Introduction

Education, playing a more and more important role in human society nowadays, enables students as well as society to make progress by obtaining technical ability, professional skills and spiritual strength. One of the most the important part of Meiji restoration in Japan is education. Meiji restoration promoted significantly social progress, helped Japan develop rapidly, became the only Asian country that kept democracy & independence in the modern times and built the foundation of being a powerful developed country for Japan. Nearly all the countries treat education as the foundation of development and their future competitiveness.

Academic performance is one of the outcomes of education that can be observed easily. It is affected by many factors such as different gender, school type, the conditions for learning, parental education level, family annual income, etc.

Propensity score is one of the methods that can estimate the effect of receiving treatment when it is not suitable to assign treatments to subjects randomly. Propensity score matching refers to the pairing of treatment and control units with similar values on the propensity score; and possibly other covariates (the characteristics of participants); and the discarding of all unmatched units.

The school-level dataset was acquired from government data website. It is used to test whether a certain intervention (lunch type and test preparation course respectively in this case) can really enhance students' academic performance in the total score of math, reading and writing. Conducting propensity score matching to eliminate bias from confounding variables (gender, parental education level, race/ethnicity), while investigating if standard lunch and test preparation course made progress in students' reading and mathematics performance.

Eliminating confounding variables can help us make more accurate conclusion if an intervention really brings higher academic achievement to students. With accurate conclusion, education department will be able to make better decision on whether implementing the intervention or not.

Description of dataset

This dataset was obtained from Kaggle for a predictive analysis project. It contains 1000 rows and 9 columns. The columns are gender, race_ethinicity, parental_level_of education, lunch, test_preparation_course, math_score, reading_score and writing_score.

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

Table 1 Description of Scores

Data Cleaning and Wrangling

The steps of data cleaning and wrangling are as follows:

1. Applied *dropna()* function to remove rows containing missing values
2. Dropped rows with value beyond the 3 standard deviations of writing_score column. Repeated the same process for reading_score and math_score columns respectively.

```
# Remove the score outliers from df
df = df[np.abs(df['math_score']-df['math_score'].mean()) <= (3*df['math_score'].std())]
df = df[np.abs(df['reading_score']-df['reading_score'].mean()) <= (3*df['reading_score'].std())]
df_cleaned = df[np.abs(df['writing_score']-df['writing_score'].mean()) <= (3*df['writing_score'].std())]
df_cleaned
```

Figure 1 Outliers removal

3. For those columns whose values are hierarchical, assigned increasing consecutive integers start from 0, to the values with level from low to high. Take parental education level for

example, some high school, associate's degree, some college, bachelor, master correspond to the integers 0 to 5.

4. For those columns whose values are non-hierarchical, applied function *pd.get_dummies* on the columns to quantify the values and deleted one column of the results to avoid dummy trap.

```
# quantify race_ethnicity
dummies1 = pd.get_dummies(df_cleaned.race_ethnicity)
dummies1
```

Figure 2 Non-hierarchical Column Quantifying

	group A	group B	group C	group D	group E
0	0	1	0	0	0
1	0	0	1	0	0
2	0	1	0	0	0
3	1	0	0	0	0
4	0	0	1	0	0
5	0	1	0	0	0
6	0	1	0	0	0

Table 2 Quantified race_ethnicity

5. Merged the newly created columns in step 4 to the dataframe and then deleted the original non-hierarchical columns.

Exploratory Data Analysis

In the first part of the project, I used standard lunch and reduced lunch to classify the dataset into 2 groups, control and test respectively. Besides, I set total score of the 3 subjects as the dependent variable to study the relation between lunch types and students' academic performance. In this case, the other variables were considered covariates, and propensity score matching was applied to minimize the influence from the covariates such as gender, parental education level. The purpose of this project is to eliminate the selecting bias in contrast test.

```
# create test and control groups by lunch type
test = df_cleaned[df_quant.lunch == "free/reduced"]
control = df_cleaned[df_quant.lunch == "standard"]
test['lunch'] = 1
control['lunch'] = 0
```

Figure 3 Test and Control Groups Classification

Before doing inferential statistics to figure out what the outcome tells us, data wrangling and propensity score matching of the test and control groups are needed. For data wrangling, I followed the steps: (1) Apply *df.dropna()* to remove the missing values in the dataframe df. (2) Remove the rows in df which contain value beyond the range of 3 standard deviation of ether one of the 3 score columns. (3) For the columns whose values can be ranked by number directly, assign consecutive integer to each type of value based on the level from low to high. (4) For those columns that cannot be quantify directly, use one hot encoding and delete one column to avoid dummy variable trap afterwards. (5) Import Matcher and classify test and control group by lunch type free/reduced and standard respectively. (6) Set the 3 types of score as dependent variables, lunch types as independent variables and others as covariates. (7) Conduct PSM using *Pymatch* module and we can obtain the pairs of matched test and control groups for inferential statistics.

```
# Matcher
m = Matcher(test, control, yvar="lunch", exclude=['math_score','reading_score','writing_score'])
```

```
Formula:
lunch ~ gender+race_ethnicity+parental_level_of_education+test_preparation_course
n majority: 643
n minority: 350
```

```
# for reproducibility
np.random.seed(20170925)
m.fit_scores(balance=True, nmodels=350)
```

```
Fitting Models on Balanced Samples: 350\350
Average Accuracy: 53.52%
```

Figure 4 Propensity Score Matching

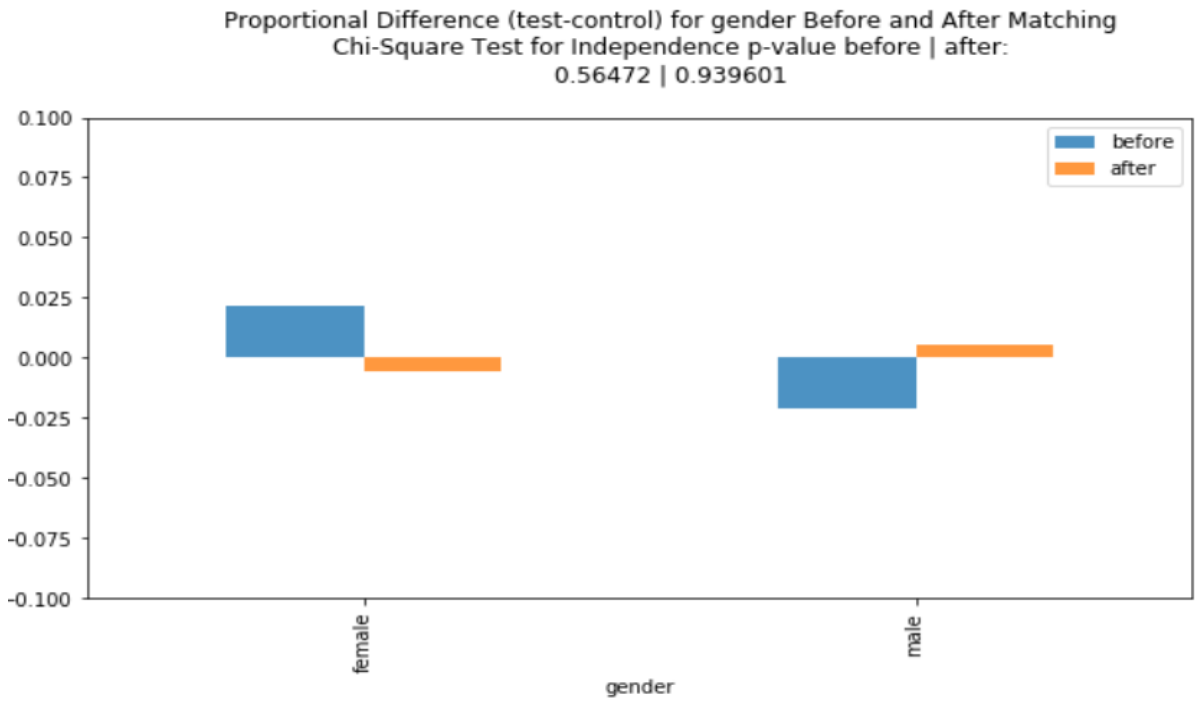

Figure 5 Proportion Difference for Gender

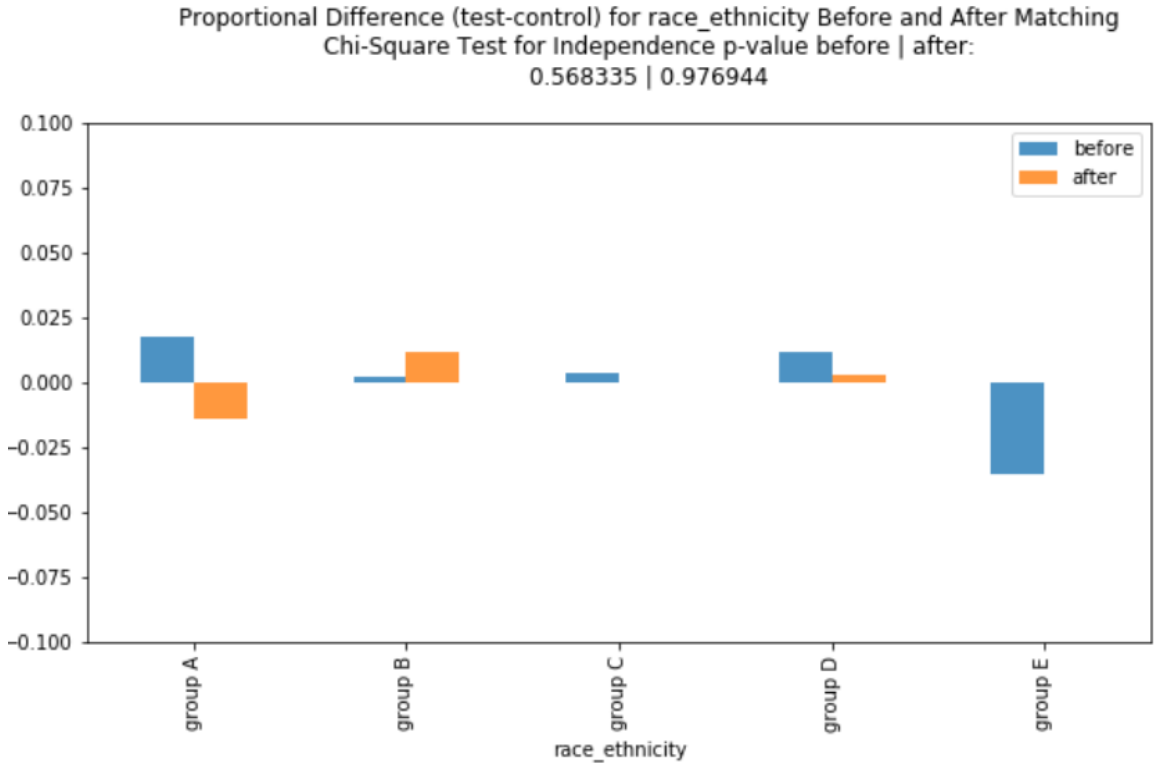

Figure 6 Proportional Difference for Race

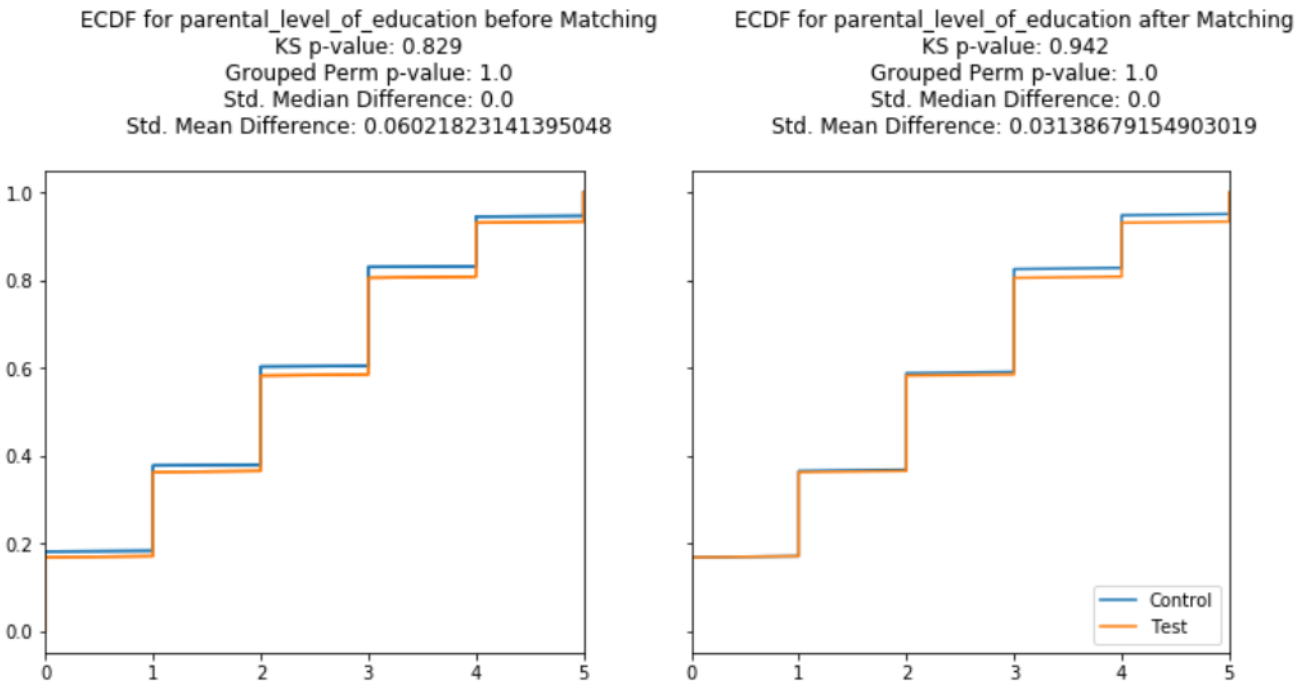

Figure 7 Assessment of Propensity Score Matching

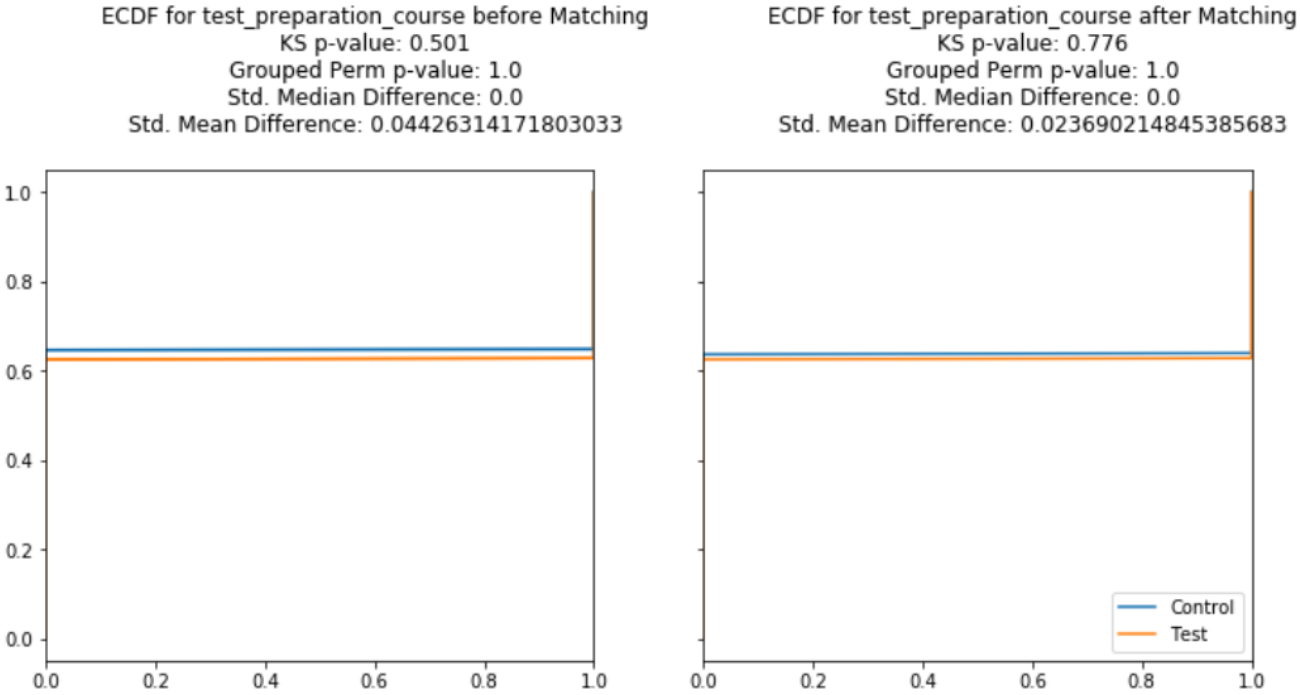

Figure 8 Assessment of Propensity Score Matching

In inferential statistic part, there are two methods to estimate the outcome.

Obtained the mean total score, standard deviation of the group with standard lunch and the group with reduced lunch respectively. Miu_1, miu_2, $sigma_1$ and $sigma_2$. Besides, obtained miu_1-miu_2 and standard deviation of mean difference $sigma_{1-2}$.

1. Assumed significance level at alpha = 5%, null hypothesis miu_1-miu_2 = 0 and alternative hypothesis: $miu_1 – miu_2 > 0$. Since(miu_1-miu_2) = 18.15, which is out of 95% CI, I rejected null hypothesis in favor of alternative hypothesis that $miu_1 > miu_2$.
2. Imported *ttest_ind* from *scipy.stats* and conducted t test based on miu_1, miu_2, $sigma_1$ and $sigma_2$. The small result also shows that I should reject null hypothesis in favor of alternative hypothesis.

```
from scipy.stats import ttest_ind
# Method 2 t test
lunch = np.random.normal(miu1,sigma1,1000)
no_lunch = np.random.normal(miu2,sigma2,1000)
t_test = ttest_ind(lunch,no_lunch)
t_test
# p_value < alpha. We can reject Ho in favor of Ha
```

Figure 5 Method 2 of Inferential Statistics

After hypothesis test, calculated correlation by importing *linregress* form *scipy.stats* and we can see the correlation r-value between lunch type and total score is -0.216, which means when lunch value is 1(test group: free/reduced lunch), students get lower score, and when lunch type is 0, students get higher score. we want to compare the difference between means of total score of these two groups. The means and standards deviation can both be obtained from our total_score columns. Thus, hypothesis test on difference of means can be an effective way and t test is chosen in our case. The steps are as follows: (1)From *scipy.stats* import *ttest_ind*. (2) Calculate means and standard deviations of test and control groups. (3) Set lunch by assigning miu_1, $sigma_1$ and size 1000 to *np.random.normal* respectively. Set no_lunch by filling in *np.random.normal* with miu_2, $sigma_2$, 1000. (4) Conduct t test and obtain p-value. (5) Compare p-value with significant level 5%.

In the second part of the project, I classified test and control groups by test_preparation_course (complete:1, none:0) and repeated the whole process mentioned above. In this case, we can see the p-value is significantly smaller than significance level 5%, which implies we should reject null hypothesis in favor of alternative hypothesis that the test group(test preparation course completed)

has higher mean total score. The r-value is 0.30, meaning that taking the course can affect the total score positively.

In a word, the results of the project show that standard lunch and test preparation course can help students perform better in academic field.