

Easy imputation with the simputation package

Mark van der Loo, Statistics Netherlands

@markvdloo | github.com/markvanderloo

```
# starring:  
library(simputation)  
# special guest:  
library(lumberjack)
```

Example data

```
data(retailers, package='validate')
ret <- retailers[3:7]
head(ret, 3)
```



```
##   staff turnover other.rev total.rev staff.costs
## 1    75        NA       NA     1130        NA
## 2     9      1607       NA     1607      131
## 3    NA      6886      -33     6919      324
```

Imputation in R

Specialized packages

- ▶ Many available (VIM, mice, Amelia, mi, ...)
- ▶ Interfaces vary (a lot)



DIY with model/predict

```
m <- lm(Y ~ X, data=mydata)
ina <- is.na(mydata$Y)
mydata[ina, "Y"] <- predict(m, newdata = mydata[ina,])
```

- ▶ Code duplication, doesn't always work

Idea of the simputation package

Provide

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



To facilitate

- ▶ experimentation
- ▶ configuration for production

The simputation interface

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



Example: linear model imputation

```
impute_lm(ret, other.rev ~ turnover) %>% head(3)
```

```
##   staff turnover other.rev total.rev staff.costs  
## 1    75        NA        NA     1130        NA  
## 2     9     1607  5427.113     1607      131  
## 3    NA     6886   -33.000     6919      324
```

Example: chaining imputations

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



```
##   staff turnover other.rev total.rev staff.costs  
## 1    75        NA 13914.261      1130        NA  
## 2     9       1607  6089.698      1607       131  
## 3    NA       6886   -33.000      6919       324
```

Example: robust imputation (*M*-estimation)

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



```
##   staff turnover other.rev total.rev staff.costs  
## 1    75        NA 178.04477     1130       NA  
## 2     9        1607 19.44232     1607      131  
## 3    NA       6886 -33.00000     6919      324
```

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)
```

Example: add random residual

$$+ \epsilon$$

```
retailers %>% impute_rlm(total.rev ~ turnover | size,  
                           add_residual="observed")
```

```
retailers %>% impute_rlm(total.rev ~ turnover | size,  
                           add_residual="normal")
```

Example: train on A, apply to B



```
m <- MASS::rlm(other.rev ~ turnover + staff  
                  , data=retailers)  
impute(ret, other.rev ~ m)
```

Currently available methods

- ▶ Model based (optional random residual):
 - standard/ M /elasticnet regression
 - CART models and Random forest
- ▶ Multivariate
 - EM-based imputation
 - missForest (=iterative random forest)
- ▶ Donor imputation (including various donor pool specifications)
 - k-nearest neighbour (based on gower's distance)
 - sequential, random hotdeck
 - Predictive mean matching
- ▶ Other
 - (groupwise) median imputation (optional random residual)
 - Proxy imputation: copy another variable or use a simple transformation to compute imputed values.



Who imputed what? Ask the lumberjack!

Lumberjack

A pipe operator that logs changes in data.



Provides

- ▶ `%>>%:` the lumberjack operator
- ▶ `start_log()`: start loggin'
- ▶ `dump_log()` : dump log to file
- ▶ `stop_log()` : end loggin'
- ▶ Several loggers

Fully extendable

- ▶ Users can provide their own loggers

Example

```
ret$id <- seq_len(nrow(ret))
logger <- cellwise$new(key="id")
imputed <- ret %>>%
  start_log(logger) %>>%
  impute_rlm(other.rev ~ total.rev + staff) %>>%
  impute_median(other.rev ~ 1) %>>%
  dump_log(stop=TRUE)
```

```
## Dumped a log at cellwise.csv
```

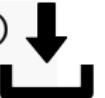
```
read.csv("cellwise.csv") %>>% head(3)
```

```
##   step           time                   expression
## 1    1 2017-07-07 07:59:26 CEST impute_rlm(other.rev ~ total.rev + staff)
## 2    1 2017-07-07 07:59:26 CEST impute_rlm(other.rev ~ total.rev + staff)
## 3    1 2017-07-07 07:59:26 CEST impute_rlm(other.rev ~ total.rev + staff)
##   key  variable old      new
## 1  11 other.rev  NA 10.9302607
## 2  12 other.rev  NA -0.2282258
## 3  18 other.rev  NA  8.8491158
```

Some pointers

Getting started

```
install.packages('simputation', dependencies = TRUE)  
install.packages('lumberjack')
```



```
vignette("intro", package="simputation")  
vignette("intro", package="lumberjack")
```



Code / issues:

github.com/markvanderloo/



Contact

@markvdloo | mark.vanderloo@gmail.com

