Complete Recovery of Values In Diophantine Systems (CORVIDS)

Sean A. Wilner (swilner2@illinois.edu)
Katherine Wood (kmwood2@illinois.edu)
Daniel J. Simons
University of Illinois at Urbana-Champaign

Introduction

Floods, server fires, lost punch cards, retired collaborators, the passage of time–a diverse set of disasters can destroy a data set. If the only copy of an old set of data sat locked in the basement of a since-demolished building, it would seem to be gone for good; often the only trace left behind is a set of summary statistics reported in a paper. A mean, a standard deviation, and a sample size may be all that we have of the original data.

For a certain type of data, these descriptive statistics actually contain enough information to rebuild each and every possible distribution of the raw data. Ordinal and categorical data, in which a response is limited to one value on a fixed scale (such as an integer from 1 to 7, where 1 may be "strongly disagree" and 7 may be "strongly agree"), have a well-constrained structure. Responses can take on a restricted, known range of values. The limits and precision of this scale, coupled with the sample size, mean, and standard deviation of the responses, are sufficient to define the entire set of possible response patterns that could generate those summary statistics. Importantly, this can be done in a closed-form manner which can be mathematically proven to find all possible solutions where such solutions exist, and to describe characteristics of the data which must be true.

The math dates back to the 3^{rd} century mathematician Diophantus of Alexandria. He studied equations that have come to be known as *Diophantine equations*. These are polynomial equations restricted to the set of integers. Setting up a system of these equations to solve for a set of unknowns that satisfy the mean, standard deviation, and sample size of a set of ordinal data yields a complete solution class for the set of response patterns that satisfy those summary statistics. With additional manipulations of this solution class, it is possible to precisely describe the characteristics of the data that generated them.

We developed an automated approach to this data reconstruction. The CORVIDS (COmplete Recovery of Values In Diophantine Systems) analysis relies on a system of Diophantine equations to solve for the number of subjects who gave each level of response. With the resulting solution class, it is possible to precisely characterize the data and enumerate every and all solutions which generate a set of summary statistics. Using this tool, one can recreate raw data from reported statistics in cases where the original data are not available, explore the properties of the data sets which satisfy constraints, and experiment with how the solution space is affected as the distribution parameters change.

Diophantine Equations

Diophantine equations take the form of

$$\sum_{e\in\mathbb{N}^n}\left(a_e\cdot\prod_{j=1}^{n}x_j^{e_j}\right)=C$$

where C is some constant, the a_ns are integer values, e_j is the j^{th} element of e, and the x_js are the j^{th} variables which can take integer values. In simple English, this is all possible n-variable polynomials restricted such that all numbers involved must be integers. Some famous examples include Pythagorean triples:

$$x_0^2+x_1^2-x_2^2=0 \qquad \text{e.g. } 3^2+4^2=5^2$$

and Fermat's Last Theorem*:

$$x_0^n+x_1^n-x_2^n=0 \qquad n>2$$

Linear equations, rather than general higher order polynomials, are of special interest to us since all linear Diophantine equations can be directly solved, a luxury not guaranteed for higher order Diophantine equations (Matiyasevich,1972).

Data Recreation

In order to use Diophantine Equations to recreate data, we must first construct equations which represent the summary statistics over our data. A natural approach to this would involve treating each data point as a variable. However, doing so results in a quadratic equation due to the squared nature of the variance calculation. In order to construct exclusively linear equations, the variables we consider are instead the number of times each value of the scale appears in the data. For example, if the data range over a scale from $1-7$, we need only 7 variables where each corresponds a number in the range $1-7$ and whose value is the number of subjects who selected that value. Using these variables, summary statistics can be represented by three equations:

$$\sum_{x_i\in\mathbb{S}}x_i=n \tag{1}$$

$$\sum_{x_i\in\mathbb{S}}i\cdot x_i=n\cdot m \tag{2}$$

$$\sum_{x_i\in\mathbb{S}}(n\cdot m-n\cdot i)^2\cdot x_i=v\cdot(n-1)\cdot n^2 \tag{3}$$

*Fermat's last theorem is specifically that no such solutions exists.

where $\mathbb{S}$ is the set of possible values in our dataset, x_i is the variable corresponding to the value i, m is the mean, v is the variance, and n is the total number of elements of our dataset. Equation 1 constrains our solution sets to have the proper number of elements, equation 2 constrains the sets to sum to the average of the values multiplied by the number of values, and 3 constrains the solutions to have the correct variance. The mean and variance are transformed from the summary statistics to achieve integer values.

This set of linear Diophantine Equations can be solved to yield the potentially infinite set of all possible integer solutions. To solve the system, we use the Hermite Normal Form for the matrix described by our system of equations, a technique laid out in Havas, Majewski, and Matthews (1998).

The result of the method described in Havas et al. (1998) is an initial solution (where any could possibly exist) to the system of equations and a basis for the vector space of all transformations to that solution which still satisfy the system of equations. Worth noting is that despite the fact that this transformation vector space is infinite (and thus our set of solutions is infinite), the set of actual solutions is bounded since each variable corresponds to the total number of occurrences of a given value in the data set. Valid solutions therefore cannot have negative values for any variable. That is, many (of the infinite) solutions produced by this method imply a negative number of occurrences of values, and thus are invalid solutions to our summary-statistics equations.

However, this technique is useful even taking the negative values into consideration, since if it fails to find a solution even while allowing negative values, no solution can exist when restricting solutions to contain only positive values.

Tolerance and rounding

CORVIDS requires extreme precision to find valid solutions, often more than would typically be reported. Further, rounding errors may occur when statistics are reported. We circumvent these issues by adding adjustable tolerance to both the mean and variance term. Tolerance puts an envelope around the mean and variance, and the algorithm will search for every viable combination of mean and variance within those envelopes and attempt to find solutions for all valid pairs.

To find valid means, the program first calculates the following range:

$$m = n(m_{reported} \pm \eta)$$

where n is the sample size, $m_{reported}$ is the provided mean, and η is the tolerance. Any whole number in this range is a mean that could have been generated by the data.

To find valid variances, the program makes use of a mathematical relationship that defines a possible variance for a given mean and sample size. Specifically, all valid variances can be reached by starting at a valid variance and moving in a step size of

$$\Delta v = \frac{2}{n-1} \tag{4}$$

(For a proof, see Appendix B).

Let $k = m(n) \mod n$ [†] and $s = \frac{2}{n-1}$, where n is the sample size and m is a valid mean. Then a valid initial variance is:

$$v_{init} = \frac{(n-k)(m-\lfloor m \rfloor)^2 + (k(\lceil m \rceil - m)^2)}{n-1} \tag{5}$$

(for a proof see Appendix C) and following from Equation 4, we know that all valid variance must have the form

$$\begin{aligned} v &= v_{init} + p \cdot \Delta v \\ &= \frac{(n-k)(m-\lfloor m \rfloor)^2 + (k(\lceil m \rceil - m)^2) + 2p}{n-1} \end{aligned}$$

for some integer p.

Even for extremely large tolerances, utilizing this relationship allows the program to rapidly determine the valid mean/variance pairs. For each pair, CORVIDS checks if any solution exist for the pair, even if it initially contains negative values; if not, the pair is discarded. If it does have a solution, a complete solve is then performed to see if it has a positive solution. These solutions are returned.

Bounds and required values

Using the same procedure, it is possible to determine whether there are certain values that must be present for a solution to be possible. We eliminate certain values or ranges of values from the scale, and solve the system again for the same sample size, mean, and variance. If, after eliminating portions of the scale, solutions do not exist, it means that there must be values there in all distributions.

For example, if a given mean, standard deviation, and sample size are no longer possible if the scale is restricted from $2-7$ instead of $1-7$, then we can say with mathematical certainty that for the data to be possible, there must be responses with a value of 1.

Forbidden values

A similar logic applies to finding values which are not possible. Instead of eliminating possible values, however, they are fixed at given points. Then, a system of equations is set up to solve for the new mean and variance with the given values fixed. If no solutions are possible, it means that the data cannot have values at the specified points.

For instance, data that are close to floor and have a very small mean and standard deviation may become impossible if a value is fixed at the upper end of the scale. This means that there are no solutions which allow for a response at that value, for any possible data.

Enumeration of all possible solutions

If the above manipulations do not sufficiently constrain the possible generating data, it is possible (often at appreciable computational expense) to exhaustively enumerate all possible data which would generate the given summary statistics.

[†] NB: $m(n)$ is not necessarily $0 \mod n$ since m is not necessarily an integer

By manipulating the basis vectors that comprise the solution space, a provably complete list of all positive solutions to the set of equations can be generated. A complete formal proof is give in Appendix A, but the general thrust of the proof is as follows:

1. Linearly combine basis vectors to produce an alternate basis such that each basis vector has one dimension on which it uniquely acts with value one (all other basis vectors are zero on that dimension)

2. Linearly combine the new basis with the initial (potentially negative) solution vector such that the new solution vector is zero at every dimension which is uniquely acted upon by some basis vector

3. We can now combine the new basis vectors with our new solution in a bounded fashion to produce all viable solutions

4. Since the sum of all dimensions on which *no* basis vector uniquely acts is -1 for each basis vector and *n* for our initial solution, our bound is a total of *n* additions of basis vectors since any more additions of basis vectors would produce a negative sum along the dimensions not uniquely acted upon (and thus a negative number). (N.B. no basis vector can be subtracted since this would introduce a negative value along the dimension on which it uniquely acts)

Examples

The case of the skewed data

Let us consider results from a hypothetical survey. 20 subjects were asked to rate how much they liked vegetables as a child on a 1 to 7 scale. Using R (R Core Team,2017) and the 'truncnorm' package (Trautmann, Steuer, Mersmann, & Bornkamp,2014), we generated data with a strong skew towards the low end of the scale. The average response was 1.85, with a standard deviation of 0.88 (0.875094, to be precise).

Given this low mean and standard deviation, most responses are likely to be toward the "dislike" end of the rating scale. What patterns of responses could yield this mean and standard deviation, and how similar would the possible patterns look?

There are only 4 possible datasets that can generate these statistics (see Figure 1 and Table 1). All solutions are extremely similar to one another. The highest possible value a data set can have is 5; these statistics become impossible if anyone were to really love vegetables as a child. The data originally generated corresponds to solution #1.

CORVIDS reveals that not only are there few solutions to these values, but also that the solutions are extremely similar to one another.

The case of the rounding error

Let us assume a different sort of problem. Imagine another 20-subject survey, this time with a mean response of 3.2 and

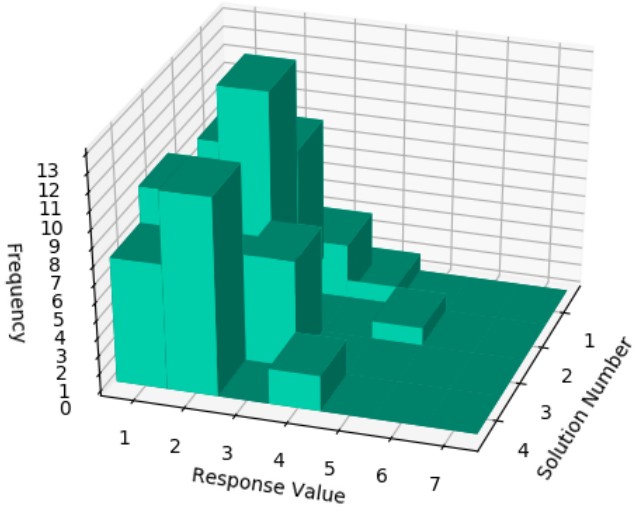

Figure 1: A 3D histogram of the solution space for the skewed data. Frequency of the scale value is on the y-axis, while the value itself is on the x-axis. The z-axis is each separate solution. In this case there are four solutions; due to the extreme skew of the data, none have a value above 5, and the solution that does contain a 5 has only 1s and 2s for its other values.

Sol #	1	2	3	4	5	6	7
1	8	8	3	1	0	0	0
2	6	13	0	0	1	0	0
3	9	5	6	0	0	0	0
4	7	11	0	2	0	0	0

Table 1: Number of respondents giving each value for the skewed data.

a standard deviation of 1.43637. However, let us say that the reported standard deviation was incorrectly rounded down to 1.43, instead of 1.44.

CORVIDS has no trouble dealing with this. Two decimal places is already insufficient precision, so we would have to set tolerance even if the statistic were correctly rounded to two decimal places. We recommend setting the tolerance up to the reported precision; here we would set the tolerance to $\pm$ .01. CORVIDS will then search for all possible valid standard deviations between 1.42 and 1.44, and will therefore find the correct value.

Running CORVIDS with this tolerance on the SD and no tolerance on the mean (since 3.2 was the exact value) results in one SD which can produce a solution, and indeed it is our original value: 1.43637. Despite the rounding error and the relatively generous envelope to search through, we were able not only to find solutions, but in this case find the original statistic as well (although it will not always be the case that only one value within the tolerance results in solutions; there may be several, and the higher the tolerance the more likely there are to be multiple sets of statistics for which solutions

exist).

Because most statistics will not be reported to the precision necessary for CORVIDS, tolerance around the mean and standard deviation was built into the program. It is thus trivial to account for rounding errors, such as this one; we just need to widen the envelope around the given statistic. If any values result in solutions, CORVIDS will find them.

The case of impossible values

CORVIDS will also return no solutions when none are possible. Take the skewed data. Let us increase the sample size to 25, adjust the mean to 2, but retain a standard deviation of 0.875094. It turns out that no combination of 25 values can produce that mean and standard deviation.

Similarly, if we attempt to reconstruct values with a sample size of 20, a mean of 2, a standard deviation of 0.875094, and a scale from 2 - 7, we again see that such values are impossible, even with a fairly generous SD tolerance of .01.

Solutions as a function of parameter values

In addition to straightforward data reconstruction, CORVIDS can be an exploratory tool.

Let us consider a 1-7 scale with 20 responses and a mean roughly in the center of the scale, and observe how the number of solutions changes as the standard deviation increases.

Mean	SD	# Sols
3.1	0.5525063	2
3.1	0.967906	16
3.1	1.372665	57
3.1	1.803505	97
3.1	2.48998	16
3.1	2.936163	1

Table 2: The number of possible solutions for different SD values.

From Table 2 we can see that the number of solutions has a U-shape when compared to the size of the standard deviation. At both very small and very large values for the standard deviation, the shape the data must take are constrained. For small values, all of the data points have to be clustered quite close to each other. For very large values, they have to be extremely far apart; the single solution for the largest value of the standard deviation in Table 2 consists solely of 1s and 7s. Between these two extremes, the number of possible solutions increases because the data have more patterns they can take. They could be relatively normally distributed, have values either close to the mean and or very far away, and so on.

By contrast, consider the same 1-7 scale with 20 responses, but this time hold the standard deviation fixed and vary the mean.

Table 3 shows that even for the same standard deviation, where the mean is located on the scale affects how many solutions can exist for that value. More solutions exist for means at the center of the scale, when data can vary on either side.

Mean	SD	# Sols
1.4	0.9947229	2
2.6	0.9947229	14
4.6	0.9947229	22
6.6	0.9947229	2

Table 3: The number of possible solutions for different mean values.

However, as the mean moves towards the endpoints of the scale, there is less ”room” for data points to vary, and fewer solutions exist.

Python Implementation

We have provided a Python implementation of the CORVIDS algorithm that provides the functionality described above, available on Github at https://github.com/katherinemwood/corvids/releases/tag/v1.0.0. The code can be downloaded from source, or a stand-alone executable can be downloaded and run.

The source code provides three primary functions. The recreateData function takes as arguments the sample mean, the sample variance, the sample size, and the maximum and minimum of the scale. If solutions exist, the function returns all possible solutions in a list. Optional arguments include precision arguments for the mean and variance; specifying these will cause the function to test every valid mean and/or variance within the range specified by the given value $\pm$ precision. These arguments are useful for circumventing rounding errors, or statistics that were reported to an insufficient precision. If no tolerance is given for the values, CORVIDS may report that no solutions exist when in fact valid solutions exist at values that are close to those reported.

This function can also check whether values are possible and whether solutions remain possible under certain value constraints. These are optional arguments that can be specified and passed. To check whether a given solution is contained the solution space, it may be passed as the checkVals argument.

The analyzeSkew function returns a list of the skewness of each solution, the mean skew, and the standard deviation of skew. This can be used as a rough measure of the heterogeneity of the solution space.

The graphData function creates an interactive 3D histogram of the solution space. Note that for large solution spaces, this can be an extremely slow and costly operation. For this reason, the function plots 40 solutions at random by default..

The code runs in multi-process mode by default to decrease runtime; this argument may be disabled.

If the stand-alone executable is run, then all interactions occur through a GUI. The solution functionality (getting all possible solutions, checking custom ranges, and forcing the inclusion of values) is available, as is the graphing.

See the documentation accompanying the code for more usage details.

Comparison to other reconstruction tools

Two extant methods exist to reconstruct or otherwise test the plausibility of Likert-style data; a Bayesian linear-inverse model (Morey,2016) and SPRITE (Heathers,2017). Both of these methods can be much faster than CORVIDS, especially for large scales and sample sizes or very generous tolerances around the mean and variance. SPRITE in particular is also more flexible than CORVIDS, capable of accepting more parameters and working with a greater diversity of data types.

However, each of these methods is approximate. They rely on a form of random sampling, and therefore cannot guarantee the completeness of their solution spaces in all cases. CORVIDS can make this guarantee; because it is closed-form and deterministic, whatever it returns will be the only, and every, possible solution to the given summary statistics and it will give the same results every time it is run. If it reports that no solutions are possible for a given set of values, that is a mathematical certainty.

Applications to non-Likert data

Applying CORVIDS to Likert-scale type data is straightforward. However, CORVIDS is not limited to this data type; it can deal with any type of data that falls on a restricted scale for which the responses are integers.

Consider a case in which the overall measure is average accuracy. 20 subjects take a short, 10-item test, and get a score out of 10; this is then converted to a percent, and the overall average is taken. It might be reported thus: "Subjects were relatively accurate overall, with an average of 69.5% and a standard deviation of 9.45%." With a bit of tweaking, we can change this into something CORVIDS can solve.

We know that the limits to our scale will be 0 to 10; a subject can either answer 0 items correctly, or they can get a perfect 10 out of 10 items correct. We convert the average percent into an average number of items; on average, subjects get 6.95 items correct (with a tolerance of .01), with a standard deviation of .945 items (and a tolerance of .001).

When these values are passed to CORVIDS, it will return 16 solutions. These solutions tell us how many subjects scored each possible value (how many got 10/10, how many got 9/10, etc). See Figure 2 for a plot of these solutions.

The following criteria have to be met for CORVIDS to be able to solve a given set of data:

1. **The response scale must be fixed**. There must be a minimum and maximum response value, either due to the nature of the scale (1 - 7 rating, 0 - 10 items correct) or because it is reported (e.g., "the minimum score in our sample was 4, while the maximum was 17").

2. **The values must be transformable to integers.** If the data have discrete steps and can be transformed to integers, they can often be solved. For example, even if a scale accepts fractional responses such as .25, as long as the minimum step size is fixed at .25, it can be transformed to an integer scale by multiplying it by 4. Even data that might appear to be continuous, such as weight or reaction time, might have a minimum step size due to the precision to which it can be measured (e.g. 1 gram, 10ms, .5 inches).

3. **The scale must be reasonably constrained.** Certain values, such as reaction time, in theory have a fixed scale (perhaps any time faster than 250ms or longer than 2000ms was discarded, giving a maximum and minimum) and a fixed step size (1ms). If precision on the mean and standard deviation is perfect, solving a scale of this size might take 30 minutes. The required time will increase if tolerance is added and admits more mean/variance pairs that have solutions.

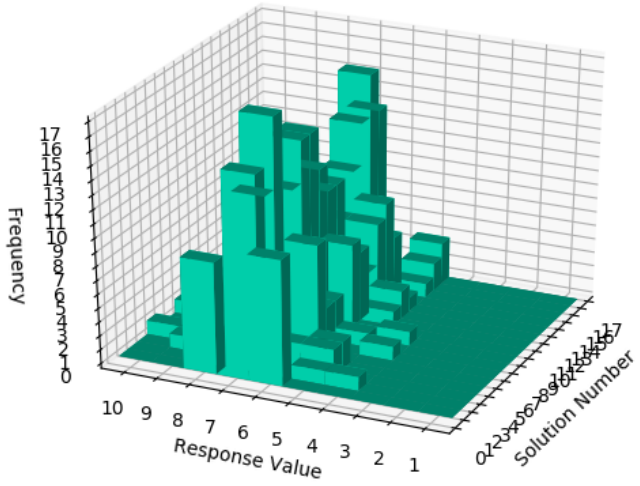

Figure 2: A 3D histogram of the solution space for the accuracy data. In this case, the response value is the number of items out of 10 that a subject answered correctly, and the frequency is how many subjects achieved that score.

Conclusion

The CORVIDS algorithm can fully reconstruct raw data from summary statistics alone. When data generated from a fixed scale, one needs only the limits and granularity of the scale (e.g. "integers 1 to 7"), the mean of the data, the standard deviation or variance, and the number of subjects collected in order to reconstruct every dataset which could have generated those summary statistics.

CORVIDS can be used in a variety of ways. Reconstruction is among its most useful; in cases where enough information is reported but the original data is unavailable, CORVIDS can return a set of solutions that is guaranteed to contain the original data. CORVIDS can also be used in an exploratory way. In some cases, the solutions will be quite constrained and similar to one another; for example, all of the solutions will be quite skewed. CORVIDS can be used to explore the solution space, and examine which possibilities remain when

the scale is restricted or certain response values are required to be present. In the case of our example with highly skewed data, there were absolutely no responses possible above a 5, for instance. CORVIDS can also be used to explore properties of distributions; one can generate a set of data according to certain values, and then explore how many other solutions exist and what their properties are.

References

Havas, G., Majewski, B. S., & Matthews, K. R. (1998). Extended gcd and hermite normal form algorithms via lattice basis reduction. *Experimental Mathematics*, *7*(2), 125–136.

Heathers, J. (2017). Introducing sprite (and the case of the carthorse child). Available from https://hackernoon.com/introducing-sprite-and-the-case-of-the-carthorse-child-58683c2bfeb

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, *22*(140), 1–55.

Matiyasevich, Y. V. (1972). Diophantine sets. *Russian Mathematical Surveys*, *27*(5), 124–164.

Morey, R. (2016). How to check likert scale summaries for plausibility. Available from http://bayesfactor.blogspot.co.uk/2016/03/how-to-check-likert-scale-summaries-for.html

R Core Team. (2017). R: A language and environment for statistical computing. Available from https://www.R-project.org/

Trautmann, H., Steuer, D., Mersmann, O., & Bornkamp, B. (2014). truncnorm: Truncated normal distribution. Available from https://CRAN.R-project.org/package=truncnorm (R package version 1.0-7)

Author Note

DS originally posed the question of algorithmically reconstructing data. SW developed the algorithm, proofs, and Python implementation. KW discovered the deterministic variance search, assisted in testing and debugging, and wrote the documentation. KW and SW drafted the manuscript. All authors critically edited the manuscript.

Appendix A

Formal Proof of completeness:

Proof. We begin by assuming a given initial solution vector (perhaps containing invalid negative values) called s_1 and a basis called $\mathbb{B}_1$ of a vector space of transformations such that the action of any element of the vector space on our initial solution produces an additional solution (which may also contain invalid negative values). Our goal is to detail a method to produce all possible solutions, $\mathbb{S}$, with exclusively positive integral values at every index.

1. Any basis vector of a vector space may be replaced by itself summed with a scalar multiple of any other basis vector, and the resulting new set of vectors is a valid basis of the original vector space.

2. Given a minimal basis ($\mathbb{B}_1$) of size m where each vector has dimensionality l, because of (1.), we can construct a new basis ($\mathbb{B}_2$) for the same vector space which has the property that of the rightmost (at least[‡]) m coordinates, exactly one has a value of 1 and the rest are all 0 where each 1 is at a different index (each vector restricted to these m unique indices under consideration is orthogonal). Doing so may result in non-integral values at some dimensions of some vectors, however, this does not cause any problems since any generated non-integral solutions can be discarded at the end.

3. Using $\mathbb{B}_2$, we can manipulate our starting solution s_1 (which may contain invalid negative values) such that the rightmost (at least) m coordinates are all zero to produce a new solution s_2 (also potentially containing negative values)

4. $\mathbb{B}_2$ and s_2 have several desirable properties:

   - Each vector in $\mathbb{B}_2$ has some dimension on which it uniquely acts
   - No fractional scalar multiples of the basis elements of $\mathbb{B}_2$ can ever preserve integer values (all valid actions by $\mathbb{B}_2$ require integer multiples of the basis elements) since they have a value of 1 on the dimension that they uniquely act upon.[§]
   - Any valid solution s_3 must have the form

     $$s_3 = s_2 + \sum_{b_i \in \mathbb{B}_2} a_i \cdot b_i$$

     where each $a_i > 0$ since s_2's rightmost m values are all 0 and each b_i acts uniquely and positively over exactly one of those rightmost values, if a basis vector is ever subtracted, it would introduce an invalid negative value which no other basis vector could compensate for.

5. Since the action of any basis vector in $\mathbb{B}_2$ on s_2 preserves its satisfaction of Equations 1 - 3, given that Equation 1 requires that the total sum of any produced s_3 be n, the sum of the coordinates of every vector in $\mathbb{B}_2$ must be 0.

6. Using (2.), we know that the sum of the last m coordinates is exactly 1. Combining this with (5.) we know that the first $l-m$ coordinates must sum to -1

7. Lastly, using the fact that s_2 satisfies Equation 1 and has values of 0 for its rightmost m coordinates, the first $l-m$

[‡]It can be the case that rather than m, the correct value may in fact be more than m when all basis vectors are zero on some dimension. However, this does not impact the proof or the algorithm in any way other than the superficial.

[§]That is, if the action of a basis vector introduces a non-integer value along the dimension on which it uniquely acts, no other basis vector can "fix" it to an integer value.

coordinates must sum to n. Thus for any valid solution s_3 as generated above,

$$\sum_i a_i \leq n$$

. Thus, since each a_i must be an integer, we can exhaustively enumerate and test all possible sets of a_is.[¶‖]

□

Appendix B

Proof. Given some initial

$$\sigma^2_{init} = \frac{\sum_{l=1}^{n} (x_l - \mu)^2}{n-1}$$

let

$$\sum_{l \neq i,j} (x_l - \mu)^2 = C$$

1.
$$\sigma^2_{init} = \frac{(x_i - \mu)^2 + (x_j - \mu)^2 + C}{n-1}$$

2. Generating a change in the data by increasing one value and decreasing another:

$$\sigma^2_{modified} = \frac{(x_i - 1 - \mu)^2 + (x_j + 1 - \mu)^2 + C}{n-1}$$

3. Taking the difference between these two yields the change in variance brought about by modifying the dataset:

$$\Delta\sigma^2 = \frac{(x_j + 1 - \mu)^2 + (x_i - 1 - \mu)^2 + C - (x_i - \mu)^2 - (x_j - \mu)^2 - C}{n-1}$$

$$\longrightarrow$$

$$\Delta\sigma^2 = \frac{(x_i - \mu)^2 + (x_j - \mu)^2 - (x_i - \mu)^2 - (x_j - \mu)^2 - 2(x_i - \mu) + 1 + 2(x_j - \mu) + 1}{n-1}$$

Expanding the terms and simplifying yields:

4.
$$\Delta\sigma^2 = \frac{2(x_i - x_j) + 2}{n-1} = \frac{2}{n-1} \cdot [(x_i - x_j) + 1]$$

meaning that the resulting variance change from moving two values always has to be a multiple of 2/(n-1).

□

[¶]In practice, we need not enumerate all combinations up to n, and this search can be considerably optimized by search space pruning; however, it is still computationally expensive for many datasets.

[‖]The theoretical time complexity for this is $\binom{n+m-1}{n}$ which grows quite quickly; however, in practice search space pruning drastically improves performance

Appendix C

Proof. Given $k = m \cdot (n) \mod n$, [**] where n is the sample size and m is the reported sample mean, we will construct the dataset with as many subjects as close to the mean as possible.

$$k + n \cdot \lfloor m \rfloor = m \cdot n$$

$$k + \sum_1^{n-k} \lfloor m \rfloor + \sum_1^{k} \lfloor m \rfloor = m \cdot n$$

Moving the k into the second summation yields

$$\sum_1^{n-k} \lfloor m \rfloor + \sum_1^{k} (\lfloor m \rfloor + 1) = m \cdot n$$

$$\text{since } \lfloor m \rfloor + 1 = \lceil m \rceil \longrightarrow$$

$$\sum_1^{n-k} \lfloor m \rfloor + \sum_1^{k} \lceil m \rceil = m \cdot n$$

And thus we know that a dataset with $n-k$ values of $\lfloor m \rfloor$ and k values of $\lceil m \rceil$ gives us the desired mean of m. Computing the variance of this valid dataset, we obtain:

$$v = \frac{(n-k) \cdot (m - \lfloor m \rfloor)^2 + k \cdot (\lceil m \rceil - m)^2}{n-1}$$

□

[**]If m is an integer, $k = 0$ and this exercise becomes trivial so we will assume m is not an integer