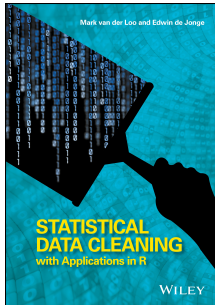


Imputation



Mark van der Loo and Edwin de Jonge

Statistics Netherlands Research & Development
Twitter: @markvdloo @edwindjonge

Tokyo | Institute of Statistical Mathematics | 2020



Try the code

```
03valid/impute.R
```



Imputing data

Need to specify

- Imputation method
- Variable(s) to impute
- Variables used as predictor

Imputation's goal

Easy to experiment, robust enough for production.

Imputation interface

```
impute_<model>(data, imputed_variables ~ predictors, ...)
```



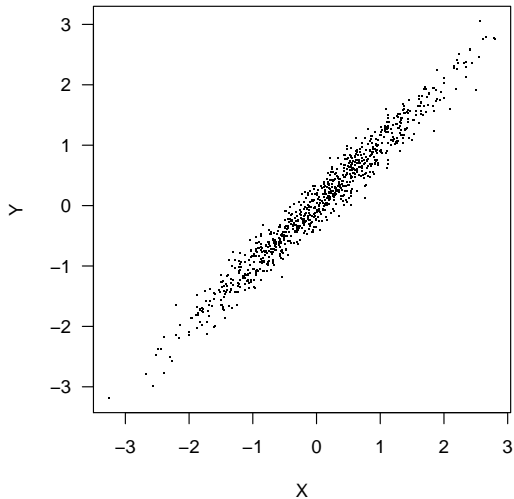
Imputing data with simulation

<model>	description
proxy	copy (transformation of) other variable(s)
median	(group-wise) median
rlm, lm, en	(robust) linear model, elasticnet regression
cart, rf	Classification And Regression Tree, RandomForest
em, mf	EM-algorithm (multivariate normal) missForest
knn	k nearest neighbours
shd, rhd	sequential, random, hot-deck
pmm	predictive mean matching
impute_model	use pre-trained model

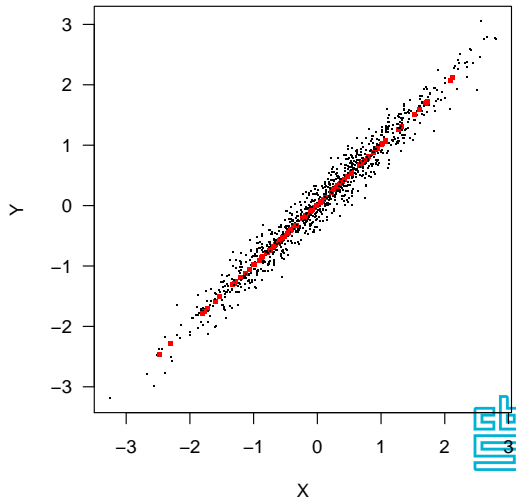


Imputation of the mean

10% missing in Y

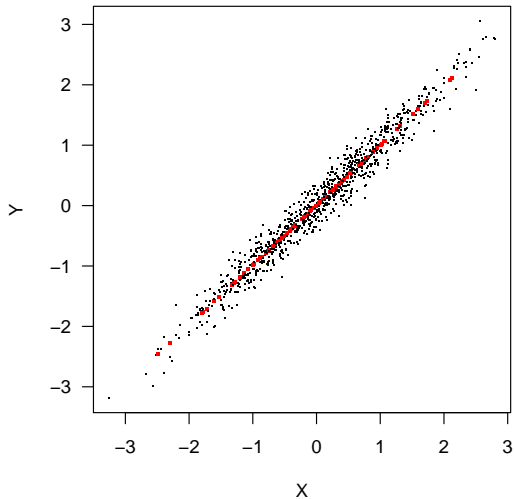


Imputation with model $Y = a + bX$

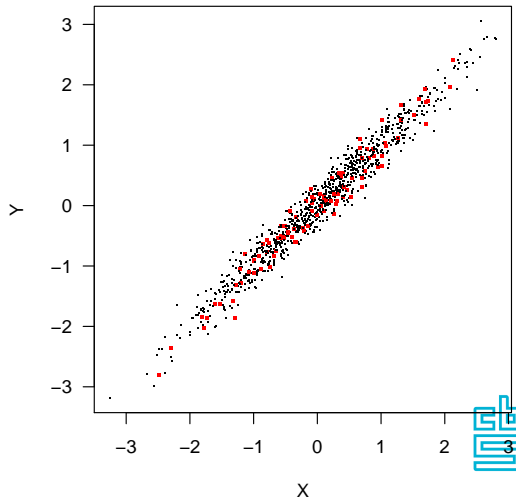


Adding a random residual

Imputation with model $Y = a + bX$



Imputation with $Y = a + bX + e$



Adding a random residual with simulation

Example

```
impute_rlm(companies, other.rev ~ turnover  
           , add_residual = "normal")
```

Options

- “none”: (default)
- “normal”: from $N(0, \hat{\sigma})$
- “observed”: from observed residuals



Chaining methods

Example

```
companies %>%  
  impute_lm(turnover ~ staff + profit) %>%  
  impute_lm(turnover ~ staff)
```



Assignment

1. Read `errors_located.csv` (`stringsAsFactors=FALSE`)
2. Make a separate data frame, selecting columns 7–14 (`staff-vat`)
3. Implement the following imputation sequence:
 - Impute turnover by copying the vat variable (`impute_proxy`)
 - Impute staff with a robust linear model based on `staff.costs`
 - Impute staff with a robust linear model based on `total.costs`
 - Impute profit as `total.rev - total.costs` (`impute_proxy`)
 - Impute everything else using `missForest` (formula: `. ~ .`)



More on missing data and (s)imputation



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 completely 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 completely 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



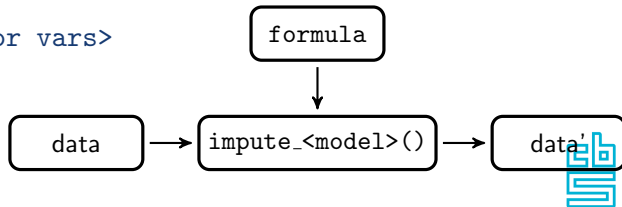
The imputation package

An imputation procedure is specified by

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

The imputation interface

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



Chaining methods

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

	##	staff	turnover	other.rev	total.rev	staff.costs	total.costs	profit	vat
## 1	75	NA	64.88174	1130	NA	18915	20045	NA	
## 2	9	1607	17.25247	1607	131	1544	63	NA	
## 3	NA	6886	-33.00000	6919	324	6493	426	NA	



Example: Multiple variables, same predictors

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

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



Example: grouping

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

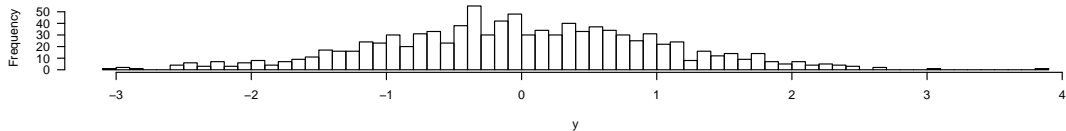
or, using dplyr::group_by

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

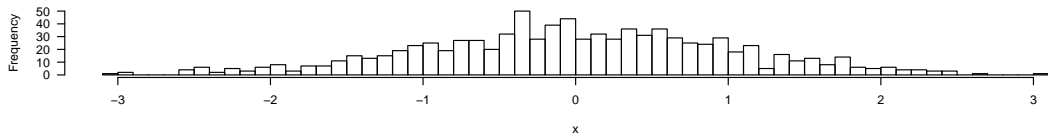


Imputation and univariate distribution

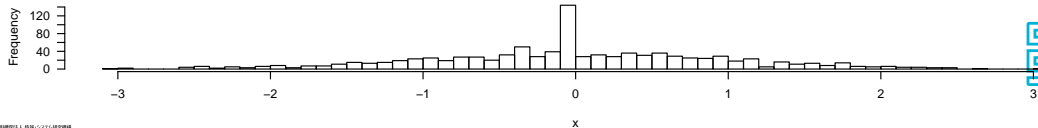
true data



10% missing values

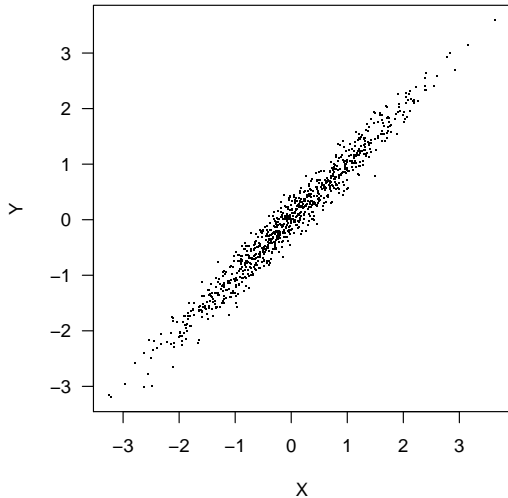


Imputating the mean

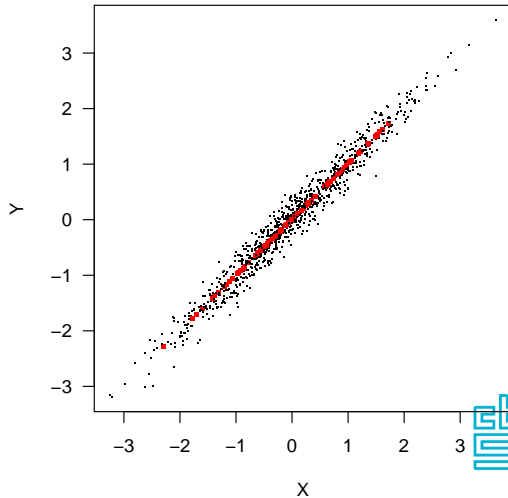


Imputation and bivariate distribution

10% missing in Y



Imputation with model $Y = a + bX$



Adding a random residual

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

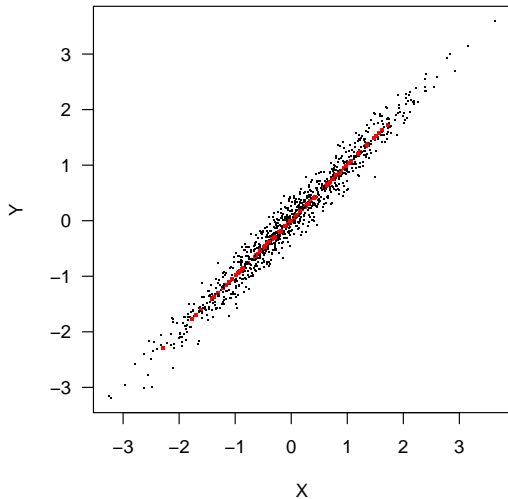
- \hat{y}_i estimated value for record i
- $\hat{f}(X_i)$ model value
- ε_i random perturbation
 - Either a residual from the model training
 - OR sampled from $N(0, \hat{\sigma})$

- + Better (multivariate) distribution
- Less reproducible

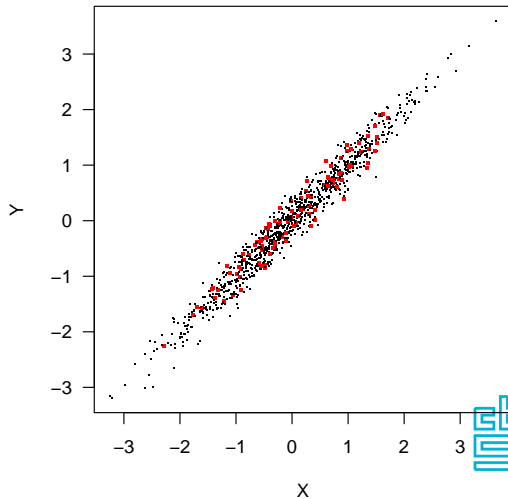


Adding a random residual

Imputation with model $Y = a + bX$



Imputation with $Y = a + bX + e$



Adding a residual with simulation

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.



Ten models.



1. Impute a proxy

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

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

```
# imputation  
impute_proxy()
```



2. Linear model

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

where

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

```
# simulation:  
impute_lm()
```



3. Regularized linear model (elasticnet)

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

where

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

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

```
# simulation:
```

```
impute_en()
```



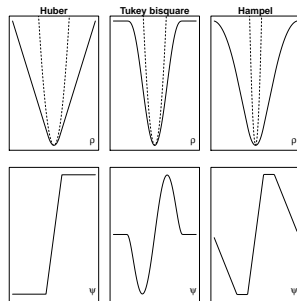
4. M -estimator

where

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

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

```
# simulation:  
impute_rlm()
```

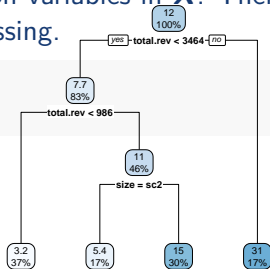


5. Classification and regression tree (CART)

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

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

```
# imputation:  
impute_cart()
```



6. Random forest

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

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

```
# simulation  
impute_rf()
```



7. Expectation-Maximization

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

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

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



8. missForest

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

1. Trivial imputation of \mathbf{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.

```
# simulation:  
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

```
# imputation  
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.

```
# imputation  
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.

```
# imputation
```

```
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.

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

