N = 100,000; α = .05

Stopping Probabilities for new rounds: findAsn([.4, .8, 1.3, 2.0, 3.0])

Convolution Approach (Uniform)	Round 1	Round 2	Round 3	Round 4	Round 5	Full Hand Count	Average Number of Ballots Examined[1]	findAsn (from rlacalc.py, used to determine the round schedule)
1%	0.36128	0.49143	0.14724	N/A[2]	N/A[2]	0.00005	43758	60014
2%	0.22897	0.33822	0.28184	0.13272	0.01810	0.00015	15766	15028
5%	0.21024	0.30780	0.26952	0.15022	0.05323	0.00899	3474	2416
10%	0.18839	0.32353	0.27338	0.14857	0.05774	0.00839	1541	608
20%	0.15385	0.38119	0.23666	0.17489	0.04408	0.00933	1108	154
50%	N/A[3]	0.41482	0.33428	0.20073	0.04747	0.0027	300	25

Motivation: Exploring the effect of round schedules on stopping probabilities and expectations.

1. Expectations are much larger for wider margins because the sum of the stopping probabilities is (around) 99% for some of these margins, not 100%. This difference adds 1% * N = .01 * 100,000 = 1,000 ballots to the expectation.

This suggests that we should desire the sum of the stopping probabilities to be very close to 100%, especially for wider margins, so that the possibility of a full hand count does not inflate the expectation.

2. With this scheme for determining the round schedule, these two rounds would have a sample greater than N = 100,000 ballots. They were nixed, and their allotted errors were given to the other rounds. (Perhaps, for the purposes of comparison, we should just throw away that error instead of reallocating it. I'm unsure)

3. With this scheme for determining the round schedule, the first round would audit 10 ballots. But even all 10 ballots for the winner is insufficient, so there is no k_min for the first round. I therefore nixed that round altogether (giving its allotted error to other rounds). (Perhaps, for the purposes of comparison, we should just throw away that error instead of reallocating it. I'm unsure)

PROCEDURE: The convolutionaudit.py code was run with the calculated round schedule, which differs for each row of the table. That round schedule and the outputted k_mins are then inputted into stoppingprobabilities.py, which delivers the data seen here.

Tidbit: There appears to be no nice structure à la the first quantile always being 41% of the mean. One way to look at it is by noting that, in the above table, the 10% and 20% rows have the first quantile .8 * 608 = 486 and .8 * 154 = 123.2, respectively, but clearly 486/1541 != 123.2/1108.

One might also observe that one audit round services many percentiles. For example, in the 10% row, the 16th through 16+38 = 54th percentiles are all .8 * 608 = 486 ballots, so the 16th through 54th percentiles are all the same fraction of the mean.

Averages for different round schedules

Average Number of Ballots Examined	[200, 400, …, 51200]	[.4 * findAsn, .8 * findAsn, …, 3.0 * findAsn]
1%	47717	43758
2%	17911	15766
5%	3498	3474
10%	955	1541
20%	304	1108
50%	200	300

N = 100,000; α = .05; β = 0
Stopping Probabilities for new rounds: findAsn([.4, .8, 1.3, 2.0, 3.0])

Traditional RLA without replacement[1]	Round 1	Round 2	Round 3	Round 4	Round 5	Full Hand Count	Average Number of Ballots Examined	findAsn (from rlacalc.py, used to determine the round schedule)
1%	0.21206	0.50319	0.27934	N/A	N/A	0.00541	51582	60014
2%	0.14420	0.33028	0.28057	0.17456	0.06380	0.00659	19098	15028
5%	0.13062	0.29056	0.25315	0.18378	0.09556	0.04633	7693	2416
10%	0.12644	0.27534	0.26727	0.17520	0.10145	0.0543	6200	608
20%	0.15385	0.24990	0.26857	0.18680	0.09080	0.05008	5199	154
50%	N/A	0.41482	0.33428	0.12497	0.08031	0.04562	4591	25

1. Since β = 0, this is also the BRAVO audit without replacement.

PROCEDURE:
The round schedule was calculated, using the findAsn function.
Formula 5 from BRLA-I was used to determine the associated k_mins. (This differs slightly from if one used the SimulateBayesianWithPrior code, as that requires α = β.) The round schedule and k_mins were inputted then inputted into stoppingprobabilities.py.

These are both supposed to be averages of the traditional RLA, so one might wonder why these two columns are (so) different.

The left column is the average number of ballots if the audit if the audit is conducted in rounds, only making the decision to stop (or not) at the end of each round. The right column assumes the audit is really carried out ballot-by-ballot, and is able to stop after each ballot examined.

Additionally, it appears that findAsn numbers reflect the traditional RLA *with* replacement. The left column, meanwhile, reflects the traditional RLA *without* replacement (which appears to be a better benchmark as it is more demanding).

Also, note the huge discrepancy. This is explained by the fact that the sum of the stopping probabilities for the wider margins is around 95%, and so the possibility of a full hand count contributes around 5000 to the expectation on the left.

Averages for different round schedules

Average Number of Ballots Examined	[200, 400, …, 51200]	[.4 * findAsn, .8 * findAsn, …, 3.0 * findAsn]
1%	N/A	51582
2%	70174	19098
5%	4091	7693
10%	1055	6200
20%	312	5199
50%	200	4591