CSE 599K

Empirical Research Methods

Winter 2025

Course introduction

Today

- Logistics and course overview
- Science vs. academia
- The scientific method

Logistics and course overview

The CSE 599K team

Instructor

- René Just (CSE2 338)
- Office hours: Wed 3pm – 4pm and by appointment
- rjust@cs.washington.edu

Teaching assistant

- Nino Migineishvili
- Office hours: TBD
- ninom@cs.washington.edu

Logistics

- CSE2 287, Mon/Wed, 1:30pm – 2:50pm.
- Lectures, discussions, presentations, and lab sessions.
- Course material, schedule, etc. on website: https://homes.cs.washington.edu/~rjust/courses/CSE599K
- Submission of assignments via Canvas: https://canvas.uw.edu

Your background and expectations

Introduction and a very brief survey

- **Field:** What is your research area/interest?
- **Stage:** How long have you been in the (BS/MS/PhD) program?
- **Experience:** What is your empirical research experience?
- **Top-2 expectations:** What do you expect from this course?

Course overview: the big picture

- **Week 1:** Introduction & the Science in CS
- **Week 2:** Qualitative vs. Quantitative Research
- **Week 3:** (Revised) Campbellian Validity system
- **Week 4:** Software Engineering meets Science & Preregistration
- **Week 5:** Data Wrangling
- **Week 6:** Parametric vs. non-parametric statistics
- **Week 7:** Common statistical methods
- **Week 8:** (Generalized) linear models
- **Week 9:** Data visualization and reporting
- **Week 10:** Project presentations & wrap up

Course overview: this week

- **Week 1:** Introduction & the Science in CS
  - **One high-level paper:** Is computer science science?
  - **Project:** brainstorm project ideas

Course overview: the project

Logistics

- 2-3 team members (justified exceptions are possible)
- Synergies with **your work** are welcome!
- We are happy to provide/discuss project ideas.

Timeline

- **Week 3/4**: Project proposal and revision
- **Week 5/6**: Methodology and revision
- **Week 8**: Data collection and initial results
- **Week 10**: Presentation and final report

Questions?

Course overview: grading

- **50%** Class project
- **20%** Assignments
- **20%** Paper reviews
- **10%** Participation

In-class exercises (graded activities) have two parts

1. In-class part: Small-group work on a problem set
2. Take-home part: Reflection and submission of deliverables

Questions?

Course overview: the even bigger picture

Other (UW) resources

- INFO 270: Calling Bullshit: Data reasoning in a digital world
  https://callingbullshit.org
- Practical Statistics for HCI
  https://depts.washington.edu/madlab/proj/ps4hci/
- Statistical Analysis and Reporting in R
  http://depts.washington.edu/madlab/proj/Rstats/

Course overview: expectations

- Engage in discussions
- Reason about research design and validity
- Read a few research papers
- Conduct a quarter-long research project
- Have fun!

Science vs. academia

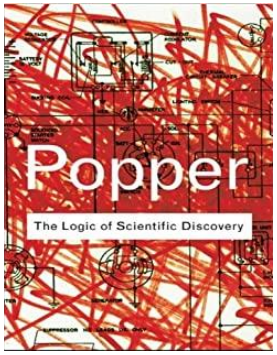
Science vs. academia

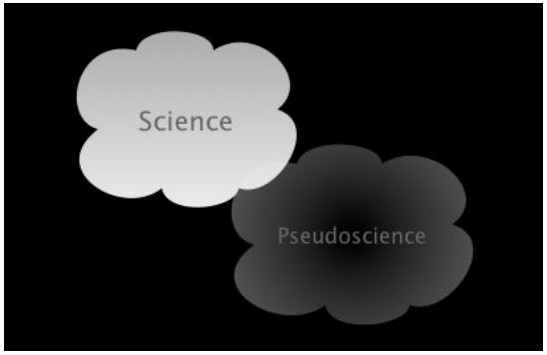
What's the difference between science and academia?

- How are they related?
- How are they different?

The scientific method

The holy grail: objectivity in science





The scientific method

Observation

The scientific method

Observation → Research question

The scientific method

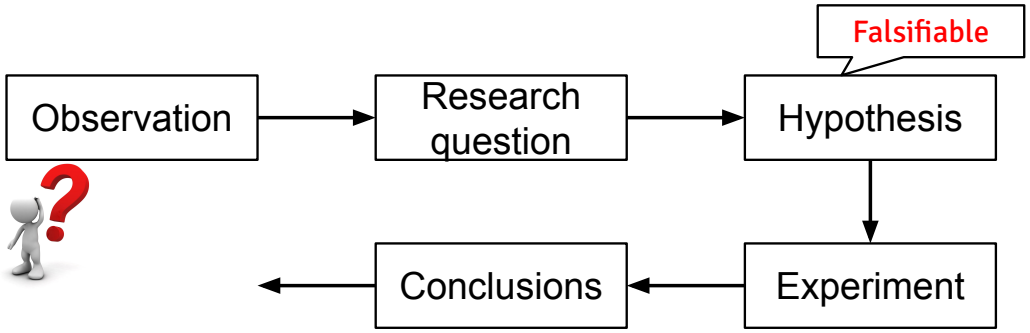
Observation → Research question → Hypothesis → Experiment → Conclusions

The scientific method

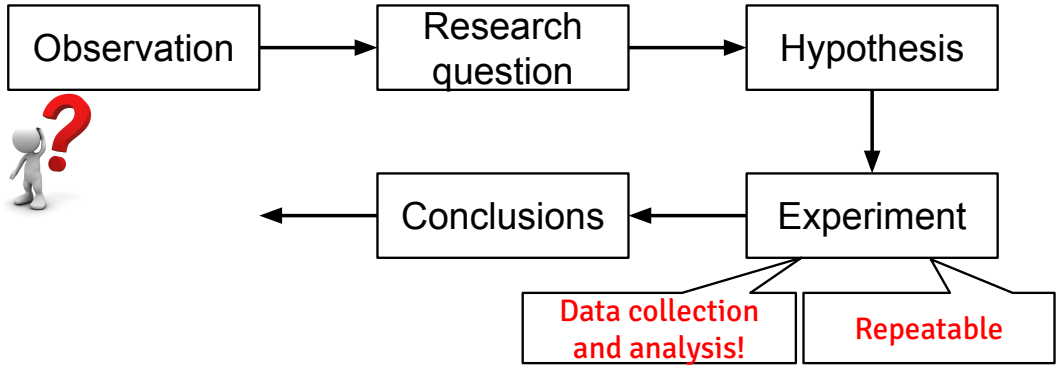
Observation → Research question → Hypothesis → Experiment → Conclusions

Operationalization/hypothesis formalization

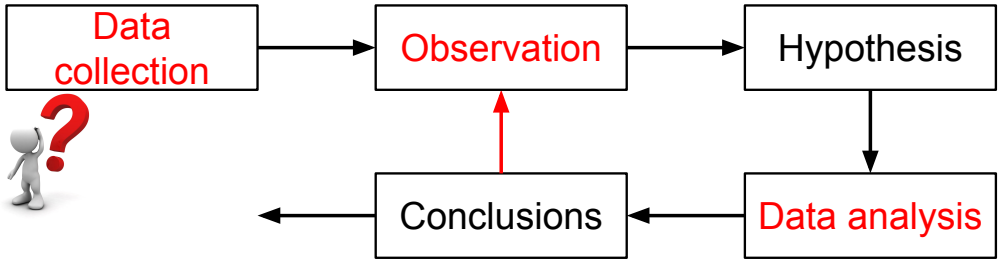
The scientific method



The scientific method

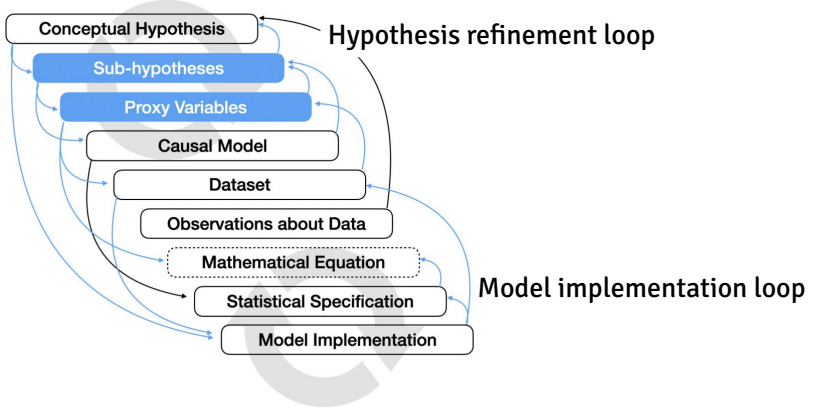


The scientific method: common mistake



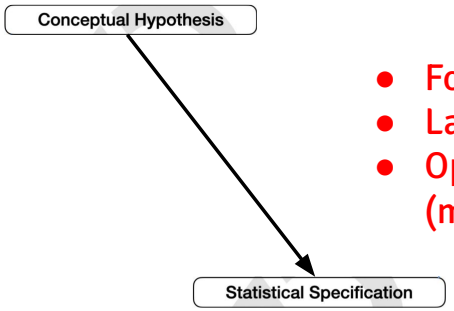
"If you torture the data long enough, it will confess."
[Ronald Harry Coase]

A more nuanced view: hypothesis formalization



Hypothesis formalization: Empirical findings, software limitations, and design implications, Jun et al., TOCHI 2022

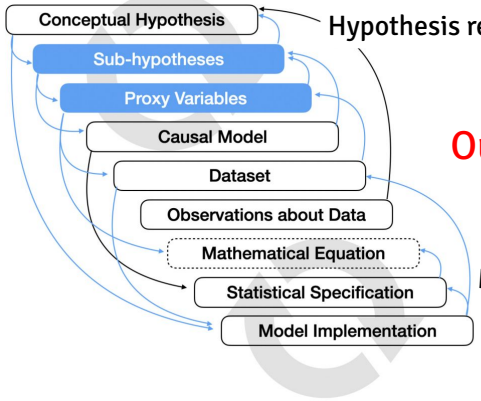
A more nuanced view: common mistake



- Focuses on statistical results
- Lacks a clear conceptual model
- Operationalization is implicit (mostly expressed in source code)

Hypothesis formalization: Empirical findings, software limitations, and design implications, Jun et al., TOCHI 2022

A more nuanced view: hypothesis formalization



Our goal is to explicate all steps.

Hypothesis formalization: Empirical findings, software limitations, and design implications, Jun et al., TOCHI 2022

A more nuanced view: a concrete example

Context

- We developed a new tool *AutoPatcher* that automatically fixes SW bugs.
- Currently, the tool *AutoCoder* is considered SOTA (state of the art).

Guiding question

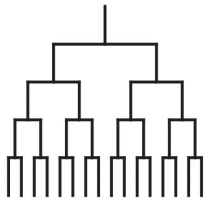
- Is *AutoPatcher* better than *AutoCoder*?

How do we operationalize this guiding question?

Is AutoPatcher better than AutoCoder ?

1. Define proxy for patch success (plausible vs. correct)
2. Choose evaluation benchmark (A-bench vs. B-bench)
3. Aggregation (mean vs. median)
4. Choose statistical test (T vs. U)

Design space

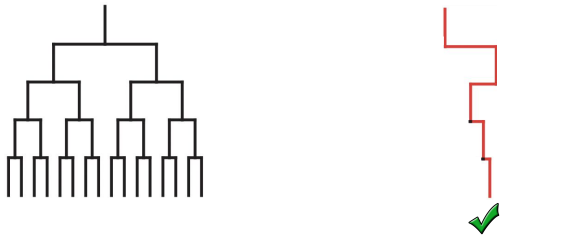


This is an oversimplification.
The actual design space is much larger.

Is AutoPatcher better than AutoCoder ?

1. Define proxy for patch success (plausible vs. correct)
2. Choose evaluation benchmark (A-bench vs. B-bench)
3. Aggregation (mean vs. median)
4. Choose statistical test (T vs. U)

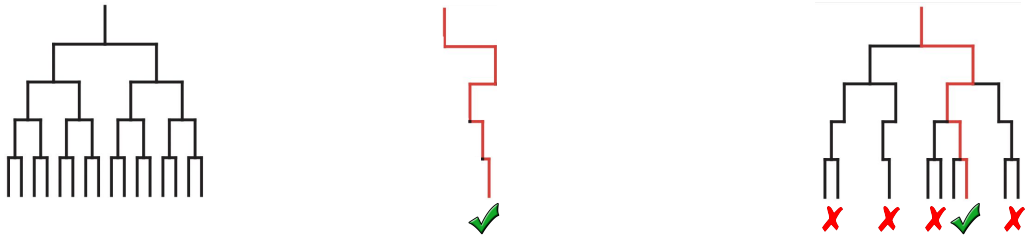
Design space **Reported design**



Is AutoPatcher better than AutoCoder ?

1. Define proxy for patch success (plausible vs. correct)
2. Choose evaluation benchmark (A-bench vs. B-bench)
3. Aggregation (mean vs. median)
4. Choose statistical test (T vs. U)

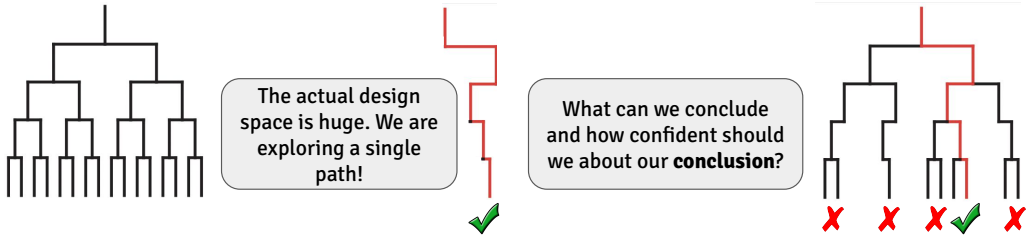
Design space **Reported design** **Alternative designs**



Is AutoPatcher better than AutoCoder ?

1. Define proxy for patch success (plausible vs. correct)
2. Choose evaluation benchmark (A-bench vs. B-bench)
3. Aggregation (mean vs. median)
4. Choose statistical test (T vs. U)

Design space **Reported design** **Alternative designs**



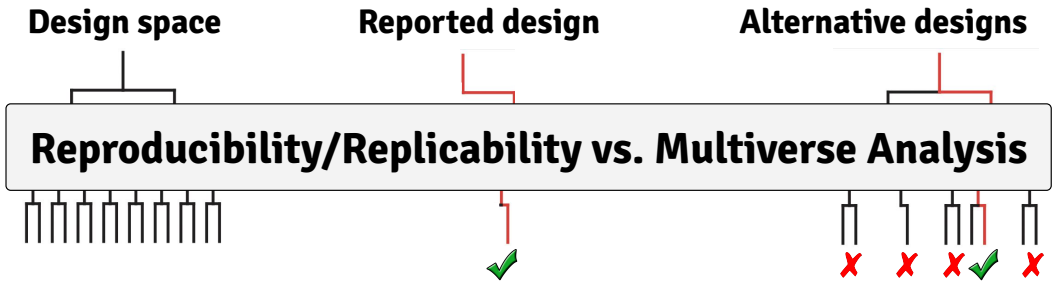
Is AutoPatcher better than AutoCoder ?

1. Define proxy for patch success (plausible vs. correct)
2. Choose evaluation benchmark (A-bench vs. B-bench)
3. Aggregation (mean vs. median)
4. Choose statistical test (T vs. U)

Design space **Reported design** **Alternative designs**

Is AutoPatcher better than AutoCoder ?

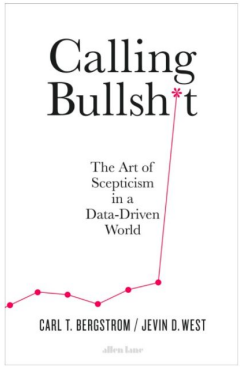
1. Define proxy for patch success (plausible vs. correct)
2. Choose evaluation benchmark (A-bench vs. B-bench)
3. Aggregation (mean vs. median)
4. Choose statistical test (T vs. U)



Empirical research: a simplified checklist

- Analysis grounded in a **conceptual model?**
- Clear **operationalization (implementation)?**
- **Implementation consistent with** the **model?**
- **Proper** use of **statistical methods?**
- Data interpreted in **context** of **prior knowledge?**
- Explored and validated **alternative hypotheses**?

Why should you care?



Report valid claims based on reproducible research.

Why I care: my favorite quotes

Collaborators, students, reviewers:

- These results are bad and cannot be true.
- If you don't trust my intuition, run your own experiments.
- These results are entirely expected.
- I have computed all the data; which statistical test should I use to show that my results are significant?
- Most papers are wrong or later obsolete, so who cares?
- I don't understand these intervals, can you give a p value?

Why I care: my favorite quotes

Collaborators, students, reviewers:

- These results are bad and cannot be true.
- If you don't trust my intuition, run your own experiments.
- These results are entirely expected.
- I have computed all the data; which statistical test should I use to show that my results are significant?
- Most papers are wrong or later obsolete, so who cares?
- I don't understand these intervals, can you give a p value?

Avoid confirmation bias; always assume you screwed up :)

Why I care: my favorite quotes

Collaborators, students, reviewers:

- These results are bad and cannot be true.
- If you don't trust my intuition, run your own experiments.
- These results are entirely expected.
- I have computed all the data; which statistical test should I use to show that my results are significant?
- Most papers are wrong or later obsolete, so who cares?
- I don't understand these intervals, can you give a p value?

Transform intuition and expectations into testable hypotheses!

Why I care: my favorite quotes

Collaborators, students, reviewers:

- These results are bad and cannot be true.
- If you don't trust my intuition, run your own experiments.
- These results are entirely expected.
- I have computed all the data; which statistical test should I use to show that my results are significant?
- Most papers are wrong or later obsolete, so who cares?
- I don't understand these intervals, can you give a p value?

"Statistical significance is the least interesting thing about the results"
[Sullivan and Fein: Using effect size -- or why the p value is not enough]

Next time

- The Science in CS
- Paper discussion: Is computer science science?