**Python Codes**

#First, install the gegravity package

pip install gegravity

#LOAD THE PACKAGES

import gegravity as ge

import pandas as pd

# Increase number of columns printed for a pandas DataFrame

pd.set\_option("display.max\_columns", None)

pd.set\_option('display.width', 1000)

import gme as gme

#Load the Gravity data

from google.colab import files

uploaded = files.upload()

gravity\_data\_location = "/content/DEBASHREE\_FINALDATASET\_200307.csv"

grav\_data = pd.read\_csv(gravity\_data\_location)

df = pd.read\_csv('/content/DEBASHREE\_FINALDATASET\_200307.csv')

print(grav\_data.head())

#Prepare Data and Econometric inputs for GE model

# Define GME Estimation Data

gme\_data = gme.EstimationData(grav\_data, # Dataset

                              imp\_var\_name="iso\_d", # Importer column name

                              exp\_var\_name="iso\_o", # Exporter column name

                              year\_var\_name = "year",  # Year column name

                              trade\_var\_name="trade\_o")  # Trade column name

# Create Gravity Model

gme\_model = gme.EstimationModel(gme\_data, # Specify data to use

                                lhs\_var="trade\_o",                               # dependent, "left hand side" variable

                                rhs\_var=[ "tariff","contig","comlang\_off","comcol",# independent variables

                                         "lndist", "asean\_FTA","internationall"],

                                fixed\_effects=[["iso\_o"],["iso\_d"]])     # Fixed effects to use

# Estimate gravity model with PPML

gme\_model.estimate()

# Print econometric results table

print(gme\_model.results\_dict['all'].summary())

#After econometric estimation, now we solve the baseline model. We assume India is the reference importer, and take cross section sample for year 2021

# Define GE model

ge\_model = ge.OneSectorGE(gme\_model,                   # gme gravity model

                       year = "2021",               # Year to use for model

                       expend\_var\_name = “gdp\_d",       # Expenditure column name

                       output\_var\_name = “gdp\_o",       # Output column name

                       reference\_importer = "ARE",  # Reference importer

                       sigma = 5)                 # Elasticity of substitution

#The following commands are not required to define or solve the GE model but can help diagnose issues that arise if the model fails to solve.

# Test that the model system of equations is computable from the supplied data and parameters

test\_diagnostics = ge\_model.test\_baseline\_mr\_function()

# See what is returned:

print(test\_diagnostics.keys())

# Check the values of the model parameters computed from the baseline data, which should be numeric with no missing values

input\_params = test\_diagnostics['mr\_params']

# Check one set of parameters, for example:

print(input\_params['cost\_exp\_shr'])

#Check scaling of outward multilateral resistances (OMRs).

# Check for OMR rescale factors that results in convergence

rescale\_eval = ge\_model.check\_omr\_rescale(omr\_rescale\_range=3)

print(rescale\_eval)

#The next set of steps would involve solving baseline and experiment GE model.

# Solve the baseline model

ge\_model.build\_baseline(omr\_rescale=100)

# Examine the solutions for the baseline multilateral resistances

print(ge\_model.baseline\_mr.head(16))

# Now, till here were the codes to solve the baseline gravity model, now we can use the solved baseline model to conduct various counterfactual experiments. As per our research objectives, the counterfactual is removing India from ASEAN and adding Australia into ASEAN. For this purpose, we change asean\_FTA value  with all member countries as 0 and 1 respectively. So, now we define the counterfactual experiment.

# Create a copy of the baseline data

exp\_data = ge\_model.baseline\_data.copy()

#We develop an array of all member countries of ASEAN.

member\_countries = ["BRN", "IDN", “IND”, “KHM”, "LAO", "MYS", "MMR", "PHL", "SGP", "THA", "VNM"]

#For our counterfactual experiment, we set the value of asean\_FTA dummy variable of India with all the member countries as 0. We use 'for' loop for this iterative purpose.

for country in member\_countries:

  exp\_data.loc[(exp\_data["iso\_d"] == "IND") | (exp\_data["iso\_o"] == country), "asean\_FTA"] = 0

  exp\_data.loc[(exp\_data["iso\_d"] == country) | (exp\_data["iso\_o"] == "IND"), "asean\_FTA"] = 0

#For our counterfactual experiment, we set the value of asean\_FTA dummy variable of Australia with all the member countries as 1. We use 'for' loop for this iterative purpose.

for country in member\_countries:

  exp\_data.loc[(exp\_data["iso\_d"] == "AUS") | (exp\_data["iso\_o"] == country), "asean\_FTA"] = 1

  exp\_data.loc[(exp\_data["iso\_d"] == country) | (exp\_data["iso\_o"] == "AUS"), "asean\_FTA"] = 1

# Define the experiment within the GE model

ge\_model.define\_experiment(exp\_data)

# Examine the baseline and counterfactual trade costs

print(ge\_model.bilateral\_costs.head(56))

# Simulate the counterfactual model

ge\_model.simulate()

# We can examine the counterfactual trade flows predicted by the model.

print(ge\_model.bilateral\_trade\_results.head())

#Now that our baseline and counterfactual models are solved, we can access and export the  GE results. We retrieve many of the different sets of model results.

# A collection of many of the key country-level results (prices, total imports/exports, GDP, welfare, etc.)

country\_results = ge\_model.country\_results

print(country\_results)

# The bilateral trade results

bilateral\_results = ge\_model.bilateral\_trade\_results

print(bilateral\_results)

# A wider selection of aggregate, country-level trade results

agg\_trade = ge\_model.aggregate\_trade\_results

print(agg\_trade)

# country multilateral resistance (MR) terms

mr\_terms = ge\_model.country\_mr\_terms

# Get the solver diagnostics, which is a dictionary containing many types of solver diagnostic info

solver\_diagnostics = ge\_model.solver\_diagnostics

mr\_terms = ge\_model.country\_mr\_terms

print(mr\_terms)

solver\_diagnostics = ge\_model.solver\_diagnostics

# Export the results to a collection of spreadsheet (.csv) files and add trade values in levels to the outputs.

from google.colab import drive

drive.mount('/content/drive')

import os

output\_directory = "/content/drive/My Drive/"

os.makedirs(output\_directory, exist\_ok=True)

ge\_model.export\_results(directory=output\_directory, name="GE\_analysis")

**Python Code File**

**India exit –**

<https://colab.research.google.com/drive/1xTf06wcm8xu_2jlYOBPNyaC5jBzb_NHH#scrollTo=JTsggBzBnXZM>

**Australia’s entry-**

[**https://colab.research.google.com/drive/1HyKvXBrFkD5XM0KFA1za9G0xCAsWco0S#scrollTo=WGbxYNSEx8uK**](https://colab.research.google.com/drive/1HyKvXBrFkD5XM0KFA1za9G0xCAsWco0S#scrollTo=WGbxYNSEx8uK)

**Stata Code File**

[**https://drive.google.com/drive/folders/1ZbdN\_uJYz\_E5qsdeP53v8610B1SuHLdO?usp=sharing**](https://drive.google.com/drive/folders/1ZbdN_uJYz_E5qsdeP53v8610B1SuHLdO?usp=sharing)