An Analysis of the Western States Lottery

Ryan Witko
January 23, 2016

Abstract

The Western States Endurance Run is America's oldest and most prestegious 100-mile trail ultramarathon. But long before runners can conquer 100 miles of mountains, canyons and river crossings, they must first conquer the "*2^(n-1)*" lottery system used to select just 270 lucky entrants to toe the starting line in Squaw Valley. This paper provides a quantitative analysis of several key features of this lottery system. First, we will examine the simulation methods used by the race organizers to predict selection odds for each entrant prior to the drawing. The following section will demonstrate a new algorithm to directly compute the odds of selection without resorting to simulations. In addition to being significantly faster and more accurate than the existing methodology, this new method provides the framework for much of the analysis in subsequent sections, starting with a look at the likelihood of drawing duplicate tickets as the lottery progresses through all 270 rounds. We again provide an algorithm to directly compute the expected frequency of duplicates without the need of simulation. Following this, we propose a model for quantifying the collective surprise and disappointment felt by lottery participants after the drawing and show how simple changes in the lottery structure can alter the aggregate happiness of the participants as a whole. Our final section forecasts future growth in the Western States lottery pool and examines the impact of this growth on the expected waiting time of new applicants. An appendix provides complete R code for the algorithms described in the paper.

Ryan Witko received his BA in Mathematics from New York University in 2005. He has worked at BlackRock Financial Management in New York since 2006. As of January 2016, he has completed 17 ultras, including the Vermont 100-miler in 2014 and the Tahoe Rim Trail 100-miler in 2015. He will have a whopping 4 tickets in the Western States lottery in 2017.

Email: witko.ryan@gmail.com
Twitter: @ryanwitko

1. Introduction

The Race[1]

The Western States Endurance Run ("Western States") is a 100-mile running race held each June in California. The race's history begins with the Tevis Cup, a 100-mile equestrian endurance ride along the Western States Trail from Tahoe City to Auburn. In 1974, Gordy Ainsleigh, a Tevis Cup rider, decided to attempt the course on foot. Twenty three hours and forty-two minutes later, Gordy arrived in Auburn and proved that 100 miles on foot in one day was possible.

In 1977, the Western States Endurance Run was born, run in conjunction with the Tevis Cup. Runners were monitored at three veterinary stations set up to serve the horses. In 1978, Western States split into a separate event, run a month earlier than the Tevis Cup. In the ensuing decades, the race has grown and matured into one of the most competitive and prestigious running races in the world.

The Western States course ascends over 18,000 feet and descends nearly 23,000 feet as it winds through the Sierras, traversing several steep canyons and fording the American River along the way. Runners finish on the track of Placer High School in Auburn and those finishing in fewer than 24 hours receive the coveted silver belt buckle. The current course record for men stands just under 15 hours and for women just under 17 hours.

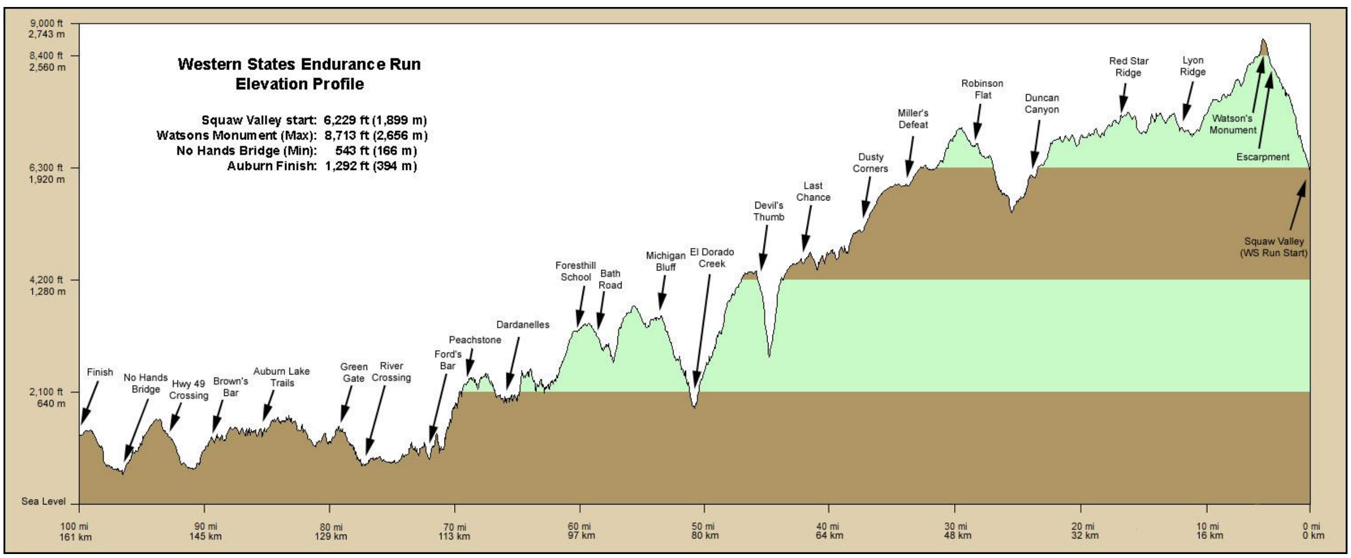

Image 1: *Course profile of Western States Endurance Run (run right to left)*

The Lottery[2]

Given the single-track nature of the course and the delicate surrounding environment, race capacity is strictly limited to about 380 runners each year. Since 1981, the race organizers have utilized a lottery system to select runners to participate in the race. While the mechanics of the lottery have evolved over the years, the race organizers have utilized a "*2^(n-1)*" system since the 2015 drawing[3]. Under

---

[1] For a more detailed history of Western States, see http://www.wser.org/how-it-all-began/
[2] For more details on the mechanics of the lottery, see http://www.wser.org/lottery/
[3] That's 2 to the power of n-1.

this system, each runner who enters the lottery and fails to gain entry will have double the number of tickets in the hat when entering the lottery the following year. In other words, first year applicants will have one ticket, second year applicants will have two tickets, third year applicants will have four tickets, and so on.

In order to apply for the lottery, runners must complete a qualifying race during the preceding 12 months. The list of qualifying races has changed over time, but currently consists of 83 races between 100 kilometers and 135 miles, with the majority being 100-mile trail races.

270 of the 380 race entries are determined via lottery. (The remaining 110 entries are various types of automatic qualifiers.) Prior to the drawing, the race organizers publicly post the ticket counts for all applicants in the lottery. They also post estimated odds of selection for runners with a given number of tickets (shown in Table 1). The drawing is then held before a live audience and streamed in real time online.

Image 2: *The silver belt buckle awarded to finishers under 24 hours*

Of particular relevance to Section 4, it should be noted that the lottery is conducted the “old fashioned” way: namely, thousands of slips of paper in an urn with a human drawing one slip at a time. Another important feature of the lottery is that no applicant can be awarded more than one entry into the race. Therefore, once a particular name is drawn, the remainder of their tickets in the urn are effectively “dead”. (It is not practical to search the urn and remove these “dead” tickets after each round.) For example, if a seventh-year applicant is selected with the first drawing, one ticket is removed from the urn but that entrant’s 63 other tickets remain. Should one of these “dead” tickets get drawn in a subsequent round, the ticket is simply discarded and the round continues until a valid applicant is drawn.

2. How Many Simulations Are Needed?

For runners hoping to gain entry into Western States via the lottery, the natural question to ask is: what are my chances of being selected? Prior to the actual drawing, the race organizers attempt to estimate these odds for each category of runner. They do this by using a Monte Carlo simulation[4] using 100,000 trials. This section will examine if this an adequate number of simulations in order to achieve an acceptable level of accuracy.

Monte Carlo Simulation

Once we know the number of applicants in each category (and therefore the number of tickets for each applicant) we can attempt to quantify the odds of selection of each runner. In particular, there is some "true" distribution describing the likelihood of all possible lottery outcomes. The mean of this distribution is what we seek to compute. However, calculating this "true" distribution is exceedingly complex due to the vast number of possible outcomes. (Below, we will see just how many outcomes are possible.)

Using Monte Carlo simulation, the race organizers can estimate this "true" distribution. In particular, using a computer, a set of 100,000 randomly lottery outcomes was computed. The observed distribution of these outcomes is then used as a proxy for the "true" distribution of outcomes. Provided enough simulations are run, statistics describing this proxy distribution will closely approximate those of the "true" underlying set.[5]

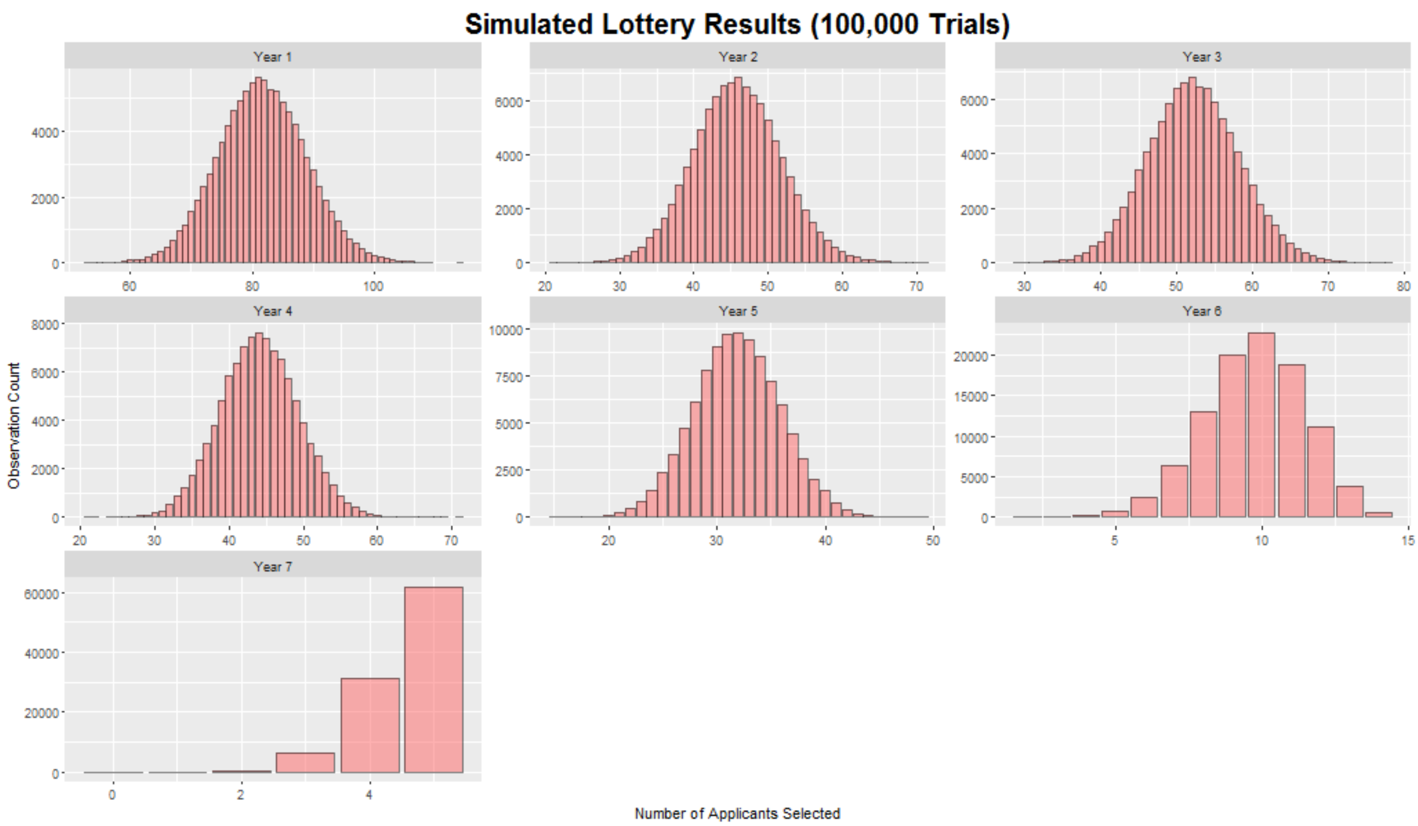

Chart 1: *Simulated distributions by applicant category using 100,000 simulations*

[4] From Wikipedia: Monte Carlo methods are "a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to use other mathematical methods."
[5] By the law of large numbers, the expected value of a random variable can be approximated by taking the empirical mean (i.e. the sample mean) of independent samples of the variable.

Chart 1 shows histograms of the number of applicants selected in each category across a simulation of 100,000 lottery outcomes. (Note that these will vary slightly from the outcomes observed by the race organizers in their simulation since the outcomes are randomly generated.) Looking at the first distribution, we can see that, on average, somewhere around 80 first-year applicants were selected. We observed very few outcomes where fewer than 60 or more than 100 first-year applicants were selected. On the other hand, for the seventh-year applicants, the most frequently-observed outcome was that all 5 applicants were selected. We saw almost no cases where fewer than 3 were selected.

Using empirical distributions like those above, the race organizers compute the odds of selection for runners in each category. Prior to the actual drawing, these odds are posted on the race website for inspection. Table 1 shows this output for the 2016 drawing.

Lottery Odds

Tickets	Entrants	Group Tally		Odds (%)	# Selected
		Count	%		
1	2233	8164903	30.240	3.656	81.6
2	639	4588392	16.994	7.181	45.9
4	377	5217775	19.325	13.840	52.2
8	171	4407801	16.325	25.777	44.1
16	71	3191822	11.822	44.955	31.9
32	14	975322	3.612	69.666	9.8
64	5	453985	1.681	90.797	4.5
Sum	3510	27000000	100.000		270.0

Simulations completed: 100,000

Average number of draws required: 303.1

Minimum number of draws required: 279

Maximum number of draws required: 336

Table 1: *Monte Carlo Summary Statistics for 2016 Drawing*

While these statistics look reasonable at first glance, one may wonder: are 100,000 simulations enough to accurately describe the true distribution of outcomes? Are those odds really accurate to three decimal places?

Total Number of Lottery Outcomes

In order to determine if 100,000 is a sufficient number of simulations, let's get a handle on the number of possible lottery outcomes. After all, if this number is sufficiently small, we could simply enumerate them all and directly compute the precise odds without resorting to Monte Carlo simulation at all.

First, we will compute the number of outcomes where we distinguish between runners. There are 3,510 applicants and 270 slots available. Thus, the total number of distinct outcomes is:

$$Number\ of\ Distinct\ Outcomes\ = \binom{3510}{270} = \frac{3510!}{3240!\,270!} =$$

$$
\begin{aligned}
&6\,257\,541\,505\,014\,500\,667\,147\,238\,991\,422\,653\,150\,996\,598\,323\,637\,539\,844\,752\,640\,074 \ddots \\
&\quad 696\,933\,012\,448\,100\,685\,339\,409\,701\,001\,662\,310\,233\,282\,635\,031\,962\,661\,137\,062\,214 \ddots \\
&\quad 125\,312\,241\,302\,750\,631\,376\,367\,206\,664\,863\,526\,133\,805\,904\,513\,403\,246\,934\,319\,008 \ddots \\
&\quad 756\,620\,765\,611\,995\,147\,518\,999\,513\,380\,943\,192\,629\,318\,571\,243\,303\,549\,247\,700\,326 \ddots \\
&\quad 865\,556\,345\,372\,699\,039\,045\,994\,728\,075\,154\,933\,428\,426\,159\,961\,267\,560\,586\,350\,128 \ddots \\
&\quad 494\,875\,895\,304\,734\,546\,105\,669\,674\,364\,582\,905\,720\,147\,952\,679\,608\,688\,728\,666\,121 \ddots \\
&\quad 805\,081\,735\,081\,999\,283\,455\,322\,935\,742\,650
\end{aligned}
$$

This number has 412 decimal digits. Clearly, in comparison to the total number of possible outcomes, 100,000 simulations seem wholly inadequate. There are trillions upon trillions of outcomes that our simulation simply did not observe. However, this is overstating the situation. Since all applicants in a particular category have the same odds of selection, we need not differentiate between individual applicants, only between different *categories* of applicant.

Number of Distinct Lottery Outcomes

We'll compute the total number of distinct lottery outcomes where we do not distinguish between outcomes that share the same number of runners from each category. For example, we will count only one outcome in which all 270 entries are awarded to first-year applicants, even though there are about 10^{356} different ways to choose 270 different first-year applicants.

It turns out there are exactly 12,705,435,449 such outcomes. (Appendix A contains the details of this calculation.) Thus, if one were to attempt to compute the "true" distribution of outcomes, one would have to consider nearly 13 billion cases. Furthermore, not all of these cases are equally probable, and in fact some are quite rare. (We will see in Section 4 several important but exceedingly rare cases.) Thus, even running 13 billion simulations will not reveal the "true" underlying distribution of outcomes.

In light of these results it would seem that 100,000 is far too few simulations to fully describe the complete distribution of possible outcomes. While that is certainly true, we'll see in the next section that if we relax our requirement of complete precision and require only a given level of precision (say one decimal place) far fewer simulations are required.

Accuracy of 100,000 Simulations

How good of an estimate does 100,000 simulations provide? We can use the Central Limit Theorem to help answer this question. Roughly speaking, this theorem says that as the number of simulations increases, the distribution of observed means behaves like a normal distribution with the variance shrinking in proportion to the number of simulations run. For example, consider the distribution of first-year applicants selected. Let $(X_1, X_2, \ldots, X_n)$ be n random samples from the "true" underlying distribution. Let M_n be the sample mean. In other words,

$$M_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Suppose the underlying distribution has mean μ and standard deviation σ. Then the Central Limit Theorem tells us that as n grows large, M_n converges to a normal distribution with mean μ and standard deviation $\sigma/\sqrt{n}$. In other words, as $n \to \infty$,

$$M_n \overset{d}{\to} N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

Using this tool, we can then use standard properties of the normal distribution to inform us about the expected behavior of M_n as n increases. For example, we know that about 95% of the normal distribution lies within 2 standard deviations of the mean. Thus, taking $n = 100{,}000$ we can compute the width of the 95% confidence interval for each category of applicant.

Consider the distribution of fifth-year applicants selected. This distribution has a mean of 44.931% and standard deviation of 5.659%. Thus, the standard deviation of M_n for this category is

$$\frac{0.05659}{\sqrt{100{,}000}} \approx 0.018\%$$

Thus, we can be 95% confident that the odds we computed using 100,000 simulations is within 0.036% of the "true" odds.

Category	Entrants	Mean μ	Standard Deviation σ	# Trials for 0.001% Accuracy	Accuracy of 100,000 Trials
1	**2,233**	3.656%	0.319%	1,627,532	± 0.002%
2	**639**	7.179%	0.917%	13,455,668	± 0.006%
3	**377**	13.845%	1.592%	40,555,093	± 0.010%
4	**171**	25.779%	3.093%	153,020,309	± 0.020%
5	**71**	44.931%	5.674%	515,169,511	± 0.036%
6	**14**	69.714%	12.232%	2,393,924,927	± 0.077%
7	**5**	90.876%	12.942%	2,680,130,532	± 0.082%

Table 2: *Number of simulations needed to achieve a given level of accuracy*

The final column in Table 2 shows the output of this calculation for each category[6]. We can see that the first-year applicant pool has the lowest standard deviation and therefore has the tightest confidence interval after 100,000 simulations. Conversely, the seventh-year applicant pool had the highest standard deviation and therefore has a confidence interval about 40 times wider than that of the first-year applicants.

We can also compute the number of trials required in order to achieve a given level of accuracy (with a particular level of confidence). The race organizers report the selection odds to three decimal places. Thus, we will compute the number of simulations needed to be 95% confident that we are within 0.0005% of the true odds (and thus accurately round to three decimal places).

Again, consider the fifth-year applicant pool. With $\sigma = 5.674\%$, we seek to solve for n such that two standard deviations of M_n is 0.0005%:

$$\frac{5.674\%}{\sqrt{n}} = \frac{0.0005\%}{2} \Rightarrow n = 515{,}169{,}511$$

[6] This analysis is not fully rigorous since the mean and standard deviation themselves were estimated using 100,000 simulations. In order to properly use the Central Limit Theorem, one needs the "true" underlying mean μ and standard deviation σ. In Section 3 we will compute the precise mean, but computing the precise standard deviation is much more difficult.

Thus, we see that if we run about 515 million simulations, we can be 95% sure that the odds we observe will be within 0.0005% of the true odds (and thus will accurately round to three decimal places). Clearly, 100,000 simulations falls far short of this threshold.

On the other hand, looking again at the final column of Table 2 reveals that we are within 0.082% in all categories after only 100,000 simulations. Thus, reporting odds to just one decimal place is fully justified.

Chart 2 shows the evolution of M_n for the first-year and fourth-year applicant pools as n increases from 1 to 100,000. The light-grey horizontal band indicates one decimal place of precision around the "true" mean. The thinner, dark-grey band indicates two decimal places of precision. We can see that for the first-year applicants, the estimated odds are almost immediately within one decimal place of precision. Furthermore, the odds stay within two decimal places after just about 10,000 simulations. On the other hand, for the forth-year applicant pool, we can see that the estimated odds have not achieved two decimal places of precision by the 100,000th simulation.

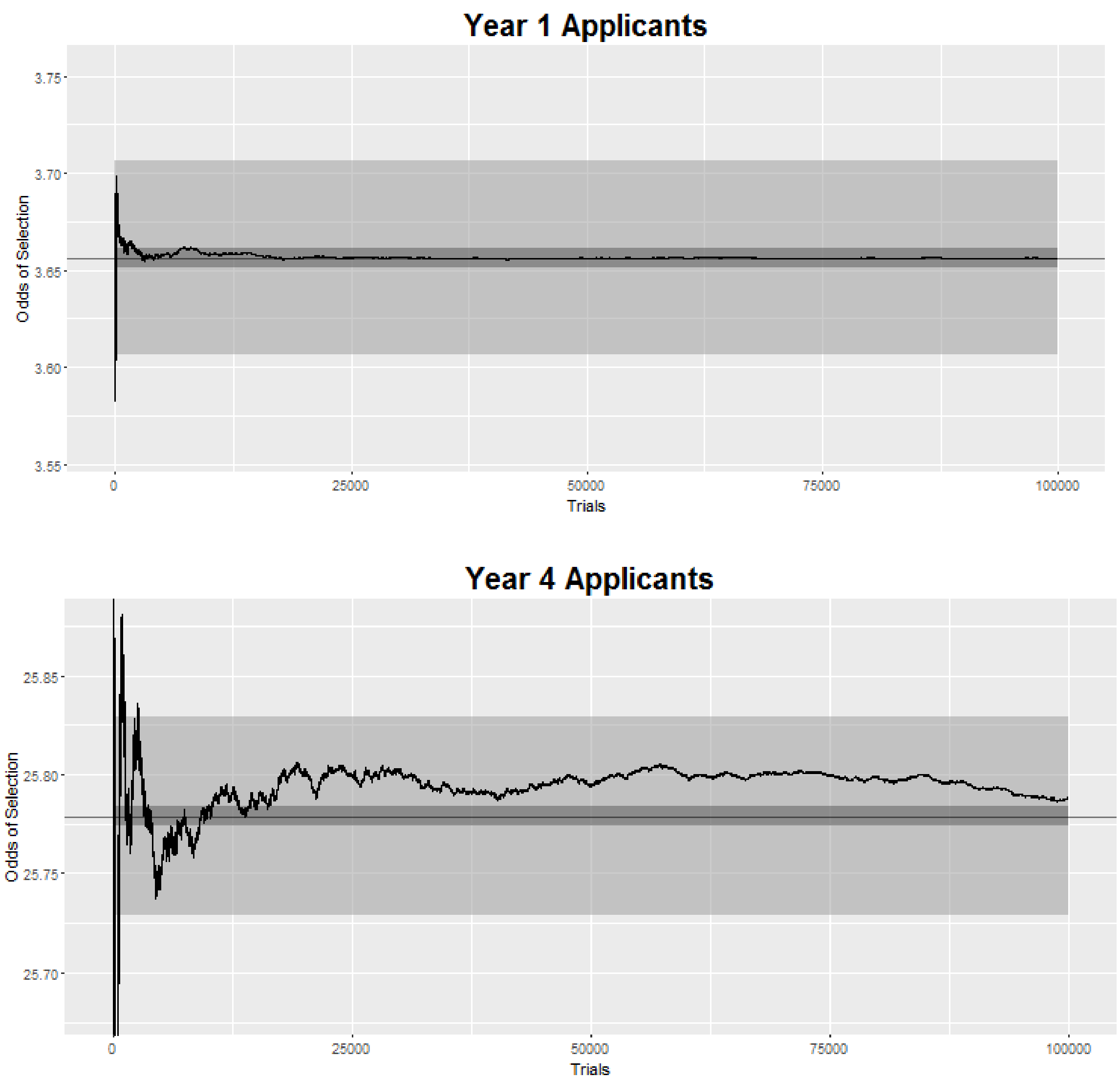

Chart 2: *Estimated number of applicants selected as trial size increases*

Actual Drawing

Using Monte Carlo simulation, we have been able to confidently compute the expected number of applicants selected from each category. On December 5th, 2015, the actual drawing took place in Auburn, California. We can therefore compare the realized results with the predicted results and quantify how "unusual" the actual results were.

To do this, we will compute the Mahalanobis distance[7] between the observed outcome and the expected outcome. We can consider each of our 100,000 trials as a point in 7-dimensional space, where the coordinates of each point are the number of applicants selected from each of the seven categories. We can picture these points as forming some sort of cluster. The Mahalanobis distance measures the distance from a given point (in our case the realized outcome) to the center of this cluster. However, unlike a standard Euclidean distance measure, the Mahalanobis distance incorporates correlations between variables.

More formally, let $\underline{x} = (x_1, x_2, \dots, x_N)^T$ be the actual lottery outcome and let $\underline{\mu} = (\mu_1, \mu_2, \dots, \mu_N)^T$ be the mean lottery outcome. Further, let S be the covariance matrix of our observations. Then the Mahalanobis distance D_M is defined by

$$D_M = \sqrt{\left(\underline{x} - \underline{\mu}\right)^T S^{-1} \left(\underline{x} - \underline{\mu}\right)}$$

In the case of the 2016 lottery, Table 3 shows the realized results.

1 Year	2 Year	3 Year	4 Year	5 Year	6 Year	7 Year
82 (2233)	49 (639)	53 (377)	38 (171)	32 (71)	12 (14)	4 (5)

Table 3: *Actual results of 2016 drawing*

Thus, setting $\underline{x} = (82, 49, 53, 38, 32, 12, 4)$ and using $\underline{\mu}$ and S computed empirically from our set of 100,000 simulations, we get $D_M = 3.32$. In other words, the realized results differ from the predicted results by 3.32 people. Relative to the total number of applicants selected (270), we actually landed quite close to the average.

In fact, with a Mahalanobis distance of 3.32, the 2016 drawing fell only in the 29th percentile with respect to distance from the mean. So not only was the absolute divergence small, it was also small relative to the total set of simulations.

Chart 3 shows a histogram of Mahalanobis distance for each of our 100,000 simulations. In red we've labeled where the realized drawing would fall. In total, the 2016 drawing was nothing special. If anything, the drawing was noteworthy for actually being "less unusual" than expected.

[7] Mahalanobis's definition was prompted by the problem of identifying the similarities of skulls based on measurements in 1927.

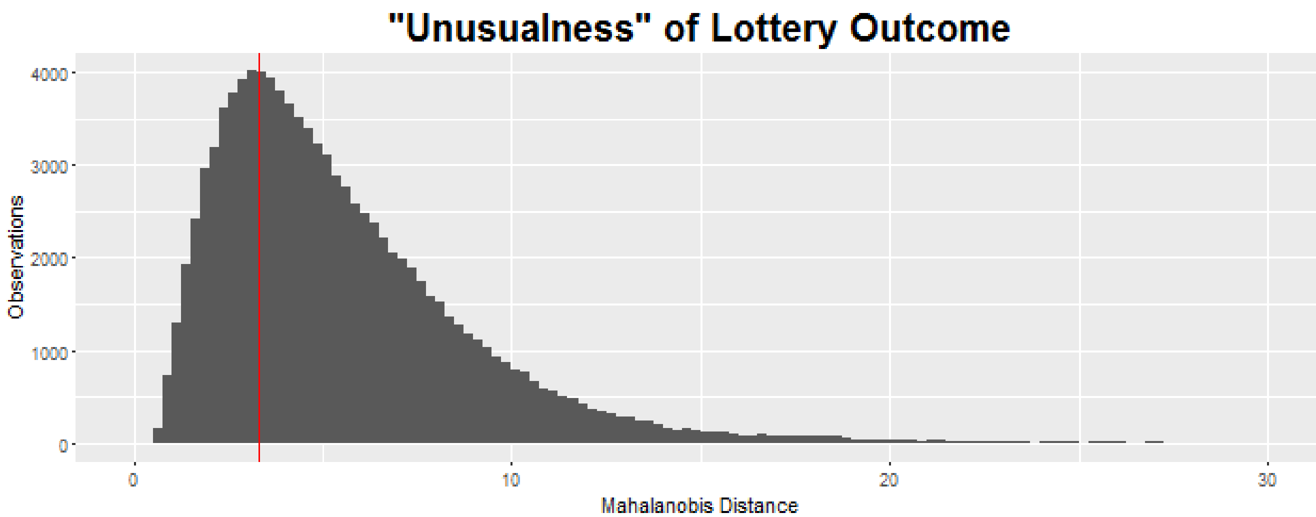

Chart 3: *The Mahalanobis distance of 100,000 simulated trials*

Conclusion

In this section, we examined the expected accuracy obtained by running 100,000 simulations. When compared to the total number of possible outcomes, 100,000 simulations is at least 6 orders of magnitude too low. However, by using the Central Limit Theorem, we saw that 100,000 was adequate to compute selection odds to one decimal place of accuracy with 95% confidence. However, in order to achieve three decimal places of accuracy (which is what the race organizers currently report), over 2.6 billion simulations would need to be run. This computational cost, coupled with the fact that these odds are only meant to serve as a rough guide, suggests that limiting the reported odds to one decimal place is both more efficient and more accurate. On the other hand, as the next section will demonstrate, it is possible to compute the odds of selection directly with complete accuracy, rendering the entire simulation approach unnecessary. Lastly, we demonstrated a way of evaluating how "unusual" a given drawing was and showed that the 2016 drawing was quite ordinary.

3. Quantification of Selection Odds

In order to compute the odds of being selected for a given number of tickets, we need not quantify the entire distribution of possible outcomes (which the Monte Carlo method approximates). Rather, we need only compute the *expected value*[8] of the number of applicants selected from each category. By working only with expected value, we can simplify the calculations and render Monte Carlo simulations unnecessary.

Since we are concerned only with the *expected* number of applicants drawn in each category, we can consider each round of the lottery to draw a fraction of runners from each category, even though in reality only a single runner from a single category will be drawn each round. Likewise, in each round an *expected* number of "dead" tickets will be removed from the hat. We will then iterate through each of the 270 rounds, updating our expected odds of selection with each round.

[8] From Wikipedia: the expected value of a random variable is intuitively the long-run average value of repetitions of the experiment it represents. For example, the expected value of a six-sided die roll is 3.5 because, roughly speaking, the average of an extremely large number of die rolls is practically always nearly equal to 3.5.

The Algorithm

First, some terminology. Let X_k be a random variable representing the category of the entrant selected in the k^{th} round. Let $A_{i,k}$ be the number of applicants remaining in the i^{th} category after the k^{th} round. Let N_i be the number of tickets for each runner in the i^{th} category. (Note that N_i does not change by round.) For example, $N_1 = 1$ and $N_7 = 64$.

Let $T_{i,k}$ be the total number of tickets remaining in the i^{th} category at the start of the k^{th} round. In other words,

$$T_{i,k} = A_{i,k} \cdot N_i.$$

(Thus, $T_{i,0}$ is the initial number of tickets in the i^{th} category.) The probability of choosing a ticket from the i^{th} category with the first draw is then

$$P(X_1 = i) = \frac{T_{i,0}}{\sum_j T_{j,0}},$$

where the denominator ranges across all *j* categories.

Let $S_{i,k}$ be the number of applicants drawn from the i^{th} category in the k^{th} round. Then the *expected* number of applicants drawn is simply the probability of selecting that category. In other words,

$$E[S_{i,k}] = P(X_k = i).$$

Again, since we are computing the *expected* number of applicants selected, these values need not be whole numbers. Table 4 shows the probabilities by category.

Category: i	Entrants Remaining: $A_{i,0}$	Tickets per Runner: N_i	Tickets Remaining: $T_{i,0}$	Category Probability: $P(X_1 = i)$	Number Selected: $E[S_{i,1}]$
1	**2,233**	1	2,233	26.91%	0.269
2	**639**	2	1,278	15.40%	0.154
3	**377**	4	1,508	18.17%	0.182
4	**171**	8	1,368	16.58%	0.166
5	**71**	16	1,136	13.69%	0.137
6	**14**	32	448	5.40%	0.054
7	**5**	64	320	3.86%	0.039
	3,511		**8,299**	**100.00%**	**1.000**

Table 4: *First ticket probability by category*

From Table 4, we can see that there is a 26.9% chance that the first ticket is drawn a first-year applicant. Likewise, there is a 3.9% chance that a seventh-year applicant is the first to be drawn. Note that since we are concerned only with the *expected* number of applicants selected, the fact that $S_{i,1}$ is not a whole number is reasonable. Lastly, we can see that $\sum_j S_{j,1}$ is precisely 1, meaning that we've fully allocated our first selection across the seven categories.

So far, the odds have been computed in a straightforward manner. What makes the calculation of odds in subsequent rounds non-trivial is the removal of "dead" tickets generated with each selection. In particular, the total number of remaining tickets ($T_{i,k}$) will change depending on which category of applicant is selected in each round.

Updating the Odds

To compute the updated odds for the second draw, we must update the number of remaining tickets for each category. We know the expected number of runners drawn in each category in the first round ($E[S_{i,1}]$), and the number of live tickets for runners in each category (N_i). Thus, we can update the number of entrants remaining in the i^{th} category after the k^{th} round as follows:

$$A_{i,1} = A_{i,0} - E[S_{i,1}].$$

Lastly, we can update the tickets remaining in the i^{th} category after the k^{th} round as follows:

$$T_{i,1} = A_{i,1} \cdot N_i.$$

We can then compute the expected reduction in ticket count by category after the first draw.

Category: i	Entrants Remaining: $A_{i,1}$	Tickets per Runner: N_i	Tickets Remaining: $T_{i,1}$	Category Probability: $P(X_2 = i)$	Number Selected: $E[S_{i,2}]$
1	**2,232.7**	1	2,232.7	26.93%	0.269
2	**638.8**	2	1,277.7	15.41%	0.154
3	**376.8**	4	1,507.3	18.18%	0.182
4	**171.8**	8	1,374.7	16.58%	0.166
5	**70.9**	16	1,133.8	13.68%	0.137
6	**13.9**	32	446.3	5.38%	0.054
7	**5.0**	64	317.5	3.83%	0.038
	3510.0		**8,290.0**	**100.00%**	**1.000**

Table 5: *Updated odds for second drawing*

By comparing Table 4 and Table 5, we can see several interesting things. First, the probability of selecting a first-year applicant is higher in the second round than the first round. This is reasonable, since each first-year applicant has just a single ticket in the lottery and the total number of tickets left in the hat has decreased after the first draw. However, the odds of selecting a seventh-year applicant is actually *lower* in the second round than the first round. This is due to the fact that almost 28% of the tickets expected to be removed from the hat came from seventh-year applicants. While the total hat was reduced by about 0.1%, the seventh-year had was reduced by about 0.7%. Therefore, the seventh-year odds actually declined.

Now that we've computed the expected number of applicants drawn from each category in the second round, we can add this to our expected number from the first round and arrive at an aggregate count after two rounds. In this way, we can iterate for a full 270 rounds and compute the precise expectation by category.

Let S_i be the total number of applicants drawn from the i^{th} category after all 270 of the lottery. In other words,

$$S_i = \sum_{k=1}^{270} S_{i,k}.$$

Then

$$E[S_i] = \sum_{k=1}^{270} E\left[S_{i,k}\right].$$

Once we have our expected number of selections by category, we can easily compute the likelihood of any particular runner being selected from that category. Let P_i be the probability of a runner from the i^{th} category being selected in the lottery at any point. Then

$$P_i = \frac{E[S_i]}{A_{i,0}}.$$

In other words, once we know the expected number of applicants drawn from the i^{th} category, we can divide by the total number of applicants in that category to arrive at our final probability of being selected.

Precise Results

Given the above algorithm, the precise probability of selection can be computed, without any need for Monte Carlo simulation. Appendix C contains R code to quickly compute the results in the case of the 2015 Western States lottery. Table 6 summarizes the results.

Category: i	Total Entrants: $A_{i,0}$	Tickets per Runner: N_i	Probability of Selection: P_i	Number Selected: $E[S_i]$
1	**2,233**	1	3.656%	81.6
2	**639**	2	7.179%	45.9
3	**377**	4	13.845%	52.2
4	**171**	8	25.779%	44.1
5	**71**	16	44.931%	31.9
6	**14**	32	69.714%	9.8
7	**5**	64	90.876%	4.5
	3,511			**270.0**

Table 6: *Precise odds of selection*

By comparing the precise results in Table 6 with the simulated results in Table 1, we can see that the expected number of runners taken in each category matches exactly to within one decimal place. However, the odds of selection in the simulated results (shown to three decimal places) differ slightly from the true odds computed above. The odds for the fifth-year applicants are slightly overstated and the odds for the sixth- and seventh-year applicants are slightly understated. However, none of these discrepancies are significant.

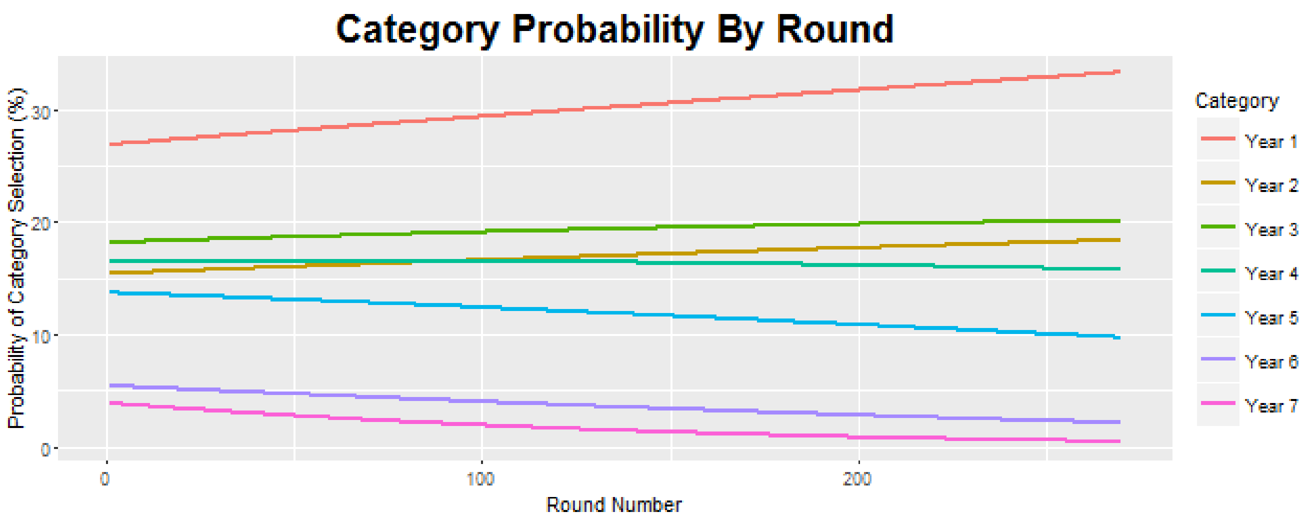

Chart 4: *Category selection odds by round*

Chart 4 shows the probability of selecting an applicant from each category as the lottery progresses. We can see that the odds of selecting a first year applicant steadily rise from about 26.9% in the first round to over 33% in the final round. Likewise, the chances of selecting second- and third-year applicants also rises throughout the lottery. On the other hand, the chances of selecting a seventh-year applicant decline from about 3.9% in the first round to just 0.5% in the final round. The odds for fourth-, fifth- and sixth-year applicants also fall as the lottery progresses.

Because of these shifting odds, the early rounds will see a higher proportion of senior applicants selected than the later rounds. For instance, in Table 6 we saw that 4.5 seventh-year applicants are expected to be selected in total. However, we expect to see 50% of these selections happen by round 75 (28% of the way through the lottery) and 75% of these selections by round 137 (51% of the way through). In summary, if you have 16 or more tickets in the lottery and plan to attend the drawing in person, don't arrive late: if you're going to be selected it will probably happen early.

Conclusion

This section has described an algorithm for computing the precise odds of selection without the use of random simulation. With this procedure in hand, there is no compelling reason for the race organizers to continue computing odds via Monte Carlo methods. Using this method will improve the Western States lottery experience in several ways. First, the accuracy of the lottery odds will increase, particularly for those runners with the most tickets in the hat. Second, the calculation of odds is now a deterministic process which is no longer subject to elements of random chance. While these random fluctuations are "smoothed out" by running a sufficient number of simulations, we saw in Section 2 that 100,000 simulations was an insufficient number to achieve our desired level of accuracy. Third, the running time of this procedure is significantly less than the simulated approach. For example, on an average PC, a simulation of 100,000 trials took about five and a half minutes to run. The deterministic procedure took one tenth of a second. That's five minutes the race organizers could have spent on more important aspects of the race.

4. Quantification of Draw Counts

Referring again to Table 1, we see that the race organizers report the following statistics from the simulation:

Average number of draws required: 303.1
Minimum number of draws required: 279
Maximum number of draws required: 336

One may ask why these statistics are relevant to the entrants in the lottery. While they do not impact the odds of any runner being selected, it's possible that the race organizers wish to gauge the amount of time the drawing will take, since it is done before a live audience and streamed online in real time. Thus, understanding how often a ticket belonging to a previously-selected applicant will be drawn may help for planning purposes.

It is important to note that these statistics merely describe the set of 100,000 observations generated during the Monte Carlo simulation run by the race organizers. They do not necessarily describe the "true" underlying distribution of potential lottery outcomes. (This is particularly acute in the case of statistics such as minimum and maximum which are sensitive to just a single extreme observation.) This section will examine what the "true" minimum, maximum and average number of draws are and how likely we are to observe such an outcome.

Minimum Number of Draws

Based on the 100,000 simulations run by the race organizers, the reported "minimum number of draws" was 279. However, we know *a priori* that this is not the true minimum required. In fact, we can easily construct an outcome which requires just 270 draws: every ticket drawn comes from the first-year applicant category. While this is unlikely, it is by no means impossible. The fact that the reported "minimum number of draws" was not precisely 270 simply indicates that not enough simulations were run. As the number of simulations goes to infinity, the likelihood of observing a lottery with only 270 required draws becomes a certainty. In other words, let $X = (X_1, X_2, \dots, X_n)$ be a random sample of size n from the distribution describing the total number of tickets drawn in each simulation. Then,

$$\lim_{n \to \infty} P(\min\{X\} = 270) = 1.$$

How likely are we to observe a drawing which requires only 270 draws? We can compute this if we keep track of the number of "dead" tickets in the hat after each round. We can then compute the probability of avoiding a "dead" ticket in all 270 rounds.

Chart 5 shows the declining chances of escaping without a duplicate ticket through each round. We see that by round 39, the likelihood of avoiding duplicates has already dropped below 50%. By round 70, there is only a 10% chance that we've avoided duplicates. By the 100^{th} round, there is only a 1% chance we have not yet drawn a duplicate.

In total, the chances of drawing 270 unique names in 270 attempts is about one in 32,504,169,501,623 trials. Thus, if we ran about 32.5 trillion simulations, we should expect to see an instance where only 270 drawings were required.

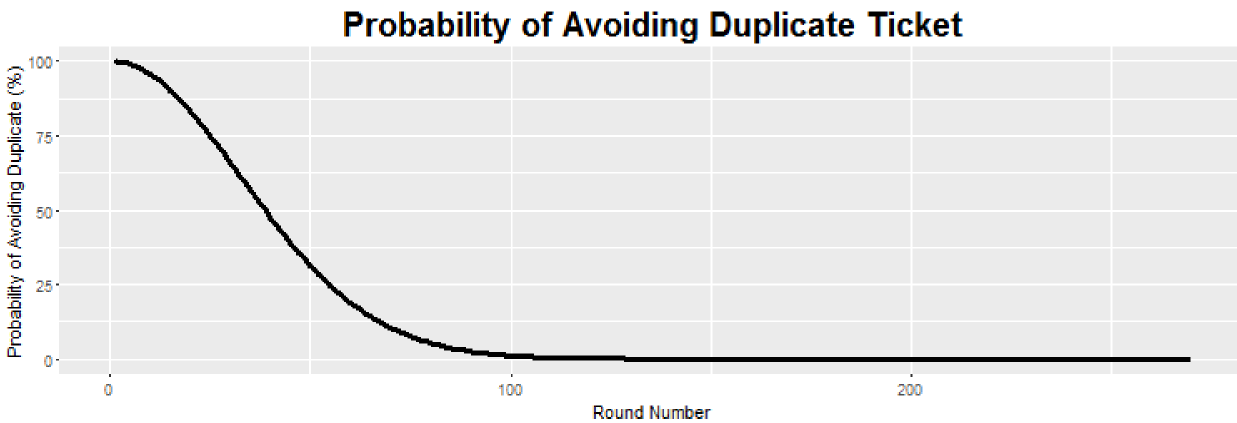

Chart 5: *Odds of escaping a duplicate draw*

Maximum Number of Draws

Based on the 100,000 simulations run by the race organizers, the reported "maximum number of draws" was 336. However, much like the minimum number of draws, we know *a priori* that 336 is not the true maximum of the distribution. It simply happened that 336 was the maximum number *observed* during a particular simulation of 100,000 drawings.

We can easily construct a drawing which requires the maximum number of draws. Simply order the applicants in decreasing order by ticket count. (Ties can be broken in any manner desired.) Then draw the tickets in that order. For example, the first ticket drawn will be from a seventh-year applicant (who has 64 tickets in the hat). The next 63 tickets will also be from this same applicant (and will be removed after each draw). The second name will be drawn with the 65^{th} ticket and will be from another seventh-year runner. Likewise, the following 63 tickets will be duplicates from this runner. We continue in this manner until all seventh-year applicants are selected. We then repeat the process with sixth-year applicants, and so on.

Category	Entrants	Tickets per Runner	Number Selected	Number of Draws
7	**5**	64	5	320
6	**14**	32	14	448
5	**71**	16	71	1,136
4	**171**	8	171	1,368
3	**377**	4	9	33
2	**639**	2	0	0
1	**2,233**	1	0	0
	3,510		**270**	**3,305**

Table 7: *Maximum number of draws required*

How many draws will be required before we fill the 270 slots for the race? Table 7 enumerates the 3,305 draws required given the ordering described above.

Why did the race organizers report a maximum of only 336 when the true maximum is nearly ten times higher? Observing a drawing which requires 3,305 draws is extremely rare and simply would not be expected to occur using only 100,00 simulations. However, as the number of simulations goes to infinity, the likelihood of observing a drawing requiring 3,305 draws becomes a certainty.

The likelihood of observing a drawing requiring 3,305 tickets can be computed as follows. We need not pick the runners in the exact order described above (though that will certainly work). Rather, we need only pick all 320 seventh-year tickets, all 448 sixth-year tickets, all 1,136 fifth-year tickets and all 1,368 fourth-year tickets among the first 3,305 draws. Further, we know we must draw all 4 tickets for 8 of the 377 third-year runners. Our 3,305th and final draw can be from any remaining runner.

The number of possible ordered draws of 3,304 tickets from a pool of 8,299 is:

$$\frac{8299!}{4995!} \approx 10^{12,615}$$

The number of ordered draws which use all of the seventh-, sixth-, fifth- and fourth-year applicants plus 8 of the 377 third-year applicants is:

$$\frac{3304!}{\binom{377}{8}} \approx 10^{10,178}$$

Thus, the probability of drawing the maximum number of tickets to get 270 unique names is[9]:

$$\frac{3304!}{\binom{377}{8}} \bigg/ \frac{8299!}{4995!} \approx 10^{-2,437}$$

We could run simulations until the end of time and likely never observe an instance of a 3,305-ticket draw. However, in theory given enough simulations, this *will* occur with certainty.

Average Number of Draws

Computing the expected number of duplicate draws is more complicated. However, we can modify our algorithm for determining the expected number of selections by category in order to compute the expected number of duplicate draws.

First, we need a method of computing the expected number of duplicates drawn at a given point in the lottery. Let T_k be the total number of tickets left in the hat at the start of the k^{th} round. Some portion of these tickets will be "dead", in the sense that another ticket for that particular runner was previously drawn. Let D_k be the number of "dead" tickets in the hat after k draws. Let N_k be the number of duplicate tickets drawn in the k^{th} round.

For example, at the start of the lottery, T_1 is the total number of tickets in the hat and $D_1 = 0$. Say a seventh-year applicant is selected with the first draw. Then $T_2 = T_1 - 1$ and $D_2 = 63$.

Let X_k be a random variable measuring the number of consecutive duplicates drawn to start the k^{th} round. Then the probability of selecting exactly n duplicates followed by a non-duplicate is

[9] This is somewhat overstating the case. This calculation assumes that 3,304 tickets will be drawn and considers the likelihood that they are all of a particular type. However, in practice the lottery stops once 270 names have been selected. Thus, this calculation is counting many instances which would never actually occur.

$$P(X_k = n) = \left(\prod_{i=1}^{n} \frac{D_k - i}{T_k - D_k}\right) \cdot \frac{T_k - D_k}{T_k - n}$$

The expected number of consecutive duplicate tickets drawn in the k^{th} round is

$$E[N_k] = \prod_{n=1}^{D_k} n \cdot P(X_k = n)$$

We can then update T_k and D_k as follows:

$$T_{k+1} = T_k - 1 - E[N_k]$$

$$D_{k+1} = D_k - E[N_k] + V_k,$$

where V_k is the expected number of new duplicates generated in drawing k (as described in Section 3).

Thus, as the lottery proceeds according to the algorithm described in Section 3, we must keep track of the total number of tickets left in the hat (T_k), the total number of dead tickets in the hat (D_k) and the cumulative sum of the expected number of duplicates drawn ($E[N_k]$).

Using the R code provided in Appendix D, we compute the total number of duplicates as 33.05192. Thus, in total, we expect the Western States Lottery to require 303.05192 draws to choose the 270 entrants. To one decimal place, this matches the simulated result exactly.

Chart 6 shows the cumulative expected number of duplicates by round. We can see that the likelihood of drawing a duplicate increases and in fact accelerates as the lottery progresses. Near the end of the 270-round lottery, we expect a duplicate ticket to be drawn about once every four rounds.

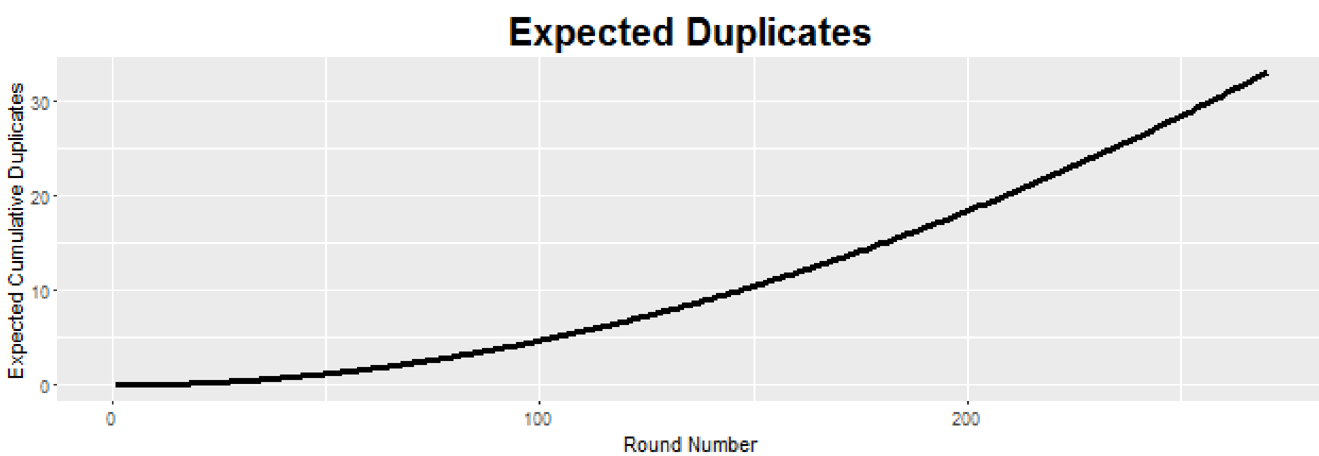

Chart 6: *Expected number of duplicates by round*

Conclusion

This section demonstrated several things. First, we showed what the true minimum and maximum required number of draws was and how likely we were to observe such outcomes. We also computed the expected number of draws to fill all 270 slots in the race. Second, we noted that, in general, describing a distribution using its maximum and minimum is not very useful. The minimum and maximum are each determined by just one extreme outcome, no matter how unlikely that outcome may be. If the race organizers wish to report the likely range of drawings required, a better option would be to use percentiles. For example, by removing the 5% of extreme cases on each end of the distribution, we could report that 90% of simulations required between 295 and 312 drawings. Third, we showed how the number of expected duplicates grows by round. This information could be useful when planning the pacing of the live lottery drawing. For instance, we can see that drawing the last 50 names will take approximately 20% longer than drawing the first 50 names.

4. Effect of Ticket Scaling on Field Composition

It is clear that the number of Western States applicants far exceeds the number of available spots and some method must be used to determine which runners will win the coveted entries. A reasonable question to ask is: why use a lottery at all? Many organizations facing a similar problem simply use a first-in-first-out queue. (For example, many NFL teams use this method for allocating season ticket packages. The waiting list for Pittsburg Steelers season tickets is currently 88,000 people long and it will take an estimated 50 years to reach the front.) Arguably, this system is the “fairest” because no one can “cut the line” ahead of someone who has been waiting longer. There is no random component deciding who wins and who doesn’t. Everyone knows where they stand at all times. One downside of such a system is the fact that someone joining the back of the line today may have little hope of reaching the front in a reasonable amount of time (as in the case of a new Steelers fan).

On the other hand, a lottery system allows a portion of the applicants to “cut the line” and win a entry ahead of those who have waited in line longer. While this is not “fair” to those being cut, it does have the advantage that it provides every entrant, no matter how low on the list, some hope that they may be selected. This keeps more potential applicants engaged and interested in a race like Western States. This in turn keeps sponsors interested and keeps demand high for future years.

Therefore, it is crucial that the Western States race organizers strike a balance between a “fair” system that is determined solely by one’s place in line and a “random” system which provides hope to every runner that they may be selected in the upcoming drawing.

Different Scaling Factors

The Western States race organizers have chosen to use the “*2^(n-1)*” system. But why not “*3^(n-1)*” or “*10^(n-1)*”? For that matter, why not “*2.5^(n-1)*”. This section will examine the impact of using different scaling factors on the composition of the field selected.

Chart 7 shows the number of runners selected from each category for a given ticket scaling factor. The counts at a scale of 2 are those we calculated in Section 3 above. You can see that as the scale increases, the composition shifts more and more towards the more seasoned applicants at the expense of the newer applicants. In other words, there is less “cutting” the line.

Essentially, the scaling factor controls the balance between a completely random drawing and a purely first-in-first-out queue. For example, a scaling factor of 1.0 awards each applicant with a single ticket and thereby gives no preference to more senior applicants. On the other hand, as the scaling factor grows to infinity, so much weight is given to seniority that the drawing becomes a simple first-in-first-out queue like that for Steelers season tickets.

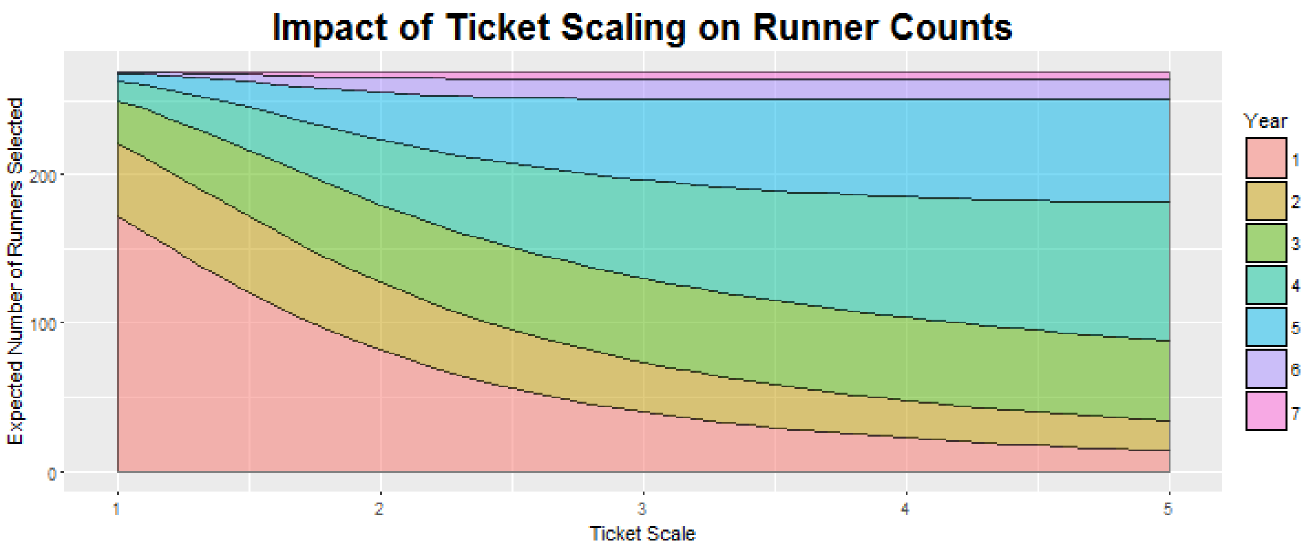

Chart 7: *Expected number of applicants selected using different ticket scales*

Disappointment and Surprise Levels

We can think of two competing forces at work: surprise and disappointment. First, we will define a "surprise" metric. To some extent, every runner selected will feel some level of surprise at their good fortune. However, not all applicants will be equally surprised. Let's say that an applicant's surprise level is inversely proportional to the number of tickets they had in the had. Thus, using a scale factor of 2, a first year applicant will be twice as surprised as a second-year applicant at being selected. Likewise, they will be 64 times more surprised than a seventh-year applicant. Applicants who are not selected will feel no surprise.

Next, we'll define an "disappointment" metric. This is the disappointment felt at not being selected. Let's say that an applicant's disappointment level is proportional to the number of tickets they had in the hat. Again, using a scale factor of 2, a seventh-year applicant who is not selected will be twice as disappointed as a sixth-year applicant and 64 times as disappointed as a first-year applicant. Applicants who are selected will feel no disappointment.

For a given ticket scaling level, we can then measure the total amount of disappointment and surprise generated across the entire pool of applicants. However, in order to compare across different scale factors, we shall normalize the surprise and disappointment levels by the total number of tickets in the hat for a given drawing.

Let S_i be the number of applicants selected from the i^{th} category, let A_i be the total number of applicants from the i^{th} category and let z be the scale factor. Then,

$$Surprise(z) = \frac{\sum_{i=1}^{k}\left(S_i \cdot z^{k-i+1}\right)}{\sum_{i=1}^{k}(A_i \cdot z^i)}$$

$$Disappointment(z) = \frac{\sum_{i=1}^{k}\left((A_i - S_i) \cdot z^i\right)}{\sum_{i=1}^{k}(A_i \cdot z^i)}$$

Chart 8 shows the surprise and disappointment levels for varying ticket scales between 1 and 5. As you can see, at a scale factor of 1.0 there is very little surprise since with equal ticket counts no one can be surprised that they "beat the odds" and were chosen ahead of someone with more tickets. On the other hand, there is a high degree of disappointment since the losers lost to people who had the same number of tickets and yet won.

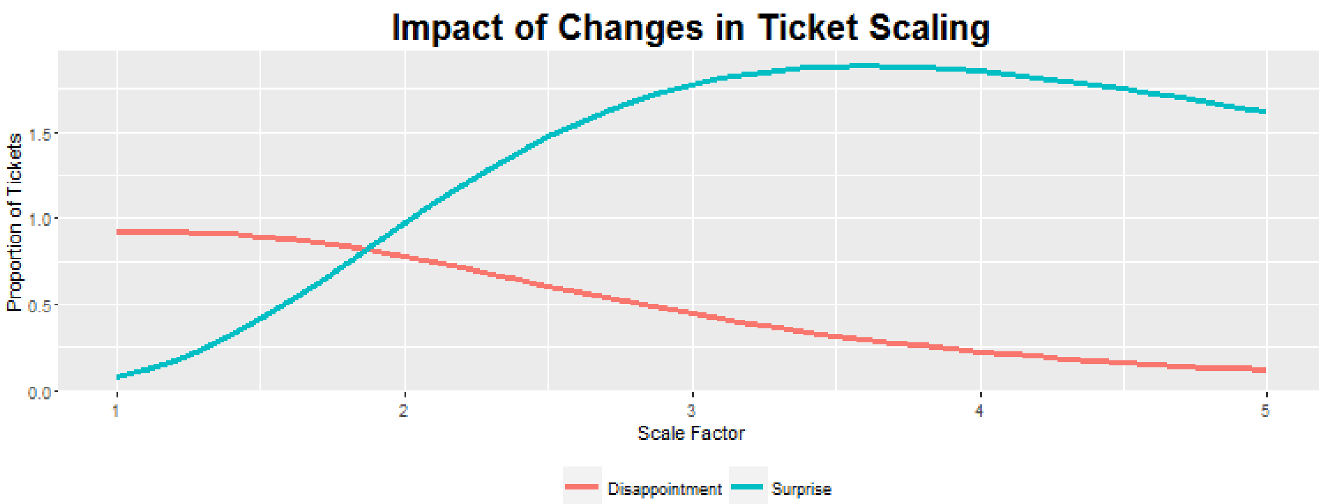

Chart 8: *Relative surprise and disappointment levels caused by different ticket scaling factors*

As we increase the scale factor above one, we begin giving preference to more senior applicants. The disappointment level decreases (since fewer senior applicants are being cut in line) and the surprise level increases (as the junior applicants who are selected feel that they "beat the odds" to be selected). At a scale factor of about 1.8, the total disappointment level equals the total surprise level. At scale factors above this point, the surprise level exceeds the disappointment level.

The total surprise level peaks around 3.5. Above this scale factor, the lottery resembles more and more closely a first-in-first-out queue and there is less and less surprise available. Chart 9 shows the net happiness level (surprise minus disappointment). Total happiness is maximized at a scale factor of about 4.0. We again see that we've balanced surprise and disappointment at about 1.8.

Fractional Scales

We saw that a scale factor of 1.8 seems to be optimal. However, since the lottery is conducted in-person using physical slips of paper in an urn, fractional scale factors are not practical. (In particular, the slips would need to be weighted in some manner, which is likely infeasible.) However, should the race organizers migrate away from the physical drawing to an electronic drawing, the use of a fractional scale factor would pose no theoretical issues.

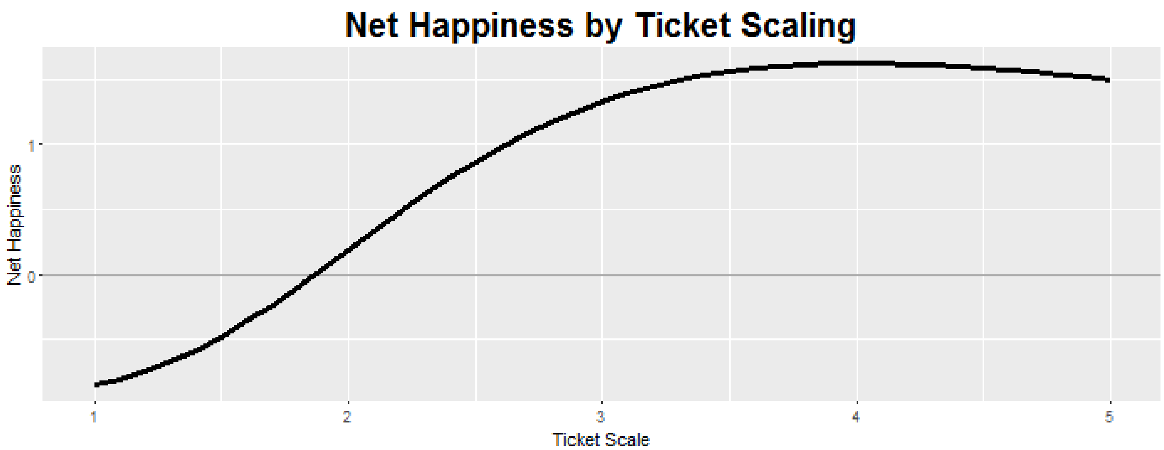

Chart 9: *Aggregate happiness level by ticket scaling factor*

Conclusion

In this section, we demonstrated the impact of changing the number of tickets awarded to lottery losers in subsequent years. We saw that as we increase the number of tickets, the lottery becomes less random and more like a simple waitlist. We saw just one way to define the "surprise" and "disappointment" caused by a particular choice of scale factor. Using this metric, it turns out that the current scale factor of two does a good job balancing total surprise and disappointment. However, maximum total happiness (defined as surprise minus anger) would actually be achieved at a scale factor near four. Above this level, the total happiness actually declines. We also noted that as long as the lottery is conducted manually, a fractional scale factor is not practical. For these reasons it appears that the current scale factor of two is actually an excellent choice.

5. Expected Time to Selection

It is clear that one's odds of being selected in the Western States lottery as a first-year applicant are not good. However, under the "*2^(n-1)*" system, it would seem that those odds are bound to improve with each year. Therefore, a natural question to ask is: how long would a new applicant expect to wait before being selected?

A Naïve Approach

One way to answer this question is to look at those selected and compute the average number of years those applicants have spent in the lottery. In other words, we can compute the average waiting time $\bar{W}$ as follows:

$$\bar{W} = \frac{\sum(i \cdot E[S_i])}{\sum E[S_i]}$$

Given the values of $E[S_i]$ found in Table 6, we can compute the average wait time of those runners selected in the 2016 Western States lottery as almost exactly 2.8. Naively, one may therefore conclude that it will only take only 2.8 years in the lottery to be selected.

However, this is misleading. To see why, consider the following analogy to the Powerball lottery[10]. We may ask, How many tickets does one need to buy, on average, to win the jackpot? Let's say in a recent drawing there were two winners: one bought 10 tickets and the other bought 30 tickets. Thus, on average, the lottery winners bought 20 tickets each. The erronious conclusion is that one should expect to win the jackpot after buying just 20 tickets. After all, the winners really did win with just 20 tickets, on average. But of course this ignores the millions of people who did not win. The winners didn't win because they bought 20 tickets; they won because they were lucky. The same principal applies to the Western States lottery: if we consider only the waiting time of the winners, we ignore the waiting time of the vast majority of applicants who were not winners (and by definition have longer waiting times).

Estimating Growth of Applicants

In order to properly compute the expected waiting time for a given runner, we must compute their odds of selection each year into the future. Given those odds, we can then easily compute the expected number of years until selection. For a given runner, we know that each year their number of tickets will double. However, what we don't know is how the rest of the field will evolve. In other words, we don't know how many new first-year applicants will appear. We also don't know how many previous applicants will continue to apply. To estimate these values in future years, we will build a model of future growth based on recent trends.

Table 8 shows the count of actual Western States lottery applications by year since 2000.

Year	Total Applicants	1 Year	2 Year	3 Year	4 Year	5 Year	6 Year	7 Year	Notes
2000	583								
2001	556								
2002	529								
2003	638								
2004	740								
2005	791								
2006	841								
2007	1048								
2008	1350								Fire year, race cancelled
2009	no lottery								Rollovers, two-time losers, autos
2010	1693								Last year for two-time losers
2011	1786	1286	500						electronic drawing
2012	1940	1221	461	258					electronic drawing
2013	2295	1486	480	207	122				manual drawing
2014	2704	1727	561	258	106	52			manual drawing
2015	2566	1427	641	281	136	57	24		First year of 2^(n-1)
2016	3510	2233	639	377	171	71	14	5	

Table 8: *Western States lottery application counts by year*

There are a few things to note about this data. First, there was no lottery held for the 2009 race (the 2008 race was cancelled due to wildfire and all entries were carried over). Second, the 2015 lottery was the first to use the "*2^(n-1)*" system. Last, the qualifying standards were tighened with the 2015 lottery, causing a slight dip in total applicants. However, you can see that the application count has bounced back in 2016 despite the tougher standards.

[10] Powerball is a multi-state lottery where players try to match five numbers between 1 and 69 and one number between 1 and 26. The odds of getting an exact match are about 1 in 292 million.

Chart 10 shows the growth in total applicants through time. The grey bars show the actual count of total applicants. (To smooth impact of the change in qualification standards, the 2015 data has been replaced with a linear interpolation of the 2014 and 2016 data.) In order to project future growth in application count, we have fit an exponential curve to the observed data. This curve has the form

$$Applicants(t) = 381 \cdot e^{0.13t},$$

where t is the number of years since 1999. This curve is shown in black in Chart 10.

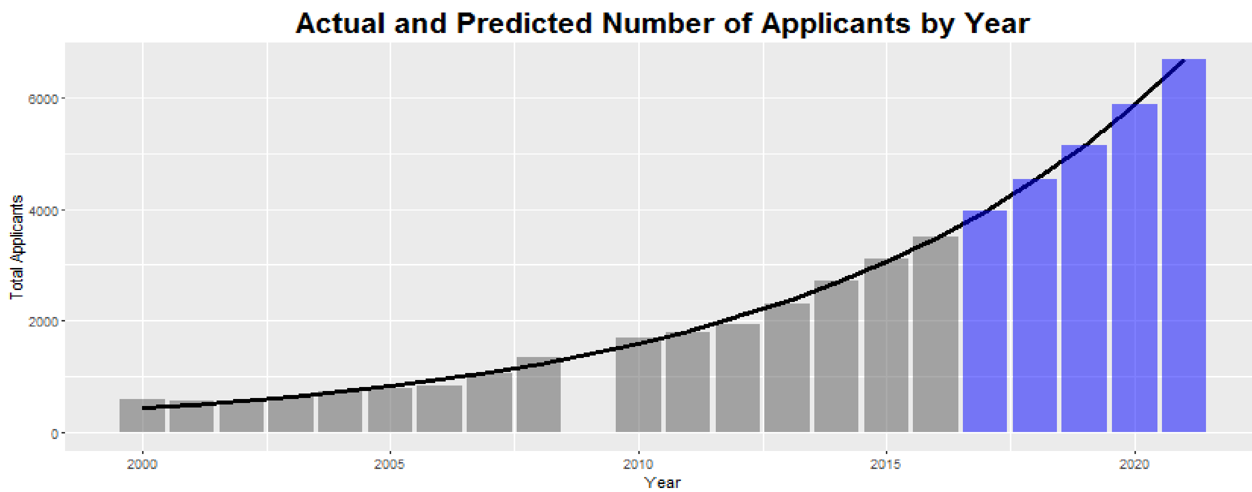

Chart 10: *Projected growth in lottery applications*

Once we have estimated the growth in the total number of applicants, we must estimate how those applicants break down by category. By looking at the actual entrants by category in the 2015 and 2016 lotteries, we can see that the applicant count seems to decay exponentially by category. Chart 11 shows this data.

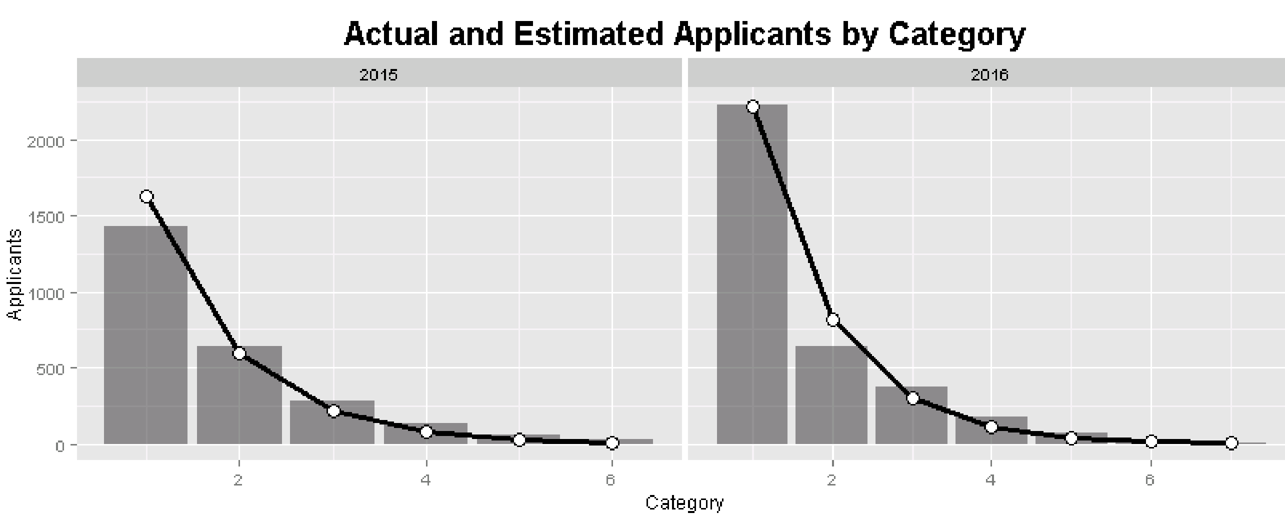

Chart 11: *Actual lottery applicant count by category with exponential decay estimate*

The black lines in Chart 11 show our estimated breakdown by category. This was computed as follows. Let's say at some future year there are projected to be N total applicants across C categories. Then our estimate for the number of applicants A_i in category i is given by:

$$A_i = \frac{N}{\sum_{j=1}^{C} e^{-j}} \cdot \frac{1}{e^i}$$

Since the denominator of first term is simply a geometric series, we could rewrite it as:

$$\sum_{j=1}^{C} \frac{1}{e^j} = \frac{1-(1/e)^{C+1}}{1-1/e} - 1 = \frac{1/e - (1/e)^{C+1}}{1-1/e}.$$

As C grows large, we see that

$$\lim_{C\to\infty} \frac{1/e - (1/e)^{C+1}}{1-1/e} = \frac{1/e}{1-1/e} = \frac{1}{e-1}.$$

Thus, we can approximate A_i by:

$$A_i = \frac{N(e-1)}{e^i} \approx 1.71828 \cdot \frac{N}{e^i}.$$

Now that we can estimate the size and breakdown of the applicant pool each year into the future, we can run the algorithm described in Section 3 and compute projected future odds by applicant category.

Chart 12 shows the projected odds for first-, second- and third-year applicants over the next ten years. As you can see, all three decline substantially. The third-year odds (which are were 13.8% in 2016) decline to about 4% by 2026. Likewise, the second-year odds decline from 7.2% in 2016 to less than 2.5% in ten years. The first-year odds, which were never good, decline to about 1%.

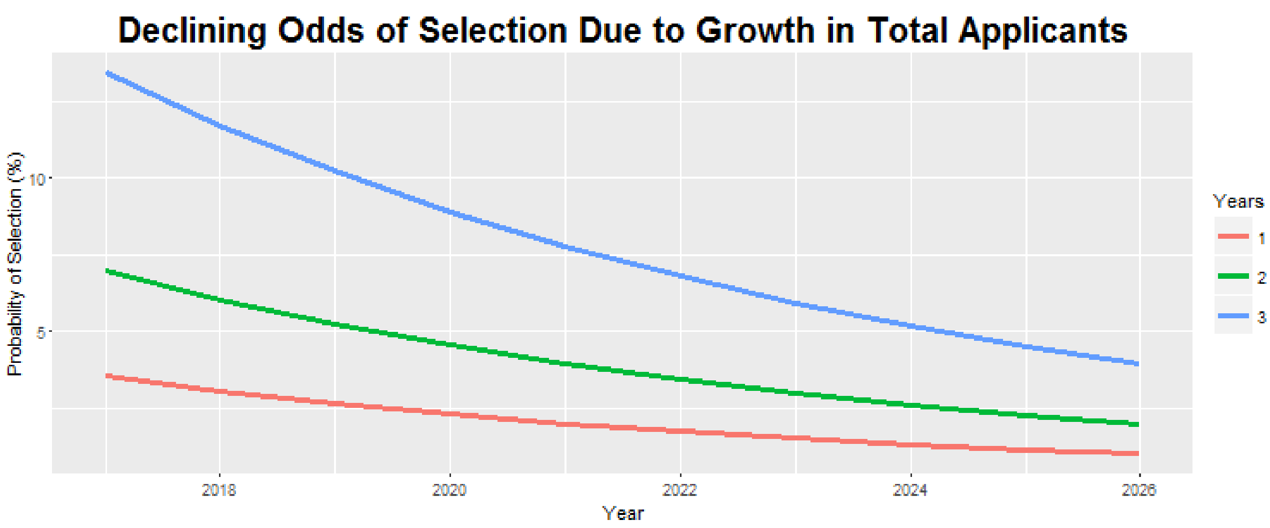

Chart 12: *Projected selection odds for first-, second- and third-year applicants*

The reason for this behavior is clear: there is a small, fixed number of available slots each year, but an exponentially-growing number of applicants. Even if each applicant doubles their tickets each year, the exponential growth of new applicants simply overwhelms the odds for everyone.

New Applicant Wait Time

Now that we have estimated the future odds for each category of applicant into the future, we are prepared to ask how long a new applicant will expect to wait before being selected. To do this, we must compute the odds of being selected in each year (and no sooner).

Let $P_{i,k}$ be the probability of a i^{th}-year applicant being selected in the k^{th} year in the future. Then the expected waiting time is given by

$$Wait = \sum_{k=1}^{\infty} k \cdot \left(\prod_{i=1}^{k-1} (1 - P_{i,i}) \right) \cdot P_{k,k}$$

In other words, for each year k from 1 to infinity, we compute the probability of *not* being selected for $k-1$ years and then being selected in the k^{th} year and weight this by k.

When we perform this calculation using our estimates for the growth of the applicant pool, we see that the expected time to selection for a new applicant in 2017 is 5.25 years. In other words, assuming the popularity of Western States continues to grow at an exponential rate into the foreseeable future, the average runner applying for the first time in 2017 will not be chosen until 2022.

Chart 13 shows the odds of selection for a first-year applicant beginning in 2017. The bad news is that the most likely year of selection is not until 2022. The good news is that it's almost certain that they will be selected by 2025.

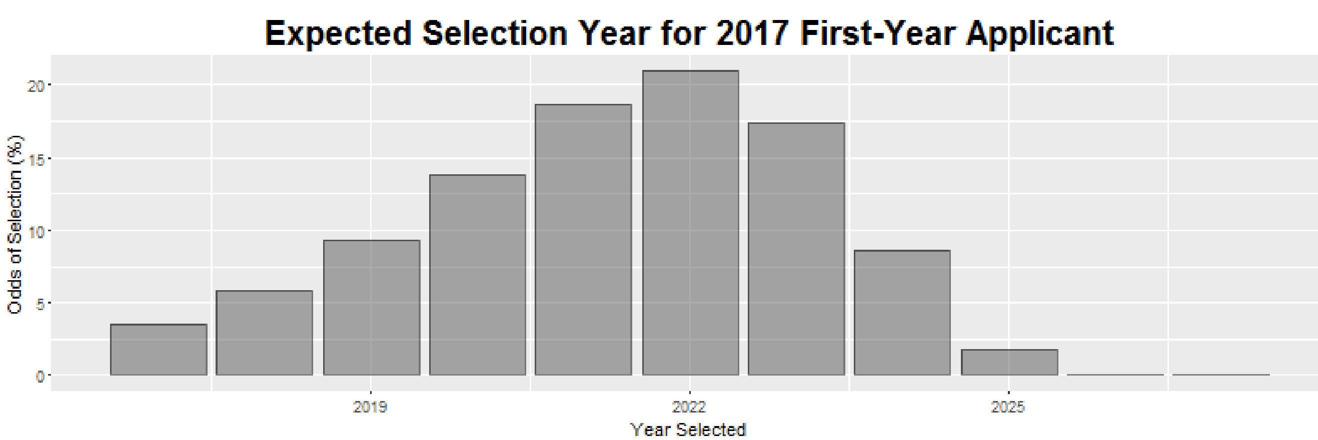

Chart 13: *Projected selection year for first-year applicant beginning in 2017*

Self-Regulation of Wait Time

The analysis above demonstrates that the expected wait time for new applicants will continue to grow into the forseeable future. But can that wait time grow indefinitely or will it level off at some point? On one hand, with an exponentially-growing applicant pool coupled with a fixed number of entries means the wait time *must* grow indefinitely. However, a more likely scenario is that a leveling-off will eventually occur due to applicant attrition. Remaining in the lottery requires finishing at least one qualifying ultramarathon every year. While every runner is different, there is certainly some limit to the number of consecutive years the average runner can run 100-mile ultramarathons. For the sake of argument, let's say this limit is ten years. Thus, due to factors such as burnout, injury, or life changes, the average runner will simply choose (or be forced) to stop applying after about ten years. The model described above captures this attrition as an exponential decay.

This dynamic results in two important facts for the Western States race organizers. One one hand, it keeps the expected wait time bounded by the length of the average applicant's ultramarathon career. Unlike Pittsburgh Steelers season tickets, we'll never see a Western States waiting time of 50 years. On the other hand, it means that most runners applying for the lottery will never actually run Western States in their lifetime. The majority of applicants will be forced to drop out before ever being selected.

Conclusion

In this section we examined the impact of a growing applicant pool on the future odds of selection. It's a simple fact that with the applicant pool growing exponentially and the Western States field fixed at about 380 runners per year, the odds for everyone are bound to decline each year. Extrapolating from recent growth patterns, we see that for a first-year applicant in the 2017 lottery the expected time until selection is about 5.25 years and possibly as many as 8 years. Attrition within the applicant pool will likely serve to keep the expected wait time bounded. However, it also means that most runners applying today will never actually be selected.

6. Recommendations

In the preceding sections we have analyzed several aspects of the Western States lottery. Based on these findings, the following recommendations are offered to the race organizers.

1. There is no need to use Monte Carlo simulation when computing the lottery odds. By using the method described in Section 3, the odds can be computed directly and accurately.
2. Reporting minimum and maximum number of draws is not particularly useful as a means of describing the range of possible outcomes, particularly for distributions with long, thin tails. In such cases, a better method of describing the likelihood of extreme events is to report percentiles, say the 5^{th} and 95^{th}.
3. The current scale factor of 2 does a good job balancing surprise and disappointment among the applicant pool as a whole. This should be maintained.
4. The exponentially-growing size of the applicant pool will contribute to declining odds across all categories of applicant. Assuming recent trends continue, we will soon reach a point where most applicants in the lottery will never actually be selected. Enhancing Western States as a "spectator sport" will be critical to keeping the interest of these runners and thereby maintaining the race's position atop the ultramarathon world.

7. Appendix

Appendix A

In this section, we compute the number of distinct lottery outcomes where we distinguish runners only by their category. For example, one outcome consists of 270 first-year applicants. Another outcome consists of 369 first-year applicants and one second-year applicant. Another outcome consists of 368 first-year applicants and two second-year applicants, etc.

We can accomplish this using generating functions. Let $G(x) = \sum p_k x^k$ where p_k is the number of distinct outcomes in a lottery with k rounds. Thus, we seek to compute p_{270}, which is the coefficient of x^{270} in the following polynomial:

$$G(x) = (1 + x + \cdots + x^{2233}) \cdot (1 + x + \cdots + x^{639}) \cdot (1 + x + \cdots + x^{377}) \cdot (1 + x + \cdots + x^{171}) \cdot (1 + x + \cdots + x^{71}) \cdot (1 + x + \cdots + x^{14}) \cdot (1 + x + \cdots + x^{5})$$

$$= \frac{1 - x^{2234}}{1 - x} \cdot \frac{1 - x^{640}}{1 - x} \cdot \frac{1 - x^{378}}{1 - x} \cdot \frac{1 - x^{172}}{1 - x} \cdot \frac{1 - x^{72}}{1 - x} \cdot \frac{1 - x^{15}}{1 - x} \cdot \frac{1 - x^{6}}{1 - x}$$

$$= (1 - x^{2234})(1 - x^{640})(1 - x^{378})(1 - x^{172})(1 - x^{72})(1 - x^{15})(1 - x^{6}) \cdot \frac{1}{(1 - x)^7}$$

The final term can be written as a geometric series with coefficients computed from the generalized binomial theorem:

$$\frac{1}{(1 - x)^7} = \sum_{k=0}^{\infty} \binom{k + 6}{k} x^k = \binom{6}{0} + \binom{7}{1} x + \binom{8}{2} x^2 + \cdots$$

It turns out there are 16 different ways to combine terms to yield x^{270} (though we won't enumerate them here). This leads to the following expression for the coefficient of x^{270}:

$$\binom{276}{6} - \binom{270}{6} - \binom{261}{6} - \binom{204}{6} - \binom{104}{6} + \binom{255}{6} + \binom{198}{6} + \binom{189}{6} +$$

$$\binom{98}{6} + \binom{89}{6} + \binom{31}{6} - \binom{183}{6} - \binom{83}{6} - \binom{26}{6} - \binom{17}{6} + \binom{11}{6}$$

$$= 12{,}705{,}435{,}449.$$

Appendix B

R code for simulating lottery results. T

```
applicants <- c( 2233, 639, 377, 171, 71, 14, 5 )
tickets.per.applicant <- 2 ^ ( seq_along( applicants ) - 1 )
race.capacity <- 270

num.categories <- length( applicants )
num.trials <- 1000

trials <- matrix( nrow = num.trials, ncol = num.categories )

for ( i in seq( num.trials ) ) {

  people.drawn <- rep( 0, num.categories )

  for ( j in seq( race.capacity ) ) {
    people.left <- applicants - people.drawn
    tickets.left <- people.left * tickets.per.applicant

    person.drawn <- sample( num.categories, 1,
                            prob = tickets.left / sum( tickets.left ) )

    people.drawn[ person.drawn ] <- people.drawn[ person.drawn ] + 1
  }

  trials[ i, ] <- people.drawn
}

num.taken <- sapply( seq( num.categories ), function( n ) mean( trials[,n] ) )

odds.of.selection <- num.taken / applicants

data.frame( category = seq( num.categories ),
            odds = round( 100 * odds.of.selection, 3 ),
            num.taken = round( num.taken, 1 ) )
```

Appendix C

R code for computing precise odds of selection in the 2015 Western States lottery.

```
applicants <- c( 2233, 639, 377, 171, 71, 14, 5 )
tickets.per.applicant <- 2 ^ ( seq_along( applicants ) - 1 )
race.capacity <- 270
num.categories <- length( applicants )

original.tickets <- applicants * tickets.per.applicant
ticket.counts <- applicants * tickets.per.applicant

for ( i in seq( race.capacity ) ) {

  prob.of.selecting.category <- ticket.counts / sum( ticket.counts )
  exp.ticket.reduction <- prob.of.selecting.category * tickets.per.applicant
  ticket.counts <- ticket.counts - exp.ticket.reduction
}

tickets.taken <- original.tickets - ticket.counts

odds.of.selection <- tickets.taken / original.tickets

num.people.taken <- odds.of.selection * applicants

data.frame( category = seq( num.categories ),
            odds = round( 100 * odds.of.selection, 3 ),
            num.taken = round( num.people.taken, 1 ) )
```

Appendix D

R code for computing expected number of duplicate draws

```
expected.duplicates <- function( num.dead, num.in.hat ) {

  alive.remaining <- num.in.hat - num.dead
  dead.remaining <- seq( num.dead, 0, -1 )
  ticket.counts <- seq( num.in.hat, num.in.hat - floor( num.dead ), -1 )

  dead.prob <- dead.remaining / ticket.counts
  alive.prob <- alive.remaining / ( ticket.counts - 1 )

  likelihoods <- cumprod( dead.prob ) * alive.prob

  sum(likelihoods * seq_along( likelihoods ) )
}

people <- c( 2233, 639, 377, 171, 71, 14, 5 )
weights <- 2 ^ ( seq_along( people ) - 1 )
race.capacity <- 270

ticket.counts <- people * weights

tickets.in.hat <- sum( ticket.counts )
dead.tickets <- 0
total.dupes.drawn <- 0

for ( i in seq( race.capacity ) ) {

  dupes.drawn <- expected.duplicates( dead.tickets, tickets.in.hat )
  total.dupes.drawn <- total.dupes.drawn + dupes.drawn
  dead.tickets <- dead.tickets - dupes.drawn

  prob.of.category <- ticket.counts / sum( ticket.counts )
  exp.ticket.red <- prob.of.category * weights
  ticket.counts <- ticket.counts - exp.ticket.red

  newly.dead.tickets <- sum( exp.ticket.red - prob.of.category )

  tickets.in.hat <- tickets.in.hat - 1
  dead.tickets <- dead.tickets + newly.dead.tickets
}

sprintf( "Total draws required: %.1f", race.capacity + total.dupes.drawn )
```