

Context

Imagine you're a shopowner and you have a number of parking spaces (e.g. 200) available in front of your store for shoppers & employees. Now, because you see more and more EVs parking there every day, you're planning on building some chargepoints.

But first, you're trying to figure out how much power these chargers actually need, and how much energy they'll consume.

Why is this relevant

If you build 20 charging stations with a maximum charging speed of 11kW each, the theoretical maximum of total power demand is 220kW (a very high number that would be expensive to satisfy, e.g. requiring a new, more powerful grid connection). You know this is only a theoretical figure, because its statistically unlikely for all 20 charging stations to be charging at full power at the same time.

By simulating how electric chargers are actually used we can simulate how high the total energy consumption (kWh) is, what peak power loads (kW) occur, and how these figures behave change with the number of chargepoints installed.

That's your task for today!

Tasks

Task 1: Logic

Tech: Feel free to use any programming language you like for this task.

Write a program to simulate 20 chargepoints with 11kW power for one year in 15-minute intervals (non-leap-year, 365 days / 35040 ticks).

The probability distribution of an EV arriving at a chargepoint at a given time is displayed in **T1**. A chargepoint can only charge one EV at a time, meaning it is blocked from receiving another EV until the prior one has left again. For simplicity, we assume that EVs depart as soon as they're done charging.

The probability distribution of an arriving EV's charging needs is displayed in **T2**. We assume that all EVs are standard vehicles and need 18kWh per 100kms.

When you've done this, calculate:

- Total energy consumed in kWh

- The theoretical maximum power demand
- The actual maximum power demand (= the maximum sum of all chargepoints power demands at a given 15-minute interval)
- The ratio of actual to maximum power demand ("concurrency factor")

Hints

- *The theoretical maximum demand is the result of the number of charging stations multiplied by the charging power*
- *The actual maximum demand should be around 77-121 kW*
- *The concurrency factor should be between 35-55%*

Bonus

- Run the program from task 1 for between 1 and 30 chargepoints. How does the concurrency factor behave?
- If you consider the impact of DST vs. mapping the hours to the 15 minute ticks.
- If you seed the probabilities vs. using random() for random-but-deterministic results.

Task 2a: Frontend

Tech: Feel free to use the Frontend stack of your choice. If you know Typescript, React, and Tailwind that's great, but everything else is fine too. You do **not** have to connect the code from Task 1 to the Frontend, just create some nice-to-use, good-looking and (slightly) functional mockups.

You are tasked with visualizing the input parameters and the output.

The input parameters could be:

- the number of charge points
- a multiplier for the arrival probability to increase the amount of cars arriving to charge (20-200%, default: 100%)
- the consumption of the cars (default: 18 kWh)
- the charging power per chargepoint (default: 11 kW)

For the output, you could visualize:

- The charging values (in kW) per chargepoint at a useful aggregation level
- An exemplary day
- The total energy charged (in kWh)
- The number of charging events per year/month/week/day

Bonus

- Create a UI to allow creating different types of chargepoints (e.g. 5 x 11kW, 3 x 22kW, 1 x 50kW).
- The deviation of the concurrency factor from the bonus task could be displayed (if the previous bonus task was completed).

Task 2b: Backend

Tech: Feel free to use the Database and Api tech of your choice. If you know Typescript, Postgres, Prisma, Node.js, Express, and GraphQL that's great, but everything else is fine too. You do **not** have to connect the code from Task 1 to the Backend (!).

You are tasked with persisting the input parameters and output values. First, create endpoints to Create, Read, Update, and Delete a set of inputs. Then, create an endpoint that runs a (mock) simulation (doesn't have to be connected!) and stores the (mock) results in the database.

The input parameters could be:

- the number of charge points
- a multiplier for the arrival probability to increase the amount of cars arriving to charge (20-200%, default: 100%)
- the consumption of the cars (default: 18 kWh)
- the charging power per chargepoint (default: 11 kW)

For the output, you could store:

- The charging values (in kW) per chargepoint at a useful aggregation level
- An exemplary day
- The total energy charged (in kWh)
- The number of charging events per year/month/week/day

Data

T1: Arrival probabilities	
Time of the day	Probability of arrival
00:00 - 01:00	0.94 %
01:00 - 02:00	0.94 %
02:00 - 03:00	0.94 %

T1: Arrival probabilities	
03:00 - 04:00	0.94 %
04:00 - 05:00	0.94 %
05:00 - 06:00	0.94 %
06:00 - 07:00	0.94 %
07:00 - 08:00	0.94 %
08:00 - 09:00	2.83 %
09:00 - 10:00	2.83 %
10:00 - 11:00	5.66 %
11:00 - 12:00	5.66 %
12:00 - 13:00	5.66 %
13:00 - 14:00	7.55 %
14:00 - 15:00	7.55 %
15:00 - 16:00	7.55 %
16:00 - 17:00	10.38 %
17:00 - 18:00	10.38 %
18:00 - 19:00	10.38 %
19:00 - 20:00	4.72 %
20:00 - 21:00	4.72 %
21:00 - 22:00	4.72 %
22:00 - 23:00	0.94 %
23:00 - 24:00	0.94 %

T1: Charging demand probabilities	
Probability	Charging demand in kilometers of range
34.31 %	None (doesn't charge)
4.90 %	5.0 km

T1: Charging demand probabilities	
9.80 %	10.0 km
11.76 %	20.0 km
8.82 %	30.0 km
11.76 %	50.0 km
10.78 %	100.0 km
4.90 %	200.0 km
2.94 %	300.0 km