How to Deal with Missing Data and Galton's Problem in Cross-Cultural Survey Research: A Primer for R

E. Anthon Eff* Malcolm Dow†

An Article Submitted to

Structure and Dynamics: eJournal of Anthropological and Related Sciences

Manuscript 1090

*Middle Tennessee State University, eaeff@mtsu.edu

†Northwestern University, mmd383@northwestern.edu

Abstract

Multiple imputation (MI) has become the preferred method for dealing with missing data in survey research. MI involves three steps: creating m multiply imputed complete datasets; estimating models on each of the m datasets using any standard statistical procedure; combining the resulting multiple estimates of each statistic of interest. This paper provides R programs for MI, and offers some advice for employing MI with data drawn from the Standard Cross-Cultural Sample (SCCS). A second set of R programs combines estimates from the m imputed data sets, and also deals with the problem of network autocorrelation effects, i.e., Galton's Problem or the non-independence of cases, using two-stage instrumental variables (IV) regression. The objective of the paper is to provide programs, advice and explanations that will help researchers employing cross-cultural survey data, especially the SCCS, to deal with the twin problems of missing data and network autocorrelation effects, using the open source statistical package R. The paper is intended to complement a recent suite of publications by Dow and Eff where both theoretical and empirical issues underlying these two problems are discussed in detail.

Keywords: multiple imputation; R; Standard Cross-Cultural Sample; Galton's problem; two-stage OLS regression

Suggested Citation:
E. Anthon Eff and Malcolm Dow (2008) "How to Deal with Missing Data and Galton's Problem in Cross-Cultural Survey Research: A Primer for R", *Structure and Dynamics: eJournal of Anthropological and Related Sciences*: Vol. 3: No. 2, Article 1.
http://repositories.cdlib.org/imbs/socdyn/sdeas/vol3/iss2/1

Introduction

Missing data is a serious problem in cross-cultural survey research, especially for regression models employing data sets such as the Ethnographic Atlas (EA) or the Standard Cross-Cultural Sample (SCCS).[1] In comparative survey research the most common procedure for handling missing data is *listwise deletion*, where one simply drops any observation containing a missing value for any of the model's variables. Listwise deletion leads to the loss of all non-missing information in the dropped rows, frequently leading to statistical analysis being based on subsamples that are no longer representative of the full sample. Estimates based on such subsamples are valid only if certain assumptions are made about the mechanism(s) by which the data become missing. These assumptions are reviewed by Dow and Eff (2009b) who conclude that they will seldom hold for cross-cultural data sets.

A superior alternative to listwise deletion that is rapidly gaining favor in the social sciences is *multiple imputation*. Here, values are *imputed*, or estimated, for the missing observations, using auxiliary data that covaries with the variable containing missing values. The qualifier *multiple* signifies that multiple data sets (typically 5 to 10) are imputed, within each of which missing values are replaced with imputed values drawn from a conditional distribution (conditional on the values of the auxiliary data). The imputed values will be different in each of the data sets: only slightly different when the variable with missing values is strongly conditioned by the auxiliary data; quite different when the variable is only weakly conditioned by the auxiliary data. Standard statistical estimation procedures are carried out on each of the *m* imputed data sets, leading to *m* estimates for parameters of interest which are subsequently combined to produce final estimates of model parameters.

A few recent cross-cultural papers point out the advantages of multiple imputation. Dow and Eff (2009b) provide a review of the issues and literature. The methods have also been used in two recent empirical studies (Dow and Eff 2009a; Eff and Dow 2009).

A second basic problem with cross-cultural data sets is that the sample cases are frequently not independent of one another due to various types of inter-societal network processes: copying, borrowing, contagion, conquest, trade, inheritance from ancestral populations, etc. This is the classic Galton's Problem in anthropology, understood more generally as the problem of cultural trait transmission. Unless the relevant networks of inter-relations are somehow included in the standard statistical modeling procedures, inconsistent and biased estimates will be generated. A new approach to Galton's Problem based on instrumental variables regression is outlined in Dow (2007, 2008), and empirical examples are reported in the Dow and Eff papers cited above. A second objective of the current paper, then, is to provide R programs that will also enable

[1] The EA is described in Murdock (1967); the SCCS is described in Murdock and White (1969) and in White (2007).

researchers to implement the new network autocorrelation effects regression approach to Galton's Problem.

The two programs presented here can be easily modified to build any OLS model utilizing SCCS data. It takes many hours of experience before one becomes proficient in writing R programs but the simplest way to begin is to copy and modify programs written by others.

Creating Multiply Imputed Data

Multiple imputation requires the following three steps. First, multiple (5 to 10) versions of the data are created using auxiliary data to estimate values where these are missing. Next, each of the imputed data sets is analyzed using whichever classical statistical procedure is required (typically a multivariate regression), and the estimated parameters stored. Finally, the multiple estimated results are combined using formulas first developed by Donald Rubin (1987). We will explain the mechanics of each of these steps in detail with respect to the SCCS data set. Figure 1 provides an overview of our procedure.

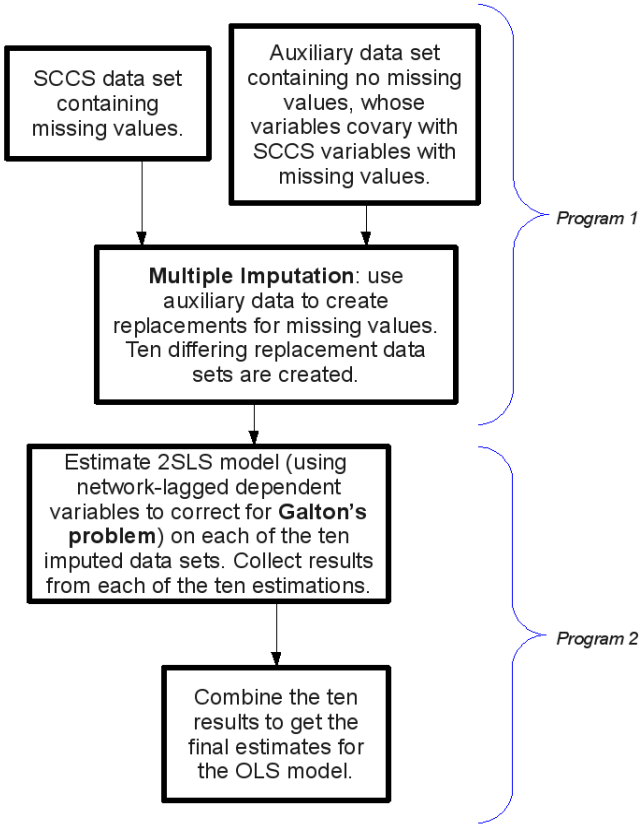

Figure 1: The flowchart provides an overview of the procedures described in this primer.

Two widely used R packages will create MI data: *mix* (Schafer 2007) and *mice* (van Buuren and Groothuis-Oudshoorn 2009). In this primer, we will use *mice* (MI by Chained Equations.)

Auxiliary data

The *mice* procedure will estimate values for missing observations using auxiliary data provided by the user. The ideal auxiliary data for the SCCS are those SCCS variables with no missing values. Imputation is a regression-based procedure, where the variable to be imputed is the dependent variable, and the auxiliary data are the independent variables.[2] This leads to several important constraints in choosing auxiliary variables. First, the procedure will succeed only if there are fewer auxiliary variables than the number of non-missing observations on the variable being imputed (so that the degrees of freedom are greater than one). Second, since most SCCS variables are scales with few discrete values, it is easy to make the mistake of choosing an auxiliary variable which—over the set of non-missing observations—is perfectly collinear with some of the other auxiliary variables, causing the MI procedure to fail. Third, auxiliary variables which create an extremely good in-sample fit with the variable to be imputed (the "imputand") might have a very poor out-of-sample fit, so that the imputed values are almost worthless. This last is the problem of over-fitting a model.

One step that reduces the problem of over-fitting is to discard all potential auxiliary variables that could not have a plausible relationship with the imputand. These would include the many SCCS variables that describe characteristics of the ethnographic study (such as date of the fieldwork or sex of the ethnographer), or that represent reliability assessments of other variables. While these variables have no missing values and may provide a good fit to the non-missing values of the imputand, that fit is entirely spurious, and one has no reason to believe that the fit would also extend to the missing values of the imputand.

Use of Principal Components as Auxiliary Variables for Imputation

Another step that diminishes the risk of over-fitting is to use the same set of auxiliary variables for all imputations, rather than selecting a unique best-fitting set for each imputand. This requires that a small number of the highest quality variables be selected. A reasonable way to do this is to use principal components analysis over a large set of variables, and select the few largest principal components as auxiliary variables. A second advantage of principal components is that each observation typically has a unique numeric value, making perfect collinearity among the auxiliary variables all but impossible.

Table 1 shows 105 SCCS variables that may be used to generate principal components. The column "category" shows the group with which each variable is

[2] The three default estimation methods used in *mice* are: 1) for a numeric variable, predictive mean matching (a semi-parametric regression method); 2) for a binary variable, logistic regression; and 3) for a nominal variable, polytomous logistic regression. Six optional methods are also available (van Buuren and Groothuis-Oudshoorn 2009).

classified when producing principal components; there are five groups. The column "nominal=1" shows whether a variable is a nominal variable (as opposed to ordinal); nominal variables are first converted to dummy variables (one for each discrete category) before calculating principal components. The variables are from the SCCS, with the exception of 20 climate variables (Hijmans et al 2005) and a variable for net primary production (Imhoff et al 2004). Values for these variables were assigned to each SCCS society using a Geographical Information System (GIS) to extract data for the location of each society. The utility of any of these categories and their associated principal components as auxiliary variables will vary with the nature of the substantive model and the variables to be imputed.

Table 1: Auxiliary variables used to create principal components

SCCS variable	category	Description	# discrete values	nominal=1
v61	Complexity	Fixity of Settlement	6	1
v62	Complexity	Compactness of Settlement	4	1
v65	Complexity	Types of Dwelling	14	1
v66	Complexity	Large or Impressive Structures	6	1
v73	Complexity	Community Integration	7	1
v74	Complexity	Prominent Community Ceremonials	4	1
v75	Complexity	Ceremonial Elements	6	1
v76	Complexity	Community Leadership	8	1
v149	Complexity	Writing and Records	5	0
v150	Complexity	Fixity of Residence	5	0
v151	Complexity	Agriculture	5	0
v152	Complexity	Urbanization	5	0
v153	Complexity	Technological Specialization	5	0
v154	Complexity	Land Transport	5	0
v155	Complexity	Money	5	0
v156	Complexity	Density of Population	5	0
v157	Complexity	Political Integration	5	0
v158	Complexity	Social Stratification	5	0
v234	Complexity	Settlement Patterns	8	1
v236	Complexity	Jurisdictional Hierarchy of Local Community	3	1
v270	Complexity	Class Stratification	5	1
v271	Complexity	Class Stratification, Secondary Feature	5	1
v920	Complexity	Large scale slaveholding systems: proportion of slaves	6	0
v1130	Complexity	Population Density	6	0
v158_1	Complexity	Sum of Cultural Complexity (v149-158)	40	0
v854	Ecology	Niche Temperature (Approximate) Adapted from William Goode, World Atlas	8	0
v855	Ecology	Niche Rainfall (Approximate) Adapted from William Goode, World Atlas	7	0
v856	Ecology	Niches Adapted from William Goode, World Atlas	15	0
v857	Ecology	Climate Type- Ordered in terms of Open Access to Rich Ecological Resources	6	0
v921	Ecology	Agricultural Potential 1: Sum of Land Slope, Soils, Climate Scales	18	0

SCCS variable	category	Description	# discrete values	nominal=1
v922	Ecology	Land Slope	5	0
v924	Ecology	Suitability of Soil for Agriculture	8	0
v926	Ecology	Climate	7	0
v928	Ecology	Agricultural Potential 2: Lowest of Land Slopes, Soils, Climate Scales	8	0
v1253	Ecology	Leishmanias	3	0
v1254	Ecology	Trypanosomes	3	0
v1255	Ecology	Malaria	3	0
v1256	Ecology	Schistosomes	3	0
v1257	Ecology	Filariae	3	0
v1258	Ecology	Spirochetes	3	0
v1259	Ecology	Leprosy	3	0
v1260	Ecology	Total Pathogen Stress	15	0
v1696	Ecology	Biome	5	1
v1913	Ecology	Mean yearly annual rainfall	185	0
v1914	Ecology	Coefficient of variation in mean annual rainfall	185	0
v1915	Ecology	Lowest yearly rainfall in the n years sampled	182	0
v1916	Ecology	Highest yearly rainfall in the n years sampled	183	0
v1917	Ecology	Difference between maxrain and minrain rainfall	183	0
bio1	Ecology	Annual Mean Temperature	186	0
bio2	Ecology	Mean Diurnal Range (Mean of monthly (max temp - min temp))	186	0
bio3	Ecology	Isothermality (P2/P7) (* 100)	186	0
bio4	Ecology	Temperature Seasonality (standard deviation *100)	186	0
bio5	Ecology	Max Temperature of Warmest Month	186	0
bio6	Ecology	Min Temperature of Coldest Month	186	0
bio7	Ecology	Temperature Annual Range (P5-P6)	186	0
bio8	Ecology	Mean Temperature of Wettest Quarter	186	0
bio9	Ecology	Mean Temperature of Driest Quarter	186	0
bio10	Ecology	Mean Temperature of Warmest Quarter	186	0
bio11	Ecology	Mean Temperature of Coldest Quarter	186	0
bio12	Ecology	Annual Precipitation	186	0
bio13	Ecology	Precipitation of Wettest Month	186	0
bio14	Ecology	Precipitation of Driest Month	171	0
bio15	Ecology	Precipitation Seasonality (Coefficient of Variation)	186	0
bio16	Ecology	Precipitation of Wettest Quarter	186	0
bio17	Ecology	Precipitation of Driest Quarter	179	0
bio18	Ecology	Precipitation of Warmest Quarter	184	0
bio19	Ecology	Precipitation of Coldest Quarter	184	0
bio20	Ecology	Altitude	186	0
NPP	Ecology	Biomass: net primary production	186	0
v67	Family&Kinship	Household Form	8	1
v68	Family&Kinship	Form of Family	12	1
v70	Family&Kinship	Descent-Membership in Corporate Kinship	5	1

SCCS variable	category	Description	# discrete values	nominal=1
		Groups		
v79	Family&Kinship	Polygamy	4	1
v80	Family&Kinship	Family size	5	1
v208	Family&Kinship	Mode of Marriage	7	1
v209	Family&Kinship	Mode of Marriage (Alternate)	6	1
v836	Family&Kinship	Rule of Descent: Primary	8	1
v1858	Region&Religion	Region	10	1
relig	Region&Religion	Religion	9	1
v3	Subsistence	Agriculture-Contribution to Local Food Supply	6	0
v5	Subsistence	Animal Husbandry-Contribution to Food Supply	6	0
v19	Subsistence	Preservation and Storage of Food	12	0
v21	Subsistence	Food Surplus Via Storage	3	0
v22	Subsistence	Food Supply (Ecological or Distribution Network)	5	0
v203	Subsistence	Dependence on Gathering	8	0
v204	Subsistence	Dependence on Hunting	10	0
v205	Subsistence	Dependence on Fishing	10	0
v206	Subsistence	Dependence on Animal Husbandry	10	0
v207	Subsistence	Dependence on Agriculture	10	0
v232	Subsistence	Intensity of Cultivation	6	0
v233	Subsistence	Major Crop Type	4	1
v243	Subsistence	Animals and Plow Cultivation	3	1
v244	Subsistence	Predominant Type of Animal Husbandry	7	1
v245	Subsistence	Milking of Domestic Animals	2	0
v246	Subsistence	Subsistence Economy	7	1
v814	Subsistence	Importance of Agriculture	17	0
v815	Subsistence	Importance Domes. Anim	13	0
v816	Subsistence	Importance Fishing	15	0
v817	Subsistence	Importance Hunting	16	0
v818	Subsistence	Importance Gathering	13	0
v819	Subsistence	Importance Trade	8	0
v820	Subsistence	Principal Subsistence Category	8	1
v833	Subsistence	Subsistence Economy: Dominant Mode	8	1
v834	Subsistence	Subsistence Economy: Subsidiary Mode	7	1
v858	Subsistence	Subsistence Type- Ecological Classification	11	1

Notes: N=186 for all variables. All variables from the SCCS, except bio1-bio20 (GIS data from Hijmans et al 2005) and NPP (GIS data from Imhoff et al 2004). All variables numeric and ordinal, except those for which nominal=1.

Figure 2 shows the percent of the total variation explained for each component in the five sets of principal components. In addition to charts for each of the five major variable groups, Figure 2 shows a chart of principal components extracted from a proximity matrix based on language phylogenetic relationships among the SCCS societies (Eff 2008). The dotted red line in each chart shows the cut-off between components retained and components discarded.

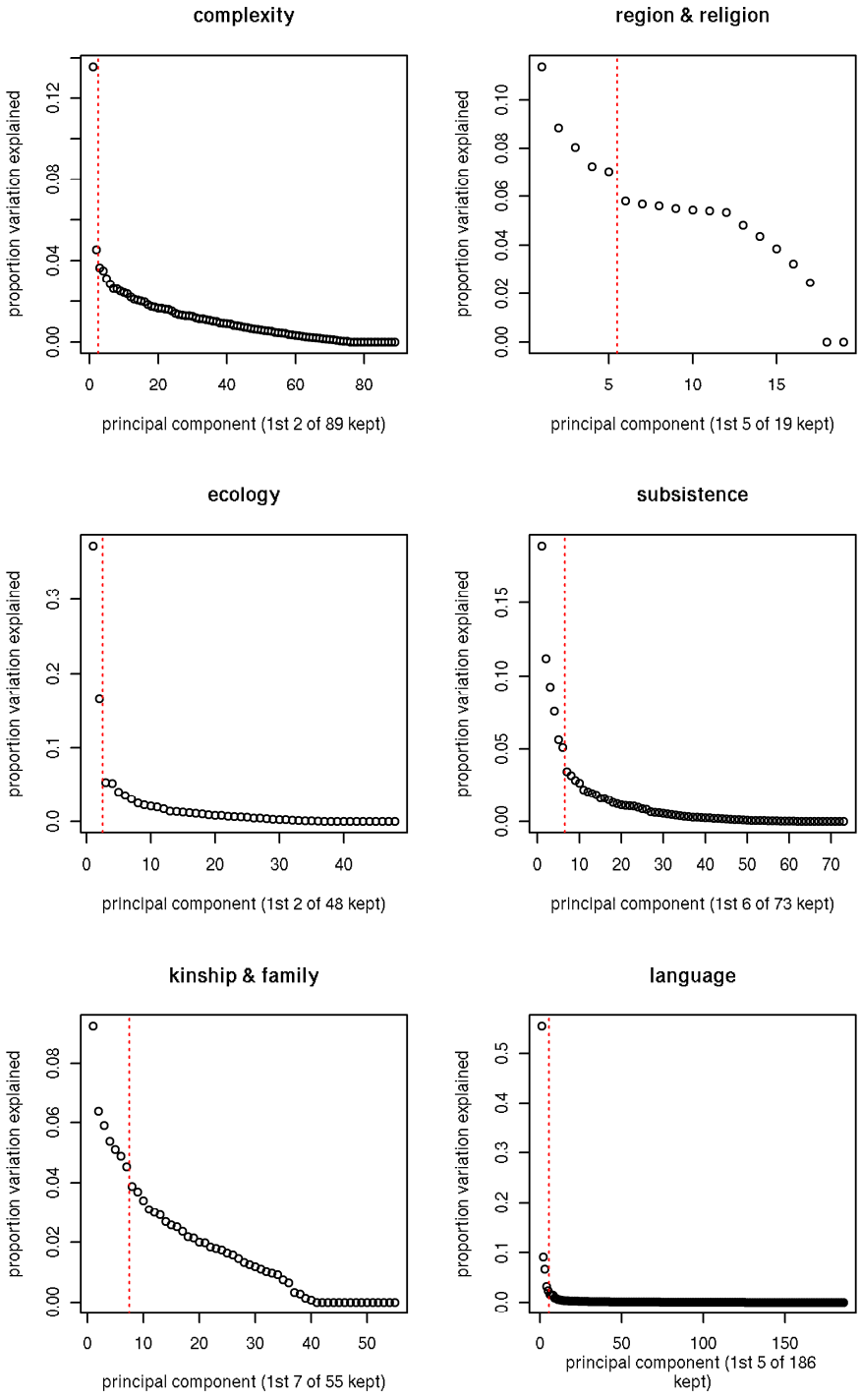

Figure 2: The proportion of variation explained for principal components. The dotted red line marks the principal components retained for the auxiliary data. The principal components for *region and religion* were not used—instead, a collapsed set of dummy variables are used in the auxiliary data.

The *Region and Religion* principal components of the combined 10 region and 9 religion dummy variables show little decay in proportion of variation explained. It therefore seems better to use dummy variables for regions and religion, perhaps after collapsing categories as we do below, rather than principal components.

Since *Galton's problem* is of such overwhelming importance in cross-cultural studies, and is often reflected in spatial or linguistic clustering of variable values, the auxiliary variables should also contain measures that capture some of the ways in which societies are connected across space or through time. The principal components of the language matrix are included for this reason, and we also include latitude and longitude (converted to radians), their squares and their cross-products.

Program 1: Multiply imputing the data set

Before beginning, the user should create a directory in which to unzip the zip folder containing the data sets and programs.[3] The first of the two programs in the appendix creates multiply imputed versions of a given dataset. The program contains comments (on the lines beginning with "#"), but each step will be briefly discussed here, as well.

The program begins by defining the working directory, where the data sets and programs are kept. Next, all objects are removed from memory, followed by an option that the commands issued in the script file be echoed in the output file (if running in batch mode) or in the console (if using the MS Windows GUI). R consists of the "base" procedures plus nearly 2,000 "packages" contributed by users, containing specialized procedures. The packages are made available through the *library* command. Here we will use two packages: *foreign*, which allows us to read and write data in many different formats; and *mice*, which will create multiple imputed datasets. The package *foreign* is part of base R, but *mice* must be "installed" before it can be "loaded" into the program. Installing is most easily done using the menu bar at the top of the MS Windows GUI.

Our imputation program calls in two external R-format datasets: *vaux.Rdata* and *SCCS.Rdata*. The first of these contains the auxiliary data, and the second is an R version of the SPSS-format SCCS data found on Douglas White's website[4] as of March, 2009. As always, it's a good idea to look at the data at hand before using them. Once an object is loaded, e.g., such as with `load("vaux.Rdata".GlobalEnv)`, useful commands in R for looking at an object called *vaux* are: `class(vaux)` (tells what class of object *vaux* is—the datasets should be of class "dataframe"), `names(vaux)` (lists the variable names), `summary(vaux)` (quintiles and mean of each numeric variable; plus frequency of first six categories for character variables), `head(vaux)` (prints first six rows of *vaux*), and `tail(vaux)` (prints last six rows of *vaux*).

---

[3] The zip folder is found at http://intersci.ss.uci.edu/wiki/spw/Eff&Dow_data&programs.zip and in supplementary files that accompany this article.

[4] Douglas White's website: http://eclectic.anthrosciences.org/~drwhite/courses/index.html. Additional items related to the methods described here may be found on the InterSci wiki at: http://intersci.ss.uci.edu/wiki/index.php/Imputing_the_data

The auxiliary data *vaux* contains the character variable *socname* (character variables are called "factors" in R; factors are interpreted as nominal variables, even when the characters are numbers). Factors can be used as auxiliary variables during the imputation process—they are converted into zero-one dummy variables, one for each discrete value, as for example, a zero-one variable for each society name. The number of discrete values should be few, however (to avoid the problem of perfect collinearity). The variable *socname* has 186 discrete values; it therefore must be removed from *vaux*, which is accomplished by the negative sign within the brackets (28 is *socname*'s column position in *vaux*). Before removing *socname* it is compared with the values of *socname* in the SCCS data set, to ensure that the rows are ordered identically in the two files. Two factors are retained in *vaux*: one for a collapsed set of Burton regions (v1858), and the other for world religions (v2002.) Both of these are described in the comments.

A block of commands loops through each of the variables in *vaux*, checks if the variable is numeric, finds the number of missing values, finds the number of discrete values, and then lists the results for all variables in a table. This step is not really necessary, but it's useful to know these facts about the auxiliary data.

Variables from the SCCS are extracted into a new dataframe *fx*. In this step some variables are modified and most are given new names. Table 2 summarizes the variables selected for dataframe *fx* (see White et al. 2009 for the codebook). Note that zero-one dummy variables are created by creating a conditional statement (a statement that is either true or false) and then multiplying that statement by one, to make it numeric (true statements equal one, false statements equal zero).

Table 2: Variables in imputed dataset

Variable name	SCCS name	Nominal=1	# missing	N	# discrete
socname	SCCS$socname	1	0	186	186
socID	SCCS$"sccs#"	0	0	186	186
valchild	(SCCS$v473+SCCS$v474+ SCCS$v475+SCCS$v476)	0	15	171	18
cultints	SCCS$v232	0	0	186	6
roots	(SCCS$v233==5)*1	0	0	186	2
cereals	(SCCS$v233==6)*1	0	0	186	2
gath	SCCS$v203	0	0	186	8
hunt	SCCS$v204	0	0	186	10
fish	SCCS$v205	0	0	186	10
anim	SCCS$v206	0	0	186	10
femsubs	SCCS$v890	0	1	185	9
pigs	(SCCS$v244==2)*1	0	0	186	2
milk	(SCCS$v245>1)*1	0	0	186	2
plow	(SCCS$v243>1)*1	0	0	186	2
bovines	(SCCS$v244==7)*1	0	0	186	2
tree	(SCCS$v233==4)*1	0	0	186	2
foodtrade	SCCS$v819	0	0	186	8
foodscarc	SCCS$v1685	0	42	144	5
ecorich	SCCS$v857	0	0	186	6
popdens	SCCS$v156	0	0	186	5
pathstress	SCCS$v1260	0	0	186	15
CVrain	SCCS$v1914/SCCS$v1913	0	0	186	185

Variable name	SCCS name	Nominal=1	# missing	N	# discrete
rain	SCCS$v854	0	0	186	8
temp	SCCS$v855	0	0	186	7
AP1	SCCS$v921	0	0	186	18
AP2	SCCS$v928	0	0	186	8
ndrymonth	SCCS$v196	0	4	182	13
exogamy	SCCS$v72	0	1	185	5
ncmallow	SCCS$v227	0	12	174	8
famsize	SCCS$v80	0	0	186	5
settype	SCCS$v234	0	0	186	8
localjh	(SCCS$v236-1)	0	0	186	3
superjh	SCCS$v237	0	2	184	5
moralgods	SCCS$v238	0	18	168	4
fempower	SCCS$v663	0	53	133	7
sexratio	1+(SCCS$v1689>85)+ (SCCS$v1689>115)	0	127	59	3
war	SCCS$v1648	0	26	160	18
himilexp	(SCCS$v899==1)*1	0	19	167	2
money	SCCS$v155	0	0	186	5
wagelabor	SCCS$v1732)	0	97	89	3
migr	(SCCS$v677==2)*1	0	81	105	2
brideprice	(SCCS$v208==1)*1	0	0	186	2
nuclearfam	(SCCS$v210<=3)*1	0	1	185	2
pctFemPolyg	SCCS$v872	0	41	145	54

Notes: All variables created from SCCS variables, as shown in Program 1, at the creation of dataframe *fx*. The SCCS code book (White et al. 2009) is found at: http://eclectic.ss.uci.edu/~drwhite/courses/SCCCodes.htm.

We next identify variables in dataframe *fx* that have missing values, and make a list of their names (*zv1*). We loop through these variables, at each iteration attaching a new variable with missing values to the auxiliary data, and saving the final data set in a temporary dataframe called *zxx*. The procedure *mice* makes imputed replications of this imputand; the non-missing values of the imputand will be the same in each replication, but the missing values will be replaced with imputed values that will differ somewhat across the 10 replications. These values are stored in another dataframe called *impdat*, which contains the imputed variables, as well as two new variables: *.imp* (an index for imputation, *.imp*=1,... 10); and *.id* (an index for society, *.id*=1,… 186). The dataframe now has one column for each imputed variable and 1,860 rows; it is sorted such that the 186 societies for imputation one are stacked on top of the 186 societies for imputation two, which are stacked on top of the 186 societies for imputation three, and so on.

Finally, those variables in the dataframe *fx* which have no missing values are attached to the dataframe *impdat* in their numeric locations for the SCCS sample (1 through 186.) This requires that 10 replications of these variables be stacked on top of each other, to create 1,860 rows, and then attached to the imputed data. The dataframe *impdat* is then saved as a permanent R-format data file, for use later.

Modifying Program 1: Imputing other survey data sets

Modifying this program to generate imputed data for one's own research would typically require changing only two sections. First, the *setwd* command at the top of the program must be changed to the directory where the two data files are stored. Second, variables and names in the command creating the dataframe *fx* should be changed to include the variables relevant for one's own research. In addition, one might occasionally encounter a situation where *mice* execution fails due to perfect collinearity (the error message will report this as a problem of "singularity"). This problem can, in most cases, be fixed by dropping the two factors (*brg* and *rlg*) from the auxiliary data. Replace `vaux<-vaux[,-28]` with `vaux<-vaux[,c(-28,-29,-30)]` to drop the two factors.

Combining estimates generated from multiply imputed data sets

Estimations are performed on each of the multiply imputed data sets, singly, and then the results are combined, using formulas presented in Rubin (1987: 76-77).[5] The estimations can be of any kind: contingency tables, OLS regression coefficients, model diagnostics such as R^2, logit marginal effects, discriminant analysis and so on. In Program 2, we present an example of an OLS regression model, in which we estimate a model with the dependent variable *depvar*— the degree to which a society values children, defined as a sum of values for early and late boys and girls.. This model has no serious theoretical basis for the independent variables; it is presented merely to illustrate how regression results are combined. This particular example is also useful to illustrate how the program corrects for Galton's problem of non-independence of cases using instrumental variables regression.

Program 2: Combining estimates from m imputed data sets

As was the case with the previous program, one must change the working directory to that directory where one's data and programs are saved. All of the packages loaded with the *library* command must first be installed, with the exception of *foreign*, which is part of base R.

While opinions differ, some statisticians advise selecting only those observations for which the dependent variable of a regression model is non-missing when combining results from the imputed data sets (von Hipple 2007). We load the SCCS data, and find those observations for which *depvar* is non-missing, and pass those observation numbers to the object *zdv*. We will use *zdv* at several places in the program to limit the observations to be analyzed to this subset.

The dataframe *impdat*, containing 10 imputed data sets is read. The command *summary()* is used to take a cursory look at the variables, and the command *hist()* allows one to see how the dependent variable is distributed.

[5] The formulas are also given in Dow and Eff (2009a: 140-141) and Eff and Dow (2008: 10-12).

Galton's problem and Instrumental Variable Regression

Galton's problem must always be considered in cross-cultural research. For our example in this paper we introduce two weight matrices (**W**), each of which will be used for three purposes: 1) to create network-lagged dependent variables; 2) to create instrumental variables for the network-lagged dependent variables; and 3) to test for any additional network autocorrelation in the residuals after the network-lagged instrumental variable has been included in the regression estimation. The first weight matrix represents proximity between cultures based on language phylogeny (Eff 2008) and the second matrix represents proximity between cultures based on great circle distance.[6] The diagonal of each matrix is set to zero, and only those rows and columns corresponding to societies for which the dependent variable is missing are retained (using *zdv*). Each cell is then divided by the row sum, so that each row sums to one. The objects *dd* and *ll* are matrices. We also create *wmatdd* and *wmatll*, which are weight matrix objects, for use in procedures that will conduct autocorrelation tests.

The list *indpv* contains the names of all potential independent variables derived from *impdat*. We will use this list when we create network-lagged dependent variables, along with the matrices *dd* and *ll*.

Program 2 executes a loop 10 times, each time selecting a different imputed dataset from *impdat*. Using those data, a two-stage OLS regression is run, coefficients and their variances are collected, and some model diagnostics are estimated and collected. The results are collected by appending them to the four NULL objects (*VIF*, *ss*, *beta*, *dng*) listed immediately before the loop. At the completion of the loop, each object will have 10 rows, one corresponding to each imputed dataset.

Within the loop, the first step selects a particular imputed dataset, and then retains only those observations for which the dependent variable is non-missing. These observations are the rows of a dataframe called *m9*. The first stage of the two-stage OLS regression consists of computing instrumental variables, the second of using the IV variables in the final regression (Wooldridge 2006: Chapter 15).

Network-lagged variables as instrumental variables (IVs)

Next, network-lagged dependent variables are created for use in the instrumental variables regression. The object *cyd* is the dependent variable (y) pre-multiplied by the weight matrix for distance *dd* (i.e., $\mathbf{W}y$). Since the weight matrix is row standardized to unity and all diagonal entries are set to zero, each observation in *cyd* will be a weighted mean of the dependent variable values in neighboring societies, with the closest societies having the highest weights. But since *cyd* is endogenous when entered as an independent variable in a regression model - that is, it is correlated with the error term in the regression equation, since it is a function of y - it must be replaced by an appropriate "instrument". If not, the regression coefficient estimates for all variables in the regression

[6] Only the 25 nearest neighbors for each society have a non-zero weight. This weight matrix is described in Dow and Eff (2008: 152).

model will be biased. We proceed by regressing *cyd* on the matrix of network-lagged independent variables **WX**, and then using as instrument the fitted value of *cyd* from the regression (*fydd*), following the procedures described in Dow (2007). [7]

Additional endogenous network variables can be added to a model, subject to limitations of sample size and possible collinearity problems. The most general network autocorrelation effects regression model is thus (Dow 2007:347):

$$y = \rho_1 \mathbf{W}_1 y + \rho_2 \mathbf{W}_2 y + \ldots + \rho_t \mathbf{W}_t y + \mathbf{X}\beta + \varepsilon$$

where the usual assumptions on the error term apply. Clearly, since y is a function of ε, each of the $\mathbf{W_i}y$ variables is also a function of ε, is thus endogenous and must be replaced by an instrumental variable. In the program we create two instrumental variables for network-lagged dependent variables: one using the language matrix (*fyll*); the other using the distance matrix (*fydd*).

Since **W** is *nxn*, and *y* is *nx1*, **W***y* is *nx1*. One thus regresses **W***y* on a *nxj* matrix of exogenous variables **Z**:

$$(\mathbf{W}y)_i = \alpha_0 + \sum_j \alpha_j z_{ji} + \mu_i$$

which gives estimated coefficients $\hat{\alpha}_j$ that can be used to create a fitted value for **W***y*:

$$\hat{(\mathbf{W}y)}_i = \hat{\alpha}_0 + \sum_j \hat{\alpha}_j z_{ji}$$

It is this fitted value that is our instrumental variable. We collect not only the fitted value, but also the R^2 of these regressions, in order to get a sense of how well our instrument fits the original network-lagged dependent variable.

Two sets of second-stage OLS regressions are performed. The first is an unrestricted model, and the second is a restricted model, containing only those independent variables coefficients that are significant (p-values ≤ .05). The results of the restricted model are collected: the variance inflation factors (VIF); the diagonal of the variance-covariance matrix for the estimated coefficients; and the estimated coefficient values. Since these regressions are each the second stage of a two-stage least-squares (the regressions creating instruments are the first), the variance-covariance matrix and R^2 must be corrected (Dow 2007:348); the corrections are performed in the block of commands headed with the comment "*corrected sigma2 and R2 for 2SLS*". When heteroskedasticity is present, White's robust variance-covariance matrix should be used (Wooldridge 2006:274), as shown in the program comments.

The list *dropt* contains the names of independent variables dropped from the unrestricted model to generate the restricted model. This list is used to perform a Wald

[7] Wooldridge (2006), chapter 15, provides a general background to instrumental variables and two-stage least squares.

test (Davidson and MacKinnon 2004:330) on the restrictions (H_0: that the true values of these coefficients equal zero). Other diagnostics collected here are: the model R^2; the Ramsey RESET test for omitted non-linear transformations of the independent variables (Wooldridge 2006:308-309); a Lagrange multiplier test for heteroskedasticity (Wooldridge 2006:279-281); the Shapiro-Wilk test for normality (Shapiro and Wilk 1965), applied to model residuals; and a Lagrange multiplier to test for additional network dependence (network lag) in the residuals using the two weight matrices (language, distance) separately (Anselin et al. 1996). Other diagnostics could easily be added. Note that—with the exception of R^2—all are statistics with a distribution, and the only figure collected is the value of the chi-squared statistic. When the original statistic is distributed in some way other than chi-squared, the p-value is used to find the appropriate chi-square statistic with one degree of freedom.

When the loop terminates, each of the four objects has 10 rows, one corresponding to each set of estimates. The first block of commands uses Rubin's formulas to combine the regression coefficients and their variances. The final value of each coefficient is simply its mean; the final value of each variance is a function both of the mean of the 10 variance values and of the variance of the 10 estimated coefficient values. The degrees of freedom are a function of the number of imputations and of the variation among the estimates, but *not* a function of the degrees of freedom in the original 10 estimated models.

Statistics without hypothesis tests, such as the VIFs and the three R^2 measures, are simply averaged to find the final value, just as the regression coefficients were. The chi-square diagnostics collected during estimation are combined in the next block of commands, using Rubin's (1987) formulas appropriate for these statistics. All final results are now contained in three objects: *bbb* (coefficients with p-values and VIFs); *ccc* (diagnostics with p-values); and *r2* (the R^2 for the final model and each of the models creating the instrumental variables). The last block of commands writes these three objects to a file called *OLSresults.csv*. With modifications, OLSresults.csv can be turned into a publication quality table.

Modifying Program 2

Users modifying this file for their own work would need to change the working directory, and then change the variable names to those appropriate for their model. The program would typically be run several times, as one develops the final model. Figure 3 below gives an overview of the model development process.

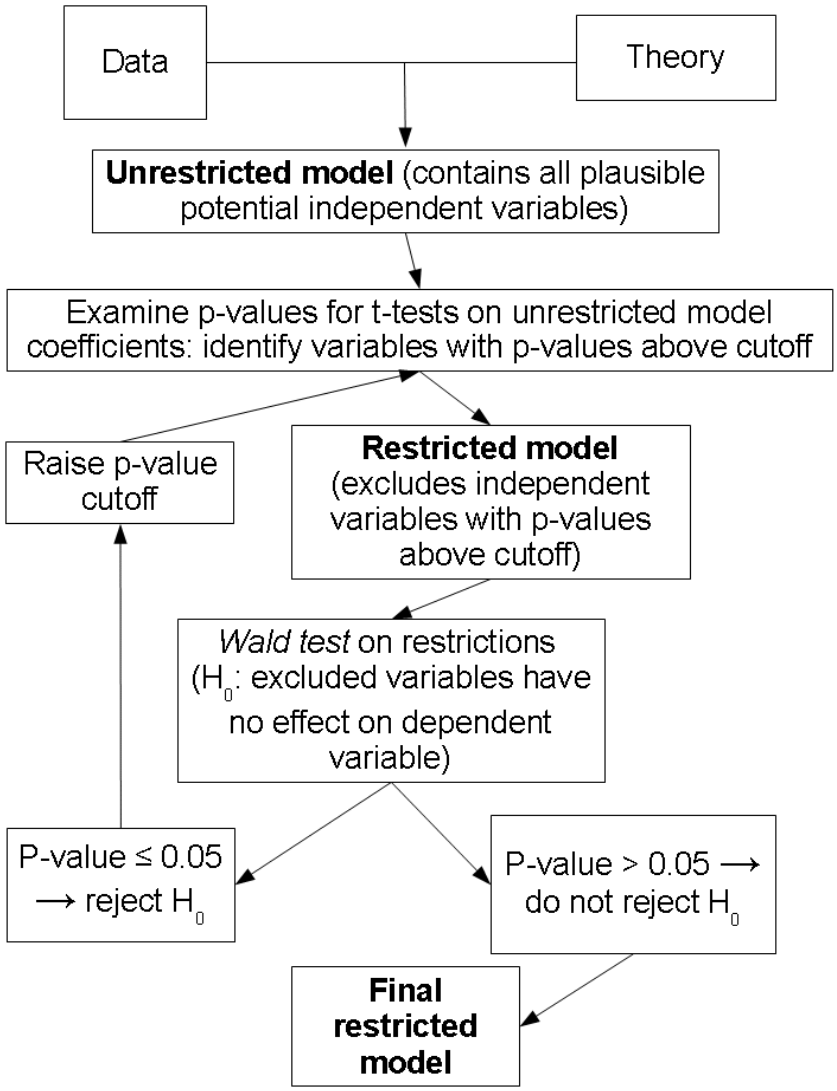

Figure 3. The model development process.

One starts by using theory to select likely determinants of the dependent variable from the data, entering all of these as independent variables in the unrestricted model (*xUR*). For the first run, also enter all of these as independent variables in the restricted model (*xR*), and make sure that at least a few of these variable names are entered in the *dropt* list. This will give you the output for the **unrestricted model**. When opened with a spreadsheet, *OLSresults.csv* for the unrestricted model looks as follows:

	x			
1	2SLS model for child value			
	coef	Fstat	ddf	pvalue	VIF
(Intercept)	-5.888	0.117	3272.225	0.732	NA
fyll	2.05	5.192	11062.306	0.023	4.264
fydd	-0.682	1.403	4955.606	0.236	3.55
cultints	1.077	3.773	31606.989	0.052	5.121
roots	-4.957	4.295	4817.677	0.038	5.031
cereals	-1.685	0.518	5070.077	0.472	7.395
gath	-0.449	0.779	5176.734	0.378	3.172
plow	-2.189	1.199	25455.972	0.274	3.188
hunt	-0.131	0.054	4352.544	0.817	5.32
fish	0.317	0.575	4315.73	0.448	3.285
anim	-0.076	0.023	2600.041	0.879	5.963
pigs	0.52	0.079	8353.731	0.779	2.257
milk	-1.662	0.801	7536.368	0.371	4.012
bovines	1.81	0.97	18457.642	0.325	4.321
tree	-5.252	3.007	6696.942	0.083	3.251
foodtrade	0.086	2.789	37736.944	0.095	1.619
foodscarc	-0.326	0.67	289.909	0.414	1.262
ecorich	-0.192	0.174	15054.621	0.676	1.85
popdens	-0.322	0.365	28611.076	0.546	3.929
pathstress	-0.084	0.195	7219.325	0.659	2.833
exogamy	-0.937	4.569	104175.349	0.033	1.478
ncmallow	-0.107	0.257	71531.545	0.612	1.672
famsize	0.305	0.352	3689.09	0.553	2.185
settype	-0.486	1.783	6408.978	0.182	4.254
localjh	-0.491	0.247	7483.425	0.619	1.874
superjh	-0.079	0.019	4393.974	0.89	2.706
moralgods	0.149	0.072	1771.716	0.788	2.266
fempower	0.33	1.127	134.268	0.29	1.392
femsubs	0.951	5.949	25262.043	0.015	1.874
sexratio	-0.127	0.02	150.467	0.887	1.361
war	-0.11	2.184	2843.41	0.14	1.405
himilexp	1.086	0.769	138.47	0.382	1.618
money	0.347	0.623	95356.347	0.43	2.364
wagelabor	-0.601	0.874	322.64	0.351	1.515
migr	0.534	0.204	179.704	0.652	1.558
brideprice	-1.094	0.762	11901.747	0.383	2.058
nuclearfam	-0.623	0.197	34959.144	0.657	2.333
pctFemPolyg	0.008	0.128	971.26	0.721	1.837

	x
R2:final model	0.264412524
R2:IV(distance)	0.923914964
R2:IV(language)	0.966727849

	Fstat	df	pvalue
RESET	3.782	334.3	0.053
Wald on restrs.	0.279	3881.84	0.597
NCV	0.057	29790.518	0.812
SWnormal	0.729	1082.343	0.393
Lagll	2.954	1694507.959	0.086
Lagdd	4.285	1716406.863	0.038

Examining the p-values for each estimated coefficient gives an indication of which independent variables can be dropped from the unrestricted model. One should then select all variables with a p-value above a cutoff (because of multicollinearity, the

cutoff should be reasonably high, 0.10 or higher) and exclude them from the **restricted model**. The names of the variables excluded (and only these) should be entered into the *dropt* list. Then run the program again. If the Wald test on the restrictions rejects the null hypothesis that the excluded variables have coefficients equal to zero, then the user should then reintroduce to the restricted model the excluded variable with the lowest p-value in the unrestricted model (and remove that variable name from the *dropt* list). This might be repeated several times, until eventually the Wald test on the restrictions ("*Wald on restrs.*", in the *ccc* object) has a high p-value, such as > 0.05, and the appropriate model is found.

The final restricted model should pass all of the *ccc* diagnostics with each p-value > 0.05. All coefficients should have p-values ≤ 0.05. It occasionally happens that some coefficients in this restricted model do not have a p-value ≤ 0.05. Try dropping those independent variables from the restricted model (and adding those variable names to the *dropt* list). If the p-value on the Wald test remains above 0.05, then it was appropriate to drop those independent variables in the restricted model.

Below is the final restricted model from Program 2, which passes all of the hurdles:

```
                x
1          2SLS model for child value

                coef         Fstat        Ddf          pvalue  VIF
(Intercept)       -9.853         0.773   997444.536   0.379  NA
fyll               1.392         7.967  1002205.226   0.005   1.32
cultints           0.796         5.702  172080609.9   0.017  1.896
roots             -2.294         4.005   4194107710   0.045  1.209
fish               0.579         5.327   1498437914   0.021  1.239
exogamy           -0.973         6.543  77725706.93   0.011  1.132
settype            -0.45         4.015   1464374798   0.045  1.685
femsubs            0.633         4.076  464824801.4   0.044  1.241
                x
R2:final model  0.106950602
R2:IV(distance) 0.923914964
R2:IV(language) 0.966727849
                Fstat        df           Pvalue
RESET               0.693  1662864.887        0.405
Wald on restrs.     0.279     3881.84         0.597
NCV                 1.104  11172006.82        0.293
SWnormal            0.492  3652353.829        0.483
Lagll               1.646  2134051.995          0.2
Lagdd               3.371  23105279.77        0.066
```

Multicollinearity is indicated by the values of the Variance Inflation Factors (VIF). A common rule of thumb is that a VIF above 10 signals that one should be concerned about multicollinearity.

The statistics at the bottom are: 1) *RESET*: Ramsey's Regression Equation Specification Error Test (H_0: model is of the correct functional form); 2) *Wald on restrs.*: Wald test for appropriateness of restricted model (H_0: dropped variables have coefficients equal to zero); 3) *NCV*: LaGrange Multiplier test for heteroskedasticity (H_0: homoskedastic residuals); 4) *SWnormal*: Shapiro-Wilk test for normality of residuals (H_0: normal residuals); 5) *lagll*: LaGrange Multiplier test for language network dependence in residuals (H_0: no autocorrelation); 6) *lagdd*: LaGrange Multiplier test for distance network dependence in residuals (H_0: no autocorrelation).

Summary

Recent papers (Dow and Eff 2009a, 2009b) have shown that missing data are a serious problem in cross-cultural survey research. The preferred method to handle the missing data problem is through multiple imputation. In this method, auxiliary data are used to impute missing values, creating five to ten separate datasets, each with slightly different imputed values. Statistical models are estimated using each of the imputed datasets, and the resulting parameter estimates combined using a well-known set of rules. If the original data set has a small N, creating additional imputed data sets may help improve the quality of results.

In this paper we first create a set of auxiliary data for the Standard Cross-Cultural Sample. Two R programs are then introduced. The first uses the auxiliary data to create imputed datasets containing variables selected from the SCCS. The second program uses these data to estimate a two-stage least-squares model, containing two or more spatial lag variables to control for Galton's problem, as described in Dow (2007). It also produces Lagrange multiplier tests for any further network autocorrelation in the residuals. Estimates are produced from each of the imputed datasets and then combined. Users should be able to use these R programs, with relatively small modifications, to estimate their own models on any cross-cultural or other survey data set.

Acknowledgements: The authors would like to thank Douglas White and two anonymous referees for their insightful and helpful comments.

Malcolm M. Dow is Professor Emeritus of anthropology and mathematical methods in the social sciences at Northwestern University. He received a BA (mathematics) and PhD (mathematical social science) from University of California, Irvine. Currently, he is engaged in a cross-cultural study (with E. Anthon Eff) testing different theories of the causes for the historical shift from polygyny to monogamy.

E. Anthon Eff is an associate professor of economics at Middle Tennessee State University. He has a BA in anthropology from the University of Louisville and a PhD in economics from the University of Texas at Austin (1989). His interests include urban and regional economics, economic anthropology, and the history of economic thought.

Program 1

```
#MI--make the imputed datasets
#--change the following path to the directory with your data and program--
setwd("c:/My Documents/MI")
rm(list=ls(all=TRUE))
options(echo=TRUE)
#--you need the following two packages--you must install them first--
library(foreign)
library(mice)

#--To find the citation for a package, use this function:---
citation("mice")

#-----------------------------
#--Read in data, rearrange----
#-----------------------------

#--Read in auxiliary variables---
load("vaux.Rdata",.GlobalEnv)
row.names(vaux)<-NULL
#--Read in the SCCS dataset---
load("SCCS.Rdata",.GlobalEnv)

#--look at first 6 rows of vaux--
head(vaux)
#--look at field names of vaux--
names(vaux)
#--check to see that rows are properly aligned in the two datasets--
#--sum should equal 186---
sum((SCCS$socname==vaux$socname)*1)
#--remove the society name field--
vaux<-vaux[,-28]
names(vaux)

#--Two nominal variables: brg and rlg----
#--brg: consolidated Burton  Regions-----
#0 = (rest of world) circumpolar, South and Meso-America, west North America
#1 = Subsaharan Africa
#2 = Middle Old World
#3 = Southeast Asia, Insular Pacific, Sahul
#4 = Eastern Americas
#--rlg: Religion---
#'0 (no world religion)'
#'1 (Christianity)'
#'2 (Islam)'
#'3 (Hindu/Buddhist)'

#--check to see number of missing values in vaux,
#--whether variables are numeric,
#--and number of discrete values for each variable---
vvn<-names(vaux)
pp<-NULL
for (i in 1:length(vvn)){
nmiss<-length(which(is.na(vaux[,vvn[i]])))
numeric<-is.numeric(vaux[,vvn[i]])
numDiscrVals<-length(table(vaux[,vvn[i]]))
pp<-rbind(pp,cbind(data.frame(numeric),nmiss,numDiscrVals))
}
```

```
row.names(pp)<-vvn
pp

#MODIFY THESE STATEMENTS FOR A NEW PROJECT
#--extract variables to be used from SCCS, put in dataframe fx--
fx<-data.frame(
socname=SCCS$socname,socID=SCCS$"sccs#",
valchild=(SCCS$v473+SCCS$v474+SCCS$v475+SCCS$v476),
cultints=SCCS$v232,roots=(SCCS$v233==5)*1,
cereals=(SCCS$v233==6)*1,gath=SCCS$v203,hunt=SCCS$v204,
fish=SCCS$v205,anim=SCCS$v206,femsubs=SCCS$v890,
pigs=(SCCS$v244==2)*1,milk=(SCCS$v245>1)*1,plow=(SCCS$v243>1)*1,
bovines=(SCCS$v244==7)*1,tree=(SCCS$v233==4)*1,
foodtrade=SCCS$v819,foodscarc=SCCS$v1685,
ecorich=SCCS$v857,popdens=SCCS$v156,pathstress=SCCS$v1260,
CVrain=SCCS$v1914/SCCS$v1913,rain=SCCS$v854,temp=SCCS$v855,
AP1=SCCS$v921,AP2=SCCS$v928,ndrymonth=SCCS$v196,
exogamy=SCCS$v72,ncmallow=SCCS$v227,famsize=SCCS$v80,
settype=SCCS$v234,localjh=(SCCS$v236-1),superjh=SCCS$v237,
moralgods=SCCS$v238,fempower=SCCS$v663,
sexratio=1+(SCCS$v1689>85)+(SCCS$v1689>115),
war=SCCS$v1648,himilexp=(SCCS$v899==1)*1,
money=SCCS$v155,wagelabor=SCCS$v1732,
migr=(SCCS$v677==2)*1,brideprice=(SCCS$v208==1)*1,
nuclearfam=(SCCS$v210<=3)*1,pctFemPolyg=SCCS$v872
)

#--look at first 6 rows of fx--
head(fx)

#--check to see number of missing values--
#--also check whether numeric--
vvn<-names(fx)
pp<-NULL
for (i in 1:length(vvn)){
nmiss<-length(which(is.na(fx[,vvn[i]])))
numeric<-is.numeric(fx[,vvn[i]])
pp<-rbind(pp,cbind(nmiss,data.frame(numeric)))
}
row.names(pp)<-vvn
pp

#--identify variables with missing values--
z<-which(pp[,1]>0)
zv1<-vvn[z]
zv1
#--identify variables with non-missing values--
z<-which(pp[,1]==0)
zv2<-vvn[z]
zv2

#------------------------------
#----Multiple imputation------
#------------------------------

#--number of imputed data sets to create--
nimp<-10
#--one at a time, loop through those variables with missing values--
for (i in 1:length(zv1)){
#--attach the imputand to the auxiliary data--
```

```
zxx<-data.frame(cbind(vaux,fx[,zv1[i]]))
#--in the following line, the imputation is done--
aqq<-complete(mice(zxx,maxit=100,m=nimp),action="long")
#--during first iteration of the loop, create dataframe impdat--
if (i==1){
impdat<-data.frame(aqq[,c(".id",".imp")])
}
#--the imputand is placed as a field in impdat and named--
impdat<-cbind(impdat,data.frame(aqq[,NCOL(zxx)]))
names(impdat)[NCOL(impdat)]<-zv1[i]
}

#--now the non-missing variables are attached to impdat--
gg<-NULL
for (i in 1:nimp){
gg<-rbind(gg,data.frame(fx[,zv2]))
}
impdat<-cbind(impdat,gg)

#--take a look at the top 6 and bottom 6 rows of impdat--
head(impdat)
tail(impdat)

#--impdat is saved as an R-format data file--
save(impdat,file="impdat.Rdata")
```

Program 2

```
#MI--estimate model with network-lagged dependent variables, combine results
rm(list=ls(all=TRUE))
#--Set path to your directory with data and program--
setwd("c:/My Documents/MI")
options(echo=TRUE)

#--need these packages for estimation and diagnostics--
library(foreign)
library(spdep)
library(car)
library(lmtest)
library(sandwich)

#-----------------------------
#--Read in data, rearrange----
#-----------------------------

#--Read in original SCCS data---
load("SCCS.Rdata",.GlobalEnv)
#--Read in two weight matrices--
ll<-as.matrix(read.dta("langwm.dta")[,-1])
dd<-as.matrix(read.dta("dist25wm.dta")[,c(-1,-2,-189)])
#--Read in the imputed dataset---
load("impdat.Rdata",.GlobalEnv)

#HERE YOU CHANGE HOW THE DEPENDENT VARIABLE IS COMPUTED FOR A NEW PROJECT
#--create dep.varb. you wish to use from SCCS data--
#--Here we sum variables measuring how much a society values children--
#--can replace "sum" with "max"
depvar<-apply(SCCS[,c("v473","v474","v475","v476")],1,sum)
#--find obs. for which dep. varb. is non-missing--
zdv<-which(!is.na(depvar))
depvar<-depvar[zdv]
#HERE GIVE THE "NAME" OF THE DEPENDENT VARIABLE THAT IS COMPUTED
depvarname<-"child value"
#--can add additional SCCS variable, but only if it has no missing values---
dateobs<-SCCS$v838
dateobs<-dateobs[zdv]

#--look at frequencies and quartiles for the dep. varb.--
summary(depvar)
table(depvar)

#--modify weight matrices---
#--set diagonal equal to zeros--
diag(ll)<-0
diag(dd)<-0
#--use only obs. where dep. varb. non-missing--
ll<-ll[zdv,zdv]
dd<-dd[zdv,zdv]
#--row standardize (rows sum to one)
ll<-ll/rowSums(ll)
dd<-dd/rowSums(dd)
#--make weight matrix object for later autocorrelation test--
wmatll<-mat2listw(as.matrix(ll))
wmatdd<-mat2listw(as.matrix(dd))
```

```
#MODIFY THESE STATEMENTS FOR A NEW PROJECT
indpv<-c("femsubs","foodscarc","exogamy","ncmallow","superjh","moralgods",
"fempower","sexratio","war","himilexp","wagelabor","famsize","settype",
"localjh","money","cultints","roots","cereals","gath","hunt","fish",
"anim","pigs","milk","plow","bovines","tree","foodtrade",
"ndrymonth","ecorich","popdens","pathstress","CVrain","rain",
"temp","AP1","AP2","migr","brideprice","nuclearfam","pctFemPolyg")

#-------------------------------------------------------
#---Estimate model on each imputed dataset------------
#-------------------------------------------------------

#--number of imputed datasets--
nimp<-10

#--will append values to these empty objects--
vif<-NULL
ss<-NULL
beta<-NULL
dng<-NULL

#--loop through the imputed datasets--
for (i in 1:nimp){

#--select the ith imputed dataset--
m9<-impdat[which(impdat$.imp==i),]
#--retain only obs. for which dep. varb. is nonmissing--
m9<-m9[zdv,]

#MODIFY THESE STATEMENTS FOR A NEW PROJECT
#--create spatially lagged dep. varbs. in stage 1 OLS--
y<-as.matrix(depvar)
xx<-as.matrix(m9[,indpv])
#--for instruments we use the spatial lag of our indep. varbs.--
#--First, the spatially lagged varb. for distance--
xdy<-dd%*%xx
cyd<-dd%*%y
o<-lm(cyd~xdy)
#--the fitted value is our instrumental variable--
fydd<-fitted(o)
#--keep R2 from this regression--
dr2<-summary(o)$r.squared
#--Then, the spatially lagged varb. for language--
xly<-ll%*%xx
cyl<-ll%*%y
o<-lm(cyl~xly)
#--the fitted value is our instrumental variable--
fyll<-fitted(o)
#--keep R2 from this regression--
lr2<-summary(o)$r.squared
m9<-cbind(m9,fydd,fyll)

#MODIFY THESE STATEMENTS FOR A NEW PROJECT
#--Stage 2 OLS estimate of unrestricted model--
xUR<-lm(depvar~fyll+fydd+dateobs+
cultints+roots+cereals+gath+plow+
hunt+fish+anim+pigs+milk+bovines+tree+foodtrade+foodscarc+
+ecorich+popdens+pathstress+exogamy+ncmallow+famsize+
settype+localjh+superjh+moralgods+fempower+femsubs+
sexratio+war+himilexp+money+wagelabor+
```

```
migr+brideprice+nuclearfam+pctFemPolyg
,data=m9)

#MODIFY THESE STATEMENTS FOR A NEW PROJECT
#--Stage 2 OLS estimate of restricted model--
xR<-lm(depvar ~ fyll + cultints + roots + fish +
    exogamy + settype + femsubs, data = m9)

#--corrected sigma2 and R2 for 2SLS--
qxx<-m9
qxx[,"fydd"]<-cyd
qxx[,"fyll"]<-cyl
b<-coef(xR)
incpt<-matrix(1,NROW(qxx),1)
x<-as.matrix(cbind(incpt,qxx[,names(b)[-1]]))
e<-y-x%*%as.matrix(b)
cs2<-as.numeric(t(e)%*%e/(NROW(x)-NCOL(x)))
cr2<-as.numeric(1-t(e)%*%e/sum((y-mean(y))^2))

#--collect coefficients and their variances--
ov<-summary(xR)
vif<-rbind(vif,vif(xR))
ss<-rbind(ss,diag(ov$cov*cs2))
#--collect robust coef. variances when there is heteroskedasticity--
#eb<-e^2
#x<-as.matrix(cbind(incpt,m9[,names(b)[-1]]))
#hcm<-inv(t(x)%*%x)%*%t(x)%*%diag(eb[1:length(eb)])%*%x%*%inv(t(x)%*%x)
#ss<-rbind(ss,diag(hcm))
beta<-rbind(beta,coef(xR))

#MODIFY THESE STATEMENTS FOR A NEW PROJECT
#--collect some model diagnostics--
dropt<-c("cereals","gath","plow","hunt","anim","dateobs",
"pigs","milk","bovines","foodscarc","ecorich",
"popdens","pathstress","ncmallow","famsize","localjh",
"superjh","moralgods","fempower","sexratio","money",
"fydd","wagelabor","war","himilexp","tree","foodtrade")


#--Ramsey RESET test--
p1<-qchisq(resettest(xR,type="fitted")$"p.value",1,lower.tail=FALSE)
#--Wald test (H0: dropped variables have coefficient equal zero)--
o<-linear.hypothesis(xUR,dropt,test="Chisq")$"Pr(>Chisq)"[2]
p2<-qchisq(o,1,lower.tail=FALSE) #find Chisq with 1 d.f. and same pvalue
#--Heteroskedasticity test (H0: homoskedastic residuals)--
p3<-ncv.test(xR)$ChiSquare
#--Shapiro-Wilke normality test (H0: residuals normal)
p4<-qchisq(shapiro.test(e)$p.value,1,lower.tail=FALSE)
#--LaGrange Multiplier test for spatial autocorrelation: language--
o<-lm.LMtests(xR, wmatll, test=c("LMlag"))
p5<-as.numeric(o$LMlag$statistic)
#--LaGrange Multiplier test for spatial autocorrelation: distance--
o<-lm.LMtests(xR, wmatdd, test=c("LMlag"))
p6<-as.numeric(o$LMlag$statistic)
#--model R2--
p7<-cr2
dng<-rbind(dng,cbind(p1,p2,p3,p4,p5,p6,p7,dr2,lr2))

}
```

```
#-------------------------------------------
#--Rubin's formulas for combining estimates--
#-------------------------------------------

#--first find final regr. coefs. and p-values--
mnb<-apply(beta,2,mean)
vrb<-colSums((beta-t(matrix(mnb,length(mnb),10)))^2)/(nimp-1)
mnv<-apply(ss,2,mean)
vrT<-mnv+vrb*(1-nimp^(-1))
fst<-mnb^2/vrT
r<-(1+nimp^(-1))*vrb/mnv
v<-(nimp-1)*(1+r^(-1))^2
pval<-pf(fst,1,v,lower.tail=FALSE)
bbb<-data.frame(round(cbind(mnb,fst,v,pval),3))
bbb$VIF[2:NROW(bbb)]<-round(apply(vif,2,mean),3)
names(bbb)<-c("coef","Fstat","ddf","pvalue","VIF")

#--Then combine the diagnostics we collected--
dng<-data.frame(dng)
names(dng)<-c("RESET","Wald on restrs.","NCV","SWnormal","lagll","lagdd",
"R2:final model","R2:IV(distance)","R2:IV(language)")
r2<-apply(dng[,7:9],2,mean)
adng<-dng[,1:6]
mdm<-apply(adng,2,mean)
vrd<-colSums((adng-t(matrix(mdm,length(mdm),nimp)))^2)/(nimp-1)
aa<-4*mdm^2-2*vrd
aa[which(aa<0)]<-0
rd<-(1+nimp^(-1))*vrd/(2*mdm+aa^.5)
vd<-(nimp-1)*(1+rd^(-1))^2
Dm<-(mdm-(nimp-1)/(nimp+1)*rd)/(1+rd)
#-All chi-sq we collected have df=1-------
pvald<-pf(Dm,1,vd,lower.tail=FALSE)
ccc<-data.frame(round(cbind(Dm,vd,pvald),3))
names(ccc)<-c("Fstat","df","pvalue")

bbb
r2
ccc

#--write results to csv file for perusal in spreadsheet--
write.csv(paste("2SLS  model  for  ",depvarname,sep=""),file="OLSresults.csv",
append=FALSE)
write.csv(bbb,file="OLSresults.csv",append=TRUE)
write.csv(r2,file="OLSresults.csv",append=TRUE)
write.csv(ccc,file="OLSresults.csv",append=TRUE)
```

References

Anselin, Luc, Anil K. Bera, Raymond Florax, and Mann J. Yoon. 1996. Simple Diagnostic Tests for Spatial Dependence. *Regional Science and Urban Economics* 26(1):77-104.

Davidson, Russell, and James G. MacKinnon. 2004. *Econometric Theory and Methods.* Oxford: Oxford University Press.

Dow, Malcolm M. 2007. Galton's Problem as Multiple Network Autocorrelation Effects. *Cross-Cultural Research* 41(4):336-363.

Dow, Malcolm M. 2008. Network Autocorrelation Regression with Binary and Ordinal Dependent Variables: Galton's Problem. *Cross-Cultural Research* 42(4):394-419.

Dow, Malcolm M, and E. Anthon Eff. 2008. Global, Regional, and Local Network Autocorrelation in the Standard Cross-Cultural Sample. *Cross-Cultural Research* 42(2):148-171.

Dow, Malcolm M, and E. Anthon Eff. 2009a. Cultural Trait Transmission and Missing Data as Sources of Bias in Cross-Cultural Survey Research: Explanations of Polygyny Re-examined. *Cross-Cultural Research* 43(2):134-151.

Dow, Malcolm M., and E. Anthon Eff. 2009b. Multiple Imputation of Missing Data in Cross-Cultural Samples. *Cross-Cultural Research* 43(3):206-229.

Eff, E. Anthon. 2008. Weight Matrices for Cultural Proximity: Deriving Weights from a Language Phylogeny. *Structure and Dynamics: eJournal of Anthropological and Related Sciences*. Vol. 3, No. 2, Article 9. http://repositories.cdlib.org/imbs/socdyn/sdeas/vol3/iss2/art9

Eff, E. Anthon, and Malcolm M. Dow. 2008. Do Markets Promote Prosocial Behavior? Evidence from the Standard Cross-Cultural Sample. MTSU Department of Economics and Finance Working Paper: http://econpapers.repec.org/paper/mtswpaper/200803.htm

Eff, E. Anthon, and Malcolm M. Dow. 2009. Market Integration and Pro-Social Behavior. *To appear in* Robert C. Marshall, Editor. *Cooperation in Economic and Social Life*. Society for Economic Anthropology Monographs Vol 26. AltaMira Press: Walnut Creek, CA.

Hijmans, R.J., S.E. Cameron, J.L. Parra, P.G. Jones and A. Jarvis, 2005. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology* 25(15):1965-1978 http://www.worldclim.org/bioclim.htm

Imhoff, Marc L., and Lahouari Bounoua. 2006. Exploring Global Patterns of Net Primary Production Carbon Supply and Demand Using Satellite Observations and Statistical Data. *Journal of Geophysical Research* 111. http://dx.doi.org/10.1029/2006JD007377

Imhoff, Marc L., Lahouari Bounoua, Taylor Ricketts, Colby Loucks, Robert Harriss, and William T. Lawrence. 2004. Global Patterns in Net Primary Productivity (NPP). Data distributed by the Socioeconomic Data and Applications Center (SEDAC): http://sedac.ciesin.columbia.edu/es/hanpp.html

Murdock, G.P. 1967. *Ethnographic Atlas*. Pittsburgh, PA: University of Pittsburgh Press.

Murdock, G.P. and White, D.R. 1969. Standard Cross-Cultural Sample. *Ethnology* 8(4):329-369.

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

Schafer, Joseph L. 2007. *mix: Estimation/multiple Imputation for Mixed Categorical and Continuous Data*. R package version 1.0-6. http://www.stat.psu.edu/~jls/misoftwa.html

Shapiro, S. S, and M. B Wilk. 1965. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* 52(3-4):591-611.

van Buuren, Stef and Karin Groothuis-Oudshoorn. 2009. *mice: Multivariate Imputation by Chained Equations*. R package version 1.21. http://CRAN.R-project.org/package=mice

von Hippel, Paul T. 2007. Regression with Missing Y's: An Improved Strategy for Analyzing Multiply Imputed Data. *Sociological Methodology* 37(1):83-117.

White, Douglas .R. 2007. Standard Cross-Cultural Sample. *International Encyclopedia of the Social Sciences*. (2nd edition). Vol "S":88-95. New York: Macmillan Reference USA. http://intersci.ss.uci.edu/wiki/pub/IntlEncyStdCross-CulturalSample.pdf

White, Douglas R., Michael Burton, William Divale, Patrick Gray, Andrey Korotayev, Daria Khalturina. 2009. *Standard Cross-Cultural Codes*. UC Irvine. http://eclectic.ss.uci.edu/~drwhite/courses/SCCCodes.htm

Wooldridge, J.M. 2006. *Introductory Econometrics: A Modern Approach*. (3rd edition). Mason, OH: South-Western College Publishers.