final

December 5, 2020

Analysis of Senator Stock Trading

Patrick Cao

Introduction

Insider trading, or trading financial assets using information that is not open to the public, is punishable by law in the United States. It is a serious violation – a maximum penalty of 20 years in prison, and up to $20 million in fines. The Securities and Exchange Commission (SEC) is responsible for identifying and conducting investigations regarding insider trading.

With that being said, U.S. senators (and other congresspeople) often have access to information about upcoming policies or laws before the general public, yet are still allowed to trade stocks. More recently, a few senators have been investigated by the SEC for insider trading using knowledge given to them about the COVID-19 pandemic.

This tutorial aims to identify potential insider trading, or suspicious trades, by U.S. senators. We will be using data from https://senatestockwatcher.com/, which itself compiles data from https://efdsearch.senate.gov/. Senate stock watcher aggregates financial disclosures from all senators into one nice dataset.

Let's first start by downloading the dataset. While Senate Stock Watcher provides an API for up to date datasets, we can just download a csv file as well. Download the "All Transactions" csv file from https://senatestockwatcher.com/api.html.

Let's load that into a pandas dataframe.

[1]:
```python
import pandas as pd
import numpy as np
import datetime

df = pd.read_csv('all_transactions.csv')
df.head()
```

[1]:
```
  transaction_date   owner ticker              asset_description asset_type  \
0       11/16/2020  Spouse     BA             The Boeing Company      Stock   
1       11/16/2020  Spouse      V                      Visa Inc.      Stock   
2       11/24/2020  Spouse   COST  Costco Wholesale Corporation      Stock   
3       11/04/2020  Spouse   COST  Costco Wholesale Corporation      Stock   
4       11/09/2020  Spouse     BA             The Boeing Company      Stock   
```

```
           type              amount comment      senator  \
0      Purchase  $15,001 - $50,000       R  Pat Roberts
1      Purchase  $15,001 - $50,000     555  Pat Roberts
2      Purchase  $15,001 - $50,000     555  Pat Roberts
3   Sale (Full)  $15,001 - $50,000     555  Pat Roberts
4   Sale (Full)  $15,001 - $50,000     sep  Pat Roberts

                                            ptr_link
0  https://efdsearch.senate.gov/search/view/ptr/b…
1  https://efdsearch.senate.gov/search/view/ptr/b…
2  https://efdsearch.senate.gov/search/view/ptr/b…
3  https://efdsearch.senate.gov/search/view/ptr/b…
4  https://efdsearch.senate.gov/search/view/ptr/b…
```

Let's clean this up a little by limiting and purchases and sales only, and dropping unnecessary columns.

[2]:

```
start_date = '2016-01-01'
end_date = '2020-11-01'

# Turn transaction_date from string into datetime
df['transaction_date'] = pd.to_datetime(df['transaction_date'])

# Remove (Full) and (Partial) after Sale type
df['type'] = df['type'].apply(lambda x: str(x).split()[0])

# Limit to only purchases and sales
df = df[(df['type'] == 'Purchase') | (df['type'] == 'Sale')]

# limit the dataset to a date range
df = df[(df['transaction_date'] > start_date) & (df['transaction_date'] <␣
 ↪end_date)]

df = df.drop(['asset_description', 'comment', 'ptr_link'], axis=1)
df.head()
```

[2]:

```
   transaction_date   owner ticker      asset_type      type  \
76       2020-10-13   Joint     --  Corporate Bond      Sale
78       2019-04-05  Spouse    NVS           Stock      Sale
79       2020-10-16   Child   NVDA           Stock  Purchase
80       2020-10-16   Child   WDAY           Stock  Purchase
81       2020-10-16   Child   NVDA           Stock  Purchase

               amount            senator
76  $1,001 - $15,000  Sheldon Whitehouse
78  $1,001 - $15,000     William Cassidy
79  $1,001 - $15,000         Ron L Wyden
```

```
80  $1,001 - $15,000         Ron L Wyden
81  $1,001 - $15,000         Ron L Wyden
```

We got our senator stock trading data, but now we need actual historical market data to visualize these trades. Let's use yfinance, a python Yahoo! Finance library, to get historical market data on SPY, an ETF tracking the S&P500.

[3]:
```python
import yfinance as yf

spy = yf.Ticker('SPY')
spy_hist = spy.history(period='max')
spy_hist.tail()
```

[3]:
```
                  Open        High         Low       Close    Volume  \
Date
2020-11-30  362.829987  363.119995  359.170013  362.059998  83872700
2020-12-01  365.570007  367.679993  364.929993  366.019989  74505000
2020-12-02  364.820007  366.959991  364.200012  366.790009  45927000
2020-12-03  366.679993  368.190002  365.500000  366.690002  62882000
2020-12-04  367.320007  369.850006  367.220001  369.850006  50676700

            Dividends  Stock Splits
Date
2020-11-30        0.0             0
2020-12-01        0.0             0
2020-12-02        0.0             0
2020-12-03        0.0             0
2020-12-04        0.0             0
```

Neat! Now we can start putting these two things together.

Exploratory Data Analysis

Let's first aggregate all the senator trading data together, and find the overall trading performance of the senate. We should somehow plot overall senator trading performance against the S&P500.

Senator Data

It's a little tough to track performance exactly. The size of each trade isn't an exact number – it's a range. Let's check out what kinds of values we're working with here.

[4]:
```python
df['amount'].unique()
```

[4]:
```
array(['$1,001 - $15,000', '$15,001 - $50,000', '$1,000,001 - $5,000,000',
       '$250,001 - $500,000', '$500,001 - $1,000,000',
       '$50,001 - $100,000', '$100,001 - $250,000',
       '$5,000,001 - $25,000,000'], dtype=object)
```

Yikes. The data doesn't have the exact value of securities traded, the best thing we can probably do is get the lower bound, upper bound, and middle value of each trade, and use that to estimate the value of each trade.

[5]:
```python
# Returns a tuple of [lower bound, middle, upper bound] for the given category
def getBounds(row):

    amount = row[0]
    trade_type = row[1]

    # This amount shows up once in the entire dataset. Let's just assume the trade is $50m for now. It won't affect our analysis too much.
    if amount == 'Over $50,000,000':
        return (5e7, 5e7, 5e7)

    split = amount.split(' - ')
    lower = int(split[0][1:].replace(',', ''))
    upper = int(split[1][1:].replace(',', ''))
    mid = lower + ((upper - lower) / 2)

    if trade_type == 'Purchase':
        return [lower, mid, upper]
    return [-lower, -mid, -upper]

bounds = pd.Series(df[['amount', 'type']].apply(getBounds, axis=1))

df['lower_amt'] = bounds.apply(lambda x: x[0])
df['mid_amt'] = bounds.apply(lambda x: x[1])
df['upper_amt'] = bounds.apply(lambda x: x[2])

df
```

[5]:
```
     transaction_date   owner ticker      asset_type      type  \
76         2020-10-13   Joint     --  Corporate Bond      Sale
78         2019-04-05  Spouse    NVS           Stock      Sale
79         2020-10-16   Child   NVDA           Stock  Purchase
80         2020-10-16   Child   WDAY           Stock  Purchase
81         2020-10-16   Child   NVDA           Stock  Purchase
…                 …       …      …               …         …
7468       2016-01-06    Self     UA           Stock  Purchase
7470       2016-11-23   Joint     --           Stock      Sale
7471       2016-01-04   Joint     --           Stock      Sale
7472       2016-01-05   Joint     --           Stock      Sale
7473       2016-01-05   Joint     --           Stock  Purchase

                 amount            senator  lower_amt  mid_amt  upper_amt
76    $1,001 - $15,000  Sheldon Whitehouse      -1001  -8000.5     -15000
78    $1,001 - $15,000     William Cassidy      -1001  -8000.5     -15000
79    $1,001 - $15,000         Ron L Wyden       1001   8000.5      15000
80    $1,001 - $15,000         Ron L Wyden       1001   8000.5      15000
```

```
81       $1,001 - $15,000         Ron L Wyden       1001   8000.5      15000
…                       …                   …         …      …        …
7468   $50,001 - $100,000         John Hoeven      50001  75000.5     100000
7470     $1,001 - $15,000    Patrick J Toomey      -1001  -8000.5     -15000
7471     $1,001 - $15,000    Patrick J Toomey      -1001  -8000.5     -15000
7472    $15,001 - $50,000    Patrick J Toomey     -15001 -32500.5     -50000
7473     $1,001 - $15,000    Patrick J Toomey       1001   8000.5      15000

[6539 rows x 10 columns]
```

Alright, now we got some rough numerical bounds of each transaction, we can group trades together by date and find the aggregate for each day.

[6]:

```
aggregate_action = df.groupby([pd.Grouper(key='transaction_date', freq='D'),␣
 ↪'asset_type', 'type']).sum()

# Groupby makes transaction_date and type an index, we want to convert those␣
 ↪back into a column
aggregate_action.reset_index(inplace=True)

aggregate_buys = aggregate_action[aggregate_action['type'] == 'Purchase']
aggregate_sells = aggregate_action[aggregate_action['type'] == 'Sale']

# print(aggregate_buys.head())
# print(aggregate_sells.head())
print(aggregate_action.head())
```

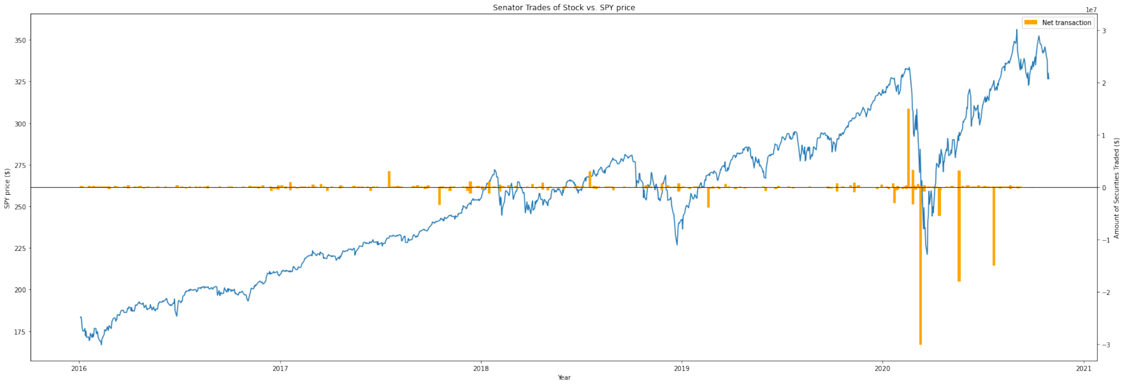
```
  transaction_date          asset_type      type  lower_amt    mid_amt  \
0       2016-01-04               Stock      Sale      -1001    -8000.5
1       2016-01-05  Municipal Security  Purchase      15001    32500.5
2       2016-01-05               Stock  Purchase      17017   136008.5
3       2016-01-05               Stock      Sale     -33005   -89002.5
4       2016-01-06               Stock  Purchase      88011   204005.5

   upper_amt
0     -15000
1      50000
2     255000
3    -145000
4     320000
```

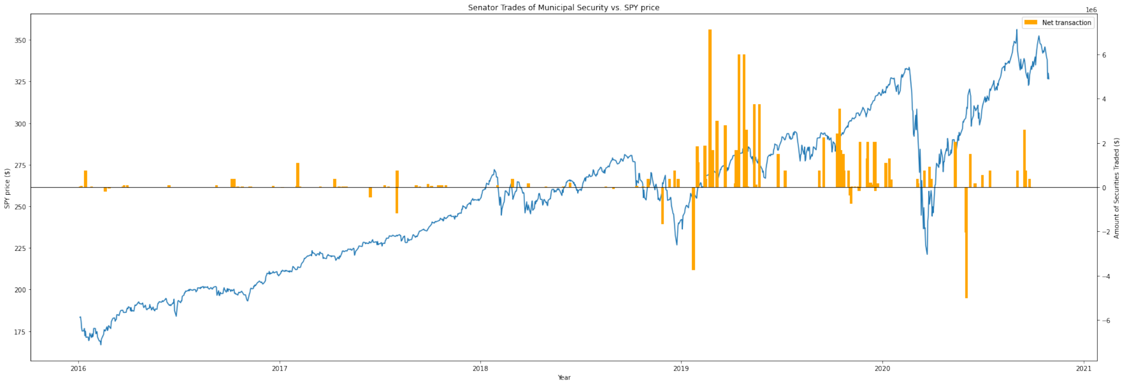
We are now ready to plot senator trades against the market.

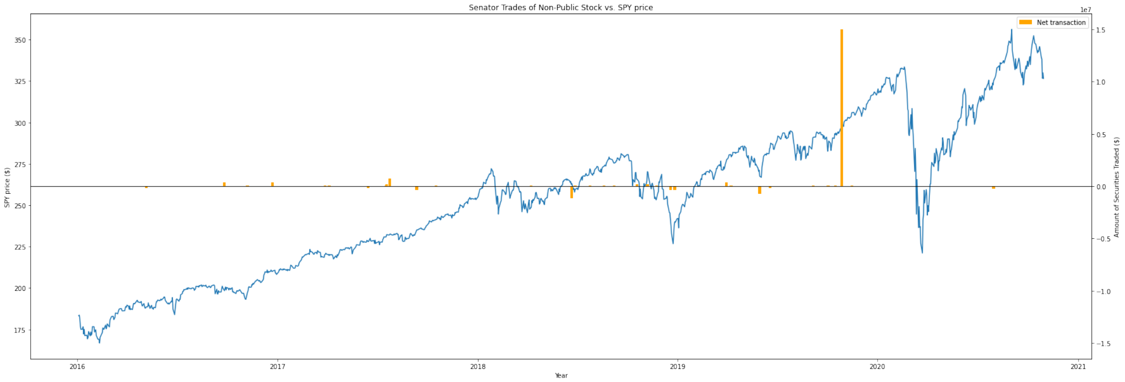
[7]:

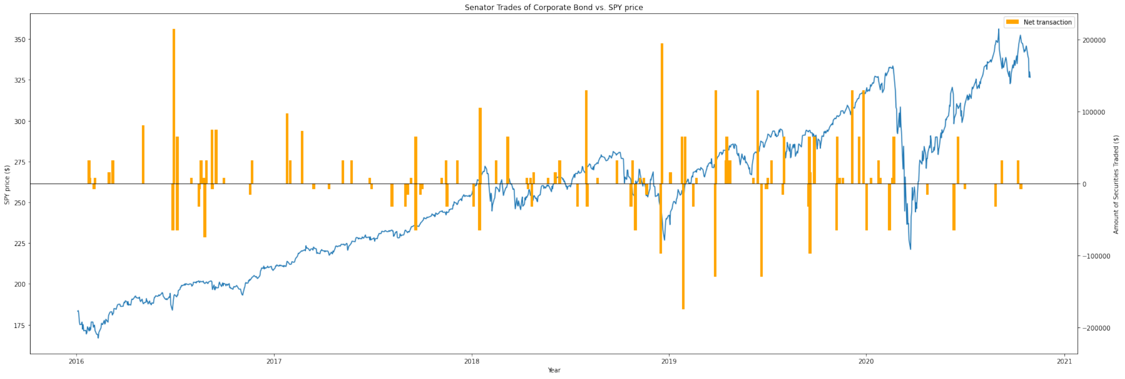
```
import matplotlib.pyplot as plt

spy_hist['Date'] = spy_hist.index
market = spy_hist[(spy_hist['Date'] > start_date) & (spy_hist['Date'] <␣
 ↪end_date)]
```

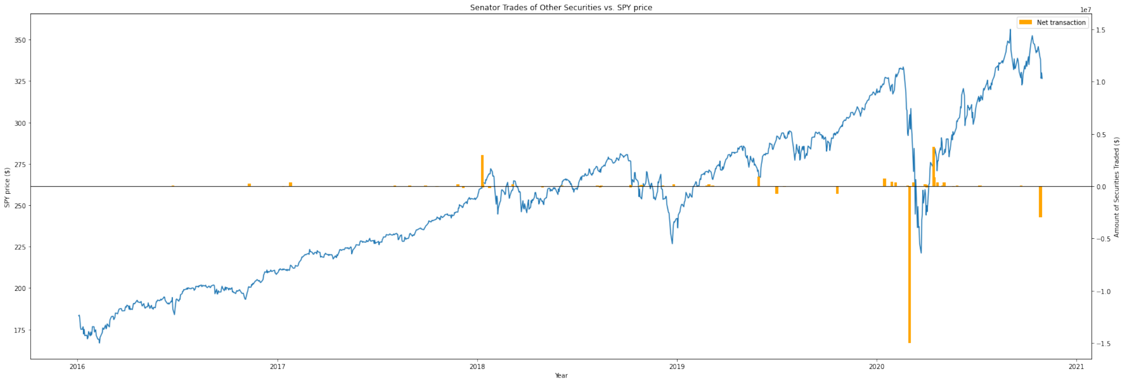
```
for asset_type in aggregate_action['asset_type'].unique():

    # buys = aggregate_buys[aggregate_buys['asset_type'] == asset_type]
    # sells = aggregate_sells[aggregate_sells['asset_type'] == asset_type]
    total = aggregate_action[aggregate_action['asset_type'] == asset_type]

    # We have a lot of data, so we should make the figure very large
    plt.figure(figsize=(30,10))

    plt.plot(market['Date'], market['Close'])

    plt.xlabel('Year')
    plt.ylabel('SPY price ($)')
    ax2 = plt.twinx()

    # scale bar graph limits s.t. 0 is centered in graph
    limit = max(abs(total['mid_amt'].min()), total['mid_amt'].max()) * 1.1
    ax2.set_ylim(-limit, limit)
    ax2.spines['bottom'].set_position(('data', 0))

    plt.ylabel('Amount of Securities Traded ($)')

    # plt.bar(buys['transaction_date'], buys['mid_amt'], width=5, label='buys',␣
 ↪color='green')
    # plt.bar(sells['transaction_date'], sells['mid_amt'], width=5,␣
 ↪label='sells', color='red')
    plt.bar(total['transaction_date'], total['mid_amt'], width=5, label='Net␣
 ↪transaction', color='orange')

    plt.title('Senator Trades of ' + str(asset_type) + ' vs. SPY price')
    plt.legend()

    plt.show()
```

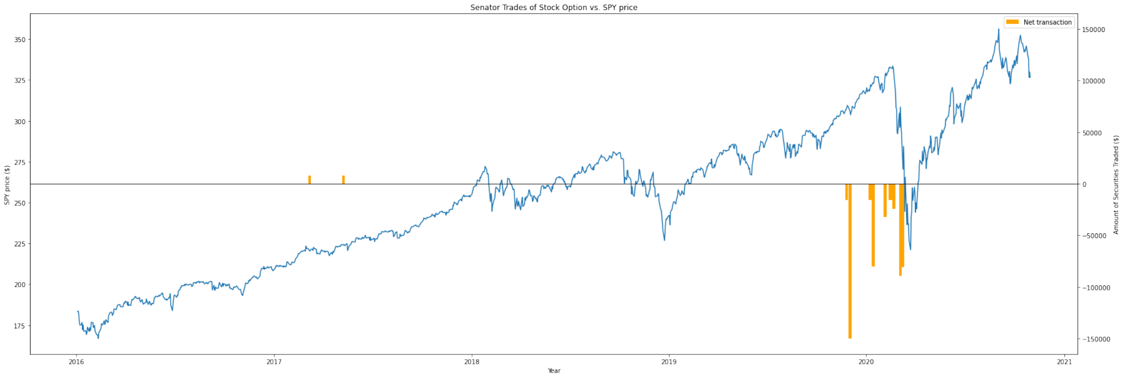



Senator Trades of Municipal Security vs. SPY price
Net transaction
SPY price ($)
Amount of Securities Traded ($)
1e6
350
325
300
275
250
225
200
175
6
4
2
0
−2
−4
−6
2016
2017
2018
2019
2020
2021
Year


Senator Trades of Non-Public Stock vs. SPY price
Net transaction
SPY price ($)
Amount of Securities Traded ($)
1e7
350
325
300
275
250
225
200
175
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
2016
2017
2018
2019
2020
2021
Year


Senator Trades of Corporate Bond vs. SPY price
Net transaction
SPY price ($)
Amount of Securities Traded ($)
350
325
300
275
250
225
200
175
200000
100000
0
−100000
−200000
2016
2017
2018
2019
2020
2021
Year





Let's also plot the net amount traded vs. future SPY price as well. We first need

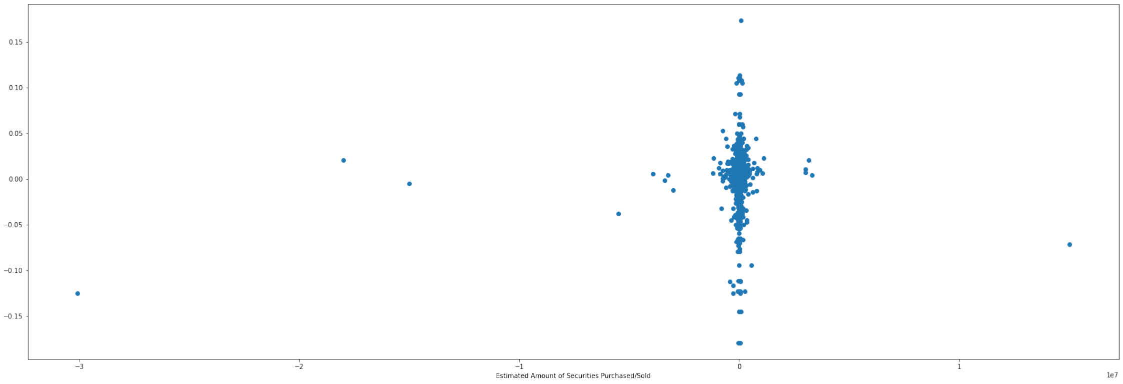
[8]:

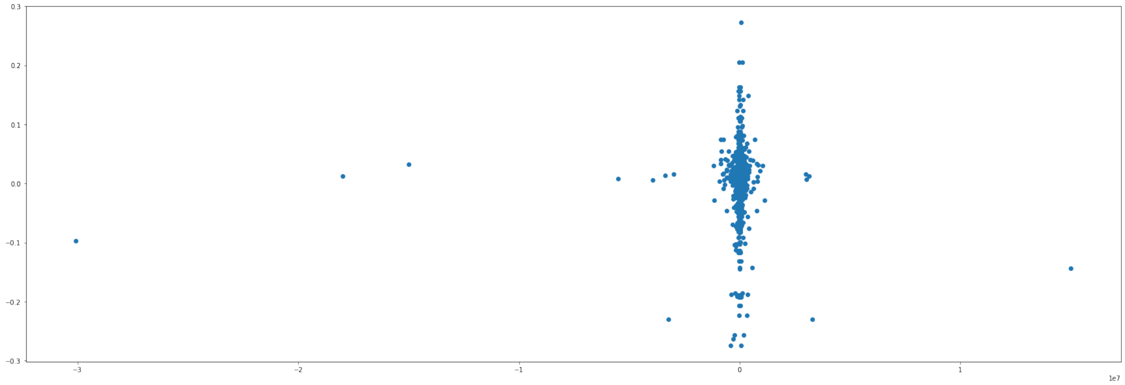
```
import pandas_market_calendars as mcal
from datetime import datetime

nyse = mcal.get_calendar('NYSE')
market_open_days = nyse.valid_days(start_date=start_date, end_date=datetime.
 ↪today().strftime('%Y-%m-%d'))

# returns a function that gets the spy price num_days from today
def get_future_spy_price(num_days):
    def h(today):
        # calculate current spy data
        today_index = market_open_days.get_loc(today, method='nearest')
        today_close = spy_hist.iloc[spy_hist.index.get_loc(today,␣
 ↪method='nearest')]['Close']

        # calculate future date and get spy data
        future_index = today_index + num_days
        future_date = market_open_days[future_index]
        future_close = spy_hist.iloc[spy_hist.index.get_loc(future_date,␣
↪method='nearest')]['Close']
        return (future_close - today_close) / today_close
    return h

aggregate_action['5_day_spy_pct_change'] = aggregate_action['transaction_date'].
↪apply(get_future_spy_price(5))
aggregate_action['15_day_spy_pct_change'] =␣
↪aggregate_action['transaction_date'].apply(get_future_spy_price(15))
# aggregate_total['30_day_spy_pct_change'] =␣
↪aggregate_total['transaction_date'].apply(get_future_spy_price(30))
```

[26]:

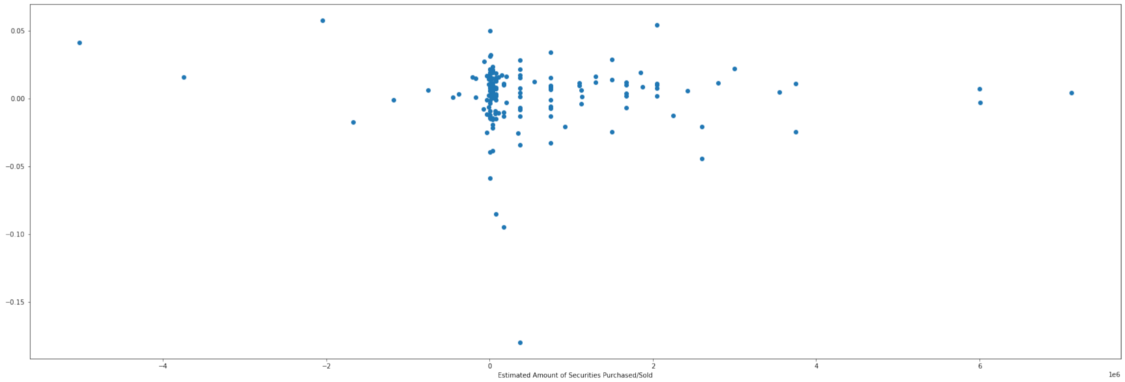
```
test = aggregate_action[abs(aggregate_action['lower_amt']) > 0]

for asset_type in test['asset_type'].unique():
    print(asset_type)
    testx = test[test['asset_type'] == asset_type]
    print(test.head())
    plt.figure(figsize=(30,10))
    plt.scatter(testx['mid_amt'], testx['5_day_spy_pct_change'])
    plt.xlabel('Estimated Amount of Securities Purchased/Sold')
    plt.show()

    plt.figure(figsize=(30,10))
    plt.scatter(testx['mid_amt'], testx['15_day_spy_pct_change'])
    plt.show()

    # plt.figure(figsize=(30,10))
    # plt.scatter(test['mid_amt'], test['30_day_spy_pct_change'])
    # plt.show()
```

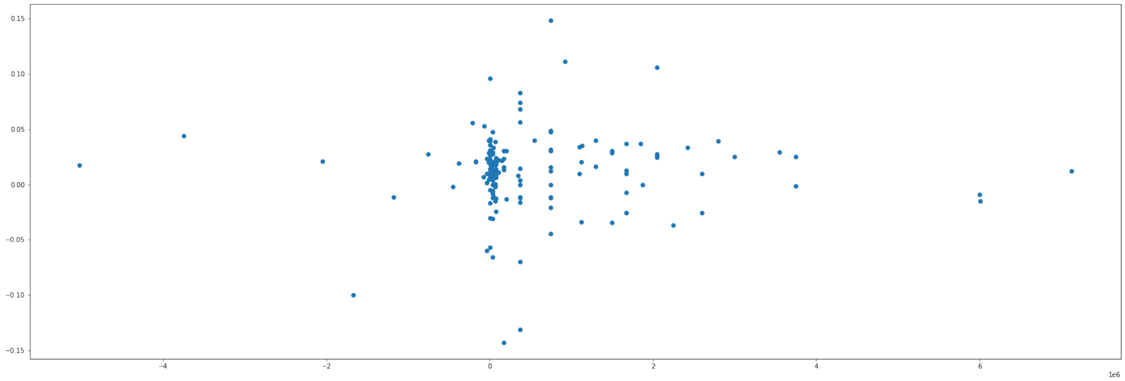
```
Stock
  transaction_date          asset_type      type  lower_amt    mid_amt  \
0       2016-01-04               Stock      Sale      -1001    -8000.5
1       2016-01-05  Municipal Security  Purchase      15001    32500.5
2       2016-01-05               Stock  Purchase      17017   136008.5
3       2016-01-05               Stock      Sale     -33005   -89002.5
4       2016-01-06               Stock  Purchase      88011   204005.5

   upper_amt  5_day_spy_pct_change  15_day_spy_pct_change
0     -15000             -0.044324              -0.053826
1      50000             -0.038240              -0.065703
2     255000             -0.038240              -0.065703
3    -145000             -0.038240              -0.065703
4     320000             -0.050246              -0.048838
```



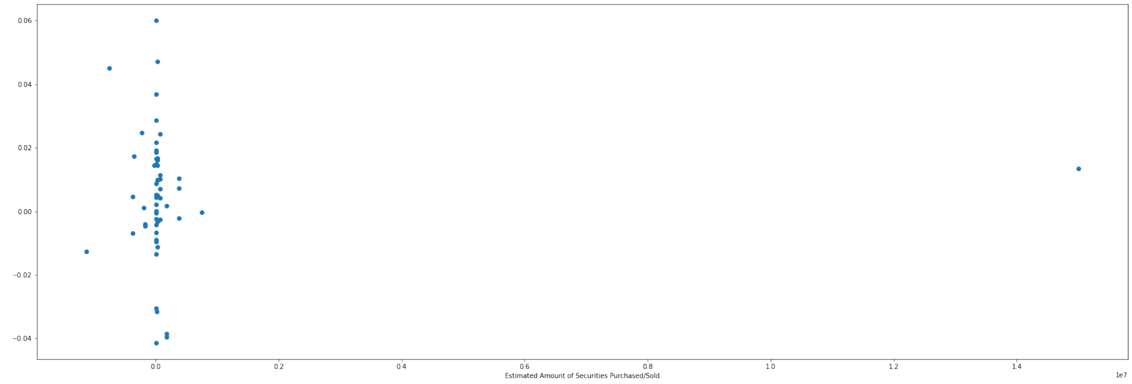


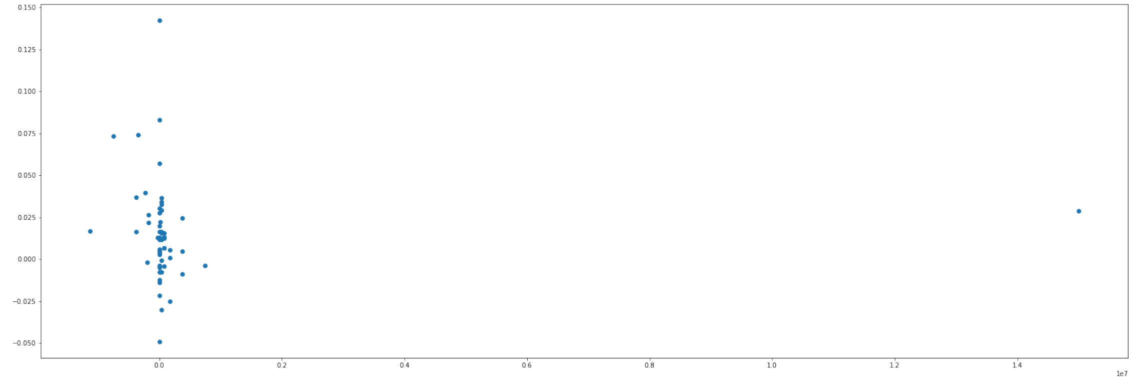
```
Municipal Security
  transaction_date          asset_type      type  lower_amt   mid_amt  \
0       2016-01-04               Stock      Sale      -1001   -8000.5
1       2016-01-05  Municipal Security  Purchase      15001   32500.5
2       2016-01-05               Stock  Purchase      17017  136008.5
3       2016-01-05               Stock      Sale     -33005  -89002.5
4       2016-01-06               Stock  Purchase      88011  204005.5

   upper_amt  5_day_spy_pct_change  15_day_spy_pct_change
0     -15000             -0.044324              -0.053826
1      50000             -0.038240              -0.065703
2     255000             -0.038240              -0.065703
3    -145000             -0.038240              -0.065703
4     320000             -0.050246              -0.048838
```



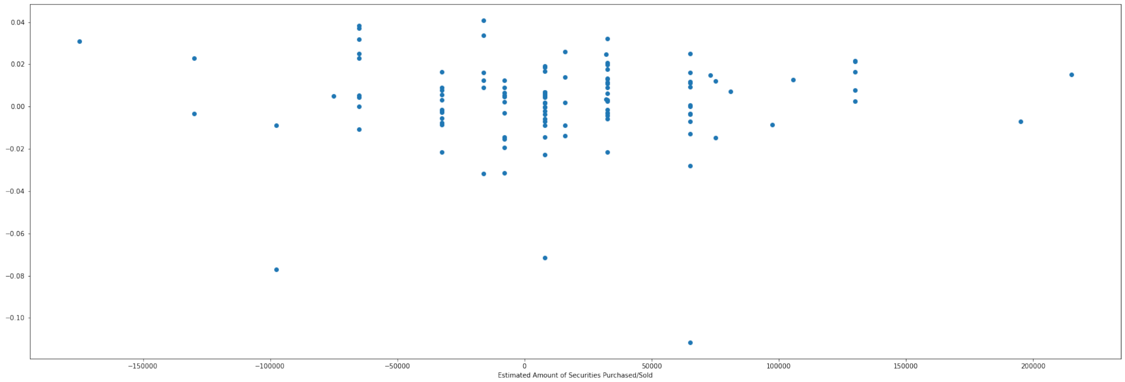


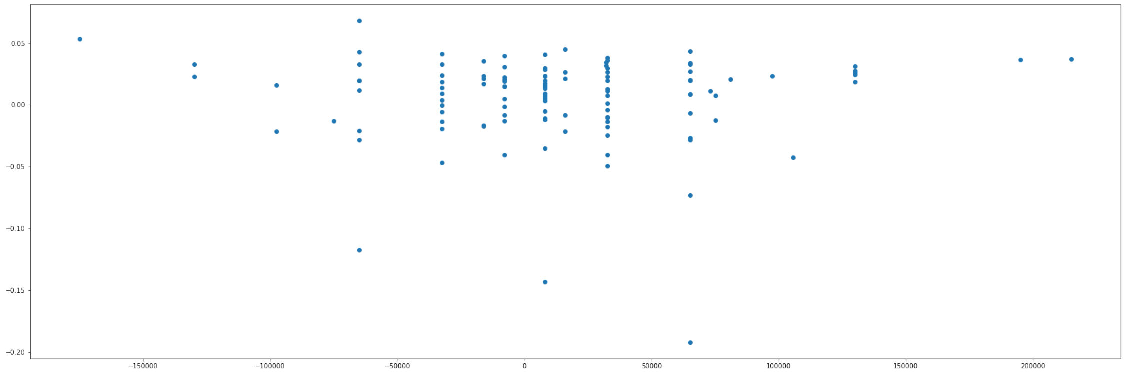
```
Non-Public Stock
  transaction_date          asset_type      type  lower_amt   mid_amt  \
0       2016-01-04               Stock      Sale      -1001   -8000.5
1       2016-01-05  Municipal Security  Purchase      15001   32500.5
2       2016-01-05               Stock  Purchase      17017  136008.5
3       2016-01-05               Stock      Sale     -33005  -89002.5
4       2016-01-06               Stock  Purchase      88011  204005.5

   upper_amt  5_day_spy_pct_change  15_day_spy_pct_change
0     -15000             -0.044324              -0.053826
1      50000             -0.038240              -0.065703
2     255000             -0.038240              -0.065703
3    -145000             -0.038240              -0.065703
4     320000             -0.050246              -0.048838
```



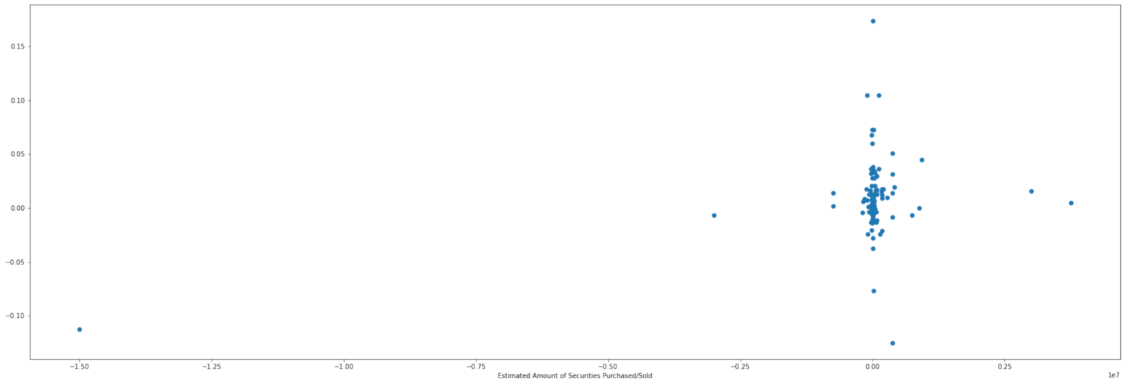


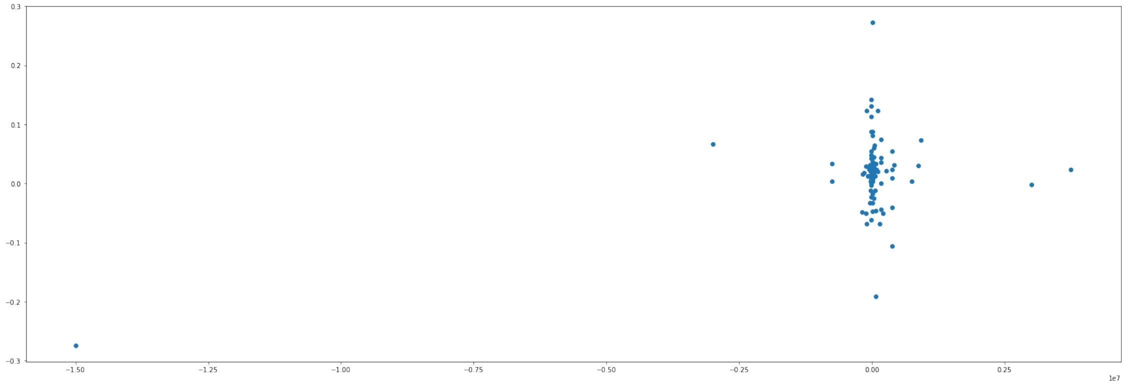
```
Corporate Bond
  transaction_date          asset_type      type  lower_amt   mid_amt  \
0       2016-01-04               Stock      Sale      -1001   -8000.5
1       2016-01-05  Municipal Security  Purchase      15001   32500.5
2       2016-01-05               Stock  Purchase      17017  136008.5
3       2016-01-05               Stock      Sale     -33005  -89002.5
4       2016-01-06               Stock  Purchase      88011  204005.5

   upper_amt  5_day_spy_pct_change  15_day_spy_pct_change
0     -15000             -0.044324              -0.053826
1      50000             -0.038240              -0.065703
2     255000             -0.038240              -0.065703
3    -145000             -0.038240              -0.065703
4     320000             -0.050246              -0.048838
```



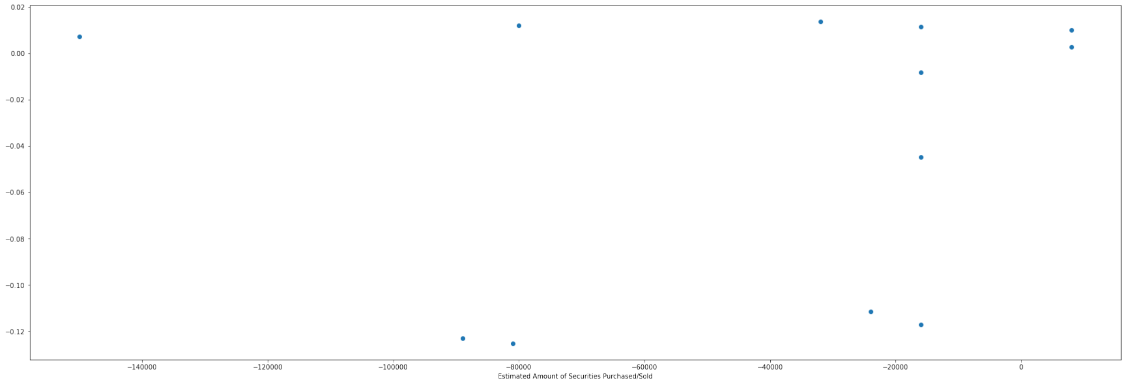


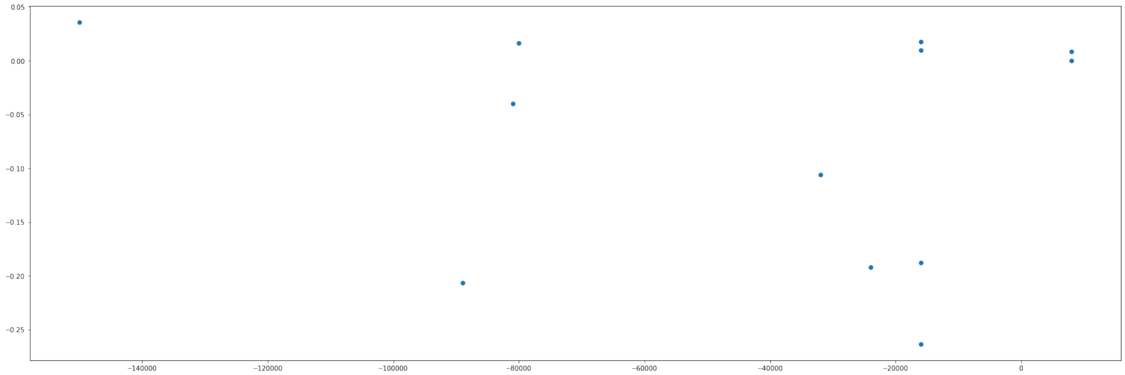
```
Other Securities
  transaction_date          asset_type      type  lower_amt   mid_amt  \
0       2016-01-04               Stock      Sale      -1001   -8000.5
1       2016-01-05  Municipal Security  Purchase      15001   32500.5
2       2016-01-05               Stock  Purchase      17017  136008.5
3       2016-01-05               Stock      Sale     -33005  -89002.5
4       2016-01-06               Stock  Purchase      88011  204005.5

   upper_amt  5_day_spy_pct_change  15_day_spy_pct_change
0     -15000             -0.044324              -0.053826
1      50000             -0.038240              -0.065703
2     255000             -0.038240              -0.065703
3    -145000             -0.038240              -0.065703
4     320000             -0.050246              -0.048838
```





```
Stock Option
  transaction_date          asset_type      type  lower_amt   mid_amt  \
0       2016-01-04               Stock      Sale      -1001   -8000.5
1       2016-01-05  Municipal Security  Purchase      15001   32500.5
2       2016-01-05               Stock  Purchase      17017  136008.5
3       2016-01-05               Stock      Sale     -33005  -89002.5
4       2016-01-06               Stock  Purchase      88011  204005.5

   upper_amt  5_day_spy_pct_change  15_day_spy_pct_change
0     -15000             -0.044324              -0.053826
1      50000             -0.038240              -0.065703
2     255000             -0.038240              -0.065703
3    -145000             -0.038240              -0.065703
4     320000             -0.050246              -0.048838
```





Grouping By Senator

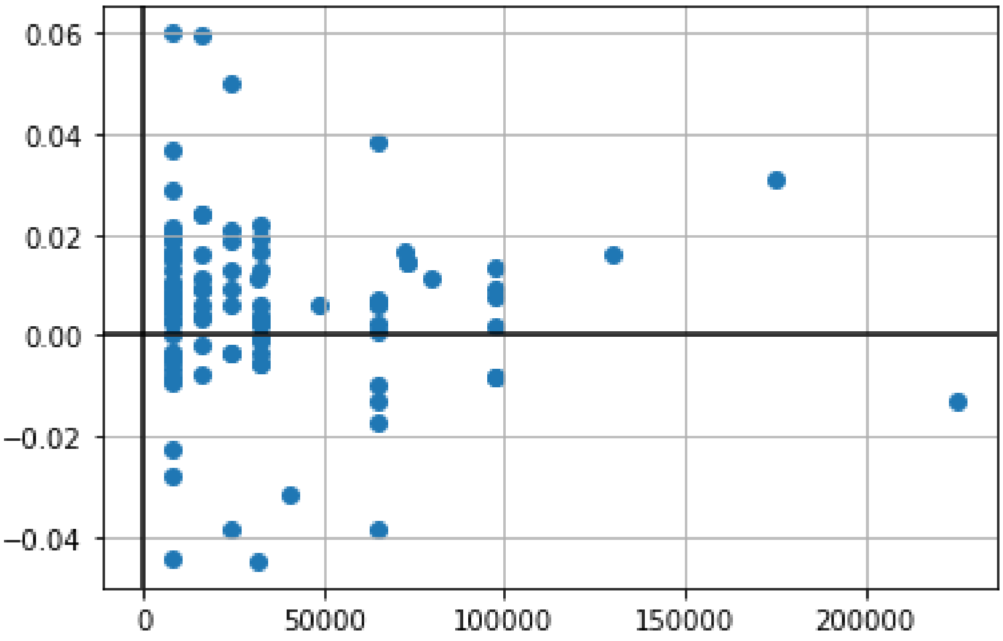
```
[10]: aggregate_action_by_senator = df.groupby([pd.Grouper(key='transaction_date',␣
       ↪freq='D'), 'asset_type', 'type', 'senator']).sum()

      # Groupby makes transaction_date and type an index, we want to convert those␣
       ↪back into a column
      aggregate_action_by_senator.reset_index(inplace=True)

      # limit the dataset to a date range
      aggregate_action_by_senator =␣
       ↪aggregate_action_by_senator[(aggregate_action_by_senator['transaction_date']␣
       ↪> start_date) & (aggregate_action_by_senator['transaction_date'] < end_date)]

      aggregate_action_by_senator['mid_amt'] = aggregate_action_by_senator.
       ↪apply(lambda row: row['mid_amt'] if row['type'] == 'Purchase' else␣
       ↪-row['mid_amt'], axis=1)
      aggregate_action_by_senator['lower_amt'] = aggregate_action_by_senator.
       ↪apply(lambda row: row['lower_amt'] if row['type'] == 'Purchase' else␣
       ↪-row['lower_amt'], axis=1)
```

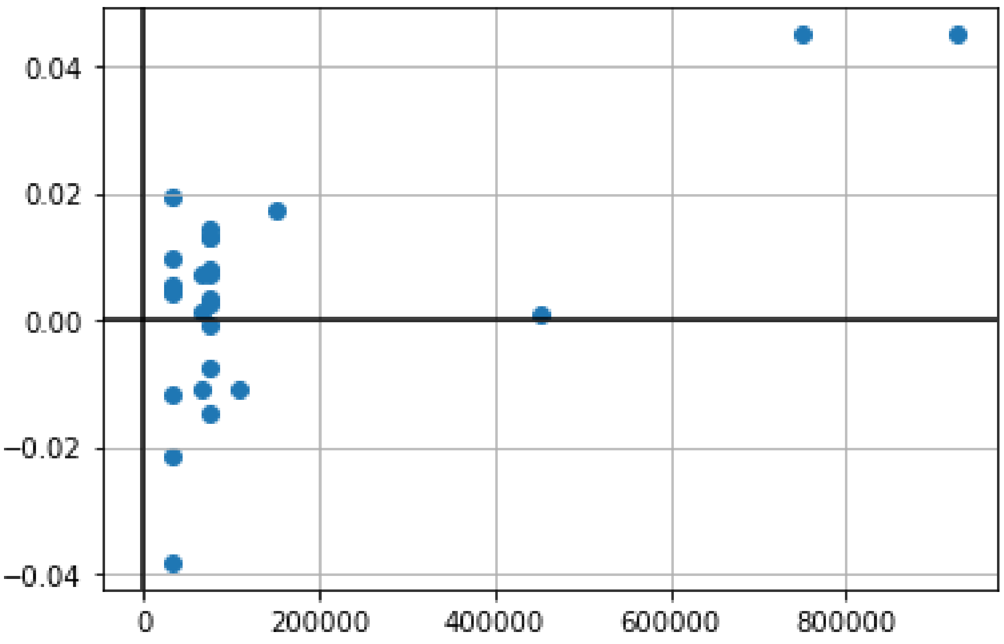
```
aggregate_action_by_senator['upper_amt'] = aggregate_action_by_senator.
 ↪apply(lambda row: row['upper_amt'] if row['type'] == 'Purchase' else␣
 ↪-row['upper_amt'], axis=1)

aggregate_action_by_senator['5_day_spy_pct_change'] =␣
 ↪aggregate_action_by_senator['transaction_date'].
 ↪apply(get_future_spy_price(5))
# aggregate_total['15_day_spy_pct_change'] =␣
 ↪aggregate_total['transaction_date'].apply(get_future_spy_price(15))


    # plt.figure(figsize=(30,10))
    # plt.scatter(senator_actions['mid_amt'], testx['5_day_spy_pct_change'])
    # plt.xlabel('Estimated Amount of Securities Purchased/Sold')
    # plt.show()

    # plt.figure(figsize=(30,10))
    # plt.scatter(test['mid_amt'], test['15_day_spy_pct_change'])
    # plt.show()
```

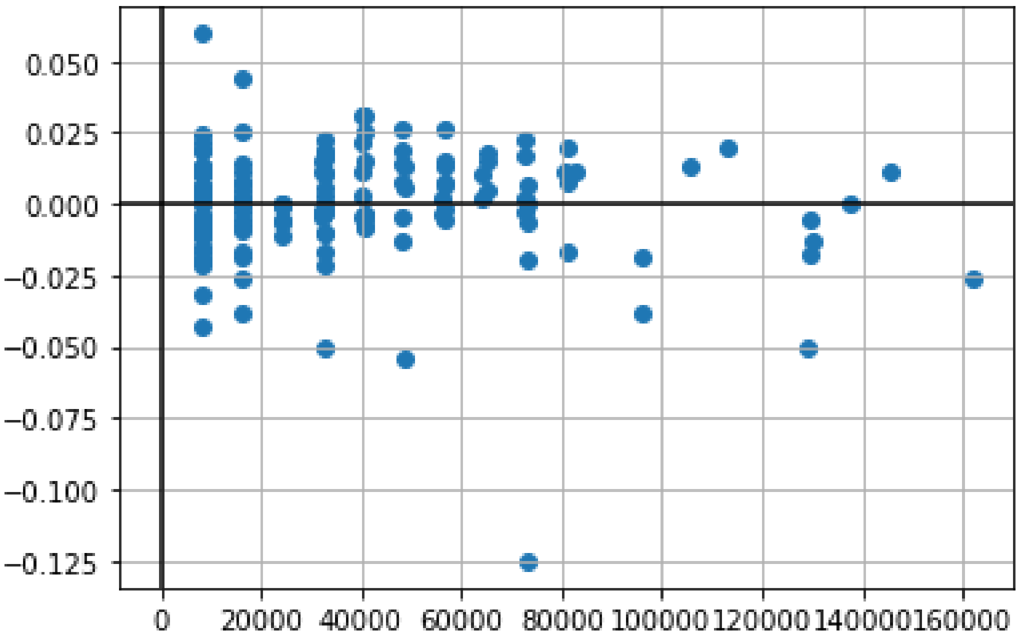
```
[11]: print(aggregate_action_by_senator)

for senator in aggregate_action_by_senator['senator'].unique():

    senator_actions =␣
 ↪aggregate_action_by_senator[aggregate_action_by_senator['senator'] ==␣
 ↪senator]

    plt.axvline(0, color='black')
    plt.axhline(0, color='black')
    plt.grid()

    plt.scatter(senator_actions['mid_amt'],␣
 ↪senator_actions['5_day_spy_pct_change'])
    plt.show()
```

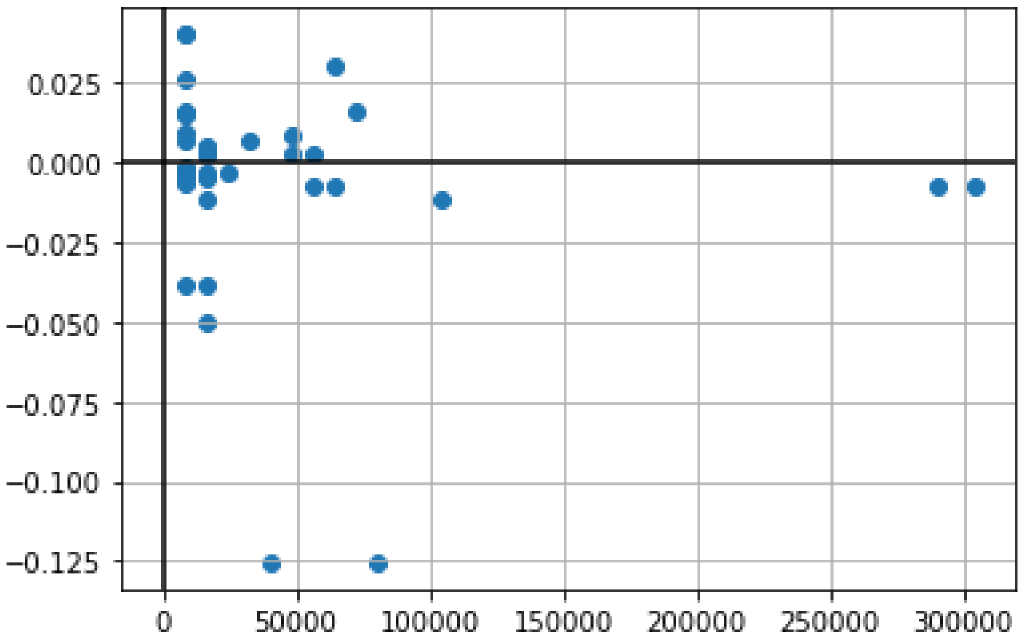
```
     transaction_date          asset_type      type            senator  \
0          2016-01-04               Stock      Sale    Patrick J Toomey
1          2016-01-05  Municipal Security  Purchase     Lamar Alexander
2          2016-01-05               Stock  Purchase    Patrick J Toomey
3          2016-01-05               Stock  Purchase  Sheldon Whitehouse
4          2016-01-05               Stock  Purchase     William Cassidy
…                 …                   …       …                 …
2550       2020-10-13               Stock      Sale         Pat Roberts
2551       2020-10-16               Stock  Purchase         Ron L Wyden
2552       2020-10-27    Other Securities      Sale       Mark R Warner
2553       2020-10-30    Other Securities  Purchase     Thomas R Carper
```

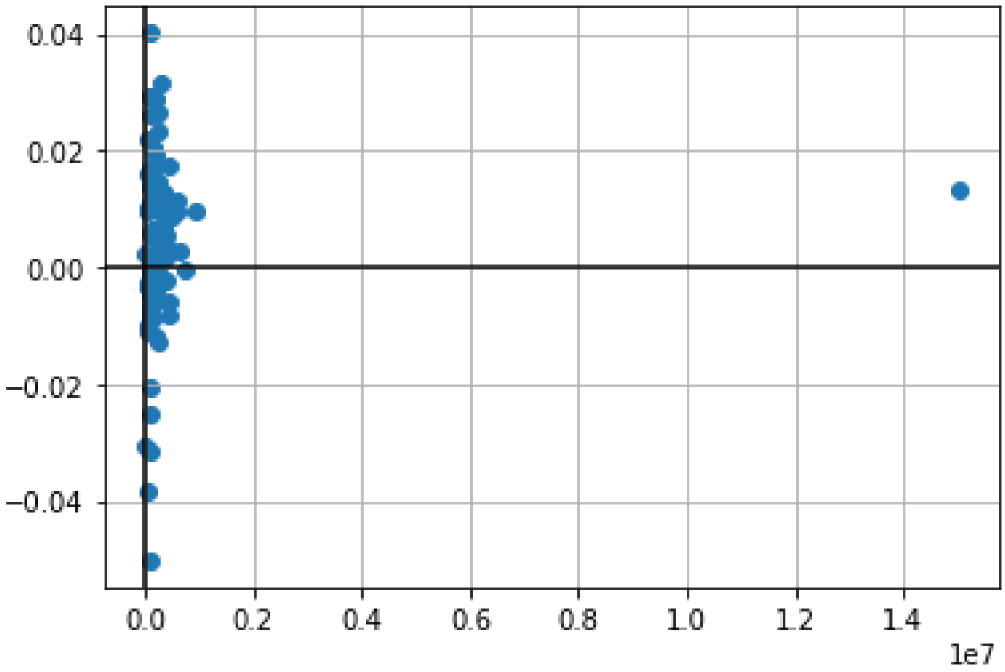
```
2554       2020-10-30    Other Securities      Sale     Thomas R Carper

      lower_amt    mid_amt  upper_amt  5_day_spy_pct_change
0          1001     8000.5      15000             -0.044324
1         15001    32500.5      50000             -0.038240
2          3003    24001.5      45000             -0.038240
3         12012    96006.0     180000             -0.038240
4          2002    16001.0      30000             -0.038240
…         …        …        …                   …
2550       2002    16001.0      30000             -0.019279
2551       6006    48003.0      90000             -0.004348
2552    1000001  3000000.5    5000000             -0.006475
2553       2002    16001.0      30000              0.072334
2554       1001     8000.5      15000              0.072334

[2555 rows x 8 columns]
```

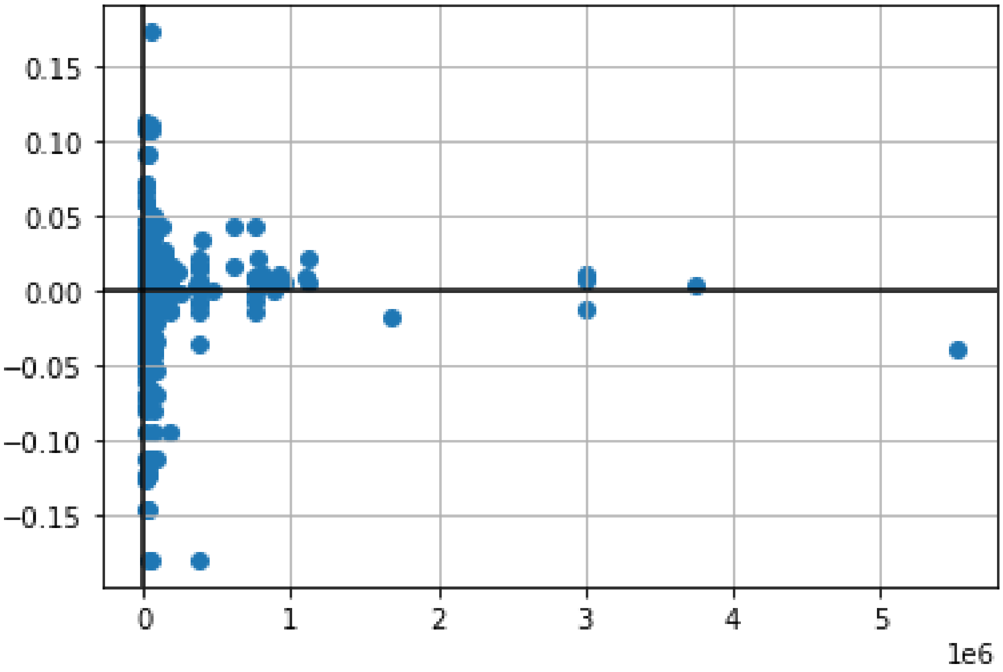


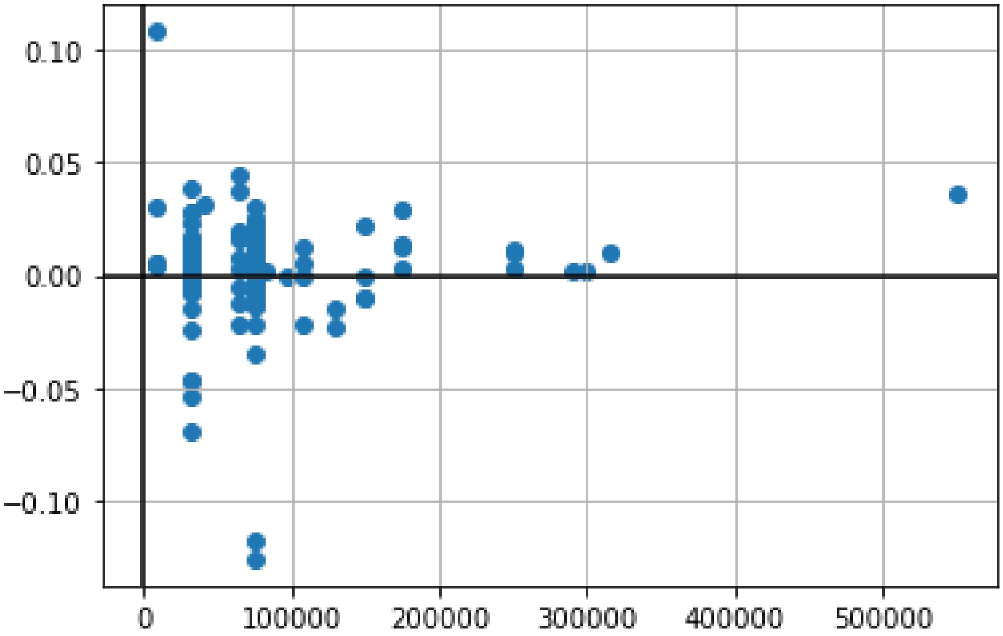

0.04
0.02
0.00
−0.02
−0.04
0
200000
400000
600000
800000

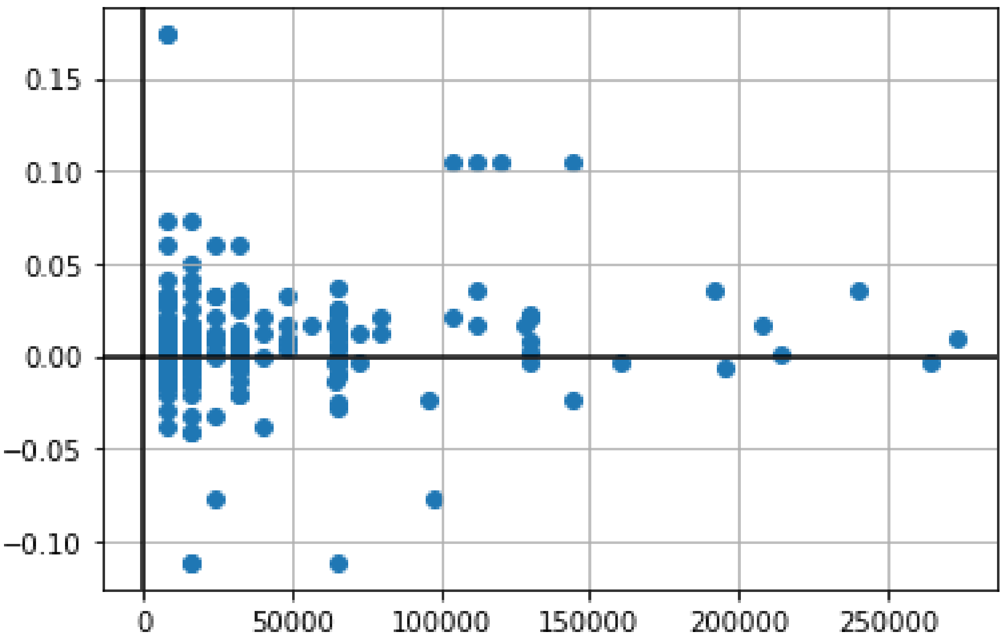

0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
0
20000
40000
60000
80000
100000
120000
140000
160000

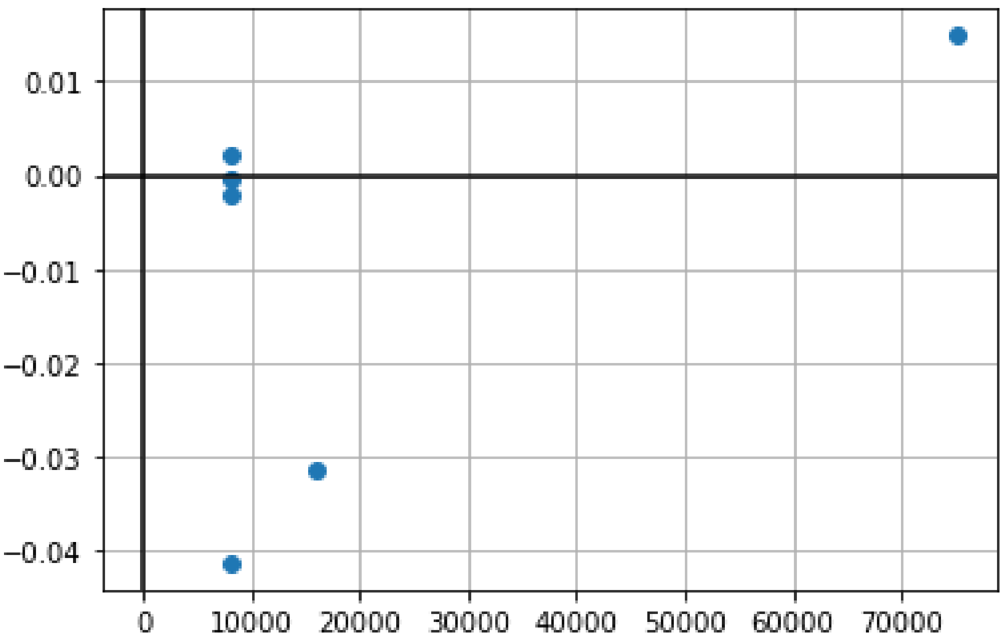

0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
0
50000
100000
150000
200000
250000
300000

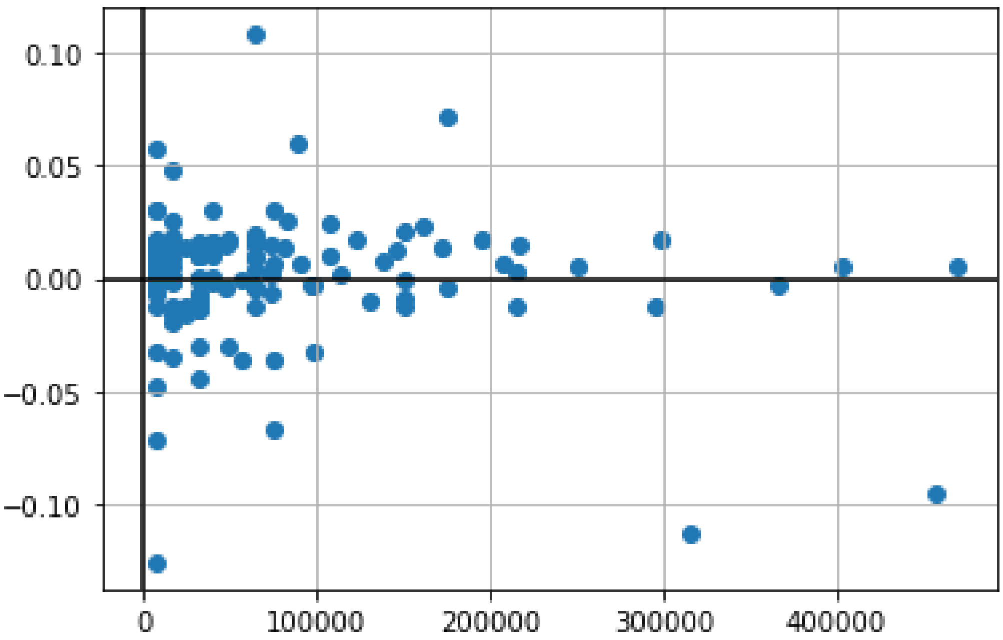

0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e7

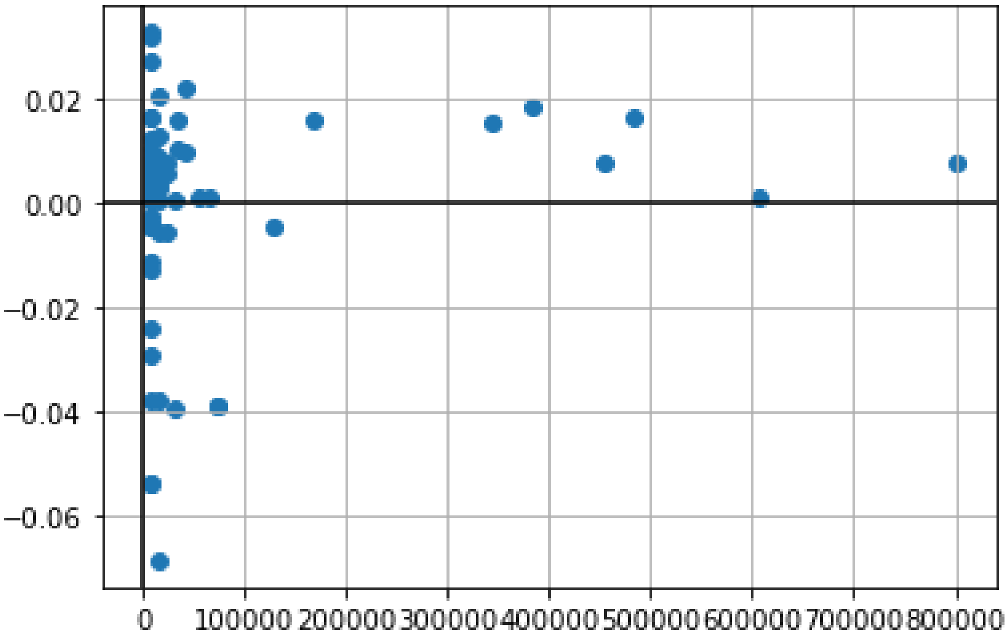

0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
0
1
2
3
4
5
1e6

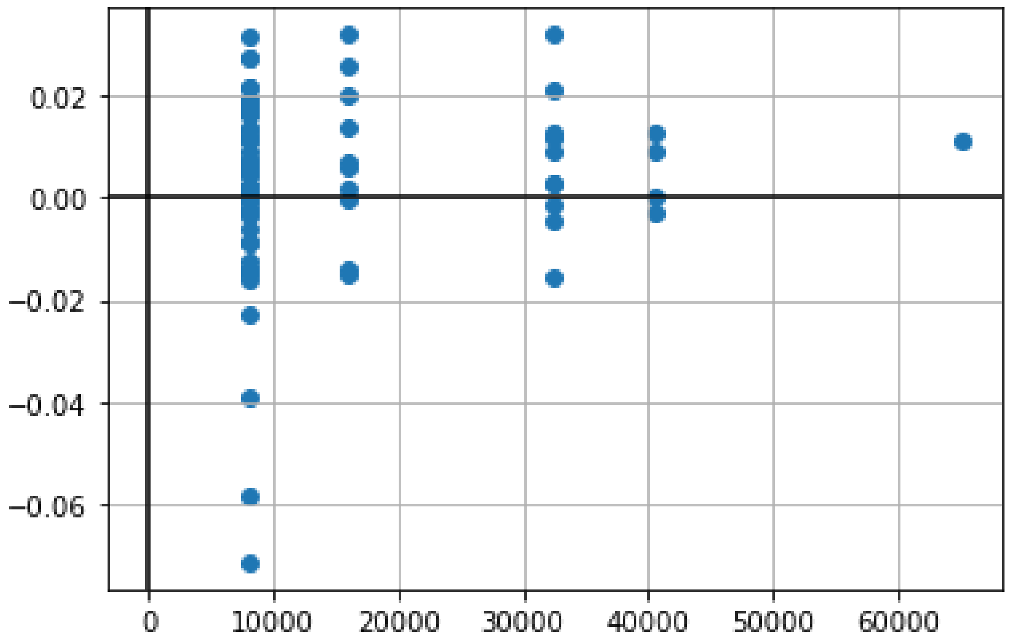

0.10
0.05
0.00
−0.05
−0.10
0
100000
200000
300000
400000
500000

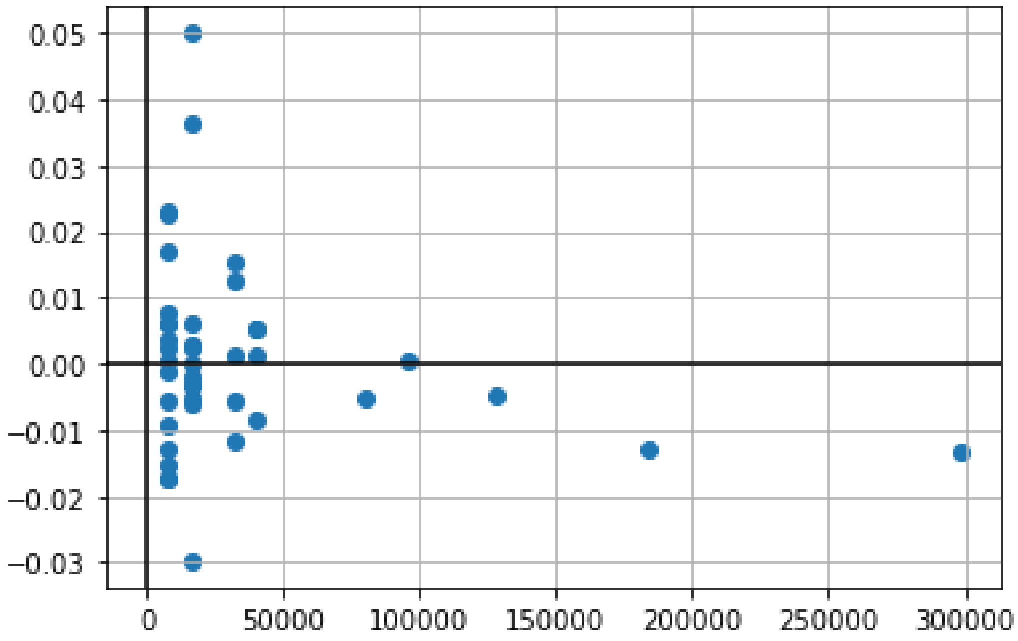

0.15
0.10
0.05
0.00
−0.05
−0.10
0
50000
100000
150000
200000
250000

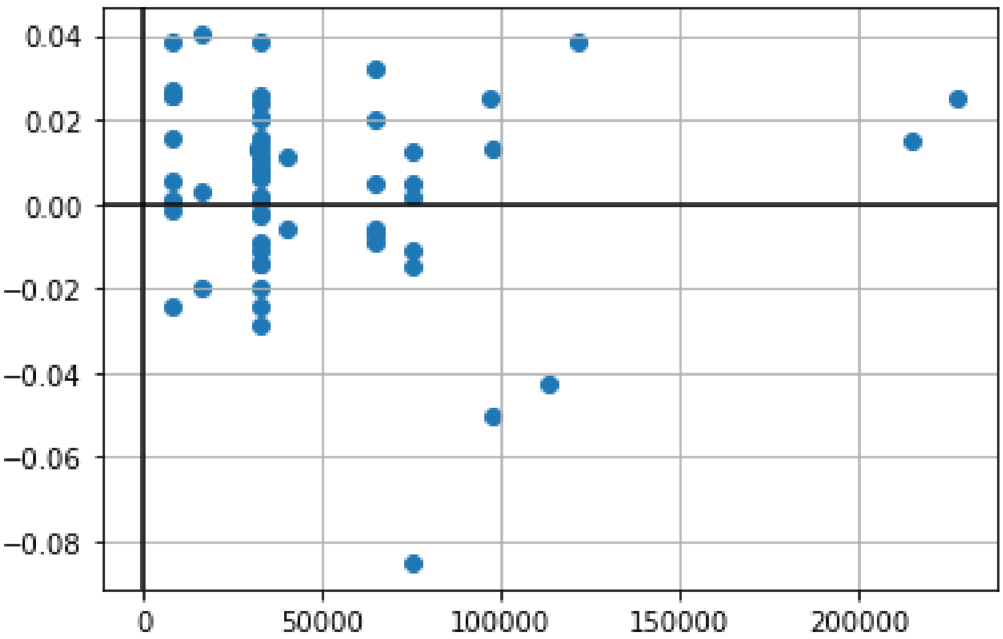

0.01
0.00
−0.01
−0.02
−0.03
−0.04
0
10000
20000
30000
40000
50000
60000
70000

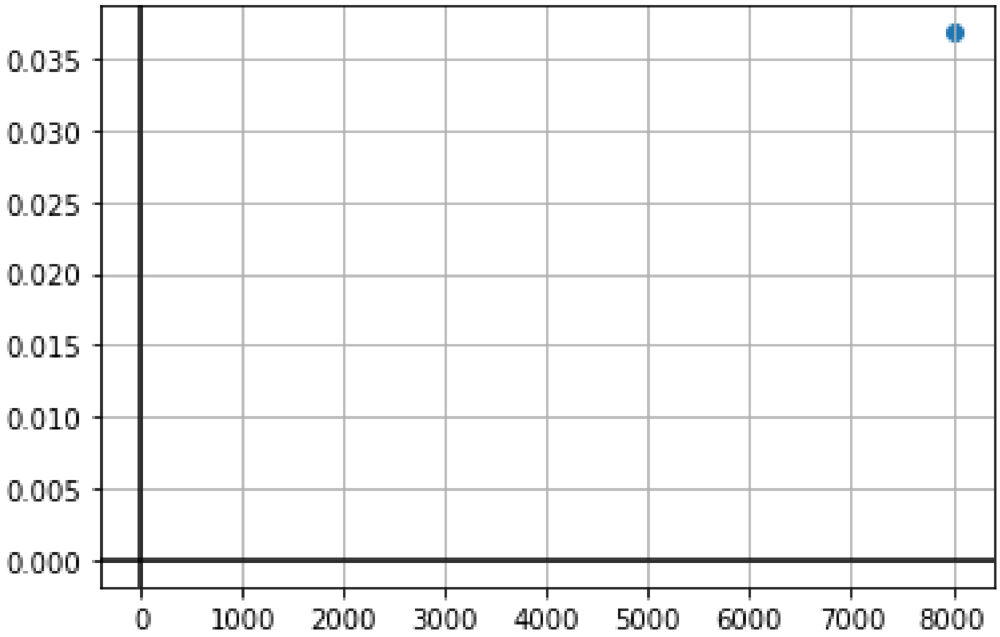

0.10
0.05
0.00
−0.05
−0.10
0
100000
200000
300000
400000

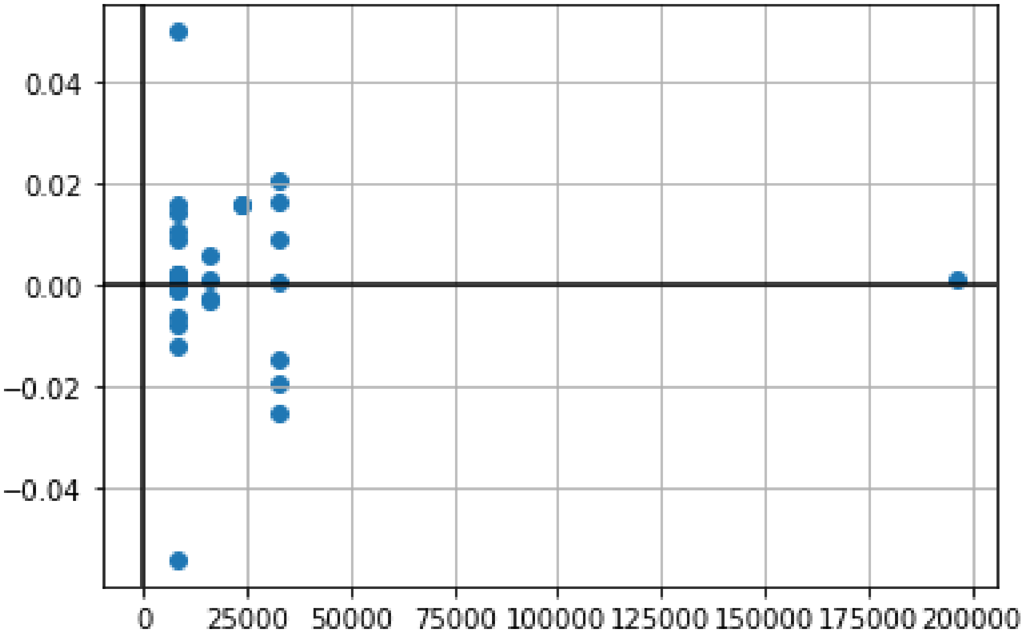

0.02
0.00
−0.02
−0.04
−0.06
0
100000
200000
300000
400000
500000
600000
700000
800000

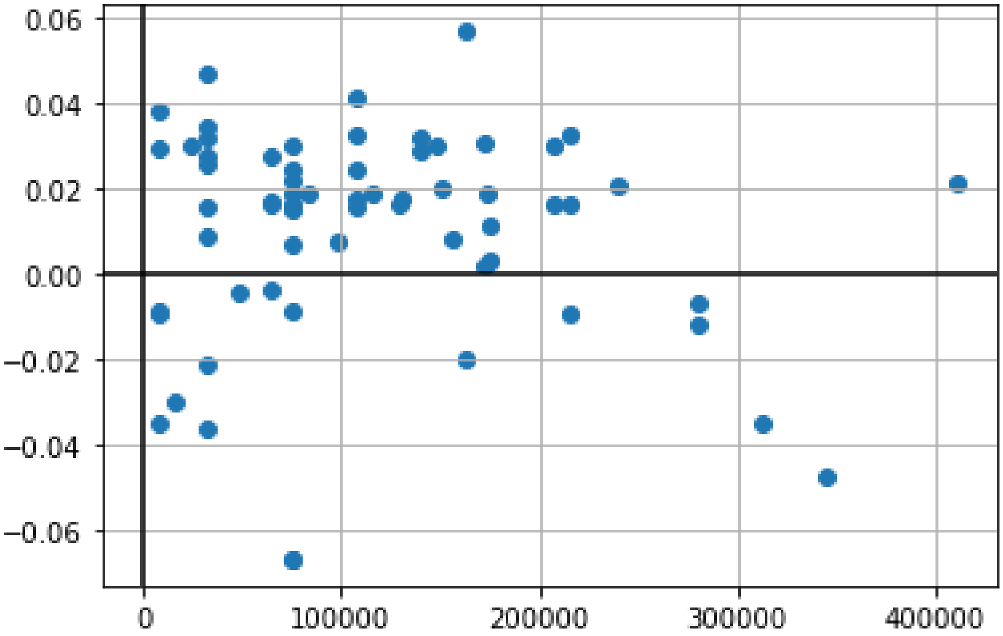

0.02
0.00
−0.02
−0.04
−0.06
0
10000
20000
30000
40000
50000
60000

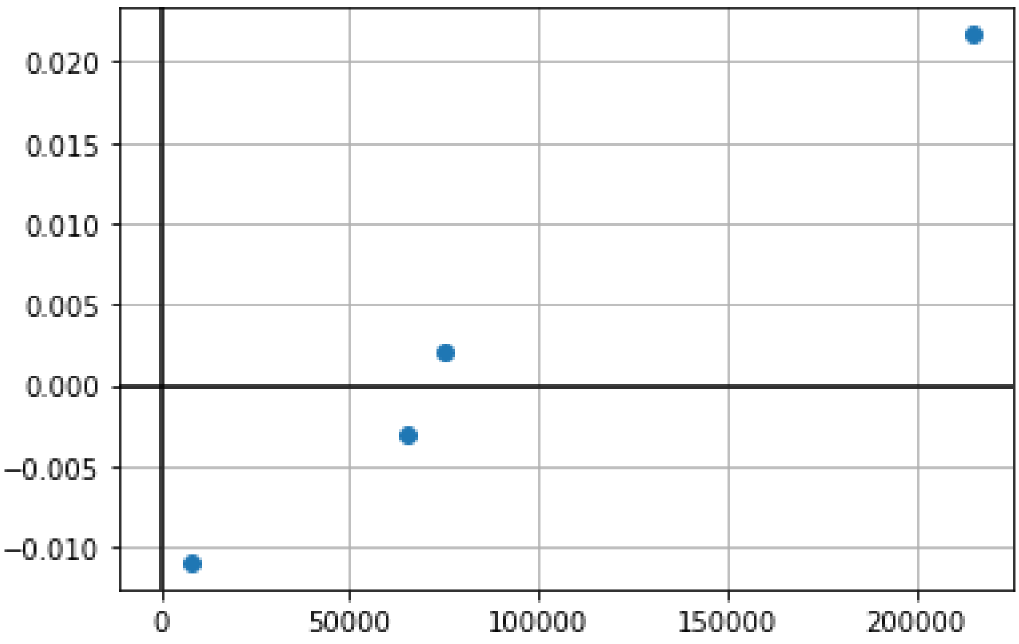

0.05
0.04
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
0
50000
100000
150000
200000
250000
300000

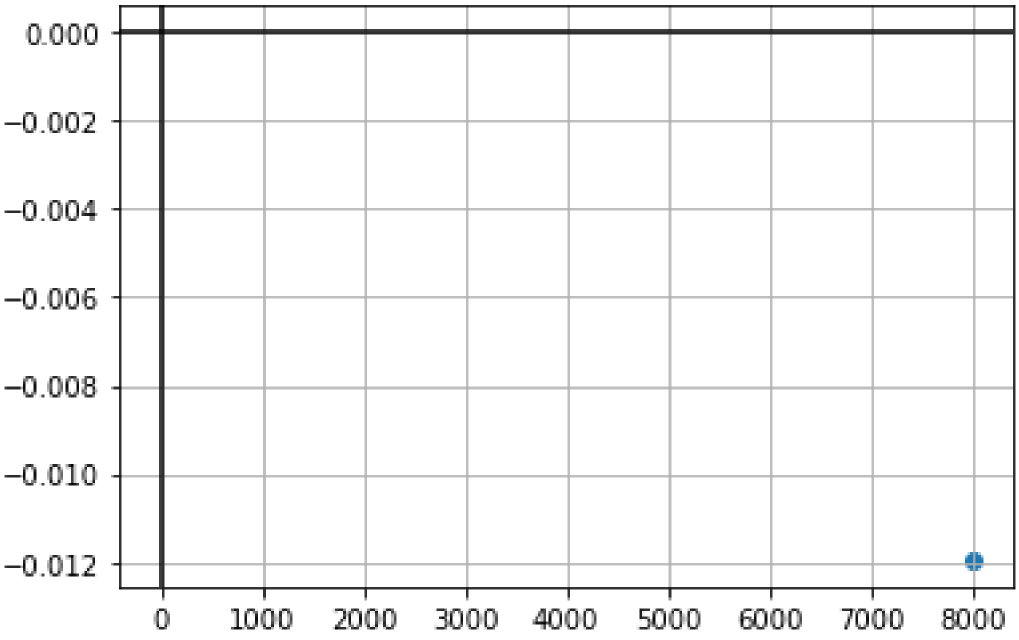

0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
0
50000
100000
150000
200000

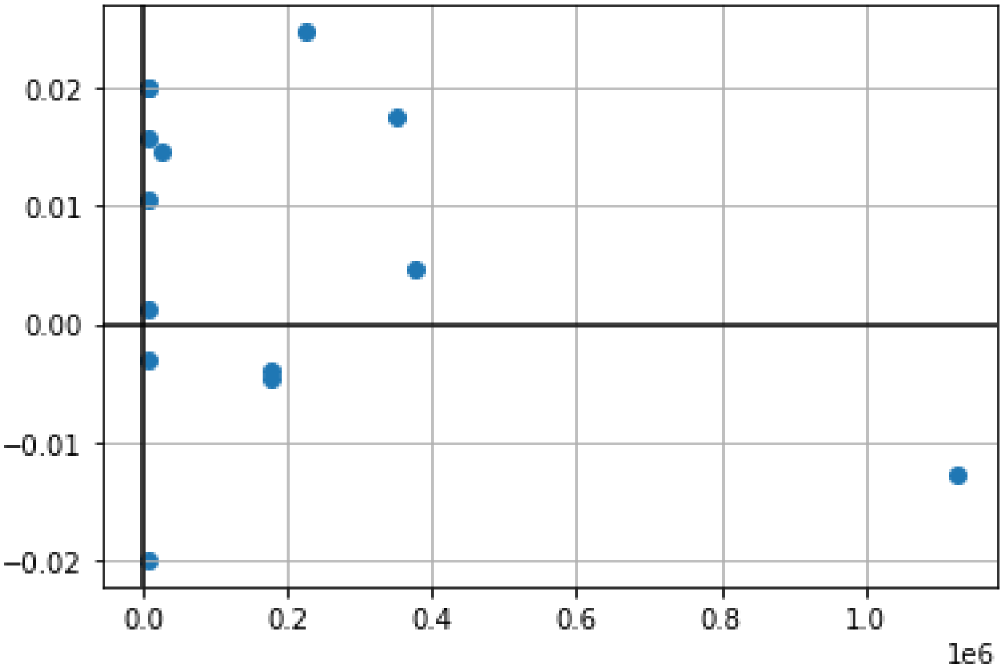

0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
1000
2000
3000
4000
5000
6000
7000
8000

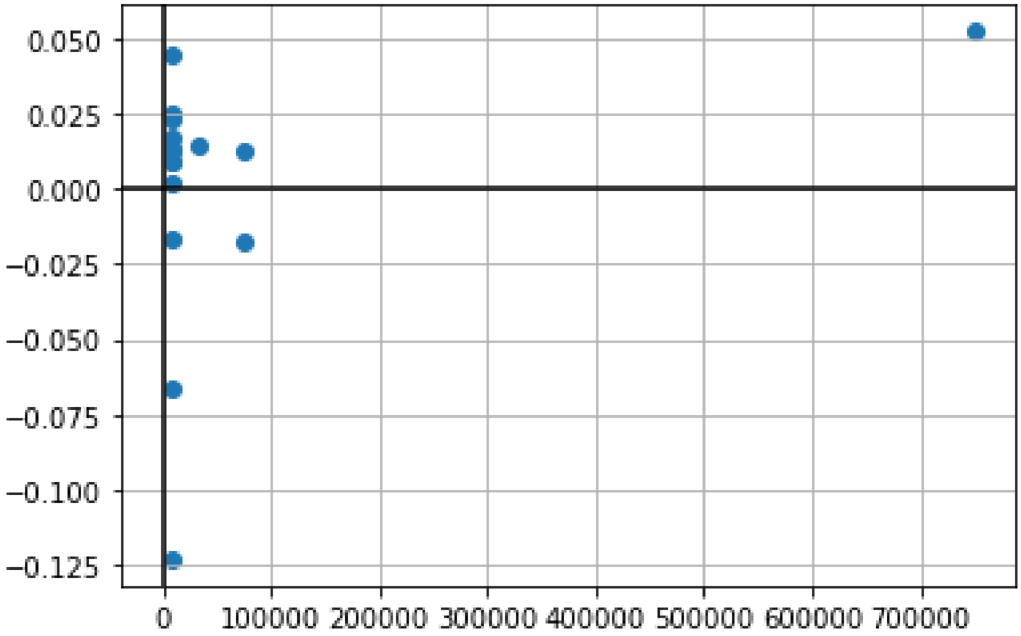

0.04
0.02
0.00
−0.02
−0.04
0
25000
50000
75000
100000
125000
150000
175000
200000

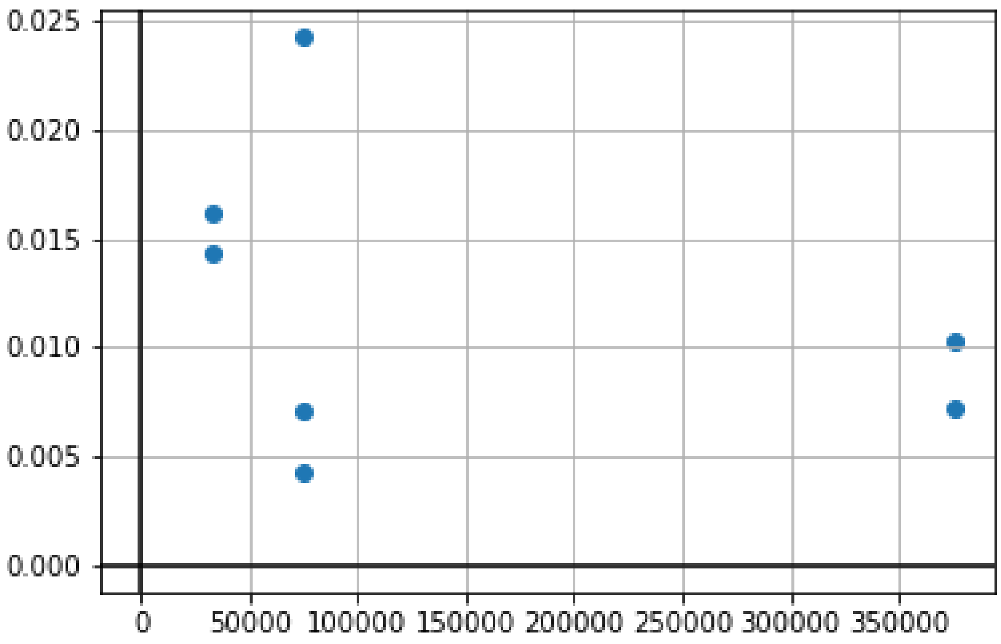

0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
100000
200000
300000
400000

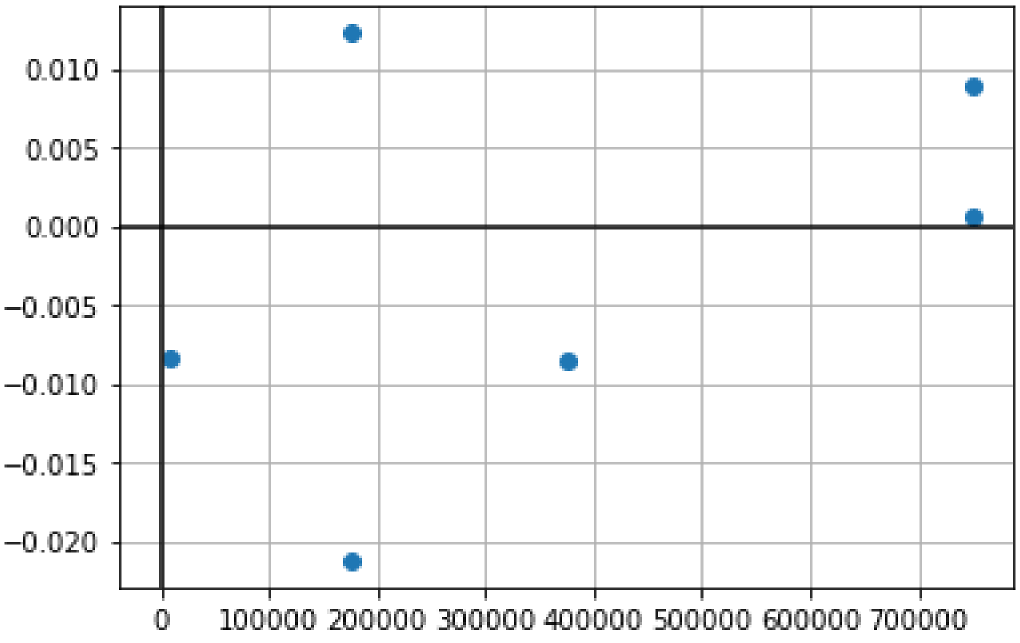

0.020
0.015
0.010
0.005
0.000
−0.005
−0.010
0
50000
100000
150000
200000

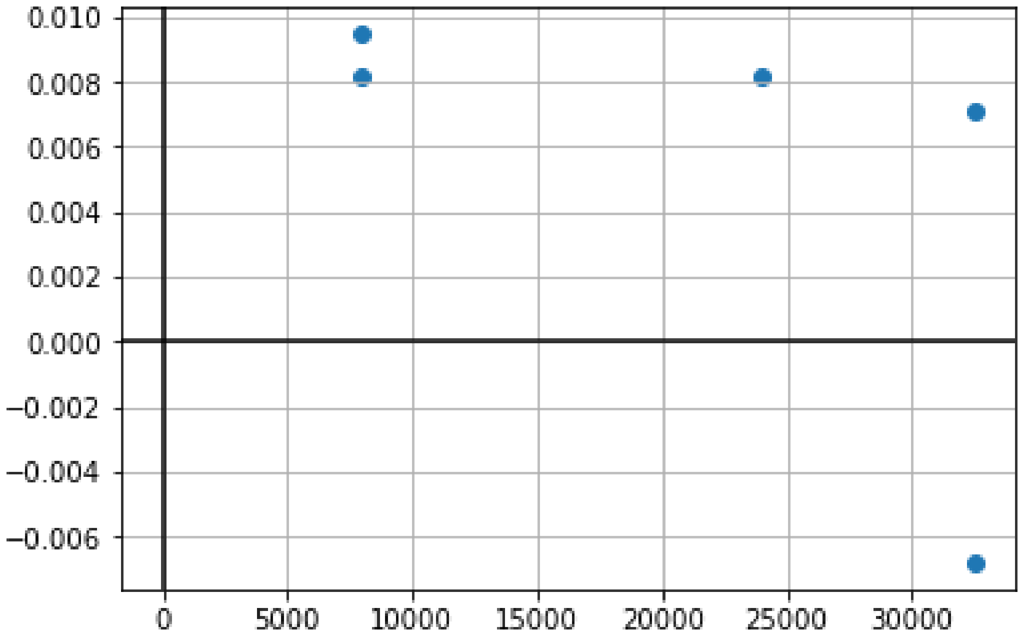

0.000
−0.002
−0.004
−0.006
−0.008
−0.010
−0.012
0
1000
2000
3000
4000
5000
6000
7000
8000

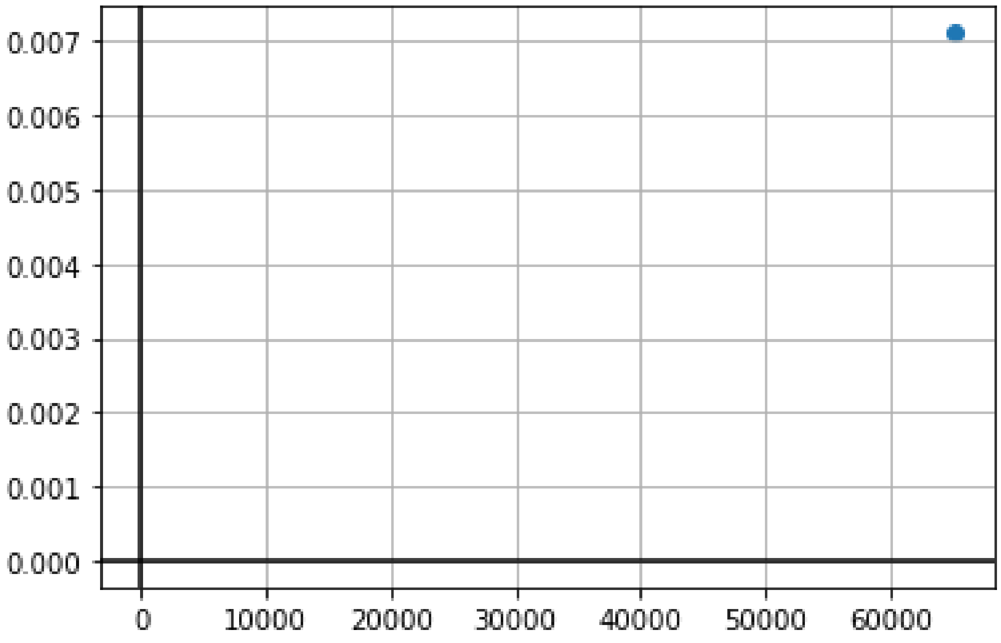

0.02
0.01
0.00
−0.01
−0.02
0.0
0.2
0.4
0.6
0.8
1.0
1e6

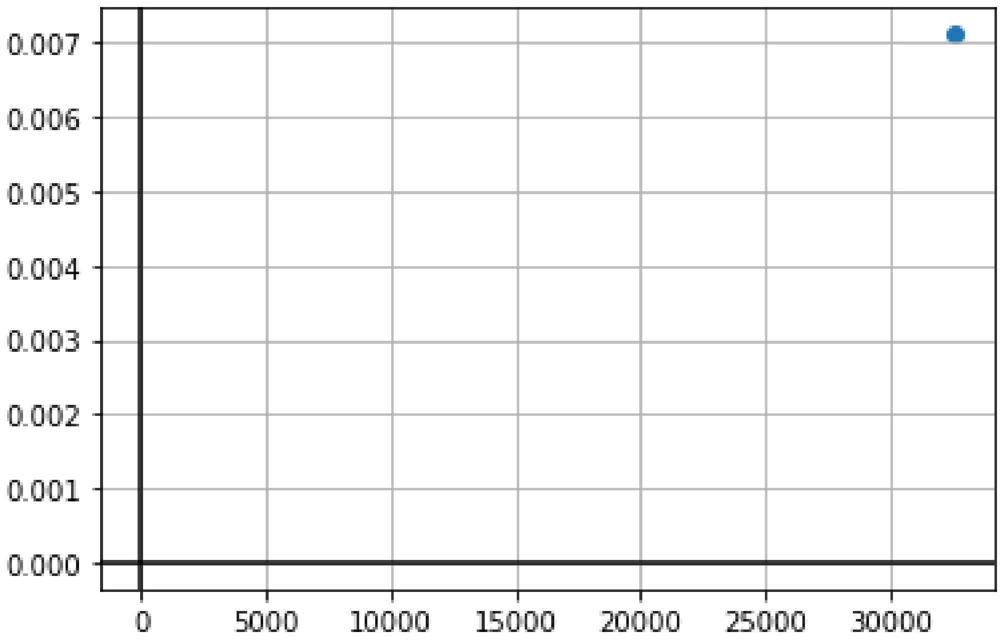

0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
0
100000
200000
300000
400000
500000
600000
700000

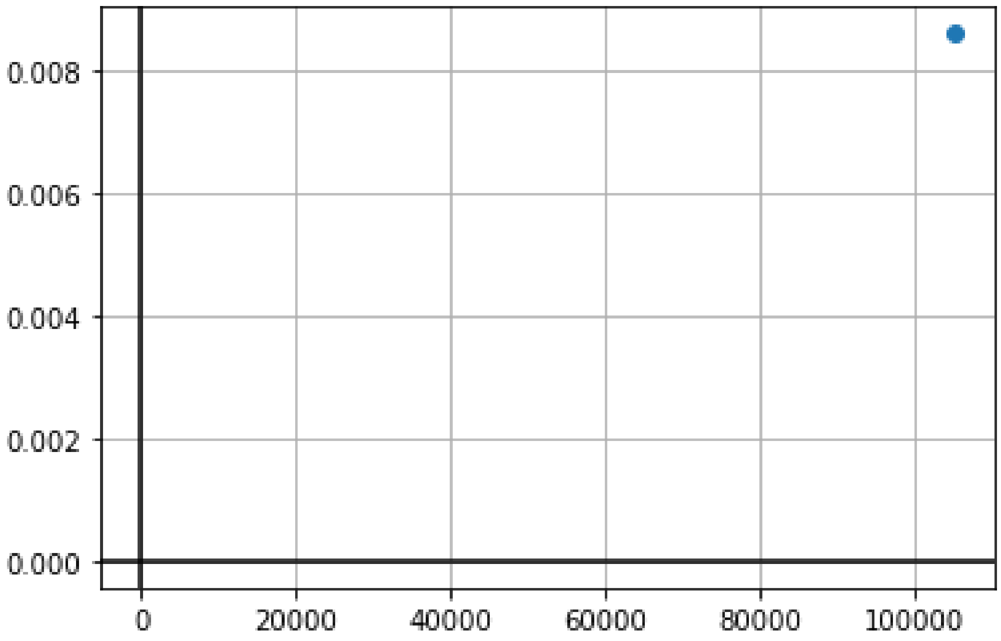

0.025
0.020
0.015
0.010
0.005
0.000
0
50000
100000
150000
200000
250000
300000
350000

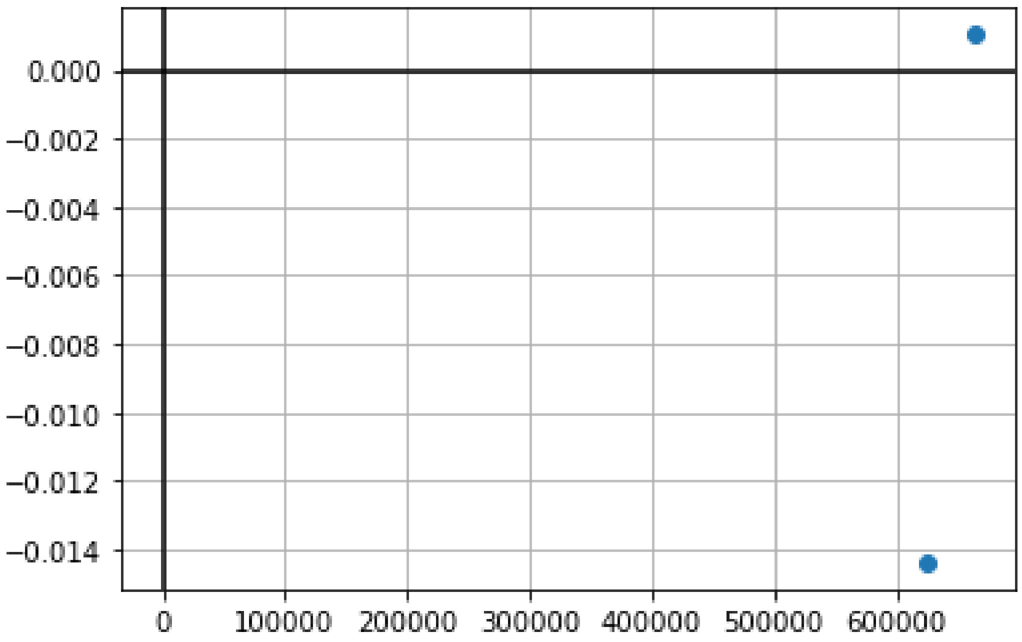

0.010
0.005
0.000
−0.005
−0.010
−0.015
−0.020
0
100000
200000
300000
400000
500000
600000
700000

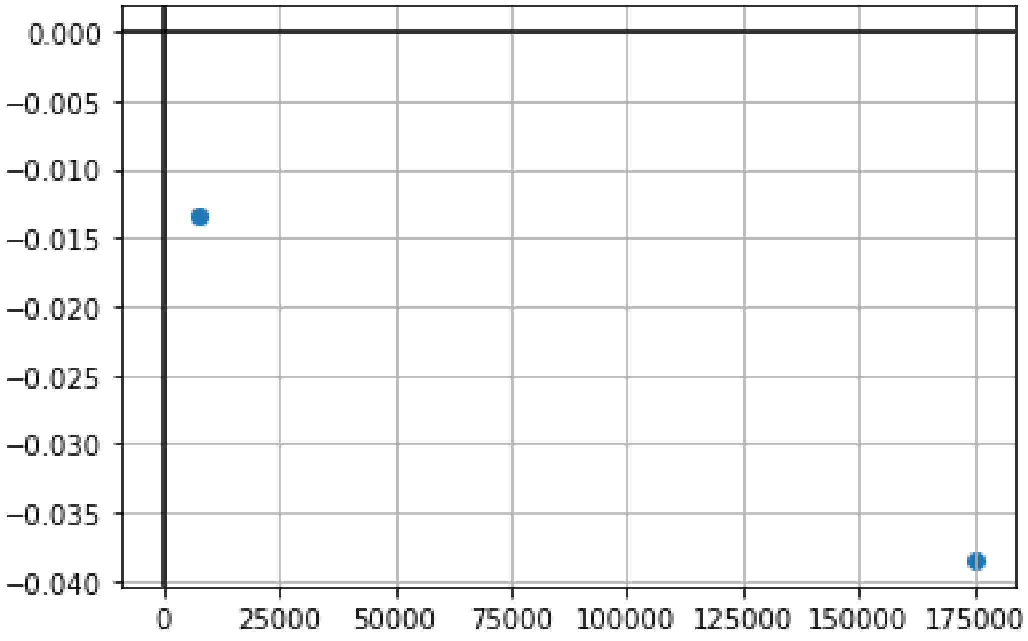

0.010
0.008
0.006
0.004
0.002
0.000
−0.002
−0.004
−0.006
0
5000
10000
15000
20000
25000
30000

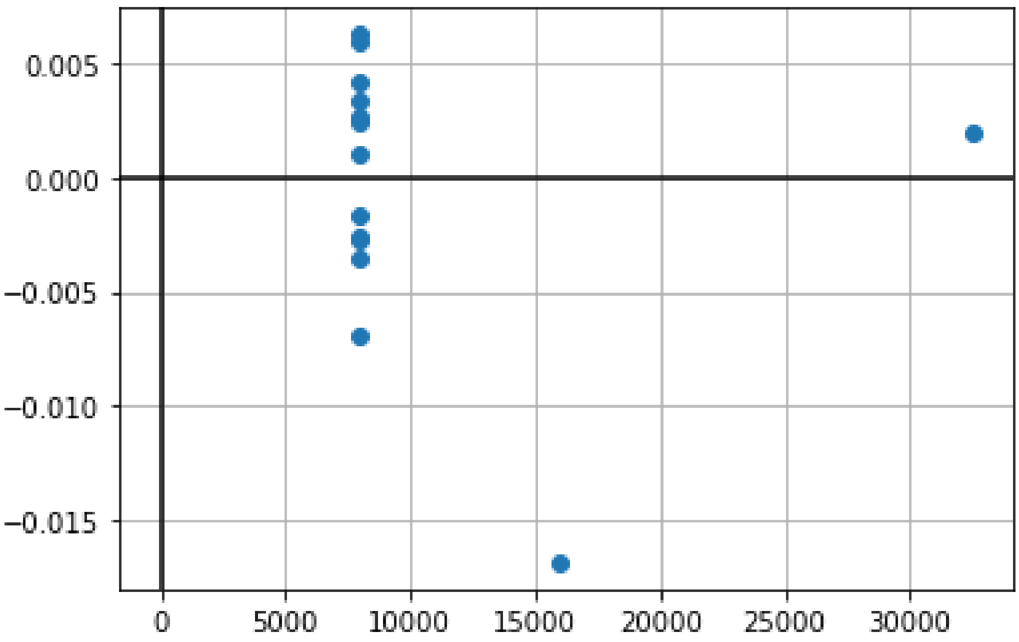

0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
10000
20000
30000
40000
50000
60000

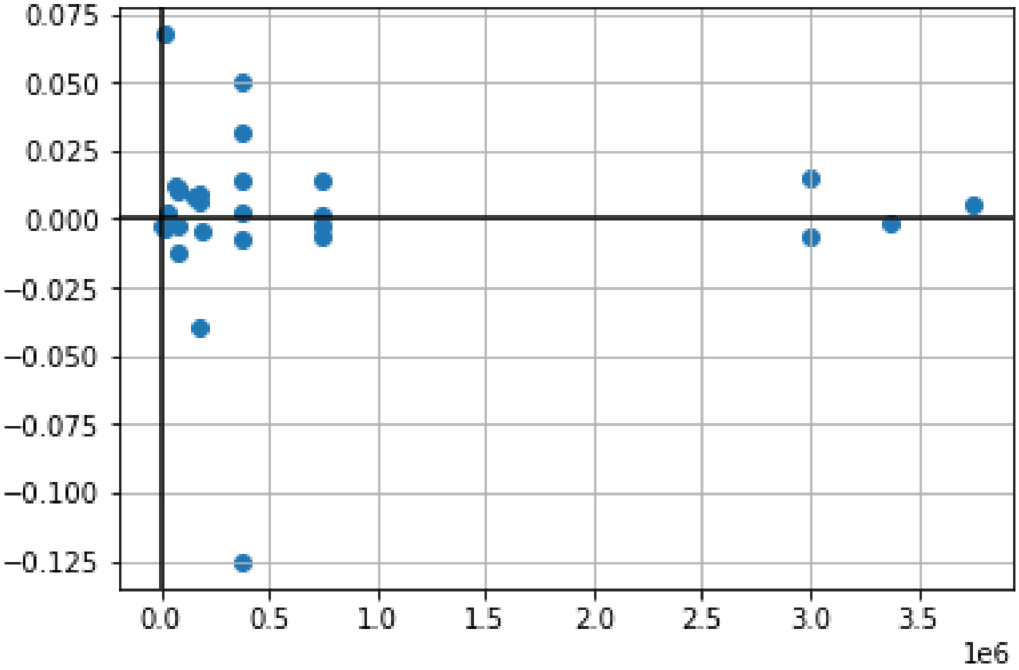

0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
5000
10000
15000
20000
25000
30000

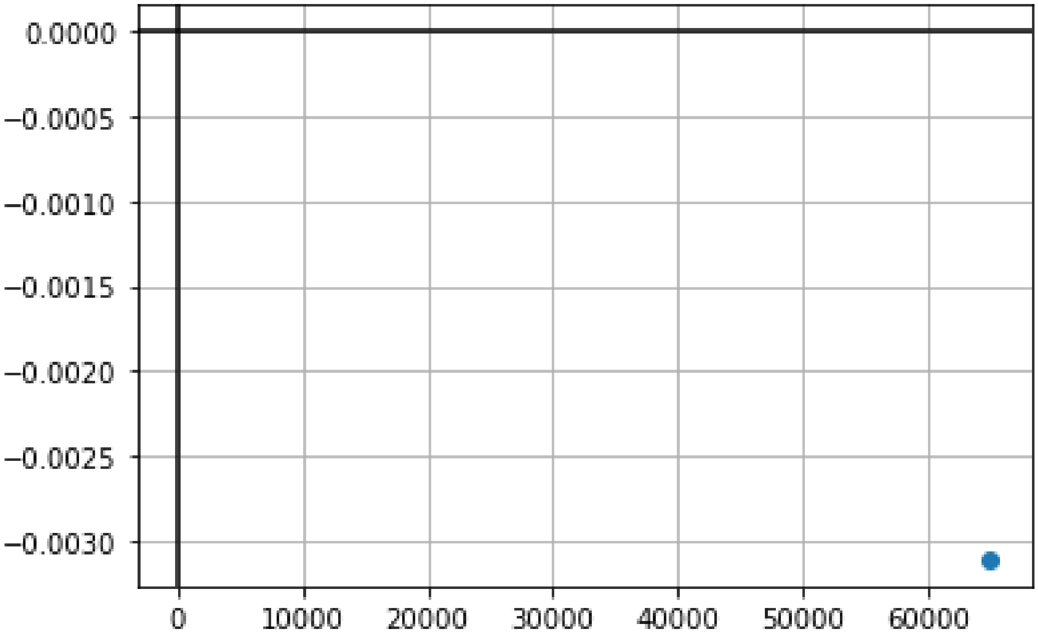

0.008
0.006
0.004
0.002
0.000
0
20000
40000
60000
80000
100000

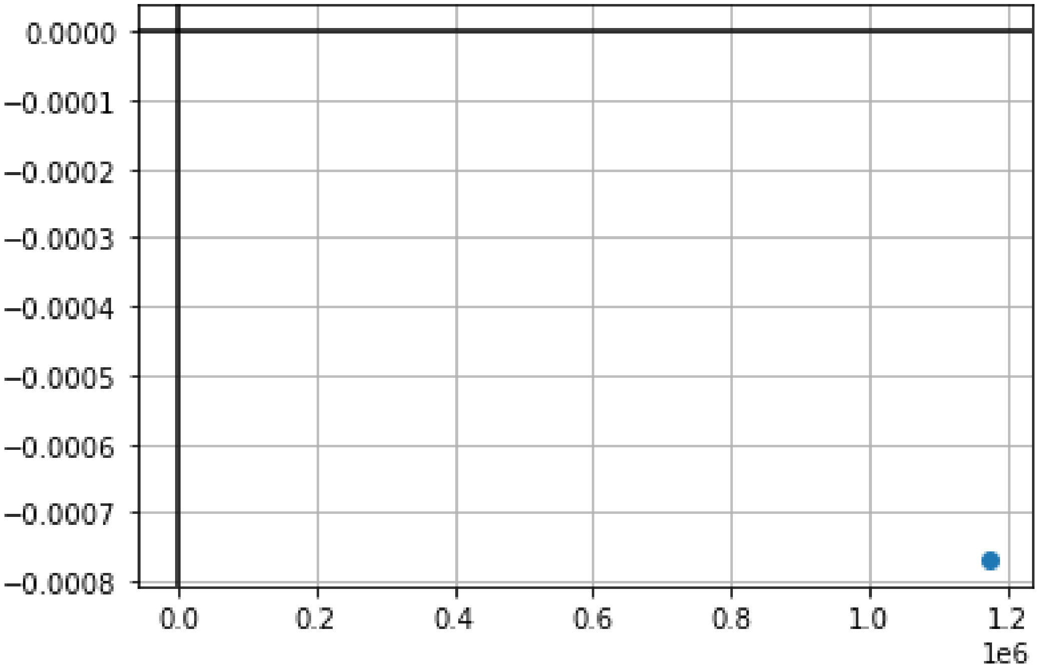

0.000
−0.002
−0.004
−0.006
−0.008
−0.010
−0.012
−0.014
0
100000
200000
300000
400000
500000
600000

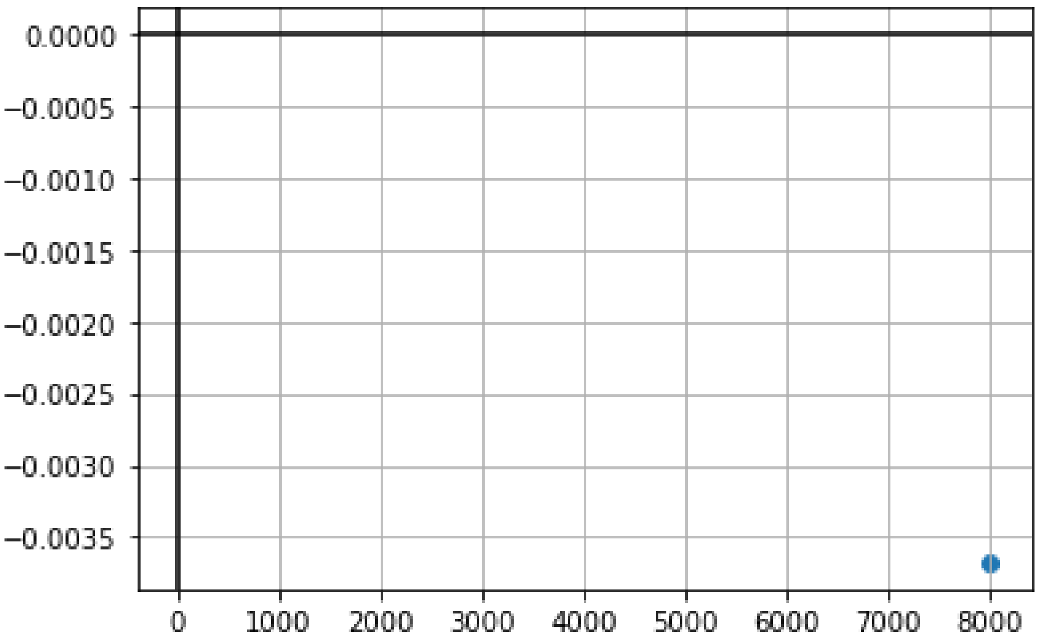

0.000
−0.005
−0.010
−0.015
−0.020
−0.025
−0.030
−0.035
−0.040
0
25000
50000
75000
100000
125000
150000
175000

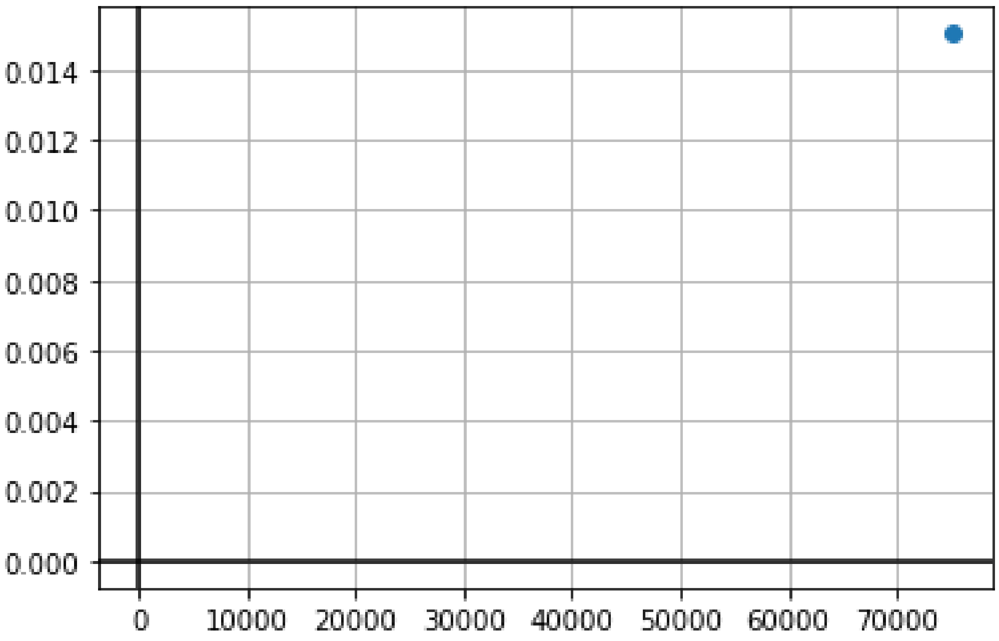

0.005
0.000
−0.005
−0.010
−0.015
0
5000
10000
15000
20000
25000
30000

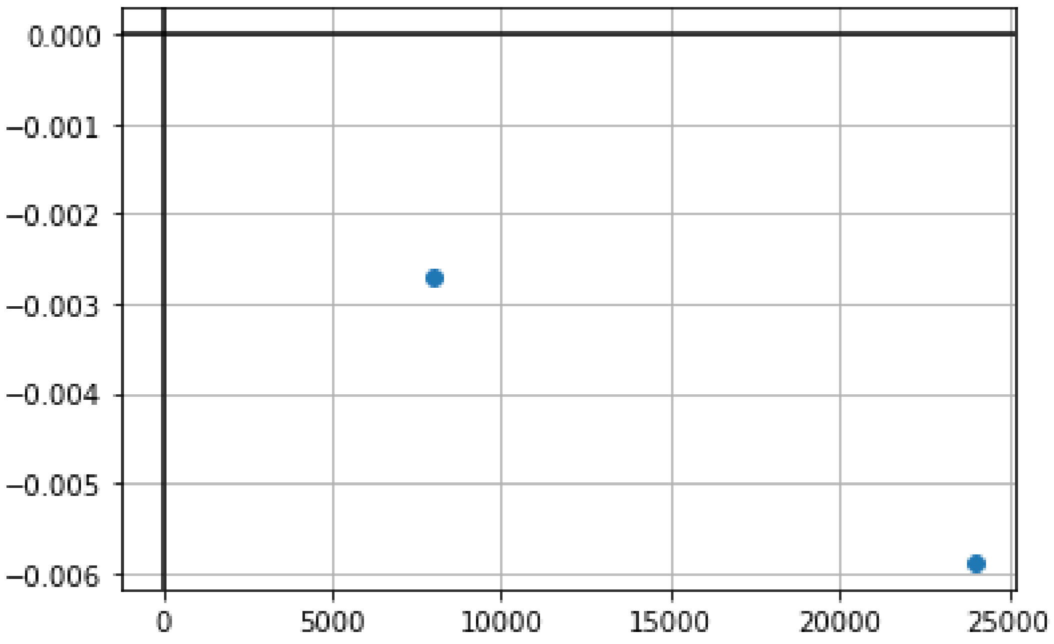

0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1e6

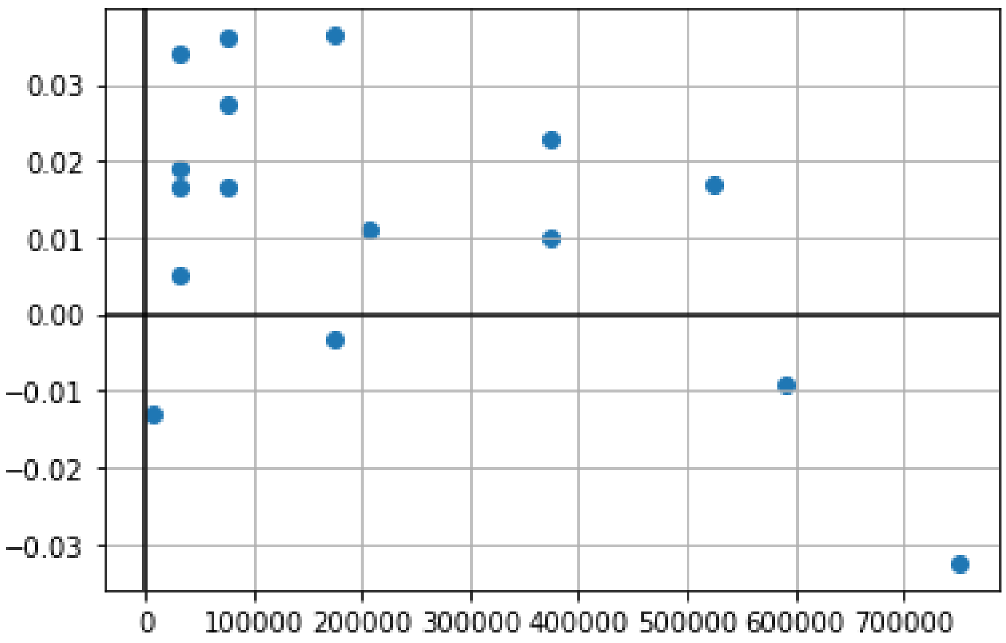

0.0000
−0.0005
−0.0010
−0.0015
−0.0020
−0.0025
−0.0030
0
10000
20000
30000
40000
50000
60000

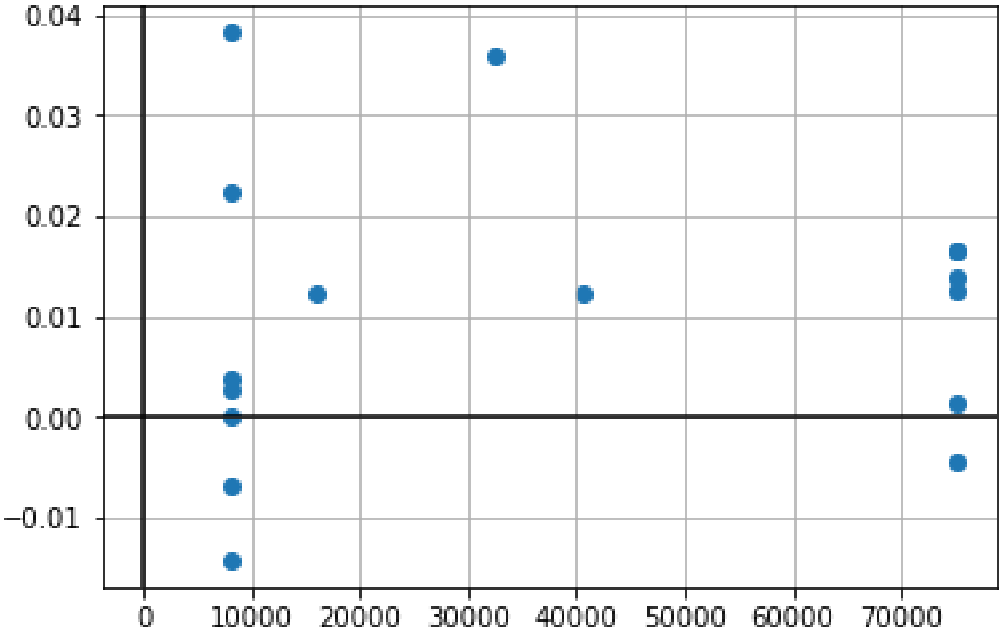

0.0000
−0.0001
−0.0002
−0.0003
−0.0004
−0.0005
−0.0006
−0.0007
−0.0008
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6

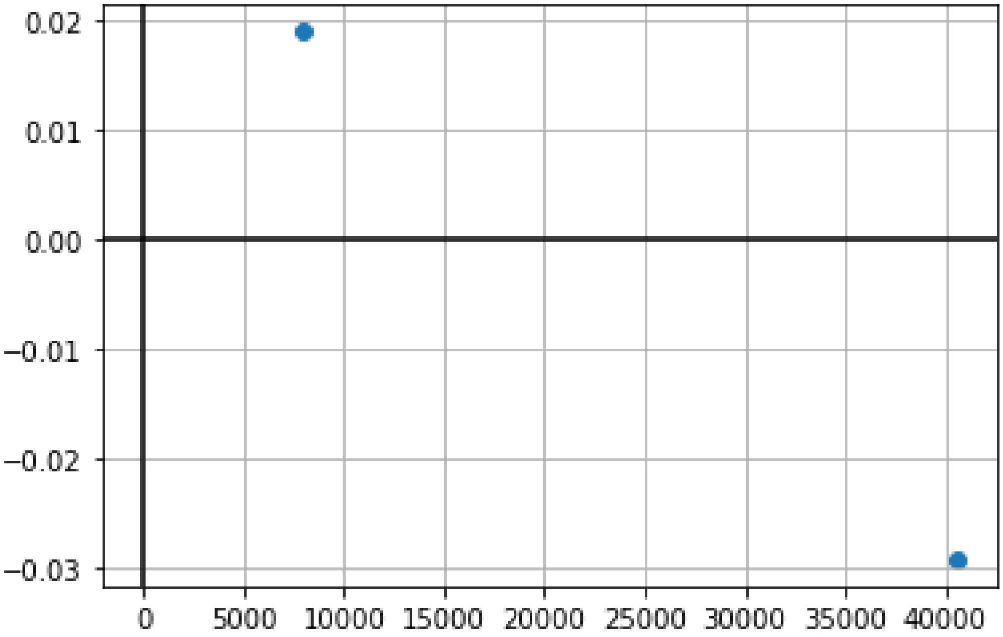

0.0000
−0.0005
−0.0010
−0.0015
−0.0020
−0.0025
−0.0030
−0.0035
0
1000
2000
3000
4000
5000
6000
7000
8000

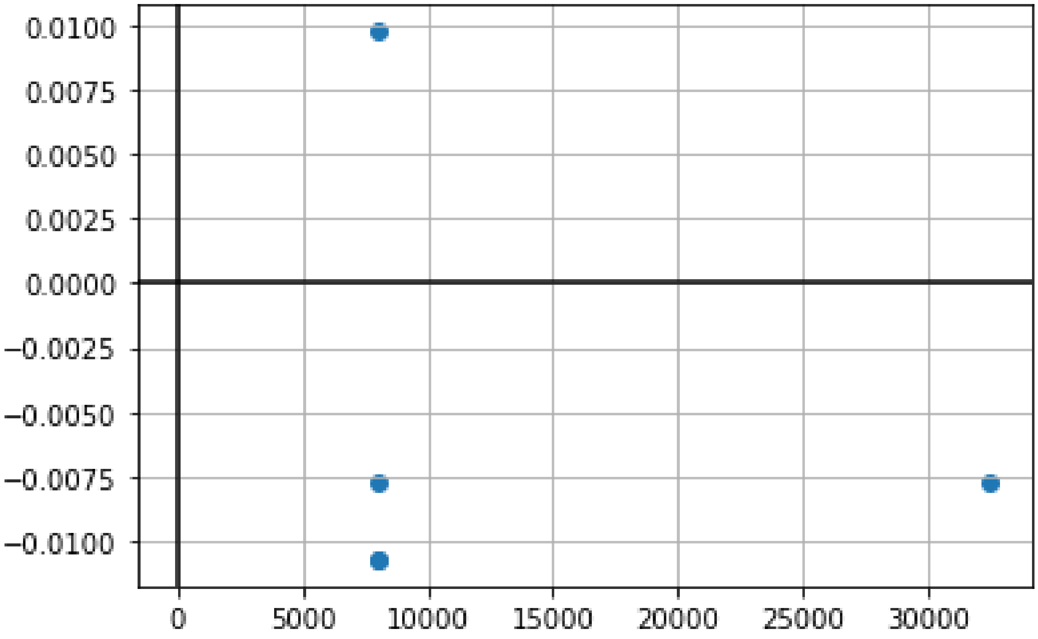

0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
10000
20000
30000
40000
50000
60000
70000

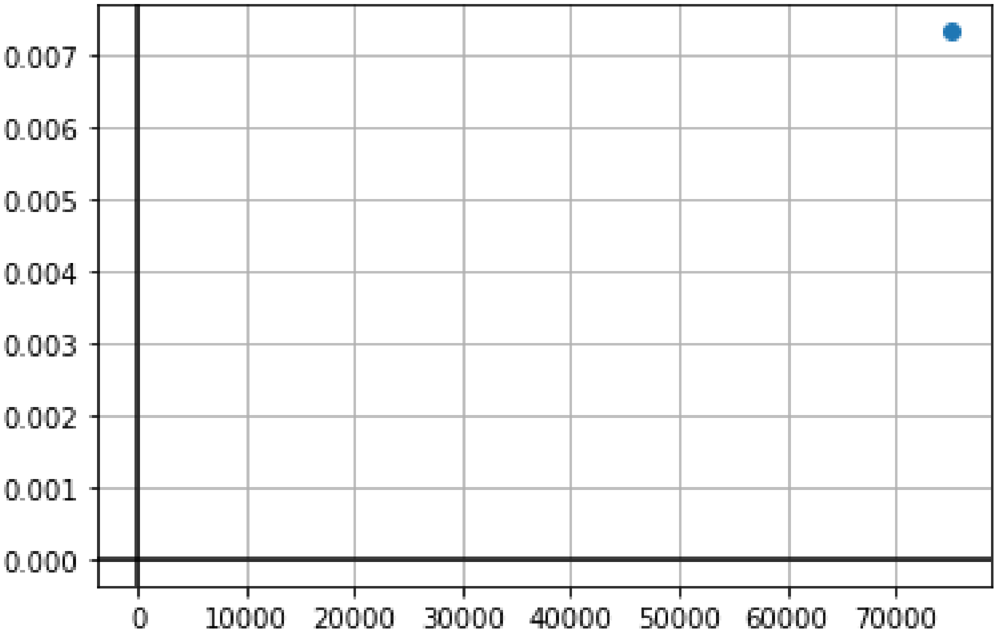

0.000
−0.001
−0.002
−0.003
−0.004
−0.005
−0.006
0
5000
10000
15000
20000
25000

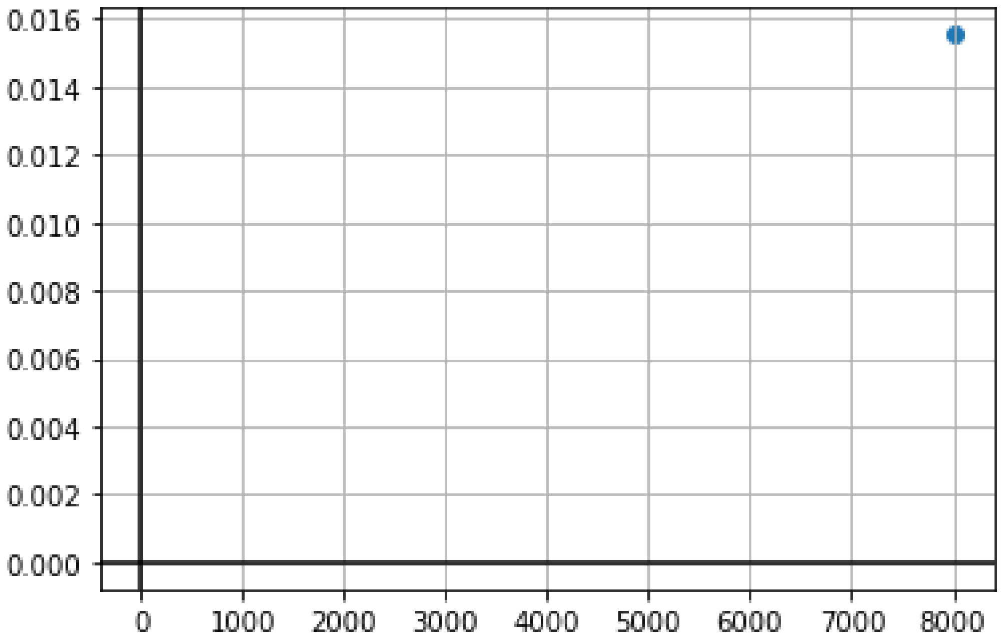

0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
0
100000
200000
300000
400000
500000
600000
700000

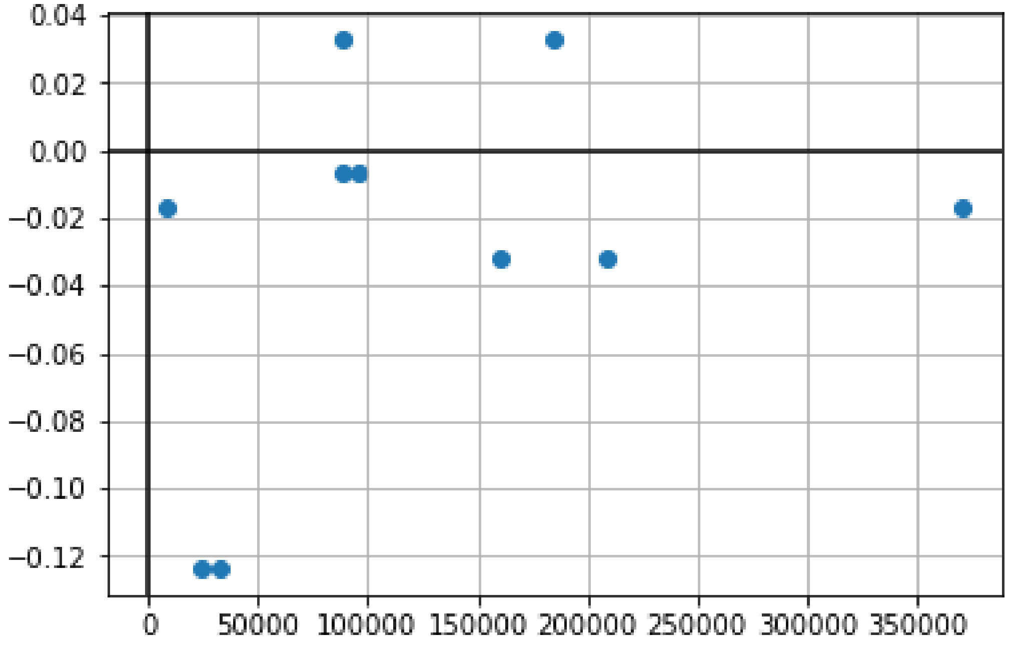

0.04
0.03
0.02
0.01
0.00
−0.01
0
10000
20000
30000
40000
50000
60000
70000

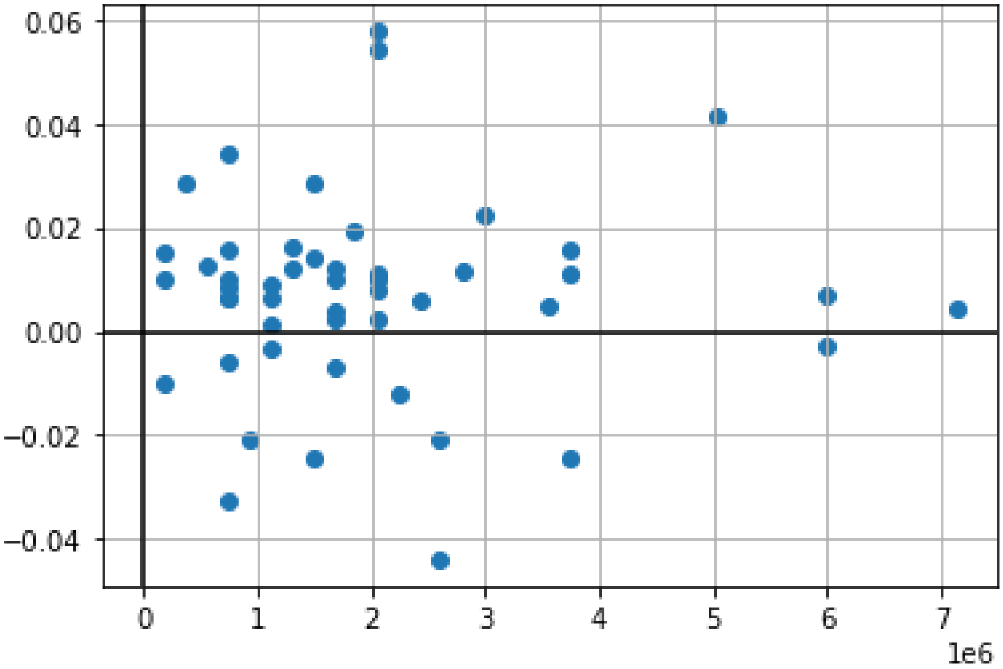

0.02
0.01
0.00
−0.01
−0.02
−0.03
0
5000
10000
15000
20000
25000
30000
35000
40000

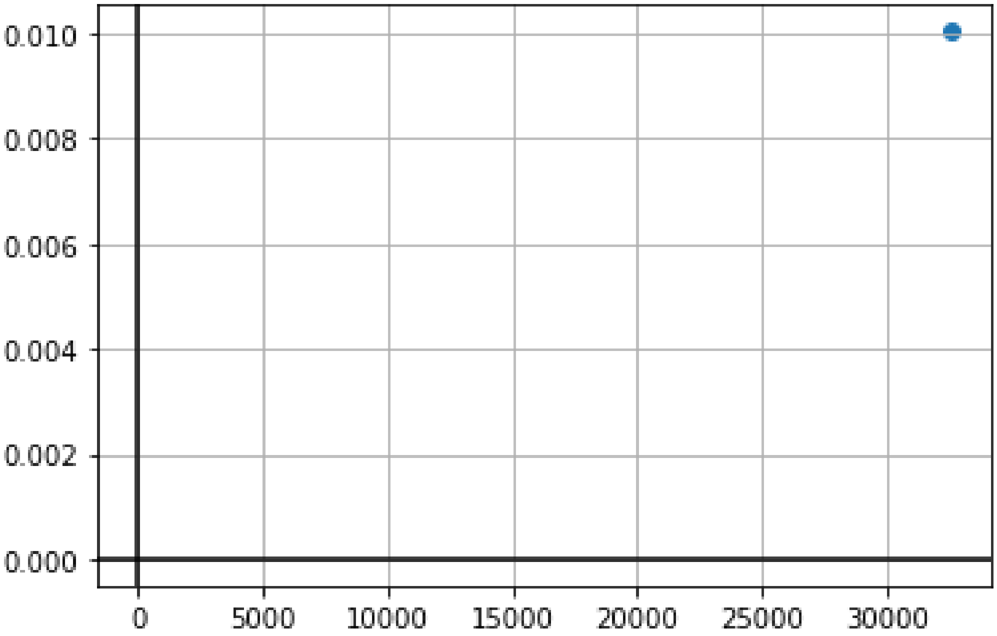

0.0100
0.0075
0.0050
0.0025
0.0000
−0.0025
−0.0050
−0.0075
−0.0100
0
5000
10000
15000
20000
25000
30000

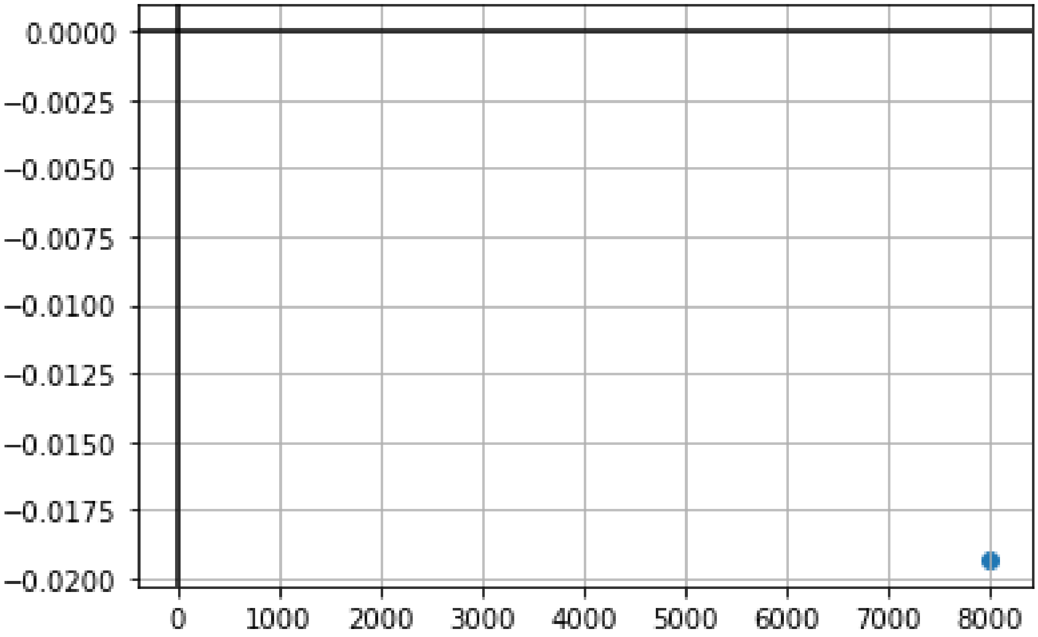

0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
10000
20000
30000
40000
50000
60000
70000

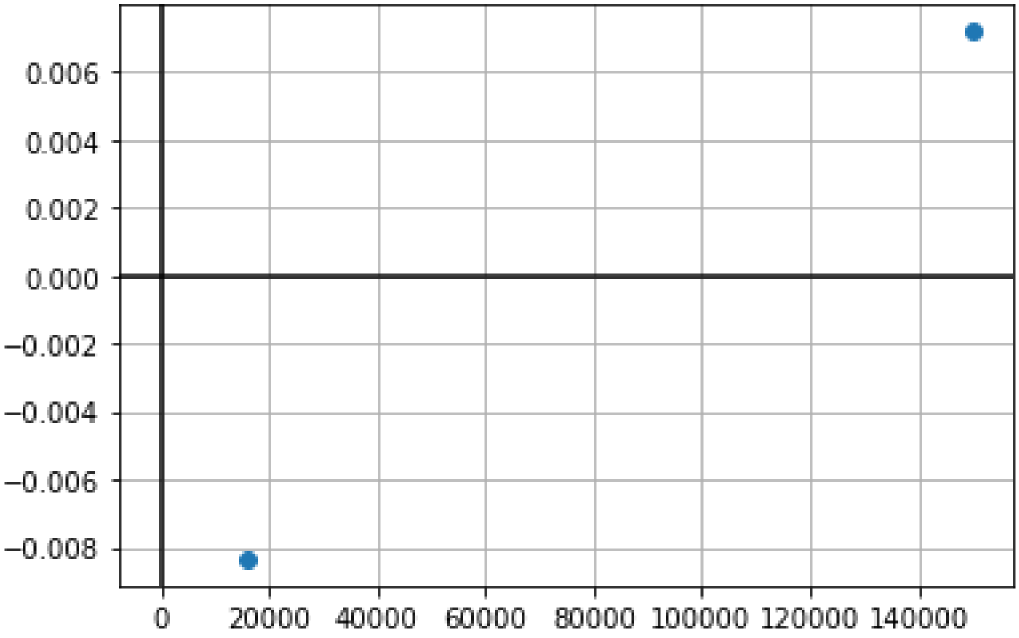

0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
1000
2000
3000
4000
5000
6000
7000
8000

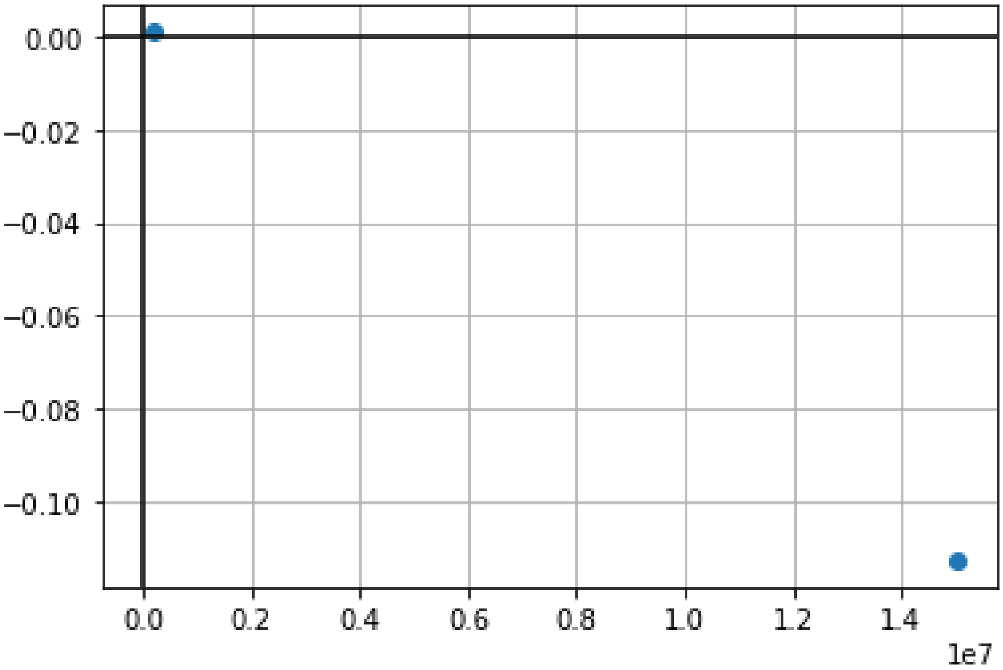

0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0
50000
100000
150000
200000
250000
300000
350000

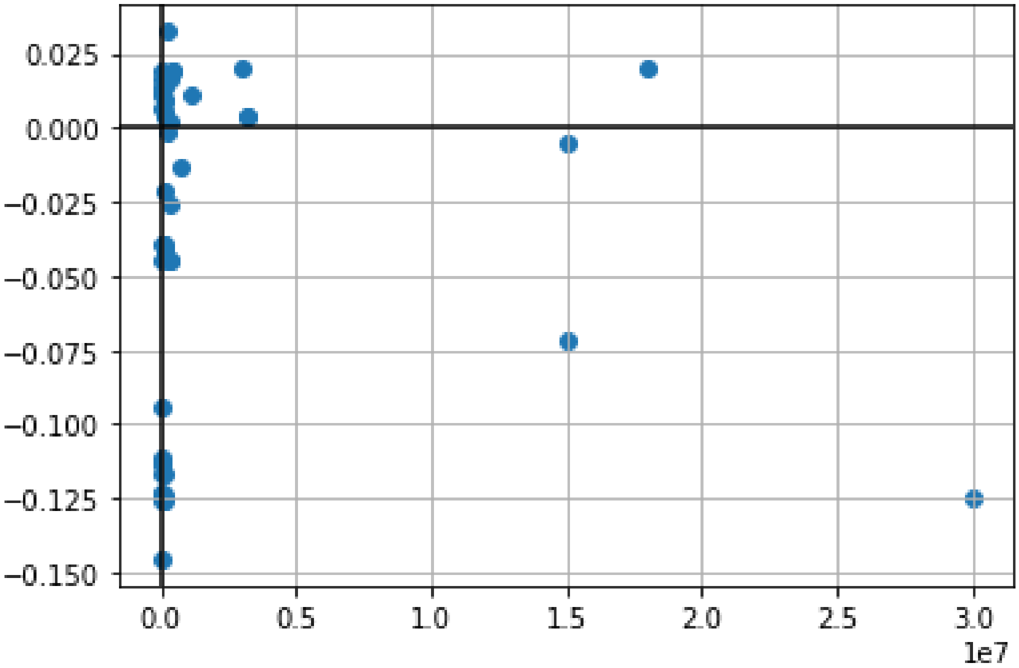

0.06
0.04
0.02
0.00
−0.02
−0.04
0
1
2
3
4
5
6
7
1e6


0.010
0.008
0.006
0.004
0.002
0.000
0
5000
10000
15000
20000
25000
30000


0.0000
−0.0025
−0.0050
−0.0075
−0.0100
−0.0125
−0.0150
−0.0175
−0.0200
0
1000
2000
3000
4000
5000
6000
7000
8000


0.006
0.004
0.002
0.000
−0.002
−0.004
−0.006
−0.008
0
20000
40000
60000
80000
100000
120000
140000





Evaluating Every Single Trade

[12]:
```
hists = {}

# returns a function that gets the spy price num_days from today
def get_future_stock_price(ticker, today, num_days):

    # print('Finding %change in price after', num_days, 'days for', ticker,␣
 ↪'on', today)

    try:
        stock_history = yf.Ticker(ticker).history(period='5y')

        # calculate current stock data
        today_index = market_open_days.get_loc(today, method='nearest')
        today_close = stock_history.iloc[stock_history.index.get_loc(today,␣
 ↪method='nearest')]['Close']

        # calculate future date and get stock data
        future_index = today_index + num_days
        future_date = market_open_days[future_index]
        future_close = stock_history.iloc[stock_history.index.
 ↪get_loc(future_date, method='nearest')]['Close']
        return (future_close - today_close) / today_close
    except Exception as e:
        return 'error' + str(e.message) if hasattr(e, 'message') else 'error:␣
 ↪unknown exception'
```

[13]:
```
stock_trades = df[df['ticker'] != '--']

# we're processing an insane amount of data here, so let's save this dataframe␣
 ↪back to the disk
# so we can read it back easily
try:
    stock_trades = pd.read_csv('stock_trading_performance.csv')
except FileNotFoundError:
    print('calculating...')
    stock_trades['5_day_pct_change'] = stock_trades.apply(lambda row:␣
 ↪get_future_stock_price(row['ticker'], row['transaction_date'], 5), axis=1)
    print('done with 5 day')
    stock_trades['15_day_pct_change'] = stock_trades.apply(lambda row:␣
 ↪get_future_stock_price(row['ticker'], row['transaction_date'], 15), axis=1)
    print('done with 15 day')
    # stock_trades.apply(lambda row: get_future_stock_price(row['ticker'],␣
 ↪row['transaction_date'], 5), axis=1)
    stock_trades.to_csv('stock_trading_performance.csv')
```

[20]:

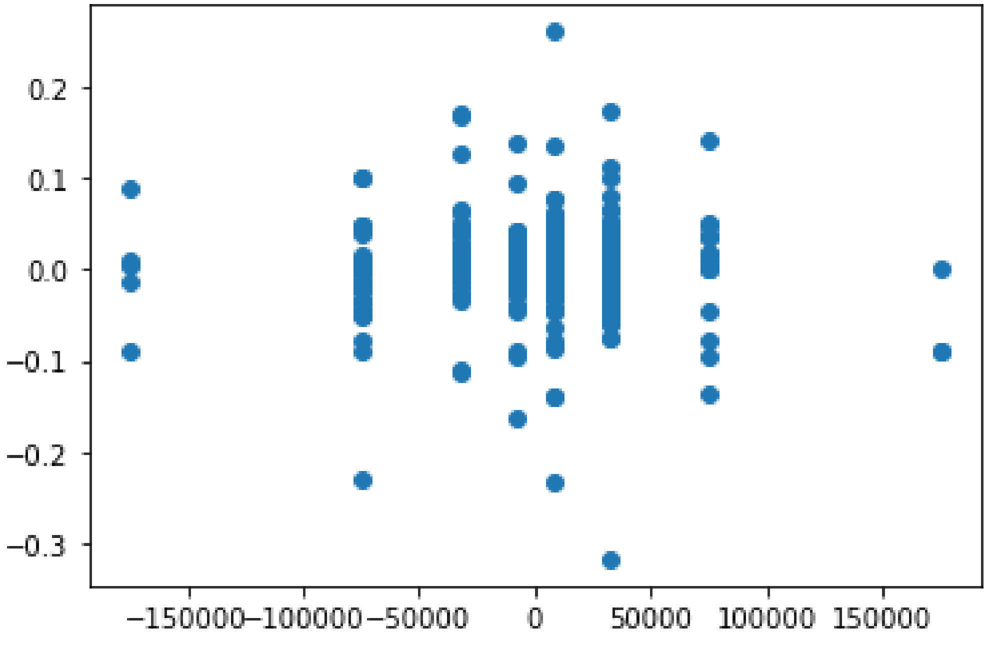
```
print(stock_trades['owner'].unique())
print(len(stock_trades['senator'].unique()))
```

```
['Spouse' 'Joint' 'Child' 'Self']
41
```

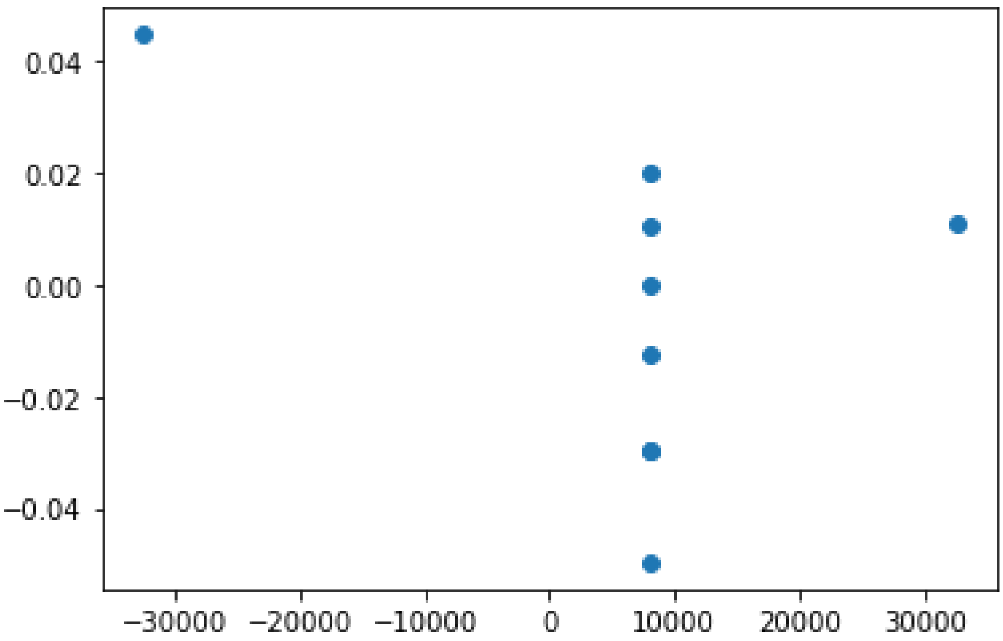
[17]:

```
stock_trades['5_day_pct_change'] = pd.
 ↪to_numeric(stock_trades['5_day_pct_change'], errors='coerce')
stock_trades['15_day_pct_change'] = pd.
 ↪to_numeric(stock_trades['15_day_pct_change'], errors='coerce')

for senator in stock_trades['senator'].unique():
    for owner in stock_trades['owner'].unique():
        print(senator, owner)

        stock_trades_by_senator = stock_trades[(stock_trades['senator'] ==␣
 ↪senator) & (stock_trades['owner'] == owner)]

        plt.scatter(stock_trades_by_senator['mid_amt'],␣
 ↪stock_trades_by_senator['5_day_pct_change'])
        plt.show()
# plt.scatter(stock_trades['mid_amt'], stock_trades['5_day_pct_change'])
# plt.show()
# plt.scatter(stock_trades['mid_amt'], stock_trades['15_day_pct_change'])
```

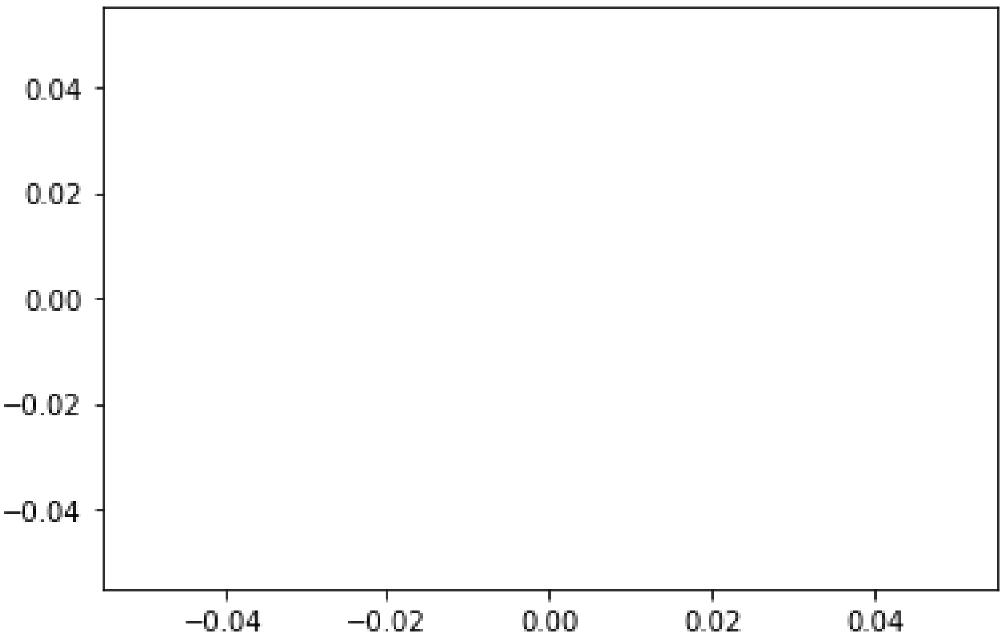
```
Pat Roberts Spouse
```



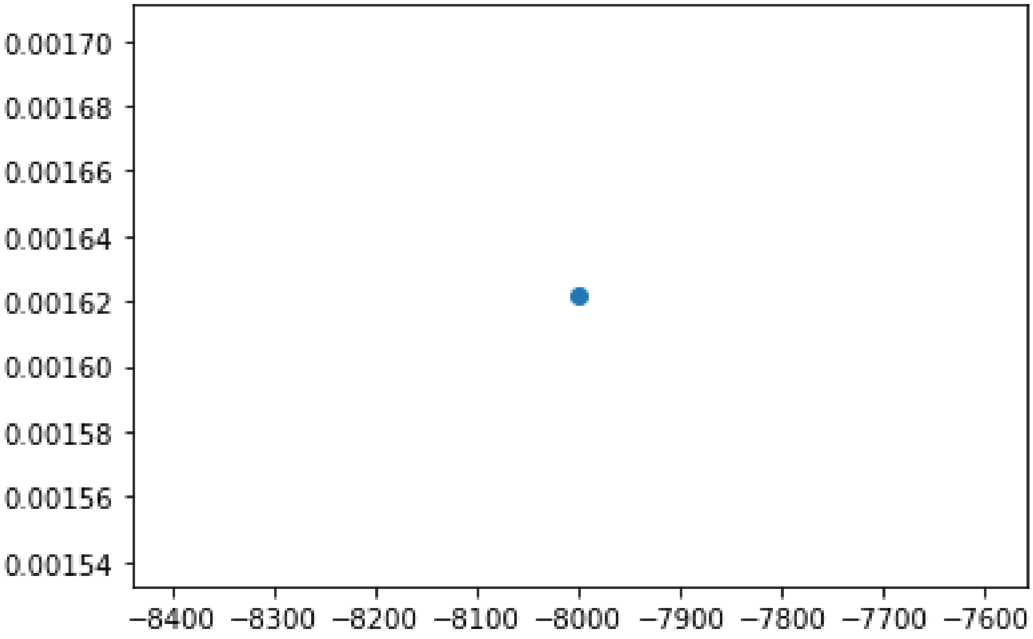
Pat Roberts Joint



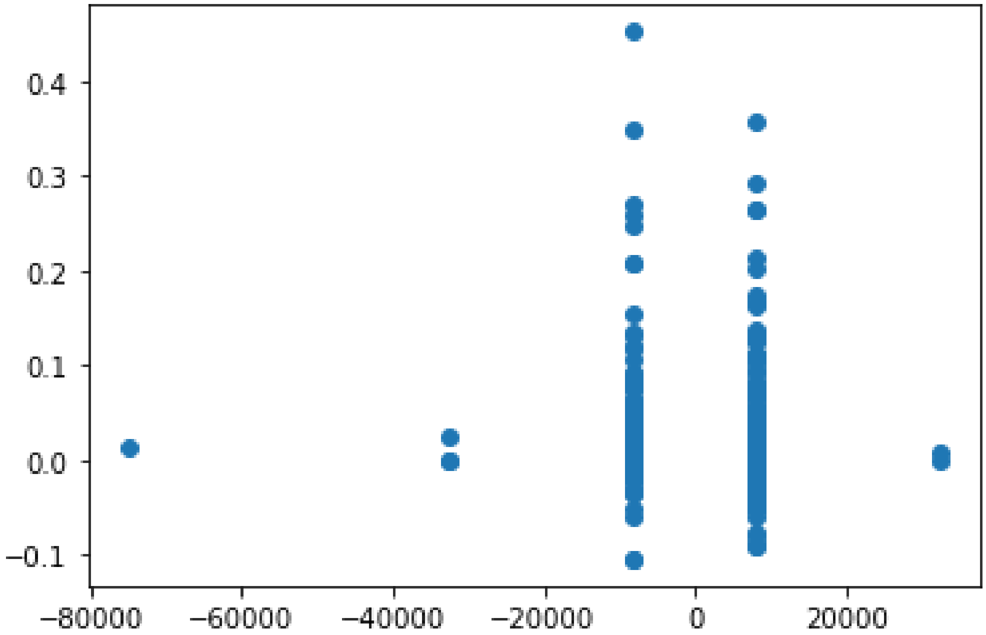
Pat Roberts Child



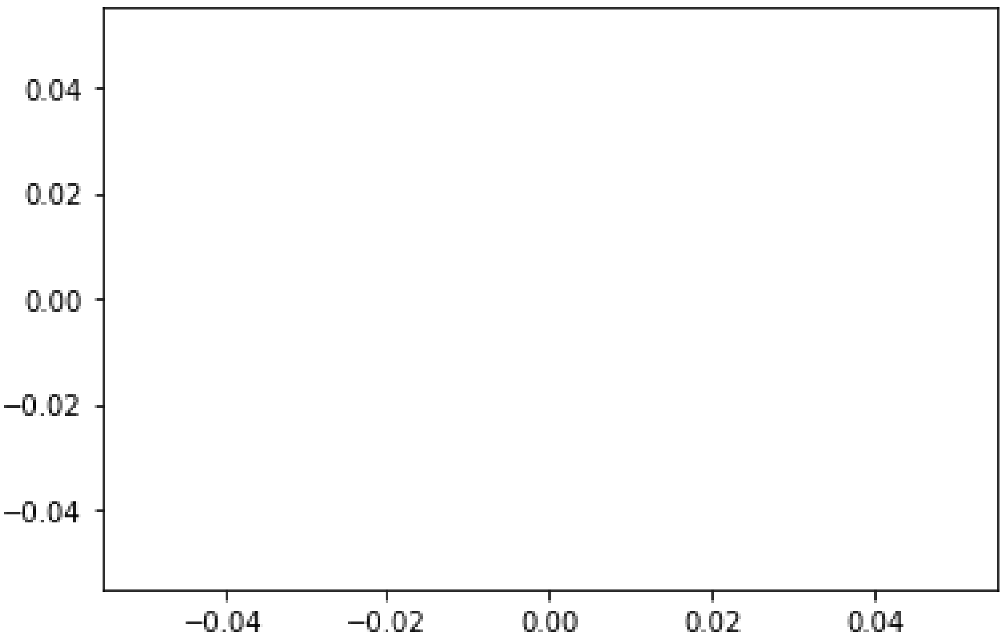
Pat Roberts Self



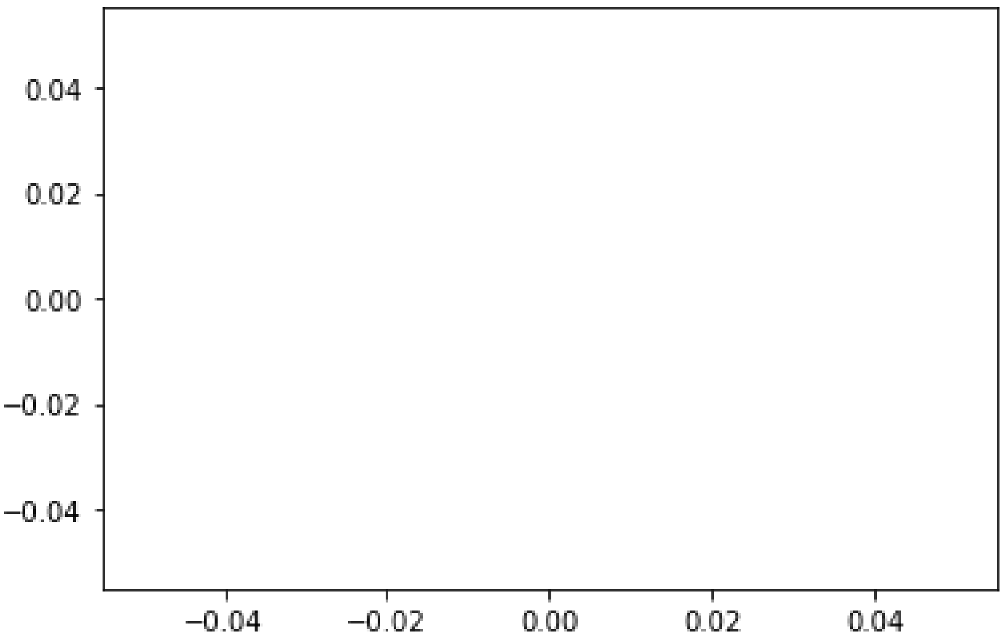
Thomas R Carper Spouse



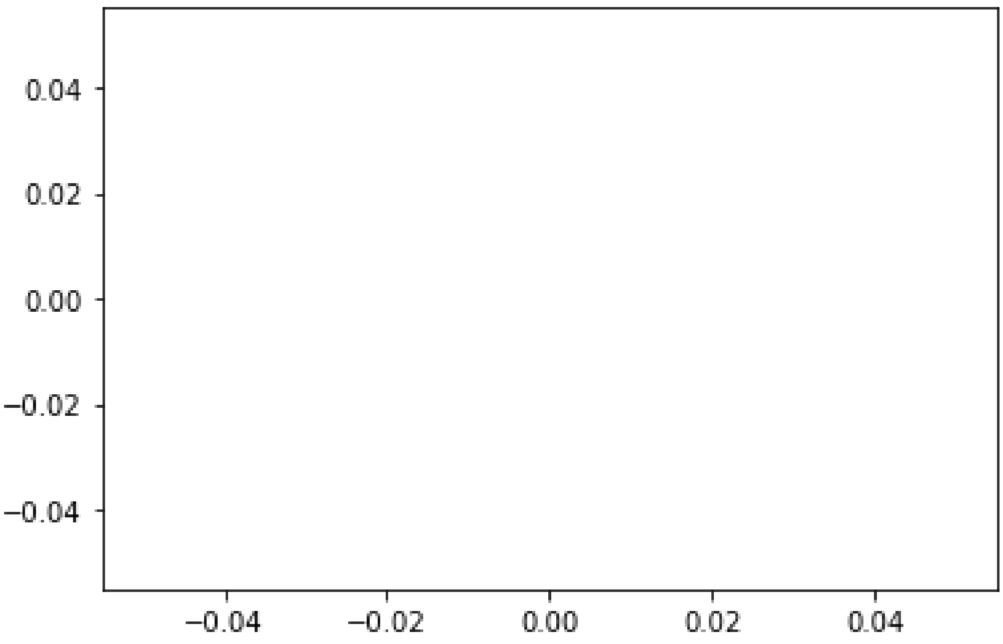
Thomas R Carper Joint



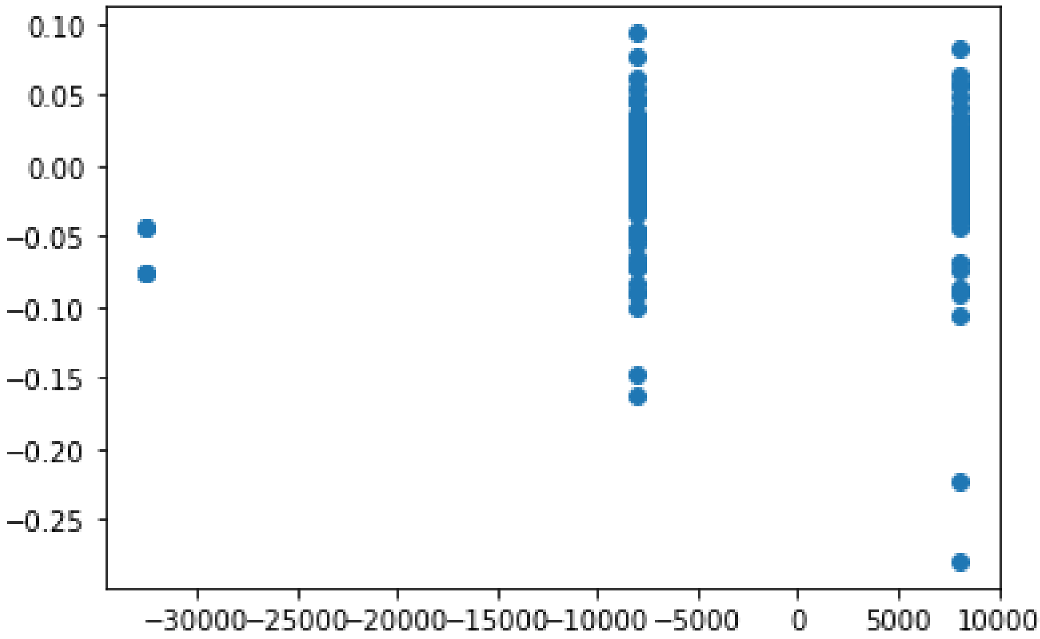
Thomas R Carper Child



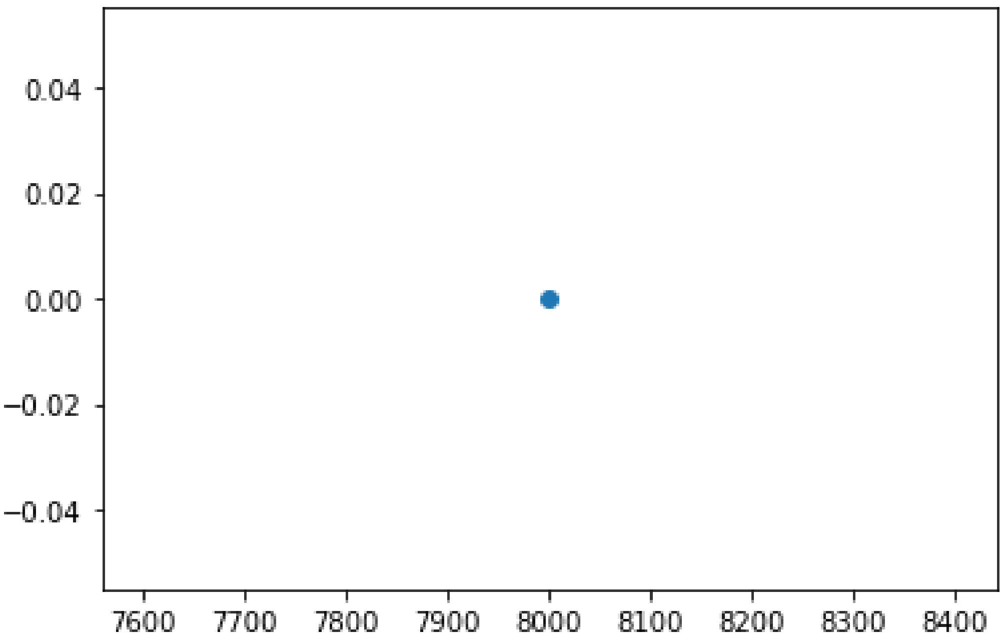
Thomas R Carper Self



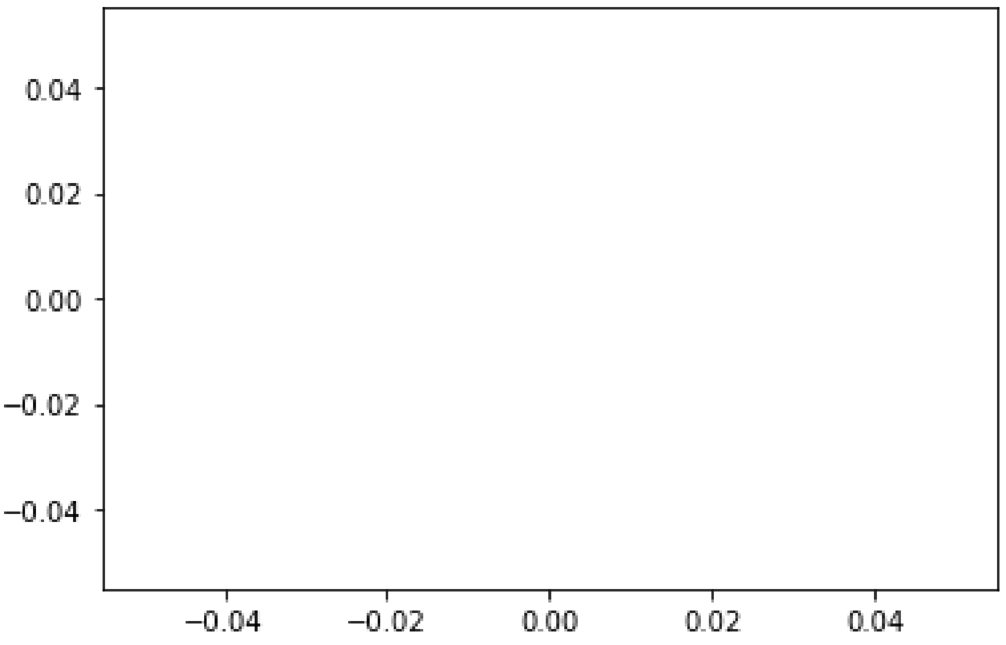
William Cassidy Spouse



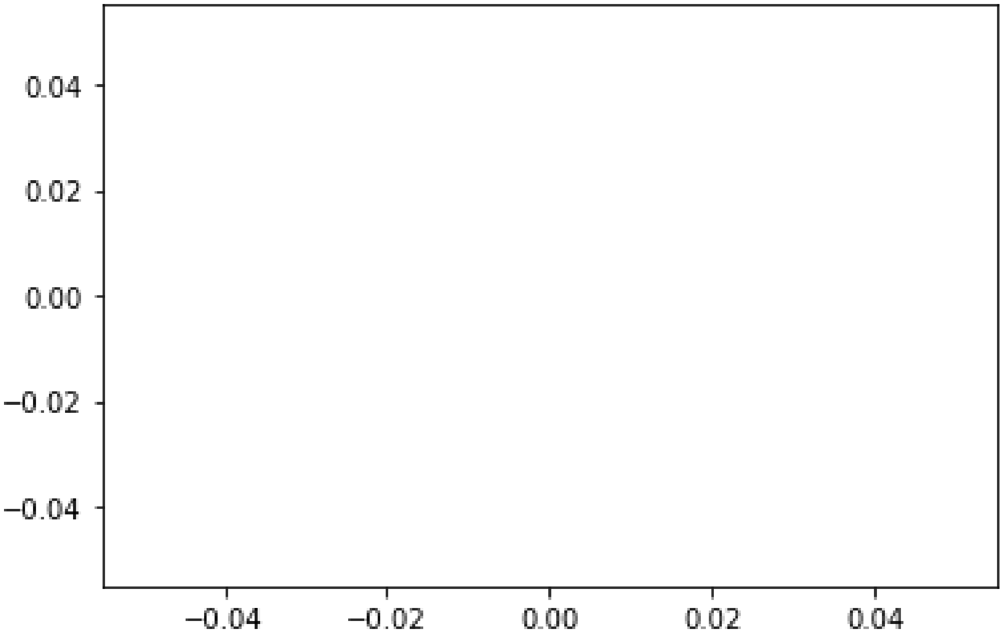
William Cassidy Joint



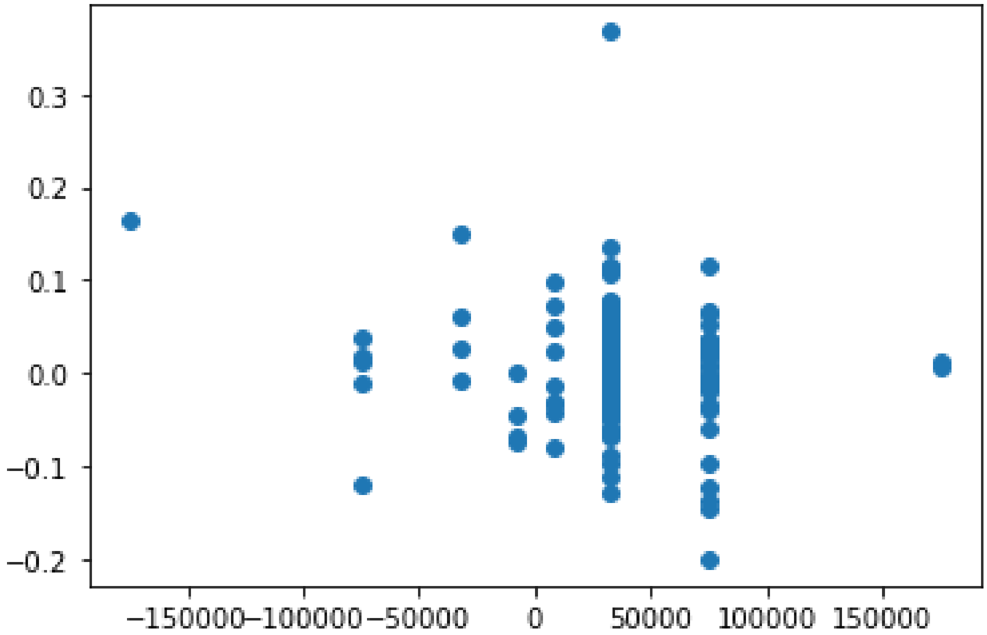
`William Cassidy Child`



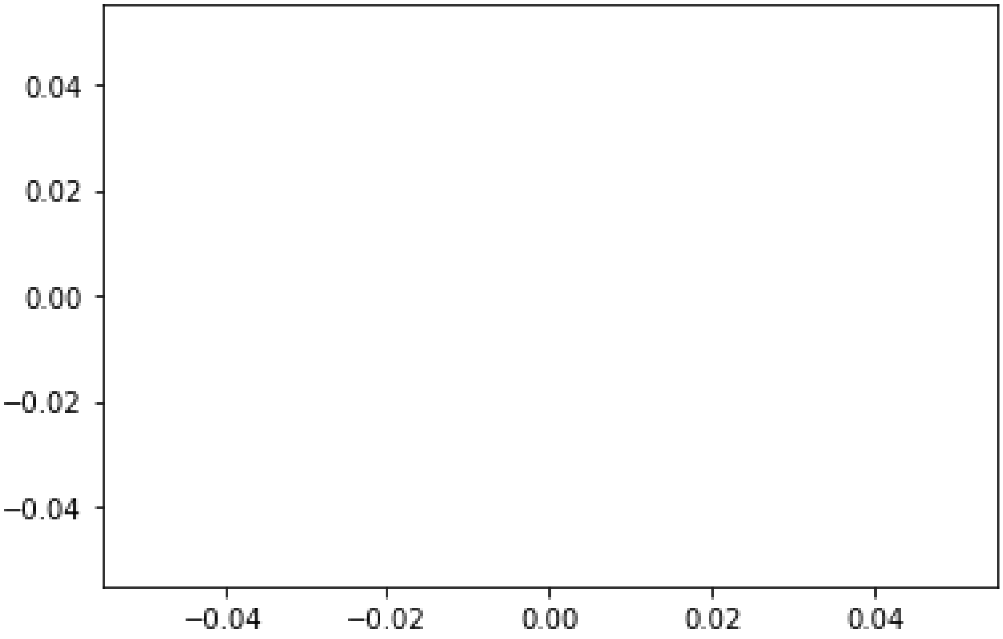
`William Cassidy Self`



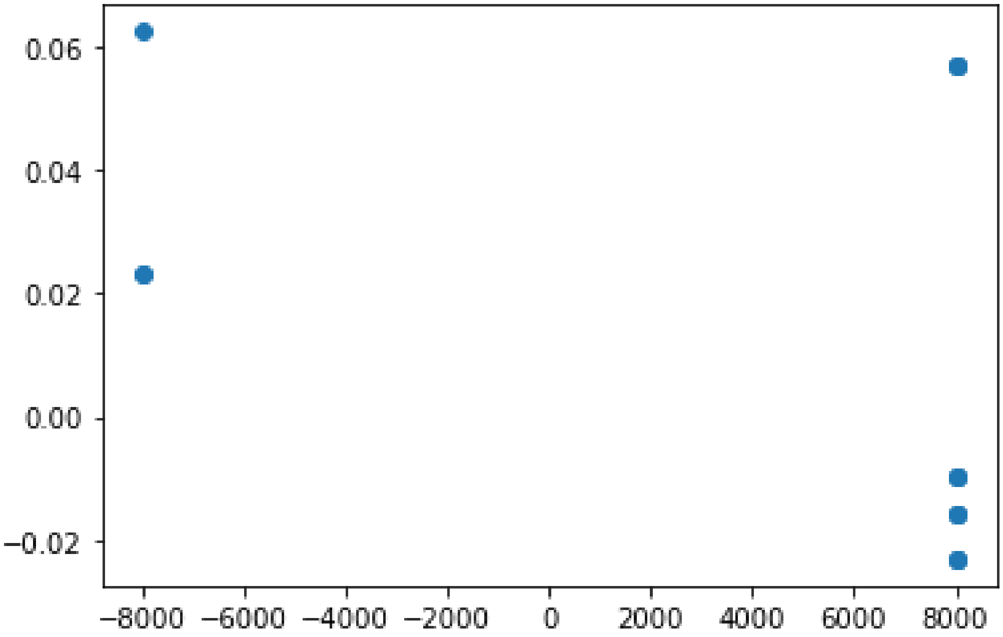
Ron L Wyden Spouse



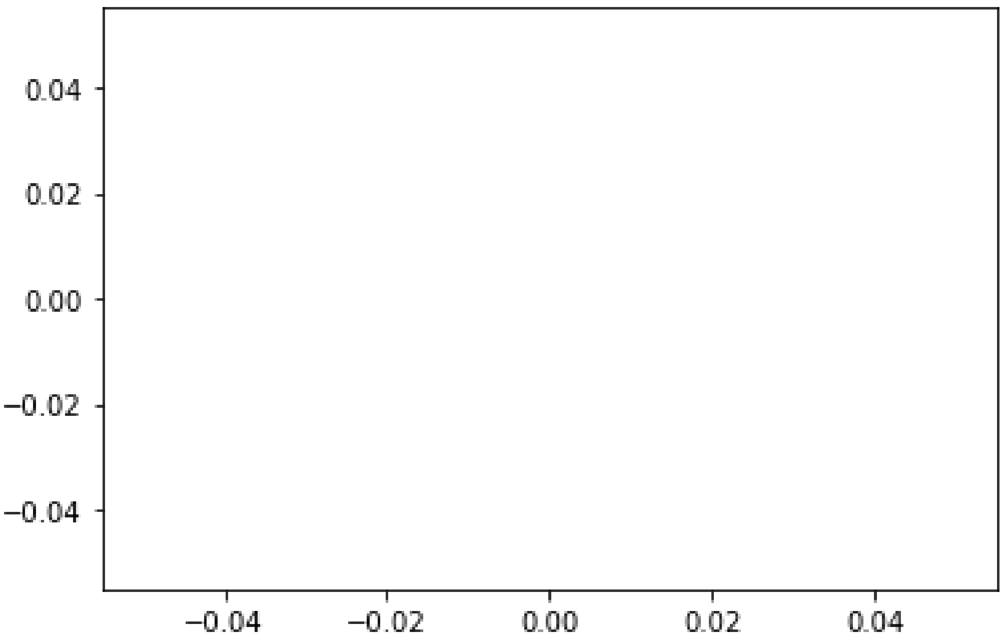
Ron L Wyden Joint



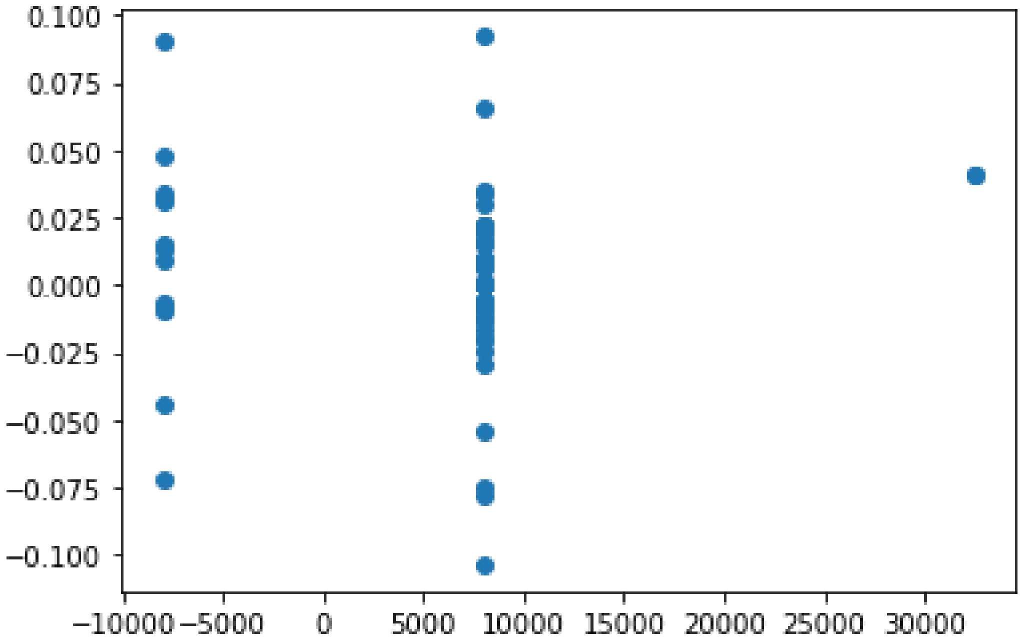
Ron L Wyden Child



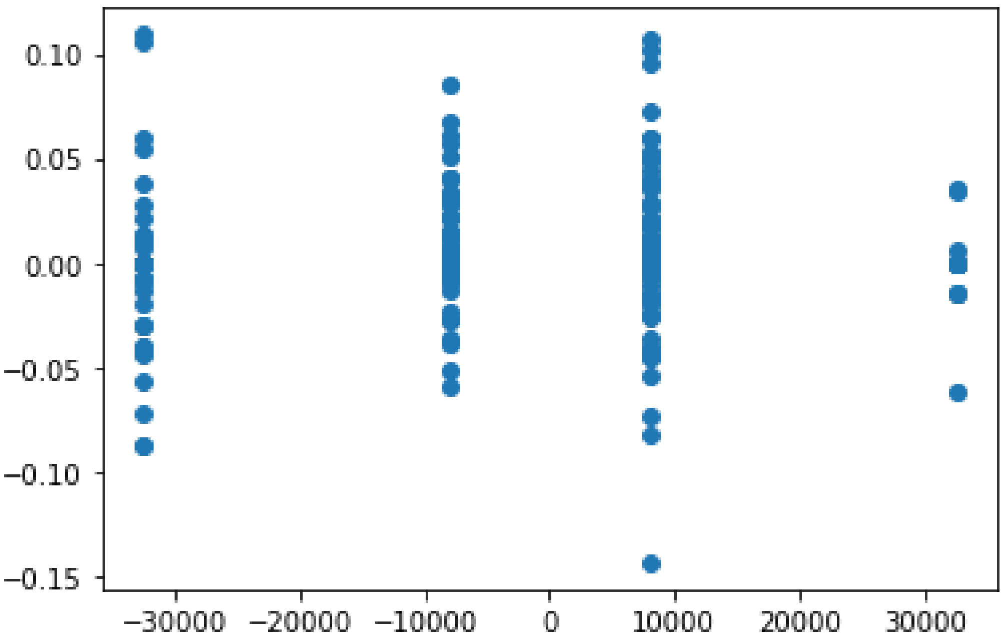
Ron L Wyden Self



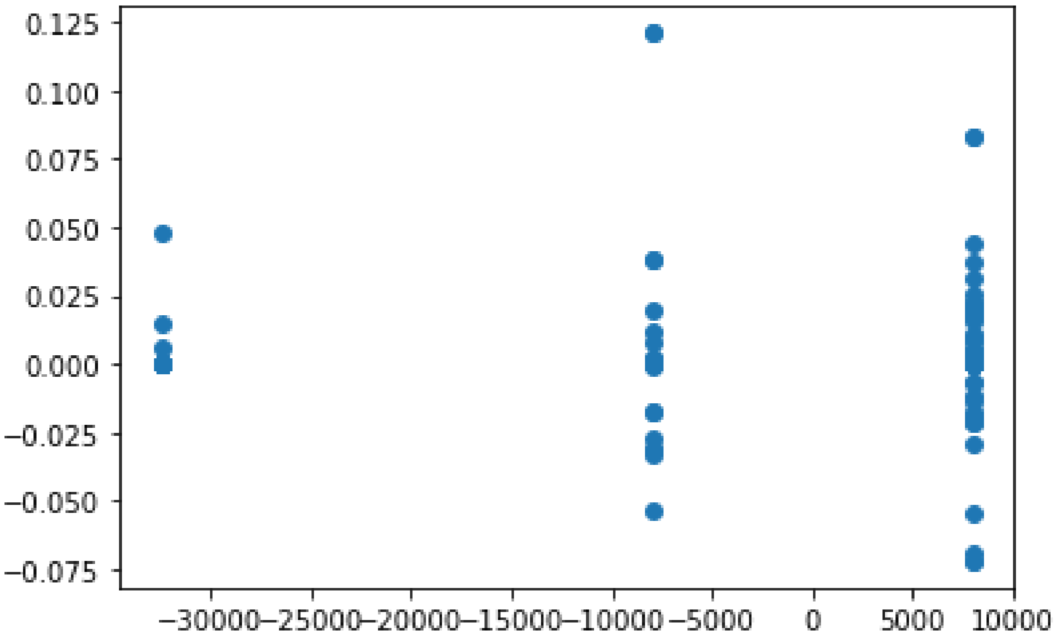
Sheldon Whitehouse Spouse



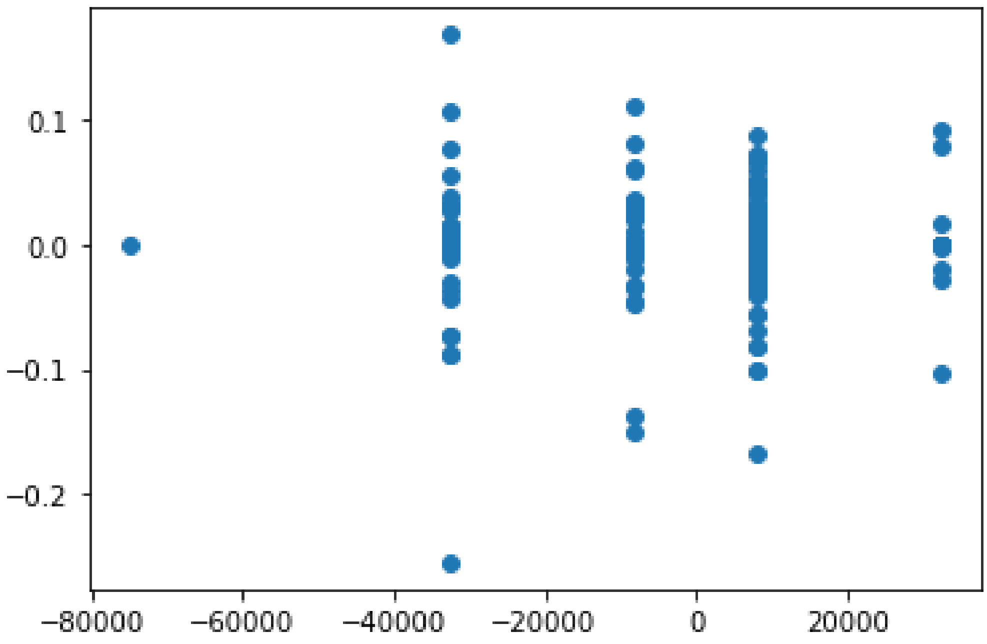
Sheldon Whitehouse Joint



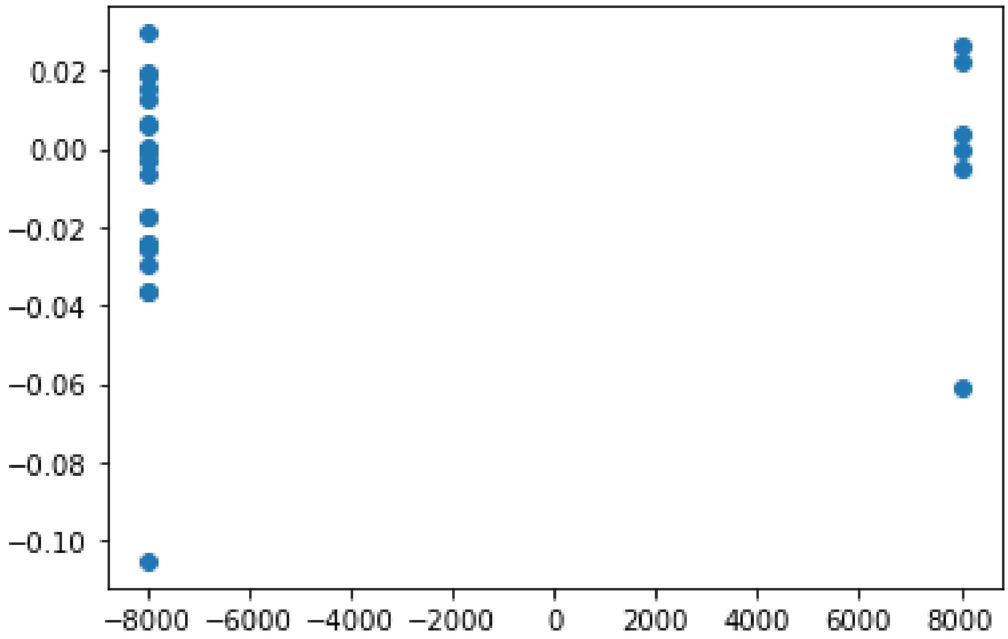
Sheldon Whitehouse Child



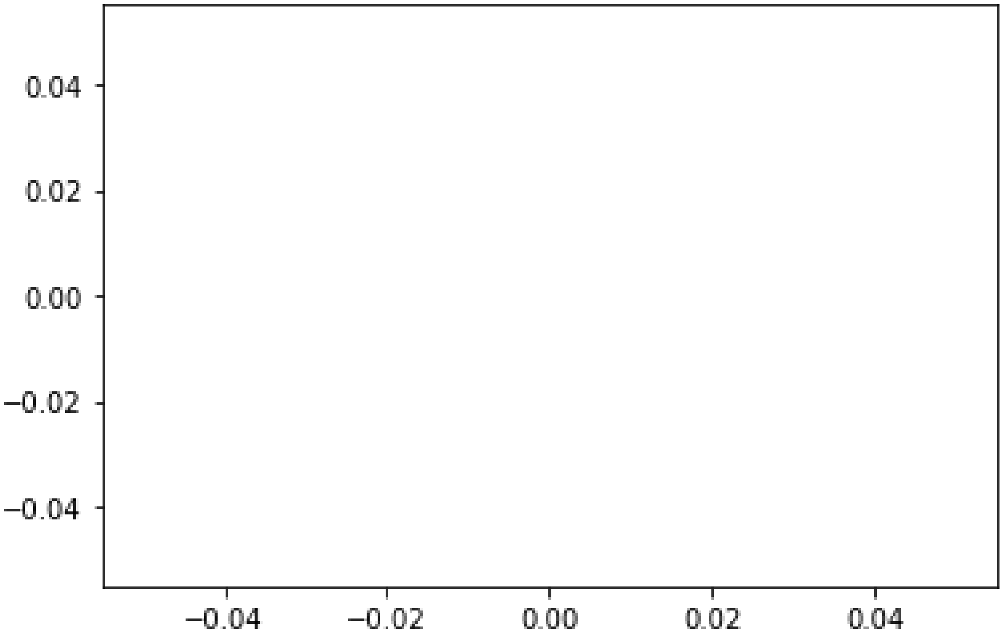
Sheldon Whitehouse Self



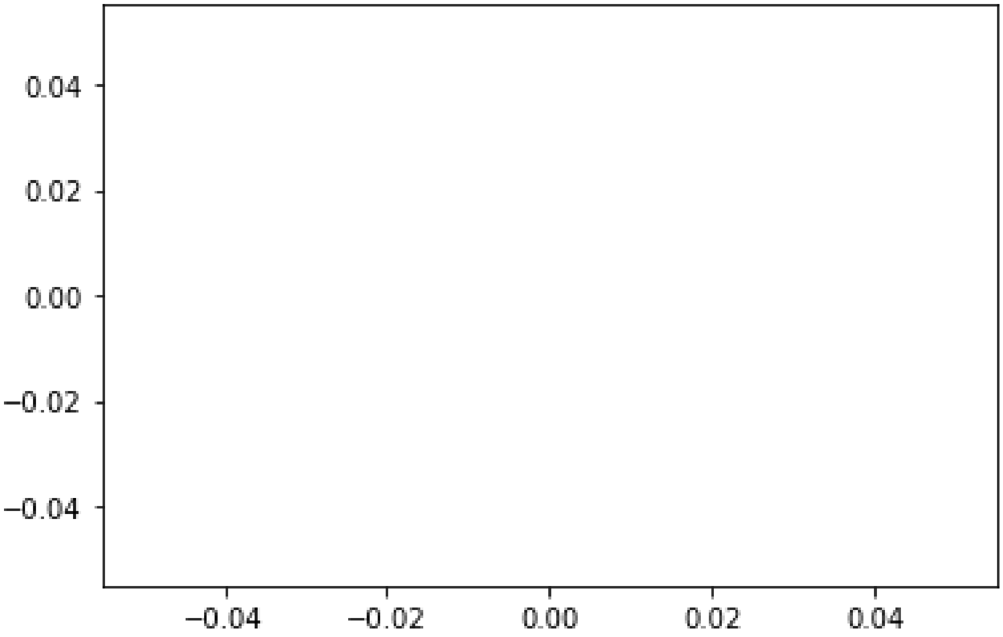
Angus S King, Jr. Spouse



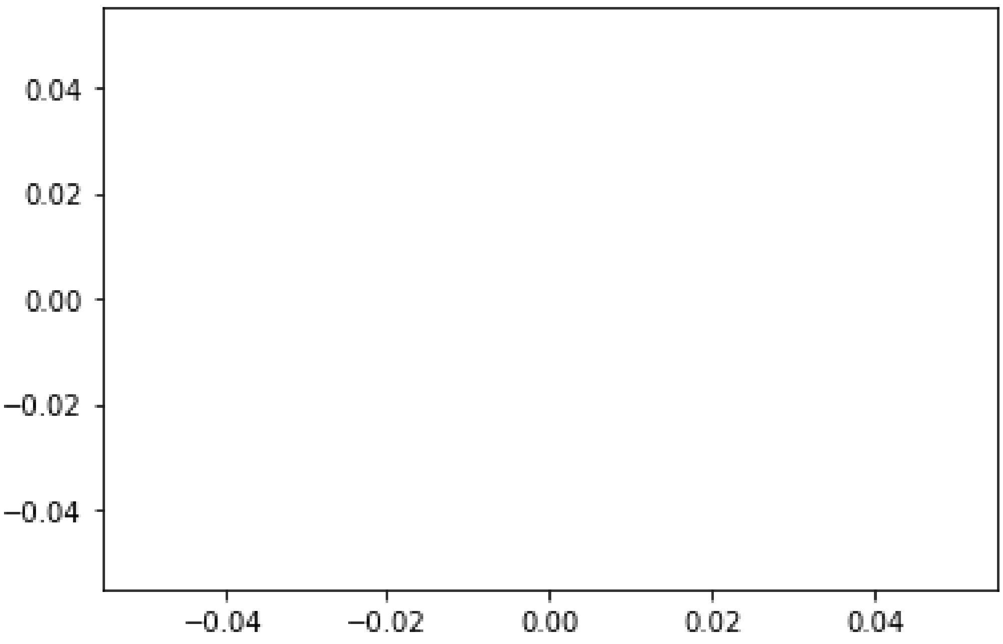
Angus S King, Jr. Joint



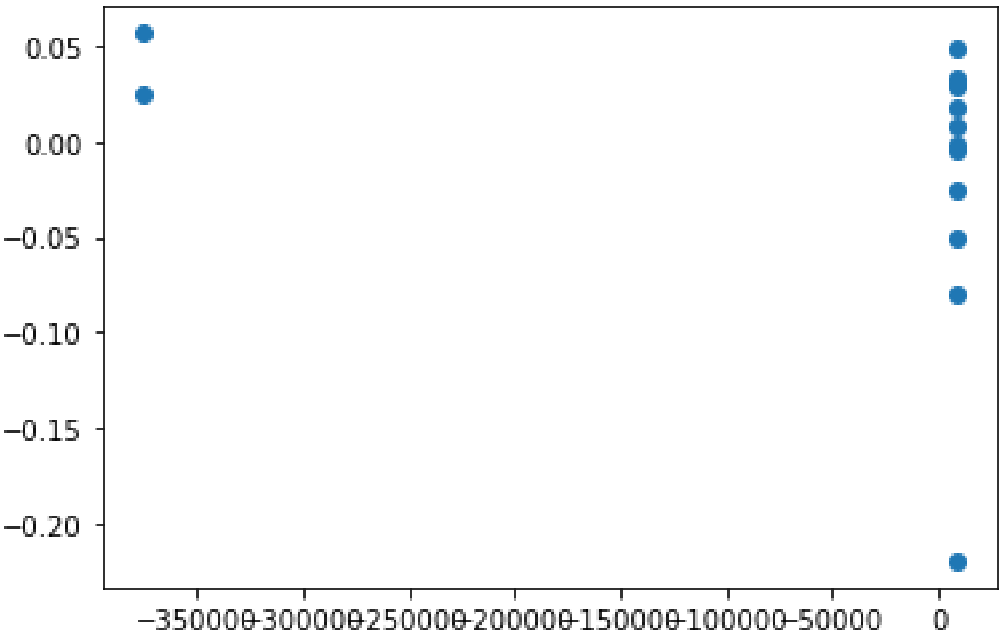
`Angus S King, Jr. Child`



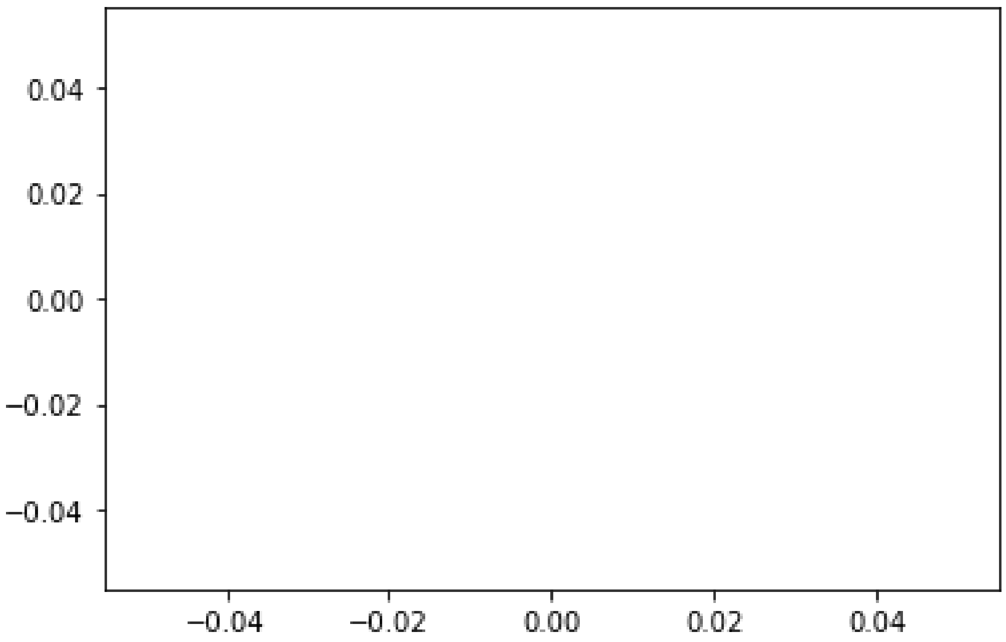
`Angus S King, Jr. Self`



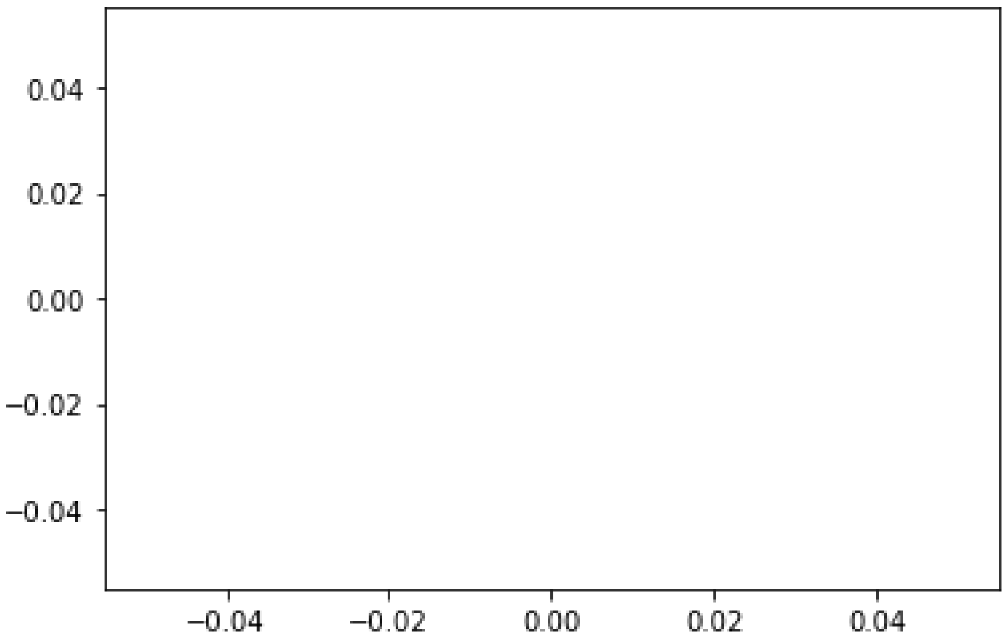
A. Mitchell Mcconnell, Jr. Spouse



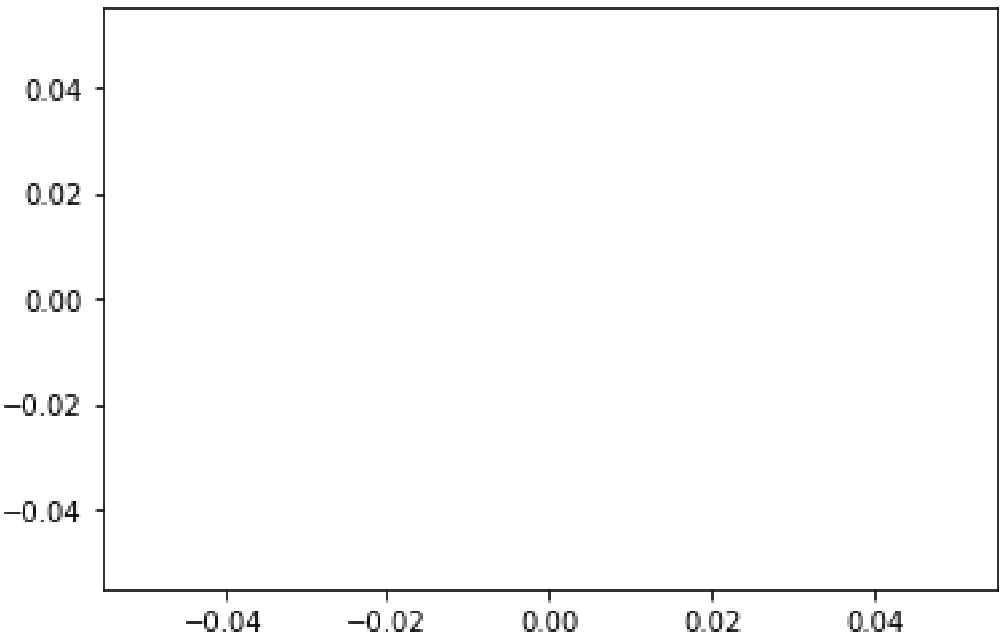
A. Mitchell Mcconnell, Jr. Joint



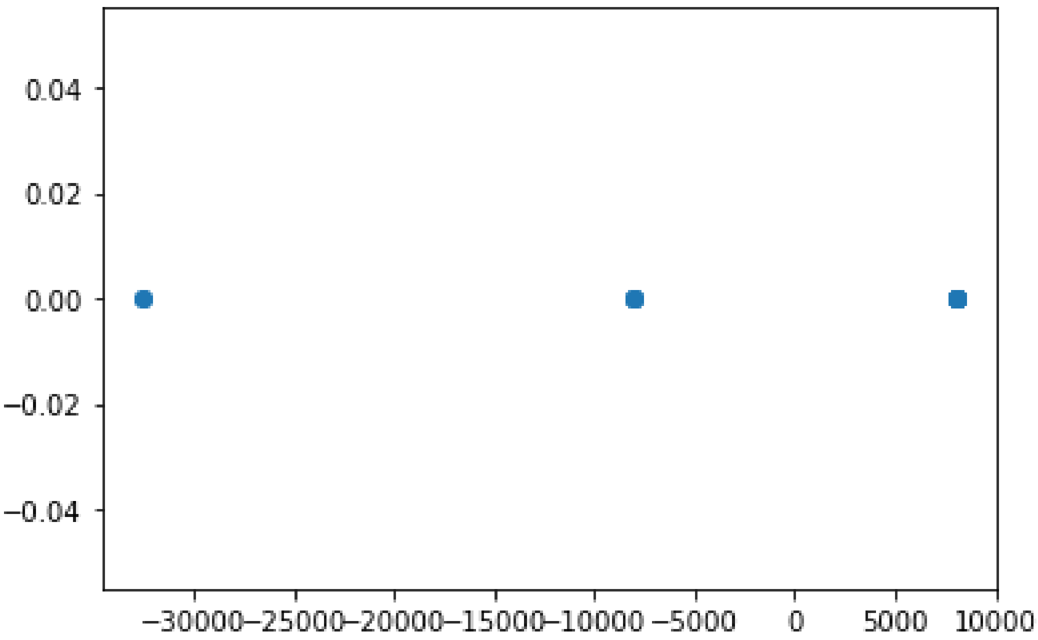
A. Mitchell Mcconnell, Jr. Child



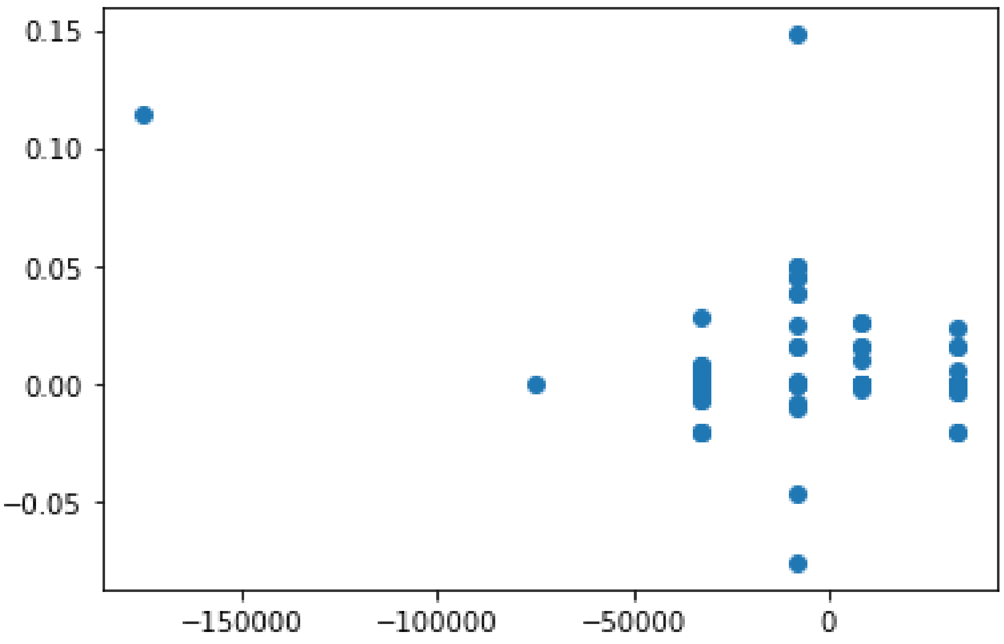
A. Mitchell Mcconnell, Jr. Self



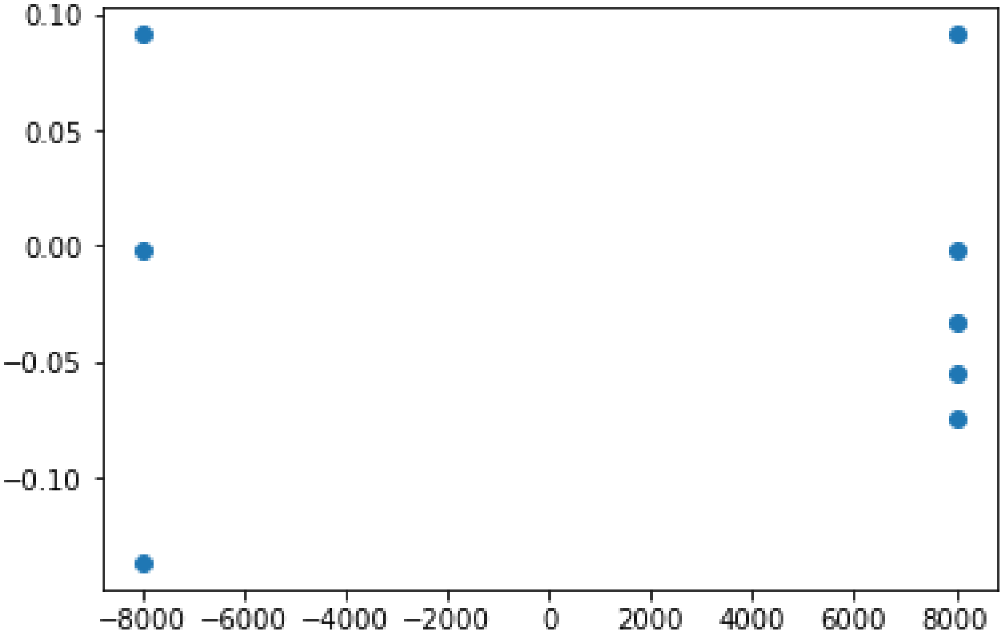
```
Patrick J Toomey Spouse
```



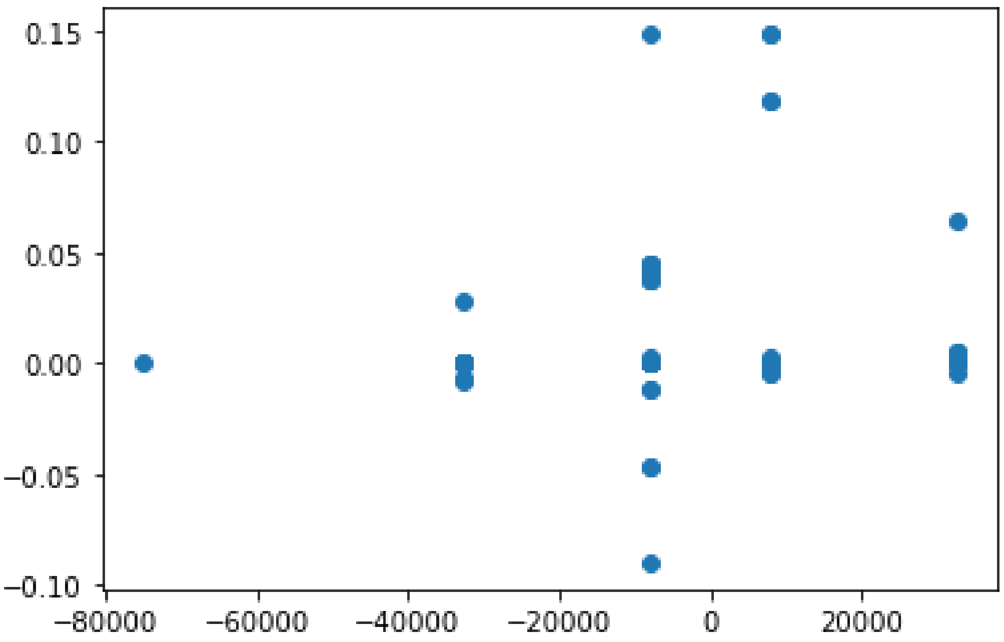
```
Patrick J Toomey Joint
```



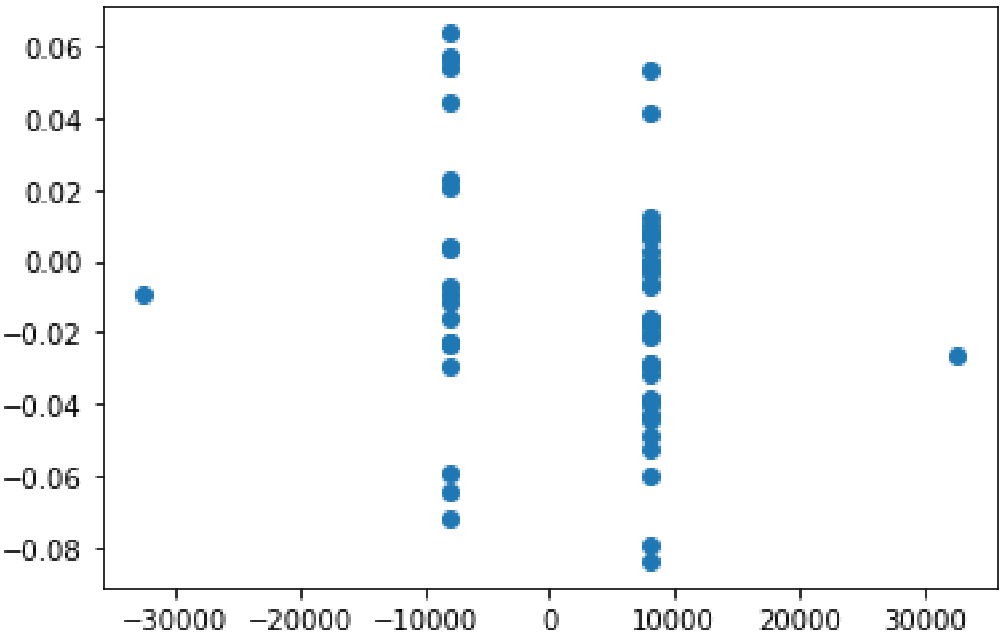
Patrick J Toomey Child



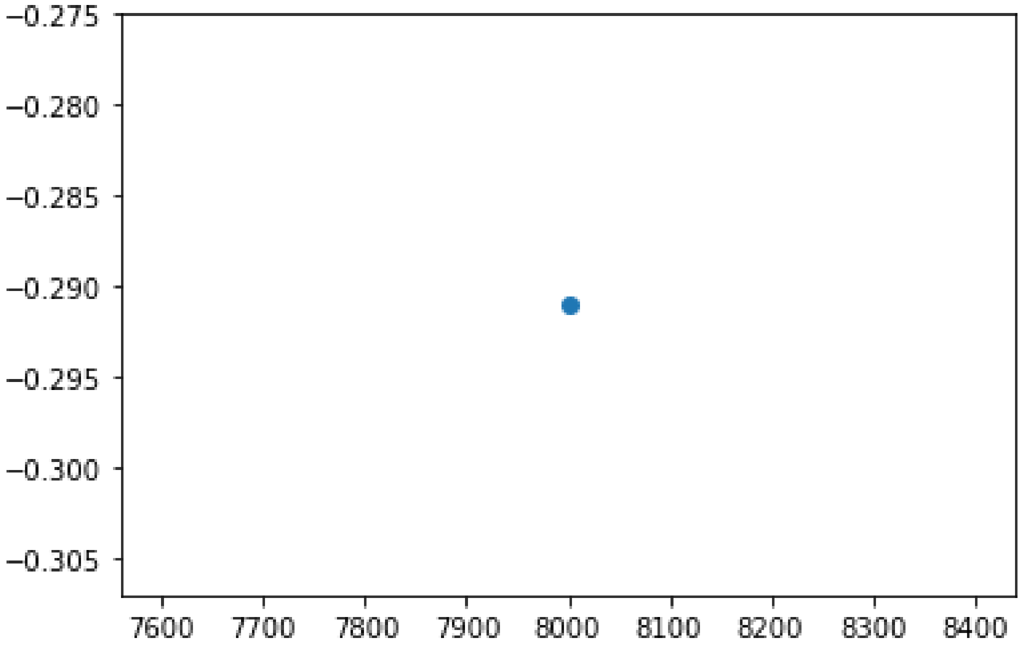
Patrick J Toomey Self



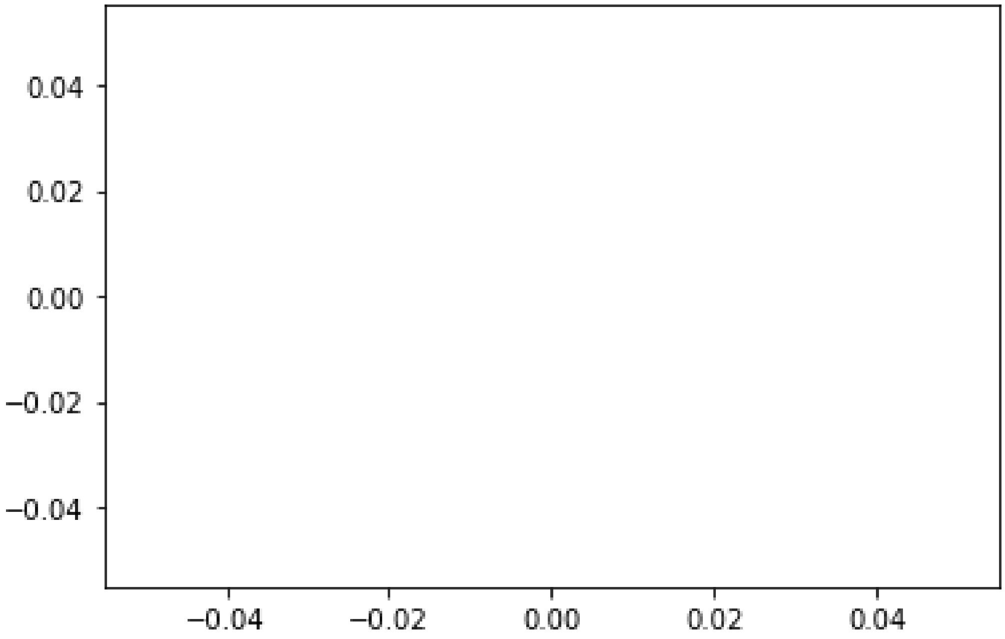
Jerry Moran, Spouse



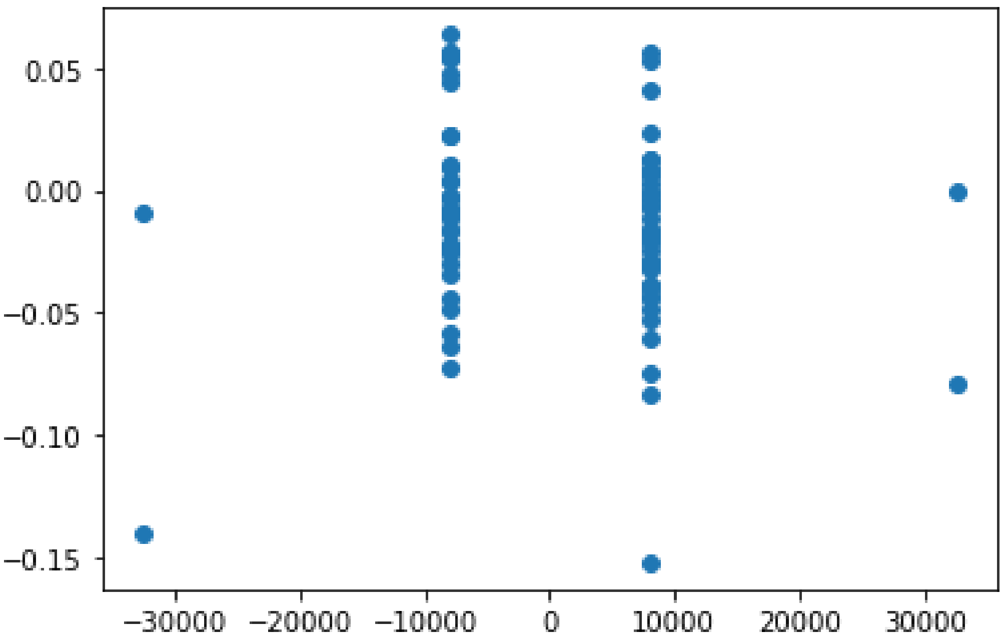
Jerry Moran, Joint



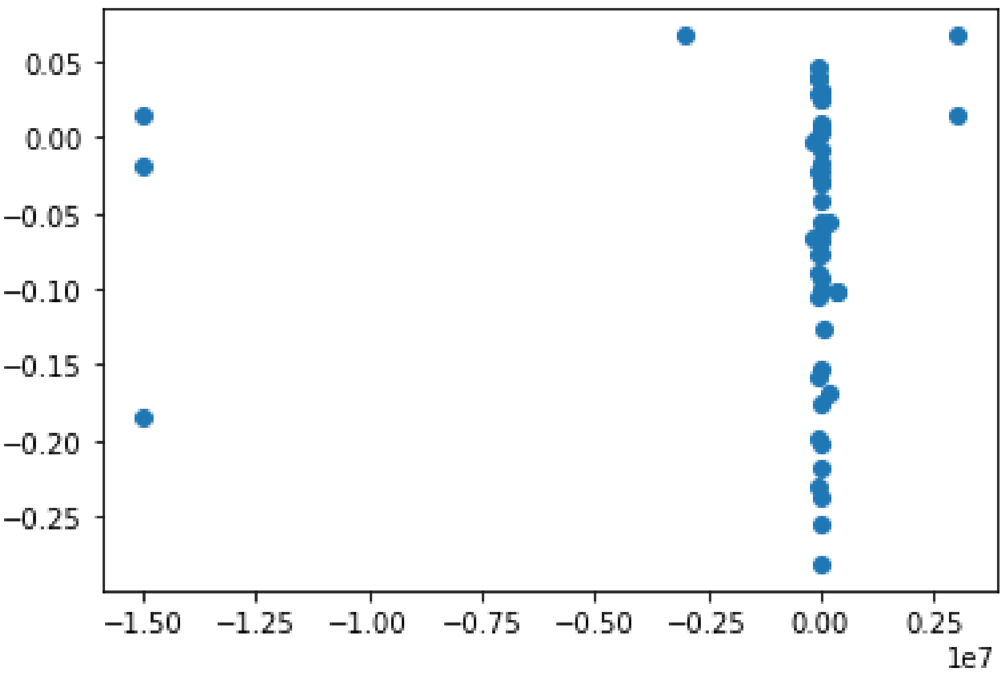
Jerry Moran, Child



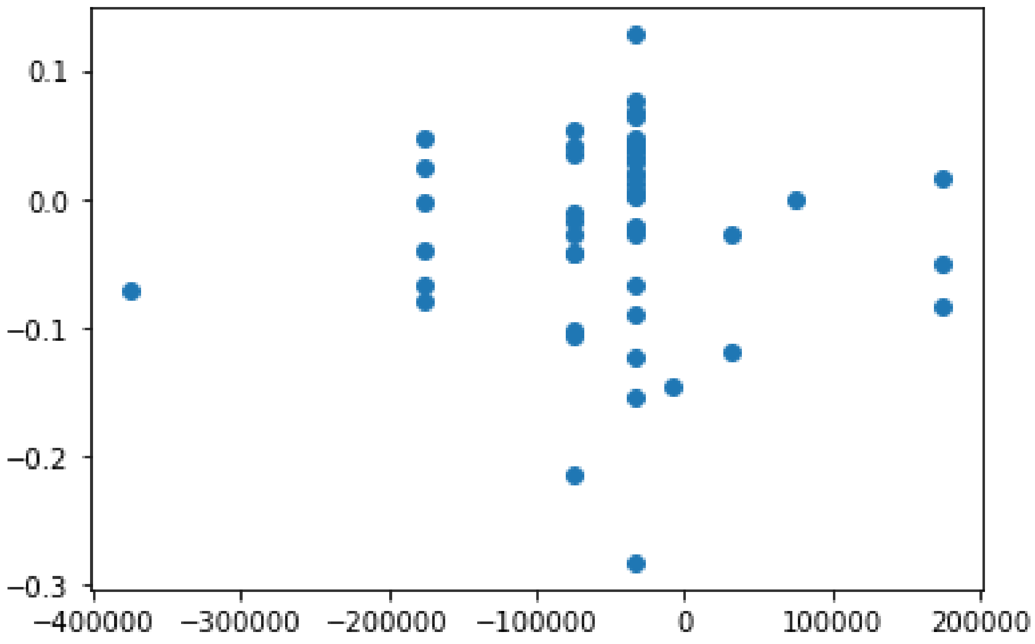
Jerry Moran, Self



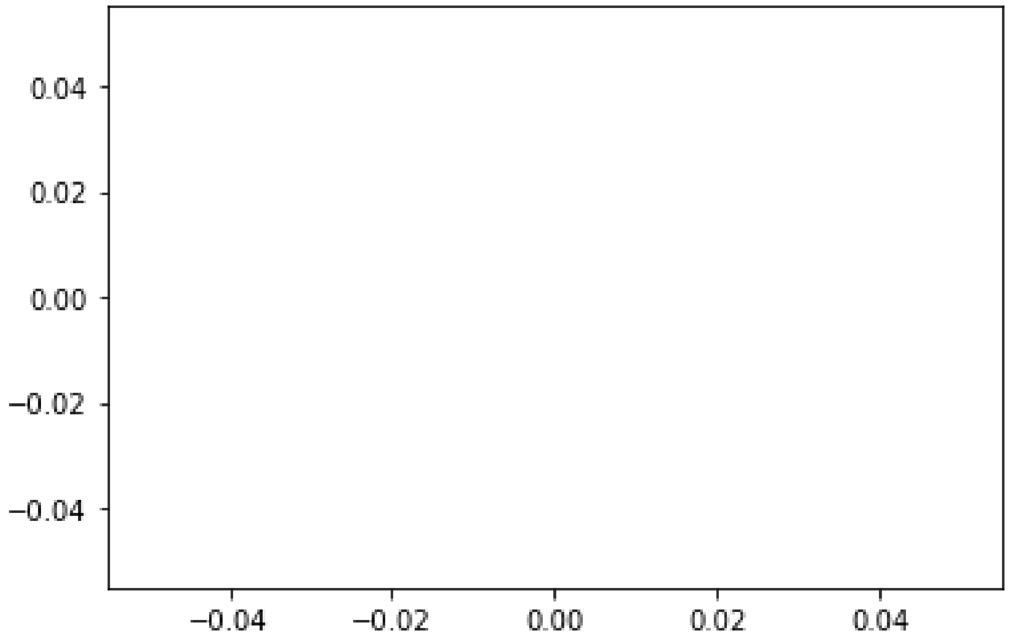
Kelly Loeffler Spouse



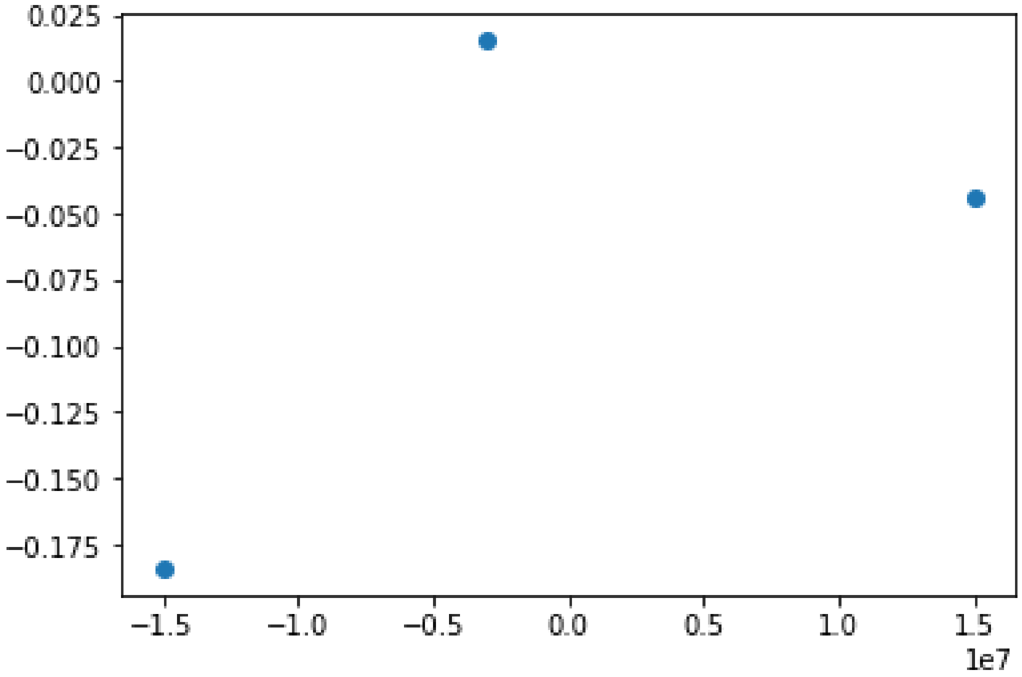
Kelly Loeffler Joint



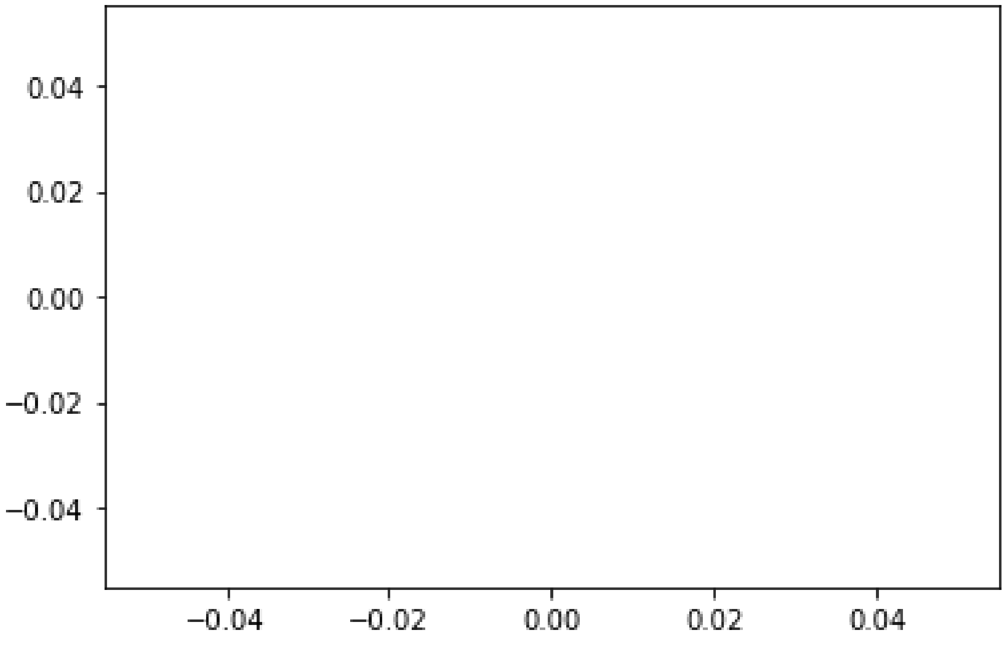
Kelly Loeffler Child



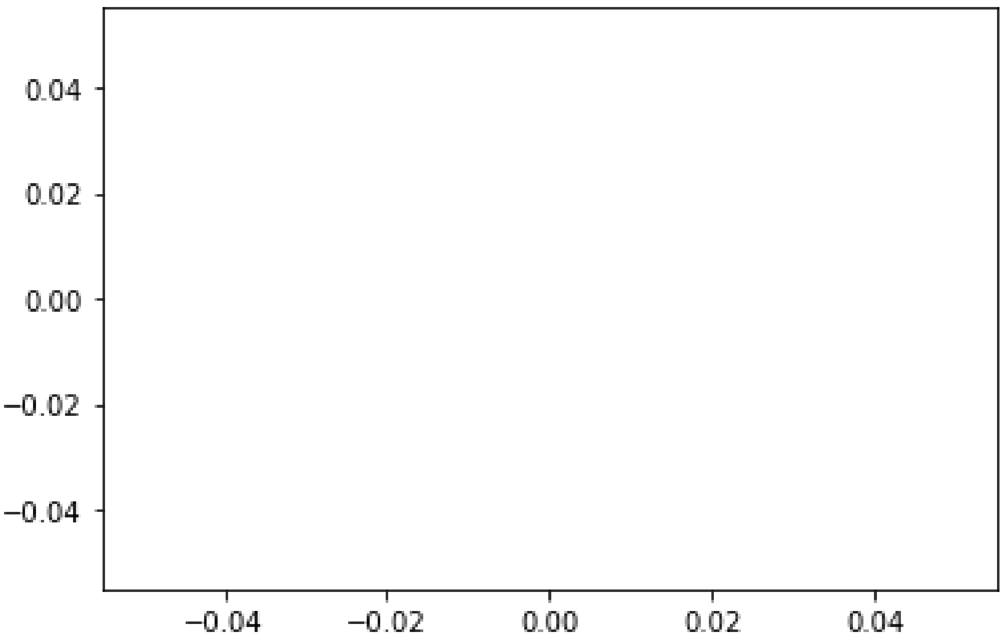
Kelly Loeffler Self



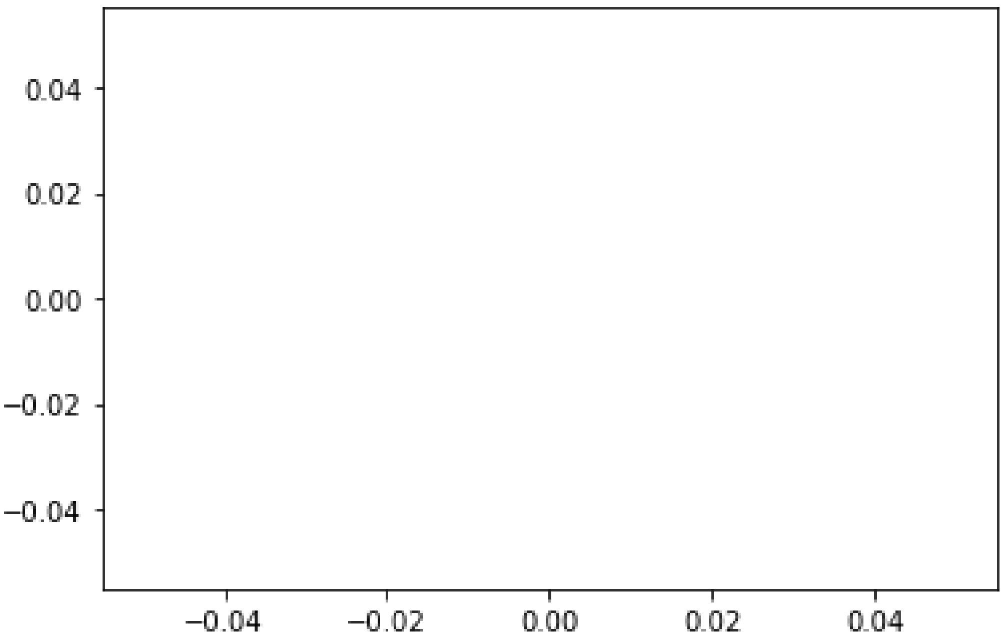
```
James M Inhofe Spouse
```



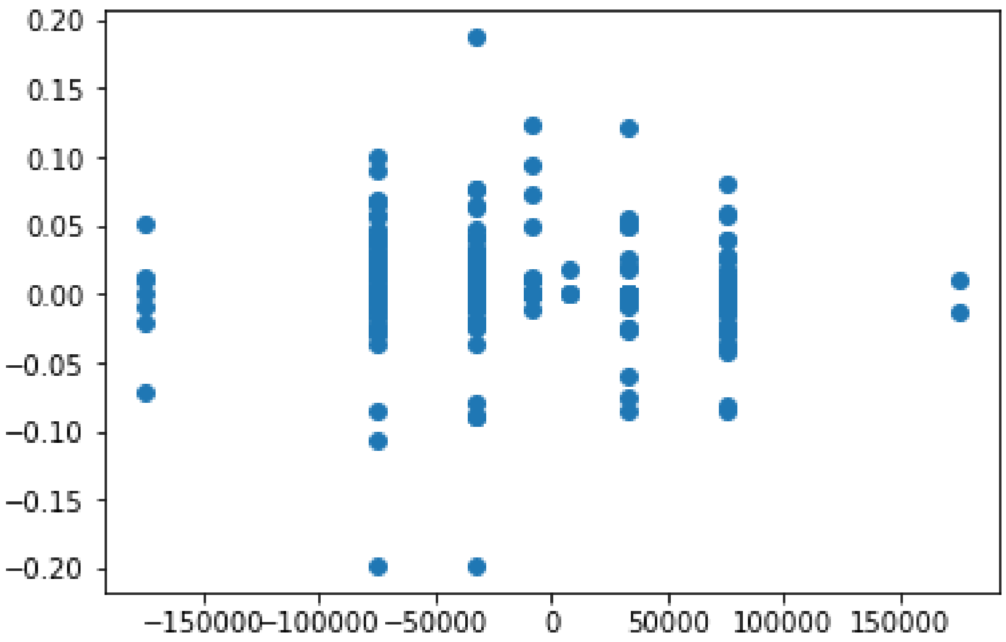
```
James M Inhofe Joint
```



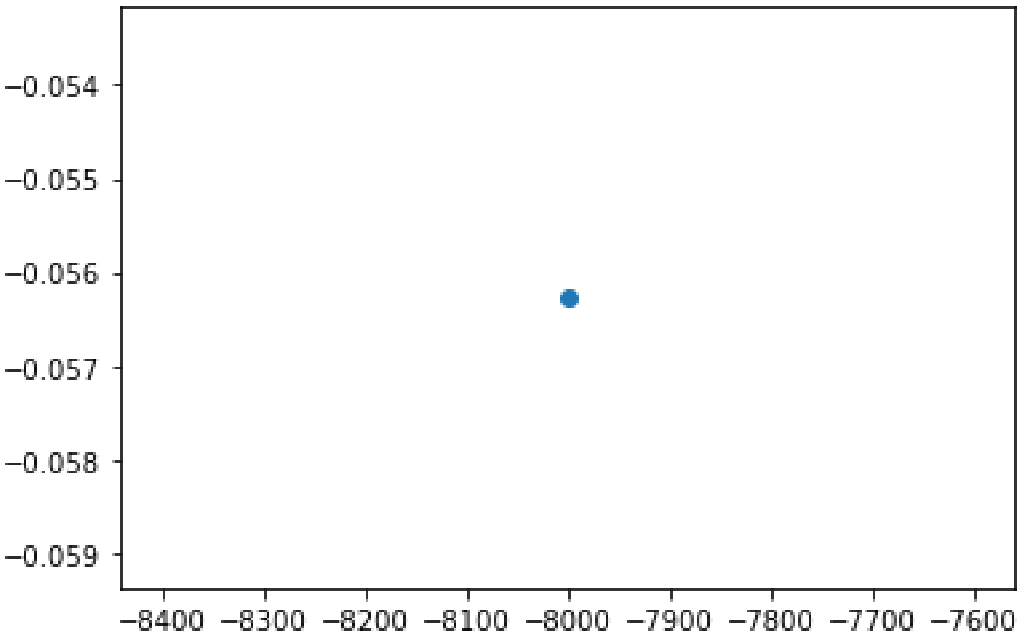
James M Inhofe Child



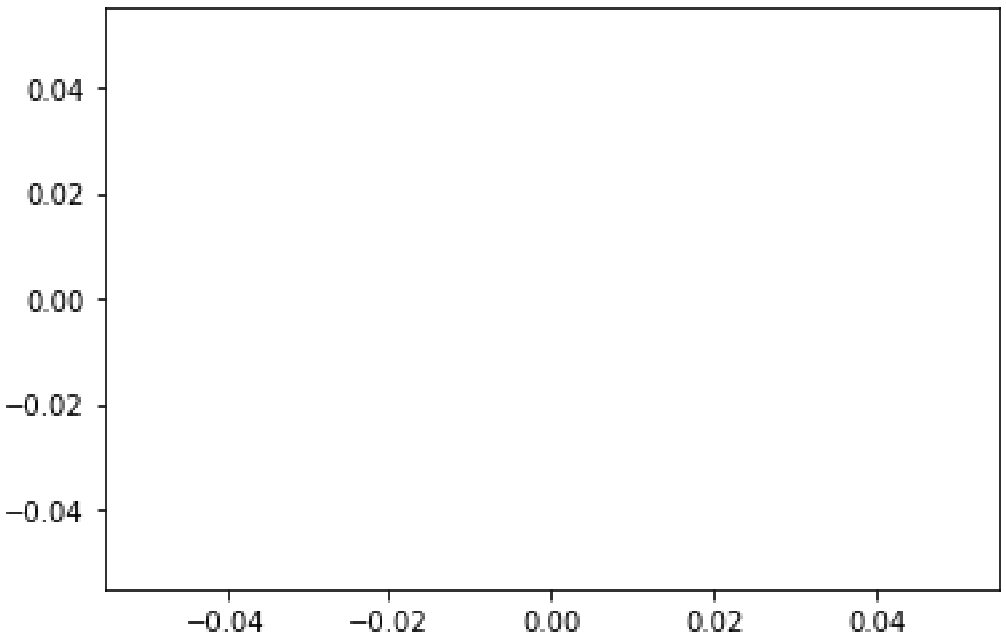
James M Inhofe Self



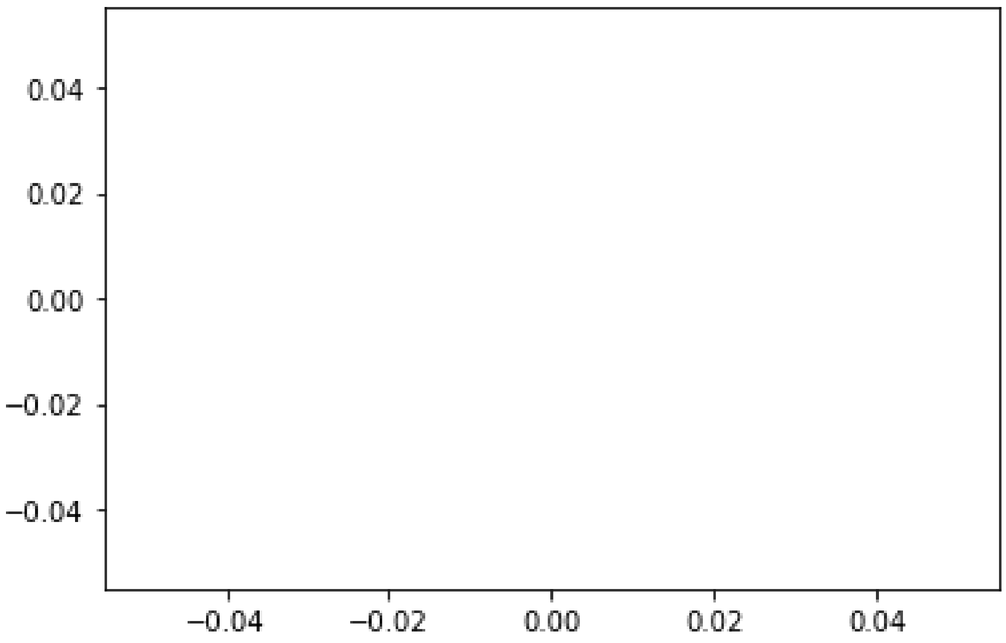
Timothy M Kaine Spouse



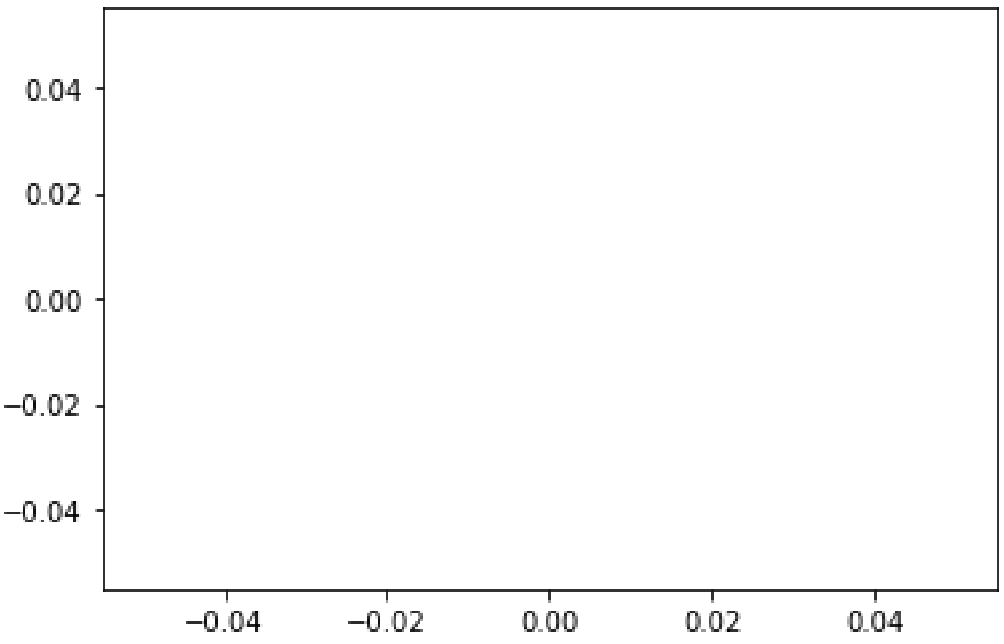
Timothy M Kaine Joint



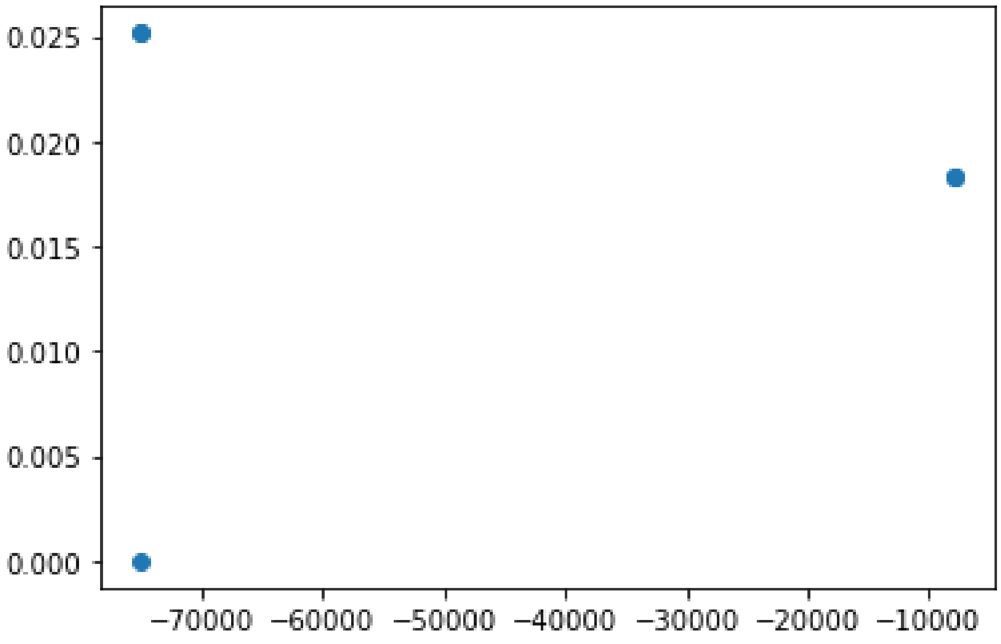
Timothy M Kaine Child



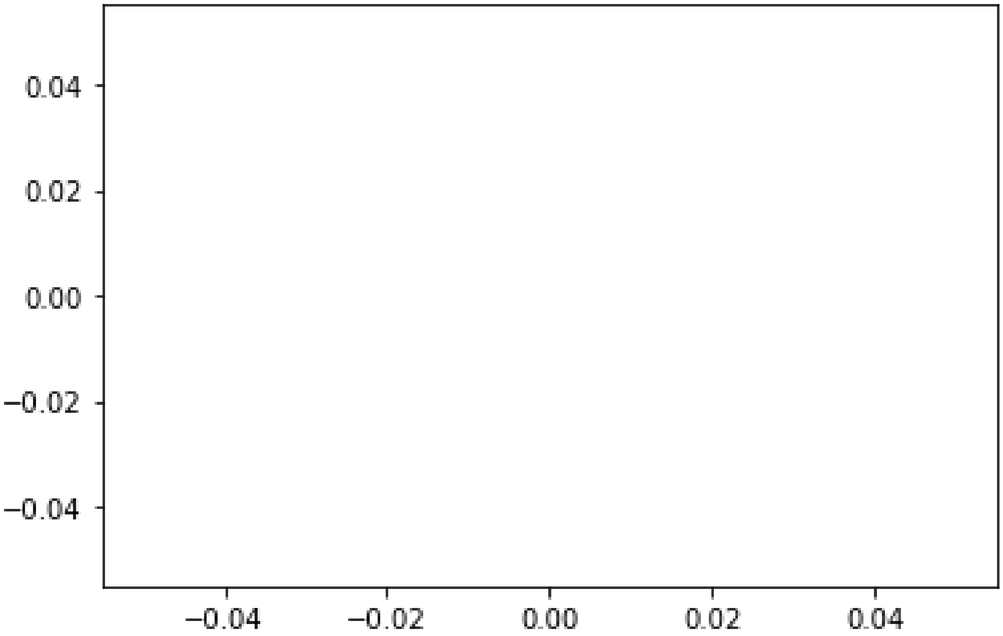
Timothy M Kaine Self



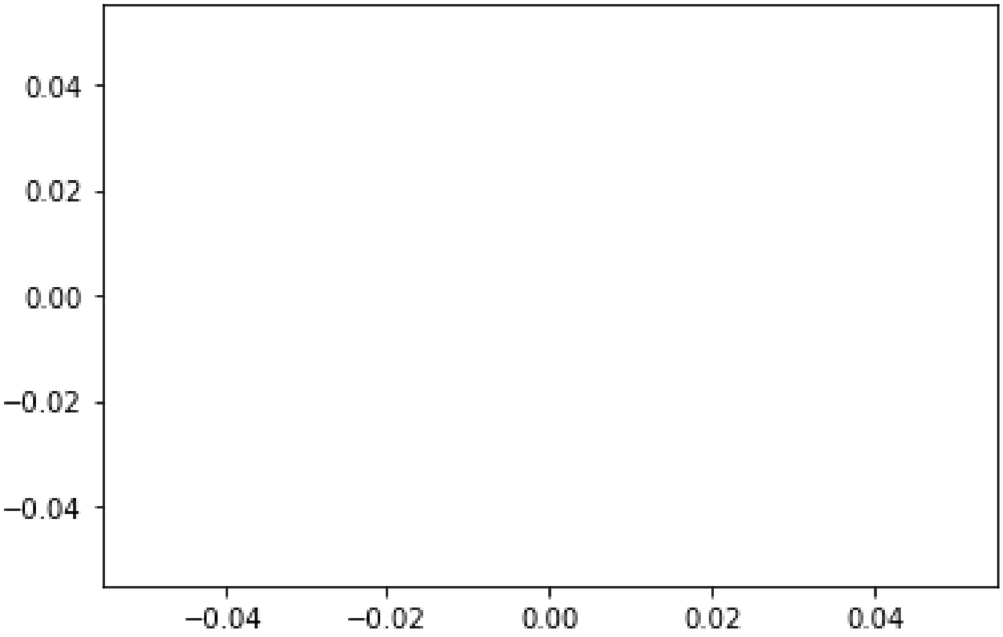
Roy Blunt Spouse



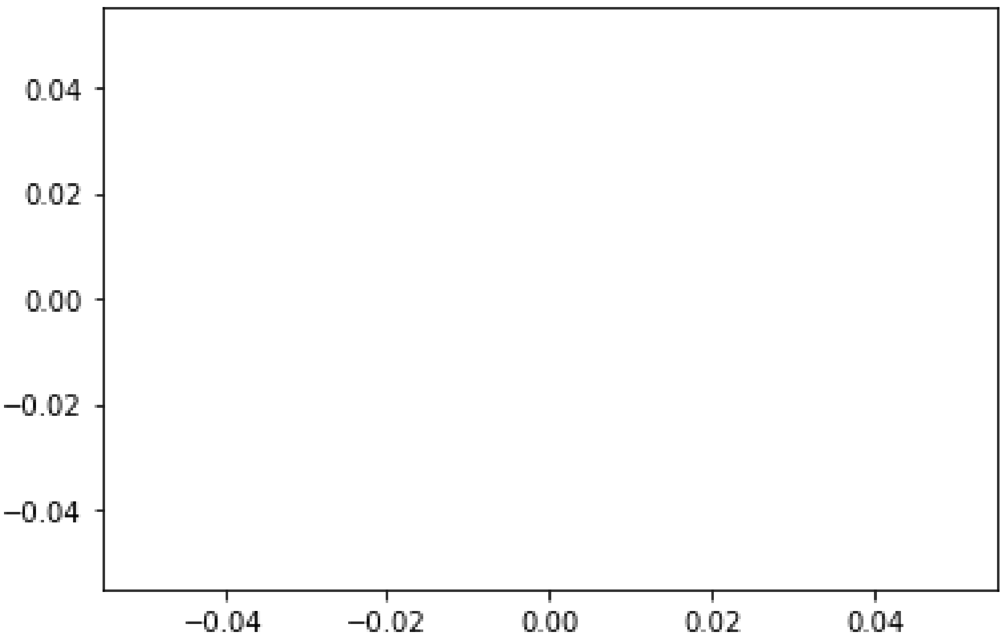
Roy Blunt Joint



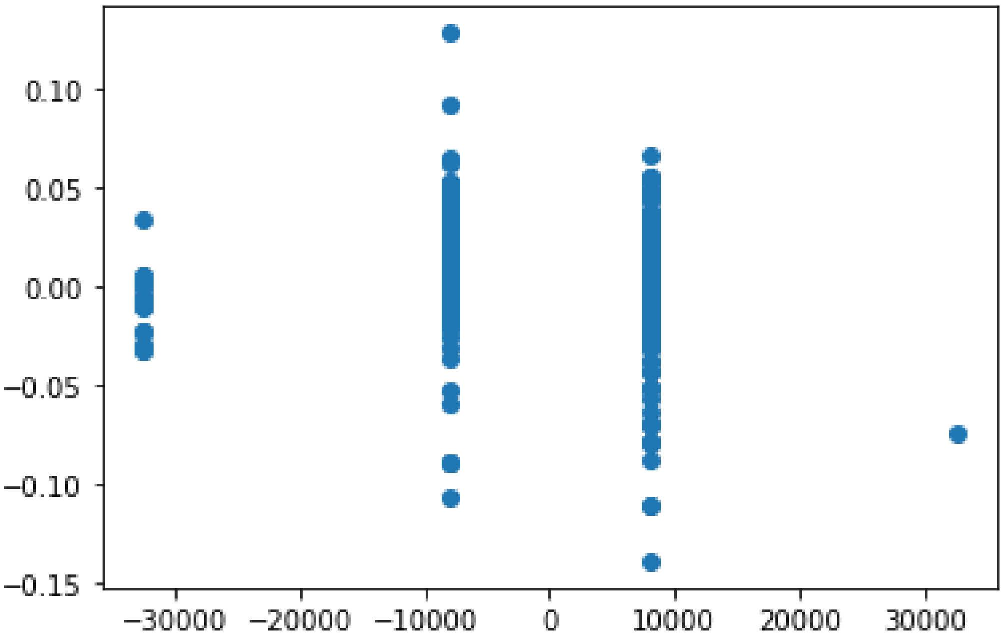
Roy Blunt Child



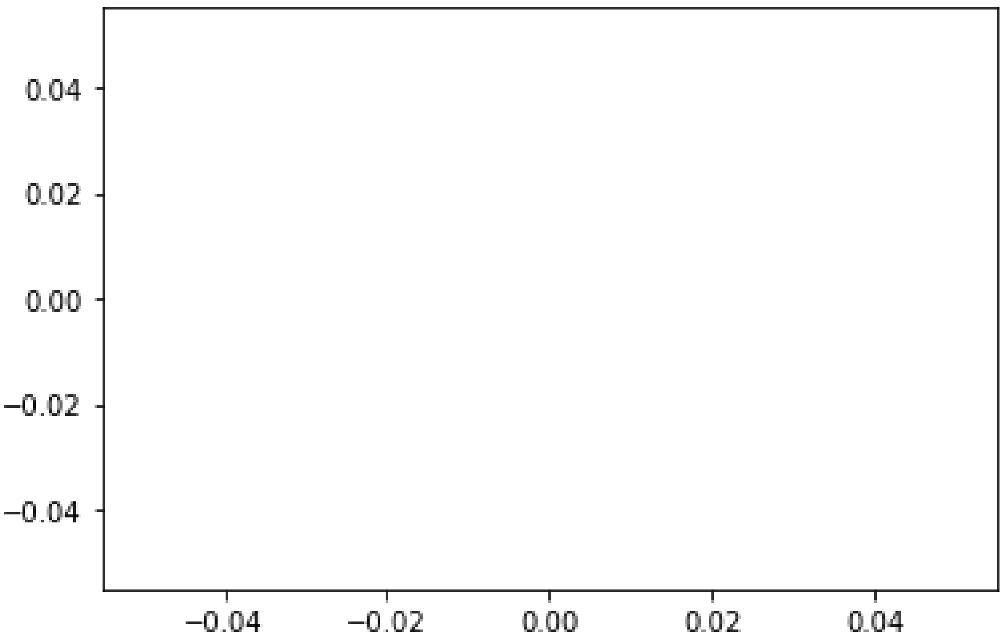
Roy Blunt Self



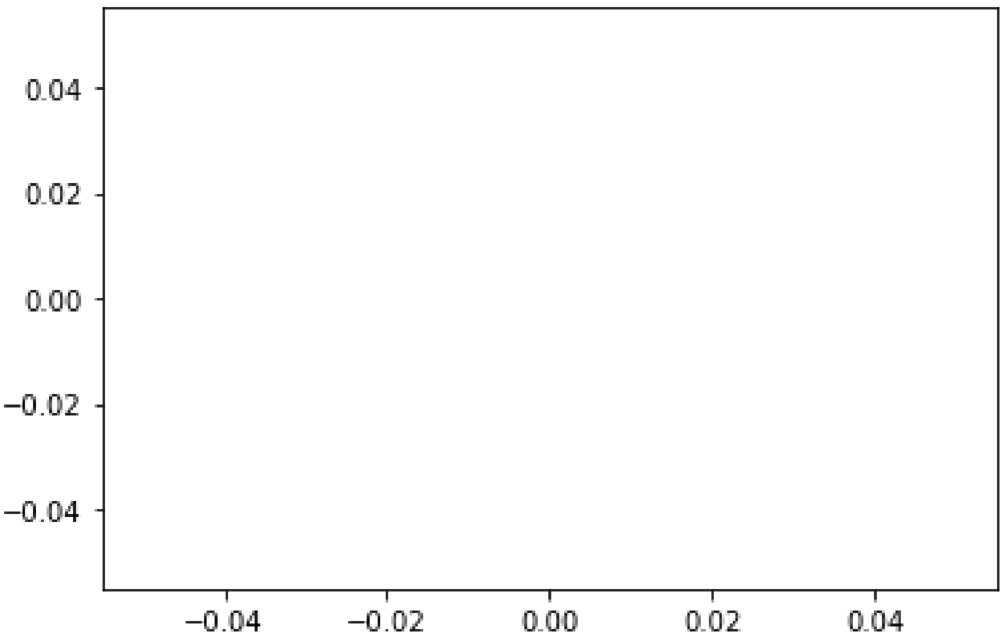
Shelley M Capito Spouse



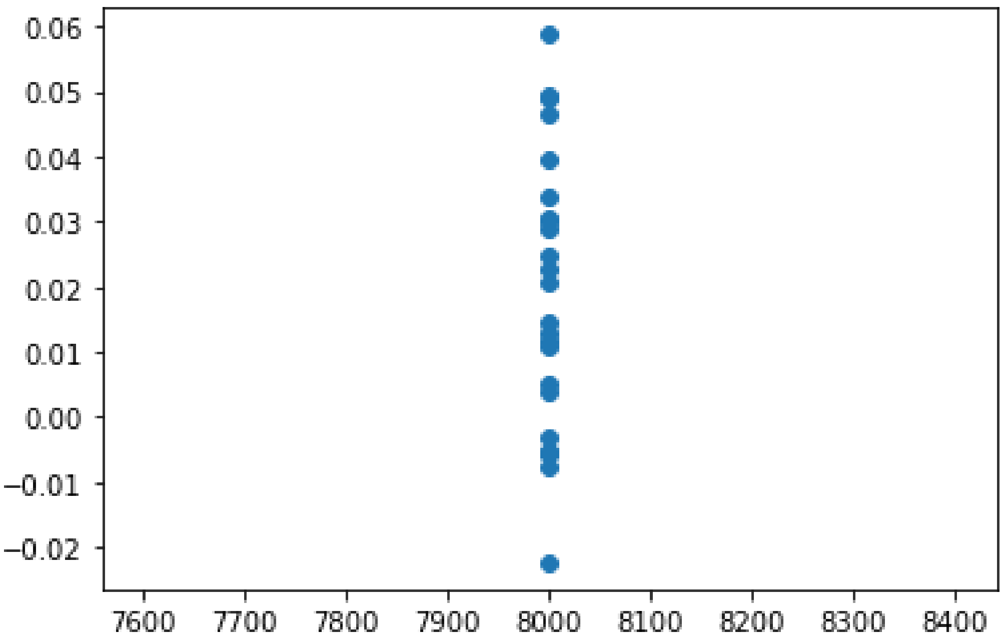
Shelley M Capito Joint



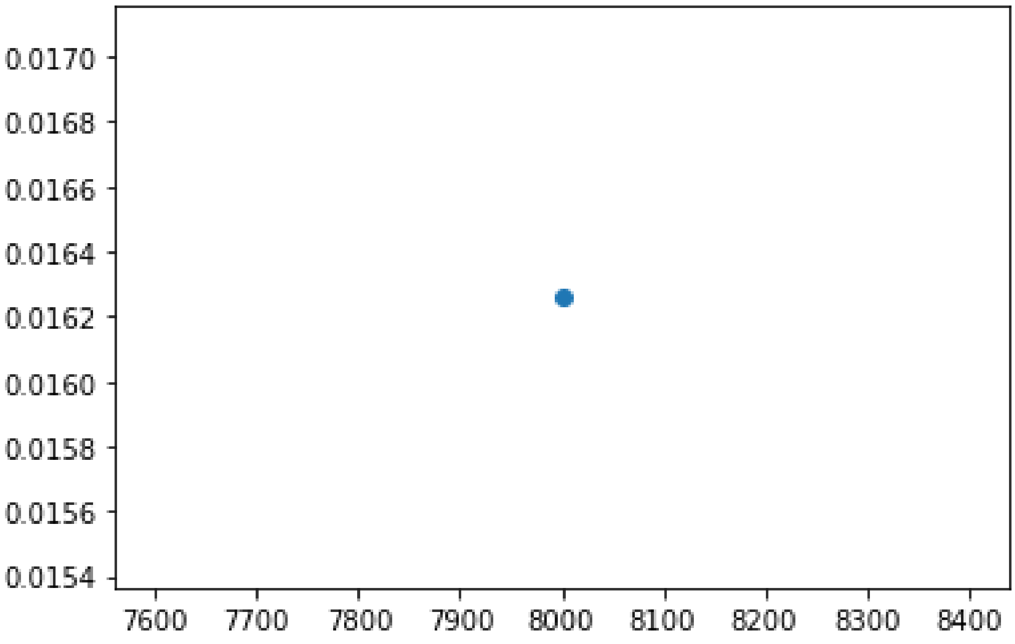
```
Shelley M Capito Child
```



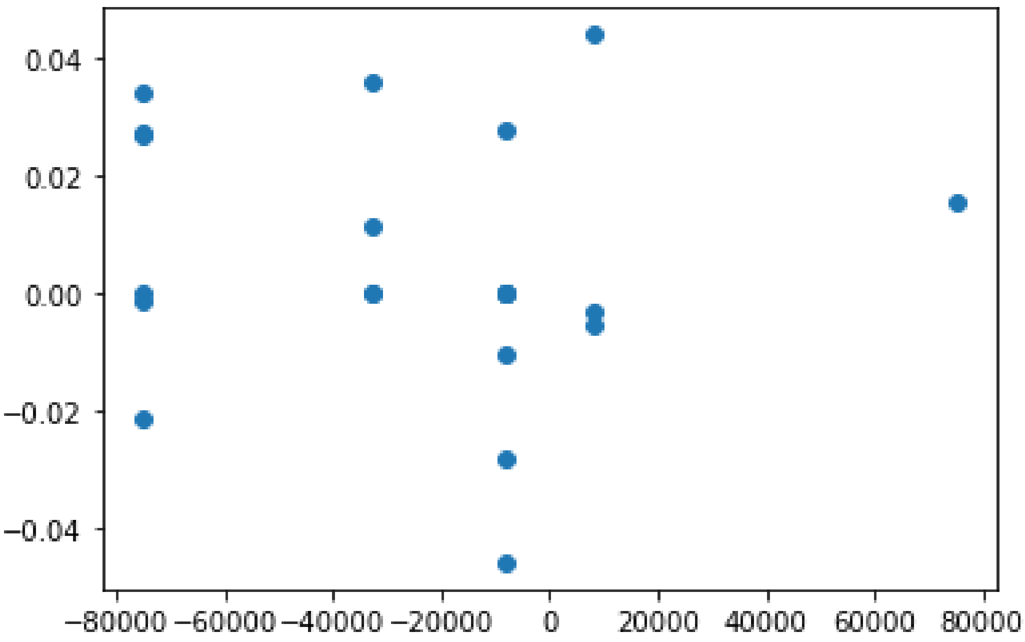
```
Shelley M Capito Self
```



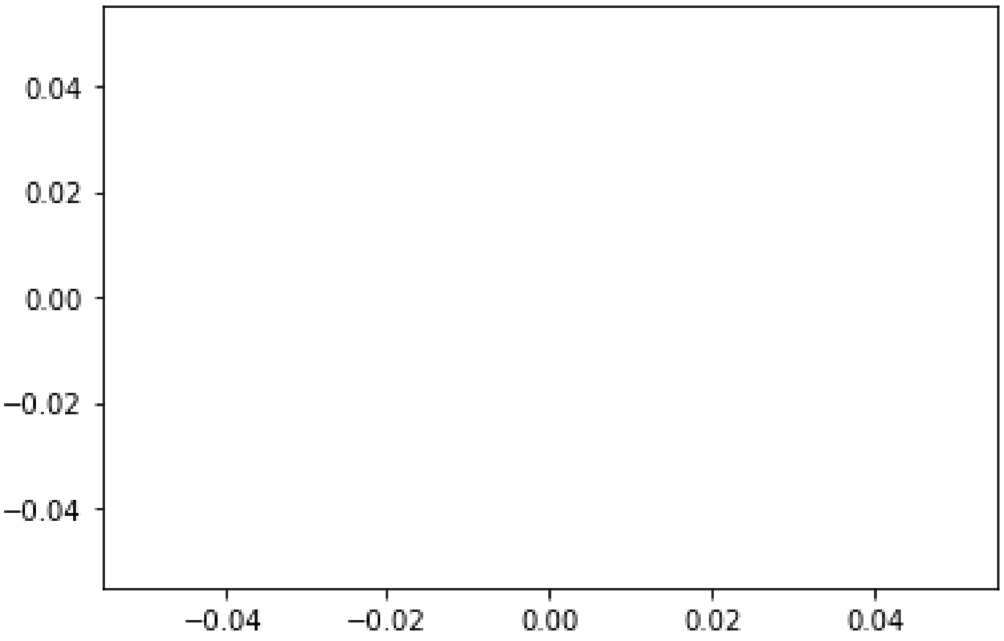
Daniel S Sullivan Spouse



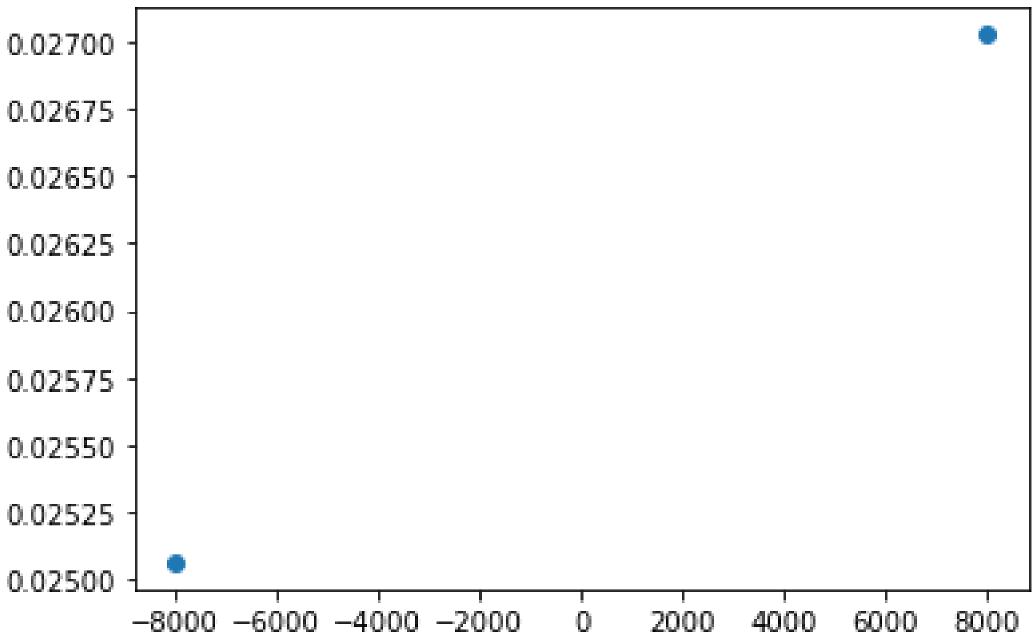
Daniel S Sullivan Joint



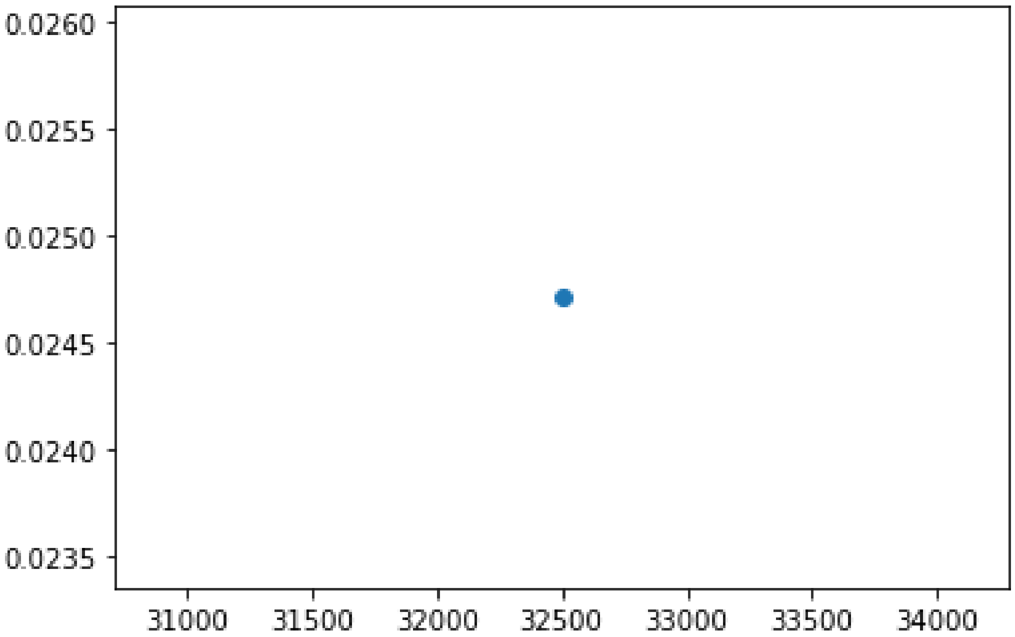
Daniel S Sullivan Child



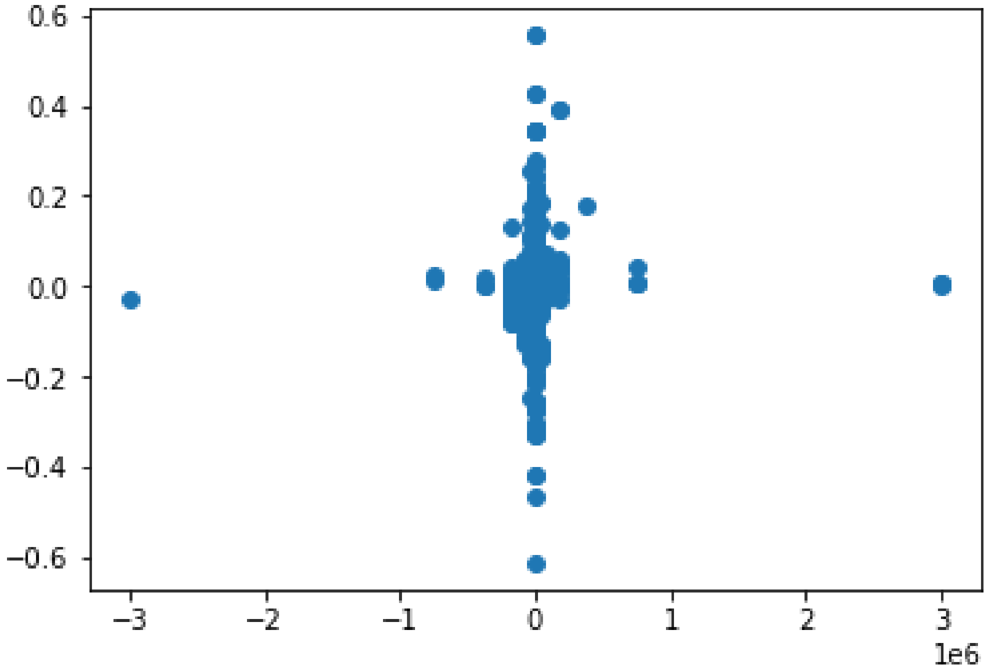
Daniel S Sullivan Self



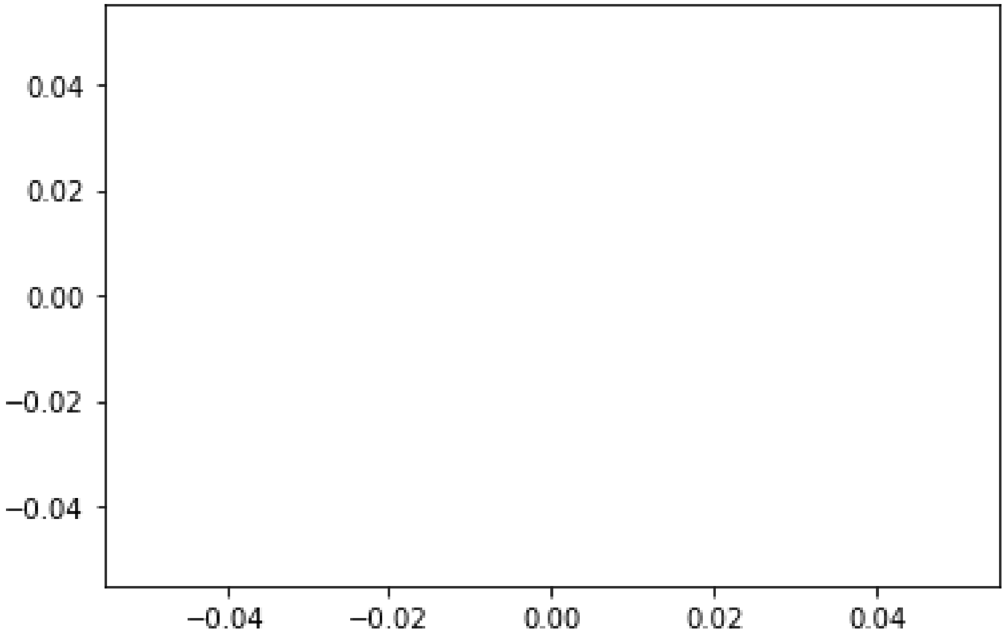
David A Perdue , Jr Spouse



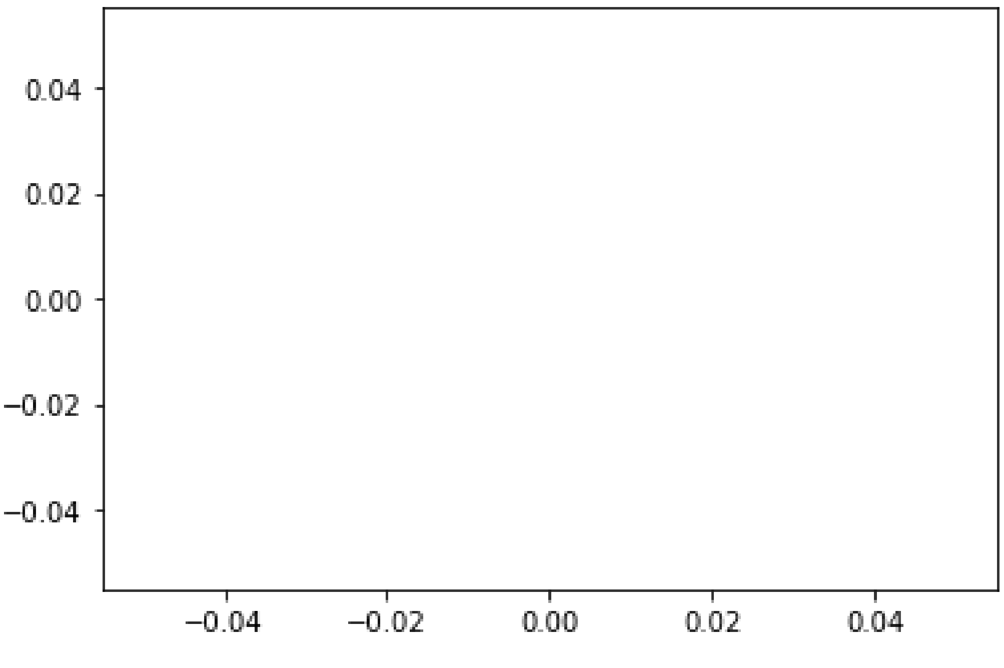
David A Perdue , Jr Joint



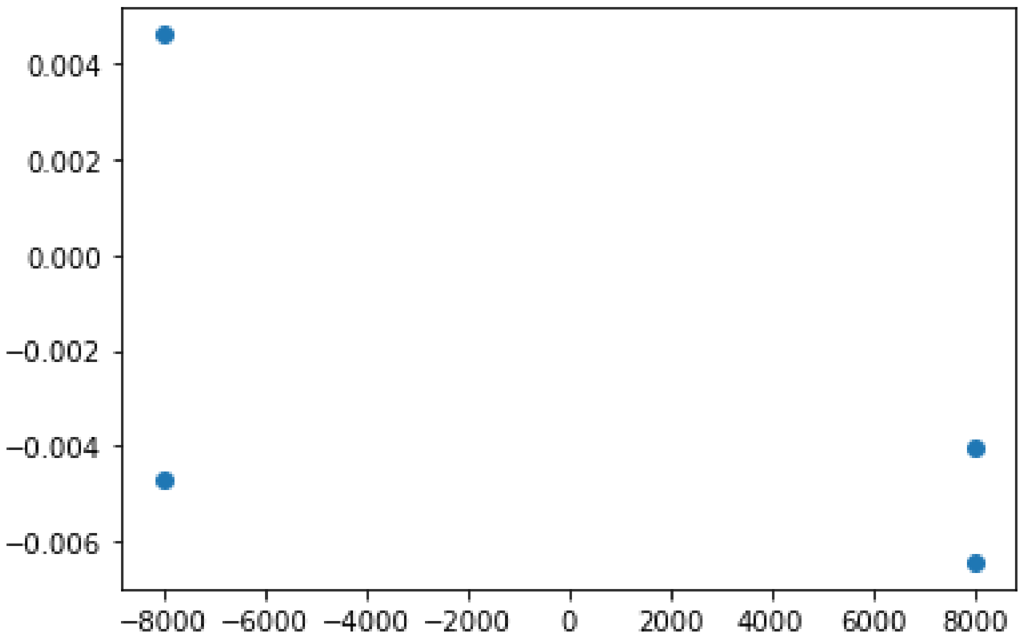
David A Perdue , Jr Child



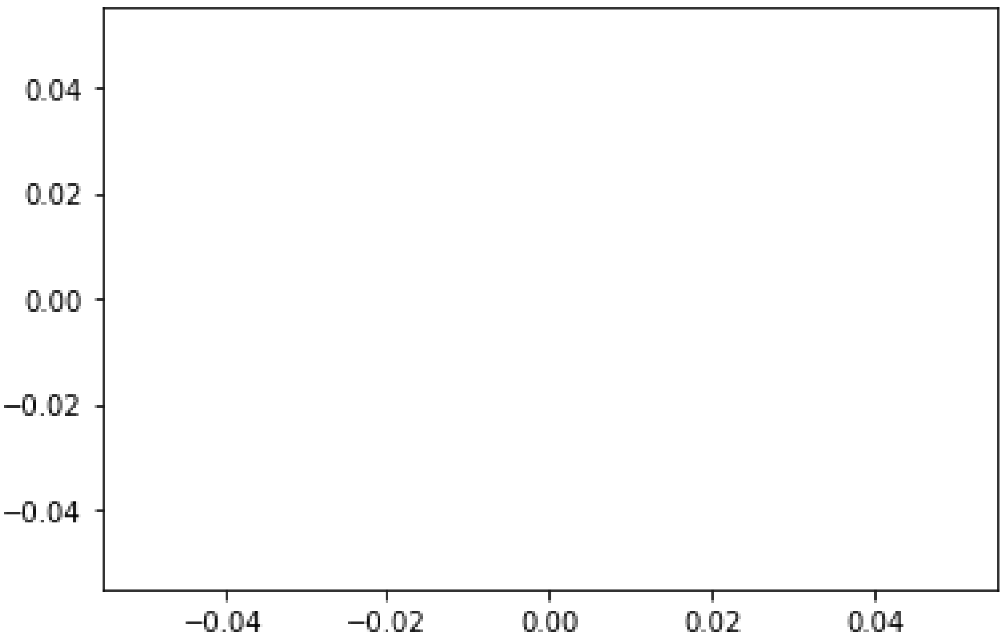
David A Perdue , Jr Self



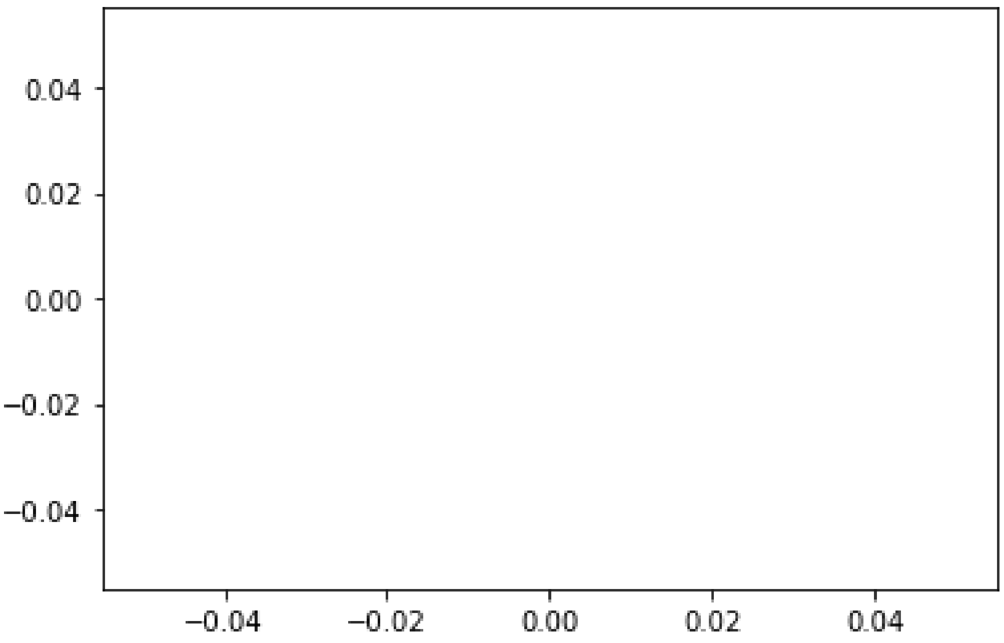
Roger F Wicker Spouse



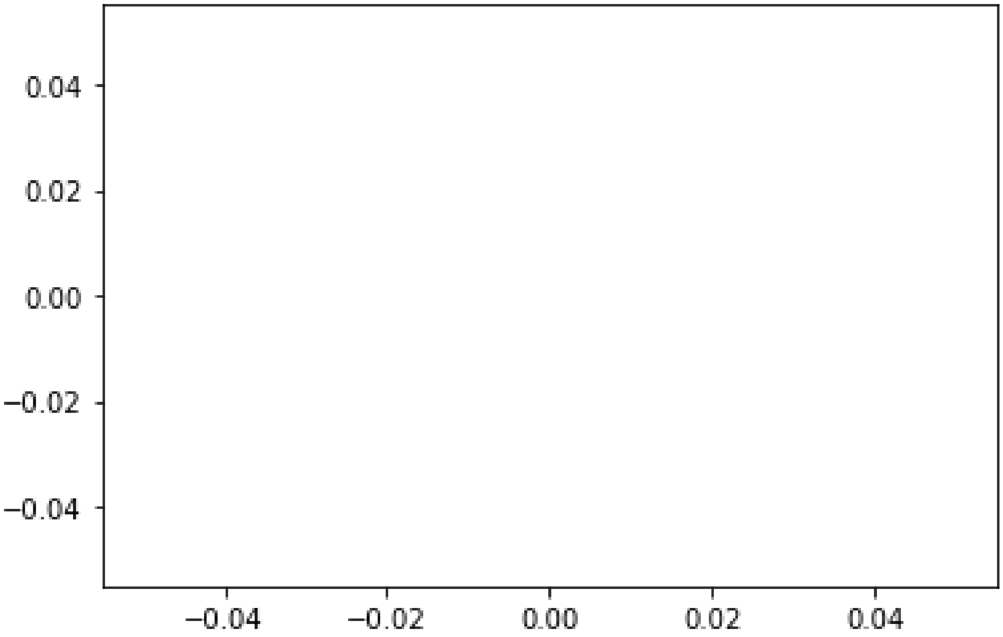
Roger F Wicker Joint



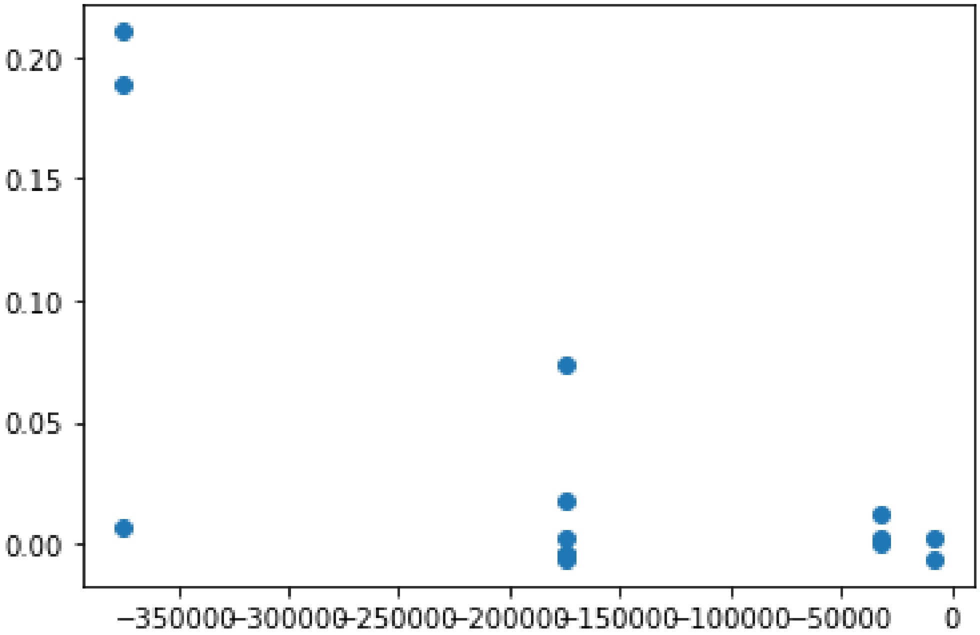
Roger F Wicker Child



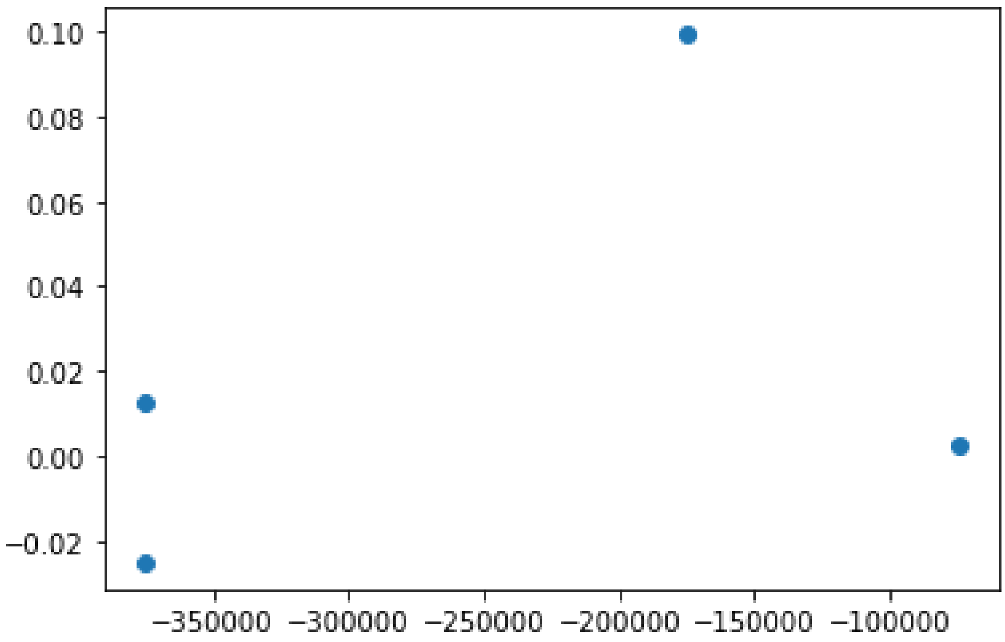
Roger F Wicker Self



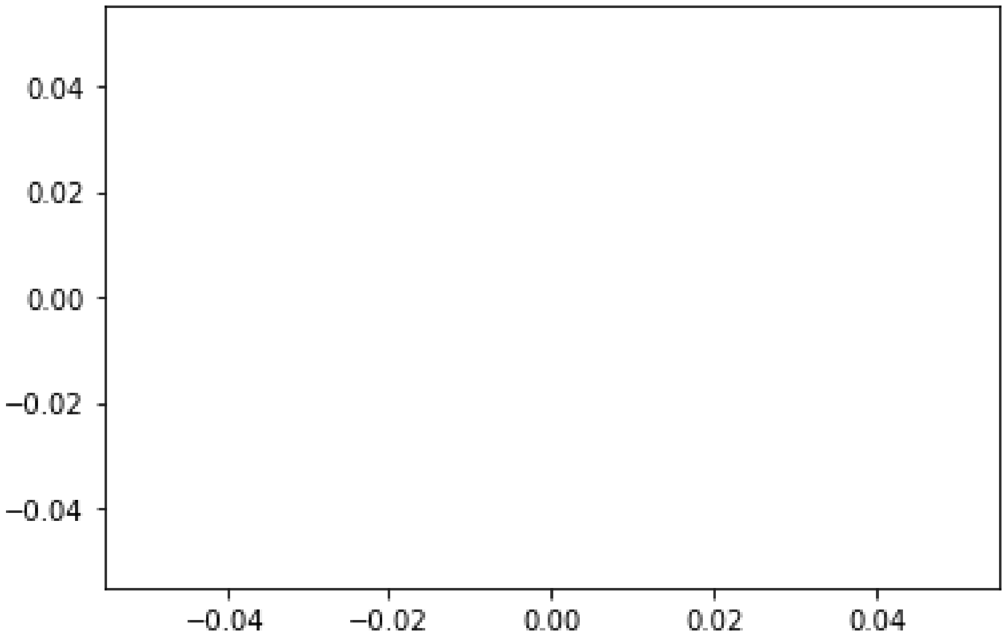
Tina Smith Spouse



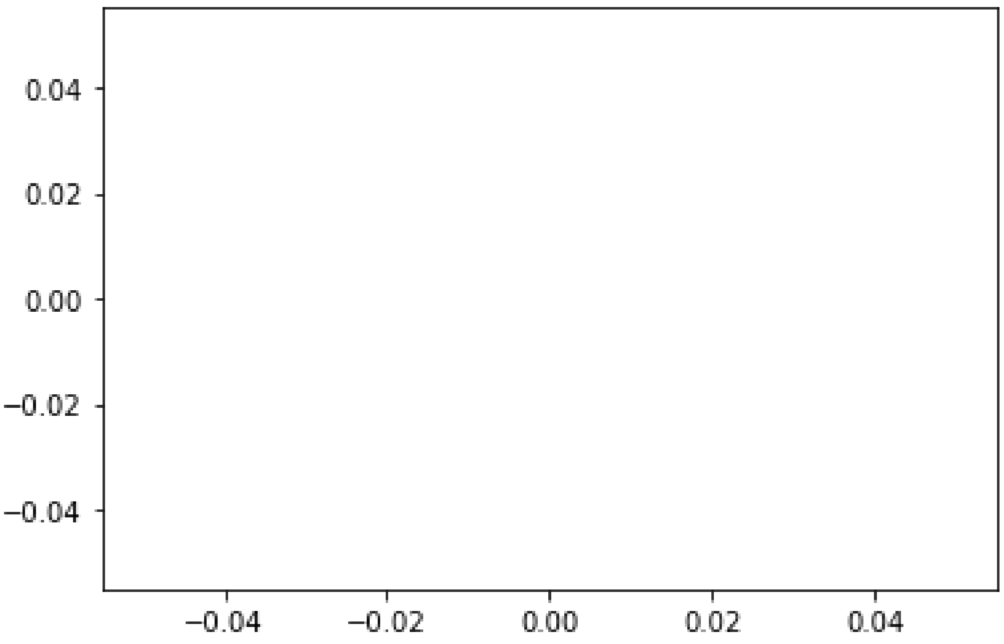
Tina Smith Joint



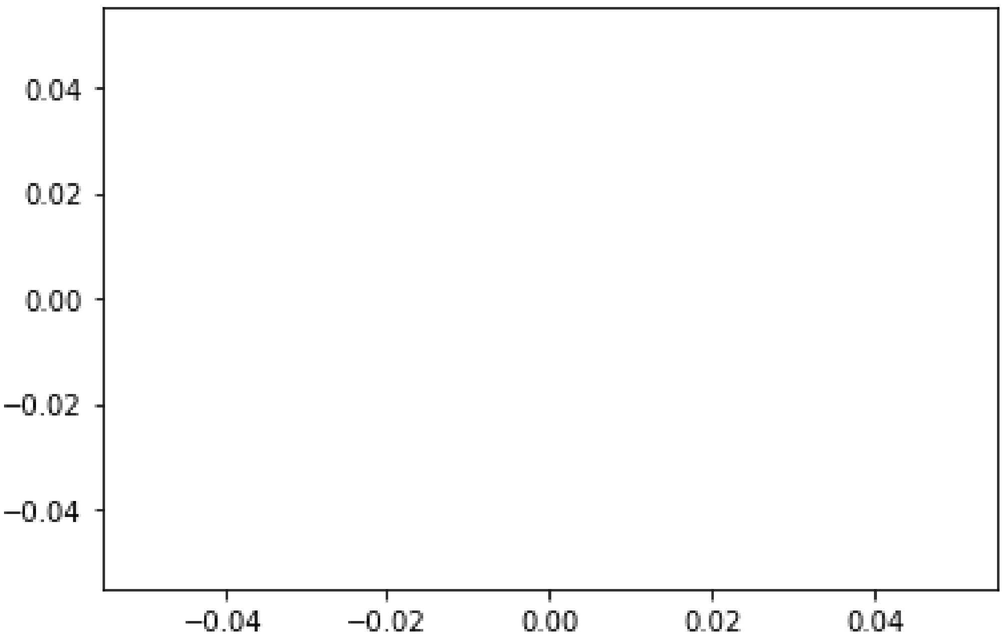
Tina Smith Child



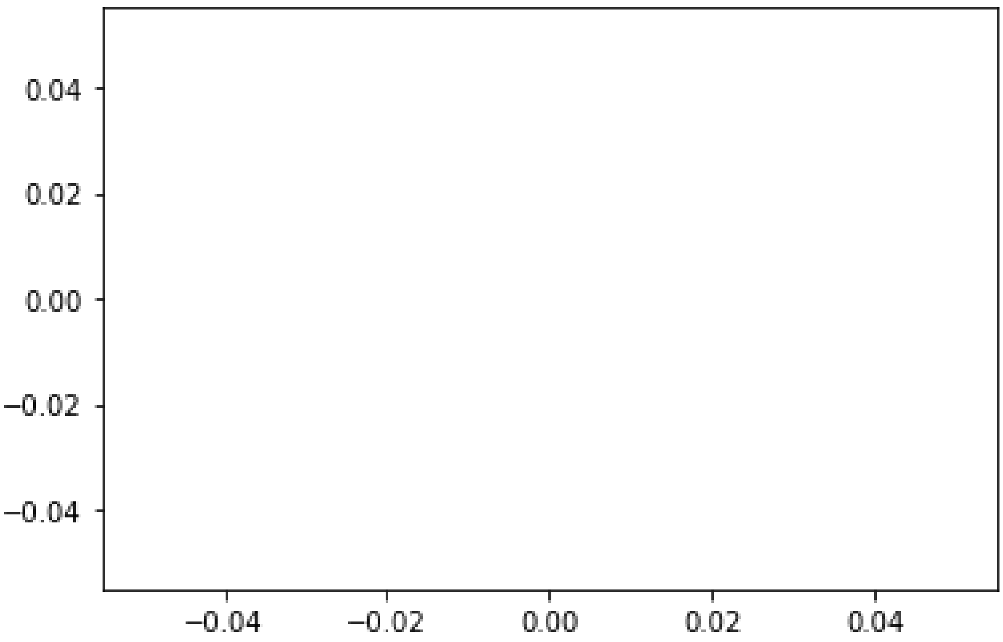
Tina Smith Self



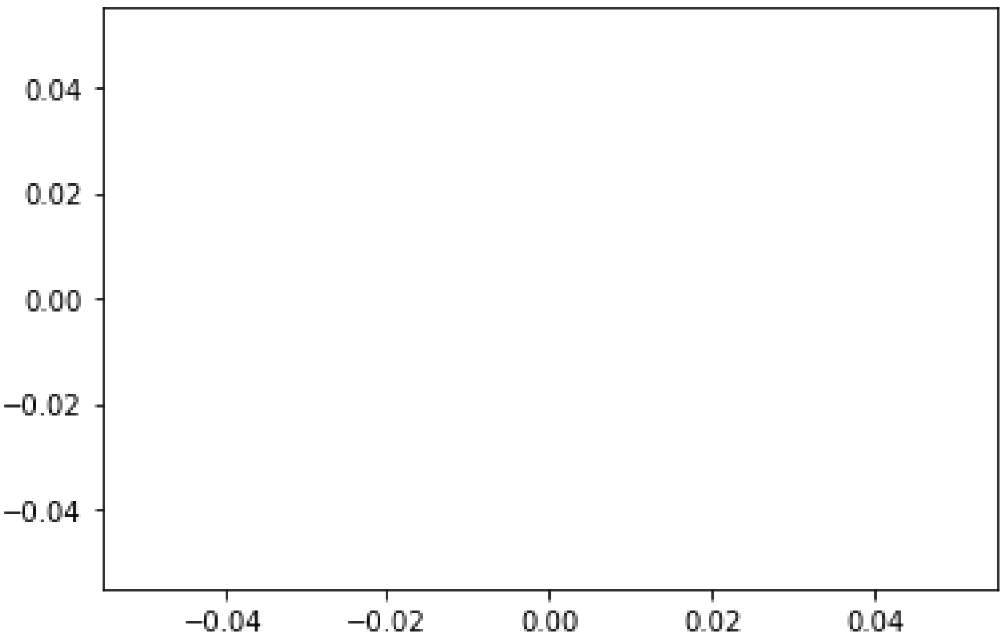
John Hoeven Spouse



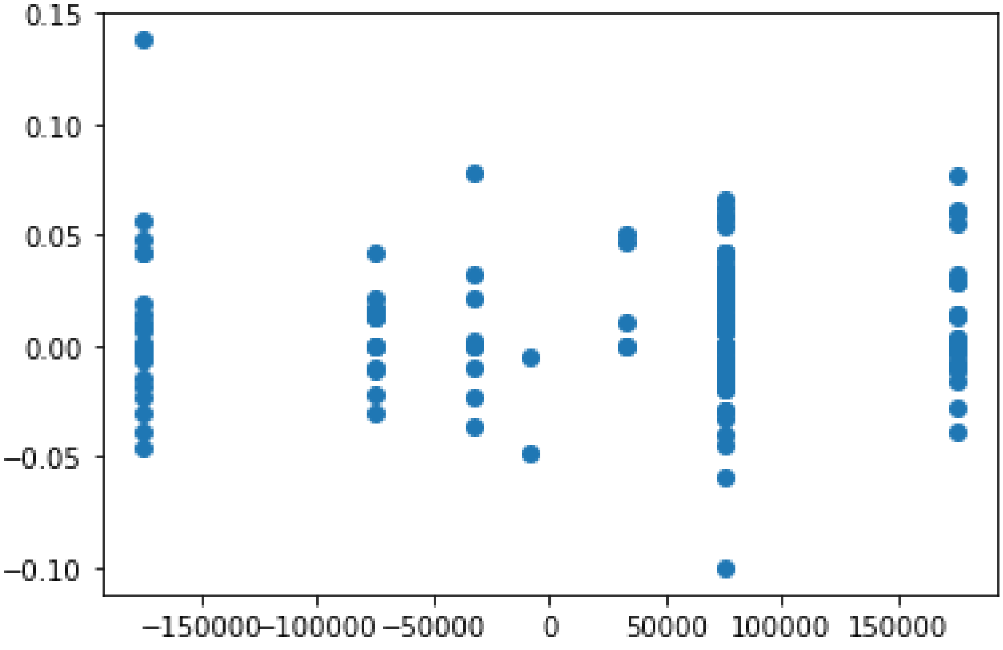
John Hoeven Joint



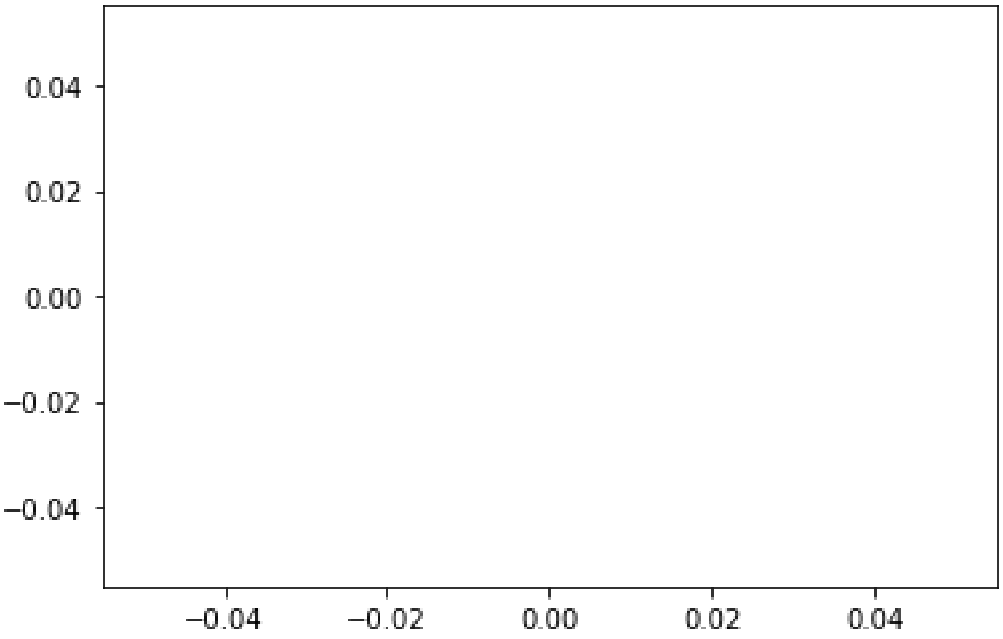
`John Hoeven Child`



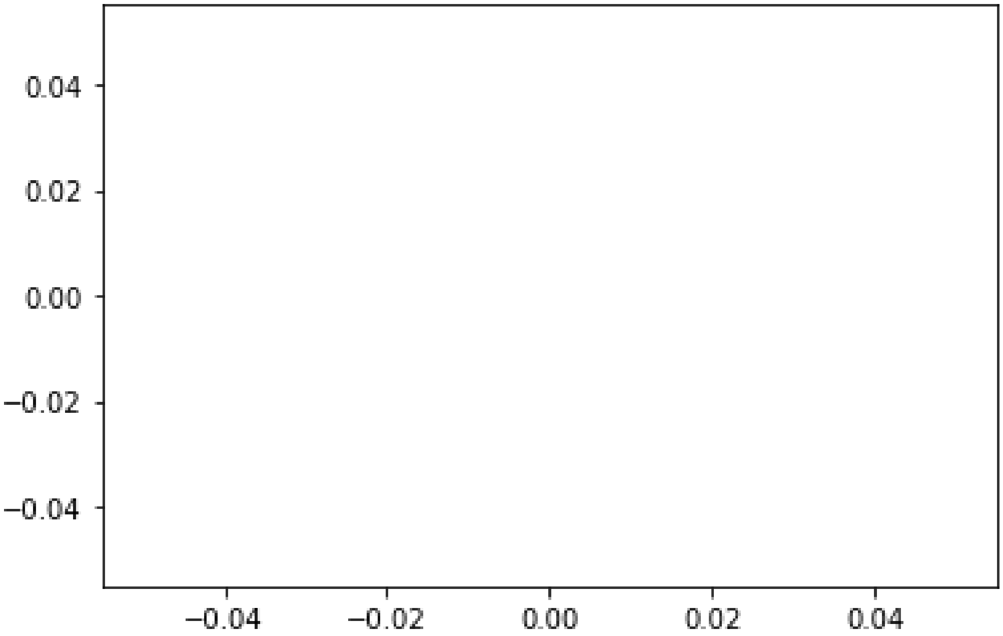
`John Hoeven Self`



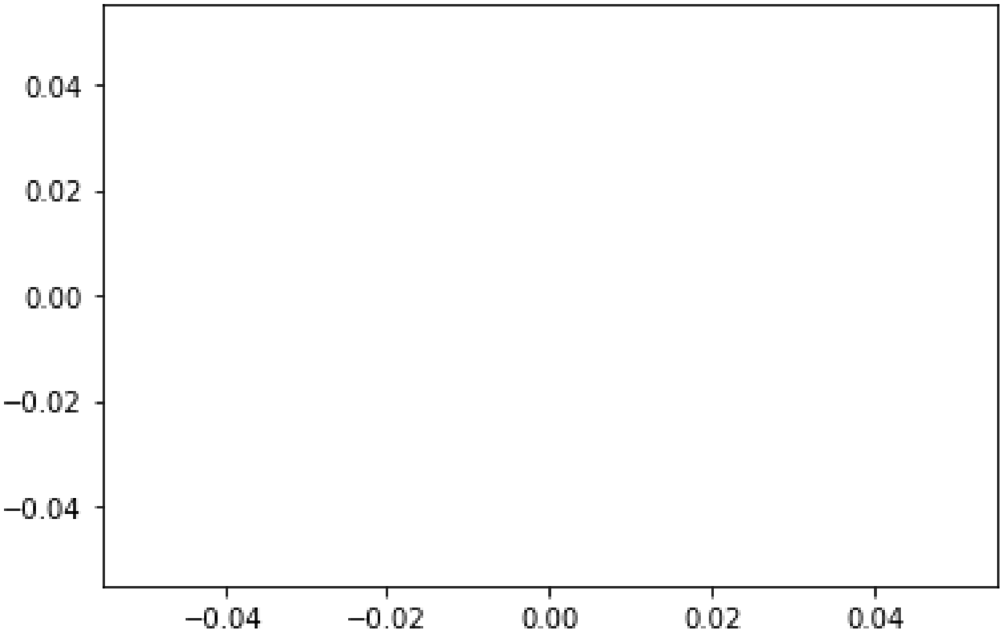
John N Kennedy Spouse



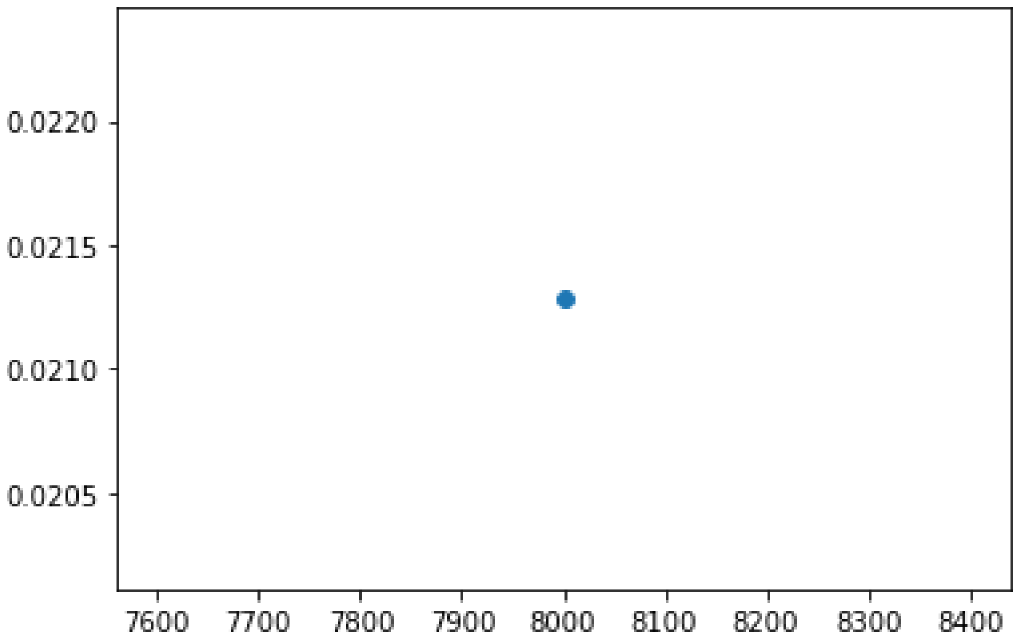
John N Kennedy Joint



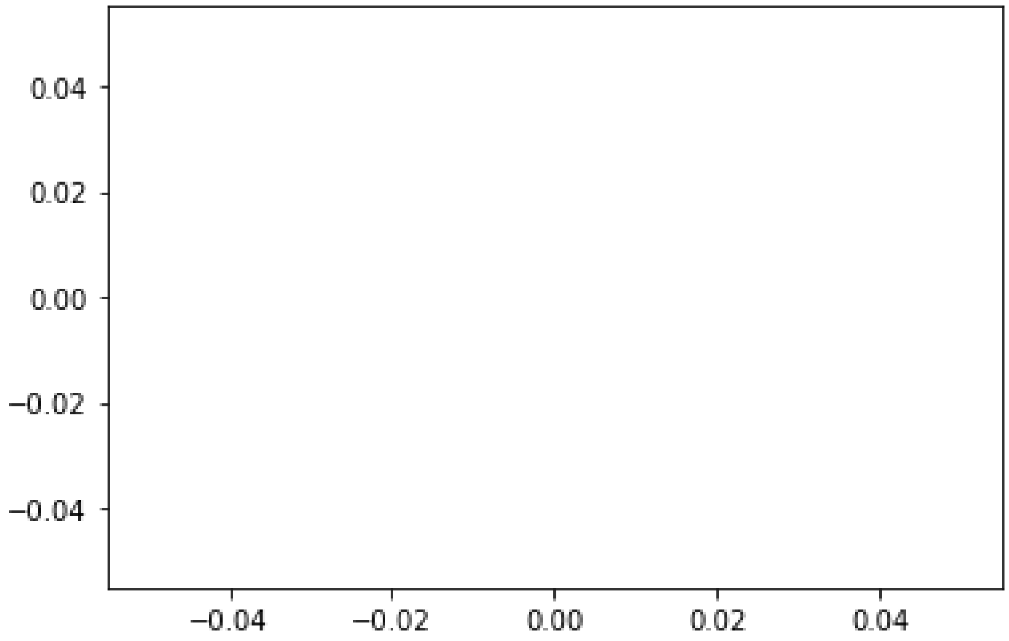
```
John N Kennedy Child
```



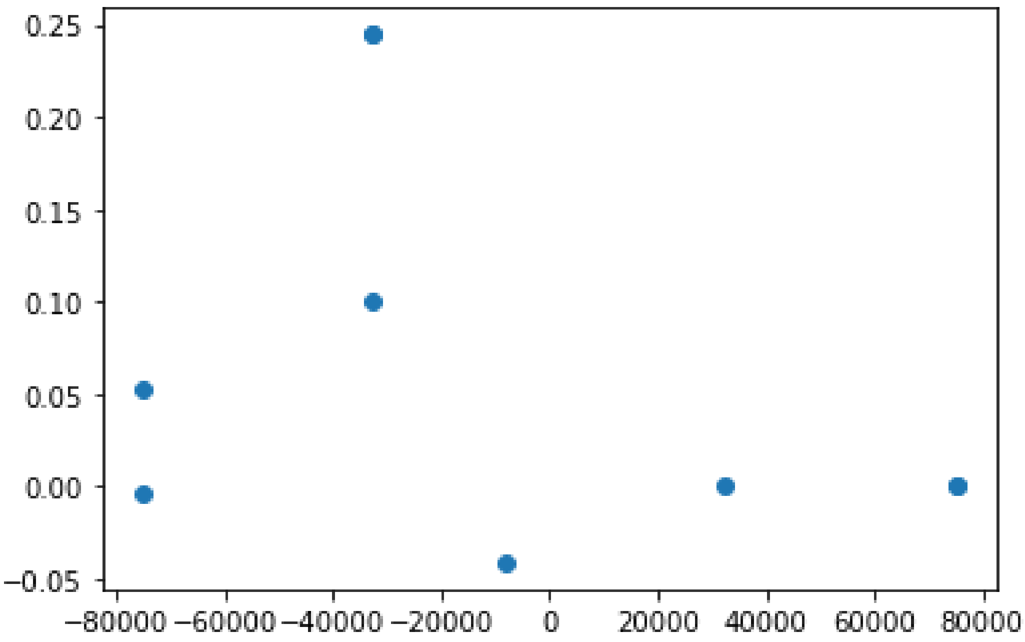
```
John N Kennedy Self
```



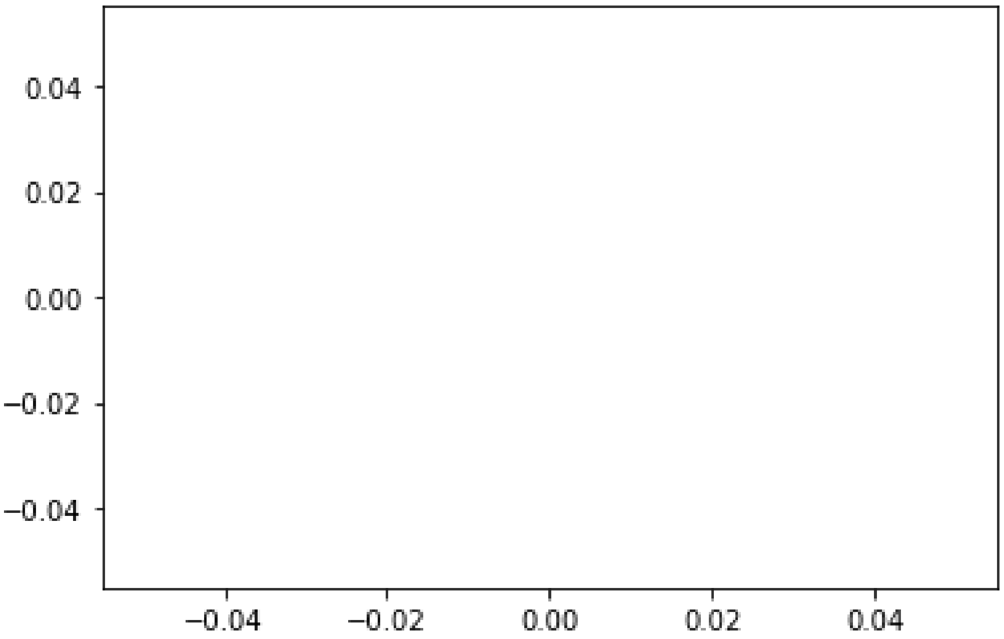
```
Rafael E Cruz Spouse
```



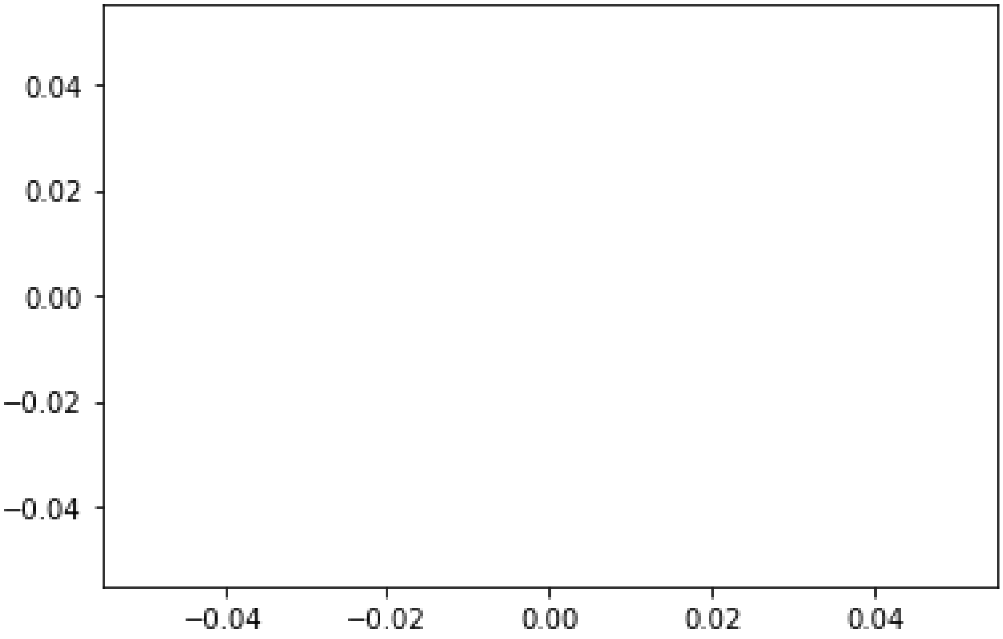
```
Rafael E Cruz Joint
```



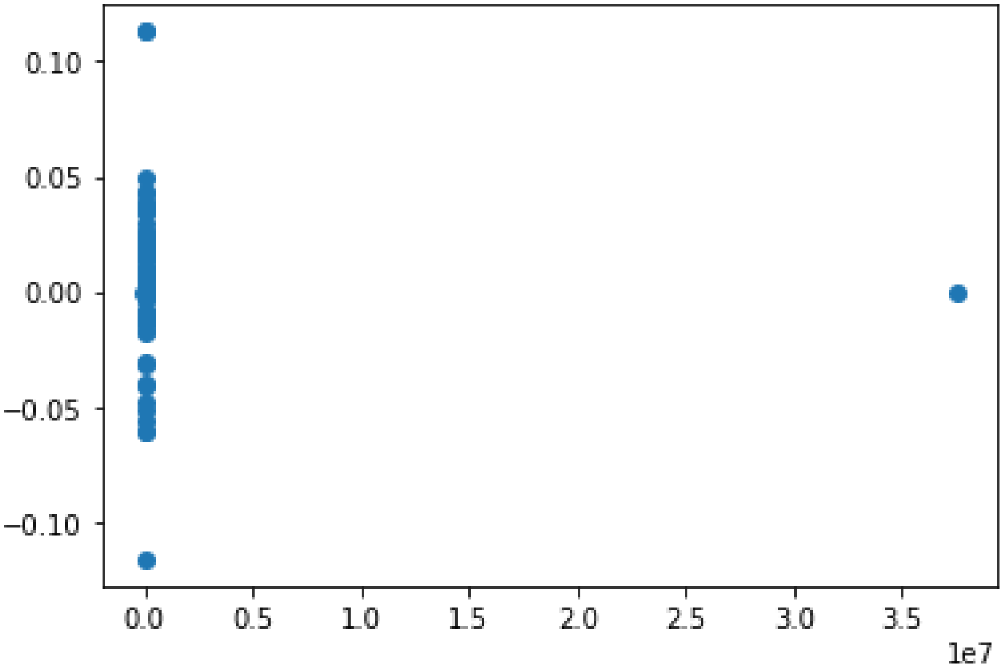
Rafael E Cruz Child



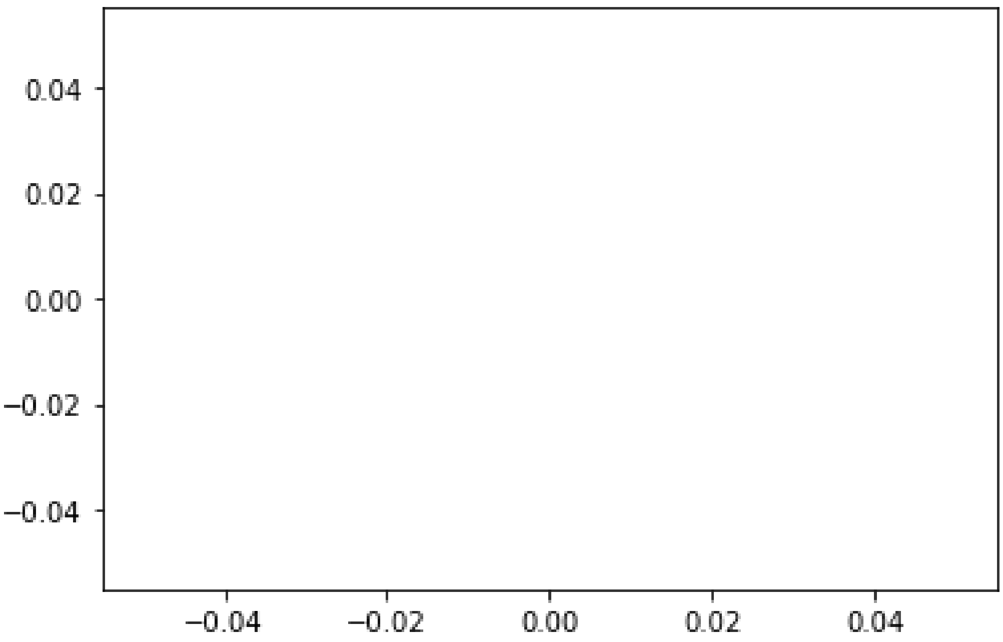
Rafael E Cruz Self



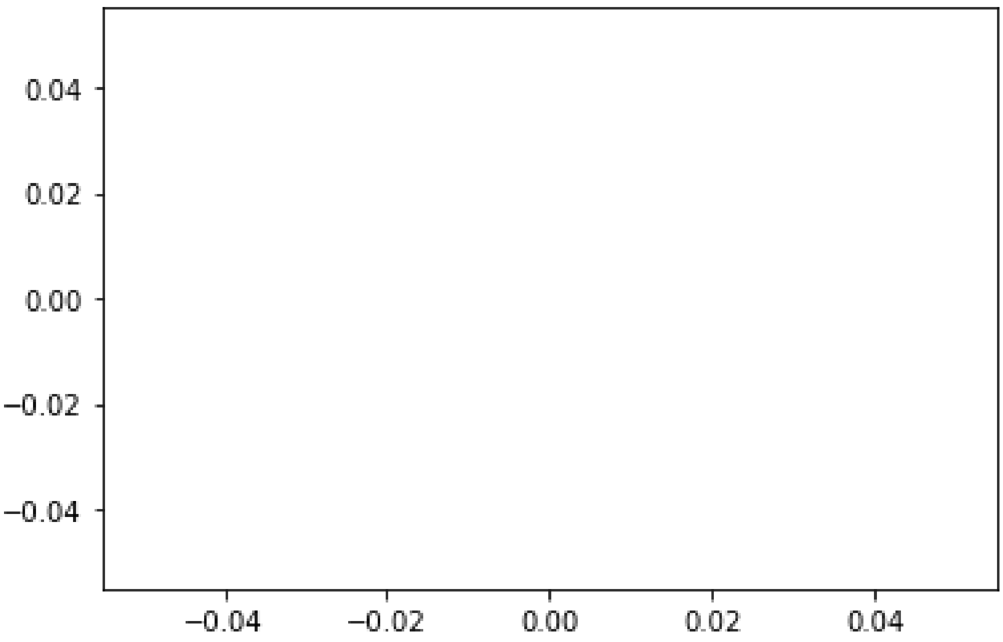
Susan M Collins Spouse



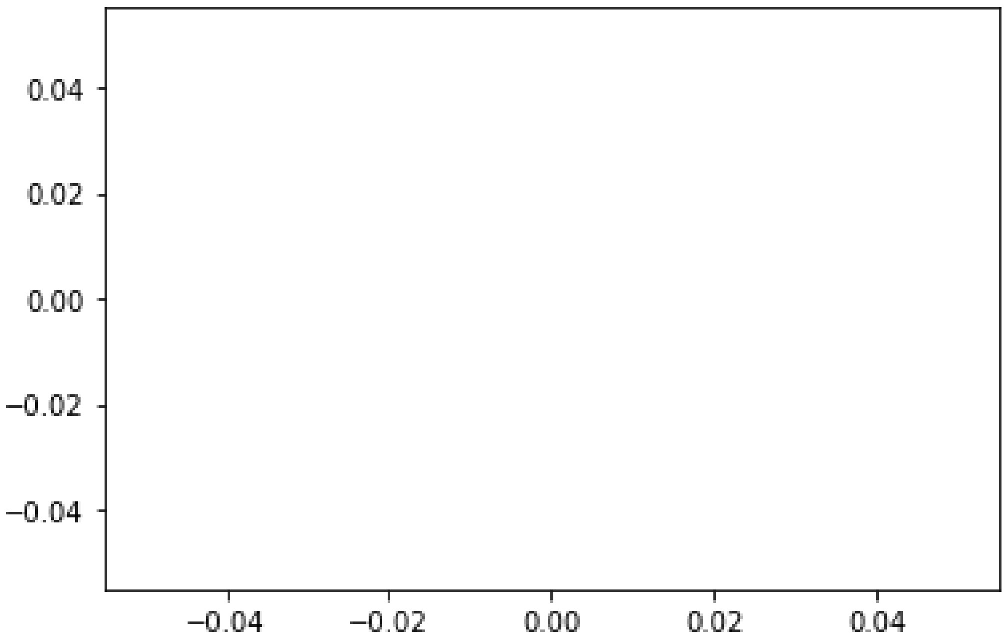
Susan M Collins Joint



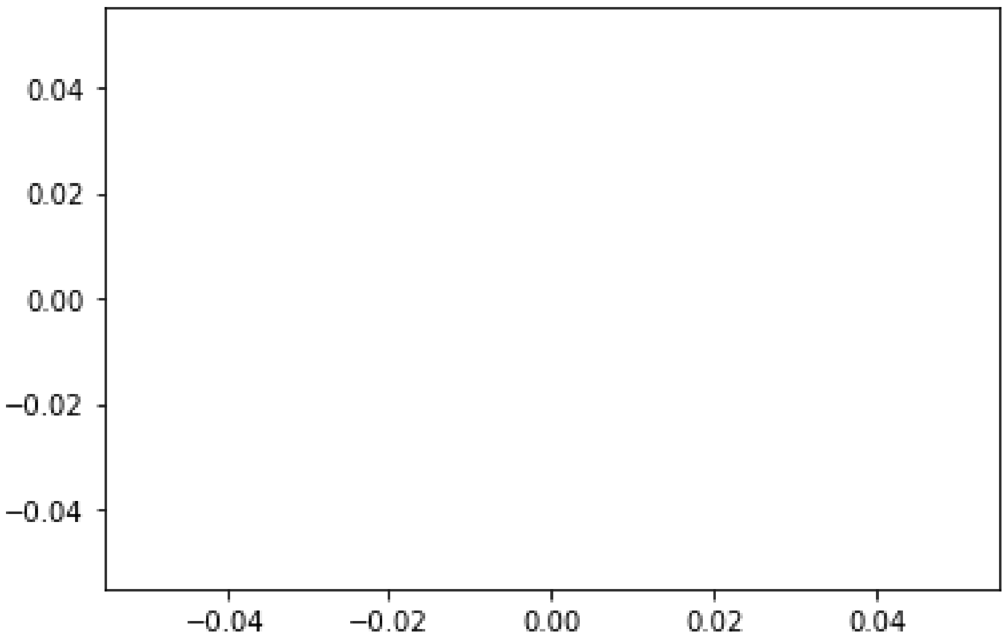
```
Susan M Collins Child
```



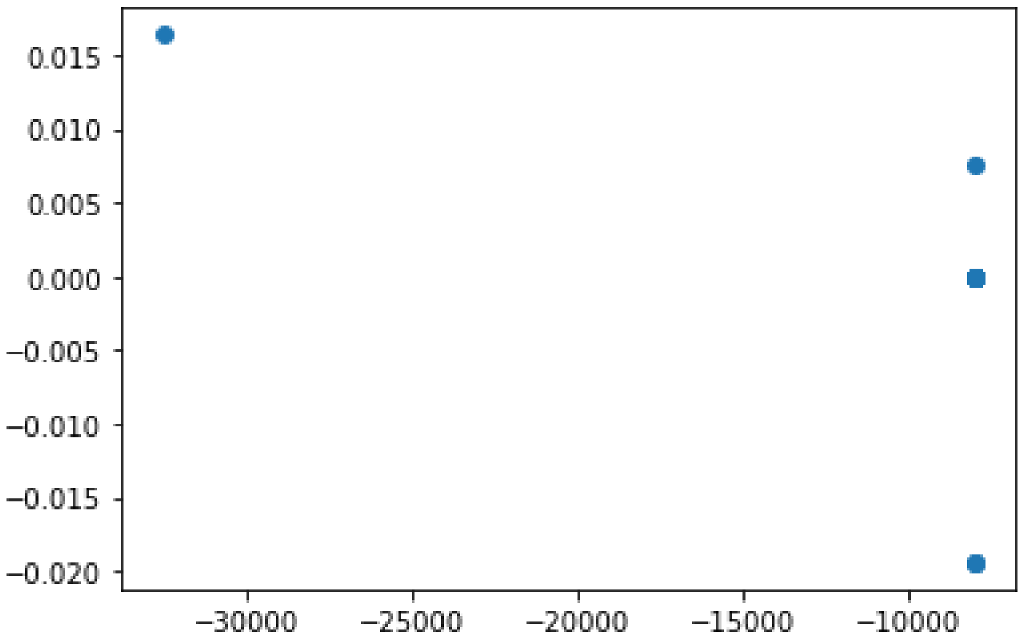
```
Susan M Collins Self
```



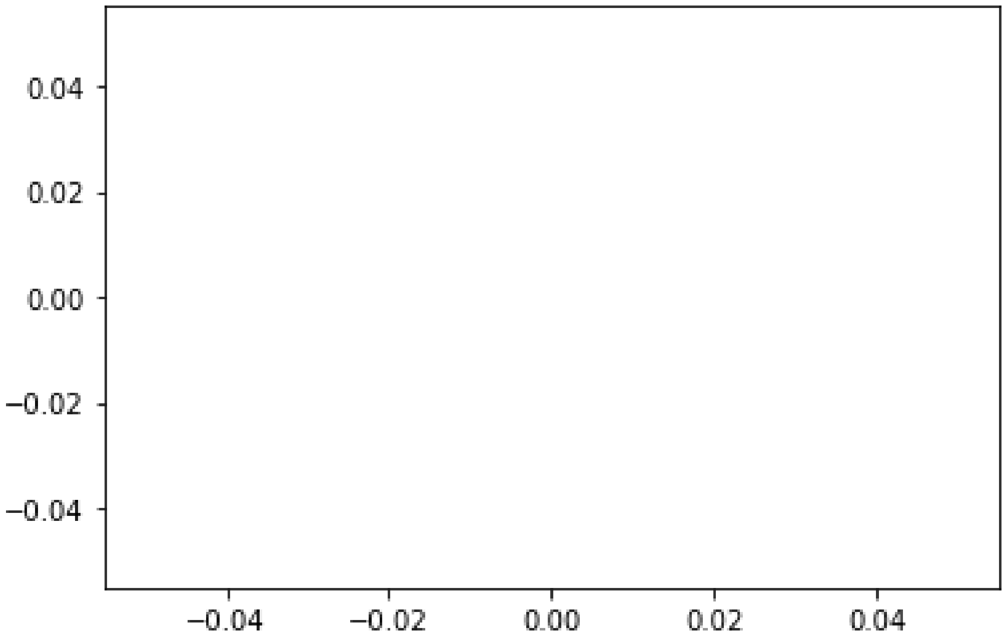
```
Thomas R Tillis Spouse
```



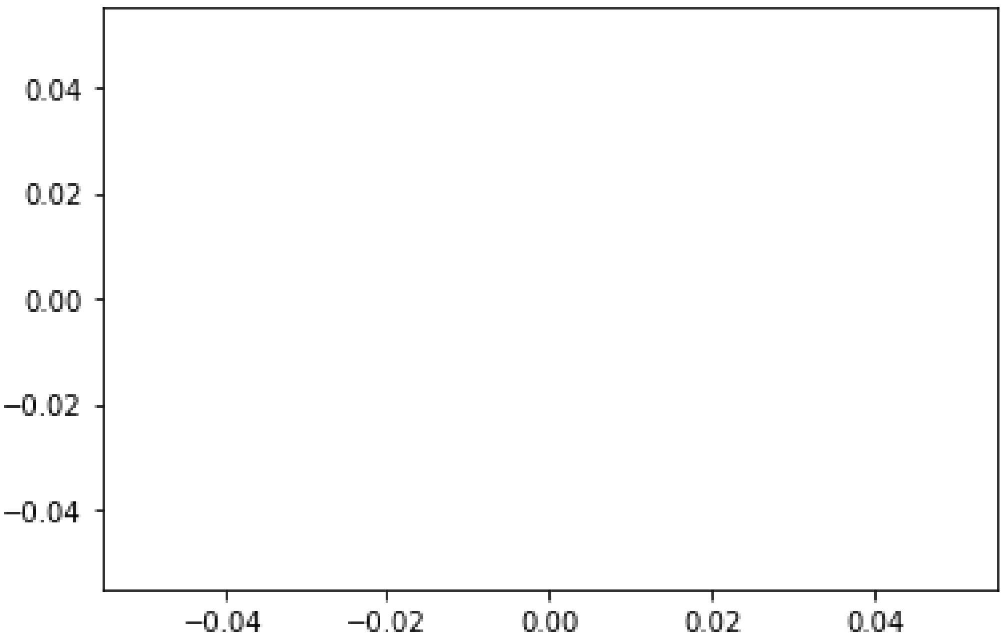
```
Thomas R Tillis Joint
```



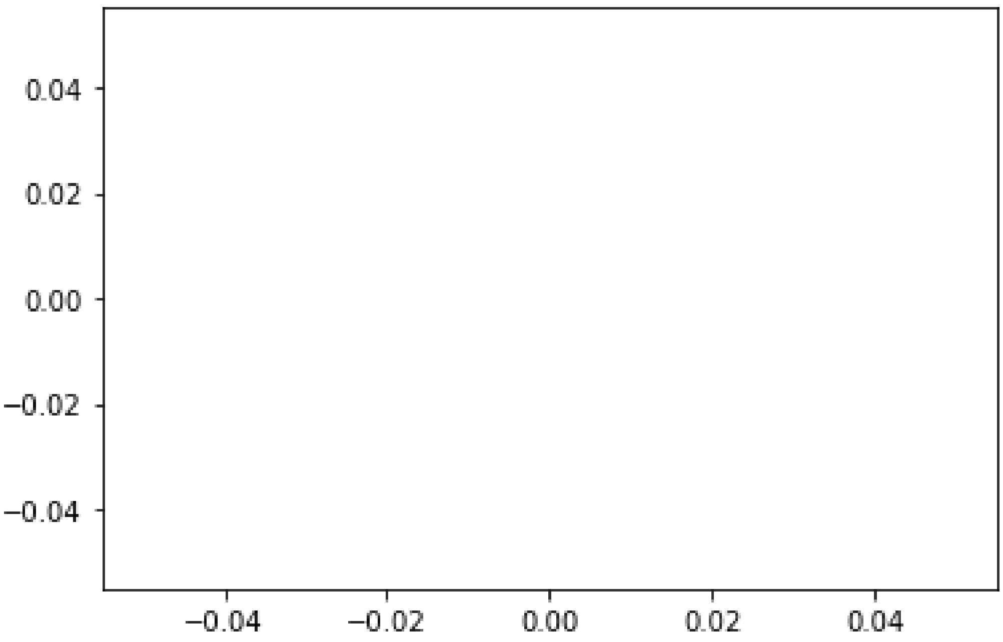
Thomas R Tillis Child



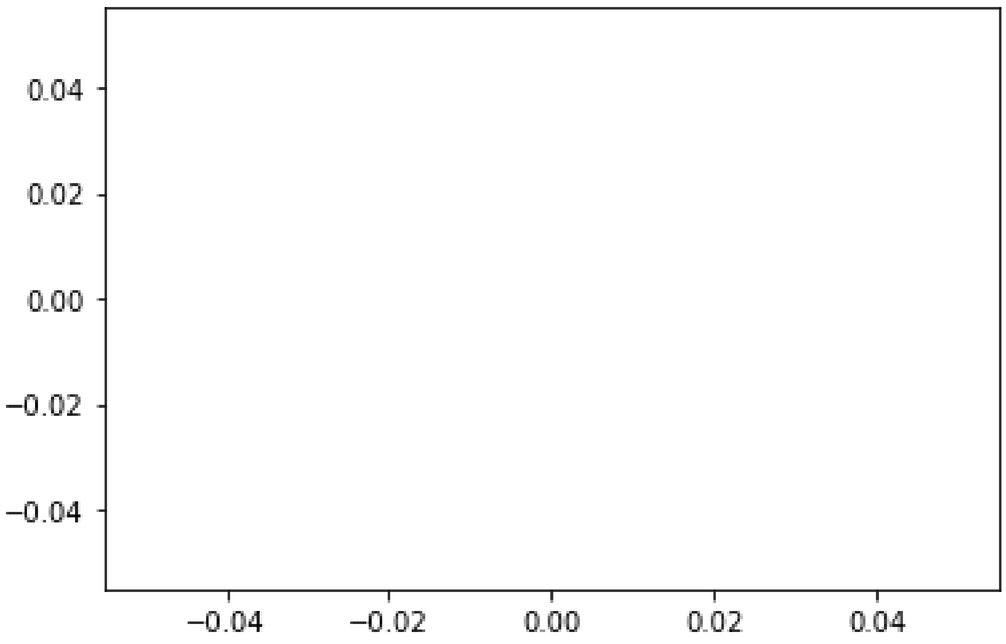
Thomas R Tillis Self



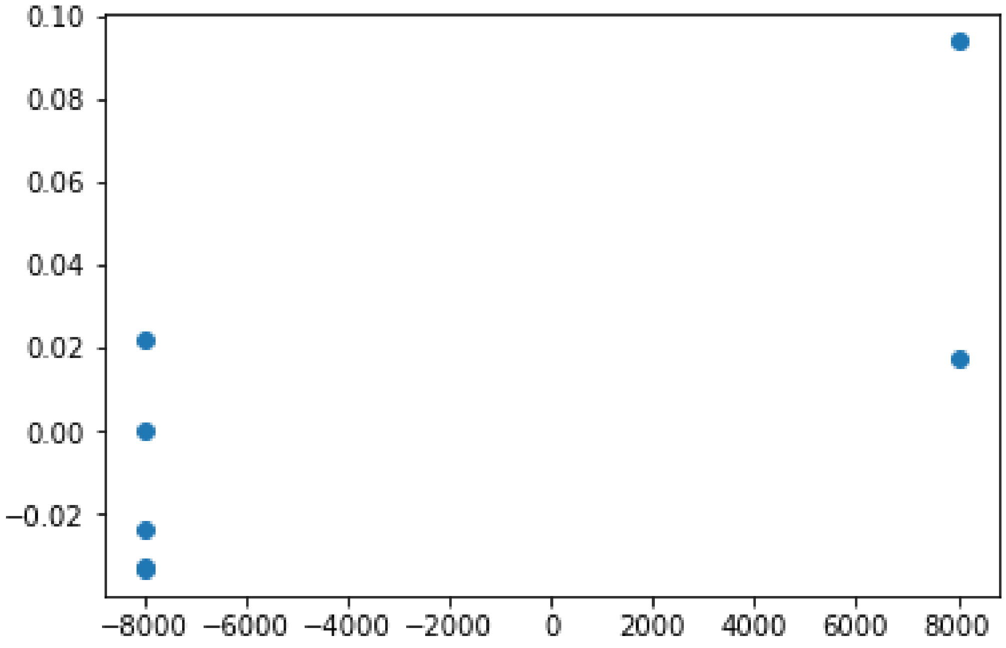
Christopher A Coons Spouse



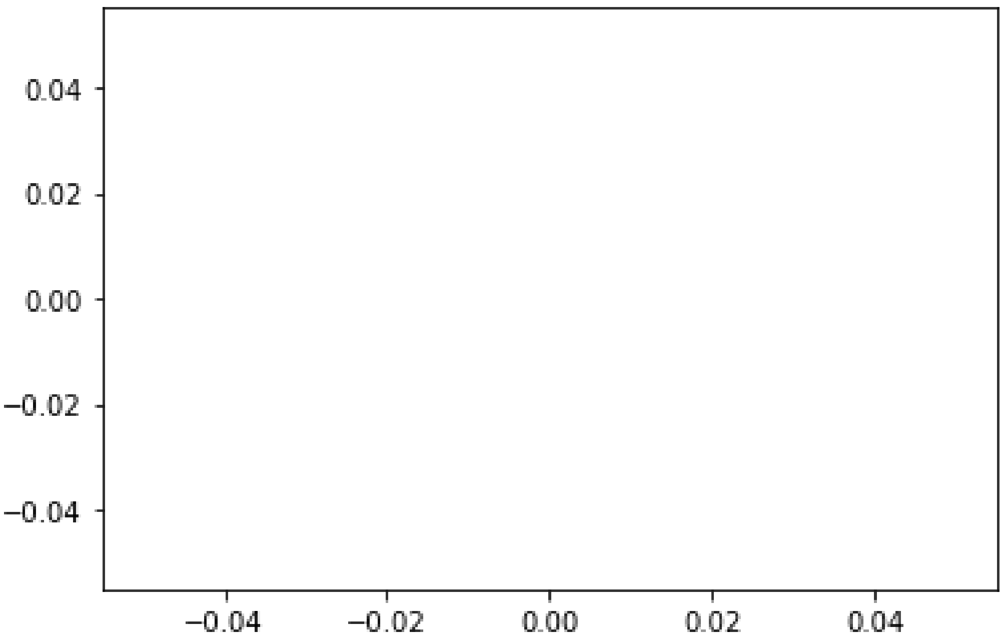
Christopher A Coons Joint



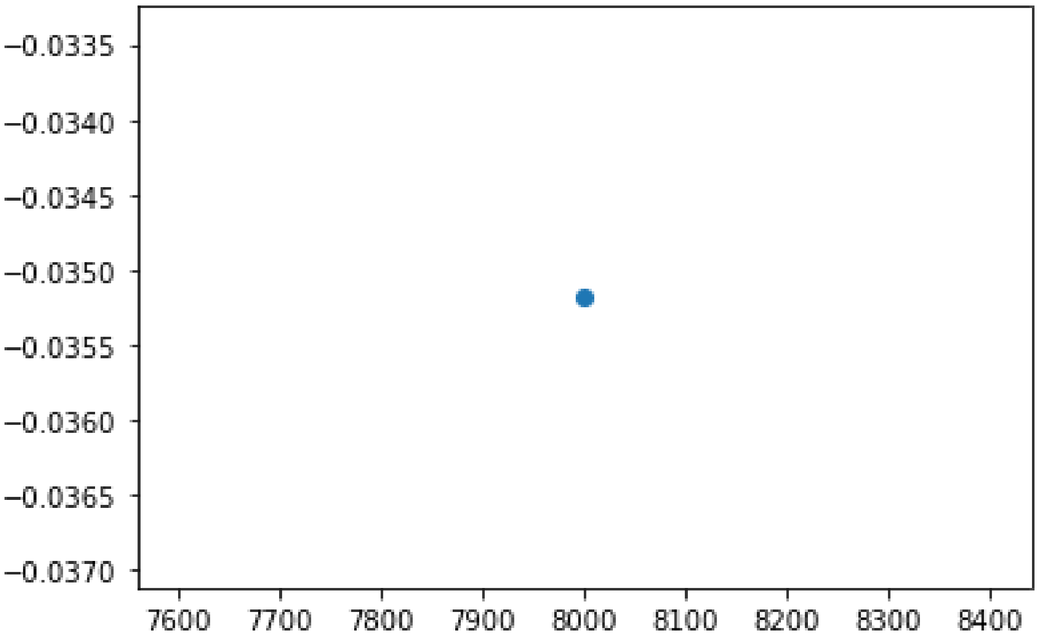
```
Christopher A Coons Child
```



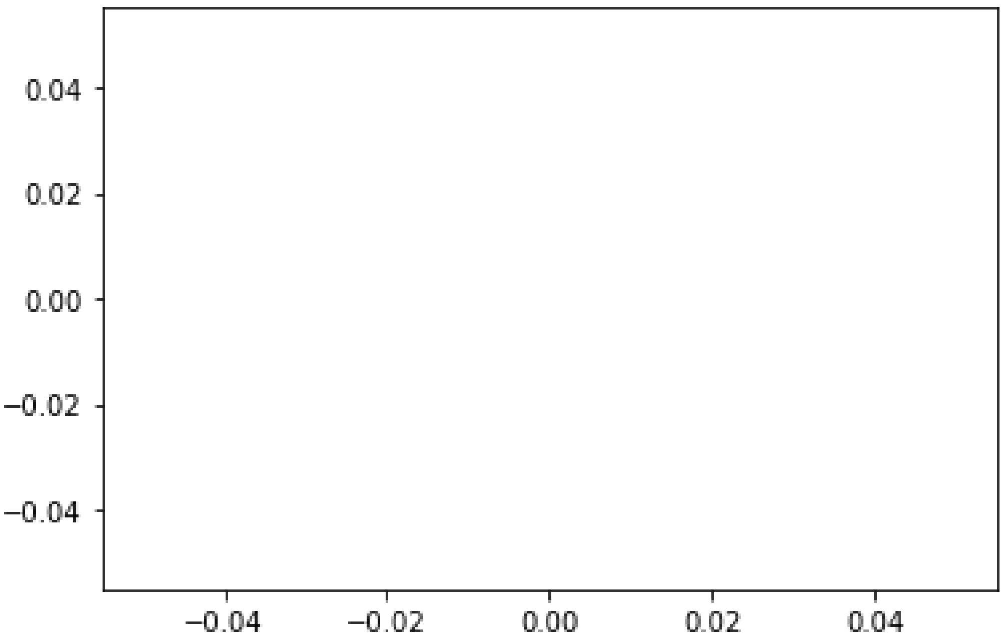
```
Christopher A Coons Self
```



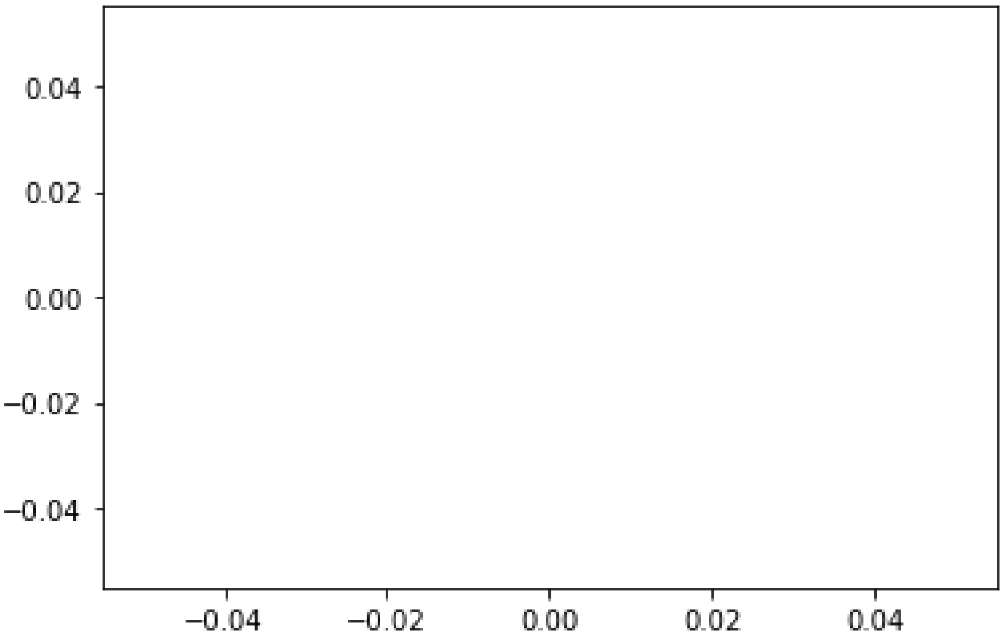
Gary C Peters Spouse



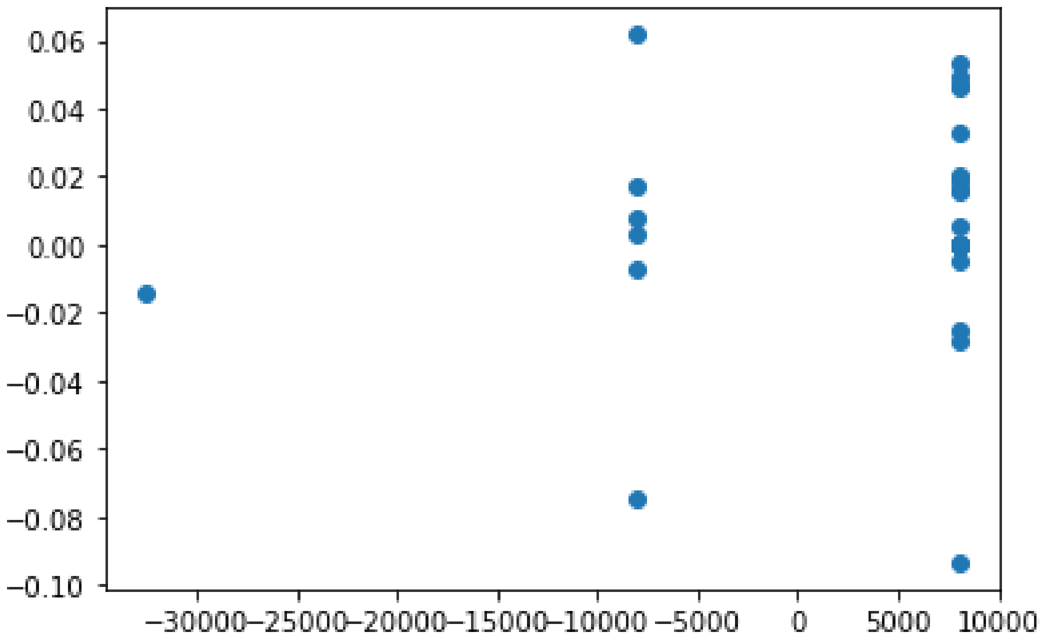
Gary C Peters Joint



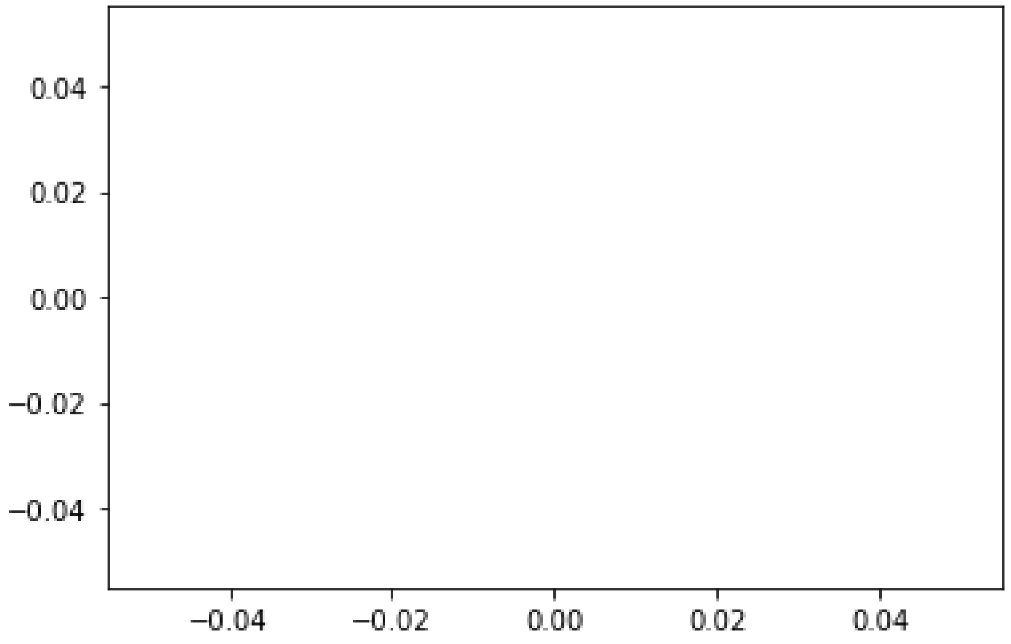
```
Gary C Peters Child
```



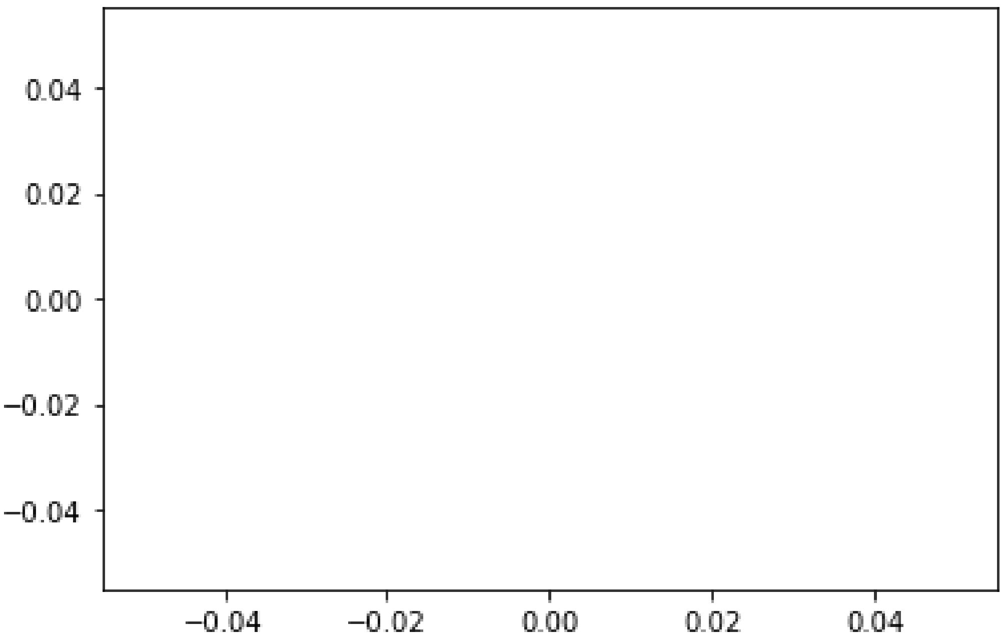
```
Gary C Peters Self
```



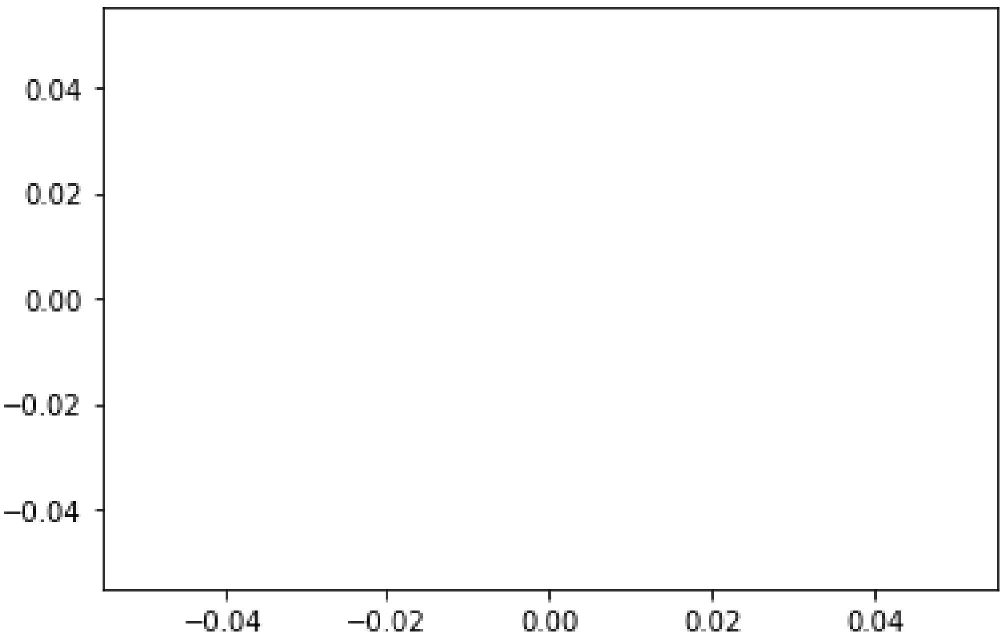
Jacklyn S Rosen Spouse



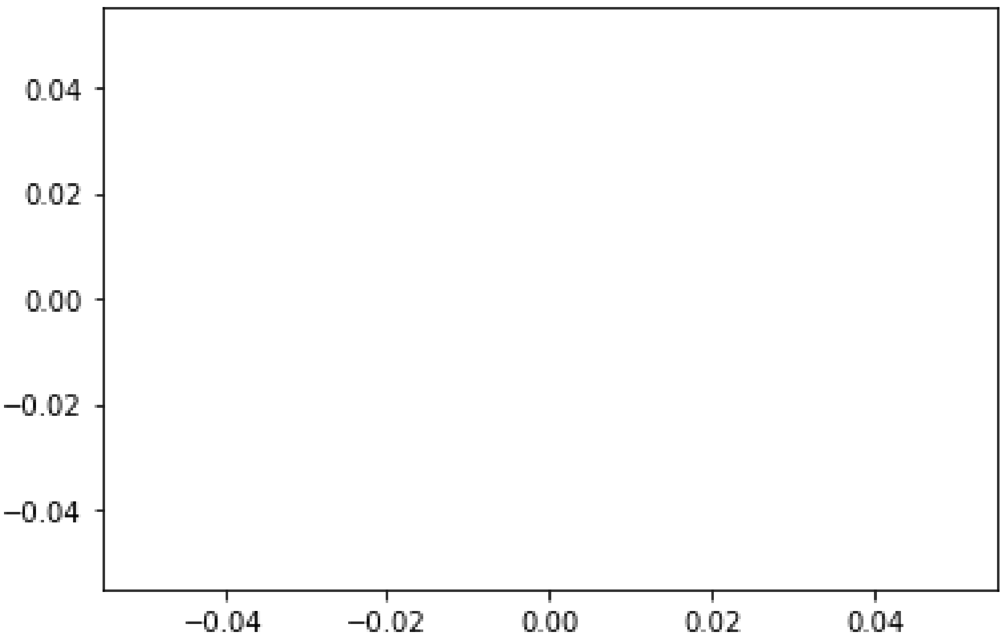
Jacklyn S Rosen Joint



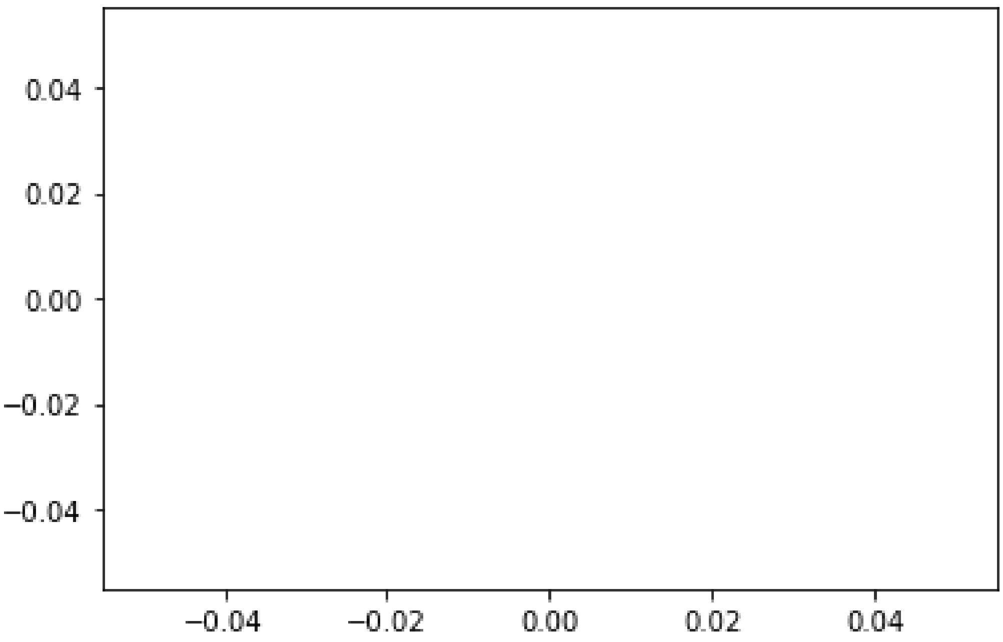
Jacklyn S Rosen Child



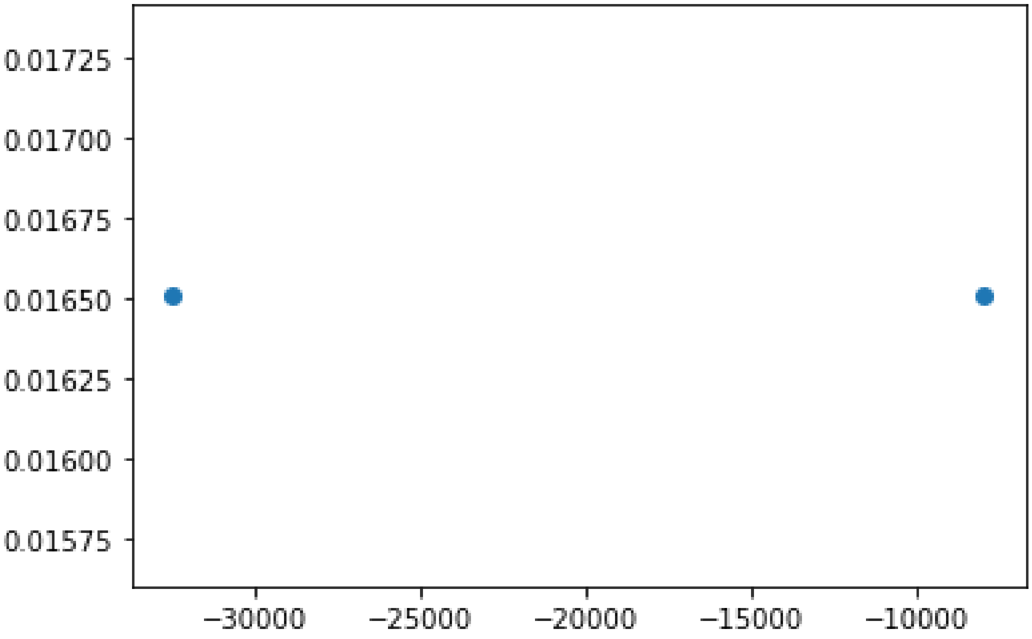
Jacklyn S Rosen Self



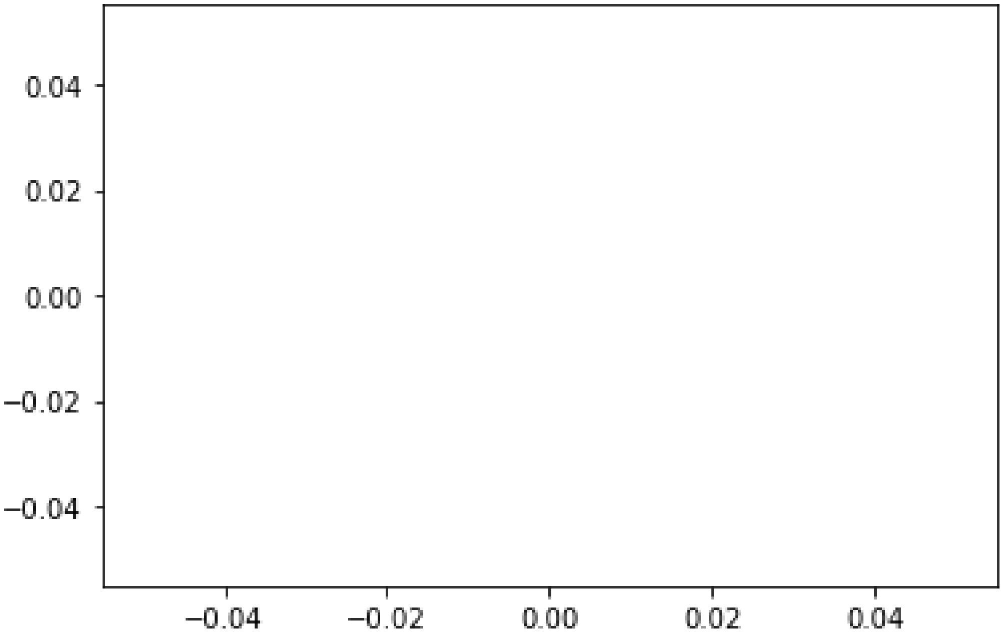
`Thomas Udall Spouse`



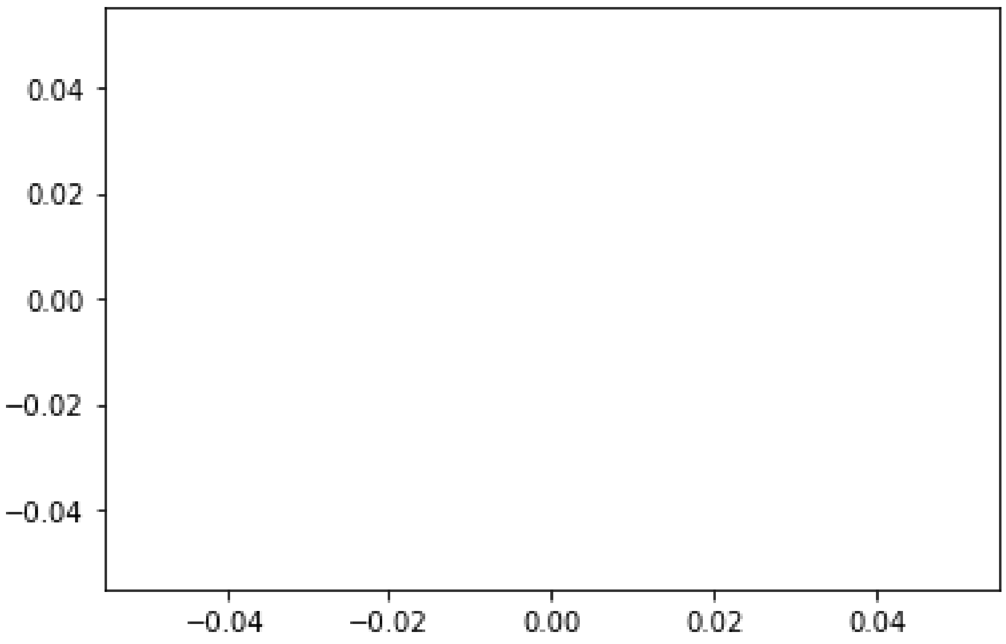
`Thomas Udall Joint`



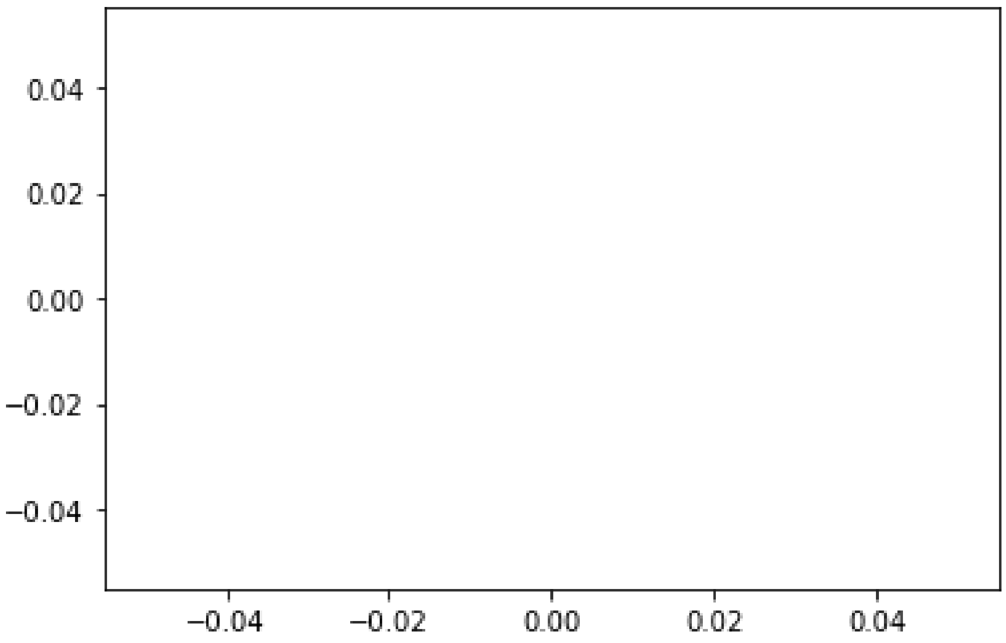
`Thomas Udall Child`



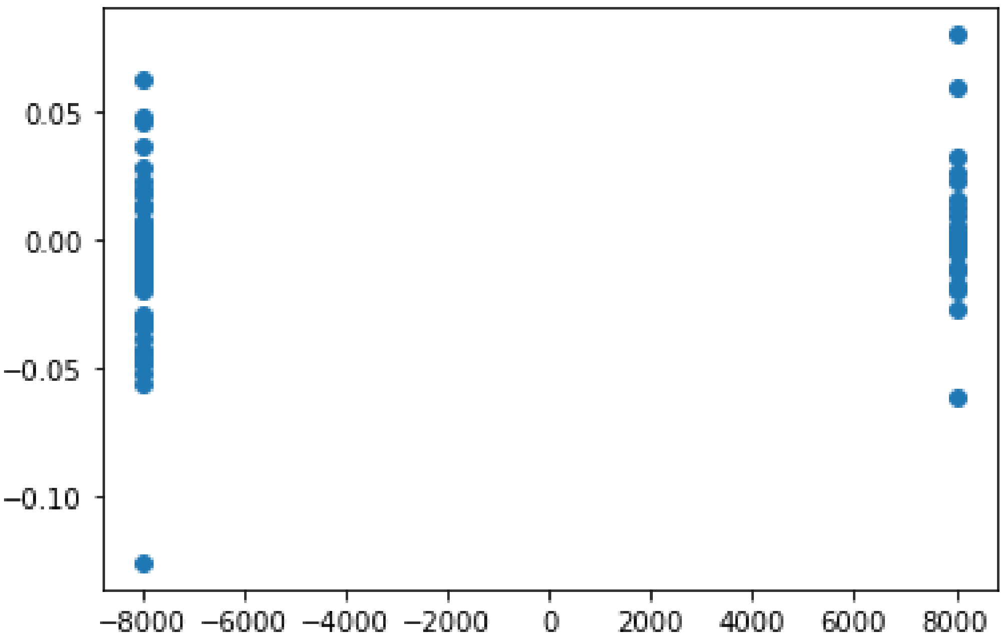
`Thomas Udall Self`



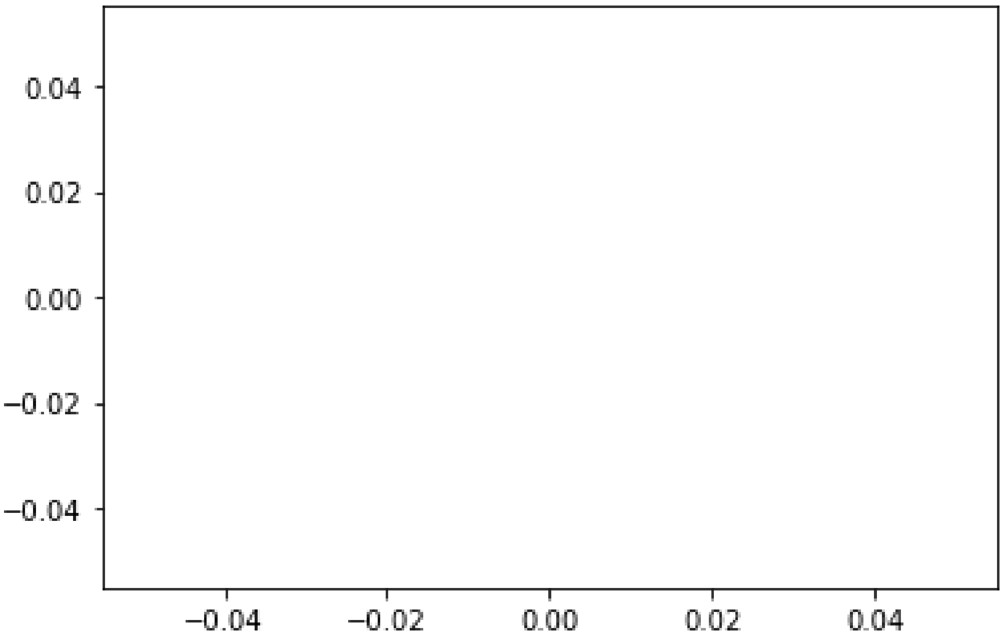
John F Reed Spouse



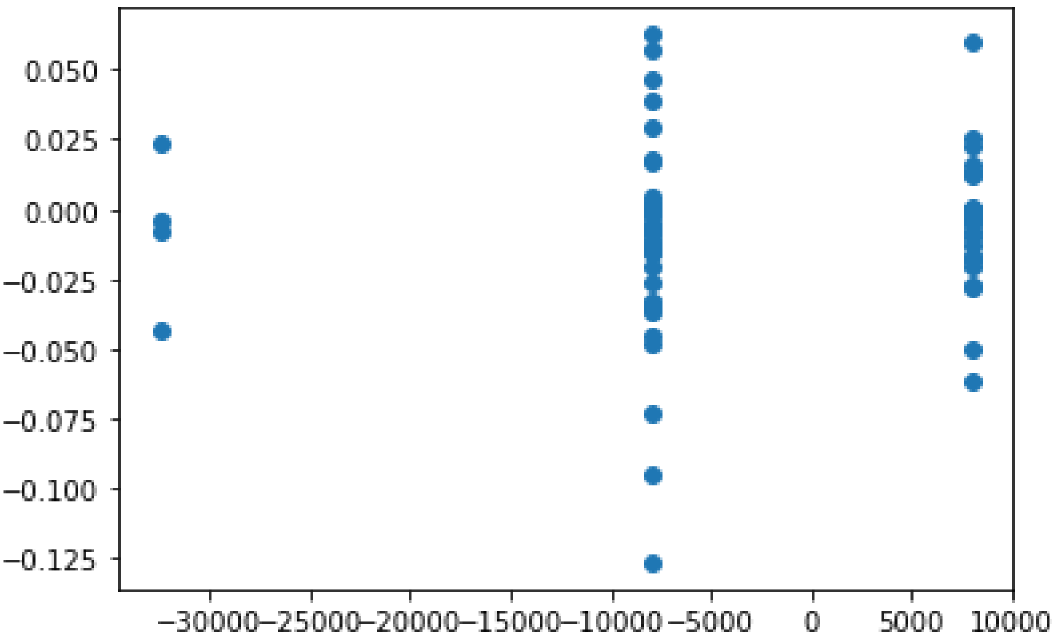
John F Reed Joint



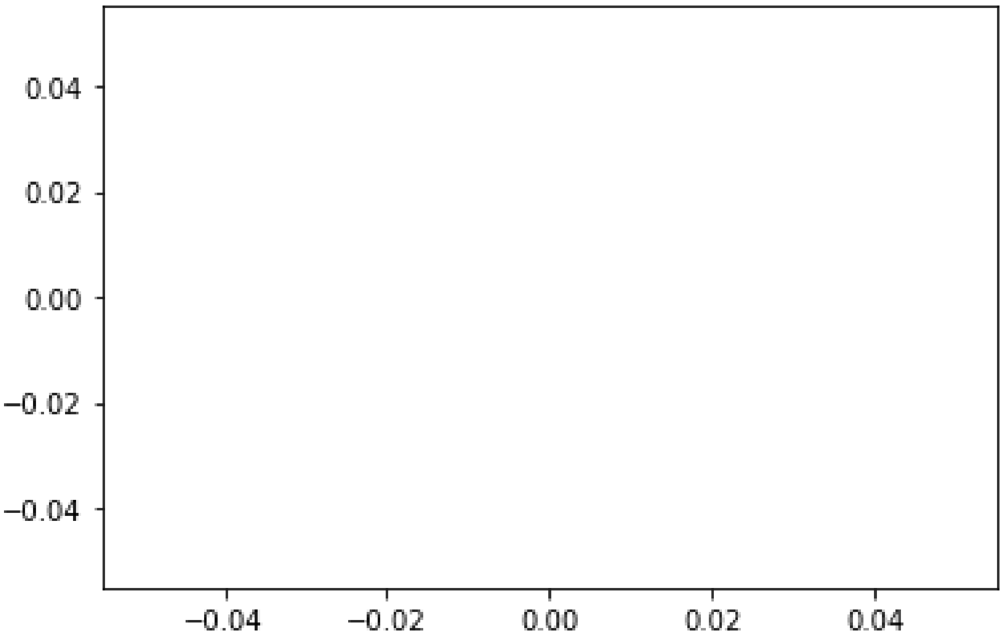
John F Reed Child



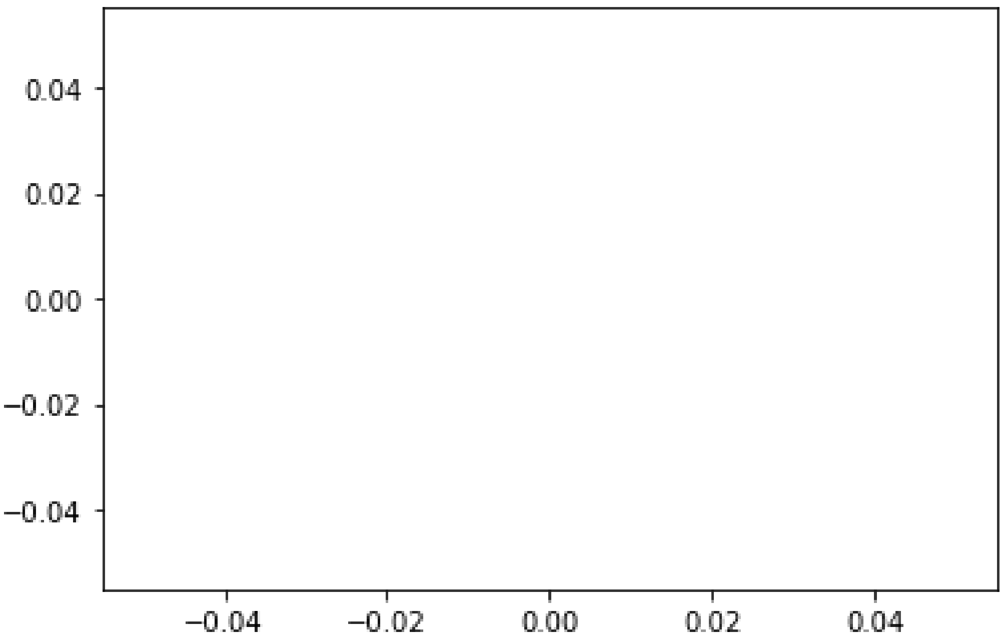
John F Reed Self



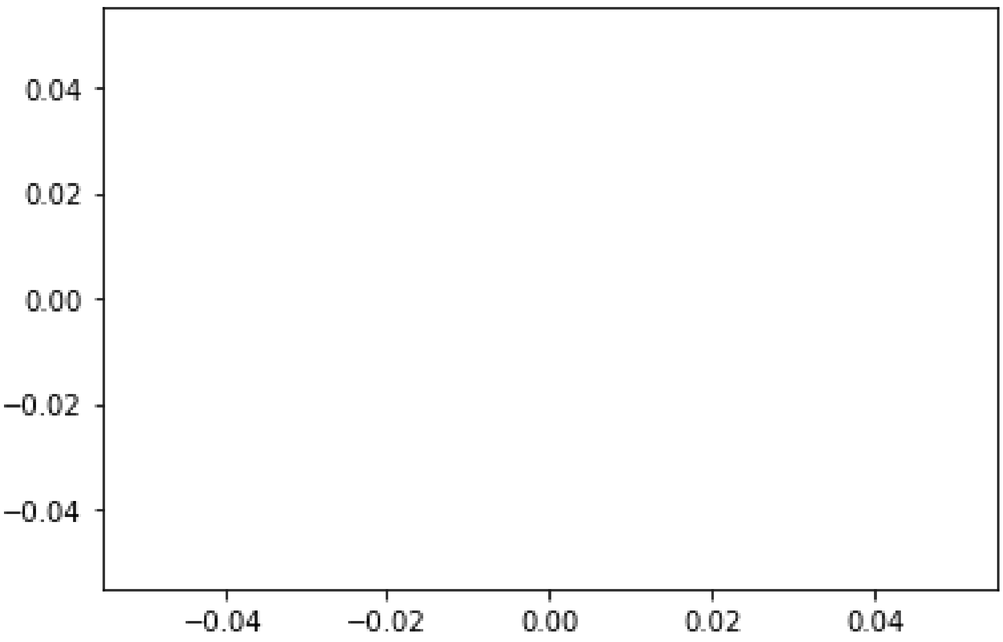
Mark R Warner Spouse



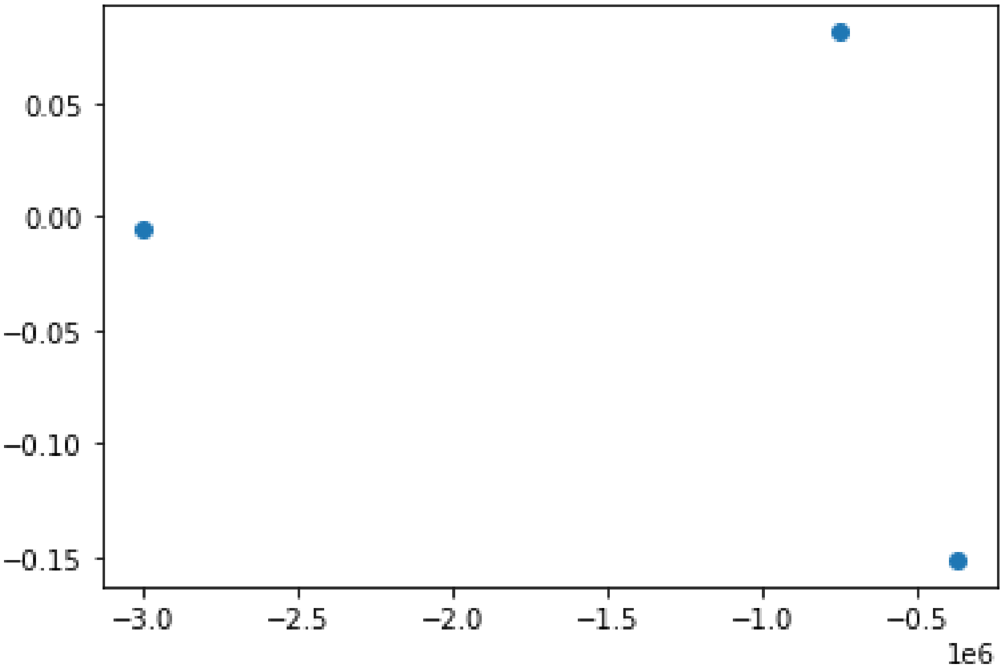
Mark R Warner Joint



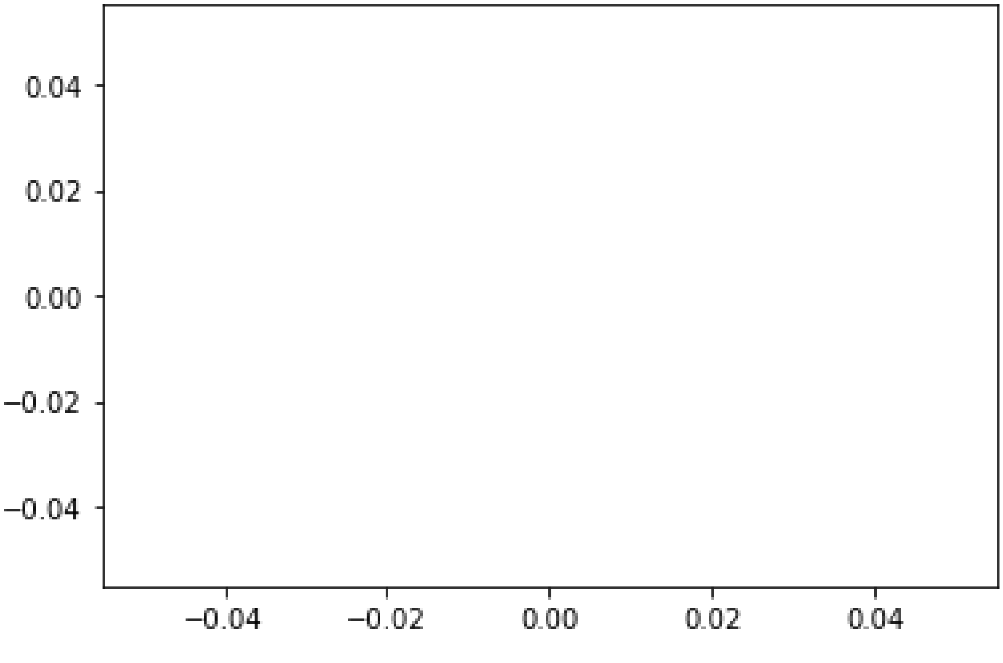
Mark R Warner Child



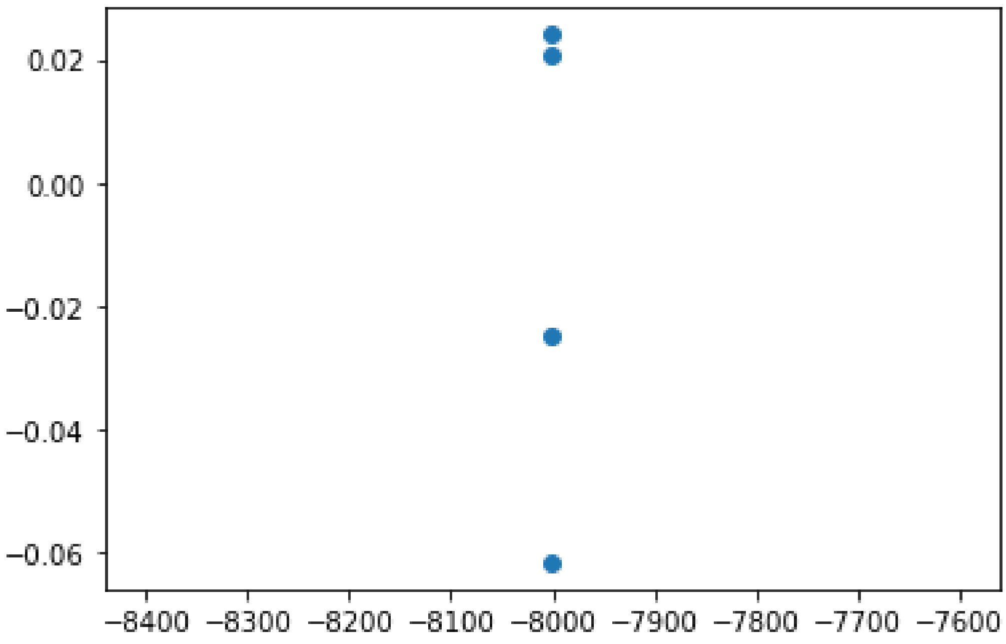
Mark R Warner Self



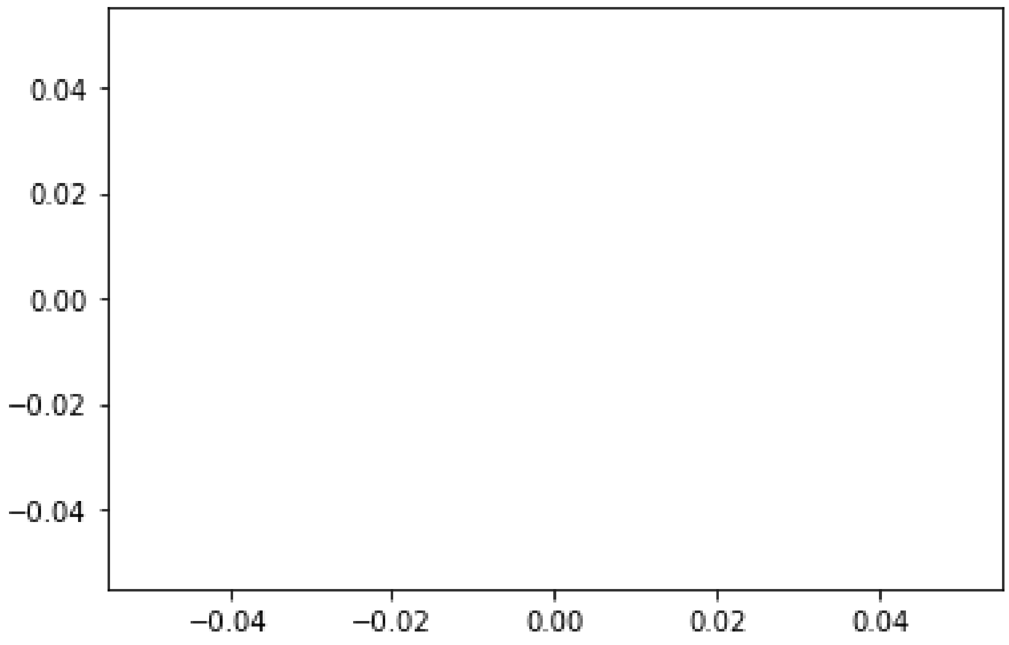
Ladda Tammy Duckworth Spouse



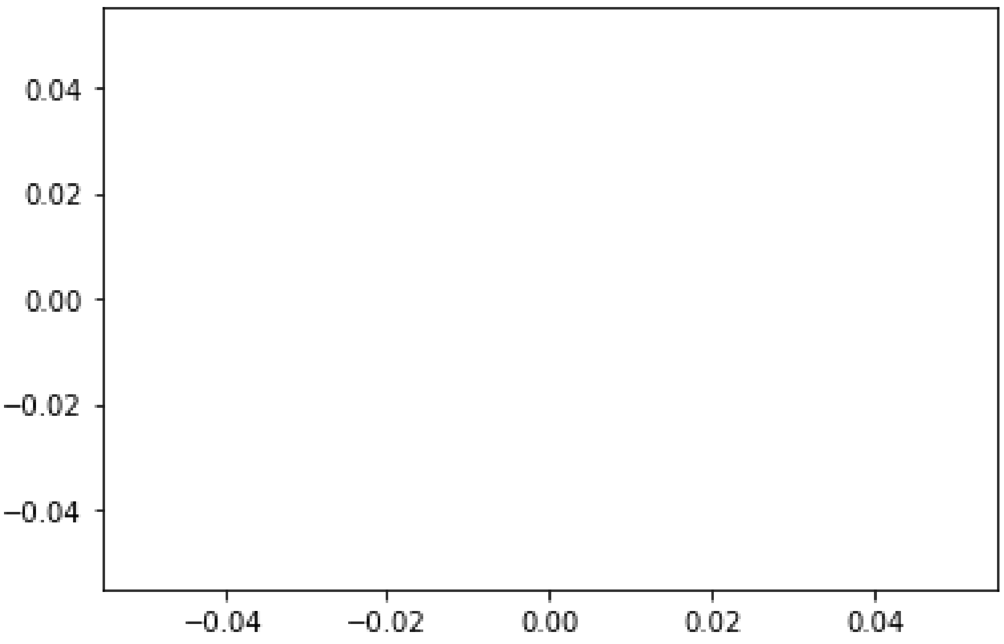
Ladda Tammy Duckworth Joint



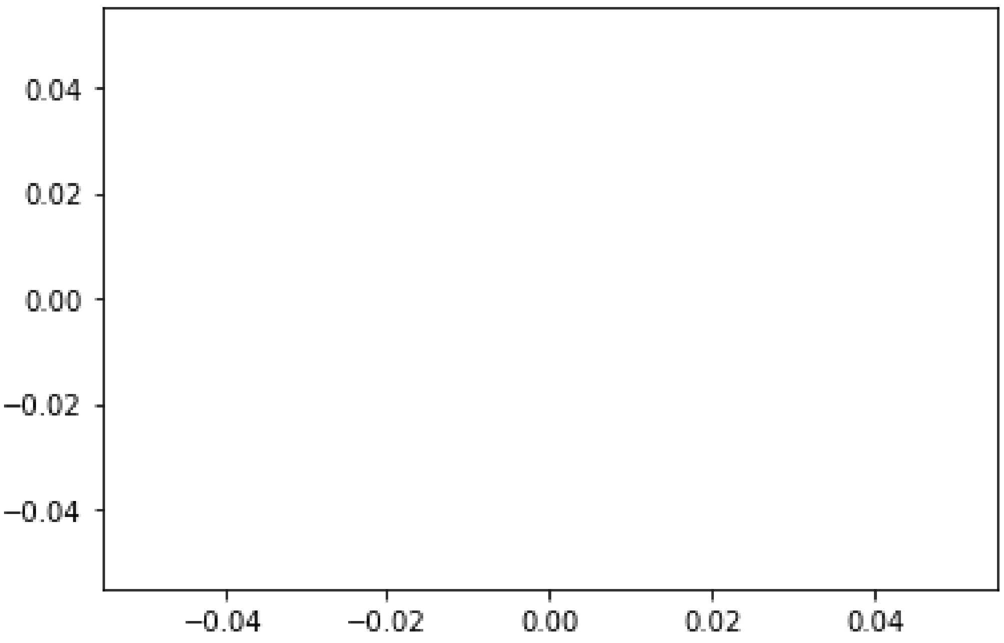
`Ladda Tammy Duckworth Child`



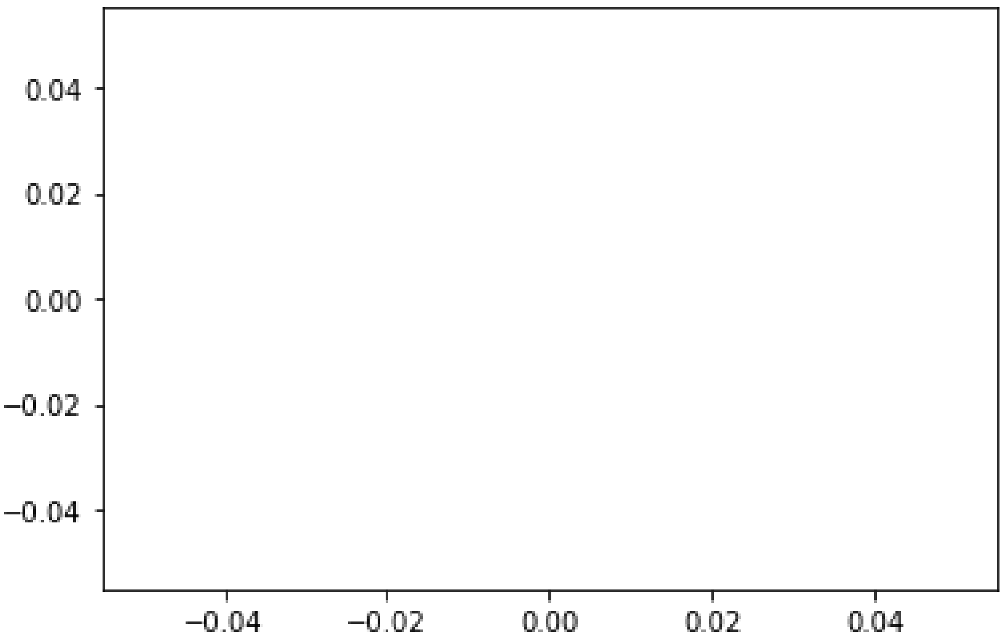
`Ladda Tammy Duckworth Self`



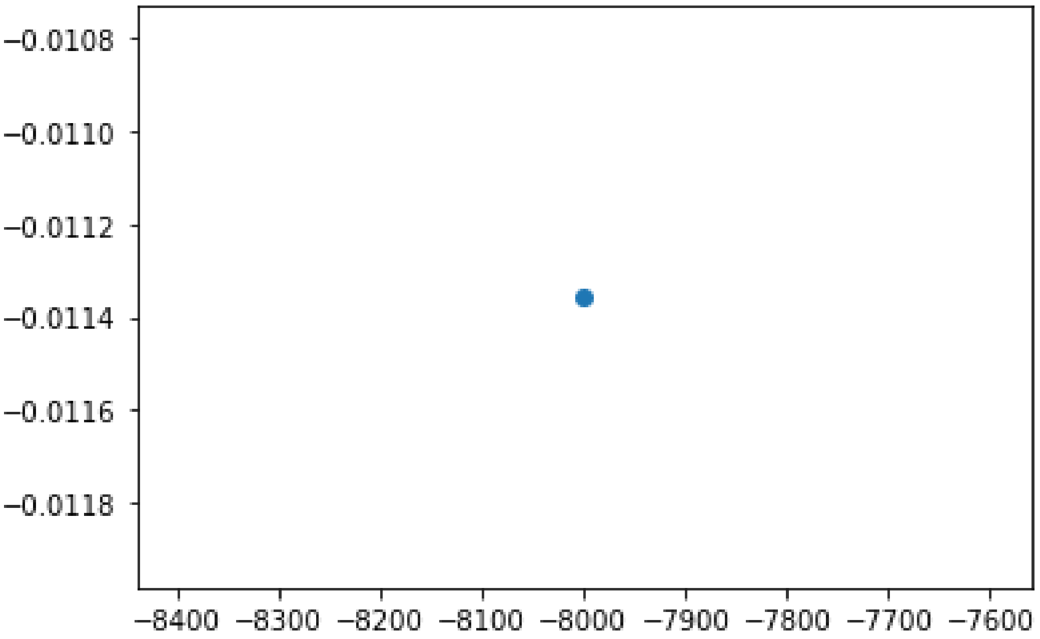
Robert P Casey, Jr. Spouse



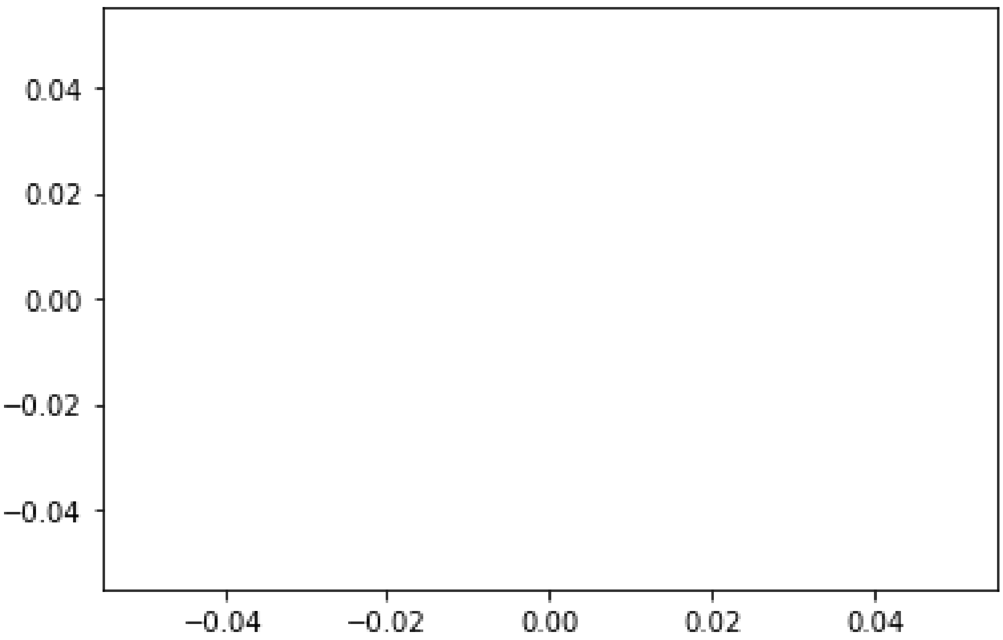
Robert P Casey, Jr. Joint



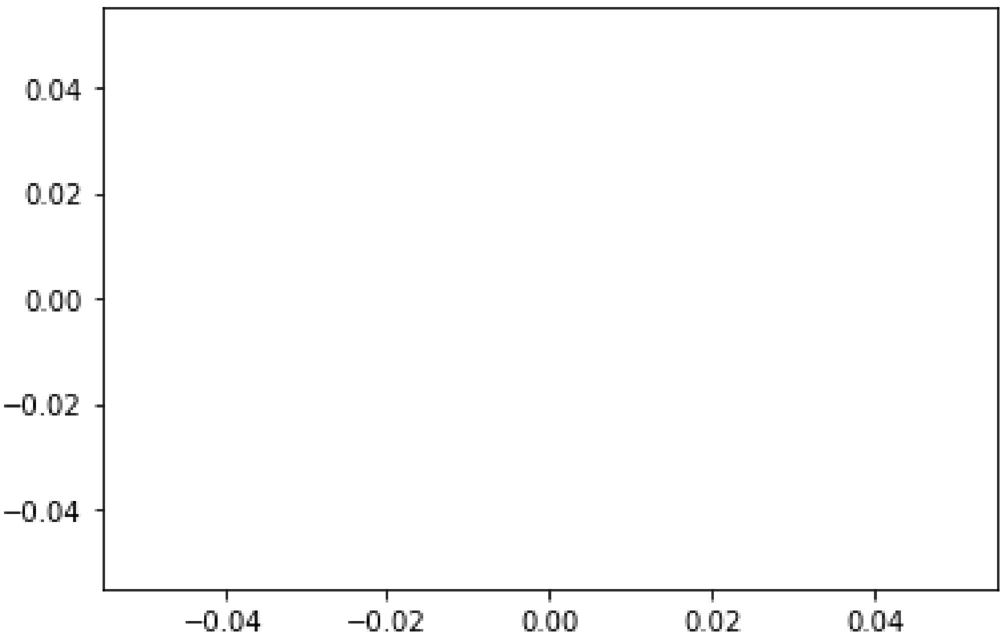
Robert P Casey, Jr. Child



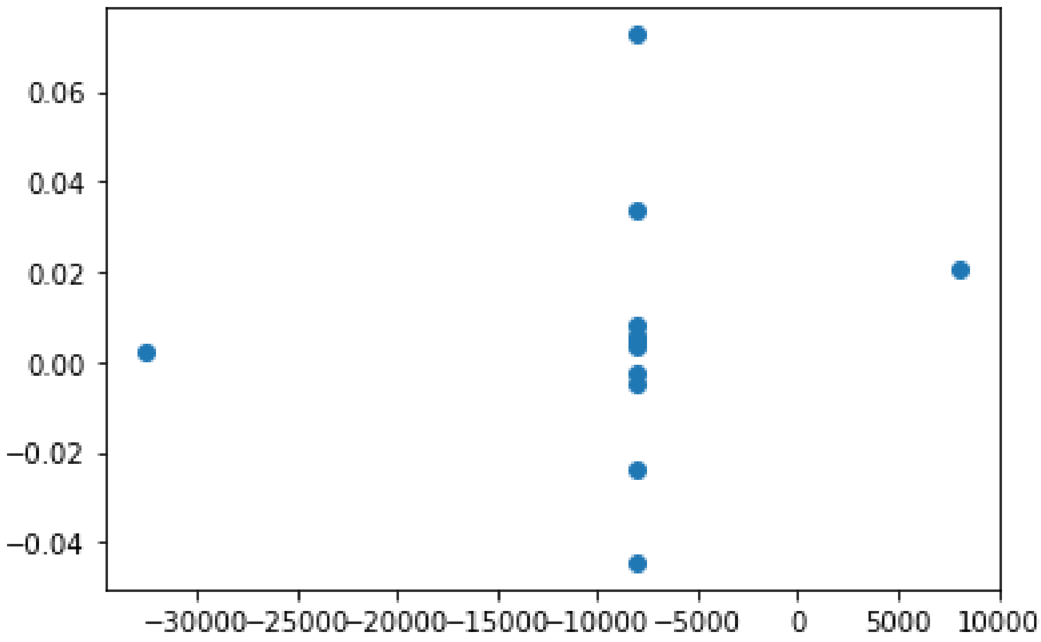
Robert P Casey, Jr. Self



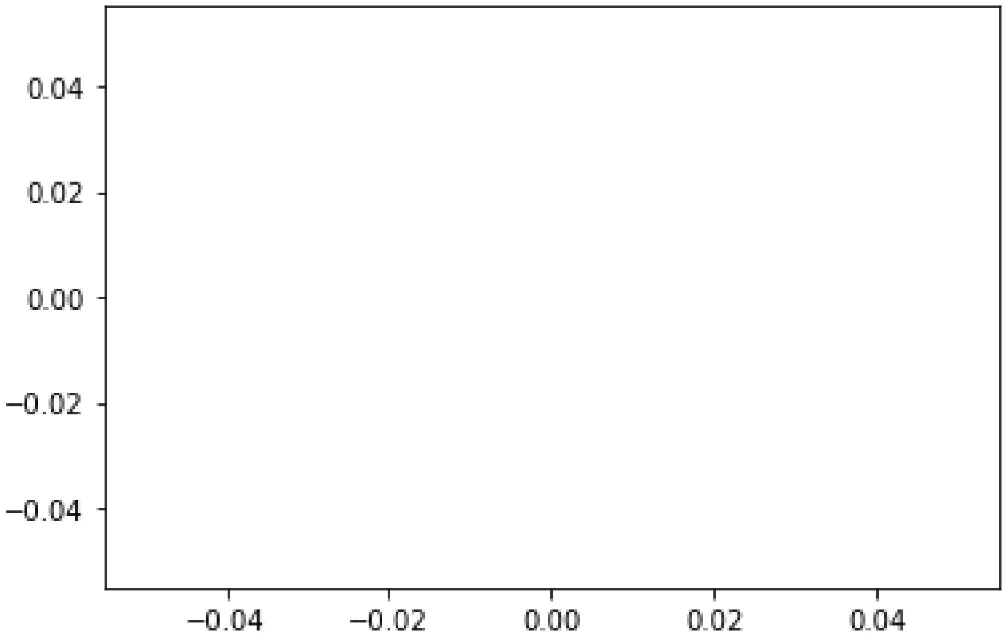
`Tammy Duckworth Spouse`



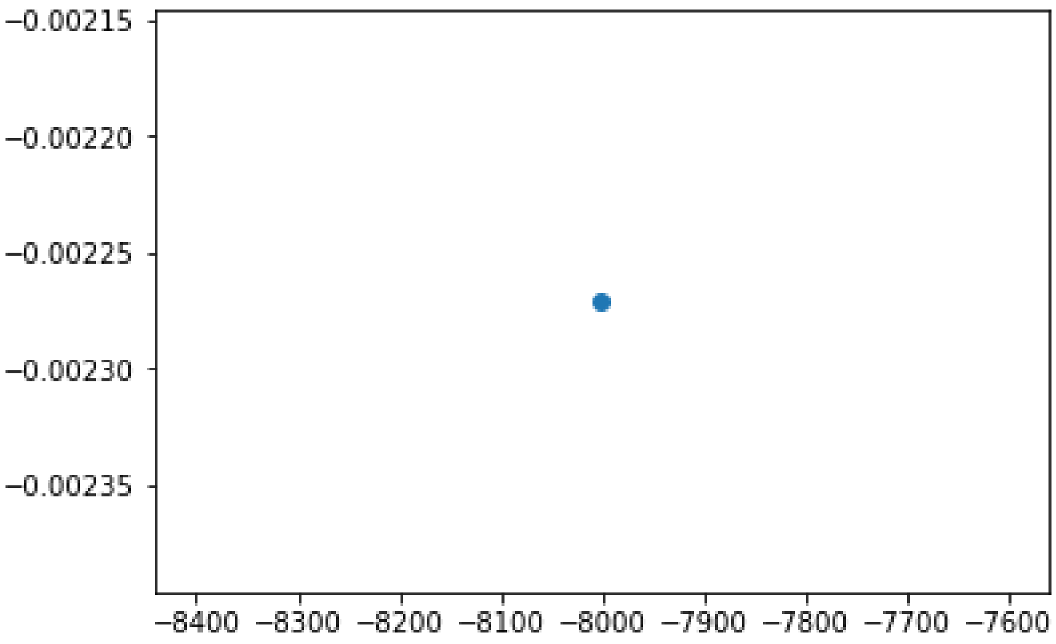
`Tammy Duckworth Joint`



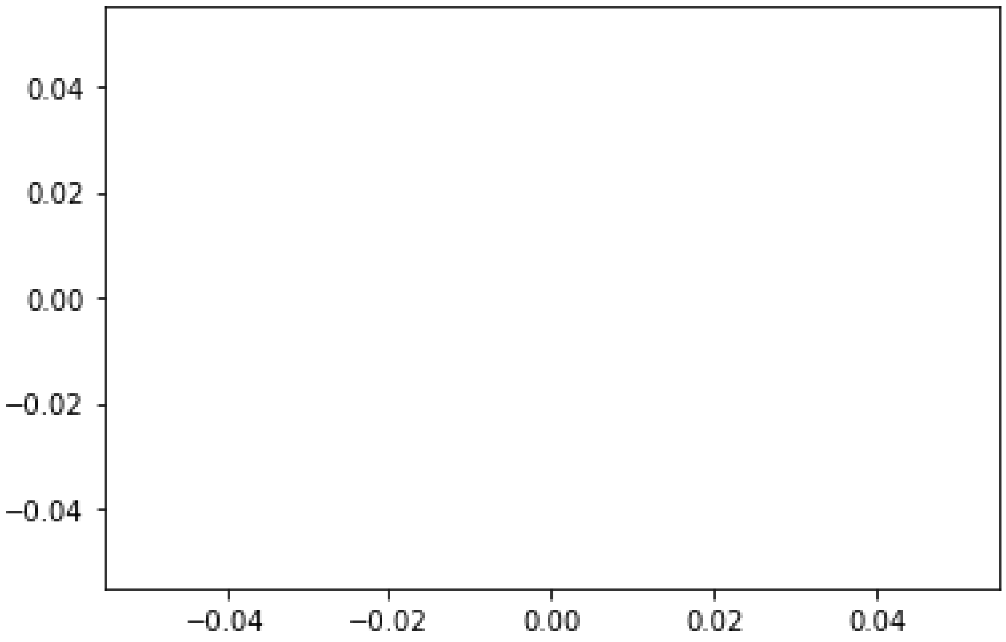
Tammy Duckworth Child



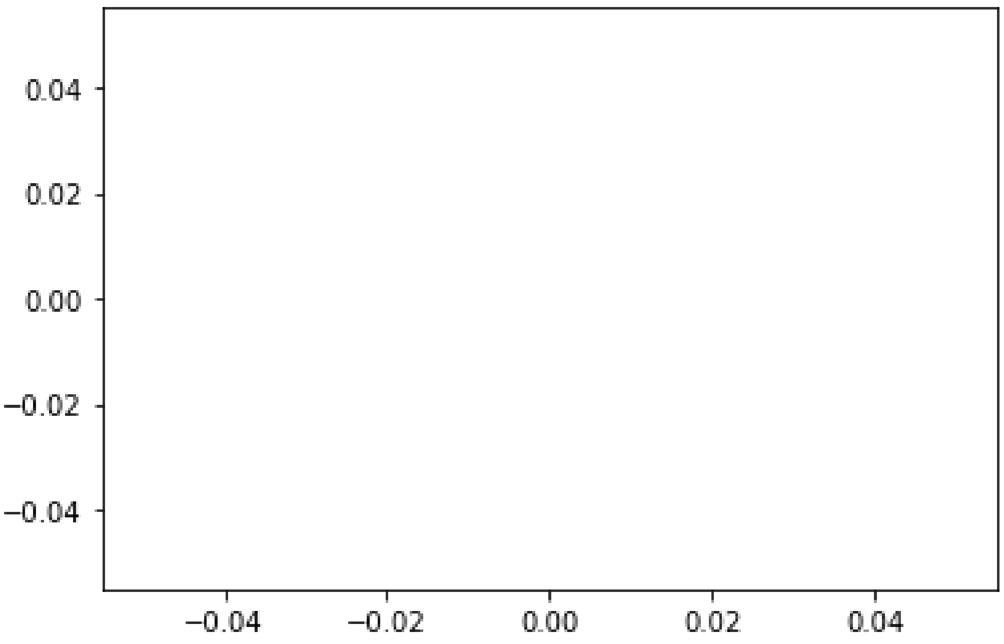
Tammy Duckworth Self



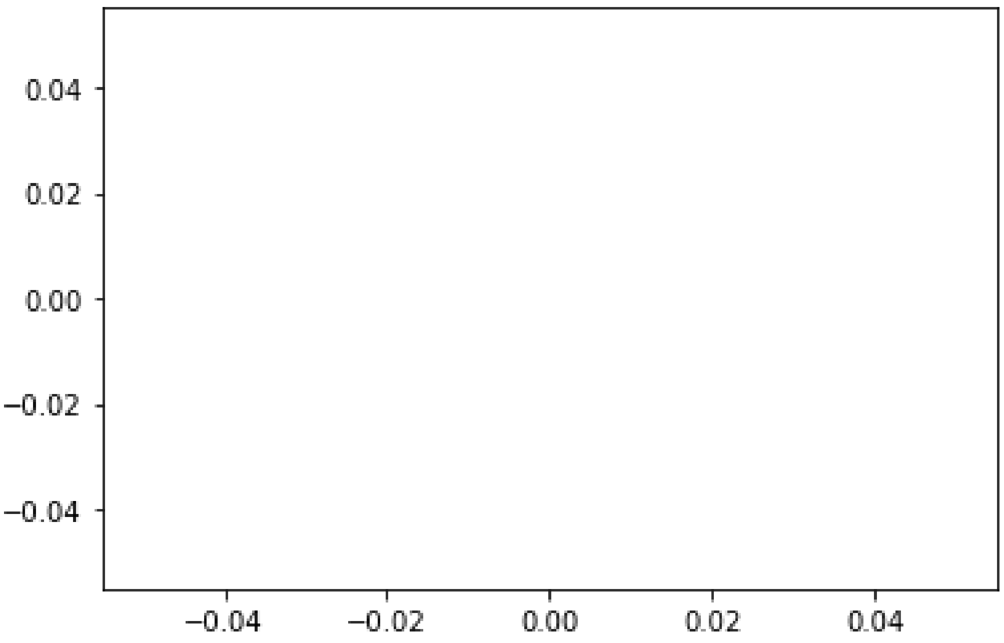
```
Michael F Bennet Spouse
```



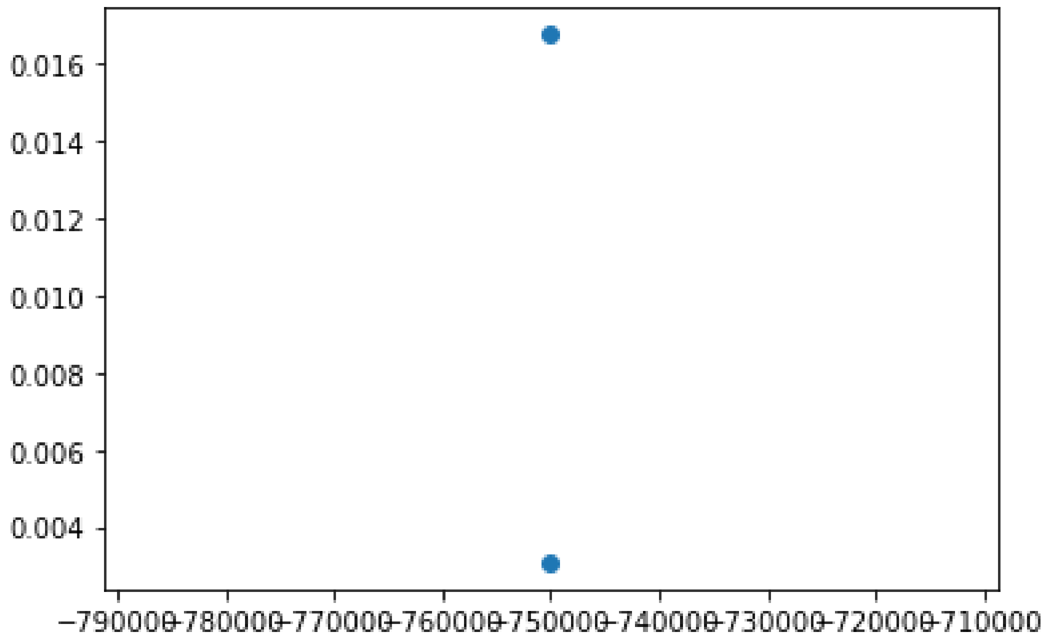
```
Michael F Bennet Joint
```



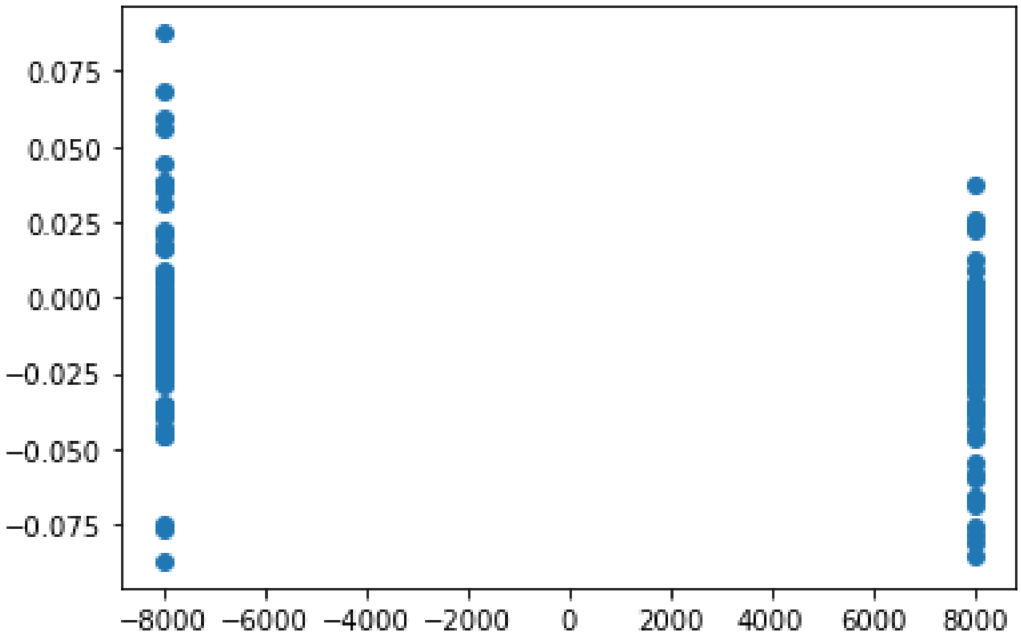
```
Michael F Bennet Child
```



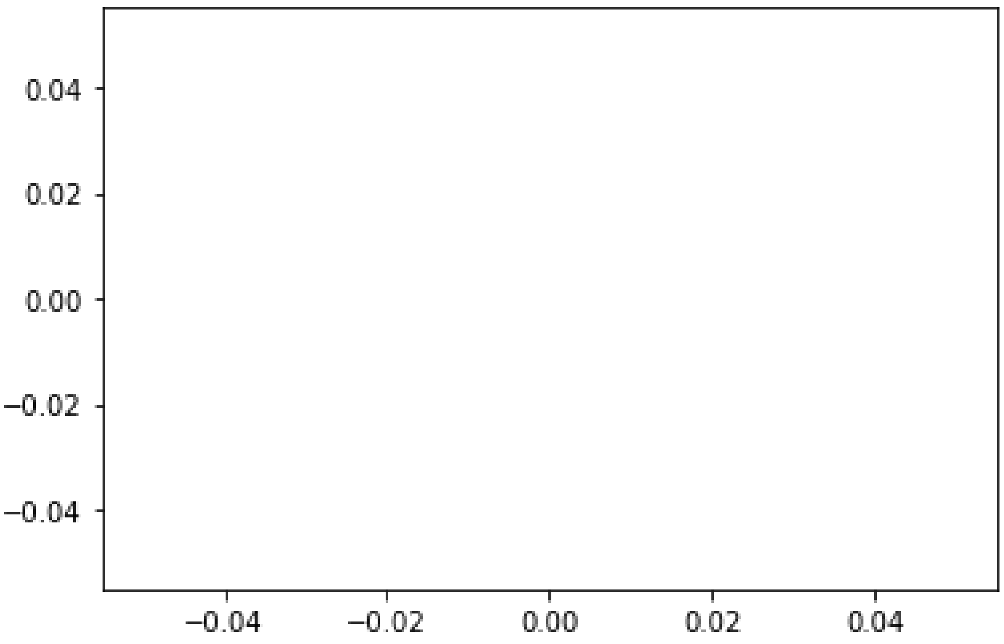
```
Michael F Bennet Self
```



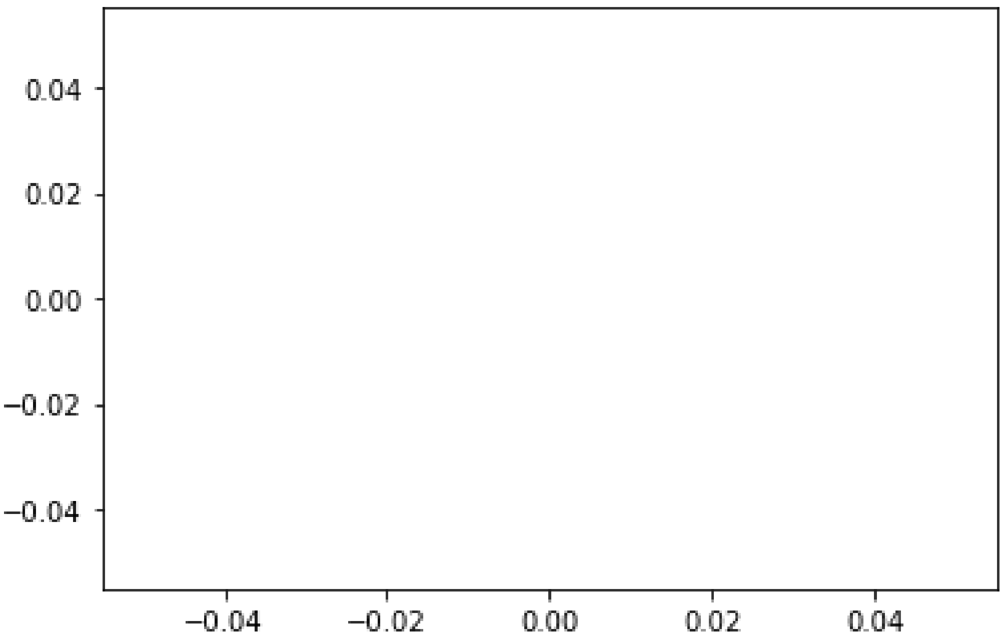
```
Patty Murray Spouse
```



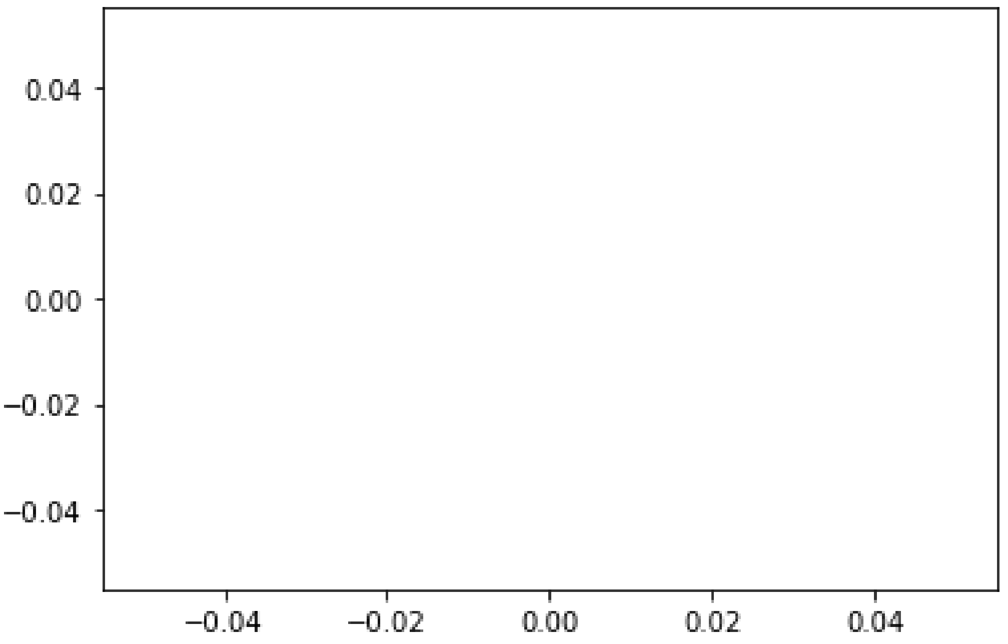
```
Patty Murray Joint
```



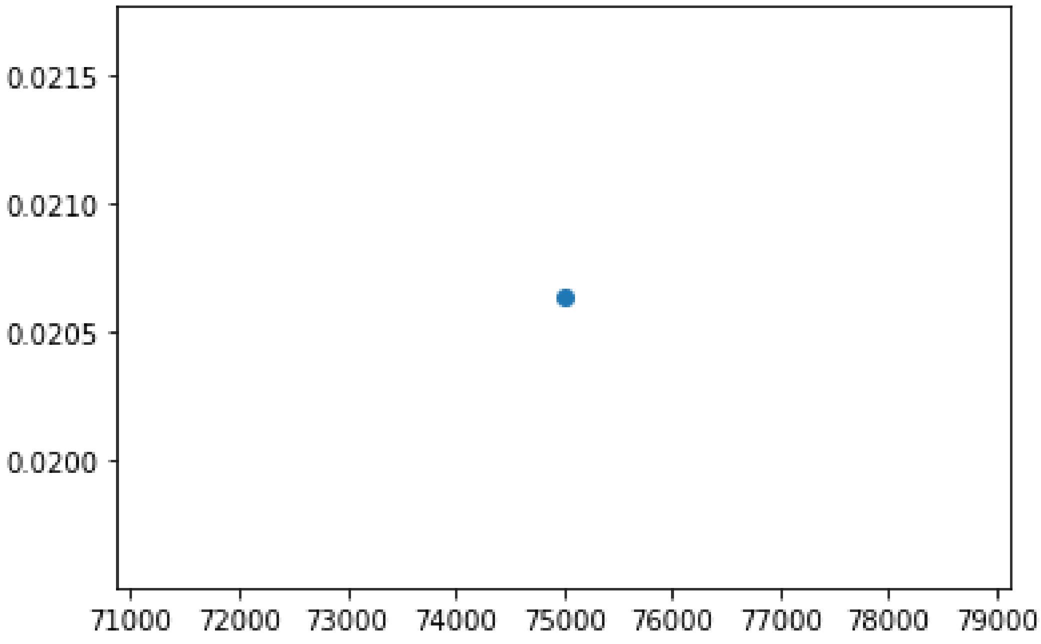
`Patty Murray Child`



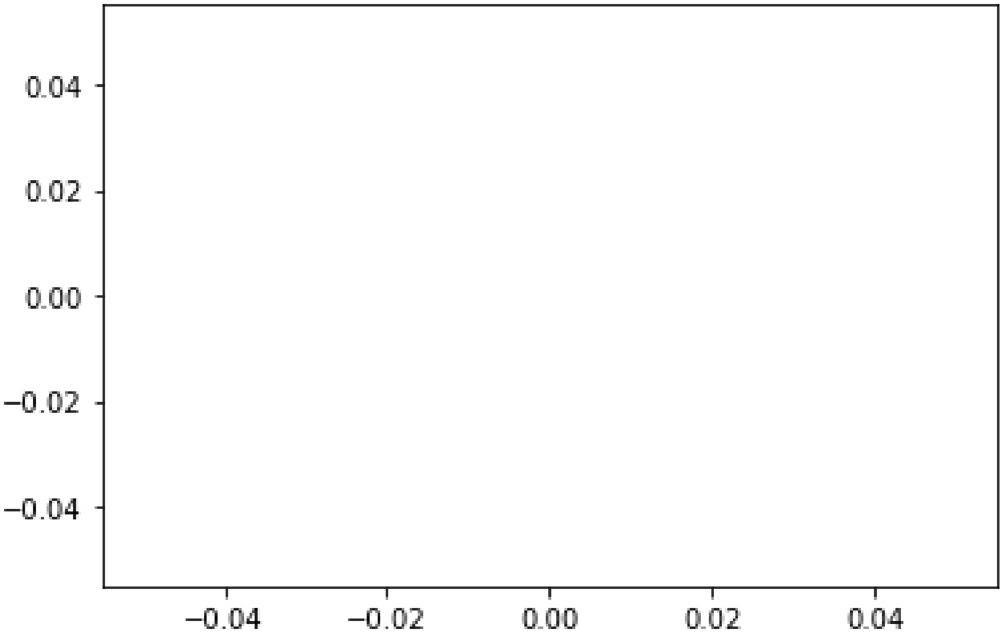
`Patty Murray Self`



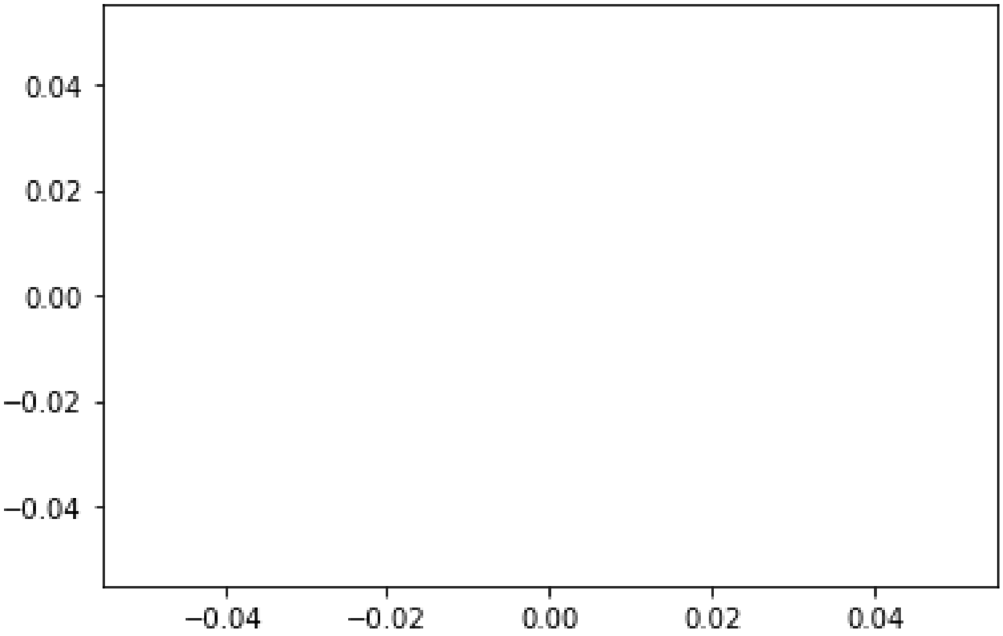
Joseph Manchin, Iii Spouse



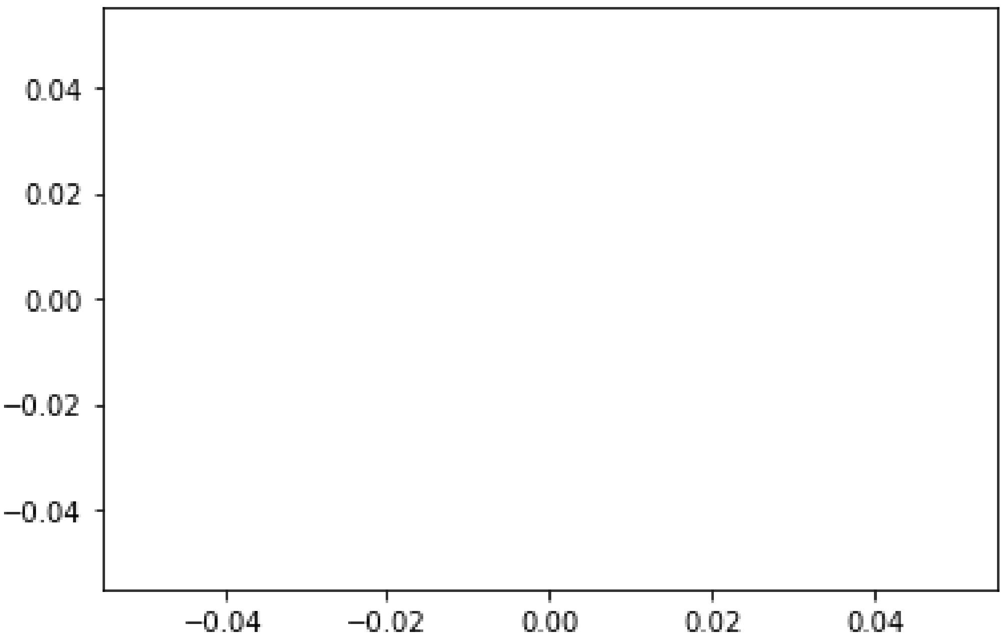
Joseph Manchin, Iii Joint



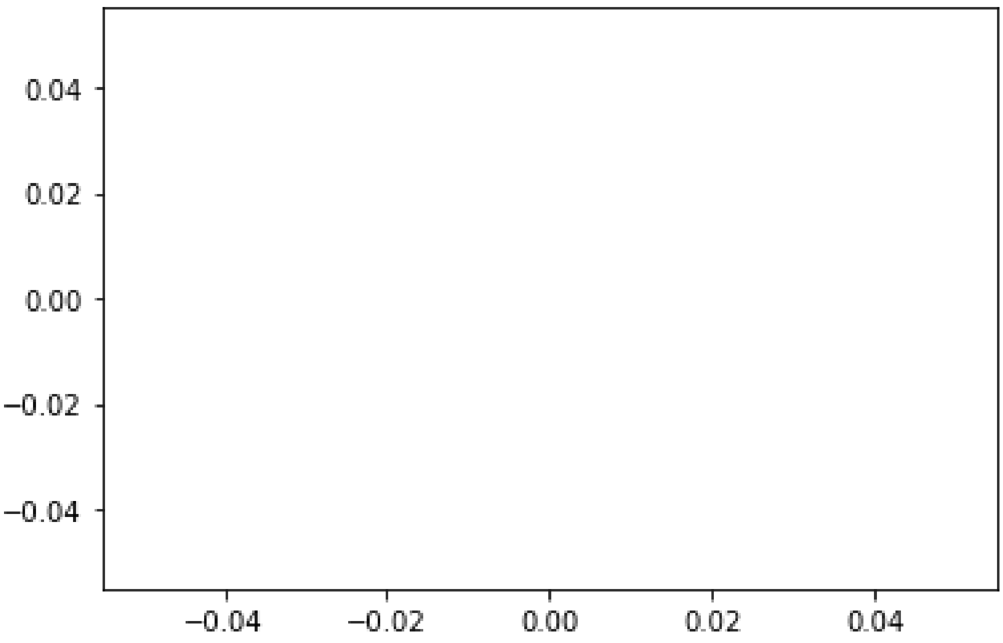
Joseph Manchin, Iii Child



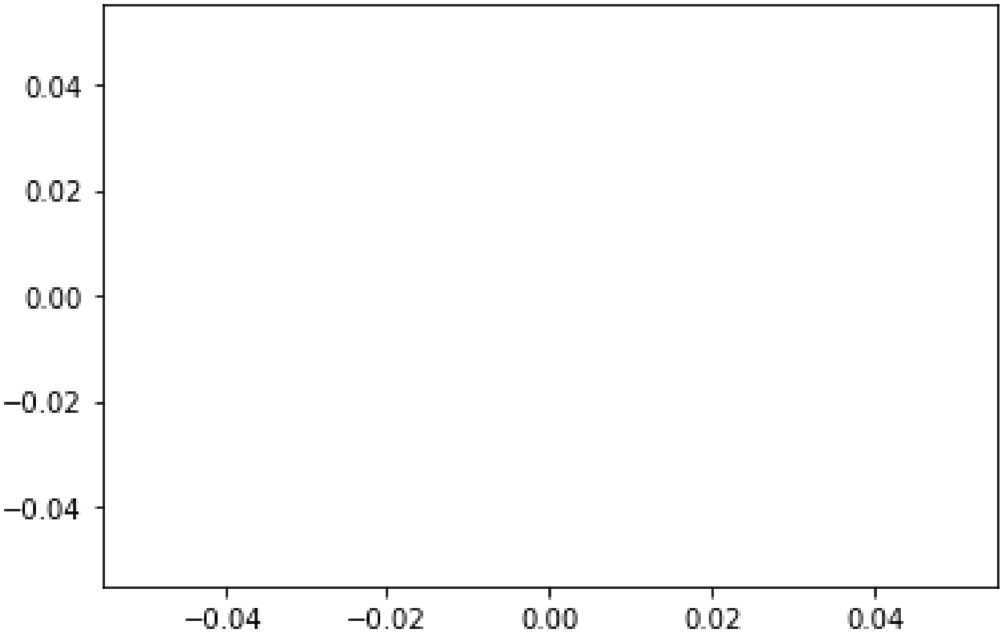
Joseph Manchin, Iii Self



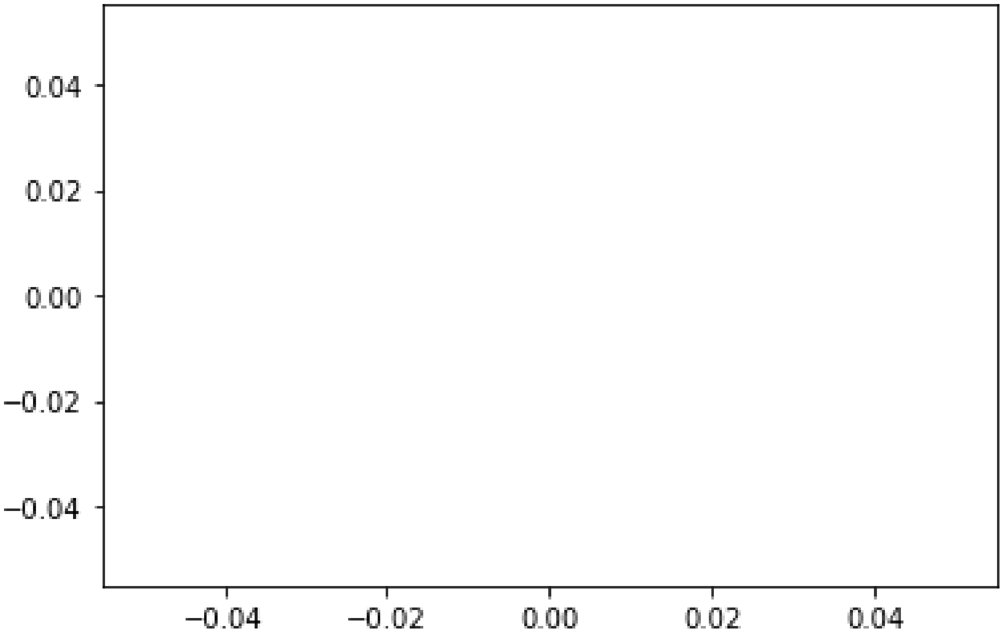
```
Chris Van Hollen Spouse
```



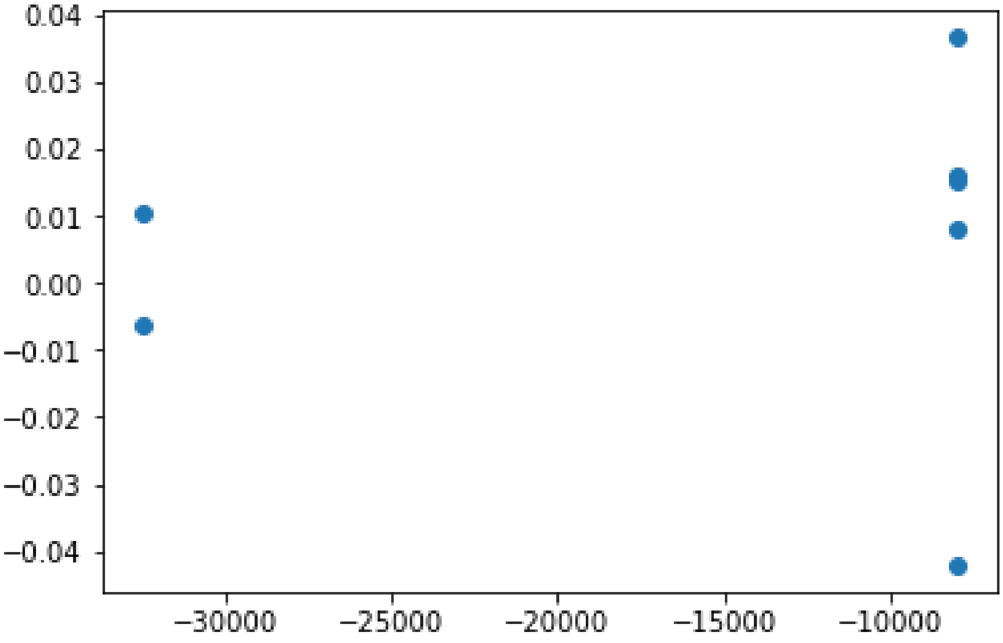
```
Chris Van Hollen Joint
```



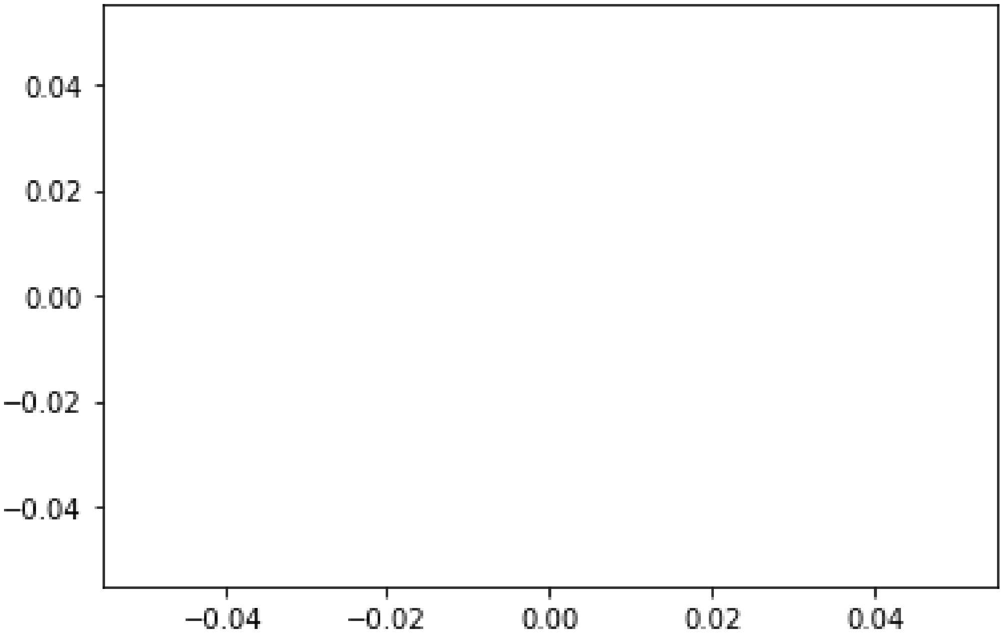
Chris Van Hollen Child



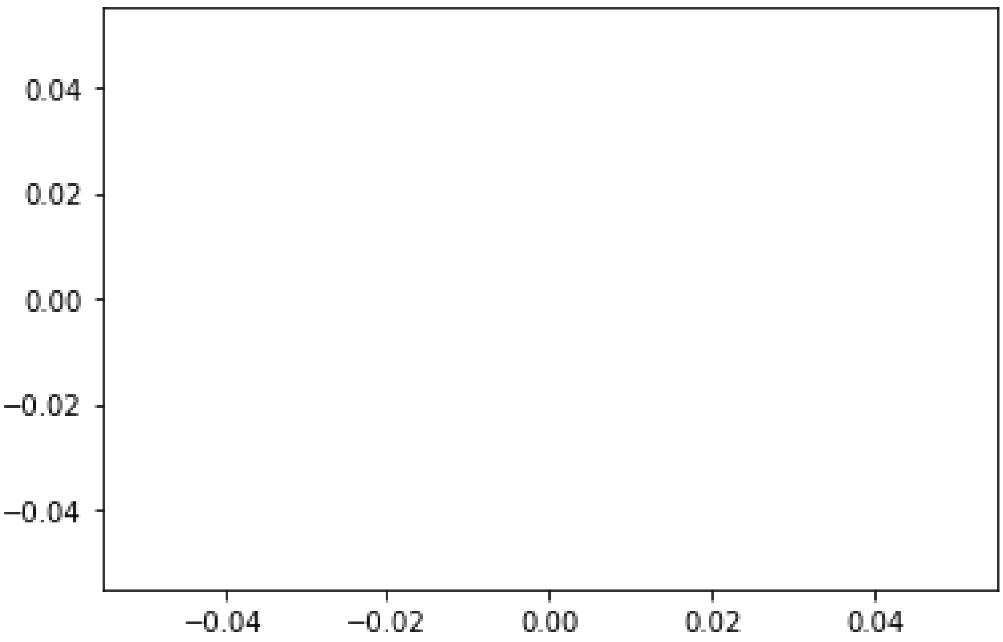
Chris Van Hollen Self



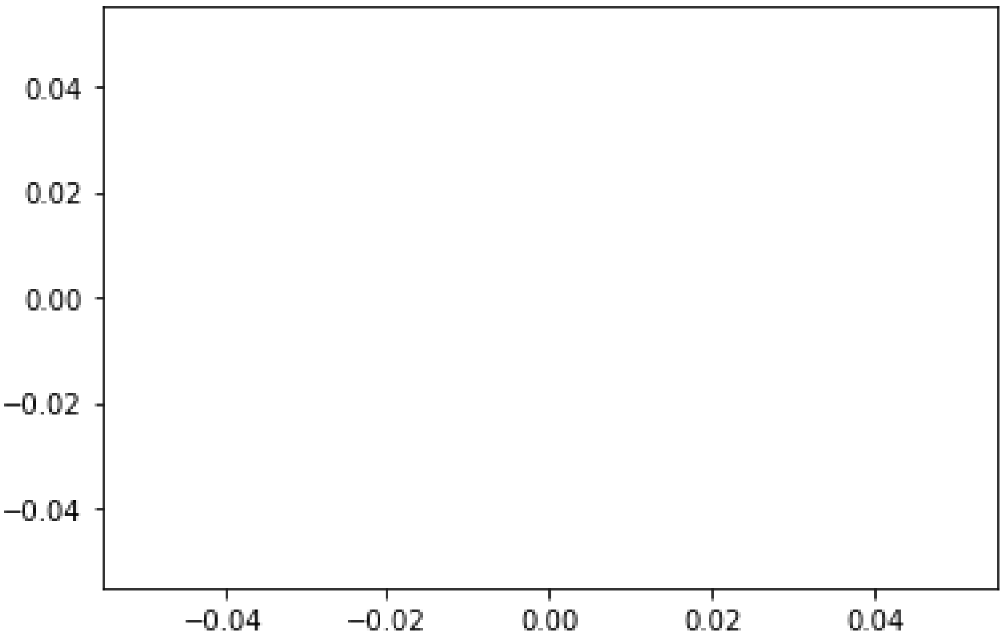
John Cornyn Spouse



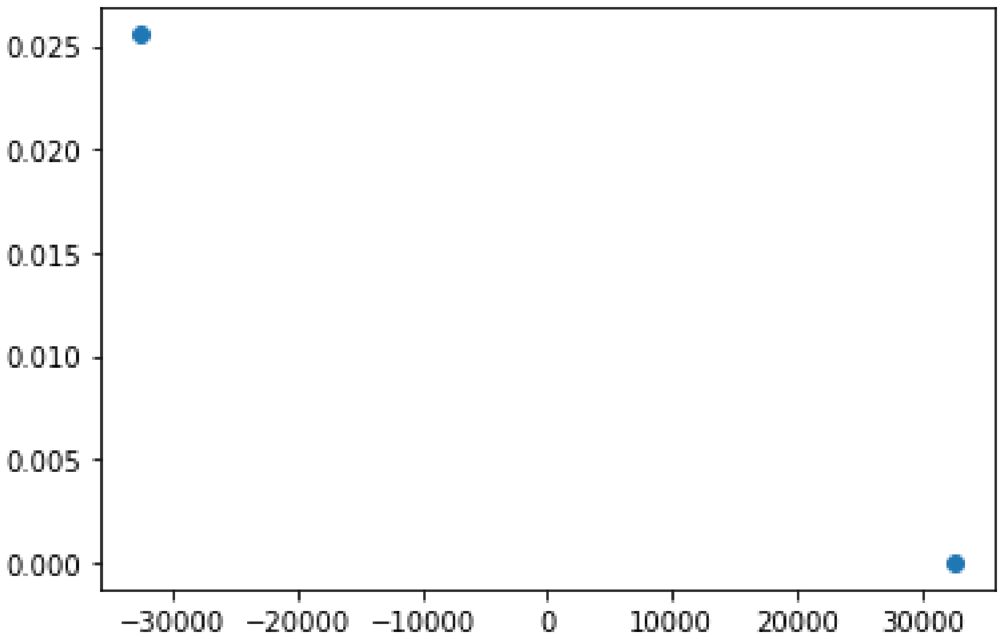
John Cornyn Joint



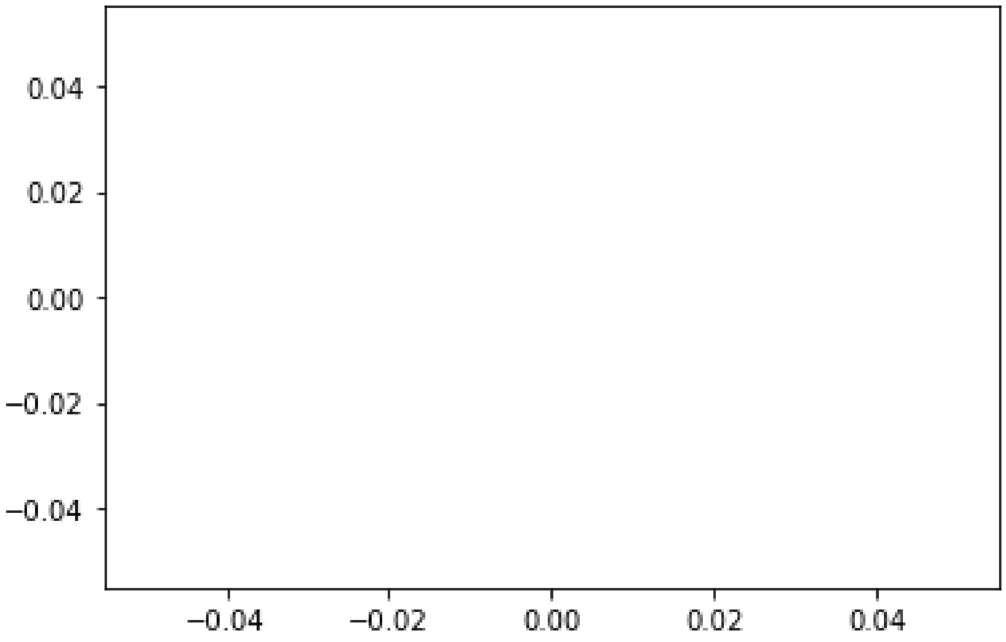
```
John Cornyn Child
```



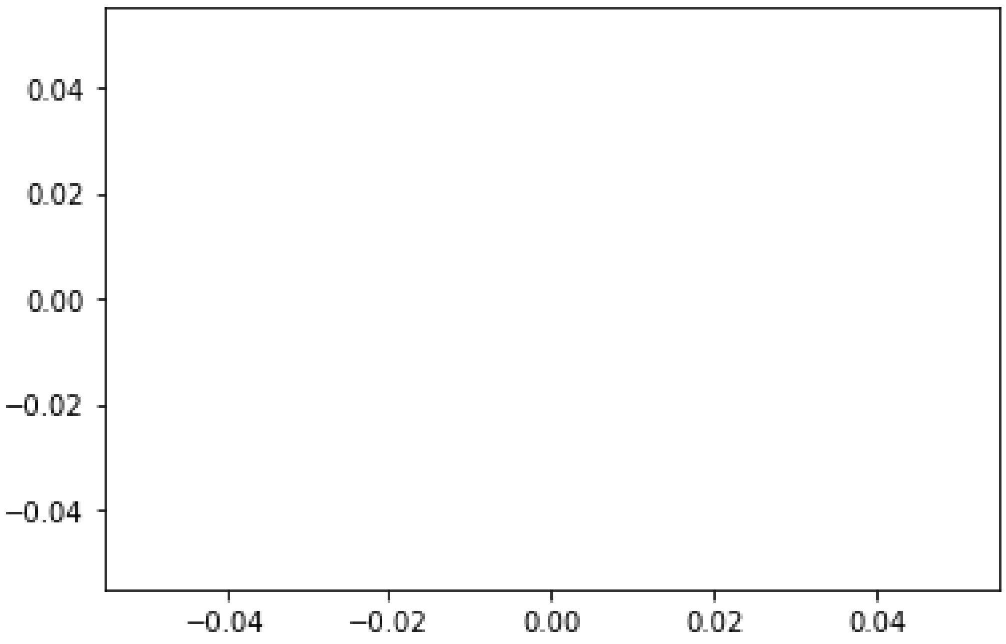
```
John Cornyn Self
```



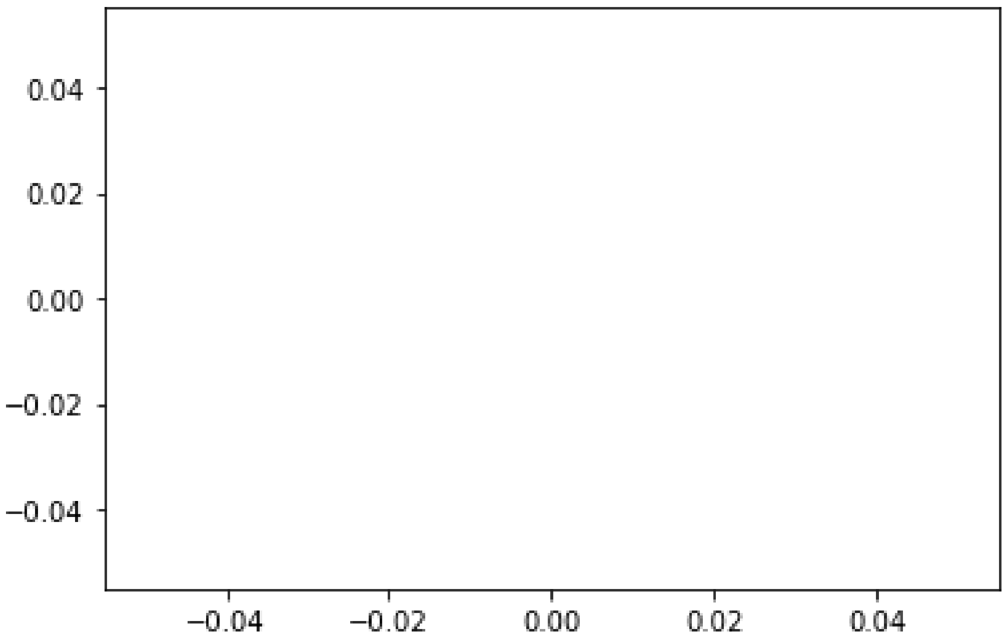
Maria Cantwell Spouse



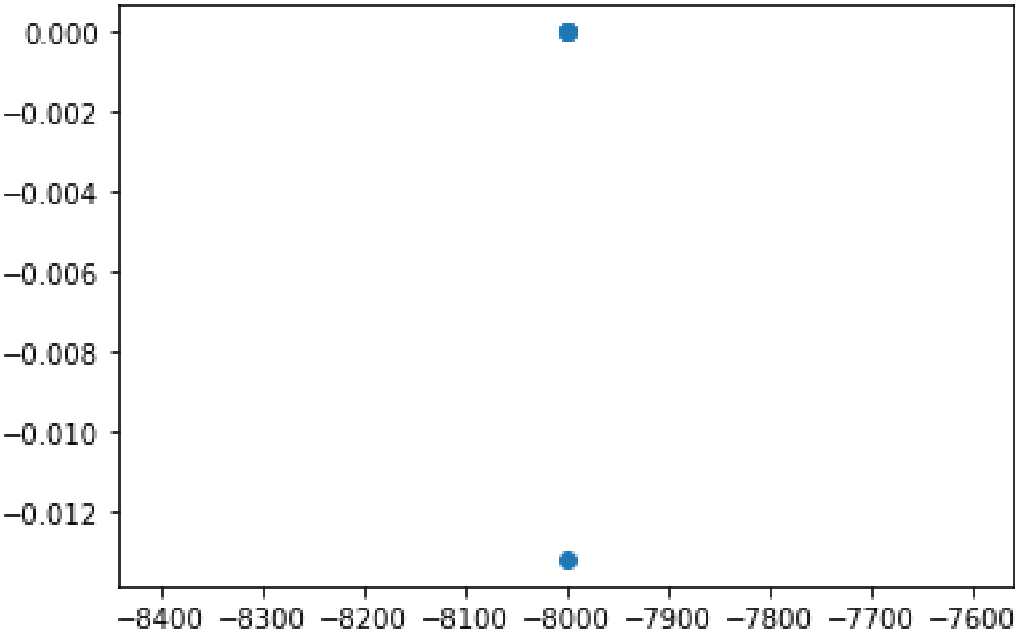
Maria Cantwell Joint



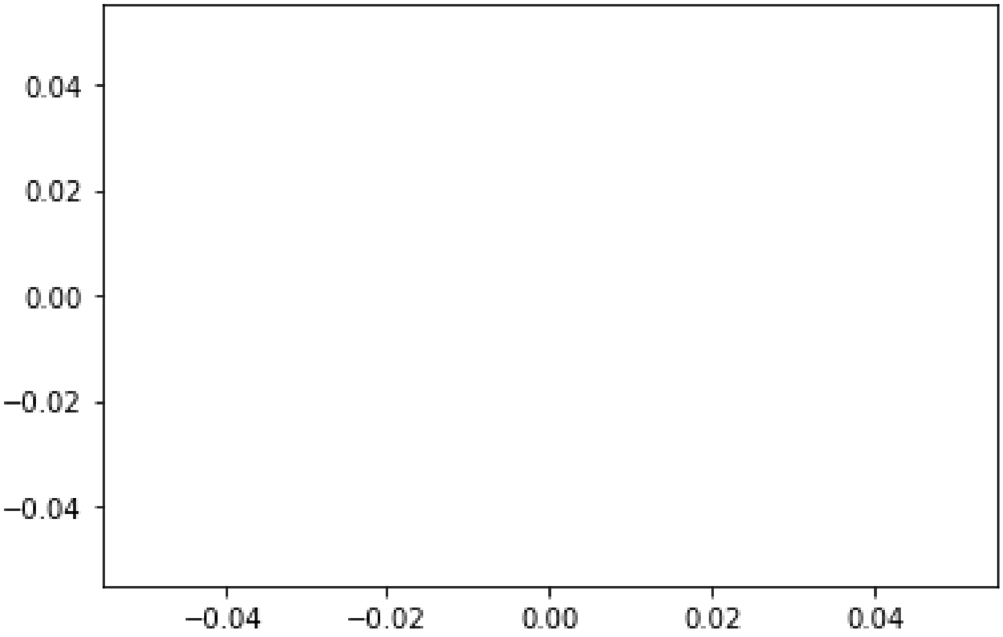
Maria Cantwell Child



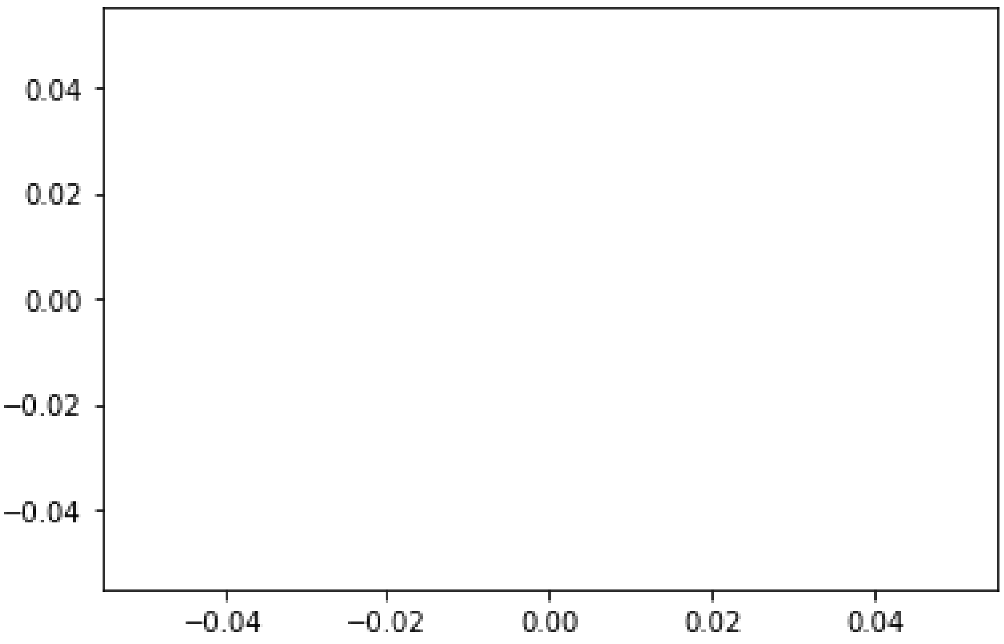
Maria Cantwell Self



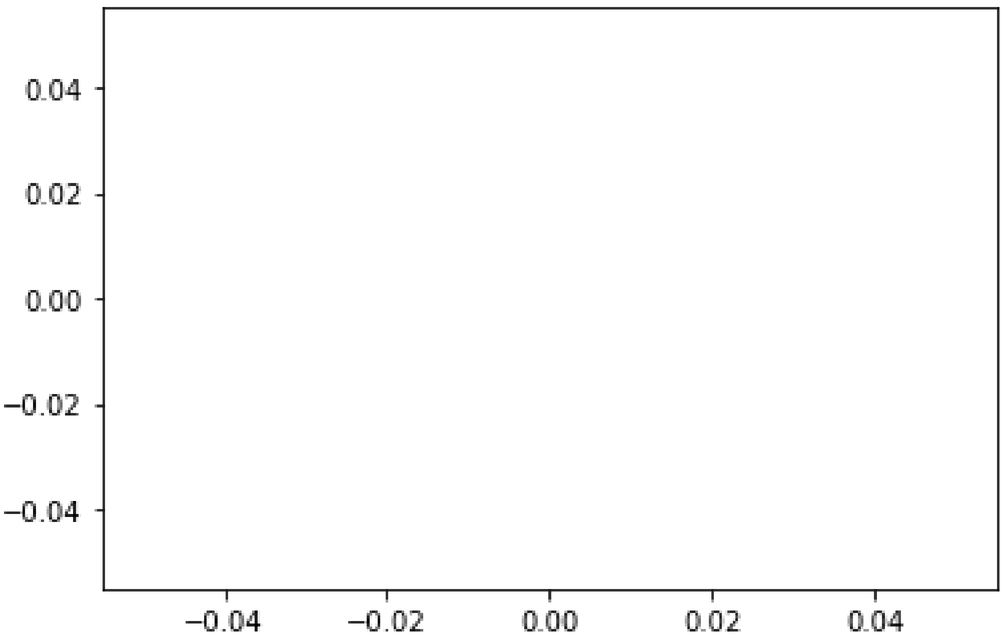
Michael B Enzi Spouse



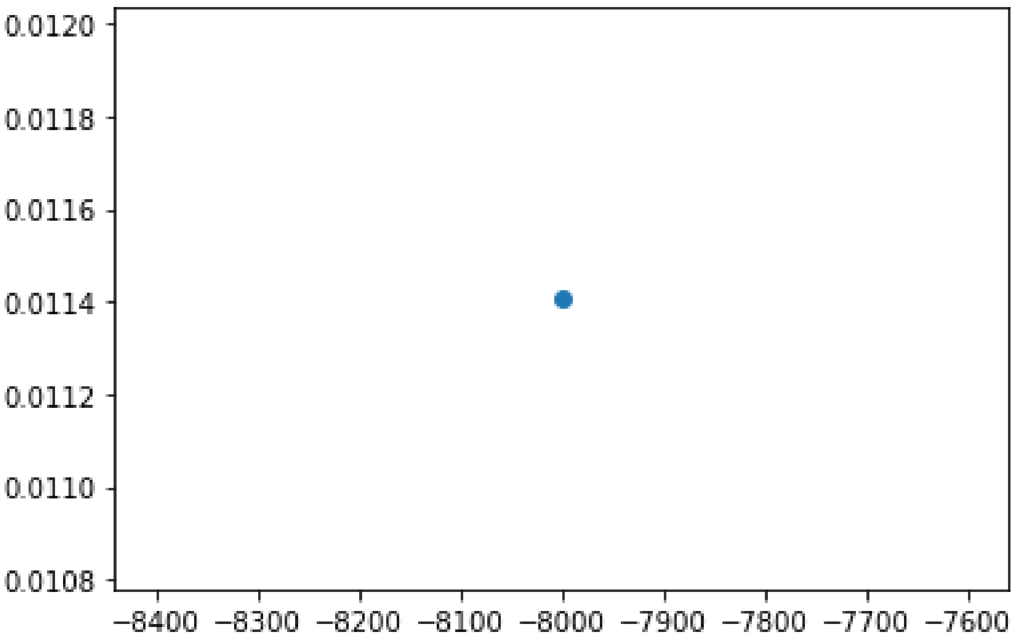
Michael B Enzi Joint



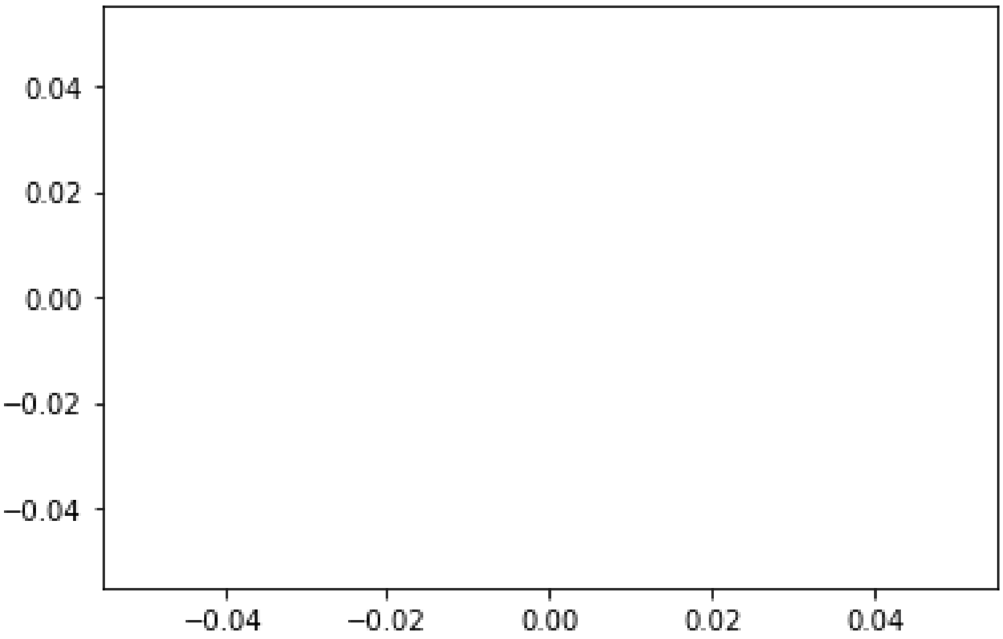
`Michael  B Enzi Child`



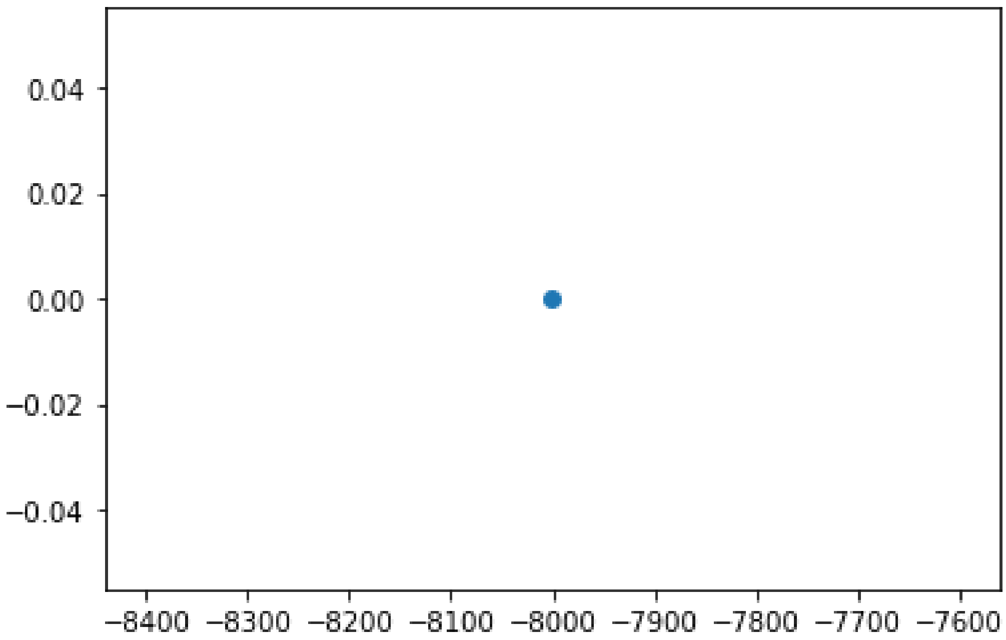
`Michael  B Enzi Self`



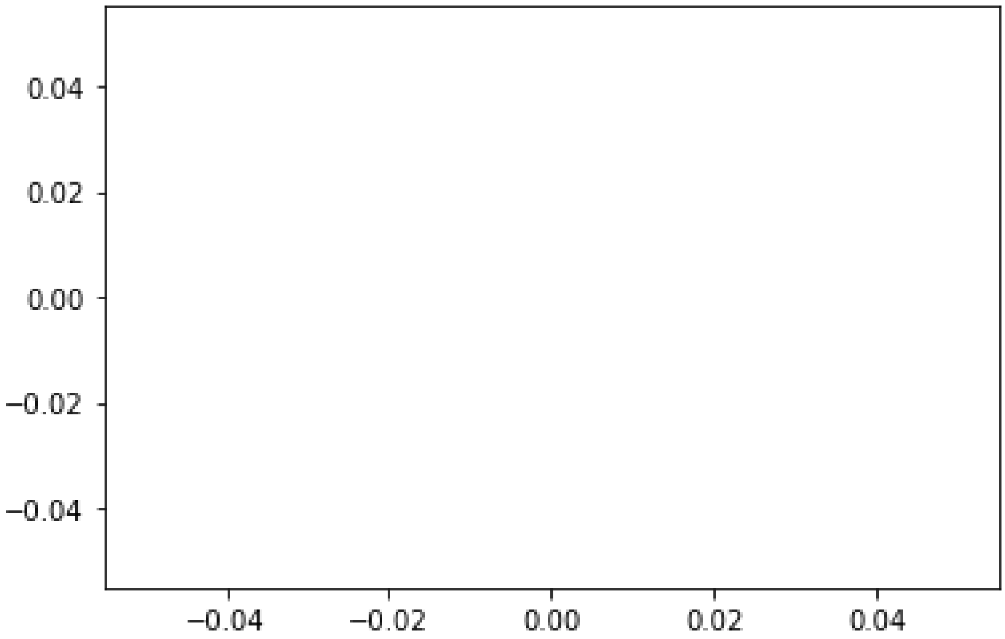
Benjamin L Cardin Spouse



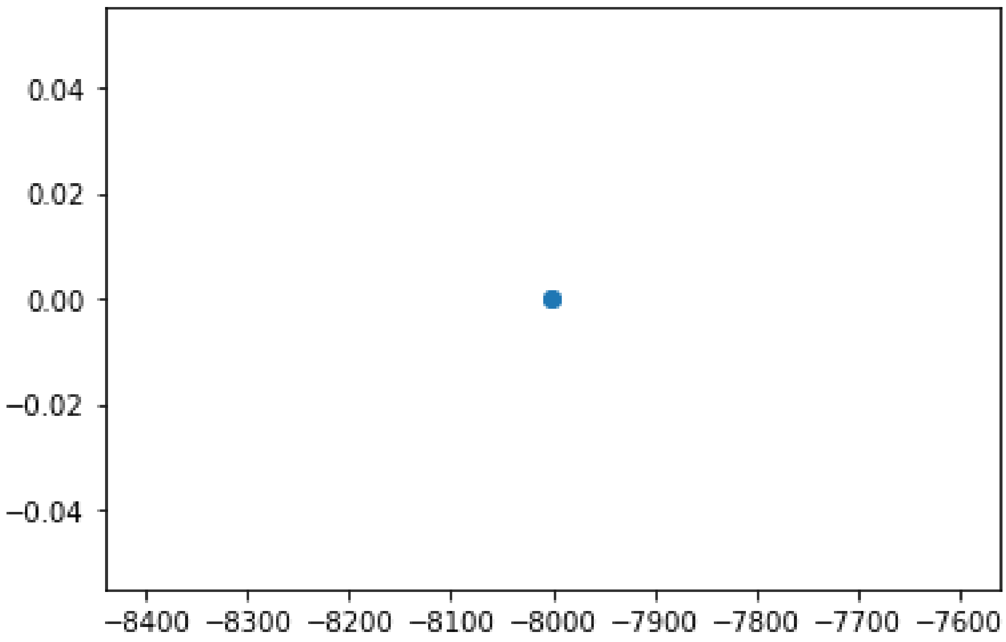
Benjamin L Cardin Joint



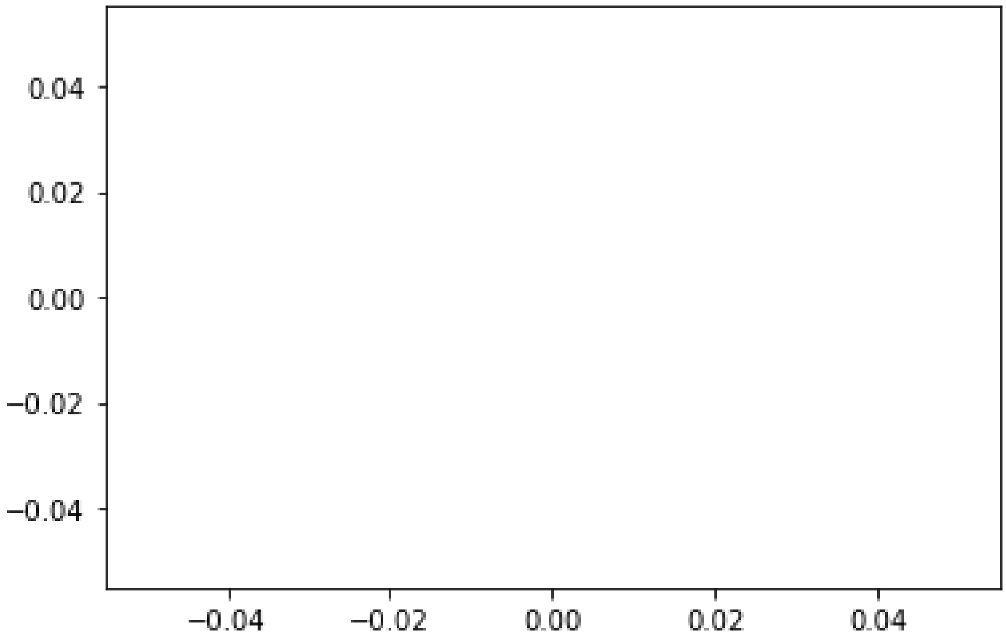
```
Benjamin L Cardin Child
```



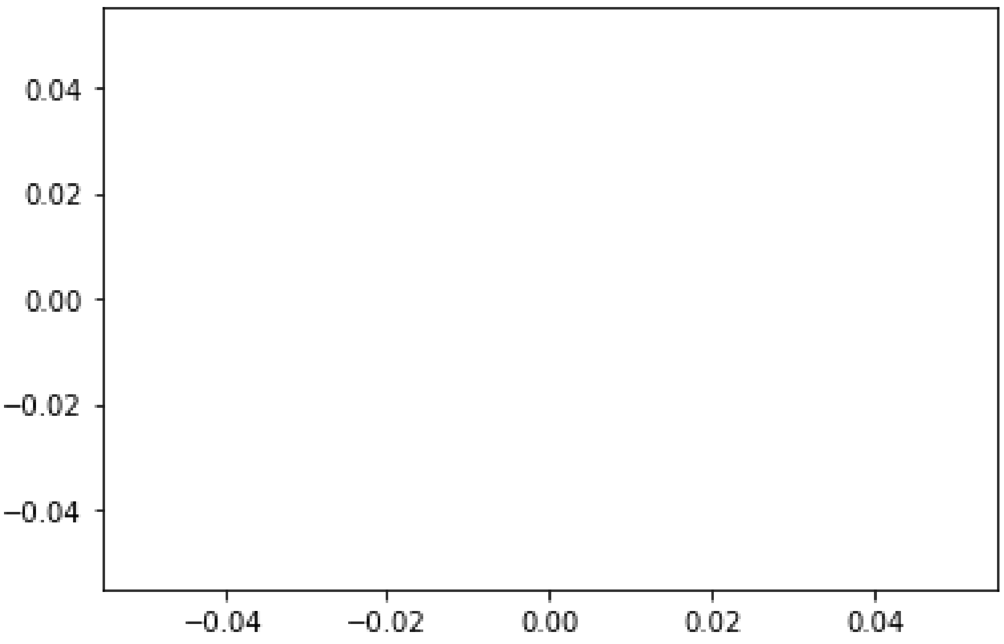
```
Benjamin L Cardin Self
```



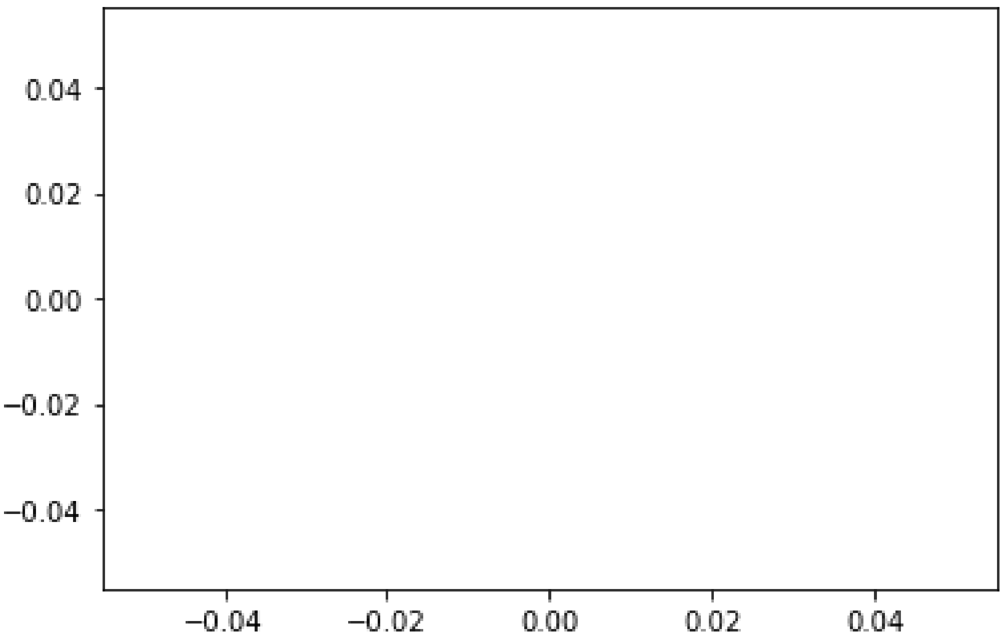
Cory A Booker Spouse



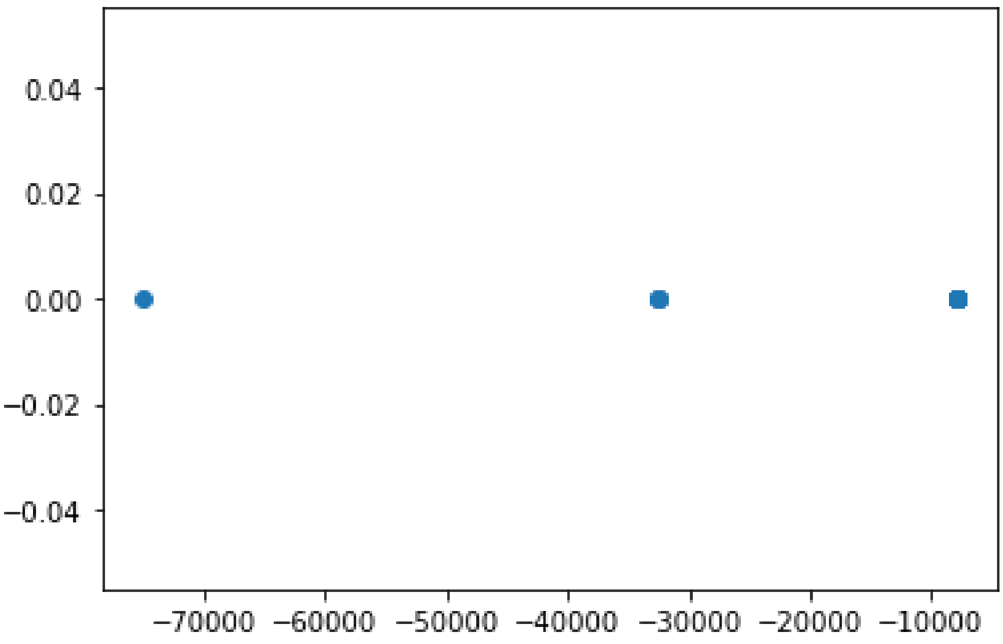
Cory A Booker Joint



`Cory A Booker Child`

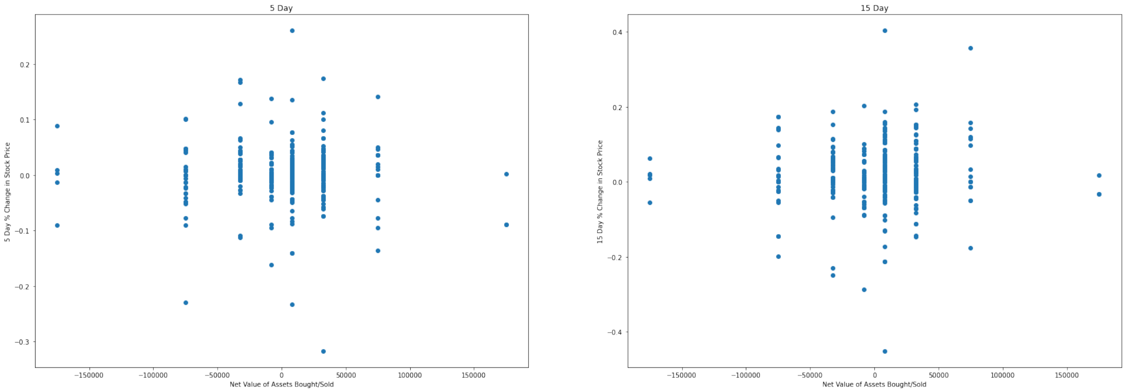


`Cory A Booker Self`

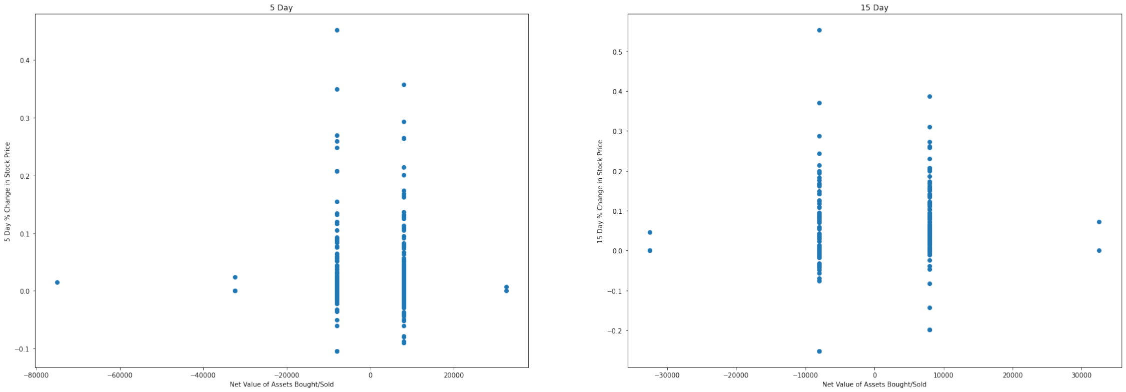


[36]:
```
for senator in stock_trades['senator'].unique():
    stock_trades_by_senator = stock_trades[stock_trades['senator'] == senator]

    fig, axs = plt.subplots(1, 2, figsize=(30, 10))
    fig.suptitle('Trade Value vs. Future Stock Price % Change for Senator ' +␣
↪senator)

    axs[0].scatter(stock_trades_by_senator['mid_amt'],␣
↪stock_trades_by_senator['5_day_pct_change'])
    axs[0].set_title('5 Day')
    axs[0].set_xlabel('Net Value of Assets Bought/Sold')
    axs[0].set_ylabel('5 Day % Change in Stock Price')
    axs[1].scatter(stock_trades_by_senator['mid_amt'],␣
↪stock_trades_by_senator['15_day_pct_change'])
    axs[1].set_title('15 Day')
    axs[1].set_xlabel('Net Value of Assets Bought/Sold')
    axs[1].set_ylabel('15 Day % Change in Stock Price')

    # plt.
    # plt.title('Trade vs. 5 Day Future Stock Price % Change for Senator ' +␣
↪senator)
    plt.show()
```

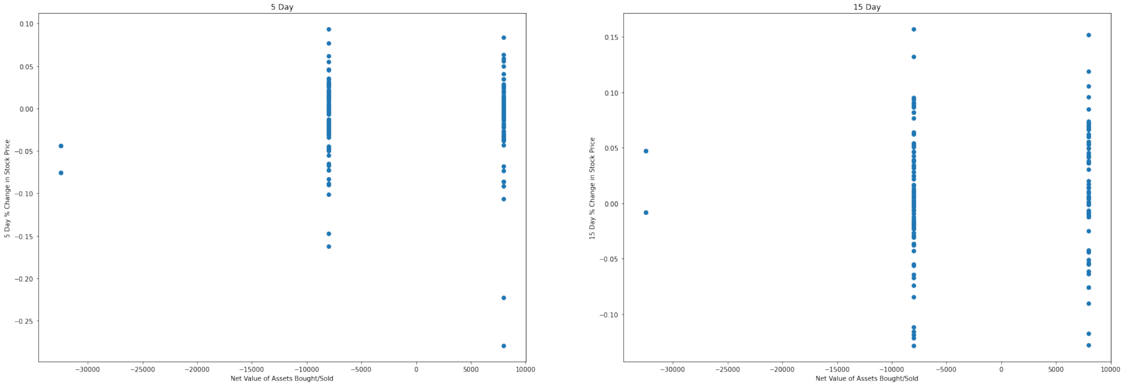
Trade Value vs. Future Stock Price % Change for Senator Pat Roberts



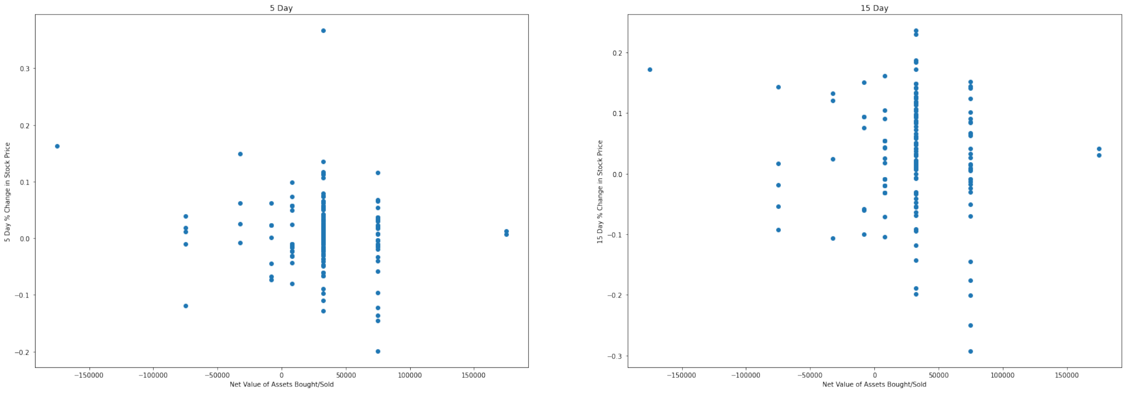
Trade Value vs. Future Stock Price % Change for Senator Thomas R Carper



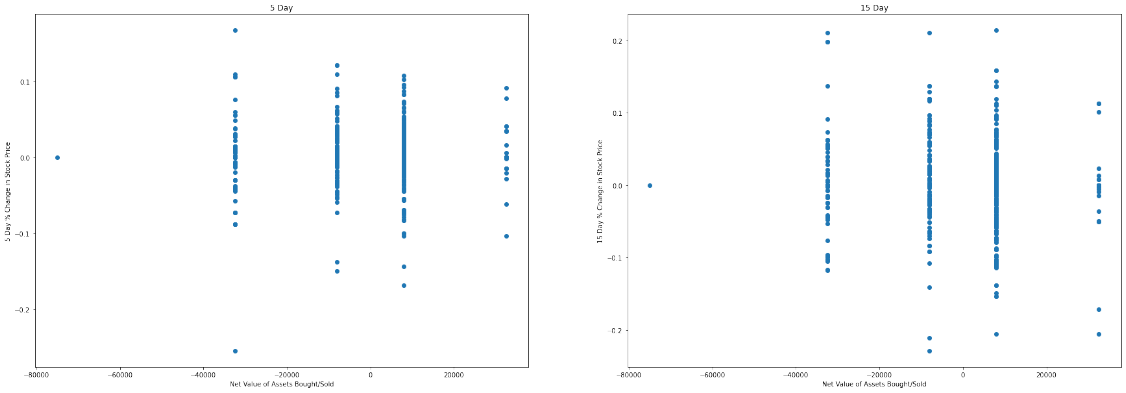
Trade Value vs. Future Stock Price % Change for Senator William Cassidy



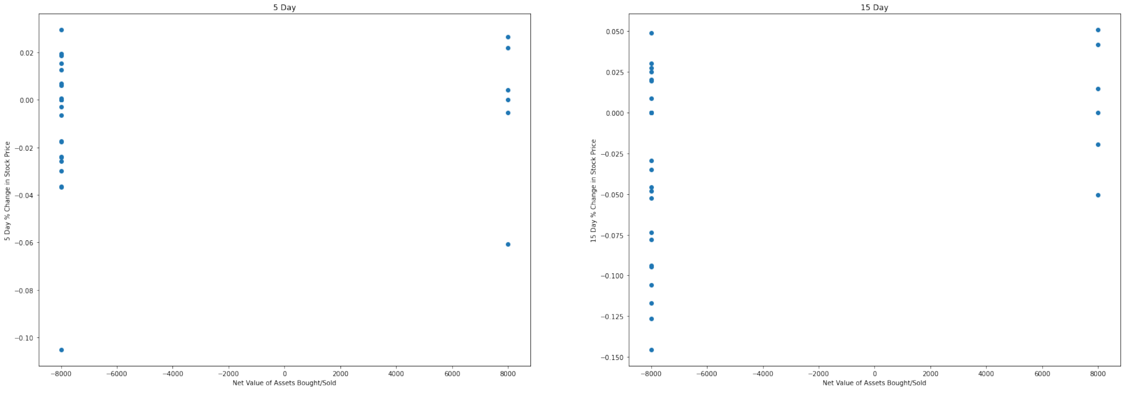
Trade Value vs. Future Stock Price % Change for Senator Ron L Wyden

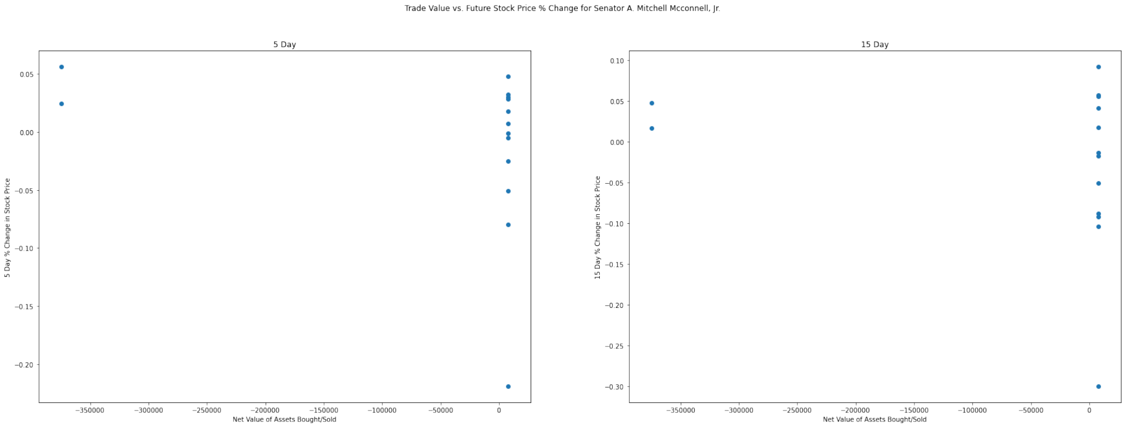


Trade Value vs. Future Stock Price % Change for Senator Sheldon Whitehouse

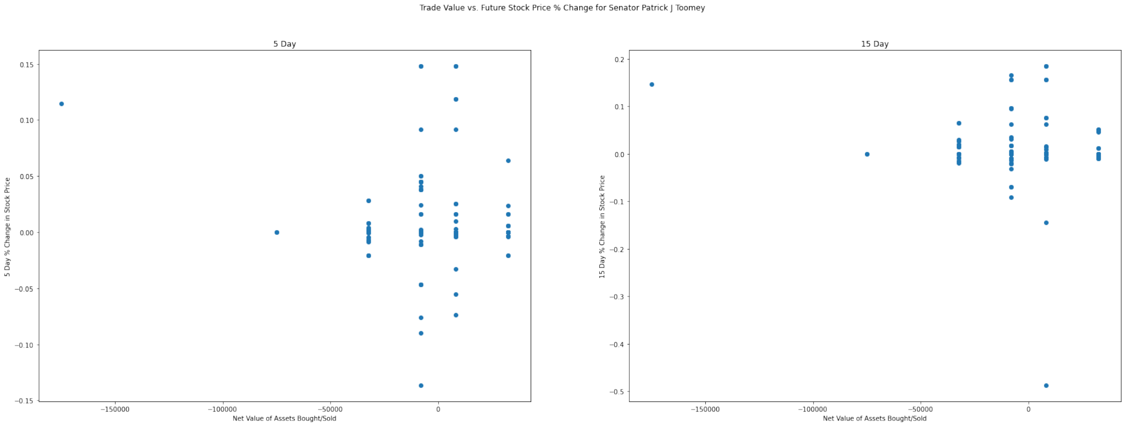


Trade Value vs. Future Stock Price % Change for Senator Angus S King, Jr.

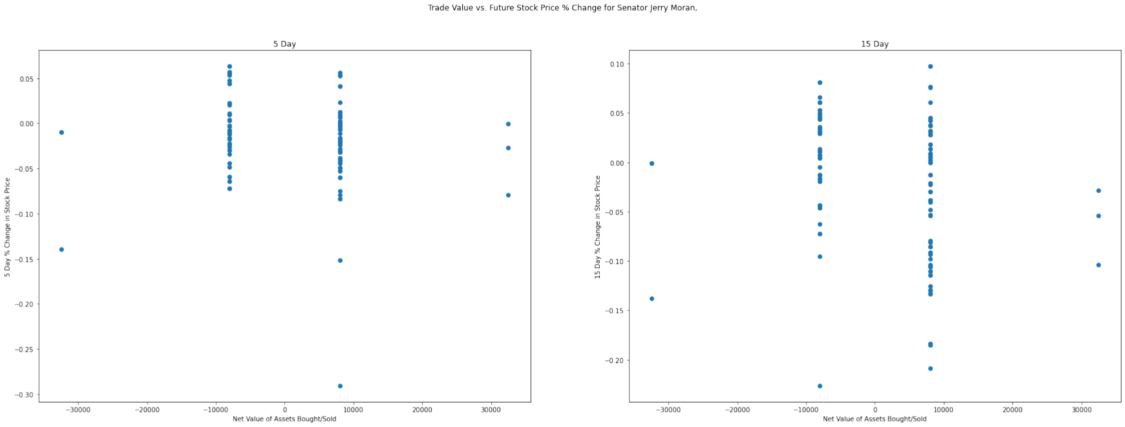




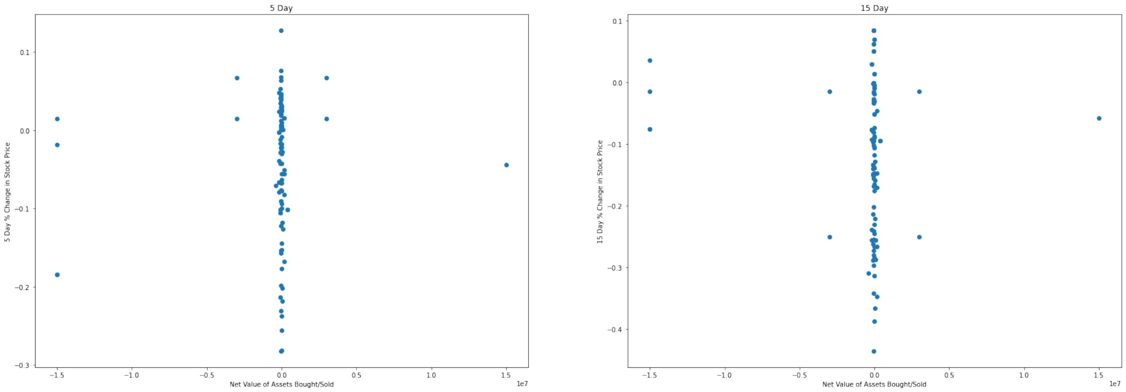
Trade Value vs. Future Stock Price % Change for Senator A. Mitchell Mcconnell, Jr.



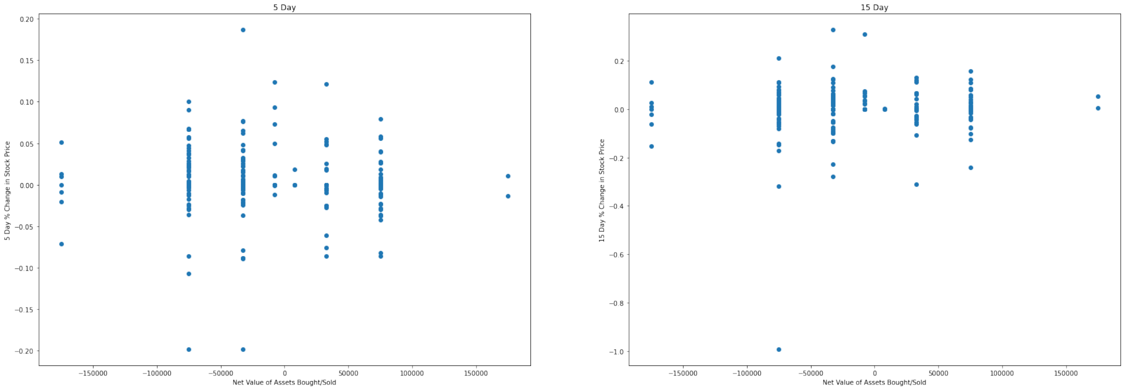
Trade Value vs. Future Stock Price % Change for Senator Patrick J Toomey



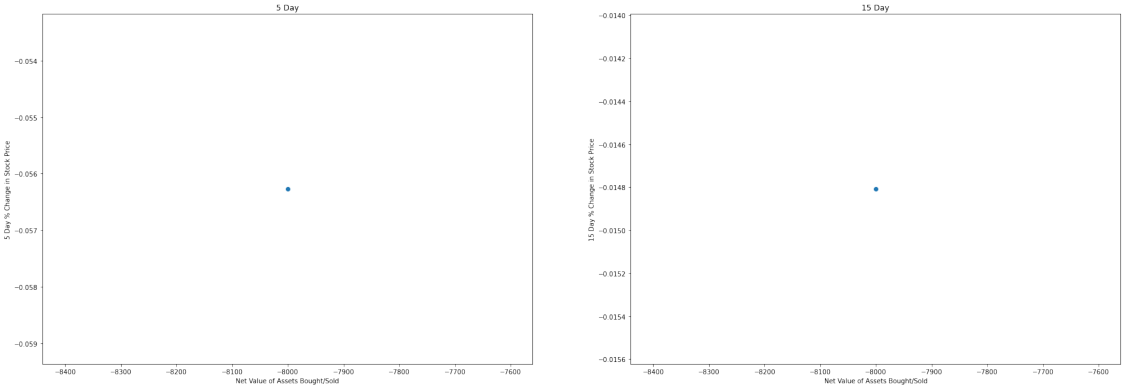
Trade Value vs. Future Stock Price % Change for Senator Jerry Moran, Trade Value vs. Future Stock Price % Change for Senator Kelly Loeffler



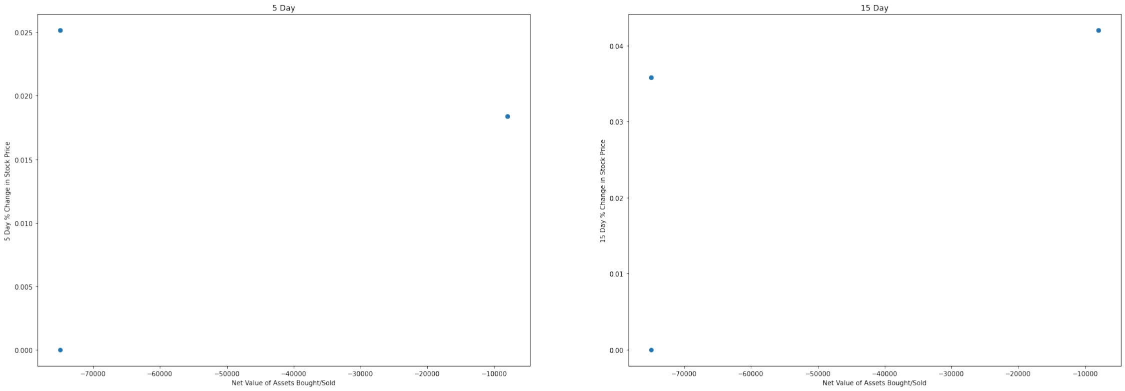
Trade Value vs. Future Stock Price % Change for Senator James M Inhofe



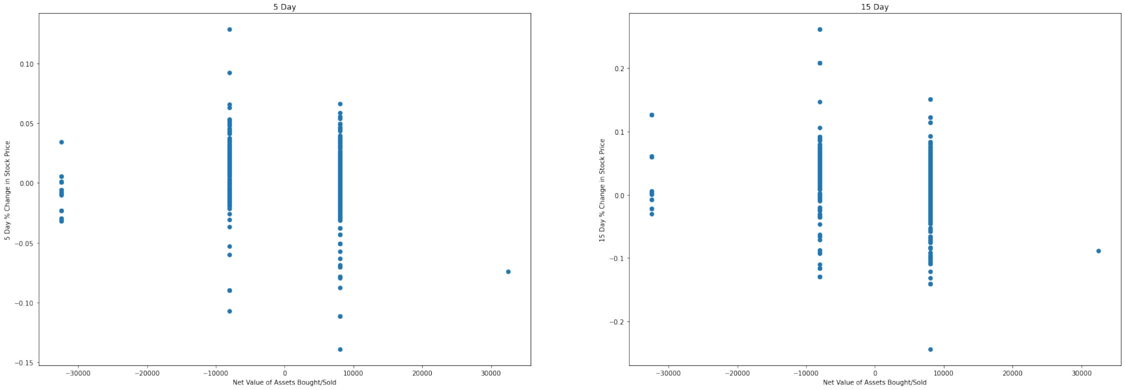
Trade Value vs. Future Stock Price % Change for Senator Timothy M Kaine



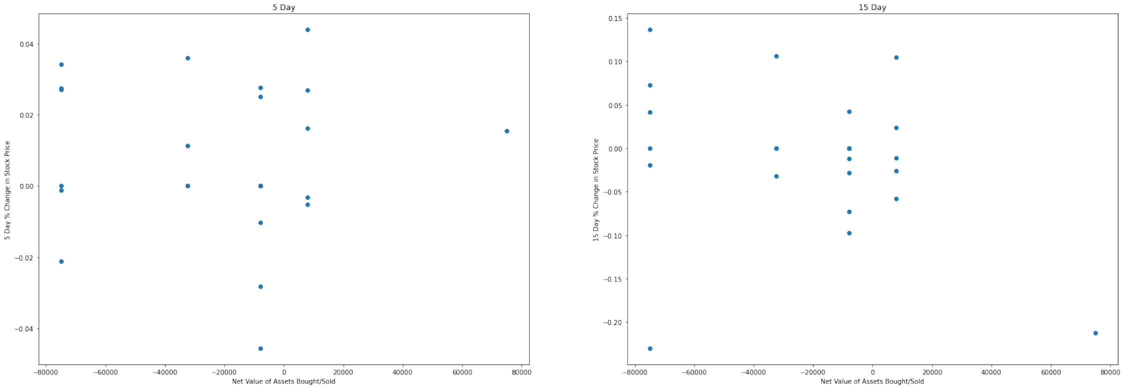
Trade Value vs. Future Stock Price % Change for Senator Roy Blunt

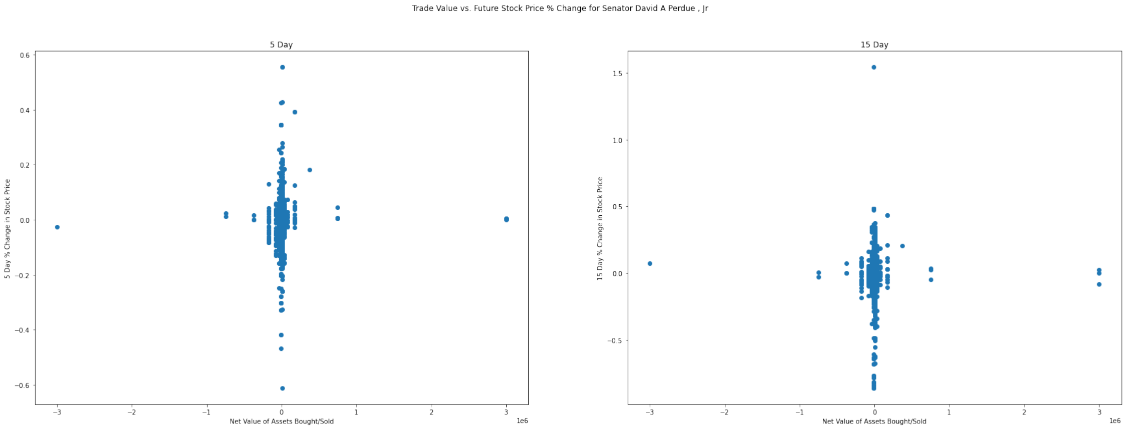


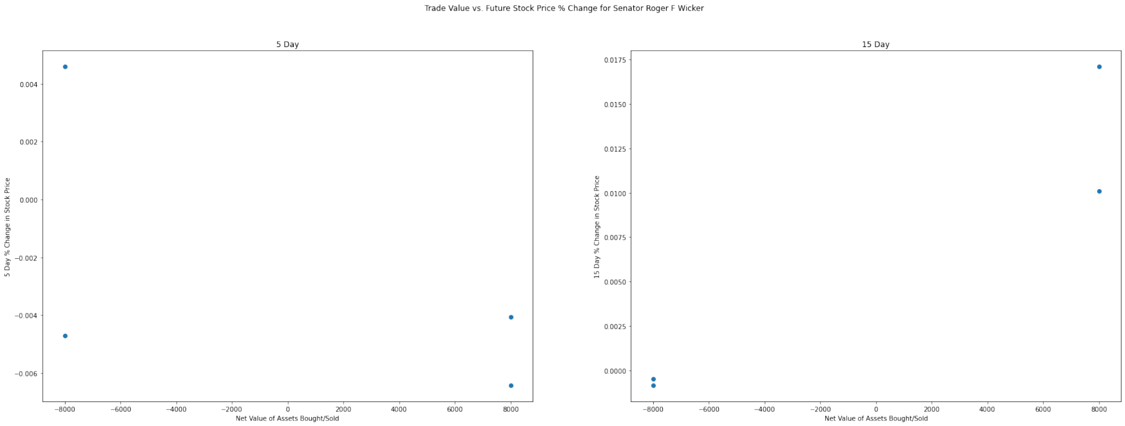
Trade Value vs. Future Stock Price % Change for Senator Shelley M Capito

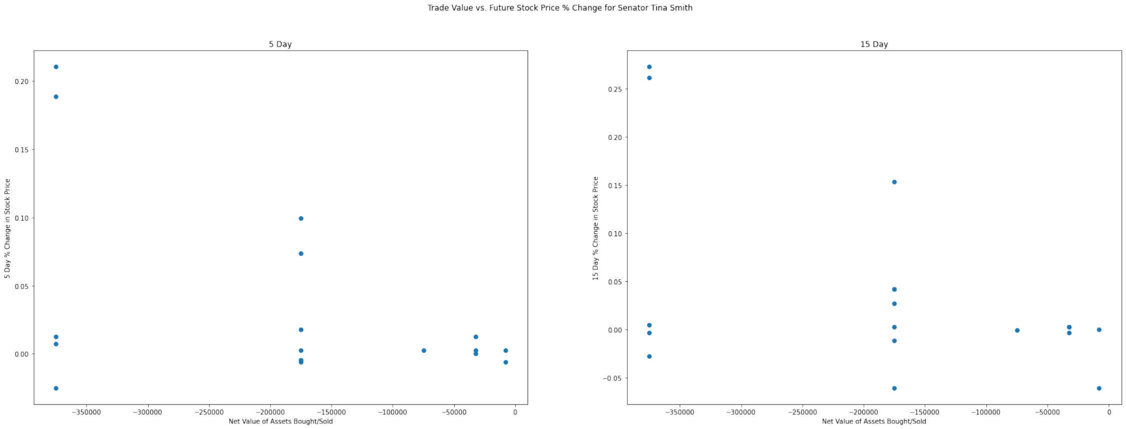


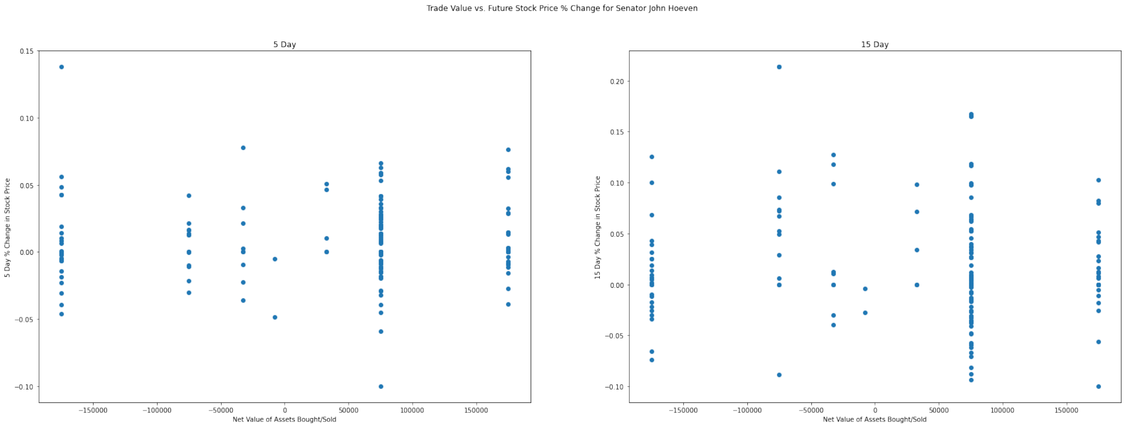
Trade Value vs. Future Stock Price % Change for Senator Daniel S Sullivan

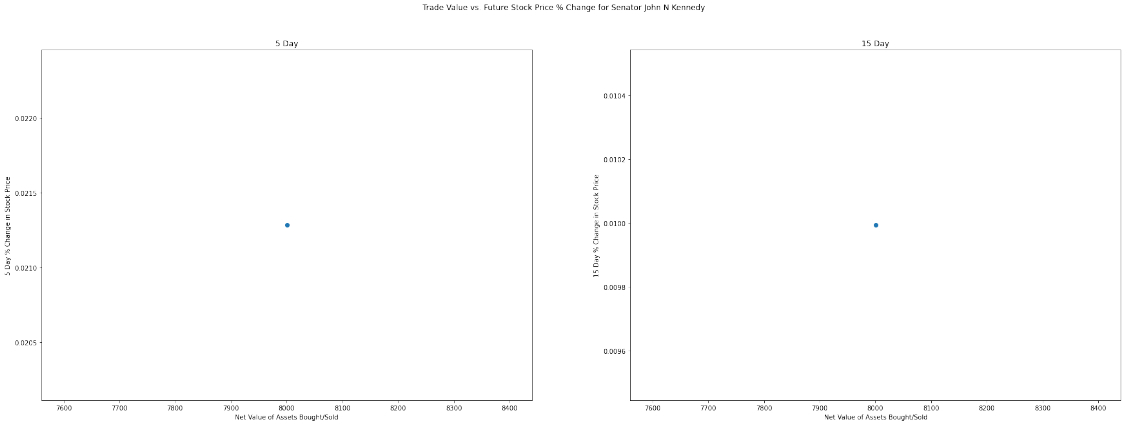


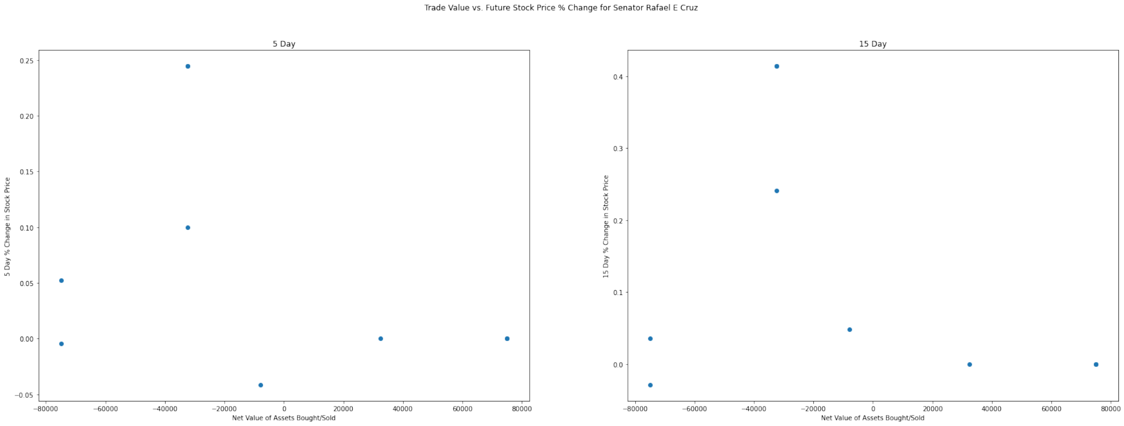

Trade Value vs. Future Stock Price % Change for Senator David A Perdue , Jr
5 Day
5 Day % Change in Stock Price
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−3
−2
−1
0
1
2
3
1e6
Net Value of Assets Bought/Sold
15 Day
15 Day % Change in Stock Price
1.5
1.0
0.5
0.0
−0.5
−3
−2
−1
0
1
2
3
1e6
Net Value of Assets Bought/Sold


Trade Value vs. Future Stock Price % Change for Senator Roger F Wicker
5 Day
5 Day % Change in Stock Price
0.004
0.002
0.000
−0.002
−0.004
−0.006
−8000
−6000
−4000
−2000
0
2000
4000
6000
8000
Net Value of Assets Bought/Sold
15 Day
15 Day % Change in Stock Price
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
−8000
−6000
−4000
−2000
0
2000
4000
6000
8000
Net Value of Assets Bought/Sold

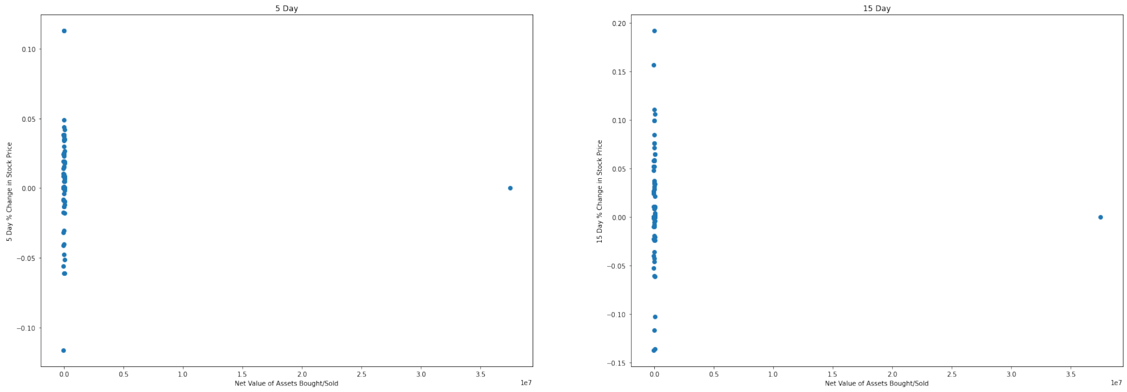

Trade Value vs. Future Stock Price % Change for Senator Tina Smith
5 Day
5 Day % Change in Stock Price
0.20
0.15
0.10
0.05
0.00
−350000
−300000
−250000
−200000
−150000
−100000
−50000
0
Net Value of Assets Bought/Sold
15 Day
15 Day % Change in Stock Price
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−350000
−300000
−250000
−200000
−150000
−100000
−50000
0
Net Value of Assets Bought/Sold


Trade Value vs. Future Stock Price % Change for Senator John Hoeven
5 Day
15 Day
5 Day % Change in Stock Price
15 Day % Change in Stock Price
Net Value of Assets Bought/Sold
Net Value of Assets Bought/Sold
0.15
0.10
0.05
0.00
−0.05
−0.10
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−150000
−100000
−50000
0
50000
100000
150000

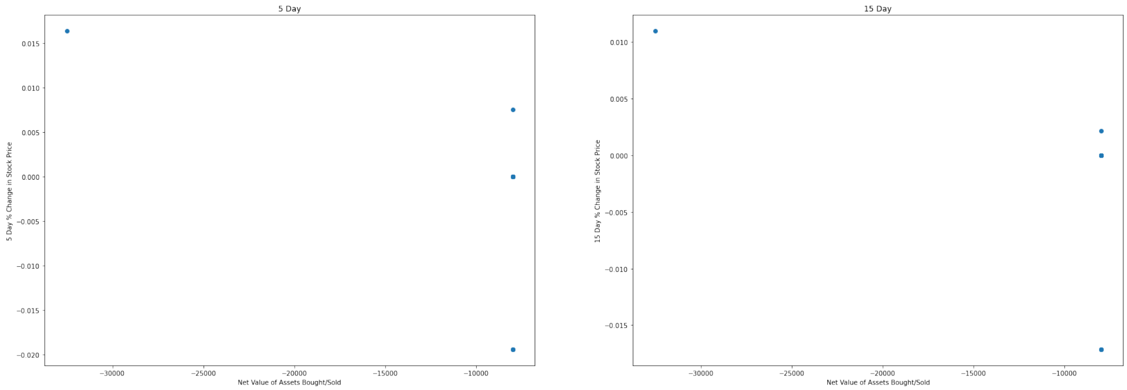

Trade Value vs. Future Stock Price % Change for Senator John N Kennedy
5 Day
15 Day
5 Day % Change in Stock Price
15 Day % Change in Stock Price
Net Value of Assets Bought/Sold
Net Value of Assets Bought/Sold
0.0220
0.0215
0.0210
0.0205
0.0104
0.0102
0.0100
0.0098
0.0096
7600
7700
7800
7900
8000
8100
8200
8300
8400


Trade Value vs. Future Stock Price % Change for Senator Rafael E Cruz
5 Day
15 Day
5 Day % Change in Stock Price
15 Day % Change in Stock Price
Net Value of Assets Bought/Sold
Net Value of Assets Bought/Sold
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
0.4
0.3
0.2
0.1
0.0
−80000
−60000
−40000
−20000
0
20000
40000
60000
80000

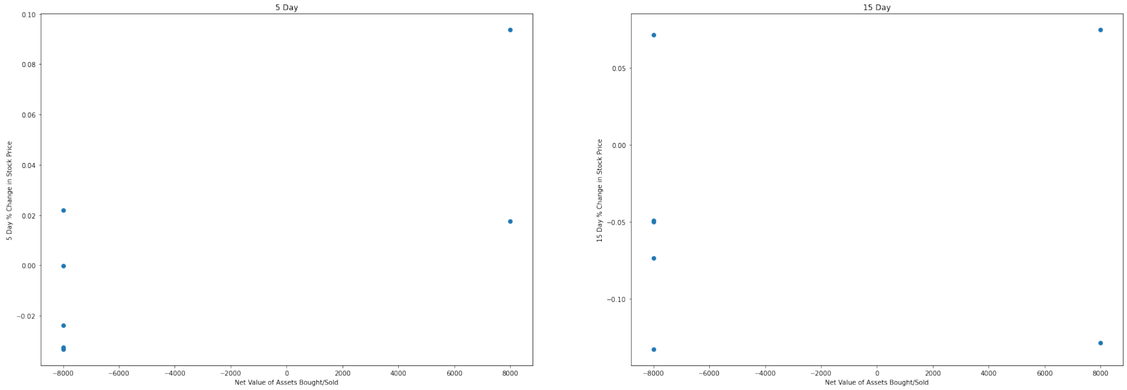
Trade Value vs. Future Stock Price % Change for Senator Susan M Collins

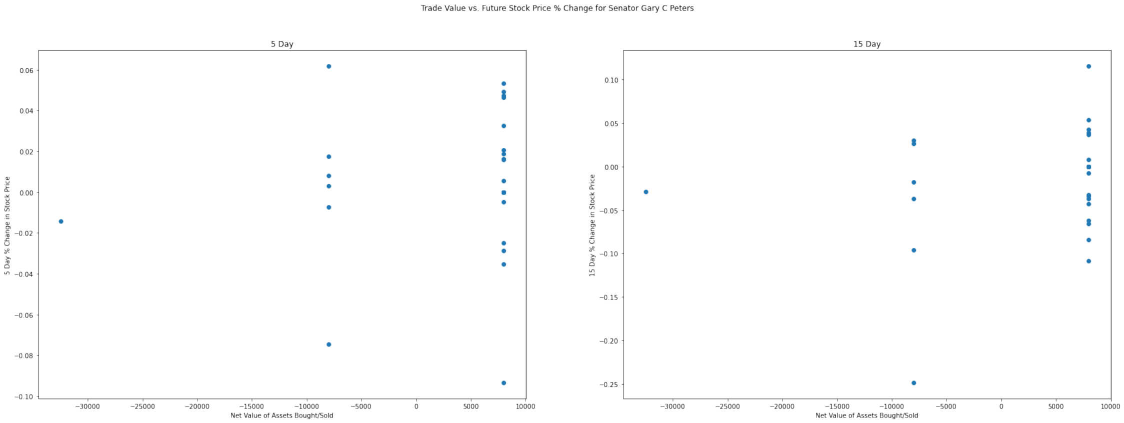


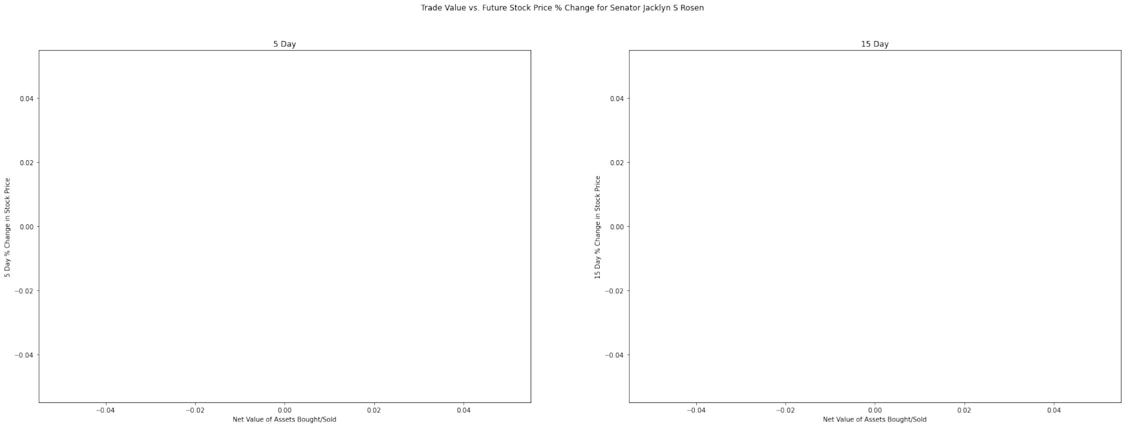
Trade Value vs. Future Stock Price % Change for Senator Thomas R Tillis

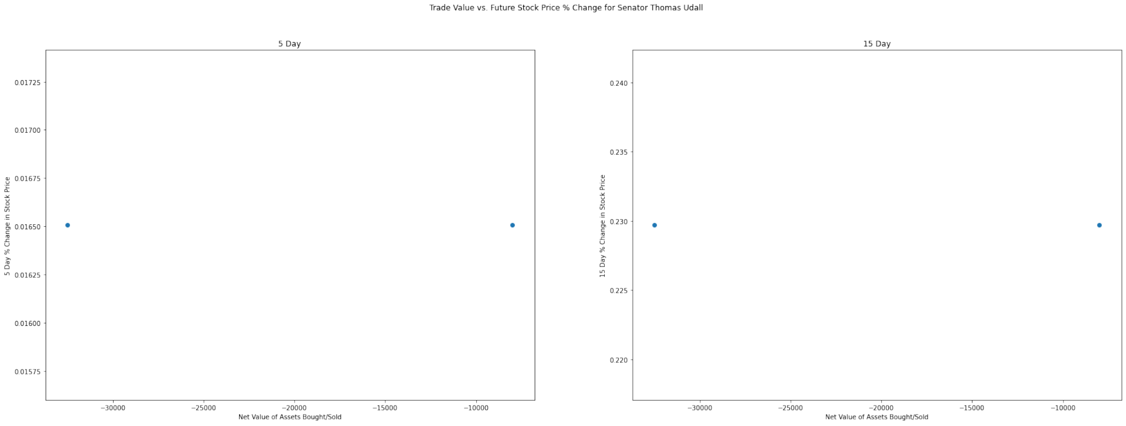


Trade Value vs. Future Stock Price % Change for Senator Christopher A Coons

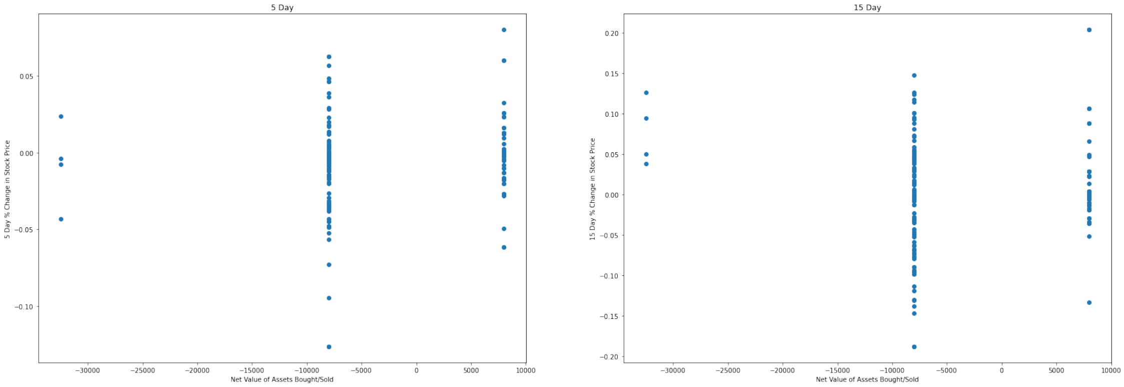



Trade Value vs. Future Stock Price % Change for Senator Gary C Peters
5 Day
15 Day
5 Day % Change in Stock Price
15 Day % Change in Stock Price
Net Value of Assets Bought/Sold
Net Value of Assets Bought/Sold
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−30000
−25000
−20000
−15000
−10000
−5000
0
5000
10000

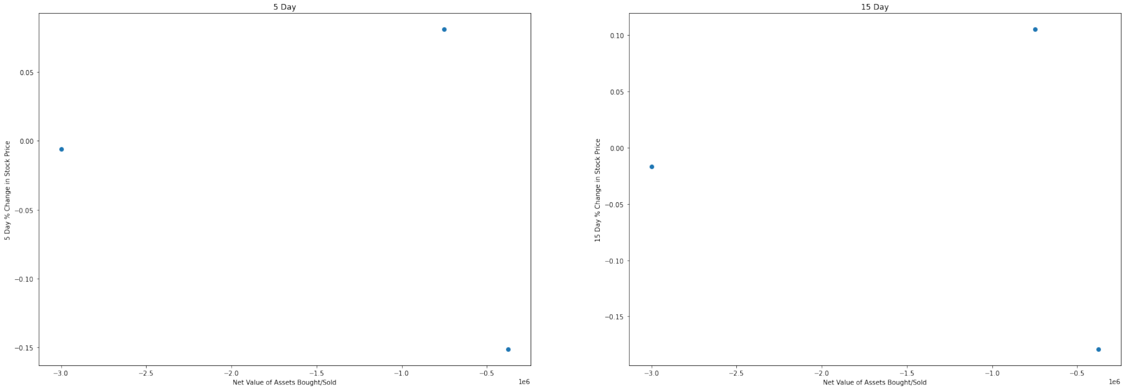

Trade Value vs. Future Stock Price % Change for Senator Jacklyn S Rosen
5 Day
15 Day
5 Day % Change in Stock Price
15 Day % Change in Stock Price
Net Value of Assets Bought/Sold
Net Value of Assets Bought/Sold
0.04
0.02
0.00
−0.02
−0.04
−0.04
−0.02
0.00
0.02
0.04


Trade Value vs. Future Stock Price % Change for Senator Thomas Udall
5 Day
15 Day
5 Day % Change in Stock Price
15 Day % Change in Stock Price
Net Value of Assets Bought/Sold
Net Value of Assets Bought/Sold
0.01725
0.01700
0.01675
0.01650
0.01625
0.01600
0.01575
0.240
0.235
0.230
0.225
0.220
−30000
−25000
−20000
−15000
−10000

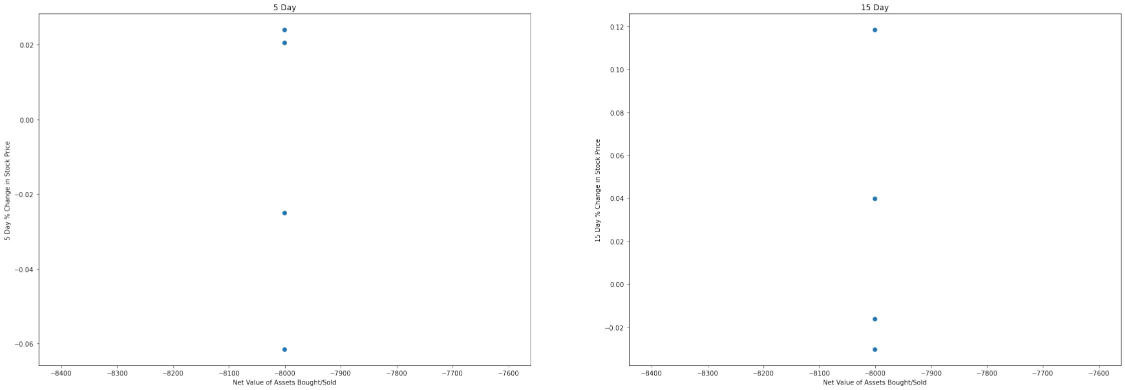
Trade Value vs. Future Stock Price % Change for Senator John F Reed



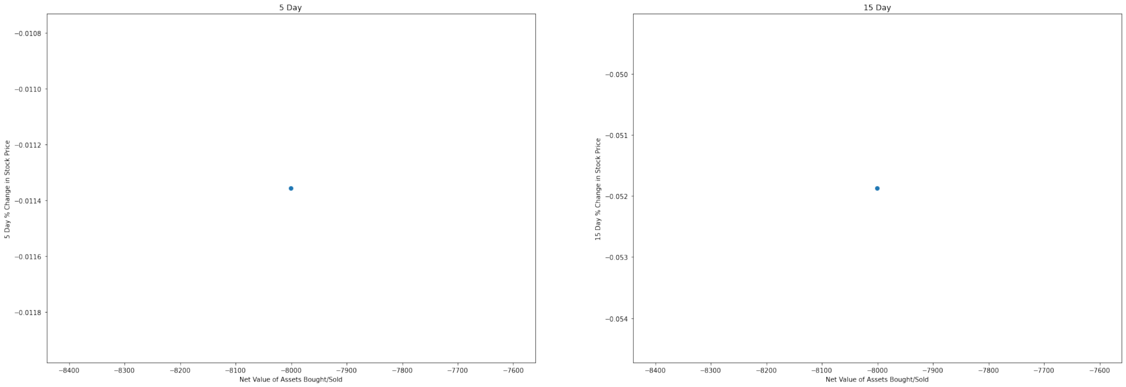
Trade Value vs. Future Stock Price % Change for Senator Mark R Warner



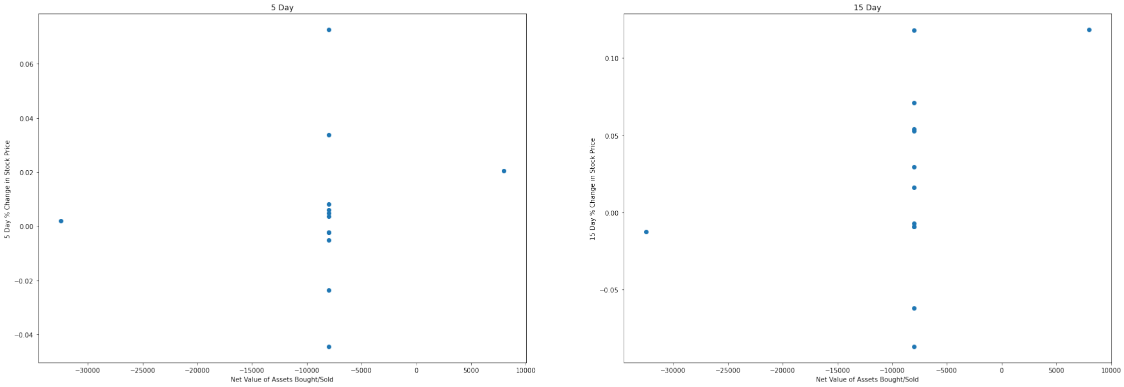
Trade Value vs. Future Stock Price % Change for Senator Ladda Tammy Duckworth



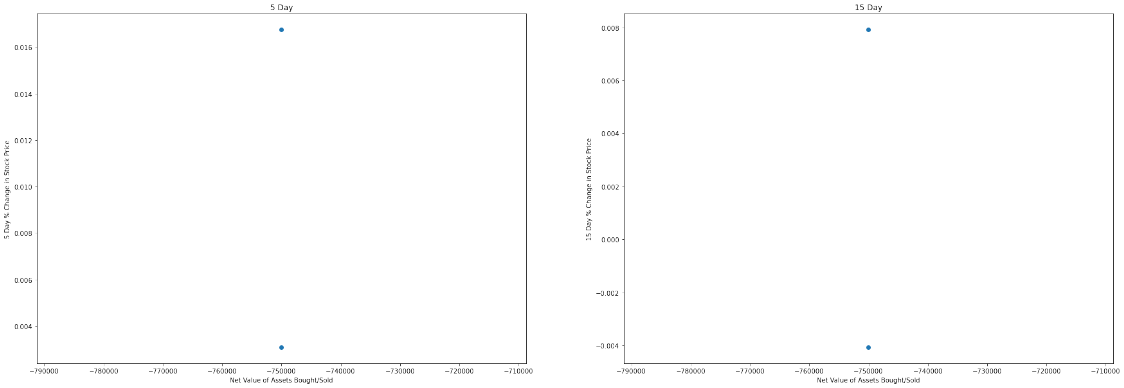
Trade Value vs. Future Stock Price % Change for Senator Robert P Casey, Jr.

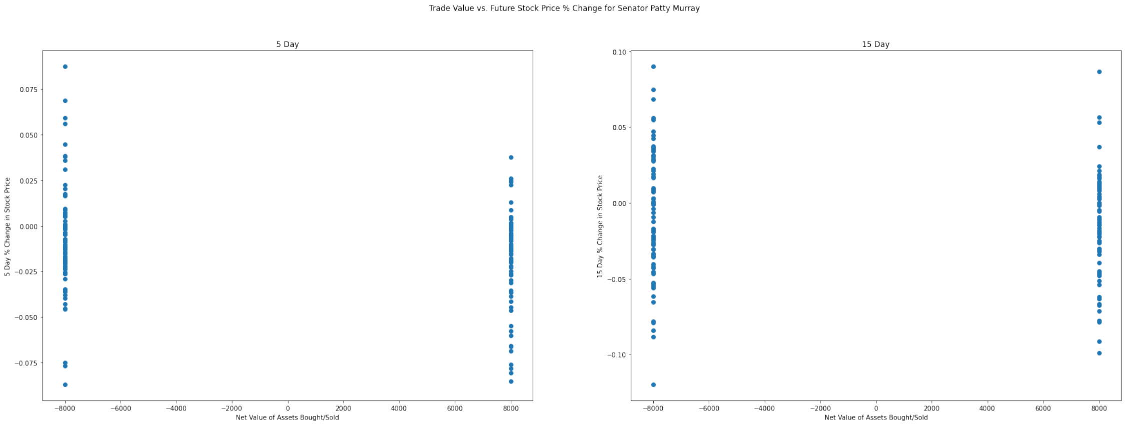


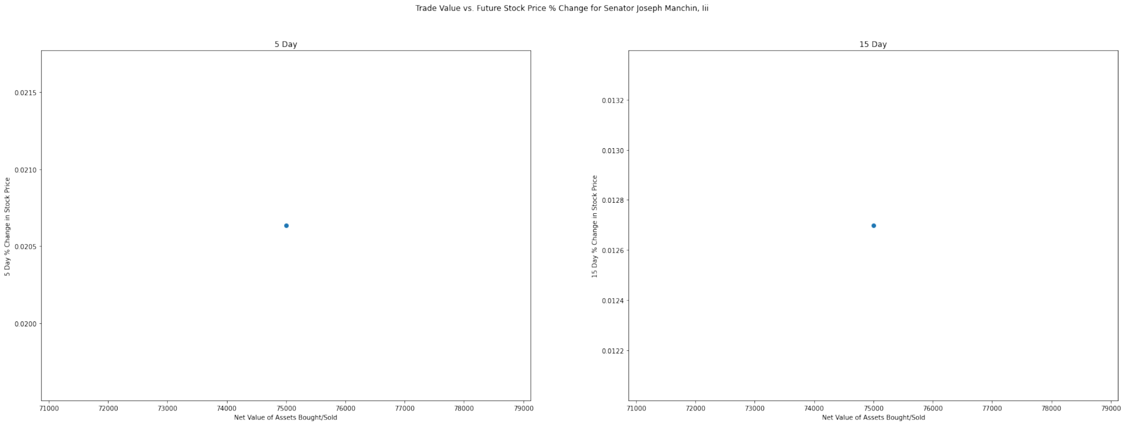
Trade Value vs. Future Stock Price % Change for Senator Tammy Duckworth

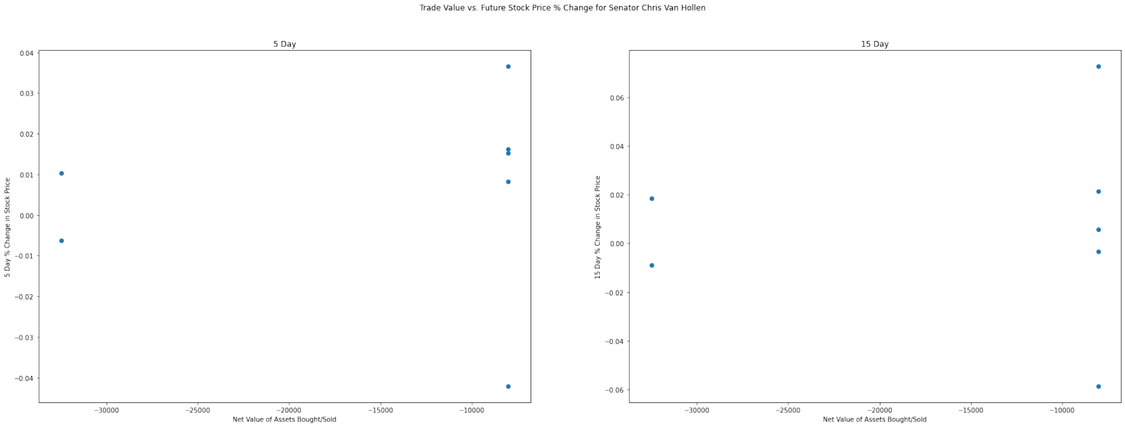


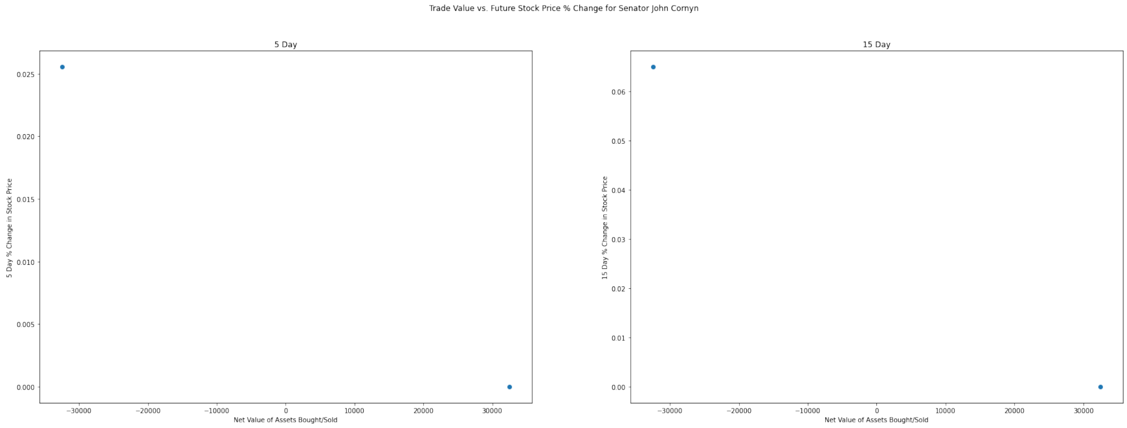
Trade Value vs. Future Stock Price % Change for Senator Michael F Bennet




Trade Value vs. Future Stock Price % Change for Senator Patty Murray
5 Day
5 Day % Change in Stock Price
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−8000
−6000
−4000
−2000
0
2000
4000
6000
8000
Net Value of Assets Bought/Sold
15 Day
15 Day % Change in Stock Price
0.10
0.05
0.00
−0.05
−0.10
−8000
−6000
−4000
−2000
0
2000
4000
6000
8000
Net Value of Assets Bought/Sold


Trade Value vs. Future Stock Price % Change for Senator Joseph Manchin, Iii
5 Day
5 Day % Change in Stock Price
0.0215
0.0210
0.0205
0.0200
71000
72000
73000
74000
75000
76000
77000
78000
79000
Net Value of Assets Bought/Sold
15 Day
15 Day % Change in Stock Price
0.0132
0.0130
0.0128
0.0126
0.0124
0.0122
71000
72000
73000
74000
75000
76000
77000
78000
79000
Net Value of Assets Bought/Sold

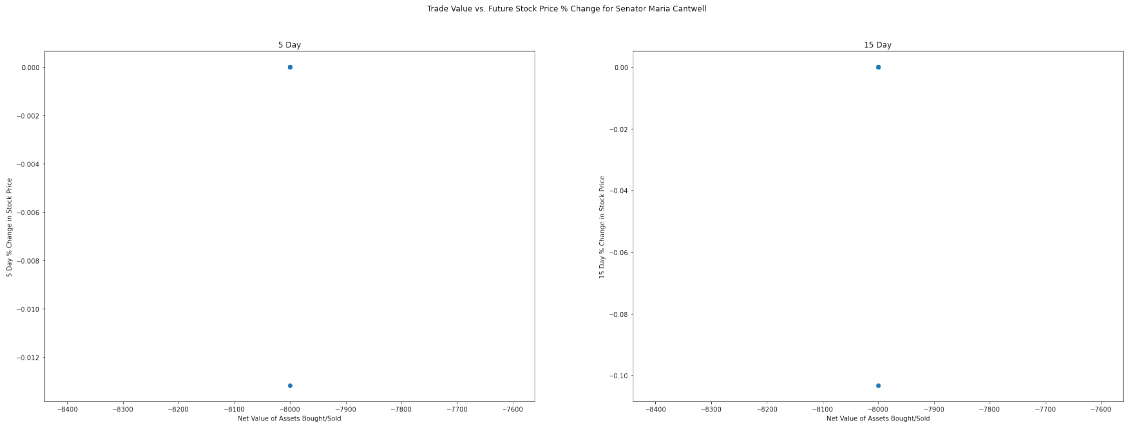

Trade Value vs. Future Stock Price % Change for Senator Chris Van Hollen
5 Day
5 Day % Change in Stock Price
0.04
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.04
−30000
−25000
−20000
−15000
−10000
Net Value of Assets Bought/Sold
15 Day
15 Day % Change in Stock Price
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−30000
−25000
−20000
−15000
−10000
Net Value of Assets Bought/Sold



Trade Value vs. Future Stock Price % Change for Senator John Cornyn

5 Day

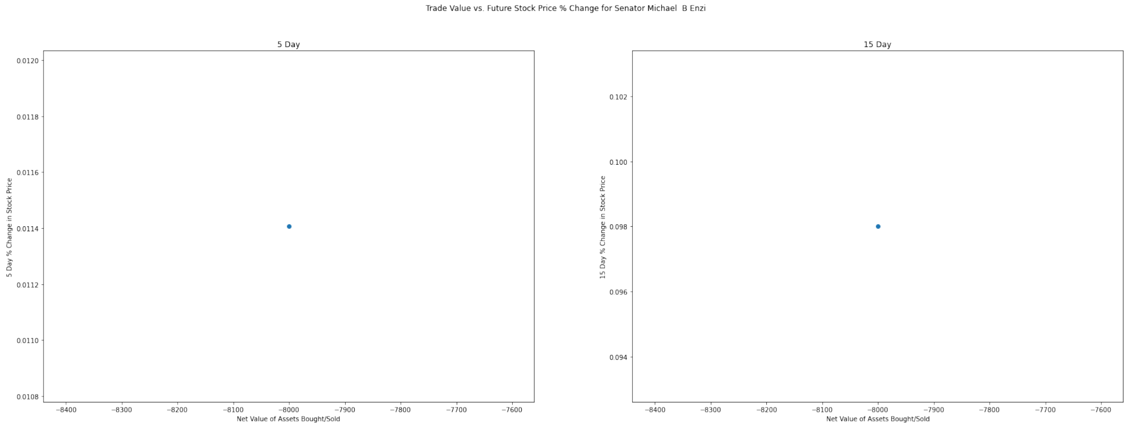
15 Day



Trade Value vs. Future Stock Price % Change for Senator Maria Cantwell

5 Day

15 Day

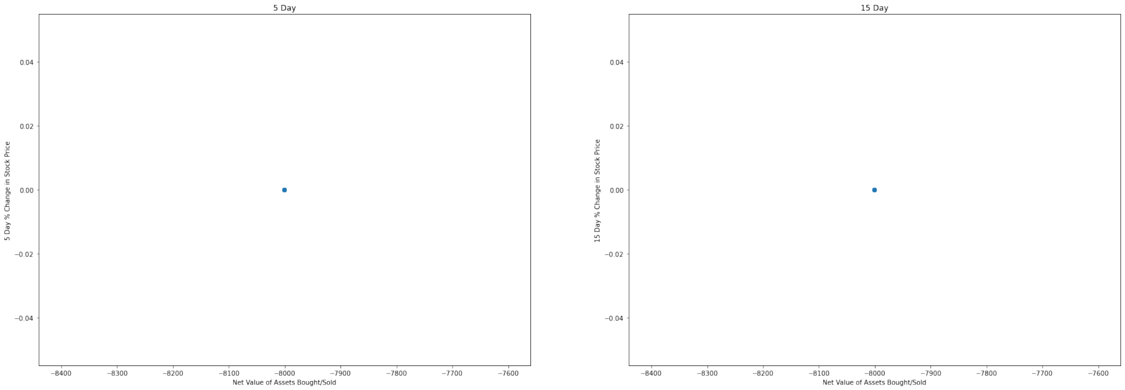


Trade Value vs. Future Stock Price % Change for Senator Michael B Enzi

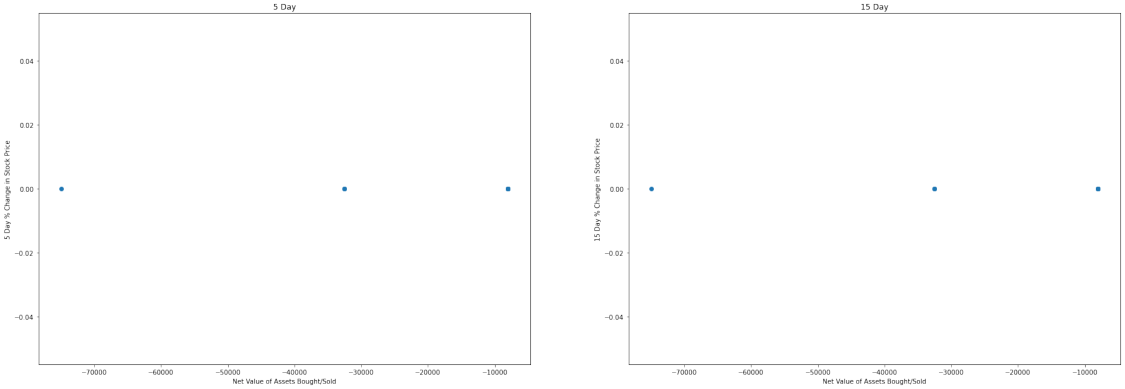
5 Day

15 Day

Trade Value vs. Future Stock Price % Change for Senator Benjamin L Cardin



Trade Value vs. Future Stock Price % Change for Senator Cory A Booker



```
[ ]:
```