Imputation and Adjustment

Mark van der Loo, Statistics Netherlands

CBS, Department of Methodology

Complutense University of Madrid, Spring 2019

UNIVERSIDAD COMPLUTENSE MADRID

Content

Imputation

Model-based estimation of missing data.

Adjustment

Adjust (imputed) fields to satisfy linear (in)equality constraints.

Missing data

Missing data

Reasons

- nonresponse, data loss
- Value is observed but deemed wrong and erased

Solutions

- Measure/observe again
- Ignore
- Take into account when estimating
- **Impute**

UNIVERSIDAD COMPLUTENSE MADRID

Missing data mechanisms

Missing comletely at Random (MCAR)

Missingness is totally random.

Missing at Random (MAR)

Missingness probability can be modeled by other variables

Not Missing at Random (NMAR)

Missingness probability depends on missing value.

UNIVERSIDAD COMPLUTENSE MADRID

You can't tell the mechanism from the data

NMAR can look like MCAR

Given Y, X independent. Remove all $y \geq y^*$. Observer 'sees' no correlation between missingness and values of X: MAR.

NMAR can look like MAR

Given Y, X with $\text{Cov}(Y, X) > 0$. Remove all $y \geq y^*$. Observer 'sees' that higher X correlates with more missings in Y: MCAR.

UNIVERSIDAD COMPLUTENSE MADRID

Dealing with missing data mechanisms

Missing comletely at Random (MCAR)

Model-based imputation

Missing at Random (MAR)

Model-based imputation

Not Missing at Random (NMAR)

No real solution.

Imputation methodology

Model based

Estimate a value based on observed variables.

Donor-imputation

Copy a value from a record that you did observe.

The simputation package

Provide

- a *uniform interface*,
- with *consistent behaviour*,
- across *commonly used methodologies*

To facilitate

- experimentation
- configuration for production

Assignment 1: Try the following code

Installation

```
install.packages("simputation", dependencies = TRUE)
```

Code to try

```
library(simputation); library(magrittr)
data(retailers,package="validate")
ret <- retailers[3:6]
ret %>% impute_lm(other.rev ~ turnover) %>% head(3)
```

Assignment 1: Try the following code

```r
library(simputation)
data(retailers,package="validate")
ret <- retailers[3:6]
ret %>% impute_lm(other.rev ~ turnover) %>% head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607  5427.113      1607
## 3    NA     6886   -33.000      6919
```

Assignment 2: Try the following code

```
# note the 'rlm'!
ret %>% impute_rlm(other.rev ~ turnover) %>% head(3)
```

UNIVERSIDAD COMPLUTENSE MADRID

Assignment 2: Try the following code

```
# note the 'rlm'!
ret %>% impute_rlm(other.rev ~ turnover) %>% head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607  17.25247      1607
## 3    NA     6886 -33.00000      6919
```

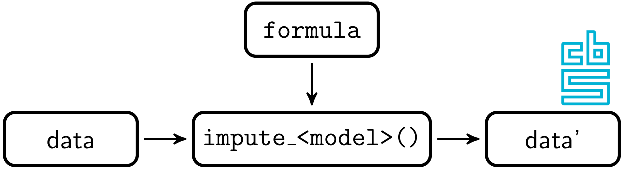
The `simputation` package

An imputation prodedure is specified by

1. The variable to impute
2. An imputation model
3. Predictor variables

The simputation interface

```
impute_<model>(data
  , <imputed vars> ~ <predictor vars>
  , [options])
```



Chaining methods

```r
ret %>%
  impute_rlm(other.rev ~ turnover) %>%
  impute_rlm(other.rev ~ staff) %>% head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA  64.88174      1130
## 2     9     1607  17.25247      1607
## 3    NA     6886 -33.00000      6919
```

UNIVERSIDAD COMPLUTENSE MADRID

Assignment 3

Expand this code so that turnover is also imputed, using on other.rev and staff as predictors.

```
ret %>%
  impute_rlm(other.rev ~ turnover) %>%
  impute_rlm(other.rev ~ staff) %>% head(3)
```

(One) solution

```r
ret %>%
  impute_rlm(other.rev ~ turnover) %>%
  impute_rlm(other.rev ~ staff) %>%
  impute_rlm(turnover ~ staff + other.rev) %>% head(3)
```

UNIVERSIDAD COMPLUTENSE MADRID

Example: Multiple variables, same predictors

```
ret %>%
  impute_rlm(other.rev + total.rev ~ turnover)

ret %>%
  impute_rlm( . - turnover ~ turnover)
```

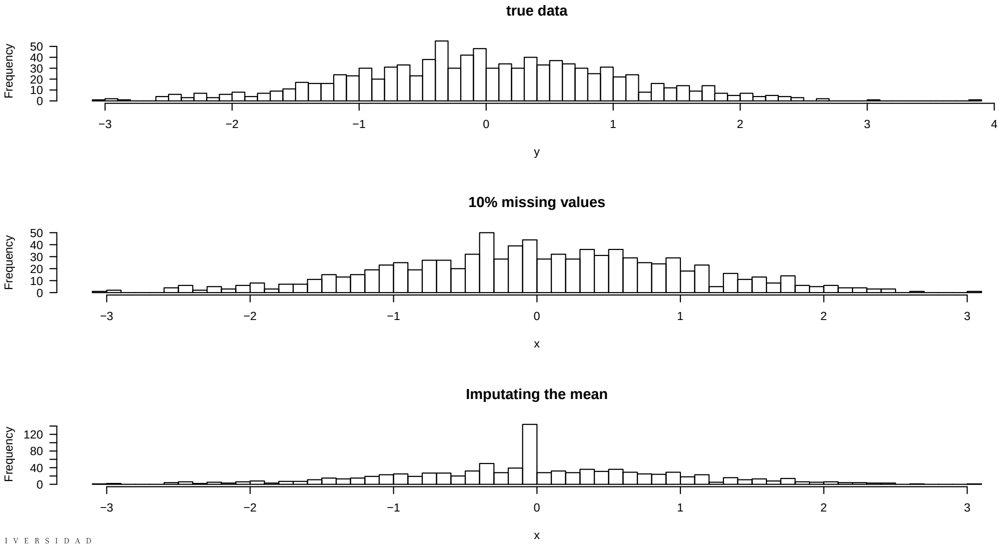
UNIVERSIDAD COMPLUTENSE MADRID

Example: grouping

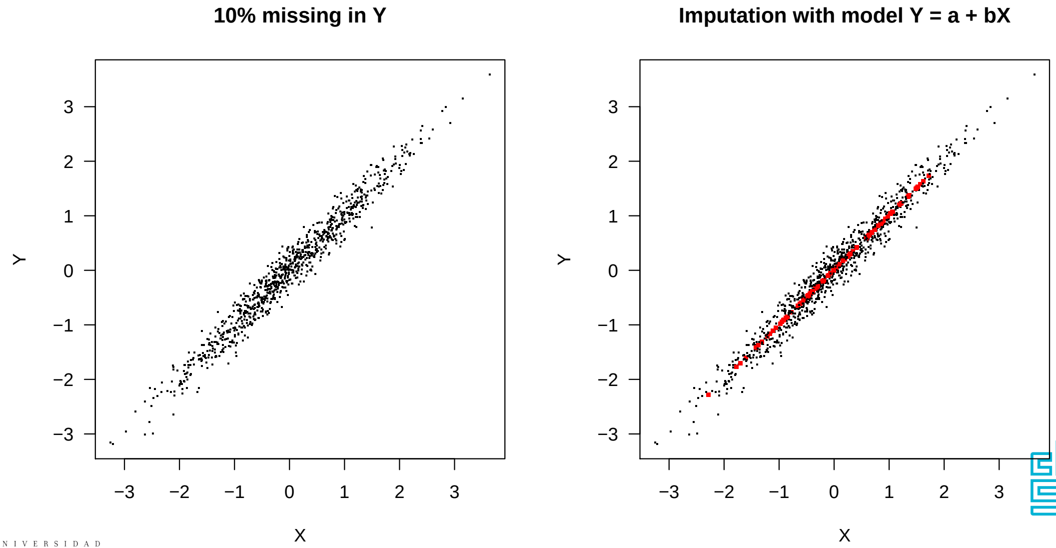
```r
retailers %>% impute_rlm(total.rev ~ turnover | size)

# or, using dplyr::group_by
retailers %>%
  group_by(size) %>%
  impute_rlm(total.rev ~ turnover)
```

UNIVERSIDAD COMPLUTENSE MADRID

Imputation and univariate distribution



Imputation and bivariate distribution



Adding a random residual

$$\hat{y}_i = \hat{f}(X_i) + \varepsilon_i$$

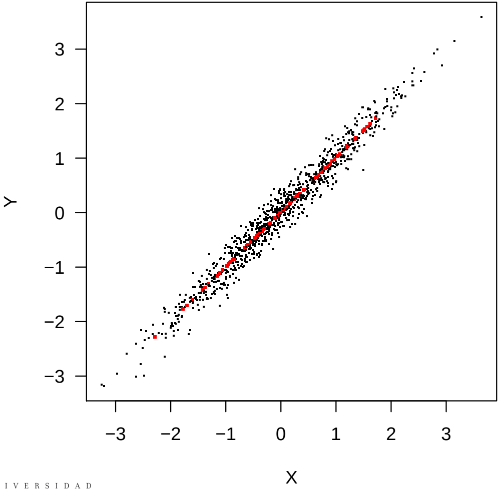
- $\hat{y}_i$ estimated value for record i
- $\hat{f}(X_i)$ model value
- ε_i random perturbation
  - Either a residual from the model training
  - OR sampled from $N(0, \hat{\sigma})$

\+ Better (multivariate) distribution

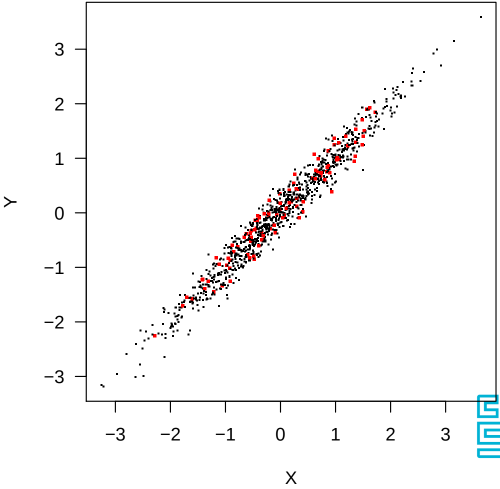
– Less reproducible

Adding a random residual

Imputation with model Y = a + bX



Imputation with Y = a + bX + e



UNIVERSIDAD COMPLUTENSE MADRID

Adding a residual with `simputation`

Try the following code

```r
ret %>%
  impute_rlm(other.rev ~ turnover
    , add_residual = "normal") %>% head(3)
```

Options

- `add_residual = "none"`: (default)
- `add_residual = "normal"`: from $N(0, \hat{\sigma})$
- `add_residual = "observed"`: from observed residuals

Compute the variance of `other.rev` after each option.

Five minutes for ten models.

1. Impute a proxy

$$\hat{\boldsymbol{y}} = \boldsymbol{x} \text{ or } \boldsymbol{y} = f(\boldsymbol{x}),$$

where $\boldsymbol{x}$ is another (proxy) variable (e.g. VAT value for turnover), and f a user-defined (optional) transformation.

```
# simputation
impute_proxy()
```

2. Linear model

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\beta},$$

where

$$\hat{\beta} = \arg\min_{\beta} \sum_i \epsilon_i^2$$

```
# simputation:
impute_lm()
```

3. Regularized linear model (elasticnet)

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\beta},$$

where

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2}\sum_i \epsilon_i^2 + \lambda\left[\frac{1-\alpha}{2}\|\beta^*\|^2 + \alpha\|\beta^*\|_1\right]$$

- $\alpha = 0$ (Lasso) $\cdots$ $\alpha = 1$ (Ridge)
- β^*: β w/o intercept.

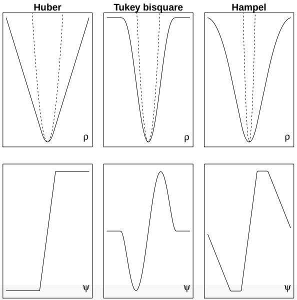
```
# simputation:
impute_en()
```

4. M-estimator

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\beta},$$

where

$$\hat{\beta} = \arg\min_{\beta} \sum_i \rho(\epsilon_i)$$



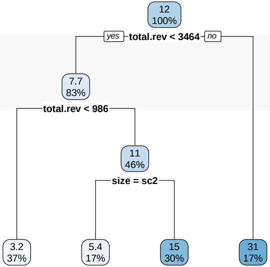

```r
# simputation:
impute_rlm()
```

5. Classification and regression tree (CART)

$$\hat{\boldsymbol{y}} = T(\boldsymbol{X}),$$

where T represents a set of binary questions on variables in $\boldsymbol{X}$. There are spare questions for when one of the predictors is missing.

```
# simputation:
impute_cart()
```



6. Random forest

$$\hat{\boldsymbol{y}} = \frac{1}{|\text{Forest}|} \sum_{i \in \text{Forest}} T_i(\boldsymbol{X}),$$

where each T_i is a simple decision tree without spare questions. For categorical $\boldsymbol{y}$, the majority vote is chosen.

```
# simputation
impute_rf()
```

UNIVERSIDAD COMPLUTENSE MADRID

7. Expectation-Maximization

Dataset $\boldsymbol{X} = \boldsymbol{X}_{obs} \cup \boldsymbol{X}_{mis}$. Assume $\boldsymbol{X} \sim P(\boldsymbol{\theta})$.

1. Choose a $\hat{\boldsymbol{\theta}}$.
2. Repeat until convergence:
   2.1 $Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}) = \ell(\boldsymbol{\theta}|\boldsymbol{X}_{obs}) + E_{mis}[\ell(\boldsymbol{X}_{mis}|\boldsymbol{\theta}, \boldsymbol{X}_{obs})|\hat{\boldsymbol{\theta}}]$
   2.2 $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}})$
3. $\hat{\boldsymbol{X}}_{mis} = \arg\max_{\boldsymbol{X}_{mis}} P(\boldsymbol{X}_{mis}|\hat{\boldsymbol{\theta}})$

```
# simputation (multivariate normal):
impute_em()
```

UNIVERSIDAD COMPLUTENSE MADRID

8. missForest

Dataset $\boldsymbol{X} = \boldsymbol{X}_{obs} \cup \boldsymbol{X}_{mis}$.

1. Trivial imputation of $\boldsymbol{X}_{mis}$ (median for numeric variables, mode for categorical variables)
2. Repeat until convergence:
   2.1 Train random forest models on the completed data
   2.2 Re-impute based on these models.

```
# simputation:
impute_mf()
```

UNIVERSIDAD COMPLUTENSE MADRID

9.a Random hot deck

1. Split the data records into groups (optional)
2. Impute missing values by copying a value from a random record in the same group

```
# simputation
impute_rhd(data, imputed_variables ~ grouping_variables)
```

UNIVERSIDAD COMPLUTENSE MADRID

9.b Sequential hot-deck

1. Sort the dataset
2. For each row in the sorted dataset, impute missing values from the last observed.

```
# simputation
impute_shd(data, imputed_variables ~ sorting_variables)
```

9.c k-nearest neighbours

For each record with one or more missings:

1. Find the k nearest neighbours (Gower's distance) with observed values
2. Sample value(s) from the k records.

```
# simputation
impute_knn(data, imputed_variables ~ distance_variables)
```

10. Predictive mean matching

1. For each variable X_i with missing values, estimate a model $\hat{f}_i$.
2. Estimate all values, observed or not.
3. For each missing value, impute the observed value, of which the prediction is closest to the prediction of the missing value.

```
# simputation: (currently buggy!)
impute_pmm()
```

UNIVERSIDAD COMPLUTENSE MADRID

Satisfy restrictions after imputation

Successive projection algorithm

Idea

Alter (imputed) values in a record $\boldsymbol{x}$ *as little as possible* to satisfy all restrictions.

As little as possible?

The minimal Eucledian distance between the original $\boldsymbol{x}$ and the adjusted record $\boldsymbol{x}^*$.

$$\boldsymbol{x}^* = \min_{\boldsymbol{x}} (\boldsymbol{x}^* - \boldsymbol{x})'(\boldsymbol{x}^* - \boldsymbol{x})$$

Successive Projection Algorithm (sketch)

Project $\boldsymbol{x}$ on each (in)equality restriction sequentially and iteratively until convergence.

Hildredth (1957) *Naval Research Logistics* **4** 79–85

Extension: weighted distance

$$\boldsymbol{x}^* = \min_{\boldsymbol{x}} (\boldsymbol{x}^* - \boldsymbol{x})' \boldsymbol{W} (\boldsymbol{x}^* - \boldsymbol{x})$$

Property

If $W_{ij} = \delta_{ij} x_j^{-1}$, then the ratios between altered variables are preserved to $\mathcal{O}(1)$.

Pannekoek & Zhang (2015) *Survey Methodology* **41** 127–144; SDCR §10.11

UNIVERSIDAD COMPLUTENSE MADRID

Implementation: `rspa`

```
library(simputation); library(validate); library(errorlocate)
SBS2000 <- read.csv("SBS2000.csv",stringsAsFactors=FALSE)
rules <- validator(.file="ruleset.R")
rules
```

```
## Object of class 'validator' with 5 elements:
##  V1: turnover >= 0
##  V2: other.rev >= 0
##  V3: turnover + other.rev == total.rev
##  V4: total.costs >= 0
##  V5: turnover - total.costs == profit
```

```
## Localize errors (important!) and replace with missings
ers    <- replace_errors(SBS2000, rules)
## A silly imputation method
imp    <- impute_proxy(SBS2000
      , turnover + other.rev + total.rev + total.costs + profit ~ 0)
```

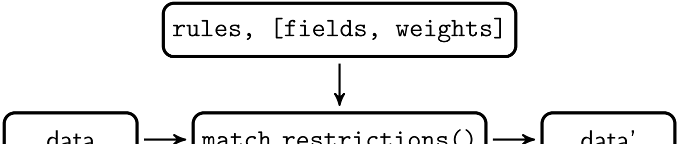
Implementation: `rspa`

```r
all( confront(imp, rules, lin.eq.eps=0.01) )
```

```
## [1] FALSE
```

```r
library(rspa)
adj     <- match_restrictions(imp, rules, adjust=is.na(ers))

all( confront(adj, rules, lin.eq.eps=0.01) )
```

```
## [1] TRUE
```



UNIVERSIDAD COMPLUTENSE MADRID