**Web Appendix for**

# Can an AI Algorithm Mitigate Racial Economic Inequality? An Analysis in the Context of Airbnb

**Table of Contents**

## Section 1. Technical Notes on Facial Analyses

The key individual demographic variables—namely, race and age—are not provided/available on Airbnb. Hence, the only option was to obtain this information from hosts' profile photos. Given the sample size, we opt for an automated approach that uses a Convolutional Neural Network (CNN, an emerging deep learning framework, see Krizhevsky et al. 2012 and Simonyan and Zisserman 2015). Specifically, we employ the ResNet-50 framework, a CNN that has led to important breakthroughs on various computer vision tasks including facial recognition and image classification (for example, Cao et al. 2018**,** He et al. 2016).

### A. The workflow of facial analysis

We build a deep learning-based classifier to predict the race and age of any Airbnb host based on their profile photo. We first construct a large training set of human face photos in which the race and age is known. We run the deep learning model (i.e., ResNet-50) on the training set, and it learns the relationship between the extracted facial features and corresponding race and age labels. Finally, the trained classifier is applied to our Airbnb face photos to predict the labels for each host in our sample. In the next subsections, we describe each step in detail.

### B. Face data set (training data)

For the race classification, we combine multiple public face databases: the color Facial Recognition Technology (FERET) Database collected by the National Institute of Standards and Technology (NIST)[1], Chicago Face Database (CFD) collected by the University of Chicago[2], Face Place database collected by Brown University[3], and part of the IMDB-WIKI image database created by the Computer Vision Lab at ETH Zurich[4].

For the age prediction, we use only the IMDB-WIKI image database, which contains 500,000 images of celebrities taken from IMDB and Wikipedia webpages. Each age label is calculated from the celebrity's date of birth and the date of the photo (Rothe et al. 2015).

### C. Preprocessing: detecting and extracting faces

When analyzing each face photo, the deep learning model first detects the presence of a face and then extracts the face; this excludes non-face content such as the person's neck, shoulders, and background. Detection and extraction are straightforward for images in the training set (i.e., the public face data) since each image contains one face. In the Airbnb sample, a few images do not contain a person's face and are discarded; other images contain multiple faces, so we extract and store all faces for analysis.

A CNN is a special kind of a deep learning model. As shown in Figure S1, a deep learning model consists of a sequence of layers, with each layer containing multiple neurons that represent a multidimensional matrix. Each neuron "carries" weight that represents the numeric value of the corresponding element. The number of layers that carry weight define the "depth" of a deep learning model.

---

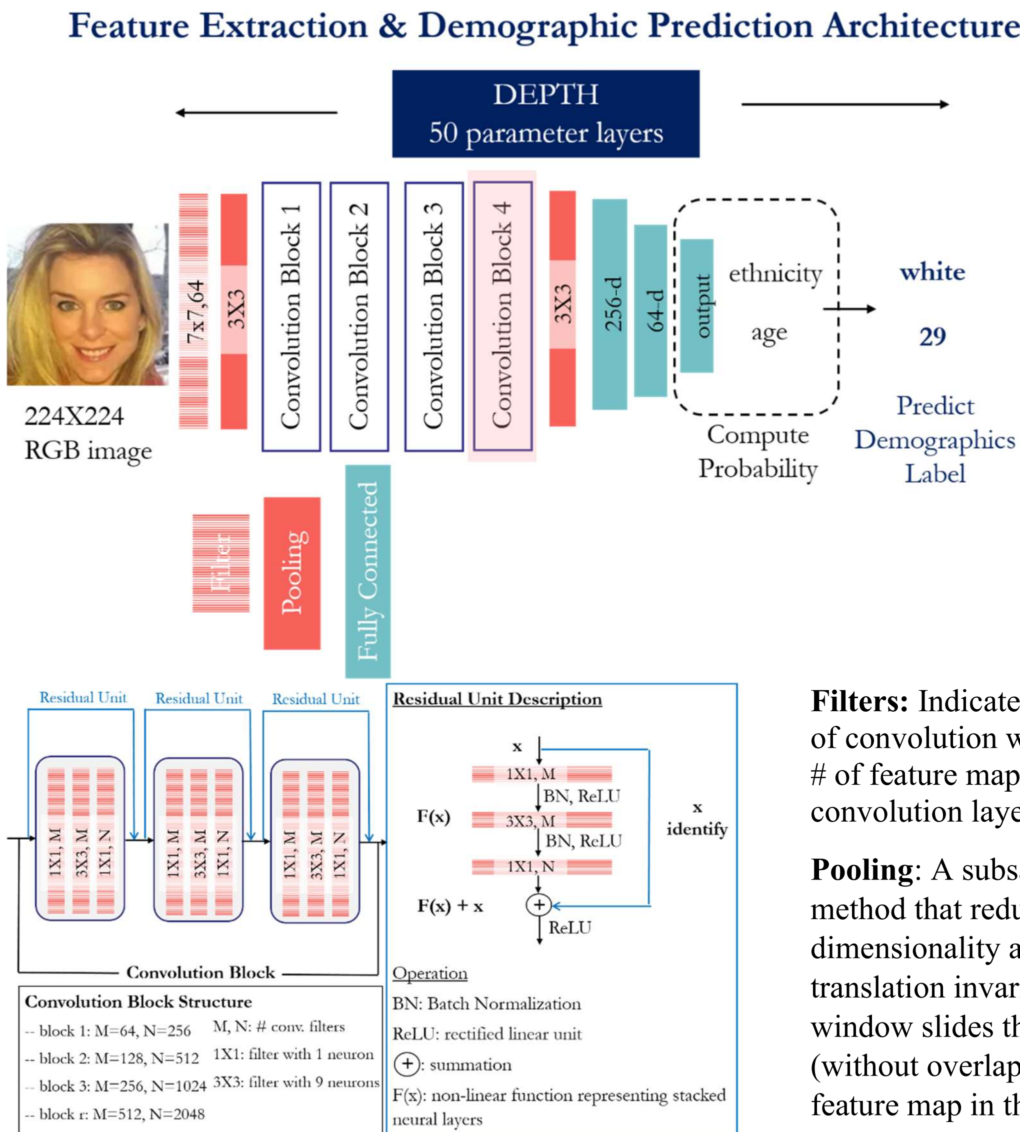
[1] https://www.nist.gov/itl/iad/image-group/color-feret-database.

[2] http://faculty.chicagobooth.edu/bernd.wittenbrink/cfd/index.html.

[3] http://wiki.cnbc.cmu.edu/Face_Place.

[4] https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/.

**Figure S1 ResNet-50: Architecture, Layer Operations, and Training Framework**




**Filters:** Indicates the number of convolution windows (i.e., # of feature maps) on each convolution layer.

**Pooling**: A subsampling method that reduces dimensionality and provides translation invariance. A 3x3 window slides through (without overlap) each feature map in that layer. The average value in the window is picked as the representation of the window.

In a deep learning framework, high dimensional data such as images and text are expressed as multidimensional matrices/arrays. The model processes the data through the neuron layers, essentially conducting matrix multiplication. CNNs are characterized by the convolution layer, which operates dot productions on the input data (described below).

Multiple CNNA architectures have been proposed, including AlexNet by Krizhevsky et al. (2012), VGG by Simonyan and Zisserman (2015), Inception/GoogleNet by Szegedy et al. (2015), and ResNet (He et al. 2016). These variants follow the classic CNN framework (i.e., they comprise a sequence of convolution layers) but vary in the number, order, and size of the layers. Inception/GoogleNet and ResNet are notable for their special computation units that are particularly effective for training the CNN.

In this study, we choose ResNet-50 (i.e., a ResNet model containing 50 parameter layers) because the residual learning functions developed by He et al. (2016) enable the model to overcome the gradient vanishing problem, which often makes it difficult to train very deep CNNs (Glorot and Bengio 2010). ResNet has demonstrated state-of-the-art performance in various tasks such as object detection, image classification, facial recognition, and realistic voice generation (for example, Cao et al. 2018, Chen et al. 2014, Lee et al. 2017). Below, we provide a brief description of the architecture of ResNet-50[5].

**D. Architecture of ResNet-50**

Figure S1presents the architecture of ResNet-50. The model comprises repeated blocks (or modules) of convolution layers that connect the input image and the output labels. As in a classic deep learning framework, the first input is the image (in this case, the face photo), a 3D matrix containing the pixel intensities of the 3 color channels (RGB). We resize all images to 224x224 pixels as required by the architecture of the ResNet-50 model.

As introduced above, the model processes data through matrix multiplication between the input image and the first layer of neurons, generating a new multi-dimensional matrix of "useful information" extracted from the image. The new matrix becomes the input for the next layer, and so on through all the layers. The final (output) layer computes the probability distribution over the labels, and the distribution is converted to the labels.

For example, the race classification task has three possible labels *l* (*l*= {*white, Black, other*}). The output layer receives input $X$ with neuron weights $W_1$ and $W_0$, and it computes:

$$prob(l|X, W_1, W_0) = \frac{\exp\left(-\left(X^T W_1^l + W_0^l\right)\right)}{\sum_{g=1}^{L} \exp\left(-\left(X^T W_1^g + W_0^g\right)\right)} \tag{S1}$$

$L$ is the total number of possible outcomes. $W_1^l$ represents the weight parameters, and $W_0^l$ represents the bias (i.e., a constant) connecting the preceding layer (i.e., the 64-d fully connected layer) to the $l^{th}$output layer (i.e., (the 3-d fully connected layer). $X^T$represents the output from the layer preceding the output layer. For image $IMG^k$ in the training set, the model outputs the label with the highest probability:

$$\widehat{Label}(Ethnicty|IMG^k) = l, s.t.\, prob(l|X, W_1, W_0) > prob(g|X, W_1, W_0)\ \forall\, \mathrm{g} \in 1 \ldots L \tag{S2}$$

Note that $X$ is the output extracted from the implementation of the input image on all preceding layers, so it is a function of the model weights and the input image (i.e., $X = \Phi(IMG^k, W)$).

The CNN model involves a sequence of weights within each layer, and the weights define the intermediate extracted vectors from each layer, including $X^T$. For each training image $IMG^k$,

---

[5] For a detailed visualization and interactive introduction to the model architecture, please visit http://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006.

we know the true race label; during training, the model adjusts the weights to optimize the accuracy of its label predictions.

**E. Operations of key layers**

We describe the convolution layer and pooling layer that characterize the CNN.

**Convolution layer**

The convolution layer is the most important and unique layer in the CNN. It comprises a stack of so-called convolution filters or convolution kernels, each of which contains a matrix with each element representing a numeric value (e.g., a 3X3 block contains 9 values)[6]. The matrix treats the input image or an intermediate input as another matrix and operates a dot production by "sliding" through the input. For a large input (e.g., face image with 224X224 pixels), the filter operates dot production for every 3X3 patch on the input matrix.

The convolution operation offers two key benefits: it reduces the dimensionality of the parameters, and it thoroughly explores and reserves the (local) spatial relationships within the input. If one convolution kernel extracts an edge of an object with a particular orientation, then operating this kernel on every small square (e.g., 3X3 and 1X1) of the image would extract all edges in that direction from the image. Together, all the kernels that extract edges would extract edges in every direction, constructing the contour of an object.

Figure S1 depicts the variation in the number of convolutional filters per block (e.g., 64, 128, 256, 512, 1024, and 2048 filters). The output data from one layer becomes the input data for the next; the filters extract features at higher and higher levels in subsequent levels. That is, the CNN extracts a hierarchical structure of features that are related to the output labels.

**Pooling layer**

Many CNNs contain a pooling layer between every pair of convolution layers. A pooling layer is a small square filter (a 3X3 matrix in our model). It applies to every 3X3 square patch on the input data (much like the convolution layer); then, it determines the average value in the 3X3 square. Pooling layers reduce the spatial size of the intermediate features and the dimensions of the trained parameters, which reduces the risk of over-fitting.

**F. Training and application**

The prediction tasks are implemented independently. For each task, we randomly split the dataset: 80% of the images became the training set, and the remaining 20% were reserved for the (hold-out) test set. We built a powerfully predictive model by taking advantage of transfer learning—that is, we built our model on top of an existing model that was created for a task that is similar to but distinct from our own. Cao et al. (2018) trained their classic ResNet-50 model on VGGFace2, a large-scale face dataset with 3.3 million images depicting over 9,000 subjects[7]. We modified the architecture slightly by removing the output layer, as it was specific to the original task (object classification), and we replaced it with three fully connected layers (dimensions of 256, 64, and 3), where the last layer is the output layer.

We initialized the model weights with the pre-trained weights from the original ResNet-50 and then fine-tuned the parameters. Information extracted from images is somewhat generic for

[6] The size of the convolution layer varies with the model architecture. 3X3 and 5X5 are common choices.

[7] http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/.

many types of tasks (e.g., most early layers detect edges and contours). Hence, we began our optimization process at a point that was already close to "optimal" because Cao et al. previously trained the RestNet-50 model for a facial recognition task. We initialized our three added layers with LeCun's uniform scaled initiation method (LeCun et al. 1998).

To ensure that the trained model would generalize to the main task (i.e., with the Airbnb host dataset), we augmented the data in real time to introduce random variation, increase the training set size, and reduce overfitting (Krizhevsky et al. 2012). Specifically, for each image in the training sample, we randomly 1) flipped the input image horizontally, 2) rescaled the input image by a multiplier of 1.2, and 3) rotated the image within 20°.

We trained the model on NVIDIA GeForce Titan X-12GB-GPU for 100 epochs, and we tested the model's performance on the hold-out test set at the end of each epoch. For optimization, we used the adaptive method of gradient descent (Adadelta optimization, see Zeiler 2012) on each mini-batch of 32 examples.

The race classifier predicted the race of each person as white, Black, or other with an accuracy of 92.5%. The age classifier predicted the person's age (between 1 and 100) with a MAE (mean absolute error) of 4.750 (years). Finally, we used the trained ResNet-50 model to predict the race and age of the Airbnb hosts in our sample.

## Section 2. Data Sources and Descriptive Statistics

### 2.1 Data sources

An overview of our main data sources and variables:

(i) We used InsideAirbnb.com, a publicly available website, to build a list of all Airbnb properties listed in October 2015 in each of the seven selected cities (Austin, Boston, Los Angeles, New York, San Diego, San Francisco, and Seattle). We also collected the unique listing ID associated with each property. We used this list to randomly sample the hosts (see Section 2.2).

(ii) We used AirDNA, a third party, to obtain the average nightly rate, monthly occupancy rate, and other characteristics mentioned on the webpage of each property. This data spans July 2015 to August 2017 and covers all Airbnb listings in the aforementioned seven cities.

(iii) We scraped the webpage of each property in our random sample to determine whether and when the host used the smart pricing algorithm. We also collected each host's profile photo to determine the race and age using a deep learning model (see Section 1).

(iv) We used Zillow Research to estimate the value of each listed property. We collected the median income and education within each zip code from the American Community Survey (i.e., US Census).

### 2.2 Sample construction

We first obtained the data from AirDNA. Our next task was to randomly sample a set of properties from the entire set of Airbnb properties. From InsideAirbnb.com, we obtained a list of the 66,424 properties (each with a unique ID) that were available in October 2015 in the seven aforementioned cities. We used *random.sample* (*population, k*) from the Python library to obtain a random sample without replacement. (*Population* is the list or sequence from which the random sample is chosen, and *k* is the length of returned sample list; we set $k$ = 13,200, about 20% of the original sample).

For each property in the random sample, we scraped its webpage for the monthly property calendar from November 2015 through August 2017. Of the 13,200 properties, 12,587 returned a valid calendar, and we excluded the rest (which returned an invalid page or an error) because we needed the calendar to determine algorithm adoption, the treatment variable. From the calendar pages that we scraped from the 12,587 listings every month, we obtained the information whether the property's price in that month was determined by the smart pricing Then, we scraped each host's profile photo. A human face was detected in 10,924 of the 12,587, and we excluded the remaining properties because we needed a face photo to determine the host's race, another key variable. (Some of the excluded photos had a non-human object such as a pet; others had a face that was too small or unclear for an accurate assessment of race and age.)

We attempted to match the 10,924 property IDs with the AirDNA dataset; we succeeded for 10,903 properties and excluded the rest. Finally, we excluded "stale vacancies," in which the property appears to be available but is never reserved because the host does not respond to booking requests. Fradkin et al. (2017) found that about 15% of guest requests are rejected due to this issue. Following Zalmanson et al. (2018), we removed properties that did not have any booking during our observation. The remaining 9,396 properties constituted our final sample.

### 2.3 Data description

***Summary statistics***

In Table S1, we report the statistics of all variables in our sample before the IPTW method (Section 3). The data represents the 9,396 properties that resulted from the selection process in Section 2.2, and we group them by algorithm adoption (2,118 adopters in column (1) and 7,278 non-adopters in column (2)).

**Table S1 Full Summary Statistics of All Variables Before IPTW, Presented by Adoption Group**

| | (1) Adopters | | (2) Non-Adopters | | (3) All Properties | |
|---|---|---|---|---|---|---|
| **VARIABLE** | **Mean** | **Std. Dev.** | **Mean** | **Std. Dev.** | **Mean** | **Std. Dev.** |
| # Properties | 2118 | | 7278 | | 9396 | |
| **Airbnb Host Demographics (Measured from Profile Photo)** | | | | | | |
| White (Race) | 0.83 | 0.37 | 0.75 | 0.43 | 0.77 | 0.42 |
| Black (Race) | 0.07 | 0.25 | 0.12 | 0.32 | 0.11 | 0.31 |
| Other (Race) | 0.10 | 0.30 | 0.13 | 0.34 | 0.12 | 0.32 |
| Age | 35.87 | 10.18 | 35.36 | 10.00 | 35.48 | 10.04 |
| # Photographed Faces | 1.39 | 0.78 | 1.36 | 0.76 | 1.37 | 0.76 |
| **Airbnb Property Performance** | | | | | | |
| Daily Revenue (on non-blocked days) | 80.31 | 98.04 | 67.41 | 101.70 | 70.91 | 100.88 |
| Occupancy Rate | 0.50 | 0.38 | 0.38 | 0.39 | 0.41 | 0.39 |
| #Reservation Days | 10.97 | 11.08 | 6.60 | 9.70 | 7.66 | 10.22 |
| #Blocked Days (on all months) | 9.10 | 11.73 | 12.96 | 13.33 | 12.03 | 13.07 |
| **Airbnb Property Characteristics** | | | | | | |
| Apartment | 0.57 | 0.49 | 0.66 | 0.47 | 0.64 | 0.48 |
| Entire Home | 0.56 | 0.50 | 0.64 | 0.48 | 0.62 | 0.49 |
| # Bedrooms | 1.29 | 0.82 | 1.32 | 0.87 | 1.31 | 0.86 |
| Number of Reviews | 42.88 | 51.64 | 28.07 | 41.78 | 31.65 | 44.82 |
| Number of Photos | 18.67 | 13.23 | 16.07 | 11.31 | 16.70 | 11.86 |
| Super Host | 0.23 | 0.42 | 0.16 | 0.36 | 0.17 | 0.38 |
| Instant Book Enabled | 0.18 | 0.38 | 0.10 | 0.30 | 0.12 | 0.32 |
| Listing Title Length | 33.28 | 7.34 | 32.15 | 6.54 | 32.43 | 6.76 |
| Listing Nightly Rate | 179.80 | 179.30 | 194.61 | 190.43 | 191.03 | 187.91 |
| Security Deposit | 163.63 | 331.45 | 140.76 | 310.24 | 146.29 | 315.65 |
| # Minimum Stay | 2.97 | 10.58 | 3.03 | 5.54 | 3.01 | 7.09 |
| Response Rate (%) | 94.06 | 12.81 | 92.08 | 15.40 | 92.65 | 14.73 |

| | | | | | | |
|---|---|---|---|---|---|---|
| # Host-Owned Listings | 2.54 | 3.92 | 3.02 | 8.72 | 2.90 | 7.84 |
| **Neighborhood Characteristics** | | | | | | |
| Home Value | 704.19 | 416.69 | 751.07 | 443.32 | 739.73 | 437.49 |
| Walk Score | 82.97 | 24.03 | 84.75 | 22.61 | 84.32 | 22.98 |
| Transit Score | 76.73 | 22.23 | 78.20 | 22.92 | 77.84 | 22.77 |
| Drive to Downtown (*min*) | 14.92 | 9.72 | 14.52 | 9.73 | 14.61 | 9.73 |
| Population Density (*Per Sq. Mile*) | 37051.98 | 31542.19 | 39340.66 | 33834.17 | 38786.95 | 33308.49 |
| Bachelor's Degree (%) | 51.36 | 19.28 | 54.42 | 19.06 | 53.68 | 19.15 |
| Median Home Earning (household income) (*1000 USD*) | 48.88 | 20.52 | 52.11 | 22.04 | 51.33 | 21.73 |
| **Airbnb Property Amenities Information** | | | | | | |
| Parking | 0.58 | 0.49 | 0.44 | 0.50 | 0.48 | 0.50 |
| Pool | 0.06 | 0.23 | 0.06 | 0.25 | 0.06 | 0.24 |
| Beach | 0.02 | 0.15 | 0.01 | 0.10 | 0.01 | 0.12 |
| Internet | 0.99 | 0.10 | 0.98 | 0.14 | 0.98 | 0.13 |
| TV | 0.74 | 0.44 | 0.75 | 0.43 | 0.75 | 0.44 |
| Dryer | 0.81 | 0.39 | 0.70 | 0.46 | 0.73 | 0.44 |
| Washer | 0.59 | 0.49 | 0.60 | 0.49 | 0.60 | 0.49 |
| Iron | 0.57 | 0.49 | 0.32 | 0.47 | 0.38 | 0.49 |
| Essentials | 0.72 | 0.45 | 0.50 | 0.50 | 0.55 | 0.50 |
| Heating | 0.97 | 0.18 | 0.93 | 0.25 | 0.94 | 0.24 |
| Microwave | 0.22 | 0.41 | 0.11 | 0.31 | 0.13 | 0.34 |
| Refrigerator | 0.25 | 0.43 | 0.12 | 0.33 | 0.15 | 0.36 |
| Laptop-Friendly | 0.52 | 0.50 | 0.30 | 0.46 | 0.35 | 0.48 |
| Fireplace | 0.13 | 0.34 | 0.13 | 0.34 | 0.13 | 0.34 |
| Elevator | 0.16 | 0.37 | 0.21 | 0.41 | 0.20 | 0.40 |
| Gym | 0.06 | 0.23 | 0.08 | 0.27 | 0.07 | 0.26 |
| Family-Friendly | 0.15 | 0.36 | 0.22 | 0.41 | 0.20 | 0.40 |
| Smoker Detector | 0.70 | 0.46 | 0.49 | 0.50 | 0.54 | 0.50 |
| Shampoo | 0.60 | 0.49 | 0.40 | 0.49 | 0.45 | 0.50 |
| Breakfast | 0.08 | 0.26 | 0.06 | 0.23 | 0.06 | 0.24 |
| AC | 1.00 | 0.05 | 0.99 | 0.09 | 0.99 | 0.08 |

Notes: Column (1), Adopters, contains all properties that used Smart Pricing at some point during our observational window (but not necessarily for the whole window); those in column (2), Non-Adopters, never adopted the algorithm. Rows for Race (white, Black, and other) report the proportion of properties owned by a host of the given race. All amenities are represented by dummy variables, set to 1 (0) if the feature is present (absent).

### 2.4 Anecdotal reasons for adopting or not adopting the algorithm

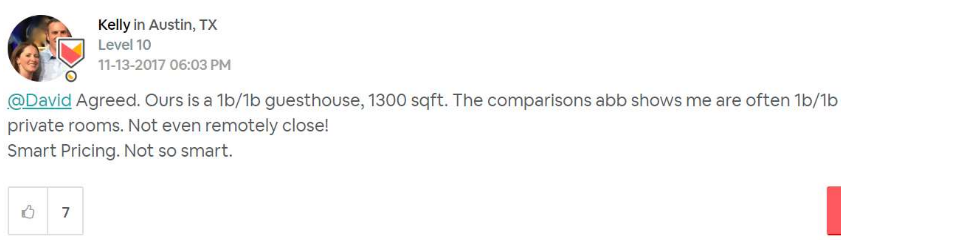
Airbnb hosts adopt the algorithm voluntary, creating a self-selection problem for our analysis. We controlled for self-selection by including covariates that may contribute to the algorithm adoption decision. We identified the primary reasons for adoption/non-adoption of the algorithm from the Airbnb Host Forum, which revealed two primary categories of opinions: 1) pro-algorithm: the algorithm reduces the time and effort required to set prices manually, and 2) anti-algorithm: the algorithm is not trustworthy (a host does not trust the algorithm or does not trust Airbnb's motives), or its price recommendations are too low. Table S2 displays selected anecdotes from the Host Forum.

**Table S2 Screenshots from Airbnb Host Forum Discussions About Why or Why Not Hosts Were Adopting the Algorithm.**

*Category 1*

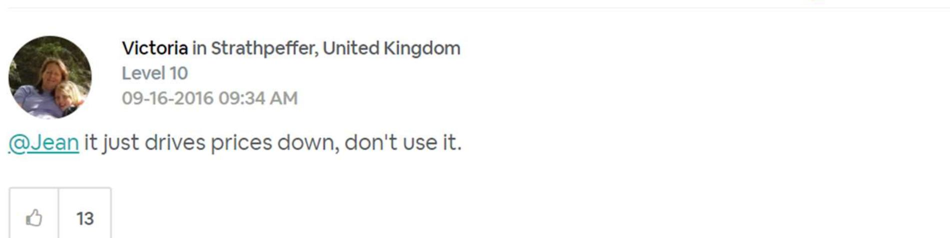
Alice & Jeff in Durham, NC
Level 10
09-16-2016 10:08 AM
We use it and love it. It has increased our booking price on average of 7% and we're booked 92% of the time. While I used to agonize over the price and if I would get a booking, now I let smart pricing do it fo

*Category 2*



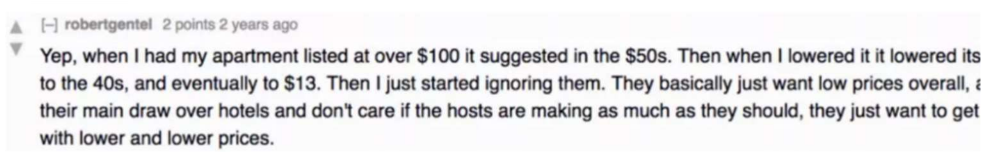
Kelly in Austin, TX
Level 10
11-13-2017 06:03 PM
@David Agreed. Ours is a 1b/1b guesthouse, 1300 sqft. The comparisons abb shows me are often 1b/1b private rooms. Not even remotely close!
Smart Pricing. Not so smart.
7

*Category 2*



Victoria in Strathpeffer, United Kingdom
Level 10
09-16-2016 09:34 AM
@Jean it just drives prices down, don't use it.
13

*Category 2*



[–] robertgentel 2 points 2 years ago
Yep, when I had my apartment listed at over $100 it suggested in the $50s. Then when I lowered it it lowered its to the 40s, and eventually to $13. Then I just started ignoring them. They basically just want low prices overall, their main draw over hotels and don't care if the hosts are making as much as they should, they just want to get with lower and lower prices.

Notes: Our review of algorithm-related discussions on the Airbnb Host Forum revealed that hosts' opinions fell into two primary categories, represented here with screenshots of anecdotes. Posts in *Category 1* favor the algorithm because it alleviates the time and effort required to monitor and adjust prices manually. Posts in *Category 2* do not favor the algorithm because it does not seem trusthworthy and because the recommended prices seem too low for the property's features.

We define several covariates that relate to the host's likelihood of falling into each category of thought:

Hosts whose opinions aligned with **Category 1** were more likely to adopt the algorithm because it eliminates the time and effort required to update prices manually. We represent Category

1 with variables that reflect the opportunity cost of time and the effort required to set prices. We account for the value of the host's time with two variables: $MedianEarning_{in}$ (the median household income in host *i*'s neighborhood *n*, conditioned on host *i*'s race and age) and $HomeValue_i$ (the average value of a home with the same size as host *i*'s property in neighborhood *n*). We collected income data from the American Community Survey[8] and home value data from Zillow Research[9]. We account for the effort required to set prices with three variables: the duration of the property's listing on Airbnb, whether the host owns multiple listings, and whether the host manages the entire place (as opposed to renting out a room). Gibbs et al. (2017) found that hosts who have more experience on Airbnb, own multiple listings, and manage their entire place are more adept at updating their prices over time and thus have a lower cost of doing so.

Hosts whose opinions aligned with **Category 2** were less likely to adopt the algorithm out of mistrust in the algorithm/Airbnb, a lack of trust in technology, or the experience of receiving prices that seemed too low. We account for the host's receptiveness toward new technology by including the host's age (predicted from the profile photo) and education at the neighborhood level ($Bachelor_{in}$ and $Graduate_{in}$ are the ratios of people in neighborhood *n* who have a bachelor's and master's degree, respectively, conditional on the race of host *i)*. We account for the low-price concern by including variables for all property features and amenities listed on the Airbnb property webpages. The algorithm accounts for standard features, but it may not incorporate unique features (Ye et al. 2018), so it is more likely to set an inappropriately low price for properties with more unique features.

### 2.5 AirDNA Data: Data Collection, Limitations, and Reliability

We used AirDNA to determine the average monthly price, monthly occupancy status, and other characteristics mentioned on the property webpages. To understand the potential concerns with the AirDNA data, we will first explain where we use this data in our analysis. We use the AirDNA data to calculate each property's average daily revenue in a month, which is the DV in our analysis. Recall that we computed the average daily revenue as

*Average Daily Revenue in month i = Average Nightly Price × Occupancy*

*Occupancy = # booked days in month i / (total # of days in month i - # blocked days in month i).*

The nightly rate is publicly available, and AirDNA scrapes this information from each host's webpage. Blocked days are days on which the host marks the property as unavailable for all guests; booked days are days on which guests rented/reserved the property. The information on daily prices is publicly available and AirDNA is able to scrape this information from each host's webpage. Thus the information on prices from AirDNA is accurate. Prior to December 2015, AirDNA used Airbnb's API to collect the numbers of booked and blocked days, but Airbnb disabled its API in December 2015. AirDNA could observe only the sum of blocked and booked days each month, so AirDNA built a proprietary ML algorithm to determine the number of blocked vs. booked days for each property in each month (i.e., to separate values of the two).

The above discussion brings out the potential concern with the AirDNA data, which is that the AirDNA's algorithm may not be accurate enough to predict which of the non-open days were

---

[8] https://www.socialexplorer.com/explore/tables. https://www.census.gov/programs-surveys/acs/.
[9] https://www.zillow.com/research/data/.

booked or blocked. We believe the algorithm is acceptably accurate for two reasons. *First*, AirDNA trained its ML algorithm on high-quality data from two sources[10]:

(i) Historical data containing the true reservation status, collected through Airbnb's API for each Airbnb rental property for 18 months prior to December 2015. This data consists of the daily true reservation status, daily prices and all other information on the property webpages.

(ii) An ongoing stream of data (since December 2015) on properties for which AirDNA can verify the true reservation status through collaborations with individual hosts and professional managers (about 650,000 properties in the US and Europe). These individual hosts and professional managers have provided Airbnb daily information on each property's true reservation status (i.e., whether it was unreserved, booked or blocked), daily prices and other property characteristics. They also have provided other useful insights about AirDNA and recommended behavioral variables to improve the ML algorithm.

Both data sources contain rich sets of explanatory variables (e.g., location, time, property characteristics, length of booking, booking lead time, historical performance) that are related to the true reservation status. Moreover, AirDNA's ML algorithm continues to learn and improve from the ongoing data stream.

*Second*, AirDNA reports that their algorithm is accurate within a 5% margin of error when tested on hold-out samples. As per AirDNA, their algorithm is consistently accurate within a 5% margin of error when they compare the predicted blocked vs. booked days rate in a month with those in the hold out sample (AirDNA randomly splits the joint data (i) and (ii) into training and hold out samples for their ML algorithm). We cautious the readers that AirDNA uses its proprietary algorithm to discriminate a booked day versus a blocked day for calculating the monthly occupancy rate. AirDNA claims that its algorithm is able to predict the monthly occupancy rate within 5% error. However since we do not have access to AirDNA's proprietary algorithm, we are not able to verify this claim.

In the AirDNA data, all the observations in the entire panel pertaining to the reservation status (that is, whether the property was booked, blocked or unreserved) were based on the predictions of AirDNA's algorithm. This is important because recall that AirDNA could observe the true reservation status prior to Dec 2015, but could only make predictions of it post Dec 2015. Since Dec 2015 lies within the span of our data (our data spans from July 2015 to Aug 2017), it is important that we maintain consistency throughout the entire panel on how reservation status is obtained. Additionally, based on our conversations with AirDNA, their algorithm has a consistent performance in terms of on the test set across time, including periods before and after December

[10] Please see https://www.airdna.co/blog/short-term-rental-data-methodology.

## Section 3. Inverse Probability Treatment Weighting (IPTW): Analysis, Variables, and Robustness Checks

IPTW is a widely-applied weighting method for constructing a balanced sample of treatment and control units (Rosenbaum 1987, Austin and Stuart 2015). The method first computes the weight of each unit as the inverse of its probability of receiving treatment and then weights all units to eliminate any systematic differences in the observed covariates between the treated and untreated groups that may explain the treatment probability. Unlike matching methods, which discard units that cannot be matched with similar units in the other group, the IPTW method creates a balanced sample without discarding any units (Guo and Fraser 2015). Monte Carlo studies have shown that the IPTW method leads to lower mean squared errors in the treatment effect estimates than matching methods such as Propensity Score Matching (PSM, see Austin 2013, Austin and Stuart 2015).

The treatment probability (also called the propensity score) is the probability that an individual unit is assigned to the treatment condition, conditional on a set of observed variables (Rosenbaum and Rubin 1983). We specify the propensity score of property $i$ as a function of $\boldsymbol{X_i}$, a set of $K$-dimensional observed covariates that impact the host of property $i$'s decision whether to adopt the algorithm (thereby self-selecting into the treatment condition). $\boldsymbol{X_i}$ also includes all observed confounders that impact the adoption decision and are also correlated with the DV in the subsequent DiD regressions:

$$\widehat{ps}_i = f(\boldsymbol{X_i \beta}) \tag{S3}$$

$\boldsymbol{X_i}$ includes both time-invariant and time-variant observables. For the time-variant observables, we use the values immediately prior to treatment (i.e., at the beginning of November 2015), following Austin (2011). To calculate the propensity score $\widehat{ps}_i$, we first estimate the parameter vector $\boldsymbol{\beta}$ by fitting a logistic regression where the input is the vector of observed covariates $\boldsymbol{X_i}$ and the output is a binary response that equals 1 if unit $i$ received treatment, and 0 otherwise. The estimated $\boldsymbol{\beta}$ maximizes the data likelihood of the observed treatment assignment (Rosenbaum and Rubin 1983), where $f$ (.) takes a logit functional form. Then, we compute the propensity score according to Equation S3, and we calculate the weight of unit $i$ as follows:

$$w_i = \frac{T}{ps_i} + \frac{1-T}{1-ps_i} \tag{S4}$$

$T$ is the binary treatment indicator (i.e., $T = 1$ if unit $i$ belongs to the treatment group). Weight $w_i$ captures the contribution of unit $i$ to the average treatment effect in the DiD regression, which uses the weighted least squares. By weighting unit $i$'s observations, IPTW creates a synthetic sample of weighted treatment and control groups in which the covariates X are balanced, thereby ensuring that the treatment assignment across the two weighted groups is independent of any observed confounders. In other words, if no unobserved confounders contributed to the treatment assignment, then the balanced sample is effectively randomized, and the estimated treatment effect is unbiased by self-selection.

### 3.1. Estimation of the Treatment Probability

We include many covariates to improve the performance of the IPTW method and reduce potential bias due to omitted variables in the subsequent DiD regression. We used a broad list of observed covariates, ***X***, that are available to us for estimating the probability of treatment assignment of each property. These variables include the ones that that we discussed in section 2.4 of the Web Appendix. Table S3 lists the observed covariates and estimated treatment probabilities, calculated from a logit function.

**Table S3 List of Variables and Estimation Results of Treatment Probability.**

| VARIABLE | Estimate | Std. Err. | p-value |
|---|---|---|---|
| # Bedrooms | -0.08686 | 0.041818 | 0.038 |
| Apartment | -0.19591 | 0.068849 | 0.004 |
| Entire Home | -0.32363 | 0.064367 | 0 |
| Listing Title Length | 0.013507 | 0.003936 | 0.001 |
| Number of Photos | 0.006519 | 0.002465 | 0.008 |
| Number of Reviews | 0.000255 | 0.000941 | 0.787 |
| Listing Nightly Rate | 5.69E-05 | 0.000242 | 0.814 |
| # Minimum Stay | -0.0035 | 0.01127 | 0.756 |
| Security Deposit | 1.64E-05 | 8.68E-05 | 0.85 |
| # Blocked Days in a Month | -0.02426 | 0.002881 | 0 |
| # Reservation Days | 0.003233 | 0.003618 | 0.372 |
| Median Home Earning (*1000 USD*) | 0.002108 | 0.002423 | 0.384 |
| Private Parking | 0.164609 | 0.167851 | 0.327 |
| Pool | 0.109914 | 0.126473 | 0.385 |
| Iron | -0.10366 | 0.212219 | 0.625 |
| Internet | 0.105076 | 0.290149 | 0.717 |
| TV | -0.11238 | 0.065056 | 0.084 |
| Dryer | 0.094626 | 0.096851 | 0.329 |
| Washer | -0.15115 | 0.077124 | 0.05 |
| Beach Nearby | -0.496169 | 0.095586 | 0 |
| Essentials | 0.338564 | 0.11026 | 0.002 |
| Heating | 0.413384 | 0.142156 | 0.004 |
| Microwave | 0.04468 | 0.163114 | 0.784 |
| Refrigerator | 0.184393 | 0.158359 | 0.244 |
| Laptop Friendly | 0.14706 | 0.088842 | 0.098 |
| Fireplace | -0.16589 | 0.080206 | 0.039 |
| Elevator | -0.1523 | 0.076067 | 0.045 |
| Gym | -0.13081 | 0.127532 | 0.305 |
| Family Friendly | 0.217464 | 0.081042 | 0.007 |
| Smoker Detector | 0.336256 | 0.100669 | 0.001 |
| Shampoo | -0.03363 | 0.084676 | 0.691 |
| Breakfast | 0.035317 | 0.106263 | 0.74 |

| | | | |
|---|---|---|---|
| AC | 0.041163 | 0.563674 | 0.942 |
| # Photographed Faces | 0.016342 | 0.035888 | 0.649 |
| Walk Score | -0.00271 | 0.001611 | 0.092 |
| Transit Score | 0.008991 | 0.002413 | 0 |
| Drive to Downtown (*min*) | -0.00178 | 0.003112 | 0.568 |
| Population Density (*Per Sq. Mile*) | 1.58E-06 | 1.44E-06 | 0.271 |
| Graduate (%) | 0.0316 | 0.008379 | 0 |
| Bachelor (%) | 0.01114 | 0.004725 | 0.018 |
| Host Age | -0.00444 | 0.002748 | 0.106 |
| Home Value (1000 USD) | -7.4E-05 | 7.67E-05 | 0.336 |
| Number of months since the property has been listed | -0.00475 | 0.002028 | 0.019 |
| Number of properties owned by the host | -0.03667 | 0.007686 | 0 |
| Observations | 9396 | | |
| Log likelihood | -4610.69 | | |

Note: The table reports the estimated propensity scores according to the IPTW method for our sample of 9,396 Airbnb properties. The dependent variable is *Smart Pricing*, a binary variable that equals 1 (vs. 0) if the property's host adopted (vs. did not adopt) Smart Pricing during our observational window (i.e., by August 2017). The independent variables are the observed property, host, and neighborhood characteristics. Time-varying characteristics are measured in the pre-treatment period one month prior to the introduction of Smart Pricing).

**3.2. Validation of IPTW: Covariates Balance Assessment**

With the estimated values of $\boldsymbol{\beta}$, we use Equation (S3) to compute the weights of the sub-sample of the observations of each property $i$. We validate the success of the IPTW method by evaluating whether the observed covariates are similar in the weighted treatment and control groups. Following prior literature (Rubin 2001, Stuart 2010), we compute the standardized difference, $d^m$, in the means of the covariates between the two groups (i.e., $\overline{\boldsymbol{X}}_{\boldsymbol{treatment}}$ and, $\overline{\boldsymbol{X}}_{\boldsymbol{control}}$) over *M*-dimensional covariates, and we normalize the difference by the sample variance, $s^2_{treatmen}$ and $s^2_{control}$:

$$d^m = \left| \frac{\bar{X}^m_{treatment} - \bar{X}^m_{control}}{\sqrt{\frac{s^2_{treatment} + s^2_{control}}{2}}} \right| \quad \text{(S5)}$$

We calculate the above statistic for each observed variable *m* in the two weighted samples. If $d^m$ is small, then it implies that the two groups are comparable in that observed variable. The means and variances are weighted by $\omega_i$, the inverse of the treatment probability: that is, $\omega_i = \frac{1}{\widehat{ps_i}}$ in the treatment group and $\omega_i = \frac{1}{1-\widehat{ps_i}}$ in the control group. The means and variances are calculated as follows:

$$\begin{cases} \bar{\boldsymbol{X}}_{\boldsymbol{treatment}} = \dfrac{\sum_{i \epsilon treatment} \omega_i \boldsymbol{X_i}}{\sum_{i \epsilon treatment} \omega_i} & \\ s^2_{treatment} = \dfrac{\sum_i \omega_i}{\left(\sum_i \omega_i\right)^2 - \sum_i (\omega_i)^2} \sum_i \omega_i (X_i^m - \bar{X}^m_{treatment})^2 & \end{cases} \quad i \ in \ treatment \ group \tag{S6}$$

$$\begin{cases} \bar{\boldsymbol{X}}_{\boldsymbol{control}} = \dfrac{\sum_{i \epsilon control} \omega_i \boldsymbol{X_i}}{\sum_{i \epsilon control} \omega_i} & \\ s^2_{control} = \dfrac{\sum_i \omega_i}{\left(\sum_i \omega_i\right)^2 - \sum_i (\omega_i)^2} \sum_i \omega_i (X_i^m - \bar{X}^m_{control})^2 & \end{cases} \quad i \ in \ control \ group \tag{S7}$$

If Equation (S5) yields a small $d^m$ (< 0.1, indicating an absolute standardized difference < 10%), then the difference in covariate *m* between the two weighted samples is considered negligible (see discussion in Austin and Stuart 2015). Table S4 shows that $d^m$ is less than 0.1 for all covariates in the weighted sample. We conclude that the IPTW method successfully removed all significant imbalances in the observed covariates that may have affected the treatment probability, thereby mitigating the self-selection issue.

**Table S4 Validation t-Test for IPTW: Covariates Balance Check.**

| | Standardized Difference | |
|---|---|---|
| | **(1)** | **(2)** |
| **VARIABLE** | **Unweighted Sample** | **Weighted Sample** |
| # Bedrooms | -0.034 | -0.003 |
| Apartment | -0.167 | 0.005 |
| Entire Home | -0.139 | -0.006 |
| Listing Title Length | 0.152 | 0 |
| Number of Reviews | 0.193 | 0.004 |
| Number of Photos | 0.153 | -0.011 |
| Listing Nightly Rate | -0.108 | 0.025 |
| # Minimum Stay | -0.028 | 0.015 |
| Security Deposit | 0.01 | -0.001 |
| # Blocked Days | -0.229 | 0.077 |
| # Reservation Days | 0.216 | -0.019 |
| Occupancy Rate | 0.12 | -0.006 |
| Median Home Earning (*1000 USD*) | -0.143 | 0.014 |
| Parking | 0.275 | -0.032 |
| Pool | -0.038 | -0.001 |
| Beach | 0.08 | -0.007 |
| Internet | 0.059 | -0.021 |
| TV | -0.02 | -0.014 |
| Dryer | 0.231 | -0.033 |

| | | |
|---|---|---|
| Washer | -0.027 | -0.009 |
| Iron | 0.529 | -0.043 |
| Essentials | 0.518 | -0.029 |
| Heating | 0.122 | -0.057 |
| Microwave | 0.322 | 0.001 |
| Refrigerator | 0.354 | -0.002 |
| Laptop Friendly | 0.461 | -0.044 |
| Fireplace | 0.002 | -0.007 |
| Elevator | -0.137 | -0.012 |
| Gym | -0.095 | -0.005 |
| Family Friendly | -0.211 | -0.006 |
| Smoker Detector | 0.512 | -0.035 |
| Shampoo | 0.452 | -0.028 |
| Breakfast | 0.053 | 0.006 |
| AC | 0.063 | -0.052 |
| # Photographed Faces | 0.024 | -0.03 |
| Walk Score | -0.052 | 0.018 |
| Transit Score | -0.038 | 0.017 |
| Drive to Downtown (*min*) | 0.063 | 0.004 |
| Population Density (*Per Sq. Mile*) | -0.052 | 0.008 |
| Bachelor (%) | -0.154 | 0.021 |
| Professional Host (#host-owned listings) | -0.109 | -0.028 |
| # Listed Month | 0.031 | 0.001 |
| Host Age | 0.038 | -0.022 |
| Home Value (1000 USD) | -0.087 | 0.05 |

Notes: The table reports the standardized difference between each covariate in the treatment and control groups in the unweighted (i.e., raw) sample (Column (1)) and the weighted sample (i.e., the result of the IPTW method; Column (2)) based on values in the pre-treatment period. Standardized differences were computed using Equation (S5). The absolute standardized difference is < 0.1 for all covariates in Column (2), confirming the success of the IPTW method.

**3.3. Addressing Unobservables: Sensitivity Analysis with the Conditional c-Dependence Test**

The unbiasedness of the treatment effect rests on the conditional independence assumption (CIA): that potential outcome *Y* is independent of treatment assignment *T,* conditional on the distributions of the observed covariates *f(X)*. Though the weighted treatment and control samples are comparable along the list observed covariates, we may be missing other relevant but unobserved variables that were not included in the propensity score calculation. This is because the treatment probabilities (propensity scores) are computed as a function of observables. If there are unobservables that affect hosts' adoption decision, then this may introduce a bias in the estimated treatment effect as the unobservables could be correlated with the dependent variable in the subsequent regression.

To deal with the issue of unobserved confounders, researchers have proposed methods to assess the sensitivity of the estimator to CIA. We chose the conditional c-dependence sensitivity test recently proposed by Masten and Poirier (2018, 2019, and 2020), which does not impose the parametric assumption on the modeling of the selection process or on the impact of treatment on the outcome variable. We acknowledge that researchers historically have used a Rosenbaum bounds sensitivity analysis, but we chose the newer test for two reasons: First, the Rosenbaum sensitivity analysis must be applied in the setting of PSM, not IPTW. Second, it tests the robustness of the average treatment effect (i.e., the effect of algorithm adoption on average daily revenue) but not the differential effect (i.e., variation in the average treatment effect by race), which is central to our research questions.

### 3.3.1. Overview of the Conditional c-Dependence Sensitivity Analysis

We followed the work of *conditional c-dependence* proposed by Masten and Poirier (2018). The objective is to determine the threshold above which the impact of hypothetical unobserved confounders would nullify the treatment effect—the higher the threshold, the greater our confidence that our estimated treatment effect is robust to unobserved confounders.

Recall that our IPTW and DiD analyses estimated the treatment effects based on the conditional independence assumption (CIA): that the probability of being selected for treatment conditional on the observed covariates (X) and unobserved potential outcome variable is equal to the probability of treatment conditional on the observed covariates only. The CIA holds as long as there are no unobserved confounders.

By contrast, the conditional c-dependence or partial independence assumption proposed by Masten and Poirier (2018) is weaker than the CIA: that under a hypothetical scenario in which the CIA is violated, the two aforementioned conditional probabilities are not identical but should be within a measurable distance from each other. Under conditional c-dependence, the deviation from the CIA, as captured by the maximum distance between the two aforementioned conditional probabilities, is described by a scalar variable $c$. If the CIA is true, then the two conditional probabilities are equal, and $c$ = zero; otherwise, there is a non-zero distance between the two conditional probabilities (i.e., $c > 0$). The distance $c$ captures the impact of unobserved confounders. The greater the magnitude of $c$, the greater the impact of unobserved confounders, and the greater the violation of the CIA.

The objective of the conditional $c$-dependence exercise is to calculate the "breakdown value": the minimum value of $c$ that would nullify our treatment effect (estimated under the CIA). Intuitively, a large (small) breakdown value implies that our treatment effect that we had obtained by assuming CIA is quite robust (quite insensitive) to hypothetical unobserved confounders: it requires a large value of $c$ to nullify the estimated treatment effect.

The conditional c-dependence is defined in the following framework:

- $Y_T$*:* the potential outcome for a given treatment $T \in \{0, 1\}$
- $T$: binary treatment status; $T = 1$ if treated (i.e., the property's host adopted the algorithm) and $T = 0$ if not
- $X$: the set of observed covariates used in IPTW
- $Y$: the observed outcome

The observed outcome is represented by $Y = \mathbf{1}\{T = 1\}Y_1 + \mathbf{1}\{T = 0\}Y_0$.

Under the CIA, we have:

$$P(T = 1|Y_T = y_T, X = x) = P(T = 1|X = x) \quad \text{(S8)}$$

where $P(\cdot)$ indicates the conditional treatment probability, $y_T \in$ support of $(Y_T|X = x)$ and $x \in$ support of $(W)$.

Under conditional c-dependence, the independence only partially holds true:

$$\sup_{y_T \in supp(Y_T|X=x)} |P(T = 1|Y_T = y_T, X = x) - P(T = 1|X = x)| \leq c \quad \text{(S9)}$$

where the sup-norm distance is the difference between the two treatment probabilities. The scalar $c$ has a value between 0 and 1, where 0 refers to the full CIA as a special case. When $c = 0$, the treatment effect based on the IPTW and DiD analyses is point-estimated; when $c > 0$, the treatment effect is defined as a range for a given significance level. The upper and lower bound around the estimated treatment effect reflect the uncertainty in the estimate of the treatment effect due to unobserved confounders. Masten and Poirier (2018) showed that the values of the lower- and upper- bounds for a given level of significance can be computed as functions of $c$. These bounds collapse to the same point estimate when $c$=0, i.e., when CIA holds true. The bounds get 'wider' as $c$ increases, i.e., when the hypothetical unobservable leads to a deviation from CIA.

The breakdown point, $c_b$, is defined as the minimum value of $c$ at which the data disproves the conjecture about the treatment at the 95% significance level. When our conjectured treatment effect is positive, $c_b$ is the point at which the lower bound curve intersects the baseline of zero (x-axis); when our conjectured treatment effect is negative, $c_b$ is the point at which the upper bound curve intersects the baseline of zero. Once we get the value of $c_b$, we can then make inferences on how robust our estimated treatment effect is to hypothetical unobserved confounders (more on this in section 3.3.4).

In our context, the treatment effects of interest are the average treatment effect (the impact of algorithm adoption on the average daily revenue; Section 3.3.2) and the differential effect (the differential impact of algorithm adoption on the average daily revenue of Black vs. white hosts; Section 3.3.3). The value of $c_b$ indicates the robustness of our estimated treatment effect to hypothetical unobserved confounders (Section 3.3.4).

### 3.3.2. Conditional c-Dependence Sensitivity Analysis: The Average Treatment Effect

We present results on the upper and lower bounds and value of $c_b$ based on a 95% significance level for the average treatment effect. We used the Stata package, *tesensitivity*, developed by the authors of conditional c-dependence (Masten and Poirier 2019), and we implemented the package with the command *teffects* for estimating the treatment effect using IPTW[11]. The formulas, theoretical derivations, and proofs are in Theorem 1 (Proposition 3-Corllary 1 and Proposition 4 of Masten and Poirier 2018).

The upper and lower bounds have the same value when $c = 0$ (i.e., the treatment effect is point-identified under the CIA), and the bounds widen as $c$ increases. As shown in Figure S2 and Table S5 (which contains the first 10 values of $c$), the breakdown point $c_b = 0.083$ for the average treatment effect (i.e., the closed interval on the estimated treatment effect would include 0 above the value of 0.083)[12]. In other words, our estimated average treatment effect would be nullified if

[11] StataCorp. 2013. Stata: Release 13. Statistical Software. College Station, TX: StataCorp LP. https://www.stata.com/manuals13/te.pdf; https://github.com/mattmasten/tesensitivity

[12] Note that this breakdown value $c_b$ was reported as an output by the software *tesensitivity*.

unobservables cause a deviation of 0.083 or more in the adoption probability (see Section 3.3.4 for the full interpretation).

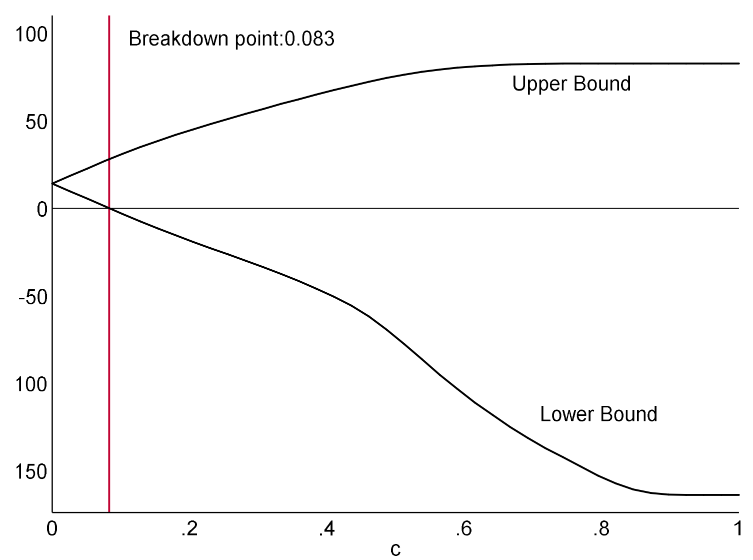
In section 3.3.4, we will benchmark the breakdown point in terms of the results obtained in prior studies that have used conditional c-dependence and other sensitivity analyses. That will give a better picture of how robust our results are to hypothetical unobserved confounders.

**Table S5 C*onditional c-Dependence* Bounds for the Average Treatment Effect**

| Analysis: conditional c-dependence | | |
|---|---|---|
| **c** | **Upper Bound** | **Lower Bound** |
| 0 | 14.13 | 14.13 |
| 0.026 | 9.77 | 18.47 |
| 0.051 | 5.39 | 22.77 |
| **0.077** | **1.00** | **27.00** |
| **0.103** | **-3.32** | **31.08** |
| 0.128 | -7.51 | 34.91 |
| 0.154 | -11.54 | 38.49 |
| 0.179 | -15.42 | 41.86 |
| 0.205 | -19.17 | 45.06 |
| 0.231 | -22.82 | 48.13 |

Notes: Each row reports the range, captured by the upper and lower bounds, of the estimated effect of algorithm adoption on average daily revenue. The degree of violation of the CIA is captured by *c*, a value between 0 and 1, where 0 refers to the full CIA as a special case. The two rows in boldface font indicate the intersection of the range with zero, where the estimated treatment effect under the assumption of $c = 0$ is nullified. The precise breakdown point, $c_b$, is 0.083.

**Figure S2 C*onditional c-Dependence* Bounds for the Average Treatment Effect: Bounds vary over the value of c**



### 3.3.3. Conditional c-Dependence Sensitivity Analysis: The Differential Effect[13]

We conducted the conditional c-dependence analysis separately for white and Black hosts and determined the value of $c$ at which the estimated treatment effects started to overlap—that is, the point at which there is no longer a significant difference in the treatment effect conditioned on race.

Formally, we are interested in the difference, $\Delta\beta$, between the treatment effects for white and Black hosts:

$$\Delta\beta = \beta_{Black} - \beta_{whit} \tag{S10}$$

where $\beta_{Black}$ and $\beta_{white}$ are the treatment effects of algorithm adoption on the property demand for black host for each racial group. We construct the upper bound (UB) and lower bound (LB):

$$UB_{\Delta\beta} = UB_{\beta_{Black}} - LB_{\beta_{white}} \tag{S11}$$

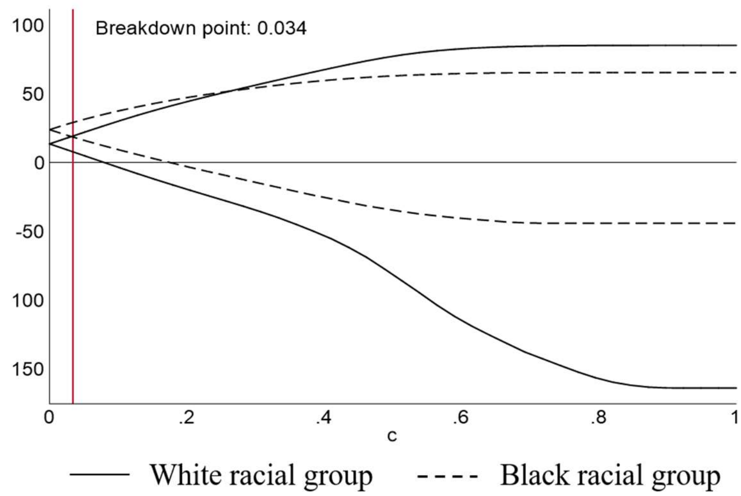
$$LB_{\Delta\beta} = LB_{\beta_{Black}} - UB_{\beta_{white}} \tag{S12}$$

Then, we plot the set of bounds, ($UB_{\beta_{Black}}$, $LB_{\beta_{Black}}$, $UB_{\beta_{white}}$, $LB_{\beta_{white}}$), as a function of $c$. Note that we are interested in testing $\beta_{Black} > \beta_{white}$, so $c_b$ is defined as the first intersection of $LB_{\beta_{Black}}$ and $UB_{\beta_{white}}$. The value of $c_b$ would give us the smallest 'amount of unobserved selection' required for $\Delta\beta$ to be zero, and that will be the breakdown point for the differential effect. We interpret the sensitivity results as conservative estimates because the comparison of $LB_{\beta_{Black}}$ and $UB_{\beta_{white}}$ is a conservative comparison; it assumes that the data aligns such that the difference, $\Delta\beta = \beta_{Black} - \beta_{white}$, is as small as possible.

In Figure S3, we plot the bounds of the estimated treatment effects for white and Black hosts. The difference between the effects diminishes as $c$ increases. The value of the breakdown point, $c_b$, is 0.034, so our estimated differential effect of the pricing algorithm would be nullified if unobservables cause a deviation of 0.034 or more in the adoption probability (see Section 3.3.4 for the full interpretation).

**Figure S3 *Conditional c-Dependence* Bounds for the Average Treatment Effects Among White and Black Hosts, Separately**

[13] We thank Matthew Masten (author of Masten and Poirier 2018, 2019) for suggesting the methodology for calculating the breakdown value of the differential effect.




### 3.3.4. Interpretation of the Conditional c-Dependence Sensitivity Analyses

We calculated a breakdown point of 0.083 for the average treatment effect (Section 3.3.2) and 0.034 for the differential effect (Section 3.3.3). The mean treatment (i.e., adoption) probability in our sample is 19.5%. The estimated average treatment effect of the pricing algorithm would be nullified if unobservables caused a deviation in the adoption decision of at least 42.6% (0.083/0.195). The estimated differential effect would be nullified if unobservables caused a deviation in the adoption decision of at least 17.4% (0.034/0.195).

We benchmark these breakdown values with results from prior studies that used conditional c-dependence analysis and the Rosenbaum bounds analysis (a different sensitivity analysis that often is used with PSM). We also conduct a "leave-one-out" analysis to aid the interpretation of our values.

1. *Benchmarking with results from prior conditional c-dependence analyses:*

The conditional-dependence analysis is a recent development with only a few applications. Masten and Poirier (2018, 2019) performed the analysis on the well-known *Lalonde 1986* dataset, which has been used to measure the causal effect of education on income. The dataset comprises two samples: a randomized experimental sample that was used by LaLonde (1986) and a reconstructed non-experimental matched sample that was used by Dehejia and Wahba (1999). In the reconstructed sample, Dehejia and Wahba addressed confounders adequately enough to calculate an estimated treatment effect that was similar to the effect in the randomized experimental sample.

Masten and Poirier (2019) performed the conditional c-dependence analysis on these two samples, and found breakdown values of 0.075 for the randomized experimental sample and 0.020 for the non-experimental matched sample. The breakdown value of 0.020 for the non-experimental matched sample serves as a useful benchmark because Dehejia and Wahba (1999) had demonstrated that the non-experimental matched sample is successful in dealing with confounders and it yields estimates of the treatment effect that are very similar to the ones estimated from the randomized experimental sample. Since our breakdown points are larger than 0.020, we conclude that both of our effects are fairly robust to unobserved confounders. We are especially confident in the robustness of the differential effect because we structured our analysis to yield a conservative estimate (assuming that the data lines up to make the difference $\Delta\beta = \beta_{Black} - \beta_{white}$ as small as possible, see details in Section 3.3.3).

*2. Benchmarking with results from prior Rosenbaum bounds analyses:*[14]

The Rosenbaum bounds analysis is a sensitivity test for assessing the robustness of the estimate of the treatments effect (that was obtained using PSM) to hypothetical unobserved factors that enter the treatment selection process (Rosenbaum 2002). While the conditional c-dependence analysis captures the impact of unobservables as the absolute distance between the conditional probabilities (as in Equation (S9)), the Rosenbaum bounds analysis captures the impact as the change in the odds ratio of treatment between two units that are otherwise similar on observables. Instead of a breakdown value, the Rosenbaum bounds analysis outputs the Gamma value: the minimum change in the odds ratio that would nullify the treatment effect.

To compare the robustness measures across conditional c-dependence analyses and Rosenbaum bounds analyses, we need to translate the breakdown value into the change in the odds ratio. Supposing that hypothetical unobservables that cause a deviation of $c_b$ (the breakdown value) in the adoption probability, we compute the altered adoption probability under the impact of the hypothetical unobservables. Then, we calculate the odds ratio of adoption in our sample and the odds ratio based on the altered adoption probability. The change in the odds ratios is the Gamma value.

In our sample, the mean adoption probability is 19.5%, so the odds ratio is 19.5%/(1 - 19.5%)) = 0.24. Given $c_b$ = 0.083 for the average treatment effect, the altered odds ratio is (19.5% + 8.3%) / [1 - (19.5% + 8.3%)] = 0.39. We calculate the Gamma value as the change in the odds ratio: 0.39/0.24 =1.63. We perform the analogous calculations for the differential effect and find that $c_b$= 0.034 translates into a Gamma value of 1.25. In other words, our estimated average treatment (differential) effect at the 95% significance level would be nullified by unobserved confounders that make the Airbnb hosts in our sample at least 63% (25%) more likely to adopt the algorithm. Our Gamma values for both effects align with the range of 1.2 to 1.6 in the literature (e.g., DiPrete et al. 2004, Sun and Zhu 2014, Manchanda et al. 2015). Again, we conclude that our results are fairly robust to hypothetical unobserved confounders.

*3. Interpreting the results using a "leave-one-out" analysis*

We use a "leave-one-out" analysis (Masten and Poirier 2018, 2020) to interpret the breakdown values of the average treatment and differential effects in terms of the impact of known covariates. Specifically, we identify known covariates that affect the adoption probability by the magnitude indicated by the breakdown value (8.3% and 3.4% for the average treatment and differential effects, respectively).

In the leave-one-out analysis, for each covariate $X^m$ in the adoption probability model, we obtain two estimated adoption probabilities: one that includes all the covariates that we used in the adoption probability model, and the other that leaves out covariate $X^m$ from the adoption probability model. A comparison of the two probabilities for each covariate $X^m \in \{X^1..X^M\}$ reveals the impact of the covariate on the adoption probability. Then, we identify the covariates with an impact greater than the impact implied by the breakdown value, and also identify the specific

[14] We chose to compare robustness measures across two different methodologies because very few studies have used the conditional c-dependence analysis. We concluded that the next best option was a comparison with the Rosenbaum bounds analysis, which has been used by many past studies. Based on our correspondence with Matthew Masten (author of Masten and Poirier 2018), the results of the two methodologies are similar enough to yield almost identical results with a one-on-one matched sample based on PSM.

covariates whose impact is the same as what the breakdown values of the treatment effect and the differential effect would imply.

Formally, suppose our main analysis included all covariates $\{X^m\}_{m=1}^M$, and we computed the adoption probability for each property $i$ as a function of the covariates: $p_i = f(X_i^1, X_i^2, \dots, X_i^M)$. The leave-one-out analysis computes *M* sets of alternative adoption probabilities. For each $m^{\text{th}}$ covariate excluded from the estimation, the leave-one-out analysis estimates $p_i^{-m} = f(X_i^1, X_i^2, \dots, X_i^{m-1}, X_i^{m=1}, \dots, X_i^M)$, where *-m* indicates that covariate $X^m$ was not used in estimating the adoption probability. The deviation of $p_i^{-m}$ from $p_i$ tells us the impact of excluding $X^m$ on the adoption probability of host *i*. The deviation is a point value for each individual host *i* and a distribution for the full sample (since the value of $X^m$ is itself randomly distributed across properties in the sample). We run the analysis with the Stata package *tesensitivity*. By default, the package reports the maximum (i.e., the supremum over the distribution) and the 50th, 75th, and 90th percentiles of the distribution for each covariate. We report all three percentiles as well as the maximum value in Table S6.

At the 50th percentile (column titled "0.5 (Median)"), none of the covariates are associated with a deviation in the adoption probability of 0.083 or more (i.e., the breakdown value of the average treatment effect). If we consider the maximum possible deviation (column titled "max"), 13 of the 44 covariates rank at or above 0.083: *# Bedrooms, Entire Home, Listing Title Length, Iron, Essentials, Smoker detector, Transit Score, Number of Photos, Security Deposit, # Blocked Days, # Host-owned Listings, # Listed Month, Listing Nightly Rate*. The breakdown value of 0.083 is closest to the maximum deviation caused by # Bedrooms (0.082). In other words, our estimated average treatment effect would be nullified if hypothetical unobserved confounders affected the adoption probability by about the same magnitude as the maximum impact of property size (in terms of # Bedrooms).

We do the same for the differential effect. Again, at the 50th percentile, none of the covariates are associated with a deviation of 0.034 or more (i.e., the breakdown value of the differential effect). If we consider the maximum possible deviation, the breakdown value is still greater than the maximum impact of 14 of the 44 observed covariates: *Pool, Internet, TV, Dryer, Microwave, Laptop friendly, Fireplace, Gym, Shampoo, Breakfast, Drive to Downtown , Population Density, Median Home Earning, Home Value*. The breakdown value of 0.034 is closest to the maximum deviation caused by # Reservation Days (0.036). In other words, our estimated differential effect would be nullified if hypothetical unobserved confounders affected the adoption probability by about the same magnitude as the maximum impact of the number of reservation days.

**Table S6 Interpreting the Sensitivity Analysis: "Leave-One-Out Analysis"**

| **Difference in the Propensity Score without the Covariate** | | | | |
|---|---|---|---|---|
| **"Left-Out" Covariate** | **Quantile in the Distribution of Propensity Score Difference** | | | |
| | **0.5 (Median)** | 0.75 | 0.9 | **max** |
| # Bedrooms | 0.003 | 0.007 | 0.013 | 0.082 |
| Apartment | 0.007 | 0.012 | 0.019 | 0.041 |
| Entire Home | 0.017 | 0.026 | 0.037 | 0.155 |
| Listing Title Length | 0.005 | 0.012 | 0.028 | 0.182 |
| Parking | 0.008 | 0.015 | 0.022 | 0.056 |
| Pool | 0.001 | 0.002 | 0.005 | 0.035 |
| Beach | 0.001 | 0.001 | 0.003 | 0.048 |

| | | | | |
|---|---|---|---|---|
| Internet | 0 | 0 | 0 | 0.005 |
| TV | 0.004 | 0.007 | 0.011 | 0.024 |
| Dryer | 0.002 | 0.003 | 0.004 | 0.016 |
| Washer | 0.004 | 0.01 | 0.018 | 0.038 |
| Iron | 0.007 | 0.017 | 0.043 | 0.119 |
| Essentials | 0.005 | 0.008 | 0.023 | 0.087 |
| Heating | 0.002 | 0.003 | 0.004 | 0.058 |
| Microwave | 0 | 0.001 | 0.004 | 0.03 |
| Refrigerator | 0.001 | 0.002 | 0.004 | 0.042 |
| Laptop Friendly | 0.001 | 0.004 | 0.01 | 0.027 |
| Fireplace | 0.002 | 0.005 | 0.011 | 0.033 |
| Elevator | 0.004 | 0.008 | 0.014 | 0.043 |
| Gym | 0.001 | 0.002 | 0.006 | 0.033 |
| Family Friendly | 0.006 | 0.011 | 0.015 | 0.054 |
| Smoker Detector | 0.004 | 0.008 | 0.021 | 0.096 |
| Shampoo | 0.001 | 0.002 | 0.004 | 0.01 |
| Breakfast | 0 | 0 | 0 | 0.001 |
| AC | 0 | 0.001 | 0.001 | 0.058 |
| # Photographed Faces | 0.001 | 0.002 | 0.003 | 0.04 |
| Walk Score | 0.002 | 0.005 | 0.01 | 0.058 |
| Transit Score | 0.01 | 0.02 | 0.032 | 0.108 |
| Drive to Downtown (*min*) | 0 | 0 | 0.001 | 0.003 |
| Population Density (*Per Sq. Mile*) | 0.001 | 0.002 | 0.003 | 0.011 |
| Graduate (%) | 0.001 | 0.003 | 0.006 | 0.033 |
| Bachelor (%) | 0.006 | 0.011 | 0.018 | 0.06 |
| Median Home Earning (1000 USD) | 0.001 | 0.002 | 0.003 | 0.027 |
| Host Age | 0.004 | 0.007 | 0.012 | 0.055 |
| Number of Photos | 0.005 | 0.01 | 0.016 | 0.269 |
| Number of Reviews | 0.001 | 0.003 | 0.006 | 0.06 |
| # Minimum Stay | 0.001 | 0.002 | 0.003 | 0.043 |
| Security Deposit | 0.001 | 0.003 | 0.005 | 0.097 |
| # Blocked Days | 0.009 | 0.017 | 0.028 | 0.086 |
| # Reservation Days | 0.004 | 0.006 | 0.01 | 0.036 |
| Home Value (1000 USD) | 0.001 | 0.001 | 0.002 | 0.031 |
| # Host-Owned Listings | 0.008 | 0.015 | 0.026 | 0.305 |
| # Listed Month | 0.008 | 0.014 | 0.022 | 0.089 |
| Listing Nightly Rate | 0.001 | 0.002 | 0.003 | 0.086 |

Notes: The values in each row represent the deviation in the estimated propensity score when the covariate is omitted from the model relative to the full model (i.e., with all observed covariates included, as shown in Table S3). When a variable is 'omitted' (i.e, a *partial model)* from the propensity score model (i.e., a *full model*), then the estimated propensity scores will be different from the results obtained from the full model. Each row in this table presents the deviations in the estimated propensity scores, when that variable is omitted. The deviations fall across a distribution for each omitted variable, so the columns report the 50th, 75th, and 90th percentiles as well as the maximum (i.e., the supremum over the distribution). As in Table S3, the logistic regressions are conducted on a set of 9, 396 unique properties and measure time-varying characteristics in pre-treatment period.

## Section 4. DiD Regressions: Analysis and Variables

We implement DiD analyses on the constructed weighted sample to estimate the effect of algorithm adoption on the average daily revenue, average nightly price, and monthly occupancy rate. We include property fixed effects, city-month fixed effects, and city-year fixed effects. We model the outcome variable (average daily revenue, average nightly rate, or monthly occupancy rate) for property *i* in period *t*:

$$Y_{it} = Property_i + \beta \cdot SmartPricing_{it} + \lambda \cdot Controls_{it} + Seasonality_{it} + \varepsilon_{it} \quad \text{(S13)}$$

$\varepsilon_{it}$ is idiosyncratic shock in $Y_{it}$. The treatment variable, $SmartPricing_{it}$, equals 1 (0) if the nightly rate of property $i$ in period $t$ was (was not) set by the pricing algorithm. The DiD estimator $\beta$ identifies how algorithm adoption affected the economic outcome in question.

Property fixed effects, $Property_i$, captures time-invariant covariates (e.g., property size, location, host demographics, neighborhood demographics, unobserved property quality). Seasonality fixed effects, $Seasonality_t$, captures city-year and city-month fixed effects that impact Airbnb property's revenue. $Controls_{it}$ represents time-varying covariates, which are listed in Table S7. As discussed in the paper, algorithm adoption rests primarily on time-invariant factors, so the inclusion of property fixed effects should control for most confounding factors.

**List of variables in DiD**

The list of time varying control variable include: property's # guest reviews, # property photos, # required minimum stays, security deposit, instant booking feature, whether a host is professional, and/or is a super host, and hosts' responsiveness to guests. We present the statistics of these variables in Table S7.

**Table S7 Summary Statistics for the DiD Control Variables.**

| | (1) Adopters | | (2) Non-adopters | | (3) All Properties | |
|---|---|---|---|---|---|---|
| **VARIABLE** | **Mean** | **Std. Dev.** | **Mean** | **Std. Dev.** | **Mean** | **Std. Dev.** |
| Number of Reviews | 42.88 | 51.64 | 28.07 | 41.78 | 31.65 | 44.82 |
| Number of Photos | 18.67 | 13.23 | 16.07 | 11.31 | 16.70 | 11.86 |
| Super Host | 0.23 | 0.42 | 0.16 | 0.36 | 0.17 | 0.38 |
| Instant Book Enabled | 0.18 | 0.38 | 0.10 | 0.30 | 0.12 | 0.32 |
| Security Deposit | 163.63 | 331.45 | 140.76 | 310.24 | 146.29 | 315.65 |
| # Minimum Stay | 2.97 | 10.58 | 3.03 | 5.54 | 3.01 | 7.09 |
| Response Rate (%) | 94.06 | 12.81 | 92.08 | 15.40 | 92.65 | 14.73 |
| # Host-Owned Listings | 2.54 | 3.92 | 3.02 | 8.72 | 2.90 | 7.84 |

Notes: The listed variables are time-varying and are captured in *Controls* in Equation (S13). The summary statistics are computed for the entire study period, over the 2, 118 adopters (column 1), over the 7,278 non-adopters (column 2), and over all the 9,396 properties (column 3).

**Section 5. Second-Stage Regression for Estimating the Main Effect of the Variable *Black***

We compute the main effect of *Black* in the DiD regression by rewriting equation (2) from the main paper as follows:

$$Y_{it} = Property_i + \beta \cdot SmartPricing_{it} + \delta \cdot (SmartPricing_{it} \times Race_i) + \lambda \cdot Controls_{it} + Seasonality_{it} + \varepsilon_{it} \quad \text{(S14)}$$

$Y_{it}$, can take the value of any of the three DVs: the average daily revenue in month *t*, average nightly rate in month *t,* and monthly occupancy rate. The main effect of *Black* cannot be identified directly from the regression in equation (S14) because it is absorbed in the time-variant property fixed effects, $Property_i$ . $Property_i$ captures the time-invariant factors, for example, neighborhood, host demographic, that impact daily revenue for $i$ prior to the launch of smart pricing. We first estimate $Property_i$ from (S14). Then we regress the estimated $Property_i$ on the time-invariant property and neighborhood characteristics, as shown in Equation (S15) using the minimum distance estimation (as discussed in Nevo 2000 and Chamberlain 1982):

$$Property_i = \alpha \cdot Race_i + \eta \cdot Characteristics_i + Neighborhood_n + \varepsilon_{it} \quad \text{(S15)}$$

We use *White* as the reference race group. We estimate α with respect to *Black* to capture the impact of being a Black (rather than a white) host on the DV, conditional on all property fixed effects and all other host and neighborhood characteristics. Note that black and white host may have very different property, neighborhood and host characteristics (e.g., white hosts on average may own properties in good location, higher neighborhood income, and with better property amenities). To ensure that the estimated effect of *Black* is attributable to race rather than other related covariates, we incorporate all observed property and neighborhood characteristics that are available to us. The covariates are captured in $Characteristics_i$ (see Table S8) and $Neighborhood_n$ (time-invariant factors such as location convenience, traffic, nearby attractions, and local demographics). After controlling aforementioned characteristics, the coefficient of *Black* indicates the conditional revenue/price/occupancy gap in a black host to his white counterpart, prior to the adoption of the algorithm.

Since we use $Property_i$ as the DV in the second-stage regression, we need to account for its sampling error to correct for the standard error estimates in the second-stage regression. We follow the procedure in Nevo (2000, 2001)[15] to use the Minimum-Distance Estimation proposed by Chamberlain (1982). Specifically, we define *f* as the $J \times 1$ property fixed effects that we obtained from the first-stage regression and *X* as the $J \times K$ property and neighborhood characteristics. Then, we regress property fixed effects on X:

$$f \sim X\beta + \epsilon \quad \text{(S16)}$$

where $\epsilon$ is the error term. The coefficient of *X* can be written as:

[15] In his work, Nevo (2001) firstto use the estimated brand fixed effects (which is akin to property fixed effects in our case) and then in the 2nd stage, regressed the estimated brand fixed effects over a set of time invariant product characteristics (which is akin to time invariant neighborhood, property and host characteristics in our case).

$$\beta = \left(X'V_f^{-1}X\right)^{-1} X'V_f^{-1}\hat{f} \quad \text{(S17)}$$

where $V_f$ is the covariance matrix of the estimated property fixed effects, $\hat{f}$, that we obtained from estimating the first-stage regression. Hence, second-stage estimation is a generalized least squares (GLS) regression where the correlations in the dependent $\hat{f}$ are weighted by the estimated covariance matrix $V_f$. We compute the standard errors (variance matrix) using standard formulas of the standard errors in a two-step parametric M-estimator (Hansen 1982, Newey and McFadden 1994)). Specifically, if we define the two stage coefficients as $\hat{\alpha} = \left(\hat{f}, \hat{\beta}\right)'$, and if $\alpha^*$ indicates the true values, then the asymptotic variance of $\sqrt{n}(\hat{\alpha} - \alpha^*)$ is given by

$$\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial g(X_i,\hat{\alpha})}{\partial \alpha'}\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i,\hat{\alpha})g(X_i,\hat{\alpha})'\right)\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial g'(X_i,\hat{\alpha})}{\partial \alpha}\right)^{-1} \quad \text{(S18)}$$

where $g(\cdot)$ represents the first-order conditions for the M-estimators in the second-stage regression.

The estimates of the second-stage regression for all three DVs are reported in Table S7. For average daily revenue, the coefficient of *Black*, $\alpha = -12.16$ ($p < 0.001$), indicates that white hosts earned \$12.16 more in average daily revenue than Black hosts, prior to the introduction of the algorithm and conditional on all other observed property and neighborhood characteristics. For the average nightly rate, $\alpha$ is insignificant, so Black and white hosts charged similar prices. For the monthly occupancy rate, prior to the introduction of the algorithm (conditional on all other observed property and hosts characteristics). $\alpha = -0.104$ ($p < 0.001$), so the occupancy of properties owned by Black hosts was 0.104 lower than the occupancy of properties owned by white hosts. prior to the introduction of the algorithm (conditional on all other observed property and hosts characteristics).

**Table S8 Regression Estimates of Property Fixed Effects on Property, Host, and Neighborhood Characteristics**

| | **(1) Gap in Daily Revenue** | | **(2) Gap in Nightly Rate** | | **(3) Gap in Occupancy Rate** | |
|---|---|---|---|---|---|---|
| **VARIABLE** | **Coefficient** | **Std. Err.** | **Coefficient** | **Std. Err.** | **Coefficient** | **Std. Err.** |
| ***Black*** | **-12.16***** | **(2.095)** | **2.024** | **(4.769)** | **-0.104***** | **(0.0104)** |
| *Other* | -6.086*** | (1.839) | -6.928 | (5.219) | -0.0361*** | (0.00930) |
| *Apartment* | -3.761* | (1.633) | -38.60*** | (5.521) | 0.0139 | (0.00750) |
| *Entire Home* | 39.80*** | (3.142) | 132.8*** | (9.812) | 0.0385* | (0.0170) |
| *Private Room* | 6.972** | (2.440) | 47.19*** | (8.796) | 0.0197 | (0.0184) |
| *# Bedrooms* | 23.61*** | (1.727) | 104.9*** | (5.947) | -0.0356*** | (0.00387) |
| *Home Value* | 0.0148*** | (0.00396) | 0.0477*** | (0.0127) | -0.00000767 | (0.00000779) |
| *Walk Score* | -0.00817 | (0.0409) | 0.00998 | (0.0749) | 0.000133 | (0.000166) |
| *Transit Score* | 0.329* | (0.155) | 0.822** | (0.282) | 0.000367 | (0.000500) |
| *Drive to Downtown(min)* | 0.0309 | (0.134) | -0.698 | (0.368) | 0.000841 | (0.000563) |
| *Bachelor (%)* | -0.0383 | (0.0862) | 0.876*** | (0.207) | -0.00116** | (0.000402) |

| | | | | | | |
|---|---|---|---|---|---|---|
| *Median Home Earning (1000 USD)* | 0.202* | (0.0975) | -0.149 | (0.230) | 0.000959** | (0.000300) |
| *Parking* | -0.408 | (1.780) | 1.897 | (4.127) | 0.00248 | (0.00664) |
| *Pool* | 10.18*** | (3.044) | 24.54* | (10.09) | 0.0264 | (0.0142) |
| *Beach* | 5.832 | (9.257) | -3.364 | (11.59) | 0.0107 | (0.0248) |
| *Internet* | 10.97* | (4.522) | -11.59 | (13.13) | 0.0930*** | (0.0217) |
| *TV* | 1.091 | (1.410) | 15.21*** | (3.026) | -0.0366*** | (0.00784) |
| *Dryer* | 3.266 | (1.890) | 1.902 | (4.650) | 0.0505*** | (0.00996) |
| *Washer* | 1.245 | (1.454) | 11.83** | (3.863) | -0.0435*** | (0.00721) |
| *Iron* | 9.878*** | (2.305) | -6.351 | (4.614) | 0.0797*** | (0.0125) |
| *Essentials* | 6.161* | (2.985) | 5.387 | (7.371) | 0.0233* | (0.0108) |
| *Heating* | -2.753 | (2.538) | 9.764 | (5.643) | 0.0177 | (0.0128) |
| *Microwave* | 4.658 | (4.217) | -1.501 | (8.619) | 0.0230 | (0.0196) |
| *Refrigerator* | -3.710 | (3.946) | -8.184 | (8.472) | 0.0249 | (0.0182) |
| *Laptop-Friendly* | -0.0548 | (1.948) | 6.755 | (5.135) | -0.0272** | (0.00876) |
| *Fireplace* | 3.390 | (2.924) | 40.74*** | (6.781) | -0.0333*** | (0.00955) |
| *Elevator* | -3.724 | (1.989) | 13.84** | (4.965) | -0.0415*** | (0.00929) |
| *Gym* | 4.900 | (5.042) | 29.58* | (12.98) | -0.0203 | (0.0157) |
| *Family-Friendly* | 0.735 | (1.825) | -8.107 | (5.290) | 0.0157* | (0.00696) |
| *Smoker Detector* | 4.388 | (2.466) | 2.690 | (5.657) | 0.0168 | (0.0114) |
| *Shampoo* | -5.448** | (2.083) | -3.309 | (5.890) | -0.0139 | (0.00910) |
| *Breakfast* | -8.443*** | (2.204) | 0.134 | (5.221) | -0.0379*** | (0.0113) |
| *AC* | -13.86* | (6.778) | -36.69 | (40.79) | -0.0148 | (0.0364) |
| *Age* | -0.0739 | (0.0616) | 0.248 | (0.154) | -0.000207 | (0.000271) |
| *# Photographed Faces* | 3.015** | (0.920) | 2.366 | (1.811) | 0.0156*** | (0.00428) |
| Fixed Effect | Neighborhood | | Neighborhood | | Neighborhood | |
| Observations | 9,396 | | 9,396 | | 9,396 | |
| R-squared | 0.31 | | 0.56 | | 0.24 | |

Notes: The regression estimates show the effect of being a Black (vs. white) host on average daily revenue, average nightly rate, and monthly occupancy rate prior to the introduction of the algorithm. The *white* racial group is used as the reference. The dependent variables are property-specific fixed effects that we estimated in the first-step DiD regression. Model is estimated on only the pre-treatment period observations (November 2015, variables were measured at the start of that period) observations were used for analyzing the revenue gap (column 1), nightly rate/price gap (column 2), occupancy gap (column 3), between the white and black ethnic groups of Airbnb hosts prior to the adoption of smart pricing algorithm. Cluster-robust standard errors at reported at the individual-neighborhood level in parentheses.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

## Section 6. Robustness Checks

### 6.1 Validating the DiD Model: Assessing the Parallel Trends in Pre-treatment Periods between Adopters and Non-Adopters for the IPTW Sample, PSM Sample, and Raw Sample

We compared the pre-treatment trends of the treatment and control groups to ensure that they followed similar trends in their property revenues prior to treatment (i.e., the algorithm adoption). We followed the standard approach: the estimation of a relative-time model (Autor 2003), which decomposes the pre-treatment periods into a series of period dummies and estimates the coefficients of the dummies that are within *k* periods prior to the treatment. Specifically, we decompose the pre-treatment periods and examine the following dummies: *Pre(j)* indicates the $j^{th}$ period prior to algorithm adoption (for $j = 1, 2, \ldots 5$); *Pre(6)* represents the entire pre-treatment period from the beginning of our observational window to the $6^{th}$ period prior to algorithm adoption.

$$Y_{it} = Property_i + \sum_{j>0} \alpha_j \cdot Adopter_i \cdot Pre(j)_{it} + \beta \cdot SmartPricing_{it} + \lambda \cdot Controls_{it} + Seasonality_{it} + \varepsilon_{it} \quad \text{(S19)}$$

We represent the treatment condition with $Adopter_i$ (equals 1 if property $i$ is in the treatment group, meaning that the host of property $i$ adopted the algorithm at some point during the study period, and 0 otherwise). Each time dummy, $Pre(j)_{it}$, equals 1 if period *t* is *j* periods prior to algorithm adoption for property *i,* and 0 otherwise. The parameters $\{\alpha_j\}$ identify the trend in the dependent variable for the treatment group, relative to the control group, in the pre-treatment periods. We set the period prior to the adoption month as a reference period (i.e., $\{\alpha_j\}_{j=1}$ was normalized to zero).

***Parallel Trends in the IPTW Sample***

We find that all values of $\alpha_j$ are not positive and significant (Table S9). In Figure S4, we plot the estimated coefficients for each pre-treatment period; all 95% confidence intervals contain zero. The parallel trends assumption is not rejected in the IPTW weighted sample.

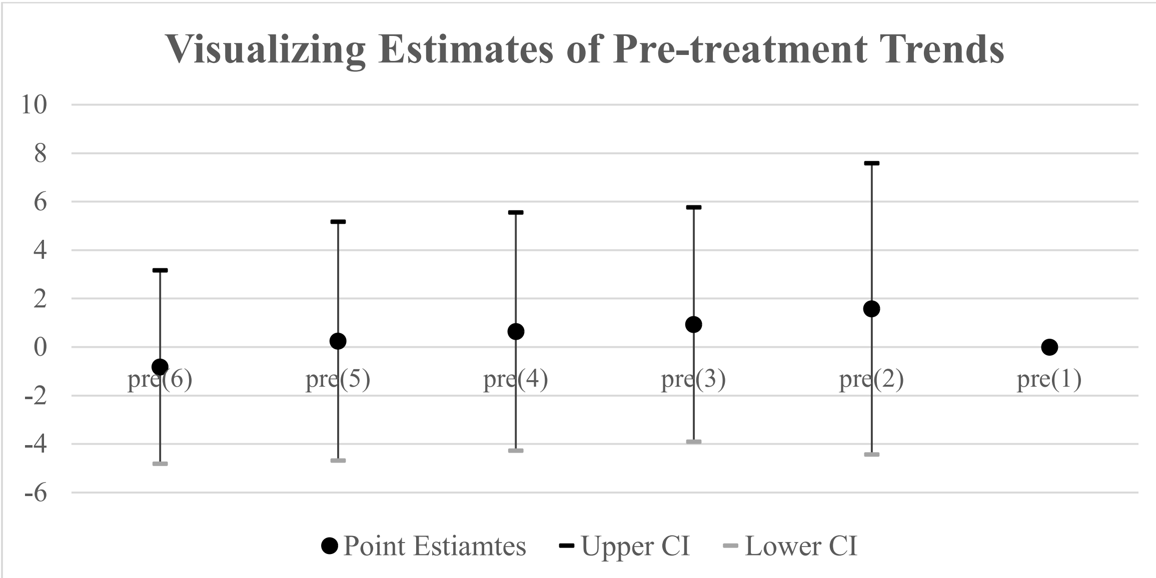
**Table S9 Validating the DiD Model: Assessing Parallel Trends Assumption in Pre-treatment Periods for the IPTW Weighted Sample**

| VARIABLE | ESTIMATES | |
|---|---|---|
| | **Coefficients** | **Std. Err.** |
| **Pre-Treatment Trends** | | |
| *Adopter·Pre(6)* | -0.825 | (2.038) |
| *Adopter·Pre(5)* | 0.240 | (2.514) |
| *Adopter·Pre(4)* | 0.634 | (2.507) |
| *Adopter·Pre(3)* | 0.927 | (2.468) |

| | | |
|---|---|---|
| *Adopter·Pre(2)* | 1.577 | (3.064) |
| *Adopter·Pre(1)—reference* | -- | -- |
| **Effect of Algorithm Adoption on Average Daily Revenue** | | |
| ***Smart Pricing*** | 6.396** | (1.996) |
| *log Number of Reviews* | 11.87*** | (1.183) |
| *log Number of Photos* | 6.772* | (2.952) |
| *log Security Deposit* | 6.084*** | (0.254) |
| *log # Min. Stays* | -10.43*** | (2.293) |
| *Instant Book Enabled* | 11.03*** | (1.466) |
| *Super Host* | 9.023*** | (1.961) |
| *Professional Host (log # listings)* | 0.353 | (4.285) |
| *Host Effort (Response Rate)* | 0.0278 | (0.0287) |
| Fixed Effect | Property | |
| Seasonality | City-Year, City-Month | |
| Observations | 162617 | |
| R-squared | 0.51 | |

Notes: The table assesses the parallel trend assumption in the DiD model for the IPTW sample of 9,396 properties. Columns present the coefficients obtained from estimating Equation (S19). The Dependent variable is property revenue (for property i in month t). The panel *Pre-Treatment Trends* reports the coefficients of the interactions *Adopter·Pre(j)*, estimated in Equation (S19), where *Pre(j)* indicates the $j^{th}$ period prior to treatment (for j = 1, 2, … 5); *Pre(6)* represents the entire pre-treatment period. The month immediately prior to the adoption month (that is, *Pre(1)*) is the reference period. All coefficients of *Adopter·Pre(j)* are non-significant, so the parallel trends assumption is not rejected.
Cluster-robust standard errors reported at the individual-property level in parentheses.
* p < 0.05 ** p < 0.01 *** p < 0.001

**Figure S4 Plot of Estimated Coefficients of the Pre-Treatment Dummies for the IPTW Sample**




***Testing Pre-treatment Trends for DiD Analysis in the PSM Sample***

We use the same relative-time model that we described for the IPTW sample. We regress the relative-time model (Equation S19) on the PSM sample. We report the estimation results in Table S10; none of the estimated coefficients for the pre-treatment dummies are positive and statistically significant, suggesting that in the PSM sample, there did not exist significant difference in the outcome variables between the adopters and the non-adopters, prior to the treatment. We plot the same results in Figure S5; all 95% confidence intervals in the pre-treatment periods contain zero. As with the IPTW sample, we do not reject the parallel trends assumption.

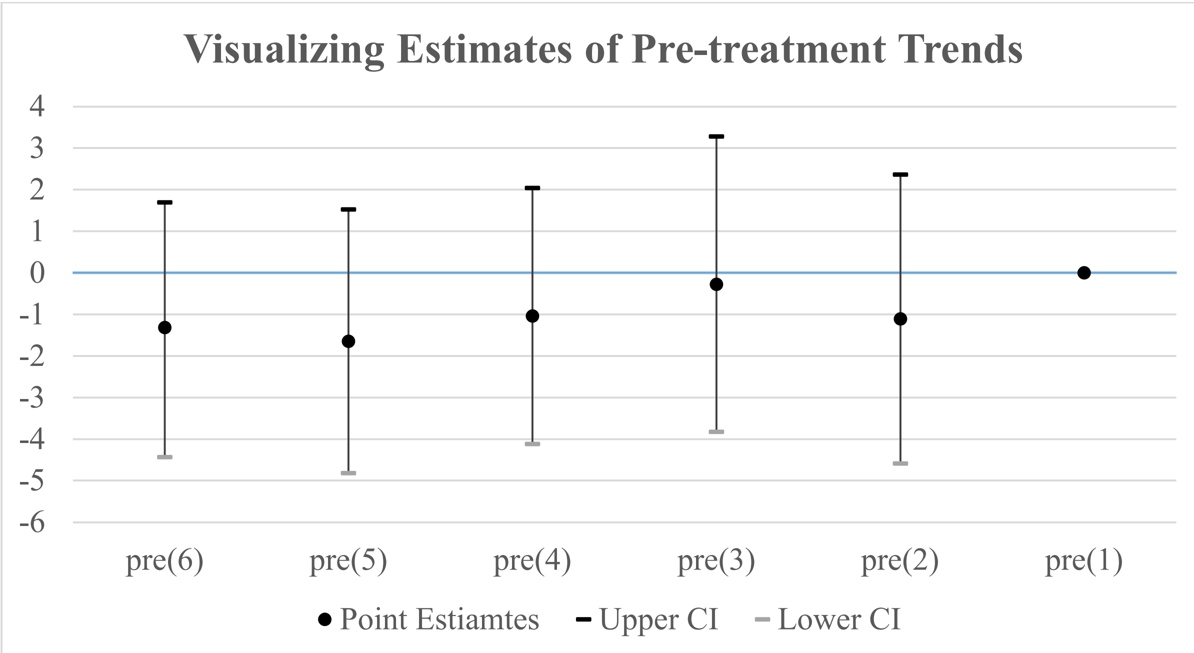
**Table S10 Pre-Treatment Trends in the PSM Sample**

| **VARIABLE** | **ESTIMATES** | |
|---|---|---|
| | **Coefficients** | **Std. Err.** |
| **Pre-Treatment Trends** | | |
| *Adopter·Pre(6)* | -1.315 | (1.590) |
| *Adopter·Pre(5)* | -1.647 | (1.617) |
| *Adopter·Pre(4)* | -1.039 | (1.572) |
| *Adopter·Pre(3)* | -0.275 | (1.813) |
| *Adopter·Pre(2)* | -1.110 | (1.773) |
| *Adopter·Pre(1)—reference* | -- | -- |
| **Effect of Algorithm Adoption on Average Daily Revenue** | | |
| ***Smart Pricing*** | 4.572** | (1.377) |

| | | |
|---|---|---|
| *log Number of Reviews* | 12.35*** | (1.474) |
| *log Number of Photos* | 6.358** | (2.399) |
| *log Security Deposit* | 2.451*** | (0.271) |
| *log # Min. Stays* | -11.76*** | (2.731) |
| *Instant Book Enabled* | 10.99*** | (1.567) |
| *Super Host* | 6.983*** | (1.238) |
| *Professional Host (log # listings)* | 2.998 | (3.452) |
| *Host Effort (Response Rate)* | 0.0252 | (0.0295) |
| Fixed Effect | Property | |
| Seasonality | City-Year, City-Month | |
| Observations | 101536 | |
| R-squared | 0.54 | |

Notes: The table assesses the parallel trend assumption in the DiD model for the PSM sample of 5,469 properties, of which 1,631 properties adopted the pricing algorithm during the observation period. (The PSM method discards units that cannot be matched with similar units, so the final sample size is smaller than in the IPTW method.) The coefficients are estimated with Equation (S19), and the table is otherwise analogous to Table S9.
Cluster-robust standard errors at reported at the individual-property level in parentheses.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

**Figure S5 Plot of Estimated Coefficients of the Pre-Treatment Dummies for the PSM Sample: DiD on PSM-Matched Sample**




***Lack of Parallel Trends in the Raw (i.e., Unweighted and Unmatched) Sample***

We use the same relative-time model that we described for the IPTW sample. We report the estimation results in Table S11; all of the estimated coefficients for the pre-treatment dummies are statistically significant. We plot the same results in Figure S6; none of the 95% confidence intervals in the pre-treatment periods contain zero. Unlike for the IPTW and PSM samples, we do reject the parallel trends assumption in the raw sample. In other words, the raw sample contained systematic differences between the treatment and control groups that could influence the outcome variable.

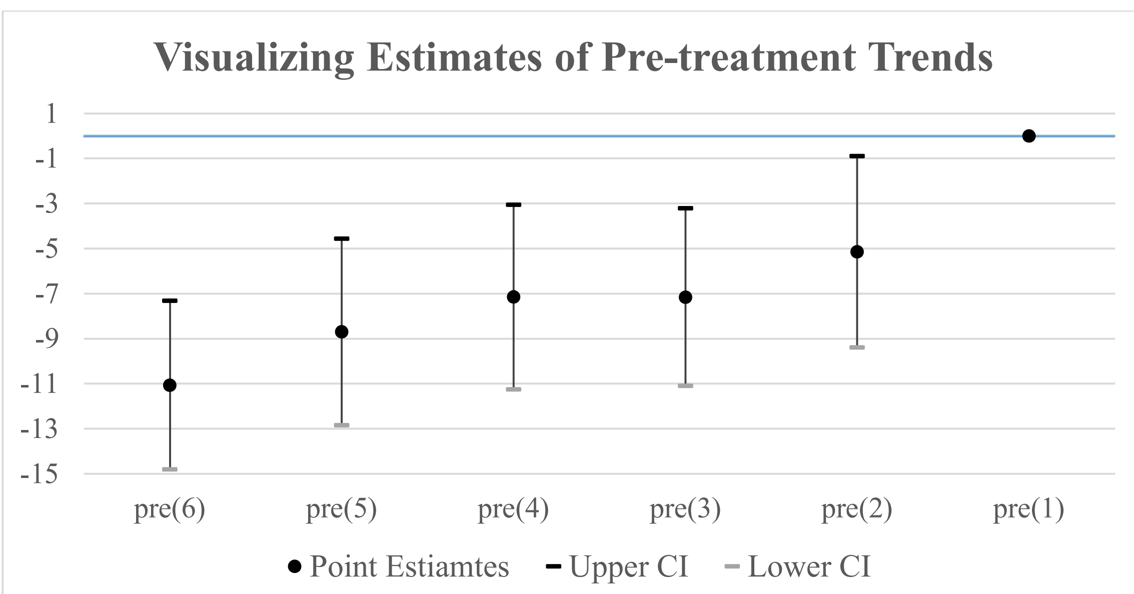
Together, the results of the relative-time model analyses for the IPTW, PSM, and raw samples indicate that the PSM and IPTW methods functioned as intended to mitigate the self-selection issue in identification, to the extent that reduced the systematic differences in the two groups that influence the outcome variable and the algorithm adoption decision.

**Table S11 Pre-Treatment Trends in the Raw Sample**

| VARIABLE | ESTIMATES | |
|---|---|---|
| | **Coefficients** | **Std. Err.** |
| **Pre-Treatment Trends** | | |
| *Adopter·Pre(6)* | -11.06*** | (1.909) |
| *Adopter·Pre(5)* | -8.699*** | (2.112) |
| *Adopter·Pre(4)* | -7.152*** | (2.089) |

| | | |
|---|---|---|
| *Adopter·Pre(3)* | -7.158*** | (2.012) |
| *Adopter·Pre(2)* | -5.141* | (2.165) |
| *Adopter·Pre(1)—reference* | -- | -- |
| **Effect of Algorithm Adoption on Average Daily Revenue** | | |
| ***Smart Pricing*** | 0.236 | (1.787) |
| *log Number of Reviews* | 13.66*** | (1.068) |
| *log Number of Photos* | 6.326** | (1.968) |
| *log Security Deposit* | 6.397*** | (0.181) |
| *log # Min. Stays* | -11.46*** | (1.889) |
| *Instant Book Enabled* | 11.30*** | (1.329) |
| *Super Host* | 9.316*** | (1.617) |
| *Professional Host (log # listings)* | 1.620 | (2.616) |
| *Host Effort (Response Rate)* | 0.012 | (0.0234) |
| Fixed Effect | Property | |
| Seasonality | City-Year, City-Month | |
| Observations | 162617 | |
| R-squared | 0.52 | |
| Notes: The table assesses the parallel trend assumption in the DiD model for the raw (i.e., unweighted and unmatched) sample of 9,396 properties. The coefficients are estimated with Equation (S19), and the table is otherwise analogous to Table S9. Cluster-robust standard errors at reported at the individual-property level in parentheses. * p < 0.05 ** p < 0.01 *** p < 0.001 | | |

**Figure S6 Plot of Estimated Coefficients of the Pre-Treatment Dummies for the Raw Sample**



### 6.2 Validating the DiD Model: Assessing the Parallel Trends Assumption for White and Black Hosts Separately

We examine the pre-treatment trends for Black hosts and white hosts separately in the IPTW sample. We consider the pre-treatment period in 6 segments, as in Section 6.1, and to assess the within ethnic group (i.e., white/black) trends, we interact *Race* with the series of pre-period dummies and the *Adopter* dummy:

$$Y_{it} = Property_i + \sum_j \alpha_j \cdot Adopter_i \cdot Pre(j)_{it} + \sum_j \eta_j \cdot Adopter_i \cdot Pre(j)_{it} \cdot Race_i + \beta \cdot SmartPricing_{it} + \lambda \cdot Controls_{it} + Seasonality_{it} + \varepsilon_{it} \quad \text{(S20)}$$

All variables are described in Section 6.1. We set the month before the adoption month as the reference period (i.e., we normalize *Pre(1)* to zero). We set *white* as the reference race, so the coefficients $\{\alpha_j\}_{j=2}^{6}$ capture the pre-treatment trends in the dependent variable among white hosts; coefficients $\{\alpha_j + \eta_j\}_{j=2}^{6}$ of the three-way interaction term capture the pre-treatment trends in the dependent variable among Black hosts. The other parameters were pooled across all hosts. [16]

Table S12 reports the estimation results. We present the trends for white ($\alpha_j$) and for black host ($\eta_j$) in column (1) and (2), respectively. All estimated coefficients for the pre-treatment dummies are statistically insignificant. We plot coefficients separately for white and Black hosts in Figure S7, and there is no visible time-trend series in either racial group. We conclude that the parallel trends assumption for the two racial groups is not rejected. In other words, the pre-treatment trend in average daily revenue was not significantly different in the treatment vs. control groups among white hosts or among Black hosts.

**Table S12 Pre-Treatment Trends in the IPTW sample, Separated by Host Race**

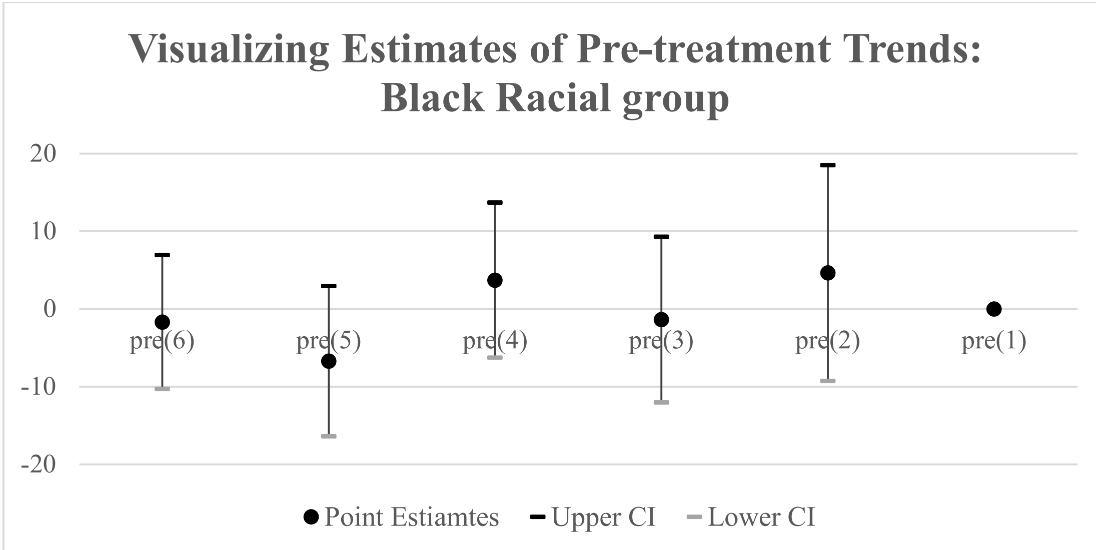
| VARIABLE | ESTIMATES | |
|---|---|---|
| | White Hosts (reference) | Black Hosts |
| Pre-Treatment Trends | | |
| *Adopter·Pre(6)* | -0.154 | -1.687 |
| | (2.689) | (4.403) |
| *Adopter·Pre(5)* | 2.485 | -6.732 |
| | (3.259) | (4.936) |
| *Adopter·Pre(4)* | 1.215 | 3.672 |
| | (3.167) | (5.042) |
| *Adopter·Pre(3)* | 1.190 | -1.377 |
| | (3.265) | (5.444) |

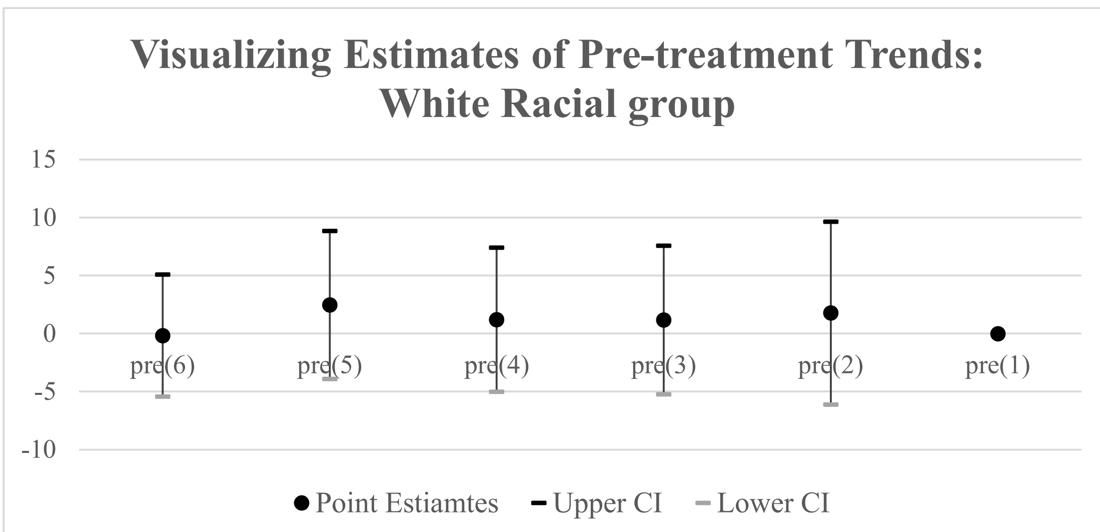
[16] The results did not change when we estimated the other parameters separately for Black and white hosts.

| | | |
|---|---|---|
| *Adopter·Pre(2)* | 1.786 | 4.616 |
| | (4.018) | (7.081) |
| *Adopter·Pre(1)—reference* | -- | -- |
| **Effect of Smart Pricing on Daily Revenue** | | |
| *Smart Pricing* | 6.388** | |
| | (2.352) | |
| *log Number of Reviews* | 11.91*** | |
| | (1.177) | |
| *log Number of Photos* | 6.729* | |
| | (2.951) | |
| *log Security Deposit* | 6.073*** | |
| | (0.255) | |
| *log # Min. Stays* | -10.43*** | |
| | (2.294) | |
| *Instant Book Enabled* | 11.04*** | |
| | (1.467) | |
| *Super Host* | 8.996*** | |
| | (1.958) | |
| *Professional Host (log # listings)* | 0.262 | |
| | (4.299) | |
| *Host Effort (Response Rate)* | 0.0278 | |
| | (0.0287) | |
| Fixed Effect | Property | |
| Seasonality | City-Year, City-Month | |
| Observations | 162617 | |
| R-squared | 0.51 | |

Notes: The table assesses the parallel trend assumption in the DiD model for the IPTW sample of 9,396 properties, with separate analyses of the trends among Black and white hosts. The table assess the critical Parallel Trend Assumption in a DiD model. Column (1) and (2) presents the relative-time pre-trend assessment (traced for 6-period back) for black ethnic sub-group only, and for white ethnic sub-group only. The coefficients are estimated with Equation (S20). The panel *Pre-Treatment Trends* reports the coefficients of the interactions *Adopter·Pre(j)*, estimated in Equation (S20), where *Pre(j)* indicates the $j^{th}$ period prior to treatment (for j = 1, 2, … 5); *Pre(6)* represents the entire pre-treatment period. The month immediately prior to the adoption month (that is, *Pre(1)*) is the reference period. All coefficients of *Adopter·Pre(j)* are non-significant, so the parallel trends assumption is not rejected.
Cluster-robust standard errors at reported at the individual-property level in parentheses.
* p < 0.05 ** p < 0.01 *** p < 0.001

**Figure S7 Estimated Coefficients of the Pre-Treatment Dummies for the IPTW Sample, Plotted by Host Race**




**6.3 Dynamic Treatment Effects**

We examine whether and to what extent the treatment effect of algorithm adoption on average daily revenue is dynamic. We consider three potential sources of dynamic treatment effects: 1) Airbnb may have changed the algorithm itself, which should improve the algorithm and thus increase the treatment effect over time; 2) as more competing properties adopt the algorithm over time, the treatment effect should decrease; and 3) the algorithm's performance may vary seasonally (i.e., between the peak season and off-peak season).

Starting with the first source (i.e., changes in the algorithm over time), we do not know the specific points in time when Airbnb made these changes. However, if Airbnb did make changes to its algorithm, then it is reasonable to believe that these changes would have been improvements in the algorithm. Therefore all else being the same, the first source would result in increase in the average treatment effect over time. Moving on to the second source (i.e., the competitive effect from other properties who have adopted the algorithm), it will result in the average treatment effect to diminish over time. And finally the third source (i.e., the seasonality effect). Seasonal variation would create a distinct pattern, but it is more difficult to identify the separate dynamic impacts from the first two sources. We parse out the three possibilities by running two regressions: The first identifies the dynamic effect of seasonality and the joint effect of the first two sources. The second regression attempts to identify the dynamic effect of each of the three sources.

In the first regression, we regress average daily revenue on three-way interaction terms involving the following variables:

(i) $SmartPricing_{it}$, a dummy variable that equals 1 if the focal host *i* adopted the algorithm in month *t*

(ii) $OffPeak_t$, a dummy variable that equals 1 if month *t* is in the off-peak season. For simplicity, we use a binary classification of the season: peak and off-peak, with the peak season as the baseline. We classify each month as either peak or off-peak for each city in the dataset (see Section 7).

(iii) $PostYear_t$, a dummy variable that equals 1 if month *t* falls between November 2015 (when Airbnb introduced the algorithm) and October 2016, and equals 2 if month *t* falls between November 2016 to August 2017. (We also use a value of 1 for the pre-launch period, but this does not make a material difference since *SmartPricing* = 0.)

The results of the first regression are reported in column 1 of Table S13. The interaction effect of *SmartPricing x OffPeak* captures whether the treatment effect varies by season; all coefficients that include this interaction term are insignificant. The *SmartPricing x PostYear* interaction captures whether the treatment effect changed from year 1 to year 2 of algorithm adoption (i.e., the joint effect of Airbnb's changes in the algorithm and competitors' algorithm adoption); all coefficients that include this interaction term are insignificant. This may indicate that the two contributors to this effect cancelled each other out (i.e., Airbnb's changes in the algorithm increased the treatment effect while competitors' algorithm adoption decreased the treatment effect) or that neither contributor has a significant dynamic effect.

In the second regression, we try to disentangle the dynamics from all three sources by introducing another variable: *Zip-code adoption rate*$_t$, the fraction of listings in the same zip-code as that of the focal property that used the algorithm in month *t*. We operationalize the zip-code adoption rate from our sample. The second regression retains the three variables from the first regression (*SmartPricing, OffPeak,* and *PostYear*). In this regression, the interaction terms involving the variables 'adoption of the algorithm' and 'seasonality' will capture the dynamics in the treatment effects that stem form the third source. The interaction terms involving 'adoption of the algorithm' and 'zip code adoption rate' will capture the dynamics that stem from the second source (the competitive effect). And the interaction terms involving 'adoption of the algorithm' and 'number of years lapsed since the launch of the algorithm' will capture the dynamics from the first source (changes in the algorithm over time).

The results of the second regression are reported in column 2 of Table S13. The interaction effect of *SmartPricing x Zip-code adoption rate* captures the impact of competitors' algorithm adoption (the second source); all coefficients that include this interaction term are insignificant. And the parameters of interest related to the first source are the coefficients of all the interaction terms that involve (*PostYear=2)* and *SmartPricing*; these coefficients also are insignificant. The results suggest that there may not be significant dynamics in the treatment effects that stem from the first source (changes in the algorithm over time).

We note two limitations of our analysis. First, the average adoption rate across all zip codes by August 2017 (the last month in our data) is only 13.8%; competition could give rise to a dynamic treatment effect at higher rates of algorithm adoption. Second, *Zip-Code Adoption Rate* is based on only the properties in our final sample, so it may not accurately capture the true adoption rate among all properties in the zip-code.

**Table S13 Sources of a Dynamic Treatment Effect: Interactions Among *SmartPricing, OffPeak, PostYear, and Zip-Code Adoption Rate***

| | ESTIMATES | |
|---|---|---|
| **VARIABLE** | (1) Interacting PostYear, OffPeak | (2) Interacting PostYear, OffPeak, Zip-Code Adoption Rate |
| ***Smart Pricing*** (Reference: OffPeak = 0, *PostYear* = 1) | 8.182*** | 10.51*** |
| | (1.738) | (2.623) |
| ***SmartPricing X OffPeak*** | -2.171 | -6.414 |
| | (3.954) | (4.631) |
| ***SmartPricing X PostYear =2*** | -1.463 | -4.064 |
| | (2.405) | (4.978) |
| ***SmartPricing X OffPeak X PostYear = 2*** | -5.996 | -2.055 |
| | (4.794) | (6.147) |
| ***Zip-Code Adoption Rate*** | | -11.44 |
| | | (10.56) |
| ***SmartPricing X Zip-Code Adoption Rate*** | | -21.91 |
| | | (12.88) |
| ***PostYear X Zip-Code Adoption Rate*** | | -9.682 |
| | | (18.17) |
| ***SmartPricing X PostYear X Zip-Code Adoption Rate*** | | 22.80 |
| | | (27.91) |
| ***OffPeak X PostYear X Zip-Code Adoption Rate*** | | 23.24 |
| | | (27.55) |
| ***SmartPricing X OffPeak X Zip-Code Adoption Rate*** | | 74.51 |
| | | (39.74) |
| ***SmartPricing X OffPeak X PostYear X Zip-Code Adoption Rate*** | | -64.45 |
| | | (50.30) |
| ***OffPeak X Zip-Code Adoption Rate*** | | -41.02* |
| | | (20.23) |
| logNumberofReviews | 11.85*** | 11.86*** |
| | (1.187) | (1.189) |

| | | |
|---|---|---|
| logNumberofPhotos | 6.837* | 6.855* |
| | (2.930) | (2.933) |
| logSecurityDeposit | 6.123*** | 6.126*** |
| | (0.254) | (0.254) |
| logMinimumStay | -10.34*** | -10.31*** |
| | (2.299) | (2.299) |
| InstantBook | 11.09*** | 11.09*** |
| | (1.465) | (1.460) |
| SuperHost | 9.030*** | 8.999*** |
| | (1.968) | (1.967) |
| lognum_listings | 0.352 | 0.336 |
| | (4.266) | (4.194) |
| Response Rate | 0.0336 | 0.0348 |
| | (0.0288) | (0.0289) |
| Fixed Effect | Property | Property |
| Seasonality | City-Year, City-Month | City-Year, City-Month |
| Observations | 162617 | 162617 |
| R-squared | 0.51 | 0.51 |

Notes: *SmartPricing* indicates whether the algorithm was adopted. *OffPeak* indicates whether month *t* falls in the off-peak season in the focal city. *PostYear* indicates whether month *t* is within 1 year or between 1 and 2 years since the introduction of the algorithm. *Zip-code adoption rate* is the fraction of properties in the same zip-code as the focal property that used the algorithm in month *t*. The dependent variable is the average daily revenue. The models are estimated on the IPTW-weighed sample of 9,396 unique properties. Rows in bold report the estimated coefficients for the interaction terms.
Robust standard errors clustered at individual property level are presented in parentheses.
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

### 6.4 A Comparison of Alternative Methods: Raw DiD, DiD with PSM, and DiD with the Synthetic Control Method

We test three alternative approaches for estimating the effect of algorithm adoption on average daily revenue and the differential effect of race.

***DiD Analysis on the Raw Sample***

The "raw" sample has not been weighted or matched to balance the treatment and control groups (we rejected the parallel trends assumption in Section 6.1), so a DiD analysis on the raw sample can yield biased estimates.

Table S14 reports the results. The DiD analysis on the raw sample estimated an average treatment effect of 7.250 and a differential effect of 9.561. These effects are overestimated relative to the effects estimated on the IPTW sample: an average treatment effect of 6.396 and a differential effect of 8.700.

**Table S14 Results of the DiD Analysis on the Raw Sample**

| VARIABLE | (1) Main Effect | | (2) Interaction with Race | |
|---|---|---|---|---|
| | **Coefficients** | **Std. Err.** | **Coefficients** | **Std. Err.** |
| ***Smart Pricing*** | 7.250*** | (1.197) | 6.006*** | (1.327) |
| ***Smart Pricing X Black*** | | | 9.651** | (3.132) |
| *Smart Pricing X Other* | | | 2.434 | (3.367) |
| *log Number of Reviews* | 14.14*** | (1.056) | 14.14*** | (1.055) |
| *log Number of Photos* | 6.486*** | (1.966) | 6.446** | (1.969) |
| *log Security Deposit* | 6.391*** | (0.181) | 6.389*** | (0.181) |
| *log # Min. Stays* | -11.52*** | (1.885) | -11.51*** | (1.885) |
| *Instant Book Enabled* | 11.34*** | (1.331) | 11.37*** | (1.331) |
| *Super Host* | 9.471*** | (1.619) | 9.505*** | (1.619) |
| *Professional Host (log # listings)* | 1.922 | (2.606) | 1.921 | (2.606) |
| *Host Effort (Response Rate)* | 0.0187 | (0.0234) | 0.0188 | (0.0234) |
| Fixed Effect | Property | | Property | |
| Seasonality | City-Year, City-Month | | City-Year, City-Month | |
| Observations | 162617 | | 162617 | |
| R-squared | 0.52 | | 0.52 | |

Notes: A DiD analysis was conducted on the raw sample of 9, 396 unique properties. The dependent variable is average daily revenue. Column (1) presents the measures of key variable Smart Pricing. The model presented in column (2) includes an interaction term of the adoption decision and the ethnicity of a host: Smart Pricing X Black.
Cluster-robust standard errors are reported at the individual-property level in parentheses.
* p < 0.05 ** p < 0.01 *** p < 0.001

***DiD Analysis on the PSM Sample***

The PSM approach resulted in a matched sample of 5,469 properties, of which 1,631 properties adopted the pricing algorithm during the observation period.

Table S15 reports the results of the DiD analysis on the PSM sample. The average treatment effect is positive and significant (b = 4.822, p < 0.001), but the differential effect is non-significant (*b* = 4.895, p < 0.11). By contrast, the DiD analysis on the IPTW sample found a significant differential effect.

**Table S15 Results of the DiD Analysis on the PSM Sample**

| VARIABLE | (1) Main Effect | | (2) Interacting with Race | |
|---|---|---|---|---|
| | Coefficients | Std. Err. | Coefficients | Std. Err. |
| ***SmartPricing*** | 4.822*** | (1.322) | 4.140** | (1.464) |
| ***SmartPricing×Black*** | | | 4.895 | (3.060) |
| *SmartPricing×Other* | | | 3.631 | (3.061) |

| | | | | |
|---|---|---|---|---|
| *Log Number_of_Reviews* | 12.35*** | (1.474) | 12.35*** | (1.474) |
| *Log Number_of_Photos* | 6.358** | (2.399) | 6.340** | (2.400) |
| *Log Security_Deposit* | 2.450*** | (0.270) | 2.452*** | (0.270) |
| *Log #Min. Stays* | -11.76*** | (2.731) | -11.77*** | (2.730) |
| *Instant_Book_Enabled* | 11.00*** | (1.566) | 11.02*** | (1.567) |
| *Super_Host* | 6.982*** | (1.238) | 7.018*** | (1.237) |
| *Log #host-owned listings* | 3.000 | (3.452) | 3.009 | (3.451) |
| *Host-_Effort(Response_Rate)* | 0.0251 | (0.0294) | 0.0249 | (0.0294) |
| Fixed Effect | Property | | Property | |
| Seasonality | City-Year, City-Month | | City-Year, City-Month | |
| Observations | 101536 | | 101536 | |
| R-squared | 0.54 | | 0.54 | |

Notes: A DiD analysis was conducted on the PSM sample of 5,469 unique properties. (The PSM method discards units that cannot be matched with similar units, so the final sample size is smaller than in the IPTW method.) The dependent variable is average daily revenue. Column (1) presents the measures of key variable Smart Pricing. The model presented in column (2) includes an interaction term of the adoption decision and the ethnicity of a host: Smart Pricing X Black.
Cluster-robust standard errors at reported at the individual-property level in parentheses.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

We offer two explanations for the discrepancy in the significance of the differential effect between the IPTW and PSM samples. First, the estimates reflect subtly different types of effects: the IPTW sample uses the average treatment effect (ATE), while the PSM sample uses the average treatment effect on the treated (ATT). Second, the PSM sample has lower statistical power because it is smaller; the matching process discarded 49% of the observations. Of particular concern, the PSM sample contains only 97 Black hosts who adopted the algorithm (versus 150 Black adopters in the IPTW sample). This small number of black adopters resulted in a non-significant estimate of the differential effect.On the other hand, when we use IPTW, we use the full sample in which we have a total of 10,903 properties with 2,118 adopters and 150 black adopters. This 55% increase in the number of black adopters resulted in a significant estimate of the differential effect when using IPTW.

We test these explanations by conducting an analysis that balances sample with a different approach: the synthetic control method (SCM). SCM uses a full sample (like IPTW) and yields the ATT (like PSM), so the results enable us to discern which explanation holds more weight. Specifically, an insignificant differential effect from the SCM analysis would suggest that the discrepancy between the IPTW and PSM samples occurred because of the difference in the type of treatment effect estimate; a significant differential effect would instead implicate the low statistical power of the PSM.

***Performing SCM Analysis***

The synthetic control strategy estimates the treatment effect by constructing a synthetic control group of units (i.e., counterfactuals) that mimic the treated units. Specifically, SCM finds a convex combination of untreated units such that the outcome of the synthetic control group closely represents the outcome of the treatment group in the pre-treatment period (Abadie et al. 2010, Abadie et al. 2015). We implement SCM with the Generalized Synthetic Control method (Xu

2017), which is appropriate for multiple treatment units, and we use the R package "gsynth" developed by Xu and Liu[17] to implement the analysis.

Table S16 reports the estimation results of SCM analysis. The estimates of both the average treatment effect (in column 1, $b = 5.471$, $p < 0.001$) and the differential effect (in column 2, $b = 7.577$, $p < 0.05$) are significant. We conclude that the DiD analysis on the PSM sample did not yield a significant estimate of the differential effect because the PSM sample had low statistical power.

**Table S16 Impact of Smart Pricing on Daily Revenue: A Generalized Synthetic Control Approach**

| **VARIABLE** | **(1) Main Model** | | **(2) Interacting with Race** | |
|---|---|---|---|---|
| | **ESTIMATES** | **S.E.** | **ESTIMATES** | **S.E.** |
| ***Smart Pricing*** | 5.471128*** | 1.285069794 | 4.584356*** | 1.375587 |
| ***SmartPricing×Black*** | | | 7.57656* | 3.074063 |
| *SmartPricing×Other* | | | 4.362226 | 2.679952 |
| *log Number of Reviews* | 13.47936*** | 0.805344 | 13.60937*** | 0.824151 |
| *log Number of Photos* | 7.160971** | 2.232651 | 7.101532** | 2.236745 |
| *log Security Deposit* | 7.20503*** | 0.14932 | 7.111945*** | 0.143913 |
| *log # Min. Stays* | -11.7408*** | 1.117515 | -11.9724*** | 1.062721 |
| *Instant Book Enabled* | 11.50455*** | 1.215182 | 11.53208*** | 1.264121 |
| *Super Host* | 7.181684*** | 1.214445 | 7.195109*** | 1.249905 |
| *Professional Host (log # listings)* | 2.097485 | 2.005196 | 2.181062 | 2.021605 |
| *Host Effort (Response Rate)* | 0.16218 | 0.23975 | 0.15988 | 0.2261 |
| Fixed Effect | Property | | Property | |
| Seasonality | City-Year, City-Month | | City-Year, City-Month | |

Notes: A DiD analysis was conducted on the SCM sample of 9, 396 unique properties. The dependent variable is average daily revenue. Column (1) presents the measures of key variable Smart Pricing. The model presented in column (2) includes an interaction term of the adoption decision and the ethnicity of a host: Smart Pricing X Black.
Cluster-robust standard errors at reported at the individual-property level in parentheses.
* p < 0.05   ** p < 0.01   *** p < 0.001

[17] Details of the implementation and inferences can be found at https://cran.r-project.org/web/packages/gsynth/gsynth.pdf.

**Section 7: Investigating a Seasonal Price Correction as an Alternative Mechanism for the Main and Differential Effects of Algorithm Adoption**

In this section, we investigate whether and to what extent an alternative mechanism—a seasonal price correction—can explain the average treatment effect (i.e., the effect of algorithm adoption on average daily revenue; Section 7.1) and the differential effect (i.e., the algorithm adoption benefits Black hosts more than white hosts; Section 7.2).

### 7.1 Seasonal Price Correction and the Average Treatment Effect

Our main regression revealed that algorithm adoption leads to a decrease in the average nightly rate (i.e., a downward price correctio, see Section 3 of the main paper) and an increase in the monthly occupancy rate. We concluded that average daily revenue increases because the increase in occupancy offsets the decrease in price.

In the main analysis, we did not investigate whether the downward price correction was the same throughout the year or whether it varied by season. It is possible, for instance, that the algorithm might make a larger downward price correction during the off-peak season because the hosts themselves may not be sufficiently sensitive to the decrease in demand. Thus in this section, we examine whether and to what extent did the algorithm introduce a seasonal price correction as opposed to a general price correction – where a general price correction refers to the case when adoption of the algorithm leads to a similar price correction across all seasons in the year, and seasonal price correction refers to the case when adoption of the algorithm leads to a downward price correction during off peak seasons only.

We first regress the average nightly rate on algorithm adoption in month *t* and the interaction between algorithm adoption and season:

$$\begin{aligned} NightlyRate_{it} &= \beta \cdot SmartPricing_{it} + \gamma \cdot SmartPricing_{it} \times OffPeak_{it} + \lambda \cdot Controls_{it} \\ &+ Seasonality_{it} + Property_{i} + \varepsilon_{it} \end{aligned} \quad \text{(S21)}$$

$SmartPricing_{it}$ is an dummy variable that equals 1 if the host of property *i* used the pricing algorithm in month *t* and 0 otherwise. $OffPeak_{it}$is an indicator variable that equals 1 if month *t* is in the off-peak season in the city of property *i* and 0 if not. The parameter β captures the impact of algorithm adoption on the nightly rate during the peak season, and the parameter γ captures the differential effect of peak vs. off-peak seasons.

We classify months into the peak season and off-peak season categories at the city level. For each city, we first calculate the average occupancy rate each month for the full year prior to the introduction of the algorithm (i.e., Nov 2014 to Oct 2015)[18]. We categorize the three months with the lowest occupancy as the off-peak season and the rest of the months as the peak season.

The regression results appear in column 1 of Table S17. The main effect is negative and significant ($\beta = -6.277$, $p < 0.05$) and the interaction effect is also negative and significant ($\gamma = -11.01$, $p < 0.05$). The results suggest that the algorithm increases revenue through two mechanisms: (a) the general price correction, whereby the algorithm decreases the nightly rate by $6.23

[18] Although our occupancy data from AirDNA starts 18 months prior to the introduction of the algorithm, we used a narrower window (July 2015 to August 2017) in our analysis because AirDNA only started its collection of dynamic variables (e.g., number of reviews, number of photos, and other time-variant property characteristics) in July 2015.

throughout the year, and (b) the seasonal price correction, whereby the algorithm decreases the nightly rate by another $11.01 during the off-peak season, as compared to the peak season, which stems from the fact that hosts do not drop their prices enough during the off-peak season.

The results are robust to:

(i) the number of months included in the off-peak vs. peak season: 3 and 9 in the main analysis; 6 and 6 as well as 9 and 3 in subsequent analyses. The basic nature of results remains the same.

(ii) the number of seasons: 2 in the main analysis; 4 in a subsequent analysis. The basic nature of the results does not change.

We also note that by using the nightly rate as the DV, we avoid the limitation inherent to the occupancy data in the AirDNA dataset (see Section 2.5). AirDNA has accurate information on each property's nightly rate. This information is publicly available on each property's website and AirDNA got this information by scraping this information from each host's webpage.

We seek additional insight into the seasonal price correction by estimating equation (S21) again with two different dependent variables: the monthly occupancy rate and the average daily revenue in month *i*. The results are reported in columns 2 and 3 of Table S17. Starting with column 2,algorithm adoption increased occupancy in the peak season by 0.0710, and the effect did not change significantly in the off-peak season. This result suggests that the demand of an individual Airbnb unit is more responsive to prices during peak seasons than during off peak seasons.[19] Moving on to column 3,algorithm adoption increased average daily revenue during both the peak season (+$7.767) and the off-peak season (+$4.115, calculated as $7.767 - $3.352); the increase was significantly greater during the peak season.

**Table S17 Effects of the Interaction Between Algorithm Adoption and Season on Nightly Rate, Occupancy, and Revenue**

| | ESTIMATES | | |
|---|---|---|---|
| **VARIABLE** | **(1) Nightly Rate** | **(2) Occupancy Rate** | **(3) Daily Revenue** |
| ***SmartPricing*** (non off-peak season as default) | -6.277* | 0.0710*** | 7.467*** |
| | (3.094) | (0.00558) | (1.492) |
| ***SmartPricing X OffPeak*** | -11.01* | -0.0110 | -3.352* |
| | (5.041) | (0.00746) | (1.326) |
| *log Number of Reviews* | 5.267 | 0.0458*** | 11.91*** |
| | (7.590) | (0.00423) | (1.181) |
| *log Number of Photos* | -49.13 | 0.0492*** | 6.812* |
| | (33.11) | (0.0134) | (2.929) |
| *log Security Deposit* | 1.544*** | 0.0257*** | 6.087*** |
| | (0.447) | (0.000938) | (0.254) |
| *log # Min. Stays* | -3.601 | -0.0520*** | -10.44*** |
| | (3.536) | (0.00692) | (2.292) |
| *Instant Book Enabled* | 0.761 | 0.0637*** | 11.05*** |

[19] Note that there are more active Airbnb listings during the peak season than during the off-peak season (Farronato and Fradkin 2018). The increase in supply could lead to greater competition during the peak season, thereby making the demand for an individual unit more sensitive to changes in its nightly rate.

| | | | |
|---|---|---|---|
| | (2.071) | (0.00602) | (1.464) |
| *Super Host* | 4.605 | 0.0317*** | 9.050*** |
| | (3.001) | (0.00597) | (1.962) |
| *Professional Host (log # listings)* | 4.935 | 0.0131 | 0.351 |
| | (4.240) | (0.0114) | (4.264) |
| *Host Effort (Response Rate)* | 0.105 | -0.000530 | 0.0261 |
| | (0.0655) | (0.00146) | (0.0288) |
| Fixed Effect | Property | Property | Property |
| Seasonality | City-Year, City-Month | City-Year, City-Month | City-Year, City-Month |
| Observations | 162617 | 162617 | 162617 |
| R-squared | 0.85 | 0.56 | 0.51 |

Notes: We estimate Equation (S21) with three different dependent variables: nightly rate (column 1), occupancy rate (column 2), and daily revenue (column 3). We use the IPTW sample of 9,396 unique properties. The estimated coefficients of *SmartPricing X OffPeak* capture the extent to which the effect of algorithm adoption changes during the off-peak season (i.e., when demand is lower).
Cluster-robust standard errors are reported at the individual-property level in parentheses.
* p < 0.05 ** p < 0.01 *** p < 0.001

### 7.2 Seasonal Price Correction and the Differential Effect

Our main regression revealed that algorithm adoption led to a greater increase in average daily revenue among Black hosts than among white hosts. Specifically, the algorithm decreased the nightly rate by a similar amount among Black and white hosts, but this downward price correction led to a greater increase in occupancy for Black hosts than for white hosts.

We proposed that the reason is because Black and white hosts face different demand curves (all else being the same), with the demand of black hosts being more responsive to prices as compared to the demand of white hosts. Yet there is an alternative explanation: the downward price correction could be larger for Black hosts during the season when demand is more sensitive to prices (the peak season, according to Section 7.1) and vice versa for white hosts. As a result, adoption of the algorithm will lead to a greater increase in occupancy over the entire year (and thereby the revenue) for black hosts as compared to white hosts.

We explore the alternative explanation by adding a three-way interaction involving *Race* to equation (S21). We set *white* as the reference race. As in Section 7.1, we conduct the first regression with nightly rate as the DV:

$$\begin{aligned} NightlyRate_{it} &= \beta \cdot SmartPricing_{it} + \beta_1 \cdot SmartPricing_{it} \times Race_i + \beta_2 \\ &\cdot SmartPricing_{it} \times OffPeakSeason_{it} \times Race_i + \gamma \\ &\cdot SmartPricing_{it} \times OffPeakSeason_{it} + \gamma_1 \\ &\cdot OffPeakSeason_{it} \times Race_i + \lambda \cdot Controls_{it} + Seasonality_{it} \\ &+ Property_i + \varepsilon_{it} \end{aligned} \quad \text{(S22)}$$

The results are reported in column 1 of Table S18. The estimates of the coefficients of *SmartPricing* (-7.411) and *SmartPricing*×*OffPeak* (-12.09) are negative and significant ($p < 0.05$). In other words, for white hosts, algorithm adoption led to a general price correction as well as a seasonal price correction. We next examine whether the differences in seasonal price correction can explain why adoption of the algorithm benefits black hosts more than white hosts. The coefficient of interaction term, *OffPeak*×*Black* is not statistically significant. This implies that prior to adoption of the algorithm, both black and white hosts behaved similarly in terms of managing their prices in periods of low demand. The coefficient of *SmartPricing*×*Black* is statistically insignificant. This implies that that adoption of the algorithm led to a similar magnitude of price correction across black and white hosts during the peak season. The coefficient of the three-way interaction, *SmartPricing*×*Black*×*OffPeak,* is also not statistically significant, which implies that algorithm adoption led to a similar downward price correction for Black and white hosts during the off-peak season.

We conclude that seasonal variation in the price correction does not explain the differential effect of algorithm adoption among Black and white hosts. The results are robust to alternative operationalizations of the seasons (the number of months included in the off-peak vs. peak season and the number of seasons, see Section 7.1). The use of the nightly rate instead of the occupancy rate avoids a limitation inherent to the AirDNA data (see Sections 2.5 and 7.1).

Finally, we investigate whether the mechanism that we have discussed in the main paper (which is that the demand of black hosts is more responsive to price changes as compared to that of white hosts) explains the differential effect of the algorithm. We re-estimate equation (S22) with the occupancy rate as the DV. The results are reported in column 2 of Table S18. The main effect of algorithm adoption is significant and positive (0.0639, $p < 0.01$). The estimated coefficients of all interaction terms involving *Race* are non-significant except for *SmartPricing*×*Black*, which has a positive and significant coefficient (0.0763, $p < 0.01$), meaning that algorithm adoption increased the occupancy rate among white hosts by 0.0639 and among Black hosts by 0.1402 (= 0.0639 + 0.0763). We conclude that algorithm adoption benefitted Black hosts more than white hosts because the demand curve faced by Black hosts is more sensitive to price changes.

**Table S18 Effects of the Interactions Among Algorithm Adoption, Season, and Host Race on Nightly Rate and Occupancy**

| | **ESTIMATES** | |
|---|---|---|
| **VARIABLE** | **(1)<br>Nightly Rate** | **(2)<br>Occupancy Rate** |
| ***SmartPricing***<br>(Reference: *OffPeak* = 0)) | -7.411* | 0.0639*** |
| | (3.659) | (0.00608) |
| *SmartPricing X OffPeak* | -12.09* | -0.0140 |
| | (5.744) | (0.00819) |
| ***SmartPricing X Black***<br>(white race as reference) | 5.003 | 0.0763*** |
| | (5.388) | (0.0202) |
| *SmartPricing X Other* | 6.875 | 0.0205 |
| | (4.985) | (0.0156) |
| ***OffPeak X Black*** | -0.586 | -0.0181 |
| | (1.028) | (0.00988) |
| *OffPeak X Other* | 3.073 | 0.00390 |
| | (2.690) | (0.00935) |

| | | |
|---|---|---|
| ***SmartPricing X OffPeak X Black*** | 15.00 | -0.0195 |
| | (8.112) | (0.0289) |
| *SmartPricing X OffPeak X Other* | 1.737 | -0.0381 |
| | (7.039) | (0.0250) |
| *log Number of Reviews* | 5.238 | 0.0456*** |
| | (7.582) | (0.00424) |
| *log Number of Photos* | -49.13 | 0.0491*** |
| | (33.09) | (0.0133) |
| *log Security Deposit* | 1.540*** | 0.0257*** |
| | (0.448) | (0.000938) |
| *log # Min. Stays* | -3.595 | -0.0521*** |
| | (3.536) | (0.00694) |
| *Instant Book Enabled* | 0.740 | 0.0640*** |
| | (2.069) | (0.00603) |
| *Super Host* | 4.691 | 0.0321*** |
| | (3.029) | (0.00599) |
| *Professional Host (log # listings)* | 4.920 | -0.0132 |
| | (4.231) | (0.0114) |
| *Host Effort (Response Rate)* | 0.105 | 0.000529 |
| | (0.0654) | (0.00146) |
| Fixed Effect | Property | Property |
| Seasonality | City-Year, City-Month | City-Year, City-Month |
| Observations | 162617 | 162617 |
| R-squared | 0.85 | 0.56 |

Notes: We estimate Equation (S22) with two different dependent variables: nightly rate (column 1) and occupancy rate (column 2). We use the IPTW sample of 9,396 unique properties. The estimated coefficients of the interaction term, *SmartPricing x OffPeak x Black,* indicate whether the differential effect of algorithm adoption among Black vs. white hosts is explained by the differential effect of algorithm adoption by season.
Cluster-robust standard errors are reported at the individual-property level in parentheses.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

## Section 8. Exploring Policy Implications

We present analyses that inform two policy recommendations for Airbnb to increase the algorithm's ability to close the revenue gap between white and Black hosts: 1) instead of including race directly in the algorithm, which specific non-race characteristics (that are correlated with race) can Airbnb add in their algorithm? 2) which socioeconomic segment(s) of black hosts Airbnb can target in order to encourage them to adopt the algorithm?

### 8.1. Adding Characteristics that Correlate with Race

Our main analyses suggest that algorithm adoption led to a greater increase in the average daily revenue for Black hosts, compared to white hosts, because Black hosts and white hosts face different demand curves. It follows that the pricing algorithm, if incorporate race to capture this difference in the demand curves, might further mitigate the racial disparity. Instead of directly including race in the algorithm, Airbnb can include alternative demographic/socio-economic/non-race variables that are correlated with race in their algorithm.

We identify candidate variables by regressing *Black* on the entire set of covariates from our propensity score estimation. We report the results in Table S19. We find the highest correlations between *Black* and (i) % of adults in the same zip-code with a bachelor's degree, (ii) number of Airbnb properties listed by the host, (iii) median household income in the zip-code, (iv) whether or not the entire home was available for rent, (v) average number of blocked days per month, (vi) number of minutes from downtown by car, (vii) access to public transit (transit score), and (viii) whether the property had heating and a refrigerator. This analysis suggests that adding these variables in the algorithm can reduce the racial gap.

However, there are two caveats in place here. For one, McFadden's pseudo $R^2$ is 0.1, which is neither especially small[20] nor large. Although the inclusion of the aforementioned covariates in the algorithm would reduce the revenue gap, the effect would be smaller than if Airbnb were to include race in its algorithm. For another, several of the aforementioned variables pertain to socioeconomic status, which may still be problematic under US policy because of the potential for disparate impacts across racial groups (Barocas and Selbst 2016).[21]

**Table S19 Regression of *Black* on Other Covariates**

| VARIABLE | Estimate | Std. Err. |
|---|---|---|
| # Bedrooms | -0.0724 | (0.0578) |
| Apartment | 0.00660 | (0.0984) |
| Entire Home | -0.217* | (0.0856) |
| Listing Title Length | -0.0248*** | (0.00560) |
| Number of Photos | 0.00191 | (0.00352) |
| Number of Reviews | -0.00441* | (0.00173) |
| Listing Nightly Rate | 0.000121 | (0.000337) |

[20] We do not consider the value of 0.1 to be low because Black hosts constitute only 10% of the sample. It is difficult to get a high value of pseudo $R^2$ without more equal representation.

[21] Both disparate treatment by the algorithm (e.g., the inclusion of race in the algorithm) and disparate impact of the algorithm (e.g., the algorithm includes a measure of SES, and the algorithm has a different impact on Black vs. white hosts) are illegal.

| | | |
|---|---|---|
| # Minimum Stay | -0.00683 | (0.0146) |
| Security Deposit | -0.000165 | (0.000134) |
| # Blocked Days per Month | -0.0292*** | (0.00356) |
| # Reservation Days | -0.0412*** | (0.00536) |
| Median Home Earning (*1000 USD*) | -0.00426* | (0.00214) |
| Private Parking | 0.0770 | (0.0932) |
| Pool | -0.157 | (0.187) |
| Iron | 0.0225 | (0.140) |
| Internet | -0.606* | (0.273) |
| TV | 0.359*** | (0.0884) |
| Dryer | -0.184 | (0.132) |
| Washer | -0.0407 | (0.119) |
| Beach Nearby | -0.173 | (0.480) |
| Essentials | -0.125 | (0.137) |
| Heating | -0.484*** | (0.138) |
| Microwave | 1.023 | (0.577) |
| Refrigerator | -1.010** | (0.371) |
| Laptop Friendly | -0.111 | (0.133) |
| Fireplace | 0.177 | (0.124) |
| Elevator | 0.177 | (0.100) |
| Gym | -0.0404 | (0.160) |
| Family Friendly | 0.0608 | (0.0982) |
| Smoker Detector | 0.344** | (0.128) |
| Shampoo | -0.191 | (0.117) |
| Breakfast | 0.00380 | (0.149) |
| AC | 1.507 | (0.866) |
| # Photographed Faces | 0.142*** | (0.0414) |
| Walk Score | -0.00198 | (0.00255) |
| Transit Score | 0.0185*** | (0.00345) |
| Drive to Downtown (*min*) | 0.0169*** | (0.00400) |
| Population Density (*Per Sq. Mile*) | -0.00000160 | (0.00000159) |
| Graduate (%) | 0.0318 | (0.113) |
| Bachelor (%) | -0.0283*** | (0.00682) |
| Host Age | -0.0327*** | (0.00424) |
| Home Value (1000 USD) | -0.0000422 | (0.000104) |
| Number of months since the property has been listed | 0.00670* | (0.00262) |
| Number of properties owned by the host | -0.0166** | (0.00567) |
| McFadden's $R^2$ | 0.100 | |
| Notes: The logistic regression assesses the power of observed covariates (the same list as in the PSM, Table S3) to predict the host's race. The DV is *Black*, and the estimation excludes samples with the host's race categorized as *Other*. Standard errors are in parentheses. * p < 0.05 ** p < 0.01 *** p < 0.001 | | |

### 8.2. Targeting/Encouraging Black Hosts to Adopt the Algorithm

We identify which socioeconomic segment(s) of Black hosts might benefit most from a targeted algorithm adoption campaign. We segmented hosts into four quartiles based on socioeconomic status (SES) at the neighborhood level. We use education as a proxy for SES and use US Census data (ACS, American Community Survey) to determine the percentage of adults within the host's zip-code who have a bachelor's degree, conditional on the host's race. We segmented the properties into quartiles (q = 1–4; higher values indicate a higher percentage of adults with a bachelor's degree).

Next, we conducted a two-step analysis:

(a) a separate IPTW+DiD analysis for each quartile, regressing average daily revenue over a two-way interaction: *SmartPricing x Black*[22]

(b) a logit regression of algorithm adoption regressed over a two-way interaction between *Black* and each quartile, controlling for all other variables in the adoption probability model (i.e., the IPTW logistic regression)

The results of the DiD regression are reported in Table S19. The main effect (i.e., the coefficient of *SmartPricing*) is positive and significant for quartiles q = 1, 2, and 3, but not 4; the magnitude is highest for quartile q = 3, followed by 1 and then 2. The differential effect (i.e., the coefficient of *SmartPricing*×*Black*) is positive and significant for q = 3 only. We speculate that the main effect may be insignificant for q = 4 because highly-educated hosts (the upper most quartile of education) may be more proficient at pricing their properties, so algorithm adoption does not significantly increase their revenue.

The results of the logit regression are reported in Table S21. The main effect of *Black* on algorithm adoption is negative and significant. However, most interactions involving *Black* are insignificant, with the exception of *Black* x *q = 4*. Algorithm adoption among Black hosts is highest for quartile q = 1, followed by 2, 3, and 4.

We note two nuances that inform the ideal segments of Black hosts for a targeted algorithm adoption campaign. Although the rate of algorithm adoption is lowest in q = 4, Black hosts in q = 4 also did not experience a significant increase in revenue by adopting the algorithm, so an algorithm adoption campaign in this population would not be expected to narrow the revenue gap.[23] The next-lowest rate of adoption among Black hosts is in q = 3, and Black hosts in this quartile also gained the largest increase in revenue by adopting the algorithm. We conclude that if Airbnb wishes to address the racial revenue gap by promoting more widespread algorithm adoption among Black hosts, it would be most efficient to target hosts in q = 3, followed by those in q = 1 and 2.

---

[22] We also performed a single IPTW+DiD regression on all four quartiles, regressing average daily revenue over a three-way interaction: *q x Black x SmartPricing*. The results were qualitatively similar to the results presented here.

[23] Hosts in q = 4 nevertheless may choose to adopt the algorithm because it is convenient that they do not have to decide and manually set prices every day.

**Table S20 IPTW and DiD Analysis: Effect of Algorithm Adoption on Average Daily Revenue by SES Quartile**

| VARIABLE | Quartile: q = 1 (bottom) | Quartile: q = 2 | Quartile: q = 3 | Quartile: q = 4 (top) |
|---|---|---|---|---|
| ***SmartPricing*** | 6.751* | 5.664* | 9.216*** | 3.244 |
| | (3.353) | (2.463) | (2.701) | (4.478) |
| ***SmartPricing×Black*** | 0.132 | 2.814 | 15.22* | 14.41 |
| | (4.928) | (4.321) | (6.245) | (17.58) |
| *SmartPricing×Other* | 7.695 | 4.902 | 2.933 | 12.00 |
| | (5.235) | (6.054) | (7.671) | (9.696) |
| *Log Number_of_Reviews* | 9.353*** | 11.98*** | 9.856*** | 14.25*** |
| | (1.588) | (2.099) | (2.608) | (3.027) |
| *Log Number_of_Photos* | 1.788 | 5.975 | 15.05*** | 10.76 |
| | (2.669) | (5.064) | (4.285) | (7.273) |
| *Log Security_Deposit* | 6.167*** | 5.192*** | 6.351*** | 6.201*** |
| | (0.465) | (0.460) | (0.423) | (0.538) |
| *Log #Min. Stays* | -15.87*** | -13.42*** | -3.010 | -11.66* |
| | (3.916) | (3.042) | (6.641) | (4.756) |
| *Instant_Book_Enabled* | 7.299** | 16.80*** | 13.62*** | 10.74 |
| | (2.299) | (2.849) | (2.781) | (6.926) |
| *Super_Host* | 8.747*** | 4.259 | 7.017 | 19.24* |
| | (2.244) | (2.734) | (3.964) | (7.530) |
| *Log#host-owned listings* | 17.56* | -1.008 | 3.917 | -16.51 |
| | (8.067) | (4.761) | (5.189) | (12.30) |
| *Host_Effort (Response_Rate)* | 0.0637 | -0.0887 | -0.0868 | 0.101 |
| | (0.0547) | (0.0842) | (0.0501) | (0.0760) |
| Fixed Effect | Property | Property | Property | Property |
| Seasonality | City-Year, City-Month | City-Year, City-Month | City-Year, City-Month | City-Year, City-Month |
| Observations | 46217 | 39712 | 40540 | 36148 |
| R-squared | 0.55 | 0.61 | 0.47 | 0.49 |

Notes: The results reflect the main effects (coefficients of *SmartPricing)* and differential effects (coefficients of *SmartPricing x Black)* of algorithm adoption on average daily revenue across four SES quartiles. Results are calculated with a separate IPTW and DiD analysis for each quartile. The quartile boundaries are determined at the city-level, and each property's quartile is determined by educational attainment in the property's zip-code. The number of observations vary because the sample size of each quartile varied slightly, and all properties with a full month blocked were discarded (as average daily revenue = average nightly price x occupancy rate would be undefinable).
Cluster-robust standard errors are reported at the individual-property level in parentheses.
* p<0.05 ** p<0.01 *** p<0.001

**Table S21 Logit Regression: Effect of Race and Education Quartile on Algorithm Adoption**

| | Estimate (Std. Error) | |
|---|---|---|
| Black | -0.535*** | (0.148) |
| Other | -0.227 | (0.160) |
| q = 2 | -0.162* | (0.0822) |
| q = 3 | -0.257** | (0.0831) |
| q = 4 | -0.128 | (0.0875) |
| Black x q = 2 | -0.106 | (0.244) |
| Black x q = 3 | 0.0739 | (0.272) |
| Black x q = 4 | -1.100** | (0.377) |
| Other x q = 2 | -0.120 | (0.235) |
| Other x q = 3 | -0.127 | (0.244) |
| Other x q = 4 | -0.488 | (0.255) |
| Bedrooms | -0.0819 | (0.0460) |
| Apartment | -0.167* | (0.0708) |
| Entire Home | -0.356*** | (0.0678) |
| Listing Title Length | 0.0129** | (0.00410) |
| Number of Reviews | 0.00716** | (0.00247) |
| Number of Photos | 0.00000811 | (0.000960) |
| Listing Nightly Rate | -0.000101 | (0.000386) |
| # Minimum Stay | -0.00257 | (0.0116) |
| Security Deposit | 0.0000367 | (0.0000919) |
| #Blocked Days | -0.0268*** | (0.00295) |
| #Reservation Days | 0.000811 | (0.00372) |
| Parking | 0.156* | (0.0685) |
| Pool | 0.0970 | (0.128) |
| Iron | 0.494*** | (0.0985) |
| Internet | 0.0806 | (0.284) |
| TV | -0.0893 | (0.0649) |
| Dryer | 0.0893 | (0.0990) |
| Washer | -0.188* | (0.0784) |
| Beach | -0.0892 | (0.214) |
| Essentials | 0.346** | (0.111) |
| Heating | 0.375** | (0.140) |
| Microwave | 0.0460 | (0.162) |
| Refrigerator | 0.178 | (0.157) |
| Laptop-Friendly | 0.149 | (0.0920) |
| Fireplace | -0.162 | (0.0855) |
| Elevator | -0.147 | (0.0837) |
| Gym | -0.0999 | (0.130) |

| | | |
|---|---|---|
| Family-Friendly | 0.219** | (0.0814) |
| Smoker Detector | 0.332** | (0.102) |
| Shampoo | -0.0459 | (0.0853) |
| Breakfast | 0.0273 | (0.110) |
| AC | 0.137 | (0.571) |
| # Photographed Faces | 0.0190 | (0.0361) |
| Walk Score | -0.00278 | (0.00165) |
| Transit Score | 0.00751*** | (0.00227) |
| Drive to Downtown (*min*) | -0.00165 | (0.00300) |
| Population Density (Per Sq. | 0.00000025 | (0.00000125) |
| Age | -0.00706* | (0.00278) |
| # Listed Month | -0.00469* | (0.00207) |
| # Host-Owned Listings | -0.0392*** | (0.00769) |

Notes: The results reflect the main effects of race (*Black*), education quartile (*q*), and their interaction (*Black x q*) on the algorithm adoption, estimated with a logit regression. The quartile boundaries are determined at the city-level, and each property's quartile is determined by educational attainment in the property's zip-code. We use the same set of variables as in the propensity score model (see Table S3). The estimated coefficients q captures how the different education level is related to the different adoption rate of Smart Pricing. The estimated coefficient of interaction terms *Black x q* assesses if the adoption rates among the Black hosts vary across different education quartiles.
Standard errors in parentheses. * $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$.

## Section 9. References


Abadie, A., A. Diamond, and J. Hainmueller (2010). Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program. Journal of the American Statistical Association 105 (490): 493-505.

Abadie, A., A. Diamond, and J. Hainmueller (2015). Comparative politics and the synthetic control method. American Journal of Political Science 59 (2): 495-510.

Austin PC. The performance of different propensity score methods for estimating marginal hazard ratios. Stat Med. 2013;32(16):2837–2849. doi:10.1002/sim.5705

Austin, P.C., E.A. Stuart, Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. *Statistics in Medicine* **34**, 3661–3679 (2015).

Autor, D.H. (2003), "Outsourcing at Will: The Contribution of Unjust Dismissal Doctrine to the Growth of Employment Outsourcing," *Journal of Labor Economics* 21(1):1-42

Barocas, S., and A. Selbst. 2016. Big Data' s Disparate Impact. California law review 104 (1): 671-729.

Cao, H., H. Bernard, L. Heutte, R. Sabourin, Improve the Performance of Transfer Learning Without Fine-Tuning Using Dissimilarity-Based Multi-view Learning for Breast Cancer Histology Images. In: Campilho A., Karray F., ter Haar Romeny B. (eds*) Image Analysis and Recognition. ICIAR* (2018).

Chamberlain, G., 1982, "Multivariate Regression Models for Panel Data," J*ournal of Econometrics*, 18 (1), 5–46

Chen, X., S. Xiang, C. Liu, C. Pan, Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett*. **11**, 1797–1801 (2014).

Dehejia, Rajeev, and Sadek Wahba. 1999. "Causal Effects in Nonexperimental Studies: Re-evaluating the Evaluation of Training Programs." Journal of the American Statistical Association 94 (448): 1053–62.

DiPrete, T. A. and Gangl, M. (2004), Assessing Bias in the Estimation of Causal Effects: Rosenbaum Bounds on Matching Estimators and Instrumental Variables Estimation with Imperfect Instruments. *Sociological Methodology*, 34: 271-310

Farronato, Chiara and Fradkin, Andrey. (2018). The Welfare Effects of Peer Entry in the Accommodation Market: The Case of Airbnb. Available at: https://ssrn.com/abstract=3131140

Fradkin, Andrey, Search, Matching, and the Role of Digital Marketplace Design in Enabling Trade: Evidence from Airbnb (March 21, 2017). https://ssrn.com/abstract=2939084

Gibbs, C., Guttentag, D., Gretzel, U. and L. Yao (2017), “Use of Dynamic Pricing Strategies by Airbnb Hosts,” International Journal of Contemporary Hospitality Management, 3(1), 2-20.

Glorot, X., Y. Bengio, Understanding the difficulty of training deep feedforward neural networks. *In Proc. AISTATS* **9**, 249–256 (2010).

Guo, S. & Fraser, M. W. (2015). Propensity score analysis: Statistical methods and applications (2nd ed.). Thousand Oaks, CA: Sage Publications, Inc.

Hansen, Lars P. (1982). Large Sample Properties of Generalized Method of Moments Estimators, *Econometrica*, 50, 1029-1054.

He, H., X. Zhang, S. Ren, J. Sun, J. Deep Residual Learning for Image Recognition. *IEEE Conference* on, ISSN: 1063-6919 (2016).

Ichino, A., F. Mealli, T. Nannicini 2008: From Temporary Help Jobs to Permanent Employment: What can We Learn From Matching Estimators and Their Sensitivity? Journal of Applied Econometrics, 23, 305--327.

Imbens, Guido, W. 2003. "Sensitivity to Exogeneity Assumptions in Program Evaluation." American Economic Review, 93 (2): 126-132.

Krizhevsky, A., I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks. (2012). 25: 1097–1105 (2012).

LaLonde, Robert. 1986. “Evaluating the Econometric Evaluations of Training Programs with Experimental Data.” The American Economic Review 76 (4). American Economic Association: 604–20.

LeCun, Y., L. Bottou, P. Haffner, Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86** (11), 2278–2324 (1998).

Lee, J., K. Cho, T. Hofmann, Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Transactions of the Association for Computational Linguistics* **5**, 365-378 (2017).

Manchanda, P., Packard, G., and A. Pattabhiramaiah, (2015), “Social Dollars: The Economic Impact of Customer Participation in a Firm-Sponsored Online Customer Community,” *Marketing Science*, 34 (3), 367-387.

Masten, Matthew, and Alexandre Poirier. 2018. “Identification of Treatment Effects Under Conditional Partial Independence.” Econometrica 86 (1): 317–51.

Masten, M.A. and Poirier, A. (2019). "tesensitivity: A Stata Package for Assessing the Unconfoundedness Assumption," 2019 Stata Conference 51, Stata Users Group.

Masten, M.A. and Poirier, A. (2020), Inference on breakdown frontiers. Quantitative Economics, 11: 41-111. doi:10.3982/QE1288.

Nevo, A. (2000), "A Practitioner's guide to the estimation of random coefficients logit model of demand," *Journal of Economics and Management Strategy*, 9(4), 513-548

Nevo, A. (2001). Measuring Market Power in the Ready-to-Eat Cereal Industry. *Econometrica* 69, no. 2: 307-42.

Newey, W.K. and D. F. McFadden (1994). Large sample estimation and hypothesis testing, in R.F. Engle III and D.F. McFadden (eds.), *The Handbook of Econometrics*, vol. 4. North-Holland, Amsterdam.

Rosenbaum, P.R., D.B. Rubin, The central role of the propensity score in observational studies for causal effects. *Biometrika* **70**, 41–55. (1983).

Rosenbaum, P.R., Observational Studies. 2nd ed. (New York: Springer, 2002).

Rothe, R., R. Timofte, L. V. Gool, DEX: Deep EXpectation of Apparent Age from a Single Image. *IEEE International Conference on Computer Vision Workshop (ICCVW)* Electronic ISBN: 978-1-4673-9711-7 (2015).

Rubin, D.B., Using propensity scores to help design observational studies: application to the tobacco litigation. *Health Services & Outcomes Research Methodology* **2**,169–188. (2001).

Simonyan, K., A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. *The 3rd International Conference on Learning Representations (ICLR)* (2015).

Stuart, E.A., Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics* **25**, 1-21. (2010).

Sun, M., F. Zhu, Ad Revenue and Content Commercialization: Evidence from Blogs. *Management Science* **59**, 2314-2331 (2013).

Szegedy, C., W. Liu, Y.Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper With Convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

Xu, Y. (2017). Generalized synthetic control method: Causal inference with interactive fixed effects models. Political Analysis 25 (1):57-76.

Ye, P., Qian, J., Chen, J., Wu, C.H., Zhou, Y., Mars, S.D., Yang, F. and L. Zhang (2018), "Customized Regression Model for Airbnb Dynamic Pricing," KDD 18, 19–23.

Zalmanson, Lior and Proserpio, Davide and Nitzan, Irit, Cancellation Policy as a Signal of Trust and Quality in the Sharing Economy: The Case of Airbnb (November 28, 2018).

Zeiler. M.D., ADADELTA: An Adaptive Learning Rate Method. *CoRR* abs/1212.5701 (2012).