import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data\_path = r'Filtered Dams (UN McGill) 2010-2022 Africa.csv'

data\_df = pd.read\_csv(data\_path)

data\_df

data\_df\_area\_filtered = data\_df[~data\_df['COUNTRY'].isin(['Algeria', 'Djibouti', 'Egypt', 'Libya', 'Morocco', 'Somalia', 'Sudan', 'Tunisia'])]

data\_df\_area\_filtered.head(5)

data\_df\_year\_filtered = data\_df\_area\_filtered[data\_df\_area\_filtered['YEAR\_DAM'].isin([2014, 2015, 2016, 2017])]

data\_df\_year\_filtered.head(5)

data\_df\_result = data\_df\_year\_filtered

data\_df\_result

data\_df\_year\_filtered.to\_csv("data\_filtered\_result.csv", index=False, sep=',')

#Read overall data

df = pd.read\_csv("subnational\_unit\_data.csv")

df\_cases = df[df['Metric'] == 'Incidence Rate']

df\_cases.head(5)

# Create and process all data for Damid 6909

df\_filtered\_6909 = df\_cases[(df\_cases['Name'] == 'Oromia') | (df\_cases['Name'] == 'Dire Dawa')]

# df\_filtered = df\_filtered\_6909[(df\_filtered\_6909['Year'] >= 2013) & (df\_filtered\_6909['Year'] <= 2018)]

df\_filtered\_6909['Treatment'] = df\_filtered\_6909['Name'].apply(lambda x: 1 if x == 'Oromia' else 0)

df\_filtered\_6909['Post'] = df\_filtered\_6909['Year'].apply(lambda x: 1 if x >= 2016 else 0)

df\_filtered\_6909['Treatment\_Post'] = df\_filtered\_6909['Treatment'] \* df\_filtered\_6909['Post']

save\_path\_6909 = './df\_filtered\_6909.csv'

df\_filtered\_6909.to\_csv(save\_path\_6909, index=False)

print(f"The file has been saved to: {save\_path\_6909}")

#Baseline Mean, SD and T-test Damid 6909

import scipy.stats as stats

pre\_data\_6909 = df\_filtered\_6909[df\_filtered\_6909['Year'] <= 2015]

mean\_Oromia = pre\_data\_6909[pre\_data\_6909['Name'] == 'Oromia']['Value'].mean()

mean\_Dire = pre\_data\_6909[pre\_data\_6909['Name'] == 'Dire Dawa']['Value'].mean()

std\_Oromia = pre\_data\_6909[pre\_data\_6909['Name'] == 'Oromia']['Value'].std()

std\_Dire = pre\_data\_6909[pre\_data\_6909['Name'] == 'Dire Dawa']['Value'].std()

print(f"Oromia Baseline Mean: {mean\_Oromia:.2f}, SD: {std\_Oromia:.2f}")

print(f"Dire Dawa Baseline Mean: {mean\_Dire:.2f}, SD: {std\_Dire:.2f}")

t\_stat, p\_value = stats.ttest\_ind(

pre\_data\_6909[pre\_data\_6909['Name'] == 'Oromia']['Value'],

pre\_data\_6909[pre\_data\_6909['Name'] == 'Dire Dawa']['Value']

)

print(f"t-Statistic: {t\_stat:.2f}, p value: {p\_value:.3f}")

#DID Damid 6909:

import statsmodels.formula.api as smf

from statsmodels.iolib.summary2 import summary\_col

model\_6909 = smf.ols('Value ~ Treatment + Post + Treatment\_Post', data=df\_filtered\_6909).fit()

print(model\_6909.summary())

# Visualise Damid 6909

import seaborn as sns

import matplotlib.pyplot as plt

sns.lineplot(data=df\_filtered\_6909, x='Year', y='Value', hue='Name', marker='')

plt.axvline(2016, color='gray', linestyle='--', label='Oromia Dam Construction')

plt.title('Malaria Incidence Rates in Oromia and Dire Dawa')

plt.xlabel('Year')

plt.ylabel('Malaria Incidence Rate (Cases per Thousand)')

plt.legend(title='')

plt.show()

# model table summary Damid 6909

models=[] # create empty list to store the models

names=[] # create empty list to store the names of the models

for year\_diff in range(1, 10): #

did=df\_cases[(df\_cases['Year'] >= (2016-year\_diff)) & (df\_cases['Year']<=(2016+year\_diff)) \

& df\_cases['Name'].isin(['Oromia', 'Dire Dawa'])] # subset the data within the window of interest around 2016

did['post']=np.where(did['Year']>=2016,1,0) # create a dummy variable indicating the period after the minimum wage increase

did['treatment']=np.where(did['Name']=='Oromia',1,0) # create a dummy variable for treatment

did['post\_treatment']=did['post']\*did['treatment'] # create an interaction term between the post and treatment variables

did\_model = smf.ols('Value ~ post + treatment + post\_treatment', did).fit() # run the difference in difference model

models.append(did\_model) # append the model to the list of models

names.append('± '+str(year\_diff)+' Year') # append the name of the model to the list of names

table= summary\_col( # create a regression table

models, # pass the models to the summary\_col function

stars=True, # add stars denoting the p-values of the coefficient to the table; \* p<0.05, \*\* p<0.01, \*\*\* p<0.001

float\_format='%0.3f', # set the decimal places to 3

model\_names=names, # set the names of the model

info\_dict = {"N":lambda x: "{0:d}".format(int(x.nobs))}) # add the number of observations to the table

print(table) # print the table

df\_filtered\_6909 = df\_filtered\_6909[df\_filtered\_6909['Name'] == 'Oromia']

df\_filtered\_6909["Year"] -= 2016

df\_filtered\_6909.head(100)

rdd\_df = df\_filtered\_6909.assign(threshold=(df\_filtered\_6909["Year"] > 0).astype(int))

model = smf.wls("Value ~ Year \* threshold ", rdd\_df).fit()

print(model.summary())

plt.figure(figsize=(8,8))

ax = plt.subplot(3,1,1)

df\_filtered\_6909.plot.scatter(x="Year", y="Value", ax=ax)

plt.title("Incidence Before/After Dam Construction (Centered at 0)");

ate\_pct = 100\*((model.params["threshold"] + model.params["Intercept"])/model.params["Intercept"] - 1)

ax = df\_filtered\_6909.plot.scatter(x="Year", y="Malaria Incidence Rate (per 1,000)", color="C0")

df\_filtered\_6909.assign(predictions=model.fittedvalues).plot(x="Year", y="predictions", ax=ax, color="C1")

plt.title(f"Impact of Dams on Malaria Incidence: {np.round(ate\_pct, 2)}% \n p={np.round(model.pvalues['threshold'], 3)}, R2={np.round(model.rsquared, 3)}")

plt.show()

# Create and process all data for Damid 6910

df\_filtered\_6910 = df\_cases[(df\_cases['Name'] == 'Oromia') | (df\_cases['Name'] == 'Harari')]

# df\_filtered = df\_filtered\_6910[(df\_filtered\_6909['Year'] >= 2013) & (df\_filtered\_6909['Year'] <= 2018)]

df\_filtered\_6910['Treatment'] = df\_filtered\_6910['Name'].apply(lambda x: 1 if x == 'Oromia' else 0)

df\_filtered\_6910['Post'] = df\_filtered\_6910['Year'].apply(lambda x: 1 if x >= 2015 else 0)

df\_filtered\_6910['Treatment\_Post'] = df\_filtered\_6910['Treatment'] \* df\_filtered\_6910['Post']

df\_filtered\_6910\_path = './df\_filtered\_6910.csv'

df\_filtered\_6910.to\_csv(df\_filtered\_6910\_path, index=False)

print(f"The file has been saved to: {df\_filtered\_6910\_path}")

#DID Damid 6910:

import statsmodels.formula.api as smf

model\_6910 = smf.ols('Value ~ Treatment + Post + Treatment\_Post', data=df\_filtered\_6910).fit()

print(model\_6910.summary())

# Visualise Damid 6910

import seaborn as sns

import matplotlib.pyplot as plt

sns.lineplot(data=df\_filtered\_6910, x='Year', y='Value', hue='Name', marker='')

plt.axvline(2015, color='gray', linestyle='--', label='Oromia Dam Construction')

plt.title('Malaria Incidence Rates in Oromia and Harari')

plt.xlabel('Year')

plt.ylabel('Malaria Incidence Rate (Cases per Thousand)')

plt.legend(title='')

plt.show()

# model table summary Damid 6910

models=[] # create empty list to store the models

names=[] # create empty list to store the names of the models

for year\_diff in range(1, 10):

did=df\_cases[(df\_cases['Year'] >= (2015-year\_diff)) & (df\_cases['Year']<=(2015+year\_diff)) \

& df\_cases['Name'].isin(['Oromia', 'Harari'])] # subset the data within the window of interest around 2015

did['post']=np.where(did['Year']>=2015,1,0) # create a dummy variable indicating the period after the minimum wage increase

did['treatment']=np.where(did['Name']=='Oromia',1,0) # create a dummy variable for treatment

did['post\_treatment']=did['post']\*did['treatment'] # create an interaction term between the post and treatment variables

did\_model = smf.ols('Value ~ post + treatment + post\_treatment', did).fit() # run the difference in difference model

models.append(did\_model) # append the model to the list of models

names.append('± '+str(year\_diff)+' Year') # append the name of the model to the list of names

table= summary\_col( # create a regression table

models, # pass the models to the summary\_col function

stars=True, # add stars denoting the p-values of the coefficient to the table; \* p<0.05, \*\* p<0.01, \*\*\* p<0.001

float\_format='%0.3f', # set the decimal places to 3

model\_names=names, # set the names of the model

info\_dict = {"N":lambda x: "{0:d}".format(int(x.nobs))}) # add the number of observations to the table

print(table) # print the table

*df\_filtered\_6910 = df\_filtered\_6910[df\_filtered\_6910['Name'] == 'Oromia']*

*df\_filtered\_6910["Year"] -= 2015*

*df\_filtered\_6910.head()*

*plt.figure(figsize=(8,8))*

*ax = plt.subplot(3,1,1)*

*df\_filtered\_6910.plot.scatter(x="Year", y="Value", ax=ax)*

*plt.title("Incidence Before/After Dam Construction (Centered at 0)");*

*rdd\_df = df\_filtered\_6910.assign(threshold=(df\_filtered\_6910["Year"] > 0).astype(int))*

*model = smf.wls("Value ~ Year \* threshold ", rdd\_df).fit()*

*print(model.summary())*

*ate\_pct = 100\*((model.params["threshold"] + model.params["Intercept"])/model.params["Intercept"] - 1)*

*ax = df\_filtered\_6910.plot.scatter(x="Year", y="Value", color="C0")*

*df\_filtered\_6910.assign(predictions=model.fittedvalues).plot(x="Year", y="predictions", ax=ax, color="C1")*

*plt.title(f"Impact of Dams on Malaria Incidence: {np.round(ate\_pct, 2)}% \n p={np.round(model.pvalues['threshold'], 3)}, R2={np.round(model.rsquared, 3)}")*

*plt.show()*

# Create and process all data for Damid 7141

df\_filtered\_7141 = df\_cases[(df\_cases['Name'] == 'Masvingo') | (df\_cases['Name'] == 'Midlands')]

# df\_filtered = df\_filtered\_6910[(df\_filtered\_7141['Year'] >= 2013) & (df\_filtered\_6909['Year'] <= 2018)]

df\_filtered\_7141['Treatment'] = df\_filtered\_7141['Name'].apply(lambda x: 1 if x == 'Masvingo' else 0)

df\_filtered\_7141['Post'] = df\_filtered\_7141['Year'].apply(lambda x: 1 if x >= 2015 else 0)

df\_filtered\_7141['Treatment\_Post'] = df\_filtered\_7141['Treatment'] \* df\_filtered\_7141['Post']

df\_filtered\_7141\_path = './df\_filtered\_7141.csv'

df\_filtered\_7141.to\_csv(df\_filtered\_7141\_path, index=False)

print(f"The file has been saved to: {df\_filtered\_7141\_path}")

#Baseline Mean, SD and T-test Damid 7141

import scipy.stats as stats

pre\_data\_7141 = df\_filtered\_7141[df\_filtered\_7141['Year'] <= 2014]

mean\_Masvingo = pre\_data\_7141[pre\_data\_7141['Name'] == 'Masvingo']['Value'].mean()

mean\_Midlands = pre\_data\_7141[pre\_data\_7141['Name'] == 'Midlands']['Value'].mean()

std\_Masvingo = pre\_data\_7141[pre\_data\_7141['Name'] == 'Masvingo']['Value'].std()

std\_Midlands = pre\_data\_7141[pre\_data\_7141['Name'] == 'Midlands']['Value'].std()

print(f"Masvingo Baseline Mean: {mean\_Masvingo:.2f}, SD: {std\_Masvingo:.2f}")

print(f"Midlands Baseline Mean: {mean\_Midlands:.2f}, SD: {std\_Midlands:.2f}")

t\_stat, p\_value = stats.ttest\_ind(

pre\_data\_7141[pre\_data\_7141['Name'] == 'Masvingo']['Value'],

pre\_data\_7141[pre\_data\_7141['Name'] == 'Midlands']['Value']

)

print(f"t-Statistic: {t\_stat:.2f}, p-value: {p\_value:.3f}")

#DID Damid 7141:

import statsmodels.formula.api as smf

model\_7141 = smf.ols('Value ~ Treatment + Post + Treatment\_Post', data=df\_filtered\_7141).fit()

print(model\_7141.summary())

# Visualise Damid 7141

import seaborn as sns

import matplotlib.pyplot as plt

sns.lineplot(data=df\_filtered\_7141, x='Year', y='Value', hue='Name', marker='')

plt.axvline(2015, color='gray', linestyle='--', label='Masvingo Dam Construction')

plt.title('Malaria Incidence Rates in Masvingo and Midlands')

plt.xlabel('Year')

plt.ylabel('Malaria Incidence Rate (Cases per Thousand)')

plt.legend(title='')

plt.show()

# model table summary Damid 7141

models=[] # create empty list to store the models

names=[] # create empty list to store the names of the models

for year\_diff in range(1, 10):

did=df\_cases[(df\_cases['Year'] >= (2015-year\_diff)) & (df\_cases['Year']<=(2015+year\_diff))\

& df\_cases['Name'].isin(['Masvingo', 'Midlands'])] # subset the data within the window of interest around 2015

did['post']=np.where(did['Year']>=2015,1,0) # create a dummy variable indicating the period after the minimum wage increase

did['treatment']=np.where(did['Name']=='Masvingo',1,0) # create a dummy variable for treatment

did['post\_treatment']=did['post']\*did['treatment'] # create an interaction term between the post and treatment variables

did\_model = smf.ols('Value ~ post + treatment + post\_treatment', did).fit() # run the difference in difference model

models.append(did\_model) # append the model to the list of models

names.append('± '+str(year\_diff)+' Year') # append the name of the model to the list of names

table= summary\_col( # create a regression table

models, # pass the models to the summary\_col function

stars=True, # add stars denoting the p-values of the coefficient to the table; \* p<0.05, \*\* p<0.01, \*\*\* p<0.001

float\_format='%0.3f', # set the decimal places to 3

model\_names=names, # set the names of the model

info\_dict = {"N":lambda x: "{0:d}".format(int(x.nobs))}) # add the number of observations to the table

print(table) # print the table

df\_filtered\_7141 = df\_filtered\_7141[df\_filtered\_7141['Name'] == 'Masvingo']

df\_filtered\_7141["Year"] -= 2015

df\_filtered\_7141.head(100)

*plt.figure(figsize=(8,8))*

*ax = plt.subplot(3,1,1)*

*df\_filtered\_7141.plot.scatter(x="Year", y="Value", ax=ax)*

*plt.title("Incidence Before/After Dam Construction (Centered at 0)");*

*rdd\_df = df\_filtered\_7141.assign(threshold=(df\_filtered\_7141["Year"] > 0).astype(int))*

*model = smf.wls("Value ~ Year \* threshold ", rdd\_df).fit()*

*print(model.summary())*

*ate\_pct = 100\*((model.params["threshold"] + model.params["Intercept"])/model.params["Intercept"] - 1)*

*ax = df\_filtered\_7141.plot.scatter(x="Year", y="Value", color="C0")*

*df\_filtered\_7141.assign(predictions=model.fittedvalues).plot(x="Year", y="predictions", ax=ax, color="C1")*

*plt.title(f"Impact of Dams on Malaria Incidence: {np.round(ate\_pct, 2)}% \n p={np.round(model.pvalues['threshold'], 3)}, R2={np.round(model.rsquared, 3)}")*

*plt.show()*

# Create and process all data for Damid 40188

df\_filtered\_40188 = df\_cases[(df\_cases['Name'] == 'Debub') | (df\_cases['Name'] == 'Semenawi Keih Bahri')]

# df\_filtered = df\_filtered\_6910[(df\_filtered\_7141['Year'] >= 2013) & (df\_filtered\_6909['Year'] <= 2018)]

df\_filtered\_40188['Treatment'] = df\_filtered\_40188['Name'].apply(lambda x: 1 if x == 'Debub' else 0)

df\_filtered\_40188['Post'] = df\_filtered\_40188['Year'].apply(lambda x: 1 if x >= 2015 else 0)

df\_filtered\_40188['Treatment\_Post'] = df\_filtered\_40188['Treatment'] \* df\_filtered\_40188['Post']

df\_filtered\_40188\_path = './df\_filtered\_40188.csv'

df\_filtered\_40188.to\_csv(df\_filtered\_40188\_path, index=False)

print(f"The file has been saved to: {df\_filtered\_40188\_path}")

#Baseline Mean, SD and T-test Damid 40188

import scipy.stats as stats

pre\_data\_40188 = df\_filtered\_40188[df\_filtered\_40188['Year'] <= 2014]

mean\_Debub = pre\_data\_40188[pre\_data\_40188['Name'] == 'Debub']['Value'].mean()

mean\_Semenawi = pre\_data\_40188[pre\_data\_40188['Name'] == 'Semenawi Keih Bahri']['Value'].mean()

std\_Debub = pre\_data\_40188[pre\_data\_40188['Name'] == 'Debub']['Value'].std()

std\_Semenawi = pre\_data\_40188[pre\_data\_40188['Name'] == 'Semenawi Keih Bahri']['Value'].std()

print(f"Debub Baseline Mean: {mean\_Debub:.2f}, SD: {std\_Debub:.2f}")

print(f"Semenawi Keih Bahri Baseline Mean: {mean\_Semenawi:.2f}, SD: {std\_Semenawi:.2f}")

t\_stat, p\_value = stats.ttest\_ind(

pre\_data\_40188[pre\_data\_40188['Name'] == 'Debub']['Value'],

pre\_data\_40188[pre\_data\_40188['Name'] == 'Semenawi Keih Bahri']['Value']

)

print(f"t-Statistic: {t\_stat:.2f}, p-value: {p\_value:.3f}")

#DID Damid 40188:

import statsmodels.formula.api as smf

model\_40188 = smf.ols('Value ~ Treatment + Post + Treatment\_Post', data=df\_filtered\_40188).fit()

print(model\_40188.summary())

# Visualise Damid 40188

import seaborn as sns

import matplotlib.pyplot as plt

sns.lineplot(data=df\_filtered\_40188, x='Year', y='Value', hue='Name', marker='')

plt.axvline(2015, color='gray', linestyle='--', label='Debub Dam Construction')

plt.title('Malaria Incidence Rates in Debub and Semenawi Keih Bahri')

plt.xlabel('Year')

plt.ylabel('Malaria Incidence Rate (Cases per Thousand)')

plt.legend(title='')

plt.show()

# model table summary Damid 40188

models=[] # create empty list to store the models

names=[] # create empty list to store the names of the models

for year\_diff in range(1,10):

did=df\_cases[(df\_cases['Year'] >= (2015-year\_diff)) & (df\_cases['Year']<=(2015+year\_diff))\

& df\_cases['Name'].isin(['Debub', 'Semenawi Keih Bahri'])] # subset the data within the window of interest around 2015

did['post']=np.where(did['Year']>=2015,1,0) # create a dummy variable indicating the period after the minimum wage increase

did['treatment']=np.where(did['Name']=='Debub',1,0) # create a dummy variable for treatment

did['post\_treatment']=did['post']\*did['treatment'] # create an interaction term between the post and treatment variables

did\_model = smf.ols('Value ~ post + treatment + post\_treatment', did).fit() # run the difference in difference model

models.append(did\_model) # append the model to the list of models

names.append('± '+str(year\_diff)+' Year') # append the name of the model to the list of names

table= summary\_col( # create a regression table

models, # pass the models to the summary\_col function

stars=True, # add stars denoting the p-values of the coefficient to the table; \* p<0.05, \*\* p<0.01, \*\*\* p<0.001

float\_format='%0.3f', # set the decimal places to 3

model\_names=names, # set the names of the model

info\_dict = {"N":lambda x: "{0:d}".format(int(x.nobs))}) # add the number of observations to the table

print(table) # print the table

df\_filtered\_40188 = df\_filtered\_40188[df\_filtered\_40188['Name'] == 'Debub']

df\_filtered\_40188["Year"] -= 2015

df\_filtered\_40188.head(100)

plt.figure(figsize=(8,8))

ax = plt.subplot(3,1,1)

df\_filtered\_40188.plot.scatter(x="Year", y="Value", ax=ax)

plt.title("Incidence Before/After Dam Construction (Centered at 0)");

rdd\_df = df\_filtered\_40188.assign(threshold=(df\_filtered\_40188["Year"] > 0).astype(int))

model = smf.wls("Value ~ Year \* threshold ", rdd\_df).fit()

print(model.summary())

ate\_pct = 100\*((model.params["threshold"] + model.params["Intercept"])/model.params["Intercept"] - 1)

ax = df\_filtered\_40188.plot.scatter(x="Year", y="Value", color="C0")

df\_filtered\_40188.assign(predictions=model.fittedvalues).plot(x="Year", y="predictions", ax=ax, color="C1")

plt.title(f"Impact of Dams on Malaria Incidence: {np.round(ate\_pct, 2)}% \n p={np.round(model.pvalues['threshold'], 3)}, R2={np.round(model.rsquared, 3)}")

plt.show()