**Execution**

1. **import and get the data into our DataFrames:**

import pandas as pd

from pathlib import Path

import fuzzymatcher

import recordlinkage

1. **Example using fuzzymatcher**

hospital\_accounts =

pd.read\_csv('https://github.com/chris1610/pbpython/raw/master/data/hospital\_account\_info.csv')

hospital\_reimbursement = pd.read\_csv('https://raw.githubusercontent.com/chris1610/pbpython/master/data/hospital\_reimbursement.csv')

1. **Print all row from Dataframe:**

hospital\_accounts.head( )

hospital\_reimbursement.head( )

1. **Columns to match on from df\_left & df\_right**

left\_on = ["Facility Name", "Address", "City", "State"]

right\_on = ["Provider Name", "Provider Street Address", "Provider City", "Provider State"]

1. **Perform the match**

matched\_results = fuzzymatcher.fuzzy\_left\_join(hospital\_accounts,hospital\_reimbursement, left\_on, right\_on,left\_id\_col='Account\_Num',right\_id\_col='Provider\_Num')

1. **Print Match**

matched\_results.head( )

1. **Reorder the columns to make viewing easier**

cols = [ "best\_match\_score", "Facility Name", "Provider Name", "Address", "Provider Street Address","Provider City", "City", "Provider State", "State"]

1. **Let's see the best matches**

matched\_results[cols].sort\_values(by=['best\_match\_score'], ascending=False).head(5)

1. **Now the worst matches**

matched\_results[cols].sort\_values(by=['best\_match\_score'], ascending=True).head(5)

1. **Look at the matches around 1**

matched\_results[cols].query("best\_match\_score <= .80").sort\_values(

by=['best\_match\_score'], ascending=**False**).head(5)

matched\_results[cols].query("best\_match\_score <= 1").sort\_values(by=['best\_match\_score'], ascending=**False**).head(10)