Changes to EBT:

# New code to estimate data for missing sectors/fuels:

### Firstly make sure to use this function at the bottom of utility functions which will delete the oldfiles from modelled\_data:

# Example usage:

# base\_path = "../../data/modelled\_data"

# file\_templates = [

#     "{ECONOMY\_ID}\_biofuels\_{SCENARIO}\_{YYYY\_MM\_DD}.csv",

#     "{ECONOMY\_ID}\_biomass\_others\_supply\_{SCENARIO}\_{YYYY\_MM\_DD}.csv",

#     "{ECONOMY\_ID}\_non\_specified\_{SCENARIO}\_{YYYY\_MM\_DD}.csv",

#     "{ECONOMY\_ID}\_other\_own\_use\_{SCENARIO}\_{YYYY\_MM\_DD}.csv",

#     "{ECONOMY\_ID}\_other\_transformation\_{SCENARIO}\_{YYYY\_MM\_DD}.csv",

#     "{ECONOMY\_ID}\_pipeline\_transport\_{SCENARIO}\_{YYYY\_MM\_DD}.csv"

# ]

# for economy in ALL\_ECONOMY\_IDS:

#     move\_files\_by\_templates(economy, file\_templates, base\_path)

# # move\_files\_by\_templates(economy\_id, file\_templates, base\_path)

### Then be aware of the new code:

## Modeling code:

                minor\_fuel\_supply\_modelling.minor\_fuels\_supply\_and\_transformation\_handler(SINGLE\_ECONOMY\_ID, first\_merge\_df, PLOT = True, CREATE\_MARS\_EXAMPLE=False)

* This one replaces the biofuels one but still does the same things, just allowing for more fuels. Be aware of the local vars stated at the top of the file.

#create newly modelled data using that first merge: estimate\_missing\_sectors\_using\_activity\_estimates(first\_merge\_df,SINGLE\_ECONOMY\_ID, MERGE\_ONTO\_INPUT\_DATA=False,SAVE\_OUTPUT\_TO\_MODELLED\_DATA\_FOLDER=True)

* This one is used for sectors we don’t want to spend tieme on modelling. It also uses locally defined vars, especially acitvity\_to\_missing\_sectors\_dict and activity\_to\_proxies\_dict
* The modelling will use total energy from proxies for activity stated in activity\_to\_proxies\_dict to model the sectors in acitvity\_to\_missing\_sectors\_dict. So e.g. the activity gas\_works is proxied by total demand of 08\_03\_gas\_works\_gas for which the trend is estimated for 'sub2sectors': ['09\_06\_01\_gas\_works\_plants', '10\_01\_02\_gas\_works\_plants'] based on their ratio of use to the proxy activity in the base year.
* Can add more sectors as we need.
* Note how this and all other new code works AFTER supply results become available and MAJOR\_SUPPLY\_DATA\_AVAILABLE is set to True. This allows for modellign to be done using those results.

### Checking and slight adjustments:

*Note the use of utils.ERRORS\_DAIJOUBU in these. You can turn it on to run things and just have files with all the errors saved to data/temp/error\_checking folder, or you can have it so it’ll throw errors and force you to fix them. For now, while we are prioritizing getting modelling out I think we keep utils.ERRORS\_DAIJOUBU false and change it once the modellers are less busy. Almost all errors seem to be in transfoamtion/supply so they are okay to save for later.*

final\_results\_df = adjust\_projected\_supply\_to\_balance\_demand(second\_merge\_df,SINGLE\_ECONOMY\_ID)

* This one is for making sure that total supply matches demand, which is even more useful after estimate\_missing\_sectors\_using\_activity\_estimates(). However I foiuind that many fuels weren’t balancing anyway. It will assign extra/less demand to the supply source (imports/exports/production) that is largest/most suitable. If the majority of supply comes from transfoamtion output then this will record the difference as a statistical discrepancy AFTER getting an A.O.K from the user in the form of writing the sector into SPECIFIED\_ALLOWED\_STATISTICAL\_DISCREPANCIES

incorporate\_capacity\_data()

I inserted refining results into incorporate\_capacity\_data and the vis system. Most importantly, make sure that the file data/processed/refining\_capacity\_all\_economies\_thousand\_barrels\_p\_day.xlsx is updated with latest refining capacity semi-regularly.

double\_check\_difference\_one\_year\_after\_base\_year (final\_energy\_df,SINGLE\_ECONOMY\_ID)

this will check for the difference between values in base and base year +1. If they are greater than a threshold e.g 10%, it will record that as an error and keep looking for more. Then it spits out all the issues. The issues that you are fine with can be recorded into config/expected\_differences\_by\_economy\_between\_2022\_2023.yaml using the supplied .yaml file in the data/temp/errors folder. I am sorry this one will take a while to get used to as well as use but the intention is to catch blatant issues and then record the others as daijoubu.

check\_for\_negatives\_or\_postives\_in\_wrong\_sectors()

this is used in merging results as well as at the end of everything. Its just to be safe as I found a bunch of incorrect signs from modellers.