collusion_of_varying_percentages

November 26, 2021

This code is designed to puggyback of of Mustafa's previously written code to go through various different collusion scanario's in an attempt to see which one is the 'optimal' amount of collusion to experiance within the Ethereum marketplace.

[1]:
```
import matplotlib.pyplot as plt
import random
from numpy import cumsum
```

[2]:
```
#define functions for future analysis

def collusion_scenario(p, epsilon, n=1000, gas_target=15000000, init_bf=100,␣
↪txn_fee=101, d=8):

    block_sizes = [gas_target]
    mempool = gas_target
    revenue = [0]
    mem_overfilled = mempool > 2 * gas_target
    base_fee = [init_bf]
    colluder_revenue = [0]
    myopic_revenue = [0]
    mempools = []


    for i in range(n - 1):
        colluding = random.random() < p
        if colluding:
            mined_amount = gas_target - epsilon
        else:
            mined_amount = (2 * gas_target) if mem_overfilled else mempool

        mempool -= mined_amount
        new_basefee = basefee(base_fee[i], mined_amount, gas_target, d)
        base_fee.append(new_basefee)
        block_sizes.append(mined_amount)
        block_revenue = (txn_fee - base_fee[i]) * mined_amount

        revenue.append(block_revenue)
```

```
        if colluding:
            colluder_revenue.append(block_revenue)
        else:
            myopic_revenue.append(block_revenue)

        mempool += gas_target
        mem_overfilled = mempool > 2 * gas_target
        mempools.append(mempool)

    data = {
        "Basefee" : base_fee,
        "Block Sizes" : block_sizes,
        "Overall Revenues" : revenue,
        "Myopic Revenues" : myopic_revenue,
        "Colluder Revenues" : colluder_revenue,
        "Avg Revenue per Colluder": sum(colluder_revenue) /␣
 ↪len(colluder_revenue) - 1,
        "Avg Revenue per Myopic": sum(myopic_revenue) / len(myopic_revenue) - 1,
        "Mempools": mempools
    }

    return data

def basefee(previous, mined_amount, gas_target, d):

    return previous*(1+(1/d)*((mined_amount-gas_target)/gas_target))
```

To get a good mix of possibilities for different collusion scenario's within our framework, we will use 10%, 20%, 40%, 60%, 80%

Now to start with 10%

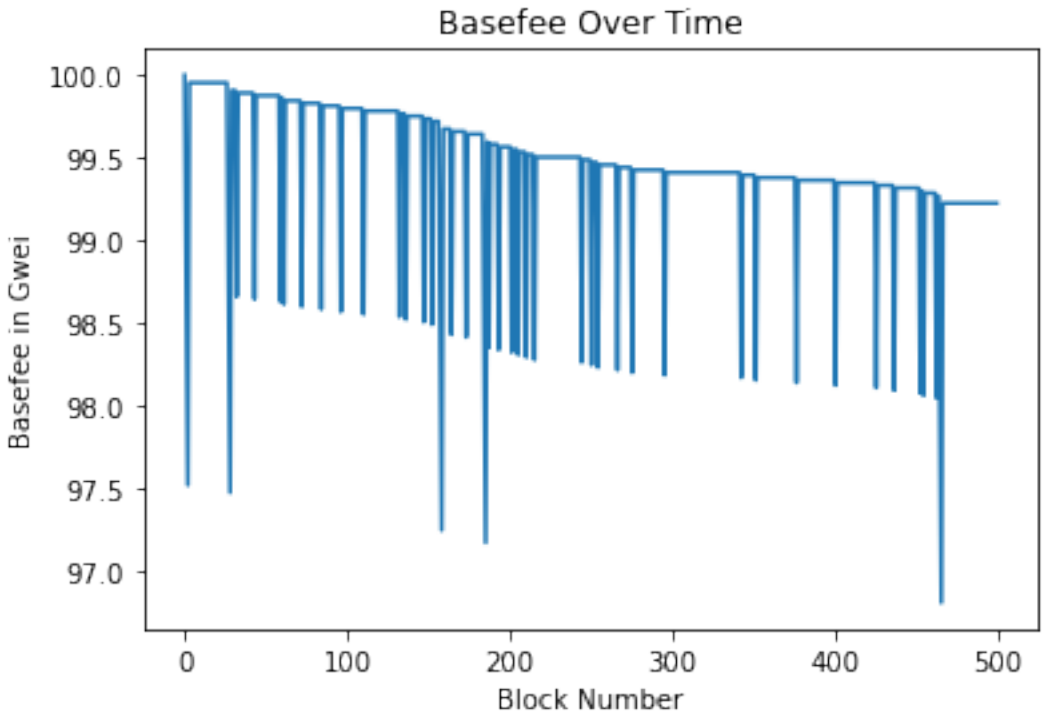
[3]:
```
sim = collusion_scenario(.1, 1500000, 500)
controll_sim = collusion_scenario(0, 1500000, 500)

print("Total Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]))
print("Total Myopic Revenue:", sum(sim["Myopic Revenues"]))
print("Total Colluder Revenues:", sum(sim["Colluder Revenues"]))
print("Average Colluder Revenue:", sim["Avg Revenue per Colluder"])
print("Average Myopic Revenue:", sim["Avg Revenue per Myopic"])
print("Average Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]) / \
      len(controll_sim["Overall Revenues"]))
```

```
Total Revenue Under No Collusion: 7485000000.0
Total Myopic Revenue: 10948375146.33928
```

```
Total Colluder Revenues: 926398241.3003154
Average Colluder Revenue: 20139091.20218077
Average Myopic Revenue: 24062361.958987426
Average Revenue Under No Collusion: 14970000.0
```

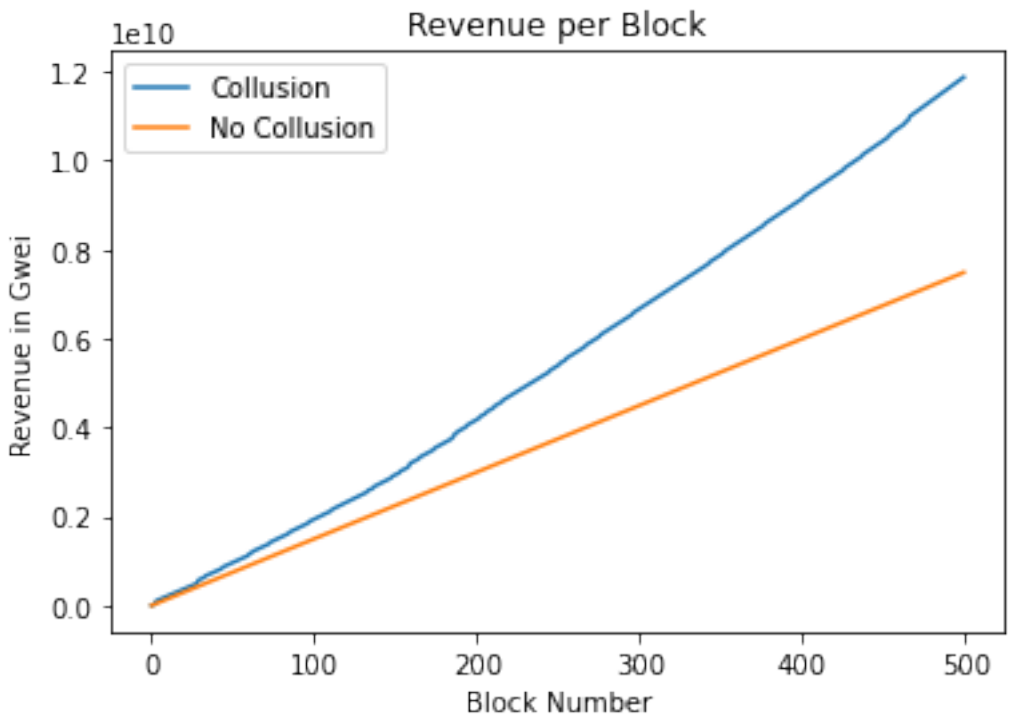
[4]:
```
##plot basefee
plt.title("Basefee Over Time")
plt.xlabel("Block Number")
plt.ylabel("Basefee in Gwei")
plt.plot(sim["Basefee"])
```

[4]: [<matplotlib.lines.Line2D at 0x7fa51686c970>]



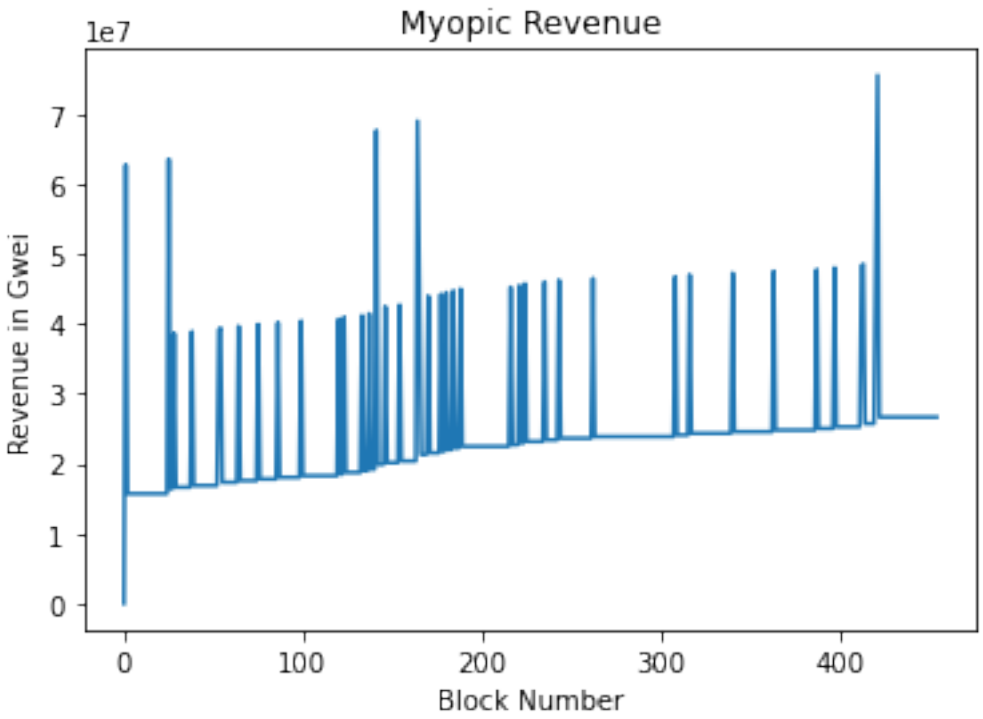
[5]:
```
##plot revenue per block
plt.title("Revenue per Block")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(cumsum(sim["Overall Revenues"]), label="Collusion")
plt.plot(cumsum(controll_sim["Overall Revenues"]), label="No Collusion")
plt.legend(loc="upper left")
```

[5]: <matplotlib.legend.Legend at 0x7fa5169b7c40>



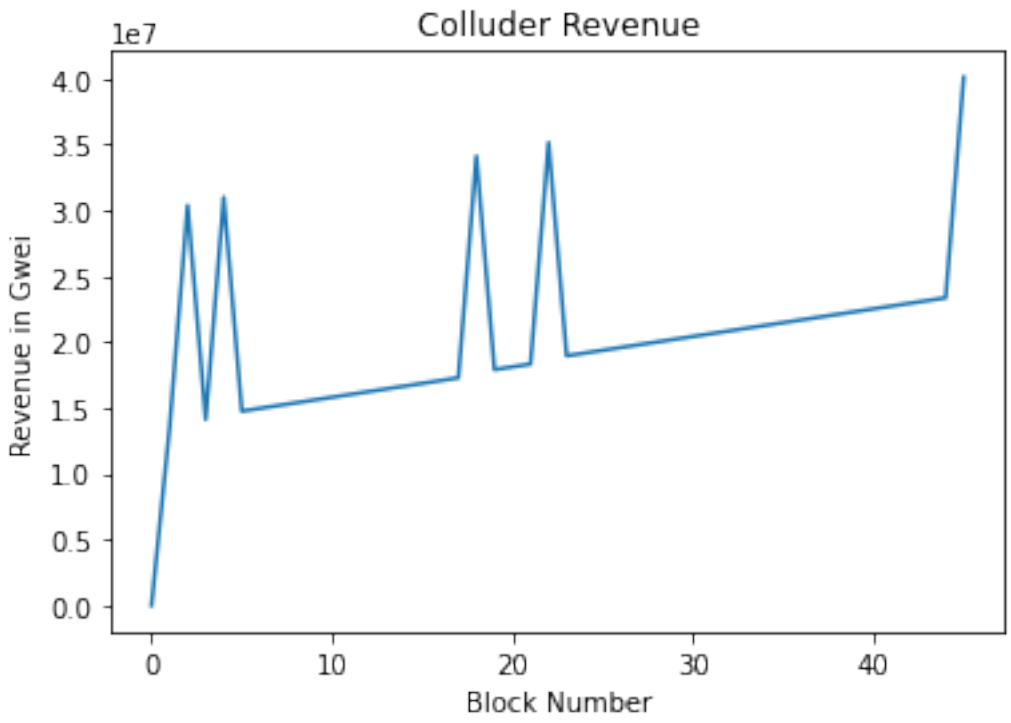
[6]:
```
##plot myopic revenue
plt.title("Myopic Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Myopic Revenues"])
```

[6]: [<matplotlib.lines.Line2D at 0x7fa516b53730>]



[7]:
```
##plot colluder revenue
plt.title("Colluder Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Colluder Revenues"])
```

[7]: [<matplotlib.lines.Line2D at 0x7fa516beefa0>]



Now to move on to 20%

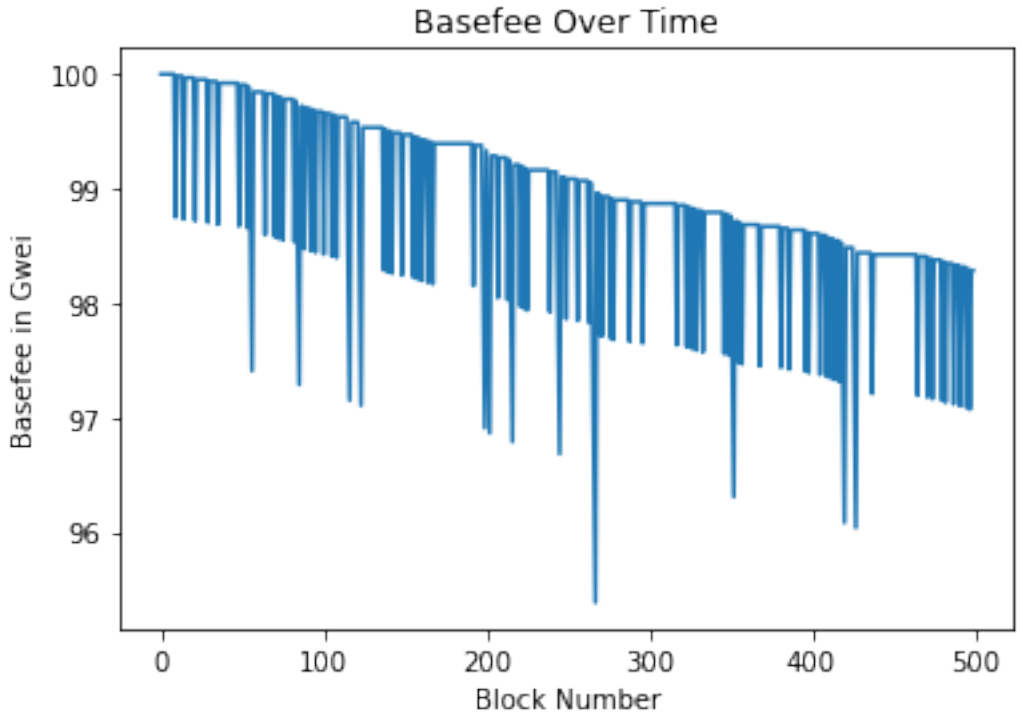
[8]:
```
sim = collusion_scenario(.2, 1500000, 500)
controll_sim = collusion_scenario(0, 1500000, 500)

print("Total Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]))
print("Total Myopic Revenue:", sum(sim["Myopic Revenues"]))
print("Total Colluder Revenues:", sum(sim["Colluder Revenues"]))
print("Average Colluder Revenue:", sim["Avg Revenue per Colluder"])
print("Average Myopic Revenue:", sim["Avg Revenue per Myopic"])
print("Average Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]) / \
      len(controll_sim["Overall Revenues"]))
```

```
Total Revenue Under No Collusion: 7485000000.0
Total Myopic Revenue: 13605869320.156475
Total Colluder Revenues: 2660272249.982294
Average Colluder Revenue: 27145634.20390096
Average Myopic Revenue: 33761461.33289448
Average Revenue Under No Collusion: 14970000.0
```

[9]:
```
##plot basefee
plt.title("Basefee Over Time")
```

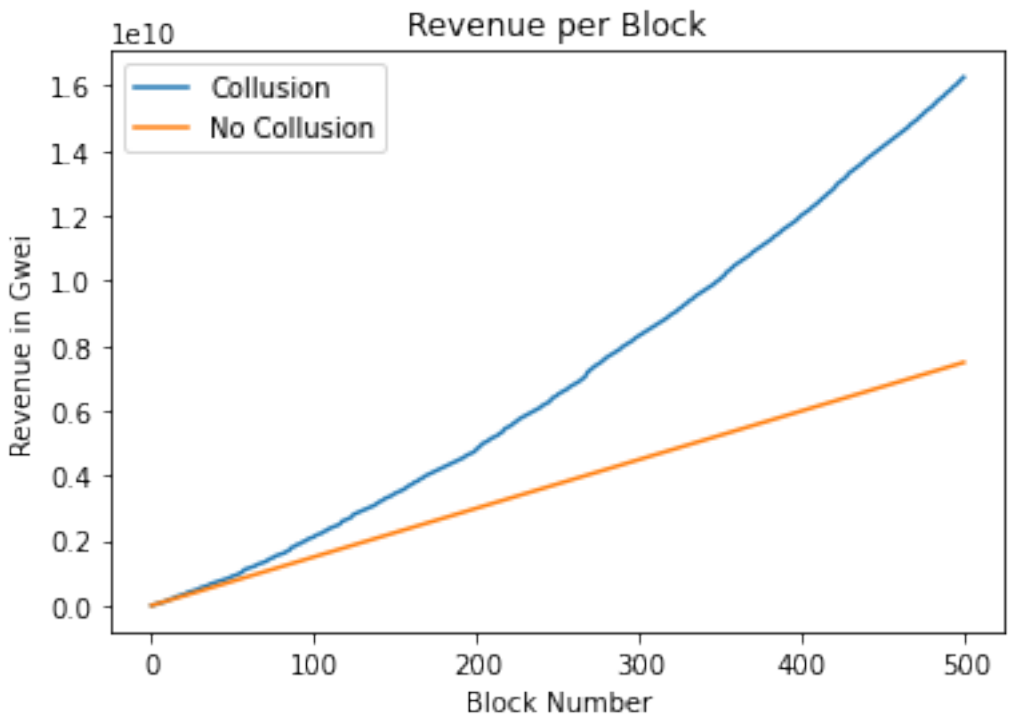
```
plt.xlabel("Block Number")
plt.ylabel("Basefee in Gwei")
plt.plot(sim["Basefee"])
```

[9]: [<matplotlib.lines.Line2D at 0x7fa516da0370>]



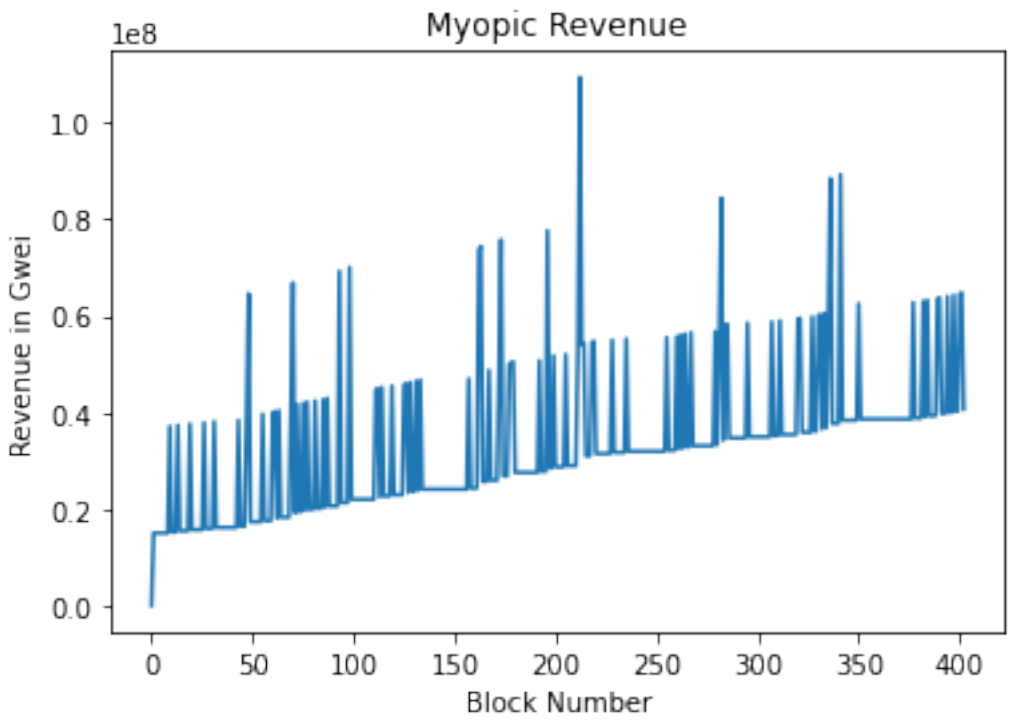
[10]:
```
##plot revenue per block
plt.title("Revenue per Block")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(cumsum(sim["Overall Revenues"]), label="Collusion")
plt.plot(cumsum(controll_sim["Overall Revenues"]), label="No Collusion")
plt.legend(loc="upper left")
```

[10]: <matplotlib.legend.Legend at 0x7fa516bc1100>



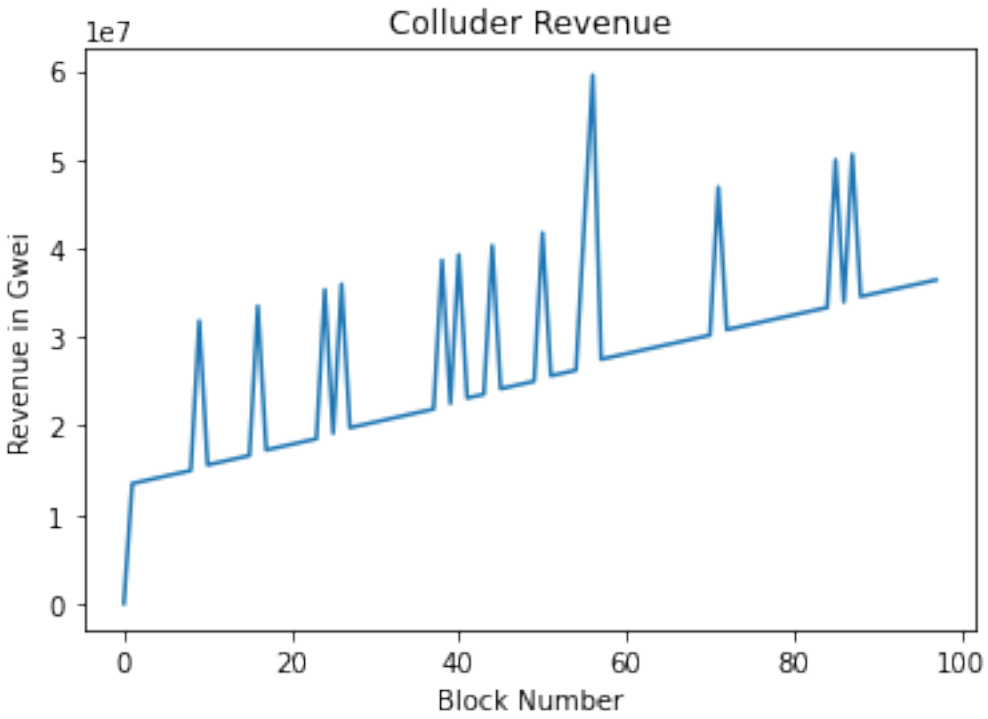
[11]:
```
##plot myopic revenue
plt.title("Myopic Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Myopic Revenues"])
```

[11]: [<matplotlib.lines.Line2D at 0x7fa517007610>]



[12]:
```
##plot colluder revenue
plt.title("Colluder Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Colluder Revenues"])
```

[12]: [<matplotlib.lines.Line2D at 0x7fa5170a2fa0>]



Now to move on to 40%

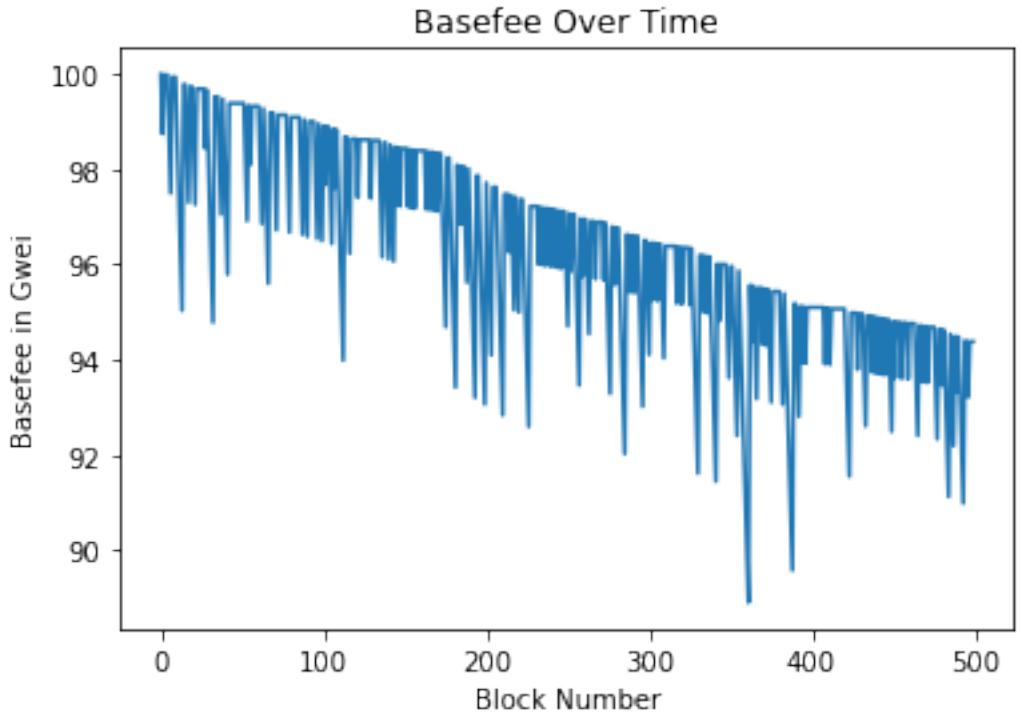
[13]:
```
sim = collusion_scenario(.4, 1500000, 500)
controll_sim = collusion_scenario(0, 1500000, 500)

print("Total Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]))
print("Total Myopic Revenue:", sum(sim["Myopic Revenues"]))
print("Total Colluder Revenues:", sum(sim["Colluder Revenues"]))
print("Average Colluder Revenue:", sim["Avg Revenue per Colluder"])
print("Average Myopic Revenue:", sim["Avg Revenue per Myopic"])
print("Average Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]) / \
      len(controll_sim["Overall Revenues"]))
```

```
Total Revenue Under No Collusion: 7485000000.0
Total Myopic Revenue: 22539616032.004265
Total Colluder Revenues: 13816864019.384932
Average Colluder Revenue: 63090702.284862705
Average Myopic Revenue: 79927715.42554703
Average Revenue Under No Collusion: 14970000.0
```

[14]:
```
##plot basefee
plt.title("Basefee Over Time")
```

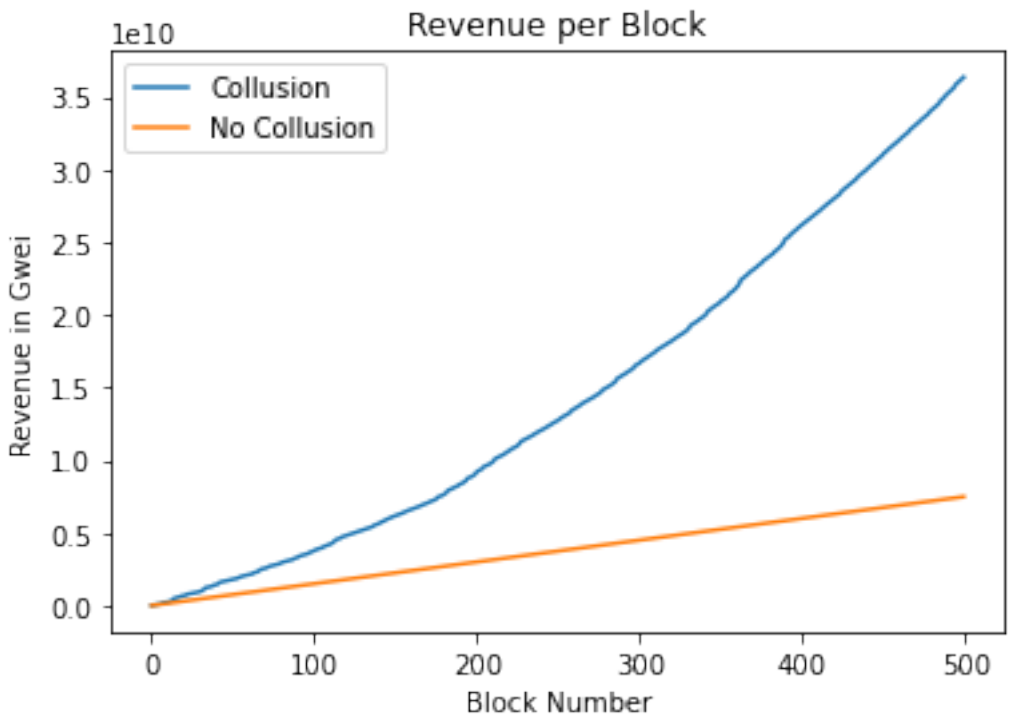
```
plt.xlabel("Block Number")
plt.ylabel("Basefee in Gwei")
plt.plot(sim["Basefee"])
```

[14]: [<matplotlib.lines.Line2D at 0x7fa51724c700>]



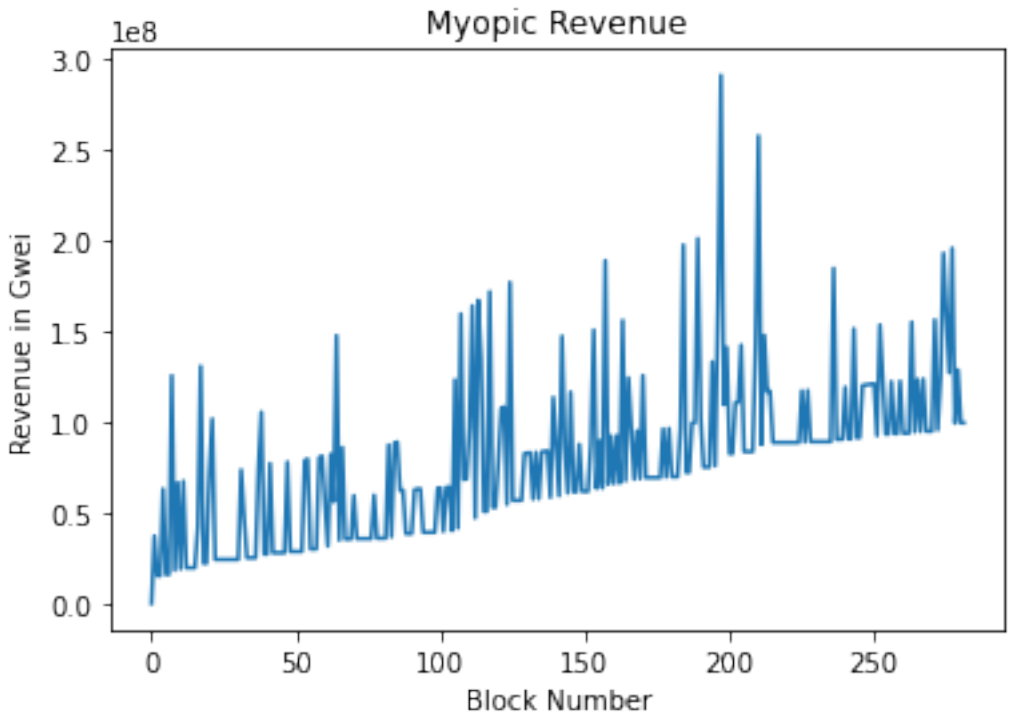
[15]:
```
##plot revenue per block
plt.title("Revenue per Block")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(cumsum(sim["Overall Revenues"]), label="Collusion")
plt.plot(cumsum(controll_sim["Overall Revenues"]), label="No Collusion")
plt.legend(loc="upper left")
```

[15]: <matplotlib.legend.Legend at 0x7fa5172adc40>



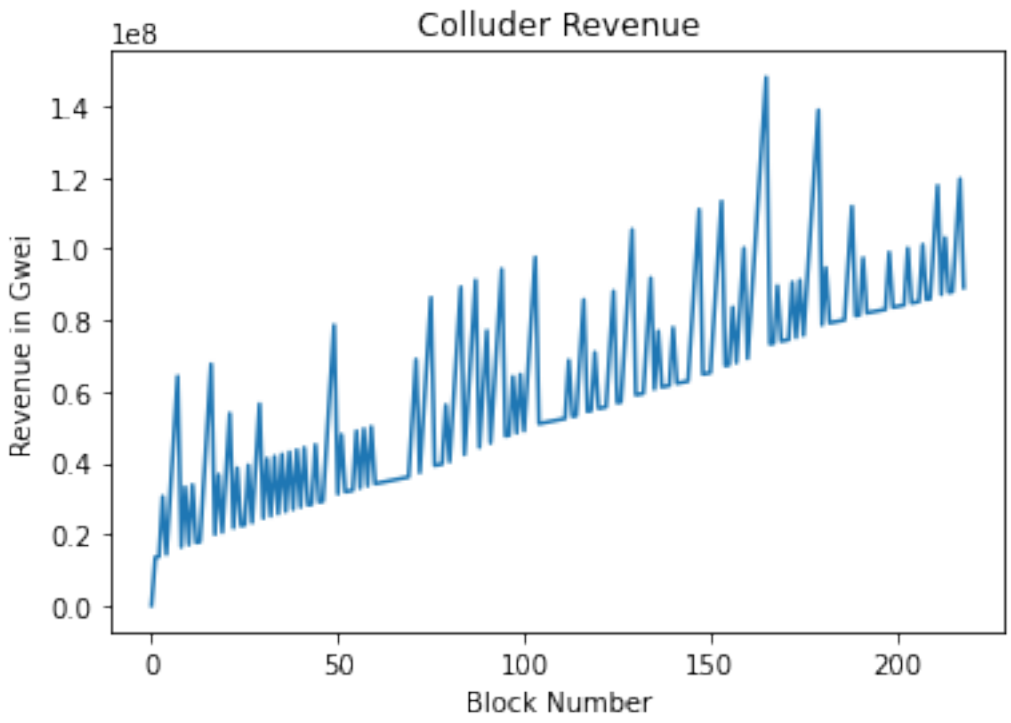
[16]:
```
##plot myopic revenue
plt.title("Myopic Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Myopic Revenues"])
```

[16]: [<matplotlib.lines.Line2D at 0x7fa5174538b0>]



[17]:
```
##plot colluder revenue
plt.title("Colluder Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Colluder Revenues"])
```

[17]: [<matplotlib.lines.Line2D at 0x7fa5175dc670>]



Now for 60%

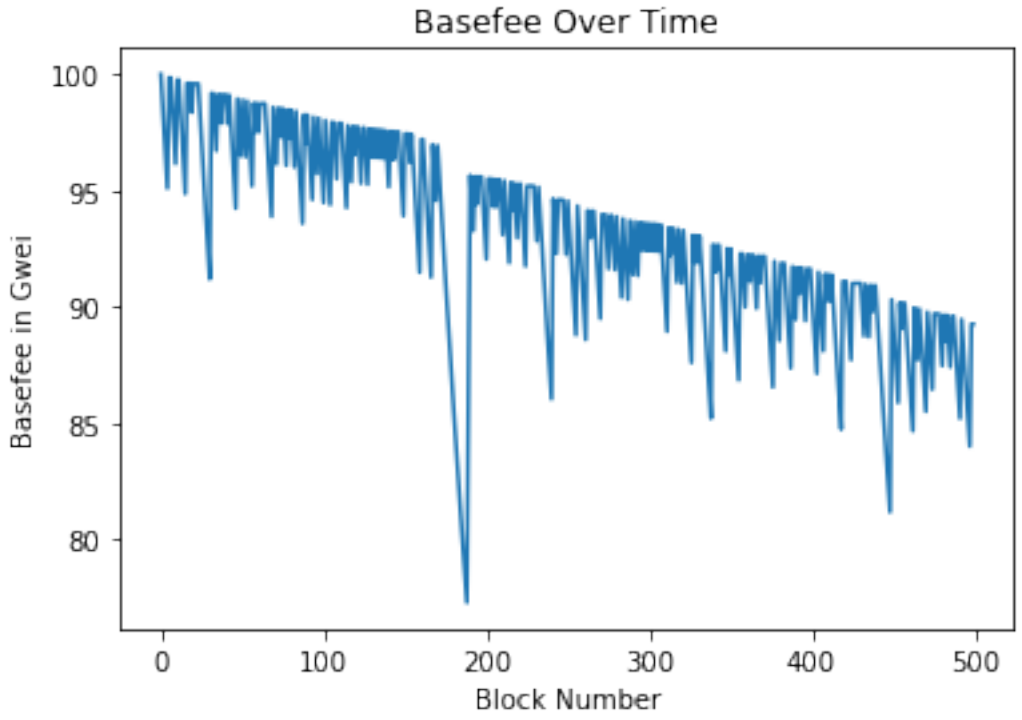
[18]:
```
sim = collusion_scenario(.6, 1500000, 500)
controll_sim = collusion_scenario(0, 1500000, 500)

print("Total Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]))
print("Total Myopic Revenue:", sum(sim["Myopic Revenues"]))
print("Total Colluder Revenues:", sum(sim["Colluder Revenues"]))
print("Average Colluder Revenue:", sim["Avg Revenue per Colluder"])
print("Average Myopic Revenue:", sim["Avg Revenue per Myopic"])
print("Average Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]) / \
      len(controll_sim["Overall Revenues"]))
```

```
Total Revenue Under No Collusion: 7485000000.0
Total Myopic Revenue: 28157615736.842392
Total Colluder Revenues: 34077454368.83009
Average Colluder Revenue: 110641084.6130847
Average Myopic Revenue: 145894381.05617818
Average Revenue Under No Collusion: 14970000.0
```

[19]:
```
##plot basefee
plt.title("Basefee Over Time")
```

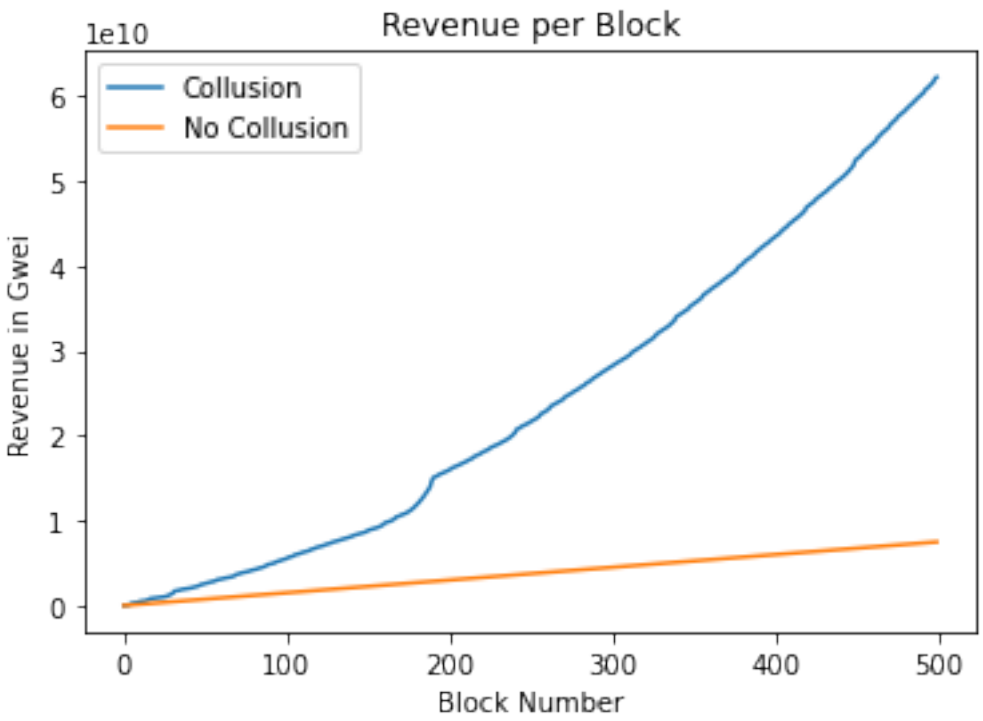
```
plt.xlabel("Block Number")
plt.ylabel("Basefee in Gwei")
plt.plot(sim["Basefee"])
```

[19]: [<matplotlib.lines.Line2D at 0x7fa517712130>]



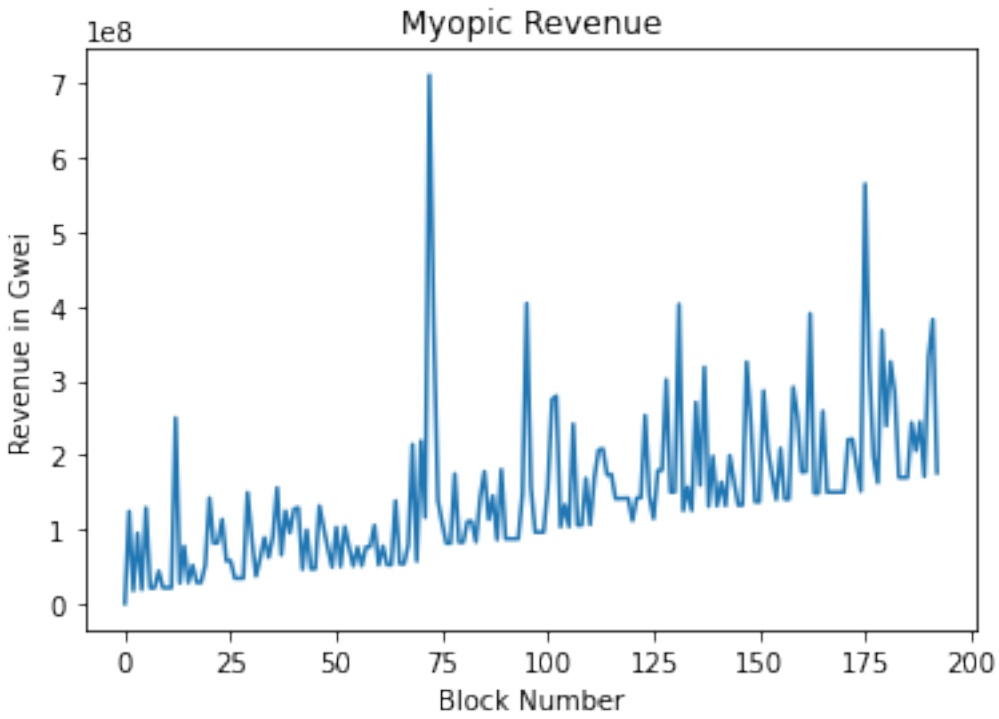
[20]:
```
##plot revenue per block
plt.title("Revenue per Block")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(cumsum(sim["Overall Revenues"]), label="Collusion")
plt.plot(cumsum(controll_sim["Overall Revenues"]), label="No Collusion")
plt.legend(loc="upper left")
```

[20]: <matplotlib.legend.Legend at 0x7fa517799250>



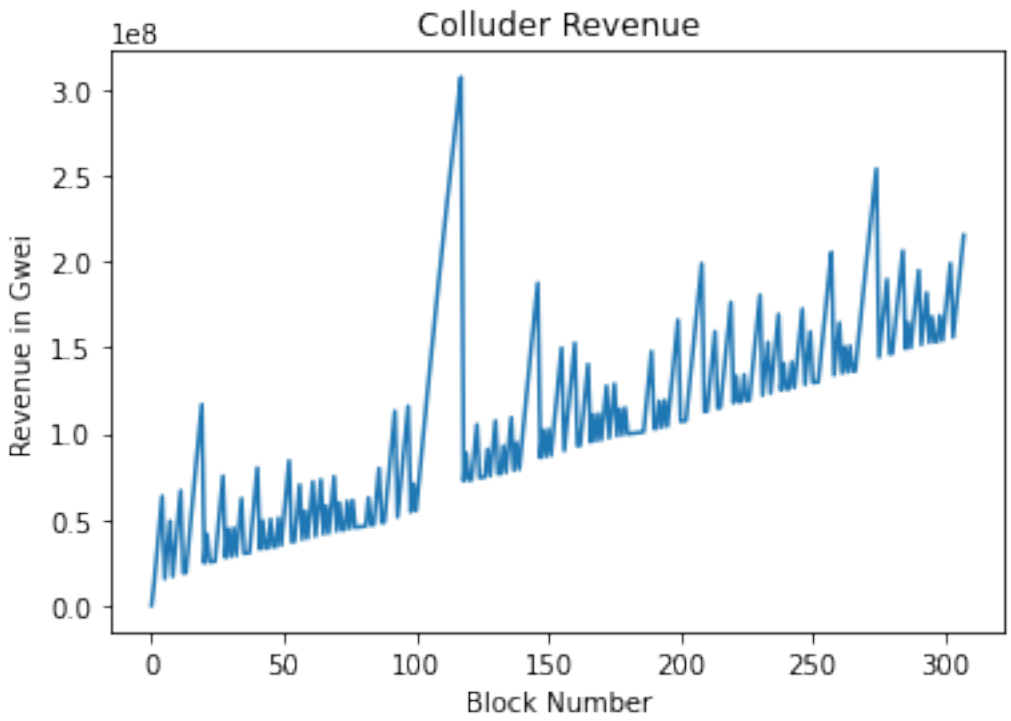
[21]:
```
##plot myopic revenue
plt.title("Myopic Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Myopic Revenues"])
```

[21]: [<matplotlib.lines.Line2D at 0x7fa51797a7c0>]



[22]:
```
##plot colluder revenue
plt.title("Colluder Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Colluder Revenues"])
```

[22]: [<matplotlib.lines.Line2D at 0x7fa517a8abe0>]



Now for 80%

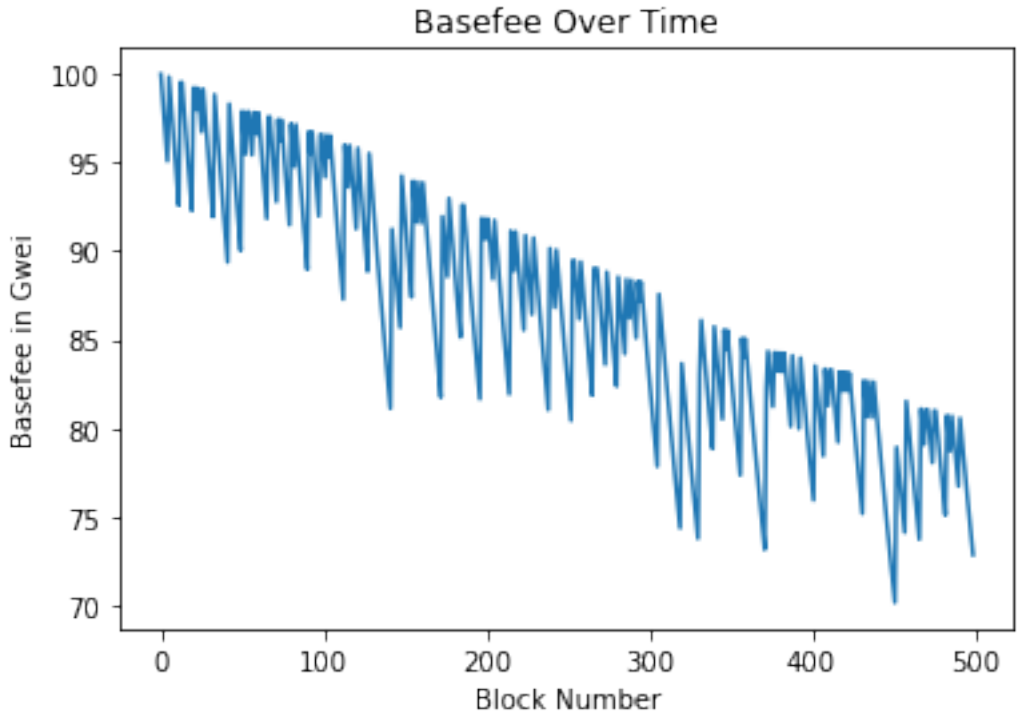
[23]:
```
sim = collusion_scenario(.8, 1500000, 500)
controll_sim = collusion_scenario(0, 1500000, 500)

print("Total Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]))
print("Total Myopic Revenue:", sum(sim["Myopic Revenues"]))
print("Total Colluder Revenues:", sum(sim["Colluder Revenues"]))
print("Average Colluder Revenue:", sim["Avg Revenue per Colluder"])
print("Average Myopic Revenue:", sim["Avg Revenue per Myopic"])
print("Average Revenue Under No Collusion:", sum(controll_sim["Overall␣
 ↪Revenues"]) / \
      len(controll_sim["Overall Revenues"]))
```

```
Total Revenue Under No Collusion: 7485000000.0
Total Myopic Revenue: 33136393158.728683
Total Colluder Revenues: 77030799648.96292
Average Colluder Revenue: 192576998.12240732
Average Myopic Revenue: 328083099.5814721
Average Revenue Under No Collusion: 14970000.0
```

[24]:
```
##plot basefee
plt.title("Basefee Over Time")
```

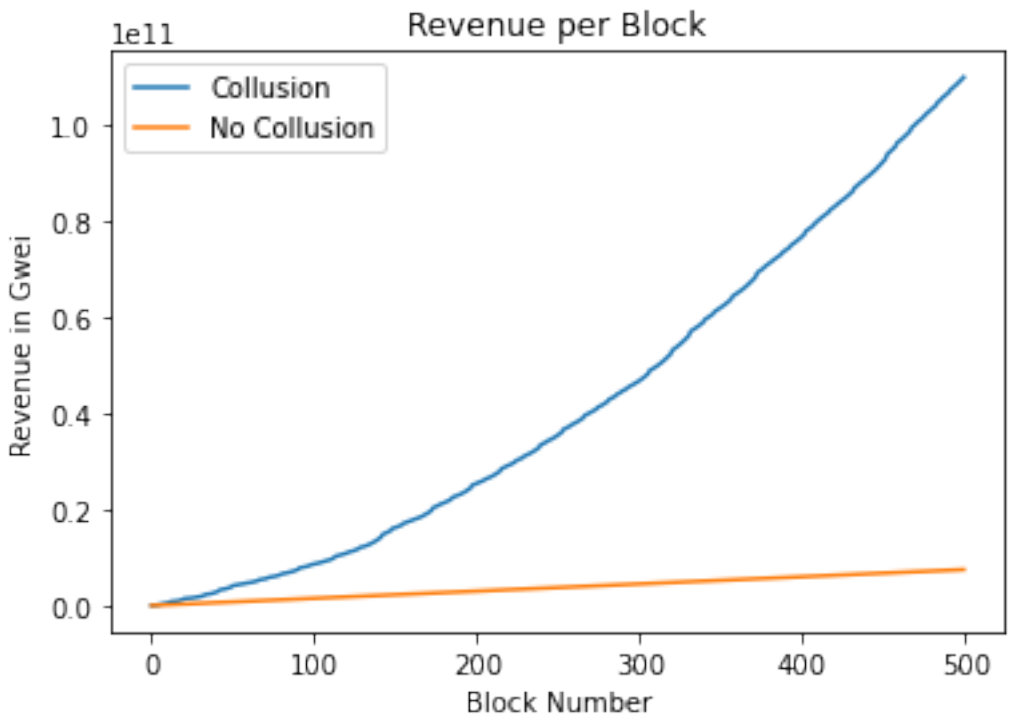
```
plt.xlabel("Block Number")
plt.ylabel("Basefee in Gwei")
plt.plot(sim["Basefee"])
```

[24]: [<matplotlib.lines.Line2D at 0x7fa517baceb0>]



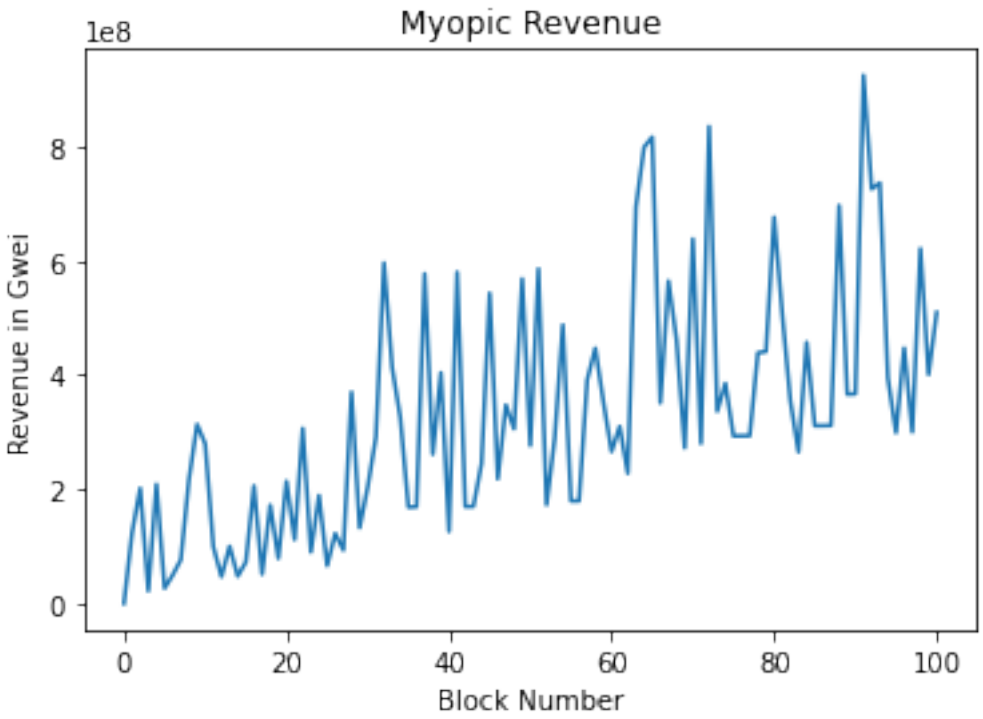
[25]:
```
##plot revenue per block
plt.title("Revenue per Block")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(cumsum(sim["Overall Revenues"]), label="Collusion")
plt.plot(cumsum(controll_sim["Overall Revenues"]), label="No Collusion")
plt.legend(loc="upper left")
```

[25]: <matplotlib.legend.Legend at 0x7fa517c6cfd0>



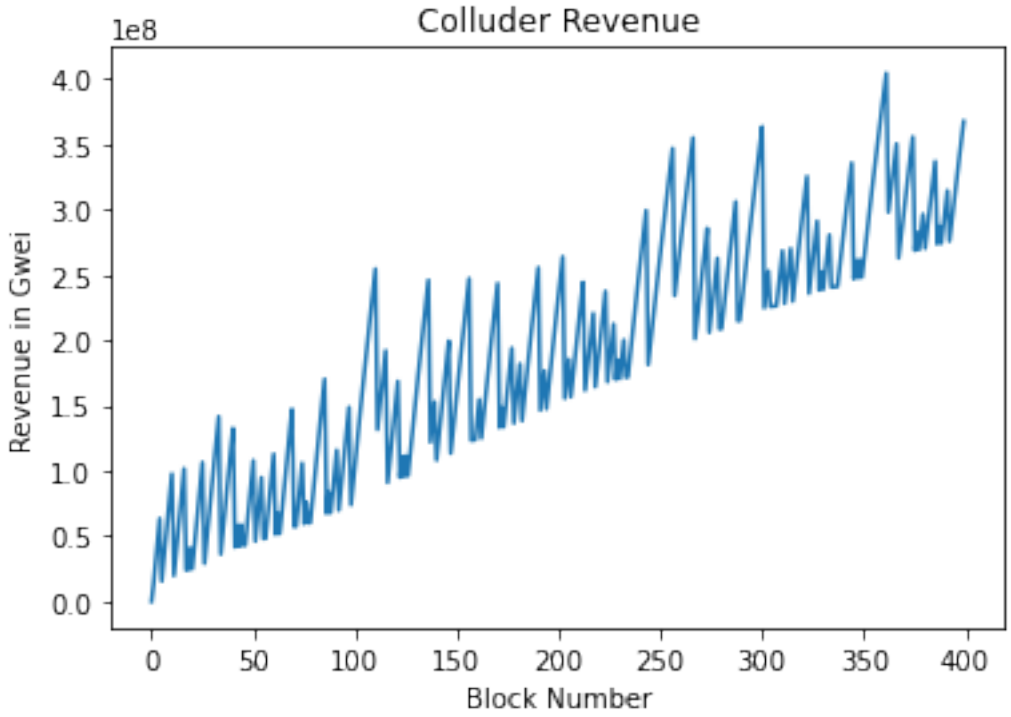
[26]:
```
##plot myopic revenue
plt.title("Myopic Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Myopic Revenues"])
```

[26]: [<matplotlib.lines.Line2D at 0x7fa517e1e730>]



[27]:
```
##plot colluder revenue
plt.title("Colluder Revenue")
plt.xlabel("Block Number")
plt.ylabel("Revenue in Gwei")
plt.plot(sim["Colluder Revenues"])
```

[27]: [<matplotlib.lines.Line2D at 0x7fa517ead820>]


Colluder Revenue
1e8
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Revenue in Gwei
0
50
100
150
200
250
300
350
400
Block Number