Missing Data Tidbits 1

Paul E. Johnson[1] [2]

[1]Department of Political Science

[2]Center for Research Methods and Data Analysis, University of Kansas

2015

Missing Data

1. Listwise Deletion
2. The Big Picture on What you Ought to Do
3. Imputation

This is a very immense literature

The focus on “Incomplete Data” flows from the highly influential research of Donald Rubin
Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). ”Maximum Likelihood from Incomplete Data via the EM Algorithm”. *Journal of the Royal Statistical Society, Series B* 39 (1): 1–38.
Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York: Wiley.
Little, Roderick J.A.; Rubin, Donald B. (1987). *Statistical Analysis with Missing Data*. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons. pp. 134–136.

Polticial Science Contribution

In political science, this burst onto our consciousness after the publication of

- King, Gary, James Honaker, Anne Joseph, and Kenneth Scheve. 2001. “Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation.” *The American Political Science Review* 95(1): 49-69.

King et al explained the problem and provided computer software to help with the work required. That has been a major (major!) success, and there is an updated version of the software for the R computing platform described here.

- James Honaker, Gary King, Matthew Blackwell (2011). Amelia II: A Program for Missing Data. *Journal of Statistical Software*, 45(7), 1-47. URL http://www.jstatsoft.org/v45/i07/.

Outline

1 Listwise Deletion

2 The Big Picture on What you Ought to Do

3 Imputation

Data Set: Columns of Same Length

- Suggest model

$$income_i = \beta_0 + \beta_1 educ + \beta_2 gender_i + e_i, \; e_i \sim N(0, \sigma^2) \qquad (1)$$

- Variables are thought of as "columns" in a data frame

row number	respondent id	*income*	*educ*	gender
1	243223	4352.5	6	M
2	151512	112423	21	F
3	515131	55345.5	13	M
4	166122	3421.4	12	M
⋮				

A Data Genie Loses Some of the Data

row number	respondent id	*income*	*educ*	gender
1	243223	4352.5	6	M
2	151512	NA	21	F
3	515131	4345.5	13	M
4	166122	3421.4	12	NA
⋮				

After Listwise Deletion

- Stat packs, since the 1970s, have typically assumed that incomplete rows should be removed entirely

row number	respondent id	*income*	*educ*	gender
1	243223	4352.5	6	M
~~2~~	~~151512~~	~~NA~~	~~21~~	~~F~~
3	515131	4345.5	13	M
~~4~~	~~166122~~	~~3421.4~~	~~12~~	~~NA~~
⋮				

- Only rows 1 and 3 survive the listwise deletion

Are The LD Estimates Valuable?

- Every student knows: you get more precise parameter estimates if you have 1000 rows of data than if you have 10 rows of data.
- Listwise deletion wastes a lot of information!
- We want to know if the OLS estimates $\hat{\beta}^{listwise\ deletion}$ are
  - unbiased
  - consistent. Does $N \to \infty$ have meaning if we throw away lots of rows?

You can see the Danger, right? Unrepresentative Samples

Example I'm making up from the top of my head

- You are studying the free-school lunch program expansion effect on educational test scores.
- You attend the school on monday and collect data from all of the children.
- On Tuesday you return to collect more data. You get lots of material in the morning, but then at 11AM some kids vanish.
  - almost of all of the kids who get the free lunch happen to be gone eating the free lunch. Their variables become missing data, "NA".
  - The only free lunchers from whom you collect data are the ones who refuse to go eat the free lunch on Tuesday, which is leftover glop from Thursday.
- Question: To what extent are estimates from this data affected by the fact some variables are NA for many free lunch students?

Some Jargon

Missing Completely at Random (MCAR). The data that is missing is gone, but there is no pattern determining which data is gone. It is *as if* your data table is sitting in the back porch and a few rain drops fall and some cells are dissolved.

Missing at Random (MAR). A value is missing, but whether it is missing or not does not depend on its own value. However, the values of the missing variables are somehow predictable from the other variables you do have (and not predictable by other unmeasured forces)

Missing Not Completely at Random (MNAR). The missing information is gone for some systematic reason. The missing mechanism is not "ignorable". Some example fixes exist for MNAR data, but they are usually specialized to particular problems (example: Heckman's selection model).

I'm aiming for the "In a nutshell" explanation

- Missing data methods can be a great career for you
- To truly master this, consider the graduate course in Psychology that studies missing data for a while semester.
- But for now, lets focus on the simple question:

what am I supposed to do if there are missing scores?

When do you have a real problem?

- If you suspect that some rows are entirely missing from your data set, you may have a problem known as "selection bias". Lets put that aside for today, work on it another time.
- If you are losing only 5% of the rows in your data due to missing values, it is unlikely that any of the fancy fixups will make much difference.
- If you believe the missings are MCAR, then listwise deletion is as good as the fancy fixups. You assert the cases you do have are "representative".
- If you have missings in more than 10%, but not more than 50% of your rows, then it is probably the case that
  - listwise deletion estimates might be biased
  - there are workable alternatives, especially if you have variables related to the missing values
- If missings affect more than 50% of rows, you are in dangerous territory.

Outline

1 Listwise Deletion

2 The Big Picture on What you Ought to Do

3 Imputation

2 Major options

1. FIML: Full Information Maximum Likelihood analysis.
2. Imputation of missing values.

   Basic Idea: make reasonable guesses, fill in NAs in the data. Repeat that several times. Re-estimate your model for each imputed data set. Pool-together the answers.

   Competing Imputation Methods

   1. Multivariate Normal Approximation.
   2. MICE: Multiple Imputation via Chained Equations.
   3. I'll ignore all of the other "ad hoc" ways. They have been shown to be flawed.

FIML

- It is endorsed most clearly in this paper
  Allison, P. D. (2012). Handling missing data by maximum likelihood. Paper 312-2012 presented at the SAS Global Forum. `http://www.statisticalhorizons.com/wp-content/uploads/MissingDataByML.pdf`
- Difficult to get this done today unless you have access to some very specialized, expensive software.
- Working through this is more mathematically challenging than we need to be today.
- People who do Structural Equation Modeling may rely on FIML because it is integrated into some software.

Outline

1 Listwise Deletion

2 The Big Picture on What you Ought to Do

3 Imputation

Imputation: Basic idea

- Rubin's proposal
  1. Use many variables, including the dependent variable and variables not planned for inclusion in the final model, to predict missings
  2. Create several "Imputed" data sets, where missings have been "filled in"
  3. Run *Each analysis* on *Each Imputed Dataset*
  4. Combine the estimates, weight them to take uncertainty into account.

Examples That Make this Believable

- Record menus of 1000s of people that eat at McDonalds.

entree	side	condiment	dessert
burger	fries	ketchup	none
burger	salad	italian dressing	none
chicken burger	fries	ketchup	cone
burger	fries	ketchup	pie

- If the Data Genie comes along and blots out the sides and/or condiments at random, you might be able to make some good guesses about what they were. (If you see fries, guess ketchup!)

Rubin's Rules for Combining Slope Estimates

- Calculation of "imputation averaged" results for Maximum Likelihood Estimates of "slope coefficients".
- **EASY** Average the imputed, $\hat{\beta} = \sum_{i=1}^{m} \hat{\beta}_j$
- **EASY** Variance of $\hat{\beta}$ is sum of
  1. average of $\widehat{Var(\hat{\beta}_j)}$, i.e., $(\sum_{i=1}^{m} = \widehat{Var(\hat{\beta}_j)})$, and
  2. a penalty for uncertainty across samples , $\frac{1}{1+m} \sum(\hat{\beta}_j - \hat{\beta})^2$.
- Ratio $\hat{\beta}/\widehat{Var(\hat{\beta})}$ is distributed as a t-statistic
- The problem of "averaging" together slope estimates is thus mostly solved, however, we have not such clear guidance on combining estimates such as the R^2, RMSE, and so forth. Unclear (so far as I know) how to do follow-up F tests for subsets of coefficients.

Yes, but How do we do these Wonderful Imputations?

- Expectation Maximization in a Multivariate Normal Approximation
- Multiple Imputation by Chained Equations

Multivariate Normal Approximation

- This was developed first. It is what Ruben had in mind.
- Championed by many leading pioneers in analysis of incomplete data
- Software for this existed first, NORM, Amelia.

Rough Sketch of Amelia

- Assume all variables are drawn from one Multivariate Normal Distribution, $MVN(\mu, \Sigma)$
- Conduct series of algorithms to estimate μ and Σ
- After estimating μ and Σ, then draw random samples from the MVN to fill in missing values
- Basic idea similar to "Norm" (J. Schafer), but algorithm may be faster.

Surprising Applicability of MVN

- Most people say “but my variables are not Normal.” (gender, survey scales, etc)
- King (and others) argue the approximation is not harmful (various reasons)
- Amelia allows user to specify variables as “nominal” and “ordinal”
    - Nominal variables: The normal imputations are “rounded off” to values in the observed scale {0,1,2}
    - Ordinal variables: Optionally “rounded off” to integers, but instructions discourage that
    - They suggest a 7 point scale might meaningfully have imputed values in-between the integers

Syntax Sketch

There are full worked examples in the workshop notes: `http://pj.freefaculty.org/guides/Rcourse/multipleImputation`

1. Get rid of extraneous variables (to speed this up)

```
datsub <- dat[ , c("names", "of", "variables", "to", "be", "
    imputed", "or", "used", "in", "imputation")]
```

2. Create imputed values

```
library(Amelia)
datimpute <- amelia(datsub, m = 10, noms = c("proper", "names",
    "of", "nominal", "vars"))
```

   - I asked for 10 sets of imputed data (Ruben suggested 5, others now say more needed).
   - IF your data includes some highly multi-correlated columns, amelia may take a long time. May be necessary to use more arguments (empirical priors as in “ridge” regression)

Syntax Sketch ...

- The list of nominal or ordinal variables does not affect the calculation of the MVN approximation, but it affects the format of the output (the extent of the rounding in the imputed variables).

3. Run the regression on each separate set.
    1. amelia creates a list structure. The imputed sets are available as datimpute$imputations.
    2. We use one of R's functions for handling a list of data structures (e.g., `lapply`)

```
allimpest <- lapply(datimpute$imputations, function(x){
    lm(dv ~ iv1 + iv2 + iv3, data = x)
})
```

    1. Remember: this can be time consuming because each separate data set must be estimated separately.

4. Use an appropriate tool to summarize the separate estimates.
    1. In the workshop notes, I demonstrate various ways this can be done. My favorite package for this is `mitools` by Thomas Lumley.

Syntax Sketch ...

```
library(mitools)
betas <- MIextract(allimpest, fun = coef)
vars <- MIextract(allimpest, fun = vcov)
summary(MIcombine(betas, vars))
```

Some Example Output

```
Multiple imputation results:
      MIcombine.default(betas, vars)
            results    se (lower upper) missInfo
(Intercept)    4.03 0.532  2.926  5.128     66 %
pclass2nd     −1.44 0.258 −1.951 −0.931     27 %
pclass3rd     −2.71 0.308 −3.339 −2.081     57 %
sexmale       −2.53 0.175 −2.872 −2.184     18 %
age           −0.05 0.011 −0.073 −0.027     73 %
```

Some Example Output

```
library(mix)
se.glm <- MIextract(allimplogreg, fun = function(x){sqrt(diag(vcov(
     x)))})
as.data.frame(mi.inference(betas, se.glm))
```

```
               est  std.err  df  signif   lower  upper     r  fminf
(Intercept)   4.03    0.532  23 1.2e-07   2.926  5.128 1.72   0.66
pclass2nd    -1.44    0.258 129 1.3e-07  -1.951 -0.931 0.36   0.27
pclass3rd    -2.71    0.308  31 6.3e-10  -3.339 -2.081 1.17   0.57
sexmale      -2.53    0.175 300 0.0e+00  -2.872 -2.184 0.21   0.18
age          -0.05    0.011  18 2.5e-04  -0.073 -0.027 2.39   0.73
```

df: degrees of freedom associated with the t reference distribution.
r: estimated relative increases in variance due to nonresponse.
fminf: estimated fractions of missing information.

Amelia Questions

- Do we really believe the data is multivariate normal?
- Is their handling of categorical variables persuasive?
- The imputer can fill in “impossible” values, like age of 244 when observed scores are in $\{1, \ldots, 99\}$

MICE

- Championed by Stef van Buuren
  - Van Buuren, S., Groothuis-Oudshoorn, K. (2011). ‘mice’: Multivariate Imputation by Chained Equations in ‘R’. *Journal of Statistical Software*, 45(3), 1-67. <URL: http://www.jstatsoft.org/v45/i03/>
  - Van Buuren, S. (2012). _*Flexible Imputation of Missing Data*. Boca Raton, FL: Chapman & Hall/CRC Press.
- Also a focus of a very large effort headed by Andrew Gelman (Columbia) called “mi”.
  - Yu-Sung Su, Andrew Gelman, Jennifer Hill, Masanao Yajima. 2011. “Multiple Imputation with Diagnostics (mi) in R: Opening Windows into the Black Box”. *Journal of Statistical Software.* 45(2)

Basic Mice Idea

- Separately process each column, predicting it from all the others. ”The algorithm imputes an incomplete column (the target column) by generating ’plausible’ synthetic values given other columns in the data.”
- Cycle through columns over and over, until model converges (in MCMC sense), then draw samples to impute.

Recommends “predictive mean matching” to select imputed values

- When filling in missings, find cases with similar predicted values to the case in question
- From among those cases, collect their list of actual observed scores
- Draw imputations from that subset of actual scores
- “Automatically” solves the problem that imputations might have impossible values
  - Imputations for categorical variables always match the original scale (sex is always 0 or 1, never 0.64)
  - When a variable is badly skewed, the PMM always selects a realistic value.

Customizes to data types

- Each column gets its own predictive model
- Defaults:

data type	default	also available
numeric	pmm (predictive mean matching)	norm, 2level
binary	logreg (logistic regression)	lda
factor	polyreg (Bayesian polytomous regression)	
factor: ordinal	polr (prop. odds logistic (MASS))	

- Possible to
  - add user-defined predictive tools
  - control the sequence of column processing

Other Handy mice Features

- complete: function can
  - return any of the individual imputed data frames
  - return all data frames combined in the “long” format (rows stacked together)
  - return all frames combined in the “wide” format (columns side-by-side)
- pool: outputs many of Rubin’s suggested diagnostic formulae (param, var, R^2)
- summary(pool( )): distills parameter estimates

Ample Diagnostic Information

All of this information is embedded in the output object

qhat: matrix of m complete data fits	b: within imputation variance
r: rel. incr var due to nonresponse	t: total variance of pooled estimates
qbar: pooled estimate	u: Variance matrices from m fits $(var \times var \times m)$
ubar: mean of variances across m fits	gamma: prop. variance explained by imputations
dfcom: df in complete analysis	df: df for pooled estimates
	fmi: fraction missing information

Default Output more Modest

```
miceTitanic <- mice( subset( titanic, select = c('survived', '
     pclass', 'sex', 'age', 'embarked')), m = 10, maxit  = 10,
     printFlag=FALSE)
miceFitTitanic <- with(data = miceTitanic, exp = glm(survived ~
     pclass + sex + age, family = binomial))
pool(miceFitTitanic)
```

```
Call: pool(object = miceFitTitanic)

Pooled coefficients:
(Intercept)      pclass2      pclass3         sex2          age
       3.97        -1.24        -2.81        -2.31        -0.05

Fraction of information about the coefficients missing due to
    nonresponse:
(Intercept)      pclass2      pclass3         sex2          age
       0.92         0.75         0.77         0.81         0.93
```

Summary looks familiar, though

```
round(summary(pool(miceFitTitanic)), 2)
```

```
               est   se    t   df Pr(>|t|) lo 95 hi 95 nmis  fmi lambda
(Intercept)   3.97 1.03  3.9 10.1     0.00   1.7   6.3   NA 0.92   0.90
pclass2      −1.24 0.40 −3.1 16.7     0.01  −2.1  −0.4   NA 0.75   0.72
pclass3      −2.81 0.41 −6.9 15.8     0.00  −3.7  −1.9   NA 0.77   0.74
sex2         −2.31 0.35 −6.5 13.8     0.00  −3.1  −1.6   NA 0.81   0.79
age          −0.05 0.02 −2.5  9.8     0.03  −0.1   0.0  680 0.93   0.92
```