

Querying GCAM Outputs

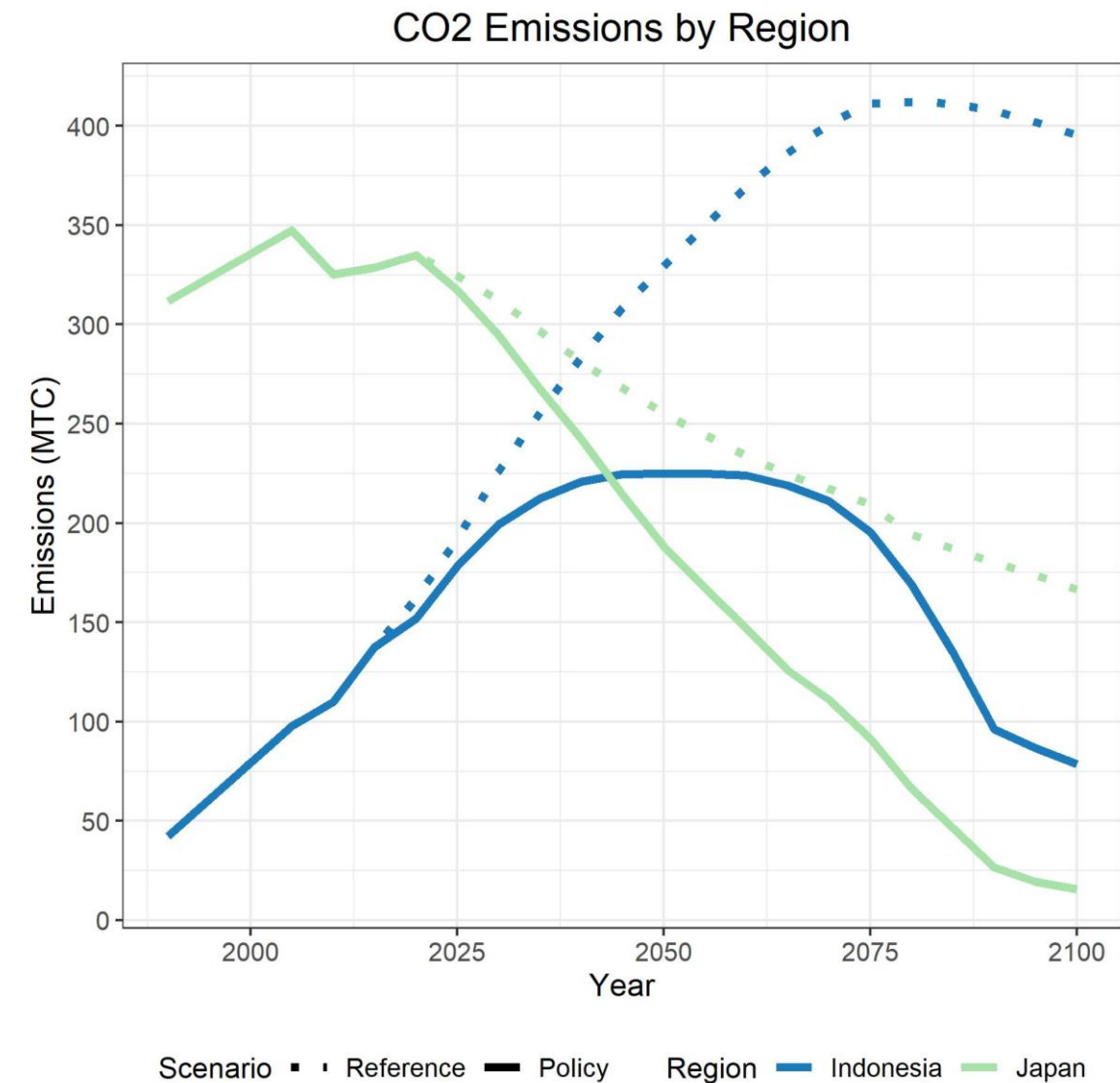
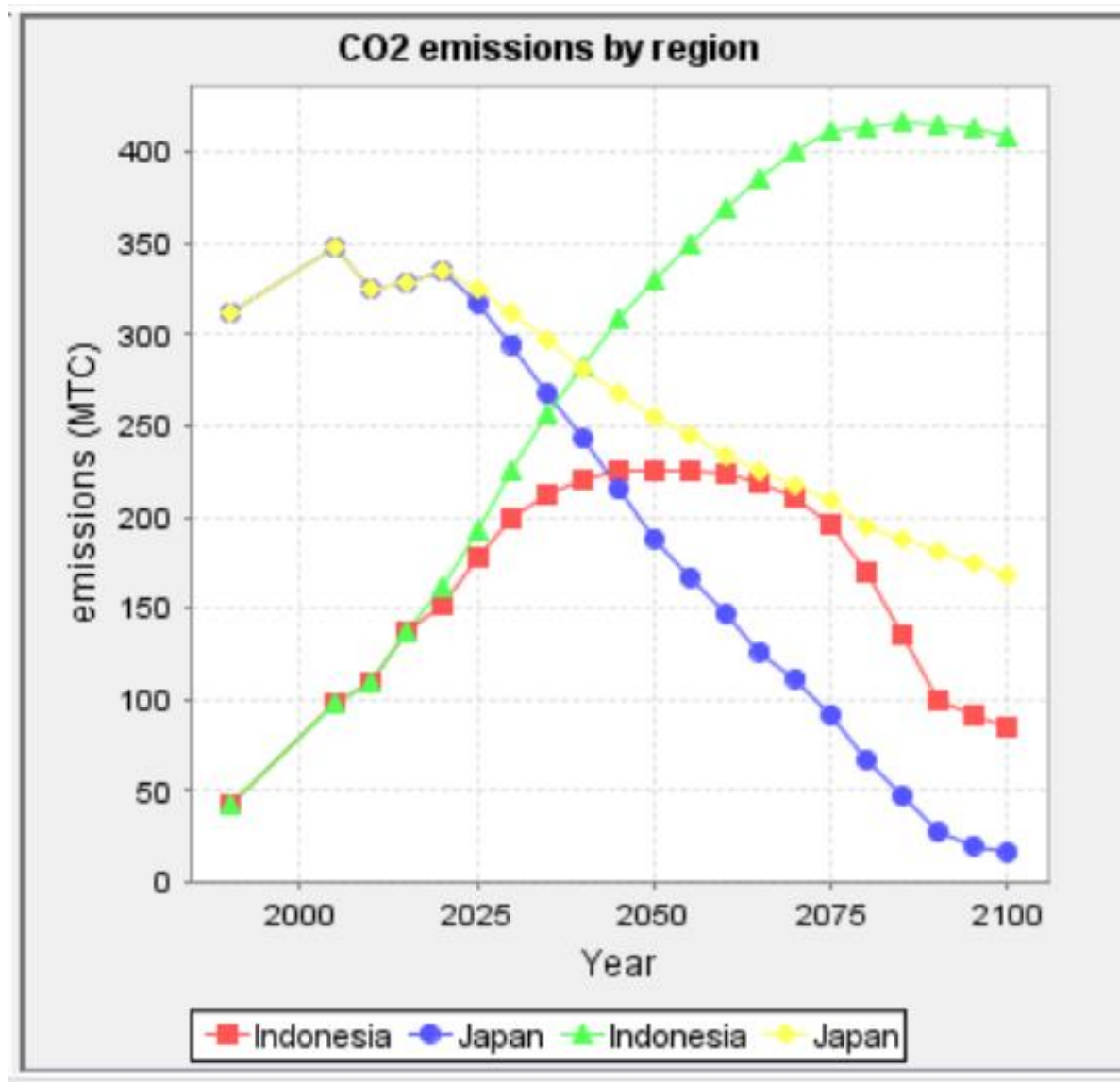
8 June 2023

Maridee Weber

Congrats, you ran GCAM! Now what?

Model Interface

rgcam



Model Interface

Model Interface (MI) is a program that comes with the Release Version of GCAM, and is used to view GCAM output.

To download MI: <https://github.com/JGCRI/gcam-core/releases/tag/gcam-v6.0>

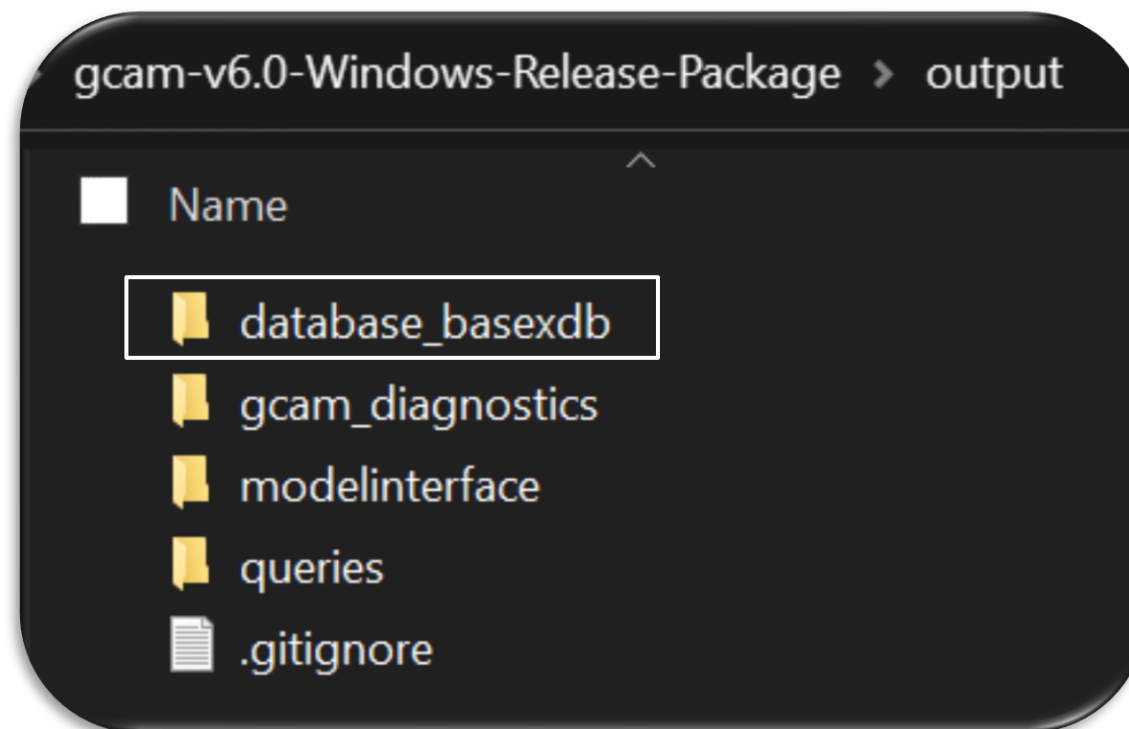
Download the .zip for your operating system, and extract all files.



Model Interface

To view GCAM results in MI, we first need a database.

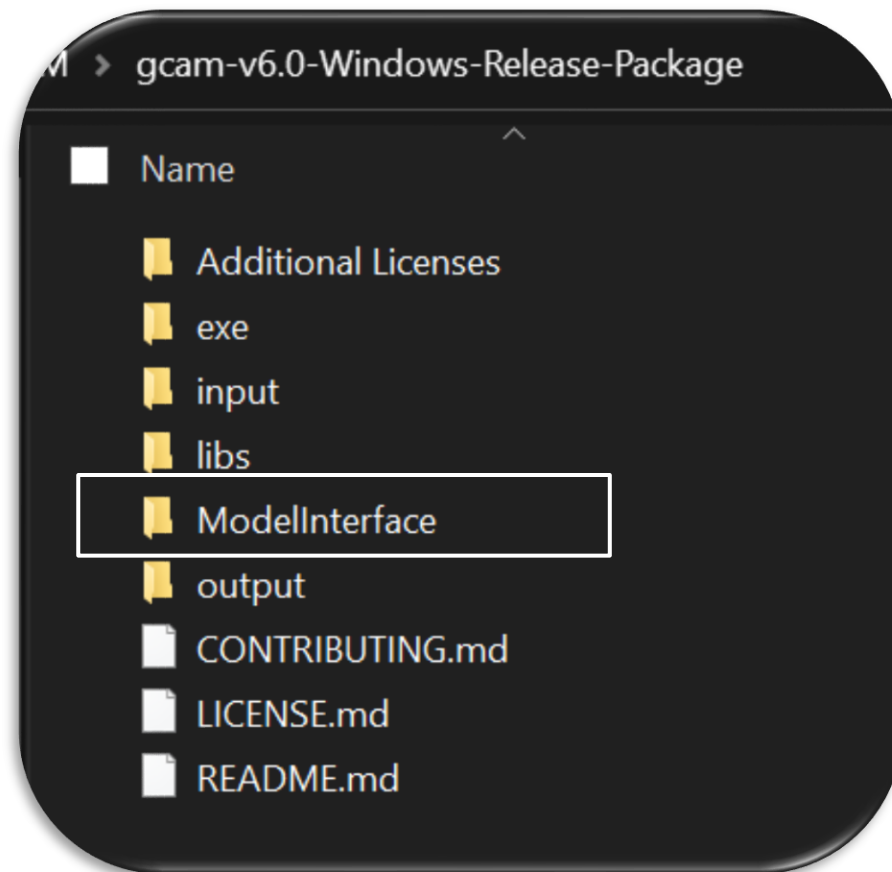
After running GCAM, you will have a database in the **output** folder. It will be called “**database_basexdb**”, unless you changed the name.



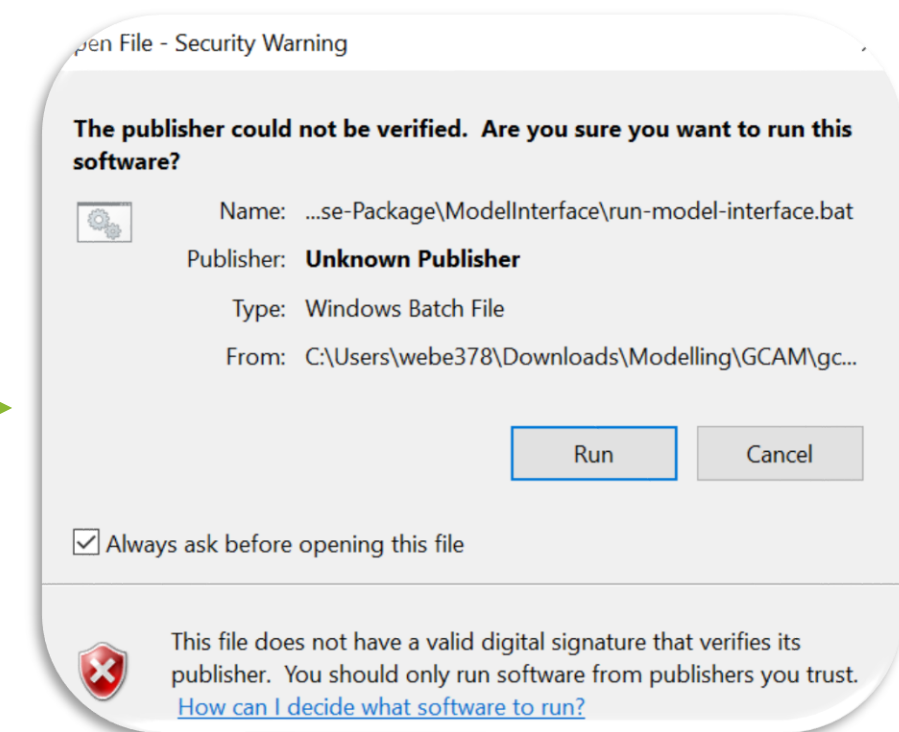
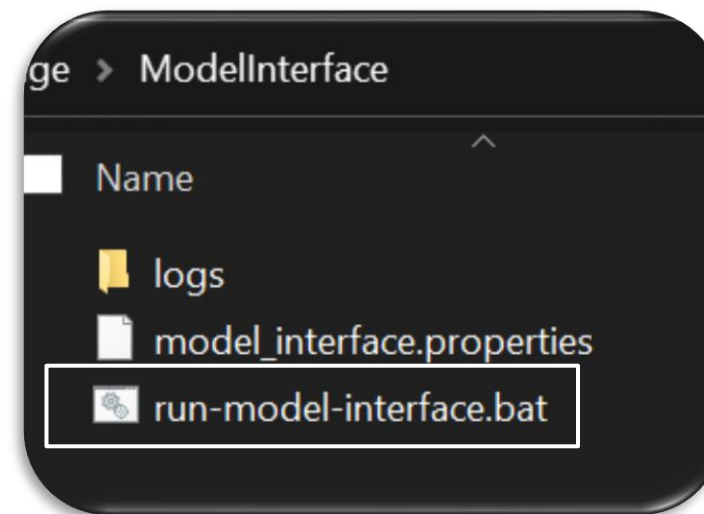
Model Interface

Model Interface lives in the
ModelInterface folder.

Run

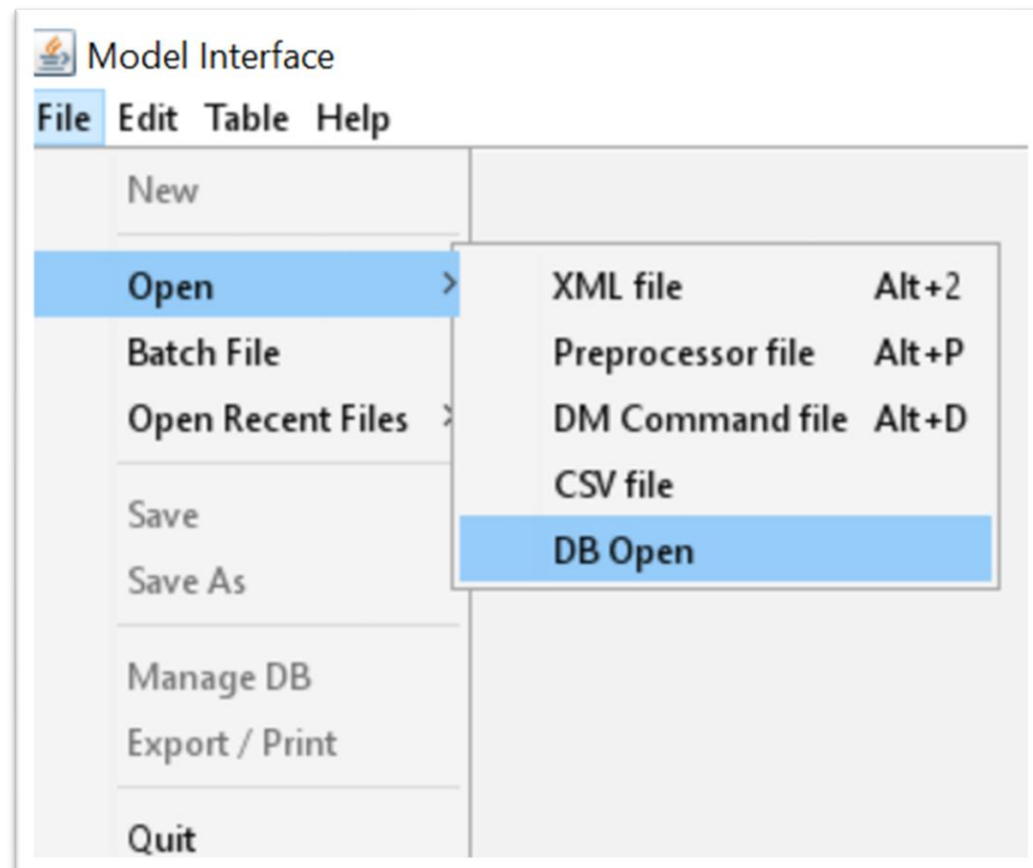


run-model-interface.bat

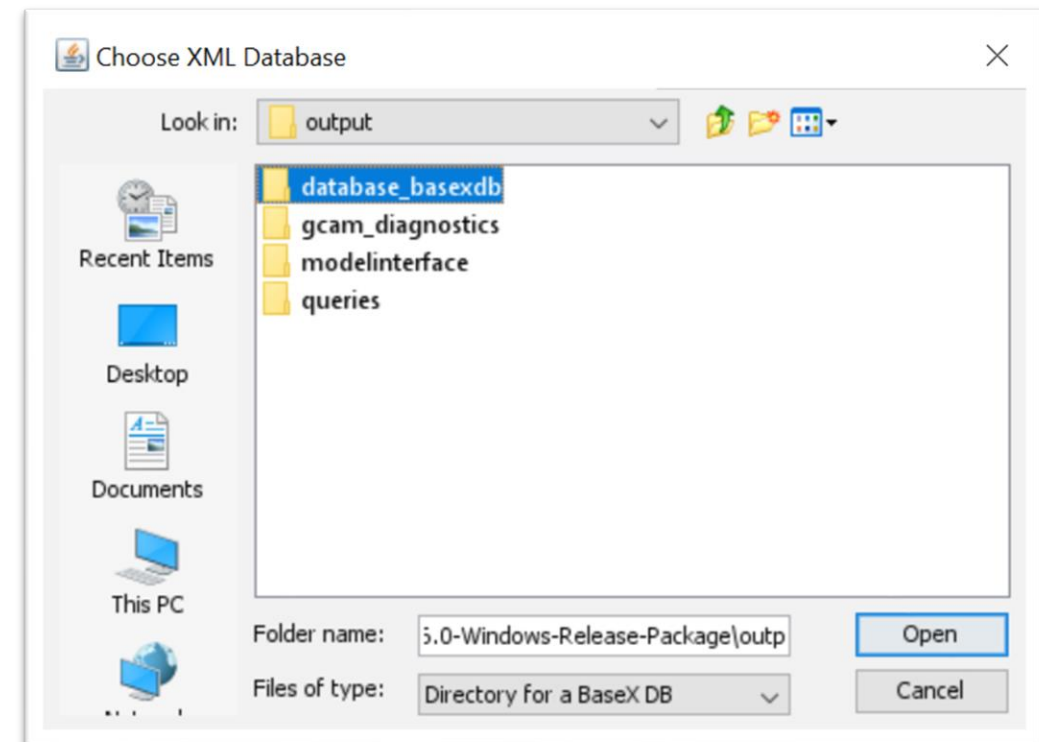


Model Interface

Once open, connect a database by going to
File > Open > DB Open



Then, select your database
and click **Open**



Model Interface

Scenario

Regions

Reference 2023-22-5T12:00

Policy 2023-22-5T12:00

USA

Africa_Eastern

Africa_Northern

Africa_Southern

Africa_Western

Australia_NZ

Brazil

Canada

Central America and Caribbean

Central Asia

China

EU-12

EU-15

Europe_Eastern

Europe_Non_EU

European Free Trade Association

India

Indonesia

Japan

Mexico

Middle East

Pakistan

Russia

South Africa

South America_Northern

South America_Southern

queries

energy

primary energy

primary energy consumption by region (avg fossil efficiency)

primary energy consumption by region (direct equivalent)

primary energy consumption with CCS by region (direct equivalent)

resource production

resource production by tech and vintage

resource supply curves

regional primary energy prices

energy transformation

electricity

elec gen by region (incl CHP)

elec gen by subsector

elec gen by gen tech

elec gen by gen tech and cooling tech

elec gen by gen tech and cooling tech and vintage

elec gen by gen tech and cooling tech (new)

elec energy input by subsector

elec energy input by elec gen tech

elec energy input by elec gen tech and cooling tech

elec prices by sector

elec gen costs by subsector

elec gen costs by tech

elec gen costs by cooling tech

Run Query

Update Single Queries

Create

Remove

Edit

Model Interface

Scenario
Reference 2023-22-5T
Policy 2023-22-5T12:07

Regions
Europe_Eastern
Europe_Non_EU
European Free Trade Association
India
Indonesia
Japan
Mexico
Middle East
Pakistan
Russia
South Africa

queries
energy
agriculture and land use
emissions
CO2 emissions
CO2 emissions by region
Could not generate list.
CO2 emissions by sector
CO2 emissions by sector (no bio)
CO2 emissions by assigned sector (no bio)

Run Query

Update Single Queries
Create
Remove
Edit

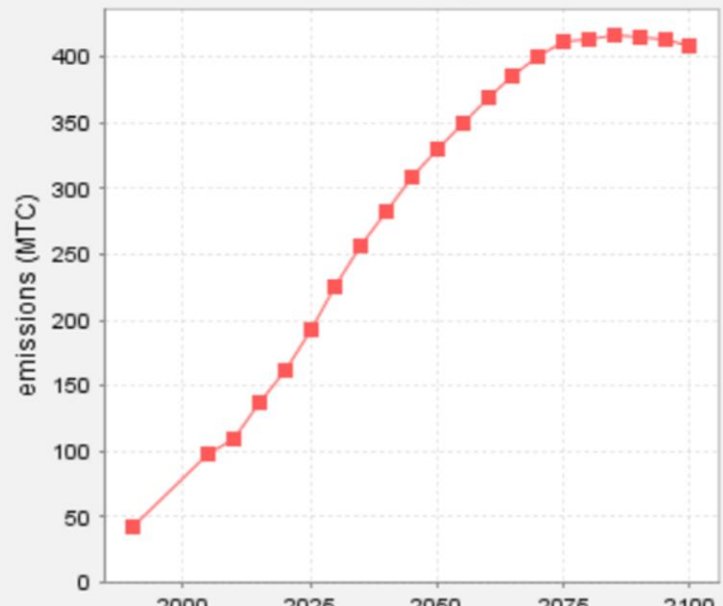
CO2 emissions by region

scenario	region	1990	2005	2010	2015	2020	2025	2030	2035	2040	2045	2050	2055	2060
Reference,...	Indonesia	42.389	97.728	110.075	137.494	162.067	192.107	225.71	255.719	283.315	308.313	329.209	349.632	360.000

This database had two scenarios, “Reference” and “Policy”.

To view results, select a scenario, region, and a query.

CO2 emissions by region



emissions (MTC)

Year

Indonesia

Model Interface

Scenario

Reference 2023-22-5T12:07

Policy 2023-22-5T12:07

Regions

Europe_Eastern

Europe_Non_EU

European Free Trade Association

India

Indonesia

Japan

Mexico

Middle East

Pakistan

Russia

South Africa

queries

energy

agriculture and land use

emissions

CO2 emissions

CO2 emissions by region

CO2 emissions by sector

CO2 emissions by sector (no bio)

Run Query

Update Single Queries

Create

Remove

Edit

CO2 emissions by region

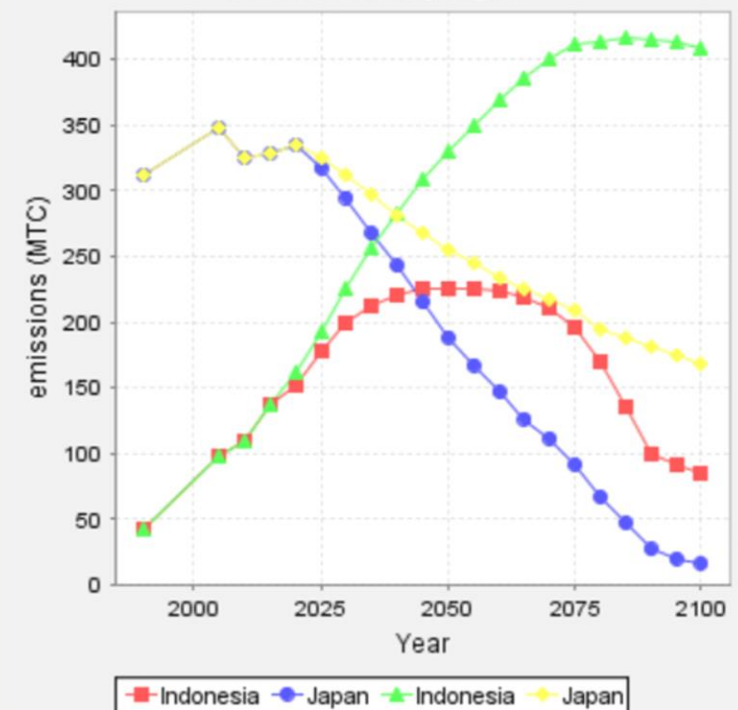
CO2 emissions by sector

scenario	region	1990	2005	2010	2015	2020	2025	2030	2035	2040	2045	2050	2100
Policy, date...	Indonesia	42.389	97.728	110.075	137.494	152.029	178.331	199.191	212.253	220.771	224.671	225.029	2100
Policy, date...	Japan	311.696	347.475	325.235	328.696	334.968	317.536	294.83	267.61	242.678	214.934	188.622	1
Reference, ...	Indonesia	42.389	97.728	110.075	137.494	162.067	192.107	225.71	255.719	283.315	308.313	329.209	3
Reference, ...	Japan	311.696	347.475	325.235	328.696	334.811	324.774	312.353	296.722	281.215	268.341	255.603	2

You can select multiple scenarios, regions, and queries at the same time.

You can also click and drag queries from Model Interface into Excel.

CO2 emissions by region



rgcam

rgcam is an R package that makes querying and visualizing GCAM output easy.

- Connecting a database
- Querying GCAM outputs
- Visualizing

To install rgcam, open an R session and run:

```
install_github('JGCRI/rgcam', build_vignettes=TRUE)
```

Additional rgcam information: <https://github.com/JGCRI/rgcam>

rgcam – Connecting a database

Running the `localDBConn` command **connects your database** to rgcam and lists the scenarios available in that database.

```
17 # 1) Connecting a database
18 # localDBConn(`path to your database`, `name of database`)
19 db1 <- localDBConn('./gcam-v6.0-Windows-Release-Package/output', 'database_basexdb')
```

```
> # 1) Connecting a database
> # localDBConn(`path to your database`, `name of database`)
> db1 <- localDBConn('./gcam-v6.0-Windows-Release-Package/output', 'database_basexdb')
Database scenarios: Reference, Policy
```

rgcam – Querying GCAM output

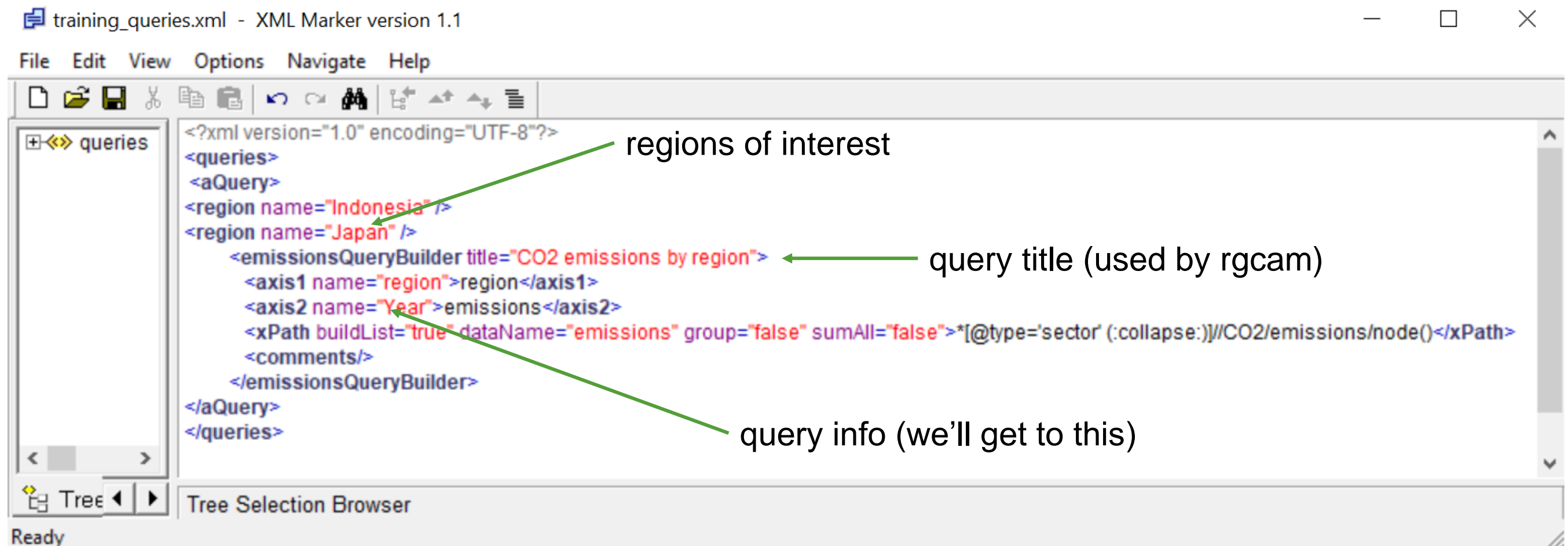
Running the `addScenario` command **extracts the query data** for a scenario in a database.

```
24 # 2) Querying GCAM output
25 # addScenario(database object, `name of file to store rgcam data`, `name of scenario`, `path to/name of query XML`)
26 prj <- addScenario(db1, 'Training.dat', 'Reference', '../GCAM Training/rgcam/training_queries.xml', clobber = TRUE)
27 prj <- addScenario(db1, 'Training.dat', 'Policy', '../GCAM Training/rgcam/training_queries.xml', clobber = TRUE)
```

```
> # 2) Querying GCAM output
> # addScenario(database object, `name of file to store rgcam data`, `name of scenario`, `path to/name of query XML`)
> prj <- addScenario(db1, 'Training.dat', 'Reference', '../GCAM Training/rgcam/training_queries.xml', clobber = TRUE)
Scenario Reference does not exist in this project. Creating.
> prj <- addScenario(db1, 'Training.dat', 'Policy', '../GCAM Training/rgcam/training_queries.xml', clobber = TRUE)
Scenario Policy does not exist in this project. Creating.
```

rgcam – Querying GCAM output

Running the addScenario command **extracts the query data** for a scenario in a database.



training_queries.xml - XML Marker version 1.1

File Edit View Options Navigate Help

queries

```
<?xml version="1.0" encoding="UTF-8"?>
<queries>
  <aQuery>
    <region name="Indonesia" />
    <region name="Japan" />
    <emissionsQueryBuilder title="CO2 emissions by region">
      <axis1 name="region">region</axis1>
      <axis2 name="Year">emissions</axis2>
      <xPath buildList="true" dataName="emissions" group="false" sumAll="false">*[@type='sector' (:collapse:)]//CO2/emissions/node()</XPath>
      <comments/>
    </emissionsQueryBuilder>
  </aQuery>
</queries>
```

regions of interest

query title (used by rgcam)

query info (we'll get to this)

Tree Selection Browser

Ready

rgcam – Querying GCAM output

Running the addScenario command **extracts the query data** for a scenario in a database.

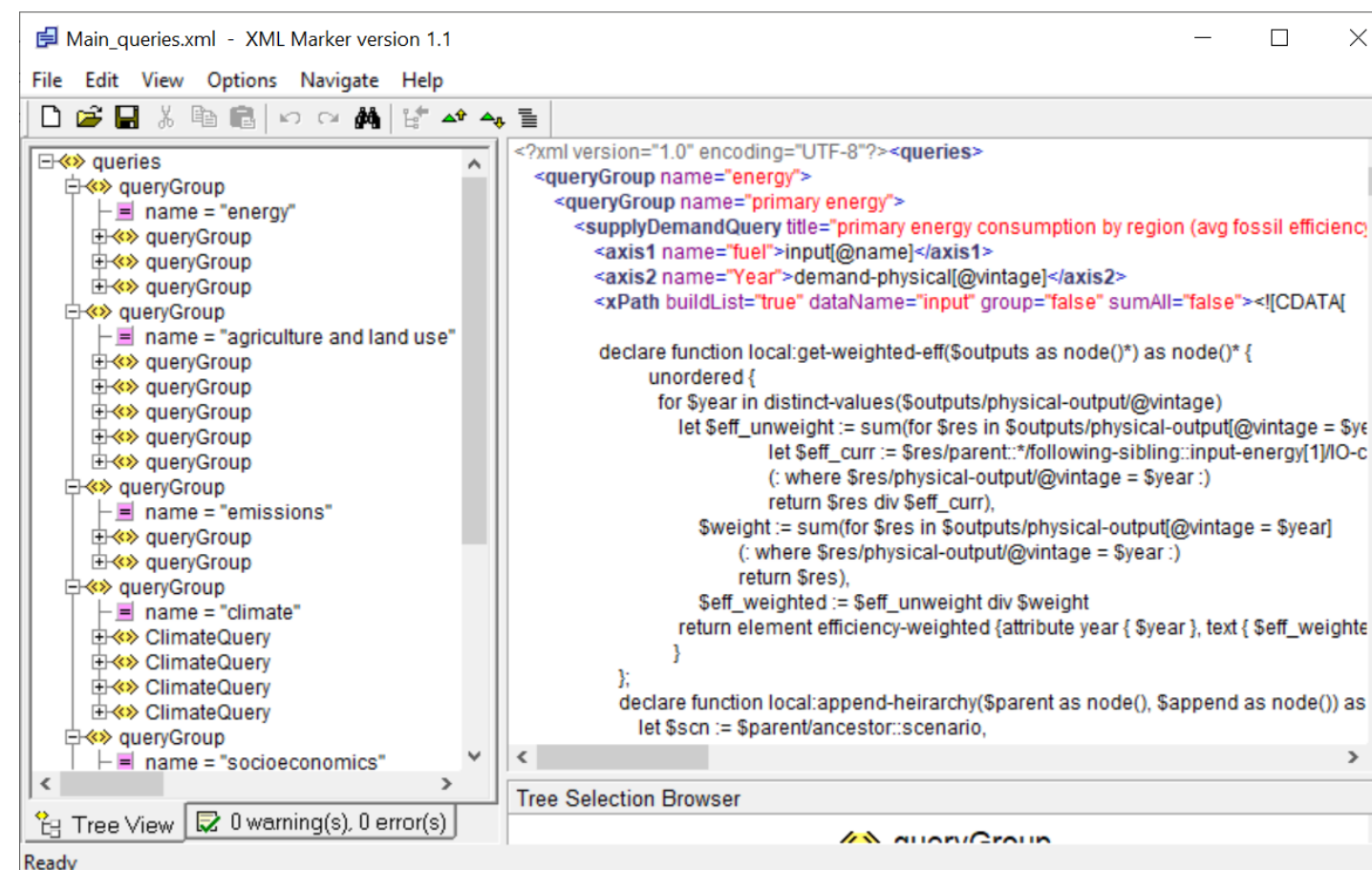
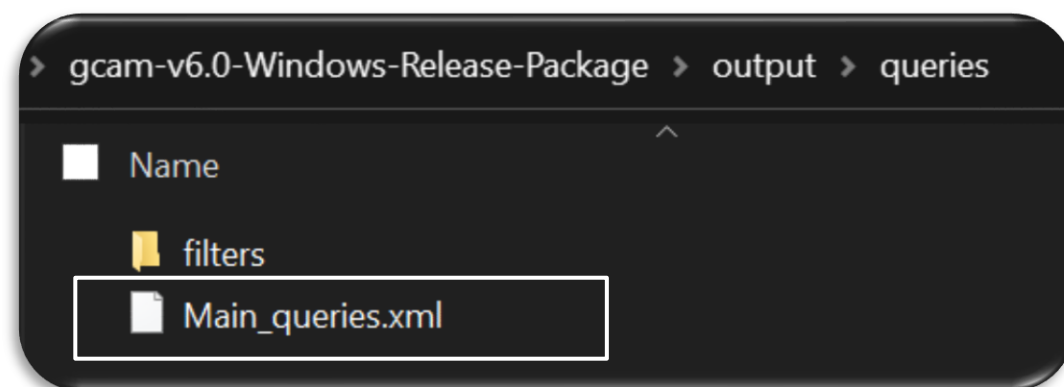


rgcam – Querying GCAM output

To query in rgcam, we need an XML that provides the query info.

We can utilize the **Main_queries.xml** to get this info.

Main_queries.xml can be found in **output/queries** folder.



rgcam – Querying GCAM output

- 1) Copy + paste the query you want from `Main_queries.xml` into your XML used in rgcam (`training_queries.xml`)
- 2) Add regions
- 3) Add `<aQuery>` before, and `</aQuery>` after.

Main_queries.xml

```
<queryGroup name="emissions">
  <queryGroup name="CO2 emissions">
    <emissionsQueryBuilder title="CO2 emissions by region">
      <axis1 name="region">region</axis1>
      <axis2 name="Year">emissions</axis2>
      <xPath buildList="true" dataName="emissions" group="fa
    <comments/>
  </emissionsQueryBuilder>
```

training_queries.xml

```
<queries>
  <aQuery>
    <region name="Indonesia" />
    <region name="Japan" />
    <emissionsQueryBuilder title="CO2 emissions by region">
      <axis1 name="region">region</axis1>
      <axis2 name="Year">emissions</axis2>
      <xPath buildList="true" dataName="emissions" group="fa
    <comments/>
  </emissionsQueryBuilder>
</aQuery>
</queries>
```

add your next query here, before `</queries>` →

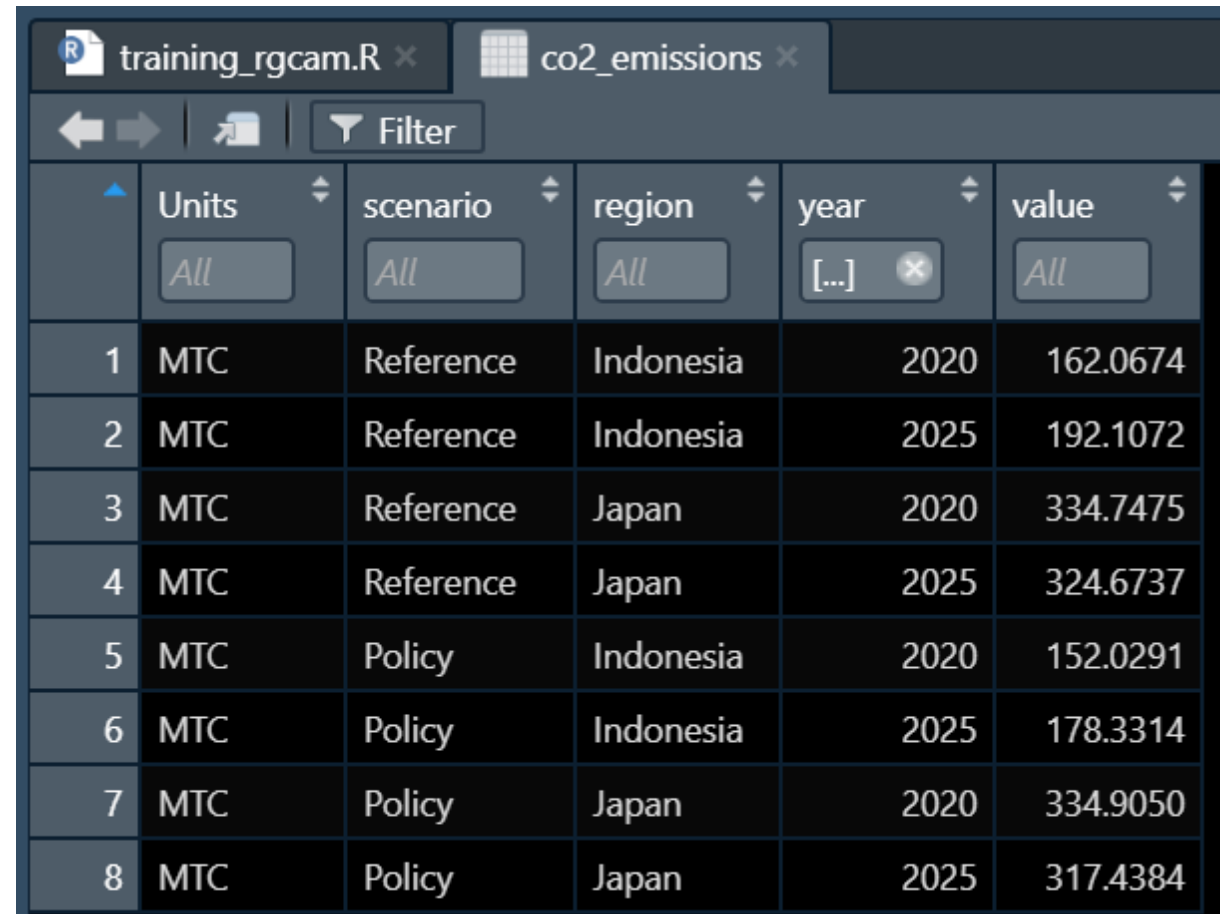
rgcam – Querying GCAM output

Running the `getQuery` command **assigns query data** to an object.

```
32 ## Retrieve queries for all scenarios in the database,  
33 # getQuery(project object, `query title from XML`)  
34 co2_emissions <- getQuery(prj, "CO2 emissions by region")
```

`training_queries.xml` had one query in it, so we retrieved one table.

We now have a table called `co2_emissions` that contains data from two scenarios, and two regions.



The screenshot shows an RStudio window with two tabs: 'training_rgcam.R' and 'co2_emissions'. The 'co2_emissions' tab is active, displaying a data table with 8 rows and 6 columns. The columns are 'Units', 'scenario', 'region', 'year', and 'value'. The 'Units' column has a dropdown menu set to 'All'. The 'scenario' column has a dropdown menu set to 'All'. The 'region' column has a dropdown menu set to 'All'. The 'year' column has a dropdown menu set to '2020'. The 'value' column has a dropdown menu set to 'All'. The table contains data for two scenarios (Reference and Policy) and two regions (Indonesia and Japan) for the years 2020 and 2025.

	Units	scenario	region	year	value
1	MTC	Reference	Indonesia	2020	162.0674
2	MTC	Reference	Indonesia	2025	192.1072
3	MTC	Reference	Japan	2020	334.7475
4	MTC	Reference	Japan	2025	324.6737
5	MTC	Policy	Indonesia	2020	152.0291
6	MTC	Policy	Indonesia	2025	178.3314
7	MTC	Policy	Japan	2020	334.9050
8	MTC	Policy	Japan	2025	317.4384

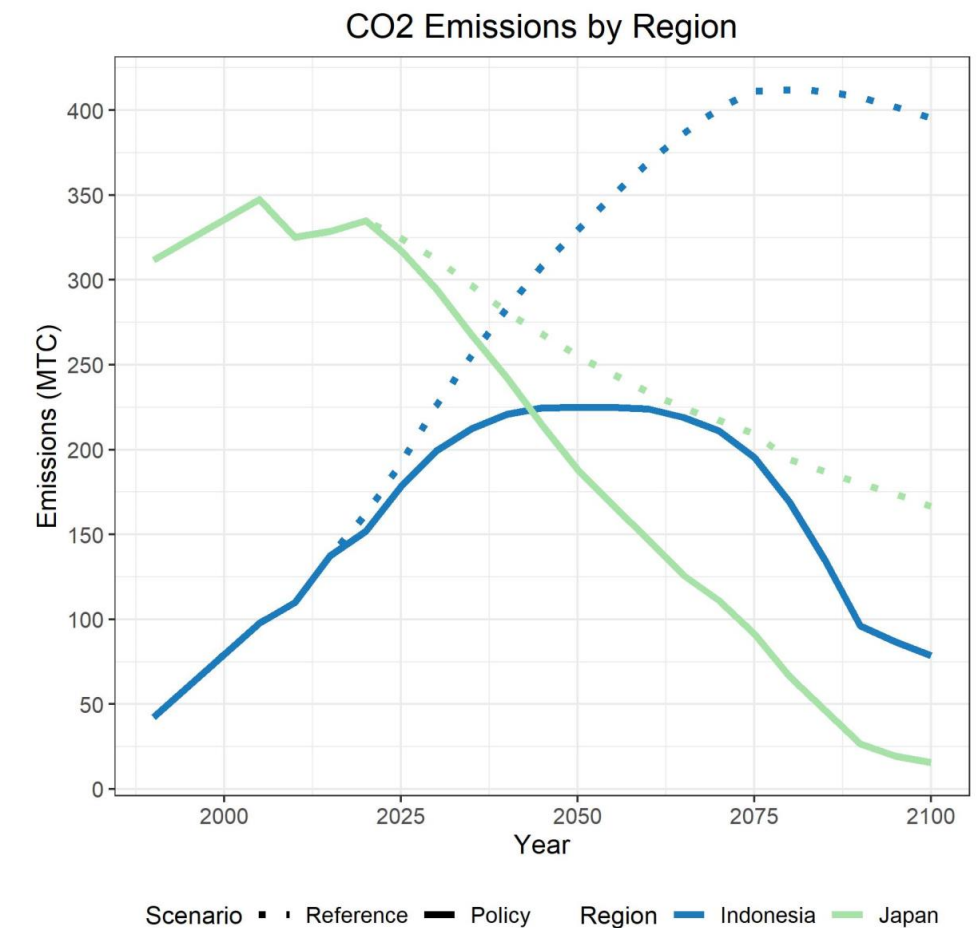
rgcam - Visualizing

Use your data visualization tools of choice, and add new databases, scenarios, and queries as needed.

```

39 # ===== CO2 Emissions
40 country_color <- c("Indonesia" = "#1C7BBC",
41                   "Japan" = "#A7E3A8")
42
43 scenario_line <- c("Reference" = "dotted",
44                  "Policy" = "solid")
45
46 co2_emissions %>%
47   filter(year > 1975) %>%
48   rename("MTC" = value,
49          "Year" = year,
50          "Scenario" = scenario,
51          "Region" = region) %>%
52   ggplot(aes(Year, MTC)) +
53   geom_line(size = 1.5, aes(color = Region, linetype = Scenario)) +
54   scale_color_manual(values = country_color) +
55   scale_linetype_manual(values = scenario_line) +
56   scale_x_continuous(name = "Year", breaks = c(2000, 2025, 2050, 2075, 2100)) +
57   scale_y_continuous(name = "Emissions (MTC)", breaks = c(0, 50, 100, 150, 200, 250, 300, 350, 400)) +
58   theme_bw() +
59   theme(legend.position = "bottom",
60         plot.title = element_text(hjust = 0.5, size = 15),
61         axis.text = element_text(size = 10),
62         axis.title = element_text(size = 12),
63         legend.text = element_text(size = 10)) +
64   labs(title = "CO2 Emissions by Region")
65
66 ggsave("co2_emissions.jpg", height = 6, width = 6)

```



Resources

Model Interface: <https://github.com/JGCRI/gcam-core/releases/tag/gcam-v6.0>

rgcam: <https://github.com/JGCRI/rgcam>

R: <https://www.r-project.org/>

RStudio: <https://posit.co/download/rstudio-desktop/>

ggplot2: <https://ggplot2.tidyverse.org/>