Methods Library in R at Statistics Norway

Øyvind Langsrud

Statistics Norway, Division for methods
Project: Modernization of KOSTRA

- Economy
- Schools
- Health
- Culture
- The environment
- Social services
- Public housing
- Technical services
- Transport and communication
### The project execution plan

<table>
<thead>
<tr>
<th>Date</th>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.16</td>
<td>Kommuneprofil</td>
<td>Kostra modernisering</td>
</tr>
<tr>
<td>8.16</td>
<td>Kommunefakta med KOSTRA-tall</td>
<td></td>
</tr>
<tr>
<td>11.16</td>
<td>Metodebibliotek, informasjonsmodell, felles datalager</td>
<td></td>
</tr>
<tr>
<td>1.17</td>
<td>Metodebibliotek</td>
<td></td>
</tr>
<tr>
<td>3.17</td>
<td>Metodebibliotek</td>
<td></td>
</tr>
<tr>
<td>6.17</td>
<td>Metodebibliotek, informasjonsmodell, felles datalager</td>
<td></td>
</tr>
<tr>
<td>8.17</td>
<td>Kommunefakta med KOSTRA-tall</td>
<td></td>
</tr>
<tr>
<td>9.17</td>
<td>Metodebibliotek, informasjonsmodell, felles datalager</td>
<td></td>
</tr>
<tr>
<td>10.17</td>
<td>Metodebibliotek</td>
<td></td>
</tr>
<tr>
<td>1.18</td>
<td>Metodebibliotek</td>
<td></td>
</tr>
<tr>
<td>3.18</td>
<td>Metodebibliotek</td>
<td></td>
</tr>
</tbody>
</table>

- **Employees at Division for methods (mathematical statistics)**
- **R programming**

**Project deliveries**

- **Kommune profil**
- **Metode bibliotek**
- **Kompis klar til å strukturere data**
- **PX-Web intern bruk**

**Methods Library**

- **Metoder for estimering og editering**
- **Starte bygging av et R-metodebibliotek**

**Statistikksentralbyra Statistics Norway**

**KOSTRA modernisering**
R Library ↔ Methods Library

- **Methods Library**
  - Functions to be integrated in IT systems at SSB
  - Included in the administrative module
  - Currently only R implementations, but future extensions (?)

- **R Library**
  - R packages developed at SSB. Especially packages meant for the methods library
  - All methods in the methods library have a corresponding R function
  - Also many useful R functions not in the methods library
  - The R library useful outside the methods library

- **Administrative module:**
  - Technical information about the methods (parameters)
  - Short descriptions for users.
  - This is used to generate user interface and to run the methods within the IT system.
An internal R programming guide

• Google's R style guide as starting point

• We recommend writing the code as portable as possible (cross-platform)
  • In practice this means that we will, when possible, make the code compatible with Renjin. Especially, this means that we will only use packages such as `data.table` and `dplyr` if this in practise makes very important efficiency improvements.
Methods library functions - strict standard

• **Input**
  - The first parameter is a data set of type `data.frame`.
  - The other parameter must be vectors (normally of length one) of type `character`, `numeric`, `integer` or `logical`.
  - A special variant of the character type is `variable name`, referring to variables in the input data set.
    - One such `variable name` parameter represents a unique identifier of observations.
  - When input is character, a list of allowed input elements can be specified.
  - When input is numeric or integer, the minimum and/or maximum can be specified.

• **Output**
  - A single data set of type `data.frame`.

The single data set standard is too restrictive and will be changed in future systems at Statistics Norway

The R code consists of several functions that are not exposed in the library and for such functions these rules do not apply.
Example: First lines of a function

```r
ThError <- function (data, id, x1, x2, ll = -2.5, ul = 2.5) 
{
  CheckInput(id, type = "varName", data = data)
  CheckInput(x1, type = "varName", data = data)
  CheckInput(x2, type = "varName", data = data)
  CheckInput(ll, min = -3, max = 0, type = "numeric")
  CheckInput(ul, min = 0, max = 3, type = "numeric")
  
  d <- GetData(data, id = id, x1 = x1, x2 = x2)
```

Always CheckInput at the beginning
Always GetData() also
CheckInput()

• All input parameters are checked by this function
• Double purpose
  • Input checked and error messages produced in a standardised way
  • The linens with CheckInput() together with the function header give a complete overview of input requirements.
    • Possible to read by a machine
    • A check list when specifying the function in the administrative module
GetData()

- Programmer can assume that GetData() is

```r
GetData <- function(data, ...) {
  a <- unlist(list(...))
  b <- data[, a, drop = FALSE]
  colnames(b) <- names(a)
  b
}
```

- When vector input: Data frame with matrix embedded in one variable created
- In reality GetData() is more advanced
  - Possible to specify more information (as list) instead of a variable name
  - Variables can be taken from specific years ...
  - Variables can be merged by identifiers ...

```
> print(z)
year  R  Age  ABC
 1 2014 4.4  10 A
 2 2015 6.6  20 A
 3 2016 2.2  30 B
 4 2014 3.2  40 B
 5 2015 8.8  50 C
 6 2016 9.9  60 C
> GetData(z, x = "Age", y = "R")
   x  y
 1 10  4.4
 2 20  6.6
 3 30  2.2
 4 40  3.2
 5 50  8.8
 6 60  9.9
```
When the standards don't fit

- Make the function(s) the way you want
  - Make a call to this function from a standardised function
  - Publish your non-standardised function externally

- Use function(s) form external package
  - Make a call to this function from a standardised function
Regression imputation

- Observations are categorised into three groups:
  - Representative: Used for imputation modelling.
  - Non-representative: Values are kept.
  - Missing or wrong: Values are imputed.
- Categorisation based on externally studentized residuals.
- Iterative process
- A general function, *LmImpute*, is the working horse within several library functions.
  - *ImputeRegression*, *ImputeRegression2*, *ImputeRegressionMulti*, *ImputeHistory*, *OutlierRegression*
### Table suppression - simple method

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Arbeid</th>
<th>Trygd</th>
<th>Soshjelp</th>
<th>Annet</th>
</tr>
</thead>
<tbody>
<tr>
<td>113</td>
<td></td>
<td>11</td>
<td>36</td>
<td>55</td>
<td>11</td>
</tr>
<tr>
<td>55</td>
<td></td>
<td>1</td>
<td>18</td>
<td>29</td>
<td>7</td>
</tr>
<tr>
<td>73</td>
<td></td>
<td>8</td>
<td>25</td>
<td>35</td>
<td>5</td>
</tr>
<tr>
<td>45</td>
<td></td>
<td>2</td>
<td>13</td>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>138</td>
<td></td>
<td>14</td>
<td>52</td>
<td>63</td>
<td>9</td>
</tr>
<tr>
<td>67</td>
<td></td>
<td>9</td>
<td>22</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>4</td>
<td>8</td>
<td>22</td>
<td>6</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>3</td>
<td>15</td>
<td>38</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>596</td>
<td>52</td>
<td>189</td>
<td>283</td>
<td>72</td>
</tr>
</tbody>
</table>
# Table suppression - optimal method

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Arbeid</th>
<th>Trygd</th>
<th>Soshjelp</th>
<th>Annet</th>
</tr>
</thead>
<tbody>
<tr>
<td>113</td>
<td>11</td>
<td>36</td>
<td>55</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>18</td>
<td>29</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>8</td>
<td>25</td>
<td>35</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>13</td>
<td>17</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>14</td>
<td>52</td>
<td>63</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>9</td>
<td>22</td>
<td>24</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>8</td>
<td>22</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>3</td>
<td>15</td>
<td>38</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>596</td>
<td>52</td>
<td>189</td>
<td>283</td>
<td>72</td>
</tr>
</tbody>
</table>
Modernization of KOSTRA: Table suppression

• Request
  • Table suppression according to a frequency rule
  • As an r function for methods library

• Starting point
  • Package sdcTable
Challenge
Making the function general and at the same time satisfying the standard for the library

- New methodology was developed
  - Automatic detection of hierarchical relationships between variables
  - Automatic creation of lists of level-hierarchy (parameter 'dimList')
  - Handling two linked tables automatically

- Result
  - Easy interface could be made

- Package easySdcTable was made
  - Made public available (CRAN)
New functionality in ‘sdcTable’ in 2017

• Interface to tau-argus
• Included in easySdcTable as well
Graphical user interface - shiny

• For users outside methods library
  • not necessarily familiar with R
• Included in easySdcTable
library(easySdcTable )
z2 <- EasyData("z2")
z2
ProtectTable(z2,c(1,3,4),5)  # With region-variable kostragr
ProtectTable1(z2,c(1,3,4),5)
ProtectTable(z2,c(1,2,3,4),5)  # Two linked tables
ProtectTable(z2,c(1,2,3,4),5,addName=TRUE)
ProtectTable1(z2,c(1,2,3,4),5,addName=TRUE)

z2w <- EasyData("z2w")
z2w
ProtectTable(z2w,1:3,4:7)  # Two linked tables

exeArgus <- "C:/XXXXXXX/R/TauArgus/TauWindows4.1.4_updated/TauArgus4.1.4/TauArgus.exe"
pathArgus <- "C:/Users/XXXXXXXXX"
PTgui(EasyData("z2w"),exeArgus=exeArgus,pathArgus=pathArgus)
library(easySdcTable)
exeArgus <- "C:/XXXXXXX/R/TauArgus/TauWindows4.1.4_updated/TauArgus4.1.4/TauArgus.exe"
pathArgus <- "C:/Users/XXXXXXXXX"
PTgui(EasyData("z2w"),exeArgus=exeArgus,pathArgus=pathArgus)
Conclusion

• Using R as the programming language within the methods library has been successful.

• Methods library and R will expand at Statistics Norway

• And in other countries ...
  • common standards
  • sharing and reuse