k-Anonymity Library Demo with k=3

In [130…

```
import warnings
warnings.filterwarnings('ignore')
```

In [131…

```
import kAnonymityLib as daio_dpt
import pandas as pd
dai_anonymization = daio_dpt.kAnonymity()
print(dai_anonymization)
```

```
k-Anonymity Class Library with k=3
```

In [132…

```
names = ['age',
 'workclass',
 'fnlwgt',
 'education',
 'education-num',
 'marital-status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'capital-gain',
 'capital-loss',
 'hours-per-week',
 'native-country',
 'income']
dai_anonymization.set_headers(names)
```

In [133…

```
dai_anonymization.read_datafile("adult-all.txt")
```

In [134…

```
df = dai_anonymization.dataframe
print(f"population size = {df.age.size}")
```

```
population size = 48842
```

In [135…

```
age_range = lambda age: ("<= 20" if age <= 20
    else ("21 - 30" if age <= 30
    else ("31 - 40" if age <= 40
    else ("41 - 50" if age <= 50
    else ("51 - 60" if age <= 60
    else ("61 - 70" if age <= 70 else "> 70"))))))
```

In [136…

```
df["age"] = df.apply(lambda x: age_range(x.age), axis=1)
df["workclass"] = df.apply(lambda x: x.workclass.replace(" ",""), axis=1)
df["workclass"] = df.apply(lambda x: "Others" if x.workclass=="?" else x.workclass, ax
df["race"] = df.apply(lambda x: x.race.replace(" ",""), axis=1)
df["education"] = df.apply(lambda x: x.education.replace(" ",""), axis=1)
```

In [137…

```
categorical = ['workclass',
 'education',
 'marital-status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native-country',
 'income',
 'age']
```

In [138…

```
feature_columns = ['race', 'sex', 'age']
```

In [139…

```
dai_anonymization.set_categorial(categorical)
```

In [140…

```
dai_anonymization.set_feature_columns(feature_columns)
```

In [141…

```
dai_anonymization.set_sensitive_column("income")
```

In [142…

```
dd = pd.Series({c: df[c].unique() for c in df})
print(dd)
```

```
age                  ['31 - 40', '41 - 50', '51 - 60', '21 - 30', '...
workclass            ['State-gov', 'Self-emp-not-inc', 'Private', '...
fnlwgt               [77516, 83311, 215646, 234721, 338409, 284582,...
education            ['Bachelors', 'HS-grad', '11th', 'Masters', '9...
education-num        [13, 9, 7, 14, 5, 10, 12, 11, 4, 16, 15, 3, 6,...
marital-status       [' Never-married', ' Married-civ-spouse', ' Di...
occupation           [' Adm-clerical', ' Exec-managerial', ' Handle...
relationship         [' Not-in-family', ' Husband', ' Wife', ' Own-...
race                 ['White', 'Black', 'Asian-Pac-Islander', 'Amer...
sex                  [' Male', ' Female']
Categories (2, object): [...
capital-gain         [2174, 0, 14084, 5178, 5013, 2407, 14344, 1502...
capital-loss         [0, 2042, 1408, 1902, 1573, 1887, 1719, 1762, ...
hours-per-week       [40, 13, 16, 45, 50, 80, 30, 35, 60, 20, 52, 4...
native-country       [' United-States', ' Cuba', ' Jamaica', ' Indi...
income               [' <=50k', ' >50k']
Categories (2, object): ['...
dtype: object
```

In [143…

```
dai_anonymization.partition_dataset()
print( len(dai_anonymization.finished_partitions) )
```

```
67
```

In [144…

```
dai_anonymization.build_anonymized_dataset()
```

In [145…

```
df1 = dai_anonymization.result_df
print(f"total records = {df1.age.size}")
```

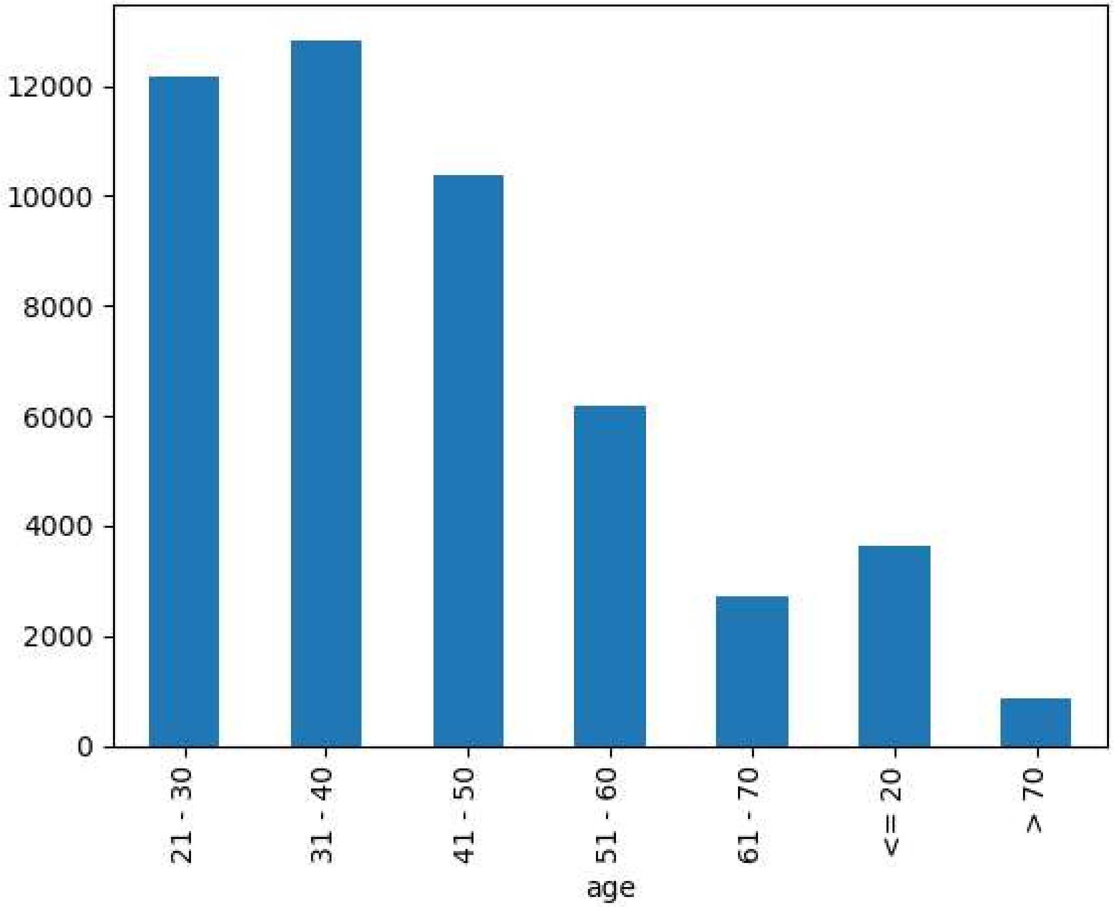
```
total records = 48780
```

df1.to_csv("result.csv")

In [146…

```
df1.groupby("age").size().plot.bar()
print(df1.groupby("age").size())
```

```
age
21 - 30    12170
31 - 40    12838
41 - 50    10363
51 - 60     6201
61 - 70     2726
<= 20       3619
> 70         863
dtype: int64
```



In [147…

```
df2 = dai_anonymization.removed_df
```

In [148…

```
df2.groupby(feature_columns).size()
```

Out[148]:

```
race                 sex     age
Amer-Indian-Eskimo   Female  51 - 60     1
                             61 - 70     4
                             > 70        1
                     Male    61 - 70     2
Asian-Pac-Islander   Female  > 70        1
                     Male    > 70        1
Black                Female  61 - 70     2
                     Male    <= 20       1
Other                Female  41 - 50     2
                             61 - 70     2
                             > 70        1
                     Male    41 - 50    38
                             61 - 70     2
                             > 70        1
White                Female  <= 20       2
                     Male    <= 20       1
dtype: int64
```

In [155…

```
df2_cols = ['age',
 'workclass',
 'education',
 'marital-status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native-country',
 'income']

d3 = df2.groupby(feature_columns)
group_list = list(d3.groups.keys())
records = []
for x in [ x for x in group_list if d3.get_group(x).age.count() > 2]:
        y=d3.get_group(x)
        z=y.to_dict()
        dd = {}
        for w in list(z['workclass']):
            for v in df2_cols:
                dd[v]=z[v][w]
            records.append(dd)
df3 = pd.DataFrame(records)
df3.groupby(feature_columns).size()
```

Out[155]:

```
race                 sex     age
Amer-Indian-Eskimo   Female  61 - 70     4
Other                Male    41 - 50    38
dtype: int64
```

In [156…

```
df3.head()
```

Out[156]:

	age	workclass	education	marital-status	occupation	relationship	race	sex	native-country	income
0	61 - 70	State-gov	HS-grad	Widowed	Adm-clerical	Unmarried	Amer-Indian-Eskimo	Female	United-States	<=50k
1	61 - 70	State-gov	HS-grad	Widowed	Adm-clerical	Unmarried	Amer-Indian-Eskimo	Female	United-States	<=50k
2	61 - 70	State-gov	HS-grad	Widowed	Adm-clerical	Unmarried	Amer-Indian-Eskimo	Female	United-States	<=50k
3	61 - 70	State-gov	HS-grad	Widowed	Adm-clerical	Unmarried	Amer-Indian-Eskimo	Female	United-States	<=50k
4	41 - 50	Private	7th-8th	Never-married	Transport-moving	Not-in-family	Other	Male	Puerto-Rico	<=50k

Conclusion : Partitioning will lost some true data

In [157…

```
dai_anonymization.finished_partitions = []
```

In [158…

```
dai_anonymization.build_anonymized_dataset()
```

In [159…

```
df1 = dai_anonymization.result_df
print(f"total records = {df1.age.size}")
```

```
total records = 48822
```

In [160…

```
df2 = dai_anonymization.removed_df
print(f"removed records = {df2.age.size}")
```

```
removed records = 20
```

In [161…

```
df2.groupby(feature_columns).size()
```

Out[161]:

```
race                sex     age
Amer-Indian-Eskimo   Female  51 - 60    1
                             > 70       1
                     Male    61 - 70    2
Asian-Pac-Islander   Female  > 70       1
                     Male    > 70       1
Black                Female  61 - 70    2
                     Male    <= 20      1
Other                Female  41 - 50    2
                             61 - 70    2
                             > 70       1
                     Male    61 - 70    2
                             > 70       1
White                Female  <= 20      2
                     Male    <= 20      1
dtype: int64
```

In [162…

```
d3 = df2.groupby(feature_columns)
group_list = list(d3.groups.keys())
[ x for x in group_list if d3.get_group(x).age.count() > 2]
```

Out[162]:

```
[]
```

Conclusion : Anonymization works better without partitioning but runs slower

In [ ]:

```
```