In [1]:
```
import dame_flame
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import heapq
from sklearn.model_selection import train_test_split
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.preprocessing import LabelBinarizer
from sklearn.linear_model import LinearRegression
from scipy import stats
from statsmodels.distributions.empirical_distribution import ECDF
```

```
C:\Users\Neha\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19: F
utureWarning: pandas.util.testing is deprecated. Use the functions in the pub
lic API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [2]:
```
STAR_Students = pd.read_spss('STAR_Students.sav')
```

In [3]:
```
#STAR_Students['gksurban'].value_counts()
# d = {"RURAL": 0, "URBAN":1, "SUBURBAN": 2, "INNER CITY": 3}

df_trunc = STAR_Students.loc[:, STAR_Students.columns.intersection(
    ['gkclasstype', 'gender', 'race', 'gkfreelunch', 'gkschid', 'gktmathss',
'gktreadss', 'g1freelunch', 'g2freelunch', 'g3freelunch'])]


d = {"WHITE": 1, "BLACK": 0, "ASIAN": 1, "HISPANIC": 0, "OTHER": 0,
     "NATIVE AMERICAN": 0}
df_trunc['race'] = df_trunc['race'].map(d)

d = {"NON-FREE LUNCH": 0, "FREE LUNCH": 1}
df_trunc['gkfreelunch'] = df_trunc['gkfreelunch'].map(d)
df_trunc['g1freelunch'] = df_trunc['g1freelunch'].map(d)
df_trunc['g2freelunch'] = df_trunc['g2freelunch'].map(d)
df_trunc['g3freelunch'] = df_trunc['g3freelunch'].map(d)


d = {"MALE": 1, "FEMALE": 0}
df_trunc['gender'] = df_trunc['gender'].map(d)
#df_trunc['gktgen'] = df_trunc['gktgen'].map(d)

d = {"WHITE": 1, "BLACK": 0}
#df_trunc['gktrace'] = df_trunc['gktrace'].map(d)

d = {"SMALL CLASS": int(1), "REGULAR CLASS": int(0),
     "REGULAR + AIDE CLASS": int(0)}
df_trunc['ksmall'] = df_trunc['gkclasstype'].map(d)

# df_trunc = df_trunc.dropna().copy()

# Create age variable counting months
#df_trunc['age'] = df_trunc['birthyear']*12 + df_trunc['birthmonth']
df_trunc = df_trunc.drop(columns=['gkclasstype'])
# Bin age into deciles
#df_trunc['age'] = pd.qcut(df_trunc['age'], q=10, labels=False)

df_trunc = df_trunc.rename(columns={"ksmall": "treated"}) ## NOTE TO SELF -- C
OME BACK TO WE SHOULDNT HAVE TO DO THIS
```

In [4]:
```
for i in df_trunc.index:
    if df_trunc.loc[i, 'g1freelunch'] == 1 or df_trunc.loc[i, 'g2freelunch'] =
= 1 or df_trunc.loc[i, 'g3freelunch'] == 1 or df_trunc.loc[i, 'gkfreelunch'] =
= 1:
        df_trunc.loc[i, 'gkfreelunch'] = 1
    else:
        df_trunc.loc[i, 'gkfreelunch'] = 0
df_trunc = df_trunc.drop(columns=['g1freelunch', 'g2freelunch', 'g3freelunc
h'])

# df_trunc=df_trunc.dropna(subset=['treated'])
df_trunc = df_trunc.dropna() # COME BACK TO!
```

In [6]:
```
ecdf_reading = ECDF(df_trunc[df_trunc['treated'] == 0]['gktreadss'])
ecdf_math = ECDF(df_trunc[df_trunc['treated'] == 0]['gktmathss'])
df_trunc['read_outcome'] = ecdf_reading(df_trunc['gktreadss'])*100
df_trunc['math_outcome'] = ecdf_math(df_trunc['gktmathss'])*100
df_trunc['outcome'] = (df_trunc['read_outcome'] + df_trunc['math_outcome'])/2
```

In [7]:
```
'''
math_array = df_trunc[df_trunc['treated'] == 0]['gktmathss']
reading_array = df_trunc[df_trunc['treated'] == 0]['gktreadss']
#df_trunc['read_outcome'] = stats.percentileofscore(reading_array, df_trunc['g
ktreadss'])
#df_trunc['math_outcome'] = stats.percentileofscore(math_array, df_trunc['gktm
athss'])
for i in df_trunc.index:
    df_trunc.loc[i, 'math_outcome'] = stats.percentileofscore(a=math_array, sc
ore=df_trunc.loc[i,'gktmathss'])
    df_trunc.loc[i, 'read_outcome'] = stats.percentileofscore(a=reading_array,
score=df_trunc.loc[i,'gktreadss'])


df_trunc['outcome'] = (df_trunc['read_outcome'] + df_trunc['math_outcome'])/2
'''

'''
df_trunc_untreated = df_trunc[df_trunc['treated'] == 0]
df_trunc_treated = df_trunc[df_trunc['treated'] == 1]
for i in df_trunc_treated.index:
    df_trunc_treated.loc[i, 'gktreadss'] = stats.percentileofscore(df_trunc_un
treated['gktreadss'], df_trunc_treated.loc[i,'gktreadss'])
    df_trunc_treated.loc[i, 'gktmathss'] = stats.percentileofscore(df_trunc_un
treated['gktmathss'], df_trunc_treated.loc[i,'gktmathss'])

# Percentile the math and reading and then average them
# but do different percentiles for the small class size people and the large p
eople.
df_trunc_untreated = df_trunc[df_trunc['treated'] == 0]
df_trunc_untreated['gktreadss'] = df_trunc_untreated['gktreadss'].rank(pct=Tru
e)*100
df_trunc_untreated['gktmathss'] = df_trunc_untreated['gktmathss'].rank(pct=Tru
e)*100
df_trunc_untreated['outcome'] = df_trunc_untreated[['gktreadss', 'gktmaths
s']].mean(axis=1)

#df_trunc_treated = df_trunc[df_trunc['treated'] == 1]
#df_trunc_treated['gktreadss'] = df_trunc_treated['gktreadss'].rank(pct=True)*
100
#df_trunc_treated['gktmathss'] = df_trunc_treated['gktmathss'].rank(pct=True)*
100
df_trunc_treated['outcome'] = df_trunc_treated[['gktreadss', 'gktmathss']].mea
n(axis=1)

df = pd.concat([df_trunc_treated, df_trunc_untreated])
df = df.drop(columns=['gktreadss', 'gktmathss'])

df = df.dropna()
'''
```

Out[7]:
```
"\ndf_trunc_untreated = df_trunc[df_trunc['treated'] == 0]\ndf_trunc_treated
= df_trunc[df_trunc['treated'] == 1]\nfor i in df_trunc_treated.index:\n    d
f_trunc_treated.loc[i, 'gktreadss'] = stats.percentileofscore(df_trunc_untrea
ted['gktreadss'], df_trunc_treated.loc[i,'gktreadss'])\n    df_trunc_treated.
loc[i, 'gktmathss'] = stats.percentileofscore(df_trunc_untreated['gktmaths
s'], df_trunc_treated.loc[i,'gktmathss'])\n\n# Percentile the math and readin
g and then average them\n# but do different percentiles for the small class s
ize people and the large people.\ndf_trunc_untreated = df_trunc[df_trunc['tre
ated'] == 0]\ndf_trunc_untreated['gktreadss'] = df_trunc_untreated['gktreads
s'].rank(pct=True)*100\ndf_trunc_untreated['gktmathss'] = df_trunc_untreated
['gktmathss'].rank(pct=True)*100\ndf_trunc_untreated['outcome'] = df_trunc_un
treated[['gktreadss', 'gktmathss']].mean(axis=1)\n\n#df_trunc_treated = df_tr
unc[df_trunc['treated'] == 1]\n#df_trunc_treated['gktreadss'] = df_trunc_trea
ted['gktreadss'].rank(pct=True)*100\n#df_trunc_treated['gktmathss'] = df_trun
c_treated['gktmathss'].rank(pct=True)*100\ndf_trunc_treated['outcome'] = df_t
runc_treated[['gktreadss', 'gktmathss']].mean(axis=1)\n\ndf = pd.concat([df_t
runc_treated, df_trunc_untreated])\ndf = df.drop(columns=['gktreadss', 'gktma
thss'])\n\ndf = df.dropna()\n"
```

In [8]:
```
md = sm.OLS(df_trunc['outcome'], sm.add_constant(df_trunc['treated']))
md.fit().summary()
```

Out[8]:

OLS Regression Results

Dep. Variable:	outcome	**R-squared:**	0.007
Model:	OLS	**Adj. R-squared:**	0.007
Method:	Least Squares	**F-statistic:**	41.30
Date:	Mon, 13 Feb 2023	**Prob (F-statistic):**	1.41e-10
Time:	21:52:23	**Log-Likelihood:**	-27274.
No. Observations:	5785	**AIC:**	5.455e+04
Df Residuals:	5783	**BIC:**	5.456e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	51.3466	0.424	120.985	0.000	50.515	52.179
treated	4.9761	0.774	6.427	0.000	3.458	6.494

Omnibus:	2194.401	**Durbin-Watson:**	1.805
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	298.389
Skew:	-0.115	**Prob(JB):**	1.61e-65
Kurtosis:	1.912	**Cond. No.**	2.42

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [9]:
```
md = smf.mixedlm(formula="outcome ~ treated", data=df_trunc , groups=df_trunc
['gkschid'])
md.fit().summary()
```

Out[9]:

Model:	MixedLM	Dependent Variable:	outcome
No. Observations:	5785	Method:	REML
No. Groups:	79	Scale:	556.8234
Min. group size:	34	Likelihood:	-26616.4107
Max. group size:	137	Converged:	Yes
Mean group size:	73.2		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Intercept	51.128	1.534	33.330	0.000	48.121	54.134
treated	5.481	0.685	8.002	0.000	4.139	6.824
Group Var	174.234	1.247				

In [10]:
```
md = smf.mixedlm(formula="outcome ~ gender+race+gkfreelunch+treated", data=df_
trunc, groups=df_trunc['gkschid'])
md.fit().summary()
```

Out[10]:

Model:	MixedLM	Dependent Variable:	outcome
No. Observations:	5785	Method:	REML
No. Groups:	79	Scale:	514.3666
Min. group size:	34	Likelihood:	-26380.6882
Max. group size:	137	Converged:	Yes
Mean group size:	73.2		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Intercept	41.435	1.679	24.676	0.000	38.144	44.726
gender[T.1]	-4.786	0.601	-7.969	0.000	-5.963	-3.609
gkfreelunch[T.0]	12.374	0.721	17.160	0.000	10.961	13.788
race	9.435	1.206	7.826	0.000	7.072	11.798
treated	5.487	0.658	8.334	0.000	4.197	6.778
Group Var	149.240	1.118				

In [11]:
```
fes = pd.get_dummies(df_trunc['gkschid'])
fes = fes.drop(columns=[161183.0])
y = df_trunc.loc[:,['outcome']]
x = df_trunc.loc[:, ['gender', 'race', 'gkfreelunch', 'treated']]
x = pd.concat([fes,x],axis=1)
x = sm.add_constant(x)
model = sm.OLS(y,x)
model.fit().summary()
```

Out[11]:

OLS Regression Results

Dep. Variable:	outcome	**R-squared:**	0.309
Model:	OLS	**Adj. R-squared:**	0.299
Method:	Least Squares	**F-statistic:**	31.12
Date:	Mon, 13 Feb 2023	**Prob (F-statistic):**	0.00
Time:	21:52:25	**Log-Likelihood:**	-26224.
No. Observations:	5785	**AIC:**	5.261e+04
Df Residuals:	5702	**BIC:**	5.317e+04
Df Model:	82		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	61.9599	2.620	23.647	0.000	56.823	67.096
112038.0	-31.3536	3.867	-8.108	0.000	-38.935	-23.773
123056.0	-18.0925	3.860	-4.687	0.000	-25.660	-10.525
128068.0	-19.9081	3.787	-5.257	0.000	-27.332	-12.484
128076.0	-27.0678	3.708	-7.299	0.000	-34.338	-19.798
128079.0	-27.7960	3.681	-7.552	0.000	-35.011	-20.581
130085.0	-18.6904	3.491	-5.355	0.000	-25.533	-11.848
159171.0	8.1353	3.204	2.539	0.011	1.855	14.416
161176.0	-18.0216	3.383	-5.328	0.000	-24.653	-11.390
162184.0	-14.2478	3.662	-3.891	0.000	-21.426	-7.069
164198.0	-9.0715	3.790	-2.394	0.017	-16.501	-1.642
165199.0	3.4669	3.870	0.896	0.370	-4.120	11.054
166203.0	-19.8036	3.554	-5.572	0.000	-26.771	-12.836
168211.0	-10.2954	3.207	-3.211	0.001	-16.582	-4.009
168214.0	2.0891	3.771	0.554	0.580	-5.303	9.481
169219.0	1.1489	3.960	0.290	0.772	-6.615	8.912
169229.0	-2.4346	2.950	-0.825	0.409	-8.217	3.348
169231.0	-28.2711	3.844	-7.355	0.000	-35.807	-20.735
169280.0	-4.6868	3.825	-1.225	0.220	-12.185	2.811
170295.0	-4.3775	3.586	-1.221	0.222	-11.408	2.653
173312.0	12.2710	3.748	3.274	0.001	4.924	19.618
176329.0	3.6933	3.554	1.039	0.299	-3.274	10.661
180344.0	-8.5509	3.256	-2.626	0.009	-14.933	-2.169
189378.0	-23.7646	3.511	-6.768	0.000	-30.648	-16.881
189382.0	-11.6319	3.620	-3.213	0.001	-18.729	-4.535

189396.0	-22.3242	3.659	-6.101	0.000	-29.497	-15.151
191411.0	-4.7020	4.065	-1.157	0.247	-12.672	3.268
193422.0	0.9930	3.723	0.267	0.790	-6.306	8.292
193423.0	-4.5674	3.460	-1.320	0.187	-11.350	2.215
201449.0	2.6263	3.073	0.855	0.393	-3.398	8.650
203452.0	-14.4352	3.257	-4.432	0.000	-20.821	-8.050
203457.0	3.2950	4.167	0.791	0.429	-4.875	11.465
205488.0	-12.4851	3.820	-3.269	0.001	-19.973	-4.997
205489.0	-13.2020	3.882	-3.401	0.001	-20.813	-5.591
205490.0	-29.9177	3.856	-7.759	0.000	-37.476	-22.359
205491.0	-17.1001	3.551	-4.815	0.000	-24.062	-10.139
205492.0	9.0822	3.568	2.545	0.011	2.087	16.077
208501.0	-12.6599	3.627	-3.490	0.000	-19.770	-5.550
208503.0	-27.5579	3.681	-7.487	0.000	-34.773	-20.343
209510.0	-19.6791	3.247	-6.061	0.000	-26.044	-13.314
212522.0	-5.5824	3.307	-1.688	0.091	-12.066	0.901
215533.0	1.1303	3.039	0.372	0.710	-4.827	7.087
216536.0	-14.3367	3.180	-4.508	0.000	-20.571	-8.103
218562.0	-2.1570	3.749	-0.575	0.565	-9.506	5.192
221571.0	-39.4847	3.212	-12.292	0.000	-45.782	-33.187
221574.0	-26.0098	3.656	-7.114	0.000	-33.178	-18.842
225585.0	-20.1816	3.409	-5.920	0.000	-26.865	-13.499
228606.0	-6.8095	3.431	-1.985	0.047	-13.536	-0.083
230612.0	7.8012	3.700	2.108	0.035	0.548	15.055
231616.0	-0.5881	3.741	-0.157	0.875	-7.923	6.746
234628.0	-6.8620	3.202	-2.143	0.032	-13.139	-0.585
244697.0	-17.2897	3.421	-5.055	0.000	-23.995	-10.584
244708.0	-21.9512	3.337	-6.578	0.000	-28.493	-15.409
244723.0	-21.2583	3.344	-6.356	0.000	-27.815	-14.702
244727.0	-3.8310	3.597	-1.065	0.287	-10.882	3.220
244728.0	-17.1737	4.097	-4.192	0.000	-25.206	-9.142
244736.0	12.3756	4.071	3.040	0.002	4.395	20.356
244745.0	6.2000	3.447	1.799	0.072	-0.557	12.957
244746.0	2.4693	4.022	0.614	0.539	-5.416	10.354
244755.0	1.2491	3.205	0.390	0.697	-5.033	7.531
244764.0	-6.0930	4.629	-1.316	0.188	-15.167	2.981
244774.0	-3.6727	3.389	-1.084	0.279	-10.316	2.971

244776.0	-8.1379	3.199	-2.544	0.011	-14.408	-1.867
244780.0	30.7077	3.893	7.889	0.000	23.077	38.339
244796.0	-10.4651	3.957	-2.645	0.008	-18.222	-2.708
244799.0	-10.6379	3.808	-2.794	0.005	-18.103	-3.173
244801.0	-12.3885	3.518	-3.521	0.000	-19.286	-5.491
244806.0	19.2458	3.182	6.049	0.000	13.009	25.483
244818.0	-16.9313	3.496	-4.843	0.000	-23.784	-10.078
244831.0	-10.3149	3.798	-2.716	0.007	-17.760	-2.869
244839.0	9.5688	3.620	2.643	0.008	2.472	16.665
252885.0	-4.8554	3.561	-1.363	0.173	-11.837	2.126
253888.0	-11.0608	4.244	-2.606	0.009	-19.381	-2.741
257899.0	-18.3971	3.159	-5.824	0.000	-24.590	-12.204
257905.0	-2.3742	3.021	-0.786	0.432	-8.296	3.548
259915.0	-13.3297	3.721	-3.583	0.000	-20.624	-6.036
261927.0	-8.9582	3.341	-2.681	0.007	-15.508	-2.408
262937.0	5.2735	3.581	1.473	0.141	-1.746	12.293
264945.0	-0.7571	3.230	-0.234	0.815	-7.088	5.574
gender	-4.7411	0.601	-7.892	0.000	-5.919	-3.563
race	9.9647	1.269	7.852	0.000	7.477	12.452
gkfreelunch	-12.3755	0.725	-17.072	0.000	-13.797	-10.954
treated	5.5098	0.659	8.364	0.000	4.218	6.801

Omnibus:	126.475	**Durbin-Watson:**	1.886
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	83.092
Skew:	-0.167	**Prob(JB):**	9.05e-19
Kurtosis:	2.517	**Cond. No.**	100.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [12]:
```
df_trunc = df_trunc.drop(columns=['math_outcome', 'read_outcome'])
```

In [13]:
```
# Do mice
'''
from sklearn.tree import DecisionTreeRegressor

from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer

imp = IterativeImputer(max_iter=10, random_state=1,
                       estimator=DecisionTreeRegressor())
imp.fit(df)
tmp_df = pd.DataFrame(data=np.round(imp.transform(df)),
                      columns=df.columns,
                      index=df.index)
# convert floats to ints because MICE creates floats
cols = list(df.columns)
cols.remove("outcome")
tmp_df[cols] = tmp_df[cols].astype('int64')
'''
```

Out[13]:
```
'\nfrom sklearn.tree import DecisionTreeRegressor\n\nfrom sklearn.experimenta
l import enable_iterative_imputer\nfrom sklearn.impute import IterativeImpute
r\n\nimp = IterativeImputer(max_iter=10, random_state=1,\n
estimator=DecisionTreeRegressor())\nimp.fit(df)\ntmp_df = pd.DataFrame(data=n
p.round(imp.transform(df)),\n                      columns=df.columns,\n
index=df.index)\n# convert floats to ints because MICE creates floats\ncols =
list(df.columns)\ncols.remove("outcome")\ntmp_df[cols] = tmp_df[cols].astype
(\'int64\')\n'
```

In [35]:
```
df_trunc = df_trunc.drop(columns=['gktreadss', 'gktmathss'])
```

In [36]:
```
df_trunc.head()
```

Out[36]:

	gender	race	gkschid	gkfreelunch	treated	outcome
133	1	0.0	169280.0	0	0.0	48.430936
246	0	1.0	218562.0	1	0.0	70.323697
263	0	0.0	205492.0	1	1.0	85.779590
266	1	1.0	257899.0	0	0.0	79.552755
275	1	1.0	161176.0	1	0.0	32.567334

In [42]:
```
# Do the matching

models = []
random_seeds = [1111, 2222, 3333, 4444]
for i in range(4):
    matching_df, holdout_df = train_test_split(df_trunc, test_size=0.2, random
_state=random_seeds[i])
    model_dame = dame_flame.matching.DAME(
        repeats=False, verbose=0, adaptive_weights='decisiontree')
    model_dame.fit(holdout_data=holdout_df)
    model_dame.predict(matching_df)
    models.append(model_dame)
```

```
4601 units matched. We finished with no more treated units to match
4574 units matched. We finished with no more treated units to match
4572 units matched. We finished with no more treated units to match
4549 units matched. We finished with no more treated units to match
```

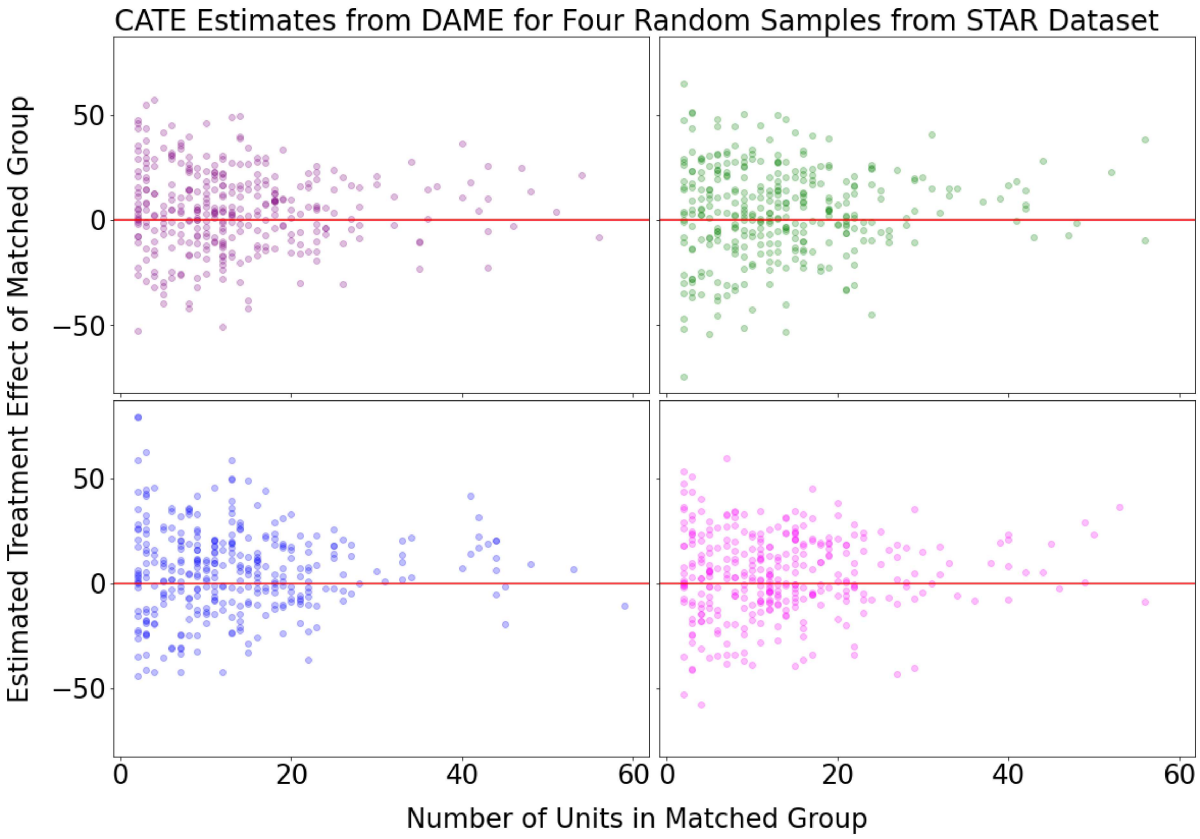
In [43]:
```
for i in range(len(models)):
    ate, var = dame_flame.utils.post_processing.var_ATE(matching_object=models
[i])
    print("ATE of trial", i, ":", ate,". Variance: ", var)
```

```
treated_col treated
ATE of trial 0 : 4.7697509502982705 . Variance:  0.6923843017391174
treated_col treated
ATE of trial 1 : 5.0187232575901835 . Variance:  0.7063379152156198
treated_col treated
ATE of trial 2 : 6.1733118164077405 . Variance:  0.7247732573502101
treated_col treated
ATE of trial 3 : 4.864852280570631 . Variance:  0.6980439193248327
```

In [44]:
```
# compute stuff for plot
# Create the plot
match_dfs = []
for i in models:
    match_dfs.append(i.input_data)

for i in range(4):
    colname = 'cates'
    match_dfs[i][colname] = dame_flame.utils.post_processing.CATE(
        models[i], match_dfs[i].index)

dame_len_groups = []
dame_cate_of_groups = []

for i in range(4):

    model_dame = models[i]
    groups = list(range(len(model_dame.units_per_group)))

    dame_cate_of_group = []
    dame_len_group = []
    dame_len_treated = []
    maxcate = 0.0
    maxgroupnum = 0
    index = 0

    flame_cate_of_group = []
    flame_len_group = []
    large_groups = []
    for group in model_dame.units_per_group:
        dame_cate_of_group.append(dame_flame.utils.post_processing.CATE(
            model_dame, group[0]))
        dame_len_group.append(len(group))

        # find len of just treated units
        df_mmg = df_trunc.loc[group]
        treated = df_mmg.loc[df_mmg["treated"] == 1]

    dame_len_groups.append(dame_len_group)
    dame_cate_of_groups.append(dame_cate_of_group)
```

```
C:\Users\Neha\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
C:\Users\Neha\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
C:\Users\Neha\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
C:\Users\Neha\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```

In [45]:
```
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (19,13),
                                             sharex=True, sharey=True)
fig.text(0.5, 0.05, 'Number of Units in Matched Group', ha='center',
         fontsize=26)
fig.text(0.05, 0.5, 'Estimated Treatment Effect of Matched Group',
         va='center', rotation='vertical', fontsize=26)
fig.suptitle("CATE Estimates from DAME for Four Random Samples from STAR Datas
et", fontsize=28, y=0.91)
ax1.axhline(y=0.0, color='r', linestyle='-')
ax2.axhline(y=0.0, color='r', linestyle='-')
ax3.axhline(y=0.0, color='r', linestyle='-')
ax4.axhline(y=0.0, color='r', linestyle='-')

ax1.tick_params(labelsize=26)
ax2.tick_params(labelsize=26)
ax3.tick_params(labelsize=26)
ax4.tick_params(labelsize=26)

ax1.scatter(dame_len_groups[0], dame_cate_of_groups[0], color="purple",
            alpha = 0.25)
#ax1.text(0.15, 0.9,'ATE: '+str(round(ates[0],2)), ha='center', va='center',
#         transform=ax1.transAxes, fontsize=26)

ax2.scatter(dame_len_groups[1], dame_cate_of_groups[1], color="green",
            alpha = 0.25)
#ax2.text(0.15, 0.9,'ATE: '+str(round(ates[1],2)), ha='center', va='center',
#         transform=ax2.transAxes, fontsize=26)

ax3.scatter(dame_len_groups[2], dame_cate_of_groups[2], color="blue",
            alpha = 0.25)
#ax3.text(0.15, 0.9,'ATE: '+str(round(ates[2],2)), ha='center', va='center',
#         transform=ax3.transAxes, fontsize=26)

ax4.scatter(dame_len_groups[3], dame_cate_of_groups[3], color="magenta",
            alpha = 0.25)
#ax4.text(0.15, 0.9,'ATE: '+str(round(ates[3],2)), ha='center', va='center',
#         transform=ax4.transAxes, fontsize=26)

plt.subplots_adjust(wspace=.02, hspace=.02)
## plt.savefig('cate-graph4.png', dpi = 200)
```



In [47]:
```
list_star_covars = []
for modelid in range(len(models)):

    # Pull out the groups with 10 or more units in the matched group
    model = models[modelid]
    large_groups = []
    for group in model.units_per_group:
        if len(group) >= 12.5:
            large_groups.append(group)

    covariates = set(models[modelid].input_data.columns) - set(['outcome', 'tr
eated', 'cates'])
    # Which covars did the large group match on?
    star_covars = dict()
    for group in large_groups:
        group_star_covars = []
        matched_df = models[modelid].df_units_and_covars_matched.loc[group]
        for covar in covariates:
            if '*' in matched_df[covar].values:
                group_star_covars.append(covar)
        cate_of_group = models[modelid].input_data.loc[group[0], 'cates']
        star_covars[cate_of_group] = group_star_covars

    list_star_covars.append(star_covars)
```

In [48]:
```
list_star_covars
```

Out[48]:
```
[{11.329540239844192: [],
  10.022175337223501: [],
  -10.124783790462047: [],
  -2.2963512066551317: [],
  -10.947821431513063: [],
  5.730430598363952: [],
  27.071118898398367: [],
  6.4272918211020595: [],
  15.322152211514712: [],
  26.52609614803832: [],
  9.798753534854356: [],
  -6.71636054141613: [],
  -15.07090790356174: [],
  4.841640140118315: [],
  10.958185761743962: [],
  34.59127973583126: [],
  17.606663372045134: [],
  -42.155632852445294: [],
  8.819263267002356: [],
  -2.2956225884320745: [],
  8.542129972819374: [],
  -8.015968618729943: [],
  -22.715974207820004: [],
  0.08367589909471462: [],
  27.221575064421625: [],
  3.0463482650287688: [],
  -11.492394915300771: [],
  7.238901243719624: [],
  0.47066021862165996: [],
  12.760935629727314: [],
  -2.950899947924249: [],
  3.9112452492734207: [],
  -9.958922206556615: [],
  -5.569690070246047: [],
  15.948909833268615: [],
  36.302005358605: [],
  21.50623036464401: [],
  29.579229764552238: [],
  17.230870603739397: [],
  20.76847047195453: [],
  13.161130525845117: [],
  -18.56790664240627: [],
  -12.362773130008115: [],
  16.75085601327261: [],
  -15.127352294747247: [],
  33.24120159553813: [],
  17.933279068978706: [],
  24.826631672443384: [],
  25.790498573578628: [],
  4.346803048267741: [],
  10.780346597000921: [],
  14.285329200080866: [],
  15.699643769047043: [],
  10.139672945454997: [],
  -15.65274689070094: [],
  2.542244017410802: [],
  11.463710492620798: [],
```

```
27.765937731653075: [],
5.200457128737327: [],
5.290718671000363: [],
3.609463800345935: [],
-8.29420970266041: [],
-4.120953792933037: [],
16.483109181404206: [],
5.792356478049584: [],
2.1953847843395593: [],
-30.712050078247266: [],
-2.3152952804546416: [],
-3.86589156360138: [],
20.185905883644942: [],
-29.780606717226448: [],
-4.361549413439711: [],
12.643003154493144: [],
38.670620212503096: [],
9.540101608300283: [],
15.899784672949977: [],
39.50333580429949: [],
-15.529728076058674: [],
8.966375852958492: [],
25.44355714296408: [],
22.40960382176099: [],
-9.814368668149243: [],
6.529733959311407: [],
0.5572911151117275: [],
15.679852640564292: [],
4.659418499299896: [],
3.9383830896062193: [],
-2.737418664030976: [],
9.515425888665341: [],
-4.639140888214271: [],
19.16357795898196: [],
23.746965222132005: [],
14.292377069434153: [],
19.103509948580367: [],
9.197485105565157: [],
29.448817662309146: [],
-7.216986127290056: [],
4.342723004694836: [],
-13.74940601775296: [],
2.7295351053690666: [],
-6.967750147883578: [],
-16.715056420393708: [],
-5.861131702495669: [],
14.531340087307484: [],
-3.9205996211185195: [],
7.615629984051026: [],
0.8653529363314334: [],
2.165184086978016: [],
-29.38363122203004: [],
-17.56651017214397: [],
-38.457058138303495: [],
-14.348017932154328: [],
15.767070257804114: [],
13.54591363973313: [],
```

```
 14.615496180745794: [],
 26.630837657524097: [],
 -13.681112013209884: [],
 -20.50181204184169: [],
 -19.334280537023304: [],
 -5.28571036354581: [],
 -5.245243390165555: [],
 -2.049404347402863: [],
 -0.8804876039865093: [],
 18.449805692207462: [],
 15.054978996787746: [],
 2.259810858773058: [],
 1.024577377431342: [],
 21.922617576805862: [],
 1.2615654943305117: [],
 22.78779891826592: [],
 27.472142798310344: [],
 6.284696482991521: [],
 16.71677236910744: [],
 6.287922466573306: [],
 -1.9003680439051749: [],
 16.463558680015325: [],
 48.98041759327898: [],
 49.307580375037745: [],
 -12.259080800593033: [],
 -2.5018532246108265: [],
 -15.98129868778691: ['gkschid'],
 -23.131876916815656: ['gkschid'],
 20.812900344034517: ['gkschid'],
 -10.592789002722277: ['gkschid'],
 -8.883782225516846: ['gkschid']},
{14.878040170849673: [],
 7.481291256309802: [],
 -9.510312649893166: [],
 -4.767841498753285: [],
 -10.666028333745153: [],
 13.852257899505375: [],
 27.07111889839836: [],
 -40.18741327859195: [],
 -14.948315624742612: [],
 4.462814901102519: [],
 20.557670968603375: [],
 25.943243808329036: [],
 1.1055514372786348: [],
 -4.6499839012811535: [],
 -22.662877851906764: [],
 25.459445268099827: [],
 -3.1683963429701123: [],
 10.565851247837912: [],
 31.311576159611747: [],
 33.081824279007385: [],
 -24.357990045536383: [],
 10.615299987056844: [],
 2.0200148257968884: [],
 -9.768260761116004: [],
 -8.119084879648263: [],
 -5.8997258404226685: [],
```

```
13.165739073819132: [],
21.814924635532492: [],
-9.845302543597583: [],
4.594571477447687: [],
-1.0934025203854958: [],
5.328891930078001: [],
10.392948642919393: [],
-1.340159201488774: [],
1.175768058644266: [],
-0.3944108118496459: [],
20.492820470581776: [],
-11.019891277489496: [],
38.26779901376222: [],
22.889588995964104: [],
38.72649774243548: [],
9.191994069681236: [],
20.591384564698124: [],
18.54813089060682: [],
-33.08555033907147: [],
-14.950521844517393: [],
22.06356560415123: [],
-17.17121615139755: [],
40.25132576837543: [],
15.230347923180808: [],
28.028618280656833: [],
18.588778818578675: [],
5.144345605798531: [],
16.634897101909708: [],
8.738753216143888: [],
11.542975803539193: [],
8.355572028663218: [],
-22.226720647773277: [],
-10.335152862951247: [],
8.266648926396897: [],
27.588337039782566: [],
24.377316530763515: [],
1.3555014296304222: [],
-22.65155259039617: [],
-6.005036064339251: [],
-12.83543365455894: [],
-1.2650464188640633: [],
1.9143123850314439: [],
18.538011695906434: [],
5.749114570463732: [],
15.974796145292807: [],
-33.22090437361009: [],
3.348383763506071: [],
14.155615956950271: [],
-1.6127172391071838: [],
19.77040606210361: [],
-30.789401008735602: [],
-5.49515416083247: [],
-7.920476072811141: [],
17.495403040251084: [],
21.4615289578225: [],
-2.0158965488839513: [],
14.06700436537352: [],
```

```
12.715327050019415: [],
6.297566098344454: [],
-18.210334129533535: [],
1.272959393789634: [],
16.5216821999624: [],
24.548276501111943: [],
-6.4019825869562155: [],
18.72543664572847: [],
20.144551519644185: [],
-2.437232088181263: [],
30.871427394778024: [],
-6.3302155554586435: [],
-8.517532627984806: [],
27.74380199324603: [],
-1.9088213491475159: [],
7.474672596985421: [],
19.10246906540457: [],
6.236306729264484: [],
16.176356383917522: [],
26.193541674794844: [],
0.9229058561897787: [],
18.632320237212753: [],
0.3493671581143758: [],
4.248414463388528: [],
14.125395648746284: [],
-4.260919049651434: [],
27.510810476896474: [],
8.64426324026028: [],
47.89017505844786: [],
10.97550213562074: [],
-20.488721804511272: [],
-2.8132187185189323: [],
-0.5096929262978094: [],
-13.955810888724166: [],
-10.812947862614294: [],
25.18059855521158: [],
-10.435507783543372: [],
-0.5587752993238482: [],
-20.34909260632019: [],
-14.426735853718803: [],
-53.476459295584576: [],
14.81964114635489: [],
14.084210051129986: [],
16.918469883629257: [],
-10.042712414839912: [],
-5.220798475060903: [],
-22.917260648913732: [],
-14.492559646377295: [],
-0.7502233373248899: [],
-5.712424467057534: [],
12.530332693416469: [],
23.729511572358135: [],
-2.6373445350465374: [],
-3.6763541211280355: [],
23.2128262150501: [],
8.118359278477882: [],
24.296099775012017: [],
```

```
 40.502182686763845: [],
 10.17259858994477: [],
 11.573815361806467: [],
 8.271371985234111: [],
 4.482058040249285: [],
 8.30189715289788: [],
 12.019191170414288: [],
 -2.5327403014578778: [],
 23.884942583978905: [],
 49.888497652582174: [],
 41.58186760114115: [],
 -15.085042418252208: [],
 2.4074477826712695: [],
 -3.299288910853029: [],
 16.9315827488548: ['gkschid'],
 -21.073222963512066: ['gkschid'],
 -7.27410716291368: ['gkschid'],
 -44.8668364112977: ['gkschid']},
{13.06972617202419: [],
 11.794862743713722: [],
 -9.855702844518092: [],
 -2.731130371975439: [],
 8.609691389333236: [],
 6.188607392549301: [],
 5.6027753442505315: [],
 11.83239789614882: [],
 25.446627131208302: [],
 15.024915575323291: [],
 -5.134836130017746: [],
 -18.734865332344945: [],
 -7.137231282431436: [],
 31.07322296351206: [],
 21.012272465200567: [],
 -17.76250271431885: [],
 -10.508298327979574: [],
 7.031302648426447: [],
 -1.8894066151293742: [],
 21.634258995267146: [],
 -10.568073906112978: [],
 -19.56353128517695: [],
 -23.554484803558196: [],
 -10.081541882876188: [],
 14.592035644667213: [],
 21.607539976702316: [],
 -10.406463371622756: [],
 0.39535458364220943: [],
 -2.697059550284152: [],
 18.366659285858688: [],
 -5.515261042230897: [],
 6.458066135605073: [],
 2.629406669436321: [],
 -1.766387800487145: [],
 19.210217445021: [],
 -16.026960986190044: [],
 31.243307800016474: [],
 20.3117051119929: [],
 48.75353485435028: [],
```

```
22.552971336792695: [],
8.228905597326658: [],
18.71727440667398: [],
-11.432807370539027: [],
-15.268511654723664: [],
12.71188534717075: [],
-21.130878840293228: [],
43.726597099465884: [],
18.623119139583583: [],
22.257499309796113: [],
20.166849047501373: [],
6.31886745136703: [],
14.020166885252138: [],
13.418044498877325: [],
11.967712709002555: [],
-0.11943003047524314: [],
-23.881887818136896: [],
10.24664734820405: [],
11.1312532291035: [],
27.473025286220242: [],
28.74872841514798: [],
-1.8127282211789293: [],
4.270559561515832: [],
-6.325879252120913: [],
14.425735700754231: [],
0.6620130137550504: [],
-36.21189123042346: [],
2.0778177838963643: [],
10.862367185569553: [],
-1.92924315760402: [],
17.99798547621009: [],
4.771498989438207: [],
-22.876439926106364: [],
-8.92358537188042: [],
-8.75957499382259: [],
17.22534939732863: [],
16.87484556461576: [],
3.3959311424100136: [],
15.753880005177265: [],
37.253299373388344: [],
-22.21975125607448: [],
-10.315871839222474: [],
-3.056938119947759: [],
18.9208616932857: [],
32.87852112676057: [],
-8.281370368755361: [],
13.79540400296515: [],
13.281717596024492: [],
-0.032197437682985: [],
22.82843258380695: [],
0.671132055490844: [],
29.22947038958899: [],
-7.431430689399548: [],
35.56166433297642: [],
19.661028371184038: [],
-1.2898126500795186: [],
17.966154627570475: [],
```

```
20.526111423852456: [],
18.893183663383823: [],
23.533481591302184: [],
7.094320307811799: [],
1.309612058314805: [],
-2.2279878099003128: [],
33.11588831233012: [],
4.500247096614785: [],
49.60361584712957: [],
-3.6993893183663396: [],
-4.442347867106939: [],
-4.065974796145305: [],
-3.9065974796145397: [],
-12.412613331812736: [],
-5.773000576558772: [],
12.345960603929477: [],
-8.531858505695332: [],
0.924406085636619: [],
5.666749032204933: [],
-20.885704088076224: [],
-32.970924964994644: [],
-2.3062350712462063: [],
-25.694517985103616: [],
20.381754980483095: [],
16.63070038162701: [],
18.455834165941496: [],
2.64128631437751: [],
26.081047689646653: [],
-23.769767729182107: [],
-28.485827244166757: [],
-10.807182274936167: [],
-3.4365550025298077: [],
4.072152211514712: [],
-14.756815748290919: [],
8.69630328488742: [],
17.026329517063388: [],
29.00420064245121: [],
39.02778713749803: [],
-8.95972325179146: [],
-1.3569722428136117: [],
17.944156165060534: [],
-1.7309117865085284: [],
22.732202179941247: [],
21.775687397220608: [],
7.598643291749298: [],
10.026680551143116: [],
11.160942261757668: [],
7.329740928703146: [],
7.870301732421822: [],
8.142150247413412: [],
3.150481838398818: [],
25.467012601927344: [],
49.162651346676554: [],
58.55796661949368: [],
-10.33582676280971: [],
-4.237706943414885: [],
-9.771435631331855: [],
```

```
 6.311259369079977: [],
 23.048966312494848: ['gkschid'],
 1.0312165390000771: ['gkschid'],
 41.69508277736594: ['gkschid'],
 9.186699142221748: ['gkschid'],
 7.057057057057044: ['gkschid']},
{15.008569938147417: [],
 5.054744681032375: [],
 -6.942300153645803: [],
 -4.62104212158718: [],
 0.21873422073247184: [],
 15.321564751425633: [],
 -20.459038120268666: [],
 -8.39826483266068: [],
 2.026878620651786: [],
 16.80462894325015: [],
 18.492710649864087: [],
 -1.5309694423853202: [],
 -39.45206325673338: [],
 -7.955824616313873: [],
 31.082400837775182: [],
 19.410262746067062: [],
 -21.489168931718964: [],
 17.894512208819116: [],
 -5.4681892524739055: [],
 18.582299012247127: [],
 -8.744971174982375: [],
 -7.76848591549296: [],
 -6.353128517694863: [],
 -5.498648456394918: [],
 12.121461713752296: [],
 20.893391538313715: [],
 -6.100253835613358: [],
 6.8587842846553: [],
 8.933425112075966: [],
 -0.3224610822831835: [],
 7.987974631414211: [],
 0.1685892517307508: [],
 -8.093786892897342: [],
 -2.555139449134998: [],
 16.10108997062295: [],
 -14.991626170277026: [],
 35.98831158134345: [],
 22.984591141803108: [],
 32.40335152862953: [],
 38.305380528786756: [],
 18.899287538094065: [],
 14.342532930375782: [],
 -19.90980973560663: [],
 -8.63328116849243: [],
 24.300716580182865: [],
 -16.876698789226587: [],
 32.99795583709593: [],
 23.15759771533586: [],
 28.726678865670706: [],
 19.194379830099777: [],
 5.2392453191133335: [],
```

```
20.77302999282243: [],
9.642092617135862: [],
13.653558779578056: [],
9.881773773545468: [],
-4.44212323382078: [],
-2.0215987784557257: [],
9.21979243884359: [],
-10.409506480670316: [],
23.838458716145894: [],
31.453202646679287: [],
-18.5260686928589: [],
0.48321115778492185: [],
-9.97137797545507: [],
-3.2747165252724955: [],
27.414957581747796: [],
-1.8588932978676986: [],
-5.509842681821922: [],
-33.87999825578859: [],
4.055417031695768: [],
-4.933283914010389: [],
18.091324001802363: [],
-25.038806840141163: [],
-30.149787614576358: [],
4.709965872018273: [],
21.766417216750803: [],
23.43917304999588: [],
1.4486039041265073: [],
7.7907503500535356: [],
16.580463643102643: [],
34.101017588786306: [],
0.9323778930895372: [],
21.45808574644748: [],
28.067086730911797: [],
-8.042377069434146: [],
12.278666841087805: [],
24.4613293797875: [],
-0.04230186611660258: [],
21.381064162754313: [],
-1.965685249599261: [],
-0.6877522444609241: [],
19.10551025450951: [],
-2.7441763820562954: [],
21.34061680021651: [],
23.949153007714912: [],
13.99449327544211: [],
18.39963759163166: [],
18.173956016802578: [],
-0.4197553743513609: [],
-12.536446750679524: [],
20.50901902643934: [],
23.89836092578866: [],
14.191582241989948: [],
1.6345441067457394: [],
-3.368750514784601: [],
-7.299494102193975: [],
-0.7880322872910028: [],
-6.871756856931057: [],
```

```
-8.502525876506596: [],
12.831006506877536: [],
-9.975153062625267: [],
-0.948027345358696: [],
8.018696977184732: [],
-33.87675869143623: [],
-15.37744007907093: [],
-36.21593106396665: [],
-10.152582159624416: [],
-0.5047830844717396: [],
17.403014578700272: [],
18.56988582535773: [],
23.315418828762063: [],
-5.30757645286927: [],
26.324026027510094: [],
-7.246010174105066: [],
-27.326414628119593: [],
-14.012003693444143: [],
5.275292103498167: [],
-0.6132488712177384: [],
8.854295362820181: [],
21.031478611167273: [],
11.0961823573017: [],
30.873349257337402: [],
-11.196284565446916: [],
-5.696091755209622: [],
20.224857919446507: [],
5.154023556543947: [],
24.692813981175725: [],
35.22811510209583: [],
10.202928094885095: [],
9.516014213938597: [],
12.611811218186318: [],
4.0545060189241156: [],
-1.3189145191063787: [],
15.734288773577127: [],
44.90960382176098: [],
9.745559124728885: [],
-3.3553661148175635: [],
26.997364302775726: [],
-4.313895066304262: ['gkschid'],
-5.789199132416336: ['gkschid'],
-40.15099368138657: ['gkschid'],
-43.3150861987037: ['gkschid']}]
```

In [49]:
```
## Check the matched group with the most units in each trial -- also which cov
ariates did they use and which units in their MMG?
```

In [50]:
```
## Run DAME and FLAME and show why we chose DAME for this dataset. What happen
s if we run FLAME?
flame_models = []
random_seeds = [1111, 2222, 3333, 4444]
for i in range(4):
    matching_df, holdout_df = train_test_split(df_trunc, test_size=0.2, random
_state=random_seeds[i])
    model_flame = dame_flame.matching.FLAME(
        repeats=False, verbose=3, adaptive_weights='decisiontree',
        missing_holdout_replace=1, missing_data_replace=1)
    model_flame.fit(holdout_data=holdout_df)
    result_flame = model_flame.predict(matching_df)
    flame_models.append(model_flame)
```

```
Iteration number:  1
        Number of matched groups formed in total:  314
        Unmatched treated units:  44 out of a total of  1416 treated units
        Unmatched control units:  197 out of a total of  3212 control units
        Predictive error of covariates chosen this iteration:  0
        Number of matches made in this iteration:  4387
        Number of matches made so far:  4387
        In this iteration, the covariates dropped are:  set()
Iteration number:  2
        Number of matched groups formed in total:  316
        Unmatched treated units:  41 out of a total of  1416 treated units
        Unmatched control units:  188 out of a total of  3212 control units
        Predictive error of covariates chosen this iteration:  517.8572126493
38
        Number of matches made in this iteration:  12
        Number of matches made so far:  4399
        In this iteration, the covariates dropped are:  race
Iteration number:  3
        Number of matched groups formed in total:  327
        Unmatched treated units:  27 out of a total of  1416 treated units
        Unmatched control units:  172 out of a total of  3212 control units
        Predictive error of covariates chosen this iteration:  687.3596079168
926
        Number of matches made in this iteration:  42
        Number of matches made so far:  4429
        In this iteration, the covariates dropped are:  gender
Iteration number:  4
        Number of matched groups formed in total:  331
        Unmatched treated units:  23 out of a total of  1416 treated units
        Unmatched control units:  162 out of a total of  3212 control units
        Predictive error of covariates chosen this iteration:  871.2494409340
983
        Number of matches made in this iteration:  56
        Number of matches made so far:  4443
        In this iteration, the covariates dropped are:  gkfreelunch
4443 units matched. No more covariate sets to consider dropping
Iteration number:  1
        Number of matched groups formed in total:  310
        Unmatched treated units:  44 out of a total of  1392 treated units
        Unmatched control units:  194 out of a total of  3236 control units
        Predictive error of covariates chosen this iteration:  0
        Number of matches made in this iteration:  4390
        Number of matches made so far:  4390
        In this iteration, the covariates dropped are:  set()
Iteration number:  2
        Number of matched groups formed in total:  314
        Unmatched treated units:  38 out of a total of  1392 treated units
        Unmatched control units:  173 out of a total of  3236 control units
        Predictive error of covariates chosen this iteration:  547.9862990328
598
        Number of matches made in this iteration:  27
        Number of matches made so far:  4417
        In this iteration, the covariates dropped are:  race
Iteration number:  3
        Number of matched groups formed in total:  322
        Unmatched treated units:  28 out of a total of  1392 treated units
        Unmatched control units:  164 out of a total of  3236 control units
```

```
        Predictive error of covariates chosen this iteration:  751.4676223296
788
        Number of matches made in this iteration:  46
        Number of matches made so far:  4436
        In this iteration, the covariates dropped are:  gender
Iteration number:  4
        Number of matched groups formed in total:  329
        Unmatched treated units:  21 out of a total of  1392 treated units
        Unmatched control units:  149 out of a total of  3236 control units
        Predictive error of covariates chosen this iteration:  932.0542372602
191
        Number of matches made in this iteration:  68
        Number of matches made so far:  4458
        In this iteration, the covariates dropped are:  gkfreelunch
4458 units matched. No more covariate sets to consider dropping
Iteration number:  1
        Number of matched groups formed in total:  314
        Unmatched treated units:  31 out of a total of  1379 treated units
        Unmatched control units:  238 out of a total of  3249 control units
        Predictive error of covariates chosen this iteration:  0
        Number of matches made in this iteration:  4359
        Number of matches made so far:  4359
        In this iteration, the covariates dropped are:  set()
Iteration number:  2
        Number of matched groups formed in total:  315
        Unmatched treated units:  30 out of a total of  1379 treated units
        Unmatched control units:  237 out of a total of  3249 control units
        Predictive error of covariates chosen this iteration:  539.1790833411
786
        Number of matches made in this iteration:  2
        Number of matches made so far:  4361
        In this iteration, the covariates dropped are:  race
Iteration number:  3
        Number of matched groups formed in total:  322
        Unmatched treated units:  21 out of a total of  1379 treated units
        Unmatched control units:  216 out of a total of  3249 control units
        Predictive error of covariates chosen this iteration:  731.2877772811
73
        Number of matches made in this iteration:  32
        Number of matches made so far:  4391
        In this iteration, the covariates dropped are:  gender
Iteration number:  4
        Number of matched groups formed in total:  326
        Unmatched treated units:  15 out of a total of  1379 treated units
        Unmatched control units:  205 out of a total of  3249 control units
        Predictive error of covariates chosen this iteration:  940.3173265999
441
        Number of matches made in this iteration:  49
        Number of matches made so far:  4408
        In this iteration, the covariates dropped are:  gkfreelunch
4408 units matched. No more covariate sets to consider dropping
Iteration number:  1
        Number of matched groups formed in total:  310
        Unmatched treated units:  38 out of a total of  1393 treated units
        Unmatched control units:  207 out of a total of  3235 control units
        Predictive error of covariates chosen this iteration:  0
        Number of matches made in this iteration:  4383
```

```
        Number of matches made so far:  4383
        In this iteration, the covariates dropped are:  set()
Iteration number:  2
        Number of matched groups formed in total:  312
        Unmatched treated units:  34 out of a total of  1393 treated units
        Unmatched control units:  199 out of a total of  3235 control units
        Predictive error of covariates chosen this iteration:  577.2231118671
331
        Number of matches made in this iteration:  12
        Number of matches made so far:  4395
        In this iteration, the covariates dropped are:  race
Iteration number:  3
        Number of matched groups formed in total:  322
        Unmatched treated units:  20 out of a total of  1393 treated units
        Unmatched control units:  183 out of a total of  3235 control units
        Predictive error of covariates chosen this iteration:  766.1047267758
408
        Number of matches made in this iteration:  42
        Number of matches made so far:  4425
        In this iteration, the covariates dropped are:  gender
Iteration number:  4
        Number of matched groups formed in total:  327
        Unmatched treated units:  15 out of a total of  1393 treated units
        Unmatched control units:  176 out of a total of  3235 control units
        Predictive error of covariates chosen this iteration:  945.9097832607
263
        Number of matches made in this iteration:  54
        Number of matches made so far:  4437
        In this iteration, the covariates dropped are:  gkfreelunch
4437 units matched. No more covariate sets to consider dropping
```

In [52]:
```
# whats the var of the ates?
for model in flame_models:
    print(dame_flame.utils.post_processing.var_ATE(matching_object=model))
```

```
treated_col treated
(5.178996045917732, 0.6725400390177182)
treated_col treated
(5.511733223163917, 0.6951311316613115)
treated_col treated
(5.783540922380401, 0.6712097855583286)
treated_col treated
(5.6080176822253796, 0.664540767588431)
```

In [ ]:
```

```