# Part 2 -- Calculation of stddev of COVERAGE PERIOD

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)
print(file_list)
```

```
['2009Q3-house-disburse-detail.csv', '2009Q4-house-disburse-detail.csv
', '2010Q1-house-disburse-detail.csv', '2010Q2-house-disburse-detail.c
sv', '2010Q3-house-disburse-detail.csv', '2010Q4-house-disburse-detail
.csv', '2011Q1-house-disburse-detail.csv', '2011Q2-house-disburse-deta
il.csv', '2011Q3-house-disburse-detail.csv', '2011Q4-house-disburse-de
tail.csv', '2012Q1-house-disburse-detail.csv', '2012Q2-house-disburse-
detail.csv', '2012Q3-house-disburse-detail.csv', '2012Q4-house-disburs
e-detail.csv', '2013Q1-house-disburse-detail.csv', '2013Q2-house-disbu
rse-detail.csv', '2013Q3-house-disburse-detail.csv', '2013Q4-house-dis
burse-detail.csv', '2014Q1-house-disburse-detail.csv', '2014Q2-house-d
isburse-detail.csv', '2014Q3-house-disburse-detail.csv', '2014Q4-house
-disburse-detail.csv', '2015Q1-house-disburse-detail.csv', '2015Q2-hou
se-disburse-detail.csv', '2015Q3-house-disburse-detail.csv', '2015Q4-h
ouse-disburse-detail.csv', '2016Q1-house-disburse-detail.csv', '2016Q2
-house-disburse-detail.csv', '2016Q3-house-disburse-detail.csv', '2016
Q4-house-disburse-detail.csv', '2017Q1-house-disburse-detail.csv', '20
17Q2-house-disburse-detail.csv', '2017Q3-house-disburse-detail.csv', '
2017Q4-house-disburse-detail.csv', '2018Q1-house-disburse-detail.csv']
```

In [3]:
```
#Try for first file
df = pd.read_csv('2009Q3-house-disburse-detail.csv', sep=',', engine = '

df['AMOUNT'] = df['AMOUNT'].apply(pd.to_numeric, errors='coerce')
df = df[df['AMOUNT'] > 0]
df.head()
```

Out[3]:

| | BIOGUIDE_ID | OFFICE | QUARTER | CATEGORY | DATE | PAYEE | START DATE | I |
|---|---|---|---|---|---|---|---|---|
| **3** | NaN | COMMUNICATIONS | 2009Q3 | OTHER SERVICES | NaN | 08Â25 P2 MFP0003163 AVAYA | 05/29/09 | 05/2 |
| **4** | NaN | COMMUNICATIONS | 2009Q3 | OTHER SERVICES | NaN | 09Â10 P2 OPR0900726C STR... | 10/04/06 | 10/( |
| **5** | NaN | COMMUNICATIONS | 2009Q3 | OTHER SERVICES | NaN | 09Â10 P2 OPR0900726C ... | 10/04/06 | 10/( |
| **7** | NaN | COMMUNICATIONS | 2009Q3 | SUPPLIES AND MATERIALS | NaN | 07Â31 S1 DY090700018 | 07/01/09 | 07/3 |
| **8** | NaN | COMMUNICATIONS | 2009Q3 | SUPPLIES AND MATERIALS | NaN | 08Â31 S1 DY090800017 | 08/01/09 | 08/3 |

In [4]:
```
df.columns
```

Out[4]:
```
Index(['BIOGUIDE_ID', 'OFFICE', 'QUARTER', 'CATEGORY', 'DATE', 'PAYEE'
,
       'START DATE', 'END DATE', 'PURPOSE', 'AMOUNT', 'YEAR', 'TRANSCO
DE',
       'TRANSCODELONG', 'RECORDID', 'RECIP (orig.)'],
      dtype='object')
```

In [5]:
```
type(df['START DATE'][3])
```

Out[5]:
```
str
```

In [6]:
```
type(df['END DATE'][3])
```

Out[6]:
```
str
```

In [7]:
```
from datetime import datetime
df['COVERAGE PERIOD'] = pd.to_datetime(df['END DATE']) - pd.to_datetime(
df['COVERAGE PERIOD'].head()
num_rows = df['COVERAGE PERIOD'].count()
print(num_rows)
```

```
97921
```

In [8]:
```
df['COVERAGE PERIOD'].mean()
```

Out[8]:
```
Timedelta('20 days 21:39:53.909375')
```

In [9]:
```
std_dev = df['COVERAGE PERIOD'].std()
print(std_dev)
```

```
51 days 11:15:38.422354
```

In [10]:
```
sdev = float(str(std_dev).split()[0])
print(sdev)
```

```
51.0
```

In [11]:
```
type(sdev)
```

Out[11]:
```
float
```

In [12]:
```
#Now start processing all files

stddev_list = []
count_rows_list = []
for file in file_list[0:31]:
    df = pd.read_csv(file, low_memory=False)
    df['AMOUNT'] = df['AMOUNT'].apply(pd.to_numeric, errors='coerce')
    df = df[df['AMOUNT'] > 0]
    df['COVERAGE PERIOD'] = pd.to_datetime(df['END DATE']) - pd.to_datet
    df['COVERAGE PERIOD'].head()
    num_rows = df['COVERAGE PERIOD'].count()
    std_dev = df['COVERAGE PERIOD'].std()
    sdev = float(str(std_dev).split()[0])
    stddev_list.append(sdev)
    count_rows_list.append(num_rows)
```

In [13]:
```
print(stddev_list)
print(count_rows_list)
```

```
[51.0, 54.0, 62.0, 55.0, 49.0, 61.0, 72.0, 55.0, 83.0, 53.0, 64.0, 48.
0, 47.0, 63.0, 83.0, 55.0, 44.0, 55.0, 61.0, 47.0, 50.0, 59.0, 69.0, 4
7.0, 67.0, 47.0, 46.0, 50.0, 41.0, 72.0, 71.0]
[97921, 90880, 105953, 99111, 53019, 74561, 89867, 84987, 79959, 76252
, 90660, 77859, 69043, 66164, 80609, 71774, 72977, 64777, 76593, 71041
, 67134, 67256, 76788, 71419, 67947, 70847, 78037, 75425, 69490, 87786
, 99822]
```

In [14]:
```
print(len(stddev_list))
print(len(count_rows_list))
```

```
31
31
```

In [15]:
```
df = pd.read_csv('2017Q2-house-disburse-detail.csv', sep=',', engine = '
df.columns = ['BIOGUIDE_ID', 'OFFICE', 'QUARTER', 'PROGRAM', 'CATEGORY',
       'SORT SEQUENCE', 'DATE', 'TRANSCODE', 'PAYEE', 'START DATE',
       'END DATE', 'PURPOSE', 'AMOUNT', 'YEAR', 'RECORDID']
df['AMOUNT'] = df['AMOUNT'].apply(pd.to_numeric, errors='coerce')
df = df[df['AMOUNT'] > 0]  # Make sure payment is positive
#print(df['START DATE'].head())
#print("\n")
#print(df['END DATE'].head())
df['COVERAGE PERIOD'] = pd.to_datetime(df['END DATE'], dayfirst = True,
    df['START DATE'], dayfirst = True, errors='coerce')
print(df.head())
df = df[df['COVERAGE PERIOD'].notnull()]
print(df['COVERAGE PERIOD'].head())
num_rows = df['COVERAGE PERIOD'].count()
std_dev = df['COVERAGE PERIOD'].std()
sdev = str(std_dev).split()[0]
#print(sdev)
stddev_list.append(float(sdev))
count_rows_list.append(num_rows)

print(stddev_list)
print(count_rows_list)
print(len(stddev_list))
print(len(count_rows_list))
```

```
  BIOGUIDE_ID                      OFFICE QUARTER                PROGRA
M  \
0         NaN  2017 OFFICE OF THE SPEAKER  2017Q2  GENERAL EXPENDITURE
S
1         NaN  2017 OFFICE OF THE SPEAKER  2017Q2  GENERAL EXPENDITURE
```

```
S
2         NaN  2017 OFFICE OF THE SPEAKER  2017Q2  GENERAL EXPENDITURE
S
3         NaN  2017 OFFICE OF THE SPEAKER  2017Q2  GENERAL EXPENDITURE
S
4         NaN  2017 OFFICE OF THE SPEAKER  2017Q2  GENERAL EXPENDITURE
S

                  CATEGORY SORT SEQUENCE DATE TRANSCODE
PAYEE  \
0  PERSONNEL COMPENSATION                                    ALTHOUSE JOS
HUA S
1  PERSONNEL COMPENSATION                                     ANDRES DOUG
LAS R
2  PERSONNEL COMPENSATION                                     ANDREWS THO
MAS S
3  PERSONNEL COMPENSATION                                      ANTELL GEO
FFREY
4  PERSONNEL COMPENSATION                                  BENJAMIN WILLI
AM C.

  START DATE END DATE                          PURPOSE    AMOUNT  YEAR
\
0     4/1/17  6/30/17  CONSERVATIVE OUTREACH DIRECTOR  20000.01  2017
1     4/1/17  6/30/17                PRESS SECRETARY  27500.01  2017
2     4/1/17  6/30/17        MEMBER SERVICES DIRECTOR  32500.00  2017
3     4/1/17  6/30/17  ASST TO THE SPEAKER FOR POLICY  41250.00  2017
4     4/1/17  6/30/17            SYSTEM ADMINISTRATOR  13250.01  2017

   RECORDID COVERAGE PERIOD
0       NaN        177 days
1       NaN        177 days
2       NaN        177 days
3       NaN        177 days
4       NaN        177 days
0   177 days
1   177 days
2   177 days
3   177 days
4   177 days
Name: COVERAGE PERIOD, dtype: timedelta64[ns]
[51.0, 54.0, 62.0, 55.0, 49.0, 61.0, 72.0, 55.0, 83.0, 53.0, 64.0, 48.
0, 47.0, 63.0, 83.0, 55.0, 44.0, 55.0, 61.0, 47.0, 50.0, 59.0, 69.0, 4
7.0, 67.0, 47.0, 46.0, 50.0, 41.0, 72.0, 71.0, 93.0]
[97921, 90880, 105953, 99111, 53019, 74561, 89867, 84987, 79959, 76252
, 90660, 77859, 69043, 66164, 80609, 71774, 72977, 64777, 76593, 71041
, 67134, 67256, 76788, 71419, 67947, 70847, 78037, 75425, 69490, 87786
, 99822, 95431]
32
32
```

In [16]:
```
for file in file_list[32:33 ]:
    df = pd.read_csv(file, sep=',', engine = 'python')
    #print(df.head())
    #print(df.describe())
    df['AMOUNT'] = df['AMOUNT'].apply(pd.to_numeric, errors='coerce')
    df = df[df['AMOUNT'] > 0]
    df['COVERAGE PERIOD'] = pd.to_datetime(df['END DATE'], dayfirst = Tr
    df['START DATE'], dayfirst = True, errors='coerce')
    #df['COVERAGE PERIOD'].head()
    num_rows = df['COVERAGE PERIOD'].count()
    #std_dev = df['COVERAGE PERIOD'].std()
    sdev = float(str(std_dev).split()[0])
    print("sdev = ", sdev)
    print("num_rows = ", num_rows)
    stddev_list.append(sdev)
    count_rows_list.append(num_rows)
```

```
sdev =  93.0
num_rows =  94826
```

In [17]:
```
print(stddev_list)
print(count_rows_list)
print(len(stddev_list))
print(len(count_rows_list))
```

```
[51.0, 54.0, 62.0, 55.0, 49.0, 61.0, 72.0, 55.0, 83.0, 53.0, 64.0, 48.
0, 47.0, 63.0, 83.0, 55.0, 44.0, 55.0, 61.0, 47.0, 50.0, 59.0, 69.0, 4
7.0, 67.0, 47.0, 46.0, 50.0, 41.0, 72.0, 71.0, 93.0, 93.0]
[97921, 90880, 105953, 99111, 53019, 74561, 89867, 84987, 79959, 76252
, 90660, 77859, 69043, 66164, 80609, 71774, 72977, 64777, 76593, 71041
, 67134, 67256, 76788, 71419, 67947, 70847, 78037, 75425, 69490, 87786
, 99822, 95431, 94826]
33
33
```

In [18]:
```
for file in file_list[33: ]:
    df = pd.read_csv(file, sep=',', engine = 'python')
    #print(df.head())
    #print(df.describe())
    df['AMOUNT'] = df['AMOUNT'].apply(pd.to_numeric, errors='coerce')
    df = df[df['AMOUNT'] > 0]
    df['COVERAGE PERIOD'] = pd.to_datetime(df['END DATE'], dayfirst = Tr
    df['START DATE'], dayfirst = True, errors='coerce')
    #df['COVERAGE PERIOD'].head()
    num_rows = df['COVERAGE PERIOD'].count()
    #std_dev = df['COVERAGE PERIOD'].std()
    sdev = float(str(std_dev).split()[0])
    print("sdev = ", sdev)
    print("num_rows = ", num_rows)
    stddev_list.append(sdev)
    count_rows_list.append(num_rows)
```

```
sdev =  93.0
num_rows =  91104
sdev =  93.0
num_rows =  68421
```

In [19]:
```
print(stddev_list)
print(count_rows_list)
print(len(stddev_list))
print(len(count_rows_list))
```

```
[51.0, 54.0, 62.0, 55.0, 49.0, 61.0, 72.0, 55.0, 83.0, 53.0, 64.0, 48.
0, 47.0, 63.0, 83.0, 55.0, 44.0, 55.0, 61.0, 47.0, 50.0, 59.0, 69.0, 4
7.0, 67.0, 47.0, 46.0, 50.0, 41.0, 72.0, 71.0, 93.0, 93.0, 93.0, 93.0]
[97921, 90880, 105953, 99111, 53019, 74561, 89867, 84987, 79959, 76252
, 90660, 77859, 69043, 66164, 80609, 71774, 72977, 64777, 76593, 71041
, 67134, 67256, 76788, 71419, 67947, 70847, 78037, 75425, 69490, 87786
, 99822, 95431, 94826, 91104, 68421]
35
35
```

In [20]:
```
numerator = 0
for i in range(35):
    numerator += (count_rows_list[i] - 1) * (stddev_list[i] ** 2)

denominator = sum(count_rows_list)  - 35
pooled_variance = numerator/denominator
print(pooled_variance)
pooled_stddev = pooled_variance ** 0.5
print(pooled_stddev)
```

```
4135.057162414594
64.30441013192325
```

- The standard deviation of COVERAGE in days is 64.30441013192325 *

In [ ]:
```

```