Problem statement:

First, document any **data anomalies, assumptions, or issues** that you face during the analysis and what approach you took to address these. These include duplicates, missing values, incorrect values etc. Once the challenge is completed include the code/doc/pdf that you created for this analysis and delete the dataset. We are also interested in seeing your **additional analysis/insights**.

Dataset:

1. telemetry_data.csv has streaming data with following columns
    a. Id - identifier for the telemetry message
    b. vehicle id – identifier for vehicle
    c. timestamp - time at which we receive data from the vehicle
    d. speed - speed at which the vehicle was travelling at given timestamp
    e. Ignition status - indicates if the vehicle is on or off
    f. EV battery level - percentage of battery available
    g. Odometer - value of odometer at timestamp
2. triggers_soc.csv has additional data for generating charging events
    a. CTS - equivalent to timestamp as seen in telemetry data
    b. Pnid - partner notification ID
    c. Name - name of the event for a given timestamp
    d. Val - value of the parameter for a given timestamp
3. vehicle_pnid_mapping.csv mapping file for PNID and vehicle ID
    a. id - vehicle ID available in telemetry data
    b. ids - list of PNIDs associated with each vehicle
4. artificial_ign_off_data.json few additional “ignition off” messages to consider
    a. vehicleId – identifier for vehicle
    b. timestamp - additional ignition off event to be added at timestamp

Expected tasks to be completed:

1. Create a table containing ignition events for each vehicle using the following logic:
    a. when we observe state change in ignition status column in “*telemetry_data.csv*” as shown below:

Previous state	Current state	Event
On	Off	ignitionOff
Off	On	ignitionOn

Then we consider this as an ignition event for that vehicle

b. when we have ignition messages in “*triggers_soc.csv*” (name = IGN_CYL), we can create an ignition event based on the value of the IGN_CYCL

val	Event
ON	ignitionOn
OFF	ignitionOff

c. generate ignitionOff events at all the timestamps mentioned in “*artificial_ign_off_data.json*"

Timestamp	Event
2021-09-23 12:23:54	ignitionOff
2022-01-03 17:45:51	ignitionOff

2. Create a table containing EV trigger events (Active, Abort, completed)
    a. available in "*triggers_soc.csv*" (name = EV_CHARGE_STATE)

3. gather all the **trigger events** together into one table
    a. combine the tables from points 1 and 2 (ignition events and EV trigger events) into one table
    b. Each row should consist of the vehicle_id, event name, event timestamp.

4. Create the **final dataset** by adding charge/EV battery level for above generated **trigger events** (ignition events and EV trigger events) to
    a. The ev battery level or vehicles can be obtained from two files
        i. "*telemetry_data.csv*" – EV_BATTERY_LEVEL column
        ii. "*triggers_soc.csv*" – name = 'CHARGE_STATE', val column
    b. For each **trigger event** get the ev battery level at that event's timestamp from either of the 2 tables mentioned in (a). If there is no battery data for the exact timestamp of the event, then use the battery data with timestamp closest to the event's timestamp. Note that this "closest timestamp" used should be within 300 seconds on either side (+300 or -300 seconds) of the event's timestamp. If there is no battery data in either file that is within 300 seconds of the event timestamp, then don't fill in the charge value.
    c. Your final dataset should contain vehicle_id, event name, event timestamp, charge level

5. after generating the **final dataset**, start generating charging events using the following logic
    a. generate a charging event when we observe increase in battery charge value is greater than
        i. 1% if the ignition state is off
        ii. 3% if ignition state is on

The charging events table generated should have each row representing one charging event with the following structure:

Vehicle_id	charging_event_start	charging_event_end	Ev_battery_level_start	Ev_battery_level_end	Battery_change
Abcd-ef	02-09-2023 03:01	02-09-2023 03:15	34.9	37.6	2.7
Ghij-kl	02-09-2023 15:23	02-09-2023 16:15	29	83.5	54.5

If you complete the assignment in advance, then reach out to us for an additional EDA task.