PART 7 -- Percentage of top 20 spenders from the Democratic Party

In [6]:
```
import numpy as np
import pandas as pd
```

Read all the *detail.csv.
Renamed "2015Q2-house-disburse-detail.csv" to "2015Q2-house-disburse-detail-old.csv"
Then renamed "2015Q2-house-disburse-detail-updated.csv" to "2015Q2-house-disburse-detail.csv". Then redirected all the filenames to "filename.txt" using the command: ls *detail.csv > filename.txt

In [7]:
```
# Create a list of filename called file_list
# Strip '\n' at the end of the filename
#Ref: https://stackoverflow.com/questions/42488579/
#remove-n-from-each-string-stored-in-a-python-list

file_list = []
with open('filename.txt', 'r', encoding='utf-8') as myfile:
    for line in myfile:
        st_line = line.rstrip()
        file_list.append(st_line)
file_list=file_list[26:30]  #Slicing 2016 files
print(file_list)
```

```
['2016Q1-house-disburse-detail.csv', '2016Q2-house-disburse-detail.csv
', '2016Q3-house-disburse-detail.csv', '2016Q4-house-disburse-detail.c
sv']
```

In [8]:
```
#Create a dataframe for each of 2016 quarter files and concatenate the 4
df1 = pd.read_csv('2016Q1-house-disburse-detail.csv', low_memory = False
df2 = pd.read_csv('2016Q2-house-disburse-detail.csv', low_memory = False
df3 = pd.read_csv('2016Q3-house-disburse-detail.csv', low_memory = False
df4 = pd.read_csv('2016Q4-house-disburse-detail.csv', low_memory = False
```

In [9]:
```
df = pd.concat([df1, df2, df3, df4])
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 385613 entries, 0 to 90674
Data columns (total 16 columns):
AMOUNT           385613 non-null object
BIOGUIDE_ID      306557 non-null object
CATEGORY         385613 non-null object
DATE             328689 non-null object
END DATE         385612 non-null object
OFFICE           385613 non-null object
PAYEE            334724 non-null object
PROGRAM          90675 non-null object
PURPOSE          385611 non-null object
QUARTER          385613 non-null object
RECIP (orig.)    334724 non-null object
RECORDID         328690 non-null object
START DATE       385612 non-null object
TRANSCODE        328692 non-null object
TRANSCODELONG    250648 non-null object
YEAR             385613 non-null object
dtypes: object(16)
memory usage: 50.0+ MB
```

In [11]:
```
df.head()
```

Out[11]:

	AMOUNT	BIOGUIDE_ID	CATEGORY	DATE	END DATE	OFFICE	PAYEE	PR(
0	380.00	NaN	SUPPLIES AND MATERIALS	03-18	02/28/16	OFFICE OF THE SPEAKER	CITI PCARD-GALLERIA FLORIST	
1	6,666.67	NaN	PERSONNEL COMPENSATION	NaN	03/31/16	OFFICE OF THE SPEAKER	ALTHOUSE,JOSHUA S	
2	25,666.67	NaN	PERSONNEL COMPENSATION	NaN	03/31/16	OFFICE OF THE SPEAKER	ANDRES,DOUGLAS R	
3	18,333.33	NaN	PERSONNEL COMPENSATION	NaN	03/31/16	OFFICE OF THE SPEAKER	ANDREWS,THOMAS S	
4	26,250.00	NaN	PERSONNEL COMPENSATION	NaN	03/31/16	OFFICE OF THE SPEAKER	ANTELL,GEOFFREY	

In [12]:
```
#Check if any column has null values

df.columns[df.isnull().any()].tolist()
```

Out[12]:
```
['BIOGUIDE_ID',
 'DATE',
 'END DATE',
 'PAYEE',
 'PROGRAM',
 'PURPOSE',
 'RECIP (orig.)',
 'RECORDID',
 'START DATE',
 'TRANSCODE',
 'TRANSCODELONG']
```

In [13]:
```
type(df['START DATE'])
```

Out[13]:
```
pandas.core.series.Series
```

In [14]:
```
print(df['START DATE'].head())
```

```
0    01/29/16
1    02/01/16
2    01/03/16
3    01/03/16
4    01/28/16
Name: START DATE, dtype: object
```

In [15]:
```
# Create a column called "START YEAR"
df['START YEAR'] = df['START DATE'].apply(lambda x : str(x)[-2: ])
```

In [16]:
```
df['START YEAR'].head()
```

Out[16]:
```
0    16
1    16
2    16
3    16
4    16
Name: START YEAR, dtype: object
```

In [17]:
```
#Consider only data with 'START DATE' in 2016
df = df[df['START YEAR'] == '16']
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 358703 entries, 0 to 90674
Data columns (total 17 columns):
AMOUNT           358703 non-null object
BIOGUIDE_ID      283807 non-null object
CATEGORY         358703 non-null object
DATE             302654 non-null object
END DATE         358703 non-null object
OFFICE           358703 non-null object
PAYEE            310562 non-null object
PROGRAM          89744 non-null object
PURPOSE          358701 non-null object
QUARTER          358703 non-null object
RECIP (orig.)    310562 non-null object
RECORDID         302654 non-null object
START DATE       358703 non-null object
TRANSCODE        302655 non-null object
TRANSCODELONG    225539 non-null object
YEAR             358703 non-null object
START YEAR       358703 non-null object
dtypes: object(17)
memory usage: 49.3+ MB
```

In [18]:
```
# AMOUNT is a string column. Convert to a float.

df['AMOUNT'] = pd.to_numeric(df['AMOUNT'], errors='coerce')
print(type(df['AMOUNT'].iloc[0]))
```

```
<class 'numpy.float64'>
```

In [25]:
```
#Ref: https://stackoverflow.com/questions/27018622/pandas-groupby-sort-d
group_by_rep_df = df.groupby(df['BIOGUIDE_ID'])['AMOUNT'].sum().sort_val
top_20_spenders = group_by_rep_df.head(20)
```

In [31]:
```
top_20_list = top_20_spenders.index
print(top_20_list)
```

```
Index(['C001103', 'K000376', 'K000362', 'N000181', 'V000132', 'L000571
',
       'A000374', 'Z000018', 'C001036', 'V000129', 'Y000033', 'B001278
',
       'P000606', 'B000287', 'T000193', 'C001049', 'B001248', 'R000580
',
       'L000576', 'P000596'],
      dtype='object', name='BIOGUIDE_ID')
```

In [33]:
```
type(top_20_spenders.index)
```

Out[33]: pandas.core.indexes.base.Index

In [34]:
```
top_20_spenders.index[2]
```

Out[34]: 'K000362'

In [37]:
```
with open('top_20_file.csv', 'w', encoding = 'utf-8') as top_file:
    for rep in top_20_list:
        top_file.write("{}\n".format(rep))
```

In [54]:
```
import json

party_list = []
for rep in top_20_list:
    filename = rep + ".json"
    with open(filename, 'r') as f:
        rep_dict = json.load(f)

    party = rep_dict['results'][0]['current_party']
    party_list.append(party)
```

In [55]:
```
print(party_list)
```

```
['R', 'R', 'R', 'R', 'D', 'R', 'R', 'R', 'D', 'R', 'R', 'D', 'R', 'D',
'D', 'D', 'R', 'R', 'R', 'D']
```

In [56]:
```
print(len(party_list))
```

```
20
```

In [58]:
```
count_democrats = 0
for item in party_list:
    if item == 'D':
        count_democrats += 1
print(count_democrats)
```

```
7
```

In [61]:
```
dem_party_percent =  100 * count_democrats/len(party_list)
print(dem_party_percent)
```

```
35.0
```

Percentage of top 20 Spenders in the Democratic party = 35

In [ ]: