PART 4

In [1]:

```
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 [2]:

```
# 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 [3]:

```
#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 [4]:

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

In [9]:

```
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 [10]:

```
df.head()
```

Out[10]:

	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 [11]:

```
#Check if any column has null values

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

Out[11]:

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

In [12]:

```
type(df['START DATE'])
```

Out[12]:

```
pandas.core.series.Series
```

In [15]:

```
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 [23]:

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

In [24]:

```
df['START YEAR'].head()
```

Out[24]:

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

In [28]:

```python
#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 [52]:

```python
# 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 [54]:

```python
#Ref: https://stackoverflow.com/questions/27018622/pandas-groupby-sort-d
group_by_office = df.groupby(df['OFFICE'])['AMOUNT'].sum().sort_values(a
group_by_office.head()
```

Out[54]:

```
OFFICE
GOVERNMENT CONTRIBUTIONS          62767919.92
CHIEF ADMIN OFCR OF THE HOUSE     42449309.00
COMMITTEE ON APPROPRIATIONS        7035246.89
CLERK OF THE HOUSE                 6019765.09
COMMITTEE ON ENERGY & COMMERCE     3154845.30
Name: AMOUNT, dtype: float64
```

GOVERNMENT CONTRIBUTIONS is the OFFICE that has the most expenditure = $62767919.92

In [56]:

```
#Let us just look at just rows having GOVERNMENT CONTRIBUITIONS in the O
govt_contrib_df = df[ df['OFFICE'] == 'GOVERNMENT CONTRIBUTIONS']
govt_contrib_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30073 entries, 93165 to 88296
Data columns (total 17 columns):
AMOUNT           29935 non-null float64
BIOGUIDE_ID      0 non-null object
CATEGORY         30073 non-null object
DATE             29967 non-null object
END DATE         30073 non-null object
OFFICE           30073 non-null object
PAYEE            29892 non-null object
PROGRAM          7270 non-null object
PURPOSE          30073 non-null object
QUARTER          30073 non-null object
RECIP (orig.)    29892 non-null object
RECORDID         29967 non-null object
START DATE       30073 non-null object
TRANSCODE        29967 non-null object
TRANSCODELONG    22722 non-null object
YEAR             30073 non-null object
START YEAR       30073 non-null object
dtypes: float64(1), object(16)
memory usage: 4.1+ MB
```

In [57]:

```
#In GOVERNMENT CONTRIBUTIONS office, we want to find the 'PURPOSE'
#that accounts for the highest total expenditure

groupby_purpose = govt_contrib_df.groupby('PURPOSE')['AMOUNT'].sum().sor
groupby_purpose.head()
```

Out[57]:

```
PURPOSE
FERS                    14876518.54
STUDENT LOANS           14661130.44
FICA                     6219593.42
HEALTH INSURANCE F       5884855.20
TSP MATCHING             5532101.82
Name: AMOUNT, dtype: float64
```

The PURPOSE is FERS that has the highest total expenditure of $14876518.54 office in the GOVERNMENT CONTRIBUTIONS office which is the office with the highest total expenditure with 'START DATE' in 2016.

In [58]:

```
#Calculate the total expenditure with START DATE in 2016
total_expenditure = df['AMOUNT'].sum()
print(total_expenditure)
```

```
339265615.4399895
```

In [60]:

```
highest_purpose_exp = groupby_purpose.max()
print(highest_purpose_exp)
```

```
14876518.54
```

In [62]:

```
#Calculate fraction of total expenditure to highest_purpose_exp
fraction = highest_purpose_exp / total_expenditure
print(fraction)
```

```
0.04384917852847192
```

In [ ]: