Calculating the Impact of EIP-1559 on Transaction Fee Variance

Author: Jack Sanderson
Supervisor: Dr Yuhua Li
Moderator: Dr Natasha Edwards

BSc Computer Science
School of Computer Science and Informatics
Cardiff University

May 2023

I. Abstract

As the Ethereum blockchain grew in popularity at the end of 2017, the demand to use the blockchain grew alongside. This demand showed a number of problems with the fundamentals of how the Ethereum blockchain was designed. Arguably, the most impactful to the user experience was the large and frequent fluctuations in the transaction fees paid by users over short time frames. This transaction fee variance drove many users away, with many more critiquing the technologies prospects. In an attempt to reduce the transaction fee variance, along with other issues, the Ethereum community produced the update EIP-1559, also referred to as the London Hard Fork. Despite anecdotal evidence that EIP-1559 has had a positive impact on reducing transaction fee variance for the average user, there is limited research on the subject. This study aims to find and then quantify the difference in transaction fee variance before and after EIP-1559 on the Ethereum blockchain. To do so, several statistical measures and graphical visualizations are performed on data acquired from a locally running Ethereum node. The results from the research show that transaction fee variance from non-outlier transactions has been reduced across a large range of transaction gas prices. Furthermore, the results also show that the same experiment can be performed on another similar blockchain and achieve remarkably similar results.

II. Acknowledgements

I would like to thank my parents and my gran for financially supporting me throughout my bachelor's degree. The unrelenting support I received from my family to pursue my passions is cherished deeply.

III. Table of Contents

IV. List of Figures

V. List of Tables

1. Introduction

The year is 2017, and the first major bubble in the crypto-currency ecosystem is in full swing. Millionaires are being made out of nowhere and everyone wants in. Unfortunately, there isn't enough room for everyone with the two largest blockchains being unable to keep up. Despite the very clear throughput issue, the fever is overwhelming and people are willing to pay to get their slice of the pie. The fees paid to use these blockchains skyrocket to never before seen highs. As the dust settles and as fortunes are made and lost a worrying crack in the underlying technology starts to show. With wildly inconsistent transaction fees people start to question the viability of using blockchains for day-to-day financial activity. Many blockchains ignored the issues, pretending it didn't exist or that it didn't matter, but the Ethereum blockchain started building. With the lofty goals of attempting to replace large parts of modern financial infrastructure those that worked on the project felt it was uniquely important to solve. On August 5th, 2021, EIP (Ethereum Improvement Proposal) 1559 or the London hard fork, goes live on the network with one of the goals being to lower transaction fee variance (Blockworks, 2021).

The aim of this project at large is to find the impact of EIP-1559 on transaction fee variance. In the context of this project, the transaction fee variance will refer to how far each transaction fee is from every other transaction fee over a set time frame. This will be done by acquiring data from the Ethereum blockchain encompassing five distinct mean transaction fees which will be referred to as target gas fees. Each mean value will have three different days before and after EIP-1559 in order to facilitate a comprehensive view of the transaction fee variance before and after the update.

To conduct this research a series of objectives will need to be accomplished. The first being the creation of a data pipeline that acquires transaction fee data from a locally running node. Thankfully, blockchains pride themselves on the openness of their underlying data with the ability to run a node being a key metric on the success of a network. The second being the creation of a set of tools to parse, process, and visualize the acquired data. The third being the calculation of a number of measures that discern a nominal value that quantifies the transaction fee variance before and after EIP-1559. Lastly, this process will be repeated for the Polygon blockchain which is similar to Ethereum in its architecture and features a remarkably similar update. This analysis will provide further evidence on whether or not the impact of EIP-1559 is reproducible.

The optimal outcome of doing this research is to enhance the understanding of transaction fee variance within the Ethereum ecosystem improving the user experience. The target audience for this study comprises individuals with an interest in blockchain, Ethereum, and the impact of EIP-1559. The intended beneficiaries are those who utilize the technology for a number of purposes on a frequent basis and are substantially affected by fluctuations in transaction fees. In light of the time and resource limitations, the scope of this project has been deliberately narrowed, to instead focus on the depth of inquiry. The investigation is exclusively centred on discerning the transaction fee variance before and after the implementation of EIP-1559. However, this one problem has a number of sub-problems, which the research project strives to comprehensively examine.

This research makes a series of assumptions that are largely present in all research projects of this type. The initial assumption of this study is that the datasets procured from the blockchain accurately represent the transaction fee variance before and after EIP-1559. To ensure that this is mitigated the research comprises an extensive array of data points that captures a more comprehensive understanding of the Ethereum blockchains transactions. The second assumption posits that no significant factors influencing fee variance are exclusively present in either dataset. Previous anecdotal evidence suggests that larger transaction fees increase transaction fee variance. In an attempt to mitigate this possible factor, the data is divided into categories based on target gas fees as outlined above. This should ensure that comparable data is analyzed in relation to one another.

The comprehensive analysis conducted during this research project is done with the aim of uncovering insights into the impact of EIP-1559 on transaction fee variance that will be useful to other researchers and developers trying to make a more fair and stable system of transacting on a blockchain. By using a rigorous methodology and addressing inherent assumptions, the findings offer valuable perspectives on the Ethereum network's transaction dynamics. Ultimately, the outcomes of this study hold the potential to improve the user experience on Ethereum and similar blockchains, impacting developers, power users, and everyday people who find value in the technology.

2. Background

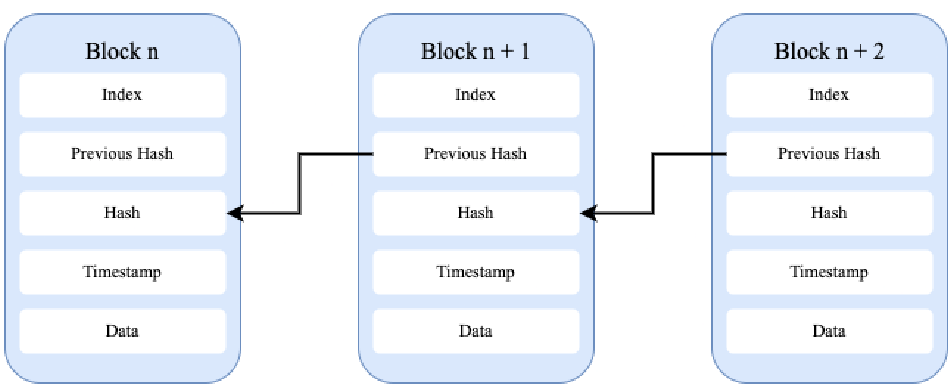
2.1. Architecture of the Ethereum Blockchain

Since its inception the Ethereum network has blossomed into the largest blockchain by both the amount of transaction fees paid to the network and by the amount of money moved on a daily basis (Mihal D 2023). Ethereum's closest competitor Bitcoin has one-fifth the number of transaction fees paid to its miners relative to Ethereum and routinely settles far less value than Ethereum. As a network, it sits in the centre of a fast-growing industry with a large number of participants wanting to use the technology leading to problems with transactions.

2.1.1. Validators and Proof-of-X

To understand these problems it's first important to understand Ethereum. At the core of any blockchain is the concept of a block, these blocks contain the data identifying the block itself and the transaction data that is stored in the block (corwintines et al. 2023). One important piece of block-identifying data is the parent hash, which identifies the block that came prior. With every block linking back to what came prior it forms a chain of blocks hence the term "blockchain". The value of collecting data into a large block instead of transactions being sent to the blockchain and confirmed individually allows the decentralized network to achieve consensus on its current state (corwintines et al. 2023). On the Ethereum blockchain blocks are created every 12 seconds, meaning that once a transaction is sent to the network it will take at most 12 seconds for the transaction to be officially recorded on the Ethereum blockchain.

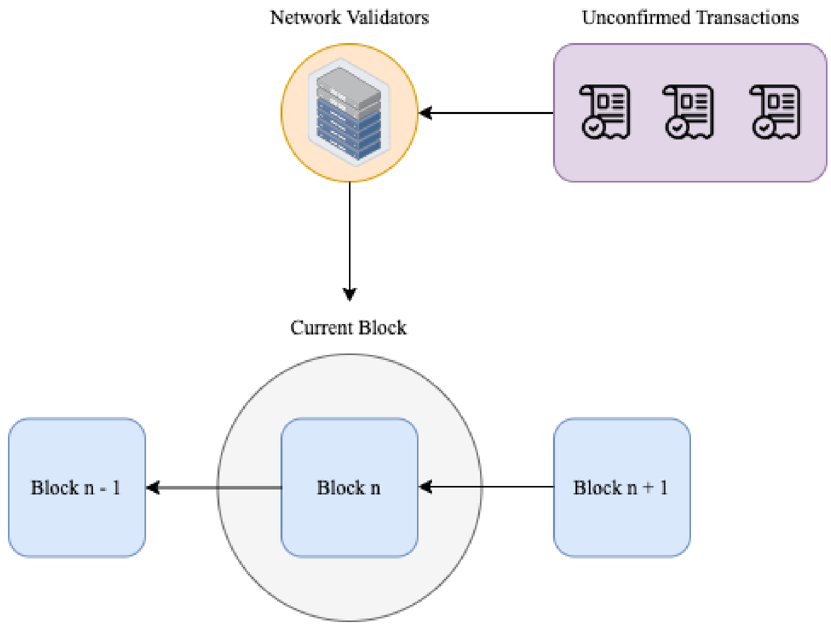
Figure 1. Blockchain Showing Link between Blocks



The three participants that allow this process to work are the validators, users, and nodes. Validators, within the Ethereum proof-of-stake blockchain, are randomly selected participants that have met a set of criteria. The most important being that they have staked 32 Ethereum and are actively running a Validator client connected to the network. Blockchains require their validators to stake something due to the “nothing-at-stake” problem. This is a theoretical problem in consensus algorithms that have a low cost of block production in that it makes sense for validators to attempt to create as many blocks as possible in the hope that one of the blocks they produce will be selected by other validators as the canonical chain (Dexter 2018). In proof-of-work systems there is a computational requirement to block creation, stopping miners from creating as many blocks as possible. They are staking the cost of the work they do on the blockchain, acting against the interests of the blockchain means they lose possible work that could have led to a reward. In proof-of-stake systems, block production is far cheaper, not requiring the large computational power in proof-of-work systems. Instead, validators stake the tokens of their respective blockchain that could be burned if they act against the interest of the canonical blockchain (Edington 2023). This would be equivalent to burning down proof-of-work mining operations that attempt to attack the chain, not only a strong enough threat to ensure that the chain won’t be attacked but in the theoretical situation that there is an attack, the blockchain removes this validator from the group of participants. The difference between all consensus algorithms including proof-of-work and proof-of-stake is what is at stake if a user decides to work against the interest of the blockchain.

In a proof-of-stake blockchain, validators are chosen at random to select the contents of the next block (LucaPenella 2023). This is in contrast to proof-of-work blockchains which Ethereum used to be prior to September 15th 2022. In a proof-of-work system, miners actively compete with one another to be selected for the next block by running complex algorithms to find a specific value (Nakamoto 2008). The processes for proof-of-work miners to find this specific value are completely arbitrary; it's only used as a means to fairly distribute both the revenue from mining blocks and the political power of choosing which transactions are featured in blocks. This system was adequate for the early stages of block production but as the networks started to grow clear problems such as electricity consumption and the rise of centralized mining pools, started to show (wackerow 2022). The research goes over transactions made before and after Ethereum made the move to proof-of-stake so from here on out both validators and miners will be referred to as validators. It’s also worth noting that despite the differences in the selection process of who chooses what goes into the next block, there are very few remaining differences between miners and validators.

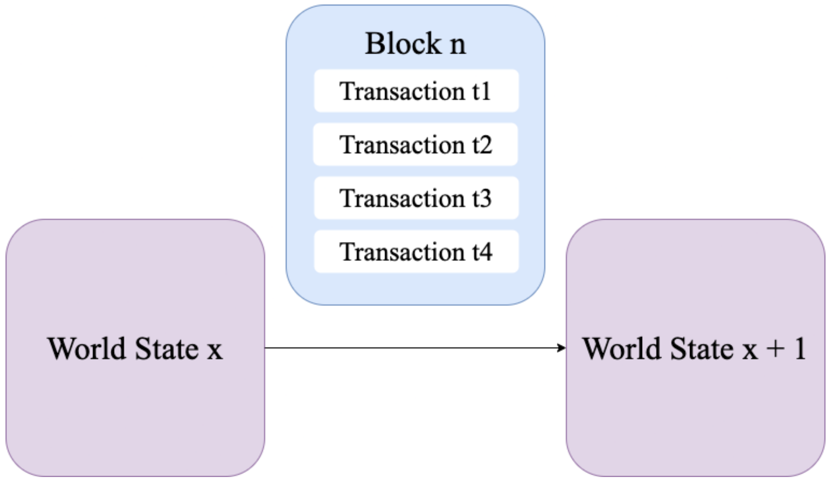
Figure 2. *Relationship between Blocks, Validators, and Unconfirmed Transactions*



2.1.2. Transactions and Transaction Fee Markets

Once a validator has been chosen by the network the validator has to decide what transactions made by users are featured in these blocks. Alongside blocks, user transactions serve as a fundamental piece of a blockchain. Unlike transactions in the modern financial world, transactions on the Ethereum blockchain can be much more than moving value from one account to another. These transactions aim to change the state of the blockchain. Transactions are made up of different pieces of data, the most important being, who is sending the money, the recipient, the cryptographically generated signature that can only be generated by the sender, the value being sent, and the gas fee. The gas fee of transactions outlines how much a user is willing to pay to have their transactions included in a block per gas. Gas is a calculation of the computational and storage requirement of the transaction itself, larger more complex transactions require more gas and therefore require higher fees. The price of gas fees is denominated in gwei, which is a unit of Ether, the Ethereum network's underlying currency (wenceslas-sanchez 2023). One gwei is equal to one billionth of an Ether and serves as a more granular measurement for transaction gas fee costs. Prior to EIP-1559, the transaction fee was a single value set by the user that is multiplied by the gas requirement of the transaction and would be received by the validator if the transaction is included in a block. A higher transaction fee serves as a greater incentive for validators to include transactions in blocks, but validators prior to EIP-1559 could include any transaction they want. This gives large groups of validators the ability to white list addresses that don't need to pay to use the blockchain.

Figure 3. Illustrating Transactions Changing Network State

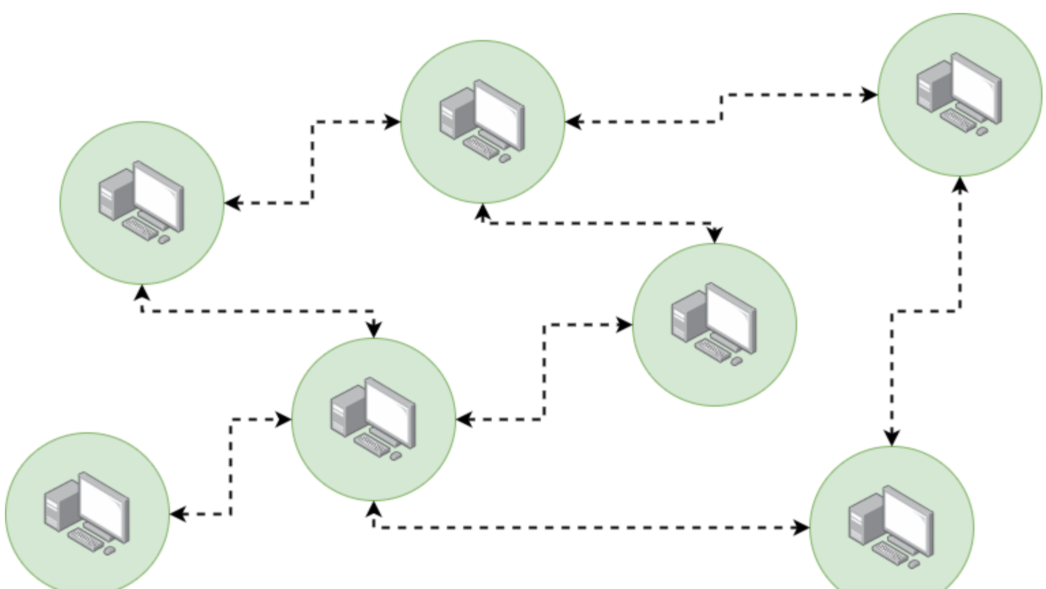


After EIP-1559 transaction fees are made up of three separate parts. The first is a transaction base fee per gas calculated by the blockchain, based on demand. For a transaction to be valid it must include the payment of this base fee. Users can then opt to include a max priority fee per gas, which is the maximum price of the transaction to be included as a further incentive to the validator. Finally, there is the max fee per gas parameter, that outlines the total transaction fee per gas that the user is willing to spend. With a base fee that moves dynamically users may find that in periods of high demand the base fee moves up rapidly and therefore the total cost of their transaction fee has moved far above what they are willing to pay, this max fee per gas is the option to create a maximum cap on the total transaction fee (wenceslas-sanchez 2023). When transactions are created they are first checked by the nodes of the blockchain, these are users that have stored the canonical chain, the new blocks added to the chain, and also define the rules for the blockchain (Orenes-Lerma 2023). If the transaction meets the validity requirements, it is moved into the mempool. The mempool serves as a holding area where transactions wait to be selected by validators. At first glance, the transaction fee game that users and validators play known as the transaction fee market may seem like an over-engineered solution to a simple problem. It instead would be far easier to include all valid transactions into a given block.

2.1.3. Nodes and Block Limits

Blocks on Ethereum currently have a set size restricting the total amount of transactions that can be included in this 12-second period, removing this parameter would also remove the need for an efficient fee market. Large blocks require large computational and storage requirements from nodes to store the transactions. Removing this bandwidth limit allows for standard DDOS attacks to occur on the network causing great instability (Hafid 2020). The Solana blockchain has far greater bandwidth than Ethereum and routinely has outages requiring centralized intervention from developers to contact node operators to restart the blockchain (Irwin 2022). The next obvious solution is to increase the computational and storage requirements of node operators to handle the number of transactions. Currently, Solana nodes require 12 core 24 thread CPUs, over 128GB of ram, and many terabytes of SSD storage (Solana 2023). Far greater requirements that can be expected of volunteers supporting a decentralized network. For reference, Ethereum nodes can be run on a raspberry pi 4 with a single terabyte of SSD storage (go-ethereum 2022). Increasing the requirements of node operators puts the rules of the network into the hands of a small group of people that can afford to run nodes. Having a modest limit to both the frequency and size of blocks is key to ensuring that the blockchain is sufficiently decentralized.

Figure 4. Structure of a Decentralized Network of Nodes



2.2 The Problem

2.2.1 Transaction Fee Variance

Due to a limited amount of block space and variable demand, the transaction fee to be included in a block changes from block to block. This transaction fee variance impacts the user experience when using the blockchain. Users often found it challenging to predict the necessary fees to ensure timely transaction processing, possibly leading to overpaying for transaction fees or much longer wait times. In the context of this research, variance is quantified using the var() function in the numpy library. This calculates variance by taking the average of the squared deviations from the mean of a set of data.

Prior to EIP-1559, users would submit transactions with a gas fee that they felt would be attractive enough to a validator to include their transaction in a block (Beck and Asher 2021). This guessing game by users created a series of inefficiencies that caused greater transaction fee variance. Furthermore, users commonly relied on external tools and services to estimate the necessary gas fee. These services all use different fee estimations based on a variety of different factors causing further frustration with users who end up overpaying or underpaying leaving the transaction stuck in the mempool. A frustration that only grew worse in periods of high congestion.

The combination of these problems created barriers for entry to new users and harmed the experience for older users. This greater barrier for entry discouraged the adoption of the many technologies that can be built on top of the Ethereum blockchain as the unpredictable and potentially extortionate transaction costs posed a challenge for both users and developers. These issues highlighted a clear need for a more predictable, fair, and user-friendly fee mechanism on Ethereum. EIP-1559 was created in an attempt to do so.

2.2.2. Features added through EIP-1559

The London Hard Fork added a series of changes to the Ethereum blockchain, the most important being the algorithmically determined base fee. As outlined above, this base fee is responsible for deciding the minimum cost to transact on the blockchain. This was added in an effort to stop the guessing game made by users and their tools when deciding on a transaction gas fee. Submitting a transaction with just the base fee should be sufficient for most transactions. This base fee changes based on the demand for the previous block using the equation (Yulin et al. 2022):

$$baseFee_{h+1} = baseFee_h(1 \ + \frac{1}{8}(\frac{gasUsed - gasTarget}{gasTarget}))$$

$h \ = \ current\ block\ height$
$gasUsed \ = \ the\ amount\ of\ blockspace\ used$
$gasTarget \ = \ the\ target\ size\ for\ a\ given\ block\ set\ at\ 15,000,000\ gas$

This base fee isn't paid to the validator, in fact, the base fee is burned, removing the tokens from circulation entirely. The only fee received by the validator is the transaction fee tip a user can choose to pay. This removes the incentive for validators to try and artificially create congestion, instead, the focus for validators is to process transactions efficiently rather than trying to extract the maximum value of users through fees (corwintines et al. 2023). Alongside the base fee, there is the transaction priority fee that serves as a tip paid to the miner in an attempt to include the transaction as soon as possible, and finally, there is the max fee, which is the most a user is willing to spend on the transaction (wenceslas-sanchez et al. 2023). If a transaction is submitted below the value of this max fee, a refund is submitted based on the following equation:

$$refund = maxFee - (baseFee + priorityFee)$$

$refund \ = \ ether\ returned\ to\ the\ user$
$maxFee \ = transaction\ fee\ cap\ set\ by\ the\ user$
$baseFee \ = algorithmically\ defined\ fee$
$priorityFee \ = \ tip\ for\ validator$

With the changes made to transaction fees, there were also changes made to the size of Ethereum blocks. Instead of having a set block size, the blocks change dynamically with demand similar to the base fee. Unlike the base fee, there is a maximum size that Ethereum blocks can be and a target block size that blocks aim to be. When a large amount of demand is created on the network, blocks increase in size to respond to this demand in an attempt to more efficiently process transactions (corwintines et al. 2023). There were other changes made to the Ethereum blockchain implemented by EIP-1559, but they have been set outside the scope of this project as they are unlikely candidates for large changes in transaction fee variance, and therefore won't be discussed. However, it is likely that the changes outlined above have had a positive impact on transaction fee variance, and that it is important to conduct research on discerning the nominal amount that transaction fee variance has been impacted.

2.2.3 Impact of Transaction Fee Variance Research

The goal of the research is to add to a relatively small knowledge base on the difference in transaction fee variance before and after EIP-1559. The maximum possible influence of this research would be a small influence on how future Ethereum updates impact the transaction fee market on Ethereum. The three stakeholders in Ethereum are all impacted differently but both validators and nodes are impacted far less than the users of the blockchain. Validators may be impacted by lower revenue as large parts of the transaction fee are burned, but they also get token emissions and much lower operating costs by moving from proof-of-work to proof-of-stake. Despite the differences in how fees are priced the laws of supply and demand dictate that as long as Ethereum block space is valuable users are going to pay to use it. No amount of research into this area is going to change this fact. Nodes are also in a similar position to validators, these participants store the different transactions, and none of the changes to the fee market are likely to impact the size of the blocks and their underlying transactions. There may be small spikes in the size of blocks in periods of high demand, but most modern computers are more than equipped to handle these changes. No amount of research into transaction fee variance is going to change this fact either. Users on the other hand are impacted enormously by a greater understanding of what impacts variance. Over the long term, a user may pay the same amount of fees regardless of how variant the transaction fees are on the Ethereum blockchain, but on a block-by-block basis, similar users will pay far different transaction fees for the same transactions. This poses a clear UX problem that may drive users away to other blockchain networks or away from blockchains as a whole. It would be similar to someone's internet bill changing drastically on a month-by-month basis, even if the average remains at a reasonable level over the long term, a large unwanted bill could be very inconvenient for the end consumer. For Ethereum to scale to serve billions of users all across the globe, it is pivotal that transaction fees that users pay on the network are fair and stable, this research intends to help in achieving this goal.

2.3. Literature Review

2.3.1. Problems with Finding Relevant Ethereum Research

Due to the nature of the blockchain space and how recent EIP-1559 occurred, the research that has been done on the area of interest is minimal. Blockchains as a whole are very complex using many different bleeding edge technologies in tandem with one another. This wide range of technologies that are possible to research means the amount of research done on a specific topic is likely to be minimal. This complexity also makes it much harder to conduct research as isolating a single part of the system can be difficult to impossible. Alongside this the Ethereum blockchain specifically has gone through a period of rapid iterative improvement which can make research outdated months after it's published. Researchers who wish to wait for many years before conducting research on the Ethereum blockchain to ensure what they are researching is still relevant won't have published their research on the field yet as EIP-1559 is only a few years old. This is compounded by the fact that conducting comprehensive research may take many years. Expecting a large body of research only two years after an update in such a fast-moving field is very hopeful. Due to this lack of research, it also means that there aren't any known methodologies or best practices on how to conduct research, most likely driving more researchers away from the field. Knowing this I understand that I have a challenge in conducting research here as well, but I feel as if it is my duty to do so. This field is important to me as I have been personally impacted, and have seen other users driven away to worse technologies because of high transaction fee variance.

2.3.2. Empirical Analysis of EIP-1559 Review

At present, the most relevant piece of research examining the impact of EIP-1559 on transaction fee variance is "Empirical Analysis of EIP-1559: Transaction Fees, Waiting Times, and Consensus Security" (Yulin et al. 2022) as the paper suggests it takes a much larger view onto researching the impacts of the update, and although this study offers a comprehensive view into understanding the impacts of EIP-1559 the depth of research concerning transaction fee variance is minimal. The research acquires data from the Ethereum blockchain, the Ethereum mempool, and exchanges to investigate the consequences of EIP-1559. The study is able to conclude that although gas prices remain at similar levels over the period of the research, the variance in gas fees and the wait time for transaction confirmation is lower post-EIP-1559, therefore, improving the user experience.

The method employed by the researchers to acquire the data for their research isn't feasible for the present study. Rather than relying on pre-existing data from the Ethereum blockchain, they used data acquired from the Ethereum mempool which is where transactions are stored before being accepted into the block. The mempool state isn't stored for any long period of time and may be different depending on how and when the data was captured. This required the researchers to create a distributed data collection system with nodes in different locations across the globe to obtain an accurate representation of events. This approach was necessary for their research as one of their measured variables was how long transactions stayed in the mempool before being confirmed by the Ethereum blockchain. For the research conducted in this paper, this variable won't be analyzed and therefore only data from the Ethereum blockchain is needed.

The researchers observed that there are a large number of outliers in the dataset for transaction fees, prompting the use of the median and IQR to calculate the gas fees for a given block. I can see the advantage of using this method, but given that, I also want to have intra-block data, I can't solely rely on block average transaction fee data. Instead, this research will remove the outliers by employing a standard IQR model, facilitating a better understanding of how the majority of users interact with the blockchain without the distortion caused by a small minority of unique users.

Furthermore, the researchers also analyzed the adoption rate of EIP-1559 transactions. After EIP-1559, transactions submitted to the Ethereum blockchain may still be in the legacy format with a single fee paid to the network. As long as this fee is greater than the base fee it is a valid transaction. The study discovered that the adoption of EIP-1559 transactions had a positive impact on the variables measured, which I believe warrants further investigation. However, The research conducted in this paper is constrained by a lack of available data, as it only considered information from approximately 100,000 blocks following EIP-1559. This limitation may have had a significant impact on the results the researchers acquired from their study. I believe that this warrants a new investigation into what was studied, something this research hopes to accomplish.

2.3.3. Transaction Fees on a Honeymoon Review

The next closest solution to the problem was the research paper "Transaction Fees on a Honeymoon: Ethereum's EIP-1559 One Month Later" (Reijsbergen et al. 2022). The research presents a noteworthy approach to identifying inefficiencies in the transaction fee market and proposes ideas on how the market for block space can be more efficient. The research examines the base fee changes during periods of transaction fee instability concluding that the static rate of change to the base fee is inferior to a variable rate of change.

Despite the paper's focus on the transaction fee market, the paper doesn't provide a comprehensive view of transaction fee variance before and after the implementation of EIP-1559. Rather, the paper focuses heavily on potential improvements to the base fee mechanism as a means of addressing further transaction fee variance. The research paper also suffers from a lack of data. As the title suggests. The researchers relied on a startlingly small one month of data post-EIP-1559. The present study benefits from a wealth of knowledge that far surmounts what was available to the previous researchers hopefully leading to more valuable results

By far the most interesting part of the research is the construction of a simulation to test the variable base fee model the researchers proposed. This approach is very appealing as it allows the researcher to test hypotheses from data that doesn't exist. However, a focus on building large and complex simulations may focus the research away from transaction fee variance towards building a suitable simulation to test transaction fee variance.

2.3.4. Dynamical Analysis of the EIP-1559 Ethereum Fee Market Review

Less relevant to the two pieces of research prior, but still providing some insight, the "Dynamical Analysis of the EIP-1559 Ethereum Fee Market" (Leonardos et al. 2021) offers a different way of analyzing the London hardfork. The researchers in the paper take the transaction fee market created through EIP-1559 to its extremes in a simulation that attempts to stress test the base fee mechanism. This was done both in theory and using a simulation the researchers designed to better understand how the base fee regulates transaction fees and block occupancies when pushed to its limit.

The research discussed conducted a very different experiment, using a method that I'm unable to do. Despite this, the research does offer deep insights into how the transaction base fee operates on a regular basis, and during extreme scenarios. This offers the ability to understand the data acquired from the present study in much greater detail.

2.3.5. Transaction Fee Mechanism Design for the Ethereum Blockchain Review

Written far before EIP-1559 was implemented on December 3rd, 2020 Tim Roughgarden, the researcher who conducted the study, analyzes the update in its entirety

(Roughgarden 2020). The research takes a wholly theoretical approach by creating mathematical proofs to either disprove or prove the arguments for or against the update. This was in an effort to conclude whether or not it would be beneficial to include EIP-1559 in the Ethereum blockchain. The research finds that EIP-1559 would have many benefits to Ethereum with the most relevant to the present study being, easier fee estimation in the form of an obvious optimal bid, and lower transaction fee variance due to increased flexibility in block sizes.

In many ways, the present study builds directly on top of the research done by Tim Roughgarden. Although the methods used in the research aren't relevant due to the two very different circumstances of both studies, it offers insight into what should be prioritized when analyzing the results. The difference in transaction fee variance will be made clear by the research, but there is also a possibility that a bi-product of this research is an insight into whether or not EIP-1559 provides more accurate transaction fee estimation. When conducting the present study, greater analysis will be conducted on the graphical visualizations to understand if there is a tangible difference in transaction fee estimation that can be detected.

2.3.6. Literature Review Conclusions

Despite the limited number of research papers focusing on the EIP-1559 update, the existing literature that does exist provides valuable insights into conducting similar research. The first article examined emphasizes the very clear importance of removing outliers to prevent skewing the dataset. This observation resonated with the initial findings of the research presented in this paper as I noticed a very similar impact on my data particularly with the growth in fringe use cases of the Ethereum blockchain that tend to favour the use of high transaction fees per gas. Alongside using outlier-removed data, the referenced research also highlights that there is an advantage in using the IQR and median over the average for pertinent statistical measurements. Due to the values on the extremes of the data, the approach offers a better understanding of the dispersion and the central tendency.

Furthermore, the second piece of research explores the possible use of simulations for investigations into the transactions on the Ethereum blockchain. Instead of building intricate simulations for the research, the use of simple regression analyses could provide substantial value to the research. I propose that variance changes based on the average gas fee of the block with higher average gas fees corresponding to higher variances, and lower average gas fees corresponding to lower variance. Building regression models could allow for the derivation of a formula that yields a more accurate representation of variance compared to a single statistical measure. The final two pieces of research discussed didn't offer knowledge on conducting research but rather offered knowledge on important aspects of the EIP-1559 update that will heavily influence the present study including the dynamics of the base fee and many of the intended consequences initially proposed prior to EIP-1559's deployment.

2.4. Methods and Tools

2.4.1. Ethereum Node

To acquire the data, it's necessary to connect to the Ethereum network locally by running an Ethereum node on a home computer. An Ethereum node is made up of two parts, the first being Geth which is the execution layer that is responsible for managing the transaction data, and the second being Pryzm which is the consensus layer that is responsible for managing the validator data. These tools were chosen due to my previous experience with the technologies alongside their proven stability. As Ethereum node technologies can be complicated, it's common for nodes to fail and stop effectively recording the history of the blockchain. Therefore the tools chosen were prioritized over others due to them being well-known for their liveness and ease of use.

2.4.2. Web3.py

To communicate with the Ethereum node, the Web3.py library was employed in conjunction with Python. This library features built-in functions that allow for the acquisition of data from the Ethereum blockchain. These functions were used to acquire the relevant information that was necessary to conduct the research. This tool was chosen due to its wealth of documentation, and due to the lacking number of competitors that offer the same number of tools.

2.4.3. Other Libraries

Alongside these two technologies, standard data science libraries and tools were used such as pandas, numpy, scipy, and scikit learn. These were used to extract, process, and analyze the relevant information in an efficient manner. These tools were chosen due to my experience with the above technologies, and due to their well-known popularity in the field of Python programming and data science.

2.4.4. Constraints

Within the research project there are a variety of limitations derived from a lack of resources, the speed at which the research needs to be completed, and my knowledge of the subject matter. This led the research to focus greater on areas that best suited the available resources. The first decision made was to focus the research heavily on a small part of one update. This focus, despite preventing a large portion of the possible breadth of study, has created the environment to enable a much larger depth of study within the given time constraints. The resources available are the technology that I had prior to the research that I owned. The main resources are a modern laptop, a computer with 4TB of SSD storage, and an internet connection. The amount of data retrieved on the subject matter is exceptional considering these resources, and I believe that stretching them any thinner by trying to analyze a greater variety of possible impacts created by EIP-1559 would have only hampered the quality of the research. My knowledge of the subject matter is two-fold, the first is my knowledge of data science, and the second is my knowledge of blockchain networks. Despite having a deep interest in both, my knowledge of data science is greatly inferior to my knowledge of blockchain networks. Hence why the research intends to use basic data science concepts very well over using more advanced data science concepts poorly. Furthermore, my knowledge of blockchain networks allowed the research to consider a much broader number of possible challenges that could be mitigated prior to the study taking place.

2.5. Objectives and Research Question

Using the knowledge acquired from the prior study, the research intends to conduct a comprehensive analysis of transactions and their respective fees that occurred on the Ethereum blockchain. This is with the intent of answering the following research questions: What is the difference in transaction fee variance before and after EIP-1559? And, is this difference reproducible on another blockchain? To best answer these questions the following set of objectives need to be accomplished:

- Build an effective data pipeline that acquires transaction fee data from a locally running Ethereum node.

- Build a set of tools that parse, process, and visualize the acquired data.

- Calculate a number of measures that discern a nominal value that quantifies the transaction fee variance before and after EIP-1559.

- Repeat this research for another, similar blockchain, that has implemented a similar update.

It's my belief that it's best to run an Ethereum node locally instead of using a cloud provider when possible, as the research is not only conducted in the spirit of a decentralized network, but it also ensures that the data acquired hasn't been manipulated. The set of tools to parse, process, and visualize the data will be built during the course of the research as already existing tools are rare and hard to find, and to provide more tools for other researchers to better understand transaction fee variance on blockchains. Multiple transaction fee variance values will be calculated to provide a greater breadth of analysis because there are few methodologies and best practices for this research that currently exists. Finally, reproducing this same experiment on another separate blockchain will produce incredibly valuable results when coming to a firm conclusion. By accomplishing the above objectives, it is my belief that the research question will be answered confidently with strong evidence to support the validity of the answer while also meeting the constraints of the project.

3. Specification and Design

The system that was built to conduct the research is made up of three distinct parts; data procurement, data processing, and data visualization. When all this is put together in tandem the system outputs clear and empirical transaction fee variance statistics that can be compared with one another.

3.1. Data Procurement

The process of procuring data started by identifying days with similar average gas fees that occurred before and after the implementation of EIP-1559. Utilizing a resource provided by etherscan.io (Etherscan 2023), which displays daily average gas fees, the goal was to find three days prior to and three days following the update that had average transaction fees per gas in gwei approximating 25, 50, 75, 100, and 125. The selection of these specific targets was based on their comprehensive representation of the majority of transaction fees that occur on the Ethereum network. Alongside this incorporating multiple days taken from multiple periods allows for more resilient research as opposed to relying on singular days potentially influenced by single outlier events and circumstances. The difference between the gas fee targets was also sufficiently significant to find any tangible differences that occur from target to target. The average transaction gas fee range deviated by 6 gwei from the corresponding target gas fee. The deviation was deemed essential, as precise alignment with the selection of target gas fees is rare. A 6 gwei range was chosen to include days with over a 5 gwei range, a seemingly reasonable estimation of a sufficient range, that also included decimal figures such as 30.5 or 55.2. Furthermore, despite the requirement of six days of data for each target gas fee, an exhaustive search was conducted to identify almost all days fulfilling these criteria, thereby enabling the selection of days that most closely fit the desired criteria.

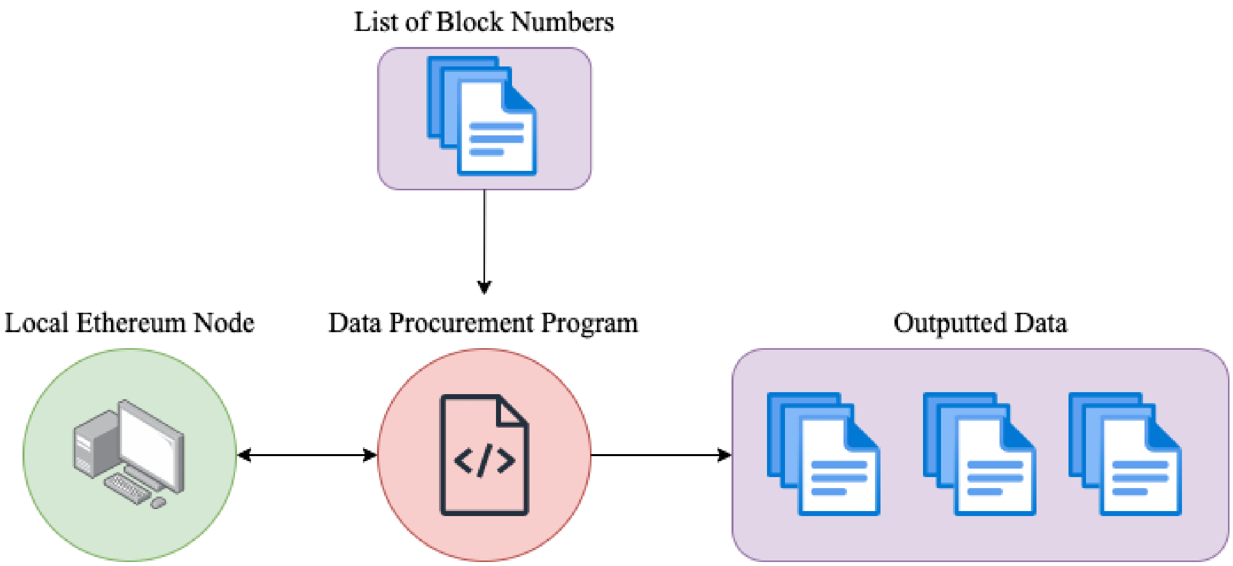
The data produced so far comprises a list of dates and their corresponding gas price criteria. Due to the absence of a pre-existing program that's able to convert the dates into block numbers, a bespoke solution was developed. This entailed modifying a previously created program, initially designed to track the historical throughput figures of different blockchains. Due to the nature of both problems, the solution could be quickly transformed into one capable of identifying the first block of any given day, based on a Unix timestamp.

By connecting to a locally running Ethereum node through the web3 library, the program takes the timestamp of the previously acquired day, the latest block on the specified blockchain, and a starting block number. Subsequently, the program takes the mean of the two blocks and checks if this new block's timestamp is above, below, or equal to the desired block's timestamp. If the found block timestamp is larger than the block we are trying to find, the latest block number is set to the current block number, if the opposite happens the first block number is set to the current block number. This is then repeated until the block with the given timestamp is found or as a fail-safe if the program is caught in an infinite loop. The program was developed with an emphasis on solving a problem quickly, rather than solving a problem competently. While the results generated may not have always been accurate, they were always sufficient to enable a manual identification of the first block corresponding to a given day.

Now that the data set includes a series of block numbers rather than a set of dates these blocks can serve as the inputs to a program designed to retrieve every 72nd block from the 7200 estimated blocks within a day. This process retrieves a file containing 100 blocks, along with their associated transactions from the Ethereum node. The specific output generated by this program is a CSV file which includes the block number, the transaction hash, the transaction gas price, the transaction base fee, the transaction priority fee, the transaction gas usage, the total cost of the transaction, and whether the transaction was new or legacy. An abundance of data was taken from the Ethereum blockchain, to accommodate the potential discovery of valuable information that could be relevant for future analysis.

Upon the completion of data collection for all of the specific days, the next objective focused on verifying the gathered data to ensure that the average of all the transaction fees was close to the averages for their specific criteria. The transaction fee data that has already been acquired is priced in wei, which is the smallest unit of Ether, being one billionth the value of a gwei. To get all the data in line with each other, a program was run that averaged all the data within the transaction gas price column of the CSV file and divided it by 10^9 to get the gwei value. These new gwei values were then compared with the target values to ensure they met the necessary criteria. Having acquired a sufficient volume of data both pre and post-EIP-1559 implementation that's sufficiently close to the targeted gas prices the data was then put into designated folders facilitating the next step of the analysis.

Figure 5. Illustration of Data Procurement Process



3.2. Data Processing

Upon the acquisition of the data, it needs to be processed further prior to the execution of more advanced calculations. This is accomplished through the use of the pandas library within a jupyter notebook using Python. The CSV files are added to the notebook as pandas dataframes and a new column is added called 'transaction gas price gwei', which adds the average gas price of each block to its gwei value. After the fact, a new dataframe is generated with all the outliers removed. This is accomplished by removing all data that is above or below 1.5 times the upper and lower quartiles. The decision to exclude outliers using this method was based on guidance from a similar research paper discussed in the literature review.

The next part of the data processing introduced a "normalized variance dataframe", this dataframe was created in an attempt to be able to compare legacy and new datasets in as close a manner as possible. In order to generate the normalized variance dataframes, the three distinct days of data for both legacy and new transactions were combined into two large dataframes, one dedicated to the new transactions and the other to legacy transactions. Subsequently, the transaction gas prices were then grouped by their respective block number, followed by the calculation of the average for the transaction gas prices within each block. Blocks falling outside a predetermined range of gas prices set between 20% to 40% away from the target gas price were also removed to ensure that the analysis couldn't be skewed by lower or higher gas prices found within certain blocks. When analyzing the larger gas prices it was necessary to use larger percentages to ensure that there was enough data to use in these dataframes, as the days with larger gas prices tended to have larger variances of average transaction fees for each block.

To conduct the analysis it was necessary to find the corresponding transaction fee variance for each block based on its contents. This required a new column in the data frames that featured this figure. For further more complex analysis, the normalized dataframes were then ordered by variance and normalized in length. This was accomplished by creating a new dataframe that only featured one instance of each block and their respective variance, instead of multiple instances featured next to the transactions that took place within the block. When creating dataframes of all the same length, it was important to ensure that data wasn't removed from any specific part of the dataframe, such as the middle, beginning or end, as this could skew the data. Instead, the 'linspace' function from the numpy library was employed which finds the required distance between each item, for the number of items one wishes to remove, to ensure the items being removed are equally distributed across the dataframe. This was done so that each dataframe had a length of 100 rows. Another column was added referred to as 'block_id' which denotes the position of the block in the dataframe when organized by variance. This allows for the creation of dataframes featuring the top 10 lowest and 10 highest variance blocks for legacy and new blocks or for analysis to be done ordered by variance. Alongside processing the legacy and new dataframes another dataframe referred to as 'only new' was created. This was done in an attempt to understand if blocks featuring only new transactions would have significantly different transaction fee variances. Basic statistical calculations were then done such as the mean, median, range, variance, standard deviation, and inter quartile range for each of the six original dataframes and their outlier removed versions which enabled a better understanding of the fundamentals of the data. The number of outlier data points that were removed was also calculated to make sure that the data wasn't changed in any significant manner.

3.3. Data Visualization

In order to best understand the data, a variety of data visualization techniques were employed. The initial set of data visualizations aimed to concentrate on the outlier removed dataframes for legacy, new, and only new. For each dataframe, the most basic visual analysis was done through two plots, the first being a set of histograms and the second being a set of scatter plots. The histograms featured the number of transactions on the Y-axis and 1 gwei wide bins on the X-axis. These histogram plots would show clearly how varied the transaction fees were over the course of each of the three given days, with the best plots being those where most of the data is heavily centred around the gas fee I was measuring with the worst plots having very spread out transactions in a wide range of bins. Within the histogram plots, I also made sure to include the kernel density estimation which creates a smooth curve over the data to better understand the shape of the distribution. The least varied data should have one clear peak while more varied data should feature a greater number of peaks.

The second visualization technique, the scatter plots, used the same dataframes with the aim of finding very similar patterns but from a different perspective. The scatter plots show a greater granularity to the dataframe which enabled an insight into the patterns that form within the transaction fees of each block. The dataframes that featured outliers weren't used as through the statistical calculations it was found that the outliers did in fact skew the data to a point where it wasn't useful in aiding comprehension of the data in any meaningful manner.

Following the visualization of the standard data, the objective shifted towards more creative approaches to analyzing and representing the diverse dataframes in an attempt to aid comprehension. The first example of this was the creation of box and whisker plots of the ten most and least variant blocks within the normalized dataset. This investigates whether there are influential edge cases where a minority of blocks are exceptionally variant or not-variant than previously understood. Taking this idea one step further, a scatter plot was created that featured all the transactions that occurred within the 100 blocks ordered by variance to see the rate at which blocks become less variant. This was one more step forward to a comprehensive understanding of the variance differences between the different dataframes, but despite the analysis offering fantastic granularity of the individual block's data, it lacks the use of empirical calculations and relies more on a viewer's opinion of what is more variant. This idea of measuring the consistency of transaction fee variance was taken one step further to its logical end through the creation of block variance curves. This line plot uses the variance calculations of each block from the three normalized dataframes legacy, new, and only new overlapping with one another to produce an insight into the clear variance differences of each block from most to least variant.

Despite what these graphs offer in communicating the difference in variance there was at no point a decisive mathematical calculation of the difference in variance. Taking any of these dataframes and calculating a standard variance figure seemed lacking given what was now known of the data, it simply doesn't tell the whole story. I, therefore, created a more descriptive calculation using a linear regression model and a curve fit model. Three linear regression models were created, the first takes the variance data from the new normalized variance dataframe as the X-axis, and the legacy normalized variance dataframe as the Y-axis. This information is then plotted and a linear regression calculation is used. The same is then done with only new data being the X-axis and legacy data being the Y-axis, and then once more with only new data being the X-axis and new data being the Y-axis. After these regressions are calculated and an equation is given they are then graphed on top of a scatter plot of the real data. It was believed that calculating the difference in variance between the different dataframes would also be fruitful. This was done by taking the differences between the variances of two dataframes then setting that as the Y-axis and using the block id as the X-axis. These calculations allow for a deeper understanding of block variance relative to other blocks. It doesn't just answer the question, what is the median or average variance of a given set of blocks? It also answers questions such as, are more variant blocks from before EIP-1559 more or less variant than the more variant blocks after EIP-1559? And, what is the rate at which blocks become more or less variant? Although this analysis has the core assumption that the two sets of blocks that are compared are identical comparisons, the research intends to counteract this with a large breadth of data.

After all of this analysis, a variance report card is generated at the bottom of the jupyter notebook that aims to put a figure on variance using basic measures, standard measures, and more advanced measures. The beginning of this section breaks down the mean, median, and counts of all the standard outlier-removed dataframes and the outlier-removed normalized variance dataframes. This builds context for the variance figures and ensures that it is reasonable to compare the dataframes. The basic analysis then finds the variance between all the legacy, new, and only new dataframes with and without the outliers. The standard analysis then finds the mean, and median block variance figures for the legacy, new, and only new outlier removed normalized variance dataframes. This then goes one step deeper and finds the mean and median for the low-variance and high-variance dataframes. Finally, the advanced analysis uses the equations generated to find the mean and median differences in variance derived from the created regression and curve models. Through all the different manners in which the variance is calculated, it's believed that a clear conclusion can be discerned from the results. Featured in this research paper are the results from the standard and advanced analysis with the context and basic analysis being omitted for brevity. If the included results found within the paper aren't sufficient, all the findings from the research can be found in the attached jupyter notebooks.

3.4. Analysis of other EVM blockchains

Due to the success of EIP-1559 on the Ethereum blockchain other EVM blockchains aimed to create a similar update to add to their blockchain. EVM stands for Ethereum Virtual

Machine which is the brain of the Ethereum blockchain allowing developers to build on top of the technology creating their own applications. EVM blockchains are very similar to Ethereum with minor changes made which usually increase throughput in exchange for lowering decentralization. One of these blockchains is called Polygon which has two-second blocks instead of twelve-second blocks with the same gas limit greatly increasing throughput. As of January 2022, Polygon implemented a very similar update to EIP-1559 which means that a similar analysis can be done. Doing analysis on other EVM blockchains is hard as the lower decentralization for a higher throughput trade-off means that running a node is far harder. Polygon nodes have much higher storage, ram, and compute requirements that are able to be acquired. On the other hand, much deeper analysis has already been done on Ethereum allowing for less evidence to support the claim. Instead of having five different target gas prices the research is only composed of one target gas price being 82.5. This value was selected due to the ease of acquiring data that was close enough to be within the requirement. Many days featured vastly dissimilar gas prices but there were in fact six days three before EIP-1559 and three after that were six gwei away from 82.5. This information was retrieved from Infura which is a cloud node solution that allows anyone to access data from a variety of blockchains and use the data in any way they see fit. When the days were acquired the same process took place, the first block number of each day was found, 100 blocks from each of the days were acquired, the data was processed in the same manner, and then finally visualized in the same manner.

4. Implementation

4.1. Introduction

A large segment of the code will not be discussed in exhaustive detail. Instead, the focus is directed towards a part of the systems that are either of great importance to the project and aren't self-explanatory, or those that, despite their simplicity, have proven exceptionally unique in their lack of clear and satisfactory solutions. The underlying motivation for this approach is in the hope that this will provide an additional resource for computer scientists who attempt similar projects.

4.2. Operating an Ethereum Node

Running an Ethereum node is deceptively challenging due to its distinctive hardware requirements and the many obscure, yet essential, software applications that can be difficult to use especially in tandem with one another. Furthermore, there are a number of different types of Ethereum nodes that can be run, all of which offer a different amount of data, and require different hardware. The type of node with the lowest hardware requirements is a light node which has the primary objective of facilitating a connection with the Ethereum network without an intermediary to allow for anyone to communicate with the network. Despite its accessibility, a light node doesn't serve as a platform for Ethereum development, and can't be used to access the data necessary for most Ethereum research. Therefore, the light node is not relevant in the context of the research project and is disregarded (Sardan 2018).

The node with the highest hardware requirements is an archive node, which stores a comprehensive state of the Ethereum blockchain at every block height. This allows for the deepest possible analysis to be done on the Ethereum blockchain at a very high speed (corwintines et al. 2023). These features come with an enormous hardware cost to be fulfilled by the end user, with the execution client of an archive node possibly requiring 12TB of storage space. As spinning disk storage isn't fast enough to keep up with the blocks being downloaded, the user must also have SSD storage which comes at a greater monetary cost (go-ethereum 2022). Combining this with the storage requirement for a consensus client and for the vast majority of users this type of Ethereum node is inaccessible.

For most research purposes, including the research done for this project, full nodes offer all the necessary data at much lower hardware requirements than archive nodes. A modern computer with a 4-core CPU, 8GB of ram, and 1TB of SSD storage is enough for a stable node (go-ethereum 2022). With hardware requirements that can be found on most modern PCs and many laptops the premise of running a full node is alluring. The shortcomings of a full node is that it only aims to store the most recent set of blocks and vital historical data. This historical data includes the transaction history, the event logs from smart contracts, and the block data, which is more than sufficient for this research.

To complicate things even further for Ethereum node operators, there are a variety of different clients that are available. Full nodes from different clients all have the same end goal but are built differently to ensure the network stays live if there was an unknown bug or exploit in one of the clients (Awosika 2022). The most stable client currently is Geth which is the Ethereum node software built in the Go programming language. A node operator that was focused on helping the network flourish should choose a minority client, as greater client distribution is beneficial for the stability of the network. As this node is aimed at acquiring data to conduct research, the most stable client is the best. On top of an execution client, an Ethereum node also needs a consensus client. From a consensus client, a node operator can acquire validator information, beacon chain blocks, and attestation information, none of which is relevant to the research being done. This allows for the consensus client to be started at a checkpoint, which starts downloading information from a much later date instead of the consensus layer's entire history. The consensus client used for this project was Prysm due to its popularity and stability. After downloading Geth version 1.11.4 and Pryzm version 3.2.2 I ran the programs using the commands featured below.

Geth Command:

```
$ ./build/bin/geth -datadir=/media/user/EthBlockchain/geth
--authrpc.jwtsecret ../consensus/jwt.hex --txlookuplimit 0 --ipcpath
/root/.ethereum/geth.ipc
```

Prysm Command:

```
$ ./prysm.sh beacon-chain --datadir=/media/user/EthBlockchain/prysm
--slots-per-archive-point 64
--execution-endpoint=http://localhost:8551 --grpc-max-msg-size 8388608
--jwt-secret=/home/user/ethereum/consensus/jwt.hex
--checkpoint-sync-url=https://sync-mainnet.beaconcha.in
--genesis-beacon-api-url=https://sync-mainnet.beaconcha.in
```

The Geth command is relatively self-explanatory, but a major unforeseen problem was the necessity of correctly setting the "--txlookuplimit 0" parameter. Without this parameter, the node only stores recent transactions. When setting the parameter to 0 it stores all the transactions that occurred on the chain which was necessary for the research that was conducted. The Prysm command is also largely self-explanatory except for the parameters "--checkpoint-sync-url=" and "--genesis-beacon-api-url=" both these parameters set the value for where the consensus client finds the point to start from. The resource beaconcha.in was used as the source for the checkpoint due to its popularity and long-standing positive reputation in the Ethereum community. Setting these checkpoints improved node sync speeds dramatically allowing the research to begin an entire day sooner than it would have otherwise.

4.3. Creating Normalized Variance DataFrames

Another key part of the research was in creating the normalized dataframes. These were used in all of the more advanced calculations of variance and are yet to be described in depth. The code is in two parts, pruning and ordering, the pruning section ensures the dataframes contain the correct values, the ordering section ensures the values are in the correct position in the dataframe

4.3.1. Pruning

The first step in pruning is combining the three collected days for either legacy or new data. This large dataframe then groups the data by the block number the transaction was featured in. For each of these block numbers and average of each transaction gas price, this is important as it's needed to remove all the blocks that aren't within a set range of average gas prices. This range is set dependent on the target gas price, and aims to only feature blocks that are close enough to the target gas price to be relevant to the research. Finally, a new DataFrame is created containing only rows with block number values that fit within this range.

4.3.2. Ordering

Given the pruned DataFrame, the variance of each block is then found using the var() function in the numpy library. This takes the average of the squared deviations from the mean of the gas prices per gwei of each transaction featured in a block. These blocks are then sorted by this variance calculation. Each DataFrame, from the legacy, new, and only new types, from each target gwei, are of different lengths, and unfortunately a large part of the analysis is dependent on these DataFrames being equivalent in the number of blocks featured. Removing a set of blocks from the beginning, end, or middle of the Dataframes could skew the data. Therefore the use of the numpy linspace() function was employed to remove evenly spaced blocks across the DataFrame until it contained 100 rows. After the fact, a unique ID is assigned to every block within each DataFrame that ranks the blocks on their variance.

The final result is two dataframes, the first contains all the transactions within each block, the other just contains the blocks and the statistics of the data within the block. This was done because different types of analysis have different requirements, some require a more granular view where it's possible to see every transaction, others only require the statistics generated from the transactions within each block.

4.4. Model Creation

The models created in the advanced analysis were used to arrive at a more exact comprehensive estimate for the variance calculation in the form of a linear and cubic equation. The code is in two parts, the first creates the linear regression model the second creates the curve model.

4.4.1. Linear Regression Model

The linear regression code uses the sklearn.linear_model library to create the linear regression model. This library takes the data from the new normalized DataFrame of a given type and from a given target gwei on the X axis, and also takes the data from the legacy normalized DataFrame of the same type and same target gwei on the Y axis. The library then fits this data into a linear regression model, and generates a list of values representing the line of best fit, alongside the equation representing the regression model. The same was then done for only new and legacy DataFrames, and also only new and new DataFrames for each target gwei.

4.4.2. Curve Model

The curve model code uses the curve_fit function from the scipy.optimize library to fit a cubic function that represents the difference in transaction fee variance when comparing two types of DataFrames. The curve model takes two parameters, the first is the ID representing the variance of the data as the X axis, the second parameter is the variance difference of two types of data by each block as the Y axis. The combinations include the difference in variance of legacy and new data, legacy and only new data, and finally new data, and only new data for each target gwei. An equation that represents the curve is then produced, and a corresponding list of values is generated which is later graphed. It’s important to note that a cubic curve was chosen as analysis revealed that it provided the most accurate fit for the different dataframes.

4.5. Miscellaneous Problems

Investigations into the Ethereum blockchain always offer unknown problems. The first problem stems from how frequently the Ethereum ecosystem evolves. From when my first personal investigations started, using Geth two years ago, the technology now features higher complexity and hardware requirements. Keeping up with all the developments in the ecosystem and understanding their implications is time-consuming. These changes lead to the second problem when doing research which is the technical complexity of Ethereum and other EVM blockchains. These technologies are complex systems involving many layers of technologies including consensus algorithms, smart contracts, and fee markets that all need to be understood to do valuable research. To understand these systems Ethereum offers a lot of documentation which sometimes acts as a two-edged sword. The information is usually available but finding it can be incredibly difficult. Initially, there was no knowledge of the “txlookuplimit” parameter, and the belief was held that an archive node was required to conduct the research. After reading a large collection of documentation, it became evident that operating a full node was sufficient. Due to many of the problems outlined a further problem is created. There are very few research projects that have been done on the Ethereum blockchain. It’s believed that this has influenced the amount of research being produced and has limited the number of relevant benchmarks and comparisons for the research project.

5. Results and Evaluation

5.1. Ethereum

5.1.1. Introduction

The results below show clearly that the transaction fee variance of the outlier removed data across the range of gas fees is lower after EIP-1559 which is shown below through charts, calculations, and tables of data. For brevity, the majority of graphs are from the 25 gwei target data as including all of the graphs for all the gas fees would have been repetitive. All other results can be found in the included tables of data or within the attached jupyter notebooks.

5.1.2. Histogram Analysis

Figure 6. Histograms for Legacy Blocks with 25 Gwei Target

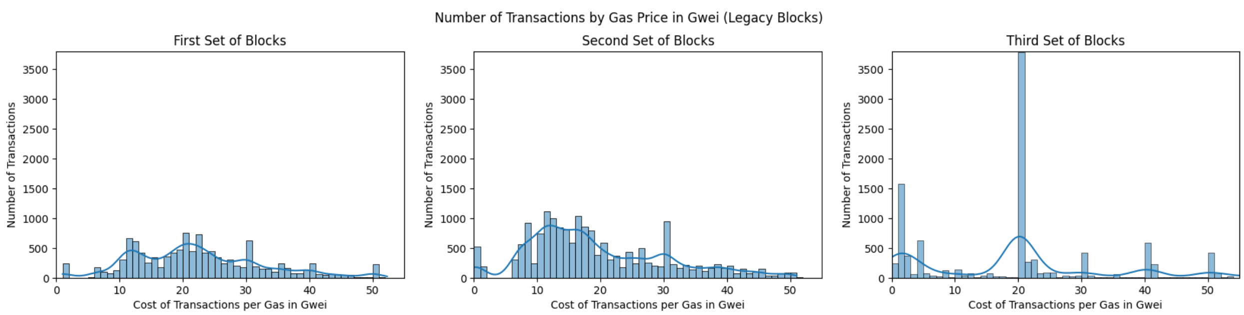


Figure 7. Histograms for New Blocks with 25 Gwei Target

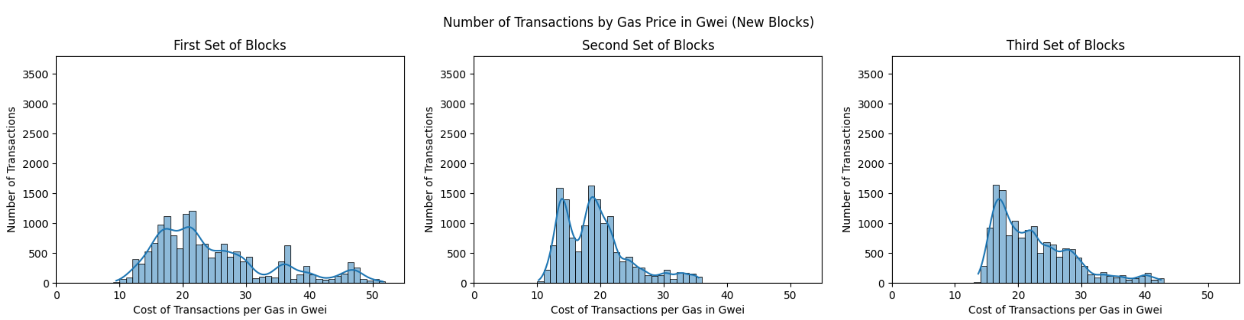
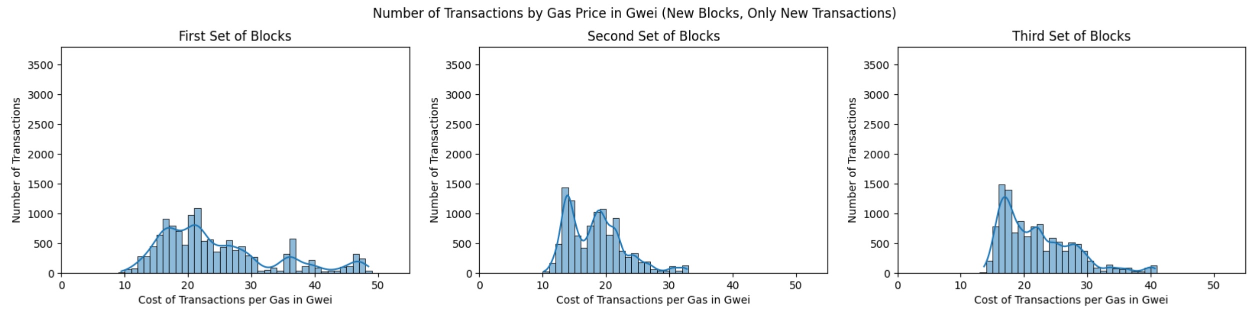


Figure 8. Histograms for Only New Blocks with 25 Gwei Target



The histogram charts above show the transaction cost with 1 gwei bins on the X-axis, and the number of transactions in each bin on the Y-axis, which shows the transaction fee variance within a day. Looking at the legacy histograms they show a strikingly high variance. The majority of transactions aren't within reasonable bounds of the target gas price of 25 or the average gas price. Alongside this, there are many distinct peaks that occur throughout the day showing that people are overspending or are finding the opportunity to underspend. Looking at the histograms for new blocks the difference couldn't be more striking with the vast majority of transactions occurring near the average gas fee paid for the day with a minor tail on the right side of the graph. Although these new blocks show that there is still variance, it appears to be far less than the variance in the legacy blocks. The histograms for only new blocks tell a similar story, but curiously within the middle of the tail, it appears to be slightly less tall.

5.1.3. Scatter Plot Analysis

Figure 9. Scatter Plot for Legacy Blockset 3 with 25 Gwei Target Average

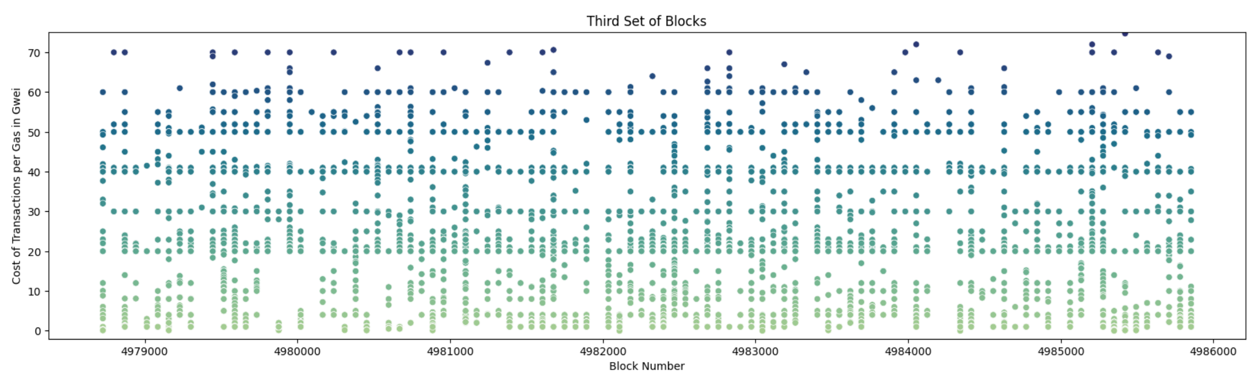


Figure 10. Scatter Plot for Legacy Blockset 3 with 25 Gwei Target Average

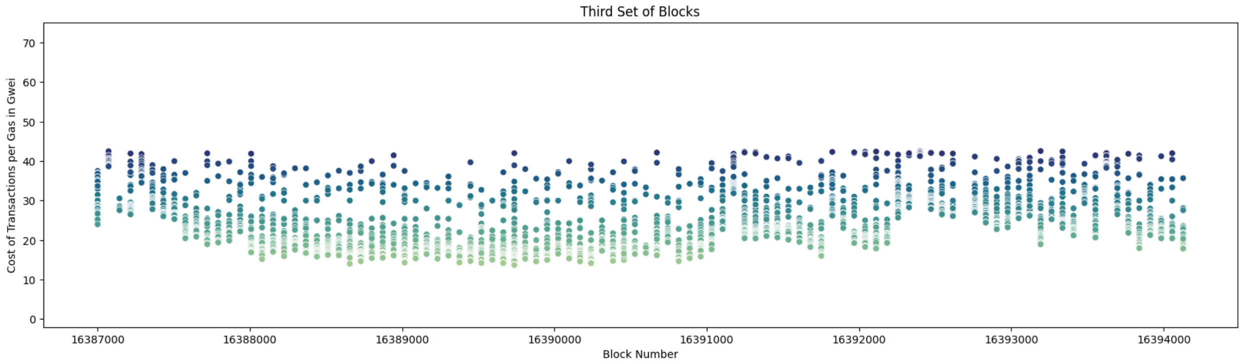
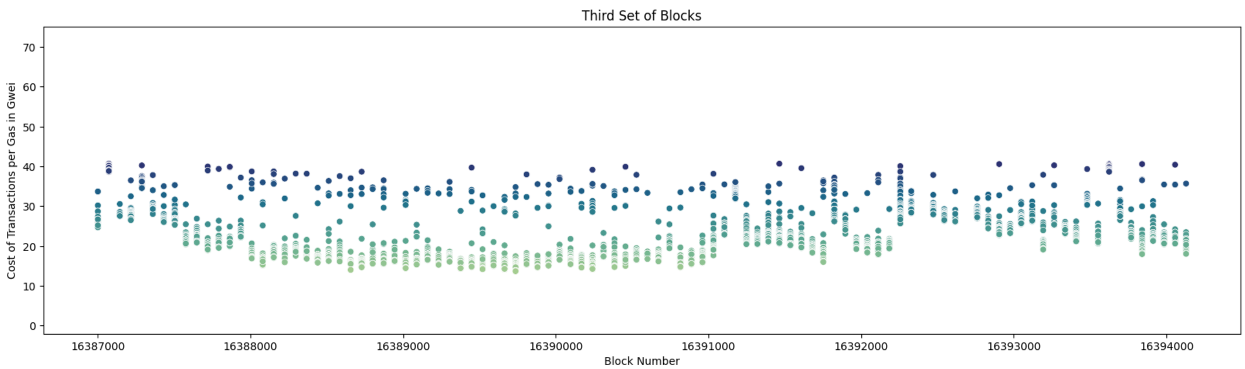


Figure 11. Scatter Plot for Only New Blockset 3 with 25 Gwei Target Average



The above scatter plots only show the third set from each of legacy, new, and only new, dataframes. I chose this set as it shows the greatest difference before and after alongside one of the possible causes. Within the legacy graph, it's clear that there are lines of transaction fees at integers divisible by ten. Previously when people picked their transaction fee, it is clear that people were more likely to select whole numbers such as 20, 30, or 40 instead of 19, 33, or 42. It wasn't just the data from the 25 gwei target dataframe, it appeared across the range of gas prices in varying severities. Two other clear cases of this occurring are on the 50 gwei target and the 75 gwei target dataframes. This shows that across gas prices it was a recurring problem, although it did seem to become less prevalent as gas prices got higher. This could have to do with people becoming more frugal with their transactions as the average transaction cost got more expensive. It seems reasonable to assume that individuals wouldn't be willing to spend an extra 10 gwei on top of their 100 gwei transaction fee, opting to try and up it one by one or by fractions of a gwei.

Figure 12. Scatter Plot for Legacy Blockset 1 with 50 Gwei Target Average

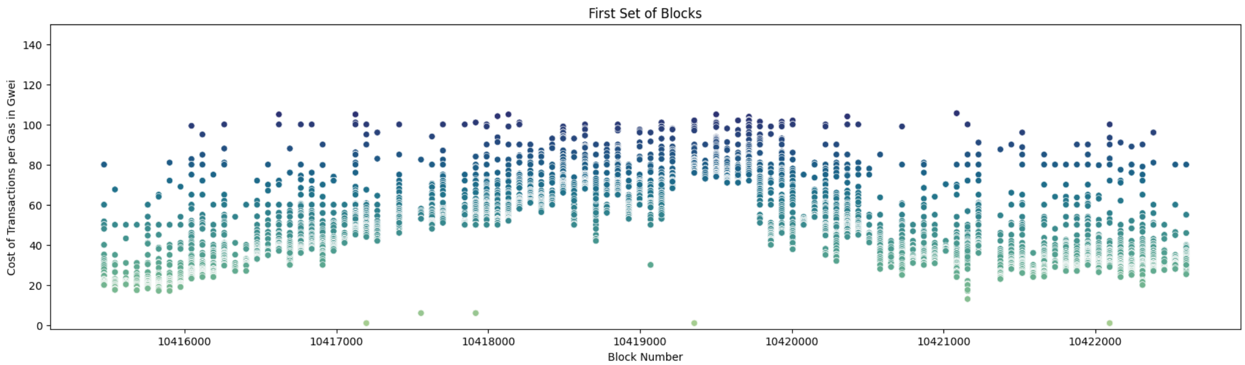
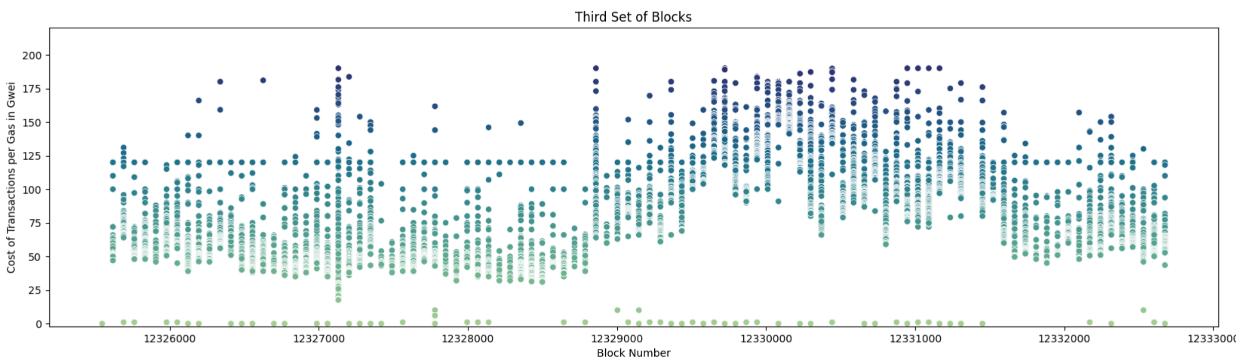


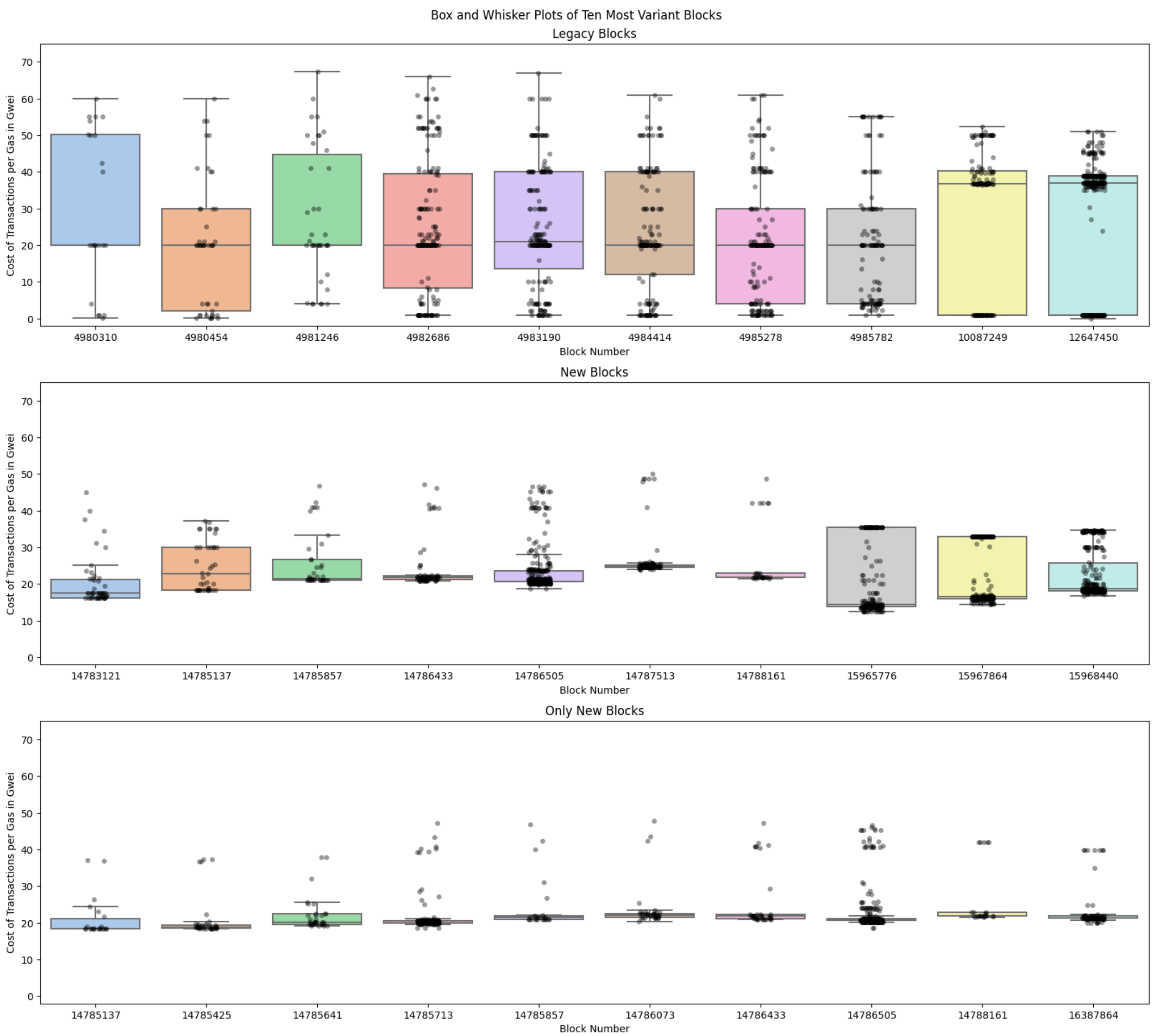
Figure 13. Scatter Plot for Legacy Blockset 3 with 75 Gwei Target Average



The problems outlined could themselves be a cause of transaction fee variance, and most importantly shows that people aren't very good at picking transaction fees. In the new and only new graphs, these lines are gone and the majority of transactions occur around the average transaction fee for the day. The only new data is very similar to the new data but has a distinct gap. I'm currently unsure what creates this gap, but my assumption would be that people creating legacy transactions are doing so as they find it easier to create faster transactions in the old model. Instead of selecting a priority fee, and a max fee, to go on top of their base fee, it may be easier for some to select a value that's slightly higher than the base fee and submit it as a legacy transaction. This is just an assumption and it has very little evidence to support the claim.

5.1.4. High Variance Box Plot Analysis

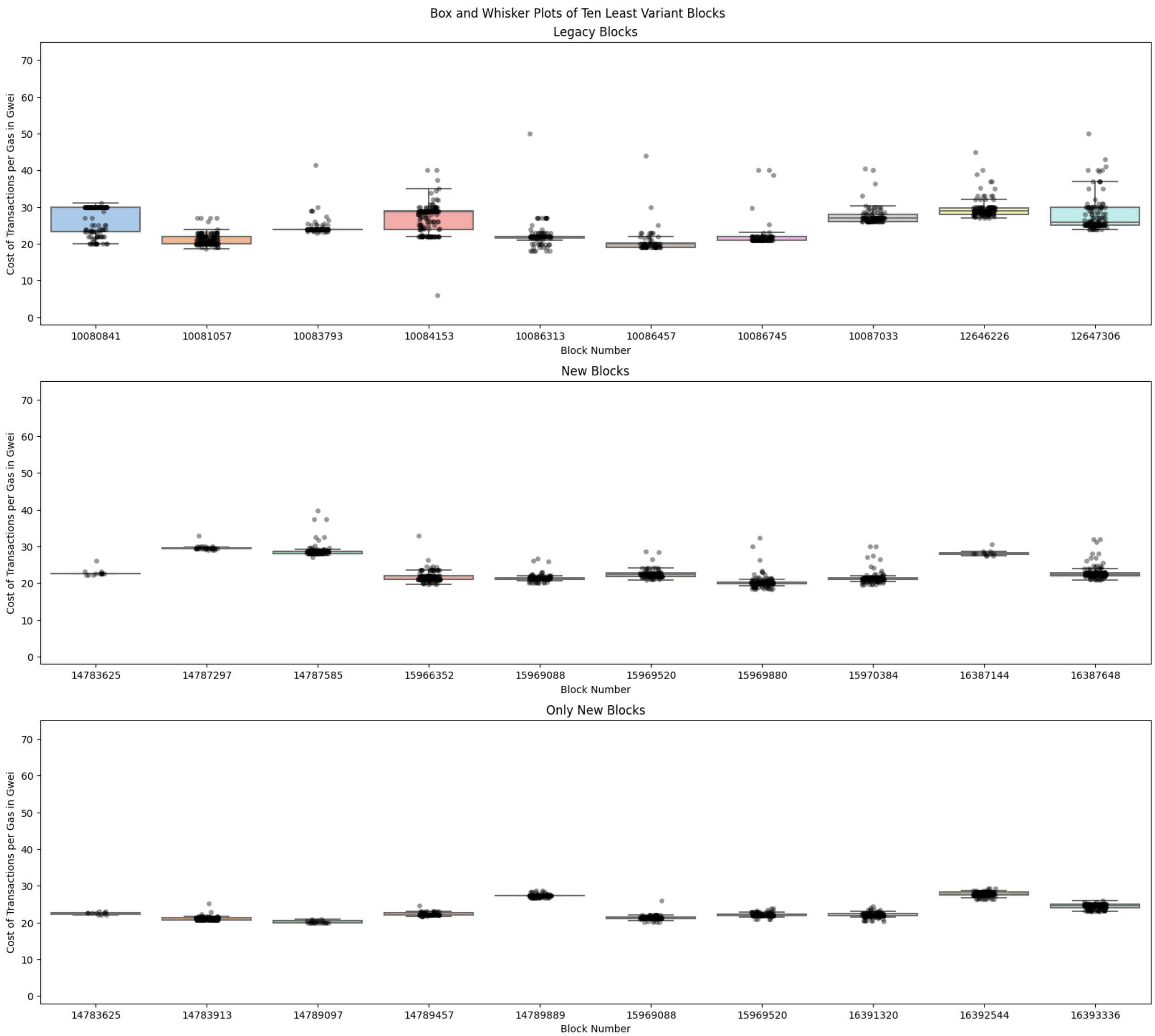
Figure 14. Highest Variance Blocks from Normalized Legacy, New, and Only New DataFra



The above boxplots show the worst-case scenarios of variance for each type of dataframe. The legacy blocks are far more variant than the other block types, but interestingly the only new blocks seem to have a much smaller variance than the new blocks. This would suggest that a higher number of only new transactions within blocks could lead to a smaller variance within what would be high variance blocks.

5.1.5. Low Variance Box Plot Analysis

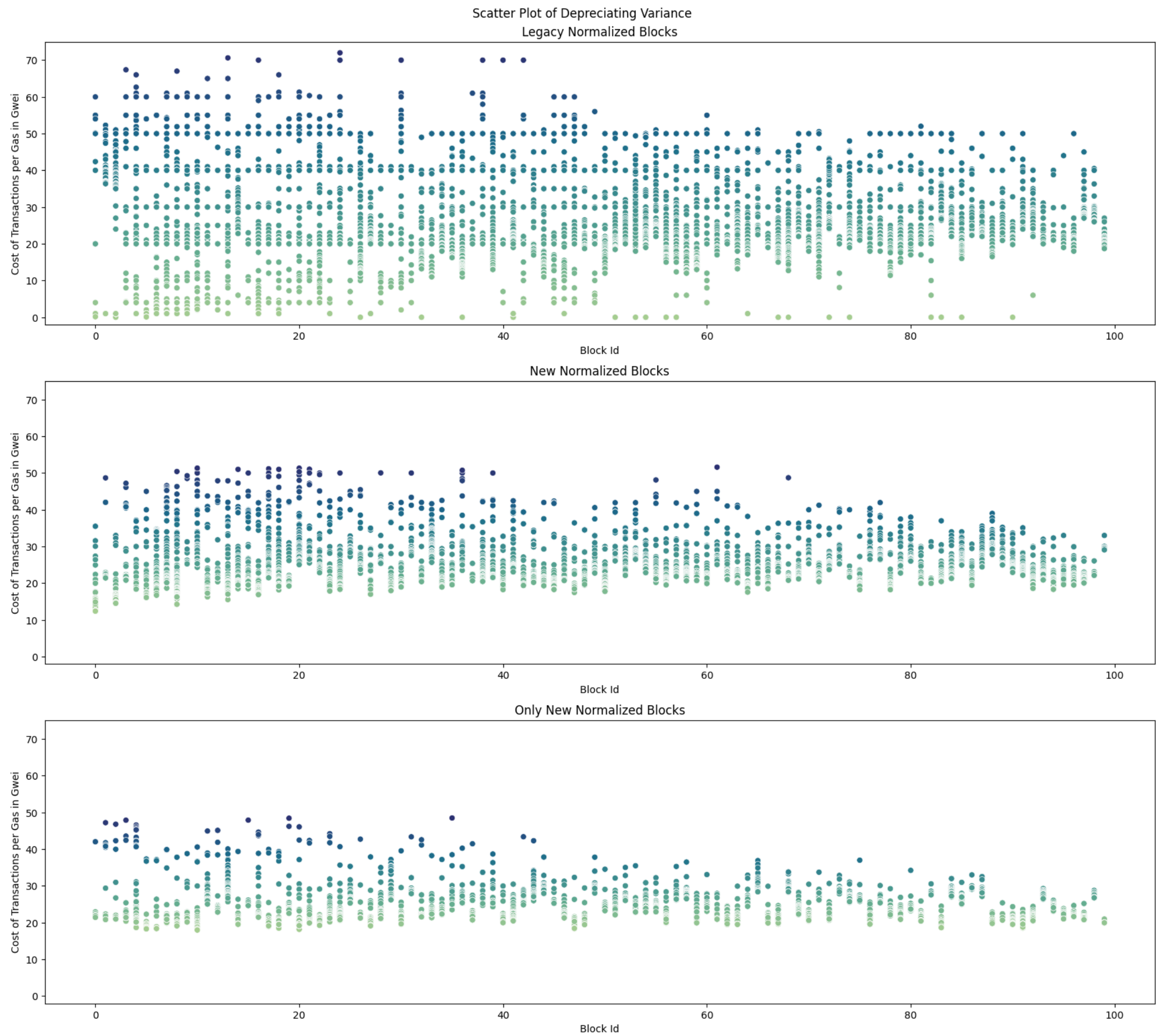
Figure 15. Lowest Variance Blocks from Normalized Legacy, New, and Only New DataFrames



The above boxplots show the best-case scenarios of variance for each type of dataframe. The new and only new boxplots show what is acceptable, the best-case scenario for variance should be very low variance. The legacy blocks show something quite striking, even in the best case the legacy blocks struggle to reduce variance down to an unnoticeable amount.

5.1.6. Variance Scatter Plot

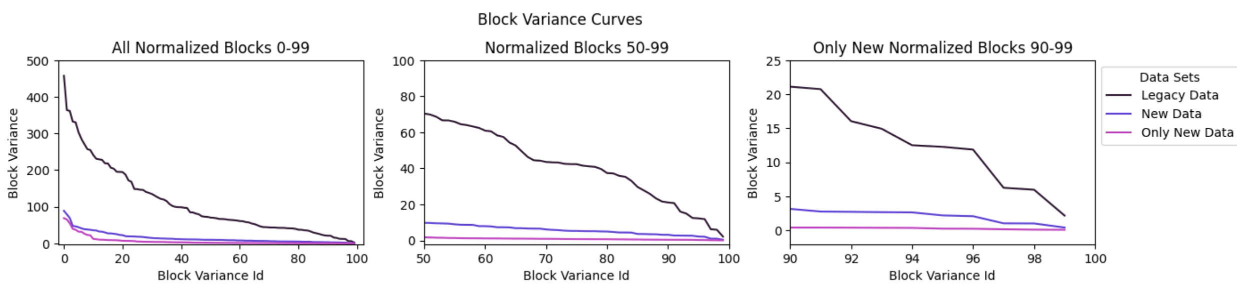
Figure 16. Scatter Plots from Normalized Legacy, New, and Only New DataFrames



The three scatter plots above show all the blocks and their transactions in the normalized dataframe ordered by variance. This shows the full breadth of the difference in transaction fee variance with blocks that have an average gas fee around the target price from all three days of each type. The block id showing the relative position of the block variance is the X-axis, and the cost of transaction per gas in gwei is the Y-axis. The top scatter plot showing legacy data, still reveals clear lines with people paying transaction fees that are integers divisible by 10. Even when the data is heavily manipulated this trend still continues showing clearly that this is a concurrent problem across many timeframes. At the bottom of the legacy scatter plot, you can see a line of transactions that occur at 0. Without a base fee forcing users to pay a minimum transaction fee for the transaction itself to be valid, users that controlled a large number of validators could include their transaction for free. An inconvenient scenario for everyday users having to compete for block space with individuals who have a clear advantage. The combination of these two problems has led to a scatter plot showing that the majority of legacy blocks have a far greater variance than their new and only new counterparts. The legacy scatter plot almost seems to make steps down in variance, in contrast with the new and only new dataframes that show a smooth decrease. Which I believe is caused by users preferring the selection of round integers as their gas fee instead of more precise transaction fees.

5.1.7. Variance Curves

Figure 17. Line Plots from Normalized Legacy, New, and Only New 25 Gwei DataFrames



Above is the variance calculations for the blocks within the unique normalized dataframes overlaid on top of each other. The leftmost line plot shows all the blocks, followed by graphs focusing on lower variance blocks. On the X-axis is the relative block id ordered by variance counting from 0 to 99 and the Y-axis is the variance of the block's transactions. Instead of the variance being implied as it is in the other graphs, the above graphs show the variance value. As has been shown in every other graph, it's clear that throughout all the relative variances the legacy blocks have higher variances than the new blocks, and the new blocks have a higher variance than the only new blocks. This trend then continues when looking at the variance curves for the other target gas prices as well.

Figure 18. Line Plots from Normalized Legacy, New, and Only New 50 Gwei DataFrames

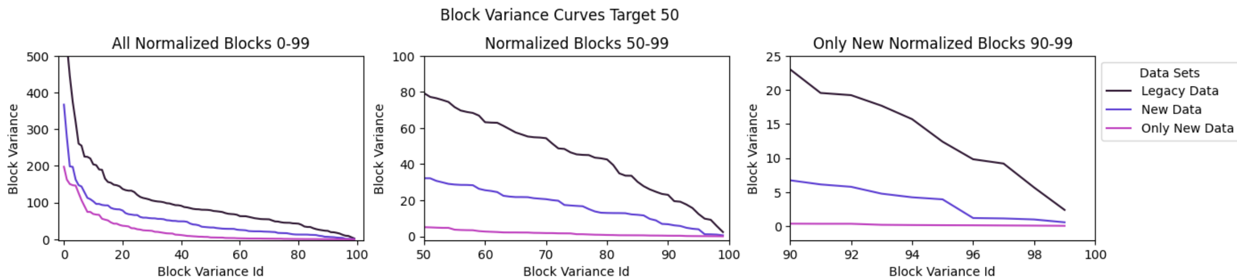


Figure 19. Line Plots from Normalized Legacy, New, and Only New 75 Gwei DataFrames

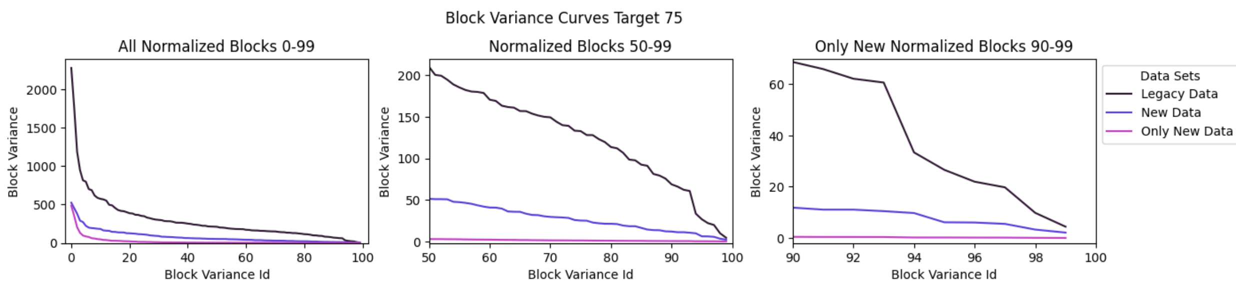


Figure 20. Line Plots from Normalized Legacy, New, and Only New 100 Gwei DataFrames

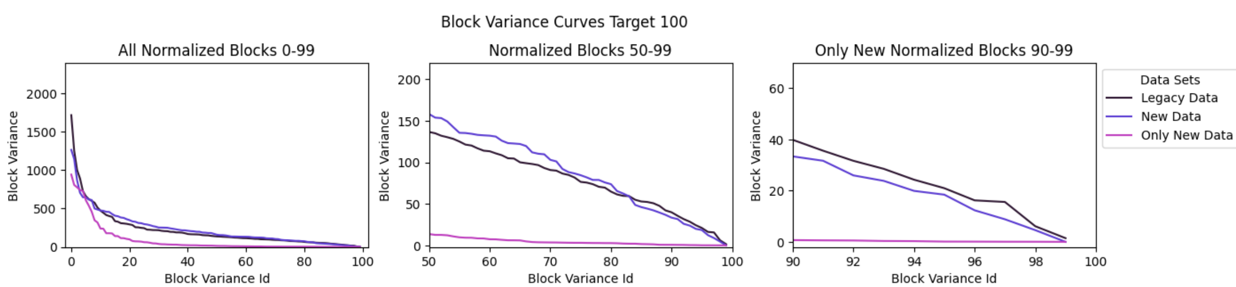
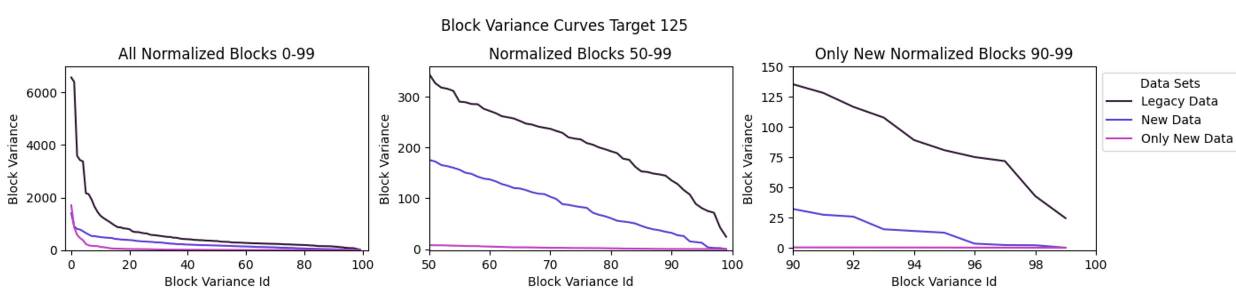


Figure 21. Line Plots from Normalized Legacy, New, and Only New 125 Gwei DataFrames



Going down the list of variance curves it's clear that this trend is clear, generally the relative variance of legacy blocks from all target gas prices is higher than that of new blocks, and the relative variance of new blocks is higher than that of only new blocks. It’s also clear that the nominal variance difference tends to shrink as the variance drops between legacy and new blocks, a relationship that doesn’t seem to be shared when comparing new and only new blocks. The one clear outlier to this trend is the target 100 dataframes, where the differences in block variance at many points in the graph are 0. Even the only new data seems to cross over with the legacy data when comparing high-variance blocks. Given the amount of evidence to the contrary, I believe this is just an outlier dataset that has uniquely low-variance legacy blocks, and uniquely high-variance new blocks.

5.1.8. Regression Models

The variance curves graph shows exactly why a single variance calculation isn’t sufficient to understand the difference between legacy and new transactions. Taking a day's worth of blocks before and after EIP-1559, and calculating the difference in variance, would lead researchers to assume that the difference in the variance of high-variance and low-variance blocks are the same. This couldn’t be further from the truth, to get a more accurate picture of the difference in fee variance, the regression models find the correlation based on the variance of the blocks. This compares relatively high variance blocks before and after EIP-1559, relatively average variance blocks, and relatively low variance blocks. From this, given a day's worth of new blocks and a given gas price, it’s possible to calculate a far more accurate theoretical variance for an individual block that happened prior to EIP-1559 and vice versa. This provides a more granular view of the difference in transaction fee variance and is therefore a far superior model than just comparing the difference in the average variance of blocks. To implement this I created two different models, a regression model and a curve model. The regression model takes the normalized unique variance dataframe for new or only new blocks on the X-axis and compares that with the normalized unique variance dataframe for the legacy or new blocks on the Y-axis. The data points are then plotted and a linear regression equation is then found. With this equation, a set of 100 normalized new or legacy blocks with a given average gas price and their corresponding variance can be taken and a theoretical corresponding set of new or legacy blocks at a specific gas price with their variance can be found.

The second model is an attempt to find the difference between the relative variance of the blocks, in other words, it’s an attempt at trying to accurately describe how the above variance curves interact with one another. The X-axis for this model is the block id of the corresponding differences in the variances of the data featured in the Y-axis. For example, in the first graph featured below, the Y-axis is the difference in the variance of the legacy values and the new values at each given block id. This block id is then flipped so that it is better understood by the reader with lower differences on the left growing to feature higher differences on the right. This was an aesthetic decision and has no real impact on the underlying equation of the curve. When creating the curve difference models I first attempted to make it a simple linear regression but found this misrepresented the data. With some light experimentation, I found that a cubic equation best represents the data and is used for all the curve models.

For each target gas price, there are three regression models the first has the new variance on the X-axis and the legacy variance on the Y-axis, the second has the only new variance on the X-axis and the legacy variance on the Y-axis, and the third has the only new variance on the X-axis and the new variance on the Y-axis. For each target gas price, there are also three curve models, all three featuring the block id on the X-axis, with the first featuring the difference between legacy and new variances, the second featuring the difference between legacy and only new variances, and the third featuring the difference between new and only new variances. Below all six graphs are featured.

Figure 22. Legacy and New 25 Gwei Normalized Variance Regression and Curve Models

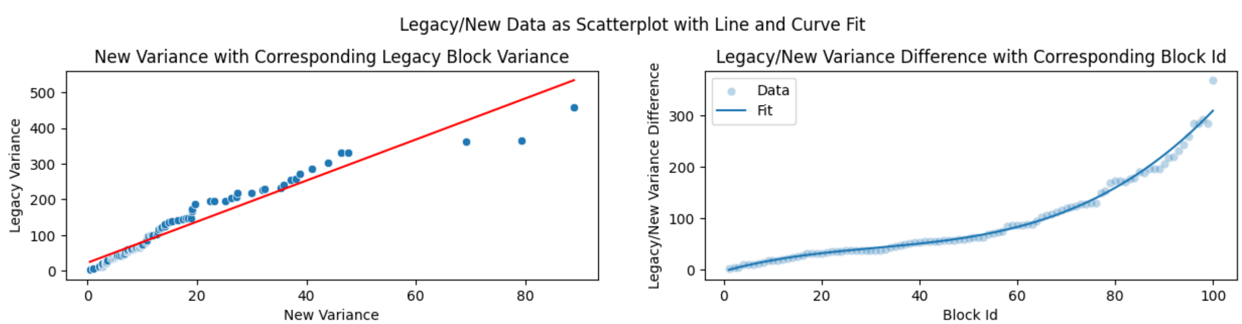


Figure 23. Legacy and Only New 25 Gwei Normalized Variance Regression and Curve Models

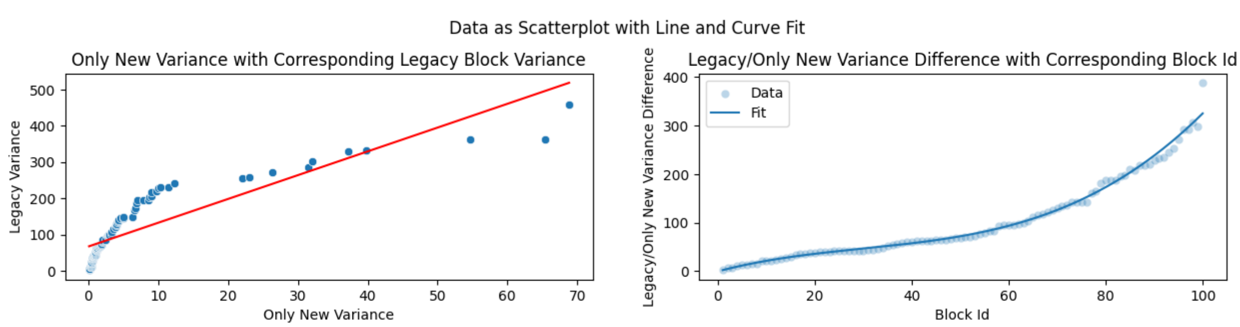
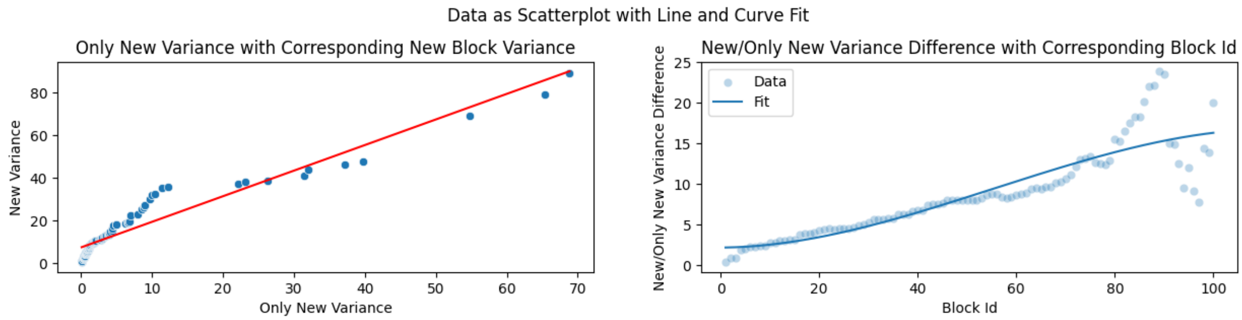


Figure 24. New and Only New 25 Gwei Normalized Variance Regression and Curve Models



Featured above are the graphs that are generated from the different models. The linear regression models show that comparison blocks are far more variant as the variance grows. It's also worth noting that the majority of data points are skewed heavily to the left of the graph with a minority to the right. This may lead to inaccuracies in comparing exceptionally variant blocks. The curve models are largely more accurate with minimal differences between the data points and the fit. The one outlier is the differences between new and only new blocks where the difference changes very rapidly on the higher variances, this may be an outlier or it may be speaking to something deeper but within the scope of this research, it is hard to tell.

For the remaining target gas prices, the graphs featuring the relationship between new and legacy transactions are featured below.

Figure 25. Legacy and New 50 Gwei Normalized Variance Regression and Curve Models

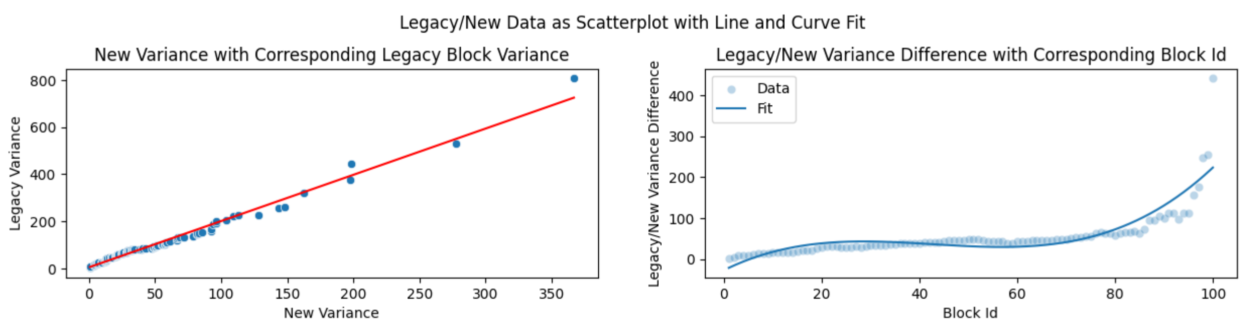


Figure 26. Legacy and New 75 Gwei Normalized Variance Regression and Curve Models

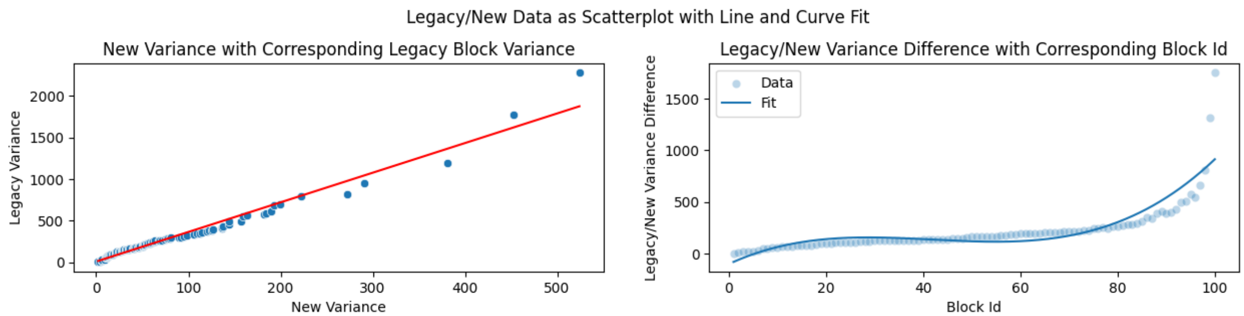


Figure 27. Legacy and New 100 Gwei Normalized Variance Regression and Curve Models

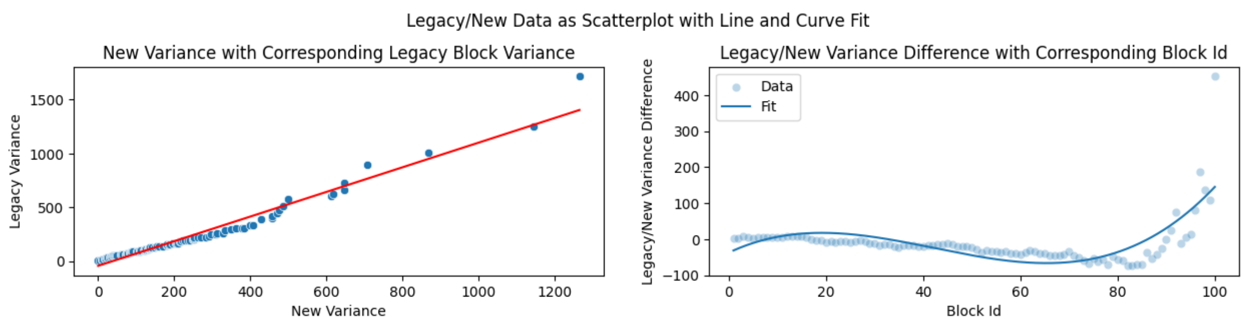
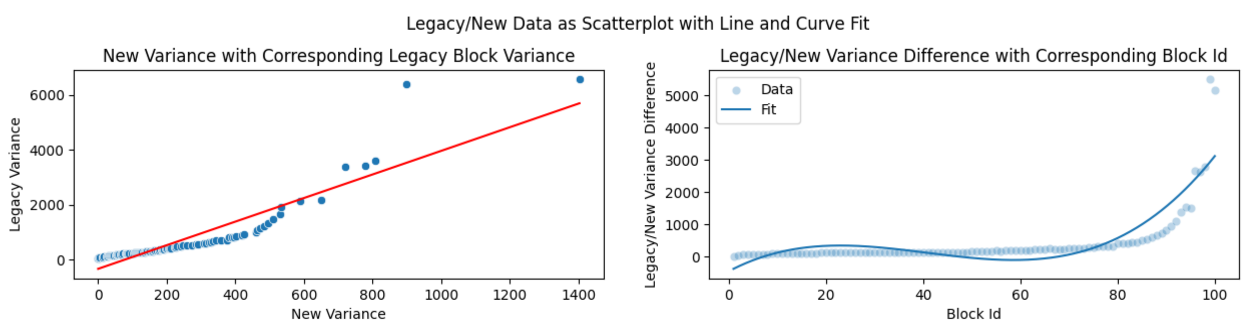


Figure 28. Legacy and New 125 Gwei Normalized Variance Regression and Curve Models



As the regression models above show, the pattern is very similar across gas prices, with the one outlier being the 100 gwei gas price target dataframe that features a difference curve model that dips into the negatives. Once more, it's believed that the figures produced by this dataframe are an outlier and represent a "worse case" variance difference scenario. Another difference to be noted by the 25 gwei gas price target dataframe is that the real values are further away from the regression line than the values of the other dataframes. There isn't any conclusion on this phenomenon, but I believe something deeper may be revealed by research that focuses on how gas prices impact transaction fee variance.

5.1.9. Model Equations

The equations for each of the lines featured in the above graphical visualizations are presented in this section. These equations are one of the products of the experiment that provide key insight into the variance differences after the implementation of EIP-1559. It's my belief that these equations provide the best answer to the research question, What is the difference in transaction fee variance before and after EIP-1559? At any point in time, a large set of transaction fee data can be gathered, the dataset can be normalized, ordered by variance, and then put into one of the created equations with the closest target gas fee. This would create a set of modelled legacy transactions with their corresponding variance, that people can use to compare modelled transaction fee variance before and after EIP-1559. A truly valuable tool to future researchers and developers interested in the subject matter.

Table 1. Linear Regression Equations from Normalized Dataframes by Target Gwei

Linear Regression Equations		
Model Type	Coefficient of X	Constant
25 Legacy / New	5.762	22.129
50 Legacy / New	1.964	4.820
75 Legacy / New	3.560	8.621
100 Legacy / New	1.145	-44.700
125 Legacy / New	4.298	-342.429

Table 2. Curve Model Equations from Normalized Dataframes by Target Gwei

Curve Model Equations				
Model Type	Coefficient of X^3	Coefficient of X^2	Coefficient of X	Constant
25 Legacy / New	0.001	-0.058	2.653	-2.765
50 Legacy / New	0.011	-1.429	73.933	-20.763
75 Legacy / New	0.005	-0.582	21.917	-105.154
100 Legacy / New	0.002	-0.215	6.377	-37.554
125 Legacy / New	0.020	-2.421	79.926	-446.796

5.1.10. Normalized Variance Table Data

A far simpler method to quantify the difference in transaction fee variance before and after EIP-11559, is to find the mean and median variances of different parts of the normalized dataframes by each gwei target. By doing so, you find a much simpler value that gives a somewhat comprehensive understanding of the result of the research. These values offer a more simple value that can be communicated easily to those who wish to receive an elementary understanding of the implementation of the update. The tables themselves are broken up by the various gwei targets that feature mean and medians from three different parts of their respective dataframes. The variance mean and median shown in the leftmost columns is the variance calculations for the entire normalized dataframe. The high variance mean and median shown only uses data from the top ten most variant blocks featured in the dataframe. The low variance mean and median shown only uses data from the top ten least variant blocks.

Table 3. Legacy, New, and Only New 25 Gwei Normalized Data

Legacy Normalized Data 25 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	110.11	71.30	311.72	302.73	10.33	11.87
New Normalized Data 25 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	15.27	9.84	53.35	41.79	2.09	2.22
Only New Normalized Data 25 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	6.56	1.83	36.10	37.21	0.29	0.28

Table 4. Legacy, New, and Only New 50 Gwei Normalized Data

Legacy Normalized Data 50 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	106.73	79.73	428.34	386.94	8.29	9.21
New Normalized Data 50 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	51.90	32.56	190.07	146.46	3.48	4.72
Only New Normalized Data 50 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	24.52	5.80	119.85	146.31	0.22	0.16

Table 5. Legacy, New, and Only New 75 Gwei Normalized Data

Legacy Normalized Data 75 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	298.77	210.98	1210.24	948.66	22.87	21.98
New Normalized Data 75 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	81.49	51.57	325.73	343.12	5.38	6.03
Only New Normalized Data 75 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	22.10	2.86	153.97	96.75	0.12	0.14

Table 6. Legacy, New, and Only New 100 Gwei Normalized Data

Legacy Normalized Data 100 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	221.30	138.35	1002.48	991.25	12.20	15.62
New Normalized Data 100 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	232.32	163.90	810.01	667.16	11.28	8.85
Only New Normalized Data 100 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	92.55	14.53	716.22	713.96	0.19	0.14

Table 7. Legacy, New, and Only New 125 Gwei Normalized Data

Legacy Normalized Data 125 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	681.58	345.99	4020.66	3418.93	64.07	74.89
New Normalized Data 125 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	238.27	176.93	945.44	848.32	1.72	0.0
Only New Normalized Data 125 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	63.88	8.14	592.73	545.77	0.02	0.0

5.1.11. Conclusion

It would be hard to conclude after seeing the above results that EIP-1559 hasn't lowered transaction fee variance across the range of different gas fees. The only outlier found was the 100 gwei target in which the fee variance was very similar, because of the overwhelming evidence to the contrary, it's hard not to see the data as an outlier. Although it's clear that transaction fee variance has dropped, the validity of the values that have been acquired is debatable. The experiment conducted in the present study has little research to compare to, and therefore it's difficult to validate the results. It's likely that the nominal figures provided are more of an estimation rather than being exact in their nature.

5.2. Polygon

5.2.1. Introduction

Alongside Ethereum, I also chose to analyze the Polygon blockchain due to the technical similarities of the two blockchains and the fact that Polygon had a remarkably similar update to their transaction fee market. Getting data for the Polygon blockchain was difficult so the amount of data that could be gathered is much lower. Instead of having multiple gas targets, I chose to only have one which ended up being 82.5. This number was picked because, out of all the data I gathered, this gas price was the only figure I could get six days' worth of data for, three being legacy and three being new. For brevity, many of the graphical visualizations have been excluded from this part of the analysis. All of which are accessible in the attached jupyter notebooks, and many of which will be explained without the corresponding visualization.

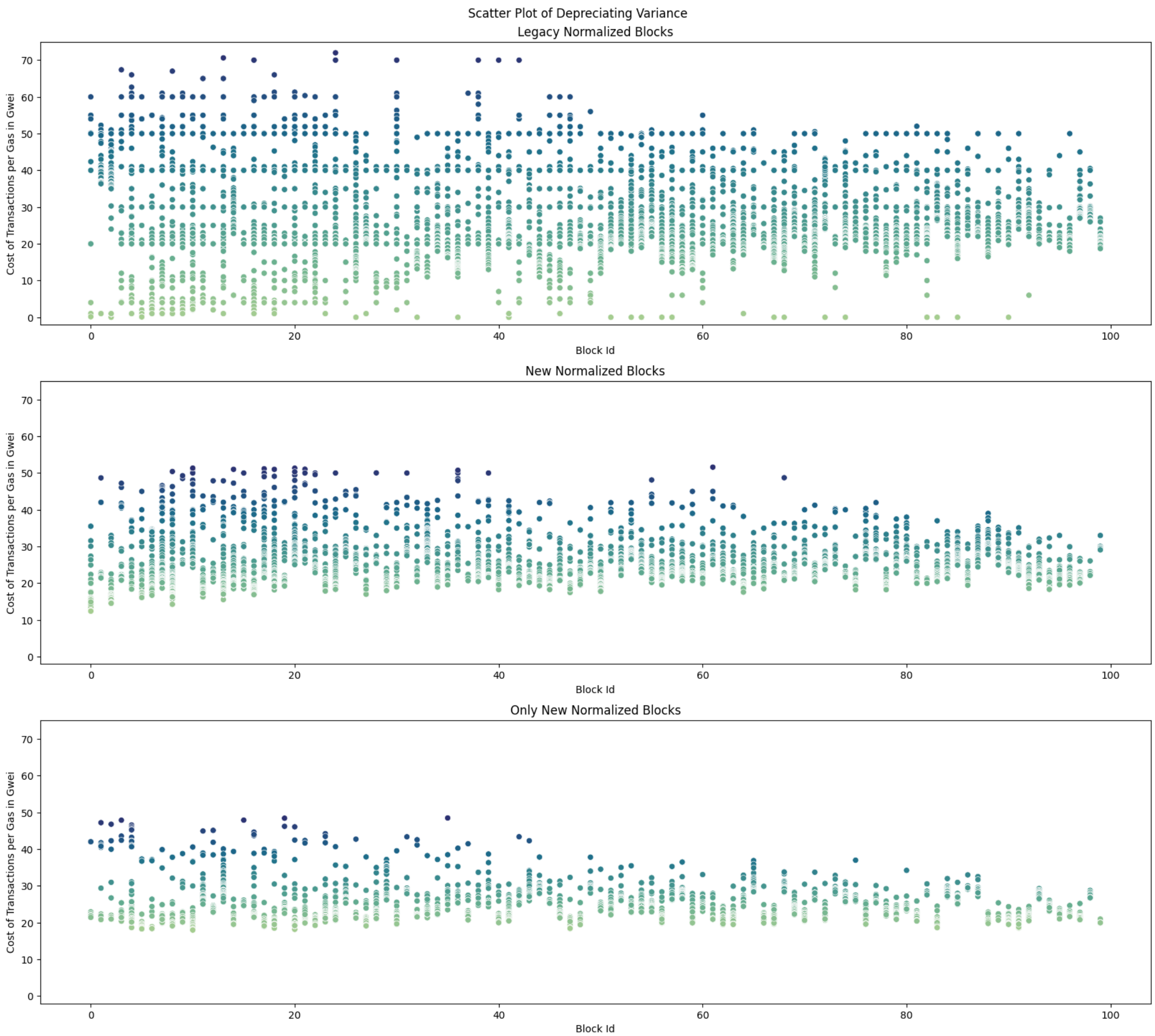
5.2.2. Polygon Analysis

When analyzing the results returned from the Polygon histogram charts the similarities shared between the Ethereum histograms are striking. The difference between the legacy and new blocks for Polygon shows a much smaller number of distinct peaks with new data being heavily skewed towards the target gwei while legacy data is spread across the X axis. When analyzing the scatter plots a similar trend is also seen. The Polygon legacy data shows a much wider range of non-outlier transaction fees, while the Polygon new data shows a much slimmer range of non-outlier transaction fees. An interesting difference between the Ethereum data and the Polygon data is that the base fee changes very little over the course of the day. This may be due to how much more block space is available for the Polygon blockchain, making it much harder for small spikes in demand to drive gas fees high enough to notably change the base fee.

When analyzing the results returned from the Polygon box plot charts the similarities shared between the acquired Polygon and Ethereum data continue. With the highest variance legacy data being significantly more variant than the highest variance new data. Interestingly, when looking at the overlapping scatter plot the lowest variance legacy data is also less variant than the lowest variance new data, but the box plots for the legacy blocks seem much smaller. After double-checking the code that creates the graph ensuring that I was using the correct dataframes I can only conclude that this is most likely due to a very large collection of data points at a specific point interfering with the visualization.

The next set of graphs showing the normalized Polygon dataframes ordered by variance show more similarities when compared to the Ethereum data. Not only does it show legacy data being more variant, and new data being less variant, it also shows a number of unique characteristics that were also seen on the Ethereum data. The first being the long lines of transactions that occur across multiple blocks with transaction fees divisible by 10, which completely disappears when looking at the new data. There are also a large number of transactions with no transaction fee, another characteristic that completely disappears when analyzing the new data. Below the graphs are displayed:

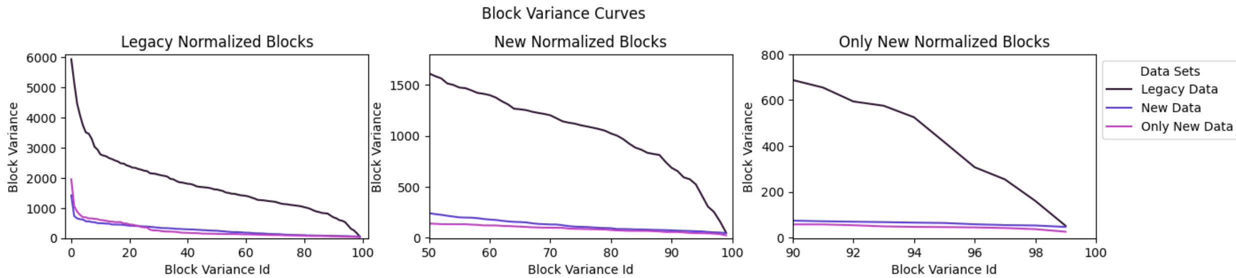
Figure 29. Scatter Plots from Polygon Normalized Legacy, New, and Only New DataFrames



5.2.7. Variance Curves

The Polygon variance curves featured below show a similar trend to the Ethereum variance curves. Generally, the variance of legacy blocks from the given target gas prices is higher than that of new blocks, and the variance of new blocks is higher than that of only new blocks. The one outlier to this rule is the only new data on the far left of the graphs where there is a large amount of overlap with the new variance. Despite this, given the similarities between the new and legacy data when comparing the information gathered from Ethereum and Polygon, it’s hard to believe that the similarities are a coincidence.

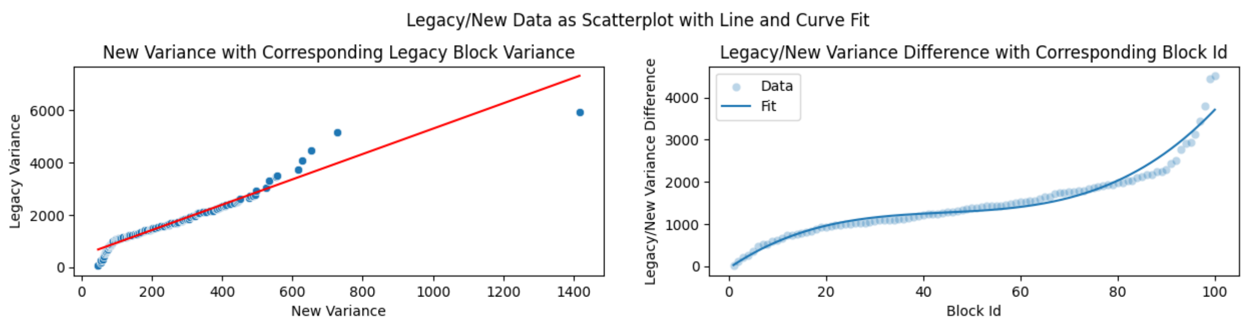
Figure 30. Line Plots from Polygon Normalized Legacy, New, and Only New DataFrames



5.2.8. Polygon Regression Models

For brevity only the regression models generated from legacy and new data are shown, with the remainder found in the attached jupyter notebooks. The legacy and new Polygon variance models show remarkably similar results to the Ethereum data. The raw data also has a very close fit to both the regression and curve models displaying their respective accuracy. In contrast, the new and only new Polygon variance models, shown in the table data and jupyter notebook, show higher variances for the only new data, which didn't occur in the Ethereum data.

Figure 31. Legacy and New Polygon Normalized Variance Regression and Curve Models



5.2.9. Model Equations

Table 8. Linear Regression Equations from Polygon Normalized Dataframes by Relationship

Linear Regression Equations		
Model Type	Coefficient of X	Constant
Legacy / New	4.84	446.86
Legacy / Only New	3.39	894.22
New / Only New	0.70	92.99

Table 9. Curve Model Equations from Polygon Normalized Dataframes by Relationship

Curve Model Equations				
Model Type	Coefficient of X^3	Coefficient of X^2	Coefficient of X	Constant
Legacy / New	0.013	-1.649	78.550	-54.804
Legacy / Only New	0.011	-1.429	73.933	-20.763
New / Only New	-0.002	0.220	-4.617	34.040

5.2.10. Normalized Variance Table Data

Table 10. Legacy, New, and Only New Polygon Normalized Data

Legacy Normalized Data 82.5 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	1768.42	1614.01	4710.07	4462.75	183.83	164.59
New Normalized Data 82.5 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	272.88	244.95	707.68	653.62	54.54	54.92
Only New 82.5 Target Gwei						
DataFrame	Variance Mean	Variance Median	High Variance Mean	High Variance Median	Low Variance Mean	Low Variance Median
Normalized	257.90	142.05	804.39	694.02	45.88	46.26

5.2.11. Polygon Data Conclusion

It would now be hard to conclude after seeing the results from the Polygon blockchain that EIP-1559 has not lowered transaction fee variance on a separate chain. There was only one gas target used but the difference between legacy transaction fee variance and new transaction fee variance is notable. By every test that was run, outlier-removed transactions within blocks routinely show lower variance compared to before EIP-1559. This not only shows that EIP-1559 had a positive impact on the Polygon blockchain, but combining this evidence with what was seen on Ethereum, it is now clear that the changes implemented and their effects are reproducible.

6. Conclusion

This study aimed to answer two research questions; What is the difference in transaction fee variance before and after EIP-1559? And, is this difference reproducible on another blockchain? This study deemed these questions important due to the impact that transaction fee variance has on the user experience for those interacting with blockchain technologies. To answer these questions the study accomplished a set of objectives including; building a data pipeline that acquires data from a locally running Ethereum node, building a set of tools that parse, process, and visualize the acquired data, calculating a number of measures that discern a nominal value that quantifies the transaction fee variance before and after EIP-1559, and finally repeating all the above objectives for a similar blockchain. By accomplishing these objectives a system was built to analyze transaction fee variance that is made up of three parts, data procurement, data processing, and data visualization. Although this analysis appears beyond comprehensive, there were other ideas I had that couldn't be fully realized due to time and resource constraints.

The first and most obvious idea I planned to research was previously referenced. I believe there would be great value in researching the link between transaction fee variance and gas fees, and that it serves as an important piece of the puzzle that needs to be understood by further research. I also planned to build a transaction fee variance interactive dashboard, where anyone who was interested in the subject could alter various parameters and see the impact on transaction fee variance in a simulated block. I believe that spreading ideas, and effectively communicating research, can be more important than the research itself, and creating this tool would aid in the efforts to communicate how transaction fee variance impacts the different stakeholders in the Ethereum ecosystem.

The results from the experiments have proven beyond reasonable doubt that EIP-1559 has reduced transaction fee variance for all non-outlier transactions. Furthermore, it is clear that this difference is also reproducible on a separate blockchain with similar characteristics. From the research, values were produced stating the difference in transaction fee variance for each target gas price, but the validity of these values is questionable. I believe they provide an estimation but are by no means exact in their nature. Furthermore, I'm unaware of a method that would enable an exact calculation of the difference in transaction fee variance and believe this would require greater research into the subject matter.

7. Reflection on Learning

Apart from the knowledge gathered from the research, this project has expanded my mind in many ways. What I had done exceptionally well was set reasonable goals that I could ensure I met. I'm fully aware that lofty goals and ambitions given a timeframe are a recipe for disaster. The focus should rather be on doing a small set of things well and going deep instead of wide. This was further cemented and is a belief I hold strongly that I apply to a number of different aspects of my life. I personally believe I could have done better in conducting the research, instead of jumping right into trying to solve the problem myself, I should have spent much more time researching material on the subject and trying to fully understand all that was already done. Given the opportunity to do a similar project at a later date, I would have spent far more time researching different parts of what I intended to do before I ever wrote a line of code. Not doing so led me to take a scattered approach which included reading research papers and outside materials throughout the project. This most likely led me to spend significantly more time on this project than I would have if I had done all the research in the beginning. When I go into my master's degree next year I will be leveraging this knowledge to hopefully be more effective in producing high-quality research. The last thing I regret not doing is planning my experiment and the process to accomplish the experiment far more effectively. This experiment morphed a lot in the process of doing the analysis because of problems I ran into. Planning better was also impacted by having not researched properly prior to the experiment, and could have been pivotal in producing a greater quantity of high-quality results. Finally, I have discovered a love for research and using data to come to accurate conclusions. I really loved this process and would love to continue to do research like this in the future.

Glossary

Wei: The smallest denomination of Ether representing 10^{18} of one Ether.

Gwei: A denomination of Ether representing 10^{9} of one Ether usually used as the denomination to represent gas fees.

Ether: The native token for Ethereum that facilitates operations for the blockchain.

Blockchain: A shared ledger that maintains a continuously growing list of records.

Blocks: A batch of data to be added to the blockchain including a hash linking it to the previous block.

Transaction: A change made to the blockchain by an externally owned account.

Block Limits: A limit set on the number of transactions that can be included in a block to ensure consensus is met prior to the creation of the next block.

Validators: A computer that meets a set of requirements, that participates in consensus on a blockchain.

Transaction Fee Market: The market that sets the price for the cost of a transaction to be included in a block.

Nodes: A computer running software that is able to connect to the Ethereum network.

EIP-1559: Ethereum Improvement Protocol (EIP) 1559, also referred to as the London Hard Fork, responsible for a number of changes to the transaction fee market.

Base Fee: A minimum fee that is necessary to include with all transactions after EIP-1559.

Transaction Fee Variance: A statistical measure used to find how far away each transaction fee is from one another.

Bibliography

Mihal. D. 2023. Money Movers Which blockchain is settling more value? Available at: https://money-movers.info/ [Accessed: 20 March 2023].

Reynolds, S. 2021. EIP-1559 is live thanks to Ethereum's London hard fork. Available at: https://blockworks.co/news/eip-1559-is-live-thanks-to-ethereums-london-hard-fork/ [Accessed: 20 March 2023].

corwintines et al. 2023. Blocks. Available at: https://ethereum.org/en/developers/docs/blocks/ [Accessed: 20 March 2023].

Dexter, S. 2018. Nothing at Stake Problem - A Forkin' Mess! Available at: https://www.mangoresearch.co/casper-nothing-at-stake-problem/ [Accessed: 21 March 2023].

Edington, B. 2023. Upgrading Ethereum, A technical handbook on Ethereum's move to proof of stake and beyond. Available at: https://eth2book.info/capella/part2/incentives/slashing/ [Accessed: 6 May 2023].

LucaPenella et al. 2023. Proof-of-Stake (POS). Available at: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/ [Accessed: 23 March 2023].

Nakamoto, S. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System.

wackerow et al. 2022. Proof-of-Work (POW). Available at: https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/ [Accessed 23 March 2023].

wenceslas-sanchez et al. 2023. Gas and Fees. Available at: https://ethereum.org/en/developers/docs/gas/ [Accessed 24 March 2023].

Orenes-Lerma, L. 2023. How Does a Blockchain Transaction Work?. Available at: https://www.ledger.com/academy/how-does-a-blockchain-transaction-work [Accessed: 1 May 2023].

Hafid, A. 2020. Scaling Blockchains: A Comprehensive Survey. IEEE Access. doi: 10.1109/ACCESS.2020.3007251.

Irwin, K. 2022. Solana Co-Founder Says 'Long-Term Fix' to Network Outages Is in the Works. Available at: https://decrypt.co/111486/solana-co-founder-long-term-fix-network-outages [Accessed: 1 May 2023].

Solana. 2023. Validator Requirements. Available at: https://docs.solana.com/running-validator/validator-reqs [Accessed: 1 May 2023].

go-ethereum, 2022. Hardware requirements. Available at: https://geth.ethereum.org/docs/getting-started/hardware-requirements [Accessed: 1 May 2023].

Beck, J and Asher, M. 2021. What is EIP-1559? How Will It Change Ethereum? Available at: https://consensys.net/blog/quorum/what-is-eip-1559-how-will-it-change-ethereum/ [Accessed: 20 April 2023].

corwintines. 2023. Intro To Ether. Available at: https://ethereum.org/en/developers/docs/intro-to-ether/ [Accessed: 1 May 2023].

Yulin, L et al. 2022. Empirical Analysis of EIP-1559: Transaction Fees, Waiting Times, and Consensus Security. doi: 10.48550/arXiv:2201.05574.

Reijsbergen, D et al. 2022. Transaction Fees on a Honeymoon: Ethereum's EIP-1559 One Month Later. doi: 10.48550/arXiv:2110.04753.

Etherscan. 2023. Ethereum Average Gas Price Chart. Available at: https://etherscan.io/chart/gasprice [Accessed: 25 April 2023].

Sardan, T. 2018. What is a light client and why you should care? Available at: https://www.parity.io/blog/what-is-a-light-client/ [Accessed 25 April 2023].

corwintines et al. 2023. Ethereum Archive Node. Available at: https://ethereum.org/en/developers/docs/nodes-and-clients/archive-nodes/ [Accessed 30 April 2023].

Awosika, E. 2022. Ethereum's Client Diversity Problem. Available at: https://hackernoon.com/ethereums-client-diversity-problem [Accessed 20 April 2023].

Leonardos, S et al. 2021. Dynamical Analysis of the EIP-1559 Ethereum Fee Market. doi: 10.48550/arXiv.2102.10567

Beiko T. 2021. London Mainnet Announcement. Available at: https://blog.ethereum.org/2021/07/15/london-mainnet-announcement [Accessed 20 March 2023].