pyMatcher Example

From github (https://github.com/benmiroglio/pymatch)

Required:

```
pip install pymatch
pip install statsmodels
pip install seaborn
```

and corrections to code:

```
9 from . import functions as uf
in c:\users\jg114\anaconda3\envs\tf2\lib\site-packages\pymatch\__init__.py
```

and

```
3 from . import functions as uf

517 # return (y == preds).sum() * 1.0 / len(y)
518 return (y.to_numpy().T == preds).sum() * 1.0 / len(y)
in c:\users\jg114\anaconda3\envs\tf2\lib\site-packages\pymatch\Matcher.py
```

In [1]:

```
import warnings
warnings.filterwarnings('ignore')
from pymatch.Matcher import Matcher
import pandas as pd
import numpy as np

%matplotlib inline
```

In [2]:

```
path = "D:\\20201007\\pymatch\\misc\\loan.csv"
fields = \
[
    "loan_amnt",
    "funded_amnt",
    "funded_amnt_inv",
    "term",
    "int_rate",
    "installment",
    "grade",
    "sub_grade",
    "loan_status"
]

data = pd.read_csv(path)[fields]
```

In [3]:

```
data[:2]
```

Out[3]:

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grad
0	16000	16000	16000.0	36 months	18.85	585.29	D	D
1	14000	14000	14000.0	36 months	12.42	467.82	B	B

In [4]:

```
test = data[data.loan_status == "Default"]
control = data[data.loan_status == "Fully Paid"]
test['loan_status'] = 1
control['loan_status'] = 0
```

In [5]:

```
m = Matcher(test, control, yvar="loan_status", exclude=[])
```

```
Formula:
loan_status ~ loan_amnt+funded_amnt+funded_amnt_inv+term+int_rate+installm
ent+grade+sub_grade
n majority: 20000
n minority: 2000
```

In [6]:

```
# for reproducibility
np.random.seed(20170925)

m.fit_scores(balance=True, nmodels=100)
```

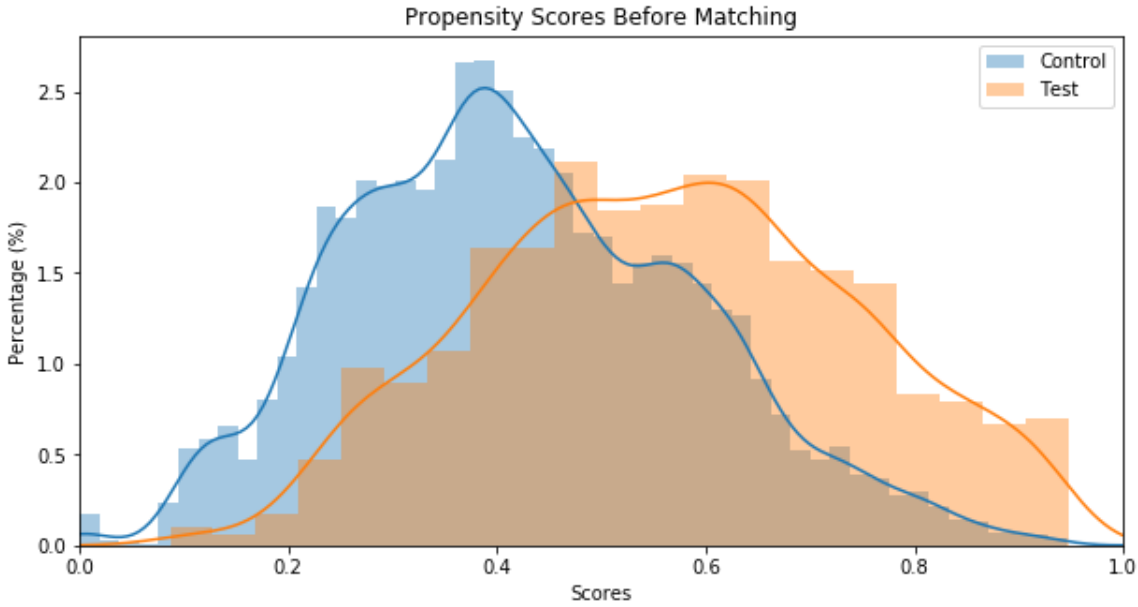
```
Fitting Models on Balanced Samples: 100\100
Average Accuracy: 65.54%
```

In [7]:

```
m.predict_scores()
```

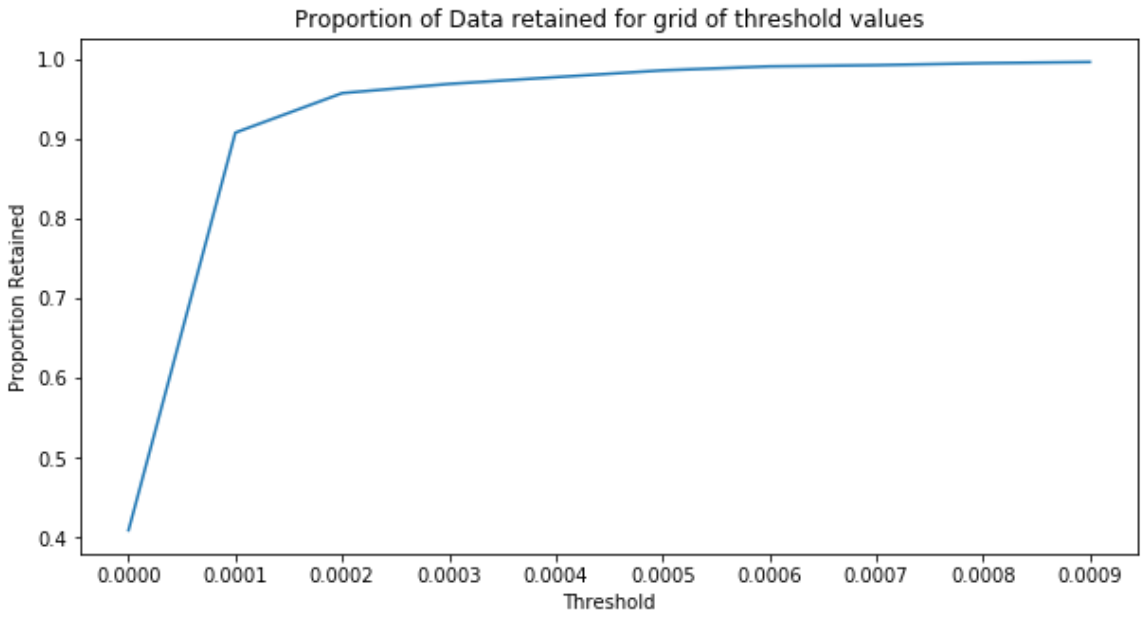
In [8]:

```
m.plot_scores()
```



In [9]:

```
m.tune_threshold(method='random')
```



In [10]:

```
m.match(method="min", nmatches=1, threshold=0.0001)
```

In [11]:

```
m.record_frequency()
```

Out[11]:

	freq	n_records
0	1	3196
1	2	235
2	3	63
3	4	21
4	5	6
5	6	4
6	7	1

In [12]:

```
m.assign_weight_vector()
```

In [13]:

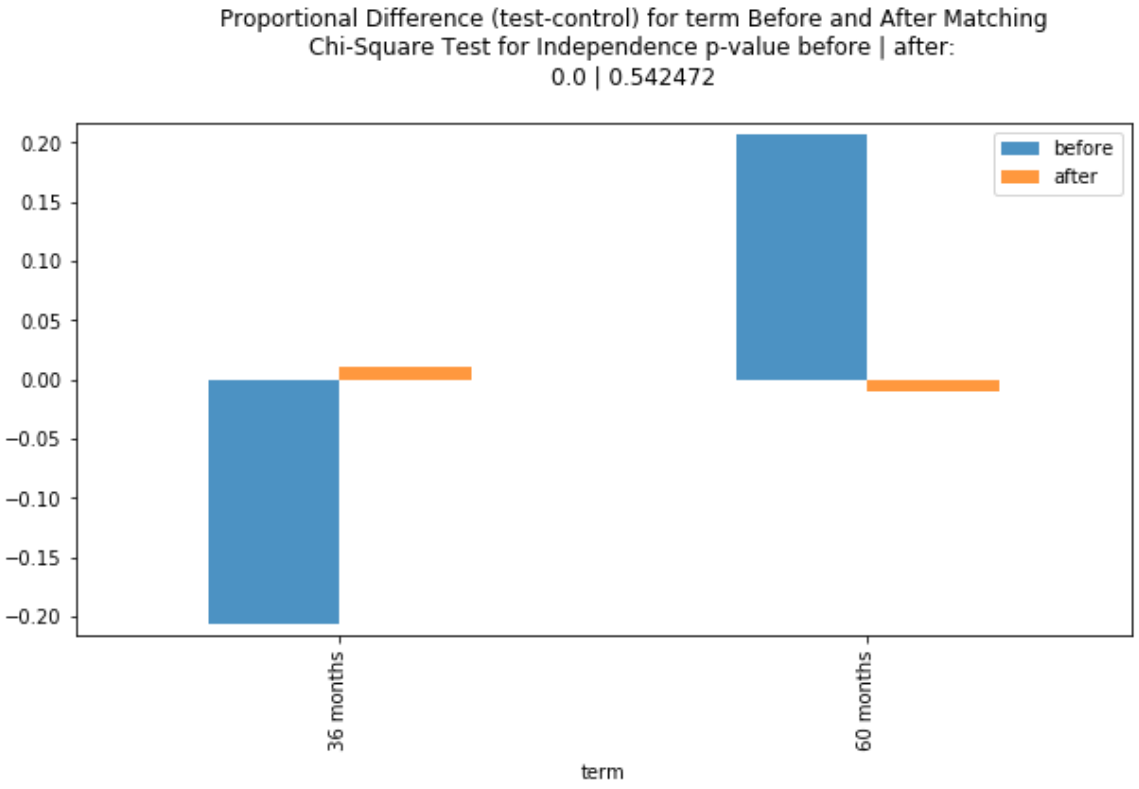
```
m.matched_data.sort_values("match_id").head(6)
```

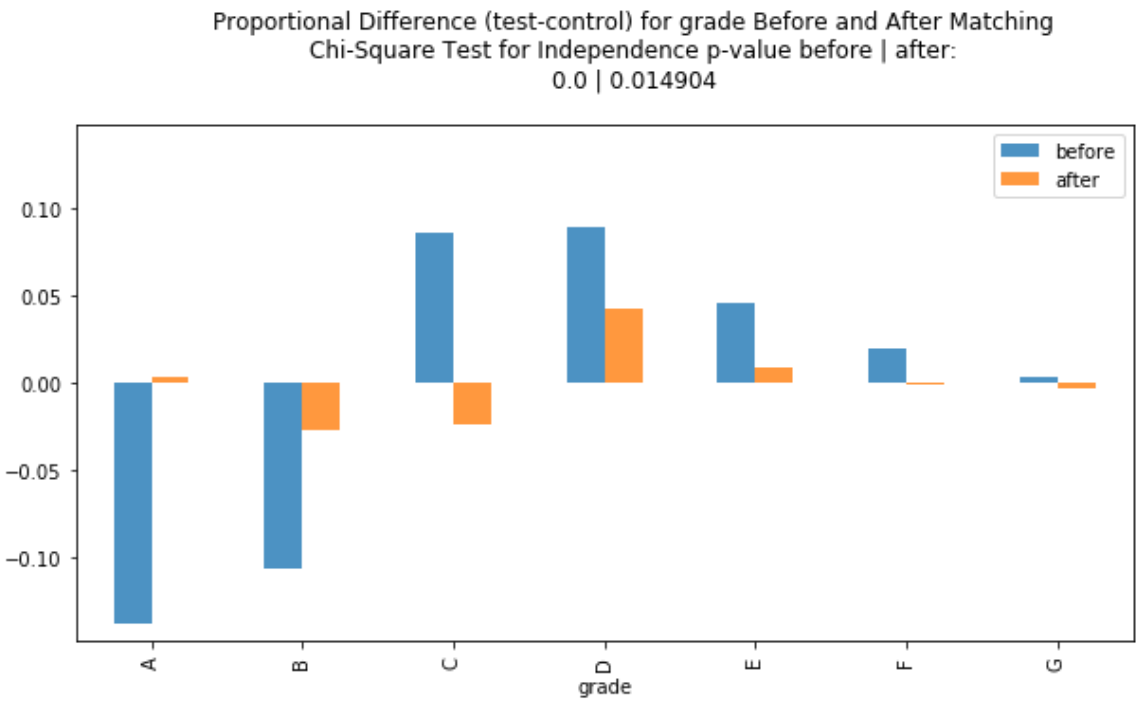
Out[13]:

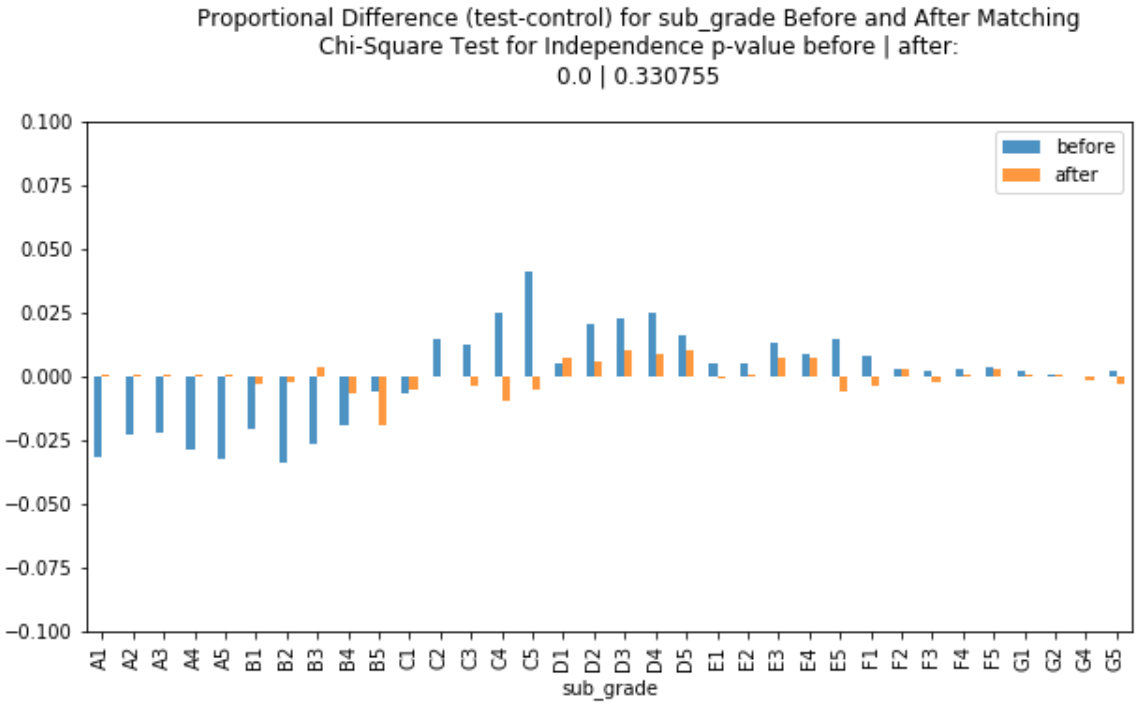
	record_id	weight	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installr
0	0	1.0	12000	12000	12000.0	60 months	12.59	27
3331	15254	1.0	12000	12000	12000.0	60 months	13.99	27
1	1	1.0	15000	15000	15000.0	36 months	10.75	48
2838	10360	1.0	15000	15000	15000.0	36 months	10.75	48
2	2	1.0	30000	30000	30000.0	36 months	6.49	91
2574	7577	1.0	4000	4000	4000.0	36 months	10.15	12

In [14]:

```
categorical_results = m.compare_categorical(return_table=True)
```


Proportional Difference (test-control) for term Before and After Matching
Chi-Square Test for Independence p-value before | after:
0.0 | 0.542472
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
before
after
36 months
60 months
term


Proportional Difference (test-control) for grade Before and After Matching
Chi-Square Test for Independence p-value before | after:
0.0 | 0.014904
0.10
0.05
0.00
−0.05
−0.10
before
after
A
B
C
D
E
F
G
grade



In [15]:

```
categorical_results
```

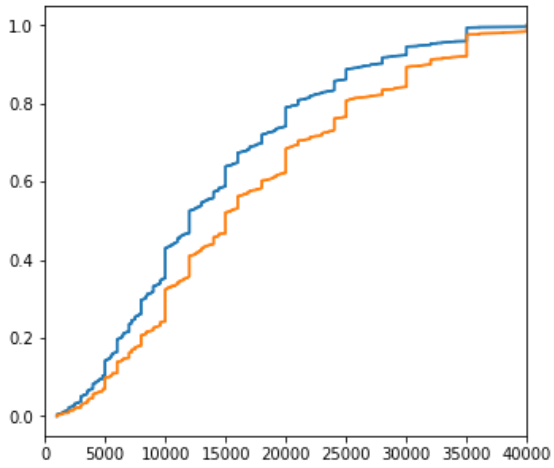
Out[15]:

	var	before	after
0	term	0.0	0.542472
1	grade	0.0	0.014904
2	sub_grade	0.0	0.330755

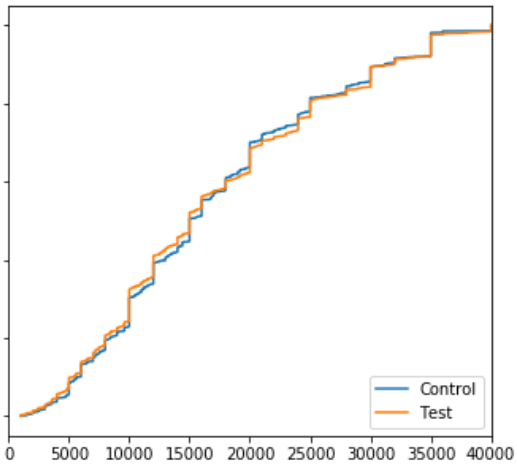
In [16]:

```
cc = m.compare_continuous(return_table=True)
```

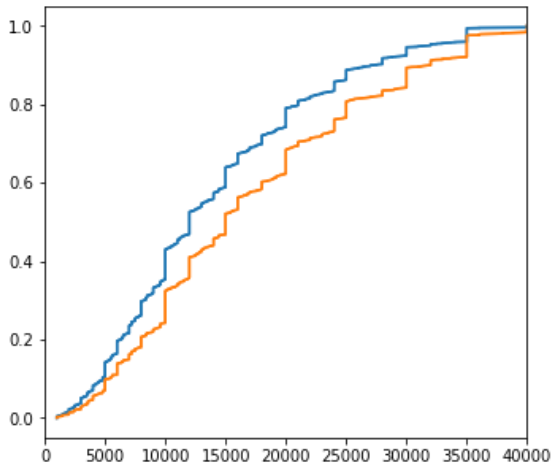
ECDF for loan_amnt before Matching
KS p-value: 0.0
Grouped Perm p-value: 0.001
Std. Median Difference: 0.3422768318963559
Std. Mean Difference: 0.3228463466109862



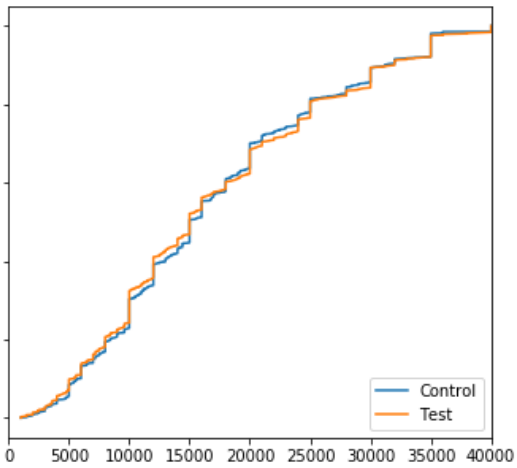
ECDF for loan_amnt after Matching
KS p-value: 0.447
Grouped Perm p-value: 0.18
Std. Median Difference: 0.0
Std. Mean Difference: -0.005341648202572528



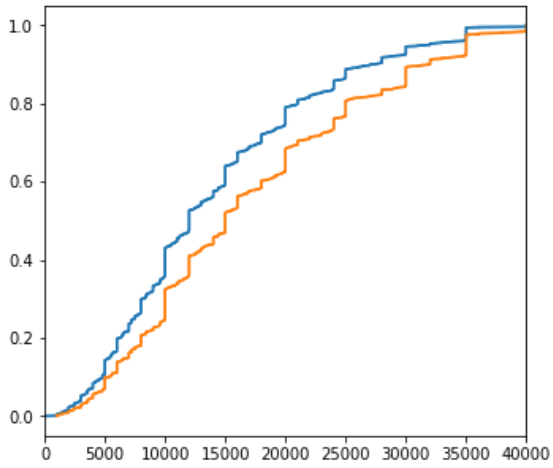
ECDF for funded_amnt before Matching
KS p-value: 0.0
Grouped Perm p-value: 0.001
Std. Median Difference: 0.3424481673626475
Std. Mean Difference: 0.323858368597366



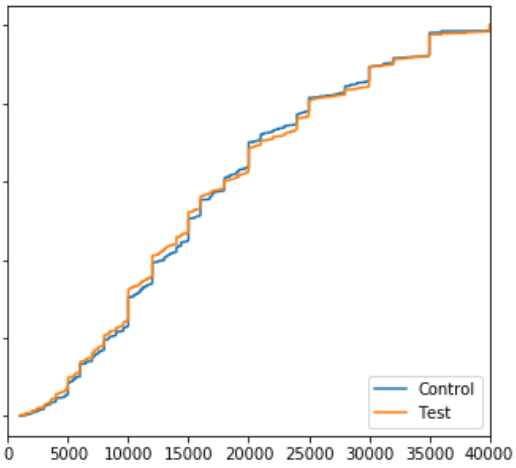
ECDF for funded_amnt after Matching
KS p-value: 0.428
Grouped Perm p-value: 0.158
Std. Median Difference: 0.0
Std. Mean Difference: -0.005341648202572528



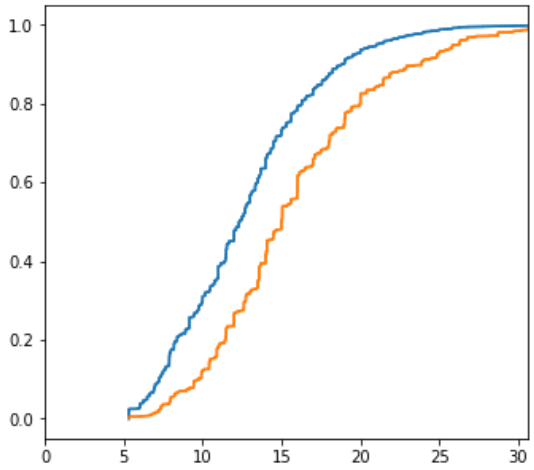
ECDF for funded_amnt_inv before Matching
KS p-value: 0.0
Grouped Perm p-value: 0.046
Std. Median Difference: 0.34222668205239587
Std. Mean Difference: 0.3258599221414641



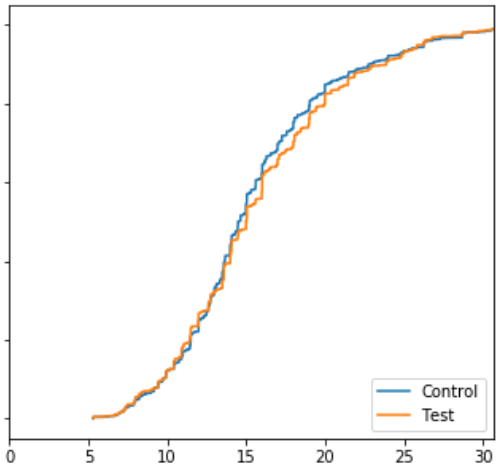
ECDF for funded_amnt_inv after Matching
KS p-value: 0.434
Grouped Perm p-value: 0.178
Std. Median Difference: 0.0
Std. Mean Difference: -0.005390631213614981

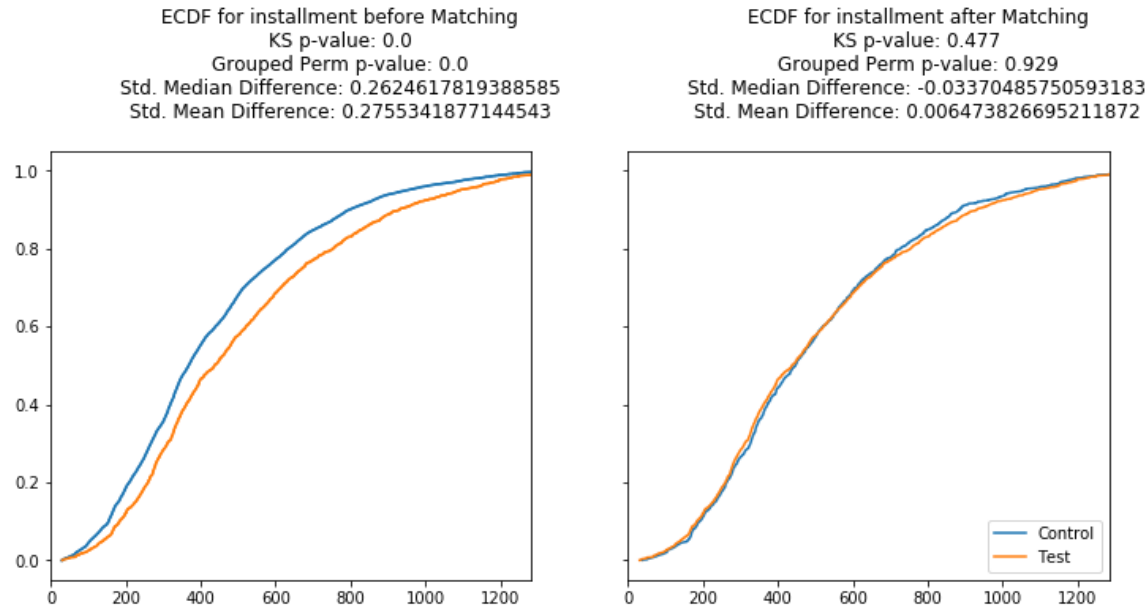


ECDF for int_rate before Matching
KS p-value: 0.0
Grouped Perm p-value: 0.0
Std. Median Difference: 0.5633106715835519
Std. Mean Difference: 0.6435446748641817



ECDF for int_rate after Matching
KS p-value: 0.009
Grouped Perm p-value: 0.089
Std. Median Difference: 0.0960143350440022
Std. Mean Difference: 0.046475738878049594





In [17]:

```
cc
```

Out[17]:

	var	ks_before	ks_after	grouped_chisqr_before	grouped_chisqr_after	std_me
0	loan_amnt	0.0	0.447	0.001	0.180	
1	funded_amnt	0.0	0.428	0.001	0.158	
2	funded_amnt_inv	0.0	0.434	0.046	0.178	
3	int_rate	0.0	0.009	0.000	0.089	
4	installment	0.0	0.477	0.000	0.929	