Extract Names by Drawer

Within each drawer, we attempted to glean the authors' names from each card. We used natural language processing from the nltk package and regular expressions to find proper nouns at the begining of the cards that started with the start letter for that drawer.

In [1]:
```
import pandas as pd
import numpy as np
pd.set_option('display.max_colwidth', None)
pd.set_option("display.max_row",None)
import nltk
import re
```

In [16]:
```
# CHUNKING WITH START LETTER PARAMETER
def chunk(df,start_letter):
    out=""
    start_letter = start_letter.upper()
    df =  re.sub(r'[0-9]+', '', df[:30])
    first_ind = df.find(start_letter)
    # If word starting with start letter does not exist, return None
    if(first_ind ==-1):
        return None
    word_tok = nltk.word_tokenize(df[first_ind:50+first_ind])
    taged_sent= nltk.pos_tag(word_tok)
    # Checks for Proper Noun,  coordinating conjunction(and,&), Proper Nouns
    grammar = "Name: {((<NN.><,>)?<NNP><.>?<CC>?<NNP>?<CC>?<NNP>*)}"
    cp = nltk.RegexpParser(grammar,loop=1)
    chunked = cp.parse(taged_sent)
    # only run once
    for subtree in chunked.subtrees(filter =  lambda x : x.label()=="Name"):
         # Generate all subtrees
        li = [i[0] for i in subtree.leaves()]
#         print(li)

        if li[0].startswith(start_letter):
            out =  " ".join([i[0] for i in subtree.leaves()])
            break
        # if no chunk start with the letter
        # Get the first word starting with the letter
        else:
            for i in chunked.leaves():
                if i[0].startswith(start_letter):
                    first = i[0]
                    out =  first + ", "+ " ".join(li)
                    break
    out = out.title()
    out = out.replace(" , ",", ")
    out = out.strip()
    endings = ["Papers", "Letters", "Diary", "Notebook", "Book", "Scrapbook", "Screenplay", "Memoir", "Card", '
            "Account", "Sketch", "Journal", "Letter", "Record", "Notes", "Ledger", "Rent", "Letterpress", "Addr
    for i in endings:
        out = out.replace(i,"")
    return out
```

In [4]:
```
df = pd.read_csv("all_text_chunked_name.csv")
df.head()
```

Out[4]:

	Text	Name
0	A. B. Davis and Company (Philadelphia, Pa.) See Davis (A. B.) and Company	B. Davis and Company
1	AeHe Roscoe (Firm: Nashville, Tenne)e Journal, 1853, Septe-1857, Dece 1 item(800 ppe)e Wholesale and retail druggist and dealer in paints, oils, and dyestuffse Summary: Journal (account book) documents the sale of chemical and herbal drugs, paint and painting supplies, dyestuffs, personal and household supplies, and garden seeds to individuals and businessese le Drugstores--Tennesseee 2e Paint shops--Equipment and supplies-—- Tennesseee 3e Dyes and dyeinge 46 Household supplies--Tennesseee 5c Herbs--Therapeutic usee 6+ Seed industry and trade--Tennessees 7e Nashville ite® nne)—--Commercee 20 MAY QO1 23804905 NDHYme	AeHe Roscoe
2	Abbeville District (8.C.) See South Carolina. Abbeville District	Abbeville District
3	Abbott, William B. Papers, 1862-1864 Frederick Co., Va. Section A 5-16-57 GUIDE 10 items	Abbott, William B
4	Abbott, William B. Papers, 1862-1864, Fre- Gerick Co., Va. 10 items. Sketch These are the papers of William B. Abbott, evidently a well-to-do farmer of Frederick Co., Va. There are several documents concerned with the evaluation of damage done to his property by C. S. A. troops in 1862, and.receipts in 1864 for hay bought from Abbott at various times in Aug., 1864 by the C. S. A. Army.	Abbott, William B

In [1]:
```
# cumulative size of drawers
cum_sum = [838, 1518, 2453, 3173, 3812, 4599, 5364, 6140, 6870, 7551, 8258, 9035, 9833, 10478, 11268, 12009, 12
42824, 43697, 44430, 45143, 45832, 46491, 47265, 48080, 48814, 49541, 50084, 50368]
#  real size of each drawer
real_size = [838, 680, 935, 720, 639, 787, 765, 776, 730, 681, 707, 777, 798, 645, 790, 741, 654, 760, 581, 658
```

In [17]:
```
#Divide the rows into their respective drawers
drawer_no = ['157', '158', '159', '160', '161', '162', '163', '164', '165', '166', '167', '169', '170', '171',
'220', '221', '222', '223', '224', '225', '226', '227', '229', '230', '231']
df_list = []
prev = None
for i in range(69):
    df_list.append(df.iloc[prev:cum_sum[i]])
    prev=cum_sum[i]
# print(dict(zip(drawer_no,[i for i in range(69)])))
```

```
{'157': 0, '158': 1, '159': 2, '160': 3, '161': 4, '162': 5, '163': 6, '164': 7, '165': 8, '166': 9, '167': 10,
'169': 11, '170': 12, '171': 13, '172': 14, '173': 15, '174': 16, '175': 17, '176': 18, '177': 19, '178': 20,
'179': 21, '181': 22, '182': 23, '183': 24, '184': 25, '185': 26, '186': 27, '187': 28, '188': 29, '189': 30,
'190': 31, '191': 32, '193': 33, '194': 34, '195': 35, '196': 36, '197': 37, '198': 38, '199': 39, '200': 40,
'201': 41, '202': 42, '203': 43, '205': 44, '206': 45, '207': 46, '208': 47, '209': 48, '210': 49, '211': 50,
'212': 51, '213': 52, '214': 53, '215': 54, '217': 55, '218': 56, '219': 57, '220': 58, '221': 59, '222': 60,
'223': 61, '224': 62, '225': 63, '226': 64, '227': 65, '229': 66, '230': 67, '231': 68}
```

In [20]:
```
# select df_list[i] to select drawer
file = df_list[7]
file.Name = file.Text.apply(lambda row : chunk(row,start_letter="B"))

file.Name = file.Name.str.replace(" , ",", ")
file.Name = file.Name.str.replace("For Information.*","")
file.Name = file.Name.str.replace("Pap.*","")
file.Name = file.Name.str.replace("See.*","")
file.Name = file.Name.str.strip()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py:5494: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  self[name] = value
```