

Tomasz Bartłomowicz, Andrzej Bąk

Uniwersytet Ekonomiczny we Wrocławiu

CZĄSTKOWY UKŁAD CZYNNIKOWY I JEGO IMPLEMENTACJA W MODULE CONJOINT PROGRAMU R¹

Streszczenie: W artykule zaprezentowano rozwinięcie pakietu `conjoint` programu R o funkcję: `caFactorialDesign()` umożliwiającą generowanie kompletnego lub częściowego układu czynnikowego oraz funkcję `caEncodedDesign()` umożliwiającą kodowanie uzyskanych eksperymentów. Podstawą działania obu funkcji jest implementacja w pakiecie `conjoint` funkcji pakietu `AlgDesign`. W proponowanym rozwiązaniu zakłada się, że do uzyskania odpowiedniego układu czynnikowego wystarczające powinny być dane dotyczące liczby branych pod uwagę zmiennych oraz ich poziomów z ich nazwami włącznie.

Słowa kluczowe: częściowy układ czynnikowy, *conjoint analysis*, R program.

1. Wstęp

W badaniach marketingowych, szczególnie w pomiarze preferencji nabywców, wykorzystywane są profile produktów, często hipotetycznych, które prezentowane są potencjalnym nabywcom do oceny. Ze względu na zakres niektórych eksperymentów, tj. liczbę uwzględnianych w badaniach zmiennych oraz ich poziomów, całkowita liczba możliwych do uzyskania profilów w postaci tzw. pełnego układu czynnikowego niejednokrotnie jest zbyt duża, aby można ją było wykorzystać w praktycznych zastosowaniach. W tej sytuacji, z wykorzystaniem narzędzi statystycznych, pod uwagę bierze się ograniczony zbiór profilów w postaci tzw. niepełnego układu czynnikowego, pod warunkiem zachowania wymaganych reguł statystycznych, które dotyczą m.in. istotnych źródeł zmienności i minimalizacji popełnianych błędów.

W artykule przedstawiono rozwinięcie pakietu `conjoint` [Bąk, Bartłomowicz 2012] programu R o funkcje umożliwiające generowanie pełnych i częściowych układów czynnikowych. W proponowanym rozwiązaniu wykorzystuje się utworzony w tym celu i udostępniany w Internecie pakiet `AlgDesign` [Wheeler 2012]. Implementacja funkcji tego pakietu w postaci odpowiednich funkcji pakietu `con-`

¹ Praca naukowa finansowana ze środków na naukę w latach 2009-2012 jako projekt badawczy nr N N111 446037.

joint oznacza możliwość zaprojektowania eksperymentu w metodzie *conjoint analysis* oraz przeprowadzenie procedury pomiaru preferencji z wykorzystaniem jednego pakietu programu R. Zakłada się, że do uzyskania odpowiedniego układu czynnikowego wystarczające są dane dotyczące liczby i nazw zmiennych oraz ich poziomów uwzględnionych w badaniu.

2. Kompletnie i cząstkowe eksperymenty czynnikowe

Ogólna idea eksperymentu polega na obserwacji i zliczaniu częstości występowania określonych wartości zmiennych objaśniających (czynników) oraz badaniu ich wpływu na wartość zmiennej objaśnianej (por. [Bąk 2004, s. 80-81; Ferguson, Takane 1997, s. 257; Walesiak i Bąk 2000, s. 35-37]). Zakładając, że dowolny obiekt można opisać przy użyciu klasycznej funkcji o postaci (1):

$$Y = f(X_1, \dots, X_m, \varepsilon) \quad (1)$$

oraz możliwa jest aproksymacja tej funkcji, w konsekwencji otrzymuje się model regresji liniowej o postaci:

$$Y = \alpha_0 + \alpha_1 X_1 + \dots + \alpha_m X_m + \varepsilon, \quad (2)$$

gdzie: Y – zmienna objaśniana; $\alpha_0, \dots, \alpha_m$ – parametry równania regresji, których estymatorami są $\hat{\alpha}_0, \dots, \hat{\alpha}_m$; X_1, \dots, X_m – zmienne objaśniające; ε – składnik losowy. Aby możliwe było oszacowanie parametrów modelu (2) niezbędny jest eksperyment, który umożliwi rejestrację pomiarów wartości zmiennej objaśnianej Y przy wykorzystaniu różnych kombinacji (profilów, wariantów) wartości zmiennych objaśniających X_1, \dots, X_m (por. [Walesiak, Bąk 2000, s. 35-37; Bąk 2004, s. 80-81]).

Należy w tym miejscu zauważyć, że liczba pomiarów niezbędna do oszacowania nieznanych parametrów modelu (2) w przypadku tzw. pełnego (kompletnego) układu czynnikowego, który uwzględnia wszystkie kombinacje poziomów poszczególnych czynników (wszystkie profile), może być na tyle duża, że przekracza zdolność oceny respondentów. Z drugiej strony, aby spełnione były względy formalne, liczba pomiarów nie może być mniejsza od liczby szacowanych parametrów modelu (2). Oznacza to, że uwzględniając obie racje, w badaniach dąży się do pewnego rodzaju kompromisu pomiędzy rozmiarem eksperymentu oraz jego jakością. Wypadkową jest tutaj tzw. niepełny (cząstkowy) układ czynnikowy uwzględniający ograniczoną liczbę kombinacji poziomów zmiennych (podzbiór profilów), który obok warunku symetrii² po-

² Warunek symetrii oznacza, że każda ze zmiennych zawiera tyle samo poziomów, tym samym liczba zarejestrowanych pomiarów względem każdego z poziomów zmiennej powinna być taka sama (symetryczna). Układy, w których dla różnych zmiennych występuje różna liczba poziomów, np. 2 poziomy dla jednej zmiennej i 3 poziomy dla drugiej zmiennej, w literaturze przedmiotu określa się

winien spełniać warunek zrównoważenia i ortogonalności (por. [Bąk 2004, s. 84-85; Dykstra 1971, s. 684; Kuhfeld 1997, s. 78; Walesiak, Bąk 2000, s. 42]).

Warunek ortogonalności oznacza, że w układzie macierzy kolumny reprezentujące układ eksperymentu są względem siebie niezależne, co przejawia się zerowymi wartościami iloczynów skalarnych obliczonymi dla wszystkich par tych kolumn. Tym samym macierz kowariancji takiego układu jest diagonalna, a szacowania poszczególnych parametrów są niezależne (por. [Bąk 2004, s. 84; Kuhfeld, Tobias, Garratt 1994, s. 547; Zwerina 1997, s. 22]).

Warunek zrównoważenia oznacza, że w układzie macierzy każdy poziom danej zmiennej występuje tyle samo razy z każdym poziomem innej zmiennej dla wszystkich par czynników. Z obliczeniowego punktu widzenia oznacza to układ, w którym wszystkie pozadiagonalne elementy macierzy kowariancji odpowiadające wyrazowi wolnemu (występującemu zazwyczaj w pierwszej kolumnie i pierwszym wierszu macierzy kowariancji układu eksperymentu) są zerowe (por. [Bąk 2004, s. 85; Kuhfeld 1997, s. 78]). Warto w tym miejscu zauważyć, iż układ jest z definicji zrównoważony, gdy cała macierz kowariancji układu eksperymentu, wliczając w to wiersz i kolumnę, dla wyrazu wolnego jest diagonalna. Jak to zostało zauważone wcześniej, układ taki jest wówczas również ortogonalny.

W przypadku modelu *conjoint analysis* cząstkowy układ czynnikowy powinien zachowywać istotne źródła zmienności oraz minimalizację błędów ocen użyteczności cząstkowych rozumiane jako spełnienie m.in. warunku ortogonalności oraz warunku zrównoważenia.

3. Układy czynnikowe w pakiecie `conjoint`

Pakiet `conjoint` [Bąk, Bartłomowicz 2011; 2012] to autorskie oprogramowanie zawierające implementację klasycznej metody *conjoint analysis* dla środowiska R. Dostępność, instalacja oraz działanie pakietu realizuje się na podobieństwo około 4000 pakietów programu R bezpłatnie rozpowszechnianych w Internecie na podstawie licencji GNU GPL. Wymagania środowiskowe to zainstalowanie w wersji 2.0.0 (lub wyższej) programu bazowego R oraz pakietów `AlgDesign` i `clusterSim`. Pakiet można bezpłatnie pobrać, a następnie zainstalować ze strony internetowej CRAN³ lub strony WWW Katedry Ekonometrii i Informatyki Uniwersytetu Ekonomicznego we Wrocławiu⁴.

W wersji bieżącej pakietu (1.35) oferowanych jest 14 funkcji, które umożliwiają: estymację parametrów modelu *conjoint analysis* oraz segmentację respondentów (funkcje: `caModel`, `caSegmentation`), szacowanie użyteczności cząstkowych

mianem asymetrycznych (por. [Bąk 2004, s. 85]). Ze względu na to, że warunek ten w praktyce jest trudny do spełnienia, w dalszych rozważaniach zostaje pominięty.

³ [URL:] <http://cran.r-project.org/web/packages/conjoint>.

⁴ [URL:] <http://keii.ue.wroc.pl/conjoint>.

oraz teoretycznych użyteczności całkowitych (funkcje: `caPartUtilities`, `caTotalUtilities`), pomiar ważności atrybutów (funkcje: `caImportance`, `caUtilities`), a także – w ramach analizy symulacyjnej – szacowanie udziału w rynku profilów symulacyjnych (funkcje: `caBTL`, `caLogit`, `caMaxUtility`). Do funkcji specjalnego przeznaczenia zaliczyć natomiast należy funkcje umożliwiające uzyskanie zbiorczych wyników wybranych pomiarów oraz symulacji (funkcje: `Conjoint`, `ShowAllSimulations` oraz `ShowAllUtilities`).

Należy zauważyć, że prezentowany pakiet `conjoint` wspiera nierealizowane dotychczas w żadnym z pakietów programu R elementy procedury tradycyjnej metody *conjoint analysis*. Warunkiem jego zastosowania jest jednak odpowiednio przeprowadzony eksperyment, którego podstawą jest odpowiedni układ czynnikowy, w przeważającej większości przypadków będący układem cząstkowym.

Tym samym w artykule przedstawiono rozwinięcie pakietu `conjoint` o funkcje umożliwiające generowanie układów czynnikowych, szczególnie umożliwiające redukcję pełnego zbioru profilów do postaci układów cząstkowych⁵ oraz kodowanie układów czynnikowych (funkcje: `caFactorialDesign` oraz `caEncodedDesign`). W proponowanym rozwiązaniu wykorzystuje się utworzony w tym celu i udostępniany w Internecie pakiet `AlgDesign` [Wheeler 2012]. Jego implementacja w pakiecie `conjoint` realizowana jest w postaci wspomnianej pary funkcji (tab. 1), które odpowiednio umożliwiają uzyskanie eksperymentu czynnikowego oraz jego zakodowanie. Oznacza to możliwość zaprojektowania eksperymentu w metodzie *conjoint analysis* oraz uzyskania wyników pomiaru preferencji od razu w pakiecie `conjoint` programu R. Do uzyskania odpowiedniego układu czynnikowego wystarczające są dane dotyczące liczby branych pod uwagę zmiennych oraz ich poziomów z nazwami włącznie.

Tabela 1. Funkcje pakietu `conjoint` – projektowanie eksperymentu

Funkcje pakietu <code>conjoint</code> – projektowanie układu czynnikowego	
<code>caFactorialDesign(data, type="fractional", cards=NA)</code> – funkcja wyznacza (pełny lub cząstkowy układ czynnikowy z zachowaniem nazw zmiennych oraz ich poziomów	
<code>caEncodedDesign(design)</code> – funkcja koduje układ eksperymentu uzyskany z wykorzystaniem funkcji <code>caFactorialDesign</code> na potrzeby działania modułu <code>conjoint</code>	
Argumenty funkcji	
<code>data</code>	układ zmiennych oraz ich poziomów wraz z odpowiednimi nazwami
<code>type</code>	rodzaj układu czynnikowego (domyślna wartość – „null”)
<code>cards</code>	liczba profilów/ kart eksperymentu (domyślnie wartość nieznaną – NA)
<code>design</code>	układ czynnikowy uzyskany z zastosowania funkcji <code>caFactorialDesign</code> (z nazwami zmiennych oraz ich poziomów)

Źródło: opracowanie własne.

⁵ Na podobieństwo polecenia `ORTHOPLAN` programu `SPSS`.

Funkcją odpowiedzialną za generowanie układów w pakiecie `conjoint` jest funkcja `caFactorialDesign`, która umożliwia wygenerowanie układu czynnikowego o zadanym z góry układzie. Umożliwia to następująca składnia funkcji:

```
caFactorialDesign(data, type="null", cards=NA),
```

gdzie: `data` – układ zmiennych oraz ich poziomów wraz z odpowiednimi nazwami, `type` – rodzaj układu czynnikowego (możliwe typy układów: „full” – pełny układ czynnikowy, „null” – „standardowy” układ czynnikowy, „fractional” – cząstkowy układ czynnikowy, „ca” – układ czynnikowy tradycyjnej metody *conjoint analysis*, „aca” – układ czynnikowy adaptacyjnej metody *conjoint analysis*, „orthogonal” – układ czynnikowy ortogonalny), `cards` – zadana liczba profili (kart eksperymentu).

Funkcją odpowiedzialną za kodowanie układów w pakiecie `conjoint` jest funkcja `caEncodedDesign`, która umożliwia zakodowanie układu czynnikowego. Umożliwia to następująca składnia funkcji:

```
caEncodedDesign(design),
```

gdzie: `design` – układ czynnikowy uzyskany z zastosowania funkcji `caFactorialDesign` (z nazwami zmiennych oraz ich poziomów). Dopuszcza się, aby argumentem funkcji `caEncodedDesign` była wartość argumentu `data` funkcji `caFactorialDesign`.

4. Przykłady zastosowania funkcji projektowania układów czynnikowych

W przykładzie eksperyment składa się z czterech zmiennych opisujących czyste płyty CD, z których dwie zmienne zawierają po trzy poziomy, dwie pozostałe po dwa poziomy realizacji. Są to odpowiednio (wraz z nazwami poziomów): pojemność (650 MB, 700 MB), cena (niska, średnia, wysoka), opakowanie (pudełko, koperta) oraz kolor (niebieski, srebrny, złoty). W języku R składnia eksperymentu przedstawia się następująco:

```
> experiment=expand.grid(
pojemność=c("650MB", "700MB"),
cena=c("niska", "średnia", "wysoka"),
opakowanie=c("pudełko", "koperta"),
kolor=c("niebieski", "srebrny", "złoty"))
```

Przykład 1. Generowanie kompletnego układu czynnikowego

W wyniku zastosowania funkcji `caFactorialDesign()` z wartością „full” atrybutu `type` otrzymuje się pełny układ czynnikowy, który w przypadku omawianego eksperymentu składa się z maksymalnej możliwej liczby 36 profili:

```

> design=caFactorialDesign(experiment, type="full")
> print(design)
  pojemność   cena opakowanie   kolor
1     650MB   niska   pudełko niebieski
2     700MB   niska   pudełko niebieski
3     650MB  średnia   pudełko niebieski
4     700MB  średnia   pudełko niebieski
5     650MB  wysoka   pudełko niebieski
6     700MB  wysoka   pudełko niebieski
7     650MB   niska   koperta niebieski
8     700MB   niska   koperta niebieski
9     650MB  średnia   koperta niebieski
10    700MB  średnia   koperta niebieski
11    650MB  wysoka   koperta niebieski
...
36    700MB  wysoka   koperta   złoty

```

Kodowanie tak otrzymanego układu z wykorzystaniem funkcji `caEncodedDesign()` pozwala uzyskać układ czynnikiowy w postaci:

```

> code=caEncodedDesign(design)
> print(code)
  pojemność   cena opakowanie   kolor
1           1     1           1     1
2           2     1           1     1
3           1     2           1     1
4           2     2           1     1
5           1     3           1     1
6           2     3           1     1
7           1     1           2     1
8           2     1           2     1
9           1     2           2     1
10          2     2           2     1
11          1     3           2     1
...
36          2     3           2     3

```

Jak nie trudno zauważyć, macierz kowariancji takiego układu jest diagonalna i przedstawia się następująco:

```
> print(cov(code))
           pojemność      cena opakowanie      kolor
pojemność 0.2571429 0.0000000 0.0000000 0.0000000
cena      0.0000000 0.6857143 0.0000000 0.0000000
opakowanie 0.0000000 0.0000000 0.2571429 0.0000000
kolor     0.0000000 0.0000000 0.0000000 0.6857143
```

Oznacza to, iż kompletny układ czynnikowy z definicji jest ortogonalny. Dodatkowym potwierdzeniem tego faktu jest macierz korelacji i jej wyznacznik równy 1:

```
> print(cor(code))
           pojemność  cena opakowanie  kolor
pojemność           1     0           0     0
cena                 0     1           0     0
opakowanie           0     0           1     0
kolor                0     0           0     1
```

```
> print(det(cor(code)))
```

```
[1] 1
```

Przykład 2. Generowanie cząstkowego układu czynnikowego o określonej liczbie kart

W wyniku zastosowania funkcji `caFactorialDesign()` z wartością „fractional” atrybutu `type` oraz zadaną wartością atrybutu `cards` otrzymuje się cząstkowy układ czynnikowy, który w przykładzie składa się z 16 profiliów:

```
> design=caFactorialDesign(experiment, type="fractional", cards=16)
```

```
> print(design)
```

```
   pojemność      cena opakowanie      kolor
2      700MB      niska      pudełko niebieski
5      650MB      wysoka      pudełko niebieski
6      700MB      wysoka      pudełko niebieski
7      650MB      niska      koperta  niebieski
9      650MB      średnia     koperta  niebieski
10     700MB      średnia     koperta  niebieski
15     650MB      średnia     pudełko   srebrny
16     700MB      średnia     pudełko   srebrny
17     650MB      wysoka     pudełko   srebrny
19     650MB      niska      koperta   srebrny
20     700MB      niska      koperta   srebrny
```

24	700MB	wysoka	koperta	srebrny
25	650MB	niska	pudełko	złoty
28	700MB	średnia	pudełko	złoty
33	650MB	średnia	koperta	złoty
36	700MB	wysoka	koperta	złoty

Ze względu na to, iż nie jest to układ ortogonalny, macierz kowariancji takiego układu nie jest diagonalna i przedstawia się następująco:

```
> print(cov(caEncodedDesign(design)))
           pojemność      cena opakowanie kolor
pojemność 0.26666667 0.06666667 0.00000000 0.00
cena      0.06666667 0.66666667 -0.06666667 0.00
opakowanie 0.00000000 -0.06666667 0.26666667 0.00
kolor     0.00000000 0.00000000 0.00000000 0.65
```

Dodatkowym potwierdzeniem tego faktu jest macierz korelacji i jej wyznacznik różny od jedności:

```
> print(cor(caEncodedDesign(design)))
           pojemność      cena opakowanie kolor
pojemność 1.00000000 0.1581139 0.00000000 0
cena      0.1581139 1.00000000 -0.1581139 0
opakowanie 0.00000000 -0.1581139 1.00000000 0
kolor     0.00000000 0.00000000 0.00000000 1
```

```
> print(det(cor(caEncodedDesign(design))))
[1] 0.95
```

Przykład 3. Generowanie układu czynnikowego tradycyjnej i adaptacyjnej metody *conjoint analysis*

W wyniku zastosowania funkcji `caFactorialDesign()` z wartościami „ca” oraz „aca” atrybutu `type` otrzymuje się odpowiednio układ czynnikowy tradycyjnej oraz adaptacyjnej metody *conjoint analysis*:

```
> design=caFactorialDesign(experiment, type="ca")
> print(design)
           pojemność      cena opakowanie      kolor
6      700MB wysoka      pudełko niebieski
9      650MB średnia      koperta niebieski
```



```

16      700MB średnia   pudełko   srebrny
20      700MB niska    koperta   srebrny
23      650MB wysoka   koperta   srebrny
25      650MB niska    pudełko   złoty
34      700MB średnia   koperta   złoty
> design=caFactorialDesign(experiment, type="aca")
> print(design)
  pojemność   cena opakowanie   kolor
1      650MB   niska   pudełko niebieski
5      650MB   wysoka  pudełko niebieski
8      700MB   niska   koperta niebieski
10     700MB   średnia  koperta niebieski
14     700MB   niska   pudełko srebrny
16     700MB   średnia  pudełko srebrny
23     650MB   wysoka   koperta srebrny
27     650MB   średnia  pudełko złoty
30     700MB   wysoka   pudełko złoty
31     650MB   niska    koperta złoty
36     700MB   wysoka   koperta złoty

```

Podobnie jak miało to miejsce w przypadku pełnego układu czynnikowego, liczba profili jest w obu przypadkach wyznaczana „odgórnie”. W przypadku kompletnego układu eksperymentu jest to maksymalna możliwa liczba profili wynikająca z iloczynu liczby poziomów dla wszystkich zmiennych. W przypadku tradycyjnej metody *conjoint analysis* minimalna liczba profili oceniana przez respondentów (n) wyznaczana jest według zależności (por. [Bąk 2004, s. 64; Walesiak, Bąk 2000, s. 31]):

$$n = p - m + 1, \quad (3)$$

gdzie: p – liczba poziomów dla wszystkich atrybutów, m – liczba atrybutów.

W przypadku adaptacyjnej metody *conjoint analysis* minimalna liczba profili ocenianych przez respondentów (n) wyznaczana jest według zależności (por. [Bąk 2004, s. 64]):

$$n = 3(p - m + 1) - p, \quad (4)$$

gdzie: p , m – oznaczenia jak w (3).

W obu prezentowanych przypadkach uzyskane układy nie są ortogonalne (macierze kowariancji tych układów nie są diagonalne, wyznaczniki macierzy korelacji różne są od jedności).

Przykład 4. Generowanie ortogonalnego układu czynnikowego

W wyniku zastosowania funkcji `caFactorialDesign()` z wartością „orthogonal” atrybutu `type` otrzymuje się ortogonalny układ czynnikowy:

```
> design=caFactorialDesign(experiment, type="fractional", cards=16)
> print(design)
  pojemność   cena opakowanie   kolor
1      650MB   niska   pudełko niebieski
4      700MB  średnia   pudełko niebieski
11     650MB   wysoka   koperta niebieski
15     650MB  średnia   pudełko   srebrny
17     650MB   wysoka   pudełko   srebrny
20     700MB   niska   koperta   srebrny
25     650MB   niska   pudełko     złoty
30     700MB   wysoka   pudełko     złoty
33     650MB  średnia   koperta     złoty
```

Kodowanie tak otrzymanego układu z wykorzystaniem funkcji `caEncodedDesign()` pozwala uzyskać układ czynnikowy w gotowej do praktycznego zastosowania postaci:

```
> code=caEncodedDesign(design)
> print(code)
  pojemność   cena opakowanie   kolor
1           1     1           1     1
4           2     2           1     1
11          1     3           2     1
15          1     2           1     2
17          1     3           1     2
20          2     1           2     2
25          1     1           1     3
30          2     3           1     3
33          1     2           2     3
```

Potwierdzeniem ortogonalności uzyskanego cząstkowego układu czynnikowego są: diagonalna macierz kowariancji oraz diagonalna (jednostkowa) macierz korelacji:

```
> print(round(cov(caEncodedDesign(design)), 8))
  pojemność   cena opakowanie   kolor
```

pojemność	0.25	0.00	0.00	0.00
cena	0.00	0.75	0.00	0.00
opakowanie	0.00	0.00	0.25	0.00
kolor	0.00	0.00	0.00	0.75

```
> print(round(cor(caEncodedDesign(design)), 8))
      pojemność  cena  opakowanie  kolor
pojemność      1    0            0    0
cena            0    1            0    0
opakowanie      0    0            1    0
kolor           0    0            0    1
```

Jak nietrudno zauważyć, uzyskanie tych samych właściwości dla układów kompletnego i cząstkowego przemawia za praktycznym wykorzystaniem tego drugiego układu.

Literatura

- Bąk A., *Dekompozycyjne metody pomiaru preferencji w badaniach marketingowych*, Prace Naukowe AE we Wrocławiu nr 1013, Seria: Monografie i Opracowania nr 157, Wydawnictwo AE we Wrocławiu, Wrocław 2004.
- Bąk A., Bartłomowicz T., *Implementacja klasycznej metody conjoint analysis w pakiecie conjoint programu R*, [w:] Prace Naukowe UE we Wrocławiu nr 176, Wydawnictwo UE we Wrocławiu, Wrocław 2011, s. 94-104.
- Bąk A., Bartłomowicz T., *Package conjoint. Conjoint analysis package*, <http://cran.r-project.org/web/packages/conjoint>, 2012.
- Dodge Y., Fedorov, V.V., Wynn H.P. (red.), *Optimal Design and Analysis of Experiments*, North-Holland, Amsterdam 1988.
- Dykstra O. Jr., *The augmentation of experimental data to maximize $(X'X)^{-1}$* , "Technometrics" 1971, no. 13(3), s. 682-688.
- Ferguson G.A., Takane Y., *Analiza statystyczna w psychologii i pedagogice*, PWN, Warszawa 1997.
- Kuhfeld W.F., *Efficient Experimental Designs Using Computerized Searches*, SAS Institute, 1997, [URL] <http://www.sawtoothsoftware.com/download/techpap/1997Proceedings.pdf>.
- Kuhfeld W.F., Tobias R.D., Garratt M., *Efficient experimental design with marketing research applications*, "Journal of Marketing Research" 1994, no. 31 (November), s. 545-557.
- Walesiak M., Bąk A., *Conjoint analysis w badaniach marketingowych*, Wydawnictwo AE we Wrocławiu, Wrocław 2000.
- Wheeler R.E., *Package AlgDesign. Algorithmic Experimental Design*, <http://cran.r-project.org/web/packages/AlgDesign>, 2012.
- Zwerina K., *Discrete Choice Experiments in Marketing*, Physica-Verlag, Heidelberg-New York 1997.

FRACTIONAL FACTORIAL DESIGN AND ITS IMPLEMENTATION IN CONJOINT PACKAGE R PROGRAM

Summary: The paper presents the development of R package for conjoint features: `ca-FactorialDesign()` function which allows to generate a complete or fractional factorial design and `caEncodedDesign()` function which allows to encode obtained experiments. The basis of both functions is the implementation of `AlgDesign` package functions in `con-joint` package. In the proposed solution, it is assumed that to obtain a relevant factorial design the number of variables and their levels with the names of variables should be enough.

Keywords: fractional factorial design, *conjoint analysis*, R program.