An introduction to imputation

Mark van der Loo and Edwin de Jonge

uRos2018

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**

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.

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.

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

Assignment 1: Try the following code

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

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607  5427.113      1607
## 3    NA     6886   -33.000      6919
## 4    NA     3861    13.000      3874
## 5    NA       NA    37.000      5602
## 6     1       25  6341.683        25
```

Assignment 2: Try the following code

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

Assignment 2: Try the following code

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

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607  17.25247      1607
## 3    NA     6886 -33.00000      6919
## 4    NA     3861  13.00000      3874
## 5    NA       NA  37.00000      5602
## 6     1       25  11.05605        25
```

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()
```

```
##   staff turnover other.rev total.rev
## 1    75       NA  64.88174      1130
## 2     9     1607  17.25247      1607
## 3    NA     6886 -33.00000      6919
## 4    NA     3861  13.00000      3874
## 5    NA       NA  37.00000      5602
## 6     1       25  11.05605        25
```

Assignment 3

Adapt this code so `turnover` is imputed, based on turnover and staff.

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

(One) solution

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

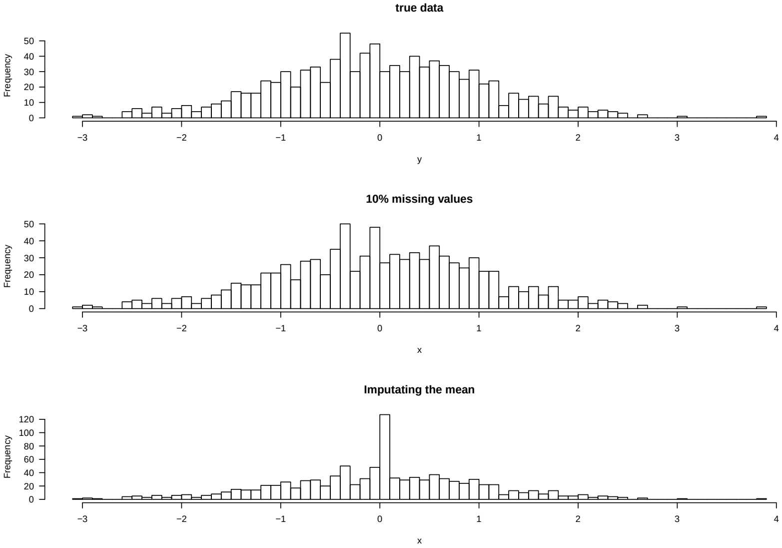
Example: Multiple variables, same predictors

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

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

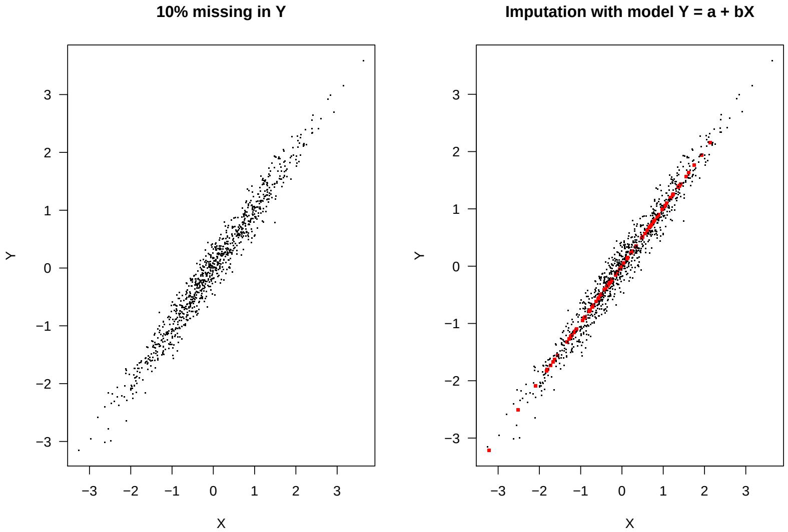
Example: grouping

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

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

Imputation and univariate distribution



Imputation and bivariate distribution



Adding a random residual

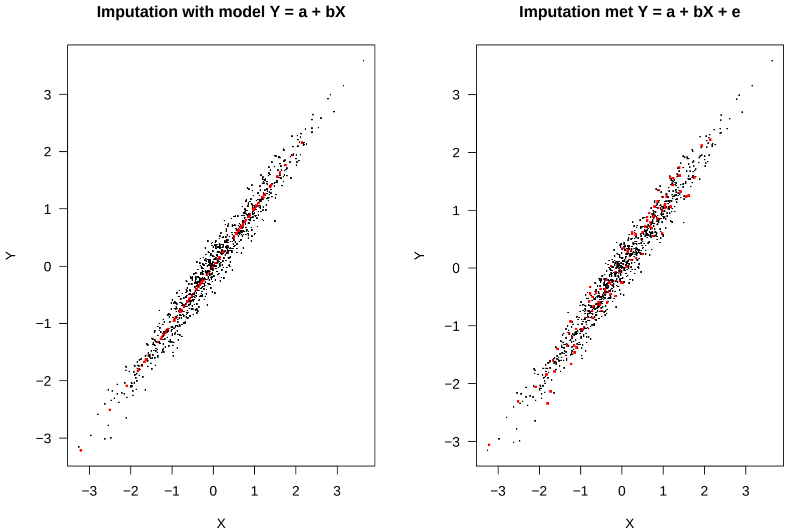
$$\hat{y}_i = \hat{f}(X_i) + \varepsilon_i$$

- $\hat{y}_i$ 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



Adding a residual with `simputation`

Try the following code

```
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

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

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

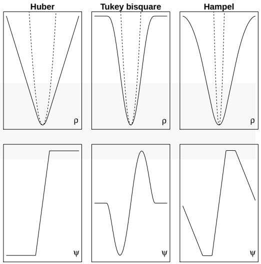
```
# simputation:
impute_en()
```

4. *M*-estimator

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

where

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



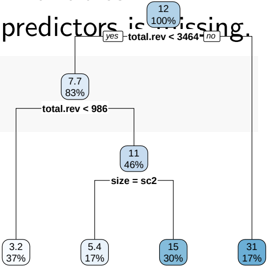

```
# simputation:
impute_rlm()
```

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.

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



6. Random forest

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

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

```
# simputation
impute_rf()
```

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:
    a. $Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}) = \ell(\boldsymbol{\theta}|\boldsymbol{X}_{obs}) + E_{mis}[\ell(\boldsymbol{X}_{mis}|\boldsymbol{\theta}, \boldsymbol{X}_{obs})|\hat{\boldsymbol{\theta}}]$
    b. $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}})$
3. $\hat{\boldsymbol{X}}_{mis} = \arg\max_{\boldsymbol{X}_{mis}} P(\boldsymbol{X}_{mis}|\hat{\boldsymbol{\theta}})$

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

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:
   a. Train random forest models on the completed data
   b. Re-impute based on these models.

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

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)
```

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()
```

Assignment 4

Read in the `irisNA.csv` dataset.

1. Use `impute_knn` to impute `Sepal.Length` and `Sepal.Width`. Use `Petal.Length`, `Petal.Width` and `Species` as predictor.
2. Use a `CART` model to impute `Sepal.Length` with all other variables as predictors (see `?impute_cart`)
3. Use `impute_lm` to impute the mean for `Sepal.Length` (the rhs of the model is `~ 1`).