Classical Nonparametric Hypothesis Tests with Applications in Social Good

by

Kellie Ottoboni

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Philip B. Stark, Chair
Professor Jasjeet Sekhon
Dean Henry Brady

Spring 2019

Classical Nonparametric Hypothesis Tests with Applications in Social Good

Abstract

Classical Nonparametric Hypothesis Tests with Applications in Social Good

by

Kellie Ottoboni

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Philip B. Stark, Chair

Hypothesis testing has come under fire in the past decade as misuses have become increasingly visible. It is common to use tests whose assumptions don't reflect how the data were collected, and editorial policies of many journals reward "p-hacking" by setting the arbitrary threshold of 0.05 to determine whether a result merits publication. In fact, properly designed hypothesis tests are an invaluable tool for inference and decision-making. Classical nonparametric tests, once reserved for problems that could be worked out with pencil and paper or approximated asymptotically, can now be applied to complex datasets with the help of modern computing power. This dissertation tailors some nonparametric tests to modern applications for social good.

Permutation tests are a class of hypothesis tests for data that involve random (or plausibly random) assignment. The parametric assumptions for common tests, like the t-test and linear regression, may not hold for randomized experiments; in contrast, the assumptions of permutation tests are *implied* by the experimental design. But off-the-shelf permutation tests are not a panacea: tests must be tailored to fit the experimental design, and there are subtle numerical issues with implementing the tests in software. We construct permutation tests and software to address particular questions in randomized and natural experiments, including identifying what, if anything, student evaluations of teaching measure, and whether voting machines malfunctioned in Georgia's November 2018 election.

Risk-limiting post-election audits (RLAs) have existed for a decade, but have not been adopted widely, in part due to logistical hurdles. This thesis uses classical nonparametric techniques, including Fisher's combination method and Wald's sequential probability ratio test, to build new RLA methods that accommodate the idiosyncratic logistics of statewide elections. A new, more flexible method for using stratified samples in RLAs makes it easier and more efficient to audit elections conducted on heterogeneous voting equipment. This thesis also develops an RLA method based on Bernoulli sampling, which allows ballots to be audited "in parallel" across precincts on Election Day. The RLA method for stratified samples of ballots was piloted in Michigan to study its performance in the face of real-world constraints.

To my parents, who always remind me to trust my gut and push me to keep learning

Contents

Contents ii

List of Figures v

List of Tables vi

1 Introduction 1
- 1.1 Preliminary concepts . . . 1
- 1.2 Goals and organization . . . 4

I Permutation Tests 7

2 Permutation tests for student evaluations of teaching 8
- 2.1 Introduction . . . 8
- 2.2 Data . . . 10
- 2.3 Methods . . . 11
- 2.4 The French natural experiment . . . 16
- 2.5 The US randomized experiment . . . 20
- 2.6 Multiplicity . . . 24
- 2.7 Code and data . . . 25
- 2.8 Discussion . . . 26

3 Pseudo-random number generators and permutations 28
- 3.1 Introduction . . . 28
- 3.2 Pseudo-random number generators . . . 29
- 3.3 Counting permutations and samples . . . 31
- 3.4 Sampling algorithms . . . 34
- 3.5 Tests of pseudo-randomness . . . 36
- 3.6 Discussion . . . 38

4 Software for randomization 40
- 4.1 Introduction . . . 40

4.2 Software projects . . . 42
4.3 Development practices . . . 46
4.4 Challenges . . . 50
4.5 Conclusion . . . 51

II Election Integrity: Security and Risk-Limiting Audits 52

5 Election integrity and electronic voting machines in 2018 Georgia 53
5.1 Introduction . . . 53
5.2 Background . . . 54
5.3 The 2018 Georgia election . . . 61
5.4 Conclusion . . . 64

6 Risk-limiting audits by stratified union-intersection tests of elections (SUITE) 67
6.1 Introduction . . . 67
6.2 Stratified audits . . . 68
6.3 Auditing cross-jurisdictional contests . . . 71
6.4 Comparison audits of overstatement quotas . . . 73
6.5 Ballot-polling audits of overstatement quotas . . . 76
6.6 Maximizing Fisher's combined p-value for a hybrid stratified audit . . . 78
6.7 Numerical examples . . . 80
6.8 Discussion . . . 81

7 Challenges of using SUITE in practice 82
7.1 Software . . . 83
7.2 Michigan pilot RLAs . . . 84
7.3 Sampling ballots . . . 87
7.4 Using comparison audits in hybrid SUITE . . . 88
7.5 Conclusions . . . 92

8 Bernoulli ballot polling: A manifest improvement for risk-limiting audits 93
8.1 Introduction . . . 93
8.2 Notation and mathematical background . . . 95
8.3 Performing the audit . . . 97
8.4 Initial sampling rate . . . 99
8.5 Implementation . . . 100
8.6 Evaluation . . . 102
8.7 Discussion . . . 104
8.8 Conclusion . . . 106

A Appendix for Chapter 3 107

A.1 Derangements 107
A.2 Fixed points of a permutation 108

Bibliography **110**

List of Figures

3.1 Algorithm for using a hash function in counter mode to generate PRNs. 32

4.1 Permutation testing algorithm used for all functions in `permute`. 42
4.2 NPC algorithm to test $H_0 : \cap_{k=1}^{K} H_{0k}$ implemented in `permute.npc`. 45
4.3 The "wheel" model of collaboration using GitHub. 47

5.1 Timeline of events pertaining to the use of DREs in Georgia since HAVA 54

7.1 Estimated workload (sample size) for an RLA of 13.7 million ballots with a 5% risk limit. The sample size is plotted on a logarithmic scale. 91

8.1 Bernoulli ballot-polling audit step-by-step procedures. 95
8.2 Simulated quantiles of sample sizes by fraction of votes for the winner for a two candidate race in elections with 10,000 ballots and 1 million ballots, for BRAVO ballot-polling audits (BPA) and Bernoulli ballot polling audits (BBP), for various risk-limits. The simulations assume every ballot has a valid vote for one of the two candidates. 103

List of Tables

2.1 Summary statistics of sections in the French university dataset. 11
2.2 Average correlation between SET and final exam score, by subject in the French university. 17
2.3 Average correlation between SET and instructor gender in the French university. 18
2.4 Average correlation between final exam scores and instructor gender in the French university. 18
2.5 Average correlation between SET and gender concordance in the French university. 19
2.6 Average correlation between student performance and gender concordance in the French university. 20
2.7 Average correlation between SET and interim grades in the French university. . 20
2.8 Difference in mean ratings by reported instructor gender (male minus female) in the US university. 23
2.9 Association between SET and reported instructor gender (male minus female) in the US university. 23
2.10 Association between SET and actual instructor gender (male minus female) in the US university. 24
2.11 Difference in mean grades by instructor gender (male minus female) in the US university. 24

3.1 The pigeonhole principle applied to PRNGs, samples, and permutations. For a PRNG of each size state space, the table gives examples where some samples or permutations must be unobtainable. 33
3.2 PRNGs and sampling algorithms used in common statistical and mathematical software packages. The 'floor' algorithm is the flawed multiply-and-floor method of generating pseudo-random integers. The 'mask' algorithm is better. 39

5.1 Counties with statistically significant ($p < 0.0001$) disparities in undervote rates between paper ballots and DREs. 63
5.2 Consistency of Results across DREs in Winterville Train Station Polling Place . 65
5.3 Consistency of Results across DREs in Winterville Train Station Polling Place, if D and R were flipped on machine 3. 65

7.1 Summary of the races audited. Ballots and votes for the candidates are the results as reported. Margins are expressed as a percentage of total ballots cast. The ballots included in the contest under audit in Lansing were restricted for the purpose of the pilot; half of Election Day precincts were omitted. 85

8.1 Estimated sampling rates needed for Bernoulli ballot polling for a 2-candidate race with a 5% risk limit. These simulations assume the reported margins were correct. 100

8.2 8 states in the 2016 US Presidential election for which a 1% BBP audit would have less than 99% chance of confirming the outcome. 104

Acknowledgments

Tukey's quote about statisticians "playing in everyone's backyard" has been overused, but the sentiment is the reason I was drawn to statistics and why I chose to work with my advisor, Philip Stark. Thanks to Philip, I've gotten to play in many backyards during my PhD. I admire his attention to detail and crisp communication, that his work is inspired by curiosity and desire to do good in the world, his willingness to work alongside me as an equal partner in research, and his insistence on doing work that is scientifically *and* morally right. He has championed me and opened unexpected doors for me. I'm incredibly grateful for his guidance and mentorship.

The Berkeley Institute for Data Science has played a central role in my graduate school experience. I'm grateful for their financial support over the years, in addition to endless opportunities to interact with and learn from academic data scientists all over the university and beyond. I did much of the work for this dissertation in the BIDS workspace, which feels like my home on campus.

I am grateful for research and travel funding from many sources over the years: the UC Dissertation-Year Fellowship, the Berkeley Institute for Data Science, State Street, the Berkeley Graduate Division, the Statistics Department KAG award, Microsoft Research, the Institute of Mathematical Statistics, and the Verified Voting Foundation.

I'm incredibly grateful to everyone who made the 2018 Michigan pilot RLAs possible. What I learned in those three days gave me more context for my research than reading a thousand papers could have. Thanks to our team of experts who travelled together and worked to make sure everything ran smoothly: Liz Howard, Monica Crane Childers, Mayuri Sridhar, Ron Rivest, and Jerome Lovato. It was only possible because the city clerks Tina Barton of Rochester Hills, Chris Swope of the city of Lansing, and Scott Borling of the city of Kalamazoo, the Michigan Bureau of Elections, led by Ginny van der Roest, volunteered to be the first to use my SUITE tool.

I'm grateful to Luigi Salmaso for inviting me to visit him at the University of Padova not once, but twice during my PhD. I had always dreamt of traveling Italy and visiting the town where my family comes from. Luigi's support and hospitality made it possible. I enjoyed our collaboration. My trips to Italy broadened my ideas about academic statistics, my role in the field, and the unlimited directions my life could go.

For lots of helpful conversations about both research and life, I'd like to thank my mentors, colleagues, and coauthors: Matt Bernhard, Anne Boring, Henry Brady, J. Alex Halderman, Mark Lindeman, Neal McBurnett, Jarrod Millman, Fernando Perez, Ron Rivest, Jas Sekhon, Stefan van der Walt, Bin Yu, and Wentao Zhan.

I'd be lost without my parents, my sister, my grandmothers, and my dear friends Rebecca Barter and Jamie Murdoch. This dissertation wouldn't have been possible without all their love and support over the years.

Chapter 1

Introduction

Modern inference problems can benefit from classical nonparametric tests. This dissertation demonstrates ways to develop nonparametric tests to solve domain-specific problems. We focus on applications concerning human preference: student evaluations of teachers and elections. These tests are designed to handle the characteristics of the data, as well as the human factors that go into collecting the data. In this chapter, we present background for key statistical concepts that appear throughout the dissertation.

1.1 Preliminary concepts

Permutation tests

Permutation tests (sometimes called randomization tests) are a class of hypothesis tests appropriate when, under the null hypothesis, the distribution of some statistic is invariant under transformations of a finite group. R. A. Fisher developed permutation tests in the context of randomized experiments, where the distribution of post-treatment data are invariant under different random allocations of treatment that fix the number of subjects per treatment group [31]. E. J. G. Pitman developed the same significance test along with a more general theory of permutation testing motivated by the problem of inference without population-level distributional assumptions [99]. Applications of permutation tests extend beyond randomized experiments to scenarios where the null hypothesis implies some group invariance, including tests for symmetry about a point, tests for stochastic dominance, and tests for correlation between variables [111].

The two-sample experiment may be the clearest illustration of how permutation tests work. Suppose we are interested in whether a treatment has a protective effect against cancer. In an experiment with $n_1 + n_2$ individuals, we randomly assign treatment to n_1 individuals and the remaining n_2 receive placebo. After following subjects for 10 years, we determine the proportions of individuals $\hat{p}_1$ and $\hat{p}_2$ who developed cancer in the treatment group and in the control group, respectively. Assuming we may ignore issues of study

drop-out and non-compliance, we conclude that the treatment reduced the risk of cancer by $\hat{p}_1 - \hat{p}_2$.

How significant is this risk difference? Instead of deciding how "extreme" the observed test statistic is relative to a parametric reference distribution, we judge it against its permutation distribution. Deliberate randomization of treatments induces a distribution for the test statistic under the null hypothesis of no effect of treatment on the outcome. This is called the "sharp" null hypothesis, because it provides detail about each individual's potential outcome had they been assigned to the other group [81]. (In contrast, the "weak" null hypothesis says that the treatment has no effect *on average*.) The randomization scheme specifies that any way of allocating treatment to n_1 subjects and placebo to n_2 subjects was equally likely to have occurred. The null hypothesis tells us that each individual's response would have been the same regardless of the assignment. Using the randomization scheme and imputing the unobserved, counterfactual responses under the null hypothesis, we can recover the distribution of risk differences that could have occurred with the $n_1 + n_2$ individuals in this experiment. Chapter 2 describes two more complex experiments and the appropriate permutation schemes for each.

To enumerate all possible randomizations is not practical for even moderate-sized datasets, as the number of possible permutations grows exponentially with the sample size. Instead, one typically estimates the randomization distribution by simulation. This step crucially depends on the assumption that the software used to construct permutations actually does so with equal probabilities, so the approximate randomization distribution is like a simple random sample (SRS) from the true distribution of all randomizations. As discussed in Chapter 3, not all pseudo-random number generators behave "randomly enough" for all simulations.

When a permutation test is available, it should be used in place of the corresponding parametric test. Permutation tests are exact, meaning that they control the type I error rate at the pre-specified level, even in finite samples; parametric hypothesis tests based on asymptotic approximations do not always guarantee good finite-sample properties [32].

Wald's sequential probability ratio test

Continuously monitoring a p-value as data come in is problematic: stopping as soon as the p-value drops below the desired significance level or collecting more data until the test reaches significance will generally inflate the type I error rate. However, there are many situations where one would like to collect data incrementally until there is conclusive evidence in favor or against the null hypothesis. [145] developed the sequential probability ratio test (SPRT) to solve this problem. The procedure is based on iteratively calculating the likelihood ratio for the data after additional observations and reaching a decision once the likelihood ratio passes a threshold.

We'll illustrate the SPRT using the simplest case of binary data. Suppose we observe independent and identically distributed (IID) Bernoulli random variables $X_1, X_2, \ldots$ with probability of success p. We'd like to test the null hypothesis $H_0 : p = p_0$ against the

alternative $H_1 : p = p_1 > p_0$ with level α and power $1 - \beta$. After m trials, there have been $T_m \equiv \sum_{i=1}^{m} X_i$ successes. At this point, the likelihood ratio is

$$\frac{p_{1m}}{p_{0m}} \equiv \frac{p_1^{T_m}(1-p_1)^{m-T_m}}{p_0^{T_m}(1-p_0)^{m-T_m}}.$$

We reject the null hypothesis if the evidence for H_1 outweighs the evidence for H_0, such that

$$\frac{p_{1m}}{p_{0m}} \geq \frac{1-\beta}{\alpha}.$$

We stop the test in favor of the null hypothesis if

$$\frac{p_{1m}}{p_{0m}} \leq \frac{\beta}{1-\alpha}.$$

If $\frac{\beta}{1-\alpha} < \frac{p_{1m}}{p_{0m}} < \frac{1-\beta}{\alpha}$, more samples should be drawn.

For any $\alpha \in (0,1)$ and $\beta \in (0,1)$, the SPRT has level no larger than α and has power at least $1 - \beta$ against the alternative H_1. Amongst sequential tests with this property, the SPRT approximately minimizes the expected sample size when the true parameter $p = p_0$ or $p = p_1$. For values $p_0 < p < p_1$, the SPRT can have larger sample sizes than fixed-sample-size tests.

If $\beta = 0$, then the inverse of the terminal likelihood ratio (the value when we stop the test after it passes above the decision threshold $1/\alpha$) can be interpreted as a p-value. This is useful when implementing the SPRT as a hypothesis test of a single null, and stopping the test in favor of the null hypothesis is not of interest.

More generally, we can use the SPRT framework to construct a test for any pointwise null and alternative hypotheses. In Chapter 3, we use the SPRT as the basis for several tests for uniformity of pseudo-random samples. Moreover, the SPRT can be used with any distributions p_{0m} and p_{1m}. In Chapter 6, we develop a version of the SPRT using the tri-hypergeometric distribution with a nuisance parameter.

Risk-limiting audits

No method for counting votes is perfect, and methods that rely on computers are particularly fragile: errors, bugs, and deliberate attacks can alter results. The vulnerability of electronic voting was confirmed in two major state-funded studies, California's Top-to-Bottom Review [14] and Ohio's EVEREST study [73]. More recently, at the 2017 and 2018 DEFCON hacking conferences, attendees with little or no knowledge of election systems were able to penetrate a wide range of U.S. voting machines [9, 10]. Given that Russia interfered with the 2016 U.S. Presidential election through an "unprecedented coordinated cyber campaign against state election infrastructure" [88], national security demands we protect our elections from nation states and other advanced persistent threats.

A risk-limiting audit (RLA) of an election contest is a procedure that has a known minimum chance of leading to a full manual tally of the ballots if the electoral outcome according to that tally would differ from the reported outcome. *Outcome* means the winner(s) (or, for instance, whether there is a runoff)—not the numerical vote totals. RLAs require a durable, voter-verifiable record of voter intent, such as paper ballots, and they assume that this audit trail is sufficiently complete and accurate that a full hand tally would show the true electoral outcome. That assumption is not automatically satisfied: a *compliance audit* [134] is required to check whether the paper trail is trustworthy.

RLAs have been conducted in California, Colorado, Indiana, Ohio, Virginia, and Denmark, and are required by law in Colorado (CRS 1-7-515) and Rhode Island (RI Gen L § 17-19-37.4). RLA legislation is under consideration in a number of other states, and bills to require RLAs have been introduced in Congress. The National Academy of Sciences [80], the American Statistical Association [1], the 2013 Presidential Commission on Election Administration [137], and California's independent oversight agency, the Little Hoover Commission [60] endorse risk-limiting post-election audits.

Current methods for risk-limiting audits are generally sequential hypothesis testing procedures: they examine more ballots, or batches of ballots, until either (i) there is strong statistical evidence that a full hand tabulation would confirm the outcome, or (ii) the audit has led to a full hand tabulation, the result of which should become the official result. There are three types of RLAs commonly used: ballot-level comparison, batch-level comparison, and ballot polling.

Ballot-level comparison involves drawing a random sample of ballots and comparing the electronic interpretation of votes, called a *cast-vote record* or CVR, to a manual interpretation of voter intent. The hypothesis tests for these audits are based on Wald's SPRT or other probability inequalities. *Batch-level comparison* involves sampling entire batches (physical groupings of ballots) and comparing the reported vote totals in the batch to a manual tally. The math is an extension of the tests for ballot-level comparison. Batch-level comparison audits are less *efficient* (i.e., require examining more ballots) than ballot-level comparisons, especially as batch sizes grow large [87].

Ballot-polling audits are akin to exit polls: they involve drawing a simple random sample of ballots and using the observed number of votes for each candidate to weigh the evidence that the reported winner is correct. The hypothesis test for ballot polling in a plurality contest is based on Wald's SPRT [63, 62]. Ballot polling does not require CVRs, so it can be implemented in any jurisdiction with a paper record of voter intent. However, ballot-polling audits generally require examining more ballots than ballot-level comparison audits.

1.2 Goals and organization

The main contribution of Part I is to illustrate the utility of carefully designed permutation tests and issues that arise when implementing them in code. Parametric hypothesis tests used to be commonplace because asymptotic approximations were the only computationally

feasible way to estimate distributions. Computational power is no longer a barrier to finding exact (or with pre-specified precision) permutation distributions. However, parametric methods have remained the default approach in software, while more computationally intensive nonparametric methods have received less attention. Many researchers run carefully planned experiments or collect rich observational data, but then apply an inappropriate inference method in a statistical program. This perpetuates a cycle of bad science: researchers who don't have a strong statistical background will tend to use statistical methods that are readily available in the software, which are perpetuated in the field's literature.

Chapter 2 develops permutation tests for two studies of student evaluations of teaching to determine what, if anything, these surveys measure. The original analyses used parametric tests to measure the extent of gender bias in the students' evaluations; we provide a more rigorous analysis using approaches that more accurately reflect the studies' designs. We explain the assumptions of our models in depth to illustrate how permutation tests can be constructed for randomized and natural experiments.

Chapter 3 is an in-depth examination of the computational tools used to generate the null distributions for permutation tests: pseudo-random number generators and sampling algorithms. Using a pseudo-random number generator with an insufficient state space and using sampling algorithms that depend on floating point arithmetic can lead to unequal sampling frequencies in code intended to produce uniform random samples. This matters for permutation tests: approximating the null distribution crucially depends on the ability to generate random permutations with equal probability.

Chapter 4 discusses packages we wrote for permutation tests in Python. `permute` contains code to run common permutation tests and `cryptorandom` contains a cryptographically-secure pseudo-random number generator and library of sampling algorithms. We explain the code's functionality, the software development practices used, and how they can be useful to anyone who writes or uses statistical software.

The main contribution of Part II is to explore the intersection of post-election statistical forensics and the human element of elections. Risk-limiting audits have existed since 2008 [127, 129, 124], yet new challenges continue to arise as more cities and states pilot them. Conducting a risk-limiting audit requires meticulous record-keeping and coordination between jurisdictions. Arguably, states should already be doing this, but challenges remain: heterogeneous voting equipment across jurisdictions, laws that regulate the timing and procedures around audits, recounts, and ballot handling, and ad hoc procedures that election officials practice in their jurisdiction. We propose new developments that solve some logistical barriers to risk-limiting audits by modifying the sampling plan.

Of course, risk-limiting post-election audits can only be conducted if there is a durable, trustworthy voter-verifiable paper record. The state of Georgia uses legacy voting equipment that does not produce a paper record. Chapter 5 gives a brief history of the election security problems in Georgia since the early 2000s, leading up to the state's contentious 2018 midterm elections. We present statistical evidence, using permutation tests, of an unusually high rate of undervotes in the Lieutenant Governor's race amongst in-person voters who used legacy machines compared to voters who used paper ballots. Since risk-limiting audits are not

possible and a forensic examination of the voting machines has not been allowed, we cannot draw strong conclusions about what happened.

Chapter 6 presents a general method for risk-limiting audits using stratified samples of ballots, called SUITE. SUITE solves a problem for states like Colorado who would like to conduct risk-limiting audits of statewide contests with votes cast on different types of voting equipment: stratifying cast ballots by the equipment on which they were cast, and using the most efficient auditing method possible in each stratum, can be more efficient than using simple random sampling with a ballot-polling audit.

We tested SUITE in Michigan's 2018 pilot risk-limiting audits. Chapter 7 describes the lessons learned from bringing a statistical method from paper to practice. While SUITE solved the problem of heterogeneous voting equipment, we observed other, more fundamental challenges during the pilots, including difficulty retrieving sampled ballots and issues accessing the cast-vote record required for the comparison audit. Pilot audits are an important way for for academics and election officials to learn and refine how the methods operate in practice.

Chapter 8 presents a risk-limiting audit procedure based on Bernoulli random sampling. Instead of drawing a random sample of ballots all at once after the election, ballots can be sampled "in parallel" across precincts on Election Day. This can reduce the need for some record-keeping, simplify the logistics of conducting an audit of a large election, and help officials plan for the costs and labor required for the first portion of an audit. The statistical test to calculate the risk in a Bernoulli ballot-polling audit is a version of Wald's SPRT for tri-hypergeometric samples.

Part I

Permutation Tests

Chapter 2

Permutation tests for student evaluations of teaching

2.1 Introduction

Student evaluations of teaching (SET) are used widely in decisions about hiring, promoting, and firing instructors. Measuring teaching effectiveness is difficult—for students, faculty, and administrators alike. Universities generally treat SET as if they primarily measure teaching effectiveness or teaching quality. While it may seem natural to think that students' answers to questions like "how effective was the instructor?" measure teaching effectiveness, it is not a foregone conclusion that they do. Indeed, the best evidence so far shows that they do not: they have *biases*[1] that are stronger than any connection they might have with effectiveness. Worse, in some circumstances the association between SET and an objective measure of teaching effectiveness is *negative*, as our results below reinforce.

Randomized experiments [18, 15] have shown that students confuse grades and grade expectations with the long-term value of a course and that SET are not associated with student performance in follow-on courses, a proxy for teaching effectiveness. On the whole, high SET seem to be a reward students give instructors who make them anticipate getting a good grade, for whatever reason; for extensive discussion, see [52, Chapters 3–5].

Gender matters too. [13] finds that SET are affected by gender biases and stereotypes. Male first-year undergraduate students give more *excellent* scores to male instructors, even though there is no difference between the academic performance of male students of male and of female instructors. Experimental work by [66] finds that when students think an instructor is female, students rate the instructor lower on every aspect of teaching, including putatively objective measures such as the timeliness with which instructors return assignments.

In this chapter, we reanalyze two sets of data from [13] and [66]. Our statistical contribution is twofold. First, we apply nonparametric permutation tests to data to investigate

[1] [21, p.17] define bias to occur when "a teacher or course characteristic affects teacher evaluations, either positively or negatively, but is unrelated to criteria of good teaching, such as increased student learning."

whether SET primarily measure teaching effectiveness or biases using a higher level of statistical rigor than the original analyses which used parametric models. Permutation tests allow us to avoid contrived, counterfactual assumptions about parametric generative models for the data, which regression-based methods (including ordinary linear regression, mixed effects models, logistic regression, etc.) and methods such as t-tests and ANOVA generally require. The null hypotheses for our tests are that some characteristic—e.g., instructor gender—amounts to an arbitrary label and might as well have been assigned at random.

Second, we use this reanalysis as a pedagogical opportunity. The methods sections of most research articles are terse and list the statistical methods used. Instead, we explain the merits of each dataset and the mechanics of the tests we use in detail. Section 2.2 explains how the data from [13] and [66] are unique among studies of SET and how their design allows us to address questions that are normally difficult to answer due to confounding. Section 2.3 explains the null hypotheses and ideas behind permutation tests, which many readers may never have encountered. We hope that this extended discussion encourages readers to think more critically about the merits and limitations of future research on SET.

The two main sources of bias we study are students' grade expectations and the gender of the instructor. We also investigate variations in bias by discipline and by student gender. We work with course-level summaries to match how institutions use SET: typically, SET are averaged for each offering of a course, and those averages are compared across instances of the course, across courses in a department, across instructors, and across departments. [132] discuss statistical problems with this reduction to and reliance upon averages.

We find that the association between SET and an objective measure of teaching effectiveness —performance on the anonymously graded final— is weak and generally not statistically significant. In contrast, the association between SET and (perceived) instructor gender is large and statistically significant: instructors whom (students believe) are male receive significantly higher average SET.

In the French data, *male* students tend to rate male instructors higher than they rate female instructors, with little difference in ratings by female students. In the US data, *female* students tend to rate (perceived) male instructors higher than they rate (perceived) female instructors, with little difference in ratings by male students. The French data also show that gender biases vary by course topic, and that SET have a strong positive association with students' grade expectations.

We therefore conclude that SET primarily do not measure teaching effectiveness; that they are strongly and non-uniformly biased by factors including the genders of the instructor and student; that they disadvantage female instructors; and that it is impossible to adjust for these biases. SET should not be relied upon as a measure of teaching effectiveness. Relying on SET for personnel decisions has disparate impact by gender, in general.

2.2 Data

French natural experiment

These data, collected between 2008 and 2013, are a census of 23,001 SET from 4,423 first-year students at a French university students (57% women) in 1,177 sections, taught by 379 instructors (34% women). The data are not public, owing to French restrictions on human subjects data. [13] describes the data in detail. Key features include:

- All first-year students take the same six mandatory courses: History, Macroeconomics, Microeconomics, Political Institutions, Political Science, and Sociology. Each course has one (male) professor who delivers the lectures to groups of approximately 900 students. Courses have sections of 10–24 students. Those sections are taught by a variety of instructors, male and female. The instructors have considerable pedagogical freedom.

- Students enroll in "triads" of sections of these courses (three courses per semester). The enrollment process does not allow students to select individual instructors. The assignment of instructors to sets of students is as if at random, forming a *natural experiment*. It is reasonable to treat the assignment as if it is independent across courses.

- Section instructors assign interim grades during the semester. Interim grades are known to the students before the students submit SET. Interim grades are thus a proxy for students' grade expectations.

- Final exams are written by the course professor, not the section instructors. Students in all sections of a course in a given year take the same final. Final exams are graded anonymously, except in Political Institutions, which we therefore omit from analyses involving final exam scores. To the extent that the final exam measures appropriate learning outcomes, performance on the final is a measure of the effectiveness of an instructor: in a given course in a given year, students of more effective instructors should do better on the final exam, on average, than students of less effective instructors.

- SET are mandatory: response rates are nearly 100%.

SET include closed-ended and open-ended questions. The item that attracts the most attention, especially from the administration, is the *overall score*, which is treated as a summary of the other items. The SET data include students' individual evaluations of section instructors in Microeconomics, History, Political Institutions, and Macroeconomics for the five academic years 2008–2013, and for Political Science and Sociology for the three academic years 2010–2013 (these two subjects were introduced in 2010). The SET are anonymous to the instructors, who have access to SET only after all grades have been officially recorded.

Table 2.1: Summary statistics of sections in the French university dataset.

course	# sections	# instructors	% Female instructors
Overall	**1,194**	**379**	**33.8%**
History	230	72	30.6%
Political Institutions	229	65	20.0%
Microeconomics	230	96	38.5%
Macroeconomics	230	93	34.4%
Political Science	137	49	32.7%
Sociology	138	56	46.4%

Data for a section of Political Institutions that had an experimental online format are omitted. Political Science and Sociology originally were not in the triad system; students were randomly assigned by the administration to different sections.

US randomized experiment

These data, described in detail by [66], are available at `http://n2t.net/ark:/b6078/d1mw2k`. Students in an online course were randomized into six sections of about a dozen students each, two taught by the primary professor, two taught by a female graduate teaching assistant (TA), and two taught by a male TA. In one of the two sections taught by each TA, the TA used her or his true name; in the other, she or he used the other TA's identity. Thus, in two sections, the students were led to believe they were being taught by a woman and in two they were led to believe they were being taught by a man. Students had no direct contact with TAs: the primary interactions were through online discussion boards. The TA credentials presented to the students were comparable; the TAs covered the same material; and assignments were returned at the same time in all sections (hence, objectively, the TAs returned assignments equally promptly in all four sections).

SET included an overall score and questions relating to professionalism, respectfulness, care, enthusiasm, communication, helpfulness, feedback, promptness, consistency, fairness, responsiveness, praise, knowledge, and clarity. Fourty-seven students in the four sections taught by TAs finished the class, of whom 43 submitted SET. The SET data include the genders and birth years of the students;[2] the grade data do not. The SET data are not linked to the grade data.

2.3 Methods

Previous analyses of these data relied on parametric tests based on null hypotheses that do not match the experimental design. For example, the tests assumed that SET of male and female instructors are independent random samples from normally distributed populations with equal variances and possibly different means. As a result, the p-values reported in those studies are for unrealistic null hypotheses and might be misleading.

[2]One birth year is obviously incorrect, but our analyses do not rely on the birth years.

In contrast, we use permutation tests based on the as-if-random (French natural experiment) or truly random (US experiment) assignment of students to class sections, with no counterfactual assumption that the students, SET scores, grades, or any other variables comprise random samples from any populations, much less populations with normal distributions.

In most cases, our tests are *stratified*. For the US data, for instance, the randomization is stratified on the actual TA: students are randomized within the two sections taught by each TA, but students assigned to different TAs comprise different strata. The randomization is independent across strata. For the French data, the randomization is stratified on course and year: students in different courses or in different years comprise different strata, and the randomization is independent across strata. The null distributions of the test statistics[3] are induced by this random assignment, with no assumption about the distribution of SET or other variables, no parameter estimates, and no model.

Illustration: French natural experiment

The selection of course sections by students at the French university—and the implicit assignment of instructors to sets of students—is as if at random within sections of each course each year, independent across courses and across years. The university's triad system groups students in their classes across disciplines, building small cohorts for each semester. Hence, the randomization for our test keeps these groups of students intact. Stratifying on course topic and year keeps students who took the same final exam grouped in the randomization and honors the design of the natural experiment.

Teaching effectiveness is multidimensional [70] and difficult to define, much less measure. But whatever it is, effective teaching should promote student learning: *ceteris paribus*, students of an effective instructor should have better learning outcomes than students of an ineffective instructor have. In the French university, in all courses other than Political Institutions,[4] students in every section of a course in a given year take the same anonymously graded final exam. To the extent that final exams are designed well, scores on these exams reflect relevant learning outcomes for the course. Hence, in each course each semester, students of more effective instructors should do better on the final, on average, than students of less effective instructors.

Consider testing the hypothesis that SET are unrelated to performance on the final exam against the alternative that, all else equal, students of instructors who get higher average SET get higher final exam scores, indicating that they learned more. For this hypothesis test, we omit Political Institutions because the final exam was not anonymous.

The test statistic is the average over courses and years of the Pearson correlation between mean SET and mean final exam score among sections of each course each year. If SET do measure instructors' contributions to learning, we would expect this average correlation to

[3]The test statistics are correlations of a response variable with experimental variables, or differences in the means of a response variable across experimental conditions, aggregated across strata.

[4]The final exam in Political Institutions is oral and hence not graded anonymously.

be positive: sections with above-average mean SET in each discipline each year would tend to be sections with above-average mean final exam scores. How surprising is the observed average correlation, if there is no connection between mean SET and mean final exam for sections of a course?

There are 950 "individuals," course sections of subjects other than Political Institutions. Each of the 950 course sections has an average SET and an average final exam score. These fall in $3 \times 5 + 2 \times 3 = 21$ year-by-course strata. Under the randomization, within each stratum, instructors are assigned sections independently across years and courses, with the number of sections of each course that each instructor teaches each year held fixed. For instance, if in 2008 there were N sections of History taught by K instructors in all, with instructor k teaching N_k sections, then in the randomization, all

$$\binom{N}{N_1 \cdots N_K} \tag{2.1}$$

ways of assigning N_k of the N 2008 History sections to instructor k, for $k = 1, \ldots, K$, would be equally likely. The same would hold for sections of other courses and other years. Each combination of assignments across courses and years is equally likely: the assignments are independent across strata.

Under the null hypothesis that SET have no relationship to final exam scores, average final exam scores for sections in each course each year are *exchangeable* given the average SET for the sections. Imagine "shuffling" (i.e., permuting) the average final exam scores across sections of each course each year, independently for different courses and different years. For each permutation, compute the Pearson correlation between average SET for each section and average final exam score for each section, for each course, for each year. Average the resulting 21 Pearson correlations. The probability distribution of that average is the null distribution of the test statistic. The p-value is the upper tail probability beyond the observed value of the test statistic, for that null distribution.

The hypothetical randomization holds triads fixed, to allow for cohort effects and to match the natural experiment. Hence, the test is conditional on which students happen to sign up for which triad. However, if we test at level no greater than α conditionally on the grouping of students into triads, the unconditional level of the resulting test across all possible groupings is no greater than α:

$$\begin{aligned}
\Pr\{\text{ Type I error }\} &= \sum_{\text{all possible sets of triads}} \Pr\{\text{ Type I error } | \text{ triads }\} \Pr\{\text{ triads }\} \\
&\leq \sum_{\text{all possible sets of triads}} \alpha \Pr\{\text{triads }\} \\
&= \alpha \sum_{\text{all possible sets of triads}} \Pr\{\text{triads }\} \\
&= \alpha. \qquad (2.2)
\end{aligned}$$

It is not practical to enumerate all possible permutations of sections within courses and years, so we estimate the p-value by performing 10^5 random permutations within each stratum, finding the value of the test statistic for each overall assignment, and comparing the observed value of the test statistic to the empirical distribution of those 10^5 random values. The probability distribution of the number of random permutations of assignments for which the test statistic is greater than or equal to its observed value is Binomial, with n equal to the number of overall random permutations and p equal to the true p-value. Hence, the standard error of the estimated p-values is hence no larger than $(1/2)/\sqrt{10^5} \approx 0.0016$.[5] Code for all our analyses is at `https://github.com/kellieotto/SET-and-Gender-Bias`. Results for the French data are below in Section 2.4.

Illustration: US randomized experiment

To test whether perceived instructor gender affects SET in the US experiment, we use the Neyman "potential outcomes" framework [81]. A fixed number N of individuals—e.g., students or classes—are assigned randomly (or as if at random by Nature) into $k \geq 2$ groups of sizes $N_1, \ldots, N_k$. Each group receives a different treatment. "Treatment" is notional. For instance, the treatment might be the gender of the class instructor.

For each individual i, we observe a numerical response R_i. If individual i is assigned to treatment j, then $R_i = r_{ij}$. The numbers $\{r_{ij}\}$ are considered to have been fixed before the experiment. (They are not assumed to be a random sample from any population; they are not assumed to be realizations of any underlying random variables.) Implicit in this notation is the *non-interference* assumption that each individual's response depends only on the treatment that individual receives, and not on which treatments other individuals receive.

We observe only one potential outcome for individual i, depending on which treatment she or he receives. In this model, the responses $\{R_i\}_{i=1}^N$ are random, but only because individuals are assigned to treatments at random, and the assignment determines which of the fixed values $\{r_{ij}\}$ are observed.

In the experiment conducted by [66], N students were assigned at random to six sections of an online course, of which four were taught by TAs. Our analysis focuses on the four sections taught by TAs. We condition on the assignment of students to the two sections taught by the professor. Each remaining student i could be assigned to any of $k = 4$ treatment conditions: either of two TAs, each identified as either male or female. The assignment of students to sections was random: each of the

$$\binom{N}{N_1 N_2 N_3 N_4} = \frac{N!}{N_1! N_2! N_3! N_4!} \tag{2.3}$$

possible assignments of N_1 students to TA 1 identified as male, N_2 student to TA 1 identified as female, etc., was equally likely.

[5] In Chapter 4, we describe other ways to calculate and interpret a permutation p-value.

Let r_{i1} and r_{i2} be the ratings student i would give TA 1 when TA 1 is identified as male and as female, respectively; and let r_{i3} and r_{i4} the ratings student i would give TA 2 when that TA is identified as male and as female, respectively. Typically, the null hypotheses we test assert that for each i, some subset of $\{r_{ij}\}$ are equal. For assessing whether the identified gender of the TA affects SET, the null hypothesis is that for each i, $r_{i1} = r_{i2}$ (the rating the ith student would give TA 1 is the same, whether TA 1 is identified as male or female), and $r_{i3} = r_{i4}$ (the rating the ith student would give TA 2 is the same, whether TA 2 is identified as male or female). Different students might give different ratings under the same treatment condition (the null does not assert that $r_{ij} = r_{\ell j}$ for $i \neq \ell$), and the ith student might give different ratings to TA 1 and TA 2 (the null does not assert that $r_{i1} = r_{i3}$). The null hypothesis makes no assertion about the population distributions of $\{r_{i1}\}$ and $\{r_{i3}\}$, nor does it assert that $\{r_{ij}\}$ are a sample from some super-population.

For student i, we observe exactly one of $\{r_{i1}, r_{i2}, r_{i3}, r_{i4}\}$. If we observe r_{i1}, then—if the null hypothesis is true—we also know what r_{i2} is, and vice versa, but we do not know anything about r_{i3} or r_{i4}. Similarly, if we observe either r_{i3} or r_{i4} and the null hypothesis is true, we know the value of both, but we do not know anything about r_{i1} or r_{i2}.

Consider the average SET (for any particular item) given by the N_2+N_4 students assigned to sections taught by an apparently female TA, minus the average SET given by the N_1+N_3 students assigned to sections taught by an apparently male TA. This is what [66] tabulate as their key result. If the perceived gender of the TA made no difference in how students rated the TA, we would expect the difference of averages to be close to zero.[6] How "surprising" is the observed difference in averages?

Consider the

$$\binom{N_1 + N_2}{N_1} \times \binom{N_3 + N_4}{N_3} \tag{2.4}$$

assignments that keep the same N_1+N_2 students in TA 1's sections (but might change which of those sections a student is in) and the same N_3+N_4 students in TA 2's sections. For each of those assignments, we know what $\{R_i\}_{i=1}^N$ would have been if the null hypothesis is true: each would be exactly the same as its observed value, since those assignments keep students in sections taught by the same TA. Hence, we can calculate the value that the test statistic would have had for each of those assignments.

Because all $\binom{N}{N_1 N_2 N_3 N_4}$ possible assignments of students to sections are equally likely, these $\binom{N_1+N_2}{N_1} \times \binom{N_3+N_4}{N_3}$ assignments in particular are also equally likely. The fraction of those assignments for which the value of the test statistic is at least as large (in absolute value) as the observed value of the test statistic is the p-value of the null hypothesis that students give the same rating (or none) to an TA, regardless of the gender that TA appears to have.

[6] We would expect it to be a least a little different from zero both because of the luck of the draw in assigning students to sections and because students might rate the two TAs differently, regardless of the TA's perceived gender, and the groups are not all the same size.

This test is conditional on which of the students are assigned to each of the two TAs, but if we test at level no greater than α conditionally on the assignment, the unconditional level of the resulting test across all assignments is no greater than α, as shown above.

In principle, one could enumerate all the equally likely assignments and compute the value of the test statistic for each, to determine the (conditional) null distribution of the test statistic. In practice, there are prohibitively many assignments (for instance, there are $\binom{23}{11}\binom{24}{11} > 3.3 \times 10^{12}$ possible assignments of 47 students to the 4 TA-led sections that keep constant which students are assigned to each TA). Hence, we estimate p-values by simulation, drawing 10^5 equally likely assignments at random, with one exception, noted below. As discussed earlier, the standard error of the estimated p-values is hence no larger than $(1/2)/\sqrt{10^5} \approx 0.0016$. Code for all our analyses is at `https://github.com/kellieotto/SET-and-Gender-Bias`. Results for the US data are in Section 2.5.

2.4 The French natural experiment

In this section, we test hypotheses about relationships among SET, teaching effectiveness, grade expectations, and student and instructor gender. Our tests aggregate data within course sections, to match how SET are typically used in personnel decisions. We use the average of Pearson correlations across strata as the test statistic,[7] which allows us to test both for differences in means (which can be written as correlations with a dummy variable) and for association with ordinal or quantitative variables.

In these analyses, individual i is a section of a course; the "treatment" is the instructor's gender, the average interim grade, or the average final exam score; and the "response" is the average SET or the average final exam score. Strata consist of all sections of a single course in a single year.

Our tests for overall effects stratify on the course subject, to account for systematic differences across departments: the hypothetical randomization shuffles characteristics among courses in a given department, but not across departments. We also perform tests separately in different departments, and in some cases separately by student gender.

SET and final exam scores

We test whether average SET scores and average final exam scores for course sections are associated. The null hypothesis is that the pairing of average final grade and average SET for sections of a course each year is as if at random, independent across courses and across years. We test this hypothesis overall and separately by discipline, using the average Pearson correlation across strata, as described in Section 2.3. If the null hypothesis were true, we would expect the test statistic to be close to zero. On the other hand, if SET do measure

[7] As discussed above, we find p-values from the (nonparametric) permutation distribution, not from the theoretical distribution of the Pearson correlation under the parametric assumption of bivariate normality.

teaching effectiveness, we would expect average SET and average final exam score to be positively correlated within courses within years, making the test statistic positive.

The numbers show that SET scores do not measure teaching effectiveness well, overall: the one-sided p-value for the hypothesis that the correlation is zero is 0.09 (Table 2.2). Separate tests by discipline find that for History, the association is positive and statistically significant (p-value of 0.01), while the other disciplines (Macroeconomics, Microeconomics, Political Science and Sociology), the association is either negative or positive but not statistically significant (p-values 0.19, 0.55, 0.62, and 0.61 respectively). Political Institutions is not reported, because the final exam was not graded anonymously. The five strata of Political Institutions are not included in the overall average, which is computed from the remaining 21 strata-level correlation coefficients.

Table 2.2: Average correlation between SET and final exam score, by subject in the French university.

	strata	$\bar{\rho}$	p-value
Overall	26 (21)	0.04	0.09
History	5	0.16	0.01
Political Institutions	5	N/A	N/A
Macroeconomics	5	0.06	0.19
Microeconomics	5	-0.01	0.55
Political Science	3	-0.03	0.62
Sociology	3	-0.02	0.61

Note: p-values are one-sided.

SET and instructor gender

The second null hypothesis we test is that the pairing (by section) of instructor gender and SET is as if at random within courses each year, independently across years and courses. If gender does not affect SET, we would expect the correlation between average SET and instructor gender to be small in each course in each year. On the other hand, if students tend to rate instructors of one gender higher, we would expect the average correlation to be large in absolute value. We find that average SET are significantly associated with instructor gender, with male instructors getting higher ratings (overall p-value 0.00). Male instructors get higher SET on average in every discipline with two-sided p-values ranging from 0.08 for History to 0.63 for Political Science (Table 2.3).

Instructor gender and learning outcomes

Do men receive higher SET scores overall because they are better instructors? The third null hypothesis we test is that the pairing (by course) of instructor gender and average final exam score is as if at random within courses each year, independent across courses and

Table 2.3: Average correlation between SET and instructor gender in the French university.

	$\bar{\rho}$	p-value
Overall	0.09	0.00
History	0.11	0.08
Political Institutions	0.11	0.10
Macroeconomics	0.10	0.16
Microeconomics	0.09	0.16
Political Science	0.04	0.63
Sociology	0.08	0.34

Note: p-values are two-sided.

across years. If this hypothesis is true, we would expect the average correlations to be small. If the effectiveness of instructors differs systematically by gender, we would expect average correlation to be large in absolute value.

On the whole, students of male instructors perform *worse* on the final than students of female instructors, by an amount that is statistically significant (p-value 0.07 overall, Table 2.4). In all disciplines, students of male instructors perform worse, but by amounts that are not statistically significant (p-values ranging from 0.22 for History to 0.70 for Political Science). This suggests that male instructors are not noticeably more effective than female instructors, and perhaps are less effective. The statistically significant difference in SET scores for male and female instructors does not seem to reflect a difference in their teaching effectiveness.

Table 2.4: Average correlation between final exam scores and instructor gender in the French university.

	$\bar{\rho}$	p-value
Overall	-0.06	0.07
History	-0.08	0.22
Macroeconomics	-0.06	0.37
Microeconomics	-0.06	0.37
Political Science	-0.03	0.70
Sociology	-0.05	0.55

Note: p-values are two-sided. Negative values of $\bar{\rho}$ indicate that students of female instructors did better on average than students of male instructors.

Gender interactions

Why do male instructors receive higher SET scores? Separate analyses by student gender shows that male students tend to give higher SET scores to male instructors than female instructors (Table 2.5). These permutation tests confirm the results found by [13]. Gender concordance is a good predictor of SET scores for men (p-value 0.00 overall). Male students

give significantly higher SET scores to male instructors in History (p-value 0.01), Microeconomics (p-value 0.01), Macroeconomics (p-value 0.04), Political Science (p-value 0.06), and Political Institutions (p-value 0.08). Male students give higher SET scores to male instructors in Sociology as well, but the effect is not statistically significant (p-value 0.16).

The correlation between gender concordance and overall satisfaction scores for female students is also positive overall and weakly significant (p-value 0.09). The correlation is negative in some fields (History, Political Institutions, Macroeconomics, Microeconomics and Sociology) and positive in only one field (Political Science), but in no case statistically significant (p-values range from 0.12 to 0.97).

Table 2.5: Average correlation between SET and gender concordance in the French university.

	Male student		Female student	
	$\bar{\rho}$	p-value	$\bar{\rho}$	p-value
Overall	0.15	0.00	0.05	0.09
History	0.17	0.01	-0.03	0.60
Political Institutions	0.12	0.08	-0.11	0.12
Macroeconomics	0.14	0.04	-0.05	0.49
Microeconomics	0.18	0.01	-0.00	0.97
Political Science	0.17	0.06	0.04	0.64
Sociology	0.12	0.16	-0.03	0.76

Note: p-values are two-sided. Positive values indicate that students tended give higher ratings to instructors of the same gender.

Do male instructors receive higher SET scores from male students because their teaching styles match male students' learning styles? If so, we would expect male students of male instructors to perform better on the final exam. However, they do not (Table 2.6). If anything, male students of male instructors perform worse overall on the final exam (the correlation is negative but not statistically significant, with a p-value 0.75). In History, the amount by which male students of male instructors do worse on the final is significant (p-value 0.03): male History students give significantly higher SET scores to male instructors, despite the fact that they seem to learn more from female instructors. SET do not appear to measure teaching effectiveness, at least not primarily.

SET and grade expectations

The next null hypothesis we test is that the pairing by course of average SET scores with average interim grades is as if at random. We use interim grades as a proxy for students' grade expectations. If students give higher SET in courses where they expect higher grades, the association between SET scores and interim grades should be positive. Indeed, the association is positive and generally highly statistically significant (Table 2.7). Political Institutions is the only discipline for which the average correlation between interim grades and SET scores is negative, but the correlation is not significant (one-sided p-value 0.61).

Table 2.6: Average correlation between student performance and gender concordance in the French university.

	Male student		Female student	
	$\bar{\rho}$	p-value	$\bar{\rho}$	p-value
Overall	-0.01	0.75	0.06	0.07
History	-0.15	0.03	-0.02	0.74
Macroeconomics	0.04	0.60	0.11	0.10
Microeconomics	0.02	0.80	0.07	0.29
Political Science	0.08	0.37	0.11	0.23
Sociology	0.01	0.94	0.06	0.47

Note: p-values are two-sided. Positive values indicate that students tended receive higher final exam scores when their instructor was the same gender.

The estimated one-sided p-values for all other courses are between 0.0 and 0.03. The average correlations are especially high in History (0.32) and Sociology (0.24).

Table 2.7: Average correlation between SET and interim grades in the French university.

	$\bar{\rho}$	p-value
Overall	0.16	0.00
History	0.32	0.00
Political Institutions	-0.02	0.61
Macroeconomics	0.15	0.01
Microeconomics	0.13	0.03
Political Science	0.17	0.02
Sociology	0.24	0.00

Note: p-values are one-sided.

In summary, the average correlation between SET and final exam grades (at the level of class sections) is positive, but only weakly significant overall and not significant for most disciplines. However, the average correlation between SET and grade expectations (at the level of class sections) is positive and significant overall and across most disciplines. The average correlation between instructor gender and SET is statistically significant—male instructors get higher SET—but if anything, students of male instructors do worse on final exams than students of female instructors. Male students tend to give male instructors higher SET, even though they might be learning less than they do from female instructors. We conclude that SET are influenced more by instructor gender and student grade expectations than by teaching effectiveness.

2.5 The US randomized experiment

The previous section suggests that SET have little connection to teaching effectiveness, but the natural experiment does not allow us to control for differences in teaching styles

across instructors. The truly randomized experiment from [66] does. As discussed above, [66] collected SET from an online course in which 43 students were randomly assigned to four[8] discussion groups, each taught by one of two TAs, one male and one female. The TAs coordinated their teaching practices: they gave similar feedback to students, returned assignments at exactly the same time, etc.

Biases in student ratings are revealed by differences in ratings each TA received when that TA is identified to the students as male versus as female. [66] find that "the male identity received significantly higher scores on professionalism, promptness, fairness, respectfulness, enthusiasm, giving praise, and the student ratings index ... Students in the two groups that perceived their assistant instructor to be male rated their instructor significantly higher than did the students in the two groups that perceived their assistant instructor to be female, regardless of the actual gender of the assistant instructor." [66] used parametric tests whose assumptions did not match their experimental design; part of our contribution is to show that their data admit a more rigorous analysis using permutation tests that honor the underlying randomization and that avoid parametric assumptions about SET. Our permutation analysis supports their overall conclusions, in some cases substantially more strongly than the original analysis (for instance, p-values of 0.01 versus 0.19 for promptness and fairness). In other cases, the original parametric tests overstated the evidence (for instance, a p-value of 0.29 versus 0.04 for knowledgeability).

We use permutation tests as described above in Section 2.3. Individual i is a student; the treatment is the combination of the TA's identity and the TA's reported gender (there are $K = 4$ treatments). The null hypothesis is that each student would give a TA the same SET score, whether that TA is apparently male or apparently female. A student might give the two TAs different scores, and different students might give different scores to the same TA.

Because of how the experimental randomization was performed, all allocations of students to TA sections that preserve the number of students in each section are equally likely, including allocations that keep the same students assigned to each actual TA constant.

To test whether there is a systematic difference in how students rate male-identified and female-identified TAs, we use the difference in pooled means as our test statistic: We pool the SET for both instructors when they are identified as female and take the mean, pool the SET for both instructors when they are identified as male and take the mean, then subtract the second mean from the first mean (Table 2.8). This is what [66] report as their main result.

As described above, the randomization is stratified and conditions on the set of students allocated to each TA, because, under the null hypothesis, we then know what SET students would have given for each possible allocation, completely specifying the null distribution of the test statistic. The randomization includes the nonresponders, who are omitted from the averages of the group they are assigned to.

[8] As discussed above, there were six sections in all, of which two were taught by the professor and four were taught by TAs.

We also perform tests involving the association of concordance of student and apparent TA gender (Table 2.9), and SET and concordance of student and actual TA gender (Table 2.10) using the pooled difference in means as the test statistic. We test the association between grades and actual TA gender using the average Pearson correlation across strata as the test statistic (Table 2.11). We find the p-values from the stratified permutation distribution of the test statistic, avoiding parametric assumptions.

SET and perceived instructor gender

The first hypothesis we test is that students would rate a given TA the same, whether the student thinks the TA is female or male. A positive value of the test statistic means that students give higher SET on average to apparently male instructors. There is weak evidence that the overall SET score depends on the perceived gender (p-value 0.12). Table 2.8 shows that the evidence is stronger for several other items students rated: fairness, promptness, giving praise, enthusiasm, communication, professionalism, respect, and caring. For seven items, the nonparametric permutation p-values are smaller than the parametric p-values reported by [66]. Items for which the permutation p-values were greater than 0.10 include clarity, consistency, feedback, helpfulness, responsiveness, and knowledgeability. SET were on a 5-point scale, so a difference in means of 0.80, observed in student ratings of the promptness with which assignments were returned, is 20% of the full scale—an enormous difference. Since assignments were returned at exactly the same time in all four sections of the class, this seriously impugns the ability of SET to measure even putatively objective characteristics of teaching.

We also conducted separate tests by student gender. In contrast to our findings for the French data, where male students rated male instructors higher, in this experiment female students rated the perceived male instructors higher (Table 2.9). Male students rated the male-identified instructor significantly (though weakly) higher on only one criterion: fairness (p-value 0.09). Female students, however, rated the male-identified instructor higher on overall satisfaction (p-value 0.11) and most teaching dimensions: praise, enthusiasm, caring, fairness, respectfulness, communication, professionalism, and feedback. Female students rated the female-identified instructors lower on helpfulness, promptness, consistency, responsiveness, knowledge, and clarity, although the differences are not statistically significant.

Table 2.9 shows that students of both genders rated the male-identified instructor higher on all dimensions. However, Table 2.10 shows that the *actual* instructor gender is weakly associated with ratings. Students rated the actual male instructor higher on some dimensions and lower on others, by amounts that generally were not statistically significant. The exceptions were praise (p-value 0.02) and responsiveness (p-value 0.05), where female students tended to rate the actual female instructor significantly higher.

Students of the actual male instructor performed worse in the course on average, by an amount that was statistically significant (Table 2.11). The difference in student performance by reported gender of the instructor is not statistically significant.

Table 2.8: Difference in mean ratings by reported instructor gender (male minus female) in the US university.

	difference in means	permutation p-value	MacNell et al. p-value
Overall	0.47	0.12	0.128
Professional	0.61	0.07	0.124
Respectful	0.61	0.06	0.124
Caring	0.52	0.10	0.071
Enthusiastic	0.57	0.06	0.112
Communicate	0.57	0.07	NA
Helpful	0.46	0.17	0.049
Feedback	0.47	0.16	0.054
Prompt	0.80	0.01	0.191
Consistent	0.46	0.21	0.045
Fair	0.76	0.01	0.188
Responsive	0.22	0.48	0.013
Praise	0.67	0.01	0.153
Knowledge	0.35	0.29	0.038
Clear	0.41	0.29	NA

Note: p-values are two-sided.

Table 2.9: Association between SET and reported instructor gender (male minus female) in the US university.

	Male students		Female students	
	difference in means	p-value	difference in means	p-value
Overall	0.17	0.82	0.79	0.11
Professional	0.42	0.55	0.82	0.12
Respectful	0.42	0.55	0.82	0.12
Caring	0.04	1.00	0.96	0.05
Enthusiastic	0.17	0.83	0.96	0.05
Communicate	0.25	0.68	0.87	0.10
Helpful	0.46	0.43	0.51	0.35
Feedback	0.08	1.00	0.88	0.10
Prompt	0.71	0.15	0.86	0.13
Consistent	0.17	0.85	0.77	0.17
Fair	0.75	0.09	0.88	0.04
Responsive	0.38	0.54	0.06	1.00
Praise	0.58	0.29	0.81	0.01
Knowledge	0.17	0.84	0.54	0.21
Clear	0.13	0.85	0.67	0.29

Note: p-values are two-sided.

Table 2.10: Association between SET and actual instructor gender (male minus female) in the US university.

	Male students		Female students	
	difference in means	p-value	difference in means	p-value
Overall	-0.13	0.61	-0.29	0.48
Professional	0.15	0.96	-0.09	0.73
Respectful	0.15	0.96	-0.09	0.73
Caring	-0.22	0.52	-0.07	0.75
Enthusiastic	-0.13	0.62	-0.44	0.29
Communicate	-0.02	0.80	-0.18	0.61
Helpful	0.03	0.89	0.26	0.71
Feedback	-0.24	0.48	-0.41	0.36
Prompt	-0.09	0.69	-0.33	0.44
Consistent	0.12	0.97	-0.40	0.35
Fair	-0.06	0.71	-0.59	0.12
Responsive	-0.13	0.64	-0.68	0.05
Praise	0.02	0.86	-0.60	0.02
Knowledge	0.22	0.83	-0.44	0.17
Clear	-0.26	0.49	-0.98	0.07

Note: p-values are two-sided.

Table 2.11: Difference in mean grades by instructor gender (male minus female) in the US university.

	difference in means	p-value
Reported	1.76	0.54
Actual	-6.81	0.02

Note: p-values are two-sided.

These results suggest that students rate instructors more on the basis of the instructor's perceived gender than on the basis of the instructor's effectiveness. Students of the TA who is actually female did substantially better in the course, but students rated male-identified TAs higher.

2.6 Multiplicity

We did not adjust the p-values reported above for multiplicity. We performed a total of approximately 50 tests on the French data, of which we consider four to be our primary results:

1FR lack of association between SET and final exam scores (a negative result, so multiplicity is not an issue)

2FR lack of association between instructor gender and final exam scores (a negative result, so multiplicity is not an issue)

3FR association between SET and instructor gender

4FR association between SET and interim grades

Bonferroni's adjustment for these four tests would leave the last two associations highly significant, with adjusted p-values less than 0.01.

We performed a total of 77 tests on the US data. We consider the three primary null hypotheses to be

1US perceived instructor gender plays no role in SET

2US male students rate male-identified and female-identified instructors the same

3US female students rate male-identified and female-identified instructors the same

To account for multiplicity, we tested these three "omnibus" hypotheses using the nonparametric combination of tests (NPC) method with Fisher's combining function [98, Chapter 4] to summarize the 15 dimensions of teaching into a single test statistic that measures how "surprising" the 15 observed differences would be for each of the three null hypotheses. In 10^5 replications, the empirical p-values for these three omnibus hypotheses were 0 (99% confidence interval $[0.0, 5.3 \times 10^{-5}]$), 0.464 (99% confidence interval $[0.460, 0.468]$), and 0 (99% confidence interval $[0.0, 5.3 \times 10^{-5}]$), respectively. (The confidence bounds were obtained by inverting Binomial hypothesis tests.) Thus, we reject hypotheses 1US and 3US.

We made no attempt to optimize the tests to have power against the alternatives considered. For instance, with the US data, the test statistic grouped the two identified-as-female sections and the two identified-as-male conditions, in keeping with how [66] tabulated their results, rather than using each TA as his or her own control (although the randomization keeps the two strata intact). Given the relatively small number of students in the US experiment, it is remarkable that *any* of the p-values is small, much less that the p-values for the omnibus tests are effectively zero.

2.7 Code and data

Jupyter (`http://jupyter.org/`) notebooks containing our analyses are at `https://github.com/kellieotto/SET-and-Gender-Bias`; they rely on the `permute` Python library (`https://pypi.python.org/pypi/permute/`). The US data are available at `http://n2t.net/ark:/b6078/d1mw2k`. French privacy law prohibits publishing the French data.

2.8 Discussion

Other studies

To our knowledge, only two experiments have controlled for teaching style in their designs: [3] and [66]. In both experiments, students generally gave higher SET when they *thought* the instructor was male, regardless of the actual gender of the instructor. Both experiments found that systematic differences in SET by instructor gender reflect gender bias rather than a match of teaching style and student learning style or a difference in actual teaching effectiveness. Numerous observational studies have also concluded that SET are biased by gender [144, 74, 107].

Instructor race is also associated with SET. In the US, SET of instructors of color appear to be biased downwards: minority instructors tend to receive significantly lower SET scores compared to white (male) instructors [75].[9] Observational studies show that SET tend to be lower for non-native English speaking instructors [136, 144]. Age, [3], charisma [118], and physical attractiveness [106, 41, 153] are also associated with SET. Other factors generally not in the instructor's control that may affect SET scores include class time, class size, mathematical or technical content, the physical classroom environment [47], and even the availability of cookies in the classroom [45].

Many studies cast doubt on the validity of SET as a measure of teaching effectiveness (see [52, Chapters 3–5] for a review and analysis, [101] for a review, and [35, 18] for exemplars). Moreover, studies have called into question whether the data from SET are trustworthy and interpretable. [123] required students to take quizzes on the course syllabus and to sign documents stating that they received grades at the first possible opportunity. Out of 40 SET, only one student reported that they understood the course objectives and how they would be evaluated, and only five students reported that they received graded documents reasonably promptly. [56] showed that instructors tend to misjudge students' definitions of terms such as "not fair," "professional," "not organized," "challenging," and "not respectful."

Some studies find that gender and SET are not significantly associated [7, 21, 28] and that SET are valid and reliable measures of teaching effectiveness [8, 20].[10] The contradictions among conclusions suggests that if SET are ever valid, they are not valid in general: universities should not assume that SET are broadly valid at their institution, valid in any particular department, or valid for any particular course. Given the many sources of bias in SET and the variability in magnitude of the bias by topic, item, student gender, etc., as a practical matter it is impossible to adjust for biases to make SET a valid, useful measure of teaching effectiveness.

[9] French law does not allow the use of race-related variables in data sets. We were thus unable to test for racial biases in SET using the French data.

[10] Some authors who claim that SET are valid have a financial interest in developing SET instruments and conducting SET.

Summary

We used permutation tests to examine data collected by [13] and [66], both of which find that gender biases prevent SET from measuring teaching effectiveness accurately and fairly. SET are more strongly related to instructor's perceived gender and to students' grade expectations than they are to learning, as measured by performance on anonymously graded, uniform final exams. The extent and direction of gender biases depend on context, so it is impossible to adjust for such biases to level the playing field. While the French university data show a positive male student bias for male instructors, the experimental US setting suggests a positive female student bias for male instructors. The biases in the French university data vary by course topic; the biases in the US data vary by item. We would also expect the bias to depend on class size, format, level, physical characteristics of the classroom, instructor ethnicity, and a host of other variables.

We do not claim that there is *no* connection between SET and student performance. However, the observed association is sometimes positive and sometimes negative, and in general is not statistically significant—in contrast to the statistically significant strong associations between SET and grade expectations and between SET and instructor gender. SET appear to measure student satisfaction and grade expectations more than they measure teaching effectiveness [132, 52]. While student satisfaction may *contribute* to teaching effectiveness, it is not itself teaching effectiveness. Students may be satisfied or dissatisfied with courses for reasons unrelated to learning outcomes—and not in the instructor's control (e.g., the instructor's gender).

In the US, SET have two primary uses: instructional improvement and personnel decisions, including hiring, firing, and promoting instructors. We recommend caution in the first use, and discontinuing the second use, given the strong student biases that influence SET, even on "objective" items such as how promptly instructors return assignments.[11]

Conclusion

In two very different universities and in a broad range of course topics, SET measure students' gender biases better than they measure the instructor's teaching effectiveness. Overall, SET disadvantage female instructors. There is no evidence that this is the exception rather than the rule. Hence, the onus should be on universities that rely on SET for employment decisions to provide convincing affirmative evidence that such reliance does not have disparate impact on women, under-represented minorities, or other protected groups. Because the bias varies by course and institution, affirmative evidence needs to be specific to a given course in a given department in a given university. Absent such specific evidence, SET should not be used for personnel decisions.

[11] In 2009, the French Ministry of Higher Education and Research upheld a 1997 decision of the French State Council that public universities can use SET only to help tenured instructors improve their pedagogy, and that the administration may not use SET in decisions that might affect tenured instructors' careers (c.f. [12]).

Chapter 3

Pseudo-random number generators and permutations

3.1 Introduction

Pseudo-random number generators (PRNGs) are central to the practice of Statistics. They are used to draw random samples, allocate patients to treatments, perform the bootstrap, calibrate permutation tests, perform MCMC, approximate p-values, partition data into training and test sets, and countless other purposes.

Practitioners generally do not question whether standard software is adequate for these tasks. This paper explores whether PRNGs generally considered adequate for statistical work really are adequate, and whether standard software uses appropriate algorithms for generating random integers, random samples, and independent and identically distributed (IID) random variates.

Textbooks give methods that implicitly or explicitly assume that PRNGs can be substituted for true IID $U[0,1)$ variables without introducing material error [135, 57, 27, 102, 96]. We show here that this assumption is incorrect for algorithms in many commonly used statistical packages, including MATLAB, Python's `random` module, R, SPSS, and Stata.

For example, whether software can in principle generate all samples of size k from a population of n items—much less generate them with equal probability—depends on the size of the problem and the internals of the software, including the underlying PRNG and the *sampling algorithm*, the method used to map PRNG output into a sample. We show that even for datasets with only hundreds of observations, many common PRNGs cannot draw all subsets of size k, for modest values of k.

Some sampling algorithms put greater demands on the PRNG than others. For instance, some involve permuting the data. The number of items that common PRNGs can permute ranges from at most 13 to at most 2084, far smaller than many data sets. Other sampling algorithms require uniformly distributed integers (as opposed to the approximately $U[0,1)$ PRNG outputs) as input. Many software packages generate pseudo-random integers using a

rounding method that does not yield uniformly distributed integers, even if PRNG output were uniformly distributed on w-bit binary integers.

As a result of the limitations of common PRNGs and sampling algorithms, the L_1 distance between the uniform distribution on samples of size k and the distribution induced by a particular PRNG and sampling algorithm can be nearly 2. It follows that there exist bounded functions of random samples whose expectations with respect to those two distributions differ substantially.

Section 3.2 presents an overview of PRNGs and gives examples of better and worse ones. Section 3.3 shows that, for modest n and k, the state spaces of common PRNGs considered adequate for Statistics are too small to generate all permutations of n things or all samples of k of n things. Section 3.4 discusses sampling algorithms, provides that some are less demanding on the PRNG than others, and shows that a common, "textbook" procedure for generating pseudo-random integers using a PRNG can be quite inaccurate. Unfortunately, this is essentially the method that R uses and that the Python `random.choice()` function uses. Section 3.5 describes tests for pseudo-randomness, including existing test batteries and new statistical tests we have applied. Section 3.6 concludes with recommendations and best practices.

3.2 Pseudo-random number generators

A *pseudo-random number generator* (PRNG) is a deterministic algorithm that, starting with an initial "seed" value, produces a sequence of numbers that are supposed to behave like random numbers. An ideal PRNG has output that is statistically indistinguishable from random, uniform, IID bits. Cryptographically secure PRNGs approach this ideal—the bits are (or seem to be) computationally indistinguishable from IID uniform bits—but common PRNGs do not.

A PRNG has several components: an internal *state*, initialized with a *seed*; a function that maps the current state to an output; and a function that updates the internal state.

If the state space is finite, the PRNG must eventually revisit a state after some number of calls—after which, it repeats. The *period* of a PRNG is the maximum, over initial states, of the number of states the PRNG visits before returning to a state already visited. The period is at most the total number of possible states. If the period is equal to the total number of states, the PRNG is said to have *full period*. PRNGs for which the state and the output are the same have periods no larger than the number of possible outputs. Better PRNGs generally use a state space with dimension much larger than the dimension of the output.

Some PRNGs are sensitive to the initial state. For unfavorable initial states, the PRNG may need many "burn-in" calls before the output behaves well.

Simple PRNGs

Linear congruential generators (LCGs) have the form $X_{n+1} = (aX_n + c) \mod m$, for a *modulus* m, *multiplier* a, and *additive constant* c. LCGs are fast to compute and require little computer memory. The behavior of LCGs is well understood from number theory. For instance, the Hull-Dobell theorem [49] gives necessary and sufficient conditions for a LCG to have full period for all seeds, and there are upper bounds on the number of hyperplanes of dimension k that contain all k-tuples of outputs, as a function of m [68]. When all k-tuples are in a smaller number of hyperplanes, that indicates that the PRNG outputs are more regular and more predictable.

To take advantage of hardware efficiencies, early computer systems implemented LCGs with moduli of the form $m = 2^b$, where b was the integer word size of the computer. This led to wide propagation of a particularly bad PRNG, RANDU, originally introduced on IBM mainframes [55, 67]. (RANDU has $a = 65539$, $m = 2^{31}$, and $c = 0$.)

More generally, LCGs with $m = 2^b$ cannot have full period because m is not prime. Better LCGs have been developed—and some are used in commercial statistical software packages—but they are still generally considered inadequate for Statistics because of their short periods (typically $\leq 2^{32}$) and correlation among outputs.

The Wichmann-Hill PRNG is a sum of three normalized LCGs; its output is in $[0, 1)$. It is generally not considered adequate for Statistics, but was (nominally) the PRNG in Excel 2003, 2007, and 2010.[1] The generator in Excel had an implementation bug that persisted for several generations. Excel didn't allow the seed to be set so issues could not be replicated, but users reported that the PRNG occasionally gave a negative output [72]. As of 2014, IMF banking Stress tests used Excel simulations [90]. This worries us.

Many other approaches to generating pseudo-random numbers have been proposed, and PRNGs can be built by combining simpler ones (carefully–see [55] on "randomly" combining PRNGs). For instance, the KISS generator combines four generators of three types, and has a period greater than 2^{210}. Nonetheless, standard PRNGs are predictable from a relatively small number of outputs. For example, one can determine the LCG constants a, c, and m by observing only 3 outputs, and can recover the state of KISS from about 70 words of output [112].

Mersenne Twister (MT)

Mersenne Twister (MT) [71] is a "twisted generalized feedback shift register," a sequence of bitwise and linear operations. Its state space is $19,937$ bits and it has an enormous period $2^{19937} - 1$, a Mersenne prime. It is k-equidistributed to 32-bit accuracy for $k \leq 623$, meaning that output vectors of length up to 623 (except the zero vector) occur with equal frequency over the full period. The state is a 624×32 binary matrix.

[1] https://support.microsoft.com/en-us/help/828795/description-of-the-rand-function-in-excel, last visited 23 October 2018.

MT is the default PRNG in common languages and software packages, including Python, R, Stata, GNU Octave, Maple, MATLAB, Mathematica, and many more (see Table 3.2). We show below that it is not adequate for statistical analysis of modern data sets. Moreover, MT can have slow "burn in," especially for seeds with many zeros [115]. And the outputs for close seeds can be similar, which makes seeding distributed computations delicate.

Cryptographic hash functions

The PRNGs described above are quick to compute but predictable, and their outputs are easy to distinguish from actual random bits [58]. Cryptographers have devoted a great deal of energy to inventing cryptographic hash functions, which can be used to create PRNGs, as the properties that make functions cryptographically secure are properties of good pseudo-randomness.

A *cryptographic hash function* H is a function with the following properties:

- H produces a fixed-length "digest" (hash) from arbitrarily long "message" (input): $H : \{0,1\}^* \to \{0,1\}^L$.
- H is inexpensive to compute.
- H is "one-way," i.e., it is hard to find a pre-image of any output except by exhaustive enumeration (this is the basis of hashcash "proof of work" for Bitcoin and some other distributed ledgers).
- H is collision-resistant, i.e., it is hard to find $M_1 \neq M_2$ such that $H(M_1) = H(M_2)$.
- small changes to M produce unpredictable, big changes to $H(M)$.
- outputs of H are equidistributed: bits of the hash are essentially IID random.

These properties of H make it suitable as the basis of a PRNG: It is *as if* $H(M)$ is a uniformly distributed random L-bit string assigned to M. One can construct a simple hash-based PRNG with the procedure outlined in Figure 3.1 which we first learned about from Ronald L. Rivest [110].

Since a message can be arbitrarily long, this PRNG has an unbounded state space. For truly cryptographic applications, the seed should be reset to a new random value periodically; for statistical applications, that should not be necessary.

3.3 Counting permutations and samples

Theorem 1 (Pigeonhole principle)**.** *If you put $N > n$ pigeons in n pigeonholes, at least one pigeonhole must contain more than one pigeon.*

Hash functions in counter mode

1. Generate a random string S with a substantial amount of entropy, e.g., 20 rolls of a 10-sided die.
2. Set $i = 0$. i is the number of values generated so far. The state of the PRNG is the string "S," with the zero byte string appended $i + 1$ times.
3. Set X_i be the hash of the state, interpreted as a (long) hexadecimal number.
4. Increment i and return to step 4 to generate more outputs.

Figure 3.1: Algorithm for using a hash function in counter mode to generate PRNs.

Corollary 1. *At most n pigeons can be put in n pigeonholes if at most one pigeon is put in each hole.*

The corollary implies that a PRNG cannot generate more permutations or samples than the number of states the PRNG has (which is in turn an upper bound on the period of the PRNG). Of course, that does not mean that the permutations or samples a PRNG can generate occur with approximately equal probability: that depends on the quality of the PRNG, not just the size of the state space. Nonetheless, it follows that no PRNG with a finite state space can be "adequate for Statistics" for every statistical problem.

The number of permutations of n objects is $n!$; the number of possible samples of k of n items with replacement is n^k; and the number of possible samples of k of n without replacement is $\binom{n}{k}$. These bounds are helpful for counting permutation pigeons:

- Stirling bounds: $en^{n+1/2}e^{-n} \geq n! \geq \sqrt{2\pi}n^{n+1/2}e^{-n}$.
- Entropy bounds: $\frac{2^{nH(k/n)}}{n+1} \leq \binom{n}{k} \leq 2^{nH(k/n)}$, where $H(q) \equiv -q\log_2(q) - (1-q)\log_2(1-q)$.
- Stirling combination bounds: for $\ell \geq 1$ and $m \geq 2$, $\binom{\ell m}{\ell} \geq \frac{m^{m(\ell-1)+1}}{\sqrt{\ell}(m-1)^{(m-1)(\ell-1)}}$.

Table 3.1 compares numbers of permutations and random samples to the size of the state space of various PRNGs. PRNGs with 32-bit state spaces, which include some in statistical packages, cannot generate all permutations of even small populations, nor all random samples of small size from modest populations. MT is better, but still inadequate: it can generate fewer than 1% of the permutations of 2084 items.

L_1 bounds

Simple probability inequalities give attainable bounds on the bias introduced by using a PRNG with insufficiently large state space, on the assumption that the PRNG is uniform on

Table 3.1: The pigeonhole principle applied to PRNGs, samples, and permutations. For a PRNG of each size state space, the table gives examples where some samples or permutations must be unobtainable.

Feature	Size	Full	Scientific notation
32-bit state space	2^{32}	4,294,967,296	4.29×10^9
Permutations of 13	13!	6,227,020,800	6.23×10^9
Samples of 10 out of 50	$\binom{50}{10}$	10,272,278,170	1.03×10^{10}
Fraction of attainable samples with 32-bit state space	$2^{32}/\binom{50}{10}$	0.418	
64-bit state space	2^{64}	18,446,744,073,709,551,616	1.84×10^{19}
Permutations of 21	21!	51,090,942,171,709,440,000	5.11×10^{19}
Samples of 10 out of 500	$\binom{500}{10}$		2.46×10^{20}
Fraction of attainable samples with 64-bit state space	$2^{64}/\binom{500}{10}$	0.075	
128-bit state space	2^{128}		3.40×10^{38}
Permutations of 35	35!		1.03×10^{40}
Samples of 25 out of 500	$\binom{500}{25}$		2.67×10^{42}
Fraction of attainable samples with 128-bit state space	$2^{128}/\binom{500}{25}$	0.0003	
MT state space	$2^{32\times624}$		9.27×10^{6010}
Permutations of 2084	2084!		3.73×10^{6013}
Samples of 1000 out of 390 million	$\binom{3.9\times10^8}{1000}$		$> 10^{6016}$
Fraction of attainable samples	$2^{32\times624}/\binom{3.9\times10^8}{1000}$		$< 1.66 \times 10^{-6}$

its possible outputs. (Failure of that uniformity makes matters even worse.) Suppose $\mathbb{P}_0$ and $\mathbb{P}_1$ are probability distributions on a common measurable space. If there is some measurable set S for which $\mathbb{P}_0(S) = \epsilon$ and $\mathbb{P}_1(S) = 0$, then $\|\mathbb{P}_0 - \mathbb{P}_1\|_1 \geq 2\epsilon$. Thus there is a function f with $|f| \leq 1$ such that

$$\mathbb{E}_{\mathbb{P}_0} f - \mathbb{E}_{\mathbb{P}_1} f \geq 2\epsilon.$$

In the present context, $\mathbb{P}_0$ is the uniform distribution (on samples or permutations) and $\mathbb{P}_1$ is the distribution induced by the PRNG and sampling algorithm. If the PRNG has n states and we want to generate $N > n$ equally likely outcomes, at least $N - n$ outcomes will have probability zero instead of $1/N$. Some statistics will have bias of (at least) $2 \times \frac{N-n}{N}$. As seen in Table 3.1, the fraction of attainable samples or permutations is quite small in problems of a size commonly encountered in practice, making the bias nearly 2.

3.4 Sampling algorithms

There are many ways to use a source of pseudo-randomness to simulate drawing a simple random sample. A common approach is like shuffling a deck of n cards, then dealing the top k: assign a (pseudo-)random number to each item, sort the items based on that number to produce a permutation of the population, then take the first k elements of the permuted list to be the sample [135, 57, 27]. We call this algorithm PIKK: permute indices and keep k.

If the pseudo-random numbers really were IID $U[0, 1)$, every permutation would indeed be equally likely, and the first k would be a simple random sample. But if the permutations are not equiprobable, there is no reason to think that the first k elements comprise a random sample. Furthermore, this algorithm is inefficient: it requires generating n pseudo-random numbers and then an $O(n \log n)$ sorting operation.

There are better ways to generate a random permutation, such as the "Fisher-Yates shuffle" or "Knuth shuffle" (Knuth attributes it to Durstenfeld) [55], which involves generating n independent random integers on various ranges, but no sorting. There is also a version suitable for *streaming*, i.e., permuting a list that has an (initially) unknown number of elements. Generating n pseudo-random numbers places more demand on a PRNG than other sampling algorithms discussed below, which only require $k < n$ pseudo-random numbers.

One simple method to draw a random sample of size k from a population of size n is to draw k integers at random without replacement from $\{1, \ldots, n\}$, then take the items with those indices to be the sample. [24] provide an elegant recursive algorithm to draw random samples of size k out of n; it requires the software recursion limit to be at least k. (In Python, the default maximum recursion depth is 2000, so this algorithm cannot draw samples of size greater than 2000 unless one increases the recursion limit.)

The sampling algorithms mentioned so far require n to be known. *Reservoir* algorithms, such as Waterman's Algorithm R, do not [55]. Moreover, reservoir algorithms are suitable for streaming: items are examined sequentially and either enter into the reservoir, or, if not,

are never revisited. Vitter's Algorithm Z is even more efficient than Algorithm R, using random skips to reduce runtime to be essentially linear in k [142].

Pseudo-random integers

Many sampling algorithms require pseudo-random integers on $\{1, \ldots, m\}$. The output of a PRNG is typically a w-bit integer, so some method is needed to map it to the range $\{1, \ldots, m\}$.

A textbook way to generate an integer on the range $\{1, \ldots, m\}$ is to first draw a random $X \sim U[0, 1)$ and then define $Y \equiv 1 + \lfloor mX \rfloor$ [102, 96]. In practice, PRNG outputs are not $U[0, 1)$: they are derived by normalizing a value that is (supposed to be) uniformly distributed on w-bit integers.

Even if X is uniformly distributed on w-bit integers, the distribution of Y will not be uniform on $\{1, \ldots, m\}$ unless m is a power of 2. If $m > 2^w$, at least $m - 2^w$ values will have probability 0 instead of probability $1/m$. If $w = 32$, then for $m > 2^{32} \approx 4.24 \times 10^9$, some values will have probability 0. Conversely, there exists $m < 2^w$ such that the ratio of the largest to smallest selection probability of $\{1, \ldots, m\}$ is, to first order, $1 + m2^{-w+1}$ [55].

R (Version 3.5.1) [103] uses this multiply-and-floor approach to generate pseudo-random integers, which eventually are used in the main sampling functions. Duncan Murdoch devised a simple simulation that shows how large the inhomogeneity of selection probabilities can be: for $m = (2/5) \times 2^{32} = 1,717,986,918$, the `sample()` function generates about 40% even numbers and about 60% odd numbers [2]. We are pleased to report that after we called attention to this issue in [93], the R development team updated the functions `sample()`, `walkerProbSampleReplace()`, and `wilcox.test()` in R version 3.6 to use the following, better approach.[3]

A more accurate way to generate random integers on $\{1, \ldots, m\}$ is to use pseudo-random bits directly. This is not a new idea; [48] describe essentially the same procedure to draw integers by hand from random decimal digit tables. The integer $m - 1$ can be represented with $\mu = \lceil \log_2(m - 1) \rceil$ bits. To generate a pseudo-random integer uniformly distributed on $\{1, \ldots, m\}$, generate μ pseudo-random bits (for instance, by taking the most significant μ bits from the PRNG output) and interpret the bits as a binary integer. If the integer is larger than $m - 1$, then discard it and draw another μ bits until the μ bits represent an integer less than or equal to $m - 1$. When that occurs, return that integer, plus 1. This procedure potentially requires throwing out (in expectation) almost half the draws if $m-1$ is just below a power of 2, but the algorithm's output will be uniformly distributed (if the input bits are). This is how the Python package Numpy (Version 1.14) generates pseudo-random integers.[4]

[2] `https://stat.ethz.ch/pipermail/r-devel/2018-September/076827.html`, last visited 17 October 2018

[3] See `https://github.com/wch/r-source/commit/e4acfd1f240a42b3fc1474a4d97017114e4eb053#diff-3fa2a86745f6ebacfa969f26aea574c7`.

[4] However, Python's built-in `random.choice()` (Versions 2.7 through 3.6) does something else that's biased: it finds the closest integer to mX.

3.5 Tests of pseudo-randomness

While defining randomness borders on being a philosophical exercise, pseudo-randomness has a concrete definition from computational complexity theory. A distribution is said to be *pseudo-random* if for any polynomial time algorithm, the probability of correctly identifying the pseudo-random from the truly random distribution is arbitrarily small.

The notion of pseudo-randomness can weakened for PRNGs used in statistical applications: sequences from a PRNG should have the same relevant statistical properties as a truly random source. Values and subsequences from a good PRNG should be equiprobable, and the values within sequences should be independent and unpredictable.

Many test batteries for PRNGs have been developed over the years. The Diehard tests of [69] were the first standardized software package for PRNG testing. The software implemented thirteen tests, including the "birthday test," based on the spacings of random points on an interval; the "monkey tests," based on the number of overlapping bit words in a stream; and the "parking lot test," based on the number of non-overlapping unit circles randomly placed on a square. This software was the state of the art until [58] released the TestU01 suite. TestU01 is the most comprehensive software to date: it includes chi-squared and other statistical tests, geometric tests, computational complexity tests, and information theoretic tests, as well as options to test the PRNG behavior at different parts of its period. NIST maintains a list of benchmark tests specifically for cryptographically secure PRNGs [120, 114].

We propose several new statistical tests based on simple random sampling and permuting sequences. Existing test batteries test the raw outputs of PRNGs. The proposed approach takes sequences of raw outputs from a PRNG, passes them through a sampling or permutation algorithm, and conducts tests using the resulting output. These algorithms take (supposedly) uniform PRNS and scale each one in a different way depending on n, k, and how many steps the algorithm has taken. Tests based on these outputs exercise the PRNGs in a new way.

A particular sampling algorithm and choice of k and n divide sequences of pseudorandom numbers into equivalence classes based on which sample they yield. (Several different sequences of PRNs yield the same sample because samples are unordered collections.) The equivalence classes of sequences are not necessarily the same as the categories considered in other chi-squared tests based on multinomial category probabilities. Using different algorithms, such as those in Section 3.4, to draw samples may uncover different inadequacies of a PRNG.

Uniformity of SRSs using the multinomial distribution

We propose two approaches to test whether each of $N = \binom{n}{k}$ samples are selected with equal probability: the usual chi-squared test and a test based on the range of the multinomial values, $\max_i n_i - \min_i n_i$, where $n_1, \ldots, n_N$ are the frequencies of samples that have equal

probability $1/N$ under the null. [51] and [154] provide the following approximation to the distribution of the range:

$$P(\max_i n_i - \min_i n_i \leq r) \approx P(W_N \leq (r - (2B)^{-1})(N/B)^{1/2}),$$

where W_N denotes the sample range of N independent standard normal random variables and B is the number of multinomial draws. It is a known result (see e.g. [44, 113]) that the distribution function for the range of B IID normal samples is given by

$$R(w) = B \int_{-\infty}^{\infty} \phi(x) \left[\Phi(x + w) - \Phi(x)\right]^{B-1} dx,$$

where ϕ and Φ are the standard normal density and cumulative distribution function, respectively.

Tests of uniformity using Wald's SPRT

We use Wald's sequential probability ratio test for Bernoulli random samples (see Chapter 1.1) as the basis for two tests for PRNG uniformity. We have applied the SPRT to test whether two events, derangements of a list and generating an unusually frequent SRS, occur more or less often than they would if we used a true source of randomness.

Uniformity of permutations using derangements

A derangement is a permutation that leaves no item in the list fixed. For instance, $(2, 3, 4, 1)$ is a derangement of the list $(1, 2, 3, 4)$, but $(1, 3, 4, 2)$ is not. If all permutations occur with equal frequency, then the probability that a permutation is a derangement is asymptotically $p_0 = e^{-1} \approx 0.3678794$ as the length of the list of numbers increases; see Theorem 2 in Appendix A for a proof. We use the SPRT to test whether derangements of large lists occur with this frequency. We specify two possible alternatives in order to whether derangements occur more or less frequency than expected: $p_1 = 1.01p_0$ and $p_1 = 0.99p_0$.

Uniformity of SRSs using the most frequent categories

[147] developed a unique SPRT for testing the null hypothesis that a multinomial random variable has equal category probabilities $1/N$. We use this test for SRSs, treating each SRS as a category. Fix some integer $s < N$ and let $p_0 = s/N$. Let p denote the probability that the sample is from one of the s most frequent categories. After the Bth draw from the distribution, we determine which s categories occurred most frequently among the first $B - 1$ draws. We say that the event occurs if the Bth draw is among these categories. The problem is thus reduced to a test of a Bernoulli random variable as described above.

The test seems peculiar for two reasons: First, the power of the test depends on the choice of the parameter s. Second, events seem dependent because whether the Bth event

occurs depends on which categories were the most frequent among draws 1 through $B-1$. However, it is a valid test because under the null, p_0 is fixed and does not depend on which samples actually occur.

Tests of serial correlation

We measure serial correlation by the number of fixed points — values that remain in the same place — between a list and a permutation of it. The distribution of the number of fixed points for random permutations is a sum of dependent, identically distributed Bernoulli random variables. The probability of observing k fixed points is related to the number of derangements of the list. This distribution converges to the Poisson distribution with parameter 1 when the list length is sufficiently large. We use the chi-squared statistic to test for goodness of fit between the empirical distribution of fixed points from permuting a list with a PRNG and the Poisson distribution. See Theorem 3 in Appendix A for a proof of the approximation.

3.6 Discussion

Any PRNG with a finite state space cannot generate all possible samples from or permutations of sufficiently large populations. That can matter. A PRNG with a 32-bit state space cannot generate all permutations of 13 items. MT cannot generate all permutations of 2084 items.

Table 3.2 lists the PRNGs and sampling algorithms used in common statistical packages. Most use MT as their default PRNG; *is* MT adequate for Statistics? Section 3.3 shows that for some statistics, the L_1 distance between the theoretical value and the attainable value using a given PRNG is big for even modest sampling and permutation problems.

We have been searching for biases that are large enough to matter in $O(10^5)$ replications or fewer, and are not idiosyncratic to a few bad seeds. We have used each of the tests described in Section 3.5; so far, we have not found a statistic with consistent bias large enough to be detected in $O(10^5)$ replications. According to these tests, MT and the SHA256 PRNG are statistically indistinguishable from random. MT must produce bias in some statistics, but which?

We attempted to use a true source of randomness, such as the NIST randomness beacon and the ANU quantum random number server, as an empirical distribution in place of theoretical distributions. However, the rate at which they produce random numbers is too slow for our tests. For instance, the ANU quantum random number server only returns 1000 random numbers at a time, and the access is limited by the streaming speed of the internet [2]. Similarly, the NIST randomness beacon returns 512 bits for every call to its server [84]. Other providers sell large streams of random numbers. Some of the tests we ran on MT and the SHA-256 PRNG required at least 10^9 random numbers; using a true source of randomness was not practical.

Table 3.2: PRNGs and sampling algorithms used in common statistical and mathematical software packages. The 'floor' algorithm is the flawed multiply-and-floor method of generating pseudo-random integers. The 'mask' algorithm is better.

Package/Language	Default PRNG	Other	SRS Algorithm
SAS 9.2	MT	32-bit LCG	Floyd's ordered hash or Fan's method [30]
SPSS 20.0	32-bit LCG	MT1997ar	floor + random indices
SPSS $\leq$ 12.0	32-bit LCG		
STATA 13	KISS 32		PIKK
STATA 14	MT		PIKK
R $\leq$ 3.5	MT		floor + random indices
R $\geq$ 3.6	MT		mask + random indices
Python	MT		mask + random indices
MATLAB	MT		floor + PIKK

We recommend the following practices and considerations for using PRNGs in Statistics:

- Consider the size of the problem: are your PRNG and sampling algorithm adequate?
- Use a source of real randomness to set the seed with a substantial amount of entropy, e.g., 20 rolls of 10-sided dice.
- Record the seed so your analysis is reproducible.
- Avoid standard linear congruential generators, the Wichmann-Hill generator, and PRNGs with small state spaces.
- Use a cryptographically secure PRNG unless you know that MT is adequate for your problem.
- Use a sampling algorithm that does not overtax the PRNG. Avoid permuting the entire population to draw a random sample: do not use PIKK.
- Beware discretization issues in the sampling algorithm; many methods assume the PRNG produces $U[0,1]$ or $U[0,1)$ random numbers, rather than (an approximation to) numbers that are uniform on w-bit binary integers.

We also recommend that R and Python upgrade their algorithms to use best practices, and use cryptographically secure PRNGs by default, with an option of using MT instead in case the difference in speed matters. We have developed a CS-PRNG prototype as a Python package, discussed further in Chapter 4.

Chapter 4

Software for randomization

4.1 Introduction

In his seminal paper "The Future of Data Analysis," John Tukey argues that researchers analyzing their data ought to have access to a library of techniques [140].

> There is a corresponding danger for data analysis, particularly in its statistical aspects. This is the view that all statisticians *should* treat a given set of data in the same way, just as all British admirals, in the days of sail, maneuvered in accord with the same principles. The admirals could not communicate with one another, and a single basic doctrine was essential to coordinated and effective action. Today, statisticians can communicate with one another, and have more to gain by using special knowledge (subject-matter or methodological) and flexibility of attack than they have to lose by not all behaving alike.

Conducting data analysis in popular open source programs enables researchers to choose from a wide variety of contributed statistical methods, rather than having to rely on a limited set of off-the-shelf methods. Researchers can build upon each others' work without having to reinvent the wheel themselves. Though written over 50 years ago, Tukey's statement about communication speaks to the current landscape of statistical software. There is a push towards contributing code and using open source statistical packages like R and Python.

The S programming language, developed at Bell Laboratories in 1976, was the first language designed for interactive data analysis [6]. In the 1990s, Robert Gentleman and Ross Ihaka wrote R, an open source offshoot of S [50], which has been widely adopted by statisticians and domain researchersCurrently, there are over 13,000 user-contributed R packages on the CRAN package repository.[1]

Python is a high-level, general purpose, open source programming language that has been gaining popularity for data analysis. Like R, Python has a strong culture of users

[1] `https://cran.r-project.org` last visited January 9, 2019.

contributing packages. Packages like NumPy[2] and SciPy[3] provide core functionality for mathematical and statistical computing, including an n-dimensional array data structure, linear algebra operations, basic statistical distributions, random samplers, and optimization tools.

Python is not primarily intended for statistical analysis, so its contributed statistical packages are not as extensive as R's. Several recent third party Python packages, including Pandas[4], Statsmodels[5], and Scikit-learn[6], have converted some data scientists from R to Python. However, the capabilities for hypothesis testing are still rather limited: the main functions are parametric tests in SciPy and linear regression models in Statsmodels. As of 2014, when we began the `permute` project, there were no Python packages for permutation testing.

The default PRNG in NumPy is the Mersenne Twister. While Mersenne Twister is widely regarded as being sufficient for statistical applications, Chapter 3 explores ways in which it may be insufficient. The `secrets` module, included with standard Python distributions, produces cryptographically secure random numbers using the operating system's random number generator, but there's no guarantees about the quality of randomness or reproducibility across platforms. We developed `cryptorandom`, a CS-PRNG with reproducible outputs. The package also includes functions to generate pseudo-random samples and permutations using a variety of algorithms.

This chapter discusses the development of the `permute` and `cryptorandom` projects in detail, along with lessons learned about software development. Section 4.2 describes the packages' current capabilities. Section 4.3 describes the open source package development model, which is not taught in most programming courses. The "best practices" we adopted for these two software packages are powerful tools that enable efficient communication, collaboration, replication, exploration of ideas, and error checking. Section 4.4 describes issues we faced, both with the software itself and with managing open source projects more generally.

Computation and data analysis are central to modern research, not just in statistics but across fields. Section 4.5 concludes with how software development best practices can be applied to all computational research to boost productivity and make results more reproducible.

---

[2] `http://numpy.org`
[3] `http://scipy.org`
[4] `https://pandas.pydata.org`
[5] `https://www.statsmodels.org/stable/index.html`
[6] `https://scikit-learn.org/`

4.2 Software projects

`permute`

The `permute` project grew out of "Permutation Tests for Complex Data: Theory, Applications and Software," a textbook about permutation testing with numerous code and data examples [98]. The code is written in R and included in the book and online. We aimed to rewrite the examples in Python, creating modular functions that could be used with other datasets.

`permute` contains functions for the "building blocks" of permutation tests.[7] At the core, all permutation tests have two components: a group invariance under the null hypothesis and a test statistic. The group invariance is defined by the structure of the data and the actual (or implied) randomized experiment. The test statistic is a quantity that is computed from a sample, used to discriminate between the null hypothesis and an alternative hypothesis. Any statistic may be used, but some may give better discriminative power.

Permutation testing algorithm

1. **Calculate the test statistic on the observed data.**
2. **Generate the permutation distribution.** Repeat a large number B, e.g. 10,000, times:
   - Permute the data according to the group invariance under the null hypothesis.
   - Calculate the test statistic on the permuted data.
3. **Calculate the p-value.**

$$P = \frac{1 + \#\{\text{ permuted stats} \geq \text{observed stat}\}}{1 + B}.$$

This is both a biased Monte Carlo approximation to a conditional p-value given the data and a conservative conditional p-value for a randomized test.

Figure 4.1: Permutation testing algorithm used for all functions in `permute`.

Most functions in `permute` follow the algorithm illustrated in Figure 4.1. The name and inputs of the function are determined by the null hypothesis, and an argument to the function allows the user to specify a test statistic.

There are several ways to compute and interpret a p-value for a permutation test. The first way is to compare the observed value of the test statistic to the empirical distribution

[7] https://statlab.github.com/permute

of the test statistics for the permuted data, and to report the fraction that are more extreme than the observed value. This is an unbiased Monte Carlo approximation to the p-value, conditional on the orbit of the data. This is the approach used for the reported p-values in Chapter 2. The second way is to take this fraction, but add 1 to both the numerator and the denominator, as in Figure 4.1. This p-value has two interpretations: it is a *biased* Monte Carlo approximation to the p-value, conditional on the orbit of the data, and an exact p-value for a randomized test, conditional on the random elements of the orbit.[8] `permute` defaults to this second approach, with an option to use the unbiased approximate p-value.

Core functionality

The scope of the `permute` project is large and we have only begun to scratch the surface of implementing permutation tests. There are functions for common tests, including a one sample test for symmetry about a point, a test for correlation between two lists of measurements, and the canonical two sample test.

Some experimental designs group subjects based on their characteristics, then randomly assign treatment within groups. In these scenarios, *stratified* permutations are appropriate: permutations are done independently in each group, then statistics are aggregated across groups. `permute` has a module specifically for stratified tests, including a two sample test for designs where two treatments are randomly assigned to subjects in groups, independently across the groups.

Nonparametric combination of tests

Researchers often test many hypotheses about the same data, whether about multiple variables or multiple outcomes. For instance, pharmaceutical developers care about whether a drug affects a variety of disease symptoms and run many statistical tests for these associations using measurements from a single group of patients. If tests are dependent, then a naive multiple testing correction like the Bonferroni adjustment can be overly conservative, resulting in tests with low power. Adjusted p-values may be informative for each individual hypothesis, but if the substantive research question is about a global hypothesis (e.g. no effect of treatment on any outcome, whatsoever) then it is desirable to consider p-values jointly.

It can be useful to test a *composite null hypothesis* in such situations. A composite null is a multivariate hypothesis that can be written as an intersection of univariate null hypotheses. When the variables under consideration are independent, Fisher's combination method can be used to obtain a global p-value: under the composite null hypothesis, if the p-values $p_1, \ldots, p_K$ for each individual hypothesis are continuous, then $-2\sum_{k=1}^{K} \log p_k$ has a chi-squared distribution with $2K$ degrees of freedom. When the p-values are dependent, this approach is not valid.

[8] See `https://github.com/pbstark/S157F17/blob/master/combining-tests.ipynb` (last visited January 23, 2019).

[98] proposed the non-parametric combination (NPC) framework for scenarios when the variables in a composite hypothesis test are dependent. The individual tests can be combined in an efficient way, so the computation time does not grow linearly with the number of hypotheses. This framework opens the door to permutation testing in domains that have traditionally relied on parametric hypothesis tests, such as psychology and social sciences [19], and in domains where hypothesis testing has been limited due to the complexity of data, such as shape analysis [17].

The NPC module in `permute` implements the algorithm in Figure 4.2. The idea is to permute dependent variables in lock-step, so any dependencies among them are preserved when approximating their multivariate permutation distribution under the intersection null hypothesis. A combining function such as Fisher's is used to combine variable-level p-values into a single global value.

Datasets

Clear use cases and examples are just as important as the code itself. `permute` contains 17 example datasets that users can load and explore. The data include examples from [98], therapeutic interventions with autistic children assessed by different raters over time [76], the student evaluations of teaching from the U.S. dataset in Chapter 2 [66], and GERD symptom scores from a clinical trial at multiple locations [92]. The goal is to provide example code along with every dataset.

`cryptorandom`

The `cryptorandom`[9] project grew out of concerns about the default PRNG in Python, as discussed in Chapter 3. This package was intended to provide a plug-in replacement for NumPy's Mersenne Twister, so all of the random number generation methods that are built on top of NumPy could use a CS-PRNG. However, as we discuss in Section 4.4, NumPy's architecture prevents users from doing this.

The package is built on top of the `hashlib`[10] library, a module for secure hash algorithms that comes built-in with standard Python distributions. We used the SHA256 hash function, which takes input messages of arbitrary lengths and outputs a 256-bit integer. The package uses the algorithm from Figure 3.1 to generate pseudo-random sequences of 256 bits.

The package uses the method described in Section 3.4 to generate pseudo-random integers on $\{1, \ldots, n\}$ using random bits directly (as opposed to the multiply-and-floor method). `cryptorandom` uses these 256 bits efficiently by using only as many as it needs from a single SHA256 output, then storing the remaining bits and using them for subsequent calls to the random integer generator. This makes the code more efficient by reducing the number of calls to SHA256.

[9] https://statlab.github.com/cryptorandom

[10] https://docs.python.org/3.7/library/hashlib.html

NPC algorithm

1. **Calculate the vector of observed test statistics** $(T_1(X_1), \ldots, T_K(X_K))$**.**

2. **Generate the** K**-variate permutation distribution.** For each b in $1, 2, \ldots, B$, for a large number B, e.g. 10,000:

   - Apply the same permutation π_b, coming from the group invariance under the null hypothesis, to all variables $X_1, \ldots, X_K$.
   - Calculate the test statistics on the permuted data $(T_1(\pi_b(X_1)), \ldots, T_K(\pi_b(X_K)))$.

3. **Convert test statistics to** p**-values.** For each $k = 1, \ldots, K$ and $b = 1, \ldots, B$, let
$$P_{bk} \equiv \frac{1 + \#\{T_k(\pi_\cdot(X_k)) \geq T_k(\pi_b(X_k))\}}{1 + B}.$$
   Do the same for the observed statistics to get $P_{1,obs}, \ldots, P_{K,obs}$.

4. **Combine tests from each permutation.** Using a combining function Φ, calculate
$$T_b \equiv \Phi(P_{1,b}, \ldots, P_{K,b})$$
   and
$$T_{obs} \equiv \Phi(P_{1,obs}, \ldots, P_{K,obs})$$

5. **Calculate the global** p**-value as**
$$P = \frac{1 + \#\{b : T_b \geq T_{obs}\}}{1 + B}.$$

Figure 4.2: NPC algorithm to test $H_0 : \cap_{k=1}^{K} H_{0k}$ implemented in `permute.npc`.

The `cryptorandom` PRNG includes a jump ahead feature. This allows the user to move ahead a specified number of steps in the PRNG's state space. This is a useful capability to efficiently reproduce randomization results and to generate independent streams of pseudo-random numbers. Independent streams are crucial for using a PRNG for simulations running in parallel: if streams of pseudo-random numbers overlap, then simulation results will not be independent. A new, independent stream can be created by jumping ahead from the starting point of the current stream past its end state. In `cryptorandom`, jumping ahead n states amounts to appending n zero bytes to the state.

In addition to the CS-PRNG, `cryptorandom` contains a sampling module with a variety of algorithms for mapping pseudo-random numbers to samples and permutations. The permutation algorithms include the Fisher-Yates-Knuth-Durstenfeld (FYKD) shuffle [55], the random sort method (generate random floats and sort the list by their order), and sampling indices without replacement. The sampling algorithms include the PIKK algorithm (see Section 3.4), using FYKD and taking the first k, recursive sampling [24], Waterman's Algorithm R [55], Vitter's Algorithm Z [142], sampling k indices without replacement, exponential sampling with weights, and elimination sampling with weights. Each function uses the SHA256 PRNG by default, but allows the user to input another PRNG with appropriate random float and random integer methods.

4.3 Development practices

We adopted common "best practices" for open source software projects. These concepts are used widely in the Python community but aren't taught broadly in computing classes. Only recently have workshops like Software Carpentry [150] and some university courses, such as Berkeley's Statistics 159/259: Reproducible and Collaborative Statistical Data Science[11] begun to teach these tools and workflows. While these coding practices have a steep learning curve, they make work more efficient in the long run by facilitating collaboration on large scale software projects and streamlining individual workflows.

Version control

In a collaborative project, it is imperative to preserve versions of the project at different time points. Doing so allows everyone to monitor progress and view additions, to track who has contributed and how much, and to recover old versions in case it becomes necessary to revert back. Saving multiple versions of files with dates or version numbers in the file name, using a hosting system such as Dropbox, or using program-specific features such as "track changes" in Microsoft Word solve some, but not all, of these problems.

A *version control system* tracks project files in a repository and allows the user to track changes in commits to the system history [77]. There are several common version control systems in use, including SVN, Mercurial, and Git. We prefer Git, as its compatibility with the GitHub website adds additional capabilities. By hosting a repository on GitHub, one can back up all work in the cloud and share all files with collaborators.

A full discussion of GitHub's capabilities is beyond the scope of this chapter, but we will describe our preferred development workflow with GitHub. Figure 4.3 illustrates this workflow, which we call the "wheel" model of collaboration because information travels in one direction around the diagram. The main repository, which is traditionally named `upstream`, is isolated and changes are never made directly in it. Instead, the first step is for a contributor to make two copies of the repository. The first copy is called a *clone*, which is a

[11] See, for instance, `http://berkeley-stat159.github.io` for lessons.

copy of the main repository on a local machine where one can independently start a history of commits. The second copy is called a *fork*, which is hosted on GitHub and is traditionally named `origin`.

When someone has created a new feature they want to contribute, they *push* (the term meaning to send the new history of commits) changes to their `origin` fork. To incorporate changes on their GitHub fork into the main repository, the user makes a *pull request*, asking the maintainers of the main repository to incorporate their new feature. Contributors can push to their own fork, but not to the main repository; the package maintainers serve as gatekeepers to the main repository and decide which pull requests to accept. Finally, once new features are incorporated in the main repository, users can *pull* these changes into the clone on their local machines.

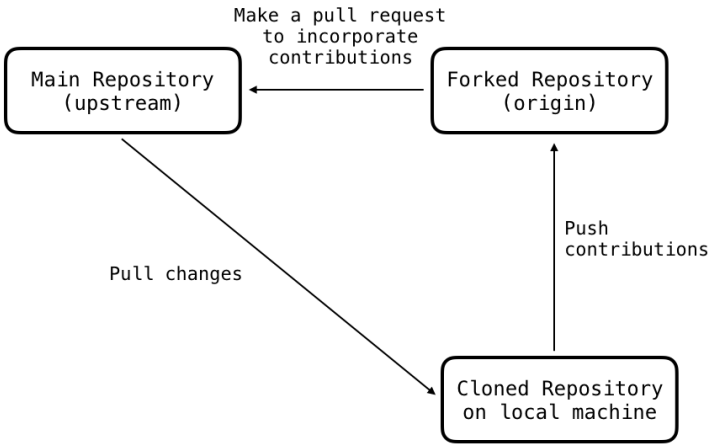

Figure 4.3: The "wheel" model of collaboration using GitHub.

This workflow sets up checks and balances so the codebase remains pristine. The ability to clone a GitHub repository enables anyone with an idea to contribute to a package, while the pull request step allows package maintainers to review code before it enters the official codebase. Automated testing and continuous integration (discussed below) facilitate the review process. This peer review process is public on GitHub's issue tracking system, with a forum in which anyone can comment. Transparent communication is critical for smooth open source development.

Documentation

Documentation for software can take many forms, including websites with narratives, galleries of example use cases, and API documentation. These forms of documentation serve different purposes: teaching new users how to use the package, publicizing a project, and

improving its discoverability [36]. We focused on creating narrative documentation and API documentation, and delayed examples until the codebase is more comprehensive.

Using GitHub makes it easy to include narrative documentation with any project. The documentation with lowest barrier to entry is a README file in a repository. README files are written in Markdown, a markup language designed for easy readability and plain text formatting. When a GitHub repository contains a README file, the Markdown gets rendered at the bottom of the repository webpage. It is common for packages to include a short description, installation instructions, and links to more information in the README.

Many open source Python projects, including `permute` and `cryptorandom`, use Sphinx[12] to generate documentation. Sphinx is designed to use reStructured Text (reST), another markup language with plain text formatting, with additional features to create table of contents organization and display code, images, and mathematics. With Sphinx, reST can be compiled into HTML for a website. GitHub also supports web documentation with GitHub Pages[13], which provides users a free website with each repository. The website contents are stored in the repository; updating the website is as simple as making a pull request (Figure 4.3). Integrating Sphinx and GitHub Pages makes it simple to create public-facing narrative documentation for a project.

Narrative documentation is useful for providing an overview of what the software does and how to download it, but users also need granular API documentation about its functionality. It is common practice to include comments within source code, usually at the very beginning of a function or object definition, succinctly describing what the code block does. In Python, these *docstrings* appear on screen when a user queries an object's help page. Moreover, when integrated with Sphinx, docstrings can be automatically rendered into HTML webpages. Our packages use the NumPy docstring standard,[14] which provides a standard syntax for describing an object's purpose, the expected inputs, and its outputs.

Finally, it is worth mentioning that good code is readable. Readability is not itself documentation, but is critical for others to be able to interpret the code, for more efficient debugging, and for collaborators to build upon the code in future iterations. Writing docstrings is only one component of writing readable code; other practices include writing modular functions that do one specific task, giving functions and variables descriptive names, and favoring several simple lines of code over long, complex lines of code. The PEP 8 style guide for Python code[15] gives a comprehensive list of coding conventions that improve readability, from code layout to naming conventions. Our projects follow PEP 8.

Testing

People make mistakes, so it is imperative to test code. Tests do not guarantee that the code works, but a battery of passing tests can give confidence that it does.

[12] `http://sphinx-doc.org`
[13] `https://pages.github.com`
[14] `https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard`
[15] `https://www.python.org/dev/peps/pep-0008/`

Tests can appear both within and outside the source code. *Assertions* are statements that enforce the programmer's expectations of how the program should behave, for instance, assertions about data types a program expects. If the inputs have the wrong type, an assertion will give an error and halt the program. *Unit tests* are written separately from the source code and check that the code returns the correct results. They should be run on simple examples where the desired output is known ahead of time. *Regression tests* check that the behavior of the code remains stable, and can help ensure that code continues to work as expected after changes are made [151].

Unit testing can be difficult for applications involving pseudo-random numbers. Without knowing the seed, we may not know exactly what the code will output. However, we can check distributional properties that are expected from theory, like uniformity of p-values over many calls to the code. Tests that show the average behavior is as expected give evidence (but not certainty) that the code for an individual call is correct.

Automation

Compiling documentation and running unit tests can be tedious, but they are critical for a project's health. They often involve many steps that must be done in sequence, making it less likely that an individual will remember how to execute them right. Automating these tasks ensures that the computer runs every step in the right order every time the task needs to be done.

The `make` system is one tool to simplify the process. A `make` file runs a series of commands to turn a *source* (a dataset, script, or some other set of files) into a *target* (a desired output). The syntax expresses dependencies amongst files, and when one source is changed, running the command `make` will update all targets that depend on that file. `permute` and `cryptorandom` both use `make` files to run unit tests, to build the documentation, and to update the package website.

Once this system is set up to run all of a package's tests, the project can deploy a *continuous integration system* to automate code testing before contributions enter the main codebase. We are most familiar with Travis CI[16], a continuous integration system commonly used with GitHub repositories. These systems take the new version of the codebase, run the entire test suite, and report the results. Continuous integration helps package maintainers review new contributions: Travis CI will raise a flag whenever code fails unit tests. Continuous integration systems can also report *code coverage*, the fraction of lines in the code that are run in a test; larger code coverage gives greater confidence that the code does what it is supposed to do.

[16] `https://travis-ci.org`

4.4 Challenges

We have encountered a number of road bumps working on `permute` and `cryptorandom`. Some of these have been issues with Python itself, while others relate to open source projects more generally.

The `cryptorandom` project was intended to be a plug-in replacement for NumPy's pseudo-random number generator, so users could import the CS-PRNG and otherwise follow an identical workflow. Unfortunately, NumPy's architecture will not allow this. NumPy tracks the internal state of the PRNG separately from calls to the PRNG, which are done using functions specific to Mersenne Twister. SciPy offers an extensive library of statistical distributions, but all of its random number generation abilities rely on NumPy and do not allow users to specify another PRNG. Therefore, we would have to implement a library of common statistical distributions within `cryptorandom` and users would have to alter their workflows to use the CS-PRNG to generate anything besides random integers and floats.

While the SHA256 hash function in `hashlib` is written in C and is itself fast, using it from Python in a CS-PRNG is slow. We spent many hours and sought advice from experts on speeding up the code. Type conversions in Python are the bottleneck: the seed and counter must be converted from string to byte string before passing them into the SHA256, then the output must be converted from a Python byte string to a long integer. Moreover, this process cannot be vectorized; the SHA256 hash function can only take one input at a time.

Generating a single random sample with `cryptorandom` takes less than a millisecond, but this is still too slow for running simulations. Even after optimizing the code, our PRNG takes nearly 200 times as long as NumPy's Mersenne Twister to generate a large number of random sample. A simulation that would take one hour using NumPy's default PRNG would take more than a week using `cryptorandom`. The PRNG can be used in situations where only one random sample is needed, for instance in Chapters 6 and 8, where a sample of ballots to audit only needs to be drawn once. We are working to implement the entire process in C, so the type conversions and looping will run faster.

The time I felt I could spend working on these Python packages was constrained by the competing demands of graduate school. A survey of graduate students and postdocs engaged in computational research showed that participants felt that developing computational resources was valued far less than publishing, winning awards, and writing grants for tenure—and by extension, not valued in their search for academic jobs [37]. However, many researchers build software tools because they are important for their work. Using "best practices" while creating software requires more time than using other practices, and may take time away from activities that are perceived as more valuable. For instance, writing the functional parts of software tend to be more rewarding, while activities like writing documentation fall by the wayside when time is limited, as there are few incentives and rewards to spending time on it [36].

Distributing the work to teams, such as undergraduate volunteers, has had varying degrees of success. We have written thorough contribution instructions in the GitHub repos-

itory and on the project website to lower the barrier to contribution, but there is a steep learning curve for the wheel workflow, rigorous documentation, and unit tests. While many undergraduate volunteers have had a strong background in programming, many have been confused by the statistics: their code follows best practices but is incorrect. Many students moved on before finishing their contributions; the lack of tangible rewards or consequences makes it easy for volunteers to quit.

4.5 Conclusion

Scientists must choose their statistics carefully and test their software thoroughly. Researchers continue to use inappropriate statistical tests because there are few non-technical resources that teach more appropriate methods; parametric tests are commonplace and widely used in the literature; and existing software for permutation tests is not sufficiently flexible or accessible to most users. `permute` and `cryptorandom` attempt to address these issues by providing a library of flexible statistical tests and sampling methods with thorough examples and documentation.

Best practices that the open source software community has developed over the years, like collaboration using distributed version control systems, documentation, and unit testing, are indispensable tools that could help academics make their research more reproducible and reduce errors [77]. Researchers have been moving away from using GUI tools to using scripts to program all of their tasks, from reading the data to producing a result. Using open source software practices in academic research benefits science by making it possible to reproduce the analyses of a paper and delve into further scientific inquiry. The "show me, don't tell me" ethos makes scientists more trustworthy to the public [126].

Moreover, adopting these practices would make it easier for statisticians to contribute packages with new statistical methods. Users in the R community have contributed thousands of statistical packages, but many of these packages are highly specialized or lack clear documentation on how to use them. Open source best practices make packages more valuable by requiring thorough documentation on inputs, outputs, and functionality; allowing users to contribute new features and comment on problematic ones; and ensuring that code works as expected in transparent, reproducible tests.

Part II

Election Integrity: Security and Risk-Limiting Audits

Chapter 5

Election integrity and electronic voting machines in 2018 Georgia

5.1 Introduction

The state of Georgia played a key role in the civil rights movement of the twentieth century and recently, their election integrity issues have come back into relief. The state has a history of problems: voting machines that are vulnerable to undetectable security breaches, systematic voter suppression, and serious security flaws with their data systems.

The 2018 midterm election in Georgia was particularly contentious and put Georgia in the national spotlight. Civil rights groups alleged that former Secretary of State Kemp attempted to suppress voters, as many counties closed polling places and the Secretary of State's office deleted or held up thousands of voter registrations in the months leading up to the election [42, 78, 83]. A federal lawsuit against the Secretary of State's office, calling for the state to quickly replace electronic voting with paper ballots in polling places, reminded the public of the security issues of the statewide direct recording electronic (DRE) systems [25]. Ultimately, the in-person voting in the 2018 election was done using DRE machines.

The 2018 election produced anomalous results that many security and election experts have blamed on DREs. The accuracy of the reported vote counts cannot be checked via an audit of paper records, for instance by a risk-limiting audit, because the DREs used in Georgia do not produce a voter-verifiable paper record.

This chapter begins with an overview of DRE voting systems and their known security flaws. While the focus is DREs and their use in Georgia, I provide a fuller picture of the election landscape by starting with a short history of recent election integrity issues in the state, including allegations of voter suppression and corruption. I describe what took place in the months leading up to Georgia's 2018 election and provide statistical evidence that random chance alone is unlikely to explain the anomalous election results.

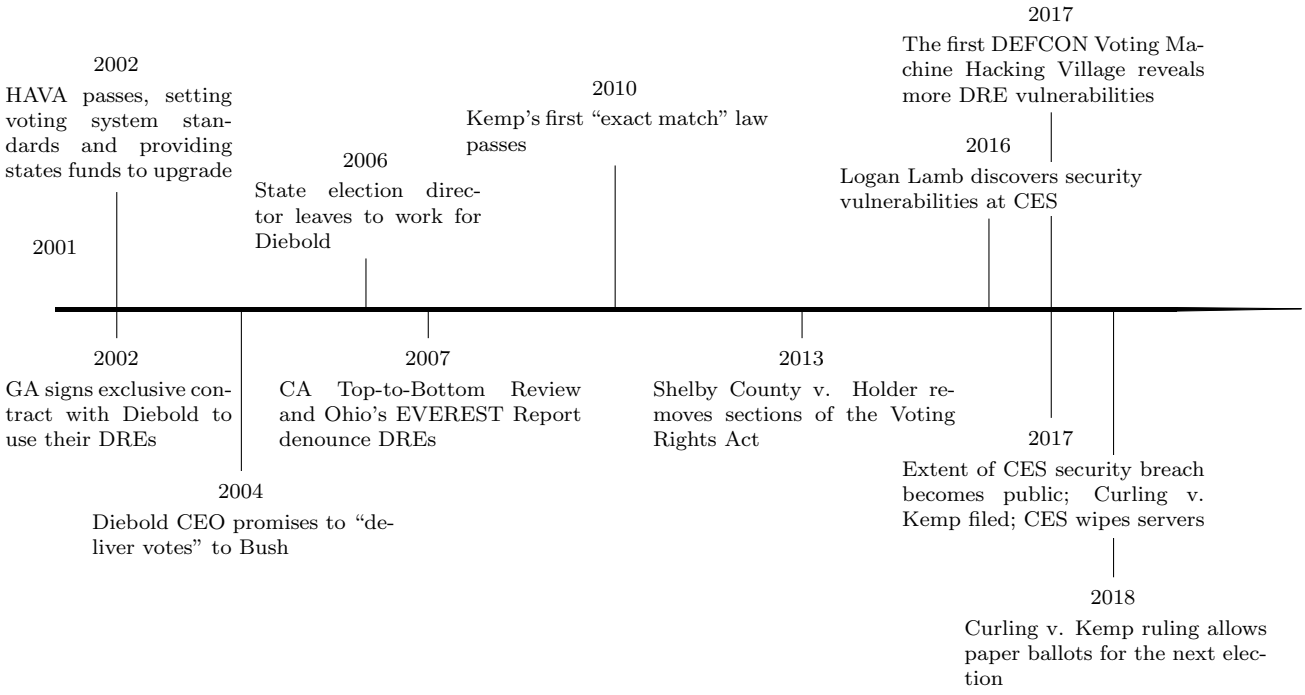

Figure 5.1: Timeline of events pertaining to the use of DREs in Georgia since HAVA

5.2 Background

DRE Voting Systems

Congress passed the Help America Vote Act (HAVA) in 2002 after Florida experienced substantial issues with its voting systems and voter access in the 2000 presidential election. HAVA set mandatory minimum standards for states to administer elections, including creating the provisional voting system, requiring statewide voter registration databases, and providing funds for states to upgrade their voting equipment to meet accessibility standards. The standards for better voting equipment were loose: states were expected to replace punchcard and lever voting systems and to provide at least one accessible voting machine per polling place [53].

The two main options states considered when upgrading their voting equipment were optical scanners and DREs. DREs were an attractive option because the computer-based system eliminates the need to print and store paper ballots. They allow states to offer

ballots in multiple languages and satisfy the accessibility requirement with aids for disabled voters, such as audio tools. In addition, DREs tabulate election results faster than a central count optical scanner. While HAVA only required that each polling place have one accessible voting machine, some states opted to use DREs exclusively [157]. In 2002, the voting machine manufacturers offering DREs were Diebold Election Systems, Election Systems and Software (ES&S), Hart InterCivic, and Sequoia Voting Systems. For the majority of this chapter, we will focus on Diebold; the company has been the lone election system provider in Georgia and has been the star of many scandals that highlight the issues with DREs and electronic voting [53].

Almost immediately, DREs began causing problems in major elections. In the 2002 Florida primaries, some machines in Miami-Dade county failed to turn on, creating long lines that prevented some voters from casting their votes. In New Mexico, faulty programming caused machines to drop a quarter of votes. In Virginia, the software on 10 machines caused one vote to be subtracted for every 100 votes cast for a particular candidate [152]. Faulty programming of ballot layouts can cause votes intended for one candidate to be recorded as votes for another candidate [85]. Numerous examples of DRE failures have continued to accumulate [26].

In 2007, two major statewide studies in California and Ohio gave concrete and conclusive evidence that DREs have incorrigible security flaws. The California Secretary of State's Top-to-Bottom Review of the state's voting systems found substantial physical and technological security flaws with Premier Election Systems' (formerly Diebold) DREs, including software vulnerabilities that would allow someone to install malicious software that records votes incorrectly or miscounts them; susceptibility to software viruses that propagate from voting machine to voting machine; unprotected information linked to each vote that could compromise the secrecy of the ballot; access to the voting system server software, allowing an attacker to corrupt the election management system database; "root access" to the voting system, allowing manipulation of every setting of devices on the network; and numerous physical security weaknesses that would allow an attacker to disable parts of the device using standard office tools [14]. The Ohio Secretary of State's EVEREST report found that the software for Premier DREs was "unstable" and lacked "sound software and security engineering practices" [73]. California decertified Premier, Hart InterCivic, and Sequoia DREs and the EVEREST study prompted Ohio to move to optical scanners.

More recently, the annual DEFCON hacker conference started a "Voting Village" to study and hack electronic voting machines. In 2017, the Voting Village acquired over 25 pieces of election equipment used in the United States and made them available to participants to hack. While the EVEREST study plan restricted the types of hacks that could be deployed against the machines, there were no such restrictions at DEFCON. Within minutes, hackers with limited knowledge of voting systems were able to penetrate several DREs, including the Premier AccuVote-TSx used in Georgia. They uncovered serious hardware vulnerabilities, including a chip whose removal would disable the entire machine [9]. The Voting Village has become a regular part of DEFCON as voting system vulnerabilities persist: the organizers reported in 2018, "while, on average, it takes about six minutes to vote, machines in at least

15 states can be hacked with a pen in two minutes" [10].

Security experts recommend that jurisdictions still using DREs conduct forensic audits of voting machines, both before and after the election. An examination of the software and machines done by an independent, neutral party could detect tampering, bugs, or hacking, and would help discourage malicious attacks [26]. But historically, it has been illegal to examine the voting machine software because it is considered "proprietary information" [97]. Without a forensic audit or a reliable paper trail, there is no way to check whether the voting machine accurately captured voters' intent.

Voter-verifiable paper audit trails (VVPATs) were introduced as a solution to improve DRE auditability. VVPATs involve adding a printer to DREs and displaying a paper record behind glass to the voter, who can then verify that their vote was cast as intended. The paper record can be used in a post-election audit, and gives assurance in case the electronic mode of casting votes fails. The NIST Auditability Working Group found that there is no satisfactory way to audit DREs without a trustworthy paper record such as a VVPAT [104].

However, VVPATs are not a perfect solution. If the printer fails or malfunctions, the paper record is compromised. They are not accessible to blind voters. Most importantly, the method of printing does not lend itself to efficient audits. VVPATs are typically printed on continuous, uncut rolls of paper, which then need to be unrolled and segmented to count votes. Moreover, some of these rolls are thermal paper, which degrade quickly when exposed to heat, light, or human touch [104].

Verifiable does not guarantee verified: voters may not check or correctly verify a VVPAT. Research on the usability of electronic review screens, an electronic copy of the ballot displayed for the voter to verify their choices before casting their vote, has shown that voters don't use the review screen effectively. Study participants often walked away from the DRE before the review screen was displayed. Errors in votes occurred at the same rate when the review screen was shown and when it wasn't shown. In experiments where contests were added to or removed from ballots, only 32% of study participants noticed the change on the review screen. In other experiments where the wrong candidate was marked, only 37% of study participants noticed the error on the review screen, though 95% of participants reported that they had checked their ballot either somewhat or very carefully [29]. These studies suggest that VVPATs may not reflect voter intent, even when voters claim to review them.

Many states have been phasing out DREs. In 2006, nearly 40% of voters used DREs to cast their vote. In 2016, 28 states were using DREs in some capacity, but most jurisdictions used paper, optical scan, or an electronic method with a paper backup [152]. Only 5 states still use paperless DREs exclusively: Delaware, Georgia, Louisiana, New Jersey, and South Carolina.

Voter suppression

Georgia was one of the states with a history of voter suppression that faced heightened scrutiny under the Voting Rights Act of 1965. Voter suppression efforts were revitalized in

2013 with the Supreme Court's decision in Shelby County v. Holder. Sections 4(b) and 5 of the 1965 Voting Rights Act required jurisdictions with prior evidence of racial discrimination to get "preclearance" from the federal government before changing their election policies, to ensure that new policies would not be discriminatory. In 2013, the Supreme Court ruled that these sections were unconstitutional because they placed undue burden on some states based on outdated evidence of discrimination against minority voters. [117].

Without strict oversight, states that were previously subject to the preclearance rule of the Voting Rights Act were free to reinstate some discriminatory policies. Almost immediately, states began to close polling places and create stricter voter registration laws. Under the preclearance rule, counties would have to show that these changes would not differentially disenfranchise minority voters. With the rule gone, a number of states including Arizona, Louisiana, and Texas have made changes that affect a large number of registered voters, many of whom are black or Latino [141].

Strategically closing polling places can reduce voter turnout for specific demographic groups. It can create long lines in the remaining polling places and force people to travel farther to access a polling place, dissuading voters from even attempting to vote. Since the court ruling, nearly a thousand polling places in the United States have been closed; many of those polling places served African American communities [141]. Local election officials in Georgia have closed 214 precincts—nearly 8% of the state's polling places—since 2012 [83]. Officials claim that consolidating low-turnout polling places is purely a cost-saving measure [148]. However, opponents have pointed out that 39 of the Georgia counties that have closed polling places have poverty rates higher than the state average and 30 have significant African American populations [83]. These closures would not have been possible prior to 2013 under the Voting Rights Act's preclearance rule.

Polling place closures in Georgia have attracted attention from the national media. In August 2018, Randolph County debated whether to close seven of its nine polling places. An independent consultant endorsed by Secretary of State Kemp suggested the proposal, which cited unresolved ADA compliance issues and low turnout as reasons to close the polling places. Civil rights groups across the country spoke out against the proposal as a blatant attempt to disenfranchise minority voters: Randolph County has around 7000 registered voters and is 61% African American. Ultimately, the county elections board voted it down, but the debate brought the broader issue of Georgia's voter suppression efforts into the spotlight [42].

Under Secretary of State Kemp's watch, voter registrations have been unpredictable. Since 2012, over 1.4 million voter registrations have been cancelled in "routine maintenance" of the voter rolls. Kemp's office says that these records were marked as inactive according to standard federal and state law. However, the "exact match" laws that Kemp has put in place make it harder to get back on the voter registration roll. The first iteration of the exact match law was passed in 2010 with preclearance from the federal government, but was dismantled after a federal lawsuit found it unconstitutional in 2016 [139]. The current law, passed in 2017, stipulates that voter applications must exactly match information on file with the Georgia Department of Driver Services or the Social Security Administration. Any

non-exact matches—as innocuous as a missing hyphen—are flagged as pending and those individuals have 26 months to correct their application. While they are still eligible to vote, civil rights groups say that a pending application might discourage someone from casting their ballot. In the months before the November 2018 election, over 53,000 voter registration applications were pending. Nearly 70% of pending applications are black, more than double the 32% black population percentage in the state [78].

Kemp vehemently denies any attempts to suppress minority voters. He has tried to distance himself from the polling place closures, stating that the decision to close a precinct is up to the county election officials. However, in 2015 his office provided a document giving county officials guidance on why and how to close polling places [83]. Kemp claims that the racial disparity in pending voter registration applications is due to sloppy voter registration efforts. His office specifically called out the New Georgia Project, a voter registration group (founded by Kemp's gubernatorial opponent Stacey Abrams) that targeted African American voters and used primarily paper forms. His office says that poorly trained canvassers who helped voters fill out paper registration forms are a main reason for the large proportion of minority non-exact matches [78].

Election Integrity in Georgia

Georgia was the first state to adopt DREs in November 2002 after HAVA was passed: just days later, the state signed a $54 million contract with Diebold Election Systems to use their AccuVote-TS/TSx DREs exclusively [157].

During the summer of 2002, Diebold began preparing more than 20,000 DREs to be used across the state for the November election. A former Diebold employee alleged that during this time, before the machines had been delivered to counties, employees were asked to install three software patches on all of the DREs that would be used statewide that year. These patches did not undergo the federal certification process for voting equipment [155]. Another former Diebold employee reported that the president of Diebold's election unit, Bob Urosevich, came to the warehouse himself to order the installation of uncertified software patches on about 5,000 machines in DeKalb and Fulton , two historically Democratic counties. He instructed employees not to discuss the patches with county officials [54].

This is particularly troubling, given that key contests in Georgia's 2002 election defied poll predictions. Longtime Democratic Senator Max Cleland was predicted to beat Republican opponent Saxby Chambliss by 3%, but in fact lost his seat by a 7% margin. Democratic Incumbent Governor Roy Barnes was predicted to win 51% to 40%, but in fact lost to Republican candidate Sonny Perdue by 6% [34, 97]. These Republican victories were a surprise in a historically Democratic state: Perdue was Georgia's first Republican governor in 130 years. Diebold's partisan leanings raised eyebrows. The company's CEO, Wally O'Dell, was a member of President Bush's "Rangers and Pioneers," an elite group of Bush supporters who raised funds for the president's 2004 campaign. At a fundraiser, O'Dell announced that he was "committed to helping Ohio deliver its electoral votes to the president" [146]. Despite

obvious concerns, the election could not be audited because of the purely electronic voting machines.

Even as security experts raised red flags about Diebold's touchscreen machines, Georgia's government officials continued to endorse them. Former Secretary of State Cathy Cox, who signed the 2002 contract with Diebold, had strong ties to the company: she even allowed them to use her portrait on their promotional materials. The election director she appointed, Kathy Rogers, supported the use of Diebold machines and helped kill house bills that would have required paper records. In 2006, she resigned and took a job as Government Liaison at Diebold [143]. Cox's successor as Secretary of State, Karen Handel, was a vocal supporter of paper audit trails and acknowledged publicly that she would not interact with Rogers as Diebold's liaison, due to the conflict of interest. However, Handel later reversed her position on electronic voting and it came to light that she had received $25,000 in campaign contributions from employees connected with Diebold's lobbying firm, Massey & Bowers [38]. Diebold has used money and power to ensure that security concerns about their machines would not stop them from remaining the sole voting machine provider in Georgia.

Georgia's election security issues reach beyond their voting machines. In 2016, a cybersecurity researcherat Oak Ridge National Laboratory, Logan Lamb, discovered that he could download files from the state's "secure" election server. Among these files were the entire voter registration database for the state of Georgia, including sensitive personal information, instructional PDFs with passwords for poll workers to sign into a central server on Election Day, and software files for the state's ExpressPoll pollbooks that are used to verify voters' eligibility [158]. This intrusion would have allowed Lamb to alter entries in the voter registration database or the pollbooks, preventing some voters from casting their ballots. Lamb's discovery was not a purely theoretical concern. The possibility of malicious hacking is real: an NSA investigation found that Russian hackers targeted 39 states in the summer and fall leading up to the 2016 presidential election [105].

These were not the only security concerns at the state's Center for Election Services (CES), housed at Kennesaw State University under a long-standing contract with the Secretary of State. For instance, the center was using an outdated version of Drupal, their content management software, that would allow hackers to seize control of a site using the software. A software patch had been available since 2014, but the election center had not installed it. Lamb notified the executive director of the CES, Merle King, of the problems he uncovered; King agreed to fix them and allegedly pressed Lamb not to talk to the media or other officials about the security issues [159].

The CES did not secure their server, nor did they inform anyone about the breach. In March 2017, another cybersecurity researcher found that the CES had not secured its files properly. He brought the issue to the attention of an IT staff at Kennesaw State University, who then raised the issue with higher authorities who made it public. This was the first time that the Secretary of State's office heard about the breach; the CES did not notify them when Lamb discovered the problem. In response to this poor management, the Secretary of State office signed a new agreement with Kennesaw State University to transfer the CES to its own offices [159].

In July 2017, state voters and the Coalition for Good Governance filed a lawsuit against the Georgia Secretary of State, alleging that they ignored evidence that the state's electoral system is vulnerable to fraud and hacking. The plaintiffs demanded that the state use paper ballots in future elections to guard against election interference [25, 26]. They requested to examine the Kennesaw State University election servers as a crucial piece of evidence in the case. Four days after the group filed the lawsuit, IT employees at the CES wiped their servers of all prior election data. They later degaussed two remaining servers. There is no evidence that the election center deliberately destroyed potential evidence, and the Secretary of State's office claims that the servers were wiped before they were officially served with the lawsuit in late July. However, Kemp's office was alerted about the lawsuit and declined to comment in the days between its filing and when the CES wiped its servers [122]. Key evidence for the legal case was permanently erased.

The November 2018 Election

The lawsuit, Curling v. Kemp, continued into September 2018, just before the midterm elections (and is ongoing at the time of this report). The current director at the CES testified that the server that each county uses to construct its ballots is "air-gapped" from the internet, but that he uses thumb drives, email, and an online repository to store and move data. A county official testified that they use "analog phone lines" to transmit results to the Secretary of State. Computer scientists have testified that these are all vulnerable channels [79]. Given that DREs are known to have security flaws and are not auditable, it is all the more crucial that the state's data and procedures before and after Election Day are secure.

The state had no good answer to the security concerns raised by the plaintiffs, but argued that there was not enough time before the election to switch to paper ballots. Their timeline for upgrading Georgia's voting systems was several years: in 2017, Kemp established the Secure, Accessible, & Fair Elections (SAFE) Commission to decide upon new voting system options, with a goal of having new systems in place for the 2020 election. Ultimately, U.S. District Judge Amy Totenberg ruled that the trade-off between election integrity and the feasibility of making changes before the impending election tipped in favor of continued use of DREs. While Judge Totenberg noted that the plaintiffs provided sufficient evidence that paperless voting has the potential to cause irreparable harm to voters, the burden of switching to paper ballots so close to the election would cause even more harm to voters by causing bureaucratic confusion.

> Ultimately, any chaos or problems that arise in connection with a sudden rollout of a paper ballot system with accompanying scanning equipment may swamp the polls with work and voters – and result in voter frustration and disaffection from the voting process. There is nothing like bureaucratic confusion and long lines to sour a citizen. And that description does not even touch on whether voters themselves, many of whom may never have cast a paper ballot before, will have

> been provided reasonable materials to prepare them for properly executing the paper ballots.

The judge scolded the state in her ruling, saying that the evidence and testimony "indicated that the Defendants and State election officials had buried their heads in the sand" [25].

The November 2018 election in Georgia exemplified many of the state's election integrity issues discussed above. Numerous civil rights groups urged Secretary of State Kemp to step down for ethical reasons. Kemp refused to resign on the grounds that other elected officials have not done so when they run for higher offices [149]. Only when the reported results showed that he won did he step down.

Per Judge Totenberg's ruling, Election Day voting in November 2018 was carried out on paperless DREs. The performance issues with DREs are not just hypothetical: machines in four polling places in Gwinnett County malfunctioned and forced voters to use paper ballots, causing some voters to wait four hours to cast their vote [64]. Anomalies appeared in the vote totals: the rate of undervotes in the Lieutenant Governor's contest was unusually high, compared to historical Lieutenant Governor races and compared to other statewide contests on the ballot. The Coalition for Good Governance brought another lawsuit against the Georgia Secretary of State, calling for a redo of the Lieutenant Governor's contest [22]. Statistical evidence of anomalies in this election appears in Section 5.3.

After the election, Kemp's office hastened to certify the election results six days before state law required it. Another civil rights group sued to delay the certification. Judge Totenberg ruled against Kemp, ordering election officials to review the nearly 27,000 provisional ballots cast. Provisional ballots are cast by voters who cannot verify their registration or identification; deliberately omitting provisional ballots is one way to disenfranchise voters and would have ensured that the margin between Kemp and his opponent Stacey Abrams remained large enough to avoid a runoff election [11].

Election cooking continues to be common practice, even with all eyes on Georgia. The SAFE Commission was scheduled to make recommendations for new voting systems in January, 2019. In early January, the Democratic Party of Georgia called on Kemp to delay any decision to purchase new voting systems as more bad behavior came to light: Kemp appointed Charles "Chuck" Harper, chief lobbyist for ES&S (the voting machine company that eventually acquired Diebold), as Deputy Chief of Staff in the governor's office [95].

5.3 The 2018 Georgia election

.

Shortly after the November 2018 election, The Coalition for Good Governance filed a lawsuit against the Secretary of State of Georgia, Robyn Crittenden, demanding a redo of the Lieutenant Governor election between Republican candidate Geoff Duncan and Democratic candidate Sarah Riggs Amico. The plaintiffs blamed malfunctioning DREs for anomalous

results in the Lieutenant Governor race, and only in that race. The plaintiffs did not specify the cause of the malfunction, whether it be faulty programming, hacking, or something else [22]. Numerous voters reported irregularities when attempting to cast their vote for Lieutenant Governor on DREs, including many who reported that the race did not appear on their ballot until they were shown the review screen. Moreover, evidence suggests that undervote rates on touchscreen voting machines were higher in predominantly African American precincts [43]. The judge overseeing the case initially agreed to let the plaintiffs examine the memory, not the programming, of machines in three counties. She eventually dismissed the case [156].

In this section, I present statistical evidence to support this claim that DREs may have malfunctioned. The first line of evidence comes from reported results at the county level: the rate of undervotes in the Lieutenant Governor race is unusually high among DRE votes (those cast on Election Day and advance in-person) compared to the rate of undervotes on absentee ballots. This pattern does not hold in as many counties for other statewide contests on the ballot. The second line of evidence comes from a single polling place in Clarke County. Photographs of printed poll tapes from each AccuVote-TS machine show how many votes were cast for each candidate in each contest. On one out of seven machines, the majority of votes were for the Republican candidate in every statewide contest. On the remaining six machines, the majority were for the Democratic candidate, matching the overall results at the polling place. For both sets of data, we use permutation tests to demonstrate that these phenomena are implausible unless something went wrong.

Undervotes for Lieutenant Governor

Undervotes occur when a voter does not mark a vote for any candidate in a contest. Historical data shows that the rate of undervotes is lowest for high profile contests, such as presidential and gubernatorial contests, and that the undervote rate generally increases for contests further down the ballot. In this case, the Lieutenant Governor race had a 4% undervote rate, while the next contest on the ballot had an undervote rate of 1.4%. Moreover, this pattern appeared only in votes cast on DREs—Election Day votes and advance in-person votes.

Data

Data were downloaded from the Georgia Secretary of State's website at `https://results.enr.clarityelections.com/GA/91639/222278/reports/detailxml.zip`. Data included total ballots cast in each county, total ballots cast by each mode of voting for each candidate by county. The file did not report ballots cast in each county by vote type. Therefore, we assumed that the total number of ballots cast, separated by county and mode of voting, was equal to the maximum number of votes cast in each contest for that county and mode of voting. These maxima were used as the baseline against which to calculate the number of undervotes in each contest, by county and mode of voting.

Methods

A hypergeometric test was used to identify anomalous counties. Under the null hypothesis, in each county undervotes are equally likely to occur among vote by mail ballots and votes cast on DREs[1]. While there may be different party preferences amongst these two groups of voters, there is no reason to believe that interest in a *contest* should differ. Conditional on the number of votes cast by mail, the number of votes cast on DREs, and the number of undervotes in the county, the number of undervotes among DRE votes is distributed as hypergeometric under the null hypothesis. This hypergeometric test is a permutation test.

Results

In 101 of 159 Georgia counties, the difference in undervote rates between mail votes and DRE votes is statistically significant at level 0.01%. In contrast, in the contests for Secretary of State, Attorney General, Commissioner of Agriculture, Commissioner of Insurance, State School Superintendent, Commissioner of Labor, Public Service Commission District 3, and Public Service Commission District 5, the difference is statistically significant in no more than 5 counties. Table 5.1 shows the exact counts.

Table 5.1: Counties with statistically significant ($p < 0.0001$) disparities in undervote rates between paper ballots and DREs.

Contest	Counties with significant undervote rate disparities
Lt. Governor	101
Secretary of State	4
Attorney General	4
Commissioner of Agriculture	5
Commissioner of Insurance	4
State School Superintendent	5
Commissioner of Labor	2
Public Service Commission District 3	4
Public Service Commission District 5	4

Party Preferences in Winterville Train Depot Polling Place

Data

A citizen photographed printed poll tapes from the seven DRE machines in the Winterville Train Depot polling place in Clarke County. The photographs were transcribed to CSV and

[1] Provisional ballots were omitted from this analysis as they are handled separately.

double checked by a second person.

The DREs showed similar numbers of voters (117, 135, 131, 133, 135, 144, 135). In this polling place, Democratic candidates won a majority in each of ten statewide contests. Every DRE reported a majority of votes for the Democratic candidate except machine 3, which reported a majority for the Republican candidate in every contest.

The Winterville Train Depot polling place is just one polling place in Georgia where voters photographed poll tapes after the polls closed. It was not selected at random. However, there is no reason to believe that problems are confined to this polling place.

Methods

If voters were directed to DREs as if at random, then the percentage of votes for each candidate should be roughly equal on each machine. Conditional on the number of ballots on each machine and the total number of votes for each candidate across machines, all permutations of votes across machines are equally likely under the null hypothesis. Permutations were done using `cryptorandom`. The test statistic was the largest absolute difference between the expected and actual fraction of Republican votes in each contest. The p-values are two-sided. The p-values for each contest were combined using Fisher's combination function to obtain a global p-value.

Results

On the assumption that voters were directed to DREs as if at random, the chance any of the seven machines would show disparities as large as machine 3 did in individual contests ranges from less than 1% to approximately 15%. Seven of the ten values are significant at level 5% or below; see Table 5.2. The global p-value for the ten tests is 0.00009%; the chance that any of the seven machines would show anomalies as large as machine 3 did is less than one in a million.

These results are entirely driven by the results on machine 3. If the Democratic and Republican party labels were flipped on the third machine, the anomaly disappears. The global p-value for the ten tests is 97%. For individual contests, no p-value is below 0.280 on the assumption that voters are directed to DREs as if at random, compared with values as small as 0.008 (and seven values below 5 percent) for the actual poll tapes. See Table 5.3.

These tests strongly suggest that machine 3 had some software or hardware problem: misconfiguration, error, defect, hack, or malfunction. The most plausible explanation is that machine 3 was misconfigured in a way that caused votes for Republican candidates to be recorded as votes for Democratic candidates, and vice versa.

5.4 Conclusion

The 2018 midterms demonstrated some of the election administration issues that plague Georgia. In the weeks leading up to the election and for several weeks after, citizens chal-

Table 5.2: Consistency of Results across DREs in Winterville Train Station Polling Place

Contest	p-value
Governor	0.114
Lt. Governor	0.025
Secretary of State	0.018
Attorney General	0.151
Commissioner of Agriculture	0.026
Commissioner of Insurance	0.030
State School Superintendent	0.097
Commissioner of Labor	0.008
Public Service Commission District 3	0.046
Public Service Commission District 5	0.025

Table 5.3: Consistency of Results across DREs in Winterville Train Station Polling Place, if D and R were flipped on machine 3.

Contest	p-value
Governor	0.464
Lt. Governor	0.795
Secretary of State	0.450
Attorney General	0.543
Commissioner of Agriculture	0.734
Commissioner of Insurance	0.604
State School Superintendent	0.807
Commissioner of Labor	0.797
Public Service Commission District 3	0.280
Public Service Commission District 5	0.939

lenged the way that the Secretary of State's office handled provisional ballots and strict voter registration laws, alleging that these practices were intended to disenfranchise minority voters. Touchscreen DRE voting machines were used statewide, even after security experts voiced their concerns and a nonprofit organization sued the state to replace them with hand-marked paper ballots. There is evidence that the DREs did in fact malfunction; we presented statistical anomalies that suggest that DREs failed to record a large percentage of votes cast in the Lieutenant Governor's race. Moreover, there is evidence to suggest these missing votes occurred at a higher rates in jurisdictions with large African American populations [43]. The Secretary of State has expressed unwillingness to investigate these issues.

Lawmakers are set to replace the state's DREs with a new system. The debate has focused on two options: hand-marked paper ballots with optical scanners and touchscreen

ballot-marking devices (BMDs). In February 2019, the state legislature voted to purchase BMDs statewide [82]. While BMDs do produce a paper record, they also have problems: the ability to cast a vote relies on the machine being functional on Election Day; there is limited evidence that voters verify their selections on the summary card; and BMDs require the same trust in software as DREs [125]. Security experts including Wenke Lee, the only computer scientist on the state's SAFE Commission, have warned against BMDs. However, many lawmakers claim that the touchscreen interface of these machines will be more modern, efficient, and familiar to voters.

How the state proceeds will reveal Georgia lawmakers' commitment to election integrity. The state House Minority Leader Bob Trammell expressed his stance on the evidence for hand-marked paper ballots [82]:

> It's unequivocally clear that cybersecurity experts have expressed concerns about the ballot-marking devices. It comes down to whether you think the opinion of election officials ... is more important than the issue of credentialed experts in the field talking about a material risk to the voting process.

Chapter 6

Risk-limiting audits by stratified union-intersection tests of elections (SUITE)

6.1 Introduction

States have begun piloting and passing laws mandating risk-limiting audits (RLAs) as additional measure to ensure the fairness and security of their elections; see Chapter 1.1 for details on RLAs. The most efficient and transparent sampling design for RLAs selects individual ballots uniformly at random, with or without replacement [130]. Risk calculations for such samples can be made simple without sacrificing rigor [131, 62]. However, to audit contests that cross jurisdictional boundaries then requires coordinating sampling in different counties, and may require different counties to use the lowest common denominator method for assessing risk from the sample, which would not take full advantage of the capabilities of some voting systems. For instance, any system that uses paper ballots as the official record can conduct *ballot-polling* audits, while *ballot-level comparison audits* require systems to generate *cast vote records* that can be checked manually against a human reading of the paper [63, 62]. (These terms are described in Section 6.3.)

Stratified RLAs have been considered previously, primarily to conform with legacy audit laws under which counties draw audit samples independently of each other, but also to allow auditors to start the audit before all vote-by-mail or provisional ballots have been tallied, by sampling independently from ballots cast in person, by mail, and provisionally, as soon as subtotals for each group are available [128, 46]. However, extant methods address only a single approach to auditing, batch-level comparisons, and only a particular test statistic.

Here, we introduce SUITE, a more general approach to conducting RLAs using stratified samples. SUITE is a twist on *intersection-union* tests [98], which represent the null hypothesis as the intersection of a number of simpler hypotheses, and the alternative hypothesis as a union of their alternatives. In contrast, here, the null is the union of simpler hypotheses,

and the alternative is the intersection of their alternatives. The approach involves finding the maximum p-value over a vector of nuisance parameters that describe the simple hypotheses: all allocations of tabulation error across strata for which a full count would find a different electoral outcome than was reported. (A *nuisance parameter* is a property of the population that is not of direct interest, but that affects the probability distribution of the data. *Overstatement* is error that made the margin of one or more winners over one or more losers appear larger than it really was. The total overstatement across strata determines whether the reported outcome is correct; the overstatements in individual strata are nuisance parameters that affect the distribution of the audit sample.)

The basic building block for the method is testing whether the overstatement error in a single stratum is greater than or equal to a quota. Fisher's combining function is used to merge p-values for tests in different strata into a single p-value for the hypothesis that the overstatement in every stratum is greater than or equal to its quota. If that hypothesis can be rejected for *all* stratum-level quotas that could change the outcome—that is, if the maximum combined p-value is sufficiently small—the audit can stop.

It is not actually necessary to consider all possible quotas: the p-value has a bounded modulus of continuity, which allows us to find upper and lower bounds everywhere using only values on a discrete grid. We present a numerical procedure, implemented in Python, to find bounds on the maximum p-value when there are two strata. The procedure can be generalized to more than two strata.

Section 6.2 presents the new approach to stratified auditing. Section 6.3 illustrates the method by solving a problem pertinent to Colorado: combining ballot polling in one stratum with ballot-level comparisons in another. This requires straightforward modifications to the mathematics behind ballot polling and ballot-level comparison to allow the overstatement to be compared to specified thresholds other than the overall contest margin; those modifications are described in Sections 6.4 and 6.5. We applied SUITE in pilot RLAs in Michigan, where some jurisdictions have cast vote records that can be linked to vote-by-mail ballots, but not Election Day ballots; we discuss the pilots in Chapter 7.

Section 6.7 gives numerical examples of simulated audits, using parameters intended to reflect how the procedure would work in Colorado. We provide example software implementing the risk calculations for our recommended approach in Python Jupyter notebooks.[1] Section 6.8 gives recommendations and considerations for implementation.

6.2 Stratified audits

Stratified sampling involves partitioning a population into non-overlapping groups and drawing independent random samples from those groups. [128, 46] developed RLAs based on comparing stratified samples of batches of ballots to hand counts of the votes in those batches: batch-level comparison RLAs, using a particular test statistic. The method we develop here is more general and more flexible: it can be used with any test statistic, and test statistics

[1] See `https://github.com/pbstark/CORLA18`.

in different strata need not be the same—which is key to combining audits of ballots cast using diverse voting technologies.

Here and below, we consider auditing a single plurality contest at a time, although the same sample can be used to audit more than one contest (and super-majority contests), and there are ways of combining audits of different contests into a single process [124, 131]. We use terminology drawn from a number of papers, notably [62].

An *overstatement error* is an error that caused the margin between *any* reported winner and *any* reported loser to appear larger than it really was. An *understatement error* is an error that caused the margin between *every* reported winner and *every* reported loser to appear to be smaller than it really was. Overstatements cast doubt on outcomes; understatements do not, even though they are tabulation errors.

We use w to denote a reported winner and ℓ to denote a reported loser. The total number of reported votes for candidate w is V_w and the total for candidate ℓ is V_ℓ. Thus $V_w > V_\ell$, since w is reported to have gotten more votes than ℓ.

Let $V_{w\ell} \equiv V_w - V_\ell > 0$ denote the contest-wide margin (in votes) of w over ℓ. We have S strata. Let $V_{w\ell,s}$ denote the margin (in votes) of reported winner w over reported loser ℓ in stratum s. Note that $V_{w\ell,s}$ might be negative in one stratum, but $\sum_{s=1}^{S} V_{w\ell,s} = V_{w\ell} > 0$. Let $A_{w\ell}$ denote the margin (in votes) of reported winner w over reported loser ℓ that a full hand count would show: the *actual* margin, in contrast to the *reported* margin $V_{w\ell}$. Reported winner w really beat reported loser ℓ if and only if $A_{w\ell} > 0$. Define $A_{w\ell,s}$ to be the actual margin (in votes) of w over ℓ in stratum s.

Let $\omega_{w\ell,s} \equiv V_{w\ell,s} - A_{w\ell,s}$ be the *overstatement* of the margin of w over ℓ in stratum s. Reported winner w really beat reported loser ℓ if and only if $\omega_{w\ell} \equiv \sum_s \omega_{w\ell,s} < V_{w\ell}$.

An RLA is a test of the hypothesis that the outcome is wrong, that is, that w did not really beat ℓ: $\sum_s \omega_{w\ell,s} \geq V_{w\ell}$. The null is true if and only if there exists *some* S-tuple of real numbers $(\lambda_s)_{s=1}^S$ with $\sum_s \lambda_s = 1$ such that $\omega_{w\ell,s} \geq \lambda_s V_{w\ell}$ for all s.[2] Thus if we can reject the conjunction hypothesis $\cap_s \{\omega_{w\ell,s} \geq \lambda_s V_{w\ell}\}$ at significance level α for all (λ_s) such that $\sum_s \lambda_s = 1$, we can stop the audit, and the risk limit will be α.

Fisher's combination method

Fix $\lambda \equiv (\lambda_s)_{s=1}^S$, with $\sum_s \lambda_s = 1$. To test the conjunction hypothesis that stratum null hypotheses are true, that is, that $\omega_{w\ell,s} \geq \lambda_s V_{w\ell}$ for all s, we use Fisher's combining function. Let $p_s(\lambda_s)$ be the p-value of the hypothesis $\omega_{w\ell,s} \geq \lambda_s V_{w\ell}$. If the null hypothesis is true, then

$$\chi(\lambda) = -2 \sum_{s=1}^{S} \ln p_s(\lambda_s) \tag{6.1}$$

[2] "If" is straightforward. For "only if," suppose $\omega_{w\ell} \geq V_{w\ell}$. Set $\lambda_s = \frac{\omega_{w\ell,s}}{\sum_t \omega_{w\ell,t}}$. Then $\sum_s \lambda_s = 1$, and $\omega_{w\ell,s} = \lambda_s \omega_{w\ell} \geq \lambda_s V_{w\ell}$ for all s.

has a probability distribution that is dominated by the chi-square distribution with $2S$ degrees of freedom.[3] Fisher's combined statistic will tend to be small when all stratum-level null hypotheses are true. If any is false, then as the sample size increases, Fisher's combined statistic will tend to grow.

If, for all λ with $\sum_s \lambda_s = 1$, we can reject the conjunction hypothesis at level α (i.e., if the minimum value of Fisher's combined statistic over all λ is larger than the $1-\alpha$ quantile of the chi-square distribution with $2S$ degrees of freedom), the audit can stop.

If the audit is allowed to "escalate" in steps, increasing the sample size sequentially, then either the tests used in the separate strata have to be sequential tests, or multiplicity needs to be taken into account, for instance by adjusting the risk limit at each step. Otherwise, the overall procedure can have a risk limit that is much larger than α. For examples of controlling for multiplicity when using non-sequential testing procedures in an RLA, see [128, 127].

The stratum-level p-value $p_s(\lambda)$ could be a p-value for the hypothesis $\omega_{w\ell,s} \geq \lambda_s V_{w\ell}$ from any test procedure. We assume, however, that p_s is based on a one-sided test, and that the tests for different values of λ "nest" in the sense that if $a > b$, then $p_s(a) > p_s(b)$. This monotonicity is a reasonable requirement because the evidence that the overstatement is greater than a should be weaker than the evidence that the overstatement is greater than b, if $a > b$. In particular, this monotonicity holds for the tests proposed in Sections 6.4 and 6.5.

One could use a function other than Fisher's to combine the stratum-level p-values into a p-value for the conjunction hypothesis, provided it satisfies these properties (see [98]):

- the function is non-increasing in each argument and symmetric with respect to rearrangements of the arguments
- the combining function attains its supremum when one of the arguments approaches zero
- for every level α, the critical value of the combining function is finite and strictly smaller than the function's supremum.

For instance, one could use Liptak's function, $T = \sum_i \Phi^{-1}(1-p_i)$, or Tippett's function, $T = \max_i(1-p_i)$.

Fisher's function is convenient for this application because the tests in different strata are independent, so the chi-squared distribution dominates the distribution of $\chi(\cdot)$ when the null hypothesis is true. If tests in different strata were correlated, the null distribution of the combination function would need to be calibrated by simulation; some other combining function might have better properties than Fisher's [98].

[3] If the stratum-level tests had continuously distributed p-values, the distribution would be exactly chi-square with $2S$ degrees of freedom, but if any of the p-values has atoms when the null hypothesis is true, it is in general stochastically smaller. This follows from a coupling argument along the lines of Theorem 4.12.3 in [39].

Maximizing Fisher's combined p-value for $S = 2$

For the remainder of the chapter, we specialize to $S = 2$ strata. The set of $\lambda = (\lambda_1, \lambda_2)$ such that $\sum_s \lambda_s = 1$ is then a one-dimensional family: if $\lambda_1 = \lambda$, then $\lambda_2 = 1 - \lambda$. For a given set of data, finding the maximum p-value over all λ is thus a one-dimensional optimization problem.

To find the maximum combined p-value (equivalently, the minimum value of $\chi(\lambda)$), we first compute $\chi(\lambda)$ for a grid of values $\lambda_0 = \lambda_- < \lambda_1 < \cdots < \lambda_n = \lambda_+$, where $\lambda_{j+1} - \lambda_j = \delta$. To account for the fact that χ may be smaller for values of λ between the sampled values, we bound its variation between grid points. The *local modulus of continuity*, $\omega(\delta; \lambda)$, is an upper bound on how much χ can change in a δ-neighborhood of λ:

$$|\chi(\lambda) - \chi(\lambda + \delta)| \le \omega(\delta; \lambda).$$

For all $\lambda \in [\lambda_j, \lambda_{j+1}]$,

$$\chi(\lambda) \ge \max\{\chi(\lambda_j) - \omega(\lambda - \lambda_j; \lambda_j), \chi(\lambda_{j+1}) - \omega(\lambda_{j+1} - \lambda; \lambda_{j+1})\}. \tag{6.2}$$

Thus for all $\lambda \in [\lambda_-, \lambda_+]$,

$$\chi(\lambda) \ge \chi_n^* \equiv \min_{j=0}^{n-1} \min_{\mu \in [\lambda_j, \lambda_{j+1}]} \max\{\chi(\lambda_j) - \omega(\mu - \lambda_j; \lambda_j), \chi(\lambda_{j+1}) - \omega(\lambda_{j+1} - \mu; \lambda_{j+1})\}. \tag{6.3}$$

If that lower bound is above χ_α, the audit can stop. If not, then either the true minimum is below χ_α or the grid $\{\lambda_j\}$ is too coarse.

Finding a lower bound on $\chi(\lambda)$ between grid points can be simplified in a number of ways, for instance, using a (bound on a) global modulus of continuity instead of a local modulus of continuity. Section 6.6 derives the a bound on the global modulus of continuity for a special case, where one stratum uses ballot polling and the other uses a comparison audit.

6.3 Auditing cross-jurisdictional contests

As mentioned above, stratified sampling can simplify audit logistics by allowing jurisdictions to sample ballots independently of each other, or by allowing a single jurisdiction to sample independently from different collections of ballots (e.g., vote-by-mail versus in-person on Election Day). SUITE allows stratified samples to be combined into an RLA of contests that include ballots from more than one stratum.

We present an example where SUITE is helpful for a different reason: it enables an RLA to take advantage of differences among voting systems to reduce audit sample sizes, which solves a current problem in Colorado.

CRS 1-7-515 requires Colorado to conduct risk-limiting audits beginning in 2017. The first risk-limiting election audits under this statute were conducted in November, 2017; the second were conducted in July, 2018.[4] Counties cannot audit contests that cross jurisdictional

[4] See https://www.sos.state.co.us/pubs/elections/RLA/2017RLABackground.html

boundaries (*cross-jurisdictional* contests, such as gubernatorial contests and most federal contests) on their own: margins and risk limits apply to entire contests, not to the portion of a contest included in a county. Colorado has not yet conducted an RLA of a cross-jurisdictional contest, although it has performed RLA-like procedures on individual jurisdictions' portions of some cross-jurisdictional contests. To audit statewide elections and contests that cross county lines, Colorado will need to implement new approaches and make some changes to its auditing software, RLATool.

Colorado's voting systems are heterogeneous. Some counties (containing about 98% of active voters, as of this writing) have voting systems that export cast vote records (CVRs) in a way that the paper ballot corresponding to each CVR can be identified uniquely and retrieved. We call counties with such voting systems *CVR counties*. In CVR counties, auditors can manually check the accuracy of the voting system's interpretation of individual ballots. In other counties (*legacy* or *no-CVR* counties) there is no way to check the accuracy of the system's interpretation of voter intent for individual ballots.

Contests entirely contained in CVR counties can be audited using ballot-level comparison audits [62], which compare CVRs to the auditors' interpretation of voter intent directly from paper ballots. Ballot-level comparison audits are currently the most efficient approach to risk-limiting audits in that they require examining fewer ballots than other methods do, when the outcome of the contest under audit is in fact correct. Contests involving no-CVR counties can be audited using ballot-polling audits [63, 62], which generally require examining more ballots than ballot-level comparison audits to attain the same risk limit.

Colorado's challenge is to audit contests that include ballots cast in both CVR counties and no-CVR counties. There is no literature on how to combine ballot polling with ballot-level comparisons to audit cross-jurisdictional contests that include voters in CVR counties and voters in no-CVR counties.[5]

Colorado could simply revert to ballot-polling audits for cross-jurisdictional contests that include votes in no-CVR counties, but that would entail a loss of efficiency. Alternatively, Colorado could use batch-level comparison audits, with single-ballot batches in CVR counties and larger batches in no-CVR counties.[6] The statistical theory for such audits has been worked out (see, e.g., [128, 124, 129, 131] and Section 6.4, below); indeed, this is the method that was used in several of California's pilot audits, including the audit in Orange County, California. However, batch-level comparison audits were found to be less efficient than ballot-polling audits in these pilots [87].

Moreover, to use batch-level comparison audits in Colorado would require major changes to RLATool, for reporting batch-level contest results prior to the audit, for drawing the sample, for reporting audit findings, and for determining when the audit can stop. The changes would include modifying data structures, data uploads, random sampling proce-

[5] See [108] for a different (Bayesian) approach to auditing contests that include both CVR counties and no-CVR counties. In general, Bayesian audits are not risk-limiting.

[6] Since so few ballots are cast in no-CVR counties, cruder approaches might work, for instance, pretending that no-CVR counties had CVRs, but treating any ballot sampled from a no-CVR county as if it had a 2-vote overstatement error. See [5].

dures, and the county user interface. No-CVR counties would also have to revise their audit procedures. Among other things, they would need to report vote subtotals for physically identifiable groups of ballots before the audit starts. No-CVR counties with voting systems that can only report subtotals by precinct might have to make major changes to how they handle ballots, for instance, sorting all ballots by precinct. These are large changes.

We show here that SUITE makes possible a "hybrid" RLA that keeps the advantages of ballot-level comparison audits in CVR counties but does not require major changes to how no-CVR counties audit, nor major changes to RLATool. The key is to use stratified sampling with two strata: ballots cast in CVR counties and those cast in no-CVR counties.

In order to use Equation 6.1, we must develop stratum-level tests for the overstatement error that are appropriate for the corresponding voting system. Sections 6.4 and 6.5 describe these tests for overstatement in the CVR and no-CVR strata, respectively.

6.4 Comparison audits of overstatement quotas

To use comparison auditing in the approach to stratification described above requires extending previous work to test whether the overstatement error is greater than or equal to $\lambda_s V_{w\ell}$, rather than simply $V_{w\ell}$. The derivation considers only a single contest, but the MACRO test statistic [124, 131] automatically extends the result to auditing any number of contests simultaneously. The derivation is for plurality contests, including "vote-for-k" plurality contests. Majority and super-majority contests are a minor modification [128].[7]

Notation

- $\mathcal{W}$: the set of reported winners of the contest
- $\mathcal{L}$: the set of reported losers of the contest
- N_s ballots were cast in stratum s. (The contest might not appear on all N_s ballots.)
- P "batches" of ballots are in stratum s. A batch contains one or more ballots. Every ballot in stratum s is in exactly one batch.
- n_p: number of ballots in batch p. $N_s = \sum_{p=1}^{P} n_p$.
- $v_{pi} \in \{0, 1\}$: reported votes for candidate i in batch p
- $a_{pi} \in \{0, 1\}$: actual votes for candidate i in batch p. If the contest does not appear on any ballot in batch p, then $a_{pi} = 0$.

[7] So are some forms of preferential and approval voting, such as Borda count, and proportional representation contests, such as D'Hondt [133]. For a derivation of ballot-level comparison risk-limiting audits for super-majority contests, see `https://github.com/pbstark/S157F17/blob/master/audit.ipynb`. (Last visited 14 May 2018.) Changes for IRV/STV are more complicated.

- $V_{w\ell,s} \equiv \sum_{p=1}^{P}(v_{pw} - v_{p\ell})$: Reported margin in stratum s of reported winner $w \in \mathcal{W}$ over reported loser $\ell \in \mathcal{L}$, in votes.
- $V_{w\ell}$: overall reported margin in votes of reported winner $w \in \mathcal{W}$ over reported loser $\ell \in \mathcal{L}$ for the entire contest (not just stratum s)
- $V \equiv \min_{w\in\mathcal{W},\ell\in\mathcal{L}} V_{w\ell}$: smallest reported overall margin in votes between any reported winner and reported loser
- $A_{w\ell,s} \equiv \sum_{p=1}^{P}(a_{pw} - a_{p\ell})$: actual margin in votes in the stratum of reported winner $w \in \mathcal{W}$ over reported loser $\ell \in \mathcal{L}$
- $A_{w\ell}$: actual margin in votes of reported winner $w \in \mathcal{W}$ over reported loser $\ell \in \mathcal{L}$ for the entire contest (not just in stratum s)

Reduction to maximum relative overstatement

If the contest is entirely contained in stratum s, then the reported winners of the contest are the actual winners if

$$\min_{w\in\mathcal{W},\ell\in\mathcal{L}} A_{w\ell,s} > 0.$$

Here, we address the case that the contest may include a portion outside the stratum. To combine independent samples in different strata, it is convenient to be able to test whether the net overstatement error in a stratum is greater than or equal to a given threshold.

Instead of testing that condition directly, we will test a condition that is sufficient but not necessary for the inequality to hold, to get a computationally simple test that is still conservative (i.e., the level is not larger than its nominal value).

For every winner, loser pair (w, ℓ), we want to test whether the overstatement error is greater than or equal to some threshold, generally one tied to the reported margin between w and ℓ. For instance, for a hybrid stratified audit, we set the threshold to be $\lambda_s V_{w\ell}$.

We want to test whether

$$\sum_{p=1}^{P}(v_{pw} - a_{pw} - v_{p\ell} + a_{p\ell})/V_{w\ell} \geq \lambda_s.$$

The maximum of sums is not larger than the sum of the maxima; that is,

$$\max_{w\in\mathcal{W},\ell\in\mathcal{L}} \sum_{p=1}^{P}(v_{pw} - a_{pw} - v_{p\ell} + a_{p\ell})/V_{w\ell} \leq \sum_{p=1}^{P} \max_{w\in\mathcal{W},\ell\in\mathcal{L}}(v_{pw} - a_{pw} - v_{p\ell} + a_{p\ell})/V_{w\ell}.$$

Define

$$e_p \equiv \max_{w\in\mathcal{W}\ell\in\mathcal{L}}(v_{pw} - a_{pw} - v_{p\ell} + a_{p\ell})/V_{w\ell}.$$

Then no reported margin is overstated by a fraction λ_s or more if

$$E \equiv \sum_{p=1}^{P} e_p < \lambda_s.$$

Thus if we can reject the hypothesis $E \geq \lambda_s$, we can conclude that no pairwise margin was overstated by as much as a fraction λ_s.

Testing whether $E \geq \lambda_s$ would require a very large sample if we knew nothing at all about e_p without auditing batch p: a single large value of e_p could make E arbitrarily large. But there is an *a priori* upper bound for e_p. Whatever the reported votes v_{pi} are in batch p, we can find the potential values of the actual votes a_{pi} that would make the error e_p largest, because a_{pi} must be between 0 and n_p, the number of ballots in batch p:

$$\frac{v_{pw} - a_{pw} - v_{p\ell} + a_{p\ell}}{V_{w\ell}} \leq \frac{v_{pw} - 0 - v_{p\ell} + n_p}{V_{w\ell}}.$$

Hence,

$$e_p \leq \max_{w \in \mathcal{W}, \ell \in \mathcal{L}} \frac{v_{pw} - v_{p\ell} + n_p}{V_{w\ell}} \equiv u_p. \tag{6.4}$$

Knowing that $e_p \leq u_p$ might let us conclude reliably that $E < \lambda_s$ by examining only a small number of batches—depending on the values $\{u_p\}_{p=1}^P$ and on the values of $\{e_p\}$ for the audited batches.

To make inferences about E, it is helpful to work with the *taint* $t_p \equiv \frac{e_p}{u_p} \leq 1$. Define $U \equiv \sum_{p=1}^P u_p$. Suppose we draw batches at random with replacement, with probability u_p/U of drawing batch p in each draw, $p = 1, \ldots, P$. (Since $u_p \geq 0$, these are all positive numbers, and they sum to 1, so they define a probability distribution on the P batches.)

Let T_j be the value of t_p for the batch p selected in the jth draw. Then $\{T_j\}_{j=1}^n$ are IID, $\mathbb{P}\{T_j \leq 1\} = 1$, and

$$\mathbb{E}T_1 = \sum_{p=1}^{P} \frac{u_p}{U} t_p = \frac{1}{U} \sum_{p=1}^{P} u_p \frac{e_p}{u_p} = \frac{1}{U} \sum_{p=1}^{P} e_p = E/U.$$

Thus $E = U\mathbb{E}T_1$. So, if we have strong evidence that $\mathbb{E}T_1 < \lambda_s/U$, we have strong evidence that $E < \lambda_s$.

This approach can be simplified even further by noting that u_p has a simple upper bound that does not depend on v_{pi}. At worst, the reported result for batch p shows n_p votes for the "least-winning" apparent winner of the contest with the smallest margin, but a hand interpretation would show that all n_p ballots in the batch had votes for the runner-up in that contest. Since $V_{w\ell} \geq V \equiv \min_{w \in \mathcal{W}, \ell \in \mathcal{L}} V_{w\ell}$ and $0 \leq v_{pi} \leq n_p$,

$$u_p = \max_{w \in \mathcal{W}, \ell \in \mathcal{L}} \frac{v_{pw} - v_{p\ell} + n_p}{V_{w\ell}} \leq \max_{w \in \mathcal{W}, \ell \in \mathcal{L}} \frac{n_p - 0 + n_p}{V_{w\ell}} \leq \frac{2n_p}{V}.$$

Thus if we use $2n_p/V$ in lieu of u_p, we still get conservative results. (We also need to re-define U to be the sum of those upper bounds.) An intermediate, still conservative approach would be to use this upper bound for batches that consist of a single ballot, but use the sharper bound (6.4) when $n_p > 1$. Regardless, for the new definition of u_p and U, $\{T_j\}_{j=1}^n$ are IID, $\mathbb{P}\{T_j \le 1\} = 1$, and

$$\mathbb{E}T_1 = \sum_{p=1}^{P} \frac{u_p}{U} t_p = \frac{1}{U} \sum_{p=1}^{P} u_p \frac{e_p}{u_p} = \frac{1}{U} \sum_{p=1}^{P} e_p = E/U.$$

So, if we have evidence that $\mathbb{E}T_1 < \lambda_s/U$, we have evidence that $E < \lambda_s$.

Testing $\mathbb{E}T_1 \ge \lambda_s/U$

A variety of methods are available to test whether $\mathbb{E}T_1 < \lambda_s/U$. One particularly elegant sequential method is based on Wald's Sequential Probability Ratio Test (SPRT) [145]. Harold Kaplan pointed out this method on a website that no longer exists. A derivation of this *Kaplan-Wald* method is in Appendix A of [133]; to apply the method here, take $t = \lambda_s$ in their equation 18. A different sequential method, the *Kaplan-Markov* method (also due to Harold Kaplan), is given in [129].

6.5 Ballot-polling audits of overstatement quotas

To use the new stratification method with ballot polling requires a different approach than [63] took: their approach tests whether w got a larger *share* of the votes than ℓ, but we need to test whether the margin *in votes* in the stratum is greater than or equal to a threshold (namely, $\lambda_s V_{w\ell}$). This introduces a nuisance parameter, the number of ballots with votes for either w or ℓ. We address this by maximizing the probability ratio in Wald's SPRT [145] over all possible values of the nuisance parameter.

In this section, we derive a ballot-polling test of the hypothesis that the margin (in votes) in a single stratum is greater than or equal to a threshold c.

Wald's SPRT with a nuisance parameter

Consider a single stratum s containing N_s ballots, of which $N_{w,s}$ have a vote for w but not for ℓ, $N_{\ell,s}$ have a vote for ℓ but not for w, and $N_{u,s} = N_s - N_{w,s} - N_{\ell,s}$ have votes for both w and ℓ or neither w nor ℓ, including undervotes and invalid ballots. Ballots are drawn sequentially without replacement, with equal probability of selecting each as-yet-unselected ballot in each draw.

We want to test the compound hypothesis that $N_{w,s} - N_{\ell,s} \le c$ against the alternative that $N_{w,s} = V_{w,s}$, $N_{\ell,s} = V_{\ell,s}$, and $N_{u,s} = V_{u,s}$, with $V_{w,s} - V_{\ell,s} > c$.

The values $V_{w,s}$, $V_{\ell,s}$, and $V_{u,s}$ are the reported results for stratum s (or values related to those reported results; see [63]). The value of c is inferred from the definition $\omega_{w\ell,s} \equiv V_{w\ell,s} - (N_{w,s} - N_{\ell,s})$. Thus,

$$c = V_{w,s} - V_{\ell,s} - \omega_{w\ell,s} = V_{w\ell,s} - \lambda_s V_{w\ell}.$$

In this problem, $N_{u,s}$ (equivalently, $N_{w,s} + N_{\ell,s}$) is a nuisance parameter: we care about $N_{w,s} - N_{\ell,s}$, but the probability distribution of the sample depends also on $N_{u,s}$.

Let X_k be w, ℓ, or u according to whether the ballot selected on the kth draw shows a vote for w but not ℓ, ℓ but not w, or something else. Let $W_n \equiv \sum_{k=1}^{n} 1_{X_k=w}$; and define L_n and U_n analogously. Then $W_n + L_n + U_n = n$.

The probability of a given data sequence $X_1, \ldots, X_n$ under the alternative hypothesis is

$$\frac{\prod_{i=0}^{W_n-1}(V_{w,s} - i) \prod_{i=0}^{L_n-1}(V_{\ell,s} - i) \prod_{i=0}^{U_n-1}(V_{u,s} - i)}{\prod_{i=0}^{n-1}(N_s - i)}, \tag{6.5}$$

using the convention that $\prod_{i=0}^{-1} d_i \equiv 1$. If $L_n \geq W_n - cn/N_s$, the data obviously do not provide evidence against the null, so we suppose that $L_n < W_n - cn/N_s$, in which case, the element of the null that will maximize the probability of the observed data has $N_{w,s} - c = N_{\ell,s}$. Under the null hypothesis, the probability of $X_1, \ldots, X_n$ is

$$\frac{\prod_{i=0}^{W_n-1}(N_{w,s} - i) \prod_{i=0}^{L_n-1}(N_{w,s} - c - i) \prod_{i=0}^{U_n-1}(N_{u,s} - i)}{\prod_{i=0}^{n-1}(N_s - i)}, \tag{6.6}$$

for some value $N_{w,s}$ and the corresponding $N_{u,s} = N_s - 2N_{w,s} + c$. How large can that probability be under the null? The probability under the null is maximized by any integer $x \in \{\max(W_n, L_n + c), \ldots, (N_s - U_n)/2\}$ that maximizes

$$\prod_{i=0}^{W_n-1}(x - i) \prod_{i=0}^{L_n-1}(x - c - i) \prod_{i=0}^{U_n-1}(N_s - 2x + c - i),$$

again using the convention that $\prod_{i=0}^{-1} d_i \equiv 1$. The logarithm is monotonic, so any maximizer x^* also maximizes

$$f(x) = \sum_{i=0}^{W_n-1} \ln(x - i) + \sum_{i=0}^{L_n-1} \ln(x - c - i) + \sum_{i=0}^{U_n-1} \ln(N_s - 2x + c - i).$$

(Here, we use the convention that $\sum_{i=0}^{-1} d_i \equiv 0$.) If we consider the domain of f to be real numbers, not just integers, the second derivative of f is everywhere negative, so f has a unique maximum on $[\max(W_n, L_n + c), (N_s - U_n)/2]$, either at one of the endpoints or somewhere on the interval. The derivative $f'(x)$ gives information about where the maximum occurs:

$$f'(x) = \sum_{i=0}^{W_n-1} \frac{1}{x - i} + \sum_{i=0}^{L_n-1} \frac{1}{x - c - i} - 2 \sum_{i=0}^{U_n-1} \frac{1}{N_s - 2x + c - i}.$$

If $f'(x)$ does not change signs, then the maximum is at one of the endpoints; set x^* to be the endpoint that makes f larger. Otherwise, the (real-valued) maximum occurs where $f'(x) = 0$ and can be found using a root finder such as Brent's method; alternatively, the maximum can be found using concave optimization. Since the maximizer for the original problem must be an integer, set x^* to be the value in $\{\lfloor x \rfloor, \lceil x \rceil\}$ that makes f larger; by concavity, that is the integer maximizer.

A conservative p-value for the null hypothesis after n items have been drawn is thus

$$P_n = \frac{\prod_{i=0}^{W_n-1}(x^* - i)\ \prod_{i=0}^{L_n-1}(x^* - c - i)\ \prod_{i=0}^{U_n-1}(N_s - 2x^* + c - i)}{\prod_{i=0}^{W_n-1}(V_{w,s} - i)\ \prod_{i=0}^{L_n-1}(V_{\ell,s} - i)\ \prod_{i=0}^{U_n-1}(V_{u,s} - i)}.$$

Because the test is built on Wald's SPRT, the sample can expand sequentially and (if the null hypothesis is true) the chance that $P_n < p$ is never larger than p. That is, $\Pr\{\inf_n P_n \leq p\} \leq p$ if the null is true.

Code implementing this approach is given in `https://github.com/pbstark/CORLA18`.

6.6 Maximizing Fisher's combined p-value for a hybrid stratified audit

Below, we present an algorithm for finding the maximum p-value in a hybrid stratified audit that combines ballot-level comparison with ballot polling. Let $p_1(\lambda)$ be the p-value from a ballot-polling audit and $p_2(1-\lambda)$ be the p-value from a ballot-level comparison audit. The calculation of these p-values is derived in Appendix 6.4 and 6.5, respectively.

The algorithm to approximate $\min_\lambda \chi(\lambda)$ is as follows:

1. Pick a grid of values $\lambda_0 \equiv \lambda_- < \lambda_1 < \cdots < \lambda_n \equiv \lambda_+$.
2. Evaluate $\chi(\lambda_j)$, $j = 0, \ldots, n$.
3. If, for any j, $\chi(\lambda_j) \leq \chi_\alpha$, the audit must escalate.
4. Otherwise, check whether $\chi_n^* \geq \chi_\alpha$.
5. If so, the audit can stop: the minimum $\chi(\lambda)$ is above χ_α. If not, refine the grid of values of λ and return to step 2.

Bounding the local modulus of continuity for Fisher's combining function

Fisher's combining function applied to the stratumwise p-values is

$$\chi(\lambda) = -2\Bigg[n_1 \log\left(1 - \frac{\lambda V_{w\ell}}{2N_1\gamma}\right) - o_1 \log(1 - 1/2\gamma) - o_2 \log(1 - 1/\gamma)$$
$$- u_1 \log(1 + 1/2\gamma) - u_2 \log(1 + 1/\gamma) + \sum_{i=0}^{W_n - 1} \log(N_{w,2}(1-\lambda) - i)$$
$$+ \sum_{i=0}^{L_n - 1} \log(N_{w,2}(1-\lambda) - c(1-\lambda) - i) + \sum_{i=0}^{U_n - 1} \log(N_2 - 2N_{w,2}(1-\lambda) + c(1-\lambda) - i)$$
$$- \sum_{i=0}^{W_n - 1} \log(V_{w,2} - i) - \sum_{i=0}^{L_n - 1} \log(V_{\ell,2} - i) - \sum_{i=0}^{U_n - 1} \log(V_{u,2} - i)\Bigg]. \tag{6.7}$$

We seek the modulus of continuity (or an upper bound on it) for this function. While $c(\lambda)$ and $N_w(\lambda)$ must be integer values (as they represent numbers of ballots), the p-value from the ballot-polling SPRT and thus $\chi(\lambda)$ make sense for non-integer values of $c(\lambda)$ and $N_w(\lambda)$, so to allow the use of calculus, we will extend the domain to real values.

Moduli of continuity obey two key properties:

- **Sublinearity**: $\omega(af + bg, \delta) \leq |a|\omega_f(\delta) + |b|\omega_g(\delta)$
- **Composition**: $\omega(g \circ f, \delta) = \omega_g(\omega_f(\delta))$

First, we find the modulus of continuity for several simple functions:

- $\log(ax - b)$: the modulus of continuity is $\log(1 + a\delta)$.
- $c(\lambda) = (V_{w,2} - V_{\ell,2}) - \lambda V_{w\ell}$: since this function is linear in λ, the modulus of continuity is $V_{w\ell}\delta$.
- $N_{w,2}(\lambda)$. When $N_{u,2}$ is known, this is also linear in $c(\lambda)$. When $N_{u,2}$ is not known, $N_{w,2}(\lambda)$ increases with c, at a rate between 0 and 1. For $N_{w,2}(\lambda)$ in the interior of the domain, taking the implicit derivative with respect to c shows that

$$\frac{dN_{w,2}}{dc} = \frac{\sum_{i=0}^{L_n-1} \frac{1}{(N_{w,2}-c-i)^2} + \sum_{i=0}^{U_n-1} \frac{2}{(N_2-2N_{w,2}+c-i)^2}}{\sum_{i=0}^{W_n-1} \frac{1}{(N_{w,2}-i)^2} + \sum_{i=0}^{L_n-1} \frac{1}{(N_{w,2}-c-i)^2} + \sum_{i=0}^{U_n-1} \frac{4}{(N_2-2N_{w,2}+c-i)^2}} \in [0, 1].$$

  Thus it also has modulus of continuity at most $V_{w\ell}\delta$. Moreover, since $N_{w,2}$ is dominated by c, the terms $N_{w,2}(\lambda) - c(\lambda)$ and $-2N_{w,2}(\lambda) + c(\lambda)$ have modulus of continuity $V_{w\ell}\delta$ and $2V_{w\ell}\delta$, respectively.
- For terms that don't involve λ, the modulus of continuity is 0.

There are four terms in $\chi(\lambda)$ that involve λ. Using the two key properties of linearity and composition, as well as the moduli for simple functions above, we obtain the following bound on the global modulus of continuity:

$$\begin{aligned}\omega_\chi(\delta) =& 2W_n \log(1 + V_{w\ell}\delta) + 2L_n \log(1 + V_{w\ell}\delta) + \\ & 2U_n \log(1 + 2V_{w\ell}\delta) + 2n_1 \log\left(1 + \frac{V_{w\ell}}{2N_1\gamma}\delta\right)\end{aligned} \tag{6.8}$$

6.7 Numerical examples

Jupyter notebooks containing calculations for hybrid stratified audits intended to be relevant for Colorado are available at `https://www.github.com/pbstark/CORLA18`.

`hybrid-audit-example-1` contains two hypothetical elections. The first has 110,000 cast ballots, of which 9.1% were in no-CVR counties. The *diluted margin* (the margin in votes, divided by the total number of ballots cast) is 1.8%. In 94% of 10,000 simulations in which the reported results were correct, drawing 700 ballots from the CVR stratum and 500 ballots from the no-CVR stratum (1,200 ballots in all) allowed SUITE to confirm the outcome at 10% risk. For the remaining 6%, further expansion of the audits would have been necessary.

If it were possible to conduct a ballot-level comparison audit for the entire contest, an RLA with risk limit 10% could terminate after examining 263 ballots if it found no errors. A ballot-polling audit of the entire contest would have been expected to examine about 14,000 ballots, more than 10% of ballots cast. The hybrid audit is less efficient than a ballot-level comparison audit, but far more efficient than a ballot-polling audit.

The second hypothetical election has 2 million cast ballots, of which 5% were cast in no-CVR counties. The diluted margin is about 20%. The workload for SUITE at 5% risk is quite low: In 93% of 10,000 simulations in which the reported results were correct, auditing 50 ballots from the CVR stratum and 25 ballots from the no-CVR stratum would have confirmed the outcome. If it were possible to conduct a ballot-level comparison audit for the entire contest, an RLA at risk limit 5% could terminate after examining 31 ballots if it found no errors. The additional work for the hybrid stratified audit is disproportionately in the no-CVR counties.

A second notebook, `hybrid-audit-example-2`, illustrates the workflow for SUITE for an election with 2 million ballots cast. The reported margin is just over 1%, but the reported winner and reported loser are actually tied in both strata. The risk limit is 5%. For a sample of 7600 ballots from the CVR stratum and 400 ballots from the no-CVR stratum, the maximum combined p-value is 1, so the audit cannot stop there.

A third notebook, `fisher_combined_pvalue`, illustrates the numerical methods used to check whether the maximum combined p-value is below the risk limit. It includes code for the tests in the two strata, for the lower and upper bounds λ_- and λ_+ for λ, for evaluating Fisher's

combining function on a grid, and for maximizing the p-value, using the local modulus of continuity for Equation 6.7 to bound any approximation error.

6.8 Discussion

We present SUITE, a new class of procedures for RLAs based on stratified random sampling. SUITE is agnostic about the capability of voting equipment in different strata, unlike previous methods, which require batch-level comparisons in every stratum. SUITE allows arbitrary tests to be used in different strata; if those tests are sequentially valid, then the overall RLA is sequential. (Otherwise, multiplicity adjustments might be needed if one wants an audit that escalates in stages. See [128, 127] for two approaches.)

Like other RLA methods, SUITE poses auditing as a hypothesis test. The null hypothesis is a union over all partitions of outcome-changing error across strata. The hypothesis is rejected if the maximum p-value over all such partitions is sufficiently small. Each possible partition yields an intersection hypothesis, tested by combining p-values from different strata using Fisher's combining function (or a suitable replacement).

Among other things, the new approach solves a current problem in Colorado: how to conduct RLAs of contests that cross jurisdictional lines, such as statewide contests and many federal contests.

We give numerical examples in Jupyter notebooks that can be modified to estimate the workload for different contest sizes, margins, and risk limits. In our numerical experiments, the new method requires auditing far fewer ballots than previous approaches would.

Chapter 7

Challenges of using SUITE in practice

MCL 168.319 authorizes the Michigan Secretary of State's office to develop a post-election audit procedure. Current procedures include performance audits at the local and county level, to check whether state regulations were followed properly before, during, and after the election, as well as a full manual recount of randomly selected precincts throughout the state [100]. However, these audits are conducted after the results have been certified, making it impossible to use any evidence of discrepancies to amend an incorrect outcome.

Moreover, they do not provide evidence that the reported outcome was determined correctly. RLAs present an opportunity to identify cases of fraud or errors where voting machines interpreted voter intent incorrectly, where voters mismarked ballots, or where ballots simply weren't tabulated, if those problems altered the outcome. For instance, a random audit of 3% of scanners in New York county on Election Day 2018 revealed small discrepancies in the number of ballots scanned and the number of ballots stored in physical batches, and concluded that each of these discrepancies were due to mishandling by the poll workers [59]. Human errors are inevitable; RLAs can provide statistical evidence that the magnitude of errors was not sufficiently large to change the election outcome.

Given the evidence of Russian hacking in the 2016 elections, the Michigan Bureau of Elections decided to pilot RLAs as additional layer of security for Michigan's elections that provides additional assurance that ballots were tallied correctly while reducing the need for full manual recounts. Rochester Hills, Lansing, and Kalamazoo were invited to pilot post-election audits to explore the feasibility and the challenges of conducting RLAs.

The three cities used different voting technologies in the 2018 general election. Rochester Hills used voting equipment that did not produce cast vote records (CVRs), making ballot or batch polling the only RLA strategy they could use. Lansing and Kalamazoo used two different types of voting equipment that produced (differently formatted) CVRs, but also had a sizable fraction of mail-in ballots with no CVRs. This presented an opportunity to use SUITE, which can handle all three of these situations. I was invited to attend the pilots as an expert in risk-limiting audits and the software developer for the SUITE tool. The pilots took place December 3-5, 2018.

Pilot RLAs have been crucial for election officials to learn how to conduct RLAs, under-

standing which approaches are practical given the available resources, and for recommending improvements. California hosted the earliest pilot RLAs in 2008 [40]. Between 2011 and 2013, fourteen RLAs were piloted in eleven California counties. The Secretary of State's office documented its best practices, costs and timing, and recommendations for modifications to existing voting systems [87]. They found that the cost was minimal compared to California's 1% manual tally law. In November, 2017, Colorado became the first state to conduct statewide risk-limiting audits. The Denver County Auditor evaluated the Colorado risk-limiting audit process and documented its recommendations for improvement, including creating more streamlined software for transmitting information between local and state-level entities and a better sampling process for handling multi-page ballots [86]. Both the November, 2018 pilot RLAs in the city of Fairfax, Virginia and in Orange County, California highlighted the importance of upgrading election hardware to newer voting systems and hardware [61, 119].

Following [40], this chapter seeks to answer the following questions about implementing SUITE in practice:

- What RLA methods are practical to use currently, and what resources are required to use them?
- What steps of the RLA are challenging?
- What processes at the local level in Michigan need to be changed to use RLAs, and in particular, to use SUITE?

7.1 Software

We began writing a software tool for SUITE[1] in October, 2018. This was intended to be a prototype user interface for these pilot RLAs rather than an industrial strength tool; the hope is that software developers will be able to base a proper program on our prototype. We chose to use a Jupyter notebook because the code for risk calculations was already written in Python and notebooks provide a user interface more similar to a website than pure Python code in the terminal would.

The SUITE tool calculates the necessary pieces of information for each step of the audit:

- It estimates the initial sample sizes needed in each stratum to stop the audit, assuming that the reported results were correct and that one-vote overstatements occur at a very low rate in the ballot comparison stratum.
- It uses the SHA256 PRNG from the `cryptorandom` package to draw the random samples of ballots in each stratum.

[1] The tool can be run interactively at `https://mybinder.org/v2/gh/pbstark/CORLA18/master?filepath=code%2Fsuite_toolkit.ipynb`.

- It reads in a ballot manifest for each stratum as a separate CSV file, then determines which ballots in which batch need to be pulled based on the samples that were drawn previously, and finally exports this data to a CSV file to be printed.
- It takes the statistics from each sample (the number of 1-vote and 2-vote overstatements and understatements in the comparison stratum, and the number of votes seen for each candidate in the polling stratum) as input and runs the risk calculation for each pair of reported winners and reported losers.
- If not all (winner, loser) pairs are confirmed, it estimates how many more ballots need to be sampled in each stratum to confirm the reported outcome, assuming that the rates of discrepancies and the rates of votes for each candidate will continue to reflect those seen in the initial samples.
- It runs each of these steps again for a second round of sampling.
- It logs each step of the process in JSON files.

I projected the SUITE tool on a large screen for the audience to watch at each pilot. The Jupyter notebook is available on GitHub and was set up to run in the cloud, but we decided to run it locally on my laptop to save the time needed to train others on how to use the tool.

7.2 Michigan pilot RLAs

In coordination with the Bureau of Elections at the Michigan Secretary of State's office, we performed three pilots of risk-limiting audit procedures. This section describes the audits in each city and the differences between them. Table 7.2 summarizes the sizes of each audit.

Since the goal of these pilots was to gain hands-on experience, instruct local election officials, compare procedures, and identify bottlenecks in the processes, we opted to reduce the population of ballots under audit to each city. The sampling frame for a true RLA includes all ballots cast for a particular contest; none of the contests under audit were entirely contained in the sampling frame of ballots. Each audit *pretended* that the ballots we had access to comprised the entire contest. Nonetheless, reducing the sampling frame made these pilots feasible to conduct in a day and illustrated RLA procedures to a wide audience.

Rochester Hills

The Rochester Hills pilot RLA took place on December 3. About 30 local election officials from around the state attended to observe the audit process. We audited Proposal 18-3, a statewide proposal to add new voting policies to the Michigan constitution. In Rochester Hills, "Yes" votes won with a 29% margin.

City	Total ballots	Winner	Loser	Margin	% Ballots with a CVR	# Ballots audited	% Ballots audited
Rochester Hills	36,666	22,999	12,343	29%	0%	76	0.2%
Lansing	21,328	10,309	7,694	12%	50%	260	1.2%
Kalamazoo	27,666	20,699	5,569	55%	19%	40	1.4%

Table 7.1: Summary of the races audited. Ballots and votes for the candidates are the results as reported. Margins are expressed as a percentage of total ballots cast. The ballots included in the contest under audit in Lansing were restricted for the purpose of the pilot; half of Election Day precincts were omitted.

Rochester Hills uses Hart InterCivic Verity tabulators, which do not provide CVRs. Thus, we used entirely ballot polling for the audit. We did not need to use different software for this: The SUITE tool reverts to pure ballot polling when the CVR stratum size is 0 ballots.

The SUITE tool estimated that we'd need to look at 76 ballots to achieve a risk limit of 5% if the reported results were correct. We had 5 two-person audit boards pull batches from ballot bags and sample from them one of two ways: by counting down to the ballot that the SUITE tool sampled (e.g. retrieving the twenty-first ballot in Election Day Batch 3 by counting ballots from the top of the stack) or by using an approximate sampling technique, k-cut, to select a ballot from the stack. k-cut involves pulling the batch chosen by the SUITE tool, then "cutting" the stack of ballots like a deck of cards several times and choosing the ballot on top to be in the sample [121]. This procedure samples ballots approximately uniformly from a batch.

Audit boards used a mix of counting and k-cut to sample ballots: if the SUITE tool sampled a ballot that would require counting more than 200 ballots to find it, the team used k-cut instead. It took about 2.5 hours for the audit boards to pull the ballots and record them, and another half hour for a separate team to tally them.

The sample contained 50 "yes" votes and 26 "no" votes. The risk based on the sample was 2%.

Lansing

The Lansing pilot RLA took place on December 4. We audited the 54-A District Court Judge race using a 9% risk limit. Cynthia Ward won over Ayanna Neal by about a 10% margin. Though this contest was contained within the city of Lansing, we reduced the sampling frame of ballots for other reasons, making this a pilot of RLA procedures rather than a true RLA.

First, there were several ballot bags that legally could not be opened. Precinct 45 was one of the randomly selected precincts to do a full hand recount for the post-election audit procedures developed under MCL 168.319, so the ballot bags containing ballots from Precinct 45 could not be opened. These bags contained ballots from other precincts as well. One way to handle this would have been the "phantoms to zombies" approach, treating each

inaccessible ballot in the sample as a vote for the reported loser [5]. Because the point of the pilot was to illustrate procedures rather than to obtain a precise risk measurement, we opted to remove ballots from the ballot bags containing Precinct 45 from the sampling frame.

Second, the SUITE tool estimated that the initial sample size needed would be over 500. Based on observing the sampling in Rochester Hills the previous day, we knew that it would not be possible to examine 500 ballots before the close of business day. To reduce the sample size needed, we reduced the sampling frame: I removed about 30 Election Day precincts from the ballot manifest and the reported vote totals.[2] This changed the fraction of ballots with a CVR from 23.6% to 50%, making the hybrid SUITE audit much more efficient (see Section 7.4). This did not substantially affect the reported margin between Ward and Neal amongst Election Day votes. The initial sample size estimate decreased to 260, with 130 ballots from the absentee votes and 130 from Election Day votes.

Lansing used 6 two-member audit boards. It took about 4 hours to pull and record all the ballots, then another hour to compare absentee, vote-by-mail ballots to the CVR and tally the Election Day sample. The audit boards used counting down to sample vote-by-mail ballots, because these ballots were linked to their corresponding CVR by position: the batches of vote-by-mail ballots and the CVR file were both supposed to contain ballots in the order in which they were scanned. They used a mix of counting and k-cut to sample Election Day ballots.

Among the 130 sampled vote-by-mail ballots, we identified 15 discrepancies between the paper ballots and the CVR. It is unlikely that these were true discrepancies; we believe that the sampled paper ballots were matched to the incorrect CVR because of counting errors in sampling the ballots or the way the batches were stored. We discuss this issue further below. With so many vote-by-mail discrepancies, the SUITE risk was 100%.

For comparison, we pooled the vote-by-mail and Election Day samples for a pure ballot polling audit. One vote-by-mail ballot was sampled three times, reducing the overall ballot polling sample size to 258 ballots. In total, there were 116 votes for Ward, 94 votes for Neal, and 48 invalid ballots or write-ins. The initial sample size estimate for ballot polling was 285 ballots, just higher than the sample size actually drawn. Overall, the risk calculated using SUITE ballot polling was 87%.[3] Given the vote proportions observed in the sample, the SUITE tool estimated that we'd need to draw 4,557 more ballots in a second round of sampling to achieve a 9% risk limit.

Kalamazoo

The Kalamazoo pilot RLA took place on December 5. We audited the governor's race, pretending that the governor's race was entirely contained in Kalamazoo city. In this contest,

[2] Another approach could have been to draw a fixed number of ballots, say 100, from the entire population and simply measure the attained risk of the sample. This approach would not have attained the desired 9% risk limit, but would have been equally illustrative.

[3] The risk calculation using BRAVO ballot polling is 38%. This difference likely occurs because there were so many votes for neither Ward nor Neal in the sample.

Gretchen Whitmer beat runner-up Bill Schuette by a margin of 54%.

The SUITE tool estimated that we would need to pull 33 ballots to achieve a 5% risk limit. Based on our experiences the previous two days, I knew this would not take long. To ensure that we would meet the risk limit and to give each audit board extra hands-on experience, I increased the total sample size to 40 ballots, 8 from the vote-by-mail ballots and 32 from Election Day. It took 4 two-member audit boards about 1.5 hours to both sample and tally the ballots.

The vote-by-mail ballots were imprinted with ID numbers, so the comparison audit involved looking up the imprinted ID in the CVR file. They did not rely on preserving the order of ballots as in Lansing, so a mix of counting and k-cut were used to sample both vote-by-mail ballots and Election Day ballots.

There were no discrepancies in the vote-by-mail sample. In the Election Day sample, there were 23 ballots for Whitmer, 8 ballots for Schuette, and 1 for Gelineau. The risk between Whitmer and Schuette was 3.7%, while the risk between Whitmer and the remaining runner-ups was below 1%.

7.3 Sampling ballots

The SUITE tool, as well as other existing RLA tools, draws random samples of ballots to be audited with and without replacement, and identifies their position within batches of ballots. The audit boards must then retrieve the batch and count down into the stack of ballots to take out the sampled ballot. This is feasible when batches are small, but quickly becomes problematic as batch sizes grow. In Rochester Hills, for instance, vote-by-mail ballots are split into batches of 100 when they are put in the tabulators. On the other hand, Election Day ballots are organized by precinct, resulting in batches of up to 1035 ballots. In Kalamazoo, the average Election Day batch contained 829 ballots and the largest batch contained 1578 ballots.

For large batches of ballots, we used k-cut in place of counting down to a high number of ballots. k-cut is an approximate method of drawing simple random samples from physical batches of paper. The idea is to "cut" the batch, as one would cut a deck of cards, choosing where to cut uniformly at random. After the batch is cut several times, the item on top of the stack is included in the sample. Based on preliminary results, 6 cuts is sufficiently many to ensure that which item appears on top is indistinguishable from true uniform random selection [121].

k-cut improves the speed of pulling ballots. I observed audit boards sampling ballots by counting and using k-cut in each city. In Rochester Hills, I observed an audit board doing both. It took 20 seconds to count a stack of 10 ballots, because after the first person counted the second one double checked. I watched the same team do k-cut and it took them 75 seconds to perform 6 cuts. In Kalamazoo, I watched one team count 77 ballots in 2 minutes. I timed another team using k-cut on a stack of over 1000 ballots to pull 2 ballots using 6 cuts: it took them 2 minutes, 5 seconds for the first ballot and 2 minutes, 22 seconds for the

second. Based on these numbers I would estimate that a faster audit board would take the same amount of time to count 75 ballots or to do 6 cuts. For a slower audit board, counting 60 ballots would take the same amount of time as doing 6 cuts. In either case, using k-cut to select ballots is faster when the audit board would have to count more than 75 ballots.

However, k-cut has implementation challenges as well. The card-cutting analogy does not hold for Michigan's ballots: they are large and difficult to physically handle unlike a deck of playing cards. Michigan's ballots are 22 inches long, so batches were too long to hold lengthwise and too floppy to hold by the middle. Some audit boards realized that they could have one person on each side of the table and move the ballots using four hands. The large batches were unwieldy and heavy; many people serving on audit boards are older women, so this was a problem.

Moreover, k-cut is only approximately drawing a simple random sample. The proofs of its approximate uniformity depend on the empirical distribution of cut sizes, which might vary by person, by batch size, by the duration of the audit, or any other unmeasured factors. We attempted to improve audit boards' cut size uniformity by providing guidance from pseudo-random number generators. Each team that used k-cut used an online pseudo-random number generator to obtain six numbers uniformly chosen between 1 and 99. This number was intended to give guidance on what percentage of the batch to cut, for each of the six cuts. While it is unlikely that the teams actually used these hints perfectly, it may have improved systematic bias in choosing cut sizes. Even so, a few audit boards preferred to count anyway; they trusted the randomness provided by the RLA tool's random sampler more than the approximation by k-cut.

7.4 Using comparison audits in hybrid SUITE

Ballot-level comparison audits are the most statistically efficient type of RLA: the number of ballots needed to end the audit, if the reported margin is correct, decreases inversely with the margin. (In comparison, the average sample size needed for ballot polling decreases inversely with the *square* of the margin.) It should be advantageous to use ballot-level comparisons when possible [62].

We ran into three issues with the comparison audit portion of hybrid SUITE: interpreting the CVR files and doing the comparisons, handling paper ballots so they can be linked to the CVR, and SUITE's efficiency as a function of the size of the comparison stratum.

CVR Files

Ballot-level CVRs are necessary to perform ballot-level comparison audits. Comparison audits were not possible in Rochester Hills, because the Hart voting system they use does not make ballot-level data available in a way that can be matched to the corresponding

paper ballot. The Dominion and ES&S voting systems, used in Lansing and Kalamazoo, respectively, make this data available. However, the files are unwieldy.[4]

Dominion's CVR files are written in JSON and use codes for each contest and candidate that aren't human readable. Lansing's CVR was 40 MB and we had issues converting it to a readable CSV file. Instead, we wrote a Python script on the fly to look up ballots by position in batch. The ES&S CVR in Kalamazoo was made up of two CSV files containing the unique ID imprinted on each ballot. Looking up ballots in this format was simpler.

For future comparison audits, it would be helpful to have automated software to parse these CVRs, look up ballots in them, and determine whether there was a discrepancy. NIST is leading an effort to create a common CVR format,[5] but there is no indication that it will be adopted soon. Moreover, there must be an automated way to determine whether a discrepancy is an overstatement, an understatement, or neither. Even the team of RLA experts at these pilots had a difficult time determining what type a discrepancy was. An incorrect determination could undermine the audit or prolong it unnecessarily.

Paper handling

To perform a comparison audit, it is crucial that paper ballots can be matched to their corresponding CVR. Lansing and Kalamazoo took different approaches to matching paper and electronic records. In Kalamazoo, the ES&S tabulator imprinted a unique identifier on their vote-by-mail ballots once they were scanned. This unique identifier was recorded in the CVR, allowing us to pull a ballot and look up its ID in the CVR file. In Lansing, the Dominion tabulator did not imprint identifiers on ballots. Instead, they relied on the position of the ballot in the batch to match paper to electronic records: the CVR file preserved the order in which ballots were tabulated. Both approaches have issues.

In Lansing, among the 130 sampled vote-by-mail ballots, we identified 15 discrepancies. We are confident that paper handling issues were the culprit because we examined more contests on the ballots with discrepancies: several other contests on those ballots showed discrepancies between the paper ballot and CVR. A number of errors due to paper handling could have caused this. First, if the audit boards counted ballots incorrectly and pulled the wrong ballot, there's no way of matching the CVR to the correct ballot. For instance, an off-by-one counting error could cause us to compare a paper ballot to the wrong CVR. Second, the city clerk reported that he saw some audit boards flip the batches upside-down, thinking that because the top ballot was upside-down that the whole batch was. This would certainly affect the order of ballots. Third, the poll workers were told to preserve the order of the vote-by-mail ballots as they were scanned, but we have no way of knowing whether they attempted to do this. It's possible that some batches were entirely out of order. Given the reported vote totals in the vote-by-mail stratum, there was approximately a 42% chance that a random CVR would match a sampled ballot for this contest. Nearly half of CVRs

[4] Colorado has worked with Dominion Voting to create CVRs in a standardized, human-readable format. This is something that the vendors can improve, but have not yet done so in most jurisdictions.

[5] See https://github.com/usnistgov/CastVoteRecords, last accessed December 17, 2018.

that were matched to sampled ballots could have been the wrong CVR, but went undetected because we only checked for discrepancies in a single contest.

The Kalamazoo comparison audit found no discrepancies among the 8 sampled vote-by-mail ballots. The audit boards pulled the 8 sampled ballots and used their unique identifiers to look up the corresponding CVRs. This process was efficient and more accurate than the approach used in Lansing. However, the reliance on ballot identifiers opens the audit to manipulation.

For instance, a malicious imprinter could skew the election by changing CVRs to reflect a desired outcome and modifying imprinted identifiers on ballots, and escape detection by this method of sampling ballots. Suppose that the number of paper ballots is equal to the number of CVRs. The adversary can change a fraction of CVRs to skew the election, then, print identifiers on the paper ballots to match the unchanged CVRs. Some identifiers will be duplicated in the paper ballots, and it is unlikely that auditing a small fraction of ballots would reveal duplicate IDs. The CVRs lead to the adversarially-desired election outcome. The ballot sampling procedure used in Kalamazoo would not detect this: by sampling ballots first and using their imprinted IDs to look in the CVR, one will never examine ballots in the portion of the CVR that has been changed.

Another approach to sampling with the help of imprinted identifiers would be to sample from the CVR, retrieve the paper ballot with the corresponding identifier. This ensures that all CVRs have an equal probability of appearing in the sample. However, this approach is vulnerable to manipulation of the paper ballots. To conduct a rigorous audit, upper bounds on the number of ballots should be used in drawing the sample and any ballots that can't be located should be treated with the "phantoms to zombies" approach [5].

This would have been difficult to implement in Kalamazoo: the vote-by-mail ballots were stored in batches of size up to 355 and were not kept in the order in which they were scanned. Locating the ballot with a particular identifier among large batches would have likely been more time consuming than counting ballots. Moreover, without a test to ensure that identifiers are unique, there is no guarantee that the retrieved ballot is the right one.

A better procedure would be to sample ballots from the manifest, then retrieve the ballot *and* CVR according to their position in the batch. This prevents manipulated imprinted IDs from biasing the sample. The "phantoms to zombies" method can be used to deal with any ballot without a corresponding CVR [5]. The imprinted IDs can serve as a check that the correct ballot and CVR are matched, and further provide evidence that the ballot manifest is correct. This sampling approach could not have been used in Kalamazoo for this election because they did not preserve the scanning order of their batches.

SUITE efficiency

SUITE was developed with Colorado in mind: as of June 2018, an estimated 98.2% of voters were in counties with CVRs. In Michigan, most voting equipment does not produce a CVR. The machines that do are used to tabulate mail-in ballots. Only a small fraction of the population is eligible to vote absentee: voters must be 60 years or older, unable

to vote without assistance in the polling place, or in another uncommon circumstance. In Kalamazoo, only 19% of the ballots audited were absentee votes and had a CVR. In Lansing, before reducing the universe of cast ballots, 23.6% of ballots were absentee votes.

Figure 7.1 shows the estimated number of ballots needed to audit a contest of approximately 13.7 million ballots. Simulations were based on actual ballot counts from California's 2008 election [16]. If 98% of ballots had CVRs, as in Colorado, and SUITE could have been used to conduct a hybrid audit, the audit workload would have been slightly higher than a pure ballot-level comparison audit. This is a substantial improvement over ballot polling: for a 5% margin, ballot polling would require ten times more ballots than SUITE. However, if only 25% of ballots had a CVR and a SUITE hybrid audit were done, the workload would have been *more* than doing a pure ballot-polling audit.

Figure 7.1: Estimated workload (sample size) for an RLA of 13.7 million ballots with a 5% risk limit. The sample size is plotted on a logarithmic scale.

Ballot polling alone is more efficient than SUITE when the fraction of ballots in the CVR stratum is low. Simulations assuming that the overall margin between the reported winner and loser is the same in both strata show that when the fraction of ballots with a CVR is low, SUITE can be substantially worse. In Lansing, the reduced universe of cast ballots contained 50% vote-by-mail ballots. If we had used ballot polling alone, the estimated initial

sample size would have been 300 ballots, while the estimated initial sample size for SUITE was 260. If the fraction of ballots had been below 40%, then ballot polling would have been more efficient. The improvement due to combining comparison and polling audits is not substantial. In Kalamazoo, the estimated initial sample size for SUITE was 33 ballots, while the estimated initial sample size for BRAVO would have been 25 ballots. Kalamazoo would have needed at least 70% of ballots to come from vote-by-mail ballots for SUITE to be more efficient than BRAVO. Without a change in absentee voting policies or upgrades to voting equipment, it would be more efficient for Michigan to use BRAVO ballot polling audits.

7.5 Conclusions

The Michigan pilot RLAs demonstrated to local election officials that RLAs are feasible and will save work, compared to the current practice of conducting full hand recounts of randomly selected precincts across the state. Rochester Hills and Kalamazoo completed their RLA procedures in several hours and met the 5% risk limit. Lansing did not meet its 9% risk limit due to discrepancies in the comparison audit portion; however, it is unlikely that they would have completed an RLA in a single day because the margin in the target contest was so small.

The ballot polling portions of the pilots went smoothly. k-cut enabled the audit boards to sample ballots efficiently from large batches, potentially saving hours relative to counting. Moreover, k-cut makes sense to use for ballot polling: since ballots needn't be matched to a CVR, it is less important to retrieve "the right" ballot than it is for a comparison audit.

The comparison portions of the pilots faced issues. The CVR files were difficult to process, and it will be a challenge for cities without imprinted ballots to match ballots to their corresponding CVR. Moreover, SUITE is less efficient than ballot polling alone in scenarios common across Michigan. Compared to Colorado, which has a primarily mail-in election, Michigan only allows a small number of residents to vote by mail. SUITE is most efficient when a large fraction of ballots can be used in the comparison audit. Without changes to their policies to increase the share of absentee voters or changes to their voting hardware to capture single-ballot CVRs from Election Day voters, SUITE will not be an efficient RLA option for Michigan. Ballot polling appears to be the best option.

While we identified RLA processes that worked for each city, a true RLA must encompass ballots from the entire contest under audit. Michigan's elections are highly decentralized: 1,520 city and township clerks administer absentee mail-in voting and Election Day voting. Participating in an audit outside the scope of their jurisdiction is a conceptual leap: local election officials are accustomed to thinking about the processes in their jurisdiction. Counties are free to choose to use Hart, Dominion, or ES&S voting equipment, making it difficult to standardize processes across the state. RLAs for countywide and statewide contests will be difficult to conduct without a more streamlined election process throughout Michigan.

Chapter 8

Bernoulli ballot polling: A manifest improvement for risk-limiting audits

8.1 Introduction

In this chapter, we present an RLA method based on *Bernoulli random sampling*. Traditional ballot-polling RLAs, discussed in Chapter 1, use simple random samples of ballots. With simple random sampling, the number of ballots to sample is fixed; with Bernoulli sampling, the *expected sampling rate* is fixed but the sample size is not. Conceptually, *Bernoulli ballot polling* (BBP) decides whether to include the jth ballot in the sample by tossing a biased coin that has probability p of landing heads. The ballot is included if and only if the coin lands heads. Coin tosses for different ballots are independent, but have the same chance of landing heads. (Rather than toss a coin for each ballot, it more efficient to implement Bernoulli sampling in practice using *geometric skipping*, described in Section 8.5.)

The logistical simplicity of Bernoulli sampling may make it useful for election audits. Like all RLAs, BBP RLAs require a voter-verifiable paper record. Like other ballot-polling RLAs [63, 62], BBP makes no other technical demands on the voting system. It requires no special equipment, and only a minimal amount of software to select and analyze the sample—in principle, it could be carried out with dice and a pencil and paper. In contrast to extant ballot-polling RLAs, BBP does *not* require a ballot manifest (although it does require knowing where all the ballots are, and access to the ballots). BBP is inherently local and parallelizable, because the decision of whether to include any particular ballot in the sample does not depend on which other ballots are selected, nor on how many other ballots have been selected, nor even on how many ballots were cast. We shall see that this has practical advantages.

Bernoulli sampling is well-known in the survey sampling literature, but it is used less often than simple random sampling for a number of reasons. The variance of estimates based on Bernoulli samples tends to be larger than for simple random samples [116], due to the fact that both the sample and the sample size are random. This added randomness complicates

rigorous inferences. A common estimator of the population mean from a Bernoulli sample is the Horvitz-Thompson estimator, which has a high variance when the sampling rate p is small. Often, p-values and confidence intervals for the Horvitz-Thompson estimator are approximated using the normal distribution [65, 23, 138], which may be inaccurate if the population distribution is skewed—as it often is in auditing problems [89].

Instead of relying on parametric approximations, we use a test based on Wald's sequential probability ratio test [145]. The test is akin to that in extant ballot polling RLA methods [63, 62], but the mathematics are modified to work with Bernoulli random samples, including the fact that Bernoulli samples are drawn without replacement. (Previous ballot-polling RLAs relied on sampling with replacement.) Conditional on the attained sample size n, a Bernoulli sample of ballots is a simple random sample. We maximize the conditional p-value of the null hypothesis (that the reported winner did not win) over a nuisance parameter, the total number of ballots with valid votes for either of a given pair of candidates, excluding invalid ballots or ballots for other candidates. A martingale argument shows that the resulting test is sequential: if the test does not reject, the sample can be expanded using additional rounds of Bernoulli sampling (with the same or different expected sampling rates) and the resulting p-values will still be conservative.

A BBP RLA can begin in polling places on election night. Given an initial sampling rate to be used across all precincts and vote centers, poll workers in each location determine which ballots will be examined in the audit, independently from each other and independently across ballots, and record the votes cast on each ballot selected. (Vote-by-mail and provisional ballots can be audited similarly; see Section 8.5.) Once the election results are reported, the sequential probability ratio test can be applied to the sample vote tallies to determine whether there is sufficient evidence that the reported outcome is correct.[1] If the sample does not provide sufficiently strong evidence to attain the risk limit, the sample can be expanded using subsequent rounds of Bernoulli sample until either the risk limit is attained or all ballots are inspected. Figure 8.1 summarizes the procedure.

BBP has a number of practical advantages, with little additional workload in terms of the number of ballots examined. Workload simulations show that the number of ballots needed to confirm a correctly reported outcome is similar for BBP and the BRAVO RLA [63]. If the choice of initial sampling rate (and thus, the initial sample size) is larger than necessary, the added efficiency of conducting the audit "in parallel" across the entire election may outweigh the cost of examining extra ballots. Using statewide results from the 2016 United States presidential election, BBP with a 1% initial sampling rate would have had at least a 99% chance of confirming the results in 42 states (assuming the reported results were in fact correct). A Python implementation of BBP is available at `https://github.com/pbstark/BernoulliBallotPolling`.

[1] The current method uses the reported results to construct the alternative hypothesis. A variant of the method does not require the reported results. We do not present that method here; it is related to ClipAudit [109].

Procedure for a Bernoulli ballot-polling audit

1. **Set initial sampling rate.** Choose initial sampling rate p_0 based on pre-election polls or set at a fixed value. If p_0 is selected based on an estimated margin, use the ASN heuristic in Section 8.4.
2. **Sample ballots and record audit data.** Use geometric skipping (below) with rate p_0 to select ballots to inspect. Record votes on all inspected ballots.
3. **Check attained risk.** Once the final election results have been reported, for each contest under audit and for each reported (winner, loser) pair (w, ℓ):
   - Calculate B_w, B_ℓ, and B_u from the audit sample.
   - Find the (maximal) p-value from B_w, B_ℓ, B_u using the test in Section 8.3.
4. **Escalate if necessary.** If, for any (w, ℓ) pair, the p-value is greater than α, expand the audit in one of the ways described in Section 8.3.

Procedure for geometric skip sampling

1. **Set the random seed.** In each polling place, use a cryptographically secure PRNG, such as SHA-256, with a seed chosen using true randomness.
2. **Sample ballots.** Following Section 8.5, for each batch of ballots: Set $Y_0 = 0$ and set $j = 0$.
   - $j \leftarrow j + 1$
   - Generate a uniform random variable U on $[0, 1)$.
   - $Y_j \leftarrow \left\lceil \frac{\ln(U)}{\ln(1-p)} \right\rceil$.
   - If $\sum_{k=1}^{j} Y_j$ is greater than the number of ballots in the batch, stop. Otherwise, skip the next $Y_j - 1$ ballots in the bundle, and include the ballot after that one (i.e., include ballot $\sum_{k=1}^{j} Y_j$)

Figure 8.1: Bernoulli ballot-polling audit step-by-step procedures.

8.2 Notation and mathematical background

We consider social choice functions that are variants of majority and plurality voting: the winners are the $k \geq 1$ candidates who receive the most votes. This includes ordinary "first-

past-the-post" contests, as well as "vote for k" contests.[2] As explained in [63], it suffices to consider one (winner, loser) pair at a time: the contest outcome is correct if every reported winner actually received more votes than every reported loser. Auditing majority and supermajority contests requires only minor modifications.[3] Section 8.3 addresses auditing multiple contests simultaneously.

Let w denote a reported winning candidate and ℓ denote a reported losing candidate. Suppose that the population contains N_w ballots with a valid vote for w but not ℓ, N_ℓ ballots with a valid vote for ℓ but not w, and N_u ballots with votes for both w and ℓ or for neither w nor ℓ. The total number of ballots is $N = N_w + N_\ell + N_u$. Let $N_{w\ell} \equiv N_w + N_\ell$ be the number of ballots in the population with a valid vote for w or ℓ but not both. For Bernoulli sampling, N may be unknown; in any event, N_w, N_ℓ, and N_u are unknown, or the audit would not be necessary.

If we can reject the null hypothesis that $N_\ell \geq N_w$ at significance level α, we have statistically confirmed that w got more votes than ℓ. Section 8.3 discusses a test for this hypothesis that accounts for the nuisance parameter $N_{w\ell}$. We assume that ties are settled in a deterministic way and that if the audit is unable to confirm the contest outcome, a full manual tally resulting in a tie would be settled in the same deterministic way.

Multi-round Bernoulli sampling

A *Bernoulli*(p) *random variable* $\mathcal{I}$ is a random variable that takes the value 1 with probability p and the value 0 with probability $1 - p$. BBP uses Bernoulli sampling, which involves independent selection of different ballots with the same probability p of selecting each ballot: $\mathcal{I}_j = 1$ if and only if ballot j is selected to be in the sample, where $\{\mathcal{I}_j\}_{j=1}^N$ are independent, identically distributed (IID) Bernoulli(p) random variables.

Suppose that after tossing a coin with probability p_0 of landing heads for every item in the population, we toss a coin with probability p_1 for every item (again, independently), and include an item in the sample if the first or second toss for that item landed heads. That amounts to drawing a Bernoulli sample using selection probability $1-(1-p_0)(1-p_1)$: an item is in the sample unless its coin landed tails on both tosses, which has probability $(1 - p_0)(1 - p_1)$. This extends to making any integral number K of passes through the population of ballots, with pass k using a coin that has chance p_k of landing heads: such "K-round" Bernoulli sampling is still Bernoulli sampling, with $\mathbb{P}\{\mathcal{I} = 1\} = p = 1-\prod_{k=0}^{K-1}(1-p_k)$.

[2] The same general approach works for some preferential voting schemes, such as Borda count and range voting, and for proportional representation schemes such as D'Hondt [133]. We do not consider instant-runoff voting (IRV).

[3] For instance, for a majority contest, one simply pools the votes for all the reported losers into a single "pseudo-candidate" who reportedly lost.

Exchangeability and conditional simple random sampling

Because the N variables $\{\mathcal{I}_j\}$ are IID, they are *exchangeable*, meaning their joint distribution is invariant under the action of the symmetric group (relabelings). Consider a collection of indices $\mathcal{S} \subset \{1, \ldots, N\}$ of size k, $0 \le k \le N$. Define the event

$$\mathcal{I}_{\mathcal{S}} \equiv \{\mathcal{I}_j = 1, \forall j \in \mathcal{S}, \text{ and } \mathcal{I}_j = 0, \forall j \notin \mathcal{S}\}.$$

Because $\{\mathcal{I}_j\}$ are exchangeable, $\mathbb{P}\mathcal{I}_{\mathcal{S}} = \mathbb{P}\mathcal{I}_{\mathcal{T}}$ for every set $\mathcal{T} \subset \{1, \ldots, N\}$ of size k, since every such set $\mathcal{T}$ can be mapped to $\mathcal{S}$ by a one-to-one relabeling of the indices.

It follows that, conditional on the attained size of the sample, $n = \sum_{j=1}^{N} \mathcal{I}_j$, all $\binom{N}{n}$ subsets of size n drawn from the N items are equally likely: the sample is conditionally a simple random sample (SRS) of size n. This is foundational for applying the SPRT to Bernoulli samples.

8.3 Performing the audit

Suppose we draw a Bernoulli sample of ballots. The random variable B is the number of ballots in the sample. Let B_w denote the number of ballots in the sample with a vote for w but not ℓ; let B_ℓ denote the number of ballots in the sample with a vote for ℓ but not w; and let B_u denote the number of ballots in the sample with a vote for both w and ℓ or neither w nor ℓ, so $B = B_w + B_\ell + B_u$.

We want to test the compound hypothesis that $N_w \le N_\ell$ against the alternative that $N_w = V_w$, $N_\ell = V_\ell$, and $N_u = V_u$, with $V_w - V_\ell > 0$.[4] The ballot polling test for overstatement quotas developed in Section 6.5 can be used here, setting the null overstatement quota c to 0.

Wald's SPRT [145] leads to an elegant escalation method if the first round of Bernoulli sampling does not attain the risk limit: simply make another round of Bernoulli sampling, as described in Section 8.3. If the null hypothesis is true, then $\Pr\{\inf_k P_k < \alpha\} \le \alpha$, where k counts the rounds of Bernoulli sampling. That is, the risk limit remains conservative for any number of rounds of Bernoulli sampling.

Auditing multiple contests

The math extends to audits of multiple contests; we omit the derivation, but see, e.g., [62]. The same sample can be used to audit any number of contests simultaneously. The audit proceeds to a full hand count unless every null hypothesis is rejected, that is, unless we conclude that *every* winner beat *every* loser in *every* audited contest. The chance of rejecting all those null hypotheses cannot be larger than the smallest chance of rejecting any of the individual hypotheses, because the probability of an intersection of events cannot be

[4] The alternative hypothesis is that the reported results are correct; as mentioned above, there are other approaches one could use that do not involve the reported results, but we do not present them here.

larger than the probability of any one of the events. The chance of rejecting any individual null hypothesis is at most the risk limit, α, if that hypothesis is true. Therefore the chance of the intersection is not larger than α if any contest outcome is incorrect: the overall risk limit is α, with no need to adjust for multiplicity.

Escalation

If the first round of Bernoulli sampling with rate p_0 does not generate strong evidence that the election outcome is correct, we have several options:

1. conduct a full hand count
2. augment the sample with additional ballots selected in some manner, for instance, making additional rounds of Bernoulli sampling, possibly with different values of p
3. draw a new sample and use a different auditing method, *e.g.*, ballot-level comparison auditing

The first approach is always conservative. Both the second and third approaches require some statistical care, as repeated testing introduces additional opportunities to wrongly conclude that an incorrect election outcome is correct.

To make additional rounds of Bernoulli sampling, it may help to keep track of which ballots have been inspected.[5] That might involve stamping audited ballots with "audited" in red ink, for example.

Section 8.2 shows that if we make an integral number of passes through the population of ballots, tossing a p_k-coin for each as-yet-unselected item (we only toss the coin for an item on the kth pass if the coin has not landed heads for that item in any previous pass), then the resulting sample is a Bernoulli random sample with selection probability $p = 1 - \prod_{k=0}^{K-1}(1-p_k)$. Conditional on the sample size n attained after K passes, every subset of size n is equally likely to be selected. Hence, the sample is conditionally a simple random sample of size n from the N ballots.

The SPRT applied to multi-round Bernoulli sampling is conservative: the unconditional chance of rejecting the null hypothesis if it is true is at most α, because, if the null is true, the chance that the SPRT exceeds $1/\alpha$ for *any* K is at most α.

The third approach allows us to follow BBP with a different, more efficient approach, such as ballot-level comparison auditing [62]. This may require steps to ensure that multiplicity does not make the risk larger than the nominal risk limit, e.g., by adjusting the risk limit using Bonferroni's inequality.

[5] Once ballots are aggregated in a precinct or scanned centrally, it is unlikely that they will stay in the same order.

8.4 Initial sampling rate

We would like to choose the initial sampling rate p_0 sufficiently large that a test of the hypothesis $N_w \leq N_\ell$ will have high power against the alternative $N_w = V_w, N_\ell = V_\ell$, with $V_w - V_\ell = c$ for modest margins $c > 0$, but not so large that we waste effort.

There is no analytical formula for the power of the sequential hypothesis test under this sampling procedure, but we can use simulation to estimate the sampling rates needed to have a high probability of confirming correctly reported election results. Table 8.1 gives the sampling rate p_0 needed to attain 80%, 90%, and 99% power for a 2-candidate race in which there are no undervotes or invalid votes, for a 5% risk limit and a variety of margins and contest sizes. The simulations assume that the reported vote totals are correct. The required p_0 may be prohibitively large for small races and tight margins; Section 8.6 shows that with high probability, even a 1% sampling rate would be sufficient to confirm the outcomes of the vast majority of U.S. federal races without further escalation.

The sequential probability ratio test in Section 8.3 is similar to the BRAVO RLA presented in [62] when the sampling rate is small relative to the population size. There are two differences between BRAVO and BBP: BBP incorporates information about the number of undervotes, invalid votes, or votes for candidates other than w and ℓ, and Bernoulli sampling is done without (as opposed to with) replacement. If every ballot has a valid vote either for w or for ℓ and the sampling rate is small relative to the population size, the expected workload of these two procedures is similar. The *average sample number* (ASN) [145], the expected number of draws required either to accept or to reject the null hypothesis, for BRAVO using a risk limit α and margin m is approximately

$$\text{ASN} \approx \frac{2\ln(1/\alpha)}{m^2}.$$

This formula is valid when the sampling rate is low and the actual margin is not substantially smaller than the (reported) margin used as the alternative hypothesis.

The ASN gives a rule of thumb for choosing the initial sampling rate for BBP. For a risk limit of 5% and a margin of 5%, the ASN is about 2,400 ballots. For a margin of 10%, the ASN is about 600 ballots. These values are lower than the sample sizes implied by Table 8.1: the sampling rates in the table have a higher probability that the initial sample will be sufficient to conclude the audit, while a sampling rate based on the ASN will suffice a bit more than half of the time.[6] The ASN multiplied by 2–4 is a rough approximation to initial sample size needed to have roughly a 90% chance that the audit can stop without additional sampling, if the reported results are correct.

The ASN formula assumes that N_u is 0; the value of p_0 should be adjusted to account for ballots that have votes for neither w nor ℓ (or for both w and ℓ). If $r = \frac{N_u}{N}$ is the fraction of such ballots, the initial sampling rate p_0 should be inflated by a factor of $\frac{1}{1-r}$. For example,

[6] The distribution of the sample size is skewed to the right: the expected sample size is generally larger than the median sample size.

Table 8.1: Estimated sampling rates needed for Bernoulli ballot polling for a 2-candidate race with a 5% risk limit. These simulations assume the reported margins were correct.

		sampling rate p to achieve ...		
true margin	ballots cast	80% power	90% power	99% power
1%	100,000	55%	62%	77%
2%	100,000	23%	30%	46%
5%	100,000	5%	7%	12%
10%	100,000	2%	2%	4%
20%	100,000	1%	1%	1%
1%	1,000,000	10.4%	14.2%	24.2%
2%	1,000,000	2.9%	4.0%	7.5%
5%	1,000,000	0.5%	0.7%	1.3%
10%	1,000,000	0.2%	0.2%	0.4%
20%	1,000,000	0.1%	0.1%	0.1%
1%	10,000,000	1.15%	1.66%	3.11%
2%	10,000,000	0.30%	0.42%	0.84%
5%	10,000,000	0.05%	0.07%	0.13%
10%	10,000,000	0.02%	0.02%	0.04%
20%	10,000,000	0.01%	0.01%	0.01%

if half of the ballots were undervotes or invalid votes, then double the sampling rate would be needed to achieve the same power as if all of the ballots were valid votes for either w or ℓ.

8.5 Implementation

Election night auditing

Previous approaches to auditing require a sampling frame (possibly stratified, *e.g.*, by mode of voting or county). That requires knowing how many ballots were cast and their locations. In contrast, Bernoulli sampling makes it possible to start the audit at polling places immediately, without even having to count the ballots cast in the polling place. This has several advantages:

1. It parallelizes the auditing task and can take advantage of staff (and observers) who are already on site at polling places.

2. It takes place earlier in the chain of custody of the physical ballots, before the ballots are exposed to some risks of loss, addition, substitution, or alteration.

3. It may add confidence to election-night result reporting.

The benefit is largest if p_0 is large enough to allow the audit to complete without escalating. Since reported margins will not be known on election night, p_0 might be based on pre-election polls, or set to a fixed value. There is, of course, a chance that the initial sample will not suffice to confirm outcomes, either because the true margins are smaller than anticipated, or because the election outcome is in fact incorrect.

There are reasons polling-place BBP audits might not be desirable.

1. Pollworkers, election judges, and observers are likely to be tired and ready to go home when polls close.

2. The training required to conduct and to observe the audit goes beyond what poll workers and poll watchers usually receive.

3. Audit data need to be captured and communicated reliably to a central authority to compute the risk (and possibly escalate the audit) after election results are reported.

Vote-by-mail and provisional ballots

The fact that Bernoulli sampling is a "streaming" algorithm may help simplify logistics compared with other sampling methods. For instance, Bernoulli sampling can be used with vote-by-mail (VBM) ballots and provisional ballots. VBM and provisional ballots can be sampled as they arrive (after signature verification), or aggregated, e.g., daily or weekly. Ballots do not need to be opened or examined immediately in order to be included in the sample: they can be set aside and inspected after Election Day or after their provisional status has been adjudicated. Any of these approaches yields a Bernoulli sample of all ballots cast in the election, provided the same value(s) of p are used throughout.

Geometric skipping

In principle, one can implement Bernoulli sampling by actually rolling dice, or by assigning a $U[0,1]$ random number to each ballot, independently across ballots. A ballot is in the sample if and only if its associated random number is less than or equal to p.

However, that places an unnecessarily high burden on the quality of the pseudorandom number generator—or on the patience of the people responsible for selecting ballots by mechanical means, such as by rolling dice. If the ballots are in physical groups (e.g., all ballots cast in a precinct), it can be more efficient to put the ballots into some canonical order (for instance, the order in which they are bundled or stacked) and to rely on the fact that the *waiting times* between successes in independent Bernoulli(p) trials are independent Geometric(p) random variables: the chance that the next time the coin lands heads will be kth tosses after the current toss is $p(1-p)^{k-1}$.

To select the sample, instead of generating a Bernoulli random variable for every ballot, we suggest generating a sequence of geometric random variables $Y_1, Y_2, \ldots$ The first ballot in the sample is the one in position Y_1 in the group, the second is the one in position $Y_1 + Y_2$, and so on. We continue in this way until $Y_1 + \ldots + Y_j$ is larger than the number of ballots in the group. This *geometric skipping* method is implemented in the software we provide.

Pseudo-random number generation

To draw the sample, we propose using a cryptographically secure PRNG based on the SHA-256 hash function, setting the seed using 20 rolls of 10-sided dice, in a public ceremony. This is the method that the State of Colorado uses to select the sample for risk-limiting audits.

This is a good choice for election audits for several reasons. First, given the initial seed, anyone can verify that the sequence of ballots audited is correct. Second, unless the seed is known, the ballots to be audited are unpredictable, making it difficult for an adversary to "game" the audit. Finally, this family of PRNGs produces high-quality pseudorandomness.

Implementations of SHA-256-based PRNGs are available in many languages, including Python and Javascript. The code we provide for geometric skipping relies on the `cryptorandom` Python library, which implements such a PRNG; see Chapter 4.

While Colorado sets the seed for the entire state in a public ceremony, it may be more secure to generate seeds for polling-place audits locally, after the ballots have been collated into stacks that determine their order for the purpose of the audit. If the seed were known before the order of the ballots was fixed, an adversary might be able to arrange that the ballots selected for auditing reflect a dishonest outcome.

While the sequence of ballots selected by this method is verifiable, there is no obvious way to verify *post facto* that the ballots examined were the correct ones. Only observers of the audit can verify that. Observers' job would be easier if ballots were pre-stamped with (known) unique identifiers, but that might compromise vote anonymity.

8.6 Evaluation

As discussed in Section 8.4, we expect that *workload* (total number of ballots examined) for Bernoulli ballot polling to be approximately the same as BRAVO ballot polling. Figure 8.2 compares the fraction of ballots examined for BRAVO audits and BBP for a 2-candidate contest, estimated by simulation. The simulations use contest sizes of 10,000 and 1,000,000 ballots, each of which has either a valid vote for the winner or a valid vote for the loser. The percentage of votes for the winner ranges from 99% (almost all the votes go to the winner) to 50% (a tie). The methods produce similarly shaped curves; BBP requires slightly more ballots than BRAVO.

As the workload of BRAVO and BBP are similar, the cost of running a Bernoulli audit should be similar to BRAVO. There are likely other efficiencies to Bernoulli audits, *e.g.*, if the first stage of the audit can be completed on election night in parallel, it might result

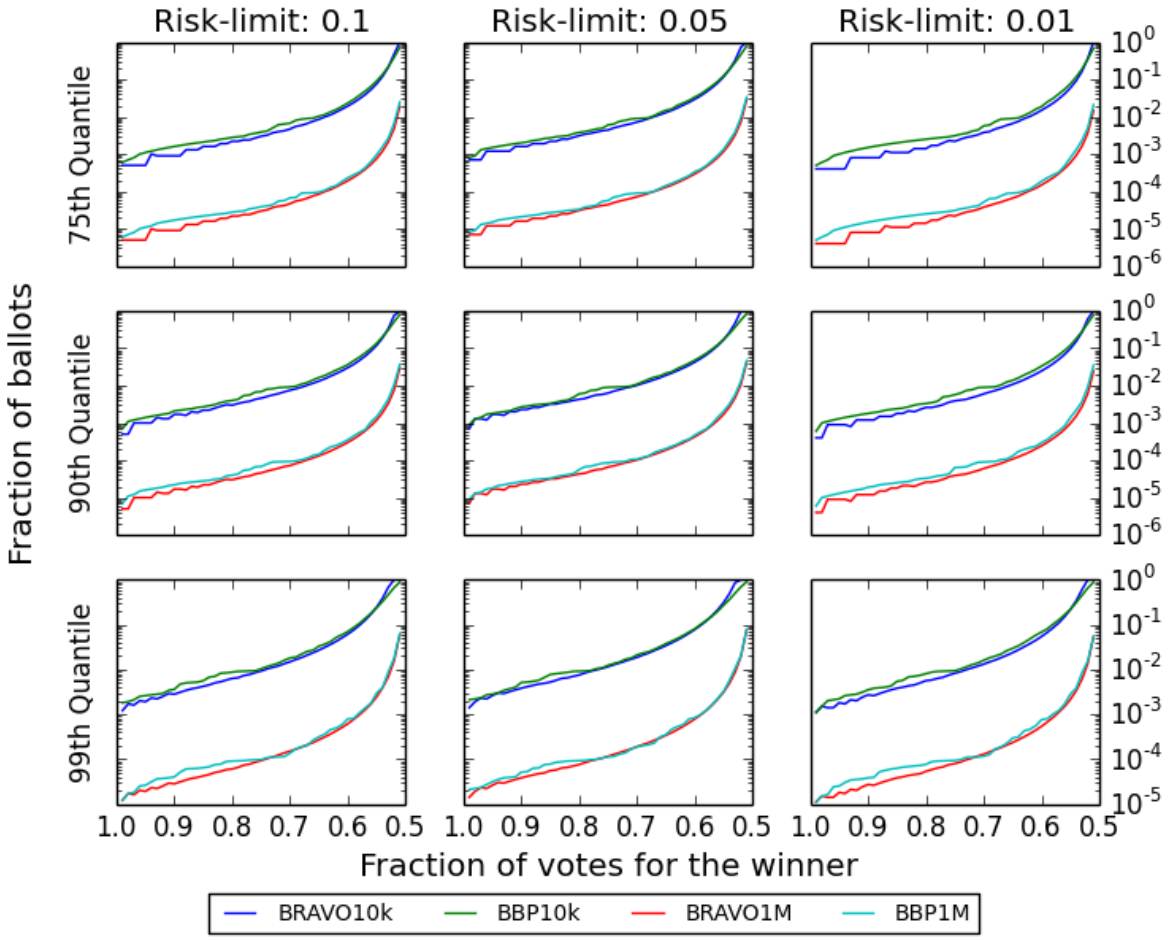

Figure 8.2: Simulated quantiles of sample sizes by fraction of votes for the winner for a two candidate race in elections with 10,000 ballots and 1 million ballots, for BRAVO ballot-polling audits (BPA) and Bernoulli ballot polling audits (BBP), for various risk-limits. The simulations assume every ballot has a valid vote for one of the two candidates.

in lower cost as election workers and observers would not have to assemble in a different place and time for the audit. Even if the cost were somewhat higher, that might be offset by advantages discussed in Section 8.7.

Empirical data

We evaluate BBP using state-level data from the 2016 U.S. presidential election, collected from OpenElections [91] or by hand where that dataset was incomplete. If the reported margins are correct, BBP with a sampling rate of $p_0 = 1\%$ and a risk-limit of 5% would have a 99% or higher chance of confirming the outcome in 42 states. Table 8.6 shows the reported

margins and estimated power in the 8 states with a lower than 99% chance of confirming the outcome. Each of these states had a reported margin below 3%, so it is likely that any RLA procedure would escalate. The workload for BPA would have a prohibitively high.

While the total sample size for a 1% BBP audit within states may appear large, and is substantially larger than the workload for BPA when the margin is wide, the mean sample size per-precinct for BBP is only about 10 ballot. If the audit is conducted in-precinct, the workload would be fairly minute.

State	Total	Margin	BBP Power	BBP Workload	BPA Workload
NH	744,296	0.37%	0.00	7,442	413,892
MI	4,799,284	0.22%	0.00	47,992	114,1942
WI	2,976,150	0.76%	0.04	29,761	96,194
PA	6,115,402	0.72%	0.18	61,154	110,278
NV	1,125,385	2.42%	0.50	11,253	9,619
MN	2,944,782	1.52%	0.54	29,447	23,763
ME	771,892	2.87%	0.55	7,718	6,573
FL	9,420,039	1.20%	0.82	94,200	40,469

Table 8.2: 8 states in the 2016 US Presidential election for which a 1% BBP audit would have less than 99% chance of confirming the outcome.

8.7 Discussion

Bernoulli ballot polling has a number of practical advantages. We have discussed several throughout the paper, but we review all of them here:

- It reduces the need for a ballot manifest: ballots can be stored in any order, and the number of ballots in a given container or bundle does not need to be known to draw the sample.
- The work can be conducted in parallel across polling places, and can be performed by workers (and observed by members of the public) already in place on Election Day.
- The same sampling method can be used for polling places, vote centers, VBM, and provisional ballots, without the need to stratify the sample explicitly.
- If the initial sampling rate is adequate, the winners can be confirmed shortly after voting finishes—perhaps even at the same time that results are announced—possibly increasing voter confidence.

- When a predetermined expected sampling rate is used, the labor required can be estimated in advance, assuming escalation is not required. With appropriate parameter choices, escalation can be avoided except in unusually close races, or when the reported outcome is wrong. This helps election officials plan.

- The sampling approach is conceptually easy to grasp: toss a coin for each ballot. The audit stops when the sample shows a sufficiently large margin for every winner over every loser, where "sufficiently large" depends on the sample size.

- The approach may have security advantages, since waiting longer to audit would leave more opportunity for the paper ballots to be compromised or misplaced. Workers will need to handle the ballot papers in any case to move them from the ballot boxes into long-term storage.

Officials selecting an auditing method should weigh these advantages against some potential downsides of our approach, particularly when applied in polling places on election night. Poll workers are already very busy, and they may be too tired at the end of the night to conduct the sampling procedure or to do it accurately. When audits are conducted in parallel at local polling places, it is impossible for an individual observer to witness all the simultaneous steps. Moreover, estimating the sample size before margins are known makes it likely that workers will end up sampling more (or fewer) ballots than necessary to achieve the risk limit. While sampling too little can be overcome with escalation, the desire to avoid escalation may make officials err on the side of caution and sample more than predicted to be necessary, further reducing expected efficiency.

Previous work

Bernoulli sampling is a special case of Poisson sampling, where sampling units are selected independently, but not necessarily with equal probability. [4] propose a Poisson sampling method in which the probability of selecting a given unit is related to a bound on the error that unit could hide. Their method is not an RLA: it is designed to have a large chance of detecting at least one error if the outcome is incorrect, rather than to limit the risk of certifying an incorrect outcome *per se*.

Stratified audits

Independent Bernoulli samples from different populations using the same rate still yields a Bernoulli sample of the overall population, so the math presented here can be used without modification to audit contests that cross jurisdictional boundaries. Bernoulli samples from different strata using different rates can be combined using SUITE [94], which can be applied to stratum-wise p-values from any method, including BBP. (This requires minor modifications to the p-value calculations, to test arbitrary hypotheses about the margin in

each stratum rather than to test for ties; the derivations in Chapter 6 apply, *mutatis mutandis*.) If some ballots are tabulated using technology that makes a more efficient auditing approach possible, such as a ballot-level comparison audit, it may be advantageous to stratify the ballots into groups, sample using Bernoulli sampling in some and a different method in others, and use SUITE to combine the results into an overall RLA.

8.8 Conclusion

We presented a new ballot-polling RLA based on Bernoulli sampling, relying on Wald's sequential probability ratio test to calculate the sample's risk limit. The new method performs similarly to the BRAVO ballot-polling audit but has several logistical advantages, including that it can be parallelized and conducted on election night, which may reduce cost and increase security. The method easily incorporates VBM and provisional ballots, and may eliminate the need for stratification in many circumstances. Bernoulli ballot polling with just a 1% sampling rate would have sufficed to confirm the 2016 U.S. Presidential election results in the vast majority of states, if the reported results were correct. The practical benefits and conceptual simplicity of Bernoulli ballot polling may make it simpler to conduct risk-limiting audits in real elections.

Appendix A

Appendix for Chapter 3

A.1 Derangements

Theorem 2. *The probability that a random permutation of a list is a derangement converges to e^{-1} as the length of the list grows.*

Proof. If all permutations occur with equal frequency, then the probability that a permutation is a derangement is given by $p_0 = \frac{D_n}{n!}$, where D_n is the number of derangements of n items. Let A_p denote the set of permutations of n items that fix at least p items, for $p = 1, \ldots, n$. The set of derangements is the complement of the union of these sets, so $D_n = n! - |\cup_{p=1}^{n} A_p|$. Using the inclusion-exclusion principle, we have

$$\begin{aligned}|\cup_{p=1}^{n} A_p| &= \sum_{p=1}^{n} |A_p| - \sum_{p<q} |A_p \cap A_q| + \cdots + (-1)^{n+1} |\cap_{p=1}^{n} A_p| \\ &= \#\text{ permutations fixing one item} - \#\text{ permutations fixing two items} + \ldots \\ &\quad + (-1)^{n+1} \#\text{ permutations fixing all items.}\end{aligned}$$

The number of permutations fixing p items is $(n-p)!\binom{n}{p}$ because after fixing p points, we can freely permute $n-p$ of them, and there are $\binom{n}{p}$ subsets of p points that we could fix. Plugging this back in, we get

$$\begin{aligned}D_n &= n! - (n-1)!\binom{n}{1} + (n-2)!\binom{n}{2} - \cdots + (-1)^n \binom{n}{n} \\ \frac{D_n}{n!} &= 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!} \\ &= \sum_{p=0}^{n} \frac{(-1)^p}{p!}.\end{aligned}$$

This sum converges quickly to the Taylor series for $e^{-1} \approx 0.3678794$; the approximation error is smaller than 10^{-8} for $n > 10$.

□

A.2 Fixed points of a permutation

Theorem 3. *The distribution of the number of points fixed between two random permutations is asymptotically Poisson distributed with parameter* 1*. For any list length* n*,*

$$|\mathbb{P}(S_n = k) - \mathbb{P}(W = k)| \leq \frac{1}{k!} \frac{1}{(n-k+1)!}.$$

Furthermore, using the Poisson approximation instead of the exact distribution to compute expected cell counts for the chi-squared statistic inflates the chi-squared statistic by a factor of at most n^{-2n}*.*

Proof. Let S_n denote the number of matches between the previous permutation and the current permutation. As before, D_n denotes the number of derangements of n items. D_n satisfies the recurrence relation $D_n = (n-1)\left(D_{n-1} + D_{n-2}\right)$ with initial values $D_1 = 0, D_2 = 1$.

$$\begin{aligned} P(S_n = 0) &= \frac{D_n}{n!} \\ P(S_n = 1) &= \frac{nD_{n-1}}{n!} \\ &\vdots \\ P(S_n = k) &= \frac{\binom{n}{k} D_{n-k}}{n!}. \end{aligned}$$

The number of matches S_n is a sum of indicator variables, one for each position in the list. $X_i = 1$ if position i is the same in the two permutations and 0 otherwise. The probability that position i is fixed is $\frac{1}{n}$, for each i.

The sum of weakly dependent Bernoulli random variables is asymptotically Poisson distributed, with parameter equal to the sum of all of the Bernoulli parameters [33]. We'll show that S_n can be approximated well by a Poisson random variable with parameter 1. We begin by writing explicitly the probability distribution of S_n.

$$\begin{aligned}\mathbb{P}(S_n = k) &= \frac{\binom{n}{k} D_{n-k}}{n!} \\ &= \frac{n! D_{n-k}}{n!(n-k)!k!} \\ &= \frac{!(n-k)}{(n-k)!k!} \\ &= \frac{(n-k)! \sum_{i=0}^{n-k} \frac{(-1)^i}{i!}}{(n-k)!k!} \\ &= \frac{1}{k!}\left(\sum_{i=0}^{n-k} \frac{(-1)^i}{i!}\right).\end{aligned}$$

Let W be a Poisson(1) random variable, so $\mathbb{P}(W = k) = \frac{e^{-1}}{k!}$. For any $k \geq 0$, the approximation error is

$$|\mathbb{P}(S_n = k) - \mathbb{P}(W = k)| = \Big|\frac{1}{k!} \sum_{i=n-k+1}^{\infty} \frac{(-1)^i}{i!}\Big| \leq \frac{1}{k!} \frac{1}{(n-k+1)!}.$$

Now, we are interested in how this approximation error shows up in the chi-squared test. Suppose we generate r random permutations of a list of length n and compute S_n for each consecutive pair. Consider the kth component of the chi-squared statistic. When we use the approximate distribution, there is some error δ in computing the expected number in category k. To first order,

$$\begin{aligned}\frac{(O_k - E_k - \delta)^2}{E_i + \delta} &= \frac{(O_i - E_i)^2}{E_i} + \delta\left(1 - \frac{O_i^2}{E_i^2}\right) \\ &= \frac{(O_i - E_i)^2}{E_i}\left(1 + \frac{\delta}{E_i}\right).\end{aligned}$$

The approximation error $\delta \leq r|\mathbb{P}(S_n = k) - \mathbb{P}(W = k)| \leq \frac{r}{k!(n-k+1)!}$ and $E_k = \frac{r}{k!} \sum_{j=0}^{n-k} \frac{(-1)^j}{j!} = \frac{r}{k!}\left(e^{-1} - \delta k!\right)$.

$$\frac{\delta}{E_k} \leq \frac{1}{(n-k+1)!(e^{-1} - \frac{1}{(n-k+1!)})} \approx \frac{1}{n^{2n}}.$$

Thus, the chi-squared statistic is inflated by a factor of at most n^{-2n}. This becomes miniscule as n increases.

□

Bibliography

1. *American Statistical Association Endorses Post-Election Audits Principles* Last visited 3/7/19 (American Statistical Association, 2019). https://www.amstat.org/asa/files/pdfs/pressreleases/2019-AuditPrinciplesRelease.pdf.
2. *ANU Quantum Random Numbers Server* Last visited 1/18/19. http://qrng.anu.edu.au.
3. Arbuckle, J. & Williams, B. D. Students' Perceptions of Expressiveness: Age and Gender Effects on Teacher Evaluations. *Sex Roles* **49,** 507–516 (2003).
4. Aslam, J. A., Popa, R. A. & Rivest, R. L. On Auditing Elections When Precincts Have Different Sizes. *2008 USENIX/ACCURATE Electronic Voting Technology Workshop* (2008).
5. Banuelos, J. H. & Stark, P. B. Limiting Risk by Turning Manifest Phantoms into Evil Zombies. eprint: arXiv/1207.3413 (2012).
6. Becker, R. A. A Brief History of S. *Computational Statistics,* 81–110 (1994).
7. Bennett, S. K. Student Perceptions of and Expectations for Male and Female Instructors: Evidence Relating to the Question of Gender Bias in Teaching Evaluation. *Journal of Educational Psychology* **74,** 170–179 (1982).
8. Benton, S. L. & Cashin, W. E. *Student Ratings of Teaching: A Summary of Research and Literature* IDEA Paper 50 (The IDEA Center, 2012).
9. Blaze, M. *et al. DEFCON 25 Voting Village Report* Last visited 4/10/19 (2017). https://www.defcon.org/images/defcon-25/DEFCON%5C%2025%5C%20voting%5C%20village%5C%20report.pdf.
10. Blaze, M. *et al. DEFCON 26 Voting Village Report* Last visited 4/10/19 (2018). https://www.defcon.org/images/defcon-26/DEF%5C%20CON%5C%2026%5C%20voting%5C%20village%5C%20report.pdf.
11. Blinder, A. Federal Judge Delays Certification of Georgia Election Results. *The New York Times.* Last visited 1/11/19. https://www.nytimes.com/2018/11/12/us/georgia-governor-election.html (2018).
12. Boring, A. *Can Students Evaluate Teaching Quality Objectively?* Le Blog de l'OFCE (OFCE, 2015).

13. Boring, A. *Gender Biases in Student Evaluations of Teachers* Document de travail OFCE 13 (OFCE, 2015).

14. Bowen, D. *Top-to-Bottom Review of voting machines certified for use in California* tech. rep. Last visited 1/11/19 (California Secretary of State, 2007). https://www.sos.ca.gov/elections/voting-systems/oversight/top-bottom-review/.

15. Braga, M., Paccagnella, M. & Pellizzari, M. Evaluating Students' Evaluations of Professors. *Economics of Education Review* **41,** 71–88 (2014).

16. Bretschneider, J. *et al. Risk-Limiting Post-Election Audits: Why and How* tech. rep. Last visited 12/17/18 (Risk-Limiting Audits Working Group, 2012). https://www.stat.berkeley.edu/~stark/Preprints/RLAwhitepaper12.pdf.

17. Brombin, C., Salmaso, L., Fontanella, L. & Ippoliti, L. Nonparametric Combination-Based Tests in Dynamic Shape Analysis. *Journal of Nonparametric Statistics* **27,** 460–484 (2015).

18. Carrell, S. E. & West, J. E. Does Professor Quality Matter? Evidence from Random Assignment of Students to Professors. *Journal of Political Economy* **118,** 409–432 (2010).

19. Caughey, D., Dafoe, A. & Seawright, J. Nonparametric Combination (NPC): A Framework for Testing Elaborate Theories. *The Journal of Politics* **79,** 688–701 (2017).

20. Centra, J. A. Student Ratings of Instruction and Their Relationship to Student Learning. *American Educational Research Journal* **14,** 17–24 (1977).

21. Centra, J. A. & Gaubatz, N. B. Is There Gender Bias in Student Evaluations of Teaching? *Journal of Higher Education* **71,** 17–33 (2000).

22. *Coalition for Good Governance, Martin, Duval, and Dufort v. Crittenden* 2018-CV-3134-18 (Superior Court of Fulton County, State of Georgia, 2019).

23. Cochran, W. G. *Sampling Techniques* 3rd ed. (Wiley, India, 1977).

24. Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. *Introduction to Algorithms, 3rd edition* (MIT Press, Cambridge, Massachusetts, 2009).

25. *Curling v. Kemp* (Order Denying Motion to Dismiss) 1:17-CV-2989-AT (United States District Court for the Northern District of Georgia, Atlanta Division, 2018).

26. *Curling v. Kemp* (Amicus Curiae Brief) 1:17-CV-2989-AT (United States District Court for the Northern District of Georgia, Atlanta Division, 2018).

27. Dahlberg, L. & McCaig, C. *Practical Research and Evaluation: A Start-to-finish Guide for Practitioners* (Sage, London, 2010).

28. Elmore, P. B. & LaPointe, K. A. Effects of Teacher Sex and Student Sex on the Evaluation of College Instructors. *Journal of Educational Psychology* **66,** 386–389 (1974).

29. Everett, S. P. *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection* Last visited 1/15/19. PhD thesis (Rice University, Houston, Texas, 2007). https://scholarship.rice.edu/handle/1911/20601.
30. Fan, C. T., Muller, M. E. & Rezucha, I. Development of Sampling Plans by Using Sequential (Item by Item) Selection Techniques and Digital Computers. *Journal of the American Statistical Association* **57,** 387–402 (1962).
31. Fisher, R. A. *The Design of Experiments* (Oliver & Boyd, Edinburgh; London, 1937).
32. Freedman, D. A. On Regression Adjustments to Experimental Data. *Advances in Applied Mathematics* **40,** 180–193 (2008).
33. Freedman, D. A. The Poisson Approximation for Dependent Events. *The Annals of Probability* **2,** 256–269 (1974).
34. Freeman, S. F. & Bleifuss, J. *Was the 2004 Presidental Election Stolen?: Exit Polls, Election Fraud, and the Official Count* (Seven Stories Press, New York, 2006).
35. Galbraith, C. S., Merrill, G. B. & Kline, D. M. Are Student Evaluations of Teaching Effectiveness Valid for Measuring Student Learning Outcomes in Business Related Classes? A Neural Network and Bayesian Analyses. *Research in Higher Education* **53,** 353–374 (2012).
36. Geiger, R. S., Varoquaux, N., Mazel-Cabasse, C. & Holdgraf, C. The Types, Roles, and Practices of Documentation in Data Analytics Open Source Software Libraries: A Collaborative Ethnography of Documentation Work. *Computer Supported Cooperative Work (CSCW)* **27,** 767–802 (2018).
37. Geiger, R. S. *et al.* Career Paths and Prospects in Academic Data Science: Report of the Moore-Sloan Data Science Environments Survey. Last visited 1/25/19. eprint: https://osf.io/preprints/socarxiv/xe823/ (2018).
38. *Georgia Unverifiable Voting System Chronology* blog. Last visited 1/11/19 (Voters Organized for Trusted Election Results in Georgia, 2014). https://voterga.org/history/.
39. Grimmett, G. R. & Stirzaker, D. R. *Probability and Random Processes* (Oxford University Press, Oxford, 2001).
40. Hall, J. L. *et al. Implementing Risk-Limiting Audits in California* in *EVT/WOTE '09* Montreal, Canada (2009).
41. Hamermesh, D. S. & Parker, A. Beauty in the Classroom: Instructors' Pulchritude and Putative Pedagogical Productivity. *Economics of Education Review* **24,** 369–376 (2005).
42. Harnik, A. & Press, A. Officials Scrap Plan To Cut Most Polling Places In Majority Black Ga. County. *WABE.* Last visited 1/11/19. https://www.wabe.org/officials-scrap-plan-to-cut-most-polling-places-in-majority-black-ga-county/ (2018).

43. Harriot, M. Thousands of Black Votes in Georgia Disappeared. *The Root.* Last visited 2/18/19. https://www.theroot.com/exclusive-thousands-of-black-votes-in-georgia-disappea-1832472558 (2019).
44. Hartley, H. O. & Pearson, E. S. Moment Constants for the Distribution of Range in Normal Samples. *Biometrika* **38,** 463–464 (1951).
45. Hessler, M. *et al.* Availability of Cookies During an Academic Course Session Affects Evaluation of Teaching. *Medical Education* **52,** 1064–1072 (2018).
46. Higgins, M. J., Rivest, R. L. & Stark, P. B. Sharper p-values for Stratified Post-Election Audits. *Statistics, Politics, and Policy* **2** (2011).
47. Hill, M. C. & Epps, K. K. The Impact of Physical Classroom Environment on Student Satisfaction and Student Evaluation of Teaching in the University Environment. *Academy of Educational Leadership Journal* **14,** 65–79 (2010).
48. Hodges Jr, J. L. & Lehmann, E. L. *Basic Concepts of Probability and Statistics* (SIAM, San Francisco, California, 1970).
49. Hull, T. E. & Dobell, A. Random Number Generators. *SIAM Review* **4,** 230–254 (1962).
50. Ihaka, R. & Gentleman, R. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics* **5,** 299–314 (1996).
51. Johnson, N. L. & Young, D. H. Some Applications of Two Approximations to the Multinomial Distribution. *Biometrika* **47,** 463–469 (1960).
52. Johnson, V. E. *Grade Inflation: A Crisis in College Education* (Springer-Verlag, New York, 2003).
53. Jones, D. W. & Simons, B. *Broken Ballots: Will Your Vote Count?* (CSLI Publications, Stanford, California, 2012).
54. Kennedy, R. J. Will the Next Election Be Hacked? Electronic Voting Machines Can't be Trusted. *Rolling Stone.* Last visited 1/11/19. https://www.organicconsumers.org/news/robert-kennedy-jr-will-next-election-be-hacked-electronic-voting-machines-cant-be-trusted (2006).
55. Knuth, D. E. *Art of Computer Programming, Volume 2: Seminumerical Algorithms* 3rd ed. (Addison-Wesley Professional, Reading, Massachusetts, 1997).
56. Lauer, C. in *To Improve the Academy: Resources for Faculty, Instructional, and Educational Development* (eds Groccia, J. E. & Cruz, L.) 1, 194–211 (Jossey-Bass, 2012). doi:10.1002/j.2334-4822.2012.tb00682.x.
57. LeBlanc, D. C. *Statistics: Concepts and Applications for Science* (Jones & Bartlett Learning, Sudbury, Massachusetts, 2004).
58. L'Ecuyer, P. & Simard, R. TestU01: A C Library for Empirical Testing of Random Number Generators. *ACM Transactions on Mathematical Software (TOMS)* **33,** 22 (2007).

59. Lehman, G. & Doty, C. L. *Memorandum: Manhattan Certification Report, General Election, November 6, 2018* Last visited 12/13/18 (Board of Elections in the City of New York, 2018).

60. *Letter to Governor Newsom and the Legislature on Voting Equipment Security* tech. rep. 247. Last visited 3/7/19 (Little Hoover Commission, 2019). https://lhc.ca.gov/sites/lhc.ca.gov/files/Reports/247/Report247.pdf.

61. Lindeman, M. *City of Fairfax, VA: Pilot Risk-Limiting Audit* tech. rep. Last visited 12/19/18 (Verified Voting Foundation, 2018). https://www.verifiedvoting.org/wp-content/uploads/2018/12/2018-RLA-Report-City-of-Fairfax-VA.pdf.

62. Lindeman, M. & Stark, P. B. A Gentle Introduction to Risk-Limiting Audits. *IEEE Security and Privacy* **10,** 42–49 (2012).

63. Lindeman, M., Stark, P. B. & Yates, V. *BRAVO: Ballot-Polling Risk-Limiting Audits to Verify Outcomes* in *EVT/WOTE '12* (2012).

64. Lockhart, P. R. Voting Hours in Parts of Georgia Extended After Technical Errors Create Long Lines. *Vox.* Last visited 1/11/19. https://www.vox.com/policy-and-politics/2018/11/6/18068492/georgia-voting-gwinnett-fulton-county-machine-problems-midterm-election-extension (2018).

65. Lohr, S. *Sampling: Design and Analysis* (Nelson Education, Boston, Massachusetts, 2009).

66. MacNell, L., Driscoll, A. & Hunt, A. N. What's in a Name: Exposing Gender Bias in Student Ratings of Teaching. *Innovative Higher Education* **40,** 291–303 (2015).

67. Markowsky, G. The Sad History of Random Bits. *Journal of Cyber Security* **3,** 1–26 (2014).

68. Marsaglia, G. Random Numbers Fall Mainly in the Planes. *Proceedings of the National Academy of Sciences of the United States of America* **61,** 25–28 (1968).

69. Marsaglia, G. *The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness* Last visited 9/23/16. www.stat.fsu.edu/pub/diehard.

70. Marsh, H. W. & Roche, L. A. Making Students' Evaluations of Teaching Effectiveness Effective. *American Psychologist* **52,** 1187–1197 (1997).

71. Matsumoto, M. & Nishimura, T. Mersenne Twister: a 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Transactions on Modeling and Computer Simulation* **8,** 3–30 (1998).

72. McCullough, B. D. Microsoft Excel's 'Not The Wichmann–Hill' Random Number Generators. *Computational Statistics & Data Analysis* **52,** 4587–4593 (2008).

73. McDaniel, P., Blaze, M. & Vigna, G. *EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing* tech. rep. Last visited 1/11/19 (Ohio Secretary of State, 2007). http://siis.cse.psu.edu/everest.html.

74. Mengel, F., Sauermann, J. & Zölitz, U. Gender Bias in Teaching Evaluations. *Journal of the European Economic Association* **17,** 535–566 (2019).

75. Merritt, D. J. Bias, the Brain, and Student Evaluations of Teaching. *St. John's Law Review* **81,** 235–288 (2008).

76. Millman, K. J., Ottoboni, K., Stark, N. A. P. & Stark, P. B. in *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences* (eds Kitzes, J., Turek, D. & Deniz, F.) (University of California Press, Oakland, California, 2017).

77. Millman, K. J. & Pérez, F. in *Implementing Reproducible Research* (eds Stodden, V., Leisch, F. & Peng, R. D.) 149–183 (Chapman and Hall/CRC, London, 2014).

78. Nadler, B. Voting Rights Become a Flashpoint in Georgia Governor's Race. *AP News.* Last visited 4/11/19. https://apnews.com/fb011f39af3b40518b572c8cce6e906c (2018).

79. Nakashima, E. In Georgia, A Legal Battle Over Electronic vs. Paper Voting. *The Washington Post.* Last visited 1/9/19. https://www.washingtonpost.com/world/national-security/in-georgia-a-legal-battle-over-electronic-vs-paper-voting/2018/09/16/d655c070-b76f-11e8-94eb-3bd52dfe917b_story.html (2018).

80. National Academies of Sciences, E. & Medicine. *Securing the Vote: Protecting American Democracy* (National Academies Press, Washington, D.C., 2018).

81. Neyman, J., Dabrowska, D. M. & Speed, T. P. On the Application of Probability Theory to Agricultural Experiments. Essay on Principles. Section 9. *Statistical Science* **5,** 465–472 (1990).

82. Niesse, M. Bill to Buy New Georgia Voting Machines Clears Committees. *The Atlanta Journal-Constitution.* Last visited 3/8/19. https://www.myajc.com/news/state--regional-govt--politics/new-georgia-voting-machines-approved-house-committee/avz21tiapWPwM1Qx4bq3AP/ (2019).

83. Niesse, M., Prabhu, M. T. & Elias, J. Voting Precincts Closed Across Georgia Since Election Oversight Lifted. *The Atlanta Journal-Constitution.* Last visited 1/11/19. https://www.myajc.com/news/state--regional-govt--politics/voting-precincts-closed-across-georgia-since-election-oversight-lifted/bBkHxptlim0Gp9pKu7 (2018).

84. *NIST Randomness Beacon* Last visited 1/18/19. https://beacon.nist.gov.

85. Norden, L. & Famighetti, C. *America's Voting Machines at Risk* tech. rep. (Brennan Center for Justice, 2015).

86. O'Brien, T. M. *Denver Elections Division Risk-Limiting Audit Process* tech. rep. Last visited 12/19/18 (Office of the Auditor, Audit Services Division, City and County of Denver, 2018). https://denverauditor.org/project/denver-elections-division-risk-limiting-audit-process/.

87. Of State, C. S. *California Secretary of State Post-Election Risk-Limiting Audit Pilot Program 2011-2013: Final Report to the United States Election Assistance Commission* tech. rep. Last visited 4/9/19 (2014). http://votingsystems.cdn.sos.ca.gov/oversight/risk-pilot/final-report-073014.pdf.

88. On Intelligence, U. S. S. C. *Russian Targeting of Election Infrastructure During the 2016 Election: Summary of Initial Findings and Recommendations* Last visited 4/11/19. 2018. https://www.burr.senate.gov/imo/media/doc/RussRptInstlmt1-%5C%20ElecSec%5C%20Findings,Recs2.pdf.

89. On Nonstandard Mixtures of Distributions, P. *Statistical Models and Analysis in Auditing: A Study of Statistical Models and Methods for Analyzing Nonstandard Mixtures of Distributions in Auditing* (National Academy Press, Washington, D.C., 1988).

90. Ong, L. *A Guide to IMF Stress Testing: Methods and Models* (International Monetary Fund, Washington, D.C., 2014).

91. OpenElections. Last visited 4/17/19. 2018.

92. Ottoboni, K., Lewis, F. & Salmaso, L. An Empirical Comparison of Parametric and Permutation Tests for Regression Analysis of Randomized Experiments. *Statistics in Biopharmaceutical Research* **10,** 264–273 (2018).

93. Ottoboni, K. & Stark, P. B. Random Problems with R. eprint: arXiv/1809.06520 (2018).

94. Ottoboni, K., Stark, P. B., Lindeman, M. & McBurnett, N. *Risk-Limiting Audits by Stratified Union-Intersection Tests of Elections (SUITE)* in *International Joint Conference on Electronic Voting* (2018), 174–188.

95. Party, G. D. *Breaking: Democratic Party of Georgia Calls on SAFE Commission to Delay Vote Following News That Voting Machine Lobbyist Is Longtime Kemp Crony* blog. Last visited 1/14/19 (2019). https://www.georgiademocrat.org/2019/01/breaking-democratic-party-of-georgia-calls-on-safe-commission-to-delay-vote-following-news-that-voting-machine-lobbyist-is-longtime-kemp-crony/.

96. Peck, R., Olsen, C. & Devore, J. L. *Introduction to Statistics and Data Analysis* (Cengage Learning, Boston, Massachusetts, 2011).

97. Peha, J. Touch-and-Go Elections: The Perils of Electronic Voting. *The Nation.* Last visited 1/15/19. https://www.thenation.com/article/touch-and-go-elections-perils-electronic-voting/ (2006).

98. Pesarin, F. & Salmaso, L. *Permutation Tests for Complex Data: Theory, Applications and Software* (Wiley, New York, 2010).

99. Pitman, E. J. G. Significance Tests Which May Be Applied to Samples from Any Populations (Parts I and II). *Supplement to the Journal of the Royal Statistical Society* **4,** 119–130, 225–232 (1937).

100. *Post-Election Audit Manual* version 11.7.2018. Last visited 4/9/19. https://www.michigan.gov/documents/sos/Post_Election_Audit_Manual_418482_7.pdf.

101. Pounder, J. S. Is Student Evaluation of Teaching Worthwhile?: An Analytical Framework for Answering the Question. *Quality Assurance in Education* **15,** 178–191 (2007).

102. Press, W., Flannery, B. P., Teukolsky, S. & Vetterling, W. *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 1988).

103. R Core Team. *R: A Language and Environment for Statistical Computing* Last visited 4/8/19. R Foundation for Statistical Computing (Vienna, Austria, 2018). https://www.R-project.org.

104. *Report of the Auditability Working Group* tech. rep. Last visited 4/11/19 (National Institute of Standards and Technology, Jan. 2015). https://www.eac.gov/assets/1/28/AuditabilityReport_final_January_2011.pdf.

105. Riley, M. & Robertson, J. Russian Hacks on U.S. Voting System Wider Than Previously Known. *Bloomberg.* Last visited 1/9/19. https://www.bloomberg.com/news/articles/2017-06-13/russian-breach-of-39-states-threatens-future-u-s-elections (2017).

106. Riniolo, T. C., Johnson, K. C., Sherman, T. R. & Misso, J. A. Hot or Not: Do Professors Perceived as Physically Attractive Receive Higher Student Evaluations? *The Journal of General Psychology* **133,** 19–35 (2006).

107. Rivera, L. A. & Tilcsik, A. Scaling Down Inequality: Rating Scales, Gender Bias, and the Architecture of Evaluation. *American Sociological Review* **84,** 248–274 (2019).

108. Rivest, R. L. Bayesian Tabulation Audits: Explained and Extended. eprint: arxiv.org/1801.00528 (2018).

109. Rivest, R. L. ClipAudit: A Simple Risk-Limiting Post-Election Audit. eprint: arxiv.org/1701.08312 (2017).

110. Rivest, R. L. *Reference Implementation Code for Pseudo-Random Sampler for Election Audits or Other Purposes* Last visited 4/8/19. https://people.csail.mit.edu/rivest/sampler.py.

111. Romano, J. P. Bootstrap and Randomization Tests of some Nonparametric Hypotheses. *The Annals of Statistics* **17,** 141–159 (1989).

112. Rose, G. KISS: A Bit Too Simple. Last visited 4/8/19. eprint: https://eprint.iacr.org/2011/007 (2011).

113. Ruben, H. Probability Content of Regions Under Spherical Normal Distributions, II: The Distribution of the Range in Normal Samples. *The Annals of Mathematical Statistics* **31,** 1113–1121 (1960).

114. Rukhin, A. *et al. Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* special publication (NIST, 2010).

115. Saito, M. & Matsumoto, M. in *Monte Carlo and Quasi-Monte Carlo Methods 2006* 607–622 (Springer, Berlin; Heidelberg, 2008).

116. Särndal, C.-E., Swensson, B. & Wretman, J. *Model Assisted Survey Sampling* (Springer-Verlag, New York, 2003).

117. *Shelby County v. Holder* 12-96 (Supreme Court of the United States, 2013).

118. Shevlin, M., Banyard, P., Davies, M. & Griffiths, M. The Validity of Student Evaluation of Teaching in Higher Education: Love Me, Love My Lectures? *Assessment & Evaluation in Higher Education* **25,** 397–405 (2000).

119. Singer, S. & McBurnett, N. *Orange County, CA: Pilot Risk-Limiting Audit* tech. rep. Last visited 12/19/18 (Verified Voting Foundation, 2018). https://www.verifiedvoting.org/wp-content/uploads/2018/12/2018-RLA-Report-Orange-County-CA.pdf.

120. Soto, J. *Statistical Testing of Random Number Generators* in *Proceedings of the 22nd National Information Systems Security Conference* Gaithersburg, Maryland. **10** (1999).

121. Sridhar, M. & Rivest, R. L. *k-Cut: A Simple Approximately-Uniform Method for Sampling Ballots in Post-Election Audits* in *Proceedings Voting'19* St. Kitts (2019).

122. Stahl, J. Georgia Destroyed Election Data Right After a Lawsuit Alleged Its Voting System Was a Mess. Why? *Slate Magazine.* Last visited 1/9/19. https://slate.com/technology/2017/10/georgia-destroyed-election-data-right-after-a-lawsuit-alleged-the-system-was-vulnerable.html (2017).

123. Stanfel, L. E. Measuring the Accuracy of Student Evaluations of Teaching. *Journal of Instructional Psychology* **22,** 117–125 (1995).

124. Stark, P. B. Auditing a Collection of Races Simultaneously. eprint: arxiv/0905.1422v1 (2009).

125. Stark, P. B. *Ballot-Marking Devices (BMDs) Are Not Secure Election Technology* Last visited 3/7/19. 2019. https://www.stat.berkeley.edu/~stark/Preprints/bmd19.pdf.

126. Stark, P. B. Before Reproducibility Must Come Preproducibility. *Nature* **557,** 613 (2018).

127. Stark, P. B. CAST: Canvass Audits by Sampling and Testing. *IEEE Transactions on Information Forensics and Security, Special Issue on Electronic Voting* **4,** 708–717 (2009).

128. Stark, P. B. Conservative Statistical Post-Election Audits. *Annals of Applied Statistics* **2,** 550–581 (2008).

129. Stark, P. B. Risk-Limiting Post-Election Audits: *P*-values from Common Probability Inequalities. *IEEE Transactions on Information Forensics and Security* **4,** 1005–1014 (2009).

130. Stark, P. B. Risk-Limiting Vote-Tabulation Audits: The Importance of Cluster Size. *Chance* **23,** 9–12 (2010).

131. Stark, P. B. Super-Simple Simultaneous Single-Ballot Risk-Limiting Audits. *2010 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '10)* (2010).

132. Stark, P. B. & Freishtat, R. An Evaluation of Course Evaluations. *ScienceOpen Research.* doi:10.14293/S2199-1006.1.-.AOFRQA.v1 (2014).

133. Stark, P. B. & Teague, V. Verifiable European Elections: Risk-Limiting Audits for D'Hondt and Its Relatives. *JETS: USENIX Journal of Election Technology and Systems* **3.1** (2014).

134. Stark, P. B. & Wagner, D. Evidence-Based Elections. *IEEE Security & Privacy* **10,** 33–41 (2012).

135. Stine, R. & Foster, D. *Statistics for Business: Decision Making and Analysis* (Pearson, New York, 2014).

136. Subtirelu, N. "She Does Have an Accent but...": Race and Language Ideology in Students' Evaluations of Mathematics Instructors on RateMyProfessors.com. *Language in Society* **44,** 35–62 (2015).

137. *The American Voting Experience: Report and Recommendations of the Presidential Commission on Election Administration* tech. rep. Last visited 3/7/19 (Presidential Commission on Election Administration, 2014). https://bipartisanpolicy.org/the-presidential-commission-on-election-administration/.

138. Thompson, M. *Theory of Sample Surveys* (Taylor & Francis, London, 1997).

139. Torres, K. Federal Lawsuit Alleges Georgia Blocked Thousands of Minority Voters. *The Atlanta Journal-Constitution.* Last visited 1/11/19. https://www.myajc.com/news/state--regional-govt--politics/federal-lawsuit-alleges-georgia-blocked-thousands-minority-voters/EKb979oRoBe4yJ3Uo1nDfP/ (2016).

140. Tukey, J. W. The Future of Data Analysis. *The Annals of Mathematical Statistics* **33,** 1–67 (1962).

141. Vasilogambros, M. Polling Places Remain a Target Ahead of November Elections. *Stateline.* Last visited 1/11/19. https://pew.org/2MCsiBT (2018).

142. Vitter, J. S. Random Sampling with a Reservoir. *ACM Trans. Math. Softw.* **11,** 37–57 (1985).

143. Voting Machine Maker Hires Former State Election Chief. *The Augusta Chronicle.* Last visited 1/11/19. https://www.augustachronicle.com/article/20061224/NEWS/312249946 (2006).

144. Wagner, N., Rieger, M. & Voorvelt, K. Gender, Ethnicity and Teaching Evaluations: Evidence from Mixed Teaching Teams. *ISS Working Paper Series/General Series* **617,** 1–32 (2016).

145. Wald, A. Sequential Tests of Statistical Hypotheses. *Annals of Mathematical Statistics* **16,** 117–186 (1945).

146. Warner, M. Machine Politics In the Digital Age. *The New York Times.* Last visited 1/11/19. https://www.nytimes.com/2003/11/09/business/machine-politics-in-the-digital-age.html (2003).

147. Weiss, L. A Sequential Test of the Equality of Probabilities in a Multinomial Distribution. *Journal of the American Statistical Association* **57,** 769–774 (1962).

148. Whitesides, J. Polling Places Become Battleground in U.S. Voting Rights Fight. *Reuters.* Last visited 1/11/19. https://www.reuters.com/article/us-usa-election-vote-precincts-insight-idUSKCN11M0WY (2016).

149. Williams, V. Georgia Groups Call on GOP Gubernatorial Nominee Brian Kemp to Step Down as the State's Elections Chief. *The Washington Post.* Last visited 1/15/19. https://www.washingtonpost.com/news/powerpost/wp/2018/08/08/georgia-groups-call-on-gop-gubernatorial-nominee-brian-kemp-to-step-down-as-the-states-elections-chief/ (2018).

150. Wilson, G. *Software Carpentry: Lessons Learned* version 2. Last visited 2/4/19. http://f1000research.com/articles/3-62/v2.

151. Wilson, G. *et al.* Best Practices for Scientific Computing. *PLoS Biology* **12.** doi:10.1371/journal.pbio.1001745 (2014).

152. Wofford, B. How to Hack an Election in 7 Minutes. *POLITICO Magazine.* Last visited 1/10/19. https://politi.co/2K2OGOv (2016).

153. Wolbring, T. & Riordan, P. How Beauty Works. Theoretical Mechanisms and Two Empirical Applications on Students' Evaluation of Teaching. *Social Science Research* **57,** 253–272 (2016).

154. Young, D. H. Two Alternatives to the Standard χ^2-Test of the Hypothesis of Equal Cell Frequencies. *Biometrika* **49,** 107–116 (1962).

155. Zetter, K. Did E-Vote Firm Patch Election? *Wired.* Last visited 1/11/19. https://www.wired.com/2003/10/did-e-vote-firm-patch-election/ (2003).

156. Zetter, K. Georgia Voting Irregularities Raise More Troubling Questions About the State's Elections. *POLITICO.* Last visited 2/13/19. https://politi.co/2SOlvas (2019).

157. Zetter, K. The Crisis of Election Security. *The New York Times.* Last visited 1/9/19. https://www.nytimes.com/2018/09/26/magazine/election-security-crisis-midterms.html (2018).

158. Zetter, K. Was Georgia's Election System Hacked in 2016? *POLITICO Magazine.* Last visited 1/9/19. https://politi.co/2moAWUS (2018).

159. Zetter, K. Will the Georgia Special Election Get Hacked? *POLITICO Magazine.* Last visited 1/9/19. http://politi.co/2heBRW2 (2017).