Unifying Data Science

Matching

In [35]:
```
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
import warnings
warnings.simplefilter('ignore')
from scipy import stats
from scipy.stats import ttest_ind
cps = pd.read_stata('https://github.com/nickeubank/MIDS_Data/blob/maste
r/Current_Population_Survey/morg18.dta?raw=true')

cps = cps[cps['lfsr94'] == 'Employed-At Work']
cps = cps[cps['uhourse'] >= 35]

cps['earnhre_dollars'] = cps['earnhre'] / 100
cps['annual_earnings'] = cps['earnhre_dollars'] * cps['uhourse'] *52

cps['female'] = (cps.sex == 2).astype('int')
cps['has_college_educ'] = (cps.grade92 > 43).astype('int')

cps.describe()
```

Out[35]:

	county	smsastat	age	sex	grade92	rac
count	133814.000000	132638.000000	133814.000000	133814.000000	133814.000000	133814.00000
mean	25.735020	1.173932	43.335458	1.440320	41.059680	1.43427
std	61.578816	0.379052	13.335412	0.496427	2.512128	1.27071
min	0.000000	1.000000	16.000000	1.000000	31.000000	1.00000
25%	0.000000	1.000000	32.000000	1.000000	39.000000	1.00000
50%	0.000000	1.000000	43.000000	1.000000	41.000000	1.00000
75%	29.000000	1.000000	54.000000	2.000000	43.000000	1.00000
max	810.000000	2.000000	85.000000	2.000000	46.000000	26.00000

8 rows × 24 columns

Exercise 1: How many observations have a college degree

In [13]:
```
college=cps['has_college_educ'].value_counts()[0]
nocollege=cps['has_college_educ'].value_counts()[1]
print (f'{college} observations have a college degree whereas {nocolleg
e} do not have one.')
```

```
113970 observations have a college degree whereas 19844 do not have on
e.
```

Exercise 2: Raw difference of earnhre_dollars between people with and without college degree.

In [14]:
```
nocollege=cps.groupby('has_college_educ')['annual_earnings'].mean()[0]
college=college=cps.groupby('has_college_educ')['annual_earnings'].mean
()[1]
print (f'The raw mean difference in earnings among employees with and wi
thout college degrees is {college-nocollege:.2f} dollars.')
```

```
The raw mean difference in earnings among employees with and without co
llege degrees is 23461.58 dollars.
```

Exercise 3: Select the covariates that may be correlated with the treatment and dependent variables, use these covariates fit a logistic model to obtain propensity score.

In [15]:
```
from pymatch.Matcher import Matcher
cps=cps[['age','smsastat','female','race','marital','annual_earnings','h
as_college_educ']]
control=cps[cps.has_college_educ==0]
treatment=cps[cps.has_college_educ==1]
```

In [16]:
```
m=Matcher(control,treatment, yvar="has_college_educ", exclude=["annual_e
arnings"])
np.random.seed(20)
m.fit_scores(balance=True, nmodels=100)
```

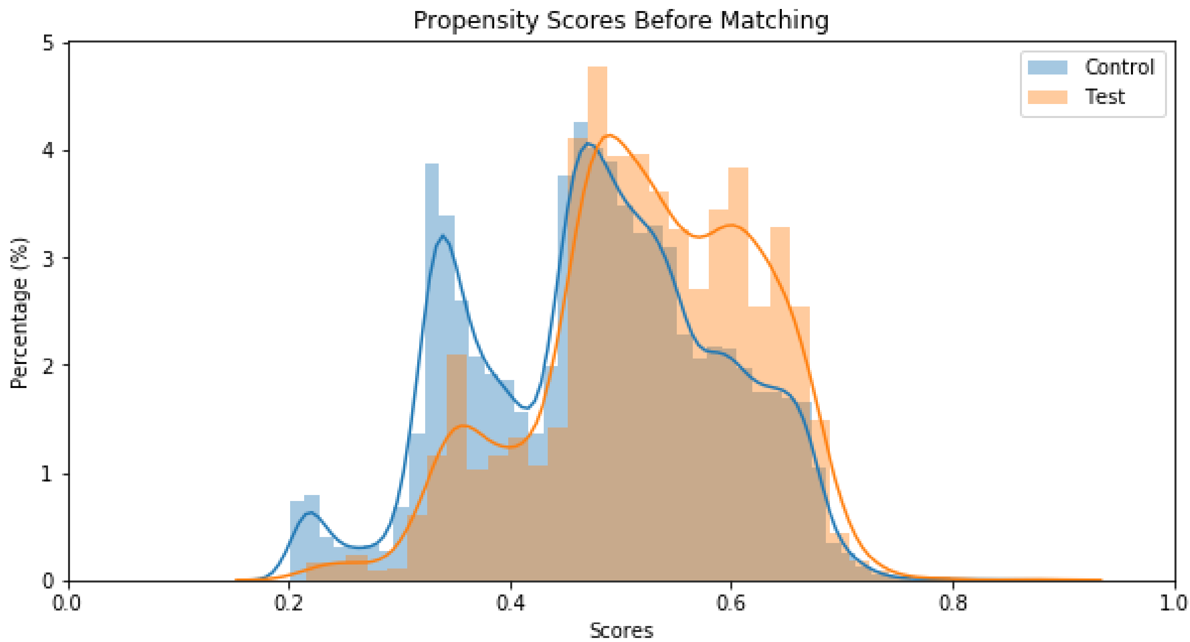
```
Formula:
has_college_educ ~ age+smsastat+female+race+marital
n majority: 62146
n minority: 2992
Fitting Models on Balanced Samples: 100\100
Average Accuracy: 58.45%
```

The average accuracy of 58.45% indicates separability within the dataset, therefore, matching procedure is justified.

In [25]:
```
#Assign a propensity score to each model in a data set.
m.predict_scores()
```

Exercise 4: Evaluate the common support of the treated and control groups.

In [26]:
```
m.predict_scores()
m.plot_scores()
```



The graph suggests that treatment and control have a good common support. In other words, treated and control groups distributions are relatively close to each other.

Exercise 5: K:1 Matching

With K:1 matching we match one observation from the majority group (people with college education) to each record in the minority group with replacement.

In [57]:
```
m.match(method="min", nmatches=1)
m.assign_weight_vector()
#Showing first 10 rows of the matched data.
matched=m.matched_data
```

In [58]:
```
matched.head(10)
```

Out[58]:

	record_id	weight	age	smsastat	female	race	marital	annual_earnings	has_college_educ
0	12	1.000000	50	1.0	1	1	4	28080.0	0
1	169	1.000000	53	1.0	1	1	3	16640.0	0
2	568	1.000000	29	2.0	0	1	1	74880.0	0
3	621	0.500000	55	1.0	1	2	7	31200.0	0
4	621	0.500000	55	1.0	1	2	7	31200.0	0
5	734	1.000000	35	1.0	0	8	7	52000.0	0
6	911	1.000000	47	1.0	0	4	7	72217.6	0
7	934	0.166667	38	1.0	1	1	7	18720.0	0
8	934	0.166667	38	1.0	1	1	7	18720.0	0
9	934	0.166667	38	1.0	1	1	7	18720.0	0

Exercise 6: t-test between the treatment and control group using the matched data

Before matching:

In [59]:
```
cps.groupby("has_college_educ").mean()
```

Out[59]:

	county	smsastat	age	sex	grade92	race	ethnic	ma
has_college_educ								
0	25.428867	1.186091	42.982013	1.430552	40.470282	1.418663	2.516019	3.35
1	27.493348	1.104421	45.365400	1.496422	44.444769	1.523937	3.567848	2.64

2 rows × 23 columns

After matching:

In [60]:
```
matched.groupby("has_college_educ").mean()
```

Out[60]:

	record_id	weight	age	smsastat	female	race	marital	a
has_college_educ								
0	54553.830214	0.309158	44.021056	1.128008	0.579211	1.542112	3.113636	
1	123944.965241	1.000000	43.986297	1.129011	0.578877	1.550468	3.105949	

Mean values from the treatment and control now indeed are closer to each other.

In [61]:
```
#Recreate treatment and control groups on the matched data.
matched_treatment=matched[matched.has_college_educ==0]
matched_control=matched[matched.has_college_educ==1]
```

In [62]:
```
for name in treatment.columns.values:
    print (name, ':', stats.ttest_ind(np.array(matched_treatment[name]),
np.array(matched_control[name])), '\n\n')
```

```
age : Ttest_indResult(statistic=0.10704339030915748, pvalue=0.914758146
5812646)


smsastat : Ttest_indResult(statistic=-0.11586562806034369, pvalue=0.907
7629261691433)


female : Ttest_indResult(statistic=0.026179269624625797, pvalue=0.97911
52238928138)


race : Ttest_indResult(statistic=-0.2387050753374385, pvalue=0.81134245
76898773)


marital : Ttest_indResult(statistic=0.11286631627847621, pvalue=0.91014
0328580261)


annual_earnings : Ttest_indResult(statistic=-26.246760034247384, pvalue
=8.305036688671084e-144)


has_college_educ : Ttest_indResult(statistic=-inf, pvalue=0.0)
```

P-values do not suggest a statistically significant difference between the treatment and control groups **accept** annual earnings and, obviously, has_college_educ, variable.

Exercise 7: regression models to estimate the effect of college education

In [63]:
```
#1. OLS model, including only the treatment variable (annual_earnings)
smf.ols('annual_earnings~has_college_educ', matched).fit().summary()
```

Out[63]:

OLS Regression Results

Dep. Variable:	annual_earnings	**R-squared:**	0.103
Model:	OLS	**Adj. R-squared:**	0.103
Method:	Least Squares	**F-statistic:**	688.9
Date:	Fri, 28 Feb 2020	**Prob (F-statistic):**	8.31e-144
Time:	11:16:16	**Log-Likelihood:**	-70702.
No. Observations:	5984	**AIC:**	1.414e+05
Df Residuals:	5982	**BIC:**	1.414e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	4.197e+04	598.562	70.126	0.000	4.08e+04	4.31e+04
has_college_educ	2.222e+04	846.495	26.247	0.000	2.06e+04	2.39e+04

Omnibus:	2617.916	**Durbin-Watson:**	1.485
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	14809.884
Skew:	2.042	**Prob(JB):**	0.00
Kurtosis:	9.536	**Cond. No.**	2.62

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

In [73]:
```
#2. OLS model, including the treatment variable and covariates.
smf.ols('annual_earnings~has_college_educ+age+smsastat+female+race+marit
al', matched).fit().summary()
```

Out[73]:

OLS Regression Results

Dep. Variable:	annual_earnings	**R-squared:**	0.124
Model:	OLS	**Adj. R-squared:**	0.123
Method:	Least Squares	**F-statistic:**	140.7
Date:	Fri, 28 Feb 2020	**Prob (F-statistic):**	2.37e-167
Time:	11:21:57	**Log-Likelihood:**	-70633.
No. Observations:	5984	**AIC:**	1.413e+05
Df Residuals:	5977	**BIC:**	1.413e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.586e+04	2457.986	18.657	0.000	4.1e+04	5.07e+04
has_college_educ	2.222e+04	837.127	26.539	0.000	2.06e+04	2.39e+04
age	128.8665	34.697	3.714	0.000	60.847	196.886
smsastat	-2170.4206	1259.138	-1.724	0.085	-4638.785	297.944
female	-6926.8894	851.291	-8.137	0.000	-8595.728	-5258.051
race	-161.6258	312.267	-0.518	0.605	-773.782	450.531
marital	-914.2866	164.841	-5.546	0.000	-1237.434	-591.139

Omnibus:	2686.706	**Durbin-Watson:**	1.487
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	16439.034
Skew:	2.074	**Prob(JB):**	0.00
Kurtosis:	9.980	**Cond. No.**	285.

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

In [74]:
```
#3. A weighted least squared model, including only the treatment variabl
e, using the weight obtained by propensity score matching.
weight=matched['weight'].values
smf.wls('annual_earnings~has_college_educ', matched, weights=weight).fit
().summary()
```

Out[74]:

WLS Regression Results

Dep. Variable:	annual_earnings	**R-squared:**	0.081
Model:	WLS	**Adj. R-squared:**	0.080
Method:	Least Squares	**F-statistic:**	523.8
Date:	Fri, 28 Feb 2020	**Prob (F-statistic):**	3.33e-111
Time:	11:22:28	**Log-Likelihood:**	-72332.
No. Observations:	5984	**AIC:**	1.447e+05
Df Residuals:	5982	**BIC:**	1.447e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	3.992e+04	926.936	43.066	0.000	3.81e+04	4.17e+04
has_college_educ	2.427e+04	1060.585	22.887	0.000	2.22e+04	2.64e+04

Omnibus:	2760.985	**Durbin-Watson:**	1.928
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	22175.789
Skew:	2.035	**Prob(JB):**	0.00
Kurtosis:	11.507	**Cond. No.**	3.90

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

In [75]:
```
#4. A weighted least squared model, including the treatment variable and covariates, using the weight obtained by propensity score matching.
smf.wls('annual_earnings~has_college_educ+age+smsastat+female+race+marital', matched, weights=weight).fit().summary()
```

Out[75]:

WLS Regression Results

Dep. Variable:	annual_earnings	**R-squared:**	0.103
Model:	WLS	**Adj. R-squared:**	0.102
Method:	Least Squares	**F-statistic:**	114.2
Date:	Fri, 28 Feb 2020	**Prob (F-statistic):**	6.72e-137
Time:	11:23:12	**Log-Likelihood:**	-72258.
No. Observations:	5984	**AIC:**	1.445e+05
Df Residuals:	5977	**BIC:**	1.446e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	3.771e+04	2840.164	13.277	0.000	3.21e+04	4.33e+04
has_college_educ	2.44e+04	1079.415	22.607	0.000	2.23e+04	2.65e+04
age	220.9950	35.888	6.158	0.000	150.642	291.348
smsastat	-1064.7045	1268.709	-0.839	0.401	-3551.832	1422.423
female	-7858.2861	902.785	-8.704	0.000	-9628.070	-6088.502
race	-65.8164	295.999	-0.222	0.824	-646.081	514.448
marital	-575.2715	175.970	-3.269	0.001	-920.237	-230.306

Omnibus:	2754.685	**Durbin-Watson:**	1.929
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	23322.397
Skew:	2.009	**Prob(JB):**	0.00
Kurtosis:	11.797	**Cond. No.**	313.

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

Summary: From the four models we have built, we observe that after adding the propensity score weights, the value of the college degree on the annual earnings increases. From 22220 dollars of difference, it grows up to 24400 dollars of difference. In other words, a person with a college degree is likely to earn, on average, 24400 dollars in annual income more than someone without a college degree.