In [1]:
```
import numpy as np
import pandas as pd
import dame_flame
```

Exercise 1

In [2]:
```
cps = pd.read_stata(
    "https://github.com/nickeubank/MIDS_Data/blob/master"
    "/Current_Population_Survey/cps_for_matching.dta?raw=true"
)
```

In [3]:
```
cps.head()
```

Out[3]:

	index	annual_earnings	female	simplified_race	has_college	age	county	class94
0	151404	NaN	1	3.0	1	30	0-WV	Private, For Profit
1	123453	NaN	0	0.0	0	21	251-TX	Private, For Profit
2	187982	NaN	0	0.0	0	40	5-MA	Self-Employed, Unincorporated
3	122356	NaN	1	0.0	1	27	0-TN	Private, Nonprofit
4	210750	42900.0	1	0.0	0	52	0-IA	Private, For Profit

In [4]:
```
col=cps[cps["has_college"]==1]
no_col=cps[cps["has_college"]==0]
```

In [5]:
```
col['annual_earnings'].mean()-no_col['annual_earnings'].mean()
```

Out[5]:
```
14158.495868997452
```

In [6]:
```
from scipy.stats import ttest_ind
ttest_ind(col['annual_earnings'], no_col['annual_earnings'], equal_var=F
alse, nan_policy='omit')
```

Out[6]:
```
Ttest_indResult(statistic=15.103039559775413, pvalue=6.831437807875693e
-48)
```

The p value smaller than 0.05. Thus, the difference on annul earnings between groups with college degree and group without college degree is statistically significant.

Exercise 2

In [7]:
```
col['simplified_race']=col['simplified_race'].astype(str)
```

```
<ipython-input-7-1c7599480595>:1: SettingWithCopyWarning:
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/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  col['simplified_race']=col['simplified_race'].astype(str)
```

In [8]:
```
col.groupby('simplified_race')['simplified_race'].count()/col['simplifie
d_race'].count()
```

Out[8]:
```
simplified_race
0.0    0.752761
1.0    0.073022
2.0    0.067613
3.0    0.106604
Name: simplified_race, dtype: float64
```

The difference is statistically significant since 0.0 group takes account of 75% of the people who have college degree.

In [9]:
```
col.groupby(["county","simplified_race"])['simplified_race'].count()/col
.groupby("county")['simplified_race'].count()
```

Out[9]:
```
county  simplified_race
0-AK    0.0                0.813953
        2.0                0.046512
        3.0                0.139535
0-AL    0.0                0.796610
        1.0                0.152542
                             ...
99-FL   2.0                0.238095
        3.0                0.047619
99-MI   0.0                0.900000
        3.0                0.100000
99-MO   0.0                1.000000
Name: simplified_race, Length: 628, dtype: float64
```

In [10]:
```
no_col.groupby(["county","simplified_race"])['simplified_race'].count()/
no_col.groupby("county")['simplified_race'].count()
```

Out[10]:
```
county  simplified_race
0-AK    0.0                0.611111
        1.0                0.022222
        2.0                0.066667
        3.0                0.300000
0-AL    0.0                0.750000
                             ...
99-FL   2.0                0.466667
99-MI   0.0                0.818182
        1.0                0.090909
        2.0                0.090909
99-MO   0.0                1.000000
Name: simplified_race, Length: 798, dtype: float64
```

The distribution is different across counties since some counties will lack certain typye of race and thus makes the distribution different.

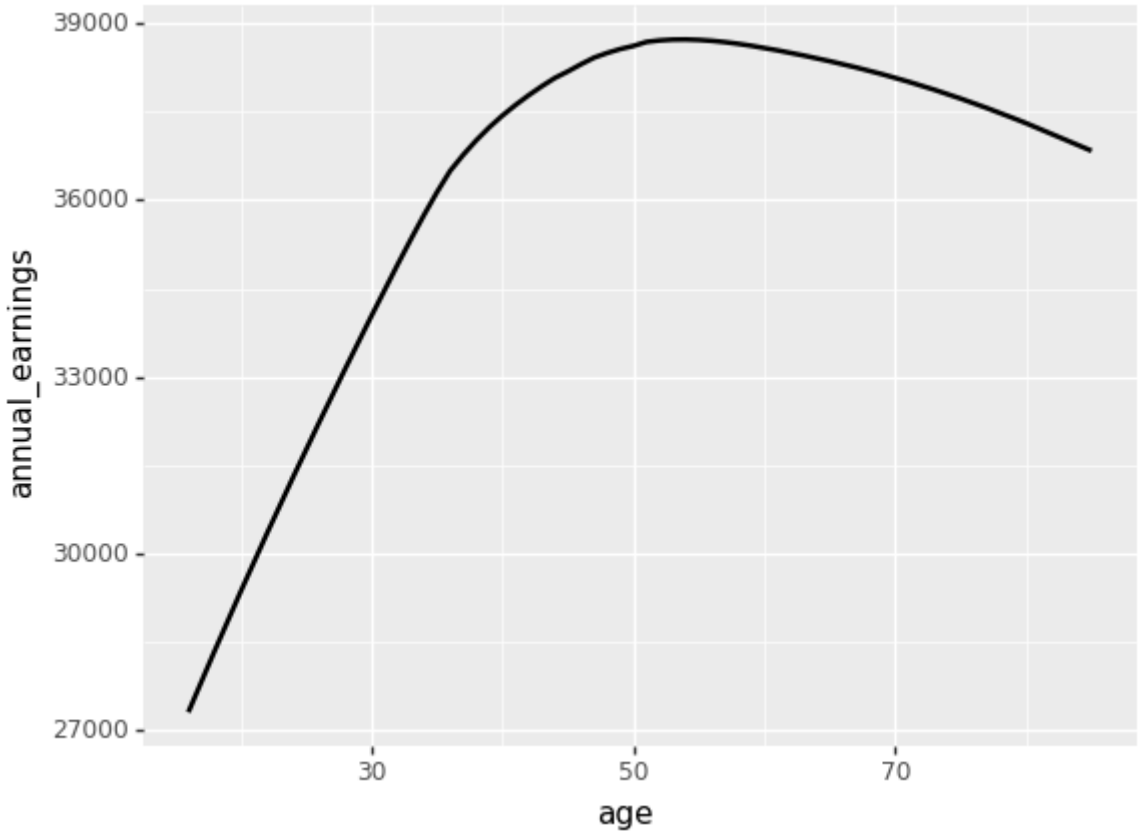
The data looks quite balanced in county but not in race.

Exercise 3

In [13]:
```
from plotnine import ggplot, geom_point, aes, geom_smooth, facet_wrap
from plotnine.data import mtcars

(ggplot(cps, aes('age', 'annual_earnings'))
 # + geom_point()
 + geom_smooth(method="lowess"))
```

```
/opt/anaconda3/lib/python3.8/site-packages/plotnine/stats/smoothers.py:
310: PlotnineWarning: Confidence intervals are not yet implementedfor l
owess smoothings.
```



Out[13]: <ggplot: (8784426297947)>

The relation does not look linear.
Matching attempts to limit the sample to units with values of X shared by units with adifferent treatment assignment. It allows for a direct comparison of outcomes. As a result, we do not have to posit a functional form.

Exercise 4

In [14]:
```
cps['categoryage']=cps["age"].apply(lambda x: str(x)[:1]+str(0))
```

Exercise 5

In [17]:
```
cps['class94']
```

Out[17]:
```
0                   Private, For Profit
1                   Private, For Profit
2         Self-Employed, Unincorporated
3                    Private, Nonprofit
4                   Private, For Profit
                          ...
11145               Private, For Profit
11146               Private, For Profit
11147                Government - State
11148               Private, For Profit
11149                Private, Nonprofit
Name: class94, Length: 11150, dtype: object
```

In [18]:
```
cps['class94']=pd.Categorical(cps['class94'])
cps['nuclass94']=cps['class94'].cat.codes
cps['county']=pd.Categorical(cps['county'])
cps['nucounty']=cps['county'].cat.codes
cps['categoryage']=cps['categoryage'].astype(int)
cps['simplified_race']=cps['simplified_race'].astype(int)
```

Exercise 6

In [19]:
```
cpsn=cps.drop(['age','class94','index','county'],axis=1)
```

In [20]:
```
cpsn=cpsn.dropna()
```

In [21]:
```
cpsn.dtypes
```

Out[21]:
```
annual_earnings    float64
female               int32
simplified_race      int64
has_college          int32
categoryage          int64
nuclass94             int8
nucounty             int16
dtype: object
```

In [22]:
```
cpsn.head()
```

Out[22]:

	annual_earnings	female	simplified_race	has_college	categoryage	nuclass94	nucounty
4	42900.0	1	0	0	50	3	10
5	31200.0	0	2	0	30	3	31
7	20020.0	0	0	1	60	3	8
8	22859.2	0	0	0	40	1	44
9	73860.8	0	0	1	30	3	24

Exercise7

In [23]:
```
model = dame_flame.matching.DAME(repeats=False, verbose=3, want_pe=True)
model.fit(
    cpsn,
    treatment_column_name="has_college",
    outcome_column_name="annual_earnings",
)
result = model.predict(cpsn)
```

```
Iteration number:  1
        Number of matched groups formed in total:  366
        Unmatched treated units:  648 out of a total of  1150 treated u
nits
        Unmatched control units:  3225 out of a total of  4365 control
units
        Predictive error of covariates chosen this iteration:  0
        Number of matches made in this iteration:  1642
        Number of matches made so far:  1642
        In this iteration, the covariates dropped are:  set()
Iteration number:  2
        Number of matched groups formed in total:  491
        Unmatched treated units:  26 out of a total of  1150 treated un
its
        Unmatched control units:  286 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  11988378
30.7153406
        Number of matches made in this iteration:  3561
        Number of matches made so far:  5203
        In this iteration, the covariates dropped are:  frozenset({'nuc
ounty'})
Iteration number:  3
        Number of matched groups formed in total:  491
        Unmatched treated units:  26 out of a total of  1150 treated un
its
        Unmatched control units:  286 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12041585
96.3923833
        Number of matches made in this iteration:  0
        Number of matches made so far:  5203
        In this iteration, the covariates dropped are:  frozenset({'sim
plified_race'})
Iteration number:  4
        Number of matched groups formed in total:  503
        Unmatched treated units:  8 out of a total of  1150 treated uni
ts
        Unmatched control units:  233 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12041733
49.104225
        Number of matches made in this iteration:  71
        Number of matches made so far:  5274
        In this iteration, the covariates dropped are:  frozenset({'sim
plified_race', 'nucounty'})
Iteration number:  5
        Number of matched groups formed in total:  503
        Unmatched treated units:  8 out of a total of  1150 treated uni
ts
        Unmatched control units:  233 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12044963
82.1393971
        Number of matches made in this iteration:  0
        Number of matches made so far:  5274
        In this iteration, the covariates dropped are:  frozenset({'nuc
```

```
lass94'})
Iteration number:  6
        Number of matched groups formed in total:  506
        Unmatched treated units:  5 out of a total of  1150 treated uni
ts
        Unmatched control units:  225 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12045980
48.2615771
        Number of matches made in this iteration:  11
        Number of matches made so far:  5285
        In this iteration, the covariates dropped are:  frozenset({'nuc
ounty', 'nuclass94'})
Iteration number:  7
        Number of matched groups formed in total:  507
        Unmatched treated units:  4 out of a total of  1150 treated uni
ts
        Unmatched control units:  224 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12099655
81.477787
        Number of matches made in this iteration:  2
        Number of matches made so far:  5287
        In this iteration, the covariates dropped are:  frozenset({'sim
plified_race', 'nuclass94'})
Iteration number:  8
        Number of matched groups formed in total:  509
        Unmatched treated units:  0 out of a total of  1150 treated uni
ts
        Unmatched control units:  215 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12099803
89.0274878
        Number of matches made in this iteration:  13
        Number of matches made so far:  5300
        In this iteration, the covariates dropped are:  frozenset({'sim
plified_race', 'nucounty', 'nuclass94'})
5300 units matched. We finished with no more treated units to match
```

Exercise8

In [24]:
```
error=model.pe_each_iter
```

In [25]:
```
error
```

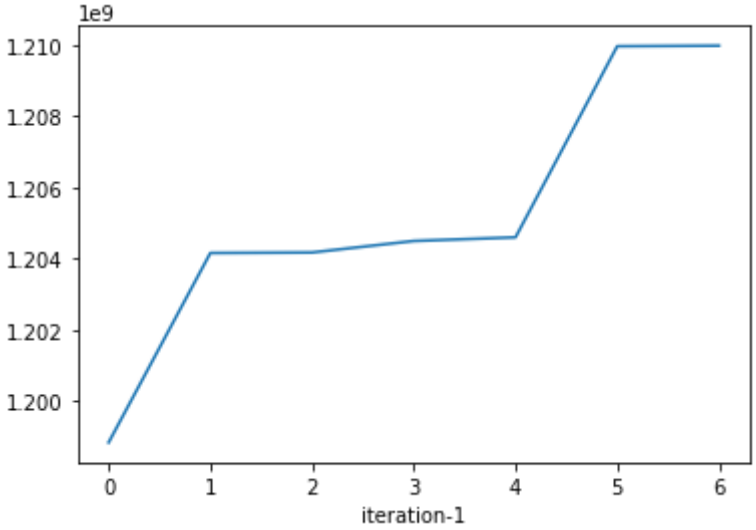
Out[25]:
```
[1198837830.7153406,
 1204158596.3923833,
 1204173349.104225,
 1204496382.1393971,
 1204598048.2615771,
 1209965581.477787,
 1209980389.0274878]
```

In [26]:
```
error[0]
```

Out[26]:
```
1198837830.7153406
```

In [27]:
```
import matplotlib.pyplot as plt
plt.plot(error)
plt.xlabel('iteration-1')
```

Out[27]:
```
Text(0.5, 0, 'iteration-1')
```



The prediction quality drops dramatically at iteration 2 and iteration 6.

Exercise 9

We can stop at Iteration 2

Exercise 10

In [28]:
```
model = dame_flame.matching.DAME(repeats=False, verbose=3, want_pe=True,
early_stop_iterations=6)
model.fit(
    cpsn,
    treatment_column_name="has_college",
    outcome_column_name="annual_earnings",
)
result = model.predict(cpsn)
```

```
Iteration number:  1
        Number of matched groups formed in total:  366
        Unmatched treated units:  648 out of a total of  1150 treated u
nits
        Unmatched control units:  3225 out of a total of  4365 control
units
        Predictive error of covariates chosen this iteration:  0
        Number of matches made in this iteration:  1642
        Number of matches made so far:  1642
        In this iteration, the covariates dropped are:  set()
Iteration number:  2
        Number of matched groups formed in total:  491
        Unmatched treated units:  26 out of a total of  1150 treated un
its
        Unmatched control units:  286 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  11988378
30.7153406
        Number of matches made in this iteration:  3561
        Number of matches made so far:  5203
        In this iteration, the covariates dropped are:  frozenset({'nuc
ounty'})
Iteration number:  3
        Number of matched groups formed in total:  491
        Unmatched treated units:  26 out of a total of  1150 treated un
its
        Unmatched control units:  286 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12041585
96.3923833
        Number of matches made in this iteration:  0
        Number of matches made so far:  5203
        In this iteration, the covariates dropped are:  frozenset({'sim
plified_race'})
Iteration number:  4
        Number of matched groups formed in total:  503
        Unmatched treated units:  8 out of a total of  1150 treated uni
ts
        Unmatched control units:  233 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12041733
49.104225
        Number of matches made in this iteration:  71
        Number of matches made so far:  5274
        In this iteration, the covariates dropped are:  frozenset({'sim
plified_race', 'nucounty'})
Iteration number:  5
        Number of matched groups formed in total:  503
        Unmatched treated units:  8 out of a total of  1150 treated uni
ts
        Unmatched control units:  233 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12044963
82.1393971
        Number of matches made in this iteration:  0
        Number of matches made so far:  5274
        In this iteration, the covariates dropped are:  frozenset({'nuc
```

```
lass94'})
Iteration number:  6
        Number of matched groups formed in total:  506
        Unmatched treated units:  5 out of a total of  1150 treated uni
ts
        Unmatched control units:  225 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  12045980
48.2615771
        Number of matches made in this iteration:  11
        Number of matches made so far:  5285
        In this iteration, the covariates dropped are:  frozenset({'nuc
ounty', 'nuclass94'})
5285 units matched. We stopped before doing iteration number:  6
```

Exercise 11

In [29]:
```
result_of_fit=result
```

In [30]:
```
def get_dataframe(model, result_of_fit):

    # Get original data
    better = model.input_data.loc[result_of_fit.index]
    if not better.index.is_unique:
        raise ValueError("Need index values in input data to be unique")

    # Get match groups for clustering
    better["match_group"] = np.nan
    better["match_group_size"] = np.nan
    for idx, group in enumerate(model.units_per_group):
        better.loc[group, "match_group"] = idx
        better.loc[group, "match_group_size"] = len(group)

    # Get weights. I THINK this is right?! At least for with repeat=Fals
e?
    t = model.treatment_column_name
    better["t_in_group"] = better.groupby("match_group")[t].transform(np
.sum)

    # Make weights
    better["weights"] = np.nan
    better.loc[better[t] == 1, "weights"] = 1  # treaments are 1

    # Controls start as proportional to num of treatments
    # each observation is matched to.
    better.loc[better[t] == 0, "weights"] = better["t_in_group"] / (
        better["match_group_size"] - better["t_in_group"]
    )

    # Then re-normalize for num unique control observations.
    control_weights = better[better[t] == 0]["weights"].sum()

    num_control_obs = len(better[better[t] == 0].index.drop_duplicates
())
    renormalization = num_control_obs / control_weights
    better.loc[better[t] == 0, "weights"] = (
        better.loc[better[t] == 0, "weights"] * renormalization
    )
    assert better.weights.notnull().all()

    better = better.drop(["t_in_group"], axis="columns")

    # Make sure right length and values!
    assert len(result_of_fit) == len(better)
    assert int(better.loc[better[t] == 0, "weights"].sum()) == num_contr
ol_obs
    return better
```

In [31]:
```
newcps=get_dataframe(model,result)
```

In [32]:
```
newcps.head()
```

Out[32]:

	annual_earnings	female	simplified_race	has_college	categoryage	nuclass94	nucounty	matc
4	42900.0	1	0	0	50	3	10	
5	31200.0	0	2	0	30	3	31	
7	20020.0	0	0	1	60	3	8	
8	22859.2	0	0	0	40	1	44	
9	73860.8	0	0	1	30	3	24	

Exercise 12

In [33]:
```
import statsmodels.formula.api as smf
smf.wls(
    "has_college ~ C(simplified_race)", newcps, weights=newcps["weights"
]
).fit().summary()
```

Out[33]:

WLS Regression Results

Dep. Variable:	has_college	**R-squared:**	0.000
Model:	WLS	**Adj. R-squared:**	-0.000
Method:	Least Squares	**F-statistic:**	0.4234
Date:	Sun, 28 Feb 2021	**Prob (F-statistic):**	0.736
Time:	11:12:49	**Log-Likelihood:**	-3719.1
No. Observations:	5285	**AIC:**	7446.
Df Residuals:	5281	**BIC:**	7473.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.2166	0.007	31.743	0.000	0.203	0.230
C(simplified_race)[T.1]	-0.0046	0.018	-0.254	0.799	-0.040	0.031
C(simplified_race)[T.2]	-0.0096	0.019	-0.510	0.610	-0.046	0.027
C(simplified_race)[T.3]	0.0178	0.020	0.890	0.373	-0.021	0.057

Omnibus:	833.342	**Durbin-Watson:**	1.996
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1280.627
Skew:	1.200	**Prob(JB):**	8.23e-279
Kurtosis:	2.765	**Cond. No.**	3.91

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

The difference is not statistically significant.

Exercise 13

In [34]:
```
smf.wls(
    "annual_earnings ~ C(has_college)", newcps, weights=newcps["weights"
]
).fit().summary()
```

Out[34]:

WLS Regression Results

Dep. Variable:	annual_earnings	**R-squared:**	0.060
Model:	WLS	**Adj. R-squared:**	0.060
Method:	Least Squares	**F-statistic:**	337.4
Date:	Sun, 28 Feb 2021	**Prob (F-statistic):**	4.25e-73
Time:	11:13:14	**Log-Likelihood:**	-61416.
No. Observations:	5285	**AIC:**	1.228e+05
Df Residuals:	5283	**BIC:**	1.228e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.914e+04	352.890	110.905	0.000	3.84e+04	3.98e+04
C(has_college)[T.1]	1.393e+04	758.156	18.369	0.000	1.24e+04	1.54e+04

Omnibus:	2844.511	**Durbin-Watson:**	1.998
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	30246.142
Skew:	2.362	**Prob(JB):**	0.00
Kurtosis:	13.725	**Cond. No.**	2.56

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

In [35]:
```
smf.ols(
    "annual_earnings ~ C(has_college)", newcps, weights=newcps["weights"
]
).fit().summary()
```

Out[35]:

OLS Regression Results

Dep. Variable:	annual_earnings	**R-squared:**	0.085
Model:	OLS	**Adj. R-squared:**	0.084
Method:	Least Squares	**F-statistic:**	487.8
Date:	Sun, 28 Feb 2021	**Prob (F-statistic):**	1.86e-103
Time:	11:13:15	**Log-Likelihood:**	-60439.
No. Observations:	5285	**AIC:**	1.209e+05
Df Residuals:	5283	**BIC:**	1.209e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.927e+04	348.262	112.769	0.000	3.86e+04	4e+04
C(has_college)[T.1]	1.379e+04	748.215	18.431	0.000	1.23e+04	1.53e+04

Omnibus:	2105.758	**Durbin-Watson:**	1.981
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	9895.006
Skew:	1.899	**Prob(JB):**	0.00
Kurtosis:	8.524	**Cond. No.**	2.56

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

There is no big change of coefficient. After including control variables, the coefficient of has_college is $1.393 * 10^4$, which is very similar to $1.379 * 10^4$, the coefficient of has_college before adding control variables.

Exercise 14

In [36]:
```
smf.wls(
    "annual_earnings ~ C(has_college)+C(simplified_race)+C(female)+C(categoryage)", newcps, weights=newcps["weights"]
).fit().summary()
```

Out[36]:

WLS Regression Results

Dep. Variable:	annual_earnings	**R-squared:**	0.150
Model:	WLS	**Adj. R-squared:**	0.148
Method:	Least Squares	**F-statistic:**	84.48
Date:	Sun, 28 Feb 2021	**Prob (F-statistic):**	1.28e-176
Time:	11:15:06	**Log-Likelihood:**	-61151.
No. Observations:	5285	**AIC:**	1.223e+05
Df Residuals:	5273	**BIC:**	1.224e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.723e+04	770.329	48.327	0.000	3.57e+04	3.87e+04
C(has_college)[T.1]	1.389e+04	721.801	19.241	0.000	1.25e+04	1.53e+04
C(simplified_race)[T.1]	-7211.6384	955.789	-7.545	0.000	-9085.380	-5337.897
C(simplified_race)[T.2]	-4746.0239	989.371	-4.797	0.000	-6685.600	-2806.448
C(simplified_race)[T.3]	-1369.3657	1055.841	-1.297	0.195	-3439.251	700.520
C(female)[T.1]	-8651.8224	600.197	-14.415	0.000	-9828.457	-7475.188
C(categoryage)[T.30]	8432.9390	856.940	9.841	0.000	6752.982	1.01e+04
C(categoryage)[T.40]	1.182e+04	897.973	13.162	0.000	1.01e+04	1.36e+04
C(categoryage)[T.50]	1.207e+04	942.883	12.806	0.000	1.02e+04	1.39e+04
C(categoryage)[T.60]	9452.8634	1160.353	8.147	0.000	7178.092	1.17e+04
C(categoryage)[T.70]	1.419e+04	3102.805	4.574	0.000	8108.601	2.03e+04
C(categoryage)[T.80]	5624.8058	5852.449	0.961	0.337	-5848.417	1.71e+04

Omnibus:	2974.299	**Durbin-Watson:**	1.987
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	40427.906
Skew:	2.412	**Prob(JB):**	0.00
Kurtosis:	15.662	**Cond. No.**	25.1

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

Exercise 15

In [37]:
```
model = dame_flame.matching.DAME(repeats=False, verbose=3, want_pe=True,
early_stop_iterations=2)
model.fit(
    cpsn,
    treatment_column_name="has_college",
    outcome_column_name="annual_earnings",
)
result = model.predict(cpsn)
```

```
Iteration number:  1
        Number of matched groups formed in total:  366
        Unmatched treated units:  648 out of a total of  1150 treated u
nits
        Unmatched control units:  3225 out of a total of  4365 control
units
        Predictive error of covariates chosen this iteration:  0
        Number of matches made in this iteration:  1642
        Number of matches made so far:  1642
        In this iteration, the covariates dropped are:  set()
Iteration number:  2
        Number of matched groups formed in total:  491
        Unmatched treated units:  26 out of a total of  1150 treated un
its
        Unmatched control units:  286 out of a total of  4365 control u
nits
        Predictive error of covariates chosen this iteration:  11988378
30.7153406
        Number of matches made in this iteration:  3561
        Number of matches made so far:  5203
        In this iteration, the covariates dropped are:  frozenset({'nuc
ounty'})
5203 units matched. We stopped before doing iteration number:  2
```

In [38]:
```
result
```

Out[38]:

	female	simplified_race	categoryage	nuclass94	nucounty
4	1	0	50	3	10
5	0	2	30	3	*
7	0	0	60	3	8
8	0	0	40	1	*
9	0	0	30	3	24
...	...	...	...	...	...
11141	0	0	60	3	*
11142	1	3	30	3	247
11143	0	3	50	3	*
11144	0	2	20	3	246
11145	0	0	20	3	99

5203 rows × 5 columns

The coeffcient does not change.