Card Catalog Summary Characteristics

Let's get into some of the characteristics of the dataset we created from the digitized Rubenstein Library card catalog. For these simple checks, we used the Python pandas package for easily working with the dataset.

Number of Collections

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv("main_file_dataset.csv")

print(df['Coll_head'].value_counts())
```

```
0    34413
1    15952
Name: Coll_head, dtype: int64
```

As we can see, there are **15,952 collections** detected by our code in the dataset. The other 34,413 cards have been classified as narrative. Let's see how this compares to the number of unique authors.

Unique Authors

In [4]:

```
authors = 1
for i in range(1, len(df)):
    if df.iloc[i]['Name'] != df.iloc[i-1]['Name']:
        authors += 1

print(authors)
```

```
10752
```

By comparing each author name to the name before it, it looks like we have **10,752 unique authors** present in the card catalog. Some of these are people, but others are organizations, let's see how many of each we have.

Author Identity: Person vs. Organization

In [5]:

```
people = 0
orgs = 0

for index, row in df.iterrows():
    if row['Coll_head'] == 1:
        entity = str(row['Name'])
        if not "," in entity or not "," in entity.split(" ")[0] or "Ministry" in entity or "England" in entity
            orgs += 1
        else:
            people += 1

print("People: ", people)
print("Organizations: ", orgs)
```

```
People:  12930
Organizations:  3022
```

When looking at each of the 15,952 collections, about **12,930 of them are associated with a person and 3,022 are associated with an organization**. People comprise about 81% of the collections authors.

Collection Outliers

In [6]:

```
# Find 5 longest collections
coll_len = 0
top_five = [[0,1]]

def addToList(num, length):
    if len(top_five) > 4:
        top_five.pop()
    i = len(top_five)-1
    while i > -1:
        if top_five[i][1] < length:
            i -= 1
        else: break
    top_five.insert(i+1, [num, length])
    return

for index, row in df.iterrows():
    if index == 0: continue
    if row['Coll_head'] == 0:
        coll_len += 1
    else:
        if coll_len > top_five[len(top_five)-1][1]:
            addToList(row['Collection']-1, coll_len)
        coll_len = 1

print(top_five)
```

```
[[3496, 342], [981, 319], [9535, 306], [719, 236], [7468, 223]]
```

In [10]:

```
top_five_coll = [3496, 981, 9535, 719, 7468]
indices = [2167, 3173, 12176, 25895, 32392]

for index in indices:
    print(df.iloc[[index]][['Name']])
    print("----------------------------")
```

```
                          Name
2167  Ball, William Watts
----------------------------
                                 Name
3173  Bedinger- Dandridge Family
----------------------------
                                            Name
12176  Dawson, Francis Warrington, I and II
----------------------------
                      Name
25895  Kirby, Ephraim
----------------------------
                            Name
32392  Munford-Ellis Family
----------------------------
```

Here we have our top five longest collections. In fifth, at 223 cards is the Munford-Ellis Family Collection. In fourth, at 236 cards is Kirby Ephraham's Collection. In third, at 306 cards is Francis Warrington Dawson, I and II's Collection. In second, at 319 cards is the Bedinger-Dandridge Family Collection. And in first, at 342 cards is William Watts Ball's Collection.