Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Probability and Statistics (01CT1401)	**Aim: How to Process , Analyze and Visualize Data**	
Task :- 2	**Date:-** 26-02-2024	**Enrollment No:-** 92200133030

Day 1: Let's play with some data!

First Step :- unzip the file and rename it to something meaningful name

```
D:\Aryan Data\Usefull Data\Semester - 4\Probability and Statistics\Tasks\TASK -2 - Statistics -Case study\Day 1> mv P000
00001-ALL.txt donations.txt
```

Getting The Count of Lines :-

```
D:\Aryan Data\Usefull Data\Semester - 4\Probability and Statistics\Tasks\TASK -2 - Statistics -Case study\Day 1> wc -l d
onations.txt
4084075 donations.txt
```

Quick Look :-

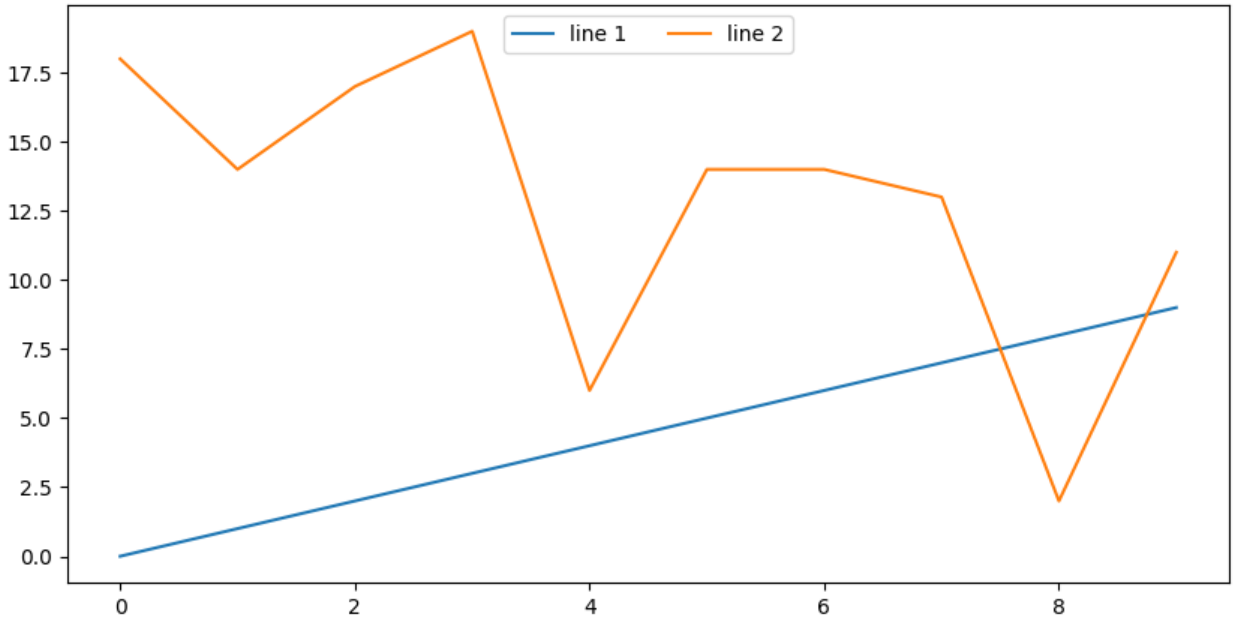
```
D:\Aryan Data\Usefull Data\Semester - 4\Probability and Statistics\Tasks\TASK -2 - Statistics -Case study\Day 1>head -n3
 donations.txt
cmte_id,cand_id,cand_nm,contbr_nm,contbr_city,contbr_st,contbr_zip,contbr_employer,contbr_occupation,contb_receipt_amt,c
ontb_receipt_dt,receipt_desc,memo_cd,memo_text,form_tp,file_num
C00420224,"P80002983","Cox, John H","BROWN, CHARLENE","EAGLE RIVER","AK","99577","","STUDENT",25,01-MAR-07,"","","","SA1
7A",288757
C00420224,"P80002983","Cox, John H","KELLY, RAY","HUNTSVILLE","AL","35801","ARKTECH","RETIRED",25,25-JAN-07,"","","","SA
17A",288757
```

Introduction to Matplotlib :-

Code :-

```
import matplotlib.pyplot as plt
import random

xs = range(10)
ys1 = range(10)
ys2 = [random.randint(0, 20) for i in range(10)]

fig = plt.figure(figsize=(10,5))
plt.plot(xs, ys1, label='line 1')
plt.plot(xs, ys2, label='line 2')
plt.legend(loc='upper center', ncol = 4)
plt.savefig('twolines.png', format='png')
```

Output :-



Sampling The Data :-

Code :-

```
import sys

with open('./donations.txt', "r") as f:
    i = 0
    for line in f:
        if i % 1000 == 0:
            print(line[:-1])
        i += 1
```

Output :-

```
C00431569,"P00003392","Clinton, Hillary Rodham","MARETT, JOSEPH","CHARLOTTE","NC","282696953","SAV/WAY FOODS","OWNER",50,12-JAN-0
8,"","","","SA17A",337693
C00431569,"P00003392","Clinton, Hillary Rodham","WOOD, SWAIN","RALEIGH","NC","276073722","SELF EMPLOYED","ATTORNEY",250,07-FEB-08
,"","","","SA17A",336278
C00431569,"P00003392","Clinton, Hillary Rodham","KACHERGIS, JOYCE W","PITTSBORO","NC","273125862","NOT EMPLOYED","RETIRED",150,07
-FEB-08,"","","","SA17A",336278
C00431569,"P00003392","Clinton, Hillary Rodham","FARMER, ROBERT","BESSEMER CITY","NC","280169694","NONE","RETIRED",50,17-JAN-08,"
","","","SA17A",337693
C00431569,"P00003392","Clinton, Hillary Rodham","STOFFERAHN, KENNETH","LINCOLN","NE","685218983","NOT EMPLOYED","RETIRED",100,25-
APR-08,"","","","SA17A",341172
C00431569,"P00003392","Clinton, Hillary Rodham","MEARA, PATRICIA","CAMPTON","NH","032230440","NETWORK & SECURITY TECHN","SOFTWARE
 ENGINEER",70,31-MAY-08,"","","","SA17A",346097
C00431569,"P00003392","Clinton, Hillary Rodham","DOLAN, KERRY","NASHUA","NH","030632826","SELF EMPLOYED","WRITER",50,18-MAR-08,""
,"","","SA17A",341838
C00431569,"P00003392","Clinton, Hillary Rodham","PIETZ, PAUL F","NORTH SWANZEY","NH","034314465","NOT EMPLOYED","RETIRED",100,22-
MAY-08,"","","","SA17A",346097
C00431569,"P00003392","Clinton, Hillary Rodham","PATTI, JOSEPH","CHATHAM","NJ","079281237","RP BENNETT CUSTOM BUILDE","PROJECT MA
NAGER",25,07-MAY-08,"","","","SA17A",346097
C00431569,"P00003392","Clinton, Hillary Rodham","SULLIVAN, LAWRENCE","FRANKLIN PARK","NJ","088231329","SOMERSET HILLS RTC","SUPV.
 RES. SERV.",100,03-JUN-08,"","","","SA17A",353643
C00431569,"P00003392","Clinton, Hillary Rodham","SANTINI, ANA","BELLE MEAD","NJ","085024935","NOT EMPLOYED","NOT EMPLOYED",1000,1
9-JUN-07,"","","","SA17A",392796
C00431569,"P00003392","Clinton, Hillary Rodham","RABNER, HAROLD","MONTCLAIR","NJ","070422006","RABNER, ALLCORN, ET AL PC","ATTORN
EY",2300,28-SEP-07,"","","","SA17A",405770
C00431569,"P00003392","Clinton, Hillary Rodham","LEIBOWITZ, SUSAN","WAYNE","NJ","074705527","GIRL SCOUTS OF NORTHERN","MEMBERSHIP
",50,12-DEC-08,"","","","SA17A",401333
C00431569,"P00003392","Clinton, Hillary Rodham","ABRAMSON, CHARLOTTE","TEANECK","NJ","076662827","JEWISH THEOLOGICAL SEMIN","EDUC
ATOR",120,24-FEB-08,"","","","SA17A",336278
C00431569,"P00003392","Clinton, Hillary Rodham","GUNES, MURAT","PALISADES PARK","NJ","076501318","SELF EMPLOYED","ARTIST",1000,31
-DEC-07,"","","","SA17A",345164
C00431569,"P00003392","Clinton, Hillary Rodham","BAKER, LINDA","HOPEWELL","NJ","08525","NOT EMPLOYED","NOT EMPLOYED",1000,24-MAR-
08,"","","","SA17A",341838
```

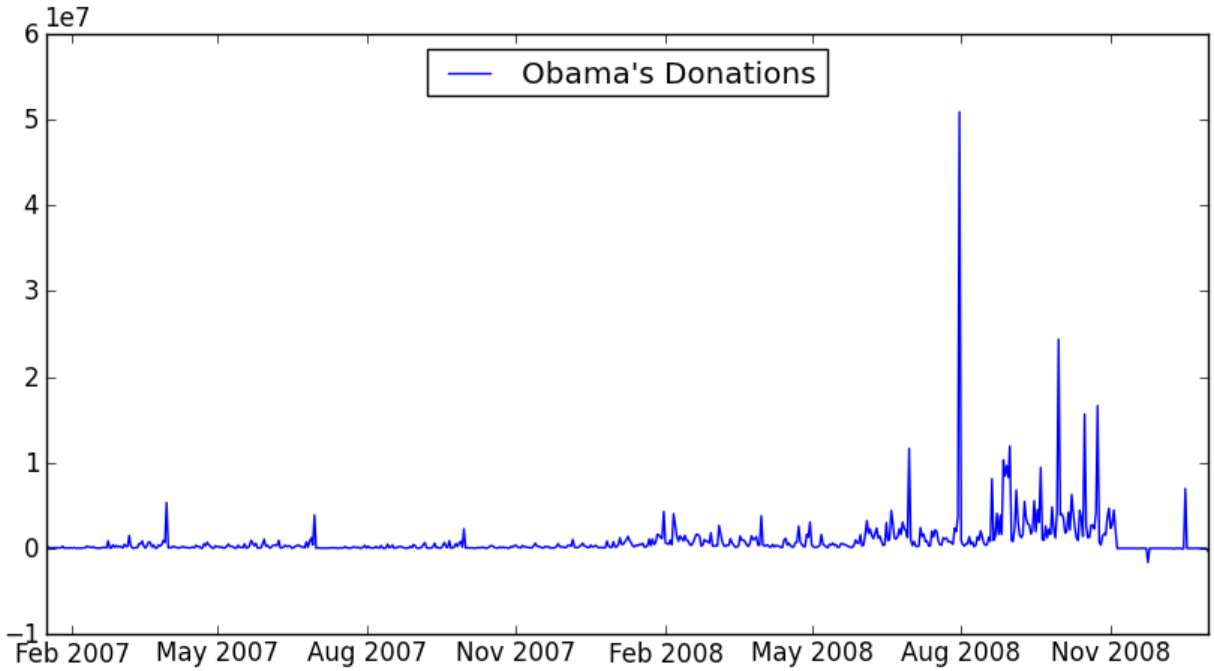
Plotting The Data :-

Code :-

```
from collections import defaultdict
import matplotlib.pyplot as plt
import csv, sys, datetime

reader = csv.DictReader(open(sys.argv[1], 'r'))
obamadonations = defaultdict(lambda:0)
for row in reader:
    name = row['cand_nm']
    datestr = row['contb_receipt_dt']
    amount = float(row['contb_receipt_amt'])
    date = datetime.datetime.strptime(datestr, '%d-%b-%y')
    if 'Obama' in name:
        obamadonations[date] += amount
```

```
sorted_by_date = sorted(obamadonations.items(), key=lambda x: x[0])
xs, ys = zip(*sorted_by_date)
plt.plot(xs, ys, label='line 1')
plt.legend(loc='upper center', ncol=4)
plt.savefig('test.png', format='png')
```

Output :-



The Case of the Negative Donation:-

Code :-

```
import csv
import datetime
import sys

reader = csv.DictReader(open('./donations.txt', "r"))
for row in reader:
    name = row["cand_nm"]
    datestr = row["contb_receipt_dt"]
```

```
    amount = float(row["contb_receipt_amt"])
    if amount < 0:
        line = "\t".join(
            [row["cand_nm"], row["contb_receipt_dt"], row["contb_receipt_amt"]]
        )
        print(line)
```

Output :-

```
C00430470,"P80002801","McCain, John S","CHANEY, MICHAEL J. MR.","VICKSBURG","MS","391805426","STATE OF MISSISSIPPI","ST
C00430470,"P80002801","McCain, John S","RIGLER, SUSAN K. MS.","GARDINER","MT","590300970","GARDINER SCHOOLS","TEACHER",
C00430470,"P80002801","McCain, John S","WILLIAMS, JAMES D. MR.","HAMPSTEAD","NC","284432476","INVENSYS","SALES",200,19-
C00430470,"P80002801","McCain, John S","POLYCHRON, JOHN MR.","WINSTON-SALEM","NC","271041113","RETIRED","RETIRED",100,2
C00430470,"P80002801","McCain, John S","WALKER, JAMES R. MR.","EDEN","NC","272890528","SELF-EMPLOYED","ATTORNEY",201,24
C00430470,"P80002801","McCain, John S","MATHENY, SHARON MS.","NEW LONDON","NC","281279102","RETIRED","RETIRED",1000,19-
C00430470,"P80002801","McCain, John S","WRIGHT, DAVID C. MR. III","CHARLOTTE","NC","282091529","ROBINSON BRADSHAW & HIN
C00430470,"P80002801","McCain, John S","WOODALL, LEONARD S. MR.","SMITHFIELD","NC","275773857","RETIRED","RETIRED",300,
C00430470,"P80002801","McCain, John S","WHEELER, CARLTON MR.","ASHEVILLE","NC","288031516","SELF","MEDICAL",100,03-JAN-
C00430470,"P80002801","McCain, John S","STOUT, RONALD I. MR.","PINEHURST","NC","283748396","RETIRED","RETIRED",25,06-OC
C00430470,"P80002801","McCain, John S","SAND, DUANE A. CDR","BISMARCK","ND","585030123","AFP FOUNDATION","STATE DIRECTC
C00430470,"P80002801","McCain, John S","HEIN, RONALD L. MR.","HOLDREGE","NE","689491039","RETIRED","RETIRED",150,25-AUG
C00430470,"P80002801","McCain, John S","WATKINS, GEORGE H. MR.","WALPOLE","NH","036085039","","RETIRED",50,01-AUG-07,""
C00430470,"P80002801","McCain, John S","MORAN, SUSAN M. MRS.","BEDFORD","NH","031104538","SKILLSOFT","CEO",252,04-AUG-0
C00430470,"P80002801","McCain, John S","NERAD, ROBERT A. MR.","NEWBURY","NH","032550293","RETIRED","RETIRED",25,23-JUL-
C00430470,"P80002801","McCain, John S","COYNE, JOHN MR. JR.","VOORHEES","NJ","080431217","RETIRED","RETIRED",100,14-AUG
C00430470,"P80002801","McCain, John S","FITZPATRICK, LAWRENCE MR.","PRINCETON JUNCTION","NJ","085501908","SELF-EMPLOYED
```

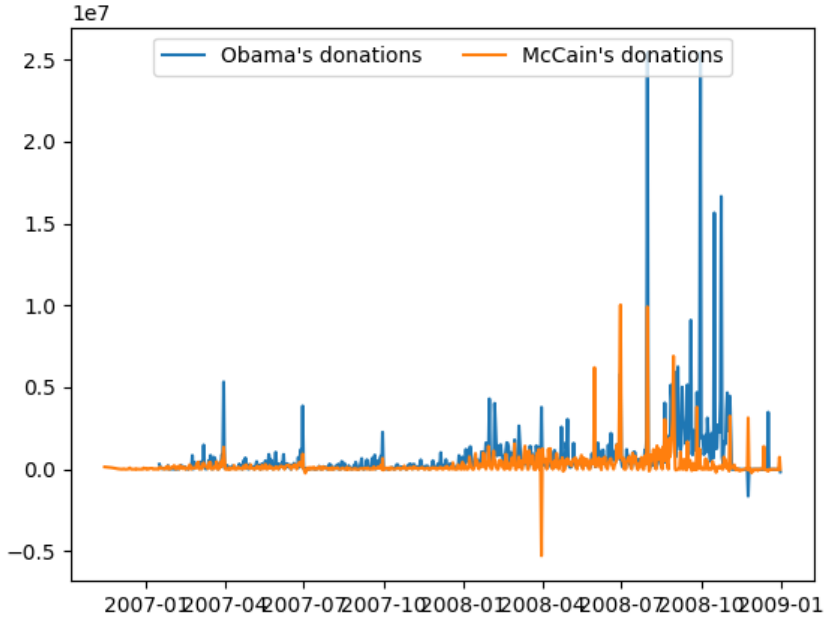
Exercise – 1 :- Plot Obama vs. McCain

Code :-

```
from collections import defaultdict
import matplotlib.pyplot as plt
import csv, sys, datetime

filename = './donations.txt'
with open(filename, 'r') as file:
    reader = csv.DictReader(file)
    obamadonations = defaultdict(lambda: 0)
    mccaindonations = defaultdict(lambda: 0)

    for row in reader:
        name = row['cand_nm']
        datestr = row['contb_receipt_dt']
        amount = float(row['contb_receipt_amt'])
```

```
    date = datetime.datetime.strptime(datestr, '%d-%b-%y')

    if 'Obama' in name:
        obamadonations[date] += amount
    elif 'McCain' in name:
        mccaindonations[date] += amount

sorted_obama_by_date = sorted(obamadonations.items(), key=lambda x: x[0])
xs_obama, ys_obama = zip(*sorted_obama_by_date)
sorted_mccain_by_date = sorted(mccaindonations.items(), key=lambda x: x[0])
xs_mccain, ys_mccain = zip(*sorted_mccain_by_date)
plt.plot(xs_obama, ys_obama, label="Obama's donations")
plt.plot(xs_mccain, ys_mccain, label="McCain's donations")
plt.legend(loc='upper center', ncol=2)
plt.savefig('task_1.png', format='png')
plt.show()
```

Output :-



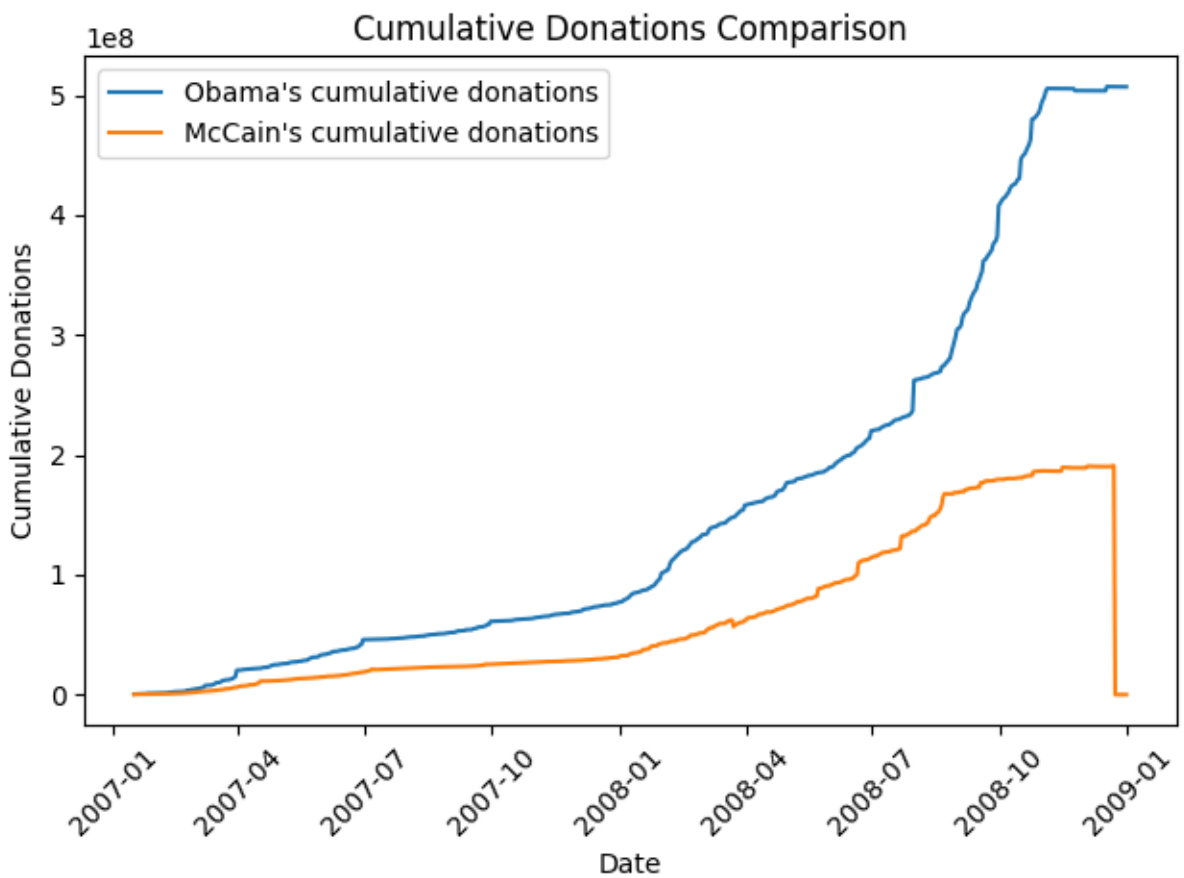
Exercide – 2 :- Cumulative Graphs

Code :-

```
from collections import defaultdict
import matplotlib.pyplot as plt
import csv, datetime
import numpy as np

filename = './donations.txt'
with open(filename, 'r') as file:
    reader = csv.DictReader(file)
    obamadonations = defaultdict(lambda: 0)
    mccaindonations = defaultdict(lambda: 0)

    for row in reader:
        name = row['cand_nm']
        datestr = row['contb_receipt_dt']
        amount = float(row['contb_receipt_amt'])
        date = datetime.datetime.strptime(datestr, '%d-%b-%y')

        if 'Obama' in name:
            obamadonations[date] += amount
        elif 'McCain' in name:
            mccaindonations[date] += amount

sorted_obama_by_date = sorted(obamadonations.items(), key=lambda x: x[0])
sorted_mccain_by_date = sorted(mccaindonations.items(), key=lambda x: x[0])

cumulative_obama = [sum(amount for date, amount in sorted_obama_by_date[:i+1]) for i in
range(len(sorted_obama_by_date))]
cumulative_mccain = [sum(amount for date, amount in sorted_mccain_by_date[:i+1]) for i in
range(len(sorted_mccain_by_date))]

dates = [date for date, amount in sorted_obama_by_date]

max_length = max(len(dates), len(cumulative_mccain))
dates = dates[:max_length]
cumulative_obama = np.pad(cumulative_obama, (0, max_length - len(cumulative_obama)), mode='constant')
cumulative_mccain = np.pad(cumulative_mccain, (0, max_length - len(cumulative_mccain)), mode='constant')

plt.plot(dates, cumulative_obama, label="Obama's cumulative donations")
```

```
plt.plot(dates, cumulative_mccain, label="McCain's cumulative donations")
plt.legend(loc='upper left')
plt.xlabel('Date')
plt.ylabel('Cumulative Donations')
plt.title('Cumulative Donations Comparison')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Output :-



Exercise – 3 :- Understand "Reattribution to Spouse"

Code :-

```
import csv
import sys
from collections import defaultdict
def filter_reattribution_contributions(reader):
    filtered_contributions = []
    for row in reader:
        if 'reattribution' in row['memo_text'].lower() or 'reattribution' in row['receipt_desc'].lower():
            filtered_contributions.append(row)
    return filtered_contributions



def calculate_cumulative_donations(filtered_contributions):
    cumulative_donations = defaultdict(lambda: 0)
    for row in filtered_contributions:
        candidate_name = row['cand_nm']
        donation_amount = float(row['contb_receipt_amt'])
        cumulative_donations[candidate_name] += donation_amount
    return cumulative_donations

def determine_preferred_candidate(cumulative_donations):
    preferred_candidate = max(cumulative_donations, key=cumulative_donations.get)
    return preferred_candidate

def main():
    reader = csv.DictReader(open(sys.argv[1], 'r'))
    filtered_contributions = filter_reattribution_contributions(reader)
    cumulative_donations = calculate_cumulative_donations(filtered_contributions)
    preferred_candidate = determine_preferred_candidate(cumulative_donations)
    print("Preferred Candidate:", preferred_candidate)

if __name__ == "__main__":
    main()
```

Output :-

```
PS D:\Aryan Data\Usefull Data\Semester - 4\Probability and Statistics\Tasks\TASK -2 - Statistics -Case study
\Day 1> & "C:/Program Files/Python312/python.exe" "d:/Aryan Data/Usefull Data/Semester - 4/Probability and S
tatistics/Tasks/TASK -2 - Statistics -Case study/Day 1/exercise1.py"
Preferred Candidate: McCain, John S
PS D:\Aryan Data\Usefull Data\Semester - 4\Probability and Statistics\Tasks\TASK -2 - Statistics -Case study
\Day 1>
```

Exercide – 4:- Pause and Think

Code :-

```
import csv
import sys
import matplotlib.pyplot as plt
from collections import defaultdict

def filter_reattribution_contributions(reader):
    filtered_contributions = []
    for row in reader:
        if 'reattribution' in row['memo_text'].lower() or 'reattribution' in row['receipt_desc'].lower():
            filtered_contributions.append(row)
    return filtered_contributions

def calculate_cumulative_donations(filtered_contributions):
    cumulative_donations = defaultdict(lambda: {'total': 0, 'reattribution': 0})
    for row in filtered_contributions:
        candidate_name = row['cand_nm']
        donation_amount = float(row['contb_receipt_amt'])
        if donation_amount > 0:
            cumulative_donations[candidate_name]['total'] += donation_amount
        else:
            cumulative_donations[candidate_name]['reattribution'] += abs(donation_amount)
    return cumulative_donations

def calculate_ratio(cumulative_donations):
    ratios = {}
    for candidate, donations in cumulative_donations.items():
        total_donations = donations['total']
        reattribution_donations = donations['reattribution']
        ratio = reattribution_donations / total_donations if total_donations != 0 else 0
        ratios[candidate] = ratio
    return ratios
```

```
def main():
    reader = csv.DictReader(open(sys.argv[1], 'r'))
    filtered_contributions = filter_reattribution_contributions(reader)
    cumulative_donations = calculate_cumulative_donations(filtered_contributions)
    ratios = calculate_ratio(cumulative_donations)

    plt.figure(figsize=(10, 6))
    plt.bar(ratios.keys(), ratios.values(), color='skyblue')
    plt.xlabel('Candidate')
    plt.ylabel('Ratio of Reattribution to Overall Donations')
    plt.title('Ratio of Reattribution to Overall Donations for Each Candidate')
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    main()
```

Output:-