An introduction to (s)imputation

Mark van der Loo and Edwin de Jonge

CBS, Department of Methodology

uRos2019, Bucharest

uRosConf 2019

Missing data

uRosConf 2019

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**

Imputation methodology

Model based
Estimate a value based on observed variables.

Donor imputation
Copy a value from a record that you did observe.

Proxy imputation
Copy or derive a value form other variables.

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()
```

uRosConf 2019

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

```r
# 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()
```

uRosConf 2019

(One) solution

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

uRosConf 2019

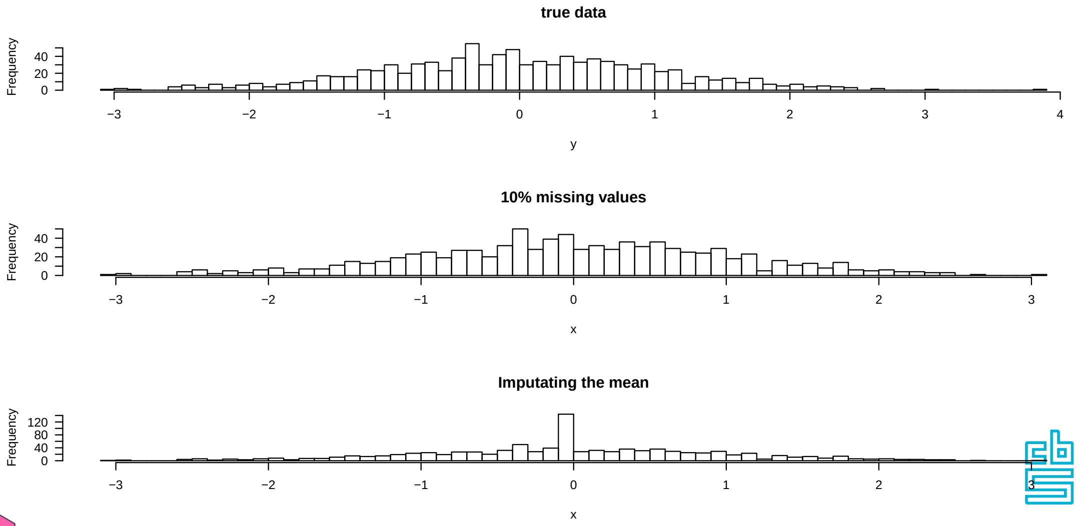
Example: Multiple variables, same predictors

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

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

Example: grouping

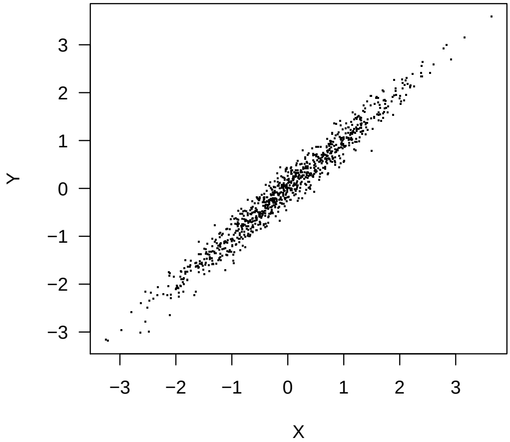
```r
retailers %>% impute_rlm(total.rev ~ turnover | size)

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

Imputation and univariate distribution

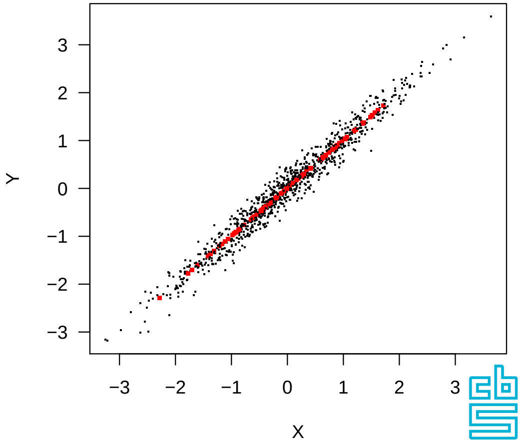


Imputation and bivariate distribution

10% missing in Y



Imputation with model Y = a + bX



uRosConf 2019

Adding a random residual

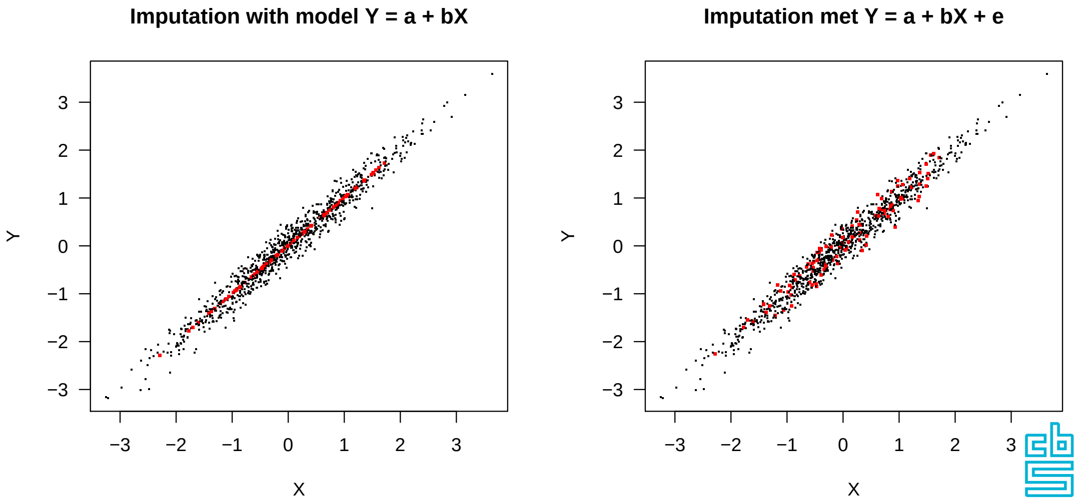
$$\hat{y}_i = \hat{f}(X_i) + \varepsilon_i$$

- $\hat{y}_i$ 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

```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.

uRosConf 2019

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()
```

uRosConf 2019

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.

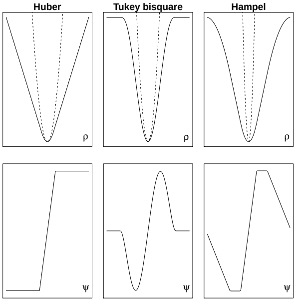
```
# simputation:
impute_en()
```

4. *M*-estimator

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

where

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




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

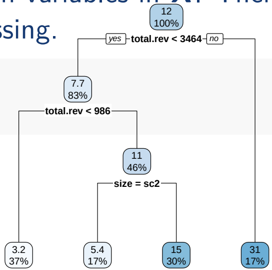
uRosConf 2019

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()
```



uRosConf 2019

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.

```r
# 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()
```

uRosConf 2019