A Disquieting Lack of Evidence for Disintermediation in a Home-Cleaning Platform

Ekaterina Astashkina
Ross School of Business, University of Michigan, Ann Arbor, MI 48109, United States, astash@umich.edu

Robert L. Bray
Kellogg School of Management, Northwestern University, Evanston, IL 60208, United States, r-bray@kellogg.northwestern.edu

Ruslan Momot
Ross School of Business, University of Michigan, Ann Arbor, MI 48109, United States, momot@umich.edu

Marat Salikhov
New Economic School and Moscow School of Management Skolkovo, Moscow, 121353, Russia, msalikhov@nes.ru

August 28, 2023

We study a sample of data from an online platform that matches home cleaners with people who want their homes cleaned. The dataset has a key feature: it reports with high frequency the geographic distances between the cleaners and their appointed residences during both the cleaners' working hours and off hours.For example, we observe 57 and 46 distance snapshots a day for the mean and median cleaner, respectively. We use these distance measurements to test whether the cleaners disintermediate the work, returning to the residences to perform some undisclosed cleanings for which they will not have to pay the platform middleman its cut. We find no evidence of such disintermediation—in fact, we find strong evidence to the contrary. Specifically, we will argue that no more than one in 83 relationships formed on the app will end in disintermediation.

Key words: platform disintermediation, platform leakage, matching markets

1. Introduction

We study platform disintermediation, by which workers and patrons conspire to transact offline to cut out a middleman app. Received wisdom suggests that disintermediation can significantly hurt a platform's bottom line (Ladd 2022) and even lead to its complete failure (*TechCrunch* 2015). Disintermediation also contributes to the informal economy, which can hamper development (see a recent report by the *International Monetary Fund*, Deléchat and Medina 2020). Due to its secretive nature, disintermediation has proven difficult (if not impossible) to estimate. However, we have a new dataset that enables us to *directly measure* the phenomenon in a home-cleaning platform – a part of a rapidly-growing $657B market of home services (*Angi Research & Economics*, 2022). Our dataset records the geographic distances between 5,391 cleaners and the 95,633 residences they clean. The times between successive cleaner-residence distance readings are generally much shorter than the cleaning times: the median time between readings is 9 minutes, whereas the median cleaning time is 215 minutes. We use these data to determine whether cleaners return to the residences for some illicit cleanings, unbeknownst to the platform. We find no such disintermediated cleanings. In fact, we have fewer proximate distance measurements than one would expect by sheer chance alone.

At first this finding surprised us, as our empirical setting seems ripe for disintermediation. First, as a local monopolist, the platform collects a sizeable commission on every job (either 20%, 30%, or 40%, depending on the cleaner's experience). The platform also operates in a post-Soviet state whose shadow economy makes up more than 40% of the country's GDP and in which there is no cultural precedent for hiring unfamiliar cleaners through centralized agencies. Further, the workers and employers of this platform generally meet in person, and can thus easily arrange alternative communication channels. Finally, subsequent appointments can be scheduled with ease, as the service is demanded at regular intervals, with little time sensitivity (unlike in the ride-sharing case, in which the service is demanded at a particular time, with little notice). However, ex post, we identified a set of the potential reasons for the observed lack of disintermediation:

- it shouldn't be too hard to find a cleaner in this city, as labor is relatively plentiful, so those who want an under-the-table cleaning arrangement probably wouldn't have to resort to downloading an app and disintermediating off of it,
- the security afforded by the platform may be particularly valued in this context, as the job entails going into strangers' apartments and handling other people's personal effects,
- due to the high level of income inequality, the cost of cleanings is relatively inconsequential to those affluent enough to hire cleaners on an app (the average cleaning costs 31.58 USD), and
- for whatever reason, relationships between the platform's cleaners and residences rarely form: 90% of the cleaner–residence pairs in our sample correspond to just one cleaning, 5.8% correspond to two, 1.7% correspond to three, and only 0.50% correspond to ten or more cleanings.

2. Literature Review

Recent theory work shows that disintermediation activity can undermine the functioning of a marketplace and lower profits. The papers in this stream expose the mechanisms behind why the undesired disintermediation behavior occurs in the first place and explore the potential remedies via a platform redesign. In particular, Sekar and Siddiq 2023 focus on the information design of a platform and study the combined use of information and pricing strategies that deter disintermediation. Hagiu and Wright 2022 study how the alteration of information, incentive, and competitive environments impacts platform's revenues in the presence of platform leakage. Chaves 2018 explore how a more nuanced pricing and matching policies can alleviate disintermediation. Finally, He et al. 2023 investigate how curating the waiting processes, investing into workers' upskilling, and modifying information environments can deter disintermediation.

From the empirical side, Gu and Zhu (2021) provided the most direct evidence of platform disintermediation. They used a dictionary of disintermediation phrases—such as "my phone number," "avoid fees," "save 10%," and "Venmo"—to assign a "disintermediation score" to the chats between customers and freelancers in an online freelance marketplace. They then ran a field experiment that showed an additional freelancer satisfaction score to a set of customers. They found that this intervention increased the disintermediation scores of the chats between the freelancers with high satisfaction scores and the customers that could observe these high scores. Gu and Zhu (2021, p. 794) concluded that

> "enhanced trust increases the likelihood of high-quality freelancers being hired. However, when the trust level is sufficiently high, it also increases disintermediation, which offsets the revenue gains from the increase in hiring high-quality freelancers."

Gu (2022) also used the change in chat log disintermediation scores as a dependent variable. However, rather than conduct a field experiment, Gu used a natural experiment: China's ban of Skype in 2017. Gu (2022, p. 19) found that

> "after Skype is blocked, assignments between mainland China freelancers and US clients become 18.7% less in detected disintermediation scores, 33.6% higher total charges, and 26.5% longer hours relative to assignments with matched freelancers in Hong Kong, Taiwan, Macau, Singapore, and Malaysia."

Karacaoglu et al. (2022) used the observational data from a European home-cleaning platform to test for disintermediation, by estimating whether cleaners take customers with them when they exit the platform. The authors concluded that they do, as, while around 66% of customers leave the platform after a typical cleaning, 74% of customers leave after a cleaner's last cleaning. Accordingly, they estimated that "the platform would enjoy about 24 percent more cleanings if not for disintermediation." Unfortunately, the authors cited several alternative mechanisms that could correlate cleaner and customer departures. And while they did an admirable job ruling out these alternatives, the authors nevertheless arrived at disintermediation more by a process of elimination than by direct observation.

Lin et al. (2022) combined the Airbnb listings in Austin, Texas, with a "novel individual mobile location dataset" to show that around 5% of units listed as not available on the app are occupied at some time between 1:00 a.m. and 5:00 a.m. The authors interpreted these off-platform occupancies as disintermediated, but that would be the case only if the occupier found the unit via Airbnb, which is not necessarily the case.

He et al. (2020) showed that third-party sellers transact less on a Chinese e-commerce site after the site starts selling competing goods. The authors explained that "the substantial decrease in the demand of treated stores lends support to the existence of the competition/disintermediation effect," but they could not definitively establish that disintermediation is the mechanism, since they did not observe the third-party sellers' offline business.

Farronato et al. (2022, p. 30) mentioned disintermediation as "another potential explanation for increased DogVacay attrition" after this dog-sitting platform was acquired by the competing dog-sitting platform Rover. However, rather than something they prove, they framed disintermediation as something they failed to disprove: since "disintermediation is hard to measure, we cannot conclusively rule it out."

3. Setting

The platform operates in a large city of a post-Soviet state. An underground subway system is the predominant mode of transportation. The subway has exceptional coverage, and is reliable and fast. Residential areas are dense—thousands of people per square kilometer—so nearly all jobs correspond to apartment units. Most of these units have the floor area of under 1000 square feet (or approximately 100 square meters), thus the majority of cleanings are performed by a single cleaner.

The platform recruits cleaners in the capacity of independent contractors; trains them with educational videos, master classes, and a trial cleaning of a mock residence; and then outfits them with cleaning supplies, equipment, and a uniform. Recruitment happens on a rolling basis, and its intensity fluctuates with the platform's future demand projections. The platform asks the cleaners to log their availability over the next seven days in the app. The cleaners denote each day as either a "work day" or a "day off" (denoting a day as "part time available" is not allowed). But these designations are for forecasting purposes only, and the workers incur no penalty for last-minute changes.

After receiving the tentative availability of its workforce, the platform opens up a conservative number of cleaning slots. A customer can reserve one of these slots by visiting the platform's website, or by calling the platform's call center. Most cleanings are scheduled several days in advance, as it's difficult to find an available cleaning slot on short notice. When signing up for a cleaning, the customer provides the address of the residence, the nature of the job, and the payment information (if a customer is already registered with the platform, this information is provided through an app). Once they have successfully secured a slot, the platform guarantees that it will be honored, or the next cleaning is free.

The cleaners sign up for the patron's jobs with the platform's mobile app. Before claiming a job, the cleaners see its remuneration, its approximate location, the number of rooms and bathrooms it entails, and a detailed list of potential service upcharges, such as the cleaning of "tableware" or "winter windows." The platform neither curates the list of jobs the cleaner chooses from nor does it employ any matching algorithm. The general rule is: any cleaner can fill *any* of the open positions. The only two (rare) exceptions to this rule are: (i) the patron specifically requested to have or not have a particular cleaner, or (ii) the customer is a "VIP client"—e.g., a celebrity or a customer who reported bad service in the past—in which case only the most highly rated cleaners see their postings.

The day before a job, the customer receives a text message with the cleaner's photo, name, average rating, and number of submitted ratings, and two to three hours before a job, the cleaner receives the exact address of the residence. The platform's call center mediates all official communications between cleaners and customers.

The cleaners log the start and end time of each cleaning on the platform's app. Since they work by themselves, the cleaners usually require several hours to finish a job: the cleaning duration quantiles are 2.6, 3.6, and 4.7 hours. Cleaners rarely clean more than two residences per day.

Customers rate each cleaning on a 5-point scale, and are encouraged to leave a private comment for the platform. The platform will send over another cleaner, free of charge, if a customer was unhappy with a cleaning. And the platform reimburses customers who claim that their cleaner broke or stole their property, or arrived more than 15 minutes late.

To the best of our knowledge, the platform did not actively dissuade disintermediation during our period of study. However, *before* signing up with the platform, the cleaners *must agree* to allow the mobile app to track their locations (this requirement is, thus, a part of their contract).

4. Data

Before we describe the dataset, we would like to first detail its provenance, to highlight the fact that *our analysis was performed independently of the platform.* In 2017, Ekaterina received the data in a no-strings-attached manner from the platform's chief data officer—the company was quite casual with its data back then, and it didn't mind sharing them with Western academics. Soon thereafter Ekaterina, Ruslan, and Marat used the sample to write a research paper on a different topic. They did not consider disintermediation and did not study the geographic distances that are central to our current investigation. Ekaterina sporadically corresponded with the platform's chief data officer as these three authors wrote their article, but that correspondence ended in the summer of 2018, when they finished their project. The sample then sat dormant on the first author's computer between 2018 and 2021, at which point Robert caught wind of it from Ruslan. As soon as Robert heard about the data's geographic distances, they knew that these distances could be the key to estimating the degree of disintermediation, which was how this project began.

So, as you see, we performed this work's analysis independent of the platform. Indeed, the idea to study disintermediation came three years after the last correspondence with the firm. None of the authors of this article have been meaningfully associated with the company, in an official or unofficial capacity, and the data we use are anonymized and at least five years old.

The data come in two raw tables: `Jobs` and `Distances`. The former table characterizes 320,383 cleanings that took place between 2014-09-06, and 2017-08-10. It comprises 95,633 residences, 5,391 cleaners, and 254,330 cleaner–residence pairs—each such pair may denote *multiple* interactions between a certain cleaner and a patron across *multiple* dates. The average cleaner cleans 254,330/5,391 = 47 distinct residences and the average residence is cleaned by 254,330/95,633 = 2.7 distinct cleaners. The time between successive cleanings of a given residence has a median of 15 days, and an interdecile range of 7 and 69 days,— with the majority of cleanings performed on a regular basis.

The `Jobs` table describes the cleanings in exquisite detail. For example, we observe the number of bedrooms and bathrooms (1.7 and 1.0, on average), whether any ironing needs doing (11% of jobs), whether the microwave, oven, or refrigerator needs cleaning (3.9%, 5.6%, and 6.3% of jobs), whether the kitchen cabinets or balcony windows need wiping down (3.4% and 1.4% of jobs), whether the cleaners need to bring their own vacuums (22% of jobs), and whether the cat litter needs changing (0.23% of jobs). The table comprises 83 such variables, but we will use only the cleaner IDs, the residence IDs, and the time of the cleanings—i.e., who worked where when.

The `Distances` table reports the geographic distances between each cleaner and the dwellings they clean, roughly every half hour the app is open on their phone (which includes the app running as the background process,[1] see the detailed discussion on page 9), between 2015-09-07 and 2016-06-23. More specifically, the table comprises a collection of snapshots that capture at a particular time the distance between a given cleaner and all the dwellings they have ever cleaned or will ever clean. For example, a representative distance snapshot indicates that at 04:30:03 on 2016-04-02, cleaner 14592 was 0.106, 2.05, 2.17, 5.30, 6.07, 6.68, 6.82, 11.9, 14.2, 14.6, 18.0, and 25.0 km from residences 39215, 40124, 10986, 6080, 35402, 22177, 24612, 13027, 31629, 32096, 15966, and 43196, which is all the units this cleaner is associated with. We have 4,323,897 such distance snapshots, across 76,112 cleaner–date pairs, so that a cleaner's distance profile is recorded 4,323,897/76,112 = 57 times a day, on average. For example, when we arrange the cleaner–date pairs by the number of snapshots and then sort by cleaner ID and date to break ties, we get cleaner 13367 on 2016-04-19 in the median case, for whom we have distance snapshots at the following times:

[1] See *Washington Post* 2019 as well as developer's documentation for Android: https://developer.android.com/training/location/background and for iOS: https://developer.apple.com/documentation/corelocation/handling_location_updates_in_the_background.

05:04, 05:49, 07:27, 08:42, 08:56, 09:10,
09:42, 09:53, 09:57, 09:59, 10:06, 10:08,
10:09, 10:11, 10:12, 10:13, 10:15, 10:17,
10:21, 10:23, 10:24, 10:25, 10:26, 10:28,
10:29, 10:39, 10:40, 10:40, 10:45, 13:08,
13:19, 13:34, 14:02, 14:22, 14:42, 17:55,
19:20, 19:46, 20:14, 20:44, 21:15, 21:45,
22:15, 22:39, 23:17, 23:48.

Each of these snapshots records the distance to each of the 142 addresses this cleaner is associated with.

We compress `Distances` down to a table called `Shortest_Distances`, which comprises a collection of (cleaner, residence, date, `shortest_distance`) quadruples, the last variable of which specifies how close the given cleaner got to the given dwelling on the given day, when they were at their nearest. Case in point: `Distances` has `distance` = 77.0, 77.2, 77.0, 76.9, 76.8, 77.0, 77.0, 76.9, and 15.9 km at `time` = 01:36, 01:37, 05:29, 05:31, 05:48, 08:53, 08:54, 09:10, and 15:58 for (cleaner, residence, date) = (11452, 22177, 2016-04-21), and thus `Shortest_Distances` has `shortest_distance` = min(77.0, 77.2, 77.0, 76.9, 76.8, 77.0, 77.0, 76.9, 15.9) = 15.9 km for (cleaner, residence, date) = (11452, 22177, 2016-04-21).

Next, we define a (cleaner, residence) pair's *active period* as the days between 69 days before its first cleaning and its last cleaning, and we define this pair's *post period* as the days after its last cleaning and before the end of the `Jobs` sample.[2] For example, cleaner 71747 first cleaned residence 71043 on 2016-12-11 and last cleaned it on 2017-03-18, so this (cleaner, residence) pair's active period spans from 2016-12-11 - 69 days = 2016-10-03 to 2017-03-18, and its post period spans from 2017-03-19 to 2017-08-10 (which is the last date in the `Jobs` sample). To limit the sample to the working relationships that most likely ended on our watch, we then remove from `Shortest_Distances` the (cleaner, residence) pairs whose post periods last less than 69 days—i.e., whose final cleanings occurred within 69 of the end of the `Jobs` sample. We call the resulting panel `Shortest_Distances_Filtered`.

Finally, we construct four samples from `Shortest_Distances_Filtered` and `Jobs`:

- the `Unobserved` sample comprises the `Shortest_Distances_Filtered` (cleaner, residence, date, `shortest_distance`) quadruples that correspond to the (cleaner, residence) pair's post period,
- the `Observed` sample comprises the `Shortest_Distances_Filtered` (cleaner, residence, date, `shortest_distance`) quadruples that correspond to the (cleaner, residence) pair's active period,

[2] We start the active period 69 days before the first job because that is the *90th* percentile of the time between a residence's successive cleanings. We don't start the active period at the first job because in this case the time range would both start and finish with a job, which would bias upwards our work rate estimate.

- the `Working` sample comprises the `Observed` observations that have a corresponding `Jobs` work order, and
- the `Non_Working` sample comprises the `Observed` observations that do not have a corresponding `Jobs` work order.

For example, because cleaner 71747 last worked at residence 71043 on 2017-03-18, the `Observed` and `Working` samples both include an observation with (cleaner, residence, date) = (71747, 71043, 2017-03-18), the `Observed` and `Non_Working` samples both include an observation with (cleaner, residence, date) = (71747, 71043, 2017-03-17), and the `Unobserved` sample includes an observation with (cleaner, residence, date) = (71747, 71043, 2017-03-19).

The following identifying assumption will help us interpret these samples.

ASSUMPTION 1. *Disintermediation is permanent: a cleaner and employer will not transact with one another on the platform after transacting under the table.*[3] *Accordingly, the* `Working` *and* `Non_Working` *samples will not contain any disintermediated cleanings.*

This assumption implies that we observe the full work schedule of a given cleaner at a given residence before their last official work day. In other words, it implies that we observe both when the worker works and when they don't during the active period. But during the post period, after the official relationship has ended, we don't know whether or not the worker worked because by this time they could have disintermediated. Accordingly, the `Unobserved` panel consists of the observations in which the given cleaner may or may not have worked at the given residence on the given day, the `Non_Working` panel consists of the observations in which the given cleaner definitely did not work at the given residence on the given day, the `Working` panel consists of the observations in which the given cleaner definitely did work at the given residence on the given day, and the `Observed` panel consists of the union of the `Non_Working` and `Working` panels.

Assumption 1 is reasonable because there's little incentive for a specific cleaner-customer pair to return to the app after establishing enough trust to disintermediate. Note, this assumption applies only to *specific* working relationships: it prevents a *given* cleaner from interacting with a *given* customer on the app, *after* they have agreed to disintermediate, but it *does not* prevent the cleaner from interacting with *other* customers on the platform, nor the customer from interacting with *other* cleaners. In other words, assumption 1 maintains that disintermediation spirits away customer-cleaner pairs, but *not* the customers or cleaners themselves.

Assumption 1 enables us to clearly delineate between working and non-working days. But we don't need this assumption to see that working days differ from the non-working days, because the `Working` panel has significantly smaller `shortest_distance` readings than the `Non_Working` panel,

[3] However, they may transact with other employers and cleaners, correspondingly.

as figure 1 illustrates. For example, 74% and 88% of the `Working` sample's `shortest_distance` values are less than 100 and 500 meters, but only 0.33% and 2.9% of the `Non_Working` sample's `shortest_distance` values are less than 100 and 500 meters. And, on the flip side, only 6.9% and 4.6% of the `Working` sample's `shortest_distance` values exceed 5 and 10 km, whereas 62% and 36% of the `Non_Working` sample's `shortest_distance` values exceed 5 and 10 km. These results indicate that the geographic distances are recorded with high enough frequency to catch the cleaners in the act of cleaning—i.e., to find them near the designated premises at the appointed time. And this fact isn't surprising, as the consecutive `Distance` snapshots are generally much shorter than the cleaning times. For example, the cleaning durations have deciles 0.62, 2.3, 2.8, 3.2, 3.6, 4.0, 4.4, 5.1, and 6.1 hours, whereas the durations between successive distance snapshots have deciles 0.01, 0.02, 0.03, 0.07, 0.15, 0.18, 0.3, 0.5, and 0.61 hours. Overall, these results justify our empirical strategy: to use the `Unobserved` sample's `shortest_distance` values to classify its observations as either "working" or "non-working."

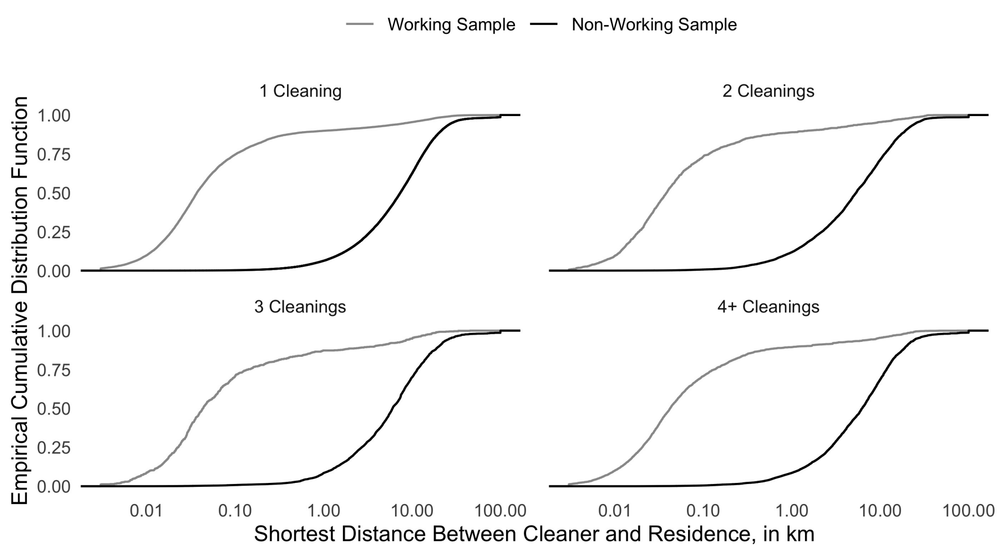

Figure 1 This figure compares the `Working` and `Non_Working` `shortest_distance` empirical distributions, by the total number of times that a given cleaner cleaned a given residence. For example, we limit the samples to the (cleaner, residence) pairs with only one `Jobs` record to create the top-left graph. Plotting by day of the week, rather than by number of jobs, yields similar results. Note that there's little overlap between the `Working` and `Non_Working` distance distributions.

To facilitate this classification task, we coarsen our `shortest_distance` values to the 200-point grid $D = \{25, 50, \cdots, 5000\}$ by rounding up all distances to the nearest multiple of 25 m, and then truncating at 5,000 m. We call the resulting variable `coarsened_shortest_distance`. This will be our primary variable of interest.

Finally, before getting to our analysis, let us highlight one additional feature of our data: *the app tends to remain open on the phones of the cleaners (incl. running in the background process), even when they are not working* as we can see in figure 2. We will gauge the length of time that the app stays open after the completion of a job with a variable called `days_idle`, which we define as the number of days since the cleaner last cleaned a residence. For example, if a given cleaner worked on Monday, Thursday, and Saturday of a given week then the Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday `days_idle` values for this cleaner-week would be 0, 1, 2, 0, 1, and 0, respectively. Figure 2 provides a separate plot for each `days_idle` value between 0 and 20, which generously covers the domain of this variable, which has a 99th percentile of 16 days, a 95th percentile of 5 days, and a median of zero days (most cleaner-days have a corresponding work order). As you see, for most cleaners the app is running more or less indefinitely, as evidenced by the geographic recordings we are able to collect, through up to 20 days of idleness. For example, the fraction of days with at least one distance snapshot—and hence the fraction for which the app is certainly running—is

$$\begin{array}{ll} 97.9\% & \text{for } \texttt{days\_idle} = 0, \\ 91.5\% & \text{for } \texttt{days\_idle} = 1, \\ 89.6\% & \text{for } \texttt{days\_idle} = 2, \\ 87.6\% & \text{for } \texttt{days\_idle} = 4, \\ 84.7\% & \text{for } \texttt{days\_idle} = 8, \quad \text{and} \\ 84.2\% & \text{for } \texttt{days\_idle} = 16. \end{array}$$

Granted, the decreasing trend from 97.9% with 0 `days_idle` to 84.2% with 16 `days_idle` indicates that people do exit out of the app when they aren't working, but only very gradually. For example, since `days_idle` is only 1.33, on average, 94% of the days in our sample have at least one distance snapshot and the average number of consecutive days with at least one geographic reading is 17.7. The fact that cleaners tend to leave the app running for many days on end suggests that the cleaners' phones will record a decent portion of the disintermediated cleanings.

5. Analysis

We will start with our most important exhibit: figure 3, which plots the distribution of `coarsened_shortest_distance` from our four primary samples. This figure illustrates several points. First, the `Observed` distribution is a linear combination of the `Working` and `Non_Working` distributions, which is understandable because the `Observed` sample is a combination of the `Working` and `Non_Working` samples. Second, the `Working` distances are generally much smaller than the `Non_Working` distances, as we previously saw in figure 1. Specifically, the `Working` distribution shoots upwards near zero km, whereas the `Non_Working` plunges downwards. The upward spike makes sense because most cleanings are detected, and the downward dip makes sense because the

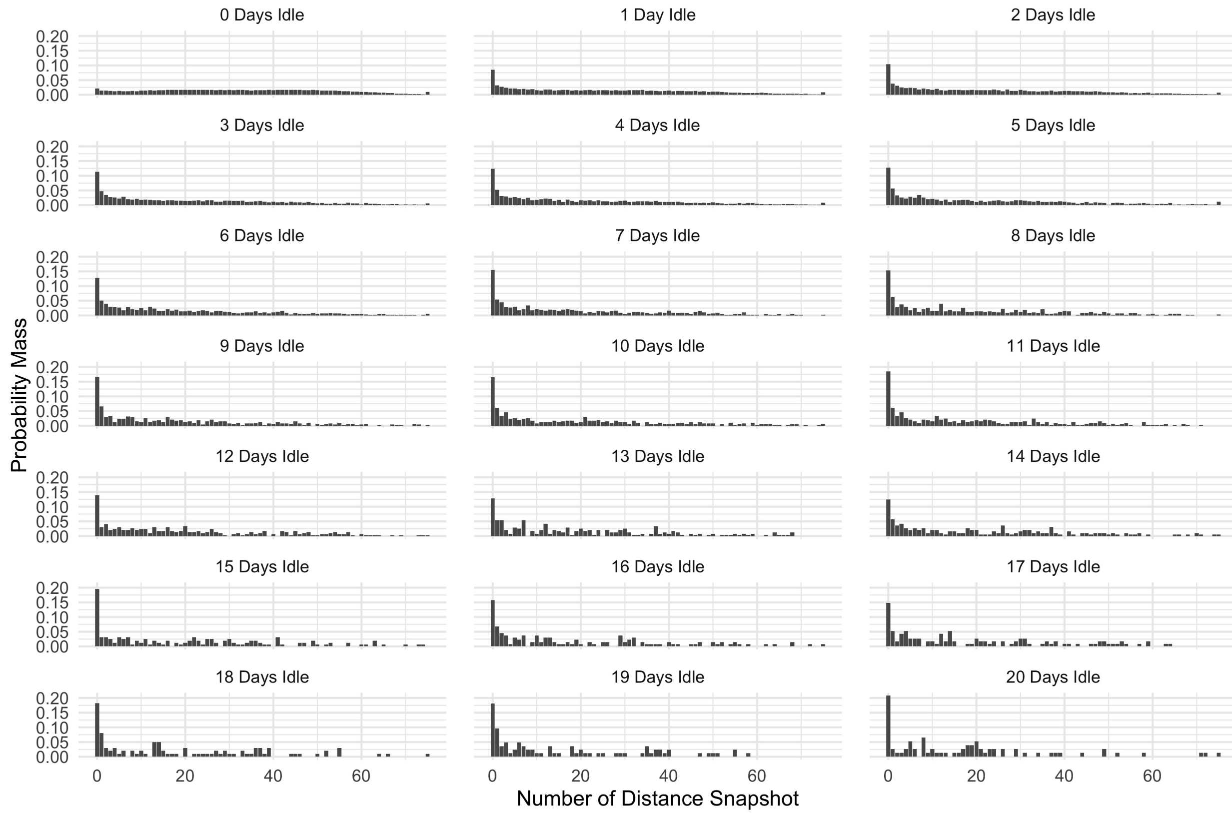

Figure 2 This figure depicts the probability mass function of the number of geographic snapshots associated with a given cleaner-date. Each plot corresponds to different value of `days_idle`, which reports the number of days since the cleaner last cleaned a residence. For example, the top-left plot corresponds to the `days_idle` $= 0$ observations, in which the given cleaner worked on the given day; the top-middle plot corresponds to the `days_idle` $= 1$ observations, in which the given cleaner did not work on the given day, but worked on the day prior; and the top-right plot corresponds to the `days_idle` $= 2$ observations, in which the given cleaner did not work on the given day or the day before that, but worked two days prior. The main takeaway is that the number of distance snapshots is high even when `days_idle` is high, which suggests that for most cleaners the app keeps running on their phones (e.g., in the background process), even when they're not working. For example, 79.2% of phones still collect geographic data—i.e., the point mass at zero accounts for only 20.8% of the probability measure—in the extreme case in which the cleaner hasn't worked in 20 days. To put this in context, only 1.14% of `days_idle` values extend beyond two weeks, and only 0.59% extend beyond three weeks.

area of land that's approximately r km from an apartment decreases linearly with r. Third, and most important, the `Non_Working` distribution almost perfectly mirrors the `Unobserved` distribution.

The stark similarity between these two distributions is our most compelling evidence for our null result. Intuitively, we would expect the `Unobserved coarsened_shortest_distance` values to match the `Non_Working coarsened_shortest_distance` values if no contracts disintermediated so that the cleanings stopped before the post period; we would expect them to match the `Observed` values if all contracts disintermediated so that the cleanings continued as before

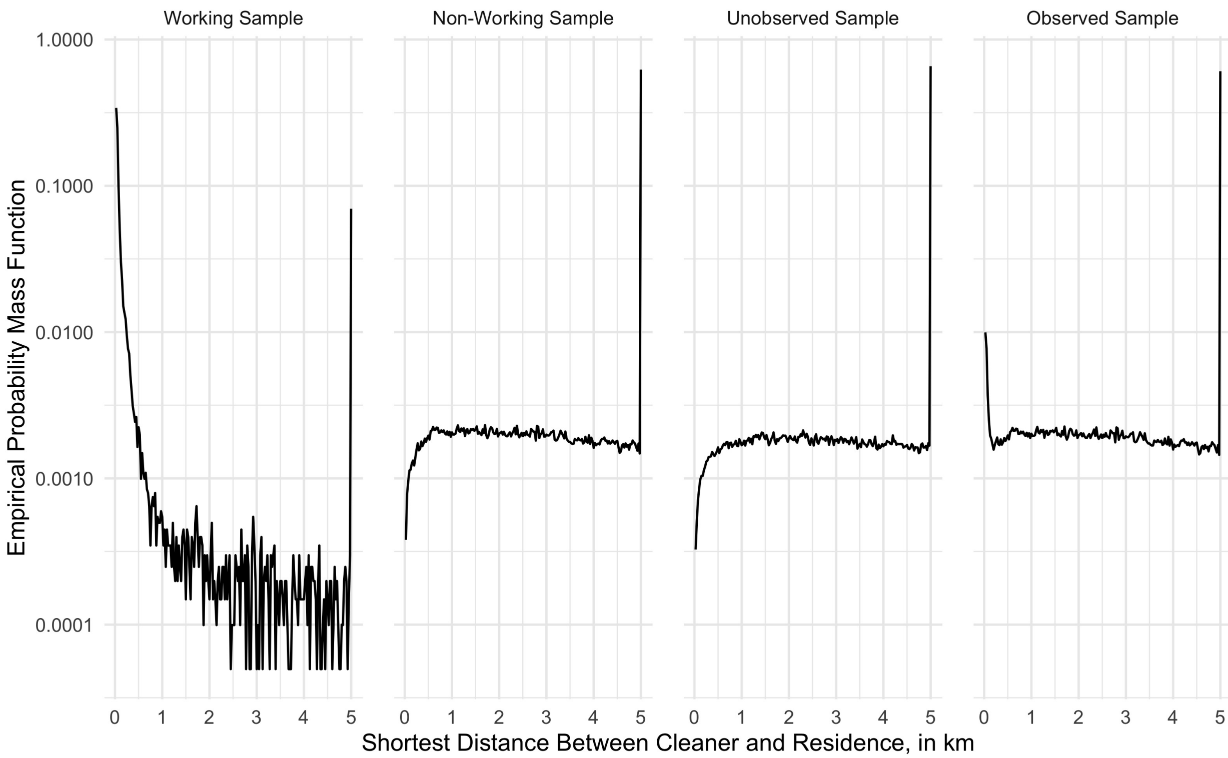

Figure 3 This figure compares the empirical distribution of `coarsened_shortest_distance` in the `Observed`, `Unobserved`, `Non_Working`, and `Working` samples. The distributions all have large point masses at 5 km, which was our truncation limit (so that `shortest_distance`≥ 5 km corresponds to `coarsened_shortest_distance`$= 5$ km). The `Working` distribution spikes near zero, which indicates that we do generally see the cleaners in the vicinity of their work. The `Observed` sample inherits this spike, but to a lesser degree, since only a fraction of this sample's observations correspond to a cleaning. In contrast, the `Unobserved` and `Non_Working` distributions both dip down near zero. The key take-away is that the `Unobserved` distribution matches the `Non_Working` distribution, but not the `Observed` distribution.

in the post period; and we would expect them to resemble a mix between the `Observed` and `Non_Working` values if a fraction of contracts disintermediated so that the cleaners still visit the dwellings in the post period, but less often. Figure 3 depicts the first of these three cases: with the characteristic dip at zero, the `Unobserved` `coarsened_shortest_distance` distribution resembles the `Non_Working` `coarsened_shortest_distance` distribution but not the `Observed` `coarsened_shortest_distance` distribution. In fact, the `Unobserved` sample has even fewer small `coarsened_shortest_distance` values than the `Non_Working` sample: the share of observations with a `coarsened_shortest_distance` value less than 25, 50, and 100 meters is 26, 15, and 7.3 times higher in the `Observed` sample than the `Non_Working` sample and is 30, 21, and 9.9 times higher in the `Observed` sample than the `Unobserved` sample. Indeed, rather than sneaking back for unofficial work, it seems the cleaners avoid their old work sites: the mean and median `shortest_distance` is 20.3 and 7.06 km in the `Non_Working` sample and 23.9 and 7.84 km in the `Unobserved` sample.

Unfortunately, figure 3 is not definitive, because although most cleaners leave the app running almost indefinitely (see figure 2 and related to it discussion on page 9), the platform's app may still not be open for all of the disintermediated cleanings (unlike conventional cleanings, which require the app to be running). Since it's impossible to detect a cleaning when the app is closed (or when the phone is switched off), we will pursue the more modest goal of estimating the frequency of disintermediated cleanings *with an open app*. In other words, we will estimate $\pi = \gamma\theta$ rather than θ, where θ denotes the rate of disintermediation and γ denotes the fraction of disintermediated cleanings for which the app is running. We should expect $\gamma < 0.94$, since the app is normally closed around 6% of the time, as we saw in section 4, and some cleaners will exit out of the app to avoid detection before performing a disintermediated cleaning. Unfortunately, we can't empirically establish the fraction of cleaners that have the foresight to exit out of the app before doing some unsanctioned cleanings, so for this we will have to use our judgement. If a reader strongly believes that *all* cleaners will be strategic *all of the time* then they may as well stop reading now, because our null result—i.e., small π estimate—is worthless unless one can be convinced that some cleaners will slip up and inadvertently disclose the disintermediated cleanings—i.e., that γ is bounded away from zero. More specifically, our null result has bite if we suppose that $\gamma \geq 0.1$, in which case our estimates suggest that fewer than one in 83 relationships ends up disintermediating.

Based on our collective experiences with house cleaners in this post-Soviet country, we suspect that no more than half of the cleaners would switch off their phones or explicitly exit out of the app and its background process (see references in footnote 1) to avoid getting caught. The working class in this country is not known for its technological sophistication. Accordingly, we can safely conclude that few cleaners would have read the app's terms of service and that few would have reflected on the implications of responding to "this app would like to track your location" with "allow while using the app" (note also that this choice was *not* even offered by either iOS or Android systems at the time when the dataset was collected). Accordingly, only a small share of cleaners should be cognizant of the geo-tracking. And even this minority of cleaners could fail to hide a disintermediated cleaning because (i) they forgot to exit out of the app (which also requires exiting the process running in the background), (ii) they are intrisically reckless or carefree, (iii) it doesn't occur to them that the platform could use their coordinates to determine whether they are disintermediating, or (iv) they don't think the platform would punish them if it does suspect they are disintermediating. This last point isn't as foolhardy as it sounds, as we are fairly certain that the app had never sanctioned any of its users for disintermediating, by the time these data were collected. All this is to say, we believe that setting $\gamma = 0.94/2 = 0.47$ would yield a fairly conservative lower bound, and setting $\gamma = 0.1$ would yield an extremely conservative lower bound.

In addition to not knowing γ, there's a second drawback we must disclose: we do not have a sample of disintermediated cleanings to train our model on. Accordingly, we must impose a second identifying assumption.

ASSUMPTION 2. *Cleaners that disintermediate with the app closed will resemble cleaners on their days off, whereas cleaners that disintermediate with the app open will resemble cleaners on their working days. More specifically,* `coarsened_shortest_distance` *will follow the* `Non_Working` *sample distribution when the app is closed during the disintermediated cleaning, and it will follow the* `Working` *sample distribution when the app is open.*

The first claim—that the distance readings of disintermediated-cleaning days should match those of official-cleaning days when the app is running—is mild, because the distance readings in this case should depend on little more than the duration of the cleanings and the frequency of the geotags, neither of which should meaningfully differ for disintermediated cleanings. And the second claim—that the distance readings of disintermediated-cleaning days should match those of non-working days when the app is not running—is conservative, because, if anything, we would expect the disintermediated distances to be shorter than the `Non_Working` distances, because we should still tend to find the cleaners in the general vicinity of the apartment, even if they turn the app off for the actual cleaning. And over-stating this distance distribution will inflate our estimate of π, as our estimator will attribute the shorter-than-expected distances to additional open-app disintermediations. Further, keep in mind that the `coarsened_shortest_distance` distributions are flat away from zero (see figure 3), so our assumption is really just specifying that we should expect a spike near zero when the phone is on and a dip near zero when it's off.

Armed with our two assumptions, we can now estimate π, the fraction of `Unobserved` observations with a disintermedated cleaning during which the app was open. To capture the distance distributions non-parametrically, we will suppose that a `coarsened_shortest_distance` is drawn from general probability mass function p^W over grid $D = \{25, 50, \cdots, 5000\}$ when it corresponds to an official cleaning or an open-app disintermediated cleaning, and we suppose that this distance is drawn from general probability mass function p^N over D when it corresponds to a closed-app disintermediated cleaning, or to no cleaning at all. In other words, our model supposes that all `Working` distances resolve from p^W, all `Non_Working` distances resolve from p^N, around a π fraction of `Unobserved` distances resolve from p^W, and around a $1-\pi$ fraction of `Unobserved` distances resolve from p^N. Our `coarsened_shortest_distance` values thus correspond to empirical log-likelihood function

$$\mathcal{L}(\pi, p^W, p^N) = \sum_{d \in D} n_d^W \log(p_d^W) + n_d^N \log(p_d^N) + n_d^U \log\left(\pi p_d^W + (1-\pi) p_d^N\right),$$

where n_d^W, n_d^N, and n_d^U represent the number of `Working`, `Non_Working`, and `Unobserved` observations with `coarsened_shortest_distance` $d \in D$.

The corresponding maximum likelihood estimator is

$$
\begin{aligned}
(\hat{\pi}, \hat{p}^W, \hat{p}^N) &= \operatorname*{arg\,max}_{\pi, p^W, p^N} \mathcal{L}(\pi, p^W, p^N) \\
\text{subject to} \quad & \sum_{d \in D} p_d^W = 1, \\
& \sum_{d \in D} p_d^N = 1, \\
\text{and} \quad & \pi,\ (1-\pi),\ p_d^W,\ p_d^N \geq 0.
\end{aligned}
$$

Our empirical model is flexible, with 399 degrees of freedom: the value of $\hat{\pi}$, plus the 200 parameters that define $\hat{p}^W$, plus the 200 parameters that define $\hat{p}^N$, minus the two sum-to-one constraints. Nevertheless, our model has a simple closed-form solution, namely

$$
\begin{aligned}
\hat{\pi} &= 0, \\
\hat{p}_d^W &= n_d^W / N^W, \\
\text{and} \quad \hat{p}_d^N &= (n_d^N + n_d^U)/(N^N + N^U),
\end{aligned}
$$

where $N^W = \sum_{d \in D} n_d^W$, $N^N = \sum_{d \in D} n_d^N$, and $N^U = \sum_{d \in D} n_d^U$ are the total number of `Working`, `Non_Working`, and `Unobserved` observations. To confirm that this boundary solution is locally optimal, we will show that $\frac{\partial}{\partial \pi} L(\pi) < 0$ at $\pi = 0$, where

$$
\begin{aligned}
L(\pi) &= \max_{p^W, p^N} \mathcal{L}(\pi, p^W, p^N) \\
\text{subject to} \quad & \sum_{d \in D} p_d^W = 1, \\
& \sum_{d \in D} p_d^N = 1, \\
\text{and} \quad & p_d^W,\ p_d^N \geq 0.
\end{aligned}
$$

We can calculate this derivative with the envelope theorem:

$$
\begin{aligned}
\tfrac{\partial}{\partial \pi} L(\pi)\big|_{\pi=0} &= \tfrac{\partial}{\partial \pi} \mathcal{L}(\pi, \hat{p}^W, \hat{p}^N)\big|_{\pi=0} \\
&= \sum_{d \in D} n_d^U p_d^W / p_d^N - n_d^U \\
&= \sum_{d \in D} \frac{N^N + N^U}{N^W} \frac{n_d^U n_d^W}{n_d^N + n_d^U} - n_d^U.
\end{aligned}
$$

In our case, this derivative amounts to -102,084, which indicates that the log-likelihood function quickly decreases as $\hat{\pi}$ increases from zero.

To illustrate that our MLE estimates are *globally* maximal, we report likelihood ratio $2L(0) - 2L(\pi)$ for various values of π in table 1. These values are uniformly positive, which confirms that our log-likelihood function is maximized at $\hat{\pi} = 0$. Further, likelihood ratio $2L(0) - 2L(\pi)$ has a chi-square limiting distribution with one degree of freedom under the null hypothesis that the true disintermediation fraction is π. Accordingly, we reject the null hypothesis that π is as large as 0.00005 at the p =0.002 level, because $2L(0) - 2L(0.00005)$ =10.7 and the 0.998 quantile of the one-degree-of-freedom chi-squared distribution is 9.55. In other words, we reject the null hypothesis that more than one in $1/0.00005 = 20{,}000$ `Unobserved` observations has an open-app disintermediated cleaning. In contrast, 4.16% of (cleaner, residence, date) triples have corresponding cleanings, while the cleaner is still officially associated with the residence.[4] Accordingly, the termination of the official relationship marks a $0.0416/0.00005 = 832$ factor decrease in the rate of open-app cleanings, from one every $1/0.0416 = 24$ days to less than one every $1/0.00005 = 20{,}000$ days. This suggests that not more than one in $0.1 \cdot 832 = 83$ relationships ends in disintermediation, assuming that γ, the fraction of disintermediated cleanings for which the app is running, exceeds 0.1. And if we suppose $\gamma \geq 0.47$, which we've argued above is likely, then our results suggest that not more than one in $0.47 \cdot 832 = 391$ relationships end in disintermediation.

Here's another possible explanation for our result: perhaps we don't detect many off-the-books cleanings not because disintermediated relationships fail to form, but because they last for only a short while. To address this possibility, we rerun our model without the `Unobserved` observations that occurred more than 90 days after the last official cleaning for a given (cleaner, residence) pair. For example, cleaner 2062 last officially cleaned residence 12536 on 2015-10-22, and so now the (cleaner, residence) = (2062, 12536) observations in our `Unobserved` sample span from 2015-10-23 to 2015-10-22 + 90 days = 2016-01-20. We then do the same, but with with 180- and 360-day windows. We faithfully reproduce our results in all three cases: we get the same $\hat{\pi} = 0$ corner solution under the 90-, 180-, and 360-day filters, and reject the same hypothesis that disintermediated cleanings occur more frequently than one out of $1/0.00005 = 20{,}000$ days (see table 2). However, we can now reject this hypothesis only at the p =0.024 level (as opposed to the p =0.0011 level, which we had before).

[4] To calculate this statistic, we first compute the number of days between the successive cleanings of each (cleaner, residence) pair. Then, to mitigate censoring, we remove the inter-cleaning time spans that started after 132 days before the end of our sample horizon. (We choose 132 days, because that is the 95th percentile of the inter-cleaning times in our sample.) We then calculate the average of the remaining inter-cleaning time spans. For example, suppose there was only one (cleaner, residence) pair, and that it had five cleanings, on 2016-01-10, 2016-02-01, 2016-02-08, 2016-02-16, and 2016-02-25. From these five cleanings we would create four time spans: 2016-02-01 - 2016-01-10 = 22 days, 2016-02-08 - 2016-02-01 = 7 days, 2016-02-16 - 2016-02-08 = 8 days, and 2016-02-25 - 2016-02-16 = 9 days. We would then disregard the last time span, because it started on 2016-02-16, which fell after the cutoff threshold of 2016-02-12 (which is 132 days before the end of our horizon, on 2016-06-23). So, in this case, we would be left with three time-span-starting cleanings—on 2016-01-10, 2016-02-01, and 2016-02-08—and 22 + 7 + 8 = 37 time span days, and would thus estimate that 3/37 = 8.11% of (cleaner, residence, date) triples correspond to a cleaning.

π	$2L(0)-2L(\pi)$	p
0.00000005	0.0102	0.9195
0.0000001	0.0204	0.8864
0.0000005	0.102	0.7493
0.000001	0.204	0.6512
0.000005	1.03	0.3111
0.00001	2.06	0.1510
0.00005	10.7	0.0011
0.0001	22.5	0.0000
0.0005	154.	0.0000
0.001	405.	0.0000
0.005	4075.	0.0000
0.01	9847.	0.0000
0.05	46821.	0.0000
0.1	68725.	0.0000

Table 1 This table reports likelihood ratio statistics $2L(0)-2L(\pi)$, and corresponding p values, for various values of π. The likelihood ratio statistic has a one-degree-of-freedom chi-squared limiting distribution under the null hypothesis that the true disintermediation rate is π. We reject the null hypothesis that π is as large as 0.00005 at the $p=$0.0011 level, because there's less than a 0.11% chance of a chi-squared with one degree of freedom exceeding 10.7. The likelihood ratio statistics being uniformly positive and increasing confirms that our empirical likelihood function is globally maximized at $\hat{\pi}=0$.

π	$2L(0)-2L(\pi)$	p
0.00000001	0.000937	0.9756
0.00000005	0.00468	0.9454
0.0000001	0.00937	0.9229
0.0000005	0.0469	0.8286
0.000001	0.0938	0.7593
0.000005	0.472	0.4919
0.00001	0.952	0.3291
0.00005	5.07	0.0243
0.0001	10.9	0.0010
0.0005	85.1	0.0000
0.001	240.	0.0000
0.005	2679.	0.0000
0.01	6700.	0.0000
0.05	37220.	0.0000
0.1	65078.	0.0000

Table 2 This table reports likelihood ratio statistics and p values in the fashion of table 1, except these figures consolidate the results from three different specifications: when the `Unobserved` sample is limited to the first 90, 180, and 360 days after each (cleaner, residence) pair's last official job. Specifically, the the table reports the most conservative results across these three specifications, tabulating the smallest $2L(0)-2L(\pi)$ values and the largest p values, across the three cases. We reject the null hypothesis that π is as large as 0.00005 at the $p=$0.024 level, because there's less than a 2.43% chance of a chi-squared with one degree of freedom exceeding 5.07.

Finally, we confirm our estimator's validity with a brief simulation study. To begin, we create a new `Unobserved` panel for each $\pi \in \{2^{-5}, 2^{-6}, \cdots, 2^{-12}\}$ by combining πN^U randomly drawn `Working` observations and $(1-\pi)N^U$ randomly drawn `Non_Working` observations. We then use our original `Working` and `Non_Working` panels and each simulated `Unobserved` panel to create

$\log_2(\pi)$	Estimate	Std. Error
-5	-5.00	0.01
-6	-6.00	0.02
-7	-6.99	0.02
-8	-7.99	0.04
-9	-8.97	0.07
-10	-9.98	0.13
-11	-10.83	0.22
-12	-11.98	0.46

Table 3 This table reports the results of our Monte Carlo simulation study. We simulate seven new `Unobserved` samples by randomly drawing πN^U observations from the `Working` sample and $(1-\pi)N^U$ observations from the `Non_Working` sample, for $\pi = 2^{-5}, 2^{-6}, \cdots, 2^{-12}$. We then use our maximum likelihood estimator to estimate $\log_2(\pi)$ (along with the corresponding $\hat{p}^W$ and $\hat{p}^N$ parameters). Finally, we tabulate our estimates of $\log_2(\pi)$ and these estimates' standard errors. (We reformulate the maximum likelihood problem in terms of $\log_2(\pi)$ to get correct standard errors.)

maximum likelihood estimates $\hat{\pi}$, $\hat{p}^W$, and $\hat{p}^N$. Lastly, we compare the chosen π values with the estimated $\hat{\pi}$ values in table 3. As you see, our estimator can reverse engineer π with high precision.

6. Conclusion

A reader can have faith that we did not p-hack our finding, because it was not the finding we wanted. Indeed, we did not set out to report a negative result; rather, we started with high hopes for detecting the theorized phenomenon. However, we soon discovered that no matter how we twisted our data, it would not cough up a story of disintermediation.

Indeed, we find essentially zero geotagged disintermediated cleanings: specifically, we reject the null that the rate of unprotected disintermediation exceeds one in 20,000 days. This implies that if *at least* 10% of cleaners leave the app running while they do their off-the-books cleanings—which should be the case, as discussed on pages 9 and 13—then the overall rate of disintermediated cleanings does not exceed one in 2,000 days.

References

Chaves IN (2018) Pricing responses to platform leakage: Optimal platform design when matches are irrevocable. Working Paper, https://isachavesv.github.io/papers/irrevocableMatches.pdf.

Deléchat C, Medina L (2020) What is the informal economy? https://www.imf.org/Publications/fandd/issues/2020/12/what-is-the-informal-economy-basics.

Angi Research & Economics (2022) The economy of everything home. https://www.angi.com/research/reports/market/.

TechCrunch (2015) Why homejoy failed ... and the future of the on-demand economy.

Washington Post (2019) It's the middle of the night. do you know who your iphone is talking to?

Farronato C, Fong J, Fradkin A (2022) Dog eat dog: Measuring network effects using a digital platform merger. *Management Science* Forthcoming.

Gu G (2022) Technology and disintermediation in online marketplaces. Working Paper, SSRN 4150094.

Gu G, Zhu F (2021) Trust and disintermediation: Evidence from an online freelance marketplace. *Management Science* 67(2):794–807.

Hagiu A, Wright J (2022) Marketplace leakage. Working Paper, http://andreihagiu.com/wp-content/uploads/2022/07/Leakage-Website.pdf.

He EJ, Savin S, Goh J, Teo CP (2023) Off-platform threats in on-demand services. *Manufacturing & Service Operations Management* Forthcoming.

He S, Peng J, Li J, Xu L (2020) Impact of platform owner's entry on third-party stores. *Information Systems Research* 31(4):1467–1484.

Karacaoglu N, Li S, Stamatopoulos I (2022) Disintermediation evidence from a cleaning platform. Working Paper, SSRN 4222023.

Ladd T (2022) The achilles' heel of the platform business model: Disintermediation. *Business Horizons* 65(3):277–289.

Lin J, Nian T, Foutz NZ (2022) Disintermediation and its mitigation in online two-sided platforms: Evidence from airbnb. Working Paper, https://questromworld.bu.edu/platformstrategy/wp-content/uploads/sites/49/2022/07/PlatStrat2022_paper_37.pdf.

Sekar S, Siddiq A (2023) Platform disintermediation: Information effects and pricing remedies. Working Paper, SSRN 4378501.