MODELING TAXI QUEUE AT LA-GUARDIA AIRPORT, NEW YORK

Tomer Eldor

Minerva Schools

CS166 Simulation Final Project

April 2018

DESCRIPTION OF SITUATION

SIMULATION OVERVIEW

New York City's LaGuardia Airport (LGA) is notorious for its mismatch in the demand and supply of taxis, with passengers often waiting over and hour, or taxis waiting of several hours. Via, a ride-hailing app (similar to Uber) headquartered in New York City, has recently started servicing rides to and from LaGuardia airport and wants to increase its quality of service (minimizing wait times and unserved requests) while maximizing profits (minimizing number of cars sent and driver working hours).

This simulation aims to model the potential dynamics of queueing of passengers and taxis in LGA airport and transporting into (and from) Manhattan.[1] It aims to model with some resemblance to reality the following processes:

1. Patterns of passenger exit times from LGA airport and ordering service requests
2. Patterns of drop-offs occurring naturally at LGA airport
3. The company's strategy for forcibly sending cars[2] to wait at LGA queue or away from it to optimize parameters.

SIMULATION RULES

The simulation is an event queue simulation of one hour (or one-time span with the same parameters) for LGA ride hailing app dynamics. The algorithm first generates the times where each event will happen, in minutes, from the start to the end of the simulated time span, and pushes all times into an event queue, or *Schedule*. It samples the event times separately for each event type given its unique distribution, schedules everything into the queue, and only after finishing scheduling all events it starts executing them sequentially.

The distribution of rider arrival times is modeled as a mixture of Gaussians around hypothetical times corresponding to plane arrival times (+X minutes of security and luggage time after drop-off). A similar process occurred in drop-off times: there are specific departure times, and riders usually arrive X minutes (around ~1.5 hours) before scheduled arrival time. However, whereas arrivals are tightly bounded around the peak of

[1] Manhattan was overwhelmingly the most common destination for taxi trips from LGA airport, from the NYC Taxi Trip data.

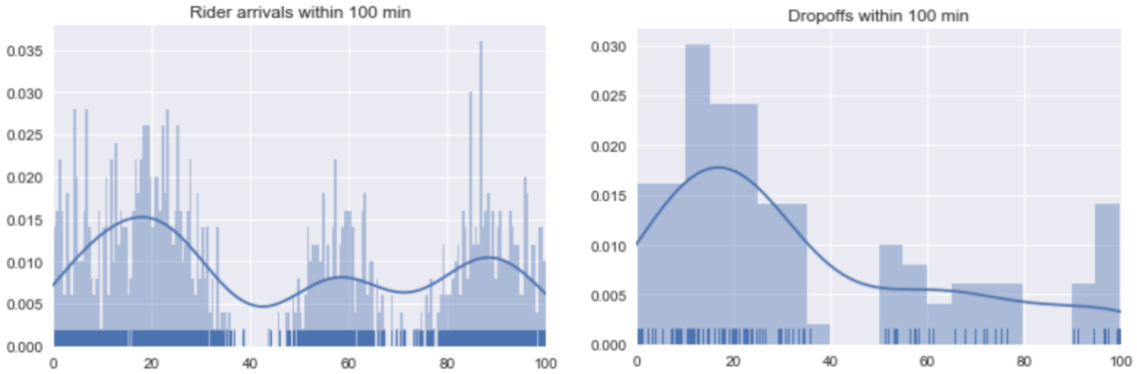
[2] It is worth mentioning that Via employs drivers and pays them hourly, so that they can send assign them to trips without the driver's discretion. The simulation goals are to test parameter values for strategies of proactively sending Taxis to wait at LGA for passengers, and measure its effects on quality of service, costs and profits to the company.

passengers exiting (since they all landed at exactly the same time), passengers drop-off times are much more highly variable because of the variability in preferences (and circumstances) for arrival times before departure. Therefore, the sampling procedure is as follows:

1. According to *F*, a set number of flights per hour (exogenous parameter set per hour), *F peaks* are randomly (uniformly) generated, representing peak time of exit or departure after a particular flight of the F flights.
2. A normal distribution around that F mean is generated.

This process occurs separately (with different F flight times) for arrivals and departures. For arrivals, the variance around the mean is small, while significantly large for departures (drop-offs).

Below is an example of the distributions of rider arrival (ride request) and drop-off times. This was calibrated to resemble observations from LGA airport taxi rides from NYC Taxi data on a Monday morning across 100 Mondays.



The rules are detailed below, for the three types of event chains:

1. **Drop-offs at LGA** – drop-offs are scheduled to leave *mean_trip_time* (30 minutes here) prior to their planned arrival time. At that timepoint, a car is reduced from the counter of available cars at Manhattan, a trip duration set to exactly the *mean_trip_time* of the simulation in order to get the desired end distribution (since the leaving time or trip duration is not of interest on their own here), the taxi is recorded as "on the way" to LGA and then arriving to LGA at the planned time, then joining the FIFO queue of taxis waiting to pick up riders from LGA. If strategy is strategy 1, it is triggered here to check whether to readjust the queue and send a taxi empty back to Manhattan.
2. **Rider Arrives**. According to the scheduled time distribution, a rider exists LGA and requests a ride. The system then:
    a. First, attempts to assign the first taxi in the waiting queue, which would get there at around 1.5 minutes (normal distribution with 0.5 minutes std), and records metrics.
        i. If strategy is Strategy 1, the strategy is triggered here to re-adjust the queue.
    b. Second, if there is no car in the queue, it suggests a car from Manhattan with a suggested (random, normally distributed) trip time to the rider.

i. Each rider has a particular patience threshold (randomly distributed around 15 minutes, with a cap at 20 minutes).
ii. If the suggested ETA is longer than the rider's patience threshold, they give up and leave ("churned".
iii. Otherwise, the ride is assigned, and a car is summoned from Manhattan to pick up that rider.
iv. Metrics of cars are adjusted, and events are pushed into queue. A trip time back to Manhattan is drawn and a trip to Manhattan is scheduled with a final arrival time, where the car will end back at the pool of cars in Manhattan.

3. **Strategy.** Company intentionally sends a car to wait at LGA pickup queue. This will be assessed through two strategies detailed below.

STRATEGIES

Here I test two different strategies for optimizing queue size at LGA, simulating to optimize each of those.

Strategy 1. An optimal number of cars at the queue is calculated by:

$$optimal\_taxis = queue\_min\_buffer \bullet (arrivals - dropoffs) / time$$

Where *queue_min_buffer* is a parameter of ideal queue waiting time on average to be optimized using the simulation. When a car is added or leaves the LGA queue, the system is triggered to check wether the queue size is within an acceptable boundary around the optimal size (set by the "tolerance" parameter – was found to work well set to 10% below or 30% above)

Strategy 2. A constant rate of sending cars to LGA.
This could be also a relateively realistic and easy strategy: by predicted total parameters per hour of approximated numbers of arivals and dropoffs, the company can calculate the gap of needed cars and distribute sending them across the hour (/time-span in question).
This is calculated by:

$$Rate = rate\_coefficient \bullet time / (arrivals - dropoffs)$$

Where *rate_coefficient* is a parameter to be optimized using the simulation.

ASSUMPTIONS

The model has many simplifying assumptions that challenge the

Firstly, assumptions about the distributions should be further refined. The distributions of arrivals and drop-off times need to be further calibrated from actual data *per hour of day, day of week, holiday, weather (cancelling flights) and other external parameters.* Currently these are just sample variables modeled after a distribution observed from

inspecting a collection of Monday mornings. The arrival distribution did exhibit patterns of random peaks looking like a gaussian mixture, but this might change per external parameters; and the drop-offs distribution looked much more random and thick to be able to reason well on a generation process of such distribution. I aim to refine these according to further inspection of data later on.

The normal distribution of trip times is probably not correct – it is not actually normal; but in reality, it would be highly dependent on many variables that are not so much of our interest or control, and we preferred not focusing on that at the moment and reducing variance due to external and unchanging factors such as trip time. The normal distribution of riders' patience is also questionable but less of primary interest.

Quantities are modeled at a lower scale after analyzing NYC Yellow Taxi data (publicly available online).

The quantities are scaled down because: first. as a simulation we care more about the relations between the variables for this preliminary analysis, and in the future can expand the simulation's granularity and size; and second, because only a fraction of riders from LGA will actually have the Via app and want to order from there. Via mention that they would not receive more than 500-1000 ride requests per hour at most from LGA airport, and that they can allocate at most 1000 vehicles there.

Via is primarily a ride-sharing company and their algorithm automatically tries to match sharing-suitable rides. However, they usually do not have the mass of ride requests yet to match many rides together from LGA yet, and thus they requested to firstly focus on modeling a single-rider taxi simulation.

PARAMETERS

The adjustable simulation parameters are:

- n_rider_arrivals (int): Total number of rider appearances/requests from LGA per timespan (default: hour). This will change per hour modeled because of patterns from real data. The more arrivals we'd have, the more taxis we'd need to send.
- n_dropoffs (int): Total number of random drop-offs at LGA per timespan (default: hour). This will change per hour modeled because of patterns from real data. The more drop-offs we have, the fewer taxis we'd need to send.
- end_time (int): total simulation time. default: hour. In the future, this entire model should run per hour, many times, and sequentially with varying parameters per hour.
- total_taxis (int): total number of taxis in our fleet available for reallocation to LGA. This is a cap on the number of available cars to send which we need as a constraint.
- strategy (int): choose strategy order number to use: 1 or 2.
- queue_min_buffer: (int/float) for Strategy 1: approximate desired average minutes for taxi to wait in queue,
- to be used in calculating ideal number of taxis in queue

- tolerance_proportion: (float) for Strategy 1: percent deviation from optimal LGa taxi queue size to tolerate before adjusting in Strategy 1.
- rate_coefficient: (float) for Strategy 2: coefficient to multiply the postulated rate of cars to send, to be optimzed
- trip_price: (int) 30$ per hour in this simulated hour. This changes as Via has lower rates for off-peak hours.
- show: (bool) whether or not to print results

OUTCOMES OF INTEREST

To quantify the "Quality of Service" and "Associated Costs and Benefits", I measure 12 variables (and their variations), focusing on minimizing the following outcome variables:

1. Riders waiting times.[3]
2. Riders left ("churned") – riders who denied service because waiting time was too long (to minimize).
3. Driver waiting times at the LGA airport queue.
4. Driver "idle driving" times – if we forcibly send empty cars from Manhattan to LGA or vice versa, we pay for that time, so we need to measure it as a loss.

I both inspect some distribution of these parameters, but also *aggregate them into a unified* ***cost function***. The cost function is a weighted function of:

1. The costs of paying drivers in idle queueing times and idle driving times, which are discounted by half because driver minutes are worth $0.5 since we pay them 30$ per 60 min.
2. The "cost" for making riders wait for a car (sum of riders waiting times), where rider waiting minutes are modeled as worth $1 because they would pay 30$ for 30min average trip time (equivalent to 1$ per minute of opportunity cost)

So, the main outcome of interest is the score cost function for minimization, modeled as:

score_cost *= 0.5 • (driving_idle_times + driver_queueing_times) + riders_wait_times + (trip_price*churned_riders)*

In addition to that, I also inspect the total profits and sometimes the individual values of each of those parameters to see where the variation came from.

[3] Rider waiting times will be captured primarily by their mean, but also present and inspect the shape of the distribution. The worst-case scenario is always 20 minutes, so it is not of interest, since we cap the waiting time at 20 minutes (above which we reject service request) and thus know the worst case is always 20 minutes; but Median might also be of interest."

IMPLEMENTATION

Code and results available on Github in a iPython notebook.

After briefly inspecting NYC taxi data for a rough idea for an example ratio of drop-offs and arrivals on a week-day morning, the ratio seemed to be around 10:2 pickups to drop-offs (around 1000:100). However, riders are *much more likely*, at least twice, to take Via (as a ride-sharing service) as *from* the airport rather than *to* the airport as they are rushing in. Therefore, I am using a sample of 1000:100.

This simulation report is an initial demo of the model's performance without yet plugging in all the true parameter values for each hour per week. Those parameters would be derived from NYC taxi data, Uber data, flight schedule data and proprietary company (Via) data later. For now, as an estimation, we simulated repeatedly a simulation with 1000 arrivals, 100 drop-offs, for 100 minutes over the various parameter values. On the right is an excerpt example from a log of a simulation run.

LGA Taxi Simulation Log Example

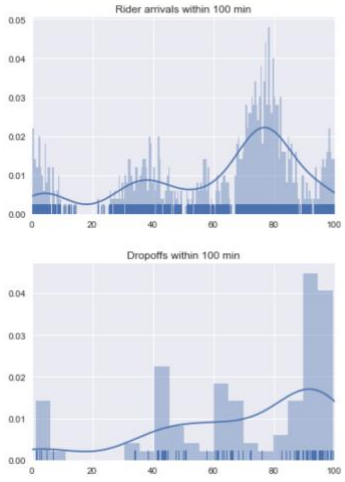
```
02.129: executing: rider_arrives, with args: []
02.129: RIDER 7 ARRIVES and requests ride; patience 11.5233405379
02.129: SENT CAR TO LGA, arriving at  10.913
02.129: SENDING CAR TO LGA: queue had 1 taxis and 17 on the way, while optimal is 20.0
02.129: LGA QUEUE TAXI waited for 12.874 min in LGA picks up rider 7 in 00.750 min
02.129: scheduled pickup for rider 7 at 02.880 min
02.129: trip will end for rider 7 after 51.5577452723 at 54.438 min

02.155: executing: rider_arrives, with args: []
02.155: RIDER 8 ARRIVES and requests ride; patience 12.0270127921
02.155: SENT CAR TO LGA, arriving at  34.585
02.155: SENDING CAR TO LGA: queue had 0 taxis and 18 on the way, while optimal is 20.0
02.155: LGA QUEUE TAXI waited for 06.685 min in LGA picks up rider 8 in 00.701 min
02.155: scheduled pickup for rider 8 at 02.856 min
02.155: trip will end for rider 8 after 37.8172195718 at 40.673 min

02.184: executing: rider_arrives, with args: []
02.184: RIDER 9 ARRIVES and requests ride; patience 7.10789191527
02.184: rider 9 LEFT. waiting 35.38 min exceeded patience 7.11 min

02.554: executing: rider_arrives, with args: []
02.554: RIDER 10 ARRIVES and requests ride; patience 16.9267564362
02.554: rider 10 LEFT. waiting 28.43 min exceeded patience 16.93 min
```

RESULTS ANALYSIS

STRATEGY 1: DYNAMIC QUEUE RE-ADJUSTMENT

Below are sample results of a simulation with strategy 1.

```
----------SIMULATING STRATEGY 1 WITH BUFFER 4 MIN------------
```



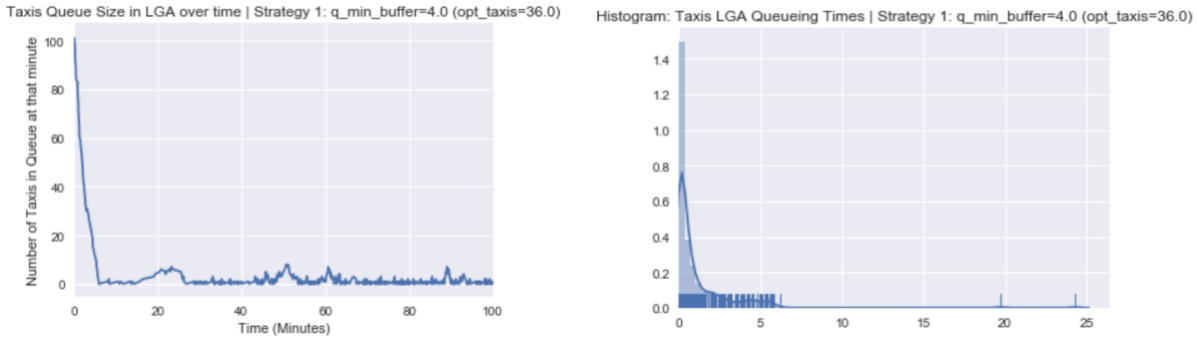
The rider arrivals and drop-offs within 100 minutes are distributed as a gaussian mixture according to our assumptions and generation function.

The optimal value for the buffer coefficient was around 10, which created a mean driver waiting time of ~4.5 minutes. However, for a buffer coefficient value as low as 4, the results weren't too far from optimal.

When the buffer coefficient was smaller than 4, there were too many cars in queue, so the queue size in LGA dropped sharply at first and stayed around zero throughout. At around 4, more fluctuations and spikes were seen in the steady state after the initial drop close to zero.

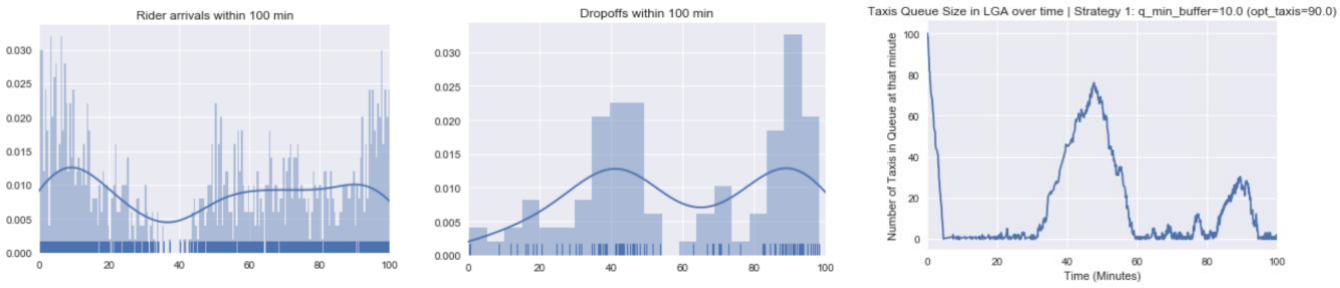
At such small buffer coefficient values, the taxi waiting times extended all the way to 10 minutes or beyond of waiting, while the riders waiting times did not change much. This is because even if a car is at the queue and is called, it takes about 1-2 minutes to arrive to its client anyway.

For a buffer coefficient of 4 (which meant projected optimal queue size of 36 taxis) created desirable distributions.

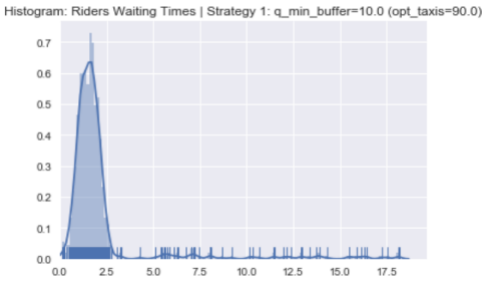


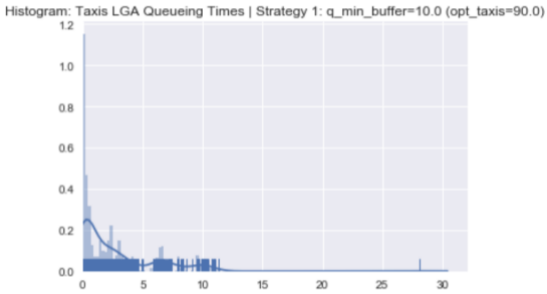
The rider waiting times were also very good, normally distributed around 1 minute, with a long and continuously fluctuating (with randomness) tail until the 20 minutes cutoff. At that rate, also taxis queueing times were very small, were the large majority waited 0 minutes, and the rest in an exponential distribution with a mean around 1 minute, extending until ~5 minutes and barely any taxis waited more than 6 minutes.

However, these meant we are sending many taxis, so resulting costs were high. Were they worth it? According to the profits and cost function, no. An optimal cost minimizing value was found around a buffer of 10. The waiting times are not *as* optimistic, but they are more cost effective in the longer term. Below is an example of taxis queues size through time, seeing larger peaks of a queue of up 70 taxis, accumulating around minute 40 when more drop-offs occurred and there was a gap in rider arrivals.



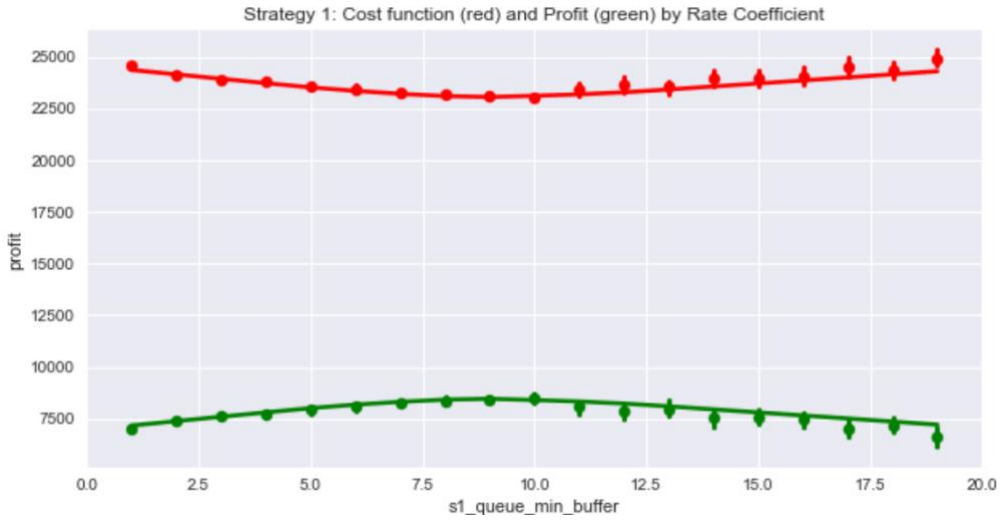
With that, riders still waited mostly just 1.5 minutes (with a long tail until 18 minutes with few samples), and Taxis mostly still watied only 0-1 minutes in queue, the longest time (except for 1 outlier) being around 10 minutes wait time.

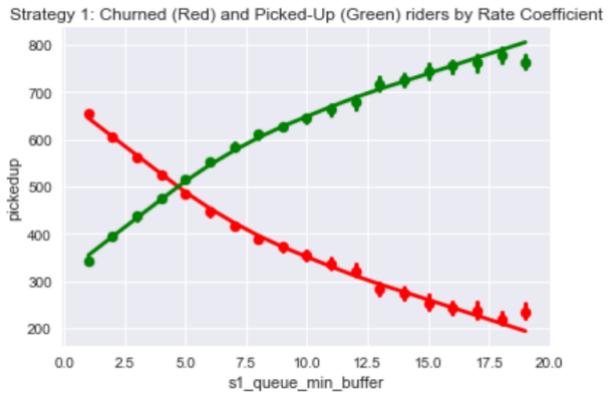


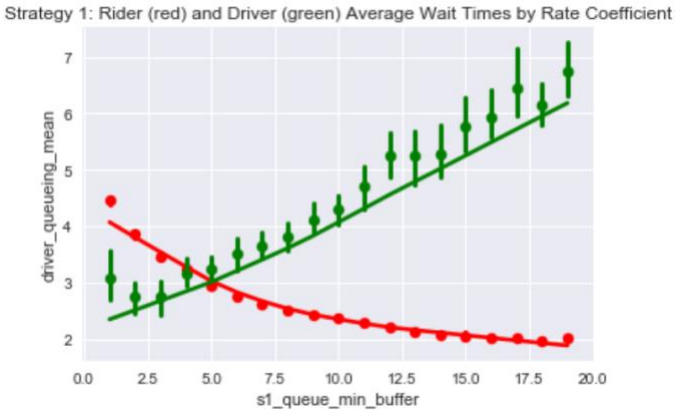


By plotting the sampled profits and cost functions per parameter value, we can see that the cost function was minimized, and profits were maximized then at a buffer value of 10.

We started picking up more riders than churning them at 5 minutes and at 10 minutes we reached about a double rate (so losing 30% of requests, 70% utilization rate). The average waiting time for riders was slightly over 2 minutes.







We see that while the picked-up riders only surpassed the number of churned riders at minute 5, and so did driver wait times vs passenger wait times, the gap increased in the favor of the rider's service quality continuously.

With strategy 1, therefore, the ideal parameter value was 10, meaning 90 taxis at the optimal queue for a simulation with 1000 arrivals and 100 drop-offs in 100 minutes.

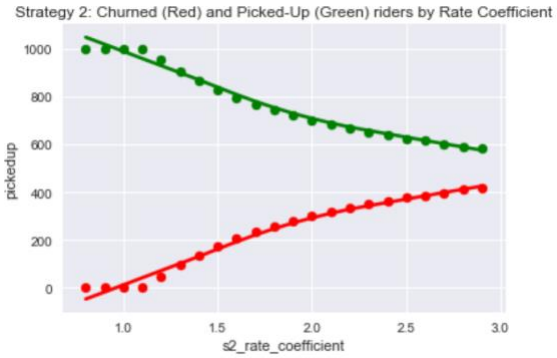
STRATEGY 2: CONSTANT CAR SENDING RATE

In strategy 2, the optimal value was a rate coefficient of 1.3, as it visibly minimized the cost function and maximized profits to **$275,000** (compared to a maximum of $8500 with strategy 1).

The distribution of waiting times was similar to before.

The waiting times increased as the rate coefficient increased.

```
number of observations: 2940
```



CONCLUSION: ADVICE

Strategy 2 maximized profits strikingly more than strategy 1 (from $8500 to $275,00). Its optimal sending rate coefficient was found to be 1.3. Therefore – from the strategies tested, the optimal strategy Via should adopt is sending cars consistently every 8.4 seconds, when hour with 600 ride arrivals and 60 drop-offs, which was equivalent to a Monday morning.

I suggest the following next stages in work:

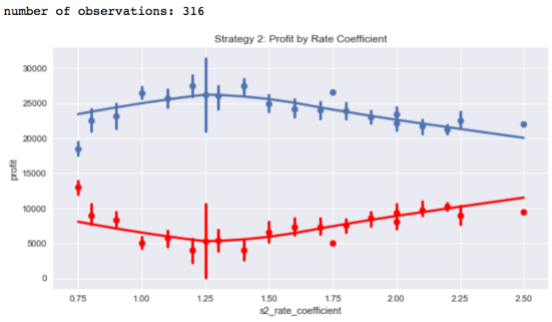
First, I would inspect more closely the data patterns of number of arrivals and drop-offs for *each hour of each day across a lengthier period of time*, and then simulate for these exact parameters, finding the optimal strategy for each combination.

Second, I would test more strategies. Specifically, if have live flight schedule data, we might be able to predict better peaks of arrivals, and thus send cars at a non-constant rate but adjust the sending rate to match the specific predicted distribution pattern.

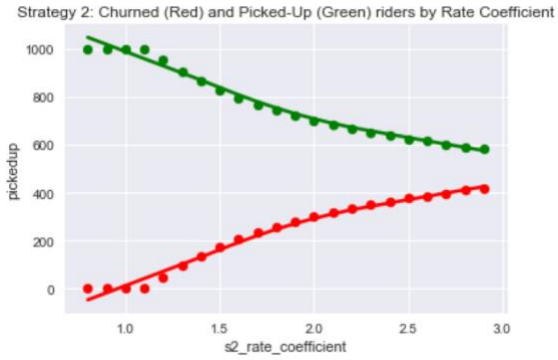
UNCERTAINTY OF RESULTS

The uncertainty increased as to a desirable value as I increased the number of simulation runs from ~20 per parameter combination set to ~200 per parameter combination set.

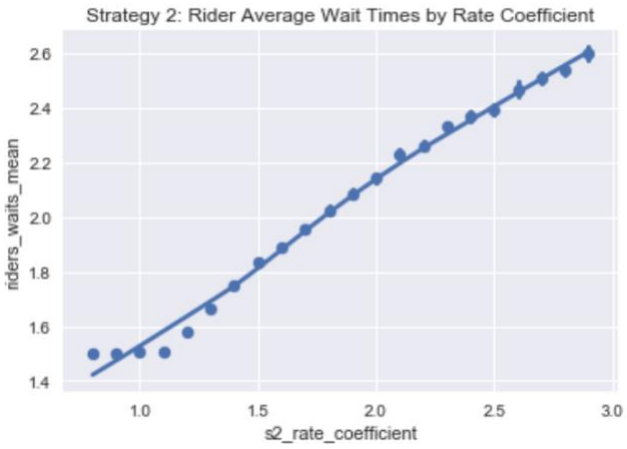
The confidence in our predicted profits changed greatly with sample size. For a sample of only 2-3 cases as in a rate coefficient value of 1.25, the variation was extreme, at $10,000 in predicted profits. In a relatively small sample of incidences, the variability was very large, mostly between $2000 – $4000 in predicted profits for a sample of ~300. See figure below, were profits are at blue on top, and the red points and line are of the cost function which was symmetrical to the profits.



```
number of observations: 2940
```



In a large sample of about 3000, the confidence decreased significantly to between $100 – $1000. In churned, picked-up riders, and rider wait times, the confidence intervals are almost invisible and are at sufficiently small quantities for out interest.



CONCLUSION

While this simulation is highly simplified, it revealed useful points of interest:

- A constant rate of sending cars is superior to keeping a steadier queue size
- For a simulation with 1000 arrivals and 100 drop-offs, strategy 1 yielded a best queue size of 90 cars maximizing profits to $8500, but *strategy 2* was by far better, and showed to be optimal with a rate coefficient of 1.3.
- More work is needed in order to make this more relevant for Via (/the company in question) and based on real data.