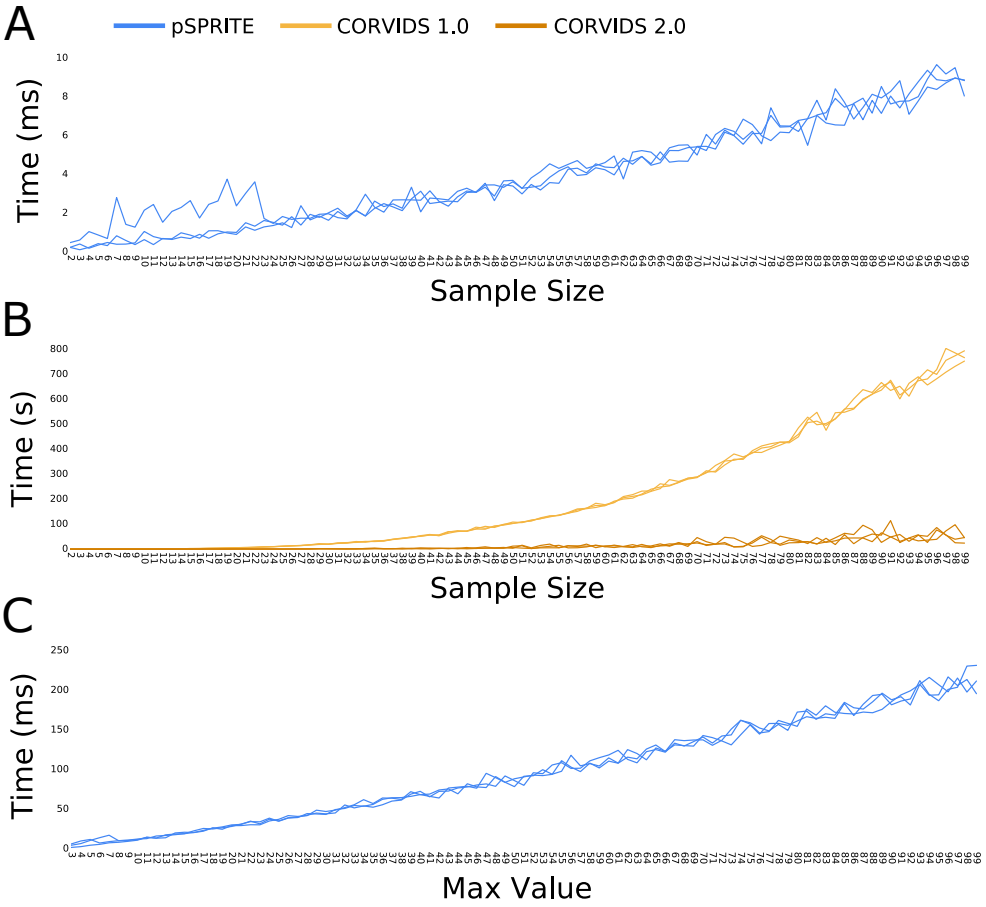

Figure 4. Performance characteristics of pSPRITE and CORVIDS. (A) SPRITE shows an approximately linear increase in run time when the sample size is increased with a constant scale (1-7). (B) In contrast, CORVIDS displays an exponential increase in time, although version 2.0 shows a significant improvement over version 1.0. (C) SPRITE also shows a linear increase in run time when scale is increased while keeping sample size constant (99).

not be able to perform productive swaps/steps, but this is rare under normative conditions. In general, in the unlikely event that a SPRITE run does not locate any solutions, the user can simply try again. When the number of restrictions is high relative to the sample size (i.e. there are fewer solutions to find), this reduces the paths SPRITE can take and increases the failure rate.

Recently, a method for returning all possible solutions given the summary statistics was published (Wilner et al., 2018). Referred to as COmplete Recovery of Values In Diophantine Systems or CORVIDS, this method utilizes a system of Diophantine equations as opposed to the heuristic approach presented here. While SPRITE cannot guarantee a solution does not exist when a solution is not found, CORVIDS can not only definitively say whether a solution exists, but return all possible solutions when one does. Unsurprisingly, given how different the two algorithms are and the goals of each (SPRITE tries to find one solution per run, while CORVIDS searches the entire solution space), the two algorithms have vastly different computational requirements.

In the social and behavioral sciences many data sets consist of Likert-type scales whose possible values are small integers (e.g., in the range 1–7), so it makes sense to give readers an idea of how these algorithms perform with this type of data. As seen in Figure 4A, pSPRITE (used for comparison since pSPRITE and CORVIDS are both written in Python) shows an approximately linear increase in run time when sample size increases from 2 to 99 (with mean and SD reported to 2 dp). Also implied in this figure, not a single run ($3X98 = 294$) failed—although failure does happen a very small percentage of the time. In contrast, CORVIDS displays an exponential increase in run time (Figure 4B decimal precision

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

.005)—while also having 0 failures (failure should be impossible for CORVIDS if the data can exist). Highlighting the difference between SPRITE and CORVIDS, it should be noted that SPRITE performs a little faster when the mean and SD are reported to low precision since this creates more solutions for SPRITE to run into, while this significantly slows CORVIDS since it has to find more solutions.

Some data sets are open-ended in that it is unclear what the upper bound of the data might be, or the upper bound can be artificially truncated at some extreme value. As a result it is worthwhile to investigate how well SPRITE might handle situations with a large number of possible values. Again, we see a linear increase in run time, as shown in Figure 4C (note that increasing both the sample size and the range of the data simultaneously will result in an exponential increase). A similar analysis could not be performed with CORVIDS given the length of the runs. Given this performance profile, we believe SPRITE will perform adequately for the needs of the majority of its users.

Use of SPRITE with non-integer data

Although the original purpose of SPRITE is to reconstruct integer values heuristically, it can also often be used to estimate the possible distributions of data that are not guaranteed to be integers, provided that the decimal fraction of any individual value can be neglected for practical purposes. For example, suppose that a researcher describes giving 50g of chocolate to 450 participants and measuring how much each of them ate by weighing the remainder on a scale with a precision of 0.1g. Thus, an individual participant might have eaten 12.7g or 45.2g of chocolate. The researcher reports that the mean amount eaten was 42.7g, with an SD of 7.4g. In such cases, provided that the number of solutions is not extremely small, SPRITE will generally still produce results that enable the user to understand the potential distribution, even though the simulated values are all integers. In this case, for example, it is clear that probably around a third of the participants ate all of the chocolate, and almost all of them ate at least half of it (Figure 5).

If greater precision is required, it is sometimes possible to simply multiply the decimal values to obtain an integer. Thus, in the example just mentioned, the problem could be expressed in terms of a mean of 427 decigrams with an SD of 74 decigrams. However, this will not always be possible, especially if the variable in question is measured to a high degree of precision, or calculated as the ratio of two measured quantities (which may result in arbitrarily long decimal fractions).

CASE STUDIES

It is easy to show that a tool can accurately reproduce data, but it is less clear how to enumerate the possible use cases for such a tool. By checking means, SDs, and test statistics we previously identified numerous anomalies in a series of publications (van der Zee et al., 2017), found similar and additional problems in other work from this lab, and these findings resulted in multiple retractions and corrections. Given the perceived likelihood that a paper from this lab might contain summary statistics that are impossible or improbable, we sought to see what SPRITE might reveal. The examples here are drawn from three papers, published within the last 15 years, with more than 1,000 collective citations at present according to Google Scholar on May 23, 2018.

Wansink, Just, Payne, & Klinger (2012) "Attractive Names Sustain Increased Vegetable Intake in Schools". 144 citations.

Wansink et al. (2012) investigated the effect of changing the names of vegetable dishes served to elementary school students. It was hypothesized and confirmed that labeling vegetables with attractive names (e.g., "X-ray Vision Carrots") increased the consumption of vegetables, compared to when the vegetables were labeled with neutral or no names. The paper contains two studies, the first of which is summarized in Table 1 below. The means and SDs could not be checked with GRIM and GRIMMER as they were only reported to 1 dp, but the size of the SDs relative to the means gave us cause for concern.

SPRITE: Although all of the mean/SD pairs in this table are unusual, the number of carrots taken in the control group stands out in particular. The mean/SD of 19.4/19.9 seems to imply that some children within the sample were taking a very large number of carrots without any intervention, so we decided to use SPRITE to explore the possible distributions of the number of carrots taken by each student.

Recreating this distribution without an upper bound restriction is potentially misleading, as the number of possible outcomes are enormous, and can produce highly unrealistic distributions. Instead, we can attempt to establish a minimum

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

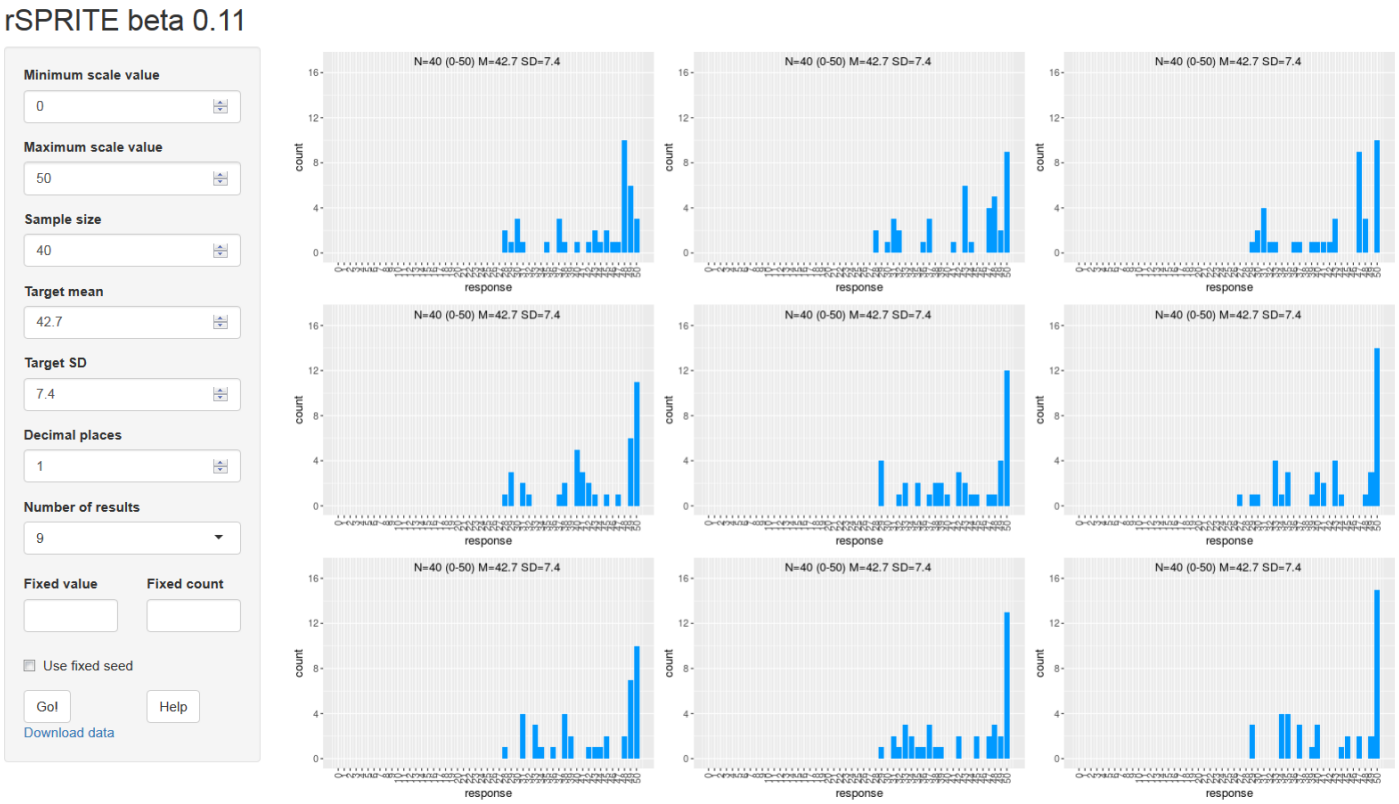

Figure 5. The rSPRITE interface for the example above. The solution histograms allow the user to infer approximations from multiple distributions.

Study 1: elementary students consumed more carrots when attractively named

	Named as "X-ray Vision Carrots" (*n*=32) Mean (SD)	Named as "Food of the Day" (*n*=38) Mean (SD)	Unnamed (Control) (*n*=45) Mean (SD)
Number taken	17.1 (17.6)	14.6 (14.5)	19.4 (19.9)
Number eaten	11.3 (16.3)	4.7 (6.7)	6.8 (8.7)
Number uneaten	6.7 (9.6)	10.3 (12.5)	13.2 (16.9)

Table 1. Reproduction of carrot data.

possible upper bound by running SPRITE with a range of maximum values (e.g. 30 to 80), and investigating the resulting distributions. Thus, we ran SPRITE under the reported parameters (mean=19.4, SD=19.9, N=45) at a series of ranges (0 to 30, 0 to 31, 0 to 32, etc.) and observed the difference between the reported SD and the intended SD (i.e. 19.9). At lower ranges, the sample does not contain a sufficient range to satisfy the SD. The lowest upper bound of the range at which a solution becomes possible is 41.

However, a range of 0 to 41 produces an unrealistic distribution, as it requires a large number of duplicate values at the upper bound (i.e. values are overwhelmingly either 0 or 41). As seen in Figure 6 below, the gradual relaxation of the upper limit lowers the amount of upper-bound and lower-bound values included in the distribution. While it might be expected that a large number of students will select 0 carrots, it is surprising to see how many carrots a number of students have to take to get the values to work.

OUTCOME: A series of concerns were related to the journal on March 17th, 2017, which was followed up on March 20th. These included the above point, in addition to the fact that the constituent means in the columns do not add up to their totals, the sample sizes change throughout the article, percent calculations in Study 2 were performed incorrectly, and

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

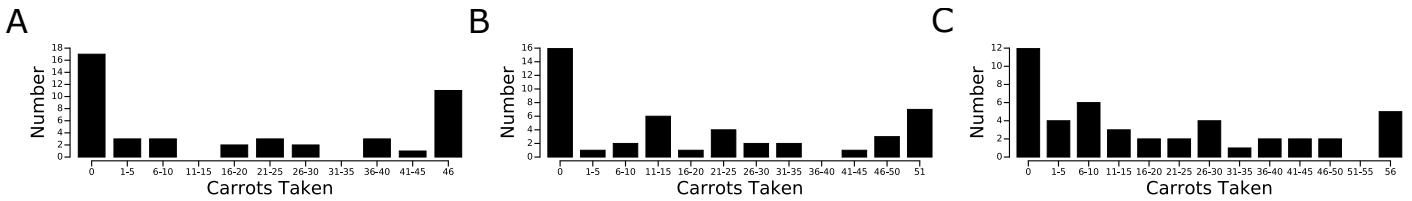

Figure 6. Possible data distributions. Possible distributions the carrot data can take for the control condition, taken (N=45, mean=19.4, SD=19.9). The maximum value for pSPRITE was set at 46 (A), 51 (B), or 56 (C).

questions about the study design. On October 18th, an article in BuzzFeed (Lee, 2017) revealed that the research described as taking place with 8 to 11 year old participants was in fact carried out with 3 to 5 year olds.

An extensive correction was issued in February 2018 (Wansink et al., 2018a), which altered a considerable amount of details, including increases to two of the cell sizes, all mean values within Table 1, the age and nature of the participants, and so on. References to "carrots" were also altered to "matchstick carrots" (i.e. a quarter carrot cut lengthwise), a detail unmentioned in the original paper which makes the above scenario a realistic upper bound. However, the predictive value of SPRITE on this upper bound cannot be checked against the original data, as only the updated and expanded data set has been released. The data release provoked further questions (included but not limited to an apparently incorrect description of the study procedure, a condition which was dropped without mention, an apparent lack of randomization in some subgroups, and a mismatch between the reported days of the study and the collected data) which were sent to the journal. The paper was then retracted due to a problem with the "funding attribution" (Wansink et al., 2018b).

Wansink, Cheney, & Chan (2003) "Exploring Comfort Food Preferences Across Age and Gender". 406 citations.

Wansink et al. (2003) outlines the basic relationships between "comfort food" preferences in standard demographic comparisons (i.e. men vs. women, young vs. old people). It reports that men prefer meal-based comfort food, while women prefer snack-based comfort food, and young people prefer more snack-based foods than old people.

Table 2
Females have different comfort food preferences than males

	Favorite comfort foods	Comfort food ratings			
		All (1003)	Females (602)	Males (401)	*F* values
Snack-related foods	Potato chips	3.0	3.0	3.1	1.8
	Ice cream	3.0	3.2	3.1	3.6
	Cookies	2.8	2.9	2.8	2.5
	Candy/chocolate	2.9	3.0	2.7	19.2
Meal-related foods	Pasta or pizza	2.8	2.7	2.9	5.5
	Steak or beef	3.0	2.8	3.2	17.8
	Casseroles/side dishes	3.0	2.9	3.1	5.7
	Vegetables or salads	2.3	2.4	2.3	3.8
	Soup	2.8	2.6	2.9	4.1

Table 2. Reproduction of comfort food data.

SPRITE: This article is atypical in that in several cases it describes the means and test statistics in the absence of the standard deviation. The combination of the sample sizes and lack of reported decimal precision prevents us from being able to use GRIM, and obviously the lack of SDs prevents use of GRIMMER. However, the test statistic is a function of

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

the means, SDs , and sample sizes. As result, the SDs can be estimated when the other two of these three parameters are reported.

In the table reproduced above (Table 2), two comparisons seem unduly large; women prefer candy/chocolate over men (mean=3.0 vs 2.7, F=19.2), and men prefer steak or beef over women (mean=2.8 vs 3.2, F=17.8). Focusing on the steak/beef comparison, it is easy to show that if we assume that the standard deviations are equal and that the reported means have not been greatly changed by rounding, a standard deviation of 1.47 for both means will produce the reported F statistic of 17.8 (if one of the SDs is smaller than 1.47, then the other will necessarily be greater). When analyzing cases such as this, it is important for investigators to consider whether imbalanced SDs could plausibly explain the reported results, although of course in many cases, such as the one-way ANOVA here, if the SDs vary widely then the original researchers should typically have reported this as a violation of the assumption of homogeneity of variance.

While this value of 1.47 might be a reasonable approximation of the true standard deviations for both men and women, it would arguably be unfair to the original authors to try and reproduce the data set based solely on this assumption. It is also possible to programatically cycle through all of the possible SD pairs for the two groups (measured to a specific precision) which produce the reported statistic, but this results in a very large number of possibilities. Hence, it is usually worthwhile to look for information in the text of the article that might put further constraints on the possible standard deviations.

Page 743 of Wansink et al. (2003) states:

> "Another way to examine the general tendency for males and females to rate comfort foods differently is to construct a surrogate measure of percentage acceptance by coding people who rated a food as 4 = *agree* or 5 = *strongly agree* as someone who accepts the food as a comfort food... In doing this, it is found that females had ... a lower acceptance percentage of meal-related foods such as steak or beef [52% vs. 77%]."

From this, we can infer that 52% of the females' ratings for steak or beef were 4 or 5, and 77% of the males' ratings were 4 or 5. With this additional constraint we can run SPRITE to see what range of standard deviations are reasonable, and then go back to see if those standard deviations are consistent with the test statistic.

Starting with the women, we can have SPRITE find the smallest and largest possible SDs given the restrictions. To get the smallest possible SD, we assume that all 52% gave a rating of 4 (which is closer to the reported mean of 2.8 than is 5), and then give SPRITE progressively smaller SDs until it can no longer find a solution. With this technique, we find that the smallest possible SD given the constraints is about 1.25. Similarly, we can find the largest possible SD by having as many women as possible respond with a rating 5 (this number is less than 52% of the number of women participants, as it is constrained by the mean, so at least some women must have responded with 4), then observing the largest SD that results in a SPRITE solution. The largest possible SD for which SPRITE finds a solution for here is 1.73. These most extreme distributions are shown in Figure 7 below. Notice that this range, 1.25-1.73, encompasses the estimated SD above (1.47). That is, the results for women can be shown to be mathematically possible as described (assuming that the range of SDs for men was similar to that for women), although the ratio of 4's to 5's might be considered somewhat improbable, especially for the smaller SDs.

For the men, the statements cited above from Wansink et al. (2003) require that 77% of the responses be either 4 or 5. However, even in the most favorable (for the original authors' claims) circumstance of all of these being 4's rather than 5's, with the rest of the responses being 1's and the most favorable possible rounding of the reported numbers, the mean must be larger than the reported mean of 3.2 (since 0.765*4 + (1 - 0.765)*1 = 3.295). As a result, at least one of the reported values for men (the mean rating of steak/beef, or the percentage who gave a response of 4 or 5) is impossible. It therefore makes little sense to explore the range of possible SDs with SPRITE, or to evaluate the plausibility of the test statistic. However, if SPRITE is run in this example, it will reveal the error without the above calculation by returning a solution which is a maximal approximation of the desired outcome (with 73.3% of men returning a value of 4). As this is less than 77%, SPRITE straightforwardly confirms that a solution cannot exist. All of the implementations of SPRITE described in the present article are able to detect some impossible situations before running the simulations, but the general problem is sufficiently complex that some such situations can only be identified by the repeated failure of SPRITE to find a solution

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

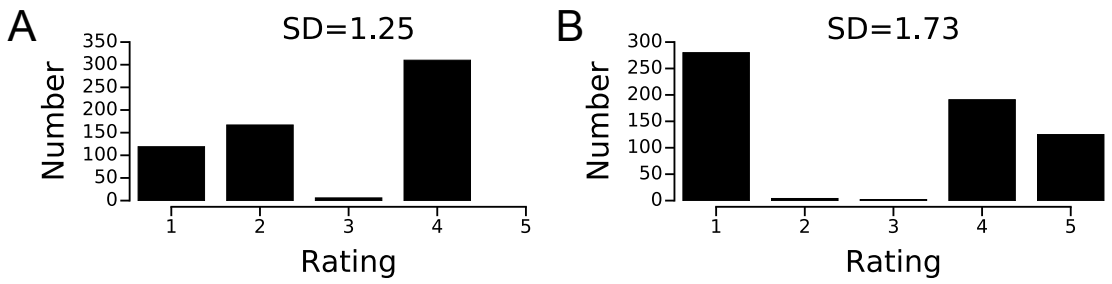

Figure 7. Most extreme distributions of responses by women. Given the constraints on the numbers of 4's and 5's, pSPRITE was used to investigate the smallest (A) and largest (B) possible standard deviations for the women's ratings of steak or beef as a comfort food.

as above (or, for complete certainty, by the use of mathematically complete solutions such as the one implemented by CORVIDS (Wilner et al., 2018).

OUTCOME: Our concerns indicating the above were made public on March 8th, 2017. The contact form for the journal was filled out on March 17th, and a reply indicating they would look into the matter and keep us informed was received on March 18th. Having received no information from the journal, the journal was contacted again on February 10th, 2018. We received a prompt reply on February 10th that a reminder would be sent to the lab, and that we would be informed if the lab responded (or didn't). We received no such update.

Wansink, Painter & North (2005) "Bottomless Bowls: Why Visual Cues of Portion Size May Influence Intake". 586 citations.

Wansink et al. (2005) investigated whether visual food cues interfered with the physical experience of satiety/fullness. In every experimental session, soup was served to 4 people (2 control, 2 experimental) in 18oz bowls. Experimental bowls were connected by a tube beneath the table to a whole separate pot of soup, which were slowly refilled imperceptibly through a gravity feed. This "bottomless bowl" was designed to decouple the experience of eating food from the appearance of the food. The salient result is that participants in the "bottomless" condition ate more food, but also dramatically underestimated their consumption. This interaction is the cornerstone of the "mindless eating" ethos—the participants perception of how much they have eaten and satiety is congruent to the amount they *appeared* to have eaten (Wansink, 2007).

SPRITE: This study reported the following table of values as reflective of the actual and estimated consumption within both groups (Table 3).

Table 1. Biased visual cues unknowingly influence overconsumption

	Visual cues of consumption		
	Accurate visual cue (normal soup bowls)	**Biased visual cue (self-refilling soup bowls)**	***F* test (1,52)**
Actual consumption volume			
Actual ounces of soup consumed	8.5 ± 6.1	14.7 ± 8.4	8.99
Actual calories of soup consumed	154.9 ± 110.3	267.9 ± 153.5	8.99
Estimated consumption volume			
Estimated ounces of soup consumed	8.2 ± 6.9	9.8 ± 9.2	0.46
Estimated calories of soup consumed	122.6 ± 101.0	127.4 ± 95.6	0.03

Table 3. Reproduction of soup data.

Elsewhere in the document, it is made clear that the relevant cell size for the "bottomless" condition is N=31, and the control group is N=23. The issue here is to reconcile the figures given in the table for the Estimated Consumption Volume with the text of the paper, which states "Indeed, of the 11 individuals who estimated they had consumed 16 or more ounces

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

of soup, only 2 of them were in the self-refilling condition". That is, 2 "bottomless" participants and 9 control participants estimated eating ≥16oz of soup.

To combine our sample parameters with the text above, SPRITE can be run with restrictions. That is, if we require n=2 participants with values of 16oz or more, these values can be exempted from the swapping procedure, and by adjusting the remaining values, the correct SD can be found. As in Case Study 1, we can run SPRITE under the reported parameters (i.e., for the "bottomless" estimation, mean=9.8 oz, SD=9.2oz, n=31) at a series of ranges (0 to 16, 0 to 17, 0 to 18, etc.) and observe the difference between the reported SD and the intended SD (i.e., 9.2). At an upper bound of 19, solutions become possible.

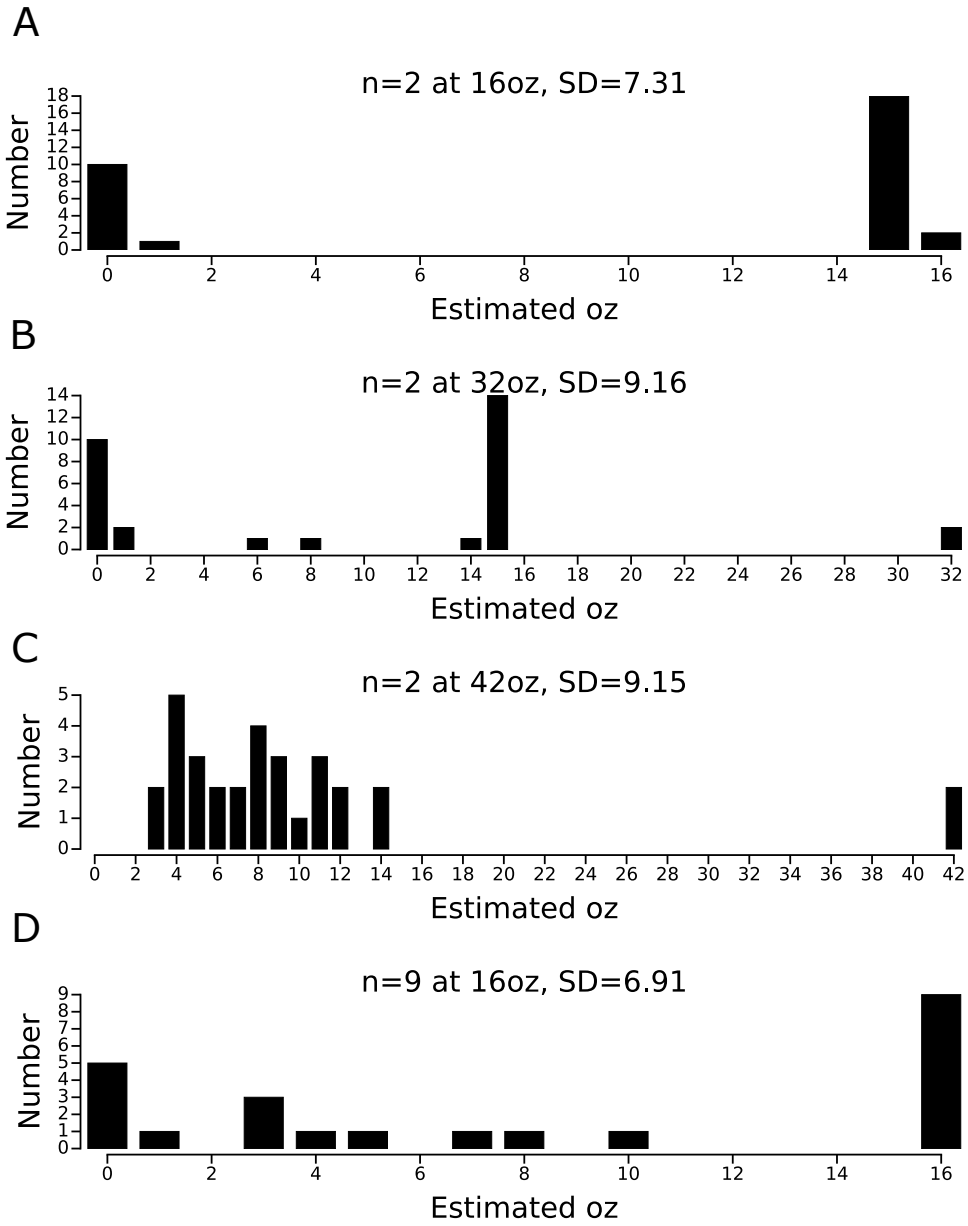

Figure 8. Possible soup distributions. pSPRITE was run with 2 data points set at either 16 (A), 32 (B), or 42 (C) for the estimated ounces of soup consumed in the "bottomless" condition. A solution was not found with the restrictions set at 16, but pSPRITE returned the closest solution. (D) A representative solution for the control condition under the most favorable assumption of n=9 at 16oz.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

However, a solution needs to be constructed where only n=2 values are equal to or greater than 16. To do this with SPRITE, we set those n=2 to appropriate values (i.e., 20, 21, 22, etc.) and observe the SDs again. At an upper bound of 31, solutions become possible but unrealistic. Further relaxing the upper bound results in more realistic distributions, however we must believe that there were two extreme outliers, and if these outliers exist, wonder why they weren't flagged during analysis (Figure 8A-D)

In the control group, the distribution is more problematic. Here, n=23 people estimated they ate 8.2oz of soup (SD=6.9), with n=9 of them estimated they ate 16oz or more of soup. The most favorable distribution is shown in Figure 8D, where all n=9 participants ate the minimum required; and SPRITE reveals this confines a lot of the sample to estimate they either consumed no soup, or very little soup (which is not mentioned in the paper). The real problem is how unrealistic it is that 9 participants estimated they ate exactly 16 ounces; if we attempt to move some of these responses much higher solutions become impossible. Hence, SPRITE reveals that neither the experimental nor the control distribution can exist without displaying extreme bimodality, in addition to extreme outliers in the experimental case, and a strange pileup of values near 16oz in the control case.

OUTCOME: Concerns with this paper were made public on February 9th. The journal has been contacted congruent to the release of the present document.

INTERPRETATION

In this section we provide some suggestions and cautions regarding the use and interpretation of SPRITE, and tools similar to SPRITE.

SPRITE is risky to use in isolation

As noted before, SPRITE does not rely on analytical procedures to generate all possible distributions that can produce a certain mean/SD combination. While SPRITE will usually generate most of the possible solutions, it is not guaranteed to give all solutions. Hence, we recommend the use of an analytical method such as CORVIDS (Wilner et al., 2018) to users who need a set of solutions that is guaranteed to be exhaustive. Additionally, in cases where SPRITE is unable to find any solutions after several runs, we strongly suggest using CORVIDS to confirm that there are no possible solutions for a given mean/SD combination. However, use of CORVIDS may not always be possible or practical, especially for large sample sizes; in such cases, we recommend prudence in making claims about the distributions found by SPRITE.

When SPRITE finds a range of solutions, it is up to the user to decide how to interpret these. For example, it is possible that all of the found solutions are highly skewed (typically with small SDs), or that they approximate bimodal distributions (typically with large SDs). While SPRITE can be a useful tool for detecting anomalies and impossibilities, it is by no means a sufficient one. SPRITE is useful for reconstructing possible data sets that can underlie reported summary statistics, which enables—but is not sufficient for—a consideration of the plausibility of the reported data and data generating procedures. In addition to a general caution in interpreting SPRITE, we have a number of specific recommendations.

Consider the data generating process

Proper reporting in a scientific article entails a thorough explanation of the process used to generate the data. That is, authors are required to make clear under what circumstances the data were produced and describe the relevant contextual factors in as much detail as is necessary. This should provide readers with an understanding of both the process of generation, and some insight into how the data might look. When making claims about the implications of summary statistics based on interpretation of solutions from SPRITE, it is necessary to consider this data generating process. That is, most SPRITE solutions cannot be interpreted on their own but only in relation to a reference class of distributions. For example, there is nothing inherently problematic about finding only skewed solutions if the data generating process will also typically only lead to skewed distributions (consider, for instance, a Likert item where increasing values represent the endorsement of a controversial opinion). Similarly, there will be cases when only finding approximately normally distributed data might be cause for further investigation—for example, if the variable of interest represents people's answers to a highly polarizing moral or political question. Whether these solutions are "odd" or "unlikely" cannot be determined by SPRITE, but will

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

depend on contextual factors. In short, users of SPRITE should hesitate before forming a judgment on the plausibility of distributions based solely on their shape, and should also use information about what kind of distributions are reasonable given the data generating procedure.

Normal distributions are not the norm

In most common applications of statistics, it is typical to deal with approximately normally distributed data. The sampling distribution of parameters such as sample means and residuals will converge to a normal distribution as sample size increases, which is useful for common tests such as ANOVA which requires residuals to be (approximately) normally distributed. While the central limit theorem is an explanation of why various kinds of aggregated data tend to be normally distributed, this is often not relevant to the distribution of raw data, which is what SPRITE deals with. For example, while the sampling distribution of dice throws is approximately normal, the distribution of the data is uniform. In this case, SPRITE should uncover many solutions with approximately uniformly distributed data. In short, we recommend readers to not start with the assumption that normally distributed data are the norm, but to carefully consider the data generation procedure and what kind of data it tends to generate.

Do not focus on individual solutions

When examining the output from SPRITE, we recommend to avoid focusing on the details of individual solutions and instead trying to understand the patterns that they form. For example, the entire set of solutions might consist purely of highly bimodal data distributions which, although containing many different combinations of item values, could be considered as one kind of solution. Similarly, in many cases the user will find that some solutions are negatively skewed, some are positively skewed, and the rest are approximately normally distributed. If the majority of individual solutions contain high skew, it becomes more reasonable to assume that the summary statistics may have been generated by highly skewed data. In general, how specific individual solutions appear is not as important as how various solutions look, and how many kinds of solutions there are. Only in cases where very few solutions exists do individual solutions become more relevant.

Note also that the distributions produced by SPRITE are not formally informative about the "probability" that the original data has any particular distribution. For example, suppose that an author reports that the ages of respondents varied from 20 to 40, and SPRITE finds 50 solutions for the given mean, SD, and sample size, only one of which contains any items with the value of 36. This does not mean that there is "only a 2% chance" that one or more of the participants was 36 years old, because the actual ages of the participants is a matter of fact, not of probability.

DISCUSSION AND CONCLUSIONS

SPRITE is a simple and flexible procedure which retrieves plausible distributions of data from summary statistics. The utility of reconstructing distributions from mean/SD/N extends to finding congruent distributions of a certain skew or kurtosis, estimating the absolute boundaries of unreported test statistics, determining realistic upper or lower bounds in unbounded data, finding exact or plausible test statistics given other parameters, retrieving unreported standard deviations when only means and test statistics are given, and so on. While all of these presume between-subjects data, SPRITE can also be used to retrieve realistic within-subject or repeated-measures data. In a paired-samples *t* test (for example, comparing the same subjects at Time 1 and Time 2) the individual SDs and test statistic are generally reported, which allows a trivial retrieval of the numerator of the test statistic (i.e., the difference score) if the nature of the paired test is known (e.g. equal or unequal variances). This score can then be subjected to SPRITE, and mapped iteratively to a second SPRITE distribution of the Time 1 score.

A series of incremental improvements to the run-time and flexibility are possible, as SPRITE spends almost all its computational time in the swapping procedure. Presently, all versions (a) perform one swap at a time, (b) swap by a unit of 1 every time (or 2 in the case of rSPRITE), and (c) determine one solution set at a time. A future version could dramatically reduce run-time by swapping an amount of values proportional to the difference between the starting solution and the eventual solution, or by larger numbers. An example: if we consider normative IQ scores in a large sample (n=500, mean=100, SD=15), and begin the procedure from mean=100 and SD=0, shuffling a single value one unit will initially

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

produce SD=0.06. Run on mSPRITE, it takes 179 successful individual swaps to reach SD=1, and 5681 to reach SD=15. This takes approximately 3.5 seconds - viable for a single solution, but not for any simulation where thousands of runs are necessary. Another possibility is modeling additional unique solutions using existing solutions as a template, which will more quickly enumerate other similar but unique solutions. However, none of the above are required to provision solutions for sample parameters typical to the behavioral and medical sciences.

In an ideal world, research would be transparently reported and provide direct access to the data underlying scientific claims, and thus SPRITE would be of greatly reduced relevance. However, given the high prevalence of reported statistical inconsistencies (e.g., Bakker and Wicherts, 2011) and the low prevalence of open or accessible data (Stodden et al., 2018), there is a pressing need for a tool which can "open the box" of summary statistics reported in scientific papers. Subject to certain restrictions, SPRITE can perform this task. In the examples presented above, it has played an important role in investigating a range of inconsistencies within the corpus of a single researcher, the accuracy of whose work is in broader question; at the time of writing, we know of 6 expressions of concern, 8 retractions, and 15 corrections.

A limitation of all of the current implementations of SPRITE is that they do not explore the range of possible samples in a uniform way. For example, with a range of 1–7, N=20, M=3.45, SD=2.06, and precision of 2 dp, CORVIDS shows 107 unique solutions made from whole units. If a user repeatedly asks mSPRITE, rSPRITE, or pSPRITE for, say, 100 of these solutions, some will be returned more often than others, because of the way in which each version generates candidate sequences of item values. However, in practice, we do not consider that this is likely to be a severe limitation. If the number of possible solutions is small—say, less than 20—SPRITE will generally find all of them rapidly. On the other hand, if the number of solutions is larger, the importance of any one solution is reduced (and it will probably be quite similar to a solution that *is* found). It is important to keep in mind that SPRITE is a practical tool, not a complete mathematical solution, and some degree of interpretation of its output will always be required.

While we hope that SPRITE will become part of the regular toolbox that editors, reviewers, and readers will employ, we would prefer it to become redundant in the review process—as open scientific data becomes the norm, SPRITE's use may eventually be confined to examining scientific work of the past whose underlying data may never become available.

ACKNOWLEDGMENTS

We would like to gratefully acknowledge the assistance of Sean Wilner and Katherine Wood in the preparation of this manuscript.

COMPETING INTERESTS

JA operates omnesres.com, oncolnc.org, and prepubmed.org. TvdZ has a blog entitled "The Skeptical Scientist" at timvanderzee.com. NJLB has a blog at sTeamTraen.blogspot.com that hosts advertising; his earnings in 2017 were €26.03.

REFERENCES

Anaya, J. (2016). The grimmer test: A method for testing the validity of reported measures of variability. *PeerJ Preprints*.

Anaya, J., Vander Zee, T., and Brown, N. (2017). Statistical infarction: A postmortem of the cornell food and brand lab pizza publications. *PeerJ PrePrints*.

Bakker, M. and Wicherts, J. M. (2011). The (mis) reporting of statistical results in psychology journals. *Behavior research methods*, 43(3):666–678.

Bakker, M. and Wicherts, J. M. (2014). Outlier removal and the relation with reporting errors and quality of psychological research. *PLoS One*, 9(7):e103360.

Bik, E. M., Casadevall, A., and Fang, F. C. (2016). The prevalence of inappropriate image duplication in biomedical research publications. *MBio*, 7(3):e00809–16.

Brown, N. J. and Heathers, J. A. (2017). The grim test: A simple technique detects numerous anomalies in the reporting of results in psychology. *Social Psychological and Personality Science*, 8(4):363–369.

Carlisle, J. (2012). The analysis of 168 randomised controlled trials to test data integrity. *Anaesthesia*, 67(5):521–537.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018

Carlisle, J. B. (2017). Data fabrication and other reasons for non-random sampling in 5087 randomised, controlled trials in anaesthetic and general medical journals. *Anaesthesia*, 72(8):944–952.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2017). shiny: Web application framework for r [computer software]. *URL http://CRAN. R-project. org/package= shiny (R package version 1.0. 0)*.

Friedman, R., Dougherty, M., and Harsting, P. (2009). 40 cases of plagiarism. *Bulletin de Philosophie Medievale*, 51:350–391.

Kuipers, T. and van der Hoeven, J. (2009). Parse.insight: Insight into issues of permanent access to the records of science in europe. https://libereurope.eu/wp-content/uploads/2010/01/PARSE.Insight.-Deliverable-D3.4-Survey-Report.-of-research-output-Europe-Title-of-Deliverable-Survey-Report.pdf.

Lee, S. M. (2017). Here's how a controversial study about kids and cookies turned out to be wrong — and wrong again. *BuzzFeed*, https://www.buzzfeed.com/stephaniemlee/who-really-ate-the-apples-though.

Nuijten, M. B., Borghuis, J., Veldkamp, C. L., Dominguez-Alvarez, L., Van Assen, M. A., and Wicherts, J. M. (2017). Journal data sharing policies and statistical reporting inconsistencies in psychology. *Collabra: Psychology*, 3(1).

Nuijten, M. B., Hartgerink, C. H., van Assen, M. A., Epskamp, S., and Wicherts, J. M. (2016). The prevalence of statistical reporting errors in psychology (1985–2013). *Behavior research methods*, 48(4):1205–1226.

Petrocelli, J. V., Clarkson, J. J., Whitmire, M. B., and Moon, P. E. (2013). When ab≠ c–c': Published errors in the reports of single-mediator models. *Behavior research methods*, 45(2):595–601.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Simonsohn, U. (2013). Just post it: The lesson from two cases of fabricated data detected by statistics alone. *Psychological science*, 24(10):1875–1888.

Sternberg, S. and Roberts, S. (2006). Nutritional supplements and infection in the elderly: why do the findings conflict? *Nutrition journal*, 5(1):30.

Stodden, V., Seiler, J., and Ma, Z. (2018). An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11):2584–2589.

van der Zee, T., Anaya, J., and Brown, N. J. (2017). Statistical heartburn: an attempt to digest four pizza publications from the cornell food and brand lab. *BMC Nutrition*, 3(1):54.

Veldkamp, C. L., Nuijten, M. B., Dominguez-Alvarez, L., van Assen, M. A., and Wicherts, J. M. (2014). Statistical reporting errors and collaboration on statistical analyses in psychological science. *PloS one*, 9(12):e114876.

Vines, T. H., Albert, A. Y., Andrew, R. L., Débarre, F., Bock, D. G., Franklin, M. T., Gilbert, K. J., Moore, J.-S., Renaut, S., and Rennison, D. J. (2014). The availability of research data declines rapidly with article age. *Current biology*, 24(1):94–97.

Wansink, B. (2007). *Mindless eating: Why we eat more than we think*. Bantam.

Wansink, B., Cheney, M. M., and Chan, N. (2003). Exploring comfort food preferences across age and gender. *Physiology & behavior*, 79(4-5):739–747.

Wansink, B., Just, D. R., Payne, C. R., and Klinger, M. Z. (2012). Attractive names sustain increased vegetable intake in schools. *Preventive medicine*, 55(4):330–332.

Wansink, B., Just, D. R., Payne, C. R., and Klinger, M. Z. (2018a). Corrigendum to "attractive names sustain increased vegetable intake in schools"[prev. med. 55 (4)(2012) 330-332]. *Preventive medicine*.

Wansink, B., Just, D. R., Payne, C. R., and Klinger, M. Z. (2018b). Retraction notice to "attractive names sustain increased vegetable intake in schools"[prev. med. 55/4 (2012) 330-332]. *Preventive medicine*, 110:116.

Wansink, B., Painter, J. E., and North, J. (2005). Bottomless bowls: why visual cues of portion size may influence intake. *Obesity*, 13(1):93–100.

Wicherts, J. M., Borsboom, D., Kats, J., and Molenaar, D. (2006). The poor availability of psychological research data for reanalysis. *American Psychologist*, 61(7):726.

Wilner, S., Wood, K., and Simons, D. (2018). Complete recovery of values in diophantine systems (corvids). *PsyArXiv*.

Wolins, L. (1962). Responsibility for raw data. *American Psychologist*, 17(9):657–658.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26968v1 | CC BY 4.0 Open Access | rec: 30 May 2018, publ: 30 May 2018