**Prereqs**

* Ensure data/processed/metadata/live\_dataset.json exists and includes:
  + provenance.raw\_root
  + provenance.subsets
  + provenance.raw\_files (for your default case: ["1. BatteryAgingARC-FY08Q4/B0005.mat"])
* Activate the same Python env you’ve been using.

**0) (Optional) Build a manifest (audit of raw files)**

& C:/Users/User/anaconda3/envs/EvFastCharge/python.exe -m src.nasa\_data\_extract.build\_manifest

Writes an inventory CSV you can ignore for the main flow, but handy for audits.

**1) Charge extract → features (single or multi-file)**

**Option A — your default single file (B0005)**

& C:/Users/User/anaconda3/envs/EvFastCharge/python.exe -m src.nasa\_data\_extract.run\_nasa\_pipeline --file "1. BatteryAgingARC-FY08Q4/B0005.mat"

This produces a processed “charge+features” CSV under data/processed/.../ (the script already picks the output path/name).

**Option B — whole subset (or many files)**

& C:/Users/User/anaconda3/envs/EvFastCharge/python.exe -m src.nasa\_data\_extract.run\_nasa\_pipeline --subset "1. BatteryAgingARC-FY08Q4" --lim 999

Tip: --file accepts multiple paths; --subset accepts multiple subset names.

**2) Materialize tables (Tbl1/Tbl2/Tbl3)**

Point --csv at the processed CSV created in step 1 and choose an output folder (I’ll use data/processed/nasa\_fy08q4\_b0005 as an example).

& C:/Users/User/anaconda3/envs/EvFastCharge/python.exe -m src.nasa\_data\_extract.make\_tables --csv "data/processed/clean\_nasa\_charge\_SINGLEFILE.csv" --outdir "data/processed" --soc\_target 0.8

Expected outputs (in --outdir):

* Tbl1\_signals.csv
* Tbl2\_episodes.csv
* Tbl3\_metadata.csv

**3) Build SoH (discharge) aligned to charge and merge into tables**

This is the key fix: join\_soh reads your live\_dataset.json and uses **the same files** as charge.

& C:/Users/User/anaconda3/envs/EvFastCharge/python.exe -m src.nasa\_data\_extract.join\_soh --live-json "data/processed/metadata/live\_dataset.json"

Expected outputs:

* TblSoH\_discharge\_capacity.csv
* Tbl1\_signals\_with\_SoH.csv
* Tbl2\_episodes\_with\_SoH.csv

You’ll also see console logs showing the **exact discharge files selected** and discovered **battery set** (should be B0005 for your default run).

**4) (Optional) EDA on the final tables**

You can run EDA on either the plain tables or the SoH-merged ones. Example with SoH:

& C:/Users/User/anaconda3/envs/EvFastCharge/python.exe -m src.eda.run\_eda --csv "data/processed/Tbl1\_signals\_with\_SoH.csv" --soh "data/processed/TblSoH\_discharge\_capacity.csv" --outdir "data/processed/figures"

Expected outputs:

* PNG plots in data/processed/figures/
* eda\_summary.json in the same folder

**5) (Recommended) Update the live metadata (quick)**

After steps 2–4, open data/processed/metadata/live\_dataset.json and fill/refresh:

* stats.row\_counts.tbl1 = number of rows in Tbl1\_signals.csv
* stats.row\_counts.tbl2 = number of rows in Tbl2\_episodes.csv
* stats.row\_counts.soh\_table = rows in TblSoH\_discharge\_capacity.csv
* stats.entities.batteries\_signals = unique battery\_id in Tbl1
* stats.entities.batteries\_soh = unique battery\_id in SoH (should match)
* stats.entities.cycles.charge / discharge = unique (battery\_id, cycle\_id) counts
* (Optional) paste the min/max ranges from eda\_summary.json into stats.ranges