Jack Deschler

CS105 Final Project

In [1]:
```
# imports
import pandas as pd
import numpy as np
import copy
```

In [2]:
```
# import data
data = pd.read_csv("Data/overall_data.csv")
data.head()
```

Out[2]:

	concentration	house	year	gender	dist
0	AAAS: African American Studies	Adams	Junior	NaN	AH
1	AAAS: African American Studies	Cabot	NaN	NaN	AH
2	AAAS: African Studies	Quincy	Senior	Male	AH
3	AAAS: African Studies	Quincy	Sophomore	Male	AH
4	Anthro: Archaeology	Currier	Junior	NaN	AH

In [3]:
```
# clean up NaNs
np.random.seed()
for i in range(len(data)):
    row = data.iloc[i]
    # first, impute male with probability 0.5, female with probability
0.5
    if pd.isnull(row['gender']):
        row.gender = 'Male' if np.random.rand() > 0.5 else 'Female'
    # now, impute class year with probability 0.33 for each
    if pd.isnull(row['year']):
        row.year = np.random.choice(['Senior', 'Junior', 'Sophomore'])

# apparently Adams gave me data on a freshman, so drop that observatio
n...
data = data[data['year'] != 'Freshman']

# one more thing, take out everything after colons, as they are subfie
lds of same field
def colons(s):
    if ':' in s:
        return s.split(':')[0]
    return s
data['concentration'] = data.apply(lambda row: colons(row['concentrati
on']), axis = 1)

data.head()
```

Out[3]:

	concentration	house	year	gender	dist
0	AAAS	Adams	Junior	Male	AH
1	AAAS	Cabot	Senior	Male	AH
2	AAAS	Quincy	Senior	Male	AH
3	AAAS	Quincy	Sophomore	Male	AH
4	Anthro	Currier	Junior	Male	AH

In [4]:
```
data.describe()
```

Out[4]:

	concentration	house	year	gender	dist
count	2743	2743	2743	2743	2743
unique	61	7	3	2	6
top	Economics	Quincy	Junior	Male	SS
freq	352	482	973	1388	1030

In [42]:
```
# this will be the function that does all the lumping
```

```
#   df -> the data, in a pandas dataframe
#   columns -> a list of columns that will be generalized on
#     this is in order of which to generalize first
#     will first generalize on columns[0], then columns[1], etc.
#   k -> level of k-anonymity, default to 5
#   outfile -> the file to write the steps to
#   returns the generalized dataframe
def lumper(df, columns, k = 5, outfile = 'out.txt'):
    df = copy.deepcopy(df)
    current = 0
    current_col = columns[current]
    def new_lump_col(x):
        acc = ""
        final = columns[-1]
        for c in columns[:-1]:
            acc += str(x[c])
            acc += "*"
        acc += str(x[final])
        return acc

    def lump(s, t1, t2):
        if s == t1 or s == t2:
            return str(t1) + "-" + str(t2)
        return s

    # set up file
    out = open(outfile, 'w')
    out.write("GENERALIZING BASED (IN ORDER) ON:\n")
    for c in columns:
        out.write("  " + c + "\n")
    out.write("\n\nSTEPS TO GENERALIZE\n")
    while(True):
        # PRE-LUMPING
        # before starting, see if we need to go to the next column to
generalize on
        if (len(df[current_col].value_counts()) < 2):
            current += 1
            current_col = columns[current]
        # create the new columns we need
        df['lump_col'] = df.apply(lambda row: new_lump_col(row), axis
= 1)
        df['freq'] = df.groupby('lump_col')['lump_col'].transform('cou
nt')
        # check if finished
        if df['freq'].min() >= k:
            break

        # LUMPING
        # find the two lowest
        temp = copy.deepcopy(df)
        min1 = df['freq'].idxmin()
        row1 = df.iloc[min1]
        lump1 = row1[current_col]
```

```
        temp = temp[temp[current_col] != lump1]
        row2 = temp.nsmallest(1, 'freq').iloc[0]
        lump2 = row2[current_col]
        # ensure that concentrations only get generalized inside overa
rching category
        #   make exceptions for NO (general studies/undeclared) as the
y could be anyone
        if current_col == 'concentration':
            while row1['dist'] != row2['dist'] and row1['dist'] != 'NO
' and row2['dist'] != 'NO':
                temp = temp[temp[current_col] != lump2]
                row2 = temp.nsmallest(1, 'freq').iloc[0]
                lump2 = row2[current_col]
        out.write("Combine: " + lump1 + " with " + lump2 + "\n")
        # lump them together
        df[current_col] = df.apply(lambda row: lump(row[current_col],
lump1, lump2), axis = 1)
    # finally, return
    out.close()
    return df
```

In [45]:
```
lumped = lumper(data, ['concentration', 'year', 'gender'], k=5, outfil
e = 'conc_year_gender_dist.txt')
lumped.head()
```

Out[45]:

	concentration	house	year	gender	dist	lump_col	freq
0	Visual & Environmental Studies-AAAS-Anthro-Und...	Adams	Junior	Female	AH	Visual & Environmental Studies-AAAS-Anthro-Und...	19
1	Visual & Environmental Studies-AAAS-Anthro-Und...	Cabot	Sophomore	Female	AH	Visual & Environmental Studies-AAAS-Anthro-Und...	23
2	Visual & Environmental Studies-AAAS-Anthro-Und...	Quincy	Senior	Male	AH	Visual & Environmental Studies-AAAS-Anthro-Und...	9
3	Visual & Environmental Studies-AAAS-Anthro-Und...	Quincy	Sophomore	Male	AH	Visual & Environmental Studies-AAAS-Anthro-Und...	19
4	Visual & Environmental Studies-AAAS-Anthro-Und...	Currier	Junior	Female	AH	Visual & Environmental Studies-AAAS-Anthro-Und...	19

In [24]:
```
print(lumped['lump_col'].value_counts())
```

```
Senior-Junior*Male*Quincy        176
Senior-Junior*Male*Adams         164
Senior-Junior*Male*Winthrop      157
Senior-Junior*Female*Eliot       152
Senior-Junior*Female*Quincy      150
Senior-Junior*Female*Adams       143
Senior-Junior*Female*Currier     134
Senior-Junior*Female*Winthrop    131
Senior-Junior*Male*Cabot         127
Senior-Junior*Male*Currier       126
Senior-Junior*Male*Eliot         124
Senior-Junior*Female*Cabot       105
Sophomore*Male*Adams              85
Sophomore*Female*Quincy           79
Sophomore*Female*Eliot            79
Sophomore*Male*Eliot              78
Sophomore*Male*Quincy             77
Sophomore*Female*Cabot            76
Sophomore*Male*Currier            68
Sophomore*Female*Adams            66
Sophomore*Female*Winthrop         66
Senior-Junior*Male*Kirkland       63
Sophomore*Female*Kirkland         58
Sophomore*Male*Kirkland           58
Sophomore*Female*Currier          57
Senior-Junior*Female*Kirkland     50
Sophomore*Male*Winthrop           49
Sophomore*Male*Cabot              49
Name: lump_col, dtype: int64
```