Queen's Economics Department Working Paper No. 1423

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

Collin Brown

Jonathan Chiu

Thorsten V. Koeppl

Department of Economics
Queen's University
94 University Avenue
Kingston, Ontario, Canada
K7L 3N6

10-2019

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

Collin Brown — ESDC

Jonathan Chiu — Bank of Canada

Thorsten V. Koeppl — Queen's University

October, 2019

(First Version: August, 2019)

Abstract

We use block level data from the Bitcoin blockchain to estimate the impact of congestion and the USD price on average 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 80%. Fee revenue could be maximized at a blocksize of about 0.6 MB when Segwit adoption remains at 40%. At this blocksize, maximum fee revenue would be roughly 1/8 of the current block reward – or the equivalent of 1.6375 BTC as a reward in the long run given current prices and demand for Bitcoin.

Keywords: Bitcoin, Payment Systems, Fees, Congestion, Segwit Protocol

JEL Classification: E42, G2

*The views expressed in this paper are not necessarily the views of the Bank of Canada or ESDC. 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.

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, competition among miners will be high, giving little or no incentives to double spend, since the PoW protocol requires 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 declined from intially 50 BTC to 0.[3]

Hence, it is important to understand if fees only can generate sufficient rewards to ensure that the Bitcoin blockchain remains tamper proof. We use block level data from the last 3 years to answer this question and find that fees at a maximum can only generate about 1/8 of the current level of block rewards.

Our estimation relies on identifying a demand model for Bitcoin transactions. We first regress fee rates – the average fee in USD paid per kbyte used to make a transaction in bitcoin – on USD prices of bitcoin and the level of congestion. There is a positive relationship between rates and these variables. This is not too surprising. Since the supply of bitcoins is limited, if demand for bitcoin as a financial asset increases, its price increases as well. This in turn will increase fee rates as users would like to settle their transactions. Similarly, 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 fixed supply of block space, they have an incentive to post higher fees to get their transactions settled.

We then rely on two events to identify an aggregate demand curve. First, there was a period in late

[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] Currently, the reward for mining a block is 12.5 BTC.

2017 until early 2018 where speculation led to sharp increases in the price of Bitcoin and fee rates. Many people have interpreted this as a bubble period and we show that controlling for this episode is important for our estimates. Second, in mid 2017, the Segwit protocol has been introduced in Bitcoin which effectively increases the supply of block space. The adoption of Segwit acts like shifts along an aggregate demand curve once we control for the speculative period. Changes in congestion and USD price are interpreted as demand shocks.

This allows us to have a simple model of fee revenues given aggregate demand for Bitcoin transactions based on our empirical model of fee rates. We then assume that congestion remains near 100% when varying blocksize to investigate what the optimal blocksize in Bitcoin would be to maximize fee revenue given average current conditions such as USD prices and Segwit adoption.[4]

Broadly speaking, our results indicate that – to maximize fee revenue at current prices – it is optimal to have much higher congestion. First, we find that the current blocksize is about 40% too large in order to maximize transaction fees. Second, Segwit helped to reduce capacity pressures, but it also lowered fees significantly by about 80%. 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, if one would like to maintain a high level of block rewards, one would need sustained price increases, since currently fees could offer at most about $10,000 in block rewards using the optimal block size.

Our main contribution is to develop a crude empirical model based on block level data for how the costs of conducting transactions in Bitcoin influence the demand for such transactions. A key aspect of our analysis is to use some natural experiments – the rapid price increase in late 2017 and the introduction of Segwit – to identify supply and demand shocks. We then put our empirical estimates to work to re-evaluate the design of Bitcoin in light of its long-run reward structure.

The economics literature on fees in Bitcoin is very small. The closest work to our own is Easley et al. (2019) who analyze the incentives for Bitcoin users to pledge positive fees based on a theoretical queuing model. They find that median wait times as a measure of congestion drive fee revenue. Huberman et al. (2019) 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 transactions demand in Bitcoin. They also do not consider the implications of the demand for

[4] Having full congestion is important to have a non-trivial model for optimal blocksize.

transactions for the optimal design of Bitcoin.[5]

Only a few papers in economics have tackled the issue how a declining reward structure will influence mining in Bitcoin in the long-run. Two exemptions are Auer (2019) and Chiu and Koeppl (2017). Both papers use the double spending problem to evaluate the safety and the cost of Bitcoin for settling transactions. As pointed out earlier, mining rewards play a major role in their analyses. Auer (2019) is sceptical that rewards will be enough in the long-run to avoid double spending. Interestingly, our estimate here put the reward that could be generated with fees alone in the long run at about 1.6375 BTC. This is close to the total amount of reward 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 August 1, 2016 to July 28, 2019. During this time the block reward for Bitcoin is constant at 12.5 BTC. We use block level data to compute daily average values for transaction fees, bitcoin prices and block summary statistics that measure the demand for Bitcoin transactions.[6] We also use auxiliary data from Bitcoin's mempool to complement this data set.[7] This gives us a data set of $N = 1,092$ observations. We report some summary statistics of the data set in the Appendix.

There are two important events during this time span which we exploit in our analysis. First, there is a rapid increase in prices followed by a price collapse. Below, we estimate two break points in the price data series which are October 30, 2017 and May 23, 2018. We refer to this time span as a "speculative period". The second event is the introduction of the SEWGIT protocol in block 481,824 on August 24, 2017 which changed the effective throughput of the Bitcoin blockchain. This is significant as it allows for more transactions to be included in the block and is a controlled

[5]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 PoW that is optimally designed in terms of its block time and block size.

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

[7]All auxiliary data has been obtained from `blockchain.info`.

experiment of how an increase in the potential throughput of the Bitcoin blockchain influences fees.

2.2 What is Segwit?

Traditionally, blocks included all the information that is relevant for a Bitcoin transactions. This is information about the transaction itself and metadata that allow to verify that a transaction is legitimate. With the SEGWIT ("segregated witness") improvement 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 understand 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, a legacy miner cannot include segwit transactions. While he can form a block containing both types of transactions, he cannot communicate the Segwit data. Consequently, all segwit-capable nodes would reject the block. Notwithstanding, non-segwit miners can build on the same chain as segwit miners, since the digest of the chain for mining a new block only uses blockheaders which do not depend on the type of transactions.

Both miners and users have an incentive to switch to 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 lower size and, thus, are cheaper given fees offered per byte.[8]

2.3 Bitcoin Prices

We calculate prices on the daily level directly from the block data. To do so, we use total fees in USD per block and divide by total fees in bitcoin per block. This gives us a consistent price

[8]Adoption could be slow due to 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.

estimate as reported in the block data. We also calculate a simple volatility measure as the absolute value of the first difference in prices.

Looking at a time series, we first fit our price sequence to a linear and a filtered trend using the HP filter. Using the method of Bai and Perron (1993), we estimate breakpoints for the time series of prices with the restriction that we limit the number to a maximum of two. Figure 1 shows the time series together with the breakpoints which are given by `10-30-2017` and `05-23-2018`. These points later define a dummy variable that is set to 1 for this period and identifies this time window as a speculative period.

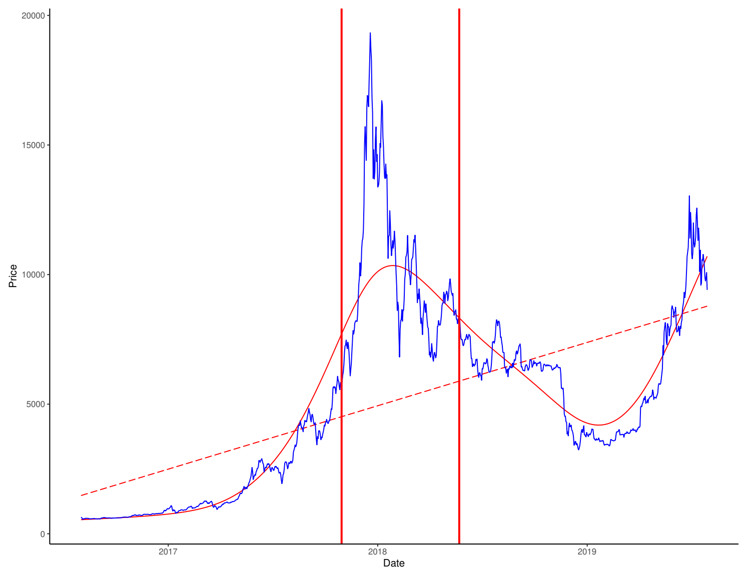


Figure 1: Bitcoin Price, Trend (linear & smoothed), Breakpoints

2.4 Congestion

2.4.1 Blockweight

The key variable in our analysis is a measure of *congestion* which expresses how much of the total, possible throughput in Bitcoin is being used by its users. We 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.

With the Segwit protocol there are two constraints. First, there is a restriction per block by "weight". 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. Note that we have expressed this as a fraction of the transaction size x_i. We then have as the first constraint

$$\sum_i 3(x_i - \rho x_i) + x_i \leq 1 \text{ Mvbyte } = 4 \text{ MB} \tag{3}$$

where vbyte is the weight of a transaction and the block is constrained by 1 Mvbyte which is equivalent to 4MB.

The second constraint is that blocks with witness data stripped still 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. If one observes blocksizes above 1 MB, it just reflects 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.

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

$$\text{congestion } = \frac{\text{blockweight}}{4 \text{ MB}} \tag{5}$$

as our congestion measure. To calculate a time series of congestion, we average blocks over each day in our sample.[9] We also use the percentage of segwit transactions in a block – averaged for each day – as a measure of the adoption of the improved 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 window for the price series and the dashed vertical line delineates the introduction date of Segwit.

[9]This also avoids the problem that blocks are found not necessarily in a specific time. Hence, blocks tend to be fuller if a block has not been found for a long time and smaller if blocks are found quickly after each other.

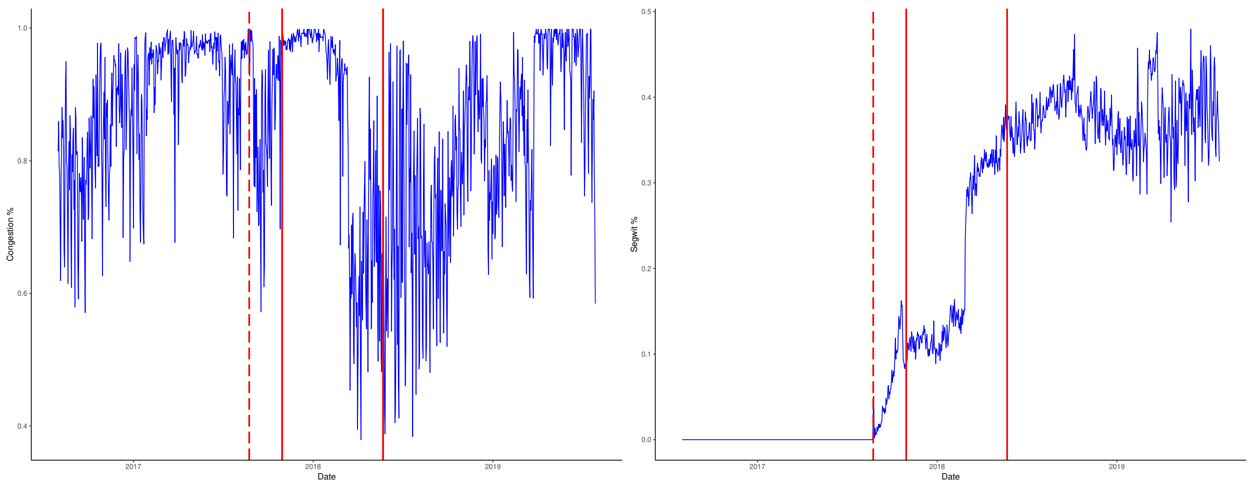


Figure 2: Congestion and Segwit Adoption

As segwit transactions increase over time, there seems to be less congestion until at the end of our time series when congestion increases as there are more transactions on the blockchain. Interestingly, during the speculative window, Segwit at first is adopted slowly until it jumps up sharply. Recently, adoption has been fairly constant around 40%.

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 in the chain. Hence, a larger number of unconfirmed transactions means more congestion as the queue of transactions to get into the blockchain is longer.

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 into 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. This measure turns out to be not very informative as the information is not very detailed. 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

	Congestion	Mempool	Transactions/Mempool	Med. Conf. Time
Congestion	1	0.456	−0.647	0.474
Mempool	0.456	1	−0.546	0.398
Transactions/Mempool	−0.647	−0.546	1	−0.430
Med. Conf. Time	0.474	0.398	−0.430	1

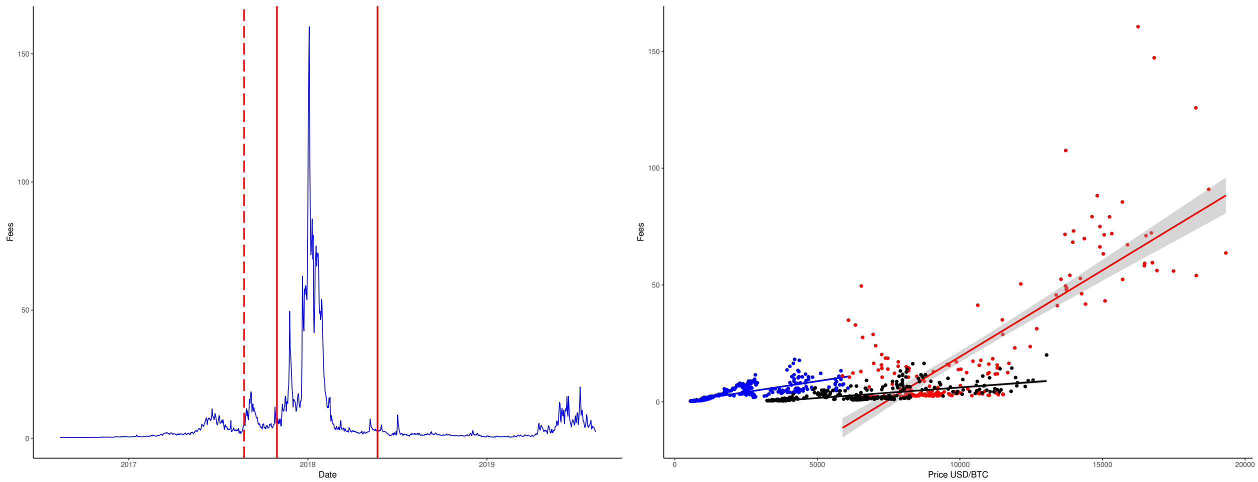
2.5 Fee Rates

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.[10]

We again need to take Segwit into consideration when we measure the cost of a transaction appropriately. After Segwit, what matters for a transaction is its size stripped of segwit data. Consider two segwit transactions. Miners prefer to include the transaction with the smaller weight for the same transaction size and fee. Hence, what matters for a user is the stripped transaction size to get a transaction to be included into a block. Consider now 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 the overall transaction size is not relevant, but the stripped size of an individual transaction matters in terms of adding congestion and, hence, matters for posting fees.

On the block level, we thus use the stripped blocksize which corresponds to constraint (4) to

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



(a) Time Series

(b) Correlation w/ Congestion

Figure 3: Fee Rates

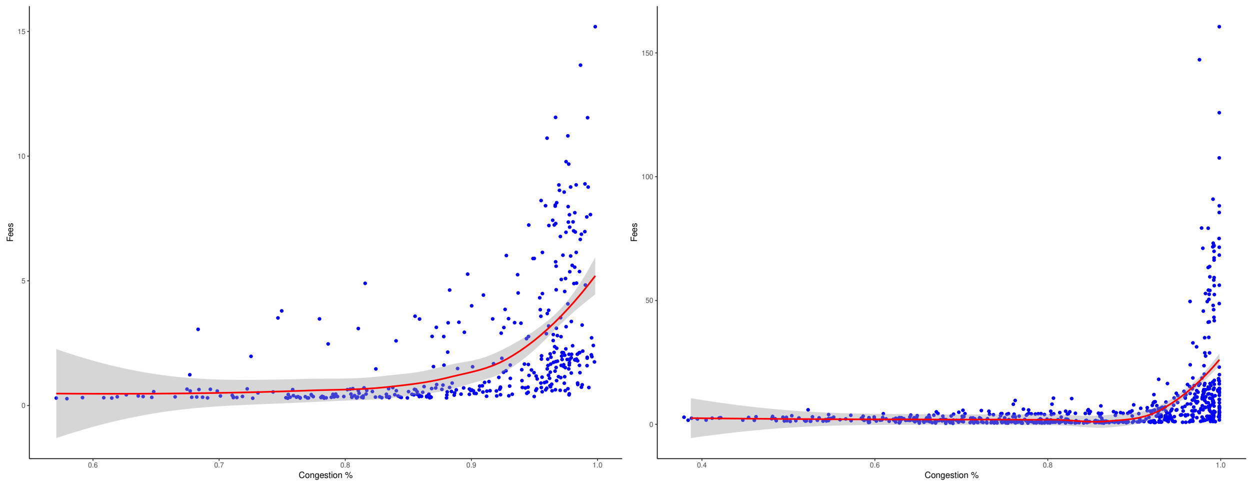
calculate

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

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 seem to be somewhat correlated with prices, especially during the speculative window. We include also a figure plotting fee rates against bitcoin prices. The blue, red and black dots indicate respectively data points before, during and after the speculative window. Figure 3b shows that the two variables are not only positively correlated, but that fees are more sensitive to price movements during the speculative period.

Finally, we also report the relationship between congestion and fees (see Figure 4). There is clearly a positive relationship as one would expect as congestion increases the scarcity of space in a block. This seems to be stable across the period before and after Segwit. Note that the scale for fees is different though, as prices are higher during the early adoption of Segwit. 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.



(a) Pre Segwit

(b) Post Segwit

Figure 4: Fee Rates and Congestion

3 Estimation

3.1 A Simple Model of Bitcoin Fees

We postulate that fees are driven by two considerations. First, the Bitcoin blockchain is a club good. Users need to post fees to have their transactions included into 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.

Volatility may matter for both, payments and speculation. If price changes a lot, it is important for payments to clear quickly so as 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 which can be described by the following linear regression model.

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

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. (2018) 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 HAC estimator for reporting standard errors on our regression results.[11] The results presented in Table 2 are using 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 chance to get 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 reestimate our benchmark specification with 2SLS, still adjusting our standard errors with a HAC estimator. Table 3 reports the results from the IV regression. We also include the results from a Durbin-Wu-Hausman test which calls for an IV approach to be warranted.

There are several interesting findings from the benchmark model. First, price has a significant positive impact on fee rates. The coefficient estimates on price are little affected by using an IV approach and by changing to a quadratic fit with respect to our congestion variable.

Second, fee rates are significantly and positively related to congestion. Including both a linear and a quadratic term introduces a non-linear relationship which seems implausible given the theoretical background for this relationship. Nothwithstanding, for sufficiently high congestion, an increase in congestion will lead to an increase in fee rates.

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

In what follows, we will rely on the specification with only a quadratic term for congestion and relegate results for the other specifications to the appendix. The fit of our model tends to increase when using only a quadratic term. Our estimates for the impact of changes in congestion on fee rates are also 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 two important features in the data. First, towards the end of 2017, the bitcoin price rose sharply, only to fall back to its old trend a few months afterwards. We interpret this phenomenon as a speculative period that added some additional demand for transactions to Bitcoin or shifted transactions away from payments to speculative ones.

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.

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

The dummy variable D_t takes a value of 1 during the speculative window 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 transaction 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 only report OLS results as a Durbin-Wu-Hausman test indicates that we cannot reject that these estimates are consistent. This is somewhat interesting, as it indicates that we can interpret the speculative period and the adoption of Segwit respectively as important demand and a supply shocks driving the fee rate during certain time periods.

The best fit is given by the models that include the interaction term of D_t and price with estimates being very close across the specifications. Prices influence fees significantly across all specifications. More importantly, during the speculative window this influence increases. Using the last column in the table, a $1,000 increase in the bitcoin price tends to raise fees by about $1.40 per kbyte stripped. This is a large increase in the costs of conducting a transaction in Bitcoin. During the time of speculation, this measure increased an additional $4.80 to $6.20, even though one has to take into account that fees – correcting for prices – were lower during this period (see the relatively large negative influence of the dummy variable D_t). 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 bitcoin quickly.

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 one percent increase in the usage of Segwit tends to decrease fees per kbyte stripped by about 17 cents. Of course, this needs to be balanced against the positive trend in the bitcoin price.

Congestion matters too when we take into account the speculative window and price developments during that time period. This confirms our intuition that higher congestion on average increases fees, not only because miners can select higher fee transactions, but also due to the fact that users submit larger fees to get their transactions confirmed in the blockchain.

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. Here we use only lagged values as instruments. In the appendix, we proceed to include other lagged variables 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.[12]

Our estimation results are very robust, in particular with respect to the movement of prices.

[12] Note that we cannot carry out this test in the full model using congestion as we cannot reject the hypothesis that OLS estimates are consistent.

Moreover, the impact of Segwit on fee rates remains significantly negative, even though it becomes smaller in size. Also, it is not surprising that our alternative congestion measure that uses both, the size of the mempool and the number of transactions becomes more significant. Interestingly, for this specification volatility becomes more significant during the speculative window. This indicates that users valued more rapid execution of their transactions during the time due to price movements.

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 some summary statistics that we will use to assess the impact on fees. All the data are taken for the period after the speculative window. The table shows that the usage of Segwit does not change depending 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 during time when blocks are full.

Table 6: Summary Statistics – Post Speculative Window

	Overall (431 days)	Full Blocks (54 days)
Fee Rate (USD/kbyte stripped)	2.71	6.61
Price (USD/BTC)	6,203	7,070
Segwit (%)	37.7	39.4
Congestion (%)	81.8	100
Avg. Transaction (per block)	1927	2499
Avg. Mempool (# of transactions)	8343	26,050
Avg. Stripped Size (KB)	759.05	913.4

Note: Daily Average from May 23, 2018 - July 28, 2019

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 blocksize of Bitcoin is restricted to 1MB and the adoption rate of Segwit is roughly constant at 40%. Hence, we can interpret changes in the level of congestion as shocks to the aggregate demand for Bitcoin transactions that shifts 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 blocksize increases. Second, congestion increases from about 80% to 100% which increases the fee rate per kbyte 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

$$\Delta \text{ Fee Revenue } = (2.71 + b \times \Delta^2 \text{ Congestion in \% }) \times 913 \text{ kbytes } - 2.71 \times 759 \text{ kbytes}$$

where $b = 5.84$ and $\Delta^2 = 1 - 0.818^2 = 0.331$. The total increase in fee revenue is thus estimated to be \$2,182, meaning that fee revenue per block would roughly double.

We could also rely on our alternative congestion measure that relates the number of transactions in a block to the size of the mempool. This measure of congestion is not censored. According to the summary statistics, during periods of full capacity, the ratio of transactions to mempool size decreases to about 0.096 from an average of 0.231. Using again our estimates from this alternative specification, the change in fee revenue is

$$\Delta \text{ Fee Revenue } = \left(2.71 + b \times \Delta \frac{\text{Transactions}}{\text{Mempool}}\right) \times 913 \text{ kbytes} - 2.71 \times 759 \text{ kbytes}$$

where $b = -10.17$. This gives us an increase in fees of \$1,671 which is about 25% lower than our original estimate above.[13]

It is interesting to view these rough estimates in terms of the next halving of Bitcoin block rewards. Around May 24th, 2020, the block reward will drop from 12.5 BTC to 6.25 BTC. To ensure the same security against double spending attacks, fees would have to pick up the slack created in the total rewards from mining a block.

Using the average price for Bitcoin after the speculative window, current block rewards are about \$80,000 on average with fees being only at the level of about 2.7% of the reward. Hence, it is highly unlikely that total rewards per block will stay constant, unless either the price of Bitcoin doubles or there is a very large increase in the usage of Bitcoin. Bitcoin throughput, however, is

[13] One could also use the size of the mempool to calculate the impact. The additional fee revenue from moving to a scenario of full blocks would be an increase in the fee rate of about \$1.77 USD/kbyte stripped yielding again a similar result.

already fairly close to full capacity. Unless users accept very long wait times to settle transactions – which seems unlikely to be the case – this implies that throughput capacity needs to increase with demand, albeit keeping congestion at a fairly high level.

4.2 Supply – The Impact of Segwit

There are two ways to increase throughput capacity further in Bitcoin.[14] The first one is a possible further adoption of the Segwit protocol by users, while the second one is a formal change in the blocksize of 1MB adopted by Bitcoin. We now view the demand function for Bitcoin as fixed on average and think of a change in the usage of Segwit as shifting the supply inducing a movement along this demand curve.

We regard Segwit adoption as a long-run decision by users that is not affected by demand fluctations. As Figure 2 shows, the adoption rate of Segwit has been stable over time after an initial introduction phase. A shift in 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 blocksize on the percent segwit being used, or

$$\frac{\text{Stripped Blocksize}_t}{\text{Blocksize}_t} = \rho_0 + \rho_1 \text{percent\_segwit}_t + \epsilon_t \tag{10}$$

using daily data for the time period after the introduction of Segwit. Our estimates for the coefficients are $\rho_0 = 0.99$ and $\rho_1 = -0.588$ (see 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 blocksize 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))[15].

This gives us a preliminary estimate for the elasticity of fee rates with respect to Segwit usage. Note that there are two effects – a direct effect of adoption (β_5) and an indirect one due to lower

[14] 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.

[15] With Segwit, some blocksizes have reached about 2.2MB in total, which translates into stripped size being about 45% of the actual (raw) size of a full block. Hence, our estimate seems to be consistent with this fact using a maximum stripped blocksize of 1MB.

congestion (β_1) which enters as a quadratic term. From equation (9), we have that a one percent increase in the usage of segwit decreases the fee rate by approximately

$$\beta_1[(x + 0.75\rho_1 \times 0.01)^2 - x^2] + \beta_5 \times 0.01 \approx (1.5\beta_1\rho_1 x + \beta_5) \times 0.01$$

per kbyte where we have neglected terms of a smaller magnitude. Note that this elasticity depends on the level of congestion x. At the average level of congestion given by $x = 0.818$, we have

$$(1.5 \times 5.84 \times (-0.588) \times 0.818 - 17.29) \times 0.01 = (-4.21 - 17.29) \times 0.01 = -0.23$$

cents per kbyte.

We can then use this estimate to see how fee revenue has changed due to the introduction of Segwit. Consider 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 $37.7 \times 0.441 = 16.63\%$ to about 98.43%. This corresponds to a fee revenue without Segwit of

$$(2.71 + 5.84(0.984^2 - 0.818^2) - 17.29(-0.377)) \times 984 \text{ kbytes } = 10.97 \times 984 \text{ kbytes } = \$10,780,$$

since stripped size equals blocksize without Segwit. Hence, according to our estimation, Segwit has led to a reduction of fee revenue by about 80%.

4.3 Optimal Blocksize

We now calculate an estimate for the maximum of fees that can be generated per block. Without full congestion, it is always optimal to decrease the blocksize in order to increase total fees raised per 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 blocksize would decrease the fee rate, but increase capacity. We now use our estimates to analyze this trade-off.

To do so, we 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

$$p(x) = (a + b) - bx \tag{11}$$

where x is the blocksize – now a design parameter – and p is the fee rate per kbytes stripped. The fee rate is equal to a whenever the design of the blocksize is $x = 1$ MB. Hence, the optimal blocksize maximizes total fee revenue $p(x)\alpha x$ and is given by

$$x^* = \frac{a+b}{2b} \tag{12}$$

where α is a constant that translates actual blocksizes into stripped blocksize for a fixed level of Segwit adoption (see equation (10)).

If we can pin down the two parameters – a and b – we obtain the optimal blocksize under the assumption that the blockchain remains at full capacity and that the demand curve is linear. For a, consider the summary statistics for full blocks from above and set $a = 6.61$.

For b, which is the elasticity of fee rates with respect to blocksize 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 blocksize equal to

$$\frac{d \text{ blocksize}}{d \text{ \% segwit}} = -\frac{1}{0.75\rho_1} = 2.27 \tag{13}$$

Since we measure blocksize in MB, this implies that 100% of Segwit would lead to an effective blocksize limit of 2.27MB. To obtain the parameter b, we use the estimated coefficient $\beta_5 = -17.29$. This yields[16]

$$b = 17.29 \times \left(-\frac{1}{0.75\rho_1}\right) = 39.21. \tag{14}$$

Finally, note that stripped size is a constant fraction of blocksize 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

$$x^* = \frac{6.61 + 39.21}{2 \times 39.21} \times 1 \text{ MB } = 0.584 \text{ MB}. \tag{15}$$

Hence, given current Segwit adoption, the current blocksize limit is about 40% too high. Since Segwit adoption is at 39.3% when blocks are full, the stripped blocksize is at

$$(1 - 0.588 \times 0.393) \times 0.584 \text{ MB } = 0.449 \text{ MB}$$

so that the estimated maximum fee revenue is given by \$10,291.

[16]Note that this implies that the intercept of the demand curve is at a fee rate of $a + b = 45.82$ \$/kbyte stripped.

To put this into perspective, this is about 1/8 of the fee revenue generated by total rewards with a block reward of 12.5 BTC. Consequently, relying on fees only and an optimally designed blocksize would generate rewards that are equally to a block reward of 1.6375 BTC. Interestingly, this is close to the optimal block reward simulated in Chiu and Koeppl (2017).

4.4 Segwit Adoption and Fee Revenue

While Segwit adoption increased throughput capacity, there seems to be strong evidence that it decreased overall fee revenue. To verify this, we again use a simple demand function that now links fee rates to actual blocksizes given by

$$p(x) = (a + 0.976b) - bx \tag{16}$$

where x is the actual blocksize. The parameter $a = 2.71$ is again taken from the summary statistics where the average blocksize is given by $0.760/(1 - 0.588 \times 0.377) = 0.976$ MB. As before, we set $b = 39.12$.

Inverting the demand function, we obtain for the actual block size given a fee rate $p(z)$,

$$x = \frac{40.98 - p(z)}{39.21}, \tag{17}$$

which yields a stripped blocksize of $x * (1 - 0.588z)$, where z is the level of Segwit adoption in %.

Finally, we can use our estimate for how Segwit adoption influences fee rates to find the optimal rate of adoption. Without Segwit adoption, the fee rate is given by \$10.97 per kbyte (see above). According to our estimates, increasing segwit by z%, the fee rate would be given by

$$p(z) = 10.97 - 17.29 \times 0.01 \times z + 5.84(0.984^2 - (0.984 - 0.01 \times z \times 0.441)^2). \tag{18}$$

Using this price function, it is easy to verify that the stripped blocksize is always decreasing in the level of Segwit adoption. Hence, not adopting Segwit at all would maximize fee revenue.

5 Conclusions

Our empirical estimates confirm that the introduction of the Segwit reduced congestion. However, our calculations also show that it lowered fees significantly. Indeed, given current prices and average demand for Bitcoin transactions, in order to maxmize fee revenue, it would be optimal to have no adoption of the Segwit protocol at all. In terms of long-run sustainability of the Bitcoin blockchain, Segwit was therefore a step in the wrong direction as it lowered fees significantly.

More importantly with revenue from block rewards falling over time to zero, fees will need to generate all the revenue. If Segwit adoption remains at 40%, 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 rewards. This once again casts some doubts on the sustainability of Bitcoin in the long-run, given that sufficiently high rewards are necessary to discourage double spending.

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. (1998), "Estimating and Testing Linear Models with Multiple Structural Changes", Econometrica, 66, pp. 47-78

Boehme, 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, 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

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/kbyte)	1,092	6.167	14.266	0.276	0.927	5.138	160.571
Price (USD/BTC)	1,092	5,128	3,753	552	1,332	7,457	19,325
Volatility	1,092	156	261	0	14	181	2,536
Segwit (%)	1,092	0.191	0.174	0	0	0.4	0
Block Weight (% of 4MB)	1,092	0.855	0.138	0.379	0.771	0.972	0.999
Transactions/Unconfirmed	1,092	0.374	0.324	0.011	0.105	0.573	1.000
Med. Confirmation Time	1,092	10.898	3.277	5.700	8.496	12.450	29.250
Mempool (# unconfirmed)	1,092	18,153.500	28,759.700	72.000	2,816.750	19,461.620	179,332.000

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 a 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.	Price	Volatility	
SE	2.08474	2.32693	0.00011	0.00158	
QS	10.09598	7.80758	0.00077	0.00710	
NW	10.15316	8.23057	0.00077	0.00652	
	(Intercept)	Cong.^{2}	Price	Volatility	
SE	1.24147	1.47362	0.00011	0.00158	
QS	7.01556	4.90835	0.00075	0.00700	
NW	7.04864	5.25952	0.00076	0.00655	
	(Intercept)	Cong.	Cong.^{2}	Price	Volatility
SE	9.23215	23.73730	15.07930	0.00011	0.00158
QS	9.80114	18.05808	12.62735	0.00073	0.00693
NW	12.50562	29.50712	20.49815	0.00075	0.00664

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

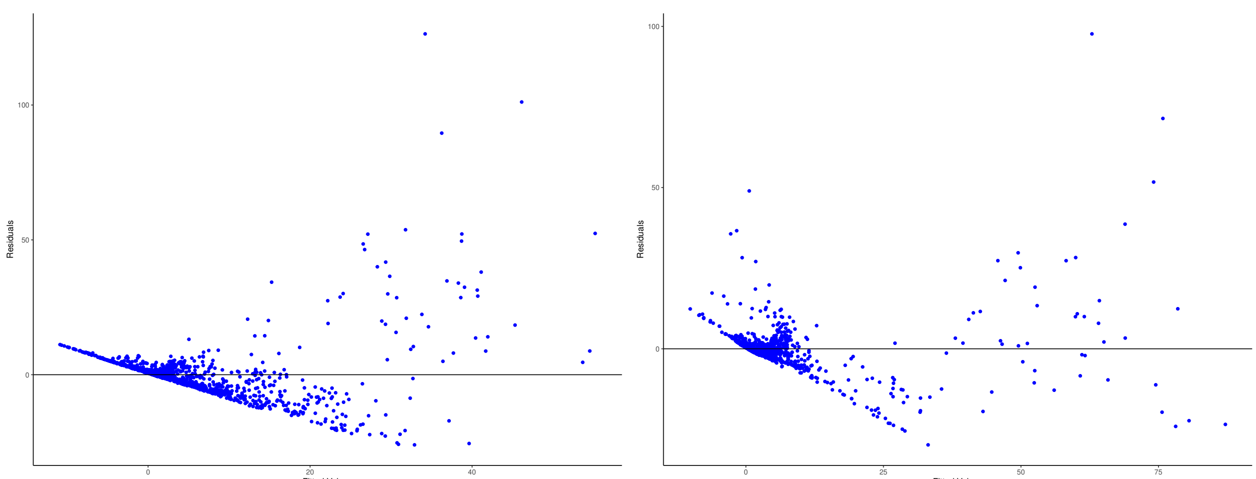


Figure 5: Heteroskedastic Errors

Linear and Linear-Quadratic Specifications

Table 9: Linear Model (OLS with HAC Estimator)

	Dependent variable: Fee Rate			
	(1)	(2)	(3)	(4)
Congestion	8.9556	7.6872	10.0037	8.2053
	(3.2552)	(4.9363)	(2.6406)	(4.5629)
	p = 0.0061***	p = 0.1197	p = 0.0002***	p = 0.0725*
Price	0.0033	0.0012	0.0034	0.0014
	(0.0009)	(0.0003)	(0.0010)	(0.0003)
	p = 0.0005***	p = 0.0003***	p = 0.0004***	p = 0.00002***
Volatility	0.0074	0.0068	−0.0106	0.0003
	(0.0047)	(0.0044)	(0.0050)	(0.0018)
	p = 0.1151	p = 0.1275	p = 0.0355**	p = 0.8864
Segwit (%)	−43.1572	−16.7863	−38.2461	−17.7782
	(13.4397)	(4.7081)	(10.6201)	(4.7896)
	p = 0.0014***	p = 0.0004***	p = 0.0004***	p = 0.0003***
D	−4.2637	−45.7899	−11.6484	−44.1003
	(4.3846)	(20.4115)	(5.4577)	(19.7320)
	p = 0.3311	p = 0.0251**	p = 0.0331**	p = 0.0257**
D × Price		0.0053		0.0048
		(0.0021)		(0.0020)
		p = 0.0100***		p = 0.0160**
D × Volatility			0.0294	0.0108
			(0.0102)	(0.0062)
			p = 0.0042***	p = 0.0828*
Constant	−10.3397	−5.8836	−10.9651	−6.5843
	(2.6751)	(4.4344)	(2.2599)	(4.1237)
	p = 0.0002***	p = 0.1849	p = 0.000002***	p = 0.1107
Observations	1,092	1,092	1,092	1,092
R^2	0.6160	0.7381	0.6657	0.7434
Adjusted R^2	0.6142	0.7366	0.6639	0.7417
Residual Std. Error	8.8610 (df = 1086)	7.3214 (df = 1085)	8.2706 (df = 1085)	7.2501 (df = 1084)
F Statistic	348.3664*** (df = 5; 1086)	509.5432*** (df = 6; 1085)	360.1590*** (df = 6; 1085)	448.5837*** (df = 7; 1084)

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

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

	Dependent variable: Fee Rate			
	(1)	(2)	(3)	(4)
Congestion	-41.3032	-80.0287	-55.6148	-81.2185
	(23.2993)	(20.6042)	(18.0737)	(19.2751)
	p = 0.0766*	p = 0.0002***	p = 0.0022***	p = 0.00003***
Congestion^2	32.1714	56.1301	42.0146	57.2309
	(14.6268)	(12.0198)	(11.4691)	(11.5295)
	p = 0.0281**	p = 0.000004***	p = 0.0003***	p = 0.000001***
Price	0.0032	0.0011	0.0033	0.0013
	(0.0009)	(0.0003)	(0.0009)	(0.0003)
	p = 0.0005***	p = 0.0008***	p = 0.0005***	p = 0.00003***
Volatility	0.0072	0.0065	-0.0111	-0.0002
	(0.0047)	(0.0044)	(0.0049)	(0.0017)
	p = 0.1271	p = 0.1449	p = 0.0254**	p = 0.9176
Segwit (%)	-42.7772	-15.5375	-37.6672	-16.5283
	(13.4978)	(4.4229)	(10.5631)	(4.5482)
	p = 0.0016***	p = 0.0005***	p = 0.0004***	p = 0.0003***
D	-4.3819	-46.9189	-11.9273	-45.2115
	(4.4303)	(19.7018)	(5.3808)	(19.1022)
	p = 0.3229	p = 0.0175**	p = 0.0269**	p = 0.0182**
D × Price		0.0055		0.0049
		(0.0020)		(0.0019)
		p = 0.0064***		p = 0.0108**
D × Volatility			0.0298	0.0110
			(0.0101)	(0.0063)
			p = 0.0033***	p = 0.0801*
Constant	8.6917	27.4198	13.8786	27.3557
	(9.4567)	(9.3659)	(7.0786)	(8.4992)
	p = 0.3583	p = 0.0035***	p = 0.0502*	p = 0.0014***
Observations	1,092	1,092	1,092	1,092
R^2	0.6181	0.7445	0.6694	0.7501
Adjusted R^2	0.6160	0.7429	0.6672	0.7482
Residual Std. Error	8.8404 (df = 1085)	7.2342 (df = 1084)	8.2294 (df = 1084)	7.1583 (df = 1083)
F Statistic	292.6701*** (df = 6; 1085)	451.2459*** (df = 7; 1084)	313.5049*** (df = 7; 1084)	406.2689*** (df = 8; 1083)

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

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 which is the average for the post-segwit data.

We can redo our analysis from Section 4 for the linear model. Increasing congestion to 100% yields an increase in fee revenue of about $1,915 which is slightly lower than in the quadratic model. The impact of Segwit on fee rates is 0.213 cents per percent increase in Segwit. Without Segwit, this gives an estimated fee revenue of $10,602.

Auxiliary Regression

Table 11: Impact of Segwit on Blocksize

	Dependent variable:
	$\frac{\text{Stripped Blocksize}}{\text{Blocksize}}$
Segwit (%)	−0.588***
	(0.009)
Constant	0.994***
	(0.003)
Observations	704
R^2	0.857
Adjusted R^2	0.857
Residual Std. Error	0.030 (df = 702)
F Statistic	4,204.773*** (df = 1; 702)
Note:	*p<0.1; **p<0.05; ***p<0.01

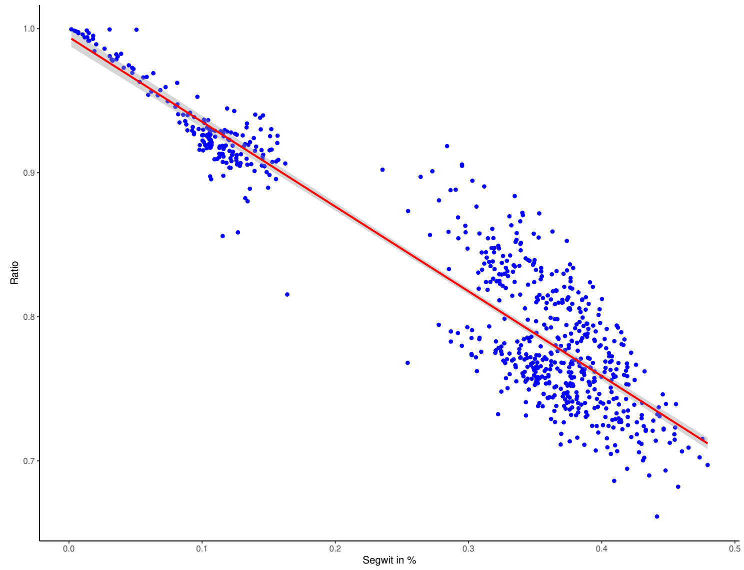


Figure 6: Impact of Segwit on Blocksize

Overidentified Models

The top line in Table 12 indicates the instruments used for each variant of the IV regression. We only report the specification with the largest p-value for the Hansen's J-test.[17] Our results show again that our estimation of the full model is fairly robust. Note that the coefficients on the congestion measure change a lot as we are looking at different units and scale relative to block weight as a percentage of 4MB. All other coefficients stay fairly the same. Recall that endogeneity of congestion was rejected for the benchmark model.

[17]We use a two step GMM procedure. Our results are robust to using an iterative method to estimate the GMM model.

Table 2: Benchmark Results (OLS with HAC Estimator)

	Dependent Variable: Fee Rate		
	(1)	(2)	(3)
Congestion	25.7238		−60.3322
	(7.8076)		(18.0581)
	p = 0.0011***		p = 0.0009***
Congestion^2		16.7854	54.9294
		(4.9083)	(12.6274)
		p = 0.0007***	p = 0.00002***
Price	0.0017	0.0017	0.0017
	(0.0008)	(0.0007)	(0.0007)
	p = 0.0259**	p = 0.0232**	p = 0.0229**
Volatility	0.0129	0.0127	0.0125
	(0.0071)	(0.0070)	(0.0069)
	p = 0.0703*	p = 0.0708*	p = 0.0719*
Constant	−26.6528	−17.1347	6.1183
	(10.0960)	(7.0156)	(9.8011)
	p = 0.0085***	p = 0.0148**	p = 0.5326
Observations	1,092	1,092	1,092
R^2	0.4774	0.4807	0.4837
Adjusted R^2	0.4760	0.4792	0.4818
Residual Std. Error	10.3270 (df = 1088)	10.2950 (df = 1088)	10.2692 (df = 1087)
F Statistic	331.3250*** (df = 3; 1088)	335.6467*** (df = 3; 1088)	254.6133*** (df = 4; 1087)

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

Table 3: Benchmark Results (IV with HAC estimator)

	Dependent variable: Fee Rate		
	(1)	(2)	(3)
Congestion	33.0725		−67.4353
	(9.9684)		(47.6770)
	p = 0.0010***		p = 0.1576
Congestion^2		21.0570	63.3233
		(6.2286)	(28.0524)
		p = 0.0008***	p = 0.0242**
Price	0.0018	0.0017	0.0017
	(0.0008)	(0.0007)	(0.0008)
	p = 0.0188**	p = 0.0190**	p = 0.0281**
Volatility	0.0118	0.0116	0.0115
	(0.0070)	(0.0070)	(0.0070)
	p = 0.0931*	p = 0.0959*	p = 0.0989*
Constant	−33.0117	−20.3770	5.8933
	(11.8136)	(7.9323)	(23.1873)
	p = 0.0053***	p = 0.0104**	p = 0.7995
Observations	1,092	1,092	1,092
R^2	0.4726	0.4766	0.4803
Adjusted R^2	0.4712	0.4752	0.4784
Residual Std. Error	10.3742 (df = 1088)	10.3346 (df = 1088)	10.3029 (df = 1087)
DWH-Test (p-value)	0.00025***	0.00036***	0.00243***

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

Table 4: Full Model – Quadratic Specification

	Dependent variable: Fee Rate			
	(1)	(2)	(3)	(4)
Congestion^2	6.0312	5.4924	6.8067	5.8353
	(2.0588)	(3.0652)	(1.6657)	(2.8398)
	$p = 0.0035^{***}$	$p = 0.0735^{*}$	$p = 0.00005^{***}$	$p = 0.0402^{**}$
Price	0.0033	0.0012	0.0034	0.0014
	(0.0009)	(0.0003)	(0.0010)	(0.0003)
	$p = 0.0005^{***}$	$p = 0.0003^{***}$	$p = 0.0005^{***}$	$p = 0.00002^{***}$
Volatility	0.0073	0.0067	−0.0107	0.0001
	(0.0047)	(0.0044)	(0.0050)	(0.0017)
	$p = 0.1192$	$p = 0.1339$	$p = 0.0324^{**}$	$p = 0.9579$
Segwit (%)	−42.8886	−16.3037	−37.8828	−17.2916
	(13.4254)	(4.6133)	(10.6015)	(4.7048)
	$p = 0.0015^{***}$	$p = 0.0005^{***}$	$p = 0.0004^{***}$	$p = 0.0003^{***}$
D	−4.2410	−45.7858	−11.6407	−44.0863
	(4.3844)	(20.2427)	(5.4313)	(19.5631)
	$p = 0.3337$	$p = 0.0240^{**}$	$p = 0.0324^{**}$	$p = 0.0245^{**}$
D × Price		0.0053		0.0048
		(0.0020)		(0.0020)
		$p = 0.0093^{***}$		$p = 0.0150^{**}$
D × Volatility			0.0295	0.0108
			(0.0102)	(0.0062)
			$p = 0.0040^{***}$	$p = 0.0813^{*}$
Constant	−7.1800	−3.4111	−7.4886	−3.9261
	(1.7496)	(2.4932)	(1.5606)	(2.3378)
	$p = 0.00005^{***}$	$p = 0.1716$	$p = 0.000002^{***}$	$p = 0.0934^{*}$
Observations	1,092	1,092	1,092	1,092
R^2	0.6167	0.7392	0.6668	0.7446
Adjusted R^2	0.6149	0.7378	0.6649	0.7430
Residual Std. Error	8.8528 (df = 1086)	7.3054 (df = 1085)	8.2576 (df = 1085)	7.2327 (df = 1084)
F Statistic	349.4129^{***} (df = 5; 1086)	512.5582^{***} (df = 6; 1085)	361.8652^{***} (df = 6; 1085)	451.4979^{***} (df = 7; 1084)

Note: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Table 5: Robustness – Different Measures for Congestion

	Dependent variable: Fee Rate			
	OLS	IV 2SLS		
	(1)	(2)	(3)	(4)
Congestion^2	5.8353			
	(2.8398)			
	p = 0.0402**			
Mempool		0.0001		
		(0.0001)		
		p = 0.0812*		
Trans./Unconf.			−10.1668	
			(4.0638)	
			p = 0.0126**	
Med. Conf. Time				1.1189
				(0.5401)
				p = 0.0386**
Price	0.0014	0.0013	0.0013	0.0014
	(0.0003)	(0.0003)	(0.0002)	(0.0002)
	p = 0.00002***	p = 0.000001***	p = 0.0000***	p = 0.0000***
Volatility	0.0001	−0.0007	−0.0010	−0.0003
	(0.0017)	(0.0016)	(0.0018)	(0.0015)
	p = 0.9579	p = 0.6781	p = 0.5864	p = 0.8655
Segwit (%)	−17.2916	−12.7592	−9.8339	−12.0409
	(4.7048)	(4.1356)	(3.0298)	(3.9397)
	p = 0.0003***	p = 0.0021***	p = 0.0013***	p = 0.0023***
D	−44.0863	−38.9132	−43.1959	−47.9050
	(19.5631)	(15.0957)	(18.4788)	(15.1568)
	p = 0.0245**	p = 0.0101**	p = 0.0196**	p = 0.0017***
D × Price	0.0048	0.0042	0.0047	0.0052
	(0.0020)	(0.0016)	(0.0019)	(0.0016)
	p = 0.0150**	p = 0.0074***	p = 0.0107**	p = 0.0011***
D × Volatility	0.0108	0.0097	0.0118	0.0109
	(0.0062)	(0.0066)	(0.0060)	(0.0059)
	p = 0.0813*	p = 0.1414	p = 0.0493**	p = 0.0659*
Constant	−3.9261	−1.7862	3.6816	−12.5711
	(2.3378)	(1.1447)	(1.1587)	(6.3976)
	p = 0.0934*	p = 0.1190	p = 0.0016***	p = 0.0497**
Observations	1,092	1,092	1,092	1,092
R^2	0.7446	0.7756	0.7290	0.7268
Adjusted R^2	0.7430	0.7742	0.7273	0.7250
Residual Std. Error (df = 1084)	7.2327	6.7793	7.4500	7.4810
DWH-Test (p-value)	0.206	0.0073***	0.00315***	0.00245***

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

Table 12: Overidentified IV Regressions

	Dependent variable: Fee Rate		
	IV Regression		
	(Pool, Cong., Time)	(Trans/Unc., Time)	(Time, Cong.)
	(1)	(2)	(3)
Mempool	0.0001**		
	(0.0001)		
Trans./Unconf.		−9.6990**	
		(3.8612)	
Med. Conf. Time			1.1035**
			(0.4614)
Price	0.0013***	0.0012***	0.0014***
	(0.0002)	(0.0002)	(0.0002)
Volatility	−0.0005	−0.0007	−0.0002
	(0.0015)	(0.0017)	(0.0014)
Segwit (%)	−12.6444***	−9.5346***	−12.1699***
	(3.0377)	(2.9931)	(3.1152)
D	−40.8620***	−38.7389**	−47.5253***
	(14.1032)	(17.0833)	(13.4261)
D × Price	0.0044***	0.0043**	0.0051***
	(0.0015)	(0.0017)	(0.0014)
D × Volatility	0.0092*	0.0082**	0.0107**
	(0.0055)	(0.0040)	(0.0050)
Constant	−1.6904*	3.5449***	−12.3917**
	(0.9726)	(1.1027)	(5.5070)
J-Test	0.2709	0.5677	0.0029
(p-value)	(0.8733)	(0.4519)	(0.9569)

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