Case 2: Out-of-Hospital Cardiac Arrest Volunteers in NYC

Chenyang Yan cy477

Hongyi Nie hn327

A. Generate a figure with the number of volunteers on the x axis and the expected survival rates from OHCA on the y axis.

To use de Maio survival function, we started by finding pi and lambda i.

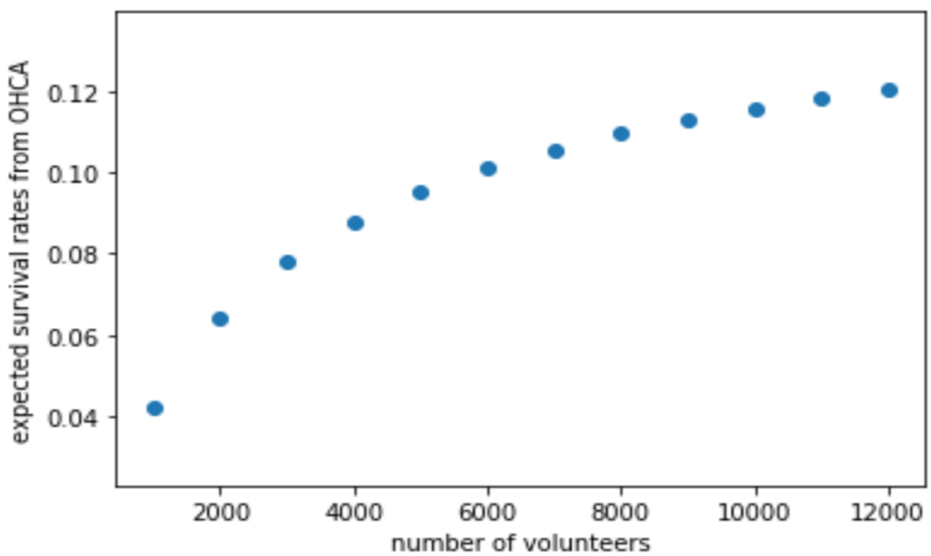
Firstly, know that the probability that it came from Location i is proportional to the population in Location i, we have 'p of coming from i' = ‘pop’/’sum(pop)’, and pi = 'pop density' = p(fall in region i) = ‘pop’/’area(m^2)’, and we want to rescale it so it become the percentage population density and sum up to 1. Then probability of population density = 'pop density' /sum(‘pop density’)

Next, since we assumed that in each location the volunteer density is proportional to the population density in that location, we can defined lambda(# of volunteers) by the formula lambda i = density of volunteers in region i (vols/ m^2) = volunteer * 0.3 * ‘'percentage of pop density' /'area (m^2). We timed 0.3 because we knew that any given volunteer is available and willing to accept an alert with probability 0.3.

Now we have both pi and lambda i, in order to get pdf f(t), we need the derivative of cumulative function P(T<t). From the class we were given the function P(T> t), so all we need to do is derivative 1- p(T>t) = $\frac{d(1-\sum_i pi\, e^{-\pi s^2(t-3)^2\lambda i})}{dt} = -\sum_i pi\, e^{-\pi s^2(t-3)^2\lambda i} *$ $(-\pi s^2(t-3)^2\lambda i)$, we achieved it by initiating the answer value to 0 and using a for loop to fulfil the sum purpose.

Next, we assume the speed = 60m/min, and use de maio function: g(t) = $(1 + e^{0.679 + 0.262t})^{-1}$, we initiated the survival list empty, and we wanted to append the integral of it to the survival list, the integral range is from 3 to 3 + x. The 3 is came from a 3-minute delay from patient collapse to the time at which volunteers receive an alert, and x is the time we defined by ourselves, we assume survival rate increase 0.01, so g(t) = 0.01 and calculate t value for $(1 + e^{0.679 + 0.262t})^{-1}$ = 0.01, so we have x = 14.947. Therefore, we tried every 1000i for i between 1000 to 12000, append the $\int_{3}^{3 + 14.947} (1 + e^{0.679 + 0.262t})^{-1} * f(t, lambda(i))$to the survival list.

The scatter plot is shown below:



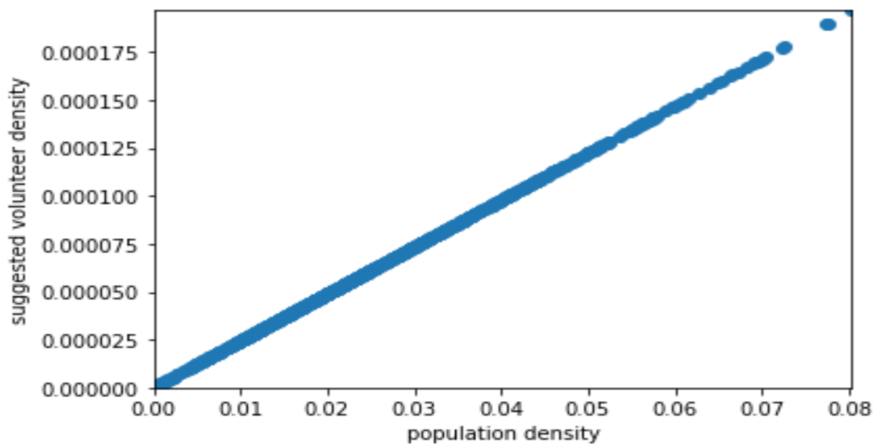
We observed that more volunteers will result in higher expected survival rates from OHCA, but the ratio decreases as the number of volunteers increases. For 5000 volunteers, the survival rate is 0.099 within 3 to 17.947min, and in order to compare the survival rate with additional 7000 volunteers for next question, we also compute the survival rate of 12000 volunteers, which is 0.1216 within 3 to 17.947min.

B. Decide the way those volunteers are located across the set of locations in NYC for 5000 + 7000 volunteers.

Then we focused on the problem, suppose that currently we have 5,000 volunteers in NYC and we could recruit an additional 7,000 volunteers. Previously we defined lambda i = density of volunteers in region i (vols/ m^2) = volunteer * 0.3 * 'percentage of pop density' /'area (m^2), since we assumed that in each location the volunteer density is proportional to the population density in that location. However, now we only consider about the probability of coming from area i, so we defined a new lambda, new_lambda i = probability of coming from region i (vols/ m^2) = volunteer * 0.3 * 'p of coming from i' /'area (m^2). Therefore, for i = 12000, $\int_{3}^{3+14.947} (1 + e^{0.679 + 0.262t})^{-1} * f(t, lambda(12000))$ = 0.1248. Thus, the survival rate is 0.1248 within 3 to 17.947min for 12000 volunteers.

Compare Part A to Part B, the survival rate increased from 0.1216 to 0.1248 for 12000 volunteers, so we concluded that it is better to assign volunteers in the sense of the intensity of the associated Poisson point process instead of assigning them by proportional to the population density.

The scatterplot where each point corresponds to a single location, with population density on the x axis and suggested volunteer density on the y axis is shown below:



From the part B answer and scatter plot, we know that the higher population density will require higher volunteer density, and highest population density with highest volunteer density would yield the biggest increase in survival rates, and assign volunteers in the sense of the intensity of the associated Poisson point process instead of assigning them by proportional to the population density will result a higher survival rate.

C. Extend the model to take that into account in a tractable way for 3D models.

We think about 2 ways to extend our model to 3D.

1. We split the NYC areas to 3 levels: High building (Average building floor level >= 20), Normal building(10<=Average building floor level<20), Low building (Average building floor level< 10). We only use 3 mins delay for low building, and for normal and high building, the delay time changes to 7mins and 12mins accordingly. Then our model become a piecewise function:

$$1- p(T>t) = \frac{d(1-\sum_i pi\, e^{-\pi s^2 (t-3)^2 \lambda i})}{dt} \text{ for low building area}$$

$$\frac{d(1-\sum_i pi\, e^{-\pi s^2 (t-7)^2 \lambda i})}{dt} \text{ for normal building area}$$

$$\frac{d(1-\sum_{i} pi\, e^{-\pi s^{2}(t-12)^{2}\lambda i})}{dt}$$ for high building area

Then do the rest of work accordingly for each building level area. This method will be potentially hard in mathematics computation since it require conjunction function to union three categories, so we have a second method.

2. This method is less expensive to compute than the first method, we still split the NYC areas to 3 levels: High building, Normal building and Low building, then we simply times a ratio to the population density to assume we have more people in high building area and less people in low building area, so it can automatically change the require time and volunteer number accordingly without change the whole model setting. For example, if there are 1000 people in a high building area, we multiply it by 1.5 so we assume the resources we need for this area are for 1500 people instead of 1000; for 1000 people in normal building area we multiply it by 1.2 so we assume the resources we need for this area are for 1200 people instead of 1000; and for 1000 people in normal building area we do not change it. This method is easy to compute since it does not change the model functions, and theoretically make sense.