Experimental design in DCM

Gianfranco Atzeni

Environmental Economics

Why we need Experimental Design?

- If we can observe how people make choices in the real world, we can study their revealed preference.
- When IIA is a reasonable approximation of reality, simple discrete choice produces good forecasts.

Often we want to study alternatives that do not exist

- Revealed preference: observing choices that people have made in the real world
- Stated preference: asking people to choose among hypothetical choices
- Revealed preference data can be used to calibrate stated preference models

Steps of the Experimental Design

1. Break the product or service into a set of attributes and levels.
2. Choose an appropriate vehicle for generating your design.
   - Tables
   - Software
   - Expert
3. Construct your design.
4. Evaluate the results.
   - Check business validity of attributes and levels.
   - Pre-test the questionnaire.
5. Return to step 1 if necessary.

How to identify the attribute list

- Define the actual or hypothetical market
- Identify al relevant substitutes
- Make sure that attributes are independent

How to select levels

- Levels of each attribute should be mutually exclusive and collectively exhaustive
- Use precise and clear statements to define levels, with metrics whenever possible.
  - Avoid using ranges to describe a single level of an attribute, such as ”weighs 3 to 5 kilos.”
  - Levels such as ”superior performance” also leave too much in question. What does ”superior performance” mean?
- Ranges of levels should be sufficiently extreme to cover the entire scope of the research.
- It is important to balance the number of levels across attributes.
- When levels are quantitative it is advisable to use realistic values.

Example

Attributes	Fashion	Quality	Price
Levels	Traditional	Standard	25
	Modern	High	149

The number of combinations is:

$$2^3 = 8$$

Number of **levels** raised to the power of the number of **attributes**

Dummy coding

Attributes	Fashion	Quality	Price
Levels	0 (Traditional)	0 (Standard)	0 (25)
	1 (Modern)	1 (High)	1 (149)

One possible approach is to use all possible 8 combinations. The respondent has to choose among the 8 all possible items resulting form the combination of 3 attributes each with two levels.

Characteristics of the design

	F	Q	P
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1
sum	4	4	4

This design is **orthogonal**: rows are perfectly uncorrelated; each pair of levels occurs equally often

It is **balanced**: each level appears an equal number of times.

Characteristics of the design

	F	Q	P
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1
sum	4	4	4

This is a **full factorial** design:

- it contains alla possible levels of the factors
- it allows you to estimate *main effects* and two-way or higher *interactions*

It is also an **orthogonal array**

- all possible interactions are estimable.

From Design to Choice set

Design

	1	2	3	4	5	6	7	8
F	0	1	1	1	0	0	0	1
Q	0	0	0	1	0	1	1	1
P	0	0	1	0	1	0	1	1

Choice set

	1	2	3	4	5	6	7	8
Fashion	Traditional	Modern	Modern	Modern	Traditional	Traditional	Traditional	Modern
Quality	Standard	Standard	Standard	High	Standard	High	High	High
Price	25	25	149	25	149	25	149	149

It is also possible to divide the choice set into two choice sets with 4 alternatives each, or 4 choice sets with 2 alternatives each.

Main effects and interactions

Main effects

- simple effect of price fashion and quality on the choice
- the effect is independent of the levels of other attributes
- for example the effect of quality on the choice is the same at a price of 25 or 149.

Interactions

- involve two or more factors
- the effect of one factor depends on the level of another
- for example the effect of quality on the choice differs when the price is 25 wrt 149.

Fractional factorial

Suppose we have five attributes:

- 2 with 4 levels
- 3 with 5 levels

This means a full factorial of $4^2 \times 5^3$=2000 possible alternatives. That are too many to handle, even if partitioned into blocks.

For this reason we need to reduce the number of alternatives to a number which is possible to handle. This design is called **fractional factorial**.

Design efficiency

Efficiency measure the goodness of a design.

- it is inversely related to the variance of the parameter estimates
- measure of efficiency are different if we consider linear model compared to logit models
- One common measure is D-efficiency, a value scaled fro 0 to 100, for linear models.
- for logit models STATA computes fractional factorial using the command `dcreate`, which employs the modified Fedorov algorithm (Cook and Nachtsheim, 1980; Zwerina et al., 1996; Carlsson and Martinsson, 2003). The algorithm maximises the D-efficiency of the design based on the covariance matrix of the conditional logit model.

How to compute a fractional factorial

In STATA is possible to install a package to generate efficient designs for discrete choice experiments: `dcreate`.

The command take the existing dataset as a full factorial that that need to be reduced to create the **choice set**.

Suppose we have a design imade of 2 four-level attributes and 4 two-level attributes:

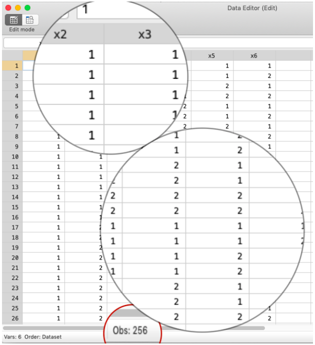
this results in $4^2 \times 2^4 = 256$ possible combinations.

To start we need to create the **full factorial** using the command `genfact`.

From full factorial to fractional factorial

First we need to define the matrix that contains the level in the design, that we denominate `levmat`:

- `matrix levmat = 4,4,2,2,2,2`
- we generate the full factorial `genfact, levels(levmat)`



We obtain a **full factorial** with 6 variables which the command denominates $x1 - x6$ with 256 alternatives.

Now suppose we want to create a **fractional factorial** with 16 alternatives.

We use the command `dcreate` by Arne Risa Hole (`a.r.hole@sheffield.ac.uk`) to obtain the result.

`dcreate`

Before creating the fractional factorial we need to create the matrix of coefficient priors to evaluate the efficiency of the design.

- `matrix b = J(1,10,0)`
- `dcreate i.x1 i.x2 i.x3 i.x4 i.x5 i.x6, nalt(2) nset(16) bmat(b)`

Where `nalt(#)` specifies the number of alternatives in the design and `nset(#)` specifies the number of choice sets in the design. `bmat(#)` specifies a matrix of coefficient priors.

Choice set

We obtain the fractional factorial with 16 choice set made of 2 alternatives each.

Data Editor (Edit)

Edit mode Save Find

x1[1] 4

	x1	x2	x3	x4	x5	x6	choice_set	alt
1	4	4	1	2	2	2	1	1
2	1	3	2	1	1	1	1	2
3	2	4	2	1	1	2	2	1
4	4	2	1	2	2	1	2	2
5	1	2	1	1	2	1	3	1
6	2	3	2	2	1	2	3	2
7	1	3	2	2	2	2	4	1
8	3	4	1	1	1	1	4	2
9	3	1	2	2	2	1	5	1
10	1	2	1	1	1	2	5	2
11	4	2	2	2	1	1	6	1
12	3	1	1	1	2	2	6	2
13	3	3	1	2	1	2	7	1
14	2	4	2	1	2	1	7	2
15	4	1	2	1	2	1	8	1
16	3	2	1	2	1	2	8	2
17	2	3	1	1	2	1	9	1
18	1	1	2	2	1	2	9	2
19	1	4	2	2	1	1	10	1
20	4	2	1	1	2	2	10	2
21	2	1	1	1	1	2	11	1
22	3	4	2	2	2	1	11	2
23	2	1	1	2	1	1	12	1
24	3	3	2	1	2	2	12	2
25	3	2	[illegible]	1	1	1	13	1
26	1	4	[illegible]	2	2	2	13	2

Obs: 32

Vars: 8 Order: Dataset

`dcreate` adds two variables:

- `choice_set` which identifies the choice set
- `alt` which identifies the alternatives within the choice set.

How to include an alternative specific constant

After `genfact`, create a matrix containing the attribute levels for the opt-out alternative. All the attribute levels are set to the base level (1)

- `matrix optout = J(1,6,1)`
- `matrix b = J(1,11,0)`
- `dcreate i.x1 i.x2 i.x3 i.x4 i.x5 i.x6, nalt(2) nset(16) fixedalt(optout) asc(3) bmat(b)`

It is also possible to divide the design into two blocks with 8 choice sets each using `blockdes`:

- `blockdes block, nblock(2)`