Ex2

1.Write a Python function center_with_matrix(data) that subtracts the row-wise mean from the input array data by multiplying with the centering matrix

In [ ]:

```
import numpy as np
#create NumPy array
data = np.array([4, 6, 9, 13, 14, 17, 18, 19, 19, 21])
```

In [ ]:

```
def avg(data):
    return float(sum(data))/len(data)
```

In [ ]:

```
#create function to center data
center_with_matrix = lambda x: x - avg(data)
#apply function to original NumPy array
data_centered1 = center_with_matrix(data)
#view updated Array
print(data_centered1)
```

```
[-10.  -8.  -5.  -1.   0.   3.   4.   5.   5.   7.]
```

In [ ]:

```
#display mean of centered array
print(data_centered1.mean())
```

```
0.0
```

1. Write a Python function centering_with_numpy(data) that performs the same operation using basic NumPy-functions.

In [ ]:

```
centering_with_numpy = lambda x: x - np.mean(x, axis=0)

#apply function to original NumPy array
data_centered2 = centering_with_numpy(data)

print(data_centered2)
```

```
[-10.  -8.  -5.  -1.   0.   3.   4.   5.   5.   7.]
```

3.Sample random data matrices with uniformly distributed entries with 10 rows and a different number of columns. Plot the number of columns of the data matrix against the elapsed runtime for both functions. Also add a legend to your plot.

In [ ]:

```
import time
import copy
def columns(x):
   data1 =  np.random.randint(0,9999,(10,x))
   return data1
print(columns(9))
```

```
[[7020 6899 4774 6673 1752  209 8195 4236 5005]
 [2639 1222 9422   79 8540 9689 3065 5582 9892]
 [8740 6063 4118 2118 2788 7669 6968 2878 7429]
 [9351 9136 5132 2345 4072 8284 6267 1015 7462]
 [9767 9319 8017 9568 3412 2774 7200 6604 9312]
 [4805 4709 9396 7321 8991 5405 3594 8217 7785]
 [2590 5219 3510 7186 4235 8591 8528 7437 3558]
 [8912 9986  750 1949 2518 8254 1777 1196 5486]
 [5995  178 6563 2876 6243 6510 9838 3998 5671]
 [3205 9946 2906 9372  694 4970 3600 5004 9347]]
```

In [ ]:

```
def ex1():
 for x in range(2,1000):
   start = time.time()
   data_centered3 = center_with_matrix(columns(x))
   #print(data_centered3)
   end = time.time()
   run1 = end - start
 return run1
#print (ex1())
```

In [ ]:

```
def ex2():
 for x in range(2,1000):
   a2 = [1]
   start = time.time()
   data_centered3 = centering_with_numpy(columns(x))
   end = time.time()
   run2 = end - start
 return a2
```

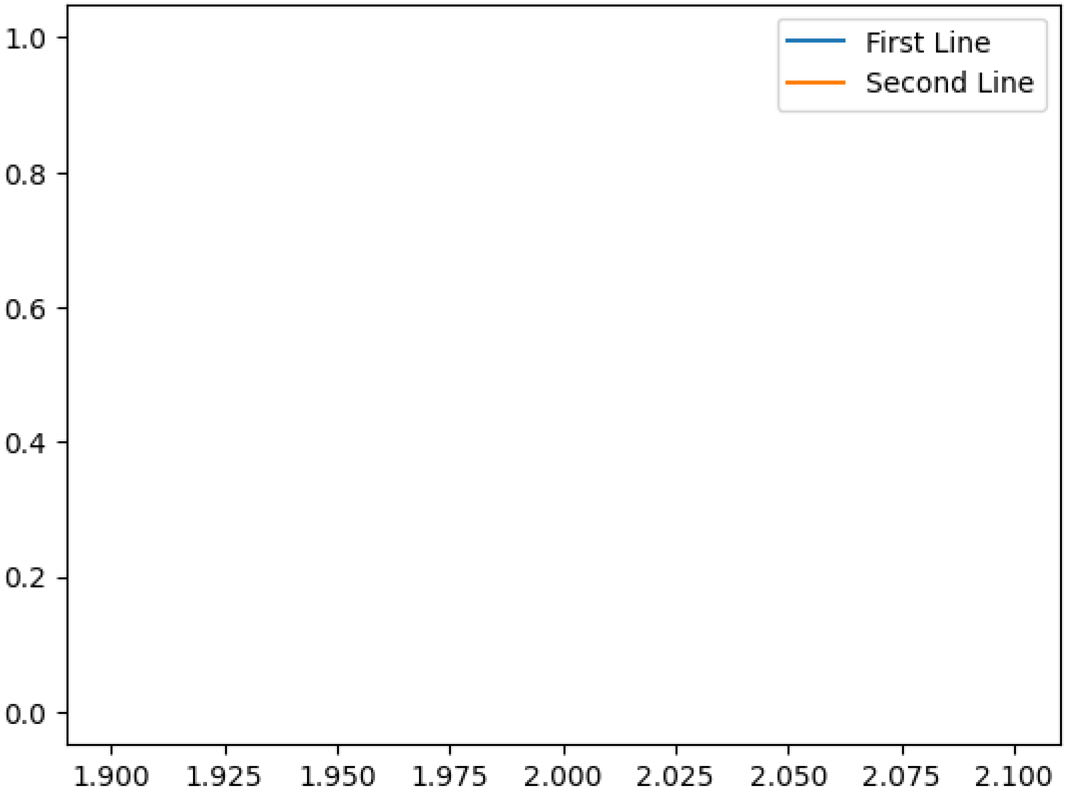
In [ ]:

```
import matplotlib.pyplot as plt
x1 = np.linspace(2, 1000, 1, endpoint=True)

y1 = ex1()
y2 = ex2()
plt.plot(x1,y1,label='First Line')
plt.plot(x1,y2,label='Second Line')
plt.legend()
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x205e44a20e0>
```


First Line
Second Line