Gorilla Data Engineer Assessment

1. Use **pandas** to calculate a transportation distribution charge for four gas meters in the United Kingdom. Save your code in a **Jupyter notebook** and upload to a public repo on **Github** (or any other platform of your choice). While solving this exercise, focus on efficiency - i.e., **use vectorised operations and avoid loops**! All data needed for the calculation can be found in the Excel file for this exercise.

Transportation distribution charges are levied by gas distribution companies for the use of their lower pressure pipelines; they cover the cost of physically transporting the gas through the pipeline. This rate is determined depending on a meter's exit zone (gas network region) and its estimated annual quantity (AQ); and it changes over time.

The daily charge is calculated by finding the correct rate for each meter and day in the forecast and multiplying this rate (in p/kWh) with the day's forecast (in kWh).

Calculate the **total cost per meter** by summing its daily charges for the full forecast period and converting to Pounds (1p = 0.01£).

Calculate the **total consumption per meter** by summing its daily consumption forecast for the full period.

Your result should be a DataFrame of the following form with all numerical values rounded to 2 decimals:

Meter ID	Total Estimated Consumption (kWh)	Total Cost (£)
14676236		
...		
...		

Example:

*Looking at meter **14676236** with exit zone **EA1** and an AQ of* **28978 kWh**

We can find the correct subset of rates in the rate table by selecting the correct exit zone and annual quantity band according to the meter properties. The AQ band is hereby determined by assuring the AQ is between the minimum AQ (*aq_min_kwh*, included) and the maximum AQ (*aq_max_kwh*, excluded, may be open-ended). For this meter, the following rates are found:

Rates determined for meter *14676236 :*

date	exit_zone	aq_min_kwh	aq_max_kwh	rate_p_per_kwh
2020-04-01	EA1	0	73200	0.2652
2020-10-01	EA1	0	73200	0.2970
2021-04-01	EA1	0	73200	0.3327
2021-10-01	EA1	0	73200	0.3726
2022-04-01	EA1	0	73200	0.4173
2022-10-01	EA1	0	73200	0.4674
2023-04-01	EA1	0	73200	0.5235
2023-10-01	EA1	0	73200	0.5863
2024-04-01	EA1	0	73200	0.6566

The rate from 2020-04-01 to 2020-09-30 is 0.2652 p/kWh
The rate from 2020-10-01 to 2021-03-31 is 0.2970 p/kWh
...
The rate from 2024-04-01 onwards is 0.6566 p/kWh

Calculate the cost per day for each meter by multiplying the forecast for that day (kWh) with the rate for that day (p/kWh) to obtain a cost in p.

Costs calculated for meter *14676236 :*

On 2020-06-01:
Cost: *0.2652 p/kWh* * 22.070768 kWh = 5.85317 p
On 2022-02-27 :
Cost : 0.3726 p/kWh * 39.466673 kWh = 14.70528 p
etc

---

2. Write a function that generates a list of random meters of any size. Examples of valid exit zones can be found in the rate table. You may randomly generate the annual quantity.

3. Write a function that generates mock consumption data given a list of meters and a start date and duration (number of days in the forecast). The data may be completely random and it doesn't have to match with the meters' annual quantities either.

4. Write a function that takes as an input a meter list and a consumption forecast table and that calculates the transportation cost table (i.e., best take your logic from task 1 and wrap it in a function). Benchmark this function using meter lists of different sizes and consumption forecasts for periods of different lengths. How does the function scale for larger sets of data?

5. What are your observations after benchmarking? Are there any steps in the cost calculation that can be improved? How would you go about improving the performance of this calculation?