Dataset Challenges in SE Data Science

What are the data sources?

Learning Objectives

1. Understand sources of software data.
2. Model the chance of finding an effect of interest.
3. Explore the limiations of software datasets.
4. Categorize sources of error in software research.

- Commits
- Issue trackers
- Source code
- Structured data
- Stack Overflow
-

What are some challenges?

1. Using the data that is available, instead of the data that answers the question.
2. Ignoring domain assumptions that produced the data, e.g. the human aspects (Secret Life of Bugs)
3. Sampling opportunistically (convenience) instead of systematically.
4. Not understanding the sensitivity of your instrument (the Kangaroo problem)
5. Questionable research practices

One of the biggest problems - and best places to spend money - is **getting good data**. The most obvious example of this is building big telescopes or high energy physics instruments.

Too often we go and use grad students to do a task instead of spending more time getting professionals. If you are looking for a small effect, you need to have a precision instrument!

Power Analysis

- Power is the probability before a study is done that it will achieve statistical significance for $\alpha < 0.05$ (or other values).
- First, we guess an "effect size" for our test (e.g., difference of means), then guess at sample size and variation.
- Most funders would expect studies to have power > 80%.
- Power analysis is inherently exploratory and hypothetical - simulation can help.
- What about study costs? Maybe it is cheap to collect data.
- See Fig 16.1 of Regression and other stories.


winners curse from Regression and Other Stories, Fig 16.1

Noisy data

- when noise is high and signal is low, statistically significant results are unlikely to replicate.
- type M - magnitude and type S - sign errors more likely in low-powered studies
- Best approach is to design a study to maximize effect size rather than population
- E.g., look at groups more likely to respond, increase treatment amounts
- Everything has an effect!

Effect sizes (briefly)

- Connect statistical outcomes (difference of means) to real-world impact.
- E.g. Cliff's D, R^2 (explained variance), Cohen's d, etc.
- Section 2.3 of my paper

> "effect size ignores the context of decision making. A raw number reflecting (for example) the standardized difference of means is hard for practitioners to interpret and must be contextualized.
> contextual, subjective judgment of observed effect sizes must be made and a ritualized interpretation avoided

- Bayesian analysis would implicitly ask us to do this: what effect is substantive? (avoid the golem!)

Secret Life of Bugs

What is the "secret"?

> The histories of even simple bugs are strongly dependent on social, organizational, and technical knowledge that cannot be solely extracted through automation of electronic repositories

Levels

1. Bug record data: bug tracker, resolution, people involved, commits associated
2. Automated conversation analysis: examining comments
3. Sense-making by informed observer
4. Direct narrative reports by participants

Stark differences in data available

Sampling

1. For many SE phenomena, there is **no suitable sampling frame**; that is, a list from which to draw a sample.
2. Some SE studies adopt poorly understood sampling strategies such as random sampling from **a non-representative surrogate population**.
3. Many SE articles evince **deep misunderstandings of representativeness**—the key criteria for assessing sampling in positivist research (see Section 2.6).

Sampling Approaches

1. Convenience
2. Purposive
3. Snowball
4. Probabilistic (random)
    i. Simple
    ii. Stratified

Sampling Challenges

1. Finding a representative sample: representativeness is the degree to which a sample's properties (of interest) resemble those of a target population
2. Randomness is not sufficient
3. Some philosophies do not view representativeness as desirable or relevant.
4. Software people are expensive
5. Real data is often under NDA
6. Software projects cover a vast universe of domains and contexts

   7 ... (crowd sourcing anecdote)

Bad Smells in Software Samples

– Incorrectly using the term “random” to mean arbitrary.
– Arguing that a convenience sample of software projects is representative because they are “real-world” projects.
– Assuming that a small sample is representative because it is random.
– Assuming that a large random sample is representative despite being selected from an obviously biased sampling frame.
– Implying that results should generalize to large populations without any claim to having a representative sample.
(Baltes and Ralph)

Exercise

Design a sampling strategy for the inferential question we considered earlier, i.e.

> (1) You are the manager of 10 teams at Spotify doing software development. Each team uses their own tools; some use Slack, others use email and IRC. Is there a connection between Slack use and team reported productivity?

and

> (2) You are looking to see how many bugs exist in primarily machine-learning software (e.g., building and training ML models with Tensorflow) vs other types of software

Data Analytics Methodology Problems

Table 1: Software Analytics Bad Smells Summary

	"Bad smell"	Core problem	Impact	Remedy
1	Not interesting	The software analytics equivalent of how "many angels can dance on a pinhead". This may result from an absence of theory.	Research that has negligible software engineering impact	Dialogue with practitioners [6, 70, 51]
2	Not using related work	Unawareness of related work on (i) the research question and/or (ii) state of the art in relevant techniques.	(i) Unwitting duplication (ii) Not using state of the art methods (iii) Using unchallenging / outdated benchmarks	Read! Utilize systematic reviews whenever possible. Search for relevant theory or theory fragments. Utilize technical guidance on data analytics methods and statistical analysis. Speak to experts.
3	Using deprecated or suspect data e.g., D has been widely used in the past ...	Using data for convenience rather than for relevance	Data driven analysis is undermined [2]	Justify choice of data sets wrt the research question.
4	Inadequate reporting	Partial reporting of results e.g., only giving means without measures of dispersion and/or incomplete description of algorithms	(i) Study is not reproducible [72] (ii) meta-analysis is difficult or impossible.	Provide scripts / code as well as data. Provide raw results [41, 79].
5	Underpowered studies	Small effect sizes and little or no underlying theory	Under-powered studies lead to over-estimation of effect sizes and replication problems	Analyse power in advance. Be wary of searching for small effect sizes, avoid Harking and p-hacking [15, 53, 79].
6	$p < 0.05$ and all that!	Over-reliance on, or abuse of, null hypothesis significance testing	A focus on statistical significance rather than effect sizes leading to vulnerability to questionable research practices and selective reporting [53]	Report effect sizes and confidence limits [33, 65].
7	Assumptions of normality and equal variances in statistical analysis	Impact of heteroscedasticity and outliers e.g., heavy tails ignored. Assumptions of analysis ignored.	Under-powered studies lead to over-estimation of effect sizes and replication problems.	Use robust statistics / involve statistical experts [58, 110].
8	No data visualisation	Simple summary statistics e.g., means and medians may mask underlying problems or unusual patterns.	Data are misinterpreted and anomalies missed.	Employ simple visualisations for the benefit of the analyst and the reader [80, 44].
9	Not exploring stability. Only reporting best or averaged results.	No sensitivity analysis. p-hacking	The impact of small measurement errors and small changes to the input data are unknown but potentially substantial. This is particularly acute for complex models and analyses.	Undertake sensitivity analysis or bootstraps/randomisation to understand variability. Minimally report all results and *variances* [73, 87].
10	Not tuning	Biased comparisons e.g., some models / learners tuned and others not. Non-reporting of the parameter settings, the tuning effort / expertise required.	Under-estimation of the performance of un-tuned predictors. Inability to reproduce results.	Read and use technical guidance on data analytics methods and statistical analysis [99, 37]. Unless relevant to the research question avoid off-the shelf defaults for sophisticated learners.
11	Not exploring simplicity	Excessively complex models, particularly with respect to sample size	Dangers of over-fitting	Independent validation sets or uncontaminated cross-validation [18].
12	Not justifying choice of learner	Permuting / recombining learners to make 'new' learners in a quest for spurious novelty	Under / non reporting of "uninteresting results" leading to over-estimates of effect sizes.	Principled generation of research questions and analysis techniques. Full reporting.

1. Not interesting
2. Not using related work
3. Deprecated or suspect data
4. Inadequate Reporting
5. Underpowered
6. Emphasizing p-values
7. Misunderstanding data distributions
8. No visualization
9. No stability analysis
10. Not tuning
11. Over complicating
12. No rationale for learner choice

See also ACM SigSOFT Empirical Review Standards

Git Promises and Perils

- **promises**: analysis Git enables
- **perils**: dangers with relying on Git

  in general: the data we see may not be an impartial log of events
- Git new on the scene at the time (replacing SVN/CVS)
- Git is distributed
- Git histories can be cleaned
  - delete/omit commits, squash / condense commits into one
  - lose the rich history of how a change was arrived at (Tesla example)
- Git branches are potentially confusing

Github Promises and Perils

- Daniel's Venn diagram of Github projects, public, useful
- Repos =/= project (forks for PRs)
- Many projects inactive or uninteresting
- Some repos are for data storage or personal use
- PRs used in different ways.
- PRs don't record all commits.
- Most pull requests appear as non-merged even if they are actually merged.
- Many active projects do not conduct all their software development in GitHub

Other Challenges

- temporal pollution
- likes and stars
- bots
- gender and demographics