Missing Data

Week 3

Simulating Missing Data

Generate data from simple regression model.

Delete some observations by MCAR.

Study the effect of missing data when using:

- Complete cases
- Mean imputation
- Random imputation

`R` commands for simulating data and for regression

`runif` returns random variable from the Uniform(0, 1)

If different interval is desired use `runif(min = a, max = b)`

`rnorm` returns random variable from the Normal(0, 1)

Different mean and standard deviation can be requested with

```
rnorm(mean = m, sd = s)
```

```
plot(x, y)
```

is used to generate a scatterplot of the variables

R commands for simulating data and for regression

`lm(formula, data)`

fits ”linear model” aka as regression to the data

Syntax is dependent variable Y ~ ind. Var1 + ind. Var2 + …

A formula has an implied intercept term. To remove this use

`y ~ x – 1`

`summary(lm)` provides the estimates of coefficients and various tests and statistics.

`predict()` provides the predicted values of the response

R commands for creating missing data

`mice` package:

`ampute(data, prop = 0.5, mech = “MAR”)`

generates multivariate missing data

`prop` specifying the proportion of missingness (number of missing cases)

`data` complete data frame or matrix

Result is of type mads (multivariate amputed data set)

To extract the “destroyed” data use `$amp`

Outline of Course

1) Missing Data Mechanisms. How are missing data generated and why should we care? Complete Case Analysis. Getting comfortable with R.
2) Simple missing data fixes: listwise deletion, available case, LVCF, mean imputation, dummy variable methods
3) **More complicated missing data fixes: weighting, hotdecking, regression imputation**
4) Building blocks and overview of multiple imputation (including regression imputation with noise)
5) Multiple imputation in practice (software in R, simple analyses, and diagnostics)
6) Multiple imputation in practice (more complicated models and considerations, more advanced diagnostics)
7) More advanced imputation and other missing data methods

Notation

Let R be the **matrix** of variables R_1, …, R_p, corresponding to variables in our dataset, X_1, …, X_p, that indicate whether a given value of the corresponding X variable is observed (= 1) or missing (= 0)

X_1	X_2	X_3	X_4
0	2	5	2
0	1	3	?
0	2	4	3
0	3	5	?
1	5	?	?
1	4	?	12
1	6	11	?
1	6	12	16

R_1	R_2	R_3	R_4
1	1	1	1
1	1	1	0
1	1	1	1
1	1	1	0
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Notation extension

Sometimes we may explicitly refer to fully observed variables by $W_1, \ldots, W_k$ and reserve the notation, $X_1, \ldots, X_p$, only for the variables in our dataset with missing data.

W_1	W_2	X_1	X_2
0	2	5	2
0	1	3	?
0	2	4	3
0	3	5	?
1	5	?	?
1	4	?	12
1	6	11	?
1	6	12	16

R_1	R_2
1	1
1	0
1	1
1	0
0	0
0	1
1	0
1	1

More advanced missing data methods

Methods that throw away data

- Non-response weighting

Methods that don’t throw away data

- Hotdecking
- Regression imputation
- Regression imputation with noise (next time)

Non-response weighting

Survey weighting background

A scientifically sound method for creating a roster of survey participants is to conduct a random sample in which n units are selected at random from a list (sample frame) of N population units, each with a known and non-zero selection probability, π_i. When $\pi_i = \pi = n/N$ for all i, the method is termed simple random sampling, but equal probabilities of selection is not mandatory — alternative, unequal sample designs can be more precise than simple random sampling depending on the population structure and estimator(s) of interest (Cochran, 1977).

Non-response weighting

- Suppose only one variable has missing data
- Calculate the probability $\Pr(R_i \mid X_i)$ of a value being missing using observed values from the other variables;
- Use these predicted probabilities to create survey weights of the form $1/\Pr(R_i \mid X_i)$ to make the complete case sample representative of the full sample once again;
- Typically we normalize by multiplying the weights by the overall (marginal) probability of missingness, $\Pr(R_i)$. This way the weights will sum to the number of people left in the complete case sample;
- This becomes more complicated when there is more than one variable with missing data
- Probabilities close to 1 or 0 can lead to crazy standard errors (due to extreme weights) though there are ways of "stabilizing" the weights

Non-response weighting

complete case dataset
plus weights

X_1	X_2	X_3	X_4
0	1	7	3
0	1	7	3
0	1	7	3
0	1	7	3
1	8	2	1
1	8	2	?
1	8	2	?
1	8	2	?
1	8	2	?
0	1	7	?

X_1	X_2	X_3	X_4	Weight
0	1	7	3	.625
0	1	7	3	.625
0	1	7	3	.625
0	1	7	3	.625
1	8	2	1	2.5

$\Pr(R_4 = 1 \mid X_1 = 0, X_2 = 1, X_3 = 7) = .2$
$\Pr(R_4 = 1 \mid X_1 = 1, X_2 = 8, X_3 = 2) = .8$
$\Pr(R_4 = 1) = .5$
weight(0, 1, 7) = .5/.2 = 2.5
weight(1, 8, 2) = .5/.8 = .625
note that Σ weights = 5 which is our complete case sample size

Extensions when several variables have missing data

- When several (or many) variables have missing data a similar approach can be used.
- In this case the indicator to be predicted, R^{cc}, is for whether or not an observation belongs to the complete case sample (or not)
- We can model $\Pr(R^{cc} \mid W)$, where W denotes fully observed variables.
- Another option is to create weights to address drop-out over time in a panel dataset. Here the missing data indicator would indicate attrition from the sample (dropping out of the study).
- This is basically what is going on with survey weights for attrition in large public-use surveys.

Hotdecking

Hotdecking – basic idea

- Replaces missing values using other values found in the dataset
- For each person with a missing value on variable *Y*, find another person who has all the same values (or close to the same values) on observed variables X_1, X_2, X_3…, and use that person's *Y* value.
- The R package VIM has a hotdeck command (we will get back to this when we begin using imputation packages)
- Stata has a hotdeck command (have to download) that randomly samples within strata defined by fully observed categorical variables

Hotdecking

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
1	2	7	3
0	4	6	?
1	5	3	1
0	3	?	2
0	4	6	2

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
1	2	7	3
0	4	6	2
1	5	3	1
0	3	8	2
0	4	6	2

Hotdecking – properties

- Typically thought of a nonparametric in that it avoids making modeling assumptions
- Census has used this technique for years
- Is potentially ok in terms of bias
- Gets complicated if many variables have missing data
- Can be problematic if there is too much missing data

Hotdecking

- Hotdecking actually comes in many varieties. The previous is a form of "exact matching" algorithm.
- However there are other ways to determine the "distance" between two observations.

Regression Imputation

- Suppose we deal with univariate missing data and we call the variable with missingness Y and the rest are denoted $\boldsymbol{X}$
- Let $\boldsymbol{X}_{\text{obs}}$ be the subset of n_1 rows of $\boldsymbol{X}$ for which Y is observed.
- Let $\boldsymbol{X}_{\text{mis}}$ be the complementing subset of n_0 rows of $\boldsymbol{X}$ for which Y is missing.
- The vector containing the n_1 observed data in Y is denoted by y_{obs}
- Finally, let the vector of n_0 imputed values in Y be indicated by $\dot{Y}$

Regression Imputation

- Within the complete cases $\boldsymbol{X}_{\text{obs}}$, build a model that predicts the values Y
- Use this model within the n_0 cases with missing data $\boldsymbol{X}_{\text{mis}}$ to predict (impute) Y
- That is,

$$\dot{Y} = \boldsymbol{X}_{\text{mis}}\widehat{\boldsymbol{\beta}}$$

where $\widehat{\boldsymbol{\beta}}$ is the LSE of $\boldsymbol{\beta}$ calculated from the n_1 cases of y_{obs} and $\boldsymbol{X}_{\text{obs}}$

- This method becomes much more complicated when there are many variables with missing data
- At best, this method will underestimate standard errors

Regression imputation: Example

X_1	X_2	X_3	Y
0	3	8	2
0	4	9	3
0	3	7	3
0	4	6	NA
1	5	10	NA
1	3	7	NA
1	2	5	NA
1	2	7	6
1	3	6	7
1	4	8	9
1	5	11	8

$\boldsymbol{X}_{\text{mis}}$

In this case we have $n_1 = 7$ and $n_0 = 4$

$\dot{Y}$ to be imputed here

Regression imputation: Example

After running the lm function in R:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.6000     1.4652   1.092   0.3547
x1             4.6000     0.5292   8.693   0.0032
x2             1.4000     0.5292   2.646   0.0773
x3            -0.4500     0.3149  -1.429   0.2483
```

Then for example, for observation 5:
$1.6 + 1(4.6) + 5(1.4) + 10(-0.45) = 8.7$

Remark: In mice this method is available as method `norm.predict`

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	7	3
0	4	6	4.5
1	5	10	8.7
1	3	7	7.25
1	2	5	6.75
1	2	7	6
1	3	6	7
1	4	8	9
1	5	11	8

Beyond Regression Imputation

- Regression imputation with noise (next time):

$$\dot{Y} = \boldsymbol{X}_{\text{mis}}\widehat{\boldsymbol{\beta}} + \dot{\varepsilon}$$

where $\dot{\varepsilon}$ is randomly drawn from the normal distribution as $\dot{\varepsilon} \sim N(0, \hat{\sigma}^2)$. In mice this method is available as method `norm.obs`

- Bayesian multiple imputation:

$$\dot{Y} = \boldsymbol{X}_{\text{mis}}\dot{\boldsymbol{\beta}} + \dot{\varepsilon}$$

where $\dot{\varepsilon} \sim N(0, \dot{\sigma}^2)$ and $\dot{\boldsymbol{\beta}}$ and $\dot{\sigma}^2$ are random draws from their posterior distribution, given the data. In mice this method is available as method `norm`