CREATING LARGE-N DATASETS FROM QUALITATIVE INFORMATION: LESSONS FROM INTERNATIONAL ENVIRONMENTAL AGREEMENTS

Ronald B. Mitchell
University of Oregon
rmitchel@uoregon.edu

Steven B. Rothman
University of Oregon
srothma1@uoregon.edu

Prepared for delivery at the 2006 Annual Meeting of the
American Political Science Association, September 1-3, 2006.

Draft. Comments welcome.

This paper is based upon work supported by the National Science Foundation under Grant No. 0318374 entitled "Analysis of the Effects of Environmental Treaties" September 2003 - August 2006. The International Environmental Agreements Project website is: http://iea.uoregon.edu/.

Abstract

Various researchers have created large datasets of variables related to war, economic sanctions, human rights, and comparative political institutions. The value of these datasets in fostering research progress in international relations depends on their providing high-quality quantitative data on fundamentally qualitative variables. Developing such large-N datasets that accurately capture variables not easily quantified requires careful attention to a range of factors involving measurement theory, particularly construct validity and reliability, and management of a research team. The paper argues that good dataset creation should develop variables and values that reflect both conceptually useful categories and accurately capture empirical variation, define the population of cases and identify members of that population, and systematically collect evidence on those cases. Coding manuals should have clear, complete, explicit, and well-documented coding rules and procedures. Training and coding should be conducted in ways that mitigate the introduction of error into the dataset. All dataset creation procedures should be carefully documented and made transparent to users, with special attention paid to providing users with evidence regarding dataset reliability and dataset construct validity.

Acknowledgements

This paper has benefited from helpful suggestions from Mark Axelrod, Thomas Bernauer, Paul Diehl, and Gary Goertz.

Introduction

Political scientists use two quite different types of data in quantitative analyses: inherently quantitative data or originally qualitative data. Datasets based on inherently quantitative evidence are composed of numbers corresponding to variables that are conceptually numeric. Population, numbers of nuclear weapons, battle deaths, or pollution levels are inherently numeric variables for which the question "should I trust this data?" translates into "how well was this counted?" In practice, most such variables are estimated rather than counted and various sources of error may create large discrepancies between the recorded and the true values of any particular variable. Yet, dataset creation requires little, if any, coder interpretation to record the values of such variables. Clear definitions of what to tally can be generated relatively easily and the problem becomes one of assessing how well objects (people living in a country, people killed in a war) were tallied.

By contrast, datasets based on originally qualitative evidence are composed of often numeric, but always quite simplified, representations of variables that are conceptually not numeric. Such datasets require that originally qualitative evidence be coded to transform it from evidence into data. A country's status with respect to democracy, protection of civil liberties, or labor rights (Marshall and Jaggers 2002; Freedom House 2006) are variables the coding of which require considerable interpretation at fundamental levels. Creating datasets with such originally qualitative evidence faces the same practical problems as with inherently quantitative evidence but also requires a far larger degree of interpretation. This need for interpretation has two important implications for those using such data. The first, well-recognized, implication involves construct validity, i.e., that the database user may not accept the database creator's definition of "what counts" as democracy, protection of civil liberties, or labor rights. A second, less frequently recognized, implication relates to reliability, i.e., that, even accepting the database creator's definitions of variables and values, the recorded values for each observation do not represent an accurate mapping of empirical evidence to those variables and values. In essence, the problem is one of ensuring that the variation in the coded data matches, as much as possible, the variation in the underlying primary evidence.

This paper uses the experience of the International Environmental Agreements (IEA) Project to identify a set of criteria and procedures for developing high-quality datasets from originally qualitative evidence. It provides some definitions useful to the subsequent discussion, delineates the goals of coding, and details procedures for developing and presenting large-N data with the greatest value to other scholars, accuracy, validity, and reliability.

Definitions and IEA Project background

Extensive scholarship exists on theories of measurement, from a wide array of fields beyond political science. Although measurement is an appropriate term for the creation of many datasets, we believe that "coding" is a more appropriate term when generating data that is nominal rather than numeric. We define coding as the translation of originally qualitative evidence into systematically comparable data. We define evidence as the undifferentiated body of information related to particular cases from which the values of specified variables will be derived. We define data as the values assigned to particular cases for specified variables, where the values and variables are analytic constructs developed by the dataset creator.

Coding involves efforts to categorize: a) a large number of phenomena or observations according to their variation with respect to b) one or more variables each of which has a limited, explicit, and clearly defined set of values, c) in a way that allows subsequent comparison of those phenomena with respect to those variables. The coding process may, but need not, produce quantitative data. Thus, this definition accepts all three of Freedom House's annual rankings as qualitative codings, even though political rights and civil liberties receive numeric codings from 1-7 while freedom status receives non-numeric codings of F, PF, and NF (Freedom House 2006). Schematically, qualitative coding can be thought of as the effort to fill in the cells in a spreadsheet in which the rows correspond to observations or cases and the columns correspond to different variables. Entries in the cells in any given column are limited to specific alternatives (the potential values of that variable).

We limit our definition to large-N efforts since systematic comparison of relatively small sets of cases can be accomplished with various other techniques, such as structured, focused comparisons or process tracing. As one moves to comparing a large set of cases, however, the time and resource constraints make most alternatives significantly more difficult or impossible to do in a way that allows systematic comparison. That said, we believe much of our discussion may be useful to those conducting comparative case studies.

We limit our discussion to efforts to categorize cases in terms of variables, which have limited, explicit, and clearly defined sets of values. While we recognize that one can categorize a large number of cases in ways that do not constrain potential values of each variable (or even the variables), doing so inhibits or prevents systematic comparison.

We distinguish variation in any dataset as either "empirical variation" or "error." These correspond to signal and noise, in the sense of the former being the "true" variation that would be captured in a dataset if a "perfect"

translation from evidence to data occurred. The error, or noise, is additional variation introduced by the coding process. Good dataset creation minimizes the introduction of error into the dataset and thereby maximizes the correspondence between values in the dataset and the values of the underlying cases.

Finally, note that we do not distinguish between independent and dependent variables. The strategies for developing accurate data are independent of whether that data is intended for use as an independent or dependent variable. This is especially true for scholars creating datasets for use by others since other scholars may well use the data in ways different from those intended by the original dataset creator.

The International Environmental Agreements Project

The International Environmental Agreements (IEA) Project set out to create a dataset that would provide systematic data on, and foster systematic research of, variation in international environmental treaties. Several websites already provide free-text or keyword search capabilities in which user-entered keywords are matched to words in a large (though often unsystematic) selection of international environmental treaties [see, for example, online datasets from CIESIN, INTFISH, FAO, and ECOLEX]. Such websites provide useful evidence but not data, in the sense that they do not facilitate efforts to identify the extent to which different agreements vary with respect to a given characteristic. Thus, consider an effort to identify all environmental treaties that include "enforcement" or "scientific" provisions (let alone, those that have "strong" versions of such provisions). Enforcement may be referred to as fines, sanctions, revocation of membership privileges, and a plethora of other options. Scientific research may be referred to with variants of the word "science" but may also be referred to as data gathering, environmental monitoring, or collaborative research. These examples suggest the limited utility of free-text searching for systematic research. First, given a desire to identify all treaties with particular provisions, one must search for a large number of words and phrases that one cannot identify deductively; yet, one faces a Catch-22 in which, to identify all the IEAs that refer to enforcement or scientific research, one must already know which IEAs refer to enforcement or scientific research to identify the terms by which they do so. Second, even carefully developed sets of search terms will generate Type I and Type II errors, i.e., will provide agreements that include the search term but do not relate to the desired concept while missing agreements that do not include the search term but do relate to the desired concept. Third, even successful identification of all agreements that refer to particular concepts is only the first step toward systematic analysis. Knowing which agreements have provisions that mention "enforcement" or "science" provides only limited insight into how those agreements vary with respect to those characteristics.

The IEA Project seeks to address these and related problems by creating a coding system that establishes a set of "variable categories" that, collectively, allow all treaty provisions to be coded as having information related to at least one of those variable categories, with subsequent coding of those provisions to identify the values of variables within each variable category. The IEA Project is particularly interested in ensuring that the data created reflect concepts useful to future researchers and are generated in a way that maximizes the degree to which they reliably and consistently map empirical variation to the permitted values and variables in the dataset.

Goals of coding

The goal in creating a dataset of IEA provisions -- a goal we believe is shared by many people and organizations that develop datasets -- has been to characterize the variation of a large number of empirical phenomena using conceptually useful categories in such a way that users can be confident that the mean and variation of the data generated corresponds as closely as possible to the mean and variation of the empirical phenomena being categorized. Scholars rarely create datasets as an end in themselves but as a necessary, perhaps unpleasant, prerequisite to answering questions of interest. Even datasets originally created by scholars for personal use can prove valuable to others, which is usually a requisite to obtain outside funding. However, data is usually most useful to others when created to have construct validity, accuracy, and reliability.

- A dataset has *construct validity* to the extent that a dataset's values correspond to and capture the concepts they claim to capture (DeVellis 2003). A dataset's construct validity corresponds to the degree to which, if coding rules and procedures were implemented perfectly, the mean of the resulting data would correspond to the construct that the coding rules claim to be coding.
- A dataset is *accurate* to the extent that the values in the dataset have a high probability of being the "true" values for the given variable and case (Neuendorf 2002). A dataset is accurate when each value corresponds, on average, to the "true" value for that variable and case, which we may not know, of course. The greater the number of values that diverge from the true value for that variable and case, the less accurate the entire dataset. In coding qualitative datasets that include nominal variables with two or more values, we are interested in something parallel to but different from a confidence interval, in the sense that we want to know how likely it is that each value of a nominal variable is the "true" value of whatever

phenomena was actually measured. What is desired, but impossible to achieve, is a confidence interval around each value to evaluate the accuracy of the dataset. Thus, one could have an accurate dataset with low construct validity if all dataset values had low variance around their respective means but those means generally did not correspond well to what the dataset creator claimed to be coding.

- A dataset is *reliable* to the extent that the codes contained in a dataset for a given variable and case would be the same if that variable and case were coded again, whether by a different coder or by the same coder at a different point in time (Carmines and Zeller 1979; Neuendorf 2002). A dataset's reliability corresponds to the degree to which the mean and variance of one coding of a variable correlate with the mean and variance of an independent coding of that variable. In essence, reliability reflects the extent to which the coding process introduces error in the dataset. The goal is to ensure that dataset values are coder-independent and sequence-independent, i.e., do not depend on who coded the case or on whether a case was coded before or after any other case.

Thus, dataset creation requires attention to several tasks, developed more fully in the rest of this paper. The dataset creator must create conceptually useful categories, i.e., they must create categories that correspond to variables (and potential values) of interest to potential users. The dataset creator must define a population from which cases will be chosen, identify the cases from that population that will be included in the dataset, and collect evidence related to the variables of interest for all included cases. The dataset creator must operationalize those variables, i.e., define variables and their potential values, in ways that facilitate systematic categorization of evidence from the cases. The dataset creator must train and deploy people who can apply this coding system to capture as much empirical variation as possible while minimizing the introduction of error. Finally, the dataset creator must document these processes in ways that allow dataset users to assess the dataset creator's success in achieving the goals.

Identify variables of interest and their potential values

Dataset creation requires converting scholarly interest from an often initially vague interest in the relationship among variables into a systematically identified list relevant dependent variables, corresponding independent variables of interest to the researcher, and other independent variables corresponding to rival hypotheses.

Initial identification of variables of interest, and their potential values, usually results from an organic process of a scholar's previous research. For dataset creation, that initial identification must be systematized through a comprehensive literature review aimed at identifying the range of variables scholars consider germane to the question at hand, and the lists of potential values scholars have identified for each. Scholars frequently define variables and their potential values in inconsistent ways. A dataset-driven literature review should be particularly attentive to such inconsistencies, since a large dataset may have opportunities to collect evidence that can arbitrate between different definitions or interpretations. Although variables and values previously identified by scholars is a necessary starting point for dataset creation, it is not sufficient, since dataset creation requires ensuring that the categories used in coding map not only theoretical categories but also empirical variation, as discussed more below.

Define the population of cases to be studied and identify the members of that population

Dataset creation also requires clearly defining the population of cases that the dataset targets and then identifying the members of that population. Defining the population carefully requires attention to the same issues when coding data because defining the population is a form of coding. Regardless of whether the dataset creator intends to code all cases in a population or only a sample of those cases, those using, or reading results derived from a dataset cannot generalize accurately unless the relationship between the cases studied and the relevant population is known.

Identifying the relevant population of cases can often prove quite challenging. In the IEA Project, initial efforts quickly demonstrated that, despite the availability of many lists of "international environmental agreements" (IEAs), little clarity or consensus existed on what to included in those lists. All three words in the phrase "international environmental agreement" needed clearer and more explicit rules in order to identify cases or observations to include in the dataset. The IEA Project needed clear rules about whether agreements between a country and an international organization would be considered "international," whether agreements would be considered "environmental" based on their goals or their effects, and whether non-binding agreements would be considered "agreements." Such rules, even though dealing only with case selection, are themselves coding rules. Cases included in a dataset must be chosen from a longer list of potential cases. In most datasets, similar questions will arise (as with the Correlates of War Project's need to define what constitutes a "war"). Dataset creation methodology cannot specify how any particular population should be defined but it does suggest three rules: a) that

population-defining rules be clear, explicit, and public, b) that those rules correspond as much as possible to common understandings of the terms used to refer to the population of cases, and c) that they clearly distinguish between cases that are part of the population and those that are not.

Once clear definitions for a population have been established, the dataset creator must identify the members of that population. Even if the dataset creator has no intention of coding all potential cases, the dataset creator should have a clear sense of what the members of that population look like so that they can clarify for themselves and potential dataset users what is the relationship between included cases and the population. Users need to know whether cases are representative of the population and, if not, the ways in which they are not.

Identifying all cases in a defined population is challenging unless the researcher selects all, or some subset of, cases from a previously identified population of cases. Even in such cases, however, the dataset creator should note and document the coding rules used in generating that earlier population of cases. Thus, using the UN Treaty Series as the population of agreements from which one draws ones cases has significant implications for the types of agreements in the dataset (e.g., regarding the under-representation of bilateral agreements and treatment of the distinction between binding and non-binding agreements). When dataset creators are breaking new empirical ground in which no previously identified population of cases exists, generating a comprehensive list of cases can pose serious obstacles. For the IEA Project, this involved efforts to identify all agreements between governments of any sort and then determining if they were international, environmental, and agreements, given the Project's definitions of those terms (see Mitchell 2003). But generating that "superlist" of agreements from which a population of IEAs could be identified was itself challenging, requiring combining all lists claiming to contain international environmental agreements (none of which proved fully comprehensive), agreement lists available from foreign and environmental ministries (particularly to identify bilateral agreements), lists available from agreement secretariats, and identification of individual agreements that were referenced in other agreements.

Once the population of cases and the set (whether the full population or some subset) to be coded have been identified, it is also important to identify subsets of the cases to be coded that can be used for coding manual development and for training. Coding manual development requires examining a subset of case material before coding begins in order to create variables and values that reflect empirical variation as well as theoretical categories (see Neuendorf 2002). Training requires examining a subset of case material so that coders gain experience applying the coding manual and so that their skill in applying the manual can be evaluated before they begin coding data that will enter the dataset. In both cases, subsets of representative cases must be set aside to avoid "tainting" the data that enters the dataset.

Methodological concerns dictate that one should not test a theory using the evidence from which the theory was derived. The argument for dataset development is similar. To ensure that the categories created for a dataset apply to all members of the population, those categories should be generated from a subset of that data and then applied to the remainder of that dataset. Put differently, evidence should be coded using a coding system developed independently of that data. This independence can be ensured only by developing the coding manual based on a review of a subset of evidence, finalizing the coding manual after that review, and then coding the cases not in that subset. Subsets used for coding manual development or training should be representative of the population of cases to ensure the coding rules can be successfully applied to all cases that might be coded. These subsets should be large enough to contain examples of all types of variation in the population so that such variation can be included in the variables, values, and coding rules and procedures.

Extracting subsets of cases that cannot be included in the dataset, at least initially, is costly, to be sure. Such cases cannot be immediately coded because coding rules will both reflect, and may use examples from, the cases used in coding manual development. This essentially "tells" coders what values to assign for those cases rather than merely the general rules that they will need to apply in the cases not in the subset. Training involves considerable interaction among coders and with the dataset creator; therefore, cases used in training cannot be included in the dataset because they are not coded independently. Cases in these subsets can be included in the final dataset if certain strategies are employed. If coding of the non-subset cases takes sufficiently long and the same coders are still coding, it may be appropriate to have those coders code those cases at the end of the coding period but being cautioned against using any examples in the coding manual. A safer approach relies on revising the coding manual to excise all references to the cases in the subset, training new coders using this revised manual and already-coded cases, and having the new coders code the original training and coding development cases. Both strategies seek to ensure that, to the extent possible, cases are all coded through identical procedures.

Identify the evidence to be collected on each case and collect it systematically

Collecting data for a large-N study requires a degree of systematicness not needed for small-N studies. In small-N studies, it is usually less difficult and less inefficient to collect additional evidence after analysis has begun.

Generating comparable data for a large number of cases, however, strongly recommends that evidence be collected systematically so that all necessary evidence for all cases is available when coding starts. Initial collection of evidence, therefore, should gather comparable evidence through comparable methods for all evidence relevant to all variables of interest.

Comparable data cannot be generated from non-comparable evidence. The most common problems arise when evidence of the value of a variable is readily available for some cases but not others. Even if evidence on the latter cases exists and is later collected, such an approach is inefficient relative to collecting it during the original data collection effort. Collecting evidence at different times or by different procedures may also introduce error in the dataset since evidence originally available may become unavailable or may only be available in some non-comparable format. The IEA Project defined "evidence relevant to each variable" as the text of the IEA (we view agreement texts as important, even if they differ from actual implementation). This choice vastly simplified evidence collection, by identifying what evidence to collect and leaving no uncertainty if such evidence were not available. Even with such a clear and neatly bounded set of evidence to collect, problems have arisen, not least being that finding original versions of older agreement texts and of agreements that have been amended is often more difficult than one might assume. For datasets that seek information about cases and variables that involve less well-bounded questions (consider the evidence related to Freedom House, Polity, or Correlates of War data) entrains far greater challenges in defining what constitutes relevant evidence for each variable, identifying appropriate and comparable sources for such evidence, determining when to stop searching for relevant evidence, and defining how to interpret the failure to find evidence on the value of a variable.

Develop a coding manual

Creating a large-N dataset from originally qualitative evidence requires developing a coding system of both coding rules and coding procedures. A coding manual consists of an explicit and complete delineation of the coding rules and procedures used for assigning the value for a particular variable to a particular case based on the available evidence. A coding manual delineates the rules by which the dataset creator seeks to map conceptually useful categories onto empirical phenomena or, alternatively, to categorize extensive empirical variation into a relatively few conceptually useful categories. Large-N datasets and their coding systems link the theoretical and the empirical and their development, therefore, involves an iterative interplay between deductive conceptualization and inductive operationalization.

A coding manual disciplines the process of identifying the variables to be entered in the dataset, their possible values, and the rules by which empirical material should be assigned those values, i.e., should be coded. A coding manual should minimize interpretation as far as possible during coding to maximize the degree to which all cases are coded in the same manner. And a coding manual should be developed to serve as the primary means to communicate coding rules not only to coders, so that they code reliably, but also to users, so that they can interpret the data appropriately. Good coding manual development, therefore, should:

- ***decide on selective or comprehensive coding,*** so that it is clear whether all evidence or only some subset of the evidence will be coded,
- ***be clear, complete, and explicit,*** so that all rules and procedures used for coding the data are documented,
- ***ensure variables and values reflect theoretically useful categories,*** so that the resultant dataset or analyses based on it have value to other scholars,
- ***ensure variables and values reflect empirical variation,*** so that the resultant dataset or analyses based on it correspond, as much as possible, to meaningful (as opposed to arbitrary) empirical distinctions,
- ***ensure values (and possibly variables) are mutually exclusive and collectively exhaustive,*** so that any given piece of evidence can be assigned one and only one value for a given variable, and so that all evidence can be assigned at least one value.

Coding manuals: deciding on selective coding or comprehensive coding

A dataset creator must first decide whether to conduct selective or comprehensive coding. Selective coding, commonly described in coding textbooks (e.g. Dawis 2000; Neuendorf 2002), asks specific questions of the evidence from each case, looking for particular, previously identified, types of variation in the cases examined. Comprehensive coding seeks to identify such variation as exists across a set of cases. Selective coding strategies are far more deductive, with more value placed on capturing the values of theoretically informed variables. Comprehensive coding strategies are more inductive, with more value placed on capturing all "potentially interesting" variation, even if such variation has not previously been identified as theoretically interesting.

Selective coding has several virtues. Variable and value categories are more likely to correspond to the theoretical distinctions from which they are derived. Selective coding almost always involves fewer variables, which makes coding easier and cheaper to implement. That said, selective coding leaves some evidence uncoded and may miss variation that a more comprehensive evaluation would have identified. Equally important, scholars interested in variables related to the dataset's cases that the dataset creator did not code must return to the original evidence.

Comprehensive coding has some of the advantages of qualitative research in inductively identifying important but previously unrecognized variation. As with case studies, the value of comprehensive coding depends considerably on the dataset creator's skill at "soaking and poking" (Feldman 2005) and identifying "potentially interesting" variation from "truly uninteresting" variation. When the evidence related to the cases to be coded is clearly defined and clearly bounded, comprehensive coding strategies have the particularly attractive feature that they can be designed to ensure that all available evidence is coded in some fashion. In a comprehensively coded dataset, for each case, each portion of the body of evidence is coded to identify the variable or sets of variables whose values could be identified from that portion.

Since the IEA Project seeks to foster research by a broad set of users interested in IEAs, comprehensive coding was an obvious choice. The IEA Project sought to reduce the occasions on which users would ask: "but why didn't they collect data on X?" Dataset creation usually involves sifting through a large amount of evidence irrelevant to the variables being coded. One cannot, of course, code every variable of interest to all potential future users. But, with projects (like the IEA Project) based on clearly bounded sets of evidence, it is possible to parse all evidence by assigning "variable-level" codes to ALL evidence from each case (i.e., to every provision in each agreement) according to the variables (or categories of variables) to which that evidence relates. This parsing of all evidence has proved useful in developing coding manuals, in training and coding the variables identified as interesting to the Project, and for providing a more efficient way of recoding any evidence if necessary. Equally important, however, making data from the comprehensive "variable-level" coding publicly available encourages future scholars to easily extract, for all cases, that subset of evidence relevant to coding new variables not coded by the Project without having to re-sift through a mountain of irrelevant evidence.

Coding manuals: being clear, complete, and explicit

The most important criterion for a coding manual is that it document all rules and procedures used to transform William James' "blooming, buzzing confusion" of good evidence into the systematic, structured clarity of good data. Documentation of coding rules serves two functions: in prospect, it tells coders how to transform evidence into data; in retrospect, it tells users how evidence was transformed into data. Both are crucial. Good coding manuals function like good legal contracts. Both should be sufficiently clear, complete, and explicit that they make it highly likely that different individuals will interpret specific, often unforeseen or unforeseeable, events in mutually consistent ways, even in the face of strongly divergent interests. Both also do so through general rules rather than by specifying a list of all possible events and their appropriate interpretations.

Consider two extremes of the coding continuum. At one extreme, a researcher might evaluate each case and assign values for relevant variables on an ad hoc and undocumented, if not fully random, basis. The dataset resulting from such a process would be likely to include many cases for which particular variables were not coded, many variables for which values were assigned on different bases depending on the case, and would provide little if any basis for knowing the extent to which either of these problems was true. At the other extreme, a researcher might establish extremely clear and rigid rules, transform them into computer algorithms, and run the resultant program on textual evidence for all the cases. Assuming proper computer programming, such computerized coding would ensure that all variables were coded for all cases and that values were assigned according to the same rules for each case. The computer would serve as a "coder" and the computer program would serve as the coding manual and, if properly notated, would document the evidence-to-data transformation rules for subsequent users. A coding manual should move toward the latter end of the spectrum in situations that require more human judgment and interpretation than computerized coding can currently provide.

Clear coding rules reduce the degree to which either conceptual ambiguity or the distribution of empirical variation allows coders or users to think it reasonable to code a given piece of evidence in more than one way. Coding involves two distinct judgments: is the evidence currently being examined relevant to the variable currently being coded and, if so, which of the potential values for that variable most closely fits the evidence. The clearer the distinctions among and boundaries between categories, the easier it is for coders to assign evidence to categories and for users to know why evidence was assigned to categories. The more the coding manual's distinctions and boundaries between categories correspond to the "natural" distinctions and boundaries found in the empirical evidence, the fewer instances in which coders will find assigning evidence to categories difficult and the fewer instances in which users will believe that codings were done in error. Put differently, categorizing evidence is easiest

when "bright lines" distinguish categories and when those bright lines fall in parts of the empirical distribution that have few cases. Yet, since it is not possible to develop perfectly unambiguous categories, coding manuals benefit by explicitly identifying situations that coders are likely to consider as ambiguous and challenging to code and providing clear guidance on how such situations should be treated.

Since individuals differ with respect to whether they can learn coding rules deductively or inductively, designing clarity into a coding manual also includes using empirical examples. Empirical examples should generally be hypothetical examples that abstract away from individual cases and with the number of total examples drawn from any given case kept to a minimum. A minor goal is to provide examples that help clarify coding rules without tainting the coding of particular cases by providing the "answers" for coding those cases in the coding manual. The more important goal, however, is to design examples to help coders learn the more abstract, conceptual, coding rules that can be applied to the range of future cases they will need to code. That is, a short list of well-chosen examples designed to move coders from the specific to the general is preferable to a long list of specific examples.

Clarity is also fostered not merely by delineating terms and coding rules explicitly but also by ensuring that the terminology used reflects, as much as possible, common understandings of the terms employed. To the extent that clear and explicit definitions of terms do not correspond with their everyday usage, both coders and users are likely to misinterpret those terms and thereby miscode the data or misuse it. A coding manual requires defining some terms in slightly different, usually more limited, ways than standard usage. But limiting such differences minimizes the likelihood that coders and users will "mistranslate" from the common definition of the term or phrase to the coding manuals definition. It therefore minimizes opportunities for coders to enter erroneous codes in the dataset due to their misinterpretation of terms and for users to draw erroneous conclusions from the dataset due to such misinterpretation.

Completeness in a coding manual involves providing guidance for all possible cases, for all possible variables, for all possible values of each variable, and for all procedures related to developing the dataset. For coders, completeness helps ensure that cases are coded in the same manner by providing answers to all possible questions and, equally important, providing the same answer regardless of when the question is asked or by whom it is asked. For users, completeness helps ensure that users know everything necessary to interpret and use the dataset's variables, values, and cases in a way consistent with how they were created.

Explicitness in a coding manual involves ensuring coding rules are precise and written. Using precise language reduces the chances for multiple valid interpretations of coding rules. Where one of a pair of synonyms has multiple meanings and the other does not, the latter should be used. Where the most appropriate terms have multiple meanings, carefully developed definitions can clarify how that term will be used in the context of the present dataset. As noted above, the fact that three terms central to the IEA Project -- "international," "environmental," and "agreement" -- had multiple meanings required the development of explicit definitions to allow systematic identification of the population of cases and to clarify to dataset users why numerous "important" cases included in other lists of "international environmental agreements" are not included in this dataset. Precise definitions of all variables and all potential values reduces time spent clarifying ambiguities and confusion during training and reduces the need for coders to determine or remember which of multiple possible interpretations of a term is the appropriate one to use in coding. Precise and complete delineation of procedures related to training and coding are prerequisites to ensuring that all coders are trained in the same way and that coders code all cases in the same way.

Written documentation provides the transparency that is, arguably, the feature most central to a coding manual's success. Regardless of what coding rules and procedures are used to create a dataset, so long as they are properly documented, any user can know how to interpret data in the dataset. Good coding manual development requires documentation "of the process *during* the process." Post-hoc documentation will usually miss many details of the coding process, details that may have substantial implications for dataset interpretation and quality.

Coding manuals: ensuring variables and values reflect theoretically useful categories

Developing a good coding manual requires transforming that initial, often ad hoc, list of variables that prompted a scholar to undertake dataset development (see above) into a systematic set of variables that can make the best use of the evidence and effort that will go into dataset creation. Once a set of cases and evidence have been identified, the dataset creator should refamiliarize themselves with the theoretical literature relevant to the questions and evidence to which the dataset will relate and, having done that, conduct a first-order, low-resolution, review of a representative subset (or all) of the evidence that will be coded. This effort should ensure that the coding process does not miss opportunities to code variables of interest to the broader scholarly community (but perhaps of little interest to the dataset creator), especially if they could be coded at little additional cost in time or resources.

Considerable attention should be taken to identify -- as clearly, completely, and explicitly as possible -- how each variable is defined, how it can be distinguished from other variables, and how previous scholars have

operationalized the variable. Out of this effort should come tightly phrased definitions, distinctions, and operationalizations for each variable that are structured, to the extent possible, in parallel phrasing such that distinctions between variables are more obvious. A similar process should address possible values for each variable. This often is more challenging than identifying variables, since scholars more often agree on an axis of variation than on the categories (or the names for the categories) that should be used to distinguish variation along that axis. Thus, people agree that color varies, but often disagree (or have different approaches to describing) variation in color. A coding manual becomes more useful if potential values are chosen for each variable so that the resultant dataset can at least contribute to, if not arbitrate, debates by showing which of several alternative ways of describing variation is easiest to identify empirically, is best correlated with other variables, or is generally more compelling. With respect to both variables and values, the goal is to ensure that the coding manual reflects theoretically useful categories and does not succumb to the temptation to focus excessively on the empirical evidence.

Coding manuals: ensure variables and values reflect empirical variation

A good coding manual should develop theoretically useful variables and their values, but should also capture empirical variation. This requires that the coding manual add, subtract, and modify variables and values derived from theoretical and often highly conceptual literature so that they correspond more closely to empirical boundaries. For coders, coding manuals that define and describe variables and values in ways that reflect empirical variation make coding easier and more consistent. For users, such coding manuals produce datasets that correspond more closely to meaningful empirical (as opposed to theoretical but arbitrary) distinctions.

Coding is easiest when cases fall into clearly delimited or "natural" groups with respect to their variation on a particular variable and, at the same time, the available categories of variables and values of the coding system correspond to those groups. Coding becomes harder as "natural" empirical boundaries begin to blur, as categories allowed by the coding system blur, or these categories are simply do not match. Coding is challenging if a coding system's clear boundaries between conceptual categories fall directly in the middle of numerous empirical cases. It can be equally challenging if empirical distinctions are obvious but the lines between conceptual categories are unclear. Developing a good coding manual requires that it be developed with knowledge of the types of variation exhibited by the cases planned for coding, best accomplished by review of a representative subset of cases.

Dataset creation begins with a large body of evidence differentiated only by the case to which it relates. Refining potential values for each variable, however, is most readily accomplished by looking at evidence by variable rather than by case. That is, determining whether a given variable should have two, three, four, or five possible values (and what those values should be) is more easily accomplished by examining a range of cases, but only examining the evidence related to that variable. Coding manual development occurs by repeating that process for all variables.

The IEA Project has developed (or perhaps reinvented) a strategy of initial "variable-level coding" that parses the evidence from a large and representative subset of cases into categories based on the variables to which that evidence relates. Although this process requires that all evidence be categorized as related to at least one variable, it does not require that evidence be categorized as related to only one variable since the "natural" points for dividing evidence often may include evidence related to several variables. Thus, single paragraphs (and sometimes even sentences) in many international treaties include evidence related to membership, entry into force, and reservations or evidence related to substantive requirements and implementation issues.

Variable-level coding consists of creating a set of collectively exhaustive values for a variable called "This evidence is related to." Values for generated from an initial literature review must be modified, however, because of at least two problems. First, most bodies of evidence include considerable empirical variation in which scholars are not interested or in which the dataset creator was not initially interested. Despite that, categorizing such evidence can help both future scholars and the dataset creator access and review that data. Thus, the clauses in most international treaties specifying official languages for the agreement or designating a country or organization to serve as depositary are unlikely to have much scholarly import. Yet, creating "Official Texts" and "Depositary" categories helped ensure that the coding system captured all evidence. Second, even after adding such categories, the list of possible values may only be made collectively exhaustive by adding categories of "Evidence unrelated to any identifiable variable" and/or "Evidence collected in error."

Developing a variable-level coding manual and applying it to a subset of cases provides a foundation for developing the value-level coding manual that is the ultimate objective of dataset creation. Improvements can often be made to a list of potential values for variables that have been identified solely from theory. Examining evidence from a subset of cases allows the dataset creator to add to, subtract from, or alter the list of variables to be included in the final dataset. First, a review of available evidence may suggest additional theoretically interesting values embedded in the evidence that might otherwise have escaped the dataset creator's notice. Second, as with variables,

comprehensive coding may require creating inductive values of variables that have little theoretical interest but allow all available evidence to be coded. Third, the values of some theoretically interesting variables may simply not be identifiable from the evidence available in the cases, suggesting that certain variables be dropped. Fourth, theoretically derived definitions of values may not fit well with the available evidence, suggesting particular redefinitions of values and the boundaries between values of a variable.

Examining only the evidence related to a particular variable makes it far easier to modify the list of potential values for that variable so it reflects empirical variation. Separating such evidence from the other evidence irrelevant to that particular variable requires having not only developed a variable-level coding manual but also having applied it to a representative subset of cases.

Choosing values that make coding easier: ensuring potential values match obvious empirical breaks

Coding is made easier, and hence coders can be trained more quickly to code more consistently, if the "bright lines" distinguishing potential values for a variable correspond to empirically "bright lines." Consider two distinctions from the international organization literature that scholars usually consider conceptually distinct: sanctions vs. rewards and binding vs. non-binding agreements. In cases like the former, empirical variation requires a "both" category rarely identified or considered theoretically interesting. Schelling's famous quote that "a promise is costly when it succeeds, and a threat is costly when it fails," (Schelling 1960, 177) would appear a reliable rule for coding rewards and sanctions. Yet, donor countries often provide assistance with the understanding that it will continue if recipients continue cooperating and will end if they do not. Thus, although no conceptual need exists for a "both rewards and sanctions" category, an empirical one certainly does.

The binding/non-binding distinction illustrates a separate point: that conceptually clear distinctions may be difficult to identify empirically. Agreements are considered legally binding if states consent to be bound (see the Vienna Convention on the Law of Treaties, Articles 2(1)(a) and 11 through 17 but also Aust 2000, 14; Brown Weiss 1997). Yet, it is often unclear and/or contested whether a given agreement is legally binding. International agreements can usually be identified as binding or not based on their titles: terms like Convention and Treaty are used for binding agreements and Declaration or Statement are used for non-binding agreements. However, the term "Memoranda of Understanding" (MOUs) is usually, *but not always*, used for non-binding agreements (Aust 2000) and, therefore, an accurate determination of which MOUs are legally binding requires the collection of more, and more difficult to acquire, evidence. The tradeoff between dataset accuracy and resources needed for evidence collection may lead the dataset creator to create a coding rule that codes all MOUs as non-binding. Likewise, it is unclear how to code intergovernmental agreements that some parties view as binding and others do not. In these and other, similar, situations, the dataset creator may simply need to make a judgment call regarding how to code. To allow the dataset creator to revisit that judgment and users to disagree with that judgment, good practice dictates adding a coding rule that identifies variables and values for which such judgments are made to allow easy recoding of those cases subsequently. Thus, theory would dictate a list of values for "bindingness" of binding or non-binding while the empirical evidence suggests that list also include MOU, contested, and no evidence available.

Coding system development also requires ensuring conceptually identified values map well to breaks in the empirical distribution. The ease -- and therefore reliability -- with which coders can categorize empirical variation into conceptual categories depends on the number of cases that fall near the transition zones among different potential values of a variable. Thus, consider a variable for which theorists have identified two ideal type values (e.g., A and not-A as illustrated by democracy and non-democracy) as ends of a spectrum; if cases are distributed bimodally, then the coding system can rely on existing definitions of those values even if theorists have not clearly identified where the transition from A to not-A occurs. Coding systems should clarify vague conceptual distinctions if many cases are distributed normally near those transition points, however, because the failure to do so not only makes coding far more difficult but also results in data based on an arbitrary distinction between A and not-A cases that does not correspond well to the empirical variation as depicted in Figures A1, A2, and A3.

Even clear distinctions may fail to capture empirical variation, however. When many cases cluster near conceptual transition points, it makes coding easier and ensures the data more accurately reflects empirical variation to create new categories, even if doing so requires some departure from conceptual categories. Thus, although free and not free states might represent a clear conceptual distinction, empirical variation might fall more neatly into three categories, as depicted in Figures B1, B2, and B3.

Ensuring that coding rules can be applied easily and consistently and that the dataset provides users a relatively accurate sense of the underlying empirical variation requires that coding rules be developed in response to a subset of evidence that can shed light on the distribution of empirical cases.

Creating data at the right level of resolution

Coding manual development also requires choices about data resolution. Too low a resolution, i.e., insufficient detail, risks creating too few categories and missing empirical variation which, in turn, effectively making the data more "noisy" and inhibits identification of correlations that do exist. Too high a resolution, i.e., excessive detail, risks creating too many categories so that underlying patterns also cannot be observed because too few cases are assigned any particular value and the number of variables and possible correlations is simply too large. Excessive detail also creates distinctions that are unlikely to be "meaningful," again blurring the ability to observe correlations that actually exist. Rather than a clear rule regarding the best level of detail, a "Goldilocks" standard seems in order, with variation and values defined to capture potentially meaningful empirical variation without wasting resources to produce so many values and variables that patterns in the data cannot be easily identified. That said, too much detail is generally preferable to too little, since excessive detail can usually be aggregated while insufficient detail cannot usually be easily disaggregated.

Coding manuals: ensure variables and values are mutually exclusive and collectively exhaustive

Variables and their potential values should also be defined to be mutually exclusive and collectively exhaustive. These criteria ensure that all evidence can be identified as related to a particular variable and can be assigned one and only one value for that variable. A comprehensive coding system, by definition, requires a collectively exhaustive list of variables so that all evidence can be categorized as relevant to at least one variable. But the value of datasets created for selective coding systems can be made collectively exhaustive by adding a residual variable of "Evidence not relevant to variables being coded," thereby allowing other scholars to code new variables by easily selecting case-related evidence that the dataset creator collected but was not interested in.

Categorizing all evidence based on the variables to which each piece of evidence is related (i.e., variable-level coding) permits coders to determine whether any given piece of evidence contains information relevant to coding a particular variable. This does not mean that a given piece of evidence cannot contain information relevant to more than one variable, however.

When designing potential values for variables, however, mutually exclusive and collectively exhaustive criteria are particularly important. Making potential values of a variable collectively exhaustive ensures coders can follow a basic rule that all cases must be assigned a value for all variables. Lists of potential values for all variables should include two generic values: "none of the above" to ensure the list is collectively exhaustive and "evidence on the value of the variable was not available" to ensure that each case is coded for each variable and avoid accidentally overlooked cases (see John and Benet-Martinez 2000; DeVellis 2003; Dawis 2000)]

Training

The foregoing procedures should produce coding rules, which, if applied "perfectly" to the cases selected, would produce values for all variables that would simultaneously reflect the coding rules and the empirical evidence. The problems, of course, arise because applying a coding manual's rules consistently to many cases is difficult. Regardless of how well a coding manual is written, high quality coding requires training coders. Careful training maximizes the likelihood that coders apply the coding rules as written and that all coders apply them in the same fashion and, hence, the correspondence between variation in the dataset and variation in real world phenomena. Coder-introduced error can be reduced by using non-expert coders and training the coders to a level of competency that minimizes external influences.

"Non-expert" Coders and "Expert" Coders

Before discussing training procedures, our choice to use independent non-expert coders deserves some explanation. Given constraints of time and money, scholars often develop datasets using "expert" coding systems (Neuendorf 2002, ch. 7). Expert coding usually involves a single individual's (usually the scholar creating the dataset) assigning values for all variables for all cases. Yet, having those who know a particular case (or particular variable) well code involve some costs when building large-N datasets. An expert's interpretation, intuition, and undocumentable evidence that are central to a case study's value can undermine the goal of large-N datasets to ensure that values, variables, and evidence are identified consistently. Individual scholars that undertake "expert coding" may fail to create clear and explicit coding rules before coding starts, may change the coding rules used (consciously or unconsciously) over time, and may fail to document the coding rules carefully. They are likely to code inconsistently across cases, even when they do establish clear coding rules in advance. They can introduce systematic error by coding in ways that differ from documented coding rules. And, precisely because their expertise will necessarily vary by case, they will tend to introduce random error in the dataset. They will draw on evidence and insights not available in the body of evidence that non-experts must use for coding, and those evidence and

insights will vary from case to case. For all these reasons, individuals implementing expert coding systems are likely to introduce bias and non-random error in datasets and to fail to document the data generating processes in ways that users can use to identify those biases and errors. Although expert coding may involve significantly less cost in terms of money and time, we believe that it comes at significant cost in terms of data quality. It is far better, in our opinion, to have fewer variables coded at a high quality than to have many variables coded at a lower quality.

The International Regimes Database (IRD) (Breitmeier, Young, and Zürn 2006) adopted an innovative expert coding approach that addresses some, but not all, of these problems. That project employed a collection of scholars and had each of two case experts independently code a large set of individual cases based on a common, pre-established coding manual. This strategy took advantage of deep case knowledge while mitigating the bias that expert coding often introduces by requiring careful use of a coding manual and by incorporating both codings of each case into the dataset. Such a strategy has considerable attraction and deserves careful consideration as a dataset creation approach. However, three disadvantages stand out. First, each scholar had prior knowledge of and predilections regarding the intellectual concepts underlying the coding manual and those likely lead them to code using both rules documented in the coding manual and, in addition, other undocumented rules from the coders' heads. Second, the knowledge and predilections of each expert differ from scholar to scholar with the result that the coding manual is interpreted differently for each case. Thus, intellectual predilections would lead contributing scholars to code the same case differently, even if variation in case expertise could be removed. Third, experts also vary in their level of case expertise, some being more expert than others. This also introduces variation that is difficult to document. The nature of the problems makes it difficult to determine the extent to which they plague the resultant database. Dataset creation requires not merely applying a coding manual consistently but allowing users to evaluate how well you have done. The IRD's use of a careful coding manual and procedures and the use of two expert coders to code each case made it more likely that codes for any individual case reflect both the coding manual and the available evidence. Its use of different coders for each case, however, introduces coder-related variation into the values of the variables that reduce the confidence users can have that one case's codings can be compared to another's.

The IEA Project chose to use non-expert coders to skirt some of these problems. Top-ranked undergrads from an introductory international relations class were selected as non-experts because of their lack of knowledge about either the case material or the theoretical underpinnings of the dataset's construction. The goal is to provide such non-experts with a coding manual and train them so that they will all code cases in a similar manner, introducing minimal bias or other variation along the way. Coder-induced variation cannot be eliminated, since each coder may have different backgrounds and interpretations of the phenomena being coded. Yet, the IEA Project sought to minimize the influence of these factors, largely through joint training and indoctrination to the principle of using the coding manual without interpretation. Using non-experts requires extensive training since they must be familiarized with basic knowledge not only of the cases being studied but also of both substantive and methodological concepts relevant to good coding. Yet, developing data that, as far as possible, reflects variation in the empirical phenomena and not additional variation due to the interpretations of any individual's idiosyncratic interpretations seems more likely to be achieved through the use of "non-expert" coders, thus warranting the additional time and costs spent on training and implementation.

Coder Training

Coder training is crucial regardless of how data is generated. Training should ensure that the coder understands the coding rules and how to translate unstructured evidence from each case into the structured data in the dataset. Although coding manuals should be as explicit as possible, all interpretation and judgment cannot be eliminated. Training helps develop consistent interpretation and judgment to minimize variation that might intrude due to either the person coding a case or when in the sequence of coding a case was coded. Training -- using ten or twenty cases in a "pilot study" mode -- would seem appropriate even for individuals developing datasets since it provides a coder experiential understanding of what coding involves, improving coding reliability across cases.

In multiple-expert coding systems, like the IRD's, training serves to familiarize the expert coders with the particular operationalizations and meanings of the terms, variables, and values of the coding manual. To the extent such training is effective, scholars who might otherwise interpret and apply a particular term in the coding manual differently would learn that, for the purposes of coding, the term should be treated in a particular way. Thus, the training would remove variation due to such different interpretations from the resultant database.

Likewise, in non-expert coding systems, training provides practice and experience with applying the coding rules to empirical case material and removing variation due to different interpretations of the coding rules. Yet, training may be more effective at removing such variation with non-experts, since they lack the pre-existing intellectual or ideological preferences that experts must suspend during coding.

Central to training, in all cases, is removing the variation that will arise if coders interpret or apply coding rules differently. Training recognizes that simply giving non-experts a coding manual and saying "apply it" would lead to significant variation in how any given case was coded. Training seeks to remove such variation through a process that creates convergence around a given view of what the coding manual requires and how it should be applied. To achieve that goal, training should be as similar as possible for all coders, usually done simultaneously with the same materials (Johnson and Bolstad 1974). Equally important, training involves having all coders regularly read, and re-read, the coding manual so they fully understand all coding rules and their application.

Training should then involve two processes to create convergence of views on how to code case material. The first process involves joint discussion and clarification of all aspects of the coding manual. However clearly written, any coding manual will have remaining ambiguities that can only be clarified through discussion. Meetings ensure all coders hear all of these clarifications. Having a person keep a meeting log to record all clarifications provides documentation of those clarifications for later reference and for any coders that could not attend a particular meeting. The second process involves generating "group think" among coders to maximize the odds that coders make similar judgments when coding evidence that does not fit well into coding manual categories. This means that the process for determining the appropriate code for particular evidence should be the same among all coders. This does not mean that coders should code based on how they think others might code the evidence. Rather, the group think involves training the coders to use the same logic and process for moving from evidence to code. This implies that interaction always occur among all coders and not among official or unofficial subgroups of coders. Coders should be forbidden from discussing coding rules, procedures, or cases except during training meetings.

No coding manual designed for complex evidence can remove all judgment and discretion from the coding process -- if it could, one could presumably use computer coding rather than human coders. Joint discussion of the "proper" interpretation of a particular coding rule among all coders helps create a common understanding of those rules, increasing the chances that all coders would code a new type of case or an ambiguous case in the same way. Wherever possible, such "proper" interpretations should be recorded in a revised version of the coding manual, but some part of training involves creating understandings that cannot be easily captured in writing.

Planning a training protocol requires identifying when training should stop and coding can begin. The goal is to ensure coders to achieve familiarity with the coding manual and consistency with other coders and validated codes for particular cases (i.e., "truth tapes") that one can have confidence that the data they produce will be both accurate and reliable (see definitions above). For the IEA Project, training for variable-level coding took three months and required coders to attend weekly training sessions, code numerous practice agreements, and take tests to assess their progress. Training concentrated on three tasks: introducing coders to the project, training coders in coding logistics, and training coders in the coding rules documented in the coding manual.

The first two weeks involved introducing the research assistants (RAs) to the project and the logistics of coding. Prior to an initial orientation session, RAs were required to read sample agreement texts, the NSF grant proposal, and published work associated with the grant and project. Orientation over the course of the first two weeks introduced coders to the detailed nature of coding, the project director's expectations, and logistics related to meeting schedules, software use, and documentation rules. The orientation sessions helped clarify expectations in ways that allowed RAs to self-select out of the project as well as for the project director to select out those lacking the requisite skills to be good coders. To address coder attrition during training and coding, more students were trained than needed.

The RAs were introduced to the 33 variable-level codes in four sequential groups ranging from easiest to hardest. Each week, RAs read the coding rules for one group prior to the meeting, discussed coding rules at the meeting, and received practice agreements. All RAs coded the same agreements. Our staggered training schedule (see Table 1) both introduced new codes and solidified knowledge of earlier codes by having RAs apply earlier codes to new agreements each week and having them apply new codes to agreements previously coded. Over the course of training, RAs became familiar with the coding rules, the coding process, coding logistics, and the empirical material (i.e., IEA texts) and could code entire agreements using all codes. RAs worked approximately 10 hours per week including a 2 hour meeting and 8 hours learning the coding manual and coding practice agreements.

Table 1 - Training Schedule

Week	Stage 1	Stage 2	Stage 3	Stage 4
1 & 2	General Orientation			
3	Agreements 1, 2	Learn		
4	Test	Test	Learn	

		Agreements 1, 2		
5	Agreements 3, 4	Agreement 3, 4	Agreements 1, 2	
6	Test	Test	Test Agreements 3, 4	
7	Agreements 5, 6	Agreement 5, 6	Agreements 5, 6	
8	Agreement 7	Agreement 7	Agreement 7	Learn
9	Test	Test	Test	Test Agreements 1, 2, 3, 4
10	Agreement 8	Agreement 8	Agreement 8	Agreements 5, 6, 7, 8

We designed and administered regular tests to assess RA progress and to expose RAs to common problems and difficult coding situations they would face during coding. The project director developed "truth tapes" that reflected the director's "best" coding of particular agreements. Along with the tests and intercoder agreement indicators, these "truth tapes" were used to assess coder progress. The tests were not used to "grade" RA performance, but to identify areas for improvement for particular coders, to determine whether training had accomplished the goal of generating high levels of intercoder agreement, and to provide a basis, if needed, to dismiss coders unable to perform coding well.

Several caveats on training are in order. First, as noted, coding of training agreements was done not-yet-trained RAs and was not done independently; therefore these agreements will need to be recoded subsequently before they can enter the dataset (Rothman 2006). Second, although the coding manual was "finalized" by the project director before training began, training identified some inconsistencies and confusing coding rules, which led to carefully documented changes to the coding manual. Training procedures were carefully documented and all RAs were required to keep work logs of time spent on different training aspects and of difficulties faced in coding. Third, when codes proved more difficult to learn than expected, training slowed down until coders gained confidence with that code with training taking longer than planned.

Evaluation of Training Process

Assessments of intercoder agreement were used to evaluate reliability against truth tapes and against other coders. Intercoder agreement measures the degree to which the codes applied by two coders match (Carmines and Zeller 1979; Cohen 1960).

To evaluate why several codes had poor reliability despite extensive training efforts focused on those codes, we provided our coding manual to Dr. Stuart Shulman and the staff of the Qualitative Data Analysis Program (QDAP) of the University Center for Social and Urban Research at the University of Pittsburgh who have extensive experience in coding a wide variety of text documents (http://qdap.ucsur.pitt.edu/). Shulman and his staff sought to replicate our coding efforts by training coders to code the same agreements we had coded. Despite their greater experience and expertise, and despite using a different training system, the QDAP staff achieved the same or lower intercoder agreement figures. Given that QDAP staff regularly achieve far higher reliabilities with other projects, we jointly interpreted this result as evidence of the complexity and difficulty of our coding goals rather than failures of our training or coding system.

Intercoder agreement (see more below) between coders and truth tapes averaged below 60% for codes at the start of each stage. After training, coder-truth tape agreement averaged between 70 and 90%. Our experience highlighted two conclusions. First, although training improves reliability for both easy and difficult, some codes appear inherently less reliable, starting with agreement levels below 60% but only improving to slightly over 70% after training. The ambiguity of the texts being coded and the difficulty of creating clear conceptual boundaries between some codes mean that some codes will always prove challenging, with a corresponding need to lengthen training time. Second, rather than introducing the easiest codes first and building up to more difficult ones, training might work better if coders are introduced to the most difficult codes first and train on them over a longer period of time.

Table 2 – Intercoder Agreement Indicators by Term

Code	Fall Term	Winter Term	Spring Term	All
TITL	100.00%	100.00%	100.00%	100.00%
SRC	100.00%	100.00%	100.00%	100.00%
CONC	100.00%	100.00%	100.00%	100.00%

TEXT	97.21%	95.04%	95.07%	96.72%
DEFN	90.11%	94.54%	100.00%	95.12%
ATTACH	96.23%	90.48%	84.66%	92.11%
MEMB	78.50%	93.72%	90.59%	89.24%
RELA	80.29%	87.26%	71.45%	87.91%
RESV	77.91%	89.20%	97.26%	87.76%
EIF	85.81%	89.53%	82.01%	87.06%
AMND	81.83%	92.47%	77.54%	86.32%
DESC	93.34%	82.56%	70.80%	83.70%
SCOP	75.77%	93.14%	89.66%	83.47%
DISP	67.10%	93.93%	100.00%	83.15%
FINAD	78.25%	90.31%	65.86%	80.13%
DEPO	75.97%	77.26%	81.96%	78.13%
NATBS	74.46%	90.72%	72.23%	76.16%
HOBS	52.71%	79.79%	71.71%	76.10%
GOAL	80.03%	61.36%	56.15%	71.61%
INFO	66.12%	73.91%	65.56%	70.10%
HOBF	58.47%	68.32%	68.14%	65.38%
SBS	60.57%	71.57%	70.80%	63.55%
SOVR	57.95%	49.05%	76.69%	63.51%
SCIR	65.51%	46.17%	64.38%	61.45%
SECS	57.62%	79.64%	90.74%	61.03%
SUBS	61.70%	50.21%	62.35%	60.92%
CONS	46.85%	79.82%	43.62%	57.25%
SECF	49.46%	59.24%	62.71%	53.32%
IMPL	46.48%	48.99%	43.09%	50.00%
FINPR	47.41%	44.89%	49.02%	49.87%
REVW	55.01%	49.72%	66.01%	47.95%
SBF	48.23%	60.59%	65.86%	43.47%
NATBF	72.46%	31.85%	19.72%	38.11%
SBF	48.23%	60.59%	65.86%	43.47%

Coding

Beyond coding manual development and training, coding procedures play an important role in dataset quality. Producing any large-N dataset usually involves coding by more than one individual, coding occurring over an extended period of time, or both. Requiring adherence to rigorous coding procedures minimizes the error that these, and other, factors introduce into a dataset.

Implementing a coding system requires attention to managing coders, dealing with the possibility of coder drift, and continuously evaluating the progress, reliability, and validity of the coding system. Good coding requires attention to finalizing the coding manual before coding starts, conducting variable-level coding before value-level coding, assigning evidence to coders, recording the codes assigned, assessing intercoder agreement, and conducting regular meetings and retraining sessions.

Finalizing the coding manual

Finalizing the coding manual before coding begins is crucial to ensure consistent coding over time and across coders. Because coding manuals must be developed using only a subset of available evidence, finalizing the coding manual before coding starts means, almost by definition, that there will be good reasons to change coding rules to better fit the empirical variation found in subsequent cases. A general rule to adopt is "no subsequent coding rule changes," accepting that the benefits bestowed by generating consistent and comparable data outweigh the costs of data that reflects empirical variation somewhat less well than it might otherwise. When such an approach proves untenable, a reasonable corollary rule is to ensure that "all changes that do get made must be carefully documented AND all previously coded cases must be recoded for affected variables."

A finalized coding manual provides a reference tool for all coders while they code. It helps coders keep track of details on a large number of variables and values and provides a "final word" on questions that arise. It allows all coders to reference a common document in determining how to code a specific case. In the IEA Project, coders were reminded during training and coding that the only acceptable response to why they assigned a particular code was to identify the exact provisions in the coding manual on which the coder had based their coding. They were also required to re-read the full coding manual at regular intervals during coding to refresh their knowledge of coding rules, thereby reducing each coder's "distance" from the coding rules and from the other coders applying those same coding rules.

Variable-level coding and then value-level coding

As already noted, the IEA Project has found more accurate datasets can be developed by separating coding into two distinct parts: an initial "variable-level" coding of all available evidence according to relevant variables followed by "value-level" coding that, sequentially, extracts evidence from all cases relevant to particular variables and then codes using the potential values for that variable.

Central to the logic of our approach is the notion that systematic coding of a large number of variables for a large number of cases is best accomplished by sequentially coding each variable for all cases (EVFAC) rather than by sequentially coding all variables for each observation (AVFEC). Consider a body of evidence related to a large number of cases, whether that be 200 countries, 50 states, 10,000 individuals, or 800 international agreements. There are several reasons to believe that coding all cases for variable Y, then for variable X1, then for variable X2, and so on through X10 will produce better results than coding the first case for Y and X1 through X10, the second case for case for Y and X1 through X10, etc. Although clearly a prediction about an empirical regularity, two factors would seem to support this claim. First, coding requires looking at evidence and placing it in mental categories. An EVFAC approach minimizes the number of mental categories the coder must keep in their head at any one time. Second, humans can make small distinctions among phenomena more accurately when the comparison involves phenomena that are relatively proximate in time or space. That skill falls off dramatically, however, as temporal or spatial proximity decreases.

Yet, many (perhaps most) social science database creators use an AVFEC rather than an EVFAC approach. This may reflect a desire to avoid the inefficiencies of sifting through a large amount of evidences to determine the value of a given variable and then re-sifting through that same evidence to determine the value of a different variable. In many cases, however, it seems likely that the efficiency gains of using an AVFEC approach comes at considerable (if often unrecognized) cost in terms of reliability. We believe an EVFAC approach can be accomplished relatively efficiently by undertaking variable-level coding followed by value-level coding, increasing reliability without excessive additional costs in time and resources.

Variable-level coding, once implemented, facilitates value-level coding by allowing coders, at any given point in time, to focus on only one or a relatively few variables, to focus on far less evidence, and to focus only on that evidence relevant to the variables currently being coded. Thus, the IEA Project hopes to identify variation in over 100 variables; each has 3, 4 or 5 possible values; producing a total of some 400 potential values. Any IEA may contain 20 to 500 subparagraphs, each of which may have information relevant to assigning any of those 400 potential values. Although numerous projects have faced more daunting coding tasks, describing the problem as "assigning one or more of over 300 values based on up to 500 paragraphs for each of 800 agreements" clarifies the inherent difficulty of such tasks.

The mental difficulty of the task, however, is driven by the number of variables and values and by the amount of irrelevant evidence than by the number of cases. Consider a reference case involving coding of a single case for a single variable. Increasing the number of cases to, say, 5,000 requires far more time but not significantly greater intellectual demands. The coder need only keep the distinctions related to that one variable in their head -- after recording their codes for each case, they can forget information related to that case and move to the next. By contrast, increasing the number of variables and values significantly increases the intellectual demands placed on the coder. The more variables and values in play, the more distinctions the coder must keep in their head simultaneously. Equally important, the more irrelevant evidence that must be evaluated only to be ignored, the more time between assignment of values for a variable and hence the more difficult it will be to make relative comparisons. Variable-level coding addresses these issues by transforming "too many variables and values and too much evidence" problem into a sequence of coding problems, each of which is a relatively easier "few variables and only relevant evidence" coding problem.

Assigning evidence to coders

Procedures for assigning evidence or cases to coders also influences reliability. As with other procedures, the overarching goal is assignment procedures that do not introduce error in the dataset but do allow coding reliability to be evaluated.

To avoid introducing error, evidence should be assigned to coders randomly. For variable-level coding, agreements have been assigned using a stratified random assignment method in which all "original" agreements were assigned randomly, followed by random assignment of protocols, and then random assignment of amendments. For value-level coding, coders will be assigned randomly to code specific variables for the evidence from all cases.

To allow subsequent reliability evaluation, the IEA Project is having all evidence coded by two independent coders. Double-coding uses considerably more resources but, we believe, produces higher quality data because it provides a means to assess the introduction of error, which can be used to determine whether data should enter the final dataset. A double-coding strategy requires forethought in deciding how coder disagreements will be treated. The IEA Project rejected "consensus coding" (in which coders themselves resolve disagreements) as a means of resolution since our coders were undergraduates for whom personality traits (e.g., self-confidence) might influence the resolution of disagreement more than substantive arguments. We also rejected resolution by the project director as essentially reverting to "expert coding," a strategy rejected initially.

When double-coding is used, the dataset creator must decide which codes will enter the dataset: codes from one coder pre-selected randomly; codes from both coders; or the average of both coders' codes. The IEA Project has adopted the "both coders" approach for variable-level coding since the subsequent value-level coding can easily re-classify "relevant" evidence as irrelevant to a variable but irrelevant evidence can only be reclassified as relevant by re-doing the variable-level coding. We plan to adopt a "one coder pre-selected randomly" approach for value-level coding, however.

Recording codes assigned by coders

Implementing a coding system also requires attention to coding logistics. The system should be designed to reduce random error (e.g., assigning a code to the wrong text, spelling a code wrong, assigning codes that are not permitted) and to facilitate subsequent manipulation of the data. A web interface provides a preferred method of data entry, that allows maximum flexibility in where coders code and can be designed to control data entry well and record data in a structure that allows subsequent manipulation. The IEA Project could not identify a programmer to design such an interface and opted to have students enter their codes in Excel spreadsheets that contained the agreement texts. This strategy ensured all students could work at home or on campus but allowed excess error in coder input. Coders typed codes into blank spaces, which allowed too many opportunities for misspelling and incorrect code placement.

In a situation similar to discovering new evidence related to a case, some of our original evidence (the treaty texts themselves) were found to contain, usually minor, errors. Such texts had to be corrected and then distributed to all coders responsible for coding that text. Both coders were required to recode the corrected text as a way to ensure reliable coding. The benefit of correcting original evidence even for minor errors is that it produces an improved body of original evidence that future scholars can use, in our case producing IEA texts in which various textual errors introduced by various sources have been removed.

Assessing intercoder agreement

Double-coding evidence has obvious costs, but also has benefits. Whether double-coding is done for all evidence or only a significant sample of evidence (usually 20-25%), double-coding improves training, coding, and documentation by allowing assessment of the quality of coding. When creating new datasets, knowing how well evidence is being transformed into data is challenging because of the lack of any baseline for evaluation. It is comparable to an exam in which the professor selected questions related to a particular vignette but had a teaching assistant choose the vignette and then failed to develop an "answer key." There is no way to know which students got the answers right. That said, if all students gave the same answer to each question, the argument would be supported that all students shared an understanding of how to interpret the questions and how to categorize evidence from the vignette. Even if all students gave an answer that the professor considered incorrect, the "wrong" answer would constitute the best estimate of the correct answer and, even more accurately, of how someone trained via the courses textbooks (coding manual) and lectures (training sessions) would answer those questions about that vignette. Note also that, when students have strong incentives to base answers on textbooks and lectures (even if they disagree with them), one need not have a large sample of students to have a sense of which answer is "correct," in the sense of "most likely to be agreed to by all students."

At least two coders are needed to estimate coding quality, since each coder's codes provide a reference for the other's. Knowing how reliably data is being coded, i.e., how much error the coding process is introducing, can contribute significantly to training, coding, and using the dataset. Although various reliability indicators compare the variance between two coders, intercoder agreement indicators (which evaluate exact matches) are preferred when working with nominal data (Tinsley and Brown 2000, Ch. 4). Among intercoder agreement indicators, Cohen's Kappa is the most commonly used indicator for intercoder agreement and identifies beyond chance agreement between two independent codings of nominal data (Cohen 1960; but see also Rothman 2006; Cohen 1968; Krippendorff 2004; Popping 1988; Neuendorf 2002, Ch. 7).

During training, intercoder agreement (IA) statistics shed light on coder understanding of the coding manual, the quality of coding procedures, and the care and consistency of coder application of codes. Although IA statistics generated during training do not reflect data quality (since coding was not done independently), they do help identify whether training has produced sufficient convergence in applying the coding rules that training can end and coding can begin. During coding, calculating IA statistics at regular intervals allows ongoing assessment of both coding quality and coder drift, which, in turn, identifies the need to retrain or replacing coders. Finally, during dataset use, after coding of some agreements is complete, intercoder agreement statistics provide users an indication of data quality and the degree of confidence they should place in the data.

In the IEA project, initial IA statistics for variable-level coding showed coding quality to be good for some variables but not others. Some variables were clearly unreliable (less than 70% agreement), while others were almost entirely reliable (over 90% agreement). Our low intercoder agreement statistics for some codes may reflect their less frequent use, since Cohen's Kappa, being based on the number of times a code is used, weights a single mismatch very heavily against codes that are used infrequently. Table 2 – Intercoder Agreement Indicators by Term shows changes in our IA statistics for all variable-level codes over time. First term IA statistics identified several codes as well below our selected reliability threshold of 70% (such as SBF and REVW). Most studies attempt to achieve high intercoder agreement levels and drop the variables or data with lower intercoder agreement levels (Neuendorf 2002). At the variable-level stage of coding for the IEA Project, since all data has been coded by two coders, we have elected to retain all coding from the variable-level coding stage. Since variable-level coding will be used primarily to retrieve evidence that will be "value-level" coded subsequently, any treaty provision coded by either coder may contain evidence related to a variable and we thus decided to retain both codes in the final dataset despite low reliabilities for some codes.

Regular meetings and retraining

Good dataset creation requires addressing changes over time in the way coder's code. Coders may develop a fuller understanding of a coding manual's rules and improve in their ability to apply those rules (see RESV in Table 2 – Intercoder Agreement Indicators by Term). Coders may even develop a sufficiently good sense of the dataset's underlying concepts that they may code cases according to how they "should" be coded rather than how the coding rules specify to code them. Alternatively, and more commonly, coders may forget specific aspects of coding rules or become less careful in their attention to the coding rules, the available evidence, or both, creating coder drift (see DESC in Table 2 – Intercoder Agreement Indicators by Term).

Regular meetings help avert coder drift before it occurs. The IEA Project held short (1/2 hour) weekly meetings among all coders at which each coder would report on their progress, logistical issues that needed addressing, and their expected progress in the week ahead. These "check-in" meetings proved invaluable for reinforcing and refreshing detailed aspects of coding rules and procedures, for sharing of coding "tricks," for building a sense of joint purpose among coders in a process in which all coding was done independently, and in fostering contact between coders and the project director that are an important part of the "compensation" for most students.

Longer (2 hour) retraining sessions were held every 4 to 6 weeks to review codes which IA statistics showed as most susceptible to coder drift. Retraining was limited to review and clarification of the coding manual, with discussion of the "correct" coding of particular cases and discussions that, explicitly or implicitly, involved changing coding rules being explicitly forbidden.

Documenting, evaluating, and reporting dataset creation process

Much of the value of carefully developing a dataset can be lost if potential users do not have access to information that they can use to evaluate what is in the dataset, how to use the dataset, how much confidence to place in the data and results derived from it, and why and where errors may exist in the dataset. Thus, documentation of all of the elements of dataset creation delineated above is a final, but crucial, step in ensuring dataset usefulness.

Documenting the meanings of variables and values and the data generating process

If the coding rules used in converting evidence into data have been fully and completely documented in a coding manual, then simply making that coding manual available to dataset users is sufficient to allow users to determine what the dataset contains, what it ignores, and how different terms are defined. Beyond this, however, good practice dictates both making it difficult for users to use the dataset without reading the coding manual while also making it difficult to misinterpret the data even if they do not read the coding manual.

Datasets usually identify complex concepts with simple signifiers that can prompt misinterpretation. Thus, one can easily run regressions using indicators of democracy, freedom, or war without knowing how Polity, Freedom House, or the Correlates of War Project how they define the values of those variables or where they place the boundaries between them. Although a dataset user bears ultimate responsibility for reading the coding manual, dataset creator's can encourage that practice by ensuring that data cannot be downloaded without the coding manual, that variables are fully labeled, that footnotes specify the meaning (and especially idiosyncrasies in the meanings) of variables and values, and that published reviews of the dataset identify the best interpretations of variables and values and likely pitfalls of interpretation.

Documenting the coding process is also important, since this is where most error enters a dataset. Making the data generation process transparent is crucial for dataset users to use the data accurately. Transparency about how the population of cases was identified, what cases were selected, how training was conducted, how coding was done, etc. is essential if users are to be able to use the data as intended, ensure it corresponds to the underlying concepts and phenomena that the user (as opposed to the dataset creator) wants to investigate, and identify potential errors and areas for improvement.

Good documentation should serve a purpose similar to that of "truth in advertising." Documentation should provides the most complete and honest description of what the dataset sought to capture and the rules and the processes used to capture it. Users may well disagree with any or all of the dataset creator's judgments about how to define variables, the potential values for each, or training and coding procedures. But only if the dataset creator provides transparent documentation can the user make an informed decision as to whether and how to use the dataset. Transparent documentation also allows other scholars to critique and/or suggest improvements to the dataset and to code additional cases or additional variables.

Documenting data quality

Good documentation also provides evidence on data reliability and data construct validity so that users know how much confidence to place in different aspects of the dataset, can compare similar datasets, and can develop better datasets (Rothman 2006). First, when no alternative source for data on a variable exists, validity and reliability indicators allow users to assess how much confidence they should place in that source. Second, when more than one source for a variable does exist, reliability and validity indicators allow users to compare the quality of, and hence choose between, different sources. Comparing nominally comparable data on a variable requires reading the coding manuals and other documentation and identifying the various differences that may exist in how a variable was defined and coded. If both datasets report common and comparable indicators, a user can use those indicators to compare the quality and content of several datasets. This can be especially useful when common data sources are replicated by more reliable, but obscure, data. Third, publishing reliability and validity indicators allows scholars to develop better quality data. New scholars may find new ways to define or code a variable that can generate more reliable data but only if they can identify those variables that have low reliabilities.

Providing Standard Reliability Estimates

Making reliability indicators available gives users information about how much confidence to place in the dataset generally, which variables and values deserve more confidence and which less, and whether the indicator chosen over- or under-estimates data quality.

Reliability indicators can be presented for the dataset overall, by case, or by variable. The IEA Project experience suggests, however, that overall indicators are of almost no use and by-case indicators are of only limited use. Reporting reliabilities by-case averages the indicators across variables, where some variables may be more reliable than others (Lombard 2002; Neuendorf 2002, 142). In the IEA Project experience, it was easy to calculate intercoder agreement statistics by case (i.e., for each IEA) as soon as two coders had coded it and those statistics were quite high. But intercoder agreement "by case" provides little insight into coding quality. Consider that an average treaty has approximately 50 provisions, each of which -- during variable-level coding -- could receive one or more of 20 possible codes (actually 33 but 20 is used here for heuristic purposes). Now consider a treaty for which 100 codes were applied, with each of the 20 codes being applied 5 times. An intercoder agreement statistic of 80% between two coders' codings of such a treaty could reflect various things: a) that the coders matched all 5 of the

times they applied 16 of the codes but never matched when they applied the other 4 codes, b) that the coders matched 4 of 5 times on all codes, or c) numerous options between these extremes. Assume further, that intercoder agreements statistics were, on average, the same for a dataset of 500 treaties. In the first instance, it suggests extremely successful development of coding rules for 16 variables and complete failure in doing so for 4 variables. That, in turn, suggests that users should have extreme confidence in using the dataset for those 16 variables and should have no confidence at all for the other four. In the second instance, it suggests solid development of coding rules for all 20 variables and strong confidence in the data for all 20 variables (given the 80% agreement levels). Notably, however, "by case" intercoder agreement statistics cannot clarify which interpretation is correct.

Given this, the IEA project believes that intercoder agreement statistics should always be calculated and presented "by variable," not "by case." Indeed, scholars usually use large-N data to evaluate different variables and their effects rather than individual cases, and they are therefore interested in how reliably each variable (rather than each case) has been coded. A major obstacle, however, is that intercoder agreement statistics cannot be calculated until enough cases have been coded to produce a reasonably large number of instances of each code. In our project, some codes were used either once or not at all in each agreement and, therefore, stable intercoder agreement statistics could not be calculated until over 40 agreements had been coded by two coders. Thus, we had to invest significant effort in coding before we had any reliable assessment of coding quality.

Providing Indications of Validity

Good dataset documentation also involves providing information on dataset construct validity. Indicators of construct validity are generally more qualitative than quantitative, but quantitative possibilities exist. Construct validity can be established, in part, by asking a set of scholars to evaluate the extent to which a coding manual's definitions correspond to and capture common understandings of the concepts involved. Describing the process and number of scholars involved and their responses in a coding manual can shed some light on construct validity.

A new dataset can be compared to other datasets capturing nominally similar variables to assess how well the data captures concepts. This type of validity test is also known as convergent and divergent validity and can be presented in a Multitrait-Multimethod matrix (Campbell and Fiske 1959). The matrix presents covariance between different datasets for similar variables. New data should covary the most with itself as an estimate of reliability, second with a dataset that measures a similar concept, then lastly with data that measures a different concept. For example, the IEA project will code the degree to which agreements provide for rewards prior to compliance or sanctions in response to non-compliance. The final codes for this variable should correspond to efforts to measure sanctions and rewards by scholars using similar definitions, but differ from efforts by those using different definitions. Although these matrices usually report covariance (such as Pearson's R), we intend to use the matrix to present indicators of agreement such as Cohen's Kappa for the IEA project. In Table 3, the extent of agreement between instances in which the same measurement technique has been applied at different times or by different individuals to the same case is represented in the diagonal as "Reliability" or "Intercoder agreement." Cells marked "A" represent convergent validity, i.e., the degree of correlation between different datasets measuring the same variable for the same cases (as in the correlation between treaties coded as having sanctions in the IEA dataset and in another dataset). Cells marked "B" represent divergent validity, i.e., the extent to which efforts to measure different concepts actually differ (as in the agreement between treaties coded as having sanctions in the IEA dataset and those coded as establishing scientific committees, or some other concept, in the IEA dataset). The A and B cells establish criterion validity by demonstrating that a particular dataset produces data whose values converge with those of other established datasets for similar variables but diverge from their own values of different concepts. Cells marked C do not present validity coefficients, but instead present a lower bound to possible correlations by correlating different measurement techniques for different concepts.

Table 3 - Multitrait-Multimethod Matrix

		Time 1 (or person 1)			
		Variable 1 Technique 1	Variable 1 Technique 2	Variable 2 Technique 1	Variable 2 Technique 2
Time 2 (or person 2)	Variable 1 Technique 1	Reliability			
	Variable 1 Technique 2	A	Reliability		
	Variable 2 Technique 1	B	C	Reliability	
	Variable 2 Technique 2	C	B	A	Reliability

When the data generated are new and no similar large-N dataset exists, as in the IEA Project's case, divergent and convergent indicators of validity cannot be determined by reference to other datasets. To provide some insight on construct validity, the IEA project intends to compare resulting data with existing case studies on a small number of treaties. Although not fully planned at this point, this may be undertaken by IEA Project personnel comparing our codings to those of published case studies or asking the authors of such case studies to evaluate our codings of the cases according to specific criterion. Although this strategy will be possible on only a small fraction of the IEA's in the Project dataset, it provides at least some evidence relevant to assessing construct validity.

In summary, providing evidence of data quality requires demonstrating both dataset reliability and dataset validity. In all cases, it seems preferable to explicitly examine and report on data quality rather than sidestepping those issues.

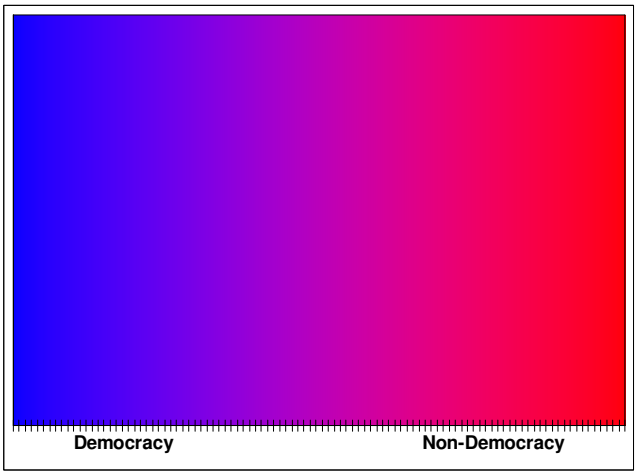
Conclusion

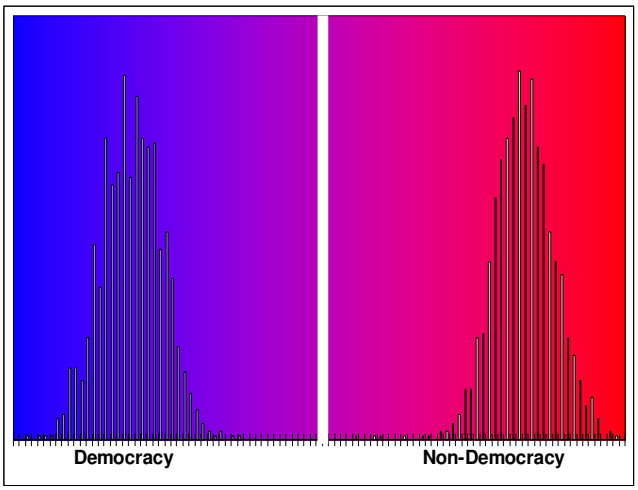
This paper has identified a range of criteria, rules, and procedures that can help produce high quality datasets, using the experience of the IEA Project as both a source of those insights and to illustrate them. The central points of the argument are that dataset creation should:

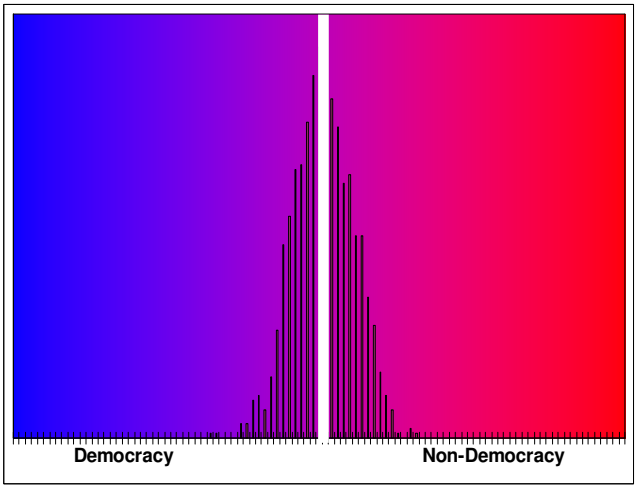
- Start with variables identified from the theoretical literature to ensure that the dataset is developed in ways that will make the dataset itself, as well as analyses of the dataset, relevant to the concerns of other scholars.
- Carefully define the population of relevant cases, identify the members of that population, and specify how the subset to be coded was selected.
- Identify the evidence to be collected for each case and collect it systematically
- Develop a coding manual that has clear, complete, explicit, and well-documented coding rules and procedures, has variables and values that simultaneously reflect theoretically useful categories and empirical variation, and has variables and values that are mutually exclusive and collectively exhaustive.
- Carefully train coders to previously identified standards before allowing any coded data to enter the dataset.
- Conduct coding in ways that mitigate the introduction of error that can be introduced because different people are used as coders or because the same person may code differently over time.
- Document, evaluate, and report dataset creation procedures, paying special attention to providing users with evidence regarding dataset reliability and dataset construct validity.

Creating a dataset with the kind of care recommended here clearly involves a major commitment of time and resources. The primary payoff is in the knowledge that one has created a dataset that, to the greatest extent possible, captures the empirical variation it claims to capture. Such a dataset maximizes the likelihood that analysis using the data can identify both trends in the variables included and underlying relationships between variables, whether within the dataset or from other datasets. In short, good dataset creation provides a crucial building block to good scholarly analysis.

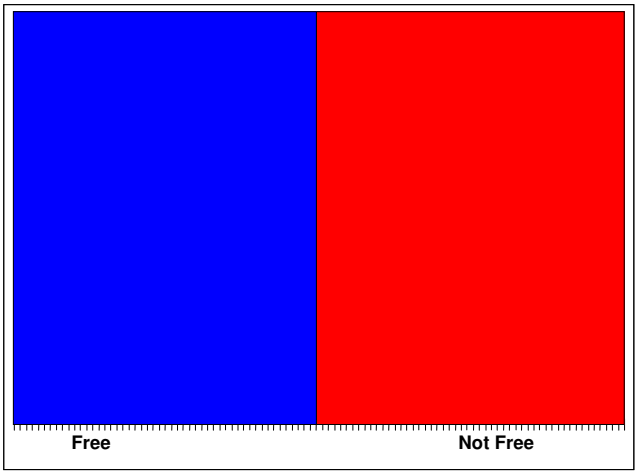
Figures A1, A2, and A3:
Conceptual fuzziness may not inhibit coding if empirical distribution is bimodal but may if it is normal

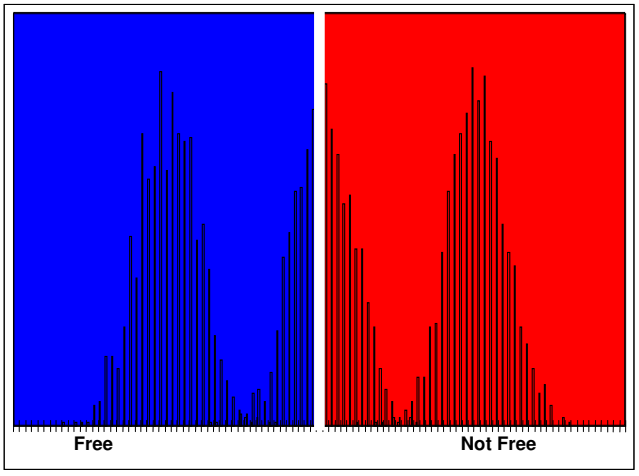


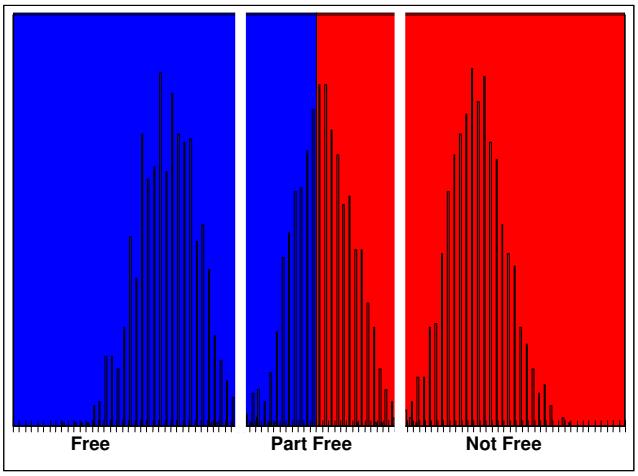




Figures B1, B2, and B3:
Clear conceptual categories may not map well to obvious empirical breaks







References

Aust, Anthony. 2000. *Modern treaty law and practice*. Cambridge, England: Cambridge University Press.

Breitmeier, Helmut, Oran R. Young, and Michael Zürn. 2006. *Analyzing international environmental regimes from case study to database*. Cambridge, MA: MIT Press.

Brown Weiss, Edith, ed. 1997. *International compliance with nonbinding accords*. Washington, DC: American Society of International Law.

Campbell, D. T., and D. W. Fiske. 1959. Convergent and Discriminant Validation by the Multitrait-Multimethod Matrix. *Psychological Bulletin* 56:81-105.

Carmines, Edward G., and Richard A. Zeller. 1979. *Reliability and Validity Assessment*. Beverly Hills: SAGE Publications.

Cohen, J. 1968. Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit. *Psychological Bulletin* 70:213-220.

Cohen, Jacob. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20 (1):37-46.

Dawis, Rene V. 2000. Scale Construction. In *Handbook of Applied Multivariate Statistics and Mathematical Modeling*, edited by H. E. A. Tinsley and S. D. Brown. San Diego: Academic Press. 65-92.

DeVellis, Robert F. 2003. *Scale Development: Theory and Applications*. Thousand Oaks: SAGE Publications.

Feldman, Martha. 2005. Interpretivism, Constructivism, and Related Approaches [Lecture] (Jan. 5). Tempe: Institute for Qualitative Research Methods.

Freedom House. 2006. *Freedom in the World Comparative Rankings: 1973-2005*. Document date: [accessed on 18 May]. Available from http://65.110.85.181/uploads/FIWrank7305.xls.

John, Oliver P, and Veronica Benet-Martinez. 2000. Measurement: Reliability, Construct Validation, and Scale Construction. In *Handbook of Research Methods in Social and personality Psychology*, edited by H. T. Reis and C. M. Judd. New York: Cambridge University Press. 339-369.

Johnson, S. M., and O. D. Bolstad. 1974. Methodological Issues in Naturalistic Observation: Some Problems and Solutions for Field Research. In *Behavior Change: Methodology, Concepts, and Practice*, edited by L. A. Hamerlynck, L. C. Handy and E. J. Mash. Champaign: Research Press.

Krippendorff, Klaus. 2004. *Content analysis: an introduction to its methodology*. Thousand Oaks, Calif.: Sage.

Lombard, M. and J. Snyder-Duch. 2002. Content Analysis in Mass Communication: Assessment and Reporting of Intercoder Reliability. *Human Communication Research* 28:587-604.

Marshall, Monty G., and Keith Jaggers. 2002. *Polity IV Dataset. [Computer file; version p4v2002]*. Document date: [accessed on 1 August 2006]. Available from http://www.cidcm.umd.edu/inscr/polity/polreg.htm.

Mitchell, Ronald B. 2003. International environmental agreements: a survey of their features, formation, and effects. *Annual Review of Environment and Resources* 28:429-461.

Neuendorf, Kimberly A. 2002. *The Content Analysis Guidebook*. Thousand Oaks: SAGE Publications.

Popping, R. 1988. On Agreement Indices for Nominal Data. In *Sociometric Research*, edited by W. E. Saris and I. N. Gallhofer. New York: St. Martin's Press. 90-105.

Rothman, Steven B. 2006. Measurement, Coding, and Reliability in the Quantification of Qualitative Information. Under Review.

Schelling, Thomas C. 1960. *The strategy of conflict*. Cambridge, MA: Harvard University Press.

Tinsley, H. E. A., and S. D. Brown. 2000. *Handbook of Applied Multivariate Statistics and Mathematical Modeling*. San Diego: Academic Press.