ABIDE_analysis

January 23, 2023

[1]:
```
import numpy as np
import pandas as pd
from dask import dataframe as dd
import matplotlib.pyplot as plt
from scipy.stats import kendalltau
from scipy.stats import rankdata
import fastHDMI as mi
```

1 Calculate MI for ABIDE data

2 Calculation for age

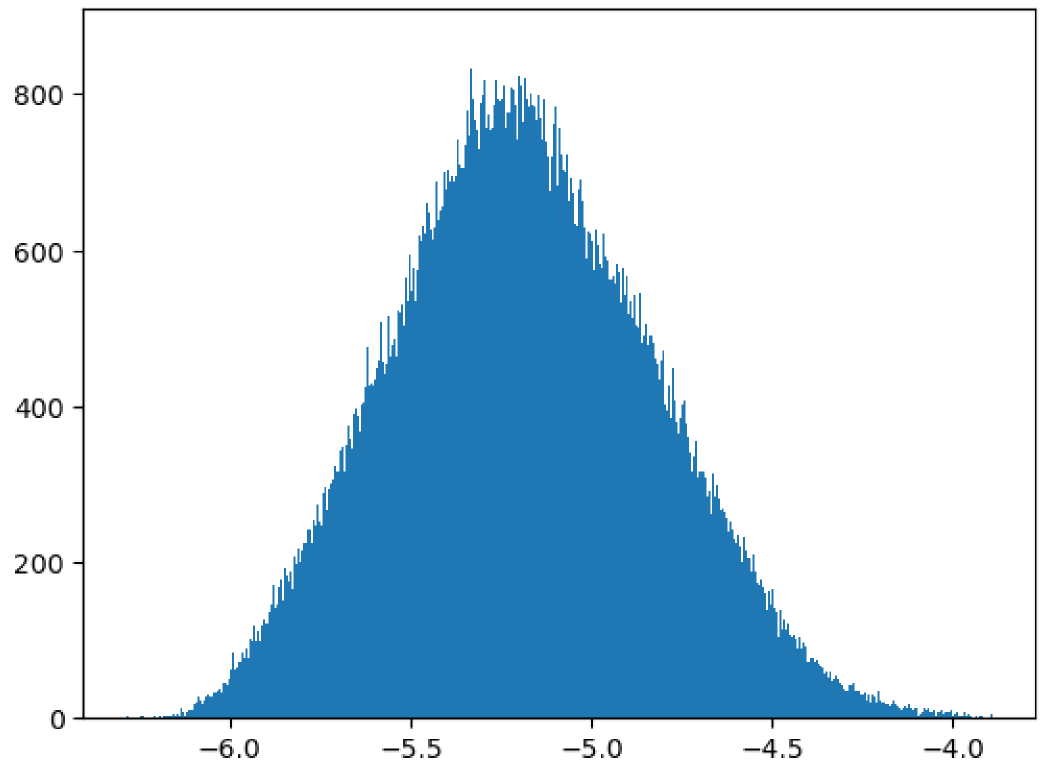
2.1 this block is only to be run on Compute Canada

[ ]:
```
csv_file = r"/home/kyang/projects/def-cgreenwo/abide_data/
 ↪abide_fs60_vout_fwhm0_lh_SubjectIDFormatted_N1050_nonzero_withSEX.csv"
# abide = pd.read_csv(csv_file, encoding='unicode_escape', engine="c")
abide = dd.read_csv(csv_file, sample=1250000)

# _abide_name = abide.columns.tolist()[1:]
_abide_name = list(abide.columns)[1:]

# print(_abide_name)

# we don't inlcude age and sex in the screening since they should always be␣
 ↪included in the model
abide_name = [_abide_name[-3]] + _abide_name[1:-3]

np.save(r"/home/kyang/ABIDE_columns", _abide_name[1:-3])

# so that the left first column is the outcome and the rest columns are areas

mi_output = mi.continuous_filter_csv_parallel(csv_file,
                                              _usecols=abide_name,
                                              csv_engine="c",
                                              sample=1250000)
np.save(r"/home/kyang/ABIDE_age_MI_output", mi_output)
```

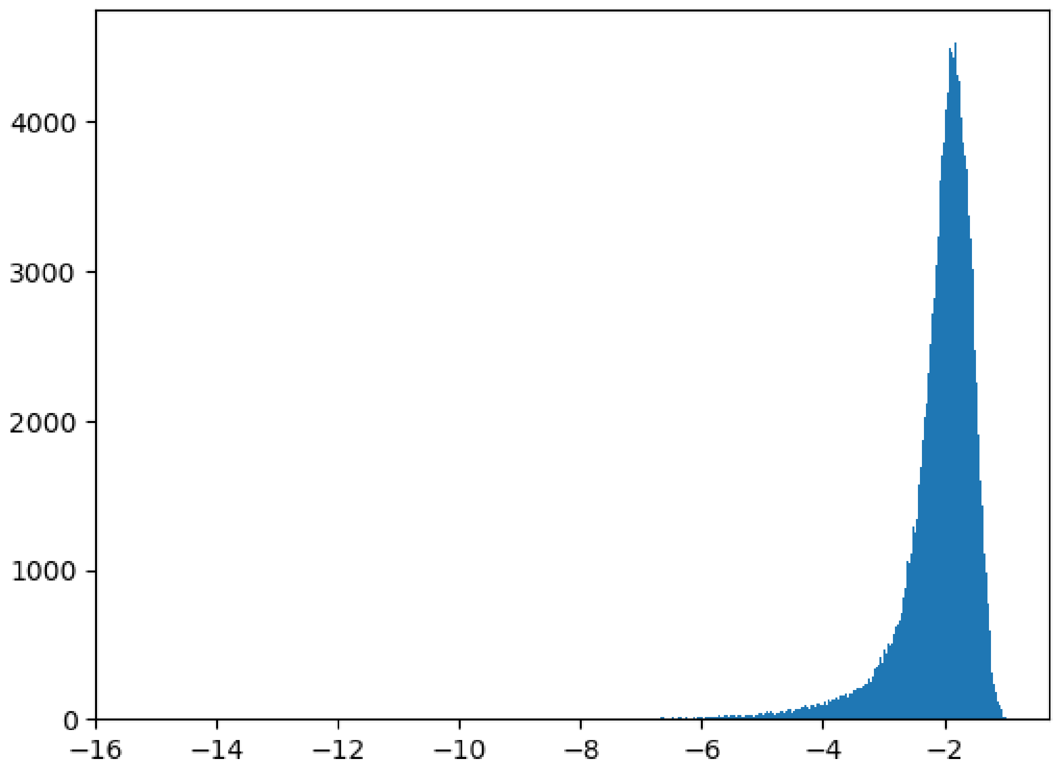
```
pearson_output = mi.Pearson_filter_csv_parallel(csv_file,
                                                _usecols=abide_name,
                                                csv_engine="c",
                                                sample=1250000)
np.save(r"/home/kyang/ABIDE_age_Pearson_output", pearson_output)
```

3 Plots

[2]:
```
abide_mi = np.load(r"./ABIDE_age_MI_output.npy")
plt.hist(np.log(abide_mi), 500)
plt.show()
```



[3]:
```
abide_pearson = np.load(r"./ABIDE_age_Pearson_output.npy")
plt.hist(np.log(np.abs(abide_pearson)), 500)
plt.show()
```

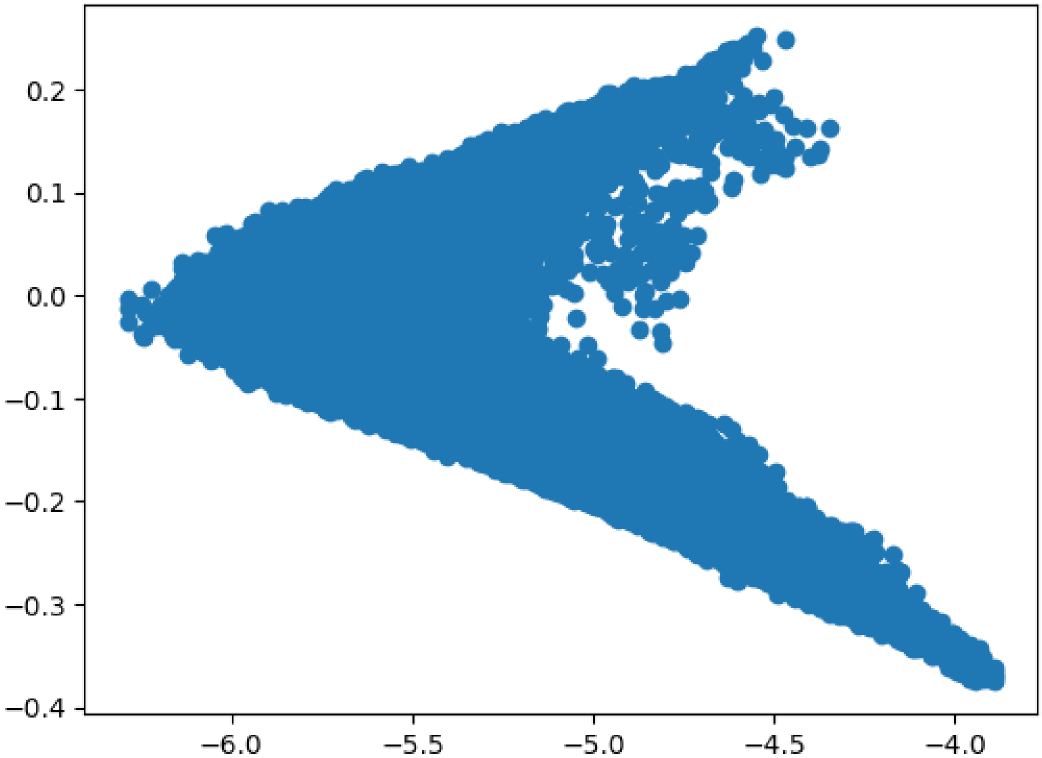


3.1 Comparing two ranking with Kendall's τ

The results show that the two ranking by mutual information and Pearson's correlation vary greatly by Kendall's tau – I also tried the Pearson's correlation between two ranking (not that I should do this) and the correlation is also very small.

So in summary, the two ranking vary greatly.

[4]:
```
plt.plot(np.log(abide_mi), abide_pearson, 'o')
plt.show()
# keep this, add different selections
# PREDICT AGE
```



[5]:
```python
print("Kendall's tau: \n",
      kendalltau(rankdata(-abide_mi), rankdata(-np.abs(abide_pearson))))
print("Pearson's correlation: \n",
      np.corrcoef(rankdata(-abide_mi), rankdata(-np.abs(abide_pearson))))
```

```
Kendall's tau:
 KendalltauResult(correlation=0.8122050157867158, pvalue=0.0)
Pearson's correlation:
 [[1.         0.94631777]
 [0.94631777 1.        ]]
```

4 Calculate MI for ABIDE data

5 Calculation for diagnosis outcome

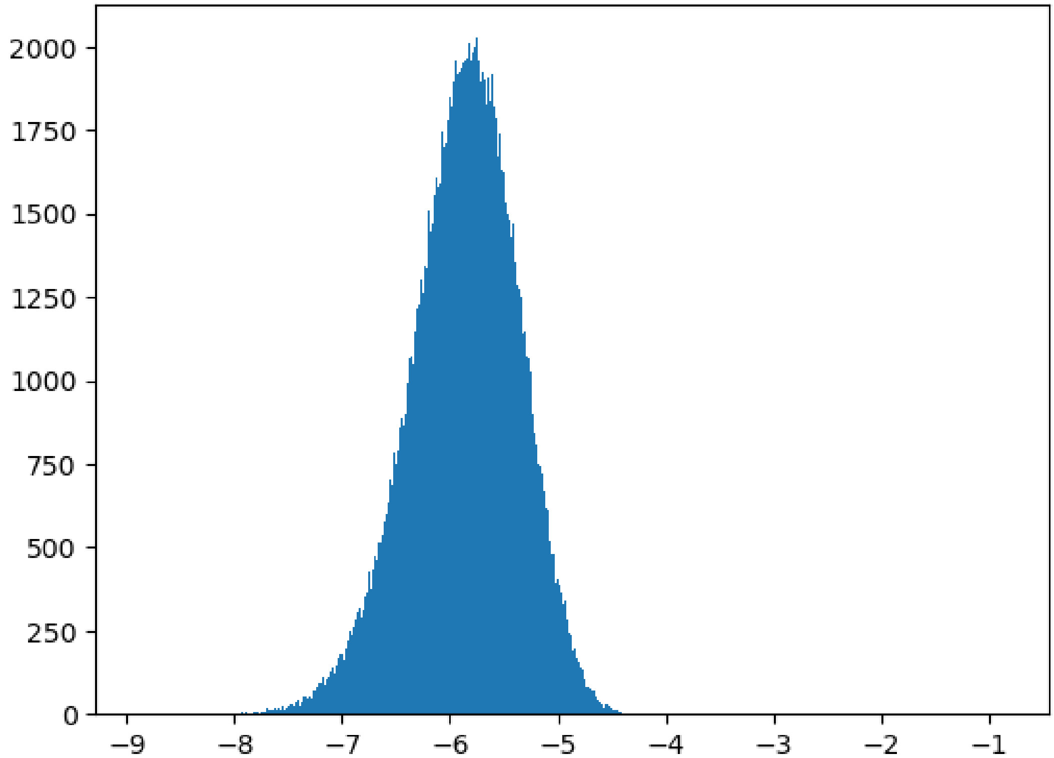
5.1 this block is only to be run on Compute Canada

[ ]:
```python
csv_file = r"/home/kyang/projects/def-cgreenwo/abide_data/
 ↪abide_fs60_vout_fwhm0_lh_SubjectIDFormatted_N1050_nonzero_withSEX.csv"
# abide = pd.read_csv(csv_file, encoding='unicode_escape', engine="c")
abide = dd.read_csv(csv_file, sample=1250000)
```

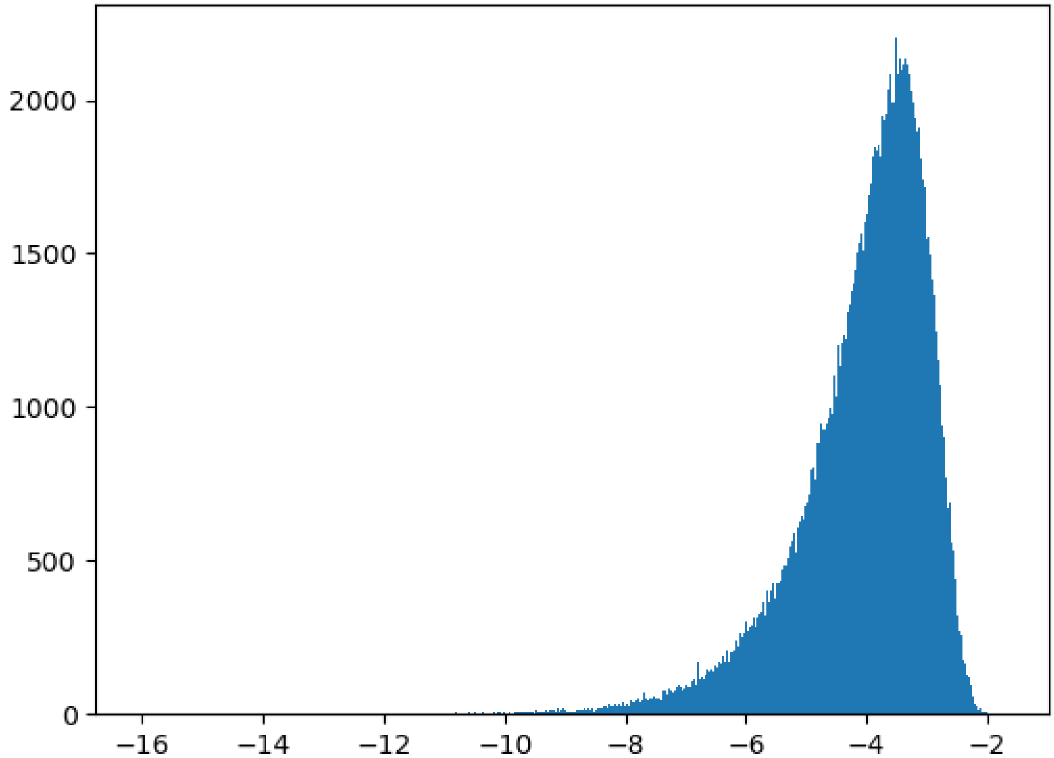
```python
# _abide_name = abide.columns.tolist()[1:]
_abide_name = list(abide.columns)[1:]

# print(_abide_name)

# we don't inlcude age and sex in the screening since they should always be␣
↪included in the model
abide_name = [_abide_name[-1]] + _abide_name[1:-3]
# so that the left first column is the outcome and the rest columns are areas

mi_output = mi.binary_filter_csv_parallel(csv_file,
                                          _usecols=abide_name,
                                          csv_engine="c",
                                          sample=1250000)
np.save(r"/home/kyang/ABIDE_diagnosis_MI_output", mi_output)

pearson_output = mi.Pearson_filter_csv_parallel(csv_file,
                                                _usecols=abide_name,
                                                csv_engine="c",
                                                sample=1250000)
np.save(r"/home/kyang/ABIDE_diagnosis_Pearson_output", pearson_output)
```

6 Plots

[6]:
```python
abide_mi = np.load(r"./ABIDE_diagnosis_MI_output.npy")
plt.hist(np.log(abide_mi), 500)
plt.show()
```



[7]:
```
abide_pearson = np.load(r"./ABIDE_diagnosis_Pearson_output.npy")
plt.hist(np.log(np.abs(abide_pearson)), 500)
plt.show()
```

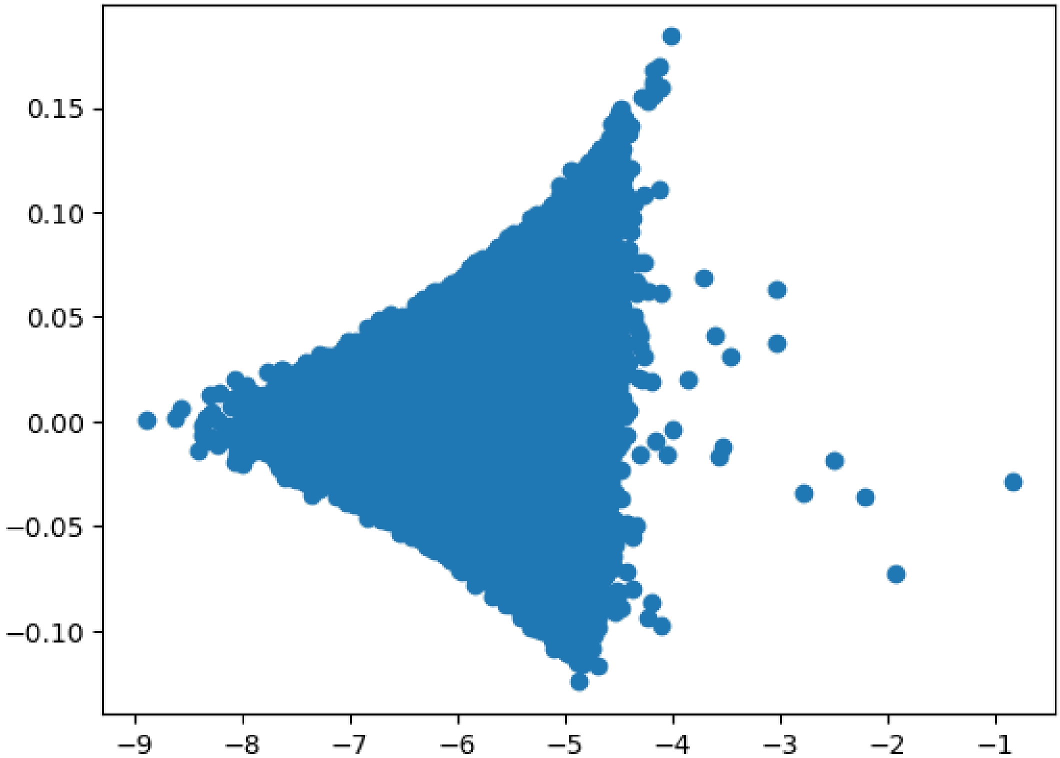


6.1 Comparing two ranking with Kendall's τ

The results show that the two ranking by mutual information and Pearson's correlation vary greatly by Kendall's tau – I also tried the Pearson's correlation between two ranking (not that I should do this) and the correlation is also very small.

So in summary, the two ranking vary greatly.

[8]:
```
plt.plot(np.log(abide_mi), abide_pearson, 'o')
plt.show()
# keep this, add different selections
# PREDICT AGE
```



[9]:
```
print("Kendall's tau: \n",
      kendalltau(rankdata(-abide_mi), rankdata(-np.abs(abide_pearson))))
print("Pearson's correlation: \n",
      np.corrcoef(rankdata(-abide_mi), rankdata(-np.abs(abide_pearson))))
```

```
Kendall's tau:
 KendalltauResult(correlation=0.2500737738382302, pvalue=0.0)
Pearson's correlation:
 [[1.         0.36310015]
 [0.36310015 1.        ]]
```

[ ]: