

## Problem statement

One interesting challenge at Tesla is determining how many cars have been in proximity to points of interest, such as superchargers. This can be abstracted to the following question: Given a set of  $N$  cars, we want to compute how many cars are within  $R$  meters of a supercharger. For simplicity, we assume that all coordinates are specified as an  $(X, Y)$  pair, and the distance metric is Euclidean.

## Sample data

- *coordinates.csv*: contains 1 million car coordinates as  $(X, Y)$  pairs with a header, units in meters
- *centroids.csv*: contains 1,000 supercharger coordinates as  $(X, Y)$  pairs with a header, units in meters

## Goal

Write code to calculate the answers to the following questions, and include the numeric answers in your response.

## Questions

1. How many cars are within 5 meters of at least one of the superchargers?
2. How many cars are within 10 meters of at least one of the superchargers?
3. What is the minimum radius  $R$  such that 80% of cars are within  $R$  meters of at least one of the superchargers?
4. Bonus: what is the maximum radius  $R$  such that no single supercharger has more than 1,000 cars within a radius of  $R$  meters? Also, what is the maximum radius  $R$  such that there are no more than 1,000 cars that have a supercharger within a radius of  $R$  meters?
5. Mega-bonus: based on this dataset alone, where should we build our next 5 superchargers, and why? Just a description of a proposed algorithm is fine.

## What we are looking for

- **Code quality**: The implementation should be simple, documented, and allow a developer to use your code and efficiently compute answers to similar questions with it. The code itself should be intuitive and simple to maintain. (Python is a good choice of language, but not required.)
- **Efficiency**: Your implementation should be fast. Any commonly available libraries can be leveraged to this effect within reason.
- **Correct answers**: Please provide the numeric answers your code generates to the above questions.

## Deliverables

- Your source code.
- The numeric answers to Questions 1-4 (but it is OK to be more open ended for Question 5).
- Documentation of the computation and memory complexity of your implementation as a function of  $K$  (number of superchargers) and  $N$  (number of cars).