TALL

SOM FORTELLER

# A Sparse Matrix Tool in R for Multidimensional Hierarchical Aggregation


Øyvind Langsrud

Statistics Norway, Oyvind.Langsrud@ssb.no, github.com/olangsrud



Use of R in Official Statistics 2020 - uRos2020 - virtual event 2-4 December 2020



Abstract: Commonly, tabular aggregations involve hierarchical classifications (hierarchies) that follow a tree structure. Free hierarchies are more advanced. Then, each code can be specified as a formula involving other codes. Addition or union is specified by the plus sign and subtraction or compliment is specified by the minus sign. Statistics Norway's modernized calculations of municipal accounts involve complicated free hierarchies. The intention was to do the computations by an implementation of the Validation and Transformation Language (VTL), but this approach turned out to be too inefficient. Instead, a general hierarchical computing method was implemented in R as the function HierarchyCompute in the package SSBtools. Hierarchies and crossed hierarchies are represented by sparse matrices according to the Matrix package. The aggregates are obtained by matrix multiplication. The matrix representation therefore makes it easy to see how each output code is calculated from the input codes. A spin-off from the general implementation is that hierarchy specifications, usual in statistical disclosure control software, can be managed. This has been utilized in the package SmallCountRounding, which can be used to protect frequency tables.


- R package `SSBtools`
- Function `HierarchyCompute()`

Statistisk sentralbyrå
Statistics Norway

# Municipal accounts

## 12367: Detailed accounting figures in the operational accounts , by accounting scope , function and type (M) 2015 - 2019

Choose table — Choose variables — Show result

List view

About table

**Contents** Mandatory *

Select all | Deselect all

Total 1 Selected 1

- Sum (NOK 1000)

**Year** Mandatory *

Select all | Deselect all

Total 5 Selected 1

- 2019
- 2018
- 2017
- 2016
- 2015

**Region** Mandatory *

Select all | Deselect all

Search

Beginning of word

Total 519 Selected 0

- EAK The whole country
- EAKUO The whole country except Oslo
- 0101 Halden
- 0104 Moss
- 0105 Sarpsborg
- 0106 Fredrikstad

12367: Detailed accounting figur

ssb.no/en/statbank/table/12367/

## Contents

Mandatory *

Select all | Deselect all

Total 1 Selected 1

- Sum (NOK 1000)

## Year

Mandatory *

Select all | Deselect all

Total 5 Selected 1

- 2019
- 2018
- 2017
- 2016
- 2015

## Region

Mandatory *

Select all | Deselect all

Search

Beginning of word

Total 519 Selected 5

- EAK The whole country
- EAKUO The whole country except Oslo
- 0101 Halden
- 0104 Moss
- 0105 Sarpsborg
- 0106 Fredrikstad
- 0111 Hvaler
- 0118 Aremark

Mandatory *

## Accounting scope

Select all | Deselect all

Total 2 Selected 2

- Municipality/county authorities
- Municipality excluded municipal units, inter-mun

## Function

Mandatory *

Primary and secondary schools

Select all | Deselect all

Select values from group

Total 6 Selected 1

- 202 Primary and lower secondary schools
- 213 Adult education at primary and lower seconc
- 215 Out of school care
- 222 School buildings
- 223 School transport
- FGK8b Primary and secondary schools, school bu

## Type

Mandatory *

Select all | Deselect all

Search

Beginning of word

Total 35 Selected 1

- Adjusted gross operating expenditure
- Gross operating expenditure
- Net operating expenditure
- Wages and salaries exclusive sick pay reimburs
- Sick pay reimbursement
- Pension premiums and taxable insurance sche
- Employers' national insurance contributions

**Show table**

Number of selected data cells are: **10**

(maximum number allowed is 300 000)

To the Top

# 12367: Detailed accounting figures in the operational accounts , by region, accounting scope , function, type, contents and year

Show table | About table

Edit and Calculate | Save table as | Table - Layout 2

csv xlsx

+ Hide empty rows

+ Save your query

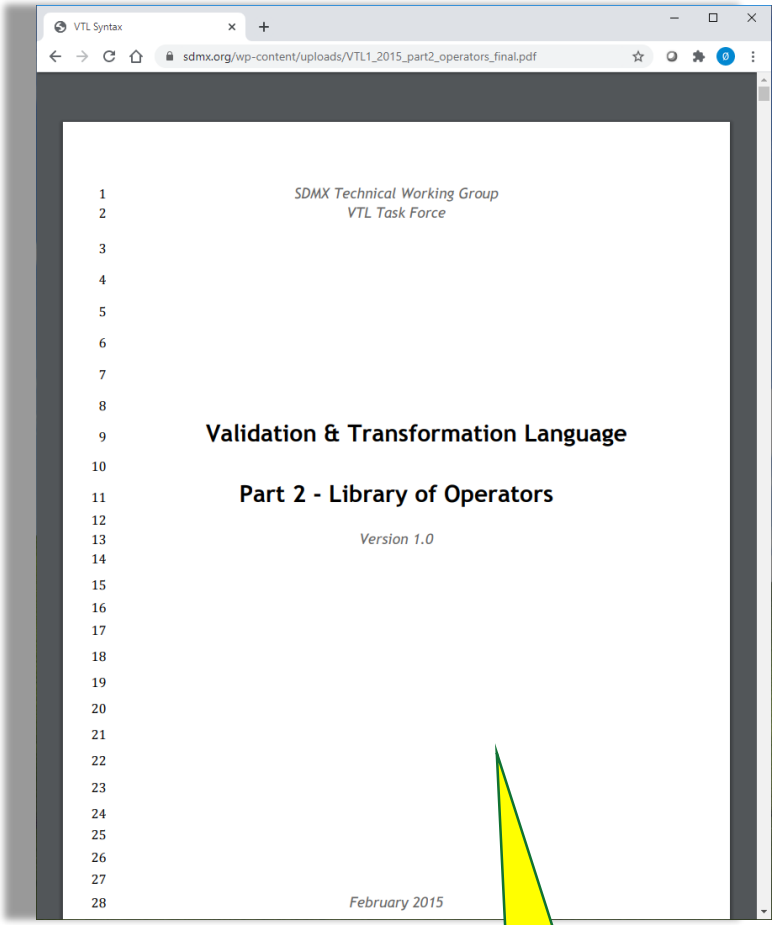
| | | | | Sum (NOK 1000) |
|---|---|---|---|---|
| | | | | 2019 |
| EAK The whole country | Municipality/county authorities | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 3 269 578 |
| | Municipality excluded municipal units, inter-municipal company and cooperation | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 3 251 842 |
| EAKUO The whole country except Oslo | Municipality/county authorities | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 2 927 529 |
| | Municipality excluded municipal units, inter-municipal company and cooperation | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 2 909 793 |
| 0101 Halden | Municipality/county authorities | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 19 042 |
| | Municipality excluded municipal units, inter-municipal company and cooperation | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 19 042 |
| 0104 Moss | Municipality/county authorities | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 13 093 |
| | Municipality excluded municipal units, inter-municipal company and cooperation | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 12 557 |
| 0105 Sarpsborg | Municipality/county authorities | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 33 010 |
| | Municipality excluded municipal units, inter-municipal company and cooperation | FGK8b Primary and secondary schools, school buildings, school transport and adult educaton | Sick pay reimbursement | 33 010 |

Footnotes

The table shows detailed accounting figures for revenues and expenditures in the municipalities.

To the Top

- 6 dimensions
  - **Year**: **1** year. We look at one year at a time (year 2016)
  - **Region**: **462** municipalities
  - **Account category**: **2** categories (not on previous slide)
  - **Accounting scope: 2** scopes
  - **Function**: **218** codes defined by a hierarchy
  - **Type**: **208** codes defined by a hierarchy




- The challenge

  Compute sum values for all 83,795,712 combinations (even if 90% of them are zero) from
  - Input file with sum values for 790,527 combinations
    - Only function and type codes from the lowest hierarchy level
    - Missing combinations in input has zero sum
  - Hierarchical specifications for function and type

*VTL turned out to be too inefficient*

Statistisk sentralbyrå
Statistics Norway

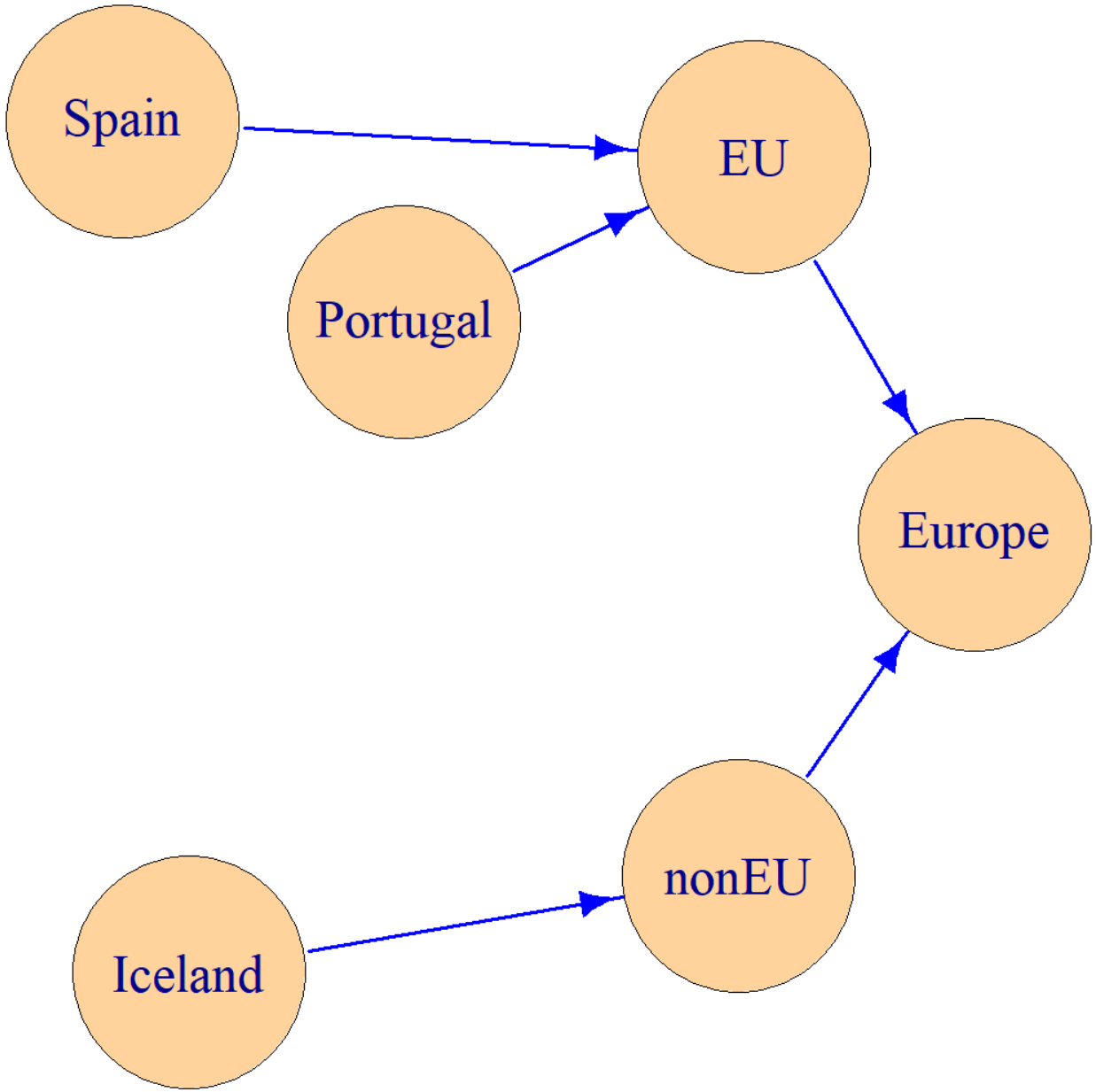
# 80 Million Challenge

- 6 dimensions
  - **Year**: **1** year. We look at one year at a time (year 2016)
  - **Region**: **462** municipalities
  - **Account category**: **2** categories (not on previous slide)
  - **Accounting scope: 2** scopes
  - **Function**: **218** codes defined by a hierarchy
  - **Type**: **208** codes defined by a hierarchy

These hierarchies are messy

- The challenge

  Compute sum values for all 83,795,712 combinations (even if 90% of them are zero) from
  - Input file with sum values for 790,527 combinations
    - Only function and type codes from the lowest hierarchy level
    - Missing combinations in input has zero sum
  - Hierarchical specifications for function and type




## formulas

Europe = EU + nonEU

EU = Portugal + Spain

nonEU = Iceland

## These are more general

- A tree structure not needed
- Sign can be negative
- Free hierarchies

## formulas

Europe = EU + nonEU

EU = Portugal + Spain

nonEU = Iceland

## sdcTable

| levels | codes |
|---|---|
| @ | Europe |
| @@ | EU |
| @@@ | Portugal |
| @@@ | Spain |
| @@ | nonEU |
| @@@ | Iceland |

## tauArgus

EU

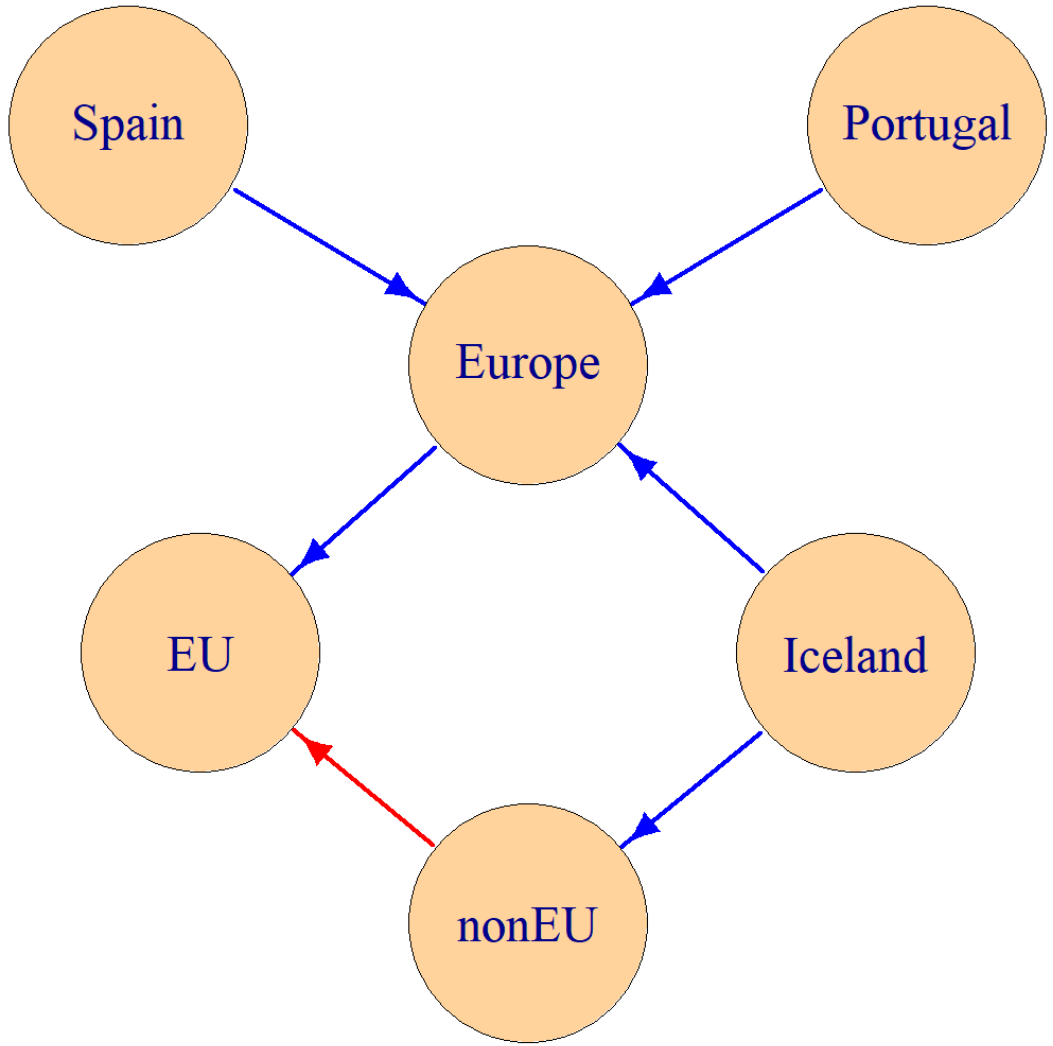
@Portugal

@Spain

nonEU

@Iceland

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| EU | Europe | 1 | 2 |
| Portugal | EU | 1 | 1 |
| Spain | EU | 1 | 1 |
| nonEU | Europe | 1 | 2 |
| Iceland | nonEU | 1 | 1 |




## formulas

Europe = Iceland + Portugal + Spain

nonEU = Iceland

EU = Europe - nonEU

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| Iceland | Europe | 1 | 1 |
| Portugal | Europe | 1 | 1 |
| Spain | Europe | 1 | 1 |
| Iceland | nonEU | 1 | 1 |
| Europe | EU | 1 | 2 |
| nonEU | EU | -1 | 2 |

# Dummy hierarchy matrix →

| | Iceland | Portugal | Spain |
|---|---|---|---|
| EU | 0 | 1 | 1 |
| nonEU | 1 | 0 | 0 |
| Europe | 1 | 1 | 1 |

- First part of the computation process
  - Input codes as columns. Found in mapsFrom level 1.
  - Matrix created by a matrix multiplication for each level
  - The levels can be found automatically
- Both specifications below give same matrix
- Input codes can also be included as rows
  - Parameter `inputInOutput =` **FALSE**

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| EU | Europe | 1 | 2 |
| Portugal | EU | 1 | 1 |
| Spain | EU | 1 | 1 |
| nonEU | Europe | 1 | 2 |
| Iceland | nonEU | 1 | 1 |

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| Iceland | Europe | 1 | 1 |
| Portugal | Europe | 1 | 1 |
| Spain | Europe | 1 | 1 |
| Iceland | nonEU | 1 | 1 |
| Europe | EU | 1 | 2 |
| nonEU | EU | -1 | 2 |

# Dummy hierarchy matrix →

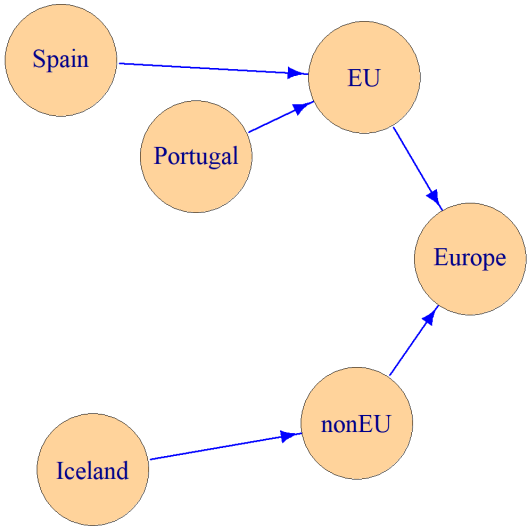
- First part of the computation process
  - Input codes as columns. Found in mapsFrom level 1.
  - Matrix created by a matrix multiplication for each level
  - The levels can be found automatically
- Both specifications below give same matrix
- Input codes can also be included as rows
  - Parameter `inputInOutput = TRUE`

| | Iceland | Portugal | Spain |
|---|---|---|---|
| EU | 0 | 1 | 1 |
| nonEU | 1 | 0 | 0 |
| Europe | 1 | 1 | 1 |
| Iceland | 1 | 0 | 0 |
| Portugal | 0 | 1 | 0 |
| Spain | 0 | 0 | 1 |

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| EU | Europe | 1 | 2 |
| Portugal | EU | 1 | 1 |
| Spain | EU | 1 | 1 |
| nonEU | Europe | 1 | 2 |
| Iceland | nonEU | 1 | 1 |

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| Iceland | Europe | 1 | 1 |
| Portugal | Europe | 1 | 1 |
| Spain | Europe | 1 | 1 |
| Iceland | nonEU | 1 | 1 |
| Europe | EU | 1 | 2 |
| nonEU | EU | -1 | 2 |

# Geographic hierarchy

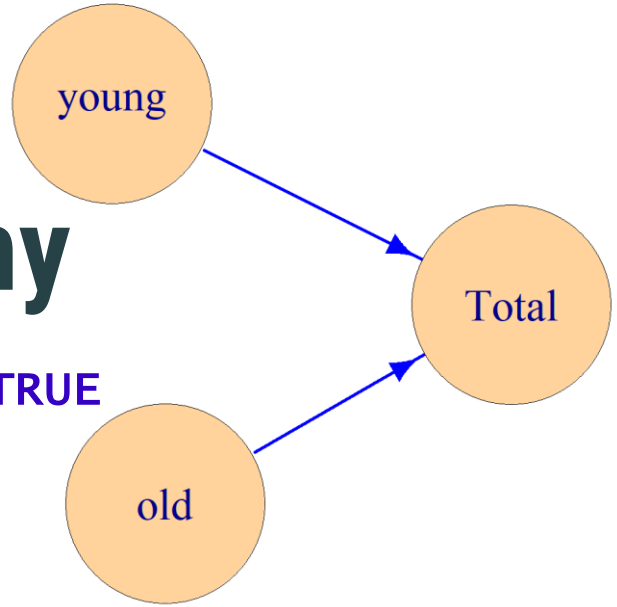



`inputInOutput = FALSE`

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| EU | Europe | 1 | 2 |
| Portugal | EU | 1 | 1 |
| Spain | EU | 1 | 1 |
| nonEU | Europe | 1 | 2 |
| Iceland | nonEU | 1 | 1 |

| | Iceland | Portugal | Spain |
|---|---|---|---|
| EU | 0 | 1 | 1 |
| nonEU | 1 | 0 | 0 |
| Europe | 1 | 1 | 1 |

# Age hierarchy




`inputInOutput = TRUE`

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| young | Total | 1 | 1 |
| old | Total | 1 | 1 |

| | young | old |
|---|---|---|
| old | 0 | 1 |
| young | 1 | 0 |
| Total | 1 | 1 |

# Input to HierarchyCompute

| age | geo | value |
|---|---|---|
| young | Spain | 66.9 |
| young | Iceland | 1.8 |
| young | Portugal | 11.6 |
| old | Spain | 120.3 |
| old | Iceland | 1.5 |
| old | Portugal | 20.2 |

`inputInOutput = FALSE`

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| EU | Europe | 1 | 2 |
| Portugal | EU | 1 | 1 |
| Spain | EU | 1 | 1 |
| nonEU | Europe | 1 | 2 |
| Iceland | nonEU | 1 | 1 |

`inputInOutput = TRUE`

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| young | Total | 1 | 1 |
| old | Total | 1 | 1 |

# Computing aggregates by HierarchyCompute

```
> z
```

| age | geo | value |
|---|---|---|
| young | Spain | 66.9 |
| young | Iceland | 1.8 |
| young | Portugal | 11.6 |
| old | Spain | 120.3 |
| old | Iceland | 1.5 |
| old | Portugal | 20.2 |

```
> HierarchyCompute(z, list(age =
ageHier, geo = geoHier), "value",
inputInOutput = c(TRUE, FALSE))
```

| age | geo | value |
|---|---|---|
| old | EU | 140.5 |
| young | EU | 78.5 |
| Total | EU | 219.0 |
| old | nonEU | 1.5 |
| young | nonEU | 1.8 |
| Total | nonEU | 3.3 |
| old | Europe | 142.0 |
| young | Europe | 80.3 |
| Total | Europe | 222.3 |

# Computing aggregates by `HierarchyCompute`

- Computed by matrix multiplication
- A model matrix (transposed)

```
> z
```

| age | geo | value |
|---|---|---|
| young | Spain | 66.9 |
| young | Iceland | 1.8 |
| young | Portugal | 11.6 |
| old | Spain | 120.3 |
| old | Iceland | 1.5 |
| old | Portugal | 20.2 |

```
> HierarchyCompute(z, list(age = ageHier, geo = geoHier), "value", inputInOutput = c(TRUE, FALSE))
```

| age | geo | value |
|---|---|---|
| old | EU | 140.5 |
| young | EU | 78.5 |
| Total | EU | 219.0 |
| old | nonEU | 1.5 |
| young | nonEU | 1.8 |
| Total | nonEU | 3.3 |
| old | Europe | 142.0 |
| young | Europe | 80.3 |
| Total | Europe | 222.3 |

=

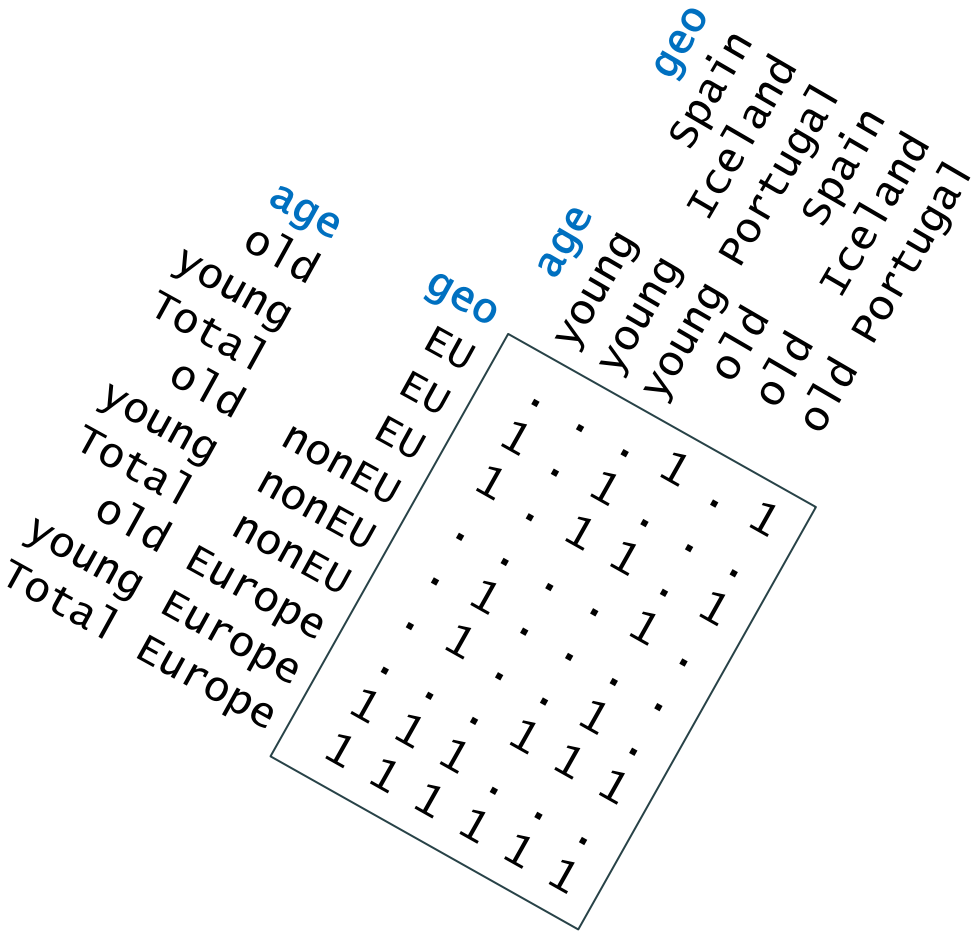
| | | | | | |
|---|---|---|---|---|---|
| . | . | . | 1 | . | 1 |
| 1 | . | 1 | . | . | . |
| 1 | . | 1 | 1 | . | 1 |
| . | . | . | . | 1 | . |
| . | 1 | . | . | . | . |
| . | 1 | . | . | 1 | . |
| . | . | . | 1 | 1 | 1 |
| 1 | 1 | 1 | . | . | . |
| 1 | 1 | 1 | 1 | 1 | 1 |

×

| value |
|---|
| 66.9 |
| 1.8 |
| 11.6 |
| 120.3 |
| 1.5 |
| 20.2 |

Statistisk sentralbyrå
Statistics Norway

# Creating the model matrix for the matrix multiplication




- The final model matrix is the result of crossing a separate model matrix for each hierarchy
- Each is a dummy hierarchy matrix extended/replicated to match data
- Crossing by column-wise Kronecker product
  - Function KhatriRao in the Matrix package

```
          1 . 1 1 . 1
EU        1 . 1 . . .
nonEU     . 1 . . 1 .
Europe    1 1 1 1 1 1

          . . . 1 1 1
old       1 1 1 . . .
young     1 1 1 1 1 1
Total
```

| | Iceland | Portugal | Spain |
|---|---|---|---|
| EU | 0 | 1 | 1 |
| nonEU | 1 | 0 | 0 |
| Europe | 1 | 1 | 1 |

| | young | old |
|---|---|---|
| old | 0 | 1 |
| young | 1 | 0 |
| Total | 1 | 1 |

# HierarchyCompute
# Extended example

```
> x
   age      geo year value
 young    Spain 2014  66.9
 young  Iceland 2014   1.8
 young Portugal 2014  11.6
   old    Spain 2014 120.3
   old  Iceland 2014   1.5
   old Portugal 2014  20.2
 young    Spain 2015  63.4
 young  Iceland 2015   1.9
 young Portugal 2015  14.2
   old    Spain 2015 119.6
   old  Iceland 2015   1.6
   old Portugal 2015  24.3
 young    Spain 2016  69.1
 young  Iceland 2016   1.9
 young Portugal 2016  12.7
   old    Spain 2016 122.1
   old  Iceland 2016   1.9
   old Portugal 2016  25.8
```

```
> HierarchyCompute(x, list(age = ageHier, geo = geoHier,
year = "rowFactor"), "value", inputInOutput = c(TRUE, FALSE))
   age      geo year value
   old       EU 2014 140.5
 young       EU 2014  78.5
 Total       EU 2014 219.0
   old    nonEU 2014   1.5
 young    nonEU 2014   1.8
 Total    nonEU 2014   3.3
   old   Europe 2014 142.0
 young   Europe 2014  80.3
 Total   Europe 2014 222.3
   old       EU 2015 143.9
 young       EU 2015  77.6
 Total       EU 2015 221.5
   old    nonEU 2015   1.6
 young    nonEU 2015   1.9
 Total    nonEU 2015   3.5
   old   Europe 2015 145.5
 young   Europe 2015  79.5
 Total   Europe 2015 225.0
   old       EU 2016 147.9
 young       EU 2016  81.8
 Total       EU 2016 229.7
   old    nonEU 2016   1.9
 young    nonEU 2016   1.9
 Total    nonEU 2016   3.8
   old   Europe 2016 149.8
 young   Europe 2016  83.7
 Total   Europe 2016 233.5
```

- `year` is extra dimension without hierarchy
  - Just a factor
  - Specified by `"rowFactor"`

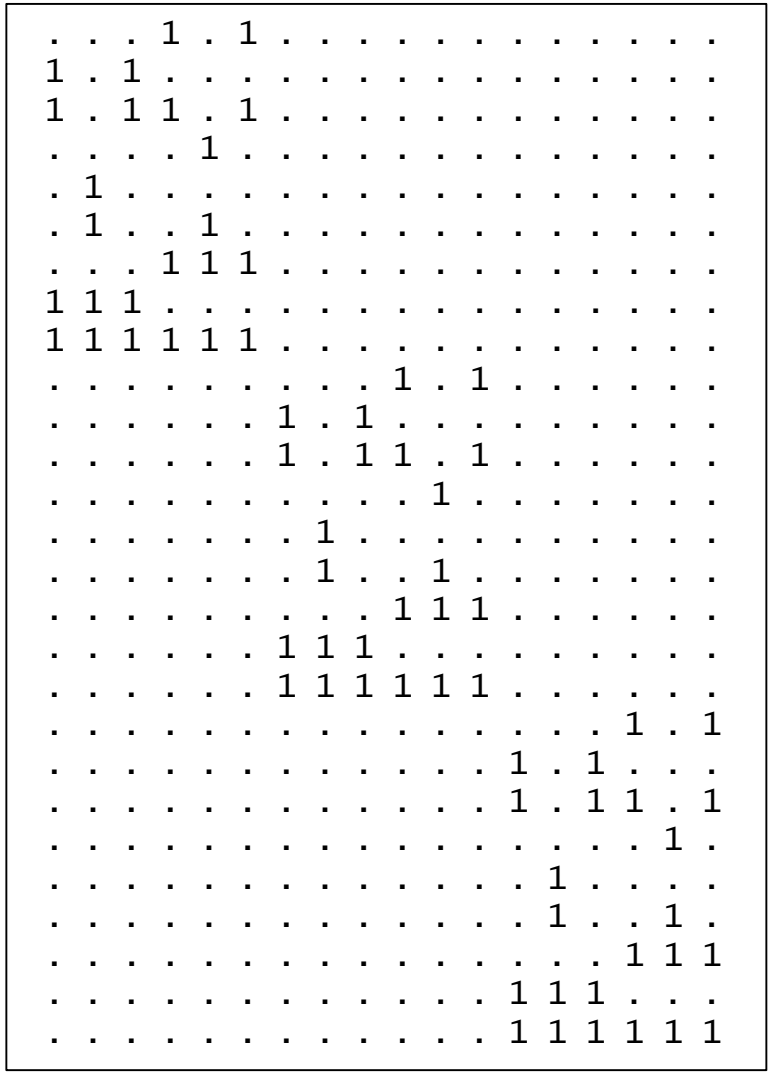
- **Here**
  - **Two hierarchies and one factor**
- **In general**
  - **No limits**

Statistisk sentralbyrå
Statistics Norway

# Computations behind the scenes

| value |
|---|
| 140.5 |
| 78.5 |
| 219.0 |
| 1.5 |
| 1.8 |
| 3.3 |
| 142.0 |
| 80.3 |
| 222.3 |
| 143.9 |
| 77.6 |
| 221.5 |
| 1.6 |
| 1.9 |
| 3.5 |
| 145.5 |
| 79.5 |
| 225.0 |
| 147.9 |
| 81.8 |
| 229.7 |
| 1.9 |
| 1.9 |
| 3.8 |
| 149.8 |
| 83.7 |
| 233.5 |

=



×

| value |
|---|
| 66.9 |
| 1.8 |
| 11.6 |
| 120.3 |
| 1.5 |
| 20.2 |
| 63.4 |
| 1.9 |
| 14.2 |
| 119.6 |
| 1.6 |
| 24.3 |
| 69.1 |
| 1.9 |
| 12.7 |
| 122.1 |
| 1.9 |
| 25.8 |

Statistisk sentralbyrå
Statistics Norway

# HierarchyCompute
# Extended example

```
> x
   age       geo year value
 young     Spain 2014  66.9
 young   Iceland 2014   1.8
 young  Portugal 2014  11.6
   old     Spain 2014 120.3
   old   Iceland 2014   1.5
   old  Portugal 2014  20.2
 young     Spain 2015  63.4
 young   Iceland 2015   1.9
 young  Portugal 2015  14.2
   old     Spain 2015 119.6
   old   Iceland 2015   1.6
   old  Portugal 2015  24.3
 young     Spain 2016  69.1
 young   Iceland 2016   1.9
 young  Portugal 2016  12.7
   old     Spain 2016 122.1
   old   Iceland 2016   1.9
   old  Portugal 2016  25.8
```

```
> HierarchyCompute(x, list(age = ageHier, geo = geoHier,
year = "colFactor" ), "value", inputInOutput = c(TRUE, FALSE))
    age     geo year value
    old      EU 2014 140.5
  young      EU 2014  78.5
  Total      EU 2014 219.0
    old  nonEU 2014   1.5
  young  nonEU 2014   1.8
  Total  nonEU 2014   3.3
    old Europe 2014 142.0
  young Europe 2014  80.3
  Total Europe 2014 222.3
    old      EU 2015 143.9
  young      EU 2015  77.6
  Total      EU 2015 221.5
    old  nonEU 2015   1.6
  young  nonEU 2015   1.9
  Total  nonEU 2015   3.5
    old Europe 2015 145.5
  young Europe 2015  79.5
  Total Europe 2015 225.0
    old      EU 2016 147.9
  young      EU 2016  81.8
  Total      EU 2016 229.7
    old  nonEU 2016   1.9
  young  nonEU 2016   1.9
  Total  nonEU 2016   3.8
    old Europe 2016 149.8
  young Europe 2016  83.7
  Total Europe 2016 233.5
```

- year is extra dimension without hierarchy
  - Just a factor
  - Specified by `"colFactor"`
    - Same output
      - Column order can differ

Change the specification of year from rowFactor to colFactor

Statistisk sentralbyrå
Statistics Norway

# Computations behind the scenes

- Reorganizing to wider format can improve efficiency
- Use `output = "matrixComponents"` to see the matrices

| 2014 | 2015 | 2016 |
|---|---|---|
| 140.5 | 143.9 | 147.9 |
| 78.5 | 77.6 | 81.8 |
| 219.0 | 221.5 | 229.7 |
| 1.5 | 1.6 | 1.9 |
| 1.8 | 1.9 | 1.9 |
| 3.3 | 3.5 | 3.8 |
| 142.0 | 145.5 | 149.8 |
| 80.3 | 79.5 | 83.7 |
| 222.3 | 225.0 | 233.5 |

=

| | | | | | |
|---|---|---|---|---|---|
| . | 1 | 1 | . | . | . |
| . | . | . | . | 1 | 1 |
| . | 1 | 1 | . | 1 | 1 |
| 1 | . | . | . | . | . |
| . | . | . | 1 | . | . |
| 1 | . | . | 1 | . | . |
| 1 | 1 | 1 | . | . | . |
| . | . | . | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

×

| 2014 | 2015 | 2016 |
|---|---|---|
| 1.5 | 1.6 | 1.9 |
| 20.2 | 24.3 | 25.8 |
| 120.3 | 119.6 | 122.1 |
| 1.8 | 1.9 | |
| 11.6 | 14.2 | 12.7 |
| 66.9 | 63.4 | 69.1 |

## Holes may occur

- Because of incomplete input
- Holes filled with zeros

# 80 Million Challenge

- 6 dimensions
  - **Year**: **1** year. We look at one year at a time (year 2016)
  - **Region**: **462** municipalities
  - **Account category**: **2** categories (not on previous slide)
  - **Accounting scope:** **2** scopes
  - **Function**: **218** codes defined by a hierarchy
  - **Type**: **208** codes defined by a hierarchy

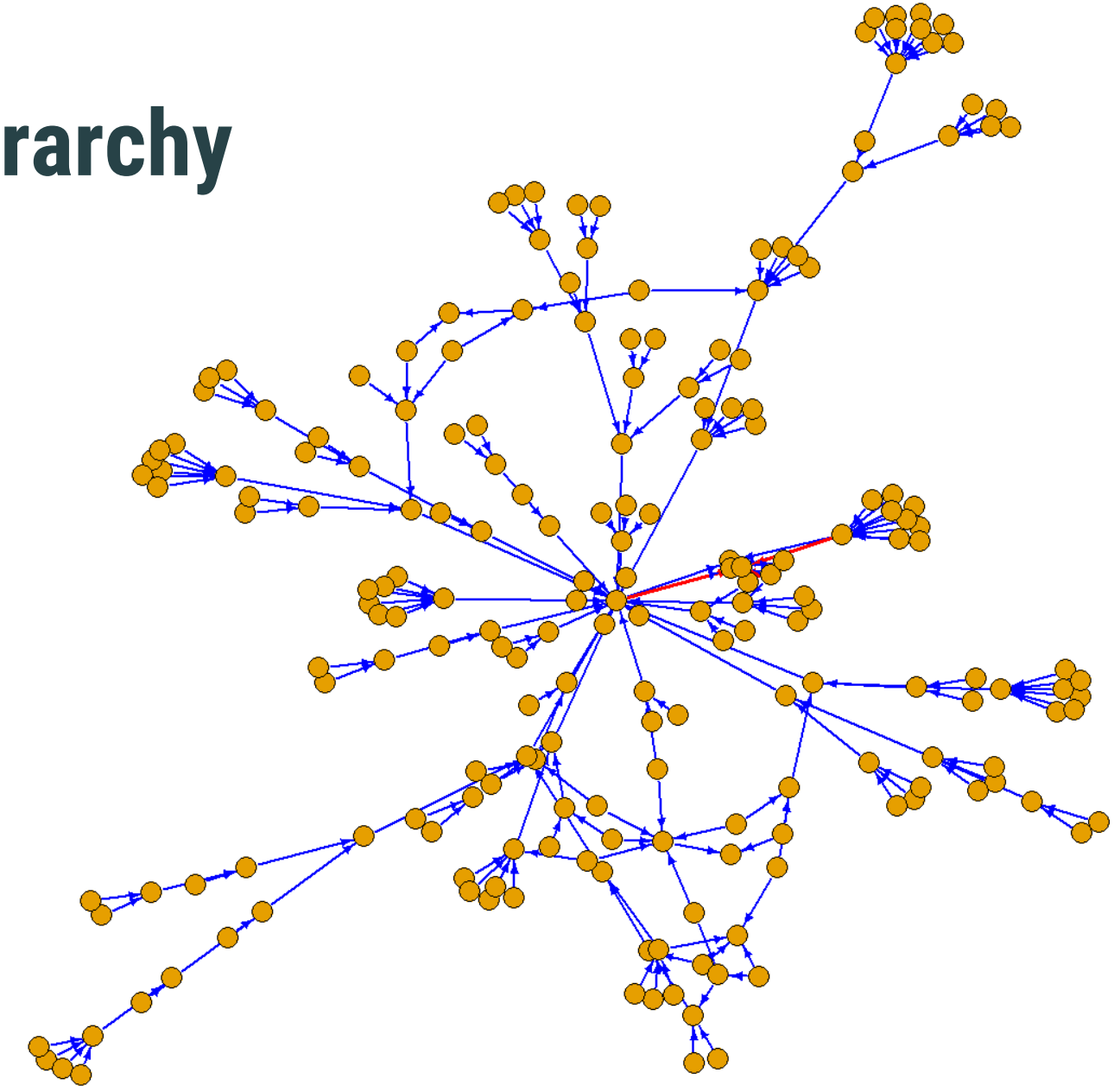
`region = "colFactor"`

These hierarchies are messy

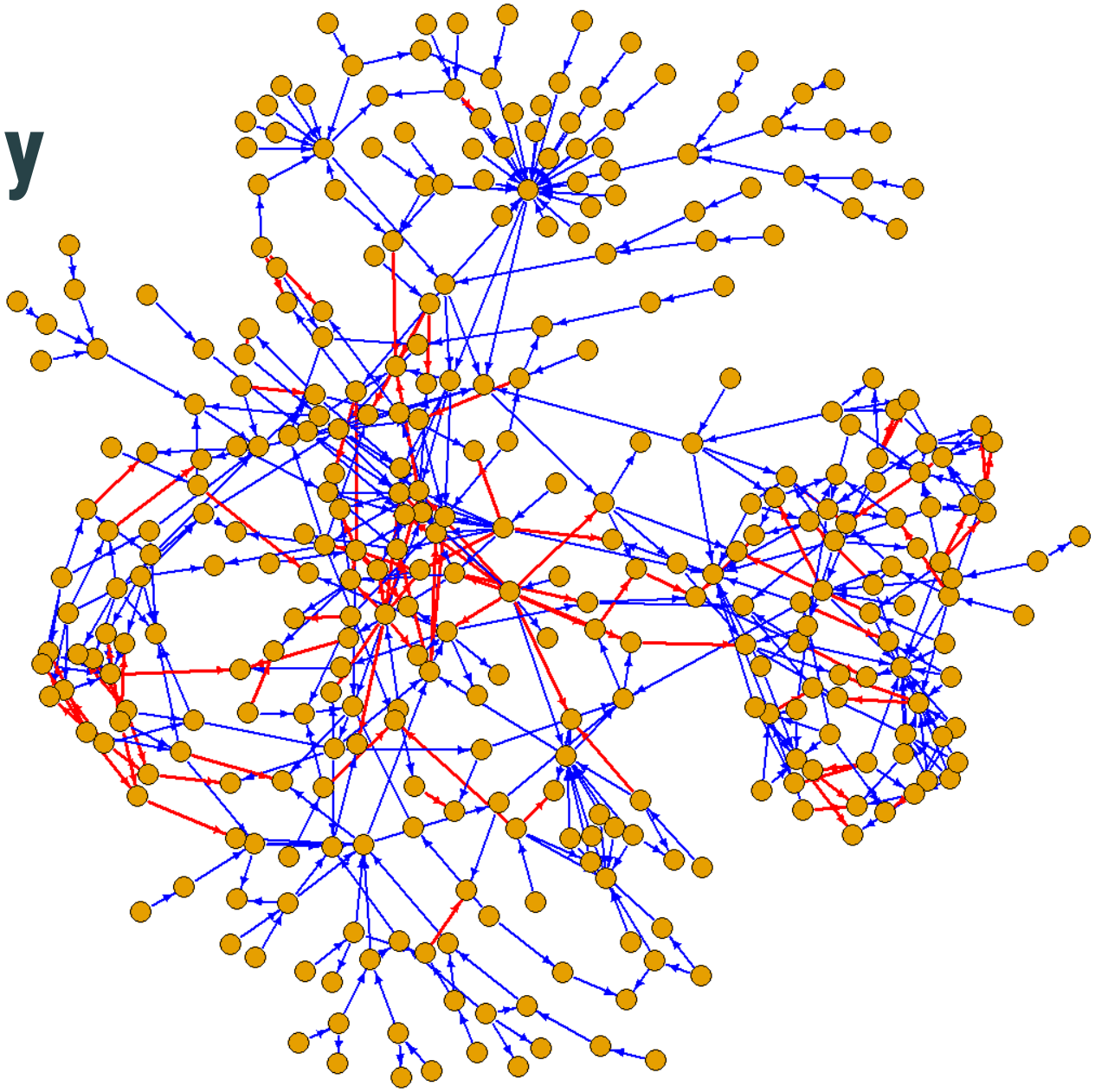
- The challenge

  Compute sum values for all 83,795,712 combinations (even if 90% of them are zero) from
  - Input file with sum values for 790,527 combinations
    - Only function and type codes from the lowest hierarchy level
    - Missing combinations in input has zero sum
  - Hierarchical specifications for function and type

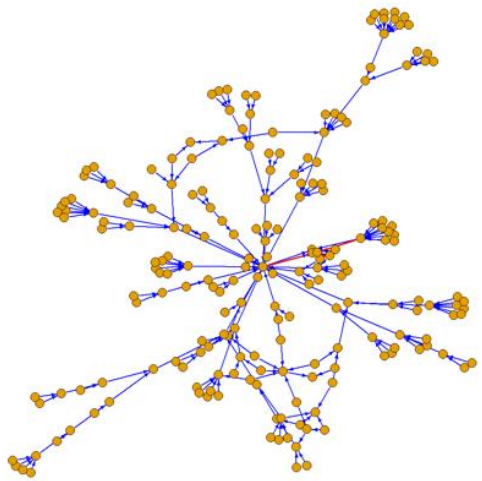
# Function hierarchy

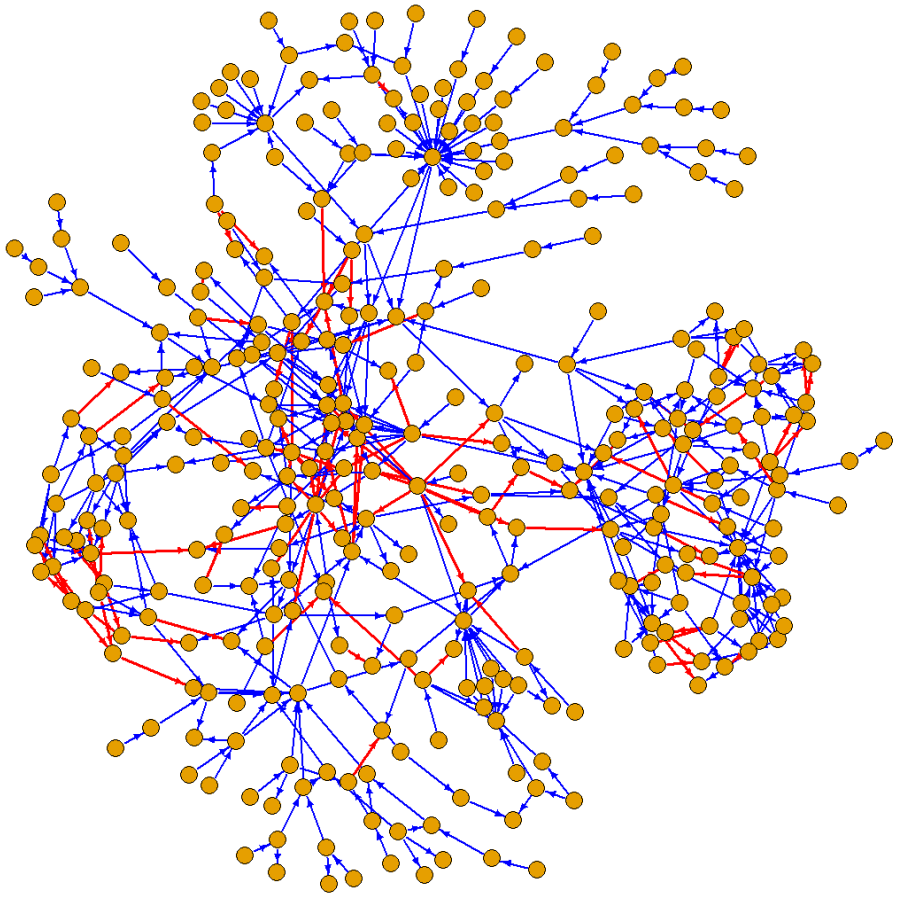


# Type hierarchy



# 80 Million Challenge





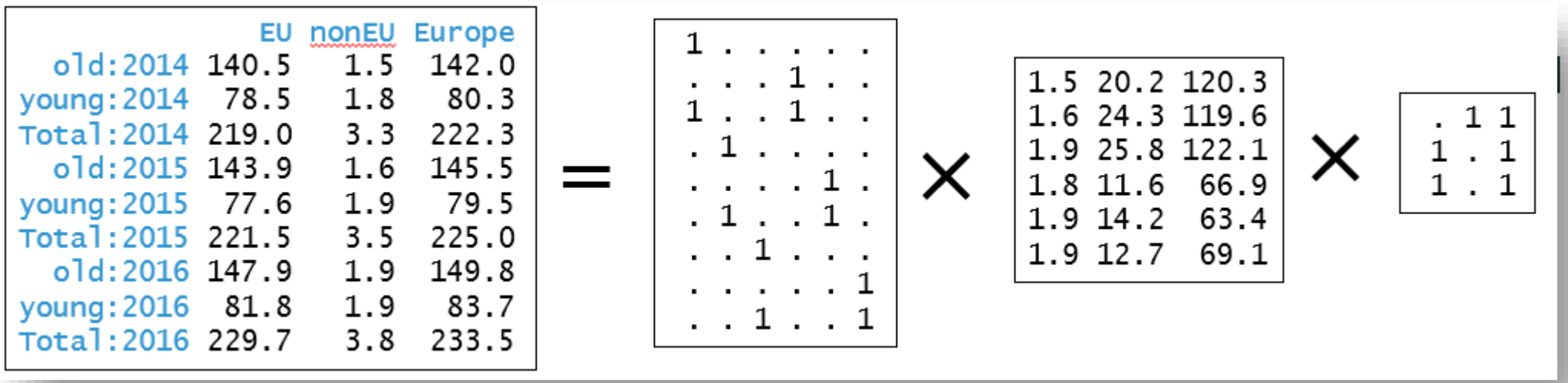
## 20 seconds computing time

- On my old laptop
- Of which 10 seconds to organize the matrix results in a data frame

Statistisk sentralbyrå
Statistics Norway

# Extensions and spin-offs - after mission completed

- Computing selected combinations for output
  - Parameters `rowSelect`, `colSelect`, `select`
- Handling of duplicates
  - Parameter `handleDuplicated`
- “Two-way” computation
  - Parameter `colVar` and function `HierarchyCompute2()`
- Model matrix as primary output and combining with formula
  - Functions `Hierarchies2ModelMatrix()` and `HierarchiesAndFormula2ModelMatrix()`
- Allowing hierarchies specified in several ways
  - Function `AutoHierarchies()`
- Making use of model matrices within statistical disclosure control
  - Package `SmallCountRounding`




  - Parameter `handleDuplicated`

- "Two-way" computation
  - Parameter `colVar` and function `HierarchyCompute2()`
- Model matrix as primary output and combining with formula
  - Functions `Hierarchies2ModelMatrix()` and `HierarchiesAndFormula2ModelMatrix()`
- Allowing hierarchies specified in several ways
  - Function `AutoHierarchies()`
- Making use of model matrices within statistical disclosure control
  - Package `SmallCountRounding`

# Extensions and spin-offs - after mission completed

```
> Hierarchies2ModelMatrix(x, list(age = ageHier, geo = geoHier, year = ""), inputInOutput = c(TRUE, FALSE))
18 x 27 sparse Matrix of class "dgCMatrix"
   [[ suppressing 27 column names 'Total:Europe:2014', 'Total:Europe:2015', 'Total:Europe:2016' ... ]]

 [1,] 1 . . 1 . . . . . . . . . . . . . . 1 . . 1 . . . . .
 [2,] 1 . . . . . 1 . . . . . . . . . . . 1 . . . . . 1 . .
 [3,] 1 . . 1 . . . . . . . . . . . . . . 1 . . 1 . . . . .
 [4,] 1 . . 1 . . . . . 1 . . 1 . . . . . . . . . . . . . .
 [5,] 1 . . . . . 1 . . 1 . . . . . 1 . . . . . . . . . . .
 [6,] 1 . . 1 . . . . . 1 . . 1 . . . . . . . . . . . . . .
 [7,] . 1 . . 1 . . . . . . . . . . . . . . 1 . . 1 . . . .
 [8,] . 1 . . . . . 1 . . . . . . . . . . . 1 . . . . . 1 .
 [9,] . 1 . . 1 . . . . . . . . . . . . . . 1 . . 1 . . . .
[10,] . 1 . . 1 . . . . . 1 . . 1 . . . . . . . . . . . . .
[11,] . 1 . . . . . 1 . . 1 . . . . . 1 . . . . . . . . . .
[12,] . 1 . . 1 . . . . . 1 . . 1 . . . . . . . . . . . . .
[13,] . . 1 . . 1 . . . . . . . . . . . . . . 1 . . 1 . . .
[14,] . . 1 . . . . . 1 . . . . . . . . . . . 1 . . . . . 1
[15,] . . 1 . . 1 . . . . . . . . . . . . . . 1 . . 1 . . .
[16,] . . 1 . . 1 . . . . . 1 . . 1 . . . . . . . . . . . .
[17,] . . 1 . . . .  1 . . 1 . . . . . 1 . . . . . . . . .
[18,] . . 1 . . 1 . . . . . 1 . . 1 . . . . . . . . . . . .
```

# ...ions ...off...

...selecte... for o...

...lect, colSe...

...duplica...

...Duplicate...

...omputa...

... and functio...

...x as pr... ...nd combining with for...

- Functions `Hierarchies2ModelMatrix()` and `HierarchiesAndFormula2ModelMatrix(`

- Allowing hierarchies specified in several ways
  - Function `AutoHierarchies()`
- Making use of model matrices within statistical disclosu...
  - Package `SmallCountRounding`

**sdcTable**

| levels | codes |
|---|---|
| @ | Europe |
| @@ | EU |
| @@@ | Portugal |
| @@@ | Spain |
| @@ | nonEU |
| @@@ | Iceland |

**tauArgus**

EU

@Portugal

@Spain

nonEU

@Iceland

| mapsFrom | mapsTo | sign | level |
|---|---|---|---|
| EU | Europe | 1 | 2 |
| Portugal | EU | 1 | 1 |
| Spain | EU | 1 | 1 |
| nonEU | Europe | 1 | 2 |
| Iceland | nonEU | 1 | 1 |

**formulas**

Europe = EU + nonEU

EU = Portugal + Spain

nonEU = Iceland

```
e6 <- SmallCountData("e6")  # As Table 4
eDimList <- SmallCountData("eDimList")
eDimList
```

```
$geo
  levels    codes
1      @    Total
2     @@       EU
3    @@@ Portugal
4    @@@    Spain
5     @@    nonEU
6    @@@  Iceland

$year
  levels codes
1      @ Total
2     @@  2018
3     @@  2019
```


As seen above, a hierarchy is specified for both variables. `eDimList$geo` is given in Table 5 and `eDimList$year` is a plain hierarchy with total code.

## Four ways to produce Table 6

The four lines below produce the same results with element `publish` as in Table 6. Ordering of rows can be different.

```
PLSrounding(e6, "freq", 5)                                          # a)
PLSrounding(e6, "freq", 5, formula = ~eu * year + geo * year)       # b)
PLSrounding(e6[, -2], "freq", 5, hierarchies = eDimList)            # c)
PLSrounding(e6[, -2], "freq", 5, hierarchies = eDimList, formula = ~geo * year) # d)
```

- In a) and c) the function uses hierarchies for the calculations. In a) the hierarchies are found automatically from the input data.
- In b) the cross-classifications are found from the formula. In addition, hierarchical relations in the input data are analysed so that `geo` and `eu` are combined into the same output column.
- In d), how to cross the hierarchies are defined by a formula.

the hierarchical information is sent as input in another way. One possibility is the table below, where the hierarchy is coded as in the r package sdcTable.

**Table 5**: Hierarchy, `geo`

| levels | codes |
|---|---|
| @ | Total |
| @@ | EU |
| @@@ | Portugal |
| @@@ | Spain |
| @@ | nonEU |
| @@@ | Iceland |

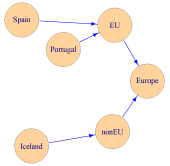
TauArgus coding. More general coding is also possible. See `?AutoHierarchies`

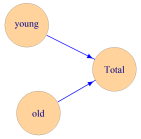
case were all possible combinations (including the inner cells) are to be
example we use 5 as a rounding base. As can be seen below, this output car
al ways. The inner cells are colored according to the rounding.

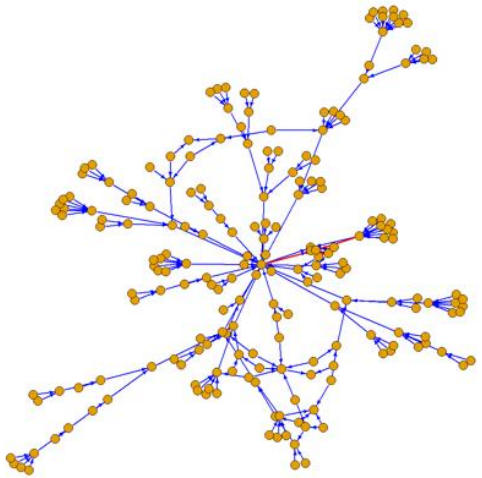
**Table 6**: Ouput data (publish)

| geo | year | original | rounded | difference |
|---|---|---|---|---|
| Total | Total | 24 | 23 | -1 |
| Total | 2018 | 12 | 12 | 0 |
| Total | 2019 | 12 | 11 | -1 |
| EU | Total | 21 | 23 | 2 |
| EU | 2018 | 10 | 12 | 2 |
| EU | 2019 | 11 | 11 | 0 |
| nonEU | Total | 3 | 0 | -3 |
| nonEU | 2018 | 2 | 0 | -2 |
| nonEU | 2019 | 1 | 0 | -1 |
| Iceland | Total | 3 | 0 | -3 |
| Iceland | 2018 | 2 | 0 | -2 |
| Iceland | 2019 | 1 | 0 | -1 |
| Portugal | Total | 8 | 10 | 2 |
| Portugal | 2018 | 3 | 5 | 2 |
| Portugal | 2019 | 5 | 5 | 0 |
| Spain | Total | 13 | 13 | 0 |
| Spain | 2018 | 7 | 7 | 0 |
| Spain | 2019 | 6 | 6 | 0 |

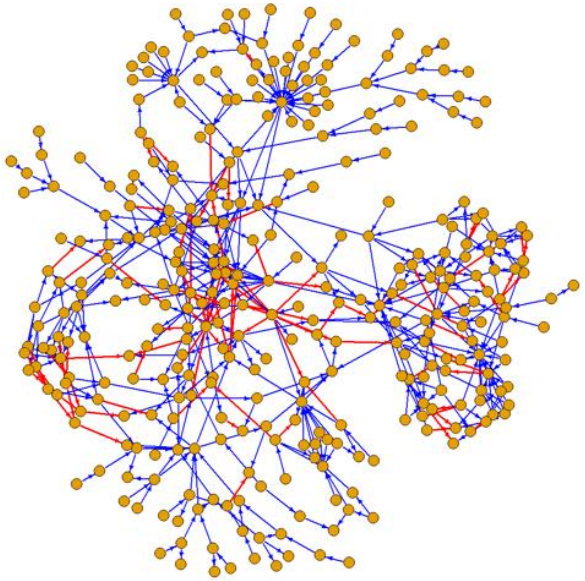
# Conclusion



- Sparse matrix tools in the R package `SSBtools`
  - Depends on R base package Matrix
- Main function `HierarchyCompute()`
- Other functions
  - `Hierarchies2ModelMatrix()`
  - `HierarchiesAndFormula2ModelMatrix()`
  - `AutoHierarchies()`
  - `FindHierarchies()`
  - `.`



Statistisk sentralbyrå
Statistics Norway