Hands On Exercise 2: Modifying a service demand

This hands-on will allow users to define their own service demand for an exogenous sector.

Learning objectives

- Define own service demand for an exogenous sector

Adding an exogenous service demand

Hands-on accompanying video: https://youtu.be/btcWsSK5pnw

As a quick example, in the residential sector a service demand could be cooking. Houses require energy to cook food and a technology to service this demand, such as an electric stove.

We will start by looking at the `default` example. This can be found in your MUSE download at `/src/muse/data/example/default/example`, or you can download it at the following link:

https://zenodo.org/record/6340451#.YiiY5y-l1pQ

The file you will need to start with is called `default.zip`, the finished version for your records is called `final_version.zip`

Next, download this and place it in a convenient location on your computer. We will now start by adding a cooking demand to this example. The default example currently only has a service demand of `heat`, so we will need to do some editing.

To achieve this, we will need to edit the`Residential2020Consumption.csv` and `Residential2050Consumption.csv`files found within the `technodata/preset/` directory. The `Residential2020Consumption.csv` file allows us to specify the demand in 2020 for each region and technology per timeslice. The `Residential2050Consumption.csv` file does the same but for the year 2050. The datapoints between these are interpolated. We will explain further details on interpolation in lecture 5.

Firstly, we must add the new service demand`cook` as a column in these two files. Next, we add the demand. We can do this in Excel, or an editor of your choice. This is how it may look like for you when you open the `Residential2020Consumption.csv` file:

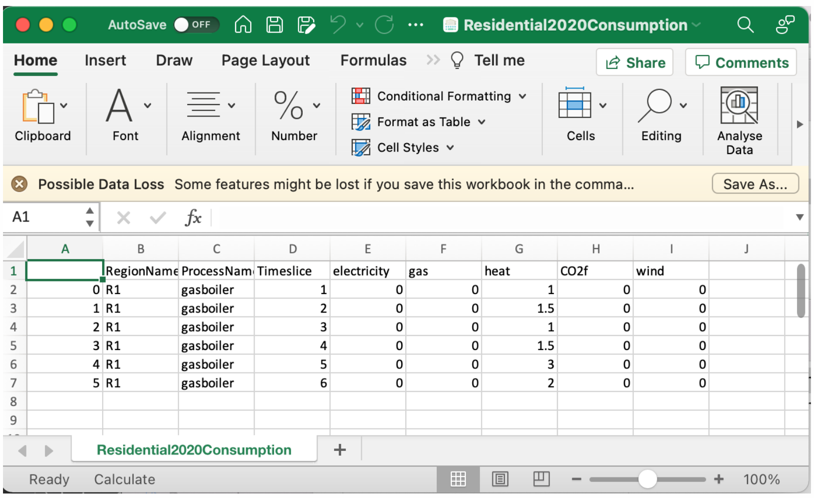
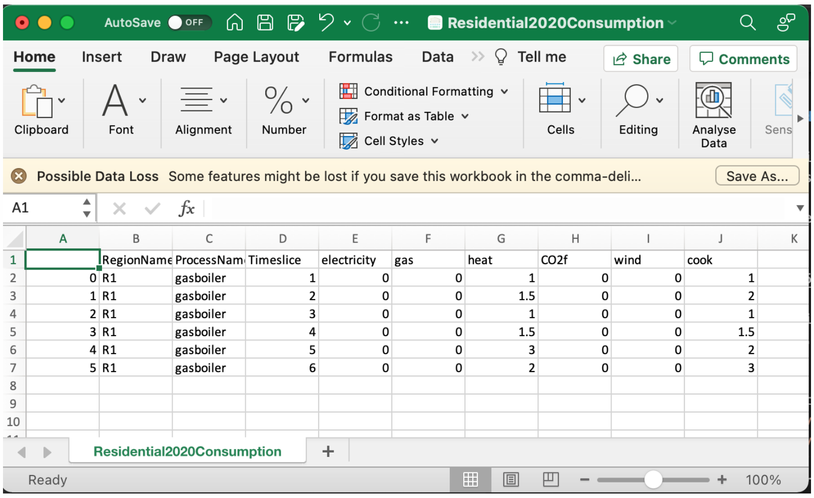


Figure 2.1: Residential2020Consumption file opened in Excel.

We will add a new column called `cook` and enter some values for each timeslice. This can be seen through the addition of a positive number in the `cook` column.



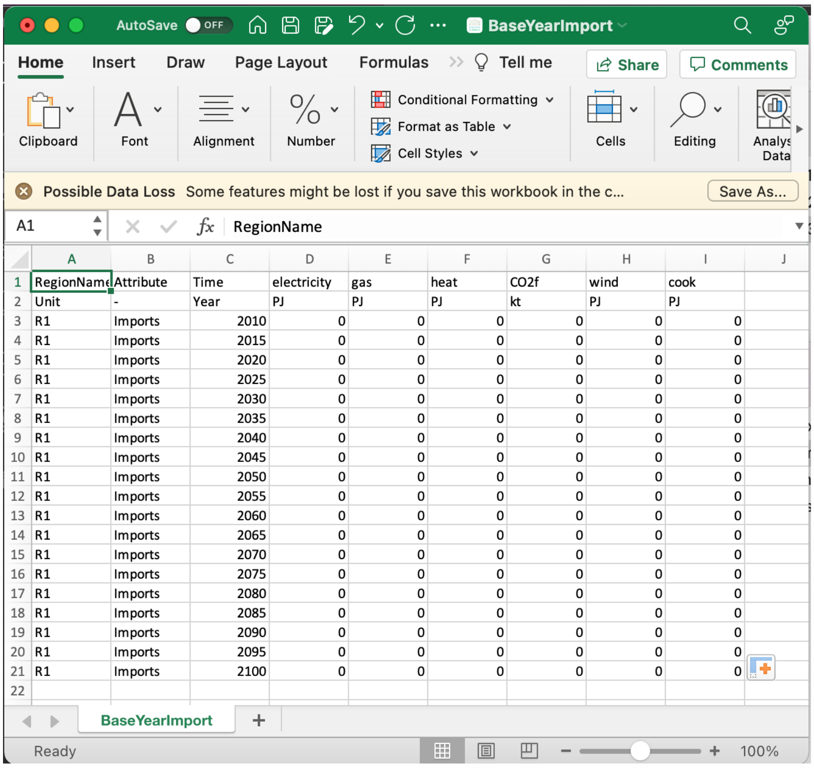
	RegionName	ProcessNam	Timeslice	electricity	gas	heat	CO2f	wind	cook
0	R1	gasboiler	1	0	0	1	0	0	1
1	R1	gasboiler	2	0	0	1.5	0	0	2
2	R1	gasboiler	3	0	0	1	0	0	1
3	R1	gasboiler	4	0	0	1.5	0	0	1.5
4	R1	gasboiler	5	0	0	3	0	0	2
5	R1	gasboiler	6	0	0	2	0	0	3

Figure 2.2: Modified Residential2020Consumption file opened in Excel.

The process is very similar for the `Residential2050Consumption.csv` file, however, for this example, we often placed larger numbers to indicate higher demand in 2050.

Next, we must edit the files within the `input` folder. For this, we must add the cook service demand to each of these files.

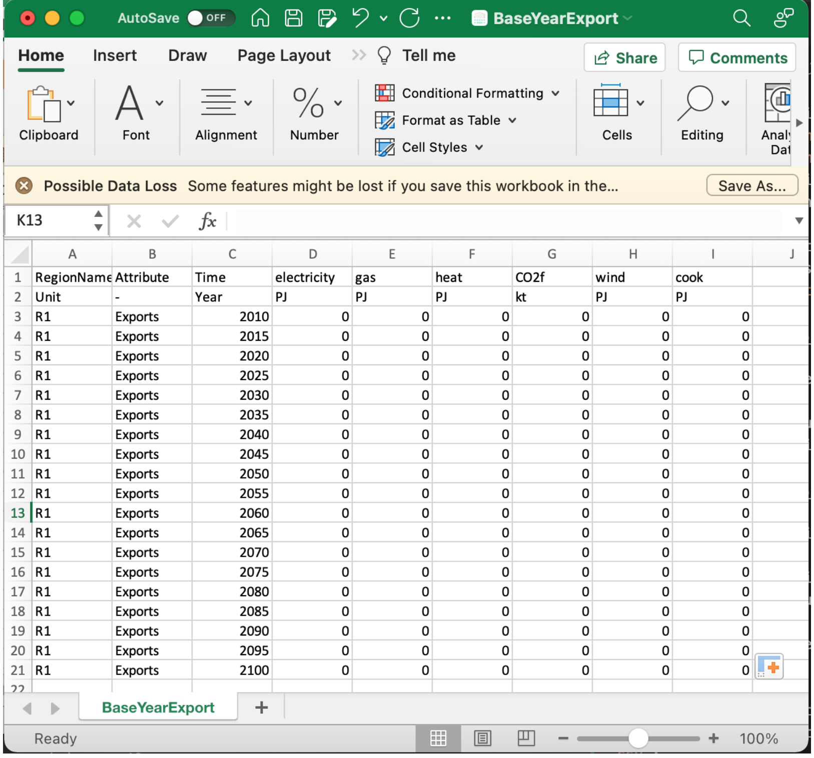
First, we will amend the `BaseYearExport.csv` and `BaseYearImport.csv` files. For this, we say that there is no import or export of the cook service demand. A brief example is outlined below for `BaseYearExport.csv`:



RegionName	Attribute	Time	electricity	gas	heat	CO2f	wind	cook
Unit	-	Year	PJ	PJ	PJ	kt	PJ	PJ
R1	Imports	2010	0	0	0	0	0	0
R1	Imports	2015	0	0	0	0	0	0
R1	Imports	2020	0	0	0	0	0	0
R1	Imports	2025	0	0	0	0	0	0
R1	Imports	2030	0	0	0	0	0	0
R1	Imports	2035	0	0	0	0	0	0
R1	Imports	2040	0	0	0	0	0	0
R1	Imports	2045	0	0	0	0	0	0
R1	Imports	2050	0	0	0	0	0	0
R1	Imports	2055	0	0	0	0	0	0
R1	Imports	2060	0	0	0	0	0	0
R1	Imports	2065	0	0	0	0	0	0
R1	Imports	2070	0	0	0	0	0	0
R1	Imports	2075	0	0	0	0	0	0
R1	Imports	2080	0	0	0	0	0	0
R1	Imports	2085	0	0	0	0	0	0
R1	Imports	2090	0	0	0	0	0	0
R1	Imports	2095	0	0	0	0	0	0
R1	Imports	2100	0	0	0	0	0	0

Figure 2.3: Modified BaseYearImport file opened in Excel.

The same is true for the `BaseYearImport.csv` file:



RegionName	Attribute	Time	electricity	gas	heat	CO2f	wind	cook
Unit	-	Year	PJ	PJ	PJ	kt	PJ	PJ
R1	Exports	2010	0	0	0	0	0	0
R1	Exports	2015	0	0	0	0	0	0
R1	Exports	2020	0	0	0	0	0	0
R1	Exports	2025	0	0	0	0	0	0
R1	Exports	2030	0	0	0	0	0	0
R1	Exports	2035	0	0	0	0	0	0
R1	Exports	2040	0	0	0	0	0	0
R1	Exports	2045	0	0	0	0	0	0
R1	Exports	2050	0	0	0	0	0	0
R1	Exports	2055	0	0	0	0	0	0
R1	Exports	2060	0	0	0	0	0	0
R1	Exports	2065	0	0	0	0	0	0
R1	Exports	2070	0	0	0	0	0	0
R1	Exports	2075	0	0	0	0	0	0
R1	Exports	2080	0	0	0	0	0	0
R1	Exports	2085	0	0	0	0	0	0
R1	Exports	2090	0	0	0	0	0	0
R1	Exports	2095	0	0	0	0	0	0
R1	Exports	2100	0	0	0	0	0	0

Figure 2.4: Modified BaseYearExport file opened in Excel.

Next, we must edit the `GlobalCommodities.csv` file. This is where we define the new commodity `cook`. It tells MUSE the commodity type, name, emissions factor of CO2 and heat rate, amongst other things.

The default version used for this tutorial is below:

Commodity	CommodityT	CommodityN	CommodityE	HeatRate	Unit
Electricity	Energy	electricity	0	1	PJ
Gas	Energy	gas	56.1	1	PJ
Heat	Energy	heat	0	1	PJ
Wind	Energy	wind	0	1	PJ
CO2fuelcom:	Environment	CO2f	0	1	kt

Figure 2.5: Non-edited GlobalCommodities file opened in Excel.

We then add a new row at the bottom to include the cook commodity:

Commodity	CommodityT	CommodityN	CommodityE	HeatRate	Unit
Electricity	Energy	electricity	0	1	PJ
Gas	Energy	gas	56.1	1	PJ
Heat	Energy	heat	0	1	PJ
Wind	Energy	wind	0	1	PJ
CO2fuelcom:	Environment	CO2f	0	1	kt
Cook	**Energy**	**cook**	**0**	**1**	**PJ**

Figure 2.6: Edited GlobalCommodities file opened in Excel.

The `CommodityName` column must be consistent internally within the model. Whereas the `Commodity` column is for your reference.

Finally, the `Projections.csv` file must be changed. This is a large file which details the expected future costs of the technology in the first benchmark year of the simulation, the subsequent and actual simulated costs will be calculated during the running of the model. We have highlighted in **bold** the changed column for this example.

RegionName	Attribute	Time	electricity	gas	heat	CO2f	wind	**cook**
Unit	-	Year	MUS$2010/I	MUS$2010/I	MUS$2010/I	MUS$2010/I	MUS$2010/I	**MUS$2010/I**
R1	CommodityP	2010	14.8148147	6.6759	100	0	0	**100**
R1	CommodityP	2015	17.8981481	6.914325	100	0.05291385	0	**100**
R1	CommodityP	2020	19.5	7.15275	100	0.08314119	0	**100**
R1	CommodityP	2025	21.9351853	8.10645	100	0.1200698	0	**100**
R1	CommodityP	2030	26.5092592	9.06015	100	0.1569984	0	**100**
R1	CommodityP	2035	26.5185186	9.2191	100	0.21487757	0	**100**
R1	CommodityP	2040	23.8518519	9.37805	100	0.27275673	0	**100**
R1	CommodityP	2045	23.9722222	9.19382934	100	0.35394801	0	**100**
R1	CommodityP	2050	24.0648147	9.00960867	100	0.43513929	0	**100**
R1	CommodityP	2055	25.3425925	8.8326256	100	0.54236558	0	**100**
R1	CommodityP	2060	25.5370369	8.65564253	100	0.64959187	0	**100**
R1	CommodityP	2065	25.3240742	8.48561271	100	0.78089262	0	**100**
R1	CommodityP	2070	23.3611111	8.31558288	100	0.91219338	0	**100**
R1	CommodityP	2075	22.2777778	8.15223313	100	1.07832169	0	**100**
R1	CommodityP	2080	22.2592592	7.98888337	100	1.24445	0	**100**
R1	CommodityP	2085	22.1759258	7.83195124	100	1.4253503	0	**100**
R1	CommodityP	2090	22.0370369	7.6750191	100	1.6062506	0	**100**
R1	CommodityP	2095	21.9444444	7.52425246	100	1.73877515	0	**100**
R1	CommodityP	2100	21.3981481	7.37348582	100	1.8712997	0	**100**

Figure 2.7: Edited Projections file opened in Excel.

Addition of a cooking technology

Next, we must add a technology to service this new demand. During this process we must be careful to specify the end-use of the technology as `cook`, which is case-sensitive.

For this example, we will add two competing technologies to the residential sector to service the cooking demand: `electric_stove` and `gas_stove` to the `Technodata.csv` file in `/technodata/residential/Technodata.csv`.

For this, we copy the `gasboiler` row for `R1` and paste it for the new `electric_stove`. For `gas_stove` we copy and paste the data for `heatpump` from region R1. In the figure below we show this, but only show the first few columns for the interest of space. We will also relax the growth constraints to ensure that the growth in technologies can meet demand.

The growth constraints are:

- MaxCapacityAddition: The maximum absolute capacity that the technology can grow in a single year.
- MaxCapacityGrowth: The maximum percentage that the technology can grow in a particular year.
- TotalCapacityLimit: The total absolute number that cannot be exceeded for a particular technology.

Due to space constraints we can't show all the values as the technodata file is very wide, but we can set the parameters to be the following for both technologies:

- MaxCapacityAddition: 100 PJ
- MaxCapacityGrowth: 20 PJ
- TotalCapacityLimit: 120 PJ

ProcessNam	RegionName	Time	Level	cap_par	cap_exp	fix_par	fix_exp	var_par	var_exp	MaxCapacity	MaxCapacity	TotalCapacit
Unit	-	Year	-	MUS$2010/I	-	MUS$2010/I	-	MUS$2010/I	-	PJ	%	PJ
gasboiler	R1	2020	fixed	3.8	1	0	1	0	1	10	0.02	60
heatpump	R1	2020	fixed	8.866667	1	0	1	0	1	10	0.02	60
gas_stove	**R1**	**2020**	**fixed**	**3.8**	**1**	**0**	**1**	**0**	**1**	**100**	**20**	**120**
electric_stov	**R1**	**2020**	**fixed**	**8.866667**	**1**	**0**	**1**	**0**	**1**	**100**	**20**	**120**

Figure 2.8: Edited technodata file opened in Excel.

As can be seen we have added two technologies with different cap_par costs to each other. We specified their respective fuels, and the enduse for both is cook.

We must also add the data for these new technologies to the following files:

- CommIn.csv
- CommOut.csv
- ExistingCapacity.csv

The `CommIn.csv` file details the input commodities for each technology. In this case, the inputs are gas and electricity. The `CommOut` file details the outputs of the technology, which will be the `cook` commodity.

We must add the input to each of the technologies (gas and electricity for `electric_stove` and `gas_stove` respectively), outputs of `cook` for both and the existing capacity for each technology.

ProcessNam	RegionName	Time	Level	electricity	gas	heat	CO2f	wind	**cook**
Unit	-	Year	-	PJ/PJ	PJ/PJ	PJ/PJ	kt/PJ	PJ/PJ	**PJ/PJ**
gasboiler	R1	2020	fixed	0	1.16	0	0	0	**0**
heatpump	R1	2020	fixed	0.4	0	0	0	0	**0**
electric_stov	**R1**	**2020**	**fixed**	**1.16**	**0**	**0**	**0**	**0**	**0**
gas_stove	**R1**	**2020**	**fixed**	**0**	**1.16**	**0**	**0**	**0**	**0**

Figure 2.9: Edited CommIn file opened in Excel.

Notice in Figure 2.9 that we had to add a column for the new`cook`. We must also do the same for the CommOut file, below:

ProcessNam	RegionName	Time	Level	electricity	gas	heat	CO2f	wind	**cook**
Unit	-	Year	-	PJ/PJ	PJ/PJ	PJ/PJ	kt/PJ	PJ/PJ	**PJ/PJ**
gasboiler	R1	2020	fixed	0	0	1	64.71	0	**0**
heatpump	R1	2020	fixed	0	0	1	0	0	**0**
electric_stov	**R1**	**2020**	**fixed**	**0**	**0**	**0**	**0**	**0**	**1**
gas_stove	**R1**	**2020**	**fixed**	**0**	**0**	**0**	**64.71**	**0**	**1**

Figure 2.10: Edited CommOut file opened in Excel.

We must do this for the `gas` and `power` sector as well. This is just for consistency within MUSE.

Next, we must edit the `residsential/ExistingCapacity.csv` file to detail how much existing capacity there is in the base year and beyond. The existing capacity details power plants, or other technologies, which are already installed in the real world and therefore not invested in by the model. It is important to have a clear idea about the real world system in the base year before we run MUSE.

ProcessNam	RegionName	Unit	2020	2025	2030	2035	2040	2045	2050
gasboiler	R1	PJ/y	10	5	0	0	0	0	0
heatpump	R1	PJ/y	0	0	0	0	0	0	0
electric_stov	**R1**	**PJ/y**	**10**	**5**	**0**	**0**	**0**	**0**	**0**
gas_stove	**R1**	**PJ/y**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Figure 2.11: Edited ExistingCapacity file opened in Excel.

Due to the additional demand for gas and electricity brought on by the new cook demand, it is necessary to relax the growth constraints for `gassupply1` in the `technodata/gas/technodata.csv` file. For this example, we set this file as follows (see bold cells):

ProcessNam	RegionName	Time	Level	cap_par	cap_exp	fix_par	fix_exp	var_par	var_exp	MaxCapacity	MaxCapacity	TotalCapacit
Unit	-	Year	-	MUS$2010/I	-	MUS$2010/I	-	MUS$2010/I	-	PJ	%	PJ
gassupply1	R1	2020	fixed	0	1	0	1	2.55	1	**100**	**5**	**500**

Figure 2.12: Edited gas/technodata file opened in Excel.

We must also ensure there are no `0` in the `ExistingCapacity.csv` for any of the sectors. This is because the MUSE model will produce an error. For error debugging it is helpful to go to the MUSE google groups. So to do this, go through the `gas/ExistingCapacity.csv` and `power/ExistingCapacity.csv` and replace them with a non-zero value, such as `0.01`. Below is an example for the `gas` sector:

ProcessNam	RegionName	Unit	2020	2025	2030	2035	2040	2045	2050
gassupply1	R1	PJ/y	15	15	7.5	**0.01**	**0.01**	**0.01**	**0.01**

Figure 2.13: Edited gas/ExistingCapacity.csv file opened in Excel.

Next, we must run the simulation with our modified input files using the following command in the directory where you saved the default example. To do this follow the instructions shown in hands-on 1:

```
python -m pip muse settings.toml
```

The figure below shows the results for this new demand in the residential sector:

sector	residential		

Sum of capacity	Column Labels		
Row Labels	electric_stove	gas_stove	Grand Total
2020	10	0.01	10.01
2025	18.99	0.01	19
2030	19.99	0.01	20
2035	20.99	0.01	21
2040	21.99	0.01	22
2045	22.99	0.01	23
2050	23.99	0.01	24
Grand Total	**138.94**	**0.07**	**139.01**

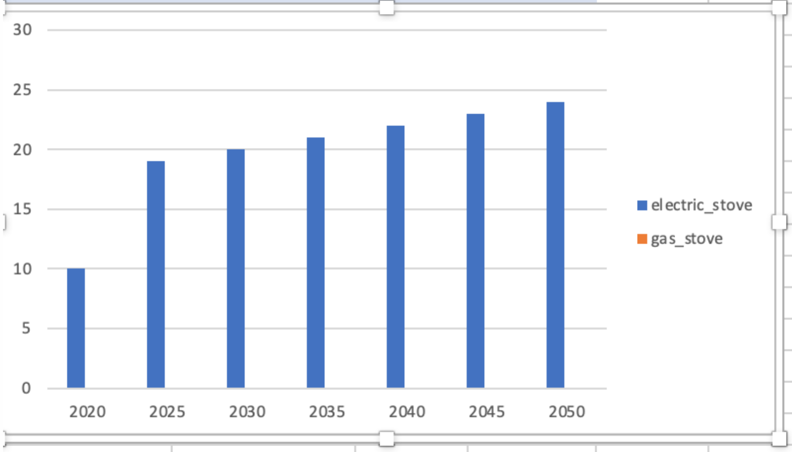


Figure 2.14: Capacity results for the residential sector.

We can see that `electric_stove` takes over completely. This is because of the lower `cap_par` value when compared to `gas_stove`. Do not be surprised if your results differ from this, as the MUSE model will change over time. The important thing is to understand the outputs from the inputs.

For the final example input data (`final_version.zip`) showed in this tutorial and results spreadsheet, please refer to the link below:

https://zenodo.org/record/6340451#.YiiY5y-l1pQ