# Installing Libraries

!pip install pyblp

# Importing libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import pyblp

import seaborn as sns

from google.colab import files

# Upload data

uploaded = files.upload()

# reading data

data = pd.read\_excel('US Luxury SUVs excluding EVs and NGVs.xlsx')

subset\_data = data.copy()

# Rename columns to remove spaces and special characters

subset\_data.columns = subset\_data.columns.str.replace(' ', '\_').str.replace('.', '').str.replace('(', '').str.replace(')', '')

# Display the first few rows of the DataFrame

data.head()

subset\_data.head(11)

# Creating the AWD dummy variable: 1 if AWD or 4WD, 0 otherwise

subset\_data["4WD\_dummy"] = np.where(subset\_data["Drivetrain"].isin(["AWD", "FWD"]), 1, 0)

# Creating the Air Suspension dummy variable: 1 if Yes, 0 if No

subset\_data["Air\_Suspension\_dummy"] = np.where(subset\_data["Air\_Suspension"] == "yes", 1, 0)

# Creating the Panoramic Sunroof dummy variable: 1 if Yes, 0 if No

subset\_data["Panoramic\_Sunroof\_dummy"] = np.where(subset\_data["Panoramic\_Sunroof"] == "yes", 1, 0)

# Select a limited number of variables for the initial estimation

variables = ['MSRP\_USD', 'Horsepower', 'Fuel\_Economy\_combined', 'AWD\_dummy', 'Air\_Suspension\_dummy', 'Panoramic\_Sunroof\_dummy']

# Drop rows with missing values in the selected columns

subset\_data = subset\_data.dropna(subset=variables + ['Sales\_Q1\_2024', 'Market\_Share'])

# Define the product characteristics (X) and the instruments (Z)

# Prepare the data for the BLP model

product\_data\_blp  = subset\_data[["MSRP\_USD", "Horsepower", "Fuel\_Economy\_combined", "AWD\_dummy", "Air\_Suspension\_dummy", "Panoramic\_Sunroof\_dummy"]]

print(product\_data\_blp.dtypes)

print(product\_data\_blp.head())

# Market data

market\_data = {

    'product\_ids': data.index,

    'market\_ids': np.ones(subset\_data.shape[0]),

    'shares': subset\_data['Market\_Share']

}

# Combine the data into the format required by pyblp

# X1: Linear Characteristics

X1\_formulation = pyblp.Formulation ('1 + MSRP\_USD + horsepower + Fuel\_Economy\_combined + AWD\_dummy + Air\_Suspension\_dummy + Panoramic\_Sunroof\_dummy')

# X2: Nonlinear Characteristics (using the same variables here for simplicity)

X2\_formulation = pyblp.Formulation('1 + horsepower + Fuel\_Economy\_combined + AWD\_dummy + Air\_Suspension\_dummy + Panoramic\_Sunroof\_dummy')

# X3: Log Cost Characteristics

X3\_formulation = pyblp.Formulation('1 +log(horsepower) + log(Fuel\_Economy\_combined) + AWD\_dummy')

product\_formulations = (X1\_formulation, X2\_formulation, X3\_formulation)

# Create the problem instance with integration method

mc\_integration = pyblp.Integration('monte\_carlo', size=200, specification\_options={'seed': 0})

problem = pyblp.Problem(product\_formulations, market\_data, product\_data\_blp, integration=mc\_integration)

# Define initial values

initial\_sigma = np.diag([1.0, 1.0, 1.0, 1.0, 1.0, 1.0])

initial\_beta = np.array([1.0, -1.0, 1.0, 1.0, 1.0, 1.0])

initial\_gamma = np.array([1.0, 1.0, 1.0, 1.0, 1.0, 1.0])

# Set up optimization routine

tnc = pyblp.Optimization('tnc', {'maxfun': 500})

# Solve the problem

result = problem.solve(initial\_sigma, initial\_beta, initial\_gamma, tnc)