# Conjoint analysis method and its implementation in `conjoint` R package[⋆]

Andrzej Bąk and Tomasz Bartłomowicz

Wrocław University of Economics,
Department of Econometrics and Computer Science
`{andrzej.bak,tomasz.bartlomowicz}@ue.wroc.pl`

**Abstract.** In microeconomics, measurement of consumers' preferences is one of the most important elements of marketing research. It helps to explain the reasons of consumers' decisions. Using some statistical methods it is possible to quantify preferences and answer the question: what product will consumer choose? One of these methods is the traditional conjoint analysis. Because empirical applications of this method depend on computer software, this paper describes an implementation of conjoint analysis method in R computer program, which now is the major noncommercial computer software for statistical and econometric analysis. That is why the purpose of this paper is to present a package `conjoint` developed for R program, which contains an implementation of the traditional conjoint analysis method. Functions of `conjoint` package are presented with examples of applications in empirical analysis of consumer preferences.

**Keywords:** conjoint analysis, R program, consumer preferences

## 1 Introduction

Conjoint analysis originated in mathematical psychology by psychometricians and was developed since the mid-sixties also by researchers in marketing and business ([3]). Conjoint analysis is a statistical method for finding out how consumers make trade-offs and choose among competing products or services. It is also used to predict (simulate) consumers' choices for future products or services. The main assumptions of conjoint analysis are presented among others in the papers [4], [5].

Conjoint analysis in marketing research is used for measurement, analysis and simulation of stated consumer preferences. Respondents are asked in a survey to rank or rate their preference for various profiles of products or services having many different characteristics (attributes and their levels). Collected data is called empirical preferences or empirical total utilities of profiles. Empirical preferences are then decomposed into part-worth utilities of each level of each attribute. In traditional conjoint model the part-worth utilities are estimated using

⋆ Scientific work financed from funds for science in the years 2009-2012 as research project N N111 446037.

ordinary least-squares method with respondents' preferences as the dependent variable and dummy variables for levels of attributes as the independent variables.

It is necessary to use computer software for applying conjoint analysis models in empirical researches. There are many computer programs carried out since $1970^{th}$, for example IBM SPSS Conjoint, Sawtooth Software, SYSTAT Conjoint Analysis, SAS/STAT and online research platforms for various models of conjoint analysis. There are also two CRAN R packages for some specific conjoint models. Package `bayesm` is designed for application on choice-based conjoint data with partial profiles. It uses an MCMC algorithm for hierarchical binary logits. Package `MCML` can be used to fit the results of a conjoint measurement experiment using generalized linear model and maximum likelihood method for estimating perceptual scales. Data for conjoint measurement is usually collected using a psychophysical procedure.

The `conjoint` is an easy to use R package for traditional conjoint analysis based on full-profile collection method and multiple linear regression model with dummy variables. The conjoint model is estimated by least squares method based on `lm()` function from `stats` package.

The aim of this paper is to present a new R package conjoint and explain its functions. The second section presents functions of the package used in typical conjoint procedure ([1], [2], [4]). The third section provides an example of using most important functions of the `conjoint` package in empirical conjoint analysis based on real survey data. In the summary some final remarks are given.

## 2 The `conjoint` package functions

The `conjoint` package is an implementation of traditional conjoint analysis method for R program ([2], [4], [7]). The package is available under the GNU General Public License with free access to source code. Installation is standard for all of R packages. Nowadays authors make available version 1.33 of `conjoint` R package. It is possible to download package from CRAN package repository[1] and home page of Department of Econometrics and Computer Science[2]. To use the package it is necessary to install base R computer program ([6]) and two other packages: `clusterSim` ([8]) for segmentation and `AlgDesign` for generating fractional factorial designs ([9]).

Current version of the `conjoint` package (1.33) has twelve following functions (in alphabetical order): `caBTL()`, `caImportance()`, `caLogit()`, `caMaxUtility()`, `caModel()`, `caPartUtilities()`, `caSegmentation()`, `caTotalUtilities()`, `ca-Utilities()`, `Conjoint()`, `ShowAllSimulations()`, `ShowAllUtilities()`. Required arguments of these functions with short characteristic are presented in Table 1 (in order of typical conjoint analysis procedure).

First two functions are used to estimate part-worth utilities of attributes levels and theoretical total profiles utilities. Function `caPartUtilities()` cal-

[1] URL: `http://cran.r-project.org/web/packages/conjoint`.
[2] URL: `http://keii.ue.wroc.pl/conjoint`.

**Table 1.** Functions of conjoint R package with required arguments

| Function's characteristic |
|---|
| `caPartUtilities(y, x, z)` – function calculates matrix of individual levels utilities for respondents (with intercept on first place) |
| `caTotalUtilities(y, x)` – function calculates matrix of theoretical total utilities for respondents (for $n$ profiles and all respondents) |
| `caImportance(y, x)` – function calculates importance of all attributes. The sum of importance should be 100% |
| `caUtilities(y, x, z)` – function calculates utilities of attribute's levels |
| `Conjoint(y, x, z)` – function returns part-worth utilities of levels (model parameters for whole sample), vector of percentage attributes' importance, sum of them and corresponding charts (barplots). The sum of importance should be 100% |
| `ShowAllUtilities(y, x, z)` – function returns basic matrix of part-worth utilities (with the intercept), matrix of total utilities (for $n$ profiles and all respondents) and vector of percentage attributes' importance, with sum of them. The sum of importance should be 100% |
| `caBTL(sym, y, x)` – function estimates participation of simulation profiles using probabilistic model BTL (Bradley-Terry-Luce). The sum of participation should be 100% |
| `caLogit(sym, y, x)` – function estimates participation of simulation profiles using logit model. The sum of participation should be 100% |
| `caMaxUtility(sym, y, x)` – function estimates participation of simulation profiles using model of maximum utility ("first position"). The sum of participation should be 100% |
| `ShowAllSimulations(sym, y, x)` – function returns 3 vectors of percentage participations using maximum utility, BTL and logit models. The sum of importance for every vector should be 100% |
| `caModel(y, x)` – function estimates parameters of conjoint analysis model |
| `caSegmentation(y, x, c=3)` – function divides respondents into three or $n$ clusters using $k$-means method – function `kmeans()` from package `stats` |

| Function's arguments | |
|---|---|
| `y` | matrix (vector) of empirical preferences |
| `x` | matrix of profiles |
| `z` | vector with level names |
| `sym` | matrix of simulation profiles |
| `c` | number of clusters (optional), default value: `c=3` |

Source: own work out.

culates matrix of individual part-worth utilities of levels for all respondents. Function returns matrix of part-worth utilities (i.e. parameters of regression model) for all dummy variables (with intercept). Function `caTotalUtilities()` calculates matrix of theoretical total utilities for respondents. Function returns matrix of total utilities for all profiles and all respondents. In case of the first of these functions the additional argument with levels names is used to name the columns of matrix of utility.

Second two functions are: `caImportance()` and `caUtilities()`. First of them calculates importance of all attributes. Function returns vector of percentage attributes' importance and corresponding bar graph. The sum of importance should be 100%. Second function calculates part-worth utilities of attribute's levels for whole sample of respondents. Function returns vector of utilities (length of this vector is equal to number of all levels).

Next two functions use a combination of other functions to work with. It means, that each of these functions returns a set of some results. Function `Conjoint()` is a link of following functions: `caPartUtilities()`, `caUtilities()` and `caImportance()`. Therefore it sums up the main results of conjoint analysis. Function returns part-worth utilities of levels (model parameters for whole sample), vector of percentage attributes' importance, sum of them and corresponding bar graph. The sum of importance should be 100%. Function `ShowAllUtilities()` returns matrix of part-worth utilities (with the intercept), matrix of total utilities (for all profiles and all respondents) and vector of percentage attributes' importance, with sum of them. The sum of importance should be 100%.

Using `conjoint` package, we can estimate participation of simulation profiles. In the group of simulation analysis functions there are three functions. Each of these functions estimates participation of simulation profiles using different model. It can be the model of maximum utility, which is also known as "model of first position" (`caMaxUtility()`), BTL model (Bradley-Terry-Luce) (`caBTL()`) or logit model (`caLogit()`). There is also `ShowAllUtilities()` function for estimating all models at the same time.

Last two functions are connected with model estimation and segmentation of respondents. If we want to estimate parameters of conjoint analysis model we should use function `caModel()`. Because it returns vector of estimated parameters of traditional conjoint analysis model, there are needed only two parameters: vector of preferences and matrix of profiles. Second function `caSegmentation()` divides respondents into three or $n$ clusters using $k$-means clustering method. Function by default takes three clusters when there are only two attributes used (matrix of preferences and matrix of profiles). Otherwise this function divides respondents into $n$ clusters.

## 3 The `conjoint` package application

To understand and maybe employ these functions and whole `conjoint` package we propose example with the study on preferences of tea consumers carried out in 2004 on a sample group of students of Wrocław University of Economics[3]. The main aim of the study was to identify determines of consumers' choice of specific brands and types of tea. Set of tea-choosing data is relevant to check the `conjoint` package.

In the example there are four attributes of tea – three attributes with three levels and one attribute with two levels. The attributes of tea are the following (with level names): price (low, average, high), variety (black, green, red), kind (bags, granulated, leaf) and aroma (yes, no).

The outcome was a fractional factorial design with 13 profiles of tea[4]. Because it isn't necessary to use all combination of profiles the full factorial design with fifty four profiles was reduced in R program by function `gen.factorial()` from `AlgDesign` package ([9]). The fractional factorial design with scores of five respondents is presented in Table 2.

The collected data are used for parameter estimation with model reflecting relations between profile evaluation and the values of attributes that characterize them. With empirical studies using traditional conjoint analysis on stated

[3] Data were collected by Małgorzata Baran.

[4] It's possible to check set of tea-choosing data in `conjoint` package. More details are in reference manual of the package.

**Table 2.** Profiles of tea with exemplary respondents' scores

| Profile | Attributes | | | | Scores | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Price | Variety | Kind | Aroma | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ |
| 1 | high | black | bags | yes | 8 | 0 | 4 | 6 | 5 |
| 2 | low | green | bags | yes | 1 | 10 | 10 | 7 | 1 |
| 3 | average | green | granulated | yes | 1 | 3 | 3 | 4 | 7 |
| 4 | average | black | leaf | yes | 3 | 5 | 5 | 9 | 8 |
| 5 | high | red | leaf | yes | 9 | 1 | 4 | 6 | 6 |
| 6 | average | black | bags | no | 2 | 4 | 1 | 3 | 10 |
| 7 | high | green | bags | no | 7 | 8 | 2 | 7 | 7 |
| 8 | average | red | bags | no | 2 | 6 | 0 | 4 | 10 |
| 9 | high | black | granulated | no | 2 | 2 | 0 | 8 | 6 |
| 10 | low | red | granulated | no | 2 | 9 | 1 | 5 | 6 |
| 11 | low | black | leaf | no | 2 | 7 | 8 | 2 | 6 |
| 12 | average | green | leaf | no | 3 | 5 | 9 | 10 | 10 |
| 13 | high | green | leaf | no | 4 | 2 | 7 | 9 | 7 |

Source: own work out.

consumer preferences, the most frequently used model is the following regression model with dummy variables:

$$\hat{U}_s = b_{0s} + b_{1s}X_{1s} + b_{2s}X_{2s} + ... + b_{7s}X_{7s} \tag{1}$$

where: $\hat{U}_s$ – theoretical total utilities perceived by $s$-th respondent, $b_{0s}$ – intercept, $b_{1s}, \ldots, b_{7s}$ – parameters of regression model (part-worth utilities of attributes levels), $s$ – number of respondent ($s = 1, 2, \ldots, 100$), $X_1, \ldots, X_7$ – dummy variables.

Using function caModel() from conjoint package it is possible to calculate this model for a individual respondent. For example, in case of first respondent there are following parameters of conjoint analysis model:

```
> library(conjoint)
> data(tea)
> caModel(y=tprefm[1,], x=tprof)

Call:
lm(formula = frml)

Residuals:
     1      2      3      4      5      6      7      8      9     10     11     12     13
 1,134 -1,489  0,310 -0,265  0,310  0,193  1,593 -1,431 -1,431  1,120  0,369  1,193 -1,606

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          3,3937     0,5439   6,240  0,00155 **
factor(x$price)1    -1,5172     0,7944  -1,910  0,11440
factor(x$price)2    -1,1414     0,6889  -1,657  0,15844
factor(x$variety)1  -0,4747     0,6889  -0,689  0,52141
factor(x$variety)2  -0,6747     0,6889  -0,979  0,37234
factor(x$kind)1      0,6586     0,6889   0,956  0,38293
factor(x$kind)2     -1,5172     0,7944  -1,910  0,11440
factor(x$aroma)1     0,6293     0,5093   1,236  0,27150
---
Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Residual standard error: 1,78 on 5 degrees of freedom
Multiple R-squared: 0.8184,     Adjusted R-squared: 0.5642
F-statistic:  3.22 on 7 and 5 DF,  p-value: 0,1082
```

It means, that function `caModel()` returns, among other things, vector of estimated para-meters of traditional conjoint analysis model for individual respondent for all variables. But if we want to estimate parameters of conjoint model with utilities of attribute levels, the most useful will be function `caUtilities()`, which returns vector of utilities for all attribute levels with intercept on first place:

```
> library(conjoint)
> data(tea)
> caUtilities(y=tprefm[1,], x=tprof, z=tlevn)
 [1]  3,3936782 -1,5172414 -1,1413793  2,6586207 -0,4747126 -0,6747126
 [7]  1,1494253  0,6586207 -1,5172414  0,8586207  0,6293103 -0,6293103
```

The same result, but for all respondents, we can get using `caPartUtilities()` function. This function from `conjoint` package calculates and returns matrix of individual part-worth utilities (parameters of regression) for all artificial variables (with intercept on first place) for every respondent. In the example there are the following part-worth utilities for five respondents:

```
> library(conjoint)
> data(tea)
> caPartUtilities(y=tpref, x=tprof, z=tlevn)
      intercept    low medium   high  black  green    red   bags granulated  leafy    yes     no
 [1,]     3,394 -1,517 -1,141  2,659 -0,475 -0,675  1,149  0,659     -1,517  0,859  0,629 -0,629
 [2,]     5,049  3,391 -0,695 -2,695 -1,029  0,971  0,057  1,105     -0,609 -0,495 -0,681  0,681
 [3,]     4,029  2,563 -1,182 -1,382 -0,248  2,352 -2,103 -0,382     -2,437  2,818  0,776 -0,776
 [4,]     5,856 -1,149 -0,025  1,175 -0,492  1,308 -0,816 -0,825     -0,149  0,975  0,121 -0,121
 [5,]     6,250 -2,333  2,567 -0,233 -0,033 -0,633  0,667 -0,233     -0,333  0,567 -1,250  1,250
```

For example, for first respondent ($s = 1$) the parameters of model for kind of tea are following: $b_5 = 0,659$(“bags”), $b_6 = -1,517$(“granulated”), $-(b_5 + b_6) = 0,859$ (“leafy”). It means, that for this respondent (but not for all of them) the most attractive is leafy tea, then tea in bags, and granulated tea is rather not desired.

If we want to estimate the parameters for all (sample consisted of one hundred) respondents, we should be using function `Conjoint()` from `conjoint` package. For example of tea the results are the following:

```
> library(conjoint)
> data(tea)
> Conjoint(y=tpref, x=tprof, z=tlevn)
[1] "Part worths (utilities) of levels (model parameters for whole sample):"
       levnms    utls
1   intercept  3,5534
2         low  0,2402
3      medium -0,1431
4        high -0,0971
5       black  0,6149
6       green  0,0349
7         red -0,6498
8        bags  0,1369
9  granulated -0,8898
10      leafy  0,7529
11        yes  0,4108
12         no -0,4108
[1] "Average importance of factors (attributes):"
[1] 24,76 32,22 27,15 15,88
[1] Sum of average importance:  100,01
[1] "Chart of average factors importance"
```

As we can see, besides part-worth utilities (parameters of regression model) for all variables and for all respondents, function `Conjoint()` returns also importance of factors. If we want to know only the importance of factors, we should use function `caImportance()` from `conjoint` package:

```
> library(conjoint)
> data(tea)
> caImportance(y=tpref, x=tprof)
[1] 24,76 32,22 27,15 15,88
```

These results mean that the most attractive is a black, leaf tea with low price and aroma. Besides that, it is also possible to calculate average importance of attributes. In this example most important are by turns: color of tea (32,22%), its type (27,15%) and price of tea (24,76%). The minimum importance has aroma (15,88%). Relative importance of attributes (factors) is presented on Figure 1.

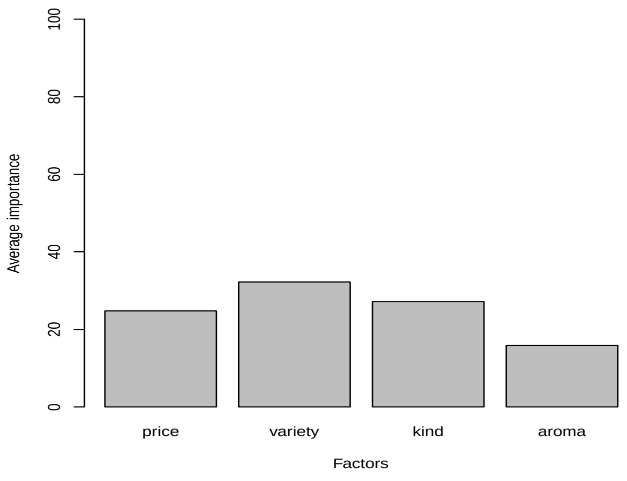

**Fig. 1.** Relative importance of attributes
Source: own work out using `conjoint` R package

If we additionally want to compare respondents' preferences (empirical total utilities) with theoretical total utilities, the next useful function of `conjoint` package is `caTotalUtilities()`. Theoretical total utilities for five first respondents:

```
> library(conjoint)
> data(tea)
> caTotalUtilities(y=tpref, x=tprof)
       [,1]  [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]  [,11]  [,12]  [,13]
 [1,] 6,866 2,490  0,690  3,266  8,690  1,807  5,407  3,431  3,431  0,879  1,631  1,807  5,607
 [2,] 1,748 9,834  4,034  2,148  1,234  5,110  5,110  6,197  1,397  8,569  7,597  5,510  3,510
 [3,] 2,793 9,338  3,538  6,193  4,138  1,441  3,841 -0,414 -0,814  1,276  8,386  7,241  7,041
 [4,]  5,834 5,310  7,110  6,434  7,310  4,393  7,393  4,069  6,269  3,621  5,069  7,993  9,193
 [5,]  4,500 1,800  6,600  8,100  6,000  9,800  6,400 10,500  6,900  5,500  5,700 10,000  7,200
```

As we can suppose, it is necessary to compare these results with empirical data (respondents' scores for all thirteen profiles given in Table 2). Using simple R commands we can see empirical preferences for the same five first respondents:

```
> library(conjoint)
> data(tea)
> colnames(tprefm)<-cbind(paste("prof",1:13,sep=""))
> tprefm[1:5,]
  prof1 prof2 prof3 prof4 prof5 prof6 prof7 prof8 prof9 prof10 prof11 prof12 prof13
1     8     1     1     3     9     2     7     2     2      2      2      3      4
2     0    10     3     5     1     4     8     6     2      9      7      5      2
3     4    10     3     5     4     1     2     0     0      1      8      9      7
4     6     7     4     9     6     3     7     4     8      5      2     10      9
5     5     1     7     8     6    10     7    10     6      6      6     10      7
```

If we want, we can calculate all these utilities (matrix of part-worth utilities with the intercept, matrix of total utilities for n profiles and all respondents, vector of utilities for attribute levels and vector of percentage attributes' importance with sum of importance) using only one function of `conjoint` package – function `ShowAllUtilities()`.

If it is needed, we can also rate respondents on three or $n$ clusters using $k$-means method. Necessary for it comes function `caSegmentation()`:

```
> library(conjoint)
> data(tea)
> caSegmentation(y=tpref, x=tprof, c=3)
K-means clustering with 3 clusters of sizes 29, 31, 40

Cluster means:
      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
1 4,808000 5,070759 2,767310 7,132138 6,843172 2,649483 3,656379 1,539724
2 3,330226 5,582000 5,214258 4,207645 3,859419 4,740871 5,173129 5,334710
3 5,480275 2,938100 1,368100 4,540275 1,973100 3,782900 1,382900 0,965750
      [,9]    [,10]    [,11]    [,12]    [,13]
1 2,063862 1,030862 6,691448 5,980517 6,801207
2 3,366968 4,838194 4,612129 6,050548 5,108613
3 2,820750 0,111225 3,450750 0,442900 0,692900

Clustering vector:
  [1] 1 2 1 2 2 3 1 2 1 1 1 1 3 3 3 3 2 3 2 3 3 1 3 2 2 1 2 2 2 2 3 1 2 1 1 1 1
 [38] 3 3 3 3 2 3 2 3 1 1 3 3 3 1 3 3 3 2 1 3 2 3 2 3 3 1 2 2 1 3 3 3 2 1 3 1 2
 [75] 1 2 2 3 1 1 2 2 2 1 3 3 3 3 2 3 2 3 2 3 3 1 3 2 1 1

Within cluster sum of squares by cluster:
[1] 1540,596 2316,512 1605,654
 (between_SS / total_SS =  41.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"
```

If we have some simulation profiles of tea, we can estimate participation of simulation pro-files using probabilistic model BTL (Bradley-Terry-Luce) – function `caBTL()`, logit model – function `caLogit()` or model of maximum utility "first position") – function `caMaxUtility()`. But instead of these three functions, it is possible to use only one function from `conjoint` package – function `ShowAllSimulations()`:

```
> ShowAllSimulations(sym=tsimp, y=tpref, x=tprof)
  TotalUtility MaxUtility BTLmodel LogitModel
1         2,19         11    18,85      13,04
2         3,69         28    28,96      34,84
3         3,10         26    30,14      35,79
4         3,59         35    22,06      16,33
```

Function `ShowAllSimulations()` returns three vectors of percentage participations using maximum utility, BTL and logit models. The syntax of this function is the same like in functions: `caBTL()`, `caLogit()`, `caMaxUtility()`. The sum of importance for every vector should be 100%.

This and many other examples of using each `conjoint` functions are placed in reference manual of this package.

## 4 Conclusions

The presented in this article `conjoint` package is a new package for R program designed mostly for statisticians, econometricians, economists and students of economics, who are interested in the research of stated preferences of consumers. As R environment and many other packages for R program `conjoint` package is available for free and useful as much as commercial specialized computer software. Nowadays, it contains easy to use functions of traditional conjoint analysis method. In future work, it is intended to include selected discrete choice methods in the `conjoint` R package.

## References

1. Bąk A., *Dekompozycyjne metody pomiaru preferencji w badaniach marketingowych [Decompositional preference measurement methods in marketing research].* Wyd. AE we Wrocławiu, Wrocław (2004).
2. Bąk A., *Analiza Conjoint [Conjoint Analysis]*, [In:] Walesiak M., Gatnar E. (Eds.), *Statystyczna analiza danych z wykorzystaniem programu R [Statistical Data Analysis using R]*, Wydawnictwo Naukowe PWN, Warszawa (2009).
3. Green P.E., Krieger A.M., Wind Y., *Thirty Years of Conjoint Analysis: Reflections and Prospects*, "Interfaces", May-June, 31 (3) supplement, S56-S73 (2001).
4. Green P.E., Srinivasan V., *Conjoint Analysis in Consumer Research: Issues and Outlook*, Journal of Consumer Research, September, 5, 103-123, (1978).
5. Gustafsson A., Herrmann A., Huber F. (ed.), *Conjoint Measurement. Methods and Application*, Fourth edition. Springer, Berlin, Heidelberg (2007).
6. R Development Core Team, *R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing*, URL: `http://cran.r-project.org/` (2011).
7. *SPSS 6.1 Categories*, SPSS Inc., Chicago (1994).
8. Walesiak M., Dudek A., *`clusterSim` package*, `http://www.R-project.org` (2011).
9. Wheeler R.E., *Package `AlgDesign`. Algorithmic Experimental Design*, `http://www.R-project.org` (2010).