Out[2]:

Click here to toggle on/off the raw code.

# Gas Fees Data Exploration

## TOC:



### Quick Data exploration

### Load .csv data, add new columns, clean up

Quick recap of some important metrics to keep in mind:

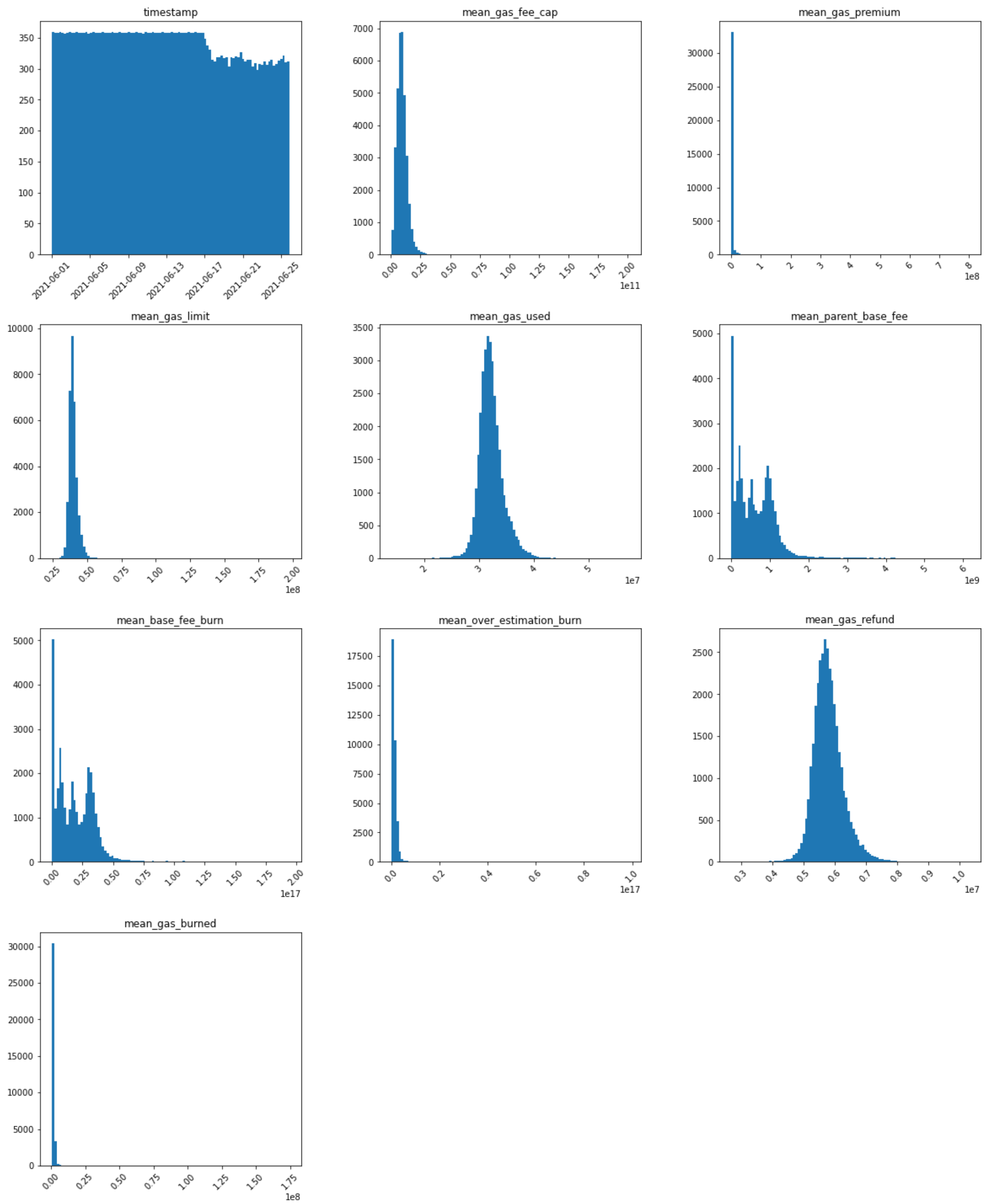
```
- TotalTxFees = (GasPremium + BaseFee) * GasLimit
- GasUsed * BaseFee = BaseFeeBurned
- GasFeeCap - BaseFee = GasPremium
- GasPremium * GasLimit = PriorityFee
- GasLimit - GasUsed = OverEstimationBurn + Refund
```

### Count, mean, std, min/max, percentile description

Out[30]:

| | mean_gas_fee_cap | mean_gas_premium | mean_gas_limit | mean_gas_used | mean_parent_ba |
|---|---|---|---|---|---|
| count | 3.427300e+04 | 3.427300e+04 | 3.427300e+04 | 3.427300e+04 | 3.4273( |
| mean | 9.332010e+09 | 1.497482e+06 | 3.989642e+07 | 3.226650e+07 | 6.1066! |
| std | 4.525614e+09 | 8.109318e+06 | 4.347213e+06 | 2.210611e+06 | 5.2089: |
| min | 5.746347e+08 | 8.200945e+04 | 2.428070e+07 | 1.408784e+07 | 1.0000( |
| 25% | 6.363798e+09 | 1.114223e+05 | 3.779839e+07 | 3.087624e+07 | 1.9999( |
| 50% | 8.880672e+09 | 3.657263e+05 | 3.932166e+07 | 3.202065e+07 | 5.4041' |
| 75% | 1.162421e+10 | 8.884868e+05 | 4.124140e+07 | 3.338107e+07 | 9.4846' |
| max | 2.011381e+11 | 7.990103e+08 | 1.974523e+08 | 5.744429e+07 | 6.1472: |

## Gas metrics distributions



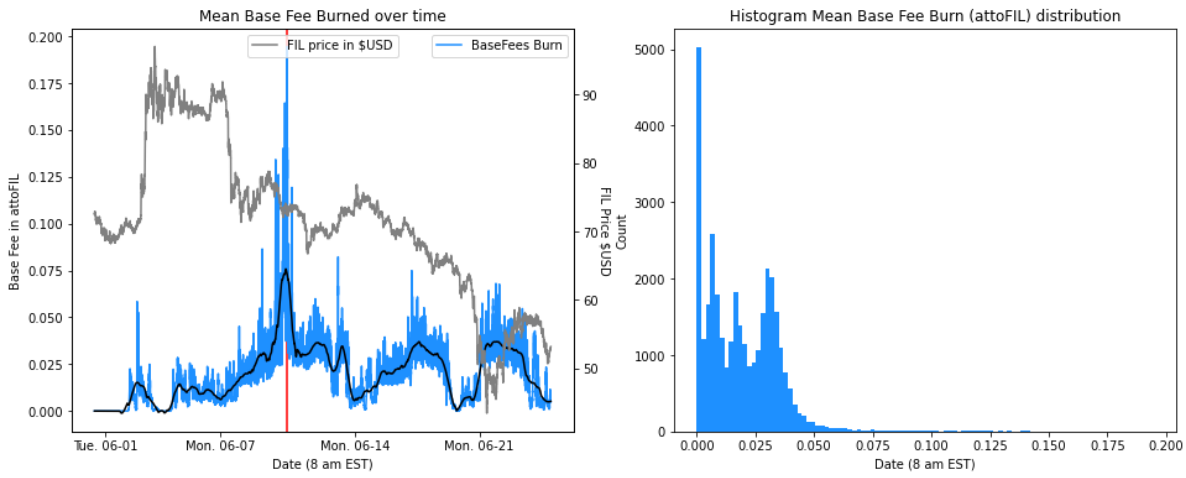

## I. Base Fees

***"Gas consumption is a cost that should be shouldered by the whole network, since every node on the network has to spend storage and computational resources to validate each message and maintain a consistent state of the network. As such, some amount of gas is burned to compensate for the network based on the gas usage of a particular message."***

**1) GasUsage multiplied by BaseFee is burned as on-chain resources are consumed, and reflect how much demand there is for blockspace. On the given time period, we show that it follows a multimodal distribution, with long tail and a peak towards values close to zero, which intuitively makes sense because gas fees will converge/go back to zero when demand is low.**

Out[42]:

Click here to toggle on/off the raw code.




**2) We then show that the amount of base fees burned per minute fluctuates according to the base fees value, while the gas used remains relatively constant and follows a normal distribution.**

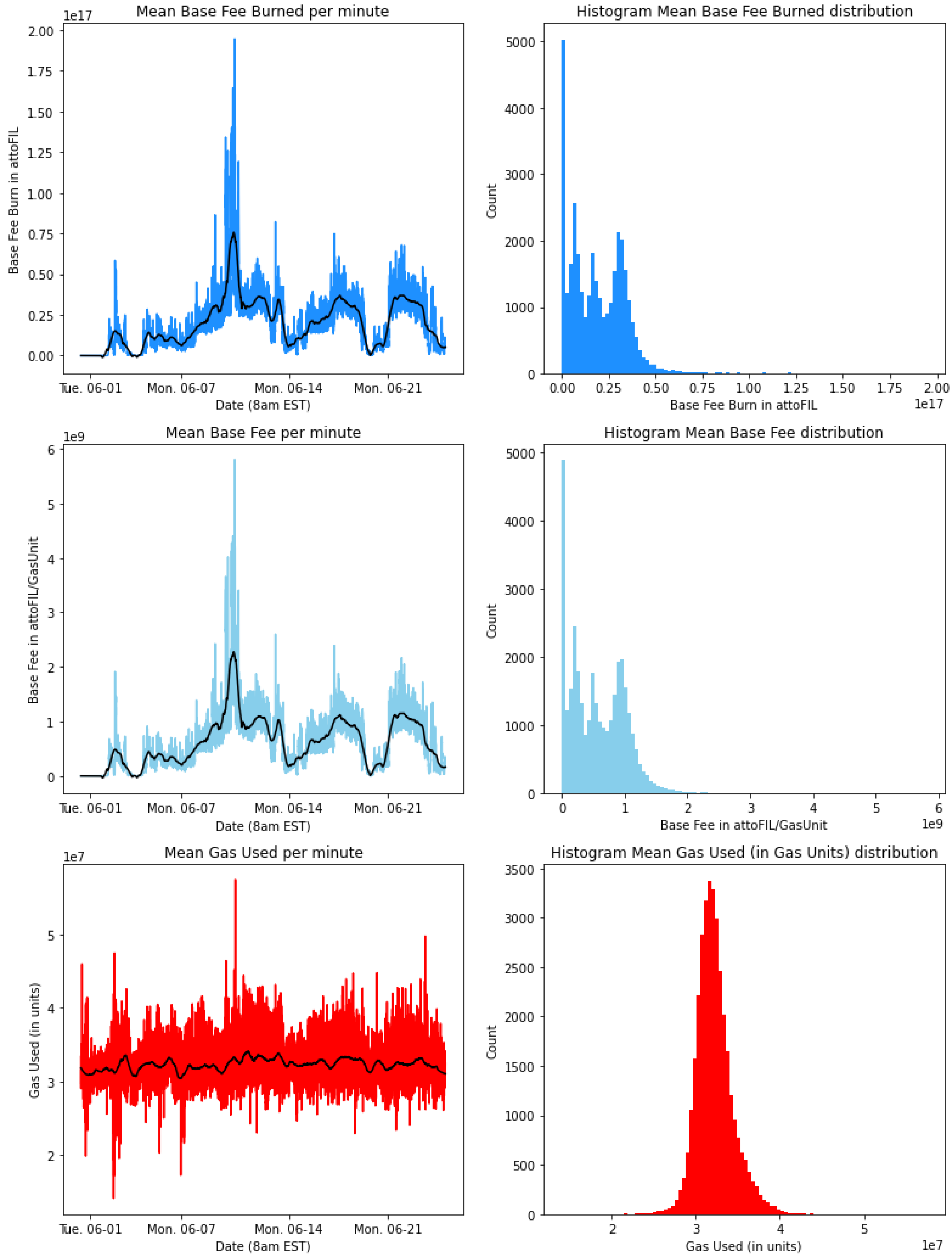
To do so, we first compute mean base fees and add the corresponding column in our dataframe

```
Mean Base Fee = Mean Base Fee Burned / Mean Gas Used
```

Then, we plot mean base fees, the mean amount of base fees burned, and the mean gas used per minute, as well as their distributions

Out[44]:

Click here to toggle on/off the raw code.



**3) We also noticed an oscillation pattern in base fees over time, so we asked whether base fees followed a consistent pattern with higher fees during week days than on the weekend.**

First, we create a new "weekend" boolean column (**True** if Saturday or Sunday, **False** if Monday, Tuesday, Wednesday, Thursday and Friday).

Out[35]:

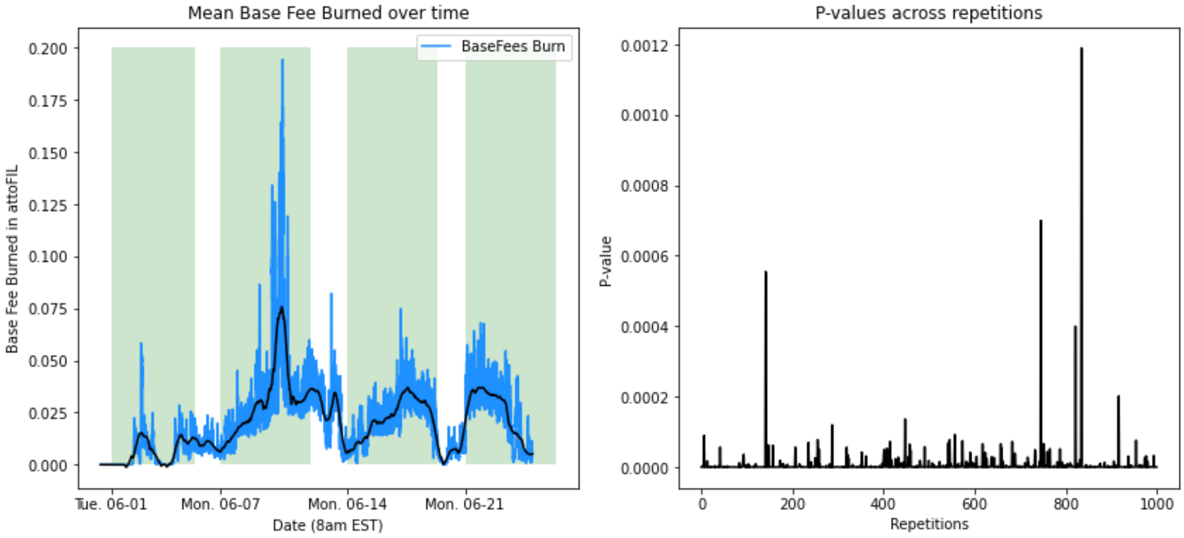
| weekend | mean_base_fee_burn |
|---|---|
| **False** | 1.975843e+16 |
| **True** | 1.848275e+16 |

Since "weekday" and "weekend" conditions were significantly unbalanced (i.e., larger number of "weekday" samples), we randomly selected conscious samples in order to match the number of "weekend" samples before performing the statistical analysis and repeated this procedure 100 times, with different segments randomly chosen every time, and observed pvalues consistently being inferior to 0.05 across repetitions.

**We found that the amount of base fees burned is significantly higher on week days, indicating an non negligible participation from professional actors in the Filecoin ecosystem.**

Out[45]:

Click here to toggle on/off the raw code.




## II. Gas Limit and Overestimating Gas Usage

Next, we focus on a particularly tricky property of the Filecoin protocol: it does not know how much gas a message will exactly consume ahead of execution. This means users have to estimate how much gas will be used for their transaction, by setting the GasLimit parameter.

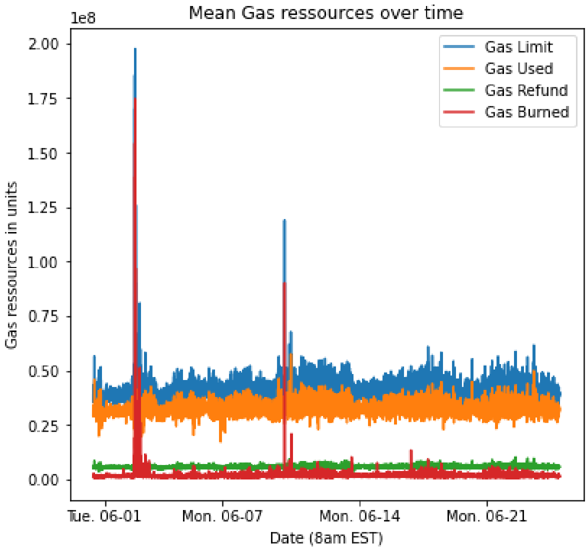
By how gas ressources are distributed over time, we show that difference between the gas limit set by users and the gas actually used to consume blockchain ressources is split between a part that is burned. The two peaks in gas limits that probably originate from errors on the user end capture the importance not to overestimate the gas limit too much since part of the overestimated fees are burned. We also see that the remaining fees are refunded to the user.

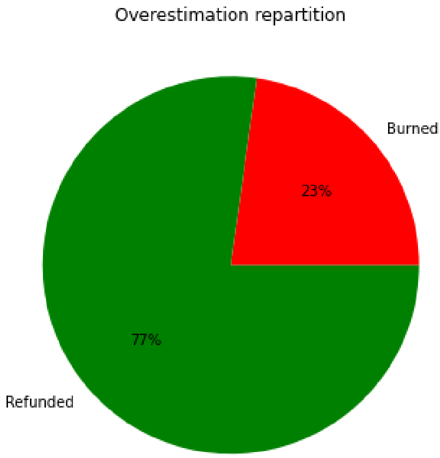
In practice, we can see that users usually set a gas limit around 20% higher that the gas that's actually used to consume blockchain ressources. Out of the 18.8% difference called the overestimation, 77% % is actually refunded to the user, and only 4.3% is lost by the user and burned (indirectly redistributing value for all network participants by reducing the circulating supply).

```
Average Diff. Between Gas Used and Gas Limit: 18.886238593257744 %
```

Out[38]:

Click here to toggle on/off the raw code.




## III. Miners' strategies

Currently, the default implementation is that miners select messages based on GasFeeCap/GasLimit to maximize their expected return given the GasLimit of a block. When the network is congested and BaseFee is high, miners can choose to underpack blocks, hence reducing total GasLimit, to bring down the BaseFee, but potentially at the expense of some GasPremium.

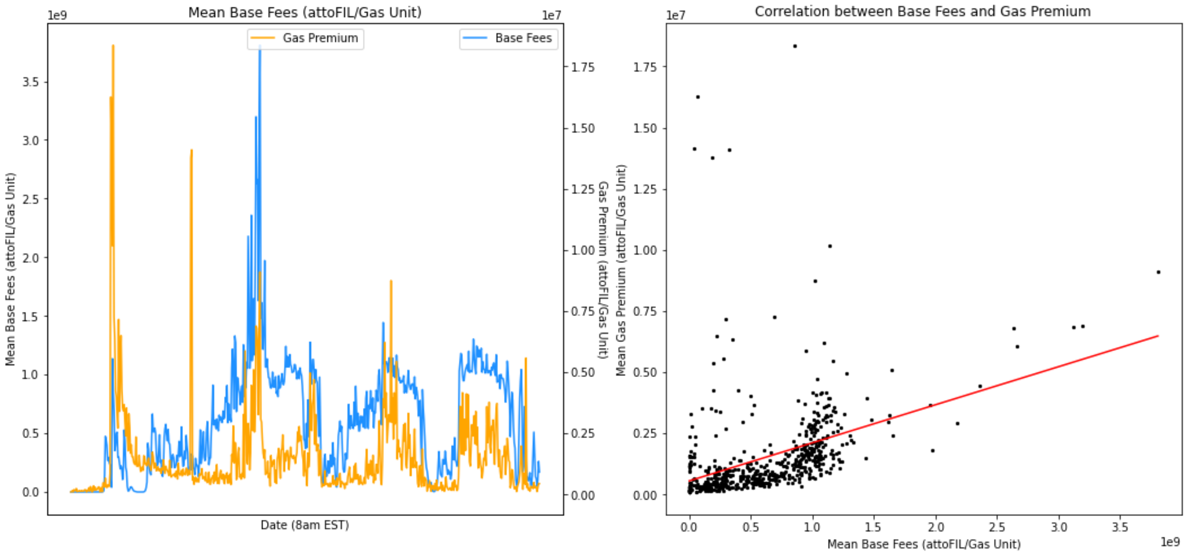
So what do miners actually do? (or at least, how did they behave during the month of June 2021)

To find out, we checked if BaseFee is correlated with GasPremium. We grouped and averaged BaseFee and GasPremium per hour, and used Spearman's rank correlation to show that BaseFee showed a significant positive correlation (Rcoef = 0.67, $p < 0.01$) with GasPremium. The results are displayed in the figure below:

```
SpearmanrResult(correlation=0.6700213424956395, pvalue=3.6906699554625
36e-79)
```

Out[46]:

Click here to toggle on/off the raw code.




## IV. Thoughts and Directions

1. More proper analyses can be done outlier to explore correlations between base fees and other gas fees components (e.g., outlier detection).
2. I always find working with averaged data a bit frustrating, I feel like getting access to transaction level data (including addresses, message types, etc.) allows taking analyses up a notch
3. Network analysis (graph theory) around events causing network congestion would be an avenue I would love to explore
4. ML (relatively straight forward) to determine features or actors that contributes to increases in tx fees
5. Computational Modeling/Predicting the impact that FVM-enabled projects will have on gas fees (and the more global supply)

## V. Bonus!

Don't forget the bonus! I built a 3D graph of FIL transactions on the BNB chain during the same time period as the assignement (June 2021). Edges represent txs, and nodes represent addresses and their associated labels queried on Dune.

[Click here (https://soispoke.github.io/CryptoEconAssignment/example/large-graph/)](https://soispoke.github.io/CryptoEconAssignment/example/large-graph/) to visualize the graph