International Baccalaureate

Math HL Internal Assessment

A Statistical Approach to Cross Country Running

Introduction

Mt. San Antonio College; November 20th, 2021, 7:45am.

I nervously breathed in the cold air as we shuffled to the starting line of the Division 3 CIF Southern Section Cross Country Championships. I glanced back at my coach, expecting some advice, but all he had for us was a brief: “make it happen, guys.”

Throughout my four years on my school’s varsity cross country team, I had seen the tremendous amount of work the team had put in to make it this far: running mile after mile, training year round, and fighting our way through the most competitive races in the country. This was it. This was our chance, for the first time in school history, to make it to the CIF State Championship. As we stood on the line, we surveyed every other team that aimed to do the same. There and then, our coach trusted us to make the right decision.

Explanation of Cross Country Scoring

While it initially appears that the objective is simply to run fast, success in cross country is dictated by the sport’s complex scoring system.

A race consists of multiple teams, each with 7 runners. Upon completion of the course (normally a 3 mile length with varying terrain), the place in which each individual runner comes in is assigned to them. From there, each team’s score is calculated by finding the sum of the score of their first 5 runners.

Hence, the score of any given team, S_{team}, can be expressed in the equation below when p_i is the place received by the ith runner on the team to finish:

$$S_{team} = \sum_{i=1}^{5} p_i$$

The placing of the teams is then the inverse of the scores. (ex: the team with the lowest score gets first place)

There are also a few rare edge cases that exist:

If for any reason, a team doesn't have 5 runners complete the race, the team is disqualified and all of their runners are removed from the placing. Thus, the runner in question loses their place and all runners with higher places are decreased by one. Based off p_{DQ}, the place of the disqualified runner, we can find p_{new}, the new place of any given runner from p_{old}, their old place, using the following logic:

$$\text{If } p_{old} > p_{DQ}, \text{ then } p_{new} = p_{old} - 1$$

In a race like the one we are analyzing, this is almost unheard of. Any experienced competitive team will do anything possible to ensure that at least 5 runners cross the line. There are, however, a few individuals who qualified for this race on time alone, but for our purposes (optimizing team score), they are insignificant and therefore removed from our initial data set.

Also, during the final comparison of scores, if any teams have the same score, the team place is awarded based on comparing each team's 6th runner to finish. This can be expressed by the following logic:

$$\text{If } S_{teamA} = S_{teamB}, \text{ then (If } p_{teamA,6} < p_{teamB,6}, \text{ teamA wins; If } p_{teamA,6} > p_{teamB,6}, \text{ teamB wins)}$$

Since both team's sixth runners can't occupy the same place ($p_{teamA,6} \neq p_{teamB,6}$), a winner is always decided by this method.

Situational Factors

In this particular case (Division 3 CIF Southern Section Cross Country Championship Final), all 16 teams have a single goal: to be within the top 7 teams, qualifying for the State Championship.

For most teams, this would be their last race of the season, so they were already well dialed in with the positions assigned to each runner and the subsequent strategies. As this was a very competitive elimination race, there was no reason to hold back.

It was a particularly cold morning and recent concerns had surfaced that the starting area was dangerously narrow. These factors surely led to inconsistent starts and many uncontrolled variables. On the initial start of the race, Oak Park's anchor collided with one of West Torrance's blockers, taking him down hard. The race was recalled, with the blocker from West Torrance taken out on account of a broken collarbone and both of Oak Park's blockers scraped up. The time that passed during this likely led to many runners getting colder, making the second start more inconsistent. This led to a jostled first half of the race, making the main strategic decision at the halfway split.

The halfway split consists of a clock with a digital timing system proceeding the steepest hill. From there, both team and individual places are announced. Given this information, each runner must decide whether or not to aggressively run (push) the second half of the course. Doing so has the potential to make up time and gain many places, but it also runs a high risk of burning out and losing places.

Outlining Paper Objectives

My goal in this paper is to simulate the possible outcomes of the Division 3 CIF Southern Section Cross Country Championship Final to determine the optimal strategy for each team. I will then compare this to existing strategies to assess their effectiveness.

For the purposes of this paper, we will define a strategy as a permutation of 7 binary decisions (one for each runner on the team) of whether to push or not. A strategy is considered optimal if it maximizes the probability of the team scoring below 192 points to finish in the top 7 and move on.

Explanation of Positions and Existing Strategies

While all runners on the team start the race in the same way, their function greatly differs. This is because common strategies generally lead to each runner being assigned a position on the team. Besides being the nth to finish for the team, their roles are completely different. For the purposes of our simulation, we'll need to assign each a $P(hang)$, the probability that they can hang onto their push, and a s_{hang}, the second half split (proportional to the first) they will likely achieve if they do so. We also need to have an s_{burn} value to predict their second half split on the chance that they fail to hang onto their push, which is subsequently (1 - P_{hang}).

Fronts consist of the first 2 runners on the team. They are mostly standout runners that are geared towards longer distances. Their job is to go out hard and establish a good pace for their team. Often, their experience leads them to be highly consistent, but since they are pushing the edge of physical performance, they have little to gain. They are expected to get lower places, so a failure to do so forces the rest of the team to take more risks.

Mids consist of the 3rd and 4th runners. They have many of the qualities of a standout distance runner but lack either the confidence or experience to do so. When necessary for the team score, they can sometimes push forward and hang onto a front. However, they often deal with consistency issues when doing so.

The Anchor is the 5th runner. Since he has the highest place of any scoring member, his performance often solely dictates the team result. Within the last mile, he must make up any deficit produced by the previous runners. Due to the pure grit and responsibility that comes with the role, if absolutely necessary, the anchor has the potential to push hard, but due to his importance, taking any risk is generally avoided. If a mid falls back, he may have to pace with them to maintain a decent team score. A newer strategy involves stringing along the blockers as insurance to improve consistency on the back end.

Blockers are the 6th and 7th runners. Although they likely won't be factored into the team score, they need to be ready for anything. There's a large divide in how these are typically run. Some choose to push aggressively in hopes of disrupting the other team's scoring or slightly improving on the anchor's place while others play it safe and sit behind the anchor to pick it up if things go wrong. These are often the most inexperienced and inconsistent runners on the team, leading to highly mixed results.

Using these factors, I came up with the following values for each role:

Position	$P(hang)$	s_{hang}	s_{burn}
Fronts	0.95	0.86	0.91
Mids	0.85	0.84	0.93
Anchor	0.90	0.85	0.91
Blockers	0.75	0.83	0.95

Approximations are centered around our control value: s_C = 0.88, representing the average expected 2nd half split of a runner who chooses to not push. I found this by dividing the sum of second half splits from the first half splits. I then looked at known points: runners who we know pushed at that point, either hanging on or burning out. Constantly tweaking these values as I went through our known points and comparing it to our conceptual understanding arrived at a model that closely reflected real-world performance.

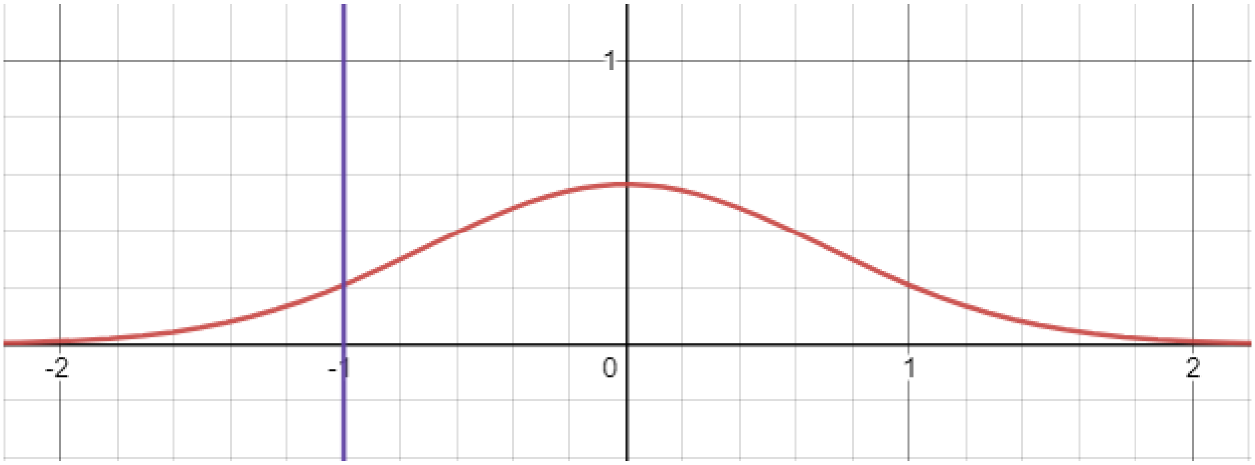
Using Data to Model Distribution over Time

Now that we can model the variation in an individual by their decision, we need to model the variation in other runners not employing our strategy. Since our goal is to discover a cutting edge-strategy (first-layer reasoning), we are operating on the assumption that all other teams would execute the same strategy they already did. However, this leaves a variable that we can't control for: the other runners kicking. In endurance running, it's an accepted practice that if anyone is nearby when approaching the line, to let out an uncontrolled sprint, therefore "kicking". This has the chance to gain the runner a place without risking burning out. Since this maneuver is dependent on the presence of another runner, changes in a runner's time due to our strategy will directly lead to a slight variation in the times of the surrounding runners.

Instead of using a distinct point to model each runner, I'll instead use a continuous density function to model their distribution. After some research, I found that a Kernel would be the best function to use. A Kernel is a symmetric function used to estimate the density of a nonparametric regression (more about this later). It's perfect for our use because by definition: at all values of x, the density can't be negative since you could never have a negative number of runners crossing the line at an instant, and the area under the curve equals one, corresponding to the one runner it represents. Since I know I’m going to later integrate this, I chose to use a particular kernel: the Gaussian kernel, based on e^{x}. The kernel function, $f_n(x)$, for a runner n that ran the time of x_n, with h as a variable bandwidth coefficient, is as follows:

$$f_n(x) = \frac{1}{h\sqrt{2\pi}} e^{-0.5\left(\frac{x-x_n}{h}\right)^2}$$

When graphed, the function appears as so [as: $x_n = 0$; $h = \frac{1}{\sqrt{2}}$ (value explained later)]:



Most of the distribution falls within a second or two, which is consistent with the approximate striking distance of a kick. Now, we can insert a vertical line at $x_i = -1$ to represent our runner that was placed one second ahead (earlier) by our strategy. While being inside the other runner might seem odd, we can actually find the area of the sections created to compute the probability of our runner getting passed and losing the spot to runner n, $P_{lose:n}$. It’s expressed as the general integral:

$$P_{lose:n} = \lim_{k \to -\infty} \int_{k}^{x_i} \frac{1}{h\sqrt{2\pi}} e^{-0.5\left(\frac{x-x_n}{h}\right)^2} dx$$

Further computing this integral leads to an application of the Gauss error function, denoted as $erf()$. It's defined for any value of x as:

$$erf(x) \;=\; \int \frac{2e^{-x^2}}{\sqrt{\pi}}$$

The final integral comes out to:

$$P_{lose:n} = \frac{erf\left(\frac{x_i - x_n}{\sqrt{2}h}\right)}{2} - \left(\lim_{k \to -\infty} \frac{erf\left(\frac{k - x_n}{\sqrt{2}h}\right)}{2}\right)$$

This allows us to plug in any time, x, we want for our runner and find the probability of runner n beating them. For the particular example graphed above (a 1 second lead going into the kick), $P_{lose:n} \approx 0.07865$. After examining the integrated equation, it becomes obvious that the convenient value of $h = \frac{1}{\sqrt{2}}$ greatly simplifies the process.

We can now combine multiple of these kernels to create a nonparametric distribution, a continuous distribution which isn't based on any known distributions. Achieving this is quite elegant actually. Finding an individual's place, p_i, in the race is just how many people they lose to, so we can simply find the sum of the earlier probability for each runner in the race.

$$p_i = \sum_{n=1}^{111} P_{lose:n} = \sum_{n=1}^{111} \frac{erf\left(x_i - x_n\right)}{2} - \left(\lim_{k \to -\infty} \frac{erf\left(k - x_n\right)}{2}\right) \approx \frac{1}{2}\left(\sum_{n=1}^{111} erf\left(x_i - x_n\right) - \sum_{n=1}^{111} erf(-\,x_n)\right)$$

The final estimation shown above comes from the understanding that we can't have a negative x since nobody can run negative time for the race. While we could account for the negative area as indicative of an additional section of the distribution, this adds much more complexity to our formula while not adding any significant value. Even for our smallest value, $x_n = 888$, I can't find a conclusive nonzero area within the negative bounds.

While it may be concerning to calculate someone is in 31.5th place, we need to understand what the 0.5 represents. It’s not that he’ll always lose to half a person, but rather half the time he’ll lose to one person. When our goal is to find the expected value, these seemingly nonsensical values don’t raise any issues as they’re just a representation of the mean.

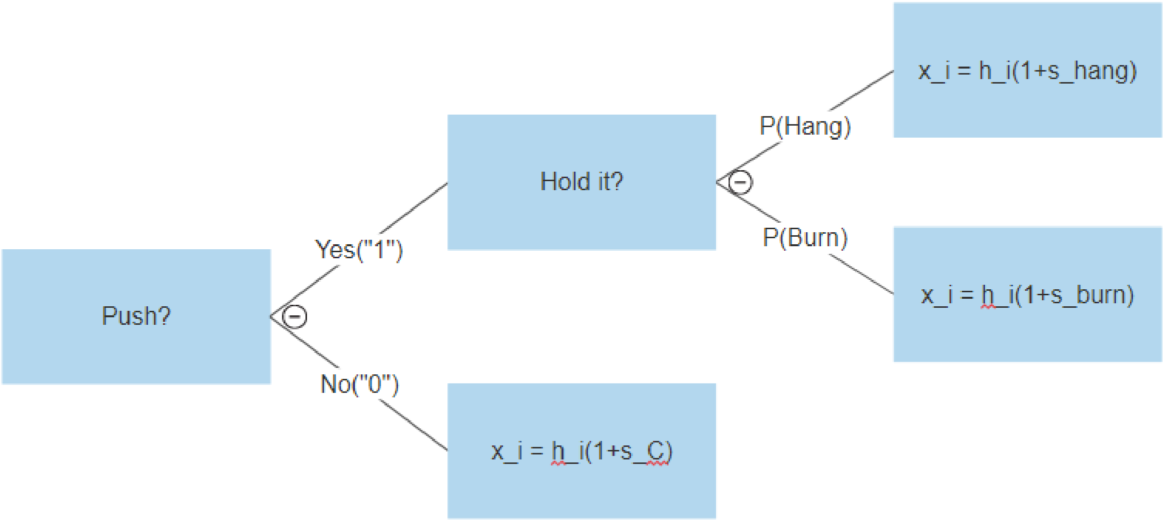
Additional concerns may surround the fact that the distribution doesn’t remove the runner we are applying the strategy to, allowing someone to get 112th place in an 111 man race. Much like the approximation earlier, this is necessary to simplify our system enough to create meaningful data since recalculating the distribution every time will be a tremendous waste of resources. Our distribution however, is still sound as assuming the 1/112 times someone nonsensically kicks against themselves, they may have seen the clock and instead are racing to beat their own goal time.

Calculating Simulated Score for Each Strategy

Now that we have models to calculate a runner’s time by their strategic choice and their place by time, we can calculate the team result for any given strategy. We do so by plugging the new value of p_i into our original score equation.

$$S_{team} = \sum_{i=1}^{5} p_i = \frac{1}{2}\sum_{i=1}^{5}\left(\sum_{n=1}^{111} erf\left(x_i - x_n\right) - \sum_{n=1}^{111} erf(-\,x_n)\right)$$

We’ll come back to this equation later because we first need to resolve what’s within the parentheses. Each value of x_i and its respective probability (given the choice), with h_i representing the known first half split, can be found with this decision tree:



If we now represent each runner on the team's choice as binary (either 1 to push or 0 to not), we can model a strategy as a 7 digit binary permutation (ex: $strategy = 0010011$). The method to calculate the probability of qualifying with a given strategy, $P(qualify|strategy)$, is as follows:

- Set $P(qualify|strategy)$ to 0.
- For each of the 2^(amount of 1's present in the strategy) different outcomes resulting from pushing:
    - For each runner:
        - Calculate x_i, the time that would be achieved using the result given in the outcome (hang, burn, or none), using the above figure.
        - Plug x_i into our formula earlier to find p_i.
    - Find the score, S_{team}, which is the sum of the places of the 5 runners with the lowest p_i. It'll likely be easier to remove the runners with the 2 highest p_i than to find the 5 highest values.

- If $S_i < 192$ (our established point threshold to qualify), add the probability of the outcome occurring (ex: $P(hang) * P(burn) * P(burn)$...) to $P(qualify|strategy)$.

Large Scale Simulation and Data

To create meaningful data, we must scale this up and automate the process. I've applied our method to a python script to achieve this, finding every $P(qualify|strategy)$.

We can significantly improve efficiency by calculating $\sum_{n=1}^{111} erf(-\ x_n)$ beforehand. We can also calculate each runner's x_i, given that they hang, burn, or don't push at all, once and use it multiple times. Below are the 3 best and 3 worst strategies (as simulated) for each team:

Pos.	Team	Best	3 Best Strategies			Worst	3 Worst Strategies		
1	West Torrance	1.0000	000011	001110	010010	0.9999	111111	111110	101111
2	Brea Olinda	1.0000	0000111	0011100	0100101	0.9999	0111111	1111110	1111111
3	Santa Margarita	1.0000	0100101	1000101	1000110	0.9962	0010110	0001110	0010111
4	Torrance	1.0000	0100101	0100110	1000101	0.9625	0001010	1111000	1111001
5	St. John Bosco	1.0000	1000101	1000110	0000000	0.9772	1111011	1111000	1111010
6	Palos Verdes	1.0000	0000001	0000010	0000011	0.8500	1110001	1010001	1010000
7	Capistrano Valley	1.0000	0000001	0000010	0000011	0.7225	1011000	0011000	0101000
8	Agoura	0.9929	0101101	0110101	1101101	0.0000	1000000	0100000	0000000

9	Pasadena	0.9403	1111100	1111101	1111110	0.0000	1100011	1100010	1100001
10	Dos Pueblos	0.6177	0111101	0111110	0111111	0.0000	1111011	1111010	1111001
11	South Torrance	0.0000	0000001	0000010	0000011	0.0000	1111111	1111110	1111101
12	San Marcos	0.0000	0000001	0000010	0000011	0.0000	1111111	1111110	1111101
13	Buena	0.0000	000000	000001	000001	0.0000	111111	111111	111110
14	Redlands East Valley	0.0000	0000001	0000010	0000011	0.0000	1111111	1111110	1111101
15	Oak Park	0.0000	0000001	0000010	0000011	0.0000	1111111	1111110	1111101
16	Servite	0.0000	0000001	0000010	0000011	0.0000	1111111	1111110	1111101

Note that for teams 11 through 16, all strategies had the same success probability, absolute 0, so the ones shown are just the ones the computer algorithm listed first. However, they do still have many similarities to the best and worst of other teams.

Analysis

While we may be tempted to use a linear regression model to represent the relationship, we need to understand that it just doesn't translate well to this situation. First off, our output (the individual strategies) are discrete in nature, since each value is either a zero or a one. So even if we isolate it down to just whether or not someone pushes, it won't fit a linear model, but a piecewise function. Also, the correct strategy for an individual runner on a team is highly dependent on the rest of the team, so we have to look at it as a whole.

In our case, the main input variable is the team's overall placement within the race at the halfway split. I'll divide these based on my observations into 4 main groups.

1st through 5th. These commonly consist of teams that were strong throughout the whole season due to their consistent pack. They've gotten out hard and plan to stay ahead. The best strategy here appears to include pushing one of your fronts. This appears to keep the rest of the team consistent but have the possibility of a much better result without running the risk of losing any member of your scoring 5 (even if they burn out, they won't fall behind a blocker). Pushing both a blocker and the anchor seems to also help build consistency since only one of them needs to finish in a good position for the team score. However, top teams like West Torrance and Brea Olinda had nothing to gain by pushing one of their front runners, instead pushing more blockers and taking absolutely no risk on the front end.

6th and 7th. This is commonly what's considered the bubble, holding the last 2 qualifying spots. In this case, a perfect strategy would guarantee moving on with 100% effectiveness. These strategies consist of an extreme emphasis on consistency, choosing to not push any of your scoring runners and only push the obvious single blocker. However, these teams were top-rated teams, taking 1st and 2nd at the preview race only a month earlier, so they both clearly cared about winning the race outright. While what they chose wasn't the absolute worst strategy (pushing the front runner, both mids, and having no anchor to fill in, leading to an 85% for Palos Verdes and 72.25% for Capistrano Valley), they defaulted to a common strategy to try to tie up loose points: executing a hard push with the 5th and 6th runner. Basically, you pack them together, try to build momentum off eachother, and embark on a reckless push for any places you can get. For Capistrano Valley, this worked as the 5th and 6th took 34 spots in the 2nd half, but Palos Verdes' 5th lost 21 spots. As a result, Palos Verdes finished 8th and didn't qualify for the

State Championship, which was a major upset. While they had varying levels of success, both could have employed a better strategy to completely ensure success.

8th, 9th, and 10th. Being right outside a qualifying spot, these teams still have a chance to make it, but only if they employ the correct strategy. In this case, they need to rely on a good performance out of almost all members, so aggressively pushing almost the whole team is the best strategy. The chance of it working out is relatively low, but it's still better than the 0% chance they had if they didn't push the 3rd and 4th. It's risky, but necessary. Agoura played it well to get the 6th spot and Dos Pueblos' high risk paid off by snagging the 7th spot.

11th through 16th. These teams are unfortunately so far back that no amount of strategy could allow them to qualify. In terms of placing higher, pushing a blocker has absolutely no risk, but could improve team score. If planning pre-race strategy, you must avoid being within this group at all times since you'll be left with absolutely no chance.

Despite almost guaranteeing a spot through the split, 4th place team Torrance employed what's close to the worst possible strategy for their position. They attempted to push their first 4 runners without the anchor or blockers to back them up, so when it went south, they dropped to 9th, losing their spot. This is an example of what was consistently the worst possible strategy for teams within the top 7: pushing the 3rd and 4th without pushing the 5th. Since they already had a spot, they had nothing to gain, and burning out will lead to the team being short of a scorer. Generally, pushing your mids is taking an unreasonable risk, especially if there is no chance of a good performance by the anchor to make up for it.

Pushing at least one blocker was always included in the optimal strategy. This makes sense because they wouldn't score otherwise, but a good performance has the chance of improving the team's score with little to no risk.

Comparison with Existing Strategies

First off, it’s an extremely common practice to push a blocker due to the simple logical benefits stated above. Our simulation verifying that certainly adds merit to the strategy. However, the continual success of the strategies involving pushing both the 5th and 6th was quite surprising. Typically, it’s only seen as a reckless title-hunting measure, but it is gradually becoming more common. The common argument against it is that it’s never worth risking the pivotal performance of your anchor, but what we observed indicates that often it’s okay to accept some risk since in the event the anchor dies, either the 6th that successfully hung on or the consistent 7th will score for minimal loss.

Another unexpected outcome was how much it decreased your odds of qualifying by pushing the mids. Many people see pushing both mids as a way to create distance and make up for an underperforming anchor, but our simulation found that it’s actually the worst thing to do. This however aligns with the pattern that most elite teams are successful due to the consistency of their mids, so anything to risk their performance likely isn’t worth it.

Playing it extremely safe and not pushing when safely ahead is a common practice, so our data verifying it is no surprise.

People often get nervous when being on the bubble, leading them to unnecessarily push, while our results showed that the best strategy is to play it as consistently as possible to try to expand your lead. However, this might be flawed since we aren’t accounting for those in contention for the spot.

It’s pretty commonly accepted that if you’re out of the qualifying spots, you must do anything possible and take any risk necessary to make it. This almost completely matches our data, but our data seemed to show that pushing the 1st runner didn't do much since there's only so many spots he can possibly gain.

Conclusion

I gained a lot of insight from this investigation. While many of the results found were new and unexpected, they were all justifiable and were valid strategies. Applying math to a topic I'm very familiar with and interested in was an amazing opportunity to understand the many practical applications of math.

If possible, I would have liked to improve on a few aspects of my experiment. Instead of using a simple piecewise function to model the result of each runner's push, I would rather have used a probability density function to create more continuous data. Given the additional computing power and data to train these functions, it could provide additional insight into the more complex ways into which we can take risks.

Additionally, in terms of game theory, this is all level 1 thinking. We are operating under the assumption that other people won't have or alter their own strategies in relation to ours, which truly isn't the case in real-world scenarios. Widespread acceptance of the strategies discovered in this investigation could lead to other strategies actually becoming more effective to combat the new ones. All strategies outlined here are under the assumption that the other teams continue to use the same strategies.

This simulation could possibly work past a simple qualifying scenario, but a new success criteria would have to be determined to evaluate the viability of each strategy. Strategy is a constantly changing world, and this is just a single step in it.

Works Cited

Niranjan Pramanik, Ph.D. “Kernel Density Estimation - Kernel Construction and Bandwidth Optimization Using Maximum...” Medium, Analytics Vidhya, 1 Oct. 2019, https://medium.com/analytics-vidhya/kernel-density-estimation-kernel-construction-and-bandwidth-optimization-using-maximum-b1dfce127073.

“Results.” Finished Results - Timing Track &amp; Field, Cross Country, Road Races, Cycling, Swimming, Paddle Board, CIF Southern Section, 20 Nov. 2021, https://finishedresults.trackscoreboard.com/meets/10131/events/5507.