DCC / ICEx / UFMG

Experiment Design

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Experiment Design

- To get the most out of the experiment, it must be carefully planned and designed
  - Design is closely related to interpretation

- An experiment consists of a series of tests of the treatments
  - A design describes how the tests are organized and run

Design and Analysis

- The experiment design and statistical analysis are closely related
  - For instance, the design determines how many tests are required for each treatment

- A proper design allows replication

General Design Principles

- Most experiment designs use some combination of three design principles

- General design principles
  - Randomization
  - Blocking
  - Balancing

Randomization

- All statistical methods require observations from independent random variables

- Randomization applies to the allocation of objects, subjects, and the order of the tests
  - It is also used to select subjects that are representative of the population

Blocking

- Blocking is used to systematically eliminate the undesired effect in the comparison among treatments
  - Within one block, the undesired effect (e.g., application size) is the same
  - We can study the effect of the treatment (e.g., new static analysis tool) on one block

Example of Blocking

- Let's suppose a experiment to investigate the effect of UML on software reliability
  - Participants are developers with varying degrees of experience in OOP
- Two blocks can be created
  - One of developers with none to small experience in OOP
  - One of developers with medium to extensive experience in OOP

Balancing

- We have a balanced design if each treatment has equal number of subjects
  - That is, the same number of subjects in each group
- Although balancing is no necessary, it has some benefits
  - It simplifies the data analysis
  - It strengthens the statistical analysis

Standard Design Types

- Some of the most frequently used experiment designs are
  - One factor with two treatments
  - One factor with more than two treatments
  - Two factors with two treatments
  - More than two factors, each factor with two treatments

One Factor with Two Treatments

One Factor, Two Treatments

- In this design, we want to compare the two treatments against each other
- Example
  - Factor: Metrics for Bad Smell Detection
  - Treatments: Concern vs. Traditional
- Assigning treatments to subjects
  - Completely Randomized Design
  - Paired Comparison Design

Completely Randomized Design

- We assign subjects randomly to each treatment
- Each subject uses only one treatment
- If we have the same number of subjects per treatment, the design is balanced

Paired Comparison Design

- It is also called crossover design
  - We assign each subject to both treatment
  - The order is randomly assigned to each subject
  - If we have the same number of subjects starting with the first and the second treatments, the design is balanced
- Warning! Subjects may gain information from the first to the second treatment

Example: One Factor, Two Treatments

Completely Randomized Design

	Treatments	
Subjects	**Treat 1**	**Treat 2**
1	X	
2		X
3		X
4	X	
5		X
6	X	

Paired Comparison Design

	Treatments	
Subjects	**Treat 1**	**Treat 2**
1	2^{nd}	1^{st}
2	1^{st}	2^{nd}
3	2^{nd}	1^{st}
4	2^{nd}	1^{st}
5	1^{st}	2^{nd}
6	1^{st}	2^{nd}

One Factor with More Than Two Treatments

One Factor, More Treatments

- In this design type, comparison is often performed on the treatment means
- Example
  - Factor: Modeling Tool
  - Treatments: ArgoUML, StarUML, LucidChart
- Assigning treatments to subjects
  - Completely randomized design
  - Randomized complete block design

Completely Randomized Design

- Subjects are randomly assigned to each treatment

	Treatments		
Subjects	**Treat 1**	**Treat 2**	**Treat 3**
1	X		
2		X	
3			X
4	X		
5			X
6		X	

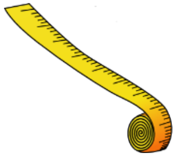
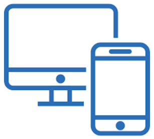
Randomized Complete Block

- Subjects use all treatments in different order

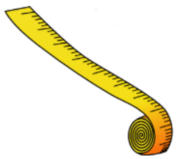
	Treatments		
Subjects	**Treat 1**	**Treat 2**	**Treat 3**
1	1st	3rd	2nd
2	3rd	1st	2nd
3	2nd	3rd	1st
4	2nd	1st	3rd
5	3rd	2nd	1st
6	1st	2nd	3rd

Two Factors

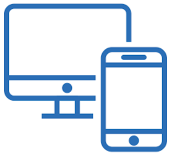
Two Factors

- The experiment gets more complex when two factors are analyzed
- It requires three hypotheses
  - One hypothesis for the effect of the first factor
  
  - One hypothesis for the effect of the second factor
  
  - One hypothesis for the interaction between the two factors
  

Examples of Two Factors



- Factor 1: Set of Metrics
  - Concern Metrics
  - Traditional Metrics



- Factor 2: Analyzed System
  - Health Watcher
  - MobileMedia
- Assigning treatments to subjects
  - 2 * 2 factorial design
  - Two-stage nested design

2 * 2 Factorial Design

- Subjects are randomly assigned to each combination of treatments

		Factor A	
		Treatment A1	Treatment A2
Factor B	Treatment B1	Subjects 4, 6	Subjects 1, 7
	Treatment B2	Subjects 2, 3	Subjects 5, 8

Two-Stage Nested Design

- Factor B is nested under Factor A
  - Factor B is different for treatments of Factor A

Factor A			
Treatment A1		Treatment A2	
Factor B		**Factor B**	
Treatment B1’	Treatment B2’	Treatment B1”	Treatment B2”
Subjects 1, 3	Subjects 6, 2	Subjects 7, 8	Subjects 4, 5

Example of 2-stage Nested Design

- Experiment investigates efficiency of unit test
  - Factor A (programming language): functional programs vs. OO programs
  - Factor B (defect proneness): Defect-prone vs. non-defect-prone
- The design has to be nested because defect-proneness is different in functional programs and OO programs

More Than Two Factors

More than Two Factors

- It is also called factorial design
- The effect on the dependent variable considers both
  - Each factor separately
  - The interaction between the factors
- Interactions can be between two or more factors

Assigning Treatments

- 2^k factorial design (general case)
  - k is the number of factors
- 2^2 factorial design (k = 2)
  - The same of 2 * 2 factorial design
  - Two factors with two treatments
- 2^3 factorial design (k = 3)
  - Three factors, each with two treatments

Example of 2^3 Factorial Design

Factor A	Factor B	Factor C	Subjects
A1	B1	C1	2, 3
A2	B1	C1	1, 13
A1	B2	C1	5, 6
A2	B2	C1	10, 16
A1	B1	C2	7, 15
A2	B1	C2	8, 11
A1	B2	C2	4, 9
A2	B2	C2	12, 14

Why and Why Not?

- The number of combinations grows rapidly
  - 8 for 2^3 and 16 for 2^4
- Effects of high-order interactions are negligible
  - It can then be simplified to consider only low-order interactions
- The major use is to identify the factors that have large effects on the system

Bibliography

- C. Wohlin et al. **Experimentation in Software Engineering**, Springer. 2012.
  - Section 8.5 – Experiment Design