PARSEC
An R package for PARtial orders in Socio-EConomics

Alberto Arcagni and Marco Fattore

University of Milano – Bicocca

September 17th and 18th, 2015
Multidimensional poverty evaluation

- Counting approach
- Alkire Foster approach (2011)
- Partial Orders approach (Fattore 2015)
• The package is available on CRAN (The Comprehensive R Archive Network) at the address https://cran.r-project.org/web/packages/parsec/
• It can be installed directly in R with the command install.packages("parsec")
• Once installed, the package can be loaded with the R command library(parsec)
1. Data management
2. Basic poset analysis
3. Poset-based evaluation
4. Alkire Foster approach
Data management
Basic poset analysis
Poset-based evaluation
Alkire Foster approach

**var2prof**

The function allows the user an arbitrary number of ordinal variables, each coded with a different scale, and produces the list of all profiles built on them.

```r
m <- c(v1 = 4, v2 = 2, v3 = 3)
profiles <- var2prof(varlen = m)

varmod <- list(
  v1 = 1:4,
  v2 = c(FALSE, TRUE), # FALSE < TRUE
  v3 = ordered(c("rainy", "cloudy", "sunny"), levels = c("rainy", "cloudy", "sunny"))
)
profiles <- var2prof(varmod, labtype = "progressive")
```
The function extracts all the unique profiles out of a population of statistical units assessed against a set of ordinal variables. It also assigns to each observed profile the correspondent absolute frequency.

```
set.seed(42)
n <- 10
v1 <- as.ordered(c("a", "b", "c", "d"))
v2 <- 1:3
pop <- data.frame(
  v1 = sample(v1, n, replace = TRUE),
  v2 = sample(v2, n, replace = TRUE)
)
profiles <- pop2prof(pop)
```
### pop2prof

```r
> pop
  v1  v2
1  d  2
2  d  3
3  b  3
4  d  1
5  c  2
6  c  3
7  c  3
8  a  1
9  c  2
10 c  2

> profiles
$profiles
   v1 v2
a/1 a  1
b/3 b  3
c/2 c  2
c/3 c  3
d/1 d  1
d/2 d  2
d/3 d  3

$freq
a/1 b/3 c/2 c/3 d/1 d/2 d/3
1 1  3  2  1  1  1
attr(,"class")
[1] "wprof"
```
From the set of profiles, the function generates the square matrix (generally labelled Z) representing the partial order (i.e., the incidence matrix of the corresponding Hasse diagram)

```
profiles <- var2prof(
    varlen = c(2, 2, 2)
)
Z <- getzeta(profiles)
```
The output of \texttt{getzeta} is an object of \texttt{class} \texttt{incidence}.

The classical function \texttt{plot} has a new \texttt{method} to show the corresponding Hasse diagram.

\texttt{plot(Z)}
Relations

• **The functions** `binary`, `reflexivity`, `antisymmetry` and `transitivity` **check** whether the input matrix is binary and represents a reflexive, antisymmetric, or transitive relation.

• `is.preorder` **checks** if the input matrix is binary and represents a preorder (reflexive and antisymmetric relation).

• `is.partialorder` **checks** if the input matrix is binary and represents a preorder (reflexive, antisymmetric and transitive relation).

• If `is.partialorder` **returns** `TRUE` the input matrix can be used as incidence matrix by setting its class to incidence.
Cover matrices

- **incidence2cover:** the function computes the cover matrix associated to the input incidence matrix.
- **cover2incidence:** the function computes the incidence matrix of a poset from the cover matrix.

```r
> incidence2cover(Z)
   111  211  121  221  112  212  122  222
111  FALSE  TRUE  TRUE  FALSE  TRUE  FALSE  FALSE  FALSE
211  FALSE  FALSE  FALSE  TRUE  FALSE  TRUE  FALSE  FALSE
121  FALSE  FALSE  FALSE  TRUE  FALSE  FALSE  TRUE  FALSE
221  FALSE  FALSE  FALSE  TRUE  FALSE  FALSE  FALSE  TRUE
112  FALSE  FALSE  FALSE  FALSE  TRUE  TRUE  FALSE  FALSE
212  FALSE  FALSE  FALSE  FALSE  TRUE  TRUE  FALSE  FALSE
122  FALSE  FALSE  FALSE  FALSE  FALSE  TRUE  TRUE  FALSE
222  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  TRUE  TRUE
attr(,"class") [1] "cover"
```
Locations of the elements

```r
> levels(Z)
111 211 121 221 112 212 122 222
   4  3  3  2  3  2  2  1
> maximal(Z)
   111 211 121 221 112 212 122 222
FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
> minimal(Z)
   111 211 121 221 112 212 122 222
   TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
> heights(Z)
   111 211 121 221 112 212 122 222
   1 2 2 4 2 4 4 8
> depths(Z)
   111 211 121 221 112 212 122 222
   8 4 4 2 4 2 2 1
```
> us <- c("122", "212", "222")
> is.upset(Z, us)
[1] TRUE
> gen.upset(Z, us)
  111  211  121  221  112  212  122  222
FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
>
> ds <- c("112", "211", "111")
> is.downset(Z, ds)
[1] TRUE
> tr <- gen.downset(Z, ds)
> tr
  111  211  121  221  112  212  122  222
FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
The evaluation function

```
> ev <- evaluation(zeta = Z, threshold = tr)
> summary(ev)

weights threshold id. function average rank abs. severity rel. severity abs. wealth gap rel. wealth gap
111  1  FALSE  1.0000000  1.000000  3.8314374  1.00000000  0.0000000  0.0000000
211  1   TRUE  1.0000000  3.335113  1.7425052  0.45894213  0.0000000  0.0000000
112  1   TRUE  1.0000000  5.087501  1.7567638  0.46212742  0.0000000  0.0000000
121  1  FALSE  0.6675827  3.341221  1.8311885  0.42910726  0.4133025  0.0826605
122  1  FALSE  0.0852820  5.634918  0.1705640  0.03412808  2.1532152  0.4963826
221  1  FALSE  0.0785726  3.914158  0.1571454  0.03142908  2.1824272  0.5040762
212  1  FALSE  0.0000000  5.687088  0.0000000  0.00000000  2.0891603  0.5011619
222  1  FALSE  0.0000000  8.000000  0.0000000  0.00000000  4.1685626  1.0000000

poverty gap  = 0.4026198
wealth gap   = 0.5168563
```
Opening Session

Data management
Basic poset analysis
Poset-based evaluation
Alkire Foster approach

plot(ev)

Arcagni and Fattore
Dealing with Complexity in society
Data management
Basic poset analysis
Poset-based evaluation
Alkire Foster approach

plot(ev)

Arcagni and Fattore  Dealing with Complexity in society  16
Opening Session

Data management
Basic poset analysis
Poset-based evaluation
Alkire Foster approach

plot(ev)

Arcagni and Fattore  Dealing with Complexity in society  17
Function **AF**, the arguments are the **profiles**, the vector of cutoffs for each variable and the global cutoff.

The function returns different results, including two objects of class **function**, **rho** and **rho_k**, whose results indicate if a profile identifies a deprived unit or not.

```r
> cutoffs <- c(2, 1, 1)
> global <- 1
> ophi <- AF(profiles, z = cutoffs, k = global)

> ophi$rho(c(1, 2, 1))
[1]  TRUE FALSE FALSE

> ophi$rho_k(c(1, 2, 1))
[1]  TRUE

> ophi$rho(c(2, 2, 1))
[1] FALSE FALSE FALSE

> ophi$rho_k(c(2, 2, 1))
[1] FALSE

> ophi$H # headcount ratio
[1] 0.5
```
Main References


Let $v_1, ..., v_k$ be $k$ ordinal variables.

Each variable can have a different number of degrees $m_1, ..., m_k$.

The vector $p = (p_1, ..., p_k)$ of $k$ scores associated to a statistical unit is called profile.

The set of possible profiles $P$ has cardinality $|P| = m_1 \cdot m_2 \cdot ... \cdot m_k$.

The set $P$ is naturally turned into a partially ordered set $(P, \leq)$ putting

$$p \leq q \iff p_i \leq q_i \ \forall i \in 1, ..., k$$

where $p$ and $q$ are elements of $P$.

An evaluation function $Eval(\cdot)$ is defined to assign a degree of poverty in $[0, 1]$.

The threshold $\tau$ is the minimal set of profiles considered as poor and scored 1 by the evaluation function.

The scores returned by the evaluation function are obtained with the following procedure:

1. Consider the set $LE$ of linear extensions of $P$;
2. For any linear extension $I \in LE$, assign score 1 to all the profiles that are below an element of $\tau$ in $I$;
3. Assign to profiles of $P$ a final poverty score averaging the scores they get on the elements of $LE$. 
Alkire and Foster methodology

- Let $v_1, \ldots, v_k$ be $k$ ordinal variables.
- A set $c_1, \ldots, c_k$ of $k$ cutoffs is exogenously defined, identifying a different threshold for each evaluation dimension.
- A statistical unit scoring a degree $d_i$ lower than $c_i$ is considered as deprived on dimension $v_i$.
- Statistical units are classified as definitely poor if the number of dimensions they are deprived on equals or exceeds an overall cutoff $c$ also to be defined exogenously.