

# An introduction to (s)imputation

Mark van der Loo and Edwin de Jonge

CBS, Department of Methodology

uRos2019



# 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**



# 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()
```



## 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 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()
```

##	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

```
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)
```



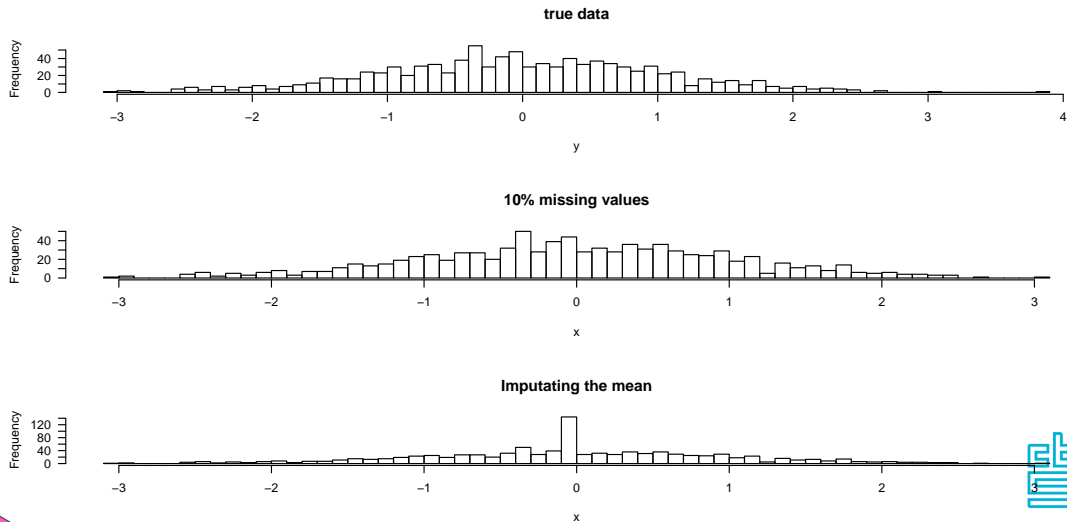
## Example: grouping

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

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



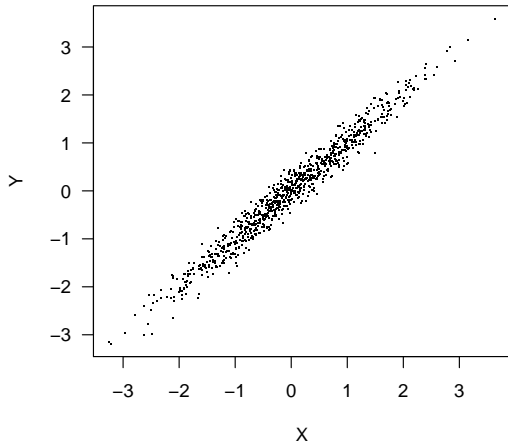
# Imputation and univariate distribution



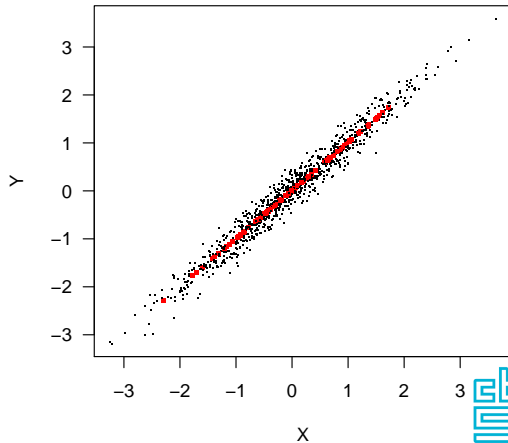


# Imputation and bivariate distribution

10% missing in Y



Imputation with model  $Y = a + bX$



# Adding a random residual

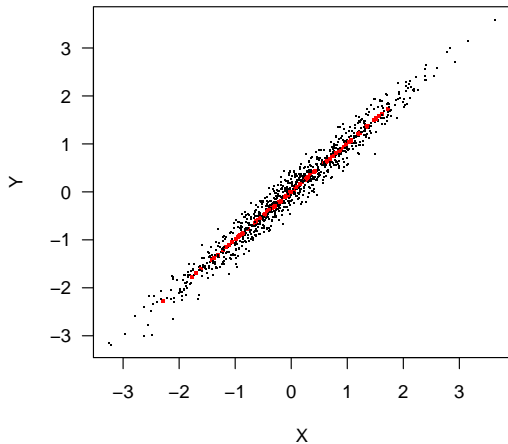
$$\hat{y}_i = \hat{f}(X_i) + \varepsilon_i$$

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

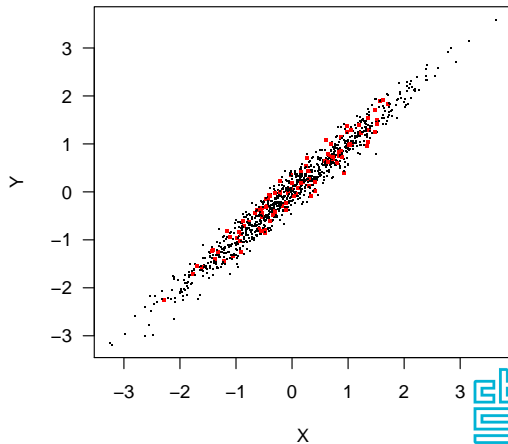


# Adding a random residual

Imputation with model  $Y = a + bX$



Imputation met  $Y = a + bX + e$



# Adding a residual with simulation

## 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

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

where  $\mathbf{x}$  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

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

```
# imputation:  
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.

*# imputation:*

```
impute_en()
```

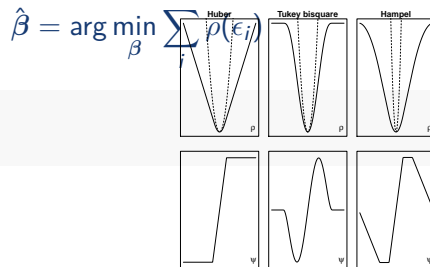




## 4. $M$ -estimator

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

where



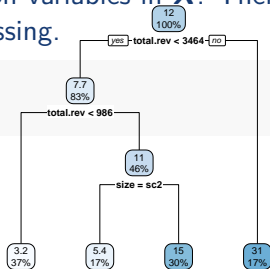
```
# simulation:  
impute_rlm()
```

## 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  $\mathbf{X} \sim P(\theta)$ .

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

```
# imputation (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()
```



# 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`).

