import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

# Read the data

read = pd.read\_csv("IPO\_data\_2025\_S1.csv", index\_col = False)

# Remove duplicate records and last Unnamed columns

read = read.drop(columns=["Unnamed: 12"])

read = read.drop\_duplicates()

# Remove observations with any missing values (all variables must be non-missing)

read = read.dropna()

# Filter for exchanges: keep only records where the exchange is either Nasdaq or New York

read = read[read['exchange'].isin(['Nasdaq','New York'])]

# Date processing: convert 'ipo\_date' to datetime format and extract the year

read['ipo\_date'] = pd.to\_datetime(read['ipo\_date'],dayfirst = True)

read['ipo\_year'] = read['ipo\_date'].dt.year

# Construct time period dummy variables

# When ipo\_year is <= 2000

read['ipo\_year\_2000'] = (read['ipo\_year'] <= 2000).astype(int)

# When 2001 <= ipo\_year <= 2009

read['ipo\_year\_2009'] = ((read['ipo\_year'] >2000 ) & (read['ipo\_year'] <=2009)).astype(int)

# When ipo\_year is >= 2010

read['ipo\_year\_2020'] = (read['ipo\_year'] >= 2010).astype(int)

# Logarithm variables: log of firm's assets and IPO size (ipo\_amount)

read['log\_assets']= np.log(read['assets'])

read['log\_IPO'] = np.log(read['ipo\_amount'])

# Leverage: debt-to-assets ratio

read['leverage'] = read['debt'] / read['assets']

# ROA: profit-to-assets ratio

read['ROA'] = read['profit']/ read['assets']

# Dummy variable indicating positive profit (profit\_positive)

read['profit\_positive'] = (read['profit'] > 0).astype(int)

# Dummy variable indicating if the firm is from California (rm\_California)

read['rm\_California'] = (read['state'] == 'California').astype(int)

# Save the processed data

read.to\_csv("S1.csv", index = False)

read

#summary statictics table

statictics\_table = ['ipo\_fees', 'assets', 'log\_assets', 'ipo\_amount', 'log\_IPO', 'leverage','ROA','profit\_positive','rm\_California','ipo\_year']

summary\_table = read[statictics\_table].describe()

summary\_table

# Define and estimate Model 1 (using assets as size measure)

model1 = smf.ols(

'ipo\_fees ~ assets + leverage + ROA + rm\_California + 0 + exchange + industry + ipo\_year\_2009 + ipo\_year\_2020 + ipo\_amount + ipo\_price',

data=read

).fit(cov\_type='HC1')

print(model1.summary())

# Define and estimate Model 2 (using log\_assets as size measure)

model2 = smf.ols(

'ipo\_fees ~ log\_assets + leverage + ROA + rm\_California + 0 + exchange + industry + ipo\_year\_2009 + ipo\_year\_2020 + ipo\_amount + ipo\_price',

data=read

).fit(cov\_type='HC1')

print(model2.summary())

# Define and estimate Model 3 (using ipo\_amount as size measure)

model3 = smf.ols(

'ipo\_fees ~ ipo\_amount + leverage + ROA + rm\_California + 0 + exchange + industry + ipo\_year\_2009 + ipo\_year\_2020',

data=read

).fit(cov\_type='HC1')

print(model3.summary())

# Make predictions using Model 1

new\_obs\_main = pd.DataFrame({

'assets': [500],

'leverage': [0.1],

'ROA': [0.05],

'ipo\_amount': [200],

'rm\_California': [1],

'exchange': ['New York'],

'industry': ['IT'],

'ipo\_price':[30],

'ipo\_year\_2009': [0],

'ipo\_year\_2020': [0]

})

pred\_main = model1.predict(new\_obs\_main)

print("Main Estimate:", pred\_main.iloc[0])

# Change the predictions ROA and ipo\_price

new\_obs\_main = pd.DataFrame({

'assets': [500],

'leverage': [0.1],

'ROA': [-0.1],

'ipo\_amount': [200],

'rm\_California': [1],

'exchange': ['New York'],

'industry': ['IT'],

'ipo\_price':[300],

'ipo\_year\_2009': [0],

'ipo\_year\_2020': [0]

})

pred\_main = model1.predict(new\_obs\_main)

print("Main Estimate:", pred\_main.iloc[0])