Statistical Evidence and Election Integrity

Philip B. Stark

Department of Statistics
University of California, Berkeley

XXIX Foro Internacional de Estadistica
UPAEP
Puebla, Mexico
29 September – 3 October 2014

Elections are subject to many kinds of error, and even the best-run elections will almost certainly get vote totals wrong by at least a little. That doesn't undermine democracy as long as the announced winners really did win. Statistics can help produce convincing evidence that the announced winners are the true winners–if there is a sufficiently complete and accurate audit trail. A risk-limiting audit is a statistical technique that examines portions of a voter-verifiable audit trail in a way that guarantees a large, known chance of leading to a complete hand count of the trail, if that hand count would reveal that the voting system found the wrong winners. Risk-limiting audits have been endorsed by the American Statistical Association and many U.S. groups concerned with election integrity, including Common Cause and Verified Voting, and they are being considered for use in some African, Asian, and European countries. Risk-limiting audits have been conducted in California, Colorado, and Ohio.

I will present several methods for risk-limiting audits, including audits of plurality contests (300 of the 500 seats in the Cámara de Diputados and 96 of the 128 seats in the Cámara de Senadores) and proportional representation contests (200 seats in the Cámara de Diputados and 32 in the Cámara de Senadores). The methods are connected to sequential nonparametric tests about the mean of a finite, bounded population. I will also discuss tradeoffs among efficiency, simplicity, transparency, and privacy.

Joint with: Mark Lindeman, Carsten Schürmann, Vanessa Teague, Vince Yates, and many others. Special thanks to all the elections officials who have let me play in their backyards.

All vote counting methods can make mistakes

- Internationally, most concerns are with electronic vote tabulation, but hand counting errs, too.
- Some countries count votes by hand, twice or more.
- Can we save effort and assure accuracy by auditing?
- What roles could audits play in elections in Mexico and other countries?

What do we want an audit to do?

Quality control in general.

Ensure that the electoral outcome is correct;
If outcome is wrong, correct it before it's official.

Outcome means the set of winners, not exact counts.

How can an audit correct a wrong outcome?

- If there's an adequately accurate audit trail, the audit could count all the votes by hand (again).
- Want to correct the outcome if it is wrong, but to do as little counting as possible when the outcome is right.
- Use statistical techniques to decide whether you have checked enough.
- "Intelligent" incremental recount: stop when there's strong evidence that there is no point continuing.

Why not just count all votes by hand (repeatedly)?

- Unnecessarily expensive and slow; accuracy decreases with fatigue.
- Instead, make a first count, then check a random sample.
- Keep checking until there's convincing evidence that the outcome is right—or until all ballots have been hand counted.
- Fatigue, staff quality, etc., may make a full hand count less accurate than a focused audit of a small random sample.
- An audit of hundreds or thousands of ballots can be more transparent than a full count: Public could actually observe the whole process.

Risk-Limiting Audits

- Endorsed by: U.S. Presidential Commission on Election Administration, American Statistical Association, Common Cause, Verified Voting Foundation, Citizens for Election Integrity Minnesota, et al.
- Mandated in law in California (AB2023, SB360) and Colorado
- Piloted in California, Colorado, Ohio—and almost in Denmark
- Rely on manual inspection of a random sample of ballots
- Audit stops when there's strong evidence that the outcome is correct
- Guaranteed big chance of correcting wrong outcomes, no matter why the outcome is wrong
- Use statistical methods to keep the workload low when outcome is right

"Stirring" is key to reducing work

- Don't have to climb into the bathtub to tell if it's hot: can just stick your toe in—if the water is stirred well.
- Don't have to drink a whole pot of soup to tell if it's too salty: a teaspoon is enough—if the pot has been stirred. (Doesn't matter whether the pot holds 0.5ℓ or 100ℓ.)

How do you stir ballots?

Random sampling is stirring

- Imagine numbering the ballots.
- Write the numbers on ping-pong balls; put in a lotto machine.
- Lotto machine stirs the balls and spits some out.
- The ballots with the numbers on the selected balls are a random sample of ballots.
- Easier to stir balls than ballots. Even easier to generate random numbers.
- Still amounts to putting ballots into a huge mixer to stir them, then taking a "teaspoon" of ballots.

Requirements

- Requires sound procedures for protecting, tracking, and accounting for ballots.
- In US, ballot accounting is uneven.
- I don't know how Mexico accounts for ballots.
- New requirement: *ballot manifests*.
- Calculations are simple; web tools are available: `www.stat.berkeley.edu/~stark/Vote/auditTools.htm`, `www.stat.berkeley.edu/~stark/Vote/ballotPollTools.htm`
- Public ritual (including dice rolling) adds transparency and trust

Ballot-polling Audits and Comparison Audits

- Ballot polling audit: sample ballots until there is strong evidence that looking at all of them would show the same election outcome.
  Like an exit poll—but of ballots, not voters.
- Comparison audit:
  1. Commit to vote subtotals, ideally, individual ballot interpretations
     (equivalent: commit to manifest of sorted, counted bundles)
  2. Check that the subtotals add up exactly to contest results
  3. Check subtotals by hand until there is strong evidence the outcome is right

Tradeoffs

- Ballot polling audit
  - Virtually no set-up costs
  - Requires nothing of voting system
  - Requires more counting than ballot-level comparison audit
  - Does not check tabulation: outcome could be right because errors cancel
- Comparison audit
  - Heavy demands for reporting
  - Requires commitment to subtotals
  - Requires retrieving ballots that correspond to subtotals
  - Checks tabulation
  - Ballot-level comparison audits require least hand counting

Both need *ballot manifest*.

Auditing which candidates in a party are seated

- Possible to audit this simultaneously, using the same sample.
- If a small number of votes separates two candidates in a party, required sample size may be very large.
- If ballots are sorted by party and candidate and there's a manifest, can reduce sample sizes substantially.
- Ballot-level comparison audits have much smaller sample sizes than ballot-polling audits when margins are small.
- $\exists$ sequential statistical methods for comparison audits as well.

Auditing Danish Elections—almost

- Joint work with Carsten Schuermann, ITU DK
- Risk-limiting audit of Danish portion of EU Parliamentary election and Danish national referendum on patent court
- Use nonparametric sequential test of hypothesis that outcomes are wrong
- Risk limit 0.1% (99.9% confidence that outcome is right)
- ≈4.6 million ballots, 98 jurisdictions, 1396 polling places
- SRS of 1903 ballots from EU race, 60 from referendum

1. first risk-limiting audit conducted at 99.9% confidence (the highest previously was 90%)
2. first risk-limiting audit of a parliamentary election
3. first risk-limiting audit of a national contest
4. first risk-limiting audit that crossed jurisdictional boundaries
5. first risk-limiting audit outside the U.S.A.
6. first risk-limiting audit of a hand-counted election
7. first risk-limiting audit to use sort-and-stack as a commitment to ballot interpretation
8. smallest margin ever audited with a risk-limiting audit (0.34%)
9. largest contests ever audited with a risk-limiting audit (2.3 million ballots in each contest, 4.6 million total)
10. largest sample ever audited in a ballot-level risk-limiting audit ($>$1900 individual ballots)

Towards reproducible social science

- Verified underlying theorems and checked formulae; currently in peer review
- Coded all algorithms twice, once in ML and once in Python
- ML provably correct; written (partly) using pair programming
- Tested both implementations independently
- Compared output to validate
- Some crucial pieces also in HTML5/Javascript, on the web
- Entire analysis is in an IPython notebook *and* an ML program
- Data are official election results; some web scraping
- All code and data in a git repo
- Photo-documentation of part of the process, including generating seed with dice

2014 Danish EU Parliamentary Election

Party	Votes	% valid votes	seats
A. Socialdemokratiet	435,245	19.1%	3
B. Radikale Venstre	148,949	6.5%	1
C. Det Konservative Folkeparti	208,262	9.1%	1
F. SF - Socialistisk Folkeparti	249,305	11.0%	1
I. Liberal Alliance	65,480	2.9%	0
N. Folkebevaegelsen mod EU	183,724	8.1%	1
O. Dansk Folkeparti	605,889	26.6%	4
V. Venstre, Danmarks Liberale Parti	379,840	16.7%	2
total valid ballots	2,276,694		
blank ballots	47,594		
other invalid ballots	7,929		
total invalid ballots	55,523		
Total ballots	2,332,217		
Eligible voters	4,141,329		
Turnout	56.32 %		

`http://www.dst.dk/valg/Valg1475795/valgopg/valgopgHL.htm` (last accessed 29 May 2014)

21-4 Danish Unified Patent Court membership referendum

yes	1,386,881	62.5%
no	833,023	37.5%
valid votes	2,219,904	
blank ballots	77,722	
other invalid votes	6,157	
total invalid votes	83,879	
total ballots	2,303,783	
eligible voters	4,124,696	
turnout	55.85%	

http://www.dst.dk/valg/Valg1475796/valgopg/valgopgHL.htm (last accessed 29 May 2014)

Initial sample size

Contest information

Ballots cast in all contests: 2303783 Smallest margin (votes): 553,858. Diluted margin: 24.04%.

Contest 1. Contest name: 2014 Denmark EU Patent Referendum

Winners: 1

Reported votes:

Candidate 1 Name: yes Votes: 1386881

Candidate 2 Name: no Votes: 833023

Add candidate to contest 1 | Remove last candidate from contest 1

Add contest | Remove last contest

Audit parameters

Risk limit: 0.1%

Expected rates of differences (as decimal numbers):

Overstatements. 1-vote: 0 2-vote: 0

Understatements. 1-vote: 0 2-vote: 0

Starting size

☑ Round up 1-vote differences. ☐ Round up 2-vote differences. Calculate size 60.

```
(* elect M = M'

   Invariant:
   Let i be the row with the highest non elected coefficient.
   M' = M where line i is altered,
   where the new "blue" eleement is the first hopeful element in xi.
*)

fun elect M =
    let
      val xs = enum 0 (map (fn (_, (x, _) :: _) :: _) => x) M)
      val (j, y) = findmax xs
    in
      elect' j M
    end
and elect' 0 ((_, (r, d) :: R) :: Ps) = ((SOME ((r, d)), R) :: Ps)
  | elect' n (P :: Ps) = P :: (elect' (n-1) Ps)

(* dhondt M n = M'

   Invariant :  M' is teh result of electing n candidates in M
*)
fun dhondt M 0 = M
  | dhondt M n = dhondt (elect M) (n-1)

(* Computing inequalities a la Philips talk.  *)
```

IP[y]: Notebook danmark14EU Last Checkpoint: May 28 07:06 (unsaved changes)

File Edit View Insert Cell Kernel Help

In [2]:

```
%matplotlib inline
import math
import numpy as np
import scipy
from scipy.stats import binom
import pandas as pd
import matplotlib.pyplot as plt
#
def dHondt(partyTotals, seats, divisors):
    '''
    allocate <seats> seats to parties according to <partyTotals> votes,
    using D'Hondt proportional allocation with <weights> divisors
    '''
    pseudoCandidates = np.array([partyTotals,]*seats, ).T/divisors.astype(float)
    sortedPC = np.sort(np.ravel(pseudoCandidates))
    lastSeated = sortedPC[-seats]
    theSeated = np.where(pseudoCandidates >= lastSeated)
    partySeats = np.bincount(theSeated[0], minlength=len(partyTotals)) # number of seats for each party
    inx = np.nonzero(partySeats)[0]  # only those with at least one seat
    seated = zip(inx, partyTotals[inx], divisors[partySeats[inx]-1])
                                       # parties with at least one seat,
                                       # number of votes that party got,
                                       # and divisor for last seated in the party,
    theNotSeated = np.where(pseudoCandidates < lastSeated)
    partyNotSeats = np.bincount(theNotSeated[0], minlength=len(partyTotals)) # number of non-seats for each party
    inx = np.nonzero(partyNotSeats)[0]
    notSeated = zip(inx, partyTotals[inx], divisors[partySeats[inx]])
                                       # parties with at least one unseated,
                                       # number of votes that party got,
                                       # and divisor for the first non-seated in the party
    if (lastSeated == sortedPC[-(seats+1)]):
        raise InputError("Tied contest for the last seat!")
    else:
        return partySeats, seated, notSeated, lastSeated, pseudoCandidates
```

Statistical formulation of RLAs

Hypothesis Test

Null: outcome is wrong (one or more apparent winners really lost)
Alternative: outcome is right

Reject null $\rightarrow$ conclude outcome is right.
Maximum significance level is the *risk*.
Maximum is over all ways the outcome could be wrong.

Sequential Testing

- Collect data until there's strong evidence that the outcome is right (or until there's a full hand count).
- Need to account for sequential data collection
- Strategy: express sufficient condition in terms of scalar properties of population of cast ballots

Parameters and Statistics

- Ballot polling: for each pair, difference in weighted tallies.
- Comparison: maximum relative overstatement of pairwise margins.
- Both reduce to nonparametric hypothesis that the mean of a finite, bounded, nonnegative population is ≥ 1.
- Surprisingly little work on "simple" problem.
- "Best" test so far is based on Wald's (1945) sequential probability ratio test

Divisors for common "highest averages" methods

Name	**used in**	**d(1)**	**d(2)**	**d(3)**	**d(4)**
D'Hondt	Belgium Denmark Luxembourg	1	2	3	4
Modified D'Hondt	Estonia	$1^{0.9}$ 1	$2^{0.9}$ 1.866	$3^{0.9}$ 2.688	$4^{0.9}$ 3.482
Sainte-Laguë	Germany	1	3	5	7
Modified Sainte-Laguë	Norway	1.4	3	5	7

party p	$t(p)/d(1)$	$t(p)/d(2)$	$t(p)/d(3)$	$t(p)/d(4)$
1	100,000	50,000	33,333	25,000
2	60,000	30,000	20,000	15,000
3	40,000	20,000	13,333	10,000
4	30,000	15,000	10,000	7,500
5	25,000	12,500	8,333	6,250

Hypothetical results for contest with $S = 4$ seats, $P = 5$ parties.

$t(p)$ is reported count for party p.
$d(s)$ is the divisor for column s; here $d(s) = s$ (D'Hondt).
$a(p)$ is actual (i.e., perfect) count for party p.

party p	$t(p)/1$	$t(p)/2$	$t(p)/3$	$t(p)/4$
1	100,000	50,000	33,333	25,000
2	60,000	30,000	20,000	15,000
3	40,000	20,000	13,333	10,000
4	30,000	15,000	10,000	7,500
5	25,000	12,500	8,333	6,250

Apparent winning "pseudo candidates," $S = 4$ seats, $P = 5$ parties

party p	$t(p)/1$	$t(p)/2$	$t(p)/3$	$t(p)/4$
1	100,000	50,000	33,333	25,000
2	60,000	30,000	20,000	15,000
3	40,000	20,000	13,333	10,000
4	30,000	15,000	10,000	7,500
5	25,000	12,500	8,333	6,250

Seat allocation is correct if, for the true tallies $a(p)$ (not just reported tallies $t(p)$) every blue cell is greater than every red cell

party p	$a(p)/1$	$a(p)/2$	$a(p)/3$	$a(p)/4$
1		$a(1)/2$	$a(1)/3$	
2	$a(2)$	$a(2)/2$		
3	$a(3)$	$a(3)/2$		
4	$a(4)$			
5	$a(5)$			

Audit needs to check that each
blue cell $>$ every red cell in all *other* rows.

Remaining inequalities guaranteed arithmetically.

$$
\begin{aligned}
B &: \#\text{ ballots cast in the contest} \\
V &: \#\text{ votes per ballot each voter is allowed to cast} \\
P &: \#\text{ parties} \\
S &: \#\text{ seats to be assigned} \\
C_p &: \#\text{ candidates in party } p \\
t(p) &: \text{reported total votes for party } p \\
a(p) &: \text{actual total votes for party } p \\
e(p) &\equiv t(p) - a(p),\ \text{ error reported vote for party } p \\
t(p,c) &: \text{reported total votes for candidate } c \text{ in party } p \\
a(p,c) &: \text{actual total votes for candidate } c \text{ in party } p \\
e(p,c) &\equiv t(p,c) - a(p,c),\ \text{ error in reported vote for candidate } c \text{ in party } p \\
d(s) &: \text{divisor for column } s \\
p_{ps} &\equiv t(p)/d(s) \\
\pi_{ps} &\equiv a(p)/d(s) \\
\mathcal{W} &: \text{pairs } (p,s) \text{ with the } S \text{ largest values of } p_{ps} \\
\mathcal{L} &: \text{pairs } (p,s),\ p = 1, \ldots, P,\ s = 1, \ldots, S \text{ not in } \mathcal{W} \\
\mathcal{W}^P &: \text{parties } p \text{ that (apparently) won at least one seat} \\
\mathcal{L}^P &: \text{parties } p \text{ that (apparently) lost at least one seat} \\
\mathcal{W}_p &: \text{candidates } c \text{ in party } p \text{ who were seated} \\
\mathcal{L}_p &: \text{candidates } c \text{ in party } p \text{ who were not seated}
\end{aligned}
$$

Pseudo-candidates

- $P \times S$ pairs (p, s) of *pseudo-candidates*.
- Candidate (p, s) reported to have received $p_{ps} = t(p)/d(s)$ votes.
- Candidate (p, s) actually received $\pi_{ps} = a(p)/d(s)$ votes.
- $\mathcal{W}$ are "apparent winners" according to reported tally.
- *apparent outcome*: # seats each party gets according to reported totals $t(p)$, $p = 1, \ldots, P$.
- *true outcome*: # seats each party would get according to true totals $a(p)$, $p = 1, \ldots, P$.
- apparent outcome is correct iff

$$\forall (p_w, s_w) \in \mathcal{W}, \; \forall (p_\ell, s_\ell) \in \mathcal{L}, \; \pi_{p_w s_w} > \pi_{p_\ell s_\ell}. \tag{1}$$

Which need checking?

For party p, define

$$\begin{aligned} s_w(p) &\equiv \max\{s : (p, s) \in \mathcal{W}\} \\ s_\ell(p) &\equiv \min\{s : (p, s) \in \mathcal{L}\}. \end{aligned}$$

These are the column indices of the last seat party p wins and the first seat party p loses, respectively. One or the other might not exist for a particular party p, if it won no seats or all S seats; at most $\min(2P, S + P)$ exist. Define

$$\begin{aligned} \mathcal{W}^P &\equiv \{p : \exists s \text{ s.t. } (p, s) \in \mathcal{W}\} \\ \mathcal{L}^P &\equiv \{p : \exists s \text{ s.t. } (p, s) \in \mathcal{L}\}. \end{aligned}$$

Audit to check whether

$$\forall p \in \mathcal{W}^P,\ \forall q \in \mathcal{L}^P \text{ s.t. } p \neq q,\ \pi_{p,s_w(p)} > \pi_{q,s_\ell(q)}. \qquad (2)$$

Wald's sequential probability ratio test

- Sequence of IID trials
- If null H_0 is true, chance of "success" is γ_0
- If alternative H_1 is true, chance of "success" is γ_1
- Set $T = 1$
- Repeat:
  - conduct trial
  - if "succeed," $T \rightarrow T \times \gamma_1/\gamma_0$
  - if "fail," $T \rightarrow T \times (1 - \gamma_1)/(1 - \gamma_0)$
  - if $T > 1/\alpha$, reject H_0 at significance level α; stop.

Ballot-polling audit: derivation

- pair of pseudo-candidates $(p_w, s_w) \in \mathcal{W}$, $(p_\ell, s_\ell) \in \mathcal{L}$
- want to determine whether $\pi_{p_w s_w} > \pi_{p_\ell s_\ell}$
- i.e., $a(p_w)/d(s_w) > a(p_\ell)/d(s_\ell)$
- i.e., $a(p_w) > a(p_\ell)\frac{d(s_w)}{d(s_\ell)}$

Ballot-polling audit: derivation

- A_p: event that a randomly selected ballot shows a vote for party p.
- $\Pr(A_p) = a(p)/B$
- If outcome is correct,

$$\Pr(A_{p_w}) \geq \frac{d(s_w)}{d(s_\ell)} \Pr(A_{p_\ell}),$$

so

$$\Pr(A_{p_w} | A_{p_w} \cup A_{p_\ell}) \geq \frac{d(s_w)}{d(s_\ell)} \Pr(A_{p_\ell} | A_{p_w} \cup A_{p_\ell}),$$

- For the outcome to be correct, need

$$\begin{aligned} \pi_{p_w|p_w p_\ell} &> (1 - \pi_{p_w|p_w p_\ell}) d(s_w)/d(s_\ell) \\ \textit{i.e.,} \quad \pi_{p_w|p_w p_\ell} &> \frac{d(s_w)}{d(s_\ell) + d(s_w)}. \end{aligned}$$

Derivation, contd.

$$\pi_{p_w|p_w p_\ell} \equiv \frac{a(p_w)}{a(p_w) + a(p_\ell)}$$

and

$$\frac{t(p_w)}{t(p_w) + t(p_\ell)} > \frac{d(s_w)}{d(s_\ell) + d(s_w)}.$$

Derivation, contd.

- Use Wald's sequential probability ratio test to test H_0:

$$\frac{a(p_w)}{a(p_w)+a(p_\ell)} \leq \frac{d(s_w)}{d(s_\ell)+d(s_w)}$$

against H_1:

$$\frac{a(p_w)}{a(p_w)+a(p_\ell)} \geq \frac{t(p_w)}{t(p_w)+t(p_\ell)}.$$

- Rejecting H_0 confirms $\pi_{p_w s_w} > \pi_{p_\ell s_\ell}$.

Derivation, contd.

- For single draw, conditional on $A_{p_w} \cup A_{p_\ell}$, if the ballot shows a vote for p_w,

$$\mathrm{LR} = \frac{\frac{t(p_w)}{t(p_w)+t(p_\ell)}}{\frac{d(s_w(p_w)}{d(s_w(p_w))+d(s_\ell(p_\ell))}}.$$

- If the ballot shows a vote for p_ℓ,

$$\mathrm{LR} = \frac{1 - \frac{t(p_w)}{t(p_w)+t(p_\ell)}}{1 - \frac{d(s_w(p_w)}{d(s_w(p_w))+d(s_\ell(p_\ell))}}$$

Ballot-polling audit: algorithm

1. Select the risk limit $\alpha \in (0, 1)$, and M, the maximum number of ballots to audit before proceeding to a full hand count. Define

$$\gamma^{+}_{ps_w(p)qs_\ell(q)} \equiv \frac{t(p)}{t(p)+t(q)} \cdot \frac{d(s_w(p)) + d(s_\ell(q))}{d(s_w(p))}$$

and

$$\begin{aligned}\gamma^{-}_{ps_w(p)qs_\ell(q)} &\equiv \left(1 - \frac{t(p)}{t(p)+t(q)}\right) \times \\ &\quad \times \left(1 - \frac{d(s_w(p)) + d(s_\ell(q))}{d(s_w(p))}\right).\end{aligned}$$

Set $T_{ps_w(p)qs_\ell(q)} = 1$ for all $p \in \mathcal{W}^P$ and $q \in \mathcal{L}^P$. Set $m = 0$.

2. Draw a ballot uniformly at random with replacement from those cast in the contest and increment m.

Ballot-polling audit: algorithm

3 If the ballot shows a valid vote for a reported winner $p \in \mathcal{W}^P$, then for each q in $\mathcal{L}^P$ that did not receive a valid vote on that ballot multiply $T_{ps_w(p)qs_\ell(q)}$ by $\gamma^+_{ps_w(p)qs_\ell(q)}$. Repeat for all such p.

4 If the ballot shows a valid vote for a reported loser $q \in \mathcal{L}^P$, then for each p in $\mathcal{W}^P$ that did not receive a valid vote on that ballot, multiply $T_{ps_w(p)qs_\ell(q)}$ by $\gamma^-_{ps_w(p)qs_\ell(q)}$. Repeat for all such q.

5 If any $T_{ps_w(p)qs_\ell(q)} \geq 1/\alpha$, reject the corresponding null hypothesis for each such $T_{ps_w(p)qs_\ell(q)}$. Once a null hypothesis is rejected, do not update its $T_{ps_w(p)qs_\ell(q)}$ after subsequent draws.

6 If all null hypotheses have been rejected, stop the audit: The reported results stand. Otherwise, if $m < M$, return to step 2.

7 Perform a full hand count; the results of the hand count replace the reported results.

Comparison audits

- In comparison audits, check whether any margin for any (winner, loser) pair can be accounted for by error.
- After re-scaling—and by using probability proportional to size sampling—can transform this into the problem of sequential one-sided tests for the mean of a finite, bounded population. The audit stops (only) when the hypothesis that the error is not large enough to alter the outcome is rejected.
- Presenting rigorous, conservative (not approximate) techniques to solve that problem in the short course yesterday, today, and tomorrow morning.

Multiplicity

Even though we are performing *many* pairwise tests, multiplicity isn't an issue: the audit proceeds to a full hand count if *any* of the null hypotheses is not rejected.

For any collection of true null hypotheses, the chance that all are erroneously rejected is less than the chance that any individual one is erroneously rejected, which is guaranteed to be at most α.

Countries in conversation about risk-limiting audits

- Denmark
- Luxembourg
- Mongolia
- Nigeria

In the U.S.: Ohio, California, Colorado.

More reading:

- http://www.coloradostatesman.com/content/995064-making-sure-votes-count
- http://arstechnica.com/tech-policy/2012/07/

  saving-american-elections-with-10-sided-dice-one-stats-profs-quest/
- http://www.huffingtonpost.com/american-statistical-association/

  leave-election-integrity-_b_3580649.html
- Stark, P.B., and D.A. Wagner, 2012. *IEEE Computing Now, 10*, 33–41. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6203498

  preprint: http://www.stat.berkeley.edu/~stark/Preprints/evidenceVote12.pdf
- Lindeman, M. and P.B. Stark, 2012. A Gentle Introduction to Risk-Limiting Audits. *IEEE Computing Now, 10*, 42–49. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6175884&tag=1

  preprint: http://www.stat.berkeley.edu/~stark/Preprints/gentle12.pdf