Methods for deterministic correction

Mark van der Loo, Statistics Netherlands

CBS, Department of Methodology

Complutense University of Madrid, Spring 2019

UNIVERSIDAD COMPLUTENSE MADRID

General strategy in data correction

- Use information stored as validation rules, and information from
  0. systematic domain knowledge (correction rules)
  1. faulty cells (e.g. spelling errors)
  2. other, correct cells (deriving values)
  3. other records (imputation)
  4. other files/sources (manual correction)

Correction based on domain knowledge

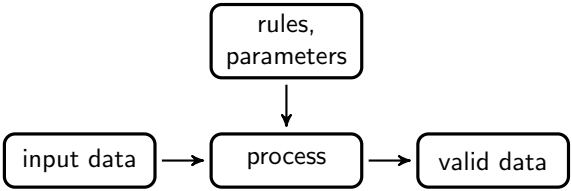
Example

- Irrelevant fields are often left open (in stead of filing zero)
- Costs are sometimes reported negative (but can be made positive)
- A surplus on a balance can be moved to a ‘rest’ post.

Common idea

`IF` some data condition holds `THEN` perform a standard activity.

Domain knowledge should be separated from process flow



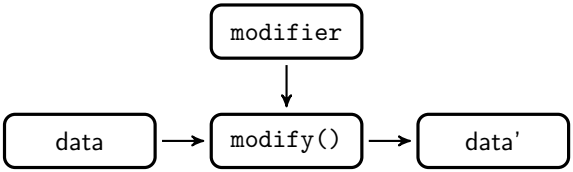
dcmodify: externalize domain knowledge

```
library(dcmodify)
SBS2000 <- read.csv("SBS2000.csv", stringsAsFactors = FALSE)
mod <- modifier(
    if (other.rev < 0) other.rev <- -1 * other.rev
  , if (staff.costs/staff > 1000) staff.costs <- staff.costs/1000
)
modify(SBS2000, mod)
```

`dcmodify`: externalize domain knowledge

Similar to `validate`

- Read rules from CLI or file
- Rules with metadata
- Apply rules



Deductive correction

Deductive correction

Deductive correction

- A set of methods specific to error circumstances
- Given the ruleset and the faulty data, can we reconstruct what went wrong?

Method: Spelling errors in numbers.

Observed data:

turnover	1024
costs	435
profit	598

Note

- $1024 - 435 = 589 \neq 598$ (so error)
- The difference $598 - 589 = 9$ is divisable by 9.

Proposition

Given a balance rule $\sum_i x_i - t = 0$. If for some record $(\boldsymbol{x}, t)$ the value $\sum_i x_i - t$ is divisable by 9, then the record can be repaired by transposition of two digits in one of the variables x_i or t.

Algorithm (sketch)

- Input: integer record, set of linear balance restrictions
- For each variable in each violated rule
  - Solve for that variable
  - Check if the solution is a typing error away from the original

SDCR §9.3

A single typing error

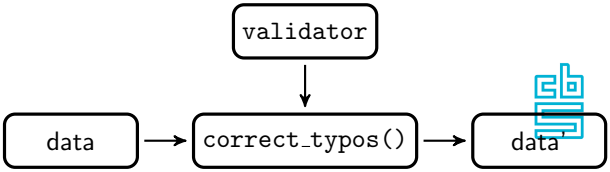
deletion, insertion, or substitution of a digit, or transposition of two adjacent digits.

Application

```
library(validate)
library(deductive)

d <- data.frame(turnover = 1024, costs = 435, profit = 598)
rules <- validator(turnover - costs == profit)
correct_typos(d, rules)
```

```
##   turnover costs profit
## 1     1024   435    589
```



Deductive imputation

Method: deductive imputation

Observed data:

turnover	1024
costs	NA
profit	589

Note

We can derive the value of *costs* using the rule

```
turnover - costs == profit
```

Method: deductive imputation

Observed costs:

housing	1024
transport	NA
staff	NA
interest	300
total	1324

Observe

Since all variables must be nonnegative, the only possible value for *transport* and *staff* is zero.

Method: deductive imputation

```
library(validate)
library(deductive)

d <- data.frame(housing=1024, transport=NA, staff=NA, interest=300
              , total=1324)
rules <- validator(housing >= 0, transport >= 0, staff >=0, interest >=0
                   , housing + transport + staff + interest == total )
impute_lr(d,rules)
```

```
##   housing transport staff interest total
## 1    1024         0     0      300  1324
```

General method for deductive imputation (sketch)

Set of restrictions $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Split according to observed and missing:

$$[\boldsymbol{A}_o, \boldsymbol{A}_m] \begin{pmatrix} \boldsymbol{x}_o \\ \boldsymbol{x}_m \end{pmatrix} = \boldsymbol{b}$$

Then, depending on the number and structure of missings, some elements of $\hat{\boldsymbol{x}}_m$ are uniquely determined.

$$\hat{\boldsymbol{x}}_m = \boldsymbol{A}^+(\boldsymbol{b} - \boldsymbol{A}_o\boldsymbol{x}_o) + (\boldsymbol{1} - \boldsymbol{A}_m^+\boldsymbol{A}_m)\boldsymbol{w}$$

- $\boldsymbol{A}^+$: Moore-Penrose inverse of $\boldsymbol{A}$
- $\boldsymbol{w} \in \mathbb{R}^{|m|}$ ($|m|$ is nr of missings in $\boldsymbol{x}$).

SDCR §9.3

The good news

R package `deductive`

`deductive::impute_lr` tries all these methods iteratively untill nothing more can be imputed, based on these methods.