Basic Experimental Designs

• Between-groups design
  • Each subject only does one variant of the experiment
  • There are at least 2 groups to isolate effect of manipulation:
    • Treatment group and control group
  + No practice effects across variants
  • Good for tasks that are simple and involve limited cognitive processes, e.g., tapping, dragging, or visual search
  - But: requires more users

• Within-groups design
  • Each subject does all variants of the experiment
  + Fewer users required, individual differences canceled out
  • Good for complex tasks, e.g., typing, reading, composition, problem solving
  - But: practice effects may occur
Basic Experimental Designs

Between-groups design

Within-groups design
Order Effects

• Within-groups design

• The behavior may be influenced by experience that occurred earlier in the sequence

• Carryover effects: changes caused by the lingering aftereffects of an earlier treatment condition.
  
  • E.g., testing the first condition causes users finger to hurt, degrading their performance in the second condition

• Progressive error: changes that are related to general experience in the study but unrelated to specific treatments
  
  • Practice effects and fatigue
  
  • E.g., the experiment takes too long overall
Counterbalancing

• Use every possible order of treatments with an equal number of individual participants

• Latin Square
  • Each condition appears at each ordinal position
  • Each condition precedes and follows each condition one time

• Example: six treatments: A, B, C, D, E, F
• **Learning curve**: relationship between experience (or time) and performance

• Rapid raise at the beginning, followed by a plateau

• In general, start measuring when the learning effect is gone!
Exercise #1

• For text entry research, describe one reason to choose a
  • Within-group design
  • Between-groups design
Experimental Design in Text Entry Research

• Usually preferred: within-group design
  • Minimizes confounding effects from the behavioral differences between participants
• Sometimes, we need a between-groups design
  • E.g., when testing whether a keyboard favors users with right-handedness over those with left-handedness
  • When there are interferences between conditions, e.g., different keyboard layouts on the same hardware
Criteria for a Good Paper

• Contribution: What new insight does it bring to the field?
• Benefits: What can one learn from this / do with this?
• Novelty: Prior publications?
• Validity: Are the claims properly backed up?
• Applicability: How good does the paper match the likely audience?
• Format: Readability and clarity
Structure of a (CHI) Review

- Overall rating: 1: definite reject – 5: definite accept
- Short summary of the contributions and benefits
  - “This paper presents… (who) will benefit from (what)”
- Concerns
  - Originality
  - Validity
  - Clarity
- Suggestions for improvement
- Reviewer’s expertise: 1: no knowledge – 4 expert
Reviewing Checklist

• Recommending accept
  • Convince yourself that it has no serious defects
  • Convince the editor that it is of an acceptable standard, by explaining why it is original, valid, and clear
  • List the changes that should be made before it appears in print
    • Where possible: indicating not just what to change but what to change it to
  • Take reasonable care in checking details, e.g., mathematics, formulas, and bibliography

• Recommending reject
  • Clearly explain the faults and, where possible, discuss how they could be rectified
  • Indicate which parts of the work are of value and which should be discarded
  • Check the paper to a reasonable level of detail

From Writing for Computer Science (Zobel, 2004)
Reviewing Checklist

• Always do the following in either case
  • Provide good references with which the authors should be familiar
  • Ask yourself whether your comments are fair, specific, and polite
  • Be honest about your limitations as a referee of that paper
  • Check your review carefully as you would check one of your own papers prior to submission

From Writing for Computer Science (Zobel, 2004)