CSE 599F
Research Methods and Data Analysis in Software Systems Research
Winter 2019

Introduction to statistical analysis

January 24, 2019
Recap: Parametric vs. non-parametric statistics

Parametric statistics
- Assumptions about the underlying distribution.
  - Examples for common assumptions:
    - Normal distribution.
    - Equal variance.
- Parametric because of the reliance on distribution parameters.
- Example: Student’s t-test, Welch’s t-test.

Non-parametric statistics
- Fewer assumptions about the underlying distribution.
- Rank-based -> more robust to outliers.
- Example: Mann Whitney u test (Wilcoxon rank sum test).
Testing paired samples

Can we use a parametric test for these distributions?
When testing paired (dependent) samples, normality is assumed for the difference!
Today

- Logistics
- Data analysis of a real, small data set
  (aka hacking an R script that works :))

2. **A large scale study of programming languages and code quality in github**
   [https://dl.acm.org/citation.cfm?id=2635922](https://dl.acm.org/citation.cfm?id=2635922)

3. **Are mutation scores correlated with real fault detection?**
   A large scale empirical study on the relationship between mutants and real faults
   [https://dl.acm.org/citation.cfm?id=3180183](https://dl.acm.org/citation.cfm?id=3180183)

4. **Techniques for improving regression testing in continuous integration development environments**
   (at Google)
   [https://dl.acm.org/citation.cfm?id=2635910](https://dl.acm.org/citation.cfm?id=2635910)

5. **Revisiting Unsupervised Learning for Defect Prediction**
   [https://dl.acm.org/citation.cfm?id=3106257](https://dl.acm.org/citation.cfm?id=3106257)

6. **Why most published research findings are false**
   [https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0020124](https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0020124)
Data analysis in R: the set up

Problem:
Prove program equivalence between a program $P$ and each of many variants $V_n$

Approach:
- Encode programs as constraint systems and use an SMT solver
- Three strategies: naive, caching, forking
- Forking is only applicable to first-order variants

Data analysis:
- 3 subject programs (TicTacToe, Tax, Triangle).
- Between 99 and 267 variants per subject program.
- Each strategy ran 5 times on each variant.
Data analysis in R: the collected data

Skeleton R script: analysis.R

Data file: timing.csv

variant,naive,caching,forking,equivalent,first.order,run,subject
11, 309.8, 157.6, 144.8, 1, 1, 1, "tax"
12, 379.5, 237.4, 254.5, 0, 1, 1, "tax"
13, 415.9, 225.9, 225.9, 0, 0, 1, "tax"

Formulas in R

- Modeling dependent and independent variables
- Example: caching ~ subject + first.order

High-level research question:
- Is forking significantly better than caching/naive for all (first-order) variants?