Swim, Bike, and Run

Harin Lim, Trey Wiedmann, Evan Zhang

"Oink."

This running pig that is shooting fireballs was generated by a scatter plot in one of Evan's data science projects. We thought it was a cool illustration and wanted to show you it. It kind of represents our problem, since our problem is about running, but really it's just something cool that we wanted to show :)

Summary Sheet

In this paper, we explore methods for planning a triathlon event to minimize congestion throughout the course, and to minimize the time that the city must close its roads for. We decide to divide the event into two days, rather than completing it in a single day. Then, we propose two different plans: one that focuses on providing an optimal competitive environment, and one that accommodates for less competitive athletes. Similar to most triathlons, both plans divide athletes into divisions based on competitiveness, and then group people in waves and stagger the start times of these waves. The competitive plan orders divisions in increasing average finish time on each day, which helps minimize congestion on the course. However, putting the slowest people last may not be fair to them, as our race will have a time limit where we absolutely must stop the race to reopen the roads, and the slowest divisions will be more likely to exceed the time limit. Thus, the more accommodating plan lets the slowest divisions go first, followed by the rest of the competitors, giving them the best chance of completing the race on-time. Our complete race itenary for both plans is written in the Analysis section of our paper.

To evaluate these different plans, we use an existing dataset provided by the problem statement. With a Python script, we simulate the two plans in order to measure how long the race would last and how spread out athletes would be by the end. We also

calculate how often competitors pass each other in the race, which indicates how competitive and also how congested a race will be. By adjusting parameters of our models, we used these simulations to decide the best timings to use in our plans.

Problem Statement

This paper focuses on problem A part I of this past HiMCM competition: https://www.comap.com/highschool/contests/himcm/2016problems.html

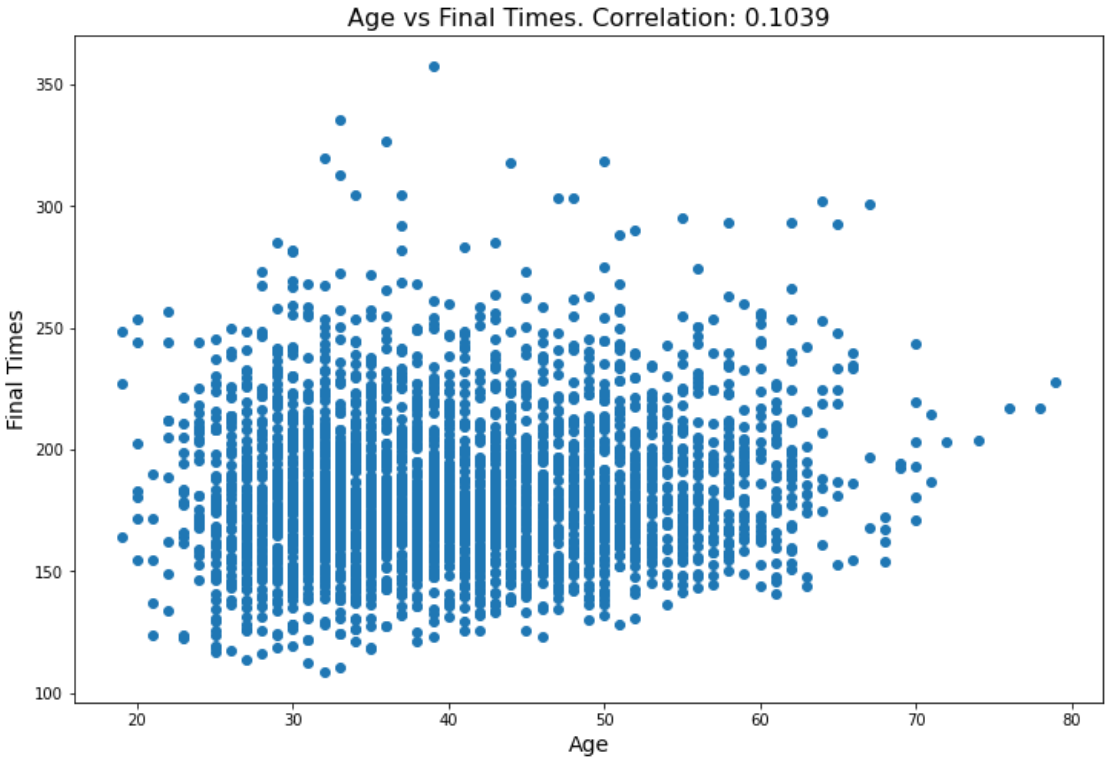
A triathlon is a multi-stage athletic race of endurance. It consists of three continuous events: swimming, biking, and running. Athletes compete for the quickest time to complete all three sections, including transition times between each section.

In this problem, we are planning for a triathlon of about 2000 competitors. The triathlon consists of a 1500m swim, a 40K bike ride and a 10K run. In order to hold the race, we most close local roads, but we cannot keep them closed for longer than 5.5 hours. We also must keep the race enjoyable for participants. Thus, we are asked to minimize both course congestion and road closure time. Traditionally, triathlons divide races into divisions of athletes of similar competitiveness. They also have athletes begin the race in waves, meaning that competitors cross the starting line in small groups at intervals rather than the entire race starting all at once. We must determine the divisions that our race has, and a schedule of wave start times.

We are given a data set from a previous triathlon to help plan for our upcoming one, which consists of 3217 people. Each entry consists of the contestant number, age, gender, and category, as well as their times spent on swimming, biking, running, and the two transition times. This race has a few categories. The first category is the male and female pro divisions, which consists of professional athletes. The next category is the male and female premier divisions, which are very competitive but not professional athletes. Then there are the male and female open divisions, which don't have restrictions to enter. Finally, there are the Athena and Clydesdale divisions, which are women over 165 lbs and men over 220 lbs respectively.

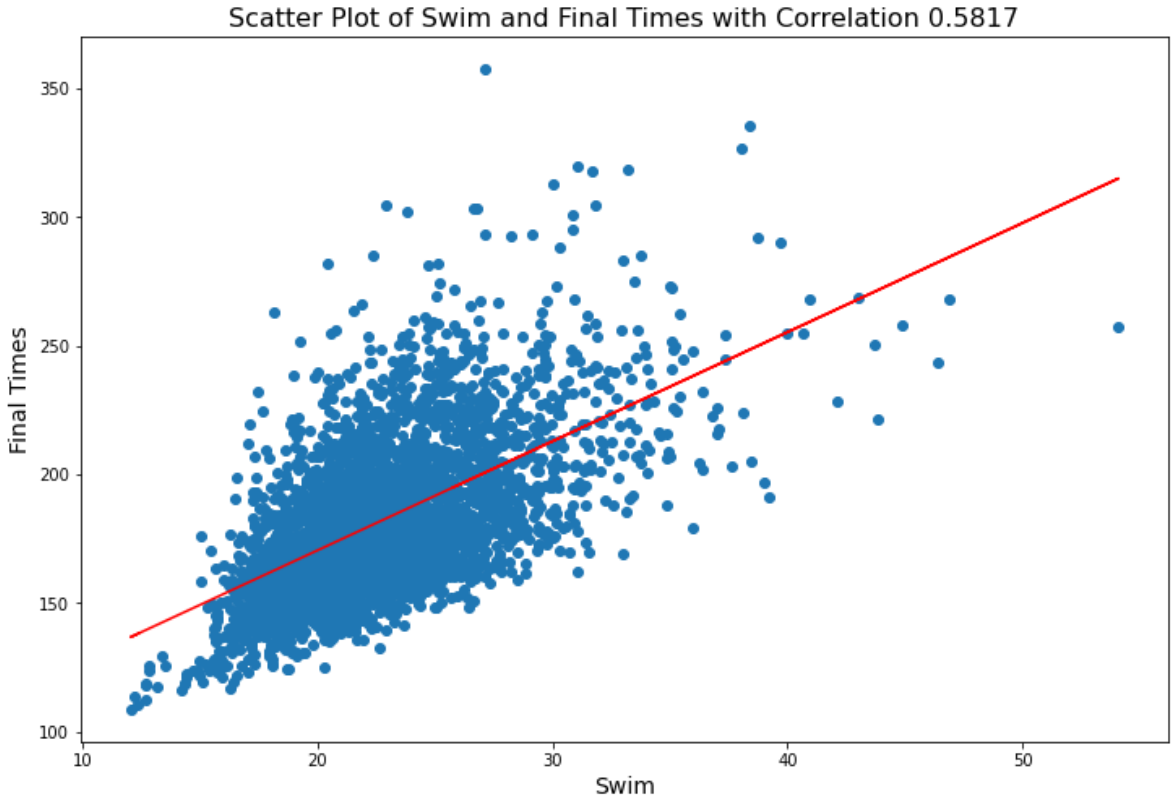
Exploratory Data Analysis

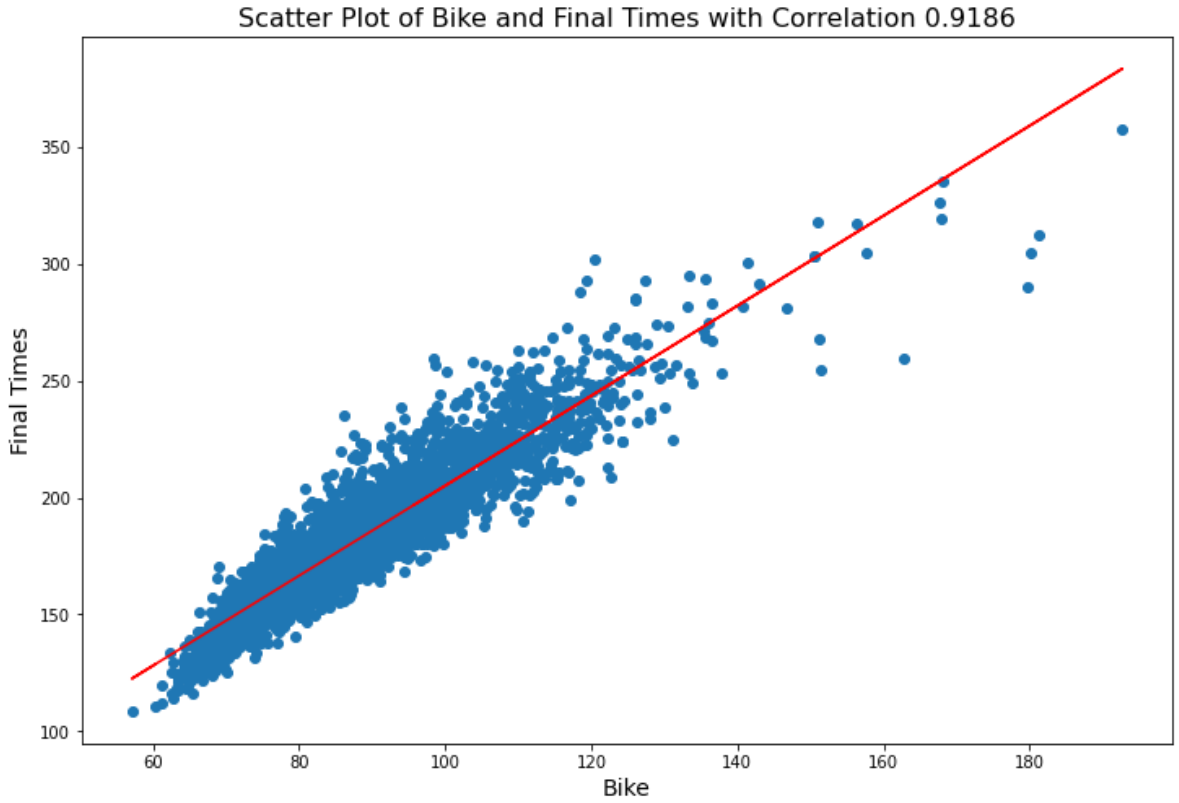
Before we begin determining the assumptions, objectives, and optimal methods for this problem, we decided to explore the data and determine what features we would like to use. The first, most intriguing question is whether age is a useful predictor of finishing times. As a baseline hypothesis, it would be reasonable to assume that those who are older would take, on average, a longer time than younger people, since older people generally become more fatigued and are not as athletic as younger people.

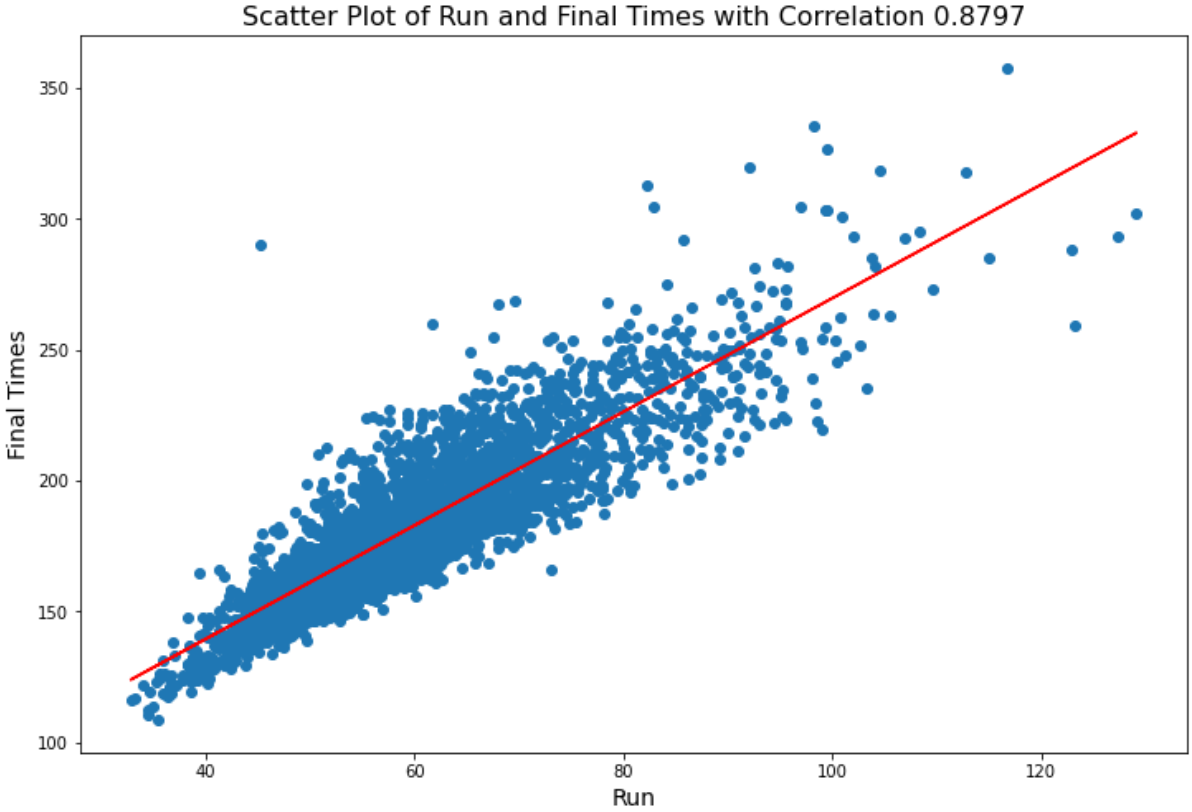


However, as seen in the scatter plot above, such an assumption would be faulty. From the graph, it seems like the fastest people are below the age of 50, but there is no clear correlation between age and final times. In fact, the correlation value for these two features is 0.1039, where an absolute r value above 0.7 implies significant correlation, and an absolute r value below 0.3 implies insignificant correlation. We could perhaps assume that the unexpectedly low correlation value is due to the fact that older people who participate in triathlons are probably fitter relative to their age. An old person who knew they don't have the athletic ability to complete a tedious triathlon would most likely not try, whereas a younger person may want to participate in a triathlon for the once-in-a-lifetime experience. This is why an initial exploratory data analysis is important, as the data oftentimes can contradict our previous beliefs, and force us to rethink the assumptions behind our reasoning.

Next, we will explore whether a participant's swim, bike, and/or run times are correlative with their final times, and which of those three features are most correlative. In other words, if we only had the information for one of the participants' subtimes, which subtime would allow us to best predict their actual final time? Those values are below:

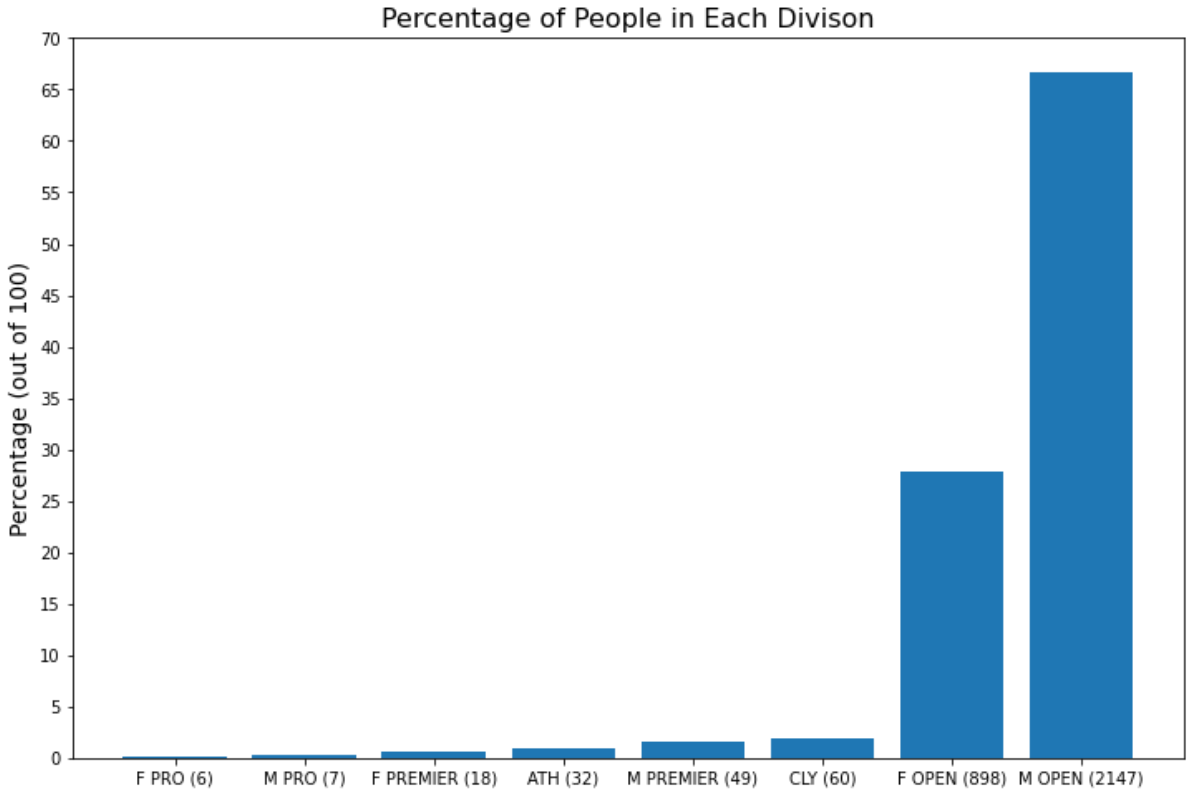



Scatter Plot of Bike and Final Times with Correlation 0.9186
Final Times
Bike
100
150
200
250
300
350
60
80
100
120
140
160
180


Scatter Plot of Run and Final Times with Correlation 0.8797
Final Times
Run
100
150
200
250
300
350
40
60
80
100
120

Overall, these results make sense. Because the swim is the first thing that athletes do in the triathlon, it is not very indicative of the athlete's endurance for an extended period of time, and endurance is very important in a triathlon that takes even the best professionals around 2 hours to complete. Furthermore, the swim is the shortest section of the triathlon, and so even the slowest swimmers aren't that much slower the fastest swimmer, whereas the slowest bikers/runners are much slower than the fastest bikers/runners. In mathematical terms, the range of swim times is much lower than the range for bike/run times. On the other hand, a participant's bike time and run time are more indicative of their final time, for reasons already described above.

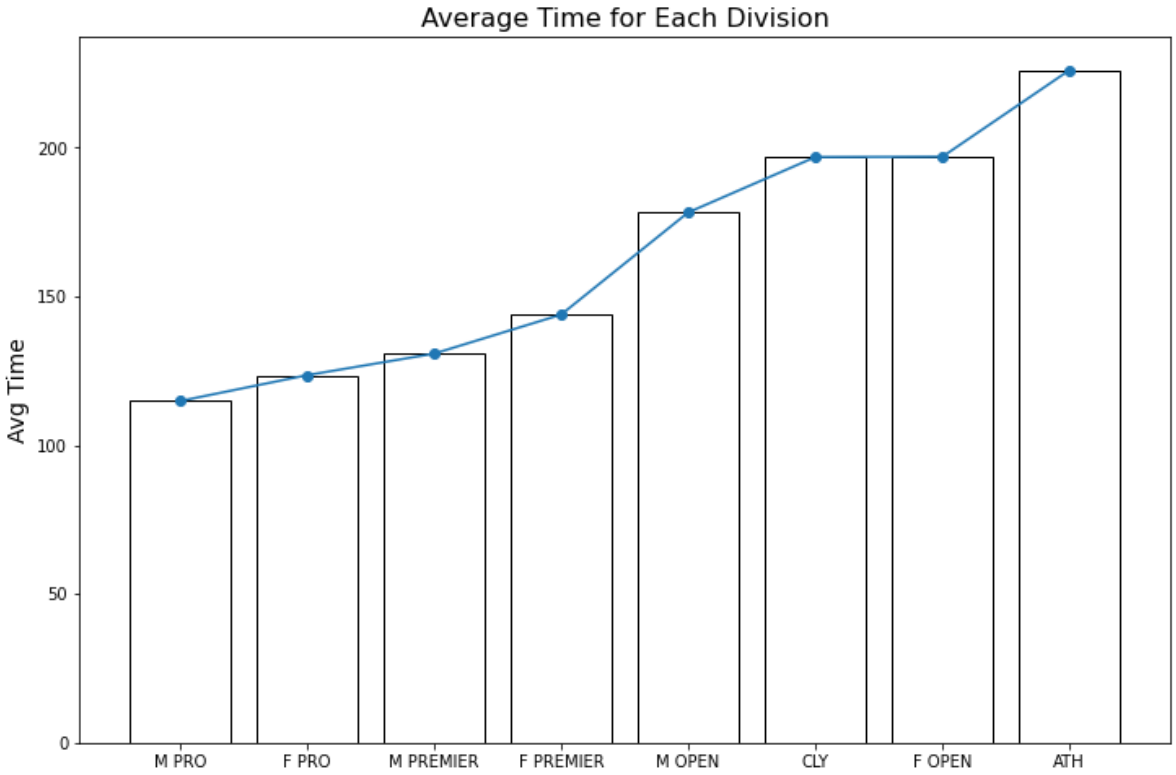
Finally, we need to explore differences between each division. We can start with differences in the number of people that are in each division, which is shown below:



Impressively, two thirds of the runners from the dataset are in the Male Open division! The Female Open division comes at a far second, and the other professional/special divisions are even less populated. This is important to keep in mind when we are determining what divisions should go first/last, and when determining the target objectives which motivates the triathlon scheduling. Since we are modeling our triathlon based on this previous triathlon, and this previous triathlon seemed more

casual (with far less professionals compared to the open section), our triathlon will consequently be just as casual.

The average time that triathletes take in each division is also important, and can be seen in the graph below:



These results are unsurprising; the people in the professional divisions are the fastest, regardless of gender, followed by the premier division, open division, and special division.

Definitions

Group/Wave: People who start the triathlon at the same time and on the same day. This is different from the word "division", which denotes a group of people that are labeled under the same division and run on the same day (**divisions can be split into groups**).

Collision: A person experiences a collision when he or she is overtaken, or overtakes another person in the race. Note, this does not mean they physically collide.

Leg/Section: Each portion of the race, i.e. swimming, biking, and running.

Wait/Start Time: The delay between the beginning of the race and when a contestant is allowed to cross the starting line with their wave.

Assumptions

Assumption 1: We expect around 2000 people for our planned triathlon.
Justification: The problem statement tells us that we should expect 2000 people for this triathlon.

Assumption 2: We cannot keep the local roads closed for more than 5.5 hours. Local roads are closed when the first wave of people start the triathlon.
Justification: We are told that we can keep them closed for no more than 5.5 hours (330 minutes). Even though the first people don't reach the roads until they finish the swimming section, we don't know how long the fastest swimmer will take. Thus, we must close the roads before the triathlon starts to guarantee that there is enough time to close them.

Assumption 3: The number of people in each division for the triathlon we are planning is proportional to the number of people in each division for our model triathlon. We therefore have to scale all the data by a factor of 0.621 (2000/3217).
Justification: We were given a previous triathlon to base our models off of, so it is reasonable to keep the amount of participants in each division for our triathlon proportional to the original dataset.

Assumption 4: We can split the triathlon into two days.
Justification: Final placements are determined by final times, not actual placement in the race, so it is fine to have competitors in the same race or even division compete on different days. We interpreted the 5.5 hour limit on road closure to be per day rather than overall.

Assumption 5: The groupings we suggest for the two days can be swappable (it doesn't matter if the people on Day 2 run on Day 1, and Day 1 people on Day 2), and weather conditions will remain the same on both days.
Justification: It shouldn't matter what group runs on Day 1 or Day 2, and including weather adds a random extraneous variable which serves no meaning.

Assumption 6: Competitors will have a constant speed during each segment of the race.
Justification: In reality, runners do not run at constant speeds. However, we make this simplifying assumption to allow us to calculate collisions between runners.

Assumption 7: Competitors within the same wave start at the exact same position in the race and at the same time.
Justification: Most likely, the starting line will be large enough to allow an entire wave to begin at the same time.

Assumption 8: If a competitor is taking longer than 5.5 hours to complete the race, we are allowed to ask them to stop.
Justification: If they begin to exceed 5.5 hours, we will need to reopen the roads, so we have to ask them to stop. Many other races implement similar cutoff times.

Objectives

The town's Mayor is focused on two objectives:

1. Minimize the congestion on the course.
2. Minimize the length of time the local roads in the town are closed for the cycling and running portions of the triathlon.

The Mayor did not explicitly ask us to focus on either of these two objectives. Therefore, we propose two different methods, one of which focuses more on Objective 1, and the other of which focuses more on Objective 2. In fact, focusing on minimizing the congestion on the course makes the triathlon more **competitive**, as with less people grouped up, runners will feel more comfortable and be able to more easily run at their pace. Furthermore, comfortability not only means minimizing congestion on the course, but also minimizing the number of collisions between athletes (more on that in the Methods section). Minimizing the length of time that the local roads are closed makes the triathlon more **fair**, as it would mean that people who are slower would be able to start earlier. In this method, slower people wouldn't feel the rush and burden of being the last ones left in the triathlon. Each of these methods have their own strengths and weaknesses, and which method the Mayor uses is strongly dependent on the primary motive behind hosting the triathlon (i.e. do we want the triathlon to be more competitive or casual?).

Methods/Algorithms

After completing our exploratory data analysis, justifying our assumptions, and stating our objectives, we need to split up the divisions and, using metrics based on our objectives, determine which splitting of divisions is the "best".

As a baseline, our suggestion is to split the triathlon into two days (see Assumption 4), with the following distribution of divisions for each day:

Day 1	Day 2
M Open (half, 1074) F Open (half, 449) CLY (60)	M Open (half, 1073) F Open (half, 449) M Premier (49) ATH (32) F Premier (18) M Pro (7) F Pro (6)
1583/3217 = 49.2%	1634/3217 = 50.8%

Note, the divisions above are in descending order of size and are not our suggestions to the Mayor. The numbers are also not scaled to the number of participants we predict (2000).

Because we would have too many groups if we let each division start separately, we will be combining the professionals into one division and the premiers into one division. The special division (ATH, CLY) will be their own separate group.

However, we do need to split the open section, as there are too many people in the divisions, even when they are split between the two days. We introduce 4 hyperparameters here: male1_group, female1_group, male2_group, female2_group. These numbers represent how many groups we will split each of the divisions into. For example, if male1_group = 5 and female1_group = 3, then we will split Day 1's M Open division into 5 groups, and Day 1's F Open division into 3 groups. We will discuss optimizing these hyperparameters later in this section.

Calculating Congestion

To calculate congestion on the course, we created a list of groups, which we then iterated through to add their wait times. Then, we append all of the runner's final wait

times into a singular 1-D array, and use that array to calculate the maximum congestion on the course. Below is the pseudocode for creating the singular 1-D array (named *merged* list):

```
given divisions, which contains a list of divisions
shuffle people for each division in divisions
create a list 'groups', which contains of split divisions as detailed
above
initialize empty list 'merged'

wait = 0
for i from 0 to number of groups:
     wait += wait_time # varies depending on the method and
hyperparameters
     groups[i] += wait # add time to each person in the ith group in
'groups'
     append all elements in groups[i] to 'merged'

sort 'merged' by ascending order
round all elements in 'merged' down to the nearest whole number
```

And using that information, here is how we are able to calculate the maximum congestion in O(n) time (note, you can loop through all numbers in the array and count how many times that element appears in the array, but that would take O(n^2) time). The below code is written in Python syntax, but should be pretty easy for a person who is not familiar with Python to follow.

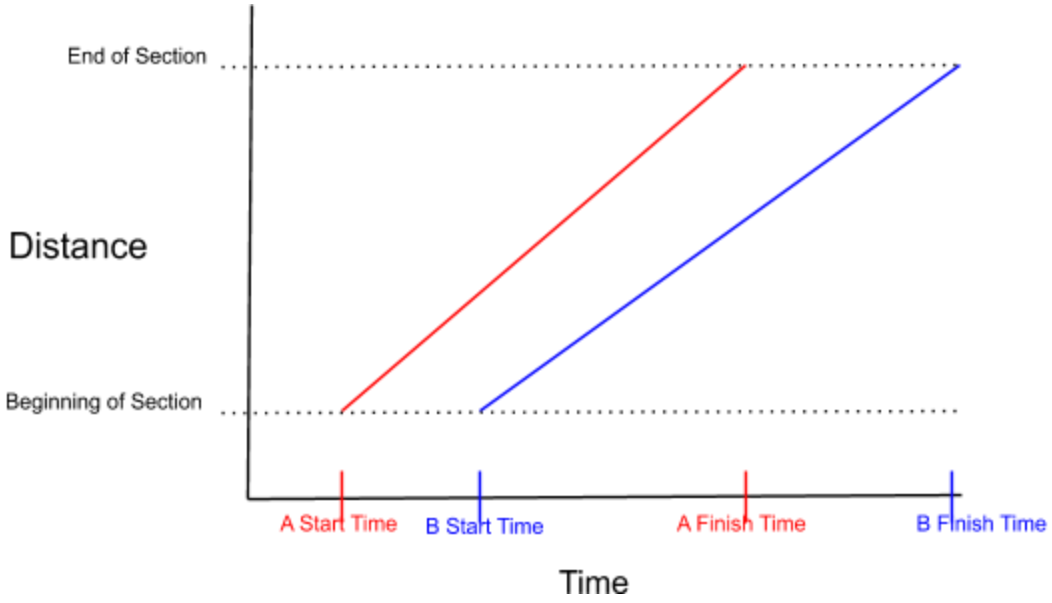
```
overlap_list = []
cnt = 1
curr = merged_list[0]
for num in merged_list:
    if num > curr:
        overlap_list.append(cnt)
        cnt = 1
        curr = num
    else:
        cnt += 1
```

```
max(overlap_list) #returns highest congestion in the course
```

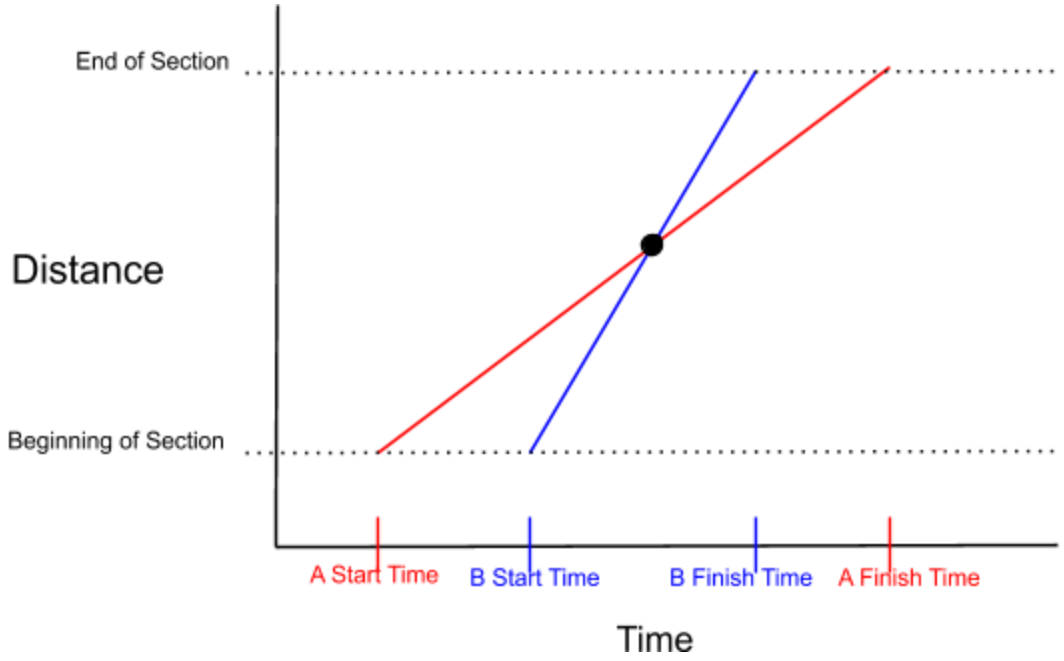
As written in the comments of the code above, the value of wait_time is dependent on the ordering of our groups (to satisfy specific objectives) and the hyperparameters we use. Before we introduce our two plans, however, let's define another important metric that we use for our analysis, collisions.

Collisions

A collision occurs when one competitor passes another competitor in the race. In order to evaluate collisions, we assumed that within each section of the race, a competitor holds a constant pace. Given the times that two contestants start and finish a section of the race, we can figure out whether one passes the other at some point in that section. Specifically, if person A starts that section before person B, but person B finishes that section before person A, person B must have passed person A. If they are running at constant speeds, they only passed each other once.



Person A and Person B do not collide because A starts before B and finishes before B.



Person A and Person B do collide because A starts before B but ends after A.

We implemented the following algorithm for both days in order to calculate how many collisions each person in the race experiences.

```
# end_times[i][j] stores the time after the start of the entire race
at which person i finishes the jth section of the race

for each section:
  for each personA:
    for each personB: # check every possible pair
     if end_times[personA][section-1] < end_times[personB][section-1]
       and end_times[personA][section] > end_times[personB][section]:
         # B passes A
         collisions[personB] += 1
         collisions[personA] += 1
```

Using these two algorithms for calculating congestion and collisions, we would like to present the following two plans to the Mayor.

Plan 1 - Most Competitive

The most competitive plan is best suited to those who are participating in the triathlon professionally. For this plan, we suggest letting the fastest divisions (by average time)

go first, and then allow the slower divisions to go after them. In that way, the most competitive divisions will not be interfered with/collide with slower divisions, and should minimize congestion on the course, at least for the professionals. For the purposes of this demonstration, let's set male1_group = male2_group = 8 and female1_group = female2_group = 4. Here's how the itinerary using this plan would look like:

Itinerary

Groups + Number of people in each group

Day 1	Day 2
M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (136) F Open (112) F Open (112) F Open (112) F Open (113) CLY (60)	Pros (13) Premiers (67) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (135) F Open (112) F Open (112) F Open (112) F Open (113) ATH (32)

Here, we will introduce our other 8 hyperparameters: d1_wait, mf1_wait, cly_wait, d2_wait, pro_premier_wait, premier_m_wait, mf2_wait, ath_wait.

Hyperparameter	Definition
d1_wait, d2_wait	Represents the waiting time between two groups that are the same division (between M Open|M Open and F Open|F Open) on their respective days.
mf1_wait, mf2_wait	Represents the waiting time between the M Open and F Open groups on their respective days.

cly_wait	Represents the waiting time between the F Open and CLY groups on Day 1.
pro_premier_wait	Represents the waiting time between the Pros and Premiers groups on Day 2.
premier_m_wait	Represents the waiting time between the Premiers and M Open groups on Day 2.
ath_wait	Represents the waiting time between the F Open and ATH groups on Day 2.

We hypothesize this plan will lead to the least collisions and lowest congestion in the race.

Plan 2 - Most Fair

The most fair plan is best suited to those who are more casual. Because most of the athletes in our model triathlon were in the Open section, if we model our triathlon in a similar manner, this will most likely be a casual triathlon. For this plan, we suggest that the slowest groups go first, except for the pros and premiers will still go first. Denoting our group-splitting parameters as above, we propose the following itinerary:

Itinerary

Groups + Number of people in each group

Day 1	Day 2
CLY (60) F Open (112) F Open (112) F Open (112) F Open (113) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (136)	Pros (13) Premiers (67) ATH (32) F Open (112) F Open (112) F Open (112) F Open (113) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (134) M Open (135)

In this plan, the slower groups (especially the special divisions), will not have to worry about being the last ones left in the race. As a result, however, there will most likely be more collisions between the athletes as the faster athletes overtake the athletes from slower divisions.

The hyperparameters that we defined in the section above are defined a bit differently for this plan:

Hyperparameter	Definition
d1_wait, d2_wait (same)	Represents the waiting time between two groups that are the same division (between M Open|M Open and F Open|F Open) on their respective days.
mf1_wait, mf2_wait	Represents the waiting time between the F Open and M Open groups on their respective days.
cly_wait	Represents the waiting time between the CLY and F Open groups on Day 1.
pro_premier_wait (same)	Represents the waiting time between the Pros and Premiers groups on Day 2.
premier_m_wait (should be premier_ath_wait but for the sake of simplicity we won't redefine)	Represents the waiting time between the Premiers and ATH groups on Day 2.
ath_wait	Represents the waiting time between the ATH and F Open groups on Day 2.

We hypothesize this plan to minimize the time the local roads need to be closed.

Analysis/Results

Plan 1 - Most Competitive

List of hyperparameters:

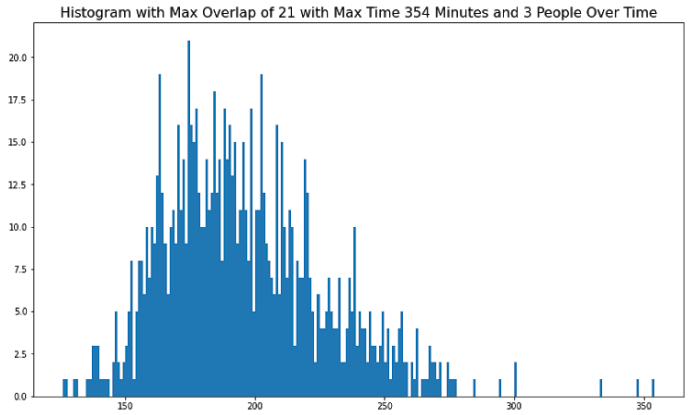
male_group = 8
female_group = 4
d1_wait = 2
mf1_wait = 3
cly_wait = 1
male2_group = 8
female2_group = 4
d2_wait = 2
pro_premier_wait = 1
premier_m_wait = 1
mf2_wait = 3
ath_wait = 1

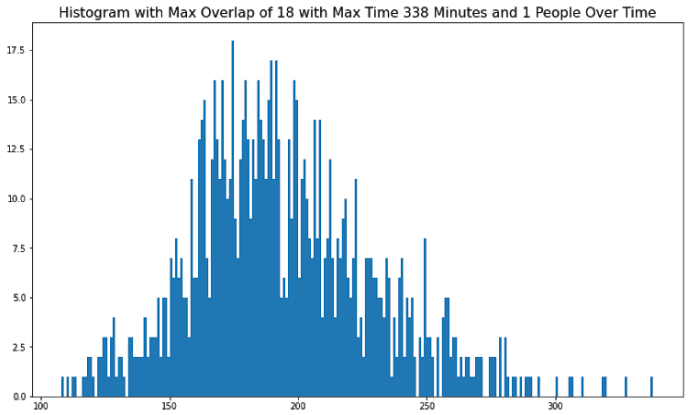
Itinerary

Groups + at what minute *t* the group is released

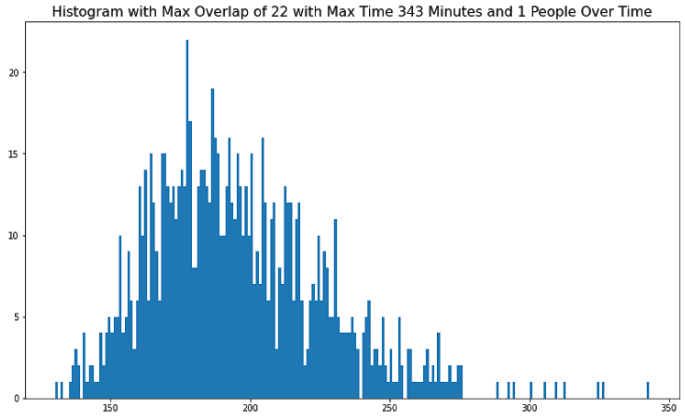
Day 1	Day 2
M Open (0) M Open (2) M Open (4) M Open (6) M Open (8) M Open (10) M Open (12) M Open (14) F Open (17) F Open (19) F Open (21) F Open (23) CLY (24)	Pros (0) Premiers (1) M Open (2) M Open (4) M Open (6) M Open (8) M Open (10) M Open (12) M Open (14) M Open (16) F Open (19) F Open (21) F Open (23) F Open (25) ATH (26)

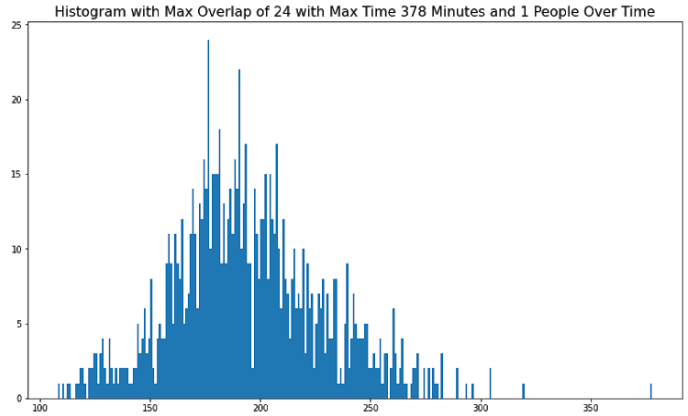
Using these hyperparameters, we get the following distribution of final times (on the left is Day 1, on the right is Day 2):



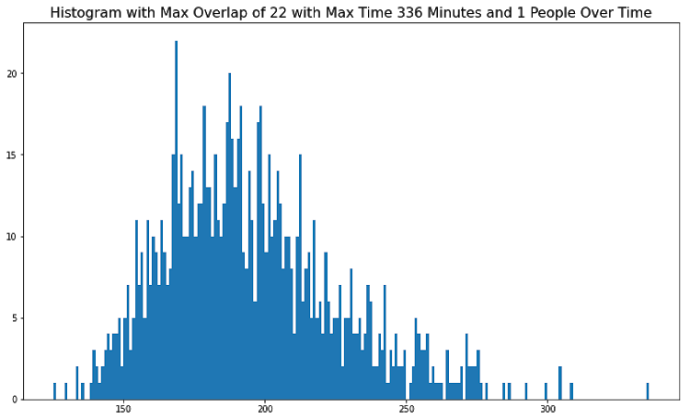


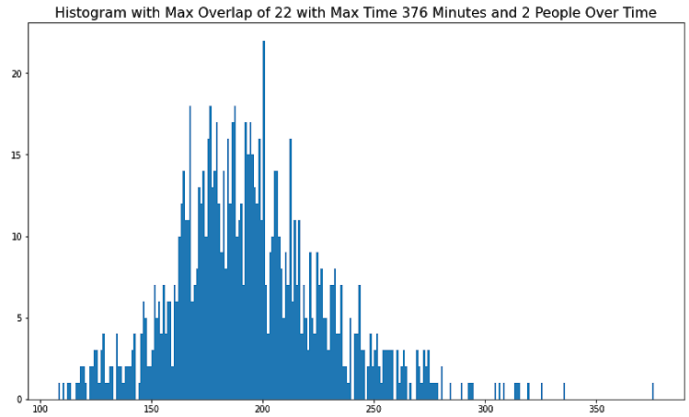
Plan 1 - Final Time 1





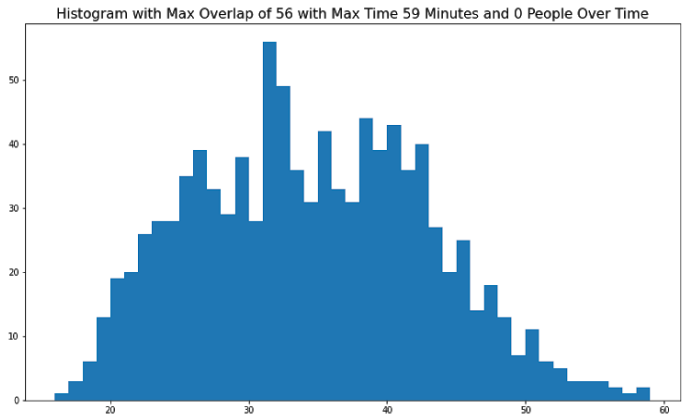
Plan 1 - Final Time 2

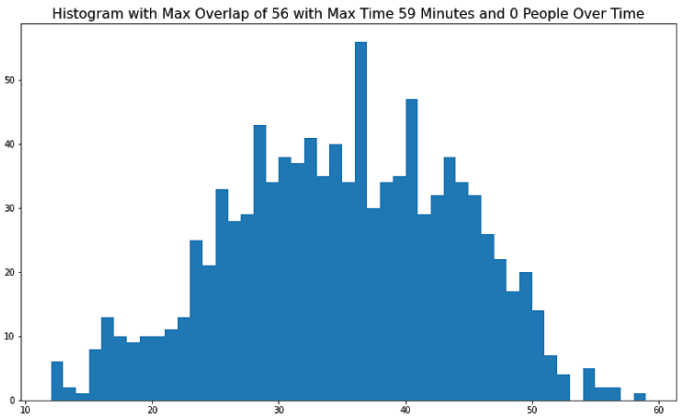




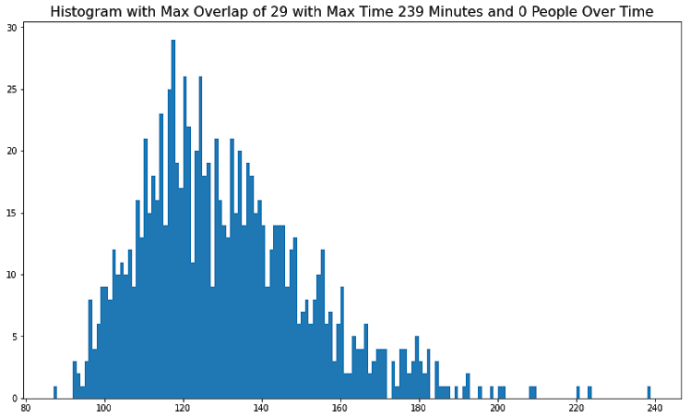
Plan 1 - Final Time 3

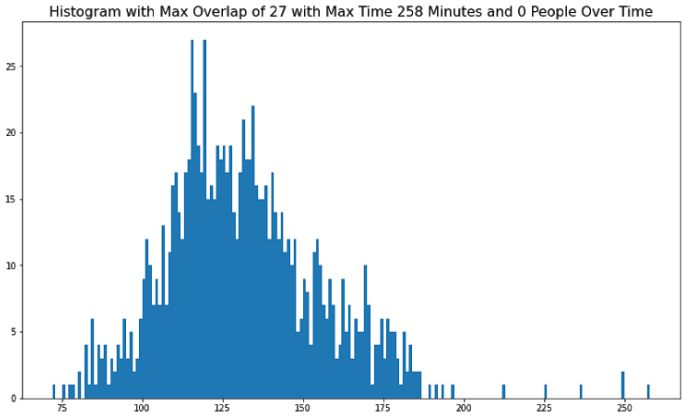
We also want to show the distribution of times after the swimming portion and then after the biking portion.





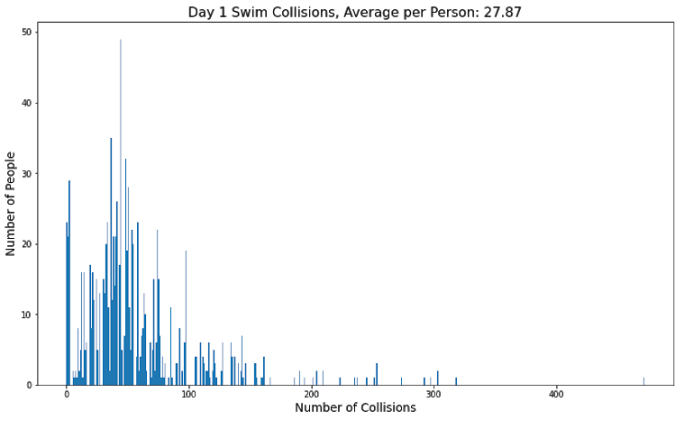
Plan 1 - After Swimming

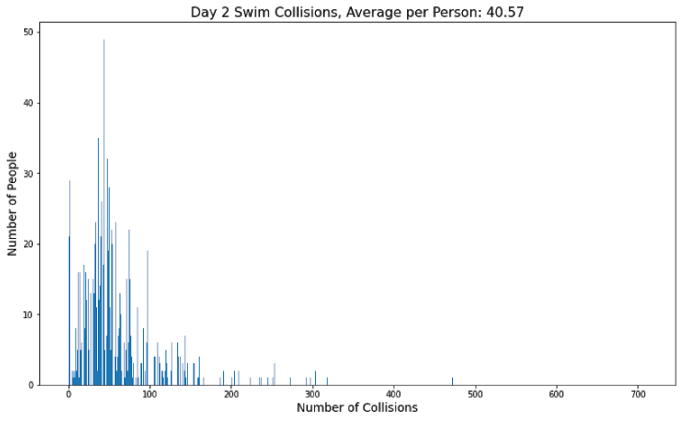




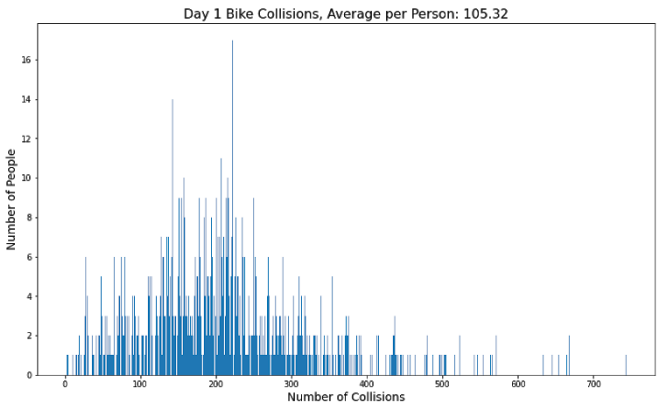
Plan 1 - After Biking

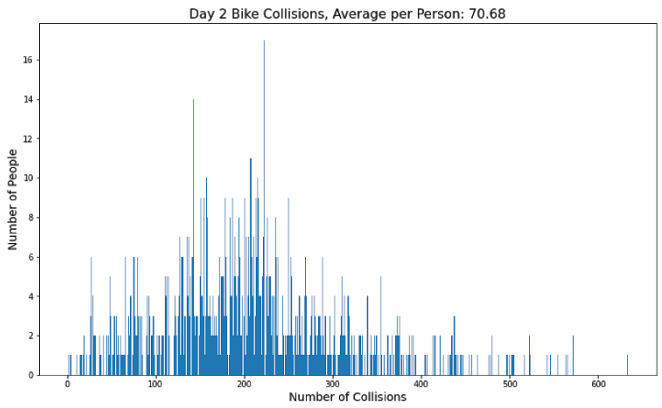
Finally, here is a graph of the number of collisions participants face, after the swim portion, bike portion, and run portion of the race, respectively.



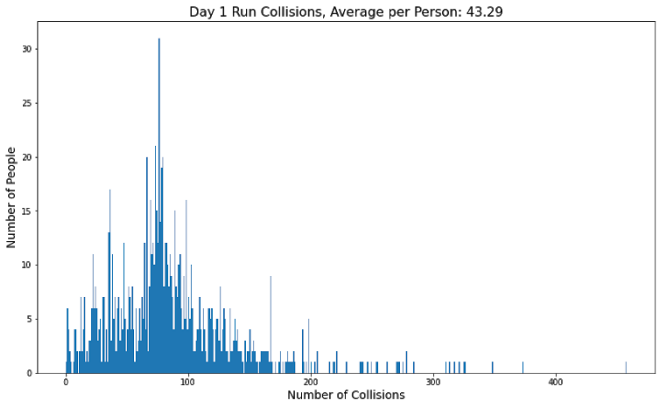


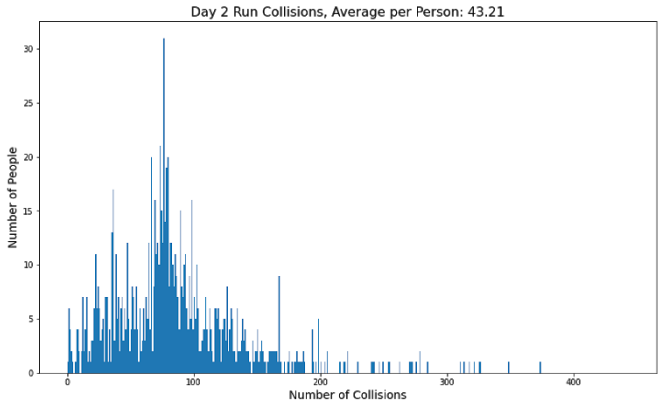
Plan 1 - After Swimming Collisions





Plan 1 - After Biking Collisions





Plan 1 - After Running Collisions

Plan 2 - Most Fair

List of hyperparameters:

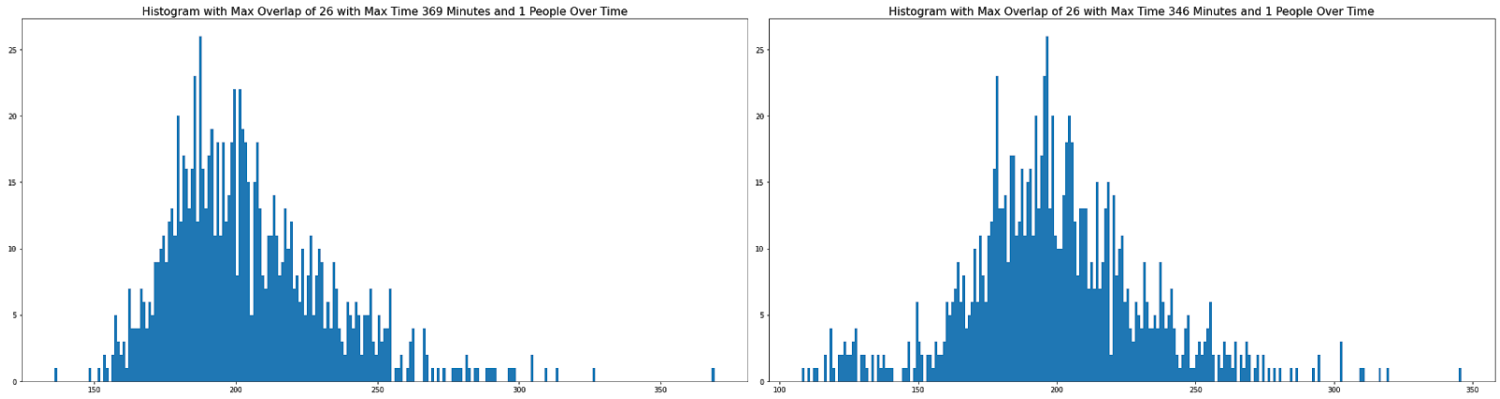
male_group = 8
female_group = 4
d1_wait = 2
mf1_wait = 5
cly_wait = 6
male2_group = 8
female2_group = 4
d2_wait = 2
pro_premier_wait = 0
premier_m_wait = 1
mf2_wait = 5

ath_wait = 6

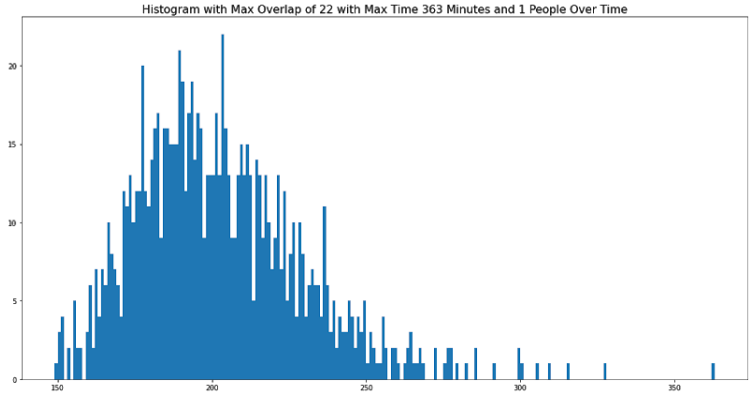
Itinerary

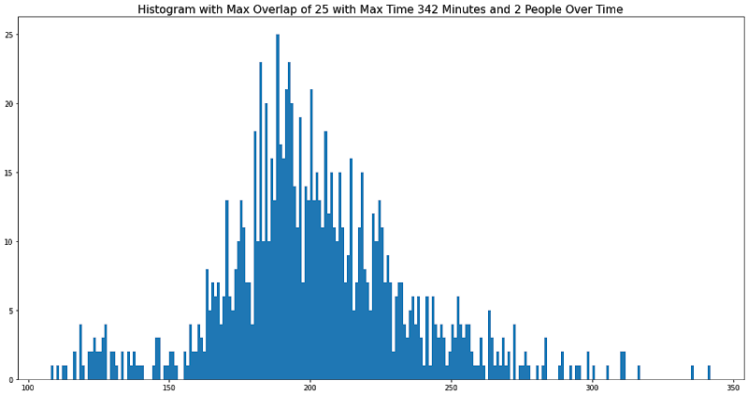
Groups + at what minute *t* the group is released

Day 1	Day 2
CLY (0)	Pros (0)
F Open (6)	Premiers (0)
F Open (8)	ATH (1)
F Open (10)	F Open (7)
F Open (12)	F Open (9)
M Open (19)	F Open (11)
M Open (21)	F Open (13)
M Open (23)	M Open (18)
M Open (25)	M Open (20)
M Open (27)	M Open (22)
M Open (29)	M Open (24)
M Open (31)	M Open (26)
M Open (33)	M Open (28)
	M Open (30)
	M Open (32)

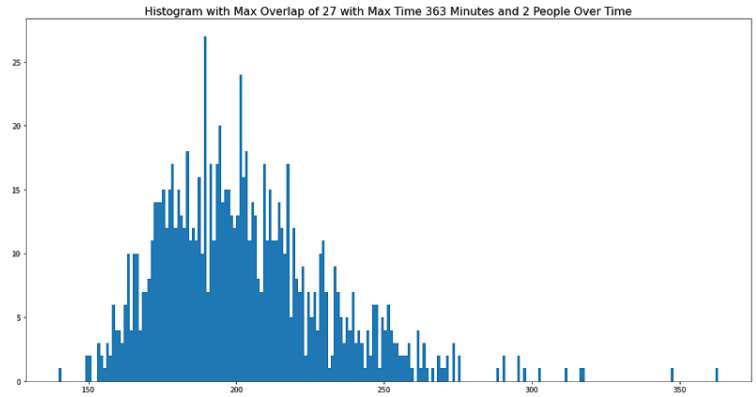


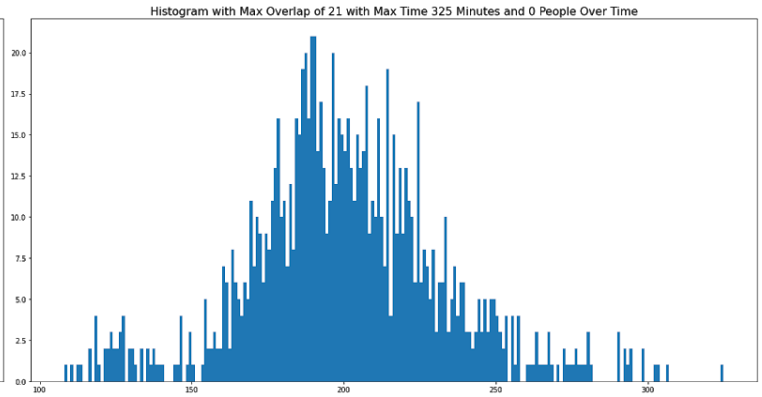
Plan 2 - Final Times 1



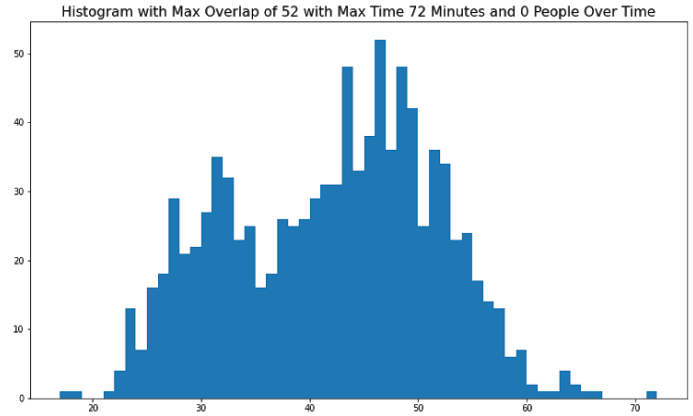


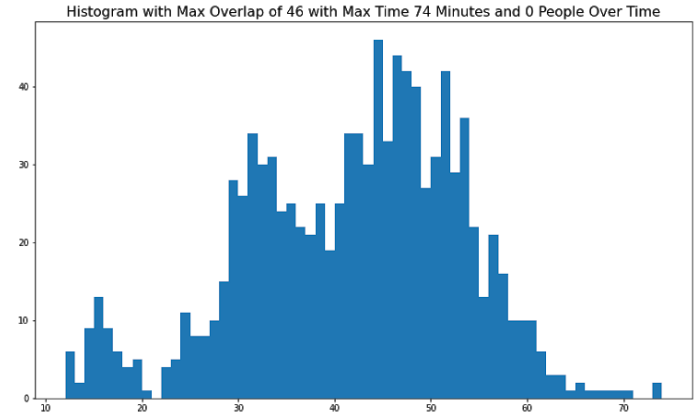
Plan 2 - Final Times 2



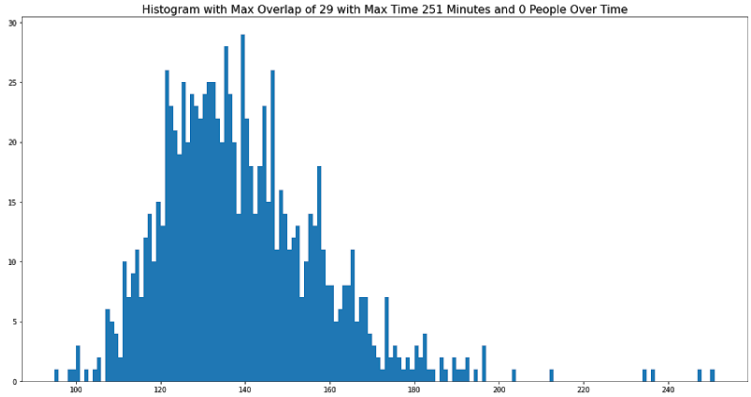


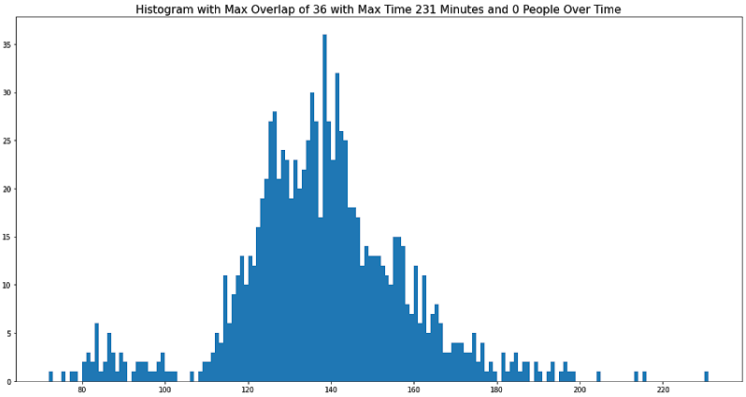
Plan 2 - Final Times 3



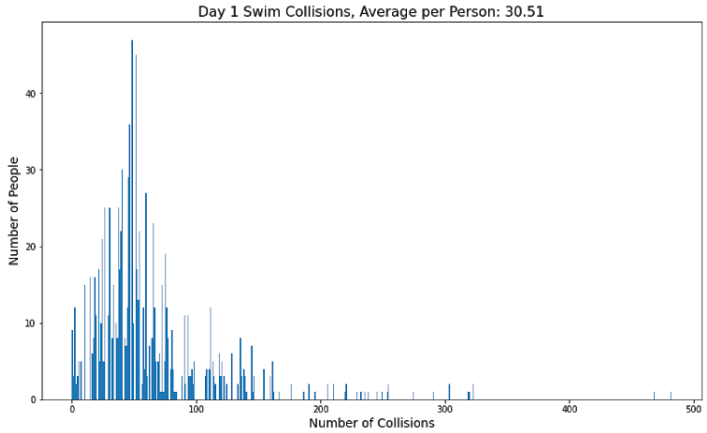


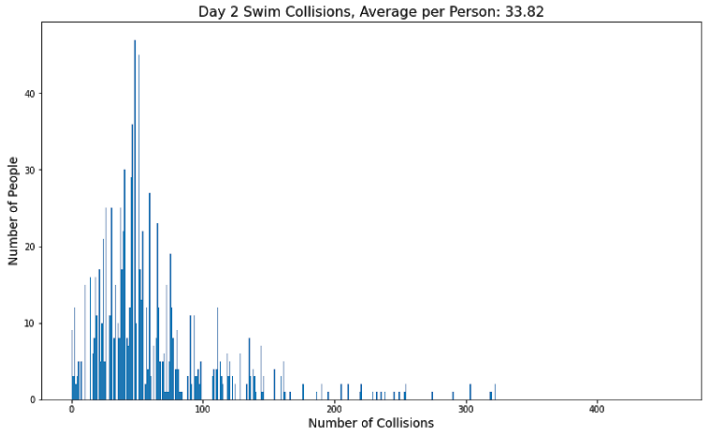
Plan 2 - After Swimming



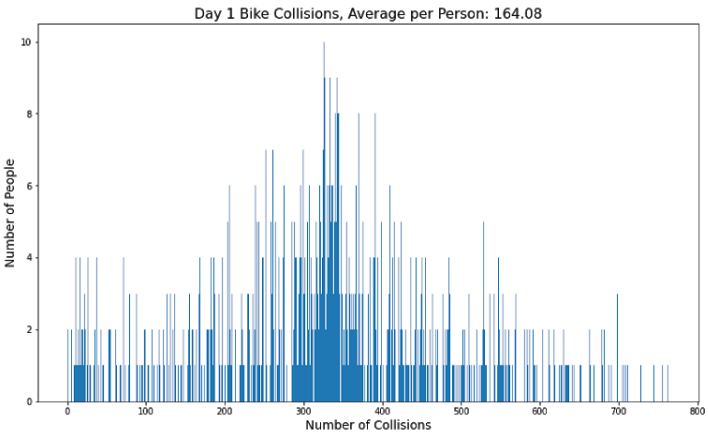


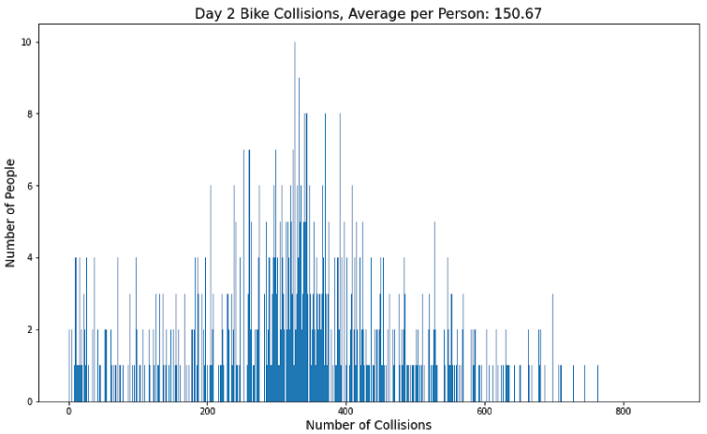
Plan 2 - After Running



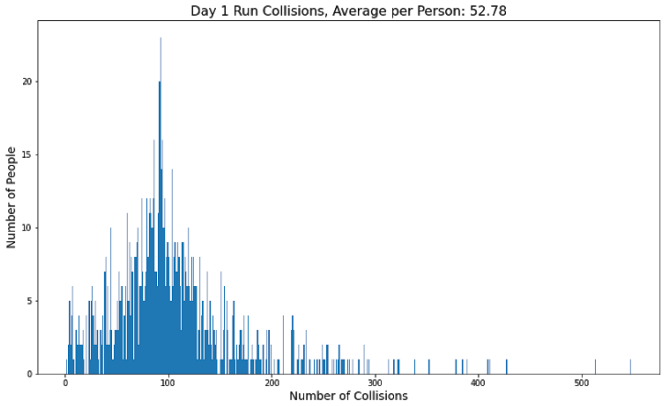


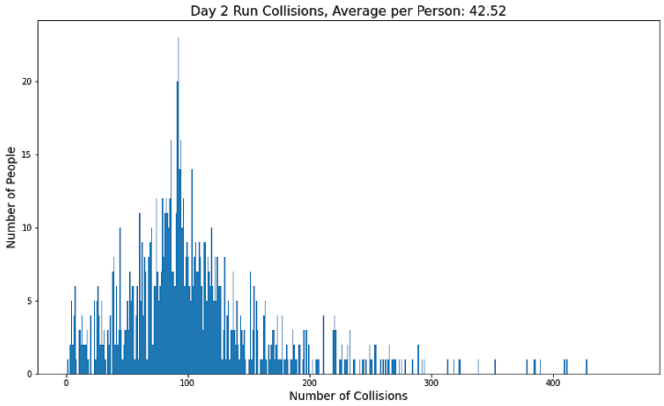
Plan 2 - After Swimming Collisions





Plan 2 - After Biking Collisions





Day 2 - After Running Collisions

Comparison of Plan 1 and Plan 2

We simulated Plan 1 and Plan 2 10000 times to generate the following data.

	Plan 1 (Day 1)	Plan 2 (Day 1)
Max Overlap	21.314 +/- 4.038	23.950 +/- 4.537
Max Time	352.368 +/- 37.838	342.772 +/- 35.982
People that are Over Time	1.793 +/- 2.229	1.013 +/- 1.703

	Plan 1 (Day 2)	Plan 2 (Day 2)
Max Overlap	20.765 +/- 3.954	23.358 +/- 4.427
Max Time	354.858 +/- 37.641	343.808 +/- 35.817
People that are Over Time	1.861 +/- 2.257	1.108 +/- 1.783

Mean of each value plus 95% confidence interval (2 STDs) are included in both tables

Coupled with the collision data, it is safe to assume that there is less overlap and congestion on Day 1, whereas the maximum time the roads have to be closed is minimized on Day 2. The plans seem quite reasonable to fit the Mayor's objectives, and therefore these are the plans that we present to the Mayor for the upcoming triathlon.

Conclusion

Sensitivity

One assumption we made was that competitors have a constant speed during each section. If this did not hold true, the collision algorithm would be affected. For example, if a person stops for a rest, someone may pass them, and then that person would start running again and regain their position in the race. However, in this situation, our collision algorithm would be a lower bound for the number of collisions, since people's positions in space and time are still continuous.

Strengths

The main strength of our model is the flexibility and options available at the Mayor's disposal. We propose two different plans that each accomplish one of the Mayor's main objectives, and leave it up to his/her discretion for choosing which plan best fits his needs. There are also 12 different hyperparameters, each of which can be fine-tuned for further experimentation and optimization. The model is also subject to stochastic variation, and so the model can be run many times to achieve the most accurate results possible.

Weaknesses

The main weakness of this model is the number of simplifications we had to make to generate our algorithm. For example, our collisions algorithm makes the assumption that runners will move at a constant speed, which is not necessarily true. Another potential weakness of our model is that the triathlon may not be able to be held on two separate days, which is something we assumed when creating this model.

Future Work

We were also asked to "explore any advantages you may achieve in terms of congestion and road closure time if you adjust the race distances of the swimming, biking, and/or running events of your triathlon." We opted not to do so in this paper, since we wanted to keep our model limited to traditional triathlons. However, by adjusting our code slightly, scaling the race distances can easily be incorporated, so that is not too large of a concern for future work.

A main point of emphasis for future work would be to allow the collision algorithm to be non-static. Currently, it relies on the indices of two arrays to be consistent; if you were to add the elements of those indices to another list (therefore keeping the pointers together with a nested tuple or list), then you could randomly shuffle the people, just like what our congestion algorithm did. Further details are in the Colab that is linked in the Appendix.

Contributions/Acknowledgements

Evan: Worked on prototyping the congestion and collision algorithm, coded the exploratory data analysis portion and congestion algorithm, devised the two different plans.
Trey: Wrote the collision algorithm, worked on the problem statement and summary, and helped with other algorithms.
Harin: Worked on writing the problem statement and conclusion, proofread the paper for grammatical errors, helped prototype the model.

Thank you to Ms. Belledin for such an amazing semester and experience in mathematical modeling! Each of us is truly grateful for this opportunity which has opened our eyes to the possibilities of using math in the real world. We hope you enjoy the pig at the top of the paper and stay healthy during this pandemic.

Appendix

Google Colab Code
Spreadsheet which we access data from