Missing Data

Multiple Imputation
Advanced techniques

Final project outline (see additional document):

- Find data set – should have been done long time ago!
- Analysis – you need to know what you want to achieve with the data, for example, regression, logistic, ANOVA, …
- Impute missing values using the steps/methods posted online and covered in class.
- After each imputation method, run the analysis you wanted to do and report the estimates and standard errors.
- Note some imputation methods are case specific, for example, LVCF is only applicable to longitudinal data
- If you created missing data yourself, compare to original data!

Final project written report:

- Introduction: a brief, half to one page, description of the data, source, types of variables, sample size, percent missing, and the analysis you want to do with these data.
- Main part: missing data imputation with various techniques. Include graphs, brief description of the method, and summary of estimates with SE after each imputation. Discuss any issues you may have encountered.
- Combined summary of results: One final table where each row corresponds to the estimates of the parameters along with SE of each imputation method.
- Discussion: Compare methods - which produced similar estimates, which had smallest SE, etc. Compare to original data (without missing) in terms of percent change in coef.

Rubric for grading the project:

- Clarity, 15%: If I scratch my head and ask myself, "what the heck are they trying to say?" several times when reading the paper, then its probably not very clear.
- Thoroughness, 60%: Did you perform all applicable imputation methods? Was the research question analysis performed after each imputation? Is there something in the data that you failed to address?
- Summary and comparison, 20%: Did you use the methods correctly and did you compare them.
- The wow factor, 5% (aka A+): Extremely well-written papers will be rewarded. Did the student go beyond the call of duty in their analysis?

Gibbs sampler

- The Gibbs' sampler is an MCMC method to generate a draw from a multivariate distribution $f(x_1, \ldots, x_p)$.
- It is used when draws from the joint distribution are hard to compute directly.
- Draws from the conditional distributions are easy to obtain

$$f(x_j \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_p)$$

- Start with some initial guess values $x_1^{(0)}, \ldots, x_p^{(0)}$
- Iterate many times to obtain a Markov chain that converges to the target distribution $f(x_1, \ldots, x_p)$.

Gibbs sampler iteration steps

- Given the current values $x_1^{(t)}, \ldots, x_p^{(t)}$, obtain new values:

- $x_1^{(t+1)} \sim f(x_1 \mid x_2^{(t)}, x_3^{(t)}, \ldots, x_p^{(t)})$
- $x_2^{(t+1)} \sim f(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)})$
- $x_3^{(t+1)} \sim f(x_3 \mid x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_p^{(t)})$

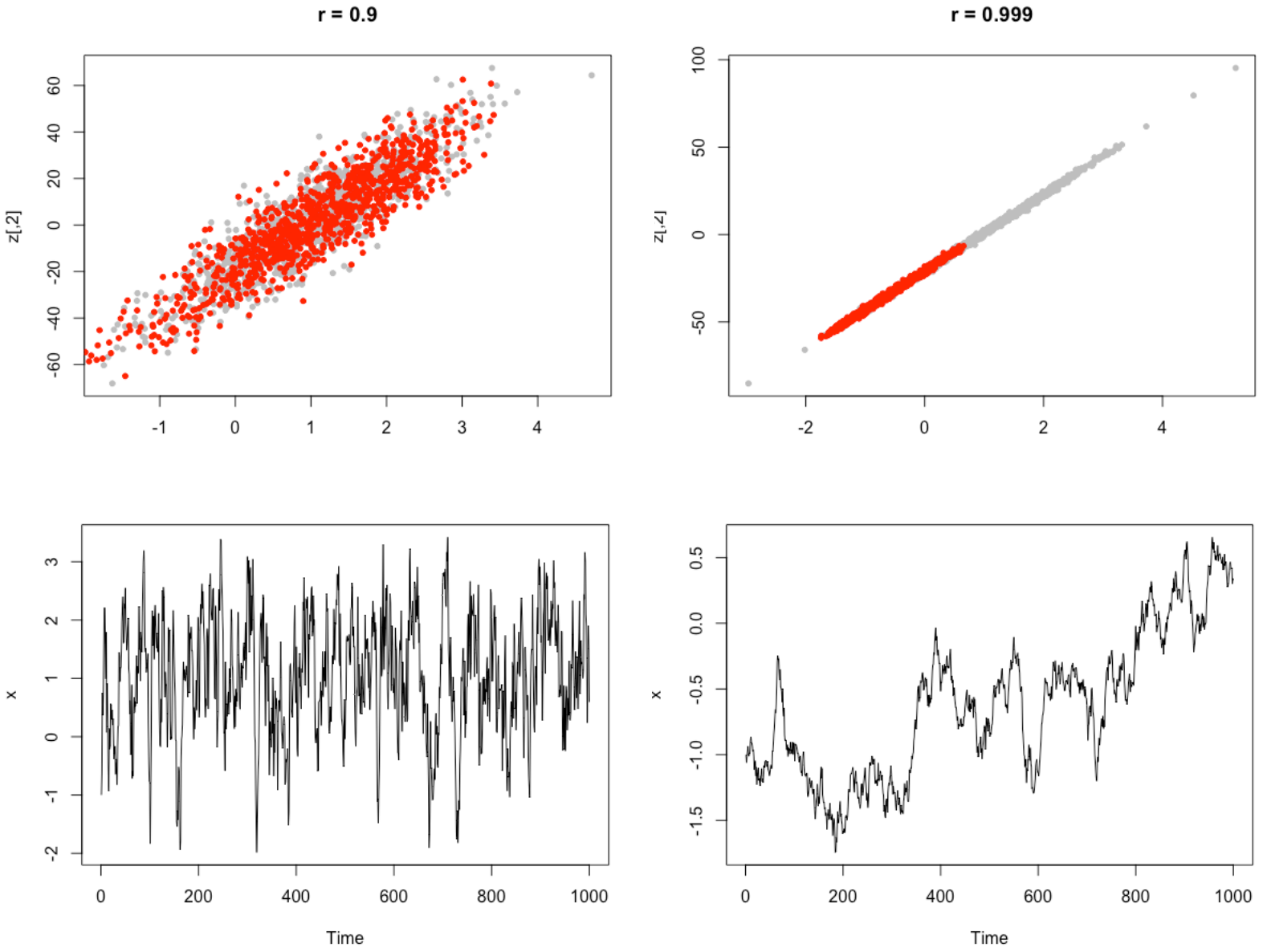
  …
- $x_p^{(t+1)} \sim f(x_p \mid x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{p-1}^{(t+1)})$

- Under mild regularity conditions it can be shown that $(x_1^{(t)}, x_2^{(t)}, x_3^{(t)}, \ldots, x_p^{(t)})$ converges to a draw from target f.

Gibbs sampler example

- We will apply the Gibbs sampler to simulate from a bivariate normal distribution.
- Formulas for the conditional distributions can be found here: https://en.wikipedia.org/wiki/Multivariate_normal_distribution
- Note there is no need for such a method as exact methods exist for multivariate normal (e.g. MASS package).
- We will use the exact simulation to judge the performance of our Gibbs sampler under different values for the parameters.
- See R file

Gibbs sampler example



Gibbs sampler connection with missing data

- When $p = 2$ the Gibbs' sampler is what is known as Data Augmentation algorithm (DA).
- The role of X_1 is Y_{mis} and of X_2 is $\boldsymbol{\theta}$
- Both distributions are conditional on Y_{obs}
- Then in the limit we obtain a draw from $(Y_{\text{mis}}, \boldsymbol{\theta} \mid Y_{\text{obs}})$
- If the model is not Bayesian, then Gibbs sampler is still applicable and Y_{mis} is sampled one component at a time, which results in a draw from $(Y_{\text{mis}} \mid Y_{\text{obs}})$ and $\boldsymbol{\theta}$ is then estimated from complete data $(Y_{\text{obs}}, Y_{\text{mis}} \mid \boldsymbol{\theta})$.

Gibbs sampler connection with missing data

- Recall the DA is ran in parallel with d chains:

- $Y_{\text{mis}}^{(d,\, t+1)} \sim f(Y_{\text{mis}} \mid Y_{\text{obs}}, \boldsymbol{\theta}^{(d,\, t)})$
- $\boldsymbol{\theta}^{(d,\, t)} \sim f(\boldsymbol{\theta} \mid Y_{\text{mis}}^{(d,\, t+1)}, Y_{\text{obs}})$

- That is, one run of the Gibbs sampler iterates to a draw from the posterior predictive distribution of Y_{mis} and the posterior predictive distribution of $\boldsymbol{\theta}$.
- If the model is not Bayesian we use the simulated Y_{mis} to estimate $\boldsymbol{\theta}$.
- See Example 10.3 from Little and Rubin.

Back to MI …

- Imputing response variables
- Practical advice regarding using `mi` for your projects
- "by" option (conditional imputation)

Imputing response variables

- If the goal of our analysis is to fit a regression and the only missingness is for the response variable
  - If the missingness is related to the outcome after conditioning on the predictors, i.e. MNAR, then using just the complete cases is wrong/biased, and MI is better.
  - If the missingness is MAR, there are no benefits in using MI, and results may just introduce extra variance.
- If the missingness occurs on predictors as well, we are often better off imputing

Advice for slow-running mi

- `mi` can run slowly on large datasets. Consider the following options
- For any dataset bigger than the ones we've been using in class set the max.minutes option, e.g.

```
> imp <- mi(mdf, max.minutes=400)
```

- If you have a dataset that is large and/or has a lot of missing data, then things can run quite slowly. A good strategy can be to diagnose the mi model on a smaller subset of the data.
- Avoid using the mi variable type "proportion" as the algorithm used to fit this type seems to be somewhat unstable
- Don't set the number of iterations for any run of mi bigger than 100 unless it runs reasonably quickly (at least under an hour). Better to run 30 or so, then feed those results into the next mi command etc.

General Advice

- Always use the most recent version of the packages and R
- If GUI is not playing nice with Rstudio try to use in standalone R.
- If you want to save all of your plots in one file you can use the following commands

```
> pdf("mi.plots.pdf")
> plot(imp)
> dev.off()
```

- Be clear to distinguish between warnings and errors. Warnings in `mi` can often be ignored. Errors definitely need to be fixed.

More info on the `pool` command

- Can add interactions, transformations, etc in pool command. For instance if you want to interact "female" with "age" the format would look like this:

  ```
  >pool(income ~ education + female*age)
  ```

  (this will include both main effects and the interaction)
- if you just want the interaction (usually not a good idea) you would use

  ```
  >pool(income ~ education + female:age)
  ```

Conditional imputation option in `mi`

```
library(mi)

data(nlsyV)
mdfs <- missing_data.frame(nlsyV, by = "first")
mdfs <- change(mdfs, y = c("income", "momrace"), what =
"type", to = c("non", "un"))
imputations <- mi(mdfs, n.iter = 30, n.chains = 3,
max.minutes = 20)
analysis <- pool(ppvtr.36 ~ first + b.marr + income +
momage + momed + momrace, data = imputations)
display(analysis)
```

So, pool() works but the plots, convergence diagnostics, etc. would require that you use list indexing notation, like

```
plot(imputations[[1]]) # for the cases where first == 0
plot(imputations[[2]]) # for the cases where first == 1
```

Back to other packages

VIM (visualization and imputation of missing values)

- `aggr` : Calculate or plot the amount of missing/imputed values in each variable and the amount of missing/imputed values in certain combinations of variables.
- `marginplot` : In addition to a standard scatterplot, information about missing values is shown in the plot margins.

MICE extras:

- `plot` function of `mice` package: shows mean and standard deviation of the variables through the iterations for the *m* imputed datasets.
- `pool.compare` function of `mice` package : used to compare models. Null hypothesis is that extra parameters are 0.

`HotDeckImputation` package

Hot Deck Imputation Methods for Missing Data

- `impute.mean` : This function imputes the column mean of the complete cases for the missing cases.
- `impute.NN_HD` : A comprehensive function that performs nearest neighbor hot deck imputation. Aspects such as variable weighting, distance types, and donor limiting are implemented. New concepts such as the optimal distribution of donors are also available.
- `impute.SEQ_HD` : Resolves missing data by sequential Hot-Deck Imputation.

`impute.NN_HD` function

Arguments

- `distance` : Distance type to use when searching for the nearest neighbor. Could be “man”, “eukl”, “tscheb”, ”mahal”, or a number for Minkowski parameter.
- `weights` : Weights by which the variables should be scaled. Can be “range”, “var”, “sd”, “none”, or a vector of numbers.
- `donor_limit` : Limits how often a donor may function as such.
- `optimal_donor` : Defines how the optimal donor is found when a donor limit is used. Could be ”no”, “rand”, “mmin”, “modifvam”, “vam”, “odd”.

Distance metrics for matching donors to recipients

In n-dimensional space distance between two points $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ is usually the Euclidean distance:

$$d_{Euc} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Other popular metrics are Manhattan (or taxicab) distance:

$$d_{Man} = \sum_{i=1}^{n} |x_i - y_i|$$

Or in general, Minkowski distance:

$$d_p = \left(\sum_{i=1}^{n} (x_i - y_i)^p \right)^{\frac{1}{p}}$$

Donors

- Sparseness of donors can lead to the over-usage of a single donor, so some hot decks limit the number of times any donor is used to impute a recipient.
- Despite its desirable properties, imputation results of a donor limited hot deck are dependent on the recipients' order of imputation, an undesirable property.
- For nearest neighbor type hot deck procedures, the implementation of a constraint on donor usage causes the stepwise matching between each recipient and its closest donor to no longer minimize the sum of all donor-recipient distances.
- Thus, imputation results may further be improved by procedures that minimize the total donor-recipient distance-sum.

Multilevel multiple imputation

- Multilevel data have hierarchical structure
- Classical examples: students within classes within schools; longitudinal studies where individual results over time are nested within the individual
- This creates natural clusters within which observations are dependent: students' scores depend on the class, which depends on the school; results over time for same individual depend on that individual
- In longitudinal studies some data vary with time, and other between the individuals

Example: `brandsma` data from package `mice`

- These data were collected by Brandsma and Knuver (1989)
- School number, cluster variable;
- Language test post (`lpo`), outcome at pupil level;
- Sex of pupil, predictor at pupil level;
- School denomination, predictor at school level.
- Research question: to predict lpo based on the other two variables

Model in level notation:

$$\text{lpo}_{ic} = \boldsymbol{\beta}_{0c} + \boldsymbol{\beta}_{1c}\,\text{sex}_{ic} + \boldsymbol{\varepsilon}_{ic}$$
$$\boldsymbol{\beta}_{0c} = \boldsymbol{\gamma}_{00} + \boldsymbol{\gamma}_{01}\,\text{den}_{c} + u_{0c}$$

Where $\boldsymbol{\varepsilon}_{ic} \sim \text{N}(0, \sigma_e^2)$, $u_{0c} \sim \text{N}(0, \sigma_u^2)$

Missingness in multilevel data

- In single-level data, missing values may occur in the outcome, or in the predictors, or in both.
- In multilevel data we may have missing in:
- The outcome;
- Level 1 predictors;
- Level 2 predictors;
- The class variable

Missing values in the level-1 predictors or the level-2 predictors have long been treated by listwise deletion.

Another ad-hoc solution is to ignore the clustering and impute the data by a single-level method.

Example

- Just two variables: school and student lpo score
- The cluster variable is complete, but there are missing scores
- Try the following methods:
- sample: Find imputations by random sampling from the observed values in lpo. This method ignores sch;
- pmm: Single-level predictive mean matching with the school indicator coded as a dummy variable;
- 2l.pan: Multilevel method using the linear mixed model to draw univariate imputations;
- 2l.norm: Multilevel method using the linear mixed model with heterogeneous error variances;
- 2l.pmm: Predictive mean matching based on predictions from the linear mixed model, with random draws from the regression coefficients and the random effects, using five donors. (miceadds library needed)

Example (continued)

- The 2l.pan and 2l.norm methods are the oldest multilevel methods. Method 2l.pan is very fast, while method 2l.norm is more flexible since the within-cluster error variances may differ. To see which of these methods should be preferred for the data, study the distribution of the standard deviation of lpo by schools.
- If the spread of standard deviations is large, then 2l.norm is preferable.