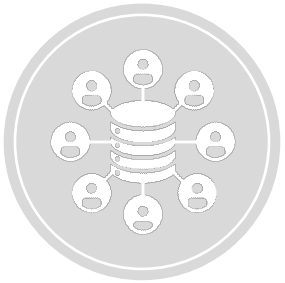
Orthogonal designs

Key concepts & study plan

Experimental design



Data collection & processing

Model specification & estimation

Interpretation & application

Orthogonal designs

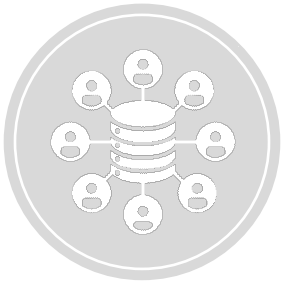
Outline

- Basic building blocks: orthogonal arrays
- Generating orthogonal designs
- Relevant videos to watch

Basic building blocks: orthogonal arrays

Key concepts & study plan

Experimental design



Data collection & processing

Model specification & estimation

Interpretation & application

Basic building blocks: orthogonal arrays

Orthogonal array

- Table where each column describes a factor in an experiment
  - Each column contains values 0, 1, 2, … (design coding) to represent levels of an attribute
  - Each row describes a specific combination of levels across all attributes
  - Balance within each column and across pairs of columns

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓

4x 0
4x 1

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓

4x 0
4x 1

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓

2x 0
2x 1
2x 2
2x 3

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓

4x 0
4x 1

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓
- Balance across 2 columns ✓

2x	0	0
2x	0	1
2x	1	0
2x	1	1

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓
- Balance across 2 columns ✓

1x	0	0
1x	0	1
1x	0	2
1x	0	3
1x	1	0
1x	1	1
1x	1	2
1x	1	3

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓
- Balance across 2 columns ✓

2x	0	0
2x	0	1
2x	1	0
2x	1	1

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓
- Balance across 2 columns ✓

1x	0	0
1x	0	1
1x	0	2
1x	0	3
1x	1	0
1x	1	1
1x	1	2
1x	1	3

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓
- Balance across 2 columns ✓

2x	0	0
2x	0	1
2x	1	0
2x	1	1

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Example – Orthogonal array

- Balance within a column ✓
- Balance across 2 columns ✓

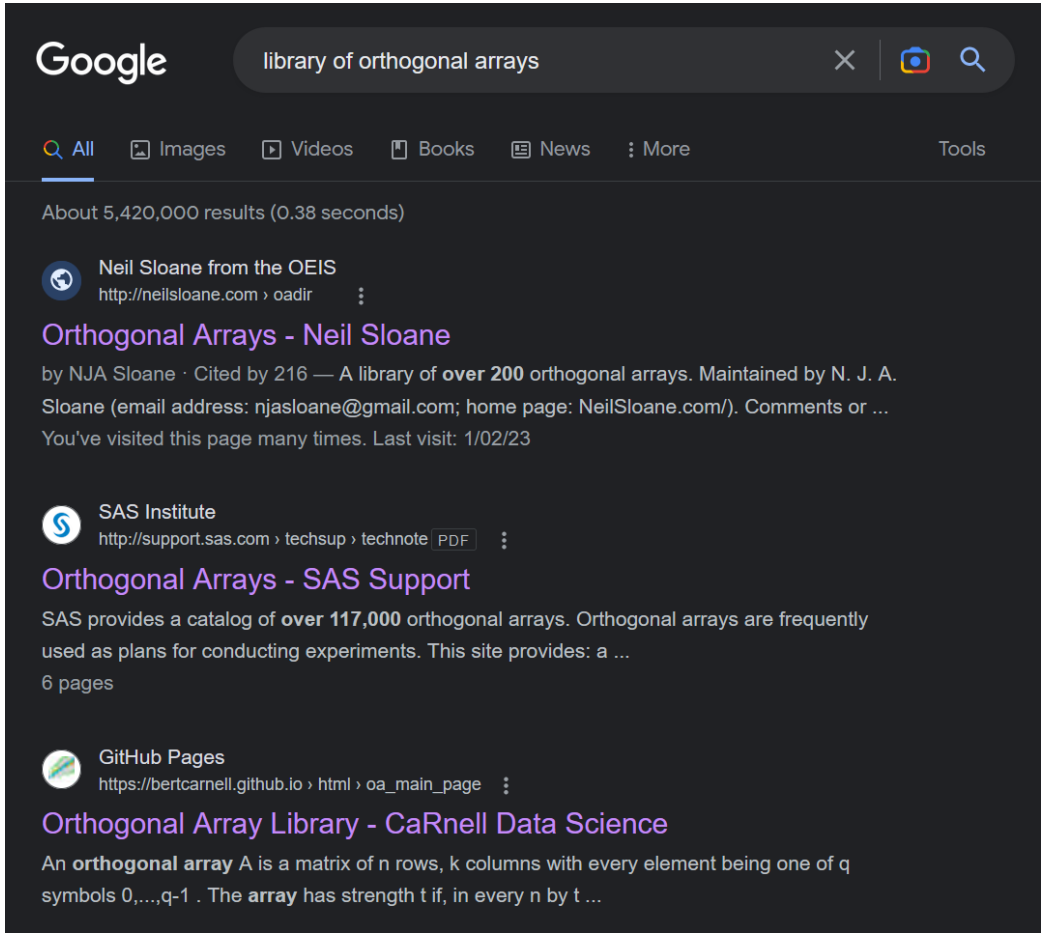
1x	0	0
1x	0	1
1x	1	0
1x	1	1
1x	2	0
1x	2	1
1x	3	0
1x	3	1

0	0	0	0
0	1	2	1
1	0	1	1
1	1	3	0
0	0	3	1
0	1	1	0
1	0	2	0
1	1	0	1

Basic building blocks: orthogonal arrays

Finding orthogonal arrays

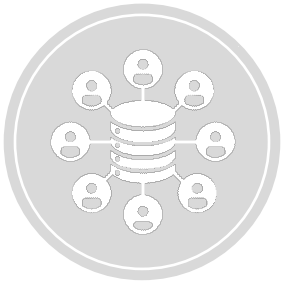
- Online libraries
  - Search for “library of orthogonal arrays”
  - support.sas.com/techsup/technote/ts723_Designs.txt
  - neilsloane.com/oadir
- Software tools
  - SAS
  - Ngene
  - SurveyEngine
  - SPSS
  - JMP



Generating orthogonal designs

Key concepts & study plan

Experimental design



Data collection & processing

Model specification & estimation

Interpretation & application

Generating orthogonal designs

Steps

1. Define list of attributes
2. Define levels per attribute
3. Find suitable orthogonal array
4. Assign columns in orthogonal array to attributes in design
5. Augment levels for generic attributes to minimise overlap
6. Replace design coding with actual attribute levels in design

Experimental design software tools generate orthogonal designs by automatically performing these steps

Generating orthogonal designs

Example

- Bike alternative with attribute ‘dist’
- Bus A and Bus B alternative with attributes ‘time’ and ‘cost’

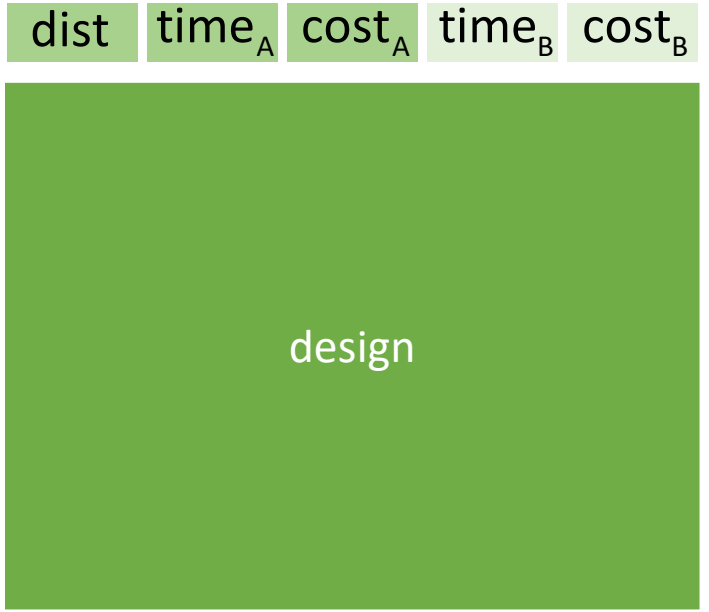
Bike	Bus A		Bus B	
dist	$time_A$	$cost_A$	$time_B$	$cost_B$
2	10	1	20	3
2	20	2	30	1
2	30	3	10	2
5	10	2	20	1
5	20	3	30	2
5	30	1	10	3
10	10	3	20	2
10	20	1	30	3
10	30	2	10	1

Generating orthogonal designs

Step 1 – Define list of attributes

- Note that two attributes are generic

$$V_{bike} = \beta_1 + \beta_2 \cdot dist$$
$$V_{busA} = \beta_3 \cdot time_A + \beta_4 \cdot cost_A$$
$$V_{busB} = \beta_3 \cdot time_B + \beta_4 \cdot cost_B$$



Generating orthogonal designs

Step 2 – Define levels per attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
design				

Generating orthogonal designs

Step 3 – Find suitable orthogonal array

- At least 3 columns with 3 levels

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
design				

Generating orthogonal designs

Step 3 – Find suitable orthogonal array

- At least 3 columns with 3 levels

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

orthogonal array

0	0	0	0
0	1	1	2
0	2	2	1
1	0	1	1
1	1	2	0
1	2	0	2
2	0	2	2
2	1	0	1
2	2	1	0

dist	time_A	cost_A	time_B	cost_B
design				

Generating orthogonal designs

Step 4 – Assign columns in orthogonal array to attributes in design

- Select any 3 columns

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

orthogonal array

0	0	0	0
0	1	1	2
0	2	2	1
1	0	1	1
1	1	2	0
1	2	0	2
2	0	2	2
2	1	0	1
2	2	1	0

dist	time_A	cost_A	time_B	cost_B

Generating orthogonal designs

Step 4 – Assign columns in orthogonal array to attributes in design

- ‘dist’ attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

orthogonal array

0	0	0
1	1	2
2	2	1
0	1	1
1	2	0
2	0	2
0	2	2
1	0	1
2	1	0

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
0				
0				
0				
1				
1				
1				
2				
2				
2				

Generating orthogonal designs

Step 4 – Assign columns in orthogonal array to attributes in design

- ‘time’ attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot dist$$
$$V_{busA} = \beta_3 \cdot time_A + \beta_4 \cdot cost_A$$
$$V_{busB} = \beta_3 \cdot time_B + \beta_4 \cdot cost_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

orthogonal array

	0	0
	1	2
	2	1
	1	1
	2	0
	0	2
	2	2
	0	1
	1	0

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
0	0			
0	1			
0	2			
1	0			
1	1			
1	2			
2	0			
2	1			
2	2			

Generating orthogonal designs

Step 4 – Assign columns in orthogonal array to attributes in design

- ‘cost’ attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot dist$$
$$V_{busA} = \beta_3 \cdot time_A + \beta_4 \cdot cost_A$$
$$V_{busB} = \beta_3 \cdot time_B + \beta_4 \cdot cost_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

orthogonal array

	0
	2
	1
	1
	0
	2
	2
	1
	0

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
0	0	0		
0	1	1		
0	2	2		
1	0	1		
1	1	2		
1	2	0		
2	0	2		
2	1	0		
2	2	1		

Generating orthogonal designs

Step 5 – Augment levels for generic attributes to minimise overlap

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
0	0	0		
0	1	1		
0	2	2		
1	0	1		
1	1	2		
1	2	0		
2	0	2		
2	1	0		
2	2	1		

Generating orthogonal designs

Step 5 – Augment levels for generic attributes to minimise overlap

- Shift the 'time' attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot dist$$
$$V_{busA} = \beta_3 \cdot time_A + \beta_4 \cdot cost_A$$
$$V_{busB} = \beta_3 \cdot time_B + \beta_4 \cdot cost_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

+1 shift

0 → 1

1 → 2

2 → 0

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
0	0	0	1	
0	1	1	2	
0	2	2	0	
1	0	1	1	
1	1	2	2	
1	2	0	0	
2	0	2	1	
2	1	0	2	
2	2	1	0	

Generating orthogonal designs

Step 5 – Augment levels for generic attributes to minimise overlap

- Shift the 'cost' attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot dist$$
$$V_{busA} = \beta_3 \cdot time_A + \beta_4 \cdot cost_A$$
$$V_{busB} = \beta_3 \cdot time_B + \beta_4 \cdot cost_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

+2 shift

0 → 2
1 → 0
2 → 1

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
0	0	0	1	2
0	1	1	2	0
0	2	2	0	1
1	0	1	1	0
1	1	2	2	1
1	2	0	0	2
2	0	2	1	1
2	1	0	2	2
2	2	1	0	0

Generating orthogonal designs

Step 6 – Replace design coding with actual attribute levels in design

- Relabel the attribute levels (often done within the survey instrument)

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
0	0	0	1	2
0	1	1	2	0
0	2	2	0	1
1	0	1	1	0
1	1	2	2	1
1	2	0	0	2
2	0	2	1	1
2	1	0	2	2
2	2	1	0	0

Generating orthogonal designs

Step 6 – Replace design coding with actual attribute levels in design

- Relabel ‘dist’ attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

Relabelling

- 0 → 2
- 1 → 5
- 2 → 10

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
2	0	0	1	2
2	1	1	2	0
2	2	2	0	1
5	0	1	1	0
5	1	2	2	1
5	2	0	0	2
10	0	2	1	1
10	1	0	2	2
10	2	1	0	0

Generating orthogonal designs

Step 6 – Replace design coding with actual attribute levels in design

- Relabel ‘time’ attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

Relabelling

0	10
1	20
2	30

dist	time$_A$	cost$_A$	time$_B$	cost$_B$
2	10	0	20	2
2	20	1	30	0
2	30	2	10	1
5	10	1	20	0
5	20	2	30	1
5	30	0	10	2
10	10	2	20	1
10	20	0	30	2
10	30	1	10	0

Generating orthogonal designs

Step 6 – Replace design coding with actual attribute levels in design

- Relabel ‘cost’ attribute

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

Relabelling

0 → 1

1 → 2

2 → 3

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
2	10	1	20	3
2	20	2	30	1
2	30	3	10	2
5	10	2	20	1
5	20	3	30	2
5	30	1	10	3
10	10	3	20	2
10	20	1	30	3
10	30	2	10	1

Generating orthogonal designs

Final orthogonal design

$$V_{bike} = \beta_1 + \beta_2 \cdot \text{dist}$$
$$V_{busA} = \beta_3 \cdot \text{time}_A + \beta_4 \cdot \text{cost}_A$$
$$V_{busB} = \beta_3 \cdot \text{time}_B + \beta_4 \cdot \text{cost}_B$$

dist	time	cost
2	10	1
5	20	2
10	30	3

dist	$time_A$	$cost_A$	$time_B$	$cost_B$
2	10	1	20	3
2	20	2	30	1
2	30	3	10	2
5	10	2	20	1
5	20	3	30	2
5	30	1	10	3
10	10	3	20	2
10	20	1	30	3
10	30	2	10	1

Generating orthogonal designs

More information

- Street, D.J., Burgess, L. & Louviere, J.J. (2005) Quick and easy choice sets: Constructing optimal and nearly optimal stated choice experiments. *International Journal of Research in Marketing*, 22(4), 459-470.
- Street, D.J. & Burgess, L. (2007) *The construction of optimal stated choice experiments: Theory and methods*. John Wiley & Sons.