ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/312085497

Dealing with missing data. Online Supplement 2 to Serious stats: A guide to advanced statistics for the behavioral sciences...

Chapter · January 2012

CITATIONS
0

READS
238

1 author:

Thom S Baguley
Nottingham Trent University

64 PUBLICATIONS **888** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project WORKAGE project on workplace policies that can support engagement and delayed retirement (EU-funded, 2013-2016) View project

Project Meta-analysis of gender differences in the Stroop Colour-Word test View project

All content following this page was uploaded by Thom S Baguley on 05 January 2017.

The user has requested enhancement of the downloaded file.

Online Supplement 2

Dealing with missing data

This supplement draws primarily on Chapters 5 and 10.

OS2.1 Dealing with missing data

Missing data is a major problem in many fields of research. Frequent causes of missing data are the nature of the variables being studied (e.g., losses due to attrition or dropout in a long study) or the impact of chance events (e.g., data loss due to equipment failure). It is also possible, but less common, to deliberately design a study so that some data are missing.

Designing a study to have missing data on some measures seems like an odd thing to do, but can be a very sensible strategy (e.g., when measuring many variables, some of which are expensive to collect). Graham (2009) reviews a number of methods for the analysis of missing data and considers some of the advantages and disadvantages of this kind of planned 'missingness'.

More often than not, missing data occur as a consequence of processes beyond a researcher's control. Understanding the nature of these processes is central to dealing with missing data. All modern methods for dealing with missing data build on an appreciation of these processes. A crucial insight is that just because data are missing does not mean that all the information associated with them is lost. This can be a difficult idea to grasp at first.

Imagine that two treatments for weight loss (A and B) are being compared in an independent groups design. There are 20 participants per group, each measured at weekly intervals over a period of six months. At the end of the study, mean weight loss in each of the two groups is similar, but group A has complete data for 18 participants and group B for only six. The dropout rate for the study is likely to be related to the effectiveness of the treatment. Participants who don't lose much weight are probably more likely to drop out. If you could factor this information back into the analysis, it might provide evidence of an advantage for treatment A over treatment B. Treatment A may be more effective than treatment B, but the initial estimate of the group B mean is biased upwards (because information about the several participants who weren't losing much weight is not included). There is information in missing data. The challenge is to find rigorous methods for recovering it and incorporating it into a statistical model.

Modern methods for dealing with missingness draw on classic work by Little and Rubin (1987). They classified the processes that cause missing data in terms of three 'mechanisms': *missing completely at random* (*MCAR*), *missing at random* (*MAR*) and *missing not at random* (*MNAR*). These mechanisms group the causes of missing data by their statistical properties. This makes it simpler to match up messy, incomplete data sets with potential solutions.

Schafer and Graham (2002) provide an excellent overview of the distinctions between these three mechanisms (and some of the potential confusion in distinguishing between them). *MCAR* is the simplest to explain. If data are missing completely at random, then the cause of missing data is entirely unrelated to any variable relevant to the analysis. Cases with incomplete data can therefore be considered a random sample of the population of possible cases. This makes *MCAR* the easiest mechanism to deal with, because its only consequence is to reduce the quantity of data available for analysis. This is undesirable (e.g., decreasing statistical power), but does not bias the analysis.

MCAR is widely considered implausible as a mechanism for missingness (Schafer and Graham, 2002). It would be unrealistic to expect dropout in a longitudinal study to be unrelated to potential predictors such as the health, socioeconomic status or personality of participants. In other studies, missing data may arise because people are unwilling to answer certain questions – and again you'd expect gaps in the data to be related to some of the measures you are interested in (e.g., political or religious views). Although the general consensus is that *MCAR* is rare (except by design), there are restricted circumstances where it may occur. Missing data may arise in experimental research through equipment failure, programming errors or other potentially random events (e.g., interruptions in power supply). It is probably safe to assume *MCAR* for data loss caused by such random events, provided one further criterion is met. This is that the proportion of cases with missing data should be small (e.g., perhaps 5% or less). This safeguard is important because if you are wrong (e.g., the equipment failure occurred only for participants that behaved in a certain way) the degree of bias will be minimal. Otherwise it is safest to assume that data are *MAR* or *MNAR*. It is possible to conduct a partial check of the *MCAR* assumption by looking at differences between cases with and without missing data (though in complex data sets you may miss subtle patterns with this kind of quick check).

MAR is the most misunderstood of the three mechanisms. The term 'random' in *MAR* is used in a very restricted sense (Graham, 2009). Data are missing at random, conditional on having incorporated information from all the available data into the analysis. *MAR* therefore assumes that missing data arise through processes that are predictable based on what has been measured (i.e., the observed data). If dropout in the weight loss example depends on how much weight each participant is losing, then the mechanism is *MAR*. This is because weight loss was measured at weekly intervals up until the point where the participant dropped out. MAR is a highly plausible mechanism in many studies. This has important implications. Not only are the missing data informative, but the incomplete cases can be used to extract that information. This will lead to a more accurate and efficient statistical model.

Ignoring the presence of MAR data introduces bias – possibly substantial bias – into the analysis. This can be illustrated with the weight loss example. If dropout is related to weight loss, group A will have fewer missing observations and produce a fairly accurate estimate of the effectiveness of treatment A. In contrast, the remaining participants in group B will be those who lost most weight. Their data will overestimate the effectiveness of treatment B. The difference between groups (looking at only those who completed the study) will be biased in favor of treatment B. It is therefore important to measure variables that are likely to predict dropout (e.g., data about treatment progress and demographic data such as age or sex). Even if dropout is not directly related to the outcome of interest, it may be essential to an unbiased analysis.

The final mechanism is *MNAR*. It can be defined as dropout that is not completely at random and not predictable from the observed data. This would, for the weight loss example, correspond to a situation in which an unobserved, unmeasured variable (e.g., personality type) predicts both missingness and weight loss. *MNAR* is the hardest mechanism to deal with (see Graham, 2009).

A practical solution for dealing with MNAR may be to follow up a random sample of missing cases in order to discover how the missing data has biased the statistical analysis. The challenge of working with MNAR data provides a further incentive to collect information about factors that may cause data to go missing (Collins *et al*., 2001). For instance, in longitudinal studies Graham (2009) suggests collecting data about participants' intentions to drop out. If not all those who intend to drop out leave the study, the information from those who remain can be used to reduce bias in the analysis.

Real data doesn't always align neatly with Little and Rubin's three mechanisms (e.g., being a mixture of all three). Fortunately, evidence suggests that the best methods for dealing with *MAR* cope well when data are *MCAR* (Schafer and Graham, 2002; Graham, 2009). The same research suggests that *MAR* solutions may reduce bias for *MNAR* data sets. This seems plausible if *MAR* methods incorporate variables that, while not causing missingness, are correlated with unmeasured variables that do cause missingness (see Collins *et al*., 2001). The following sections focus on methods for dealing with data that are *MAR*. They start with a brief survey of methods known to be inadequate for dealing with *MAR*, before introducing one of the best available methods for dealing with missing data: *multiple imputation* (MI).

OS2.1.1 Deletion methods

The most widely adopted strategy for dealing with missing data is to drop or remove cases with missing values from the analysis. This strategy could be explicit, but is often adopted implicitly because of the way that software typically handles missing data (see Box OS2.1). The most popular method is *listwise* or *casewise* deletion. This involves dropping cases from an analysis if any observation is missing for that case. Casewise deletion has two major deficiencies. The most obvious is the loss of statistical power or precision arising from reduced sample size. A less well-appreciated deficiency is that casewise deletion biases parameter estimates and inference (unless data are *MCAR*).

Even if the *MCAR* assumption is plausible, casewise deletion should be avoided because it is inefficient. Even small proportions of missing data in a data set can be very damaging to statistical power if there are many cases with missing data. Because the variables with missing data typically differ between cases, a small proportion of missing values can lead to many (perhaps most) cases being dropped. If 5% of observations are missing at random from ten variables, around 40% of cases will have some missing data.[1]

Pairwise deletion involves using as many cases as possible in a given analysis. This could mean retaining as much data as possible for each of several analyses (e.g., producing a correlation matrix). However, it can also refer to the practice of computing values (e.g., variances and covariances) that are then combined for a more complex analysis (Peugh and Enders, 2004). Like casewise deletion, pairwise deletion also assumes data are *MCAR* and can introduce additional problems in complex analyses (see Graham, 2009). Casewise or pairwise deletion are acceptable choices only if the total number of cases with missing data is small and the *MCAR* assumption is plausible.

Box OS2.1 Coding missing data

There are no universal rules for coding missing data. The main conventions are to code missing data with an unusual or impossible numeric code such as −99 or with a text string such as 'NA'. Some software will also accept blank slots or entries – but these are occasionally treated as zeroes (e.g., by some Excel functions).

If you do use a numeric code it is important to select one that will be recognized by whatever software is being used as 'missing' rather than as an error or a zero. This can often be defined by the user (e.g., in SPSS), and will lead to problems if not done consistently. Using −99 for a missing value code will work for some analyses, but it could be a legitimate value in others. A particularly bad choice is zero, partly because it is often a legitimate value and partly because it may be hard to detect an incorrect code. It is very easy to make a mistake (e.g., for missing values to be unintentionally included in an analysis). For this reason avoiding numeric codes is preferable if at all possible.

Instead, use a text string such as 'NA' (where software allows). 'NA' is the built-in missing value indicator in R (which has a range of functions and options for handling missing data). A missing value code is important because it acts as a 'place holder' in the data structure. Paired data provide a simple illustration of the problem. If one observation from a pair is missing, omitting the observation entirely would mess up the remaining pairings (by shifting the sequence of data in one sample but not the other). On getting unusual or unexpected results from an analysis, missing value codes (along with data entry errors) should be among the first things to check.

OS2.1.2 Mean substitution and related regression methods

Rather than deleting cases with missing values an attractive alternative is to substitute each one with an estimate of its true value. In *mean substitution* (also called *mean imputation*) a missing observation is estimated from the mean of the available observations. There are several variants of mean substitution, depending on the design of the study. In a repeated measures study you could either use the mean of all repeated measurements (the grand mean) or the mean of the measurements in that sample (the sample mean). In an independent measures study you could use the mean of the group in question, the grand mean of all groups or the grand mean weighted by group *n*.

Mean substitution is not recommended, regardless of whether data are *MCAR*, *MAR* or *MNAR*. Even if the mean was an unbiased estimate (which it rarely is), the procedure inflates sample size. Missing data are less informative than non-missing data (the value that should have been observed is not known with certainty). Adding plausible values into the sample ignores this property, and leads to standard errors that are too small (being computed from a spuriously large sample size). As the variances and covariances are also underestimated, it will also bias estimates of these parameters (as well as standardized effect size metrics such as *d* or *r*).

A more sophisticated alternative is to use all the available data to predict missing values (e.g., via regression). The predicted values could then be substituted for the missing data. This method is known as *conditional mean substitution* or *regression imputation*. For simple study designs this is identical to mean substitution. Regression imputation can be extended to take into account other variables that predict missingness, and produces unbiased parameter estimates for intercepts and slopes (and hence means) if data are *MAR* or *MCAR*. However, because the imputed data fall exactly on the regression line they are less variable than true values would have been. Again, this leads to biased estimates of variances and covariances (and all statistics and inferences derived from them). An ingenious solution is to add random error to each imputed observation. This is known as *single imputation* (see Schafer and Graham, 2002). If the proportion of missing values for each variable is small, single imputation performs reasonably well for *MAR* or *MCAR* data (with accurate parameter estimates and relatively low bias for variances, covariances or inferences).

Schafer and Graham (2002) also discuss other practices for handling missing data. Many of these merely disguise the problem rather than solve it. A common practice is to add dummy variables to code 'missingness'. This allows the statistical analysis to run with all cases, but because it separates out estimates of missing and non-missing data the analyses do not incorporate any potential information from the missing cases.[2] Jones (1996) explains how these practices can lead to highly misleading results. The dummy variable approach produces coefficients equivalent to casewise deletion, but also spuriously inflates the sample size.

OS2.1.3 Multiple imputation

Two methods that improve on single imputation are *maximum likelihood* and *multiple imputation* (*MI*). Maximum likelihood methods involve fitting a regression model to find the most likely parameter estimates for a particular set of data. As they rely on iteratively fitting a model, they can cope with data structures that present problems for standard least squares estimation. These approaches provide an elegant solution to some missing data problems – but may not be practical for many everyday situations. Maximum likelihood methods have applications other than for missing data (some of which are considered in Chapter 17 and Chapter 18). The following discussion focuses on *MI*. The two approaches are, in principle, equally effective when missing data are *MAR* or *MCAR*. An advantage of *MI* is that it is somewhat easier to implement for standard regression models.

MI is a logical extension of single imputation. Single imputation produces a data set that attempts to account for uncertainty in the prediction of imputed values. The weakness of single imputation is that it underestimates the uncertainty introduced by imputation. Single imputation generates one imputed data set containing observed data and imputed values (predictions from a regression model to which random noise has been added). What happens if a second imputed data set is generated for the same observed data? The two imputed data sets should have similar parameter estimates, but because each one has added random noise to the imputed values, they won't be identical. The presence of differences between the data sets implies that there is a source of uncertainty that single imputation ignores. Multiple imputation attempts to take into account the additional uncertainty implied by this between-imputation variability.

The imputed values from single imputation can be regarded as a sample from a population distribution summarizing all possible imputed values. From this perspective, single imputation ignores the error inherent in sampling from this population. A potential analogy here is with a random effects meta-analysis in which random error is incorporated from two sources (within and between studies). Multiple imputation has to take account of sampling error within each imputed data set and between imputed data sets. Just as you can estimate overall parameter estimates in random effects meta-analysis by pooling estimates across several studies, you can do the same in *MI* by pooling estimates across several imputed data sets. This leads to more precise, less biased parameter estimates and inferences. The larger the number of imputed data sets being pooled, the more accurate the parameter estimates and inferences.

A drawback of *MI* is the additional effort required to impute multiple data sets, apply the same statistical model to each data set and then pool the resulting statistics. Rubin (1987) provided a relatively simple set of equations for pooling coefficients and standard errors from regression models. If the number of imputed data sets is m, and b_j is a coefficient (intercept or slope) from the j^{th} imputed data set, then the pooled estimate $\bar{b}$ ('b-bar') of that coefficient is the arithmetic mean of the coefficient from each imputed data set:

$$\bar{b} = \frac{\sum_{j=1}^{m} b_j}{m} \qquad \text{Equation OS2.1}$$

Within-imputation variance can be combined in the same way. The within-imputation variance for the j^{th} imputed data set is $\hat{\sigma}^2_{b_j}$ (the sampling variance or squared *SE* of the coefficient). The pooled within-imputation variance is thus the arithmetic mean of m sampling variances:

$$\hat{\sigma}^2_{\bar{b}} = \frac{\sum_{j=1}^{m} \hat{\sigma}^2_{b_j}}{m} \qquad \text{Equation OS2.2}$$

To obtain *SE*s for the pooled coefficients requires both the within-imputation and between-imputation variance. The between-imputation variance is calculated by applying the usual inferential sample variance formula (treating the coefficients as observations):

$$\hat{\tau}^2_{\bar{b}} = \frac{\sum_{j=1}^{m} (b_j - \bar{b})^2}{m-1} = \frac{\sum_{j=1}^{m} \left(b_j^2\right) - m\bar{b}^2}{m-1} \qquad \text{Equation OS2.3}$$

Combining the two sources of variation is the next step. It applies a formula analogous to that in random effects meta-analysis:

$$\hat{\sigma}^2_{MI} = \hat{\sigma}^2_{\bar{b}} + \left(1 + \frac{1}{m}\right)\hat{\tau}^2_{\bar{b}} \qquad \text{Equation OS2.4}$$

The square root of this sampling variance ($\hat{\sigma}_{MI}$) is the *SE* of the multiply imputed coefficient. A test statistic for a test of the null hypothesis H_0: $\bar{b} = 0$ is therefore:

$$\frac{\bar{b}}{\hat{\sigma}_{MI}} \sim t(\nu) \qquad \text{Equation OS2.5}$$

This statistic has a t distribution with degrees of freedom equal to

$$\nu = (m-1)\left(1 + \frac{m\hat{\sigma}^2_{\bar{b}}}{(m+1)\hat{\tau}^2_{\bar{b}}}\right)^2 \qquad \text{Equation OS2.6}$$

and a CI for the imputed coefficient can be constructed as:

$$\bar{b} \pm t_{\nu,1-\alpha/2}\hat{\sigma}_{MI} \qquad \text{Equation OS2.7}$$

OS2.1.4 Setting up multiple imputation

Rubin's equations are an effective, but cumbersome, way to combine parameter estimates for multiply imputed data sets. Even using specialist software a number of important decisions have to be made. These include what variables to include in the imputation model, whether to transform variables for imputation and how many imputations to use (see Graham, 2009).

MI works best when including all variables in the imputation model – regardless of whether they are part of the subsequent analysis and regardless of whether they are predictors or outcomes. Sometimes it is necessary to exclude variables because they are perfectly collinear (i.e., one is correlated perfectly with another variable or a combination of other variables) or because there are too many. In both cases the problem is simply a practical one of running the imputation analysis. Where the analysis uses a derived variable (one computed from other variables in the model) this should be included in the imputation model unless it is collinear with other predictors. For instance, a psychometric scale derived as the sum of several subscales will be collinear with the subscales (and perhaps only the subscales should be included in the imputation model). On the other hand, if the intention is to use a square root transformation of one of the predictors, this will not be perfectly collinear with the untransformed variable and both can be included.

Variables that are bounded in some way usually need to be transformed prior to imputation (though the transformation can be reversed for the analysis). Thus a logarithmic transformation for a variable constrained to be greater than zero will guarantee that the imputed values will also be greater than zero when the transformation is reversed (see Su *et al*., 2010). Categorical variables are trickier to deal with. Treating them as continuous will generally produce reasonable results, although alternative approaches are available (Graham, 2009; Su *et al.,* 2010).

Recommendations for the number of imputations to be used vary greatly. Rubin (1987) has shown that the efficiency of *MI* methods depends on the rate of missing information (M_r) and the number of imputations m. Efficiency (relative to complete data) is high when the amount of missing information is low and the number of imputations is large. Even small m will be sufficient to produce efficient estimates in most cases. Early *MI* work suggested that $m = 3$ or $m = 5$ was sufficient, whereas later work recommended values of $m = 10$ or $m = 20$ (e.g., see Schafer and Graham, 2002). Recent findings reviewed by Graham (2009) indicate that the impact on statistical power of the number of imputations should also be considered. Graham suggests that with 50% missing information $m = 40$ should ensure a negligible loss of statistical power (e.g., 1% or less). For this reason, unless missing values are rare, it is sensible to set m to a high value (e.g., 30 to 60), where sufficient computing resources are available.

Imputation of missing data is a rapidly developing field. It has the power to dramatically reduce bias and increase statistical power relative to crude approaches such as casewise deletion. *MI* also requires additional assumptions – including distributional assumptions (e.g., that the imputed residuals are sampled from an independent, normal distribution with constant variance), but you will generally be better off using *MI* than not. One reason for this is that any violations of the assumptions are diluted because only a small proportion of the overall data is likely to be imputed (Schafer and Graham, 2002). If the proportion of missing data is very large (e.g., over 50%) imputation may still be the best approach, but no method is likely to be very accurate in such cases.

Example OS2.1 This example compares an analysis of complete data with an analysis of incomplete data. We'll use a small simulated data set with $n = 100$. The data are samples from normally distributed variables with a mean of zero and an *SD* of one. These are a predictor (x), an outcome (y) and a third 'auxiliary' variable (*aux*). The auxiliary variable is correlated with x in the population ($\rho = -.50$), but not correlated with the outcome y. The population correlation between x and y is set at .70. The linear regression of y on x for the complete data set produces the prediction equation:

$$\hat{y} = 0.01153 + 0.75121x$$

We created an incomplete data set by replacing the observed Y with a missing value code for any case where *aux* was greater than zero. This mimics the kind of dropout that might occur if participants were recruited to a study for an initial measurement, with an outcome measured at a later date. In this instance 49 out of a possible 100 outcomes were replaced with missing values. The *mi* package (Su *et al.*, 2010) was then used to impute three data sets in R. All three variables (x, y and *aux*) were included in the prediction model. As missingness is determined entirely by *aux*, this meets the conditions of the *MAR* mechanism and so *MI* should produce a more accurate model than the incomplete data set.

The coefficients and standard errors of the linear regression of y on x for the complete data set, incomplete data and for each of these three imputed data sets (labeled $j = 1$ to 3) are shown in Table OS.1. The estimates from the incomplete data are badly biased (relative to the complete data set). This is most obvious for the intercept – but also true for the slope.

Table OS2.1 Coefficients and standard errors for the linear regression of y on x, for complete data, incomplete data and three imputed data sets in Example OS2.1

	Intercept		**Slope**	
Data set	**Coefficient**	**SE**	**Coefficient**	**SE**
Complete	−0.0115	0.0731	0.7512	0.0774
Incomplete	0.3975	0.1059	0.8734	0.1135
$j = 1$	0.0074	0.0765	0.7656	0.0811
$j = 2$	−0.0222	0.0778	0.6895	0.0826
$j = 3$	−0.1091	0.0864	0.6749	0.0916

In this case the imputed data sets produce coefficients and *SE*s that are closer to those for complete data than for the incomplete data. There is a lot of variability between imputed data sets (which is to be expected given that over 50% of y values are missing).

Multiple imputation involves pooling these coefficients and *SE*s. Combining the intercept and slopes is the easiest step and is done by taking the arithmetic mean of the imputed coefficients. The pooled intercept $\bar{b}_0$ is therefore $(0.0074+-0.0222+-0.1091)/3=-0.0413$ and the pooled slope $\bar{b}_1$ is $(0.7656+0.689+0.6749)/3=0.710$. Even with $m=3$ and around half the data missing, the pooled *MI* estimate of the regression line is close to that of the complete data set. *MI* won't always produce estimates this good, but can be expected to produce coefficients more similar to the complete data set than the incomplete data set.

To obtain the within-imputation variance, the sampling variances are averaged:

$$\hat{\sigma}^2_{\bar{b}_0}=\frac{\sum_{j=1}^{m}\hat{\sigma}^2_{b_{0,j}}}{m}=\frac{(0.0765)^2+(0.0778)^2+(0.0864)^2}{3}\approx 0.006456683$$

Because this result feeds into a later equation it is a good idea to retain as many decimal places as possible in the result. The within-imputation variance for the slope is:

$$\hat{\sigma}^2_{\bar{b}_1}=\frac{\sum_{j=1}^{m}\hat{\sigma}^2_{b_{1,j}}}{m}=\frac{(.0811)^2+(0.0826)^2+(0.0916)^2}{3}\approx 0.00726351$$

The between-imputation variance depends on deviations of the imputed intercept and slope from the pooled averages $\bar{b}_0$ and $\bar{b}_1$ respectively. For the intercept $\hat{\tau}^2_{\bar{b}_0}$ will be a variance computed treating the m estimates as observations:

$$\hat{\tau}^2_{\bar{b}_0}=\frac{\sum_{j=1}^{m}(b_{0,j}-\bar{b}_0)^2}{m-1}=\frac{\sum_{j=1}^{m}\left(b^2_{0,j}\right)-m\bar{b}^2_0}{m-1}$$

$$=\frac{0.0074^2+(-0.0222)^2+(-0.1091)^2-3\times(-0.0413)^2}{2}\approx 0.00366667$$

The corresponding calculation for the slope is:

$$\hat{\tau}^2_{\bar{b}_1}=\frac{\sum_{j=1}^{m}\left(b^2_{1,j}\right)-m\bar{b}^2_1}{m-1}=\frac{0.7656^2+0.6895^2+0.6749^2-3\times 0.71^2}{2}\approx 0.00237181$$

The combined variances are therefore

$$\hat{\sigma}^2_{MI,b_0}=\hat{\sigma}^2_{\bar{b}_0}+\left(1+\frac{1}{m}\right)\hat{\tau}^2_{\bar{b}_0}=0.006456683+(1+1/3)\times 0.00366667\approx 0.01134558$$

and

$$\hat{\sigma}^2_{MI,b_1}=\hat{\sigma}^2_{\bar{b}_1}+\left(1+\frac{1}{m}\right)\hat{\tau}^2_{\bar{b}_1}=0.00726351+(1+1/3)\times 0.00237181\approx 0.01042592$$

Taking the square roots gives $\hat{\sigma}_{MI,b_0}=0.1065156$ and $\hat{\sigma}_{MI,b_1}=0.1021074$. Both standard errors are larger than observed for the complete data. This makes sense (as inferences in the presence of missing data should be more uncertain than for complete data). The *df* for the imputed error terms are

$$\nu_{b_0} = (m-1)\left(1+\frac{m\hat{\sigma}^2_{\bar{b}_0}}{(m+1)\hat{\tau}^2_{\bar{b}_0}}\right)^2 = 2\left(1+\frac{3\times 0.006456683}{4\times 0.00366667}\right)^2 \approx 10.77$$

and

$$\nu_{b_1} = (m-1)\left(1+\frac{m\hat{\sigma}^2_{\bar{b}_1}}{(m+1)\hat{\tau}^2_{\bar{b}_1}}\right)^2 = 2\left(1+\frac{3\times 0.00726351}{4\times 0.00237181}\right)^2 \approx 21.74$$

A test of the null hypothesis that the population intercept is zero can be reported as:

$$t(10.77) = -0.39, SE = 0.107, p = .706, 95\%\ \text{CI}\ -0.041[-0.276, 0.194]$$

This leads to the same conclusion as that for the complete data set, but differs from that for the incomplete data set. For the slope, the test and CI are:

$$t(21.74) = 6.95, SE = 0.102, p < .001, 95\%\ \text{CI} = 0.710\ [0.498, 0.922]$$

This is a similar result to that of the complete data set and more accurate than the estimate for the incomplete data set.

Using only three imputations is a good way to illustrate the calculations, but for a real application *m* should be larger. Given that around 50% of the data are missing, $m = 40$ might be a reasonable choice to get a precise measure of the between-imputation variance and maintain statistical power. Running the same analysis twice with $m = 40$ produces similar pooled coefficients and standard errors (identical to two decimal places), suggesting that this is sufficient. The pooled intercept for the regression was around -0.07 with $SE = 0.13$. The pooled slope was around 0.68 with an $SE = 12$.

The `mi` package is one of several ways to do *MI* in R. It is one of the easiest to use and has a number of useful features. It automatically recognizes and pre-processes some data types (e.g., dichotomous variables and variables constrained to be > 0). It can also be used in an interactive mode that takes you through the imputation process step-by-step.

OS2.2 R code for Online Supplement 2

OS2.2.1 Missing data and multiple imputation (Example OS2.1)

Several R packages can impute missing data. Imputation is demonstrated here using `mi` (Su et al., 2010). This also has an interactive version (not described here). Missing values should, as a rule, be coded as `NA` within R. When loading data, or when missing values arise in other contexts, R will usually code them `NA` automatically. Functions such as `is.na()` exist to work with missing data and many functions have options for handling `NA` codes. Most regression functions have an argument `na.action` that sets how the function behaves on encountering `NA` codes. This is usually set to `na.action = na.omit` or `na.action = na.fail` (for further details see ?NA and ?na.fail). The former drops missing cases (equivalent to casewise deletion), while the latter causes the call to fail and return a warning (a safe option for complex regression functions). To check your present settings:

```
getOption('na.action')

?NA
?na.fail
```

For Example OS2.1, the first step is to import the incomplete data file from the SPSS file. Then load mi and use the mi() function to impute $m=3$ data sets.

```
library(foreign)
partial.data <- read.spss('mi_partial.sav', to.data.frame=TRUE)

install.packages('mi')
library(mi)
n.imputed <- 3
imputed.dat <- mi(partial.data, n.imp = n.imputed, n.iter =
  30, check.coef.convergence=TRUE)
```

The imputed.dat object holds all three data sets in a format that supports analysis via regression functions such as lm() or glm(). Increase the number of iterations to n.iter=40 or n.iter=50 if there are problems with convergence. See ?mi for further details.

Use mi.data.frame() to extract a data frame from the imputation model object. The following commands extract the first data set and run a simple linear regression and then obtain the summary of the output:

```
imputed.data.1 <- mi.data.frame(imputed.dat, m=1)
lm(imputed.data.1$y ~ imputed.data.1$x)
summary(lm(imputed.data.1$y ~ imputed.data.1$x))
```

You can also use mi to analyze all data sets and combine the regression coefficients for you. Here lm.mi() runs a linear regression model on the imputed data sets in imputed.data. The functions display(), coef() and se.coef() are used to get summaries, coefficients and *SE*s from mi objects (Gelman and Hill, 2007; Su *et al.*, 2010).

```
imputed.model <- lm.mi(y.do ~ x, imputed.dat)
display(imputed.model)
coef(imputed.model)
se.coef(imputed.model)
```

Compare this with the analysis of incomplete cases:

```
inc.mod <- lm(y.do ~ x, data=partial.data, na.action=na.omit)
display(inc.mod)
```

Although the `mi` package combines estimates of imputed regression coefficients and *SE*s it doesn't provide CIs or tests (largely because the authors of the `mi` package focus on statistical modeling rather than hypothesis testing). However, Rubin's calculations can be implemented through basic arithmetic operators or via the `mitools` package. Using `mitools` has the advantage of automating the awkward *df* calculation for the CIs. This sequence of commands extracts the data frames from the `mi` output and creates an imputation list object that `mitools` will recognize.

```
install.packages('mitools')
library(mitools)

dataset.list <- list(mi.data.frame(imputed.dat, m=1),
  mi.data.frame(imputed.dat, m=2), mi.data.frame(imputed.dat, m=3))

mitools.list <- imputationList(dataset.list)
```

If *m* is large this becomes tedious, so the steps can be automated by writing a function such as:

```
mi.to.mitools <- function(imputed.data.from.mi, m =
  imputed.dat@m) {
    # mi and mitools must be installed and loaded
    data.list <- as.list(1:m)
    for (i in 1:m) data.list[[i]] <-
         mi.data.frame(imputed.data.from.mi, m = i)
    mitools.list <- imputationList(data.list)
    mitools.list
}

mi.to.mitools(imputed.dat)
```

These commands run the regression model on the data frames and combine them using functions found in `mitools`. Unlike `mi`, a 95% CI for each parameter is included in the summary and the `df` can be obtained if required:

```
mi.mods <- with(mitools.list, lm(y.do~x))
summary(MIcombine(mi.mods))
MIcombine(mi.mods)$df
```

The output of the imputations and pooled results will vary every time you impute new data. Unless you pool large numbers of imputed data sets your results will not necessarily be similar to those reported in Example OS2.1.

OS2.2.2 R packages

Gelman, A., Hill, J., Yajima, M., Su, Y.-S., and Pittau, M. G. (2011) *mi*: Missing data imputation and model checking. R package version 0.09-14.

R-core members, DebRoy, S., Bivand, R., I. (2011) *foreign*: Read data stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase. R package version 0.8-42.

Lumley, T. (2010) *mitools*: Tools for multiple imputation of missing data. R package version 2.0.1.

OS2.3 Notes on SPSS syntax for Online Supplement 2

OS2.3.1 Multiple imputation

Recent versions of SPSS (e.g., from 17 onwards) provide multiple imputation facilities that are fairly easy to run. These support pooling of results using Rubin's equations for several types of analysis.

The following syntax imputes missing values on the outcome variable (`y.do`) for 40 data sets for the `mi_partial.sav` data set used in Example OS2.1 using the predictor `x` and auxiliary variable `aux`. The data are delivered to a new SPSS data window (`partial_m40`), but not saved to disk, by using `DATASET DECLARE` and adding an `/OUTFILE` subcommand.

SPSS data file: `mi_partial.sav`

```
DATASET DECLARE partial_m40.

MULTIPLE IMPUTATION x y.do aux
  /IMPUTE METHOD=FCS MAXITER= 30 NIMPUTATIONS=40
SCALEMODEL=LINEAR INTERACTIONS=NONE
    SINGULAR=1E-012 MAXPCTMISSING=NONE
  /CONSTRAINTS x(ROLE=IND)
  /CONSTRAINTS aux(ROLE=IND)
  /IMPUTATIONSUMMARIES MODELS DESCRIPTIVES
  /OUTFILE IMPUTATIONS=partial_m40.
```

The `/IMPUTE` subcommand includes several statements relating, among other things, to the method of imputation, the maximum number of iterations the imputation process will go through for each data set (30 in this case) and the number of imputations. The `FCS` (fully conditional specification) method used here is broadly similar to that used in the R package *mi*. The easiest way to obtain the syntax is to set up the analysis via the menus in SPSS and select paste. For a simple imputation such as this, the main concerns are the number of iterations and the number of imputations. The `/IMPUTATIONSUMMARIES` subcommand provides a summary of the imputation model and useful descriptive statistics on missingness. The `/CONSTRAINTS` subcommand allows variables to be included only as predictors in the imputation model, as imputed variables or both. Here the constraints are unnecessary (because only `y.do` has missing values).

Some SPSS commands automatically recognize the imputed data sets and perform a pooled analysis when run from menus. If using syntax directly, the crucial steps are to make sure the imputed data sets are in the active window and to specify a subcommand such as `/MISSING = LISTWISE`. This is required to analyze the original partial data set for comparison with the pooled and imputed data analyses. The following syntax runs the simple linear regression of the imputed outcome variable `y.do` on `x` and pools the output.

```
DATASET ACTIVATE partial_m40.

REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /NOORIGIN
  /DEPENDENT y.do
  /METHOD=ENTER x.
```

The pooled analysis gives a very similar output to that obtained using *mi* in R when $m = 40$. The pooled intercept for Example OS2.1 is −0.053 with an *SE* of 0.144 and the slope is 0.682 with an *SE* of 0.124. SPSS also reports *t* and *p* values (reported under 'Sig.').

SPSS also has features for inspecting the imputed data (including 'marking' them by highlighting imputed values in a different color) and selecting imputed data sets in the data view (e.g., look at the <Edit> and <View> menus). If an imputed data set is active, SPSS also identifies all analyses that can pool imputed data sets with an imputed data icon.

OS2.4 Notes

1. The probability of no missing data for a case is $.95^{10} \approx .60$ and so $(1 - .60) = .40$ is the proportion of cases with missing data.
2. If the data simply do not exist (rather than being potentially available but missing), the dummy variable approach may still be appropriate.

OS2.5 References

Collins, L. M., Schafer, J. L., and Kam C. M. (2001) A Comparison of Inclusive and Restrictive Strategies in Modern Missing Data Procedures. *Psychological Methods*, 6, 330–51.

Gelman, A., and Hill, J. (2007) *Data Analysis using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Graham, J. W. (2009) Missing Data Analysis: Making it Work in the Real World. *Annual Review of Psychology*, 60, 549–76.

Jones, M. P. (1996) Indicator and Stratification Methods for Missing Explanatory Variables in Multiple Linear Regression. *Journal of the American Statistical Association*, 91, 222–30.

Little, R. J. A., and Rubin, D. B. (1987) *Statistical Analysis with Missing Data.* New York: Wiley.

Peugh, J. L., and Enders, C. K. (2004) Missing Data in Educational Research: A Review of Reporting Practices and Suggestions for Improvement. *Review of Educational Research*, 74, 525–56.

Rubin, D. B. (1987) *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.

Schafer, J. L., and Graham, J. W. (2002) Missing Data: Our View of the State of the Art. *Psychological Methods*, 7, 147–77.

Su, Y. S., Gelman, A., Hill, J., and Yajima, M. (2010) Multiple Imputation with Diagnostics (mi) in R: Opening Windows into the Black Box. *Journal of Statistical Software.*

View publication stats