Staff Working Paper/Document de travail du personnel—2022-2

Last updated: January 10, 2022

What Drives Bitcoin Fees? Using Segwit to Assess Bitcoin's Long-Run Sustainability

by Colin Brown,[1] Jonathan Chiu[2] and Thorsten V. Koeppl[3]

[1]Statistics Canada

[2]Banking and Payments Department
Bank of Canada
jchiu@bankofcanada.ca

[3]Queen's University

DOI: https://doi.org/10.34989/swp-2022-2 | **ISSN** 1701-9397

Acknowledgements

The views expressed in this paper are not necessarily the views of the Bank of Canada or Statistics Canada. This research was supported by SSHRC Insight Grant 435-2014-1416. The authors declare that they have no relevant or material financial interests that relate to the research described in this paper.

Abstract

Can Bitcoin remain tamper proof in the long run? We use block-level data from the Bitcoin blockchain to estimate the impact of congestion and the USD price on fee rates. The introduction and adoption of the Segwit protocol allows us to identify an aggregate demand curve for bitcoin transactions. We find that Segwit has reduced fee revenue by about 70%. Fee revenue could be maximized at a block size of about 0.6 MB when Segwit adoption remains at current levels. At this block size, maximum fee revenue would be equivalent to 1/8 of the current average block reward. Hence, large sustained price increases are required to keep mining rewards constant in the long run.

Topics: Digital currencies and fintech, Payment clearing and settlement systems
JEL codes: E42, G2

1 Introduction

Bitcoin is a decentralized payment system without a central institution that verifies and settles transactions. The blockchain underlying Bitcoin uses a Proof-of-Work (PoW) protocol where miners compete to include transactions into the blockchain, thereby keeping track of changes in ownership.[1]

This mining competition is crucial to make the blockchain tamper proof. Users can try to undo their transactions by double spending where they replace their own transactions with an alternative one that sends the payment back to themselves.[2] When high rewards are offered for updating blocks, mining activity will be high, giving little or no incentives to double spend, since the PoW protocol induces honest miners to make expensive investments to win the competition.

In Bitcoin, the rewards for the PoW are financed in two ways. First, there are rewards in new bitcoins for miners. And second, there are fees in bitcoins pledged by users to have their transaction included into a block. Importantly, the design of Bitcoin is such that in the long run only fees will be used to finance this competition as block rewards decline from 50 bitcoins initially to 0 bitcoins.

Hence, it is important to understand if only fees can generate sufficient rewards to ensure that the Bitcoin blockchain remains tamper proof. We use block-level data from 2016 to early 2021 to answer this question. Using current demand for Bitcoin transactions, we find that fees can at most generate about 1/8 of the current level of average block rewards in USD. This casts some doubt on the long-run viability of the Bitcoin blockchain.

To arrive at this conclusion, we develop a simple empirical model based on block-level data for how the demand for bitcoin influences fees associated with transactions. A key aspect of our analysis is to use two natural experiments – two rapid price increases since late 2017 and the introduction of Segwit (segregated witness) – to identify supply and demand effects.

We first regress fee rates – the average fee in USD paid per KB for a transaction in bitcoin – on the level of congestion, USD prices of bitcoin and a measure of price volatility. There is a positive relationship between fee rates and these variables.[3] This is not surprising. As congestion increases, there are two positive effects on fee rates. When blocks are full, only transactions with the highest fee rates are selected by miners. Moreover, as more users compete to make payments given a

[1]For an introduction to the Bitcoin protocol, see Böhme et al. (2015).

[2]For details and an economic analysis of this problem, see Chiu and Koeppl (2017).

[3]To account for endogeneity of congestion, we resort to using lagged congestion as an instrumental variable.

fixed supply of block space, they have an incentive to post higher fees to get their transactions settled. Similarly, if demand for bitcoin as a financial asset increases its price increases as well. As a consequence, fee rates are likely to increase, since they are transaction costs to settle a financial trade with a higher value. Finally, higher volatility increases the need to settle a transaction quickly, again leading to higher fees.

Next, we rely on two events to identify an aggregate demand curve for Bitcoin. First, there were periods of sharp increases in the price of Bitcoin – one from late 2017 to early 2018 and another from late 2020 to the end of our data set. Many people have interpreted these as speculative periods. Second, in mid-2017, the Segwit protocol was introduced. The adoption of Segwit over time by Bitcoin users increases block space and thus acts like a shift along a long-run aggregate demand curve once we control for the speculative period. Changes in congestion and in the USD price of Bitcoin can then be interpreted as demand shocks that influence fee rates.

This allows us to use our empirical model of fee rates to estimate the maximum fee revenue that can be generated by creating higher congestion. First, we find that the current block size is about 45% too large in order to maximize transaction fees. Second, Segwit helped to reduce capacity pressures, but it also lowered fees significantly by about 70%. In fact, given current demand for Bitcoin transactions, to maximize block rewards it would be optimal to not have the Segwit protocol in place at all. Third, fee rates increase with the USD price of Bitcoin. In order to maintain the level of block rewards, one would need sustained price increases in the neighborhood of 15-20% per year as block rewards diminish.

There is very little economics literature on fees in Bitcoin. The closest work to our own is Easley et al. (2019), who use a theoretical queuing model to analyze the incentives for Bitcoin users to pledge positive fees. They find that median wait times as a measure of congestion drive fee revenue. Huberman et al. (2017) were the first to theoretically model the relationship between congestion and fee rates. Both these models do not, however, look at the empirical relationship between fees and overall demand for Bitcoin. They also do not consider the implications of the demand for transactions for the optimal design of Bitcoin. In a related context, Chiu and Koeppl (2019) investigate how fees are set by users that compete to have asset trades settled on a blockchain based on a PoW protocol that is optimally designed in terms of block time and block size. Finally, Lehar and Parlour (2020) use fees to study collusion among miners. They argue that despite high demand, prior to 2020 blocks were not always full, confirming that miners intentionally reduce throughput to keep fees and, hence, mining rewards high.

Only a few papers in economics have tackled the issue of how a declining reward structure will influence mining in Bitcoin and, hence, its safety in the long run. Two exceptions are Auer (2019) and Chiu and Koeppl (2017). Both papers use the double spending problem to evaluate the safety and cost of Bitcoin for settling transactions. As pointed out earlier, mining rewards play a major role in their analyses. Auer (2019) is skeptical that rewards will be enough in the long run to avoid double spending. Interestingly, our estimate here puts the reward that could be generated by fees alone in the long run at around 1.6 BTC at current average prices outside speculative periods. This is close to the total amount of rewards that Chiu and Koeppl (2017) find to be optimal in their quantitative exercises for an efficient system. They also argue, however, that it is in general inefficient to rely on fees rather than on seignorage to raise rewards for a cryptocurrency.

2 Data Description

2.1 Overview

The time horizon for our analysis is January 1, 2016 to March 31, 2021.[4] We use block-level data to compute daily average values for transaction fees, bitcoin prices and block summary statistics that measure the throughput for bitcoin transactions.[5] We also use auxiliary data from Bitcoin's Mempool to complement this data set.[6] This gives us a data set of $N = 1,917$ observations. We report some summary statistics of our data set in the appendix.

There are two important features to note when looking at our data set. First, there are episodes of large movements in the bitcoin price. These occur in the second half of 2017 and in 2020. Below, we estimate three break points in the data series for bitcoin prices that roughly correspond to these episodes. We refer to these two time episodes as "speculative periods", when Bitcoin demand might not have reflected primarily payment activity. Second, the effective throughput of the Bitcoin blockchain changed when the Segwit protocol was introduced in block 481,824 on August 24, 2017. This is significant as this protocol allows for more transactions to be included in a block and, thus,

[4]In the appendix, we show that our results are not affected when also including data from 2014 and 2015. We exclude these extra years from the main analysis since Bitcoin gained broad popularity only around 2016.

[5]All block-level data have been obtained from the repositories of `blockchair.com`.

[6]All auxiliary data have been obtained from `blockchain.info`. Mempool data, however, are only available from August 2016 onwards.

can be used as a controlled experiment for how an increase in the potential throughput of the Bitcoin blockchain would influence fees.[7]

2.2 What is Segwit?

Traditionally, blocks included all the information that is relevant for Bitcoin transactions. This is information about the transaction itself and metadata that verify if a transaction is legitimate. With the Segwit protocol, transactions can be broadcasted, but with meta information such as signatures, public key and other information (witness data) being stripped and verified separately from the block. Even though the size of a block is still capped at 1MB, this protocol allows the number of transactions in a block to increase.

The upgrade was a so-called "soft fork", implying that not the entire network needs to follow it. Nodes in the blockchain are free to use Segwit or not. The key to understanding this fact is that transactions with stripped information can still be determined to be legitimate by all nodes – legacy or Segwit – given the blockchain. A legacy node can simply not access the Segwit data of the transaction. Similarly, miners can choose whether to mine Segwit blocks or not. Crucially, legacy miners cannot include Segwit transactions. While they can form a block containing both types of transactions, they cannot communicate the Segwit data. Consequently, all Segwit-capable nodes would reject the block. Notwithstanding this, non-Segwit miners can build on the same chain as Segwit miners, since the digest of the chain for mining a new block uses only block headers that do not depend on the type of transactions.

Both miners and users have an incentive to switch to a Segwit operating mode. Miners can still include non-Segwit transactions if they want to. Hence, there is no reason for a single miner not to follow the protocol. When using Segwit, transactions have a smaller size and, thus, are cheaper given fees offered per byte.[8]

[7] During our time period, Bitcoin block rewards also halved twice. However, these events are immaterial to our analysis.

[8] Adoption could be slow for several reasons. There could be collusion among some miners not to follow the protocol in order to increase fees by keeping congestion high. Not all users may adopt wallets to store coins and conduct transactions that are compatible with Segwit. Also, Segwit supports the so-called Lightning Network where Bitcoin payments can be processed in a similar way to bilateral netting, which may not appeal to all users.

2.3 Bitcoin Prices

We calculate prices on the daily level directly from the block data. To do so, we use the total daily average fees in USD per block and divide this by the total daily average fees in bitcoin per block. This gives us a consistent price estimate as reported in the block data. We also calculate a simple volatility, which is the absolute value of the first difference in prices.

We first fit our time series of prices to a linear and filtered trend using the Hodrick-Prescott (HP) filter. Using the method of Bai and Perron (2003), we then estimate breakpoints for this time series with the restriction that we limit the number of breakpoints to a maximum of three. Figure 1 shows the time series together with the breakpoints, which are given as `10-12-2017`, `05-23-2018` and `07-19-2020`. These points later define a dummy variable that is set to 1 for the period between the first two dates and after the third date. The dummy variable is then used to identify these windows as speculative periods.[9]

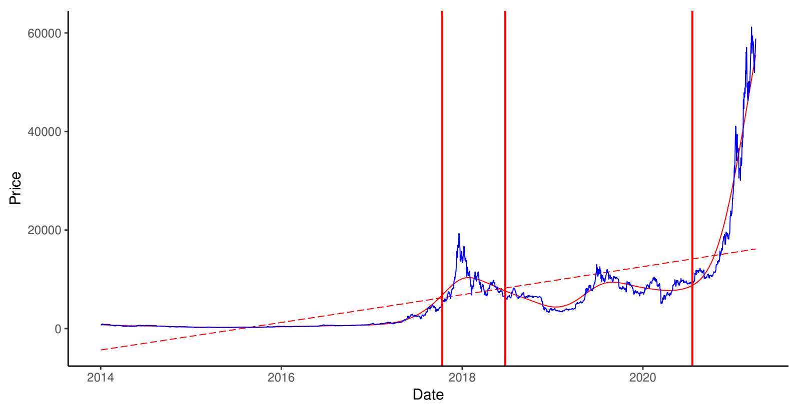


Figure 1: Bitcoin Price (in USD), Trend (linear & smoothed), Breakpoints

2.4 Congestion

2.4.1 Blockweight

The key variable in our analysis is *congestion*. Congestion refers to how much of the possible throughput is being taken up by transactions that are included in the blockchain. Hence, we

[9]The precise dates of these breakpoints crucially depend on the horizon of our analysis. We decided to use only data from after August 1, 2016 to determine these breakpoints so that they roughly line up with the large increases in bitcoin price relative to the trend.

measure congestion as the % of capacity used in a block. Before Segwit, capacity per block is given by the constraint

$$\sum_i x_i \leq 1MB \tag{1}$$

where x_i is the size of transaction i in bytes.

The Segwit protocol uses a different measurement of the block size called "weight". A weight unit is given by the 1/4,000,000 part of a block, and the block size is restricted to 4 mega weight units (MWU). The weight of a transaction is calculated as

$$3(x_i - \rho_i x_i) + x_i \tag{2}$$

where ρ_i refers to the witness data included in the transaction, which we have expressed as a fraction of the transaction size x_i. Note that this is equivalent to every byte of non-witness data of a transaction contributing 4 weight units, but witness data contributing only 1 weight unit per byte.

There are then two constraints on block size with the Segwit protocol. First, we have a restriction on the total weight of a block given by[10]

$$\sum_i 3(x_i - \rho x_i) + x_i \leq 4\text{MWU}. \tag{3}$$

Second, blocks with witness data stripped still need to satisfy the 1 MB limit imposed by Bitcoin, or

$$\sum_i (1 - \rho_i)x_i \leq 1 \text{ MB}. \tag{4}$$

This ensures that the blockchain stays consistent across legacy and non-legacy nodes.

First, note that a non-Segwit block by definition has $\rho_i = 0$ for all transactions. It follows immediately that both measures are identical. Second, for Segwit blocks by construction, the second constraint holds whenever the first constraint is satisfied. Block sizes above 1 MB reflect that it is a Segwit block and witness data have been removed to include more transactions. This implies that the first constraint is the relevant one, if considering the block level and not individual transactions.

[10]Theoretically, this implies that blocks could have a limit of 4 MB, if there were only witness data associated with transactions. Empirically, the maximum block size with Segwit, however, is around 2.3 MB.

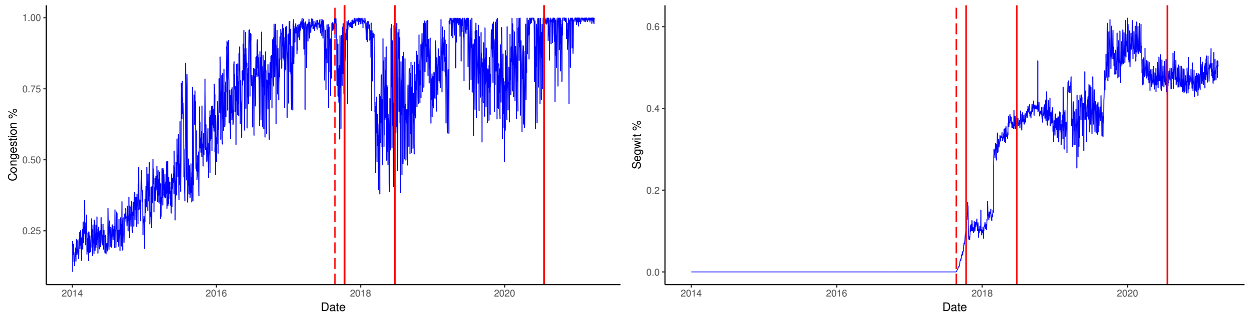


Figure 2: Congestion and Segwit Adoption

In summary, since we can scale pre-Segwit blocks by a factor of 4, we will use blockweight (as defined in (2)) as a percentage of 4MWU or

$$\text{CONGESTION} = \frac{\text{blockweight}}{\text{4MWU}} \tag{5}$$

as our congestion measure. To calculate a time series of congestion, we average blocks over each day in our sample.[11] We also use the percentage of Segwit transactions in a block – averaged for each day – as a measure of the adoption of the protocol. Figure 2 shows how our congestion measure changes over the time horizon and the degree to which Segwit has been adopted over time. The red solid vertical lines indicate our windows for the price series and the dashed vertical line delineates the introduction date of Segwit.

As Segwit transactions increase over time, there is less congestion until the end of our time series when congestion increases again. Interestingly, during the first speculative window, Segwit at first was adopted slowly until it increased sharply. More recently, adoption was fairly constant around 50%, even as congestion remained fairly high during the second speculative period.

2.4.2 Alternative Congestion Measures

We also look at some alternative congestion measures. First, we look at the daily stock of unconfirmed transactions in the Mempool for Bitcoin. The Mempool contains transactions that have been submitted to the network but have not yet been included in a block. Hence, a larger number of unconfirmed transactions means more congestion as the queue of transactions to get in the blockchain is longer.

[11] This also avoids the problem that blocks are not necessarily found in a specific time. Hence, blocks tend to be fuller if a block has not been found for a long time and emptier if blocks are found in rapid succession.

Second, an alternative measure can be constructed by relating the outflow of transactions from the Mempool to its size. Specifically, we calculate

$$\frac{\text{Avg. \# Transactions per Block}}{\text{\# of Unconfirmed Transactions}} \tag{6}$$

on a daily level. For consistency, we cap this measure at 1. There is no congestion if the outflow of transactions exceeds the size of the Mempool since all transactions get included immediately in a block. Importantly, an increase in this measure means less congestion.

Our final measure is the median confirmation time for a transaction on a daily level.[12] Table 1 shows that all these measures are correlated with each other with the right sign. By design, the congestion measure that relates the flow of transactions to the average number of unconfirmed transactions is negatively correlated with all other measures.

Table 1: Correlation

	Cong.	Mempool	Alt. Cong.	Med. Conf. Time
Cong.	1	0.480	−0.617	0.537
Mempool	0.480	1	−0.540	0.432
Alt. Cong.	−0.617	−0.540	1	−0.420
Med. Conf. Time	0.537	0.432	−0.420	1

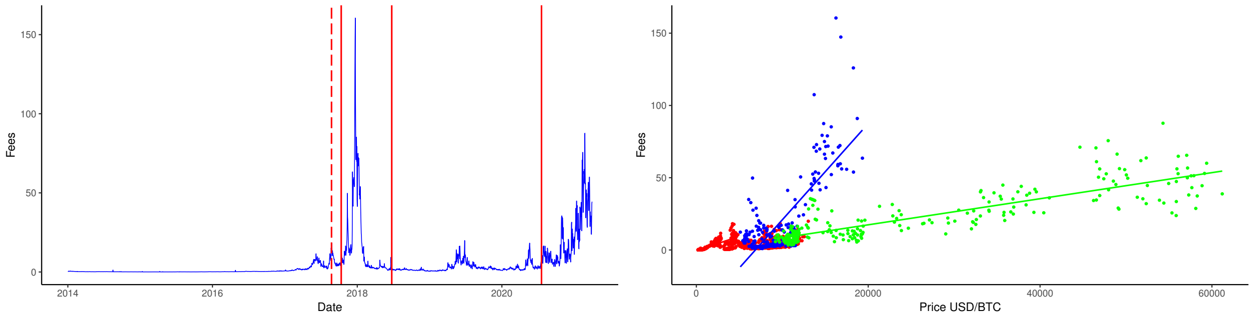
2.5 Fee Rates

To capture the cost of a Bitcoin transaction, we use fee rates that control for the size of the transaction. Also, we report these rates as USD denominated since we view all transactions being made for services in the numeraire of USD. Hence, what matters is the cost in USD and not in bitcoins.[13]

We again need to take Segwit into consideration so that we measure the cost of a transaction appropriately. After Segwit, what matters for a transaction is its size stripped of Segwit data.

[12] Note that data of these measures vary across sources as Mempools differ across miners, whereas block data are the same across all sources.

[13] This may not be true for certain transactions such as ones that consolidate bitcoin balances.



(a) Time Series

(b) Correlation w/ Congestion

Figure 3: Fee Rates (in USD)

Consider two Segwit transactions. Miners prefer to include the transaction with the smaller weight for the same transaction size and fee. Hence, for a user to get a transaction included in a block, what matters is the stripped transaction size. Consider a Segwit and a non-Segwit transaction. Miners again will prefer the smaller stripped size, for the same transaction size and fee. Hence, non-Segwit transactions of a small size can be more attractive than large Segwit transactions even though witness data cannot be stripped from the former ones. Importantly, for non-Segwit transactions, stripped transaction sizes and transaction sizes are identical. We thus conclude that not the overall size but the stripped size of an individual transaction matters in terms of adding congestion and, hence, for posting fees.

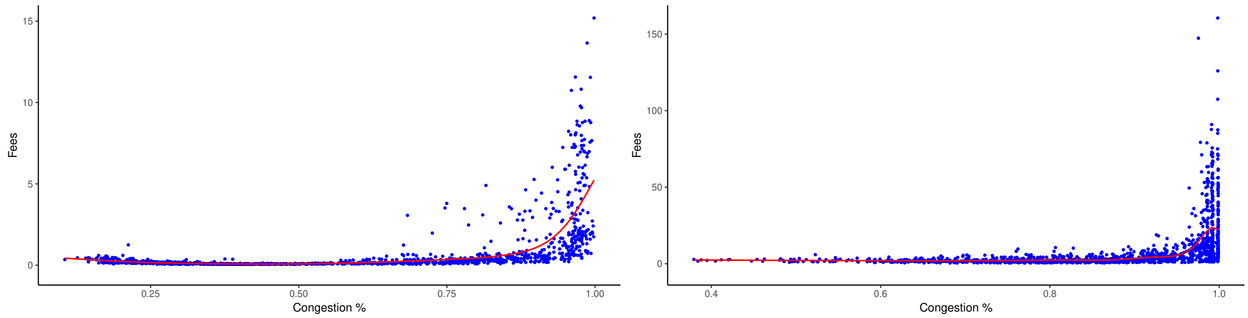
On the block level, we thus use the stripped block size corresponding to constraint (4) to calculate

$$\text{fee rate} \;=\; \frac{\text{Total fees per block in USD}}{\text{Stripped block size}} \tag{7}$$

as our measure for the costs of making a Bitcoin transaction. In our analysis, we again average the fee rate over blocks on the daily level.

Figure 3a plots the time series of fee rates. Interestingly, fee rates increase sharply during the speculative windows. We then plot in Figure 3b fee rates against bitcoin prices for different time periods. The red dots indicate data points outside the speculative window, whereas the blue ones refer to the first speculative episode and the green ones to the second one. Hence, fees are not only positively correlated but are more sensitive to price movements, at least during the first speculative period.

Finally, we also report the relationship between congestion and fees (see Figure 4). There is clearly a positive relationship since congestion reflects the scarcity of block space. This seems to be stable



(a) Pre-Segwit

(b) Post-Segwit

Figure 4: Fee Rates (in USD) and Congestion

across the period before and after Segwit. Note that the scale for fees is different, as prices are higher, especially during the first speculative window. Importantly, we include a non-parametric fit in these graphs, which indicates that the relationship between fees and congestion is convex, especially when congestion nears its limit.

3 Estimation

3.1 A Simple Model of Bitcoin Fees

We postulate that fees are driven by three considerations. First, the Bitcoin blockchain works like a "club good". Users need to post fees to have their transactions included in the blockchain. Since block sizes are capped at 1MB, users face a fixed supply of capacity to have their transactions processed. Hence, as blocks get full, users are willing to post higher fees.

Second, Bitcoin can be seen as a speculative instrument. As prices increase, Bitcoin becomes a more attractive financial asset to invest in. Hence, speculators have an incentive to pay higher fees to purchase Bitcoin.

Finally, volatility of prices may matter for both payments and speculation. If prices change a lot, it is important for payments to clear quickly to lock in the exchange rate of Bitcoin to other physical currencies. For speculators speed matters, as they want to move in and out of the market quickly, in particular during times when prices change a lot. These motivations lead us to formulate a simple model of Bitcoin fees that can be described by the following linear regression model.

$$\log \text{fee\_rate}_t = \alpha_0 + \alpha_1 \text{ congestion}_t + \alpha_2 \text{ congestion}_t^2 + \alpha_3 \log \text{price}_t + \alpha_4 \log |\Delta \text{price}_t| + \epsilon_t \quad (8)$$

Note that we have chosen to use log-transformed values for the fee rate, the price and its volatility. This is for two reasons. First, the fit of the model inceases significantly when using the log transformation. Second, the transformation allows us to interpret most parameters as elasticities.

We expect that the influence of all regressors is positive on fee rates. Figure 4 shows that there is a non-linearity in how fee rates vary with congestion. Hence, we have also introduced a quadratic term for congestion in the regression. We will also report results for two other specifications, where congestion *either* enters only linearly *or* enters only as a quadratic term. The motivation for the latter specification is that it forces a monotone relationship between fee rates and congestion, which can be motivated from basic theory (see, for example, Easley et al. (2019) and Chiu and Koeppl (2019)).

3.2 Results

Our benchmark formulation exhibits a lot of heteroskedastic and autocorrelated errors. For that reason, we resort to a heteroskedasticity- and autocorrelation-consistent (HAC) estimator when reporting standard errors on our regression results.[14] The results presented in Table 2 use a quadratic spectral kernel (see Andrews (1991)).

From a theoretical perspective, there is an endogeneity problem, as we are working with equilibrium realizations of average fee rates. As congestion increases, we expect fees to increase for two reasons. First, there is a selection by miners for high fees. Second, participants will pledge higher fees to increase their chances of getting into the blockchain. As fees increase, however, some participants will choose not to make a payment in Bitcoin, switch to other means of payment or simply wait. In other words, we expect congestion to be endogenous.

To address this issue, we introduce an instrument – congestion lagged by one day – and re-estimate our benchmark specification with 2SLS, still adjusting our standard errors with an HAC estimator. Table 3 reports the results from an IV regression. We also include the results from a Durbin-Wu-Hausman test, which calls for an IV approach to be warranted with a p-value close to 0.

There are several interesting findings from the benchmark model. First, fee rates are significantly and positively related to congestion. Second, the price level and price changes both have a significant positive impact on fee rates. The coefficient estimates on price and volatility are minimally affected

[14] See the appendix for a comparison of different estimators.

Table 2: Benchmark Results (OLS with HAC Estimator)

	Dependent variable: log Fee Rate		
	(1)	(2)	(3)
Congestion	3.7616		−19.0530
	(0.2270)		(1.5350)
	p = 0.0000***		p = 0.0000***
Congestion^2		2.4790	14.4031
		(0.1341)	(1.0167)
		p = 0.0000***	p = 0.0000***
log Price	0.7350	0.7279	0.7034
	(0.0248)	(0.0239)	(0.0216)
	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.1093	0.1018	0.0801
	(0.0146)	(0.0141)	(0.0131)
	p = 0.0000***	p = 0.0000***	p = 0.0000***
Constant	−8.9489	−7.5006	0.1482
	(0.2275)	(0.1751)	(0.5936)
	p = 0.0000***	p = 0.0000***	p = 0.8029
Observations	1,917	1,917	1,917
R^2	0.8093	0.8193	0.8416
Adjusted R^2	0.8090	0.8190	0.8413
Residual Std. Error	0.6523 (df = 1913)	0.6350 (df = 1913)	0.5946 (df = 1912)
F Statistic	2,705.63*** (df = 3; 1913)	2,890.41*** (df = 3; 1913)	2,540.40*** (df = 4; 1912)

Note: *p<0.1; **p<0.05; ***p<0.01

Table 3: Benchmark Results (IV with HAC estimator)

	Dependent variable: log Fee Rate		
	(1)	(2)	(3)
Congestion	4.9114		−32.3181
	(0.3040)		(3.3204)
	p = 0.0000***		p = 0.0000***
Congestion^2		3.1732	23.2688
		(0.1768)	(2.0920)
		p = 0.0000***	p = 0.0000***
log Price	0.7229	0.7148	0.6758
	(0.0242)	(0.0234)	(0.0211)
	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.0921	0.0839	0.0505
	(0.0148)	(0.0144)	(0.0135)
	p = 0.0000***	p = 0.0000***	p = 0.0002***
Constant	−9.7663	−7.8438	5.1947
	(0.2631)	(0.1791)	(1.3120)
	p = 0.0000***	p = 0.0000***	p = 0.0001***
Observations	1,917	1,917	1,917
R^2	0.7999	0.8106	0.8251
Adjusted R^2	0.7996	0.8103	0.8247
Residual Std. Error	0.6682 (df = 1913)	0.6501 (df = 1913)	0.6249 (df = 1912)
DWH Statistic	118.6	126.4	91.28
Note:			*p<0.1; **p<0.05; ***p<0.01

by using an IV approach and by changing from a linear to a quadratic fit, with respect to our congestion variable.

In what follows, we rely on the specification with only a quadratic term for congestion and relegate results for the other specifications to the appendix. Using only a quadratic term avoids the possibility of a negative relationship between fee rates and congestion for low levels of congestion, which seems theoretically implausible. Our estimates also tend to be very robust across the different specifications, as we show in the appendix.

3.3 Natural Experiments: Speculation and Segwit

Our simple benchmark specification does not take into account some important features in the data. First, there were two periods where the price of Bitcoin rose sharply, with the first one being followed by a sharp decline. As pointed out earlier, we interpret these time periods as speculative periods that added some additional, temporary demand for Bitcoin transactions.

Second, in August 2017, the Segwit improvement protocol came into effect. This protocol allows for more transactions to be included in a block. Hence, as usage of the protocol increases, the capacity for payments increases on the Bitcoin blockchain. We take these effects into account by estimating the following model that controls for these demand and supply shifts.

$$\log \text{fee\_rate}_t = \beta_0 + \beta_1 \text{congestion}_t^2 + \beta_2 \log \text{price}_t + \beta_3 \log |\Delta \text{price}_t| + \beta_4 \text{percent\_segwit}_t + \beta_5 D_t + \epsilon_t \quad (9)$$

The dummy variable D_t takes a value of 1 during speculative windows and 0 otherwise. Hence, one would expect that a positive coefficient would pick up an increase in the fee rate due to speculation. The second additional variable picks up the usage of Segwit as a percentage of all transactions. As users use the Segwit protocol, the space for transactions in blocks increases, which is equivalent to providing more throughput for transactions – or, equivalently, supply – in the Bitcoin blockchain. Consequently, we expect the sign of the coefficient to be negative.

Table 4 reports the results, including several interaction terms for the speculative period. We report only IV results based on 2SLS with lagged congestion as an instrumental variable, as Durbin-Wu-Hausman tests indicate that we should reject the hypothesis that simple OLS estimates are consistent.

The parameter estimates remain essentially unchanged across the different models. Also, in all

Table 4: Full Model – Quadratic Specification / IV with HAC

	Dependent variable: log Fee Rate			
	(1)	(2)	(3)	(4)
Congestion^2	2.1988	2.2353	2.2100	2.2214
	(0.1098)	(0.1131)	(0.1067)	(0.1139)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Price	1.2642	1.2743	1.2639	1.2676
	(0.0452)	(0.0471)	(0.0455)	(0.0492)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.0317	0.0304	0.0415	0.0396
	(0.0085)	(0.0084)	(0.0092)	(0.0094)
	p = 0.0002***	p = 0.0003***	p = 0.00001***	p = 0.00003***
Segwit (%)	−3.3786	−3.3880	−3.4166	−3.4143
	(0.2499)	(0.2538)	(0.2528)	(0.2524)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
D	0.1483	0.9136	0.4082	0.6443
	(0.0729)	(0.7313)	(0.1268)	(0.8385)
	p = 0.0420**	p = 0.2117	p = 0.0014***	p = 0.4424
D × log Price		−0.0821		−0.0295
		(0.0749)		(0.0958)
		p = 0.2733		p = 0.7586
D × log Volatility			−0.0502	−0.0428
			(0.0186)	(0.0264)
			p = 0.0072***	p = 0.1050
Constant	−10.6319	−10.7315	−10.6611	−10.6925
	(0.2767)	(0.3003)	(0.2781)	(0.3153)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
Observations	1,917	1,917	1,917	1,917
R^2	0.9099	0.9095	0.9102	0.9100
Adjusted R^2	0.9097	0.9093	0.9099	0.9097
Residual Std. Error	0.4486 (df = 1911)	0.4496 (df = 1910)	0.4480 (df = 1910)	0.4485 (df = 1909)
DWH Statistic	112.4	113.9	113.1	111.4

Note: *p<0.1; **p<0.05; ***p<0.01

models, fee rates tend to be higher during speculative periods, but this effect is not always highly significant.

Congestion still matters when we take into account the speculative window and price developments during that time period. A 1% increase in congestion increases fee rates by about 4% when the level of congestion is around 85%. This confirms our intuition that higher congestion on average increases fees, not only because miners can select higher fee transactions but also because users submit larger fees to get their transactions confirmed in the blockchain.

The impact of Segwit on fees is always negative at a very high level of significance. This gives us some confidence in treating this parameter estimate seriously later on. The effect seems to be relatively large. A 1% increase in the usage of Segwit tends to reduce the fee rate by more than 3%.

The bitcoin price significantly increases fees across all specifications. More importantly, during the speculative window this influence increases, but this effect is not always significant. Using the first column in the table, a 1% increase in price tends to raise the fee rate by about 1.26%. Interestingly, the price and its volatility tend to matter less during the speculative time periods.

This indicates that (i) Bitcoin transactions are not necessarily driven by pure payment concerns and (ii) that during speculative periods, investors are willing to pay more to purchase bitcoins quickly. The reason is, with a payment motive, the transaction value in USD is constant and, hence, fees in USD should not depend on the price of Bitcoin. In financial transactions, however, fees in USD are seen as a percentage of the entire transaction, and such transaction costs tend to matter less when the value of the transaction increases. This could explain the negative sign on the cross effect between speculative periods and price.

3.4 Robustness

We now check the robustness of our results with respect to the variable used for congestion. In Table 5, we report the results for our full model when using these other measures. Note that all models require an IV approach, as the Durbin-Wu-Hausman test is significant. Our estimation results are very robust. The impact of prices on fee rates remains positive and highly significant, while Segwit still has a negative impact of the same magnitude.[15]

4 Sensitivity of Bitcoin Fees

We now use our estimates to help us determine how much revenue can be generated from fees in Bitcoin. Table 6 reports the summary statistics that we use to assess the impact on fees. All the data are taken for the period after the introduction of Segwit and for the period between the two speculative episodes.

This is important since we are explicitly trying to control for time periods where bitcoin transactions are dominated by short-run speculation and not by payment purposes or long-run investment. Moreover, in all of our quantitative exercises we rely on the estimated coefficients where we control for the speculative window by introducing a dummy variable (see Table 4 above).

The table shows that the usage of Segwit does not depend on whether blocks are full or not. Also, the increase in the fee rate when blocks are full during a day is larger than predicted by our estimates. However, this is not surprising in that the variable measuring congestion is censored at 1. We therefore also report the Mempool size.

[15] We also include other lagged variables referring to congestion as additional instruments and conduct a GMM estimation. This allows for an overidentification test where we can test the joint hypothesis that the model is misspecified and some instruments are not valid. For all combinations of such additional instruments, the model is misspecified or the instruments are not valid.

Table 5: Robustness – Different Measures for Congestion

	Dependent variable: log Fee Rate			
	(1)	(2)	(3)	(4)
Congestion^2	2.2214			
	(0.1139)			
	p = 0.0000***			
log Mempool		0.4982		
		(0.0248)		
		p = 0.0000***		
Trans./Unconf.			−3.1033	
			(0.2630)	
			p = 0.0000***	
Med. Conf. Time				0.1436
				(0.0132)
				p = 0.0000***
log Price	1.2676	1.1547	1.2358	1.4147
	(0.0492)	(0.0482)	(0.0516)	(0.0530)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.0396	0.0526	0.0480	0.0464
	(0.0094)	(0.0105)	(0.0155)	(0.0130)
	p = 0.00003***	p = 0.000001***	p = 0.0021***	p = 0.0004***
Segwit (%)	−3.4143	−2.8756	−2.7032	−3.5864
	(0.2524)	(0.2217)	(0.2466)	(0.2436)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
D	0.6443	1.9543	2.4236	−1.7266
	(0.8385)	(0.5078)	(0.6867)	(0.6000)
	p = 0.4424	p = 0.0002***	p = 0.0005***	p = 0.0041***
D × log Price	−0.0295	−0.1517	−0.2219	0.1727
	(0.0958)	(0.0585)	(0.0830)	(0.0684)
	p = 0.7586	p = 0.0096***	p = 0.0076***	p = 0.0117**
D × log Volatility	−0.0428	−0.0802	−0.0559	−0.0150
	(0.0264)	(0.0200)	(0.0312)	(0.0267)
	p = 0.1050	p = 0.0001***	p = 0.0740*	p = 0.5734
Constant	−10.6925	−12.8499	−8.0758	−11.6261
	(0.3153)	(0.3334)	(0.3626)	(0.3784)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
Observations	1,917	1,704	1,703	1,704
R^2	0.9100	0.8817	0.7329	0.8328
Adjusted R^2	0.9097	0.8812	0.7318	0.8321
Residual Std. Error	0.4485 (df = 1909)	0.4425 (df = 1696)	0.6647 (df = 1695)	0.5261 (df = 1696)
DWH Statistic	111.4	229.6	234.2	93.28

Note: *p<0.1; **p<0.05; ***p<0.01

Table 6: Summary Statistics – Post Segwit / Excl. Speculative Periods

	Overall (757 days)	Full Blocks (100 days)
Fee Rate (USD/KB stripped)	2.84	6.48
Price (USD/BTC)	7,334	7,788
Segwit (%)	44.0	43.3
Congestion (%)	83.7	100
Avg. Transaction (per block)	2,050	2,409
Avg. Mempool (# of transactions)	12,204	32,023
Avg. Stripped Size (KB)	766.86	906.51

Note: Daily Averages from June 24, 2018 to July 19, 2020

In what follows, we conduct four quantitative exercises to assess the viability of Bitcoin in the future. For all our exercises, we correct for speculative demand for bitcoins.

The first two exercises look at how much fee revenue could be generated based on our estimated model for fees. We then infer the price increase that would be necessary to keep block rewards constant when Bitcoin ceases to pay significant block rewards.

Our last two exercises concern the optimal design of throughput. We first look at the impact of Segwit adoption on overall fee revenue. Finally, we ask what the optimal block size in Bitcoin would be to maximize fee revenue. To do so, we use our estimates to set up a simple linear demand model for Bitcoin transactions.

4.1 Demand – The Impact of Congestion

We first hold the usage of Segwit fixed and consider an increase in demand for transactions in Bitcoin that drives congestion to reach 100%. The block size of Bitcoin is restricted to 1MB and the adoption rate of Segwit is held constant around 40-50%. Hence, we can interpret changes in the level of congestion as shocks to the aggregate demand for Bitcoin transactions that shift the demand curve upwards, inducing a movement along a given supply curve.

There are two impacts on fees from such a demand shock. First, the total stripped block size increases. Second, congestion increases from about 80-100%, which increases the fee rate per KB stripped. We can use the summary statistics and our estimate for the elasticity of fee rates to calculate the total impact on fees. Using our estimates from (9), we obtain

$$\Delta \log \text{ Fee Rate } = \beta_1 \times \Delta \text{ Congestion}^2$$

where $\beta_1 = 2.1988$ and Δ Congestion$^2 = 1 - 0.837^2$. Hence, $\Delta \log$ Fee Rate $= 0.6584$ so that the percentage increase in the fee rate is given by $e^{0.6584} - 1$ or 93.16%. This leads to an estimated fee rate of 5.49 USD/KB. Hence, the total change in fees per block is given by

$$5.49 \text{ USD/KB } \times 907 \text{ KB } - 2.84 \text{ USD/KB } \times 767 \text{ KB } = 2,797 \text{ USD}$$

meaning that fee revenue per block would roughly double.

We could also rely on transactions in the Mempool as an alternative congestion measure. This measure of congestion is not censored. According to the summary statistics, during periods of full capacity, the size of the Mempool increases by 162.4%. Hence, we have

$$\beta_1 \times \Delta log \text{ Mempool} = 0.4982 \times 162.4 = 81 \tag{10}$$

or an increase in the fee rate of about 81%. This translates into a slightly lower gain in overall fees of 2,484 USD.

Using the average price for Bitcoin for the time period, block rewards are on average about 88,500 USD with fees being only about 2.5% of the total reward. Hence, it is highly unlikely that total rewards per block in USD will stay constant, as the block reward in bitcoin will decline over time unless there is a very large increase in the usage of Bitcoin. Bitcoin throughput, however, is already fairly close to full capacity. Unless users accept very long wait times to settle transactions, which is unlikely, this implies that block capacity needs to increase with demand, albeit keeping congestion at a fairly high level.

4.2 Demand – Long-Run Price Trend

Our estimation shows that fee rates increase with price. Hence, we can exploit our estimate to see whether price increases alone can increase fee rates sufficiently to keep the total mining rewards constant as block rewards decrease over time. This thought experiment is based on the fact that constant block rewards keep the hashrate and, thus, the security of the blockchain constant.[16]

For this experiment, we take the average block reward of 88,500 USD as given and ask by how much the price of Bitcoin needs to increase to generate that amount as fee revenue. Holding the average stripped block size of 767 KB constant, the fee rate has to increase to about 115 USD/KB. This implies a significant cost increase for using Bitcoin as a payment instrument but could still be sufficiently small for large enough transaction sizes in USD.

Using the price elasticity of $\beta_2 = 1.2642$, the bitcoin price would need to be about 32 times the average one used for calculating the average reward for our sample. Alternatively, we could use the fact that in the long run congestion reaches 100% and use the data for when blocks are full. This would lower the required bitcoin price to about 13 times the average one used for calculating the average reward.

As Bitcoin block rewards mostly disappear over the next 20 years due to halving of the rewards, we can translate these measures into annual growth rates that are required to keep block rewards constant. This leaves us with a range of about 13.7% to 18.9% growth in the price per year. These figures are not out of line, taking into account that our linear trend estimates in Section 2 yield an average annual growth rate of about 60% for the bitcoin price over the 7 years of data we use.

4.3 Supply – The Impact of Segwit

There are two ways to increase throughput capacity further in Bitcoin.[17] The first is further adoption of the Segwit protocol by users, while the second is a formal change in the block size of 1 MB adopted by Bitcoin.

We regard Segwit adoption as a long-run decision by users that is not affected by demand fluctuations. As Figure 2 shows, the adoption rate of Segwit has been stable over time after an initial introduction phase. Hence, we can view the demand function for Bitcoin as fixed on average and think of a change in the usage of Segwit as shifting the effective supply, inducing a movement along this demand curve.

[16] This implicitly assumes that there are no extra incentives to double spend when the price of Bitcoin increases. This would not be the case when a double spending attack relied on using Bitcoin balances that were acquired long in the past.

[17] Another alternative is using different channels such as the Lightning Network, which settles transactions off the blockchain roughly based on the idea of netting transactions.

Segwit adoption reduces our congestion measure, as the blockweight of a Segwit transaction is smaller than the blockweight of the original transaction. To translate changes in Segwit into changes in congestion, we first regress the ratio of stripped size to block size on the percent of Segwit being used, or

$$\frac{\text{Stripped block size}_t}{\text{Block size}_t} = \rho_0 + \rho_1 \text{percent\_segwit}_t + \epsilon_t \tag{11}$$

using daily data for the time period after the introduction of Segwit. Our estimates for the coefficients are $\rho_0 = 0.99$ and $\rho_1 = -0.587$ (for details, see the appendix). Note that we can set the coefficient for ρ_0 simply to 1 for theoretical reasons. The coefficient ρ_1 is an estimate for the reduction ρ_i in block size due to the Segwit protocol (see equation (4)). Using the definition of blockweight, we thus obtain as our estimate that congestion changes by about $0.75\rho_1 = -0.441$ percentage points for each percentage of adoption of Segwit (see equation (3)).[18]

Consider then a situation where there is no usage of the Segwit protocol whatsoever. According to our estimate this would have increased the average daily congestion level by $44 \times 0.75\rho_1 = 19.404\%$ and consequently to a level of 100% congestion, from the mean of 83.7%. Beyond this indirect effect, there is also a direct effect from the reduction of Segwit, which is given by the estimated parameter β_4.

This gives us a preliminary estimate for the elasticity of fee rates with respect to Segwit usage. With full congestion, we have a percentage increase of the fee rate equal to

$$(e^{\beta_1(1-0.837^2)} - 1) \times 100 + \beta_4 \times 44 = 241.82$$

percent. Using the initial fee rate of 2.84 USD/KB and 1MB for a full block without Segwit,[19] our estimate indicates that fee revenue has changed by about 68% from 6,686 USD to 2,178 USD due to the introduction of Segwit.

4.4 Supply – Optimal Block Size

We next calculate an estimate for the maximum of fees that can be generated per block. Without full congestion, it is always optimal to decrease the block size in order to increase total fees raised per

[18] With Segwit, some block sizes have reached up to 2.3 MB in total, which translates into the stripped size being about 43.5% of the actual (raw) size of a full block. Hence, our estimate seems to be consistent with this fact using a maximum stripped block size of 1 MB.

[19] Recall that the stripped size equals the full block size without Segwit.

block. Since the capacity constraint is not binding, increasing congestion does not limit throughput, but increases the fee rate. For this reason, we look at a situation where demand is such that Bitcoin is at full capacity; i.e., all blocks are full during a day. In such a situation, an increase in the block size would decrease the fee rate but increase capacity. We now use our estimates to analyze this trade-off.

To do so, we again view the empirical relationship between Segwit and fee rates as a supply curve shifting along a given demand curve. We assume that the demand curve takes on a simple linear form described by

$$p(x) = (a + b) - bx \tag{12}$$

where x is the block size – now a design parameter – and p is the fee rate per KB stripped. The fee rate is equal to a whenever the design of the block size is $x = 1$ MB. Hence, the optimal block size maximizes total fee revenue

$$p(x)\alpha x \tag{13}$$

and is given by

$$x^* = \frac{a + b}{2b} \tag{14}$$

where α is a constant that translates actual block sizes into stripped block size for a fixed level of Segwit adoption (see equation (11)).

If we can pin down the two parameters – a and b – we obtain the optimal block size under the assumption that the blockchain remains at full capacity and that the demand curve is linear. For a, we consider the summary statistics for full blocks from above and set $a = 6.48$ USD/KB.

For b, which is the elasticity of fee rates with respect to block size x, we use our estimate for how Segwit changes fee rates. Note that we can translate changes in Segwit once again into changes in congestion according to equation (3). A change in Segwit is equivalent to a change in block size equal to

$$\frac{d \text{ block size}}{d \text{ \% Segwit}} = -\frac{1}{0.75\rho_1} = 2.27. \tag{15}$$

Since we measure block size in MB, this implies that 100% of Segwit would lead to an effective block size limit of 2.27 MB. This yields a change in fee rates of

$$\beta_4 \times \left(-\frac{1}{0.75\rho_1}\right) = 7.67 \tag{16}$$

percent due to a 1% change in block size. To obtain the parameter b, we use the fact that a 100% reduction (or $x = 0$) would yield an increase of 767% in fee rates or $b = 6.48 \times 7.67 = 49.7$ USD/KB.[20]

Finally, note that stripped size is a constant fraction of block size if we hold Segwit adoption constant. Hence, assuming that we have full congestion and a linear demand function for transactions, the block size that maximizes fees is given by

$$x^* = \frac{6.48 + 49.7}{2 \times 49.7} = 0.565. \tag{17}$$

Hence, given current Segwit adoption, the current block size limit is about 45% too high. Since Segwit adoption is at about 44% when blocks are full, the optimal stripped block size is at

$$(1 - 0.587 \times 0.44) \times 0.565 \text{ MB } = 0.419 \text{ MB}$$

so that the estimated maximum fee revenue is given by 11,777 USD.

To put this into perspective, this is about 1/8 of the fee revenue generated by total rewards over the sample we use in this section. Consequently, relying on fees only and an optimally designed block size would generate rewards that are equal to a block reward of 1.6 BTC using the average price of Bitcoin. Interestingly, this is close to the optimal block reward simulated in Chiu and Koeppl (2017).

5 Conclusions

Our empirical estimates confirm that the introduction of the Segwit protocol reduced congestion. As a consequence, we show that the protocol – while increasing throughput – has lowered fees significantly. Indeed, given current prices and average demand for Bitcoin transactions, in order to maximize fee revenue it would be optimal to have a much smaller block size or, equivalently, no adoption of the Segwit protocol at all.

This underscores the hesitation of the Bitcoin community to agree with a protocol change in 2015 when an increased block size of 8MB was under discussion. As Segwit has effectively doubled the block size, we show here that concerns about block size increases were warranted.

[20] Note that this implies that the intercept of the demand curve is at a fee rate of $a + b = 56.18$ USD/KB, which corresponds to the maximum fee rate of a Bitcoin transaction, given our linear model.

With revenue from block rewards falling over time to zero, fees will eventually need to generate all the revenue. If Segwit adoption remains at 40-50%, the maximum revenue for mining that could be generated through fees using an optimal design of the block size is only about 1/8 of current total rewards per block.

In summary, our paper casts some doubt on the sustainability of Bitcoin in the long run if it relies on high rewards to avoid double spending attacks. Sustained price increases of about 15-20% per year over the next 20 years would be required to keep block rewards at a constant level through to 2040. According to our empirical findings, such price increases would be accompanied by steep increases in transaction fees, possibly making regular payments in Bitcoin prohibitively expensive.

References

Andrews, D. (1991), “Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation”, Econometrica, 59, pp. 817-858.

Auer, R. (2019), “Beyond the Doomsday Economics of ‘Proof-of-Work’ in Cryptocurrencies”, BIS Working Paper No. 765.

Bai J. and Perron P. (2003), “Computation and Analysis of Multiple Structural Change Models”, Journal of Applied Econometrics, 18, pp. 1-22.

Böhme, R., Christin, N., Edelman, B. and Moore, T. (2015), “Bitcoin: Economics, Technology, and Governance”, Journal of Economic Perspectives, 29, pp. 213-238.

Chiu, J. and Koeppl, T. (2017), “The Economics of Cryptocurrency – Bitcoin and Beyond”, Queen’s Economics Department, Working Paper 1389.

Chiu, J. and Koeppl, T. (2019), “Blockchain-based Settlement of Assets Trades”, Review of Financial Studies, 32, pp. 1716-1753.

Easley, D., O’Hara, M. and Basu, S. (2019), “From Mining to Markets: The Evolution of Bitcoin Transaction Fees”, Journal of Financial Economics, forthcoming.

Huberman, G., Leshno, J. and Moallemi, C. (2017), “An Economic Analysis of the Bitcoin Payment System” Columbia Business School, Research Paper No. 17-92.

Lehar, A. and Parlour, C. (2020), “Miner Collusion and the Bitcoin Protocol”, mimeo, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3559894.

Appendix

Summary Statistics

Table 7: Block Level (averaged daily) or Daily Average

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Fee Rate (USD/KB stripped)	1,917	7.006	13.934	0.109	0.897	6.057	160.513
Price	1,917	7,923.414	9,641.191	372.470	1,259.530	9,525.600	61,200.000
Volatility	1,917	231.370	498.774	0.020	13.540	231.630	7,724.000
Blockweight (% of 4 KWU)	1,917	0.857	0.133	0.379	0.769	0.978	0.999
Transactions/Unconfirmed	1,704	0.267	0.251	0.009	0.074	0.374	1.000
Med. Confirmation Time (min)	1,704	10.468	3.598	3.367	7.848	12.356	29.250
Mempool (# of unconfirmed)	1,704	24,309.800	33,746.460	174.240	4,577.227	28,800.910	226,159.700

Standard Errors – Comparison

Table 8 compares standard errors for our three benchmark models across three different methods. The first one uses spherical errors, the second one uses an HAC estimator based on a quadratic spectral kernel (see Andrews (1991)) and the last one uses the methods of Newey and West.

Table 8: Standard Errors – Benchmark

	(Intercept)	Cong.	log Price	log Volatility	
SE	0.15255	0.11841	0.01883	0.01128	
QS	0.22754	0.22696	0.02478	0.01457	
NW	0.25722	0.26367	0.02857	0.01618	
	(Intercept)	Cong.^2	log Price	log Volatility	
SE	0.12893	0.07245	0.01834	0.01100	
QS	0.17509	0.13409	0.02393	0.01414	
NW	0.19874	0.15784	0.02743	0.01571	
	(Intercept)	Cong.	Cong.^2	log Price	log Volatility
SE	0.48073	1.15912	0.72859	0.01724	0.01039
QS	0.59358	1.53501	1.01674	0.02158	0.01310
NW	0.76484	2.00373	1.35161	0.02782	0.01610

Figure 5 shows standard errors plotted against the fitted values for the IV regressions of the benchmark model with the quadratic term only on the left and the full model with all interaction terms on the right. Similar graphs arise for the other specifications, all showing that residuals are not normally distributed.

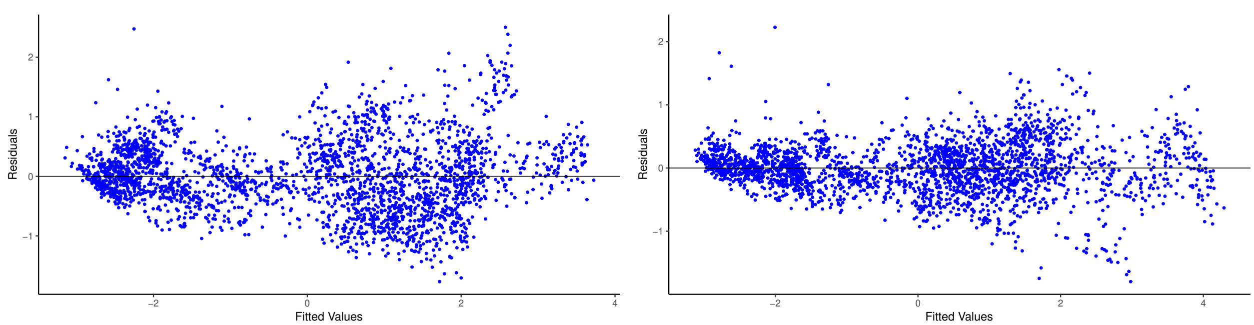


Figure 5: Heteroskedastic Errors

Linear and Linear-Quadratic Specifications

Tables 9 and 10 show the regression results from the alternative specifications. Three features stand out. First, the impact of price on fee rates does not change across specifications. Second, the impact of Segwit is roughly constant across the three models. Finally, the impact of congestion on fee rates does not vary much from the specification used in the main part of the paper, if one looks at a level of around 80% congestion.

Table 9: Linear Model (IV with HAC Estimator)

	Dependent variable: log Fee Rate			
	(1)	(2)	(3)	(4)
Congestion	3.4058	3.4581	3.4250	3.4349
	(0.1842)	(0.1897)	(0.1793)	(0.1910)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Price	1.2762	1.2853	1.2759	1.2779
	(0.0456)	(0.0480)	(0.0464)	(0.0506)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.0355	0.0343	0.0455	0.0444
	(0.0087)	(0.0086)	(0.0095)	(0.0096)
	p = 0.00005***	p = 0.0001***	p = 0.000002***	p = 0.000004***
Segwit (%)	−3.4416	−3.4505	−3.4799	−3.4788
	(0.2518)	(0.2566)	(0.2582)	(0.2581)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
D	0.1714	0.8590	0.4350	0.5647
	(0.0752)	(0.7446)	(0.1325)	(0.8590)
	p = 0.0229**	p = 0.2489	p = 0.0011***	p = 0.5111
D × log Price		−0.0737		−0.0162
		(0.0763)		(0.0981)
		p = 0.3342		p = 0.8692
D × log Volatility			−0.0509	−0.0468
			(0.0192)	(0.0270)
			p = 0.0080***	p = 0.0828*
Constant	−12.0011	−12.1113	−12.0383	−12.0595
	(0.2897)	(0.3251)	(0.2931)	(0.3469)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
Observations	1,917	1,917	1,917	1,917
R^2	0.9055	0.9052	0.9058	0.9057
Adjusted R^2	0.9053	0.9049	0.9055	0.9054
Residual Std. Error	0.4593 (df = 1911)	0.4603 (df = 1910)	0.4587 (df = 1910)	0.4591 (df = 1909)
DWH Statistic	110.6	111.5	111.3	109.0

Note: *p<0.1; **p<0.05; ***p<0.01

Table 10: Linear-Quadratic Model (IV with HAC Estimator)

	Dependent variable: log Fee Rate			
	(1)	(2)	(3)	(4)
Congestion	−18.4972	−18.4531	−18.3138	−18.3402
	(2.7033)	(2.7398)	(2.7170)	(2.7265)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
Congestion^2	13.7670	13.7750	13.6613	13.6950
	(1.7107)	(1.7263)	(1.7174)	(1.7161)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Price	1.2449	1.2548	1.2448	1.2503
	(0.0428)	(0.0439)	(0.0429)	(0.0452)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.0161	0.0148	0.0241	0.0212
	(0.0083)	(0.0082)	(0.0089)	(0.0089)
	p = 0.0519*	p = 0.0707*	p = 0.0071***	p = 0.0174**
Segwit (%)	−3.2553	−3.2647	−3.2870	−3.2834
	(0.2382)	(0.2409)	(0.2401)	(0.2394)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
D	0.0090	0.7553	0.2190	0.5727
	(0.0685)	(0.6433)	(0.1225)	(0.7358)
	p = 0.8952	p = 0.2406	p = 0.0741*	p = 0.4365
D × log Price		−0.0800		−0.0442
		(0.0658)		(0.0840)
		p = 0.2241		p = 0.5992
D × log Volatility			−0.0403	−0.0292
			(0.0183)	(0.0247)
			p = 0.0277**	p = 0.2376
Constant	−3.2539	−3.3685	−3.3505	−3.3870
	(1.1420)	(1.1784)	(1.1496)	(1.1787)
	p = 0.0045***	p = 0.0044***	p = 0.0037***	p = 0.0042***
Observations	1,917	1,917	1,917	1,917
R^2	0.9165	0.9163	0.9168	0.9166
Adjusted R^2	0.9162	0.9160	0.9165	0.9162
Residual Std. Error	0.4319 (df = 1910)	0.4327 (df = 1909)	0.4313 (df = 1909)	0.4319 (df = 1908)
DWH Statistic	66.73	67.57	66.8	66.29

Note: *p<0.1; **p<0.05; ***p<0.01

Auxiliary Regression

Table 11: Impact of Segwit on Block Size

	Dependent variable: Stripped Block Size/Block Size
Segwit (%)	−0.5868
	(0.0117)
	p = 0.0000***
Constant	0.9894
	(0.0041)
	p = 0.0000***
Observations	1,316
R^2	0.854
Adjusted R^2	0.854
Residual Std. Error	0.034 (df = 1314)
F Statistic	7,694.490*** (df = 1; 1314)
Note: OLS w/ HAC estimator	*p<0.1; **p<0.05; ***p<0.01

Robustness – Hashrate

Bitcoin demand could also depend on how secure the blockchain is for making payments. A proxy for security is the hashrate on the Bitcoin network. Higher mining activities show up as a higher hashrate to solve for new blocks.

The Bitcoin protocol is designed so that the average block time is roughly constant at 10 minutes. Hence, a larger hashrate speeds up the time to solve for a new block and, consequently, the protocol increases the difficulty of solving a block. A more difficult problem also makes it more difficult to attack the Bitcoin blockchain via a double spending attack. In summary, a higher hashrate increases the difficulty, which leads to more security for making payments in bitcoin.

Table 12 shows the correlation of the hashrate with other variables in our empirical specification.

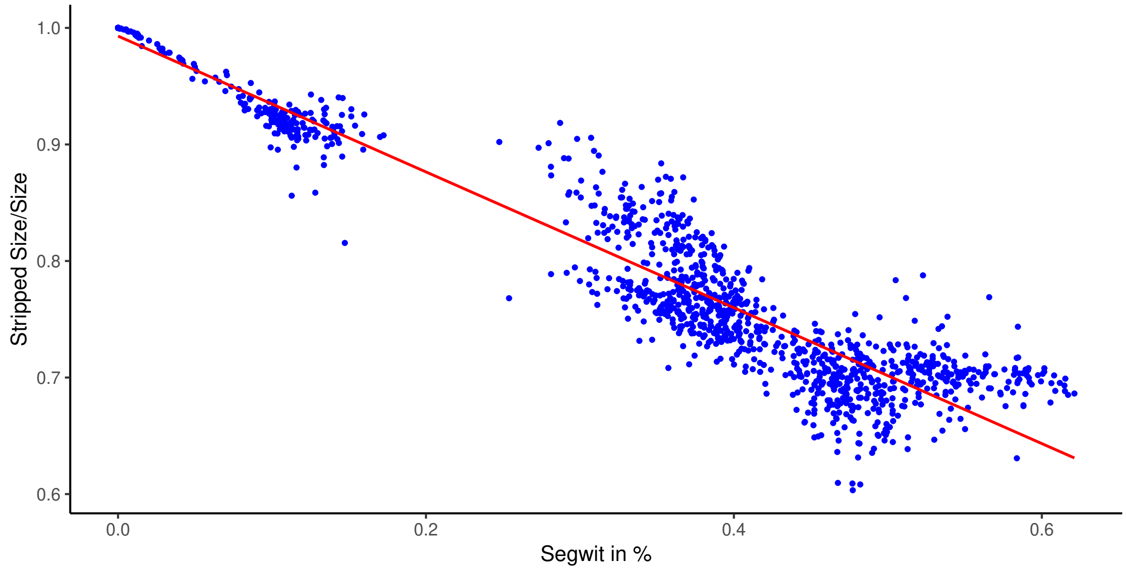


Figure 6: Impact of Segwit on Block Size

This raises some concerns, as our other key variables such as price and congestion clearly move together with the hashrate. This raises a problem of collinearity between the hashrate and two of our other explanatory variables.[21]

Table 12: Correlation w/ Hashrate

	Hashrate	Price	Reward	Congestion	Fee Rate
Hashrate	1	0.750	0.687	0.481	0.348
Price	0.750	1	0.932	0.442	0.698
Reward	0.687	0.932	1	0.489	0.783
Congestion	0.481	0.442	0.489	1	0.374
Fee Rate	0.348	0.698	0.783	0.374	1

Furthermore, fees are part of the block reward and thus matter for the hashrate, so there is again an endogeneity problem. Below, we report results when we adjust our model to include the logarithm of the hashrate. The first model uses it as an additional explanatory variable in our benchmark IV

[21] Also, one can easily verify that halving the block reward had little to no impact on the growth of the hashrate.

regression. The second one uses both lagged congestion and the lagged hashrate as instrumental variables. The last one uses the hashrate only as an additional instrument for a GMM estimation. Here the Hansen J Test shows a p-value of 0.06338, so we cannot reject the hypothesis that both instruments are valid at a 5% confidence level. Hence, the hashrate could serve as an instrument to control for the endogeneity of some other variables.

Table 13: Models w/ Hashrate

	Dependent variable: log Fee Rate		
	IV Benchmark	*IV w/ add. Instrument*	*GMM*
	(1)	(2)	(3)
Congestion^2	2.2778	2.3172	2.1672
	(0.1250)	(0.1292)	(0.1077)
	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Price	1.3537	1.3984	1.2610
	(0.0688)	(0.0749)	(0.0459)
	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.0299	0.0291	0.0296
	(0.0085)	(0.0086)	(0.0085)
	p = 0.0005***	p = 0.0008***	p = 0.0005***
log Hashrate	−0.1495	−0.2240	
	(0.0800)	(0.0939)	
	p = 0.0618*	p = 0.0171**	
Segwit (%)	−2.6497	−2.2863	−3.3571
	(0.4494)	(0.5091)	(0.2568)
	p = 0.0000***	p = 0.00001***	p = 0.0000***
D	0.1117	0.0934	0.1544
	(0.0778)	(0.0791)	(0.0730)
	p = 0.1516	p = 0.2379	p = 0.0345**
Constant	−4.9707	−2.1475	−10.5805
	(3.0071)	(3.5390)	(0.2809)
	p = 0.0985*	p = 0.5440	p = 0.0000***

Note: *p<0.1; **p<0.05; ***p<0.01

The results show that the coefficients on all of our main variables are very close to the benchmark model we use in our analysis in Section 4. The hashrate can be significant, albeit on a much lower level than the other main variables. Also, the sign is negative, meaning that a 1% increase in the hashrate lowers fee rates. This is hard to interpret. One possible explanation is that block times decrease over small time intervals as difficult adjustments lag by up to two weeks. This can lower confirmation lags, which leads to less urgency to have short verification lags (see Chiu and Koeppl (2019)).

Benchmark Results for Extended Data Series

Table 14 presents results for the IV regressions in the extended model for an extended data set that starts on January 1, 2014. The results do not change, except for a smaller impact of congestion on fee rates. This is not surprising, as congestion rates were lower in 2014-2016 compared to post-2016. Interestingly, including more data increases the fit of the model.

Table 14: Extended Data Set (2014-01-01 to 2021-03-31)

	Dependent variable: log Fee Rate			
	(1)	(2)	(3)	(4)
Congestion^2	1.4039	1.4024	1.4037	1.4009
	(0.0693)	(0.0706)	(0.0690)	(0.0701)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Price	1.2818	1.2774	1.2815	1.2728
	(0.0389)	(0.0395)	(0.0392)	(0.0409)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
log Volatility	0.0459	0.0457	0.0486	0.0539
	(0.0068)	(0.0070)	(0.0073)	(0.0069)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
Segwit (%)	−3.7030	−3.6961	−3.7146	−3.7266
	(0.2395)	(0.2417)	(0.2414)	(0.2419)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
D	0.1790	−0.5535	0.2695	−0.8651
	(0.0855)	(0.7794)	(0.1334)	(0.9097)
	p = 0.0365**	p = 0.4777	p = 0.0435**	p = 0.3417
D × log Price		0.0780		0.1413
		(0.0783)		(0.1018)
		p = 0.3191		p = 0.1654
D × log Volatility			−0.0172	−0.0536
			(0.0180)	(0.0263)
			p = 0.3408	p = 0.0413**
Constant	−10.1268	−10.0943	−10.1297	−10.0769
	(0.2246)	(0.2303)	(0.2262)	(0.2376)
	p = 0.0000***	p = 0.0000***	p = 0.0000***	p = 0.0000***
Observations	2,647	2,647	2,647	2,647
R^2	0.9536	0.9538	0.9536	0.9540
Adjusted R^2	0.9535	0.9536	0.9535	0.9538
Residual Std. Error	0.4045 (df = 2641)	0.4040 (df = 2640)	0.4045 (df = 2640)	0.4032 (df = 2639)
DWH Statistic	65.99	64.15	63.68	63.33

Note: *p<0.1; **p<0.05; ***p<0.01